11
Analogue to digital conversion

Up to now we have considered inputs as being digital in operation i.e. the input is either a 0 or 1. But suppose we wish to make temperature measurements, but not just hot or cold (1 or 0). We may for example require to:

(a) Sound a buzzer if the temperature drops below freezing.
(b) Turn a heater on if the temperature is below 18°C.
(c) Turn on a fan if the temperature goes above 25°C.
(d) Turn on an alarm if the temperature goes above 30°C.

We could of course have separate digital inputs, coming from comparator circuits for each setting. But a better solution is to use 1 input connected to an analogue to digital converter and measure the temperature with that.

Figure 11.1 shows a basic circuit for measuring temperature. It consists of a fixed resistor in series with a thermistor (a temperature sensitive resistor).

The resistance of the thermistor changes with temperature causing a change in the voltage at point X in Figure 11.1.

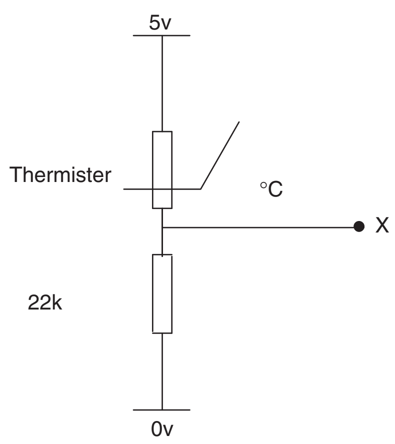

Figure 11.1 Temperature measuring circuit

As the temperature rises the voltage at X rises.
As the temperature decreases the voltage at X reduces.

We need to know the relationship between the temperature of the thermistor and the voltage at X. A simple way of doing this would be to place the thermistor in a cup of boiling water (100°C) and measure the voltage at X. As the water cools corresponding readings of temperature and voltage can be taken. If needed a graph of these temperature and voltage readings could be plotted.

Making an A/D reading

In the initial example let us suppose:

- 0°C gave a voltage reading of 0.6v
- 18°C gave a reading of 1.4v
- 25°C gave a reading of 2.4v
- 30°C gave a reading of 3.6v

The microcontroller would read these voltages and convert them to an 8-bit number where 0v is 0 and 5v is 255. I.e. a reading of 51 per volt or a resolution of 1/51v, i.e. 1 bit is 19.6mv.

So 0°C = 0.6v = reading of 31 ($0.6 \times 51 = 30.6$)
18°C = 1.4v = 71 ($1.4 \times 51 = 71.4$)
25°C = 2.4v = 122 ($2.4 \times 51 = 122.4$)
30°C = 3.6v = 184 ($3.6 \times 51 = 183.6$)

If we want to know when the temperature is above 30°C the microcontroller looks to see if the A/D reading is above 184. If it is, switch on the alarm, if not keep the alarm off. In a similar way any other temperature can be investigated – not just the ones listed. With our 8 bits we have 255 different temperatures we can choose from. The PIC 16C773 and PIC 16C774 have 12-bit A/D converters and can have 4096 different temperature points.

Analogue to Digital conversion was introduced to the PIC Microcontrollers with the family called 16C7X devices: 16C71, 16C73 and 16C74. Table 11.1 shows some of the specifications of these devices.

Table 11.1 16C7X Device specifications

Device	I/O	A/D Channels	Program Memory	Data Memory	Current Source/Sink
16C71	13	4	1k	36	25mA
16C73	22	5	4k	192	25mA
16C74	33	8	4k	192	25mA

This family of devices has now been superceded by the 16F87X devices shown in Table 11.2.

Table 11.2 16F87X Devices

Device	I/O	A/D Channels	Program Memory	Data Memory	Current Source/Sink
16F870	22	5	2k	128	25mA
16F871	33	8	2k	128	25mA
16F872	22	5	2k	128	25mA
16F873	22	5	4k	192	25mA
16F874	33	8	4k	192	25mA
16F876	22	5	8k	368	25mA
16F877	33	5	8k	368	25mA

The device I shall consider in this section is the 16F818. The Device Family Specifications are shown in Table 11.3.

Table 11.3 16F818/9 Device specifications

Device	I/O	A/D Channels	Program Memory	Data Memory	Current Source/Sink
16F818	16	5	1k	128	25mA
16F819	16	5	2k	256	25mA

The 16F818 device needs extra registers that the 16F84 does not have, to handle the A/D processing.

