

**PICDEM-2**  
**USER'S GUIDE**

---

---

# PICDEM-2 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.

The Microchip logo and name are trademarks of Microchip Technology Incorporated. PIC, PICmicro, PICSTART, PRO MATE, and MPLAB are a registered trademark of Microchip Technology Inc. in the U.S.A. and other countries.

© 2000 Microchip Technology Incorporated.

ACCESS.bus is a trademark of the ACCESS.bus Industry Group.

C&K is a registered trademark of C&K Components Inc.

Digi-Key is a registered trademark of Digi-Key Corporation.

IBM and PC/AT are registered trademarks of International Business Machines Corporation.

I<sup>2</sup>C is a trademark of Philips Corporation.

MS-DOS and MS Windows are trademarks of Microsoft Corporation.

Newark Electronics is a registered trademark of Premier Industrial Corporation.

Optrex is a registered trademark of Optrex America.

All rights reserved. All other trademarks mentioned herein are the property of their respective companies.

# PICDEM-2 User's Guide

---

---

---

---

**Table of Contents**

---

---

**Chapter 1. Introduction**

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

**Chapter 2. Getting Started**

2.1	PICDEM-2 as a Stand-alone Board – Preprogrammed Device .....	5
2.2	PICDEM-2 as a Stand-alone Board – Sample Programs .....	6
2.3	PICDEM-2 Used with an In-Circuit Emulator .....	8

**Chapter 3. Tutorials**

3.1	Main Routine .....	10
-----	--------------------	----

**Chapter 4. A/D Demo**

4.1	Main Routine .....	12
4.2	InitializeAD Routine .....	13
4.3	SetupDelay Routine .....	13
4.4	Service_Int Routine .....	14

**Chapter 5. I<sup>2</sup>C/EEPROM Demo**

5.1	SSP Module .....	16
5.2	MSSP Module .....	25

**Chapter 6. LCD Demo**

6.1	4-Bit Interface .....	34
6.2	8-Bit Interface .....	37

# PICDEM-2 User's Guide

---

## Chapter 7. USART Demo

7.1	Main Routine .....	42
7.2	IntVector Routine .....	43
7.3	RcvError Routine .....	44

## Appendix A. Hardware Detail

A.1	Processor Sockets .....	45
A.2	Display .....	45
A.3	Power Supply .....	45
A.4	RS-232 Serial Port .....	46
A.5	Switches .....	46
A.6	Oscillator Options .....	46
A.7	Analog Input .....	46
A.8	Serial EEPROM .....	47
A.9	Keypad Header .....	47
A.10	LCD Header .....	47
A.11	Sample Devices .....	47
A.12	Board Layout and Schematics .....	48

<b>Worldwide Sales and Service .....</b>	<b>52</b>
--	-----------

---

---

## Chapter 1. Introduction

---

---

### 1.1 Welcome

Thank you for purchasing the PICDEM-2 demonstration board from Microchip Technology Incorporated. The PICDEM-2 is a simple board which demonstrates the capabilities of the 28- and 40-pin PIC16CXXX and PIC18CXXX devices.

The PICDEM-2 can be used stand-alone with a programmed part, or with an emulator system, such as MPLAB<sup>®</sup>-ICE. Sample programs are provided to demonstrate the unique features of the supported devices.

The PICDEM-2 Kit comes with the following:

1. PICDEM-2 Demonstration Board
2. Sample devices
3. Sample programs (3.5-inch disk)
4. PICDEM-2 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.

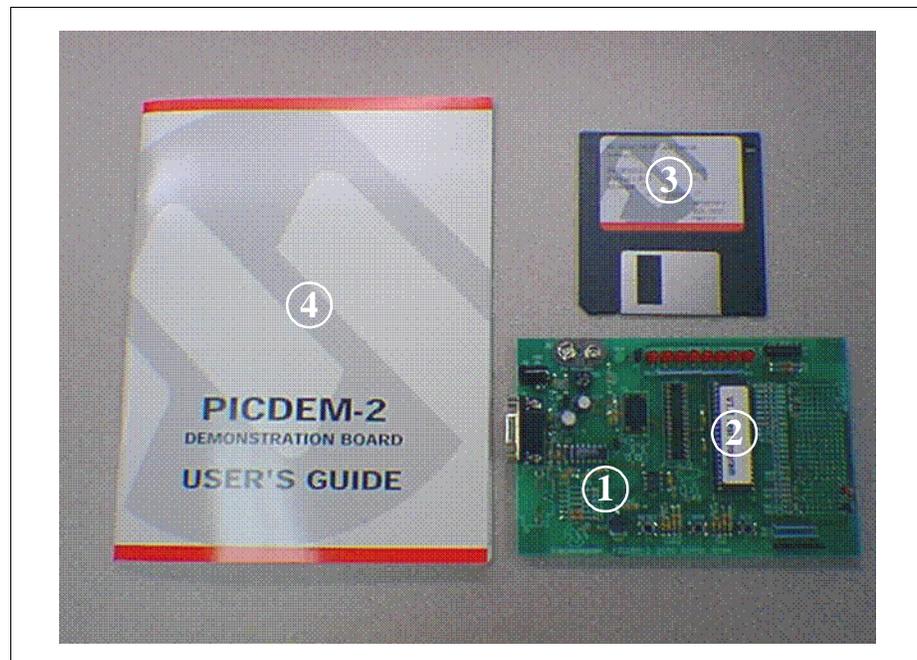


Figure 1.1: PICDEM-2 Kit

# PICDEM-2 User's Guide

---

## 1.2 PICDEM-2 Demonstration Board

The PICDEM-2 demonstration board has the following hardware features:

1. 40- and 28-pin DIP sockets (Although 2 sockets are provided, only one device may be used at a time.)
2. On-board +5V regulator for direct input from 9V, 100mA AC/DC wall adapter or 9V battery, or hooks for a +5V, 100mA regulated DC supply.
3. RS-232C socket and associated hardware for direct connection to RS-232C interface.
4. Unpopulated holes for ACCESS.bus™ connector and associated hardware.
5. 5K pot for devices with analog inputs.
6. Three push button switches for external stimulus and RESET.
7. Green power-on indicator LED.
8. Eight red LEDs connected to PORTB for displaying 8-bit binary values.
9. Jumper to disconnect LEDs from PORTB.
10. Socket for “canned” crystal oscillator.
11. Unpopulated holes provided for crystal connection.
12. Unpopulated holes provided for Timer1 external crystal.
13. Jumper to disconnect on-board RC oscillator (approximately 2MHz).
14. 128 x 8 Serial EEPROM.
15. LCD module header.
16. Keyboard header.
17. Prototype area for user hardware.

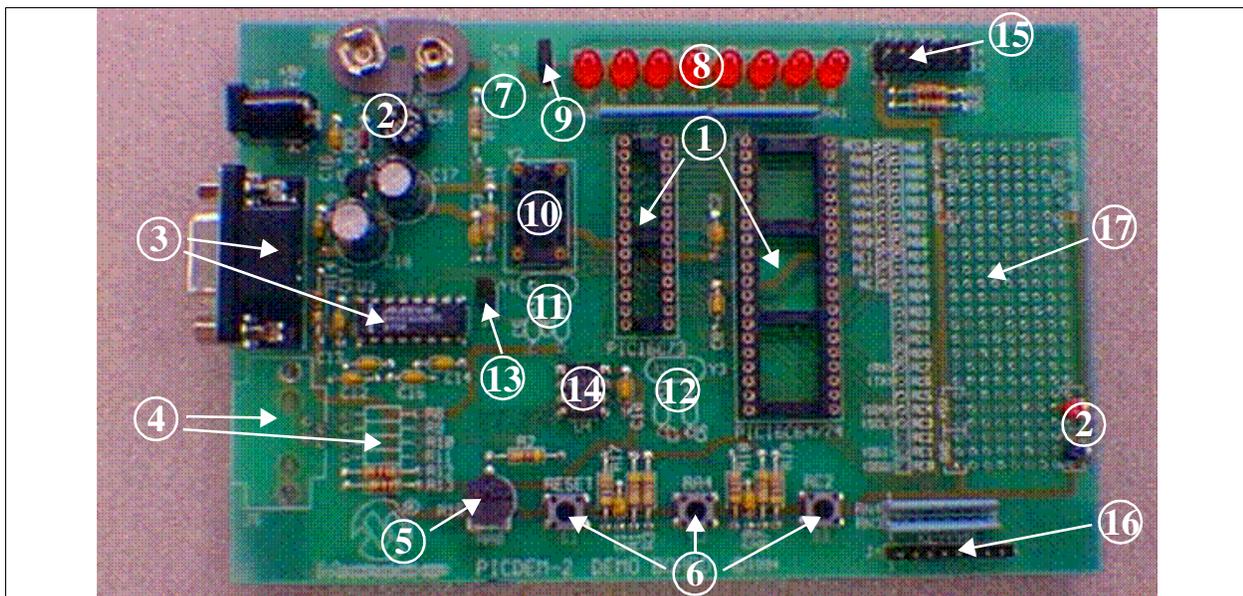


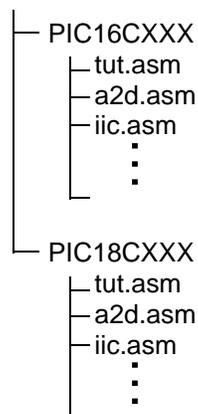
Figure 1.2: PICDEM-2 Hardware

## 1.3 Sample Devices

Several UV erasable devices are included. The device types may change, but will generally include a 28-pin and 40-pin device.

## 1.4 Sample Programs

The PICDEM-2 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 directory structure of the 3.5-inch disk is:



## 1.5 PICDEM-2 User's Guide

This document describes the PICDEM-2 demonstration board, and tutorial and demonstration software, to give the user a brief overview of the PIC16CXXX series of Microchip microcontrollers, as well as MPLAB IDE. Detailed information on individual microcontrollers may be found in the device's respective data sheet. Detailed information on the emulation systems may be found in the respective emulator's user's guide.

**Chapter 1:** Introduction – This chapter introduces the PICDEM-2 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-2 up and running as a stand-alone board or with an emulator.

**Chapter 3:** Tutorial – This chapter provides a detailed description of the tutorial program.

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

**Chapter 5:** I<sup>2</sup>C/EEPROM Demo – This chapter provides a detailed description of the demonstration program for the SSP and MSSP modules in I<sup>2</sup>C mode. The program uses I<sup>2</sup>C to write to the serial EEPROM.

# PICDEM-2 User's Guide

---

**Chapter 6:** LCD Demo – This chapter provides a detailed description of the demonstration program for interfacing a keypad to an LCD. A 4-bit and an 8-bit interface program is provided.

**Chapter 7:** USART Demo – This chapter provides a detailed description of the demonstration program for the USART module.

**Appendix A:** Hardware Description: This appendix describes in detail the hardware of the PICDEM-2 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](http://www.microchip.com)).

- *Microcontroller Data Book* (DS00158) 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<sup>®</sup> User's Guide* (DS30082)
- *PICSTART<sup>®</sup> Plus User's Guide* (DS51028)
- *MPLAB-ICE User's Guide* (DS51159)
- *Microchip Third Party Guide* (DS00104)

---

---

## Chapter 2. Getting Started

---

---

The PICDEM-2 may be used as a stand-alone board or with an emulator. For a list of PICmicro<sup>®</sup> 8-bit microcontroller-compatible emulators, please refer to the *Microchip Third Party Handbook* or the *Development Tool Ordering Guide*.

### 2.1 PICDEM-2 as a Stand-alone Board – Preprogrammed Device

The PICDEM-2 may be demonstrated immediately by following the steps listed below:

- Make sure the pre-programmed sample device is in the appropriate socket on the PICDEM-2 board.
- Make sure there are jumpers on J6 (to enable the LEDs) and J7 (to enable the on-board RC oscillator).
- Apply power to the PICDEM-2 (Figure 2.1). For information on acceptable power sources, see Appendix A.
- Press push-button S2 repeatedly to see the LEDs count up from 00h to FFh. Press push-button S1 to reset.

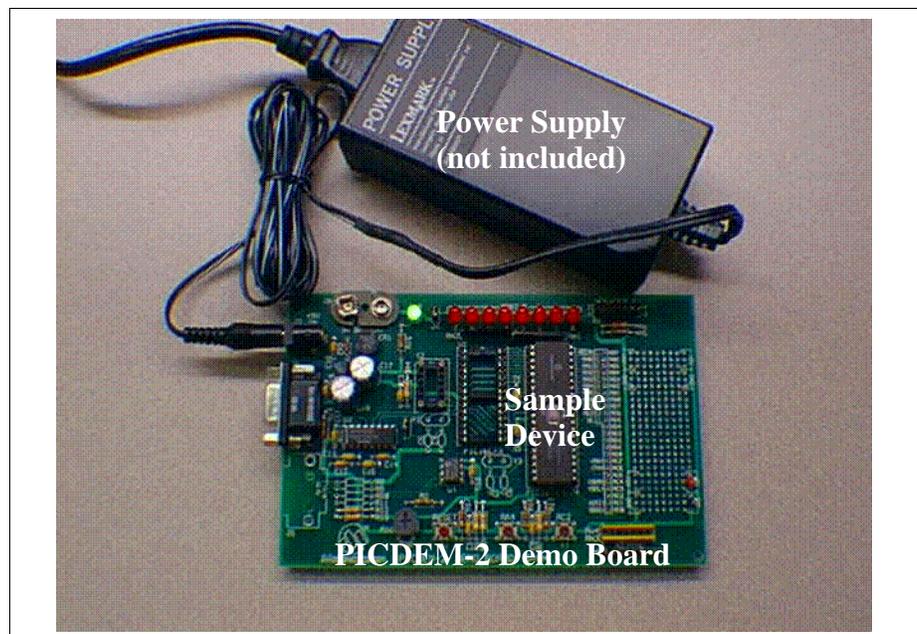


Figure 2.1: PICDEM-2 Stand-Alone

# PICDEM-2 User's Guide

---

## 2.2 PICDEM-2 as a Stand-alone Board – Sample Programs

To demonstrate PICDEM-2 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-2 board.
- Make sure there are jumpers on J6 (to enable the LEDs) and J7 (to enable the on-board RC oscillator).
- Apply power to the PICDEM-2 (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 (i.e., perform a blank check) before attempting to program it.

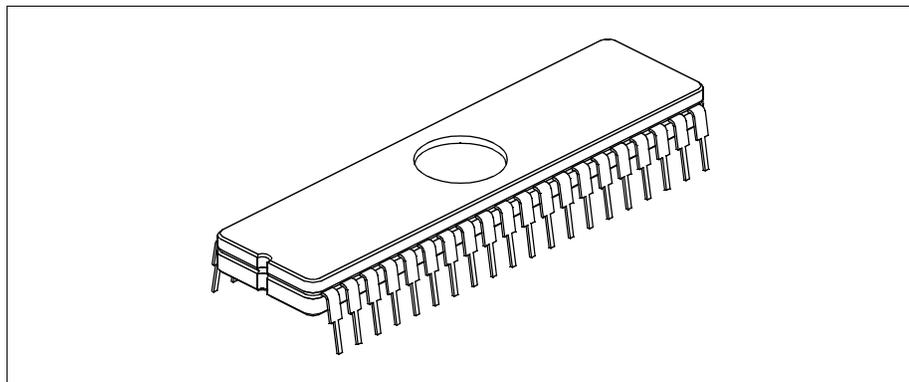


Figure 2.2: PIC16C74 Windowed Device

## 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-2 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 IDE.

Sample programs may be used to program the sample device once they have been assembled. Microchip Technology's MPLAB Integrated Development Environment 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.

<p><b>Note:</b> Microchip does not recommend code protecting windowed devices.</p>
--

# PICDEM-2 User's Guide

---

## 2.3 PICDEM-2 Used with an In-Circuit Emulator

To use PICDEM-2 with an in-circuit emulator, refer to the emulator's user's guide for instructions on how to connect, power up and configure the emulator to the PICDEM-2.

Configure the PICDEM-2 for the desired oscillator as described in Table 2.1. Refer to the emulator's user's guide for any oscillator configuration requirements.

