MicroTech

MT Chips
Hardware
Reference

Version 3.52

www.mcu.hk

Octobert, 2010

Introduction

This manual outlines all the information and procedures necessary to get your microcontroller (MCU) projects running on MT chips. It is strongly recommended that you read this manual at least once to get an overview of the design philosophy and architecture of MT chips.

The major features that distinguish MT chips from other similar products on the market are:

- **all** I/O pins can be used both for input and output <u>without</u> the need for setting direction registers.
- **all** I/O pins can be used to generate pulses for speakers/devices and moving R/C servos.
- **8 channel, 10-bit A/D converters** (on selected chips).
- **any** 5 I/O pins can be used as external interrupt inputs.
- a **timer function** which can generate regular interrupts at user defined intervals.
- multiple baud rates up to 19200 for serial communications (except for MT-08 and MT-14 which are fixed at 9600). For ARM versions of MT chips, the maximum baud rate is 115200.
- a **power down** system function is used to put the chip into power saving mode to save the supplying battery energy, a negative pulse on the WU pin will wake up the chip previously put into power down mode, then it continues execution where it has left off.

MT chips come in a variety of different sizes - 8, 14, 20, 28 and 40 pin. They differ only in the size and number of TinyC programs that can be stored within the chips, the amount of run-time memory units available and the TinyC functions supported.

TinyC is a C-like programming language used to develop programs for MT chips. Although tiny in terms of data type and language constructs, and with just over a dozen functions, it is still powerful enough for developing complex programs. It is easy to learn and can be mastered in a short period of time. A TinyC program can have a maximum size of 32K program words (64K bytes).

TinyC was designed and developed by Victor Lee of MicroTech Systems Limited (www.mcu.hk) with contributions from Frank Crivelli of Ozitronics (www.ozitronics.com).

Please refer to the '**TinyC Programming Reference**' for further details.

What are MT chips?

MT chips are based on enhanced 8051 or Atmel AVR flash microcontrollers that have been programmed with the MT system code. The MT system code is a 16-bit run-time interpreter which executes programs written in the TinyC language. The MT system code has a built-in downloader which enables the chip to be reprogrammed (>100,000 times) directly via a serial connection to a PC, eliminating the need for an expensive IC programmer. This makes developing programs for MT chips a simple and easy task.
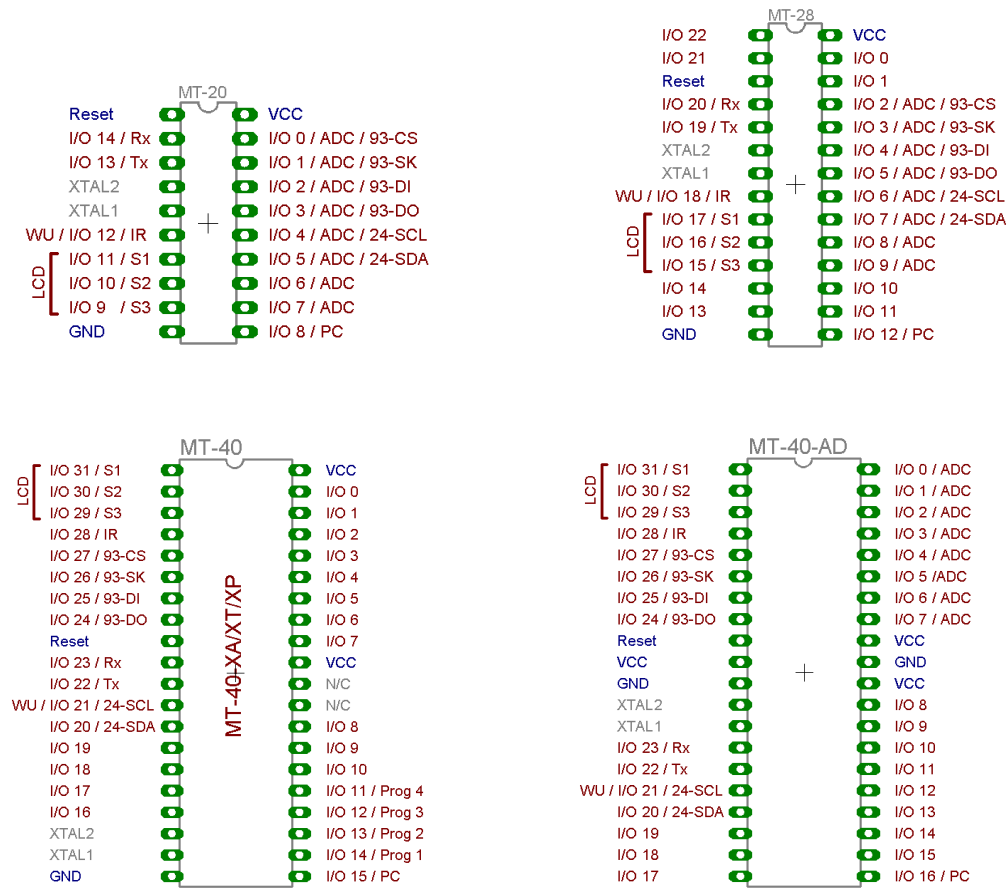
There are two classes of MT chips. One class, MT-40r1 and MT-40r2, are designed for operating on the MT MicroBoard and cannot be used as standalone chips. The other class consists of 10 types available in 8, 14, 20, 28 or 40 pin DIP packages. The primary difference between the chips is the number of I/O pins and memory available. The TinyC programming language is common to all MT chips. However not all functions are supported on all the chips. For example, the readadc() function is available only on those chips with A/D converters. Unsupported functions are simply ignored if a program is loaded into an MT chip that does not support those functions.

The following table and diagrams show the technical specification and pin layouts for the MT chips.

