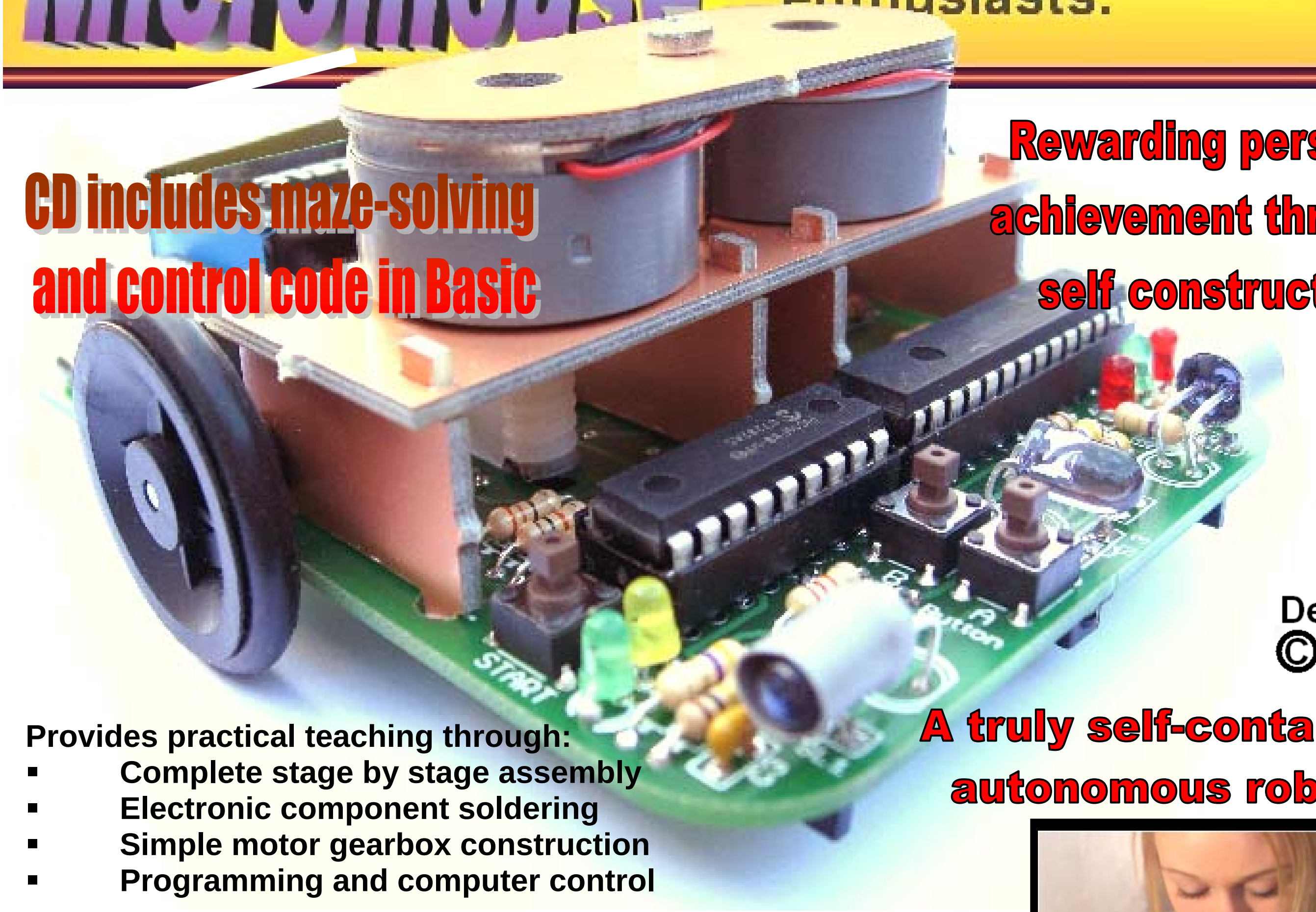
This is a cut down version of the Picaxe Micromouse Manual supplied with the kit.

- **The kit includes the entire maze solving software**
- **Many tips and hints on completing the challenge**
- **Photos and Videos of the completed Micromouse**

maze-solving Micromouse

The autonomous robot

Ideally suited for both students and amateur enthusiasts.



CD includes maze-solving and control code in Basic

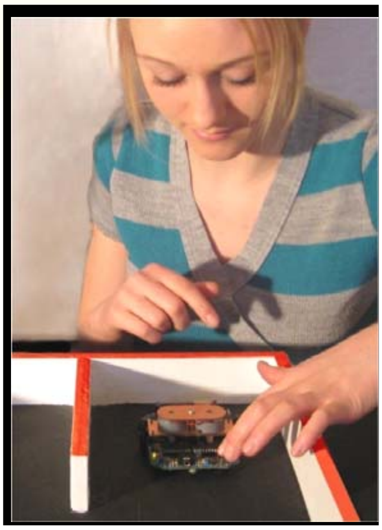
Rewarding personal achievement through self construction

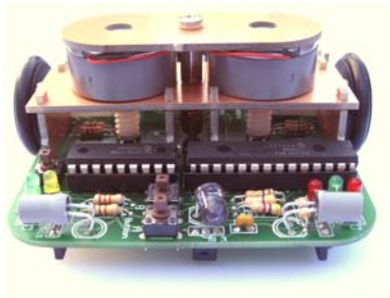
Design
© 2007

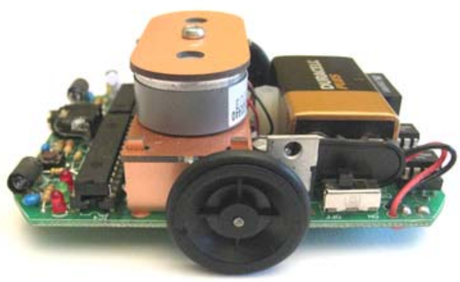
A truly self-contained autonomous robot.

Provides practical teaching through:

- Complete stage by stage assembly
- Electronic component soldering
- Simple motor gearbox construction
- Programming and computer control







Kit Features:

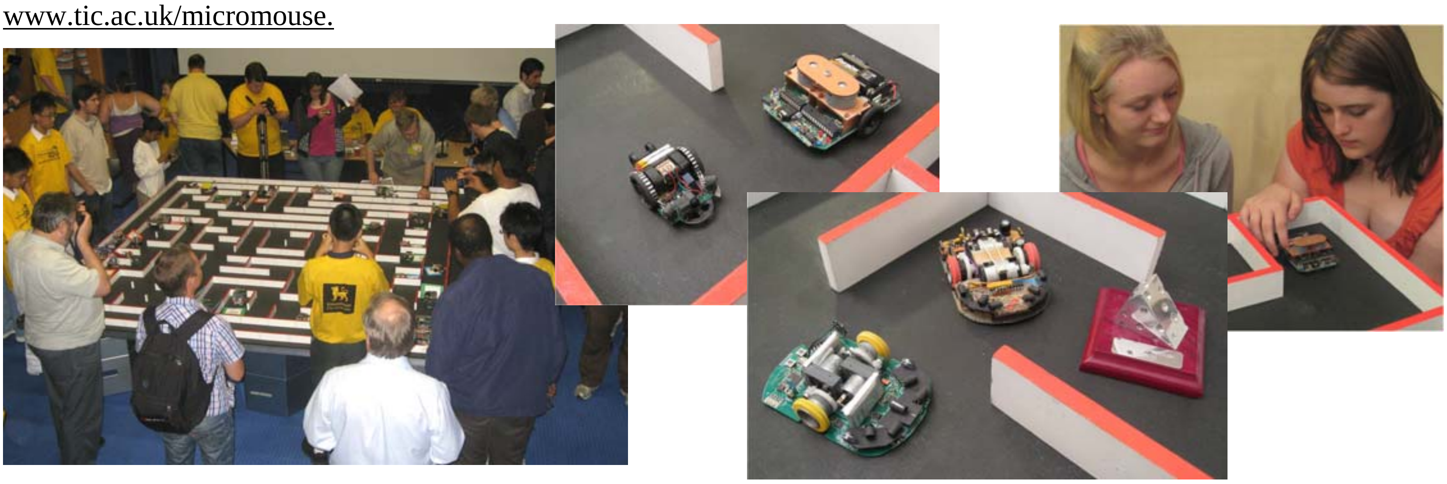
- Full photo construction plans and component explanation
- Dual processor allows team programming
- Encouraging the link between hardware and software understanding
- Maze solving algorithm included

Winner of the best Engineered mouse UK MICROMOUSE COMPETITION 2007
Achieved 4th place UK MICROMOUSE COMETITION 2008

