



# elektor

## electronics

DOUBLE ISSUE



Sensors

Actuators

Microcontrollers

Power Supplies & Chargers

Mechanics

Communications

Tips & Tricks

Miscellaneous



# The Best in Electronic Kits



675+ New Products

430+ Pages Full Colour

Visit us at [www.jaycarelectronics.co.uk](http://www.jaycarelectronics.co.uk) and obtain your free copy of our 430+ page, full colour catalogue. We have an extensive range of electronics kits and the best in innovative gadgets.



## Jacob's Ladder High Voltage Display Kit

**KC-5445** £11.75 + post & packing  
With this kit and the purchase of a 12V ignition coil (available from auto stores and parts recyclers), create an awesome rising ladder of noisy sparks that emits the distinct smell of ozone. This improved circuit is suited to modern high power ignition coils and will deliver a spectacular visual display. Kit includes PCB, pre-cut wire/ladder and all electronic components.

- 12V car battery or >5Amp DC power supply required



Improved model for 2007

HIGH VOLTAGE



## 4 Channel Guitar Amplifier Kit

**KC-5448** £28.75 + post & packing  
The input sensitivity of each of the four channels is adjustable from a few millivolts to over 1 volt, so you can plug in a range of input signals from a microphone to a line level signal from a CD player etc. A headphone amplifier circuit is also included for monitoring purposes. A three stage EQ is also integrated, making this a very versatile mixer that will operate from 12VDC. Kit includes PCB with overlay and all electronic components.



## Stereo VU and Peak Meter Kit

**KC-5447** £20.50 + post & packing  
Accurately monitor audio signals to prevent signal clipping and ensure optimum recording levels. This unit is very responsive and uses two 16-segment bargraphs to display signal levels and transient peaks in real time. There are a number of display options to select, and both the signal threshold and signal level calibration for each segment are adjustable. Kit supplied with PCBs, LCD and all electronic components. Accuracy within 1dB for signals above -40dB.

- Case not included use HB-6082 £2.95



Short form kit

## 50MHz Frequency Meter MKII Kit

**KC-5440** £20.50 + post & packing  
This compact, low cost 50MHz Frequency Meter is invaluable for servicing and diagnostics. This upgraded version features an automatic indication of units (Hz, kHz, MHz or GHz) and prescaler. Kit includes PCB with overlay, enclosure, LCD and all electronic components.

- 8 digit reading (LCD)
- Prescaler switch
- 3 resolution modes
- Powered by 5 x AAA batteries or DC plugpack

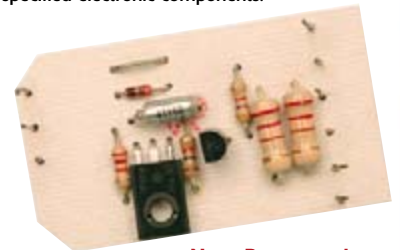


Improved model for 2007

Requires 5VDC wall adaptor (Maplin L66BQ £7.79)

## Variable Boost Kit for Turbochargers

**KC-5438** £6.00 + post & packing  
It's a very simple circuit with only a few components to modify the factory boost levels. It works by intercepting the boost signal from the car's engine management computer and modifying the duty cycle of the solenoid signal. Kit supplied in short form with PCB and overlay, and all specified electronic components.



Note: Prototype shown

## Programmable High Energy Ignition System

### Ignition System

**KC-5442** £26.25 + post & packing  
This advanced and versatile ignition system can be used on both two & four stroke engines. The system can be used to modify the factory ignition timing or as the basis for a stand-alone ignition system with variable ignition timing, electronic coil control and anti-knock sensing. Kit supplied with PCB, diecast case and all electronic components.

Features include:

- Timing retard & advance over a wide range
- Suitable for single coil systems
- Dwell adjustment
- Single or dual mapping ranges
- Max & min RPM adjustment
- Optional knock sensing
- Optional coil driver

### Hand Controller

**KC-5386** £25.95 + post & packing  
This LCD hand controller is required during the initial setting-up procedure. It plugs into the main unit and can be used while the engine is either running or stopped. Using this Hand Controller, you can set all the initial parameters and also program the ignition advance/retard curve. Kit supplied with silk screened and machined case, PCB, LCD, and all electronic components.

### KC-5386 Hand Controller



### Ignition Coil Driver

**KC-5443** £13.00 + post & packing  
Add this ignition coil driver to the KC-5442 Programmable Ignition System and you have a complete stand-alone ignition system that will trigger from a range of sources including points, Hall Effect sensors, optical sensors, or the 5 volt signal from the car's ECU. Kit includes PCB with overlay and all specified components.

### KC-5442 Ignition System



KC5443 Coil Driver

### Knock Sensor

**KC-5444** £5.00 + post & packing  
Add this option to your KC-5442 Programmable High Energy Ignition system and the unit will automatically retard the ignition timing if knocking is detected. Ideal for high performance cars running high octane fuel. Requires a knock sensor which is cheaply available from most auto recyclers. Kit supplied with PCB, and all electronic components.

Log on to [www.jaycarelectronics.co.uk/elektor](http://www.jaycarelectronics.co.uk/elektor) for your FREE catalogue!

0800 032 7241

(Monday - Friday 09.00 to 17.30 GMT + 10 hours only).  
For those who want to write: 100 Silverwater Rd  
Silverwater NSW 2128 Sydney AUSTRALIA

All prices in £ Stg

Free 430+ page Catalogue



# Jaycar Electronics

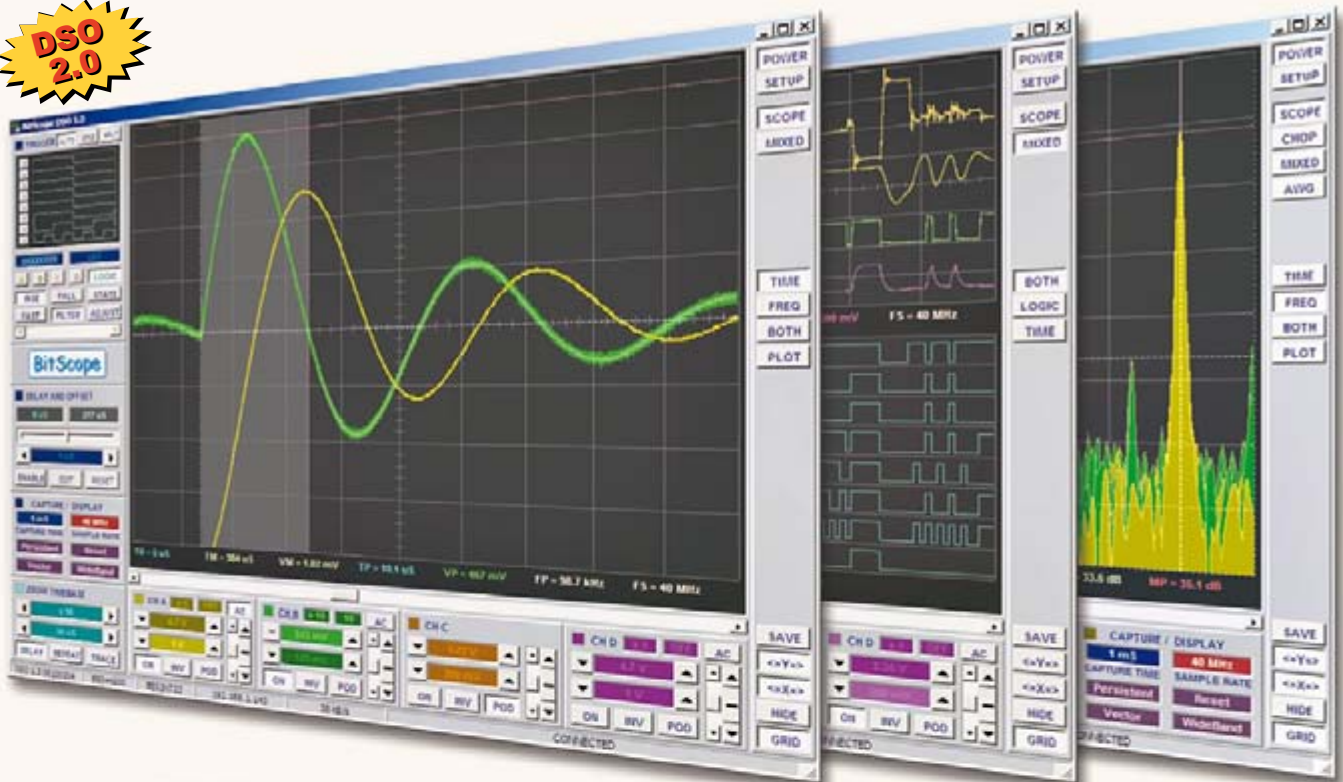
[www.jaycarelectronics.co.uk](http://www.jaycarelectronics.co.uk)



# BitScope PC Oscilloscopes & Analyzers

## DSO Test Instrument Software for BitScope Mixed Signal Oscilloscopes

DSO 2.0



4 Channel BitScope



2 Channel BitScope



Pocket Analyzer

### Digital Storage Oscilloscope

- ✓ Up to 4 analog channels using industry standard probes or POD connected analog inputs.

### Mixed Signal Oscilloscope

- ✓ Capture and display up to 4 analog and 8 logic channels with sophisticated cross-triggers.

### Spectrum Analyzer

- ✓ Integrated real-time spectrum analyzer for each analog channel with concurrent waveform display.

### Logic Analyzer

- ✓ 8 logic, External Trigger and special purpose inputs to capture digital signals down to 25nS.

### Data Recorder

- ✓ Record anything DSO can capture. Supports live data replay and display export.

### Networking

- ✓ Flexible network connectivity supporting multi-scope operation, remote monitoring and data acquisition.

### Data Export

- ✓ Export data with DSO using portable CSV files or use libraries to build custom BitScope solutions.

## BitScope DSO Software for Windows and Linux

BitScope DSO is fast and intuitive multi-channel test and measurement software for your PC or notebook. Whether it's a digital scope, spectrum analyzer, mixed signal scope, logic analyzer, waveform generator or data recorder, BitScope DSO supports them all.

Capture deep buffer one-shots or display waveforms live just like an analog scope. Comprehensive test instrument integration means you can view the same data in different ways simultaneously at the click of a button.

DSO may even be used stand-alone to share data with colleagues, students or customers. Waveforms may be exported as portable image files or live captures replayed on other PCs as if a BitScope was locally connected.

BitScope DSO supports all current BitScope models, auto-configures when it connects and can manage multiple BitScopes concurrently. No manual setup is normally required. Data export is available for use with third party software tools and BitScope's networked data acquisition capabilities are fully supported.



[www.bitscope.com](http://www.bitscope.com)

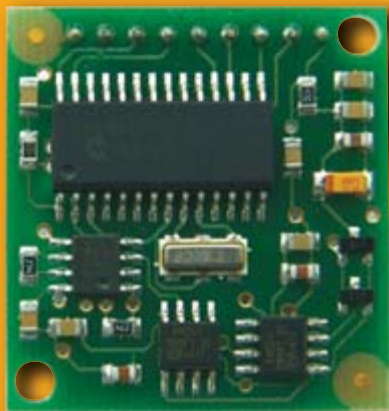
# Summer Circuits 2007

## Contents

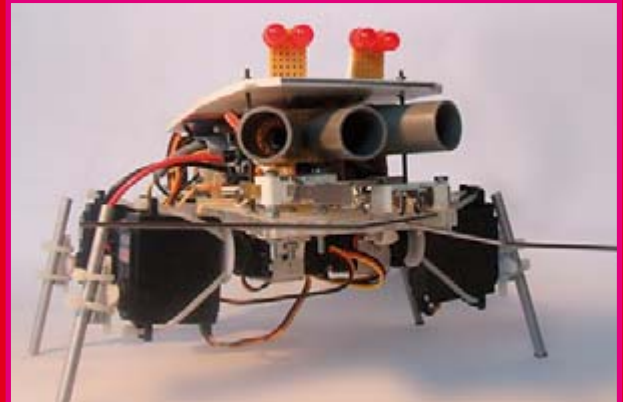
Colophon	6
Alphanumski Puzzle	140
Elektor SHOP	145
Sneak Preview	148

## SENSORS

A/D Converter for Robots	25
Bat's Ear	30
CMUCam1 Vision System	18
CO Sensor	35
Compass Sensor for Lego Mindstorms NXT	36
From Cassette Recorder to Robot Propulsion	26
An Inclinometer for your Robot	32



IR Close Object Detector	27
Light Sensing with an LED	38
Light-seeking Robot	29
An Obstacle Detecting Robot	10
Overheat Detector Alarm/Switch	21
PIC or Basic Stamp IR Telemeter	21
PIR Sensor	28
Positioning with Photodiode Arrays	33
A Robot that won't lose its bearings	24
Sensor for Line Following Robots	20
Simple D/A Converter for Robots	17
Sound Activated Switch	23
Stereo Robot Ears*	13
Ultrasonic Distant Obstacle Detector	37
Whiskers on Robots	34
Wireless Pulse Sensor	12



## ACTUATORS

12 V Bidirectional Motor Control	44
3 Amp PWM DC Motor Controller*	50
Catapult for Robots... or Other Uses	56
Complete Stepper Motor Driver	46
Controlling Servos	47
Driving Higher Power Motors	51
Driving Stepper Motors	59
Driving Stepper Motors: KISS	53
MotoBox*	40
PIC 12C508 Stepper Motor Controller	48
PIC Indicator Relay	55
Robot Footballer	54
Servo to Motor Conversion	58
zBot: 10-A Power Stage for DC Motor	45

## MICROCONTROLLERS

Low <sup>2</sup> Cost USB Demo Board	75
LPC900 Programmer*	72



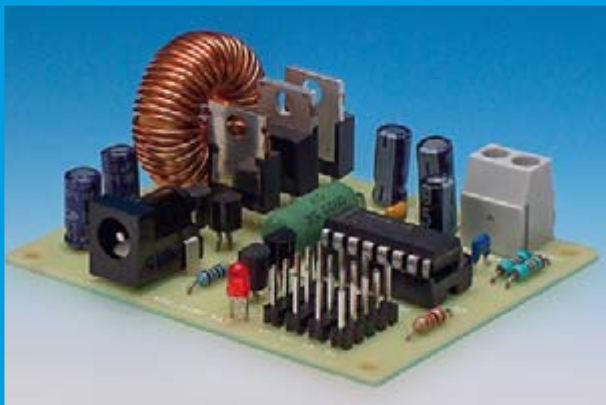
Optimised STK200/300 Programmer	77
Propeller Prototyping Board for BoeBot*	69



Satnav for Robots	78
Serial Interface for the Propeller*	81
Servo Control from a PC	68
'Teaclipper' Postage-Stamp Programmer	71
Which Brain for my Robot?	62

## POWER SUPPLIES & CHARGERS

3-A Wide-input Adjustable Switching Regulator	84
Deep Discharge Protection for Rechargeable Cells	94
Dual Battery*	90
Fast Charger for NiMH Batteries*	93
LDO Regulator with Soft Start or Tracking	95
Lithium Charger*	88
Mini Power Inverter	87
Multi-purpose NiCd & NiMH Charger	91



Paralleling LiPo Batteries	85
PWM Voltage Dropper	89
Switch-Mode 555 Supply	86
USB Converter*	82
Voltage Stabiliser	86
zBot: Solar/Battery Power Supply	83

## MECHANICS

Antieu-robot	99
Babybot	100
Bolo	98
Hunter	104
Minimalist Motor	102
Robot MOPS	107
Torque is Cheap	101
Trembly	104
Tribot	103

## COMMUNICATIONS

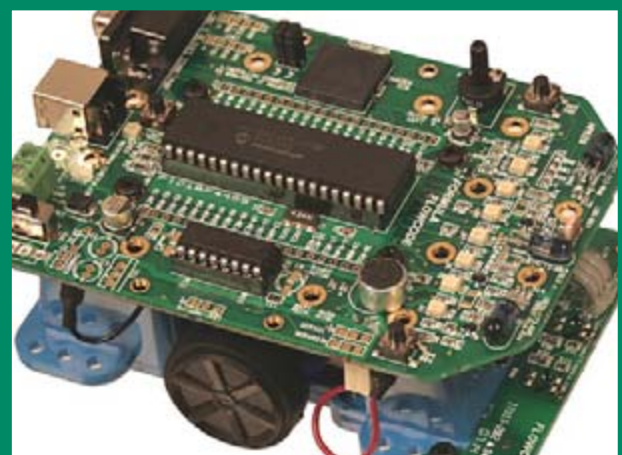
2.4 GHz Antenna for Robotics Vehicles	110
IR Communications using a UART	113
IR Remote Control with the R8C	115
Radio Remote Control for PDAs and Smartphones	115
Receiver for RC5 Remote Controls	112
Removed Pulse Detector	114
Robot Voice	117
zBot: Wireless Link	114

## TIPS & TRICKS

Balancing Robot	118
DIY Wheels	120
R8C Measures Negative Voltages	121

## MISCELLANEOUS

Chaotic Fireflies	126
Clap Controlled Switcher	132
Colour Tamer	126
Football with Robots	138
Formula Flowcode Buggy	122
KiCAD: a high-level tool	134
New Lego Mindstorms NXT Motor Block	133
Philips ME Construction Kits	130



A Robot with an Elephant's Memory	131
Stepped Volume Control	127
Walking Works!	128

\* PCB design included



Volume 33, Number 367/368, July/August 2007 ISSN 0268/4519

*Elektor Electronics* aims at inspiring people to master electronics at any personal level by presenting construction projects and spotting developments in electronics and information technology.

**Publishers:** Elektor Electronics (Publishing), Regus Brentford, 1000 Great West Road, Brentford TW8 9HH, England. Tel. (+44) 208 261 4509, fax: (+44) 208 261 4447 [www.elektor-electronics.co.uk](http://www.elektor-electronics.co.uk)

The magazine is available from newsagents, bookshops and electronics retail outlets, or on subscription. *Elektor Electronics* is published 11 times a year with a double issue for July & August.

Under the name *Elektor* and *Elektuur*, the magazine is also published in French, Spanish, German and Dutch. Together with franchised editions the magazine is on circulation in more than 50 countries.

**International Editor:** Mat Heffels ([m.heffels@segment.nl](mailto:m.heffels@segment.nl)), Wisse Hettinga ([w.hettinga@segment.nl](mailto:w.hettinga@segment.nl))

**Editor:** Jan Buiting ([editor@elektor-electronics.co.uk](mailto:editor@elektor-electronics.co.uk))

**International editorial staff:** Harry Baggen, Thijs Beckers, Ernst Krempelsauer, Jens Nickel, Guy Raedersdorf.

**Design staff:** Antoine Authier, Ton Giesberts, Paul Goossens, Luc Lemmens, Jan Visser, Christian Vossen

**Editorial secretariat:** Hedwig Hennekens ([secretariaat@segment.nl](mailto:secretariaat@segment.nl))

**Graphic design / DTP:** Giel Dols, Mart Schroijgen

**Managing Director / Publisher:** Paul Snakkers

**Marketing:** Carlo van Nistelrooy

**Customer Services:** Margriet Debeij ([m.debeij@segment.nl](mailto:m.debeij@segment.nl))

**Subscriptions:** Elektor Electronics (Publishing), Regus Brentford, 1000 Great West Road, Brentford TW8 9HH, England. Tel. (+44) 208 261 4509, fax: (+44) 208 261 4447  
Internet: [www.elektor-electronics.co.uk](http://www.elektor-electronics.co.uk)  
Email: [subscriptions@elektor-electronics.co.uk](mailto:subscriptions@elektor-electronics.co.uk)  
Rates and terms are given on the Subscription Order Form

**Head Office:** Segment b.v. P.O. Box 75 NL-6190-AB Beek The Netherlands  
Telephone: (+31) 46 4389444, Fax: (+31) 46 4370161

**Distribution:** Seymour, 2 East Poultry Street, London EC1A, England  
Telephone: +44 207 429 4073

**UK Advertising:** Huson International Media, Cambridge House, Gogmore Lane, Chertsey, Surrey KT16 9AP England.  
Telephone: +44 1932 564999, Fax: +44 1932 564998  
Email: [p.brady@husonmedia.com](mailto:p.brady@husonmedia.com)  
Internet: [www.husonmedia.com](http://www.husonmedia.com)  
Advertising rates and terms available on request.

**International Advertising:** Frank van de Raadt, address as Head Office  
Email: [advertenties@elektuur.nl](mailto:advertenties@elektuur.nl)  
Advertising rates and terms available on request.

**Copyright Notice**

The circuits described in this magazine are for domestic use only. All drawings, photographs, printed circuit board layouts, programmed integrated circuits, disks, CD-ROMs, software carriers and article texts published in our books and magazines (other than third-party advertisements) are copyright Segment b.v. and may not be reproduced or transmitted in any form or by any means, including photocopying, scanning or recording, in whole or in part without prior written permission from the Publishers. Such written permission must also be obtained before any part of this publication is stored in a retrieval system of any nature. Patent protection may exist in respect of circuits, devices, components etc. described in this magazine. The Publisher does not accept responsibility for failing to identify such patent(s) or other protection. The submission of designs or articles implies permission to the Publishers to alter the text and design, and to use the contents in other Segment publications and activities. The Publishers cannot guarantee to return any material submitted to them.

**Disclaimer**

Prices and descriptions of publication-related items subject to change. Errors and omissions excluded.

© Segment b.v. 2007

Printed in the Netherlands

Advertisement

**PEAK** electronic design ltd  
 tel. 01298 70012  
 fax. 01298 70046  
 www.peakelec.co.uk  
 sales@peakelec.co.uk

Handheld Test Gear - Cool, Smart.

- Atlas DCA Semiconductor Analyser £250
- Atlas ECB Passive Component Analyser £75
- Atlas ESR Capacitance and ESR Meter £35
- Atlas SCR Triac/Thyristor Analyser £90

Prices include UK delivery and VAT

**componentbin.com** Online Store!

Our online store contains an ever expanding variety of products designed for engineers, hobbyists and students. These include:

- ARM Development Systems
- Ethernet Modules
- RF Modules
- Graphic LCD Displays (With example software!)
- Home Automation
- Power Conversion
- Components & Connectors
- And Much More!
- Educational Discounts Available

Full product support with reference software and secure online ordering to kickstart your development.

[www.componentbin.com](http://www.componentbin.com)  
 +44 (0) 1354 778224





# How many pieces of test equipment can you buy for £99?

With a PoScope USB instrument you get the features of an oscilloscope, spectrum analyser, chart recorder, logic analyser (with UART, SPI, I<sup>2</sup>C and 1-wire serial bus decoding), pattern generator and square-wave/PWM generator. That's equivalent to six pieces of test equipment for £99 including UK delivery and VAT.

PoScope is a low-cost USB-based instrument that adds invaluable test equipment features to your desktop or notebook PC. Being PC-based, all measurements can be printed, copied to the clipboard and saved as text, bitmap or vector graphics for subsequent analysis or to import into other programs. PoScope is ideal for use by electronics hobbyists, students and engineers alike and is particularly suited to those developing with microcontrollers such as PIC and AVR.

**PoScope provides the following operation modes:**

- 2-channel oscilloscope with 100Hz to 200kHz sampling, -20V to +20V input range, 10-bit ADC resolution, absolute, differential and external triggering, adjustable pre-trigger and marker measurements.
- 2-channel spectrum analyser with klirr factor measurement, Hamming, Hanning, Blackman and Blackman-Harris FFT window functions.
- 2-channel chart recorder with 0.01Hz to 200kHz sampling, maximum, minimum and average voltage measurements for each channel and waveform record over several tens of hours.
- 16-channel (8 when pattern generator used) logic analyser with 1kHz to 8MHz sampling, versatile triggering with adjustable pre-trigger, external clocking, preset pulse miss, preset bit sequence/edge, decoding of UART, SPI, I<sup>2</sup>C and 1-wire serial interfaces.
- 8-channel 1kHz to 1MHz pattern generator with tabular waveform formatting or direct timing chart plotting on the screen.
- Square-wave/PWM (pulse width modulation) generator.

Compatible with Microsoft Windows ME, 2000 and XP, PoScope is supplied with easy-to-use software and a USB cable. Oscilloscope probes and logic analyser test lead/clip sets are available separately.

Order now on **Freefone 0800 612 2135** or online at [www.paltronix.com](http://www.paltronix.com)

**PALTRONIX**  
EQUIPMENT FOR ELECTRONICS DEVELOPMENT, TRAINING & EXPERIMENTATION

**Paltronix Limited**  
Unit 3 Dolphin Lane, 35 High Street, Southampton SO14 2DF  
Telephone: 0845 226 9451 Facsimile: 0845 226 9452  
Email: [sales@paltronix.com](mailto:sales@paltronix.com) Web: [www.paltronix.com](http://www.paltronix.com)



All major credit and debit cards accepted





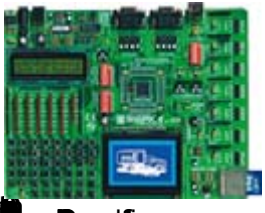
**PICFlash**  
with mikroICD support



**LvPICFlash**  
with mikroICD support



**dsPICFlash**  
with mikroICD support



**ARMFlash**

**8051prog2**







**DAC Board**



12-bit  
DAC  
PIC  
16C45

**AD Converter**



12-bit  
AD  
PIC  
16C45

**CAN Board**



CAN  
PIC  
16C45

**CAN PI Board**



CAN  
PIC  
16C45

**IDA Board**



IDA  
PIC  
16C45

**On Board Keypad**



4x4  
PIC  
16C45

**Board**



PIC  
16C45

**Board**



PIC  
16C45

**IDA Board**



IDA  
PIC  
16C45

**Keypad Board**



4x4  
PIC  
16C45

**Board**



PIC  
16C45

**EFM Board**



EFM  
PIC  
16C45

**Board**



PIC  
16C45

**Board**



PIC  
16C45

**Board**



PIC  
16C45

**Board**



PIC  
16C45

**Board**



PIC  
16C45

**Board**



PIC  
16C45

**Board**



PIC  
16C45

**Board**



PIC  
16C45

**Board**



PIC  
16C45

**Board**



PIC  
16C45

**Board**



PIC  
16C45

**Board**



PIC  
16C45

**Board**



PIC  
16C45

**Board**



PIC  
16C45

**Board**



PIC  
16C45

**Board**



PIC  
16C45

**Board**



PIC  
16C45

**Board**



PIC  
16C45

mikroBasic for PIC MCU

PIC COMPILER

mikroPascal for PIC MCU

PIC COMPILER

mikroC for PIC MCU

PIC COMPILER

mikroBasic for dsPIC30/33 and PIC24 MCU

PIC30/33/24 COMPILER

mikroPascal for dsPIC30/33 and PIC24 MCU

PIC30/33/24 COMPILER

mikroC for dsPIC30/33 and PIC24 MCU

PIC30/33/24 COMPILER

mikroBasic for AVR MCU

AVR COMPILER

mikroPascal for AVR MCU

AVR COMPILER



SOFTWARE AND HARDWARE SOLUTIONS FOR EMBEDDED WORLD

# An Obstacle Detecting Robot

C. Tavernier

[www.tavernier-c.com](http://www.tavernier-c.com)

When first trying our hand at robotics, we're generally in a hurry to build a mobile robot that has a degree of autonomy. It's with this aim in mind that we've produced this article, to enable you, in record time, to build a mobile robot capable of detecting and avoiding obstacles. Of course, given the relative simplicity of the solutions employed, it will be fairly easy to catch it out, but as long as you take a little care over the obstacles it might encounter, it'll still create quite an impression. And you'll be able to use this starting point as a springboard for your own developments. To deal with the mechanical issues that often pose problems for many amateur robot-builders, we're making use of a 'Rogue Blue' base ([www.roguerobotics.com](http://www.roguerobotics.com)), sold as a kit and very easy to assemble without special tools. What's more, this kit is distributed in France by Lextronic ([www.lextronic.fr](http://www.lextronic.fr)), who can despatch to any of the neighbouring countries.

As you can see from the photo, it consists of two pre-cut circular plates able to accommodate two radio-control servos, used as traction motors. Supplied with the base, these come already modified (as explained elsewhere in this issue) to enable them to turn continuously. Two wheels with large-diameter tyres are also provided, fixing directly onto the servo shafts, while the front and rear of the robot's lower plate have PTFE 'skids' taking the place of a jockey wheel.

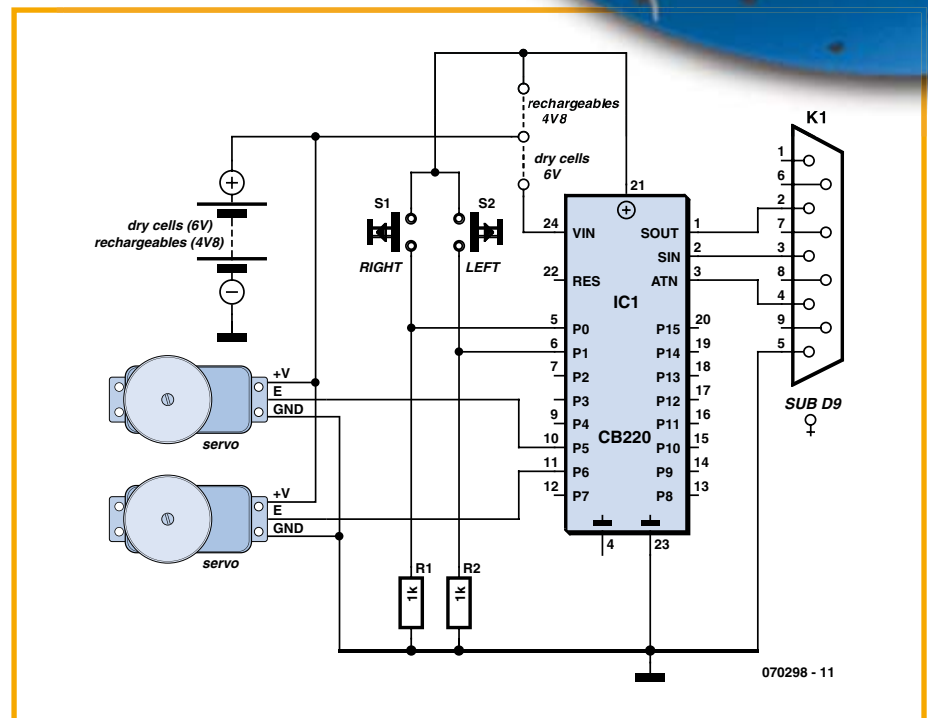
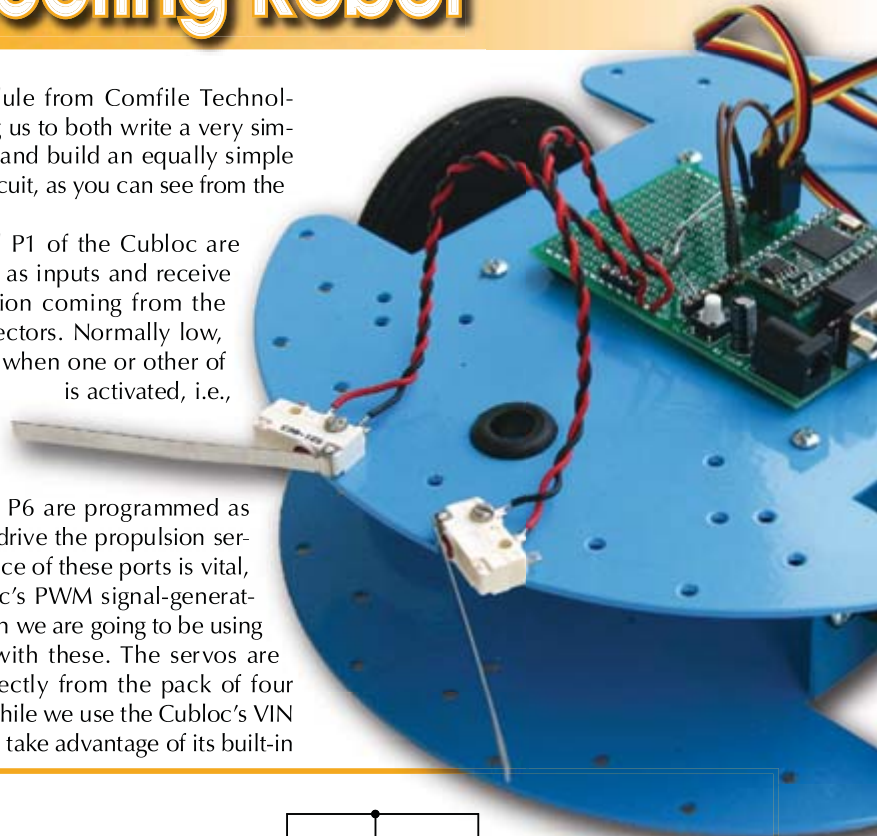
Some self-adhesive Velcro lets us secure a battery holder for primary cells or rechargeables between the two circular plates, leaving the whole of the top plate free for the electronics. Given the supply voltage for the servos and the electronics, we have used a 4-cell battery holder that takes four 1.5 V AA (R6) cells, and will also take rechargeable NiMH batteries of the same size, in the event of intensive use.

So as to get our robot up and running quickly, we've used a 'whisker'-type obstacle detector. To achieve this, we use two long-lever microswitches — or two ordinary lever-operated microswitches with a few centimetres of piano wire soldered to them — mounted on the top plate. They are positioned at an angle of around 45–60° to each other, with their centrelines intersecting on the robot's front/back centreline. This gives us one obstacle detector on the front right, and another on the front left. The robot's brain is entrusted to a Cubloc

CB 220 module from Comfile Technology, allowing us to both write a very simple program and build an equally simple electronic circuit, as you can see from the diagram.

Ports P0 and P1 of the Cubloc are programmed as inputs and receive the information coming from the obstacle detectors. Normally low, they go high when one or other of the whiskers is activated, i.e., in the presence of an obstacle.

Ports P5 and P6 are programmed as outputs and drive the propulsion servos. The choice of these ports is vital, as the Cubloc's PWM signal-generating instruction we are going to be using only works with these. The servos are powered directly from the pack of four 1.5 V cells, while we use the Cubloc's VIN input so as to take advantage of its built-in



5 V regulator. However, if you are going to use NiMH rechargeables instead of primary cells, you'll have to use the VDD input to power the Cubloc, as in this case the voltage available is only 4.8 V. Connector DB9 is designed for connecting the Cubloc to a PC, to program it with the software we'll be suggesting in a moment. Given the simplicity of the circuit, it can be built on perforated prototyping board or on a CB220-Proto test board, which comes with this connector already pre-wired. The software needed to control the whole

thing is very simple, even for someone with only faint notions about programming. The Cubloc's Basic language is both simple and very powerful. The source listing is available on the Elektor website, as well as the author's own site ([www.tavernier-c.com](http://www.tavernier-c.com)), but it's so short, you can also just type it yourself directly into the Cubloc Studio editor, which is the Cubloc's (free) development tool, and can be downloaded from [www.comfiletech.com](http://www.comfiletech.com). The listing is very easy to analyse. After defining the type of Cubloc in use and the



operating sense of the ports P0, P1, P5, and P6, we also define a variable 'obstacle'. The main program can then commence, in the form of a continuous Do Loop.

We start by testing for one or other of ports P0 or P1 going high, and if this is the case, the robot is stopped by means of the two PWM instructions that follow. Given that we are using modified radio-control servos, remember they are stopped when they receive pulses of 1.5 ms; they rotate at full speed in one direction for 2 ms pulses; and at full speed in the other direction for 1 ms pulses.

Note too that, as the servos are mounted in reversed orientations in the Rogue Blue base, they need to rotate in opposite directions to make the robot go forwards or backwards. Because of the mechanical and electrical tolerances, 1.5 ms pulses don't always give exactly stop. So you may need to go back and tweak the first parameter of the PWM instructions (3410 in this example).

Once the robot has stopped, we test to see if the left or right whisker has been activated, and set the variable 'obstacle' accordingly. One last test checks if both whiskers have been activated simultaneously, and if this is the case — meaning the robot has encountered an obstacle directly ahead — reverse drive is applied (2 ms pulses to one servo and 1 ms to the other).

The variable 'obstacle' is then analysed by means of a 'select' box that the Cubloc's powerful Basic has borrowed from C. If the variable 'obstacle' has a value of 0, no obstacle has been encountered and our robot sets off forwards. If the variable 'obstacle' has a value of 1 or 2, an obstacle has been encountered to right or left, and so a quarter-turn is made in the opposite direction. However, if the variable 'obstacle' has the value 3, an obstacle directly ahead has been encountered, and the robot does a complete U-turn.

Watch out! Depending on what you call front/back and left/right on your particular robot, you may need to swap round the successive PWM instructions in the listing we've just given, so that they do indeed produce the movements intended.

You might also need to tweak the first parameter of the various PWM instructions. We explained the reason for this above for stop, but the same thing happens for rota-



### Full source listing for obstacle-detecting robot management program

```

\ Obstacle avoider robot on a Rogue Blue base

Const Device = CB220

Dim Obstacle As Byte

Input 0      \ Right whisker input
Input 1      \ Left whisker input
Low 5        \ P5 output for PWM
Low 6        \ P6 output for PWM
Delay 1000

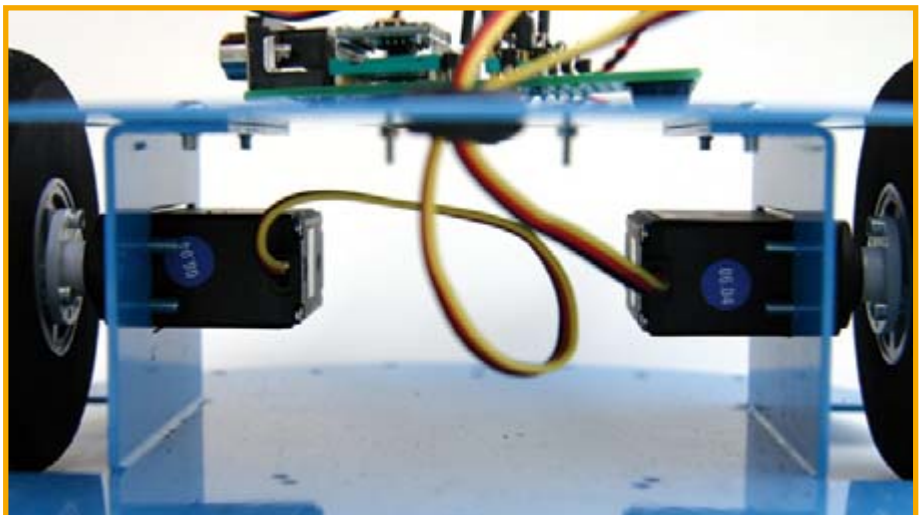
Do
If In(0)=1 Or In(1)=1 Then      \ Whisker activated?
Pwm 1,3410,32768                \ Servos stopped
Pwm 0,3410,32768
Pause 800
If In(0) = 1 Then                \ Right whisker?
Obstacle = 1
Else
Obstacle = 2                    \ Left whisker!
End If

If In(0) = 1 And In(1)= 1 Then   \ Right and left whiskers?
Obstacle =3
End if
Pwm 1,3590,32768                \ Backward
Pwm 0,3195,32768
Pause 1500
Else
Obstacle = 0                    \ No whisker activated
End If

\ Obstacle variable analysis

Select Case Obstacle
Case 0                          \ No obstacle
Pwm 0,3590,32768                \ Forward
Pwm 1,3195,32768
Case 1                          \ Obstacle on the right side
Pwm 0,3600,32768                \ Slight turn to the left
Pwm 1,3600,32768
Pause 1000
Case 2                          \ Obstacle on the left side
Pwm 1,3180,32768                \ Slight turn to the right
Pwm 0,3180,32768
Pause 1000
Case 3                          \ Head-on obstacle
Pwm 0,3750,32768                \ Full half turn
Pwm 1,3750,32768
Pause 1500
End Select
Loop

```



tion of the servos at full speed in one direction or the other. If your robot fails to travel in a straight line when running forwards or backwards, it's just because the servos are not turning at the same speed for pulses of the same width. In this case, all you have to do is make minor adjustments to the first parameter of one or the other of the PWM

instructions in order to get correct results. Don't be afraid to, the Cubloc's program memory is virtually infinitely reprogrammable (a minimum of 10,000 cycles guaranteed by the manufacturer of the microcontroller it's fitted with!) After a certain period of use, you'll doubtless realize the limitations of this robot.

Then it's up to you to develop it, by adding, for example, obstacle detectors of the same type, but to the rear, an ultrasonic distant obstacle detector, a line-follower function, etc. This issue of Elektor already ought to give you some good ideas to get you started.

(070298-1)

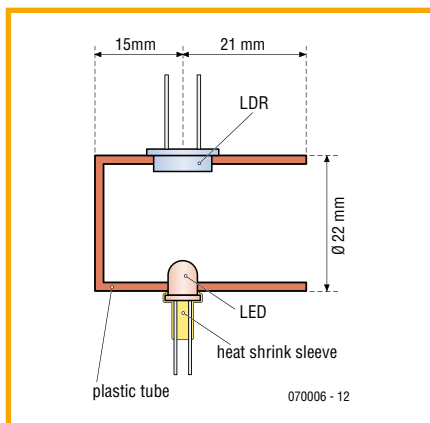
# Wireless pulse sensor

## For robots and other control applications

Markus Bindhammer

This pulse sensor is designed to be used for communication between man and machine, giving a robot the ability to react to the pulse rate of its human mentor. The digital output of the circuit makes it useful in other applications as well.

The sensor itself consists of an ordinary LDR (with a resistance when illuminated of 300  $\Omega$  and a dark resistance of around 10 M $\Omega$ ) and a bright LED (D1). The LED must have an output of at least 1000 mcd



as light from it must pass through the finger and illuminate the LDR. Now, when the heart pumps a pulse of blood through the blood vessels, the finger becomes slightly more opaque and so less light reaches the LDR. This can be converted into an electrical pulse.

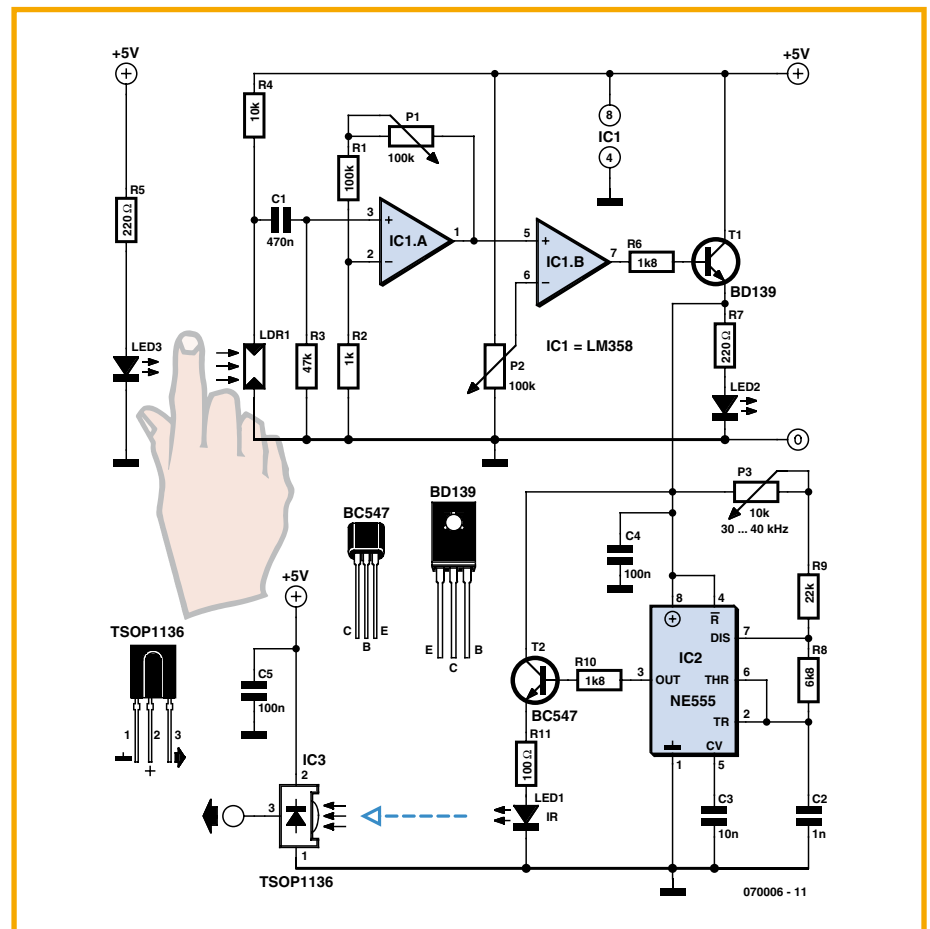
With each pulse the resistance of the LDR, and hence the voltage at the input to opamp IC1.A, changes. The gain of the opamp is set by potentiometer P1 in the feedback path. The sensitivity of the circuit can be adjusted using potentiometer P2 at the input to comparator IC1.B. T1 forms an output driver that not only lights LED D2 to give a local indication of

the detected pulse, but also powers up a standard squarewave oscillator circuit built around IC2, a 555 timer. At its output this produces a signal modulated by the pulse, with a frequency that can be set from 30 kHz to 40 kHz using potentiometer P3. A driver stage interfaces the output of the 555 to an IR emitter diode, which can send the modulated signal to IR receiver module IC3. The more power used to drive the IR LED, the greater the range of the link: R11 can be altered to achieve the desired LED current. The demodulated output of the receiver module can be fed directly to a microcontroller. The centre frequency of the receiver module used will determine

the correct setting of P3.

A pulse sensor can be made from a simple 40 mm length of plastic tube, closed at one end, chosen to fit snugly over the fingertip. Holes to mount the LED and LDR are made 15 mm from the closed end, and the components are glued suitably into the holes so that they face towards the centre of the tube. The connecting wires are isolated from one another and the whole sensor enclosed in a length of heat-shrink tubing to exclude external light from the LDR. If this construction seems a bit bulky, it is possible to reuse a clip from a commercial heart rate monitor.

(070006-1)





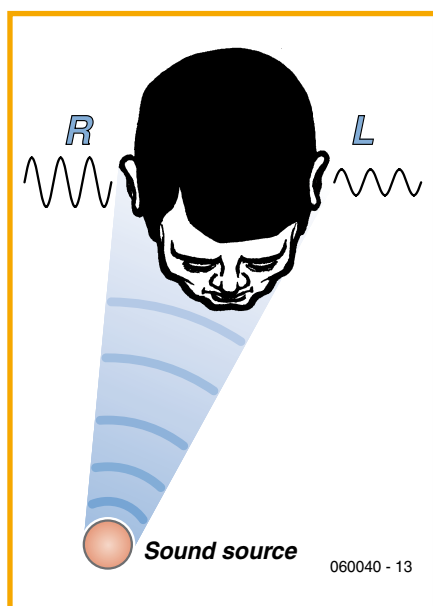
# Stereo Robot Ears

## Get your robot to home-in on a sound source

Claude Baumann & Laurent Kneip

Service robots of the future will most likely need to act on spoken commands and be able to recognise voices. This article takes a look at one aspect of this behaviour namely locating the position of a sound source using the cross-correlation function. A technique is developed which drastically cuts down on the number crunching so that even a basic microcontroller fitted with binaural sensors can pinpoint a continuous audio signal with an accuracy of just 10 degrees.

The GASTON Lego robot built in 2003 by the students at a school in Luxembourg [1] (main picture) has a number of interesting features the most obvious of which is its rudimentary 'face' which is used to express a limited range of emotions. In addition it is able to detect sounds and turn its head in the direction of the sound source. It uses an array of three microphones together with a microcontroller to make a simple 'precedence sensor' which measures the time difference produced when a sound pressure wave-front (made by a clap or finger click) strikes the microphones.



**Figure 1.** The head acts as a low-pass filter attenuating frequencies above 1 kHz. The ear nearest the sound source will hear the sound louder than the far ear.

Despite its impressive functionality, GASTON cannot work with a continuous audio signal let alone follow a moving sound source. This ability requires a more complex approach which we will go on to investigate here.

### How we track down sounds

The human ear is an amazingly complex and sensitive organ. Together with dedicated regions of the brain it enables us to extract meaningful information from the general cacophony which is continually assailing our ears. Amongst other things it has been demonstrated that we are able to identify the bearing (azimuth) of a sound source with an accuracy of just 3°. Applying a crude analogy from engineering systems we could say that the process of hearing uses a number of subsystems. Most obviously we, along with all other creatures (excluding mantids apparently) are equipped with two ears hence 'binaural' which in conjunction with dedicated regions of the brain we use to identify the direction of a sound source using several different methods:

#### a. Interaural Level Difference — ILD

Low frequencies have wavelengths which are greater than the diameter of the head; they extend around to the far ear with very little loss of amplitude. At frequencies above 1 kHz however the head acts as a low-pass filter providing up to 20 dB attenuation to the signal so there is a significant reduction in the sound level reaching the far ear (**Figure 1**) which enables the brain to estimate the position of the sound source.

#### b. Interaural Time Difference — ITD

With a sound originating from one side the pressure waves arrive at the ears with a slight time difference. The brain interprets the two signals, applying a type of neural cross-correlation function. The phase shift between the two signals gives the angle of the sound source (the azimuth  $\alpha$ ).

It can be seen from **Figure 2** that sound emanating from any of the points  $M(u,v)$  lying on the hyperbola given by the equations

$$\begin{aligned} u^2/a^2 - v^2/b^2 &= 1 \\ a &= \Delta x / 2 \\ b^2 &= k^2 - a^2 \end{aligned}$$

where  $k$  is the half distance between the ears. These points produce exactly the same time difference at the ears. The term  $\Delta x$  is the distance the sound travels in the time  $\Delta t$ , with  $\Delta x = c \times \Delta t$ . The speed of sound  $c$  equals 343 m/s at 25°C.

The hyperbola approaches the asymptote given by:

$$v = b/a \times u$$

where  $\tan(\beta) = b/a$ .

$$\beta = \arctan \left( \sqrt{\frac{4k^2}{c^2} \cdot \frac{1}{\Delta t^2} - 1} \right)$$

For R (right ear):  $\alpha = 90^\circ - \beta$ ; for L (left ear) the corresponding  $\alpha = -(90^\circ - \beta)$ .



The ear must also be able to determine if the sound emanates from in front of or behind the ear and also either from above or below. The points in three-dimensional space where a sound source will produce identical time differences in the left and right ear actually form the surface of a hyperboloid. It is thought that the brain can resolve these spatial ambiguities by detecting subtle changes in the signal spectrum caused by the outer ear shape, absorption/scattering by the torso and head, localising by turning of the head and possibly also by detecting Doppler effects.

Whatever processes the brain uses to resolve the left-right direction problem it is interesting to plug some values into the formula for a sound source positioned in front and to the right of the head. At a bearing  $\alpha = 20^\circ$  and assuming an ear separation of 17.5 cm, a time difference of 175  $\mu\text{s}$  will be apparent at the ears. At an azimuth of just  $3^\circ$  the time difference will only be 27  $\mu\text{s}$ . It is difficult to imagine how the brain (with a neuron switching time in the millisecond range) can resolve such short time differences and gives us some insight into the complex processes it is capable of. The limits of ITD for localising a continuous tone are clear; a sound wave will

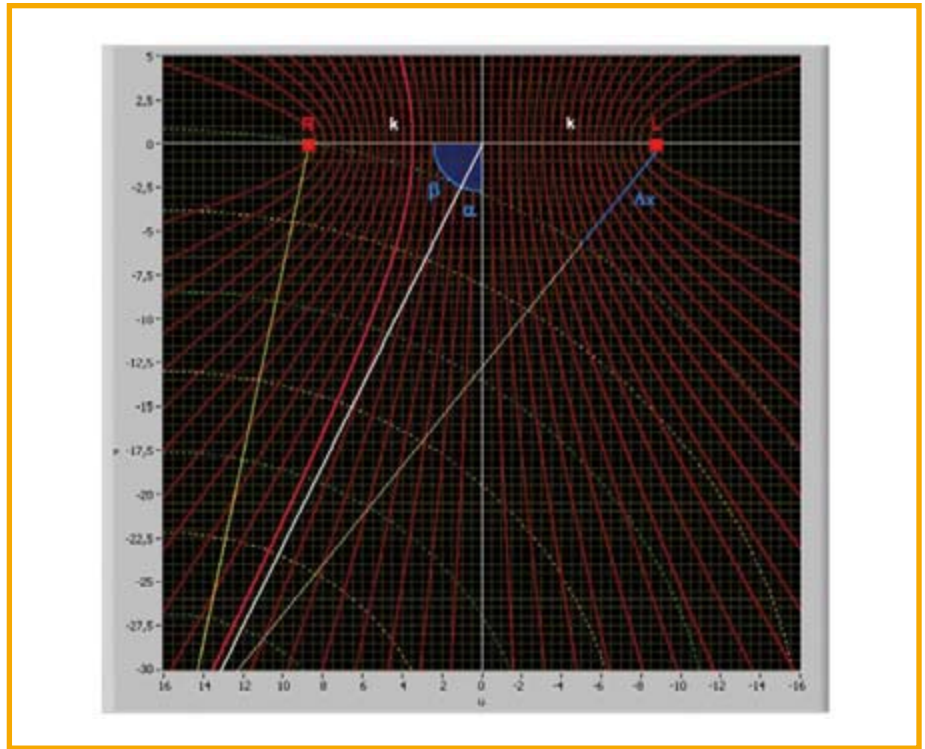


Figure 2. A sound pressure wave from the side will not arrive at both ears at the same time. Points in space where a sound source produces identical delays at the ears are on a hyperbola (red) approaching the asymptote (white). The brain cannot identify the position but only approximate the direction of a sound source anywhere along this line.

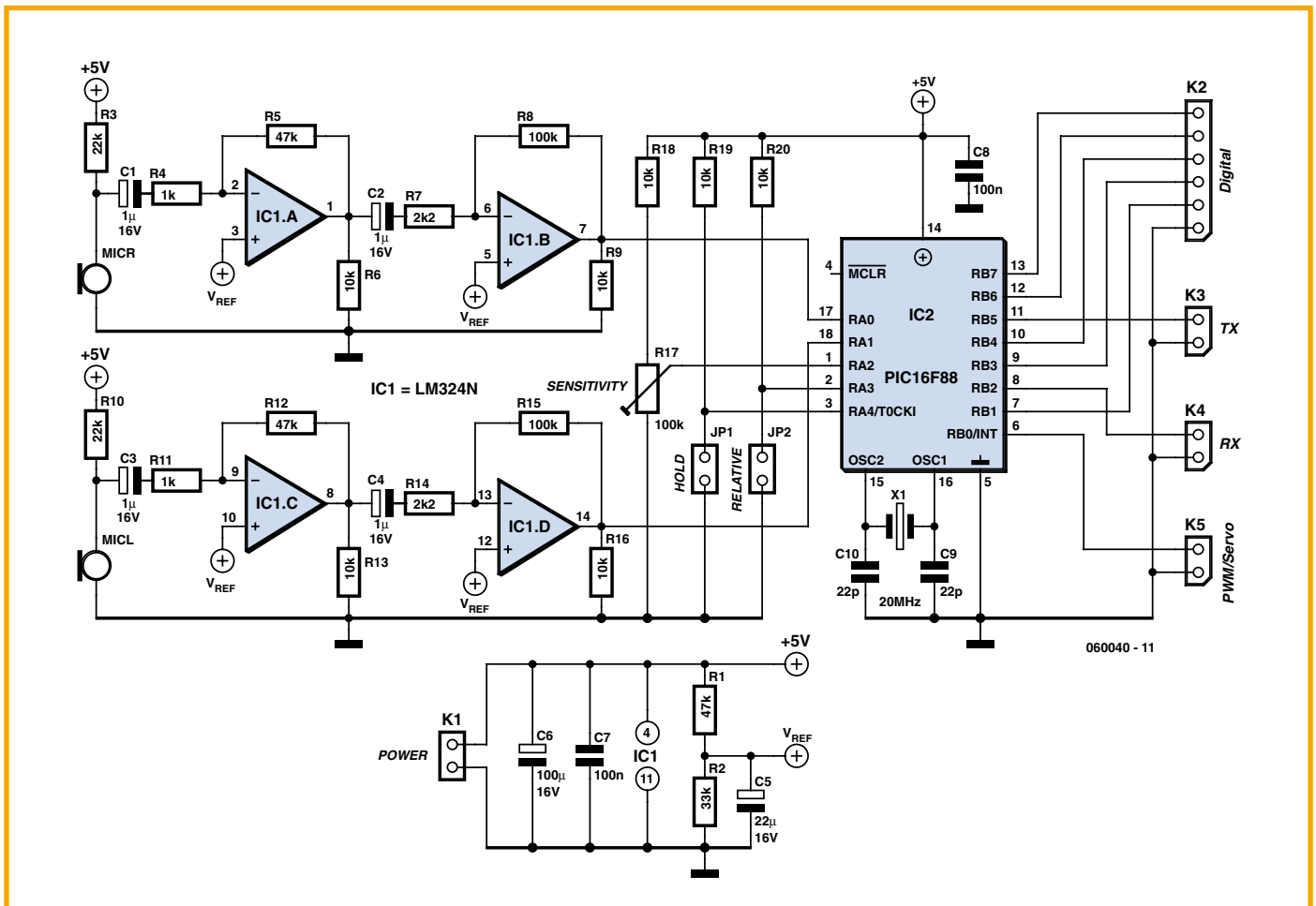


Figure 3. The circuit has two audio channels each with an electret microphone and two-stage amplifier. The two amplifier outputs are connected to the two A/D converter inputs on the microcontroller.



take around 500  $\mu\text{s}$  to travel the distance between the ears which equates to a half cycle or a  $180^\circ$  phase shift between the two signals. At frequencies of 1 kHz and above it is no longer clear if the signal in one ear leads or lags the other ear.

### c. The Precedence effect

Strong reflections and standing waves produce localised highs and lows in sound pressure when a continuous tone is reproduced in an acoustically reflective enclosed space, making it almost impossible to pin-point the sound source. When the sound source is discontinuous however it has been shown that the brain is able to identify the source, using a modified ITD measurement it gives precedence to the arrival of the first sound pressure wave at the ears and appears to open a 'neural time window' of around 1 ms for this purpose, ignoring any subsequent echoes. The phase difference of the two signals gives the azimuth bearing of the sound source. The LEGO robot GASTON uses this technique but it is not of much use for continuous sounds.

## Cross-Correlation

When there are two identical signals shifted in time, it is common practice to apply the cross-correlation operation to them. Together with Fourier analysis these two operations form the backbone of techniques used in digital signal processing. The digital cross-correlation is derived from two continuous waveforms as follows: The value at  $\Delta t = 0$  is found by sampling the waveforms at intervals  $t$ , multiplying the samples from the two waveforms together and then summing them and normalising the result. The cross-correlation at another position, say  $\Delta t = 1 \mu\text{s}$  is then performed in the same way except this time the second signal is shifted in respect to the first by  $1 \mu\text{s}$ . The maximum sampling rate is chosen to ensure that the processor can perform the necessary calculations along with any other work it needs to do within the sampling time constraints.

It can be determined by the resultant cross-correlation whether the two signals are 'correlated' for example if the first signal is not in phase with the second signal but otherwise very similar the cross-correlation function will show a marked peak corresponding to the phase shift between the two signals. Cross-correlation calculations involve many multiplications and additions; it is hardly surprising that in the DSP world it has a reputation for being particularly processor hungry.

## Reducing computer loading

In order to perform cross-correlation with a general purpose microcontroller it is

necessary to find ways of reducing the number of computations that the processor is required to make. For this application it is valid to limit the number of phase shifted operations taking into account the distance between the two microphones. It can be shown also that maximising the product sum (cross-correlation function) is equivalent to minimising another function which is much simpler to calculate. For each phase shift the sum of the differences of both values (squared) can be transformed so that cross-correlation function  $\gamma(\tau)$  in the equation

$$\begin{aligned} f(\tau) &= \sum [x(t) - y(t + \tau)]^2 \\ &= \sum [x^2(t) + y^2(t + \tau) - 2x(t)y(t + \tau)] \\ &= \underbrace{\sum x^2(t) + \sum y^2(t + \tau)}_{c_1} - 2 \sum x(t)y(t + \tau) \\ &= c_1 - 2 \cdot N \cdot \gamma(\tau) \end{aligned}$$

with  $\gamma(\tau) = (2N)^{-1} [c_1 - f(\tau)]$ .

The value of  $c_1$  is constant for every phase shift while the square of the values are added in each case irrespective of which phase position it is. ( $N$  is the constant geometric mean of all the signal values produced by normalising  $\gamma(\tau)$ ). It is evident that when the cross-correlation value is at maximum it corresponds to a minimum of the deviation sum of  $f$ . The following expression is therefore valid for our application and makes fewer demands on precious processor resources.

$$g(\tau) = \sum |x(t) - y(t + \tau)|$$

This expression is a close approximation to cross-correlation. The two measured samples are subtracted (ignore the sign) to give the absolute difference. Summing them produces a non-normalised value which is at a minimum when the waves are correlated. The technique requires far fewer calculations than the classic cross-correlation method and is at least 20 times faster.

## Construction of the binaural sensors

The circuit in (Figure 3) could hardly be simpler. It consists of two audio channels each with own electret microphone and two-stage audio amplifier. Each audio signal is then fed to two pins of a microcontroller which have been configured as A/D converter inputs. A preset pot, R17, allows sensitivity adjustment.

Bearing data is output every tenth of a second using several different interfaces, which gives the sensor the flexibility to be used in many different types of robot. The

8-bit bearing information is sent from TX serially using a UART (2400 1 N 8), it is also output as a 5-bit parallel digital value. A servo output (PWM/Servo) provides a signal with a width from 1 to 2 ms at a 50 Hz repetition rate suitable for driving a standard modelling servo. Jumper 1 (Hold) is provided for test purposes, it introduces a two seconds pause between readings during which time the last valid output is maintained. Jumper 2 (Relative) defines how the output behaves when the received sounds are too quiet to make a measurement; with it fitted the output returns to the middle position (azimuth  $\alpha = 0$ ), with it removed the output retains its last valid position. This gives the system a degree of flexibility, for example if the microphones are mounted on a robot which can turn and move toward the sound it is better to fit the jumper otherwise the robot continues to turn when no sound is detected. When the microphones are fixed and the output is used to pan a webcam say, it is better to remove the jumper otherwise the camera continually pans back to the centre position when no sound is detected.

The RX input can be used later for microcontroller firmware updates. A 100-k $\Omega$  pulldown resistor is used to avoid a floating input.

## Timing and resolution

For this application we will consider the frequency range from 200 to 1000 Hz that the processor calculates the ITDs (signal delays) it requires a sufficiently high sampling rate which could only be achieved by careful optimising of the program code. The PIC16F88 architecture is organised so that data from the two channels can be stored in two 96-byte banks (banks 2 and 3) as quickly as possible by switching a single bit and using indirect addressing.

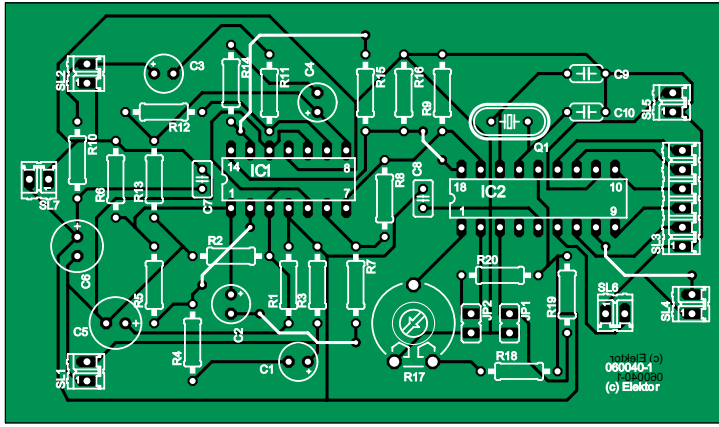
The PIC16F88 contains a 10-bit A/D converter but for this application the two least significant bits are ignored so that we use an 8-bit value. The sampling rate is 20 kHz. The values are smoothed by an FIR-filter. Any erroneous values detected by the program are over-written with the previous correct value.

At a frequency of 1 kHz a  $180^\circ$  phase shift is measured after 0.5 ms i.e. the time taken for ten samples. The smallest detectable time difference corresponds to a minimum distance of:

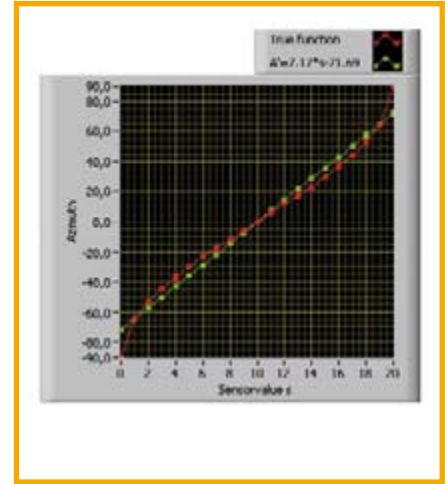
$$d_{\min} = 50 [\mu\text{s}] \cdot 343 [\text{m/s}] = 1.7 \text{ cm}$$

This gives the optimum spacing between the microphones ( $2k$ ) of:

$$2k = 10 [\text{Samples}] \times d_{\min} = 17 \text{ cm}$$



**Figure 4.** Component layout of the binaural sensor PCB.  
The PCB files can be downloaded from (3).



**Figure 5.** Resolution of the sound source bearing is a function of its azimuth.  
The average value is approximately 9°.

## Components list

### Resistors

- R1,R5,R12 = 47kΩ
- R2 = 33kΩ
- R3,R10 = 22kΩ
- R4,R11 = 1kΩ
- R6,R9,R13,R16,R18,R19,R20 = 10kΩ
- R7, R14 = 2kΩ
- R8, R15 = 100kΩ
- R17 = 100kΩ preset
- Rx = 100kΩ

### Capacitors

- C1,C2,C3,C4 = 1μF 16 V
- C5 = 22μF 16 V
- C6 = 100μF 16 V
- C7,C8 = 100nF
- C9,C10 = 22pF

### Semiconductors

- IC1 = LM324
- IC2 = PIC16F88, programmed, order code **060040-41** from Elektor SHOP

### Miscellaneous

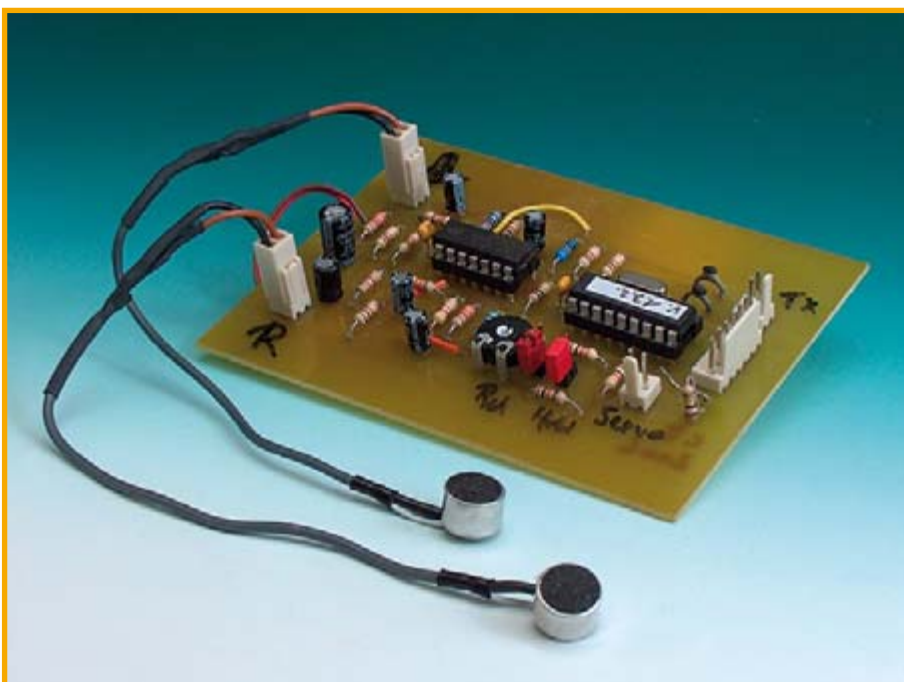
- Q1 = 20MHz quartz crystal
- DIL14 socket
- DIL18 socket
- SL1,SL2,(SL4),SL5,SL6,SL7 = 2-way SIL pinheader (SL4 bridged with a 100k resistor, see text)
- SL3 = 6-way SIL pinheader
- JP1,JP2 = jumper
- MicR, MicL = CZ034 electret microphone insert
- PCB, ref. **060040-1**, free artwork download from Elektor website

When the sound source is at the extreme right or left of the field (azimuth  $\alpha = \pm 90^\circ$ ), a time difference of  $\pm 10$  samples will be measured. A minimum of 20 calculations are made for the correlation calculation. When the microphones are mounted on either side of a (solid) head this has the effect of increasing the microphone spacing, the pressure wave from one side needs to travel around the curve of the head before it reaches the second microphone. Assuming a sphere with microphones mounted along an axis, the arc length is  $r \times \pi$ , where  $r$  is the radius of the head and should be chosen so that the arc length is not greater than 17 cm.

The accuracy of the bearing calculation is a function of the azimuth of the sound source. It can be seen from **Figure 5** that when the source is central in front of the microphones an accuracy of approximately 5° can be expected while sound from the sides achieve around 11° and only 25° at the edge which all together gives an average figure of 9°. The highly directional nature of the electret microphones response characteristics meant that further study the longitudinal response was not worthwhile.

The PIC16F88 microcontroller used in this project was programmed using the Ultimate\_PIC tools which are based on the Labview programming environment. The beta version of Ultimate\_PIC is available from The Center for Engineering Education Outreach (CEEO) at Tufts University Massachusetts.

The assembler and hex files for this project Binaural\_v132.asm and Binaural\_v132.hex can be freely downloaded from the *Elektor Electronics* website [3] where a pre-programmed PIC controller can be ordered as



**Figure 6.** The binaural sensor prototype.



an alternative from the SHOP section. The PCB layout and component placement is shown in **Figure 4**; the PCB artwork files are also available for download from [3].

### Construction and Test

A picture of the prototype is shown in Figure 6. Shielded cable must be used to connect the two microphones to the controller board to help reduce pick-up of electrical noise.

The current consumption of just 10 mA can be supplied by almost any 5 V stabilised power unit. The microphones are mounted 17 cm apart, pointing forwards. A 500 Hz tone is an ideal sound source for test purposes but voice/music from a radio is also suitable. Turning the sensitivity preset to the left will increase the sensor sensitivity.

A simple DC output level can also be achieved by connecting a 2R/R resistor network to the 5-bit Digital output (**Figure 7**). The resultant DC output level is buffered by IC1A.

(060040-1)

### Web links

- (1) [www.convict.lu/Jeunes/RoboticsIntro.htm](http://www.convict.lu/Jeunes/RoboticsIntro.htm)
- (2) [www.ultimaterobolab.com](http://www.ultimaterobolab.com)
- (3) [www.elektor-electronics.co.uk](http://www.elektor-electronics.co.uk)

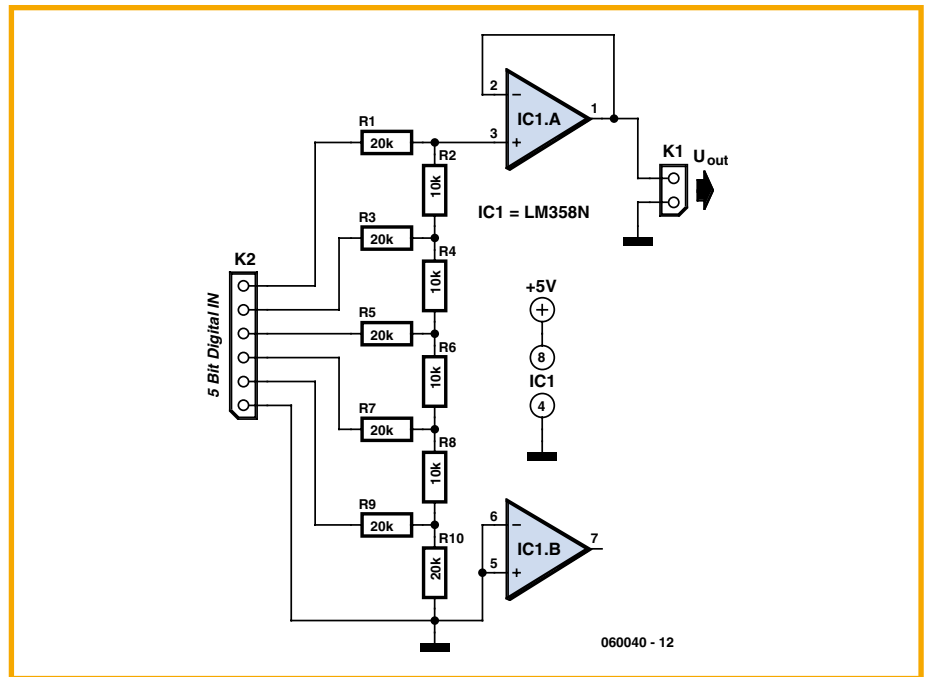


Figure 7. The resistor network functions as a simple D/A converter.

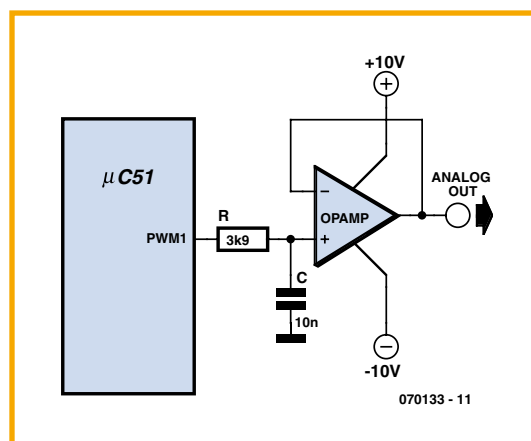
Sensor output values:			
Sound direction	(s+2) 8-bit-Integer (TX)	PWM/Servo	(s+6) 5-bit-Integer (Digital)
Extreme right	22	2 ms	26
Central	12	1.5 ms	16
Extreme left	2	1 ms	6

# Simple D/A Converter for Robots

Tilo Gockel

Sometimes it's necessary to add a D/A converter to a microcontroller for a specific application. This can be fairly simply accomplished by interfacing an off-the-shelf D/A converter to the microcontroller's bus. An even simpler and more cost-effective solution based on an application note from Microchip [1] is shown here. A microcontroller produces a PWM (pulse width modulated) digital output signal which is filtered by a low-pass RC filter. Although the PWM signal has a fixed repetition rate the on-to-off ratio is varied from 0 to 100 % which, after filtering gives an analogue output signal proportional to the ratio. A single digital output from a port pin (driven from an on-chip timer for example) can therefore form the basis of an analogue output signal.

Guidelines to calculate values for the RC low-pass filter are given in the application note. Using the values suggested (3.9 kΩ



and 10 nF) gives a -3 dB corner frequency of around 4 kHz. When driven by a PWM frequency of approximately 20 kHz the filter will be suitable for outputting audio tones and voice signals with a bandwidth of 4 kHz. This simple filter will attenuate the 20 kHz fundamental PWM frequency by 14 dB which may not sound like much but the human ear has its own high fre-

quency roll-off (the characteristics of which change as we age) so frequencies this high will be barely audible.

Any standard operational amplifier, for example the TL071 can be used in this application. Lower frequency signals all the way down to DC can also be handled by this circuit and in this case the low-pass filter corner frequency can be reduced further which will give better attenuation of the PWM fundamental and reduce ripple on DC output signals. One typical application of the circuit is speed control of a DC motor in accordance with the industry standard ±10V. The circuit will connect to the motor via a suitable power driver stage. In this case the electromechanical properties of the motor itself will act as a low pass filter.

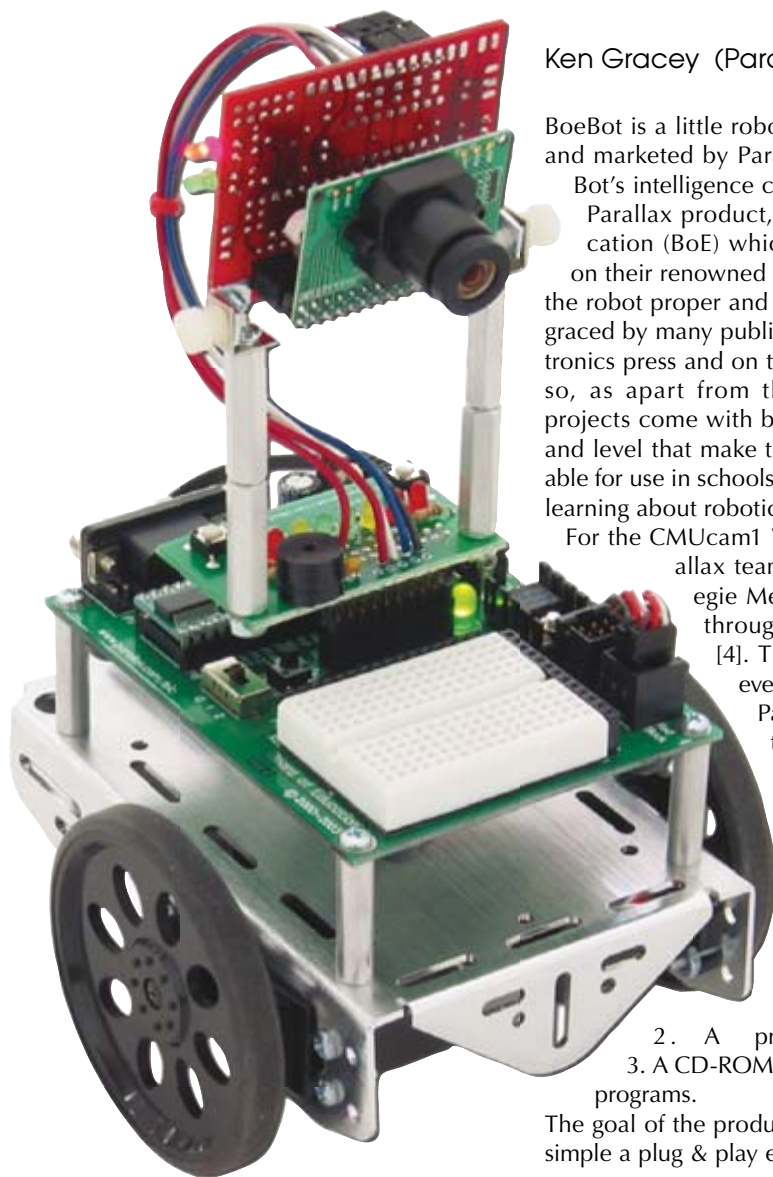
(070133-1)

### Web link

- (1) <http://www1.microchip.com/downloads/en/AppNotes/00538c.pdf>

# CMUCam1 Vision System

## Seattle Robotics give BoeBot (and other robots) vision!



Ken Gracey (Parallax, Inc.)

BoeBot is a little robot vehicle designed and marketed by Parallax Inc. [1]. BoeBot's intelligence comes from another Parallax product, the Board of Education (BoE) which in turn is based on their renowned BASIC Stamp. Both the robot proper and the BoE have been graced by many publications in the electronics press and on the Internet. Rightly so, as apart from their low cost the projects come with backup of a volume and level that make them perfectly suitable for use in schools and, in general, for learning about robotics [2].

For the CMUCam1 Vision System, Parallax teamed up with Carnegie Mellon University [3] through Seattle Robotics [4]. The product is however only available from Parallax and their distributors like Milford Instruments [5].

The CMUCam1 BoeBot package includes

1. A CMUCam1 mounted on an AppMod user interface board.
2. A printed user manual.
3. A CD-ROM that includes demo programs.

The goal of the product is to give you as simple a plug & play experience as possi-

ble. Plug in the hardware, upload the demo code then press the buttons and watch your BoeBot use its new vision system to "see" and respond to its environment.

### Purpose

The CMUCam1 AppMod™ vision system consists of a CMUCam1 vision system mounted on a plug in AppMod board. Included on the AppMod board is a simple user interface consisting of two buttons, eight LEDs and one piezo speaker. The user interface serves these three purposes:

- 1) A simple menu selection system which works with the included demo code to allow you to select and run one of eight robot vision demonstration programs.
- 2) Provide visual feedback from the LEDs as the demo programs run showing where the CMUCam1 is seeing the tracked target.
- 3) Indicate the colour of objects it sees by illuminating the corresponding LED (e.g., red LED for a red object).

With some clever circuitry on the AppMod board the two buttons, eight LEDs and one piezo speaker only require four I/O pins on your Basic Stamp2 thereby conserving I/O pins for other uses.

### An 8-function demo

With the CMUCam1 AppMod Vision System plugged into the BoeBot AppMod connector the next step is to program the main demo code from the CD-ROM to the Basic Stamp2. The demo code has eight functions, each of which demonstrating a

Table 1. Basic command set

Command	Parameter(s)	Description <sup>1</sup>
\r	none	The 'Enter' or 'Return' key. Set the camera board into an idle state.
GM	none (\r)	Get the Mean colour value in the current image.
L1	value \r	Control the green LED tracking Light.
MM	mode \r	Controls the Middle Mass mode which adds the centroid coordinates to the normal tracking data.
NF	active \r	Controls the Noise Filter setting. It accepts a Boolean value 1 (default) or 0. A value of 1 engages the mode while a value of 0 deactivates it.
PM	mode \r	Puts the board into Poll Mode. Setting the mode parameter to 1 engages poll mode while 0 (default) turns it off.
RS	none (\r)	ReSets the vision board. Note, on reset the first character is a /r.
SW	(x y x2 y2) \r	Sets the Window size of the camera. It accepts the x and y Cartesian coordinates of the upper left corner followed by the lower right of the window you wish to set.
TC	(R <sub>min</sub> R <sub>max</sub> G <sub>min</sub> G <sub>max</sub> B <sub>min</sub> B <sub>max</sub> ) \r	Track a Colour. Accepts the minimum and maximum RGB (CrYCb) values and outputs a type M or C data packet (set by the MM command).
TW	none (\r)	Track the colour found in the central region of the current Window.

<sup>1</sup> Complete descriptions may be found in the user manual.



Table 2. Advanced commands		
Command	Parameter(s)	Description <sup>1</sup>
CR	( reg1 value1 (reg2 value2 ... reg16 value16) )\r	Sets the Camera's internal Register values directly. The register locations and possible settings can be found in the Omnivision documentation.
DF	none (\r)	Dump a Frame out the serial port to a computer.
DM	value \r	sets the Delay before packets that are transmitted over the serial port.
GV	none (\r)	Gets the current Version of the firmware from the camera.
HM	active \r	puts the camera into Half-horizontal resolution Mode for the DF command and the LM command when dumping a bitmap image.
II	none (\r)	Uses the servo port as a digital Input.
LM	active \r	Turns on Line Mode which uses the time between each frame to transmit more detailed data about the image.
RM	bit_flags \r	Engage the Raw serial transfer Mode. It reads the bit values of the first 3 (lsb) bits to configure settings.
S1	position \r	Lets you Set the position of servo 1. 0 turns the servo off and holds the line low. 1-127 will set the servo to that position while it is tracking or getting mean data.
SM	value \r	Used to enable the Switching Mode of colour tracking.

<sup>1</sup> Complete descriptions may be found in the user manual.

capability of the CMUcam1. When you first turn on the BoeBot the eight LEDs on the AppMod interface board will flash several times, then the piezo speaker will beep indicating it is ready. Each function starts with one audible beep and stops with two beeps. The LEDs flash during each function in different patterns to let you know how that function is operating. The eight functions of the demo comprise:

- 1. Calibrate Lighting** – about 20 seconds are needed to allow CMUcam1 to calibrate to the ambient lighting conditions.
- 2. Sample and Save Colour** – it takes about a second to lock onto the colour of an object held in front of the cam; the LEDs will flash and the speaker will beep twice when done. The colour values are saved in the Basic Stamp2 EEPROM.
- 3. Track Colour** – the robot moves forward, backward, right and left to follow the colour of the object saved during function 2.
- 4. Move & Avoid** – the robot will move forward and avoid objects using the vision sensor only. It works by sampling the colour range of the floor in front of the robot.
- 5. Adaptive Tracking** – the robot locks onto the first colour it sees and tracks that colour (it only goes fwd, right and left, not bwd). If it loses the object for about five seconds it then locks onto the next colour it sees and tracks that until it loses that, etc.
- 6. Line Following** – it is assumed a black line about 1.2 cm (.5 in.) wide is present on a white line tracking course.
- 7. Finger Point and Move** – the robot will backup, turn right and turn left in relation to finger pointing. This is done with the CMUcam1 using the same downward facing angle as is used on all the other functions.

**8. Show Colour** – the robot will light up either all of the red, green or yellow LEDs in response to the colour of the object placed in front of it. This works well with 5-cm diameter coloured rubber ball or plastic block.

For best performance with all of the above functions, the tilt of the camera should be pointing down looking just in front of the robot.

## Basic Stamp2sx and Basic Stamp2p

This CMUcam1 AppMod and demo code will also work with the more powerful 2sx and 2p versions of the Basic Stamp. Separate versions of the demo code for each Stamp can be found on the CDROM. The code changes are minor and related to the 2.5 times faster execution speed. You can do more with the CMUcam1 and the 2sx and 2p Stamps due to their faster speed and larger memory. The Basic Stamp2 is limited to 9600 baud serial interface speed to talk to the CMUcam1 but the 2sx and 2p can both talk to the CMUcam1 at its maximum serial data rate of 115,200 baud. The CMUcam1 is switched to the 115,200 baud rate by removing two jumpers that can be found on its board. The higher baud rate means BoeBot can respond much faster to the vision system.

## About the CMUCam and the module

The CMUcam1 is an SX28 microcontroller [6] interfaced with an OV6620 Omnivision CMOS camera [7] on a chip that allows simple high level data to be extracted from the camera's streaming video. The board communicates using a TTL level serial port

and has the following functionality:

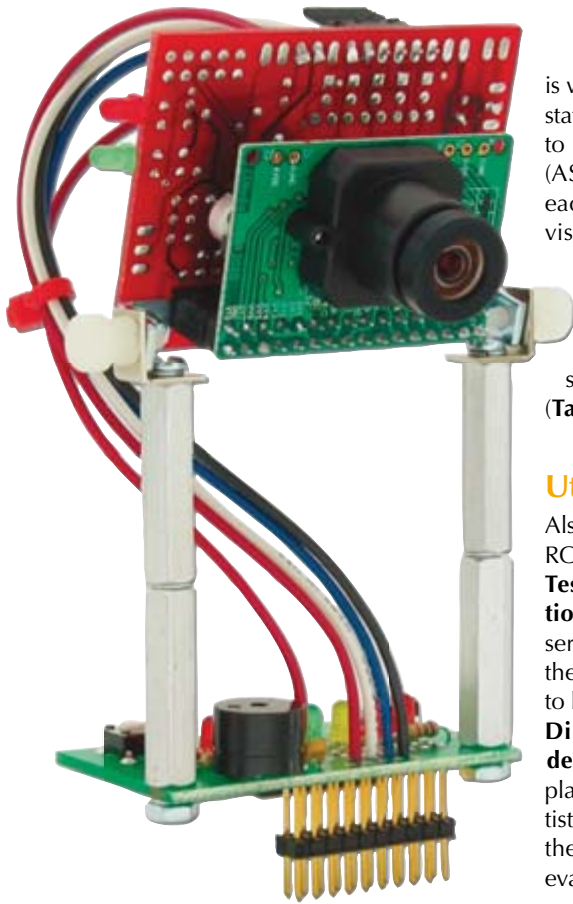
- Track user defined colour blobs at 17 frames/s
- Find the centroid of the blob
- Gather mean colour and variance data
- Arbitrary image windowing
- 80x143 resolution
- 9600 baud serial communication
- Automatically detect a colour and drive a servo to track an object
- Slave parallel image processing mode off a single camera bus (advanced function)
- Ability to control one servo or have one digital I/O pin (advanced function)
- Adjust the camera's image properties (advanced function)

When using the camera outside, due to the sun's powerful IR (infrared) emissions, even on relatively cloudy days, it will probably be necessary to use either an IR cutoff filter or a neutral density-3 camera filter to decrease the ambient light level. A lens taken from a cheap drugstore pair of sunglasses when placed over the camera lens will allow the CMUcam1 to work in sun lit conditions.

## Serial comms and command sets

The serial communication parameters are as follows: 9600 baud, 8 data bits, 1 stop bit; no parity, no flow control (no Xon/Xoff or hardware).

All commands are sent using visible ASCII characters, i.e., 123 is three bytes "123". Upon a successful transmission of a command, the ACK string should be returned. If there was a problem in the syntax of the transmission, or if a detectable transfer error occurred, an NCK string is returned. After either an ACK or an NCK, an \r is returned. When a prompt ('\r' followed by a ':') is returned, it means that the camera



is waiting for another command in the idle state. White spaces do matter and are used to separate argument parameters. The `\r` (ASCII 13, carriage return) is used to end each line and activate each command. If visible character transmission exerts too much overhead, it is possible to use varying degrees of raw data transfer ('Raw mode').

The system supports two command sets — basic (**Table 1**) and advanced (**Table 2**).

### Utility programs

Also included on the Seattle Robotics CD-ROM are the following utility programs.

**Test CMUcam1 to BoeBot communication.** This program sets up a 9600-baud serial connection between the Stamp and the CMUcam1. It then tells the CMUcam1 to blink its green LED.

**Display CMUcam1 tracking data on debug screen.** The first data packet displayed by the Debug screen is the 'S' (Statistics) packet which tells you the colour of the object it is tracking. This will let you evaluate the ability of your camera to track

an object. Try different colour objects and different sizes to see the effects on the tracking data. This is an important program that you will re-use many times as you find more and more interesting things to do with your CMUcam1 and BoeBot. This program allows you to see and understand exactly what your BoeBot can see with its CMUcam1 vision system.

A short video showing a CMUcam1-equipped BoeBot locating a red object may be found at [8].

(070132-1)

### Web references

- (1) [www.parallax.com](http://www.parallax.com)
- (2) [www.stampsinclass.com](http://www.stampsinclass.com)
- (3) [www.seattlerobotics.com](http://www.seattlerobotics.com)
- (4) [www.cs.cmu.edu/~cmucam](http://www.cs.cmu.edu/~cmucam)
- (5) [www.milinst.com](http://www.milinst.com)
- (6) [www.ubicom.com/processors/sx/sx\\_family.html](http://www.ubicom.com/processors/sx/sx_family.html)
- (7) [www.ovt.com](http://www.ovt.com)
- (8) [www.seattlerobotics.com/video.htm](http://www.seattlerobotics.com/video.htm)

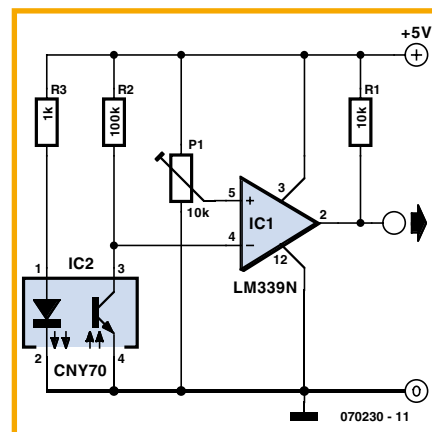
# Sensor for Line Following Robots

David Gustafik

One of the traditional 'disciplines' in which robots are supposed to compete is the Line Follower Competition. Robots go along a pre-drawn black line (usually duct tape) on a white-ish surface (usually paper, cardboard or plastic).

This discipline requires special sensors. Usually, these are made out of reflective optosensors (such as CNY70, LTH-209). This sensor contains a phototransistor and an infrared LED. These sensors are pointed at the surface on which the robot is supposed to show off its speed and agility. The LED emits infrared light on the surface and the phototransistor acts as a receiver. The black coloured line to follow reflects far less light than the white surface it is affixed on. The current that flows through a phototransistor depends on the intensity of the light detected. Therefore, more current will flow through the transistor when it is above a white surface. In this way, the sensor can also be used as a surface detector.

The minimal count of sensors necessary for making a line following robot is... two — one on the left side and one on the right. It is advisable to use at least three sensors — one left, one right and one in the middle



as a failsafe precaution against the robot dropping off a table edge.

In this circuit, the voltage on the phototransistor is compared with a reference level set by P1. When IC2 is illuminated, the voltage on it drops. Comparator IC1A compares the voltage against the set reference. If the reference voltage is higher than that on phototransistor, the comparator's output is drops to (almost) zero. This occurs when there is a black line under the sensor. The output signal from the comparator is then connected to either a microprocessor or any control logic that (hopefully) responds by adjusting

the course of the robot .

The circuits needs to be calibrated before use. The best method is to set the P1 preset to the centre of its travel. Next, place the sensor above the surface it is supposed to detect, where it is white. Note that the height of the sensor above the surface is important. It doesn't matter that much when using for instance a CNY70, but an LTH209 for example only works in a very small range of heights (around 3.8 mm). If the response from the output of the comparator is good (i.e., pin 2 High) move the sensor above a line. If the result is again as expected (pin 2 Low) you're done calibrating the circuit. If not, repeat the process and adjust P1 until the correct calibration is achieved.

The schematic for only one of four channels that can be made with just one LM339 IC. The pull-up resistor at comparator pin 2 is used because the LM339 has open-collector outputs. R3 determines the current going to the IRLED.

Many comparators can be used, the LM339N just happened to be available. The same may apply largely to the optosensor, but note that many different pinouts exist so check out that datasheet.

(070230-1)



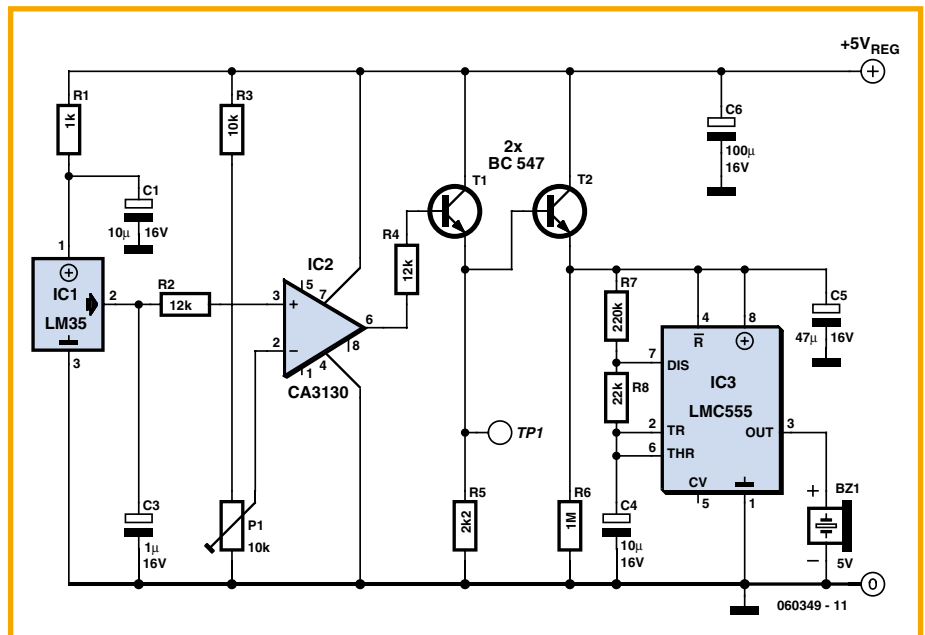
# Overheat Detector Alarm/Switch

T. K. Hareendran

At the heart of this circuit is a precision integrated temperature sensor type LM35 (IC1), which provides an accurately linear and directly proportional output in mV, over the zero to +155 degrees C temperature range. The LM35 develops an output voltage of 10 mV/K change in measured temperature. Designed to draw a minimal current of its own, the LM35 has very low self heating in still air.

Here the output of the LM35 is applied to the non-inverting input of a comparator wired around a CA3130 opamp (IC2). A voltage divider network R3-P1 sets the threshold voltage, at the inverting input of the opamp. The threshold voltage determines the adjustable temperature trip level at which the circuit is activated.

When the measured temperature exceeds the user-defined level, the comparator pulls its output High to approx. 2.2 V causing transistor T1 to be forward biased instantly. T2 is also switched on, supplying the oscillator circuit around IC3 with sufficient voltage to start working. The 555 set up in astable mode directly drives active



piezoelectric buzzer Bz1 to raise a loud alert. Components R7, R8 and C4 determine the on/off rhythm of the sounder. A transistor based relay driver may be driven off the emitter of T1 (TP1). Similarly,

replacing the piezo sounder with a suitable relay allows switching of high-power flashers, sirens or horns working on the AC mains supply.

(060349-1)

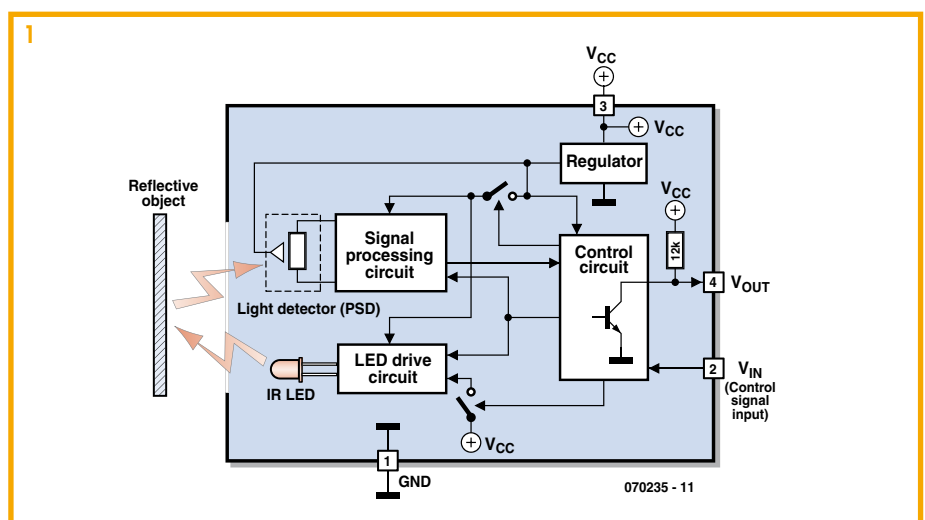
# PIC or Basic Stamp IR Telemeter

with a PIC or a Basic Stamp

C. Tavernier

Although the simplest robots may be content to simply detecting obstacles, many robots that require precision in their positioning need to be able to measure distances accurately. To achieve this, it is necessary to use a telemeter, which can be infrared or ultrasonic. IR is very suitable for measuring short distances (a few centimetres up to a few tens of centimetres), while ultrasound is more suitable for distances from a few tens of centimetres to several metres.

