AN1387

Using PIC32 MCUs to Develop Low-Cost Controllerless (LCC) Graphics Solutions

Author: *Adam Folts,*
Microchip Technology Inc.

INTRODUCTION

As the demand for Graphical Embedded Applications becomes more popular, so does the need for developing cost effective solutions. Many companies today offer solutions involving an additional internal or external graphics controller as part of a system, which may result in higher costs and more complicated designs. In most cases, for a simple embedded Graphical User Interface (GUI), these graphics controllers are not necessary, but a suitable solution may not be available. One solution that is becoming more popular is to use a controllerless solution. This solution uses microcontroller (MCU) peripherals to create a "virtual" graphics controller for graphics rendering without taking up large amounts of CPU time (in this solution less than 5%).

The Low-Cost Controllerless Graphics PICtail™ Plus Daughter Board (referred to as the LCC Graphics Board), was designed to showcase a technique offered by Microchip that utilizes this low-cost controllerless method and is designed to work with many existing PIC32 starter kits.

This application note describes the LCC Graphics Board that enables the implementation of a low-cost controllerless graphic method using a PIC32 microcontroller.

Basic Graphics Definitions

Pixel – One dot of color data on a LCD.

Refresh Rate — Defined in Hertz (Hz), the rate at which the LCD panel frame is being redrawn per second.

Resolution – Defined in horizontal by vertical dimensions, the number of pixels a certain display panel contains. For example, a QVGA LCD panel that is 320x240 pixels has a horizontal pixel count of 320 and a vertical pixel count of 240.

Pixel Clock (PCLK) – LCD panels uses this signal to synchronize the sampling of incoming color data. The clock signal needs to be faster for higher resolutions so that all pixels of a frame can be clocked.

Frame Buffer – Volatile memory where pixel color data is stored for the purposes of refreshing an LCD screen of a certain resolution

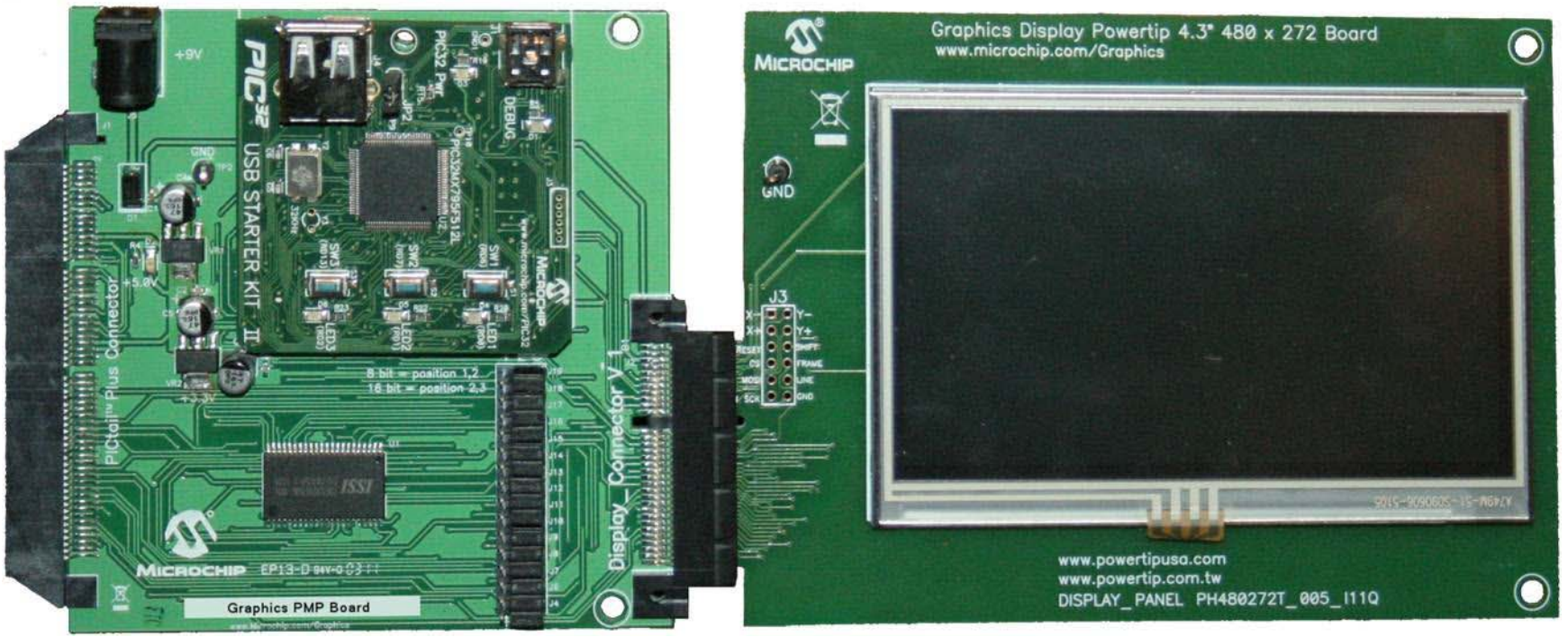
Pixel Throughput – Speed at which a pixel can be redrawn. The time to draw an entire frame would be pixel throughput multiplied by the LCD screen resolution.

Color Depth – Defines how many possible colors a pixel can be drawn. Commonly represented in bits per pixel (BPP), a common color depth is 16 BPP, where the color data is represented in a 565 RGB color format. In this format, 565 represents 5 red, 6 green, and 5 blue color bits.

BOARD SETUP

Figure 1 shows the setup for a PMP-configured LCC Graphics Board with a USB Starter Kit II, connected to a 480 x 272 LCD display.

FIGURE 1: LOW-COST CONTROLLERLESS (LCC) GRAPHICS BOARD SETUP



SETTING UP A CONTROLLERLESS GRAPHICS SYSTEM

In general, a controllerless graphics system needs to send a frame of pixel information to a display glass at a certain rate. This refresh rate is usually around 60 Hz. To do this, the system must constantly send frame data to the LCD panel. At first inspection, it seems like this task would take up most of the CPU time in an MCU. However, this is not the case for PIC32 MCUs that contain a DMA for data transfer. With a DMA transferring the pixel data, less than 5% of CPU time can be used to achieve a "virtual" graphics controller.

PIC32 MCUs have a built-in Direct Memory Access (DMA) peripheral. This peripheral can transfer data from one location to another without CPU intervention. In a controllerless graphics method, the DMA is set up to transfer one line of frame data at a time through the Parallel Master Port (PMP). Each line consists of many pixels. The DMA sends a portion of the frame buffer during one transfer. A PMP or Timer interrupt request is used to trigger the next DMA transfer until a line is transferred. In PIC32 devices with non-persistent interrupts, a timer is used as the DMA trigger source.

