Evaluation kit for Microchip PIC18F67J50 USB MCU with 1.5” 262k OLED color display

Revisions

		Date
First draft		8th Oct 2010

Chapter 1. Introduction

1.1 Overview

PIC18 OLED EVK (Rev 1E) is a development board for Microchip PIC18F67J50 USB microcontroller (MCU) completed with the following peripherals:

- 1.5" 262k OLED color display
- MicroSD socket
- High performance MOSFET switching between battery and USB power
- 16Mbits serial Flash memory
- Tri-color ultra-bright LED
- 5-way joystick with 3x4 numeric keypad wired in matrix
- Buzzer with an amplifier circuit
- Battery level measurement circuit
- 32.768kHz clock crystal connected to Timer1 external clock input for Real-Time-Clock applications
- 2x22 2.00mm expansion port for application modules

1.2 Packing List

Every kit must contain the following components. Please contact us at john@techtoys.com.hk if you find anything missing from the package

- Evaluation kit
- Adapter for RJ11 6-pin socket to 2.54mm 6-pin receptacle for ICD2/3 or PICKit2/3 online debug
- USB cable (A to mini-B)
- CD-ROM containing all source code and projects

1.3 Operational Requirements

To communicate with and program the evaluation kit, the following hardware and software requirements must be met:

- PC compatible system running Microsoft Windows XP SP2 or above
- One available USB port on the PC or a powered USB hub
- MicroSD flash memory card of 2GB or smaller
- 3 pieces AAA alkaline battery
- 1 piece CR1220 coin cell
- Microchip ICD2 or ICD3. This is an optional item for project development at the source code level.
- PICKit2 or PICKit3 debugger. This is similar to Microchip ICD2/3. Further information about Microchip ICD and PICKit debugger is available from www.microchip.com

1.4 Initial Board Setup

Every evaluation kit has been preloaded with a Bootloader program and an associated demonstration program (section 2.3.1e) for quality assurance. Basically you will require only 3 pieces AAA alkaline battery to power-up the kit and view a demonstration out-of-the-box.

1.4.1 Installing the Software

All software for this evaluation kit has been prepared under the following environment:

- MPLAB Integrated Development Environment (IDE) version 8.30 or above, which can be downloaded from this hyperlink.

  http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469&part=SW007002

  The latest IDE version is v8.56 at time of writing this manual. By following the installation procedure for default settings, MPLAB will be installed under ***C:\Program Files\Microchip\MPLAB IDE***, which is also the same environment we are using.

- MPLAB C complier (MPLAB C18) for PIC18 MCUs version 3.30 or above, which can be downloaded from this hyperlink.

  http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en010014

  There are LITE, Standard-Eval, and Upgrade versions available. The Stardard-Eval version is a time-limited evaluation edition of the C Complier for 60 days. Registration is required prior to download. For 60 days, the Standard Evaluation Edition of MPLAB C Complier for PIC18 MCUs will function as the full version. After 60 days, the compiler differs from the full version in some optimizations and extended mode removed. All examples provided with PIC18 OLED EVK work for both versions, before and after that 60 days period.
  Be following the installation procedure for default settings, MPLAB C18 will be installed under ***C:\MCC18***, being the same environment we are using.

- Microchip Application Libraries version 2010_04_28 or above. The main page of the libraries is found from this web page.

  http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2680&dDocName=en547784

  The Microchip Application Libraries includes full source code for the following software libraries: USB, Graphics, Memory Disk Drive, TCP/IP Stack, mTouch Capacitive Touch, and Smart Card Library. This is a big library collection of 15MB for not only PIC18 MCU series, but also other MCU series such as sdPIC and PIC32. We won't use all of them but only few projects for PIC18 USB device series.

1.4.2 Connecting the hardware

The evaluation kit is connected to a PC by the provided USB cable (A to mini-B) for the following scenarios:

- Download another hex file via Bootloader program
- Experiment on USB device, e.g. USB Mass Storage, HID, etc.
- Supply power from a USB port instead of batteries

Online debug via an ICD2/3 or PICKit2/3 is allowed. A PICKit2/3 debugger is connected directly to J1. No adapter is required.

Just in case you would like to use an ICD2/3, a 6-pin 2.54mm ICSP interface is provided.

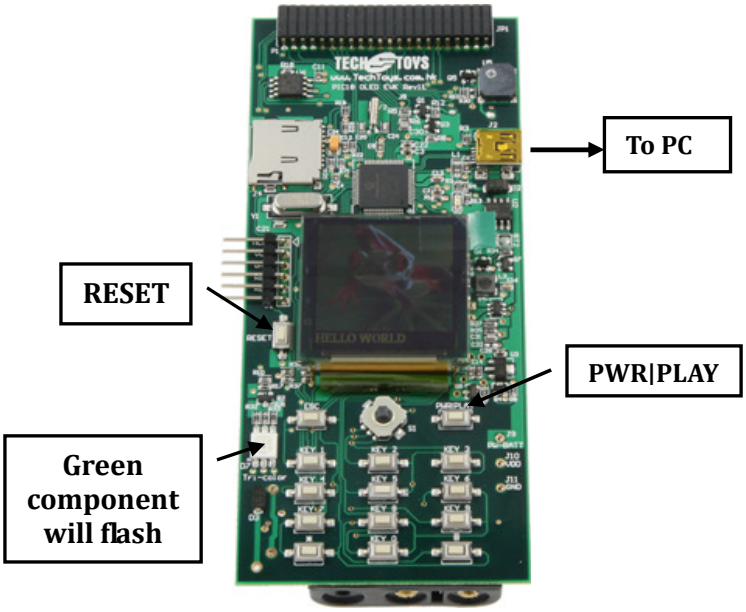
Chapter 2. The Demonstration Application

2.1 Download a program by Bootloader

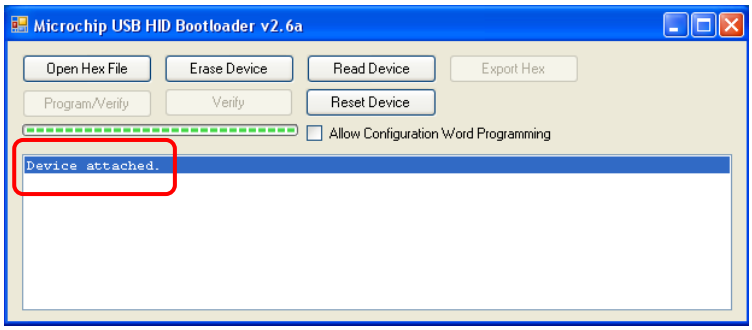
The Flash program memory of Microchip's PIC18F67J50 device is readable, writable and erasable during normal operation over the entire VDD (3.3V in our case) range. This feature gives users the ability to perform bootloader operations. Further information about bootloader can be found from an application note AN1310 (http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1824&appnote=en546974) provided by Microchip. Because there is no non-erasable region for Bootloader program, every PIC18 OLED EVK board is manually programmed a Bootloader by an external programmer during quality assurance. As a result, the Bootloader itself is a piece of program that may be erased if you have the proper programming tool such as ICD3 or PICKit3. Special attention should be paid for this regarding your own program design. More details about this can be found at section 2.2 and section 2.3.1a of this manual.