**Table: 2.1 OSCILLATOR SELECTION**

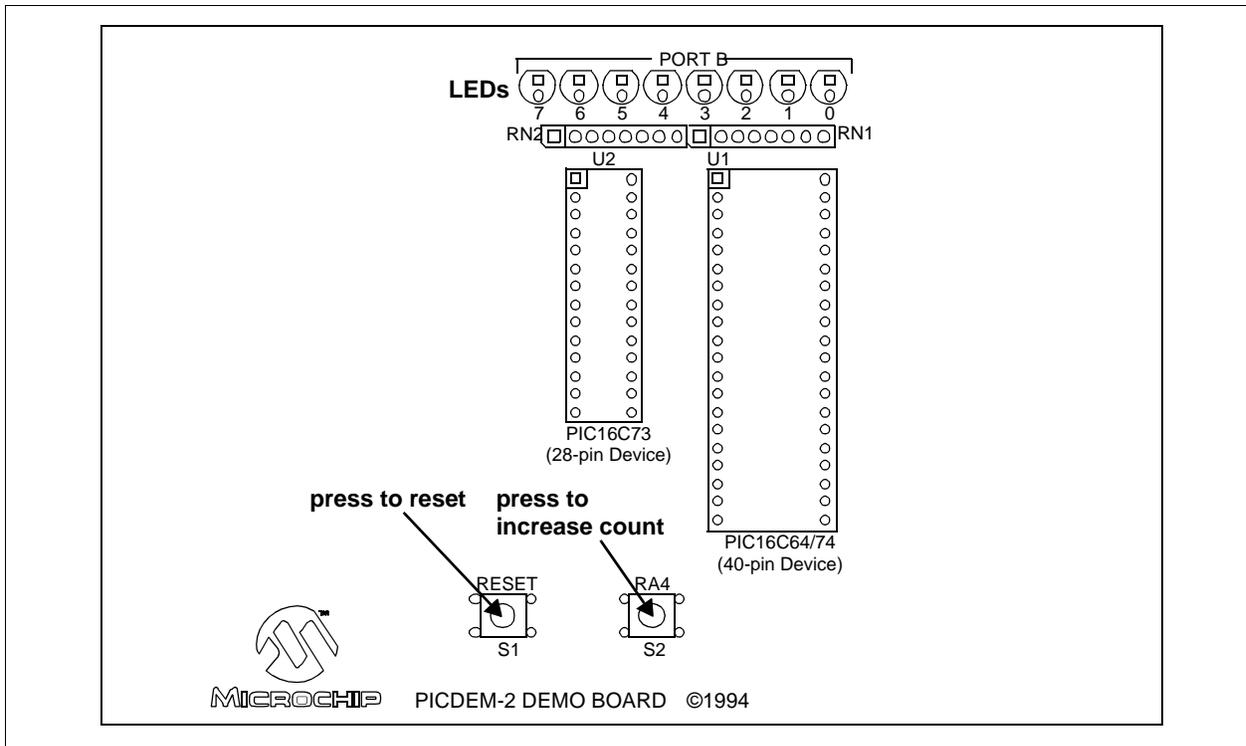
<b>Oscillator Selection on PICDEM-2</b>	<b>Modification on PICDEM-2</b>
RC	J7 installed, Y2 empty, Y1 empty
Crystal	J7 removed, Y2 empty, crystal in Y1, caps C2 and C3 installed
Canned Crystal	J7 removed, oscillator in Y2, Y1, C2, C3 empty
Oscillator	J7 removed, Y2, Y1 empty

**Chapter 3. Tutorials**

The `tut.asm` tutorial program is pre-programmed into the sample device. This program is listed on the included 3.5" program disk for user reference. I.e., 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 S2 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 S1.

For detailed information on the PICDEM-2 hardware, please refer to Appendix A.



**Figure 3.1: Tutorial**

# PICDEM-2 User's Guide

---

## 3.1 Main Routine

The tutorial program is extremely simple. It begins by configuring PORTB as all outputs to drive the 8 LEDs and initializing a counter. Then it waits in an infinite loop until S2 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 S2 press will roll the count back to 00h.

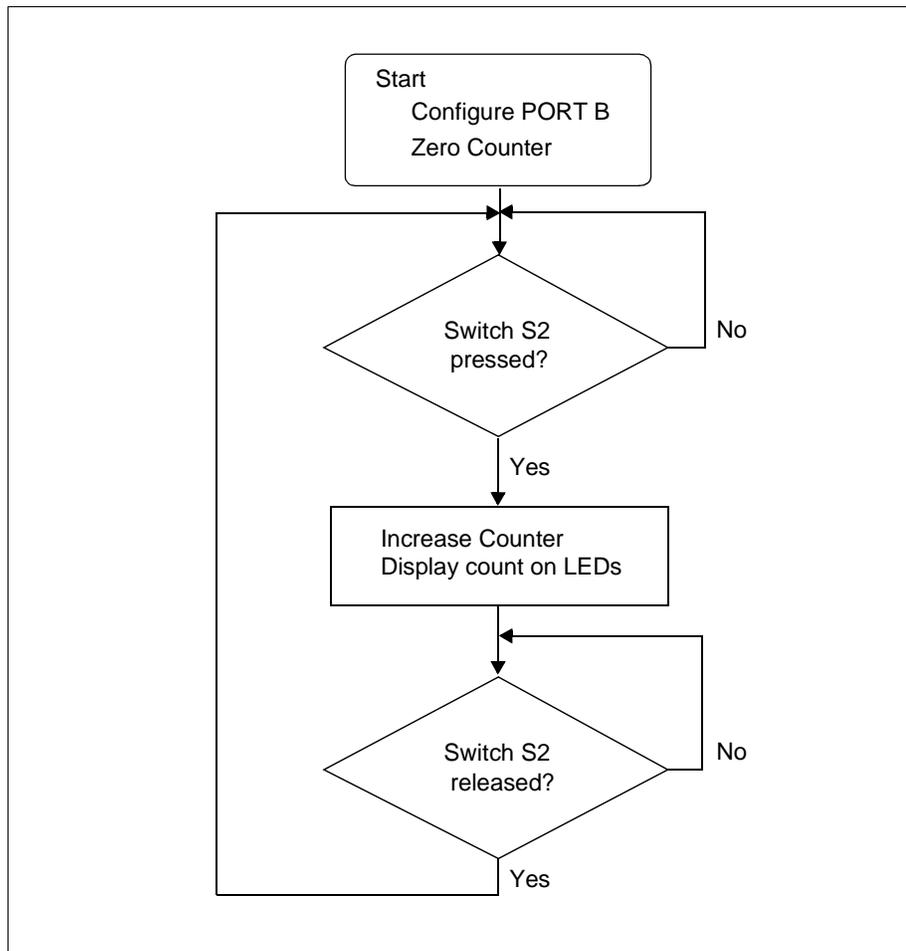
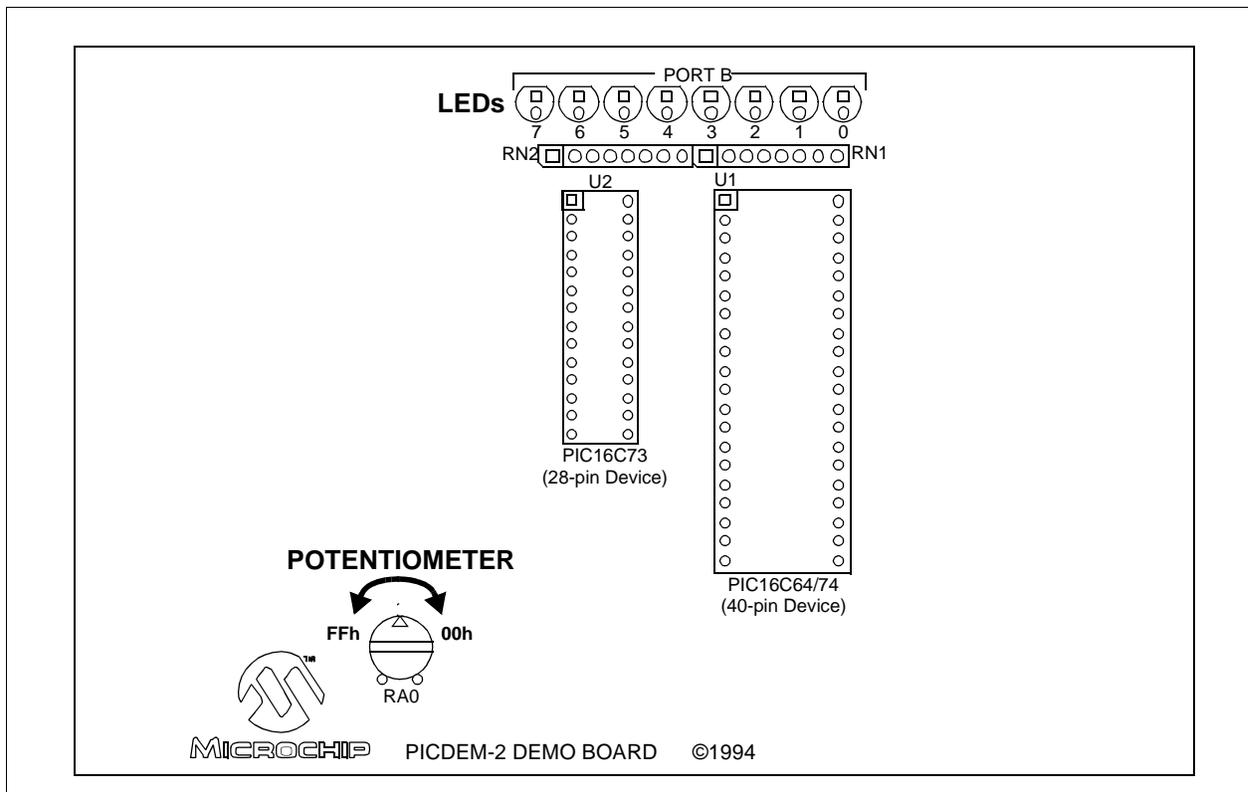


Figure 3.2: Main Routine

**Chapter 4. A/D Demo**

The `a2d.asm` program is a simple implementation of the 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. The program is interrupt driven, i.e., the program sits in an infinite loop until the A/D converter interrupts when the conversion is complete and then updates the values on the PORTB LEDs.

For more information on A/D module operation, please refer to the *PICmicro Mid-range Microcontroller Family Reference Manual* for an operational description and a list of related application notes. For detailed information on the PICDEM-2 hardware, please refer to Appendix A.



**Figure 4.1: A/D Demo**

# PICDEM-2 User's Guide

---

## 4.1 Main Routine

The main routine of `a2d.asm` handles the initialization of the PICmicro MCU and then waits in an infinite loop. The first few lines set PORTA as inputs and PORTB as outputs. `InitializeAD`, the Analog to Digital converter initialization routine, is then called. After the return from `InitializeAD`, the routine waits for `Tad` (A/D conversion clock) and then starts an A/D conversion. The program then enters an infinite loop with the reading of the A/D and the updating of the LEDs handled by an interrupt routine `Service_Int`.

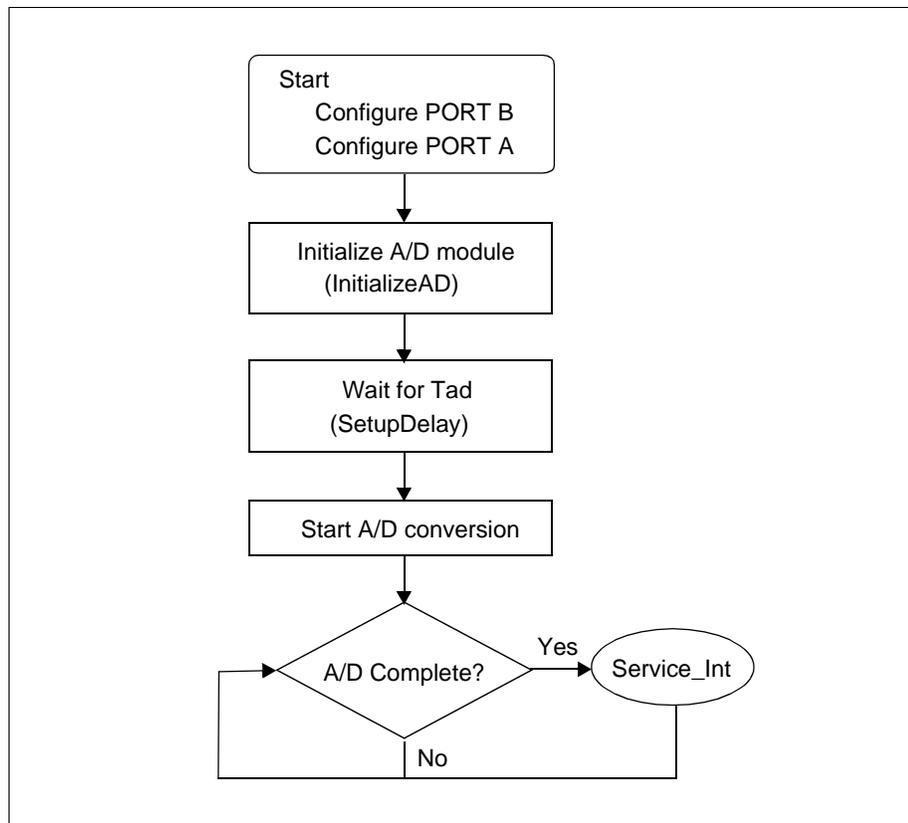


Figure 4.2: Main Routine

## 4.2 InitializeAD Routine

The `InitializeAD` routine sets up the A/D converter for use with the PICDEM-2 board. The routine sets RA0, RA1, and RA4 as analog inputs, enables the A/D interrupt, sets the A/D to use the internal RC oscillator, selects channel 0, clears the A/D interrupt flag, enables peripheral interrupts and then enables global interrupts.

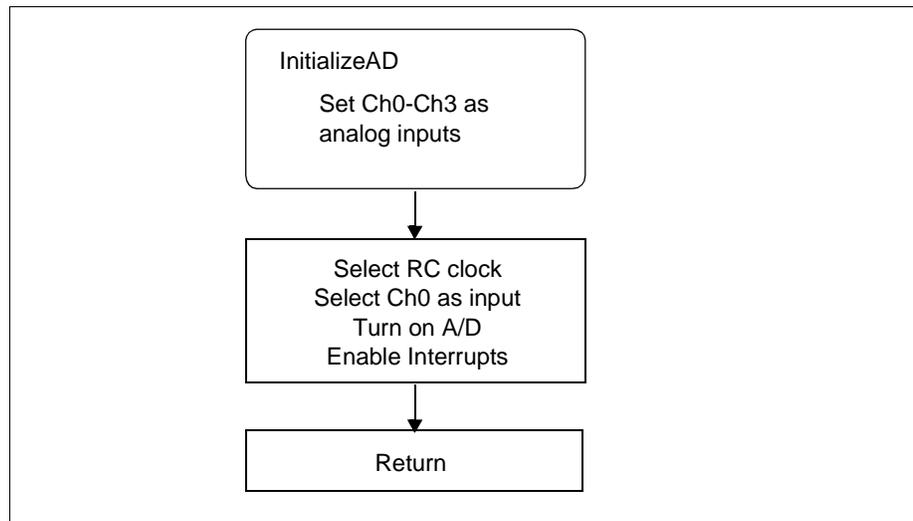


Figure 4.3: Initialize A/D

## 4.3 SetupDelay Routine

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

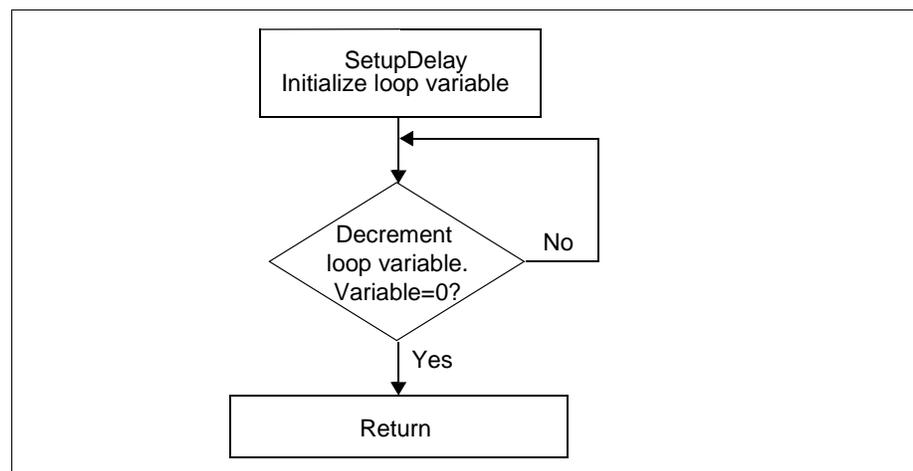


Figure 4.4: Setup Delay

# PICDEM-2 User's Guide

---

## 4.4 Service\_Int Routine

This subroutine handles the A/D converter interrupt and updating of the LEDs on PORTB. The first thing the routine does is to check if the A/D interrupt flag is set. If it is not, the interrupt was caused by some other peripheral, and the program returns to the main routine with interrupts re-enabled. If the interrupt was caused by the A/D converter, the A/D result register is read, and the value put on the LEDs. The A/D interrupt flag is then reset, and the peripheral interrupts are enabled. The routine then waits for the sample time ( $2 \cdot T_{AD}$ ), starts the next conversion, and returns with global interrupts enabled.

