

An Intelligent Robot!?

by Kieron McGeever

What do we mean by an “Intelligent Robot”?

One that can solve problems and build Picasso type structures out of car parts or one that can blunder its way around a maze? Any of those types of robot would be amazing, but it is far better to keep it small and simple.

If you have built the TEP Rover kit it is just a small step to give it some simple autonomous operations. Instead of using the slider switch control supplied with the kit, fit to it instead an IQ controller board, this will allow you to write simple programmes that will



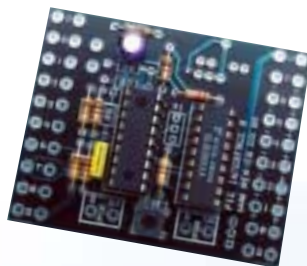
▲ IQ Controller

control the motor of the buggy, making it go forward and turn to the left and the right. The IQ controller board has three outputs that can be controlled separately, one for each motor and one controlling an LED for example. The outputs can be simply programmed from the

buttons on the board and there is even provision for having a separate power supply for the motors and the boards' circuits. The disadvantages of the IQ board are that you only have forward and stop on each motor (i.e. on/off) and it loses its programme when it is turned off. But, a lot of fun can be had with just this combination, for example, dancing competitions, obstacle races, and dropping a marker onto a target. The main advantage is that it is cheap to buy and simple to operate, but not intelligent.

PIC Chips

A robot might be thought to demonstrate some intelligence if it appeared to make decisions for its self, for example turning away from obstacles that are in its path. This can be achieved fairly simply by using a PIC chip, instead of an IQ board, attached to a “High Power Project Board” The chip to use is a 16F627 which has two analogue inputs, four digital inputs and eight outputs, four of which, when it is fitted to the high output board, are routed through a motor driver chip (L293D). This combination of chips allows you to have forward and reverse on two standard dc motors. The ability of the PIC chip to be programmed through the “Chip Factory” or computer software so that its outputs are controlled by its inputs means that a robot can steer its own way through an obstacle course.



▲ PIC Control board

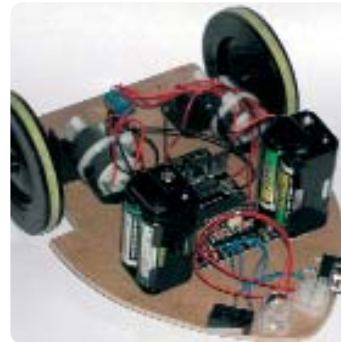
Building a PIC Controlled Robot

Using the components from the Rover Kit as a starting point for the robot you will need:

- 1 PIC High Power Project Board,
- 1 L293D motor driver chip,
- 1 PIC16F627 chip,
- 2 large wheels,
- 2 LDR's,
- 2 micro-switches,
- 2 battery snaps,
- 2 6 volt battery holders,
- 1 double pole, single throw switch (it can be a DPDT)
- 2 resistors (5.6k)



The motors and gearboxes from the kit need to be adapted so that they will sit side by side on the chassis, as the robot is going to be a two and not four wheeled vehicle, to make steering and control easier. To do this the axles will need to be shortened on one side of the gear box so that you have a left handed and right handed motor. The chassis of the robot can be any shape you want it to be, the two versions shown here are round and arrow shaped, but any



suitable design will do. The arrow shape is my preference because it allows room for the positioning of the components all on one surface. The position of the components is shown in **diagram 1**.

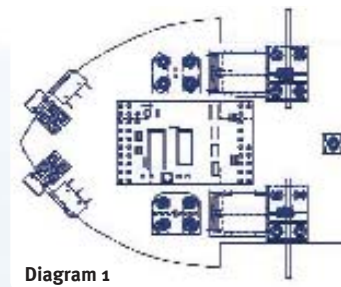


Diagram 1

Fix the gearboxes down with 4mm nuts and bolts, but everything else with double sided sticky pads for ease of modifications later on.

Fit the motor driver chip onto the PIC board and cut the connector resistor shown in **diagram 2** so that separate

power supplies can be attached to drive the motors and the PIC chip and attach the battery connectors.

On the right hand side of the board attach wires to output pins 4, 5, 6, and 7 labelled motor A and motor B.

