PIC16F/18F MCU Mini Development Board Kit (PIC18F452 MCU)

LAB MANUAL

AMOTECH LABS

Embedded systems | Industrial Automation | Robotics

What'sApp /call: 832973756



Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations

Contents

Overview of PIC16F/18 MCU Development Kit	3
Introduction to PIC18F452 Microcontroller	4
MPLAB X IDE for Programming	6
MPLAB IPE for Program Downloading	11
Lab1. LED Blinking	.14
Lab3. LCD_Interfacing	16
Lab5. ADC interfacing	19
Lab6. Pulse Counter	24
Lab6. PWM Program	28





Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations

Overview of PIC18F452 Development Board Kit:



2. ICSP Conn & PICkit 3 Programmer.

Most of above labels are self explanatory. Below is the description needed for some of the labels. Also check brief Introduction to PIC18F452 & Schematic of the Kit in following pages for detail understanding.. **1. PIC16F/18F MCU:** This kit can be used with multiple 40Pin DIP MCU's from PIC like PIC16F887A/887, PIC18F4550/4580/4520 etc and all those which are pin compatible with them.

- 2. ICSP Conn.: It is connected to PIC Kit 3 to program the MCU. RST pin of the kit is connected on a side where there is a arrow symbol indicating RST pin of the Programmer.
- **3. MCU Ports.:** Port pins of Port A,B,C,D & E of PIC MCU can accessed with these connectors. Label of each pin is given on the right side of each pin and left side labels are of UDB connectors.
- 4. UDB Conn.: These connectors are used to attach this kit to 'Universal Development Board(UDB)' by Amotech Labs. Check UDB details on our website.
- **5. Keypad DIP.:**This DIP switch connects 4x4 Keypad to PortD externally.
- 6. +5V & GND Conn.: Using this connectors +5V GND can be given to external devices. Don't short them.
- 7. Power I/P : Provides power to the board via USB or an adapter.
- 8. Reset Sw : Resets the microcontroller to restart program execution.
- **9. LCD Contrast Pot :** Adjusts the 16x2 LCD display contrast.
- **10. Switches:** User inputs for controlling various functions.

AMOTECH LABS

Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations



Introduction to PIC18F452 Microcontroller

		1 /	
MCLR/VPP	⊢ 1	\bigcirc	40 🗋 🖛 🛏 RB7/PGD
RA0/AN0	→ 12		39 🗋 🛶 🗕 RB6/PGC
RA1/AN1	→→] 3		38 🗖 🛶 🗕 RB5/PGM
RA2/AN2/VREF-	→ 1 4		37 🗋 🛶 🕶 RB4
RA3/AN3/VREF+	→→ 1 5		36 🗆 ◄—► RB3/CCP2*
RA4/T0CKI	→ 1 6		35 - ←→ RB2/INT2
RA5/AN4/SS/LVDIN	→ 17		34 🗍 ←→ RB1/INT1
RE0/RD/AN5	→ 18	22	33 - + RB0/INT0
RE1/WR/AN6	→ 1 9	4	32 - VDD
RE2/CS/AN7	→ 10	8	31 - Vss
VDD	— → 11	5	30 - + RD7/PSP7
Vss	→ 112	ĕ	29 arr RD6/PSP6
OSC1/CLKI	→ n 13	ш	28 - + RD5/PSP5
OSC2/CLKO/RA6	→ 14		27 5 RD4/PSP4
RC0/T1OSO/T1CKI	→ 115		26 4 + RC7/RX/DT
RC1/T1OSI/CCP2*	→ 16		25 H - RC6/TX/CK
RC2/CCP1	• 17		24 A RC5/SDO
RC3/SCK/SCL	→ □ 18		23 RC4/SDI/SDA
RD0/PSP0	→ □ 19		22 H RD3/PSP3
RD1/PSP1			21 B - RD2/PSP2
			21

Features of PIC18f452:

- ✓ Operating Frequency : DC 40 Mhz
- ✓ Program Memory (Bytes): 32K
- ✓ Program Memory (Instructions) : 16384
- ✓ Data Memory (Bytes): 1536
- ✓ Data EEPROM Memory (Bytes): 256
- ✓ Interrupt Sources 18
- ✓ I/O Ports : Ports A, B, C, D, E
- ✓ Timers : 4
- ✓ Capture/Compare/PWM Modules : 2
- ✓ Enhanced Capture/ Compare/PWM Modules : 2
- ✓ Serial Communications : MSSP, Addressable USART
- ✓ Universal Serial Bus (USB) Module : No
- ✓ Streaming Parallel Port (SPP) : Yes
- ✓ 10-Bit Analog-to-Digital Module : 8 Input Channels
- ✓ Comparators : 2
- ✓ Programmable Low-Voltage Detect : Yes
- ✓ Instruction Set : 75 Instructions; 75 Instructions
- ✓ Packages: 40-pin DIP, 44-pin PLCC, 44-pin TQFP

AMOTECH LABS



Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations

www.amotechlabs.com 🕒 +91-8329537565 M amotechlabs@gmail.com

4

MPLAB X IDE for Programming PIC18F/16F MCU's

Below are the steps to program any PIC18F/16F MCU in MPLAB X IDE.

