MADLAB MADBOT

Construction

Construction is in two parts. First the mechanical gearbox base is built, then the electronics are assembled on the printed circuit board (PCB). Finally the two parts are joined together.

Gearbox base

Carefully follow the instructions included with the gearbox base, paying close attention to the diagrams.

Note the following points:

- as stressed in 4.2, the central plate 'F' must be perfectly upright on 'G' or the gears will not mesh properly
- the motors audibly click when pushed fully home in their metal supports
- the small piece of clear tubing pushed onto the screws is used to hold the nuts firmly in place
- the two blue gears on each side point in different directions
- don't omit the small brass tubes, these keep the gears in the correct positions and ensure smooth running
- don't overtighten the screws, particularly those holding the wheels in place as the plastic may crack

You may find it easier to solder the smoothing capacitors and wires to the motors before fitting them into the gearbox as it requires some care to do this once they are in place in a finished gearbox (see below).

Fit either the wheels or the legs onto the shafts. The wheels are recommended rather than the legs as the latter give rise to somewhat erratic movement (and MADBOT is prone to flipping onto its back). But try both as it is easy to swap between legs and wheels. The legs might be better if MADBOT is used on thick carpets as the wheels will not give good traction on these surfaces.

Trim any burrs left on the rims of the wheels.

The gearbox might require some lubrication with a lightweight oil if it ever sounds like it is not running smoothly.

Circuit board

First solder the two short un-insulated wires to the pairs of holes marked LINK. Then fit and solder the resistors (R1 to R9) and trim their legs. Identify the resistors by the coloured stripes on the body. Next fit and solder the capacitors, paying attention to the polarity of the electrolytic capacitors C1, C4, C5 and C6 (negative is marked by a stripe on the side of the body, and also the shorter leg). The polyester (C2) and ceramic (C3) capacitors can be fitted either way around. Note that there should be two capacitors left over, these are soldered directly to the motors (see below).

Then fit the regulator (REG). The symbol on the board indicates the orientation of the regulator (flat side of the component against the flat side of the symbol).

Solder the light sensor (LDR) to the board either way around. Be careful when soldering as excessive heat may melt the plastic.

Solder the eyes (LED1 and LED2) putting the shorter leg into the hole with the line. Solder the IR transmitters (IR1 and IR2) matching the shorter leg with 'k' on the board. The small lenses on the transmitters should be pointing towards the front of the board. Next solder the IR module (RECEIVER). Again its lens should be pointing forwards.

Solder the speaker (PIEZO), slide switch (S1), and keyboard socket (PS/2). Take care when soldering the PS/2 socket as the pins are very close together.

Next fit the chip sockets IC1 and IC2 (matching the notch in the socket against the notch in the symbol on the board). Again care should be taken when soldering these components to avoid solder bridges between the pins. It is not recommended that the chips are soldered directly to the board.

Solder the battery connectors (BATT+ and BATT-) matching the shape to the symbol on the board (the hexagonal connector is positive, the circular negative). Make sure the connectors are pushed fully into the board and are square with the board, and all the holes are well soldered.

Don't fit the chips until you have thoroughly checked your construction. Check that all the components have been inserted correctly and that there are no dry joints and no solder bridges between pins. Check that all component legs and other wires have been trimmed close on the underside of the board (so no shorts result when the base is attached). Then carefully bend the legs of the chips inwards a little with your fingers. Fit the chips into their sockets matching the small notch in the chip to the notch in the socket.

Insert 6 AA cells into the battery box, observing the correct polarity. (If one of the cells is reversed in the battery box then MADBOT may appear to be only partially working.) The cells used should ideally be rechargeable NiMH or NiCd types, but if disposable cells are used they should be good quality alkaline ones. It is recommended that high-capacity rechargeable NiMH cells are used. Connect the battery box to the battery connectors on the board.

The software includes a power-on self-test. Move the slide switch S1 to the ON position and (after a second or two) both LEDs should flash and a double beep should sound.

Turn MADBOT off, connect a PS/2 keyboard to the socket then turn MADBOT back on. Keyboard options are provided to test the IR obstacle sensors and the LDR light sensor. Function key F11 echoes the obstacle sensors to the LEDs (left & right eyes). Waving your hand in front of the IR sensors should cause the LEDs to come on. Move you hand left and right and check that both sensors work. Function key F12 echoes the light sensor to the LEDs. Covering and uncovering the light sensor should cause the LEDs to flicker. Completely covering the sensor should cause all the LEDs to light.