Although it is still possible to construct a telemeter using standard resources, these days it's not really worthwhile because of the availability of ready-to-use integrated



modules that are all relatively accurate, cheap, and compact. Taking a look at IR telemeters, the Sharp range is currently the largest and most readily available, insofar

as we wish to stick with products at prices compatible with an 'amateur' robot. This range, whose part numbers all begin with GP2..., includes telemeters that pro-

### Listing 1. Use of GP2Dxx with a Basic Stamp.

```
Vin          con    0          ` Definition of control input
Vout         con    1          ` Definition of data output
Measr        var    Byte       ` Allocation of one byte for the result

Measr = 0          ` Initialization of variable « Measr »
Read:
Vin = 0           ` Validation of telemeter
Wait:
IF Vout = 0 THEN Wait ` Wait until result is available
SHIFTIN Vout, Vin, 2, [Measr]
Vin = 1           ` Put telemeter to sleep
Pause = 1        ` Pause as a precaution

` The result of the measurement is available in the variable 'Measr'
```

vide on/off outputs (though these aren't really telemeters as such!), information in analogue form, and information in digital form. Though the versions providing analogue information would appear to be the easiest to use, it's absurd to use them in a robot driven by a microcontroller, which will immediately convert this analogue voltage into a digital signal via its built-in converter so as to be able to make use of it. So it is better to have digital information available right from the outset, even if it might seem a little harder to read at the telemeter output.

In these circumstances, two types are currently readily available: the GP2D02, capable of measuring from 10 cm to 80 cm approximately, and the GP2D021, capable of measuring from 4 to 30 cm approximately. These two types are fully compatible both mechanically and electrically, and so everything we are about to write is

equally applicable to both of them.

The principle of an IR telemeter is relatively simple: an LED emits an IR beam that, if it encounters an object, is reflected back towards a photodiode. Left at that, such a system is an on/off type and is really more an obstacle detector than a true telemeter. Although certain of the Sharp telemeters do work along these lines, the two types we've chosen are capable of performing true distance measurement, as the rays reflected by the object are no longer picked up by just a simple photodiode, but by a CCD array.

As a result, the angle of incidence of the reflected beam reaching this array varies according to how far away the detected object is, and hence allows true distance measurement, provided there is a minimal amount of signal processing to exploit the information generated by the CCD sensor.

This is the case in the Sharp telemeters of this type, the internal block diagram of which is shown in **Figure 1**. In those telemeters with a measurement validation input (as in the models chosen), the LED is only powered under its control, allowing a very significant reduction in quiescent power consumption. In the other telemeters, it's on all the time.

The CCD sensor is followed by a signal processing circuit that allows an output to be generated as either an on/off type signal for the simplest telemeters, an analogue signal, or lastly, 8-bit digital information, as in the models that interest us.

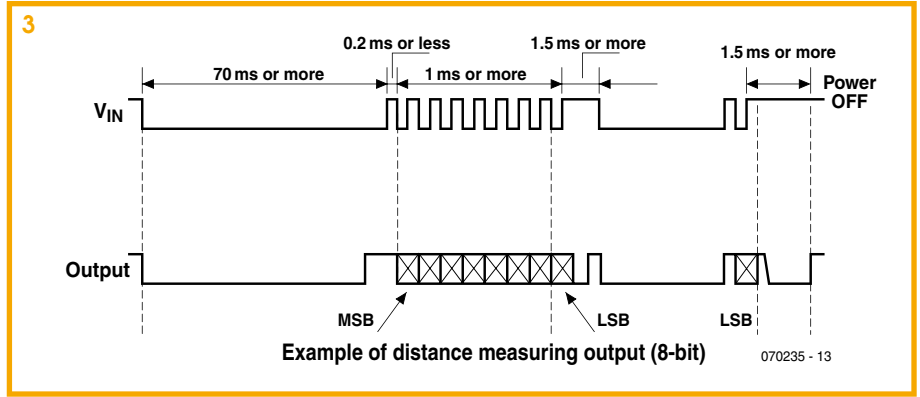
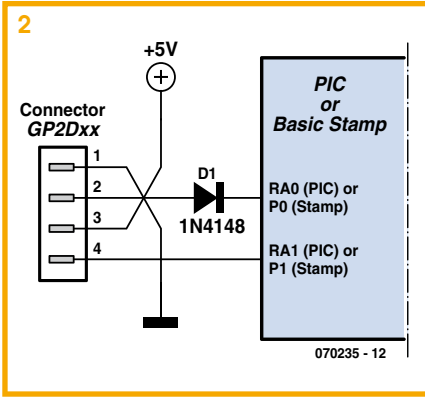
So as to be compatible with a maximum number of robotics solutions, we've opted to show you how to use such a telemeter with either a Basic Stamp or a PIC microcontroller programmed in Basic or machine code.

### Listing 2. Use of GP2Dxx with a PIC controller.

```
Read
BCF    Porta.0          ` Validation of telemeter
NOP
Wait
BTFSS  Porta.1          ` Wait until result is available
GOTO   Wait
BSF    Porta.0          ` Vin goes High
CLRF   Measr           ` Initialization of the variable 'Measr'
MOVLW  8                ` Get ready to read 8 bits
MOVWF  Count
BCF    Status,C         ` Zero the carry
NOP
Readbit
BCF    Porta.0          ` Make clock Low
NOP
NOP
RLF    Measr,f          ` Rotation of preceding bit
BTFSC  Porta.1          ` Read data bit
BSF    Measr,0
BSF    Porta.0          ` Make clock High
NOP
NOP
DECFSZ Count,f         ` Count down number of bits to read
GOTO   Readbit

`The result of the measure is available in the variable 'Measr'
```



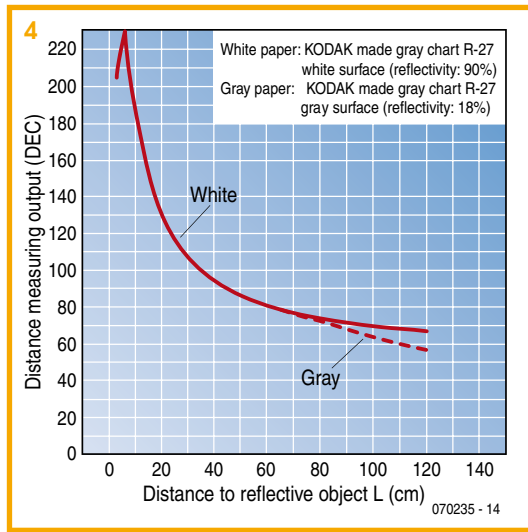


The telemeter application circuit can be summed up as shown in **Figure 2**, whether it's with a PIC or a Basic Stamp. The GP2D02 or GP2D021 detector is powered all the time, but because it has a control input, it consumes virtually nothing when there is no measuring taking place. A glance at the timing diagram in **Figure 3** shows us that this input is used not just for validating the measurement, but also as a clock for transferring the reading to the  $V_{out}$  terminal. Hence it needs to be controlled by the associated microcontroller, but as it must not be subjected to a voltage above 3 V, diode D1 isolates it from the microcontroller output when the latter is high.

The timing diagram in **Figure 3** should enable you to easily follow the listings of the very short programs we've written for using this sensor, whether in

Basic, for the Basic Stamp and PICs programmed in Basic, or in PIC assembler, for those of you who prefer machine code.

As far as the Basic Stamp is concerned, just the instruction SHIFTRIN is all it takes to read the result of the sensor's measurement. For the PIC, there will obviously need to be a few more instructions to generate the read clock and recover the relevant data. In both cases, these programs provide the digital data returned by the telemeter following the measurement in the variable 'Mearst'.



Then it's up to your robot's management program to use this value directly, or to linearise it using a conversion table, if you want to perform actual distance measurement.

In fact — and this is perhaps the sole shortcoming of these telemeters — the information they provide is far from linear, as shown in **Figure 4**.

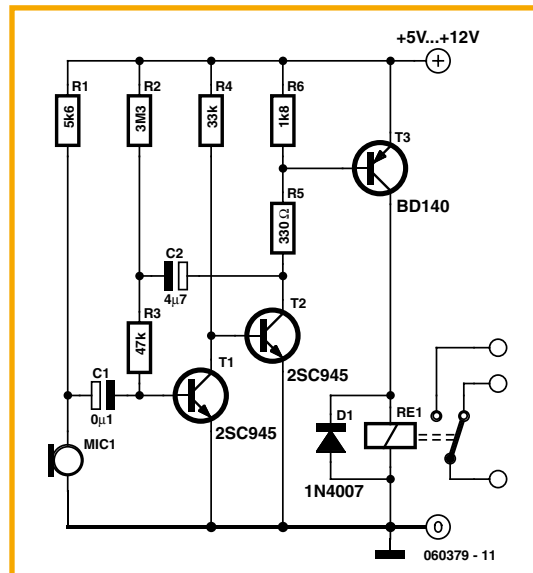
## Sound Activated Switch

Hesam Hoshiri

Control by sound may be very useful, not just on a robot but also for a bit of home automation, for example a sound-activated light responding to a knock on the door or a hand clap. The light will be automatically switched off after a few seconds. An alternative use is burglar protection — if someone wants to open the door or break something the light will come on, suggesting that someone's at home.

The circuit can work from any 5–12 VDC regulated power supply provided a relay with the suitable coil voltage is used.

When you first connect the supply voltage to the circuit, the relay will be energised because of the effect of



capacitor C2. Allow a few seconds for the relay to be switched off. You can increase or decrease the 'on' period by changing the value of C2. A higher value results in a longer 'on' period, and vice versa. Do not use a value greater than 47  $\mu$ F.

Biasing resistor R1 determines to a large extent the microphone sensitivity. An electret microphone usually has one internal FET inside which requires a bias voltage to operate. The optimum bias level for response to sound has to be found by trial and error.

**All relevant electrical safety precautions should be observed when connecting mains powered loads to the relay contacts.**

# A Robot that won't Lose its Bearings

C. Tavernier

Creating a robot capable of following an exact course without needing any external physical help — like a line marked on the ground, for example — has for a long time been a very tricky operation to achieve. Thanks to the boom in amateur robotics on the one hand, and to the development of new sensors on the other, it is today possible to make robots with functions that only a few years ago would have seemed more like science fiction. And this is just what we're proposing now, with the construction of a robot capable of detecting terrestrial magnetic North, and hence to follow any angular direction with respect to that, exactly as you would find your bearings using a compass. To do this, it uses a successor to the good old needle compass, in the form of a module made by Devantech called a CMP03 (or CMPS03, depending on versions and documents).

This module, supplied in the form of a small, ready-to-use PCB as shown in the photo, contains two Philips KMZ51 magnetic field sensors mounted at right angles, with their output signals processed by a suitably-programmed PIC16F872 microcontroller. It is capable of delivering angular position information with respect to terrestrial magnetic North with an accuracy of up to a 1/10°. This information is available in the form of a PWM signal where the width of the pulses represents this angle, though in this case the resolution is only 1°. It is also available via an I<sup>2</sup>C bus that, depending on which register is read in the module, can make this angle available in the form of a 16-bit word, offering a precision of 0.1°. If a robot is fitted with such a module, it's then possible at any moment to find out the angle its trajectory is making with magnetic North, and hence to steer it exactly as you would yourself using a compass. The only 'problem' that may arise is the interfacing of the CMP03 module with the microcontroller fitted in the robot. So in order to cater for the greatest number of robotic configurations possible, we're going to show you how to employ both means of dialogue offered by the module:



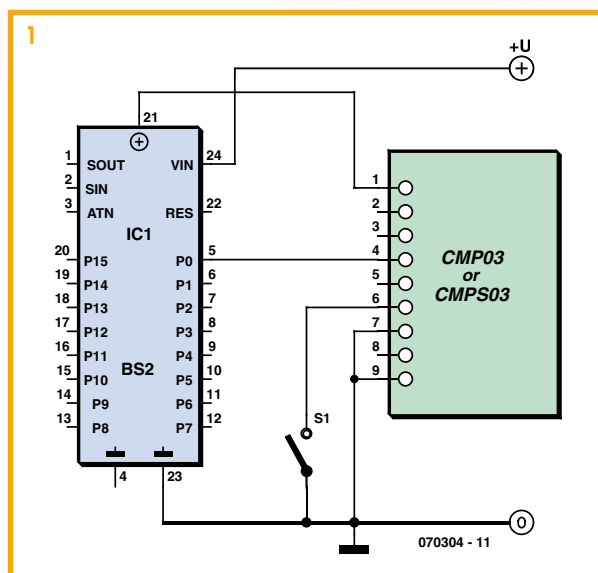
- Position is the angle in degrees with respect to magnetic North.
- Width is the width of the high pulses in milliseconds.

Reading such information with a Basic Stamp II or a PIC programmed in Basic amounts to just two lines of program:

```
PULSIN 0, 1, PULSE  
POSITION = (PULSE -  
500) / 50
```

The first line enables the instruction PULSIN to read the high duration of the pulse generated by the CMP03 module. The second line merely applies the above relationship, given that the resolution of the measurement performed by PULSIN is 2 μs in the case of the Basic Stamp II. So we then have the angle in degrees with respect to magnetic North in the variable POSITION.

If you want greater precision, or if your microcontroller does have an I<sup>2</sup>C interface available, it's possible to use this interface to dialogue with the CMP03 module, as is shown by way of example in Figure 2. It has been designed for a Cubloc CB220 or a PIC programmed in Basic with a compiler that has an I<sup>2</sup>C library, which is the case for most of them these days.



the PWM signals and the I<sup>2</sup>C bus. Figure 1 shows a circuit for using the PWM signals. It has been designed for use with a Basic Stamp II, but can be transposed for any PIC microcontroller programmed in Basic that does not have I<sup>2</sup>C functions. Interrupt S1, present on pin 6 of the CMP03 module, is not involved in the dialogue process, but makes it possible to calibrate the module as per a procedure given in the documentation, which we won't reproduce here, as it is perfectly straightforward. The information supplied by the CMP03 module is a succession of high pulses separated by low states of 65 ms duration. The width of the high pulses indicates the angle of the principal axis of the module with respect to North, according to the following relationship:

$$\text{Position} = (\text{Width} - 1) \times 10$$

where:

The only precautions to be taken with this circuit concern the I<sup>2</sup>C bus pull-up resistors, which do need to be fitted as they are not built in to either the CMP03 module or the microcontroller (whichever type it is). If you are using a PIC programmed in Basic, you also need to ensure you correctly choose the ports intended to handle the SDA and SCL signals of the I<sup>2</sup>C bus, as certain Basic compilers impose restrictions here.

To be in a position to write the corresponding program, all you then need to know is that the I<sup>2</sup>C address of the CMP03 module is C0 and that four main registers are accessible to us through this address:

- register 0 contains the module's software version number;
- register 1 contains the angle coded in one byte. Hence this value changes from 0–255 corresponding to a circle from 0–360°;
- registers 2 and 3 contain the angle,

coded this time in two bytes, in the form of a number between 0 and 3599 (expressed in decimal), which is in fact the angle in degrees multiplied by 10.

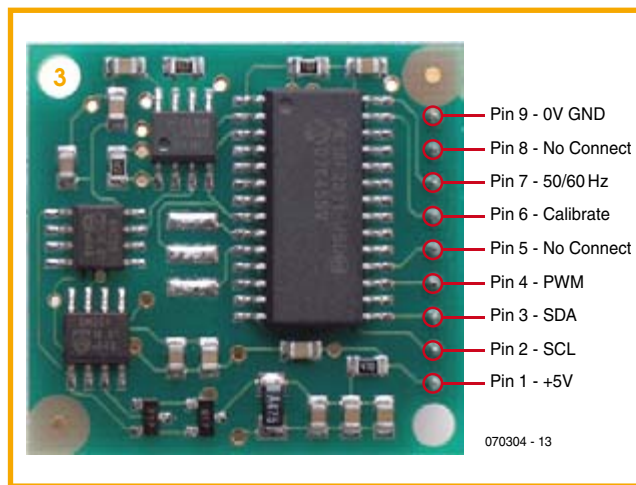
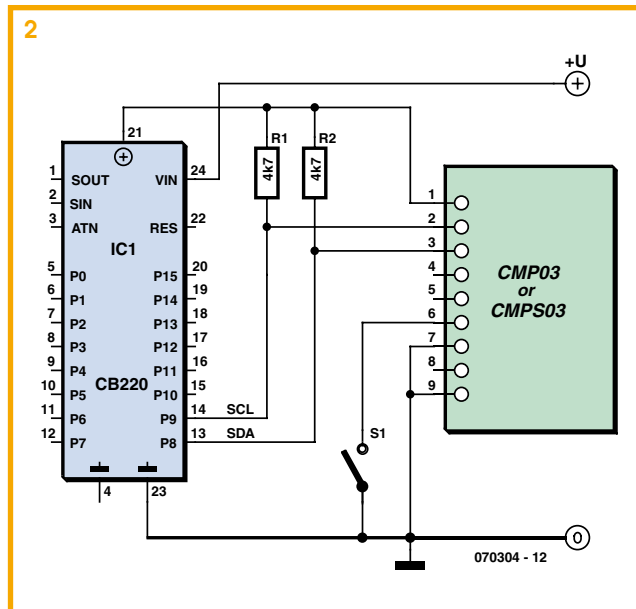
From that point on, reading this information via an I<sup>2</sup>C bus is quite straightforward, as shown in the very short listing below, written for the Basic language of the Cubloc:

```

I2CSTART
Temp = I2CWRITE (&HC0)
Temp = I2CWRITE(0)
I2CSTART
Temp = I2CWRITE(&HC1)
Version = I2CREAD(0)
Temp = I2CWRITE(&HC1)
Position8 = I2CREAD(0)
Temp = I2CWRITE(&HC1)
Position16.byte1 = I2CREAD(0)
Temp = I2CWRITE(&HC1)
Position16.byte0 = I2CREAD(0)

```

The first three instructions address the module and select the first register to be read. The next instructions read the four previously-described registers in succession, thus making the software version number available in the variable `Version`, the 8-bit coded angle in the variable `Position8`, and the 16-bit angle in the 16-bit variable



Position16. The variable `Temp` is not used for anything, but is required by the particular syntax of the I<sup>2</sup>C instructions of the Cubloc's Basic. By the same token, the constant 0 that must be present in the `I2CREAD` instructions has no particular meaning. Hence if you use this listing with a PIC programmed in Basic, a slight adaptation might prove necessary, depending on the compiler you are using. So, whether you choose the PWM or I<sup>2</sup>C version for interfacing with the CMP03 module, it gives you position information about your robot's trajectory with respect to magnetic North. All that remains for you to do is to make good use of it so your robot won't 'lose its bearings'.

(070304-1)

### Web Links

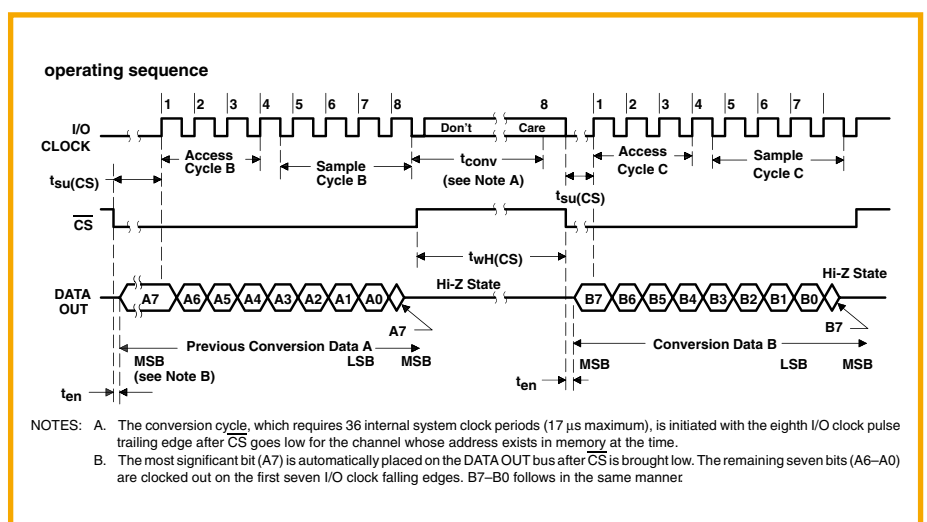
- Devantech  
[http://www.robot-electronics.co.uk/shop/Compass\\_CMPS032004.htm](http://www.robot-electronics.co.uk/shop/Compass_CMPS032004.htm)
- A little background reading:  
<http://zedomax.com/blog/zedomax-diy-hack-lets-make-a-digital-compass/>
- <http://zedomax.com/blog/2006/08/16/digital-compass-using-cmps03/>

# A/D Converter for Robots

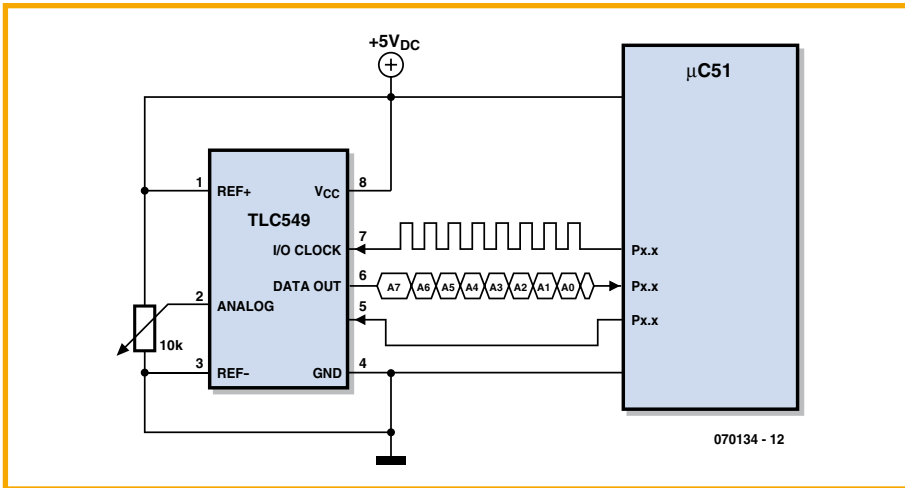
Tilo Gockel

The TLC549CP analogue to digital converter (A/D) from Texas Instruments is a good choice for applications in the field of robotics (especially those using 8051-compatible microcontrollers). These particular converters are readily available, low-cost and easy to use.

A quick look at the TLC549 data sheet indicates the timing waveforms for the I/O Clock, DATA OUT and CS signals (Figure 1). A circuit to test the converter was quickly constructed using a variable resistor as a potential divider. Figure 2 shows the simple interface between a TLC549CP and a 8051-compatible microcontroller. A short function which reads the serial







data from the TLC549 and returns the value addat is given below. Tests with an AT89S8252 Controller-Board (the Elektor Electronics December 2005 Flash Micro Board) indicated that the `_wait()` can be omitted because the controller runs slowly

enough not to need it (faster processors may require it but it has been 'commented out' in this listing). The actual function written in C is:

```
unsigned char ReadADC1 () {
```

```

unsigned char count;
unsigned char addat = 0;
P1_B2 = 0; // clk
P3_B0 = 0; // Chip Select
P3_B0 = 1; //
//_wait(); // > 20 usek (50kHz)
P3_B0 = 0;
for(count = 0; count
    < 8; count++){
    addat = addat << 1;
    if (P1_B3 == 1) ++addat;
    P1_B2 = 1;
    //_wait();
    P1_B2 = 0;
}
return addat;
}

```

As an example two TLC549 A/D converters can be wired to a microcontroller to measure the amount of light falling onto two Light Dependant Resistors (LDR). This application will require two of the interface connections shown in Figure 2. Connect the LDR either in place of the variable resistor or in parallel to it.

(070134-1)

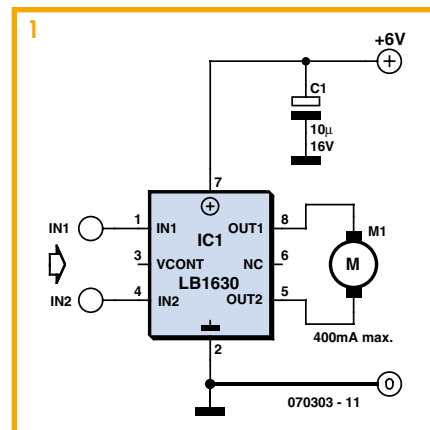
# From Cassette Recorder to Robot Propulsion

B. Broussas

There are currently three principal methods for the propulsion of a mobile robot: the modified radio-control servo, the stepper motor, and the DC motor. All have advantages and disadvantages, which are important to be familiar with before making your choice.

The modified radio-control servo offers numerous advantages, the main one being that it offers relatively high tractive power without needing a reducing gearbox, as this is already contained within the servo case. So all you have to do is mount it onto the robot and fix the wheels directly onto its shaft. What's more, a radio-control servo is powered from 4.8 V, which is particularly handy for robots using four 1.2 V rechargeable batteries.

The major drawback of the radio-control servo lies not so much in its pulse drive mode, for which various solutions are proposed elsewhere in this issue, as in the lack of precision in the behaviour of the servo with respect to the pulse width. Although theoretically the servo runs at maximum speed in one direction or the other for pulses of 1 or 2 ms and stops for pulses of



1.5 ms, experience shows that differences of 10–20% in the pulse width needed are sometimes encountered. These differences make it necessary to calibrate the propulsion control programs of robots fitted with servos on an individual basis, thereby ruling out any reproducibility from one model of robot to another.

Stepper motors do not suffer from these drawbacks, but don't usually include any built-in mechanical reduction, meaning you have to provide external reduction if you don't want to end up with too little

torque. If you don't fancy building such a device yourself, it's possible to use the 'gearboxes' sold as kits by various robotics or modelling retailers, but experience shows these are usually designed for standard DC motors, and are unsuitable for comparatively bulkier stepper motors. Moreover, driving steppers obliges us to use either a specialized IC, or a collection of logic ICs in association with power transistors.

So if you are forced to fall back on external reduction, for example because you don't want to use a servo, the DC motor then becomes the natural choice, especially since certain 'gearbox' kits are sold with such motors. So, all that remains is to drive them correctly.

Although conventional transistor-based circuits are still usable, there is also one very simple solution, directly inspired by the (old) cassette recorders in which DC motors were widely used. It involves using an LB1630 IC made by Sanyo, which can be cannibalized from many cassette recorders that have been thrown out, or else bought new, from Lextronic for example ([www.lextronic.fr](http://www.lextronic.fr)).

Available in an 8-pin DIP package, the

LB1630 is ridiculously simply to use, as shown in **Figure 1**. In fact, all it needs to be able to work is one external decoupling capacitor. It's controlled by two logic signals, which are TTL-compatible when the circuit is powered from a voltage of the order of 5 V. The supply must be between 2.5 and 6 V, and must not under any circumstances exceed 7 V, at risk of destroying the IC.

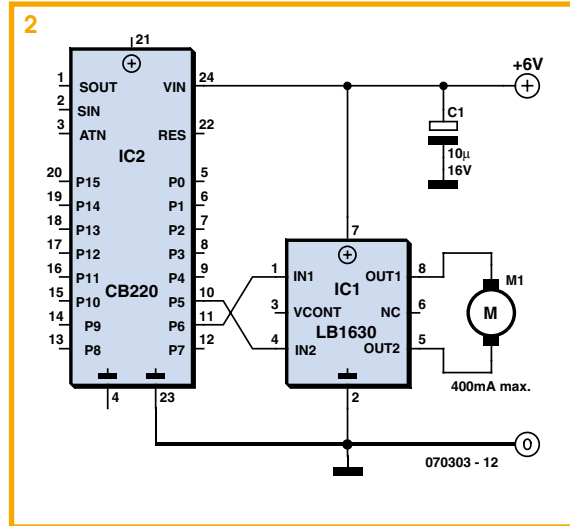
The current drawn by the motor being driven can be up to 400 mA maximum, though peaks up to 2 A are allowed, but only in the form of pulses whose duration may not exceed 50 ms at a duty cycle of 10%.

The protection diodes, vital when driving an electric motor using transistors, are built into the LB1630 and so don't need to be added to the circuit shown.

The two inputs IN1 and IN2 allow logic control of the motor, as per the **Table**.

Inputs IN1 and IN2 of the LB1630 just need to be in opposite states for the motor to turn one way or the other. So it's very easy to control using two parallel port lines from any microcontroller. Independently of this 'on/off' type control to make the motor run in one direction or the other, the speed can also be controlled. All that is needed is to apply PWM

Truth table for LB1630 motor control IC.				
IN1	IN2	OUT1	OUT2	Motor
H	L	H	L	Forward run
L	H	L	H	Reverse run
H	H	HiZ	HiZ	Stop
L	L	HiZ	HiZ	Stop



pulses to one or the other of inputs IN1 or IN2.

**Figure 2** shows one way of performing this sort of speed control using a Cubloc CB220, which has the advantage over the many other microcontrollers that can be

programmed in Basic of being able to generate continuous PWM signals. With such a circuit, all you have to do is write, for example:

```
OUT 6,0
PWM 0, SPEED, 255
```

to make the motor run in one direction at a speed that can be adjusted by means of the variable SPEED, which can vary from 0–255, and:

```
OUT 6,1
PWM 0, (255-SPEED), 255
```

to make it run at the same speed but in the other direction.

Note too that, given that IN1 and IN2 of the LB1630 are logic inputs, several of them can be connected in parallel, so as to control several motors in an identical manner. But watch out! If you're controlling two traction motors positioned back-to-back on either side of a robot, they'll need to turn in opposite directions to make the robot move forwards or backwards. In this case, you need to cross over the inputs to the LB1630s (IN1 of one goes to IN2 of the other and vice-versa) if you are controlling them together, or else wire the motors in opposite senses to the OUT1 and OUT2 outputs.

(070303-1)

# IR Close Object Detector

B. Broussas

Although ultrasound is well suited to detecting distant objects, it is quite unusable for closer objects, i.e. when the detection distance comes down to around a centimetre or less. Under these conditions, two solutions can be used: the bumper or similar device activating one or more microswitches — but that is still a mechanical solution — or the infra-red detector we're suggesting building here, combining the elegance of electronics with freedom from any moving parts.

The principle of such a detector is very simple. A transmitting element, which here is going to be a simple LED, emits a more or less directional beam of IR. Positioned next to this transmitter, a receiving element, which may be a photodiode or phototransistor, is suitably oriented so that

under normal circumstances it doesn't receive anything. But as soon as an obstacle is present at a suitable distance, it reflects part of the light emitted by the LED back onto the photodiode or transistor; the presence of an output signal from the latter then indicates the proximity of this obstacle.

The distance up to which this process works properly very clearly depends on numerous factors: the luminous power emitted by the LED, the sensitivity of the detector, but also — and above all — on the reflective properties of the obstacle. A black cat will be much less easily detected than a white wall!

In answer to a question frequently asked in robotics classes, note that this principle works just as well using visible light, but the use of IR simply makes it possible to avoid, to some extent, the sensor is being

dazzled by ambient light. Of course, if you use a robot fitted with such a sensor in full sunlight or beneath a halogen spot, this anti-dazzle effect probably won't be very effective, given the high level of IR radiated by such sources!

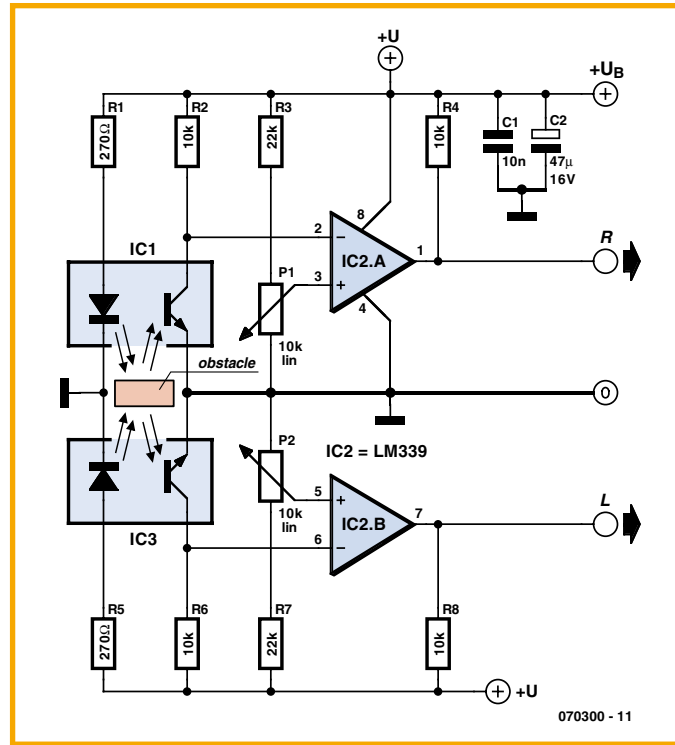
Note too that this system is not a telemeter, and so is unable to give the slightest information about the distance of the obstacle. The only parameter that actually relates to this distance is the amplitude of the reflected signal, but this depends also, indeed to a very large extent, on the reflective properties of the particular obstacle (think again of that black cat and white wall).

So, our detector is capable of operating over a range extending from a few mm to around 20 mm or so, depending on the type of sensor used. What's more, it's not confined to simple obstacle detection

in the conventional sense. For example, in the case of a robot that's meant to stay on a table, all you have to do is judiciously position such detectors around the underside of the edge of the robot's chassis. As soon as it gets too close to the edge, the sensor stops receiving the signal reflected by the table, indicating that it needs to turn back.

Construction of our IR obstacle detector is very simple, as the figure shows. Given the IC used, it's possible to build two at once, which is not unhelpful. As the detection zone of such a system is relatively limited, we've planned to use two sensors that we've called R and L for right and left, though this doesn't bear any particular relation to their actual positions on the robot. The figure shows the circuit of a single channel, the other is obviously identical; only decoupling capacitors C1 and C2 are common to them both.

The LED in the sensor IC1 is permanently powered via resistor R1, while the collector of the phototransistor in this detector is taken to the positive rail via R2. So when the transistor is off, i.e. when it is not receiving any light, meaning there is no obstacle, we have a voltage at this point approaching the power rail. As the transistor starts conducting, that is, when a suf-



ficiently reflective and/or close obstacle reflects the light emitted by the diode back onto the phototransistor, this level drops. This information is shaped by comparator IC2A, whose switching threshold can be adjusted using P1. In this way, the circuit can be adapted to different sensors and the detection range can be adjusted to some extent. The circuit output is TTL-compatible if it is powered from 5 V and, given

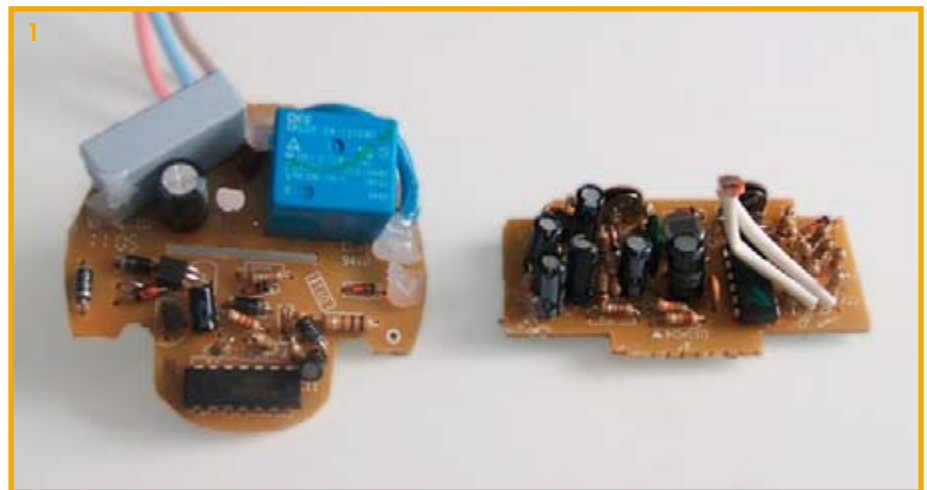
around 5 mm, but with significantly better efficiency than the CNY70, and it proves harder to dazzle. Lastly, if you double the stakes again (i.e. eight times the price of the CNY70), you can use the HOA1180-003, still from Honeywell, very hard to dazzle and which detects up to a distance of 15 mm.

(070300-1)

## PIR-Sensor

Abraham Vreugdenhil

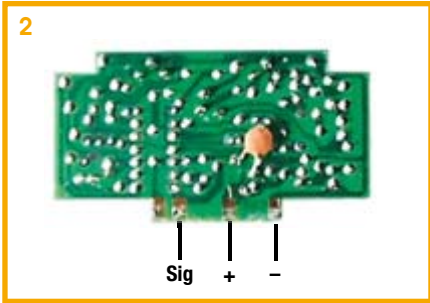
When designing a robot, a choice has to be made as to the types of sensors that it will have. This choice will be determined mostly by the purpose of the robot. But the degree of complexity required in using the sensor and the cost of the sensor also play a role, of course. Sensors that are favourable in these respects are for example bumpers and feelers with micro-switches, IR distance sensors from Sharp and ultrasonic sensors. If we want to detect moving warm objects, such as people and animals, then PIR (passive infrared radiation) sensors from Eltec, in particular, become a consideration, such as the Eltec-442. This is a very nice sensor, but the price is a problem unfortunately, more than 60 dollars. Conrad Electronics also have a PIR



sensor available, the LHI958 (order number 178730) for just over £ 2.50. The disadvan-

tage of this sensor is that an amplifier has to be added in order to obtain a usable



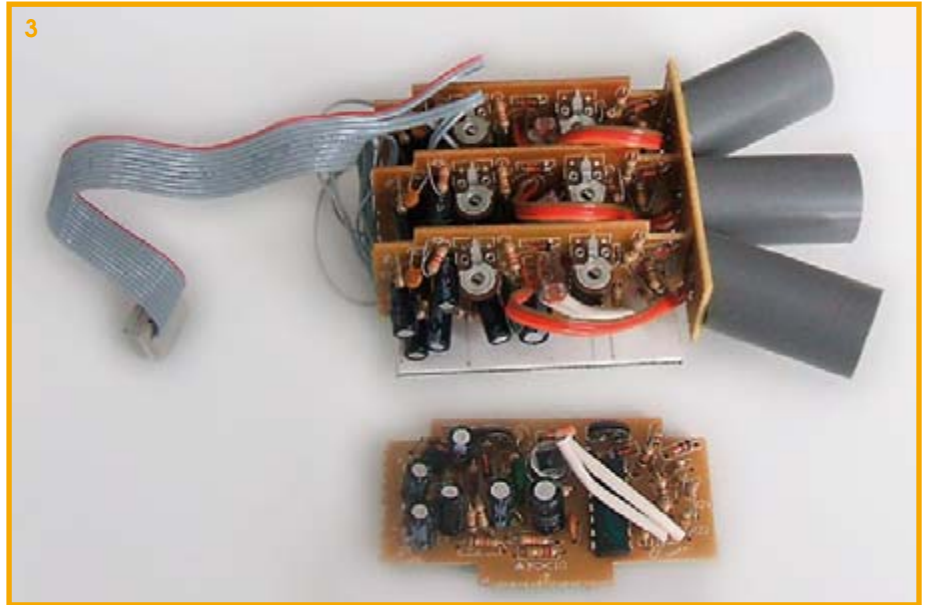


output signal. The documentation for the sensor is not particularly clear about this. Another solution is a sensor that we often meet in daily life: the well-known movement detector for outdoor lighting, which is available from any builder's market or hardware store for a reasonable price. These are offered for sale at less than £ 7.00. After disassembly of the sensor, the main board with its daughter board remain (Figure 1). The daughter board contains the PIR sensor and accompanying electronics. The connection points for the power supply and output signal can be found on the back (Figure 2). The sensor is normally powered from 8 V, but it still works well at 5 V. A robot will often be fitted with multiple PIR sensors that are mounted at different angles. To achieve this, we can mount

three sensors on a piece of prototyping board and limit the view of each sensor with a short section of electrical conduit. The length of the conduit determines the field of view. The sensors on their own have a field of view of 140 degrees, so the shielding is definitely required. It is advantageous if the fields of view of the sensors overlap. In this way three sensors can be

used to make five detection zones. It is of course also possible to use more sensors so that a greater resolution is obtained. In this manner it is reasonably cheap to build a nice PIR sensor unit. The one shown in the example (Figure 3) comprises three PIR sensors. This sensor unit is easy to build and works well.

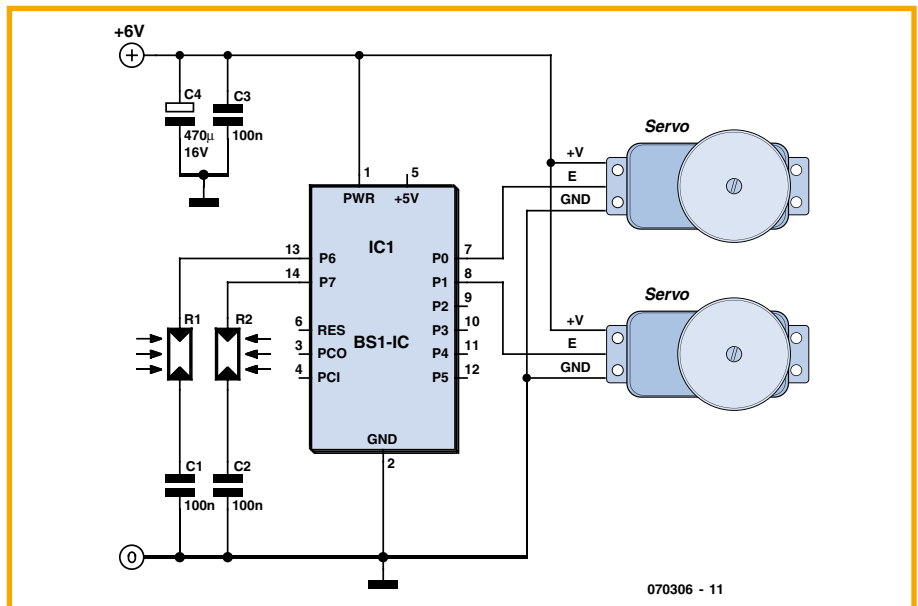
(070189-1)



# Light-seeking Robot

B. Broussas

Whatever the interest of the many types of robot that can be built today, the mobile robot is still an unmissable stage through which any robotics amateur has to pass, for at least two reasons. The mobile robot presents a concentration of the difficulties, and hence solutions, that may be encountered in robotics. You have to deal with problems of mechanics and kinetics in order to manage its movements, problems of sensors, which can be extremely diverse depending on what we want to detect, or on the other hand avoid, behavioural intelligence problems for processing the information provided by these sensors, etc. But the second reason why the robotics amateur needs to pass through the 'mobile robot' stage is often much less prosaic, as it's simply aimed at impressing the people around you (parents, friends, girl/boy-friend). What could be more impressive than this 'thing' straight out of a 50s sci-fi movie, moving around all by itself following a line on the ground, avoiding chair



legs, or responding to a signal from its master? So, robotics novices of all kinds, you will have realized that a mobile robot is what

we're going to suggest constructing. And so you'll be able to see quickly just what your own hands are capable of creating, we've chosen some solutions that are sim-

ple, but no less successful in producing a certain effect.

Quite simply, it's a light-seeking robot — a sort of moth (on wheels) if you prefer, since, just like its counterpart in the living world, it is always going to head for the brightest source it can find in the room where you let it loose.

To simplify construction and enable you to be up and running in just a few hours, or less, after reading this article, we suggest you take advantage of a mechanical base that's available in a kit. Having opted for propulsion using modified servo motors, we suggest two different bases: the Rogue Blue base from Rogue Robotics ([www.roguerobotics.com](http://www.roguerobotics.com)) or the Carpet Rover 2 base from LynxMotion ([www.lynxmotion.com](http://www.lynxmotion.com)). Of course, if you are good at mechanics, there's nothing to stop you building such a base yourself. It just needs to be propelled by two modified radio-control servos, and so will need a jockey wheel at the front and/or rear.

The modification to convert the servos into propulsion motors is explained elsewhere in this issue, but if you have any doubts about doing it yourself, as of quite recently you can also now buy such servos already pre-modified by their manufacturer. Take a look at Lextronic for example for this ([www.lextronic.fr](http://www.lextronic.fr)).

For our robot's 'brain', to show you it's not always necessary to use the very latest 32-bit microcontroller, we've decided to use the smallest of the Basic Stamps, the Basic Stamp I.

The complete circuit looks like **Figure 1**. The two ports P0 and P1 of the Basic Stamp I are used to drive the right and left propulsion servos. The brightness is measured using two photoresistors or LDRs (still called CdS cells in some literature) connected to ports P6 and P7 of the Basic Stamp I. The odd mode of connection used here makes it possible to use an instruction specific to the Basic Stamp I, the instruction **POT**, which measures the charging time of a capacitor connected to one of its ports, i.e. thereby the resistance of the LDR and thus the brightness falling on it.

For the robot to be able to head towards the brightest part of the room where it is operating, these two LDRs must be mounted pointing forwards, separated from each other by a small piece of cardboard or opaque PCB in such a way they can't both receive the same illumination.

The assembly can be powered by four 1.5 V batteries. This voltage is applied directly to the servos and to the unregulated PWR input of the Basic Stamp I. Watch out! Under no circumstances connect the servo supplies from the Basic Stamp I's + 5 V output – its built-in 5 V regulator wouldn't appreciate it!

The software part of our robot is at least as simple as the hardware part, as you can judge from the listing below:

#### Listing

```
PINS = 0
DIRS = %00001111
SYMBOL RightStop = 150
SYMBOL LeftStop = 150
SYMBOL Move = 30
SYMBOL LightDif = b2
SYMBOL RightLDR = b6
SYMBOL LeftLDR = b7
Main :
    POT 7, 128, LeftLDR
    POT 6, 128, RightLDR
    LightDif = RightLDR - LeftLDR
    b0 = RightStop + Move - LightDif
    b1 = LeftStop - Move + LightDif
    PULSOUT 0, b0
    PULSOUT 1, b1
GOTO Main
```

This listing is very easy to analyse. After an initial phase to define the labels used and the reservation of the RAM in the Basic Stamp I, we go on to measure the light using the instruction **POT**. This instruction returns, in the variable **LeftLDR** (or **RightLDR**), a number representing the resistance of the LDRs connected to P6 and P7 divided by a constant called a scaling factor. You may need to adjust this parameter to suit the characteristics of the LDRs you use.

The values thus obtained are subtracted

one from the other to yield information about the difference in lighting between the two cells. The calculation of the pulse lengths to be applied to the servos can then be performed, noting that **RightStop** and **LeftStop** are the values making it possible to make the servos stop, and that **Move** is a parameter intended to set the basic speed of the servos, to which is added or subtracted the result of the difference in illumination.

Hence, for example, if **LightDif** has the value 50, b0 will be  $150 + 30 - 50$ , i.e. 130, while b1 will be  $150 - 30 + 50$ , i.e., 170. Given that the resolution of the **PULSOUT** pulse is 10  $\mu$ s, the program will thus generate 1.3 ms pulses for one servo and 1.7 ms ones for the other, causing the robot to turn towards the direction of the LDR that is receiving the most light.

So this program is fully functional, but, given the spread in the characteristics of both the servos, with respect to their drive pulses, and the resistance of the LDRs, it will undoubtedly be necessary for you to tweak certain numerical parameters again to obtain satisfactory results. To do so, note that:

- **RightStop** and **LeftStop** are equal to 1/10 of the pulse width that makes the right and left servos stop.

- **Move** lets you define the rotational speed of the servos when the robot is going straight ahead. It is equal to 1/10 of the difference between the pulse width for stop and the pulse width desired for moving straight ahead.

- The coefficients 128 used in the **POT** instructions can also be adjusted between 1 and 255 in order to obtain satisfactory behaviour of the robot, given the LDRs used and the ambient light in the place where the robot is operating.

Note too that if your robot seems to shy away from the light instead of moving towards it, you've probably reversed the wiring between the right and left servos or right and left LDRs (all this is relative, of course, depending on what you call front and back on the robot).

Now it's over to you...

(070306-1)

## Bat's Ear

Alexander Wiedekind-Klein

Genau genommen ist To be precise, this ultrasonic distance measuring device is more than just an ear, since it generates pulses of sound at 40 kHz as well as lis-

tening for their reflections. The circuit in **Figure 1** is divided into two parts. At the top is the 40 kHz oscillator and a 'push-pull' output stage built around IC1.C and IC1.D. The oscillator is switched on and off according to the 5 V logic level on a

control input (pin 2 of K1). P1 is adjusted to set the oscillator frequency exactly equal to the resonant frequency of the ultrasonic transducer, nominally 40 kHz.

Reflected signals are amplified by IC2.A and IC2.C, rectified by D1 and buffered

by IC2.D. This circuit forms the analogue front end, and is connected to a microcontroller for subsequent signal processing. We will look below at the factors that need to be borne in mind when considering the digital signal processing algorithm to be used.

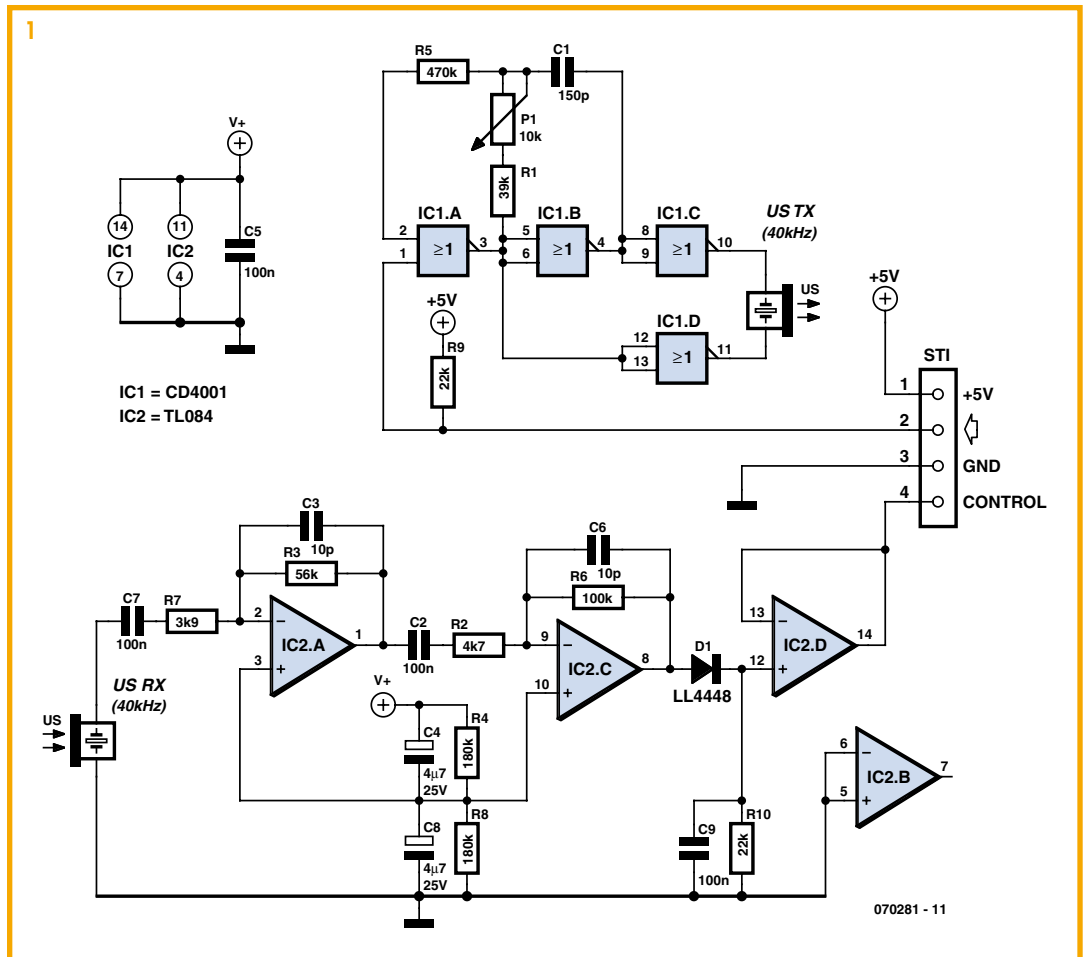
After a burst of ultrasound lasting approximately 2.5 ms is transmitted we sample the envelope of the received signal as delivered by the analogue front end for approximately 50 ms. In this time, sound travels approximately 16 m, and so we have a maximum range of 8 m, because the sound must travel to the distant object and back. **Figure 2** shows a typical received signal. The green rectangular pulse represents the signal at the control input (pin 2 of K1), which switches on the oscillator for the 2.5 ms pulse period. During this time ( $t_1$ ) we can already see some signal at the output of the receiver,

as it is impossible to avoid some direct reception of the transmitted pulse. This effect has to be taken into account in subsequent processing.

The second peak in the signal, after time  $t_2$ , is a reflection from an object. The time is proportional to the distance to the object. Measurement of time  $t_2$  commences at the middle of the transmit burst (i.e., approximately 1.25 ms after the oscillator is enabled), and finishes when the amplitude of the reflected signal reaches its peak value. In air the distance to the object measured in centimetres is easy to calculate: to a good approximation it is equal to the time to the reflection in milliseconds multiplied by 16. For example, a time of 10 ms for  $t_2$  corresponds to a distance of 160 cm.

If an object is very near, the reflected sound will be very loud and be received after a very short time, possibly while the pulse is still being transmitted (**Figure 3**). In this case it is best to measure the time taken for the received signal to reach half its maximum amplitude from when the oscillator is switched on. This time can then be used to form an estimate for the distance to the object.

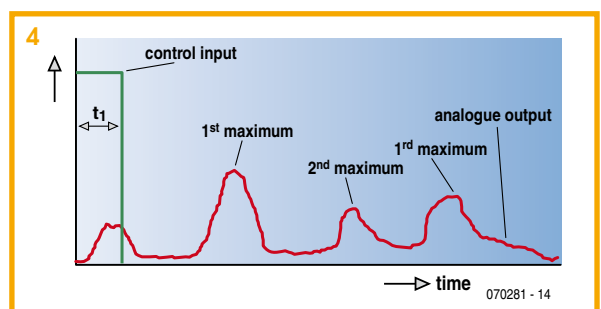
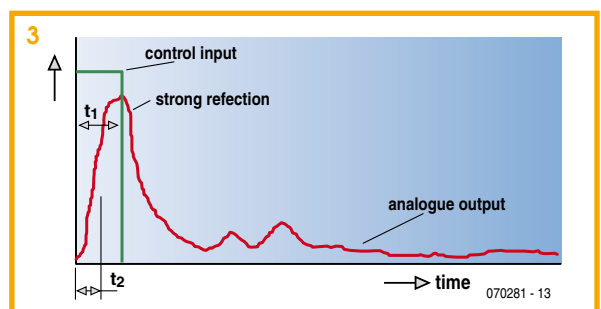
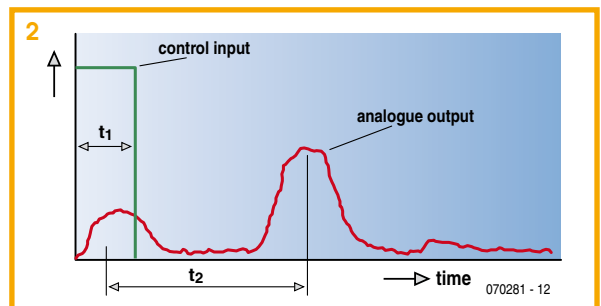
If there is a number of reflecting objects at different distances there will be several reflected pulses of different amplitudes



(**Figure 4**). In this case it quickly becomes apparent that if we used a simple threshold detector in place of the microcontroller it would be impossible to get reliable results. A microcontroller with 'intelligent' software could, for example, be programmed to calculate the distance to the nearest object or to report the distances to several objects simultaneously.

Since the only connections to the circuit are a +5 V supply and ground, a control signal for the transmitter and the analogue envelope signal returned by the receiver to the processing hardware, it is straightforward to wire up four copies of the circuit mounted at right angles to one another. In a robotics application this would give the robot the ability to detect objects in any position relative to itself.

(070281-1)





# An Inclinometer for Your Robot

C. Tavernier

www.tavernier-c.com

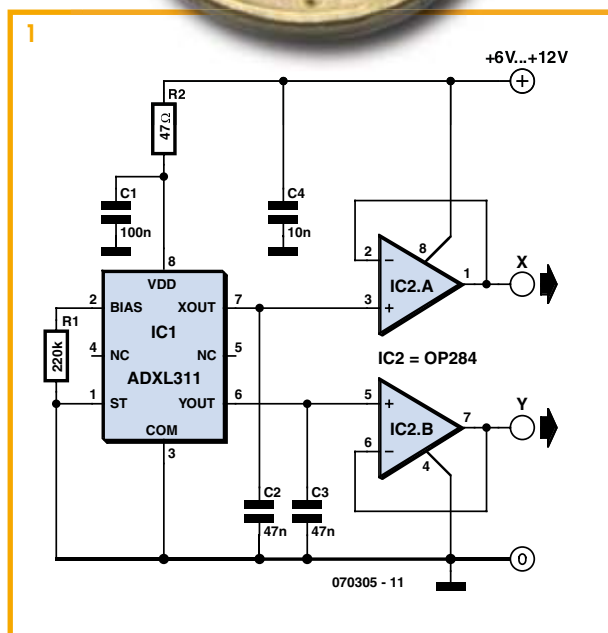
If your mobile robot's sole function is to roam about the tiling or wooden floors of your home, it's not very likely to have much need of the sensor we're going to be describing in this article. However, if it has to confront the harsh realities of the ground of the outside world, with its holes and bumps, an inclinometer may prove extremely useful in order for it not to keel over at the first, ever so slight unevenness.

Before electronics knew how to accomplish all the feats that we are used to today, an inclinometer was a purely mechanical system, with all the difficulties of implementation, cumbersomeness, and lack of accuracy this implied. Might as well say that its use in an amateur robot was, if not impossible, at least very difficult.

For a few years now, this has no longer been the case, thanks to the marketing by Analog Devices of 'solid state' accelerometers, i.e. produced in the form of ICs, with no visible moving parts.

In fact, it's still impossible to measure acceleration without employing some kind of moving part, but nowadays this consists of a minute polysilicon structure suspended by four springs of the same material above the chip of the accelerometer IC. When this mobile element is subjected to acceleration, it deforms, and this deformation is revealed by a variation in the capacitance between a plate located on the mobile element and two fixed plates on the chip itself. Two out-of-phase squarewave signals are applied to the fixed plates. When the mobile plate is subjected to acceleration and moves, these become unbalanced, and phase demodulation yields a voltage proportional to the acceleration.

Of course, the user is quite unaware of all this going on, but has



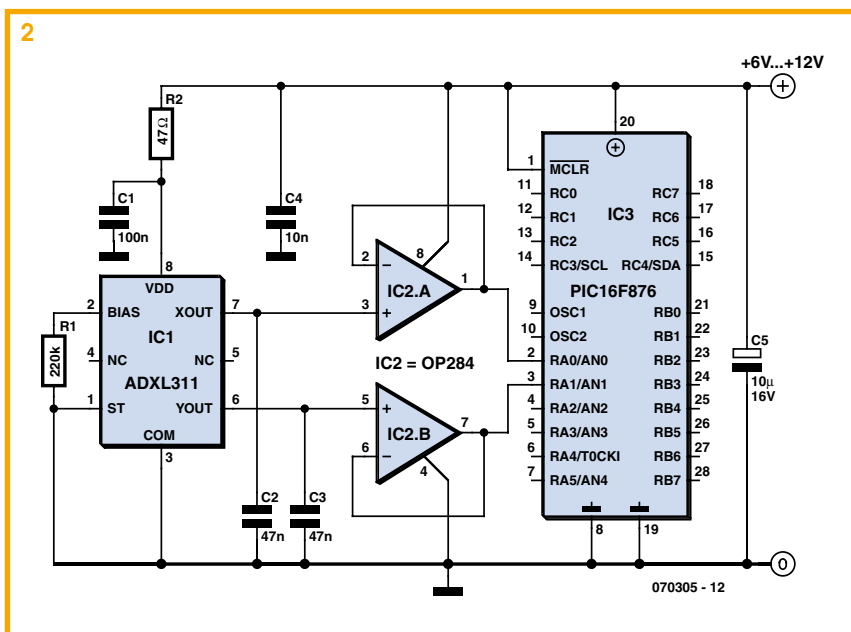
available at the accelerometer IC output information reflecting the acceleration registered — in analogue or digital

form, depending on the type of IC chosen.

For our robotics application, we have decided to adopt a relatively inexpensive accelerometer in the form of the ADXL311 from Analog Devices. Do note right away, however, that this IC is no longer being produced, but is still widely available from retailers. If it should eventually disappear completely, it could be replaced by the ADXL320, much more recent and electrically compatible, only the pin-out being different.

The ADXL311 actually includes two highly sensitive accelerometers at right-angles, with positioning better than 0.1°. Because of this, and if it is placed parallel to the surface of the Earth, it is influenced by the acceleration due to the Earth's gravity, and so can indicate left/right inclination (roll) or forward/backward (tilt). In this way we create a dual-axis inclinometer.

These inclinations can be exploited in an absolute form if we want to know exactly the angle between the inclinometer, hence the IC carrying it, and the ground, or in a relative form, if we want just a limit indication of what the robot can withstand before it topples over.



In the case of the ADXL311 or the ADXL320, the absolute inclination is given by the equations:

$$\text{Tilt} = \arcsin (A_x/A_{x0})$$

and

$$\text{Roll} = \arcsin (A_y/A_{y0})$$

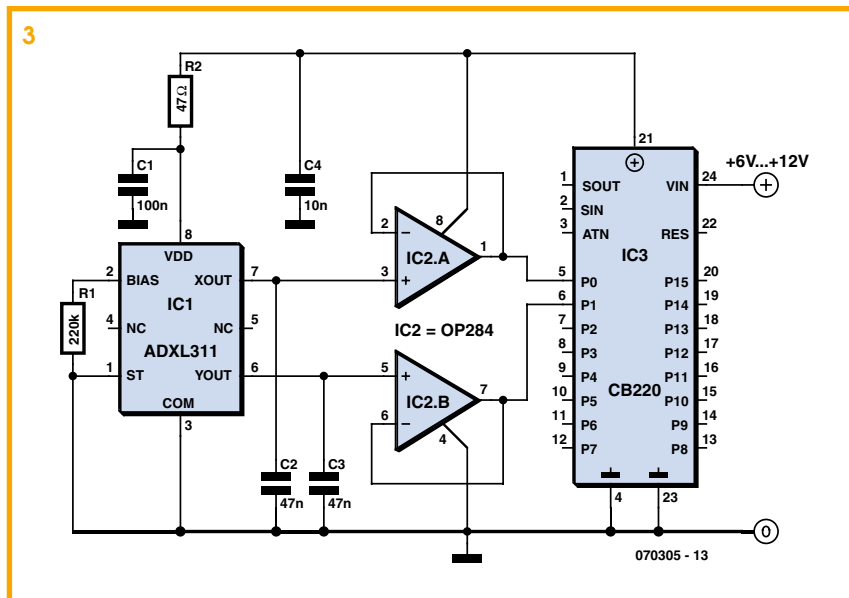
where  $A_x$  and  $A_y$  are the analogue voltages supplied by the accelerometer when it is inclined, and  $A_{x0}$  and  $A_{y0}$  the voltages supplied when it is perfectly horizontal.

As shown in **Figure 1**, the accelerometer application circuit doesn't amount to very much. The only important elements are in fact capacitors C2 and C3, which filter the accelerometer output voltage. It actually has a passband of several kHz, and can therefore react to very fast vibrations, not at

all what we want in an application as an inclinometer. With the values chosen here, the passband is restricted to 100 Hz, easily enough.

The two op. amps arranged as follows avoid any external influence on these filter capacitors and allow the accelerometer to be connected without special precautions to any microcontroller or ADC input.

The only minor problem you might encounter constructing this project is that the accelerometer is in an SMD package, which is not always easy for soldering onto an amateur PCB. There is now a module, ref. Accel, from Lextronic ([www.lextronic.fr](http://www.lextronic.fr)), that includes all the components in Fig-



ure 1 mounted on a tiny 15 × 20 mm PCB (see photo).

Using our accelerometer as an inclinometer involves measuring its analogue output voltages on both axes and subtracting from them the voltages at rest, that is, when the IC socket is perfectly parallel to the ground.

**Figure 2** shows an example of the use of this inclinometer with a PIC microcontroller with a built-in ADC, while **Figure 3** shows the same type of circuit, this time with a Cubloc CB220.

We are not giving you a program for exploiting the information supplied by the inclinometer as, in both cases alike, it takes

just two instructions to access the tilt or roll information.

So for example you would write

```
tilt = Adin(0)
tilt = tilt - 512
` adjust according to the voltage output
` when the inclinometer is horizontal
```

to recover the tilt information using a Cubloc CB220.

While you might write, for example,

```
t i l t =
Adc_Read(0)
tilt = tilt - 512 ` adjust according to the voltage output
` when the inclinometer is horizontal
```

to recover the tilt information with a PIC programmed in Basic (in this example, MikroBasic compiler and 10-bit ADC).

The same instructions will obviously be used for roll, but using analogue channel 1 if you have adopted the circuit diagrams of Figures 2 or 3. Then it only remains for you to exploit this data to prevent your robot's falling over on terrain that's too steep for it!

(070305-1)

# Positioning with Photodiode Arrays

Pascal Choquet

Industrial production-line robots require high-resolution sensors to measure the position of robotic actuators so that the tools can be accurately guided onto the work piece. A photo diode array together with an external light source is often used as a sensor in this application. This chip contains a line of photo diodes together with a series of sample and hold (S/H) circuits which take a snapshot of the readings of each photo diodes at the same instant and then outputs these integrated analogue values serially from a single output. The S/H circuits are important because the sensitivity of each element to the light quanta is dependent on the integration period; without the S/H the last element would show

the highest sensitivity.

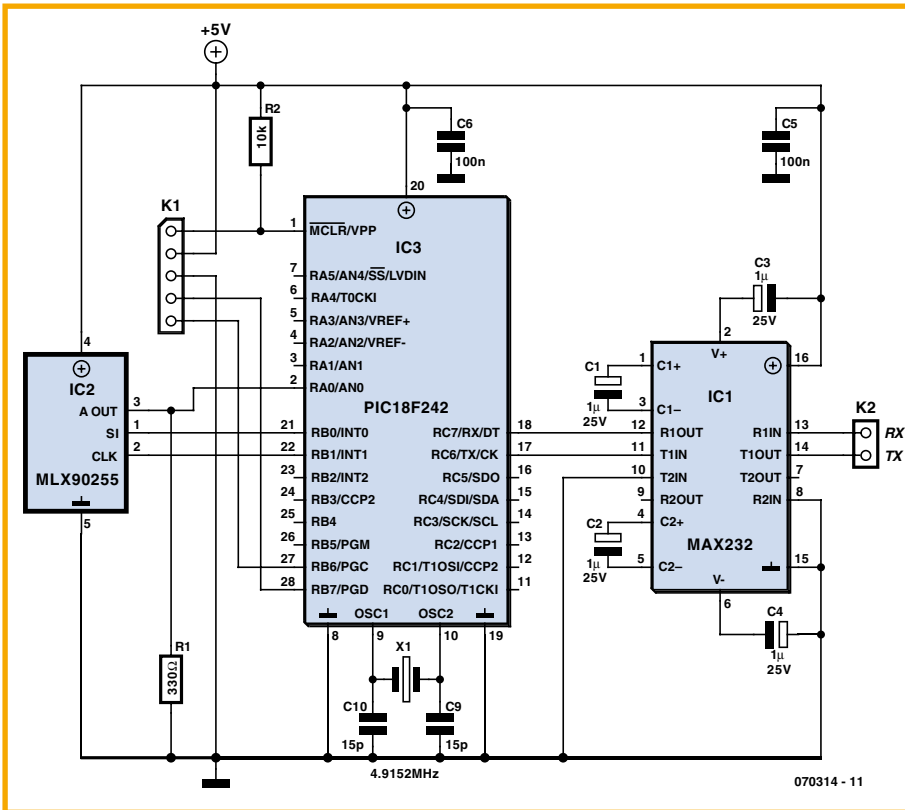
The accompanying table lists the most important properties of some common arrays. The sensitivity is dependant on the active diode surface area and the integration time.

From the outside these arrays look very simple, apart from the two supply connections there are only three signals for connection to a microcontroller: A clock input (CLK), a start impulse (SI) input and an analogue output signal (AO). AO should be loaded with a 330 Ω resistor to ground. To readout the array values the controller firstly generates the clock signal and then sets SI high (with sufficient set-up time) before a rising clock edge. On successive falling clock edges the value of each

individual pixel will be output at AO. The microcontroller reads each level and stores its value.

A feature of the MLX90255 type array is that the first two values read out are dummies, the first of the 128 real values appears at the third clock edge. The two values after the 128<sup>th</sup> value are also dummies, the read out therefore requires 132 clock edges in total to read all the data and the final 133<sup>rd</sup> edge reinitialises the shift register. The gain of the pixels at either end of the array is about 15 % greater than those in the centre (cosine weighting) this compensates for the light loss experienced at the edges when the array illumination is provided by a single LED.

The integration period begins at the 18<sup>th</sup>



clock edge and continues until the next SI signal. The output values are the result of the previous integration period so if the array is not continuously scanned then it is necessary to make two complete scans to get meaningful results. The first scan cycle after power up is used to initialise digital levels on the chip, the values read are invalid and should be discarded. The integration time is equal to the pixel count minus 18 divided by the clock frequency. The sensitivity can be easily controlled by the microcontroller.

The circuit diagram shown here consists of the photodiode array together with a basic PIC microcontroller and a driver chip for an RS232 serial interface connection. The author has produced a program written in C which can be used in the controller. The source files (**070314-11.zip**) are available to download free of charge from the *Elektor Electronics* website. A scan cycle is initiated using a terminal program by entering 'Strg S'. The values are separated with semicolons so they can be easily used in an Excel table.

Photodiode arrays can be used in robotics for imaging based on the pinhole camera principle, they have also been used in line-following applications where they offer good resolution and can be mounted relatively far away from the floor. Together with a prism or optical grating the array can be used to perform simple yet precise colour recognition.

(070314-1)

Photodiode Array properties				
Array	MLX90255	TLS1301	TLS1401	TLS208R
Pixel (* see text):	128 (+ 4*)	102	128	512
Pitch (DPI):	385	300	400	200
Weighting:	Cosine	Equal	Equal	Equal
Length x width (µm):	200 x 66	85 x 77	63,5 x 55,5	120 x 70
Output (V):	0.125 - 2.4	0 - 2.0	0 - 2.0	0 - 2.0

# Whiskers on Robots

Alexander Wiede-kind-Klein

Sometimes sophisticated sensors based on video cameras, infrared or ultrasound are not quite up to the job, and we have to resort to somewhat more primitive switch-type sensors. These work like an animal's whiskers (or 'antennae' on insects), detecting nearby objects in the environment. When disturbed, a sensor sends a pulse to the robot to indicate that an object is present. Sensors that are both sensitive and robust



can be made using steel guitar strings. The material is very flexible as well as being

conductive. The idea could not be more simple: we pass the wire through a metal tube, and when the wire is in contact with an object in the environment it bends and the two make electrical contact. Responsiveness and sensitivity depend chiefly on the length and stiffness of the piece of guitar string used.

Do-it-yourself construction should not present great difficulties even to the most mechanophobic

reader: see **Figure 1**. We proceed as follows.

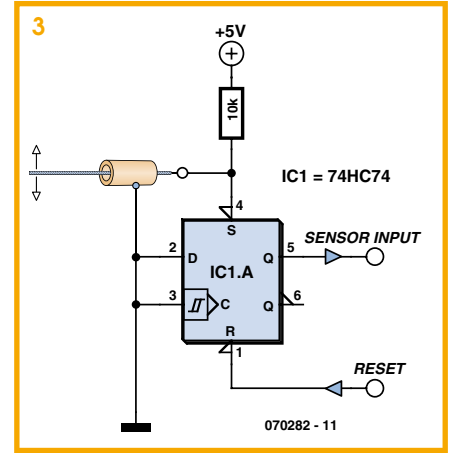




1. Cut a length of steel guitar string (8 cm to 10 cm is enough), and saw off a length of about 2 cm of 4 mm diameter brass tube. Deburr the edges inside and out.
2. Solder a wire to one end of the brass tube and another wire to the end of the guitar string. Insulate the joint on the guitar string using heatshrink tubing.
3. Slide the string into the tube so that only about the first 10 mm from the end of the tube is insulated. Fix the string centrally

within the tube using hot-melt glue. Be careful not to allow too much glue to run down inside the tube.

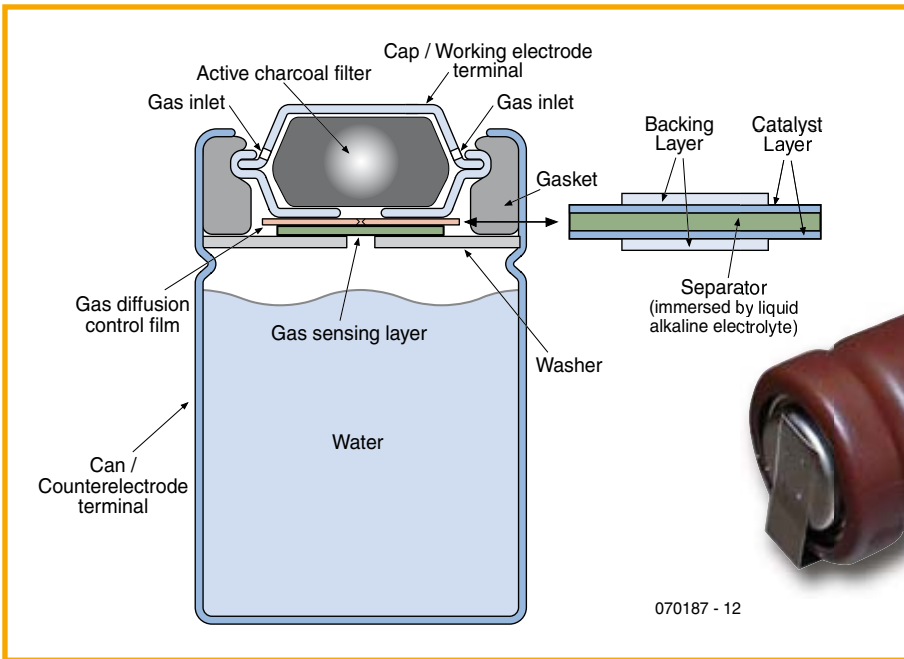
The result should look like **Figure 2**. Of course, you are free to experiment with variations on this construction! As we have described it this robot whisker is essentially just a simple switch contact. To ensure that even the gentlest collision does not pass unnoticed we recommend that you use the whisker to trigger a flip-flop



as shown in **Figure 3**. The microcontroller in the robot can then read the state of the flip-flop at its leisure and then reset it.

(070282-1)

# CO Sensor



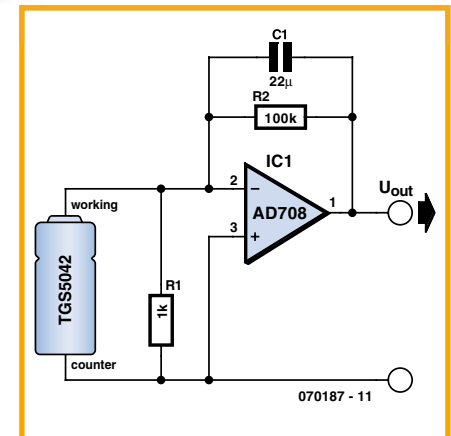
**Figure 1.** Internal structure of the sensor.

The Figaro TGS5042 sensor is a carbon monoxide sensor that is used primarily in industrial applications such as smoke detectors, fire detection equipment and ventilation controllers for indoor car parks and the like. The sensor is quite suitable for use in battery-powered applications, and it has several advantages over conventional sensors. The electrolyte is environmentally

friendly, and the housing is leak-proof. The sensor can measure CO concentration up to 1%, and it has a temperature range of  $-40\text{ }^{\circ}\text{C}$  to  $+70\text{ }^{\circ}\text{C}$ . The housing has the same form as an AA battery.

### A few specifications:

- Suitable for use in battery-powered equipment



**Figure 2.** A simple sample application circuit.

- High sensitivity and accuracy for CO
- Linear relationship between CO concentration and output voltage
- Low sensitivity to ethanol
- Low sensitivity to other gasses that may be present

**Figure 1** shows the internal structure of the TGS5042 sensor. The gas-sensitive layer for CO is located between a stainless-steel ring (counter electrode) and a nickel-plated cap (working electrode). The cap is packed in a sort of film and several supporting lay-

ers. All of this is packaged in a cylindrical stainless-steel housing. The lower compartment is filled with water, and the cap end is filled with an activated charcoal filter.

**Figure 2** shows the schematic diagram of a basic application circuit for the TGS5042. The sensor generates an extremely small current, which is converted into a voltage by an instrumentation amplifier formed by IC1 and R2. Resistor R1 is necessary to prevent polarisation of the sensor, which might otherwise occur when the circuit is switched off.

It is essential to avoid applying a voltage to the sensor under any conditions. Doing so would permanently damage the sensor. The voltage across the sensor must always be less than 10 mV.

**Some potential applications for the sensor are:**

- Residential CO detectors
- CO monitors for industrial applications
- Ventilation control for indoor car parks

(070187-1)

# Compass Sensor for Lego Mindstorms NXT

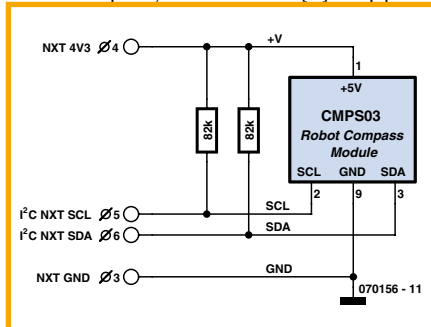
Zeno Otten

A few years ago, a considerable amount of attention was devoted in Elektor Electronics to the construction of sensors for the intelligent control brick (RCX) from Lego Mindstorms [1].

There is now a successor. The NXT is the heart of the new Mindstorms. Using this system, computer hobbyists can develop, build and, in particular, program, numerous robots, to their heart's content.

With the compass sensor that is described here the NXT can determine its direction with an accuracy down to a few degrees. This allows a robot to be built that's capable of navigation.

The company Devantec [2] supplies



a ready-to-go compass module type CMPS03. Two mutually perpendicular Philips KMZ51 hall-sensors are used to detect the Earth's geomagnetic field. With a small PIC-controller a value between 0 and 360 degrees is calculated and made available in digital form at the output of the module. The communication with the outside world takes place via the I<sup>2</sup>C protocol or via a PWM output.

The module requires a 5 V supply voltage

and consumes about 20 mA. This module is eminently suitable for use with the NXT.

The new NXT has the option of connecting sensors that use the I<sup>2</sup>C protocol. This allows a sensor to be connected to the NXT using an RJ12 plug.

The NXT does not have internal pull-up resistors on the I<sup>2</sup>C bus. So, these have to be added externally. Lego suggest resistors with a value of 82 kΩ on both the data line (SDA) as well as the clock line (SCL).

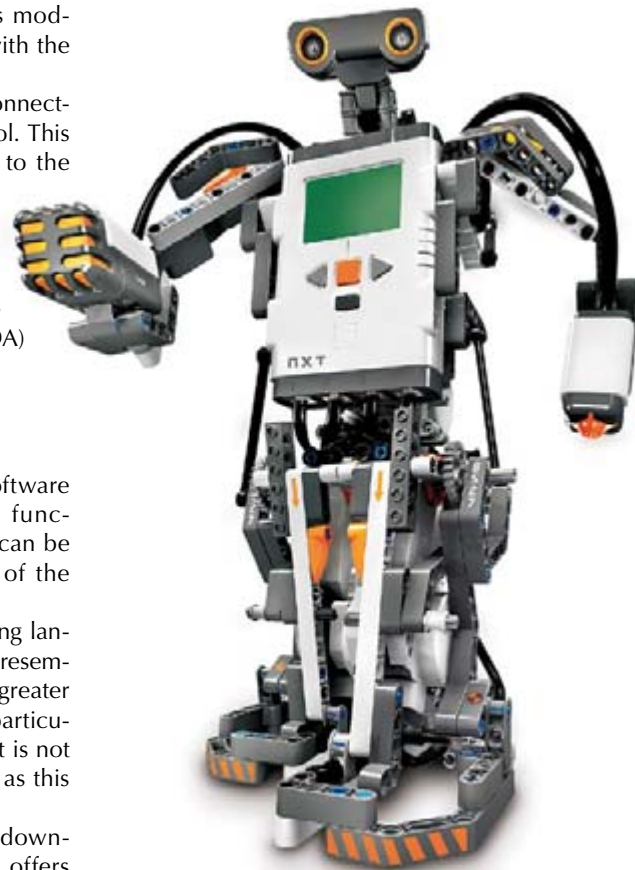
## Software

The standard Lego Mindstorms software is based in dragging graphical function blocks. Only the parameters can be changed while the functionality of the blocks is fixed.

Not Exact C (NXC) is a programming language for the NXT that has a strong resemblance to C. This permits a much greater flexibility when programming. In particular when it concerns hardware that is not officially supported by Lego, such as this compass sensor.

The compiler (BricX) [3] can be downloaded free, is simple to use and offers many options for programming the NXT.

The program compass.nxc (which can be downloaded free from the Elektor Electronics website as file number **070156-11.zip**) continually reads the compass sensor. The measured values are then processed by the robot who will sequentially 'point' to North, South, East and West. The values measured by the electronic compass can also be read from the display on the NXT brick.



(070156-1)

## References

- (1) Compass sensor for Lego RCX, Elektor Electronics July/August 2002
- (2) Devantec: [http://www.robot-electronics.co.uk/shop/Compass\\_CMPS032004.htm](http://www.robot-electronics.co.uk/shop/Compass_CMPS032004.htm)
- (3) BricX: <http://bricxcc.sourceforge.net/>

# Ultrasonic Distant Obstacle Detector

B. Broussas

The first sensor a robot usually gets fitted with is an obstacle detector. It may take three different forms, depending on the type of obstacle you want to detect and also — indeed, above all — on the distance at which you want detection to take place.

For close or very close obstacles, reflective IR sensors are most often used, an example of such a project appears elsewhere in this issue. These sensors are however limited to distances of a few mm to ten or so mm at most.

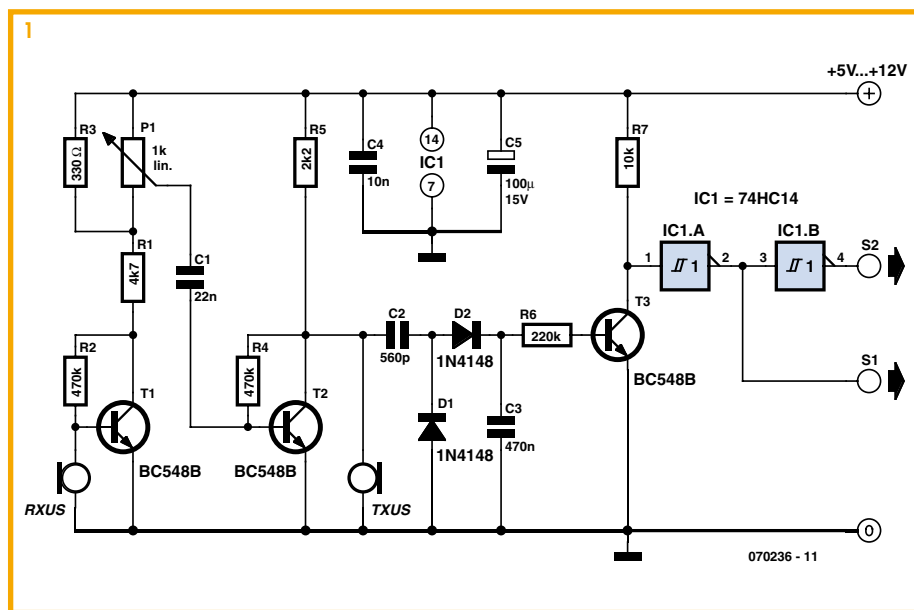
Another simple and frequently-encountered solution consists of using antennae-like contact detectors or 'whiskers', which are nothing more than longer or shorter pieces of piano wire or something similar operating microswitches. Detection takes place at a slightly greater distance than with IR sensors, but is still limited to a few cm, as otherwise the whiskers become too long and hinder the robot's normal movement, as they run the risk of getting caught up in things around it.

For obstacles more than a couple of cm away, there is another effective solution, which is to use ultrasound. It's often tricky to use, as designers think as if they needed to produce a telemeter, when in fact here we're just looking at detecting the presence or absence of obstacles, not measuring how far away they are.

So here we're suggesting an original approach that makes it possible to reduce the circuit required to a handful of cheap, ordinary components. Our solution is based on the howlround or feedback effect all too familiar to sound engineers. This effect, which appears as a more or less violent squealing, occurs when a microphone picks up sound from speakers that are connected to it via an amplifier. Feeding back the output signal from the speaker into the input (the microphone) in this way creates an acoustic oscillator.

Our detector works on the same principle, except that the microphone is an ultrasound receiver while the speaker is an ultrasonic emitter. They are linked just by a very easily-built ordinary amplifier. Feedback from the output to the input occurs only when the ultrasonic beam is reflected off the obstacle we are trying to detect.

As **Figure 1** shows, the receiver RXUS is connected to the input of a high-gain



amplifier using transistors T1 and T2. As the gain of this stage is very high, it can be reduced if necessary by pot P1 to avoid its going into oscillation all on its own, even in the absence of an obstacle. The output of this amplifier is connected to the ultrasonic emitter TXUS, thereby forming the loop that is liable to oscillate due to the effect of feedback.

When this takes place, i.e. when an obstacle is close enough to the ultrasonic transducers, a pseudo-sinewave signal at their resonant frequency of 40 kHz appears at the amplifier output, i.e. at the terminals of the transmitting transducer. This signal is rectified by D1 and D2 and filtered by C3 and, if its amplitude is high enough, it produces a current in R6 capable of turning transistor T3 on to a greater or lesser extent.

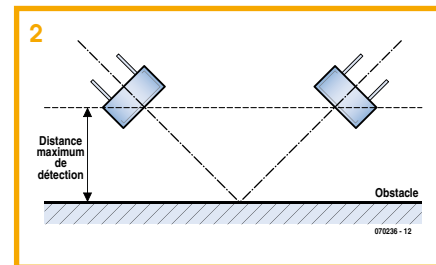
Depending on the nature and distance of the obstacle, this process does not necessarily happen in a completely on/off manner, and so the level available at T3 collector may be quite poorly-defined. The Schmitt CMOS inverters are there to convert it into a logic signal worthy of the name. So in the presence of an obstacle, S1 goes high and S2 goes low.

Powering can be from any voltage between 5 and 12 V. The gain, and hence the circuit's detection sensitivity, does vary a bit with the supply voltage, but in all cases P1 makes it possible to achieve a satisfactory setting.

Although it is very simple, under good con-

ditions this circuit is capable of detecting a normally-ultrasound-reflective obstacle up to around 5 or 6 cm away. If a smaller distance is needed, you simply have to reduce the gain by adjusting P1.

Building the circuit is straightforward. Both transducers are 40 kHz types that can be found in any retailers, and the other components couldn't be more ordinary. However, one precaution is needed when wiring up the transducers. Even though they aren't strictly speaking polarised as such, one of their terminals is common with the metal case, and this is the one that must be connected to the circuit earth, on both



emitter and receiver. The circuit should work at once, and all you have to do is adjust P1 to set the detection distance you want — but this is also dependent on the positioning of the transducers. For optimum operation, we recommend you angle them as shown in **Figure 2**.

(070236-1)



# Light Sensing with an LED

Andreas Grün

Many robotic applications require a sensor to measure light levels. The conventional approach as shown in **Figure 1** uses an A/D converter to measure the voltage drop across resistor R1 produced by the photo current through a photo transistor. The fixed value of R1 limits the light range which can be measured; a high resistor value is suitable for measuring low light levels while a low resistance is good in bright conditions. The resolution of the A/D converter also plays a part in determining the range of light levels that can be measured.

A little-used property of a standard LED

is its reverse-biased photocurrent mode. An LED also produces a light-induced photocurrent but at a much reduced value compared to a photo-transistor. Direct measurement of the current is not so easy but another property of the diode can be exploited which is described in [1]. In this paper it explains that a useful property of a reverse-biased LED is its relatively large capacitance, the technique is to charge up this capacitor and then allow the photocurrent to discharge it. The time taken for the capacitor to discharge is dependant on the amount of light falling on the LED (**Figure 2**). Charging and time measurement can be easily performed using a single I/O pin of a microcontroller and switching it between output mode and high-impedance input mode. The measurement is performed in two stages:

1. The pin is configured as an output and set to high to charge up the LED capacitance (**Figure 3**).
2. The pin is configured as an input Pin (any pull-up resistor is disconnected) and the time is measured until the input voltage

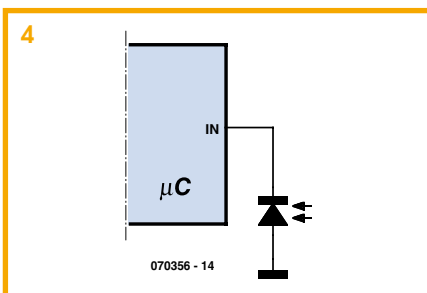
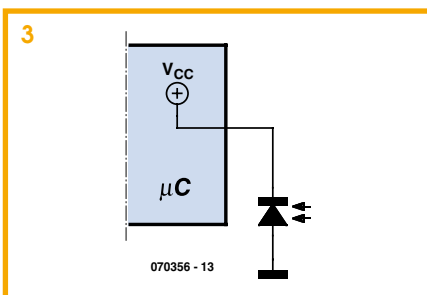
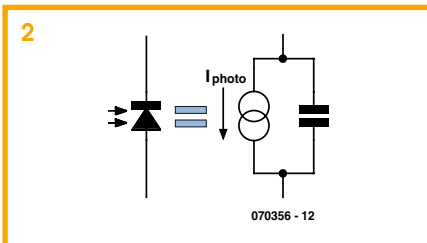
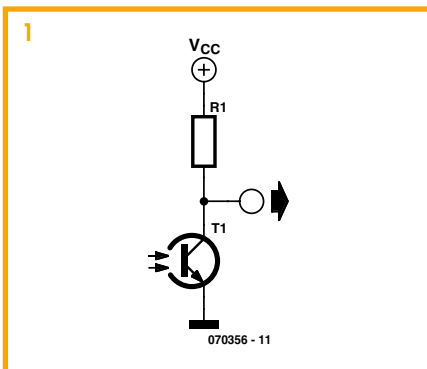
level falls below the lower input threshold level (**Figure 4**).

The **Example program** is a listing for an Atmel AVR processor which measures light intensity. The program toggles all the output bits from port A after each cycle so that it produces an output square wave with approximately 50 % duty cycle at a frequency proportional to the measured illumination. The frequency varies from millihertz (in a darkened room) up to several hundred kilohertz when light shines directly onto the LED. This measurement range would be difficult to achieve using an A/D converter. Narrow beam LEDs have a corresponding narrow 'detection angle' making them more directional which may be beneficial in some applications. Different LEDs are sensitive to specific colours which can also be useful in some robotic applications.

(070356-1)

## Web link

(1) [www.merl.com/publications/TR2003-035/](http://www.merl.com/publications/TR2003-035/)



## Listing Example program:

```
#include <avr/io.h>
#include <avr/interrupt.h>

#define LEDPIN 0x40 // LED on PB6

int main()
{
    unsigned char cr=0,cb=0;
    DDRB = 0x00; // PORTB input
    DDRA = 0xff; // PORTA output for display LEDs
    PORTB = 0; // off
    PORTB = LEDPIN; // PB6 hi

    sei();

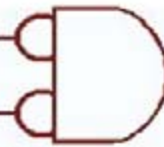
    while(1)
    {
        if((PINB & LEDPIN) == 0) // discharge complete
        {
            PORTB = LEDPIN; // PB6 hi

            // multiple times
            to get enough charging time
            DDRB |= LEDPIN; // PB6 output and hi, charges LED
            DDRB |= LEDPIN; // PB6 output and hi, charges LED
            DDRB |= LEDPIN; // PB6 output and hi, charges LED
            DDRB |= LEDPIN; // PB6 output and hi, charges LED

            DDRB &= ~LEDPIN; // PB6 input, still charging w/ pullup
            PORTB = 0; // switch off pullup
            PORTA ^= LEDPIN; // toggle PORTA for display LEDs
        }
    }

    return(0);
}
```

## Compact Control Design



Compact Control Design has a range of off the shelf products designed to meet your control, monitoring and acquisition needs. All products use the latest technology to provide top performance in a small size and at a low cost. We also provide a full hardware and software design service.

### P0701 USB PIC

USB 2.0 interface. Can be bus powered or self powered.  
Powerful PIC18F4580 Microcontroller running at 40MHz. Up to 10MIPS performance.  
All microcontroller I/O pins available except RA6, RA7 (oscillator) and RC6, RC7 (serial port).  
Bootloader pre-programmed and download software included, enabling quick and easy programming of applications.  
No need for a separate programming device.  
Many example applications and firmware available, see Compact Control Design download page.  
Connector has standard 40 pin 0.1" pitch 0.6" wide footprint.  
High quality tuned pin connectors suitable for most IC sockets and prototyping boards.

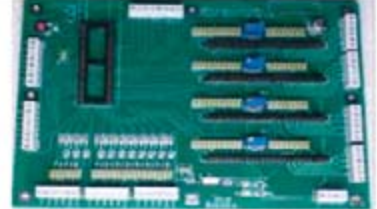


Compatible with Microchip's MPLab 'free of charge' programming environment. Libraries and linker scripts included to support assembler programs (MPLab) and popular compilers.

**PRICE:£26.00 + VAT**

### P0704 Developer Board

The P0704 developer board is an ideal way to get started with our USB-PIC module and motor driver modules. All of the USB-PIC module I/O signals are available through screw terminals making connections to sensors, switches, lamps, relays etc. easy. Ports B & E (11 I/O signals) can also be configured with pull-ups and input filtering suitable for connection to limit switches, home position sensors etc. The board supports up to 4 motor driver modules, each module position accepts either a Stepper motor module or a DC motor module. The board allows bus-powered or self powered operation of the USB-PIC module & includes a P0615 mini regulator so only a single power supply is required for the motor driver modules.



All options are configured using jumpers, and stepper motor drive current can be easily adjusted for each module by variable resistors. All connections are made by high quality screw terminals. The board has been designed to accommodate other driver modules as they become available.

**PRICE:£55.00 + VAT**

### P0613 DC Motor

Pulse width modulation control for DC motors, electro-magnets etc.  
It has a motor supply voltage of 8 to 36V. The maximum drive current is 2.5 Amp.  
There are pulse and direction inputs. The PWM control is up to 100KHz.  
Mode input for controlling motor braking and sleep input for power saving.  
There is built in short circuit and over temperature protection, a fault output pin activates if either of these is detected. No heat sink is required.  
The board has dimensions of 66x30mm and is 12mm high.  
There is an adapter available providing easy to use screw terminals for all connections. All the control inputs are opto-isolated.



**PRICE:£14.00 + VAT**

### P0612 Stepper Motor Driver

The unit has a motor supply voltage of 5 to 30V. The maximum drive current per phase is 750mA.  
It has current mode control. The drive current is controlled with a resistor. It has a selectable step size of full, half, 1/4 + 1/8.  
There is a step frequency of 0 to 200KHz and reset and sleep inputs for initialization and power saving.  
It is a compact size with dimensions of 66x32mm by 12mm high.  
The P0612 does not require a heat sink. There is an adapter available which provides easy to use screw terminals for all connections. All the control inputs are opto-isolated.



**PRICE:£15.00 + VAT**

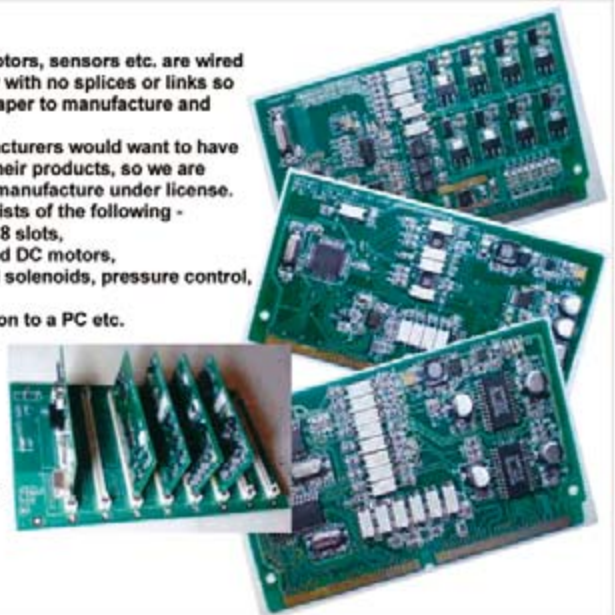
### MonCon

MonCon is a product range intended to form the intelligence at the heart of any equipment from bench-top scientific instruments, production equipment, ATE etc. up to large process control systems.  
The MonCon range takes a new approach to monitoring and control by using modules that encapsulate a complete task, such as the stepper motor controller module that includes all inputs and outputs necessary to form a complete stepper motor controller/driver including encoder feedback.  
The general purpose modules, such as the Analogue input board are designed to be customized at minimum cost. We can supply such modules to your requirements at little or no additional cost.  
The MonCon range is based on a collection of modules, each performing specific and well defined tasks. All modules plug into a back plane which provides power distribution, intercommunication and incorporates the necessary connectors linking the MonCon system to the rest of the equipment.  
The modules and backplane connectors have been designed to simplify the interconnection requirements within your equipment.

Most devices, such as stepper motors, sensors etc. are wired to the MonCon backplane directly with no splices or links so the wiring loom is simplified, cheaper to manufacture and more reliable.

We understand that many manufacturers would want to have full control over critical parts of their products, so we are happy to allow our customers to manufacture under license. The product range currently consists of the following - standard back planes with 4, 6 or 8 slots, controller modules for stepper and DC motors, controller modules for valves and solenoids, pressure control, flow control etc. a USB interface to allow connection to a PC etc. various I/O modules, Parallel I/O, relay output and Analogue I/O modules.

The MonCon range has been designed with flexibility in mind. Backplanes and modules to meet your requirements can generally be designed & supplied within 6 weeks of receiving a full specification.





