PIC24F Starter Kit

User's Guide

Table of Contents

Chapter 1. Overview 1
1.1 Overview 1
1.2 Operational Requirements 2
1.3 Quick Start 2
1.3.1 Installing the Software 2
1.3.2 Connecting the Hardware 2
Chapter 2. The Demonstration Application 4
2.1 Features used throughout the demo application.................... 4
2.1.1 Interactive Display System 4
2.1.2 Five Push Button Switches 4
2.1.3 Time and Date (RTCC) 5
2.2 Specific Demo Highlights 5
2.2.1 RGB LED Control (Three PWMs and Peripheral Pin Select).................... 5
2.2.2 USB Flash Drive Interface (USB Embedded Host) 6
2.2.3 Real-Time Data Graphing (A/D and Display Multitasking).................... 6
2.2.4 Real-Time Data Capture (Multitasking with USB Embedded Host).................... 7
2.2.5 Other Interactive Demos 7
Chapter 3. Electrical Characteristics 9
3.1 Overview 9
3.2 Starter Kit Configuration 9
3.2.1 Embedded Host, Debug Mode 9
3.2.2 Device, Debug Mode.................... 9
3.2.3 Device, Release Mode.................... 10
3.2.4 Embedded Host, Release Mode (Stand-Alone Mode).................... 11
Chapter 4. Developing an Application 13
4.1 Setting up an Example Application for Debug 13
4.2 Running the Example Application.................... 14
4.3 Debugging the Example Application.................... 15
4.3.1 Editing Application Code 15
4.3.2 Using Breakpoints and Mouseovers.................... 15
4.3.3 Using Watch Windows.................... 16
4.4 Programming the Debugged Applications 16
4.5 Creating Other Applications 17
4.6 Determining Device Support and Reserved Resources.................... 17
4.7 Debug Version Information 17
4.8 Troubleshooting.................... 17
4.8.1 Debug Connection Problems.................... 17
4.8.2 Programming Problems 17
Chapter 5. Hardware Detail 18
5.1 Application Functional Overview 18

5.2 Programmer/Debugger Functional Overview 19
5.3 Board Components 19
5.3.1 Programmer/Debugger Components 20
5.3.2 Application Components 20
Chapter 6. Mechanical Drawing 22
Chapter 7. Appendix 23
Chapter 8. Contact Us 29

NOTES:

Product Version : Ver 1.1

Document Version : Ver 1.0

Chapter 1. Overview

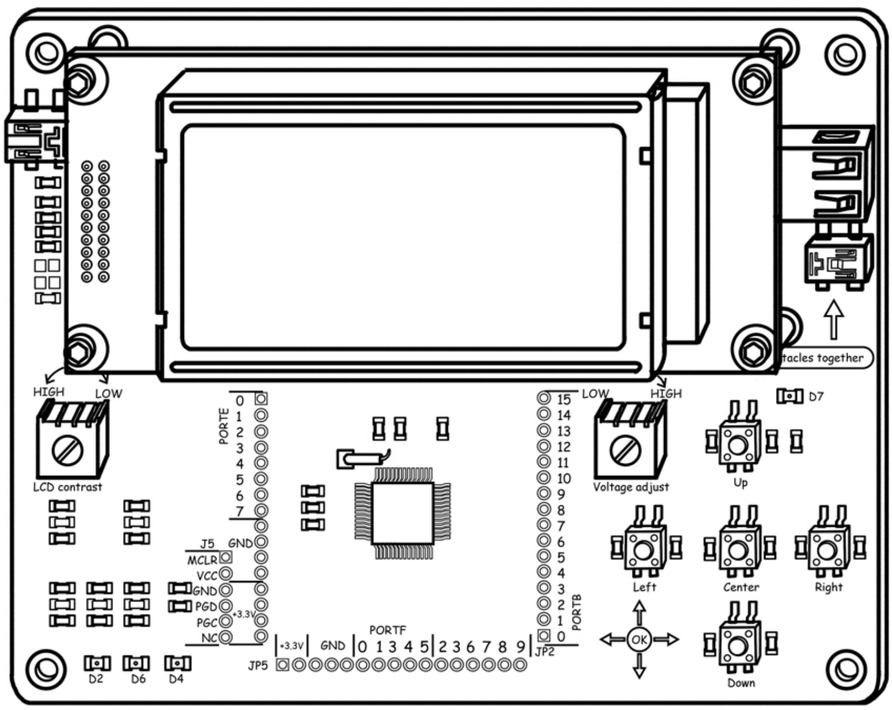
1.1 Overview

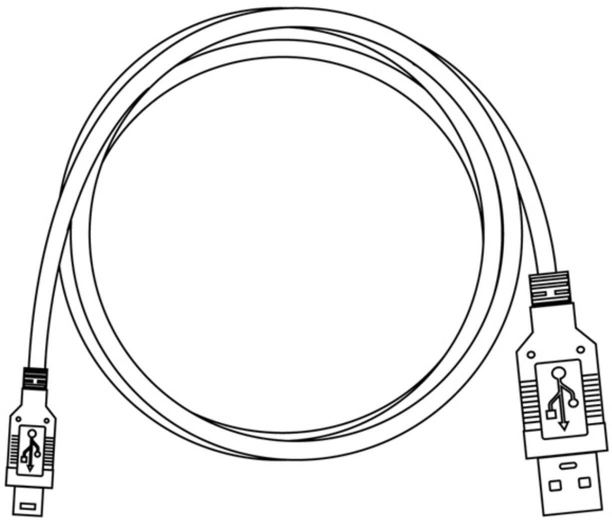
Thanks for purchasing Sure Electronics Technology's PIC24F Starter Kit. This board is intended to introduce and demonstrate the capabilities and features of PIC24F microcontrollers. In addition, the starter kit has on-board in-circuit debug circuitry so that you may develop and debug your own applications.

The PIC24F Starter Kit an all-in-one solution for debugging and programming applications Microchip's MPLAB Integrated Development Environment (IDE). A USB connection to a host computer supplies communications and power to the board; no additional external power supply is needed. For independent host-side USB operation, the starter kit may be disconnected from the PC and powered at test points for independent functionality.

The starter kit includes integrated debug and programmer circuitry that allows applications to be programmed onto the board's PIC24F MCU device and then debugged, all using MPLAB IDE. The need for an additional programmer or hardware interface has been completely eliminated.

FIGURE 1-1 OVERVIEW





1.2 Operational Requirements

To communicate with and program the PIC24F Starter Kit, the following hardware and software requirements must be met:

. PC compatible system

. One available USB port on the PC, or a powered USB hub

. Microsoft® Windows® 2000 SP4, Windows XP SP2, or Windows Vista™ (32-bit)

Note: Only initial testing has been performed on 32-bit Windows Vista for this release. The 64-bit version is not supported at this time.

In addition, the following is needed for some of the application demos:
USB Flash Drive (not included)

1.3 Quick Start

With its pre-installed demo application, the PIC24F Starter Kit is designed to be used straight out of the box. Except for a single connection to a computer, no additional hardware or configuration is necessary.

1.3.1 Installing the Software

Before connecting the starter kit to any computer for the first time, it is very important to install MPLAB IDEB V8.12 (or higher version) and C30 complier. This ensures that the proper USB drivers for communicating with the starter kit programmer/debugger are installed and ready to recognize the board.

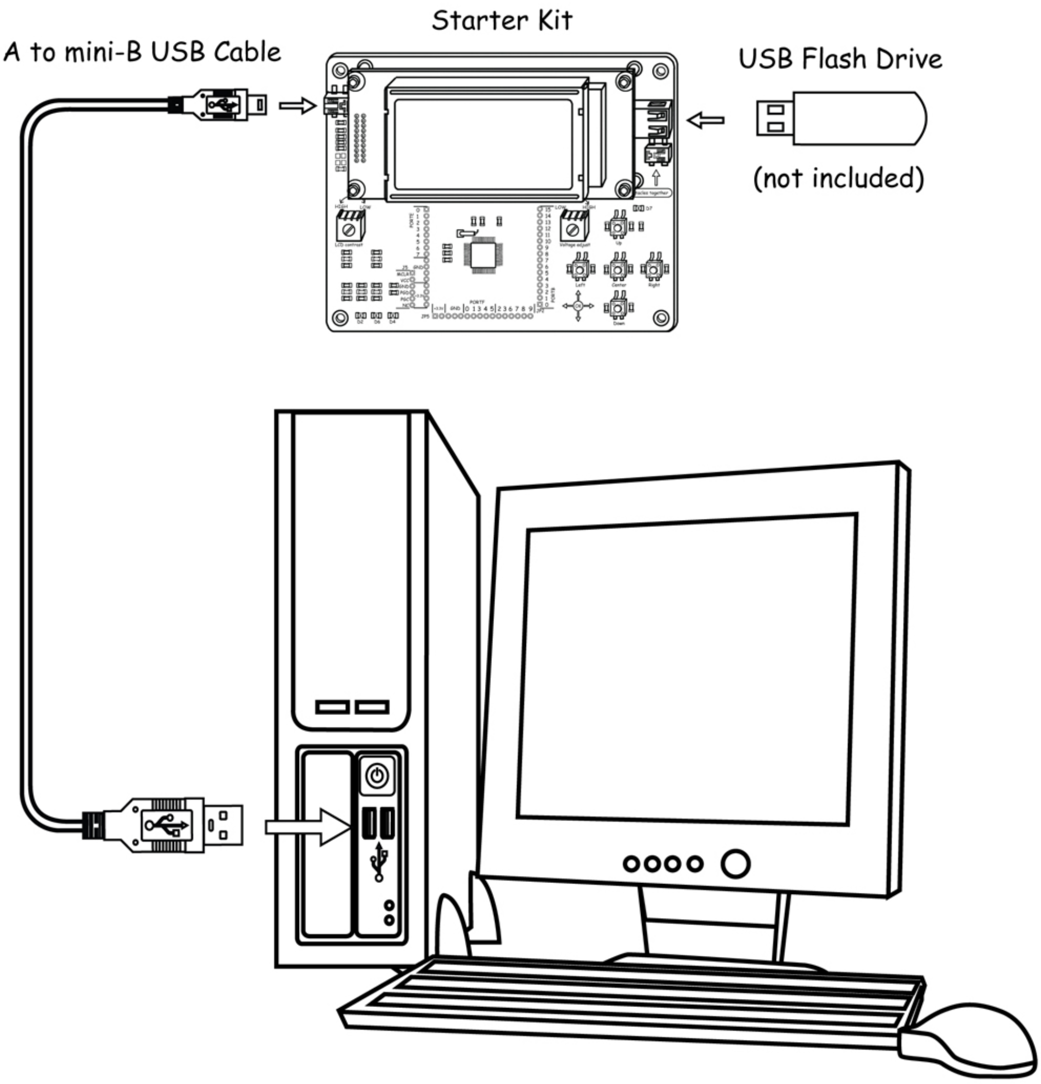
1.3.2 Connecting the Hardware

Once the starter kit software is installed, connect the provided USB cable (A to mini-B) to any available USB port on the PC or powered hub, then to the starter kit at the mini-B receptacle, J3, on the programmer/debugger side of the board (Figure 1-2). The PC USB connection provides communication and power to the board. A USB Flash drive, used for portions of the demo application, may be connected to the starter kit at any time.