The 16F818 has 5 Analogue Inputs AN0, AN1, AN2, AN3 and AN4.

Configuring the A/D device

In order to make an analogue measurement we have to configure the device. HEAD818.ASM has to have the CONFIGURATION SECTION changed to make some of the PORTA inputs Analogue inputs. PORTB has been set as an output port.

To configure the 16F818 for A–D measurements three registers need to be set up.

- ADCON0
- ADCON1
- ADRES

ADCON0

The first of the A/D registers, ADCON0 is A to D Control Register 0.

ADCON0 is used to:

- Switch the A/D converter on with ADON, bit0. This bit turns the A/D on when set and off when clear. The A/D once it is turned on can be left on all of the time but it does draw a current of 90μA, compared to the rest of the microcontroller which draws a current of 15μA.
- Instruct the microcontroller to execute a conversion by setting the GO/DONE bit, bit2. When the GO/DONE bit is set the micro does an A/D conversion. When the conversion is complete the hardware clears the GO/DONE bit. This bit can be read to determine when the result is ready.
- Set the particular channel (input) to make the measurement from. This is done with two Channel Select bits, CHS0, CHS1 and CHS2, bits 3, 4 and 5.

The Register ADCON0 is shown in Figure 11.2.

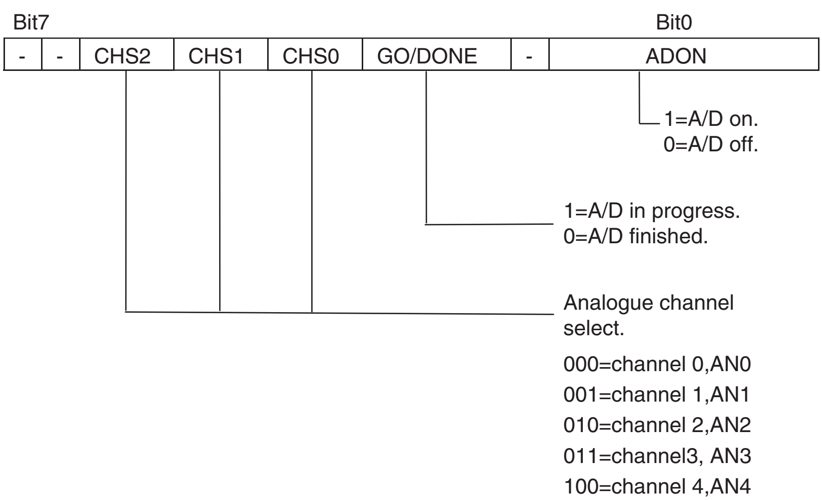

Figure 11.2 ADCON0 Register

ADCON1

In ADCON1, A to D Conversion Register 1, only bits 0, 1, 2 and 3 are used.

They are the Port Configuration bits, PCFG0, PCFG1, PCFG2, and PCFG3 that determine which of the pins on PORTA will be analogue inputs and which will be digital.

The ADCON1 register is illustrated in Figure 11.3 and the corresponding Analogue and Digital inputs are shown in Table 11.4.

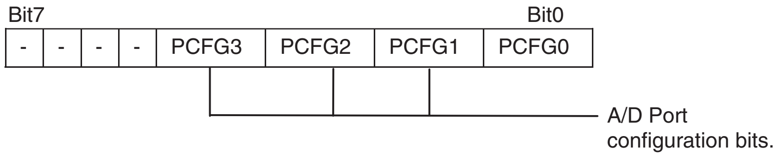

Figure 11.3 ADCON1 Register

Table 11.4 ADCON1 Port configuration

PCFG	AN4	AN3	AN2	AN1	AN0	Vref+	Vref–
0000	A	A	A	A	A	Vdd	Vss
0001	A	Vref+	A	A	A	AN3	Vss
0010	A	A	A	A	A	Vdd	Vss
0011	A	Vref+	A	A	A	AN3	Vss
0100	D	A	D	A	A	Vdd	Vss
0101	D	Vref+	D	A	A	AN3	Vss
011X	D	D	D	D	D	Vdd	Vss
1000	A	Vref+	Vref–	A	A	AN3	AN2
1001	A	A	A	A	A	Vdd	Vss
1010	A	Vref+	A	A	A	AN3	Vss
1011	A	Vref+	Vref–	A	A	AN3	AN2
1100	A	Vref+	Vref–	A	A	AN3	AN2
1101	D	Vref+	Vref–	A	A	AN3	AN2
1110	D	D	D	D	A	Vdd	Vss
1111	D	Vref+	Vref–	D	A	AN3	AN2

