13
The 16F87X microcontroller

PIC manufacturers, Arizona Microchip, are continually expanding their range of microcontrollers. They have recently introduced a range of flash devices (no eraser required), with the family name 16F87X. This 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 range is similar to the 16F84 and 16C711 combined – but bigger and better. They have more I/O, program memory, data memory, and EEPROM data memory than the 16F84, and have more analogue inputs than the 16C711 and use 10-bit A/D conversion instead of 8 bits. So the A/D resolution is 1024 instead of 256.

16F87X family specification

Device	Program Memory	EEPROM Data Memory (bytes)	RAM (bytes)	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 13.1.

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

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

Figure 13.1 16F87X memory map

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.

The 16F872 microcontroller

In order to demonstrate the operation of the 16F87X series we will consider the 16F872 device. This is a 28 pin device with 22 I/O available on three ports. PortA has six I/O, PortB has eight I/O and PORTC has eight I/O. Of the six I/O available on PortA five of them can be analogue inputs. The header for the 16F872, HEAD872.ASM, configures the device with five analogue inputs on PortA, eight digital inputs on PortC and eight outputs on PortB. The Port Configuration for the device is shown in Figure 13.2.

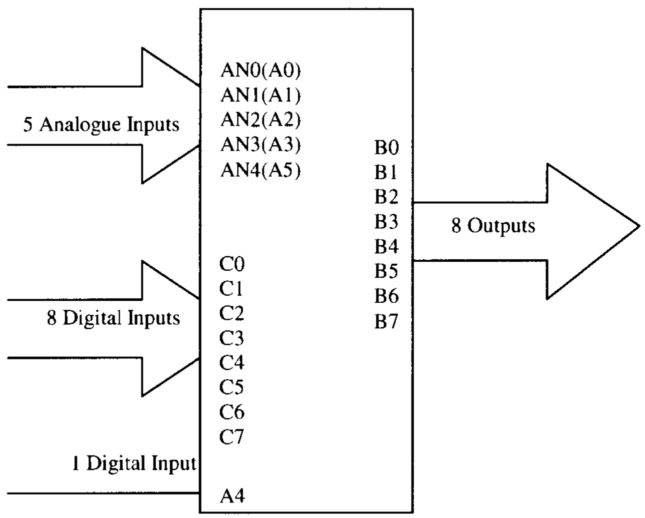

Figure 13.2 Port Configuration of the 16F872

The 16F872 has been configured in HEADER872.ASM to allow all the programs used previously to be copied over with as little alteration as possible.

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

- The first difference here is that the OPTION register now has a file address, it is file 1 in Bank1.
- 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, EECON1 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 ADCON0 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 two 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'11111111'
        MOVWF   TRISA
```

- The recommended command now uses the file TRISA rather than the previous TRIS PORTA. 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
```

These instructions now use the file name OPTION_R to move the data into the Option Register instead of the previous command OPTION.

- 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 13.3 and the flowchart is drawn in Figure 13.4.

