Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

PICDEM-1
USER'S GUIDE

Information contained in this publication regarding device applications and the like is intended by way of suggestion only. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip.

PICDEM-1 User’s Guide

Table of Contents

Chapter 1. Introduction

1.1 Welcome 1
1.2 PICDEM-1 Demonstration Board 2
1.3 Sample Devices 3
1.4 Sample Programs 3
1.5 PICDEM-1 User's Guide 3
1.6 Reference Documents 4

Chapter 2. Getting Started

2.1 PICDEM-1 as a Stand-alone Board – Preprogrammed Device 5
2.2 PICDEM-1 as a Stand-alone Board – Sample Programs 6
2.3 PICDEM-1 Used with the PICMASTER Emulator – Sample Programs 8
2.4 PICDEM-1 Used with MPLAB-ICE – Sample Programs 10

Chapter 3. Tutorials

3.1 Main Routine 12
3.2 Source Code – PIC16C54 13

Chapter 4. A/D Demo – PIC16C71

4.1 Main Routine 16
4.2 InitializeAD Routine 17
4.3 SetupDelay Routine 17
4.4 Detailed Code Description 18
4.5 Source Code – PIC16C71 20

Chapter 5. EEPROM Demo – PIC16F84

5.1 Main Routine 24
5.2 WriteAll Routine 25
5.3 CheckAll Routine 26

5.4 BlinkLEDs Routine 27
5.5 Detailed Code Description 27
5.6 Source Code – PIC16F84 32

Chapter 6. USART Demo – PIC17C42

6.1 Main Routine 36
6.2 Service_pref and Service_recv Routines 37
6.3 Init_ports and Init_serial Routines 38
6.4 Xmit Routine 39
6.5 Detailed Code Description 39
6.6 Source Code – PIC17C42 42

Appendix A. Hardware Detail

A.1 Processor Sockets 45
A.2 Display 45
A.3 Power Supply 45
A.4 RS-232 Serial Port 45
A.5 Switches 46
A.6 Oscillator Options 46
A.7 Analog Input 46
A.8 Sample Devices 46
A.9 Board Layout and Schematics 47

Index

Index 49

Worldwide Sales and Service

Worldwide Sales and Service 52

Chapter 1. Introduction

1.1 Welcome

Thank you for purchasing the PICDEM-1 demonstration board from Microchip Technology Incorporated. The PICDEM-1 is a simple board which demonstrates the capabilities of several Microchip microcontrollers, specifically from the PIC16C5X, PIC16C7XX, PIC16X8X, and PIC17C4X families.

The PICDEM-1 can be used stand-alone with a programmed part, or with an emulator system, such as PICMASTER®. Sample programs are provided to demonstrate the unique features of the supported devices.

The PICDEM-1 Kit comes with the following:

1. PICDEM-1 Demonstration Board
2. Sample device
3. Sample programs (3.5-inch disk)
4. PICDEM-1 Demonstration Board User's Guide (this document)

If you are missing any part of the kit, please contact your nearest Microchip sales office, listed in the back of this publication, for help.

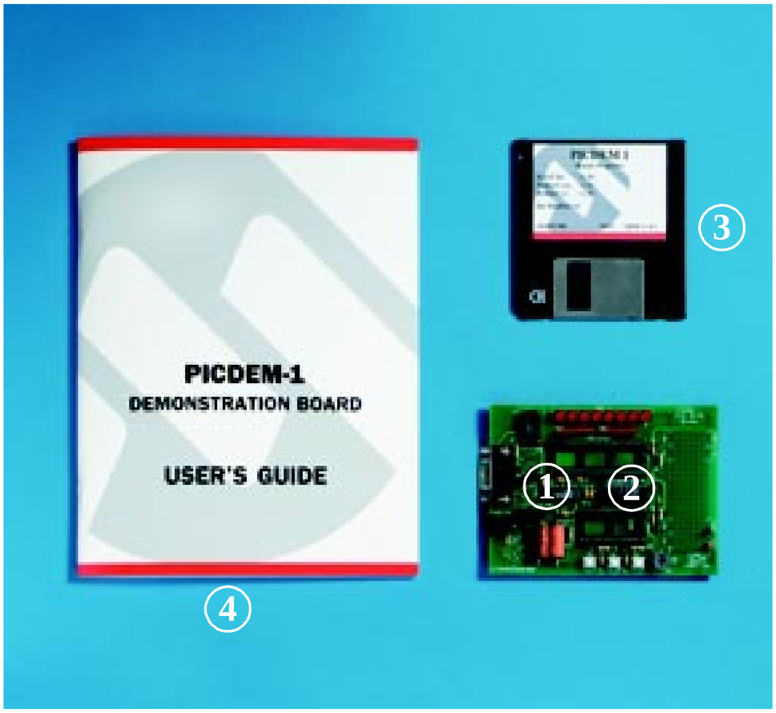

Figure 1.1: PICDEM-1 Kit

1.2 PICDEM-1 Demonstration Board

The PICDEM-1 demonstration board has the following hardware features:

1. 18-, 28- and 40-pin DIP sockets (Although 3 sockets are provided, only one device may be used at a time.)
2. On-board +5V regulator for direct input from 9V, 100 mA AC/DC wall adapter or hooks for a +5V, 100 mA regulated DC supply.
3. RS-232C socket and associated hardware for direct connection to RS-232C interface.
4. 5K pot for devices with analog inputs.
5. Three push button switches for external stimulus and RESET.
6. Eight red LEDs connected to PORTB for displaying 8-bit binary values.
7. Socket for "canned" crystal oscillator.
8. Unpopulated holes provided for crystal connection.
9. Jumper to disconnect on-board RC oscillator (approximately 2 MHz).
10. Prototype area for user hardware.