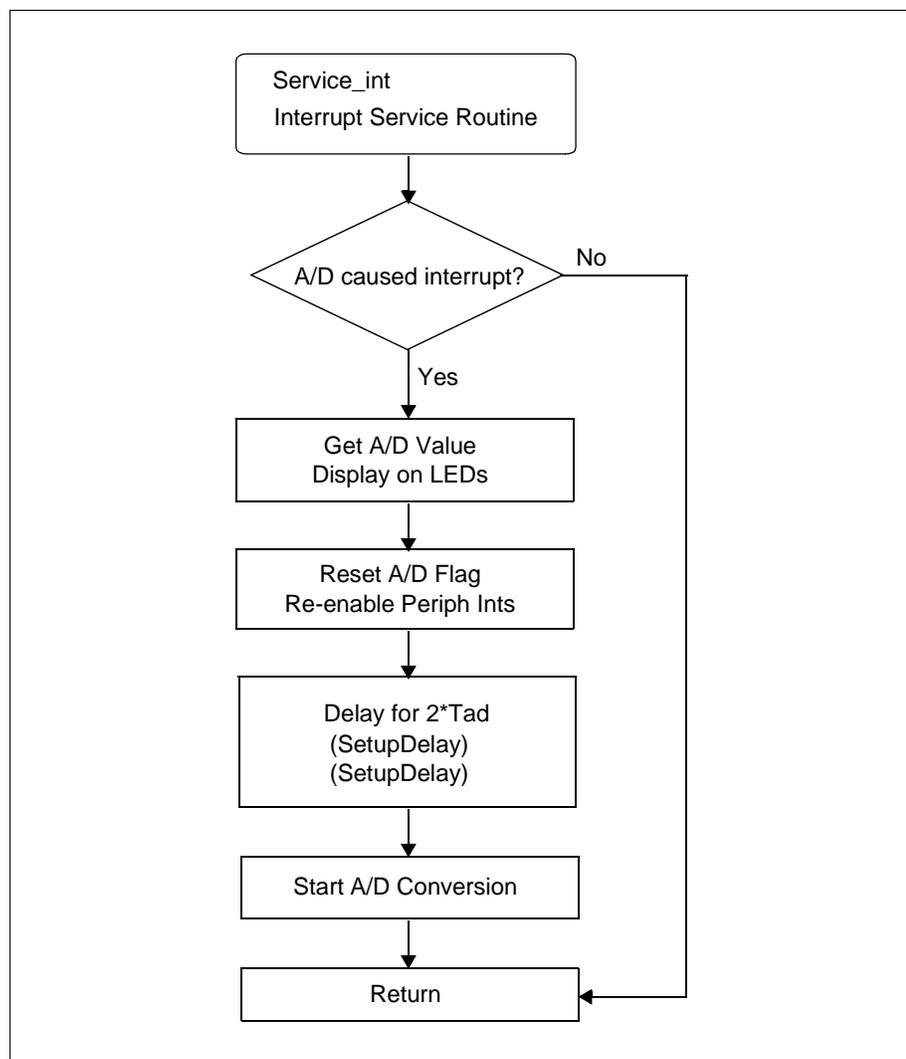


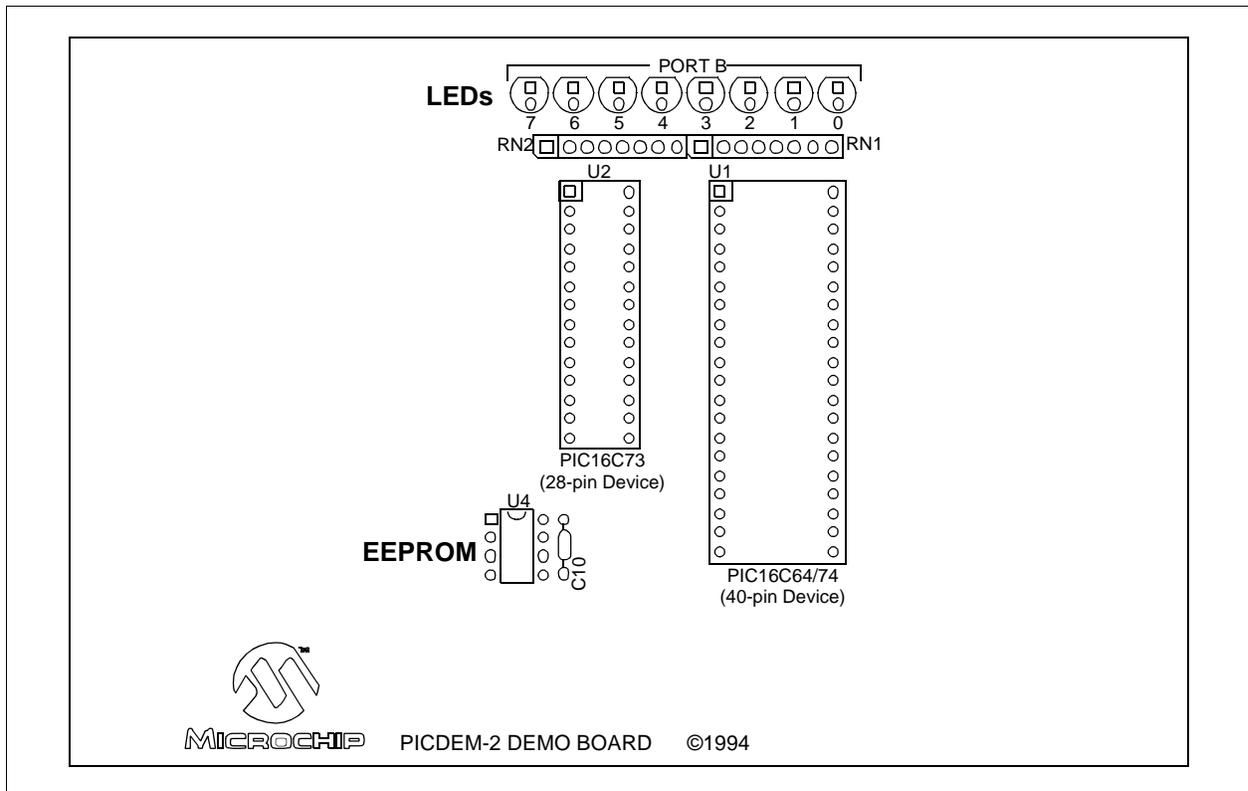
Figure 4.5: Service Interrupt

**Chapter 5. I<sup>2</sup>C/EEPROM Demo**

The `iic.asm` program demonstrates the use of the I<sup>2</sup>C bus. The program writes 0 through FFh to each address of the serial EEPROM, and then reads back the value. The LEDs indicate the data presently being written. If an error occurs, all of the LEDs flash.

For more information on I<sup>2</sup>C operation of the SSP module, please refer to the *PICmicro Midrange Microcontroller Family Reference Manual* for an operational description and a list of related application notes. For more information on serial EEPROM, please refer to the *Non-Volatile Memory Products Data Book*.

For detailed information on the PICDEM-2 hardware, please refer to Appendix A.



**Figure 5.1: I<sup>2</sup>C/EEPROM Demo**

# PICDEM-2 User's Guide

---

## 5.1 SSP Module

This section presents the flowcharts to implement a software I<sup>2</sup>C master mode in hardware. For MSSP operation, refer to Section 5.2.

### 5.1.1 Main Routine

The main routine is actually preceded by several register file, bit and vector assignments (Figure 5.2). The program begins by configuring PORTB as outputs and PORTC as inputs. The output data value will be read from PORTB, so it is initialized to one.

PORTC is cleared to set SCL and SCA low when not in tri-state. This is important as I<sup>2</sup>C is an open-drain bus. That is, only the zero values are actually driven to the bus. Ones are transmitted by making the pin an input (tristate) and allowing the pullup resistor to pull the line high. Data is transmitted in this manner by initializing the SCL and SDA port pins to zero and then writing the output value to the corresponding TRIS bit. A one written to the bit causes the pin to tristate pulled high and a zero causes it to become an output, driven low.

Interrupts are not used in this program, so they are disabled. Also, the SSP is configured to enable I<sup>2</sup>C master mode.

The write/read loop begins with the initialization of the EEPROM address to 00h, and the device address to A0h. Then the write loop executes by writing the output data value to each address in the EEPROM (Figure 5.3). After the write loop, the EEPROM address and device address are re-initialized for read. The read loop executes by reading data values for each EEPROM address and verifying their accuracy (Figure 5.4).

After the read loop, the output data value is incremented by one and this value displayed, via PORTB, on the LEDs. Then the write/read loop begins again.

If an error is encountered when writing or reading the EEPROM, or if an EEPROM value read is incorrect, the program jumps to the error loop, which flashes the LEDs (Figure 5.5).

# I<sup>2</sup>C/EEPROM Demo

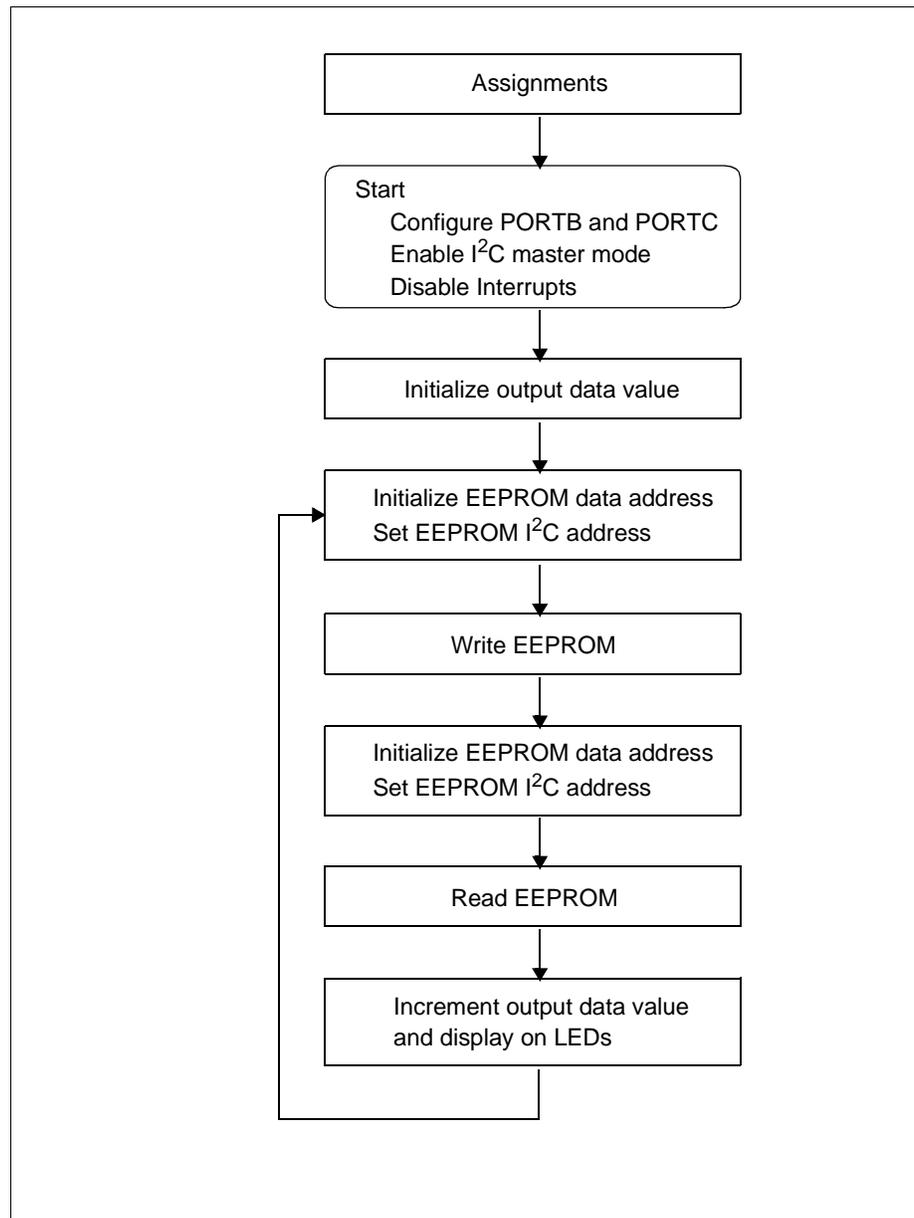
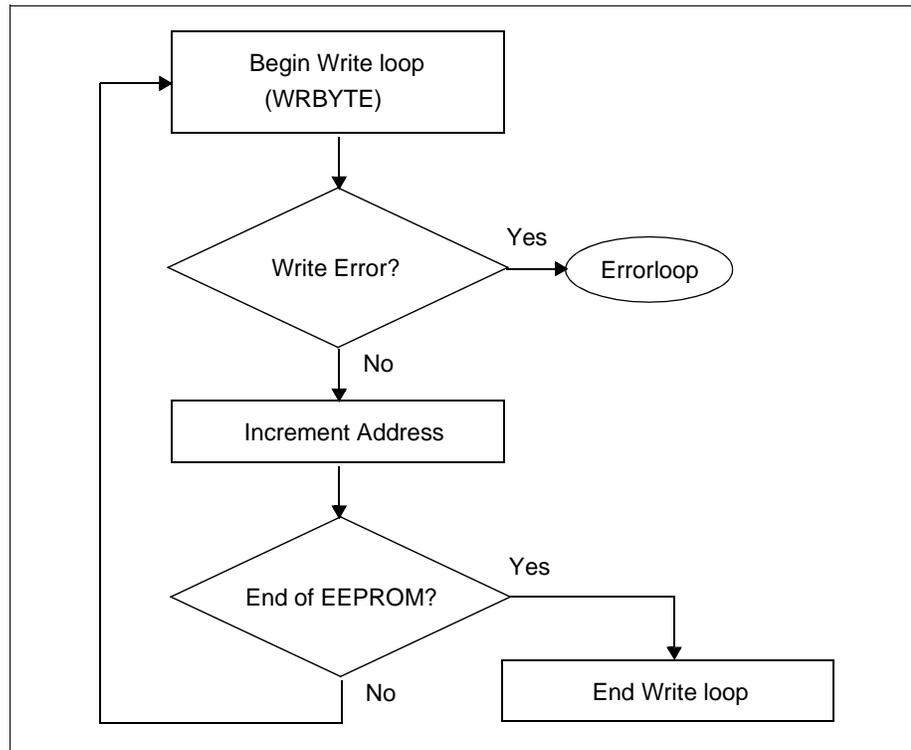


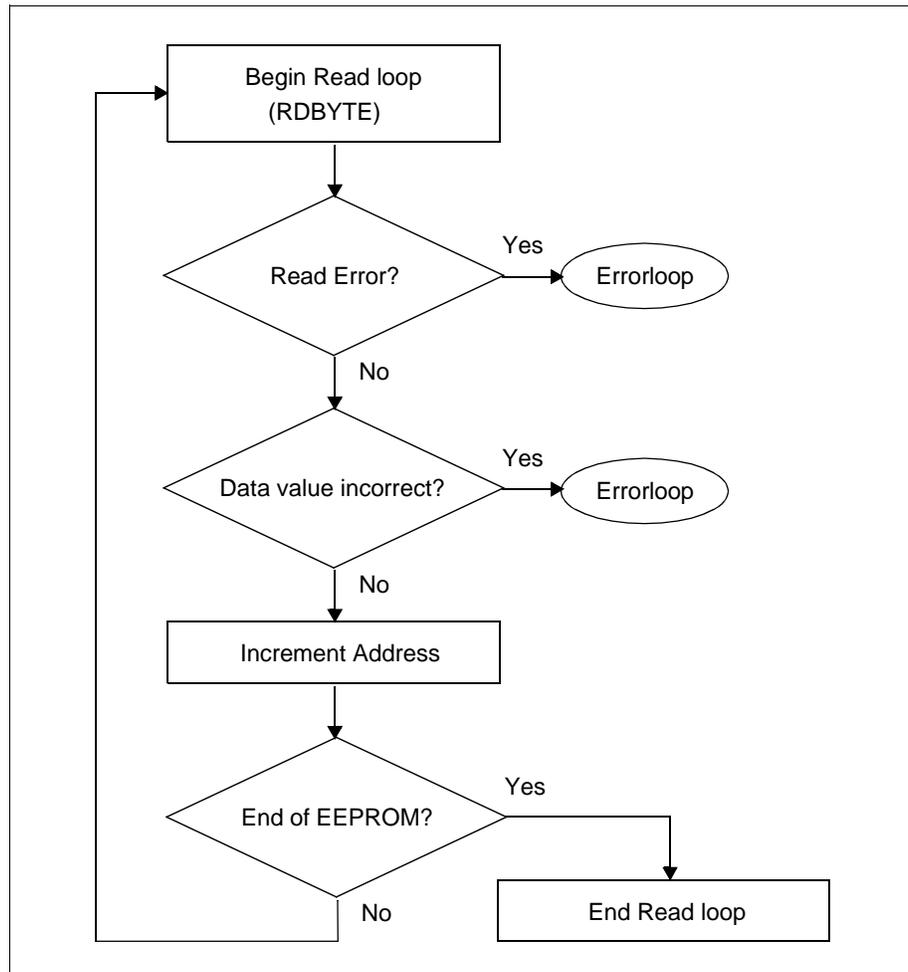
Figure 5.2: Main Routine

# PICDEM-2 User's Guide

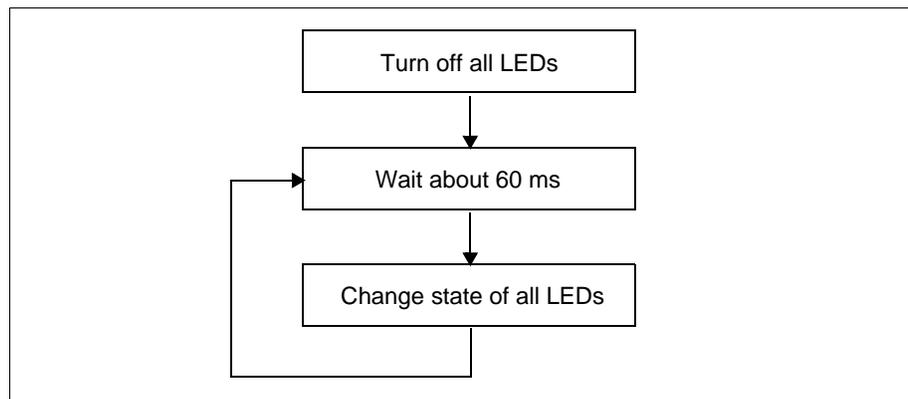
---



**Figure 5.3: Write EEPROM**



**Figure 5.4: Read EEPROM**



**Figure 5.5: Errorloop**

# PICDEM-2 User's Guide

---

## 5.1.2 WRBYTE and RDBYTE Routines

The `WRBYTE` and `RDBYTE` routines write and read one byte to the serial EEPROM device, respectively. Both routines begin by generating a start bit and then transmitting the slave address and word address. To write to the EEPROM, data followed by a stop bit is transmitted. To read the EEPROM, another start bit and the slave address is transmitted, followed by a data read and transmission of a stop bit. The received data is stored in a file register.

