16
The 16F87X microcontroller

The 16F87X range includes the devices, 16F870, 16F871, 16F872, 16F873, 16F874, 16F876 and 16F877. They are basically the same device but differ in the amounts of I/O, analogue inputs, program memory, data memory (RAM) and EEPROM data memory that they have.

The 16F87X have more I/O, program memory, data memory, EEPROM data memory and analogue inputs than the 16F818.

16F87X family specification

Device	Program Memory	EEPROM Data Memory (bytes)	RAMBytes	Pins	I/O	10 bit A/D Channels
16F870	2k	64	128	28	22	5
16F871	2k	64	128	40	33	8
16F872	2k	64	128	28	22	5
16F873	4k	128	192	28	22	5
16F874	4k	128	192	40	33	8
16F876	8k	256	368	28	22	5
16F877	8k	256	368	40	33	8

16F87X memory map

The 16F87X devices have more functions than we have seen previously. These functions of course need registers in order to make the various selections.

The memory map of the 16F87X showing these registers is shown in Figure 16.3.

The 16F87X devices have a number of extra registers that are not required in the applications we have looked at. For an explanation of these registers please see Microchip's website @ www.microchip.com, where you can download the data sheet as a pdf (portable document file), which can be read using Adobe Acrobat Reader.

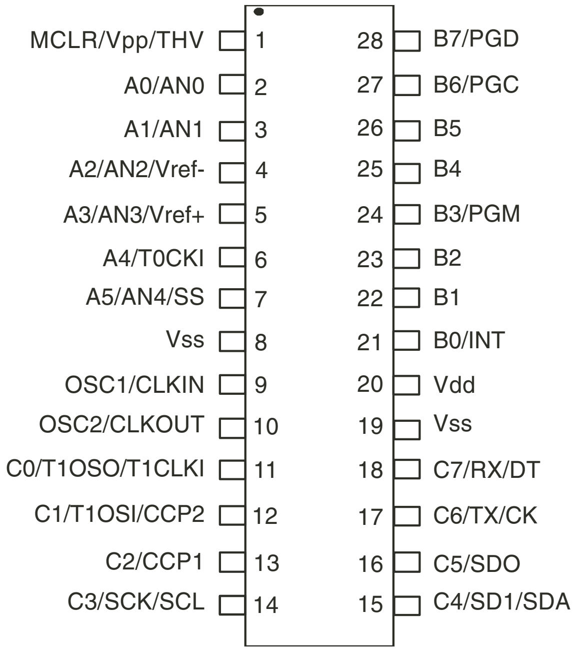


Figure 16.1 The 16F870/2/3/6 pinout

The 16F872 microcontroller

In order to demonstrate the operation of the 16F87X series we will consider the 16F872 device. This is a 28pin device with 22 I/O available on 3 ports. PortA has 6 I/O, PortB has 8 I/O and PORTC has 8 I/O. Of the 6 I/O available on PortA 5 of them can be analogue inputs. The header for the 16F872, HEAD872.ASM, configures the device with 5 analogue inputs on PortA, 8 digital inputs on PortC and 8 outputs on PortB. The port configuration for the device is shown in Figure 16.4.

The 16F872 has been configured in HEAD872.ASM, using a 32 kHz crystal, to allow all the programs used previously to be copied over with as little alteration as possible.

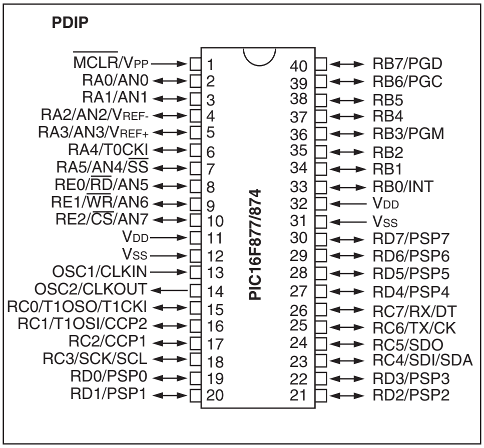
Devices included in this Data Sheet:

- PIC16F873
- PIC16F874
- PIC16F876
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- Up to 8K × 14 words of FLASH Program Memory,
  Up to 368 × 8 bytes of Data Memory (RAM)
  Up to 256 × 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and
  Oscillatior Start-up Times (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial and Industrial and Extended temperature ranges
- Low-power consumption:
  - < 2 mA typical @ 3V, 4 MHz
  - 20 µA typical @3V, 32 kHz
  - < 1 µA typical standby current

Pin Diagram



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, rescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I^2C™ (Master/Slave)
- Universal Synchronous Asychronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external $\overline{RD}$, $\overline{WR}$ and $\overline{CS}$ controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

Figure 16.2 The 16F87X data sheet

<table>
<tr><th>Address</th><th>File Name
Bank0</th><th>File Name
Bank1</th><th>File Name
Bank2</th><th>File Name
Bank3</th></tr>
<tr><td>00h</td><td>Ind.Add</td><td>Ind.Add</td><td>Ind.Add</td><td>Ind.Add</td></tr>
<tr><td>01h</td><td>TMR0</td><td>Option</td><td>TMR0</td><td>Option</td></tr>
<tr><td>02h</td><td>PCL</td><td>PCL</td><td>PCL</td><td>PCL</td></tr>
<tr><td>03h</td><td>Status</td><td>Status</td><td>Status</td><td>Status</td></tr>
<tr><td>04h</td><td>FSR</td><td>FSR</td><td>FSR</td><td>FSR</td></tr>
<tr><td>05h</td><td>PORTA</td><td>TRISA</td><td></td><td></td></tr>
<tr><td>06h</td><td>PORTB</td><td>TRISB</td><td>PORTB</td><td>TRISB</td></tr>
<tr><td>07h</td><td>PORTC</td><td>TRISC</td><td></td><td></td></tr>
<tr><td>08h</td><td>PORTD</td><td>TRISD</td><td></td><td></td></tr>
<tr><td>09h</td><td>PORTE</td><td>TRISE</td><td></td><td></td></tr>
<tr><td>0Ah</td><td>PCLATH</td><td>PCLATH</td><td>PCLATH</td><td>PCLATH</td></tr>
<tr><td>0Bh</td><td>INTCON</td><td>INTCON</td><td>INTCON</td><td>INTCON</td></tr>
<tr><td>0Ch</td><td>PIR1</td><td>PIE1</td><td>EEDATA</td><td>EECON1</td></tr>
<tr><td>0Dh</td><td>PIR2</td><td>PIE2</td><td>EEADR</td><td>EECON2</td></tr>
<tr><td>0Eh</td><td>TMR1L</td><td>PCON</td><td>EEDATH</td><td></td></tr>
<tr><td>0Fh</td><td>TMR1H</td><td></td><td>EEADRH</td><td></td></tr>
<tr><td>10h</td><td>T1CON</td><td></td><td rowspan="18">General
Purpose
Register
96 bytes</td><td rowspan="18">General
Purpose
Register
96 bytes</td></tr>
<tr><td>11h</td><td>TMR2</td><td>SSPCON2</td></tr>
<tr><td>12h</td><td>T2CON</td><td>PR2</td></tr>
<tr><td>13h</td><td>SSPBUF</td><td>SSPADD</td></tr>
<tr><td>14h</td><td>SSPCON</td><td>SSPSTAT</td></tr>
<tr><td>15h</td><td>CCPR1L</td><td></td></tr>
<tr><td>16h</td><td>CCPR1H</td><td></td></tr>
<tr><td>17h</td><td>CCP1CON</td><td></td></tr>
<tr><td>18h</td><td>RCSTA</td><td>TXSTA</td></tr>
<tr><td>19h</td><td>TXREG</td><td>SPBRG</td></tr>
<tr><td>1Ah</td><td>RCREG</td><td></td></tr>
<tr><td>1Bh</td><td>CCPR2L</td><td></td></tr>
<tr><td>1Ch</td><td>CCPR2H</td><td></td></tr>
<tr><td>1Dh</td><td>CCP2CON</td><td></td></tr>
<tr><td>1Eh</td><td>ADRESH</td><td>ADRESL</td></tr>
<tr><td>1Fh</td><td>ADCON0</td><td>ADCON1</td></tr>
<tr><td>.</td><td rowspan="3">General
Purpose
Register
96 bytes</td><td rowspan="2">General
Purpose
Register
80 bytes</td></tr>
<tr><td>6Fh</td></tr>
<tr><td>7FH</td><td></td><td></td><td></td></tr>
</table>