To bring up the Bootloader, first connect a USB cable to PC. Launch the executable *HIDBootLoader.exe* that is included under the folder:
CDROM:\ Firmware\USB Device - Bootloaders\HID – Bootloader

Hold the key ***PWR|PLAY*** and press-&-release ***RESET***.



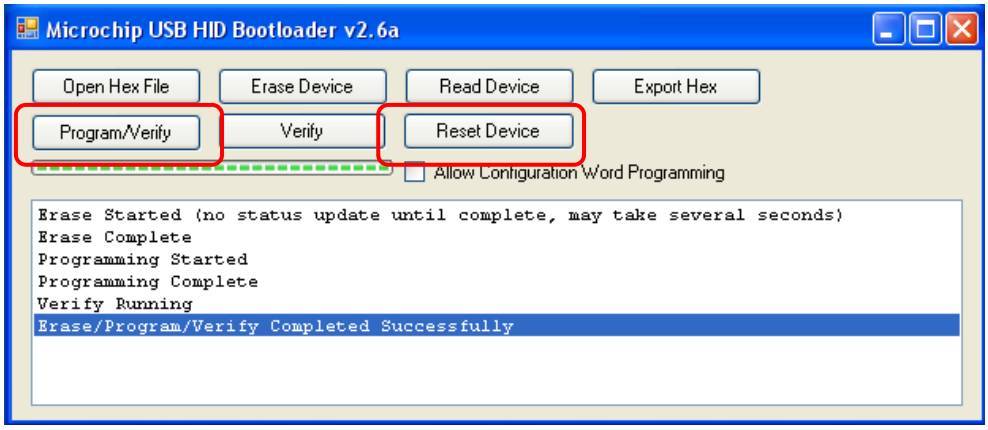
Upon successful connection, the message box of HIDBootLoader.exe will show "Device attached" and the green component of the tri-color LED will flash. These are the signatures for a successful connection. Now it would be possible to browse a hex file and download.



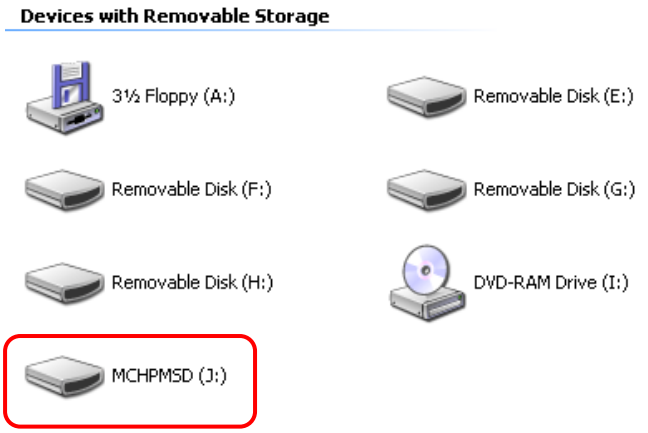
2.1.1 Reprogram the board for USB Mass Storage – SD Card Reader

This section shows you the procedure to download a hex code for USB Mass Storage with Bootloader.

i. Prepare a microSD card of capacity smaller than 2GB. Format the card by FAT16 or FAT32. Copy few files to it, preferably files of smaller size. Plug this card to MicroSD receptacle on PIC18 OLED EVK.
ii. Connect a USB cable from PIC18 OLED EVK to a PC. Battery is optional on PIC18 OLED EVK.
iii. Launch *HIDBootLoader.exe from the folder CDROM:\ Firmware\USB Device - Bootloaders\HID – Bootloader.* You may have to copy the CD-ROM content to a convenient directory for speed.
iv. Hold ***PWR/PLAY*** key, press and release ***RESET***. The message box on USB HID Bootloader will change to "Device attached".
v. Click Open Hex File, browse to folder at *CDROM:\ Firmware\ USB Device - Mass Storage - SD Card reader\Mass Storage - SD Card Reader – Firmware.* Click Program/Verify.
vi. After step v finished, click Reset Device from the Bootloader program.



vii. Check from your PC that a new drive is available from My Computer. Try few operations on it to verify everything works fine like file create, delete, etc.

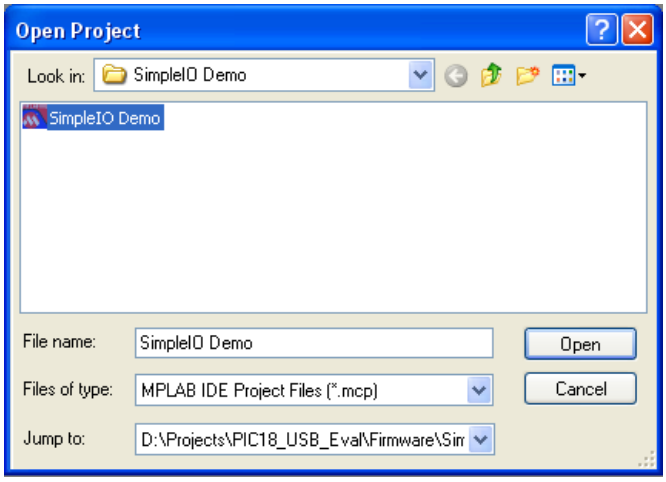


2.2 Online debug and program development

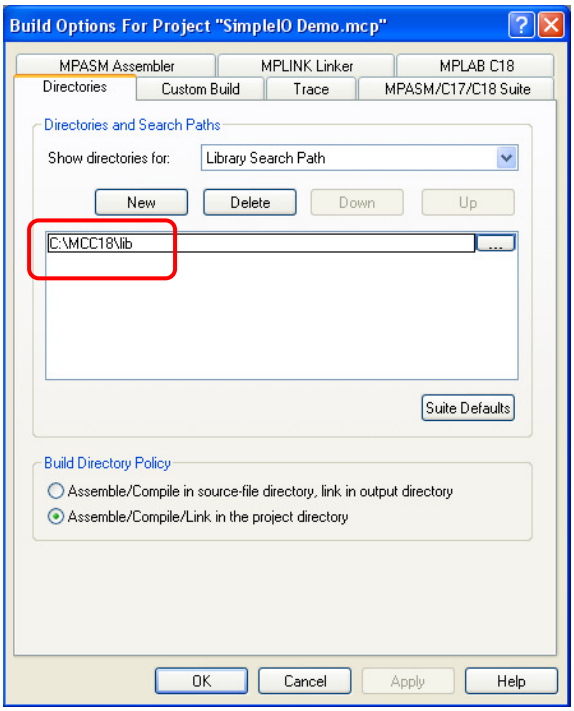
If you have an ICD2/3 or a PICKit2/3 debugger, online program debug with break points is possible. Please note that once the MCU got programmed by an external programmer (PICKitx or ICDx), the embedded Bootloader firmware will be erased. However, it is always possible to reload the Bootloader firmware by following the procedure described in section 2.3.1a.