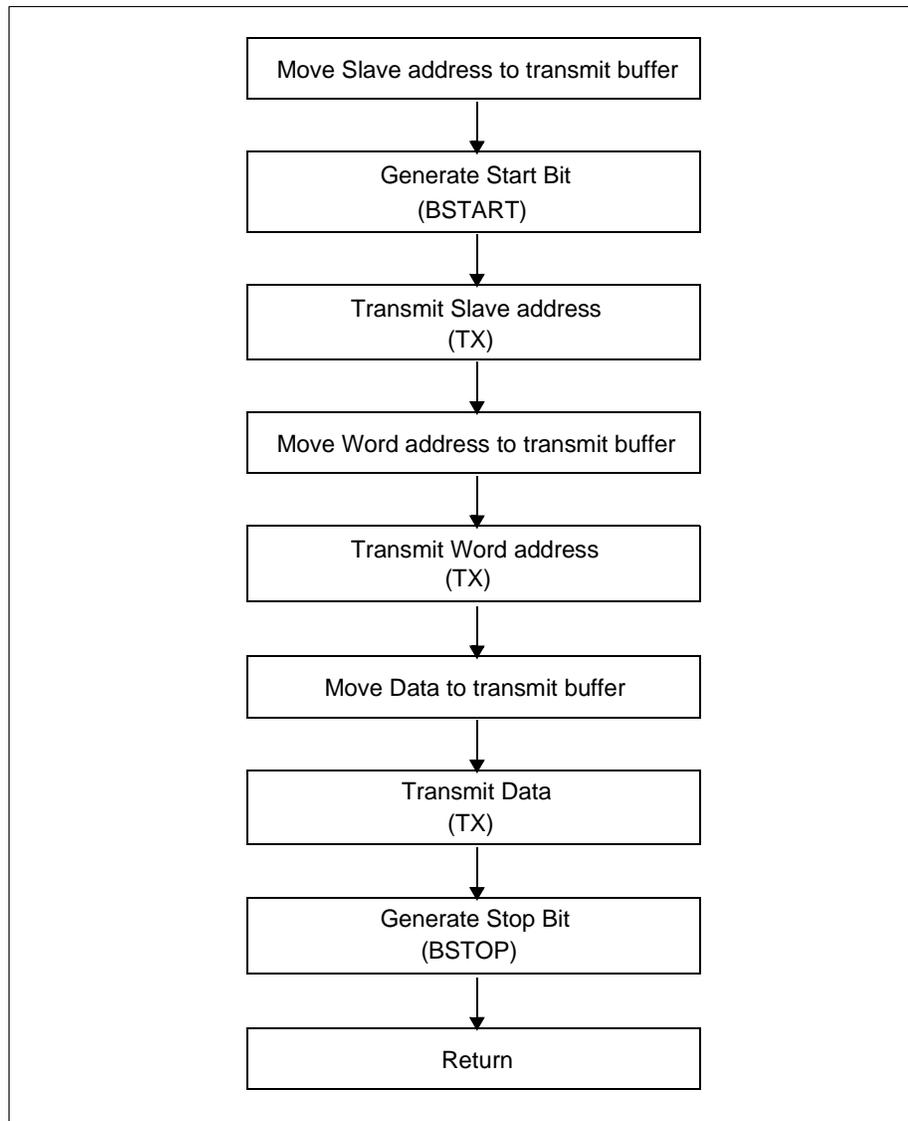
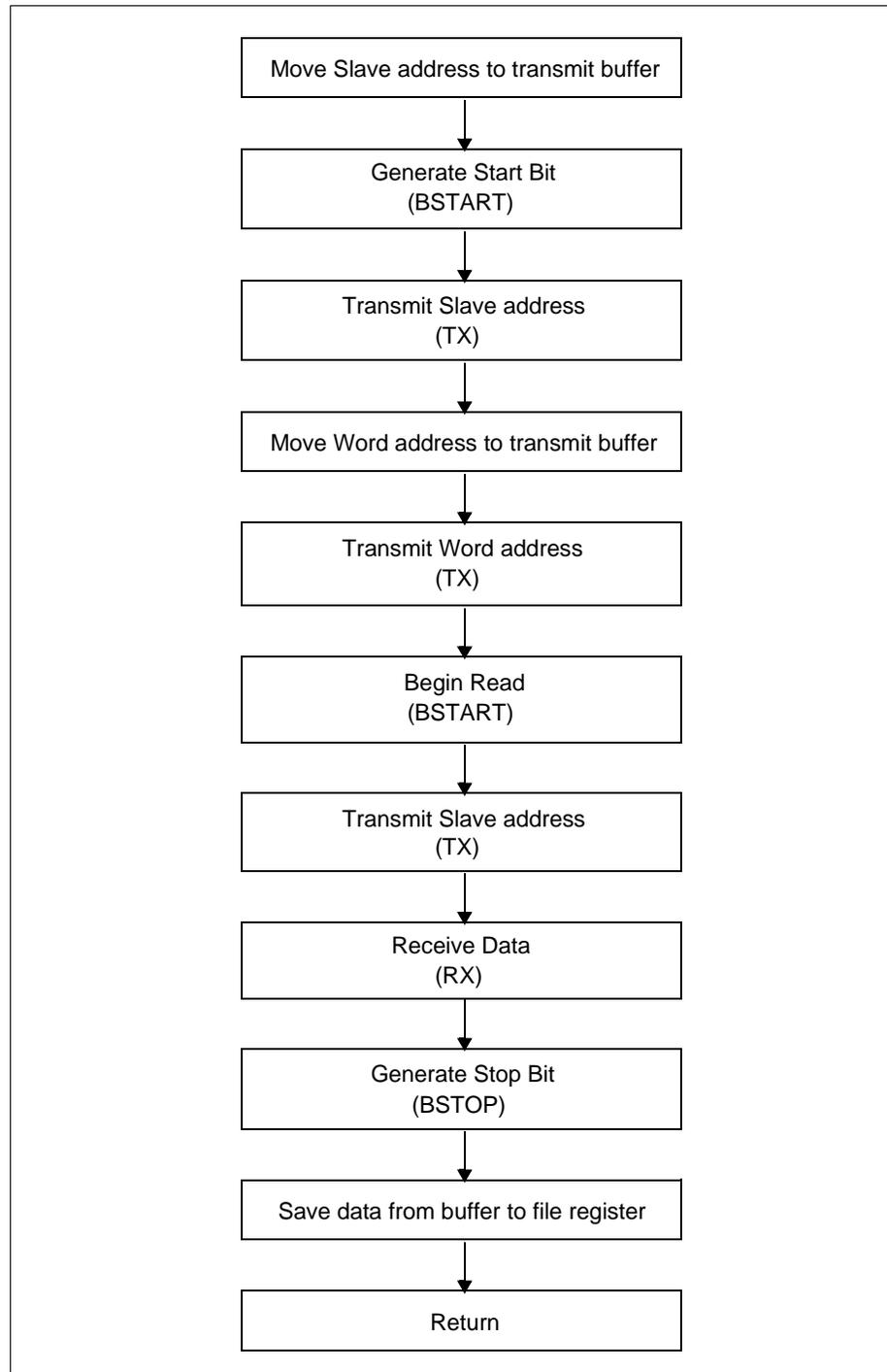


Figure 5.6: Write Byte



**Figure 5.7: Read Byte**

# PICDEM-2 User's Guide

## 5.1.3 TX and RX Routines

The TX and RX routines transmit and receive 8 data bits, respectively. Data is transmitted/received one bit at a time by left bit-shifting the transmit/receive buffer. A counter, set initially to 8, is decremented until all 8 bits of data are sent/received. Finally, acknowledgment bits are received/transmitted. In the case of transmission, if an acknowledgment is not received, an error flag is set.

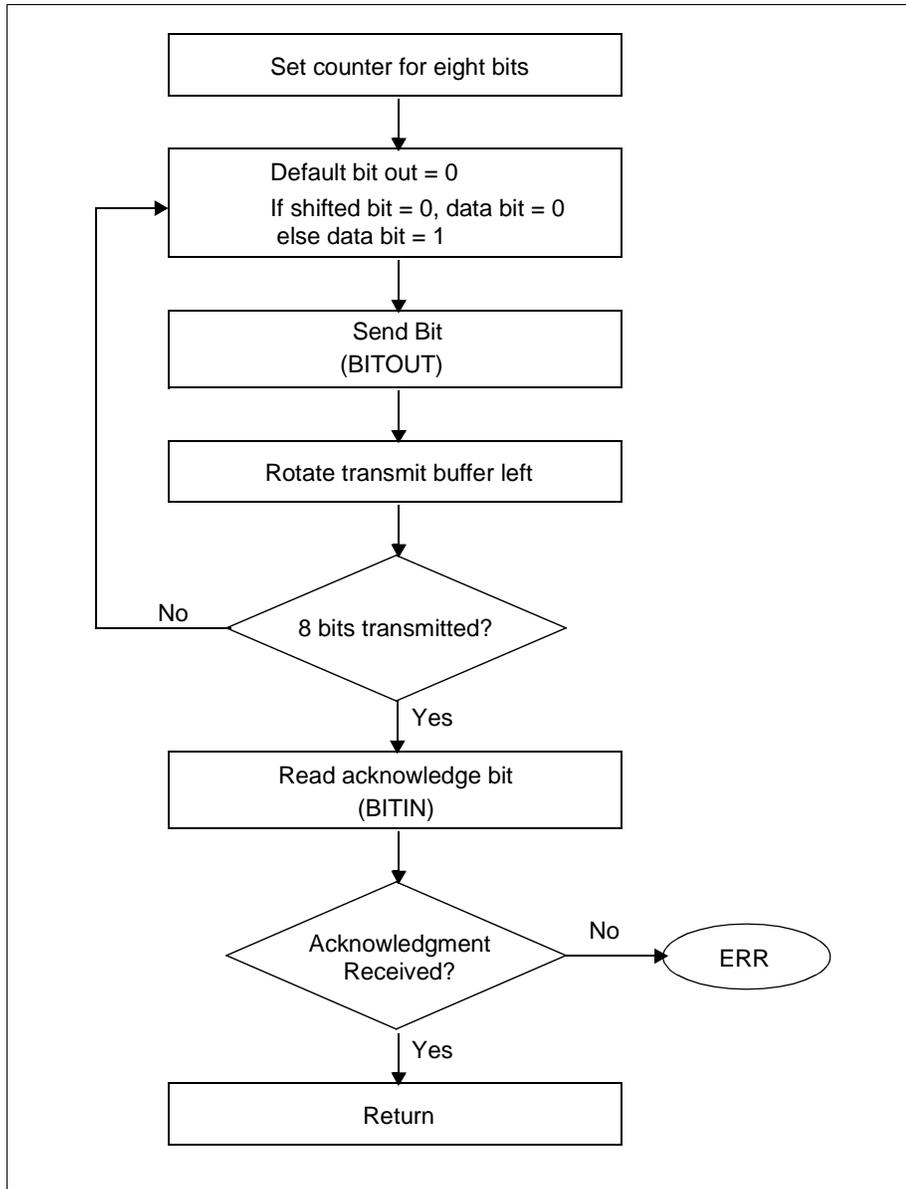
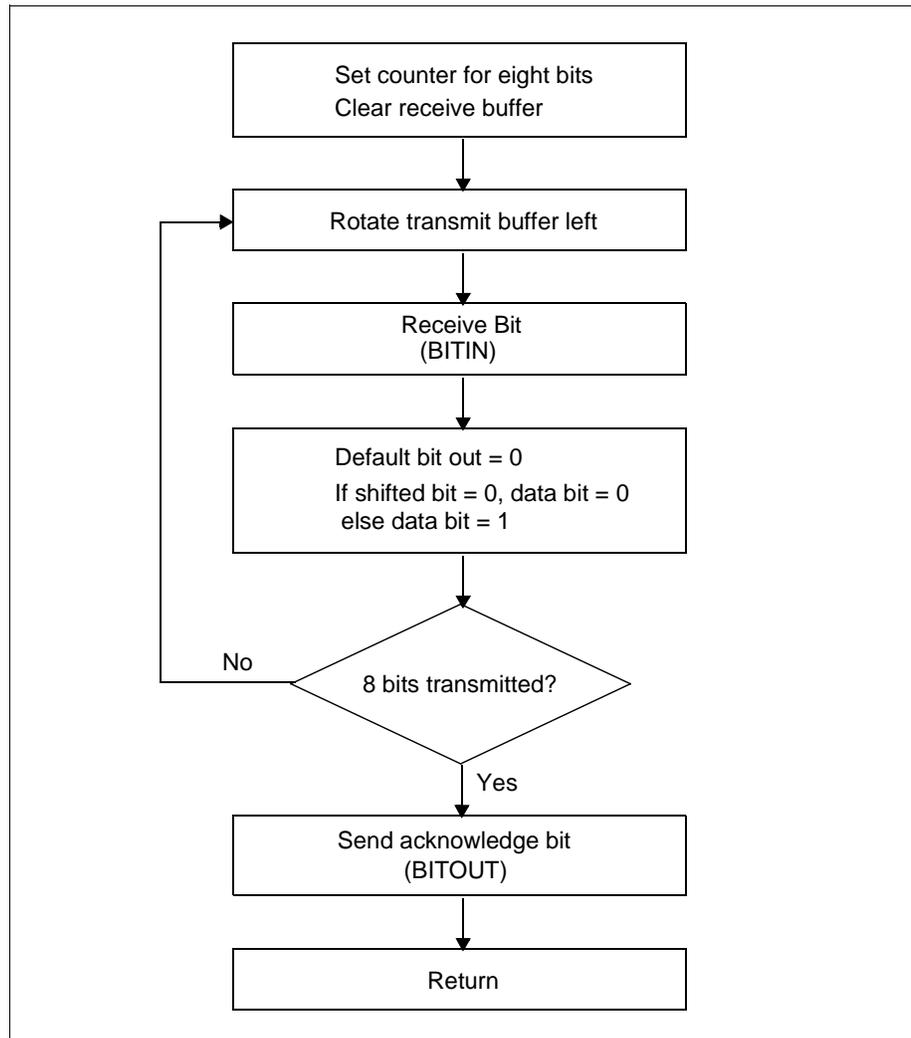


Figure 5.8: Transmit



**Figure 5.9: Receive**

# PICDEM-2 User's Guide

---

## 5.1.4 BITOUT and BITIN Routines

The `BITOUT` routine performs single-bit data transmission from the PICmicro to the serial EEPROM. The `BITIN` routine performs single-bit data reception from the serial EEPROM to the PICmicro.

## 5.1.5 BSTART and BSTOP Routines

The `BSTART` and `BSTOP` routines generate the start and stop bits, respectively.

## 5.1.6 Delay and Err Routines

The `Delay` routine provides a 10.78 ms delay. The `Err` routine identifies errors.

## 5.2 MSSP Module

This section presents flowcharts to implement communication with Serial EEPROM using the MSSP module in master mode. For SSP operation, refer to Section 5.1.

### 5.2.1 Main Routine

The main routine is actually preceded by several register file, bit and vector assignments (Figure 5.2). The program begins by configuring PORTB as outputs and PORTC as inputs. The output data value will be read from PORTB, so it is initialized to one.

The next part of the code initializes the MSSP module for master I<sup>2</sup>C mode. The slew rate control is disabled and the clock rate is set to 100 kHz. Both the RC3 and RC4 pins are made inputs as the MSSP module will control the data direction.

Interrupts are not used in this program, so they are disabled. Also, the MSSP is configured to enable I<sup>2</sup>C master mode.

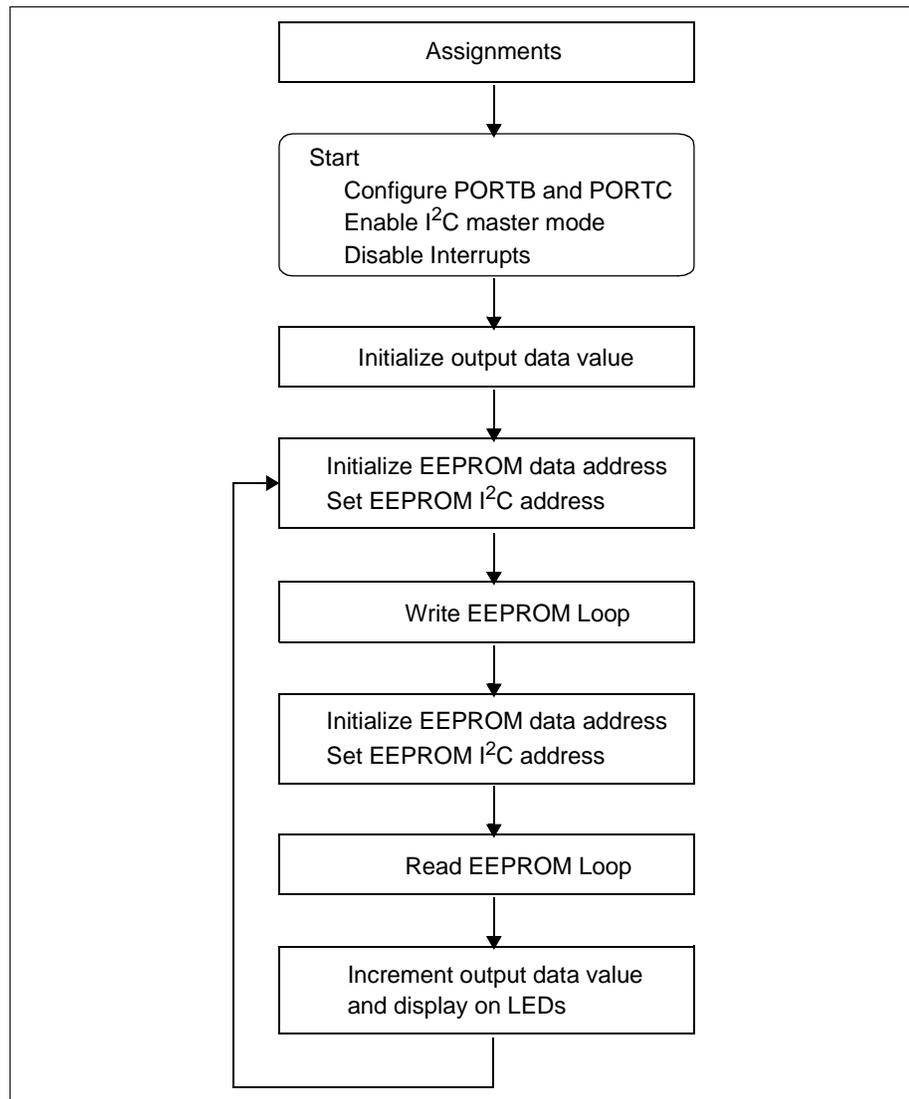
The write/read loop begins with the initialization of the EEPROM address to 00h, and the device address to A0h. Then the write loop executes by writing the output data value to each address in the EEPROM (Figure 5.3). After the write loop, the EEPROM address and device address are re-initialized for read. The read loop executes by reading data values for each EEPROM address and verifying their accuracy (Figure 5.4).