Note: Never use two receptacles together.

If the cable is connected correctly, the green Power and Target Power LEDs (D6 and D2) are lit. The LCD displays a reversed raster and start-up screen, while the tri-color LED cycles through a sequence of colors. After this power-on sequence, the LCD will display the “PIC24F Starter Kit” main menu. At the same time, a sequence of pop-up balloons in the system tray (lower right of desktop) should appear, stating that (1) new hardware has been found, (2) drivers are being installed and (3) the new hardware is ready for use. If you do not see these messages and the starter kit does not work, try unplugging and reconnecting the USB. If this does not work, refer to Section 4.8.

FIGURE 1-2: STARTER KIT SETUP



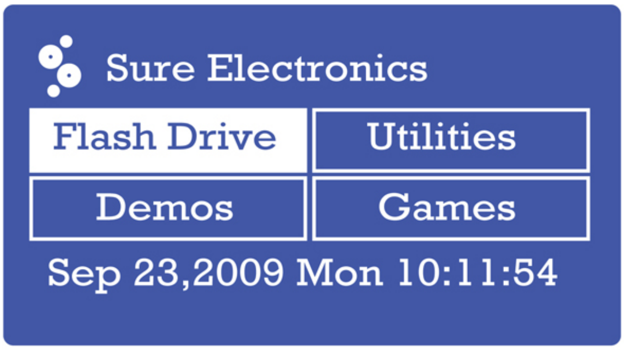
Chapter 2. The Demonstration Application

2.1 Features used throughout the demo application

2.1.1 Interactive Display System

With the exception of input and display hardware, the entire demo application runs on the starter kit's PIC24FJ256GB106 microcontroller, without the need of additional interfaces or external logic for processing support. This microcontroller simultaneously controls and monitors all activities through a processing intensive interactive display system.

FIGURE 2-1 STARTER KIT MENU DISPLAY (MAIN MENU SHOWN)

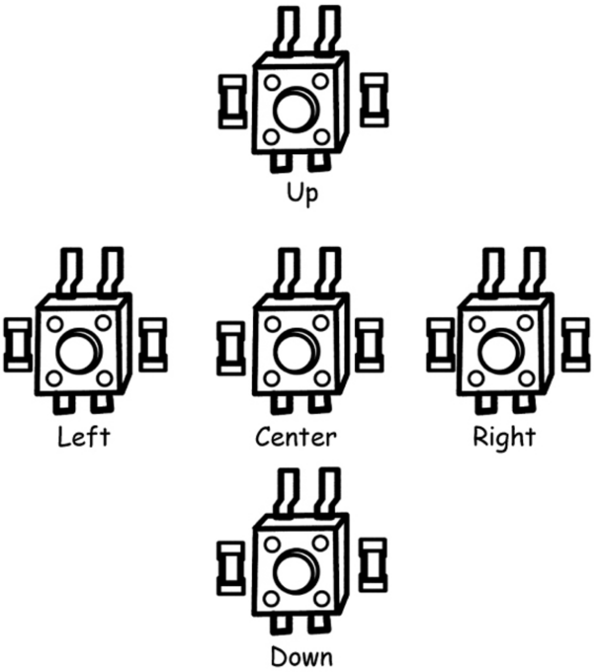


The microcontroller drives the on-board128*64LCD display .Even with a simple visual interface, a graphics display requires a great deal of processing power to maintain and update the display while executing its associated application. At 16 MIPS, the PIC24FJ256GB106 microcontroller has more than enough processing power for this type of application.

2.1.2 Five Push Button Switches

To control most of the demo application's features, the PIC24F microcontroller uses five A/D converter channels to implement five push button switches on the starter kit board. The Push Button Switches (Figure 2-2) are used to select all menu options and input user choices for all options. At each of the menu screens, the up/down arrows (5 and 2) and left/right arrows (1 and 4) serve to highlight a menu option displayed on the LCD, and the center key (3) is used to select the option.

FIGURE 2-2 PUSH BUTTON SWITCHES

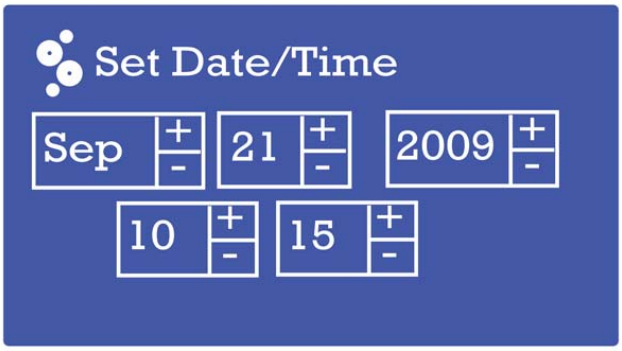


2.1.3 Time and Date (RTCC)

The PIC24F microcontroller has an on-chip Real-Time Clock and Calendar (RTCC) module that the application uses to provide a continuous display of the date and time (in 24-hour format) on the main menu. As the starter kit does not have an on-board battery, the date and time must be set each time power is applied to the board.

To set the RTCC, select "Utilities" from the main menu, then "Date/Time" from the "Utilities" menu. Use the left/right buttons to scroll between fields; the border of the selected field becomes bolded. Use the up/down buttons to set the value. When finished, press the center button to return directly to the main menu.

FIGURE 2-3 SET DATE/TIME DISPLAY



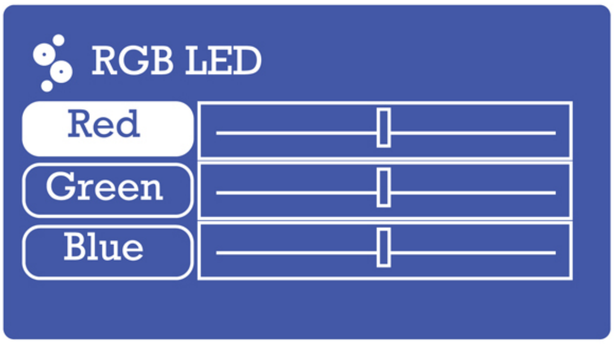
2.2 Specific Demo Highlights

2.2.1 RGB LED Control (Three PWMs and Peripheral Pin Select)

Three of the PIC24F microcontroller's PWM modules control the three-color LED (D7), and can be adjusted to control the LED's brightness and color. To provide more current to the LED channels and increase brightness, each channel is tied to two of the microcontroller's output pins. The Peripheral Pin Select feature of PIC24F devices allows users to map the output of a single module (in this case, a PWM) to more than one pin, avoiding any issues of coordinating separate pins in the process. The use of multiple pins for a single PWM is transparent to the rest of the application.

To access this feature, select "Demos" from the Main display, then "RGB LED" from the "Demonstrations menu". The LED lights up at this point. By default, all three colors are set at their brightness midpoint.

FIGURE 2-4 DEFAULT RGB LED CONTROL DISPLAY



Use the up/down buttons to select a color component, and the left/right buttons to adjust that color's intensity. When finished, press the center button to extinguish the LED and return to the main display.

2.2.2 USB Flash Drive Interface (USB Embedded Host)

This demo shows the ability of the PIC24F microcontroller to function as an embedded host by reading data from a USB Flash drive. To access the demo, select "Flash Drive" from the main menu.

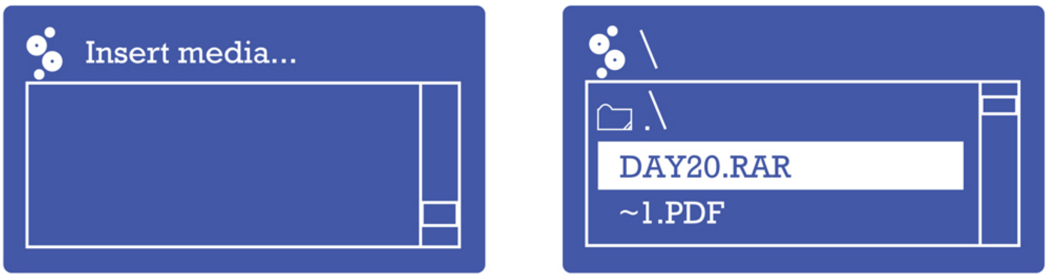
If a USB Flash drive with files is plugged in to the USB A-receptacle (J2), the display shows the drive volume name and top level file structure in a list box (Figure 2-5). Long file and directory names are truncated according to earlier Microsoft 8+3 file system conventions (e.g., Longfilename.doc would be truncated as LONGFI~1.doc). Use the up/down buttons to scroll through the directory.

Subdirectories are indicated by names with a folder symbol next to them, and are opened by pressing the center button. To move up a directory level, select the "--" entry at the top of the menu window and press the center button.

If a Flash drive is not present, the display shows an empty menu window and the message "Insert media". If the Flash drive has no content, the display will show only the drive volume name.

To exit the Flash Drive demo and return to the main menu, press the left button.