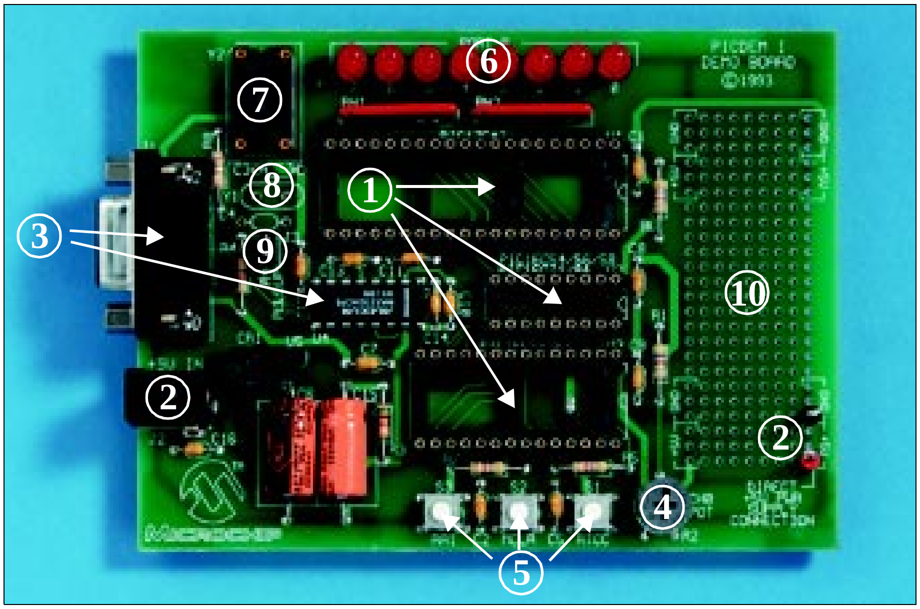

Figure 1.2: PICDEM-1 Hardware

1.3 Sample Devices

Several UV erasable devices are included. The device types may change. The supplied devices are typically one of the following:

- PIC16C54/55/56/57/58 or equivalent
- PIC16C71 or equivalent
- PIC16F83/84 or equivalent
- PIC17C42/43/44 or equivalent

1.4 Sample Programs

The PICDEM-1 Kit includes a 3.5” disk with sample demonstration programs on them. These programs may be used with the included sample device or with an emulator system. The programs are:

- `tut.asm` – Tutorial
- `demo71.asm` – PIC16C71 Demo
- `demo84.asm` – PIC16F84 Demo
- `demo42.asm` – PIC17C42 Demo

1.5 PICDEM-1 User’s Guide

This document describes the PICDEM-1 demonstration board, tutorial and demonstration software, to give the user a brief overview of the PICmicros® supported, as well as the PICMASTER. Detailed information on individual microcontrollers may be found in the device’s respective data sheet. Detailed information on the PICMASTER emulation system may be found in the *PICMASTER User’s Guide* (DS51037).

Chapter 1: Introduction – This chapter introduces the PICDEM-1 and provides a brief description of the hardware.

Chapter 2: Getting Started – This chapter goes through a basic step-by-step process for getting your PICDEM-1 up and running as a stand-alone board or on the PICMASTER emulator system.

Chapter 3: Tutorials – This chapter provides a detailed description of the tutorial programs.

Chapter 4: A/D Demo – PIC16C71 – This chapter provides a detailed description of the demonstration program for the A/D module on the PIC16C71.

Chapter 5: EEPROM Demo – PIC16F84 – This chapter provides a detailed description of the demonstration program for the data EEPROM on the PIC18F84.

Chapter 6: USART Demo – PIC17C42 – This chapter provides a detailed description of the demonstration program for the serial (RS-232) communications capabilities of the PIC17C42.

Appendix A: Hardware Detail: This appendix describes in detail the hardware of the PICDEM-1 board.

1.6 Reference Documents

Reference Documents may be obtained by contacting your nearest Microchip sales office (listed in the back of this document) or by downloading via the Microchip website (www.microchip.com).

- *Microchip Technical Library CD-ROM* (DS00161) or individual data sheets
- *MPLAB*™ *IDE, Simulator and Editor User's Guide* (DS51025)
- *MPASM User's Guide with MPLINK and MPLIB* (DS33014)
- *MPLAB PRO MATE User's Guide* (DS30082)
- *PICSTART Plus User's Guide* (DS51028)
- *MPLAB PICMASTER Emulator User's Guide* (DS30421)
- *PICMASTER Probe Specification* (DS51024)
- *MPLAB-ICE User's Guide* (DS51159)
- *Microchip Third Party Guide* (DS00104)
- *PICmicro Midrange Reference Manual* (DS33023)

Chapter 2. Getting Started

The PICDEM-1 may be used as a stand-alone board or with an emulator. The emulator discussed in this chapter is the PICMASTER. However, other emulators may be used. For a list of PICmicro-compatible emulators, please refer to the *Microchip Third Party Handbook* or the *Development Tool Ordering Guide*.

2.1 PICDEM-1 as a Stand-alone Board – Preprogrammed Device

The PICDEM-1 may be demonstrated immediately by following the steps listed below:

- Make sure the preprogrammed sample device is in the appropriate socket on the PICDEM-1 board.
- Make sure there is a jumper on J3 (to enable the on-board RC oscillator).
- Apply power to the PICDEM-1 (Figure 2.1). For information on acceptable power sources, see Appendix A.
- Press pushbutton S3 repeatedly to see the LEDs count up from 00h to FFh. Press pushbutton S2 to reset.