After the read loop, the output data value is incremented by one and this value displayed, via PORTB, on the LEDs. Then the write/read loop begins again.

If an error is encountered when writing or reading the EEPROM, or if an EEPROM value read is incorrect, the program jumps to the error loop, which flashes the LEDs (Figure 5.5).

# PICDEM-2 User's Guide

---



**Figure 5.10: Main Routine**

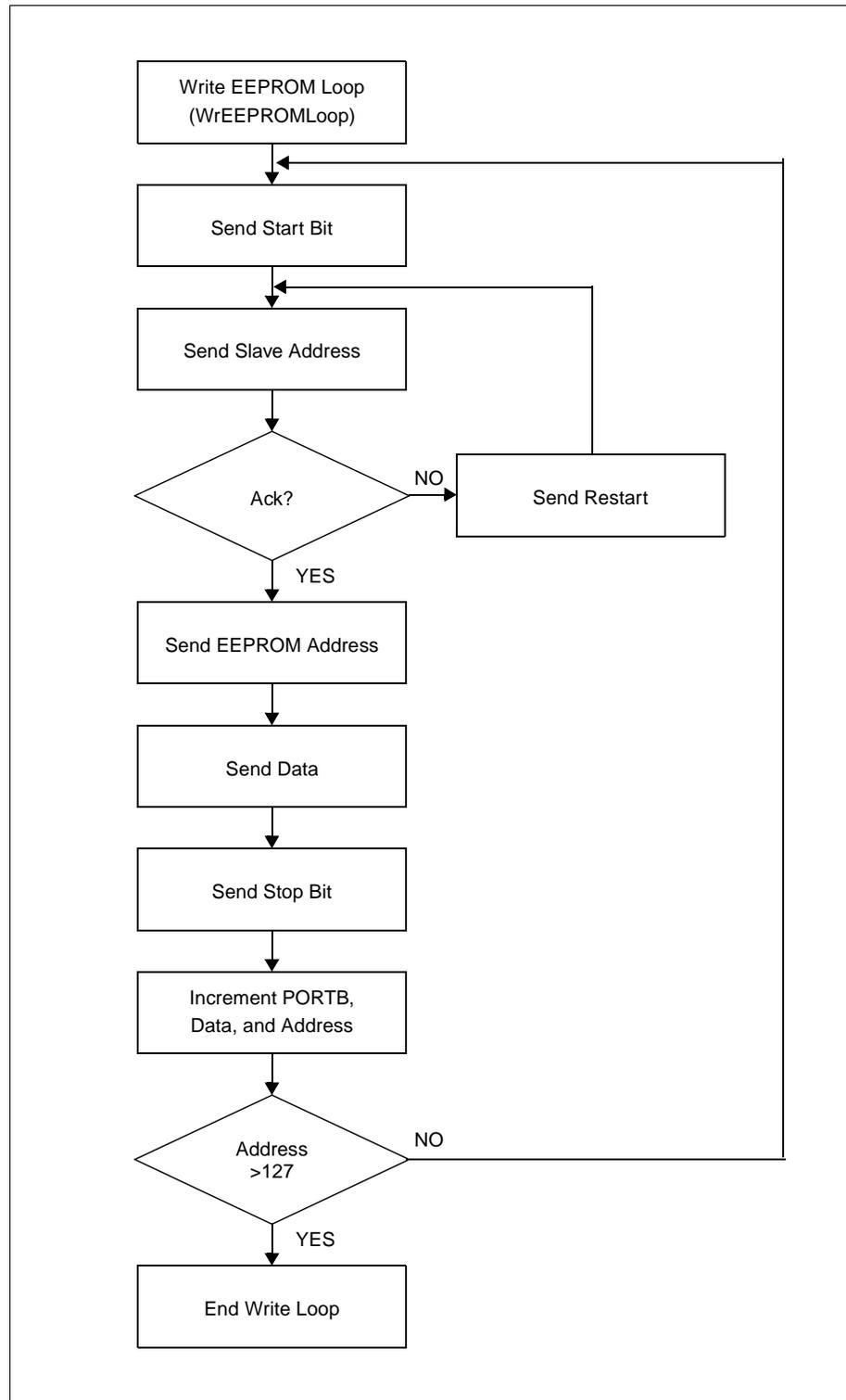


Figure 5.11: Write EEPROM Loop

# PICDEM-2 User's Guide

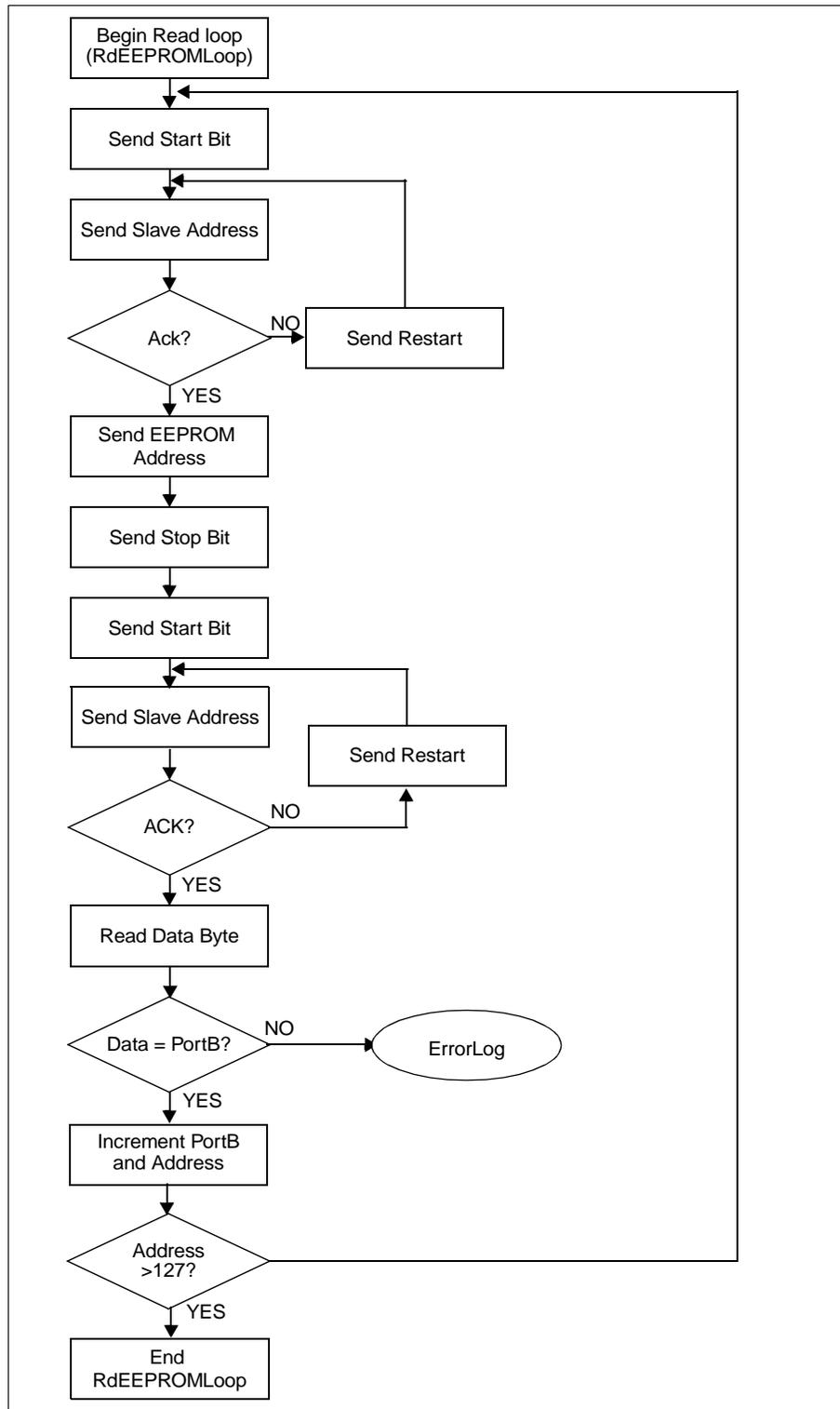
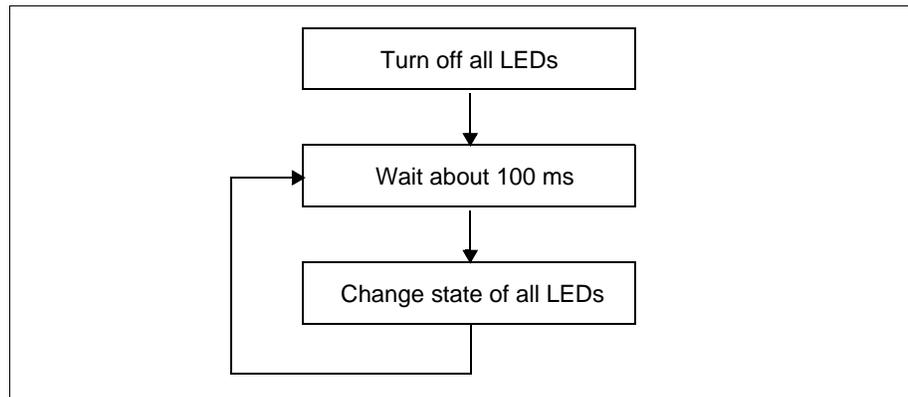


Figure 5.12: Read EEPROM Loop



**Figure 5.13: Errorloop**

# PICDEM-2 User's Guide

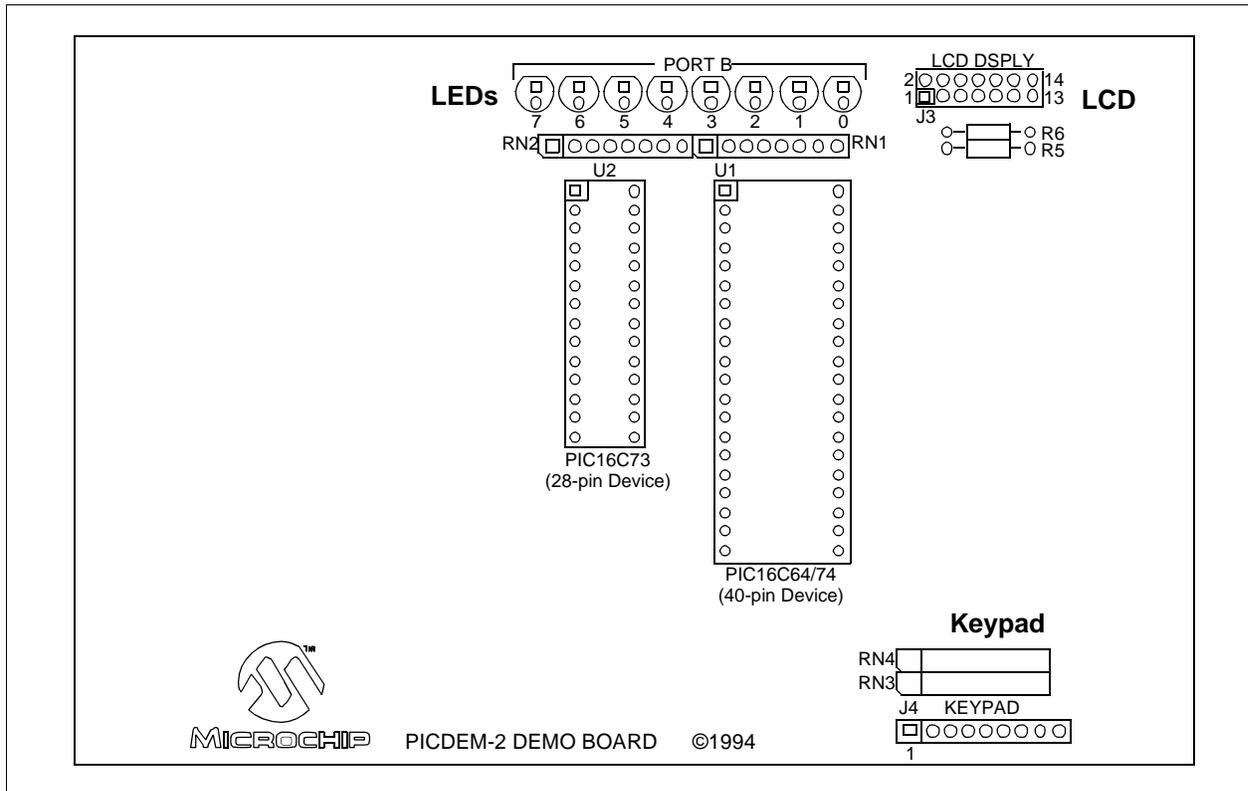
---

NOTES:

**Chapter 6. LCD Demo**

The `kbd4bit.asm` and `kbd8bit.asm` programs demonstrate how to interface to an LCD module in both the 4-bit and 8-bit interface modes. The programs read the keypad and output the keypresses to the LEDs and the LCD module. Two include files on the disk, `lcd4bit.asm` and `lcd8bit.asm`, have all of the routines necessary to run the display.

For more information on LCD operation, please refer to *Interfacing PICmicro Microcontrollers to an LCD Module - AN587*. For detailed information on the PICDEM-2 hardware, please refer to Appendix A.



**Figure 6.1: LCD Demo**

# PICDEM-2 User's Guide

Any 3x4 or 4x4 keypad may be used, such as C&K part number 4B01H322PCFQ available from Newark Electronics. Figure 10-1 shows typical connections for the keypad.

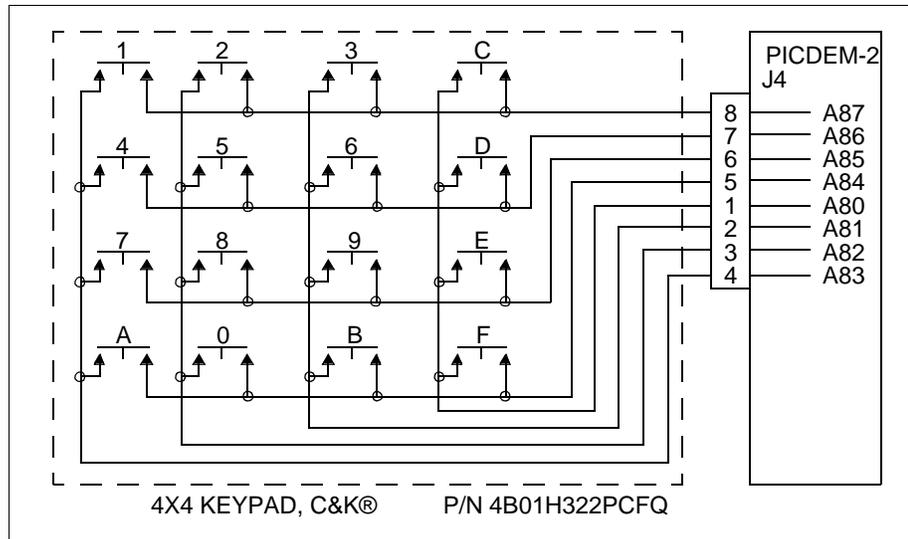
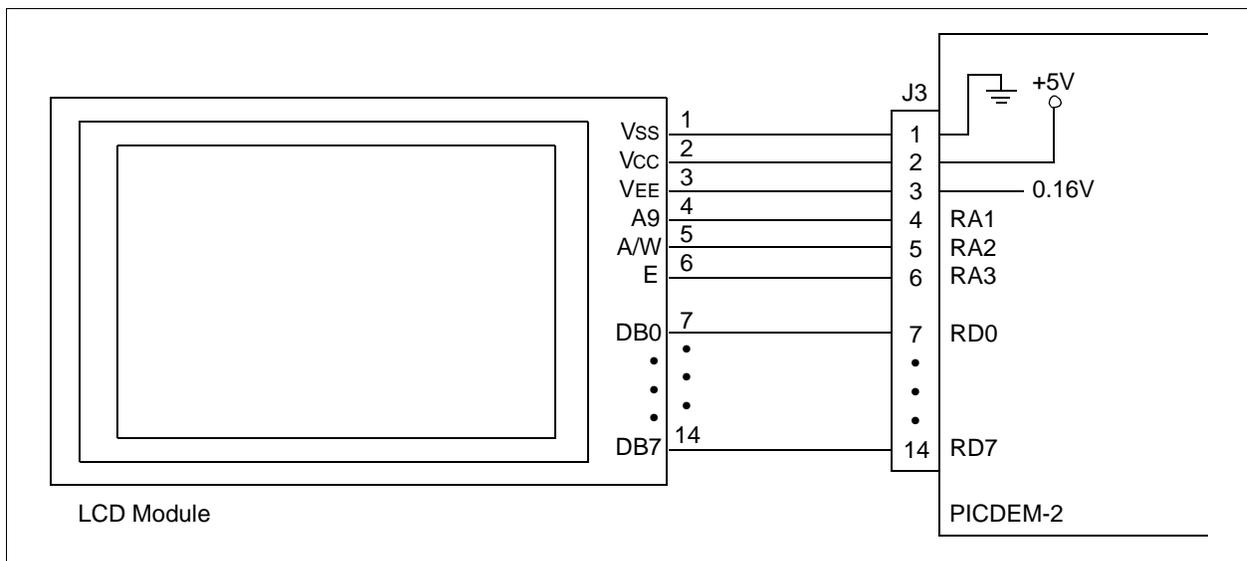


Figure 6.2: Keypad Connections

Many different LCD modules may be used. One is the 16-character-by-2-line Optrex DMC-16207N available from Digi-Key. Figure 10-2 shows the connections to the LCD module. Table 10-2 shows the signal connections for the LCD connector on the PICDEM-2 board.