Figure 16.3 The 16F87X memory map

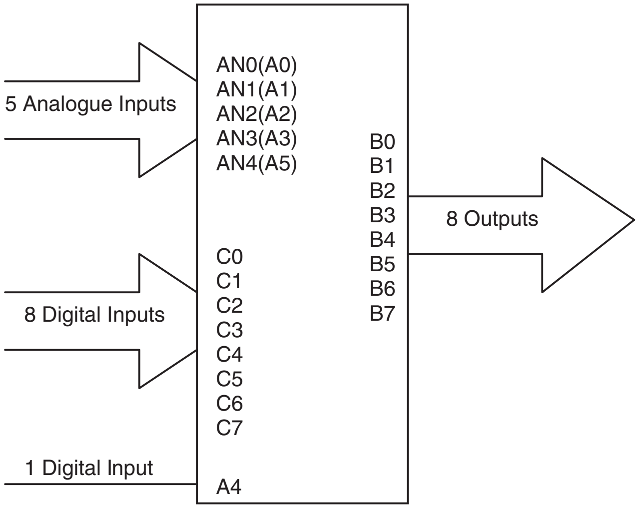


Figure 16.4 Port configuration of the 16F872

The 16F872 header

HEAD872.ASM
;EQUATES SECTION

```
TMR0        EQU    1
OPTION_R    EQU    1
PORTA       EQU    5
PORTB       EQU    6
PORTC       EQU    7
TRISA       EQU    5
TRISB       EQU    6
TRISC       EQU    7
STATUS      EQU    3
ZEROBIT     EQU    2
CARRY       EQU    0
EEADR       EQU    0DH
EEDATA      EQU    0CH
EECON1      EQU    0CH
EECON2      EQU    0DH
RD          EQU    0
WR          EQU    1
WREN        EQU    2
ADCON0      EQU    1FH
ADCON1      EQU    1FH
ADRES       EQU    1EH
```

```
        CHS0      EQU     3
        GODONE    EQU     2
        COUNT     EQU     20H

;*****************************************************
        LIST      P=16F872
        ORG       0
        GOTO      START

;*******************************************************
; SUBROUTINE SECTION.

;1 SECOND DELAY
DELAY1   CLRF     TMR0               ;Start TMR0
LOOPA    MOVF     TMR0,W             ;Read TMR0 into W
         SUBLW    .32                ;TIME-W
         BTFSS    STATUS,ZEROBIT     ;Check TIME-W=0
         GOTO     LOOPA
         RETLW    0                  ;Return after TMR0=32

;0.5 SECOND DELAY
DELAYP5  CLRF     TMR0               ;Start TMR0
LOOPB    MOVF     TMR0,W             ;Read TMR0 into W
         SUBLW    .16                ;TIME-W
         BTFSS    STATUS,ZEROBIT     ;Check TIME-W=0
         GOTO     LOOPB
         RETLW    0                  ;Return after TMR0=16

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

START     BSF       STATUS,5    ;Bank1
          MOVLW     B'11111111'
          MOVWF     TRISA       ;PortA is input

          MOVLW     B'00000000'
          MOVWF     TRISB       ;PortB is output

          MOVLW     B'11111111'
          MOVWF     TRISC       ;PortC is input

          MOVLW     B'00000111'
          MOVWF     OPTION_R    ;Option Register, TMR0 / 256
```

```
        MOVLW   B'00000000'
        MOVWF   ADCON1      ;PortA bits 0, 1, 2, 3, 5 are analogue
        BSF     STATUS,6    ;BANK3
        BCF     EECON1,7    ;Data memory on.
        BCF     STATUS,5
        BCF     STATUS,6    ;BANK0 return
        BSF     ADCON0,0    ;turn on A/D.
        CLRF    PORTA
        CLRF    PORTB
        CLRF    PORTC

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

Explanation of HEAD872.ASM

Equates Section

- We have a third port, PORTC file 7 and its corresponding TRIS file, TRISC file 7 on Bank1. The TRIS file sets the I/O direction of the port bits.
- The EEPROM data file addresses have been included. EEADR is file 0Dh in Bank2, EEDATA is file 0Ch in Bank2, EECON is file 0Ch in Bank3 and EECON2 is file 0Dh in Bank3.
- The EEPROM data bits have been added. RD the read bit is bit 0, WR the write bit is bit 1, WREN the write enable bit is bit 2.
- The Analogue files ADRES, ADCON1 and ADCON2 have been included as have the associated bits CHS0 channel 0 select bit 3 and the GODONE bit, bit 2.

List Section

- This of course indicates the microcontroller being used, the 16F872 and that the first memory location is 0. In address 0 is the instruction GOTO START that instructs the micro to bypass the subroutine section and goto the configuration section at the label START.

Subroutine Section

- This includes the 2 delays DELAY1 and DELAYP5 as before.

Configuration Section

- As before we need to switch to Bank1 to address the TRIS files to configure the I/O. PORTA is set as an input port with the two instructions

```
MOVLW    B'00000111'
MOVWF    TRISA
```

PORTB and PORTC are configured in a similar manner using TRISB and TRISC.

- The Option register is configured with the instructions

```
MOVLW    B'00000111'
MOVWF    OPTION_R
```

- The A/D register is configured with the instructions