FIGURE 2-5 TYPICAL DISPLAYS FOR THE FLASH DRIVE INTERFACE



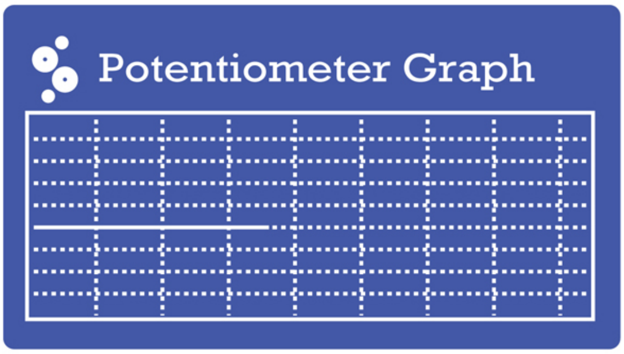
2.2.3 Real-Time Data Graphing (A/D and Display Multitasking)

As an additional demonstration of the application's ability to interact with users in real time, the Data Graphing demo shows the capture and conversion of analog data to graphic information as it happens. In this demo, the input from the potentiometer (R47) is converted by the microcontroller's A/D converter to digital information, and plotted as a time vs. amplitude graph on the LCD display.

To access this demo, select "Demos" from the main menu, then "Graph" from the "Demonstrations" menu. A horizontally scrolling graph appears with a solid line indicating the current position of the potentiometer. Turning the potentiometer through its entire range changes the y-coordinate of the line in real time. At its default setting, the application updates the value from the potentiometer every 100 ms; each time axis division represents approximately 2 seconds. To increase or decrease the graph's update rate, press the right or left buttons respectively.

To exit the demo and return to the main menu, press the center button.

FIGURE 2-6 DATA GRAPHING DISPLAYE



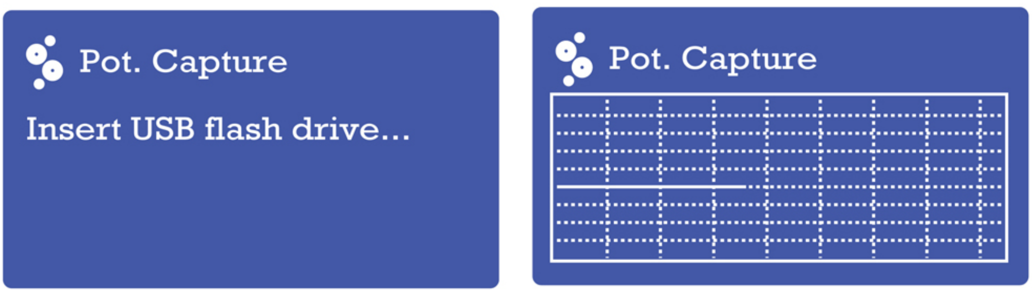
2.2.4 Real-Time Data Capture (Multitasking with USB Embedded Host)

An extension of the Data Graphing demo shows many of the capabilities of the PIC24F microcontroller, all running at the same time: interactive display, multitasking and USB embedded host functionality. This feature allows users not only to graph data and view the results in real time, but also accurately record the results to an external device.

To access this demo, first insert a USB Flash drive into the USB A-receptacle (J2). Select "Demos" from the main menu, then "Capture" from the "Demonstrations" menu. A graph identical in operation to that in the Data Graphing demo appears. (If a Flash drive is not present, a prompt screen will appear instead.) The difference is that the digital value from the potentiometer conversion is written to the Flash drive to a new comma delimited file, CAPTURE.csv. Data is written with a second value representing the time of capture. As the application is configured, the capture interval is 10 ms, and is independent of the graphic display's update rate.

To exit the demo and return to the main menu, press the center button.

FIGURE 2-7 DATA GRAPHING DISPLAYS



2.2.5 Other Interactive Demos

As a final demonstration of the starter kit's processing power and graphics display capabilities, several other interactive demos are provided for the user to investigate. These

are accessed by selecting “Games” from the main menu.
Users may find that the three demos provided (“Shu Box”, “Shapelet” and “Blaster”) similar to some vintage video games. As such, the operations of these demos are regarded to be self-explanatory. Users who require more information will find additional instructions in the comments accompanying the application source code.

Chapter 3. Electrical Characteristics

This chapter discusses how to configure the hardware of the MPLAB Starter Kit for PIC24F for various USB prototypes.

3.1 Overview

In its default configuration, the application side of the starter kit functions as a USB embedded host. Even though the board is drawing power from its connection to the host PC as a bus-powered device, the demo application functions independently of the host PC in communicating with USB peripheral devices.

However, the starter kit can also be used as a platform for developing USB device (peripheral) applications. The mini-B receptacle (J2) on the application side provides the interface for an external USB host.

When setting it up as a prototyping platform, the starter kit board can be configured as either a USB embedded host or a USB device (peripheral). At the same time, the starter kit can be configured to operate with or without the functionality of the programmer/debugger. In terms of the MPLAB IDE environment, we can think of the application as being in Debug or Release mode.

The hardware configurations described here cover the vast majority of cases that the user is likely to see in developing USB applications. Other configurations and USB applications may be possible.

3.2 Starter Kit Configuration

3.2.1 Embedded Host, Debug Mode

This is the default configuration used when the starter kit is first connected to a host PC (see Section 1.3 "Initial Board Setup"). It is also used when USB embedded host applications are still being developed and debugged. In both instances, power to the starter kit is provided through the USB cable.

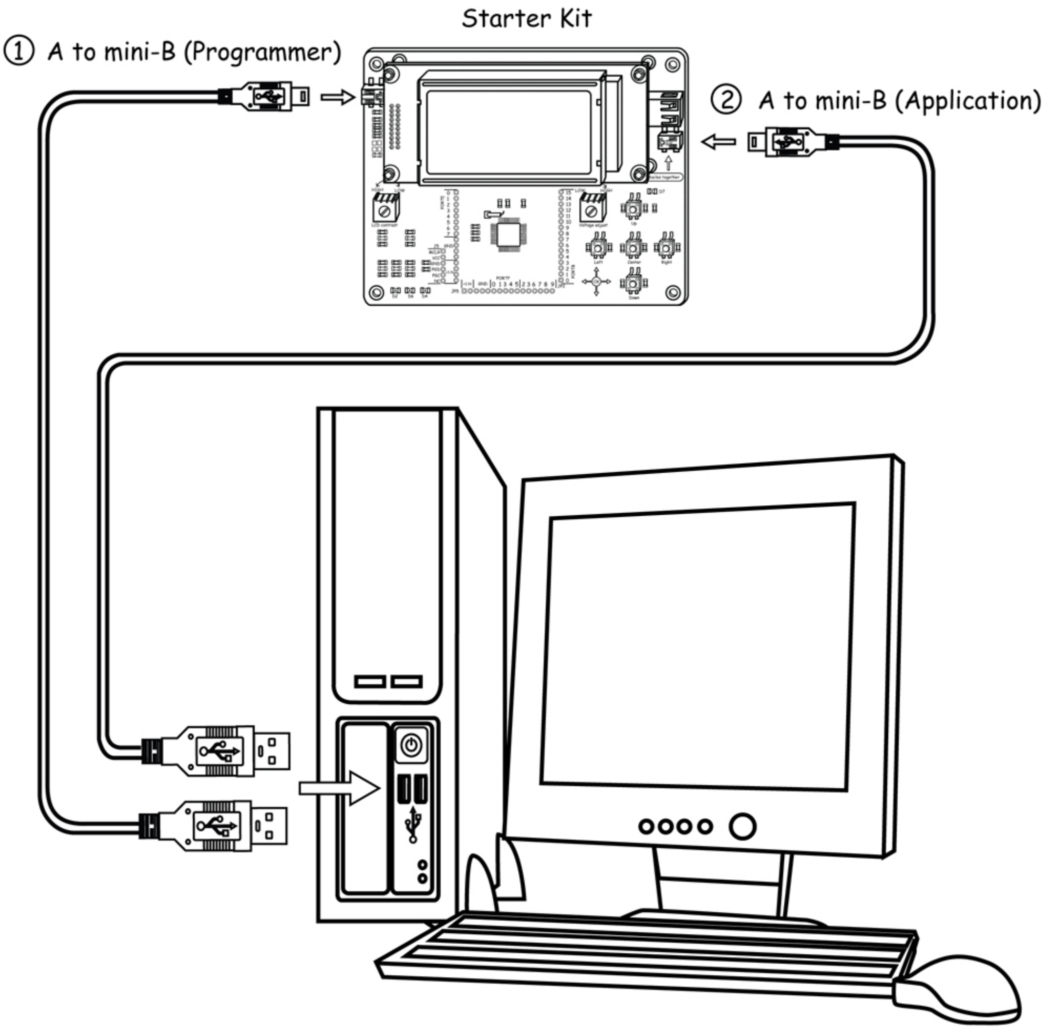
3.2.2 Device, Debug Mode

In this configuration, application code for a USB device is still being developed and debugged. This requires connections to both the programmer/debugger and application sides of the board, using two A to mini-B USB cables (Figure 3-1):

1. Connected from the host PC to the programmer/debugger (J3).
2. Connected from the host PC to the application side (J1).

The programmer side cable provides the interface to MPLAB IDE, while the application side cable provides the interface between the device and host applications. Both cables provide power to the board.