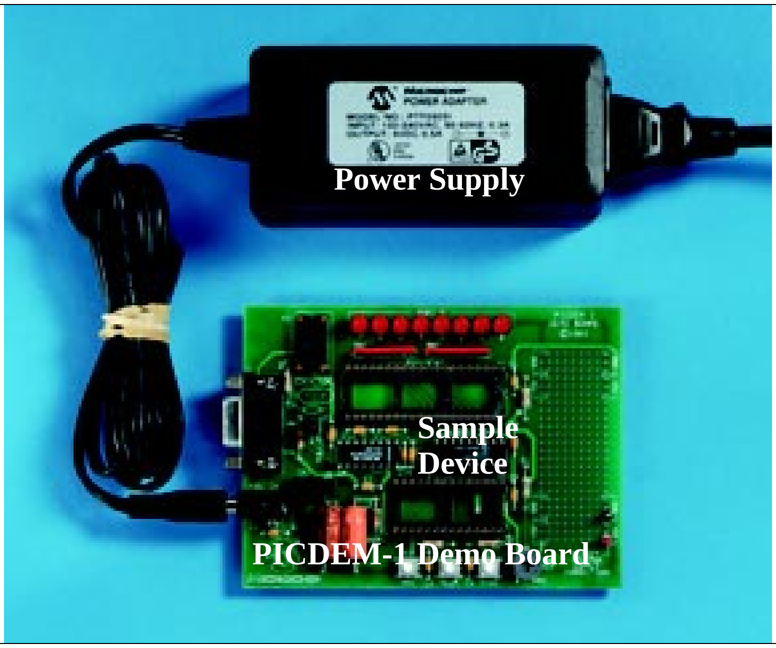

Figure 2.1: PICDEM-1 Stand-Alone

2.2 PICDEM-1 as a Stand-alone Board – Sample Programs

To demonstrate PICDEM-1 operation with one of the sample programs, the sample device will have to be erased and reprogrammed. Once the device has been reprogrammed:

- Make sure the sample device is in the appropriate socket on the PICDEM-1 board.
- Make sure there is a jumper on J3 (to enable the on-board RC oscillator).
- Apply power to the PICDEM-1 (Figure 2.1). For information on acceptable power sources, see Appendix A.
- Consult the appropriate chapter in this document for information on the execution of each demo.

2.2.1 Erasing the Sample Device

To erase an EPROM device:

- Remove any labels covering the device window. If you do not have a windowed device (Figure 2.2), you cannot reprogram it. A windowed version of all EPROM devices may be ordered by requesting the JW package.
- Place the device in an Ultraviolet (UV) EPROM Eraser. The amount of time required to completely erase a UV erasable device depends on: the wavelength of the light, its intensity, distance from UV source, and the process technology of the device (how small are the memory cells).
- Verify that the device is blank (e.g., perform a blank check) before attempting to program it.

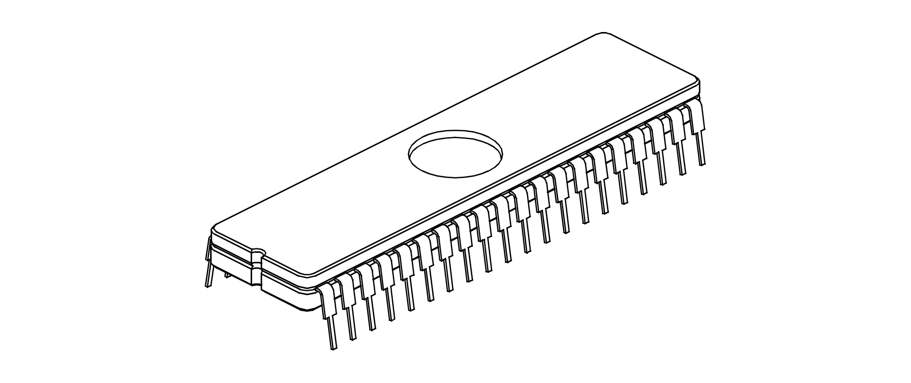

Figure 2.2: Windowed Device

To erase an EEPROM/Flash device:

- Enable your programmer in MPLAB. If you change programmers, you will have to restart MPLAB.
- Place the device in the programmer.
- Select *Erase Program Memory* or the equivalent erase command from the Programmer menu.

> **Note:** You do not have to erase an EEPROM/Flash device before reprogramming it.

2.2.2 Reprogramming the Sample Device

To reprogram the erased sample device, the following will be necessary:

1. Sample programs installed on the hard drive

   The PICDEM-1 package includes a 3.5-inch disk which contains sample programs for all the processor types supported. Instructions on how to install the programs can be found in the readme file also on the disk.

2. An assembler, such as MPASM available with MPLAB

   Sample programs may be used to program the sample device once they have been assembled. Microchip Technology's MPLAB Integrated Development Environment (IDE) includes an assembler (MPASM). However, other assemblers may be used. For a list of PICmicro-compatible assemblers, please refer to the *Microchip Third Party Handbook*.

3. A device programmer, such as PRO MATE® II or PICSTART® Plus

   Once the sample program is in hex file format, a programmer may be used to program a blank device. Microchip Technology's PRO MATE II or PICSTART Plus programmers may be used. Both are compatible with MPLAB. However, other programmers may be used. For a list of PICmicro-compatible programmers, please refer to the *Microchip Third Party Handbook*.

If the code protection bit(s) have not been programmed, the on-chip program memory can be read out for verification purposes.

> **Note:** Microchip does not recommend code protecting windowed devices.

2.3 PICDEM-1 Used with the PICMASTER Emulator – Sample Programs

To use the PICDEM-1 with the PICMASTER emulator, the following will be necessary:

1. PICDEM-1 demonstration board.
2. PICMASTER emulator pod.
3. PICMASTER device-specific probe.
4. Logic probes with power supply pins or external power supply (optional).
5. PC with MPLAB IDE software and MPLAB User's Guide.
6. Ribbon cable to connect pod with probe.
7. Pin connector to connect probe with PICDEM-1.