Chip Type	Program Storage	Memory	I/O	Crystal (MHz)	ADC	RS232	IR	MT-LCD	24C02	93C56
MT-40r1	5 x 4K	896	20	22.1184	No	Yes	Yes	No 7-Seg	Yes	Yes
MT-40r2	1 x 4K 4 x 1K	512	20	22.1184	No	Yes	Yes	No 7-Seg	Yes	Yes
MT-40-XX	1 x 16K 1 x 6K	896	32	22.1184	No	Yes	Yes	Yes	Yes	Yes
MT-40-XP	3 x 8K	896	32	22.1184	No	Yes	Yes	Yes	Yes	Yes
MT-40-XA	5 x 4K	896	32	22.1184	No	Yes	Yes	Yes	Yes	Yes
MT-40-XT	1 x 8K	512	32	22.1184	No	Yes	Yes	Yes	Yes	Yes
MT-40	1 x 4K 4 x 1K	512	32	22.1184	No	Yes	Yes	Yes	Yes	Yes
MT-28	1 x 1K	128	23	22.1184	8	Yes	Yes	Yes	Yes	Yes
MT-28A	1 x 10K	256	23	22.1184	8	Yes	Yes	Yes	Yes	Yes
MT-20	1 x 1K	128	15	22.1184	8	Yes	Yes	Yes	Yes	Yes
MT-20A	1 x 10K	256	15	22.1184	8	Yes	Yes	Yes	Yes	Yes
MT-40-AD	1 x 512	860	32	11.0592	8	Yes	Yes	Yes	Yes	Yes

Note:
- Program storage and memory are expressed in words (16-bit).
- RS232 serial functions for MT-14 and MT-08 chips are implemented in software and available only in 9600 baudrate and in half-duplex mode.
- Pin connections and schematics for the MT-40r1 and MT-40r2 chips are described in a separate document for MicroBoard. These chips are specifically designed for and only available with the MicroBoard.
- Different chip packages (PLCC, SMD) and sizes of program storage arrangements are available on request.

MT-20

Reset	VCC
I/O 14 / Rx	I/O 0 / ADC / 93-CS
I/O 13 / Tx	I/O 1 / ADC / 93-SK
XTAL2	I/O 2 / ADC / 93-DI
XTAL1	I/O 3 / ADC / 93-DO
WU / I/O 12 / IR	I/O 4 / ADC / 24-SCL
I/O 11 / S1	I/O 5 / ADC / 24-SDA
I/O 10 / S2	I/O 6 / ADC
I/O 9 / S3	I/O 7 / ADC
GND	I/O 8 / PC

(LCD bracket: I/O 11 / S1, I/O 10 / S2, I/O 9 / S3)

MT-28

I/O 22	VCC
I/O 21	I/O 0
Reset	I/O 1
I/O 20 / Rx	I/O 2 / ADC / 93-CS
I/O 19 / Tx	I/O 3 / ADC / 93-SK
XTAL2	I/O 4 / ADC / 93-DI
XTAL1	I/O 5 / ADC / 93-DO
WU / I/O 18 / IR	I/O 6 / ADC / 24-SCL
I/O 17 / S1	I/O 7 / ADC / 24-SDA
I/O 16 / S2	I/O 8 / ADC
I/O 15 / S3	I/O 9 / ADC
I/O 14	I/O 10
I/O 13	I/O 11
GND	I/O 12 / PC

(LCD bracket: I/O 17 / S1, I/O 16 / S2, I/O 15 / S3)

MT-40 (MT-40-XA/XT/XP)

I/O 31 / S1	VCC
I/O 30 / S2	I/O 0
I/O 29 / S3	I/O 1
I/O 28 / IR	I/O 2
I/O 27 / 93-CS	I/O 3
I/O 26 / 93-SK	I/O 4
I/O 25 / 93-DI	I/O 5
I/O 24 / 93-DO	I/O 6
Reset	I/O 7
I/O 23 / Rx	VCC
I/O 22 / Tx	N/C
WU / I/O 21 / 24-SCL	N/C
I/O 20 / 24-SDA	I/O 8
I/O 19	I/O 9
I/O 18	I/O 10
I/O 17	I/O 11 / Prog 4
I/O 16	I/O 12 / Prog 3
XTAL2	I/O 13 / Prog 2
XTAL1	I/O 14 / Prog 1
GND	I/O 15 / PC

(LCD bracket: I/O 31 / S1, I/O 30 / S2, I/O 29 / S3)

MT-40-AD

I/O 31 / S1	I/O 0 / ADC
I/O 30 / S2	I/O 1 / ADC
I/O 29 / S3	I/O 2 / ADC
I/O 28 / IR	I/O 3 / ADC
I/O 27 / 93-CS	I/O 4 / ADC
I/O 26 / 93-SK	I/O 5 /ADC
I/O 25 / 93-DI	I/O 6 / ADC
I/O 24 / 93-DO	I/O 7 / ADC
Reset	VCC
VCC	GND
GND	VCC
XTAL2	I/O 8
XTAL1	I/O 9
I/O 23 / Rx	I/O 10
I/O 22 / Tx	I/O 11
WU / I/O 21 / 24-SCL	I/O 12
I/O 20 / 24-SDA	I/O 13
I/O 19	I/O 14
I/O 18	I/O 15
I/O 17	I/O 16 / PC

(LCD bracket: I/O 31 / S1, I/O 30 / S2, I/O 29 / S3)

Note:

- I/O pins are counted in the clockwise direction, where the first I/O pin (I/O 0) is top right. This is in contrast to conventional ICs where pins are counted anti-clockwise.
- The bottom-right pin is the PC Load pin.
- The 'S1, S2 and S3' (marked as LCD) pins are used to interface to the MT Serial LCD.
- The appropriate crystal is supplied with the MT chip.
- The user can disable the LCD functions and use them for normal I/O by connecting a 10K pull-down resistor on the S1 pin.
- A negative pulse on the WU pin will wake up the chip previously put into power down mode, then it continues execution where it has left off.

About MT chips memory

There are two types of memory within an MT chip. The first type is 'program memory' which is used to store the TinyC programs. Program memory is re-writable Flash memory and can be programmed more than 100,000 times although this number could be much larger. Since Flash memory is re-writeable, it is not necessary to erase the program storage area before another download.

Programs stored in Flash memory are non-volatile and will not 'disappear' when power is removed. The total number of program storage areas varies between different types of MT chips (see table).

The other type of memory is called 'run-time memory', commonly known as RAM. It is used to store the values of data variables and the MT system stack as the program runs. Each memory unit is 16-bits wide, allowing unsigned integer values from 0 to 65535 and signed integer values from and –32768 to +32767. Different types of MT chips have different amounts of run-time memory units (see table).

Serial Download Circuit

The serial download circuit is identical for all MT chips. It is a minimal configuration for a serial communication, consisting of just three wires from the MT chip to the serial port of a PC. One wire sends data from the computer to the serial input of the MT chip, one wire transmits data from the serial output of the MT chip to the computer and the third wire provides a common ground connection.

The PC COM port uses RS232 voltage levels so therefore it is necessary to have an RS232 voltage conversion circuit at the MT chip as well. The most commonly used circuit is a MAX232 IC with four small capacitors to act as charge pumps. A simpler circuit using just two transistors is also shown. It uses an electrolytic capacitor connected in the reverse direction to act as the charge pump. There are two RS232 channels in a single MAX232 chip but one is used. An alternative RS232 IC is the DS275. It has just one RS232 channel but can be difficult to get.