During data transfers, the PMP strobes a read or write signal after each pixel transfer. The read/write strobes of the PMP peripheral act as the pixel clock for the display glass. After each line of pixel data is transferred, the CPU is interrupted by the DMA and certain timing signals (e.g., HSYNC, VSYNC, and DEN) necessary for LCD panels are updated. This is repeated continuously until an entire frame has been drawn. The frame is stored in volatile memory so the image can be dynamic. In this setup, SRAM memory is used. This configuration is the foundation for a controllerless graphics system. The system can be set up to use internal SRAM memory or external SRAM memory. A diagram of each system can be seen in Figure 2 and Figure 3.

FIGURE 2: EXTERNAL MEMORY METHOD

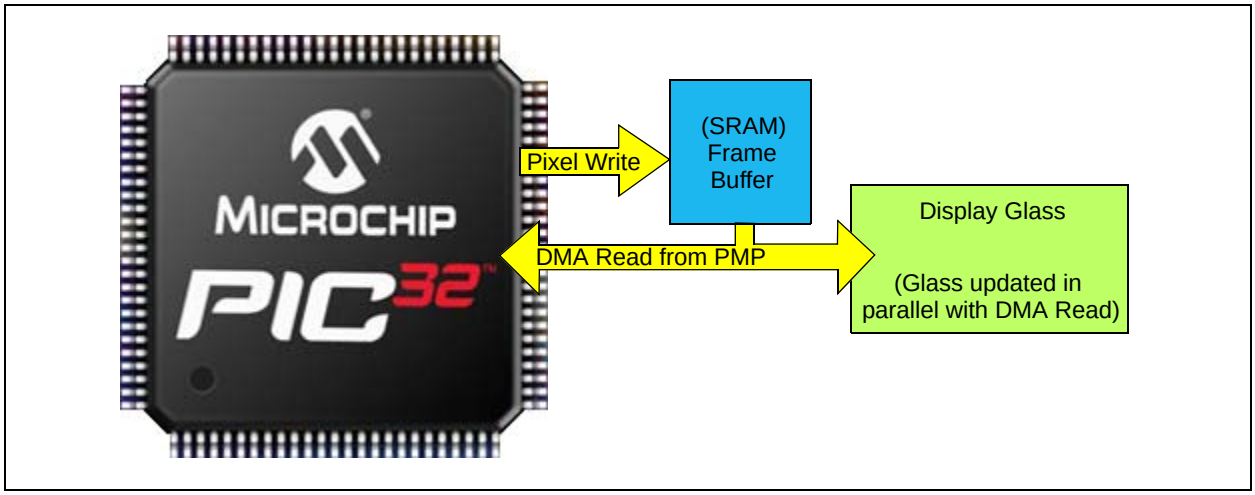
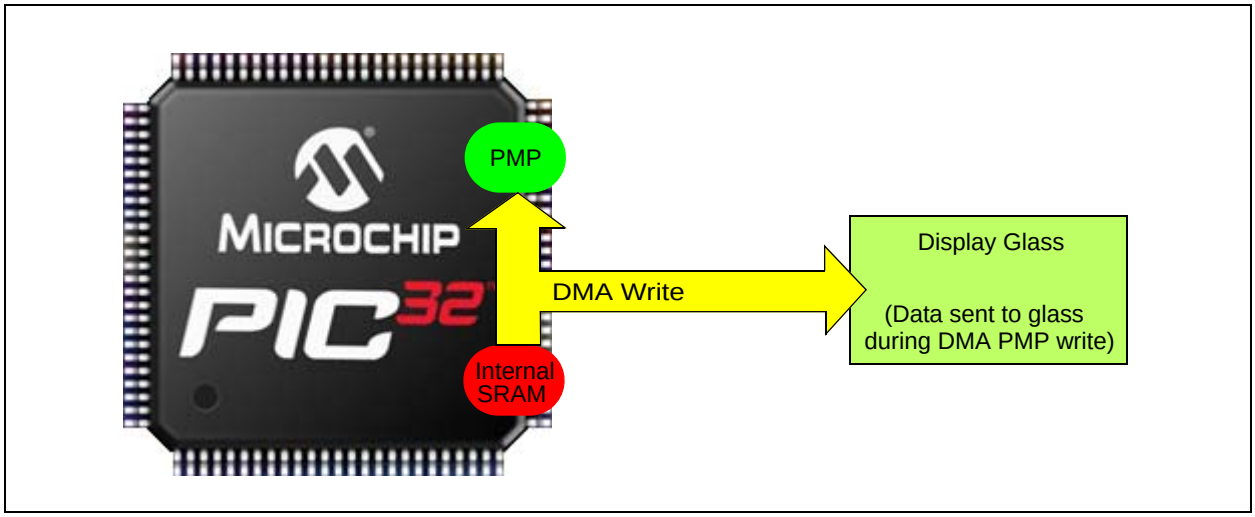


FIGURE 3: INTERNAL MEMORY METHOD



DMA AND PMP INITIALIZATION ROUTINE

Code to set up and initialize the DMA and PMP is provided in Example 1. This code is a snippet from the LCC driver software, and is being provided as a reference to show how easy it is to set up a system for controllerless graphics. It is not intended for copy and reuse purposes.