FIGURE 3-1: STARTER KIT SETUP (DEVICE, DEBUG MODE)

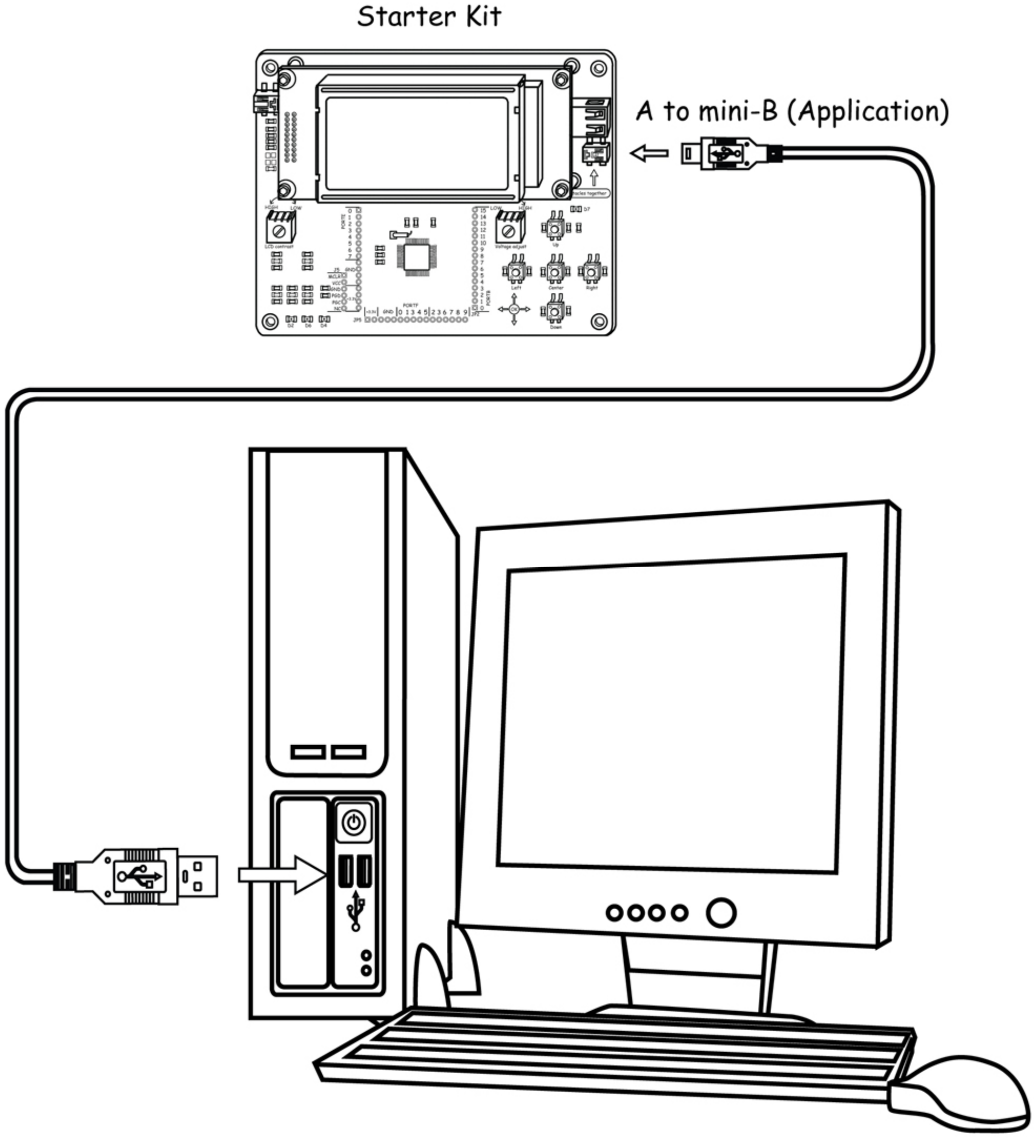


3.2.3 Device, Release Mode

In this configuration, the starter kit has been programmed with a debugged, stand-alone USB device application; programmer/debugger support is not needed.

The only connection required is a USB cable between J1 and the host PC (Figure 3-2). The cable provides both power and data connection to the host side application, making it a bus-powered application.

FIGURE 3-2: STARTER KIT SETUP (DEVICE, RELEASE MODE)



3.2.4 Embedded Host, Release Mode (Stand-Alone Mode)

In this configuration, the starter kit has been programmed with a debugged, stand-alone USB embedded host application; programmer/debugger support is not needed. Because of the absence of a host PC, this configuration can truly be considered a “stand-alone” operation.

Since there are no connections to a host PC, the starter kit board must be physically modified to accept an external power supply. A regulated power supply providing 5 VDC must be available.

Make the following modifications to the starter kit:

1. Install posts at the VBUS and GND test points on the board.
2. Populate the site for R16 (adjacent to the VBUS test point) with a 0 ohm resistor.
3. Supply +5 VDC and ground connections from the external power supply to the VBUS and GND connections on the board.

If the modifications are done correctly, and power is connected correctly, the green Target Power LED (D2) is lit. The starter kit also goes through its power-on sequence before LED1 displays the “PIC24 Starter Kit” main menu.

When configured for external power operation, the programmer/debugger function is essentially disabled. To re-enable these functions, disconnect the external power from the board and remove the resistor/jumper from R16 before connecting the USB cable to J3.

> **Note:** If you externally power VBUS via the test point, please make certain that you first disconnect any USB cables connected to a computer or hub. Otherwise, damage to the PC's USB port or the external hub may result.

Chapter 4. Developing an Application

The MPLAB Starter Kit for PIC24F may be used with MPLAB® IDE, which allows the starter kit to be used as an in-circuit debugger as well as a programmer for the featured device.

In-circuit debugging allows you to run, examine and modify your program for the device embedded in the starter kit hardware. This greatly assists you in debugging your firmware and hardware together.

Special starter kit software interacts with the MPLAB IDE application to run, stop and single-step through programs. Breakpoints can be set and the processor can be reset. Once the processor is stopped, the register's contents can be examined and modified.

4.1 Setting up an Example Application for Debug

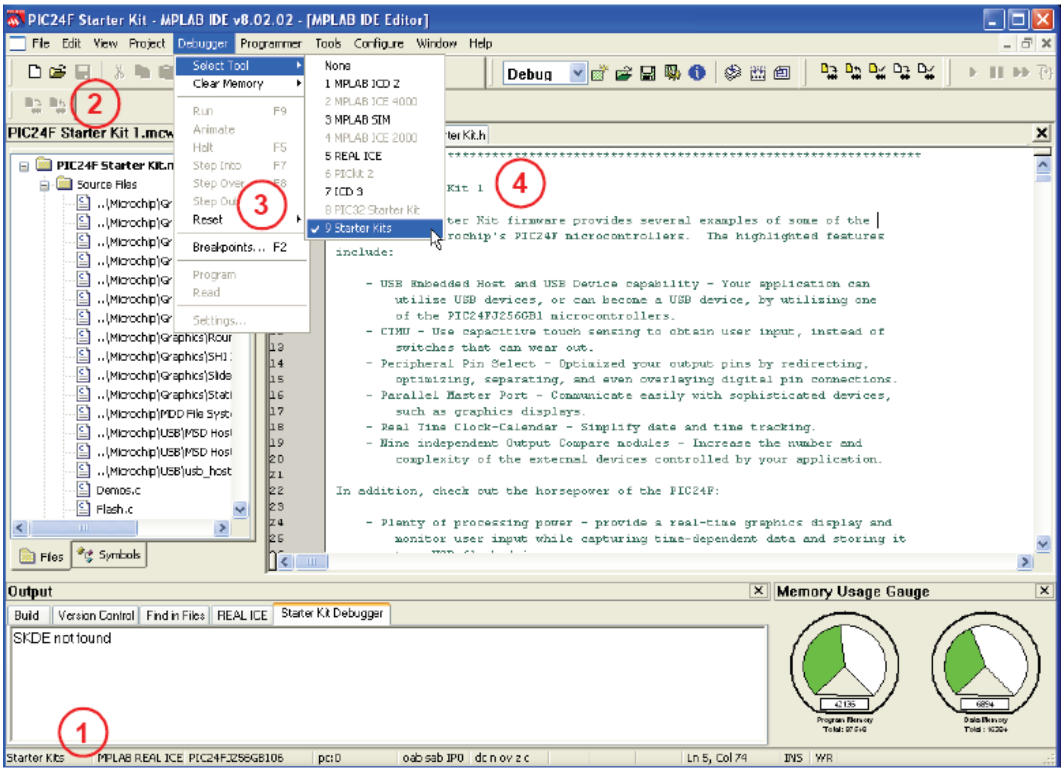
The MPLAB IDE software (DB-DP500-Installer) that is installed on your PC and an example application (STARTER_KITS_PIC24) that you may use to examine debug features of the starter kit can be downloaded from http://www.sure-electronics.net/download/index.php?name=DB-DP500&type=0. To prepare the application for debug:

1. Launch MPLAB IDE. The example application project and related workspace will open.
2. Select *Project>Build All* to build the application code. The build's progress will be visible in the **Build** tab of the Output window.
3. Select *Debugger>Select Tool>Starter Kits*. MPLAB IDE will change to add starter kit debug features (Figure 4-1):

(1) The status bar will show Starter Kits as the debug tool
(2) A Starter Kit debug toolbar will be added
(3) The Debugger menu will change to add Starter Kit debug functions
(4) The Output window will display communication status between MPLAB IDE and the starter kit on the **Starter Kit Debugger** tab.

Also, several device resources are used for debug. For details, see **Section 4.6 "Determining Device Support and Reserved Resources".**

FIGURE 4-1: STARTER KIT AS DEBUG TOOL



4. Select *Debugger>Program* to program the application code into the PIC24FJ256GB106 device on the starter kit. The debug programming progress will be visible in the **Starter Kit** tab of the Output window.

Note:

1. Debug executive code is automatically programmed in the upper program memory of the starter kit device when the starter kit is selected as a debugger.
2. Debug code must be programmed into the target device to use the in-circuit debugging capabilities of the starter kit.

4.2 Running the Example Application

The starter kit executes in either real time (Run) or steps (Step Into, Step Over, and Animate.) Real-time execution occurs when you select **Run** in MPLAB IDE. Once the device code is halted, either by **Halt** or a breakpoint, you can step.

The following toolbar buttons can be used for quick access to commonly used debug operations:

TABLE 4-1 TOOLBAR BUTTONS

Debugger Menu	Toolbar Buttons
Run	
Halt	
Animate	
Step Into	
Step Over	
Reset	

To see how these options function, do the following:

1. Select *Debugger>Reset>Processor Reset* or click the **Reset** button to reset the

program.

2. Select *Debugger>Run* or click the **Run** button. Observe how the application operates.

3. Select *Debugger>Halt* or click the **Halt** button to stop the program execution. A green solid arrow will mark the line of code in the File window where the program halted.

4. Select *Debugger>Step Into* or click the **Step Into** button to step the program execution once. The green solid arrow will move down one line of code in the File window. Click the button several times to step through some code.