Any general purpose PNP and NPN transistors can be used for the transistor circuit.
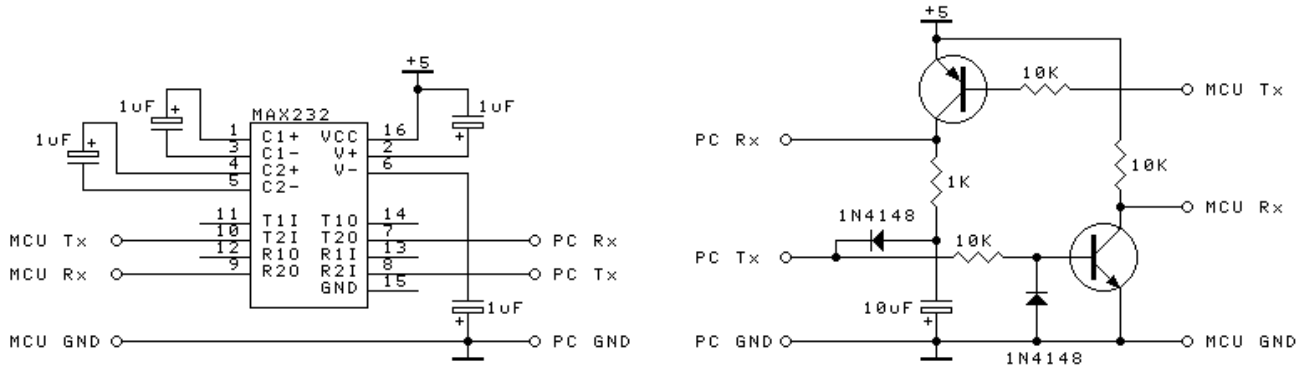
Reset Circuit

The reset circuit shown is suitable for all 8051 based systems. The capacitor acts as a delay timer which keeps the reset pin in a logic HIGH state to reset the MCU. It changes to a logic LOW state as it charges through the 10K resistor. In most cases this resistor can be omitted as the charge current can be provided by the reset pin of the MCU, but it is recommended to keep the resistor for increased stability. The reset pin must be kept LOW for the MCU to run properly. The push button is used to discharge the capacitor and manually reset the MCU.
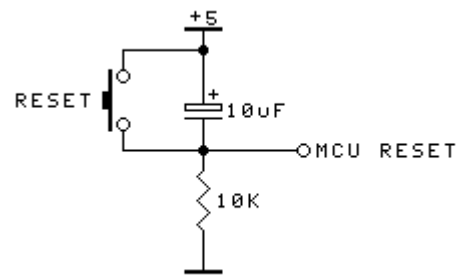
The MT-40-AD chip uses a negative reset pulse. In this case simply swap the VCC and GND connections. The MT-08 and MT-14 chips do not have a reset pin - the only way to reset them is to power them off then on again.

Crystals

Apart for the MT-14 and MT-08 chips, which use an internal 8MHz oscillator, the MT chips require an external crystal (either 22.1184MHz or 11.0592MHz) to operate. The crystal is required because the time related functions such as delay(), sound(), servo(), ircode(), setcomm(), etc, require an accurate timebase and are designed to operate at these frequencies. These functions will not work correctly if other crystal frequencies are used. The crystal frequencies used allow accurate baud rates for serial communication.
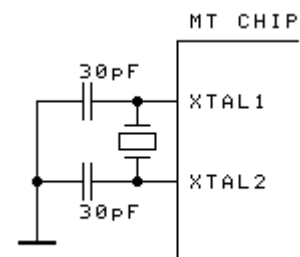
Normally the crystals require a load capacitor for proper startup and operation. However only the following MT chips require them: MT-40r1, MT-40-XX, MT-40-XP, MT-40-XA and MT-40-AD. Capacitor values can range from 22 – 33pF.

Power Supply

Most logic gates require a 5V DC power supply for proper operation. This includes the MT chips although they are able to operate at slightly lower voltages. However it is strongly recommended that a 5V power supply be used. This enables other circuits, such as the IR decoder and the serial communication to function correctly. These circuits do not work properly at voltages less than 5V.

Note: Never connect more than 5VDC directly to a MT chip, otherwise the IC may get damaged.

A simple 5V power using a 7805 voltage regulator is shown at right. All the MT Demo Boards are fitted with this chip. The 7805 regulator is able to provide current up to 1A and can take input voltages up to 35V DC. However the higher the input voltage the greater the power dissipated in the regulator. Since the regulator needs at least a 2V voltage drop across it the minimum input voltage should be 7V. A 9-12V DC wall adapter will do the job nicely. If the regulator gets hot then a heatsink will be required.
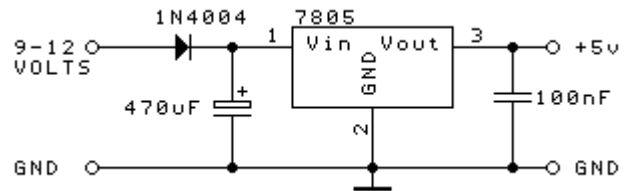
R/C servos and DC motors can use a lot of current when they move so a separate power source should be used to power them. In this case the negative ends of both the 5V and motor power supplies should be connected together to provide a common 'ground' reference point.

Starting out with the MT system

When starting out with the MT system, a MicroBoard or a project pack is recommended. These packages use the same software and include manuals, schematics, programming software and lots of sample programs, MT serial LCD module, battery cable and a serial download cable. However, the demo board and type of MT chip varies in each pack as indicated below.

MB-1 MicroBoards fitted with MT-40r1 chip, battery and serial download cables.
MB-2 MicroBoards fitted with MT-40r2 chip, battery and serial download cables.
MT-20 Project Pack, MT-20 demo board, MT-20 chip, serial LCD, battery and serial download cables.
MT-28 Project Pack, MT-28 demo board, MT-28 chip, serial LCD, battery and serial download cables.
MT-40 Project Pack, MT-40 demo board, MT-40 chip, serial LCD, battery and serial download cables.

All the boards have their own datasheets containing connection details, circuit diagrams etc. These documents and programming software are available for download from MicroTech website at www.mcu.hk.

Developing programs for MT Chips

MT chips are designed to execute programs written in the MicroTech **TinyC** language. The MicroTech Editor is provided for program development. It provides an Integrated Development Environment (IDE) with built in text editor and downloader. The IDE invokes an external compiler program, 'cmp.exe', to compile the TinyC programs. The downloader transfers the compiled program to the MT chip.