As mentioned previously the microcontroller will convert an analogue voltage between 0 and 5v to a digital number between 0 and 255. But suppose our analogue readings of say, temperature, go from 0.6v representing a temperature of 0°C to 3.6v representing a temperature of 30°C. It would make sense to have our analogue range go from 0.6v to 3.6v. We can set this by using two reference voltages. One at the low setting of 0.6v called Vref−, connected to AN2. The other setting of 3.6v for Vref+, connected to AN3. The two right hand columns in Table 11.4 show that PCFG Set at 1000 will set the A/D configuration using AN3 and AN2 as the reference voltages. In this book I have not used any reference voltages but have used 5v, Vdd and 0v. Vss as the references.

ADRES

- The third register is ADRES, the A to D RESult register. This is the file where the result of the A/D conversion is stored. If several measurements require storing then the number in ADRES needs to be transferred to a user file before it is overwritten with the next measurement. The 16F818 micro is a 10 bit A/D. The top 8 bits are stored in ADRESH and the lower 2 bits in ADRESL. In this book I am only using 8 bits and have called the file ADRES.

Analogue header for the 16F818

```
;HEAD818A.ASM for 16F818.     This sets PORTA as analogue/digital
;                             INPUTs.
;                             PORTB is an OUTPUT.
;                             Internal oscillator of 31.25kHz chosen
;                             The OPTION register is set to /256 giving
;                             timing pulses 32.768ms.
;                             1second and 0.5 second delays are
;                             included in the subroutine section.

;*******************************************************

; EQUATES SECTION

TMR0        EQU     1       ;means TMR0 is file 1.
STATUS      EQU     3       ;means STATUS is file 3.
PORTA       EQU     5       ;means PORTA is file 5.
PORTB       EQU     6       ;means PORTB is file 6.
ZEROBIT     EQU     2       ;means ZEROBIT is bit 2.
ADCON0      EQU     1FH     ;A/D Configuration reg.0
ADCON1      EQU     9FH     ;A/D Configuration reg.1
ADRES       EQU     1EH     ;A/D Result register.
CARRY       EQU     0       ;CARRY IS BIT 0.
TRISA       EQU     85H     ;PORTA Configuration Register
TRISB       EQU     86H     ;PORTB Configuration Register
OPTION_R    EQU     81H     ;Option Register
OSCCON      EQU     8FH     ;Oscillator control register.
COUNT       EQU     20H     ;COUNT a register to count events.

;*******************************************************

        LIST        P=16F818    ;we are using the 16F818.
        ORG         0           ;the start address in memory is 0
        GOTO        START       ;goto start!
```

```
;*******************************************************
;Configuration Bits

__CONFIG H'3F10'    ;sets INTRC-A6 is port I/O, WDT off, PUT
                    ;on, MCLR tied to VDD A5 is I/O
                    ;BOD off, LVP disabled, EE protect disabled,
                    ;Flash Program Write disabled,
                    ;Background Debugger Mode disabled, CCP
                    ;function on B2,
                    ;Code Protection disabled.

;****************************************************

;SUBROUTINE SECTION.

;0.1 second delay, actually 0.099968s
DELAYP1  CLRF    TMR0             ;START TMR0.
LOOPB    MOVF    TMR0,W           ;READ TMR0 INTO W.
         SUBLW   .3               ;TIME-3
         BTFSS   STATUS,ZEROBIT   ;Check TIME-W = 0
         GOTO     LOOPB           ;Time is not = 3.
         NOP                      ;add extra delay
         NOP
         RETLW   0                ;Time is 3, return.

;0.5 second delay.
DELAYP5     MOVLW     .5
            MOVWF     COUNT

LOOPC       CALL      DELAYP1
            DECFSZ    COUNT
            GOTO      LOOPC
            RETLW     0

;1 second delay.
DELAY1      MOVLW     .10
            MOVWF     COUNT

LOOPA       CALL      DELAYP1
            DECFSZ    COUNT
            GOTO      LOOPA
            RETLW     0
```

```
;*****************************************************

;CONFIGURATION SECTION.

START    BSF      STATUS,5      ;Turns to Bank1.

         MOVLW    B11111111'    ;8 bits of PORTA are I/P
         MOVWF    TRISA

         MOVLW    B'00000100'   ;A0,A1 and A3 are analogue.
         MOVWF    ADCON1

         MOVLW    B'00000000'
         MOVWF    TRISB         ;PORTB is OUTPUT

         MOVLW    B'00000000'
         MOVWF    OSCCON        ;oscillator 31.25kHz

         MOVLW    B'00000111'   ;Prescaler is /256
         MOVWF    OPTION_R      ;TIMER is 1/32 secs.

         BCF      STATUS,5      ;Return to Bank0.

         BSF      ADCON0,0      ;Turn ON A/D

         CLRF     PORTA         ;Clears PortA.
         CLRF     PORTB         ;Clears PortB.

;*****************************************************
;Program starts now.

END
```