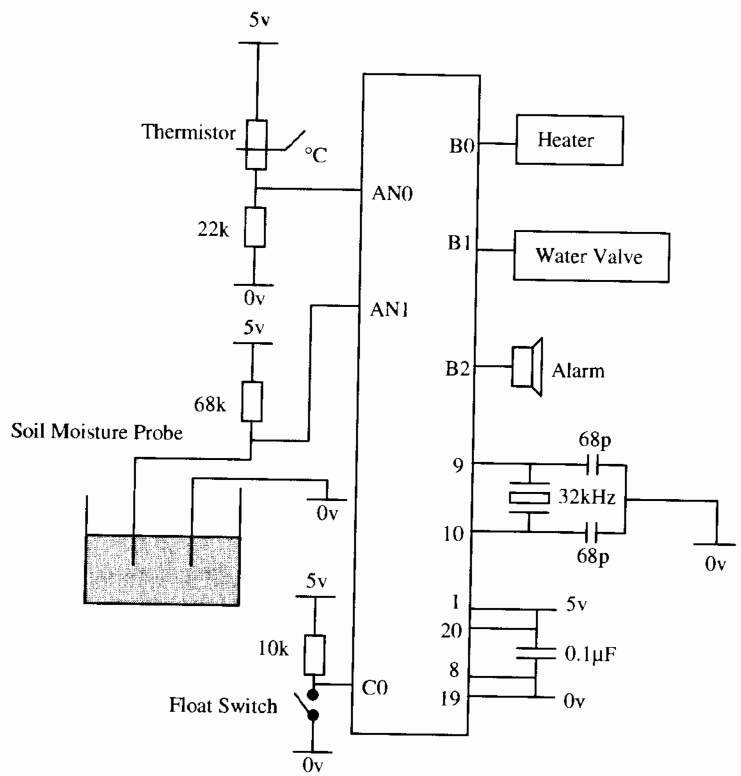

Figure13.3 Greenhouse control circuit

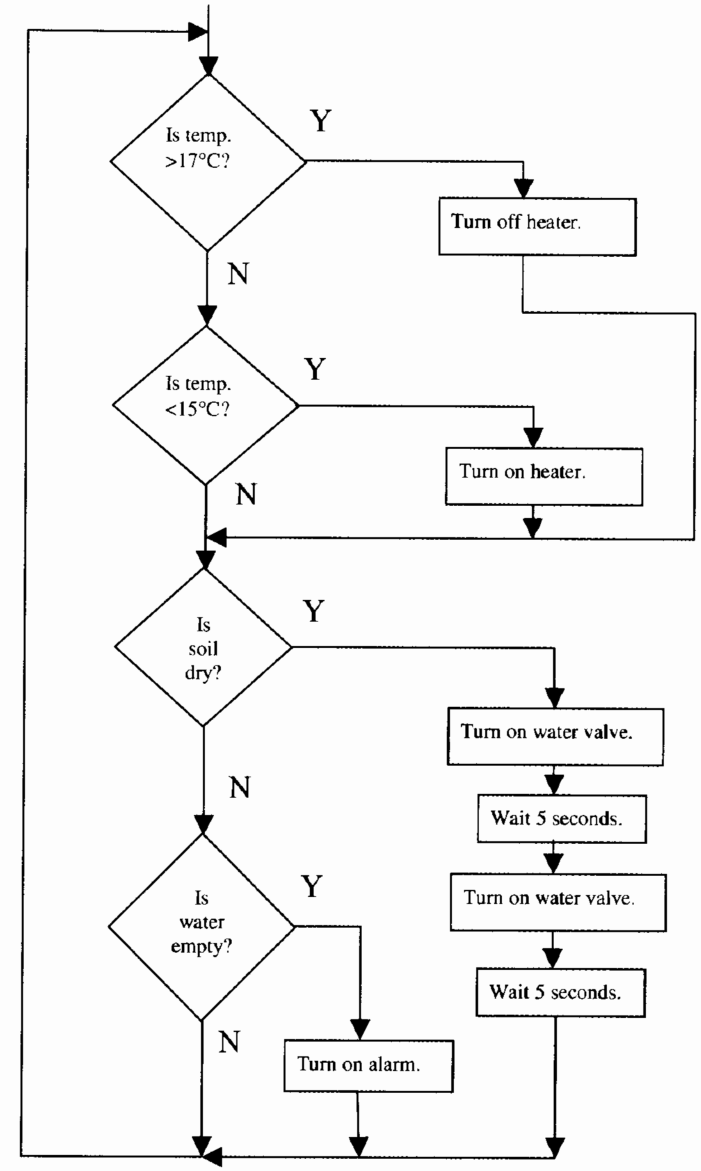

Figure 13.4 Greenhouse control flowchart

Greenhouse program

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

```
;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
;******************************************************
;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 8 we only used one 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 bit 0, i.e. BSF ADCON0,CHS0; when moving back to AN0 clear the channel select bit. The eight channels, AN0 to AN7, are seclected using bits CHS2, CHS1, CHS0.

- The temperature is read on AN0 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 goto 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 program 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 13.5.

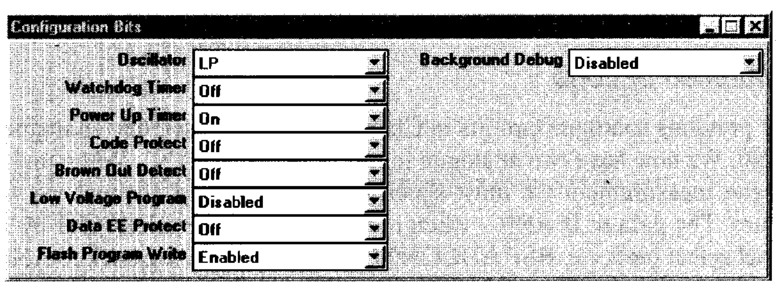

Figure 13.5 Greenhouse program configuration bits

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.
- The A/D converter configuration is adjusted using A/D configuration

register 1, i.e. ADCON1 shown in Figure 13.6.

ADFM				PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

Figure 13.6 ADCON1, A/D Port Configuration Register 1

Bit 7 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 require two result registers, ADRESH and ADRESL. Two formats are available: (a) the most significant bits of ADRESH read as 0, with ADFM = 1;

ADRESH								ADRESL							
0	0	0	0	0	0										

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.

Figure 13.7 shows the A/D Port Configuration settings for PCFG3, PCFG2, PCFG1 and PCFG0.

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

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

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

Figure 13.7 A/D Port Configuration.

14
The 16F62X microcontroller

The 16F62X family of microcontrollers includes the two devices, 16F627 and 16F628. At the time of writing they are the latest editions to the PIC microcontroller range.

The 16F62X microcontrollers are flash devices and have 18 pins and data EEPROM just like the 16F84, but they have more functions. Notably there is an onboard oscillator so an external crystal is not required. This frees up two pins for extra I/O. The 16F62X in fact can use 16 of its 18 pins as I/O.

The 16F62X devices are set to replace the popular 16F84 device – not only do they have more functions, but they are also about 30% cheaper!

Figure 14.1 shows the specification of the 16F62X devices and the 16F84 for comparison.

Device	Flash Program Memory (bytes)	RAM Data Memory (bytes)	EEPROM Data Memory (bytes)	Timer Modules	I/O Pins
16F627	1024	224	128	3	16
16F628	2048	224	128	3	16
16F84	1024	68	64	1	13

Figure 14.1 The 16F62X specification

16F62X oscillator modes

The 16F62X can be operated in eight different oscillator modes. They are selected when programming the device just like the 16F84.

The options are:

- LP Low Power Crystal, 32.768kHz
- XT 4MHz Crystal
- HS High Speed Crystal, 20MHz
- ER External Resistor (2 modes)

Newnes
An imprint of Elsevier Science
Linacre House, Jordan Hill, Oxford OX2 8DP
225 Wildwood Avenue, Woburn, MA 01801-2041

First published 2002

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN 0 7506 4812 0

For information on all Newnes publications visit our website at newnespress.com

Typeset by Avocet Typeset, Brill, Aylesbury, Bucks
Printed in Malta by Gutenberg Press Ltd

Contents