```
MOVLW    B'00000000'
MOVWF    ADCON1
```

Setting PORTA bits 0, 1, 2, 3 and 5 as analogue inputs.

- We turn to Bank3 by setting Bank select bit, STATUS,6 (bit 5 is still set) so that we can address EECON1, the EEPROM data control register. BSF EECON1 then enables access to the EEPROM program memory when required.
- We then turn back to Bank0 by clearing bits 5 and 6 of the Status register and clear the files PortA, PortB and PortC.

16F872 Application – a greenhouse control

In order to demonstrate the operation of the 16F872 and to develop our programming skills a little further consider the following application.

- A greenhouse has its temperature monitored so that a heater is turned on when the temperature drops below 15°C and turns the heater off when the temperature is above 17°C.
- A probe in the soil monitors the soil moisture so that a water valve will open for 5 seconds to irrigate the soil if it dries out. The valve is closed and will not be active for a further 5 seconds to give the water time to drain into the soil.
- A float switch monitors the level of the water and sounds an alarm if the water drops below a minimum level.

The circuit diagram for the greenhouse control is shown in Figure 16.5 and the flowchart is drawn in Figure 16.6.

Greenhouse program

In order to program the analogue/digital settings consider the NTC Thermister. As the temperature increases the resistance of the thermister will decrease and so the voltage presented to AN0 will increase.

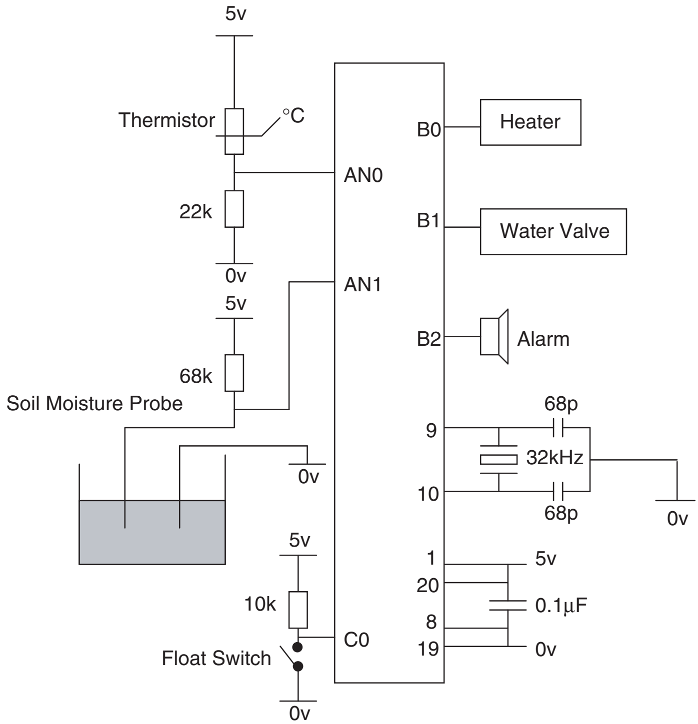


Figure 16.5 Greenhouse control circuit

Let us assume the voltage is 2.9v at 15°C and 3.2v at 17°C they correspond to digital readings of $2.9 \times 51 = 147.9$ i.e. 148 and $3.2 \times 51 = 163.2$ i.e. 163. (N.B. 5v = 255, so 1v = 51 we are using an 8 bit A/D.)

Our program then needs to check when AN0 goes above 163 and below 148.

As the soil dries out its resistance will increase. Let us assume in our application dry soil will give a reading of 2.6v, (on AN1), i.e. $2.6 \times 51 = 132.6$ i.e. 133. So any reading above 133 is considered dry.

The float switch is a digital input showing 1 if the water level is above the minimum required and a 0 if it is below the minimum.

Greenhouse code

The code for the greenhouse uses HEAD872.ASM with the program instuctions added and saved as GREENHO.ASM.

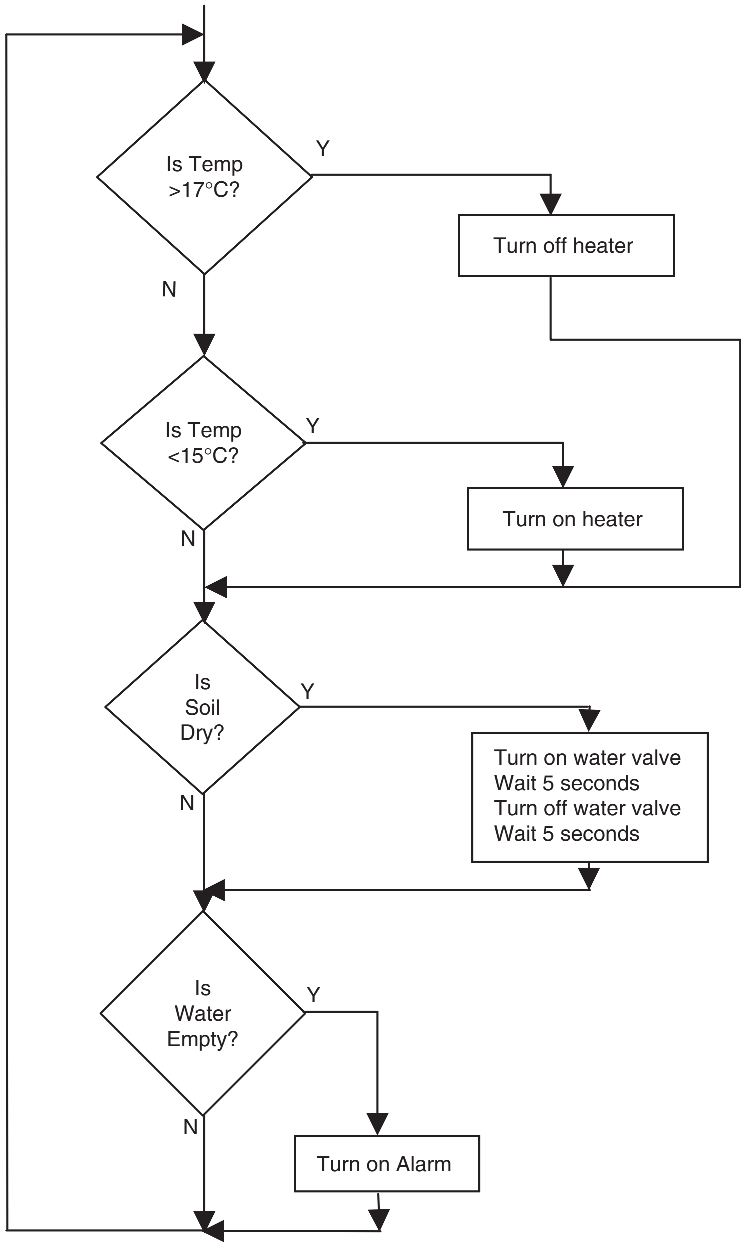


Figure 16.6 Greenhouse control flowchart