1. Double click on MPLAB X IDE icon on the desktop & Open MPLAB X IDE.



2. To create new project. Select File -> New Project.



AMOTECH LABS

Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations

3. After that below window will appear. Select Microchip Embedded ->Standalone Project & Click next.

:ps	Choose Project						
1. Choose Project	Q Filter:						
	Categories: Projects:						
	Microchip Embedded Other Embedded Samples	Standalone Project Existing MPLAB IDE v8 Project Prebuilt (Hex, Loadable Image) Project User Makefile Project Library Project					
	Description:						
	Creates a new standalone applicatio project.	n project. It uses an IDE-generated makefile to build your					

4. Select Family >Advanced 8-bit MCUs(PIC18) then select Device>PIC18F452 or any other MCU. In Tool you can select No Tool.

🔀 New Project					×
Steps	Select Device				
 Choose Project Select Device Select Header Select Plugin Board Select Compiler Select Project Name and Folder (Optional) Add Project 	Family: Device: Tool:	All Families PIC18F452 No Tool	 ✓ ✓ ✓ 	Show All	
MPLAB X IDE					
< Back Add An	other Project	Next >	Finish	Cancel	Help

Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations

www.amotechlabs.com 🕒 +91-8329537565 M amotechlabs@gmail.com

AMOTECH

LABS

7

5. Now select the previously installed latest version of XC8 compiler and then click Next.

New Project	X	
Steps 1. Choose Project 2. Select Device 3. Select Header 4. Select Plugin Board 5. Select Compiler 6. Select Project Name and Folder	Select Compiler Toolchains	
MPLAB X IDE		
	< Back Next > Finish Cancel Help	

6. Enter your Project name > Click on Browse and Choose Project Folder location > Click on Finish.

🔀 New Project				×
Steps	Select Project Name	and Folder		
1. Choose Project 2. Select Device 3. Select Header	Project Name:	LED_Blinking]	
 Select Tool Select Plugin Board 	Project Location:	C:\Users\User\MPLABXProjects	Browse	
6. Select Compiler 7. Select Project Name and	Project Folder:	C:\Users\User\MPLABXProjects\LED_Blinking.X]	
MPLAB X IDE	Overwrite existing Also delete source Set as main project Use project location	g project. es. ct on as the project folder		
	Encoding: ISC	D-8859-1		
		< Back Next > Finish	Cancel H	ielp



Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations

7. Now our project is displayed in Project window.



8. Create the Main file for developing code. Right Click on Source file in the project -> Select new -> Select main.c File.

Projects ×			Start Page 🗙 📮	Configuratio
LED_Blinking Header Files Configura Important File Linker Files	ations_Header_File.h es		Source History 1 - /* Mi 2 * an 3 * 4 * TH	crochip d any de
E Carce P	New	>	Directory	-
⊡ 🦾 Loadable	New Logical Folder Add Existing Item Add Existing Items from Folders Find		 main.c xc8_header.h pic_8b_asm_fu Text File 	nc.asm
	Conv	L	13 * BF	EN ADVIS
	Paste	Ctrl+V	14 * FU 15 * TN	LLEST EX
	Remove From Project		16 * AN	Y, THAT
	Rename		18 * MI	CROCHIP
LED_Blinking - Da	Properties		19 * TE	RMS.



Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations

9

9. Give name to C file and click on finish.

🛿 New main.c		>
Steps	Name and Location	
1. Choose File Type	File Name: blink_of_led	
	Extension: c v	
	Set this Extension as Default	
	Project: LED_Blinking	
	Folder:	Browse
	Created File: C:\Users\User\MPLABXProjects\LED_Blinking.X\blink_of_led	.c
	1	કે
	< Back Next > Finish	Cancel Help

- 10. Write your program.
- 11. Afer you complete your program then Build it by clicking on a Hammer as seen below.

🔀 MPLAB X IDE v3.45 - LED_Blinking : default

File	Edit	View	Navigate	Source	Refactor	Run	Debug	Team	Tools	Window	Help					
Ċ	9		-	9 (defau	ult		~	r	• 👸 •		• 🄽	- 🏠	• 🖓	•	

12. Afer building a Project, output Window appears which gives errors and warnings if any. otherwise Build Successful message appears as seen below.

🖥 Ou	tput - LED_Blinking (Build, Load) ×									
	- · ·										
	Memory Summary:										
	Program space	used	64h	(100)	of	8000h	bytes	(0.3%)		
	Data space	used	6h	(6)	of	800h	bytes	(0.3%)		
	Configuration bits	used	7h	(7)	of	7h	words	(100.0%)		
	EEPROM space	used	Oh	(0)	of	100h	bytes	(0.0%)	т	
	ID Location space	used	8h	(8)	of	8h	bytes	(100.0%)	T	
	Data stack space	used	Oh	(0)	of	7A0h	bytes	(0.0%)		
	You have compiled in FR Using Omnicient Code Ger you could have produced See http://www.microchig	E mode. meration up to 6 o.com/MP	that 0% sma LABXCo	is a alle: comp:	availa r and ilers	able 400 for	in PR % fast more :) mode, er code informa	tion.		
	<pre>make[2]: Leaving directory 'C:/Users/User/MPLABXProjects/LED_Blinking.X' make[1]: Leaving directory 'C:/Users/User/MPLABXProjects/LED_Blinking.X'</pre>										
	BUILD SUCCESSFUL (total	time: 4	s)								
	Loading code from C:/Use	ers/User	/MPLAE	BXPro	ject:	s/LE	D_Blin	king.X/	dist/default/pro	duction/LED_Blinking.X.production.hex	
	Loading completed										

12. After BUILD SUCCESSFUL, Hex file is created in the project folder. In project folder Hex file is created in dist/default/production. You can also see a path to the folder under BUILD SUCCESSFUL.



Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations



MPLAB X IPE for Programming PIC18F/16F MCU's

Below are the steps to program any PIC18F/16F MCU in MPLAB X IDE.

1.Open MPLAB IPE software. Select the Family and Device name of the MCU you want to program. Connect the USB of PICKit 3/3.5 Programmer to your PC, after connecting it the programmer name will appear in the tool. Now connect 12V power supply adapter to PIC18F/16F Kit and PICKit 3's 6 Pin connector to your PIC18F/16 kit directly as seen in the Kit Overview Pic on Page 3. Also, you can connect it using 6-pin F-F connector wires. PICKit 3's first pin indicated with arrow should be connected to RST pin of your PIC18F/16F kits 6-pin connector. Now, click on 'Apply' and then on 'Connect'.



2. Once you click on Connect, below pop-up window may appear if you have not done Programming Power Source and Voltage level settings. Programming voltage and Power settings should be done before downloading the Program.





Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations

3. To confirm if correct Programming Voltage and Power Source is selected, we need to click on Setting and select Advance Mode. Then below Pop-Up window will appear. Enter the default password '**microchip**' as given in the Hint and press Log in. Now you go in Advanced mode.

-	<						-	_	~
MPLAB IPE	v6.20					_	- L]	×
File Settings \	view Tools Window Help								
Operate									- 🗆
Device and	Tool Selection			Results					Î
Family:	All Families	Ŧ			Checksum:	82D8		62	
Device:	PIC18F452	Advanced Mode	Apply	×	Pass Count:	201			
Tool:	Select Tool	Log on to advanced mode			Fail Count:	44 245			
		Password: Hint: Default pa	assword is microchip						
B	Program	Change Password	Log in Forgot Password	1		Blank Cl	heck		
Hex File:	Click on browse to select a	Keep me logged in			Br	owse	lear sele	ction	
SQTP File:	Click on browse to select a	SQTP file			Br	owse C	lear sele	ction	
Output - IPE	x								

4. Once you are in Advance Mode, you will see the 'Options' tab on the left. Click on it and select 'Power' tab, then below window will appear. Un tick the below check box and select the Voltage level as 5.0 Volt. Now, must be powered by external Power Supply before downloading the program.

(MPLA	B IPE v6.20							
File	Setti	ings View Tools Window Help							
æ	Deperate Power Settings ×								
ions	Pov	ver Settings							
Opt									
9		Voltage options							
		Power target circuit from PICkit3							
		Voltage Level	5.0 👻						
		Program Options							
		Use low voltage programming							
	l		Reset to defaults						

AMOTECH LABS

Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations

5. Once Voltage and Power Source is selected, click on '**Operate**' tab. Now, select you Hex file by by clicking on 'Browse' and navigate to your Hex file location. In your Projet Folder you need to navigate to dist>default>production and select your Hex File.

-						
Operate	Power Settings ×			4		
Device and Tool Selection			Results			
Family:	All Families	•	CP=OFF Che	cksum: 82D8	82D8	
Device	Hex file selection	Annhi			×	
Tool:	\leftrightarrow \rightarrow \checkmark \uparrow 🚞 « di	st > default > production	~ C Searc	h production	ρ	
	Organize 👻 New folder			≣ • □	0	
	Pictures	Name	Date modified	Туре	Si	
	📒 Lift Cylinder 🏾 📌	ADC_LCD_Interface.X.production.hex	24-02-2025 09:59	HEX File		
Hex Fil	🕖 Music 🔹 🖈					
SQTP	🛂 Videos 🔹 🖈					
SQTP	🛂 Videos 🔹 🖈					
SQTP Coutp	 Videos * All Program pdf in Tes production 					
SQTP Coutp MPI MPI	Videos * All Program pdf in Te- production production					
SQTP Coutp MPI MPI	 Videos All Program pdf in Te- production production New Programs 					
SQTP Coutpo MPI MPI	 Videos All Program pdf in Te- production production New Programs 					
SQTP Coutp MPI MPI Inv	 Videos All Program pdf in Teo production production New Programs 			File (*.HEX;*.hex)	~	

6. Once you select the Hex file you will see the notification Hex File Loaded Successfully in 'Output IPE'. Click on 'Program' button. You will see PICKit 3 LED's flashing and 'Programming...' in Output-IPE. And, finally 'Programming complete' once downloaded. Now remove PICKit 3 and see your program running on the Kit.