Head818A.ASM explained

HEAD818A.ASM is similar in operation to HEAD818.ASM outlined in Chapter 6, with the following extras:

- The Carry Bit in the status register, that indicates if a calculation is +ve or −ve, it is bit 0 and has been equated to 0.

- In the Configuration Section A0, A1 and A3 are set as Analogue inputs, A2, A4, A5, A6 and A7 are set up as digital inputs with:

```
MOVLW    B'00000100'
MOVWF    ADCON1
```

The A/D converter is switched on with:

```
BSF      ADCON0,0
```

A/D Conversion – example, a temperature sensitive switch

To introduce the working of the A/D converter we will consider a simple example. i.e. Turn an LED on when the Temperature is above 25°C and turn the LED off when it is below 25°C.

The diagram for this Temperature Switch Circuit is shown in Figure 11.4.

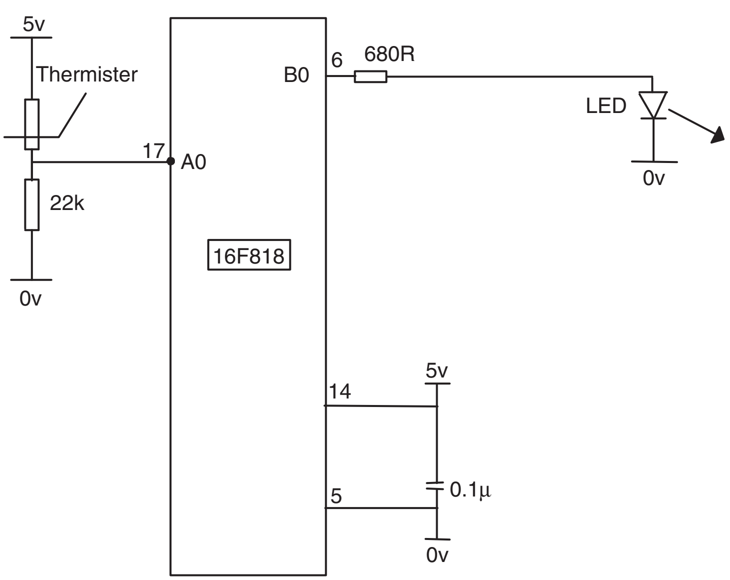

Figure 11.4 Temperature switch circuit

Taking the A/D reading

The A/D converter has been switched on in the header and it automatically looks at Channel 0 unless told otherwise. In order to make the measurement the GO/DONE bit, bit2 is set and we wait until it is cleared with:

```
        BSF        ADCON0,2   ;Take measurement, set GO/DONE
WAIT    BTFSC      ADCON0,2   ;Wait until GO/DONE is clear
        GOTO       WAIT
```

The measurement will then be in the A/D Result register, ADRES.

Determining if the temperature is above or below 25°C

Suppose the voltage on the analogue input, Channel 0, A0 is 2.4v when the temperature is 25°C. The required A/D reading for 2.4v is $2.4 \times 51 = 122$. We therefore need to know when the A/D reading is above and below 122, i.e. above and below 25°C.

Previously we have seen how to tell if a value is equal to another by subtracting and looking at the zerobit in the status register (Chapter 5).

There is another bit, bit 0 in the status register called the Carry Bit, which indicates if the result of a subtraction is +ve or −ve. If the Carry Bit is set the result was +ve, if the bit is clear the result was −ve. So we can tell if the number is above or below a defined value.

The code for this is:

```
MOVF    ADRES,W        ;Move Analogue result into W
SUBLW   .122           ;Do 122 – ADRES, i.e. 122-W
BTFSC   Status,Carry   ;Check the carry bit. Clear if ADRES>122 i.e. −ve
GOTO    TURNOFF        ;Routine to turn off LED
GOTO    TURNON         ;Routine to turn on LED
```

The analogue measurement is moved from ADRES into W where we can subtract it from 122. NB. The subtraction always does, Value – W.