5. Select *Debugger>Reset>Processor Reset*, click the **Reset** button to reset the program again. The arrow will disappear, meaning the device is reset.

4.3 Debugging the Example Application

For the example code given, everything works fine. However, when you are developing code, it will likely not work the first time and need to be debugged. MPLAB IDE provides an editor and several debug features, such as breakpoints and Watch windows, to aid in application code debugging.

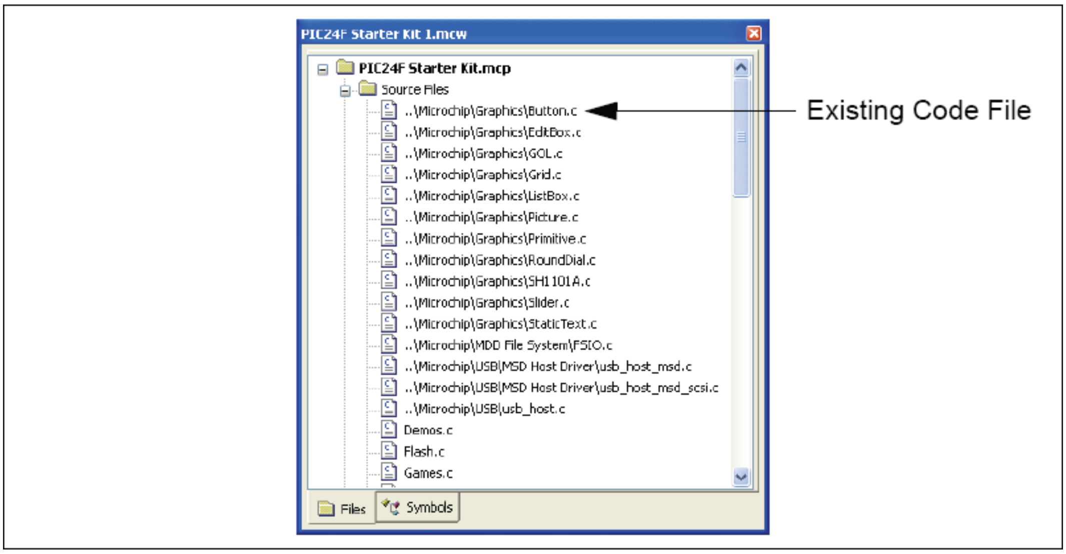
4.3.1 Editing Application Code

To view application code so it may be edited, do one of the following:

. Select *Edit>New* to create new code or *Edit>Open* to search for and open an existing code file.

. Double-click on a file in the Project window to open an existing code file. See an example Project window in Figure 4-2.

FIGURE 4-2: EXAMPLE PROJECT



4.3.2 Using Breakpoints and Mouseovers

To set a breakpoint in code, use one of the following methods:

. **Double-Click in Gutter**: Double-click in the window gutter next to the line of code where you want the breakpoint. Double-click again to remove the breakpoint.

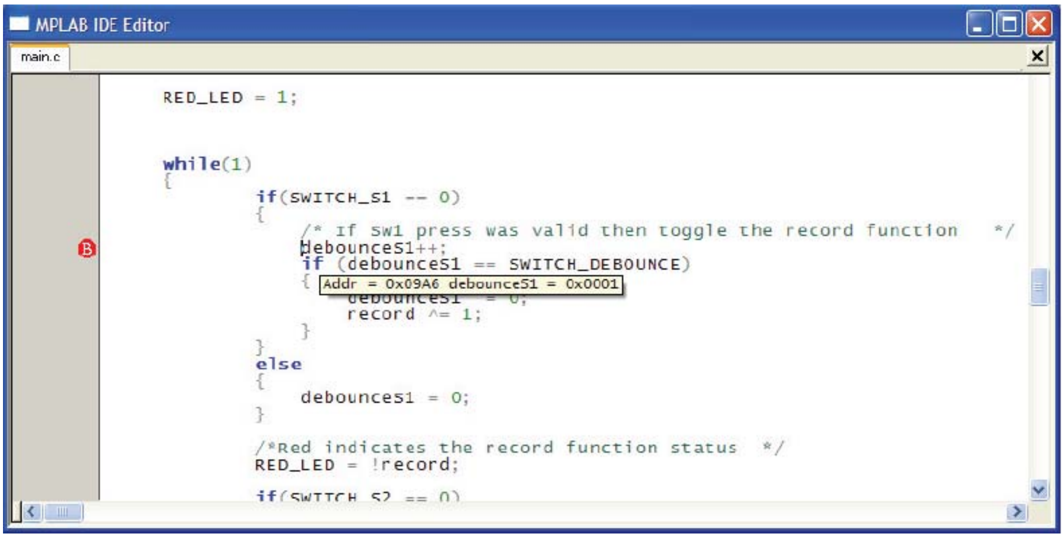
. **Pop-up Menu**: Place the cursor over the line of code where you want the breakpoint. Then, right click to pop up a menu and select “Set Breakpoint”. Once a breakpoint is set, “Set Breakpoint” will become “Remove Breakpoint” and “Disable Breakpoint”. Other options on the pop-up menu under Breakpoints are for deleting, enabling or disabling all breakpoints.

. **Breakpoint Dialog**: Open the Breakpoint dialog (*Debugger>Breakpoints*) to set, delete, enable or disable breakpoints.

A breakpoint set in code appears as a red “stop sign” with a “B”. Once code is halted, hovering over variables pops up the current value of those variables (Figure 4-3.)

Note: This feature must be set up. From the menu bar, select *Edit>Properties*; from the **Properties** dialog, select the **Tooltips** tab, and then click the checkbox for “Enable Variable Mouseover Values”.

FIGURE 4-3: EXAMPLE BREAKPOINT



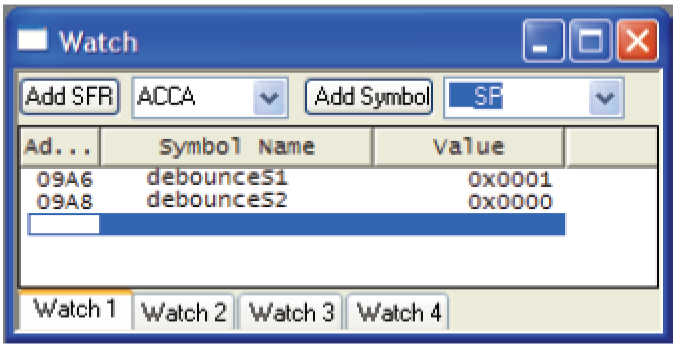
4.3.3 Using Watch Windows

To use a Watch window:

1. The Watch window is made visible on the desktop by selecting *View>Watch*. It contains four selectable watch views (via tabs) in which to view variables (SFRs, symbols and absolute addresses).
2. Select an SFR or symbol from the list and click the related **Add** button to add it to the Watch window, or click in the “Address” column and enter an absolute address.

A Watch window populated with the SFRs and Symbols will look like Figure 4-4.

FIGURE 4-4: EXAMPLE WATCH



4.4 Programming the Debugged Applications

When the program is successfully debugged and running, the next step is to program the device for stand-alone operation in the finished design. When doing this, the resources reserved for debug are released for use by the application. To program the application, use the following steps:

1. Disable starter kits as a debug tool by selecting *Debugger>Select Tool>None*.
2. Select starter kits as the programmer in the *Programmer>Select Programmer* menu.

3. Select *Programmer>Program*.
Now the starter kit will run independently.

4.5 Creating Other Applications

This starter kit is just one way to use PIC24F microcontrollers in an application. Other tools and resources exist to support these devices.

. PIC24 Demo Boards: Several boards, such as the Explorer 16 Starter Kit, are available for developing applications.

. MPLAB C30 C Compiler, Full Version: More optimization options than the student version for full-scale development.

. Application Notes: Libraries and example applications with source code for using the PIC24F family of microcontrollers.

4.6 Determining Device Support and Reserved Resources

Due to the built-in in-circuit debugging capability of ICD devices and the In-Circuit Serial Programming™ (ICSP™) function offered by the debugger, the starter kit uses some on-chip resources when debugging. It also uses program memory and file register locations in the target device during debugging. These locations are not available for use by user code. In the MPLAB IDE, registers marked with an “R” in register displays represent reserved registers.

4.7 Debug Version Information

Selecting either *Debugger>Settings* or *Programmer Settings* from the menu bar opens the Starter Kit Settings dialog. Currently, there is only one (**Info**) tab on this dialog, displaying the following information:

. Firmware Version: The version of firmware on the starter kit board.

. Debug Exec Version: The version of the debug executive that is loaded into the PIC24F device program memory to enable debug operation.

4.8 Troubleshooting

4.8.1 Debug Connection Problems

While using the starter kit as a debugger, you may get the error “Unable to Enter Debug Mode” when programming the device. This can result from communication being lost between the starter kit and MPLAB IDE. To resolve this:

1. Unplug the USB cable from the starter kit.
2. Plug the USB cable back into the starter kit.

MPLAB IDE should automatically reconnect to the starter kit. If this does not work, do the following:

1. Check the USB connection between the PC and starter kit at both ends.
2. If using a USB hub, make sure it is powered.
3. Make sure the USB port is not in use by another device.

4.8.2 Programming Problems

If during the course of developing your own application you can no longer program the device on the starter kit, you may have set device Configuration bits to code-protect or some other state that prevents programming. To view the settings of the Configuration bits, select *Configure>Configuration Bits*

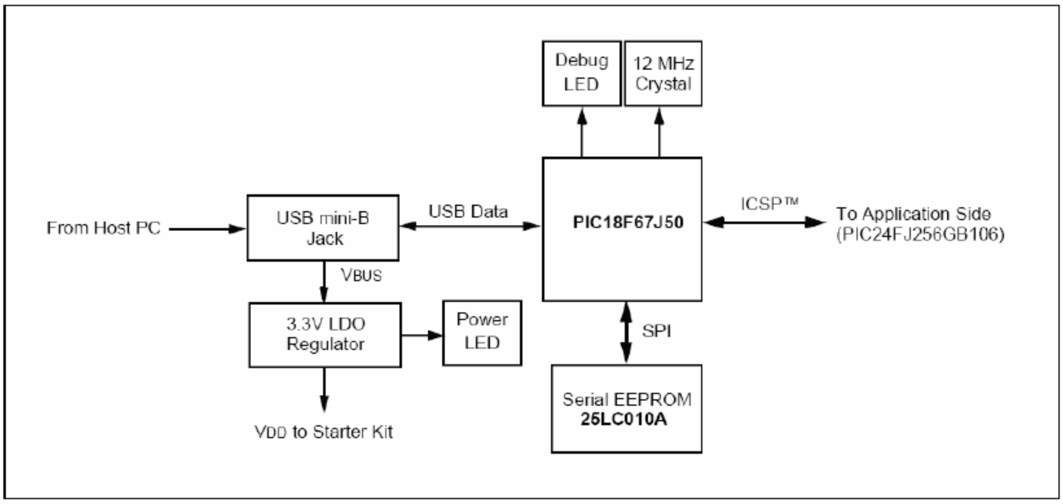
Chapter 5. Hardware Detail

This chapter provides a functional overview of the MPLAB Starter Kit for PIC24F, and identifies the major hardware components.