# MotoBox

## Drives stepper, DC and servo motors

Joseph Zamnit (MCAST, Malta)

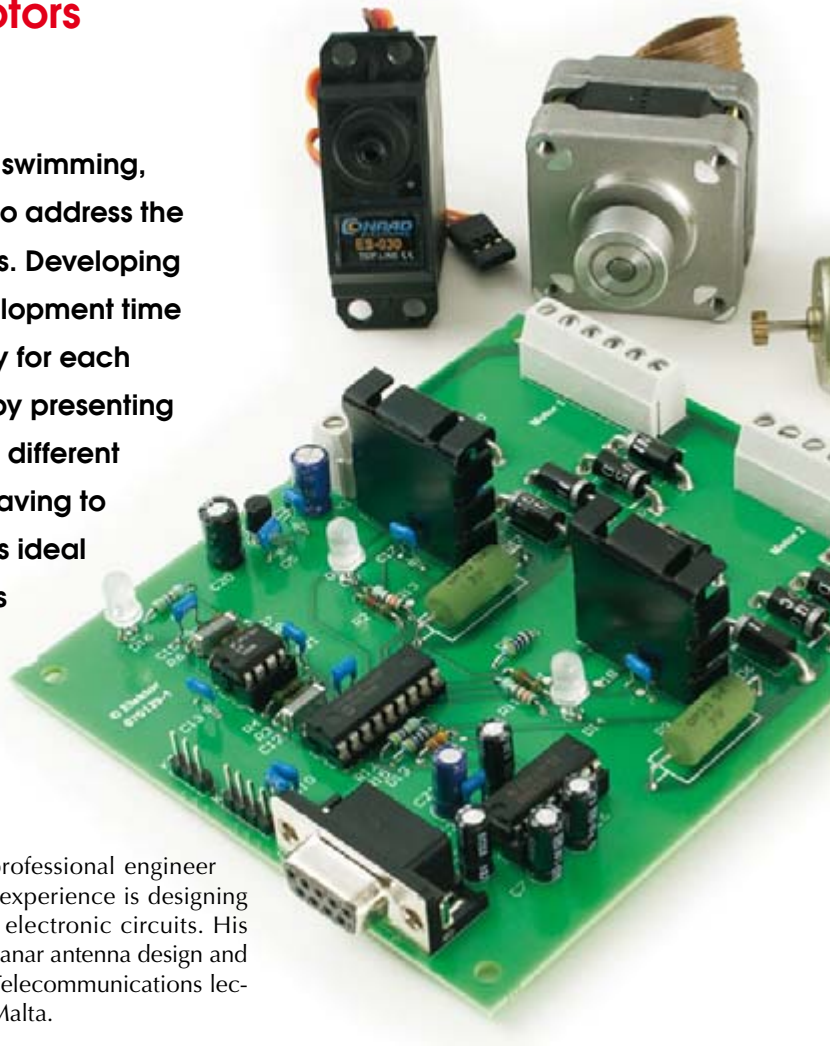
Before a robot moves, no matter if that's walking, swimming, cruising a maze or tearing up asphalt, you need to address the interfacing and driving of different types of motors. Developing the relevant algorithms will increase project development time as tweaks and adaptations seem to be necessary for each new application. This project solves these issues by presenting a generic motor driver board that can drive three different types of motors, stepper, DC and servo, without having to program the algorithm for each motor. MotoBox is ideal for newcomers to robotics as well as for old hands who can use the board for prototyping and development. The board can also be used with two different types of motors simultaneously!

The board is programmed by way of an RS232 interface. This is a common protocol found in many microcontrollers. It can be driven from the serial port of a computer or via another microcontroller. The script used to control the motors is a very easy 6-character command set. It has been kept this way in order to have a very user-friendly interface. The script also allows the state of the board to be read back to the user. Additionally the board monitors the current taken by each motor driver and switches off the driver if a short-circuit is detected.

The author is a professional engineer with many years experience is designing and constructing electronic circuits. His specialisation is planar antenna design and currently he is a Telecommunications lecturer at MCAST, Malta.

### Circuit description

Referring to the schematic in **Figure 1**, the heart of the circuit is a PIC16F628A which controls and monitors the motor drivers. The microcontroller is configured to use its internal 4 MHz oscillator, thus freeing the pins associated with the crystal to be used as the serial interface. PORTB is used as an output to drive the motor controllers, while PORTA is used for various functions, to read the value of the current, control the drivers and control the driver LED's. The MCLR pin is held High by means of R15 and D13. An ICSP programming connector is also included in order to program the device in place. The PGC and PGD lines are decoupled by a 100nF capacitor.



### MotoBox Specifications

- Drives 3 types of motors:
  - bipolar stepper motors
  - DC motors
  - servo motors
- Motor actuated using serial port @ 9600,N,1
- Up to 13.8V supply for motors
- Easy to use command set
- Current monitoring, up to 2 A (DC and servo motor) or 4 A (stepper motor)
- LED indicator: green 'ok', red 'error'
- Individual control of each motor driver bridge

The module is powered by a 12-V supply which is regulated down to 5 V by means of an 78L05. Power-On status is shown by the green LED D16. The module has been designed to operate from a maximum of 13.8 V from a freshly charged lead-acid battery and down to 7 V. It is important for the module not to share the power source used to drive the motor because of the current surges, spikes and noise induced by the motor.

A MAX232 is used to interface the module to the serial port of a desktop computer or laptop. It is imperative that the MAX232 be included in the circuit or else make sure that RA7 (receive pin) is held Low when not used. A floating input will cause the microcontroller to interpret noise as data and may result in erroneous operation. The module can also be controlled by another microcontroller by using connector K2. The motors are driven by the power stages provided by the L298, each driving a single



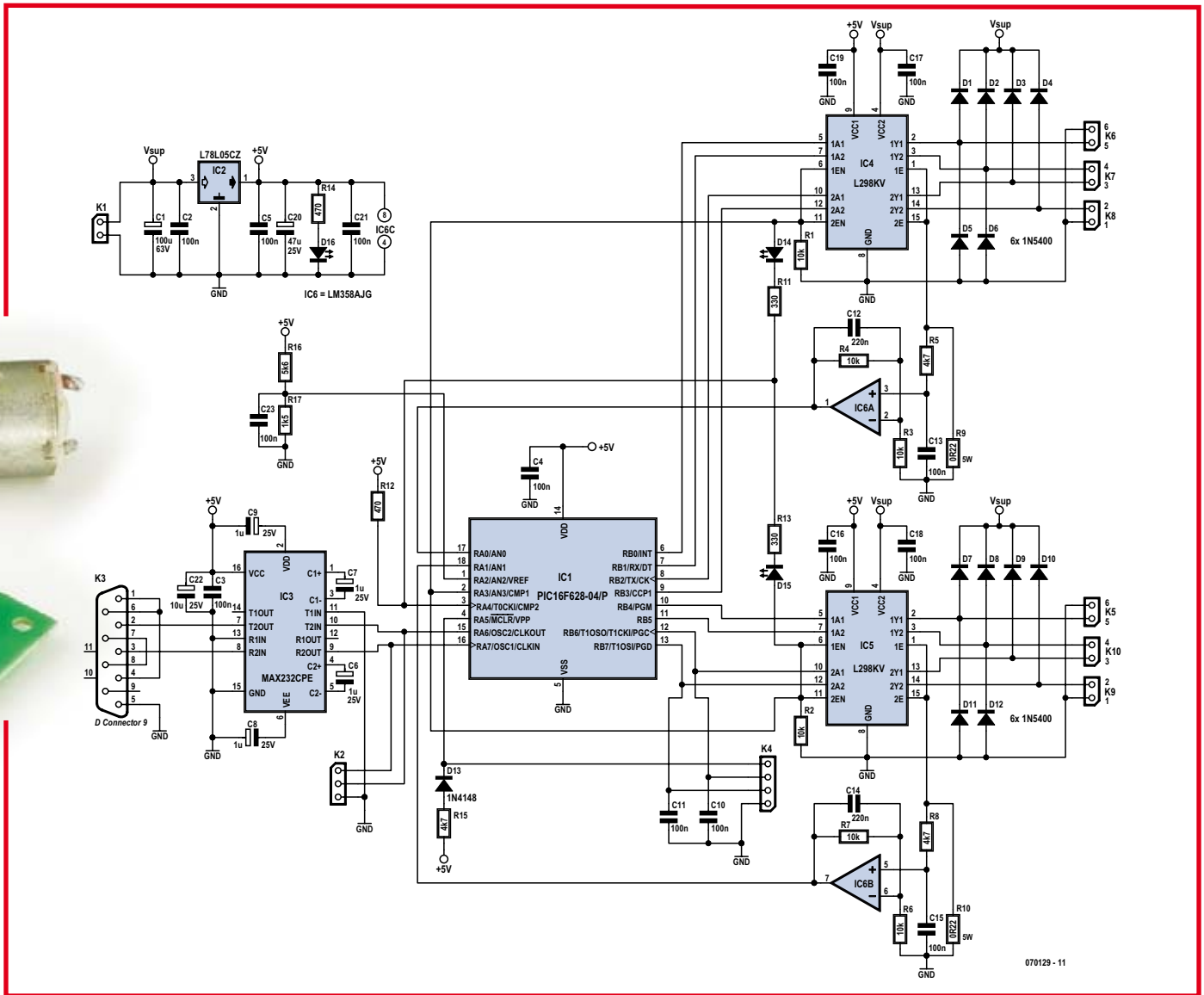


Figure 1. Circuit diagram of Motobox, the 'one size fits all' controller for motors typically used in robots.

motor. The L298 is a dual half-bridge driver capable of 2 A per channel, with separate enabling and current sensing for each bridge. It is suited to driving different types of motors. In this project the L298 is enabled by the microcontroller and the Enable line is pulled Low on power-up to ensure that the IC is disabled. Decoupling capacitors, 100 nF each are placed near the IC. A DC amplifier is built around the LM358 which senses the voltage across the sense resistor. It is followed by a lowpass filter and a buffer amplifier whose gain decreases with frequency filters any noise. DC motors generate spikes when the armature makes contact with the commutator, which is reflected as a current spike and appears across the sense resistor. Current spikes may cause the microcontroller to detect a short-circuit and thus stop the motor. The filter will smooth out any voltage spikes. In addition, an algorithm is implemented in the microcontroller firmware that detects an overcurrent condition

for a specified time before switching off the affected channel.

### Construction

Good news: the project does not have

exotic components to source, with the possible exception of the clip-on heatsink — but then that's available from Mouser. Through-hole components are used in the project to simplify construction and all components are easy to obtain as well as

Table 1. Driver options chart								
	Motor header function							
	Output							
	Motor 1				Motor 2			
µC Pin	RB3	RB2	RB1	RB0	RB7	RB6	RB5	RB4
Header pin	2	3	4	5	2	3	4	5
Mode								
Stepper driver	Each output drives a motor phase				Each output drives a motor phase			
DC motor driver (1)	✓	✓			✓	✓		
Servo driver (2)	✓(3)	P(4)	P(4)	✓(3)	✓(3)	P(4)	P(4)	✓(3)

(1) Differential drive  
 (2) Use the ground pin on the connectors for the servo  
 (3) Allows 2 servos to driven simultaneously with the same signal  
 (4) Used to power servo motor

mount on the PCB, of which the component overlay is shown in **Figure 2**. There's little to say on the components used except that the 0.22-Ω resistor is preferably not of the wirewound type and the bi-colour LEDs have two pins, not three! If you do not envisage interfacing the board to a PC, the sub-D connector may be left out. However, the MAX232 should always be present. Likewise, IC4 and IC5 should always be soldered to the PCB.

### Dry testing

Once the circuit has been constructed it should be tested for correct operation. Power up the circuit with none of the ICs inserted. D16 should light up. Check if there is 5 V on the relevant socket pins for IC1, IC3, IC4, IC5 and IC6. The supply voltage should also be present on IC4 and IC5, socket pin 4. If this is correct, switch off and insert the programmed microcontroller and the rest of the ICs. The microcontroller can be programmed in-circuit using connector K4.

If the board is to be controlled from another microcontroller, connector K2 may be used. This is a straight through connector and the Transmit/Receive signals must be crossed before being connected to the board. If the female sub-D connector is going to be used, the signals are crossed over on the board itself.

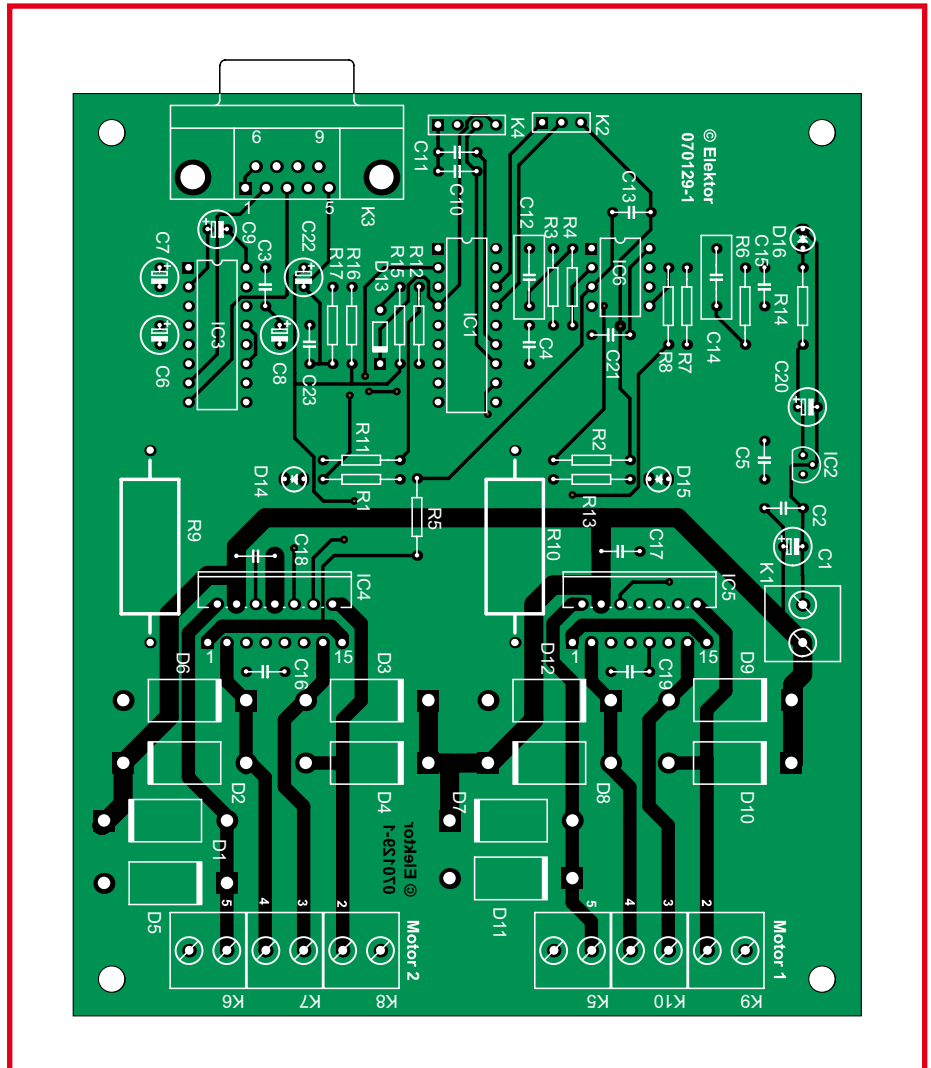
### Motors get connected...

The best way to fully test the board is to connect it to the PC and use HyperTerminal to send commands. Attach a motor according to one of the configurations shown in **Figure 3** in order to test the board. Use the motor driver options chart, **Table 1**, to connect the motor to the correct pins.

### ... and controlled

As soon as the board is powered up correctly, it will return the character 'I' to the Host to signal that it's been initialised and is waiting for a command. All commands are six characters long, you can find them listed in **Table 2**.

Let's assume a DC motor has been connected to the board. For the motor to turn in the forward direction the command 'DC1FOR' has to be typed in lowercase or uppercase. The software will convert all letters to uppercase. When the command has been sent it will be executed, the character 'A' indicating that reception is acknowledged and the actual command is echoed. This is very useful when the board is used with a (dumb) terminal. If the command has been sent incorrectly, the letter 'E' will be sent back indicating

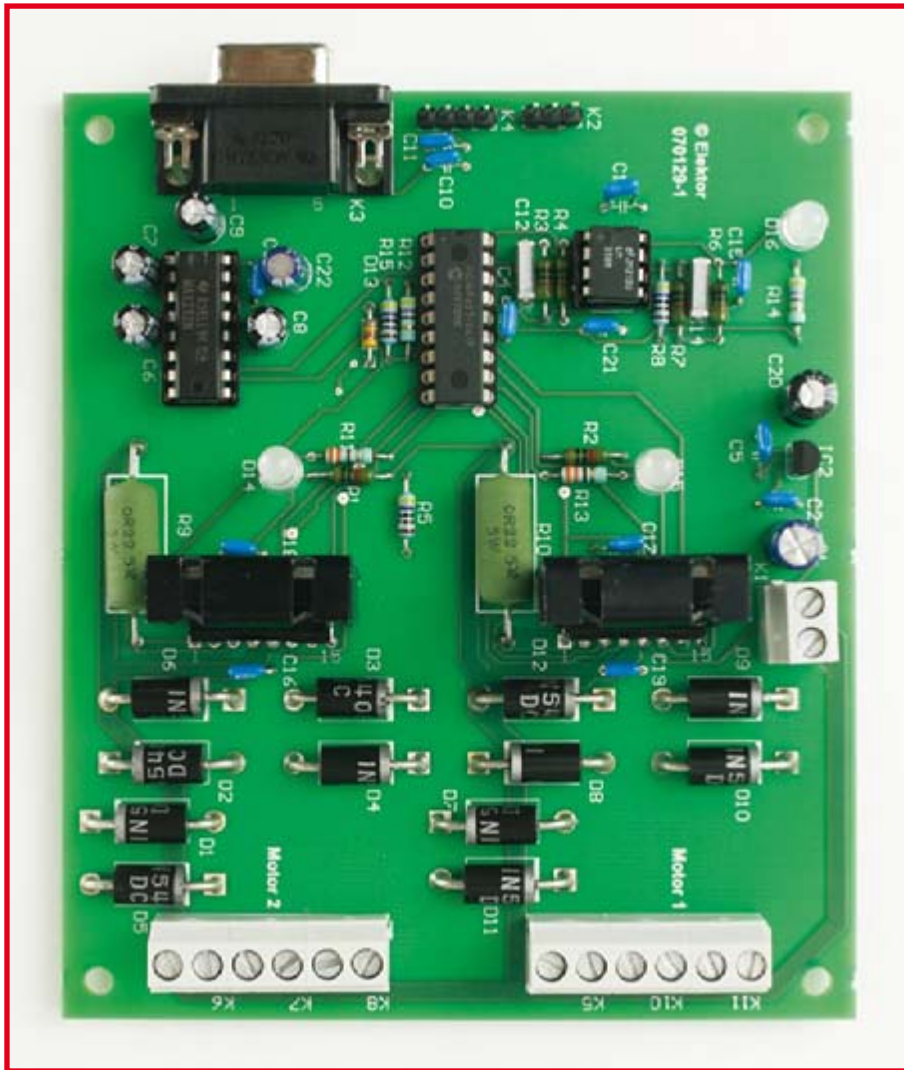


**Figure 2.** Component mounting plan of the PCB designed for Motobox. The copper track layout is a free download from the Elektor website.

an erroneous command. The command buffer is cleared and the board waits for a new command. The LEDs will turn light

up when the system is working correctly and red when the motor driver has been turned off.

Table 2. Command syntax		
	Command	Command Description
1	ST1FOR	Stepper 1 cw
2	ST1BAK	Stepper 1 ccw
3	ST2FOR	Stepper 2 cw
4	ST2BAK	Stepper 2 ccw
5	STBFOR	Both steppers cw — simultaneously
6	STBBAK	Both steppers ccw — simultaneously
7	STSYNC	Get both steppers in same position in case of slip
8	AMSTOP	Emergency stop. Stop All motors — e.g. overcurrent
9	ST1FXX	Turn stepper 1 xx (999 steps max) positions cw
10	ST1BXX	Turn stepper 1 xx (999 steps max) positions ccw
11	ST2FXX	Turn stepper 2 xx (999 steps max) positions cw
12	ST2BXX	Turn stepper 2 xx (999 steps max) positions ccw
13	DC2FOR	DC motor 2 ON/cw
14	DC2BAK	DC motor 2 ON/ccw
15	DC2OFF	DC motor 2 OFF
16	DC1FOR	DC motor 1 ON/cw
17	DC1BAK	DC motor 1 ON/ccw



Stepper motor commands allow the spindle to be turned cw (clockwise) and ccw (counter clockwise) by a number of steps

(maximum 999 steps). When this command is selected, a 3-digit number must be provided, for example 050 to move 50 steps.

18	DC1OFF	DC motor 1 OFF
19	DCBFOR	DC both motors cw
20	DCBBAK	DC both motors ccw
21	DCBOFF	DC both motor OFF
22	SV2FOR	Servo 1 cw
23	SV2BAK	Servo 1 ccw
24	SV2MID	Servo 1 centre position
25	SV2HLD	Servo 1 hold
26	SV1FOR	Servo 2 cw
27	SV1BAK	Servo 2 ccw
28	SV1MID	Servo 2 centre position
29	SV1HLD	Servo 2 hold
30	MTYPEX	Return motor type
31	STDELX	Configure stepper motor delay
32	ECHOST	Configure echo. Default = ON
33	AMONXX	All motors ON
34	MONOFF	Do not monitor current consumption
35	CMONON	Monitor current consumption
36	RESUME	Enable Motors after overcurrent condition

Even if steps less than 100 will be done the leading zero must be provided.

The DC, servo motor and status commands are executed immediately, but the stepper motor commands are subject to a delay (default 20 ms) so the command will be acknowledged after the stepper motor has been actuated. Sending a command within this interval will cause it to get lost and will be not executed. The type of motor in use can be read back at any time using the command 'MYPEX', which will return a character depending on the type of motor being used.

### Overloading, spikes, surges...

Through the analogue comparator, the software section will continuously monitor the current consumed. The board will switch off the motors if the current consumption exceeds the specifications for the motor type in use.

Simple filtering takes place so that the controller will not switch off the motors at 'just any' current spike. Rapid on and

## COMPONENTS LIST

### Resistors

- R1-R4,R6,R7 = 10k $\Omega$
- R5,R8,R15 = 4k $\Omega$
- R9,R10 = 0.22 $\Omega$  5W
- R11,R13 = 330 $\Omega$
- R12,R14 = 470 $\Omega$
- R16 = 5k $\Omega$
- R17 = 1k $\Omega$

### Capacitors

- C1 = 100 $\mu$ F 40V radial
- C2-C5,C10,C11,C13,C15-C19,C21 = 100nF
- C6-C9 = 1 $\mu$ F 25V radial
- C12,C14 = 220nF
- C20 = 47 $\mu$ F 25V radial
- C22 = 10 $\mu$ F 25V radial
- C23 = 100nF

### Semiconductors

- D1-D12 = 1N5400
- D13 = 1N4148
- D14,D15 = bicolour LED, 5mm, 2 terminals
- D16 = LED, 5mm
- IC1 = PIC16F628-04/P, programmed, order code 070129-41
- IC2 = 78L05
- IC3 = MAX232
- IC4,IC5 = L298N with heatsink for Multiwatt15 case (mouser.com)
- IC6 = LM358
- K1,K5-K10 = 2-way PCB terminal block, lead pitch 5mm
- K2 = 3-way SIL pinheader
- K3 = 9-way sub-D socket (female), angled pins, PCB mount
- K4 = 4-way SIL pinheader
- PCB, ref. 070129-1
- PIC source & hex code files, free download # 070129-11.zip from Elektor website.



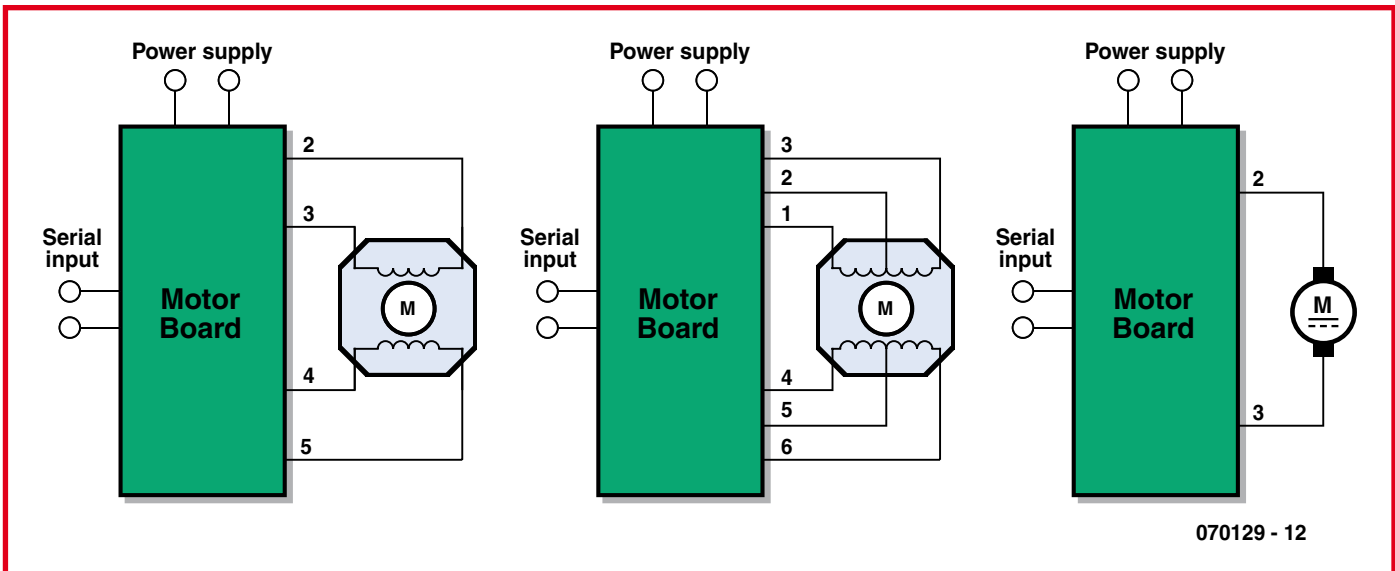


Figure 3. How to connect up your motors.

off switching of DC motors will cause large current spikes, which may cause the controller to shut down the particular motor driver. You can disable the current monitoring but this is not recommended. Overcurrent can cause the L298 to overheat and get destroyed. A 'RESUME' command has to be sent after the motor driver has been switched off to resume motor operations. A grace period is given to enable another command to

be sent and try to get the motor unstuck from its current position.

### Software

The source code and hex code for the PIC16F628A is available as a free download from the Elektor Electronics website. The file number is **070129-11.zip** (July/August 2007).

(070129-1)

### Author's websites and email address

<http://telecomms.no-ip.org>

[www.mcast.edu.mt](http://www.mcast.edu.mt)

[jozamm@gmail.com](mailto:jozamm@gmail.com)

### Web links

[www.microchip.com](http://www.microchip.com)

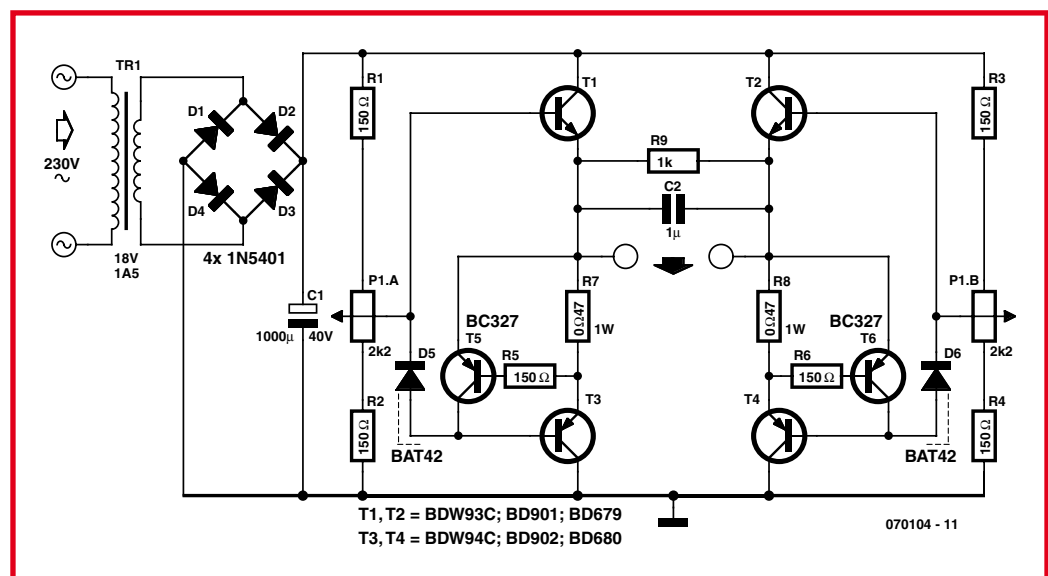
[www.st.com/stonline/products/literature/ds/1773.pdf](http://www.st.com/stonline/products/literature/ds/1773.pdf)

# 12 V Bidirectional Motor Control

Stefan Brandstetter

This simple circuit drives DC motors with a maximum current of 1 A and can be built with readily available components. The output voltage is adjustable between 0 and 14 V and the polarity can be changed so that not only motor speed but also rotation direction can be adjusted by turning a knob. The circuit is also ideal as a controller for a DC model railway or small low-voltage hobby tool.

Power for the circuit is supplied by a 18 V mains transformer rated at 1.5 A. Diodes D1 to D4 rectify the supply and capacitor C1 provides smoothing to give a DC output voltage of around 24 V. A classic 'H' bridge configuration is made up with transistors



T1/T3 and T2/T4. Transistors T5 and T6 together with resistors R7 and R8 provide

the current sense and limiting mechanism. The maximum output current limit can be

changed from 1 A by using different value resistors for R7 and R8:

$$I_{OUT} = 0.6 V / R$$

where R gives the value for R7 and R8. For increased current limit the mains transformer and diodes will need to be changed to cope with the extra current as well as the four transistors used in the bridge confi-

guration. Motor speed control and direction is controlled by a twin-ganged linear pot (P1). The two tracks of P1 together with R1/R2 and R3/R4 form two adjustable potential divider networks. Wiring to the track ends are reversed so that as the pot is turned the output voltage of one potential divider increases while the other decreases and vice versa. In the midway position both dividers are at the same vol-

tage so there is no potential difference and the motor is stationary. As the pot is rotated the potential difference across the motor increases and it runs faster. The voltage drop across D5 and D6 is equal to the forward voltage drop VBE of the bridge transistors and ensures that the motor does not oscillate in the off position with the pot at its mid point.

(070104-1)

# zBot: 10-A Power Stage for DC Motor



Jens Altenburg

If you look at the chassis of the zBot vehicle<sup>1</sup>, you'll find two parts requiring intelligent control: the steering servo and the DC motor.

The so called H-bridge is the normal circuit for electronic control of revolution speed and direction. The DC motor of a Tamiya car is powerful enough to propel zBot at up to 20 miles per hour. The motor then consumes more than 10 A, so we choose high-current power MOSFETs for the driver stage. There are lots of different devices to choose from.

The MOSFET we require has to supply the maximum motor current and, importantly, it has to be switched with gate voltages of about 5 V. In this case, the microcontroller switches the power stage ('low side') directly. For high side driving level shifters are necessary.

The schematic of the H-bridge power stage shows a few inverters, NAND gates and two tri-stateable drivers. These logic functions are very important as the easier way, i.e., directly controlling all four MOSFET has a fatal disadvantage.

In case of a software crash it could happen that two or more MOSFETs are switched on incor-

rectly,

for example, T4 and T7. In that case, the current through the transistors is limited by the internal resistors of the MOSFETs

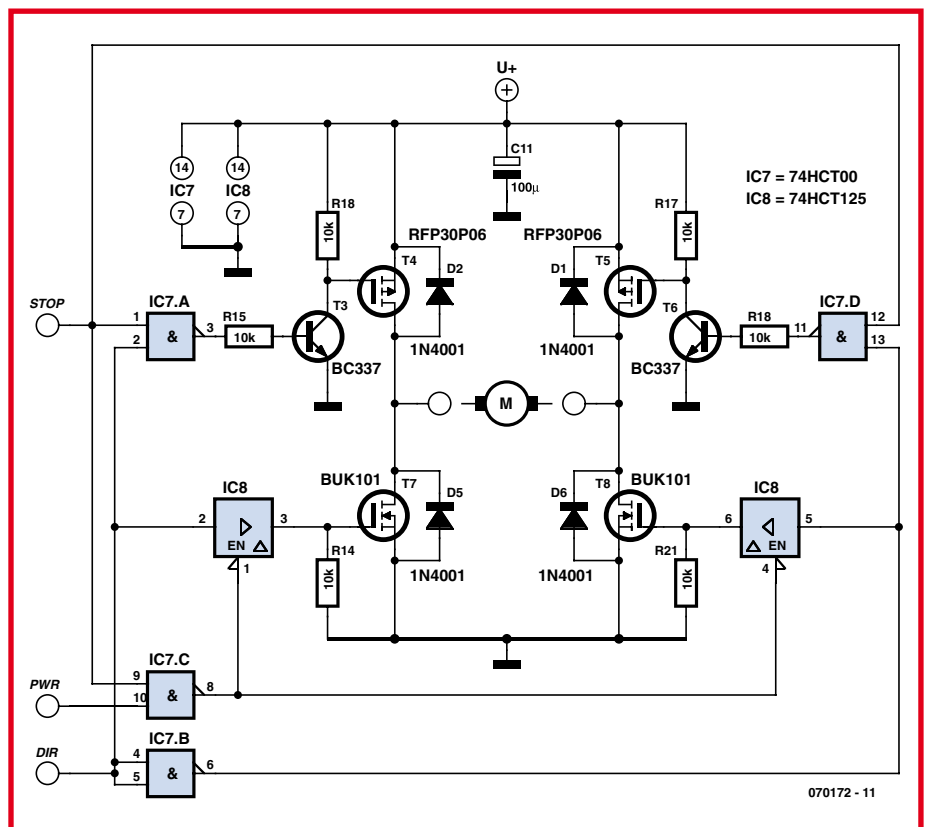
(about 10 mΩ) only. Such a fatal error would destroy the MOSFETs. The logic functions configured here effectively avoid illegal states.

To control the DC motor, three signals are needed: DIR, PWM and STOP. DIR controls the direction of the motor revolution, PWM the speed, and STOP brakes the motor.

The software module for the DC motor is called dcm.c.

(070172-1)

(1) The complete document called Zbot — the Robot Experimental Platform is available for free downloading from the Elektor Electronics website. The file number is 070172-11.zip (July/August 2007).



# Complete Stepper Motor Driver

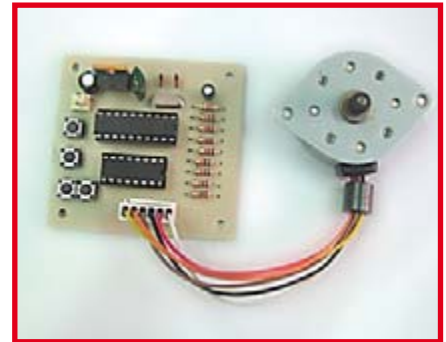
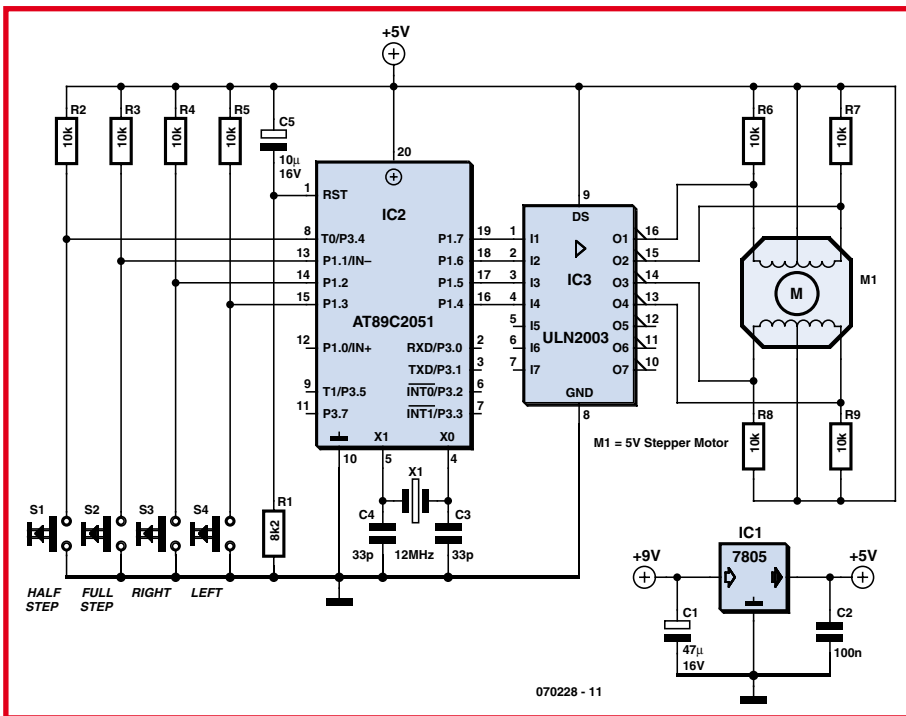


Table 1

Step angle (degrees)	Steps per revolution
0.72	500
1.8	200
2.0	180
2.5	144
5.0	72
7.5	48
15	24

Hesam Moshiri

With this circuit you can make a stepper motor do just about anything it will need to do in robotics application: rotation to the left or right, in full-step or half-step mode.

Stepper motors convert electrical pulses into mechanical movement. In applications like hard disks, printers and photocopiers (to mention but a few), stepper motors are used for rotation and/or accurate position control of mechanical assem-

blies. Every stepper motor has one permanently magnetic axle called the *rotor*. This is surrounded by a fixed part called the *stator*. Usually, stepper motors have four stator wires with two or one common wire, which is normally connected to the positive supply voltage.

By applying a controlled sequence of pulses to the individual stator windings, the rotor will start to rotate. Stepper motors may differ in size, shape, power, supply voltage, cost, accuracy, and so on, but

importantly in the number of steps that make up one complete spindle revolution. This property also determined the step angle as shown in **Table 1**.

For example for a motor specified as having a 1.8-degree angle,  $360 / 1.8 = 200$  pulses for a complete spindle revolution.

Two pulsing schemes are available to drive the motor: 'full-step' or 'half-step'. The two modes are summarized in **Table 2** and **Table 3** respectively.

Applying half-step pulses to the motor will increase the accuracy at which the spin-

Table 2. Full-step mode.

Rotation to the right (cw)	Step	Winding A	Winding B	Winding C	Winding D	Rotation to the left (ccw)
↓	1	1	0	0	0	↑
	2	0	1	0	0	
	3	0	0	1	0	
	4	0	0	0	1	

Table 3. Half-step mode.

Rotation to the right (cw)	Step	Winding A	Winding B	Winding C	Winding D	Rotation to the left (ccw)
↓	1	1	0	0	0	↑
	2	1	1	0	0	
	3	0	1	0	0	
	4	0	1	1	0	
	5	0	0	1	0	
	6	0	0	1	1	
	7	0	0	0	1	
	8	1	0	0	1	



dle can be turned. In the case of our 1.8-degree angle motor, half-step driving then requires 400 steps per revolution. Another important advantage of half-step pulsing is more motor power, which usually translates in more torque. The circuit of the motor driver is designed around an Atmel microcontroller type AT89C2051 ticking at 12 MHz and one high voltage/high current Darlingon transistor array type ULN2003. The motor drive pulses generated by the microcontroller under firmware control are fed to the ULN2003 via four port lines P1.4 through P1.7. The motor's stator windings are connected to the corresponding output

pins on the ULN2003. The ULN2003 can supply up to 500 mA on each output pin. Note that a 5-V stepper motor is used in this circuit. The source code file and the firmware (hex file) for the AT89 micro may be downloaded free of charge from the Elektor website as archive # **070228-11.zip**. After constructing your circuit, power it up. Press the Full Step or Half Step button. Then press Left or Right and you will see your motor start to rotate using the mode selected. You can change between full and half step at any time. All this is based on the assumption that you have wired up your motor correctly.

The AT89 source code contains a number of directions to help you 'change wires' in software rather than by soldering and getting confused by the different wire colours. In practice, you will notice that full-step mode yields higher spindle speed with low motor torque, whereas half-step mode is good for increased torque and accuracy, at the cost of speed. That is why stepper motors powering wheels etc. are controlled such that they start and end their operation in half-step mode, with full step mode in between to achieve maximum speed.

(070228-1)

# Controlling Servos

## Using a PIC programmed in Basic, a Basic Stamp or a Cubloc

C. Tavernier

Through robotics, radio-control servos are currently experiencing a new lease of life, thanks to their characteristics, which although not originally designed for such applications, turn out in fact to be well suited to it. Current radio-control servos are very compact, bearing in mind they contain not only their own mechanism, but also dedicated drive electronics, which only need simple TTL or CMOS logic signals as an input. The power they are able to supply can be quite considerable, for the most powerful of them (originally intended for 'large' model planes or boats); and lastly, they are usually supplied with a host of accessories such as crank arms, perforated wheels, etc., making it easier to interface them with the elements to be operated. There are currently two fundamental ways of using a servo in a robot. The first, described elsewhere in this issue, consists of converting the servo into a propulsion motor, which admittedly is rather taking it away from its original function. The second, which we're going to be looking at here, involves its use for positioning. Whether in an arm, or to turn a platform carrying a camera, a telemeter, or any other unit, our servo is ideal for this.

We won't insult you by telling you what a servo is like, since even if you aren't a radio-control enthusiast, you're bound to

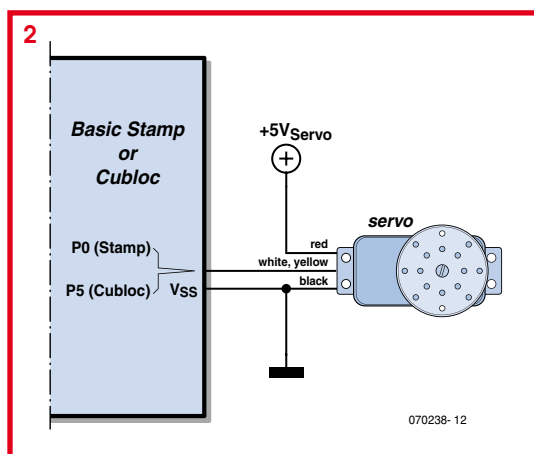
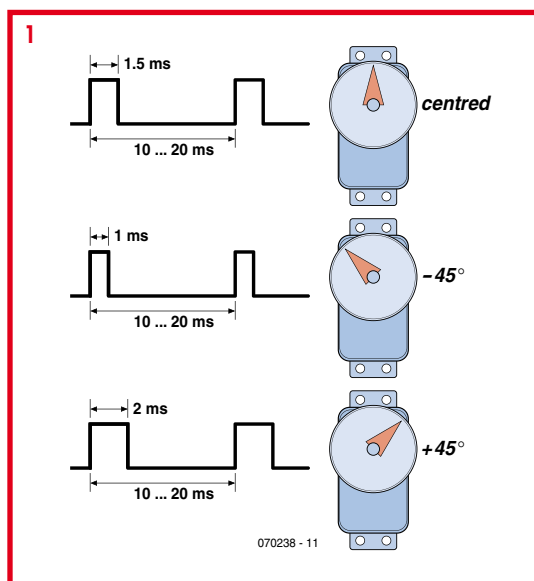
have come across them before. However, here are just a few pieces of information that it's important to be aware of so as to be able to make use of them.

In electrical terms, a servo has just three colour-coded wires. The red and black wires are for powering it, at between 4.8 and 6 V. The third wire, yellow or white (or in practice any colour other than red or black), is used to convey commands to the servo in the form of pulse-width-modulated (PWM) signals.

**Figure 1** illustrates both the coding principle of these pulses and the effect they have on the position of the servo. Note first of all that they must be repeated at such a rate that there is not more than 10–20 ms separation between two successive pulses. In theory, this repetition is not absolutely vital; but with it, the servo will be able to maintain the position set by the width of the pulses received. If the pulses do not repeat, the servo will indeed go to the position dictated by the last pulse received, but, as soon as that stops, the slightest force on its shaft will cause it to lose the position attained.

Notice from the figure:

- a 1.5-ms pulse places the servo in its position referred to as centred or rest;
- a 1-ms pulse makes the servo turn to its maximum anti-clockwise position, which usually represents an angle of 45° with respect to its rest position;
- a 2-ms pulse makes the servo turn to its maximum clockwise position, which too usually represents an angle of 45° with respect to its rest position.



Intermediate positions can be obtained by varying the pulse width between 1 ms and 2 ms. For use in robotics, it is even possible to go further and apply pulses a little shorter than 1 ms or a little longer than 2 ms to the servo, thereby achieving a total angle of rotation of 180°. But watch out! At this point, we have gone outside the specifications for the servo, which is in danger of jamming in these extreme positions, destroying its motor, its electronics — or if you're lucky, both at once!

Connecting a servo to a microcontroller (Basic Stamp, PIC, Cubloc) is very simple, as shown in **Figure 2**. The only point to watch out for is the servo power supply. Given the relatively high current drawn by the servo when it turns, it's best to supply it off a separate voltage rail from the microcontroller. Where this is not possible, you need to ensure excellent decoupling between them — for example, by supplying the servo and the microcontroller via two separate regulators.

As far as the software is concerned, controlling a servo using a Basic Stamp or a PIC programmed in Basic requires only a very few lines of program. In fact, all that's needed is to call up the two instructions PULSOUT and PAUSE.

Here, by way of an example, is a program that makes a servo turn slowly from its extreme position on one side to its extreme position on the other side, and so on. In

#### Listing 1

```

` Basic Stamp I and II or in
Basic programmed PIC version

loop:
    for b2 = 100 to 200
        for b3 = 1 to 5
            pulsout
0,b2 ` servo connected to port P0
            pause 15
        next
    next
goto loop

```

#### Listing 2

```

` Cubloc Version

Const Device = CB220
Dim Position As Integer
Low 5
` Servo is connected to port P5
Do
    For Position = 2300 to
Position = 4300 Step 20
        Pwm 0, Position, 32768
        Delay 100
    Next
Loop

```

this way it can, for example, move a sensor in one plane over a total amplitude of 90°, see **Listing 1**.

The instruction PULSOUT generates pulses

with a variable duration from  $100 \times 10 \mu\text{s}$  to  $200 \times 10 \mu\text{s}$  depending on the changing value of the loop variable b2, while the spacing between these pulses is set at 15 ms by means of the instruction PAUSE. This program is written here in Basic Stamp I language, but transposing it for Basic Stamp II or for use with a PIC programmed in Basic only requires modification of the end values and the loop variable b2 increment. The resolution of the instruction PULSOUT is now  $2 \mu\text{s}$  instead of  $10 \mu\text{s}$ , so the various values need to be multiplied by 5.

Using a servo with a Cubloc from Comfile Technology is just as simple, but uses an instruction called PWM, as per **Listing 2**. In this type of application, the advantage of the Cubloc over the Basic Stamp is that the PWM command generates the pulses indefinitely, even if the program continues on to something else. In the case of the Basic Stamp, the instruction PULSOUT generates only one pulse, and so has to be called from a loop in order to produce them continuously, preventing the Basic Stamp from doing anything else. If you don't want to use a Cubloc, another solution consists in using a specialized IC, like the MIC 800 from Micronics ([www.micronics.com](http://www.micronics.com)), which can control up to 8 servos simultaneously in a stand-alone manner (if necessary, refer to the Elektor Summer Circuits edition 2006).

(070238-1)

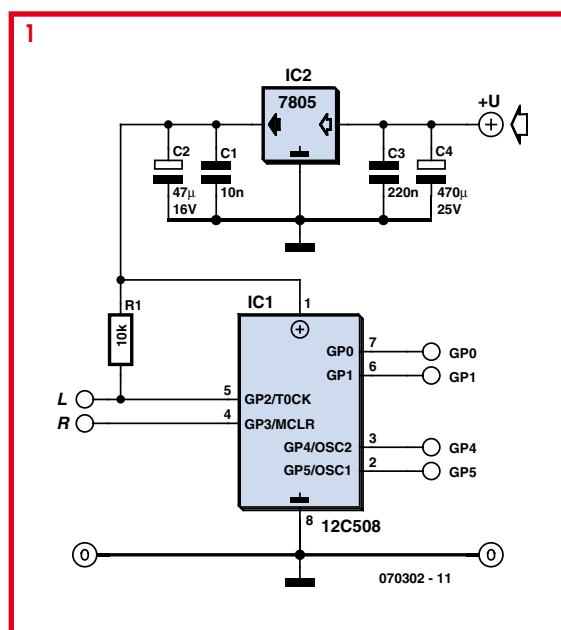
# PIC12C508 Stepper Motor Controller

C. Tavernier

[www.tavernier-c.com](http://www.tavernier-c.com)

When we're not using a stepper motor to ensure precise positioning of a robot element, it can be used as a traction motor, in place of the standard modified servos presented elsewhere in this issue. Under these conditions, there's no longer any need to 'count the steps' the motor has to make, as all we want is to make it rotate continuously in one direction or the other.

Several solutions are open to us for driving the motor, a number of which are presented in this issue: using a specialized stepper motor driver IC, using one or more suitably-programmed microcontroller parallel ports, or building a driver based around conventional logic ICs.



However, these solutions are far from satisfactory when using a stepper motor for traction. They all require pulses to be generated continuously for as long as we want the motor to run, either requiring an additional programmable oscillator, or using up resources from the robot's main microcontroller.

So we've decided to suggest another approach with this stepper motor driver specifically designed for making the motor turn in one direction or another, under the control of a simple logic level. And as the propulsion motors in robots usually go in pairs, we're even going to offer a dual driver, by diverting a very common and inexpensive IC from its original function.

Since a stepper motor used for propulsion doesn't need to be accurate

in terms of positioning, and hence, in the precision of the steps, simple single-pole models are eminently suitable. So, our circuit is designed for motors of this type.

This lets us control the motor via two TTL- or CMOS-compatible logic inputs. When these two inputs, labelled L and R, are logic high or floating (they have their own pull-up resistors), the motor stays still, but in braked mode, since it's a stepper motor. When the L input is taken to logic low, the motor rotates in one direction (arbitrarily, to the left, whence the label L) while if the R input is taken low, it turns the other way. If both inputs are taken to ground at the same time, the R input has priority, and so the motor turns in that direction.

The motor's speed of rotation is fixed, but, since we are giving you the source listing of the software used for this application, it's very easy for you to modify this if it doesn't suit you, or indeed even to include the possibility of external adjustment if necessary.

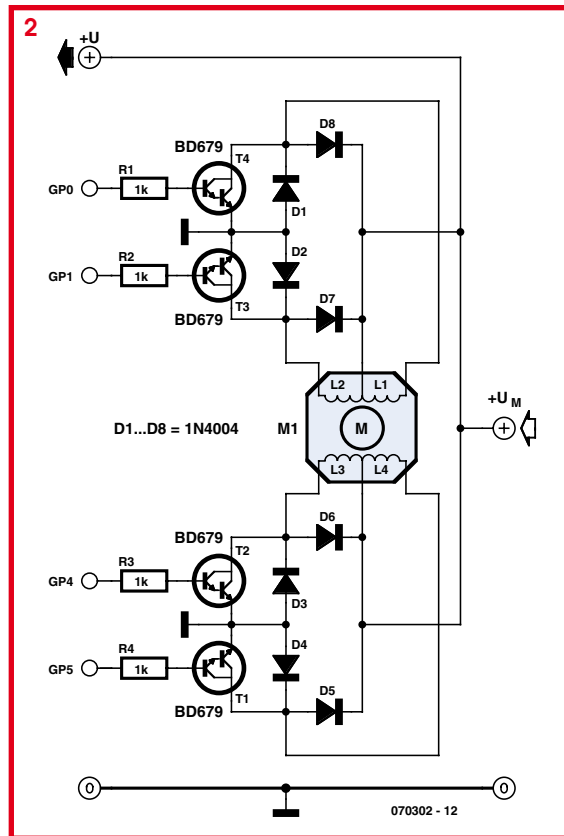
The circuit of the 'intelligent' part of our controller is shown in **Figure 1**, as you can see it uses a PIC12C508 microcontroller from Microchip. Used here in internal clock and reset circuit mode, it needs no external components for these functions, so all its port lines are available.

Parallel ports GP2 and GP3 are used as inputs, and as GP2 does not have an internal pull-up resistor, this is performed by R1. Parallel ports GP0, GP1, GP4, and GP5 are used as outputs for generating the pulses for the motor windings. These can be amplified by two types of power stages, depending on the type and number of motors to be driven; we'll take a look at those circuits in a moment.

The 12C508 needs to be powered from 5 V, derived from the motor supply by means of a conventional 3-terminal voltage regulator IC2.

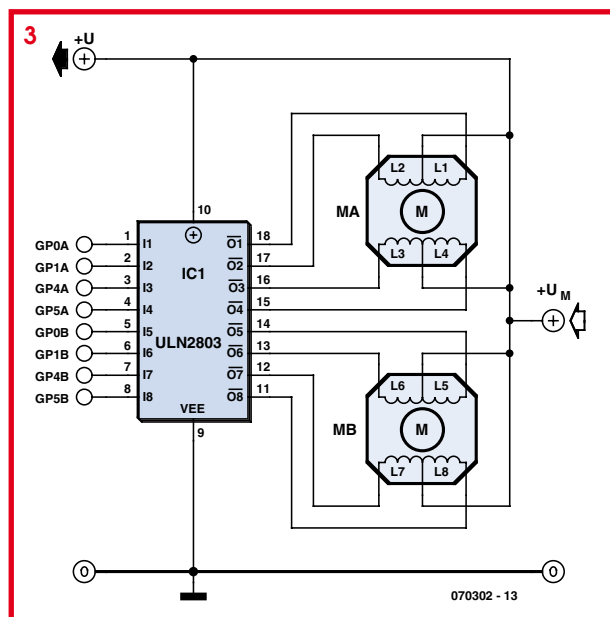
If the controller is only intended for a single motor, or if the motor to be driven draws more than 500 mA per winding, the power stage shown in **Figure 2** can be used. It employs conventional bipolar transistors that, given their characteristics, are able to switch currents of 3 A. Diodes D1–D8 clip the spurious spikes generated by the abrupt switching of the current in the motor windings and protect the transistors.

However, if the motor used draws less than 500 mA, and more importantly, if you need to drive two motors of this type, an elegant and ingenious solution exists, as



**Table 1. Programming step duration by modifying a constant used in the program.**

Binary Step constant	duration
10010010	1 ms
10010011	2 ms
10010100	4 ms
10010101	8 ms
10010110	16 ms
10010111	32 ms



shown in **Figure 3**. This uses a standard ULN2803, usually used to drive relays, but which includes eight medium-power Darlington transistors along with their protection diodes. So, this IC is able to properly drive any kind of single-pole stepper motor, as long as the voltage required doesn't exceed 50 V and the current per winding is under 500 mA.

In addition, as the ULN2803 contains eight identical stages, it can be preceded by two controllers like the one in Figure 1 and in this way drive two robot propulsion motors: one on the left and one on the right, marked MA and MB in this figure.

Constructing one or other of these versions is very straightforward. The PIC 12C508 needs to be programmed with the file that you'll find in object form, as well as in source form, in case you'd like to modify it, on the Elektor website, as well as on the author's own site ([www.tavernier-c.com](http://www.tavernier-c.com)).

If you build the transistor power amplifier, note that T1–T4 don't need a heatsink as long as the motor consumption doesn't exceed 1 A. Other-

wise, bolt them onto a small aluminium plate a few cm<sup>2</sup>. To simplify mechanical construction, it can be common to the four transistors, but in this case you'll need to use the standard insulating accessories of mica washers and shouldered washers, as the collectors of these transistors are connected to the metal parts of their cases.

If you construct the ULN2803-based version, there are no special precautions to be observed, other than to not exceed the IC's maximum current capacity of 500 mA.

As we are providing you with the full source listing of the software programmed into the 12C508, you'll be able to modify it to suit your needs. If you are unfamiliar with PIC microcontroller assembler, here are the details you'll need for the most important modification you might want to make: changing the speed of the control pulses to the motors, and thus, their speed of rotation. The control word may be found in **Table 1**.

To do this, all you have to do is modify the binary constant on the line:

```
MOVLW B'10010101'
```

```
MOVLW B'10010101'
```

just above the line containing OPTION in the source listing. With the original value, the duration of one step is 8 ms, but the table above indicates what constant to use according to the step duration that you may want.

(070302-1)



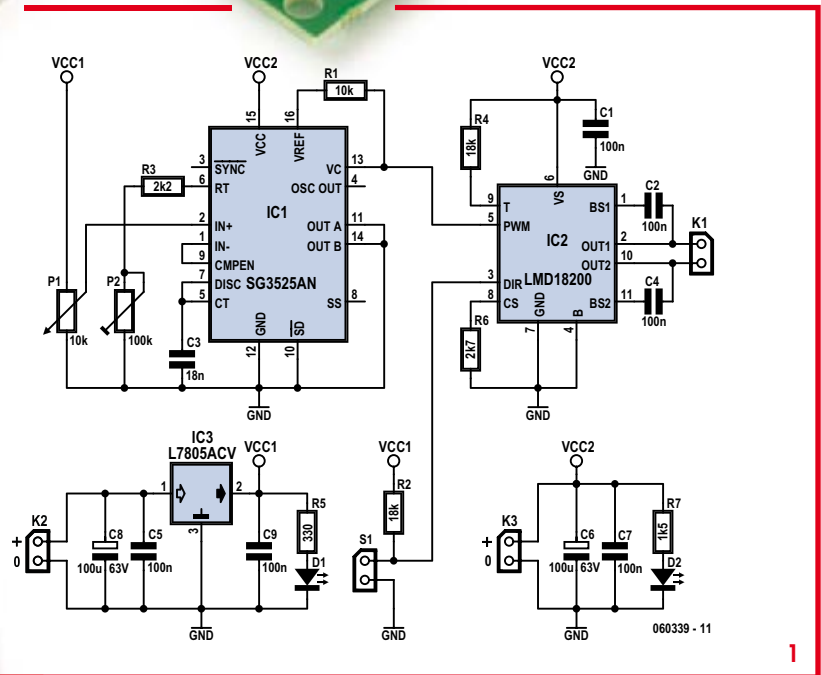
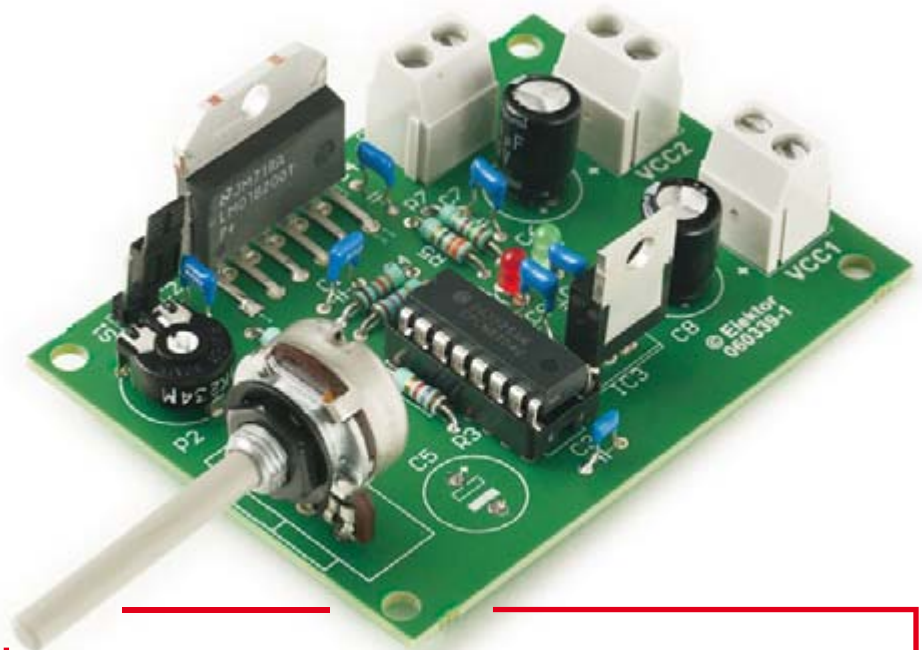
# 3 Amp PWM DC Motor Controller

Rajkumar Sharma

This circuit is intended for motion control applications, a common occurrence in robotics! This affordable PWM DC Motor controller can control any PMDC motor specified at 12 V to 30 V and 3 Amps max. Motor direction is controlled with a slide switch and motor speed, with an ordinary potentiometer.

The circuit diagram in **Figure 1** shows ICs type LMD18200 and SG3525 at the heart of the circuit. The SG3525 is a pulsewidth modulator control circuit and the LMD18200, an H-Bridge to enable the motor to be run in both directions.

The SG3525 affords frequency control and duty cycle control. The oscillator frequency is determined by the components attached to pins 5 and 6. Preset P2 serves to adjust the frequency between 1.16 kHz and 35 kHz. Although it is generally recommended to stay above 20 kHz as otherwise the motor will produce audible sound, in some cases that's just not possible depending on the motor you're using. Pot P1 determines the duty cycle, which can be adjusted from 10% to 100% to effectively control the motor speed. Internal transistors are used in a such a way as to obtain 100% duty cycle. The internal driver transistors are grounded by pins 11 and 14 for alternate oscillator cycles. Pin 16 of the IC is the REF V terminal, which gives 5 V out. Resistor R1 feeds the supply voltage to an

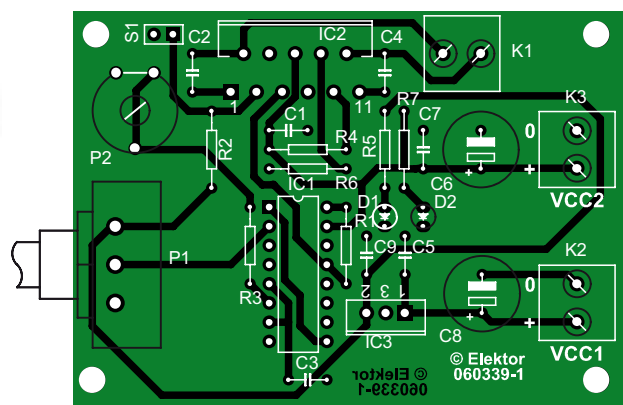


## Features

- Motor supply voltage 12 V to 30 V DC
- Delivers up to 3 amps continuous output
- PWM frequency adjustable from 1.16 kHz to 35.1 kHz
- PWM continuously adjustable by pot
- Duty cycle 0 to 100%
- Direction control by switch
- Shorted load protection
- Thermal shutdown
- No SMD components
- No microcontroller

internal open-collector transistor for TTL-level PWM output.

Moving on to the LMD18200, slide switch S1 (on header S1) governs the Direction control input (pin 3) to change the direction of the motor from cw to ccw or vice versa. R4 is connected to thermal flag pin T (pin 9), which is not used here. The function may be used to flag a warning when



the chip temperature is 145 degrees. The IC is automatically shut down when 170 degrees C is reached. Pin 8 of the LMD18200 is the current sense input. R6 connects this pin to ground. The Brake input (pin 4) is hard wired to ground. C2 and C4 at the motor output are 'bootstrap' capacitors. Pins 2 and 10 are the H-Bridge outputs powering the DC motor.

In the power supply section, capacitors C5 through C9 serve to suppress noise on the two supply rails. The L7805ACV voltage regulator for the logic supply accepts any unregulated DC voltage between 7.5 V to 18 V applied to K2. The other supply connector, K3, is for the motor power. The capacity of the motor supply of course depends on the motor used. If the motor is specified at 12 V then R7 should be 1 k $\Omega$ , and if it is 24 V then 1k $\Omega$ 5 should be fitted. If you want to use a heavy-duty motor consuming more than about 1 A, it may be worthwhile to strengthen the copper tracks to/from K2/K3 with lengths of 1.5 mm<sup>2</sup> solid copper wire.

## COMPONENTS LIST

### Resistors

R1 = 10k $\Omega$   
 R2,R4 = 18k $\Omega$   
 R3 = 2k $\Omega$   
 R5 = 330 $\Omega$   
 R6 = 2k $\Omega$   
 R7 = 1k $\Omega$ 5  
 P1 = 10k $\Omega$  potentiometer  
 P2 = 100k $\Omega$  preset

### Capacitors

C1,C2,C4,C5,C7, C9, = 100nF

C3 = 18nF  
 C6,C8 = 100 $\mu$ F 63V

### Semiconductors

D1,D2 = LED  
 IC1 = SG3525AN  
 IC2 = LMD18200  
 IC3 = L7805ACV

### Miscellaneous

K1,K2,K3 = 2-way PCB terminal block, lead pitch 5mm  
 S1 = slide switch  
 PCB, ref. 060339-1  
 from www.thepcbshop.com

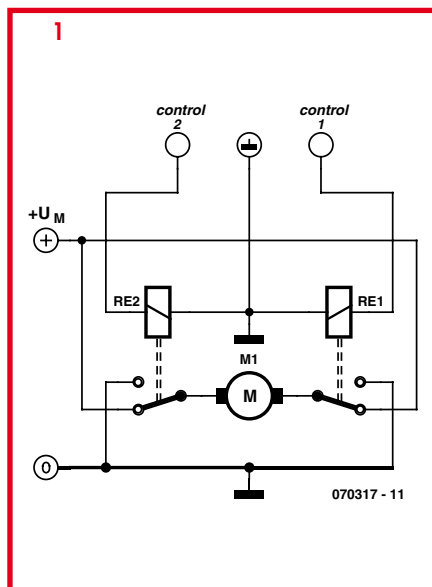
If you want to interface the driver with a source supplying 0-5 V, simply remove potentiometer P1 and apply the analogue voltage to pin 2 of IC.

**Figure 2** shows the PCB designed for the driver, which should fit many applications at crucial locations in a robot. After all, most forms of motion of a robot will

require a motor of some kind. The board has been designed for compactness whilst using leaded components only, i.e., no SMDs in sight here. The copper track layout and component mounting plan are contained in free download no. **060339-1.zip** from our website.

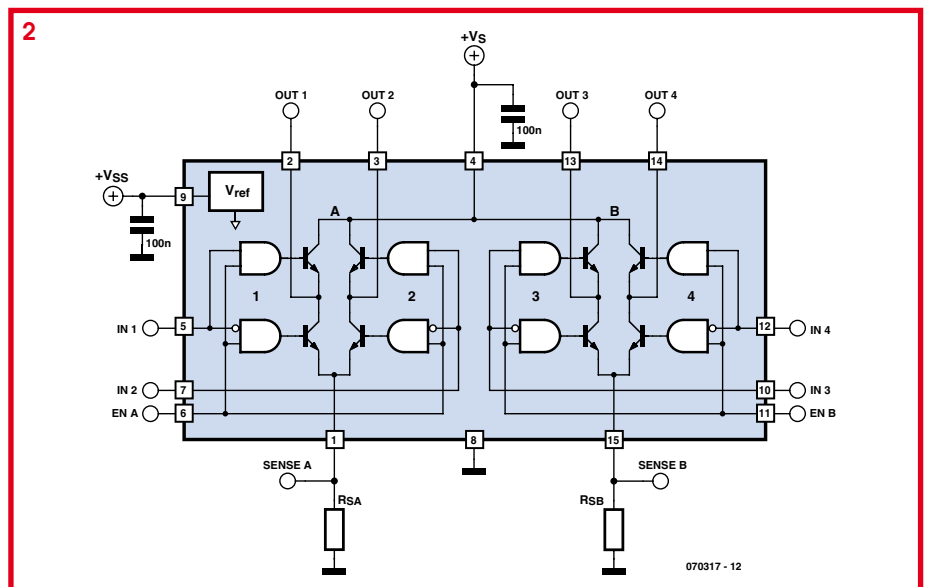
(060339-1)

# Driving Higher Power DC Motors



## B. Broussas

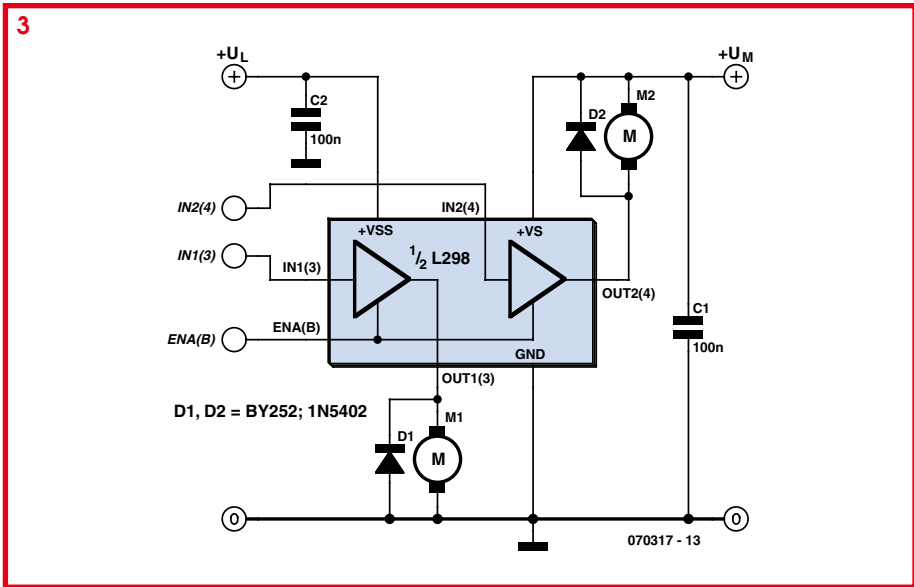
Driving the 'small' motors that may be used in robotics doesn't usually pose much of a problem. Servo motors actually have their own drive electronics, stepper motors can be easily driven by conventional power transistors or by ULN2803 ICs as has been shown elsewhere in this issue, as they rarely draw more than a few hundreds of mA. For small DC motors, simple transistors will suffice, unless you prefer for



example the LB1630 from Sanyo, though limited alas to a current of 400 mA and a voltage of 6 V.

On the other hand, as soon as the motor starts drawing 1 A or more, or its supply voltage exceeds 20 V or so, the situation gets more complicated — all the more so because many of you don't much care for power electronics. So, this article hopes to give you a few ideas or research paths for driving such motors.

The first method for controlling a higher-power DC motor is none other than the good old relay, or to be precise, pair of relays. As long as you wire them as shown in **Figure 1**, you have control over the operating direction, depending on which relay is engaged and which is not, as well as a stop control that acts as a very effective electrical brake, when both relays are in the same position. In this situation, the motor is short-circuited and is braked by



its own back emf (electromotive force). Relays capable of switching 10 A and yet only requiring 5 V and a few tens of mA for their coils are commonplace nowadays (see for example the Finder relays) and so can be used in this way without difficulty. If you don't like this electromechanical solution, we recommend you to use the bridge power IC, one worthy representative of which is the L298 from ST Microelectronics. As its internal block diagram

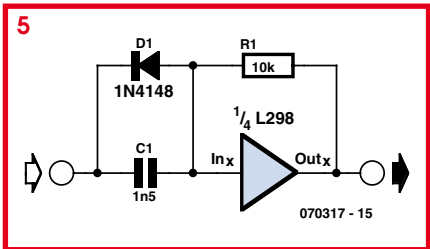
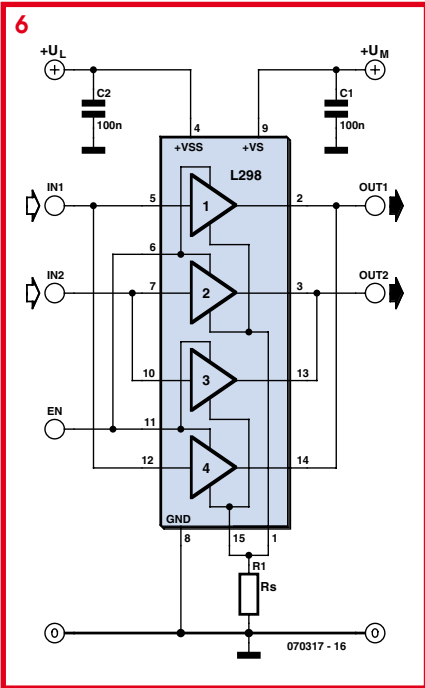
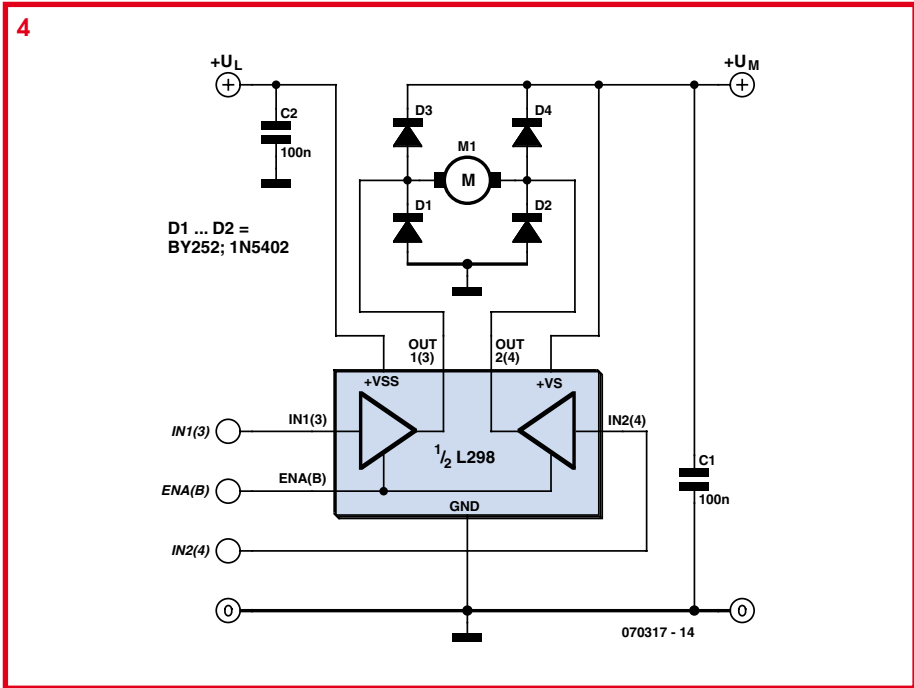
shows (Figure 2), this IC includes four bridge power amplifiers, preceded by logic control circuitry. Originally designed for driving 'big' stepper motors, this IC is suitable for a host of other applications, of which here are a few examples. By virtue of the relative independence of the amplifiers it contains, it can be used to drive four motors, as long as you are content with a single direction of rotation. It is then possible to take one of these motor

connections to earth or to the supply rail, as indicated in Figure 3. By juggling with the combinations of logic levels on the control and enable inputs of the L298, you can even have two options for motor stopping, as indicated in the table below: the 'freewheel' or unbraked mode, or the braked mode, as seen previously with the relay circuit. Table 1 shows the relevant logic level combinations.

If the direction of rotation of the motor has to be able to be changed, it is necessary to use a bridge or H connection, as shown in Figure 4. Note that it is possible to drive two motors in this way from one L298, since the IC contains four amplifiers. So, a single L298 is usually enough for the right and left propulsion motors of a mobile robot.

Although the circuit does have protection against overheating, be aware that you can increase its operating safety by monitoring the current drawn by the motors. To do this, all that you have to do is to fit a very low value resistor between the SENSE A or SENSE B inputs and earth. All the current drawn by the motor connected to the corresponding amplifier will then pass through this resistance, and by simply applying Ohm's law and measuring the voltage at these inputs, it is possible to monitor this current.

If you don't wish to use this monitoring, you



**Table 1.**

EnA(B)	In1(3)	In2(4)	M1	M2
H	H	H	Braked	Running
H	L	L	Running	Braked
L	X	X	Freewheel stop	Freewheel stop



are recommended to protect the IC against possible shorts of its outputs to earth, which are the most likely to occur in a robot (a motor terminal touching the metal chassis, for example!) In this case, STMicroelectronics recommends the circuit in **Figure 5**. This circuit trips in 10  $\mu$ s and resets by itself when the short disappears.

The L298 is capable of withstanding a maximum supply voltage of 46 V and each of its power amplifiers can supply a current of 2 A, already a more than comfortable value, even for a relatively heavy mobile robot. If that isn't enough for you, it is also possible to connect the power amplifiers

in parallel, as long as you go about it the right way. You then have a maximum output current of 3.5 A. To do this, you must adhere to the circuit in **Figure 6** and no other; that is to say, you must only parallel amplifiers 1 and 4 on the one hand, and 2 and 3 on the other.

Just before we reach the end of our article, do note that the L298 does not include built-in protection diodes, so it is vital to provide them externally as we have done in each of our figures, otherwise the L298 is guaranteed to be destroyed the first time the robot's wheels turn!

This IC is of course not the only one that

can be used for driving higher-power DC motors for robotics applications. More recent and/or higher performance packages do currently exist. But the L298 does have the advantage of being readily available, inexpensive, and able to fulfil a wide range of needs, which to our mind more than justifies this presentation of its various modes of use. And if you are ever so slightly curious, you'll find copious application notes about it on the STMicroelectronics website ([www.st.com](http://www.st.com)), which will be a good source of additional ideas for implementing it.

(070317-1)

# Driving Stepper Motors: KISS

## Without using a specialised IC

B. Broussas

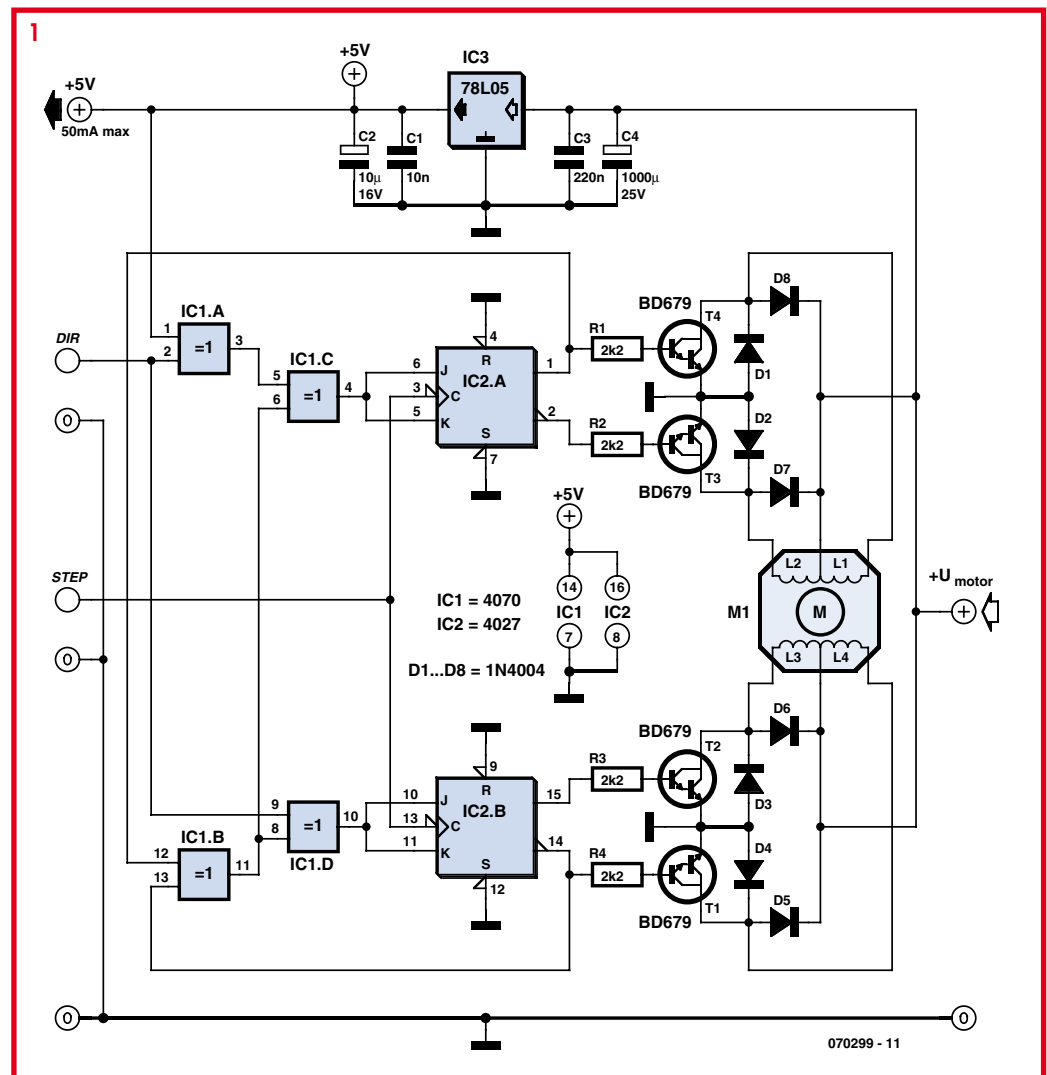
Used a great deal in robotics, where it is a direct competitor for DC motors and radio-control servos, the stepper motor does however suffer from the comparative difficulty encountered in driving it. Unlike its DC equivalents, which rotate as soon as power is applied, the stepper motor requires a succession of pulses applied to its various windings to make it turn. On the other hand, and provided its mechanical capacities are not exceeded, the number of basic steps moved by the motor corresponds to the number of pulses applied to its windings. Hence in this way we have available to us virtually automatic positioning information, impossible to obtain with a DC motor.

In many robots we've seen, driving the stepper motor(s) was the job of a specialized IC, one of the front-runners being the L293 from ST Microelectronics which, despite its age, is still very much current. Yet there are many other solutions, such as direct control by one of the ports of the microcontroller that runs the robot — or the one we are suggesting here, which just uses... two standard CMOS logic ICs!

As you maybe already know, there are actually two types of stepper motor: single-

pole motors and 2-pole ones. While the first only need single pulses sent to their four windings, the latter require inversion of the signal polarity applied to the wind-

ings. So as not to complicate our circuit unduly, we have designed it for single-pole motors, the timing diagram for which is given in the table below.



Reading the columns of this table from 1 to 4 corresponds to rotation of the motor in one direction, while going from 4 to 1 reverses the direction of rotation. Each column of this table corresponds to one mechanical step of the motor. These steps vary from 1.8° to 7.5°, depending on the type of motor chosen.

So the circuit of our driver without specialized ICs, shown in **Figure 1**, is very simple, since its 'intelligence' is in fact confined to two logic ICs, IC1 and IC2, which are simple exclusive-OR (XOR) gates and a dual J-K flip-flop, while the power stage is built around perfectly ordinary general-purpose bipolar transistors.

The pulses to make the motor turn must be applied to the STEP input. Each pulse makes the motor turn through a single step in one direction or the other; this direction is determined of course by the state of the DIR input. This acts on exclusive-OR gates IC1a and IC1d, used here as programmable inverters.

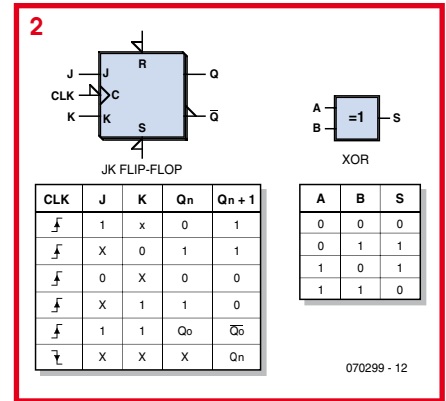
Remember that an exclusive-OR gate can be regarded as a gate that inverts or not the signal from one of its inputs depending on the state of its other input. This is easy to see from the truth table in **Figure 2**. If input A is '0', the output is the same as the signal applied to input B (0 gives 0 and 1 gives 1). However, if input A is '1', the signal applied to input B appears inverted at the output (0 gives 1 and 1 gives 0). Nothing very new there, but we did want to underline this interpretation of the truth table of the exclusive-OR used as a programmable inverter, as we have noticed that many of you aren't familiar with it (or have forgotten!)

The truly active part of the circuit is formed by the two J-K flip-flops IC2A and IC2B. **Figure 2** sums up the truth table for these flip-flops, which is made simpler here because J and K are always both at the same level. When these inputs are '1', the flip-flops change state at each clock pulse, i.e. for each pulse applied to the STEP input. In the reverse situation, i.e. when J and K are both '0', the outputs Q and  $\bar{Q}$  remain in the previous state.

If you still have doubts that this will indeed generate the timings in the table above, arm yourself with some graph paper, a pencil, and some patience, and draw out the timing diagrams of the signals supplied, when DIR is at some arbitrary level of your choice.

The power stage is built using bipolar transistors, protected from the voltage spikes generated by the current switching in the motor windings by diodes D1–D8. With the transistors used, it is possible to switch currents of up to 3 A, allowing plenty of flexibility in the choice of stepper motor. The logic side of the circuit is powered from a fixed 5 V supply, stabilized by IC3, making the STEP and DIR inputs TTL-compatible. This supply may also be used to power the circuit prior to this driver, as long as you don't exceed around 50 mA with the regulator chosen.

If your motor is powered at 6 V, it is advisable to replace IC3 by an LM2936Z5, for example, which is a low-volts-drop 5 V regulator. For correct operation, the 78L05 originally specified for IC3 requires almost 2 V between input and output — clearly impossible to obtain with a motor supply of only 6 V.



Table

Step number	1	2	3	4
Winding 1	1	1	0	0
Winding 2	0	0	1	1
Winding 3	1	0	0	1
Winding 4	0	1	1	0

Still on the subject of the motor supply voltage, note that it can be up to 24 V if necessary. As long as the current drawn by the motor doesn't exceed 1 A, the transistors don't need a heatsink, but one is advisable above that. It need only be a few cm<sup>2</sup>, since the transistors are operating here in switching mode and so dissipate relatively little power.

(070299-1)

#### Web Link

L293 spec. sheet

<http://www.st.com/stonline/books/pdf/docs/1328.pdf>

# Robot Footballer

by Julian Straub

You will do doubt have seen pictures from 'RoboCup' showing robots booting footballs from one end of the pitch to the other. Building an electromechanical robot like this is entirely within the capability of the hobbyist with the help of a few cheap everyday items.

In order to give the ball a good kick the robot's feet are powered by linear solenoids. Acceleration is more important than force, however, and so we eschew readily-

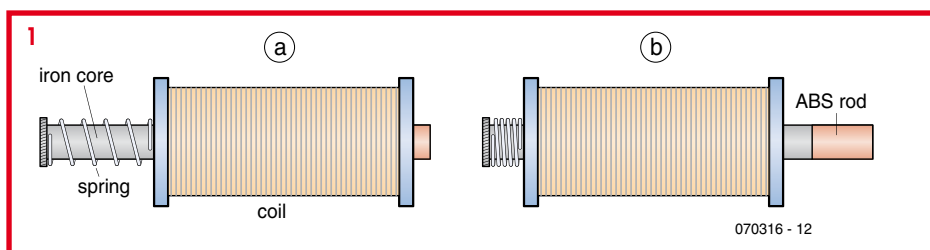
available solenoids which generally operate on 12 V or 24 V and which, although powerful, are much too slow for our purposes.

The integral of force over time (or impulse) produced by a coil with an iron armature depends, disregarding constant factors such as turns count, coil geometry and perme-

ability, on the change in the coil current. The faster we wish to change the current, the higher the voltage we will have to use. And so we need a high voltage supply.

We can generate a high voltage using the flash from a disposable camera of the sort that can sometimes be had for free from photography shops.

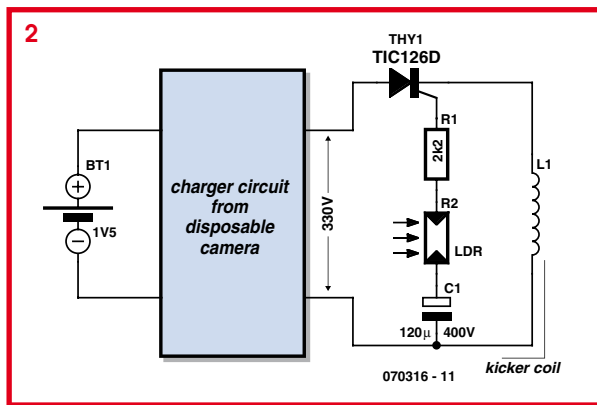
The camera electronics includes a high-voltage cascade circuit with a storage capacitor for the flash. These components are ideal for pressing into service as part



of a robotic footballer.

Open the camera carefully. First remove the battery making sure not to burn your finger by touching the capacitor contacts. For safety's sake discharge the capacitor using a resistor of a few kilo-ohms before removing the printed circuit board. Because we will later want the capacitor to be charged continuously, bridge the power supply switch connections. The circuit in the camera tested by the author (made by Kodak) charges a 120  $\mu$ F high voltage capacitor to 330 V in 16 s from a 1.5 V battery.

Next we turn to the sewing box for inspiration. We need two cotton reels from which we will fashion inductors using enamelled copper wire. On the one hand it is advantageous to use wire that is very thin so that we can have as many turns as possible and hence a high inductance, while on the other hand the high ohmic resistance of this arrangement limits the maximum current that can be achieved; we need to find a good compromise. To simplify



making the windings with very fine wire, first wrap the coil former with a layer of thin double-sided adhesive tape. This will hold the wire in place as you wind the first layer. Use adhesive tape again after each successive layer of wire. Finally, wrap the finished coil in insulating tape so that just the two connection wires (with extra insulation) protrude.

The two iron cores can with a little luck be found in the clearance bin at an electronics shop. If not, you can resort to do-it-

yourself: the cores can be ordered from any metal warehouse that can supply steel rounds. Ensure that you do not buy vanadium steel or a non-ferrous metal. The size should be chosen so that the lengths of metal pass through the cotton reels without too much play. In each drill a hole in one end and fit a small washer to prevent the light compression spring from sliding down. The spring ensures that after each kick (Figure 1a) the foot will return smartly to its initial position (Figure 1b). The cores are fitted into the coils and a plastic cylinder, which will be the part that actually makes contact with the ball, is attached to the free ends.

Figure 2 shows how simple the drive circuit can be. A type TIC126D thyristor wired between the high-voltage generator and the coil triggers the kick. The thyristor is in turn triggered optically via an LDR, which ensures isolation between the high voltage electronics and the control circuit.

(070316-1)

# PIC Indicator Relay

Herbert Musser

Members of the motorcycling fraternity will welcome this design. Some types of motor cycle are notorious for having very short life expectancy of their filament indicator lamps. Vibrations transmitted through the frame are the culprit (especially if the indicators are mounted on the ends of long stalks). After-sales replacement LED lighting clusters are far more reliable but they create another problem: they draw less current than the filament lamp and cause the indicator relay to switch the indicators on and off far too quickly. The first attempt to solve the problem with an analogue electronic flasher circuit was not successful the first charge cycle of a capacitor was almost twice as long as successive periods, so the flash rate was not constant (maybe an integrator circuit would have given better results).

A better solution was produced using the mini PIC

circuit and FET shown here. This design also has a built-in bonus feature; motorcyclists are notorious for forgetting to cancel their indicators so the circuit includes a time-out function (jumper selectable) which automatically cancels the indicator after 120 blink cycles.

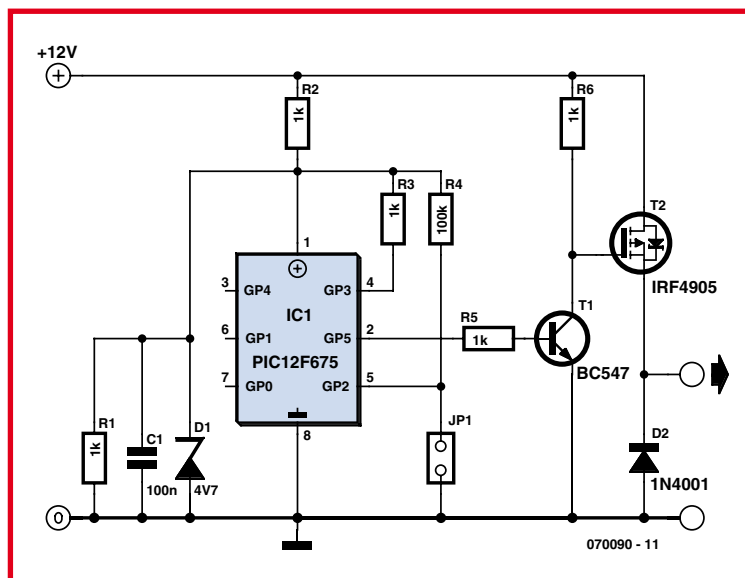
The circuit uses very few components and the finished circuit can usually be fitted into the existing indicator relay housing

with a space of around 20 mm x 30 mm. The output signal from the PIC controls the driver (T1) which then switches the HEXFET power transistor (T2). The IRF4905 has an extremely low on-resistance of just 20 m $\Omega$  and is capable of switching 74 A maximum. The supply (derived from indicator relay) is limited to 4.7 V by diode D1 and smoothed by C1 to reduce the effects of any interference from the motorcycle supply.

Firmware for this design applicable to the PIC controllers 12F629, 12F683 and 12F675 is available to download free of charge from the Elektor Electronics website, look for file # 070090-11.zip.

The finished circuit is reliable, impervious to vibrations, load current changes and best of all can be built for less than 7 pounds (approx. 10 euros). As a final thought you should check that traffic regulations allow the use of such home-made designs to be fitted to the motorcycle before it is driven on the road.

(070090-1)





# Catapult for Robots ...or Other Uses

Pascal Liégeois

In this age of laser rays, it seems anachronistic to talk of catapults — but it's not as absurd as you might think.

Many robot competition themes around the world have involved at some point picking up balls, of different formats according to the competition, and projecting them into a receptacle, often at quite substantial distances for our little robots.

There is one well-known type of very light ball, the ping-pong ball, that is very often used as a projectile in this type of competition.

Besides picking up these balls, projecting them often poses a problem of accuracy and reliability.

In this short article, the author is suggesting his own solution — not necessarily the best in the world, but at least it is proven. This catapult re-arms all by itself after each shot, within 2 seconds, and its range can be adjusted by altering the ballistic curve, using just a single potentiometer.

The very simple, cheap electronics don't require any programmable components, and have an output available to tell the carrier about the status of the catapult.

## Circuit

The main element of the system is a perfectly ordinary standard servomotor, as used in modelling. This type of actuator is a small marvel, containing as it does a position-servoed motor with step-down gearing, by way of a potentiometer and suitable electronics.

A servomotor is controlled using a fixed frequency signal (50 Hz) whose pulse width is variable, generally from 1 to 2 ms.

To produce this signal here, we use the famous NE555 (IC1) as an astable, wired in such a way as to provide the required frequency. Diode D1 in parallel with resistor R1 determine the duty cycle, and set the negative-going part of the pulse at around 18 ms. The width of the positive-going part is adjustable by means of R2 and P1 or P2.

The output of the 555 feeds the input to the servomotor.

The 'electromechanical' part of the circuit is based on the use of a DPDT relay and two microswitches. Sw1 is used to trigger automatic re-arming of the catapult, while Sw2 fulfils two functions: it gives information about catapult re-arming, and once

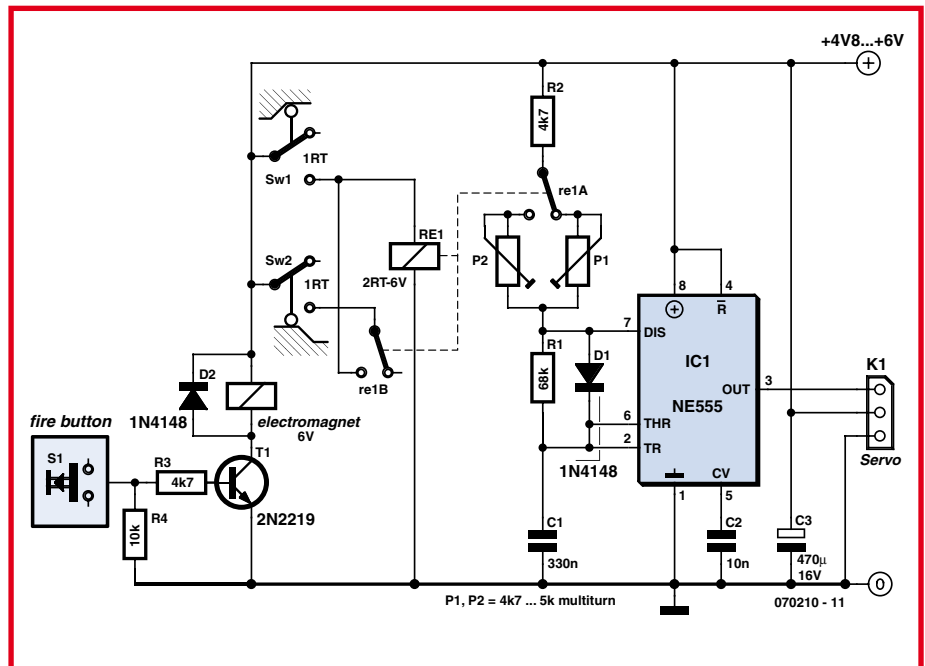


Figure 1. The catapult electronics.

this re-arming has taken place, it lets us reposition the servomotor in the firing position.

In the circuit diagram, Sw2 is shown operated, corresponding to the catapult's re-armed position.

Referring to the drawing in **Figure 1**, it's easy to follow the operation of the catapult.

When the solenoid is briefly powered via the 'FIRE' input that controls transistor T1, the lever L is released, pulled up by spring R. This lever ends its travel up against the rubber stop G attached to the motor's servo arm, wheel P.

In coming to rest against this stop the projectile is fired, and the lever also operates Sw1, energizing the relay RE1, which in turn latches via its contact re1B and microswitch

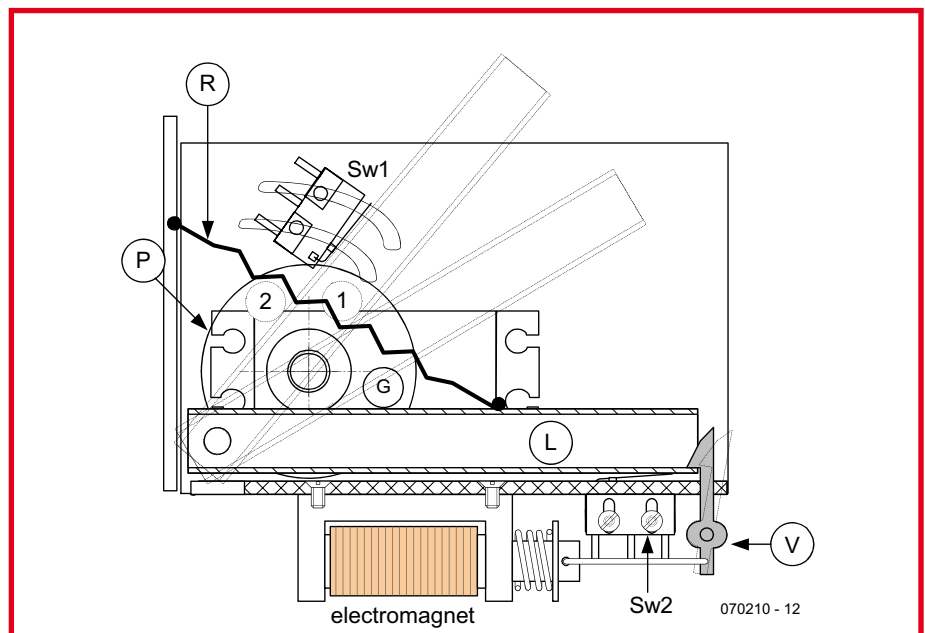


Figure 2. Operating diagram of the catapult.

