1
Introduction to the PIC microcontroller

A microcontroller is a computer control system on a single chip. It has many electronic circuits built into it, which can decode written instructions and convert them to electrical signals. The microcontroller will then step through these instructions and execute them one by one. As an example of this a microcontroller could be instructed to measure the temperature of a room and turn on a heater if it goes cold.

Microcontrollers are now changing electronic designs. Instead of hard wiring a number of logic gates together to perform some function we now use instructions to wire the gates electronically. The list of these instructions given to the microcontroller is called a program.

The aim of the book

The aim of the book is to teach you how to build control circuits using devices such as switches, keypads, analogue sensors, LEDs, buzzers, 7 segment displays, alpha-numeric displays, radio transmitters etc. This is done by introducing graded examples, starting off with only a few instructions and gradually increasing the number of instructions as the complexity of the examples increases.

Each chapter clearly identifies the new instructions added to your vocabulary.

The programs use building blocks of code that can be reused in many different program applications.

Complete programs are provided so that an application can be seen working. The reader is then encouraged to modify the code to alter the program in order to enhance their understanding.

Throughout this book the programs are written in a language called assembly language which uses a vocabulary of 35 words called an instruction set. In order to write a program we need to understand what these words mean and how we can combine them.

The complete instruction set is shown in Chapter 19 Instruction Set, Files and Registers.

All of the programs illustrated in the book are available from:
http://books.elsevier.com/uk//newnes/uk/subindex.asp?maintarget=
companions/defaultindividual.asp&isbn=0750648120

You will of course need a programmer to program the instructions into the chip. The assembler software, MPASM, which converts your text to the machine code is available from Microchip on www.microchip.com this website is a must for PIC programmers.

Program memory

Inside the microcontroller the program we write is stored in an area called EPROM (Electrically Programmable Read Only Memory), this memory is non-volatile and is remembered when the power is switched off. The memory is electrically programmed by a piece of hardware called a programmer.

The instructions we program into our microcontroller work by moving and manipulating data in memory locations known as user files and registers. This memory is called RAM, Random Access Memory. For example in the room heater we would measure the room temperature by instructing the microcontroller via its Analogue to Digital Control Register (ADCON0) the measurement would then be compared with our data stored in one of the user files. A STATUS Register would indicate if the temperature was above or below the required value and a PORT Register would turn the heater on or off accordingly. The memory map of the 16F84 chip is shown in Chapter 6.

PIC Microcontrollers are 8 bit micros, which means that the memory locations, the user files and registers are made up of 8 binary digits shown in Figure 1.1.

Bit 0 is the Least Significant Bit (LSB) and Bit 7 is the Most Significant Bit (MSB).

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
1	0	1	1	0	0	1	0

MSB -- LSB

Figure 1.1 User file and register layout

The use of these binary digits is explained in Appendix C.

When you make an analogue measurement, the digital number, which results, will be stored in a register called ADRES. If you are counting the number of times a light has been turned on and off, the result would be stored as an 8 bit binary number in a user file called, say, COUNT.

Microcontroller clock

In order to step through the instructions the microcontroller needs a clock frequency to orchestrate the movement of the data around its electronic circuits. This can be provided by 2 capacitors and a crystal or by an internal oscillator circuit.

In the 16F84 microcontroller there are 4 oscillator options.

- An RC (Resistor/Capacitor) oscillator which provides a low cost solution.
- An LP oscillator, i.e. 32kHz crystal, which minimises power consumption.
- XT which uses a standard crystal configuration.
- HS is the high-speed oscillator option.

Common crystal frequencies would be 32kHz, 1MHz, 4MHz, 10MHz and 20MHz.

Newer microcontrollers, such as the 16F818 and 12F629, have an oscillator built on the chip so we do not need to add a crystal to them.

Inside the Microcontroller there is an area where the processing (the clever work), such as mathematical and logical operations are performed, this is known as the central processing unit or CPU. There is also a region where event timing is performed and another for interfacing to the outside world through ports.

The microcontroller system

The block diagram of the microcontroller system is shown in Figure 1.2.