The carry bit tells us if the A/D result is above or below 122.

N.B. If the result of the subtraction is zero the carry is also 1. It must be 1 or 0. Being +ve or zero does not matter in this example.

We have then found out if the result is equal to or above 122, or if it is less than 122.

When the measurement is made we then goto one of two subroutines, TURNON or TURNOFF. These subroutines are not very grand but they could easily be more complicated, even hundreds of lines long.

Program code

The full code for this Temperature Sensitive Switch Program is shown below as TEMPSENS.ASM

```
;TEMPSENS.ASM.      This sets PORTA as analogue/digital INPUTs.
;                   PORTB is an OUTPUT.
;                   Internal oscillator of 31.25kHz chosen
;                   The OPTION register is set to /256 giving timing
;                   pulses 32.768ms.
;                   1second and 0.5 second delays are included in the
;                   subroutine section.

;*****************************************************
;EQUATES SECTION

TMR0        EQU     1       ;means TMR0 is file 1.
STATUS      EQU     3       ;means STATUS is file 3.
PORTA       EQU     5       ;means PORTA is file 5.
PORTB       EQU     6       ;means PORTB is file 6.
ZEROBIT     EQU     2       ;means ZEROBIT is bit 2.
ADCON0      EQU     1FH     ;A/D Configuration reg.0
ADCON1      EQU     9FH     ;A/D Configuration reg.1
ADRES       EQU     1EH     ;A/D Result register.
CARRY       EQU     0       ;CARRY IS BIT 0.
TRISA       EQU     85H     ;PORTA Configuration Register
TRISB       EQU     86H     ;PORTB Configuration Register
OPTION_R    EQU     81H     ;Option Register
OSCCON      EQU     8FH     ;Oscillator control register.
COUNT       EQU     20H     ;COUNT a register to count events

;*****************************************************

    LIST        P=16F818       ;we are using the 16F818.
    ORG         0              ;the start address in memory is 0
    GOTO        START          ;goto start!
```

```
;*****************************************************
; Configuration Bits

__CONFIG H'3F10'   ;sets INTRC-A6 is port I/O, WDT off, PUT
                   ;on, MCLR tied to VDD A5 is I/O
                   ;BOD off, LVP disabled, EE protect disabled,
                   ;Flash Program Write disabled,
                   ;Background Debugger Mode disabled, CCP
                   ;function on B2,
                   ;Code Protection disabled.
;*************************************************
;SUBROUTINE SECTION.

TURNON      BSF     PORTB,0     ;Turn on LED on B0
            GOTO    BEGIN       ;Return to monitor

TURNOFF     BCF     PORTB,0     ;Turn off LED on B0
            GOTO    BEGIN       ;Return to monitor
;*****************************************************
;CONFIGURATION SECTION.

START       BSF         STATUS,5        ;Turns to Bank1.

            MOVLW       B'11111111'     ;8 bits of PORTA are I/P
            MOVWF       TRISA

            MOVLW       B'00000100'     ;A0,A1 and A3 are analogue.
            MOVWF       ADCON1

            MOVLW       B'00000000'
            MOVWF       TRISB           ;PORTB is OUTPUT

            MOVLW       B'00000000'
            MOVWF       OSCCON          ;oscillator 31.25kHz

            MOVLW       B'00000111'     ;Prescaler is /256
            MOVWF       OPTION_R        ;TIMER is 1/32 secs.

            BCF         STATUS,5        ;Return to Bank0.

            BSF         ADCON0,0        ;Turn ON A/D

            CLRF        PORTA           ;Clears PortA.
            CLRF        PORTB           ;Clears PortB.
```

```
;****************************************************
;Program starts now.

BEGIN   BSF     ADCON0,2   ;Take measurement, set GO/DONE

WAIT    BTFSC   ADCON0,2   ;Wait until GO/DONE is clear
        GOTO    WAIT

        MOVF    ADRES,W    ;Move Analogue result into W
        SUBLW   .122       ;Do 122–ADRES, i.e. 122–W
        BTFSC   STATUS,
                CARRY      ; Clear if ADRES>122
        GOTO    TURNOFF    ;Routine to turn off LED
        GOTO    TURNON     ;Routine to turn on LED
END
```

Another example – a voltage indicator

Previously we have looked at a single input level. But with our 8 bit micro we could look at 255 different input levels.