At this point you can be reasonably confident that your board is functioning correctly. Now it's time to join the two parts together. Turn MADBOT off again and remove the battery box.

Solder the remaining two ceramic capacitors (C7 and C8) across the tags of the two motors on the gearbox base using short pieces of the sleeving (cut each piece in two) to insulate the legs and prevent them from shorting on the metal casing of the motor. Then solder the insulated flexible wires to the tags of each motor as well. The other ends of the wires are soldered to the holes on the board within the marked outlines MOTOR1 and MOTOR2, first feeding the wires up through the adjacent pair of slightly larger holes (not the large screw hole). The positive tag on each motor is the tag furthest from the base of the gearbox (i.e. the one nearest the ground when the gearbox is upright) and may be marked by a red dot on the body of the motor. The corresponding positive hole on the circuit board is marked by a ‘+’ sign.

Reconnect the battery box, turn MADBOT on and press function key F1 to check that both motors run (after a 2 second delay), then turn MADBOT off.

After MADBOT has been verified to be working correctly, screw the circuit board to the gearbox base using 4 of the screws provided with the base. Tuck the motor wires in out of the way under the board well away from the gears.

Repeat the tests described above. If MADBOT no longer functions properly then a dry joint probably exists somewhere on the board. The act of screwing the board to the base applies stress to it and borderline joints may fail under the stress.

The battery box should be secured to the circuit board using the Velcro dots. The back edge of the PS/2 socket is a good place to attach a hooks dot (bend the dot in the middle at right angles and stick it into the crook of the socket), with a matching loops dot on the corner of the battery box. Alternatively stick loop dots

to the board under the battery box, with hook dots on the bodies of the AA cells themselves. Only do this if you are using rechargeable cells, and you will need to keep the cells with the dots always in the same position in the battery box. Stick an extra couple of the softer loop dots on the board under the battery box to cushion it.

Source code

The documented source code for the MADBOT firmware is downloadable from the MadLab website and is 'open source'. In other words you are encouraged to study its operation and modify it to suit your particular requirements. To do this you will need access to a PIC programmer. A suitable programmer is PICSTART Plus from Microchip but there are many other capable programmers on the market (it must be able to program 16F628A's though).

See http://www.madlab.org/kits/madbot.html for further information.

MADBOT User Guide

MADBOT responds to the keyboard as follows:

F1 - roam mode (obstacle avoidance)
F2 - maze mode (wall following)
F3 - push mode (searches for objects then pushes)
F4 - avoid mode (triggered by changes in light level)
F5 - light mode (searches at random for light regions)
F6 - dark mode (searches at random for dark regions)
F7 - dance mode (plays a recorded sequence of moves)
shift F7 - edit mode (records a sequence of moves)
shift F10 - adjust steering (corrects for left/right motor bias)
F11 - echo IR sensors to LEDs (left & right) (**ESC** to exit)
F12 - echo LDR sensor to LEDs (left & right + keyboard) (**ESC** to exit)
shift PAGE UP - increase speed
shift PAGE DOWN - decrease speed
shift HOME - IR sensors detuning off
shift END - IR sensors detuning on (3 settings)
UP ARROW - move forward
DOWN ARROW - move backward
LEFT ARROW - turn left
RIGHT ARROW - turn right
ESC - restart

With a keyboard attached MADBOT can be driven about using the arrow keys (cursor keys). For example to move it forward press and hold down the **UP ARROW** key. Release the key to stop. Autonomous modes (i.e. MADBOT moves by itself) are selected by pressing one of the functions keys **F1** to **F7**.

In roam mode (**F1**), MADBOT moves forward until it hits an obstacle, backs up, changes direction, then moves forward again. In maze mode (**F2**), MADBOT moves forward until it hits a wall, then attempts to follow the course of the wall. In push mode (**F3**), MADBOT spirals outwards looking for an object, then when it's found something attempts to push it forward. In avoid mode (**F4**), MADBOT waits until it's approached then takes avoiding action. In light mode (**F5**), MADBOT searches at random until it finds an area of light. In dark mode (**F6**), MADBOT searches at random until it finds an area of dark. In dance mode (**F7**), MADBOT plays back a recorded sequence of moves (see below).