The complete low cost maze-solving Micromouse kit

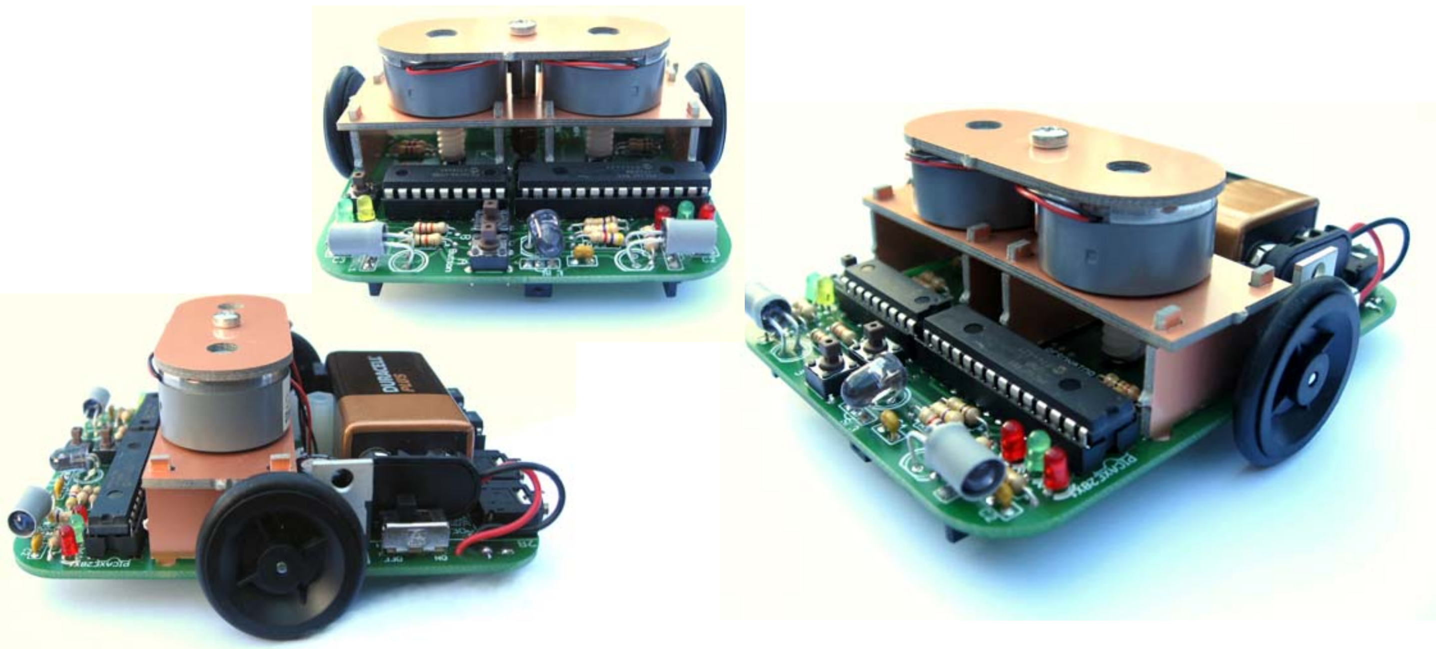
What is Micromouse ?

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 set 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 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.tic.ac.uk/micromouse.



Although a similar challenge was held in 1979, it was not until 1980 at the Euromicro event in London UK, that the universal maze sizes and centre goal squares were set. This became the challenge that is Micromouse as we know it today. 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 3 metres per second.

Introducing The Maze Solving Micromouse Kit



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 Flowchart diagrams to assist understanding and instructions and tips on fine-tuning. The key design elements are fully explained providing you with everything you require to build a robot mouse, setting a minimum benchmark for all Micromice.

The kit is an ideal introduction into computer control. It shows BASIC program equivalents to Flowchart constructs and 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.
- Small ruler and craft knife.
- Wire cutters / small screwdrivers.
- Small piece of kitchen foil and a drop of oil.

If you are using the software provided, you will require both the Picaxe 28X1 and the Picaxe 18X. It is possible to compile the program so that it can be run on just the Picaxe 28X1. This will lose the advantage of being able to split the software into two separate programs for easy development; possibly with separate teams working on control and maze solving. The serial download cable can be ordered from www.picaxe.co.uk . The free program editing software can also be downloaded from this site.

- 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. This algorithm is used in applications such as satellite navigation and route finding on the roads.
- 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.

THIS COMPLETE MANUAL IS ALSO IN PDF FORMAT ON THE CD.

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 motor into place to fit flush.

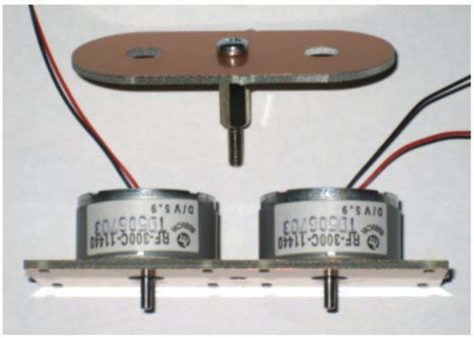


Fig 2
Note the worm gears are not pushed tight against the motors allow about a 3mm gap.

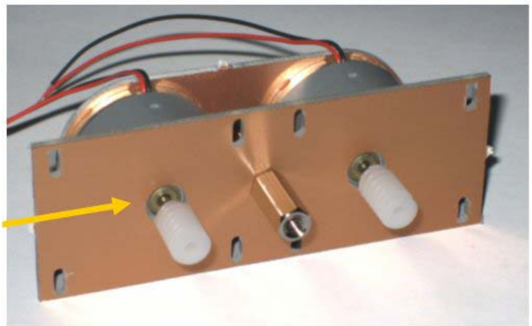


Fig 3
Make sure the wheels are not pushed too tightly against the side assembly.

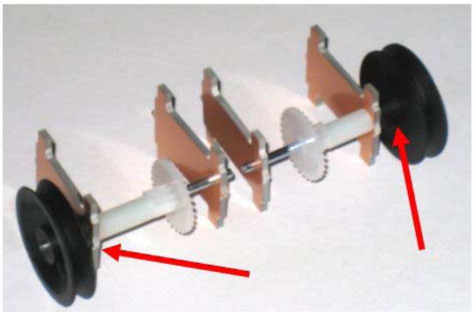


Fig4
A small piece of black card is secured using a contact adhesive. Be careful not to get any glue on the teeth of the worm gear. Once the glue has dried carefully trim the card to the same width as the gear.

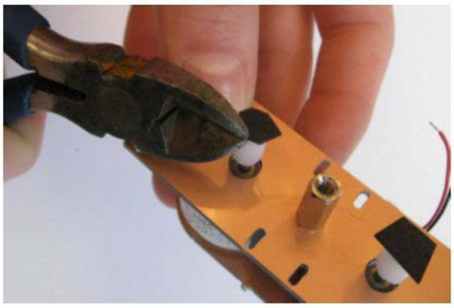


Fig 5
Now glue small pieces of kitchen foil with the shinny side up to cover half of the black card, and trim to size when dry.

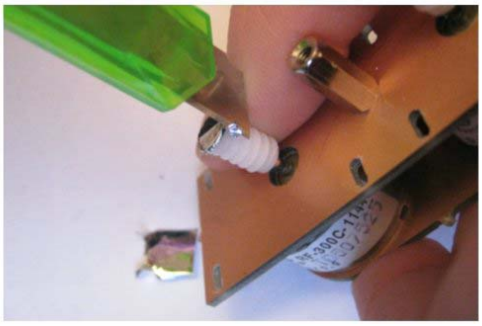
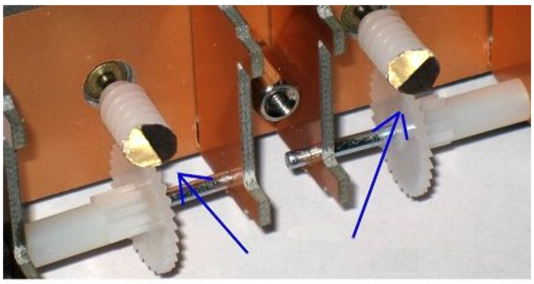
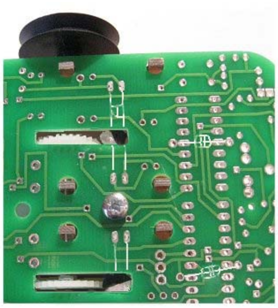
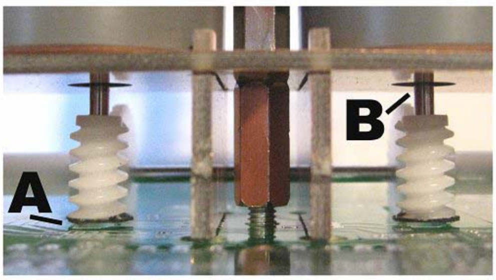


Fig 6
The black and silver discs form the wheel counters and should be as close to half and half as possible, to ensure the counts are as even.







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 1 mm clearance from the PCB. To adjust place a small flat head screwdriver at point "B" and slowly turn it. This will ease the gear into to right place.

The small IR phototransistor receives pulses as the motor turns. This is counted by the processor. Both ON and OFF pulses are counted for manoeuvres, but it is only necessary to count the ON pulses (61 counts per square) when travelling in a straight line.

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, this may rub the foil off on the PCB causing the wheel counters to fail and your mouse to turn continually.

- An alternative to the foil and black card can be found in additional information at the end of this manual.

When building your first Micromouse, a common mistake is to overcomplicate your design. It is very tempting to use far too many sensors and multi layer control boards. Ultimately this will only make your mouse extremely difficult to fine tune. A slow precise mouse that finds the centre will always beat a fast erratic one that does not. This kit has been designed to simplify the overall problems in hardware and software design, by using minimal components without trading off reliability.

