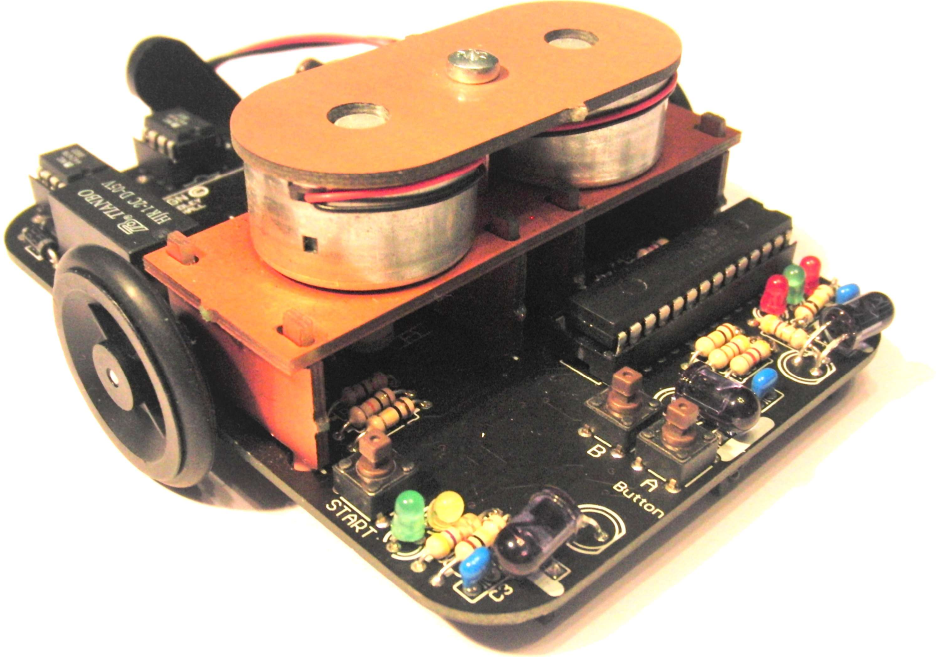
# The autonomous robot kit

## Ideally suited for both students and amateur enthusiasts.



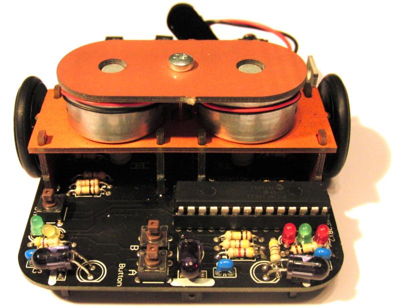
- Complete photo construction plans with simple component explanation.
- Maze solving algorithm included

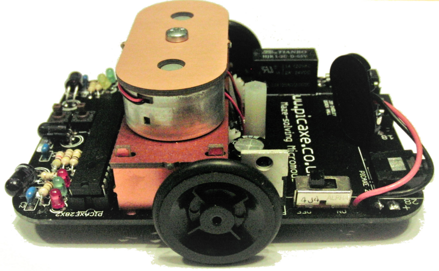
This kit provides practical teaching through a stage by stage assembly; including electronic component soldering, motor gearbox construction and computer control through programming, encouraging the link between hardware and software understanding.

**What is Micromouse ?**

Micromouse is an event where small autonomous robot mice compete against each other to solve a maze. The maze consists of 18cm square cells with 5cm high walls set over a 16 by 16 grid. The mice must find their way unassisted from the start corner of a maze, with an unknown layout, to the centre. As they search the maze mice track their position and plot walls as they discover them, in order to map the optimal route. Once the route has been discovered the mouse returns to the start square and runs this path in the shortest possible time. Mircomouse events are regularly held in schools, colleges, universities and exhibition centres around the world. Details of the UK championships can be found at www.micromouseonline.com.

Over the years the challenge to create an autonomous robot that can solve a maze has spread worldwide with regular international championships. The original challenge was to find the centre of the maze (which in itself demands skills in micro robotics and programming), but over the years the events have developed into a race to the centre, with mice capable of speeds in excess of 4 metres per second.





This low cost Micromouse kit is designed for both students and amateur enthusiasts. It demonstrates computer control responding to an external environment and has been developed and designed by the 2007 UK Micromouse champions Derek Hall and Jim Chidley.

The kit features detailed photo construction plans in easy-to-follow stages, programming in BASIC with instructions and tips on fine-tuning. The key design elements are fully explained providing you with everything you require to build a robot mouse.

The kit is an ideal introduction into computer control. It demonstrates D&T techniques in easy stages, to aid understanding. The kit offers a simple solution to a complex problem but it requires a degree of competence in soldering. Care will be needed during construction and it is advisable to use a well-lit desk and a fine-tipped  soldering iron.

To complete construction you will require the following:

- Download cable.
- PP3 9v alkaline battery or equivalent.
- Fine tipped soldering iron and solder.
- Hot melt glue gun.
- Contact adhesive or Superglue gel.
- Small ruler and craft knife.
- Wire cutters / small screwdrivers.
- A drop of oil.
- The free program editing software will need to be downloaded from www.picaxe.co.uk

- Micromouse is comparable to many state-of-the-art technologies, from autonomous factory vehicles to planetary exploration. Important skills learned from the construction and control of Micromice can also be directly applied to both current and future car design.

- The maze-solving algorithm used in Micromice finds the shortest route from the current position to a destination and re-routes when it finds a blocked path, similar to the algorithm used in a Sat-navs.

- Parking sensors on cars are becoming commonplace to detect obstacles. Advanced systems go as far as parking the car for you, while many millions of dollars are currently being invested globally in developing totally autonomous (driverless) cars. In 2007, the DARPA Urban Challenge proved the abilities of computer-controlled cars with a first prize of $2 million.

- Battery powered DC motors are now being use in cars, demonstrating the abilities of a non-polluting energy source. In 2006 Tesla Motors announced the production of the Tesla Roadster. It uses Lithium Ion batteries to travel 245 miles on one charge, accelerate from 0-60 in under 4 seconds and give a top speed of 135 mph.

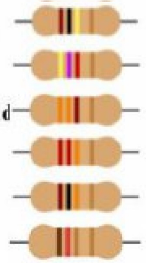
- As part of a training exercise, which could lead to evaluations of potential outposts on the Moon or Mars, Nasa has been sending robots to an isolated rocky polar desert, within a crater in the Arctic Circle. These vehicles are required to be autonomous in case communication is lost.

- Automated guided vehicles are becoming commonplace in a factory environment helping to reduce costs of manufacturing and increase efficiency.

## Components

### Resistor Colours and Values

| | | |
|---|---|---|
| 100k | x6 | Brown Black Yellow Gold |
| 4k7 | x3 | Yellow Violet Red Gold |
| 330r | x2 | Orange Orange Brown Gold |
| 22k | x1 | Red Red Orange Gold |
| 10k | x3 | Brown Black Orange Gold |
| 4.7r | x3 | Yellow Violet Gold Gold |



Push button x3

PP3 battery connector

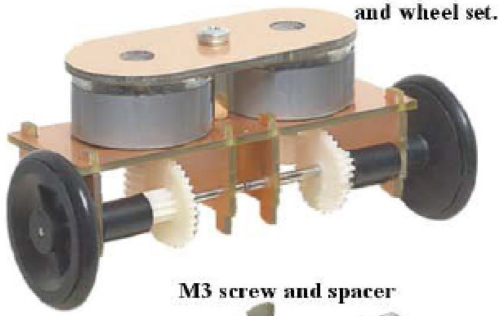
Yellow 3mm led x1
Green 3mm led x2
Red 3mm led x2

IR 5mm leds x3

Low dropout
Voltage
Regulator

1A Diode 3x

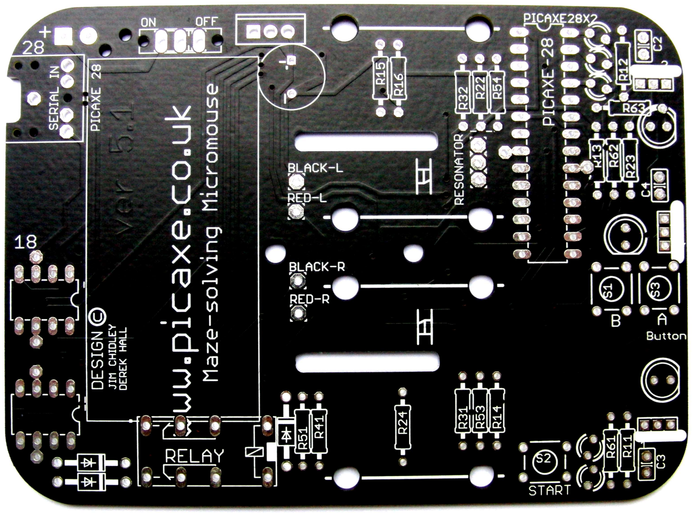
Twin motor, worm gear box
and wheel set.



Phototransistor
x2

Light to voltage
converter x3

M3 screw and spacer



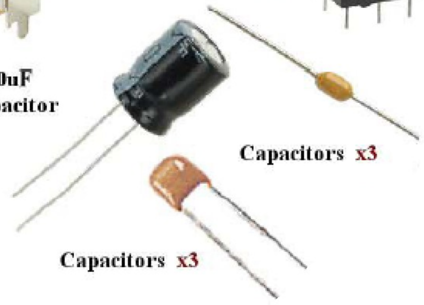
16mhz resonator

Slide switch

Drawing pin x2

Relay 5V
DPDT

1000uF
Capacitor

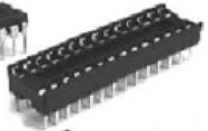


Capacitors x3

Capacitors x3

CD -

3.5 Download jack socket



Picaxe 28x2
IC holders 28 pin

Mosfet Driver chip x2
IC holder 8 pin x2

IC holders 18 pin

**Stage 1**

Assemble the motor and gearbox components as detailed below. The steel axles will need to be each cut down to 42 mm in length with a hacksaw. Take special care when positioning the counters.

Fig 1
Twist the motors into place to fit flush.

Fig 2
Note the worm gears are not pushed tight against the motors, allow about a 3mm gap.

Fig 3
Make sure the wheels are not pushed too tightly against the side assembly.

Fig 4
A small piece of white card, with a heavy black print or black marker pen dividing it in half, is then glued to the worm gear, using super glue gel or contact adhesive. This is then trimmed to be the same width as the worm gear.

Fig 5
The black and white discs form the wheel counters and should be as close to half and half as possible to ensure the wheel counts are even.

Fig 6
Temporally push the motor gear assembly into position through the PCB and screw it into place. At point "A" the worm gears should have about 1mm clearance from the PCB. To adjust place a small flat head screwdriver at point "B" and slowly prise the gear down. The small IR phototransistor receives pulses as the motor turns, both on and off pulses are counted by the processor.

Fig 1

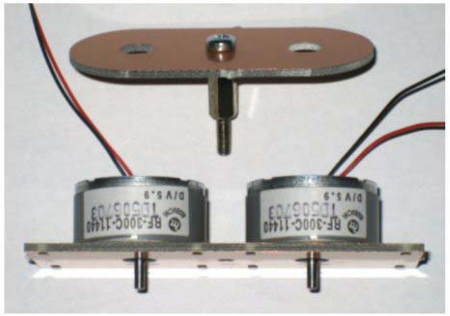


Fig 2

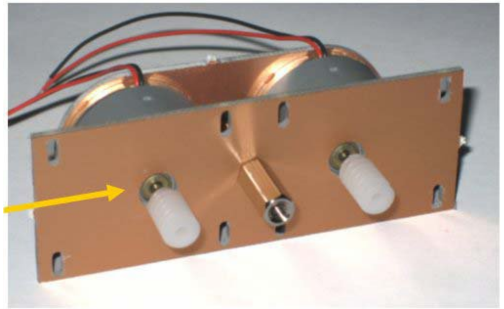


Fig 3

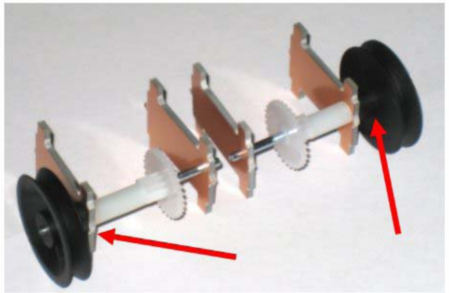


Fig 4

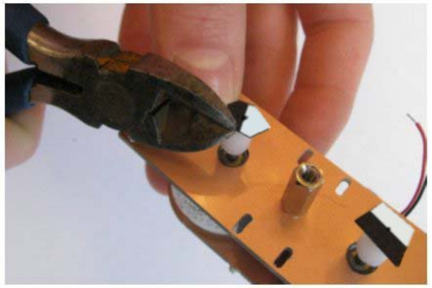


Fig 5

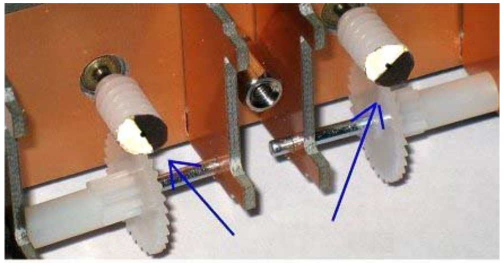
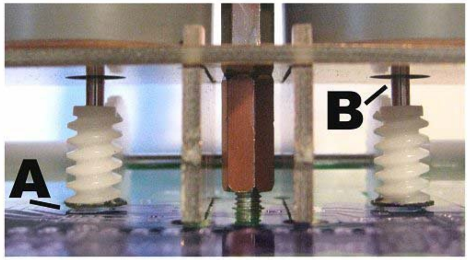


Fig 6



Remember to allow for the internal play in the motor when setting the distance at point ”A”. As the motor stops and starts the worm gear will move up and down very slightly, this may rub the foil off on the PCB causing the wheel counters to fail and your mouse to turn continually.

## Stage 2

Use a fine tipped soldering iron and a well lit desk. Keep a damp sponge by the soldering station and use it to wipe off excess solder. Do not allow globs of solder to remain on the tip. A soldering iron with excess solder on the tip will cause a poor soldering job and the globs may come off while you're soldering, ruining the connection. Take care when soldering static sensitive components. Always use a grounded soldering iron and avoid wearing polyester or acetate clothing, as these tend to develop large amounts of static. An antistatic wrist strap is the most effective way of eliminating static discharge. Try and solder the components neatly and as close to the board as possible and remember to check orientation when prompted. Be careful with the amount of solder you use as too much may run and connect across pads. If you make a mistake use a solder sucker to remove any excess solder.

