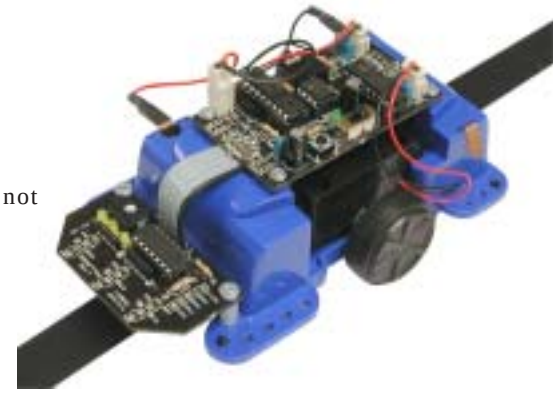
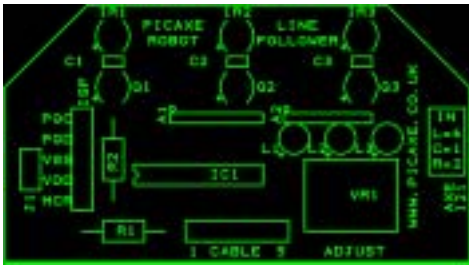
13) Using the AXE121 Line Follower Kit

The optional infra-red line-followr kit (part AXE121) allows the AXE120 micro-robot to follow a line drawn on the floor. Three infra-red sensors are used to detect the edges of the line, so that the micro-robot can follow the line.



IMPORTANT NOTES:
Do not remove ST-7L INFRARED SENSORS from the separate packet until required, as the infrared sensors and EL-7L infrared LEDs look identical - do not mix up! IR1-3, Q1-3 and C1-3 are mounted on the BOTTOM of the PCB.

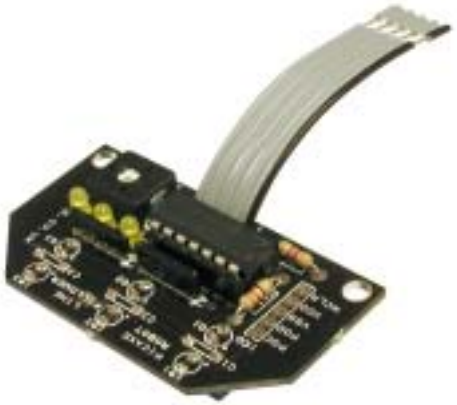
Kit Contents/Assembly:



Hardware
PCB
5 pin CABLE
2x M3 nuts
2x M3 20mm bolts
2x nylon spacers

Components mounted on TOP of PCB (ie solder on bottom of PCB)

IC1	14 pin IC socket and pre-programmed PIC16F676
R1	4k7 resistor (yellow violet red gold)
R2	12R resistor (brown red black gold)
A1	4k7 resistor array (marked 472G)
A2	1k resistor array (marked 102G)
VR1	100k preset
L1-3	3mm yellow LED



1) Solder the two resistors R1 (4k7) and R2 (12R) in position.
2) Solder the IC1 14 pin socket and VR1 preset in position.
3) Solder the three yellow LEDs L1-3 in position, making sure the long leg is in the hole marked +.
4) Solder resistor arrays A1 and A2 in position. Make sure the ink 'dot' at the end of the array aligns with the dot marked on the PCB (to the left).
5) Solder the 5 core cable in the position marked CABLE.
6) Place the microcontroller into the 14 pin socket, ensuring pin1 (dent) faces the left hand side.

Note that the positions marked ISP and JP1 are not normally used.

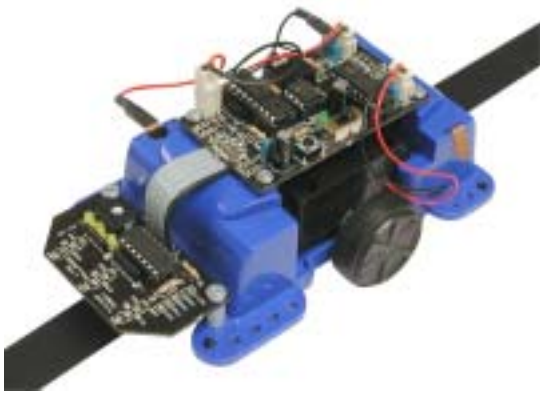
Components mounted on BOTTOM of PCB (ie solder on top of PCB)

C1-3	100nF capacitors
IR1-3	3mm infrared (clear) LED EL-7L
Q1-3	3mm infrared phototransistor ST-7L

1) Solder the three capacitors C1-3 in position. These act as a 'wall' between the infrared LED and phototransistor, to ensure only reflected light is detected.
2) Solder the three clear infrared LEDs IR1-3 in position, making sure the long leg is in the hole marked +. Make sure the LED sits 'flat' and straight on the PCB.
3) Solder the three ST-7L phototransistors Q1-3 in position, making sure the long leg is in the hole marked +. Make sure the phototransistor sits 'flat' on the PCB.

Connecting to micro-robot and testing.

1) Remove control PCB from top of the micro-robot.
2) Solder the 5 pin cable to the main micro-robot control PCB. Use the **INNER** set of 5 holes marked '**CABLE**', not the outer SRF005 connector. The cable should come from below the control PCB, so that you are soldering on the top of the control PCB.
3) Use the M3 nuts, bolts and spacers to mount the line follower module at the front of the micro-robot as shown.
4) Replace control PCB on top of micro-robot.
5) Switch the micro-robot on. Hold the micro-robot above a black line (e.g. black insulation tape) on white background. Each yellow LED should come on as the corresponding sensor is moved over a black line. Use the preset resistor to adjust the sensitivity.



Note - ensure you connect the cable to the inner CABLE holes on the microrobot PCB (not the outer SRF005 connection holes!)

Advanced Use Only (optional adjustments).

1) Solder a jumper link or switch in position JP1 to invert colours (ie to follow a white line on a black background).
2) The PIC16F676 can be re-programmed (if desired, by users familiar with assembler code) via the header marked ISP.

Operation.

The pre-programmed PIC16F676 micro-controller cycles each infra-red LED/sensor pair in turn. Each LED is quickly switched on and off by itself to create a burst of light. The amount of light reflected from the ground is then measured by the infra-red sensor (white and black backgrounds reflect different amounts of light). If the reflected light level indicates a black line, the corresponding output LED is lit.

Sample program.

This sample program shows how to follow a line. It is a simple program using no speed control - naturally you can be more creative, and, for instance, adjust the program so the micro-robot goes faster when over the black line, or adjust the program so that it 'hunts down' the nearest black line - be creative!

```
init: pause 100               ' motor controller start up pause
main:
      if input1 is on then go_f     ' forward
      if input2 is on then go_l     ' left
      if input6 is on then go_r     ' right
      goto go_s                     ' stop as no line nearby

go_f: let pins = %10100000          ' go forward
      goto main

go_l: let pins = %00100000          ' go left
      goto main

go_r: let pins = %10000000          ' go right
      goto main

go_s:
      let pins = %00000000          ' stop - not over line!
      goto main
```