```
;GREENHO.ASM
;EQUATES SECTION
            TMR0        EQU     1
            OPTION_R    EQU     1
            PORTA       EQU     5
            PORTB       EQU     6
            PORTC       EQU     7
            TRISA       EQU     5
            TRISB       EQU     6
            TRISC       EQU     7
            STATUS      EQU     3
            ZEROBIT     EQU     2
            CARRY       EQU     0
            EEADR       EQU     0DH
            EEDATA      EQU     0CH
            EECON1      EQU     0CH
            EECON2      EQU     0DH
            RD          EQU     0
            WR          EQU     1
            WREN        EQU     2
            ADCON0      EQU     1FH
            ADCON1      EQU     1FH
            ADRES       EQU     1EH
            CHS0        EQU     3
            GODONE      EQU     2
            COUNT       EQU     20H

;*************************************************
            LIST        P=16F872
            ORG         0
            GOTO        START
;****************************************************
;Configuration Bits

__CONFIG H'3F30'        ;selects LP oscillator, WDT off, PUT on,
                        ;Code Protection disabled.
;****************************************************
;SUBROUTINE SECTION.

;1 SECOND DELAY
DELAY1   CLRF     TMR0              ;Start TMR0
LOOPA    MOVF     TMR0,W            ;Read TMR0 into W
         SUBLW    .32               ;TIME-W
         BTFSS    STATUS,ZEROBIT    ;Check TIME-W = 0
```

```
            GOTO    LOOPA
            RETLW   0                     ;Return after TMR0 = 32 >

;0.5 SECOND DELAY
DELAYP5     CLRF    TMR0                  ;Start TMR0
LOOPB       MOVF    TMR0,W                ;Read TMR0 into W
            SUBLW   .16                   ;TIME-W
            BTFSS   STATUS,ZEROBIT        ;Check TIME-W = 0
            GOTO    LOOPB
            RETLW   0                     ;Return after TMR0 = 16

;5 SECOND DELAY
DELAY5      CLRF    TMR0                  ;Start TMR0
LOOPC       MOVF    TMR0,W                ;Read TMR0 into W
            SUBLW   .160                  ;TIME-W
            BTFSS   STATUS,ZEROBIT        ;Check TIME-W = 0
            GOTO    LOOPC
            RETLW   0                     ;Return after TMR0 = 160

HEAT_ON     BSF     PORTB,0     ;Turn heater on
            GOTO    SOIL        ;Check soil moisture

HEAT_OFF    BCF     PORTB,0     ;Turn heater off
            GOTO    SOIL        ;Check soil moisture