5.1 Application Functional Overview

Figure 5-1 illustrates the main functions of the starter kit:

FIGURE 5-1: APPLICATION SIDE BLOCK DIAGRAM



The application side of the starter kit is centered on the PIC24F256GB106 microcontroller, which requires very little additional hardware to perform its tasks. All application code is stored in the device's Flash program memory. In addition to the application core, the preloaded demo uses substantial parts of the Microchip USB Stack Library, the Microchip Memory Disk Drive File System and the Microchip Graphics Library to function.

The application accepts user inputs from five push button switches and the potentiometer (R47). The microcontroller uses one of its A/D converter channels to sample and convert the potentiometer's value to a digital value for the Data Graphing and Data Capture demos. Five additional A/D channels are used to monitor the individual switch. The values from these channels are analyzed with the A/D value to determine when a push-and-release event occurs on any of the pads. The application firmware determines which action to take based on the application's current context.

The microcontroller drives the 128*64LCD display and the tri-color LED (D7). The microcontroller uses the PORTE I/O to drive the LCD through the RE7:RE0 data lines, while RB15, RD11, RD2, RD4 and RD5 serve as control signals.

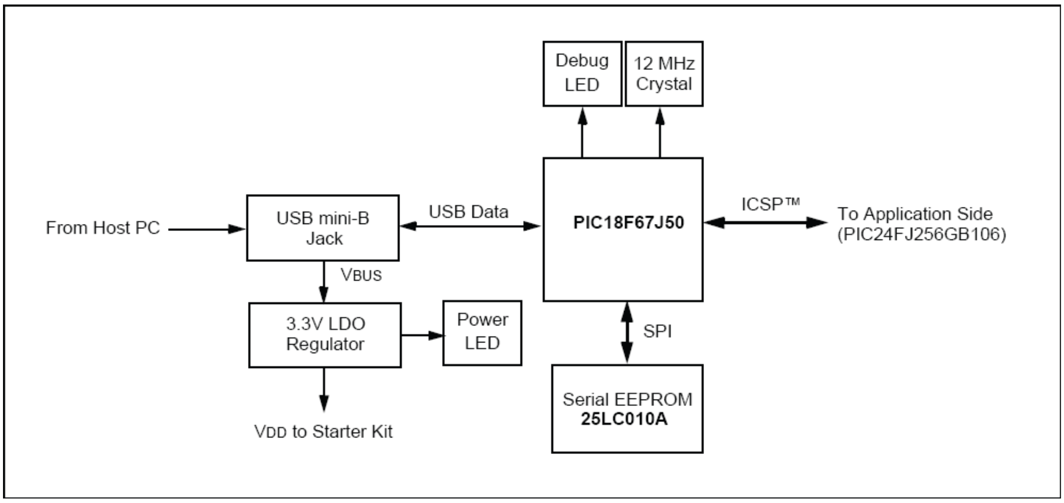
The individual channels of the tri-color LED are driven directly by three of the microcontroller's PWM modules, assigned to outputs on pins RF4, RG6 and RG8.

The microcontroller uses an on-chip USB On-The-Go (OTG) engine and transceiver to communicate with USB A and mini-B receptacles on the application side. While both receptacles are populated, only the A receptacle is used in this version of the demo application. The mini-B receptacle is available for users who may wish to design a peripheral application.

5.2 Programmer/Debugger Functional Overview

Figure 5-2 illustrates the debugging/programming operation of the starter kit.

FIGURE 5-2: STARTER KIT PROGRAMMER/DEBUGGER BLOCK DIAGRAM



In its default configuration, the Starter Kit functions as a USB bus-powered device. Power is provided via the USB cable; the nominal 5 volt unregulated supply is regulated by a CYT6167 3.3 volt low-dropout (LDO) linear regulator. Proper main system power is indicated by the green LED (D6).

The debugging and programming side of the Starter Kit is controlled by a PIC18F67J50 microcontroller running at 48 MHz. The PIC18F67J50's built-in USB engine provides the communications interface between the Starter Kit and the host PC. The microcontroller manages debugging or programming of the target PIC24FJ256GB106 by controlling the target's MCLR, PGC1/EMUC1, and PGD1/EMUD1 signals. Target power is switched on and off via a p-channel MOSFET (Q1) configured as a high-side switch. Target clocking is also provided by the PIC18F67J50.

A Sure Electronics 25LC010A serial EEPROM is used to store the starter kit's serial number and debug control information.

5.3 Board Components

Figure 5-3 identifies the key hardware components for the starter kit.

FIGURE 5-3: PIC24F STARTER KIT COMPONENT LAYOUT

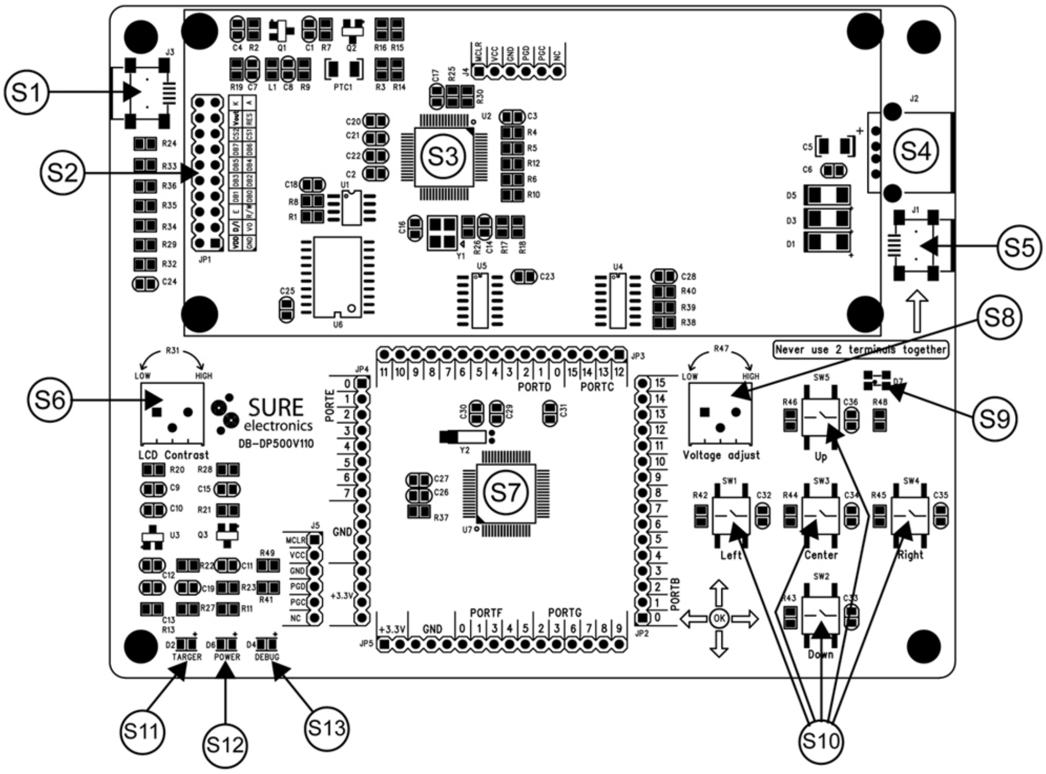


TABLE 5-1: PIC24F STARTER KIT COMPONENT DESCRIPTIONS

Ref	Debug/Programmer Component	Ref	Application Component
S1	Mini-B USB Receptacle (J3)	S8	Potentiometer (R47)
S2	LCD Display	S9	Tri-Color LED Pads (D7)
S3	PIC18F67 J50 Microcontroller (U2)	S10	Five push button switches
S4	USB A Receptacle (J2)	S11	Target Power LED (D2)
S5	Mini-B USB Receptacle (J1)	S12	System Power LED (D6)
S6	LCD Contrast Adjustment (R31)	S13	Debug LED (D4)
S7	PIC24F256GB106 Microcontroller (U6)		

5.3.1 Programmer/Debugger Components

The components listed here (in order of their reference tags in Figure 5-3) are the key components of the programmer/debugger side of the starter kit:

S1. **Mini-B USB Receptacle (J3):** Provides system power and bidirectional communication between the host PC and starter kit.

S3. **PIC18F67J50 Microcontroller (U2):** Controls the programming/debugging operations of the target PIC24FJ256GB106 microcontroller.

S12. **System Power LED (D6):** When lit, indicates that the starter kit is powered via the USB.

S13. **Debug LED (D4):** When lit, indicates that communication between the starter kit and MPLAB IDE has been successfully established.

5.3.2 Application Components

The components listed here (in order of their reference tags in Figure 5-3) are the key components of the application side of the starter kit:

S2. **LCD Display:** A 128 x 64 pixels array provides a wide range of graphics and alphanumeric display options.

S4. **USB A Receptacle (J2):** The application uses this receptacle to connect with USB peripherals while operating as an embedded host.

S5. **Mini-B USB Receptacle (J1):** Physically identical to J3, this provides USB connectivity for the application side when it is functioning as a peripheral device.

S6. **LCD Contrast Adjustment (R31):** Adjust LCD contrast.