Operate		
Device and To	pol Selection	Results
Family:	All Families	CP=OFF Checksum: 4E24
Device:	PIC18F452	Pass Count: 195
Tool:	PICkit3 S.No : BUR161106730 * Disconnec	ct Fail Count: 38
		Total Count: 233
Hex File:	rogram Erase Read	Verify Elank Check
Hex File:	rogram Erase Read D:\Files\AMOTECHQuotes\Pune University Joshi Madam\PiC18F452_Pro Click on browse to select a SQTP file	Verify Blank Check ograms_mplab_xNew Progr Browse Clear selection Browse Clear selection
Hex File: SQTP File:	rogram	Verify Slank Check ograms_mplab_x:New Progr Browse Clear selection Browse Clear selection
Hex File: SQTP File: Coutput - IPE Loading c Approxima 2025-02-2 2025-02-2 Device Er	rogram	Verify Elank Check ograms_mplab_x!New Progr Browse Clear selection Browse Clear selection Nadam\PIC10F452_Programs_mplab_x\New Programs
Hex File: SQTP File: Coutput - IPE Loading of Approxima 2025-02-2 2025-02-2 Device Er Programmi	rogram	Verity Elank Check ograms_mplab_xINew Progr Browse Clear selection Browse Clear selection Nadam\PIC10F452_Programs_mplab_x\New Programs
Hex File: SOTP File: Couput - IPE Loading c Approxima 2025-02-2 2025-02-2 Device Er Programmi The follo program ar	rogram Erase Read D:FHEMAMOTECH/Quotes/Pune University Joshi Madam/PIC18F452_Pro Click on browse to select a SQTP file x	Verify Elank Check ograms_mplab_x\New Progr Browse Clear selection Browse Clear selection Browse Clear selection Nadam\PICl0F452_Programs_mplab_x\New Programs
Hex File: SOTP File: Cotput - IPE Leading c Approxima 2025-02-2 Device Er Programmi The follo program configura	rogram Erase Read D:(Files/AMOTECH/Quotes/Pune University Joshi Madam/PIC18F452_Pro Click on browse to select a SQTP file Click on select a SQTP file	Verity Elank Check ograms_mplab_x!New Progr Browse Clear selection Browse Clear selection Nadam\PIC10F452_Programs_mplab_x\New Programs
Hex File: SOTP File: Cotput - IPE Leading c Approxima 2025-02-2 2023-02-2 Device Er Programmi The follo programmi Programmi	Image: State Image: State D:Files/AMOTECHQuotes/Pune University Joshi Madam/PIC18F452_Pro Click on browse to select a SQTP file Image: State	Verify Elank Check ograms_mplab_x(New Progr Browse Clear selection Browse Clear selection . Madam\PIC18F453_Programs_mplab_x\Nev Programs

AMOTECH LABS

Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations

Lab 1 - LED Blinking

Aim: To study the LED's Interfacing with PIC18F452 mini development.



Procedure:

- 1. Read the LED Blink Program on next page with all the comments which explains the program.
- 2. Open the Project folder for the program in MPLAB X IDE or write it by making new project as per MPLAB X IDE procedure explained in the manual. Build and generate the Hex file for the same.
- 3. Open MPLAB IPE and follow the procedure explained to download the Hex file for the above
- program.
- 4. Use PICkit3 programmer to program the PIC microcontroller with the HEX file.
- 5. Make sure that the arrow side of the programmer must be connected to the RST pin of the ICSP connector of the board.

AMOTECH LABS



Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations

Program:

```
1 #include<xc.h>
2 #define XTAL FREQ 20000000 // Define the crystal frequency (8 MHz)
3 #include<pic18f452.h> // Header file to include PIC18F452 MCU configuration
4 #pragma config OSC = HS // High Frequency External Crystal selected
5 #pragma config WDT = OFF //Watch Dog Timer OFF
6 #pragma config LVP = OFF //Low Voltage Programming OFF
7 // LED Pin
8 #define LED LATCbits.LATC2 // RC.2 connected to LED
9 voidmain ()
10 {
     TRISCbits.TRISC2 = 0; // Configure LED Pin as Output Pin
11
12
     while(1)
13
     {
      LED = 0;
14
15
        delay ms(1000); // 1000 mSec delay
      LED = 1:
16
17
        delay ms(1000); // 1000 mSec delay
18
     }
19 }
20
```



Lab 2 - LCD Interfacing

10K Pot

Aim: To study the 16x2 LCD Interfacing with PIC18F452 mini development kit.

Procedure:

- 1. Read the LCD Interfacing Program on next page with all the comments which explains the program.
- 2. Open the Project folder for the program in MPLAB X IDE or write it by making new project as per MPLAB X IDE procedure explained in the manual. Build and generate the Hex file for the same.
- 3. Open MPLAB IPE and follow the procedure explained to download the Hex file for the above program.
- 4. LCD contrast can be adjusted using 10k pot beside LCD. Once you download the program and remove the programmer, you will see above contents on the LCD.
- 5. Make sure that the arrow side of the programmer must be connected to the RST pin of the ICSP connector of the board.