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 x4. CHECK ORIENTATION match the notch in the socket with the notch in the outline.
- Battery connector cut wires to 45mm in length and pass the wires through the holes CHECK ORIENTATION (the square pad is positive)
- Diodes x3 CHECK ORIENTATION (these will help protect noise created by the motors affecting the processors)

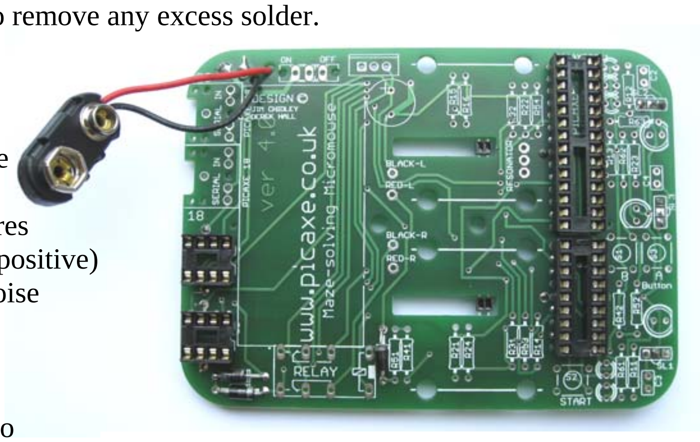
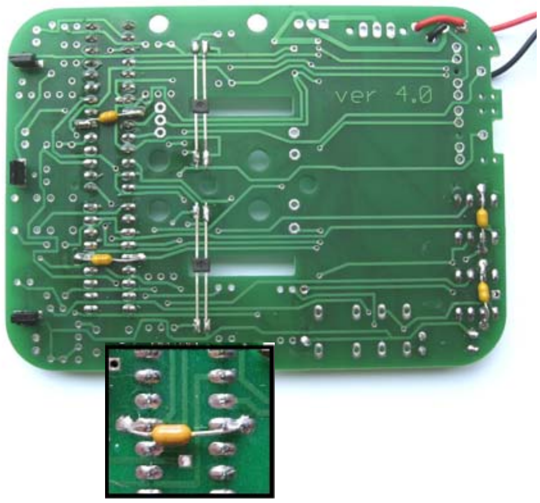


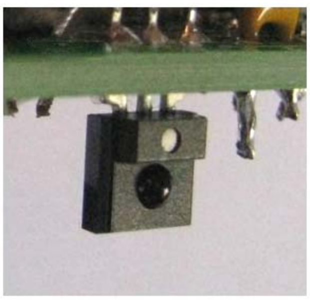
Fig 2 Turn board over

- C5,C6,C7,C8 (these connect across power legs of the four ICs to smooth out any power spikes)

- Light to Voltage Converters x3. CHECK ORIENTATION STATIC SENSITIVE. Use minimal solder as the pads are very close (after soldering make sure none of the legs short circuit using a resistance tester and take care when soldering additional components close by)

- Phototransistor x2.STATIC SENSITIVE CHECK ORIENTATION Centralize and surface mount. The two shiny areas face towards the board and one corner is chamfered, an outline is shown in white on the board. NOTE: Both chamfered corners point to the centre of the PCB.





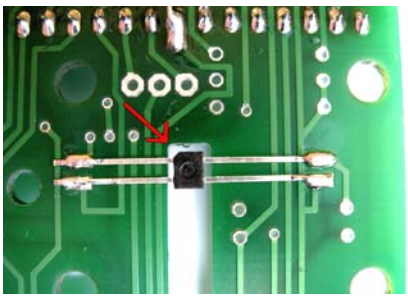


Fig 3

Resistors x 21. Be careful to align resistors through the correct component holes.

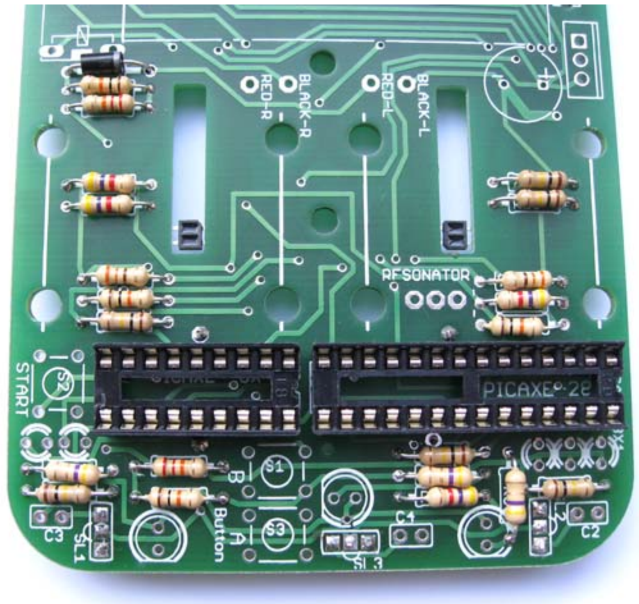
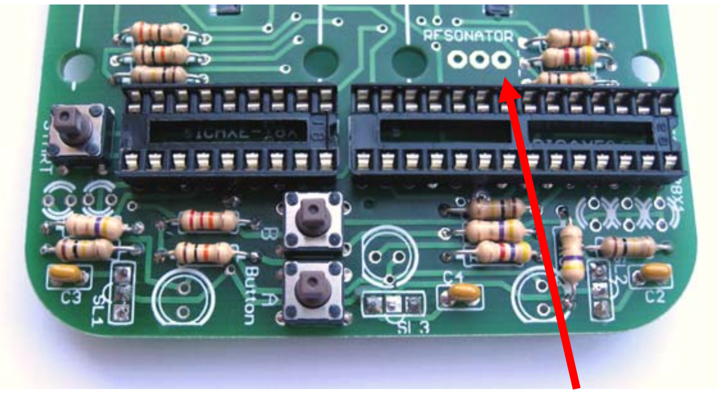
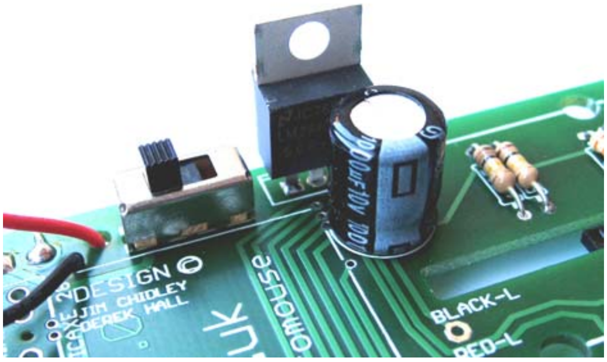


Fig 4

- Capacitors x3 C2,C3,C4
- Push buttons x3 button "A", "B" and START
- ON/OFF Slide switch
- Voltage regulator CHECK ORIENTATION STATIC SENSITIVE
- CHECK ORIENTATION (the grey strip is negative and goes as shown.)



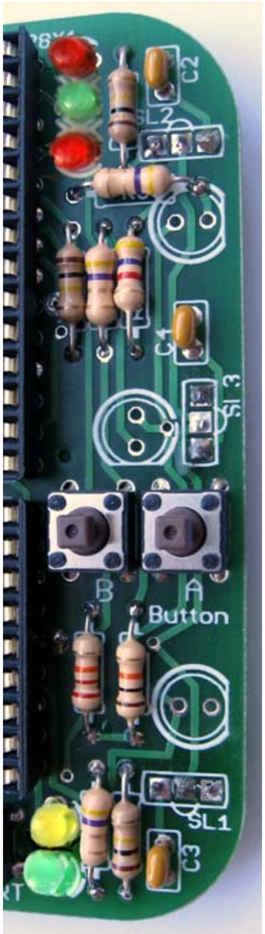


A resonator may be added at a later date but this is not necessary for the completion of this kit. See additional information at the end of this manual or Help in Picaxe programming editor for more details.

RED

GREEN

RED



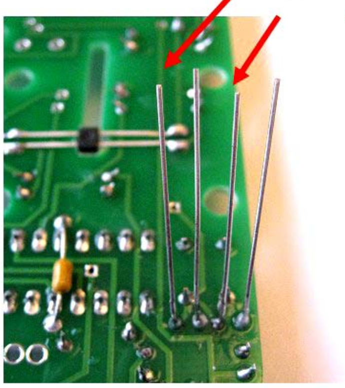
YELLOW

GREEN

Fig 5

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

Cathode



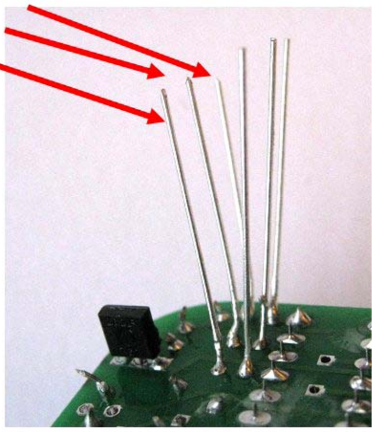


Fig 6

- Mount the motor / gearbox assembly (align the holes up and screw to attach)

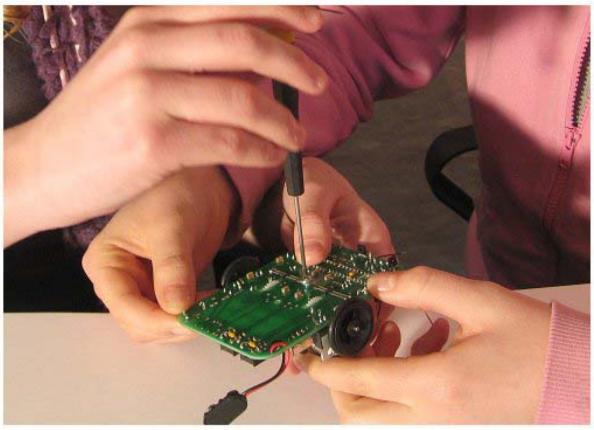


Fig 7

- 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

- Jack sockets x2. (Each Picaxe needs a separate download jack plug)
- Relay. (the right hand motor is reversed by the relay, allowing the mouse to turn on the spot)