Solder in the following components:

### Fig 1

- IC HOLDERS x3. Check the orientation by matching the notch in the socket with the notch in the outline.
- BATTERY CONNECTOR. Cut the wires to 45mm in length and pass the wires through the holes. Check the orientation, red wire goes to the square positive pad.
- DIODES x3 Check the orientation with the white outline. These components will help prevent electrical noise created by the motors affecting the processor.

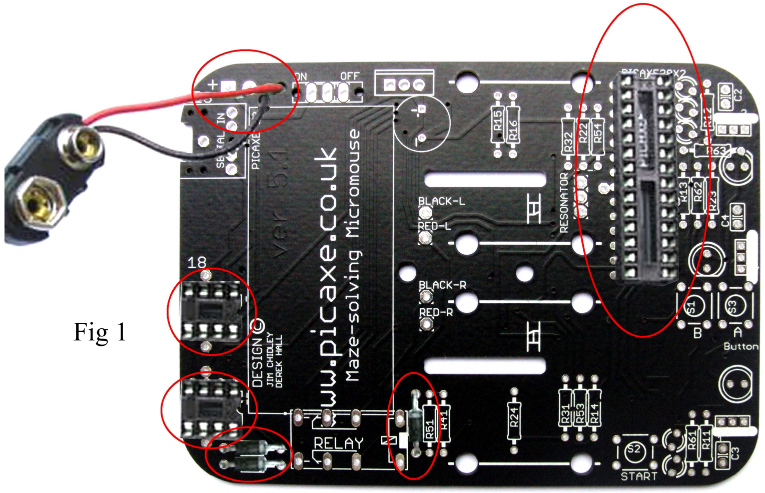


Fig 1

### Fig 2 Turn board over

- 1uf CAPACITORS x3. C5,C7 and C8 connect across the power pins on the three ICs, to smooth out any power spikes.
- Light to Voltage Converters x3. Check the orientation and be aware that these are static sensitive. Use minimal solder as the pads are very close, it is advisable to inspect the solder pads after soldering to unsure they are not bridged.
- Phototransistor x2. Check the orientation and be aware that these are static sensitive. The two shiny areas face towards the board with the chamfered corner orientated as shown on the outline. Centralize and hold in place whilst you solder to anchor one of the legs. Take your time soldering all four legs on each phototransistor and inspect the joint to ensure they are not bridged.

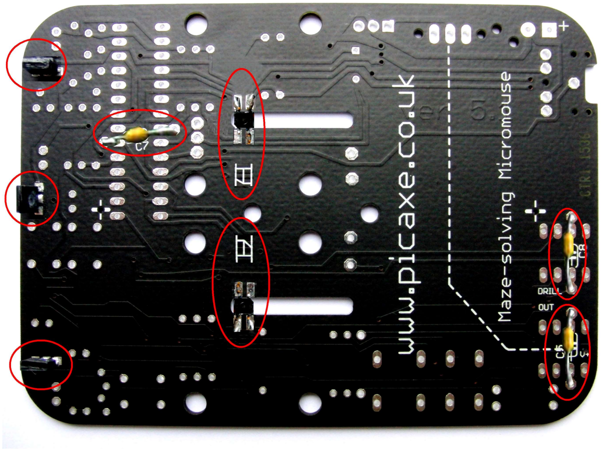


Fig 2

**Fig 3**
Be careful to align all 18 resistors through the correct component holes.

| | | |
|---|---|---|
| **R11,R12,R13,R14,R15,R16** | 100k | x6 Brown Black Yellow Gold |
| **R22,R23,R24** | 4k7 | x3 Yellow Violet Red Gold |
| **R31,R32,** | 330r | x2 Orange Orange Brown Gold |
| **R41** | 22k | x1 Red Red Orange Gold |
| **R51,R53,R54** | 10k | x3 Brown Black Orange Gold |
| **R61,R62,R63** | 4.7r | x3 Yellow Violet Gold Gold |

- Add the 16mhz resonator.

**Fig 4**

- ON/OFF Slide switch
- Voltage regulator. Check orientation, this component is also static sensitive.
- Large 1000uf Capacitor. Check orientation, the grey strip is negative and goes as shown.

Fig 4

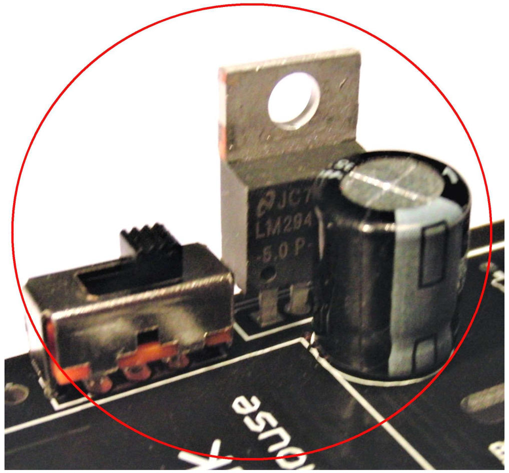


Fig 3

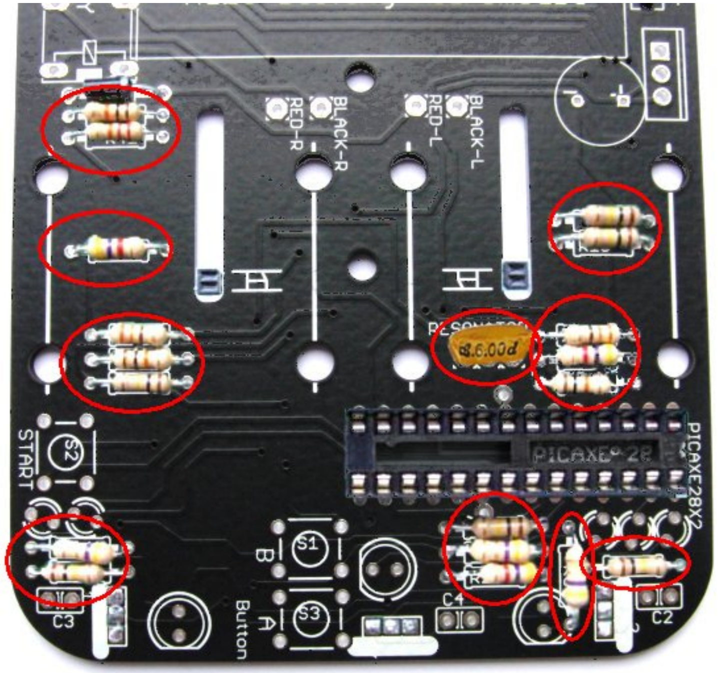


**Fig 5**

- Capacitors x3 C2,C3,C4
- Push buttons x3 button “A”, “B” and START

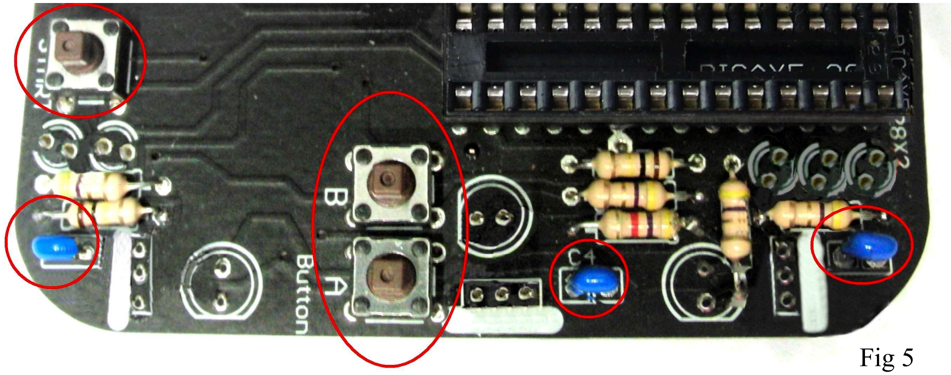


Fig 5

**Fig 6**

- Colour LEDs Red x2, Green x2, Yellow x1 with built in resistors. Check the orientation (the shorter leg is the cathode and connects to ground, use the pictures as a guide)

**Fig 7**

- IR LEDs x3.
  Bend the LED legs to fit as shown, do not have the LED's protruding passed the edge of the PCB. Notice the cathode is the short leg and has a chamfer on the base of the LED.

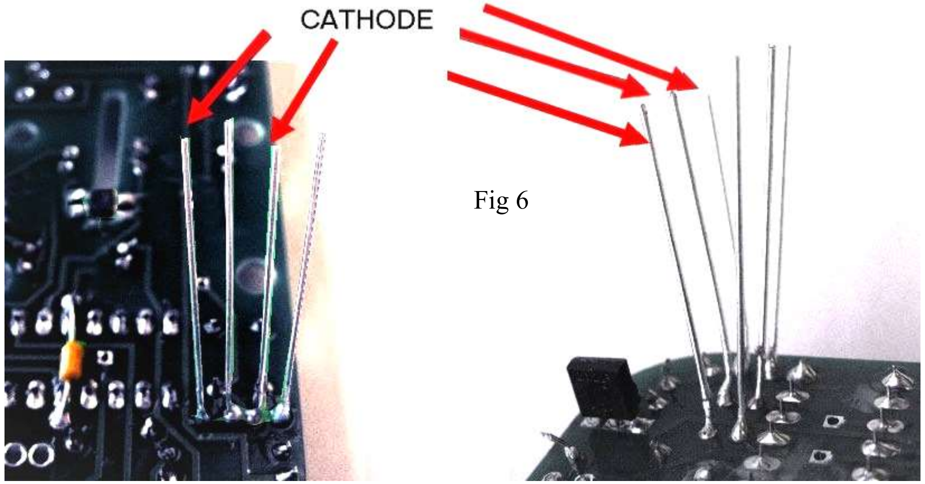


Fig 6

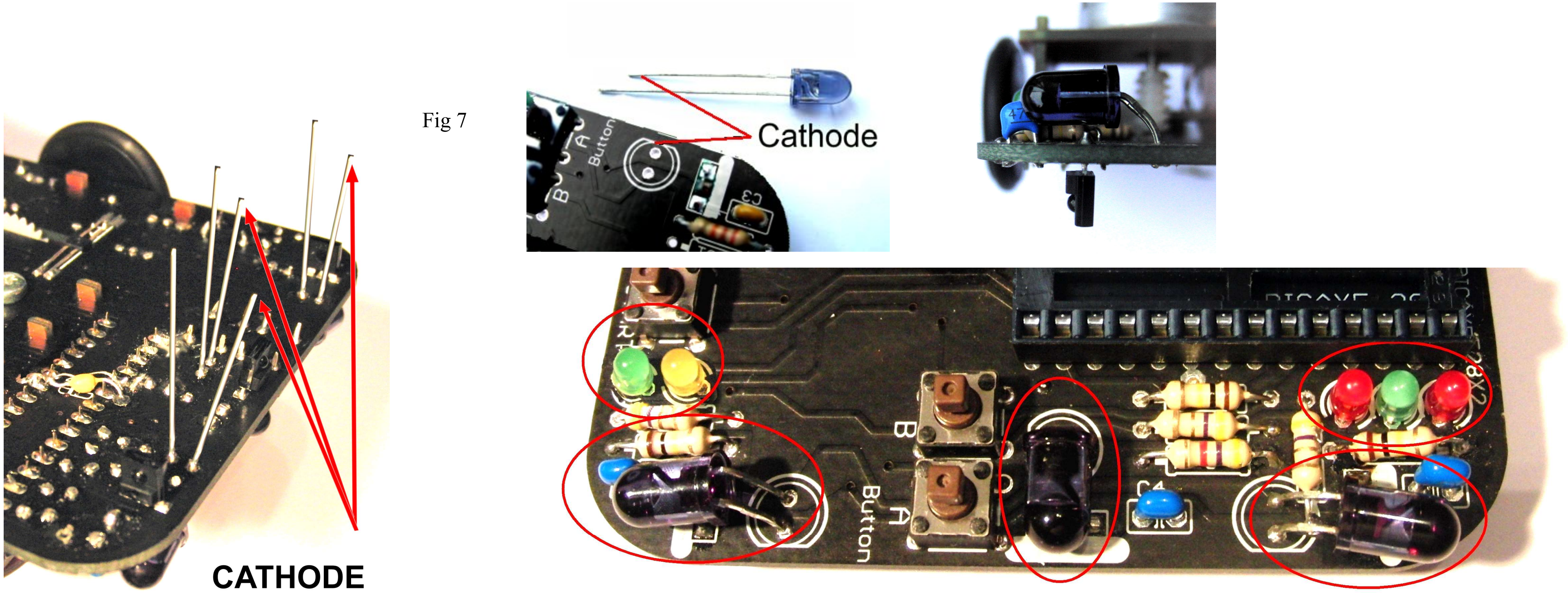


Fig 7

**Fig 8**

- Mount the motor / gearbox assembly. Align the holes up and screw in to place. Be careful not to over tighten the screw as this will prevent the gears running smoothly.

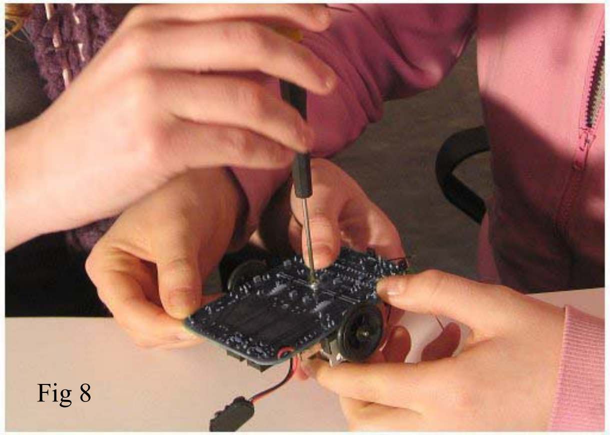


Fig 8

**Fig 9**

The 3 chips can now be added.

Check orientation and be careful as these components are static sensitive.

- Picaxe 28 chip
- Driver chips x2
- Solder the motor wires as shown.

| | |
|---|---|
| BLACK-L =Left motor black wire | RED-L =Left motor red wire |
| BLACK-R =Right motor black wire | RED-R =Right motor red wire |

- M3 Plastic screw peg. This will secure the battery when in place.
- Relay. (the right hand motor is reversed by the relay, allowing the mouse to turn on the spot)
- Jack socket.