2.2.1 Online debug the first program by a PICKit3 / ICD3

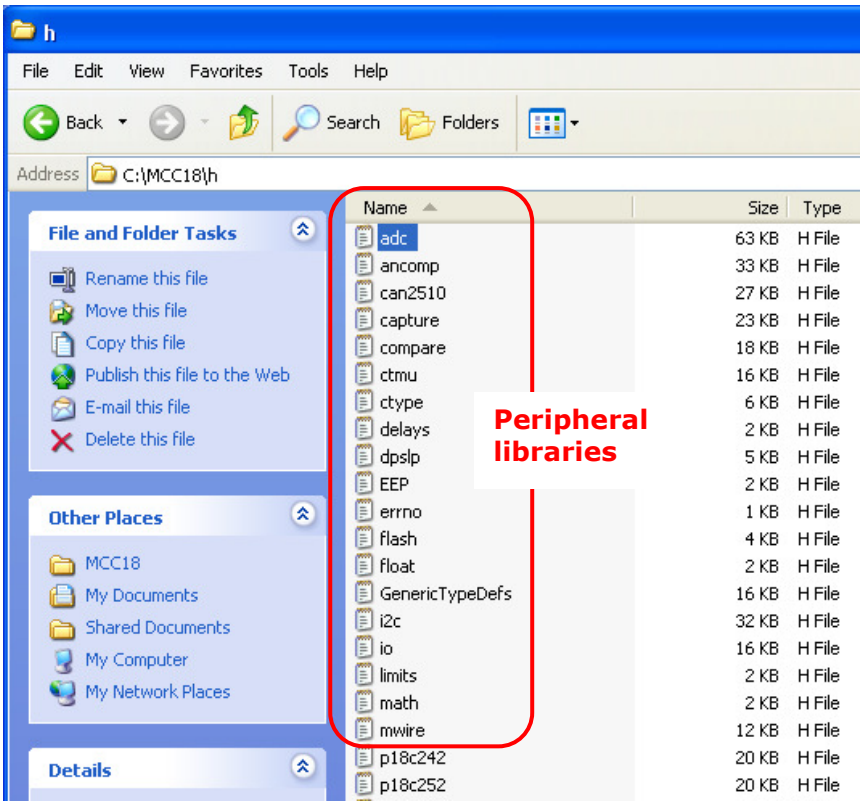
2.2.1.1 Launch MPLAB. Under Project →Open, browse to the folder ..\Firmware\SimpleIO\SimpleIO Demo and open the project SimpleIO Demo.mcp.



2.2.1.2 Make sure there is no "missing file" error message. This error stems from a different development environment or settings. My path of installation for C18 is located under C:\MCC18. If it happens to be a different path in your PC, MPLAB may not be able to locate some of the files necessary for compilation. Open the Build Options box under Project→Project Options to make sure the Library Search Path matches with your own development environment; otherwise, the program will not compile. There are standard libraries included with MPLAB C18 for basic MCU features. By browsing C:\MCC18\h you will find all those p18xxx.h header files and peripheral libraries, such as adc.h, ctmu.h, delays.h, to name few of those. This is also why the path



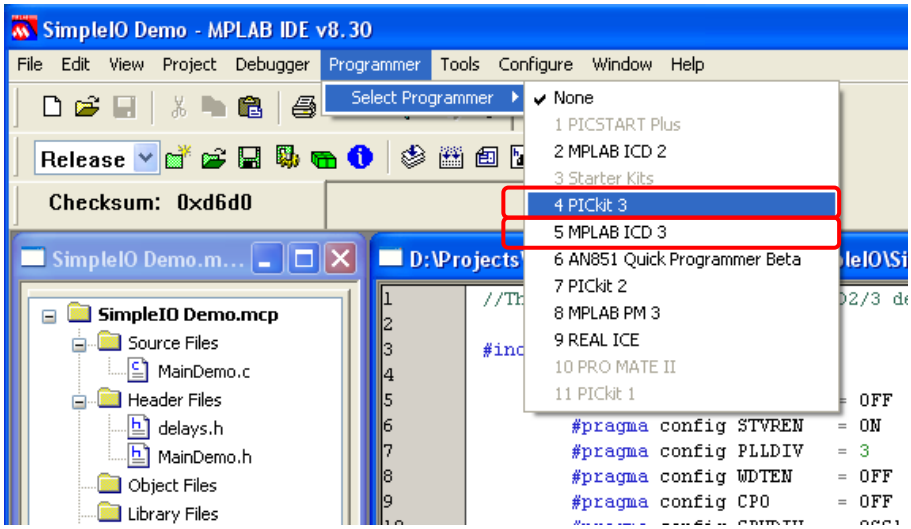
C:\MCC18\lib has been included in the Library Search Path option because I would like to use simple software delay in the demo.



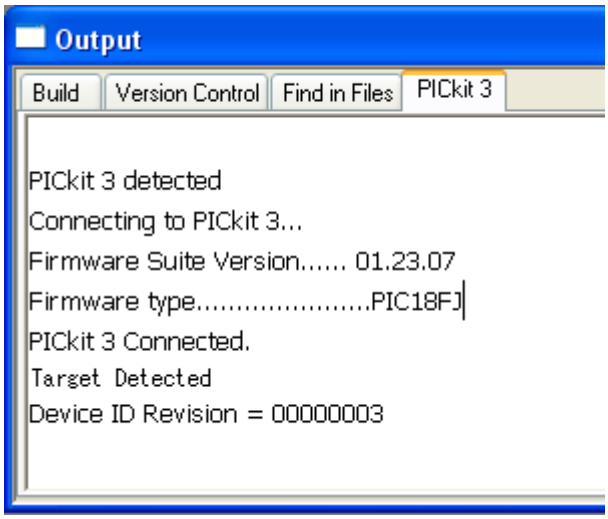
2.2.1.3 Connect the hardware accordingly. An adapter for ICD3↔2.54mm pin header is included if an ICD2/3 is used.

2.2.1.4 Make sure *HIDBootLoader.exe* is not running if you are using a USB cable to supply power to PIC18 OLED EVK kit. For some unknown reason, HID Bootloader connection cannot co-exist with PICKit3. It is either PICKit3 or HID Bootloader connected to the board at any time. Nevertheless, ICD3 does not have such restriction. It is also possible to supply power by batteries, of course. Once "Target Detected" in MPLAB, the Bootloader will be disconnected automatically (see 2.2.1.5 below).

2.2.1.5 Under Programmer→Select Programmer, select PICKit3 (or ICD3 if you are using an ICD3).



Upon a successful connection, the Output message will show this message.