AMOTECH LABS



Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations

Program:

```
1 #include<xc.h>
2 #define XTAL FREQ 20000000 // Define the crystal frequency (8 MHz)
3 #include<pic18F452.h>
4 #include<stdio.h>
5 #include<string.h>
6 #include<stdlib.h>
7 #pragma config OSC = HS
8 #pragma config WDT = OFF
9 #pragma config LVP = OFF
10 #pragma config BOR = OFF // Brown-out Reset disabled
11 // LCD Pins
12 #define RS LATDbits.LATD0
                                    // RS connected to RF0
13 #define EN LATDbits.LATD1
                                    // EN connected to RE1
14 #define D4 LATDbits.LATD4
                                    // Data line 4
15 #define D5 LATDbits.LATD5
                                    // Data line 5
16 #define D6 LATDbits.LATD6
                                    // Data line 6
17 #define D7 LATDbits.LATD7
                                   // Data line 7
18 void LCD Init();
                            /*Initialize LCD*/
19 void LCD Command(unsigned char); /*Send command to LCD*/
20 void LCD_Char(unsigned char x); /*Send data to LCD*/
21 void LCD String(const char *); /*Display data string on LCD*/
22 void LCD_String_xy(char, char, const char *);
23 void LCD_Clear();
                          /*Clear LCD Screen*/
24 int main(void)
25 {
26
     /* OSCCON = 0x72; //Use internal oscillator and
27
              // set frequency to 8 MHZ*/
28
     TRISD = 0x00; // Set PORTD as output for LCD data
29
       LCD Init(); /*Initialize LCD to 5*8 matrix in 4-bit mode*/
30
       LCD String xy(1,0,"PIC Mini Kit by"); /*Display string on 1st row*/
31
       delay ms(1000);
32
     LCD_String_xy(2,0,"Amotech Labs"); /*Display string on 2nd row,*/
33
       delay ms(1000);
34
       while(1);
35 }
    36 /*
37 void LCD_Init()
38 {
39
     LCD Command(0x02); // Initialize LCD in 4-bit mode
     LCD Command(0x28); // 2 lines, 5x7 matrix
40
41
     LCD Command(0x0C); // Display ON, Cursor OFF
42
     LCD Command(0x06); // Increment cursor
43
     LCD_Command(0x01); // Clear display
44
     delay ms(2);
45 }
46 void LCD Command(unsigned cha r cmd)
47 {
48
        RS = 0; // Command mode
```

AMOTECH LABS

Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations

AMOTECH LABS Sharing Enthusiasm for Embedded Systems & Automation.

```
49
     D4 = (cmd >> 4) \& 0x01;
50
     D5 = (cmd >> 5) \& 0x01:
51
     D6 = (cmd >> 6) \& 0x01;
52
     D7 = (cmd >> 7) \& 0x01;
53
     EN = 1; delay ms(1); EN = 0; // Latch the command
54
     D4 = cmd & 0x01:
55
56
     D5 = (cmd >> 1) \& 0x01;
57
     D6 = (cmd >> 2) \& 0x01;
58
     D7 = (cmd >> 3) \& 0x01:
59
     EN = 1; delay ms(1); EN = 0; // Latch the command
60 }
61 void LCD Char(unsigned char data)
62 {
63
       RS = 1; // Data mode
64
     D4 = (data >> 4) \& 0x01;
65
     D5 = (data >> 5) \& 0x01;
66
     D6 = (data >> 6) \& 0x01;
67
     D7 = (data >> 7) \& 0x01;
68
     EN = 1; delay ms(1); EN = 0; // Latch the data
69
70
     D4 = data \& 0x01;
71
     D5 = (data >> 1) \& 0x01;
72
     D6 = (data >> 2) \& 0x01;
73
     D7 = (data >> 3) \& 0x01;
74
     EN = 1; delay ms(1); EN = 0; // Latch the data
75 }
76 void LCD String(const char *msg)
77 { while((*msg)!=0) {
78
       LCD Char(*msg);
79
         msg++;
                    }
80 }
81 void LCD_String_xy(char row, char pos, const char *msg) {
82 char location=0;
83 if(row<=1) {
84 location=(0x80)|((pos)\&0x0f);
                                    /*Print message on 1st row and given column*/
85 LCD Command(location); }
86 else {
87 location=(0xC0)|((pos)\&0x0f);
                                   /*Print message on 2nd row and given column*/
88 LCD Command(location);
89 }
90 LCD_String(msg);
91 }
92 void LCD_Clear()
93 {
      LCD_Command(0x01); /*clear display screen*/
94
95
        delay ms(3);
96 }
```

Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations

Lab 3 - ADC Interfacing

Aim: To study the ADC interfacing with PIC18F452 mini development kit.



Procedure:

- 1. Read the ADC Interfacing Program on next page with all the comments which explains the program. Open the Project folder for the program in MPLAB X IDE or write it by making new project as perMPLAB X IDE procedure explained in the manual. Build and generate the Hex file for the same.
- 2. Open MPLAB IPE and follow the procedure explained to download the Hex file for the above program.
- 3. Use PICkit3 programmer to program the PIC microcontroller with the HEX file.
- 4. Make sure that the arrow side of the programmer must be connected to the RST pin of the ICSP connector of the board.





Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations

Program:

```
1 #include<xc.h>
2 #include<pic18F452.h>
3 #include<stdio.h>
4 #include<string.h>
5 #include <stdlib.h>
6 #define XTAL FREQ 20000000 // Define the crystal frequency (20 MHz)
7 #pragma config OSC = HS
8 #pragma config WDT = OFF
9 #pragma config LVP = OFF
10 #pragma config BOR = OFF // Brown-out Reset disabled
11 // LCD Pins
12 #define RS LATDbits.LATD0// RS connected to RE0
13 #define EN LATDbits.LATD1// EN connected to RE1
14 #define D4 LATDbits.LATD4// Data line 4
15 #define D5 LATDbits.LATD5// Data line 5
16 #define D6 LATDbits.LATD6// Data line 6
17 #define D7 LATDbits.LATD7// Data line 7
/*Initialize LCD*/
19 void LCD Init();
20 void LCD Command(unsigned char); /*Send command to LCD*/
21 void LCD_Char(unsigned char x); /*Send data to LCD*/
22 void LCD_String(const char *); /*Display data string on LCD*/
23 void LCD_String_xy(char, char , const char *);
24 void LCD Clear():
                             /*Clear LCD Screen*/
25 void ADC Init();
26 int ADC_Read(int);
27 #define vref 5.00
28 int main(void)
29 {
30
     char data[10];
31
     TRISD = 0x00; // Set PORTD as output for LCD data
32
     unsigned int digital;
33
     float voltage;
34
     int k;
35
     char ADC_Array[5];
36
    // OSCCON = 0x72; // Use internal oscillator and
37
             // set frequency to 8 MHZ
38 LCD Init(); /*Initialize LCD to 5*8 matrix in 4-bit mode*/
39 ADC Init();
40 LCD String xy(1,0,"voltage is..");/*Display string on 1st row,5th location*/
41
     LCD Command(0xC0);
42
43
       while(1)
44
    {
45
       digital=ADC Read(3);
46
       /*Convert digital value into analog voltage*/
47
       voltage= digital*((float)vref/(float)1023);
48
       for(k=0; k<=3; k++)
                              /* Convert the result into ASCII*/
```

Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations

AMOTECH LABS Sharing Enthusiasm for Embedded Systems & Automation.

```
49 {
                     /* Separate each digit of the integer */
50
    ADC Array[k] = digital%10+'0';
    digital = digital/10;
51
52 }
53 LCD Command(0XC0);
54 for(k=3; k>=0; k--)
55 {
56
    LCD Char(ADC Array[k]);
                                 /* Display the result on LCD */
57 }
58
       /*It is used to convert integer value to ASCII string*/
59
       sprintf(data,"%.2f",voltage);
60
61
       strcat(data," V"); /*Concatenate result and unit to print*/
62
       LCD String xy(2,6,data); /*Send string data for printing*/
        __delay_ms(250);
63
64
     }
65 }
66 void ADC_Init()
67 {
                          /*Set as input port*/
68
     TRISA = 0xff;
69
     ADRESH=0;
                           /*Flush ADC output Register*/
     ADRESL=0;
70
71 }
72 int ADC_Read(int channel)
73 {
74
     unsigned int digital;
75
     ADCON0 =(ADCON0 & 0b11000111)|((channel<<3) & 0b00111000);
76
    /*channel 0 is selected i.e.(CHS3CHS2CHS1CHS0=0000)& ADC is disabled*/
77
    // ADCON0 |= ((1<<ADON)|(1<<GO));/*Enable ADC and start conversion*/
78
    ADCON0bits.ADON=1;
79
    ADCON0bits.GO nDONE=1;
80
    /*wait for End of conversion i.e. Go/done'=0 conversion completed*/
81
     while(ADCON0bits.GO nDONE==1);
82
     digital = ADRESH;
83
     digital <<= 8; //Left shift the lower byte into the higher byte of tvalue
84
     digital += ADRESL; // Insert the TL0 byte into the lower byte of tvalue
85
     digital >>= 6;
86
     return(digital);
87 }
89 void LCD_Init()
90 {
91
     LCD Command(0x02); // Initialize LCD in 4-bit mode
     LCD_Command(0x28); // 2 lines, 5x7 matrix
92
93
     LCD Command(0x0C); // Display ON, Cursor OFF
94
     LCD Command(0x06); // Increment cursor
95
     LCD Command(0x01); // Clear display
96
       _delay_ms(2);
97 }
```

```
AMOTECH
LABS
```

Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations

```
98 void LCD Command(unsigned char cmd)
99 {
100
         RS = 0; // Command mode
101
      D4 = (cmd >> 4) \& 0x01:
102
      D5 = (cmd >> 5) \& 0x01;
103
      D6 = (cmd >> 6) \& 0x01;
104
      D7 = (cmd >> 7) \& 0x01;
105
      EN = 1; delay ms(1); EN = 0; // Latch the command
106
107
      D4 = cmd \& 0x01;
108
      D5 = (cmd >> 1) \& 0x01;
109
      D6 = (cmd >> 2) \& 0x01;
110
      D7 = (cmd >> 3) \& 0x01;
111
      EN = 1; delay ms(1); EN = 0; // Latch the command
112 }
113 void LCD Char(unsigned char data)
114 {
115
        RS = 1; // Data mode
116
      D4 = (data >> 4) \& 0x01:
      D5 = (data >> 5) \& 0x01;
117
      D6 = (data >> 6) \& 0x01;
118
119
      D7 = (data >> 7) \& 0x01;
120
      EN = 1; delay ms(1); EN = 0; // Latch the data
121
122
      D4 = data \& 0x01;
123
      D5 = (data >> 1) \& 0x01;
      D6 = (data >> 2) \& 0x01:
124
      D7 = (data >> 3) \& 0x01;
125
126
      EN = 1; delay ms(1); EN = 0; // Latch the data
127 }
128 void LCD String(const char *msg)
129 {
         while((*msg)!=0)
130
131
         ł
          LCD Char(*msg);
132
133
         msg++;
134
      }
135 }
136 void LCD String xy(char row, char pos, const char *msg)
137 {
138
      char location=0;
139
      if(row<=1)
140
      {
         location=(0x80) | ((pos) & 0x0f); /*Print message on 1st row
141
                               * and desired location*/
142
143
         LCD Command(location);
144
      }
```

AMOTECH LABS



AMOTECH LABS Sharing Enthusiasm for Embedded Systems & Automation.

145	else
146	{
147	location=(0xC0) ((pos) & 0x0f); /*Print message on 2nd row
148	* and desired location*/
149	LCD_Command(location);
150	}
151	
152	LCD_String(msg);
153 }	
154 vo	bid LCD_Clear()
155 {	
156	LCD_Command(0x01); /*clear display screen*/
157	delay_ms(3);
158 }	
159	
160	





Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations

Lab 4 - Pulse Counter

Aim: To study the Pulse counter program with PIC18F451 mini board.



.Procedure:

- 1. Read the pulse counter Program on next page with all the comments which explains the program.
- 2. Open the Project folder for the program in MPLAB X IDE or write it by making new project as per
- MPLAB X IDE procedure explained in the manual. Build and generate the Hex file for the same.
- 3. Open MPLAB IPE and follow the procedure explained to download the Hex file for the above program.
- 4. make the connections as above shown in figure.







Program:

1 #include<xc.h> /*The #include <xc.h> directive is used in PIC microcontroller programming with the MPLAB X IDE and XC8 compiler */ 2 // Define the crystal frequency (20 MHz) 3 #define XTAL FREQ 2000000 4 #include<pic18F452.h> 5 #include<string.h> 6 #include<stdio.h> // For sprintf 7 #pragma config OSC = HS 8 #pragma config WDT = OFF //Watchdog Timer is disabled. 9 #pragma config LVP = OFF /* LVP is disabled*/ // Brown-out Reset disabled 10 #pragma config BOR = OFF 11 // LCD Pins 12 #define RS LATDbits.LATD0 // RS connected to RE0 13 #define EN LATDbits.LATD1 // EN connected to RE1 // Data line 4 14 #define D4 LATDbits.LATD4 // Data line 5 15 #define D5 LATDbits.LATD5 16 #define D6 LATDbits.LATD6 // Data line 6 17 #define D7 LATDbits.LATD7 // Data line 7 18 // Function prototypes 19 void LCD_Init(void); 20 void LCD Command(unsigned char cmd); 21 void LCD_Char(unsigned char data); 22 void LCD_String(const char *str); 23 void LCD_Clear(void); 24 void LCD SetCursor(unsigned char row, unsigned char column); 25 void setup(); 26 void displayCount(unsigned int count); 27 // Global variables 28 volatile unsigned int pulseCount = 0; 29 30 void setup() { 31 // Configure Timer1 in Counter Mode 32 T1CON = 0x87;//Timer1 ON, External clock from RC2 (T1CKI), Prescaler 1:1 33 TRISCbits.TRISC0 = 0; // Set RC2/T1CKI as input 34 // Configure Timer0 in Counter Mode for Ra4 35 TOCON = 0xA8;// Timer0 ON, 8-bit counter mode, 36 //External clock source on RA4/T0CKI, no prescaler 37 // Set RA4 as input TRISAbits.TRISA4 = 1; // Set PORTD as output for LCD data 38 TRISD = 0x00;// Initialize the LCD 39 LCD_Init(); // Clear Timer0 and Timer1 counts 40 TMROL = 0;41 **TMR0H = 0**: 42 TMR1H = 0;// Clear Timer1 High byte 43 TMR1L = 0;// Clear Timer1 Low byte 44 }

AMOTECH LABS



AMOTECH LABS Sharing Enthusiasm for Embedded Systems & Automation.