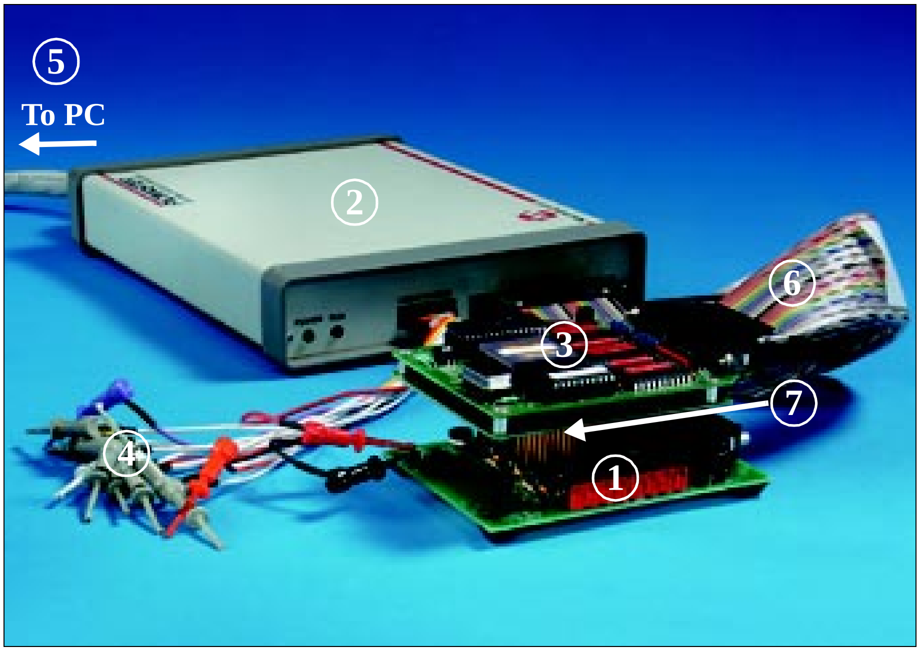

Figure 2.3: PICDEM-1 with the PICMASTER Emulator

Chapter 2. Getting Started

This chapter goes through a basic step-by-step process for getting your PICDEM-1 up and running using the PICMASTER emulator system.

1. Assemble the PICMASTER emulator
   - Follow the instructions in your PICMASTER emulator package and connect the emulator pod to an IBM® compatible PC. **Make certain that the power switches for both the PC and the PICMASTER are in the OFF position before connecting.**
   - Connect the appropriate probe to the PICMASTER pod using the ribbon cable provided (6 of Figure 2.3). For information on which probe to choose, refer to the *PICMASTER Probe Specification*.
   - Connect the logic probes to the emulator pod.
2. Prepare the PICDEM-1
   - Remove any preprogrammed microcontrollers installed in the PICDEM-1.
   - Make sure there is a jumper on J3 (to enable the on-board RC oscillator).
3. Prepare the probe – AC1650011 Example
   - Determine the power source and set J5 accordingly; +5 VSYS comes from the PICMASTER system (internal) or +5 VEXT comes from an external power source.
   - Determine whether or not to enable master clear $\overline{\text{MCLR}}$ by connecting/disconnecting J6. It is recommended that J6 be connected ($\overline{\text{MCLR}}$ enabled) for most applications.
   - Select the oscillator option to be used. Table 2.1 lists the modifications necessary for each option.

TABLE: 2.4 OSCILLATOR SELECTION

Oscillator selection on PICDEM-1	Modification on PICDEM-1	Modification on Probe
RC	J3 installed, Y2 empty, Y1 empty	Select J4 = EXTCLK, Select SW1 = 11 (RC)
Crystal	J3 removed, Y2 empty, crystal in Y1, caps C3 and C4 installed	Select J4 = EXTCLK, Select SW1 = 10 (HS), 01 (XT) or 00 (LP)
Canned Crystal	J3 removed, oscillator in Y2, Y1, C3, C4 empty	Select J4 = EXTCLK, Select SW1 = 10 (HS), 01 (XT) or 00 (LP)
PICMASTER’s Oscillator	J3 removed, Y2, Y1 empty	Select J4 = INTCLK

4. Connect the PICDEM-1 to the emulator.
   - Connect the PICMASTER header probe to the appropriate socket on the PICDEM-1 using a connector (7 of Figure 2.3). Example: If you are emulating the PIC16F84, connect the 18-pin header on the probe card to the 18-pin socket on PICDEM-1 (U2).
5. Power supply considerations.
   - If the PICMASTER power probes are to be used to power the PICDEM-1, connect the Red probe clip to the hook marked +5V on PICDEM-1 and the Black probe clip to the hook marked GND on PICDEM-1.

> **Note:** If an external wall mount power supply is being used, **do not** connect it to the PICDEM-1 yet.

6. Power-up sequence

   The following power up sequence must be followed **exactly**:
   - Power up the PC.
   - Start Microsoft® Windows®.
   - Power up the PICMASTER.
   - If an external wall mount power supply is being used for the PICDEM-1, connect it now.
7. Installing the sample programs

   The PICDEM-1 package includes a 3.5-inch disk which contains sample programs for all the processor types supported. Instructions on how to install the programs can be found in the readme file also on the disk.
8. Starting MPLAB
   - Start MPLAB by double clicking on the MPLAB icon (Windows 3.1 or later) or selecting it from the Start menu (Windows 95/98).
   - Refer to the *MPLAB User's Guide* for how to configure the application to work with the emulator and how to run the demonstration programs.

2.4 PICDEM-1 Used with MPLAB-ICE – Sample Programs

Microchip's latest emulator, MPLAB-ICE, may be used with the PICDEM-1 board by using PICMASTER probes. Refer to the *MPLAB-ICE User's Guide* (DS51159) for information on how to use PICMASTER probes with MPLAB-ICE. Then follow the setup directions from the previous section.