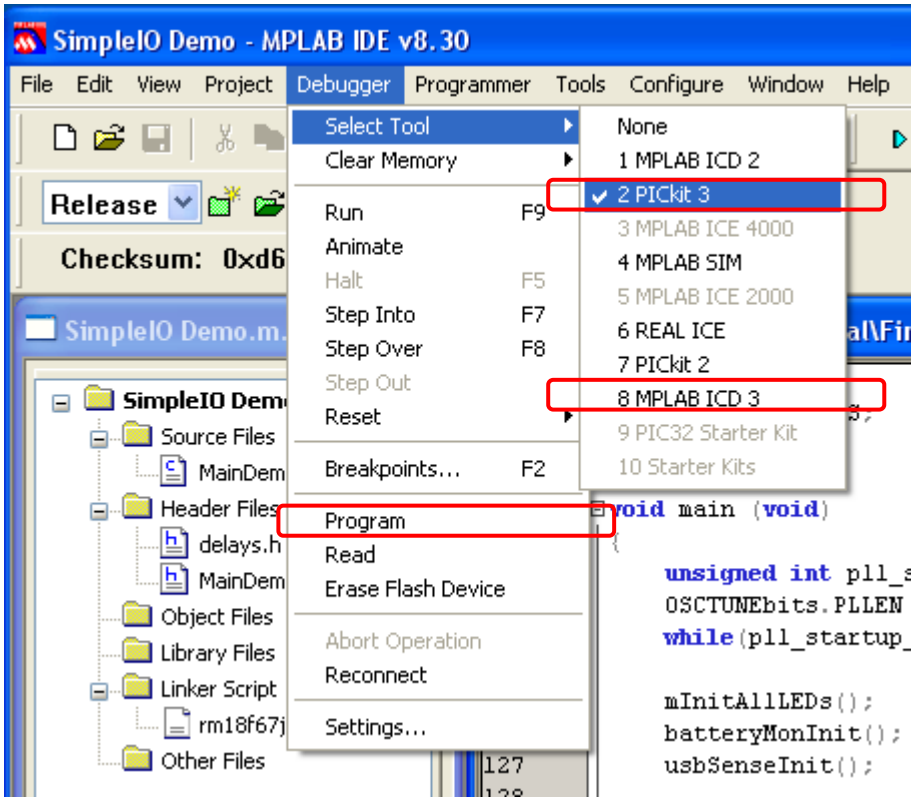


Just in case there is annoying beep from the buzzer upon connection, reboot the MCU by holding ***PWR/PLAY*** key and then press-&-release ***RESET*** to bring up the Bootloader firmware. The reason of the beep noise is that the built-in OLED display demo is using pins RB7 and RB6 for keypad. These two pins multiplex with

the in-circuit programming header. A connection via these pins cause virtual key-press action which triggers audio beep from the buzzer. To get around this problem, we simply run the Bootloader firmware by PWR|PLAY and RESET therefore pins RB6&7 no longer occupied.

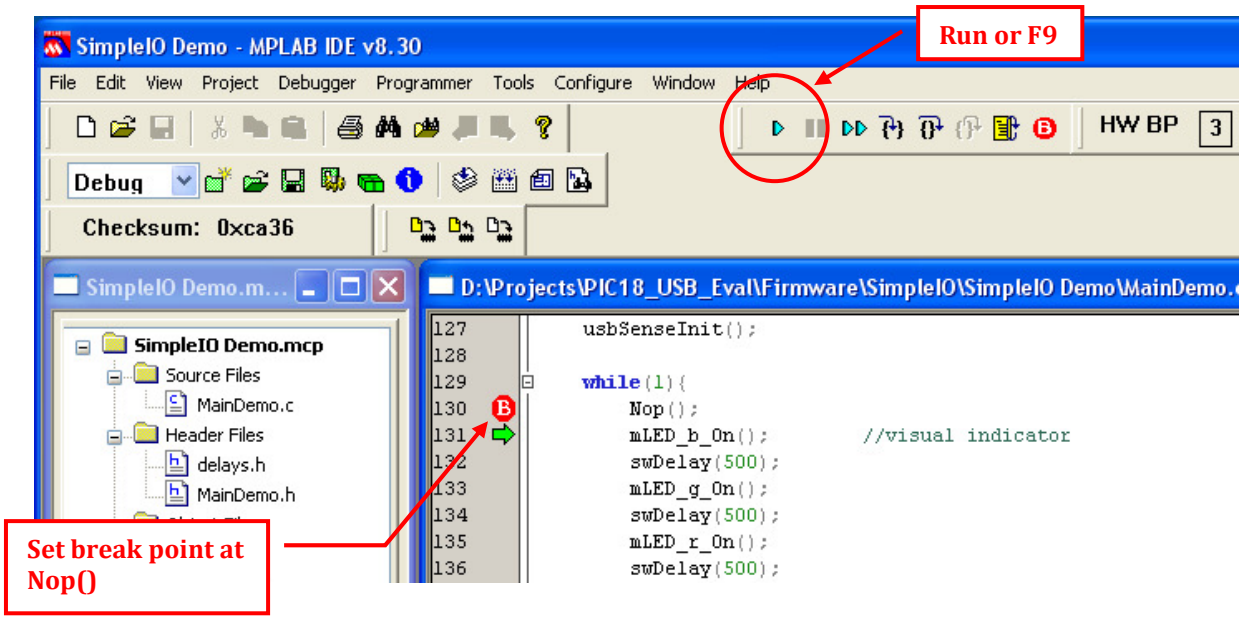
2.2.1.6 Click *Program* from Programmer→Program to load the board with this simple IO demo to the MCU. Individual color on the tri-color LED will blink with option of hauling by holding PWR|PLAY key or a USB cable plug-in.

2.2.1.7 Finally, let's try single stepping the program with break points. First, unload the PICKit3 from *Programmer→Select Programmer→None*. MPLAB doesn't allow double selection of programmer and a debugger. Under *Debugger→Select Tool*, select *PICKit3 (or ICD3).*



After connection, click *Program*. MPLAB will prompt you for a rebuild. Go ahead to rebuild.

Set a break point by double clicking on the gray border at Nop() (line 130). Click *Run (F9)* and *Step Over (F8)*. Upon each F8 pressing, the program pointer will step down the program. Observe the LED reaction as the program is stepped over.



2.3 Using the Microchip Application Libraries

A nice thing to use Microchip's products is that we don't write software. There is no need to start everything from scratch. We just copy and modify from existing software libraries for our hardware. Application notes and software libraries are provided under the main web page of Microchip PIC18F67J50 at this hyperlink.
http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en027177

This section describes the procedure to use the Application Libraries for PIC18 OLED EVK hardware.

2.3.1 Microchip Application Libraries

Microchip has USB software to support USB on her complete MCU series. The software is royalty free source code with example projects. Just to name few of those, demos include device CDC demo, printer demo, device composite HID and mass storage, generic driver demo, etc.

The original Application Libraries is a big collection of libraries for not only PIC18 MCU series, but also PIC24F, PIC24H, dsPIC, and PIC32. At time of writing, the latest library version is v2010-08-04. We will use only few of the libraries and modify them for our own hardware. All files necessary for us were copied to the accompany CD-ROM. It is not a strict requirement to download the original libraries from Microchip. However, because the original libraries contain numerous documentations and help files, it would be recommended to keep them as references.