WATER_ON BSF        PORTB,1     ;Turn water on
            CALL    DELAY5
            BCF     PORTB,1     ;Turn water off
            CALL    DELAY5
            GOTO    WATER       ;Check water level

ALARM_ON    BSF     PORTB,2     ;Turn alarm on
            GOTO    BEGIN       ;Repeat the process

ALARM_OFF BCF       PORTB,2     ;Turn alarm off
            GOTO    BEGIN       ;Repeat the process

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

START       BSF     STATUS,5    ;Bank1
            MOVLW   B'11111111'
            MOVWF   TRISA       ;PortA is input

            MOVLW   B'00000000'
            MOVWF   TRISB       ;PortB is output
```

```
            MOVLW    B'11111111'
            MOVWF    TRISC        ;PortC is input

            MOVLW    B'00000111'
            MOVWF    OPTION_R     ;Option Register, TMR0/256

            MOVLW    B'00000000'
            MOVWF    ADCON1       ;PortA bits 0, 1, 2, 3, 5 are
                                  ;analogue
            BSF      STATUS,6     ;BANK3
            BCF      EECON1,7     ;Data memory on.
            BCF      STATUS,5
            BCF      STATUS,6     ;BANK0 return
            BSF      ADCON0,0     ;turn on A/D.
            CLRF     PORTA
            CLRF     PORTB
            CLRF     PORTC

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

;Check the temperature on AN0
BEGIN    BCF      ADCON0,CHS0       ;C to select AN0
         BSF      ADCON0,GODONE
WAIT1    BTFSC    ADCON0,GODONE
         GOTO     WAIT1
         MOVF     ADRES,W
         SUBLW    .163              ;163 – W

         BTFSS    STATUS,CARRY      ;C if W > 163 i.e. hot
                                    ;(above 17°C)
         GOTO     HEAT_OFF

         MOVF     ADRES,W
         SUBLW    .148              ;148 – W

         BTFSC    STATUS,CARRY      ;S if W < 148 i.e. cold
                                    ;(below 15°C)
         GOTO     HEAT_ON

;Check the soil moisture on AN1
SOIL     BSF      ADCON0,CHS0       ;S to select AN1
         BSF      ADCON0,GODONE
WAIT2    BTFSC    ADCON0,GODONE
         GOTO     WAIT2
```

```
            MOVF     ADRES,W
            SUBLW    .133                ;133 – W
            BTFSS    STATUS,CARRY        ;C if W > 133 i.e. dry
            GOTO     WATER_ON