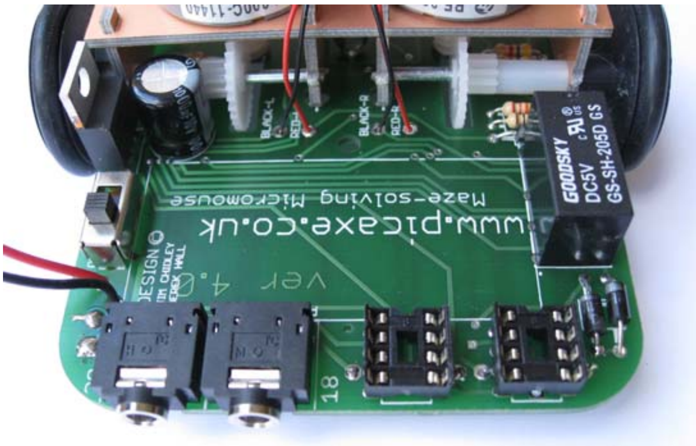
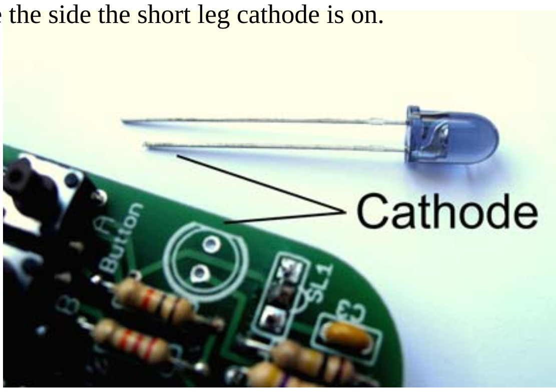


Fig 8

- IR LEDs x3. (Cut shrink wrap to two 7mm lengths and slide over two LEDs for side wall sensing. These will help narrow the beam for the side sensors creating a sharper image as the wall disappears.) Bend the LED legs to fit as shown below do not have the LED's protruding passed the edge of the PCB. Notice the side the short leg cathode is on.



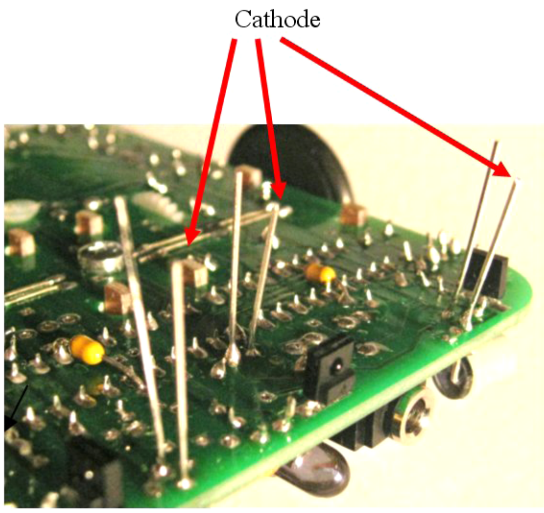
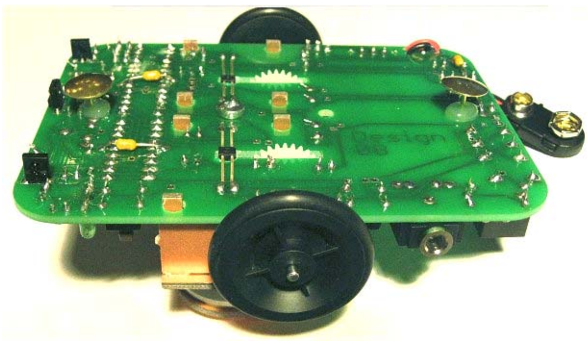
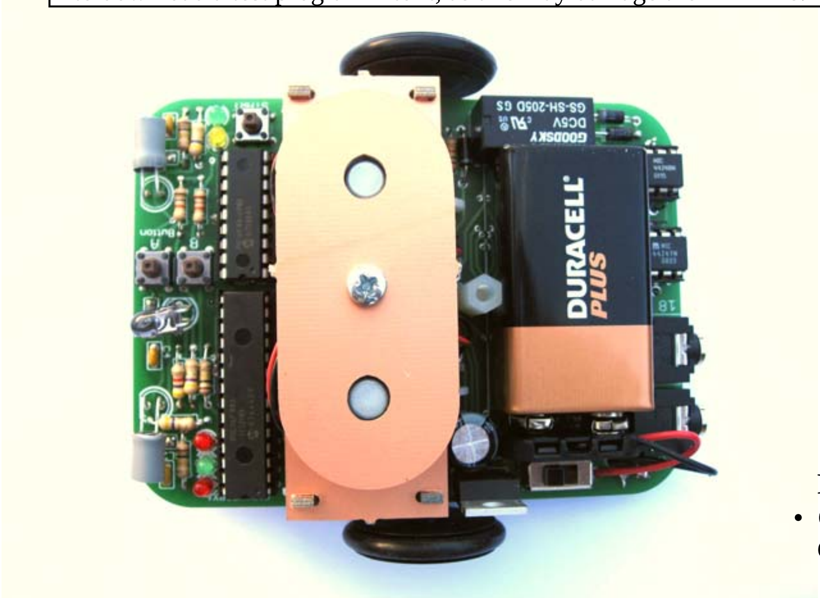


Fig 9

- 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.
- M3 Screw peg. (this will secure the battery when in place)



Do not power up your mouse for any length of time until you are ready to download a test program into it, as this may damage the IR LEDs.



Use a tiny amount of glue at the back of the LED to secure shrink-wrap in place.

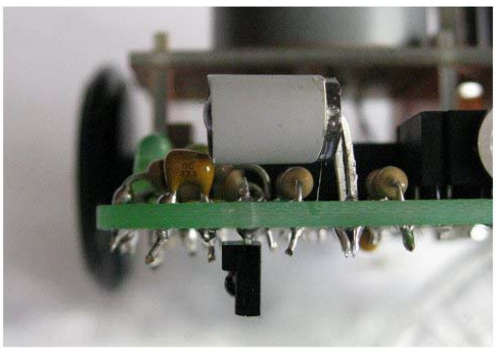


Fig 10

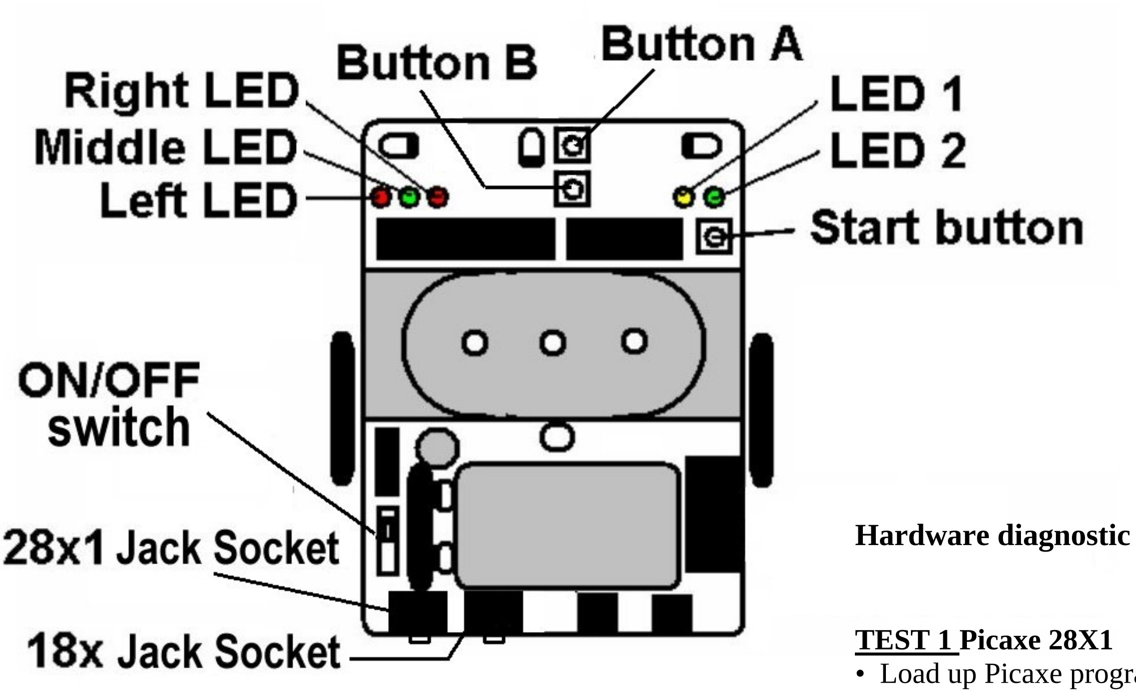
- Carefully insert the two driver chips and two Picaxe chips. CHECK ORIENTATION STATIC SENSITIVE

Construction is complete.

Test routines

There are 3 test programs and 2 final programs on the CD-rom supplied. The tests are all in BASIC with some Flowchart equivalents. By examining the flowcharts you can follow the program broken down to its simplest form. This will aid in the understanding of how the program works. 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.

Once all the tests 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).



Hardware diagnostic Tests 1 and 2

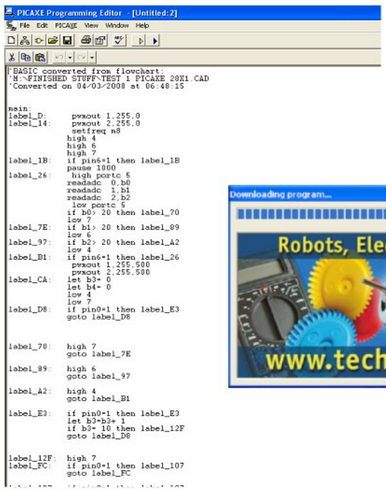
TEST 1 Picaxe 28X1

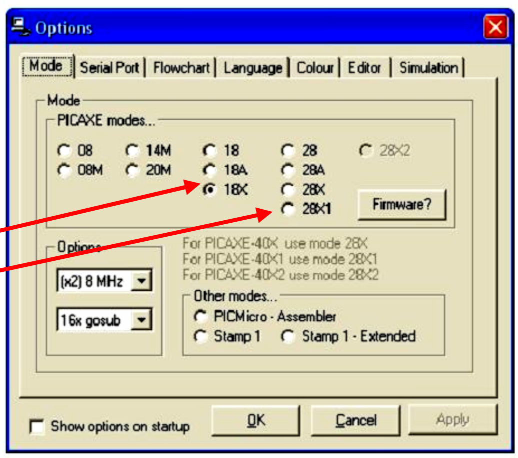
- Load up Picaxe program TEST 1 into Picaxe programming editor.
- Check **28X1** is selected in VIEW / OPTIONS.
- Switch on your mouse.
- Insert download cable into **28X1 jack socket**.
- Press F5 to run program. (Whilst downloading the program into Picaxe, hold your mouse off the ground as the motors may turn at random).
- Right, Middle and Left LED's now stay on.