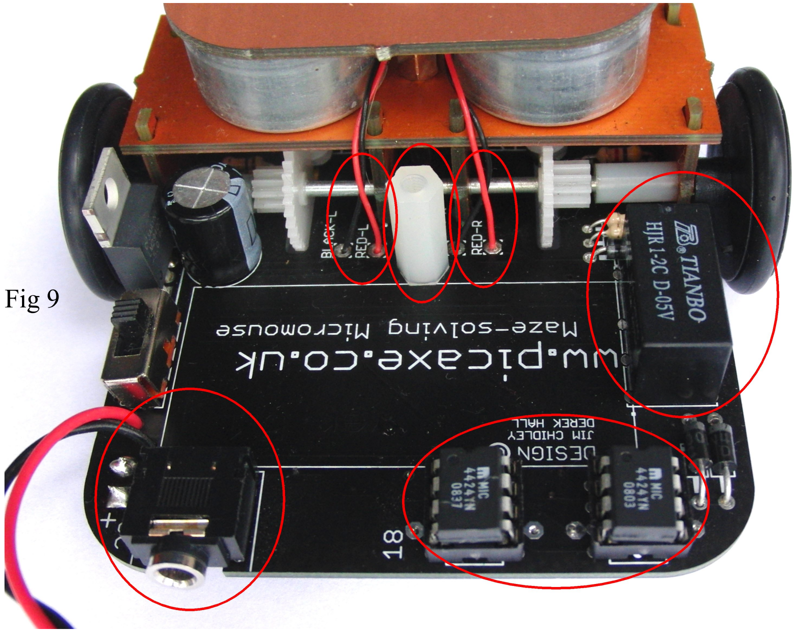


Fig 9

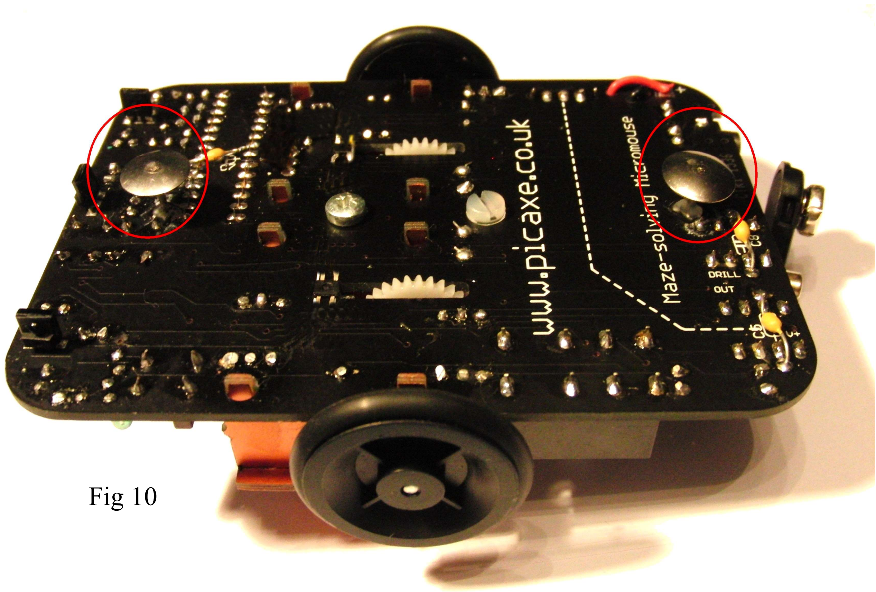


Fig 10

**Fig 10**

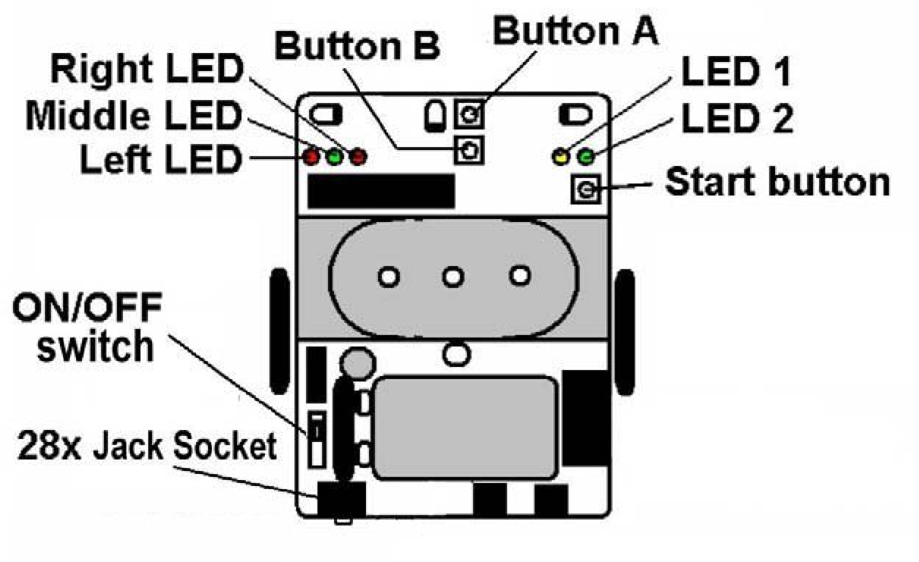
- Drawing pins x2. Attach the front drawing pin first ensuring the light to voltage converters are at lest 2mm from the ground to help reduce unwanted IR reflection from the maze base. also take care not to short circuit any component legs. Locate the small holes (without pads) for position, the point of the drawing pin should be slightly below the top surface of the PCB, push the drawing pins in allowing your mouse to rock very slightly by 1 or 2 mm, additional securing may be added using a glue gun.

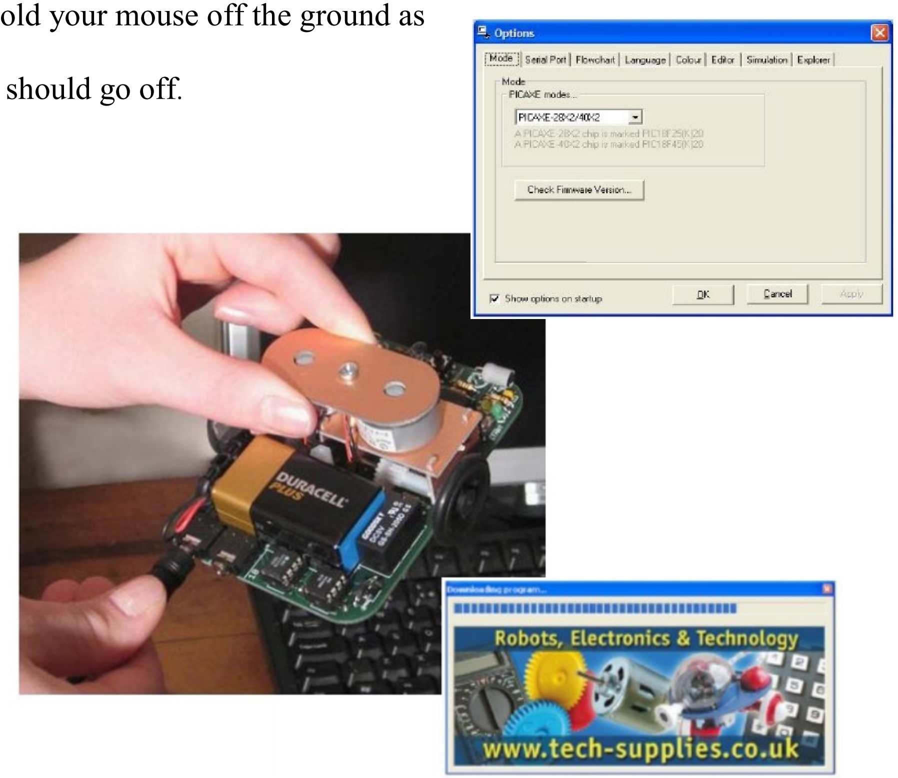
**Test routines**
There are 2 test programs and 1 final program on the CD-rom supplied.

Work your way through the tests ensuring components work and the manoeuvres are as accurate as possible. If you require any more information on any of the commands select HELP/BASIC COMMANDS in the Picaxe programming editor.

**Hardware diagnostic Test 1**

- Load up Picaxe program TEST 1 into the Picaxe programming editor on your computer.
- Check 28X2 is selected in VIEW / OPTIONS.
- Switch on your mouse.
- Insert download cable into the jack socket.
- Press F5 to run program. (Whilst downloading the program into Picaxe, hold your mouse off the ground as the motors may turn at random).
- All five LED's should come on, then LED 1 (yellow) and LED 2 (green) should go off.




Options

Mode | Serial Port | Flowchart | Language | Colour | Editor | Simulation | Explorer

Mode

PICAXE modes...

PICAXE-28X2/40X2

A PICAXE-28X2 chip is marked PIC18F25(K)20
A PICAXE-40X2 chip is marked PIC18F45(K)20

Check Firmware Version...

☑ Show options on startup

Downloading program...

Robots, Electronics & Technology
www.tech-supplies.co.uk

**TEST 1** is designed to test that all the elements of control and sensing are working correctly. By following the instructions below you will highlight any construction errors that may have occurred. Run the test mode only for a few minutes at a time, as the voltage regulator may start to get hot and your battery life will suffer.

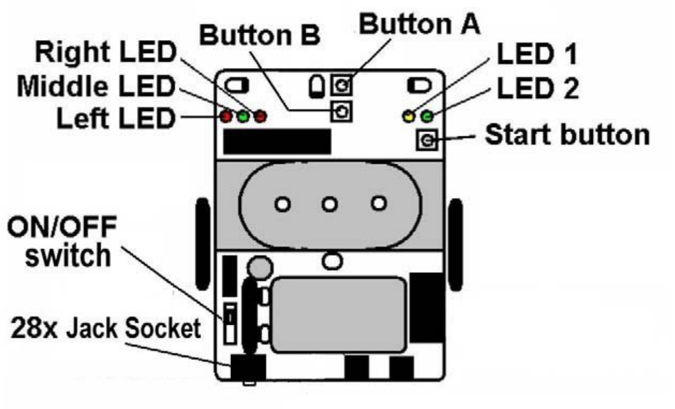
- **Testing buttons "Start" button and infrared LEDs and sensors.** Press button "Start" button. LEDs. Left, middle and right go off. The corresponding LEDs will now light up if an object is brought close enough to be detected by the left, middle and right wall sensors.

- **Testing motors and left wheel counter.** Hold your mouse off the ground and press the "Start" button. Both wheels turn in a forward direction. The left LED flashes on and off every 20 left wheel counts. Moving the worm gear away from the sensor will increase the time the LED is off, try to make the on/off time similar.

- **Testing right wheel counter.** With your mouse still off the ground press the "Start" button. Both wheels turn in a forward direction. The right LED flashes on and off every 20 right wheel counts. Moving the worm gear away from the sensor will increase the time the LED is off, try to make the on/off time similar.

- **Testing relay to reverse right wheel.** With your mouse still off the ground press the "Start" button. The right wheel reverses for half a second then continues forward for half a second and then repeats.

- **Testing button "A".** Press button "A" to stop the motors. Press button "B"(reset) to restart the test sequence.



**Simple faultfinding (part 1)**

- Ensure the correct Picaxe is selected during programming in VIEW / OPTIONS.

- Use a resistance meter to check your soldering for dry joints and solder bridges between track or pads. Check electrical components have been assembled correctly and that component legs are cut short and not touching each other.

- Check gear and wheels are not rubbing against circuit board or gear assembly.

- Check the battery charge using a voltmeter and ensure all components are correctly orientated.

**Simple faultfinding (part 2)**

- Check worm gear is not too far away from phototransistor and that the phototransistor is aligned. Also check the black and white card is still attached to the worm gear. You can use the ENCODER TEST PROGRAM or modify the test program to keep the pwmout at 0 instead of 500, and reduce “if b3=10” to “if b3=2 “. You can now manually turn the worm gear slowly with a matchstick.

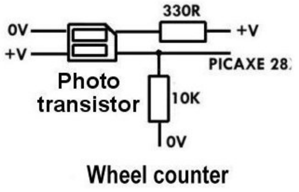
- The wall sensors can pick up a TV remote. Point it directly at each sensor. During the infrared LED test if you press a button on the remote the LED’s corresponding to the left, middle or right sensor will light up.

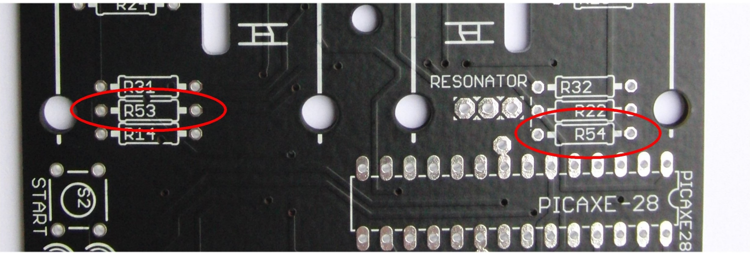
- Use a video camcorder with night vision to view infra-red emitters during the infrared LED test.

If you find it very difficult to make the wheel counters work, make sure the black and white disc is as close to the PCB as possible and not getting caught on the drive gear.

An additional solution for inconsistent wheel counts is to decrease the 10k resistor to 6k8. Making this resistor smaller will increase the range of the sensor.R54 for the left counter and R53 for the right counter.

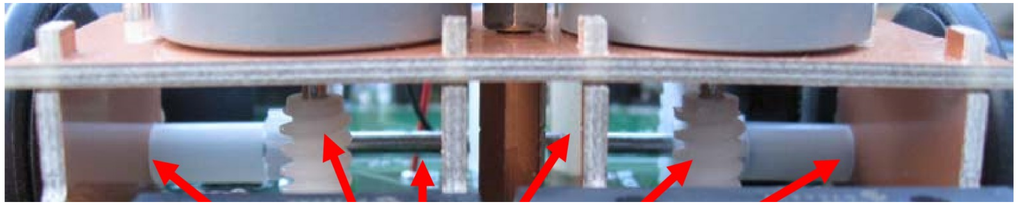
The brightness of the internal IR led could also be increased by reducing the 330r resistor to 100r.





Once your mouse is working correctly, a very tiny drop of oil may be applied to the worm gear and axle holes to smooth meshing. Use a small flat screwdriver to apply the oil while the motors are turning.

Be careful not to damage the gears or use too much as this may cause your wheel counters to come unstuck. Regular cleaning should be maintained to keep the gears free from any dust and hair.