Sw2 (by this time returned to its rest state). The servo motor starts to turn clockwise and the rubber stop forces the lever back to its re-armed position. At the end of the travel, the lever hooks under the trigger catch; at the same time, Sw2 is operated, and unlatches the relay, which goes back to rest, commanding the servomotor to return to the firing position. Preset P1 lets us adjust the upper position of the stop and thereby the range of the shot, as explained in **Figure 3**. P2 lets us set the latching point of the lever in the re-armed position.

Figure 1 shows two firing positions (greyed) of the lever, and the corresponding positions for the rubber stop (numbered 1 & 2). To adjust the firing range, you simply need to know that, logically enough, in position 1 the ball will go higher, and in position 2 the ball will go less high. Everything depends on how it is being used: if you want to drop a ping-pong ball in a pocket in the ground, it's best to plan on getting there via successive bounces, and so to fire higher. On the other hand, if you are aiming for a basket high up, you need to aim 'spot on' into it, and so allow the lever to go higher. Once the adjustments have been set, you'll be amazed by the repeatability of this system.

## Construction

### Electronics

The electronic part is relatively simple and can be built on a small piece of prototyping board. The DIL relay RE1 can be fitted into a turned-pin DIP14 socket. The servomotor connector can be made using three sections (a 10 mm length) of 2.54 mm (0.1") pitch SIL pinheader strip.

Mark the signal pin, so as to avoid any mistakes when connecting the servomotor. The IC can be fitted into an 8-pin socket. Presets P1 and P2 should preferably be multiturns, horizontal or vertical. Check your wiring carefully. Power the circuit without IC1 or the relay fitted. Check the supply rails to IC1 and to the commons of the switches, which will be connected to the circuit via wires of around 10 cm or so. Check the presence of +V<sub>CC</sub> on the central pin of the servomotor connector. Connect the 'FIRE' input briefly to +V<sub>CC</sub> and check that the solenoid operates.

### Mechanics

Although not terribly complicated, the mechanics do require a little care all the same. The drawing in **Figure 2** details the key parts and elements of the system. The chassis is made mainly from a piece of L-section aluminium angle, or an equivalent folded section. The servomotor, fitted with an approximately 35 mm diameter wheel as its servo arm, is mounted on the vertical plane of this angle.

The pivot for the lever is slightly forward of the servomotor shaft. In my own case, I made this lever out of 5 mm square brass tube. This hollow section allows the little catch to hook into the lever once it is re-armed. This catch is operated by the solenoid via a small connecting rod. The solenoid is a 6 V type, mounted under the horizontal plane of our aluminium angle. The positioning of the microswitches is important, particularly that of Sw2, whose position is set once the optimum re-arming position has been set. This setting can only be done once the electronics described above have been built. Sw2 is mounted onto the angle by way of a small

bracket with two oblongs fixing slots, to allow the microswitch to be positioned.

For Sw1, two curved slots will need to be made in the vertical plane, where the servomotor is mounted, so as to be able to adjust the firing range, in conjunction with preset P1. Once the mechanics have been completely finished, the lever should be put into the lowest position, which will slacken the solenoid spring. Check that the catch hooks properly onto the lever by at least 1 mm. This action must take place without forcing, the solenoid spring must allow the catch to hinge to the right before hooking into the tube.

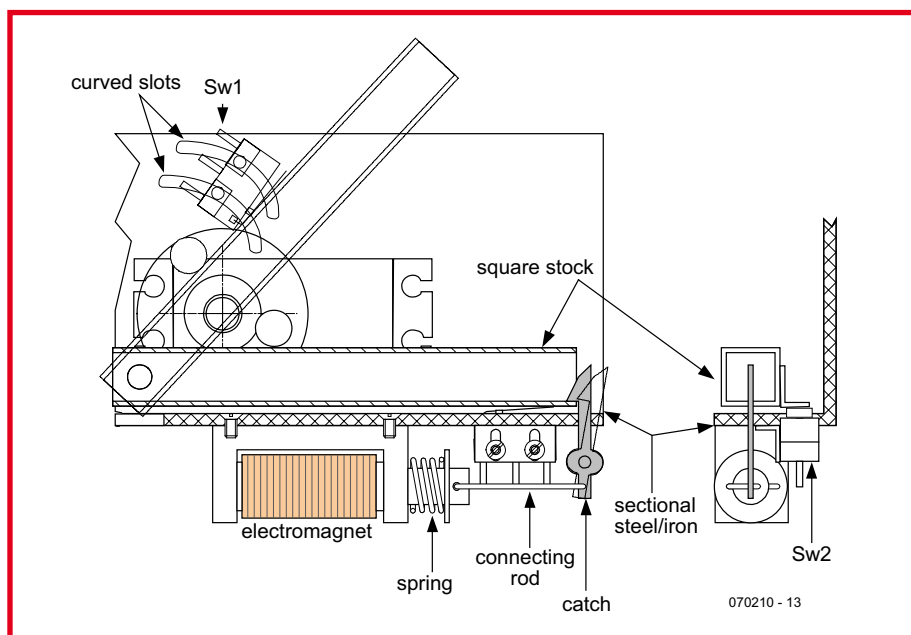
Check that the solenoid is properly fitted with its return spring, which may be fitted between the coil and the armature, or actually inside the coil, within the space where the armature moves. The spring must push the armature lightly so that it comes back out of the coil once it has been activated.

## Adjustment

Setting-up is easy. Don't fit the relay into its socket. Put the lever into the re-armed position and check that the catch holds it properly in the horizontal position. Release the catch and make sure the lever is pulled up properly by the firing spring. Apply power; the servomotor will take up a random position. Using P2, get the servomotor, via the rubber stop, to position the lever horizontally until the catch engages. Adjust the position of Sw2 so that it is operated by the small bracket attached to the lever. If the servomotor fails to operate, check the circuit, the soldering, and that the servomotor connector is the right way round — i.e. that the signal and earth pins are not reversed. Turn off the power. Fit the relay into its socket. Re-apply power. The servomotor should take up some random position. Adjust P1 to bring the roller into any firing position (1, for example). Turn off the power and adjust Sw1 so it is operated by the lever in its upper position.

Re-apply power. The catapult should re-arm all by itself and the servomotor should then return to the upper position as described above. Everything is now working. That's fine. Operate the solenoid to check that firing takes place correctly. You can now fit the lever with a support for the projectile (ball).

It's worth noting that the unused NC contact of Sw2 carries +V<sub>CC</sub> indicating the catapult is re-armed...



**Figure 3.** Mechanical construction details.

(070210-1)

# Servo to Motor Conversion

Paul Goossens

Servos, originating from their application in model building, are usually used to operate arms, feet and other 'tools' of a robot. In addition to these obvious uses they are also very suitable as a motor to drive the wheels of a robot, for example. To do this, the standard servo does need to be modified first however.

Servos have been used for a long time in the model construction arena. As a result they are readily available and often at attractive prices.

## Standard servo

The purpose of a standard servo is to bring the shaft into a certain position and to keep it there. At the input the servo expects a train of digital pulses. The duration of these pulses determines the position that the shaft has to assume.

The internal electronics uses a potentiometer which is mechanically coupled to the shaft to measure the position of the shaft. If the length of the pulses does not correspond with the present position of the shaft then the electronics will drive the internal motor.

If the shaft is too far to the right then the motor will turn the shaft to the left and the other way around. The instant that the shaft reaches the correct position the motor will be turned off.

## Small adjustment

A standard servo is therefore not suitable to turn wheels or similar things. However, with a little bit of tinkering we can make a servo suitable for this job.

The trick is very simple. The potentiometer is replaced by a voltage divider with two 10 k $\Omega$  resistors. In this way the electronics 'thinks' that the shaft is always in the centre position.

If we now give the servo a pulse of 1.5 ms duration then nothing happens. The servo will turn the motor off. If we make the pulse duration 1 ms then the servo will attempt to turn the shaft to the left-most position. To that end the servo will let the motor turn to the left. On its sensor input it continues to 'see' that it is in the centre position. The motor will therefore continue to turn to the left.

To let the motor turn the other way we supply a pulse that lasts longer than 1.5 ms (2 ms, for example). The motor stops again when we make the pulse 1.5 ms long.

Note that most servos have a mechanical

end-stop, which prevents the shaft from turning any further. We have to remove these two end-stops first otherwise the motor will go up in smoke during testing!

## Mechanical

This modification doesn't mean much from the electronics perspective. The skill is to do this in the small housing of a servo. As an example we use a cheap servo from Conrad (Figure 1). On the bottom of the servo are four screws that we remove first. After this, the bottom cover can be removed.

Carefully loosen the cover plate. This contains a number of gears that can easily drop out. We need to know exactly how these are placed in the enclosure, because we need to put them back in the exact same place later on! Taking a picture of the inner works including the gears can be a very handy reference later on when putting the servo back together. Once that is done the servo looks like Figure 2.

Remove the output shaft. This sits on the shaft of the potentiometer. This shaft is fitted with a ridge, which together with 2 ridges in the housing forms the mechanical end-stop. This is undesirable, therefore remove this ridge with a sharp knife (Figure 3).

The next job is to remove the PCB from the housing. Should this not come out easily, then a careful push on the shaft of the potentiometer usually ensures that it comes out of the housing after all. The motor should have a small gear. This will sometimes be caught by the housing. If that is the case you need to retrieve the gear and put it back on the motor shaft.

The potentiometer and motor are easily recognised. We now replace the potentiometer with two resistors. The resistors are each individually soldered to one of the outside connections, where the potentiometer used to be. The other connections for both resistors are then soldered to the middle connection.

Voilà, the servo is now converted. If all is well, your servo will look about the same as our prototype in Figure 4.

What's left to do is to put the servo back together. Make sure that all the gears are in the correct position and everything is free to rotate. Also check that the little gear on the motor is still in the right place.

Finally we screw the bottom cover back in place. The servo is now ready for use, but now as a motor and no longer as a servo!



(070358-1)



# Driving Stepper Motors

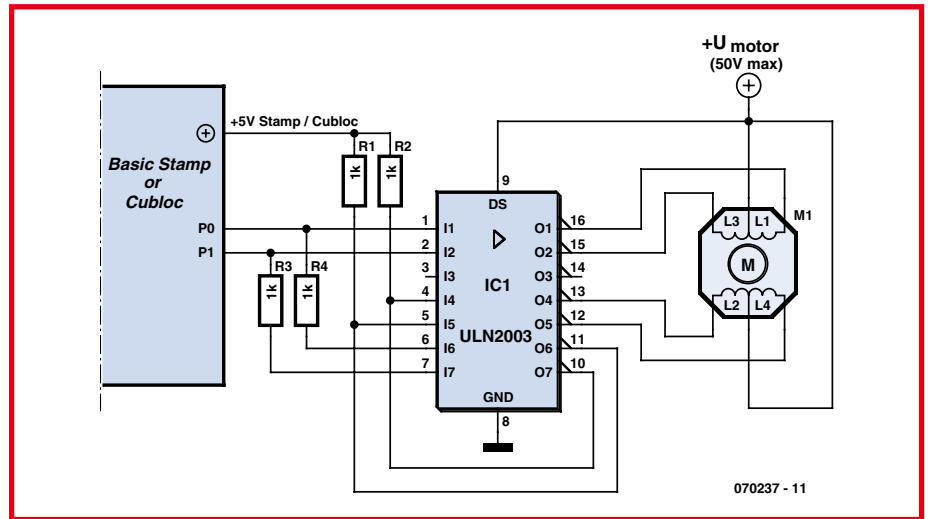
## Using a Basic Stamp or a PIC programmed in Basic

C. Tavernier

When we want to motorize a robot, two main solutions are open to us: the DC motor, used alone or in a converted radio-control (RC) servo system, as explained elsewhere in this issue; and the stepper motor. Although the DC motor is very suitable for everything to do with propulsion, the stepper motor is more suitable when it comes to performing precise positioning, as required for a robot arm, a sensor moving in one plane, etc.

But while controlling a DC motor is relatively simple (it turns as soon as it is powered), controlling a stepper motor is a little more tricky. They only turn when their various windings receive pulses, which have to be presented in a quite specific order to make the motor turn one way or the other.

Because of this pulsed drive, these motors don't turn continuously, but in fact advance at each pulse by one basic step — whence their name. The size of these steps can vary between 1.8 and 7.5°, depending



per motor: single-pole and double-pole. The former are simpler to drive, as all you have to do is apply voltage or not to their windings, while in 2-pole motors, the voltage applied to these windings has to be regularly inverted, slightly complicating the circuitry that's required.

**Table 1** indicates the order in which a single-pole motor's windings must be powered to make it turn. Going through the columns of this table from 1 to 4, the motor turns clockwise, while reversing the order changes the direction of rotation. Each column of this table corresponds to one

mechanical step of the motor — remember, that's between 1.8 and 7.5 degrees depending on type.

There are numerous specialist ICs on the market for driving stepper motors, and various solutions are on offer in this issue of Elektor, but if your robot uses a Basic Stamp or a PIC microcontroller programmed in Basic, there is an extremely simple and cheap solution for making it drive this type of motor.

All we actually need to do is use a perfectly ordinary ULN2003 or ULN2803,

Table 1

Step number	1	2	3	4
Winding 1	1	1	0	0
Winding 2	0	0	1	1
Winding 3	1	0	0	1
Winding 4	0	1	1	0

on the type of motor used. This stepped advance makes it possible in principle for the program controlling the motor to know its position very accurately. But for this to remain true, we have to take care not to exceed the motor's maximum allowable loading, as in that case the motor might fail to advance for every pulse received, and thus 'skip' some steps.

Another far from negligible advantage of the stepper motor is that if it is powered but does not receive any pulses, it remains blocked where it is. So we have a sort of electric brake — though of course still on condition that the motor's load capacity is not exceeded, as mentioned above.

There are currently two families of step-

```

` Control of a unipolar stepper motor
` The step number is put in w1
` The rotation direction is defined by b0

` Variable's definition
Symbol direction = b0
Symbol incr = w1
Symbol index = w2
Symbol delay = b6

` Initialization

dirs = %00000011
pins = %00000001
b1 = %00000001

` Here the application program must initialize
` incr, direction and delay with the required values

if direction = 0 then incrincr
b1 = b1 ^ %00000011

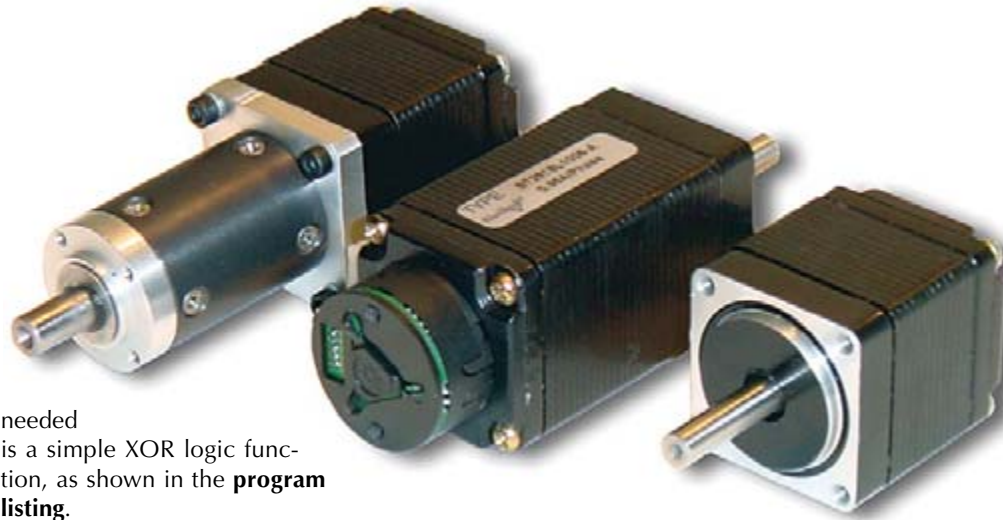
incrincr:
  for index = 1 to incr
    pins = pins ^ b1
    b1 = b1 ^ %00000011
    pause delay
  next
    
```

respectively seven-way or eight-way integrated power Darlington's normally used to drive relay coils. The required software is very simple, thanks to a couple of tips suggested by Parallax (the manufacturer of the Basic Stamp).

The first is to note that the status of windings 1 and 2 on the one hand, and windings 3 and 4 on the other, is always opposite, as indicated in the attached table. Because of this, the motor can be driven using just two of the Basic Stamp outputs, as shown in the very simple circuit we are suggesting.

Windings 1 and 3 are driven from two lines of the Basic Stamp port, after amplification by the ULN2003 (or 2803). Windings 2 and 4 receive these signals after inversion, performed using two of the spare amplifiers in the ULN2003 (or 2803), which is overkill but perfectly practical. Note the presence of the two essential 1 kΩ pull-up resistors, connected to the outputs of the amplifiers in the ULN2003 (or 2803), as the Darlington's are only open-collector.

The second tip suggested by Parallax consists of directly calculating the sequence of signals to be applied to the Basic Stamp's P0 and P1 outputs, rather than getting these data from a table. All that's actually



needed is a simple XOR logic function, as shown in the **program listing**.

This short example of code may be included as is into a more complete application. As can be seen, it makes the stepper motor connected as shown in the figure turn through the number of steps previously loaded into w1. The direction of rotation is determined by the contents of b0. If b0 is anything other than 0, the motor turns one way; if not, it turns the other way. This program also lets us define the wait time between each step, by means of the data used in the PAUSE instruction; the only proviso is not to reduce this delay too much, taking into account the motor

being used and the load it is driving.

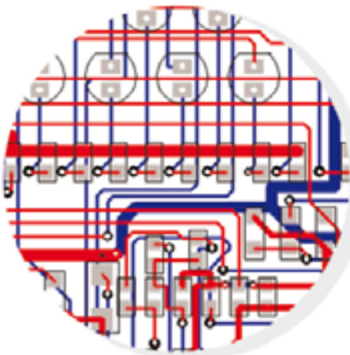
To be as general as possible, note that this example of code has been written in Basic Stamp I language. So it is fully transposable, without restriction, to any other type of Basic Stamp, as well, of course, as to any PIC programmed in Basic, since the majority of Basic compilers for PICs are compatible with the Basic Stamp I language. It can likewise be easily migrated to a PicBasic or a Cubloc from Comfile Technology.

(070237-1)

**No1 Number One Systems**

**Easy-PC** for Windows

**The World Beating PCB design software**

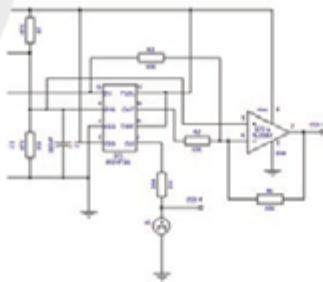


**Version 11 features**

- Design revision analysis
  - Star/Delta points
  - Apply layout pattern & groups
  - Design calculators
  - Electrical rules check (ERC)
  - Shape editing panel
  - Merge/subtract shape editing
  - Track analysis
- Plus many more exciting features...

**The new Easy-PC reaches even higher!**

The breathtaking new Easy-PC for Windows Version 11 is released. Winning accolades the world over, V11 of the market leading Easy-PC delivers even more remarkable value for money and sets the new benchmark for performance in PCB CAD.



Just try a demonstration copy of Easy-PC and prepare to be amazed...

Easy-PC delivers a simple to understand, fully integrated Schematics & PCB layout in a single application. Design and rules checks at all stages ensure integrity at all times. Professional manufacturing outputs allow you to finish the design process with ease.

**Stop press... Stop press... Stop press...**  
Easy-PC is supported running under Windows Vista

STILL FROM ONLY  
**£247**  
PLUS VAT

Call for a brochure, prices & CD on +44 (0)1684 773662  
or e-mail sales@numberone.com  
you can also download a demo from

**www.NumberOne.com**

Oak Lane, Bredon, Tewkesbury, Glos. GL20 7LR. United Kingdom



# Low voltage Rail-to-Rail Op Amps



## Solutions for extended temperature applications

### Microchip's Selected Extended Temperature Range Op Amps

Part #	GBWP	IQ Typical ( $\mu$ A)	Vos Max (mV)	Temp. Range ( $^{\circ}$ C)	Supply Voltage	Features
MCP6275/85/95	2/5/10 MHz	170/230/445	3	-40 to +125	2.0 to 5.5V	Rail-to-Rail Input/Output, Dual Connected with Chip Select
Selected Standard Op Amps						
MCP6231/2/4	300 kHz	20	5	-40 to +125	1.8 to 5.5V	Rail-to-Rail Input/Output
MCP6241/2/4	650 kHz	50	5	-40 to +125	1.8 to 5.5V	Rail-to-Rail Input/Output
MCP6001/2/4	1 MHz	100	4.5	-40 to +125	1.8 to 5.5V	Rail-to-Rail Input/Output
MCP6271/2/3/4	2 MHz	170	3	-40 to +125	2.0 to 5.5V	Rail-to-Rail Input/Output
MCP601/2/3/4	2.8 MHz	230	2	-40 to +125	2.7 to 5.5V	Rail-to-Rail Output
MCP6281/2/3/4	5 MHz	445	3	-40 to +125	2.2 to 5.5V	Rail-to-Rail Input/Output
MCP6021/2/3/4	10 MHz	1000	0.5	-40 to +125	2.5 to 5.5V	Rail-to-Rail Input/Output
MCP6291/2/3/4	10 MHz	1000	3	-40 to +125	2.4 to 5.5V	Rail-to-Rail Input/Output

Do you need low power consumption and extended temperature range operation? Microchip has a broad range of extended temperature op-amps with Gain-Bandwidths from 300 kHz to 10 MHz,

including many devices in ultra-compact SOT-23 and SC-70 packages. Microchip also offers the innovative MCP62X5 dual-connected op-amp family.

**microchip**  
**DIRECT**  
www.microchipdirect.com

Now  
**Pb-free!**  
RoHS Compliant

 **MICROCHIP**  
www.microchip.com/opamps



# Which Brain for my Robot?



So, after briefly discussing the special requirements typical of robots, we're going to present a certain number of microcontrollers and try to highlight their strengths in a robotics context.

## The need creates the system

Compared to a conventional electronic project, a robot possesses certain particular features that have a direct influence on the choice of which microcontroller to fit it with. So whether it is fixed or mobile — and the first robots amateurs build are very often mobile, as these are admittedly the most spectacular — a robot always includes one or more motors. As you've been able to discover throughout this issue of *Elektor*, these may take the form of radio-control servos, stepper motors, or DC motors. None of these are controlled in exactly the same way, but all of them require the microcontroller to know how to generate pulses more or less repetitively. Our robot is obviously fitted with sensors. Although the simplest versions make do with simple 'whiskers' or contact-based obstacle detectors, as robots evolve, they become literally covered in sensors, some of which can be highly complex.

## A mini practical guide

C. Tavernier

[www.tavernier-c.com](http://www.tavernier-c.com)

**More than any other electronics project, these days a robot can virtually not manage without at least one microcontroller to run it. So of course this raises the question of how best to select one, and this article is here to help you in this delicate task. Although the simplest robots can get by with virtually any type of microcontroller programmed in the language of your choice, as the complexity of the robot increases, it becomes clear that certain ICs are more suitable than others for a purely robotic application.**

The information they furnish is most often digital, from the simple on/off information of an open or closed switch to the complex NMEA frames from a GPS receiver. A few, less common sensors also furnish information in analogue form, and it's important not to overlook these ones.

So our robot's microcontroller must have numerous parallel port lines for on/off type information, but also asynchronous and synchronous serial interfaces (I<sup>2</sup>C, SPI, etc.) for sensors providing more complex information (electronic compasses, inclinometers, etc.), along with at least one analogue-to-digital converter for analogue information.

So far, everything we've been discussing is still within the scope of all current microcontrollers, and it's not much help to us in choosing. But the situation is actually more complicated than you might think from this discussion, which might be described as 'static'. When our robot is moving, there is a need to simultaneously control its motors, interpret the information provided by the sensors, and take the necessary decisions that entails. On the simplest robots with a small number of sensors, all this can be managed using standard sequential programming; but as soon as the number or complexity of the sensors increases, the situation soon becomes unmanageable. It then becomes necessary to resort to multitasking, i.e. to a mode of operation in which the microcontroller handles the sensors, the motors, and the decision-making all together and 'at the same time'. Sadly, not all microcontrollers or programming languages are able to manage this by a long way.

The last particular feature of robots is that, unlike conventional electronics projects,

they are often built by amateurs coming from backgrounds other than electronics. Mechanics, modellers, those who are simply curious all get involved in designing robots. For all these designers, who contribute a great deal to the world of robotics as they have a different view from the electronics enthusiasts, the microcontroller needs to be simple to implement and program. This simplicity sometimes founders on the reef of the multitasking we've just been talking about, but we're going to see that, by judiciously choosing the microcontroller, it is possible to reconcile the irreconcilable.

## Ordinary or special microcontrollers?

If electronics no longer holds any secrets for you and if you're not afraid of programming, you can obviously choose a standard microcontroller for your robot. PIC from Microchip, AVR from Atmel, etc. The list is long, especially as each manufacturer offers a wide range of ICs with a great variety of resources.

Hence from Microchip, the PIC18 family is gradually supplanting the PIC16 family that has been delighting amateurs for many years. These new ICs actually perform better, are more powerful, and hardly any dearer. As for the development tools, the unassailable MPLAB, increasingly user-friendly and of course still free, works just as well for either, so the transition is a gentle one. And if the power of the PIC18 isn't enough for you, the PIC24 family is all ready to replace it, as discussed in *Elektor* issue 343's presentation of the Explorer-16.

It's the same picture with Atmel, where

the AVR ICs from the ATmega range — scarce and expensive just a few years ago — are now within everyone's pocket, with their innumerable internal resources and sometimes impressive memory capacities. Here again, the AVR Studio development tool is free and available from the Atmel website.

In spite of all that, it's not these 'classic' ICs that we're going to be looking at — especially since this subject has already been covered in Elektor (issue 322), but some 'special' microcontrollers that are proving highly successful in robotics because of their easy implementation and the particular features of their programming languages. You're probably familiar with the oldest of them, none other than the famous Basic Stamp — but these days this is far from being the only one. Originally dreamt up by Parallax, this concept has had its imitators, and ever since we've seen numerous microcontrollers coming onto the market aspiring to be its descendants, while of course claiming to do much better. So these ICs are the ones we suggest you choose from.

### A forerunner that has aged well

For those of you who might not already know it, the Basic Stamp, developed and marketed in 1993 in the United States by Parallax, is a microcontroller that behaves as if it were directly programmable in Basic — but this particular feature is far from being the only one to have ensured its success. It is also a ready-to-use microcontroller, needing neither a clock crystal, an external reset circuit, nor even a stabilized supply to operate. All this is already built in.

Like any self-respecting microcontroller, the Basic Stamp has to be programmed, but this programming is done in Basic, easy to use and accessible to everyone, to the point it has almost become a standard upon which all its successors have been based. No programmer is required, as it only amounts to... a simple cable to link the Basic Stamp to the serial port of any PC, even an old or very basic model. The development tool, intended for program-writing, is completely free and available for download from the Parallax website.

Even though the simplest of all the Basic Stamps, the Basic Stamp 1, can be used to drive a robot, we unhesitatingly advise using at least the Basic Stamp 2, to benefit from the more numerous resources and a fuller instruction set. What's more, many successors to the Basic Stamp 2 (referred to from now on as BS2) are pin-compatible with it, allowing for possible future upgrading as a robot evolves, without needing to modify the associated electronics.

Figure 1 show both the physical appear-

**Table 1: Pinout for the Basic Stamp 2 in the 24-pin package, adopted by many of its competitors** (Basic Atom 24, Cubloc CB220, Javelin Stamp, among others).

Name	Pin no.	Function
S <sub>OUT</sub>	1	Programming output (PC serial port)
S <sub>IN</sub>	2	Programming input (PC serial port)
ATN	3	Programming input (PC serial port)
V <sub>SS</sub>	4 et 23	Ground
P0 ÷ P15	5 ÷ 20	Input/output ports
V <sub>DD</sub>	21	5 V stabilized output (input if VIN is too low)
RST	22	Manual external reset input (if required)
V <sub>IN</sub>	24	Unstabilized positive supply from 5–15 V (12 V for 2E, 2SX, and 2P24)

ance of the BS2 and its pinout, while the functions of the various signals available are listed in Table 1. Like all its successors, the Basic Stamp is not a 'true' integrated circuit, but a tiny PCB the size of a 24-pin DIL IC fitted with a number of SMD components, including a microcontroller, its clock and reset circuitry, an EEPROM memory for storing the program, and a 5 V regulator to power it.

Based on an 'old' PIC16C57 at 20 MHz, the BS2 is programmed directly in Basic, sometimes called PBasic, and is capable of executing around 4,000 instructions per second, while its memory can store around 500 lines of program. Its planetary success, and the word is not too strong, has pushed Parallax to put onto the market other BS2s, whose strong points are summed up here rapidly.

The first evolution to have seen the light of day, the Basic Stamp 2SX or BS2SX, is in fact a significantly faster version of the BS2. It executes on average 10,000 instructions per second, by replacing the BS2's microcontroller with a SX28 from Ubicom. The program memory is also larger, accepting around 4,000 lines of program. All this is of course transparent for the user and the BS2SX instruction set is identical the BS2's, apart from three new instructions to manage this extra memory.

The BS2SX and its relatively large program

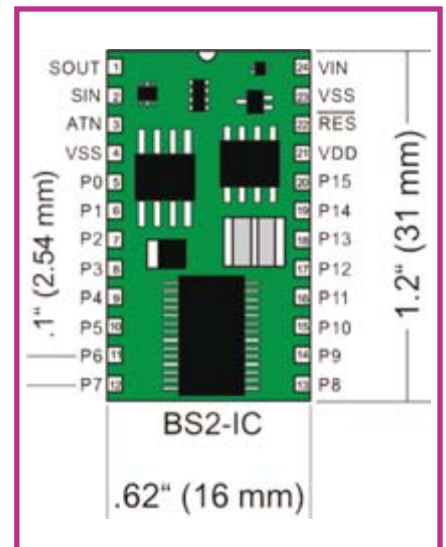


Figure 1. The physical appearance, dimensions, and pinout of the Basic Stamp 2 have been taken up by many of its competitors: Basic Atom 24, Javelin Stamp, Cubloc CB220 amongst others.

memory created a demand among certain Basic Stamp users wanting to benefit from this much larger memory, but not needing the speed (and hence the price-tag!) of the BS2SX. So Parallax has offered them the Basic Stamp 2E — simply a 'degraded' ver-

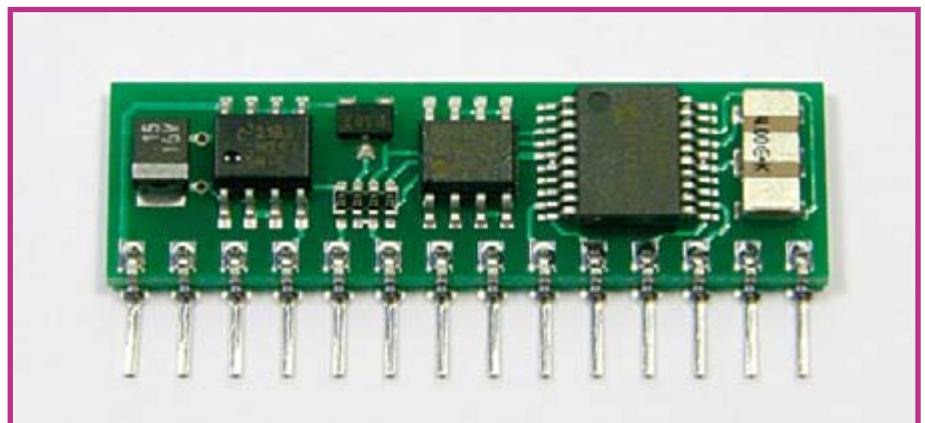


Photo 1. Though it's been a bit left behind now, here's the father of all the Basic Stamps, the Basic Stamp 1.

**Table 2: Principal characteristics of the various Basic Stamps and the Javelin Stamp.**

Parameter	Basic Stamp 2	Basic Stamp 2SX	Basic Stamp 2E	Basic Stamp 2P24	Basic Stamp 2P40	Javelin Stamp
Microcontroller	PIC16C57	SX28	SX28	SX48	SX48	SX48
Clock frequency	20 MHz	50 MHz	20 MHz	20 MHz	20 MHz	25 MHz
Program memory (bytes)	2 K	8 x 2 K	8 x 2 K	8 x 2 K	8 x 2 K	32 K
Program memory (instructions)	500	4 000	4 000	4 000	4 000	–
Working memory (bytes)	32	32	32	38	38	32 K
Scratch memory (bytes)	–	64	64	128	128	–
Speed (inst./sec.)	4 000	10 000	4 000	12 000	12 000	8 500
Number of Basic instructions	36	39	39	55	55	0 (Java)
Parallel inputs/outputs	16	16	16	16	32	16
Max. source/sink current per output	20/25 mA	30/30 mA	30/30 mA	30/30 mA	30/30 mA	30/30 mA
Max. source/sink current per chip / per group of 8 inputs/outputs	40/50	60/60	60/60	60/60	60/60	60/60
Programming interface	PC serial port 9,600 baud	PC serial port 9,600 baud	PC serial port 9,600 baud	PC serial port 9,600 baud	PC serial port 9,600 baud	PC serial port 28,800 baud
Supply voltage	5 - 15 V	5 - 12 V	5 - 12 V	5 - 12 V	5 - 12 V	5 - 24 V
Operating current	8 mA	60 mA	20 mA	40 mA	40 mA	80 mA
Stand-by current	100 µA	200 µA	100 µA	400 µA	400 µA	stand-by mode

sion, in terms of speed only, of the BS2SX. So, the Basic Stamp 2E has all the characteristics of the BS2SX, but offers the same program execution speed as the BS2.

The Basic Stamp 2P24s and 2P40s offer more innovations, but can be presented together as their characteristics are identical except for one detail we'll look at in a moment. Apart from being faster than the already fast BS2SX, achieving 12,000 instructions per second, they also have an extended instruction set. The 36 or 39 instructions of the BS2 or BS2SX increase to 55, introducing some very powerful and extremely handy instructions capable of directly driving an LCD alphanumeric display, talking to peripherals over the I<sup>2</sup>C bus, or driving ICs with a Dallas '1-Wire' bus. This evolution is done cleverly, however, and the same 36 BS2 instructions are included within the 55 instructions of the BS2P24 and BS2P40. The BS2P24 uses a pinout compatible with the other Basic Stamps, while the BS2P40 uses the 40-pin DIL IC format, allowing it to have 16 additional parallel port lines compared with the 24-pin packages.

To help guide your choice, **Table 2** summarizes the most important details of the various versions of Basic Stamp. Note that, for reasons of convenience, it includes the Javelin Stamp, described later.

All these Basic Stamps are wonderful in robots, as their instruction set has really

been designed for microcontroller-oriented use. So to make parallel port P2 go high, we simply write `HIGH P2`; to make it generate pulses we use the instruction `PULSIN`; to receive data in asynchronous serial form, we use `SERIN`, while to make it output synchronous serial data we use `SHIFTOUT`.

Because of the very simple and explicit syntax of these instructions, anybody can write programs for a Basic Stamp after just a few hours of practice and with no previous knowledge of programming. Moreover, given the seniority and success of the Basic Stamp, the library of programs available is immense. You only have to do a bit of Googling to see for yourself.

So in our opinion, the Basic Stamp is a good choice for someone wanting to make a start in robotics, even if it does present in our view two drawbacks, of unequal importance: it's still expensive compared to other similar ICs; and it doesn't support multitasking. This point must, however, be taken relatively, inasmuch that a number of microcontrollers that do support it are pin-compatible with the 24-pin Basic Stamps, allowing easy substitution in the event of your robot's evolving in this direction.

### Basic Stamp 'clones'

The success of the Basic Stamp has clearly made some people envious, and various

products have tried to imitate it, while seeking to overcome some of its shortcomings. At least two products fall into this category: the Basic Atom 24 from Basic Micro and the PICBasic range from Comfile Technology.

Based on a PIC16F876, the Basic Atom 24, pin-compatible with the 24-pin Basic Stamp 2s, has a program memory the same size as the BS2E. Faster overall, at around 33,000 instructions per second, it also offers more internal resources, including an ADC, two PWM ports, and up to a point supports interrupts. Its instruction set is also fuller than the 'classic' Basic Stamps, and is related to that of the Basic Stamp 2P, though is a little fuller because of the increased internal resources. So it may represent a worthwhile alternative to the Basic Stamp, especially since it's a little cheaper to buy.

On the downside, we must note all the same that it isn't multitasking either, and its availability leaves something to be desired, as the product has clearly not enjoyed the success its designers were hoping for (or else it just came on the scene too late). As it is not used a great deal, the library of programs for it is nothing like that of the Basic Stamp.

As for the PICBasic, it is, or rather was, an alternative to the Basic Stamp 2 designed by Korean company Comfile Technology. We won't talk about it here, as it's



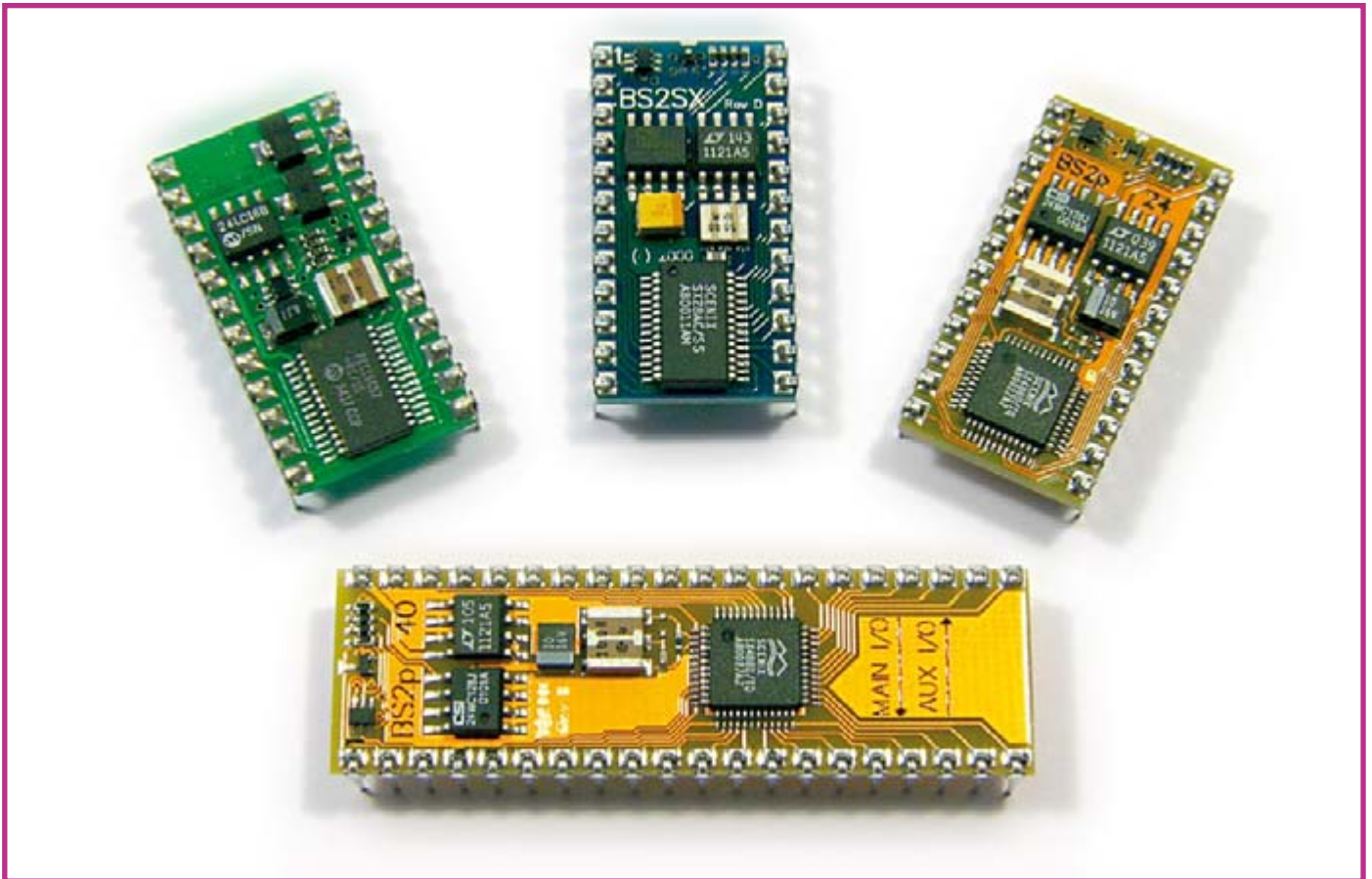


Photo 2. The Basic Stamp's offspring.

clearly on the road to extinction, if we are to believe Comfile's website, in favour of the Cubloc range from this same manufacturer. When you discover in a moment the possibilities of the Cubloc, and given that a CB220 (entry-level Cubloc) costs virtually the same as a PICBasic 2S, you'll easily understand why it's being dropped.

### A first step towards multitasking

Once again, it is Parallax who made the innovation in terms of multitasking with two distinct products for completely different purposes. The first, and also the oldest, is the Javelin Stamp, much less well known than the Basic Stamp. It has to be said that its price (around £ 45) might have something to do with this...

So the Javelin Stamp is physically like a Basic Stamp 2, but is programmed in Java. Of course, it isn't just that which makes it multitasking, but the fact that it has two operating modes: a foreground mode, where it executes the main program written in Java, and a background mode where a certain number of tasks can be performed independently of, and hence at the same time as, the main program.

These tasks are executed by means of virtual peripherals or VPs of which, as far as background mode is concerned, there are five: UART, PWM signal generator, 32-bit

timer, 1-bit ADC, and delta-sigma ADC. So, for example, the background generation of PWM signals proves very interesting for robotics applications, since many motors are controlled by signals of this type. So the Javelin Stamp can, for example, control a robot's motors and convert the analogue information coming from a sensor, while still continuing to execute its main program.

Independently of these specific features, the Javelin Stamp uses a Ubicom SX48 processor operating at 25 MHz, giving it a speed of 8,500 instructions per second; its other key characteristics are summarized in **Table 2**, to let you make a quick comparison with the Basic Stamps.

So the partially multitasking character of the Javelin Stamp does make it an interesting processor for robotics applications, but in our view it suffers from two drawbacks: its excessive price, compared with 'competing' processors; and the fact that it is programmed in Java, which is quite a difficult language to master for anyone who has never done any programming before.

### One microcontroller with two heads

The Cubloc range from Comfile Technology is much more innovative in the area of multitasking. As seen in **Figure 2**, showing

the Cubloc's internal structure, this microcontroller is in fact a twin one, including on the one hand, a processor programmed in Basic, and on the other, a processor controlled in Ladder. This language, which you may not have heard of, is none other than the language used for programmable automata. The Cubloc's twin processors can of course operate simultaneously, making the Cubloc truly multitasking — provided of course we program in both Basic and Ladder.

To make it easier to get to grips with, it is of course possible to only program it in one language or the other. If you choose Basic, the syntax is close to that of the Basic Stamp with, however, lots of new instructions making it easier to handle PWM signals, or drive an I<sup>2</sup>C bus, keyboard interfacing, etc.

Although the Cubloc range currently comprises four main models, whose key characteristics are summarized in **Table 3**, we recommend starting out with the CB220. Apart from being it's the cheapest of the Cublocs, it's also pin-compatible with the Basic Stamp 2, making it possible to develop a single-tasking application using the latter into a multitasking application, without needing to make the slightest modification to the electronics.

What's more, given that it's based on an ATmega128 at 18 MHz, it can execute around 36,000 instructions per second

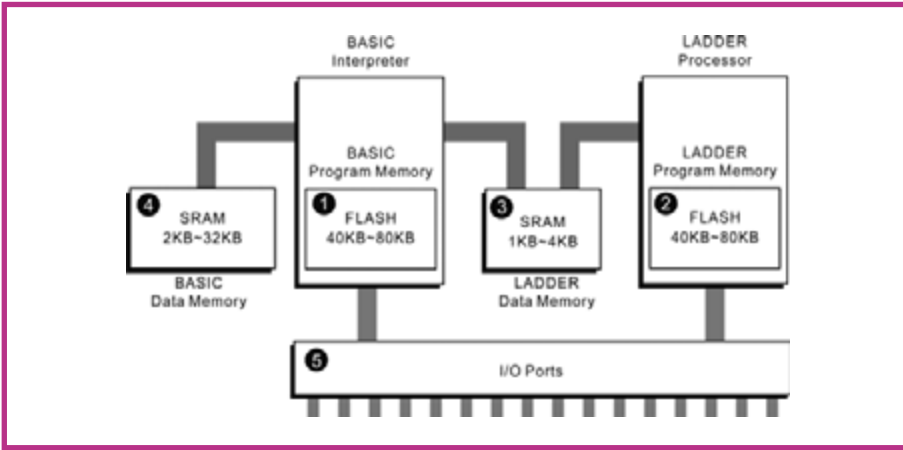


Figure 2. Internal architecture of the Cubloc CB220.

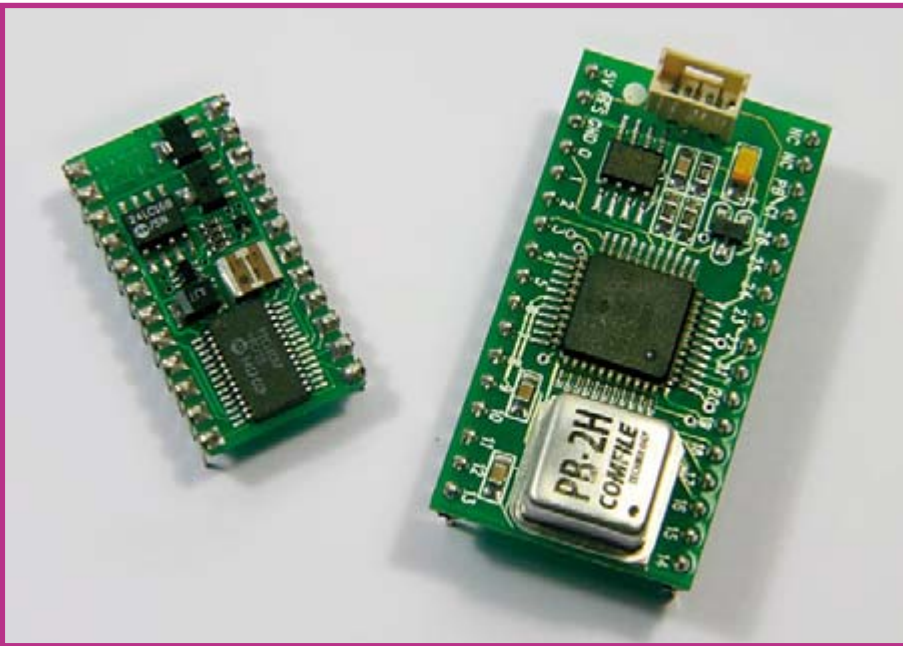


Photo 3. The PicBasic (right) marked Comfile Technology's entry into the world of these special microcontrollers. It wasn't yet pin-compatible with the Basic Stamp 2 (left).

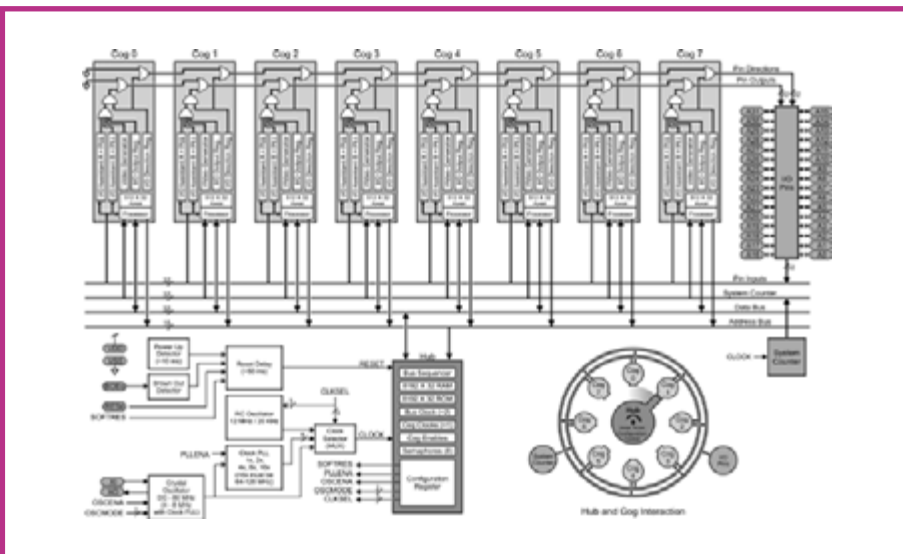


Figure 3. Internal architecture of the Propeller – impressive, and allowing true multitasking.

and its program memory offers a very comfortable capacity of 80 kB. Note too that it has an 8-channel 10-bit ADC and a 3-channel, 16-bit DAC capable of generating PWM signals.

So the Cubloc CB220 seems to us a good choice today in terms of processors intended for robotics, as it combines lots of advantages in a single package: hardware compatibility with the Basic Stamp 2, the simplicity of Basic programming, and the possibility of multitasking; all this for a price that is still reasonable compared with the other products in this survey.

### And finally, some true multitasking

Rest assured, Elektor readers, we have not been paid to write this article by Parallax, even though it's another of this company's products we're going to be talking about next: the Propeller. We have to admit that ever since the first Basic Stamp came out almost 15 years ago, Parallax hasn't been resting on its laurels. But while the previous products were based on existing processors, the Propeller is a real 'chip' developed by Parallax.

As **Figure 3** shows, even at first glance it's an impressive product, consisting of no less than eight independent functional blocks, the Cogs, each comprising its own processor and some RAM. These Cogs are linked via a bus managed by a Hub that takes care of synchronizing their exchanges. They all share the product's 32 uncommitted input/output lines.

Very few specialized peripherals are built in to the Propeller, but this isn't a problem, given that its programming language, called Spin, is in fact an object-oriented language. So if you need, for example, an RS 232 serial port, you only need to delve into the voluminous object library made available to you by Parallax to find what you're looking for there. You can do the same for generating PWM signals, driving an I<sup>2</sup>C bus, an LCD display, etc.

Given that there are eight Cogs available, it's possible to run up to eight different applications simultaneously. So a robot fitted with this sort of processor has no difficulty in managing its wheel motors, while analysing the data from several sensors and taking the relevant decisions.

Of course, Spin is a little daunting and writing your first instructions is a bit of a pain for anyone who's never programmed before — but it's well worth all the effort. What's more, Parallax places at your disposal a forum and above all a space on its website where anyone can upload the various object modules they have developed for the Propeller. This participatory library currently has over 75 object modules covering the most diverse fields, and

**Table 3: Principal characteristics of the various Cublocs.**

Parameter	CB220	CB280	CB290	CB405
Microcontroller	ATmega128	ATmega128	ATmega128	ATmega2560
Clock frequency	18.432 MHz	18.432 MHz	18.432 MHz	18.432 MHz
Program memory	80 K	80 K	80 K	200 K
Dynamic memory (RAM)	2 K (Basic) 1 K (Ladder)	2 K (Basic) 1 K (Ladder)	24 K (Basic) 4 K (Ladder)	51 K (Basic) 4 K (Ladder) 55 K (pile)
Data EEPROM	4 K	4 K	4 K	4 K
Speed (instr./s)	36,000	36,000	36,000	36,000
Inputs/outputs	16	49	91 (33 entrées, 32 sorties et 26 entrées/sorties)	64
Serial ports	1 RS-232 1 TTL	1 RS-232 1 TTL	1 RS-232 1 TTL	4 RS-232
ADC (10-bit)	8 channel	8 channel	8 channel	16 channel
DAC (16-bit, PWM)	3 channel	6 channel	6 channel	12 channel
External interrupts	–	4	4	4
Fast counters	2 x 32 bits	2 x 32 bits	2 x 32 bits	2 x 32 bits
Real-time clock	–	–	Oui	–
Supply voltage	5 - 12 V	5 V	5 V	5 V
Operating current	40 mA	40 mA	70 mA	50 mA
Package	24-pin DIL (BS2 compatible)	64-pin module	108-pin module	80-pin module

is continually growing.

So if you want to build a robot that requires true multitasking, the Propeller is currently one of the best solutions there is, all the more so because the price is only a quarter of that of a Basic Stamp, a Cubloc CB220, or similar product.

### An odd multitasking PIC

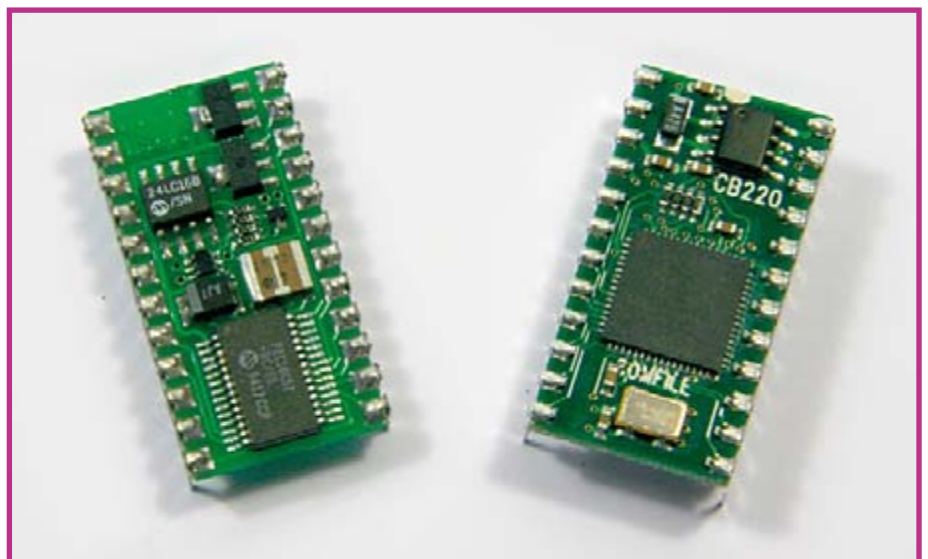
We couldn't end this review of special processors for robots without mentioning an IC that's relatively unknown on this side of the world, despite its undeniable interest: the OOPic. As its name might lead us to suppose, this IC is none other than a PIC microcontroller that can be programmed in object oriented language — 'OO' standing for 'Object Oriented' — but that's not all... The OOPic is in effect a PIC that you will be able to program in the language of your choice: Basic, C, or even Java — but this program will consist of simple scripts calling up objects. These objects, currently 130 of them, are capable of managing virtually anything you can imagine driving with a microcontroller, and, rather than list them here, we'll send you off to the OOPic website to discover them for yourselves. All these objects are able to operate simultaneously and independently of each other as background tasks while the main program is running. So with an OOPic, it's very easy to produce a multitasking application. But the concept goes further than that. It's possible to link objects together to form what are called virtual circuits that perform complete functions. In con-

crete terms, this means that objects can exchange data between them, as a background task, without your main program even having to bother about it.

Another advantage, and not the least, of the OOPic is the possibility of networking it; a maximum of 128 OOPics can be connected together to I<sup>2</sup>C ICs, thanks to a 3-wire bus and without any additional external components. So with the OOPic it's possible to create what is called semi-distributed robot architecture, i.e. to no longer do multitasking, but multiprocessor operation. One processor looks after the

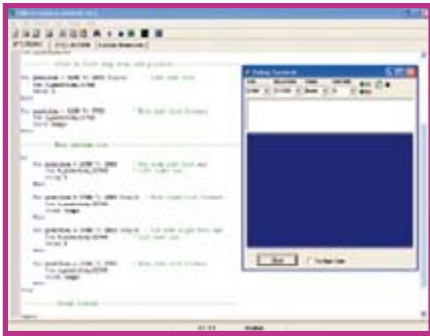
management of the propulsion, another this-or-that type of sensor, and so on, while a 'central' processor now only has to look after decision-making on the basis of the information sent to it.

Of course, implementing such an architecture is no longer within the grasp of a robotics novice, but after starting out with a Basic Stamp or a Cubloc, for example, it is still accessible to anyone who cares to take the trouble, especially since it needs no special investment, since the OOPic is programmed in the language of your choice and the objects are available to you free, and programming is done using



**Photo 4. The Cubloc CB220 (right) is directly interchangeable with the Basic Stamp 2 (left).**

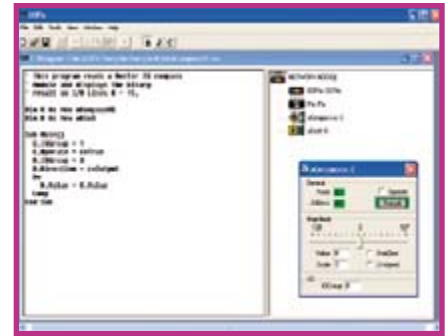




**Photo 5.** The development tools for Basic Stamp and Cubloc include a very useful editing terminal.



**Photo 6.** The Propeller development tool showing an example of Spin — not exactly a barrel of laughs, but the results are worth the effort!



**Photo 7.** The OOPic development tool allows graphical interaction with the objects used by the program.

a simple cable connected to the parallel port of any PC.

## Conclusion

This overview is of necessity incomplete, especially when you think that the programming manual of just one of the ICs presented here runs to at least a hundred or so pages! But we hope that we have helped you discover or re-discover some of the processors that are particularly well suited to robotics and which, while being simple to implement, do not in any way sacrifice performance.

(070319-1)

Addresses		
URL	Company	Comments
www.atmel.com	Atmel	Manufacturer of AVR microcontrollers
www.basicmicro	Basic Micro	Manufacturer of Basic Atom
www.comfile.co.kr/english2/	'PicBasic' website	Manufacturer of PicBasic
www.comfiletech.com	Comfile Technology General site	Manufacturer of Cubloc
www.microchip.com	Microchip	Manufacturer of PIC microcontrollers
www.oopic.com	Savage Innovations	Manufacturer of OOPic
www.parallax.com	Parallax	Manufacturer of: Basic Stamp, Javelin Stamp, Propeller
www.tavernier-c.com	-	Author's website

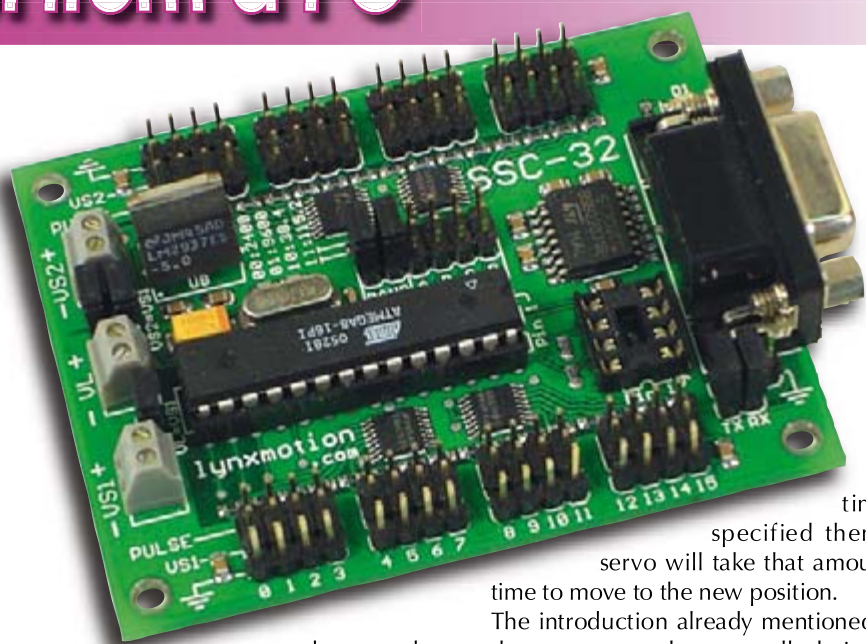
# Servo Control from a PC

Bas Lijten

Driving multiple servos via a microcontroller can be quite a problem for many people. Fortunately, a servo controller is available which can be controlled from a PC: the SSC-32.

The SSC-32 is an open-source controller which is connected to the serial port. It has 32 outputs, which makes it possible to control 32 servos simultaneously. Each output can also be used as a general purpose output with TTL levels. In addition, the microcontroller has four digital inputs and there is an extra socket for fitting an EEPROM. This was not being used at the time of writing, but it is likely that it will be used in future updates of the firmware.

The servos that are connected to the controller can be controlled in a very simple way. Only the numbers of the servos to be controlled, the pulse width (position) and the speed or time need to be known in



order to make a servo move. If a speed is specified then the servo will move to the required position at that speed. If a

time is specified then the servo will take that amount of time to move to the new position. The introduction already mentioned that the servos can be controlled simultaneously. This is possible by means of a 'Group Move' command. This is done by setting the numbers of the servos, the

pulse width and the time that the movement should take to reach the new position all in one command. In this way all servos move simultaneously.

This Group Move feature can be especially handy if, for example, you would like to make a robot arm move with a fluid motion. The controller carries out its own calculations, such as the speed at which the servos have to turn.

The controller also contains functions to drive a hexapod, a robot with 6 legs. In this way there is no need to come up with

an algorithm for the PC to let the robot walk, because the functions are already there to make the robot move its left or right side with a single command.

The code is freely available since the microcontroller contains open-source software. As a result you can add, improve and remove functionality yourself. Not happy with the hexapod code? You can then 'easily' rewrite it.

In the same way you can also add functionality. For example code to move a robot arm to a particular location in a Cartesian

coordinate system. By doing so there is no need to control individual servos from the PC but simply send one coordinate to move the arm the correct way.

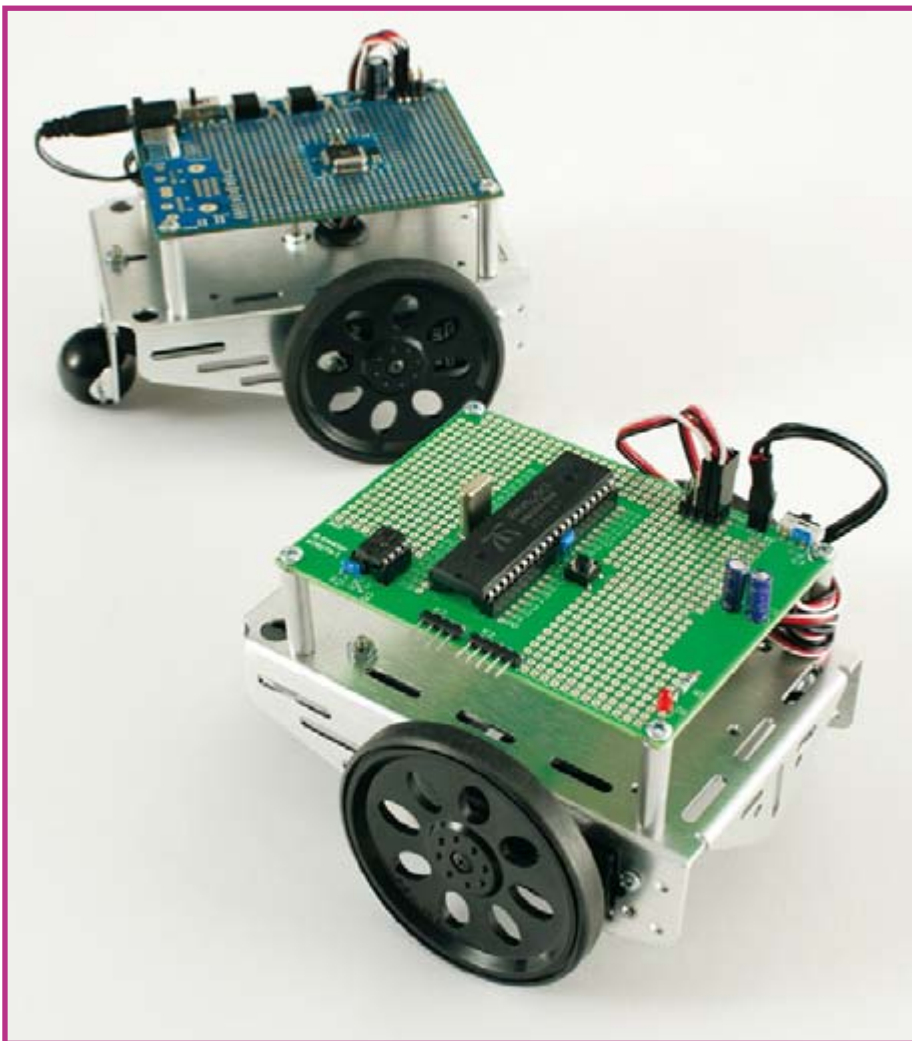
Because the controller is both easy to drive and easily modified by an experienced programmer it is very suitable for anyone who would like to spend some time with robotics.

(070373-1)

**Manufacturer of the controller:**

<http://www.lynxmotion.com>

# Propeller Prototyping Board for BoeBot



The Propeller chip made by Parallax Inc. is a bit of a strange animal in the world of microcontrollers. This IC consists of eight

32-bit processor cores that are given access one by one to the peripherals and processor memory. This makes the Propeller

extremely fast and it can work without the use of an interrupt mechanism: tasks that used to require an interrupt routine can now be run in their own processor core (called a 'COG' in Propeller-speak). This processor is fast enough to directly drive a VGA monitor for example, and also perform other tasks at the same time as well. We don't have enough space here to go into detail of the operation and the software for the Propeller. All this information and documentation, example applications and extensive development software can be freely downloaded from the Parallax website.

## BoeBot

As you may know, Parallax is also the manufacturer of the BoeBot robot kit, a frame with all the required hardware to make a mobile robot that can be equipped with various sensors. All this is controlled by a processor board using a Basic Stamp or Javelin Stamp. The BoeBot with the Basic Stamp was previously covered extensively in a series of articles in *Elektor Electronics*, but that was quite some time ago: end 1999 / early 2000 to be exact. But it's still going strong after all this time, since the BoeBot is even now widely used in education. That was the original purpose of this robot, since 'Boe' stands for 'Board of Education'.

## Upgrade

It was only a matter of time before this robot was upgraded with a processor board for the Propeller chip. Parallax has recently brought out a prototyping board

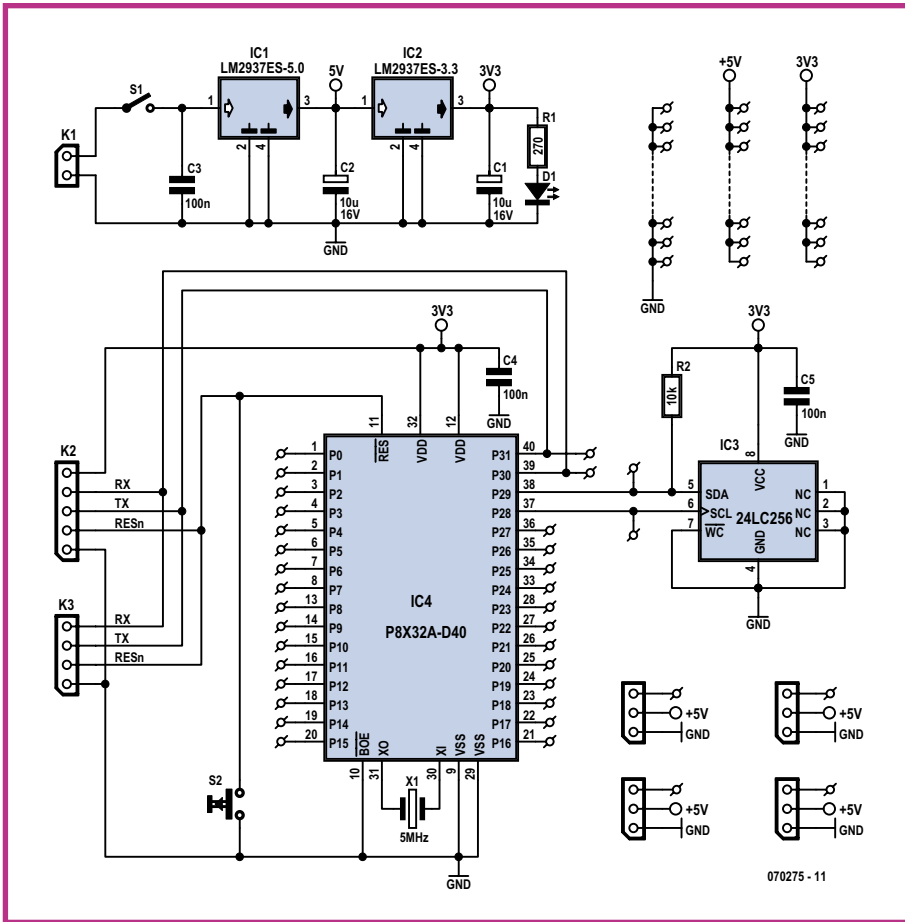


Figure 1: The circuit diagram of the Propeller prototyping board.

that is made to measure for the BoeBot frame. It is noticeable that part of this board now has an ordinary prototyping area for the addition of extra electronics,

whereas the 'old' Basic Stamp board came equipped with a mini breadboard. On the Propeller board we therefore have to solder any extra components, which is a bit more

time consuming than simply inserting them into a breadboard, but it does make the robot more reliable. Despite being carefully constructed and having well thought out software a robot can still bump into something or become a victim of a passing pet who suddenly discovers a new playmate. When a breadboard is used, some components could become dislodged; with a prototyping board the chances of survival are greater.

The prototyping board is well laid out, has everything you need and is reasonably priced, but we did see a potential disadvantage (especially in education): both the processor as well as the EEPROM are SMD versions. If something is wrongly connected it could mean the end of the processor and in many cases also the end of the board. Not everybody will have the right soldering equipment to replace such parts.

Parallax had no objections when we asked if we could design a version of the board for use with classic DIL ICs. Should something go wrong with one of the ICs on this board it's just a question of simply placing a new chip into a socket (after first finding out what caused the problems, of course!).

These components obviously take up more room of the Boe-Bot frame than their surface mount counterparts. The DIL version of the processor in particular is much larger than its little brother in a LQFP package. Because of this we have left out a few features that were present on the Parallax board, such as the combined VGA, keyboard and mouse connector, since we felt that they're unlikely to be missed in a

## Components list

### Resistors

R1 = 270Ω  
R2 = 10kΩ

### Capacitors

C1, C2 = 10μF 16V radial  
C3, C4, C5 = 100nF

### Semiconductors

IC1 = LM2937ES-5.0  
IC2 = LM2937ES-3.3  
IC3 = 24LC256  
IC4 = Propeller chip P8X32A-D40 (Parallax)

### Miscellaneous

X1 = 5MHz quartz crystal  
K2 = 5-way SIL pinheader  
K3 = 4-way SIL pinheader  
S1 = on/off switch  
S2 = 1 make contact  
4 pcs 3-way SIL pinheader

PCB, order code **070275-1** from Elektor SHOP

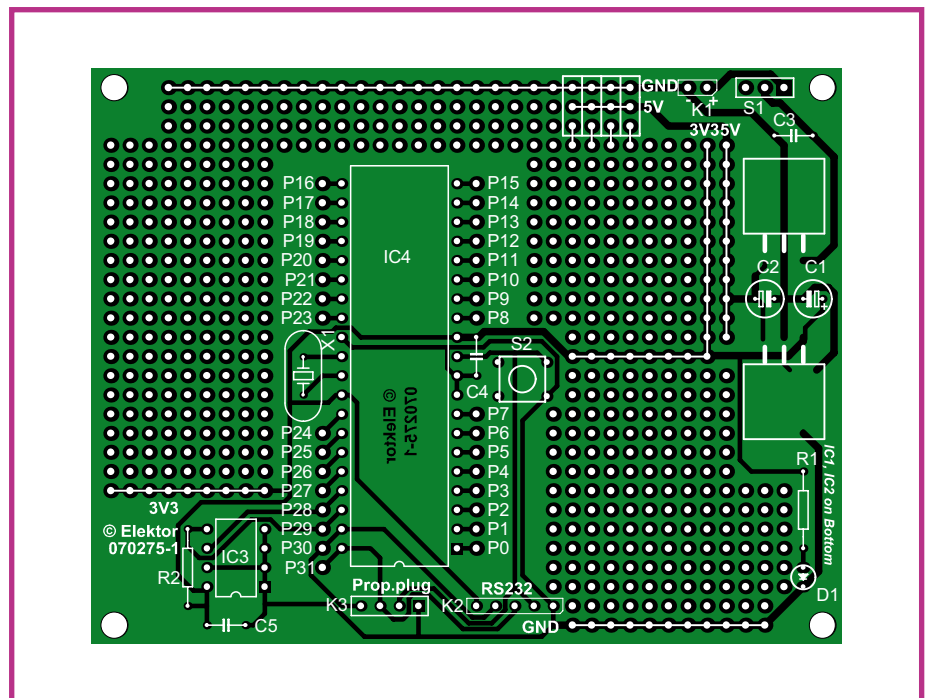


Figure 2: The board layout is single-sided. If you expect to use the prototyping area a lot then a double-sided through-hole plated version would be preferable.



mobile robot.

## Circuit diagram

The circuit diagram (shown in **Figure 1**) is conspicuous by its simplicity. Two low dropout voltage regulators provide 3.3 V for the processor and the EEPROM, and 5 V for peripheral devices that require a higher voltage, such as the modified servo motors that propel the BoeBot. Remember that the input pins of the Propeller can't withstand 5 V. Next to S1, the power switch, is a group of four three-pole SIL connectors for the connection of servos and other peripherals that require a 5 V supply.

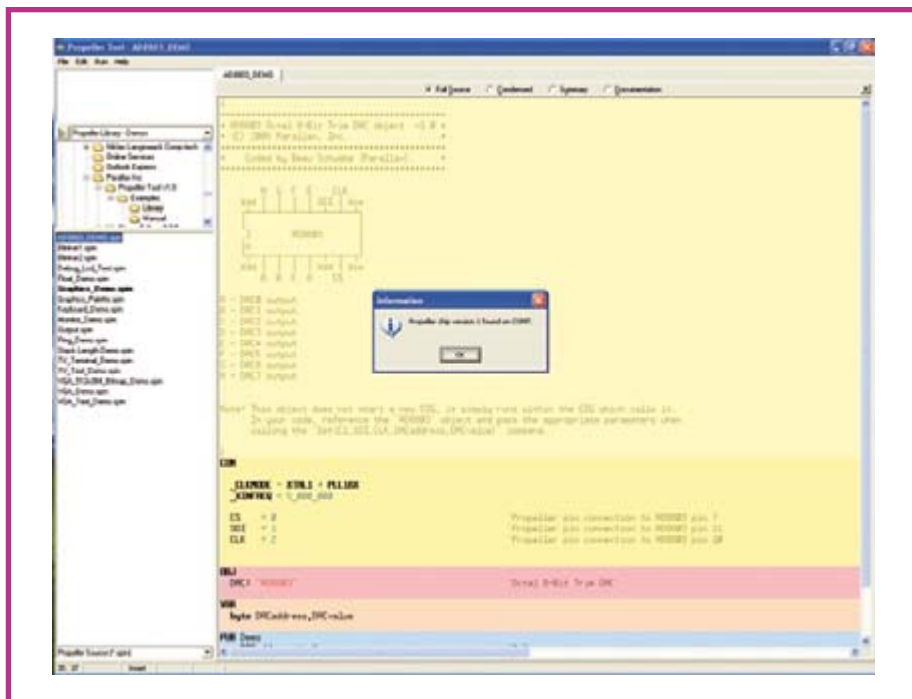
A power source with a voltage between 5.5 V to 26 V should be connected to K1, but we would advise against using too high a voltage because of the heat dissipation in IC1.

There are two different ways of implementing the programming interface to a PC: K3 is the connector for use with the Propeller Plug by Parallax, which uses a USB link. K2 is used for the simple serial interface that is described elsewhere in this issue. D1 indicates that the supply voltage is present, and S2 is the reset switch.

## Practical side

The PCB is also very straightforward (see **Figure 2**). We have intentionally chosen a single-sided layout because this makes it easier to etch it yourself. Both voltage regulators are soldered on the bottom of the board.

IC3 isn't strictly required to start using the Propeller. When the processor starts it runs its bootloader routine, which first checks if there is a communications link to the host PC and then waits for a program to be downloaded. The user then has the choice of either loading the program into the internal program memory of the Propeller or into the external EEPROM.



**Figure 3:** This message confirms that the Propeller Tool has made a connection and the fun can begin!

If no communications with a PC are possible, the bootloader will attempt to load a program from the EEPROM; if nothing is found there either, the bootloader stops and the processor turns itself off automatically.

A program can be directly transferred from the PC into the internal program memory and then executed, but remember that this memory is volatile. When the power is turned off, all memory contents are lost.

The crystal can be left out as well, since the Propeller initially uses its internal RC oscillator that runs at a speed of 12 MHz, which is fast enough for most applications. X1 will only be used once the program has set the relevant clock registers.

Once power is applied to the processor

and the circuit is connected to the PC, the 'Propeller Tool' should be started and the F7 key pressed (or from the *Run* menu choose the *Identify Hardware* option). The serial ports of the PC are then scanned one by one for the presence of a connection to the Propeller board and if everything is in order a message will appear like the one shown in **Figure 3**. The COM port number will obviously depend on which port the interface is connected to.

Once this message has appeared we can get to work with the Propeller and explore the exciting world of this microcontroller.

(070275-1)

## Web Link:

[www.parallax.com/propeller](http://www.parallax.com/propeller)

# 'TEAclipper' Postage-stamp Programmer

Richard Hoptroff

The art of discrete electronics has, over the past decade or so, become subsumed by machine code inside microcontrollers. The firmware is the magic in today's electronic circuits, and rightly so. It's faster, cheaper,

easier and more flexible than making changes to the hardware.

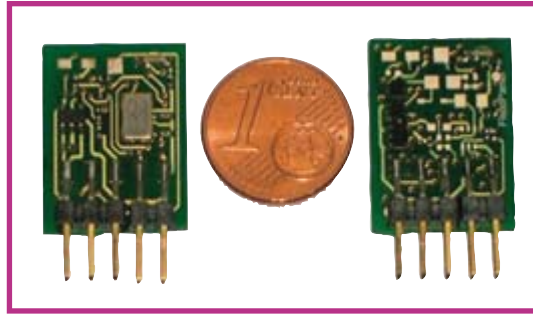
Unfortunately, exchanging firmware between people remains in the Dark Ages. If you want to buy someone else's firmware, what can you do? At best you buy a pre-programmed chip from the

creator or from a publishing service such as Elektor SHOP or [www.hexwax.com](http://www.hexwax.com). Worst case, you get a hex file and program the microcontroller yourself – provided you have all the necessary equipment to do so. And if there's a bug in the firmware, it's not exactly easy to get an

upgrade.

If only firmware could be more like software. Software is so easy to deliver that we do it without thinking, and as such has generated one of the most profitable industries in the world.

Having the size of a postage stamp, the TEAclipper from FlexiPanel Ltd ([www.flexipanel.com](http://www.flexipanel.com)) seems a step in the right direction. This microcontroller programmer is an easy, reliable firmware delivery mechanism. It can be pre-loaded with firmware and mailed to a customer, or the customer can download firmware over the internet and send it to the TEAclipper via a USB adapter. The



TEAclipper is then inserted into the target PCB and generates all the signals necessary to program the microcontroller. Connection is via a 5-pin header which also provides power to the TEAclipper.

Since only a temporary connection is required, a socket is not necessarily needed. The pins can be pressed against plated-through holes in the PCB for the few seconds required for programming.

The number of programming cycles can be specified, after which the memory self-erases. This allows firmware to be bought and sold in fixed quantities.

TEAclippers are currently available for programming Parallax's BASIC Stamps and Microchip's PIC Microcontrollers, but support for further microcontroller platforms is planned.

(070117-1)

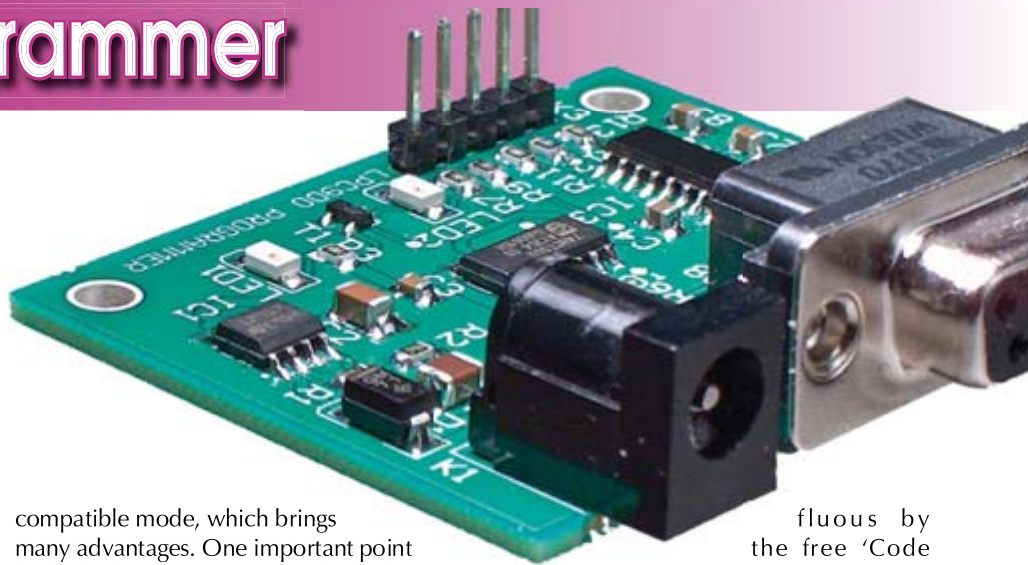
## LPC900 programmer

Jürgen Wickenhäuser

The LPC900 family is the Swiss Army knife of 8051-compatible microcontrollers. The 'LPC' in the part number stands for 'low pin count': the NXP (formerly Philips) LPC900 family [1] consists of a range of small and easy-to-use microcontrollers ideal for small-scale high-speed applications. Since the LPC900 family is based on an 8051 core it is easy to learn how to use the devices. However, the LPC900 is more than just a slightly spruced-up version of the 8051. The most important features are as follows:

- 2-cycle high-speed 8051 core (six times as fast as a standard 8051);
- from 1kB to 16 kB of flash memory with full ISP and IAP functionality;
- internal precision 7.3728MHz RC oscillator, ideal for baud rate generation up to 115 kbaud without an external crystal;
- CPUs available in DIL as well as tiny SMD (TSSOP) packages;
- a minimum system requires a single capacitor as the only external component;
- wide range of on-chip peripherals: brown-out detector, watchdog timer, comparators, A/D converter;
- operating voltage 2.4 V to 3.6 V.

The only significant difference from the standard 8051 is the improved I/O structure: they can now also work in a CMOS-



compatible mode, which brings many advantages. One important point to note in this regard is that on reset the ports are set to CMOS input mode, and must if necessary be suitably initialised before use.

Otherwise the LPC900 is very easy to use. The datasheet is rendered almost super-

fluous by the free 'Code Architect' software by Embedded Systems Academy [2]. This tool is capable of creating snippets of C source code directly (see **Figure 1**).

Loading code into the LPC900 microcon-

### COMPONENTS LIST

#### Resistors

- R1 = 240Ω SMD (0805)
- R2 = 390Ω SMD (0805)
- R3,R4,R7,R9,R11,R12,R13 = 1kΩ SMD (0805)
- R5,R6,R8,R10 = 220kΩ SMD (0805)

#### Capacitors

- C1 = 4μF7 25V SMD (1206 or 1210)
- C2 = 10μF 6V SMD (1206 or 1210)
- C3 = 100nF SMD (0805)
- C4,C5,C6,C7,C8 = 1μF SMD (0805)

#### Semiconductors

- D1 = GF1M

- T1 = BC857 SMD (SOT23)
- LED1,LED2 = LED, red, SMD (0805)
- IC1 = LM317 SMD (SO8)
- IC2 = 74HCT00 SMD (SO14)
- IC3 = MAX3232 (SO16)

#### Miscellaneous

- K1 = mains adaptor socket for PCB mounting
- K2 = 9-way sub-D socket, angled, PCB mount
- K3 = 5-way SIL pinheader
- PCB, ref. 070084-1, from [www.thepcbshop.com](http://www.thepcbshop.com)

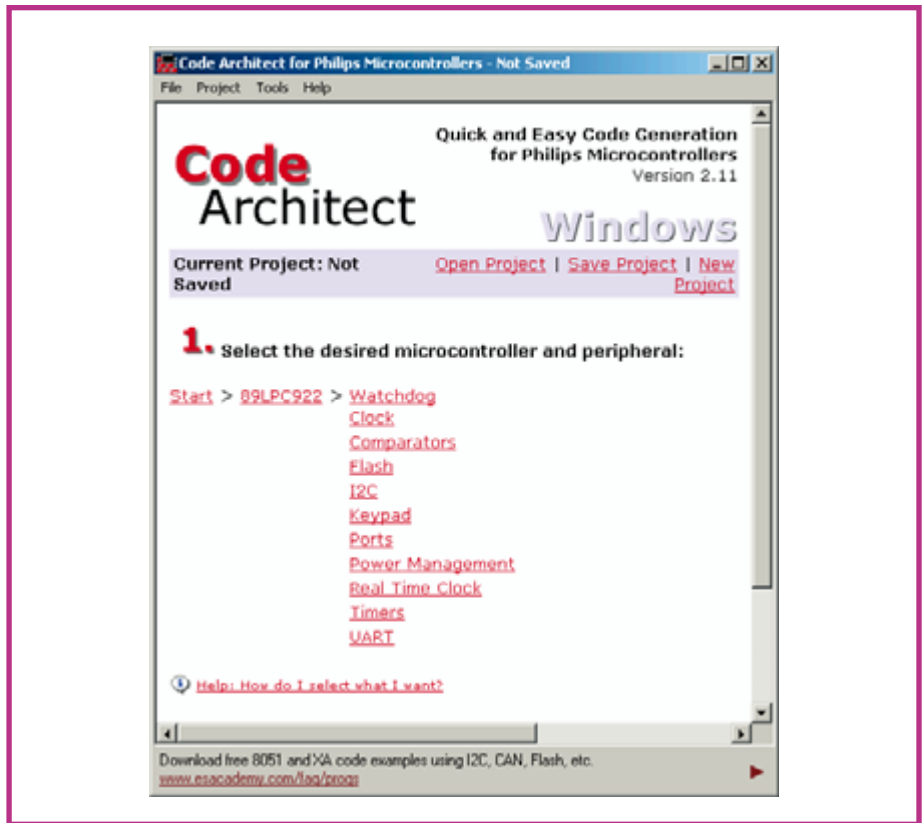
troller proceeds with a minimum of fuss. When the CPU receives a series of three pulses on its reset pin within a specified time window of being powered up, the CPU jumps to a bootloader, which then proceeds to communicate using the RXD and TXD signals. After code is downloaded these three signals are of course available for their normal use. A five-way cable is required for download.

The hardware required is minimal, as the circuit in **Figure 2** shows. This circuit is capable of supplying the LPC with power and also provides an RS-232 interface, ideal for use with a terminal program. The printed circuit board for the design is shown in **Figure 3**.

Operation is straightforward. The CPU can be powered up and down using the DTR signal. The RTS signal functions as a switch: if RTS is active then signals from RXD are also presented to the reset pin of the CPU, and can thus be used to activate the bootloader. Note that R13 ensures that the supply voltage to the target hardware drops rapidly when it is switched off. It is advisable to limit the capacitance on the 3.3 V rail in the target hardware to approximately 10  $\mu$ F, since otherwise the microcontroller being programmed might not correctly execute a power-on reset.

The free program 'Flash Magic', also from Embedded Systems Academy, can be used to simplify programming the devices. This will be familiar to *Elektor Electronics* readers from the RFID reader project [3]. This tool also includes a suitable terminal program.

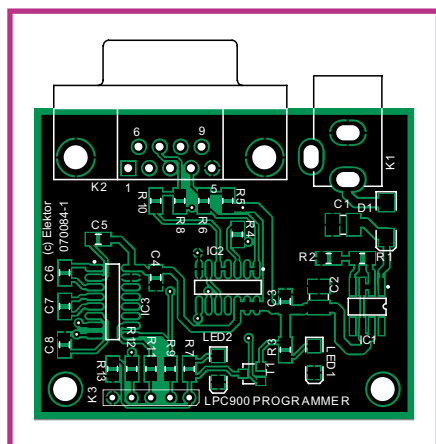
An optimising C compiler is also needed to write programs for the LPC900 family. There are various commercial products available; here we will briefly discuss how to use  $\mu$ C/51 [4]. For practically all LPC900



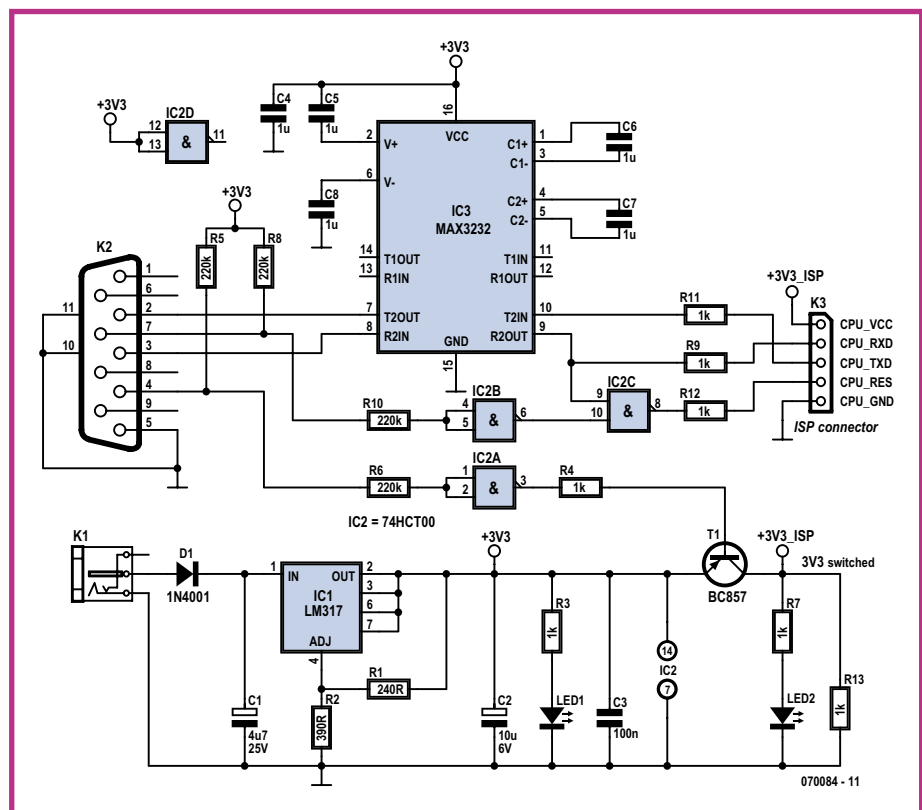
**Figure 1.** 'Code Architect' is a free tool that makes using the LPC900 microcontroller very straightforward.

family devices the free demonstration version of the compiler is entirely adequate. The only restriction of the free version is

the 8 kB code size limit, but this is already enough to write very complex applications for an 8051-compatible microcontroller,



**Figure 3.** Layout and component mounting plan for the printed circuit board.



**Figure 2.** Circuit of the LPC900 programmer.



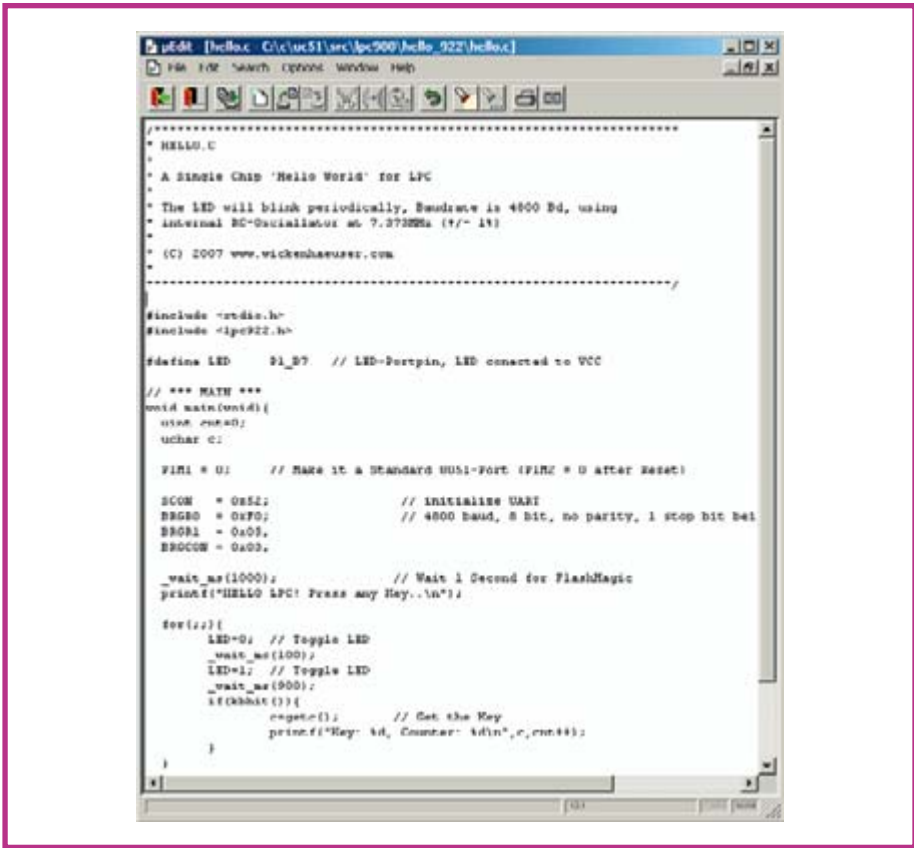


Figure 4. Initialisation specific to the LPC900 requires just five instructions.

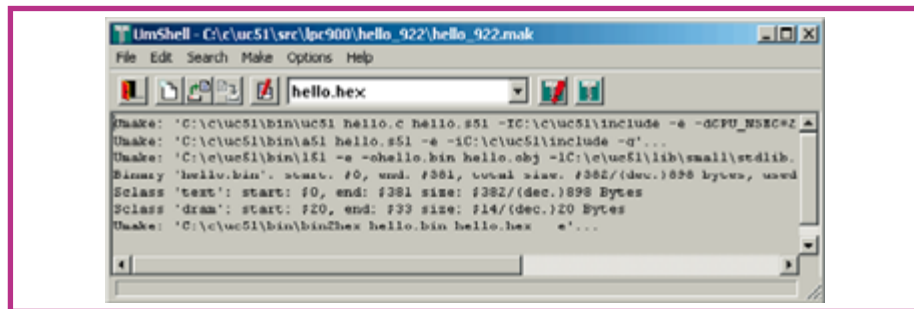


Figure 5. The µC/51 compiler uses a well-structured classical 'make' system.

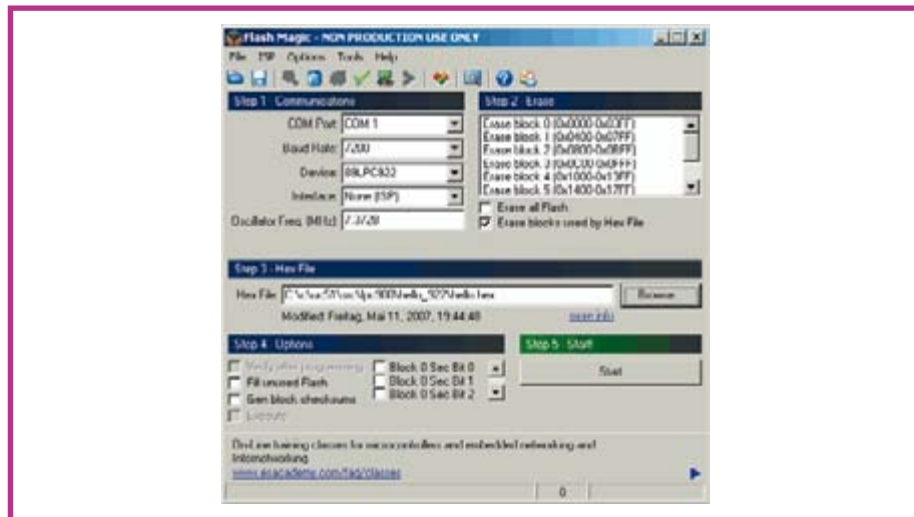


Figure 6. Flash Magic is a free tool for programming LPC microcontrollers.

because the compiler produces very compact code. In particular, the compiler is very parsimonious with the limited internal RAM in the microcontroller, using a graph-based optimisation algorithm to enable multiple re-use of memory areas. For example, even with this very tiny CPU we have a fully-featured 'printf()' function call. The system also includes a reliable and comprehensive floating-point library. The µC/51 system was developed as a tool for the company GeoPrecision [5], and has been used and maintained there for years.

The software tools work very well together and make development very quick: there are just three steps from source code to working program.

For demonstration purposes we used an 89LPC922 with an LED connected to port pin P1.7. The listing shown (Figure 4) is an example project included with µC/51 since version 1.20.06. As can be seen from Figure 5, the compiler uses a well-structured classical 'make' system. Downloading a program to the microcontroller is an intuitive operation (Figure 6). It is recommended that you configure the terminal program embedded within 'Flash Magic' so that it is launched immediately the application is started on the LPC. Note in particular that the RTS and DTR signals must be correctly configured (both active): see Figure 7.

(070084-1)

### References and links

- 1) <http://www.standardics.nxp.com/products/lpc900> (NXP, manufacturer of the LPC900 family).
- 2) <http://www.esacademy.com> (Embedded Systems Academy: Flash Magic, Code Architect).
- 3) ELEKTOR RFID Reader (using an LPC936), Elektor Electronics, September 2006, p. 26.
- 4) <http://www.wickenhaeuser.de> (µC/51 compiler, demonstration version).
- 5) <http://www.geo-precision.com> (geotechnical research and development).



Figure 7. Settings for the RTS and DTR signals in the Flash Magic terminal program.

