Chapter 1

Introduction to the Microcontroller and C

A microcontroller is an integrated circuit that has a number of memory locations embedded inside it which are used to store instructions that are to be executed. These locations are called registers, and instructions are written to these registers to enable the microcontroller to perform an operation.

The memory location is 8 bits wide which means it can store 8 bits of information (Figure 1.1). The 8 bits in the memory are identified by numbers starting on the right with the least significant bit, bit 0, and moving to the left to the most significant bit, bit 7.

Suppose we wish to turn on an LED connected to an output pin, as shown in Figure 1.2. An instruction has to be written to the output port register to output a logic 1 to turn the LED on or output a logic 0 to turn it off.

The microcontroller we will use in this book is a PIC18F1220 manufactured by Microchip, although the codes can easily be adapted for other Microchip microcontrollers. The PIC18F1220 has 16 inputs/outputs (I/O) which means it has 16 inputs or outputs which can be configured as inputs or outputs by instructing the microcontroller via a register, the tristate (TRIS) register (Figure 1.3). TRIS means the port pin can be (i) an input, or an output which is switched (ii) high or (iii) low, three states.

The memory locations in the microcontroller are 8 bits wide so 16 I/O will require two 8 bit registers called PORTA and PORTB.

Suppose we wish to turn on an LED which we are going to connect to bit 4 on PORTB. We first of all have to instruct the microcontroller to ensure that PORTB bit 4 is an output. At the moment it does not matter what the rest of

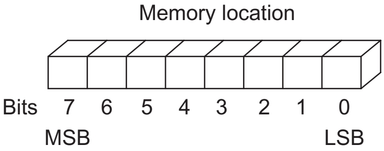

FIGURE 1.1 A microcontroller memory location.

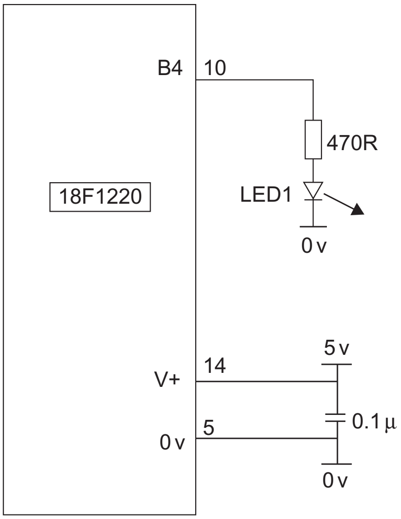

FIGURE 1.2 A basic microcontroller circuit.

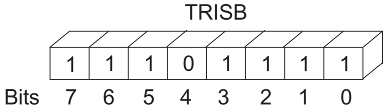

FIGURE 1.3 The TRIS register.

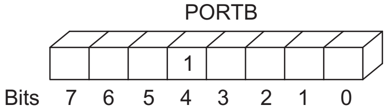

FIGURE 1.4 Writing to PORTB.

PORTB is doing, so now let's make bit 4 an output and the other 7 bits inputs. We do this with the following instruction:

TRISB = 0b11101111;

0b means the number is a binary one.

Note a 1 sets the pin as an input, a 0 sets the pin as an output.

Now that PORTB bit 4 is an output, we can write a logic1 to it with:

PORTBbits.RB4 = 1; (Figure 1.4).

There are several ways in which we can give the microcontroller instructions, called programming. These program languages are assembly, basic, C, or a number of flowchart languages. The language that we are going to use in this book is the C programming language, which is a high-level language that is very versatile. The previous book “PIC in Practice” written by the author, DW Smith, used the assembly language to program the microcontroller.

C is a very comprehensive and versatile language, which usually means there is a lot to learn. Throughout this book I will introduce the C language as and when required and only those instructions that are needed to perform the control. So you will not need to become a C programmer in order to program the micro in C!