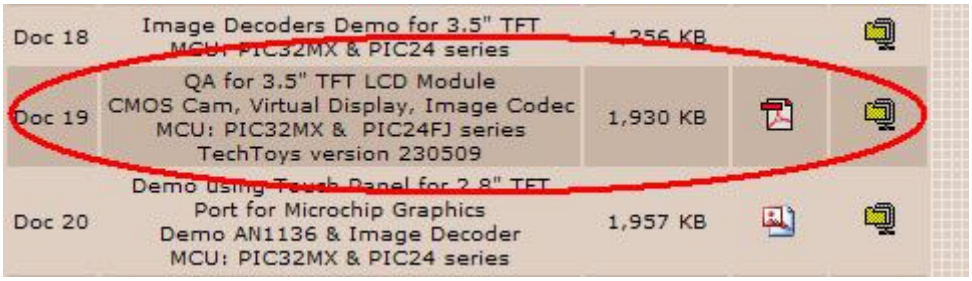
Using Microchip Graphics Library version 1.60

CMOS Camera Interface, Virtual Display, and Image Codec

1 How to use this demo:

From our web site under the following address, download the compressed file with full source code.

http://www.techtoys.com.hk/PIC_boards/PIC24-Eval-C/PIC24-Eval-RevC.htm



Development environment: MPLAB IDE v8.20
C32 compiler version: MPLABC32-v105-Evaluation
C30 v3.11b Student Edition
Microchip Graphics Library v1.60

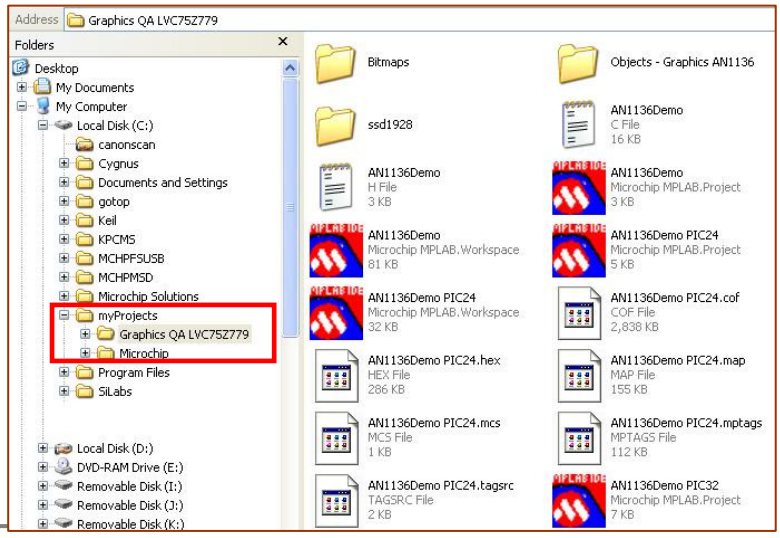
Unzip the project to any folder of your own choice. You may need to download WinRAR from www.Download.com to unzip this project.

After unzip, you will get two new folders.