On the right hand side of the board inputs 0 and 1 are the analogue

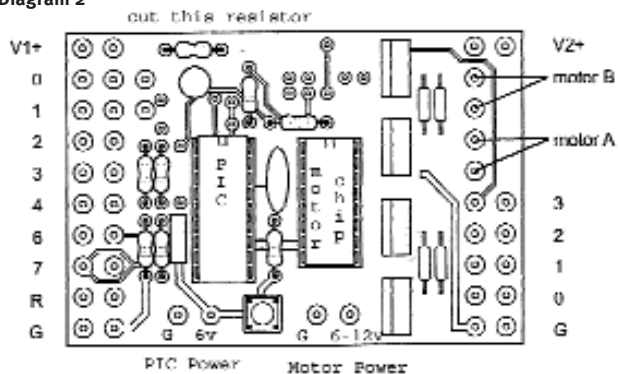
Continued on next page >>

An Intelligent Robot!?

by Kieron McGeever

(continued)

Diagram 2



inputs and have three connecting pads each and inputs 2, 3, 4, and 6 are the digital inputs with two connecting pads each.

Attach the micro-switches via wires to inputs 2 and 3 using the common and normally open pins on the switches.

The LDR's are a little more difficult to attach as they use the 5.6k resistors to set the sensitivity of the inputs.

The 5.6k resistor is attached to the **outer pad of input 0** and the **centre pad** and then two wires, to attach the LDR to, are fixed to the **centre pad** and the **inner pad** (note, the resistor and a wire goes into the centre pad).

Do the same for input 1. The wires are then attached to the LDR with a double connector block for ease of sticking to the base.

You can experiment with the value of the 5.6k resistor but I have found this value works for average lighting conditions indoors.

Now fix the PIC board to the chassis and connect the wires for motor A to the left hand motor and the wires for motor B to the right hand motor.

Fix the micro-switches and the LDR's to the front of the robot, these positions can be easily altered later to suit modifications in your design.

Fit the battery boxes and as an optional extra, connect the battery snaps via the DPST switch.

You are now ready to write your first PIC programme and test your robot. Using the "Chip Factory", power up and select programme New and chip 18L (the code for the 16F627) and input the following programme:

```
00      out o80    both motors go forward
01 If 3 on  out o16  left motor forward right motor stopped
02 If 2 on  out o64  right motor forward left motor stopped
03          goto 00  return to start
04          end programme
```

Burn this programme onto your PIC and fit it to your robot.

When the robot is powered up check that both motors are running in the same direction and forwards. If one of the motors is running in reverse and the other forwards change the polarity on the

reversing motor by swapping the wires round.

Now press the left hand micro-switch and the right hand motor should stop and the left hand motor go forward. The opposite should happen for the right hand micro-switch. Adjust the positions of the switches if this does not happen.

When you have your robot running correctly you are in the position to do some serious programming. But, first try this:

```
00      out o80
01 if a = b  goto 04
02 if a > b  goto 07
03 if b > a  goto 10
04 if 2 on   goto 10
05 if 3 on   goto 07
06          goto 00
07          out o16
08          wait o10
09          goto 00
10          out o64
11          wait o10
12          goto 00
13
```

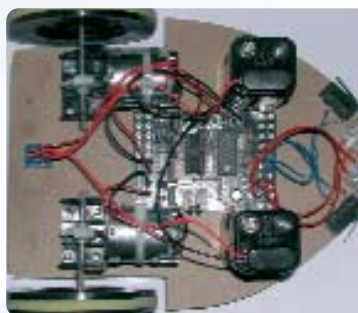
This programme will make your robot autonomous, but only test it by placing it on the floor, otherwise damage could occur.

In the above programme, a and b are the analogue inputs connected to the LDR's and the chip is being asked to compare the light level falling on each LDR. If you have positioned the LDR's so that they look to the left and the right, then the robot will only go in a straight line if the light levels are the same.

The micro-switches act as bumpers and if the robot bumps into an obstacle it will turn away from it. This is a very simple programme but it will make the robot perform some very complicated behaviour.

If you can answer these two simple questions you are ready for some advanced programming in the next article.

1. Is the robot "Light seeking"?
2. How could you reverse this action?



📧 If you have any further questions you can contact me at: mac@cercot.demon.co.uk

Kieron McGeever runs the TEP Development Centre at Rossett School, Harrogate.