Developing TinyC programs for MT chips requires the following items:
1. PC running Microsoft Windows 98, 2000, XP, etc
2. MicroTech Editor software package
3. MT microcontroller chip
4. MT demo board or a breadboard / PCB with a RS232 circuitry
5. 5V DC power supply
6. MT serial download cable
7. MT serial LCD unit (optional but highly recommended)

Apart from items 1 and 5, the rest are included in all MT Project Packs. The MT serial LCD unit is strongly recommended for displaying messages when downloading and running programs. The supplied serial download cable has a 9-pin 'D' connector for direct connection to a PC COM port. If your computer does not have this type of connector then a USB to serial adapter/converter is needed.

Software Installation

The MicroTech IDE software is freely available and the latest version can be downloaded from the MicroTech website at www.mcu.hk. To install the software you require a computer running Windows 98 or later with approximately 5MB free space.

1. Download and unzip the MT software pack to the hard disk.
2. There is no install or setup program to run. Simply create a shortcut to the editor software, 'Microtech.exe', and place it on the desktop for easy access.
3. Double click on the desktop shortcut to run the Editor.
4. Click 'File → Options' to change the optional settings.

    The most important setting here is the COM port number. The software will automatically display all the available COM ports on the computer. Select the COM port that will be used for downloading.

    Change the 'Delay Timing' to adjust the downloading speed. The smaller the number the faster the download speed. However setting too fast a speed will corrupt the download process.

    Change the 'Working Directory' to the directory where the sample programs are located. Click the 'Save Settings' button to save the new settings.

Developing programs for MT chips

1. The MicroTech Editor invokes the external program, 'cmp.exe', to compile programs.
2. TinyC programs are simply text files - any text editor can be used.
3. Right clicking anywhere on the text will invoke the popup command menu.
4. Compile the program.
5. Programs are saved automatically before being compiled.
6. Download the program to the MT chip using the built-in downloader.

    There is an external command line downloader, 'Downloader.exe', for transferring programs into MT chips, and it takes parameter settings from the 'Microtech.ini' file. 'cmp.exe' and 'Downloader.exe' are provided so that users can integrate them into their preferred programming editors for program development. Output messages from these commands are stored in files CMP.TMP and DL.TMP.

Running programs using the MT chips Simulator

The Simulator enables users to develop and test programs without the need for any hardware. It does not provide any debugging tools such as breakpoints or single stepping but it does support the various MT chip I/O functions.

1. Compile the TinyC program.
2. Click on the Simulator button on the Editor menu bar. You will be prompted for the name of the program to simulate (defaults to the current file). Select the file and click 'Open'.
3. Click the 'Run' button to run the program.
4. Note that, the maximum code size that the Simulator can handle is about 4K words.

The timings and delays in the simulator are different to those in real MT chips. The simulator is used to show that a TinyC program is running and working.

All the TinyC functions are simulated, including the 5 I/O pin interrupts, the timer interrupt (partly) and reading and writing data to serial EEPROMs. The data for the EEPROMs are stored in the 24C02.DAT and 93C56.DAT files.

Press the buttons within the Demo box to see a demonstration of the various functions. They are disabled when a TinyC program is running, and re-enabled when the program is halted by the STOP button.

Click the I/O Pin boxes to set the pin states. If the box is checked then the output is low (0V).

Use the Sliding Bar to set values for the readadc() function.

Use the keypad (bottom-left) to input values for ircode(), getkey(), scankey() and serial readm() functions. The SYS and USER buttons are tied to I/O pins 19 and 18, respectively. They toggle the pin state each time they are clicked. **Note:** On a real MicroBoard, these 2 keys are activated only when they are kept pressed down.

Output to serial port by all the writeX() functions goes to the COM Output box.

The values shown in the Sound and Servo boxes are the I/O pin numbers, frequencies and degrees for sound() and servo() functions.

Function writec() does not clear the LCD before or after displaying data.

The MT Serial LCD unit controlling commands are not simulated yet!

Click on the Exit button to save the EEPROM data to disk files and exit the simulator.

Downloading, storing and running programs for MT chips with single storage area

1. Connect the download cable to a COM port on the PC.
2. Pull the PC pin (bottom right pin) LOW (connect to ground).
3. Switch on or press the Reset button on the Demo Board or User breadboard.

    The MT-40-AD chip needs a negative pulse for reset. There is no reset pin on MT-08 and MT–14 chips. The only way to reset them is to power off then on again.
4. The "PC" message will be displayed on the serial LCD if connected.
5. Click the Download button on the Editor menu bar.
6. After a successful download, the message, "OK" appears. Soon after, another message, "Run" appears.
7. A number is displayed on the serial LCD indicating the amount of free run-time memory units available.
8. Program starts running.

Downloading, storing and running programs for MT chips with multiple storage areas

1. The MT-40 series chips might need pull-up resistors (5K~10K) on I/O pins 0-7 to operate properly.
2. Connect the download cable to a COM port on the PC.
3. Pull the PC pin (bottom right pin) LOW.

    By default the program is downloaded to storage area 0. To use another storage area pull the desired program storage area pin LOW at the same time as the PC pin.
4. Switch on or press the reset button on the Demo Board.
5. The "PC-" message will be displayed on the LCD.
6. Click the Download button on the Editor menu bar.
7. The "PC-x" message will be displayed on the LCD, where x is the selected storage area.
8. After a successful download, the message, "OK" appears. Soon after, another message, "Run-x" appears.
9. A number appears indicating the amount of free run-time memory units.
10. Program starts running.

Tips:
*   Keep the program storage area pin LOW to select it for running.
*   The PC pin needs to keep LOW only for downloading purpose.
*   The default program storage area and 'program to run' is 0.

- Instead of setting the I/O pins manually, compiler directives can be used to specify the storage area to store the program and the default program to run after reset. However, the PC pin still be pulled LOW to initiate the download process.

-

Downloading and storing programs for MicroBoard

1. Connect the download cable to a COM port on the PC.
2. Switch on or press the reset button on the MicroBoard.
3. Press the PC button until message, "PC 0-4" appears then press button 0 - 4 to select the program storage area. Wait until the selected digit disappears on the display before starting to download.
4. Click the Download button on the Editor menu bar.
5. The red and green LEDs on the MicroBoard will flash as the program is downloaded.
6. After a successful download, the message, "Done" appears.
7. Soon after, the message "Run 0-4" appears. Press button 0 - 4 to select the program to run.
8. A number appears indicating the amount of free run-time memory units.
9. Program starts running.