**Table: 6.1 LCD Connection Signals**

PICDEM-2 Pin No.	Symbol	Signal
1	V <sub>ss</sub>	Ground
2	V <sub>cc</sub>	+5V
3	V <sub>EE</sub>	LCD Contrast
4	RS	Register Select
5	R/W	Read/Write
6	E	Enable
7	DB0	Data bit 0
•	•	
•	•	
14	DB7	Data bit 7



**Figure 6.3: LCD Connections**

# PICDEM-2 User's Guide

---

## 6.1 4-Bit Interface

### 6.1.1. Main Routine (kbd4bit.asm)

The program initializes the ports, keypress variables and LCD, then goes into an infinite loop where it looks for keypresses from the keyboard. When it receives one, it relays the ASCII value of the key to the LCD routine for display, then returns to look for another keypress.

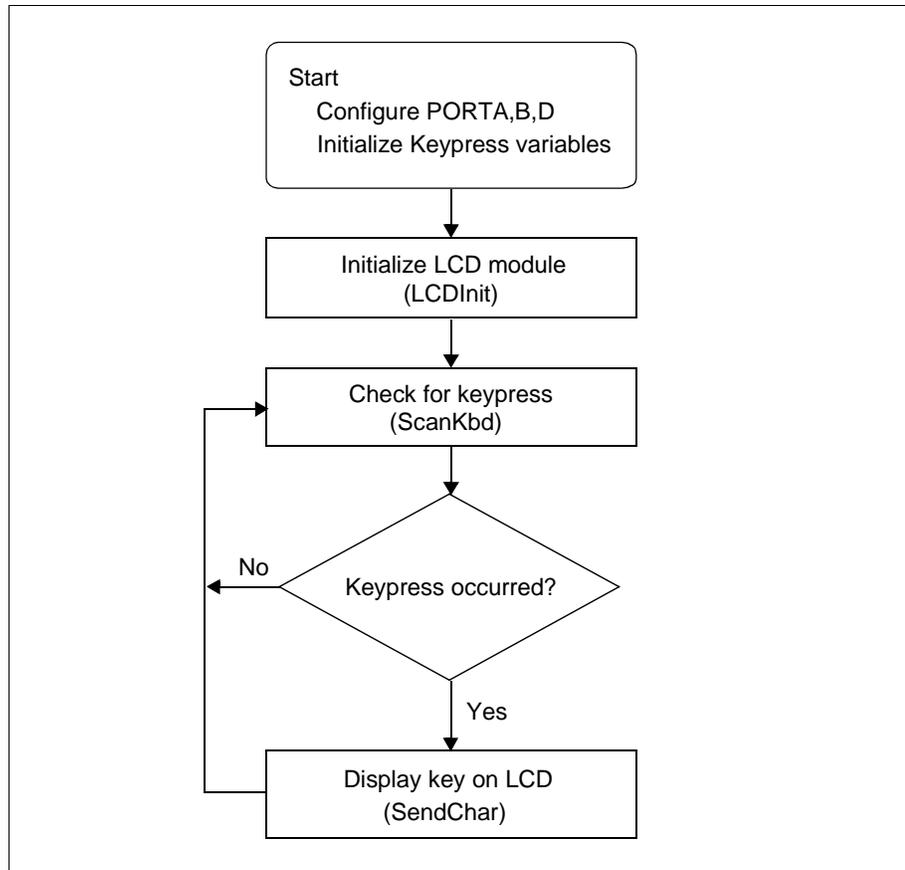


Figure 6.4: Main Routine

6.1.2. ScanKbd Routine (kbd4bit.asm)

This routine scans the keyboard for a keypress. When it detects one, it returns the ASCII value of the key.

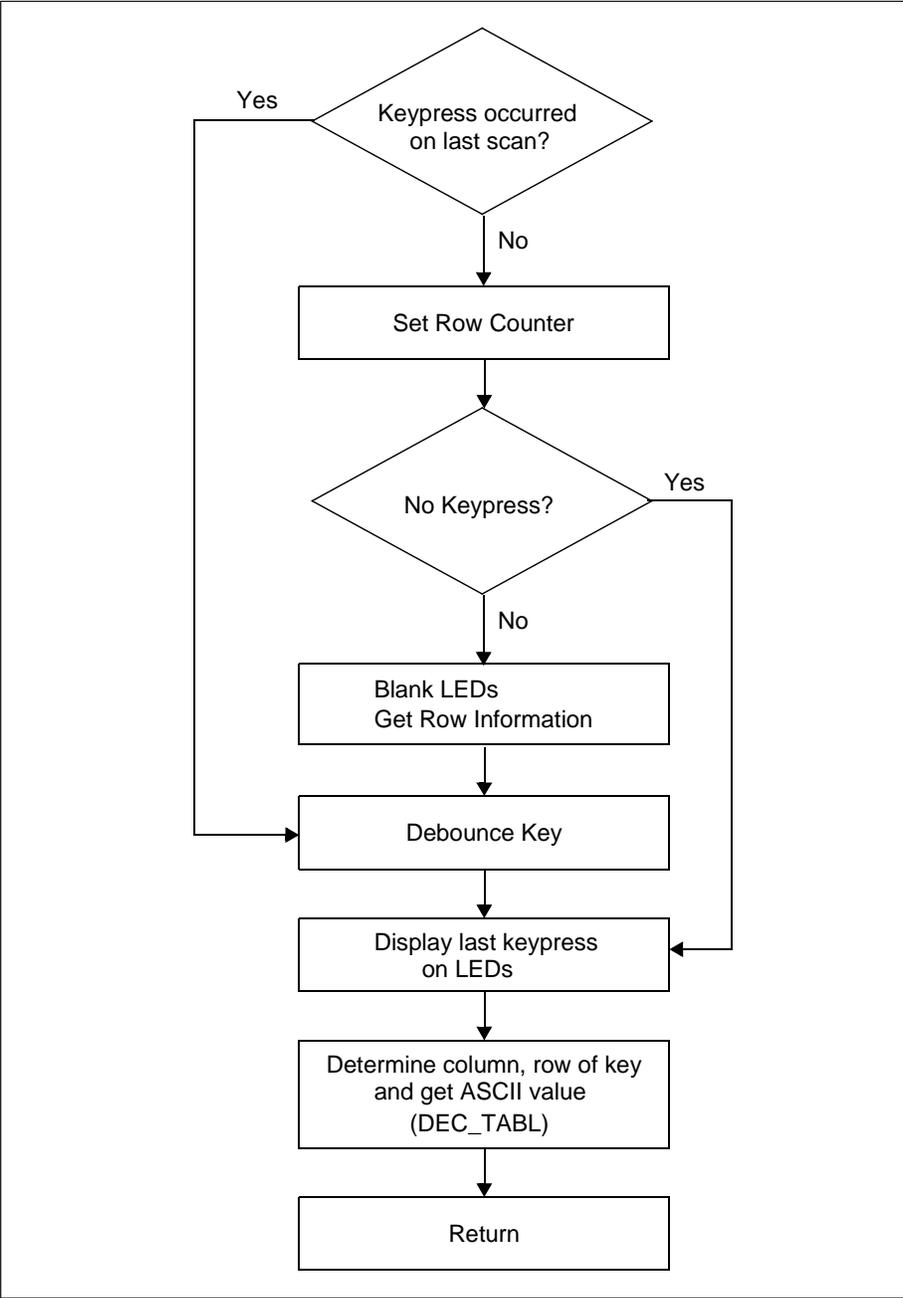


Figure 6.5: Keyboard Scan

# PICDEM-2 User's Guide

---

## 6.1.3. DEC\_TABL Routine (kbd4bit.asm)

This routine is basically a look-up table that returns the ASCII value of the received keypress.

## 6.1.4. LCDInit Routine (lcd4bit.asm)

This routine initializes the LCD module.

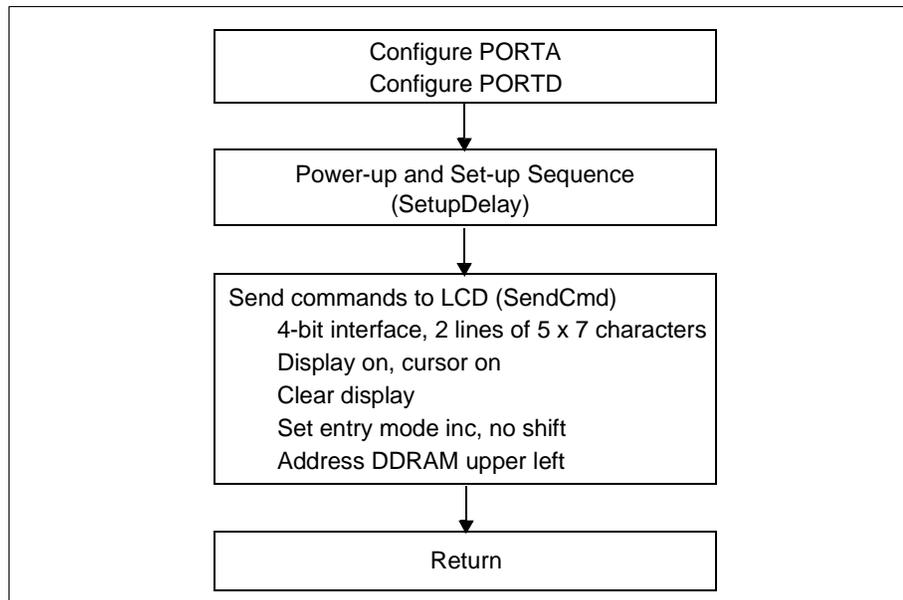


Figure 6.6: LCD Initialization Routine

## 6.1.5. SetupDelay Routine (lcd4bit.asm)

This routine is basically a decrementing variable loop. The variable is initialized with a value and then decremented in a loop until it reaches zero value. Then it jumps out of the loop and returns to the calling routine.

## 6.1.6. SendCmd Routine (lcd4bit.asm)

This routine sends a command to the LCD. It splits the command into upper and lower nibbles (4 bits) and sends these to the LCD, upper nibble first.

## 6.1.7. BusyCheck Routine (lcd4bit.asm)

This routine checks the LCD busy flag. It returns only when not busy.

## 6.1.8. SendChar Routine (lcd4bit.asm)

This routine sends a character to the LCD. It splits the character into upper and lower nibbles (4 bits) and sends these to the LCD, upper nibble first.

## 6.2 8-Bit Interface

### 6.2.1. Main Routine (kbd8bit.asm)

The program initializes the ports, keypress variables and LCD, then goes into an infinite loop where it looks for keypresses from the keyboard. When it receives one, it relays the ASCII value of the key to the LCD routine for display, then returns to look for another keypress.

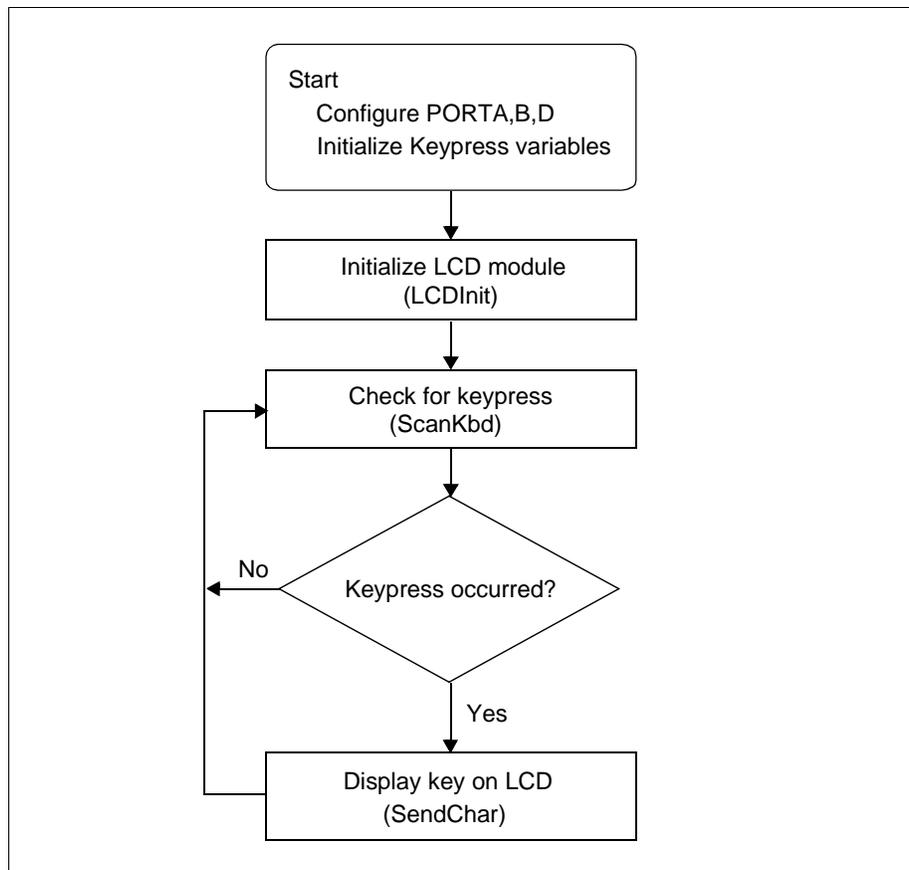


Figure 6.7: Main Routine

## 6.2.2. ScanKbd Routine (kbd8bit.asm)

This routine scans the keyboard for a keypress. When it detects one, it returns the ASCII value of the key.

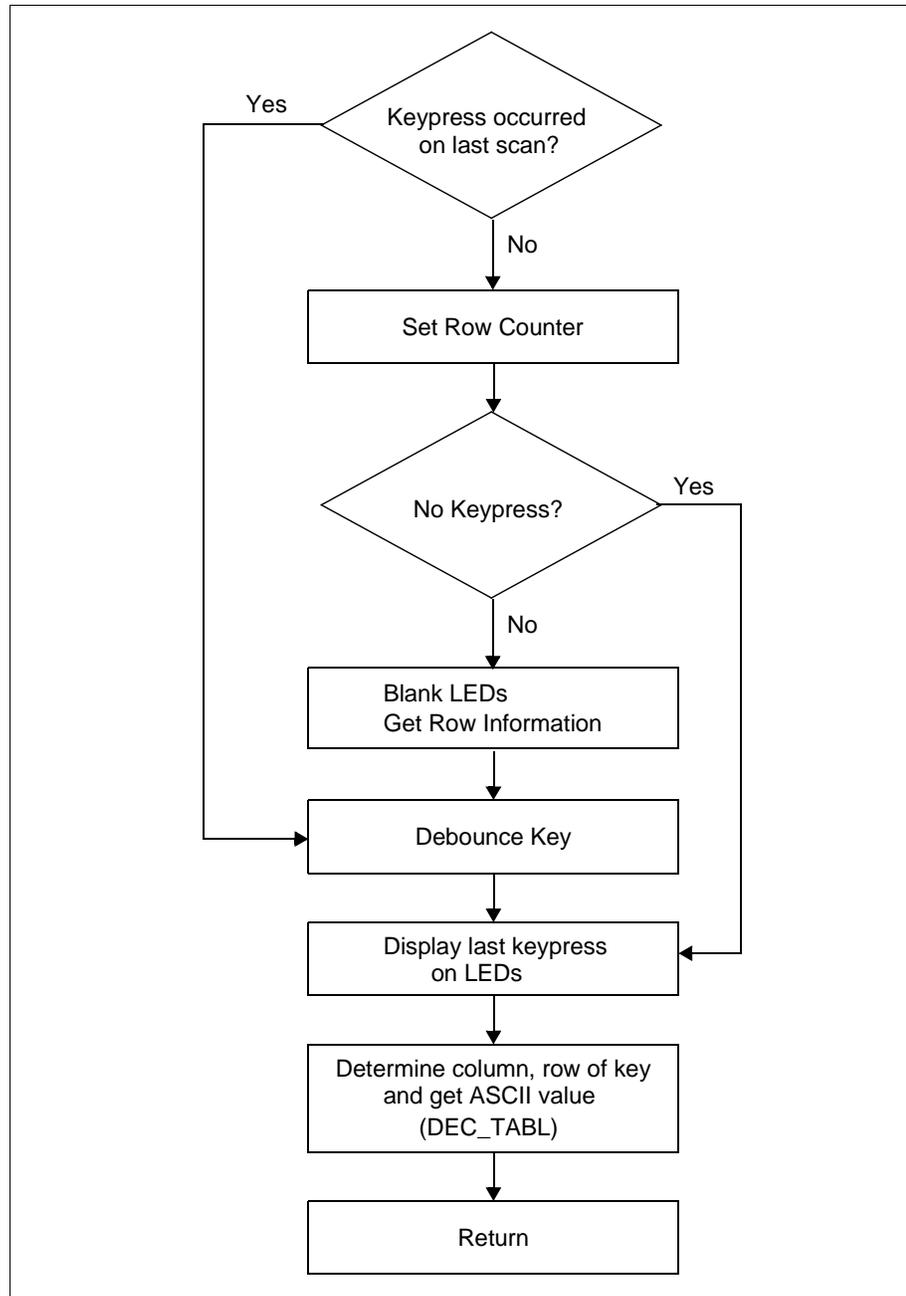


Figure 6.8: Keyboard Scan

## 6.2.3. DEC\_TABL Routine (kbd8bit.asm)

This routine is basically a look-up table that returns the ASCII value of the received keypress.