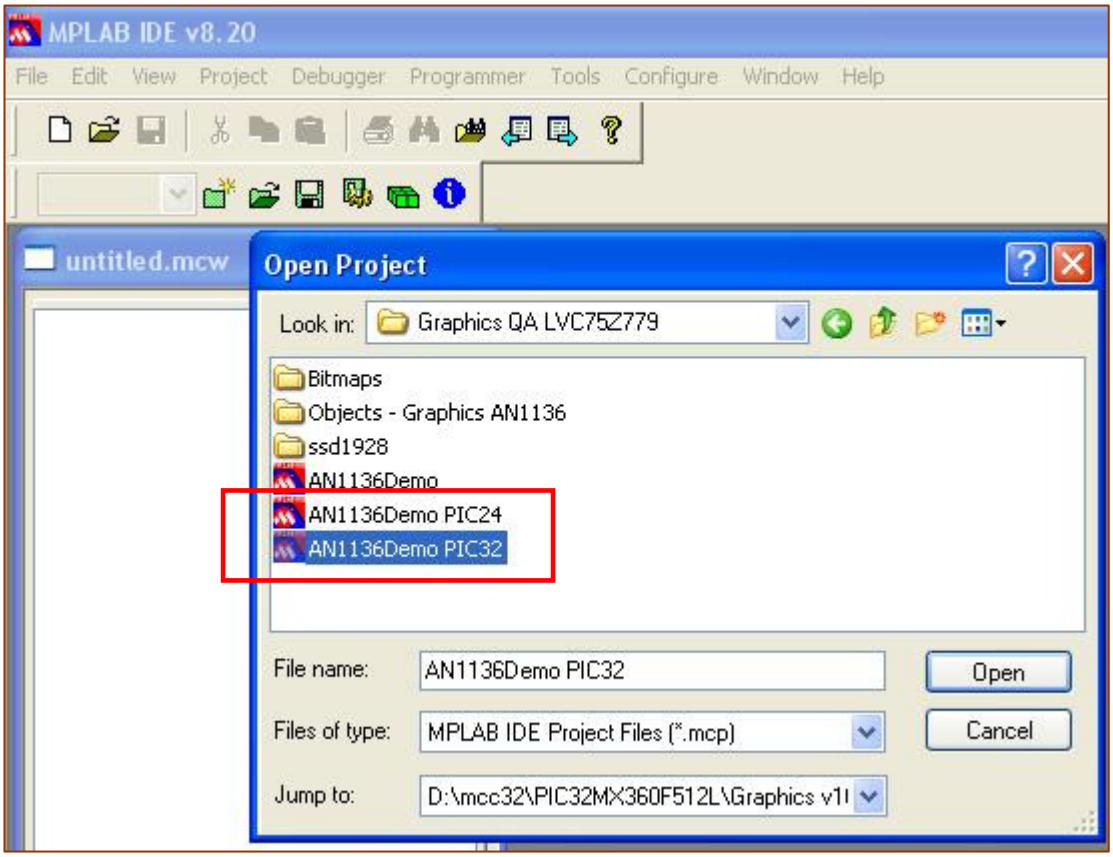
1. C:\myProjects\Graphics QA LVC75Z779
   This folder contains all project workspaces and ssd1928 folder which contains the driver for Solomon SSD1928
2. C:\myProjects\Microchip
   This folder contains the original Microchip Graphics Library v1.60.

Every project is a self-contained project. That means you don't need to keep track of every little change to the original graphics library file inside the Microchip folder (and sub-folders). It is because we will have to modify some of the source code of the graphics library, e.g. GOL.c and GOL.h and others for development of custom widgets or any particular application.

The downside is that, we need to keep multiple copies of the Graphics Library versions.



Launch MPLAB, under *Project* → *Open*, browse to C:\myProjects\Graphics QA LVC75Z779, select the PIC model you are using. Workspace AN1136Demo PIC32.mcp for PIC32MX series, and AN1136Demo PIC24.mcp for PIC24FJ series.



Under *Project* → *Build All* to generate the hex code required. The step remains is to debug or hex code download to your target development system by PICKit2, ICD2/3.