Chapter 3. Tutorials

The `tut.asm` tutorial program is pre-programmed into the sample device. This program is listed on the included 3.5-inch program disk for user reference. For example, if the sample device has been reprogrammed with another sample program, the tutorial may be reassembled and reprogrammed into the device.

The tutorial program functions as follows. Pressing the switch S3 causes the LEDs to binary count up to FFh (255 decimal). Past FFh (all LEDs on), the count rolls over to zero (all LEDs off). The program may be reset by pressing the switch S2.

For detailed information on the PICDEM-1 hardware, please refer to Appendix A.

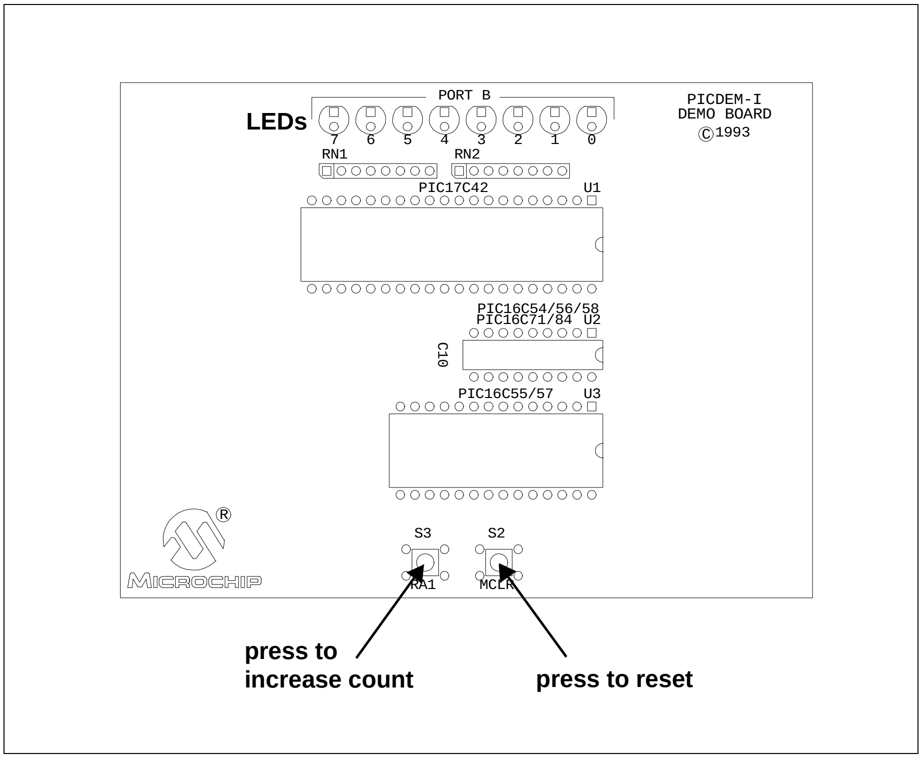

Figure 3.1: Tutorial

3.1 Main Routine

The tutorial program is extremely simple. It begins by configuring PORTB as all outputs to drive the eight LEDs and initializing a counter. Then it waits in an infinite loop until S3 is pressed. This increases the counter by one, and this number is displayed in binary on the LEDs. Once the count reaches FFh, the next S3 press will roll the count back to 00h.