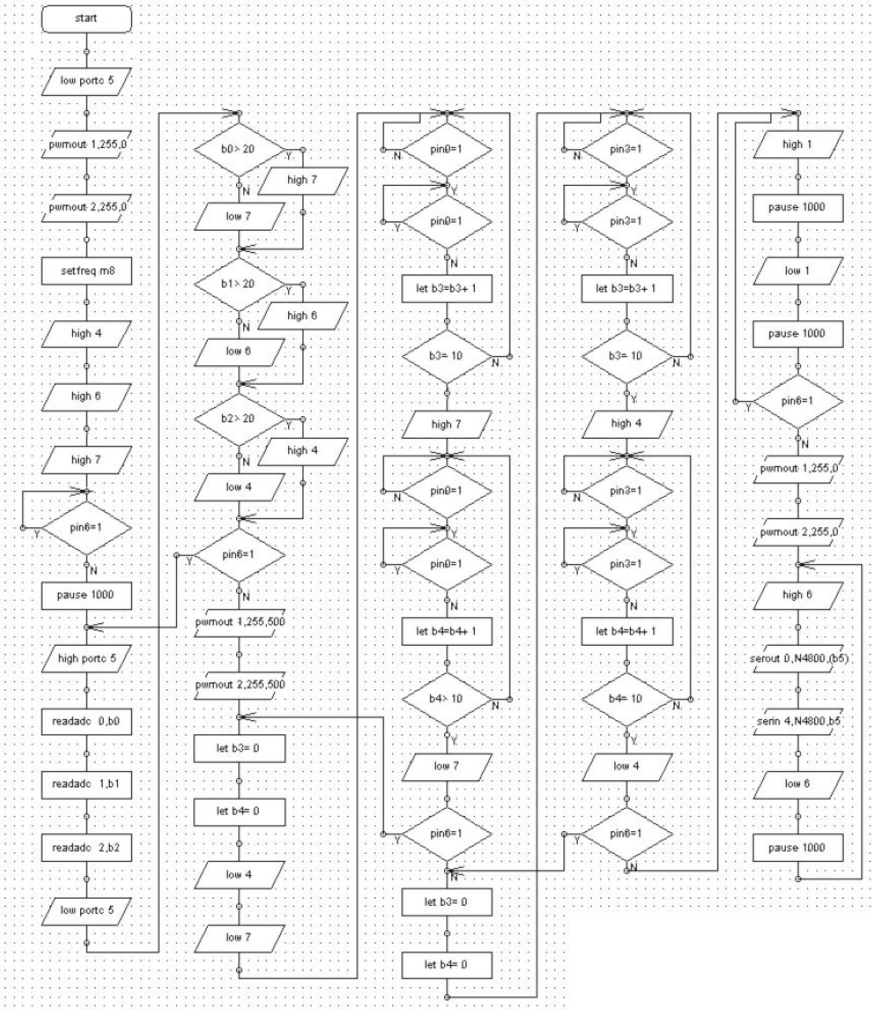


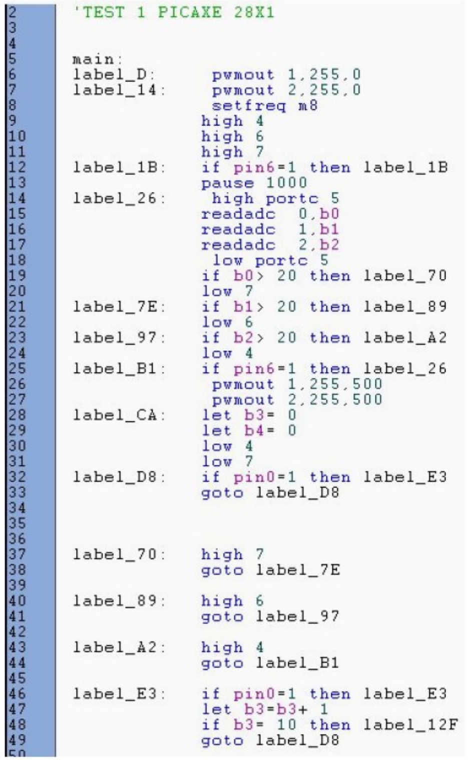
Put a tiny drop of oil on the gear and where the axle goes through the holes.

Below is **Test 1** in Flow chart form and its Basic equivalent. For a novice programmer this will help as a reference to understanding how the commands work in controlling your mouse.

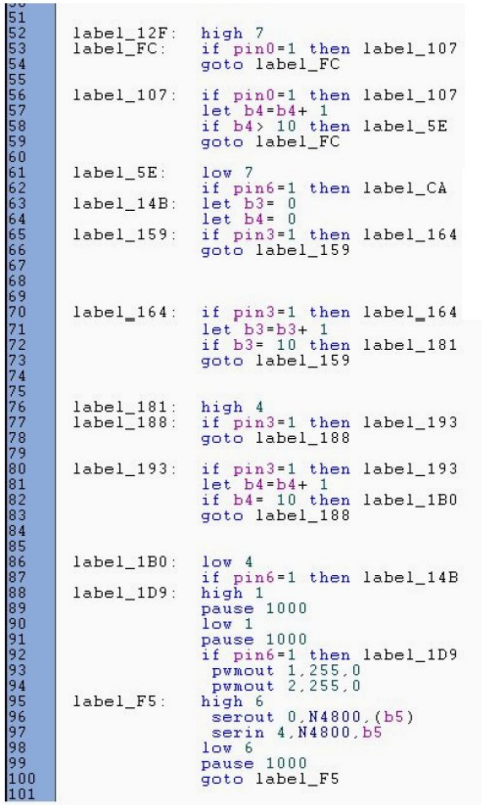
Once all the test are completed your mouse will be ready to run. It is important to remember that calibration of the straightening values and front sensor reading may need to be adjusted if you run your mouse in a maze with different wall coatings (painted or plastic) or under very different lighting conditions.

**TEST 1**
**Picaxe 28X1**
flowchart





```
'TEST 1 PICAXE 28X1


main:
label_D:        pwmout 1,255,0
label_14:       pwmout 2,255,0
                setfreq m8
               high 4
               high 6
               high 7
label_1B:      if pin6=1 then label_1B
               pause 1000
label_26:       high portc 5
               readadc  0,b0
               readadc  1,b1
               readadc  2,b2
                low portc 5
               if b0> 20 then label_70
               low 7
label_7E:      if b1> 20 then label_89
               low 6
label_97:      if b2> 20 then label_A2
               low 4
label_B1:      if pin6=1 then label_26
                pwmout 1,255,500
                pwmout 2,255,500
label_CA:      let b3= 0
               let b4= 0
               low 4
               low 7
label_D8:      if pin0=1 then label_E3
               goto label_D8



label_70:      high 7
               goto label_7E

label_89:      high 6
               goto label_97

label_A2:      high 4
               goto label_B1

label_E3:      if pin0=1 then label_E3
               let b3=b3+ 1
               if b3= 10 then label_12F
               goto label_D8
```

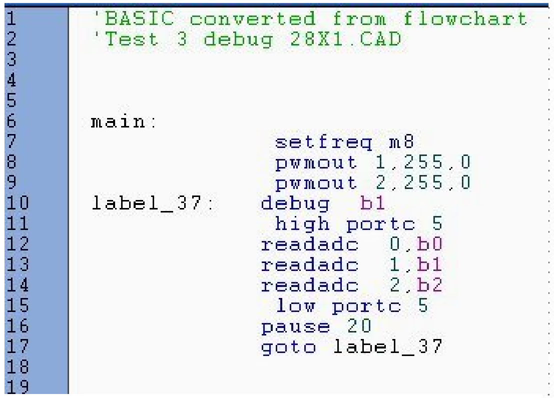


```

label_12F:     high 7
label_FC:      if pin0=1 then label_107
               goto label_FC

label_107:     if pin0=1 then label_107
               let b4=b4+ 1
               if b4> 10 then label_5E
               goto label_FC

label_5E:      low 7
               if pin6=1 then label_CA
label_14B:     let b3= 0
               let b4= 0
label_159:     if pin3=1 then label_164
               goto label_159



label_164:     if pin3=1 then label_164
               let b3=b3+ 1
               if b3= 10 then label_181
               goto label_159


label_181:     high 4
label_188:     if pin3=1 then label_193
               goto label_188

label_193:     if pin3=1 then label_193
               let b4=b4+ 1
               if b4= 10 then label_1B0
               goto label_188


label_1B0:     low 4
               if pin6=1 then label_14B
label_1D9:     high 1
               pause 1000
               low 1
               pause 1000
               if pin6=1 then label_1D9
                pwmout 1,255,0
                pwmout 2,255,0
label_F5:      high 6
                serout 0,N4800,(b5)
                serin 4,N4800,b5
               low 6
               pause 1000
               goto label_F5

```

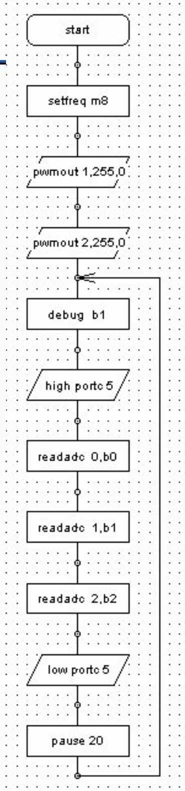
**TEST 2** will help determine the values needed for each sensor to keep your mouse central and detect when a wall has disappeared.

- Load up Picaxe program TEST 2 debug into the Picaxe programming editor.
- Check 28X2 is selected in VIEW / OPTIONS.
- Switch on your mouse.
- Insert download cable into 28X jack socket and leave connected.
- Press F5 to run program. (Whilst downloading program into Picaxe hold your mouse off the ground as the motors may turn at random)

**Test 2** Flowchart and Basic conversion

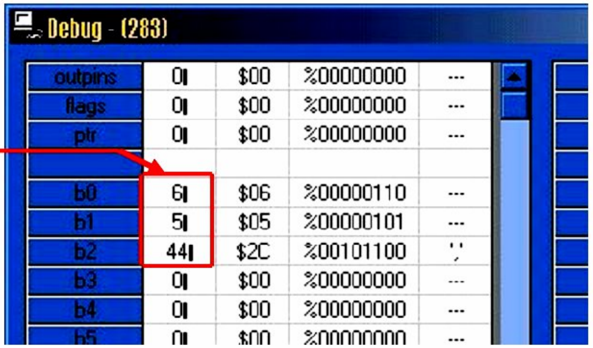


```
'BASIC converted from flowchart
'Test 3 debug 28X1.CAD



main:
             setfreq m8
             pwmout 1,255,0
             pwmout 2,255,0
label_37:   debug  b1
             high portc 5
            readadc  0,b0
            readadc  1,b1
            readadc  2,b2
             low portc 5
            pause 20
            goto label_37
```



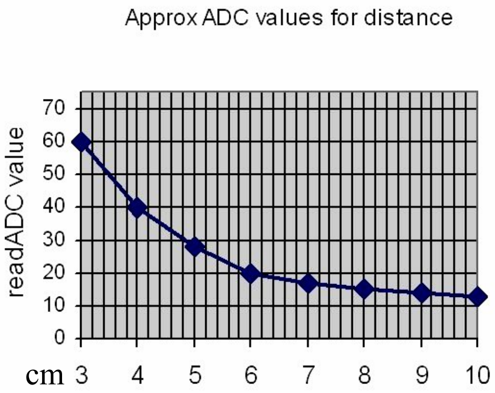
## Sensor calibration

With the download cable still connected and the power switch on ,you will notice the variables on the debug screen will vary as the sensors detect an object.



```
outpins  0   $00  %00000000  ---
flags    0   $00  %00000000  ---
ptr      0   $00  %00000000  ---

b0       6   $06  %00000110  ---
b1       5   $05  %00000101  ---
b2      44   $2C  %00101100  ','
b3       0   $00  %00000000  ---
b4       0   $00  %00000000  ---
b5       0   $00  %00000000  ---
```

readadc 0 Left sensor = b0
readadc 1 Front sensor = b1
readadc 2 Right sensor = b2



As you move the wall within the range of the sensors you will notice that the ADC values are not linear. The distance achieved between low readings is much greater than with higher reading. Saturation will be reached at a value 156. Stray infrared light reflecting off the black floor or through the PCB can give a low value, so when setting up try and avoid setting levels below 7.

Place your mouse on the matt black maze base and position a wall as shown and record the values from the five simple tests. When moving the wall closer the sensor will increase the relevant variable in the debug program.

The setup values for your sensors will need to be recorded. Use a list as below to record your values as you complete the tests. These will need to be entered into your main control program. The values vs. distance will vary from sensor to sensor this is ok as the sensors are not matched during production.

Record your variables from the tests the following tests.

| | | |
|---|---|---|
| fig 1 | left_straighten | = Value of left sensor when no straightening is required |
| fig 1 | right_straighten | = Value of right sensor when no straightening is required |
| fig 2 | left_wall | = Sets distance for left wall detection |
| fig 3 | right_wall | = Sets distance for right wall detection |
| fig 4 | front_wall | = Sets distance for front wall detection |
| fig 4 | reset_frontwall | = Detects a close front wall and resets wheel counter |

fig 1

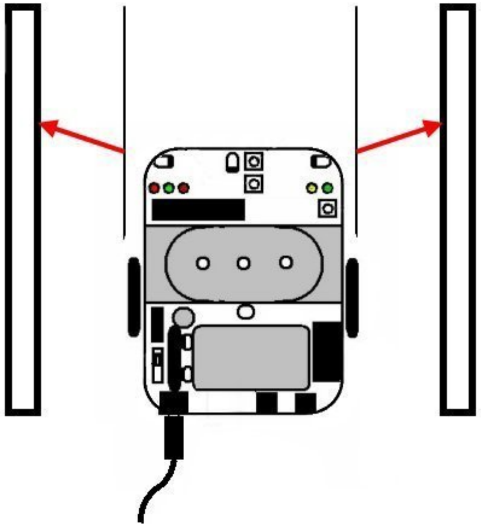


fig 2

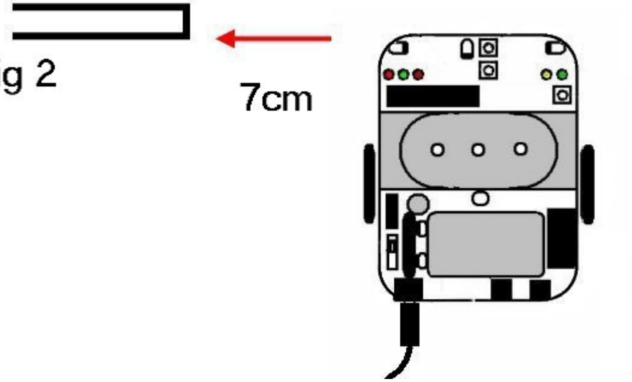


fig 3

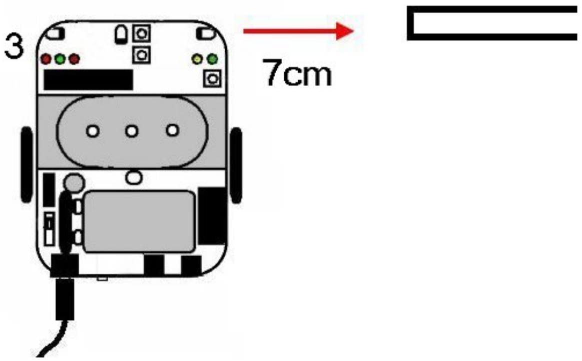
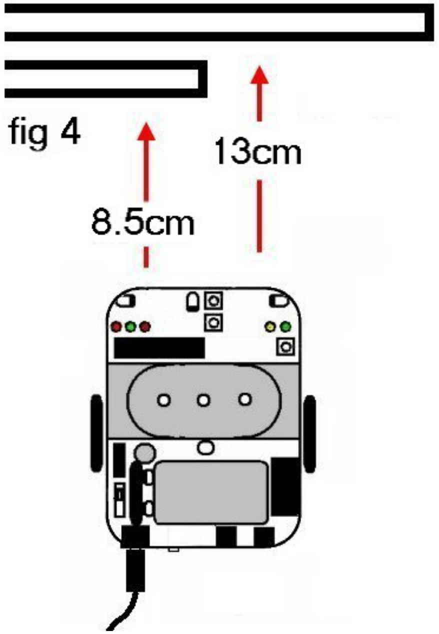