There is a 2 second delay after selecting mode **F1** to **F7** to allow MADBOT to be turned off and the keyboard disconnected. When turned on again the selected mode is activated. In other words when MADBOT is powered up with no keyboard connected then it automatically enters the last mode selected when the keyboard was attached.

The speed of the motors can be reduced if necessary by pressing **shift PAGE DOWN**, or increased again by pressing **shift PAGE UP**. The default speed is the maximum speed. Note: reducing the speed will make MADBOT sluggish and liable to get stuck.

The IR sensors detuning decreases the range of the sensors. Turn detuning on if you find the sensors are too sensitive to obstacles. Each time **shift END** is pressed more detuning is selected. Press **shift HOME** to turn detuning off.

The steering, speed and detuning settings are stored in non-volatile memory and retained after power is removed.

Edit mode

Press **shift F7** to enter edit mode, then use the following keys to record a 'robot dance':

UP ARROW - move forward, followed by duration in 1/8s (**1** to **9, A** to **F**)
DOWN ARROW - move backward, followed by duration in 1/8s (**1** to **9, A** to **F**)
LEFT ARROW - turn left, followed by duration in 1/8s (**1** to **9, A** to **F**)
RIGHT ARROW - turn right, followed by duration in 1/8s (**1** to **9, A** to **F**)
S - stop movement, followed by duration in 1/8s (**1** to **9, A** to **F**)
B - beep
F - flash LEDs
L - loop to start (and stop recording)
BACKSPACE, **DELETE** - delete last move
ENTER, **ESC** - finish recording

Times are specified in fractions of a second, from 1/8 second to 15/8 seconds. The keys **A** to **F** are equivalent to the numbers 10 to 15. So the shortest move forward is **UP ARROW** followed by **1**, and the longest move backward is **DOWN ARROW** followed by **F**.

Here's a short example of a dance that moves MADBOT in a figure-of-eight then beeps, flashes and repeats:

UP ARROW 8 RIGHT ARROW 3 UP ARROW 8 LEFT ARROW 3
UP ARROW 8 LEFT ARROW 3 UP ARROW 8 LEFT ARROW 3
UP ARROW 8 LEFT ARROW 3 UP ARROW 8 RIGHT ARROW 3
UP ARROW 8 RIGHT ARROW 3 UP ARROW 8 RIGHT ARROW 3
S F B F L

Up to 100 dance moves can be stored in memory. The dance is retained in memory when the power is removed.

Note: distances travelled and angles turned through in move sequences depend on the state of the battery charge and the traction of the wheels with the surface, so are generally not very repeatable.

Steering adjust

Press **shift F10** if MADBOT veers from moving in a straight line when moving forward. This might be caused by slight imbalances in the two sides of the gearbox, if one set of gears was slightly stiffer than the other for instance. The use the following keys to adjust the steering:

LEFT ARROW - adjust movement to the left
RIGHT ARROW - adjust movement to the right
ENTER, **ESC** - finish adjusting

So for example if MADBOT curves to the right when moving forward then press the **LEFT ARROW** key until it straightens.

Component List

Resistors

R1, R6	10k (brown, black, orange, gold)
R2, R4, R5	150R (brown, green, brown, gold)
R3	100R (brown, black, brown, gold)
R7	4k7 (yellow, purple, red, gold)
R8	10R (brown, black, black, gold)
R9	1R (brown, black, gold, gold)

Capacitors

C1, C5, C6	100u electrolytic
C2	100n polyester (yellow or blue, square)
C3, C7, C8	100n ceramic (brown, marked '104')
C4	10u electrolytic

Semiconductors

REG	78L05 regulator
LED1, LED2	8mm red
IR1, IR2	IR transmitter (clear)
RECEIVER	IR module (black)
IC1	16-pin socket + L293D H-bridge driver
IC2	18-pin socket + PIC16F628A-I/P microcontroller (FB1X)

Miscellaneous

LINK	2 x wire links
LDR	light dependent resistor
PIEZO	piezo speaker
S1	slide switch
PS/2	PS/2 socket
BATT+, BATT-	battery connectors

Battery box	6 x AA
Velcro dots	4 x loops + 4 x hooks
Pre-cut wires	x 4
Sleeving	x 2

Gearbox base
PCB