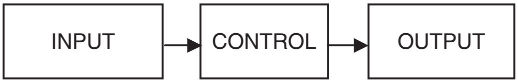

Figure 1.2 The basic microcontroller system

- The input components would consist of digital devices such as, switches, push buttons, pressure mats, float switches, keypads, radio receivers etc. and analogue sensors such as light dependant resistors, thermistors, gas sensors, pressure sensors, etc.
- The control unit is of course the microcontroller. The microcontroller will monitor the inputs and as a result the program would turn outputs on and off. The microcontroller stores the program in its memory, and executes the instructions under the control of the clock circuit.
- The output devices would be made up from LEDs, buzzers, motors, alpha numeric displays, radio transmitters, 7 segment displays, heaters, fans etc.

The most obvious choice then for the microcontroller is how many digital inputs, analogue inputs and outputs does the system require. This would then specify the minimum number of inputs and outputs (I/O) that the microcontroller must have. If analogue inputs are used then the microcontroller must have an Analogue to Digital (A/D) module inside.

The next consideration would be what size of program memory storage is required. This should not be too much of a problem when starting out, as most programs would be relatively small. All programs in this book fit into a 1k program memory space.

The clock frequency determines the speed at which the instructions are executed. This is important if any lengthy calculations are being undertaken. The higher the clock frequency the quicker the micro will finish one task and start another.

Other considerations are the number of interrupts and timer circuits required, how much data EEPROM if any is needed. These more complex operations are considered later in the text.

In this book the programs requiring analogue inputs have been implemented on the 16F818 and 16F872 micros. Programs requiring only digital inputs have used the 16F84 and 16F818. The 16F818 and 16F84 devices have 1k of program memory and have been run using a 32.768kHz clock frequency or the internal oscillator on the 16F818. There are over 100 PIC microcontrollers, the problem of which one to use need not be considered until you have understood a few applications.

Types of microcontroller

The list of PIC Microcontrollers is growing almost daily. They include devices for all kinds of applications, for example the 18F8722 has 64k of EPROM memory, 3938 bytes of RAM (User files), 1024 bytes of EEPROM, 16 10-bit

A/D channels, a voltage reference, 72 inputs and outputs (I/O), 3–16 bit and 2–8 bit timers.

There are basically two types of microcontrollers, Flash devices and One Time Programmable Devices (OTP).

The flash devices can be reprogrammed in the programmer whereas OTP devices once programmed cannot be reprogrammed. All OTP devices however do have a windowed variety, which enables them to be erased under ultra violet light in about 15 minutes, so that they can be reprogrammed. The windowed devices have a suffix JW to distinguish them from the others.

The OTP devices are specified for a particular oscillator configuration R-C, LP, XT or HS. See Appendix A Microcontroller Data.

16C54 configurations are:

16C54JW	Windowed device
16C54RC	OTP, R-C oscillator
16C54LP	OTP, LP oscillator, 32kHz
16C54XT	OTP, XT oscillator, 4MHz
16C54HS	OTP, HS oscillator, 20Mhz

In this book the two main devices investigated are the 16F84 and the 16F818 flash devices. The 16F84 at present is the main choice for beginners, but should be replaced in popularity by the better and cheaper 16F818. They have their program memory made using Flash technology. They can be programmed, tested in a circuit and reprogrammed if required without the need for an ultra violet eraser.

Microcontroller specification

You specify a device with its Product Identification Code.
This code specifies:

- The device number.
- If it is a Windowed, an OTP, or flash device. The windowed device is specified by a JW suffix. OTP devices are specified by Oscillator Frequency, and the Flash devices are specified with an F such as 16F84.
- The oscillation frequency, usually 04 for devices working up to 4MHz., 10 up to 10MHz or 20 up to 20MHz. 20MHz devices are of course more expensive than 4MHz devices.
- Temperature range, for general applications 0°C to +70°C is usually specified.