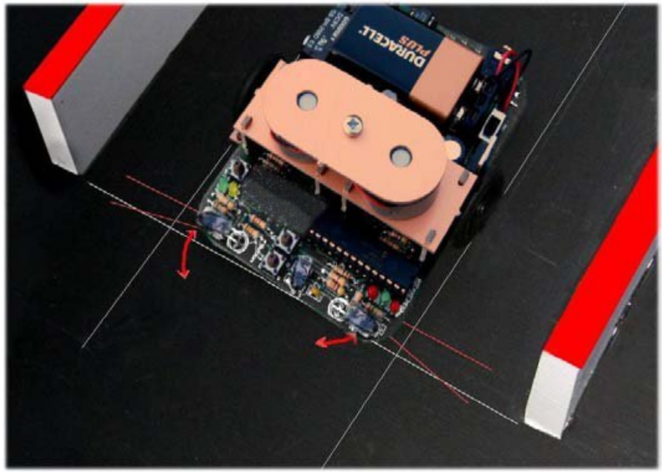
fig 4



Setting up the point when the side sensors no longer can detect a wall is done by lining up the front of you mouse level with the point the walls end.

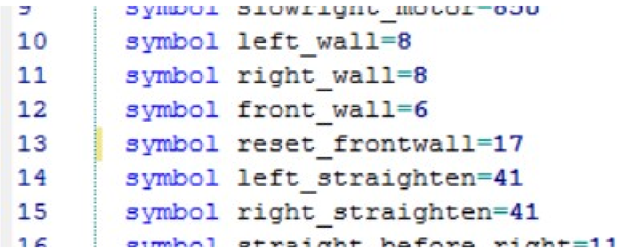
Now carefully bend the LEDs in the direction shown, so that the debug values for b0 and b2, drop by 50% as they see the end of the post. Both sides should see the wall end at the same time. This is so that whilst travelling straight, the wheel counter can reset at the same point using either wall. As a wall disappears the straight-line count is reset to 43 allowing a further travel of 18 counts before stopping to map the walls.

Having your side sensors pointing slightly forward will give you a sharper image as the wall disappears. The sharper image will prevent your mouse trying to straighten briefly on a wall as it disappears. Be careful of having your sensors pointing to far forwards as this may give an uneven distance-reset depending on how straight your mouse is at this point.



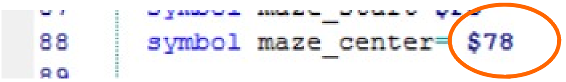
**Enter your set-up values**

Adjust the values at the start of the Picaxe Micromouse program, to the setup values you determined in Test 2 and save your program. These values will be unique to your mouse.

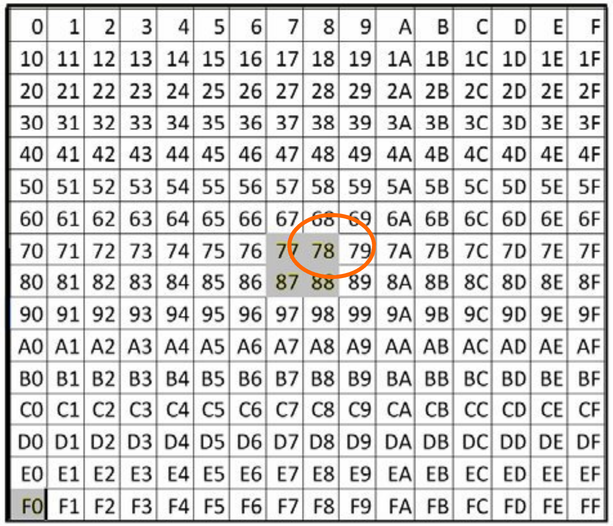


```
symbol slowright_motor=650
symbol left_wall=8
symbol right_wall=8
symbol front_wall=6
symbol reset_frontwall=17
symbol left_straighten=41
symbol right_straighten=41
symbol straight_before_right=11
```

Line 88 sets the maze_centre (the target cell).
Using the graph on the right you can change this cell to solve a smaller maze.



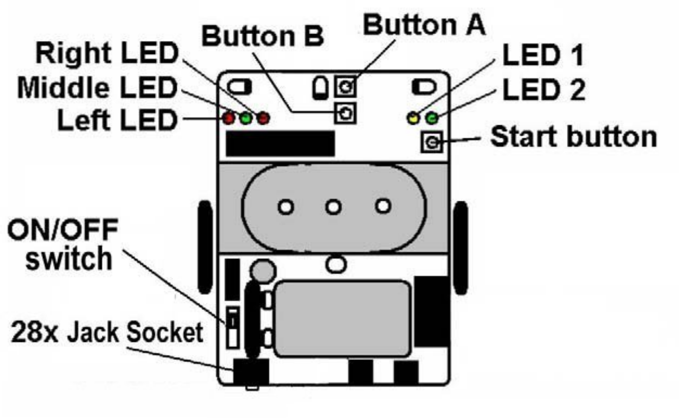
```
87    symbol maze_start=$F0
88    symbol maze_center= $78
89
```



| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 2A | 2B | 2C | 2D | 2E | 2F |
| 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 3A | 3B | 3C | 3D | 3E | 3F |
| 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 4A | 4B | 4C | 4D | 4E | 4F |
| 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 5A | 5B | 5C | 5D | 5E | 5F |
| 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 6A | 6B | 6C | 6D | 6E | 6F |
| 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 7A | 7B | 7C | 7D | 7E | 7F |
| 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 8A | 8B | 8C | 8D | 8E | 8F |
| 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 9A | 9B | 9C | 9D | 9E | 9F |
| A0 | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | AA | AB | AC | AD | AE | AF |
| B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 | B8 | B9 | BA | BB | BC | BD | BE | BF |
| C0 | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | CA | CB | CC | CD | CE | CF |
| D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | DA | DB | DC | DD | DE | DF |
| E0 | E1 | E2 | E3 | E4 | E5 | E6 | E7 | E8 | E9 | EA | EB | EC | ED | EE | EF |
| F0 | F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 | F9 | FA | FB | FC | FD | FE | FF |

You should have now changed the values at the start of your program and saved this copy.

- Switch on your PICone micromouse.
- Insert the download cable into the jack socket.
- Press F5 to run program.



**Resetting the Picaxe chip**
Always reset your mouse when starting in a new maze. With the power switch on hold down Button B then hold down Button A. Release Button B then release Button A. All LEDs will rapidly blink after each other, leaving only the Right, Middle and Left LED's on. Your mouse is now ready to run.

**Soft reset**
While your mouse is running, you can press Button B at any time to stop at any time. Your mouse can then be restarted in the start cell whilst retaining its maze map.

**LED 1&2**
The yellow LED 1 indicates Picaxe 28X2 is busy solving, the green LED 2 indicates the centre is found and the mouse is now heading back to the start. All LEDs slowly flash when the maze is deemed unsolvable.

**Straight-line set-up**
Set up a long straight corridor and after resetting your mouse, position it ready and press the Start Button. There will be a pause for half a second before your mouse starts. Your mouse will follow the corridor straightening on the left hand wall and the Right, Middle and Left LEDs indicate the walls your mouse is detecting. As your mouse reaches the end of the corridor pick it up and press Button B only. Your mouse will now retain the maze map it has created. Remove the walls, press start and check if your mouse travels in a straight line without looking to map walls.

Pulse Width Modulation (PWM) allows the Picaxe to continue running the program whilst the motors are constantly powered at a controlled speed in the background. As it will be highly unlikely that both motors are evenly matched, fine-tuning will be necessary by adjusting the values of the PWM motor outputs. These variables are found at the start of the Picaxe Micromouse program (they are set at a default level of 1000), adjust these and reload the program into your 28X2 Picaxe until your mouse can travel in a straight line. Record these levels and re-save your copy. The highest value that can be used is 1023.

Typically the PWM outputs may vary by up to 10%. If you find you have to vary the PWM outputs by a vast amounts to ensure a straight line is achieved. Check the worm gears are not rubbing against the PCB or the wheels are not pushed too tightly against the gear assembly.

## Fine tuning your mouse

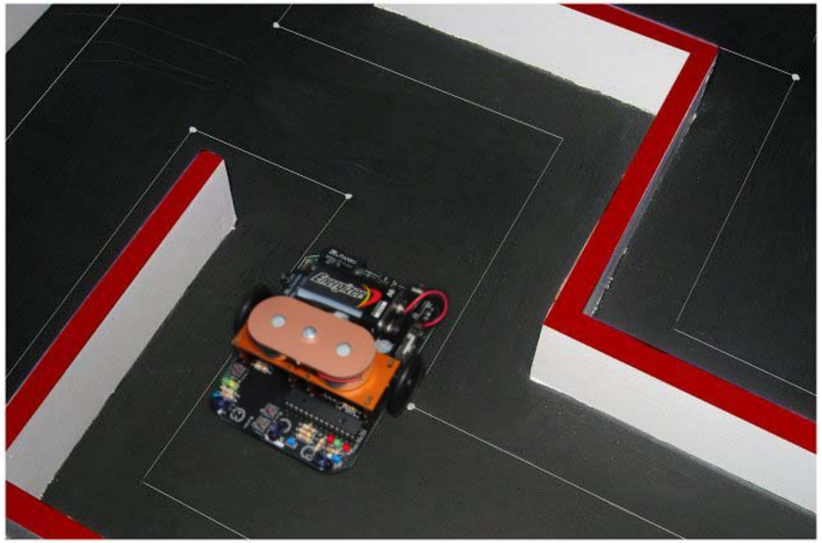
This stage is the most important and possibly the least recognised. Even a precision made mouse with an incredibly fast processor will struggle to navigate a simple maze without a considerable amount of time spent fine-tuning. By keeping this mouse design simple it will aid you immensely during your fine-tuning process, as there are with fewer variables to consider. Setting your mouse to complete simple basic manoeuvres using minimal sensor readings will help identify poor incorrect settings.

With you mouse retaining the long corridor maze map, you set in the previous page. Your mouse should run down the centre of the corridor irrespective of which wall it is straightening on, you can test this by putting alternate walls on either side of the corridor. At this point you may want to tweak the values of your “left_straighten” or “right_straighten” in your program.

At this point a good tip is also to video your mouse, running down a long corridor where both sidewalls end at the same point. On playback in slow motion both Right and Left LEDs should go off simultaneously. Make adjustments to the IR LEDs if necessary then permanently glue them in place, using a heat glue gun.

## Testing manoeuvres  through a maze

As your mouse moves through the maze it can be very difficult to spot any slight turning or straightening errors. The default sensor values should only need minor adjustment to work with unpainted MDF wall, painted walls will be more reflective. Make sure your maze base is painted mat black, ideally black board paint. This is to minimise ambient light affecting your optical sensors. Draw parallel pencil lines through each square along the line the wheels will take when your mouse is centred. Put white correction fluid dots where the lines cross. When the inside wheel locks to turn the mouse the wheel should sit directly on the white target dot. As you make your mouse run faster a video camera will aid with set up and exact turn alignment.



If your mouse turns unexpectedly check to see if the corresponding LED was lit to indicate a wall was present. Use alkaline batteries for your mouse; a new battery should last in excess of 30 minutes. As the battery runs down the IR LEDs lose power so your mouse will start to move closer to the wall it is straightening on (default left wall), it may also not be able to sense a front wall. Your mouse will also run slower.

**Adjustable variables for fine tuning** (These are found at the start of the program)

| | |
|---|---|
| left_motor | = 'sets left motor speed for straight line (higher number travel faster max 1023) |
| right_motor | = 'sets right motor speed for straight line (higher number travel faster max 1023) |
| left_wall | = 'sets distance for left wall detection (lower number greater distance) |
| right_wall | = 'sets distance for right wall detection (lower number greater distance) |
| front_wall | = 'sets distance for front wall detection (lower number greater distance) |
| reset_frontwall | = 'detects a close front wall and resets wheel counter (for long dead ends or zig zags) |
| left_straighten | = 'value of left sensor when no straightening is required |
| right_straighten | = 'value of right sensor when no straightening is required |
| straight_before_right | = 'sets distance of the short straight before right turn (higher number travel further) |
| angle_right | = 'sets amount of right turn (higher number turns further) |
| straight_after_right | = 'sets distance of the short straight after right turn (lower number travel further) |
| straight_before_left | = 'sets distance of the short straight before left turn (higher number travel further) |
| angle_left | = 'sets amount of left turn (higher number turns further) |
| straight_after_left | = 'sets distance of the short straight after left turn (lower number travel further) |
| straight_turnround | = 'distance travelled before a turn round (higher number travel further) |
| angle_turnround | = 'sets amount of turn round (higher number turns further) |

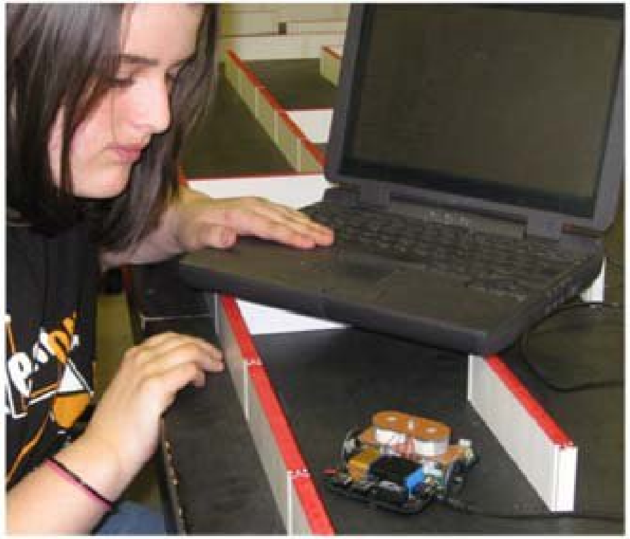
The outputs and inputs pins and variables used in the Main Control programme for reference are listed Below:

**28X2 Program Outputs**

| | |
|---|---|
| Right wheel | pwmout 1 |
| Left wheel | pwmout 2 |
| Relay | out 1 |
| Led right side red | out 4 |
| Led middle green | out 6 |
| Led left side red | out 7 |
| Inferred Leds | out C.5 |

**28X2 Program Inputs**

| | |
|---|---|
| Left wheel counter | pinc. 0 |
| Right wheel counter | pinc. 3 |
| Start button | pinc. 6 |
| Left sensor | adc 0 |
| Front sensor | adc 1 |
| Right sensor | adc 2 |



If your mouse is not running exactly down the centre of a corridor, it will become very unreliable when faced with a series of turns one after another. Remember as you adjust the variables they will all affect each other (changing the position your mouse at the start of a manoeuvre will cause your mouse to finish the manoeuvre at a different point).