S7. **PIC24F256GB106 Microcontroller (U6):** This provides the processing power for the demo applications and application development on the starter kit. The microcontroller features 256 Kbytes of Flash program memory and 16 Kbytes RAM.

The demo application uses an external 12 MHz signal from the programmer side as clock source. Custom applications that do not use the USB module may also use the microcontroller’s on-chip FRC oscillator as a clock source. (USB applications must use the 12 MHz programmer clock source, as the tolerance of the FRC oscillator exceeds USB specifications.)

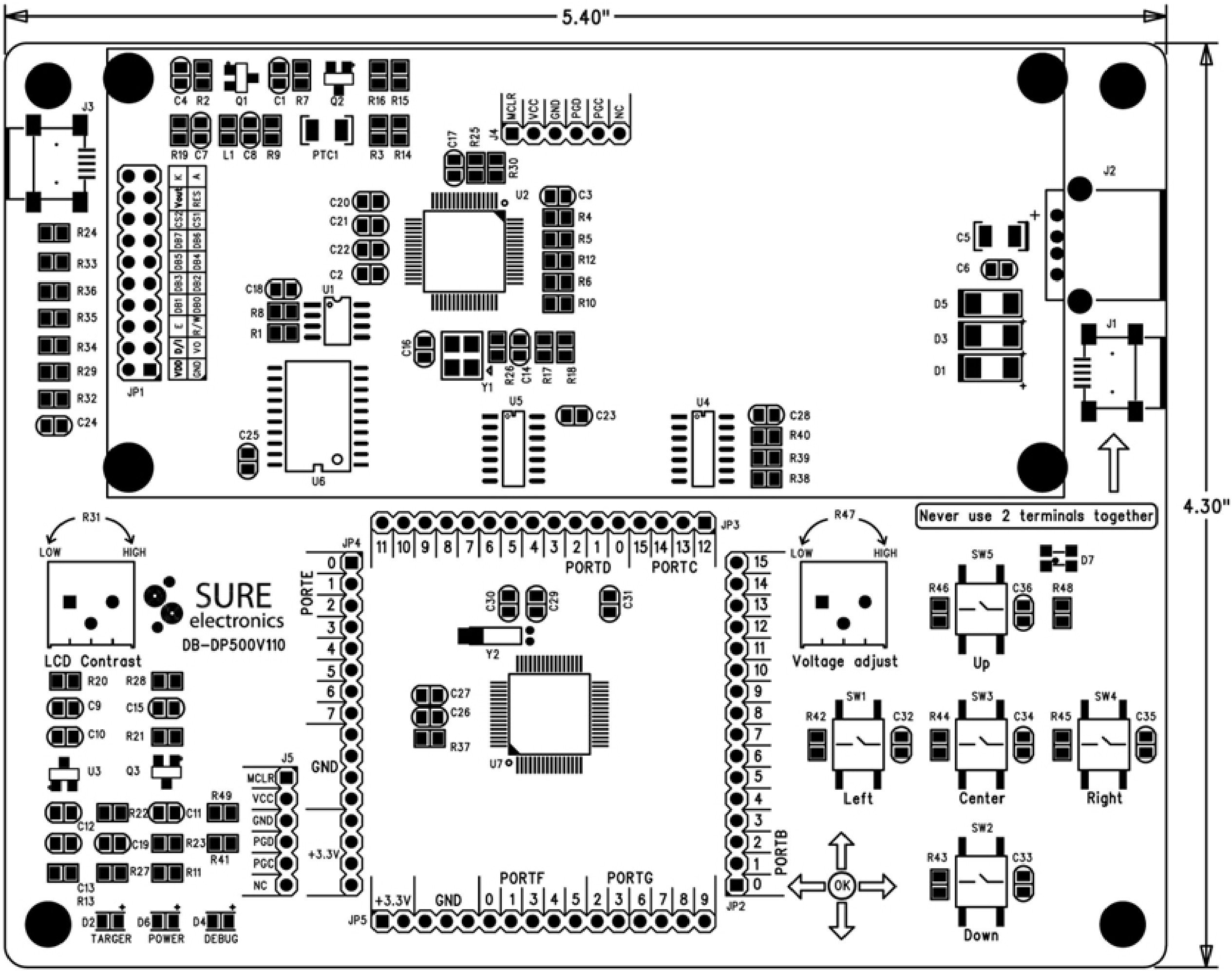
S8. **Potentiometer (R47):** Provides an analog input to the microcontroller for certain demo applications.

S9. **Tri-Color LED Pads (D7):** One of these locations is populated with a three-channel (RGB) LED; the choice of the populated site depends on the availability of components at the time of the board’s assembly. Functionally, the LEDs at any of these sites are interchangeable.

S10. **Five Push Button Switches:** Five A/D converter channels used to respectively implement five push button switches.

Chapter 6. Mechanical Drawing

FIGURE 6-1 MECHANICAL DRAWING



Chapter 7. Appendix

The following schematic diagrams are included in this appendix:

Programmer/Debugger:

. Figure 7-1: Programmer/Debugger Control System and EEPROM

. Figure 7-2: Programmer/Debugger USB Interface, Target Power Switching and Regulation

Application:

. Figure 7-3: Application Microcontroller and Associated Components

. Figure 7-4: LCD Display and LED

. Figure 7-5: Application Side USB Connectors and control buttons.