## 6.2.4. LCDInit Routine (lcd8bit.asm)

This routine initializes the LCD module.

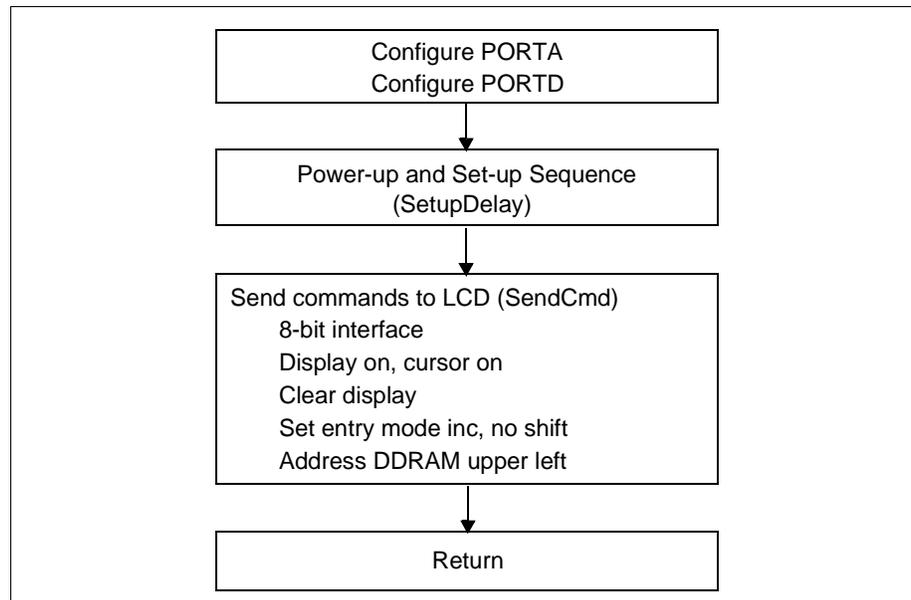


Figure 6.9: LCD Initialization Routine

## 6.2.5. SetupDelay Routine (lcd8bit.asm)

This routine is basically a decrementing variable loop. The variable is initialized with a value and then decremented in a loop until it reaches zero value. Then it jumps out of the loop and returns to the calling routine.

## 6.2.6. SendCmd Routine (lcd8bit.asm)

This routine sends a command to the LCD.

## 6.2.7. BusyCheck Routine (lcd8bit.asm)

This routine checks the LCD busy flag. It returns only when not busy.

## 6.2.8. SendChar Routine (lcd8bit.asm)

This routine sends a character to the LCD.

# PICDEM-2 User's Guide

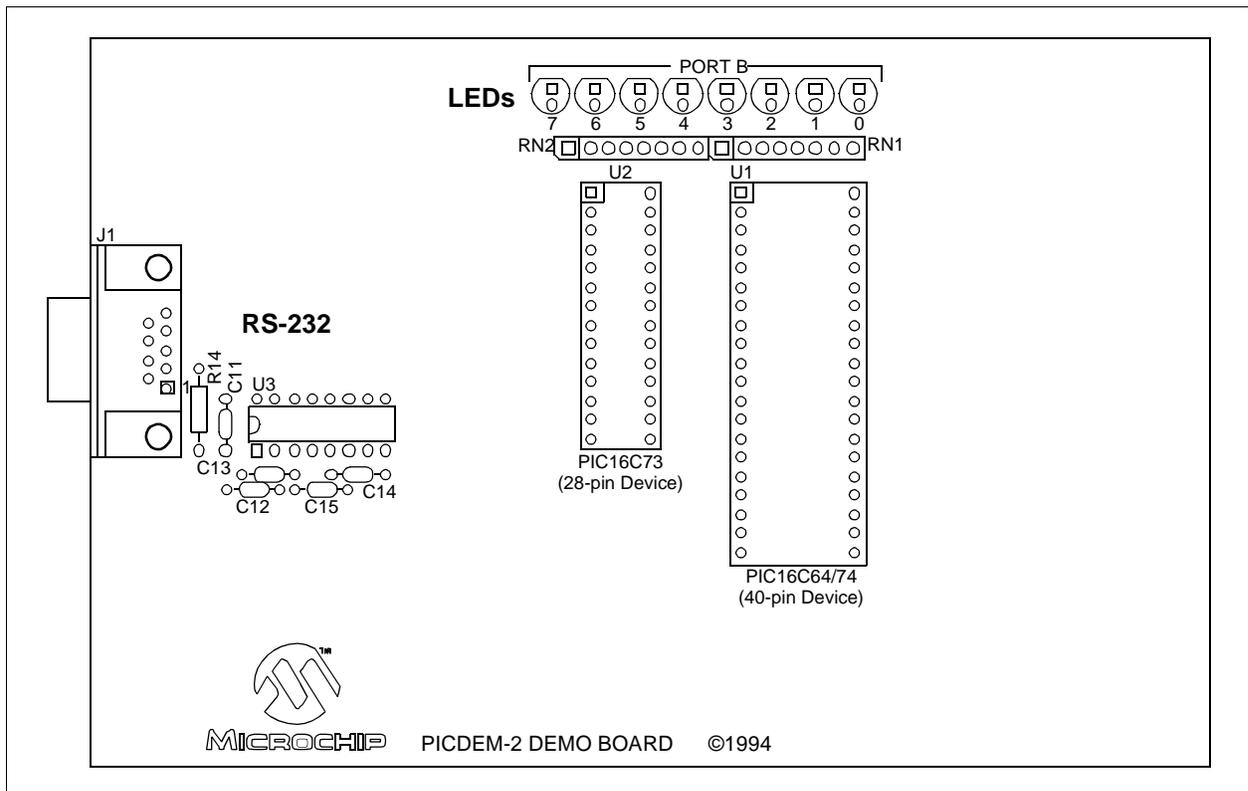
---

NOTES:

**Chapter 7. USART Demo**

The `usart.asm` program demonstrates the USART peripheral. The program echoes the values received through the serial port back to the host computer. Values are also displayed on the LEDs. The program operates at 9600 baud, 1 stop bit, and no parity.

For more information on USART 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-2 hardware, please refer to Appendix A.



**Figure 7.1: USART Demo**

# PICDEM-2 User's Guide

---

## 7.1 Main Routine

The main routine sets up PORTB for LED display and configures the USART module. When data is received over the RS-232 communication line, an interrupt is generated and the program jumps to the `IntVector` routine.

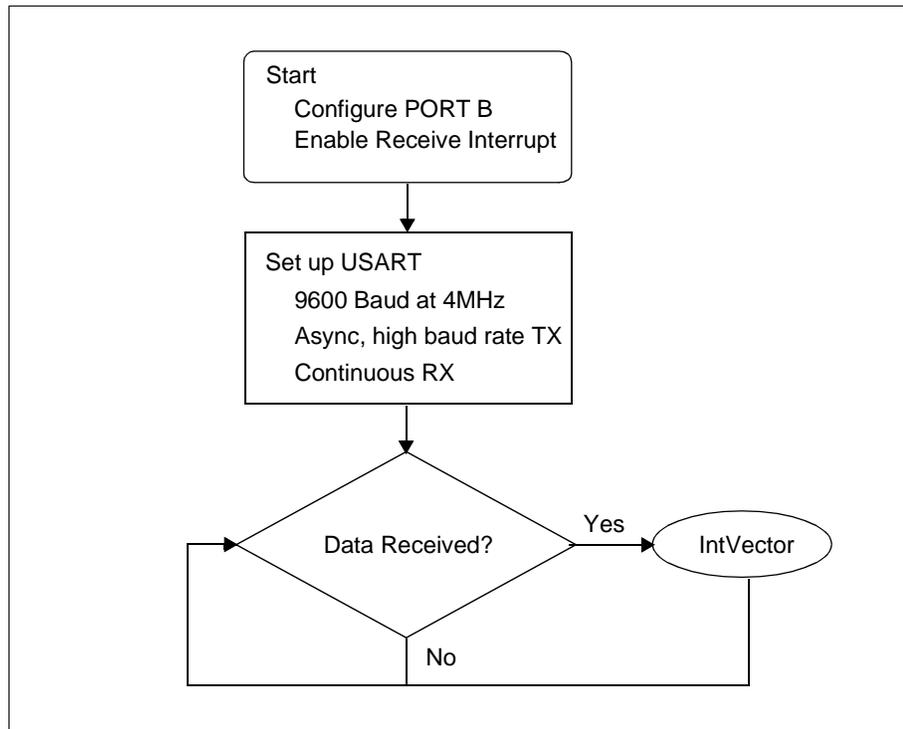


Figure 7.2: Main Routine

## 7.2 IntVector Routine

The interrupt service routine reads the transmitted data, displays the value on the LEDs, and transmits the same data back to the sender. This routine also checks for receive errors, and jumps to the `RcvError` routine if any are found. It also determines if the interrupt is other than a receive interrupt, and returns to the main program if it is.

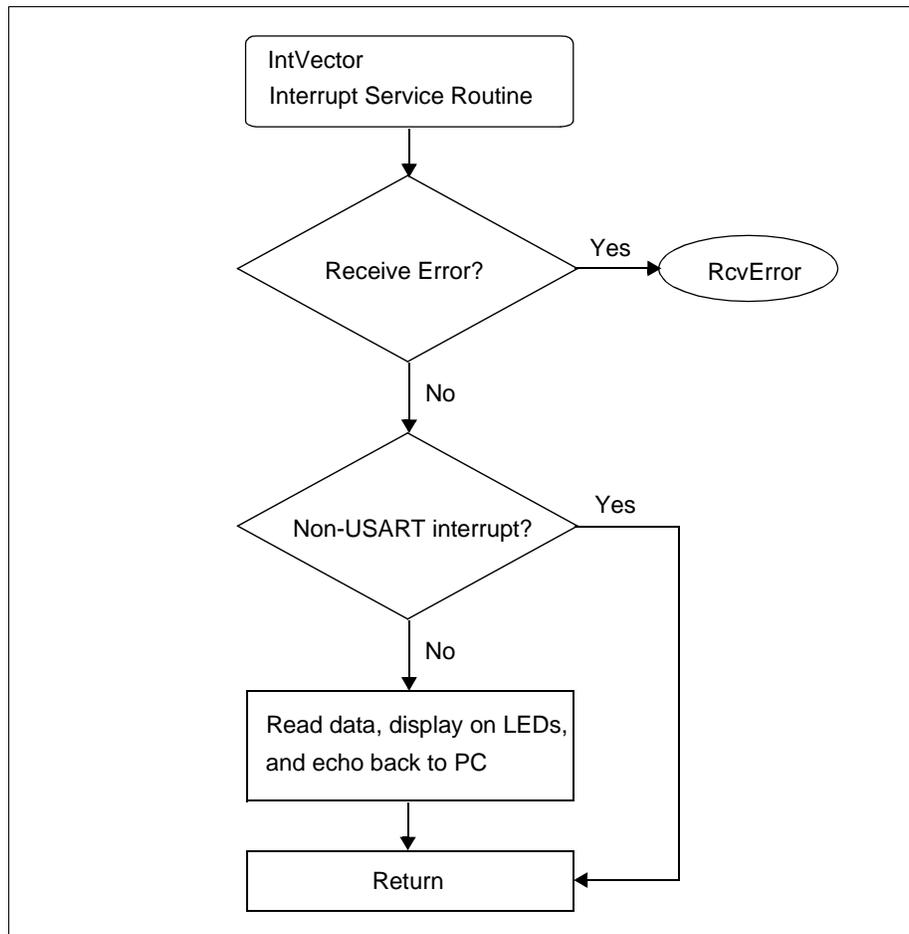
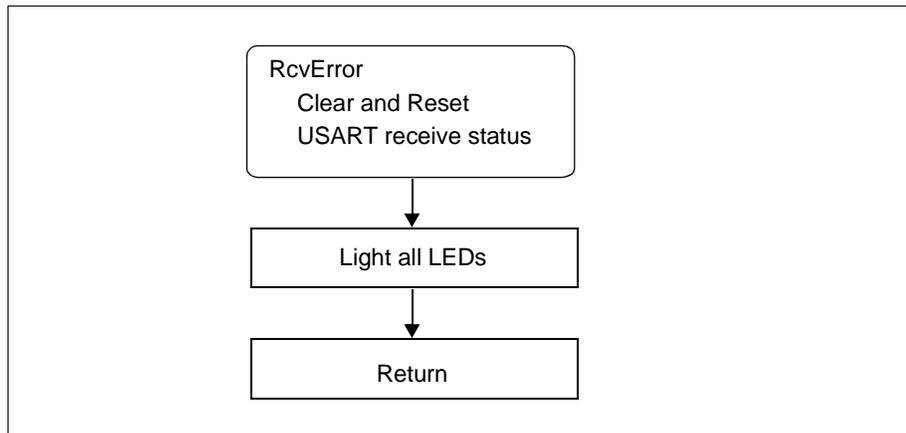


Figure 7.3: Interrupt Vector Routine

## 7.3 RcvError Routine

The receive error routine clears and then resets the USART receive status bit. It indicates that an error has occurred by lighting all the LEDs before returning to the main program.



**Figure 7.4: Receive Error Routine**

---

---

## Appendix A. Hardware Detail

---

---

The PICDEM-2 hardware is extremely simple and is intended to illustrate the ease of use of various PICmicro MCUs. The PICDEM-2 features the following hardware elements:

### A.1 Processor Sockets

Although two sockets are provided, only one device may be used at a time.

- 40-pin socket
- 28-pin socket

### A.2 Display

Eight red LEDs are connected to PORTB of each processor type. The PORTB pins are set high to light the LEDs. These LEDs may be disconnected from PORTB by removing jumper J6.

One green LED is provided to determine whether there is power to the PICDEM-2 board (LED lit) or not (LED off).

### A.3 Power Supply

There are three ways to supply power to PICDEM-2:

- A 9V battery can be plugged into J8.
- A 9V, 100mA unregulated AC or DC supply can be plugged into J2.
- A +5V, 100mA regulated DC supply can be connected to the hooks provided.

In-circuit emulator users have a regulated +5V power supply available in the logic probe connector and can easily connect to the hooks on PICDEM-2 (Red probe to +5V and Black probe to GND).

**Note 1:** The exact power up sequence listed in Getting Started (Chapter 2) of this guide must be followed.

**2:** The PICDEM-2 kit does not provide a power supply.

# PICDEM-2 User's Guide

---

## A.4 RS-232 Serial Port

An RS-232 level shifting IC has been provided with all necessary hardware to support connection of an RS-232 host through the DB9 connector. The port is configured as DCE, and can be connected to a PC using a straight through cable.

The PIC16C74 has its RX and TX pins tied to the RX and TX lines of the MAX232A.

## A.5 Switches

Three switches provide the following functions:

- S1 -  $\overline{\text{MCLR}}$  to hard reset the processor
- S2 - Active low switch connected to RA4
- S3 - CCP1 for capture of timer

Switches S1 and S3 have debounce capacitors whereas S2 does not, allowing the user to investigate debounce techniques.

When pressed, the switches are grounded. When idle, they are pulled high (+5V).

## A.6 Oscillator Options

- RC oscillator (2MHz approximately) supplied. This oscillator may be disabled by removing jumper J7.
- Pads provided for user furnished crystal and two capacitors.
- Socket provided for clock oscillator.

Provisions are also provided for an external crystal oscillator for Timer1.

## A.7 Analog Input

A 5K ohm potentiometer is connected through a series 470 ohm resistor to AN0.

The pot can be adjusted from VDD to GND to provide an analog input to the parts with an A/D module.

## A.8 Serial EEPROM

A 24LC01 1K (128x8) serial EEPROM is included on the board to illustrate I<sup>2</sup>C bus concepts. Provisions for an ACCESS.bus™ connector (J5) and associated resistors (R8-R13) are wired in parallel with the serial EEPROM.