Note: During download processes described above the error message, "Size Error", will be displayed if the size of the program is too large for the program storage area Another error message, "Version Error", will be displayed if the firmware of the MT chip is not compatible with the version of compiler.

Running programs for MicroBoard

1. Switch on or press the reset button on the MicroBoard.
2. Wait until message "Run 0-4" appears then press button 0 - 4 to select the program to run.
3. A number appears indicating the amount of free run-time memory units.
4. Program starts running.

INTERFACE CIRCUITS

This section describes various commonly used methods for interfacing with MT chips. **Note:** The MT-40xxx chips require pullup resistors on I/O pins 0-7 for true TTL logic level compatibility. A value of 10K is fine.

Output Switching

If the output devices do not draw too much current (i.e. <20ma), they can be connected directly to the I/O pins of MT chips. All the I/O pins of MT chips will be logic HIGH after power up or reset. Therefore the use of negative logic (pull pin LOW) for driving outputs is recommended and used. All transistors used here are of PNP type or you can use P-channel MOSFETs instead.

Some of the pins on MT chips are assigned for specific devices, such as the IR pin for IR detector. If you want to connect an IR detector it **must** be connected to this dedicated IR pin. Any other dedicated pins when not connected to their corresponding devices can be used as normal I/O pins.

A general transistor interfacing circuit is shown in Fig 1. This circuit can be used to drive low current devices, such as speakers, buzzers, etc. If the output device needs more current a commonly used method called "darlington pairing" is used (Fig 2). This consists of 2 transistors - the general low power transistor works as a pre-amp stage to drive a second power transistor which in turn drives the output device.
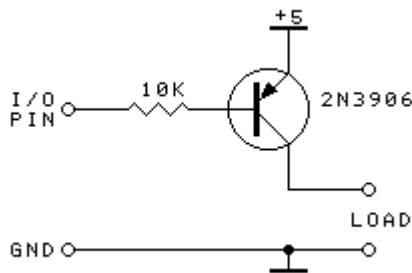
Fig 1
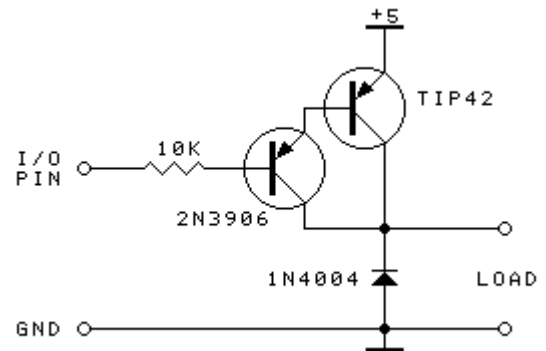
Fig 2

There are many darlington transistors available on the market which have internal circuitry similar to that above but packaged within a single case. Examples are: TIP107, TIP127 and TIP147.

If the output device is a relay, solenoid or motor then a reverse voltage called 'back EMF' will be generated whenever power to the device is removed (switched off). This back EMF voltage is very large and can damage the driving transistor. Therefore, it is necessary to connect a diode across the output device to suppress this back EMF. The general diode 1N4004 is commonly used, but a fast recovery diode may be needed if operating in a fast environment.
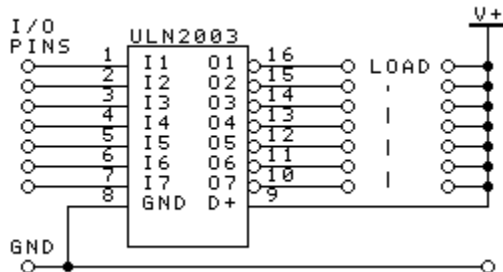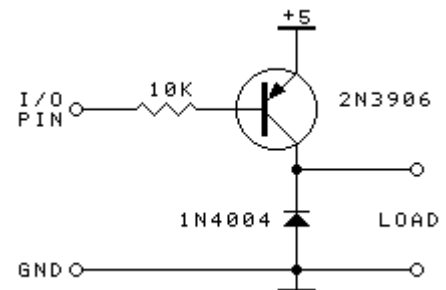

A number of darlington paired transistors packaged in a single IC are also available, such as ULN2003 and ULN2803. Both have back EMF suppression diodes built-in which make them very convenient when driving a number of outputs simultaneously. ULN2003 has 7 output channels, whereas ULN2803 has 8 channels and each channel is able to provide up to 500mA of current. The outputs are 'open collector' which means multiple outputs can be tied together to provided extra current switching capability.

Another feature of these chips is that they are able to switch loads at greater than 5 volts eg. 12/24V relays.

Note: Both these ICs use a HIGH on the input to activate the output. All MT chip I/O pins are HIGH on power up or reset.This means that when connected to an MT chip the output(s) will be activated (on).

In order to control high current power devices, a P-channel power MOSFET is used instead of a darlington transistor. FETs are voltage controlled devices whereas normal bipolar transistors are considered to be current controlled devices. Power MOSFETs have very low internal resistance so can handle much higher currents without getting too hot.

Sometimes it is necessary to electrically isolate the output driving circuit from the rest of the MCU circuit, such as when switching mains voltage to household lights or appliances. In this case, commonly used methods include the use of an opto-coupler or relay.
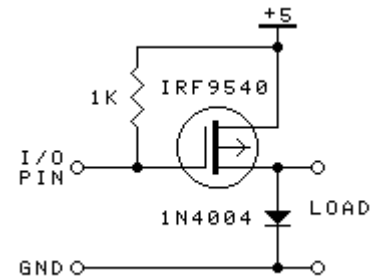
Light Emitting Diode (LED)

In MCU projects LEDs are often used as indicators. LEDs draw very little current and therefore they can be connected directly to the I/O pins of the MCU. Resistors ranging in value from 200 to 1K are connected in series with the LEDs to limit the current flow. Bicolour or tricolour LEDs are also available. These LEDs emit different coloured light depending on the direction of current flow or which pins are used.
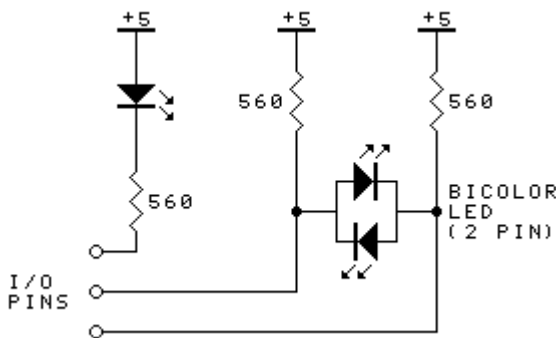
The MT chips can sink up to 20mA but can only source around 40uA. So the normal method of driving LEDs is to have a pullup resistor to 5V and pull them low, as shown.