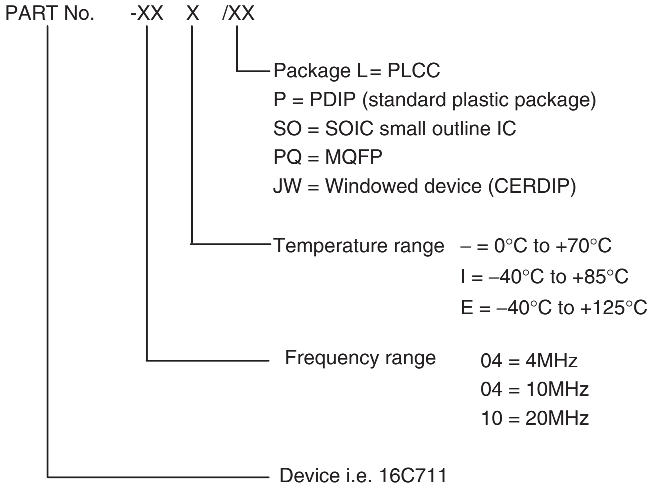
PART No. -XX X /XX
Package L= PLCC
P = PDIP (standard plastic package)
SO = SOIC small outline IC
PQ = MQFP
JW = Windowed device (CERDIP)
Temperature range – = 0°C to +70°C
I = –40°C to +85°C
E = –40°C to +125°C
Frequency range 04 = 4MHz
04 = 10MHz
10 = 20MHz
Device i.e. 16C711

Figure 1.3 Product identification system

The Product Identification System for the PIC Micro is shown in Figure 1.3.

Using the microcontroller

In order to use the microcontroller in a circuit there are basically two areas you need to understand:

1. How to connect the microcontroller to the hardware.
2. How to write and program the code into the microcontroller.

1 Microcontroller hardware

The hardware that the microcontroller needs to function is shown in Figure 1.4. The crystal and capacitors connected to pins 15 and 16 of the 16F84 produce the clock pulses that are required to step the microcontroller through the program and provide the timing pulses. (The crystal and capacitor can be omitted if using an on board oscillator in e.g. 16F818). The 0.1µF capacitor is placed as close to the chip as possible between 5v and 0v. Its role is to divert (filter) any electrical noise on the 5v power supply line to 0v, thus bypassing the microcontroller. This capacitor must always be connected to stop any noise affecting the normal running of the microcontroller.

Microcontroller power supply

The power supply for the microcontroller needs to be between 2v and 6v. This can easily be provided from a 6v battery as shown in Figure 1.5.

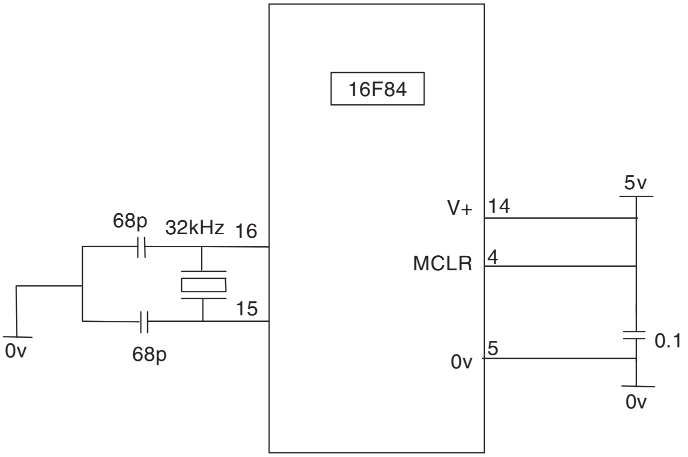

Figure 1.4 The microcontroller circuit

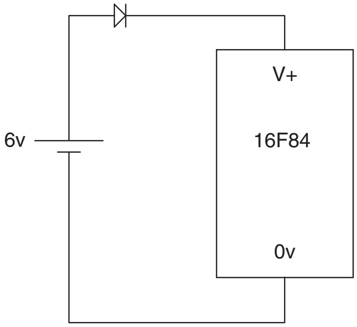

Figure 1.5 Microcontroller power supply

The diode in the circuit drops 0.7v across it reducing the applied voltage to 5.3v. It provides protection for the microcontroller if the battery is accidentally connected the wrong way round. In that case the diode would be reversed biased and no current would flow.