Suppose we wish to use the LEDs connected to PORTB to indicate the voltage on the Analogue Input AN0. So that as the voltage increases then the number of LEDs lit also increases.

In HEAD818.ASM we have configured the micro so that the voltage reference is Vdd i.e. the 5v supply. This was done with the instructions:

```
MOVLW    B'00000100'
MOVWF    ADCON1
```

This means that 5v will give a digital reading of 255 in our 8 bit register ADRES. The resolution of this register is $5v/255 = 19.6mV$.

Suppose our LED ladder was to increment in 0.5v steps as indicated below:

Vin = 0–0.5v	All LEDs off,	$0.5v = 0.5/5 \times 255 = 25.5 = 26$
Vin = 0.5–1.0v	B0 on,	$1.0v = 1/5 \times 255 = 51$
Vin = 1.0–1.5v	B1 on,	$1.5v = 1.5/5 \times 255 = 76.5 = 77$
Vin = 1.5–2.0v	B2 on,	$2.0v = 2/5 \times 255 = 102$

Vin = 2.0–2.5v	B3 on,	2.5v = 2.5/5 × 255 = 127.5 = 128
Vin = 2.5–3.0v	B4 on,	3.0v = 3/5 × 255 = 153
Vin = 3.0–3.5v	B5 on,	3.5v = 3.5/5 × 255 = 178.5 = 179
Vin = 3.5–4.0v	B6 on,	4.0v = 4/5 × 255 = 204
Vin = 4.0–5.0v	B7 on.	

The circuit diagram for this voltage indicator is shown in Figure 11.5 and the Flowchart is shown in Figure 11.6.