Bicolour LEDs are available in both 2 and 3 pin versions. The 2 pin version is the most common. Changing the LED colour in the 2 pin version requires reversing the direction of current flow throught the LED. The circuit shows a simple method of reversing current flow through the LED - just output a low on the appropriate I/O pin. Only one of the I/O pins connected to the bicolour LED can be low at any one time. If both are high or low then the LED will be off.

Buzzer and Speaker

The difference between a buzzer and a speaker is simply that a buzzer has a built-in circuit which vibrates and produces sound at a fixed frequency, whereas a speaker must be supplied with a stream of electrical pulses in order to make a sound.

If the buzzer or speaker does not draw too much current, they can be connected directly to the MCU I/O pins with the 'other' side connected to 5V. Otherwise the standard transistor interfacing circuits shown earlier can be used.

The TinyC function '**sound()**' takes values for frequency and duration as parameters and drives a speaker to produce an audible tone.

IR Detector

The following circuit can be used for most IR detector integrated ICs but you will need to verify with the datasheet to determine the pinout. The resistor/capacitor combination acts as a low pass filter to provide a noise free supply for the IR module. You might need to change the values of the resistor and capacitor as required. The LED part of the circuit is optional and serves as an indicator for the presence of an incoming IR beam.

The TinyC function '**ircode()**' is used to read any IR codes detected by the module.

**Note**: The diagram shown a typical IR module viewed from it's back!
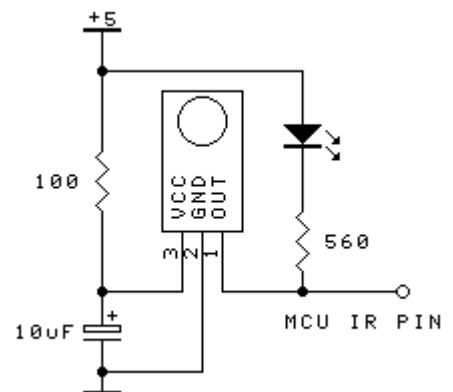
Liquid Crystal Display (LCD) Modules

An LCD is superior to LED displays in terms of power consumption and the ability to display the full ASCII character set or even graphics. They are available in a range of sizes - 1, 2 or 4 lines with 8, 16, 20 or 40 characters per line.

LCD modules are supplied with an 8-bit parallel data interface plus 3 control lines which requires a lot of MCU I/O pins to make them work. To ease the task and reduce the number of I/O pins required, a serial LCD controller was developed using just 3 control lines plus 2 power supply lines. It enables MT chips to send characters and control functions to an LCD very easily.

The following circuit shows how to connect the MT serial LCD to an MT chip. Note that on the serial LCD module it is necessary to connect pins 4 and 5 (labelled 'MT' on the board) together. This selects the MT communication protocol. A document describing the MT protocol can be downloaded from the website.
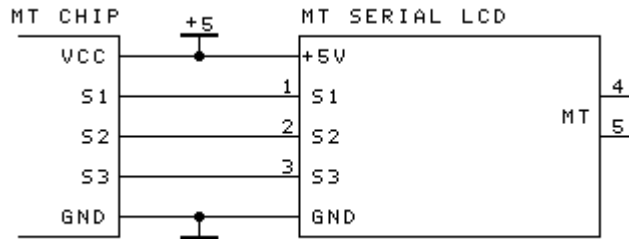
Of course a standard LCD module can be connected directly to the MT chip but you will need to write all the control functions yourself ie. initialise the LCD, clear the screen, write character, etc. The following circuit shows the pin connections required to do this. The 10K trimpot is needed to provide a 'contrast' adjustment for the LCD.
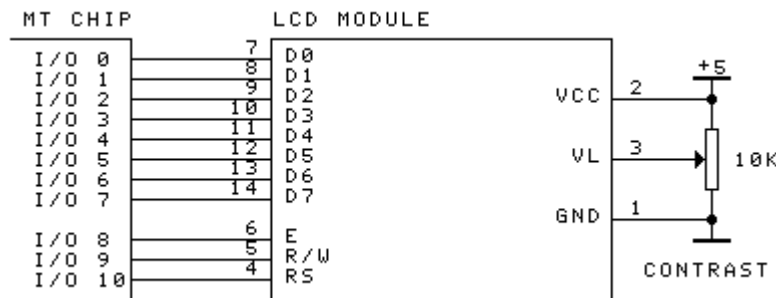
Any I/O pins can be used for the LCD control lines E, R/W and RS. However I/O pins 0-7 are recommended for the 8-bit data lines to the LCD because the TinyC function **writeb()** function outputs a byte to these pins. This greatly simplifies writing data to the LCD module. Without this function the user program would need to output the data byte one bit at a time.

Refer to the sample '**lcd1.prg**' program on how this setup works.

7-segment LED Displays

7-segment LED displays have a number of advantages over LCD types. They are much easier to see under bright lighting conditions, they have a wider viewing angle and they are available in physically larger sizes, making them easier to read at longer distances. They are also more robust than the LCD types and are available in a number of different colours.

A 7-segment displays consist of eight individual LEDs arranged so that they can display numbers and hexadecimal digits from 0-F. Each segment is identified by a letter 'A, B, C, D, E, F, G' plus 'DP' for the decimal point.

One end of all the LEDs are connected together to provide a 'common' point. If all the positive ends (anodes) are connected together it is called '**common anode**'. If all the minus ends (cathodes) are connected together it is called '**common cathode**'. The choice to use common anode or common cathode depends on your particular circuit. The MT chips can 'sink' more current than they can 'source' so common cathode is more suitable. In this case a low on the I/O pin will turn the segment on and a high will turn it off. The 'common anode' pin is connected to +5 volts.
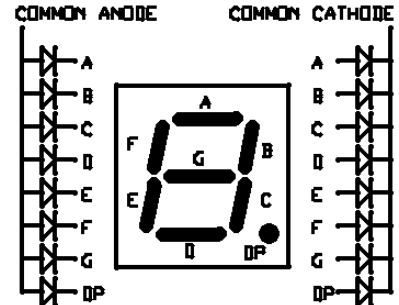
A major disadvantage of using 7-segment displays is that they require lots of I/O pins to drive them. Each display digit requires eight pins to control it, one for each segment. For a 4-digit display that adds up to 32 I/O pins!