7805, Voltage regulator circuit

Probably the most common power supply connection for the microcontroller is a 3 terminal voltage regulator, I.C., the 7805. The connection for this is shown in Figure 1.6.

The supply voltage, Vin, to the 7805 can be anything from 7v to 30v.

The output voltage will be a fixed 5v and can supply currents up to 1amp. So battery supplies such as 24v, 12v, 9v etc. can be accommodated.

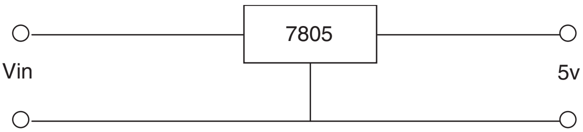

Figure 1.6 The voltage regulator circuit

Power dissipation in the 7805

Care must be taken when using a high value for Vin. For example if Vin = 24v the output of the 7805 will be 5v, so the 7805 has $24 - 5 = 19$v across it. If it is supplying a current of 0.5amp to the circuit then the power dissipated (volts × current) is $19 \times 0.5 = 9.5$watts. The regulator will get hot! and will need a heat sink to dissipate this heat.

If a supply of 9v is connected to the regulator it will have 4v across it and would dissipate $4 \times 0.5 = 2$watts.

In the circuits used in this book the microcontroller only requires a current of 15μA so most of the current drawn will be from the outputs. If the output current is not too large say <100mA (0.1A) then with a 9v supply the power dissipated would be $4 \times 0.1 = 0.4$watts and the regulator will stay cool without a heatsink.

Connecting switches to the microcontroller

The most common way of connecting a switch to a microcontroller is via a pull-up resistor to 5v as shown in Figure 1.7.

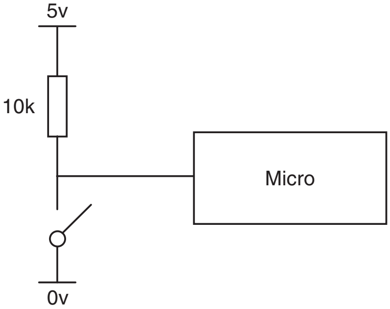

Figure 1.7 Connecting a switch to the microcontroller

When the switch is open, 5v, a logic 1 is connected to the micro.

When the switch is closed, 0v, a logic 0 is connected to the micro.

Some Microcontrollers such as the 16F84 and 16F818 have internal pull ups connected to some of their I/O pins. PORTB in the above devices.

Figure 1.8 shows how the switch is connected using the internal pull up.

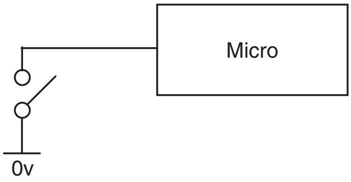

Figure 1.8 Connecting a switch using an internal pull up

Connecting outputs to the microcontroller

The microcontroller is capable of supplying approximately 20–25mA to an output pin. So loads such as LEDs or small relays can be driven directly. Larger loads require interfacing via a transistor, for dc or a triac, for ac. Opto-coupled devices provide an isolated interface between the microcontroller and the load.

The LED connection to the Micro is shown in Figure 1.9.

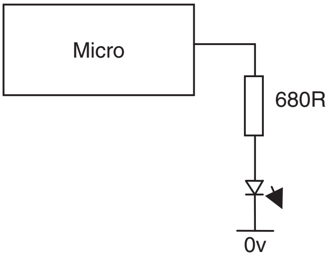

Figure 1.9 Connecting an LED to the microcontroller

2 Programming the microcontroller

In order to have the microcontroller perform some controlling action you need to communicate with it and tell it what those instructions are to be. When we communicate with one another we use a spoken language, when we communicate with a microcontroller we use a program language. The program language for the PIC Microcontroller uses 35 words (instructions) in its vocabulary. A few more instructions are used in the bigger microcontrollers.

In order to communicate with the microcontroller we need to know what these 35 instructions are and how to use them. Not all 35 instructions are used in this book. In fact you can write meaningful programs using only 5 or 6 instructions.