Your mouse is typically looking for the end of a post to reset the wheel counter, remember this exact point may vary slightly depending on the reflectivity of the post.

To help with more reliable dead-ends adjust the straight_turnround variable to go as deep into a dead-end as possible without hitting the wall.

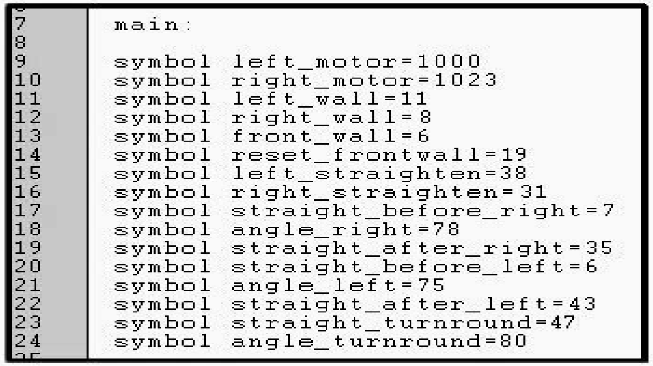
Taking some time to fine tune your mouse on your maze will make a tremendous difference with the performance. By adjusting the variables by one or two then rerunning you will easily be able to spot errors in your settings.

Set-up the turn values in order the following order:
1. straight before the turn,
2. angle
3. straight after the turn.

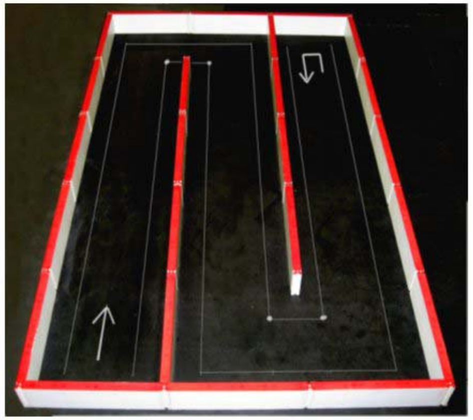
Try to make your mouse turn with the inside wheel on the white dot, when your mouse stops to make a decision, the wheels should be halfway between the end of the post and the white dot.

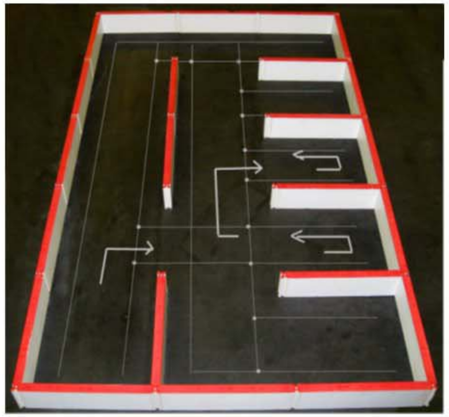
**Here are some maze set-ups that will test your mouse to the full.**

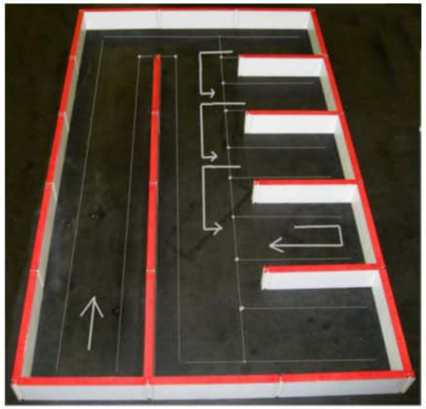
The left and right variables will probably not be symmetrical as many factors can cause variations in values, as you can see from this typical listing used by "PICone" shown below.

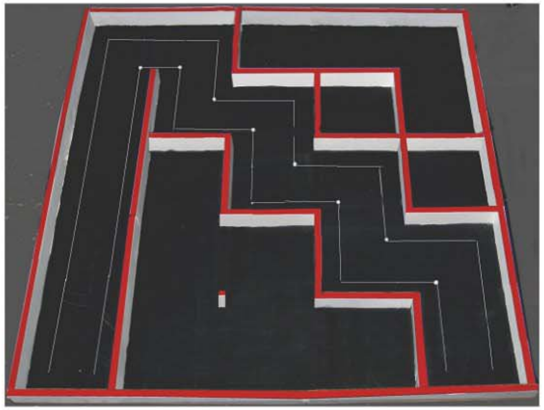


```
main:

symbol left_motor=1000
symbol right_motor=1023
symbol left_wall=11
symbol right_wall=8
symbol front_wall=6
symbol reset_frontwall=19
symbol left_straighten=38
symbol right_straighten=31
symbol straight_before_right=7
symbol angle_right=78
symbol straight_after_right=35
symbol straight_before_left=6
symbol angle_left=75
symbol straight_after_left=43
symbol straight_turnround=47
symbol angle_turnround=80
```









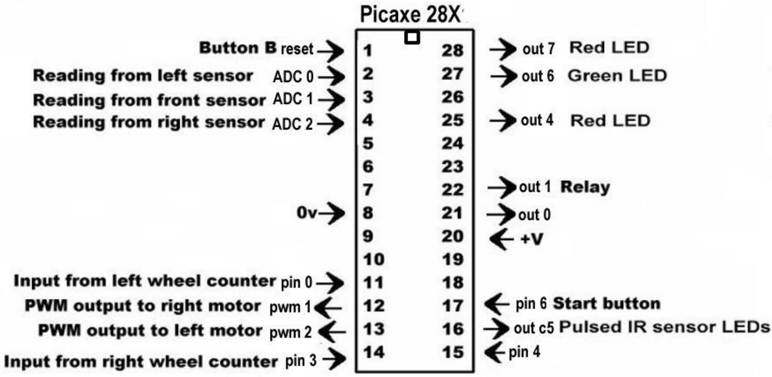
Do not expect your mouse to complete these mazes without considerable fine-tuning first. The last three mazes shown are extremely difficult and some competitive mice may have difficulty with these layouts.

If your mouse constantly turns left when the first left hand turn is attempted, check the right hand wheel counter is still attached. If your mouse constantly turns right when it attempts its first right hand turn, check the left hand wheel counter is still attached. If your mouse travels into a wall check the corresponding LED was lit to indicate a wall was detected, adjusting the wall sensing value to a lower number will help.

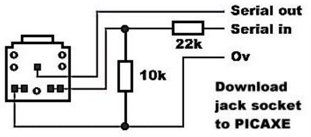
After time your gears may wear and your axels may run more smoothly, so you should occasionally re-check your set up.

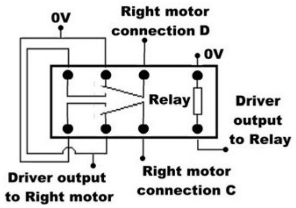
Now you have completed your mouse there are many options open for you to increase speed, through simple hardware modification or by enhancing the program, this is up to you.

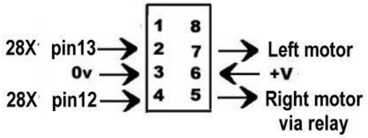
The following information is to help
assist you with any changes you choose to make.

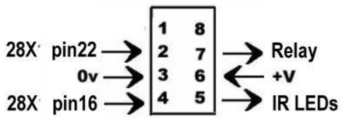


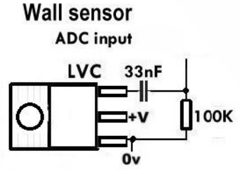
Picaxe 28X
Button B reset 1
Reading from left sensor ADC 0 2
Reading from front sensor ADC 1 3
Reading from right sensor ADC 2 4
5
6
7
0v 8
9
10
Input from left wheel counter pin 0 11
PWM output to right motor pwm 1 12
PWM output to left motor pwm 2 13
Input from right wheel counter pin 3 14
28 out 7 Red LED
27 out 6 Green LED
26
25 out 4 Red LED
24
23
22 out 1 Relay
21 out 0
20 +V
19
18
17 pin 6 Start button
16 out c5 Pulsed IR sensor LEDs
15 pin 4









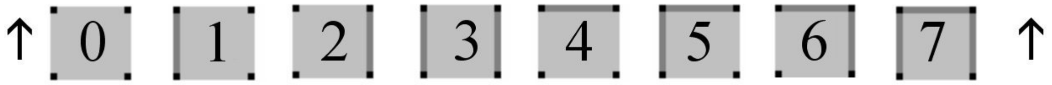


**Wall location and distance correction**

The distance correction system simply creates two variables “wall_config” and “pre_wall_config”. Comparing these wall configuration samples the mouse can look at if the value has reduced. When travelling along a straight the mouse is setup to slightly over shoot each square, but is pulled back inline by resetting as a sidewall disappears, or if a front wall is detected. In the program featured on the disc variables b3 and b4 are used.

After the wheel counter is reset on the edge of a wall, your mouse will count forward and stop at the point it needs to be to turn. A move in any direction is possible from this point. If a left wall is present “wall_config” will =1, if a right wall is present “wall_config” will = “wall_config” +2 and if a front wall is present another 4 is added. This will give you a variable that you can use to plot your maze walls as shown below.

The variable “wall_config” represents the wall configuration with your mouse travelling in the direction shown. Variables 3,5,6 and 7 require no solving decision, so your mouse will not stop to solve.



If you have sped up your mouse, you may find that it twists slightly when it comes to a halt. The resistance though the relay will allow the left motor to start and stop slightly quicker. This can be overcome by making sure the PWM command for the right motor comes first and adding a slight pause of 10 or 20 in the program between both PWM commands.

## RUNNING YOUR MOUSE QUICK START

It is important to remember that calibration of the straightening values and front sensor reading may need to be adjusted, if you run your mouse in a maze with different wall coatings (painted or plastic) or under different lighting conditions.

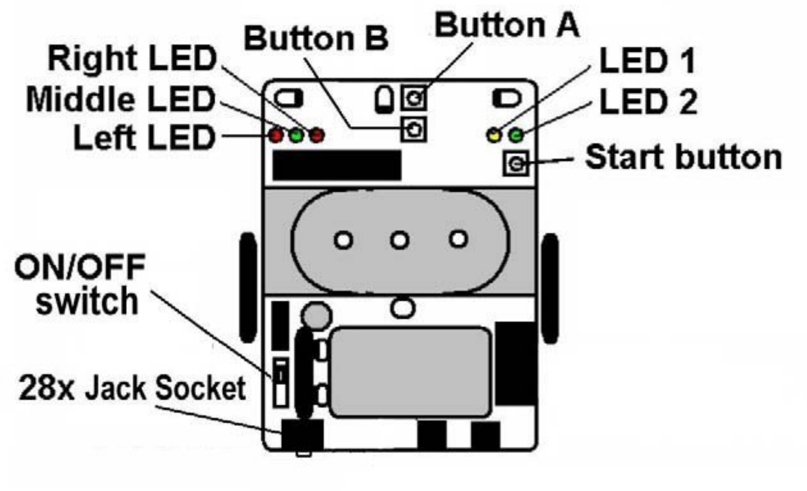
Always do a total mouse reset when starting in a new maze. With the power switch on, hold down Button B then hold down Button A, release Button B then release Button A. All LEDs will rapidly blink after each other, then only Yellow LED comes on then off. When the Right red, Middle green and Left red LEDs stay on your mouse is ready to run. The maze map is now reset, press the "Start" button after a short pause your mouse will start moving.

Whilst running Right LED, Middle LED and Left LED, indicate the walls the mouse can see. LED 2 indicates the centre of the maze is found and the mouse is heading back to the start.

If only one direction is available your mouse will continue on, but when faced with a choice of directions to turn, your mouse may stop and recalculate the maze. When the program has decided which direction to take your mouse will again start moving.

You can stop your mouse at any time by pressing Button B only, this will allow your mouse to be restarted in the start square without loosing its map memory. (Very useful if your mouse crashes during a search run, so long as you press this button before your mouse manages to put in phantom walls.)

Your mouse will try to head to the centre of an assumed 16x16 cell maze, if this is not possible your mouse will stop and both LEDs 1 and 2 flash alternatively to indicate the maze is unsolvable. Your maze map will now need to be reset.



Your mouse will always reset on the end posts, so having a post in the middle of the target cell will make your mouse more reliable when leaving the centre.

Experience has told us that generally your mouse will tend to be more reliable when a new battery has been run briefly this may also loosen up any sticky gears.

With out a total mouse reset, your mouse will retain its maze map even when switched off. Using the Micromouse Maze Editor you can retrieve its maze map at a later time.

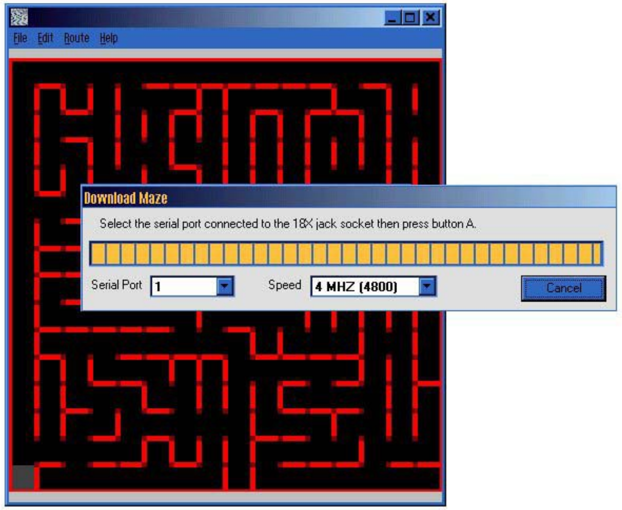
**Micromouse Maze Editor** (included on the CD supplied)

In the Software file open the Maze Editor, start setup to copy files. After installation is complete the program will appear in the programs list. The serial port will default to the one used for the Picaxe Programming editor.