# Low<sup>2</sup> Cost USB Demo Board

## C your way through USB

Martin Valle

Slick graphics, reading analogue values and making them appear in graphs, cockpit-style meters and dials, all on USB... *How do they do it?*

- USB Quick & Easy**

  - 2 analogue inputs (1 on pot)
  - 2 digital inputs (switches)
  - 4 digital outputs (LEDs)
  - 1 (one) IC: PIC18F4550 (available ready-programmed)
  - Full suite of software building blocks
  - Project software for Builder C++ 6
  - Demo program for 2-channel A/D with graph/meter readout on PC

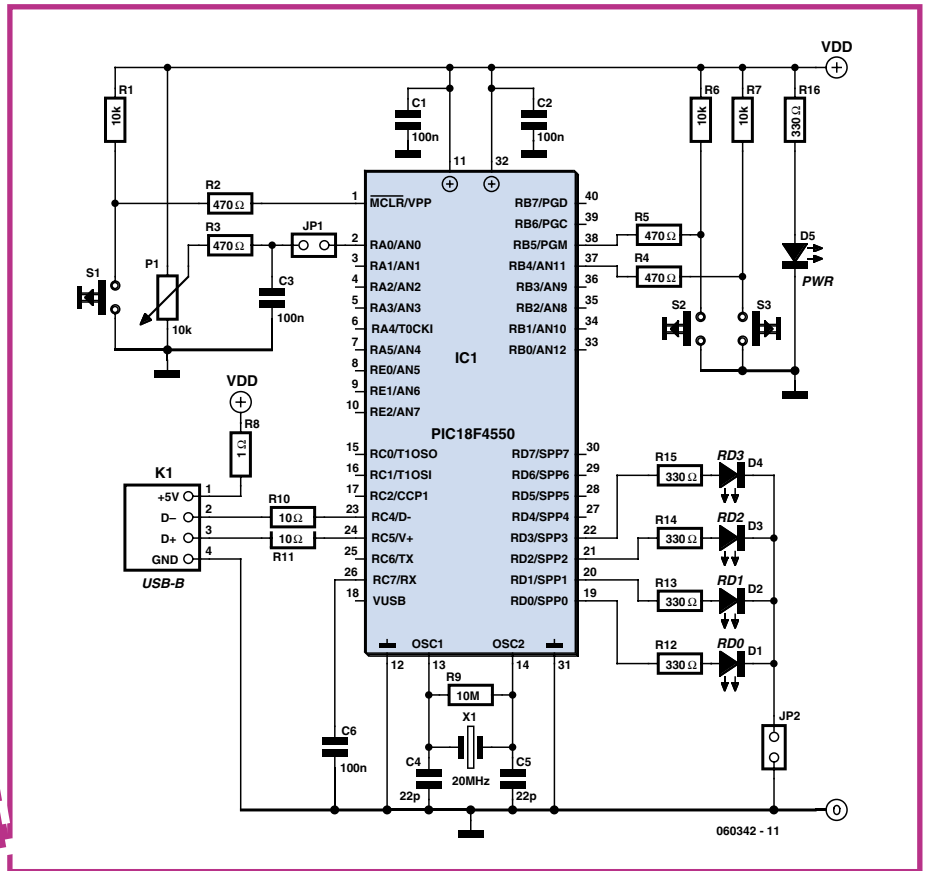


Figure 1. Schematic of the USB demo board.

Like this: a single board connected up to the PC via a USB link and sporting an advanced PIC microcontroller. Add some software and there's your USB demo board doubling as a development system to help you discover how USB is implemented on a microcontroller programmed to handle analogue and digital I/O for real world applications.

free of charge from the Elektor website as archive # **060342-11.zip**. The ready-programmed PIC18F4550 for the project is available too, it's item **060342-41** from the Elektor SHOP.

There are actually four zipped files: **MCHPFSUSB.zip** contains all the project components freely available from Microchip plus the custom project for the microcontroller, the demo version software and

## Hardware

At the heart of the circuit in **Figure 1** is a PIC18F4550 microcontroller from Microchip. This has built-in USB connectivity — if you know how to activate it! The micro is clocked at 20 MHz by quartz crystal X1. The switches (except S1), LEDs (except D5) and the potentiometer connected up to the PIC micro are your basic I/O (input/output) devices.

The circuit should be easy to build on a piece of prototyping board or Veroboard (a.k.a. perfboard or stripboard).

## Software

In stark contrast with the minimal hardware, the software for this project is quite extensive. You will like to hear that it's available

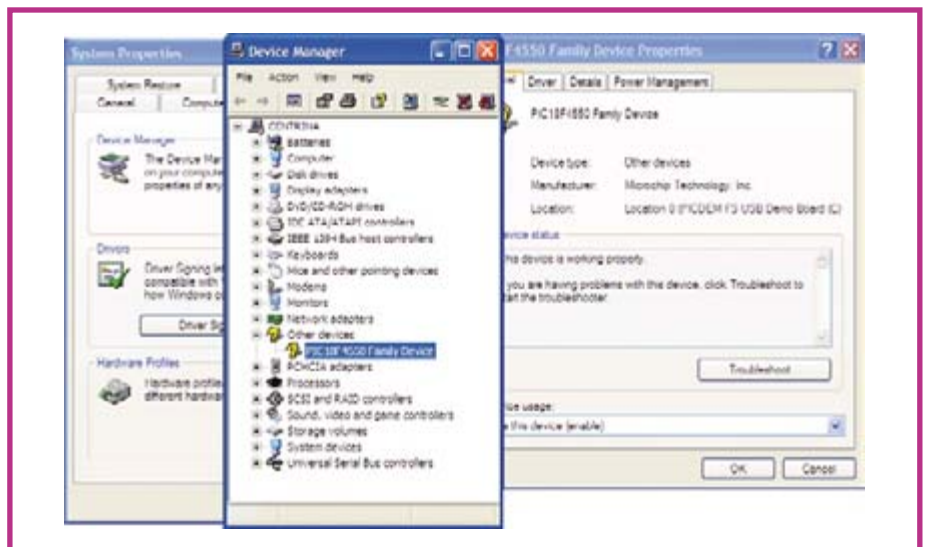


Figure 2. Device Manager showing the installation of the PICDEM FS USB Demo Board.

the bootloader 'talker'.

**Project.zip** contains all the files needed to build the project in Builder C++ 6.

**REQUIRED.zip** contains the files of the *project.zip* that you have to copy if you want to make a new project.

**Without builder.zip** contains all the files needed to run the .exe file without the Builder installed in the computer.

Here's how to process the software.

1. Extract the file *MCHPFSUSB.zip* to the C: root directory. Check that there is no duplicated folder MCHPFSUSB, like: C:\MCHPFSUSB\MCHPFSUSB\folders\_xx, instead of: C:\MCHPFSUSB\folders\_xx.
2. Using a suitable programmer, program the PIC18F4550 microcontroller with the file: C:\MCHPFSUSB\fw\\_factory\_hex\picdemfsusb.hex. The chip is also available ready-programmed.

## Waking up the F4550

Once you have built the project and checked it for mistakes, you can connect the USB to the host. For the first time connection, Windows XP is recommended. As soon as you connect the board to the host, LEDs D0 and D1 will start to blink, then the host will detect the device as 'PICDEM FS USB Demo Board (C) 2004' and ask for the drivers.

You should select the drivers located in: C:\MCHPFSUSB\Pc\MCHPFSUSB Driver\Release

Windows will nag that this device does not pass the Windows logo test. Ignore and simply continue the installation of the device.

You can check for the proper installation by exploring in the Device Manager window, it should look like **Figure 2**.

Keep pushbutton S3 (RB4) pressed and the reset the microcontroller by pressing and releasing S1. This procedure takes the microcontroller into 'bootloader' mode. The host will detect a new device, and it is necessary to repeat the driver installation procedure with the same driver location:

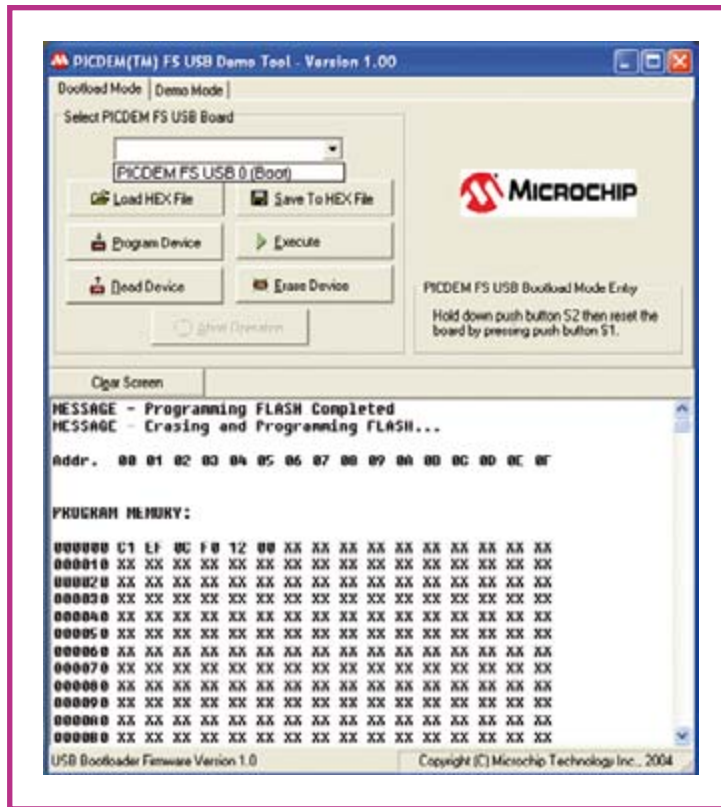


Figure 3. PICDEM FS USB tool showing the board (device) connected.

C:\MCHPFSUSB\Pc\MCHPFSUSB Driver\Release\

From now on you can load new programs

mode is shown in **Figure 4**. It allows reading an approximation of the position of the potentiometer connected to RA0, control the state of the LEDs connected to RD2 and RD3 and measure a temperature of an SPI sensor that's not actually connected to the board (so don't pay attention to the temperature readings).

The Demo firmware uses a few pins to monitor the USB main voltage (this is not implemented in the schematic). It may happen that this check fails after connecting the device to the computer. Fortunately, the error applies to the **Demo firmware only**, not to the Bootloader. There is an easy way to avoid this — in every project contained in the MCHPFSUSB folder there is a file C:\MCHPFSUSB\fw\project\_name\_folder\autofiles\usbcfg.h. It contains the declarations that are causing the problem. Just comment-out (//) the two SENSE\_IO definitions as shown **Listing 1**.

Having done this you can reload the Demo project, or any project for that matter, and all should work fine.

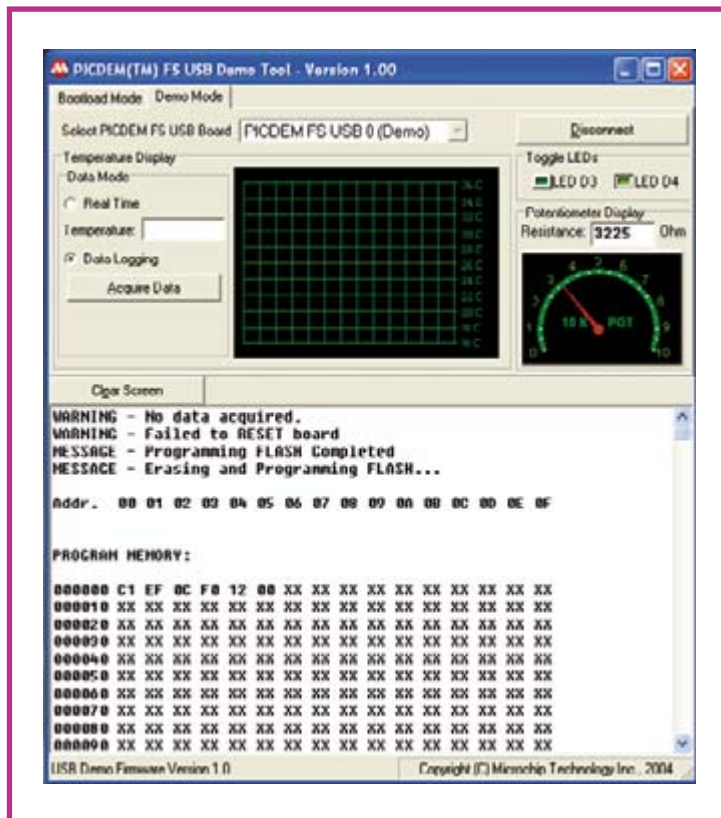


Figure 4. Demo Mode of the Pdfsub tool.



## Build it!

Now, it is time to explore the example application. As before, reset the microcontroller with S3 pressed, to run the Bootloader and load the micro with the hex file located in:

C:\MCHPFSUSB\fw\Hid02\\_output\  
NICAPM.hex

Then, run the program.

You can run the PC application directly without C++ Builder 6 installed in the computer by using the stuff in the archive file called: *without\_builder.zip*. Obviously the microcontroller must be connected to the host and running the NICAPM firmware. The window of the demo application should look like **Figure 5**, showing the signals of Channel 0 (AN0, pin 2) and Channel 1 (AN1, pin 3). There's also the archive file called *project.zip* if you want to modify something in the PC project.

If you want to make a new Builder project, you only have to look after two things:

1. Add the hid.lib file by doing:  
Project → Add to project → hid.lib
2. Change the Data alignment from Quad word to byte by doing:  
Project → Options → Advanced  
Options → Data alignment.

(060342-1)

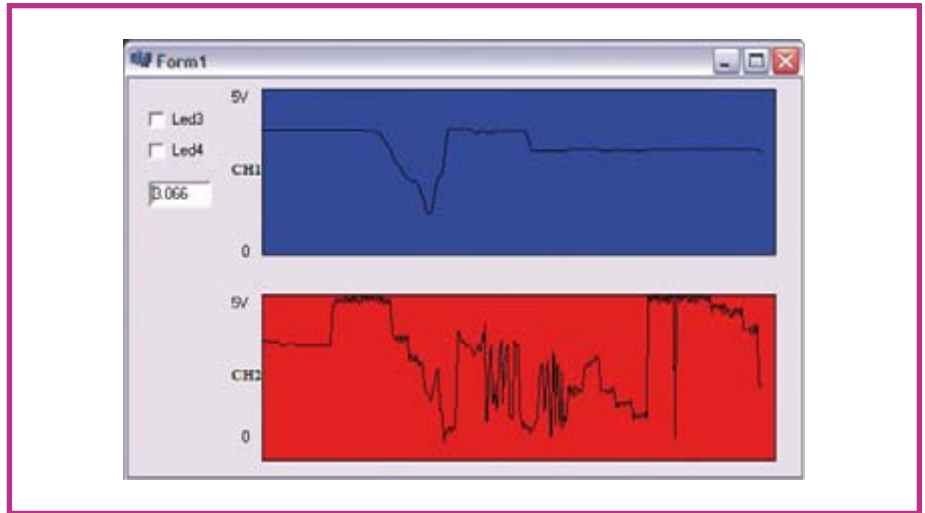


Figure 5. Two ADC Channels and USB of the PIC in action.

## Listing 1.

```

/** D E F I N I T I O N S *****/
#define EP0_BUFF_SIZE      8    // 8, 16, 32, or 64
#define MAX_NUM_INT       1    // For tracking Alternate Setting

/* Parameter definitions are defined in usbdrv.h */
#define MODE_PP            _PPBMO
#define UCFG_VAL           _PUEN|_TRINT|_FS|MODE_PP

// #define USE_SELF_POWER_SENSE_IO
// #define USE_USB_BUS_SENSE_IO
    
```

# Optimised STK200/300 Programmer

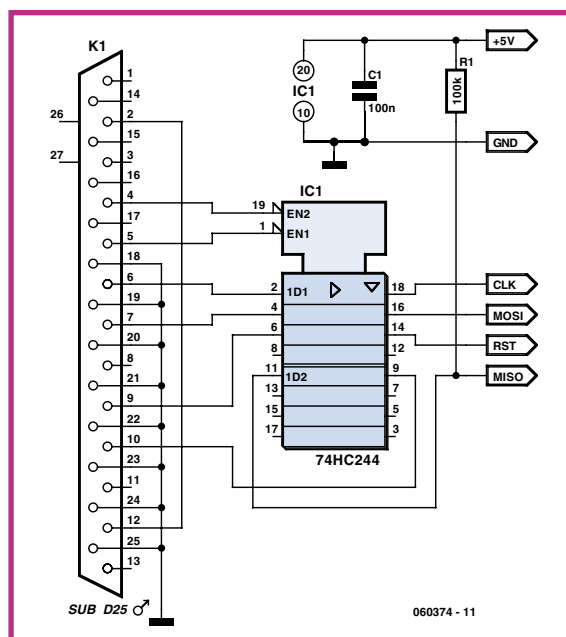
## for AVR Micros

Hesam Moshiri

The STK200/300 programmer is found in nearly every programmer software for Atmel AVR microcontrollers. The programmer shown here differs from other, similar, circuits in not requiring any extra power supply for itself, while still offering STK200 as well as STK300 programmer functionality.

In case you did not know, AVR microcontrollers can be programmed in-circuit with only five wires: Clock, MOSI, MISO, Reset and Ground. To these should be added the +5 V supply voltage taken from the microcontroller on the target board.

The programmer schematic contains



nothing more than one buffer IC type 74HC244, one 25-pin male sub-D connector for hooking up to the parallel printer port ('Centronics') on the PC, a 100 kΩ pull-up resistor on the MISO line and a .1 μF decoupling capacitor on the +5 V supply rail.

With some tinkering, the complete circuit can be fitted in the sub-D connector housing. A short length of flat-cable and a 6-way IDC socket at the target board side complete the programmer. After programming, you simply disconnect the programmer cable from the target board.

The STK200 or STK300 programmer hardware is available in lots of microcontroller programmer software, for example, BASCOM and CodeVision.

(060374-1)

# Satnav for Robots

## GPS guidance for autonomous vehicles

Ulli Sommer

**A (frequently unfulfilled) ambition of every robot builder is to make their machine capable of autonomous navigation. This is an ideal application for a GPS receiver module: these have recently become very cheap to buy. Our GPS-based navigation system is built around an ATmega32, programmed using BASCOM BASIC. It communicates with the outside world using an I<sup>2</sup>C bus.**



Any robotics hobbyist would dream of being able to build a robot which, like the famous Mars Sojourner Rover, can autonomously negotiate unknown terrain. Ideally one would just program in the coordinates of the desired destination and the little chap would make his own way there automatically. Although fully autonomous robots must remain a pipe dream for now, a solution is available to the navigation problem, as we demonstrate here with a circuit board designed to be added to a domestic surveillance robot (see large photograph).

Rather than develop a navigation system from scratch ourselves we make use of

low-cost receiver modules that receive and process signals from the GPS satellite positioning system. To this we add a moderately powerful microcontroller that can be programmed using free software.

### ATmega at the helm

Our GPS-based navigation system is built around an Atmel ATmega32, which appears at the heart both of the circuit diagram (**Figure 1**) and of the prototype printed circuit board (**Figure 2**). The microcontroller is programmed in a dialect of BASIC using the BASCOM development system, which is widely used and availa-

ble for free (for the demonstration version at least) download from the manufacturer [1]. Also, the source and object files for the navigation program are available for free download from the *Elektor Electronics* website. The file reference is **070350-11.zip**. An ISP cable is also required, obtainable for example from [3].

Any commercial GPS 'mouse' receiver can be used as long as it has an RS-232 interface. If the interface uses TTL signal levels it can be connected directly to our printed circuit board; if, on the other hand, it uses standard RS-232 levels (up to  $\pm 15$  V) a MAX232 level shifter must be connected

## How to program track points

Before attempting any autonomous journeys we must program a series of set coordinates ('track points') into the navigation system. The first step is to replace the GPS mouse with a (null modem) data cable, connected to a PC. On the PC, start up a terminal emulator program such as Hyperterminal. A terminal emulator is also included in the BASCOM package.

The interface parameters must be set as for the mouse (i.e., to 4800 baud). When connection has been set up, press the reset and programming buttons simultaneously. Then first release the reset button and then the programming button around a second later.

You should see a welcome message (which indicates how to get help) and a prompt on the terminal.

The following commands are also available: 'Data' lists the stored GPS data; 'Input' allows the GPS data to be edited; and 'Reset' restarts the navigation system.

The coordinates of the track points can be determined using a separate GPS system; alternatively, the robot can be moved manually to each track point in turn and the coordinates read off the LCD.

With the track point data programmed in, the robot can be left to its own devices!

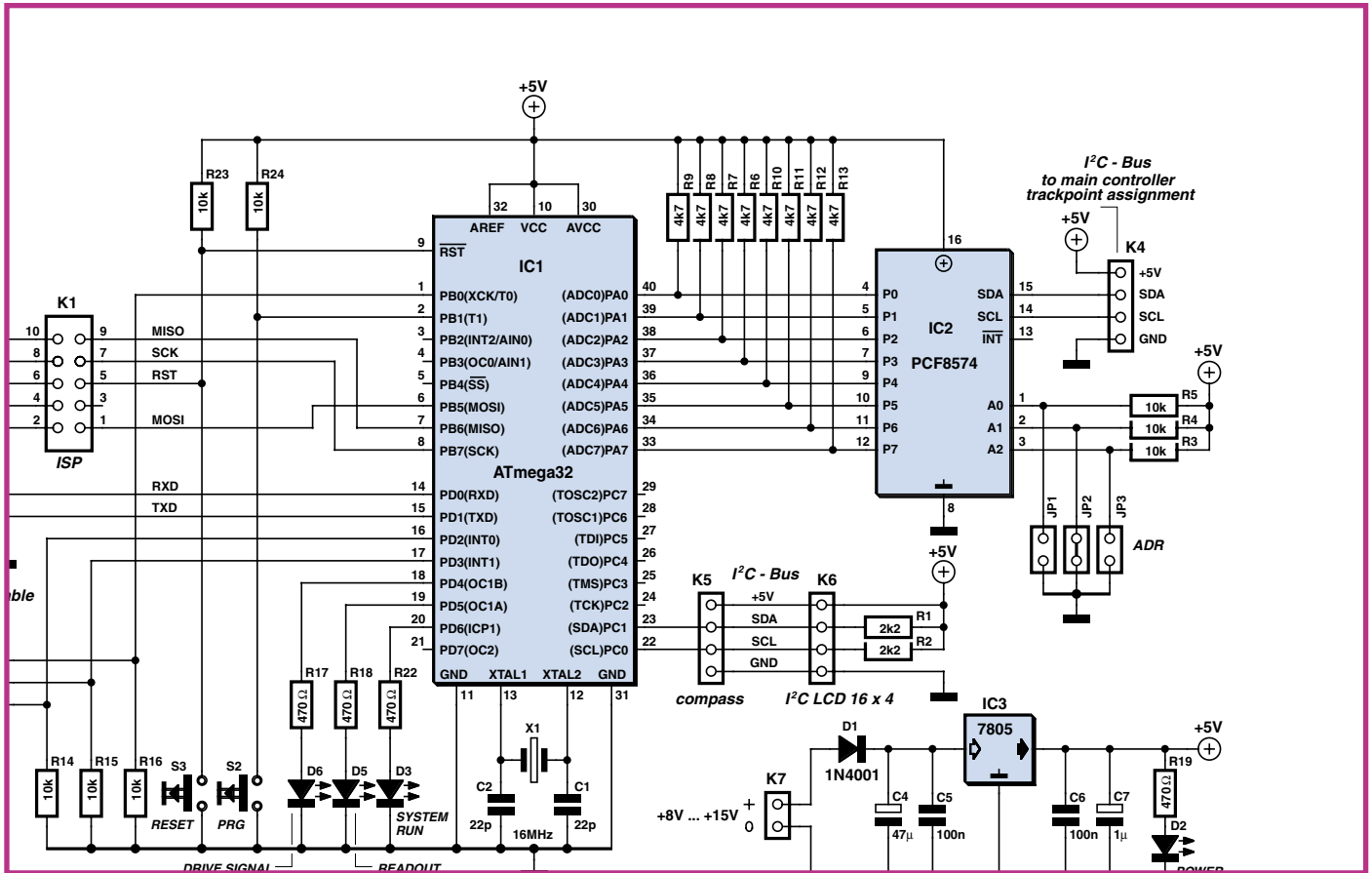


Figure 1. The ATmega32 communicates with the compass and the LCD over an I<sup>2</sup>C bus; the GPS mouse is connected via RS-232. The PCF8574 I<sup>2</sup>C interface chip takes on the task of accepting a control byte and making it available on pins 33 to 40 of the ATmega.

in between. Often a robot's main circuit board will already have a suitable level shifter IC on it.

The GPS mouse gives the exact geographic coordinates (latitude and longitude) in a defined format: see text box. It can also determine the compass direction of movement if its speed is more than about 3 km/h to 5 km/h. Since we wish to determine orientation even when stationary we require an additional 'electronic compass' (see block diagram in Figure 3). A suitable compass

module is, for example, the Devantech CMPS03, available from [4]. This compass is connected to the I<sup>2</sup>C port on our printed circuit board.

If we want to display the position and ori-

entation we will also need an LCD module. The microcontroller is rather lacking in I/O port pins, and so the most practical solution is to drive the display also via the I<sup>2</sup>C bus. There do exist LCD modules with built-in

I<sup>2</sup>C interfaces, but an alternative is to use an I<sup>2</sup>C interface chip such as the PCF8574 [5] (see Figure 4). We will see another use of this device in our circuit later on.

### Motor control

Of course we do not just want our robot to display where it is; we want it to make its

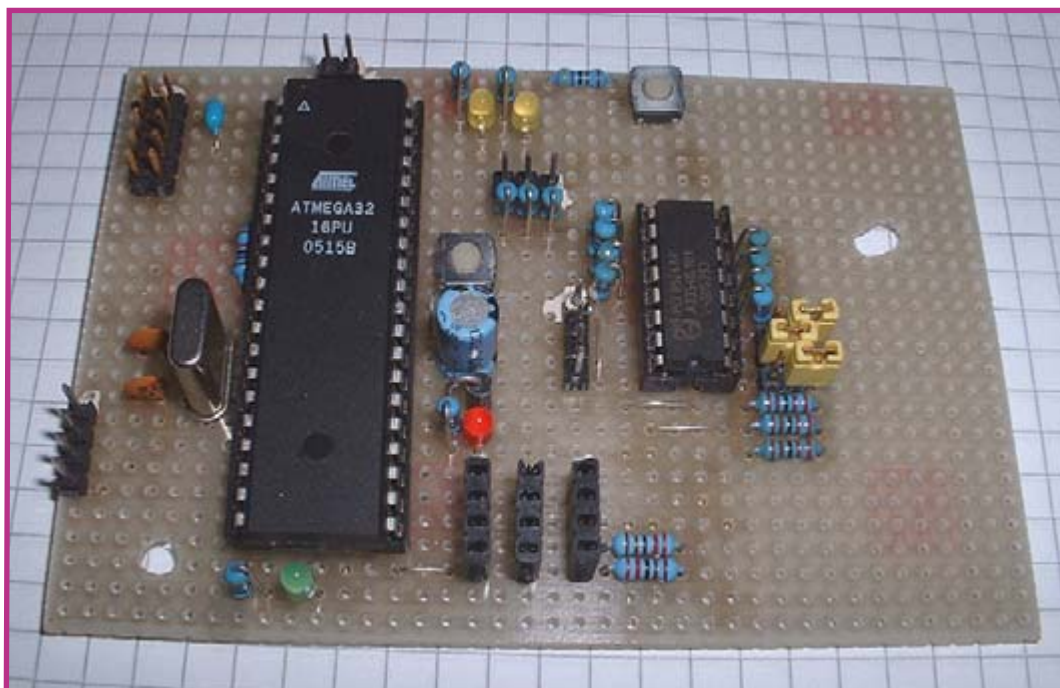


Figure 2. Prototype navigation system printed circuit board for autonomous robots.



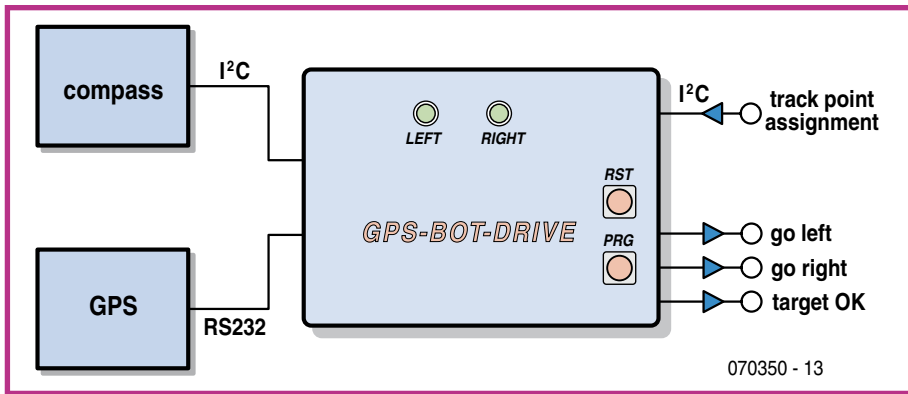


Figure 3. Simplified block diagram of the navigation system. The buttons are used when programming. Commands for the motors are output on two port pins.

way to our desired destination. In doing this the navigation system outputs direction control information on two pins. Output pins Motor\_l and Motor\_r combined give the desired direction of travel as follows:

Motor_l	Motor_r	Function
0	0	STOP
1	1	straight on
1	0	turn left
0	1	turn right

Now we need a way to tell the robot navigation system where we want it to go. To do this we first need to program a series of set coordinates ('track points') into the unit, which is done using the cable before attempting any autonomous journeys (see text box). In normal operation the navigation system then only needs to be told which track point it should try to visit next, which can, for example, be done by the robot's main processor.

This second communications challenge is also solved using the I<sup>2</sup>C bus, which is easy to use from within BASCOM BASIC. A PCF8574 I<sup>2</sup>C interface chip takes on the task of accepting a control byte and making

it available on pins 33 to 40 of the ATmega (see Figure 1). The software configures the device to run in input mode. For a simpler hardware design, it is of course possible to dispense with the interface chip and drive the port pins of the ATmega32 high and low directly and in parallel.

To select, for example, track point 1 using the I<sup>2</sup>C interface we must send the number '1' to the PCF8574. In BASCOM BASIC this might be done as follows.

```
I2cstart
I2cwbyte &H7A (address of
PCF8574: see data sheet for
addressing scheme)
I2cwbyte 1 (track point number)
I2cstop
```

When the track point selection byte has been sent, the navigation system determines the direction to the destination. This calculated direction is then compared to the current orientation of the robot (obtained from the compass). In this calculation we ignore the curvature of the earth, since we do not expect our robot to embark on long-distance journeys!

If the desired and actual directions are in agreement, the robot advances in a straight line. If, as it moves, the robot should deviate from the line to the destination, the

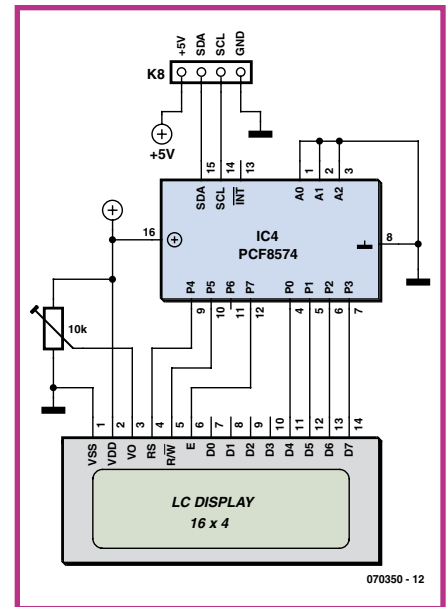


Figure 4. The LCD is driven over an I<sup>2</sup>C bus, the PCF8574 interface chip making a second appearance in the circuit (see Figure 1).

navigation system swings into action and brings it back on course. If the quality of the GPS signal is too poor or too few satellites are visible the robot will wait until an adequate signal is available to recalculate the desired course.

When the destination is reached the robot stops and takes the pin Dest\_ok high. This signal can be used by the robot's main processor, for example to load up the next track point so that the machine traces out a predetermined course.

(070350-1)

#### Web links

- (1) <http://www.mcselec.com>
- (2) <http://www.elektor-electronics.co.uk>
- (3) <http://www.kanda.com>
- (4) <http://www.robot-electronics.co.uk>
- (5) <http://www.nxp.com/cgi-bin/pldb/pip/pcf8574>

#### GPS mouse data

A typical data packet received from a GPS mouse using the simple \$GPGGA protocol might appear as follows.

```
$GPGGA,191410,5212.9324,N,00007.5930,E,1,04,4.4,11.5,M,48.0,M,,*73
```



Here '\$GPGGA' is the protocol type, '191410' the time, '5212.9324,N' the latitude, '00007.5930,E' the longitude, '1' indicates that the reading is valid and '04' is the number of satellites in view. As you can see, this covers all the information we need. All we need to do is switch the GPS mouse to the \$GPGGA protocol and set the communication speed to 4800 baud. Other protocols should be disabled, and the reporting interval set to approximately 1 s. The settings are made using the software provided with the GPS mouse.



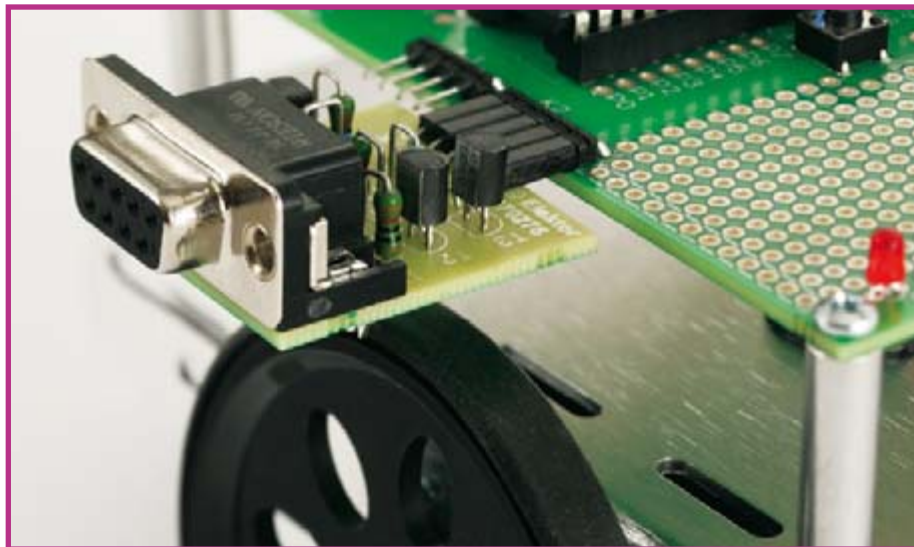
# Serial Interface for the Propeller

## Simple and inexpensive

Luc Lemmens

The Propeller prototyping board described elsewhere in this issue needs a programming interface, just like the board available from Parallax. The manufacturer offers the Propeller Plug and the Propeller Clip for this purpose. They can be used to link the board to a PC via a USB port. These mini-boards are fitted with an FTDI FT232 IC, which has appeared quite regularly in the magazine. The difference between the Plug and the Clip is in how they connect to the processor board. The Plug connects to a 4-way SIL header, while the Clip connects to four solder pads at the edge of the board. The latter type of connection was used in the first version of the Propeller demo board, and it is actually no longer relevant. The Propeller Plug is the right USB interface for the prototyping board from Parallax and our prototyping board. This little board costs around twenty euros, but if you want to save a bit of money and prefer to use the old faithful RS232 interface (and your PC has a serial port), you can build the simple serial interface described here.

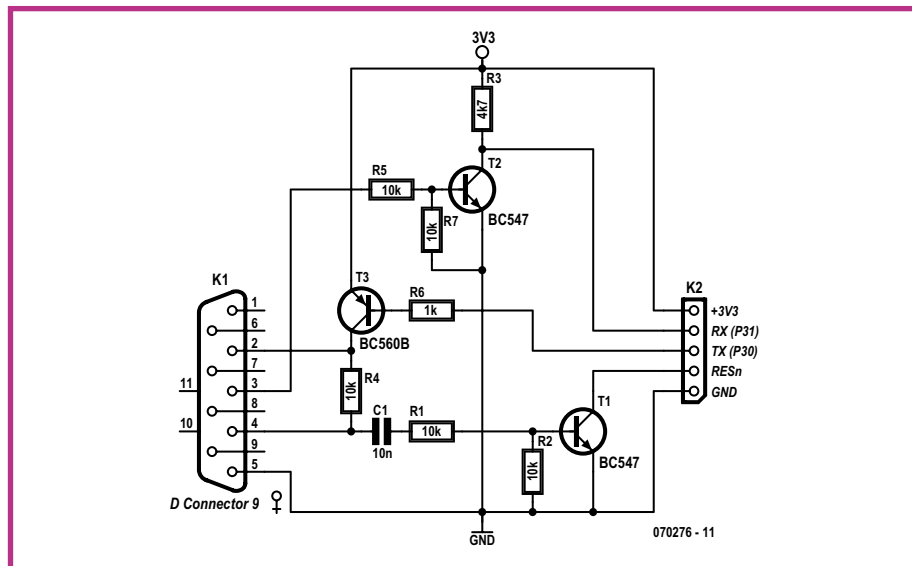
Three garden-variety transistors, a handful of resistors, and a capacitor are all it takes to let the Propeller communicate with a PC via the serial port. The interface actually consists of nothing more than three inverters and level shifters, which enable the Propeller board (which is powered from a 3.3-V supply) to talk to the COM port of a PC, which operates with  $\pm 12$ -V signals. Connector K2, which provides the link to the Propeller board, has intentionally been laid out with the signal lines in the same sequence as on the Propeller Plug, but here we need an additional line for the 3.3-V supply voltage. This makes it possible to use the interface board with the Parallax board as well. However, in that case a small modification is necessary for the supply voltage connection.



We have designed a small PCB for this circuit, but of course it's no problem to build it on piece of perforated prototyping board instead. With a bit of effort, you can probably even make it so compact that the entire circuit fits in a plug housing for a 9-way RS232 connector. Fortunately, when you use this interface

you don't have to worry about configuring all sorts of settings (baud rate, number of bits and so on) — the Propeller Tool development software does all this for you. Use a 1-to-1 cable for the serial link (not a crossover cable or null modem cable).

(070276-1)



## COMPONENTS LIST

### Resistors

- R1, R2, R4, R5, R7 = 10k $\Omega$
- R3 = 4k $\Omega$
- R6 = 1k $\Omega$

### Capacitors

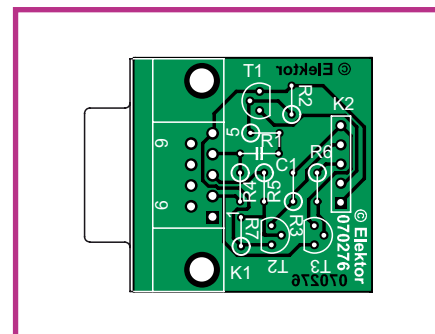
- C1 = 10nF

### Semiconductors

- T1, T2 = BC547
- T3 = BC557

### Miscellaneous

- K1 = 9-way sub-D socket (female), angled pins, PCB mount
- K2 = 5-way SIL socket
- PCB, ref. 070276-1 from [www.thepcbshop.com](http://www.thepcbshop.com)

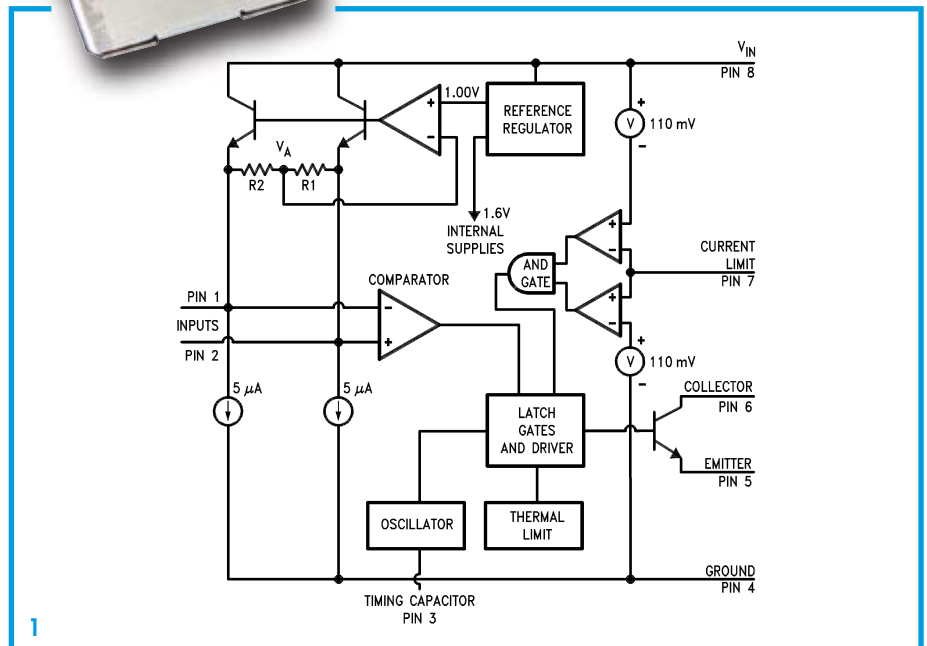
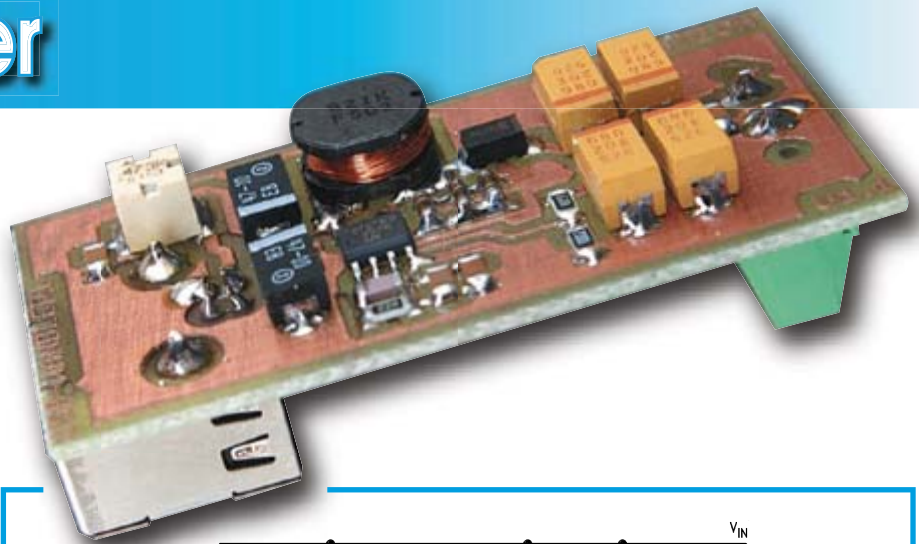


# USB Converter

Jörg Schnyder

Does this sound familiar: you buy a small piece of equipment, such as a programming & debugging interface for a microcontroller, and you have to use a clunky AC wall adapter to supply it with power? It's even worse when you're travelling and there's no mains socket anywhere in sight. Of course, you can use the USB bus directly as a power source if the supply voltage is 5 V. If you need a higher voltage, you can use the USB converter described here. This small switch-mode step-up converter can generate an output voltage of up to 15 V with a maximum output current of 150 mA.

The LM3578 is a general-purpose switch-mode voltage converter. **Figure 1** shows its internal block diagram. Here we use it as a step-up converter. **Figure 2** shows the necessary components. Voltage conversion is achieved by switching on the internal transistor until it is switched off by the comparator or the current-limiting circuit. The collector current flows through coil L1, which stores energy in the form of a magnetic field. When the internal transistor is switched off, the current continues flowing through L1 to the load via diode D1. However, the voltage across the coil reverses when this happens, so it is added to the input voltage. The resulting output voltage thus consists of the sum of the input voltage and the induced voltage across the coil. The output voltage depends on the load current and the duty cycle of the internal transistor. Voltage divider R5/R6 feeds back a portion of the output voltage to the comparator in the IC in order to regulate the output voltage. C5 determines the clock frequency, which is approximately 55 kHz. Network R4, C2 and C3 provides loop compensation. The current-sense resistor for the current-limiting circuit is formed by three 1-Ω resistors in parallel (R1, R2 and R3), since SMD resistors with values less than 1 Ω are hard to find. The output voltage ripple is determined by the values and internal resistances of capacitors C11, C8, C7 and C6. The total effective resistance is reduced by using several capacitors, and this also keeps the construction height of the board low. L2, C1, C9 and C10 act as an input filter. Ensure that the DC resistance of coil L2 is no more than 0.5 Ω. Use a Type B PCB-mount USB connector for connection to the USB bus. A terminal strip with a pitch of 5.08 mm can be used for the output voltage connector. Of course, you can also solder a cable directly to the board. Two additional holes are provided

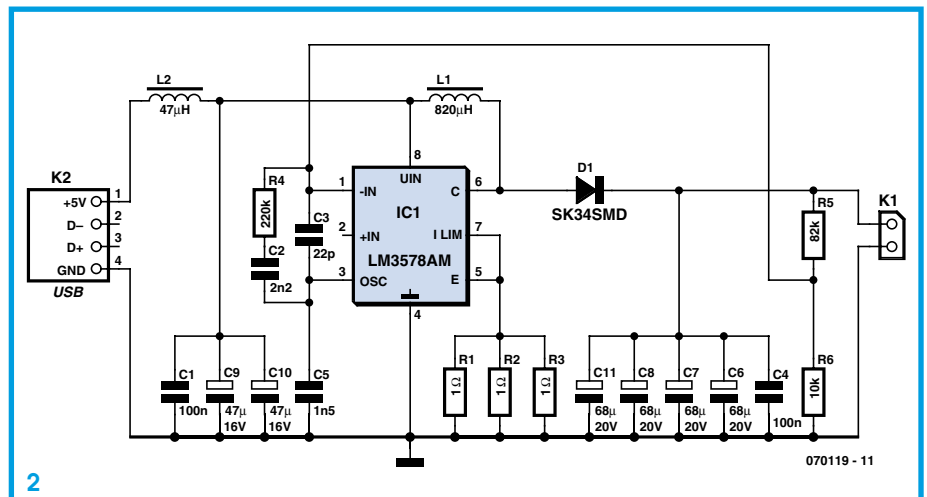


in the circuit board for this purpose.

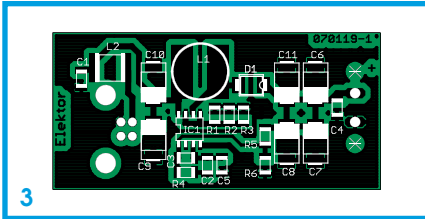
As we haven't been able to invent a device that produces more energy than it consumes, you should bear in mind that the input current of the circuit is higher than the output current. As a general rule, you can assume that the input current is equal to the product of the output current and the output voltage divided by the input

## R5 and R6 for other output voltages:

6V:	R5 = 47k, R6 = 9,1k
12V:	R5 = 110k, R6 = 10k
15V:	R5 = 130k, R6 = 9,1k







voltage and divided again by 0.8. Specifically, with an output current of 100 mA at 9 V, the input current on the USB bus is approximately 225 mA.

Finally, **Figure 3** shows a small PCB layout for the circuit. All of the components except the connector and the terminal strip are SMDs.

(070119-1)

**Web link**

Author's homepage: [www.systech-gmbh.de](http://www.systech-gmbh.de)

**components list**

(for  $U_O = 9\text{ V}$ )

**Resistors**

- R1, R2, R3 = 1Ω
- R4 = 220kΩ
- R5 = 82kΩ
- R6 = 10kΩ

**Capacitors**

- (SMD 1206)
- C1 = 100nF
- C2 = 2nF
- C3 = 22pF
- C4 = 100nF
- C5 = 1nF
- (tantalum SMD 7343)
- C6 = 68μF 20V
- C7 = 68μF 20V
- C8 = 68μF 20V
- C9 = 47μF 16V

C10 = 47μF 16V

C11 = 68μF 20V

**Inductors**

L1 = 820μH (SMD CD105)

L2 = 47μH (SMD 2220)

**Semiconductors**

D1 = SK34SMD (Schottky)

IC1 = LM3578AM (SMD SO8)

**Miscellaneous**

K1 = 2-way PCB terminal block, lead pitch 5mm (optional)

K2 = USB-B connector

PCB layout, free download from Elektor website, **070119-1.pdf**

# zBot: Solar/Battery Power Supply

Jens Altenburg

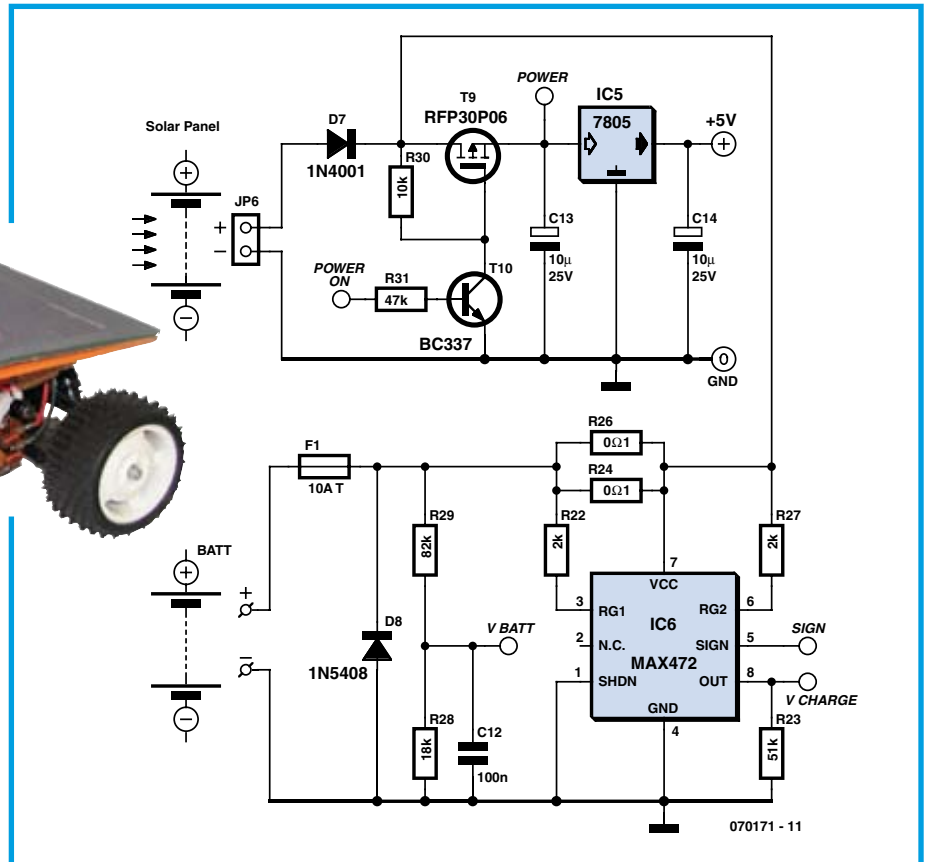
One of the most important problems of mobile robot platforms is associated with the power supply. With exception of some special systems, batteries, dry or recharge-



able, are the most commonly seen sources of electrical power in robots.

The power supply system of zBot<sup>1</sup> consists of two parts, the main accumulator and the auxiliary battery. The main power source was realised with a NiCd or NiMH battery pack. Its size was adapted to fit the battery holder of the Tomya chassis (six 1.2V/1400 mAh C cells). The main power is activated for DC motor driving and for the servos.

The auxiliary system, two Alkaline AAA



batteries, is for the power supply of the microcontroller only. A third (optional) power source is the solar panel. It is not really necessary for initial experiments but it helps to keep the robot

autonomous longer. The circuit of zBot's main power supply system includes a special feature: the charge control circuit based around a MAX472. For effective operation, we have to know the

exact capacity of the battery. Imagining the discharge voltage diagram, we know that the voltage is virtually stable for most of the discharge time and suddenly breaks down when the battery runs out of capacity. This time is very short, so it could happen that the robot could be lost.

A simple voltage control doesn't give us the information we need. The only way to obtain exact values is monitoring the discharging. The MAX472 gives two values, the current

through R24/R26 as a proportional voltage at pin 8, and the current direction through the resistors (SIGN). Both values allow calculation of the charging (solar panel) or discharging of the battery pack.

The auxiliary power is shown in the CPU unit. The two alkaline batteries support the CPU, the wireless radio modem and the navigation system (compass) only. The reason for the division of the power supply is simple. With the help of the aux-

iliary system, zBot communicates with the operator wirelessly. In this way, the independent power source increases the security of the system.

(070171-1)

(1) The complete document called Zbot — the Robot Experimental Platform is available for free downloading from the Elektor Electronics website. The file number is 070172-11.zip (July/August 2007).

# 3-A Wide-input Adjustable Switching Regulator

Luc Lemmens

The PTN78060 is a series of high-efficiency, buck-boost, integrated switching regulators (ISR) from good old Texas Instruments (TI).

The caseless, double-sided package has excellent thermal characteristics, and is RoHs compliant.

The PTN78060 devices operate from a remarkably wide input voltage range:

Device	V <sub>in</sub>	V <sub>out</sub>
PTN78060Wa(x)	7 to 36 V	2.5 to 12.6 V
PTN78060HA(x)	15 to 36 V	11.85 to 22.0 V
PTN78060AA(x)	9 to 29 V	-15 to -3V

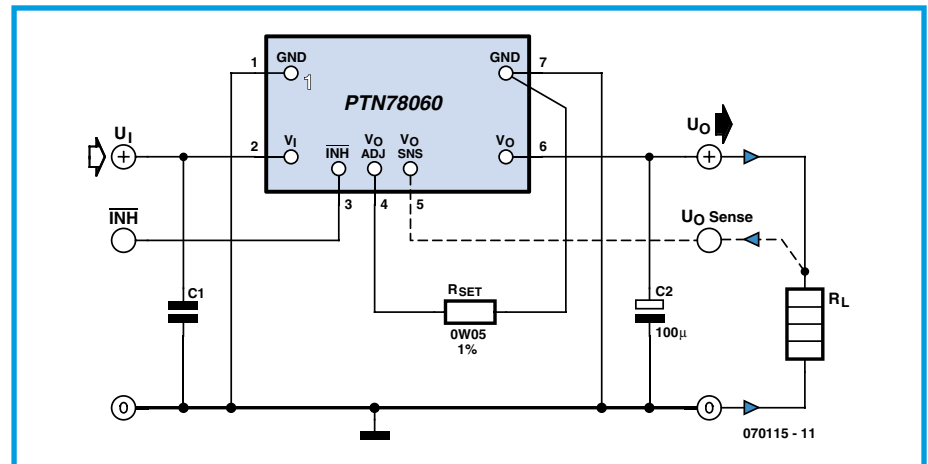
Note that the -A version supplies a negative output voltage.

The devices provide high-efficiency step-down voltage conversion for loads of up to 3 A.

The PTN78060 devices are suited to a wide variety of general-purpose applications that operate off 12-V, 24-V, or tightly regulated 28-V dc power, hence are ideal for running low-voltage electronics from a very high power 24-V battery unit salvaged from an electric wheel chair and migrated into a robot.

The output voltage V<sub>O</sub> can be set to any value over a wide adjustment range using a single external resistor R<sub>SET</sub>, using the equation

$$R_{SET} = 54.9k\Omega \times (1.25V/V_O - V_{MIN}) - R_p$$



Device	V <sub>O</sub> (desired) (V)	R <sub>SET</sub> (standard value) (kΩ)	V <sub>O</sub> (actual) (V)	V <sub>i</sub> range (V)
PTN780x0W	2.5	Open	2.5	7 to 25
	3.3	78.7	3.306	7 to 33
	5.0	21.0	4.996	7 to 36
	12.0	0.732	12.002	14.5 to 36
PTN780x0H	12.0	383	12.000	15 to 36
	15.0	15.0	14.994	18 to 36
	18.0	4.42	18.023	21 to 36
	22.0	95.3	21.998	26 to 36

If pin 4 is left open, the output voltage defaults to the lowest value. Limiting ourselves to the two positive-output regulators, for the -W version, V<sub>MIN</sub> and R<sub>p</sub> are 2.5 V and 6.49 kΩ respectively; for the -H device, the values 11.824 V and 6.65 kΩ should be used.

For the output to remain in regulation, the input voltage must exceed the output by a minimum differential voltage. Another consideration is the pulsewidth modulation (PWM) range of the regulator's internal control circuit. For stable operation, its

operating duty cycle should not be lower than a certain minimum percentage. This defines the maximum advisable ratio between the regulator input and output voltage magnitudes.

For satisfactory performance, the operating input voltage range of the PTN78060x must satisfy the following requirements.

1. For PTN78060W devices supplying output voltages lower than 10 V, the minimum input voltage is (V<sub>O</sub>+2 V) or 7 V, whichever is higher.

2. For PTN78060Ws supplying output voltages of 10 V and higher, the minimum input voltage is ( $V_O+2.5$  V).

3. The maximum input voltage for PTN78060W is  $10V_O$  or 36 V, whichever is less.

4. For PTN78060H output voltages lower than 19 V, the minimum input voltage is ( $V_O+3$  V) or 15 V, whichever is higher.

5. For PTN78060H output voltages equal to 19 V and higher, the minimum input voltage is ( $V_O+4$  V).

As an example, the **Table** gives the operating input voltage range for some commonly used output bus voltages.

The modules are protected against load faults with a continuous current limit characteristic. Under a load-fault condition, the output current increases to the current limit

threshold. Attempting to draw current that exceeds the current limit threshold causes the module to progressively reduce its output voltage. Current is continuously supplied to the load until the fault is removed. Once it is removed, the output voltage promptly recovers. When limiting output current, the regulator experiences higher power dissipation, which increases its temperature. If the temperature increase is excessive, the module overtemperature protection begins to periodically turn the output voltage off.

The inhibit feature can be used wherever there is a requirement for the output voltage to be turned off. The power module switches off the output voltage when the Inhibit control (pin 3) is pulled to ground, for example, by a switching FET.

Finally, good attention should be paid to the quality of the capacitors on  $V_I$  and  $V_O$

as they determine the regulator stability and overall performance to a substantial degree. Summarizing the extensive information on capacitor selection found in the datasheets, the minimum requirement for C1 is 2.2  $\mu$ F (!) worth of ceramic capacitors for the -W device and 14.1  $\mu$ F (!!)

for the-H device. Tantalum caps are not recommended. Similarly, at the regulator output, C2 should be at least 100  $\mu$ F worth of low-ESR electrolytics.

(070115-1)

#### Datasheets

<http://focus.ti.com/docs/prod/folders/print/ptn78060h.html>

<http://focus.ti.com/docs/prod/folders/print/ptn78060w.html>

<http://focus.ti.com/docs/prod/folders/print/ptn78060a.html>

## Paralleling LiPo Batteries



Paul  
Goossens

LiPo (Lithium Polymer) batteries have a number of advantages compared to NiCd and NiMH batteries. In addition to having a lower weight for the same capacity, LiPo batteries can also be made in various shapes. The first property is eagerly exploited by manufacturers of mobile phones, MP3 players and the like.

Beside these advantages, LiPo batteries also have a few disadvantages. One of these disadvantages is that they are not able to supply the same amount of current as their NiCd and NiMH brethren. The maximum current is typically 10 C, where C is the nominal capacity. Newer versions are able to supply 15 C to 30 C continuously, but you will be paying a much higher price for those!

Using a battery rated at 1000 mAh this means that a normal LiPo cell may be

loaded at up to 10,000 mA or 10 A. The current is often allowed to be double that for short periods of time, but that is not so beneficial to the life expectancy of the

LiPo cells!

There are many cases where we would like to draw more current from the battery. This can be done by connecting multiple cells in parallel.

#### Current limiting

Connecting multiple cells in parallel is, in principle, a simple soldering job. We don't have to waste any words on that! However, we do have to make sure — before the cells are connected in parallel — that they all have exactly the same voltage

across their terminals. If there is even only a small difference between the source voltages then during and after connecting the batteries in parallel a large equalising current can flow. This current will discharge the battery with the higher output voltage and charge the battery with the lower voltage, until both voltages are the same. This equalising current obviously has to be smaller than the maximum charging current (typically 1 C).

Before we can connect the cells in parallel we have to take measures to limit any equalizing current. The difference in voltage is often so small that a simple current regulator does not work properly. However, using a resistor we can limit this current quite easily.

#### Manual control

To do this correctly we need to know the maximum charging current for both batteries. We then measure the voltage across both batteries. The difference between these two voltage we call the difference voltage. The negative terminals can now be soldered together. Now we temporarily solder a resistor between the two positive terminals. The value of this resistor has to be at least the difference voltage divided by the maximum charging current.

The battery with the greatest amount of charge will now charge the other battery



at a limited rate. The latter will therefore charge slowly. After a while the difference voltage will reduce and therefore the charging current as well. If this process doesn't go quickly enough

for your liking then you can adjust the resistor value from time to time to increase the charging current again. Both positive terminals may be directly connected together once the difference voltage has dropped

so low that a resistor of 10 mΩ would have been enough. A new and more powerful LiPo battery is now a fact.

(070274-1)

# Switch-Mode 555 Supply

Martijn Geel

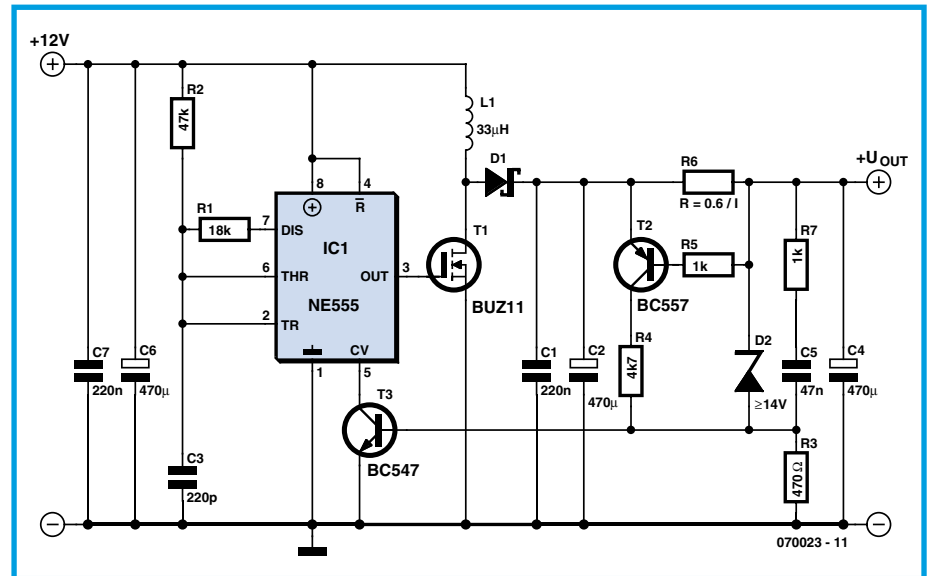
This switch-mode power supply is built around a 555 timer IC. It provides a maximum output voltage of 40 V with a 12-V input voltage. The voltage can easily be set using a Zener diode, and it must be higher than the input voltage (the minimum output voltage is always 12 V).

The NE555 is used in an unconventional way here. In the normal configuration, the output of the oscillator IC is low longer than it is high. With the configuration used here, the output can be high for a shorter time than it is low.

The NE555 switches FET T1 on and off. When T1 is conducting, energy is stored in L1. When T1 stops conducting, this energy is transferred to C1 and C2 via Schottky diode D1, so the voltage on these capacitors rises.

The voltage is limited by Zener diode D2. If the voltage rises above the Zener voltage, the current through the Zener diode causes T3 to conduct. This reduces the voltage on pin 5 of the NE555, which in turn decreases the relative duration of the high level on pin 3. T1 thus conducts for a shorter interval, so less energy is stored in L1 and the output voltage is stabilised.

Current limiting is provided by R6, R5 and T2. If the voltage across R6 is more than 0.6 V, T2 starts to conduct. This drives T3 into conduction, causing the voltage to decrease in order to limit the current.



C5 and R7 provide a soft-start effect. The value of R1 can range from 22 kΩ for an output voltage of 15 V to 10 kΩ for an output voltage of 40 V.

For the sake of safety, limit the Zener voltage to a maximum of 40 V. T1 and T2 can be rated for a maximum of 50 V. The FET is not critical; you may already have one in your spare parts bin that can switch enough current. If the coil becomes warm, the core is too small or the wire is too thin. The Schottky diode is the only component that is actually critical. Do not use an ordinary diode, since it will become much to

hot. You're bound to find a Schottky diode in an old computer power supply (just check for a forward voltage of 0.2 V on the diode range of your multimeter).

The supply shown here can deliver approximately 200 W. The input supply voltage can range from 7 V to 15 V. Don't forget that the maximum voltage the NE555 can handle is 15 V.

Finally, this power supply is not short-circuit proof. A slow-blow fuse on the 12-V side is recommended.

(070023-1)

# Voltage Stabiliser

Alexander Wiedekind-Klein

Electric motors used in robot applications often make sudden and heavy demands on their power supply. Although the batteries normally used have a low internal

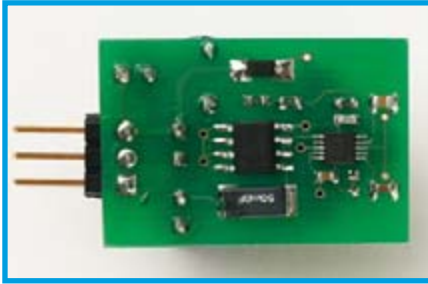
resistance, they nevertheless sometimes have difficulty maintaining their output under load and can be damaged by current spikes. Not all the electronics in the robot can cope with these effects, the voltage regulation provided by ordinary three-

terminal devices not always being up to the job. This electronic voltage stabiliser is a solution to that problem.

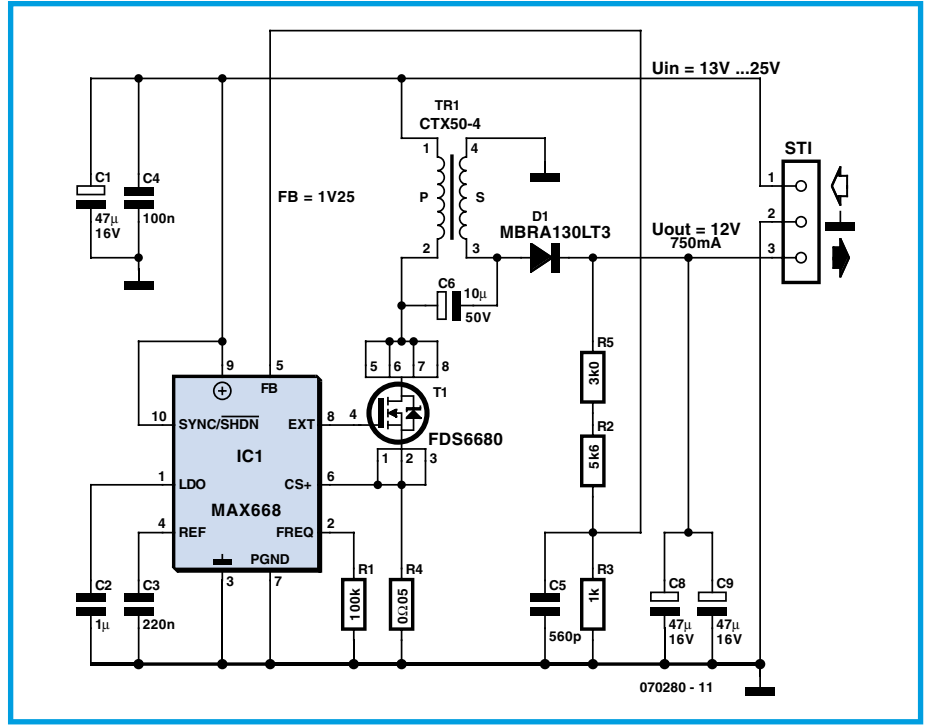
The circuit is based around a compact switching regulator which is capable of



producing a steady DC voltage of 12 V at its output over input variations from 13 V to 25 V, at loads of up to 750 mA. Its three-



pin form factor makes it a simple replacement for conventional three-terminal voltage regulator ICs. The buck-boost switching circuit uses an SMD power FET for T2, and to achieve high efficiency (approximately 90 %) a Schottky switching diode for D1. The most specialised component is the miniature transformer designed for use in this type of supply. For the prototype we used a Coiltronics CTX50-4. The current limit is set by R4. The output



voltage is scaled by the voltage divider formed by R3 and the series combination of R5 and R2. The output voltage is controlled so that a voltage of 1.25 V appears across R3 and hence on the feedback input (pin 5) of IC1. The circuit can be modified for different output voltages by changing the component values in the voltage divider. For lower output voltages the input voltage can also be reduced correspondingly.

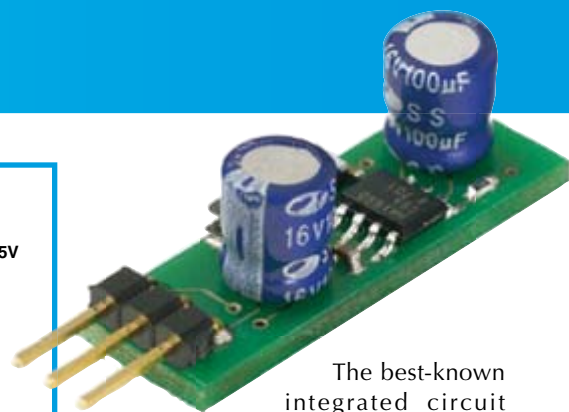
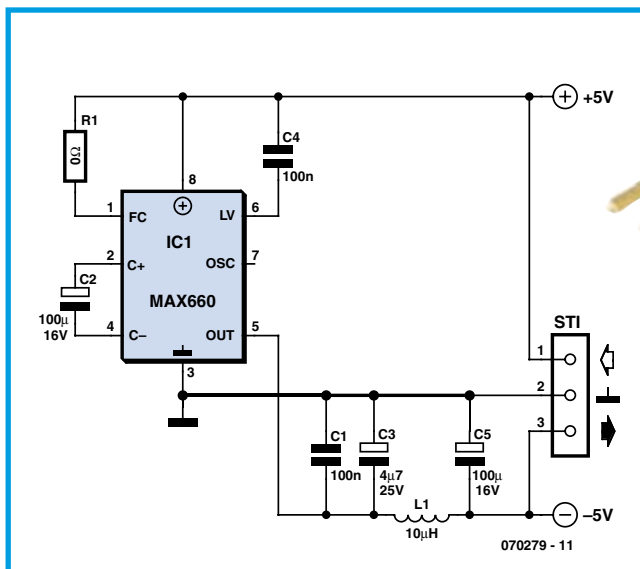
Gerber files for the prototype printed circuit board are available for free download from the Elektor website, ref. **070280-1.zip**. The SMD components used have the following outlines: R1, R2, R3, R5, C3, C4 and C5: 0603; C2: 0805; IC1: SSOP-12; T1: SO-8. All the SMD capacitors are ceramic, and electrolytics C1 and C8 must have a low ESR. R4 is a 50 mΩ SMD resistor rated at 1 W.

(070280-1)

# Mini Power Inverter

Alexander Wiedekind-Klein

Even robot systems occasionally need a negative supply voltage for some purpose or other, and in this kind of application in particular there is a need for an effective circuit that does not make greater demands than necessary in terms of current or space. If a low-current -5 V supply is needed and only +5 V is available, a natural manufacturer to turn to is Maxim, and indeed in this case they do not let us down.



The best-known integrated circuit made by this company is the MAX232, a level shifter for serial ports with an integrated charge pump that does not need an external inductor. Along the same lines, although with a more stable output voltage and higher efficiency, is the MAX660. The device can 'mirror' any input voltage between 1.5 V and 5.5 V. With a 5 V input the output is typically -4.7 V with a load of 100 mA. Efficiency at 10 mA is around

96 % and at 100 mA is around 88 %. With an open-circuit output the IC draws a quiescent current of just 120  $\mu$ A.

There is little to say about the circuit itself. The 0  $\Omega$  resistor on pin 1 selects the operating frequency. With R1 fitted, the circuit operates at 80 kHz; without it, at 10 kHz.

The combination of L1 and C5 slightly reduces ripple on the output voltage; the choice of inductor is not as critical as it would be if it formed part of the switching circuit.

Gerber files for the printed circuit board (which uses some SMD components) are

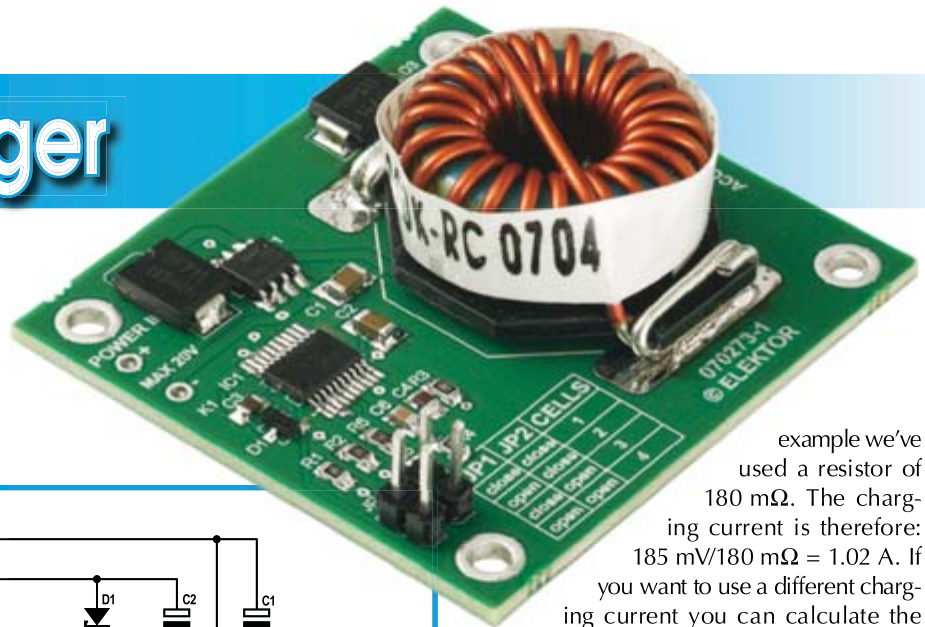
available for download from the Elektor website, ref. **070279-11.zip**. R1, C1 and C4 are 0603 SMDs and C3 is an SMD tantalum electrolytic capacitor. Either the MAX660CSA or the MAX660M can be used; both come in SO8 packages. L1 is a 10  $\mu$ H SMD inductor rated at 300 mA.

(070279-1)

# Lithium Charger

Paul Goossens

Batteries based on Lithium, such as LiPo (Lithium-polymer) and Lithium-Ion ones are ideal candidates to supply a robot with power. Compared to other types of battery they are lighter, which results in a lower mechanical strain on the chassis. The avail-

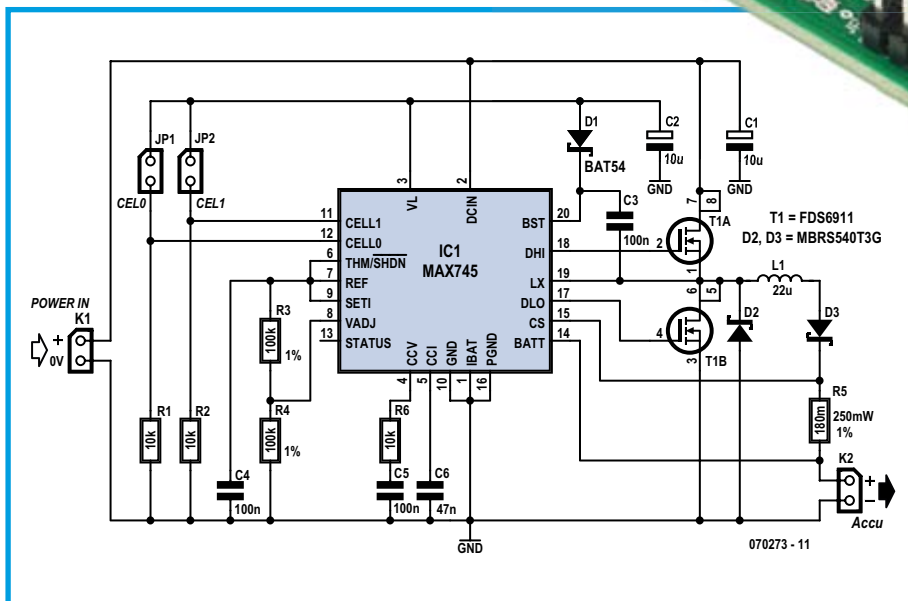


example we've used a resistor of 180  $\Omega$ . The charging current is therefore:  $185 \text{ mV} / 180 \text{ m}\Omega = 1.02 \text{ A}$ . If you want to use a different charging current you can calculate the value for R5 using the formula:

$$R5 = 185 \text{ mV} / I_{\text{CHARGE}}$$

A pair of jumpers is used to select the number of cells in the battery. The four possible settings are shown in **Table 1**.

JP1	JP2	Number of cells
Closed	Closed	1
Open	Closed	2
Closed	Open	3
Open	Open	4



ability is good as well, and they are manufactured in many shapes and sizes.

## Charging

The charging of Lithium batteries is a very exact science. If the wrong method is used there is a real chance that they'll burst into flames. For this reason it is only sensible that you always use a proper charger. With the use of a MAX745 such a charger can easily be constructed at home.

During the charging process the charge current should not rise above 1C. This means that for a 1200 mAh battery it may not be charged with a current of more than 1.2 A! Furthermore, the terminal voltage for this type of battery may never rise above 4.25 V per cell. In principle a

lithium charger is nothing more than a current source with a (precision!) maximum output voltage.

## Charge controller

In our case the charger uses a step-down converter. In this way very little power is wasted in the charger and it can operate without the use of a heatsink. The complete charge controller is inside IC1. A few external components are required for the step-down converter. These are FET T1 and its surrounding components.

The battery voltage is measured via pin 14 (BATT). The voltage difference between BATT and CS is measured by the IC to control the charging current. The IC tries to keep this potential difference to 185 mV. In our

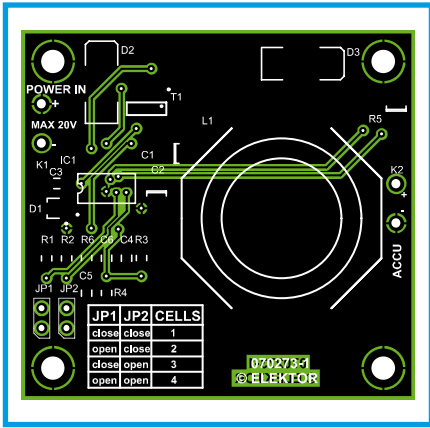
These jumpers tell the circuit how many cells are connected in series inside the battery. This is very important, since it determines what the maximum voltage may be across the battery.

The maximum voltage per cell is adjusted via  $V_{\text{adj}}$  and can be set between 3.95 V and 4.45 V. Resistors R3 and R4 set the terminal voltage to 4.25 V in this case. Because  $V_{\text{adj}}$  has a narrow operating range we have to use 1% (close tolerance) resistors to set the voltage very accurately!

## Construction

Thanks to the use of a double-sided PCB the construction of this circuit is very simple. All components are mounted on the top side of the board. When soldering the





## COMPONENTS LIST

### Resistors

- R1, R2, R6 = 10kΩ (SMD 0805)
- R3, R4 = 100kΩ 1% (SMD 0805)
- R5 = 0.18Ω 0.25W (SMD 1210), e.g. Digikey P.18SCT-ND

### Capacitors

- C1, C2 = 10μF 25V (SMD 12010)
- C3, C4, C5 = 100nF (SMD 0805)
- C6 = 47nF (SMD 0805)

### Semiconductors

- D1 = BAT54 (SOT-23)
- D2, D3 = MBR540T3G (SMC), e.g. Digikey MBR540T3GOSCT-ND
- IC1 = MAX745
- T1 = FDS6911 (SOIC12), e.g. Digikey FDS6911CT-ND

### Miscellaneous

- L1 = 22μH (JW-MILLER PM2110-220K-RC), e.g. Digikey M8760-ND
- JP1, JP2 = jumper with 2-way SIL pinheader PCB, order code **070273-1** from Elektor SHOP)

coil it may be necessary to let the soldering iron heat up a bit more first. The connecting leads are quite chunky and they require a fair amount of heat to raise them to the correct temperature.

When all parts have been soldered and the circuit has been checked you can power the circuit via K1 with a DC voltage of no more than 24 V. You should always double-check that you have set the jumpers for the

correct number of cells before connecting the LiPo or Li-Ion battery! With a charging current of 1 C an empty cell should be fully charged in about an hour and a quarter.

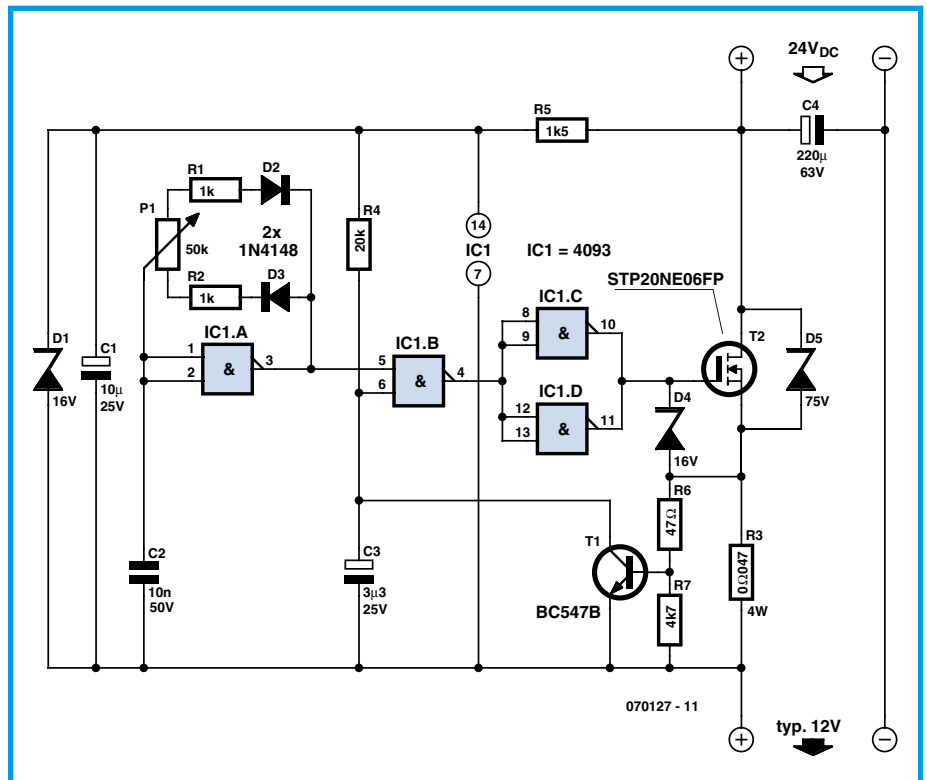
(070273-1)

# PWM Voltage Dropper

Von Stefan Brandstetter

This circuit was developed to allow a car trailer, designed for 12 V operation, to be used as a trailer for a van with a 24 V supply. A number of copies of the circuit we made, for the left and right indicators, brake lights, number plate light and reversing lights, and these have been in trouble-free operation for several years. The advantage of this compact circuit is that it dissipates very little power because it uses pulse width modulation. In addition, its quiescent current consumption is practically zero.

A simple pulse generator is constructed using IC1.A, C2, R1 and R2. Normally (when T1 is not conducting) RC combination R4/C3 ensures that IC1.B passes the square wave signal to FET switch T2. Shunt resistor R3 measures the output current. If the maximum safe output current is exceeded, T1 turns on and short-circuits C3; IC1.B no longer passes the square wave signal to the switching transistor. The output current falls to zero, T1 turns off and C3 is recharged via R4. As soon as the input threshold of IC1.B (half the supply voltage) is exceeded, the PWM signal once again starts to drive T2. Thus even if there is a continuous short circuit on the output there will be occasional pulses of output current. R5, D1 and C1 reduce the input voltage of 24 V to a value of 16 V more suitable for powering the CMOS 4093 Schmitt trigger IC. D4 and D5 protect T2 from voltage spikes, which are practically



unavoidable in this circuit because of the inductance of the wiring. Any standard N-channel FET able to withstand 100 V can be used for T2.

With the component values shown the circuit is suitable for use with 12 V lamps at up to 60 W. The current limit, set by R3 (47 mΩ) is around 12 A. The current limit is essential because cold lamps present a

very low resistance when voltage is first applied. The mark-space ratio is set to approximately 1:3 (25 % on-time) using P1. The circuit can be modified for use at higher currents, and it can also be used as the basis of a simple and efficient speed controller or light dimmer.

(070127-1)

# Dual Battery

## For enhanced operational reliability

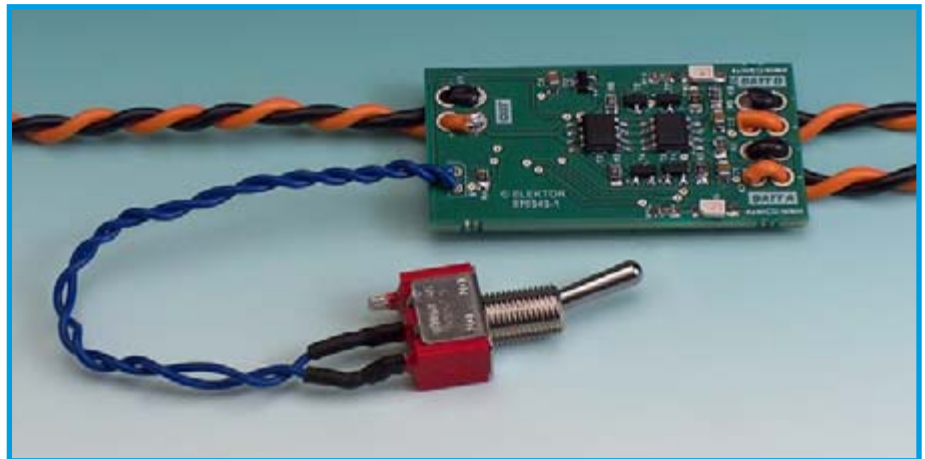
Paul Goossens

Using rechargeable batteries to power circuits is a proven method for providing energy to mains-independent equipment. A major disadvantage of this is that the battery usually turns out to be empty at the most inopportune moment. As a user, you are unexpectedly confronted with the fact that the circuit suddenly doesn't work any more. Sometimes this is only a minor inconvenience, but at other times it can be a catastrophe. For instance, just imagine what happens to a model airplane if the radio receiver stops working in flight due to an empty battery. We can assure you that the consequences are anything but pleasant.

## Solution

The solution to this problem is actually quite simple: use two batteries! When one of the batteries becomes discharged, the second one can take over and continue supplying power.

Of course, all this must happen automatically, so we need a handy circuit that takes



care of everything for us. The design presented here is intended to be used with circuits (such as receivers used in models) that use NiCd batteries composed of four cells. The circuit is quite compact, and thanks to the accompanying PCB populated with SMDs, it is easy to fit into existing equipment.

## Simple

The operating principle is simple: IC2 measures the terminal voltage of battery A. If it drops below 4.38 V, the RESET output goes low, and otherwise it remains high.

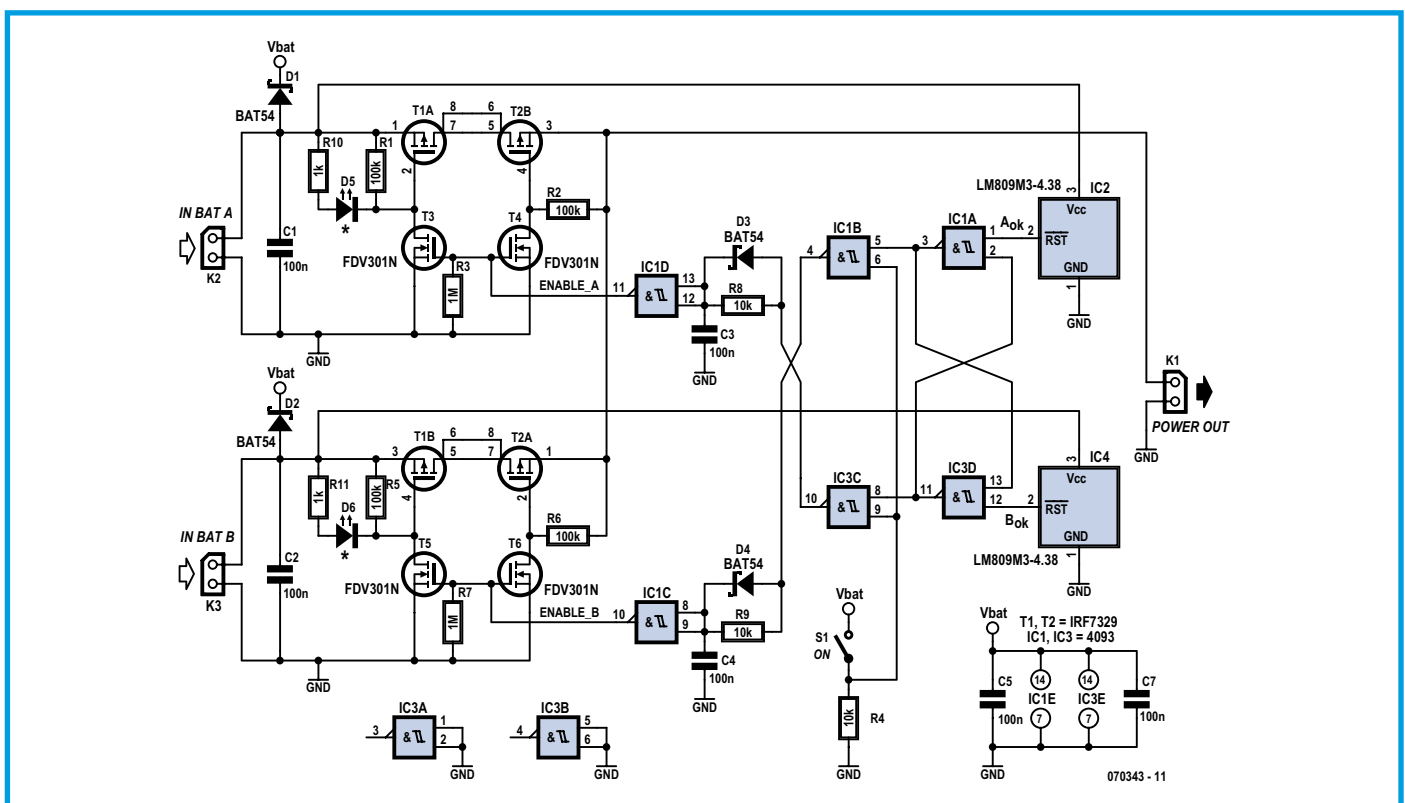
IC4 does the same thing, but for battery B.

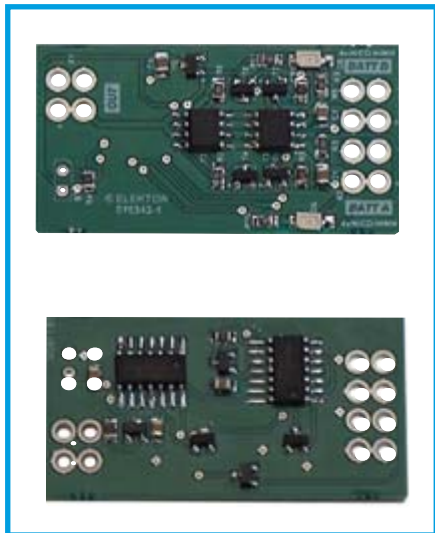
Both signals go to a flip-flop consisting of IC1a and IC3d, which determines which of the batteries is to be used.

If the voltage across battery A is too low, the output of IC1a will always be high. As a consequence, battery B will be active. The same thing applies in reverse to the output of IC3d.

When both batteries are discharged, they will both power the circuit, in keeping with the motto 'better a little bit of juice than no juice at all'.

Components D3, R8 and C3 provide





a switch-on delay that causes battery switch-on to be delayed somewhat. This is because it is undesirable to have both batteries power the circuit at the same time during switchover from one battery to the other. That would cause large equalization currents to flow due to the difference between the terminal voltages of the two batteries.

### Switch

The best choice for the switching device is a FET instead of a bipolar transistor. This saves energy, since no base current is nec-

essary. A disadvantage of a MOSFET is that it always has an intrinsic diode. This diode is quite annoying in this circuit, since the one battery can charge the other battery via the diode. A simple solution would be to wire a diode in series to prevent this. Unfortunately, a diode always has a voltage drop (approximately 0.3 V with a Schottky diode).

To solve this problem, we use a second MOSFET wired in the opposite direction. The underlying trick here is that the channel of a FET conducts in both directions when it is switched on. This eliminates the effect of the forward voltage of the internal diode.

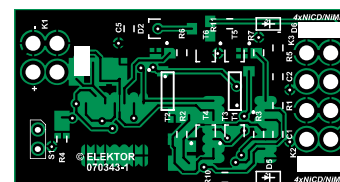
LEDs D5 and D6 indicate which battery is in use.

### Use

The circuit is very easy to use. Connect a four-cell NiCd battery to each of the battery inputs (K2 and K3). Then connect output K1 to the circuit to be powered.

Switch on the supply voltage with switch S1. The LEDs now indicate which battery is in use. If things ever get so far that both batteries become deeply discharged (Heaven forbid!), this can be recognised by the fact that both LEDs are lit.

(070343-1)



## Components list

(all R and C: SMD 0805 case)

### Resistors

R1, R2, R5, R6 = 100kΩ  
R3, R7 = 1MΩ  
R4, R8, R9 = 10kΩ  
R10, R11 = 1kΩ

### Capacitors

C1-C6 = 100nF

### Semiconductors

D1-D4 = BAT54 (SOT-23)  
D5, D6 = LED rood (SMD 1206)  
IC1, IC3 = 4093 (SOIC-14)  
IC2, IC4 = LM809M3-4.38 (SOT-23)  
T1, T2 = IRF7329 (SOIC-8)  
T3-T6 = FDV301N (SOT-23)

### Miscellaneous

Connecting wires  
PCB no. **070343-1** (see [www.elektor-electronics.co.uk](http://www.elektor-electronics.co.uk))

# Multi-purpose NiCd & NiMH Charger

C. Tavernier

[www.tavernier-c.com](http://www.tavernier-c.com)

Unless your robot is frugal enough to make do with primary cells without breaking the bank, or is environmentally-friendly and runs off solar panels, it will probably need to use rechargeable batteries as its energy source.

Although very many chargers are currently available, they're not always suitable for our needs, in terms of the types and number of batteries they can handle. What's more, certain of them do not take very good care of the batteries entrusted to them, which can seriously shorten their life.

So this article proposes building your own tailor-made charger, using an IC that's already old, but still very much current: the MAX713 from Maxim. As all robots

are different, we're not going to suggest a completely finished circuit, but will instead explain how to adapt it to suit the characteristics of the batteries you'll be wanting to recharge.

The MAX713's basic application circuit is shown in the figure, but as you can see, certain elements have no values shown. In addition, there are various configuration links. Via these various elements, the MAX713 lets you charge from one to 16 cells (a cell is a basic 1.2 V element), define the charging current, define the end-of-charge float current, and lastly, select the mode for detecting end of charge. As far as the latter is concerned, and so as to be compatible with any batteries you are likely to use in your robot, we've left out the temperature detection method, which requires a thermal sensor (NTC or equivalent) inside the battery. So resistors R4 and R5 in conjunction with

the hard-wired links to inputs TH1 and TLO program the MAX713 into the mode that detects end of charge by voltage variation.

So now let's see how to determine the other elements that are still open to you, so you'll be able to build a charger that's just right for your needs. Note right away that the configuration links can either be hard-wired on the PCB that you'll be designing for your charger, or else connected to multi-way switches to create a multi-purpose charger.

You first need to decide  $I_{fast}$ , the charging current for your batteries, whose capacity  $C$  is expressed in ampère-hours (Ah). This can be calculated from:  $I_{fast} = C/t$  where  $t$  is the desired charging time in hours. Watch out! The MAX713 does not handle times over 4 hours. And take care not to pick a value for  $I_{fast}$  above  $4C$ , which is currently the maximum current permitted for



fast-charging NiCad and NiMH batteries. If you are able to choose a lower current, so much the better, it will prolong battery life. Program this charging time by wiring pins PGM2 and PGM3 of the MAX713 as per **Table 1**.

Then choose the number of cells to be charged at the same time. For block batteries, you can find the number of cells by dividing the nominal voltage of the battery by 1.2 V. So a 9.6 V battery will contain eight cells. If the number of cells is 11 or more, the circuit can't be used as is, and in that case it's better to charge your batteries in two goes. Program this number by wiring pins PGM0 and PGM1 of the MAX713 as per **Table 2**.

Then choose the unstabilized DC supply voltage for your charger (VA in the figure) so that it is at least 1.5 V higher than the maximum voltage of the battery to be charged. If your battery has less than four cells, this rule no longer applies, as the MAX713 supply has to be a minimum of 6 V.

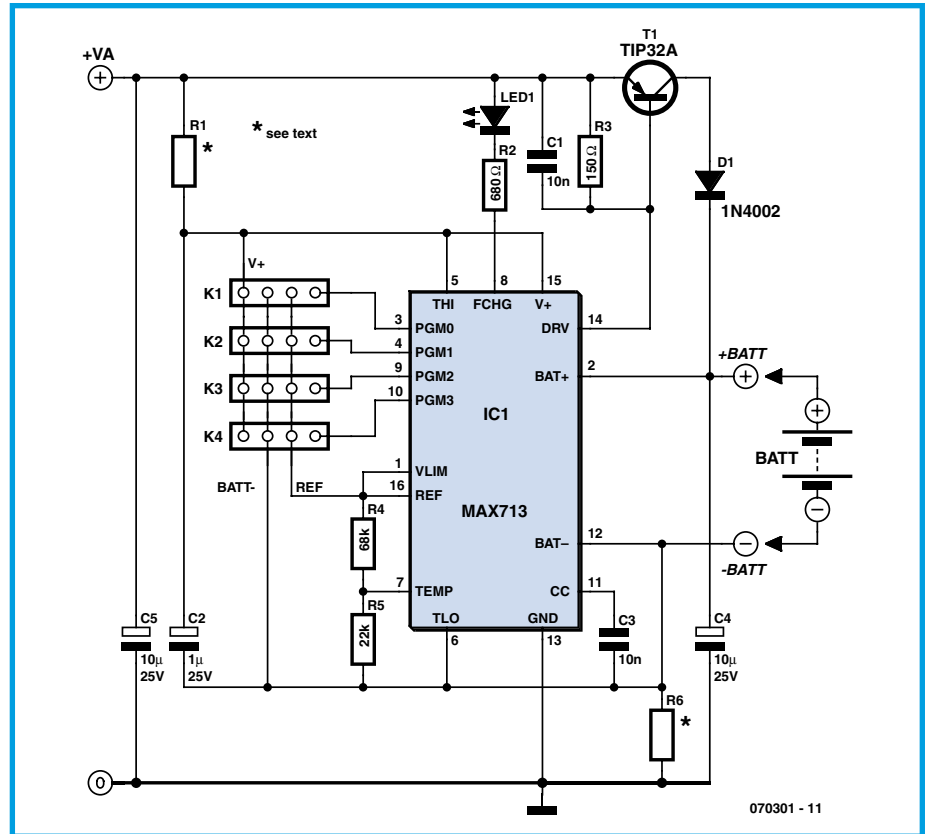
Then determine the maximum power dissipated in T1 using the following equation:

$$P_D = (V_A - V_{BATTmin}) \times I_{fast}$$

where  $V_{BATTmin}$  is the minimum voltage of the battery to be charged. Choose T1 accordingly, if necessary fitting it with an appropriate heatsink.