EXAMPLE 1: DMA AND PMP INITIALIZATION SOURCE CODE

```
//Suspend ALL DMA transfers
DMACONbits.SUSPEND =1;

#define PMP_CONTROL
(PMP_ON|PMP_MUX_OFF|PMP_READ_WRITE_EN|\PMP_CS2_EN|PMP_CS2_POL_LO|PMP_WRITE_POL_LO|PCLK_POLARITY
)
#define PMP_MODE
(PMP_DATA_LENGTH|PMP_MODE_MASTER2|\PMP_WAIT_BEG_1|PMP_WAIT_MID_1|PMP_WAIT_END_1)
// Set up the PMP
// PMP is setup to use data and address lines, in 16 bit PMP mode
// PMP wait states are set to fastest
mPMPOpen(PMP_CONTROL, PMP_MODE, PMP_ADDRESS_LINES, PMP_INT_ON);

//Set PMP address to 0
PMADDR = 0x0000;

// Open the desired DMA channel
DmaChnOpen(1, 0, DMA_OPEN_DEFAULT);

// Set the transfer event control: what event is to start the DMA transfer
DmaChnSetEventControl(1, DMA_EV_START_IRQ(_TIMER_2_IRQ));

// Set the transfer parameters: source & destination address, source & destination size,
// number of bytes per event source is the PMP, destination is a dummy array, source size is
// 2 for 16-bit color, first destination size is the backporch, transfers per event is two.
DmaChnSetTxfer(1, (void*)&PMDIN ,&GraphicsFrame[0] , 2, HBackPorch, 2);

// Set INT controller priority to 7 for highest priority
INTSetVectorPriority(INT_VECTOR_DMA(1), INT_PRIORITY_LEVEL_7);

// Set INT controller sub-priority
INTSetVectorSubPriority(INT_VECTOR_DMA(1), INT_SUB_PRIORITY_LEVEL_3);

// Enable the transfer done interrupt, when all buffer transferred
DmaChnSetEvEnableFlags(1, DMA_EV_BLOCK_DONE);

// Enable the interrupt in the INT controller
INTEnable(INT_SOURCE_DMA(1), INT_ENABLED);

// Once configured, enable the DMA channel
DmaChnEnable(1);

// Turn on Timer2 to act as a "trigger" for the pixel clock (DMA transfer)
OpenTimer2(T2_ON | T2_SOURCE_INT | T2_PS_1_1, 10);

// Start ALL DMA transfers
DMACONbits.SUSPEND = 0;
```

BASICS OF THIN-FILM TRANSISTOR (TFT) LCD PANELS

There are different types of LCD glass to choose from the marketplace. The controllerless graphics method was designed to work with TFT LCD panels, but can work with CSTN or MSTN glass with minor modifications.

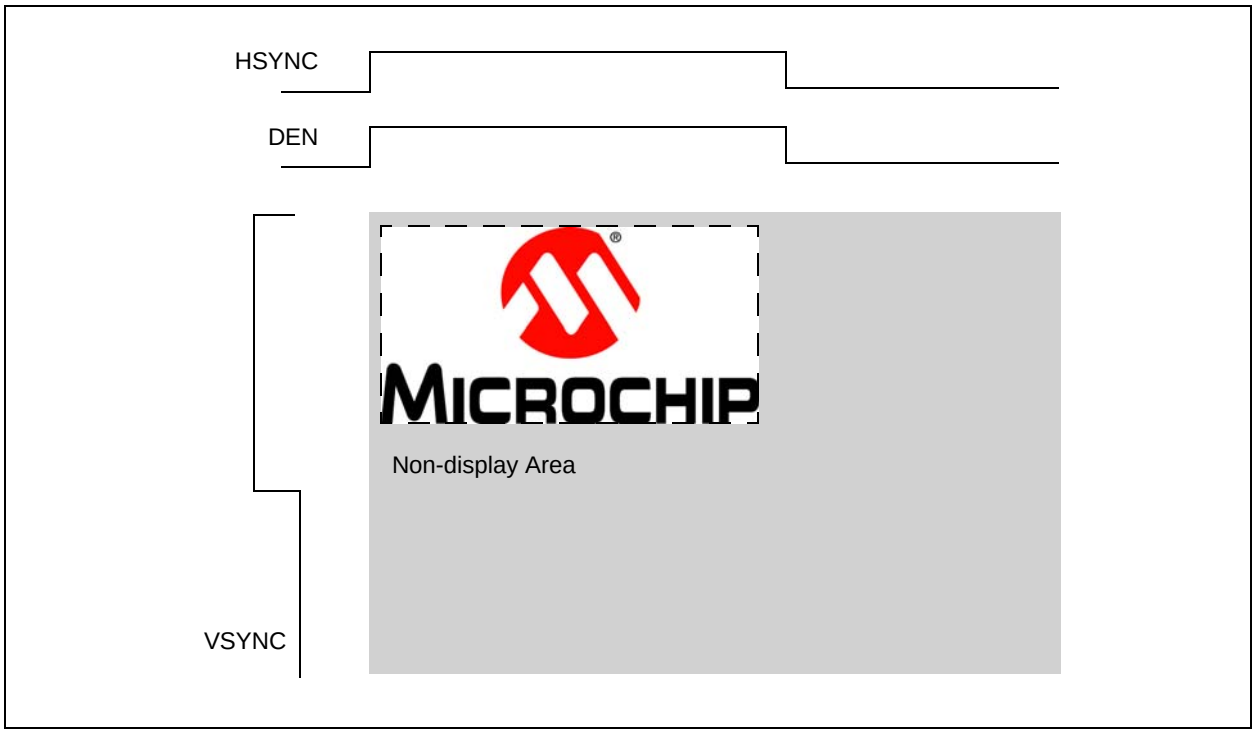
Figure 4 shows a typical timing of a TFT display panel.

The data lines consist of the pixel color information. Most LCD panels can have anywhere from 8 to 24 color data lines depending on the color depth of the LCD panel. These data lines supply the LCD panel with the raw color data of each pixel.

The clock signals HSYNC, VSYNC, DEN, and PCLK are all used to synchronize the pixel data with the graphics frame and the LCD panel. The sync lines tell the LCD panel when the data is at the start or end of a line (HSYNC) or a frame (VSYNC). The DEN or data enable line lets the LCD panel know when valid pixel data is being sent to the LCD panel. DEN is needed for TFT type LCD panels because there is time needed to set up the LCD panel for proper pixel locations. This setup time is seen in Figure 4 as the gray non-display area. The Microchip logo represents the area of the timing where valid pixel data needs to be sent to the screen. This is where the DEN is asserted high. Remember, data is sent one line at a time until the entire frame is drawn. The PCLK signal is not shown, but is the clock source for the whole system. One clock pulse from the PCLK updates the LCD panel. All other clock lines need to be synchronized to the pixel clock to achieve proper image output. Not all display panels have HSYNC, VSYNC, and DEN lines. This application note covers panels that can be used, in order to explain each line and its purpose. LCD panels not containing all of these signals, such as HSYNC and VSYNC, can still be used with the controllerless graphics setup.

The LCC software driver is set up to help with synchronization needing certain timing parameters such as pulse width, front porch, and back porch for both horizontal and vertical pulses. Once these values are compiled into the LCC Graphics driver, the LCD Panel displays the frame. These timing constants can be found in the data sheet of the specific LCD Panel to be used. It is usually a value given in pixel clocks. These constants help set up the proper frequencies needed for accurate image data on the display.