45 void main() { 46 setup(); 47 LCD Clear(); 48 while (1) 49 { // Read the Timer1 counter value (pulse count) 50 51 unsigned int t1PulseCount = (TMR1H << 8) | TMR1L; 52 unsigned int t0PulseCount = (TMR0H << 8) | TMR0L; 53 char t0buffer[16],t1buffer[16]; 54 sprintf(t1buffer, "SW1 Count: %u", t1PulseCount); //Format the count value sprintf(t0buffer, "SW2 Count: %u", t0PulseCount); 55 //Format the count value 56 // LCD Clear(); LCD SetCursor(1,1); 57 58 LCD String(t1buffer); // Display on LCD 59 LCD SetCursor(2,1); LCD_String(t0buffer); 60 // Display on LCD 61 // Update the display every 500 ms delay ms(5); 62 } 63 } 64 void displayCount(unsigned int count) { char buffer[16]; 65 66 LCD Clear(); sprintf(buffer, "Count: %u", count); // Format the count value 67 LCD String(buffer); // Display on LCD 68 69 } 70 void LCD Init(void) { // Initialize LCD in 4-bit mode 71 LCD Command(0x02); 72 LCD Command(0x28); // 2 lines. 5x7 matrix 73 LCD Command(0x0C); // Display ON, Cursor OFF 74 // Increment cursor LCD Command(0x06); 75 LCD Command(0x01); // Clear display 76 delay ms(2); 77 } 78 void LCD Command(unsigned char cmd) { 79 RS = 0; // Command mode 80 D4 = (cmd >> 4) & 0x01;81 D5 = (cmd >> 5) & 0x01;D6 = (cmd >> 6) & 0x01;82 83 D7 = (cmd >> 7) & 0x01;84 **EN** = 1; 85 _delay_ms(1); EN = 0; // Latch the command 86 D4 = cmd & 0x01;D5 = (cmd >> 1) & 0x01;87 D6 = (cmd >> 2) & 0x01;88 89 D7 = (cmd >> 3) & 0x01;90 EN = 1; delay ms(1); EN = 0;// Latch the command 91 } 92 void LCD_Char(unsigned char data) { RS = 1; // Data mode 93 94 D4 = (data >> 4) & 0x01;95 D5 = (data >> 5) & 0x01;

AMOTECH LABS

Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations



AMOTECH LABS Sharing Enthusiasm for Embedded Systems & Automation.

```
96
      D6 = (data >> 6) \& 0x01;
97
      D7 = (data >> 7) \& 0x01;
      EN = 1; delay ms(1); EN = 0;
98
                                                    // Latch the data
99
100
      D4 = data \& 0x01;
101
      D5 = (data >> 1) \& 0x01;
102
      D6 = (data >> 2) \& 0x01;
103
      D7 = (data >> 3) \& 0x01;
104
      EN = 1; delay ms(1); EN = 0;
                                                    // Latch the data
105 }
106 void LCD_String(const char *str) {
107
      while (*str) {
108
         LCD Char(*str++);
109
      }
110 }
111 void LCD_Clear(void) {
      LCD Command(0x01);
                                                    // Clear display
112
113
      __delay_ms(2);
114 }
115 void LCD SetCursor(unsigned char row, unsigned char column) {
116
      unsigned char position;
      if (row == 1) position = 0x80 + column - 1;
117
118
      else if (row == 2) position = 0xC0 + column - 1;
119
      LCD Command(position);
120 }
121
```



Lab 5 - PWM

Aim: To study the PWM with PIC18F452 mini development kit



.Procedure:

AMOTECH

LABS

- 1. Read the PWM Program on next page with all the comments which explains the program.
- 2. Open the Project folder for the program in MPLAB X IDE or write it by making new project as per MPLAB X IDE procedure explained in the manual. Build and generate the Hex file for the same.
- 3. Open MPLAB IPE and follow the procedure explained to download the Hex file for the above program.
- 4. LCD contrast can be adjusted using 10k pot beside LCD. Once you download the program and remove the programmer, you will see above contents on the LCD.
- 5. Make sure that the arrow side of the programmer must be connected to the RST pin of the ICSP connector of the board.

Embedded/IoT Development | Automation Sevices | Training & Counselling Embedded Development Boards/Kits |PLC & Automation raining Staations www.amotechlabs.com B +91-8329537565 M amotechlabs@gmail.com

28

Program:

```
1 #include<xc.h>
2 #include<pic18F452.h>
3 #include<string.h>
4 // Configuration Bits
5 #pragma config OSC = HS
                               // High-speed oscillator
6 #pragma config WDT = OFF
                               // Watchdog Timer Disable
7 #pragma config LVP = OFF
                               // Low Voltage ICSP Disable
8 #pragma config BOR = OFF // Brown-out Reset Disable
9 #define XTAL FREQ 20000000 // Define system clock frequency (20 MHz)
10 void main()
11 {
12
     unsigned int duty cycle; /* Set CCP1 pin (RC2) as output for PWM */
13
     TRISCbits.TRISC2 = 0;
14
     PR2 = 124; /* Load period value in PR2 register for 100us period*/
15
     CCPR1L = 1: /* Load initial duty cycle */
     T2CON = 0x04; /* Timer2 ON, Prescaler = 4 */
16
17
     CCP1CON = 0x0C; /* Set PWM mode */
18
     TMR2 = 0;
                   /* Clear Timer2 */
19
     T2CONbits.TMR2ON = 1; /* Turn ON Timer2 */
20
     while(1)
21
     {
22
       for(duty cycle = 1; duty cycle < 124; duty cycle++)
23
       {
24
          CCPR1L = duty cycle;
25
            delay_ms(20);
26
       }
27
          delay ms(500);
28
29
       for(duty cycle = 124; duty cycle > 1; duty cycle--)
30
       {
31
          CCPR1L = duty cycle;
32
            _delay_ms(20);
33
       }
34
          delay ms(500);
35
     }
36 }
37
```