Then determine the value of resistor R1 so the current drawn by the MAX713 will be 5 to 20 mA, using the equation:  $R1 = (V_A - 5) / I$  where  $I$  is between 5 and 20 mA.

Lastly, determine the value of resistor R6 by using the equation:  $R6 = 0.25 / I_{fast}$  and its power by using  $P_{R6} = 0.5 I_{fast}^2$  (theoretically  $0.25 I_{fast}^2$  in fact, but it's best to use



a safety factor of 2, hence the modified equation).

Your charger is now operational, and is extremely simple to use; but because of the processors inside the MAX713, it is essential to make the connections to PGM0 to PGM3 before applying power to the circuit, otherwise they may not be taken into account correctly. This is no problem for a hard-wired circuit, but if your charger uses configuration switches at this point, you'll need to power down and power up again to confirm any configuration changes

made via these switches.

The LED lights when the charger is in fast-charge mode (at the current  $I_{fast}$  determined above). It goes out when fast-charging is over and the charger goes into float charge mode. The current generated in this mode is sufficiently low that the battery may be left connected to the charger indefinitely if necessary.

To make sure our explanation is crystal-clear, here by way of example are the calculations for a charger for a pack of four 1.2 V NiMH batteries with a capacity of

Maximum charge time (min)	PGM3	PGM2
22	V+	REF
33	V+	BATT-
45	O/C	REF
66	O/C	BATT-
90	REF	REF
132	REF	BATT-
180	BATT-	REF
264	BATT-	BATT-

Number of cells	PGM1	PGM0
1	V+	V+
2	O/C	V+
3	REF	V+
4	BATT-	V+
5	V+	O/C
6	O/C	O/C
7	REF	O/C
8	BATT-	O/C
9	V+	REF
10	O/C	REF
11	REF	REF
12	BATT-	REF
13	V+	BATT-
14	O/C	BATT-
15	REF	BATT-
16	BATT-	BATT-

1,800 mAh that we want to recharge in two hours.

- Calculate  $I_{fast}$ :  $I_{fast} = C/t$ , i.e.,  $1.8/2 = 0.9$  A or 900 mA.
- PGM2 and PGM3 connections: PGM2 tied to BATT- and PGM3 tied to REF, as we want a charge time of 2 hours, i.e. 120 minutes (in fact, we'll get a maximum of 132 minutes).
- PGM0 and PGM1 connections: PGM0 to V+ and PGM1 to BATT- since our battery comprises four cells.
- Determine  $V_A$ :  $V_A = 6.3$  V minimum.

We'll choose 9 V, to obviate any problems with possible supply voltage variations.

- Power dissipated in T1:  $P_D = (9 - 4*) \cdot 0.9$ , i.e. 4.5 W. So we'll choose, for example, a TIP32A, giving us an excellent safety margin ( $P_{Dmax} = 40$  W).
- \* fully discharged battery voltage estimated at 4 V.
- Calculate R1:  $R1 = (9-5)/0.01^{**}$ , = 400  $\Omega$ . We'll use the closest preferred value, i.e. 390  $\Omega$ .
- \*\* : a current of 10 mA was chosen.
- Calculate R6:  $R6 = 0.25/0.9$ , = 0.27  $\Omega$ .

- Calculate the power in R6:  $P_{R6} = 0.5 \times 0.9 = 0.45$  W. So a 0.50-W type is going to be fine.
- As you can see, it's taken us all of five minutes to produce a charger tailored perfectly to our needs. Now it's your turn...

(070301-1)

#### Web Links

MAX713 spec. sheet:

[www.maxim-ic.com/quick\\_view2.cfm/qv\\_pk/1666](http://www.maxim-ic.com/quick_view2.cfm/qv_pk/1666)

# Fast Charger for NiMH Batteries

Antoine Authier  
& Karel Walraven

At the time of writing, the latest AA NiMH (Nickel metal Hydride) batteries have a capacity of up to 2900 mAh. Using an original-type conventional battery charger (supplying 125 mA), the charging time will be extremely long.

The charger we propose here should accelerate the recharging process of NiMH batteries, which are becoming more and more common (we must do our bit for the environment).

The design is based on the MAX712 made by Maxim (Integrated Products to be precise, which was bought by Dallas Semiconductor; quite a long story), operating in switched mode, it can supply a maximum fast charge current calculated as

$$I_{charge} = 250 \text{ mV} / R1$$

or not less than 1 A if  $R1 = 0.25$  ohms. Under these conditions, the battery will be charged in just over two hours.

The Maxim circuit is not only intelligent, but it also includes an ADC (analogue to digital converter), a system to detect charge completion, a timer, and a temperature monitoring module. The four configuration pins that are included allow users to set the parameters as they please. These pins are used to set the parameters for the number of cells to be charged, the maximum charging period, as well as the method to detect when it is fully charged (inflection point or negative slope). You can refer to the datasheet to find out more. The MAX712 is intended for NiMH batteries, with charge completion at the inflection point of the voltage curve ( $\partial V/\partial t = 0$ ).

The maximum power supply voltage is 15 V. The power supply voltage must be

at least 2 V above the maximum charging voltage in order to compensate for voltage fluctuations during charging. Therefore, for a maximum charging voltage of 1.6 V per cell, a 15-V power supply voltage is used to charge 8 series-connected batteries. A 12-V voltage level (supplied, for example, by a car battery) is used to recharge six cells. The power supply must be able to supply 1 A. It is important to be certain of its specification. If the requirement is not fulfilled, the integrated circuit will not operate correctly and may not correctly detect completion of the fast-charge (entailing a risk of damage that could affect the connected batteries).

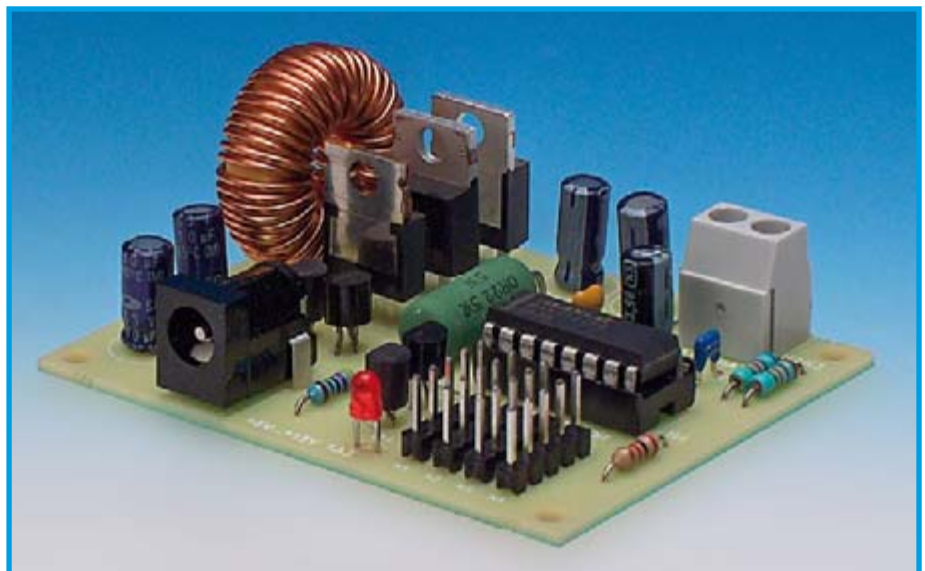
### Setting the circuit parameters

- The PRGM0/PRGM1 pins are used to regulate the number of cells to be charged. A note concerning the use of a battery cradle: during recharging: each contact can

represent a 1- $\Omega$  series resistance, which is seen as a 1-V potential difference at 1 A. The power supply voltage may not be adequate for this configuration — therefore, it is preferable to verify this detail before beginning the project.

- For security reasons, it is preferable to properly configure the maximum charging period with the PRGM2/PRGM3 pins.

- On this setup, the temperature control circuit for the batteries is deactivated. At the end of the fast-charge, the circuit will power the batteries with a maintenance charge (trickle). Let's examine the circuit's electronics. T1 is used as a current source supplying the 8 mA necessary to power the MAX712. D3 ensures that the battery does not discharge into the circuit in case it is not powered. The LED D1 lights up when the circuit is in fast-charge mode. T5 may be mounted

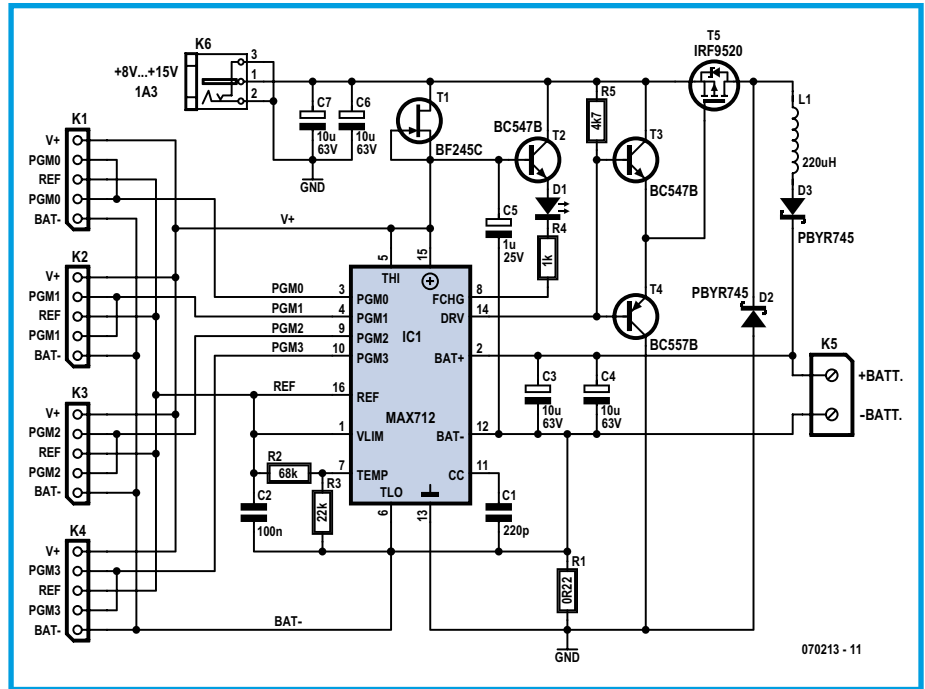


on heatsink, if necessary. The characteristics of coil L1 are not critical; a traditional 100  $\mu$ H/5 A suppressor choke will work fine. The same holds true for diodes D2, D3 and the MOSFET transistor T5; they too are not critical in this application. You can use any Schottky diode that can withstand 3 amps and include any MOSFET with a lower drain resistance.

A compact PCB was designed for the circuit. Mounting the components should be all plain sailing, but do not forget the two wire links on the board. Inductor L1 is a toroid 'suppressor choke' with a good size. Connectors K1-K4 allow different charging parameters to be set up.

Since the calculation principle is the same as for the NiCd charger in the MAX713 in the other article, we refer you to the calculation example proposed there. Use the same tables to set the parameters of this circuit as the ones given in that article.

(070213-1)



## Components list

### Resistors

- R1 = 0.22 $\Omega$  5W
- R2 = 68k $\Omega$
- R3 = 22k $\Omega$
- R4 = 1k $\Omega$
- R5 = 4k $\Omega$

### Capacitors

- C1 = 220pF
- C2 = 100nF
- C3, C4, C6, C7 = 10 $\mu$ F 63V radial
- C5 = 1 $\mu$ F 25V radial

### Inductor

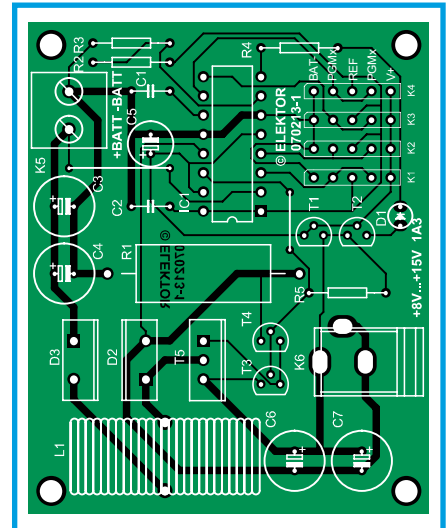
- L1 = 220 $\mu$ H suppressor choke

### Semiconductors

- D1 = LED
- D2, D3 = PBYR745
- T1 = BF245B or -C
- T2, T3 = BC547B
- T4 = BC557B
- T5 = IRF9520
- IC1 = MAX712CPE

### Miscellaneous

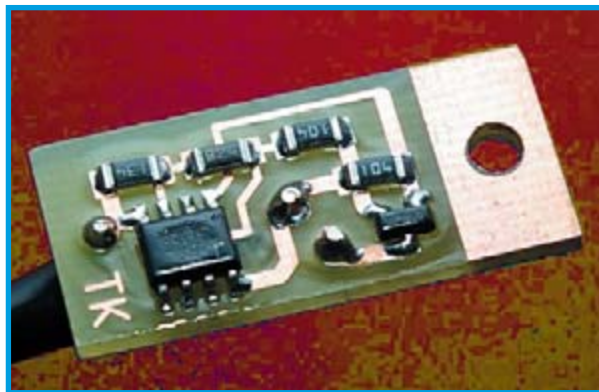
- K1-K4 = 5-way SIL pinheader
- K5 = 2-way PCB terminal block, lead pitch 5mm
- K6 = DC supply jack, PCB mount
- PCB no. **070213-1**, see Elektor SHOP



# Deep Discharge Protection for Rechargeable Cells

Tilman Küpper

With this circuit built into the power supply of a battery powered device, it will prevent the rechargeable cells from being completely drained when you forget to turn the equipment off. When the battery voltage drops below a pre-set limit (9.5 V in this example) the circuit will automatically disconnect the battery. Power is re-connected when the voltage rises above an upper threshold level (10.5 V here),



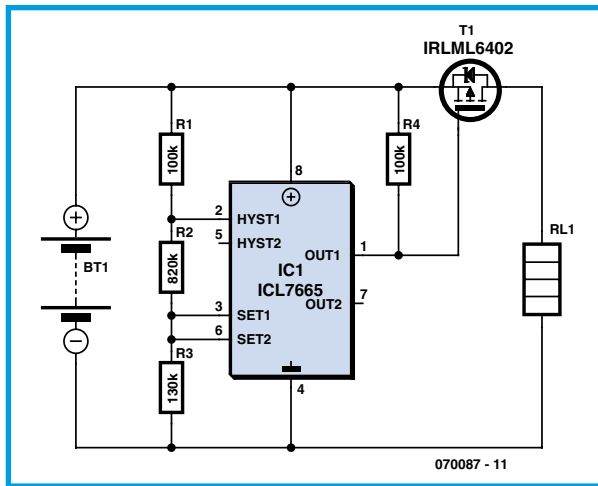
this will typically occur after the equipment has been plugged into its recharging station. The circuit is designed to use as little power as possible.

The ICL7665 from Intersil forms the heart of the circuit. This IC contains two comparators together with a voltage reference and consumes just 3  $\mu$ A. The circuit only uses one of the comparators, the values of resistors R1 to R3 shown in the diagram will cause the circuit to switch



at the levels mentioned above. The comparator output switches the P-channel MOSFET T1 which in turn controls power to the load  $R_{LOAD}$ .

The switching threshold levels and hysteresis can be changed by using different values of resistor for R1 to R3. Increasing the value of R3 to 300 K $\Omega$  will raise the upper threshold level to 12.5 V. The ICL7665 data sheet gives examples of suitable resistor values that can be used here. The PCB layout uses SMD components so the finished circuit takes up very little space when installed in the equipment. A



fine-tipped soldering iron should be adequate to mount the components and there shouldn't be any problems provided you do not choose to use very small resistor packages. Once the circuit has been tested the entire PCB can be protected by encapsulating it with a short length of heat shrink sleeving.

(070087-1)

#### Links

Data sheet IRL7665: [www.intersil.com/data/fn/fn3182.pdf](http://www.intersil.com/data/fn/fn3182.pdf)

Data sheet IRLML6402: [www.irf.com/product-info/datasheets/data/irlml6402.pdf](http://www.irf.com/product-info/datasheets/data/irlml6402.pdf)

# LDO Regulator with Soft Start or Tracking

Dirk Gehrke  
(Texas Instruments Germany)

The devices described here are tailored to fit the needs of modern DSPs, processors and FPGAs that require low supply voltages at relatively high currents, plus the capability of supply voltage ramp-up and sequencing in a defined manner. The latter two requirements are not easily solved using discrete components. The TPS74x01 family fits the bill.

At the time of writing the family consists of three parts, called TPS74201 (1.5 A with soft-start), TPS74301 (1.5 A with tracking) and TPS74401 (3 A with soft-start). The family is expected to grow soon.

Figure 1 shows a simplified connection diagram for the three devices. These regulators require a low power bias voltage,  $V_{BIAS}$ , and a power input voltage,  $V_{IN}$ , from which  $V_{OUT}$  will be derived. All three regulators are capable of providing output voltages down to 0.8 V and the device in the QFN package also includes an integrated supervisory circuit with open-drain output that goes to high impedance when the output voltage reaches regulation ('power good' or PG). The TPS74301 can provide up to 1.5 A output current and has a TRACK pin which allows the user to input a ramp signal for the output voltage to follow, effectively implementing either simultaneous or ratiometric sequencing.

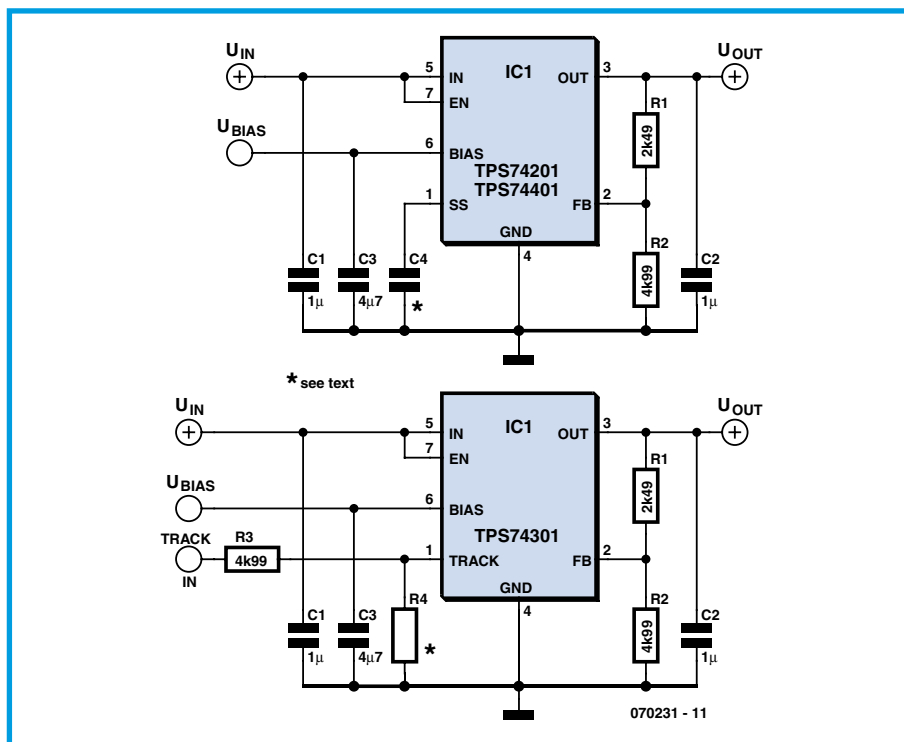
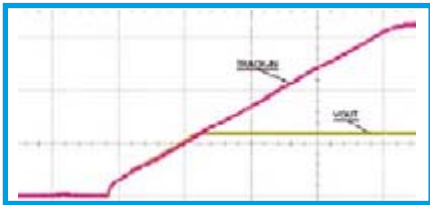


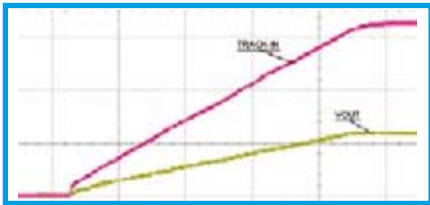
Figure 1. Schematic of TPS74201, TPS74301 and TPS74401 (see text for values of R4 and C4).

The relevant connections are shown separately. The TPS74201 and TPS74401 can provide up to 1.5 A and 3.0 A DC current, respectively, and have an SS pin which allows the user to set the linear ramp rate of the output voltage.

For the TPS74301, the value of R4 (in a voltage divider) allows the user to select either simultaneous or ratiometric sequencing. R4's value is calculated from the equations in the datasheet, assuming an external 3.3-V ramp signal is applied



**Figure 2.** Tracking LDO TPS74301 with simultaneous startup ( $R4 = 10k\Omega$ ).  
Timebase: 2 mV/div.



**Figure 3.** Tracking LDO TPS74301 with ratiometric startup ( $R4 = 1.78k\Omega$ ).  
Timebase: 2 mV/div.

to the TRACK IN pin. When the value of  $R4$  selected to be  $10\text{ k}\Omega$  in this particular example, the output of the TPS74301 will follow the external ramp signal within a few millivolts until the TPS74301 reaches its regulated voltage. This is called *Simultaneous Sequencing* (Figure 2). The TPS74301 output voltage will have the same rate of rise ( $dv/dt$ ) as the external ramp signal but

a different soft-start time. Changing resistor  $R4$  to a value of  $1.78\text{ k}\Omega$  results in the TPS74301 output voltage reaching its regulated voltage at the same time the externally applied tracking signal reaches its maximum voltage (e.g., 3.3 V). This is called *Ratiometric Sequencing* (Figure 3). Although the external ramp signal and the TPS74301 will have different rates of rise, they will have the same soft-start time.

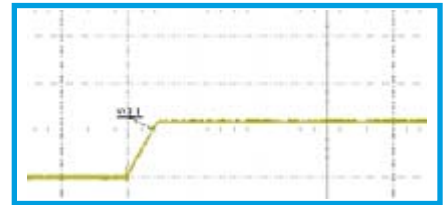
For the TPS74201 and TPS74401 the capacitor value can be used to program the soft-start ramp. In this example, values of 1 nF and 10 nF for  $C4$  were used to realize ramps with 1 ms and 10 ms rise time respectively (Figures 4 and 5 respectively). With no capacitor attached to this pin the default soft-start time will be 500  $\mu\text{s}$ . This 500- $\mu\text{s}$  start up time is also valid for the TPS74301 when applying a voltage greater than 800 mV to the TRACK pin.

(070231-1)

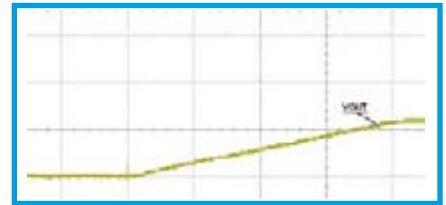
#### Literature

TPS74401 3.0A Ultra-Low Dropout Linear Regulator, Texas Instruments Literature # SBVS066C.

TPS74301 1.5A Ultra-Low Dropout Linear Regulator with Programmable Sequencing, Texas Instruments Literature # SBVS065C.



**Figure 4.** Soft-Start LDO TPS74201 and TPS74401 with 1ms soft startup ( $C4 = 1\text{nF}$ ).  
Timebase: 2 mV/div.



**Figure 5.** Soft-Start LDO TPS74201 and TPS74401 with 10ms soft startup ( $C4 = 10\text{nF}$ ).  
Timebase: 2 mV/div.

TPS74201 1.5A LDO with Programmable Soft-Start, Texas Instruments Literature # SBVS064C

TPS74x01EVM-118 User's Guide, Texas Instruments Literature # SLVU143.

Sequencing Power Supplies in Multiple Voltage Rail Environments by David Daniels, Dirk Gehrke and Mike Segal, Texas Instruments Literature # SLUP228 or <http://focus.ti.com/lit/ml/slup228/slup228.pdf>

NEW



#### Ordering:

Use the order form at the back or visit our online shop.

Elektor Electronics (Publishing)  
Regus Brentford | 1000 Great West Road  
Brentford TW8 9HH | United Kingdom  
Tel. +44 208 261 4509  
sales@elektor-electronics.co.uk

## Formula Flowcode Buggy

USB-programmable robot vehicle

- A complete solution: robot + software + curriculum
- Line following and maze solving
- High-tech specifications
- Also programmable with C or ASM
- E-blocks compatible
- Motivating for education and hobby

Built and ready to use for only  
**£ 85.00 / US\$ 169.00**

**elektor**  
electronics

More information on [www.elektor-electronics.co.uk](http://www.elektor-electronics.co.uk)





## Motor Drivers/Controllers

Here are just a few of our controller and driver modules for AC, DC, Unipolar/Bipolar stepper motors and servo motors. See website for full details.

### NEW! PC / Standalone Unipolar Stepper Motor Driver

Drives any 5, 6 or 8-lead unipolar stepper motor rated up to 6 Amps max. Provides speed and direction control. Operates in stand-alone or PC-controlled mode. Up to six 3179 driver boards can be connected to a single parallel port. Supply: 9Vdc. PCB: 80x50mm.  
Kit Order Code: 3179KT - **£11.95**  
Assembled Order Code: AS3179 - **£18.95**



### NEW! Bi-Polar Stepper Motor Driver

Drive any bi-polar stepper motor using externally supplied 5V levels for stepping and direction control. These usually come from software running on a computer. Supply: 8-30Vdc. PCB: 75x85mm.  
Kit Order Code: 3158KT - **£17.95**  
Assembled Order Code: AS3158 - **£27.95**



### NEW! Bidirectional DC Motor Controller

Controls the speed of most common DC motors (rated up to 16Vdc/5A) in both the forward and reverse direction. The range of control is from fully OFF to fully ON in both directions. The direction and speed are controlled using a single potentiometer. Screw terminal block for connections.  
Kit Order Code: 3166KT - **£16.95**  
Assembled Order Code: AS3166 - **£25.95**



### DC Motor Speed Controller (100V/7.5A)

Control the speed of almost any common DC motor rated up to 100V/7.5A. Pulse width modulation output for maximum motor torque at all speeds. Supply: 5-15Vdc. Box supplied. Dimensions (mm): 60Wx100Lx60H.  
Kit Order Code: 3067KT - **£13.95**  
Assembled Order Code: AS3067 - **£21.95**



Most items are available in kit form (KT suffix) or assembled and ready for use (AS prefix).

## Controllers & Loggers

Here are just a few of the controller and data acquisition and control units we have. See website for full details. Suitable PSU for all units: Order Code PSU445 £8.95

### Serial Isolated I/O Relay Module

Computer controlled 8-channel relay board. 5A mains rated relay outputs. 4 isolated digital inputs. Useful in a variety of control and sensing applications. Controlled via serial port for programming (using our new Windows interface, terminal emulator or batch files). Includes plastic case 130x100x30mm. Power Supply: 12Vdc/500mA.  
Kit Order Code: 3108KT - **£54.95**  
Assembled Order Code: AS3108 - **£64.95**



### Computer Temperature Data Logger

4-channel temperature logger for serial port. °C or °F. Continuously logs up to 4 separate sensors located 200m+ from board. Wide range of free software applications for storing/using data. PCB just 38x38mm. Powered by PC. Includes one DS1820 sensor and four header cables.  
Kit Order Code: 3145KT - **£18.95**  
Assembled Order Code: AS3145 - **£25.95**  
Additional DS1820 Sensors - **£3.95 each**



### Rolling Code 4-Channel UHF Remote

State-of-the-Art. High security. 4 channels. Momentary or latching relay output. Range up to 40m. Up to 15 Tx's can be learnt by one Rx (kit includes one Tx but more available separately). 4 indicator LED's. Rx: PCB 77x85mm, 12Vdc/6mA (standby). Two and Ten channel versions also available.  
Kit Order Code: 3180KT - **£44.95**  
Assembled Order Code: AS3180 - **£51.95**



### NEW! DTMF Telephone Relay Switcher

Call your phone number using a DTMF phone from anywhere in the world and remotely turn on/off any of the 4 relays as desired. User settable Security Password, Anti-Tamper, Rings to Answer, Auto Hang-up and Lockout. Includes plastic case. Power: 12Vdc approved. 130x110x30mm. Power: 12Vdc.  
Kit Order Code: 3140KT - **£54.95**  
Assembled Order Code: AS3140 - **£69.95**



## Infrared RC Relay Board

Individually control 12 on-board relays with included infrared remote control unit. Toggle or momentary. 15m+ range. 112x122mm. Supply: 12Vdc/0.5A  
Kit Order Code: 3142KT - **£47.95**  
Assembled Order Code: AS3142 - **£59.95**



## PIC & ATMEL Programmers

We have a wide range of low cost PIC and ATMEL Programmers. Complete range and documentation available from our web site.

### Programmer Accessories:

40-pin Wide ZIF socket (ZIF40W) **£15.00**  
18Vdc Power supply (PSU010) **£19.95**  
Leads: Parallel (LDC136) **£4.95** / Serial (LDC441) **£4.95** / USB (LDC644) **£2.95**

### NEW! USB & Serial Port PIC Programmer

USB/Serial connection. Header cable for ICSP. Free Windows XP software. Wide range of supported PICs - see website for complete listing. ZIF Socket/USB lead not included. Supply: 16-18Vdc.  
Kit Order Code: 3149EKT - **£37.95**  
Assembled Order Code: AS3149E - **£49.95**



### NEW! USB 'All-Flash' PIC Programmer

USB PIC programmer for all 'Flash' devices. No external power supply making it truly portable. Supplied with box and Windows Software. ZIF Socket and USB lead not included.  
Assembled Order Code: AS3128 - **£44.95**



### "PICALL" PIC Programmer

"PICALL" will program virtually all 8 to 40 pin serial-mode AND parallel-mode (PIC16C5x family) programmed PIC micro controllers. Free fully functional software. Blank chip auto detect for super fast bulk programming. Parallel port connection. Supply: 16-18Vdc.  
Assembled Order Code: AS3117 - **£24.95**



### ATMEL 89xxxx Programmer

Uses serial port and any standard terminal comms program. Program/ Read/ Verify Code Data, Write Fuse/Lock Bits, Erase and Blank Check. 4 LED's display the status. ZIF sockets not included. Supply: 16-18Vdc.  
Kit Order Code: 3123KT - **£24.95**  
Assembled Order Code: AS3123 - **£34.95**





# Bolo

Abraham Vreugdenhil

Bolo is a light-seeking robot built into a plastic ball. The advantage of a ball is that if it gets stuck, it can always go back the same way it came. If you put a robot **inside** a ball, it can always back up and then roll away from any obstacle it runs into.

## Drive mechanism

To enable the robot to drive the ball, the shafts of the motors are fitted with lengths of bicycle valve tubing to give them grip. The shafts rest directly on the inner surface of the ball. The robot also has a single support wheel made from a plastic bead. A round hoop is fitted to the top of the robot so it will always land on its 'feet' (wheels) if it is thrown with a swinging motion.

## Motors

The rotational speed of the motors is on the high side. If you were to let the robot run continuously, it would pass through the available space rather quickly. To avoid this, the motors are switched on for one second and then off for one second. After they are switched off, the light level is measured and a new decision is made as to which direction the ball should roll for another second.

## Electronics

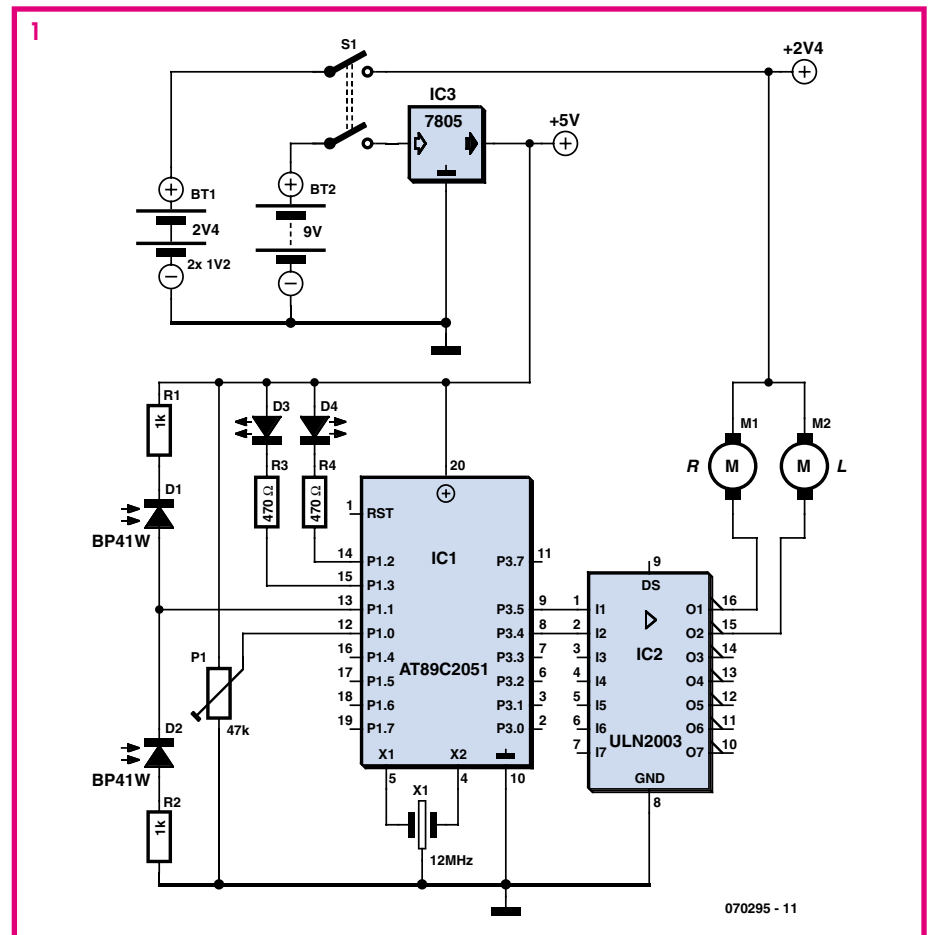
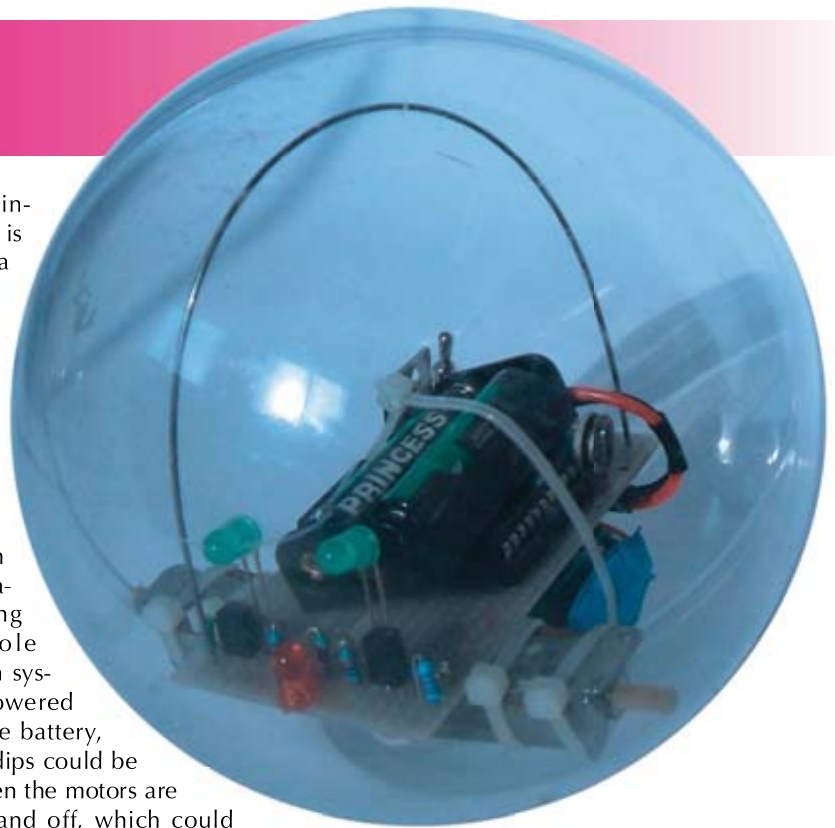
The selected microcontroller is an 89C2051. Among other things, it incorporates a comparator that is used in this design. Two BPW41 photodiodes connected in series are used as the light sensors. The junction of the two diodes is connected to one input of the comparator. A 1-k $\Omega$  resistor is connected in series with each BPW41 in order to limit the current through the sensors if the light is excessively bright. A 47-k $\Omega$  potentiometer is connected to the second comparator input. This is used to set the light sensitivity. The two eyes are formed by LEDs, which are connected to the microcontroller by a 470- $\Omega$  resistor. The two motors are driven by the microcontroller via Darlington transistors. The author did not have these on hand, so an IC with an array of seven Darlington transistors (type ULN2003) was used for the prototype, with only two of the transistors actually being used. If more power is needed, two or more inputs and outputs can simply be connected in parallel to boost the power. The motors are powered by two AAA batteries in series,

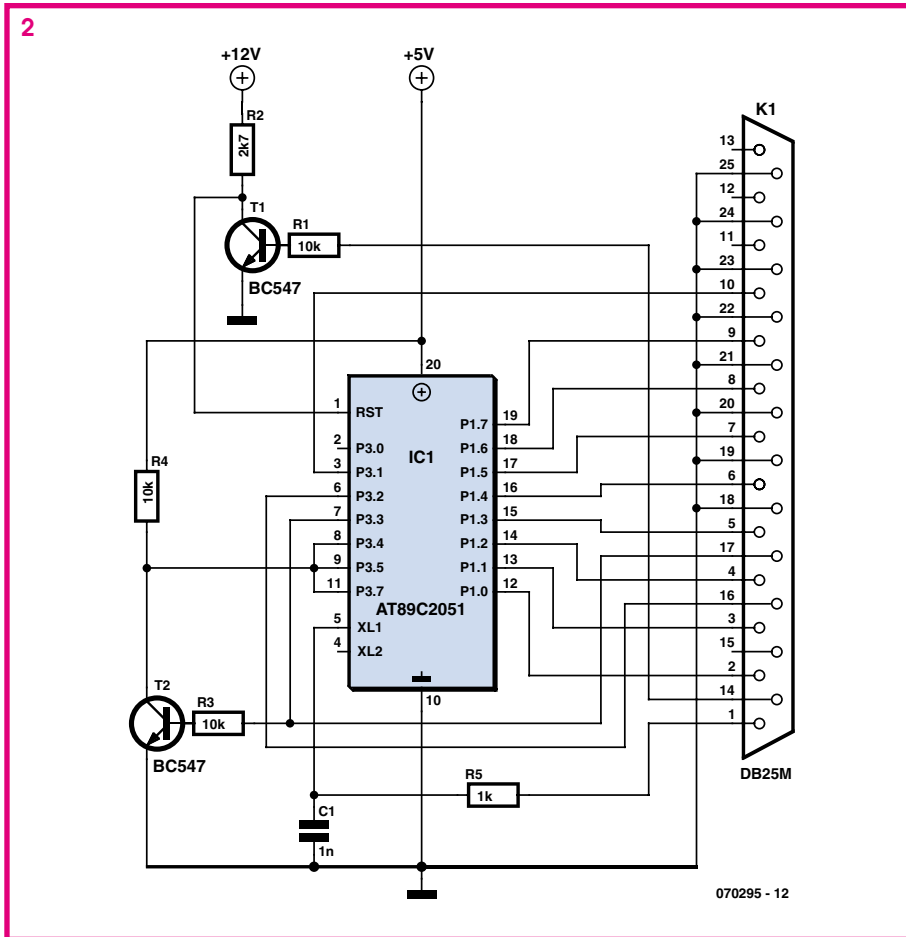
and the remaining electronics is powered by a 9-V block battery. A 7805 voltage regulator generates a stabilised voltage from this battery. Both voltages can be switched on and off simultaneously using a double-pole switch. If both systems were powered from the same battery, brief voltage dips could be produced when the motors are switched on and off, which could cause the microcontroller to be reset. To prevent this, it's a good idea to use separate supplies for the two systems.

## Behaviour and/or extension

A possible extension would be an RC5

receiver, so the robot could receive commands from a 'standard' remote control unit. This would make it possible to steer the ball in a particular direction or select one of several different behaviour patterns





(such as sight seeking, light avoiding, or random). There are many conceivable

behaviour patterns for robots. The compiler used for the microcontroller in this

robot, BASCOM-8051, provides a specific command for receiving RC5 signals. Such an interface would thus be easy to implement. As it stands, Bolo only behaves as a light-seeking robot.

## Programming

The 89C2051 can be programmed using a simple programmer. BlowIt is a programmer consisting of just a few voltage regulators, two transistors, and a few resistors and capacitors [1]. Of course, it also has a 20-pin IC socket and a 25-way D-sub connector. All of this can be assembled on a piece of perforated prototyping board. The author also added a transistor and an LED to indicate when data is being written to the microcontroller. The schematic diagram for the BlowIt programmer and additional information are available on the Internet. This programmer can easily be driven by BASCOM-8051 or BASCOM-LT.

The software for Bolo can be downloaded free of charge from the *Elektor Electronics* website as file no. **070295-11**.

(070295-1)

## Web links

(1) [www.geocities.com/dinceraydin/8051/index.html](http://www.geocities.com/dinceraydin/8051/index.html)

Het programma voor Bolo is gratis te downloaden van de Elektuur-website (EPS 070295-11).

# Antieu-robot

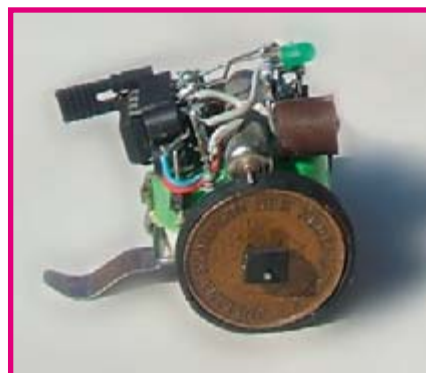
Abraham Vreugdenhil

The name of this robot is actually a bit ambiguous. The hyphen in the name can also be placed in a different position, giving the name 'anti-Eurobot'. This refers to the wheels, which have been made from pre-Euro coins.

The motto of this robot is 'small but functional'. How small can we make a standalone robot?

To start with we'll need very small motors. The author found that the motors used as vibrators in mobile phones were perfectly suitable. These so-called pager-motors are small and run very fast. They normally come with a small weight, which is mounted off-centre on the spindle to generate the vibrations. This can easily be removed with a pair of cutters.

The two motors are driven by a pair of sim-



ple BC557 or BC537 transistors. At the base of the transistor we connect a 10 k pull-up resistor. We also require a small processor. The AVR range from Atmel includes an 8-pin version, the 90S2343. Apart from the two supply pins and the reset pin it has 5 I/O pins. This may not look like much, but

a simple robot doesn't need many.

The biggest problem is the power source. For this we've chosen a 3.6 V 160 mAh NiMH battery, which has small dimensions. This battery is used as a frame, with the rest of the circuit built around it.

The last component we need is a sensor to detect obstacles. For this we used an IS471F infrared sensor. It reacts to an IR signal that's generated by an IR LED controlled via the same sensor.

And finally we come to the wheels. In keeping with the name of the robot, the author used some old-fashioned Dutch 5-cent pieces for these. We realise that these are difficult to get hold of outside The Netherlands and any other copper coins with a diameter of about 21mm will do as well. Two coins were glued together and we drilled a 1 mm hole through the centre for the axle. A small ring can be cut

from a bicycle inner tube and glued to the coins to make the tyre. The neck of a balloon is also suitable to give the wheels more grip.

If we let the robot travel too fast and then stop it, it would be liable to topple due to its weight and high centre of gravity. Because of this we've limited its speed. This also results in a less nervous looking robot.

The revs of the motors are much too high with a 3.6 V supply. Using a pulse width modulation of 25% reduces it to a manageable level.

We don't let the Antieu-robot ride continuously either, but let it stop at certain intervals. The main reason for this is that it stops it racing to the edge of its area; the other reason is that it gives it a somewhat intelligent and thoughtful characteristic. It almost appears as if it's studying its surroundings before continuing on its journey. When an obstacle is encountered it will

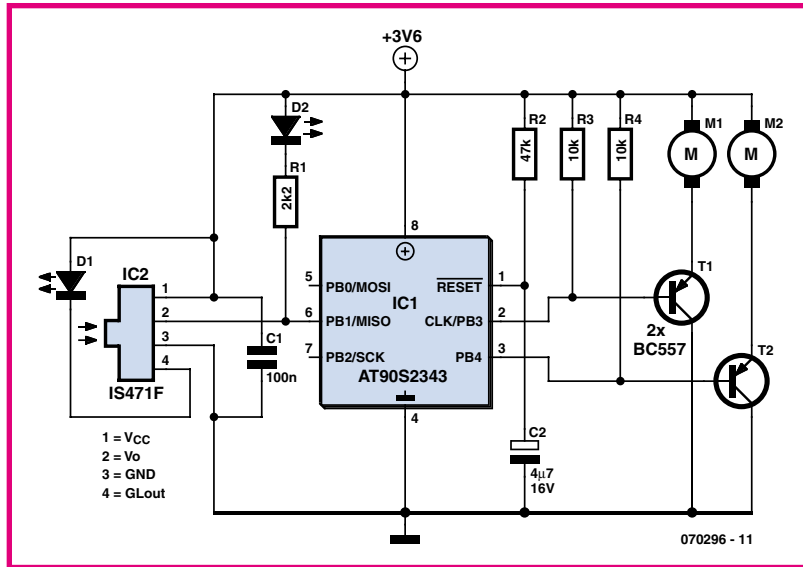
turn left or right, depending on an internal counter. If it still sees the obstacle it continues turning in the same direction until the obstacle disappears from view.

The program gives the robot a simple object avoiding behaviour. The robot rides along until it sees an object, which it will then try to get around. Only about 600 bytes have

responding power source hanging at the right height somewhere in the room. When it makes contact with this power source the robot could take a bit of a rest whilst it recharges its battery.

The program for the Antieu-robot can be downloaded from the Elektor website as file number **070296-11.zip**

(070296-1)

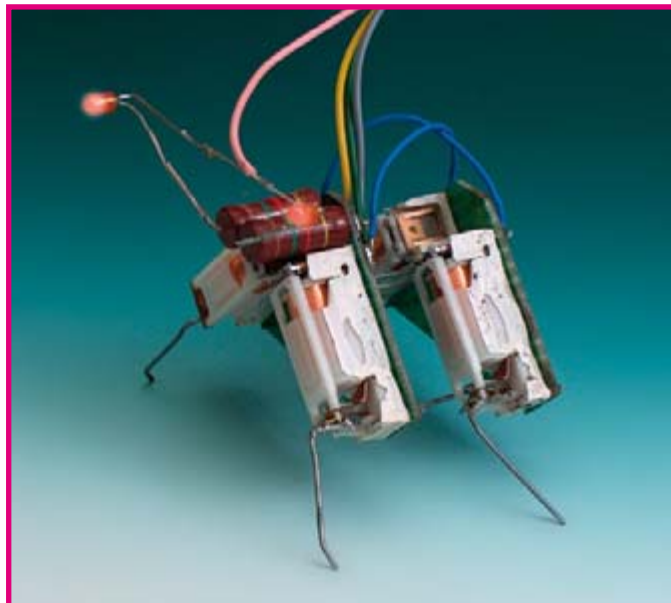


## Babybot

There are many conceivable and feasible techniques to give motion to a robot. In most cases ordinary electric motors, servos and stepper motors offer the simplest solution. In particular if the actuators are required to deliver a considerable amount of power. But for very small objects other types of actuators also become suitable, such as the one used in this mini robot.

The term 'robot' may be a little of an exaggeration in this case. This creature has to be controlled from the outside and does not have any other artificial intelligence, but the method of motion is quite unusual and with a little modification could also prove useful in other projects.

BabyBot walks with the aid of four small PCB relays, which have been specially modified for this purpose. The covers have been removed and the legs have



been soldered directly to the switching contacts. It will be clear that the step size of the leg will be minimal. After all, the 'stroke' that the switching contact can make is very small.

The legs cannot support and move much weight either. On the one hand this has to do with the limited pulling power of the electromagnet and on the other hand with the fragile construction of the inner workings of these small relays. It is likely that the link from magnet to switching contact will fall apart from time to time.

And this means — depending on the type of relay — a lot of fiddling to put it back in the right place. BabyBot is by no means a real robust construction, but it is useful for experimentation and anyhow, the whole thing looks quite nice.

(070278-1)

**Video clip of the walking mini robot:**

[http://www.xj3.nl/\\_dreijer/backsite/willem/babybot.avi](http://www.xj3.nl/_dreijer/backsite/willem/babybot.avi)



# Torque is Cheap

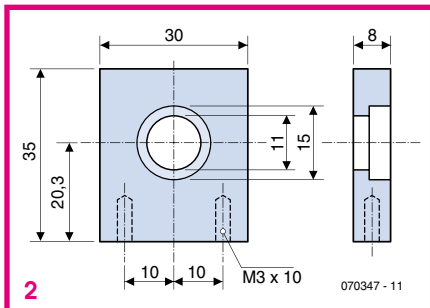
Bernhard Spitzer

While on the lookout for low-cost drive solutions the author's eye fell upon a cheap battery-operated cordless screwdriver in an electronics shop (**Figure 1**). These are available for just a few pounds (for example, Conrad order code 481576) and include a powerful '380' motor and a multiple-stage planetary drive. The classification of motors as '380', '540' and so on follows the part numbers in the range produced by Mabuchi Motors. A '380' motor corresponds to the Mabuchi RS380 and has a diameter of 29 mm and a length of approximately 40 mm. The nomi-



With the gearbox removed the two black halves of the case can easily be sepa-

reversing switch. The two halves of the case should now be cut away in such a way that the motor mounting remains intact. Drill a hole at the lowest point of the lower half for a fixing screw. Finally we need a bearing mount to support the wheel, for which we use a small piece of plastic (see the drawing in **Figure 2** for dimensions). For smooth running we use a 15 mm × 10 mm × 4 mm ball bearing designed for use in model cars (for example made by Tamiya). The 15 mm diameter recess (**Figure 3**) can be made using a Forstner bit. The hole in the mounting block should be made at half the motor enclosure



nal voltage is 6 V with a maximum current of approximately 4 A, with a power of between 10 W and 15 W depending on the model: see, for example, Conrad Electronics order code 244511.

To drive a wheel on a robot model we require a bearing on the drive axle. We must also dispense with all the unnecessary parts that come with the motor: first remove the battery case and then the gearbox, by pushing out the two pins (seen in **Figure 1** and **Figure 5** near to where the black and orange parts of the case meet) using a 2 mm pin.

rated. Now the motor connections must be straightened out in order to remove the

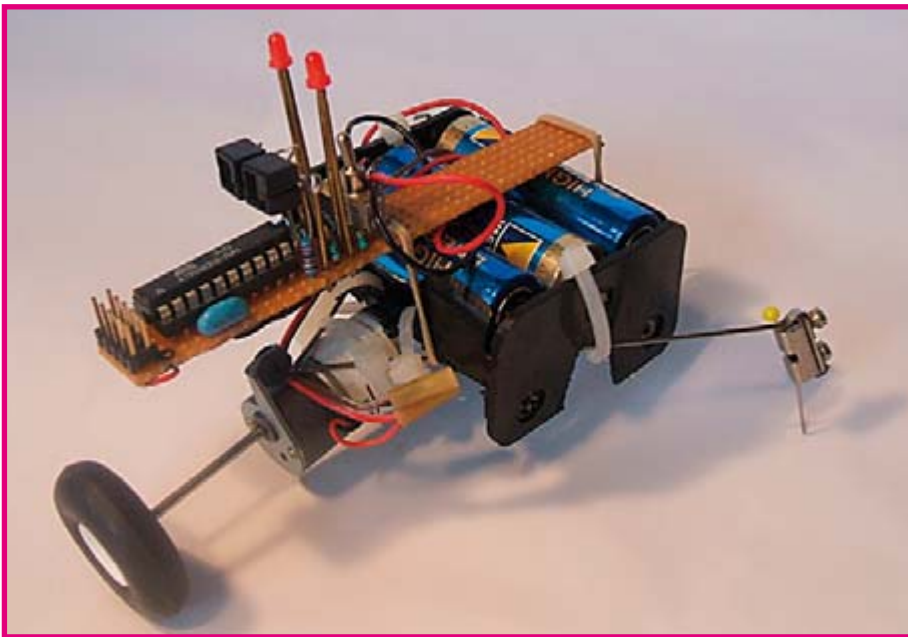


diameter from the edge (here 20.3 mm) so that the shaft will later be able to pass exactly through it. The finished bearing mount with bearing fitted is shown in **Figure 4**. The two M3×16 screws are used to fix the mount to the baseplate of the vehicle.

**Figure 5** shows the motor and gearbox ready for installation, with bearing mount fitted. Either wheels with a 10 mm axle hole can be used, glued directly to the shaft, or the wheel can be glued to a screwdriver bit which is then fitted to the motor.

(070347-1)

# Minimalist Motor



Abraham Vreugdenhil

Eenvoudig, apart en toch een doel, dat The design brief for this robot was that it had to be simple, yet unusual, and also have a specific purpose. But should we really call this a robot? It consists of one motor, one wheel, sensors, a microcontroller, LEDs, batteries and 'legs'. If the motor slowly turns one revolution to the left and then one to the right, and the 'legs' offer more resistance in one direction than the other, the robot will move slightly forward. If it also keeps turning towards the light it becomes a true light seeker.

A detailed description of some of the parts follows.

## Wheel

A normal rubber wheel is used for this robot. This gives sufficient grip on the floor, sometimes a bit too much. Particularly in the roll direction, but also perpendicular to the roll direction there is a lot of grip and/or resistance. But any resistance perpendicular to the roll direction will severely restrict the forward movement of the robot. It would have been better to use an omnidirectional wheel instead. These

wheels have smaller wheels/rollers at right angles to the rim, reducing the perpendicular resistance to almost zero. This will work much better than a plain rubber wheel.

## Sensor

For the lights sensors an old favourite is used, the CNY-70. This sensor consists of an IR LED and an IR photo diode and is usually used to detect and track a line on the floor. It is of course also possible to use just the photo diode of this sensor. If we connect two of these in series, with a resistor at each end to limit the current at high brightness levels, we end up with a neat sensor that provides a voltage at the junction of the two sensors that is proportional to the difference in light intensity on the sensors. This junction is connected to one of the inputs of the comparator in the microcontroller. The other input is connected to half the supply voltage, provided by a potential divider formed by two resistors connected to the positive and negative supply. The output signal of the comparator can be read using an internal variable (bit). This way we know at which side of the robot the light is brighter and we can then steer it in that direction.

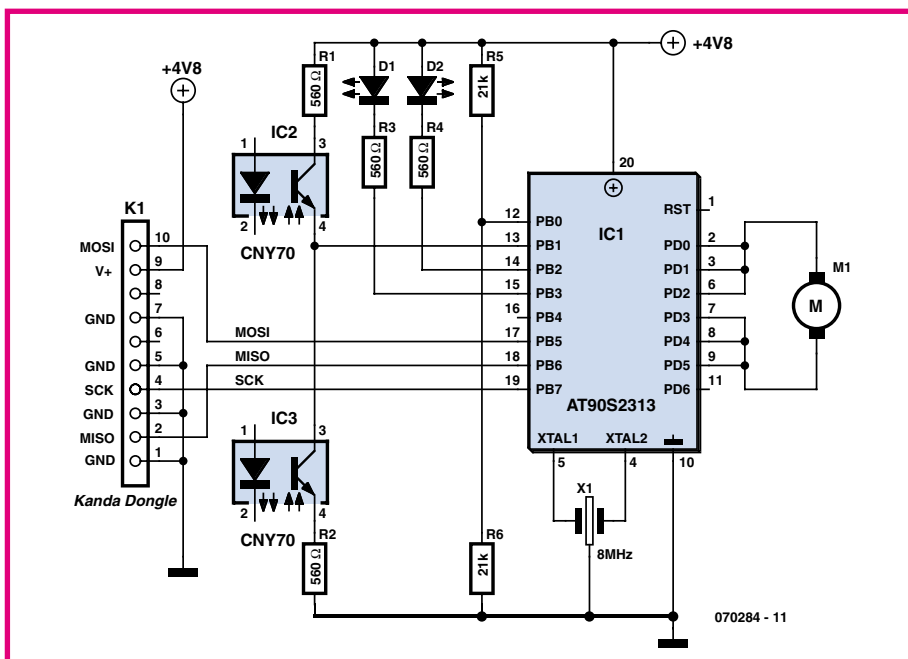
## Microcontroller and motor driver

Most microcontrollers are capable of sourcing a fair amount of current. In this circuit we use this property to directly drive the motor via the microprocessor. When we need to supply larger currents we just connect a few outputs in parallel.

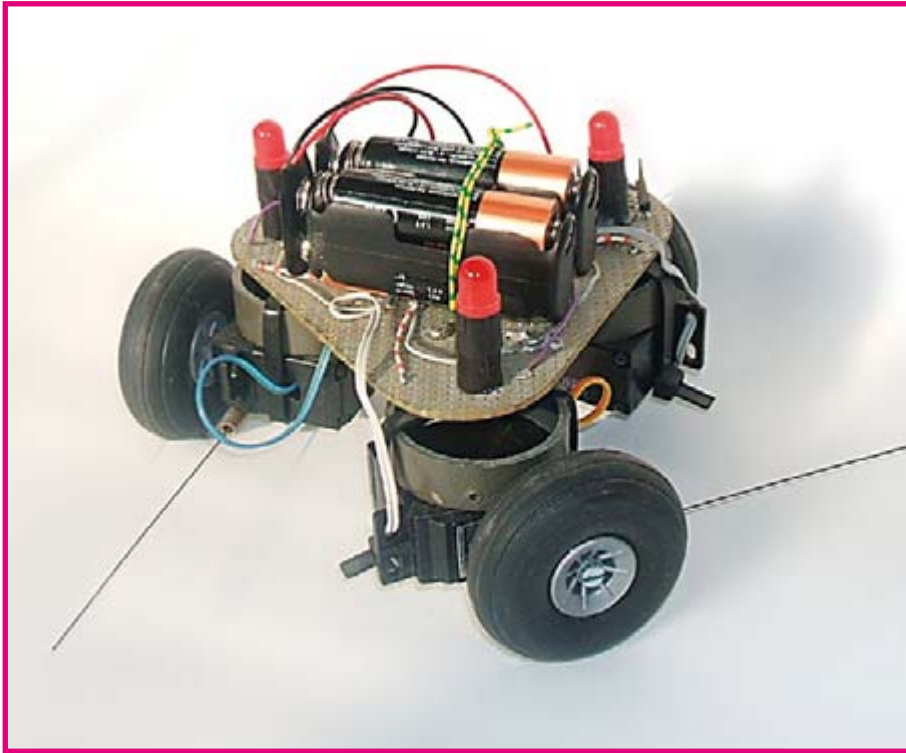
A microcontroller is of course a necessity in every robot. This time our choice was for an AT90S2313. This can be easily programmed in BASIC with the help of BAS-COM-AVR. It has a comparator on-chip, sufficient I/O pins to link together for parallel outputs, etc. An ISP connection is also made available, so it can be easily re-programmed during testing, or at a later stage if you'd like to try out a different program in the robot.

A pair of 3-mm LEDs is mounted above the eyes for show. The LEDs are connected from the positive supply to the processor via 1 k $\Omega$  resistors. To complete the robot circuit there is also a battery holder for four AA cells and an on/off switch.

The AT90S software for the Minimalist Motor can be downloaded from the Elektor website as file number **070284-11.zip**.



# TriBot



Connections to the 89C2051		
Pin	Function	Connects to
1	RST	
2	RxD	
3	TxD	
4	Xtal2	
5	Xtal1	
6	P3.2	(RC5 expansion)
7	P3.3	Motor 2
8	P3.4	Motor 3
9	P3.5	Motor 1
10	GND	
11	P3.7	LED 1
12	P1.0	Antenna 1
13	P1.1	Antenna 2
14	P1.2	Antenna 3
15	P1.3	LDR 1
16	P1.4	LDR 2
17	P1.5	LDR 3
18	P1.6	LED 2
19	P1.7	LED 3
20	VCC	

## Abraham Vreugdenhil

This is a triangular robot with three wheels that can only rotate in one direction. Despite this it can make its way towards a light! That's TriBot.

The robot has three LDR light sensors, which are used to determine which side faces the brightest light. It also has three antennae that are used to detect when it is about to bump into something.

For the processor we used the 89C2051 by Atmel, a nice processor at a reasonable

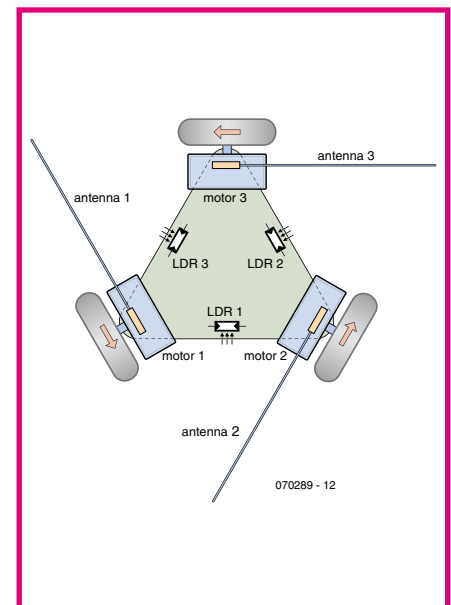
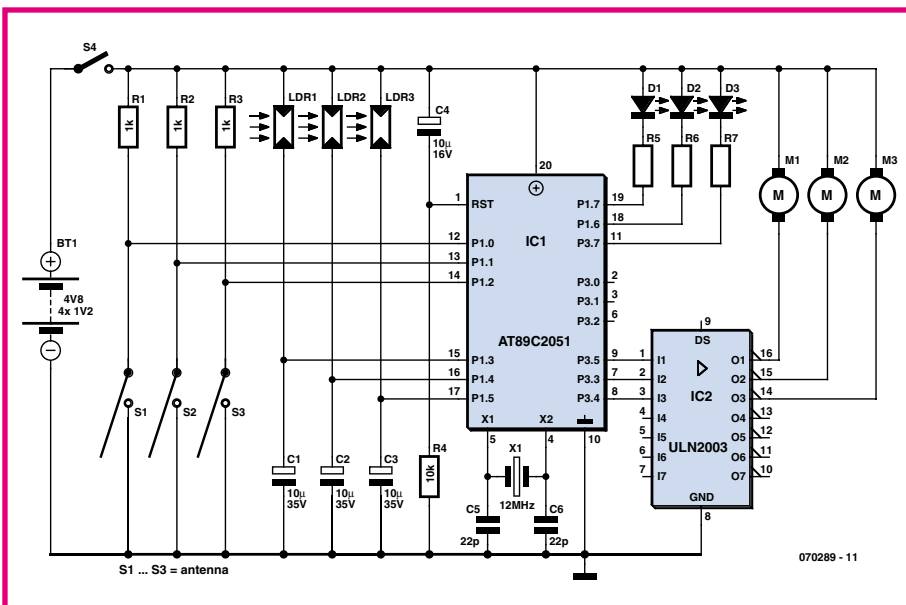
price, which has a 2 k program memory. It is also possible to use a different processor, as long as it has at least nine I/O pins: three for the LDRs, three for the antennae, three for the motors and, if they're available, three for the LEDs.

The measurement of the light intensity via LDRs is implemented using the LDRs in series with a 1 nF capacitor and calculating the RC time-constant. The value of the LDR (which is affected by the light intensity) changes the RC time-constant of this

network (if you refer to the program it will make the working clearer).

The compiler used here (BASCOM-LT or BASCOM-8051) has a function that does all the hard work for us: GETRC(pin number).

The antennae are made of copper pipes with a piece of spring-steel wire through the centre. They are separated from each other with a length of isolated wire. When the spring-steel wire is moved it makes contact with the copper pipe, which is





detected by the processor. The motors are old servos, with all the internal electronics removed. The motors are driven directly via a transistor. They can only be turned on or off and can rotate in only one direction. When all three motors are turned on at the same time the robot will rotate around its axis. If we then turn one of the three motors off, TriBot will rotate about the stationary wheel. Turning on the station-

ary motor and turning off another one will cause the robot to 'stagger' in a fairly straight line towards its destination. To add some visual appeal we've added three 8-mm LEDs. When TriBot is powered up they show a moving pattern and the motors all turn on momentarily (as a self-test). The LEDs are connected in such a way that the LED lights up near the motor that is turned off. Using three motors to propel the robot

in circular motions isn't exactly the most efficient method of travel. (In that case we should have used omnidirectional wheels instead of normal rubber ones.) But that wasn't a requirement for the design of this robot. It just had to look nice and behave in an amusing way. The program for TriBot can be downloaded from the Elektor website as file number **070289-11.zip**.

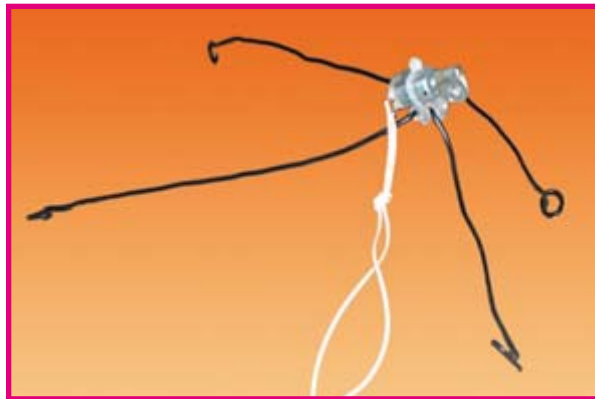
(070289-1)

## Trembly

Abraham Vreugdenhil

The author made this 'construction' for his daughter, who gave it the name 'Trembly' because this robot moves with such a cute trembling motion.

It consists of a little motor fitted with an eccentric weight, a single terminal block, two short legs at the rear and two long legs at the front, all made from electrical wire. The rear legs are fitted with small rubber feet or caps. The motor shaft is



aligned to the longitudinal axis of the long front legs. In other words, the direction of vibration of the eccentric weight is toward the short legs. When the motor is switched on, the eccentric weight causes the robot to start vibrating, with the result that it moves forward. This robot can be built quickly with all sorts of bits and pieces from your junk box, and kids just love it.

(070288-1)

## Hunter

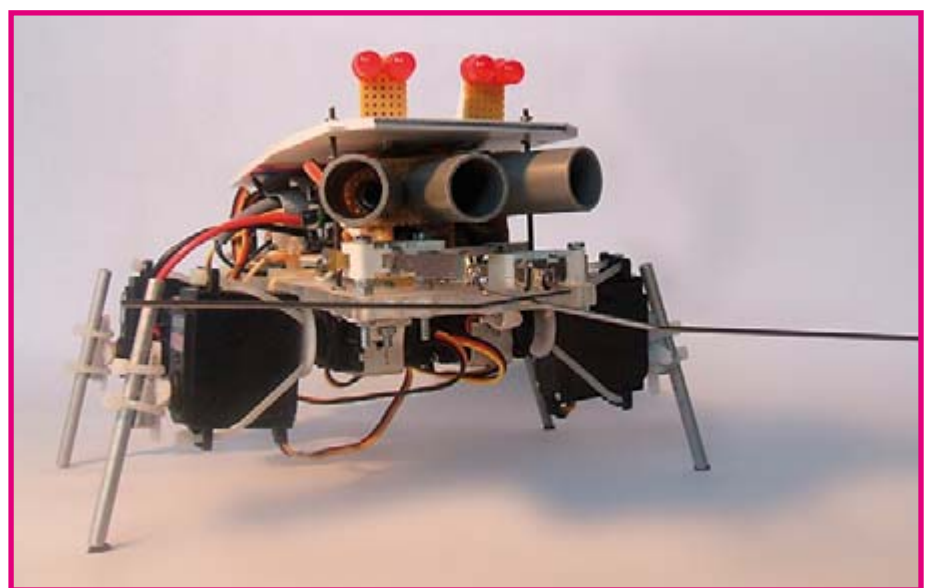
Abraham Vreugdenhil

Four-footed walking robots occupy a special place in the gamut of DIY robots. Walking on four legs has always been a challenge. Building these robots thus creates a strong feeling of satisfaction.

Besides the choice of this form of locomotion, you have to select the sensors you want to fit to your robot so it can explore its surroundings. In this case, we selected feelers for short-distance sensing in order to avoid objects. In addition, the robot will be able to detect moving warm objects, such as people and animals, at a greater distance using a passive infrared (PIR) sensor.

The choice of microcontroller is also important in designing a robot. It must have an adequate number of I/O lines and sufficient memory capacity. In addition, you have to weigh the cost against the desired functions (which means behaviour).

Behaviour is one of the most important parameters. It determines how the robot will respond to the information it receives from the various sensors.



### Servo legs

Developing a nice mechanical design for the legs is a difficult task. There are

many conceivable possibilities using rods and levers, each of which has its specific advantages and disadvantages. In this case, we decided to use a very simple design.

Two servos are strapped together using cable ties. The housing of one servo is secured to a Plexiglas base plate, and an aluminium rod with a diameter of 6 mm is fitted to the shaft of the other servo. The first servo acts as a hip joint, while the other one acts as a knee joint. All four legs are build using this construction.

If you look at the walking motion of a human leg, you see than the knee joint cannot bend any further forward than the fully extended state. The knee can bend backward much further, up to nearly 180 degrees. By contrast, the hip joint can bend forward as well as backward. The servos are fitted such that the legs of the robot have the same freedom of motion.

The walking motion of each leg is divided into 28 steps. The leg moves backward slowly in 22 steps, and then forward quickly in 6 steps. The positions of the knee and hip servos corresponding to these 28 steps are stored in a table. By operating the servos at a fixed interval using the values from this series of numbers, we obtain a nice walking motion of the leg. The servos on the other side of the robot's body must be inverted, since otherwise these legs would walk backward.

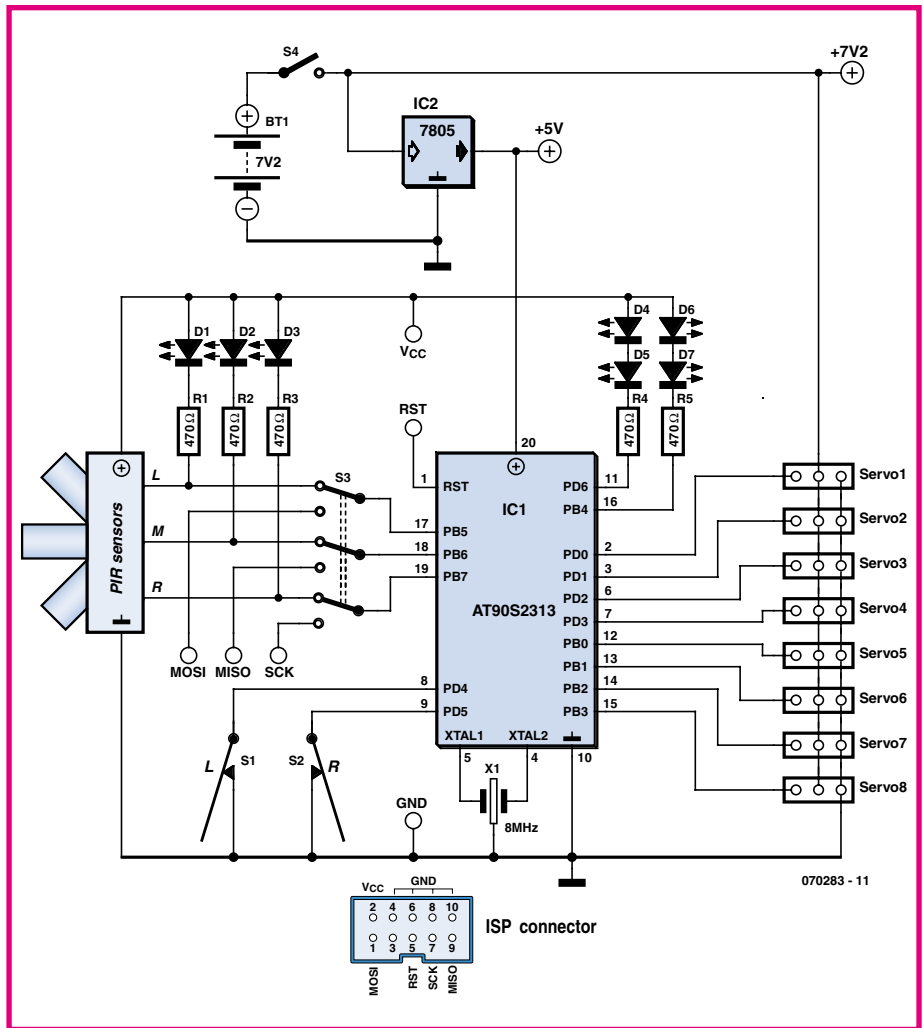
There are four legs, and the number of steps per leg is 28. In order to obtain a stable walking motion, the standard offset between the four legs is set to seven positions in the series of numbers (e.g. left front 1, right rear 8, right front 15, and left rear 22).

To enable the robot to turn, we retard the motion of the two left legs and advance the motion of the two right legs.

When the robot is walking, all 28 steps of the walking motion are always executed before the sensor information is examined again, which means a complete stepping motion of the legs is executed each time. This simplifies the structure of the software and ensures that the legs are always in the same state when a new decision regarding the walking direction must be taken. The feelers have sufficient range of motion that the robot can still manage to move after a full step has been completed.

### Feelers

Two simple microswitches are used for the feelers (i.e., antennae or whiskers). A steel wire (bicycle spoke) is attached to each of the microswitches. A sliding clip is fitted to the spoke and then slid over the microswitch. Two nuts are fitted underneath one of the switches so it is not at the same height as the other one. Otherwise the two feelers would interfere with each other too much. The feelers are connected between two microcontroller input pins and ground. No pull-up resistor is necessary, since the microcontroller has built-in resistors. The feeler switches are closed when they are not activated.



### Eyes

The eyes of this robot do not function as sensors. They are purely decorative. The eyes consist of two 8-mm red LEDs connected in series. They are powered from the 5-V supply via a 150-Ω current-limiting resistor, and they are connected to an output line of the microcontroller. The eyes switch from on to off after each step, but they remain in a fixed state while the robot is moving backwards or turning, depending on the motion that is being performed.

iting resistor, and they are connected to an output line of the microcontroller. The eyes switch from on to off after each step, but they remain in a fixed state while the robot is moving backwards or turning, depending on the motion that is being performed.

AT90S2313 pin assignments		
Pin	Function	Connection
2	PD0	Left front leg hip
3	PD1	Left front leg knee
6	PD2	Left rear leg hip
7	PD3	Left rear leg knee
8	PD4	Left feeler
9	PD5	Right feeler
11	PD6	Right eye
12	PB0	Right front leg hip
13	PB1	Right front leg knee
14	PB2	Right rear leg hip
15	PB3	Right rear leg knee
16	PB4	Left eye
17	PB5 - MOSI	Left PIR sensor / ISP via jumper
18	PB6 - MISO	Middle PIR sensor / ISP via jumper
19	PB7 - SCK	Right PIR sensor / ISP via jumper

## PIR sensor

Various models of PIR are available commercially, such as the Eltec 422 from Acroname. The main disadvantage of this model is its price. There's also the HI-859 from Conrad. Its disadvantage is its inconvenient electrical interface. The signal from this sensor must first be amplified before it can be used.

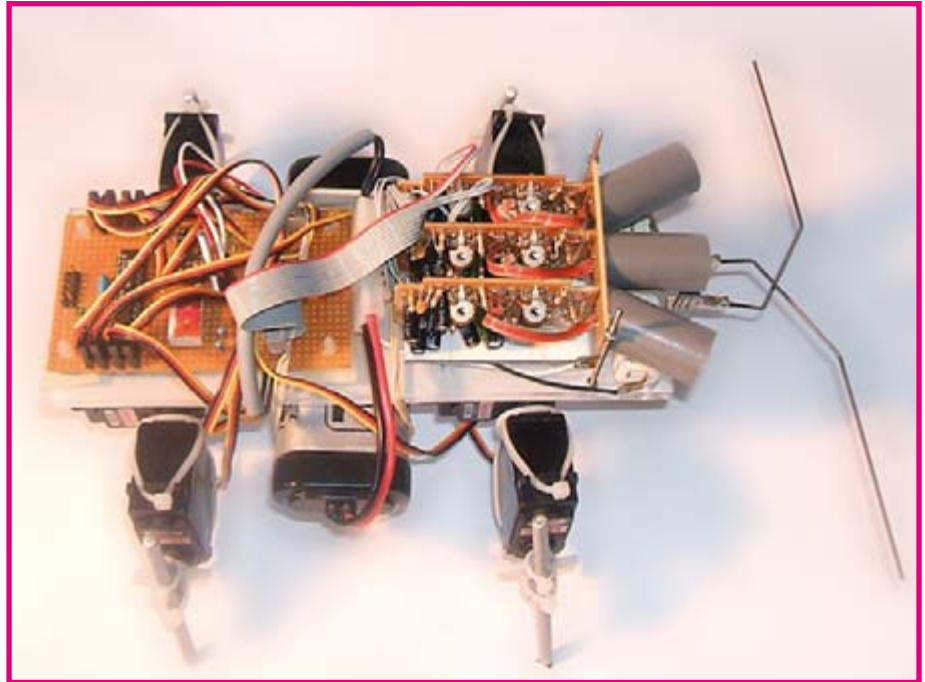
As an alternative, we selected a well-known sensor that is relatively inexpensive: a motion detector available in DIY home improvement shops for less than £ 10. The first thing to do is to remove PCB with the sensor from the motion detector, after which you can fit the sensor to a separate PCB. Three sensors are placed next to each other so the total field of view can be divided into different regions. This yields more information that can be used to influence the robot's behaviour. The circuit boards for the sensor elements are still useful. They hold all the electronics necessary to generate a digital signal at the output. All of this for less than 10 quid per sensor element!

In principle, each of the sensor elements has a detection field of view of 140 degrees. In practice, the angle is larger than this due to reflections and the like. Collimation and screening by means of pieces of electrical conduit with a length of 2 to 3 cm is thus not a bad idea. If the sensors are arranged so their fields of view overlap, we can distinguish five sensor regions.

To make the activity of the PIR sensors visible, a LED is connected to the output of each PIR sensor. These LEDs are connected to the 5-V supply via resistors. The LED is on if the PIR sensor does not detect anything, and it switches off if the associated PIR sensor detects a moving warm object.

## Microcontroller and compiler

The selected microcontroller is an Atmel AT90S2313 (see the schematic diagram in **Figure 1**). It has 2 kB of program memory, which provides plenty of room to program intelligent behaviour. The AT90S2313 has 15 I/O pins, of which three can be used for ISP. An ISP port is quite handy during testing and in case of future software extensions. This port is fitted to the robot using a 10-way CANDA connector. You can program the device using the Sample Electronics Programmer [1], among other options. Only three resistors are necessary to connect a printer port to this port. Quite a few I/O lines are necessary for controlling the hardware, which comprises eight servos, two feelers, two LEDs, and three PIR sensors. This makes a total



of 15 devices. For this reason, the three PIR sensors are connected to the robot in parallel with the ISP port via jumpers. In normal operation, the PIR sensors are connected to the robot. If a new program must be downloaded to the robot, the jumpers must be switched over to make the ISP port available.

The robot software was programmed in Basic and compiled using BASCOM-AVR [2]. This an excellent compiler for Atmel AVR microcontrollers. The freeware version of BASCOM-AVR can easily handle the 2-kB program capacity of the AT90S2313.

The software for the Hunter can be downloaded free of charge from the *Elektor Electronics* website as item no. **070283-11.zip**.

## Power supply

The robot is powered by a 7.2-V rechargeable battery of the type used in models. An on/off switch is fitted to the battery. A regulated 5-V supply voltage for the electronics is generated by an old friend in this area: a 7805 in a TO220 package. The servos are powered directly from the battery via the switch.

## Behaviour

The name of the robot says a lot about its behaviour. It hunts for something. Its job is to detect and follow warm moving targets, such as people and animals. As long as the robot can see something, it will pursue it. If it can't see anything, it advances by five steps in the hope of seeing something. After these five steps it remains standing

and waits for new prey. The behavioural rules can be summarised as follows:

1. If nothing is detected, do nothing. (sight = 1)
2. If sight > 1, take one step (sight = sight - 1).
3. Check for obstacles after each step. If an obstacle is detected, walk backward and then turn away from the obstacle. (sight = 6)
4. If sight < 5, check the PIR sensors. If something is detected, turn in the direction of the detected object. The options for the turning direction are: left, forward left, straight ahead (no turn, with sight = 6), forward right, and right. (sight = 6)

## Conclusion

The objective was to build a four-legged robot with interesting behaviour. This objective was ultimately achieved, and with a reasonably limited budget. The microcontroller memory is pretty well filled by the current software, but it would still be possible to devise a more efficient behaviour algorithm. This means that there are plenty of options for experimenting with this robot. Thanks to the onboard ISP port, programming the robot is easy. This robot is a means, not an end. Let's hunt!

(070283-1)

## Web Links

- (1) [http://avrhelp.mcselec.com/Sample\\_Electronics\\_cable\\_programmer.html](http://avrhelp.mcselec.com/Sample_Electronics_cable_programmer.html)
- (2) [www.mcselec.com](http://www.mcselec.com)

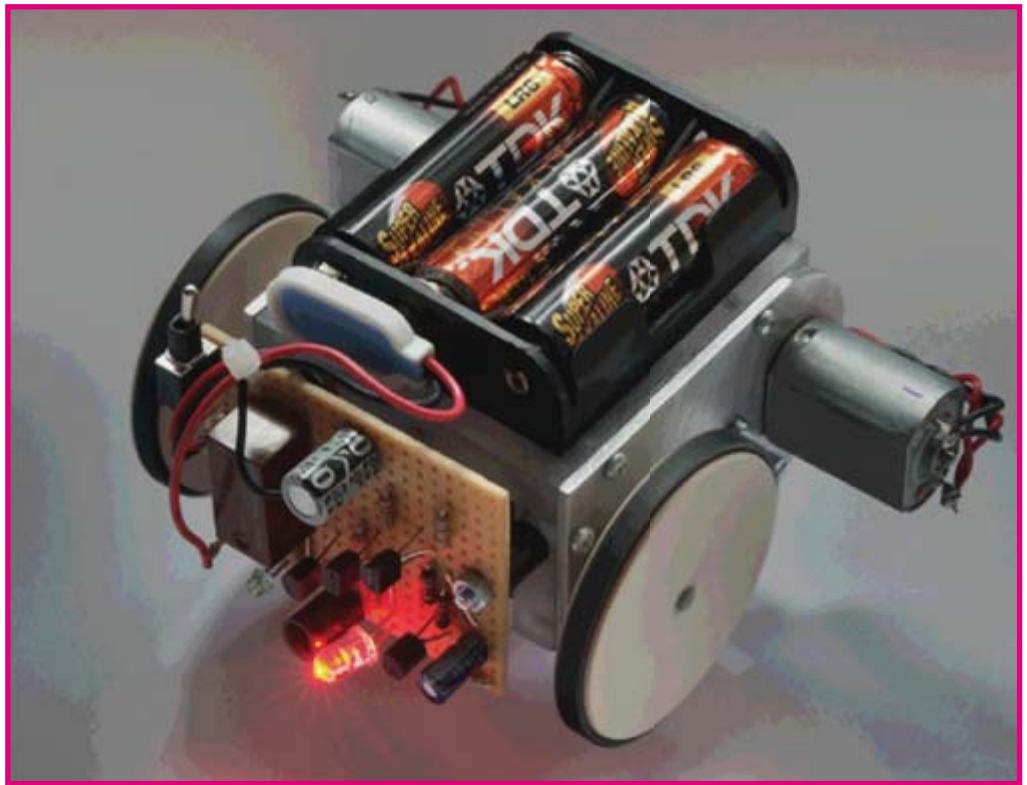


# Robot MOPS

Markus Bindhammer

MOPS is a small robot which generally bumbles around the floor and performs avoidance manoeuvres whenever it detects an obstacle in its path. MOPS uses a forward facing LED to illuminate its path and a phototransistor to detect light reflected from obstacles, as soon as an obstacle is detected MOPS goes into reverse and turns for a few seconds on its two wheels before setting off again in another direction.

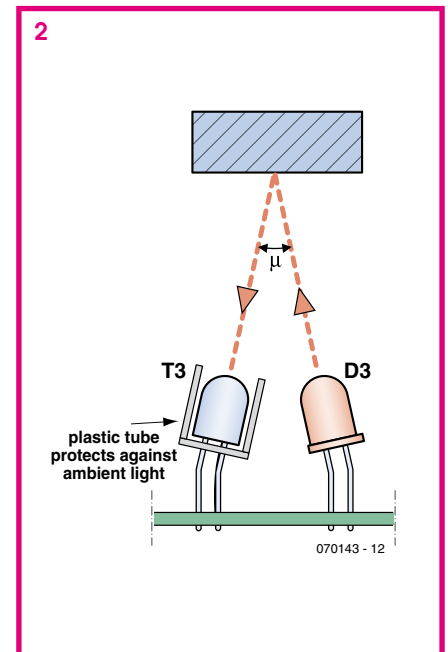
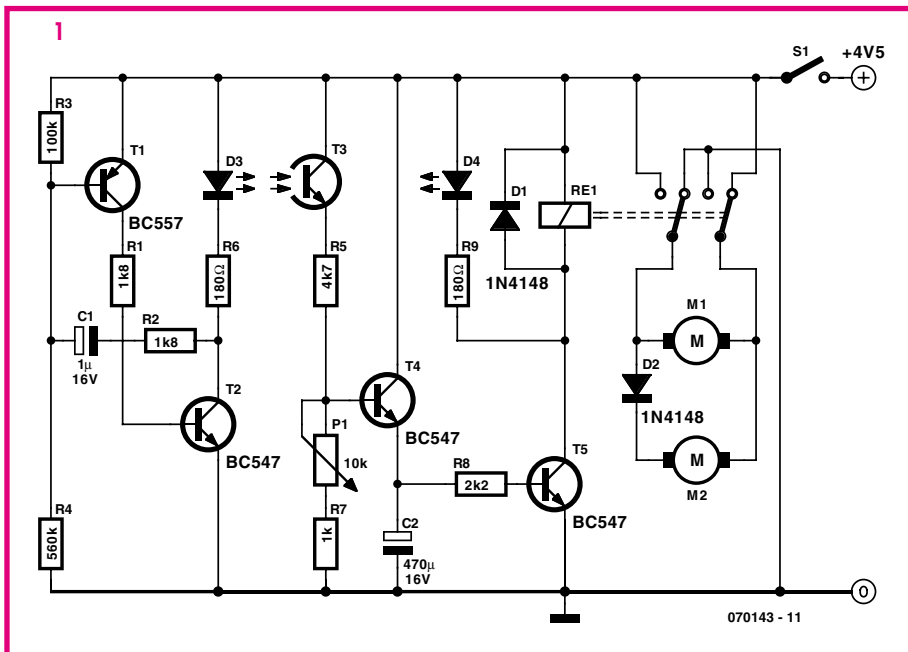
A look at the circuit diagram in **Figure 1** indicates that MOPS is built (rare for this day and age) entirely from discrete components. Resistors R1 to R4 together with C1, T1 and T2 form a multivibrator circuit which continually switches the LED on and off. On power-up current flows through R4 and the base-emitter junction of transistor T1. T1 is therefore conducting and current flows through R1 and the base of T2 which is also conducting. Current through the collector of T2 and R6 lights up the LED. During this time capacitor C1 is charging up and when the voltage level gets sufficiently high the base of T1 becomes reverse biased and T1 turns off. T2 will also turn off along with the LED. C1 begins to discharge until the base of T1 is low enough to begin conducting again. T2 switches on which reinforces the low



base voltage on T1 via C1 and the cycle continues.

The LED light source for obstacle illumination does not need to flash, it can be lit continuously but there are two reasons for the flashing LED: firstly it conserves battery power, giving MOPS a longer range and secondly (and more importantly) a flashing light looks much more impressive than a

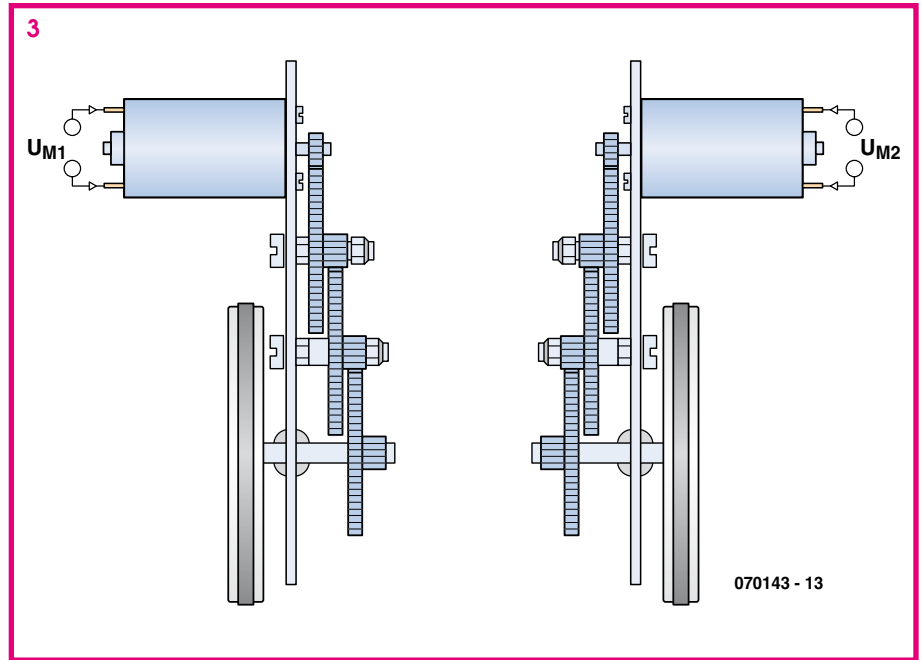
boring old continuous light source. When reflected light falls on the phototransistor T3 a current flows through R5 to ground which produces a voltage at the base of T4 to make it conduct. The values of R5, P1 and R7 affect the switching threshold so adjustment of P1 will help to reduce spurious detections caused by external light sources. Turning P1 (a preset could be used here instead) so that it



increases its resistance will make T4 more sensitive and vice versa. T4 conducts in synchronism with the flashing LED so capacitor C2 acts as a reservoir capacitor to ensure that once T4 begins to conduct, sufficient energy is stored in C2 to ensure that T5 remains conducting continuously until the obstacle is out of range and T4 switches off. It takes a few seconds for C2 to discharge and during this time MOPs is performing its avoidance manoeuvre.

T5 switches a double pole relay which has the effect of performing this manoeuvre. In normal forward motion the relay switches the positive and negative supplies to both motors and diode D2 is conducting. When an obstacle is detected the relay switches and reverses the polarity of the motor connections. D2 now becomes reverse biased and no current can flow through motor M2 while M1 goes into reverse. This gives MOPs the reverse and turn response to obstacles. A few seconds after the obstacle is no longer detected the relay switches back and MOPs carries on as before but in a different direction.

**Figure 2** shows a close up of MOPs's eye mounted in a tube which helps to make obstacle detection more directional and reduces the effects of external light



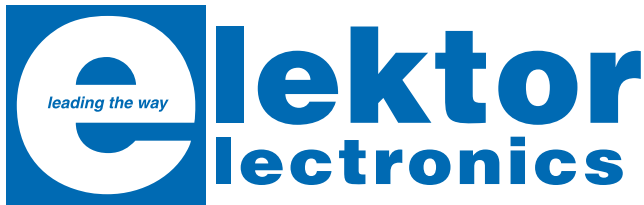
sources. With this set-up it was possible to detect obstacles at a distance of 10 cm. The range depends largely on the reflective properties of the obstacle so darker objects will only be seen at shorter ranges. The circuit can be modified to read microswitches connected to contact feelers mounted on the front of MOPs this will help avoid col-

lisions with matt black objects. **Figure 3** shows the layout of the two motor and gearbox assemblies.

(070143-1)

**Web link**

[www.elexs.de/robo1.htm](http://www.elexs.de/robo1.htm)



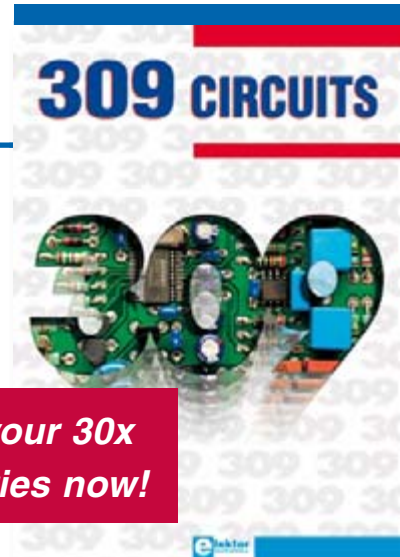
**NEW**

## 309 Circuits

The present tenth edition of the popular '30x Circuits' series of books once again contains a comprehensive variety of circuits, sub-circuits, tips and tricks and design ideas for electronics. These 309 Circuits offer a representative indication of present-day electronics.

Regular '30x series' enthusiasts will no doubt know what to expect: 309 Circuits contains many fully elaborated electronics projects. In addition, there are numerous ideas, each of which with a potential for use in your own research, projects and applications. Among many other inspiring topics, the following categories are well presented in this book: test & measurement, RF (radio), computers and peripherals, audio & video, hobby and modelling, microcontrollers, home & garden, power supplies & battery chargers, etcetera.

ISBN 978-0-905705-69-9  
Approx. 432 pages  
£19.95 / US\$ 39.95



**Complete your 30x circuits series now!**

Order now using the Order Form in the Readers Services section in this issue.  
Elektor Electronics (Publishing)  
Regus Brentford  
1000 Great West Road  
Brentford TW8 9HH  
United Kingdom  
Tel. +44 208 261 4509

See also [www.elektor-electronics.co.uk](http://www.elektor-electronics.co.uk)



**PCB Certified By UL, ISO  
And Over 1,000 Customers  
In 39 Countries**

Up To 50 Layers  
2.5mil Track/Space, 0.1mm Holes  
Blind & Buried Vias  
Lead-free, RoHS-compliant  
Flex, Mixed, Teflon, Rogers  
Quickturn Available

**\$50** for 5pc 4"x4" 2L  
**\$188** for 3pcs 4"x4" 4L  
**\$288** for 3pcs 4"x4" 6L  
**Free** Soldermask & Silkscreen

**www.ezpcb.com**  
**sales@EzPCB.com**

**Schaeffer**  
**AG**

## FRONT PANELS & HOUSINGS

Cost-effective single units and small production runs

Customized front panels can be designed effortlessly with the Front Panel Designer. The Front Panel Designer is available free on the Internet or on CD.

- automatic price calculation
- delivery in 5-8 days
- 24-Hour-Service if required



Sample price: 30,42 €  
plus VAT/shipping

Schaeffer AG · Hohentwielsteig 6a · D-14163 Berlin · Tel +49 (0)30 8 05 86 95-0  
Fax +49 (0)30 8 05 86 95-33 · Web info@schaeffer-ag.de · www.schaeffer-ag.de

## No Compromise Oscilloscope



Other oscilloscopes in this price range force you to compromise on one of the key specifications: sampling rate, memory depth or bandwidth.

The PicoScope 5000 series is a no compromise PC oscilloscope at a price every engineer can afford.

[www.picotech.com/scope422](http://www.picotech.com/scope422)

01480 396395

**pico**  
Technology



1GS/s sampling rate

250MHz bandwidth

128M sample buffer memory

125MS/s 12 bit AWG built in

PicoScope 5203  
32M buffer £1195

PicoScope 5204  
128M buffer £1795



# 2.4 GHz Antenna for Robotics Vehicles

Ragnar Jensen

Radio equipment using the license-free 2.4-GHz ISM (industrial/scientific/medical) band is also rife among robotics fans, just think of what you can do with Bluetooth, wireless cameras, remote control, or even a WLAN client or access point fitted on a robot vehicle! Point is, such homebrew applications typically require an antenna that's (1) omnidirectional, (2) 'sort of' flexible in view of the damage and bashing it may have to take and (3) low<sup>2</sup> cost.

The antenna shown here fulfills all three requirements brilliantly. See for yourself, the pictures say more than 1 kword.

You will need:

- a short piece of 50-Ω coax cable like RG58(C)U with a crimped-on BNC plug ('borrow' a cable from the IT dept.);
- a sharp (hobby) knife;
- a pointed tool like a strong needle or a watchmaker's screwdriver;
- a ruler;
- a soldering iron (optional);
- a hot glue gun;
- common sense and about 30 minutes of your time.

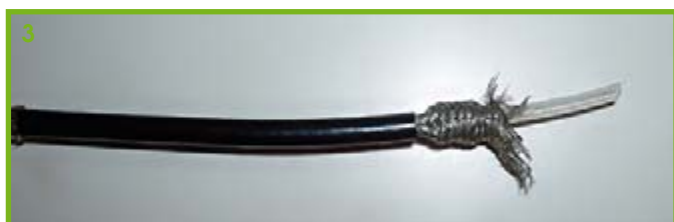
Here goes.



The raw material: a short piece of 50-Ω coax cable.



Cut off about 40 mm of the outer insulation. This will expose the braid that forms the cable shield.



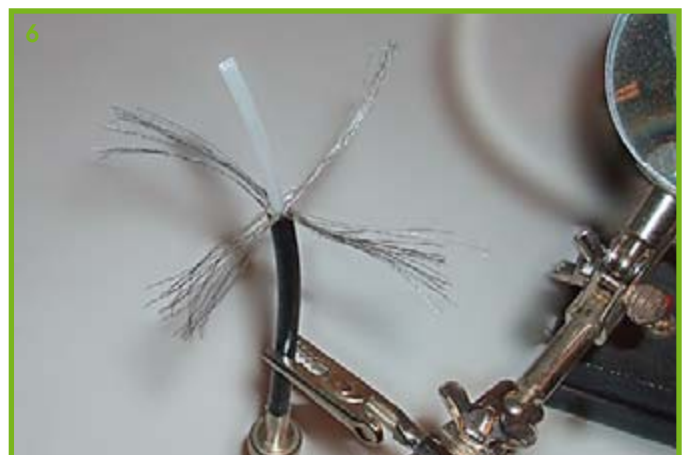
Push down the braid to expose the inner insulation.



Using the pointed tool, carefully unweave the braid strands.



Until your hairdressing looks something like this...



Distribute the strands into four equally sized and likewise spaced bundles (wow, tresses!)



Now bend them into a 45-degree angle with respect to the cable. This angle will result in an antenna impedance of about  $50 \Omega$ .



Cover the bundles with solder. Although this step is optional, it does enable the ground plane elements to hold their shape better, and prevents untwisting of the strands.