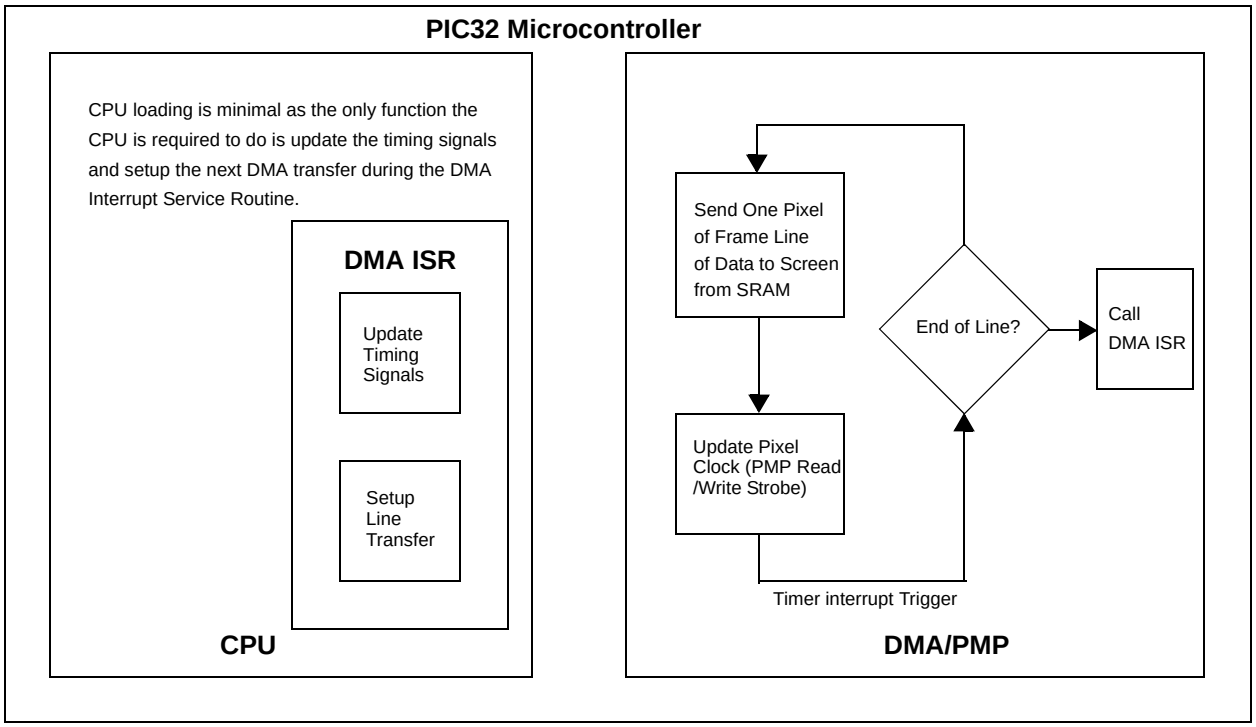
FIGURE 4: CLOCK SIGNAL SETUP TIMES



LCC GRAPHICS FLOWCHART

Figure 5 shows a flow chart of what is happening inside the PIC32 microcontroller when a graphics frame is being sent to a display. The block labeled DMA/PMP shows what the DMA and PMP peripherals are performing, sharing the data bus with the CPU. The block labeled CPU shows the tasks needed for graphics rendering. The DMA ISR (Interrupt Service Routine) is the only code that needs to be written besides setting up the DMA and PMP peripherals to send a graphics frame to a display. This flowchart does not specify where the graphics frame is stored (either internally or externally) nor does it describe updating the frame image

FIGURE 5: FLOWCHART OF PIC32 MICROCONTROLLER



TIMING DIAGRAMS

Many questions come up as to how the whole system works. A good way to address these questions are by the use of what is produced at run-time. This can be seen in some of the timing diagrams that are created in the system. This section shows several different timing diagrams of the system to show how low-cost controllerless graphics works.

The first timing diagram in Figure 6, shows the pixel clock that is generated by the PMP strobe (either read or write). This clock is needed by TFT LCD panels and the PIC32 device can generate signals up to 15 MHz, which is more than enough for WQVGA solutions. All the timing diagrams were taken with a pixel clock of 13 MHz. To change the speed of the pixel clock, the system can slow down the PMP or DMA transfer speed. An SRAM with support for these pixel clocks needs to be considered. This signal can be seen by probing the SHIFT test point seen on most Microchip graphics display boards.

Figure 7 shows a typical frame rate of a graphics system. You will notice the rate is around 60 Hz, which is a typical refresh rate seen in today's market. This signal can be seen by probing the FRAME test point on most Microchip graphics display boards.