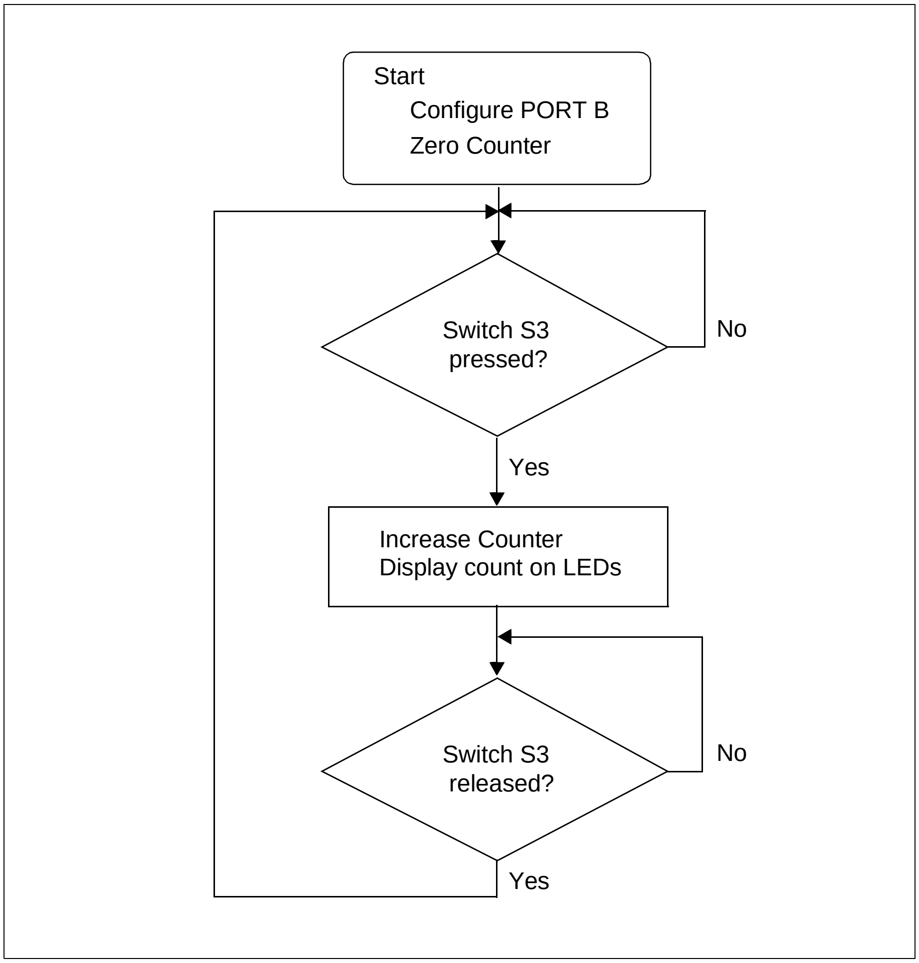

Figure 3.2: Main Routine

3.2 Source Code – PIC16C54

The tutorial program listed below is for the PIC16C54 device. It can be modified for all other supported devices. See files titled `tut??.asm`, where ?? represents the last two numbers of a device.

```
             list p=16c54
;
;  This program runs on the PICDEM-1 demo board.
;  In the Demo board, Port B is connected to 8 LEDs.
;  RA1 is connected to a switch (S3). This program increments
;  the file register count each time S3 is pressed.
;  The value of count is displayed on the LEDs connected
;  to Port B.
;  Net result is that Leds should increment in a binary
;  manner every time S3 is pressed.
;
     #include <P16C5X.INC>
;
COUNT   equ     0x10

;
     org     00h
Start
     movlw   0
     movwf   PORTB           ; config port b as output
     tris    PORTB
     clrf    COUNT           ; clr count
Loop
     btfss   PORTA,1         ; see if RA1 pressed
     goto    IncCount        ; yes then inc count
Endloop
     goto    Loop            ; else check again
IncCount
     incf    COUNT,F         ;inc count
     movf    COUNT,W         ;
     movwf   PORTB           ; display on port b
Debounce
     btfss   PORTA,1         ; wait for key release
     goto    Debounce        ; not release then wait
Enddebounce
     goto    Loop            ; else check key press again
;
;
     org     0x01FF           ; 16C54 reset vector.
     goto    Start
;
     end
```

NOTES:

Chapter 4. A/D Demo – PIC16C71

The `demo71.asm` program is a simple implementation of the PIC16C71's analog-to-digital (A/D) converter. The program reads A/D channel 0 and displays the results on the LEDs connected to PORTB. If the potentiometer is turned all the way clockwise, all of the LEDs are off. If the potentiometer is turned all the way counter-clockwise, all of the LEDs are turned on.

PIC16C5X and PIC16X8X devices do not have an on-board A/D converter. However, an external A/D circuit may be constructed in the prototype area and the board modified to accommodate it. For more information, please refer to *AN513 – Analog to Digital Conversion Using a PIC16C54* (DS00513).

For more information on A/D module operation, please refer to the *PICmicro Midrange Microcontroller Family Reference Manual* for an operational description and a list of related application notes. For detailed information on the PICDEM-1 hardware, please refer to Appendix A.

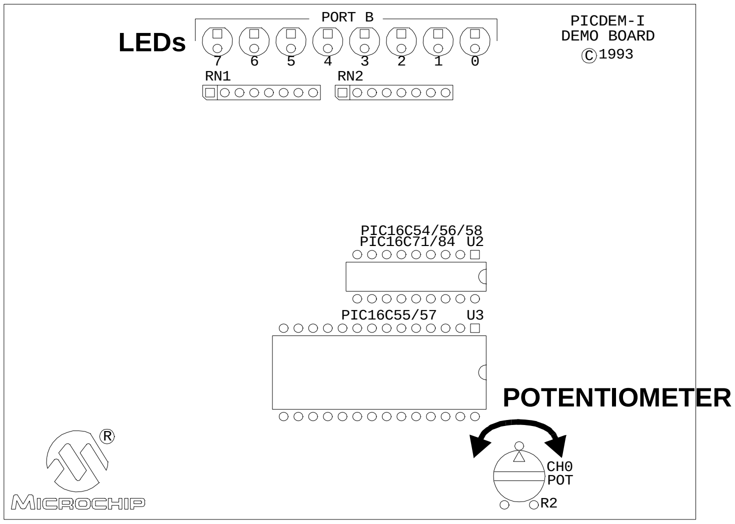

Figure 4.1: A/D Demo

4.1 Main Routine

The main routine of `demo71.asm` handles the initialization of the PIC16C71 and then uses the A/D interrupt flag to signal a completed A/D conversion and update the value on the LEDs. The first few lines set PORTB as outputs (for the LEDs). `InitializeAD`, the Analog-to-Digital converter initialization routine, is then called. After the return from `InitializeAD`, the value in the A/D output register is retrieved and displayed on the LEDs. The routine then waits for T_{AD} (A/D conversion clock) and then starts an A/D conversion of a value specified by the potentiometer. When the A/D conversion is complete (ADIF bit set), the program loops back to display the current conversion value on the LEDs and to get another A/D value for conversion.