One common method used to reduce pin count is to connect the segments pins of each display together. The 'common' pin of each display is connected separately so that each digit can be controlled individually. The technique used to display numbers using this arrangement is called '**multiplexing**', where each digit is only turned on for a short period of time ie. each digit is is switched on and off rapidly in succession. The persistence of the human eye makes it appear the all the digits are on at the same time.

The following schematic shows how to connect up a 4-digit 7-segment display to MT chips. The sample program '**7seg.prg**' (found in the TinyC installation folder) demonstrates how to control them.

The 8 x 100 ohm resistors are there to protect the LEDs. If the program stops or 'hangs' for some reason then one digit will be permanently on. The resistors will limit the current through each LED to a safe level.
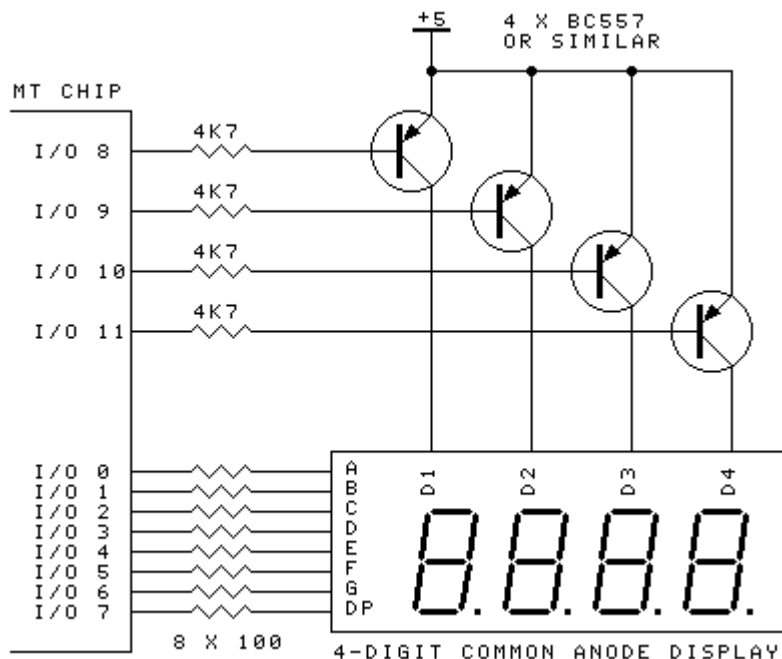
Analog-to-Digital Converter (ADC)

Potentiometers, LDRs (light dependent resistors) and thermistors are devices whose resistance changes according to some external condition. Thermistors are used to measure temperature. LDRs are used to measure light levels. This changing resistance is used to produce a variable DC voltage which is proportional to the parameter being measured.

The circuit at right gives an example of how to connect up these devices so that they produce a varying DC output voltage. The resistor can be in the range from 1K to 100K depending on the type of device used. Change it as required to give optimal readings.

Since MCUs are digital in nature, these analog readings must be converted into a digital form that the MCU can understand. Analog-to digital converters (ADCs) are used to for this purpose.
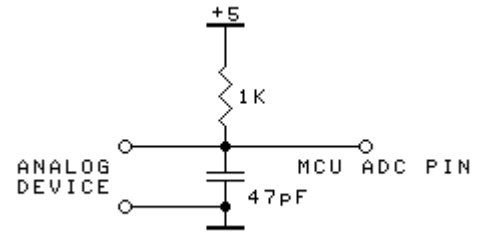
The ADCs available in some MT chips offer 10-bit resolution, meaning that the input DC voltage is converted into a number ranging from 0 to 1023 (2^{10}). The ADC determines this number by comparing the input voltage against a reference voltage, 5 volts in this case. The following formula is used to determine the input voltage based on the ADC output.

$$Vin = \frac{NUM}{1024} \, x Vref$$ where NUM is the number output by the ADC and Vref = 5V.

Example: If the ADC output is 750 then Vin = 3.6621 volts

The TinyC function **readadc()** returns the ADC output number (on MT chips where an ADC is available).

Serial Communication

Serial communications is a universal standard for transferring data from one MCU or computer to another. If both devices use digital signals, such as two MCUs, then they can be connected directly without the need for any voltage conversion circuit.

However many devices, including PCs, use RS232 voltage levels on their serial COM ports. RS232 uses –12V to represent a logic '1' and +12V for a logic '0'. In this case an RS232 voltage conversion circuit is required to interface the MT chip serial port to the PC.

The most commonly used circuit is a MAX232 IC with four small capacitors to act as charge pumps. A simpler circuit using just two transistors is also shown. It uses an electrolytic capacitor connected in the reverse direction to act as the charge pump. There are two RS232 channels in a single MAX232 chip but one is used. An alternative RS232 IC is the DS275. It has just one RS232 channel but can be difficult to get.

Any general purpose PNP and NPN transistors can be used for the transistor circuit.
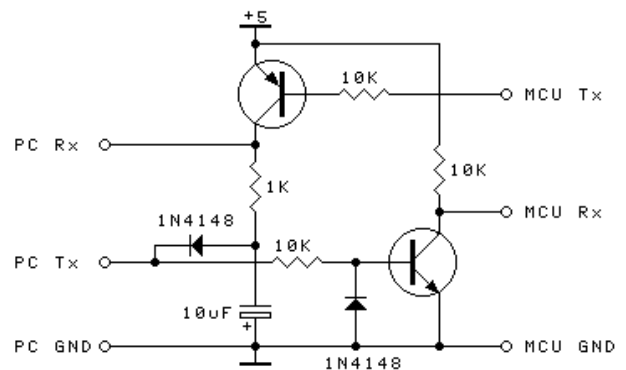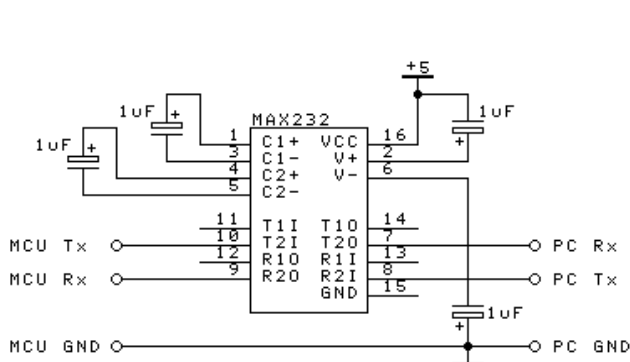
Serial EEPROMs

MT chips provide specific I/O pins for interfacing to 93Cx6 and 24C0x serial EEPROMs. The 93Cx6 EEPROM uses a 4-wire protocol to transfer data whereas the 24C0x EEPROM uses the 2-wire I2C protocol. Please refer to the EEPROM datasheet for further information.