FIGURE 6: CLEAN PIXEL CLOCK TIMING DIAGRAM

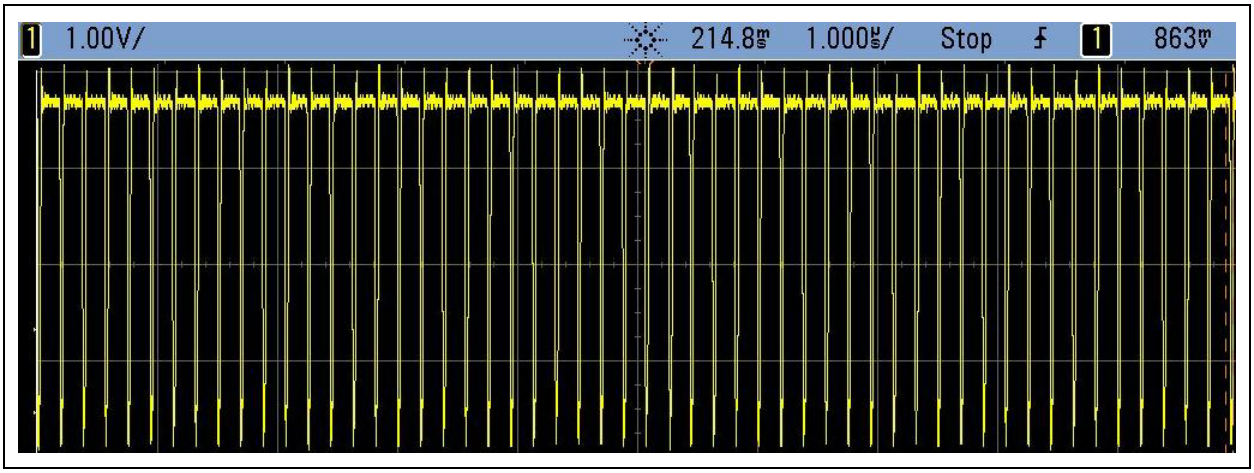


FIGURE 7: FRAME RATE TIMING DIAGRAM

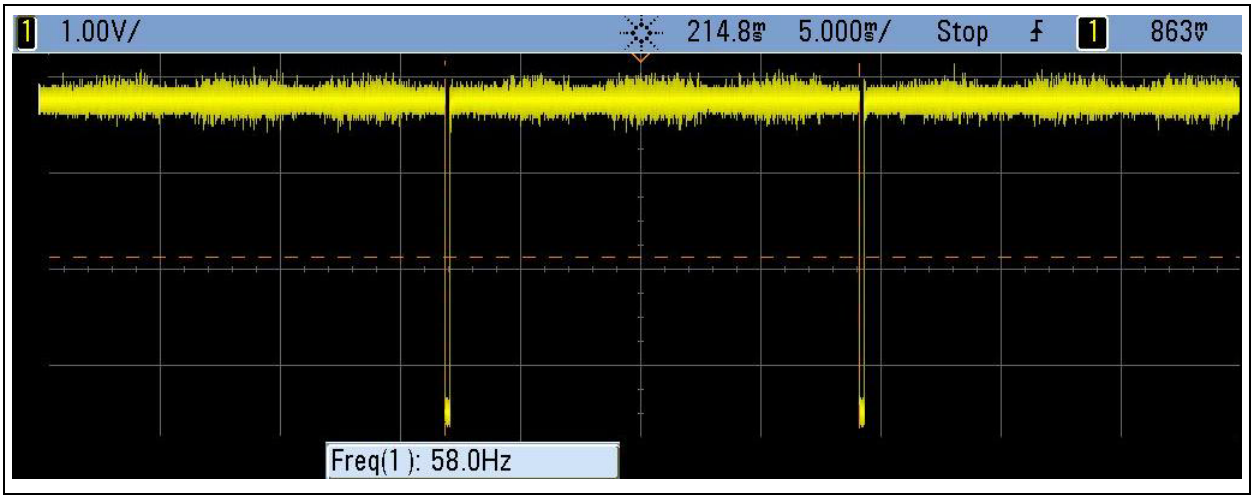


Figure 8 shows a zoomed out pixel clock in external memory code. You will notice two areas in the diagram where the pixel clock is stopped. This is where the ISR is updating LCD timing constraints. This is the only portion of LCC that requires CPU bandwidth. Everything else is handled by the DMA transfer. The delta x shows how much time the ISR routine absorbs, which equates to 80 clock cycles. Therefore, if we consider a WQVGA display refreshing at 60 Hz, with 272 lines and two interrupts per line, the LCC solution only uses roughly (60* 272 * 2 * 80) = 2.6 MIPS.

Figure 9 shows the affect the `putpixel` function has on the pixel clock. `Putpixel` is the function that updates the frame being displayed. The function is designed to update two frames per pixel halt. The total time to update two pixels is 800 ns. Knowing this time the user can change the software to monitor the update rate so that the refresh rate does not fall below a certain refresh (50 Hz).

FIGURE 8: ISR TIMING DIAGRAM

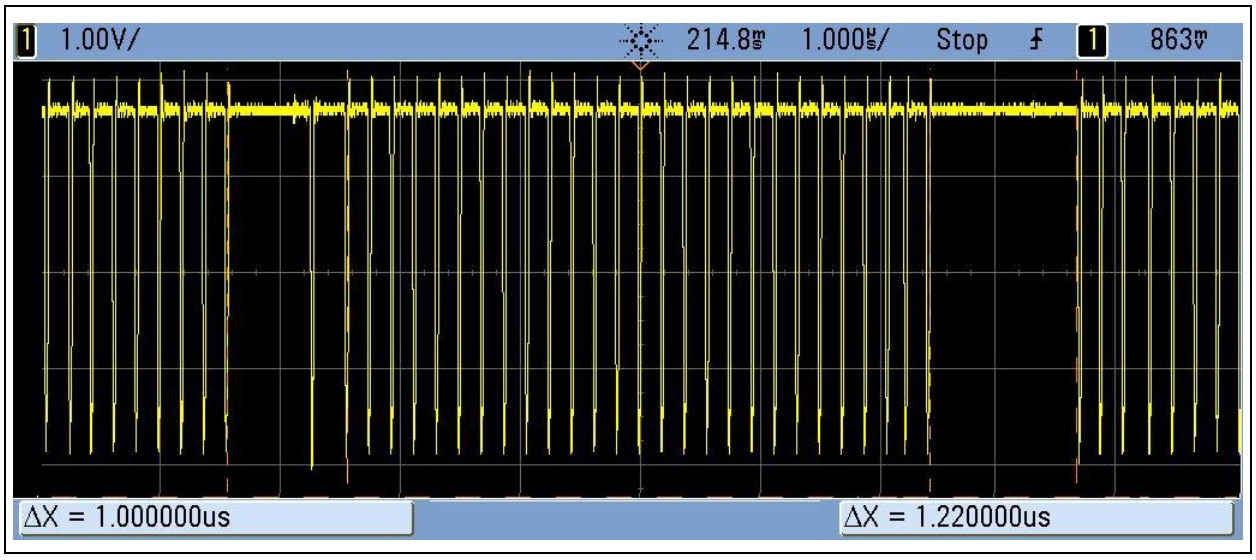
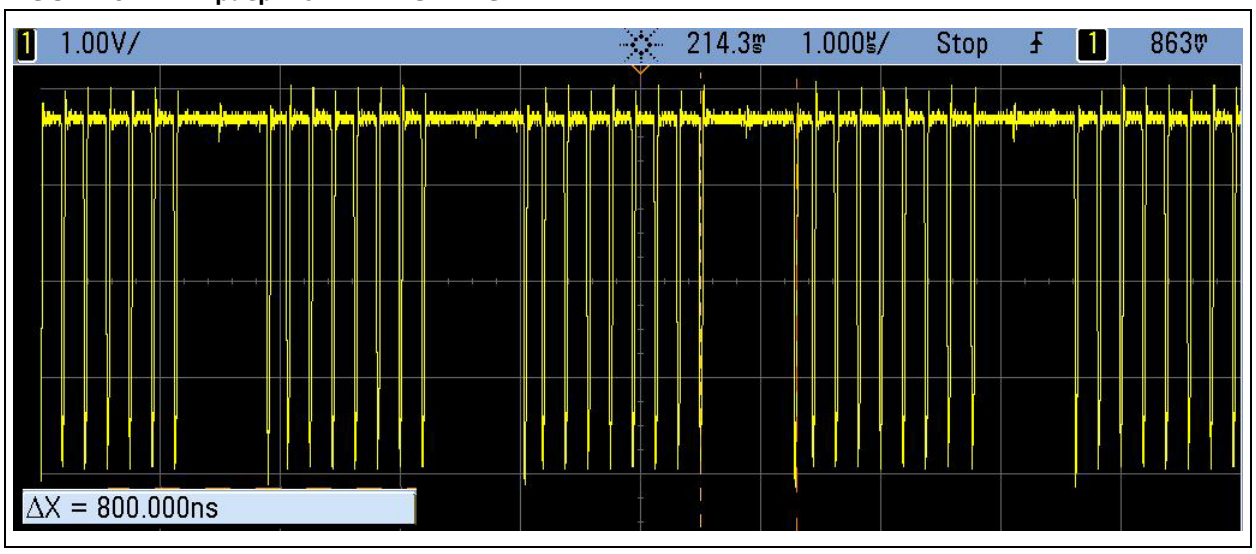