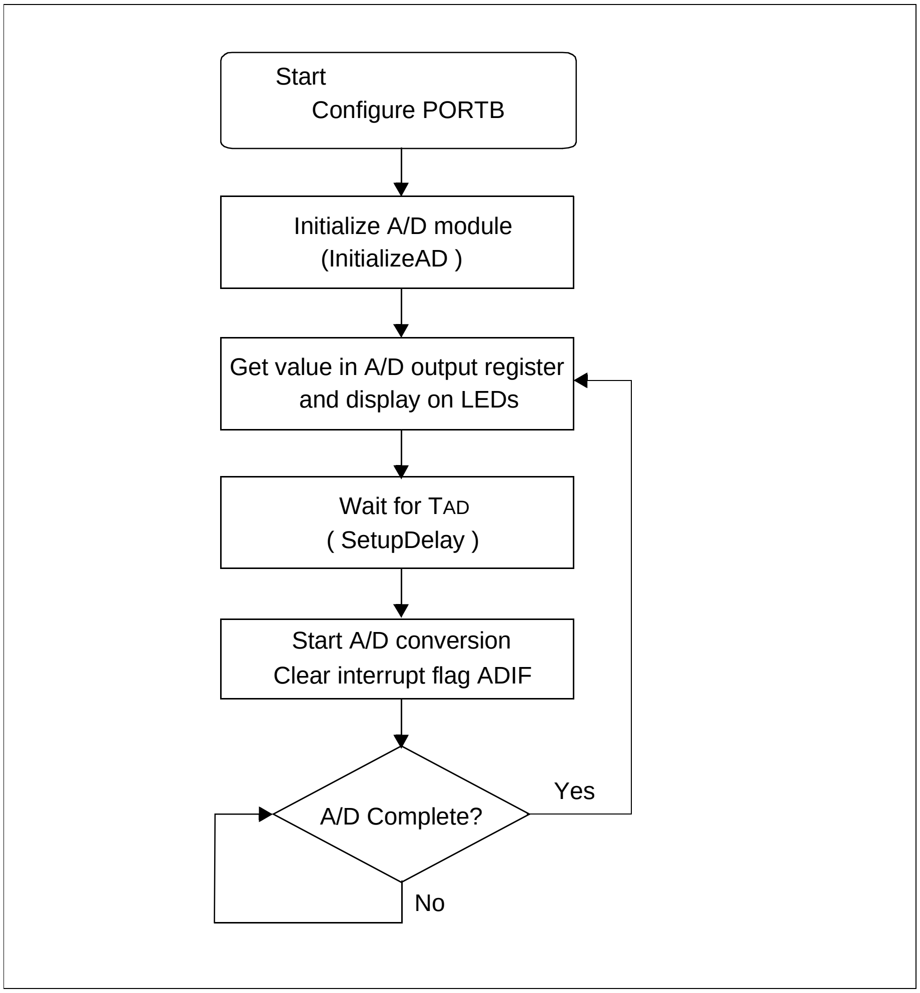

Figure 4.2: Main Routine

4.2 InitializeAD Routine

The `InitializeAD` routine sets up the A/D converter for use with the PICDEM-1 board. The routine sets Ch0 through Ch3 as analog inputs, sets the A/D to use the internal RC oscillator, selects channel 0, and clears the A/D result register.

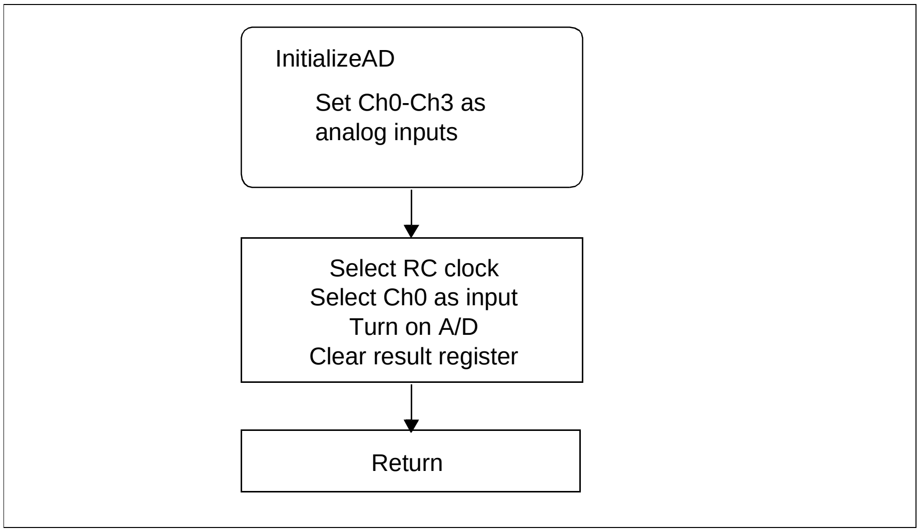

Figure 4.3: Initialize A/D

4.3 SetupDelay Routine

This subroutine delays for greater than one A/D converter clock period (2 to 6 µs when using the internal RC clock as this program does) to allow for sampling time.

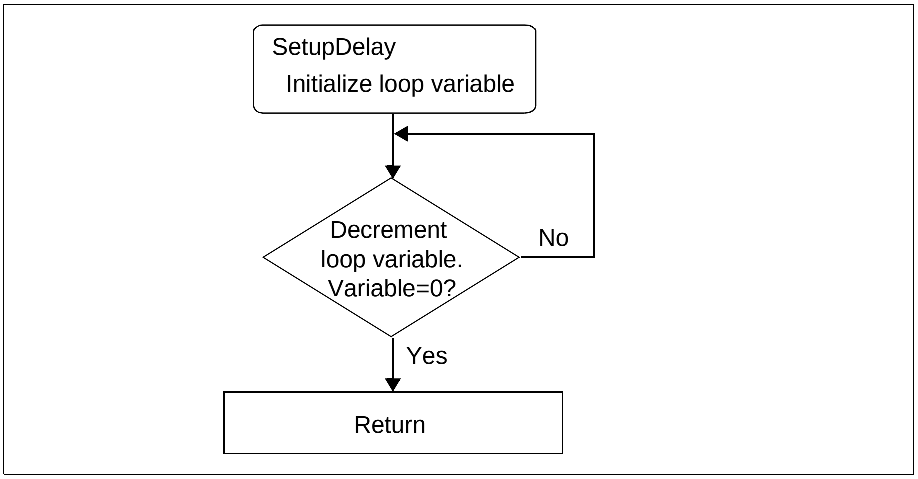

Figure 4.4: Setup Delay

4.4 Detailed Code Description

The program starts out by defining three constants: TEMP, a general purpose file register; ADIF, the bit location (1) of the A/D interrupt flag in ADCON0; and ADGO, the A/D GO/DONE bit in ADCON0. The next sections of code define the reset and interrupt vectors.

```
TEMP    equ     10h
ADIF    equ     1               ;A/D interrupt flag
ADGO    equ     2               ;A/D enable bit
;
     org     0x00
     goto    Start              ;reset vector
;
     org     0x04
     goto    Service_int        ;interrupt vector
```