Microchip Application Libraries download includes the following:

Library	Current Version	PIC18F (8-bit)	PIC24F (16-bit)	PIC24H (16-bit)	dsPIC (16-bit)	PIC32 (32-bit)
USB Framework	2.7a	x	x			x
Graphics Library	2.10		x	x	x	x
Memory Disk Drive (MDD)	1.2.6	x	x	x	x	x
TCP/IP Stack	5.25	x	x	x	x	x
mTouch Capacitive Touch Library	1.20	x	x			
Smart Card Library	1.01	x	x			

Microchip Application Libraries

Downloads

Microchip Application Libraries v2010-08-04

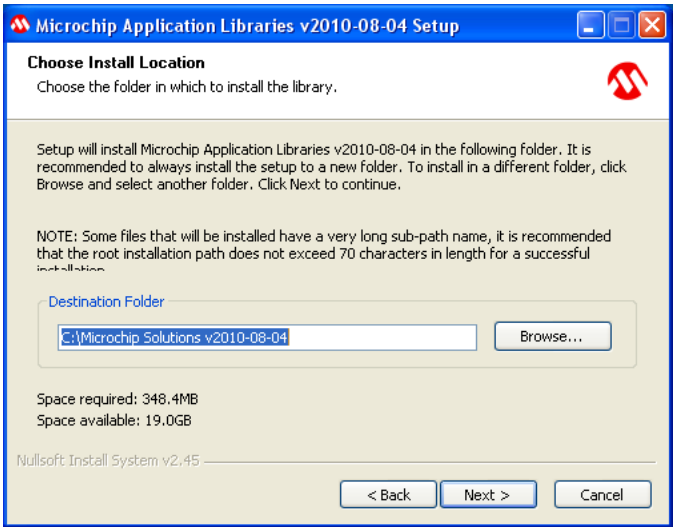
We are using not more than 1/10 of the libraries!

Downloads

Title	Date Published	Size	D/L
Microchip Application Libraries Release Notes		8 KB	
Microchip Application Libraries Help Files		14593 KB	

This is the zipped file to download

After download and extract the 85MB zipped file, an installer will be created. Double-click to install by following the default options. A folder will be created under *C:\Microchip Solutions v2010-08-04*. Help files in html format is available at folder at *C:\Microchip Solutions v2010-08-04\Microchip\Help*. More documentation is located at *C:\Microchip Solutions v2010-08-04\Microchip\USB\Documentation\Getting Started*.



2.3.1a Bootloader

Demo location - *CD-ROM:\Firmware\USB Device - Bootloaders\HID - Bootloader\HID Bootloader - Firmware for PIC18F87J50 Family Devices*

This demo was modified from the original Bootloader at *C:\Microchip Solutions v2010-08-04\USB Device – Bootloaders .*

Prior to starting, please read "*Getting Started - Using the Device - USB Device – Bootloaders.htm"* at

CD-ROM:\Firmware\USB Device - Bootloaders\HID – Bootloader
OR
C:\Microchip Solutions v2010-08-04\Microchip\USB\Documentation\Getting Started if you have downloaded the original Microchip ApplicationLibraries.

The following modifications have been done to use this Bootloader:

i. Add #define PIC18_OLED_EVK in usbcfg.h
ii. Add PIC18_OLED_EVK in io_cfg.h and match LED macros with PIC18 OLED EVK. Listings for all changes summarized below:

```
…
#if defined (PIC18_OLED_EVK)                                          (1)

#define tris_usb_bus_sense  TRISBbits.TRISB1                          (2)

#if defined(USE_USB_BUS_SENSE_IO)
#define usb_bus_sense       PORTBbits.RB1
#else
#define usb_bus_sense       1
#endif

#define self_power          0                                         (3)
//Add your hardware specific I/O pin mapping here
#define mLED_1              LATCbits.LATC3                            (4)
#define mLED_2              LATCbits.LATC7                            (5)

#define mLED_1_On()         mLED_1 = 0;
#define mLED_2_On()         mLED_2 = 0;

#define mLED_1_Off()        mLED_1 = 1;
#define mLED_2_Off()        mLED_2 = 1;

#define mLED_1_Toggle()     mLED_1 = !mLED_1;
#define mLED_2_Toggle()     mLED_2 = !mLED_2;

#define mInitAllLEDs()      {TRISC= 0b01110111; \
                             TRISAbits.TRISA5 = 0; LATAbits.LATA5 = 0;} (6)
```

Listing (1) tests if PIC18_OLED_EVK hardware is being used. The same statement will be repeated throughout main.c to notify the compiler, that it is a piece of different hardware to standard demo board.

Listing (2) defines the input for V_{BUS} notification. It is RB1 on PIC18 OLED EVK. Please check the schematic in section 3 for details.

Listing (3) defines self_power to zero to avoid compilation error. We have to keep it for usb9.c.

Listing (4-5) defines the major difference for hardware. Visual indicator is allowed with the application. There is a tri-color LED onboard for this. The green, red, and blue components are wired to RC3, RC7, and RC6 respectively with the current source controlled by RA5. Right now it is the green and red component chosen for blinking upon successful Bootloader connection. You may try to change this for different visual effect.

Listing (6) marks a major difference for PIC18 OLED EVK with Microchip's demo board. First, it is required to match pin RC3 and RC7 with TRISC= 0b01110111. If it is going to be RC7 and RC6 for Listings 4-5, we need to change TRSIC to a binary value '0b00111111' for outputs. Second, because there is a source current control by RA5, we have to set it an output low to open the current source. Here comes the syntaxes TRISAbits.TRISA5 = 0 and LATAbits.LATA5 = 0.

iii. Finally, the source code main.c was modified for PIC18_OLED_EVK. Search the keyword "PIC18_OLED_EVK" over main.c it is not surprised to find just slight modification has been made next to PIC18F87J50_FS_USB_PIM, which is the original hardware for a Microchip's module.

Feel free to make changes for a different visual effect by using different LED colors under io_cfg,h.

Upon a successful program compile, follow this procedure to reload the Bootloader:

1. Install batteries and remove any USB cable connected to PIC18 OLED EVK.
2. Connect a PICKit3 or ICD3 programmer and program the MCU.
3. Unload the programmer from Programmer menu with Programmer→Select Programmer→None.
4. Remove PICKit3/ICD3.
5. Finally connect a USB cable to PIC18 OLED EVK and launch *HIDBootLoader.exe* from CD-ROM:\Firmware\USB Device - Bootloaders\HID – Bootloader.

2.3.1b USB HID Joystick Demo

Demo location - *CD-ROM:\Firmware\USB Device - HID - Joystick\Firmware,* and the name of project is