For more information on the serial EEPROM, please refer to the most recent version of the *Non-Volatile Memory Products Data Book*.

## A.9 Keypad Header

J4 is a nine pin header which may be used as a keypad interface. Any 3x4 or 4x4 keypad may be used. For an example keypad, see *LCD Demo* (Chapter 6).

## A.10 LCD Header

J3 is a 14 pin header which may be used as an LCD module interface. Many different LCD modules may be used. For more information on LCD modules, refer to the manufacturers' data sheets. For an example LCD, see *LCD Demo* (Chapter 6).

## A.11 Sample Devices

A sample part programmed with a simple program is included in the PICDEM-2 kits.

Table 10-1 lists the I/O features and port connections for each processor type.

**TABLE A-1: PORT CONNECTIONS**

Processor	LEDs	RS-232 TX/RX	S1	S2	S3	Pot R16	Keypad Connector	LCD Control/Data	EEPROM/ACCESS.BUS
28-pin	PORTB	RC6/RC7 <sup>(1)</sup>	$\overline{\text{MCLR}}$	RA4	RC2	RA0 <sup>(2)</sup>	PORTB	—	RC3/RC4
40-pin	PORTB	RC6/RC7	$\overline{\text{MCLR}}$	RA4	RC2	RA0	PORTB	PORTA/D	RC3/RC4

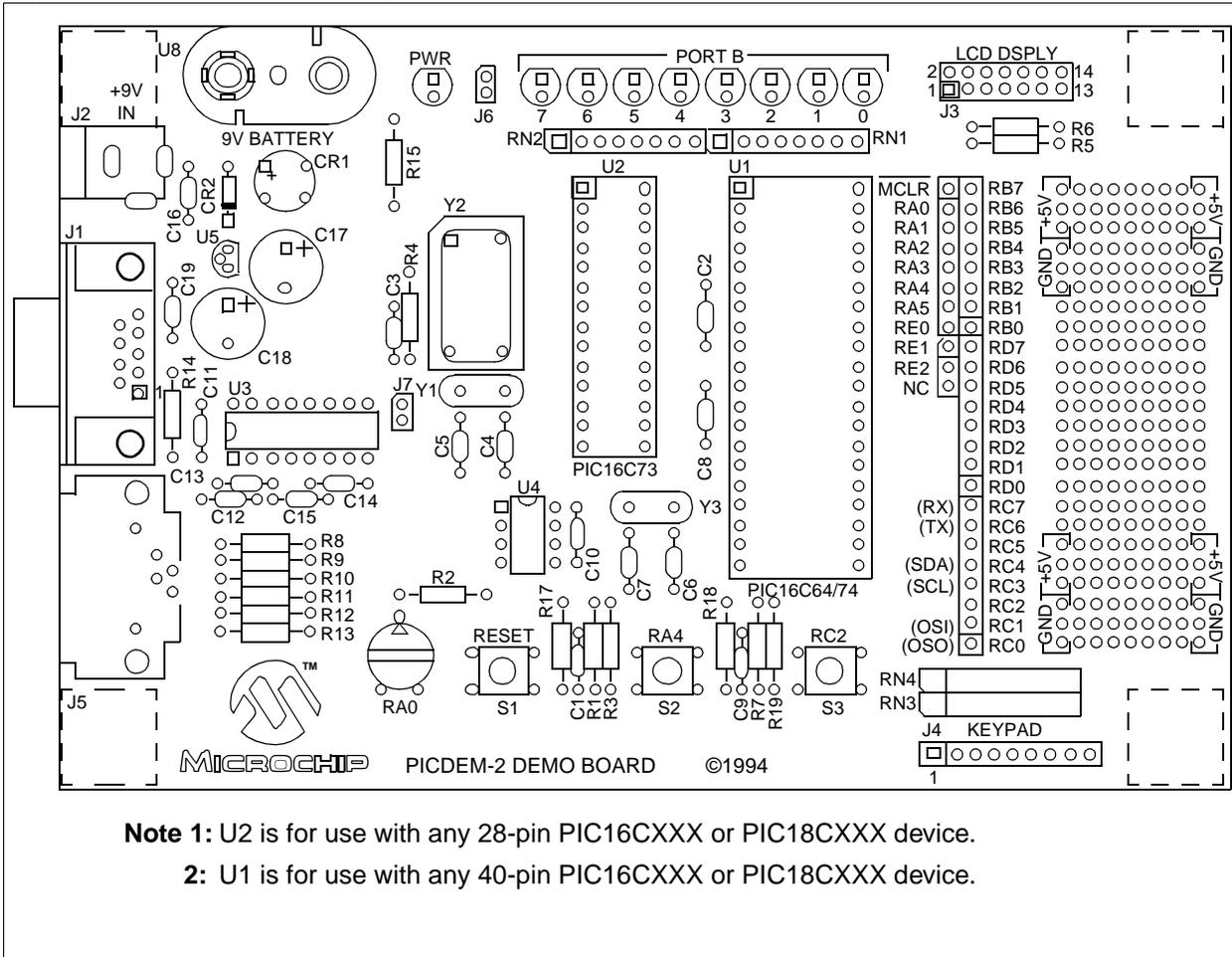
**Note** 1: PIC16C64 does not have a USART module.  
2: PIC16C64 does not have an A/D module.

# PICDEM-2 User's Guide

## A.12 Board Layout and Schematics

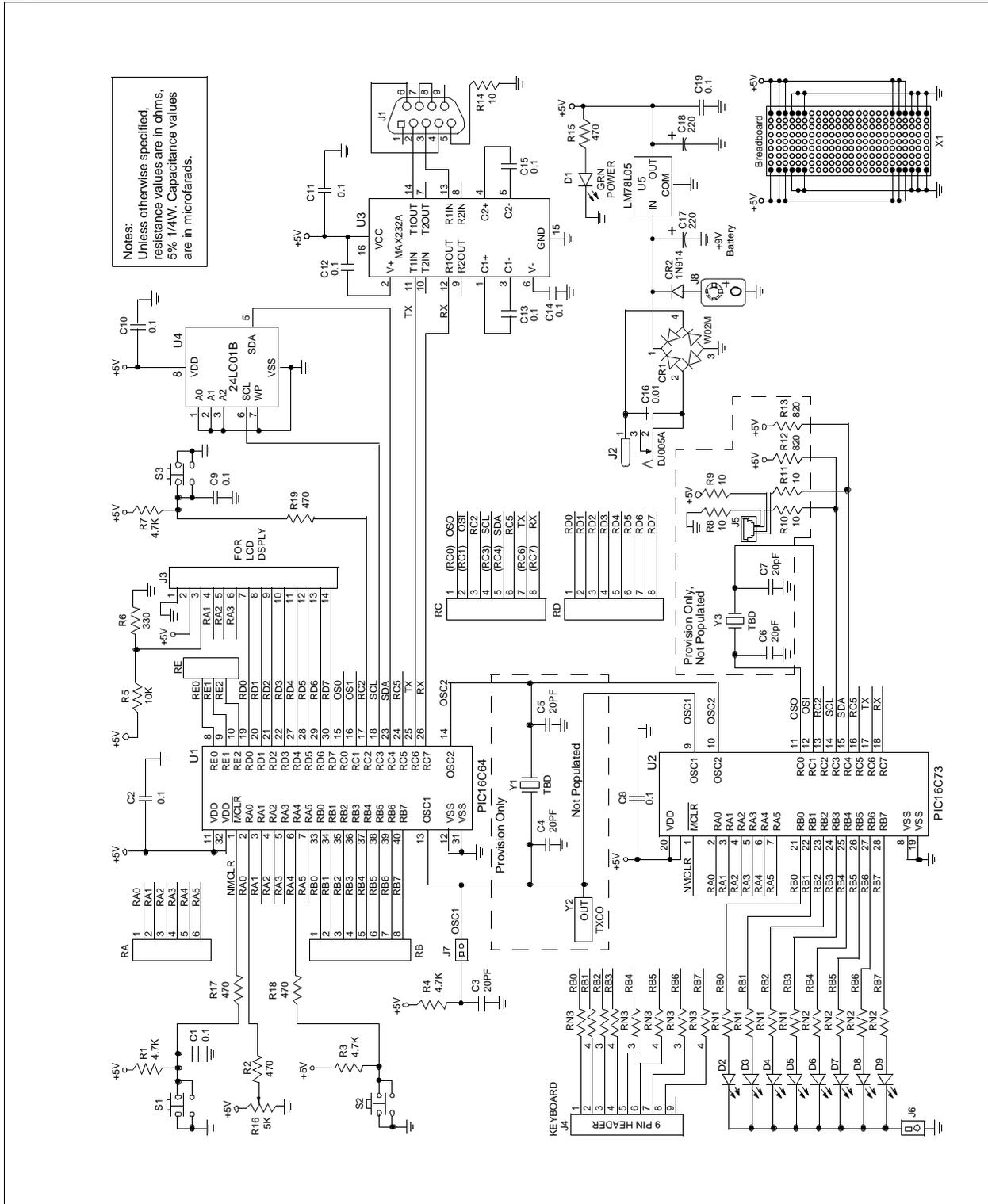
The following figures show the parts layout (silkscreen) and schematics for the PICDEM-2 board.

Figure A-1: PICDEM-2 Parts Layout



# Hardware Detail

Figure A-2: PICDEM-2 Schematic



# PICDEM-2 User's Guide

---

NOTES:



## Index

<b>A</b>			
A/D Demo .....	11		
A/D Input .....	2, 11, 46		
<b>B</b>			
Board .....	1, 2, 5, 45		
Parts Layout .....	48		
Power Supply .....	5, 45		
Schematic .....	49		
Silkscreen .....	48		
<b>D</b>			
Demonstration Board. <i>See</i> Board			
Demonstration Programs. <i>See</i> Sample Programs			
<b>E</b>			
EEPROM, Serial .....	2, 15, 47		
<b>H</b>			
Hardware .....	45		
<b>I</b>			
I <sup>2</sup> C Demo .....	15		
<b>K</b>			
Keypad .....	2, 31, 32, 47		
<b>L</b>			
LCD .....	2, 33, 47		
LCD Demo .....	31		
LEDs			
Green Power .....	2, 45		
Red Display .....	2, 5, 9, 11, 15, 31, 41, 45		
<b>M</b>			
MPASM .....	4, 7		
MPLAB .....	4, 7		
<b>P</b>			
PIC16C64			
I <sup>2</sup> C Demo .....	15		
LCD Demo .....	31		
Tutorial Program .....	9		
PIC16C74			
A/D Demo .....	11		
I <sup>2</sup> C Demo .....	15		
LCD Demo .....	31		
Tutorial Program .....	9		
		USART Demo .....	41
		PICDEM-2 Board. <i>See</i> Board	
		PICDEM-2 Kit .....	1
		PICSTART® Plus .....	4, 7
		PRO MATE® II .....	4, 7
		Pushbuttons. <i>See</i> Switches	
		<b>R</b>	
		RC Oscillator .....	46
		Reference Documents .....	4
		RS-232 .....	2, 41, 46
		<b>S</b>	
		Sample Devices .....	1, 3, 6, 47
		Sample Programs .....	1, 3, 6
		Schematic .....	49
		SCI .....	41
		SSP Module, I <sup>2</sup> C Mode .....	15
		Switches .....	2, 5, 9, 46
		<b>T</b>	
		Tutorial .....	9
		<b>U</b>	
		USART Demo .....	41

# PICDEM-2 User's Guide

---

NOTES:

**NOTES:**



## WORLDWIDE SALES AND SERVICE

### AMERICAS

#### Corporate Office

Microchip Technology Inc.  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 480-786-7200 Fax: 480-786-7277  
Technical Support: 480-786-7627  
Web Address: <http://www.microchip.com>

#### Atlanta

Microchip Technology Inc.  
500 Sugar Mill Road, Suite 200B  
Atlanta, GA 30350  
Tel: 770-640-0034 Fax: 770-640-0307

#### Boston

Microchip Technology Inc.  
5 Mount Royal Avenue  
Marlborough, MA 01752  
Tel: 508-480-9990 Fax: 508-480-8575

#### Chicago

Microchip Technology Inc.  
333 Pierce Road, Suite 180  
Itasca, IL 60143  
Tel: 630-285-0071 Fax: 630-285-0075

#### Dallas

Microchip Technology Inc.  
4570 Westgrove Drive, Suite 160  
Addison, TX 75248  
Tel: 972-818-7423 Fax: 972-818-2924

#### Dayton

Microchip Technology Inc.  
Two Prestige Place, Suite 150  
Miamisburg, OH 45342  
Tel: 937-291-1654 Fax: 937-291-9175

#### Detroit

Microchip Technology Inc.  
Tri-Atria Office Building  
32255 Northwestern Highway, Suite 190  
Farmington Hills, MI 48334  
Tel: 248-538-2250 Fax: 248-538-2260

#### Los Angeles

Microchip Technology Inc.  
18201 Von Karman, Suite 1090  
Irvine, CA 92612  
Tel: 949-263-1888 Fax: 949-263-1338

#### New York

Microchip Technology Inc.  
150 Motor Parkway, Suite 202  
Hauppauge, NY 11788  
Tel: 631-273-5305 Fax: 631-273-5335

#### San Jose

Microchip Technology Inc.  
2107 North First Street, Suite 590  
San Jose, CA 95131  
Tel: 408-436-7950 Fax: 408-436-7955

### AMERICAS (continued)

#### Toronto

Microchip Technology Inc.  
5925 Airport Road, Suite 200  
Mississauga, Ontario L4V 1W1, Canada  
Tel: 905-405-6279 Fax: 905-405-6253

### ASIA/PACIFIC

#### Beijing

Microchip Technology, Beijing  
Unit 915, 6 Chaoyangmen Bei Dajie  
Dong Erhuan Road, Dongcheng District  
New China Hong Kong Manhattan Building  
Beijing 100027 PRC  
Tel: 86-10-85282100 Fax: 86-10-85282104

#### Hong Kong

Microchip Asia Pacific  
Unit 2101, Tower 2  
Metroplaza  
223 Hing Fong Road  
Kwai Fong, N.T., Hong Kong  
Tel: 852-2-401-1200 Fax: 852-2-401-3431

#### India

Microchip Technology Inc.  
India Liaison Office  
No. 6, Legacy, Convent Road  
Bangalore 560 025, India  
Tel: 91-80-229-0061 Fax: 91-80-229-0062

#### Japan

Microchip Technology Intl. Inc.  
Benex S-1 6F  
3-18-20, Shinyokohama  
Kohoku-Ku, Yokohama-shi  
Kanagawa 222-0033 Japan  
Tel: 81-45-471-6166 Fax: 81-45-471-6122

#### Korea

Microchip Technology Korea  
168-1, Youngbo Bldg. 3 Floor  
Samsung-Dong, Kangnam-Ku  
Seoul, Korea  
Tel: 82-2-554-7200 Fax: 82-2-558-5934

#### Shanghai

Microchip Technology  
Unit B701, Far East International Plaza,  
No. 317, Xianxia Road  
Shanghai, 200051 P.R.C.  
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

### ASIA/PACIFIC (continued)

#### Singapore

Microchip Technology Singapore Pte Ltd.  
200 Middle Road  
#07-02 Prime Centre  
Singapore 188980  
Tel: 65-334-8870 Fax: 65-334-8850

#### Taiwan, R.O.C

Microchip Technology Taiwan  
10F-1C 207  
Tung Hua North Road  
Taipei, Taiwan, ROC  
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

### EUROPE

#### Denmark

Microchip Technology Denmark ApS  
Regus Business Centre  
Lautrup høj 1-3  
Ballerup DK-2750 Denmark  
Tel: 45 4420 9895 Fax: 45 4420 9910

#### France

Arizona Microchip Technology SARL  
Parc d'Activite du Moulin de Massy  
43 Rue du Saule Trapu  
Batiment A - 1er Etage  
91300 Massy, France  
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

#### Germany

Arizona Microchip Technology GmbH  
Gustav-Heinemann-Ring 125  
D-81739 München, Germany  
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

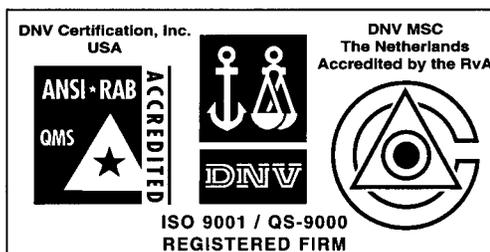
#### Italy

Arizona Microchip Technology SRL  
Centro Direzionale Colleoni  
Palazzo Taurus 1 V. Le Colleoni 1  
20041 Agrate Brianza  
Milan, Italy  
Tel: 39-039-65791-1 Fax: 39-039-6899883

#### United Kingdom

Arizona Microchip Technology Ltd.  
505 Eskdale Road  
Winnersh Triangle  
Wokingham  
Berkshire, England RG41 5TU  
Tel: 44 118 921 5858 Fax: 44-118 921-5835

01/21/00



Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELoc® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.

All rights reserved. © 2000 Microchip Technology Incorporated. Printed in the USA. 2/00 Printed on recycled paper.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. 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, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, except as maybe explicitly expressed herein, under any intellectual property rights. The Microchip logo and name are registered trademarks of Microchip Technology Inc. in the U.S.A. and other countries. All rights reserved. All other trademarks mentioned herein are the property of their respective companies.



**MICROCHIP**

Microchip Technology Inc.  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199

Tel: 480.786.7200 Fax: 480.899.9210

© 2000 Microchip Technology Inc. Printed in the U.S.A. 2/00  
DS30374a