The actual program begins at Start, where PORTB is defined as all outputs by writing 0’s to the PORTB tristate register TRISB. Note that 0’s were written to the PORTB register prior to setting the port to outputs so that the port would initialize with all outputs in the low state.

```
Start
     movlw   B’00000000’     ;set port b as
     movwf   PORTB           ;all outputs
     bsf     STATUS, RP0
     CLRF    TRISB           ;       /
     bcf     STATUS, RP0
;
     call    InitializeAD
```

After initializing PORTB, `InitializeAD` is called. The page register is set for page 1, RA0 through RA3 are selected to be analog inputs, and VREF is selected to be VDD (if all four pins are selected to be analog inputs, VREF is set to VDD by default) by writing to ADCON1. The page register is then reset for page 0, the internal RC oscillator is selected for the conversion clock, and ADON, the A/D on/off bit is set so that the A/D converter module is operating. The last action this routine takes before returning is to clear the A/D result register.

```
InitializeAD
     bsf     STATUS,5        ;select pg1
     movlw   B’00000000’     ;select ch0-ch3...
     movwf   ADCON1          ;as analog inputs
     bcf     STATUS,5        ;select pg0
     movlw   B’11000001’     ;select:RC,ch0..
     movwf   ADCON0          ;turn on A/D.
     clrf    ADRES           ;clr result reg.
     return
```

The Update routine reads the A/D result register, writes the value it got to the LEDs connected to PORTB, calls `SetupDelay`, clears the interrupt flag and enables the next conversion.

```
Update
     movf    ADRES,W          ;get a/d value
     movwf   PORTB            ;output to port b
     call    SetupDelay       ;setup time >= 20uS.
     bcf     ADCON0,ADIF      ;clear int flag
     bsf     ADCON0,ADGO      ;start new conversion
```

The `SetupDelay` routine provides a software delay so that the sample and hold circuit has enough time to settle before the A/D conversion is started. The `SD` loop takes 2 µs when TEMP is not equal to zero, so a value of nine loaded into TEMP would provide a delay of 18 µs. With the 2 µs execution time of the two moves before the `SD` loop and the call and return execution times, a delay of greater than 20 µs is provided for settling.

```
SetupDelay
     movlw   .3
     movwf   TEMP
SD
     decfsz  TEMP, F
     goto    SD
     return
```

The Loop routine tests the A/D interrupt flag, and loops until it indicates that the conversion has finished (ADIF = 0). When the conversion has finished, it jumps back to Loop so that the A/D value can be read and output to the LEDs.

```
Loop
     btfsc   ADCON0,ADIF      ;a/d done?
     goto    Update           ;yes then update new value.
     goto    Loop             ;no then keep checking
```

4.5 Source Code – PIC16C71

```
             LIST P=16C71
;
;  TITLE   "Single channel A/D (SAD)"
;  This program is a simple implementation of the PIC16C71’s
;  A/D. One channel is used (CH0).
;  The A/D is configured as follows:
;       Vref = +5V internal.
;       A/D Osc. =  internal RC
;       A/D Channel = CH0
;  Hardware for this program is the PICDEM1 board. The program
;  converts the potentiometer value on CH0 and displays it as
;  a 8 bit binary value on PORTB.
;
;
       #include <P16C71.INC>
;
TEMP    equ     10h
ADIF    equ     1               ;A/D interrupt flag
ADGO    equ     2               ;A/D enable bit
;
     org     0x00
     goto    Start           ;reset vector
;
     org     0x04
     goto    Service_int     ;interrupt vector
;
;
     org     0x10
Start
     movlw   B’00000000’     ;set port b as
     movwf   PORTB           ;all outputs
     bsf     STATUS, RP0
     CLRF    TRISB           ;       /
     bcf     STATUS, RP0
;
     call    InitializeAD
Update
     movf    ADRES,W         ;get a/d value
     movwf   PORTB           ;output to port b
     call    SetupDelay      ;setup time >= 20uS.
     bcf     ADCON0,ADIF     ;clear int flag
     bsf     ADCON0,ADGO     ;start new conversion
Loop
     btfsc   ADCON0,ADIF     ;a/d done?
     goto    Update          ;yes then update new value.
     goto    Loop            ;no then keep checking
;
```

```
;No interrupts are enabled, so if the program ever reaches here,
;it will be returned with the global interrupts disabled.

Service_int
     return                  ;do not enable global.
;
;
;
;Initializes and sets up the A/D hardware.
;Select ch0 to ch3 as analog inputs, fosc/2 and read ch0.
;
InitializeAD
     bsf     STATUS,5        ;select pg1
     movlw   B’00000000’     ;select ch0-ch3...
     movwf   ADCON1          ;as analog inputs
     bcf     STATUS,5        ;select pg0
     movlw   B’11000001’     ;select:RC,ch0..
     movwf   ADCON0          ;turn on A/D.
     clrf    ADRES           ;clr result reg.
     return
;
;This routine is a software delay of 20uS for the a/d setup.
;At 4Mhz clock, the loop takes 2uS, so initialize TEMP with
;a value of 9 to give 18uS, plus the move etc. should result in
;a total time of > 20uS.

SetupDelay
     movlw   .3
     movwf   TEMP
SD
     decfsz  TEMP, F
     goto    SD
     return


     END
```