TEST 2 Picaxe 18X

- Load up Picaxe program TEST 2 into Picaxe programming editor.
- Check **18X** is selected in VIEW / OPTIONS.
- Switch on your mouse.
- Insert download cable into **18X 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).
- LED 1 and LED 2 now flash alternatively.

With the power switch still on press button "B"(reset). This will cause all five LEDs to go off while it is held down. Button "A" will stop LED 1 and LED 2 from flashing.





Select correct Picaxe
PICAXE 18X
PICAXE 28X1

TEST 1 and 2 are diagnostic tests 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 “A” and infrared LEDs and sensors.**
Press button “A”. LEDs 1 and 2 will now go out. Press “Start” button. Left, middle and right LEDs go off. The corresponding LEDs will light up if an object is brought close enough to be detected by the left, middle and right wall sensors.

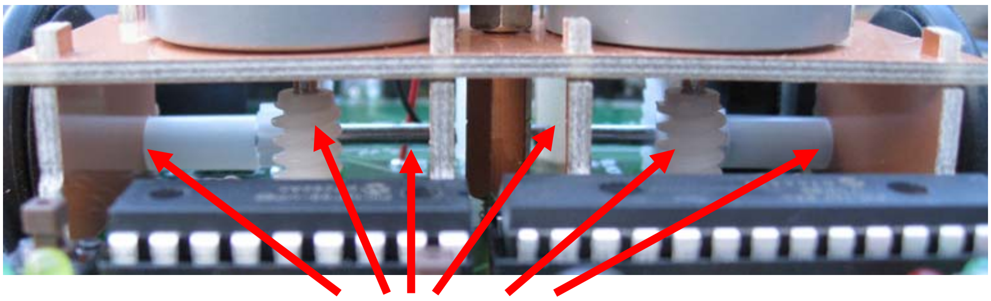
• **Testing motors and left wheel counter.**
Hold your mouse off the ground and press the “Start” button. Both wheels turn in a forward direction. The **lef**t LED flashes on and off every 20 **left** wheel counts. Moving the worm gear away from the sensor will increase the time the LED is off, try to 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. Right wheel reverses for half a second then continues forward for half a second and then repeats.

• **Testing serial communications between Picaxe28X1 and Picaxe18X.**
Press “Start” button. Middle, 1 and 2 LEDs light for 1 second then go off simultaneously. This then repeats continually. Press button “B”(reset) to restart test sequence.

At this point 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.



Apply minimal oil to these points.

Simple faultfinding

- Ensure correct Picaxe is selected during programming in VIEW / OPTIONS and the correct jack socket is selected.
- 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 worm gear is not too far away from phototransistor and that the phototransistor is aligned. Also check the black paper and foil are attached.
- 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.
- Check charge in battery using a voltmeter and components are the right way round.

TEST 3 debug 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 3 debug Picaxe 28X1 into Picaxe programming editor.
- Check **28X1** 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)

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.

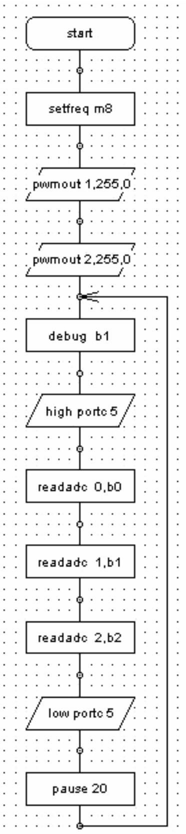
readadc 0 Left sensor = **b0**
readadc 1 Front sensor = **b1**
readadc 2 Right sensor = **b2**

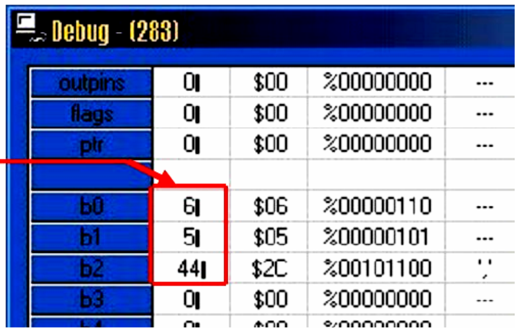
As you move the wall within the range of the sensors you will notice that the ADC values are not linear. The distance achieved between low readings is much greater than with higher reading. Saturation will be reached at a value 156. Stray infrared light reflecting off the black floor or through the PCB can give a value of 6, so when setting up try and avoid setting levels below 7.

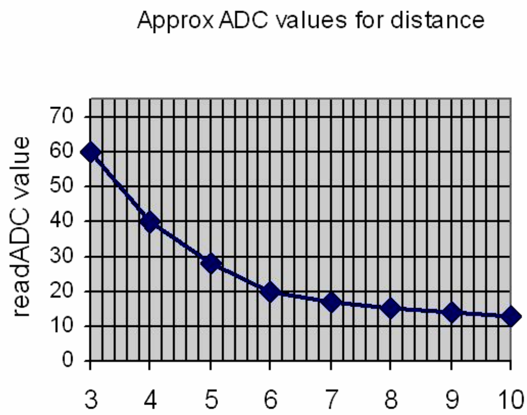
Place your mouse on the matt black maze base and position a wall as shown on the next page. Record the values from the five simple tests. When moving the wall closer the sensor will increase the relevant variable in the debug program.

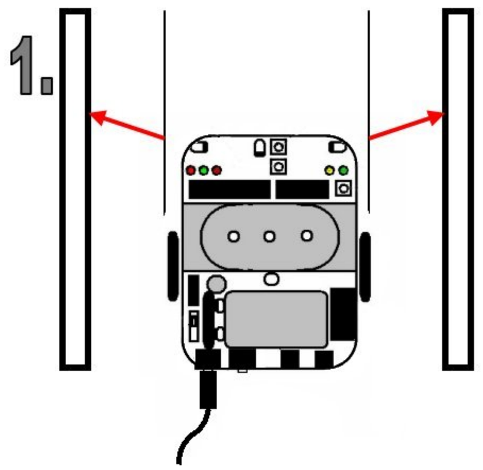
Test 3 Flowchart and Basic conversion

```
'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
```





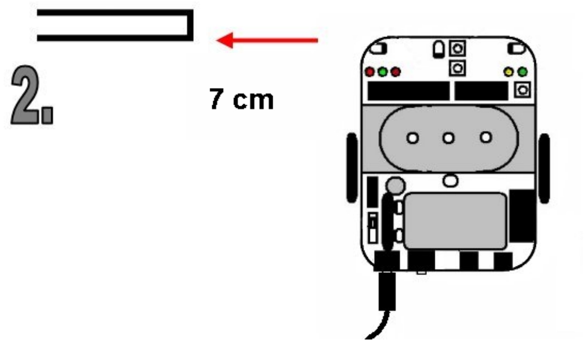




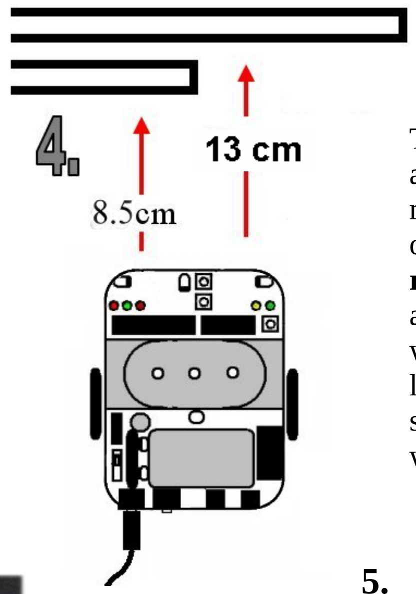
1. With your mouse centred between two sidewalls record your debug sensor readings for **b0** and **b2**.
The b0 value = left_straighten
The b2 value = right_straighten

2. With your mouse PCB at a distance of 8cm from the end of a right hand wall record the maximum sensor reading achieved for **b2.**
The b2 value = left_wall

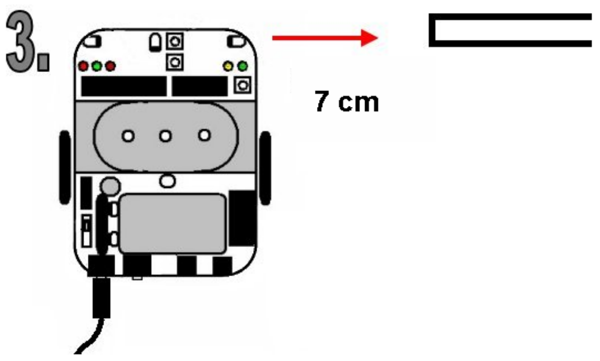
3. With your mouse PCB at a distance of 7cm from the end of a left hand wall record the maximum sensor reading achieved for **b0.**
The b0 value = right_wall