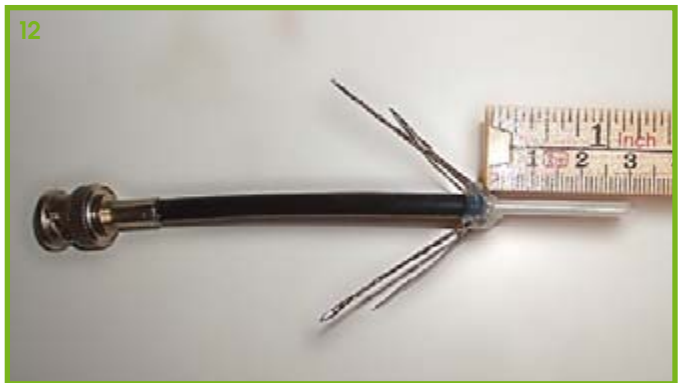
Done soldering!



Apply a few drops of hot glue...



... will help keep the proper shape.



Cut the radials and the radiator to a length of 30 mm (yes that's  $0.25 \lambda$ ).

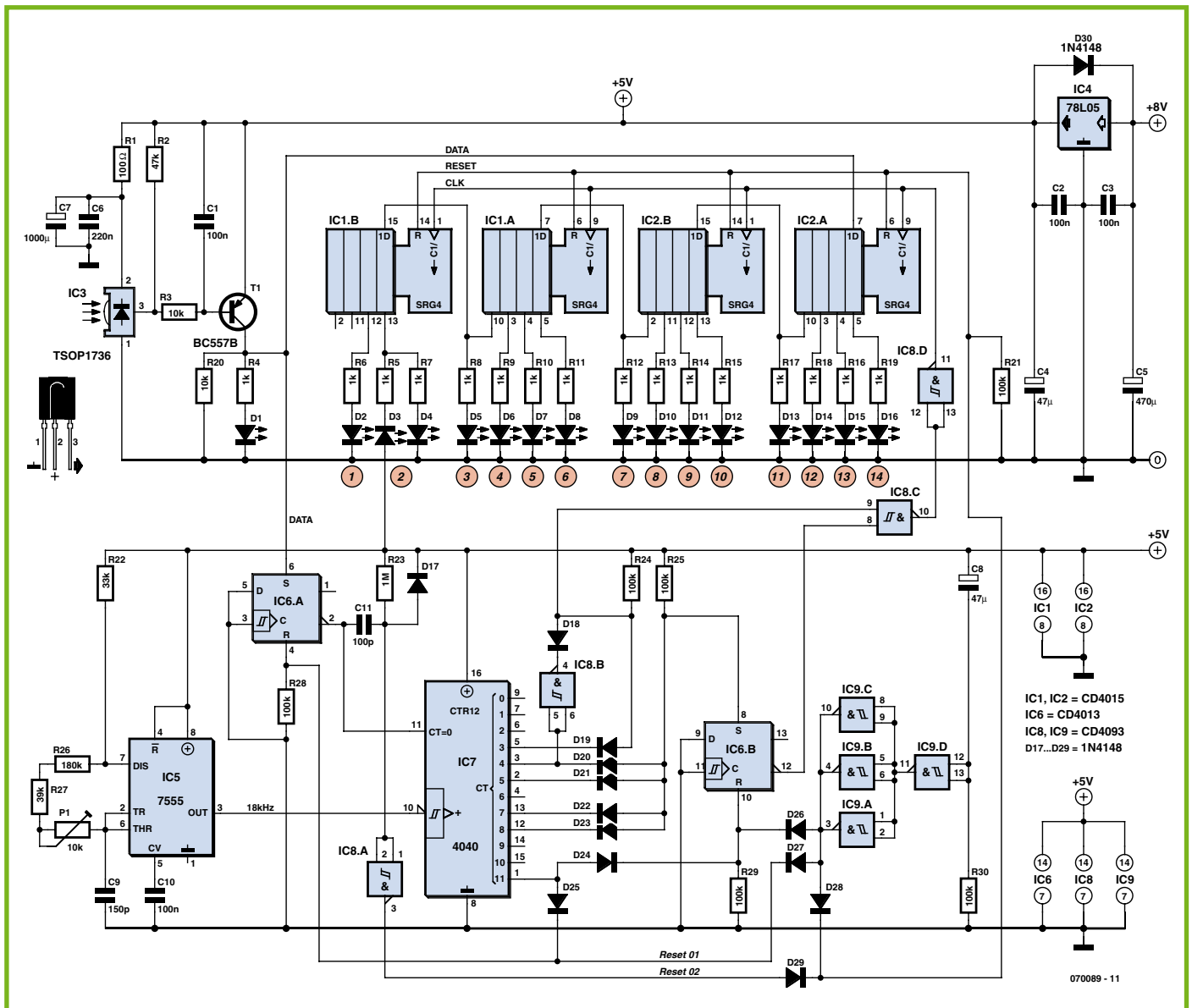


And we are done!

(070142-1)

(reproduced with modifications and extensions from CQ-TV issue 217)

# Receiver for RC5 Remote Controls



Thomas Möll

Software decoding of remote control signals using the RC5 protocol does not present a significant challenge to a modern microcontroller, while for a pure hardware solution specialised RC5 decoder ICs are available. Nevertheless it is interesting to look at how we might process RC5 signals using ordinary components. This way not only do we learn about how the code works, but also the resulting circuit is easier to adapt to different applications. Although the circuit was originally designed just to display the address and command emitted by a 'universal remote control' when each button was pressed, it could be used, for example, to add a remote control facility to an audio amplifier using a standard remote control unit. Indeed, virtually

any household appliance could be controlled in this way: just choose an address that is not yet allocated and define your own commands.

The TSOP1736 infrared receiver inverts the bits in the received stream. T1 inverts them again so that they are now available with the correct polarity. The LED connected to its collector indicates when data bits are received.

This signal is low (0 V) for the first half of the start bit and high (5 V) for the second half. This pattern represents a '1' bit, and FF01 (CD4013) will be set. The complementary output of this flip-flop will therefore be low, enabling the CD4040 divider. The 18 kHz square wave clock for this divider is generated by our old friend the NE555. At the same time, the differentiator formed by C11 and R23 generates a low-

going pulse which is inverted by Schmitt trigger inverter ST8. The resulting high-going pulse is used to clear the CD4015 shift register.

The Q4 output of the CD4040 (pin 5) carries a square wave at 1125 kHz, corresponding to a period of 888.8 μs and a pulse width of 444.4 μs. Output Q5 (pin 3) of the CD4040 is inverted by Schmitt trigger ST4; the output of this gate is therefore initially high. The diode matrix forms an AND gate which sets one input to ST1 high just as Q4 rises for the first time. Since the complementary output of FF02 is also high, a high-going pulse (inverted by ST2) is delivered to the CD4015 shift register, causing it to take one sample of the incoming data stream. This process is repeated 1.333 μs after the start of each bit period, or exactly three-quarters of the way into



each bit. This is the key to the circuit: the value of the signal sampled at this point gives the encoded data bit. After 14 bits the RC5 packet is complete. At this point a diode matrix forming an AND gate at the outputs of the CD4040 sets FF02. Its output goes low and the clock to the shift register is blocked.

One cycle of the transmit protocol takes 64 bit times. Q11 of the CD4040 counter goes high 32 bit times after the start of the RC5 packet, resetting FF01 and thereby stopping the counter. The data bits at the outputs will be held until a new packet from the transmitter sets FF01 again, whereupon the output is cleared and the bits read in afresh.

The stop pulse on pin 11 of the CD4040 can be used to validate the output data. When building the circuit it is a good idea to fit a test point at the output of the NE555 so that the 18 kHz clock can be set accurately. All of the timing in the circuit depends on this signal.

(070089-1)

# IR Remote Control with the R8C

G. van Zeijts

Over the years various articles have appeared in *Elektor Electronics* about microcontrollers that pick up the pulses from an IR remote control and do something with it.

Unfortunately this capability was not available for the R8C microcontroller until now. That is why the author plunged right in and created this capability, mainly for his own use but also made it available to others who may be interested.

The functionality has been designed in such a way that it can optionally be used with either a Philips (RC5) or Sony remote control.

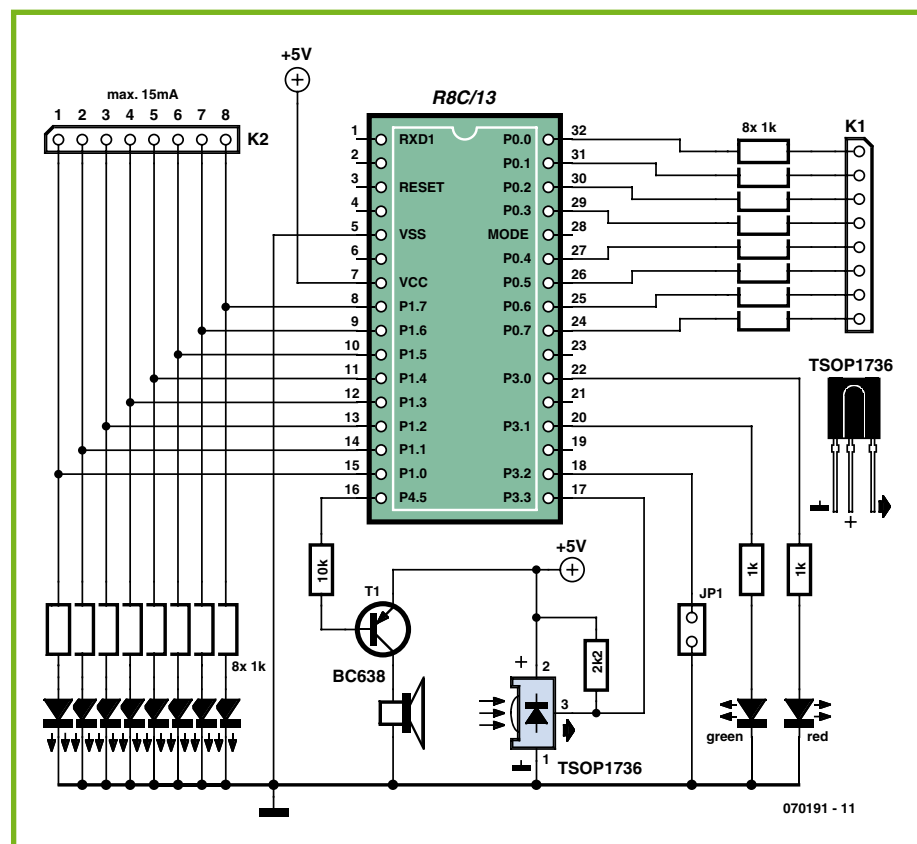
The system provides a 7-bit code at an output port, to 'inform' a computer or other microcontroller which button has been pressed.

The 8 bits of the other output port can be controlled directly with the buttons 2 through 9 of the remote control. This allows 8 digital devices to be directly switched or controlled remotely.

The program has been written in C using the HEW software and has the following functions.

Depending on the position of jumper JP1, pulses from Philips (RC5) are decoded (open) or pulses from Sony (jumper in place).

1. Bit 7 of port P0 indicates whether RC5 or Sony pulses are being used. Bit 7 'High' = RC5 and bit 7 'Low' = Sony.
2. The code for the most recent button that was pressed is on port P0. Bits 0 through to 6 are used for this. Bit 7 is used to indicate RC5 or Sony.
3. The eight bits on port P1 (output) are directly driven high or low with buttons 2 through 9 of the remote control. When the button is pressed for the first time the output goes high. The next press makes the output low. These eight buttons can therefore control eight digital things from a distance. The state of all the bits on port P1 are not affected by pressing any of the



other buttons on the remote control, with exception of the following three.

Button '1' makes all eight bits of P1 high. Button '0' and button 'off' make all eight bits of P1 low.

Eight LEDs indicate the present state of the eight bits — they load port P1 with about 3 mA. Via connector K2, P1 can therefore be loaded for 'heavier' purposes with a further 17 mA at the very most (but allow a margin just to be safe, so 15 mA max, for example).

4. If RC5 is selected with JP1 (open) and the microcontroller receives pulses from a 'non-RC5' remote control (or the other way around) a short alarm signal results:

- The red LED on P3.0 flashes briefly;
- The outputs 0 to 6 of port P0 become '0' (= no standard code);

- Bit 7 of port P0 gives a flashing signal;
- Brief acoustic signal on pin 16 (bit P4.5).

A short description of the schematic: The well-known TSOP1736 (infra-red detector) is directly connected to the input and its output is pulled high with a 2k2 resistor.

A second input is used to read the state of jumper JP1 (select between RC5/Sony).

Output P0 is used to signal the RC5/Sony-code in hexadecimal form. These outputs can be directly connected to another computer or microcontroller through the 1 k resistors, with the intention that this computer/microcontroller can act on the received code.

Output P1 can be used by the user to switch a 'digital something' with an inter-

face circuit that you have to build yourself. (K2 can be loaded with a maximum of 15 mA.)

Low-power LEDs indicate the present state of the eight bits of port P1.

The green LED on pin 20 functions as on/off indicator for the circuit.

It is of course also possible to use a piezo buzzer for the acoustic alarm on P4.5 (P4.5 may be loaded by up to 8 mA).

The software for this project is a free download from the Elektor website — see archive file **070191-11.zip**.

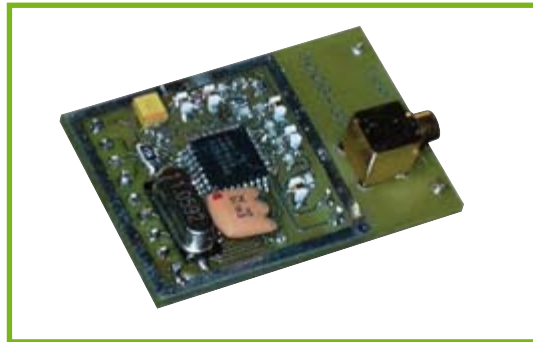
(070191-1)

# zBot: Wireless Link

Jens Altenburg

There exist a lot of wireless communication modules, all approved for use within the ISM radio bands (industrial/scientific/medical), for example, 433 MHz, 866 MHz and recently also 2.4 GHz. You get simple and cheap ones with low transmission data rates, and you can find excellent high-speed systems. "How much (will the thing set me back)?" is the most frequently asked question if you search for an RF module. Low-speed non-intelligent modules are cheap; high-speed intelligent ones, pretty expensive. That's easy but it doesn't help.

The CT-Video GmbH ([www.ct-video.com](http://www.ct-video.com)) markets a special module with high-speed digital data transmission capability and no intelligence, at a reasonable price. The



module is based on a fully integrated transceiver with a digital interface. It is used in zBot<sup>1</sup> with good results.

The module comes as a small fully assembled and tested board. The board includes the complete RF sections. It works in the 433 MHz ISM band and has a transmit power of 10 mW coupled with a receiver

sensitivity of about -108 dBm.

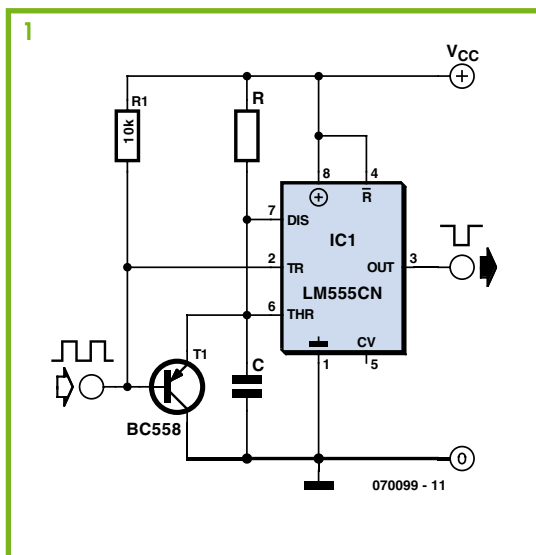
For implementing the module in your own projects, a software module is available, too. The module only needs a few resources of the micro, some GPIOs (general purpose input output pin) and one UART channel. The UART has to be programmed to give a baud rate of 76.8 kbits/s. The baud rate needs to be accurate, if not, you'll suffer an effect that manifests itself as low receiver sensitivity.

The software module for the wireless radio is a file called `rf433.c`.

(070173-1)

(1) The complete document called Zbot — the Robot Experimental Platform is available for free downloading from the Elektor Electronics website. The file number is 070172-11.zip (July/August 2007).

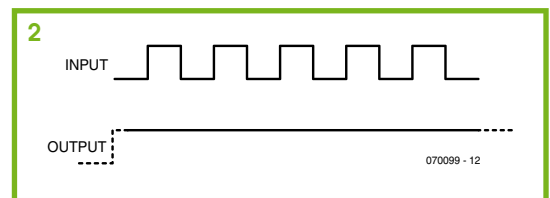
# Removed Pulse Detector



Hesam Moshiri

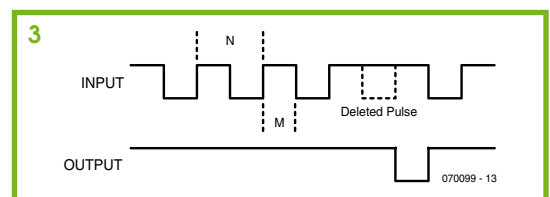
A detector to flag missing pulses is among the more important circuits in robotics. When pulses are applied to the circuit shown in **Figure 1**, the output signal will be continuously High (i.e., nearly  $V_{cc}$ ) as illustrated in **Figure 2**.

For the detector to operate, certain conditions in respect of pulse timing must be met. With reference to the timing diagram shown in **Figure 3**, the values of components  $R$  and  $C$  in the circuit may be calculated from



$$\tau = 1.1 RC$$

taking into account that



$$1 \text{ k}\Omega < R < 1 \text{ M}\Omega \text{ and } M < \tau < N$$

When a correct signal is being applied to the input (Figure 2) the circuit will be triggered by another pulse before the constant time ( $\tau$ ) expires. Therefore the output sig-

nal remains High. If one or more pulses are missing, for example, owing to a fault, a bash on the head from rebot Mathilda, or simply bad reception on the remote control channel, the output signal drops Low briefly. The resultant flag signal can

be sensed by another circuit, for example, a microcontroller or another sensing that acts on the interruption in the pulse stream. If the worst comes to the worst, the autopilot should be switched on!

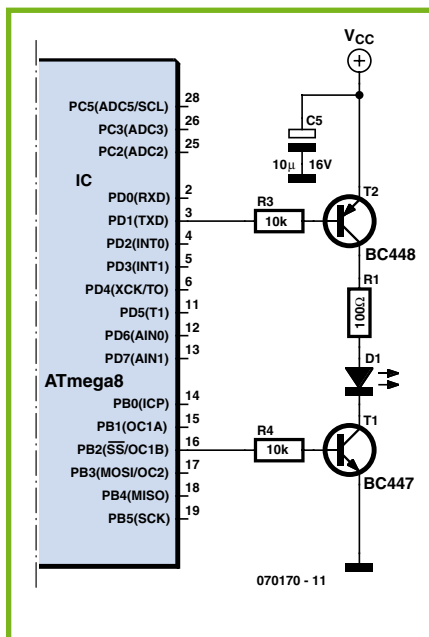
(070099-1)

# IR Communications using a UART

Dominik Tewiele

If you wish to set up a medium-range (5 m to 10 m) communication link between two robots or between a robot and a base station, infrared light can be an economical alternative to using radio modules. Tried-and-tested standard protocols and supporting components are available for the modulation necessary to suppress the effect of ambient infrared light. Practically every modern microcontroller sports one or more asynchronous serial interfaces (UARTs), which make perfect partners for IR receivers and transmitters.

An example of a suitable receiver is the TSOP17xx, where the 'xx' stands for the modulation frequency, measured in kilohertz. Its output can be connected directly to the RxD pin of a microcontroller. Because of the open-collector output stage, it is possible to connect several receiver modules in parallel to increase the area covered. The transmitter consists simply of an IR diode and a couple of discrete components. A timer in the microcontroller can



be used to provide modulation, or alternatively an external NE555 can be used. In this example we are using an ATmega8 with Timer1 configured so that the output

compare registers OCR1A and OCR1B control the frequency and pulse width of the signal on output PB2. The NPN transistor then applies the required modulation. Here again, we can wire several IR LEDs in parallel to increase the transmit range and coverage angle. The series current limiting resistor for the IR LEDs should be chosen with consideration for the desired range and the maximum pulse current that the LEDs can handle. This last figure can be found on the LED's datasheet, which will also help determine a suitable pulse width to set in the software.

The maximum baud rate that can be achieved will depend on the receiver chosen. Using a TSOP17xx around 1200 baud is possible, which should be adequate for simple control commands. For bidirectional communication it will be necessary to build both a receiver circuit and a transmitter circuit at each end of the link. It is worth bearing in mind that because of the effect of reflections scope for full-duplex operation is rather limited.

(070170-1)

# Radio Remote Control for PDAs and Smartphones

Peter Zirngibl

On his website, under the title 'AVR Blue Remote', the author describes (in German) a Bluetooth-based remote control unit featuring six output relays and six sensor inputs. The site also includes Smartphone software (avrblueremote.exe) and microcontroller software (avrblueremote.hex). These can be used as the basis for projects

such as a short-range (up to 10 m or so) remote garage door opener or a remote lighting controller. The software is free for use by private individuals.

Any Smartphone running the Windows Mobile 5.0 operating system can be used as the transmitter. The receiver used is the postage-stamp-sized Blue Nice Com III Bluetooth module with integrated chip antenna from Amber Wireless. The module is based

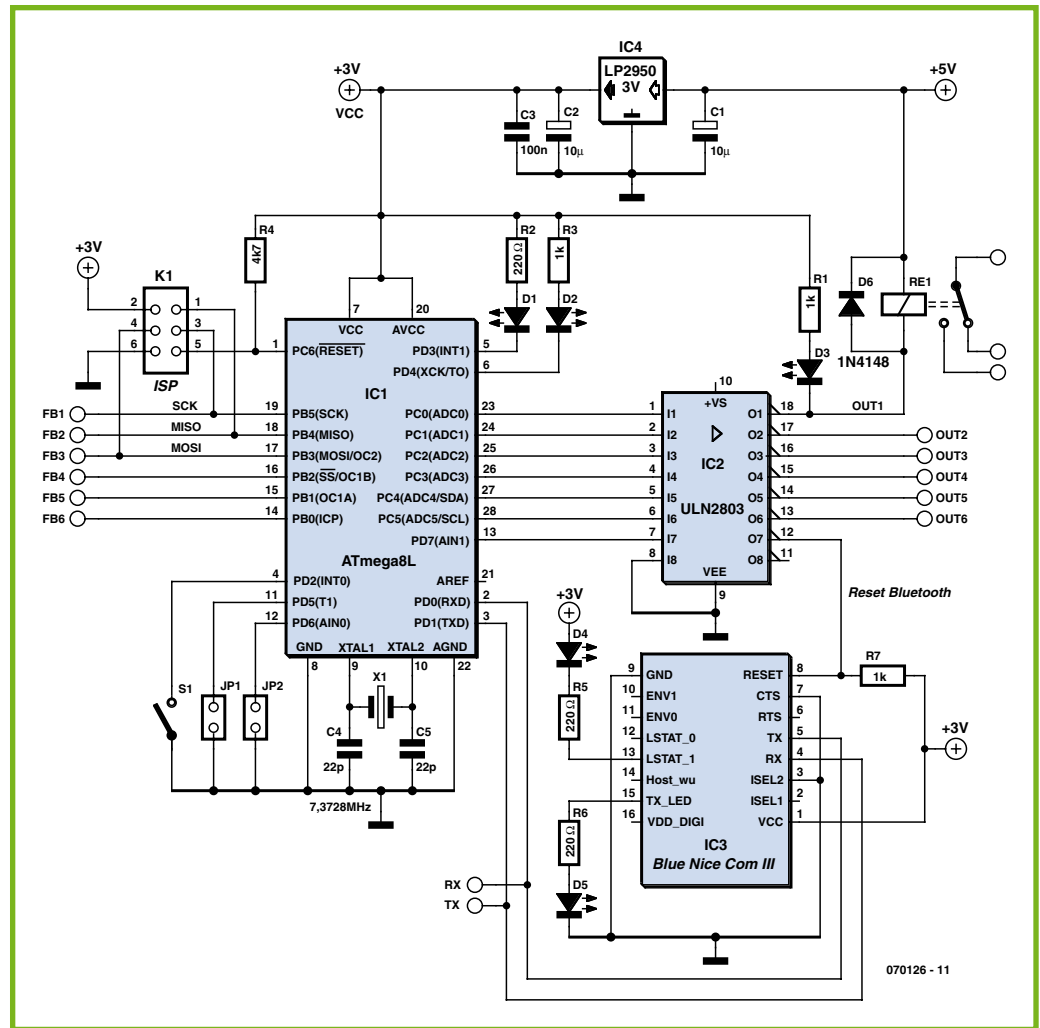
around the LMX9820A from National Semiconductor, and decoded messages are passed over a UART-like interface (TX and RX signals) between it and an Atmel AVR ATmega8L microcontroller. Connected to the outputs of the microcontroller is a ULN2803 octal driver which can comfortably switch enough current to drive power relays. Completing the circuit are a 3 V voltage regulator (type LP2950-3V) and an ISP



(in-system programming) connector. Four LEDs indicate the status of the connection: LED1 shows when the microcontroller has received data correctly and LED2 indicates when a timeout has occurred. The Bluetooth module's LEDs indicate the link status (LED3) and transmit mode (LED4).

The printed circuit board layout for the circuit has to meet several constraints. For maximum range there should be no ground plane, conductors, components or other metal parts within 8 mm of the antenna; other requirements are set out in the manual for the module. The microcontroller can be programmed using the ISP interface: you must of course make sure that the pinout of the connector is compatible with that of your programmer. Suitable programmer circuits can be found on the Internet as well as within the pages of *Elektor Electronics*, and further information is provided on Atmel's website. The sensor inputs can be used for any desired purpose or simply left floating. With the hardware built we next turn to the accompanying Smartphone software. You will first need to install the appropriate version of Compact Framework 2.0 (available for download from Microsoft):

- Windows Mobile 5.0 Pocket PC and Smartphone: NETCFv2.wm.armv4i.cab;
- Pocket PC 2003 and 2003 SE: NETCFv2.ppc.armv4.cab;
- Windows XP: netcfsetupv2.msi.



Next, copy the file `avrblueremote.exe` to the target platform (for example onto an SD card for use with a Smartphone). The two Bluetooth devices, the Smartphone and the receiver board, now need to be 'paired' (which only needs to be done once). The program can now be started, and the COM port set with a click or two of the mouse on the upward- and downward-pointing arrows. The central square opens the interface, creating a connection with the receiver. The arrows now allow the outputs on the receiver board to be switched on and off; the sixth output is switched by pressing the space key.

Towards the top of the display the six sensor inputs are represented as LEDs. If you should move out of range of the receiver, these LEDs will be extinguished and a timeout bar will appear. If the timeout period should expire without a valid signal being received the interface will be closed.

(070126-1)

### Web links

Author's website (in German): <http://www.clipswitch.de/avrblueremote.html>

Bluetooth module information: <http://www.amber-wireless.de/en/produkte/bluetooth/default.php?fnum=109221360256>

Bluetooth module manual: [http://www.amber-wireless.de/pdf/OPC1601\\_MA.pdf](http://www.amber-wireless.de/pdf/OPC1601_MA.pdf)

LMX9820A datasheet: <http://www.national.com/pf/LM/LMX9820A.html>

AVR programmer: [http://www.atmel.com/dyn/products/tools\\_card.asp?tool\\_id=2726](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2726)

AVR Studio: [http://www.atmel.com/dyn/products/tools\\_card.asp?tool\\_id=2725](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725)

# Robot Voice

Pascal Choquet

Fans of the film '2001 a Space Odyssey' will no doubt recall the polite yet sinister voice of HAL, the ship's computer. It stands to reason that all proper robots need a (not necessarily menacing) voice.

Those of you who imagine that a voice box would require a whole heap of ICs are mistaken; the ISD2500 ChipCorder family of ICs from Winbond contains almost all the necessary hardware in a single IC to record and playback audio messages. Included on the IC is a microphone preamp and AGC suitable for a low-cost electret type microphone, an output amplifier to drive a loudspeaker, memory, an oscillator, an A/D and a D/A converter. There are four basic models; 2560, 2575, 2590 and 25120, the numbers following 25 indicate the available recording time in seconds. The memory capacity of each version is actually the same but longer recording times are achieved by using a lower sampling rate. The chip with the shortest recording time therefore offers the best audio quality.

The simplest circuit required to use the device in playback mode only is shown in **Figure 2**, the only external components required are just two decoupling capacitors. This circuit can be used in the robot whilst the circuit shown in **Figure 1** can be used for both recording and playback. A socket for IC1 fitted in both circuits will allow the chip to be moved into the robot once the sounds have been recorded.

Recordings are made by following this sequence.

First switch S3 to record mode (a low on pin 27). A press of S2 now begins the recording which is ended by another press on S2; a third press of S2 starts the next recording period and so on. This can continue until there is nothing more to record or when LED D2 lights to show that the memory is full. Playback can be performed by momentarily toggling S1 and switching S3 into 'play' mode, now with each press of S2 the recorded messages will be sequentially played back. The recordings can be overwritten by toggling S1, switching S3 to record and then using S2 to start recording from the beginning again.

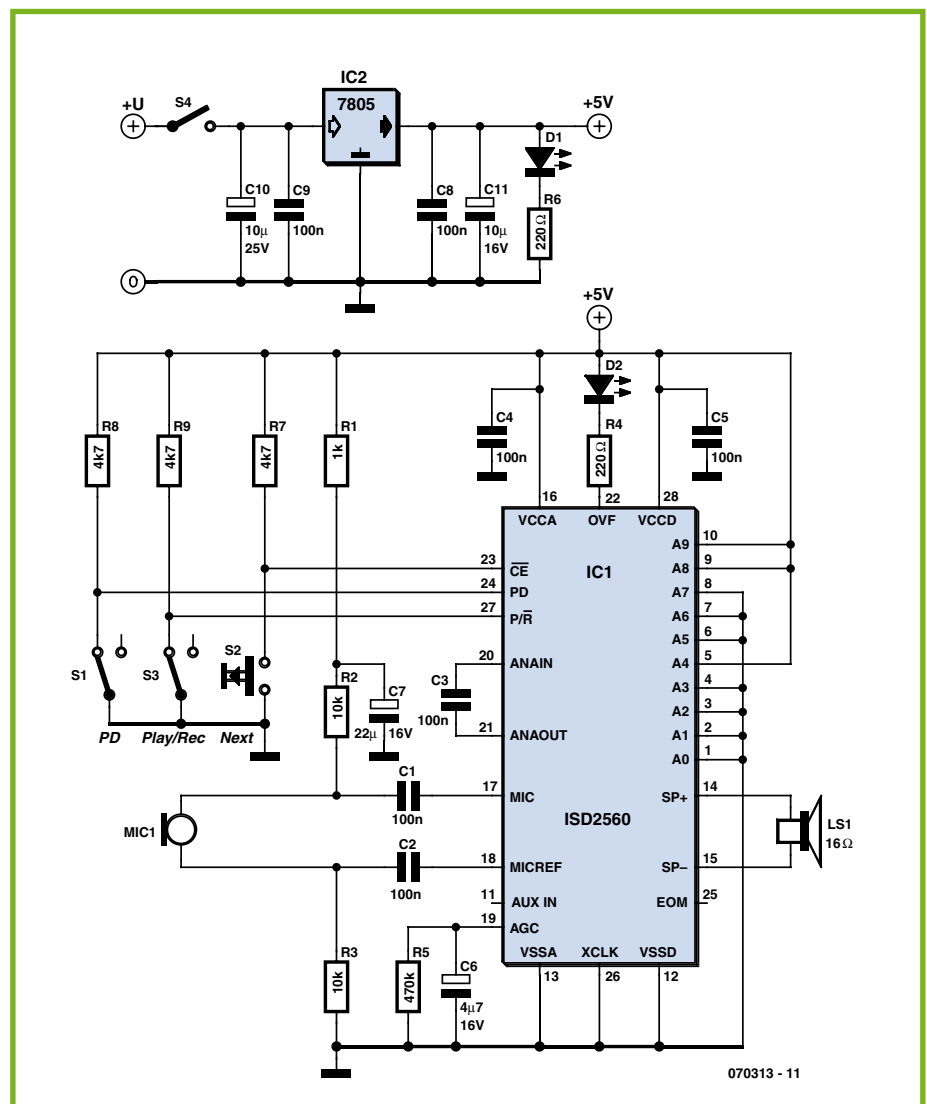
Some flexibility in the playback mode allows individual messages to be linked together; each recorded message is terminated by an EOM (End Of Message) flag when it is stored in the chip. Instead of

storing complete phrases like 'obstacle ahead' for example it is more efficient to store 'obstacle' then 'ahead', 'to the right', 'to the left' and 'behind' and likewise for numbers 'one', 'two', 'hundred' 'point' etc allows voicing of the complete range of numbers from these basic elements.

The minimum playback circuit shown in **uses** the A0, PD, /CE and /EOM signals interfaced to the robot microcontroller. For

low to fast-forward through the second message, waiting for the /EOM flag to go low. Once this occurs A0 is reset to '0' and a low pulse on /CE plays back the third message.

The /EOM output pulse can be less than 10 ms wide so it is better to use it to interrupt the processor rather than just poll its status. An example C code listing has been written by the author for a 16-bit Texas



playback PD is reset to '0' and to play the first message a low pulse is given on /CE. With A0 at '0' playback occurs at normal speed but with A0 at '1' the chip enters 'fast forward' mode where it advances through the message at 800 times its normal playback speed. When the third message needs to follow the first for example, the processor sets A0 to '1' and pulses /CE

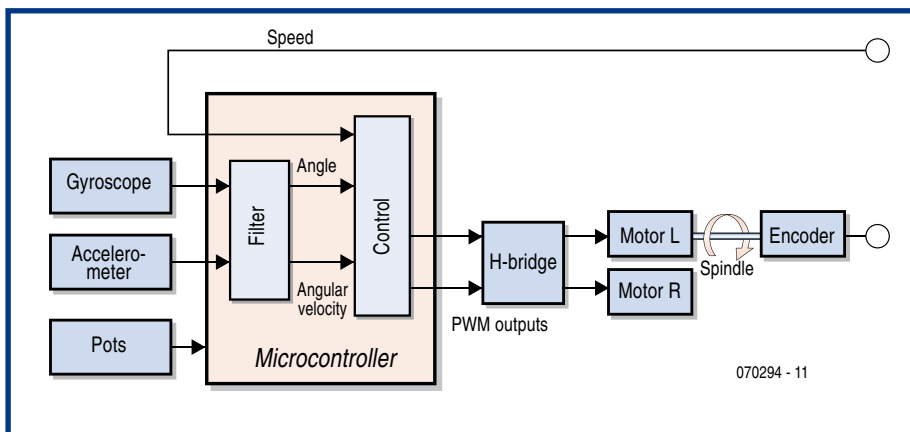
MSP430 microcontroller interfaced to this chip and is available for free download from the *Elektor Electronics* website, ref. **070313-11.zip**.

(070313-1)

## Web link

[www.winbond-usa.com/mambo/content/view/153/283/](http://www.winbond-usa.com/mambo/content/view/153/283/)

# Balancing Robot



David den Boer

The most famous balancing robot, also viewed from the general public's perspective, is the Segway, invented by Dean Kamen [5]. This little cart on two wheels is a mode of transport for people, and these days adorns the streets of many large cities. The principle of balancing on two wheels has inspired many robot builders to the construction of such a robot. A few other examples are the NBOT built by D. Anderson [3] and 'Joe le Pendule' built by co-workers at the Polytechnic School in Lausanne [2]. This type of robot comprises a number of sensors and a drive circuit clustered around one or more microcontrollers (Figure 1). When building such a robot, a few significant hurdles have to be surmounted. This article will hopefully help with this.

## The physics

The principle of a balancing robot is obviously to let the robot move forwards and backwards in such a way that the robot remains upright, the centre of gravity of the robot has to be always directly above the robot. A simple analogy is the balancing of a broom handle on the tip of a finger. This problem is sometimes also known as the 'inverted pendulum problem'. From your physics classes you will no doubt remember that the period of the pendulum is the square root of the quotient of the length and gravity, a longer pendulum therefore has a longer period (see Equation 1). Roughly, it can be deduced from this principle that a balancing robot with a high centre of gravity is more stable and thanks to the longer period is easier to keep balanced. A first simple step in obtaining a good functioning balancing robot is therefore building a robot with a high centre of gravity. This can be done by building a tall robot, but also by the high placement of

heavy parts, such as the batteries, or by artificially raising the centre of gravity with additional weights at the top of the robot.

## Control

A number of variables need to be known to control the robot. The first variable is obviously the angle of the robot: if the robot is not vertical it will continue to fall over. The speed at which this angle changes (angular velocity) is the second variable that is important. If the robot moves through the balancing point with a certain speed, the angle of the robot at that instant is zero degrees. The robot is not stable however, it is, after all, moving through the balance point with a certain angular velocity. This has to be anticipated by the controller; so this angular velocity is also important when controlling the robot. Finally, the speed of the robot is of importance, since it is the intention to control the position of the robot. By feeding these three variables back to the motors the robot can be controlled into a stable position (see Equation 2). A mathematical/physical basis of this control strategy can be found in [1], among others, including the derivation of the relevant equations of motion.

## Sensors

To determine the aforementioned variables an accelerometer and a gyroscope are usually used.

With an accelerometer the acceleration that the sensor is subject to can be measured. The direction of acceleration is also measured. A sensor that is frequently used is the ADXL202 from Analog Devices, which can measure acceleration in two mutually perpendicular directions. Because the sensor is also sensitive to the static acceleration due to gravity (g), the

sensor can also determine the angle of the robot with respect to the Earth. The acceleration observed by the sensor is shown in Figure 2 with the vectors  $am1$  and  $am2$ . When the sensor is not subjected to any other acceleration, the sensor is only sensitive to the angle of the sensor with the direction of gravity (and gravity itself, which is constant). As soon as the sensor is also subjected to a dynamic acceleration (a), the sensor will observe this as well. This is the case, for example, when the robot is driving forwards or backwards.

The resulting acceleration  $am2$  then depends

on the dynamic acceleration, gravity and the angle.

This gives a direct insight into the problem:

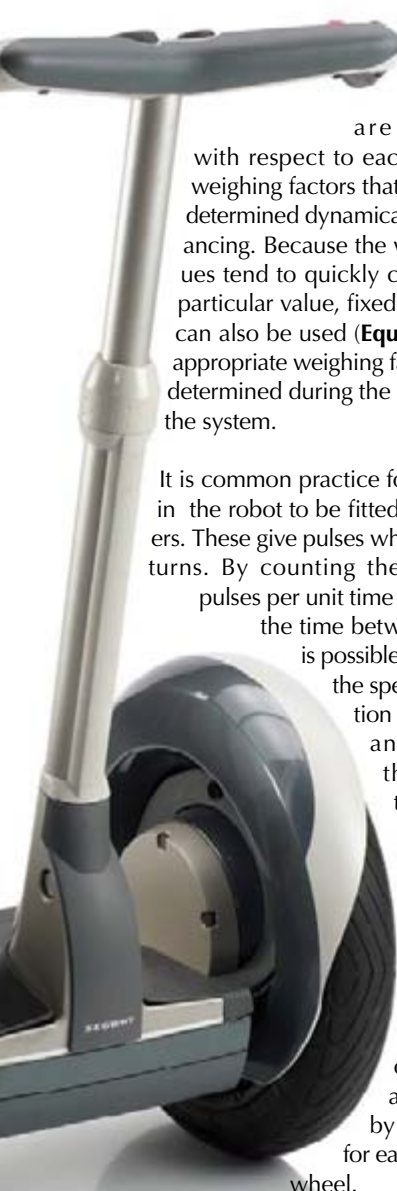
the instant that the robot moves, the signal from the sensor cannot be used any more to directly determine the angle of the robot.

Using a gyroscope, the angular velocity of the sensor can be determined. The output signal is directly proportional to the speed at which the sensor rotates around its own axis. By simply integrating this signal the angle of the robot can be calculated. The accuracy is a problem however. If the sensor has a small static offset of, for example, 0.1 %s per measurement then the error after 100 times of measuring and integrating has increased to 10 °. It is possible to build a balancing robot based solely on a gyroscope, but you will see however that after some time the robot will start to swing and become unstable. The small amount of drift that these sensors have is amplified by the process of integration.

By combining a gyroscope and an accelerometer the disadvantages of both sensors can be compensated for. In general this is done with a so-called Kalman-filter. This sums the result from the gyroscope, the value from the accelerometer and the calculated angle from the previous measurement cycle. These three values when summed







are weighted with respect to each other. The weighing factors that are used are determined dynamically while balancing. Because the weighing values tend to quickly converge to a particular value, fixed relationships can also be used (**Equation 3**). The appropriate weighing factors can be determined during the calibration of the system.

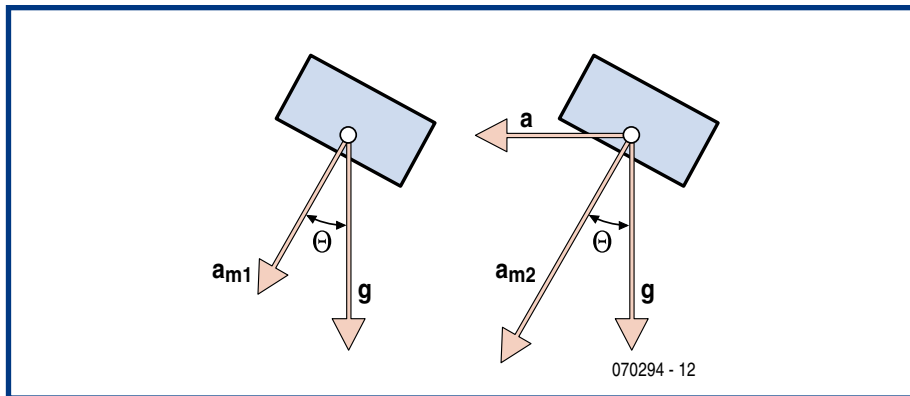
It is common practice for the motors in the robot to be fitted with encoders. These give pulses when the motor turns. By counting the number of pulses per unit time or measuring the time between pulses it is possible to determine the speed of revolution of the wheels and therefore the speed of the robot. Which of these methods gives the highest resolution is determined by the number of pulses that are produced by the encoder for each turn of the wheel.

### Actuators

Another special point of interest are the actuators. While balancing the robot the motors have to be frequently switched into the forward and backward direction. However, the combination of motors and gearbox is not without friction or backlash. When the voltage across the motors increases this does not immediately result in power from the actuators, there is a certain amount of offset. This is not a problem in many applications, but in the case of the balancing robot this null point is passed all the time; we are controlling around this null point after all. By compensating for this offset in software when driving the motors the stability of the robot will improve dramatically. Apart from that, the capability of the robot, the nimbleness in staying upright are also determined by the capacities of the motors and batteries. The greater the amount of power that the motors can deliver the better the robot is able to remain upright.

### Architecture

In the control loop that is executed by the software in the microcontroller a number



of things have to be carried out simultaneously. The sensors have to be read, calculations have to be carried out and the actuators for the robot have to be driven. At the same time it is often also desirable to communicate with a PC for the purpose

of data acquisition. For all this it can be a good idea not to use one relatively powerful processor but to use a number of smaller processors which send their data to one central processor. In this way the timing of measuring and controlling can

$$T = 2\pi \sqrt{\frac{l}{g}}$$

$T$	period (s)
$g$	gravitational acceleration (m/s <sup>2</sup> )
$l$	length from pivot to centre of mass (m)

$$PWM = k_1 \cdot \theta + k_2 \cdot \dot{\theta} + k_3 \cdot v$$

$\theta$	Angle of the robot (°)
$\dot{\theta}$	Velocity at which the angle changes (°/s)
$PWM$	Drive for the motors 0-100%
$v$	Speed of the robot (m/s)
$k_1 \dots k_3$	Feedback factors, constant

$$\theta_B[n] = k_4 \cdot \theta_B[n-1] + k_5 \cdot \dot{\theta}_G[n] + k_6 \cdot \theta_v[n]$$

$\dot{\theta}_G[n]$	Angular velocity measured by the gyroscope (°/s)
$\theta_B[n]$	Angle of the robot, calculated in measurement n (°)
$\theta_v[n]$	Angle of the robot, determined by accelerometer in measurement n (°)
$\theta_B[n-1]$	Angle of the robot, calculated in measurement n-1 (°)
$k_4 \dots k_6$	constants

be divided, which make the programming task a great deal easier.

## Sum of parts

A balancing robot consists of a number of parts that together have to hold the robot upright: sensors, any filters, a controller and drive system for the robot. Putting all this together in one go is very ambitious, a better chance of success is obtained by first testing and calibrating the individual parts. A good method for this is the temporary addition of a small arm to the robot. This arm is attached to the robot so that it can hinge with the aid of a potentiometer, which functions as the hinge. The other end of this arm is fitted with a wheel that rests on the floor. When the robot loses its balance the position of the potentiometer changes and therefore also its output. The output of the potentiometer has a direct relationship with the angle of the robot and can be used for calibration purposes. Note that it is important to choose a potentiometer with a shaft that turns easily.

The next step is the controller. Because of the arm, a read-out of the angle is available and this can be used to test and calibrate the controller for the robot. As already mentioned, feedback from the angle of the robot, angular velocity and speed of the robot are essential for the successful control of the robot. It is possible to calculate the necessary feedback factors mathematically. However this makes a complete and detailed physical description of the robot and the behaviour of the motors essential. It is simpler to determine these factors experimentally. This can be done,

for example, by connecting a number of digital or analogue potentiometers to the microcontroller. The program that runs in the microcontroller reads the position of the potentiometers and converts the relevant values into feedback values. The calibration can now be done with a lot of patience. A first step is to increase the feedback factor for the angle. If this factor is too small then the robot will react slowly; if this is too large then the robot will quickly oscillate around the balance point. In the latter case the feedback factor can be reduced and the feedback factor for speed can be increased. In this way the robot will quickly gain in stability. A final step can be made by increasing the feedback for speed. This will result in better positioning for the robot. A feedback factor that is too large however will make the robot very unstable.

A second step is the calibration of the sensors. The measurement values from the sensors (gyroscope, accelerometer) can be read into a PC via the microcontroller. Sensors that provide a read-out in the form of a pulse width modulated signal are quite common. By comparing the measured pulse width with the reading from the potentiometer that is mounted to the arm the angle and angular velocity can be easily derived. During the calibration, the drive for the robot can be switched off. By moving the robot back and forth by hand the angle and angular speed are changed and the calibration can be performed. When calibrating the accelerometer it is important to move the robot back and forth very slowly, so that the dynamic acceleration is as small as possible and only the accelera-

tion due to gravity is observed.

A third step is the calculation of the angle and angular velocity: the results from the sensors have to be combined in such a way that the angle of the robot and the speed at which this angle changes is obtained. The calculation that makes this possible has been described above. The weighing factors can be determined by moving the robot for some time (1 minute, for example) and reading the values from the sensors (accelerometer, gyroscope and potentiometer on the arm). A spreadsheet on the PC can subsequently be used to analyse the values and determine the correct values for the weighing factors.

(070294-1)

## Web links

- (1) <http://robotics.ee.uwa.edu.au/theses/2003-Balance-Ooi.pdf> This document describes a final year project during which a balancing robot was built. The research is quite detailed with respect to the physical and mathematical modelling of the problem.
- (2) <http://leiwwww.epfl.ch/joe/> Polytechnic school of Lausanne. The website is in French but there are a number of nice movies.
- (3) <http://www.geology.smu.edu/~dpa-www/robo/nbot/> This is the website of D. Anderson who built the NBOT.
- (4) [www.dena.demon.nl](http://www.dena.demon.nl) Dutch website on the construction of a balancing robot.
- (5) [www.segway.com](http://www.segway.com) Importer of the Segway.
- (6) [www.sparkfun.com](http://www.sparkfun.com)

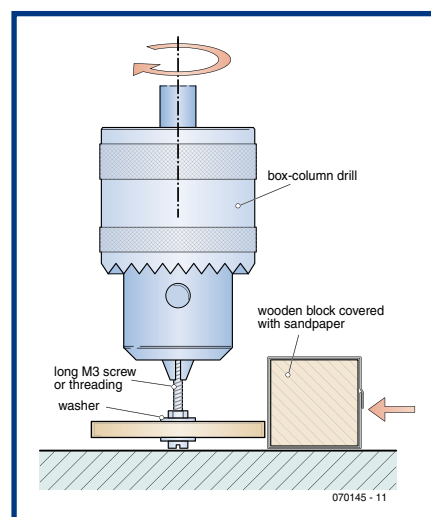
# DIY Wheels

## Marcus Bindhammer

You don't have to reinvent the wheel for your robot, but you may have to make your own somehow if you can't or don't want to buy them ready made. The hand-crafts aspect of making your own wheels is relatively undemanding, and one hand (but not a left hand...) is enough.

Use a compass to draw circles with a diameter of 50 mm on a piece of 5-mm plywood, and then cut or saw the discs out. Drill a 3-mm hole in the centre of each disc.

As you can see from the drawing, an M3 screw with a length of at least 30 mm must

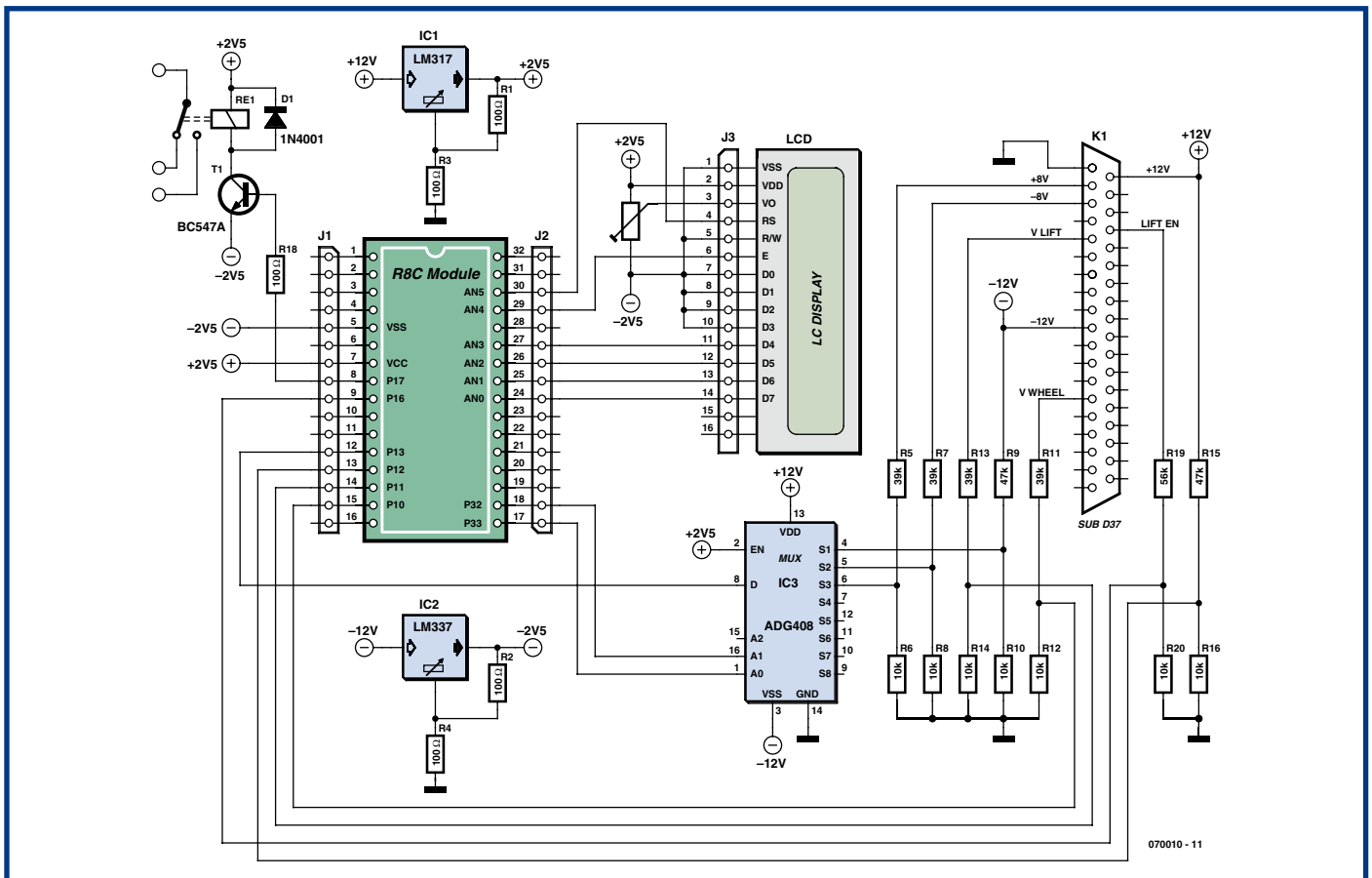


be fitted in the hole. Now secure the screw using a washer and nut, and then clamp it in the chuck of a drill press. Using a small block of wood wrapped in sandpaper, you can quickly sand the disc into a nice circular shape.

Finally, fit a length of toothed rubber belt around the circumference of each disc (old video recorders are a good source) or glue flat elastic bands to the running surfaces. After you remove the screws, all you have to do is fit the wheels on the axles and secure them with a bit of superglue.

(070145-1)

# R8C Measures Negative Voltages



Sven van Vaerenbergh  
UH Brussels, Radio Therapy Dept.

The author had to design a monitor circuit for the so-called Hercules Table (an electrically movable platform) at the University Hospital in Brussels that would watch several operating voltages and could stop the table in case of emergency.

The movement of the platform (up/down) is done with a DC voltage between +8 V and -8 V, where the magnitude of the voltage determines the speed and the polarity determines the direction. There are also a number of fixed voltages that need to be monitored.

An R8C module from Elektor Electronics was selected for the implementation of this project, because it is very easy to connect an LCD module to it and because it has a good A/D converter with 10-bit resolution on board.

The connection for the LCD (2x16 char.) is identical to that of the application board in Elektor Electronics (March 2006, page 38). In addition to the controller, a multiplexer of the type ADG 408 is used. This is required because we need to measure multiple voltages.

An unusual requirement for this project is that both positive and negative voltages need to be measured. Normally, the A/D converter of the R8C can only measure voltages between 0 and 5 V. The solution was found by using a symmetrical voltage for the R8C module, that is,  $\pm 2.5$  V. These voltages are generated by an LM317 and an LM337. We normally connect ground (Vss) to pin 5 of the R8C module. In this case that becomes -2.5 V. In this way we can measure voltages from +2.5 V to -2.5 V. The only places in the schematic that are connected to ground are the voltage dividers for the two voltage regulators (R1 to R4) and the voltage dividers (R5 to R16, R19 and R20) for the signals to be measured (available at connector K1, where all the important signals from the Hercules Table are available).

We can see that the schematic also contains a relay that can be switched by the R8C to immediately stop the table movement. This is optional.

Now we have to take special care when we have negative voltages. The resolution of the 10-bit A/D-converter in the R8C amounts to  $5 \text{ V}/1024 = 0.00488 \text{ V}$  per bit. Because of the symmetrical power supply,

the converter will give an output value of 512 when the input voltage is zero volts. We can display a minus sign on the LCD when the value is smaller than 512.

In C code it looks like this:

```

if ( u < 512 ) // less than
0, then minus sign
{
    lcddata (45);
}

```

We also have to adjust the voltage levels in software, because these have been lowered by the 6 voltage dividers. We also have to drive the multiplexer. In C code it looks like this:

```

pd3_3 = 1; //port 3.3 as output
p3_3 = 0; //port 3.3 => for
channel selection via analogue MUX
pd3_2 = 1;
p3_2 = 0; //port 3.2 => for
channel selection via analogue MUX

```

Three channels are directly connected to the A/D-converter. One channel is multiplexed for the voltages of +8 V, -8 V and -12V.

(070010-1)



# Formula Flowcode Buggy

## A low-cost robot not just for schools & education

Bart Huyskens,

St Joseph's institute (Schoten, Belgium)

**Out of Belgian schools has arrived an £ 85 (125 €) robot buggy that the designer hopes can reverse the decline in the study of electronics and technology across Europe and, potentially, wider.**



Like many in the engineering teaching fraternity, the author is increasingly aware that electronics – and most engineering and science related activities – fail to attract young people. To people who have grown up with a fascination for technology this is a great shame and it really is very strange when you consider that young people are fascinated by all things electronic – like i-pods, digital cameras and mobile phones. But for some reason our schools fail to capitalise on this interest and attract young people to technical careers.

Long and hard thought was given about how to solve this and what the real problems are (including the facts that electronics can be mathematical, parts of the subjects are difficult to understand, and it has a very dull image). All sorts of ideas have been tried, and some have partially worked, but not to our satisfaction. There is one exception that stands out – Lego. Lego's NXT robot looks cool, is cool, and is fun to use. But it is expensive, does not

really teach anything about electronics, and you can't really 'get under the hood' and use it as a platform for learning more than just simple programming.

About six months ago, during a conversation with engineers at Matrix Multimedia in England, we had an idea that might go some way to solve this problem – a programmable robot buggy of our own, with more functionality than the Lego NXT, that does have 'open architecture' which can be used for a wide range of teaching and learning activities in electronics and technology. In the six months since, we have been working hard to make these ideas come to life and we have called the project 'Formula Flowcode'.

Formula Flowcode is a complete hardware and software robotics solution for learning about electronics and programming. This article describes what Formula Flowcode does, how it works, how we hope it can be used to learn all about robotics...

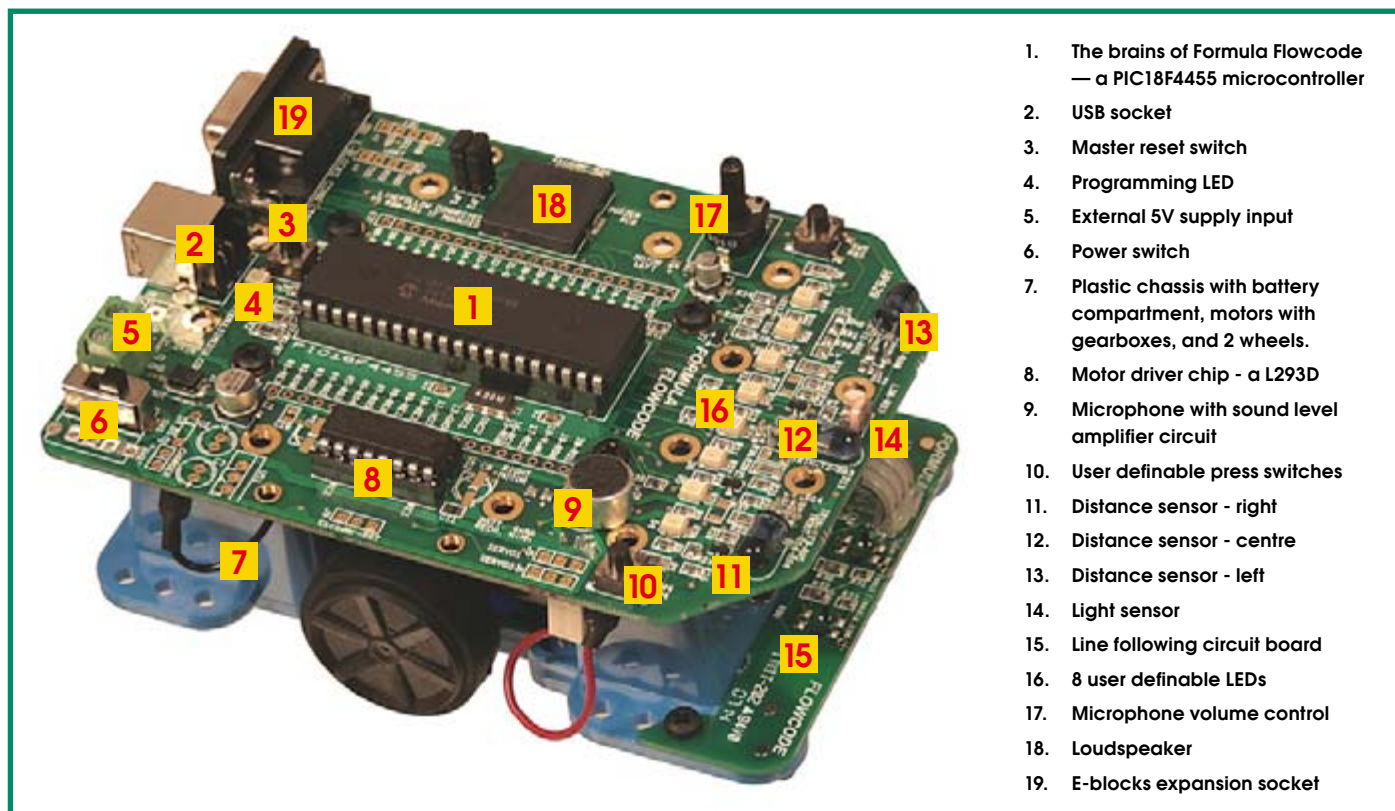


Figure 1. Formula Flowcode functional parts.

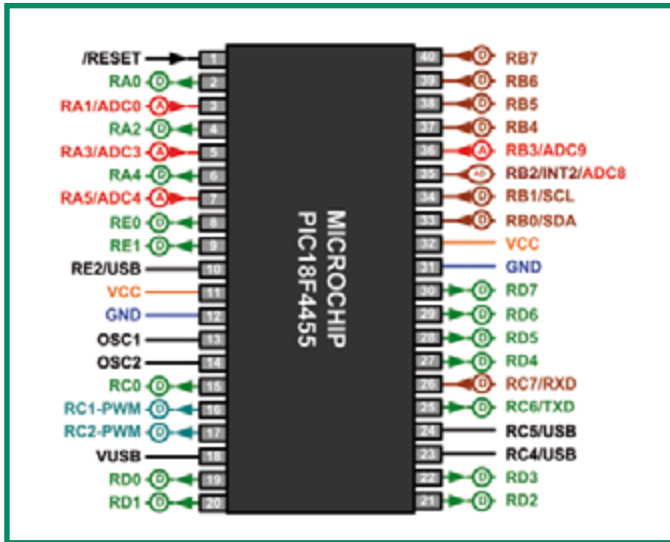


Figure 2. This diagram of the PIC18F4455 is purposely simplified, showing only the actually used functions of the pins of the PIC. D stands for 'Digital I/O' and A for 'Analogue input'.

## And here it is

Figure 1 shows a picture of Formula Flowcode and a list of the components and features on the buggy. But how does it all work? Let's start at the beginning. The whole Buggy is designed around the new and very powerful PIC18F4455 microcontroller from Microchip. This 40-pin device operates at 24 MHz and will execute programs at an amazing speed of 6 MIPS. A meet & greet diagram of this CPU is shown in Figure 2. The device connects directly to your USB port and contains a bootloader program so that it can be programmed using a version of Flowcode 3 supplied free of charge with the buggy. The PIC18F4455 has two separate hardware PWM outputs, a UART, I<sup>2</sup>C, Analogue Inputs, Pin-, Port- and Timer Interrupts and a lot of Digital I/O.

## Driving DC Motors

The two DC motors with separate gearboxes are powered by a classic L293D IC with two full H-bridges, see Figure 3. The direction and speed of each motor can be programmed separately in Flowcode and this makes the buggy capable of performing the craziest moves you can imagine.

As you can see on the schematic, the L293D uses four PIC outputs for its control. Two outputs (RE0 and RE1) set the direction while two (hardware generated) PWM signals at RC1 and RC2 govern the speed of each wheel.

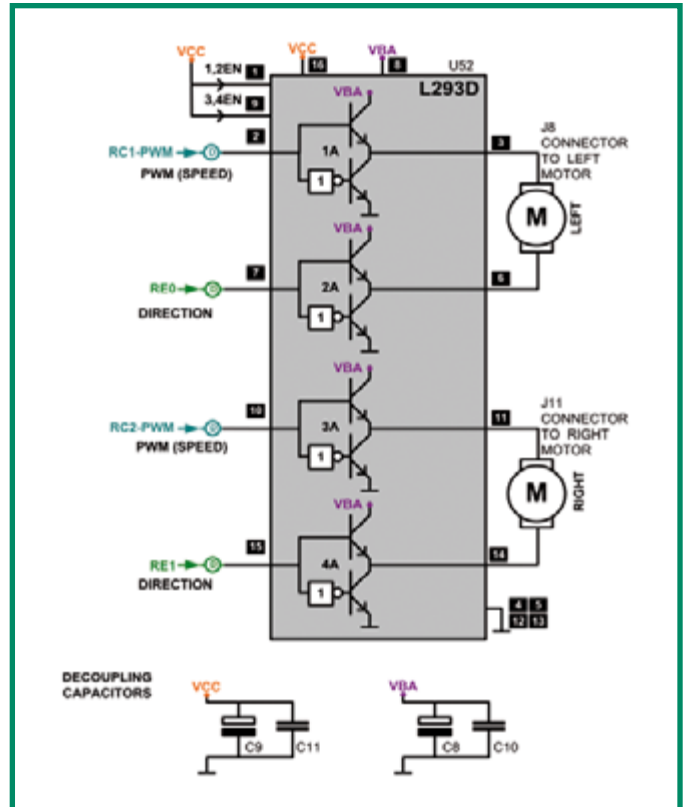


Figure 3. The L293D Double H-bridge motor driver circuit.

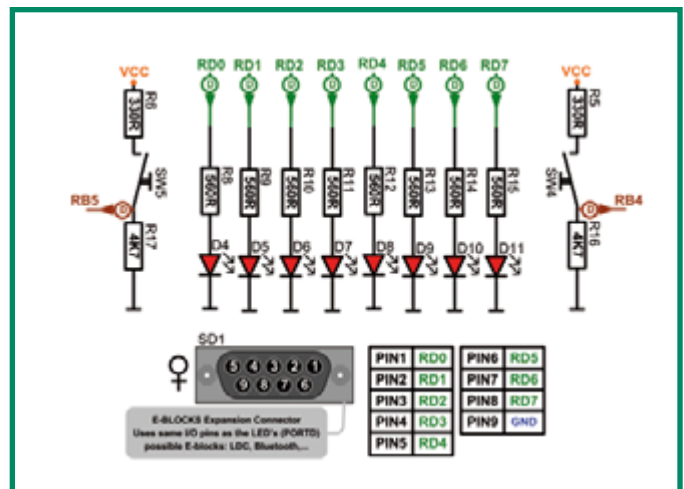


Figure 4. Two switches, eight LEDs and the E-Blocks D-type connector.

## LEDs, Switches and E-blocks Expandability

The schematic of the I/O hardware is shown in Figure 4. The eight LEDs and two switches at the front of the Buggy will come in handy for your first steps with Flowcode and will prove to be very useful when debugging your more complex programs on this 8-bit PIC microcontroller. In parallel with the eight LEDs, Formula Flowcode has an E-blocks connector. This will give the users the capability of very easily expanding the Buggy with standard E-blocks like LCD, Bluetooth and many more.

## Sound I/O and light sensor

The buggy can react to sound (hand claps) using the amplified microphone circuit connected to RB2. As shown in Figure 5, this sound sensor may be used as a digital input, an external interrupt

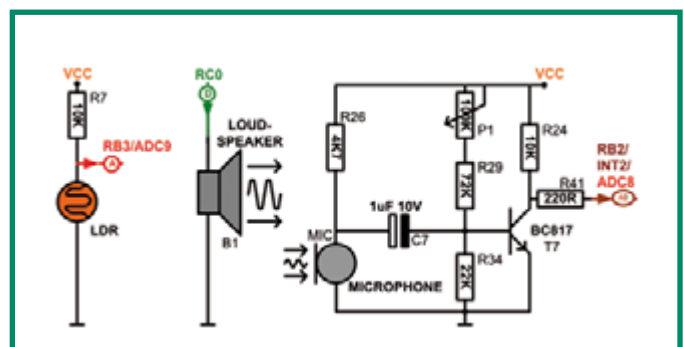


Figure 5. The microphone, loudspeaker and light sensor circuitry.

and an IR sensitive photo transistor producing an analogue voltage that's inversely proportional to the amount of IR light reflected.

### Line following circuit

Also on the front of the buggy is a small daughterboard that contains two I<sup>2</sup>C intelligent infrared transceivers capable of detecting black and white surfaces at distances from 1 mm up to 6 mm without any error. The circuit is shown in **Figure 7**. These sensors allow the buggy to follow lines on a table or mat.

### Power supply

This section of the circuitry is shown in **Figure 8**, using a combination of pictorial elements and of course the schematic. The small chassis is powered from four NiMH AA rechargeable batteries which give between 4 and 6.2 V. The circuit board also has space for additional components that allow you to connect larger batteries to the chassis, using a 5 V regulator.

### Additional circuitry

In addition to the standard circuitry discussed above, the chassis is also fitted with a fair number of expansion connectors. These include I<sup>2</sup>C expansion, wheel encoder inputs, servo inputs, and much more.

### Using Formula Flowcode

By now you should have a good idea of how the hardware works and what makes it tick. But how is it programmed, how is it used, and how do students know what to do?

The Formula Flowcode buggy is supplied with a free (reduced functionality) copy of Flowcode – a graphical software language for microcontrollers. Flowcode allows users to directly download a program to the buggy using the USB lead. On removing the USB lead and pressing the reset switch, the buggy starts to run the program.

### Courseware for pole position

The new version of Flowcode released with the buggy includes a pulsewidth modulation (PWM) component for controlling the motors. As Flowcode is now available in around 10 languages (including Chinese) it will be usable by children as young as 12 virtually anywhere in the world.

This robot buggy is officially named 'Formula Flowcode', after the Formula Ford where the cars are all identical and winning only comes down to the driver's skills. In this case, winning comes down to the *programming skills* of the user. As you might expect there are a number of separate 'courses' that users have to complete to increase their skills level. These courses start with getting a single LED to light up, and finish with full maze solving using a custom made chassis, using wheel encoders and what have you. This is the really clever idea behind Formula Flowcode — it is great for complete beginners to robotics and electronics, and it will also provide a considerable challenge to those with relevant degrees in electronics and computer science.

The courses include such subjects as:

- Light a single LED;
- Make a 'NightRider' effect on the LEDs on the front of Formula Flowcode;
- Develop a program that uses the on-board light sensor to allow

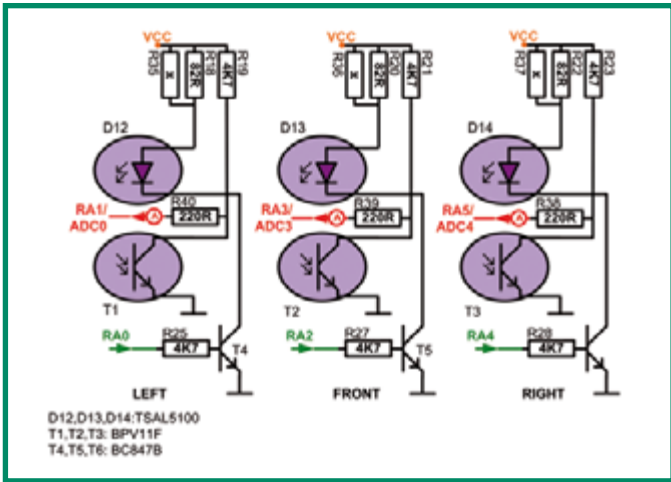


Figure 6. Distance measurement circuit on board the Formula Flowcode Buggy.

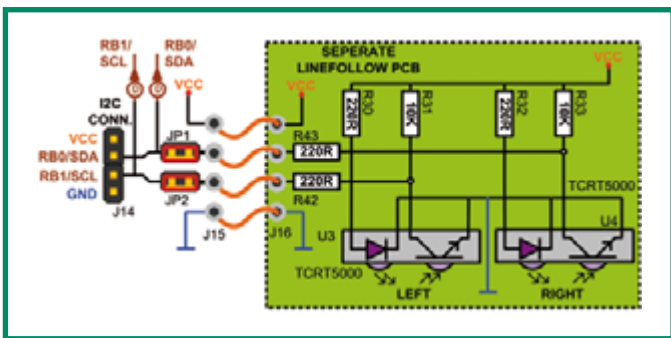


Figure 7. Line follower circuit.

or even as an analogue input. The buggy also includes a simple high impedance speaker that can be used to generate frequencies between 100 Hz and 17 kHz. At the front of the buggy is a small forward facing light sensor that allows the vehicle to measure light intensity in the forward direction.

### Infrared distance measuring circuit

The buggy also includes three distance sensors on the front, left and right of the main circuit board. **Figure 6** shows the circuit diagram. A single sensor is a combination of an IR LED that emits IR light,

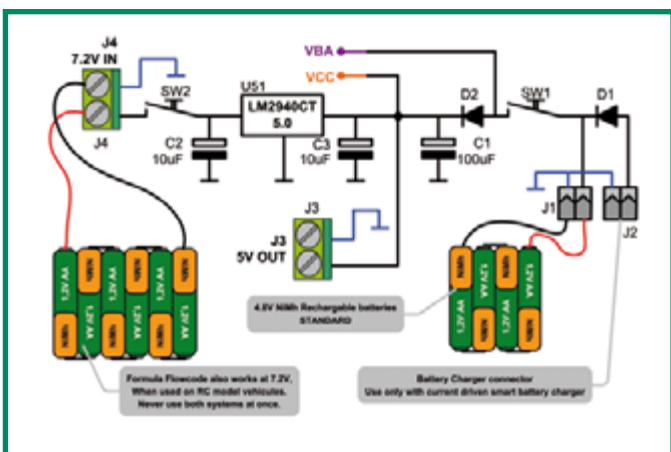


Figure 8. The simple but all-important power supply.



Formula Flowcode to steer towards the light from a hand held torch;

- Develop a program that allows Formula Flowcode to 'dance to the music'. Every time you clap it must turn through 90 degrees;
- Develop a program that allows Formula Flowcode to follow a 2-metre long line made up from a black insulating tape. Beat a time of 30 seconds;
- Develop a program that allows Formula Flowcode to solve a 64-cell maze using the left-hand wall following technique.

These courses are all explained to students in the form of task-based worksheets. Some are competitive, in terms of time or function, and will form the basis for regional competitions that we hope will be run by Elektor and educational institutions across Europe. Other courses are just



Figure 9. Typical line following exercise.

## Pit stop! How can you get involved?

The Formula Flowcode robot buggy was purposely developed to motivate people to want to learn more about robotics and electronics – from 12 year old pupils who have a curiosity about the subject, right up to those enjoying retirement and still wanting to learn and keep mentally active. In the USA the First Robotics programme ([www.usfirst.org](http://www.usfirst.org)) has been quite successful at stimulating engineers in industry to collaborate with young people to compete in robotic events. We hope to achieve something similar here; by providing a low cost hardware software robot, providing online support and quality curriculum. By running workshops and competitions we hope people will be captured by electronics. If you feel that our aims are worthwhile and achievable, if you want to take part in this programme, or if you are interested in the competitions and workshops then please let us know. If the level of interest is high then we can write more articles and issue special 'courses' for Elektor readers. The author and his colleagues can be contacted by email on [barthuyskens@scarlet.be](mailto:barthuyskens@scarlet.be), or through Elektor.

**You can purchase a Formula Flowcode buggy from the Elektor SHOP at a cost of £ 85 or € 125 including VAT.**

for fun or in-depth exploration of programming and hardware features. Support for all of this will be available on Matrix Multimedia's website forum accessible via [www.matrixmultimedia.com](http://www.matrixmultimedia.com) where users ask each other questions and swap programs. We also plan to run workshops on Formula Flowcode to get young people up and running and interested in technology.

In addition to this Elektor will be publishing a book which will take users through the process of developing programs and additional circuitry for the buggy. This will form a complete suite of tutorials which will teach technology to budding electronics enthusiasts up to 16 years of age.

(070323-1)



Figure 10. Solving a simple 'left' maze.

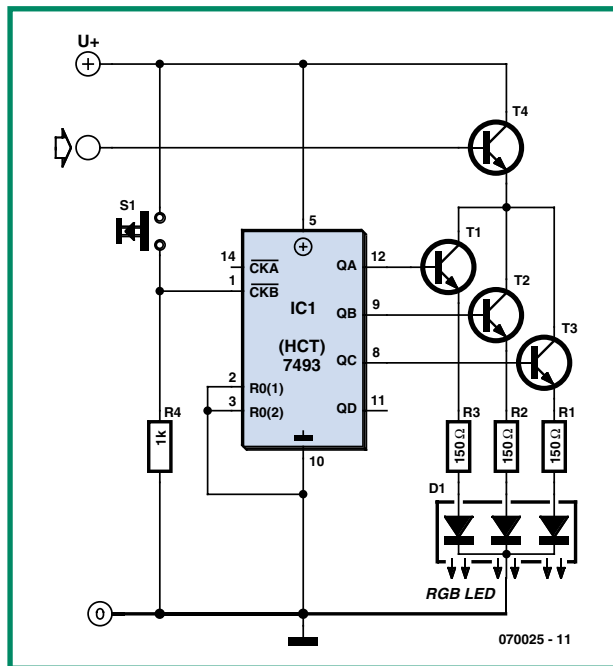
# Colour Tamer

## RGB LED adjustment

Nivard van de Boogaard

With this simple circuit you can easily control the three LEDs in an RGB LED. Each time switch S1 is pushed we increase the (digital) value at the output of the counter-IC, IC1, an HCT7493. So, with each button push, a different LED or combination of LEDs lights up; or, in the case of an RGB-LED, the LED produces a different colour each time. After the highest value the counter will start from the beginning again. In this way all colour combinations will have their turn and there is also no need to build a separate reset-facility. Nice to embellish a robot with, but it also gives a nice effect for a modded PC.

To prevent the outputs of IC1 from being overloaded, we have added T1 through to T3. As soon as an output from the counter-IC becomes active (high), it turns the corresponding transistor on and current will flow through the LED. To limit the current through the LEDs we deployed R1 to R3.



The type of transistor that is used for T1 to T3 is not very critical. Standard LEDs require relatively little current, about 20 mA. A BC337 will therefore work fine, as will a BC547.

The power supply to the LEDs can be switched off with T4. This is handy when,

for example, the circuit is built into a PC. As long as IC1 remains powered it will remember its counter value and therefore also the corresponding colour setting of the LEDs or RGB-LED. In this way it is not necessary to adjust the colour every time the PC is switched on. T4 needs to be able to handle more current than T1, T2 and T3, but even a BC847 with its 100 mA maximum collector current rating is still below the limit with three LEDs.

We use R4 to define the logic level (low) on the clock input when the switch is open.

The HCT7493 is a 4-bit binary ripple counter, which internally consists of four master-slave flip-flops that form a divide-by-two and a divide-by-eight. RO(0) and RO(1) can be used to reset these two sections. We don't need the

reset function for this application so we connected RO(0) and RO(1) to ground to prevent unwanted behaviour. If need be, the clock input \CKA could be connected to the power supply rail to make this input insensitive to noise.

(070025-1)

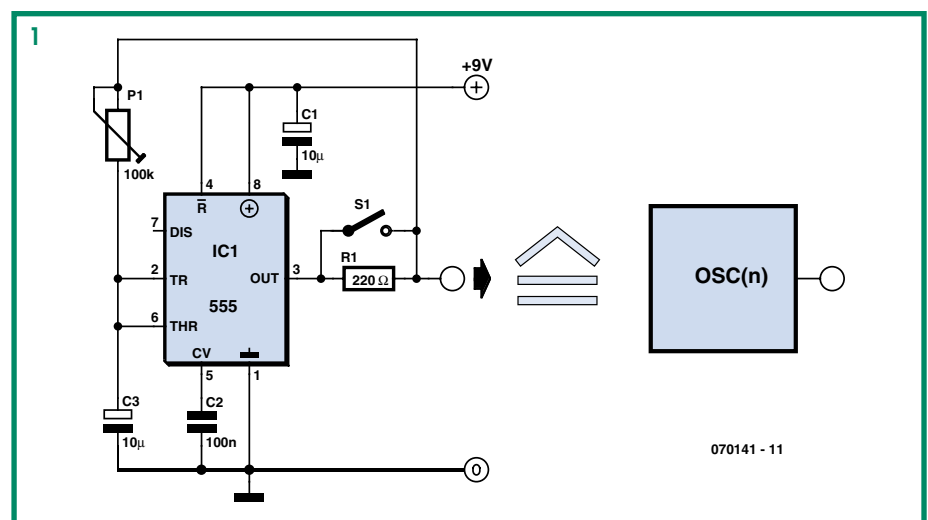
# Chaotic LED Fireflies

Jonathan Hare

Here we couple LED oscillators together to produce some interesting effects for an exclusive optical touch to your robot.

As seen in the circuit diagram in **Figure 1**, instead of using the discharge pin on a 555, the timing capacitor can be charged and discharged using the output (via a preset P1). If we assume the 555's output resistance is very low (i.e. use a bipolar rather than a CMOS 555) this circuit provides a 50:50 mark-space ratio whose output frequency is independent of load. However, if we deliberately increase the output resistance by using a series resistor (R1) the timing will now also be dependent on the current taken by the load (because R1 will effectively drop the available charging voltage to the P1/C3 timing circuit).

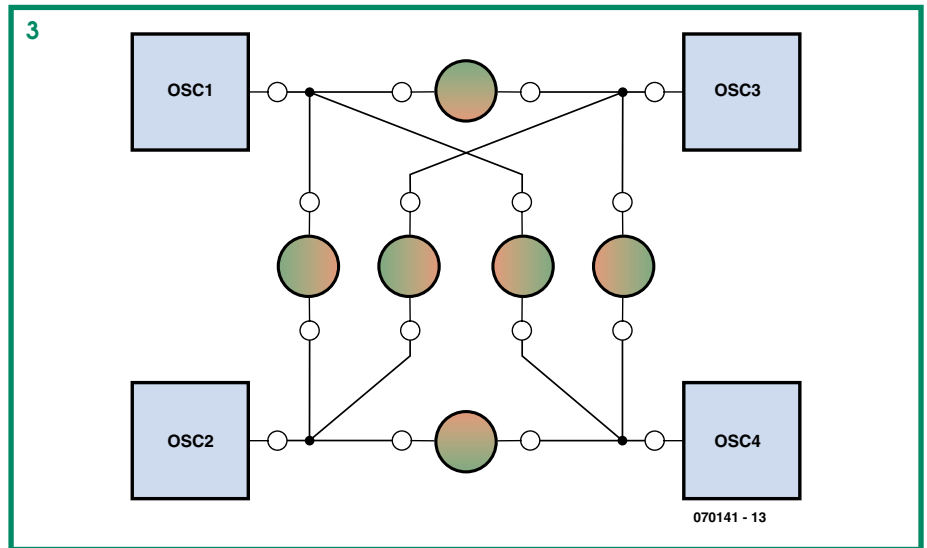
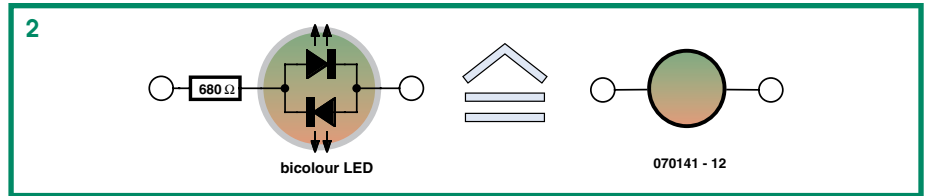
Now, imagine a number of such oscillators whose outputs are connected to each



other via current limiting resistors and bi-colour LEDs (**Figure 2**).

A possible constellation of oscillators and LEDs, each with their own symbol from

Figures 1 and 2, is shown in **Figure 3**. Each oscillator's timing will be dependent on the state of the other oscillators because these will determine the current that flows through the LEDs. For example, if all the outputs are High (or all Low) there will be no potential differences and so no current will flow through the LED circuits. In this case, all the oscillators will be at maximum frequency. Other combinations of outputs will light some of the LEDs and these currents will thus effect each oscillators timing. Chaos rules! The R1s therefore couple the oscillators to each other. A switch across each R1 allows control of the coupling. Setting the oscillator frequencies to about 2 Hz with the P1s shows the complex flashing of the LEDs switching between off, red and green. Sometimes the LEDs seem to settle down pulsing together. This is rather like an electronic version of what is observed in nature when a group of fireflies congregate in a bush — they pulse together and maybe our little circuit is a simple version of this rather complex natural feedback system. If the frequency is raised to ca. 100 Hz, varying mixing (beating) of the flashing red and green colours cause a 'wave' of changing colour to go through the array of LEDs. Including light dependent resistors (LDR) in series with R1 might be a way of making



each of our LED fireflies 'see' each other. Even without the LDRs, with three or more coupled oscillators there might also be the

intriguing possibility of observing chaotic behaviour of the oscillators.

(070141-1)

# Stepped Volume Control

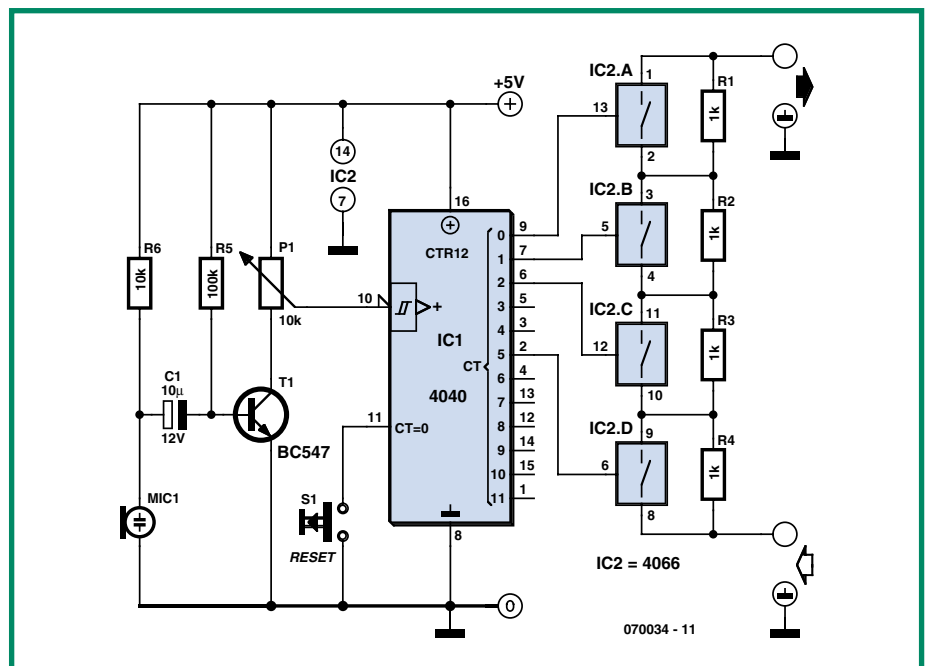
Raj K. Gorkhali

Louder music, sirens or speech in response to higher ambient noise levels? This simple circuit has the answer, and it may enable your robot to be at least as noisy or loud-mouthed as the others in an arena.

The circuit consists basically of a microphone, a level detector, a 4-state counter and four analogue switches connected to a resistive ladder network.

Looking at the circuit diagram, the signal from electret microphone M1 is amplified by T1 whose collector voltage appears across a potentiometer. M1 gets its bias voltage through R4. Depending on the setting of P1, the 4040 counter will get a clock pulse when a certain noise level (threshold) is exceeded. The counter state determines the configuration of the four electronic switches inside the 4066 and so the series resistance effectively seen in the audio signal line.

The circuit should be powered from a 9-V regulated supply or a battery and will consume a few milliamps only.



Switch S1 allows the counter to be reset, switching all 4066 switches to off, i.e., the

highest attenuation will exist in the audio path as in that case none of the 1-kΩ resis-

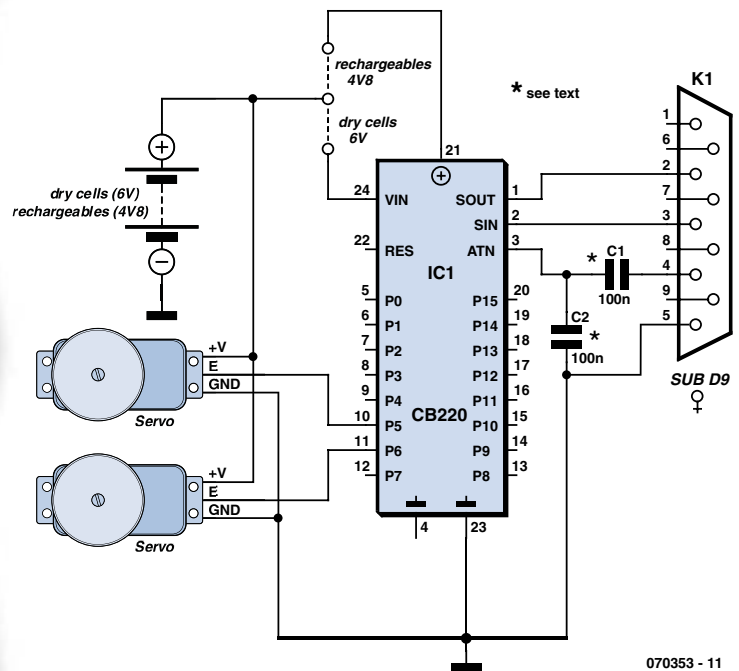


tors are shorted out. To calibrate the circuit, disconnect the 4040 clock input (pin 10) from the wiper of P1, and temporarily ground it through a 100 kΩ resistor. Now pulse the clock input by briefly connecting it to the +9 V line; you will see

the counter outputs change state and with them, the bilateral switches in the 4066.

(070034-1)

# Walking Works!



070353 - 11

## C. Tavernier

Although the majority of robots built by amateurs move around using wheels or caterpillar tracks, two categories occupy a place apart because of the spectacular way they move: the hexapod robot, also called spider robot (though spiders actually have eight legs!), and the walking robot. It may seem relatively easy to make the wheels turn using motors, which is the

sole requirement in a conventional mobile robot, but reproducing walking, be it of an insect in a hexapod or of a human in a walking robot, might appear markedly more difficult.

In this article, we'll see how to do this for a walking robot, which we've opted to buy in kit form for the mechanical part. Although these days any good handyman can build a 'conventional' mobile robot base, i.e. with wheels or tracks, the mechanical construction of a walking robot is much trickier, mainly because of the particular action involved in walking, which we'll be describing in a moment.

To our knowledge, there are currently only two mechanical kits for walking robots sold at an affordable price: the famous Toddler from Parallax ([www.parallax.com](http://www.parallax.com)) and, closer to home, the Yeti from Arexx Engineering ([www.arexx.com](http://www.arexx.com)). As these two robots operate in virtually the same fashion, in terms of the principle of

their walking, we settled on the kit for the Yeti, distributed by Lextronic ([www.lextronic.fr](http://www.lextronic.fr)), as he is slightly cheaper than his brother from Parallax. But everything we say about one is going to be true, with only minor variations, for the other too.

So, these walking robots both consist of two rigid legs, in the sense that they don't have a central articulation like our knee. They consist of two arms (that's a bit much for a pair of legs!) arranged to form a parallelogram, and have a foot articulated either side of the plane that forms.

Although it might not seem very much, all it actually takes to make such a robot walk is two normal — i.e. unmodified — radio-control servos. The first, clearly visible at the front of the robot, controls the feet by means of two long connecting rods, while the other, fitted under the robot's belly, acts on the rear arm of each leg. So having thus set the scene, let's now see how such a figure is able to walk.

To make it easier to follow our explanation, we're going to refer to 'right' and 'left', but obviously this is purely relative. When the robot is at rest, both feet are flat on the ground alongside each other. Then the foot servo turns in a direction that will make the robot's body lean over onto the left foot, which obviously has the effect of lifting the right foot off the ground. Then the leg servo turns too, making the right leg, free to move as its foot is now off the floor, move forward. Then the foot servo turns in the opposite direction, making the robot's body lean over onto the right foot, lifting now the left foot off the floor. The leg servo rotates again to make the left leg, itself now free to move, advance — and thus ends the first step.

As you will note from reading this, or if you have already watched the videos available on both the Parallax website for the Toddler and on the Lextronic site for the Yeti, the robot is actually in a constant state of near imbalance throughout its walk, which can only work correctly if the robot's weight is correctly distributed... and if the servo movements are neither too abrupt, nor too large in amplitude.

Even though these walking robot kits are also available with electronics chassis included, sometimes very elaborate as in the case of the Yeti with an ATmega8 processor from Atmel and its C compiler, a simple Basic Stamp II or a Cubloc CB 220 is enough for taking your first steps (literally as well as figuratively!) As shown in the figure, which applies to both these microcontrollers that are, don't forget, pin-compatible, apart from the microcontroller itself, no other active components are required to make our robot walk.

The servo control inputs are connected from two parallel ports which can be any ones in the case of the Basic Stamp, but must be P5 and P6 in the case of the Cubloc CB 220, as its PWM instructions only work on these two.

Capacitors C1 and C2 call for a comment: C1 and C2 only need to be fitted if using a BS II. If using a CB 220, C1 will be replaced by a wire link, and C2 will not be fitted, its two pads being simply left empty.

All the rest is just a matter of programming, based essentially around the PWM instruction in the case of the Cubloc and around PULSOUT in the case of the Basic Stamp. As you will have gathered from the explanation above, to make our robot walk, all we have to do is make the servos turn alternately in one direction or the other, in a well-defined sequence.

We are offering you two complete source listings for making our robot walk, one for Basic Stamp II and one for Cubloc, on download from the Elektor website or from the author's own site ([www.tavernier-c.com](http://www.tavernier-c.com)). Here are just a few comments to prove to you how simple they are and enable you to adapt them easily to your own needs.

Let's start by recalling that in the Cubloc the instruction PWM is used as follows:

```
PWM port, ratio, period
```

This generates on the corresponding port (0 for P5 and 1 for P6) a PWM signal whose duty cycle is defined jointly by 'ratio' and 'period'. The advantage of the Cubloc is that this instruction lets us generate the relevant signal continuously, as soon as it has been called at least once.

In the case of the Basic Stamp, the instruction PULSOUT is used in the following way:

```
PULSOUT port, duration
```

This generates a pulse on the corresponding port for a period equal to 2  $\mu$ s multiplied by the value of the 'duration' parameter. The drawback to PULSOUT compared with PWM is that this generation is not repetitive. So if we want to generate repetitive pulses, we need to employ loops, making the Basic Stamp version of the program a little more complicated compared with the Cubloc version.

Lastly, let's remember that a servo takes up the rest position when it receives 1.5 ms pulses, and moves towards its end positions in one direction or the other for pulses of 1.0 ms or 2.0 ms respectively. From this point on, it's easy to follow one or other of the suggested listings. To put the robot's feet into the rest position, for the Cubloc we write:

```
PWM 0, 3150, 32768  
PWM 1, 3150, 32768
```

And for the Basic Stamp II :

```
FOR Pulses = 1 TO 100 STEP 5  
    PULSOUT TiltServo, 750  
    PULSOUT StrideServo, 750  
    PAUSE 25  
NEXT
```

In both cases, this causes 1.5 ms pulses to be generated for both servos, hence placing them into the rest position. Note that, in the case of the Cubloc, it is necessary to adjust the 3150 parameters so that they take the servos correctly to the rest position, while for the Basic Stamp, it is necessary to adjust the two 750 parameters for the same purpose.

To lift one foot, we will therefore write for the Cubloc:

```
FOR Position = 3150 TO 2850 STEP  
-1  
    PWM 0, Position, 32768  
    DELAY 1  
NEXT
```

And for the Basic Stamp:

```
FOR Pulses = 750 TO 620 STEP -5  
    PULSOUT TiltServo, Pulses  
    PULSOUT StrideServo, 750  
NEXT
```

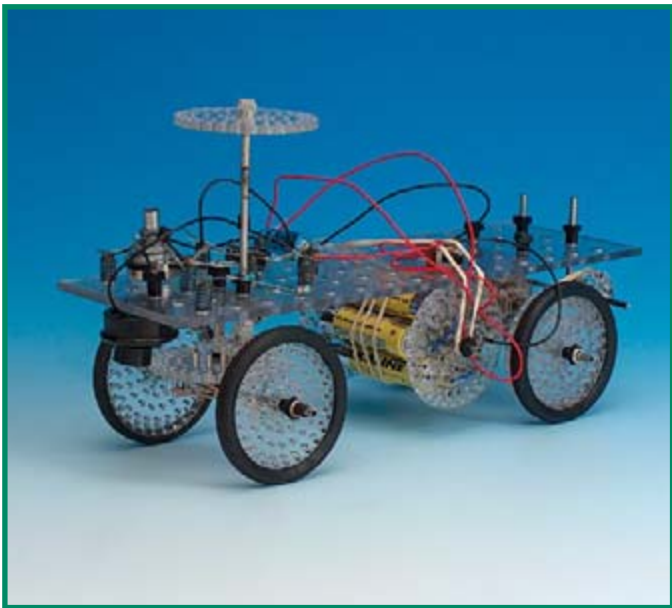
Note the increase in 'complexity' created by the fact that, unlike PWM, PULSOUT does not operate continuously. So we need to add into the Basic Stamp's 'lift foot' loop the generation of the pulses that will maintain the leg servo in the rest position. In the case of the Cubloc, this generation is taken care of automatically by the PWM instruction that originally put the servos into the rest position.

We'll leave you to analyse the rest of these two listings for yourself — as you can see, they are in fact nothing but a succession of the groups of instructions we've just been looking at, with numerical parameters each time appropriate for the pulses needed to move the servos into the required positions.

Before ending, let's just clarify that these two listings are extracts only. The Parallax one, originates from the Parallax website ([www.parallax.com](http://www.parallax.com)), from where we strongly recommend downloading the document entitled '*Advanced Robotics with the Toddler Robot*' (available in PDF format) containing a very good survey of the various methods of programming the walk. For the Cubloc, the full program may be found on the Lextronic website ([www.lextronic.fr](http://www.lextronic.fr)). We thank Parallax and Cubloc for their kind collaboration.

(070353-1)

# Philips ME Construction Kits



## Luc Lemmens

Many of our readers — especially the older generation — will remember the legendary EE (Electronic Engineering) experimenter kits. They were part of the Philips (not 'Phillips') product line for many years, from 1963 until well into the 1980s. Many electronics professionals owe their interest in the field to these kits. The first kits were primarily intended to promote Philips electronic components among hobbyists, but in the later years, the complexity of the designs increased and the range of kits was oriented more towards educational use.

In the late 1980s, Philips sold the electronic experimenting kits business to the German firm Schuco. It continued to sell them until the mid-1990s.

The ME (mechanical engineering) experimenter kits from Philips are much less well known, but they were only available for around five years in the 1960s. It would appear that the ME series was much less successful than the EE series, which is why they were only available for a relatively short time. Anyone who ever tried to work with them — such as the author — will certainly know why they weren't a resounding success like the EE boxes.