FIGURE 7-1 PROGRAMMER/DEBUGGER CONTROL SYSTEM AND EEPROM

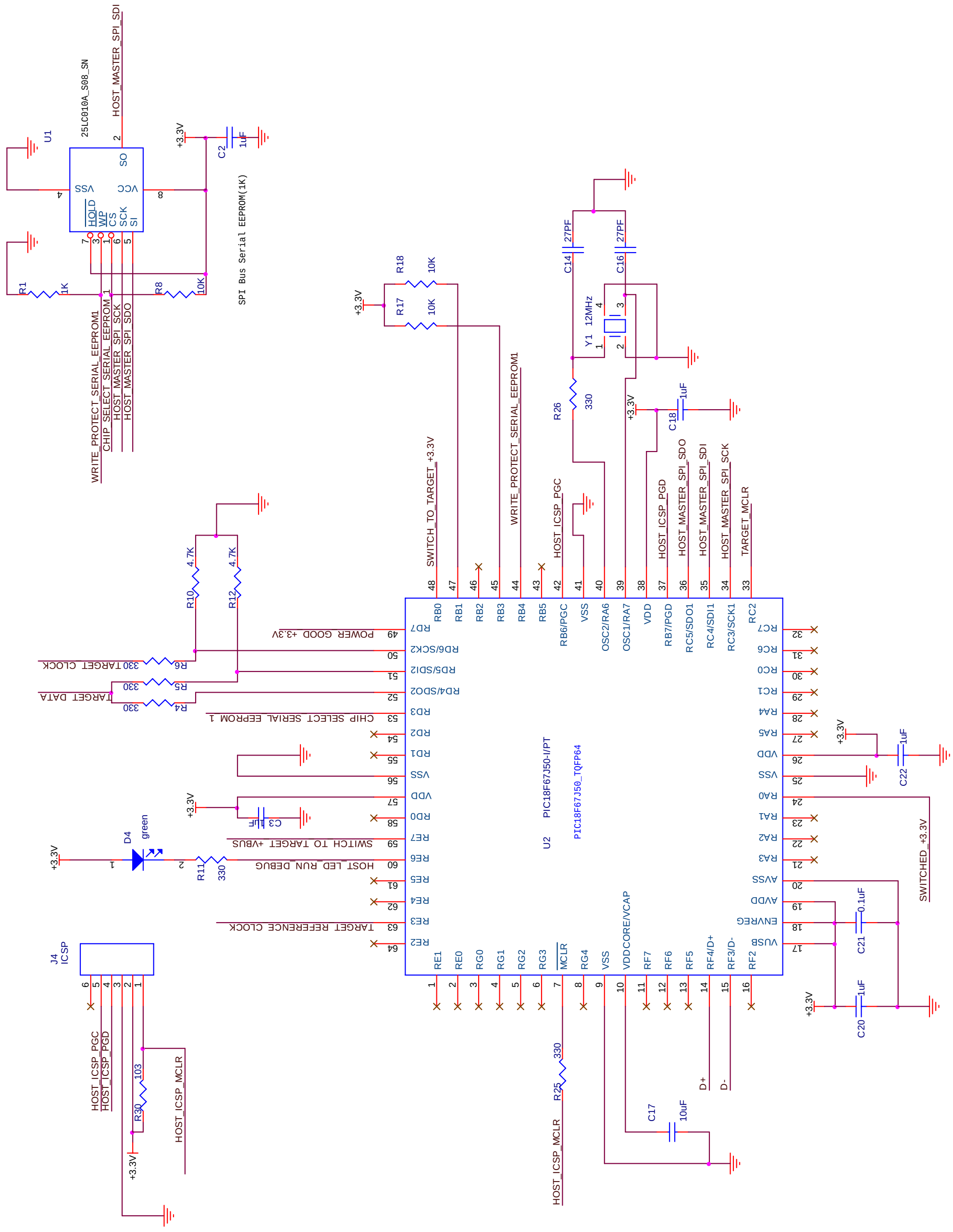


FIGURE 7-2 PROGRAMMER/DEBUGGER USB INTERFACE, TARGET POWER SWITCHING AND REGULATION

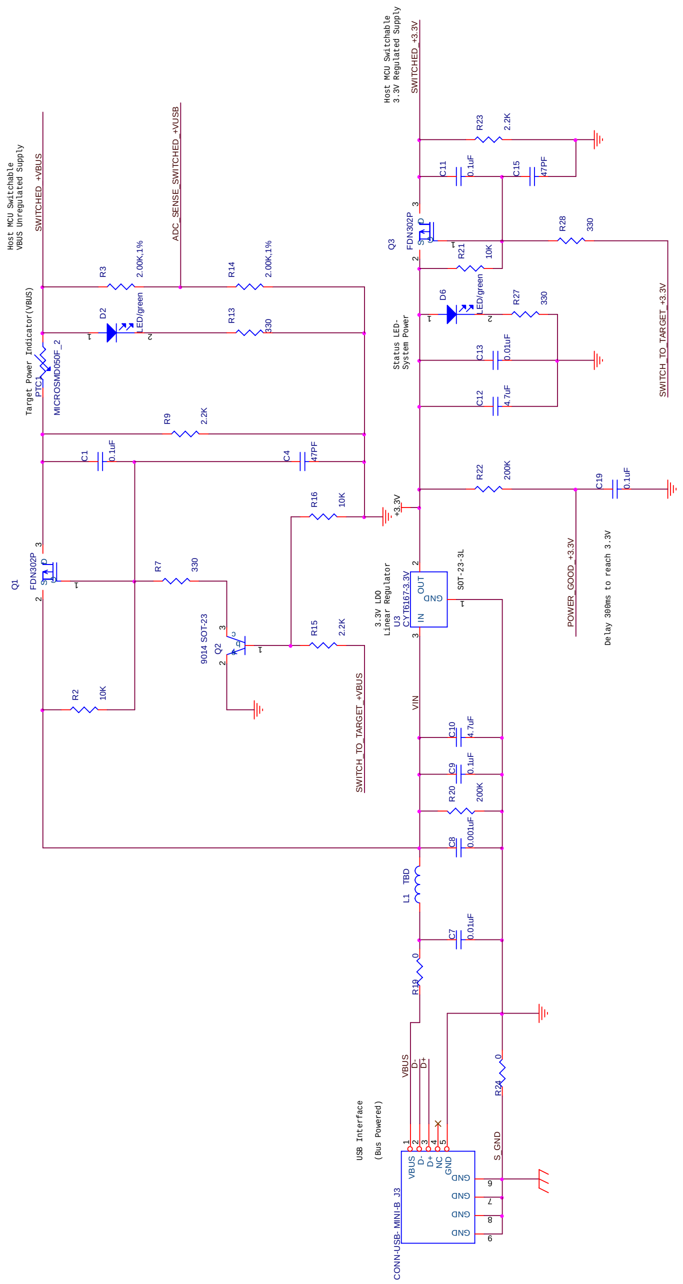


FIGURE 7-3 APPLICATION MICROCONTROLLER, TOUCH SWITCHES AND ASSOCIATED COMPONENTS

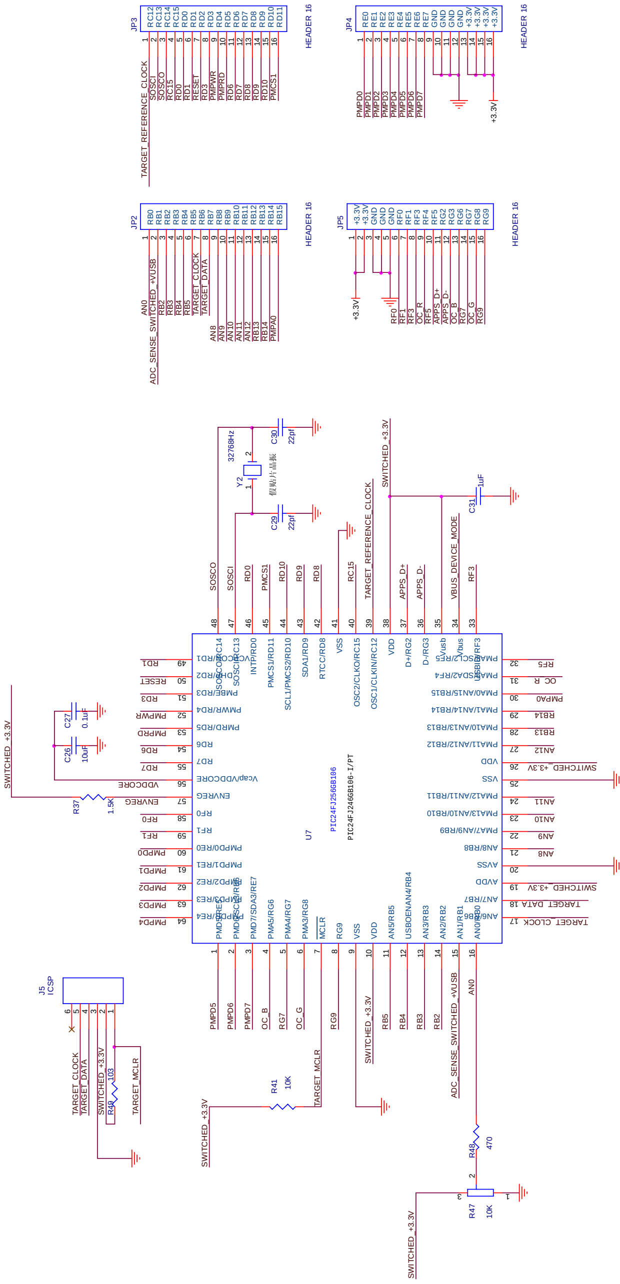


FIGURE 7-4 LCD DISPLAY AND LED

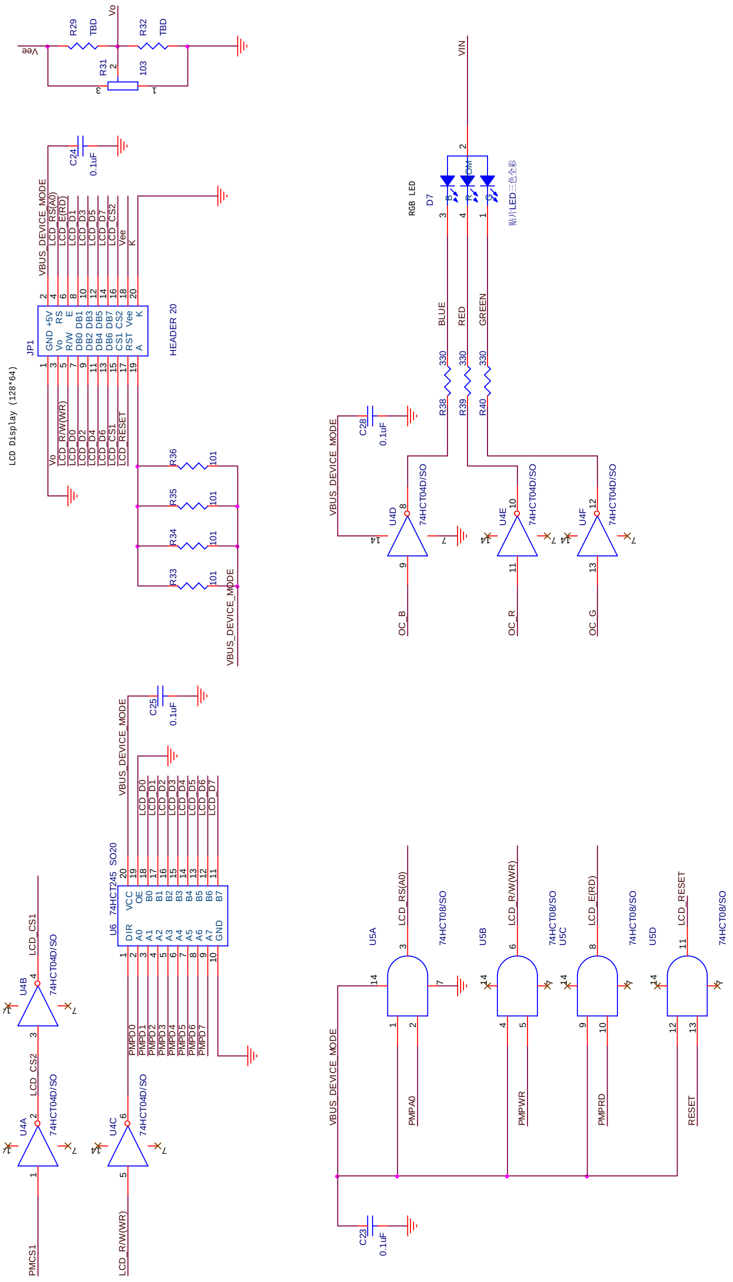
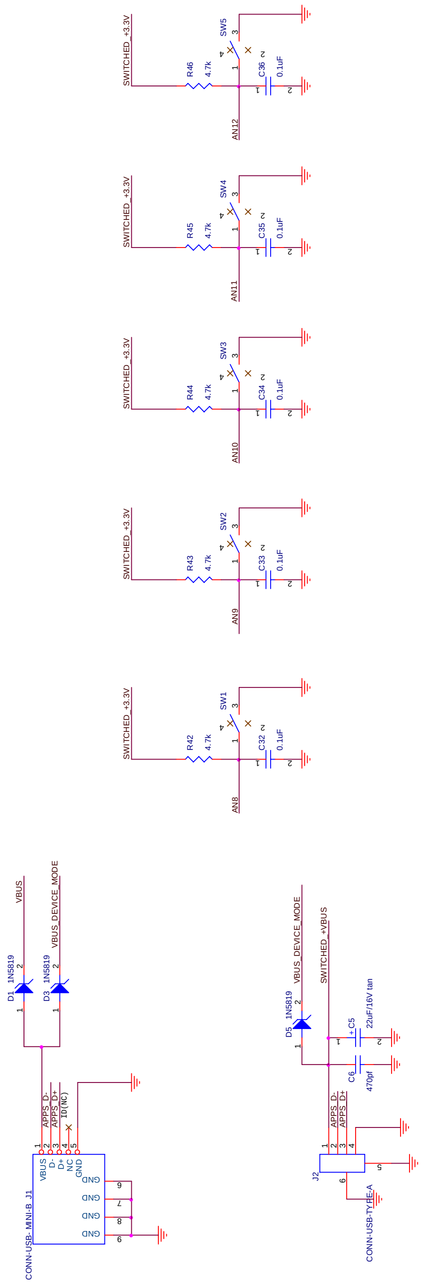


FIGURE 7-5 APPLICATION SIDE USB CONNECTORS AND CONTROL BUTTONS



Chapter 8. Contact Us

Sure Electronics Co., Ltd.
5F, Zone A,
Qinhuai Technology Innovation Center
105-2 DaMing Rd (ZIP:210022)
Nanjing
P.R.China
Tel: +86-13601408832 (For technical questions only)
+86-25-66606340 (English service, from GMT1-10AM)
Fax: +86-25- 66606341-866
Website: www.sure-electronics.com
www.sure-electronics.net