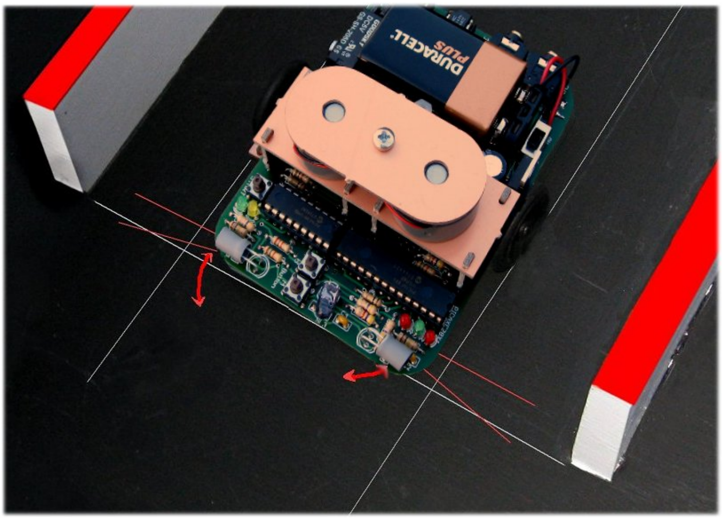


4. Position a wall directly in front of your mouse at a distance of 12cm and record your **b1** debug sensor reading**.**
The b1 value = front_wall
Now record the reading at 8.5cm and enter
The b1 value = reset_frontwall



The straight-line count for a single square is set to make your mouse slightly over shoot. The 8.5cm **reset_frontwall** setting allows your mouse to reset when travelling down a long dead end. The 13 cm setting is used to map the wall.

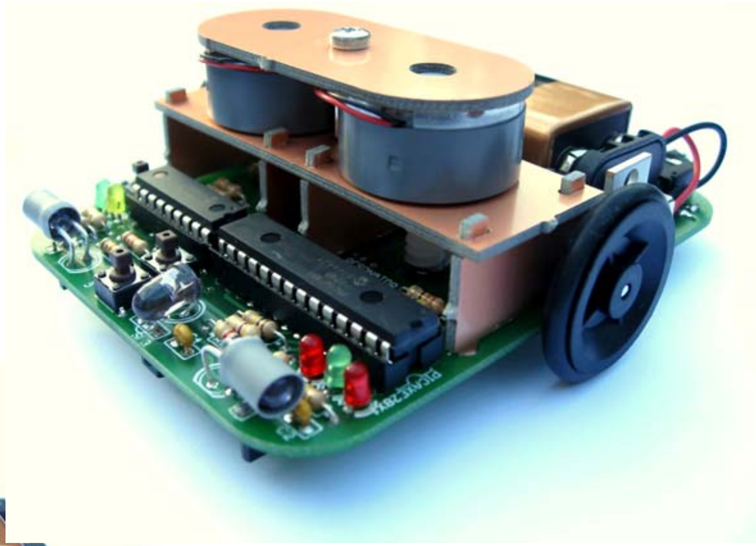


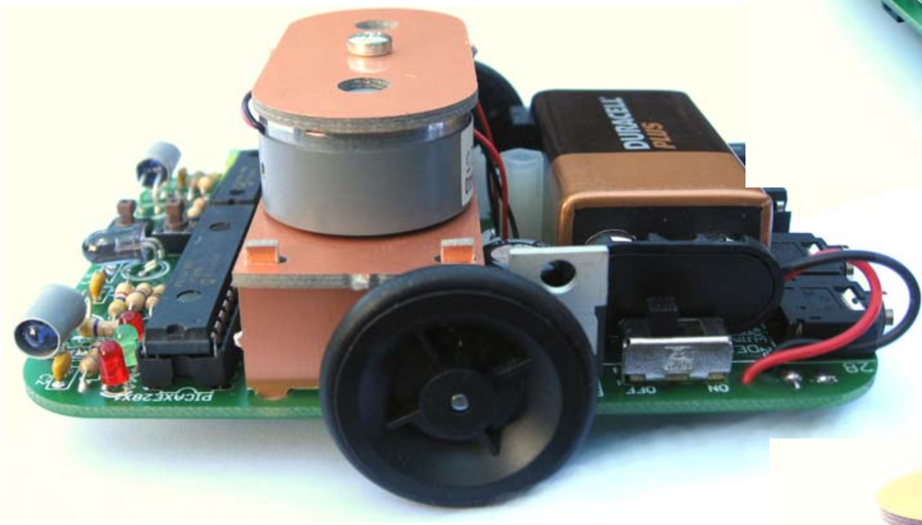


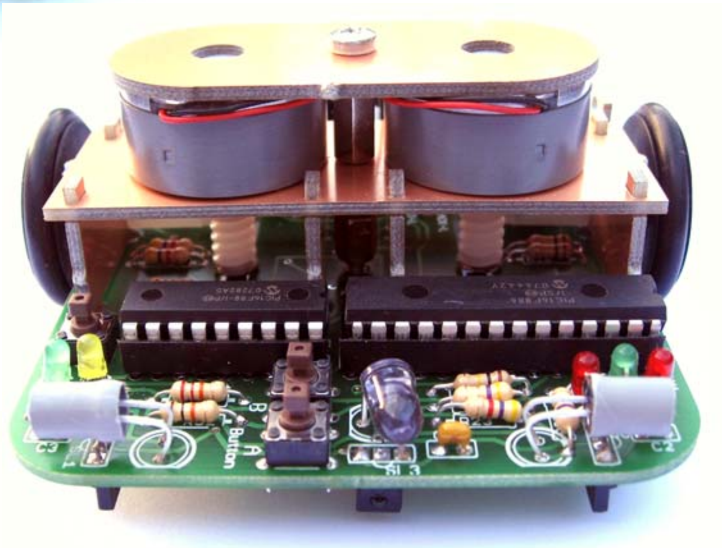
5.
Setting up the point when the side sensors no longer can detect a wall is done by lining up the front of you mouse level with the point the walls end.

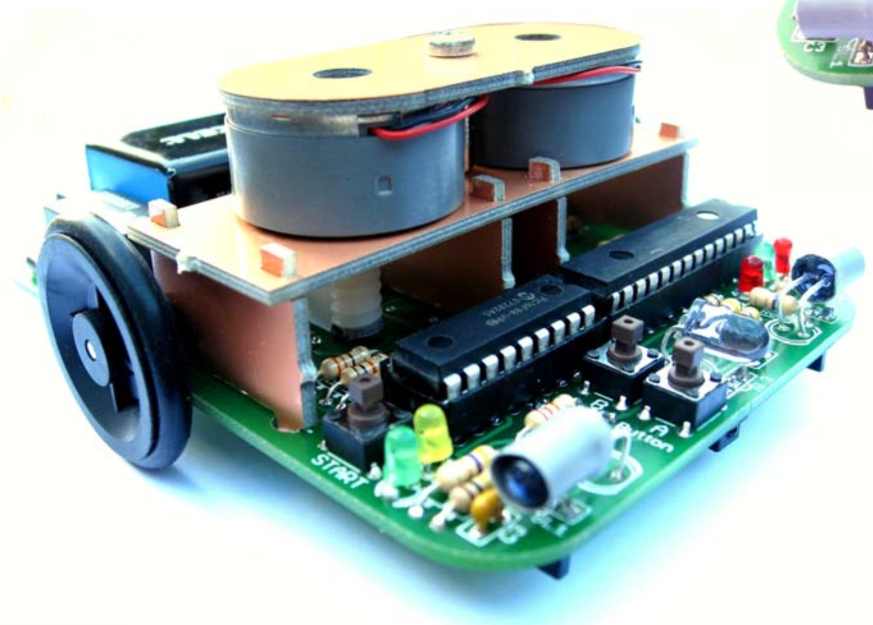
Additional videos on Youtube

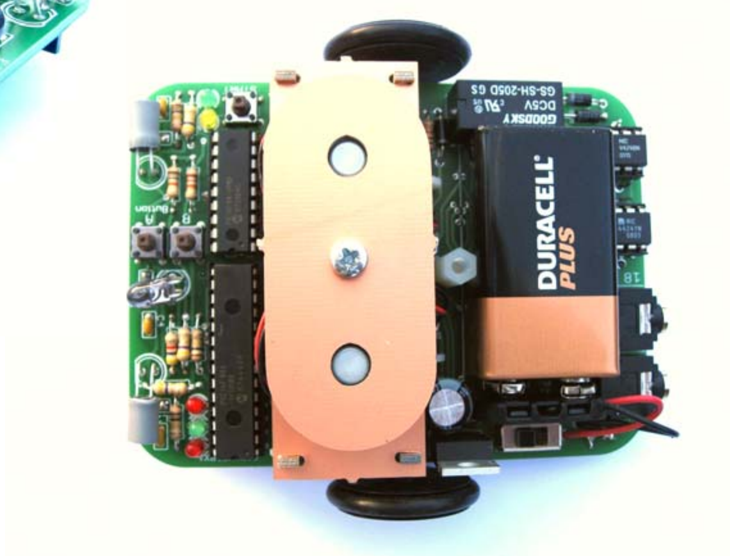
http://uk.youtube.com/watch?v=wcMGUwdNfT0
http://www.youtube.com/watch?v=4ZcOlplgfBo
http://www.youtube.com/watch?v=C0hSSCJfE2I
http://www.youtube.com/watch?v=ualWsvS88lc
http://www.youtube.com/watch?v=h-TDB6gTgOY
http://uk.youtube.com/watch?v=J4tZ1Z3wbbo
http://uk.youtube.com/watch?v=bsGovk9ksIg





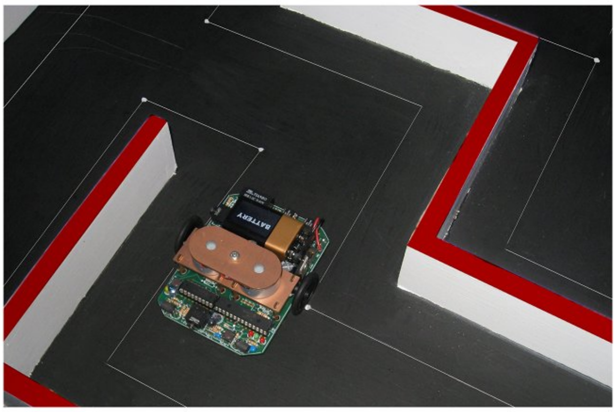






Fine tuning your mouse

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. Keeping your mouse design simple will aid you immensely during your fine-tuning process with less variables to consider. Setting your mouse to complete simple basic manoeuvres using minimal sensor readings will help identify poor incorrect settings.