FIGURE 9: `putpixel` TIMING DIAGRAM



RENDERING NEW PIXELS ON THE FRAME

Rendering new pixels in the screen is just as important as refreshing the screen. This is performed by the CPU writing to the display buffer. If the frame is stored externally, the DMA transfer is suspended while the frame is being updated. This is necessary since there is only one PMP peripheral and it is being shared by the “virtual” graphics controller, or DMA transfer. This method does affect the refresh rate of the screen. The amount of pixel updates needs to be monitored to prevent too large of a refresh rate change, otherwise, the change will be perceptible by the human eye. This is done by using a pixel count variable within the “virtual” graphics controller that is updated on every pixel write and cleared during every DMA interrupt.

HARDWARE DESCRIPTION

The LCC Graphics Board is built with simplicity and low cost in mind. The on-board 512 Kbyte SRAM IC is only needed if the application is using the LCC Graphics Board for external memory. Otherwise, if a PIC32 MCU with enough internal memory is used with the board, no external memory is necessary. The equation to calculate whether a certain PIC32 MCU has enough volatile memory to store a data frame is provided in Equation 1.

In the following example, the internal memory demo for this board uses a starter kit that has a PIC32MX795F12L device, which has 128 Kbytes of SRAM. The demonstration uses QVGA resolution and 8 BPP Graphics; therefore, the equation would be:

76,800 = 320 x 240 x 1

This technique still leaves almost half of the internal SRAM for program use, while giving the designer 255 color choices to create a meaningful embedded user interface application.

If extra memory is needed, there are many different types to choose from. A parallel SRAM was used for this board, but any type of volatile memory can be used, as long as it is fast enough to support the required frame rate.

There are 15 jumpers on the LCC Graphics Board to change from Internal Memory mode to External Memory mode. Set jumper pins 1 and 2 when using internal SRAM. Set jumper Pins 2 and 3 when using external SRAM.

The LCC Graphics Board was made to be used with many different PIC MCUs. The board provides a starter kit connector to connect starter kits, such as the PIC32 USB Starter Kit II. In addition, there is a PICtail™ Plus connector to connect the board to an Explorer 16 Development Board.

Since LCC uses no graphics controller, this makes many graphics features available through software including simple alpha blending, scrolling, and the use of multiple frames. A user is no longer limited by the graphics controller used, but instead, the PIC microcontroller used.

EQUATION 1:

$$SRAM\ size = (x\ pixel\ resolution) \times (y\ pixel\ resolution) \times (Bytes\ of\ Color\ Depth)$$

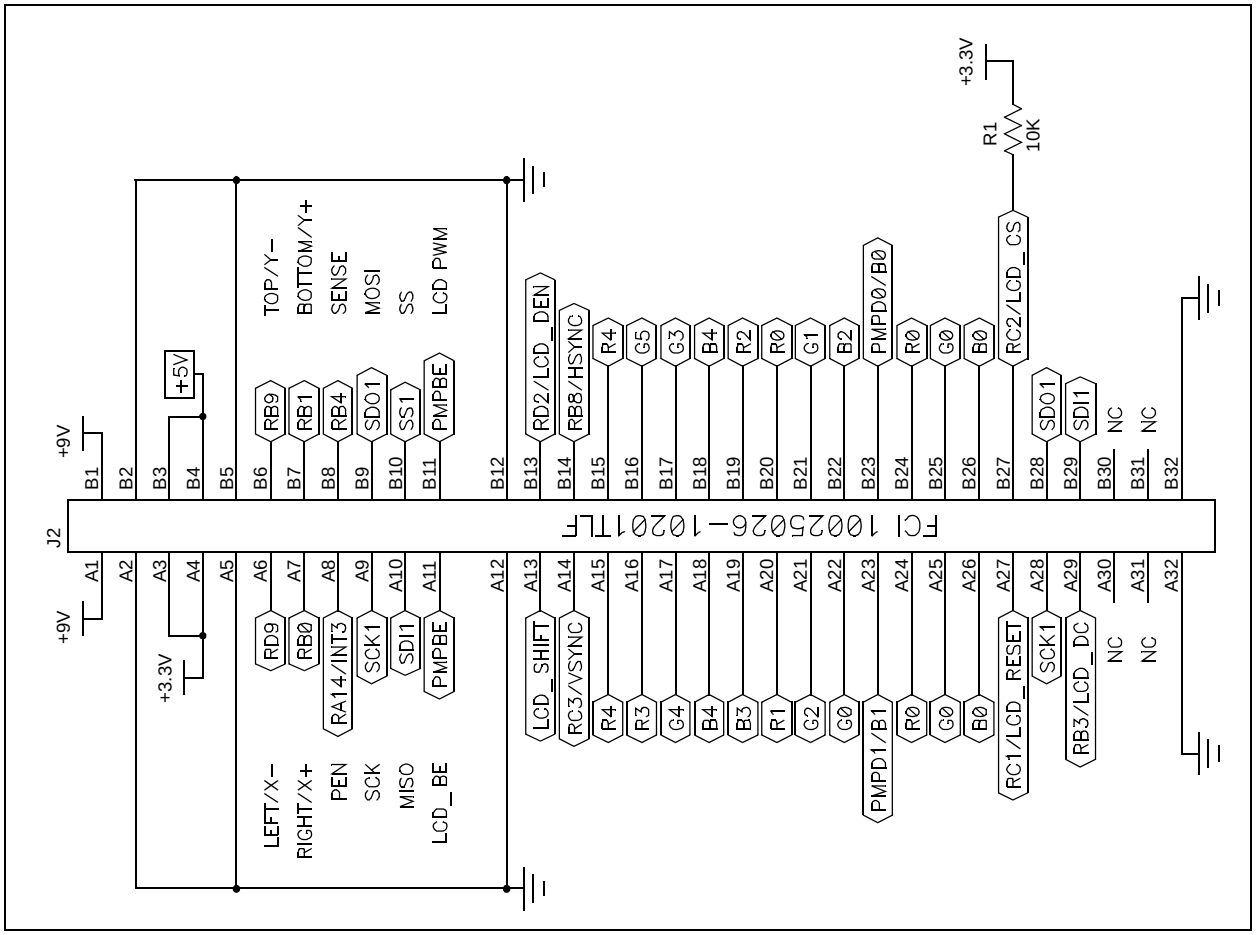
Display Connector

LCD panels are connected to the display connector, V 1 (see Figure 1). Descriptions of each display connector pin are listed in Table 1. The connector pins are shown in Figure 10.