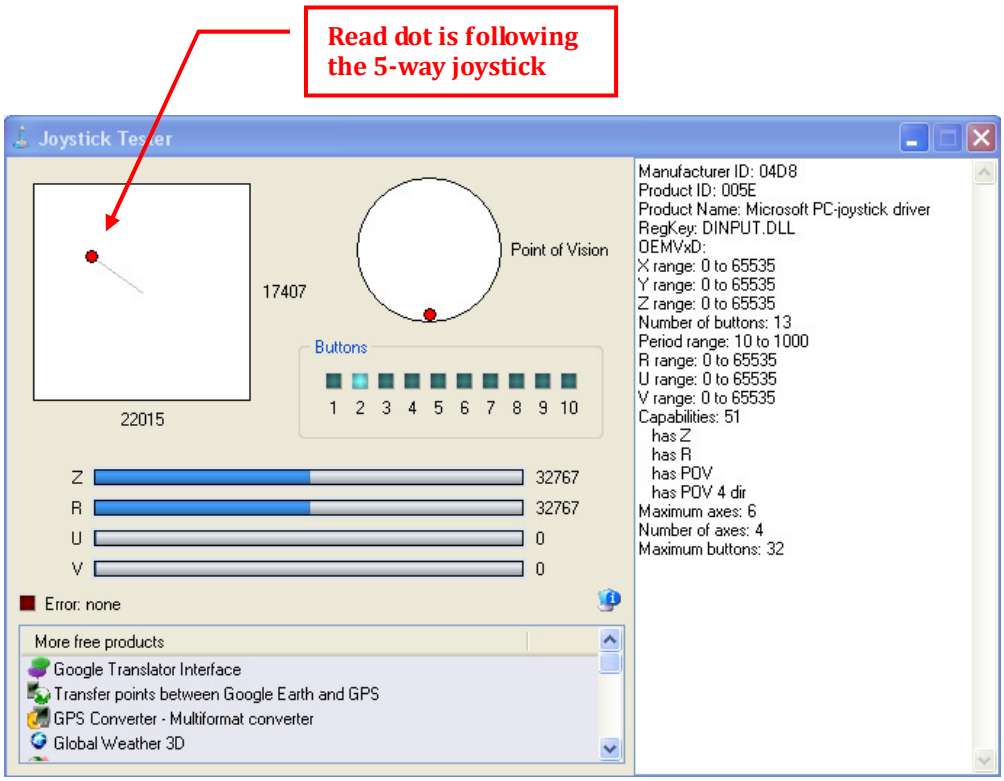
USB Device - HID - Joystick - C18 - PIC18F67J50 OLED EVK.mcp

This demo was modified from the original joystick demo for PIC18F87J50 PIM at *C:\Microchip Solutions v2010-08-04\USB Device - HID - Joystick.*

The same joytester.exe is used for testing the joystick function.

If you are using a Bootloader, you may directly download the hex code "*USB Device - HID - Joystick - C18 - PIC18F67J50 OLED EVK.hex*" which is located at the project directory. Follow the same procedure as "***HOLDING PWR|PLAY***" key and "**PRESS RESET**". Connect PIC18 OLED EVK to *HIDBootLoader.exe,* browse to *USB Device - HID - Joystick - C18 - PIC18F67J50 OLED EVK.hex* and click *Program/Verify*.

Reset the board and launch joytester.exe. Now you will be able to adjust the position of the red dot by the 5-way joystick onboard.



Turning to the source code, under the workspace there are only three files to be modified although the whole USB stack looks daunting.

1. HardwareProfile.h :
   A new definition for PIC18 OLED EVK has been added under __18CXX section.

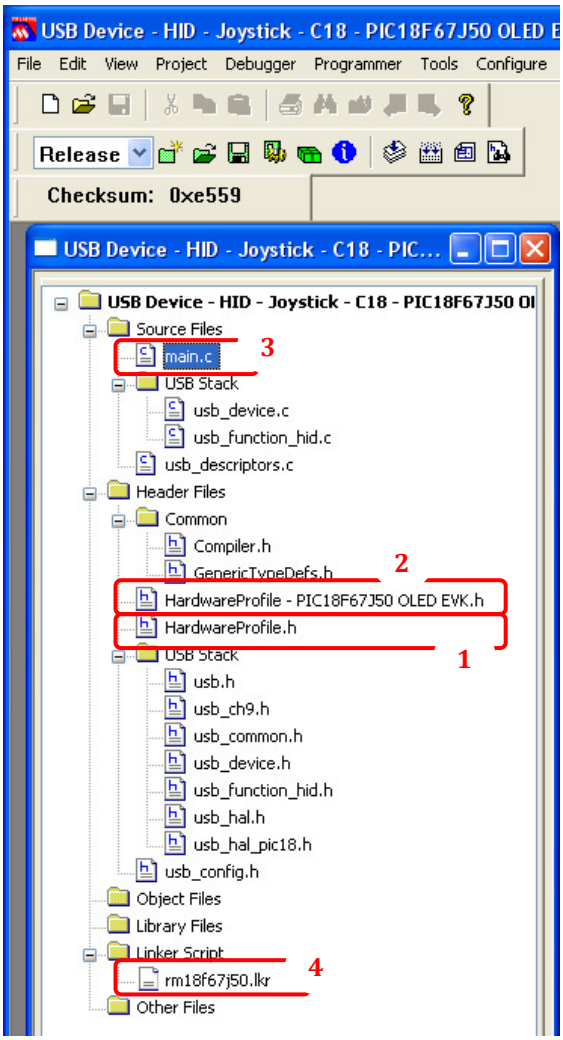
   #include "HardwareProfile - PIC18F67J50 OLED EVK.h"

2. HardwareProfile-PIC18F67J50 OLED EVK.h :
   This header file specifies if the program is compiled for Bootloader or an external programmer, pinouts, clock frequency, as well as the keyword PIC18_OLED_EVK as a signature of our hardware in main.c

3. main.c :
   Add new hardware side-by-side to PIC18F87J50_PIM making use of the keyword PIC18_OLED_EVK. Modifications to sw3 to sw_left and joystick(void) function also made for different response to joytester.exe.

4. rm18f67j50lkr :
   Not really any change to the original linker script but only the name change from p18f87j50 to p18f67j50 as they share the same memory footprint.

   These are the only changes made to fit our particular hardware design. The same method applies to examples below.



2.3.1c USB MSD Card Reader Demo

Demo location - *CD-ROM:\Firmware\USB Device - Mass Storage - SD Card reader\Mass Storage - SD Card Reader - Firmware,* and the name of project is

USB Device - Mass Storage - SD Card reader - C18 - PIC18F67J50 OLED EVK.mcp

If you are using a Bootloader, you may directly download the hex code "*USB Device - Mass Storage - SD Card reader - C18 - PIC18F67J50 OLED EVK.hex*" which is located at the project directory. Follow the same procedure as "***HOLDING PWR|PLAY***" key and "**PRESS RESET**". Connect PIC18 OLED EVK to *HIDBootLoader.exe,* browse to *USB Device - Mass Storage - SD Card reader - C18 - PIC18F67J50 OLED EVK.hex* and click *Program/Verify*. This procedure is the same as that of section 2.1.1 so I am not repeating here.

There is nothing I have touched on the USB Stack and MDD File System. Similar to previous demo, the only files to be modified are *HardwareProfile.h*, a new header *HardwareProfile-PIC18F67J50 OLED EVK.h,* and slight modification to *main.c* and *rm18f67j50_SD_Reader.lkr* linker script.