The addressing scheme implemented in MT chips is only 8 bits long, therefore the address space is in the range of 0-255 (256 bytes). The TinyC functions '**readm()**' and '**writem()**' are used for reading and writing 8-bit data to the EEPROMs.

Note: The 93Cx6 EEPROM can operate in either 8-bit or 16-bit data mode. MT chips only support 8-bit mode so make sure the EEPROM is configured correctly (ORG pin connected to ground).
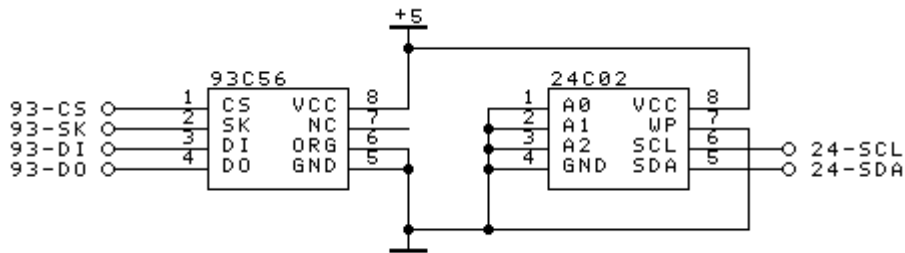
Radio Control Servo Motors

Servos are electronics devices which convert electrical signals into mechanical movement. They are often used and found in radio controlled cars and airplanes. Inside a servo is a potentiometer connected to the motor shaft that turns as the motor turns. The output of the potentiometer allows the internal servo circuitry to determine how far the motor has turned (its position) and compare it against an incoming signal telling it where it should be.

A typical servo has three wires, two for power to the motor and circuitry and the other for the incoming control signal. The control signal is a positive pulse whose width varies between 0.75ms and 2.25ms, repeated about every 20ms. The pulse width determines the servo's position. A pulse width of 1.5ms moves the servo to its centre position.

Normally servos they require a large current (~1A) to power them but this can vary depending on the servo. Power is often supplied from rechargeable batteries.

The TinyC **servo()** function is used to send control signals to servos. The **servo()** function is able to control individual or multiple servos. See the example programs for a demonstration on how to do this.
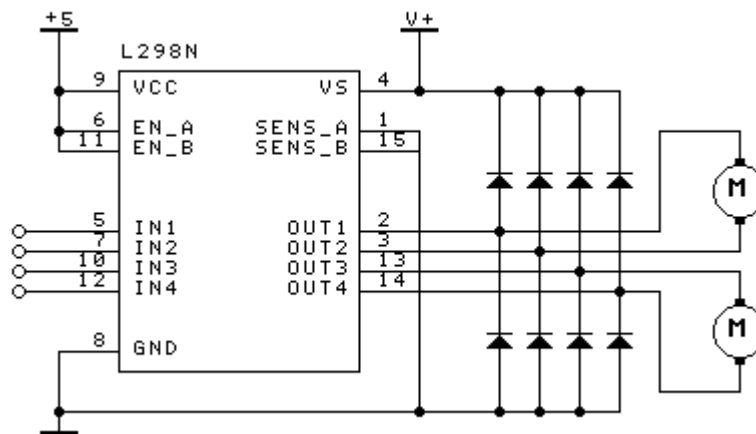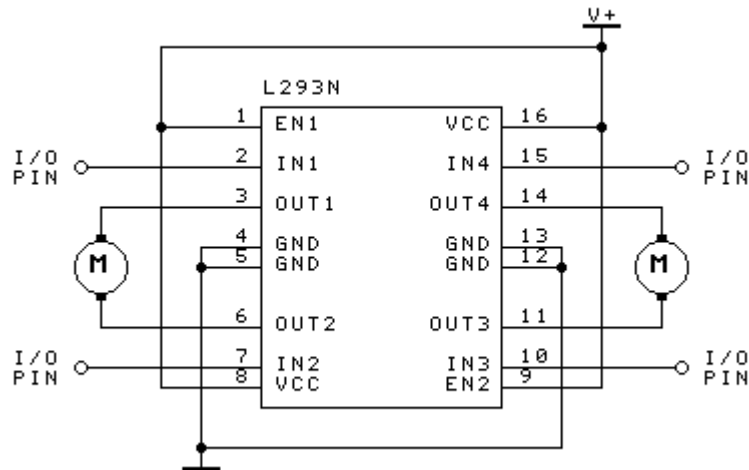
Controlling DC and Stepper Motors

Both DC and stepper motors can be interfaced with MT chips. Stepper motors are used to control the movements of printers, floppy disks and hard disks. Stepper motors move in 'steps' where each 'step' represents a degree of angular movement. For example, a 72-step stepper motor moves 5 degrees per step; a 100 step motor moves 3.6 degrees per step.

Stepper motors come in two types, unipolar and bipolar, depending on the method use to control them. Both types have two coils to make them step. In unipolar motors the coils are centre-tapped giving a total of four coils. The centre-taps are tied together and connected to a V+ voltage. The motor is stepped by energising any two coils in sequence. Bipolar motors have just two coils and are stepped by reversing the voltage polarity on each coil in sequence.

Since motors are magnetic in nature, a back EMF suppression diode is needed for each coil in the motor. The transistor or MOSFET circuits shown previously can be used to control DC motors. However the drawback with this method is that the motor only turns in one direction. In order to control a motor to spin in both directions (or for controlling bipolar steppers) a circuit called the H-bridge is used. The H-bridge allows the voltage polarity across a DC motor to be reversed.

Many H-bridge motor controller ICs, such as the L293D, L298 and LMD18200, make the task of controlling DC motors much easier. The L293D is used on the MicroBoard. It has built-in suppression diodes and is the most popular motor controller available.

In both the above circuits the motors could be individual DC motors or the coils of a bipolar stepper motor. Therefore each circuit can control either two DC motors or a single bipolar stepper motor.

DC and stepper motors can also be controlled using the MT Motor Controller board. The MT Motor Controller uses a 3-wire interface and protocol similar to the serial LCD controller to control DC and stepper motors. However it cannot be used on the 3-wire 'LCD' interface described earlier – it will not work. A document describing the MT protocol can be downloaded from the website.

A sample program '**mctrl.prg**' is given showing how to communicate with this board.