TABLE 1: DISPLAY CONNECTOR PINS

Pin Name	Description
RD9, RB0, RB9, RB1	4-wire Resistive Touch Screen Lines.
R0-R4	Red Color Data Lines.
G0-G5	Green Color Data Lines.
B0-B4	Blue Color Data Lines.
RC1/LCD_RESET	LCD Reset Line.
RC2/LCD_CS	LCD Chip Select Line.
SCK1, SD01, SDI1, $\overline{SS1}$	SPI Lines for Communication for Timing or Touch Controllers.
RBS/LCD_DC	LCD DC Power Enable Line.
LCD_SHIFT	Pixel Clock.
RD2/LCD_DEN	Data Enable (DEN) Line.
RC3/VSYNC	VSYNC Line.
RB8/HSYNC	HSYNC Line.
PMPBE	Backlight Enable Line (PWM Capable).

FIGURE 10: DISPLAY CONNECTOR SCHEMATIC



SOFTWARE USAGE

The LCC Graphics Board uses the Microchip Graphics Library, which is a powerful library that makes creating a GUI such as the one presented in this application note fast and easy. The Microchip Graphics Library is available for download free-of-charge from the Microchip Application Libraries web page (www.microchip.com/MAL).

There are two main ways to use software with the Graphics Library. One method is using internal SRAM memory. This first method uses the write strobe of the PMP for the pixel clock. Jumper rows 1 and 2 on the LCC Graphics Board need to be set for this configuration. In this setup, all color is 8 BPP and no external SRAM is used. SRAM from inside the PIC32 MCU is continuously writing its pixel values to the PMP. For 8 BPP color, a 332 RGB color format is used (i.e., 3 color values for red, 3 for green, and 2 for blue). This is a common color format, which is due to the fact that red is an easier color for the human eye to detect than blue.

The other method is to use external SRAM memory. This method uses the read strobe of the PMP for the pixel clock. Jumper rows 2 and 3 on the LCC Graphics Board need to be set for this configuration. In this setup, all color is 16 BPP and the external SRAM contains the graphics frame that is continuously being read. For 16 BPP color, a 565 RGB color format is used (i.e., 5 color values for red, 6 for green, and 5 for blue).

In both methods, when connecting to an LCD panel with more than 16 color lines, the unused color lines are tied to the Most Significant Bits of the last color bit being used. This ensures that a full color scale from white to black can be achieved.

DEMONSTRATION SOFTWARE

Demonstration software specifically for the LCC Graphics Board is available on the LCC Graphics web page at www.microchip.com. The demonstration showcases both internal and external memory methods and contains many different user interface screens that could be created by the use of the Microchips Graphics Library and Microchip's Graphics Display Designer. Other software for the LCC Graphics Board is contained within Microchip's Graphics Library, where the LCC graphics driver is contained, which is able to run most of the demonstrations found in the Microchip Graphics Library.

CONCLUSION

In conclusion, this application note has provided a description of the Low-Cost Controllerless (LCC) Graphics PICtail Plus Daughter Board, which can be used to enable the implementation of a low-cost controllerless graphic system using a PIC32 microcontroller.

REFERENCES

- Microchip Application Libraries – www.microchip.com/MAL
- PIC32 Device Family – www.microchip.com/PIC32
- Microchip Graphics Support – www.microchip.com/graphics

APPENDIX A: SOURCE CODE

Software License Agreement

The software supplied herewith by Microchip Technology Incorporated (the "Company") is intended and supplied to you, the Company's customer, for use solely and exclusively with products manufactured by the Company.

The software is owned by the Company and/or its supplier, and is protected under applicable copyright laws. All rights are reserved. Any use in violation of the foregoing restrictions may subject the user to criminal sanctions under applicable laws, as well as to civil liability for the breach of the terms and conditions of this license.

THIS SOFTWARE IS PROVIDED IN AN "AS IS" CONDITION. NO WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE. THE COMPANY SHALL NOT, IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.

All of the software covered in this application note is available as a single WinZip archive file. This archive can be downloaded from the Microchip corporate Web site at:

www.microchip.com

APPENDIX B: SCHEMATICS

FIGURE B-1: LCC GRAPHICS PICtail™ PLUS DAUGHTER BOARD (SHEET 1 OF 2)