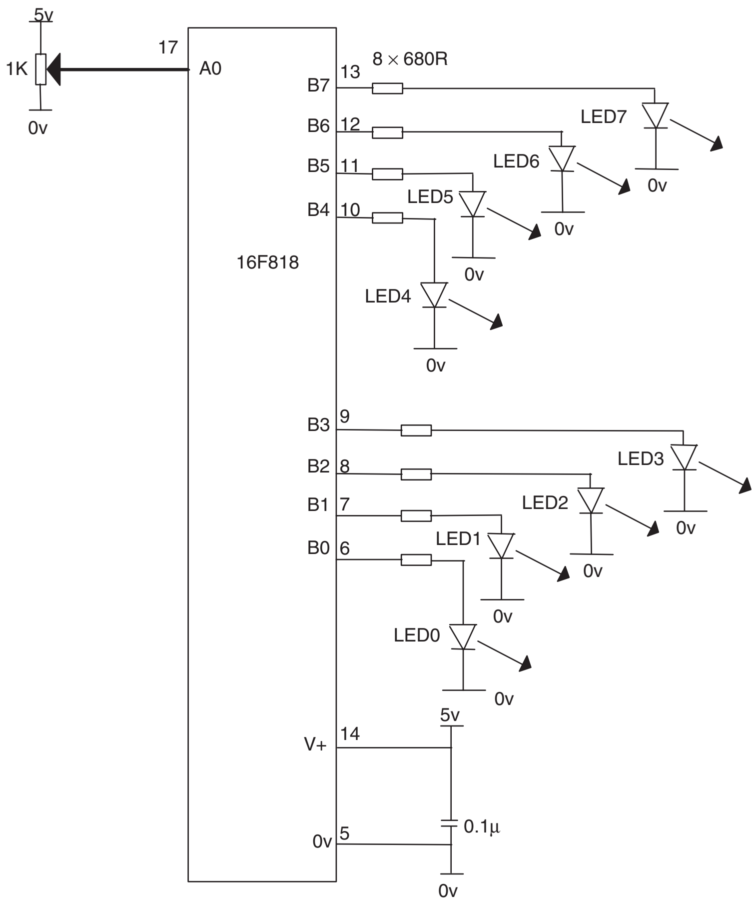

Figure 11.5 Circuit for the voltage indicator

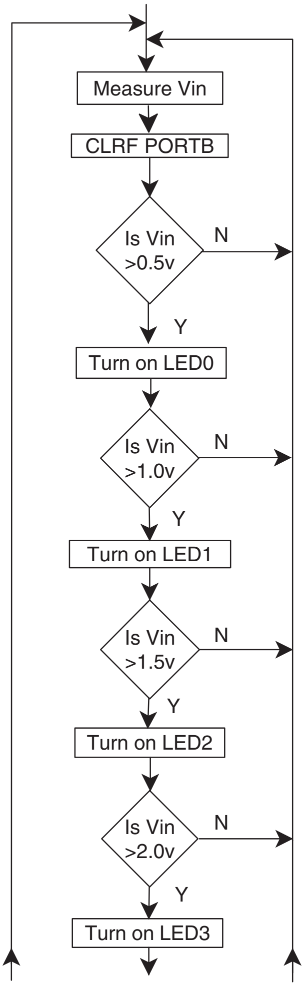

Figure 11.6 Flowchart for the voltage indicator

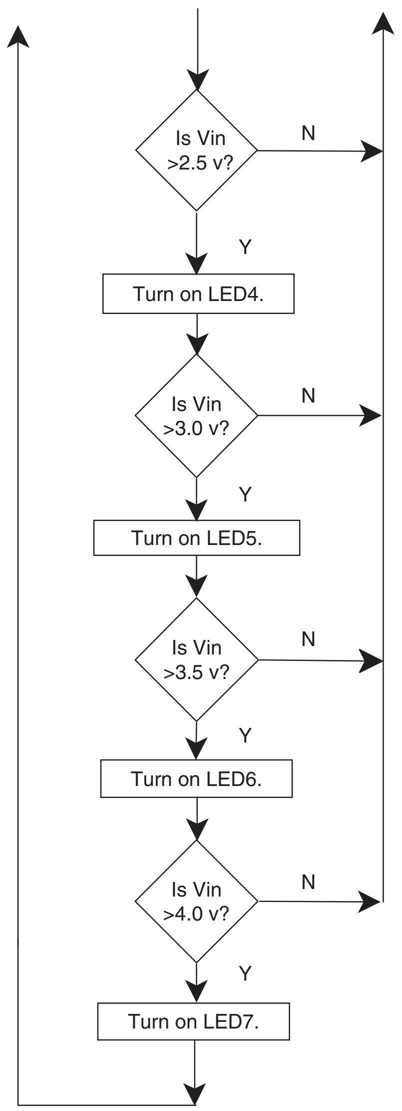

Figure 11.6 Continued

Voltage indicator, program solution

HEAD818A.ASM is altered to produce the program VOLTIND.ASM for the Voltage Indicator Circuit.

```
;VOLTIND.ASM.        This sets PORTA as analogue/digital
;                    INPUTs. PORTB is an OUTPUT.
;                    Internal oscillator of 31.25kHz chosen
;                    The OPTION register is set to /256 giving timing
                     pulses 32.768ms.
;                    1second and 0.5 second delays are included in the
                     subroutine section.

;*******************************************************
; EQUATES SECTION

TMR0        EQU   1     ;means TMR0 is file 1.
STATUS      EQU   3     ;means STATUS is file 3.
PORTA       EQU   5     ;means PORTA is file 5.
PORTB       EQU   6     ;means PORTB is file 6.
ZEROBIT     EQU   2     ;means ZEROBIT is bit 2.
ADCON0      EQU   1FH   ;A/D Configuration reg.0
ADCON1      EQU   9FH   ;A/D Configuration reg.1
ADRES       EQU   1EH   ;A/D Result register.
CARRY       EQU   0     ;CARRY IS BIT 0.
TRISA       EQU   85H   ;PORTA Configuration Register
TRISB       EQU   86H   ;PORTB Configuration Register
OPTION_R    EQU   81H   ;Option Register
OSCCON      EQU   8FH   ;Oscillator control register.
COUNT       EQU   20H   ;COUNT a register to count events
;*******************************************************

    LIST        P=16F818    ;we are using the 16F818.
    ORG         0           ;the start address in memory is 0
    GOTO        START       ;goto start!

;*******************************************************
; Configuration Bits

__CONFIG H'3F10'    ;sets INTRC-A6 is port I/O, WDT off, PUT on,
                    ;MCLR tied to VDD A5 is I/O
                    ;BOD off, LVP disabled, EE protect disabled,
                    ;Flash Program Write disabled,
```

```
                              ;Background Debugger Mode disabled, CCP
                              ;function on B2,
                              ;Code Protection disabled.

;*****************************************************
;CONFIGURATION SECTION.

START    BSF        STATUS,5      ;Turns to Bank1.
         MOVLW      B'00011111'   ;5bits of PORTA are I/P
         MOVWF      TRISA
         MOVLW      B'00000010'   ;A0, A1 are analogue
         MOVWF      ADCON1        ;A2, A3 are digital I/P.
         MOVLW      B'00000000'
         MOVWF      TRISB         ;PORTB is OUTPUT
         BCF        STATUS,5      ;Return to Bank0.
         MOVLW      B'00000001'   ;Turns on A/D converter,
         MOVWF      ADCON0        ;and selects channel AN0
         CLRF       PORTA         ;Clears PortA.
         CLRF       PORTB         ;Clears PortB.

;*****************************************************
;Program starts now.

BEGIN    BSF        ADCON0,2        ;Take Measurement.
WAIT     BTFSC      ADCON0,2        ;Wait until reading done.
         GOTO       WAIT
         MOVF       ADRES,W         ;Move A/D Result into W
         CLRF       PORTB           ;Clear PortB.

         SUBLW      .26             ;26-,W. W is altered
         BTFSC      STATUS,CARRY    ;Is W> or <26