;Check water is above minimum
WATER       BTFSC    PORTC,0             ;C if below minimum
            GOTO     ALARM_OFF
            GOTO     ALARM_ON
END
```

Explanation of code

In the previous analogue circuits in Chapter 11 we only used 1 analogue input on AN0. We now have two analogue inputs on AN0 and AN1. When making an analogue measurement we must specify which analogue channel we wish to measure. The default is AN0 when moving to AN1 we select AN1 by setting channel select bit0 i.e. BSF ADCON0,CHS0.

When moving back to AN0 clear the channel select bit. The 8 channels, AN0 to AN7 are seclected using bits, CHS2, CHS1, CHS0.

- The temperature is read on AN0 with and then checked to see if it is greater than 17°C, by subtracting the A/D reading from 163 (the reading equating to 17°C). The carry bit in the status register indicates if the result is +ve or −ve being set or clear. We then go to turn off the heater if the temperature is above 17°C or check if the temperature is below 15°C. In which case we turn on the heater.
- The soil moisture is checked next. AN1 is selected and the reading compared this time to 133 indicating dry soil. The program either goes to turn on the water valve if the soil is dry or continues to check the water level if the soil is wet.
- If the water level is below minimum then the alarm sounds, if above minimum the alarm is turned off. The program then repeats the checking of the inputs and reacts to them accordingly.

Programming the 16F872 microcontroller using PICSTART PLUS

Once the pogram GREENHO.ASM has been saved it is then assembled using MPASMWIN. The next step as previously is to program GREENHO.HEX into the micro using PICSTART PLUS.

This process has been outlined in Chapter 2, but there are a few more selections to attend to in the ‘Device Specification’ Section.

- Select the device 16F872, if this device is not available you will require a later version of MPLAB, obtainable from www.microchip.com.
- Set the fuses.

Configuration bits

The configuration bit settings when programming the 16F872 for the Greenhouse program are shown in Figure 16.7.

Configuration Bits

Address	Value	Category	Setting
2007	3F30	Oscillator	LP
		Watchdog Timer	Off
		Power Up Timer	On
		Brown Out Detect	Off
		Low Voltage Program	Disabled
		Flash Program Write	Enabled
		Background Debug	Disabled
		Data EE Read Protect	Off
		Code Protect	Off

Figure 16.7 Greenhouse program configuration bits

Reconfiguring the 16F872 header

- The port settings are changed as they were for the 16F84 i.e. a 1 means the pin is an input and a 0 means an output.
- The Option Register is configured as in the 16F84 see also Chapter 19.
- The A/D convertor configuration is adjusted using A/D configuration register 1, i.e. ADCON1 shown in Figure 16.8.

ADFM				PCFG3	PCFG2	PCFG1	PCFG0
bit7							bit0

Figure 16.8 ADCON1, A/D port configuration register 1

Bit7 is the A/D Format Select bit, which selects which bits of the A/D result registers are used. I.e. the A/D can use 10 bits which requires two result registers, ADRESH and ADRESL. Two formats are available.
(a) the most significant bits of ADRESH read as 0, with ADFM = 1

ADRERSH

0	0	0	0	0	0		

ADRESL

Or (b) the least significant bits of ADRESL read as 0, with ADFM = 0

ADRESH							

ADRESL							
		0	0	0	0	0	0

For 8 bit operation condition (b) is used with ADRESH as the 8 most significant bits of the A/D result. This is the default configuration used in HEADER872.ASM where ADRESH (ADRES in the equates) is register 1Eh in Bank0.

Table 16.1 A/D Port configuration

PCFG3: PCFG0	AN7 E2	AN6 E1	AN5 E0	AN4 A5	AN3 A3	AN2 A2	AN1 A1	AN0 A0	Vref+	Vref–
0000	A	A	A	A	A	A	A	A	Vdd	Vss
0001	A	A	A	A	Vref+	A	A	A	A3	Vss
0010	D	D	D	A	A	A	A	A	Vdd	Vss
0011	D	D	D	A	Vref+	A	A	A	A3	Vss
0100	D	D	D	D	A	D	A	A	Vdd	Vss
0101	D	D	D	D	Vref+	D	A	A	A3	Vss
011X	D	D	D	D	D	D	D	D	Vdd	Vss
1000	A	A	A	A	Vref+	Vref–	A	A	A3	A2
1001	D	D	A	A	A	A	A	A	Vdd	Vss
1010	D	D	A	A	Vref+	A	A	A	A3	Vss
1011	D	D	A	A	Vref+	Vref–	A	A	A3	A2
1100	D	D	D	A	Vref+	Vref–	A	A	A3	A2
1101	D	D	D	D	Vref+	Vref–	A	A	A3	A2
1110	D	D	D	D	D	D	D	A	Vdd	Vss
1111	D	D	D	D	Vref+	Vref–	D	A	A3	A2

Table 16.1 shows the A/D Port Configuration settings for PCFG3, PCFG2, PCFG1 and PCFG0.

A = Analogue Input, D = Digital input.
Vdd = +ve supply, Vss = −ve supply.
Vref+ = high voltage reference.
Vref− = low voltage reference.
A3 = PortA,3 A2 = PortA,2 etc.

N.B. AN7, AN6 and AN5 are only available on the 40 pin devices 16F871, 16F874 and 16F877.