The Micromouse Maze Editor software allows you to create, view and save .MAZ files. To create a maze, open a new maze, then just click to add or remove walls.

Features :

- Viewing popular maze files.
- Create and save .MAZ files or .MAZE files.
- Up load a maze from you Micromouse.
- Export a set maze to your Micromouse.

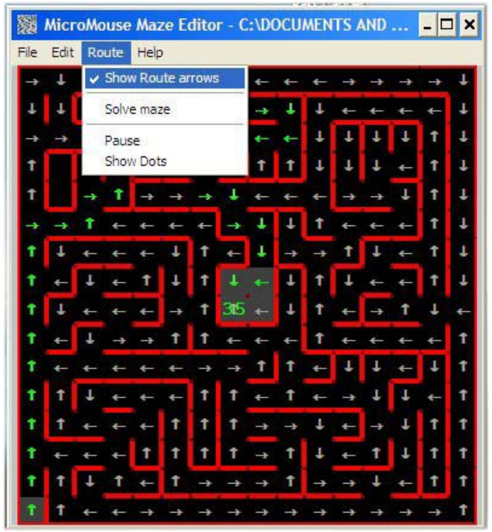


Download Maze

Select the serial port connected to the 18X jack socket then press button A.

Serial Port 1

Speed 4 MHZ (4800)



After your mouse has been run in a maze, you can view a maze it has stored. With your Micromouse turned on connect the Jack lead and click LISTEN FOR MAZE in the FILE MENU, then press Button “A” on your mouse. The maze your mouse has mapped will now be shown.

The uploaded maze will show route arrows from the run-start square; to the target or point the mouse was stopped. This is a very useful tool to ensure your mouse has not missed walls or added phantom ones in, it also allows you to record a competition maze for future reference without having to draw them by hand.

You can also load a set maze, setting the route your mouse will take allowing manoeuvres to be exactly calibrated with no walls. Create a maze then EXPORT MAZE DATA in the FILE MENU, this can then be copied and pasted into Maze Solving program, replacing the blank maze.

An additional feature is the ability to highlight the route to the centre your mouse will take. Select solve maze then show route arrows, these are then shown in green along with the total number of cells travelled.

## Increasing the power to the motors

This hardware modification increases the volts to the motors. First by drilling out the via shown, you isolate the +volts to the motor driver chip. You can then connect the +volts from the 9v battery to the motor driver directly. This will give the motors 9v instead of the regulated 5v, although it is recommended that you regulate the 9v using an adjustable low drop out regulator.

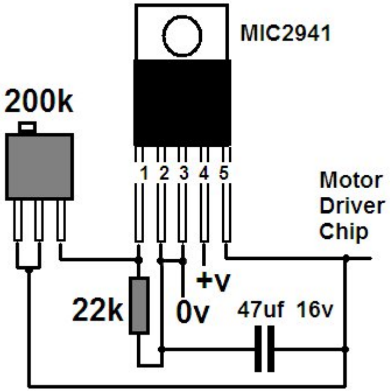
Using the simple circuit shown below you can adjust the voltage used by the motor driver chip.

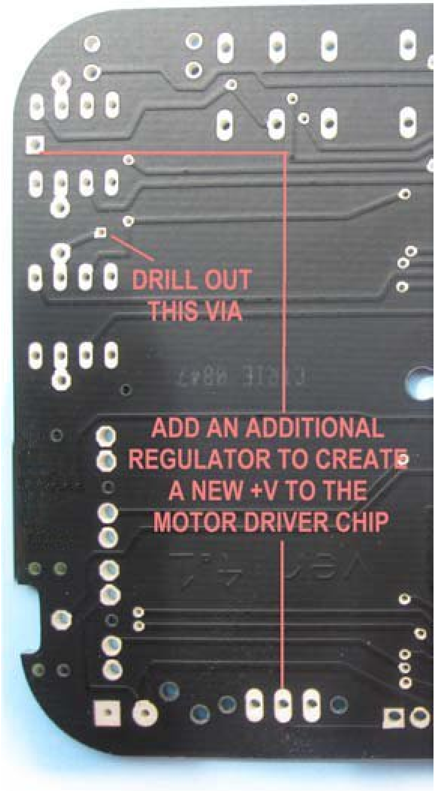
MIC2941

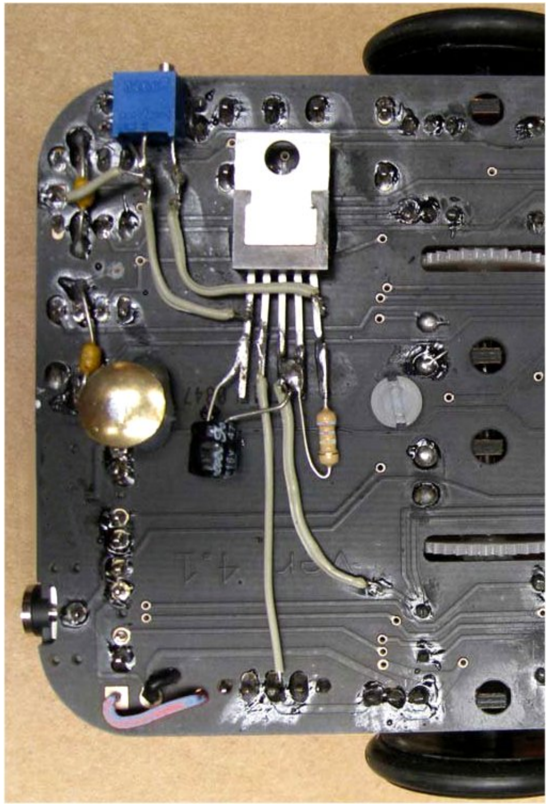
1. Adjust
2. Shutdown
3. Ground
4. Input
5. Output

These components are available from WWW.RAPIDONLINE.COM.

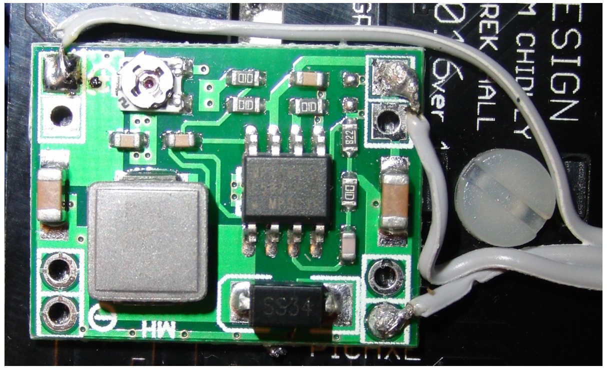
| | |
|---|---|
| Regulator | 82 4020 |
| Pot 200k | 68 0077 |
| Capacitor | 11 1510 |







An alternative is to use a Buck regulator like the one shown below.



This MP1584 Adjustable 3A DC-DC Converter, is a Step Down Buck Voltage Regulator. Is low cost, about the size of a 1p coin and available from various suppliers.

If you choose to try and increase the power to the motors you may also want to consider using Li-Po battery cells.

The neck of a rubber balloon stretched over the wheels will also give you more grip when accelerating and braking.

## MAKE THE KIT UNIQUE

The PICone micromouse kit is designed as an introduction into micromouse. The micromouse competition encourages unique and diverse ways to navigate the maze at speed. By individually modifying your kit it is possible to run your mouse at speeds seen by PICone Turbo. Listed below is the modification we did to reliably achieve those speeds.

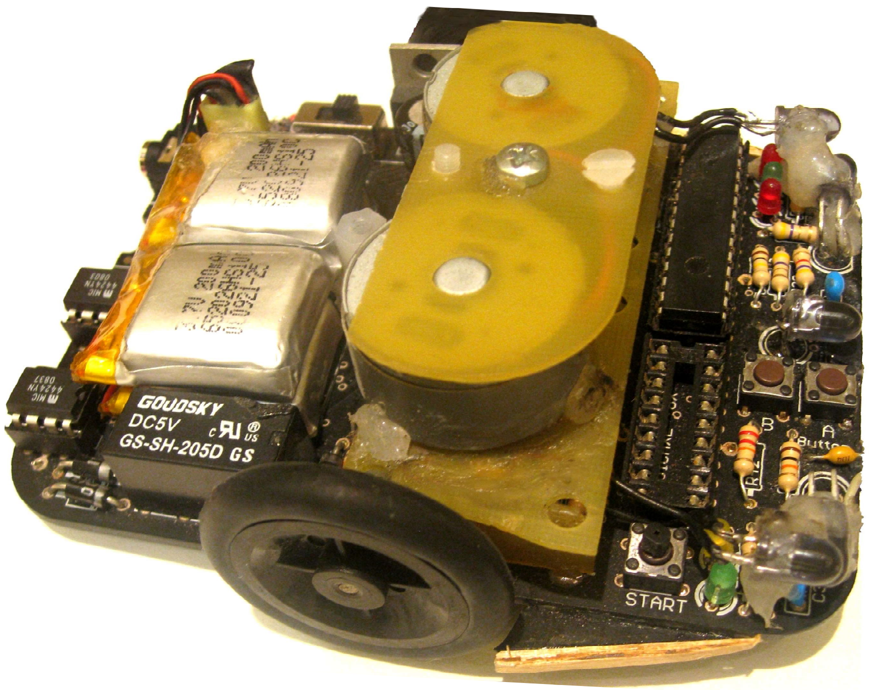
- The TIMER command has been used to count the left wheel counter. This allows the program to check the TIMER reading less often than the actual wheel counter, so the wheel can turn faster without the possibility of missing counts.

- An additional driver chip has been added which powers two IR leds. The new IR leds point forward and are glue gunned on top to the side IR leds. All three forward IR leds can be powered individually and read by the front sensor, this allows PICone Turbo to navigate long comb type maze configurations and diagonals on a speed run.

- The 9v battery has been replace by four 200mah Li-Po batteries cells, two of which power the electronics where as all four power the motors through an additional regulator.

- The gear box has been modified by using a smaller gear cog meshing with the original worm gear and moving the axles slightly forward. The wheel counting discs are also larger and split into quarters, doubling the wheel counts.

- Small guides have been added either side in case the wheels get caught on the end posts whilst navigating diagonals.



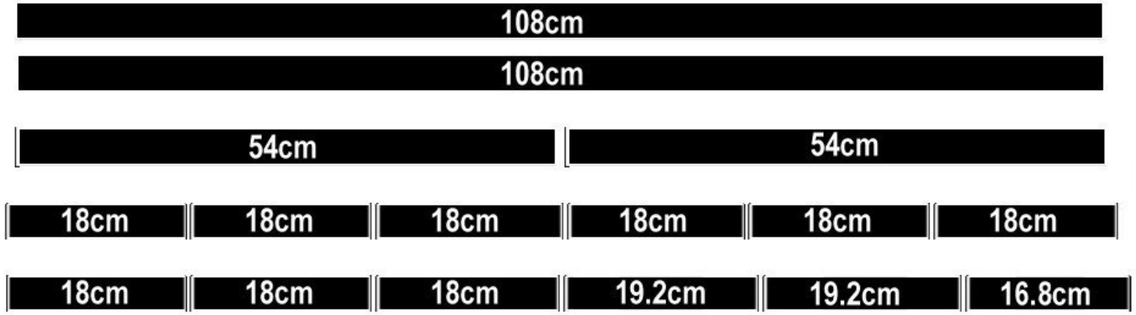
**PICone Turbo**

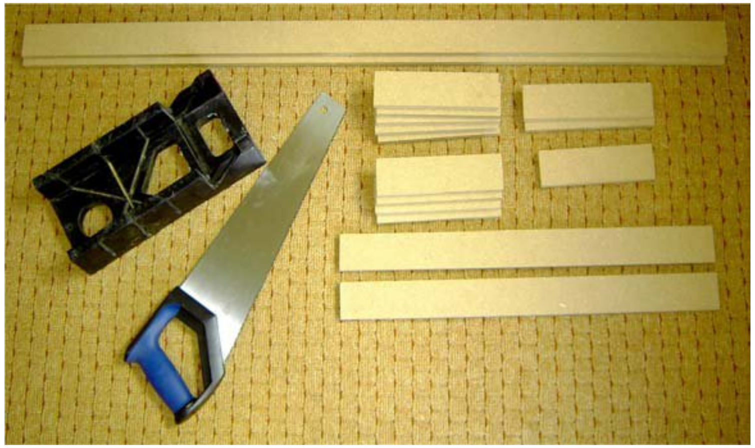
**Create you own DIY micromouse maze**

The following instructions demonstrate how to create a low cost Micromouse maze with a changeable layout.

- Wicks Blackboard paint
- B&Q Hardboard 1220 x 610 x 3mm
- B&Q MDF 1220 x 607 x 12mm
- B&Q Cutting MDF into 5 x 5cm strips

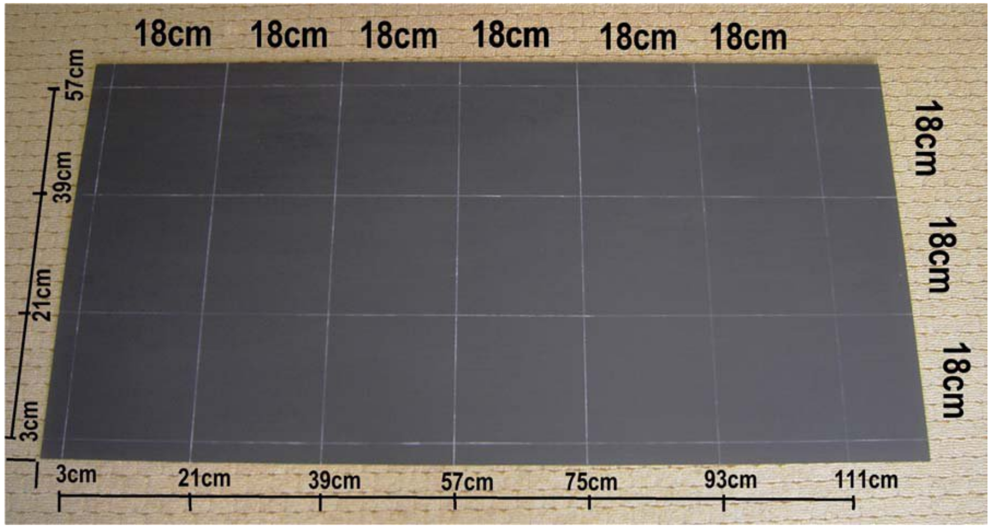
Paint the Hardboard with two coats of blackboard paint and cut the 5 MDF strips as shown below using a mitre block, then lightly sand any rough edges.