         GOTO       BEGIN           ;W is <26 (0.5v)

         MOVF       ADRES,W         ;Move A/D Result into W
         BSF        PORTB,0         ;Turn on B0.
         SUBLW      .51             ;51-,W. W is altered
         BTFSC      STATUS,CARRY    ;Is W> or <51
         GOTO       BEGIN           ;W is <51 (1.0v)

         MOVF       ADRES,W         ;Move A/D Result into W
         BSF        PORTB,1         ;Turn on B1.
         SUBLW      .77             ;77-,W. W is altered
         BTFSC      STATUS,CARRY    ;Is W> or <77
         GOTO       BEGIN           ;W is <77 (1.5v)
```

```
        MOVF        ADRES,W         ;Move A/D Result into W
        BSF         PORTB,2         ;Turn on B2.
        SUBLW       .102            ;102-,W. W is altered
        BTFSC       STATUS,CARRY    ;Is W> or <102
        GOTO        BEGIN           ;W is <102 (2.0v)

        MOVF        ADRES,W         ;Move A/D Result into W
        BSF         PORTB,3         ;Turn on B3.
        SUBLW       .128            ;128-,W. W is altered
        BTFSC       STATUS,CARRY    ;Is W> or <128
        GOTO        BEGIN           ;W is <128 (2.5v)

        MOVF        ADRES,W         ;Move A/D Result into W
        BSF         PORTB,4         ;Turn on B4.
        SUBLW       .153            ;153-,W. W is altered
        BTFSC       STATUS,CARRY    ;Is W> or <153
        GOTO        BEGIN           ;W is <153 (3.0v)

        MOVF        ADRES,W         ;Move A/D Result into W
        BSF         PORTB,5         ;Turn on B5.
        SUBLW       .179            ;179-,W. W is altered
        BTFSC       STATUS,CARRY    ;Is W> or <179
        GOTO        BEGIN           ;W is <179 (3.5v)

        MOVF        ADRES,W         ;Move A/D Result into W
        BSF         PORTB,6         ;Turn on B6.
        SUBLW       .204            ;204-,W. W is altered
        BTFSC       STATUS,CARRY    ;Is W> or <204
        GOTO        BEGIN           ;W is <204 (4.0v)

        BSF         PORTB,7         ;Turn on B7.
        GOTO        BEGIN
END
```

Operation of the voltage indicator program

The code to make the analogue measurement is the same as in the Temperature Switch Circuit. Once the measurement has been taken the program checks to see if the digital value of the input is >26 if it is B0 LED is switched on. The program then checks to see if the measurement is >51, if so then B1 LED is lit. If the reading is >77 then B2 LED is lit etc. When the value is less than the one being checked then the program branches back to the beginning, makes another measurement and the cycle repeats.

NB. After the A/D reading the LEDs are cleared before being turned on, in case the voltage has dropped.

To check if a reading (or any number) is > say 26.
Put the number into W.
Take W from 26 i.e. 26-W by SUBLW .26

If the result is +ve, the number is <26 and the carry bit is set in the Status Register. If the number is >26 the result is −ve and the carry bit is clear.

Problem

To check your understanding of the previous section, try this.

Turn a red LED on only when the input voltage is above 3v and turn a yellow LED on only when the input voltage is below 1v and turn a green LED on only when the voltage is between 1v and 3v.

Hint

Check for voltage >3v if true GOTO RED
If not check for voltage <1v if true GOTO YELLOW
If false then GOTO GREEN.