2.3.1d USB CDC Demo

Demo location - *CD-ROM:\Firmware\USB Device - CDC - Basic Demo\CDC - Basic Demo - Firmware,* and the name of project is

USB Device - CDC - Basic Demo - C18 - PIC18F67J50 OLED EVK.mcp

If you are using a Bootloader, you may directly use the hex code "*USB Device - CDC - Basic Demo - C18 - PIC18F67J50 OLED EVK.hex*"

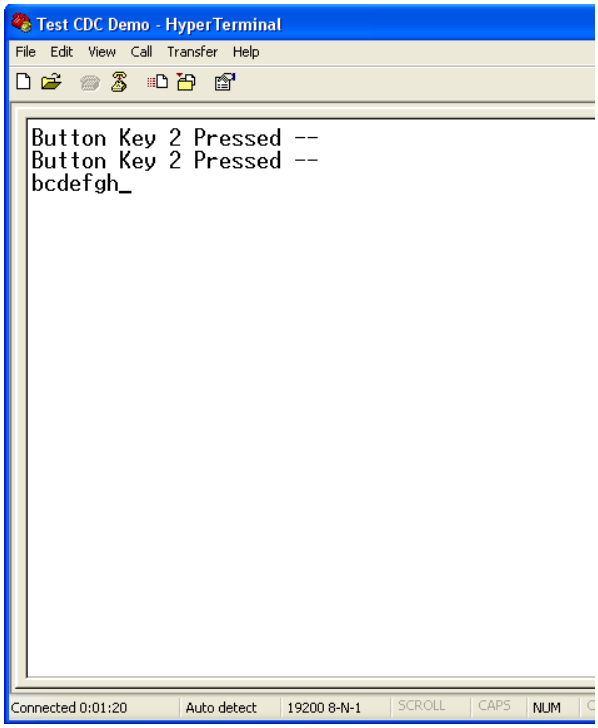
This demo allows the device to appear like a serial (COM) port to the host. In order to run this demo first compile and program the PIC18 OLED EVK with either Bootloader or PICKit3/ICD3. Then, attach the device to the host. If the host is a PC and this is the first time you have plugged this device into the computer then you may be asked for a .inf file located at the path *CD-ROM:\Firmware\USB Device - CDC - Basic Demo\inf.*

The baud rate is 19200bps 8-n-1 with no flow control. Use any COM port program like HyperTerminal. There are two tests to performed:

1. Press key 2 from PIC18 OLED EVK's keypad. HyperTerminal will respond with a string "Button Key 2 Pressed –"
2. Input any key from the PC's keyboard. PIC18 OLED EVK will echo the input with ASCII+1, therefore a letter 'a' will echo a letter 'b' and so on.

Below shows a screen shot as I pressed twice on PIC18 OLED EVK's key 2 and clicked 'abcdefg' from the PC's keyboard.

It is the function void ProcessIO(void) to specify such string. If you need to change this text, go ahead to ProcessIO(void) under main.c



2.3.1e Displaying primitive graphics on 1.5" OLED color display

Demo location - *CD-ROM:\Firmware\OLED Keypad Demo,* and the name of project is *OLED Keypad Demo.mcp*

If you are using a Bootloader, you may directly use the hex code "*OLED Keypad Demo.hex*" to view this demo.

User interaction is allowed to show different primitives including rectangles, bevels, arcs, picture and font drawing on the OLED display with different click action to the 5-way joystick.

Table below summarizes the key action for individual display on OLED

5-way joystick	Numberic keypad	Display
Left		Line demo
Center down		Arc demo
Down		Circle demo
Right		Bevel demo
Up		Clear screen with pure yellow
	ESC	Fill bevel
	*	Print "HELLO WORLD"
	#	Put an image on OLED
	1	Bar demo
	PWR|PLAY	Long press to put the device in sleep mode, draining 20-30uA

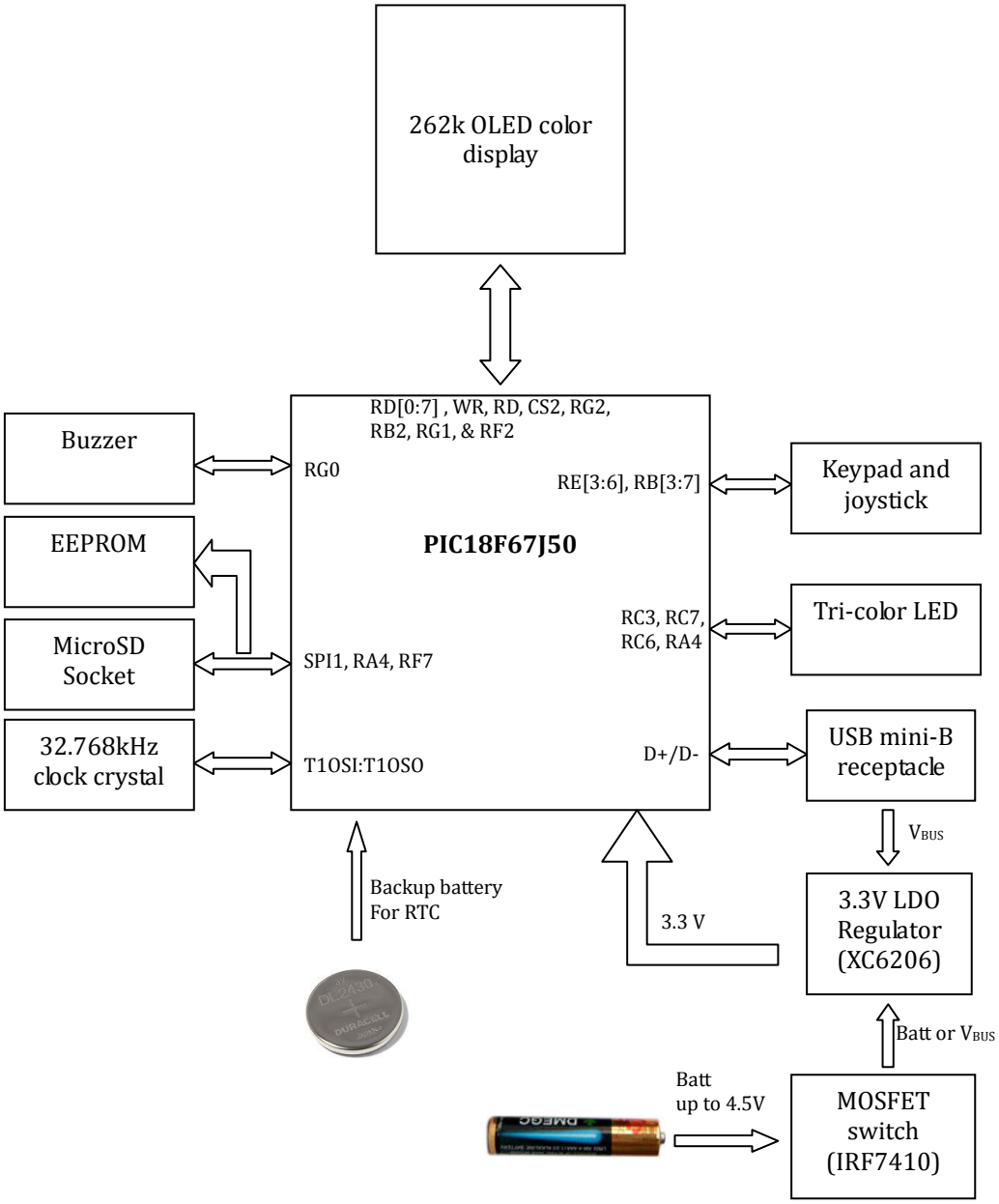
Because the original Graphics library provided by Microchip does not support C18 devices, a custom made primitive.c and SSD1355.c driver have been developed and all source code is included in the accompany CD-ROM.

The full source code is included in the folders *CD-ROM:\Firmware\OLED Driver and CD-ROM:\Firmware\keypad Driver* for the OLED and keypad drivers, respectively.