The ME kits couldn't compete with other mechanical construc-

tion systems that were very popular at the time, such as Meccano and Fisher. Many of the parts were much too fragile for clumsy children's hands, and in some cases the proposed structures did serious damage to the components. In particular, the pins that were used for gearwheels and all sorts of connections between axles were subjected

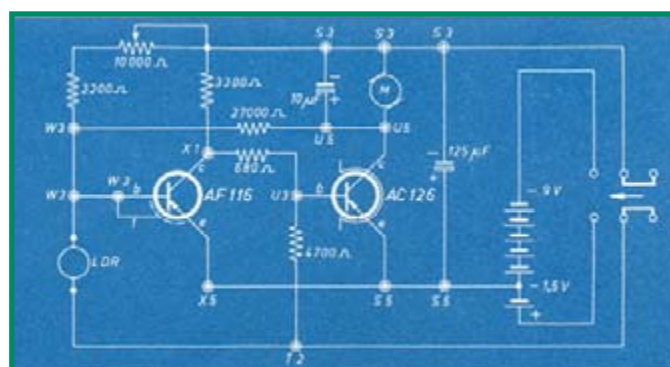
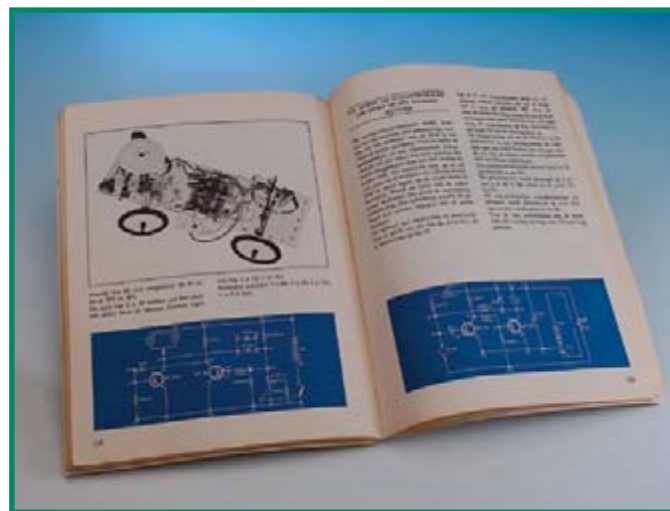
to heavy loads in these designs, and they could head straight for the rubbish bin after the project. Fortunately, these pins and other parts could be obtained as spare parts at that time.

No matter how nice some of the projects looked on paper, they were often not especially solid or robust. As a result, many

hours of painstaking assembly work were often rewarded with a mechanical construction that was quite capable of self-demolition. The ME kits used only clamped connections, and in many cases they were not good enough to accommodate all the mechanical forces.

But the nice thing about the ME system was the enormous variety of structures you could make with them — from mechanical clocks to real water pumping installations. There were also construction projects that used parts from the EE series in a combination of mechanical and electronic engineering. An example is the car in the photo. It stops automatically when it drives on top of a dark surface. Nowadays this is a very simple application with quite simple technology, but it had a certain magic for a small boy!

Philips also tried to get even younger children interested in mechanical engineering and thus create a pool of new customers for the ME kits. Philoform, a construction system that





strongly resembled Lego Technic and could be used together with ME, was introduced in 1968. However, the end of the line for these mechanical construction materials from Eindhoven came in 1970. Incidentally, the first ME kit, the ME1200, had a very strong feature with regard to mechanical engineering. The mechanical parts were housed in a wooden box with a sliding lid. Probably for this reason, they have survived the years relatively intact,

and you can regularly find complete or practically complete kits offered on Ebay and similar auction sites. They usually change hands for around twenty to thirty pounds, naturally depending on their general condition. Just as with all old things, there are collectors who are interested in them, and there are various websites where you can find more information.

(070277-1)

### Web Links

- <http://ee.old.no/mechanics>
- <http://sharon.esrac.ele.tue.nl/~pa0ib/bouwdozn/index.html>
- [www.hansotten.com/philipsme1200.html](http://www.hansotten.com/philipsme1200.html)
- [www.girdersandgears.com/norelco.html](http://www.girdersandgears.com/norelco.html)

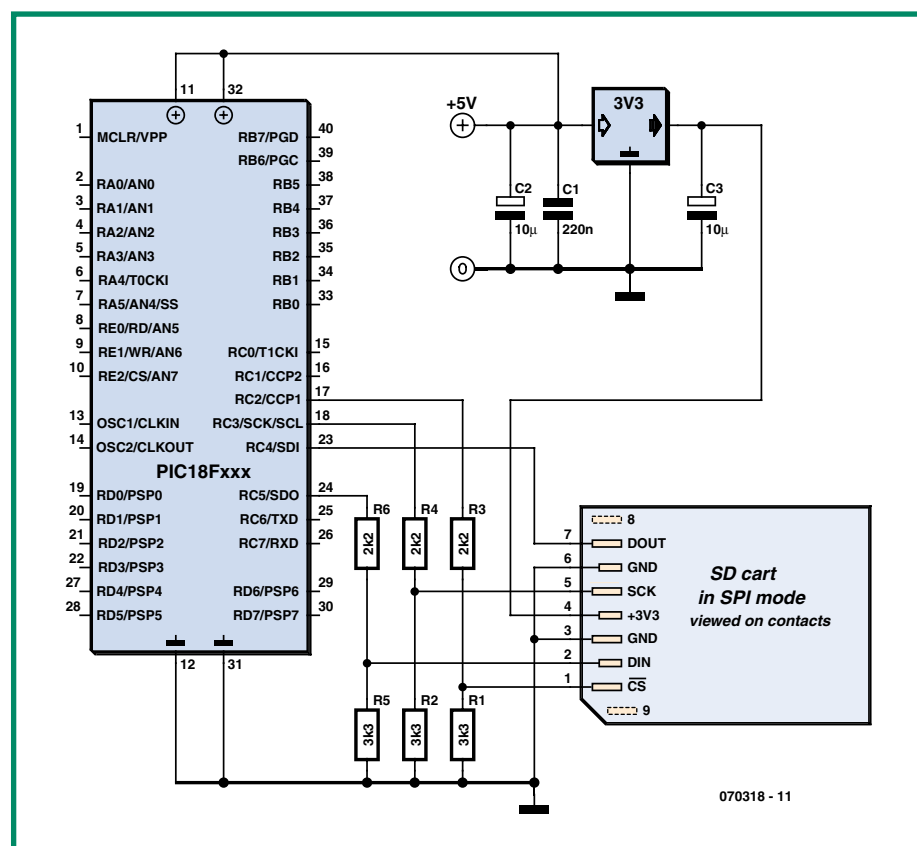
# A Robot with an Elephant's Memory

B. Broussas

Although 'just for fun' robotic applications can usually make do with the few bytes to few hundred bytes of Flash memory that's available in most current microcontrollers, certain more 'serious' or more complex applications do need much greater memory capacities. A mobile robot may hold in its memory a complete mapping of an area in the form of co-ordinates like those provided by a GPS, for example. Alternatively, it may be required to collect a large quantity of data furnished by its sensors. Faced with such a situation, it is of course possible to produce one or more special memory cards using the high-capacity Flash EEPROM packages available on the market today. However, this approach fairly soon comes up against numerous problems. The first is making the necessary PCBs. Most memories of this type are only available in SMD packages, and their close pin spacing makes producing a PCB a tricky job for amateurs, not to mention the difficulty you then have soldering such ICs correctly. The second problem is that as these memories are intended above all for the professional market, they're sometimes very difficult for amateurs to get hold of.

So in this article we're proposing an original solution to this problem, provided your robot is fitted with at least one PIC microcontroller and you don't mind programming it in Basic. You'll agree these are relatively minor constraints, especially when you think that by doing it this way you'll be able to give your robot a gigabyte or even more of memory for just a few tens of pounds!

The memory we've adopted is quite simply the memory sold in the form of SD cards (Secure Digital), originally intended for digital cameras and portable music devices. This memory is very inexpensive today (around £ 7 for 1 GB at the time of



writing), very compact, and unwaveringly reliable, provided you do not exceed the maximum number of write cycles, which is however hundreds of thousands, or even millions, depending on how optimistic the manufacturers are...

The hardware interfacing of such memory with a PIC microcontroller is relatively easy, as the SD-type memories' mode of operation is compatible with the SPI-type synchronous serial interface available in these microcontrollers. The only thing to watch out for is the electrical levels, as these memory cards work on 3.3 V while the PICs in our robots are most often powered from 5 V. The figure suggests a circuit

that can be used with all PIC microcontrollers in the PIC18 family from Microchip. However, the problem is noticeably trickier when it comes to the software for managing these memory cards. Contrary to what we might at first think, these are not just simple EEPROM Flash memories with serial access, but modules that have their own internal intelligence. So it's not possible to read or write directly to these memories as you would do with an ordinary serial access EEPROM like a 24C16, for example.

The dialogue has to respect a precise protocol, as the card only recognizes and responds to a certain number of com-

mands. We also have access to various internal registers with quite specific purposes. Lastly, the location of the data in the card is not just 'any old how', but follows a principle similar to that found on diskettes and hard disks, using in particular a FAT (File Allocation Table) that shows where the data, contained in files as a result, are stored.

Even though it is possible to manage all this information by writing the necessary sub-routines yourself in machine language, this is a long, tedious, and error-prone task.

Very fortunately, if you're interested in using such a memory card in your robot, there is one Basic compiler (for PIC micro-controllers only, at the time of writing) that has a full management library available for SD-type memory cards wired as shown in our figure.

This is the MikroBasic compiler from Mikroelektronika, which you can find on the publisher's website ([www.mikroelektronika.co.yu](http://www.mikroelektronika.co.yu)), with a working demo version allowing you to try it out before buying. Apart from those standard functions available in all Basic compilers for PICs worthy of the name on the market today, it has a full management library for SD-type memory cards (also for Compact

Flash types, but they're not the subject of this article).

We're not going to detail here the sixteen instructions available for manipulating SD cards, especially since you can download the manual for this compiler free of charge from the publisher's website. Just be aware that using this product makes the management of such cards ever so much easier, to say the very least!

So for example, if you have filled a buffer and want to now store it onto the SD card, all you have to do is write:

```
status = Mmc_Write_Sector
(number, buffer) where:
```

- `status` is a variable containing a numeric code returned by the command indicating the outcome of its execution (0 for successful, 1 for an error sending the command, 2 for an error during the writing proper).
- `number` is the number of the SD memory sector we want to write to (we explained above that the data storage is similar to that on a hard disk, and now you can see this in practice).
- `buffer` is the label marking the start of a buffer able to hold up to 512 bytes, which is the size of the SD card sectors. Reading the information stored on the SD

card and transferring it to the robot processor's RAM is just as simple, and amounts to a single line of code:

```
status = Mmc_Read_Sector
(number, buffer) where:
```

- `status` is a variable containing a numeric code indicating the outcome of the execution of the command (0 for success, 1 for failure).
- `number` is the number of the SD memory sector we want to read.
- `buffer` is the label marking the start of a buffer whose size must be at least 512 bytes; this buffer is going to receive the data read from the selected sector on the card.

As you will note, it would be difficult for things to be much easier, even though in this article, by nature only a summary, we have not said anything about the card's FAT management instructions. However, the latter are much less frequently used, once the structure of the card is defined, and a detailed, annotated example about these is given in the compiler manual.

So then, if your robot needs lots of memory, lend it the SD card out of your camera and pay a visit to Mikroelektronika ([www.mikroelektronika.co.yu](http://www.mikroelektronika.co.yu)).

(070318-1)

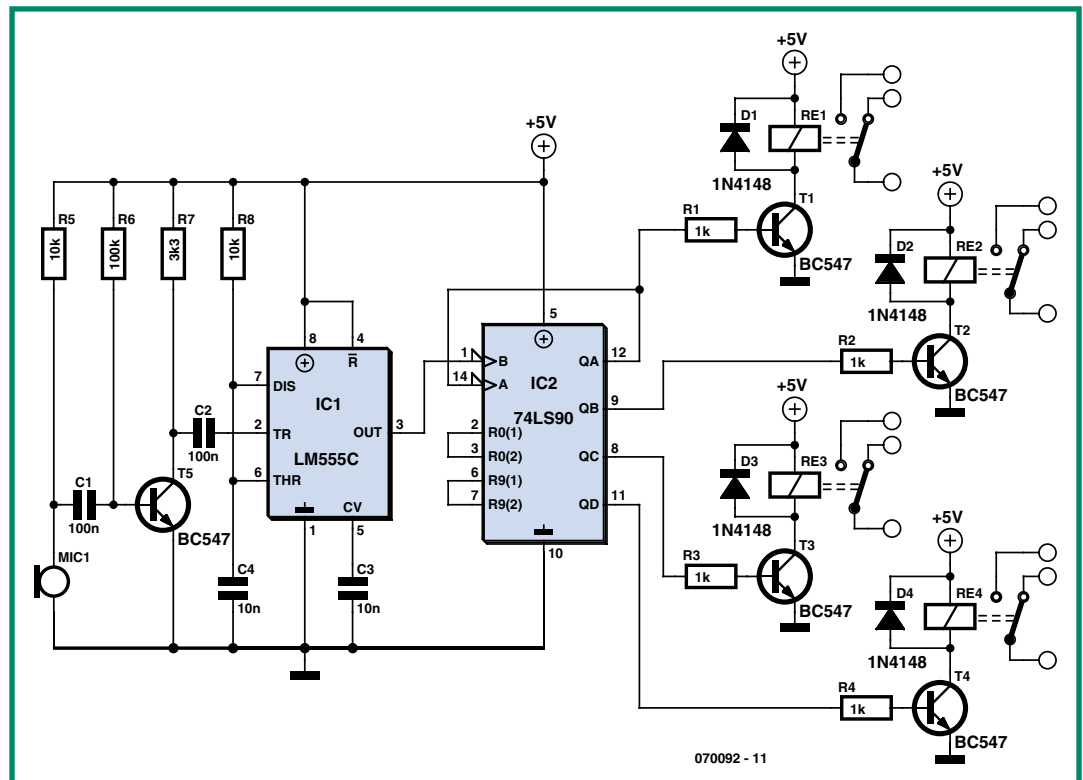
# Clap Controlled Switcher

Raj. K. Gorkhali

The circuit presented allows you to control home electrical appliances like TVs, fans, lighting, etc. by clapping your hands. Four different electrical equipments can be controlled using changeover contacts on relays.

The circuit diagram shows a condenser (electret) microphone M1 connected to the input of preamplifier T5. The sound picked up by the microphone is preamplified and fed to the input (pin 2) of a 555 timer IC set up in monostable configuration. The output of the 555 is connected to the clock input of a 7490 counter.

Whenever a pulse arrives at the clock input of IC2 (pin 14), it produces a 4-bit binary equivalent code at its four outputs. As an exam-



ple, when the first pulse is applied to the 555's TRIG input, the binary coded output on the 7490 will be 0001, for the second pulse the output will be 0010, and so on. For the 15<sup>th</sup> pulse, the output will be 1111. On receiving the next pulse, IC2 automatically cycles back to state 0000. The counter's four outputs control driver transistors T1 through T4. These, in turn, control the four relays RE1 through RE4, and their contacts, the equipment to be controlled. Four rectifier diodes, D1 through D4, are connected across the four relay coils to prevent back-emf surges

upsetting the operation of the circuit. The circuit can be tested in a simple manner. Power the circuit from a regulated 5 V (or 6 V) supply. Temporarily disconnect the CLKA input of the 7490 from the 555 output. Solder a wire on the CLKA input and use it to touch the positive supply rail. Each time a clock pulse is applied in this way, RE1, RE2, RE3 and RE4 should energise or de-energise in one of 16 different configurations. Re-establish the connection between IC1 and IC2 and clap your hands near to the microphone. The relays should respond

as with the clock pulse test. Finally, connect four electrical devices to the relay contacts. The use of a 5 V regulated supply is recommended for this circuit. The relay contacts should be rated for 230 VAC as well as for the maximum current the electrical equipment is likely to draw.

**All relevant electrical safety precautions should be observed when connecting mains powered loads to the relay contacts.**

(070092-1)



## New Lego MINDSTORMS® NXT Motor Block

Antoine Authier

In 2006, the Lego MINDSTORMS® RCX blocks became NXT blocks. They have a quite surprising avant-garde shape — in place of the traditional compact RCX block, for the NeXT generation compact motor the Lego engineers have opted for a 'pistol' profile. This block comprises a motor, a rotary encoder, and a step-down gearbox. Its weight is all of 80 g! The DC motor seems to be a standard type. It is located in the larger-diameter light-grey cylindrical part, under the Lego logo. Powered from 9 V, it draws 60 mA off load with a speed of 170 rpm (360 rpm

and around 3.5 mA for an RCX motor). The internal step-down gearing increases the torque available at the output disc, which is great for power applications. On the other hand, the slower rotation will be seen as a drawback for speed applications. The data provided by the rotary optical encoder allow the NXT unit software to determine the angle of rotation to the nearest degree. Not having dissected the block, we can only surmise how this precision is obtained from the encoding disc that only has 12 slots. The block contains two electronics assemblies: one is the module that conditions the signal from the optical detector, the other protects against poten-

tial overloads. This comprises over-current protection in the form of a resettable Polyswitch® fuse, and a 15 V surge limiter. At one end of the block is the orange drive disc. At the other end is the RJ11 power input and data output connector. It has a key to avoid confusion with a standard telephone cable. Interested readers will find the characteristics and views of the innards of the block on Philo's web pages [1], [2].

(070371-1)



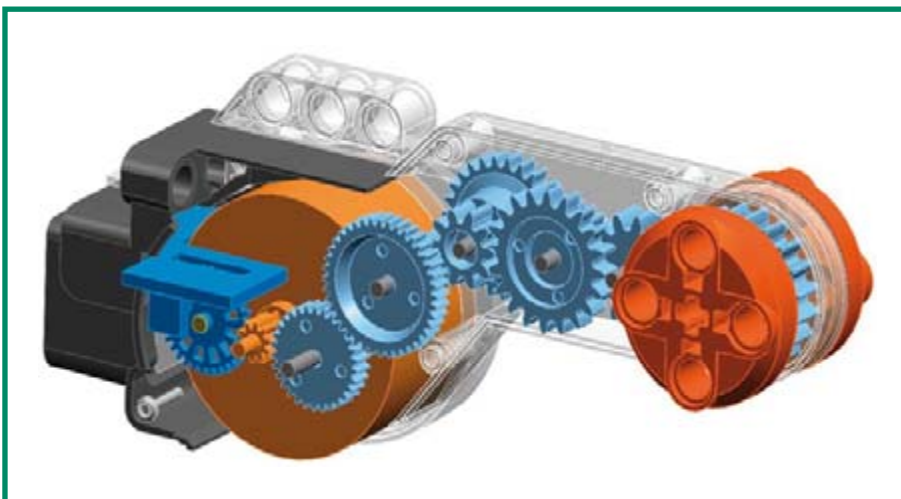
### Web links

(1) Philo's NXT® motor internals:

<http://www.philohome.com/nxtmotor/nxt-motor.htm>

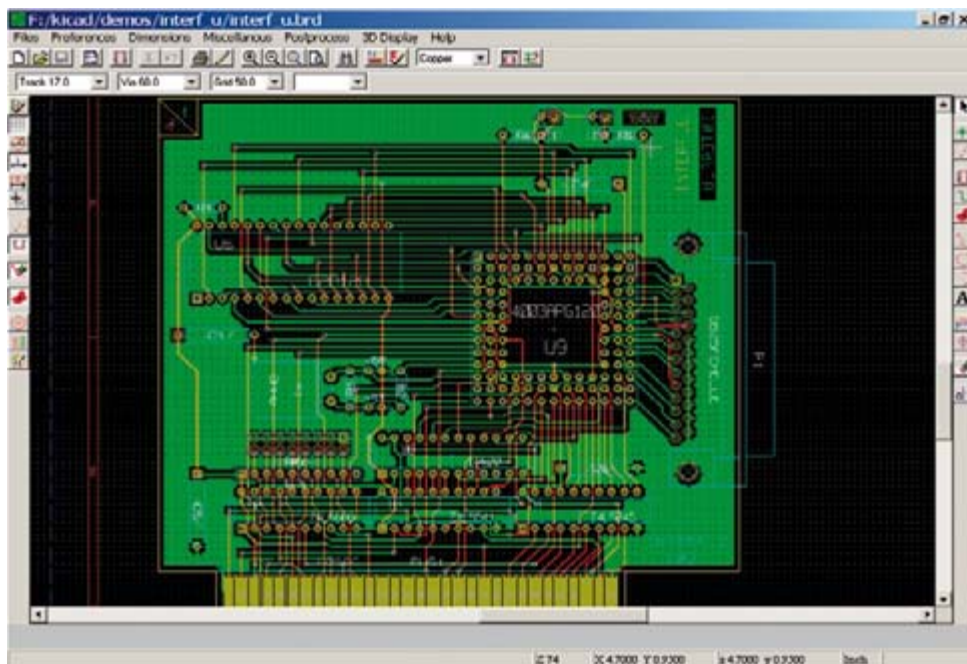
(2) Lego® 9 V Technic Motors characteristics compared:

<http://www.philohome.com/motors/motor-comp.htm>





# KiCad: a high-level tool



The supported OSs are numerous in addition to Windows (2000, XP, and W98 with slight restrictions), the others were delivered, tested and ready to operate with Mandriva and CentOS distributions. It also was just integrated into the Debian distribution, thanks to the efforts of a few dynamic volunteers. Users have also compiled sources in numerous other OSs: Solaris, FreeBSD, etc. Mac OS X remains an exception, because, even if KiCad can be compiled on these machines, its operation is currently still hampered by a bug from the Open Source wxWidgets graphic library, used by KiCad. Let's hope that this problem will be resolved soon: the new version will be distributed as soon as that happens, and the same is true for those that operate in Linux and Windows.

KiCad is available in the following languages: French (original language), English, German, Spanish, Portuguese

(Brazilian), Italian, Slovenian, and Hungarian for the user interface (GUI).

The user manuals are available in the primary four languages. Versions in German, Hungarian, Polish, Korean and Russian are at various stages of translation. Tutorials are also appearing in several languages: French, English, Brazilian. All of these documents have been created by volunteers who believe in Open Source and free software.

If the price of this software package defies all competition, that does not mean that you will have an 'inferior' tool. Even though KiCad is far from being an overly complicated software package like Orcad and Altium, its qualities are nonetheless remarkable: you can judge by the screenshots. The graphic interface, simple and very easy to learn, is intuitive and powerful, thanks to one of the many features that uses a 3-button mouse. The keyboard shortcuts are limited in number but efficient. The various output formats (printer, Postscript, Gerber, hole-making and automatic placement files) are flawless and adapt to all printers, as opposed to some DOS software.

This software package is composed of:

- **KiCad**: project manager, from which one can launch the following programs.
- **EeSchema**: simple or hierarchical schematic capture.
- **CVPCB**: used to link components with their schematic footprints.
- **PCBNEW**: design of printed circuits.
- **Gerbview**: display of Gerber files.

## Installation

KiCad is available on the DVD that accompanied *Elektor Electronics*, November 2006. More recent versions can be downloaded from the websites devoted to KiCad (see links 1 and 2). At the time this article was written, the current version is dated August 28, 2006. Archives in .tgz or .zip format are about 70 MBytes. To install the software package, you just need to decompress the

## Electronic CAD available to all

Rémy Halvick

**For the month of November, 2005, we had electronic CAD as the theme of Elektor magazine. The issue came with a free DVD packed with software, most of it operating as a Windows demo version. One of the programs stood out due to several unique features. Actually, KiCad is a software package distributed for free under a GPL license, operating in Linux, Windows and Mac OS X environments. As an added treat, this marvel is available in a remarkable number of languages!**

Things have greatly changed since the time when electronics hobbyists (professional and/or amateur), peered over schematic diagrams for many hours, drawing circuits with pencil and paper, then produced it all over again for a 'clean' version. Businesses were the first to have the means to utilise CAD software such as Orcad or Protel, on powerful systems with staff especially trained for this task. For a long time, this was too expensive for amateurs.

Electronics fans today are much more spoiled — they can utilise programs that use little resources, at reasonable prices; some even come as 'light' version for free, but with restrictions that seriously limit would-be users.

KiCad was of course included on the free 'Kaleidoscope' DVD. It was developed by a professor/researcher, Jean-Pierre Charras, from the Joseph Fourier University in Grenoble, France, in order to learn programming in C++, as he claims. The first rough drafts were begun in 1992 in DOS, the most recent versions are available as downloads on the university website (see the links at the end of this article).

archive in C:\Program Files\ or /usr/local for Linux users, and to place a link pointing to the executable KiCad file in the sub directory X:\kicad\winexe or /usr/local/kicad/linux. That is the extent of it — no further torture will be inflicted on your precious PC.

## KiCad

The KiCad project manager (**Figure 1**) allows you to create or select a project; meaning mainly schematics and a printed circuit. In this way, you also have access to the language selection options for the graphic interface and online help.

## EeSchema

EeSchema (**Figure 2**) lets you input a simple or tree (hierarchical) structure. The screenshot is used to get an idea of the simplicity of the interface which does not, however, sacrifice functionalities.

The menu toolbar only has three sections: File, Preferences, and Help. In addition to the traditional open/save/print options, the File menu allows you to generate drawing files in PostScript, HPGL, and SVG formats.

In Preferences, you can select which libraries to be used (which will appear when adding components), as well as various options: colour, display and grid pitch (scale), orientation and incremental values for repetitive tasks; all of these options may be modified, of course, if the need should arise, although the default options satisfy most needs. The help menu is very standard.

Three icon bars give you access to most of the tools which you will need. The one on the left lets you manage the graphic 'look': grid display, its pitch, measurement units (millimetres or inches), cursor shape, orientation of the lines (by 45° increments or any orientation); the icon (A) lets you display the hidden power connections to the parts.

*Due to their small size, we propose icons in a magnified version in Figure 8 with the reference letter. The icon H is actually a double icon.*

The upper bar has various tools: file manipulation (open, save); button (B) allows you to choose the page format (A4 to A0 and A to E, as well as a custom format defined by the user) and to fill in the various sections of the of the schematic. The next version of KiCad, which apparently is about to be released, will add an Undo/Redo function to **EeSchema**. The next two icons relate to the **Libedit** component display /editor (**Figure 3**); in fact, you can create any special symbol that you might need for your schematic. The **CVPCB** and **PCBnew** icons follow after the traditional editing tools (cut, copy, paste) and print; we will examine their role a little later. The 4 following tools deal with display: + and - zoom, drawing refresh and auto zoom which lets you have a better look by reframing the entire schematic. These functions are also accessible from the F1 to F4 function keys. When the diagram becomes cluttered, sometimes it is difficult to find R59 or U12; you can then use the search tool by clicking on (C).

The following icon (D) allows you to generate a netlist in different formats; you can even have yours by creating a plug-in! Before arriving at this point, you would have taken care to number the components, thanks to the automatic annotation tool (E).

The next-to-the-last tool in the top toolbar (F) is very useful: it deals with verifying that the electrical rules are respected or DRC (Design Rules Check). The principle is the following: each component pin is defined while it is being drawn as input, output, open-collector, 3-state, etc. The DRC tool will carry out various plausibility checks: output connected to the power supply, unconnected gate input and others; you can define the checks as well as their result: error, warning or no error in the options tab. This is used to avoid gross errors and forgotten connections.

The last icon is for generating the list of components (BOM = bill of material), which will help you with your shopping, especially if you export it to a spreadsheet in order to optimise supply sources.

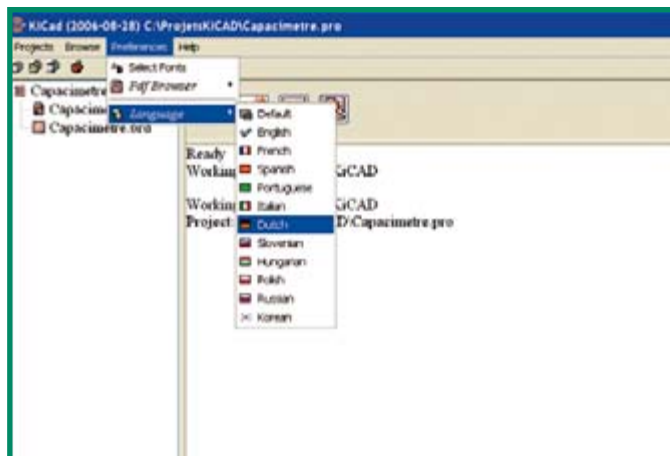


Figure 1. KiCad is project-based, just like similar products on the market.

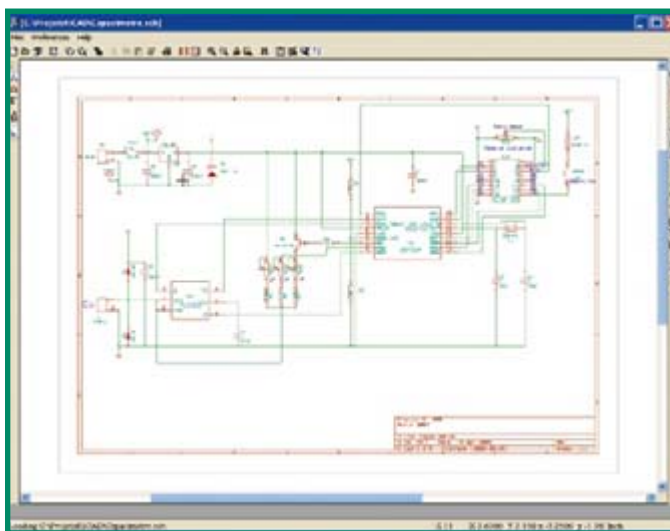


Figure 2. EeSchema: easy schematic capture.

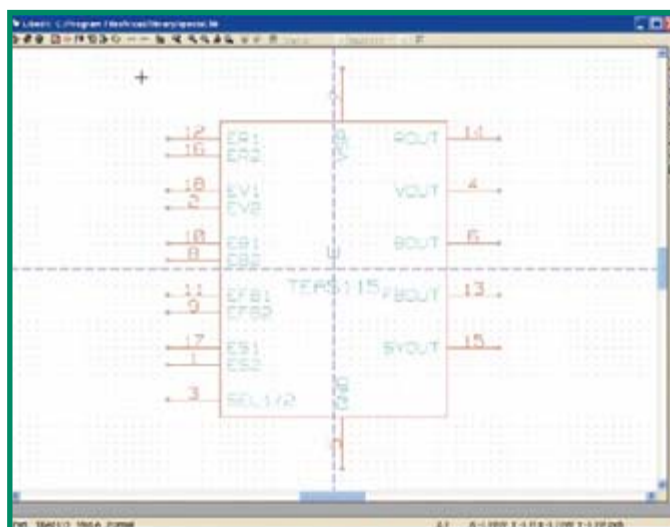


Figure 3. If you haven't found the component you need on the Internet, there is nothing to stop you from designing it with Libedit!



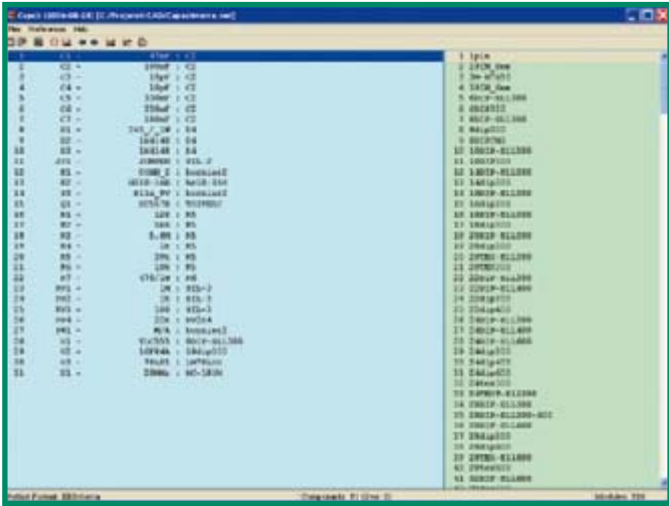


Figure 4. CVPCB: choice of component case.

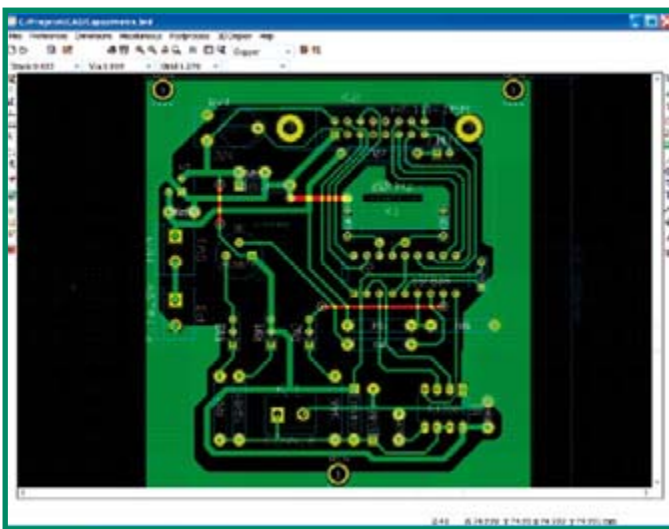


Figure 5. Design PCBNew printed circuits.

The right icon bar groups the different drawing tools: adding component (the gate), connection by wire or bus, labels, commentaries and other embellishments that will improve the presentation of your schematic.

The rest of the required commands are accessible from the context-

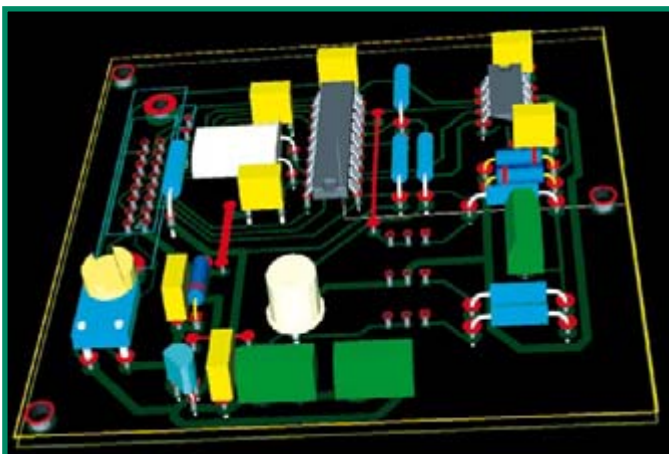


Figure 6. 3D display of the capacitance meter as described in *Elektor*.

tual menus, with one right mouse click. That is one of the strong points of Kicad, which, by proposing the tools at the time they are needed, makes it possible to preserve a clear and easy to grasp interface. Nothing like those heavyweights with their dreadfully cluttered user interface, almost impossible to master by anyone who is not a specialist. These contextual commands are dependent on the part you choose to click on, using the right button. For example, clicking on a component opens the following possibilities: move, orient, edit, copy, or delete the component, front/rear zoom, auto function, recalculate the drawing, select the scale. The menu is adapted depending on whether you clicked on a component, a wire, a text field, etc.

The mouse also makes it possible to display a zone that you will select by clicking with the wheel: without a doubt, efficient and practical!

The status bar, at the bottom of the window, gives you the following information: zoom factor, absolute and relative coordinate of the cursor and measurement units (inches or millimetres).

## CVPCB

CVPCB, accessible from EeSchema or KiCad, makes it possible to link a case to each component from the net list that you will have created in EeSchema (Figure 4). There, also, in addition to the print libraries that come as standard (through-hole or SMC components), you can download many others on the Internet. If you use KiCad intensively, you can eventually use the automatic association system component case that makes it possible to automate this task.

## PCBNew

You gain access to PCBNew from the KiCad project manager (recommended), or directly using the icon (G) (Figure 5). This printed circuit design software is made in the image of EESchema: simple, easy to get used to and easy to use; that does not mean its performance is lagging, here are some examples: 16 copper layers, 12 technical layers (lithography, resist coating, dimensioning, etc.) components, through-hole or SMC, work done to one/ten-thousandth of an inch, dynamic rats-nest, Design Rules Check, ground plane, and it has a very capable high-performance internal router that can operate in a single-sided layer! What more can an amateur ask for? And a 3D view of the board surface just for the visual pleasure. It is shown in Figure 6!

But let us begin by the beginning: PCBNew. The general philosophy of this graphic interface is the same as the one for EeSchema: simple, easy approach, but also as efficient, thanks to an intensified utilisation of the mouse and the contextual menus, and the defining of the two operating modes: placement (H left) and routing (H right). These two modes will affect the contextual menus that will be shown.

We will not insult you by reviewing file menus and preferences, except to remind you of the output formats: PostScript, HPGL, Gerber 274X, and Excellon, in addition to your favorite printer. The 'Dimensions' menu is used to define... the dimensions by default of the traces, vias, pads and texts. The 'Miscellaneous' section groups IC detailing accessories.

The post processors enable the automatic placement file generation of components and hole-making files. This gives you the possibility of having your IC made by a professional, by sending him the necessary files. The '3D' and 'Help' menus speak for themselves. Under the different menu bars you will find scrolling lists that allow you to easily change the width of the trace, dimensions of the vias, grid pitch number of the layer and zoom.

The icons located on the left of the window are related mainly to what is represented on the screen: display of the polar coordinates in the status bar, measurement units, shape of the cursor, display of the rats-nest (representation by segments of the connections to be routed),



automatic erasing of traces that you have re-routed, display of pads and traces in full lines or in contours, in high-contrast display. The upper icon toolbar, just like in EESchema, groups the file commands and the selection of the sheet format. The next icon allows you to access the module editor (or footprints) of the components in the unlikely case that the libraries supplied and those available on the Internet are insufficient. Its operation very much resembles the one of the LibEdit component editor, which means you should feel more comfortable with it.

We have no specific comment on the following print and tracing icons except that they resemble the ones in EESchema.

**(D)** is the starting point for designing a printed circuit: reading the netlist. Your components are found 'in bulk' next to your sheet. To spread out the components in order to be able to then gather them, go to placement mode **(I)**. With one right click, do global move and place → Move all of the modules: and all of your components will be carefully aligned. If you prefer, after having defined the contours of your printed circuit (select the PCB contour layer and define a closed figure that pleases you), then Global move and place → Autoplace all of the modules. It will do half of your work by optimising the length of the connections. Any intermediary state is conceivable with the interactive auto placement options.

It is already time to move from placement mode to routing mode with the icon **(J)**. With one right click, do Global autorouting → Select layer coupling. It usual to choose a one-sided circuit board; in other words, the top layer will be in copper, just like the lower layer. The autorouting feature (global autorouting → Autoroute all modules), will greatly lessen the work, if it does not route your entire board. All that remains is to finish in manual mode, or to move a few components and re-route the overall project. Manual routing is accomplished, of course, with the mouse, and you will quickly notice that PCBNew knows how to place your traces in a well-disciplined fashion, without laboriously defining the smallest change in orientation. Displaying the rats' nest **(K)**, (see **Figure 7**) enables quick and reliable work. If straps are required, they will be shown on the traces on the component layer (in red on Figure 6). Once the routing has been finished, you can add centring targets, dimensioning, and any other graphics such as a logo, copyright symbol, etc.

A wise precaution consists of carrying out a DRC check, in order to ensure that no routing error or short-circuit still exists. All that is left is for you to do is to print or plot, to start with on paper, in order to determine the factor of the precise scaling, setting it to the scale requested by your printer. You can then print a transparent to isolate your IC or to generate the files requested by your supplier. Of course, professionals may demand much more from KiCad; a certain number of companies are already using it around the world.

## In conclusion

KiCad is a real windfall for the creative electronics fans among you. It thus becomes possible to create, exchange and modify schematics and printed circuit board designs at will. No more searching for a jack connector with unobtainable placement or 'butchering' a circuit in order to adapt it ever so slightly. With KiCad, you have total freedom, especially if, as we hope, electronic magazines publish more or less finalised versions of schematics and/or PCBs of proposed projects on the Internet, rather than a fixed PDF. That would signify that anyone and everyone can adapt, convert or even transform them at will (maybe with a simulator in the future?.. But hush! nothing has been done on that yet!). Better yet, be assured that in case of problems, help will be available to you rapidly on the user's forum (link [3]).

(060373-1)

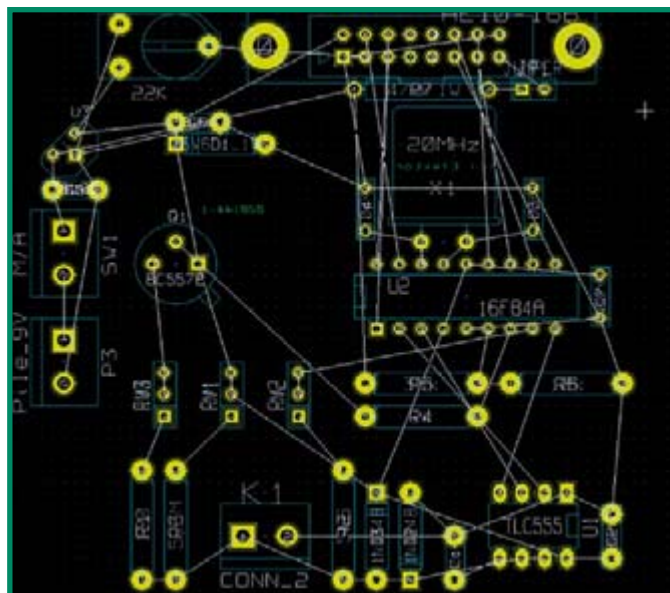


Figure 7. The rats' nest in white lines.

## Internet links

### (1) Kicad 1 homepage

[www.lis.inpg.fr/realise\\_au\\_lis/kicad/index.html](http://www.lis.inpg.fr/realise_au_lis/kicad/index.html)

### (2) Kicad 2 hmepage

[iut-tice.ujf-grenoble.fr/kicad/index.html](http://iut-tice.ujf-grenoble.fr/kicad/index.html)

### (3) An active group!

<http://groups.yahoo.com/group/kicad-users/>

### (4) Libraries, user guides

<http://www.kicadlib.org/>

### (5) KiCADWiki

[http://kicad.bokeoa.com/wiki/index.php/Main\\_Page](http://kicad.bokeoa.com/wiki/index.php/Main_Page)

### (6) Utilities

<http://www.rohrbacher.net/kicad/quicklib.php>

### (7) Goodies from Brazil: footprints, utilities, etc.

<http://www.reniemarquet.cjb.net/kicad.htm>

### (8) KiCAD, from the professional point of view

<http://xtronics.com/reference/kicad.html>

### (9) Developers

<http://developer.berlios.de/projects/kicad>

### (10) Kicad hosted in a free world

<http://kicad.sourceforge.net/en/index.shtml>

### (11) Kicad, from the Russian point of view.

<http://ru.wikipedia.org/wiki/Kicad>

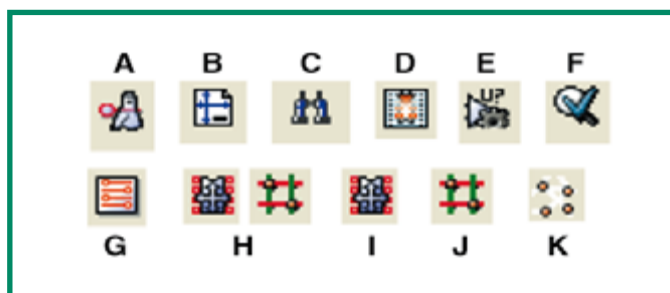


Figure 8. The main icons on a large scale and in low resolution (the originals are a standard size, 16 x 16 pixels).

# Football with Robots



In this Summer Circuits issue with robots as its theme, we can hardly afford to overlook the RoboCup. The RoboCup is a collective international project with the objective of fostering research in artificial intelligence, robotics and related fields. Football has been selected as the basis, and the ultimate objective is to have a team of fully autonomous humanoid robots in 2050 that can defeat the world football champions.

Four leagues are open to participants: the simulation league, the small and mid-sized league, the four-legged league, and the humanoid league. A team from Philips is participating in the mid-sized league (MSL).

The robots in the MSL must be DIY constructions and must be autonomous. This means that all their intelligence must be on-board, and nobody is allowed to control the robot from the sideline.

The Philips robots are equipped with the following components to enable them to play football:

- two Vision cameras (one fixed camera at the front and one omnidirectional camera)
- a PC
- a PLC controller
- a wireless network interface
- four wheel units
- four sensors for detecting the white lines
- a ball handler
- a shooting mechanism
- batteries



The Vision cameras are the eyes of the robot. The robot filters the incoming images, using colour to recognise the ball, the goals, the corner posts and other robots. The camera at the front helps with precise orientation. The robot uses the omnidirectional camera, which is oriented upward toward a special mirror, to keep an eye on its surroundings so it can see where it is and what is happening around it.

The PLC controller monitors the status of



the robot: are the batteries still OK, is the shooting mechanism working properly, etc. The robots on the team communicate with each other via the wireless network, so they don't try to steal the ball from each other.

The built-in PC runs on Linux and processes all the data used by the robot to determine where it is and what its best plan is for getting the ball and scoring. The white-line sensors help the robot determine where it is.

The PC drives the four wheels of the robot via a special module. Each wheel can be turned and driven independently. This means that the robot can make a pirouette



while heading toward the ball.

The purpose of the ball handler is to keep the ball as close to the robot as possible without actually grasping it. Grasping the ball is against the rules. The ball handler lets the robot move sideways and backwards without losing the ball. It also helps the robot shoot the ball straight forward. It holds the ball directly in front of the shooting mechanism, which can 'kick' the ball at approximately 30 km/hr. The energy for the shot comes from a spring, which can be wound up in a couple of seconds and is released when the software gives the order to shoot.

A total of eight robots participate in each game (four on each side). They play on a field with dimensions of 8 by 12 metres, and the game begins with looking for the ball after the referee gives the starting signal. After the ball is found, the trick is to travel toward the opponent's goal in order to score, and of course to try to avoid losing the ball to an opposing player. Naturally, the robots also try to steal the ball from the robot of the opposing team before it can score a goal.

As the robots operate entirely autonomously, their behaviour must be programmed in advance. A collision is punished by a yellow card, and a repeat offence yields a red card, which means the robot has to leave the field. The robot shoots at the goal as soon as the goal opening is large enough. However, winning takes more than just firing off hard shots; speed and tactics are at least equally important.

(070359-1)





ANDRE LAMOTHE'S  
**XGAMESTATION**  
LEARN STEP-BY-STEP HOW TO BUILD  
AND DESIGN YOUR OWN VIDEO GAME CONSOLE!

Design inspired by the  
Atari 800/2600, Sinclair ZX Spectrum,  
Apple II & Commodore 64!

Parallax SX Key Compatible!

**INCLUDES:**

- Tools, Demos & Utilities!
- Great for Hobbyists AND Students!
- The Fun Way to Learn Embedded Systems!
- Fully Assembled XGS Micro Edition Unit!
- Complete Software Development Kit!
- eBook on Designing the XGS Console!
- Cables and Power Supply included!

**WWW.XGAMESTATION.COM**  
Ph: 925.736-2098 | SUPPORT@NURVE.NET

NURVE NETWORKS LLC  
403 Camino Arroyo West  
DANVILLE, CA 94526 USA

**PCBs in  
24 hours  
from £30.00**

**www.PCBTrain.com**

**Tel: 01635 40347** Newbury Electronics Ltd  
Fax: 01635 36143 Faraday Road Newbury Berks RG14 2AD  
e-mail: circuits@newbury.com.co.uk www.newburyelectronics.co.uk

**ISO 9002/UK**

**PCB-POOL®**  
SERVICING YOUR COMPLETE PROTOTYPE NEEDS

**1 EUROCARD**  
(160 x 100 mm)  
+ Tooling  
+ Photoplots  
+ VAT  
**€49**

Price example  
Any size and contour possible!

**Optional:**

- Soldermask
- Fast-turnaround
- Silkscreen
- 4-Layer Multilayer
- 6-Layer Multilayer

**DOWNLOAD OUR  
FREE LAYOUT SOFTWARE!**

**Freephone 0800-3898560**  
Tel.: +353 (0)61 701170  
Fax: +353 (0)61 701165  
pcb-pool@beta-layout.com

**PCB-POOL.COM**

*Simply send your files  
and order ONLINE:*

**ElmScan 5 USB Scan Tool**

All Major Credit Cards Accepted!

- Diagnose any car with EOBD
- Based on the ELM327 IC
- Supports all EOBD and OBD-II protocols
  - ✓ ISO15765-4 (CAN)
  - ✓ ISO14230-4 (Keyword 2000)
  - ✓ ISO9141-2
  - ✓ J1850 VPW
  - ✓ J1850 PWM
- 9600 or 38400 baud
- Supported by more software than any other OBD interface
- Inexpensive
- Full Speed USB 1.1
- Automatic protocol detection
- Package includes cables, software CD, and Quick Start Guide
- Buy from your local UK distributors!

**www.ElmScan5.com/elektor**

**Your price conscious PCB supplier**

**EURO CIRCUITS**

Online price calculation  
Online ordering  
Online order tracking  
Online 24/24H and 7/7D

Interested? Contact us: +44 2088 167 005  
E-mail: euro@eurocircuits.com  
**www.eurocircuits.com**

- Verified**
  - pooling for standard boards
  - up to 6 layers
  - from 1 to 1000 pieces
  - from 3 working days onwards
- A la carte**
  - pooling with more options
  - up to 8 layers
  - from 1 to 1000 pieces
  - from 3 working days onwards
- On demand**
  - your board, our challenge
  - up to 16 layers
  - from 1 piece onwards
  - from 3 working days onwards



# Alphanumski

Puzzle creator: Géry Szcepanski

In this bumper double issue of **Elektor**, staunch supporters of our monthly **Hexadoku** puzzle (their numbers growing steadily) find themselves confronted with a horrific attack on patience, intuition and intelligence, by a monster puzzle baptised 'Alphanumski' by its creator.



The method of solving the 36x36 cell Alphanumski puzzle shown here is basically the same as for a 9x9 cell Sudoku, a 16x16 cell Hexadoku or even last year's 25x25 cell Alphadoku.

This month we're using all letters of the alphabet (A through Z) and all numerals (0 through 9) to be entered in the cells that make up lines (1x36 horizontally), columns (1x36 vertically) and boxes (6x6).

In Alphanumski, **all** letters of the alphabet (A through Z) and **all** numerals 0 through 9 should occur **only once** in every line (1x36), every column (1x36) and every box (6x6; identified by red outlines and a background colour).

A number of clues are given in the puzzle and these represent the start situation.

All correct entries received for the puzzle go into a draw for a main prize and three lesser prizes. All you need to do is send us the **combination of seven letters and numerals in the grey boxes**. The puzzle is also available as a **free download** from our website..

(070151-1)

		A			R		3	D						1	Z	W	
D	6	L		3	T	1	Q	5				G	J	A		2	
O		H			E	G		7		0			K	U	Y		
7		0			S	L	P	6	H	Y	2		N		9	I	Q
G	Q	N	B	J		I	E		8		V	4	5	D		3	L
	C			Y	I	Z	S	K	N			E			X	7	R
	1					B	J	X	T			S		O			V
5	I		L	M	Q			A	D	O	8		U	2	Y		9
	V	8	C	2		4	5		M	G	R				0	B	6
0		E	T	W	B	Q	9			N		X	F	3			1
J		3		O	G	W					E	I		L		T	
K	A	6			U	V	I			P		D	J	N		G	8
	W	1	5		F	J	G	0			P	9	4		I	2	
	3	V		7	8		O		W	H	N			P	L	U	S
9			N			6	D	U	B			5			M	E	Y
		K		U	M	C	X		R	4		A	D		G	N	3
P	2		O	D		E	Y			S		Q	K		T		
6	Y		Q	R	X	9		2	K	7	A			F	O		H
	0	S	K	5	D		2		Y	R		G	H		V		I
T	M		A	I	N	U	1	C		W	Z	B			8		
Y		9	F	L	2	0				Q		Z	T	E			D
1	U		E		P		N		O	L		F		4	2	5	X
	G	B	6			H	T			X	3	L	0			P	A
	Z	4		8		D	A	7	6	V		N			J	Q	U
8		2		4	V	Y	O	N		K	J		Z	A	Q	W	P
	B			S		T	L	V	U			Y	2		4	H	
Q		F	3	6					2	M	D		B		E	1	T
	7	I	W	P	A	R	F		X		S	6	9	U			0
	D			0	C	3	6	P		I		K	V		F	L	J
U	J	Y	H		K	5	Z		A			I				D	O
M	F		8	H	Y		R	1	J	3			7	W	D		G
		R	J		3		M	B		D	Y	P	L		6		E
B	S	Q		X	W	P		O			T	1		0	3		
I		5		T	6	S	7				W	2	Q	8			
C			2	A			K	L		6	Q	U			N	R	4
L				1	7	2		H	V	8	9	O	A	I	B	X	

# A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 1 2 3 4 5 6 7 8 9

## brain wrecker for the holidays

	Y	M		7		I		J	G	6	X		F	Q		L	8
C			B	8					S	O		I	E	O		N	
D	1		J	I	F	P	L				R		X			9	S
		A	O	U	T	B	3	W	Z	E		G	J	1	R		
O	S		9	X	W	7	T	2	A	H		Y				C	
L			2	4		9		D	O	5		U	P	H			B
G	E		8			Q	5	L	I	O			9	7	6		R
W	F	T	R	N						P	7	X		E	O		
	I	L		3	Y	N	A	F				O	K	D		1	
		J	Z	C	V	H	Y		4	D	K		U	2	I	G	O
	D	S		2					M			B		A			
O	B			9		2	1	X	E	Z		M			L		C
		N				M	H	E	X	A	D	O	K	U		V	L
T	G	2	Y	D	K	0	B	5	9	1	6			J	C	Z	
	X			H	O		7			2	P	A	R		S	F	
9		V		S		T	Q	O	W			0	1	6		2	7
A	0		5	4	U		J	G				B	N		X		
Z	L		M	1	E		C	4		S	N		G	I	P	8	5
E		P				X	W		L		8	N	C	4		T	6
			H	J	L	5			Y	9	G			D		E	2
			O		A	3			P		7	4	W		8	U	J
M	W		C	Y	D	Z	V		T	J		H			G		
R	8	F	7	K	Z	O			2	U		Q	M	5	9	1	
S	2	5	T	G		E	K	0		M	C			L	X	P	
B	3		S		X			U	6			C	I		F		G
1				W	6			7	3	F	0	E	8		A		
	9	Y	4	0	I	R	X	H	N	K	A	Z	S		J	5	
	O	H								Q		2		Y		3	K
U	A		2	5	Q	W	S			Y		1			B		N
		7	N		R		P	9		G			4	W	M	6	0
2		X	V		5	S		B	P		L		A		Q	0	
H		8	I	Z	0	F	N	1			T			C	5	U	X
N	R	9			M	A	2	Y	7		H		L		8		D
				L	G					K	X	J		N		O	3
3	J	W		B			0		D	V		S	H		T	M	
	K	U	D	T	S	C	M	6	5			J	W	G	E	R	Y

## Solve Alphanumski and win!

Correct solutions received enter a prize draw for an

**E-blocks Starter Kit Professional**  
worth **£248.55**

and three  
**Elektor Electronics**  
**SHOP Vouchers**  
worth **£35.00** each.

We believe these prizes  
should encourage all our  
readers to participate!



## Participate!

Please send your solution (the numbers in the grey boxes) by email to:

**editor@elektor-electronics.co.uk**

**Subject: Alphanumski 07-2007 (please copy exactly).**

Alternatively, by fax or post to:

**Elektor Electronics Hexadoku**

**Regus Brentford**

**1000 Great West Road**

**Brentford TW8 9HH**

**United Kingdom.**

**Fax (+44)(0)208 2614447**

The closing date is **1 September 2007.**

The competition is not open to employees of Segment b.v., its business partners and/or associated publishing houses.

## Prize winners

The solution of the May 2007 Hexadoku is: **B789E**.

The **E-blocks Starter Kit Professional** goes to:  
Michele Casartelli (I).

An **Elektor SHOP voucher** worth £35.00 goes to:

David Meiklejohn (AUS) ;

Tomas Bakke (N) and

Franz Klein (UK).

Congratulations everybody!

# ELEKTOR SHOWCASE

To book your showcase space contact Huson International Media

Tel. 0044 (0) 1932 564999 Fax 0044 (0) 1932 564998

## ATC SEMITEC LTD

[www.atcsemitec.co.uk](http://www.atcsemitec.co.uk)

Thermal and current-sensitive components for temperature control and circuit protection;

- NTC Thermistors
- Thermostats
- Thermal Fuses
- Current Diodes
- Re-settable Fuses
- Temperature Sensors

Call today for free samples and pricing

Tel: 01606 871680 Fax: 01606 872938

## AVIT RESEARCH

[www.avitresearch.co.uk](http://www.avitresearch.co.uk)

USB has never been so simple...

with our USB to Microcontroller Interface cable. Appears just like a serial port to both PC and Microcontroller, for really easy USB connection to your projects, or replacement of existing RS232 interfaces.



See our webpage for more details. From £15.00.

## BAEC

<http://baec.tripod.com>

"The British Amateur Electronics Club Archive Website. Archiving extracts from 140+ Newsletters from 1966-2002. Currently have interesting and useful selected articles from 12 Newsletters. Also a section about built electronics projects with schematics and photos. Plus useful info., downloads and links. NO ADVERTS!"



## BETA LAYOUT

[www.pcb-pool.com](http://www.pcb-pool.com)

Beta layout Ltd Award-winning site in both English and German offers prototype PCBs at a fraction of the cost of the usual manufacturer's prices.



## DECIBIT CO.LTD.

[www.decibit.com](http://www.decibit.com)

Smallest 2.4 GHz ISM band MCU embedded transceiver modules. Complete tiny solution, ready to transmit RF data by only attaching as example a CR2032 coin cell to it.



## EasyDAQ

[www.easydaq.biz](http://www.easydaq.biz)

We design & supply low cost USB/RS232 based data acquisition, automation & control products:

- USB connected & powered, 8 opto isolated voltage inputs chans, 4DIO & 4 onboard relays capable of switching 240V@10A. With Labview, VC & VB examples - £60



## EASYSYNC

<http://www.easysync.co.uk>

EasySync Ltd sells a wide range of single and multi-port USB to RS232/RS422 and RS485 converters at competitive prices.



## ELNEC

[www.elnec.com](http://www.elnec.com)

- device programmer manufacturer
- selling through contracted distributors all over the world
- universal and dedicated device programmers
- excellent support and after sale support
- free SW updates
- reliable HW
- once a months new SW release
- three years warranty for most programmers



## FIRST TECHNOLOGY TRANSFER LTD.

<http://www.ftt.co.uk/PICProTrng.html>

Microchip Professional C and Assembly Programming Courses. The future is embedded.

Microchip Consultant / Training Partner developed courses:

- Distance learning / instructor led
- Assembly / C-Programming of PIC16, PIC18, PIC24, dsPIC microcontrollers
- Foundation / Intermediate



## FUTURE TECHNOLOGY DEVICES

<http://www.ftdichip.com>

FTDI designs and sells USB-UART and USB-FIFO interface i.c.'s.

Complete with PC drivers, these devices simplify the task of designing or upgrading peripherals to USB



## FUTURLEC

<http://www.futurlec.com>

Save up to 60% on

- Electronic Components
  - Microcontrollers, PIC, Atmel
  - Development Boards, Programmers
- Huge range of products available on-line for immediate delivery, at very competitive prices.



## LONDON ELECTRONICS COLLEGE

<http://www.lec.org.uk>

Vocational training and education for national qualifications in Electronics Engineering and Information Technology (BTEC First National, Higher National NVQs, GCSEs and Advanced Qualifications). Also Technical Management and Languages.

## MODular ElectroniCS

[www.modetron.com](http://www.modetron.com)

- Plug and Program
- FREE application s/w
- Hobbyist ease-of-use
- Professional finish with enclosure and LEXAN faceplate
- We will design and brand your custom application
- Growing range of PSU's, i/o modules, displays and microcontrollers



## MQP ELECTRONICS

[www.mqp.com](http://www.mqp.com)

- Low cost USB Bus Analysers
- High, Full or Low speed captures
- Graphical analysis and filtering
- Automatic speed detection
- Bus powered from high speed PC
- Capture buttons and feature connector
- Optional analysis classes



## NEW WAVE CONCEPTS

[www.new-wave-concepts.com](http://www.new-wave-concepts.com)

Software for Hobbyists:

- Livewire - circuit simulation software, only £34.99
- PCB Wizard - PCB design software, only £34.99
- Circuit Wizard - circuit, PCB and breadboard design software, only £59.99

Available from all Maplin Electronics stores and [www.maplin.co.uk](http://www.maplin.co.uk)





# products and services directory

## PCB WORLD

<http://www.pcbworld.org.uk>

World-class site: Your magazine project or prototype PCB from the artwork of your choice for less. Call Lee on 07946 846159 for details. Prompt service.

## ROBOT ELECTRONICS

<http://www.robot-electronics.co.uk>

Advanced Sensors and Electronics for Robotics

- Ultrasonic Range Finders
- Compass modules
- Infra-Red Thermal sensors
- Motor Controllers
- Vision Systems
- Wireless Telemetry Links
- Embedded Controllers



## SOURCEBOOST TECHNOLOGIES

<http://www.sourceboost.com>

Next generation C compiler and development products at highly affordable prices:

- C, C++, and Basic compilers for PIC12, PIC16, PIC18
- Modern IDE, with PIC simulator, source level debugger and virtual devices.
- RTOS for PICmicro.
- PIC based controller and Development boards.
- Download and try for Free from <http://www.sourceboost.com>



## SYTRONIC TECHNOLOGY LTD

[www.m2mtelemetry.com](http://www.m2mtelemetry.com)

Supplier of wireless modules and accessories for remote monitoring M2M applications.

- GSM/GPRS TCP/IP modules
- Embedded GSM/GPRS modem
- Development Kits
- GPS modules
- GSM/GPRS antennas
- Adapter cables



Online ordering facilities.  
Tel: 01728 685802

## COMPONENTBIN.COM

[www.componentbin.com](http://www.componentbin.com)

Kickstart your development with modules and parts from componentbin.com

- ARM7 modules
  - Ethernet modules
  - Superb Graphic LCD displays (all with example software)
- and much much more...  
Online ordering and great prices!



## ULTRALEDS

[Ultraleds.co.uk](http://www.ultraleds.co.uk)

<http://www.ultraleds.co.uk>

tel: 0871 7110413 / 01625 576778

Large range of low cost Ultra bright leds and Led related lighting products. Major credit cards taken online with same day depatch.

## USB INSTRUMENTS

<http://www.usb-instruments.com>

USB Instruments specialises in PC based instrumentation products and software such as Oscilloscopes, Data Loggers, Logic Analysers which interface to your PC via USB.



## VIRTINS TECHNOLOGY

[www.virtins.com](http://www.virtins.com)

PC and Pocket PC based virtual instrument such as sound card real time oscilloscope, spectrum analyzer, signal generator, multimeter, sound meter, distortion analyzer, LCR meter. Free to download and try.



[www.elektor-electronics.co.uk](http://www.elektor-electronics.co.uk)

## SHOWCASE YOUR COMPANY HERE

Elektor Electronics has a feature to help customers promote their business, Showcase - a permanent feature of the magazine where you will be able to showcase your products and services.

- For just £220 + VAT (£20 per issue for eleven issues) Elektor will publish your company name, website address and a 30-word description
- For £330 + VAT for the year (£30 per issue for eleven issues) we will publish the above plus run a 3cm deep full colour

image - e.g. a product shot, a screen shot from your site, a company logo - your choice

Places are limited and spaces will go on a strictly first come, first served basis. So-please fax back your order today!

I wish to promote my company, please book my space:

- Text insertion only for £220 + VAT
- Text and photo for £330 + VAT

NAME: ..... ORGANISATION: .....

JOB TITLE: .....

ADDRESS: .....

..... TEL: .....

PLEASE COMPLETE COUPON BELOW AND FAX BACK TO 00-44-(0)1932 564998

COMPANY NAME .....

WEB ADDRESS .....

30-WORD DESCRIPTION .....

.....

.....

.....

.....



**“Elektor? Fun in finding solutions for my electronics hobby. Never failed to impress my boss or my wife.”**

– Thomas Gosling, 38, electronics enthusiast –

## Electronics at all the right levels

Secure a head start in electronics with a subscription!

### Advantages to subscribers:

- ✓ Cheaper than 11 issues from the newsstand
- ✓ Subscribers get up to **40% discount** on special Elektor products
- ✓ As a welcome gift you get a free **1GB MP3 player** worth £ 34.50
- ✓ No queues, travelling, parking fees or ‘sold out’ Elektor is supplied to your doorstep every month
- ✓ Always up to date – read your copy before everyone else



**FREE 3-in-1 welcome gift: 1GB MP3 player, USB Stick and Voice Recorder**

[www.elektor-electronics.co.uk/subs](http://www.elektor-electronics.co.uk/subs)  
**Tel. +44 (0) 208 261 4509**

Or use the subscription order form near the end of the magazine.  
144

**elektor**  
the world-leading network  
for electronics  
elektor electronics - 7-8/2007

Order now using the Order Form in the Readers Services section in this issue.

Elektor Electronics (Publishing)  
Regus Brentford  
1000 Great West Road  
Brentford TW8 9HH  
United Kingdom

Telephone +44 208 261 4509  
Fax +44 208 261 4447  
Email: sales@elektor-electronics.co.uk

More information on [www.elektor-electronics.co.uk](http://www.elektor-electronics.co.uk)

## CD-ROM BESTSELLERS

### Ethernet Toolbox

This CD-ROM contains all essential information regarding Ethernet interfaces! To help you learn about the Ethernet interfaces, we have compiled a collection of all articles on this topic that have appeared in Elektor Electronics and complemented them with additional documentation and links to introductory articles on Ethernet interfaces. It includes a collection of data sheets for dedicated Ethernet interface ICs from many different manufacturers. The CD-ROM provides a wealth of information about connectors and components for the physical layer (PHY) and specific software tools for use with the Ethernet (Software).

ISBN 978-90-5381-214-3 | £18.90 (US\$ 37.90)



### Elektor 2006

This CD-ROM contains all editorial articles published in Elektor Electronics Volume 2006. Using the supplied Acrobat Reader program, articles are presented in the same layout as originally found in the magazine. All free, printed, supplements our readers got last year, like the Visual Basic, C and i-TRIXX booklets are also contained on the CD. The Elektor Volume 2006 CD-ROM has a rather different look and feel than previous editions. It's gone through a makeover in more than one way!

ISBN 978-90-5381-207-5 | £16.25 (US\$ 28.75)



### Robotics

A large collection of data-sheets, software tools, tips, tricks and Internet links to assorted robot constructions and general technical information. All aspects of modern robotics are covered, from sensors to motors, mechanical parts to microcontrollers, not forgetting matching programming tools and libraries for signal processing.

ISBN 978-90-5381-179-5 | £12.05 (US\$ 21.25)



## Microcontroller Basics

Microcontrollers have become an indispensable part of modern electronics. They make things possible that vastly exceed what could be done previously. Innumerable applications show that almost nothing is impossible. There's thus every reason to learn more about them. This book offers more than just a basic introduction. It clearly explains the technology using various microcontroller circuits and programs written in several different programming languages. In the course of the book, the reader gradually develops increased competence in converting his or her ideas into microcontroller circuitry.



ISBN 978-0-905705-67-5

230 Pages

£18.70 (US\$ 33.70)

## Visual Basic for Electronics Engineering Applications

This book is targeted towards those people that want to control existing or home made hardware from their computer. After familiarizing yourself with Visual Basic, its development environment and the toolset it offers are discussed in detail. Each topic is accompanied by clear, ready to run code, and where necessary, schematics are provided that will get your projects up to speed in no time.



ISBN 978-0-905705-68-2

476 Pages

£27.50 (US\$ 51.50)

## BESTSELLING BOOKS

Top-5

- 1 **Visual Basic for Electronics Engineering Applications**  
ISBN 978-0-905705-68-2 £27.50 (US\$ 51.50)
- 2 **Microcontroller Basics**  
ISBN 978-0-905705-67-5 £18.70 (US\$ 33.70)
- 3 **PC-Interfaces under Windows**  
ISBN 978-0-905705-65-1 £25.95 (US\$ 52.00)
- 4 **Modern High-end Valve Amplifiers**  
ISBN 978-0-905705-63-7 £25.95 (US\$ 52.00)
- 5 **308 Circuits**  
ISBN 978-0-905705-66-8 £18.20 (US\$ 37.00)

More bestsellers on [www.elektor-electronics.co.uk](http://www.elektor-electronics.co.uk)



Order now using the Order Form in  
the Readers Services section in this issue.

## Stand-Alone OBD2 Analyser

(June 2007)

Kit of parts, incl. case, cable, front panel foil  
and mounting materials

070038-72

£ 51.70 / US\$ 97.50



Free downloads:

Online simulator examples  
(070038-21)

Manual  
(070038-W1)

See [www.elektor-electronics.co.uk](http://www.elektor-electronics.co.uk)

## Software Defined Radio

(May 2007)

Ready-populated and  
tested board

070039-91

£ 72.45 / US\$ 126.50



## Speedmaster

(May 2007)

The circuit voted winner of  
the R8C Design Competition!

Ready-populated and tested  
board (excl. R8C module)

070021-91

£ 51.70 / US\$ 97.45



### No. 367 JULY/AUGUST 2007

	£	\$
<b>Dual Battery</b>		
070343-1 PCB, bare	5.20	9.75
<b>Fast Charger for NiMH Batteries</b>		
070213-1 PCB, bare	www.thepcbshop.com	
<b>Lithium Charger</b>		
070273-1 PCB, bare	8.90	12.95
<b>Low Cost USB Demo Board</b>		
060342-81 CD-ROM, project software	5.20	9.75
060342-41 PIC18F4550, programmed	15.15	28.50
<b>LPC 900 Programmer</b>		
070084-1 PCB, bare	6.90	12.95
<b>MotoBox</b>		
070129-1 PCB, bare	www.thepcbshop.com	
070129-41 PIC16F628-04/P, programmed	14.80	27.95
<b>Propeller Prototyping Board for BoeBot</b>		
070275-1 PCB, bare	11.00	20.75
<b>Satnav for Robots</b>		
070350-81 CD-ROM, project software	5.20	9.75
070350-41 ATmega32, programmed	16.50	31.15
<b>Serial Interface for the Propeller</b>		
070276-1 PCB, bare	www.thepcbshop.com	
<b>Stereo Robot Ears</b>		
060040-1 PCB, bare	www.thepcbshop.com	
060040-81 CD-ROM, project software	5.20	9.75
060040-41 PIC16F88, programmed	10.00	18.85

### No. 366 JUNE 2007

<b>2.4 GHz WiFi Spectrum Analyser</b>		
070040-1 PCB, bare	8.25	15.50
070040-11 CD-ROM, Linux & Windows software	5.20	9.75
<b>Coil Clinic</b>		
060195-1 PCB, bare	7.60	14.25
060195-11 CD-ROM, project software	5.20	9.75
060195-41 Atmega48-20PU, programmed	5.20	9.75
<b>Linux Oscilloscope</b>		
060241-11 CD-ROM, project software	5.20	9.75

060241-W Program listing	free download	
<b>Stand-Alone OBD2 Analyser</b>		
070038-72 Kit of parts, incl. case, cable, front panel foil and mounting materials	51.70	97.50
070038-21 Online simulator examples	free download	
070038-W1 Manual	free download	

### Whistles from on high

060044-1 PCBs, bare; set for TX and RX	12.00	22.75
060044-11 CD-ROM, project software	5.20	9.75
060044-41 ATtiny15PC, programmed	10.00	18.75

### No. 365 MAY 2007

<b>Software Defined Radio</b>		
070039-91 Ready-populated and tested board	72.45	126.50
070039-11 CD-ROM, project software	5.20	9.75
<b>Thank you for Flying USB FliteSim</b>		
060378-1 PCB, bare	www.thepcbshop.com	
060378-41 PIC18F2550I/SP, programmed	15.50	29.25
<b>Universal JTAG Adaptor</b>		
060287-1 PCB, bare, with programmed microcontroller	11.00	20.75
060287-41 EP900LC only, programmed	P&P only P&P only	
<b>Magnetometer</b>		
050276-1 PCB, bare	www.thepcbshop.com	
<b>Speedmaster</b>		
070021-91 Ready-populated and tested board (excl. R8C module)	51.70	97.45
<b>Seismograph</b>		
060307-1 PCB, bare	www.thepcbshop.com	
060307-11 CD-ROM, project software	5.20	9.75
060307-41 ATtiny45, programmed	10.35	19.50
<b>ATtiny as RDS Signal Generator</b>		
060253-41 ATtiny2313-20, programmed	4.20	7.80

### No. 364 APRIL 2007

<b>Battery Charge-n-Check</b>		
050073-1 PCB, bare, main board	10.30	19.50
050073-2 PCB, bare, display board	10.30	19.50
050073-11 CD-ROM, project software	5.20	9.75
050073-41 ST7FMC2S4, programmed	16.90	31.85

Due to practical constraints, final illustrations and specifications may differ from published designs. Prices subject to change. See [www.elektor-electronics.co.uk](http://www.elektor-electronics.co.uk) for up to date information.

Elektor Electronics (Publishing)  
Regus Brentford  
1000 Great West Road  
Brentford TW8 9HH  
United Kingdom  
Tel.: +44 (0) 208 261 4509  
Fax: +44 (0) 208 261 4447  
Email: [sales@elektor-electronics.co.uk](mailto:sales@elektor-electronics.co.uk)

## Wireless USB in miniature

(March 2007)



iDwarf -168 Transmitter module (built & tested)

050402-91

£ 24.10 / US\$ 45.45

iDwarf Node Board (built & tested)

050402-91

£ 17.20 / US\$ 32.45



iDwarf Hub Board (built & tested)

050402-93

£ 17.20 / US\$ 32.45



## Kits & Modules

### g-Force on LEDs

(April 2007)

PCB set, bare, incl. 2 MMA260 sensors, BDM cable parts

060297-71

£ 10.00 / US\$ 18.85



### USB Stick with ARM and RS232

(November 2006)

Assembled and tested board

060006-91

£ 79.90 / \$ 149.95



### g-Force on LEDs

060297-71	PCB set, incl. 2 MMA260 sensors, BDM cable parts	10.00	18.85
060297-11	CD-ROM, project software	5.20	9.75

### Programmer for Freescale 68HC(9)08

060263-1	PCB, bare	<a href="http://www.thepcbshop.com">www.thepcbshop.com</a>	
----------	-----------	--	--

### A Simple Mains Inverter

060171-1	PCB, bare	<a href="http://www.thepcbshop.com">www.thepcbshop.com</a>	
----------	-----------	--	--

### Very Simple Clock

060350-1	PCB, bare	<a href="http://www.thepcbshop.com">www.thepcbshop.com</a>	
----------	-----------	--	--

### E-blocks Light Chaser Squared

075032-1	PCB, bare	<a href="http://www.thepcbshop.com">www.thepcbshop.com</a>	
----------	-----------	--	--

### No. 363 MARCH 2007

#### AVR drives USB

060276-1	PCB, bare	10.00	18.85
060276-11	CD-ROM, project software incl. source code	5.20	9.75
060276-41	ATmega32-16PC, programmed	8.95	16.85

#### Wireless USB in Miniature

050402-1	PCB, bare, iDwarf prototyping board	8.30	15.60
050402-91	iDwarf -168 Transmitter module (built & tested)	24.10	45.45
050402-92	iDwarf Node Board (built & tested)	17.20	32.45
050402-93	iDwarf Hub Board (built & tested)	17.20	32.45

#### Mobile Phone LCD for PC

060184-1	PCB, bare	<a href="http://www.thepcbshop.com">www.thepcbshop.com</a>	
060184-11	CD-ROM, project software	5.20	9.75
060184-41	ATmega16-16PC, programmed	8.95	16.85

#### Scale Deposit Fighter

070001-1	PCB, bare	<a href="http://www.thepcbshop.com">www.thepcbshop.com</a>	
----------	-----------	--	--

### No. 362 FEBRUARY 2007

#### ... 3, 2, 1 Takeoff!

050238-1	Transmitter PCB, bare	<a href="http://www.thepcbshop.com">www.thepcbshop.com</a>	
050238-2	Receiver PCB, bare	<a href="http://www.thepcbshop.com">www.thepcbshop.com</a>	

#### MP3 Preamp

060237-1	PCB, bare	<a href="http://www.thepcbshop.com">www.thepcbshop.com</a>	
----------	-----------	--	--

#### A Telling Way of Telling the Time

050311-1	PCB, bare	<a href="http://www.thepcbshop.com">www.thepcbshop.com</a>	
050311-31	CPLD, programmed	35.50	66.95

### FPGA Course (9)

060025-9-11	CD-ROM, course software incl. source code	5.20	9.75
-------------	---	------	------

### Explorer-16 Value Pack

060280-91	Four components packaged together in a single box	122.90	232.50
-----------	---	--------	--------

### No. 361 JANUARY 2007

#### Sputnik Time Machine

050018-1	PCB	<a href="http://www.thepcbshop.com">www.thepcbshop.com</a>	
050018-11	CD-ROM, project software (incl. source code)	5.20	9.75
050018-41	AT89C2051, programmed	3.40	6.45

#### Very Simple Clock

060350-1	PCB	<a href="http://www.thepcbshop.com">www.thepcbshop.com</a>	
060350-11	CD-ROM, project software (incl. source code)	5.20	9.75
060350-41	PIC16F628-20, programmed	5.50	10.35

### FPGA Course (8)

060025-8-1	Software (incl. source code)	5.20	9.75
------------	------------------------------	------	------

### No. 360 DECEMBER 2006

#### Shortwave Capture

030417-1	PCB, bare (receiver board)	<a href="http://www.thepcbshop.com">www.thepcbshop.com</a>	
030417-2	PCB, bare (control & display boards)	<a href="http://www.thepcbshop.com">www.thepcbshop.com</a>	
030417-41	AT90S8515-8PC, programmed	11.40	21.45

### No. 359 NOVEMBER 2006

#### USB Stick with ARM and RS232

060006-1	PCB, bare	11.00	20.75
060006-41	AT91SAM7S64, programmed	27.60	51.95
060006-91	Assembled & tested board	79.90	149.95
060006-81	CD-ROM, all project software	5.20	9.75

### No. 358 OCTOBER 2006

#### PIC In-Circuit Debugger/Programmer

050348-1	PCB	5.20	9.75
050348-41	PIC16F877, programmed	17.90	33.75
050348-71	Kit, incl. PCB, controller, all parts	34.50	64.95

Products for older projects (if available) may be found on our website [www.elektor-electronics.co.uk](http://www.elektor-electronics.co.uk)  
home construction = fun and added value

**EXTRA!**

**FREE LED DRIVER**  
with every issue!

**With preassembled components!**  
**Experiment with white LEDs!**



**Equipment Test: audio amplifier modules**

Home construction of power amplifiers remains a popular activity among audiophile electronics enthusiasts. Besides the high-end audio construction projects Elektor has become famous for, you may also consider using one of the power amplifier modules available commercially either ready-built or as a kit. These modules require only a power supply and an enclosure to make an audio amp. We have collected a number of interesting AF power modules and put them through their paces in our audio lab. The test results are presented in the September issue.

**GPS Tracker**

Ten years ago a technical novelty, now dead common: positioning systems using GPS (Global Positioning System). GPS-based navigators are extremely popular, especially among motorists. However, a GPS receiver is not just suitable for determining one's own location on the globe, but also that of an object, for example, your car! Elektor's GPS Tracker was developed specifically for that purpose. It's a small circuit comprising a GSM modem, a GPS receiver and a miniature antenna. When the GSM modem is texted (by SMS), the receiver will return its current coordinates. This allows you to track the object the circuit is attached to.



**RESERVE YOUR COPY NOW!**

UK mainland subscribers will receive the magazine between 18 and 21 August 2007.

The September 2007 issue goes on sale on Thursday 23 August 2007 (UK distribution only).  
Article titles and magazine contents subject to change, please check [www.elektor-electronics.co.uk](http://www.elektor-electronics.co.uk).

**NEWSAGENTS ORDER FORM**

SHOP SAVE / HOME DELIVERY

Please save / deliver one copy of *Elektor Electronics* magazine for me each month

Name: .....

Address: .....

Post code: .....

Telephone: .....

Date: .....

Signature: .....



Please cut out or photocopy this form, complete details and hand to your newsagent.  
Elektor Electronics is published on the third Thursday of each month, except in July.  
Distribution S.O.R. by Seymour (NS).

[www.elektor-electronics.co.uk](http://www.elektor-electronics.co.uk) [www.elektor-electronics.co.uk](http://www.elektor-electronics.co.uk) [www.elektor-electronics.co.uk](http://www.elektor-electronics.co.uk)

**Elektor Electronics on the web**

All magazine articles back to volume 2000 are available online in pdf format. The article summary and parts list (if applicable) can be instantly viewed to help you positively identify an article. Article related items are also shown, including software downloads, circuit boards, programmed ICs and corrections and updates if applicable. Complete magazine issues may also be downloaded.

In the Elektor Electronics Shop you'll find all other products sold by the publishers, like CD-ROMs, kits and books. A powerful search function allows you to search for items and references across the entire website.

**Also on the Elektor Electronics website:**

- Electronics news and Elektor announcements
- Readers Forum
- PCB, software and e-magazine downloads
- Surveys and polls
- FAQ, Author Guidelines and Contact





Please supply the following. For PCBs, front panel foils, EPROMs, PALs, GALs, microcontrollers and diskettes, state the part number and description; for books, state the full title; for photocopies of articles, state full name of article and month and year of publication. PLEASE USE BLOCK CAPITALS.

Description	Price each	Qty.	Total	Order Code
<b>309 Circuits</b>	£ 19.95			
<b>Formula Flowcode Buggy</b>	£ 85.00			
<b>CD-ROM Ethernet Toolbox</b>	£ 18.90			
<b>CD-ROM Elektor 2006</b>	£ 16.25			
<b>Visual Basic for Electronics Engineering Applications</b>	£ 27.50			
<b>Sub-total</b>				
<b>P&amp;P</b>				
<b>Total paid</b>				

Prices and item descriptions subject to change. The publishers reserve the right to change prices without prior notification. Prices and item descriptions shown here supersede those in previous issues. E. & O.E.

Name \_\_\_\_\_

Address + Post code \_\_\_\_\_

Tel. \_\_\_\_\_ Email \_\_\_\_\_

Date \_\_\_\_\_ Signature \_\_\_\_\_

EL07/08





### METHOD OF PAYMENT

(see reverse before ticking as appropriate)

**Bank transfer**

**Cheque**  
(UK-resident customers ONLY)

**Giro transfer**

\_\_\_\_\_

Expiry date: .....

Verification code: .....

SWITCH ONLY:

Start date: .....

Issue number: .....

Please send this order form to \*  
(see reverse for conditions)

**Elektor Electronics** (Publishing)

Regus Brentford  
1000 Great West Road  
Brentford TW8 9HH  
United Kingdom

Tel.: +44 208 261 4509  
Fax: +44 208 261 4447  
www.elektor-electronics.co.uk  
sales@elektor-electronics.co.uk

\*USA and Canada residents may (but are not obliged to) use \$ prices, and send the order form to:  
Old Colony Sound Lab  
P.O. Box 876, Peterborough  
NH 03458-0876. Tel. (603) 924-6371, 924-6526,  
Fax: (603) 924-9467  
Email: custserv@audioXpress.com



**Yes, I am taking out an annual subscription to Elektor Electronics and receive a free 1GB MP3 player.**



- I would like:
- Standard Subscription (11 issues)**
  - Subscription-Plus (11 issues plus the Elektor Volume 2007 CD-ROM)**

\* Offer available to Subscribers who have not held a subscription to Elektor Electronics during the last 12 months. Offer subject to availability. See reverse for rates and conditions.

Name \_\_\_\_\_

Address + Post code \_\_\_\_\_

Tel. \_\_\_\_\_ Email \_\_\_\_\_

Date \_\_\_\_\_ Signature \_\_\_\_\_

\* cross out what is not applicable

EL07/08





### METHOD OF PAYMENT

(see reverse before ticking as appropriate)

**Bank transfer**

**Cheque**  
(UK-resident customers ONLY)

**Giro transfer**

\_\_\_\_\_

Expiry date: .....

Verification code: .....

SWITCH ONLY:

Start date: .....

Issue number: .....

Please send this order form to

**Elektor Electronics** (Publishing)

Regus Brentford  
1000 Great West Road  
Brentford TW8 9HH  
United Kingdom

Tel.: +44 208 261 4509  
Fax: +44 208 261 4447  
www.elektor-electronics.co.uk  
subscriptions@elektor-electronics.co.uk

## ORDERING INSTRUCTIONS, P&P CHARGES

Except in the USA and Canada, all orders, except for subscriptions (for which see below), must be sent BY POST or FAX to our Brentford address using the Order Form overleaf. On-line ordering: <http://www.elektor-electronics.co.uk>

**Readers in the USA and Canada** may (but are not obliged to) send orders, except for subscriptions (for which see below), to the USA address given on the order form. Please apply to Old Colony Sound for applicable P&P charges. Please allow 4-6 weeks for delivery.

**Orders placed on our Brentford office** must include P&P charges (Priority or Standard) as follows:

UK: £4.00 Europe: £5.00 (Standard) or £7.00 (Priority) Outside Europe: £8.00 (Standard) or £12.00 (Priority)

## HOW TO PAY

All orders must be accompanied by the full payment, including postage and packing charges as stated above or advised by Customer Services staff.

**Bank transfer** into account no. 40209520 held by Elektor Electronics (Publishing) / Segment b.v. with ABN-AMRO Bank, London. IBAN: GB35 ABNA 4050 3040 2095 20. BIC: ABNAGB2L. Currency: sterling (UKP). Please ensure your full name and address gets communicated to us.

**Cheque** sent by post, made payable to Elektor Electronics (Publishing) / Segment b.v.. We can only accept sterling cheques and bank drafts from UK-resident customers or subscribers. We regret that no cheques can be accepted from customers or subscribers in any other country.

**Giro transfer** into account no. 34-152-3801, held by Elektor Electronics (Publishing) / Segment b.v. Please do not send giro transfer/deposit forms directly to us, but instead use the National Giro postage paid envelope and send it to your National Giro Centre.

**Credit card** VISA, Access, MasterCard, JCBCard and Switch cards can be processed by mail, email, web, fax and telephone. Online ordering through our website is SSL-protected for your security.

## COMPONENTS

Components for projects appearing in Elektor Electronics are usually available from certain advertisers in this magazine. If difficulties in the supply of components are envisaged, a source will normally be advised in the article. Note, however, that the source(s) given is (are) not exclusive.

## TERMS OF BUSINESS

**Delivery** Although every effort will be made to dispatch your order within 2-3 weeks from receipt of your instructions, we can not guarantee this time scale for all orders. **Returns** Faulty goods or goods sent in error may be returned for replacement or refund, but not before obtaining our consent. All goods returned should be packed securely in a padded bag or box, enclosing a covering letter stating the dispatch note number. If the goods are returned because of a mistake on our part, we will refund the return postage. **Damaged goods** Claims for damaged goods must be received at our Brentford office within 10-days (UK); 14-days (Europe) or 21-days (all other countries). **Cancelled orders** All cancelled orders will be subject to a 10% handling charge with a minimum charge of £5.00. **Patents** Patent protection may exist in respect of circuits, devices, components, and so on, described in our books and magazines. Elektor Electronics (Publishing) does not accept responsibility or liability for failing to identify such patent or other protection. **Copyright** All drawings, photographs, articles, printed circuit boards, programmed integrated circuits, diskettes and software carriers published in our books and magazines (other than in third-party advertisements) are copyright and may not be reproduced or transmitted in any form or by any means, including photocopying and recording, in whole or in part, without the prior permission of Elektor Electronics (Publishing) in writing. Such written permission must also be obtained before any part of these publications is stored in a retrieval system of any nature. Notwithstanding the above, printed-circuit boards may be produced for private and personal use without prior permission. **Limitation of liability** Elektor Electronics (Publishing) shall not be liable in contract, tort, or otherwise, for any loss or damage suffered by the purchaser whatsoever or howsoever arising out of, or in connexion with, the supply of goods or services by Elektor Electronics (Publishing) other than to supply goods as described or, at the option of Elektor Electronics (Publishing), to refund the purchaser any money paid in respect of the goods. **Law** Any question relating to the supply of goods and services by Elektor Electronics (Publishing) shall be determined in all respects by the laws of England.

January 2007

## SUBSCRIPTION RATES FOR ANNUAL SUBSCRIPTION

	Standard	Plus
United Kingdom	£41.90	£48.80
<b>Surface Mail</b>		
Rest of the World	£54.50	£61.40
USA & Canada	US\$ 95.50	US\$106.50
<b>Airmail</b>		
Rest of the World	£68.90	£75.80
USA & Canada	US\$120.00	US\$131.00

## HOW TO PAY

**Bank transfer** into account no. 40209520 held by Elektor Electronics (Publishing) / Segment b.v. with ABN-AMRO Bank, London. IBAN: GB35 ABNA 4050 3040 2095 20. BIC: ABNAGB2L. Currency: sterling (UKP). Please ensure your full name and address gets communicated to us.

**Cheque** sent by post, made payable to Elektor Electronics (Publishing) / Segment b.v.. We can only accept sterling cheques and bank drafts from UK-resident customers or subscribers. We regret that no cheques can be accepted from customers or subscribers in any other country.

**Giro transfer** into account no. 34-152-3801, held by Elektor Electronics (Publishing) / Segment b.v. Please do not send giro transfer/deposit forms directly to us, but instead use the National Giro postage paid envelope and send it to your National Giro Centre.

**Credit card** VISA, Access, MasterCard, JCBCard and Switch cards can be processed by mail, email, web, fax and telephone. Online ordering through our website is SSL-protected for your security.

## SUBSCRIPTION CONDITIONS

The standard subscription order period is twelve months. If a permanent change of address during the subscription period means that copies have to be despatched by a more expensive service, no extra charge will be made. Conversely, no refund will be made, nor expiry date extended, if a change of address allows the use of a cheaper service.

Student applications, which qualify for a 20% (twenty per cent) reduction in current rates, must be supported by evidence of studentship signed by the head of the college, school or university faculty. A standard Student Subscription costs £33.50, a Student Subscription-Plus costs £40.40 (UK only).

Please note that new subscriptions take about four weeks from receipt of order to become effective.

Cancelled subscriptions will be subject to a charge of 25% (twenty-five per cent) of the full subscription price or £7.50, whichever is the higher, plus the cost of any issues already dispatched. Subscriptions cannot be cancelled after they have run for six months or more.

January 2007



ISBN 978-90-5381-214-3  
£18.90 / US\$ 37.90

**NEW**

## CD-ROM Ethernet Toolbox

### Software Tools en Hardware Tips

This CD-ROM contains all essential information regarding Ethernet interfaces! To help you learn about the Ethernet interfaces, we have compiled a collection of all articles on this topic that have appeared in Elektor Electronics and complemented them with additional documentation and links to introductory articles on Ethernet interfaces.

It includes a collection of datasheets for dedicated Ethernet interface ICs from many different manufacturers. To help you with your own projects, the CD-ROM provides a wealth of information about connectors and components for the physical layer (PHY) and specific software tools for use with the Ethernet (Software).

All of the documents are PDF files.

- Controllers & Microcontrollers
- Boards with Ethernet connectivity
- Serial to Ethernet
- CAN to Ethernet
- Connectors
- Physical Layer Transceiver
- IEEE 802@ standard

Order now using the Order Form in the Readers Services section in this issue.  
Elektor Electronics (Publishing)  
Regus Brentford  
1000 Great West Road  
Brentford TW8 9HH  
United Kingdom  
Tel. +44 208 261 4509

See also [www.elektor-electronics.co.uk](http://www.elektor-electronics.co.uk)

## INDEX OF ADVERTISERS

ATC Semitec Ltd, Showcase . . . . .	<a href="http://www.atcsemitec.co.uk">www.atcsemitec.co.uk</a> . . . . .	142	Number One Systems . . . . .	<a href="http://www.numberone.com">www.numberone.com</a> . . . . .	60
Avit Research, Showcase . . . . .	<a href="http://www.avitresearch.co.uk">www.avitresearch.co.uk</a> . . . . .	142	Nurve Networks . . . . .	<a href="http://www.xgamestation.com">www.xgamestation.com</a> . . . . .	139
BAEC, Showcase . . . . .	<a href="http://baec.tripod.com">http://baec.tripod.com</a> . . . . .	142	Paltronix . . . . .	<a href="http://www.paltronix.com">www.paltronix.com</a> . . . . .	7
Beijing Draco . . . . .	<a href="http://www.ezpcb.com">www.ezpcb.com</a> . . . . .	109	PCB World, Showcase . . . . .	<a href="http://www.pcbworld.org.uk">www.pcbworld.org.uk</a> . . . . .	143
Beta Layout, Showcase . . . . .	<a href="http://www.pcb-pool.com">www.pcb-pool.com</a> . . . . .	139, 142	Peak Electronic Design . . . . .	<a href="http://www.peakelec.co.uk">www.peakelec.co.uk</a> . . . . .	6
Bitscope Designs . . . . .	<a href="http://www.bitscope.com">www.bitscope.com</a> . . . . .	3	Pico . . . . .	<a href="http://www.picotech.com">www.picotech.com</a> . . . . .	109
Compact Control Design . . . . .	<a href="http://www.compactcontrol.co.uk">www.compactcontrol.co.uk</a> . . . . .	39	Quasar Electronics . . . . .	<a href="http://www.quasarelectronics.com">www.quasarelectronics.com</a> . . . . .	97
Decibit Co. Ltd, Showcase . . . . .	<a href="http://www.decibit.com">www.decibit.com</a> . . . . .	142	Robot Electronics, Showcase . . . . .	<a href="http://www.robot-electronics.co.uk">www.robot-electronics.co.uk</a> . . . . .	143
EasyDAC, Showcase . . . . .	<a href="http://www.easydaq.biz">www.easydaq.biz</a> . . . . .	142	Scantool . . . . .	<a href="http://www.ElmScan5.com/elektor">www.ElmScan5.com/elektor</a> . . . . .	139
Easysync, Showcase . . . . .	<a href="http://www.easysync.co.uk">www.easysync.co.uk</a> . . . . .	142	Schaeffer AG . . . . .	<a href="http://www.schaeffer-ag.de">www.schaeffer-ag.de</a> . . . . .	109
Eltec, Showcase . . . . .	<a href="http://www.eltec.com">www.eltec.com</a> . . . . .	142	Showcase . . . . .	. . . . .	142, 143
Eurocircuits . . . . .	<a href="http://www.eurocircuits.com">www.eurocircuits.com</a> . . . . .	139	SourceBoost Technologies, Showcase . . . . .	<a href="http://www.sourceboost.com">www.sourceboost.com</a> . . . . .	143
First Technology Transfer Ltd, Showcase . . . . .	<a href="http://www.ftt.co.uk">www.ftt.co.uk</a> . . . . .	142	Sytronic Technology Ltd, Showcase . . . . .	<a href="http://www.m2mtelemetry.com">www.m2mtelemetry.com</a> . . . . .	143
Future Technology Devices, Showcase . . . . .	<a href="http://www.ftdichip.com">www.ftdichip.com</a> . . . . .	142	Tsien (UK) Ltd, Showcase . . . . .	<a href="http://www.componentbin.com">www.componentbin.com</a> . . . . .	6, 143
Futurlec, Showcase . . . . .	<a href="http://www.futurlec.com">www.futurlec.com</a> . . . . .	142	Ultraleds, Showcase . . . . .	<a href="http://www.ultraleds.co.uk">www.ultraleds.co.uk</a> . . . . .	143
Jaycar Electronics . . . . .	<a href="http://www.jaycarelectronics.co.uk">www.jaycarelectronics.co.uk</a> . . . . .	2	USB Instruments, Showcase . . . . .	<a href="http://www.usb-instruments.com">www.usb-instruments.com</a> . . . . .	143
JB Systems, Showcase . . . . .	<a href="http://www.modetron.com">www.modetron.com</a> . . . . .	142	Virtins Technology, Showcase . . . . .	<a href="http://www.virtins.com">www.virtins.com</a> . . . . .	143
Labcenter . . . . .	<a href="http://www.labcenter.com">www.labcenter.com</a> . . . . .	152			
London Electronics College, Showcase . . . . .	<a href="http://www.lec.org.uk">www.lec.org.uk</a> . . . . .	142			
Microchip . . . . .	<a href="http://www.microchip.com">www.microchip.com</a> . . . . .	61			
Mikro Elektronika . . . . .	<a href="http://www.mikroe.com">www.mikroe.com</a> . . . . .	8, 9			
MQP Electronics, Showcase . . . . .	<a href="http://www.mqp.com">www.mqp.com</a> . . . . .	142			
New Wave Concepts, Showcase . . . . .	<a href="http://www.new-wave-concepts.com">www.new-wave-concepts.com</a> . . . . .	142			
Newbury Electronics . . . . .	<a href="http://www.newburyelectronics.co.uk">www.newburyelectronics.co.uk</a> . . . . .	139			

**Advertising space for the issue of 24 September 2007  
may be reserved not later than 28 August 2007**

with Huson International Media – Cambridge House – Gogmore Lane –  
Chertsey, Surrey KT16 9AP – England – Telephone 01932 564 999 –  
Fax 01932 564998 – e-mail: [gerryb@husonmedia.com](mailto:gerryb@husonmedia.com) to whom all  
correspondence, copy instructions and artwork should be addressed.



# PROTEUS

# DESIGN SUITE

## TIME FOR A CHANGE ?



## NEW IN DESIGN SUITE 7:

**NEW:** Redesigned User Interface includes modeless selection, modeless wiring and intuitive operation to maximise speed and ease of use.

**NEW:** Design Explorer provides easy navigation, design inspection tools and cross-probing support to improve quality assurance and assist with fault finding.

**NEW:** 3D Visualisation Engine provides the means to preview boards in the context of a mechanical design prior to physical prototyping.

**NEW:** Simulation Advisor includes reporting on simulation problems with links to detailed troubleshooting information where appropriate.

**NEW:** Trace capability within both MCU and peripheral models provides detailed information on system operation which allows for faster debugging of both hardware and software problems.

**NEW:** Hundreds of new device models including PIC24, LPC2000, network controllers and general purpose electronic components.

## Electronic Design From Concept To Completion

**Labcenter**  [www.labcenter.com](http://www.labcenter.com)  
**Electronics**

Labcenter Electronics Limited  
Registered in England 4692454

E-mail: [info@labcenter.com](mailto:info@labcenter.com)

Tel: +44 (0) 1756 753440

Fax: +44 (0) 1756 752857

Registered Address: 53-55 Main Street, Grassington, North Yorks, UK, BD23 5AA