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) and may not be able to sense a front wall. Your mouse will then slow down in speed.

The outputs pins, inputs pins and variables used in the Main Control programme are listed below:

28X1 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 C5**
Communication output	**out 0**

28X1 Program Inputs	
Left wheel counter	**pin 0**
Right wheel counter	**pin 3**
Start button	**pin 6**
Communication input	**pin 4**
Left sensor	**adc 0**
Front sensor	**adc 1**
Right sensor	**adc 2**

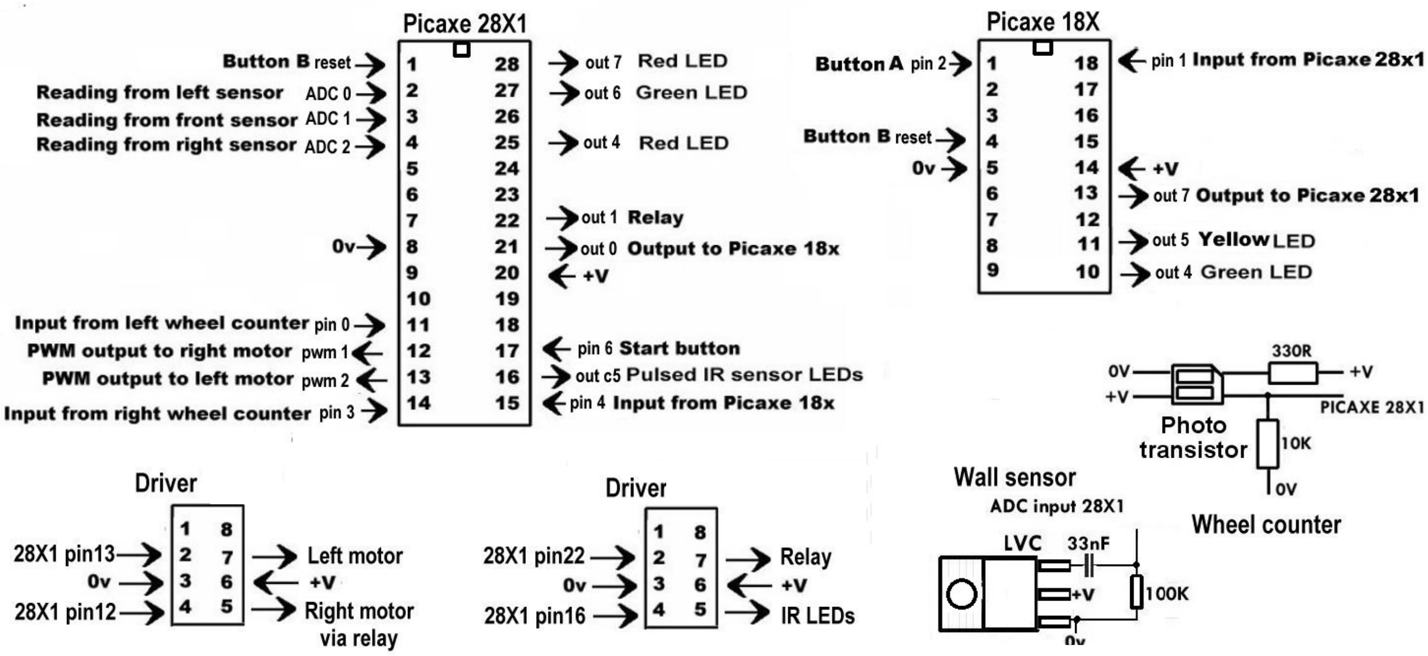
28X1 Program Variables			
Straight-line counter	**b5**	Wall config for previous cell	**b4**
Left sensor	**b0**	Straight-line wheel counter	**b5**
Front sensor	**b1**	Communication input from 18x	**b6**
Right sensor	**b2**	Wheel count during manoeuvres	**b7**
Wall config for current cell	**b3**	Setting wheel count manoeuvres	**b8**

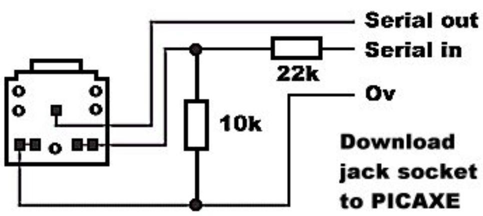
28X1 Adjustable variables for fine tuning (These are found at the start of the programme)

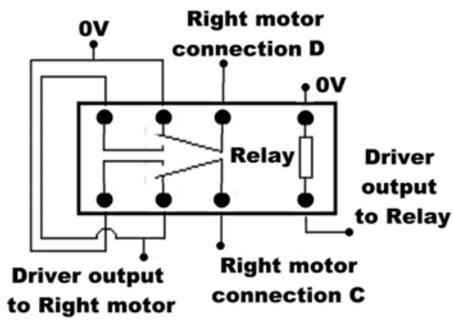
left_motor= 'sets left motor speed for straight line (higher number travel faster max 1023)
right_motor= 'sets right motor speed for straight line (higher number travel faster max 1023)
left_wall= 'sets distance for left wall detection (lower number greater distance)
right_wall= 'sets distance for right 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 (used for long dead ends and 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)

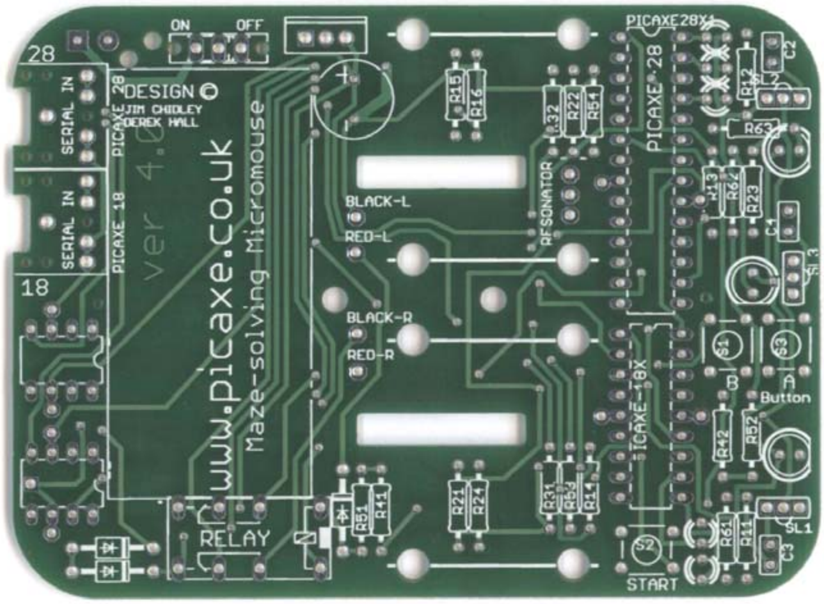
Additional information

This diagram shows a basic break down of both the Picaxe 28X1 and Picaxe 18X and pins used.





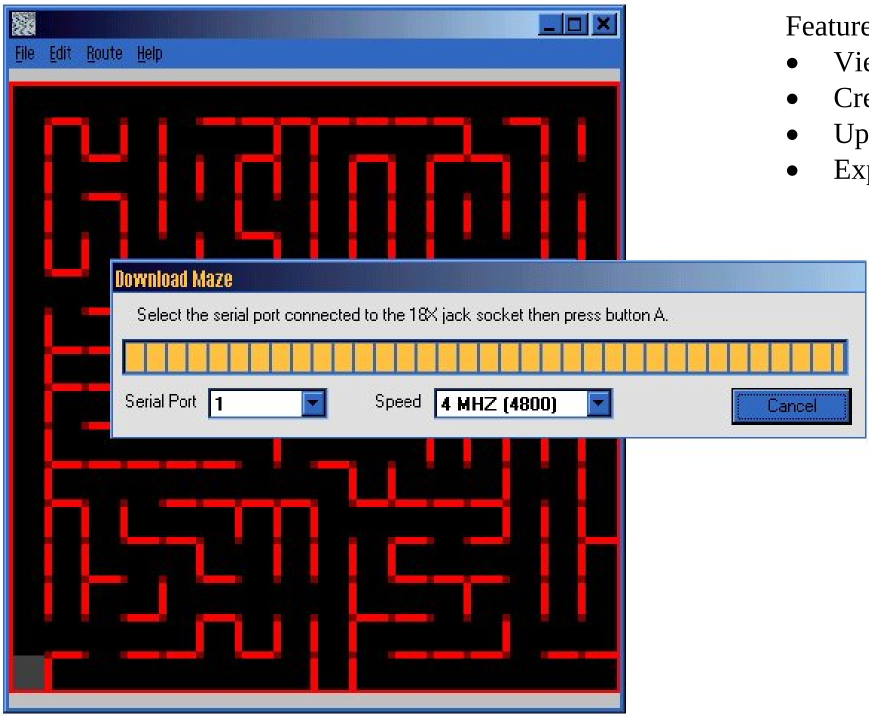




Micromouse Maze Editor (included on the CD supplied)

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.