Chapter 3. Hardware

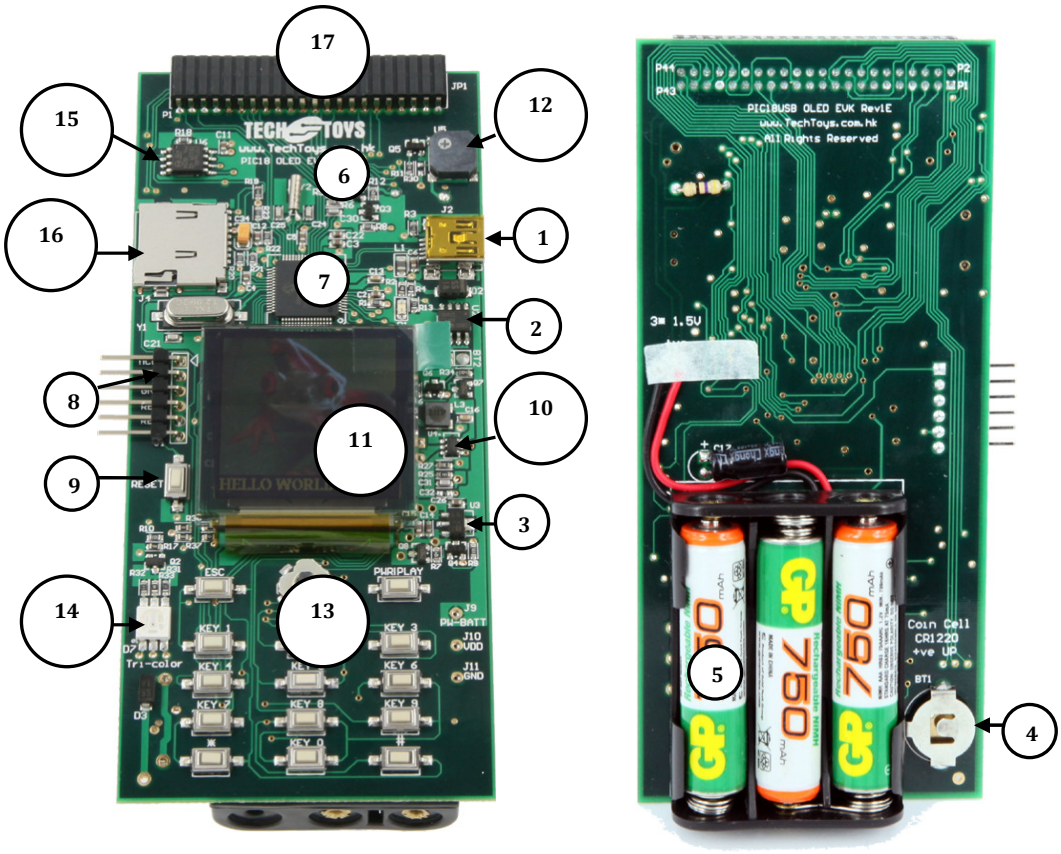
This chapter provides an overview of the PIC18 OLED EVK hardware.

3.1 Block Diagram



3.2 Board Layout

Figure below identifies the key hardware components for the kit.



Ref	Component	Schematic	Designator
1	USB Mini-B receptacle	Page 1	J2
2	IRF7410 MOSFET		U2
3	XC6206-3.3V LDO		U3
4	CR1220 coin cell holder		BT1
5	3*AAA battery holder		BT2
6	Test Point for battery voltage		J5
7	PIC18F67J50 MCU	Page 2	U1
8	2.54mm 6-pin header for ICSP		J1
9	Reset Key		SW1
10	FP6736 boost DC-DC	Page 3	U4
11	OLED display		J3
12	Buzzer	Page 4	U5
13	Keypad & 5-way joystick		S1 & SW[2:15]
14	Tri-color LED		D7
15	16Mbits EEPROM		U6
16	MicroSD socket		J4
17	22x2 2.00mm receptacle	Page 5	JP1

3.3 Map of connection

Pin #	Name	OLED	Peripherals					Expansion port
			Keypad	Buzzer	Tri-color LED	MicroSD	EEPROM	
1	RE1/PMWR	√						√
2	RE0/PMRD	√						√
3	RG0/ECCP3			√				√
4	RG1/TX2	√						
5	RG2/RX2	√						
6	RG3/PMCS1 /P3D							√
7	MCLR	MCLR is connected to only J1 for MCU reset						
8	RG4/PMCS2	√						√
9	VSS	GND						
10	VDDCORE	Connected 10uF ceramic capacitor C13						
11	RF7/SS1						√	√
12	RF6/AN11							√
13	RF5/AN10							√
14	D+	To USB Mini-B receptacle only						
15	D-	To USB Mini-B receptacle only						
16	RF2	√						
17	VUSB	To VDD (3.3V) bypassed with 1uF and 0.1uF ceramic capacitors						
18	ENVREG	To VDD (3.3V) bypassed with 1uF and 0.1uF ceramic capacitors						
19	AVDD	To VDD (3.3V) bypassed with 1uF and 0.1uF ceramic capacitors						
20	AVSS	GND						
21	RA3/Vref+							√
22	RA2/AN2	AN2 is connected to a potential divider for battery measurement						
23	RA1/AN1							√
24	RA0/AN0							√
25	VSS	GND						
26	VDD	To VDD (3.3V) bypassed with 0.1uF ceramic capacitor						
27	RA5/AN4				√			√
28	RA4					√		√
29	T1OSI	To 32.768kHz crystal for Real-Time-Clock applications						
30	T1OSO	To 32.768kHz crystal for Real-Time-Clock applications						
31	RC6/TX1				√			√
32	RC7/RX1				√			√
33	RC2/ECCP1					√		√
34	RC3/SCK1					√	√	√
35	RC4/SDI1					√	√	√
36	RC5/SDO1					√	√	√
37	RB7/PGD		√	RB7/PGD is used for ICSP port J1 as well				
39	CLKI	12 MHz crystal						
40	CLKO	12 MHz crystal						
41	VSS	GND						
42	RB6/PGC		√	RB6/PGC is used or ICSP port J1 as well				
43	RB5/KBI1		√					√
44	RB4/KBI0		√					√
45	RB3/INT3		√					√
46	RB2/INT2	√						√
47	RB1/INT1	INT1 is also used to detect USB cable insert for a low-to-high transition						√
48	RB0/INT0	√	INT0 is connected solely to SW2 for a stand-alone tact switch					√
49	RD7/PMD7	√						√
50	RD6/PMD6	√						√

3.3 Map of connection (continued)

Pin #	Name		Peripherals					Expansion port
		OLED	Keypad	Buzzer	Tri-color	MicroSD	EEPROM	
51	RD5/PMD5	√						√
52	RD4/PMD4	√						√
53	RD3/PMD3	√						√
54	RD2/PMD2	√						√
55	RD1/PMD1	√						√
56	VSS	GND						
57	VDD	VDD (3.3V)						
58	RD0/PMD0	√						√
59	RE7/ECCP2	RE7 is also connected to Q3 for battery measurement trigger						√
60	RE6		√					
61	RE5		√					
62	RE4/P3B		√					√
63	RE3/P3C		√					√
64	RE2/PMBE	√						√