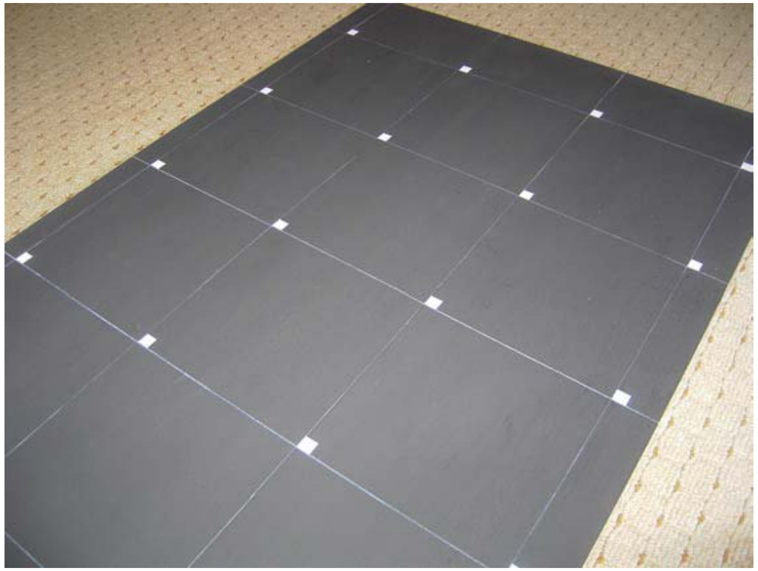


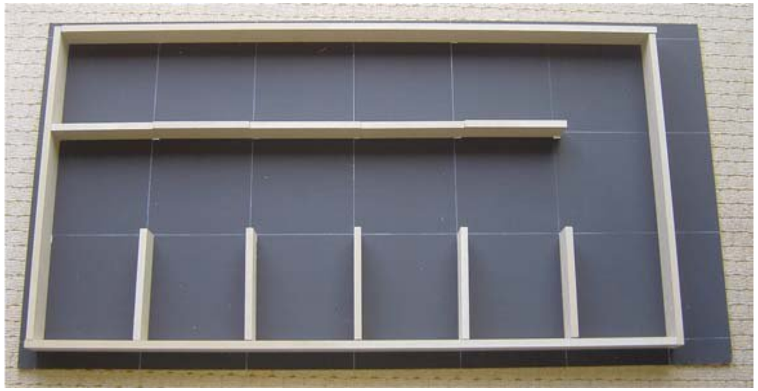
Cut out twenty-eight small 1.2cm x 1.2cm squares of sticky white paper.

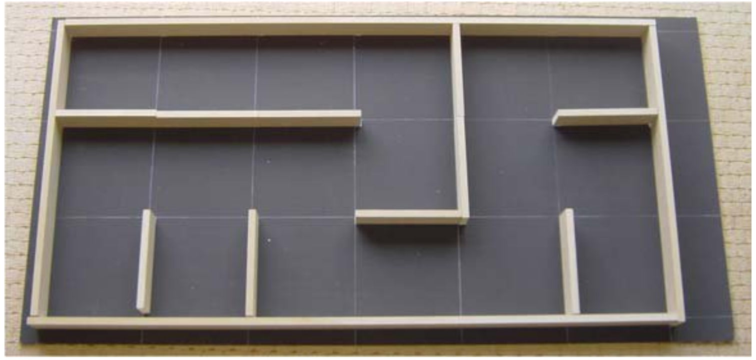
Once the black paint is dry, use a pencil to draw an 18cm grid as shown.



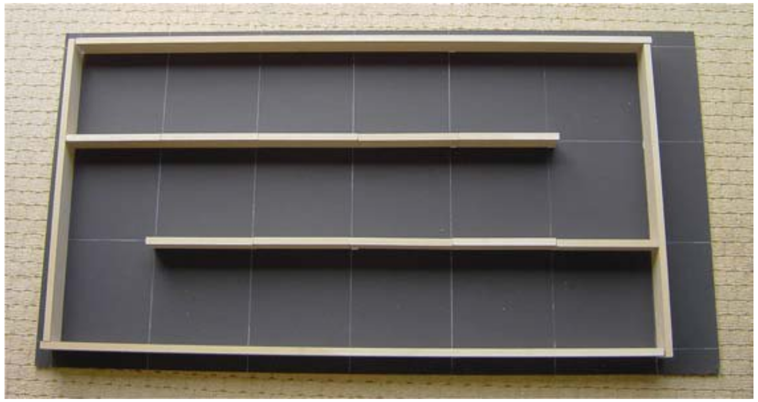
Stick a small white square in the corner at every point the pencil lines cross.



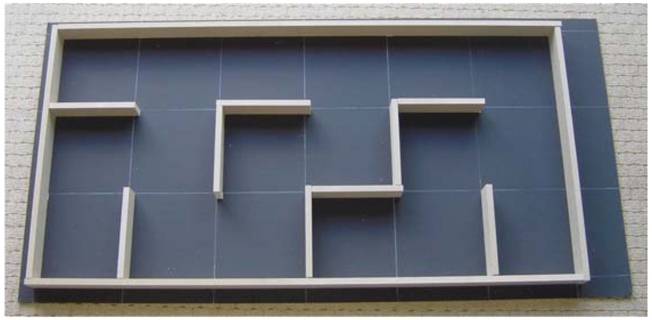
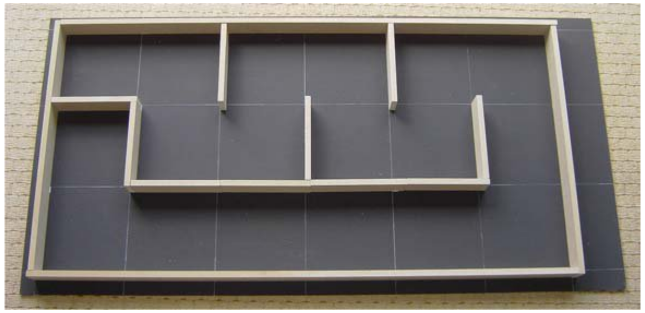
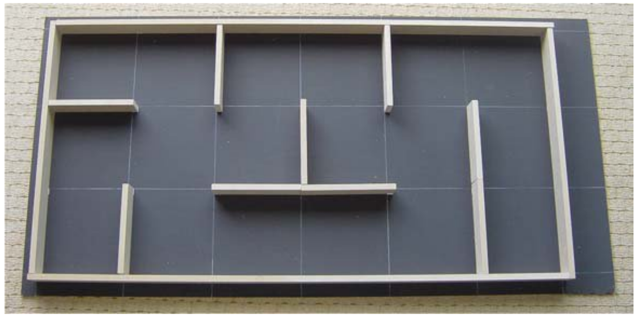




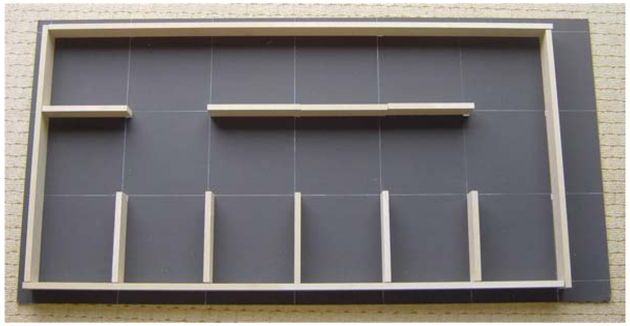
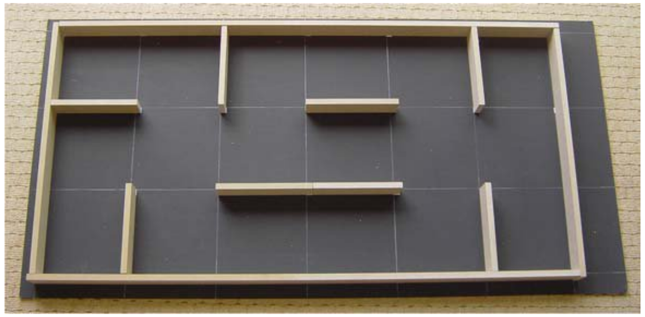
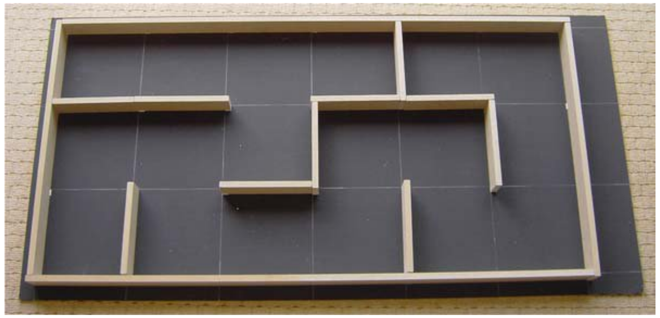
Place your maze blocks so that the white squares are covered.



Remember these maze walls are only slightly less reflective than plastic.

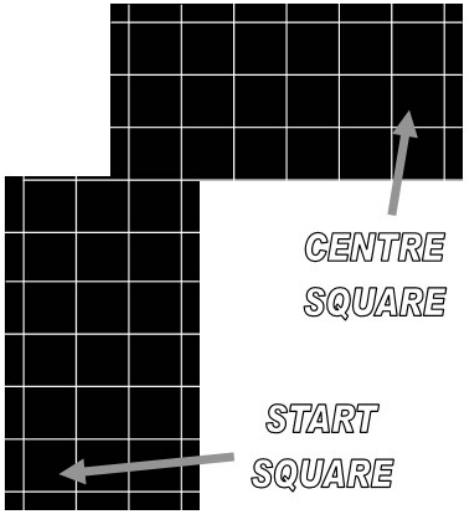




Remember if you are running a non-contact mouse your maze walls will not need to be fixed down.

Pay a few pounds more and double the maze size to 36 cells. Ask for eight 5cm strips to be cut from your MDF and buy an additional sheet of Hardboard.

With a maze size of 36 cells and your hardboard positioned as shown. A route to the centre of a full size maze (16 x 16 cells) is possible. Remember to make your 18cm pencil grid go to the edge of the hardboard.



**PICone a short history of achievements**

PICone is the prototype for the Picaxe Maze-solving Micromouse featured in this kit. Listed below are some great achievements. Full competition results are available at www.micromouseonline.com

UK Micromouse 07  Held on Saturday the 30th June 2007 at Technology Innovation Centre, Birmingham. PICone was awarded Best Engineered Mouse. PICone was entered into the non-contact wall following class.  Although it was still under development it was programmed with a simple routine enabling it to follow a wall. But unlike the rest of the wall followers it could track its position and indicate when it had reached the centre of the maze.

ROBOtic 07  Held on Saturday 24th November 2007 at Technology Innovation Centre, Birmingham. PICone achieved 4TH place in the Maze-solving Finals. PICone was now able to solve the maze and ran extremely well. Although its final time was quite slow its reliability shone through and stood out as the only maze-solving mouse to complete the final maze with out incurring a touch penalty. By staying out of the top three this kit does not overshadow new mice with true innovation but clearly demonstrates that it can also be competitive.

Techfest 08 Asia's Largest Science and Technology Festival  Held over the last weekend in January 08 at IIT Bombay, India. PICone achieved a time comparable to 2nd place against student only entries. Once again PICone showed extreme reliability achieving its goal with out any exterior assistance. The competition was only open to students and with over 30 maze-solving mice in the finals Micromouse is proving very popular in India. Once the students had ran there mice PICone was ran in the same maze in front of the vast audience and successfully demonstrated simplistic control with a time of 66 seconds.

Minos 08 Held on Sunday 13th April 2008 at Royal Holloway, University of London PICone achieved 4th place maze-solving finals, competition is heating up as new faster mice are on the horizon. PICone still managed a respectable time, keeping it fast enough to set a good bench mark for Micromouse without depriving the top prizes from true individual innovation.

UK Micromouse 08  Held on Saturday 28th June 2008 at The Technology Innovation Centre, Birmingham. PICone achieved 4th place Maze-solving Finals. Steady and reliable PICone achieved a respectable time without incurring a touch penalty, once again remaining out of the top three but proving a simple Basic program running on Picaxe processors will achieve success in Micromouse.

ROBOtic 08  Held on Saturday 29th November 2008 at Technology Innovation Centre, Birmingham. PICone achieved 4TH place in the Maze-solving Finals. PICone Turbo achieved 5TH. Both of the PICone entries demonstrated consistence. The new PICone Turbo had some minor adjustments to the 18x basic program allowing it to guess a direction when faced with a decision, this reduced its time spent stopped when solving the maze. In addition a direct link from the unregulated 9v battery to the motor driver chip was used (as mentioned earlier in this manual). Unfortunately the final maze had an unforgiving stretch of 10 continuous turns including a single dead end, this proved slightly too much for an unregulated power supply, preventing PICone Turbo from achieving an extremely impressive time.

UK Micromouse 09 Held on Saturday 27th June 2009 at The Technology Innovation Centre, Birmingham. A slightly modified PICone achieved 4th place Maze-solving Finals. We had removed the Picaxe 18x and ran the program entirely on a new 28x2 at 40mhz. also an adjustable regulator was added between 9v battery and the motor driver chips, this was set at about 8v giving more speed. Some small tweaks had to be made to the basic program to run the 28x2 but nothing major, but we did improve the dead end manoeuvre making single dead ends more reliable, by continuing into the dead end until the front sensor reaches a required value. Another very impressive outing for PICone.

**Check out our videos on YouTube**

| | |
|---|---|
| How to build a low cost MicroMouse maze | http://www.youtube.com/watch?v=YmxA92Sz_VI |
| Introducing the Maze solving Micromouse | http://uk.youtube.com/watch?v=wcMGUwdNfT0 |
| PICone Best engineered award UK 2007 | http://www.youtube.com/watch?v=C0hSSCJfE2I |

**NEW VIDEOS**

| | |
|---|---|
| P0W PicOne Turbo at APEC USA2016 | https://www.youtube.com/watch?v=9A55ZC0wqkc |
| TURBO  at BCU Tecfest UK 2013 | https://www.youtube.com/watch?v=aTMWBCV5L_A |
| PicOne Micromouse RoboTic 2011 | https://www.youtube.com/watch?v=8yBQG1L2_YM |