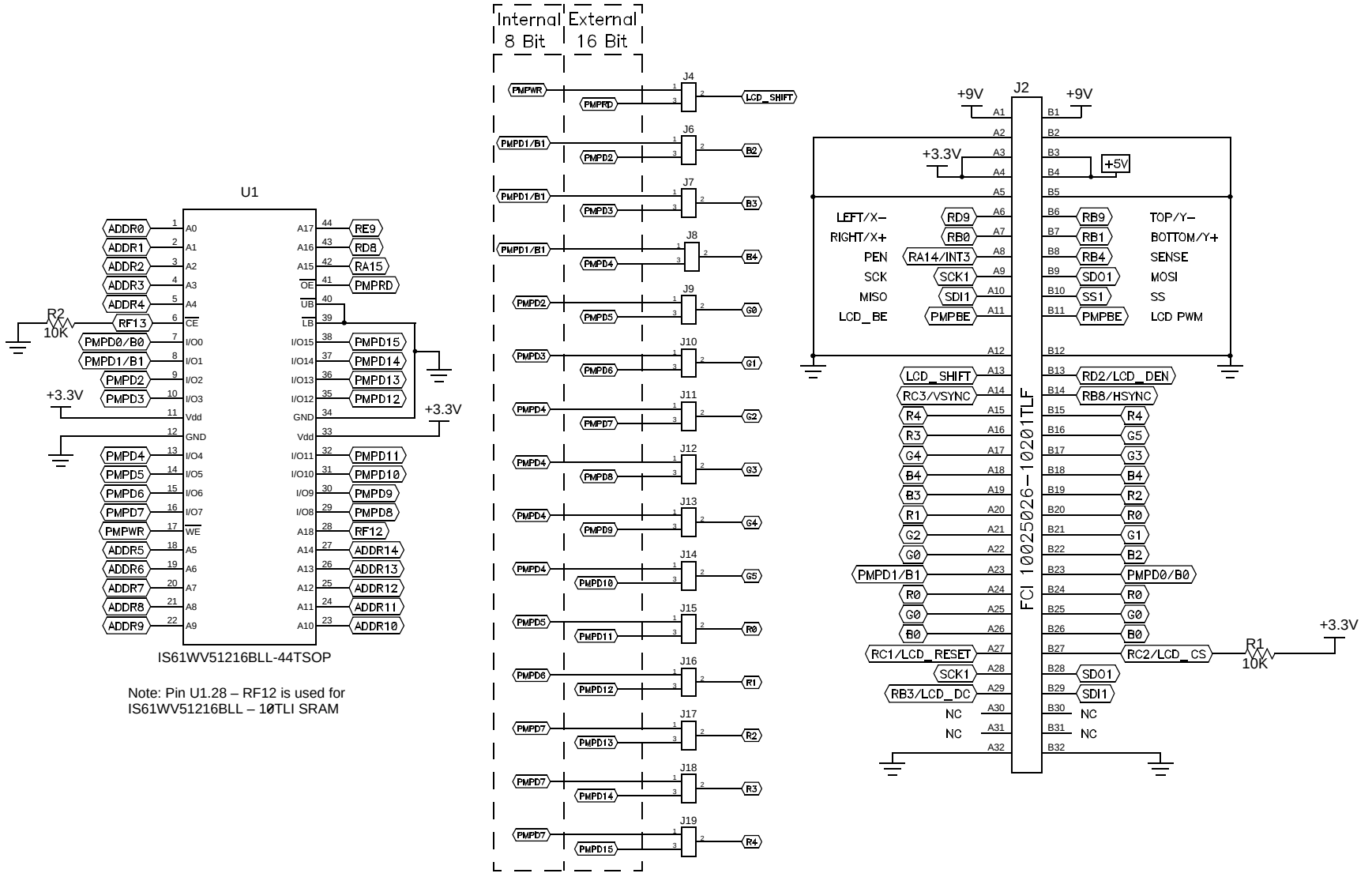
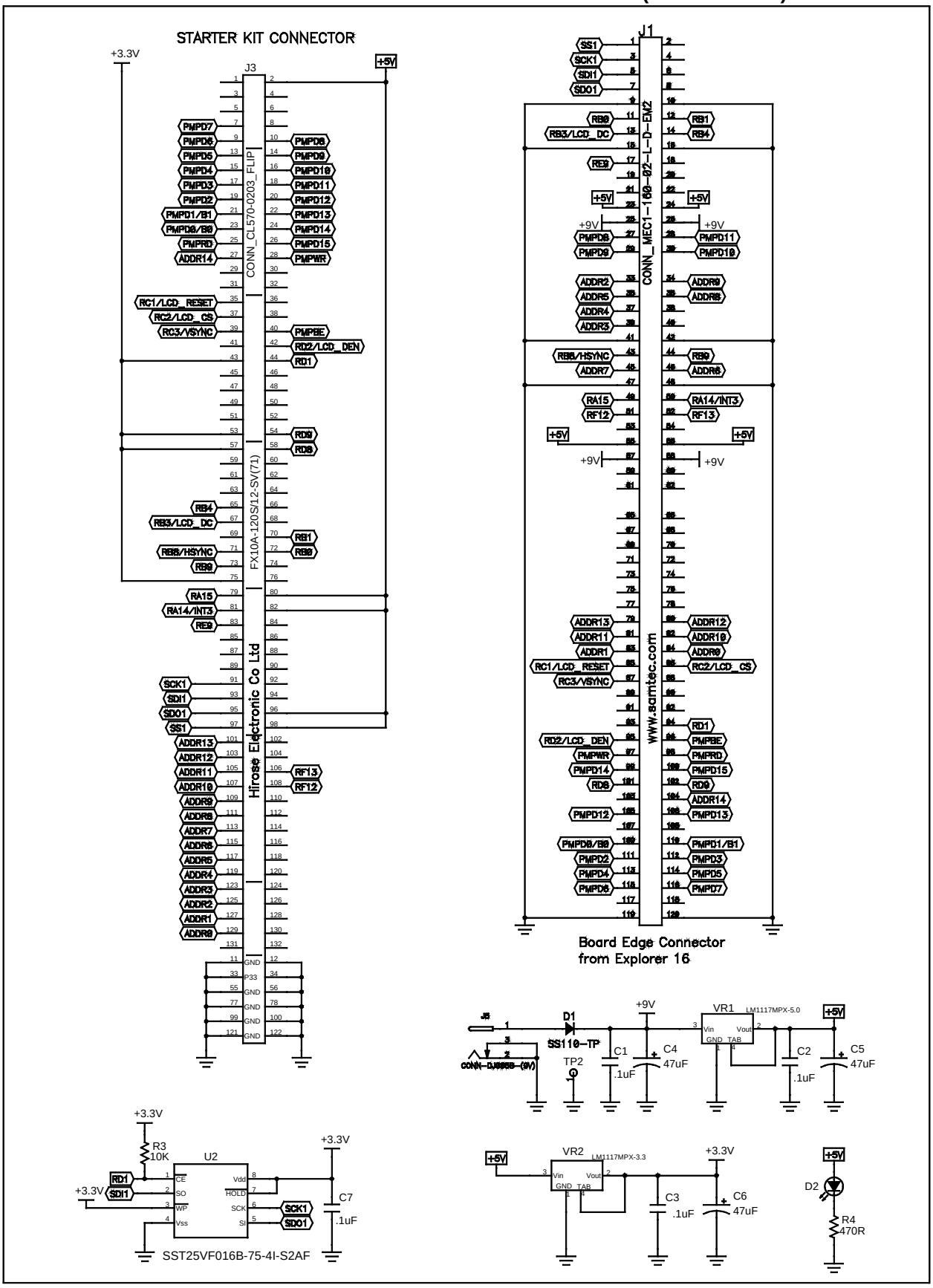


FIGURE B-2: LCC GRAPHICS PICtail™ PLUS DAUGHTER BOARD (SHEET 2 OF 2)



Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

Printed on recycled paper.

ISBN: 978-1-61341-361-6

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2009

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-2819-3180
Fax: 86-571-2819-3189

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-6578-300
Fax: 886-3-6578-370

Taiwan - Kaohsiung
Tel: 886-7-213-7830
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

05/02/11

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Microchip:
AC164144