After your mouse has been run in a maze, you can view a maze it has stored in the 18x Picaxe. With you r Micromouse turned on connect the Jack lead to the 18x socket click LISTEN FOR MAZE in the FILE MENU then press Button “A” on your mouse, the maze your mouse has mapped will 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 the competition maze for future reference without having to draw it by hand.

You can also load a set maze into the Picaxe 18x, setting the route your mouse will take helping with setup by allowing manoeuvres to be exactly calibrated without interference from wall straightening. Create a maze then EXPORT MAZE DATA in the FILE MENU, this can then be copied and pasted into the 18x Maze Solving program, replacing the blank maze.

DIY MICROMOUSE MAZE FOR less than £17

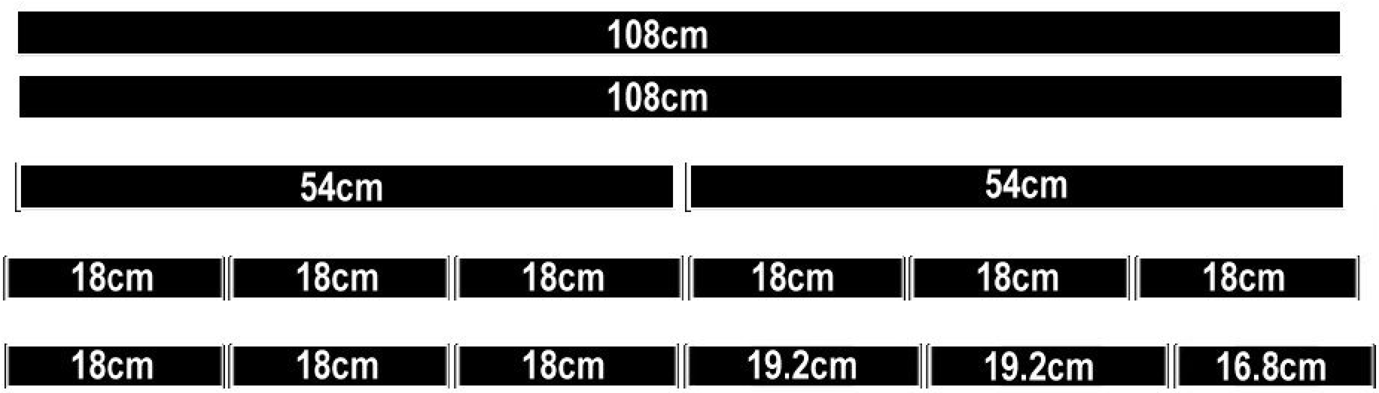
- **LOW COST**
- **EIGHTEEN CELL MAZE**
- **CHANGEABLE LAYOUT**

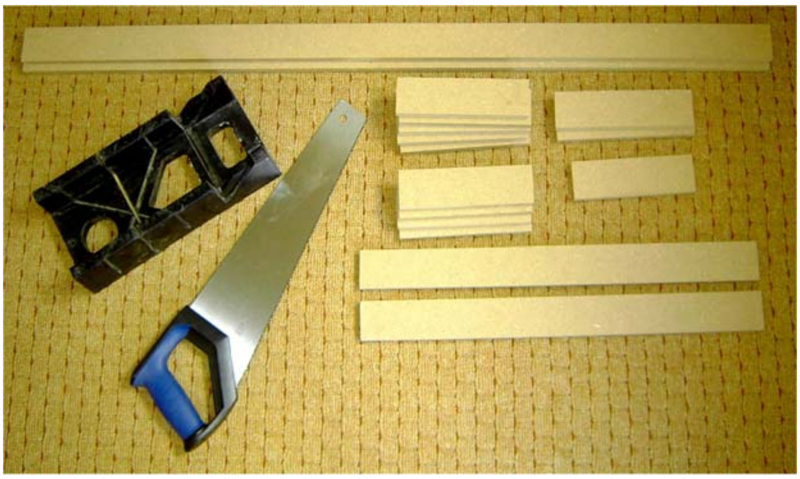
The following instructions demonstrate how to create a low cost Micromouse maze.

- 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.

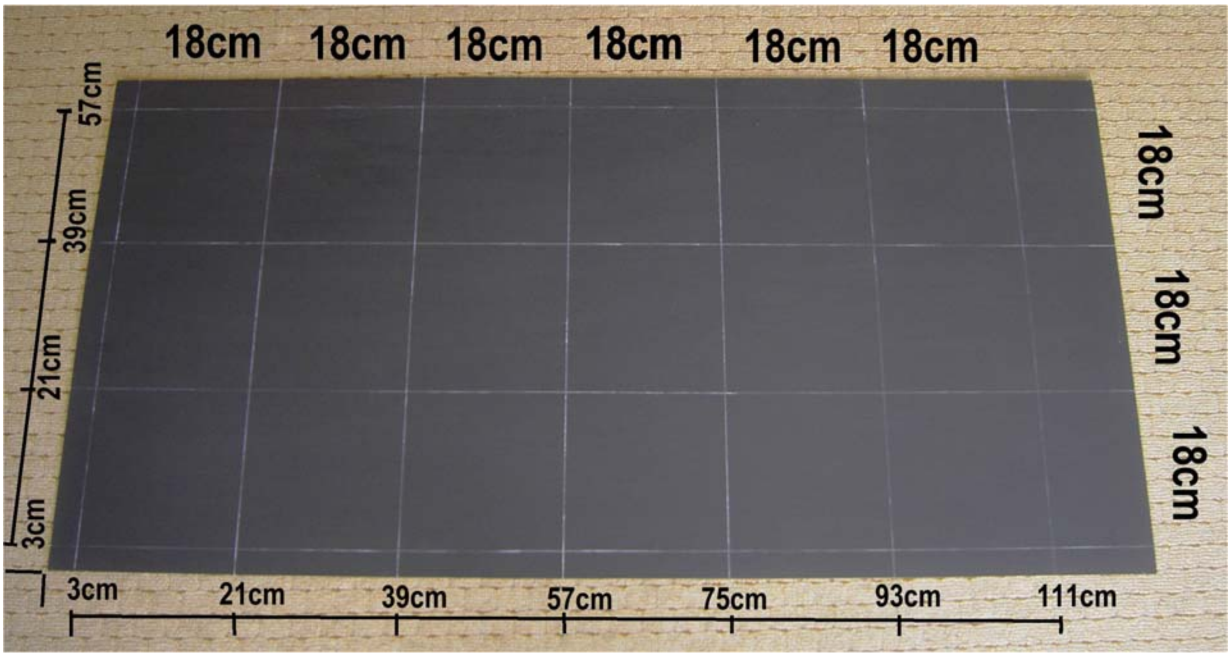
Cut the 5 MDF strips as shown using a mitre block, then lightly sand any rough edges.





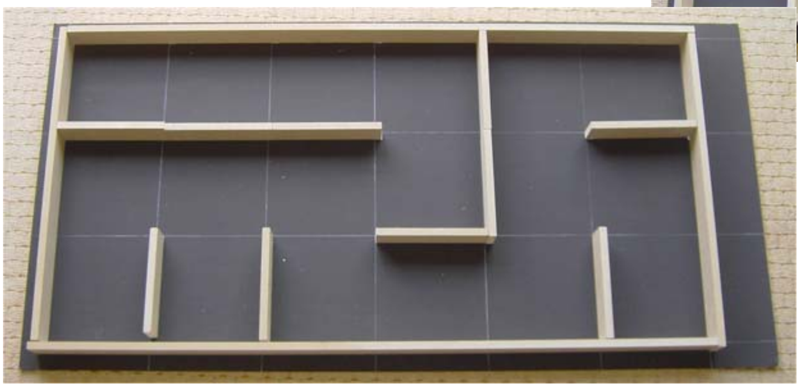
Cut out twenty-eight small 1.2cm x 1.2cm squares out of sticky white paper.

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.

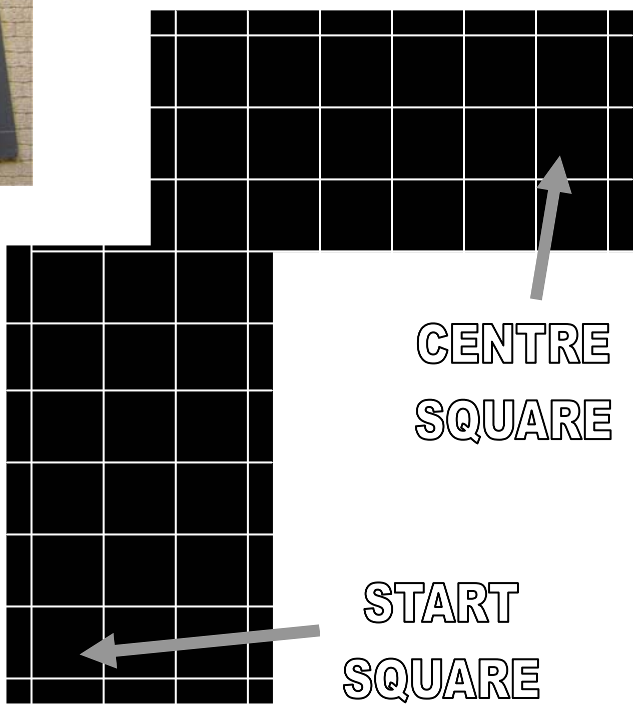


These maze walls are only slightly less reflective than then plastic walls.

If you are running a non-contact mouse your maze wall do not need to be fixed.

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.

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.

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 66seconds.

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.