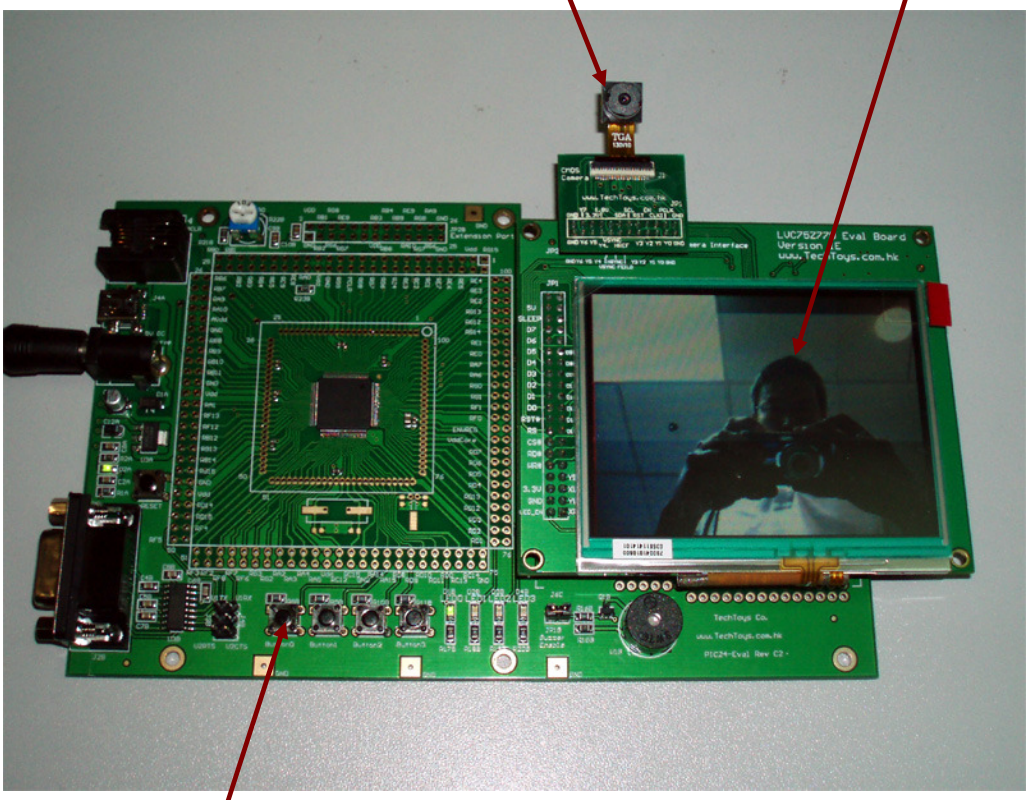
2 Prepare the hardware:

Plug-in 3.5" TFT module (LVC75Z779) with CMOS camera module as shown below.

Apply 5V DC (pin positive) to DC power jack. An alternative is to use a USB cable for power.

CMOS camera module (part # TGA130V10_BO)

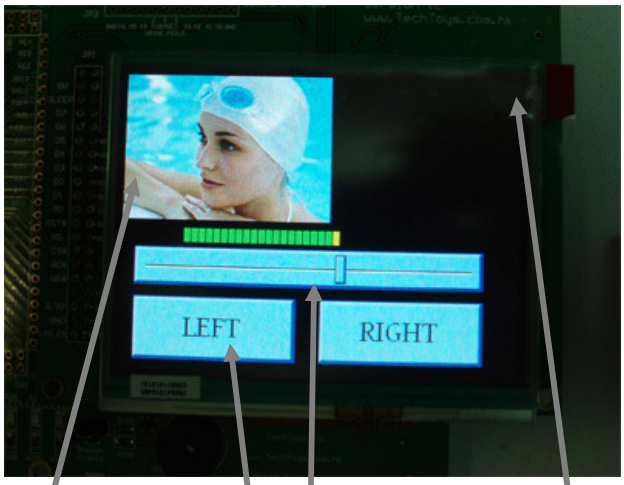
3.5" TFT Module
(Part # LVC75Z779 Eval Board (Version 1E))



This demo is a simple CMOS camera demo + Touch Panel demo with virtual display supported. The first thing you see is probably a self-portrait from the 1.3MPixel CMOS module. Once you are happy with this demo, click Button 0 to proceed to Touch Panel demo. If you don't have a CMOS camera, what you will see is a black page. Don't worry! Just click Button 0 to proceed anyway.

3 Image Decoder APIs has been added to this demo, thus we could use both interfaces for Touch Panel and JPEG/BMP codec.

Running this program will give you the following result.



Jpeg decoded & displayed

Standard widgets for Touch Panel

A number is being updated off-screen (virtual display)

Now, we may use the *LEFT* button or S*lide Bar* to scroll the whole display to the left. The counter which has been off-screen is now displayed

As we pan the page by repeat pressing "LEFT", a number being updated beyond the physical display area will be shown. This demonstrates how double buffering can be done, though there is not enough display RAM in SSD1928 for the whole page in 16-bit mode.

4 More on Virtual Display:

Virtual display is commonly called double buffer/back buffer. Please refer to the reference section for more resources.

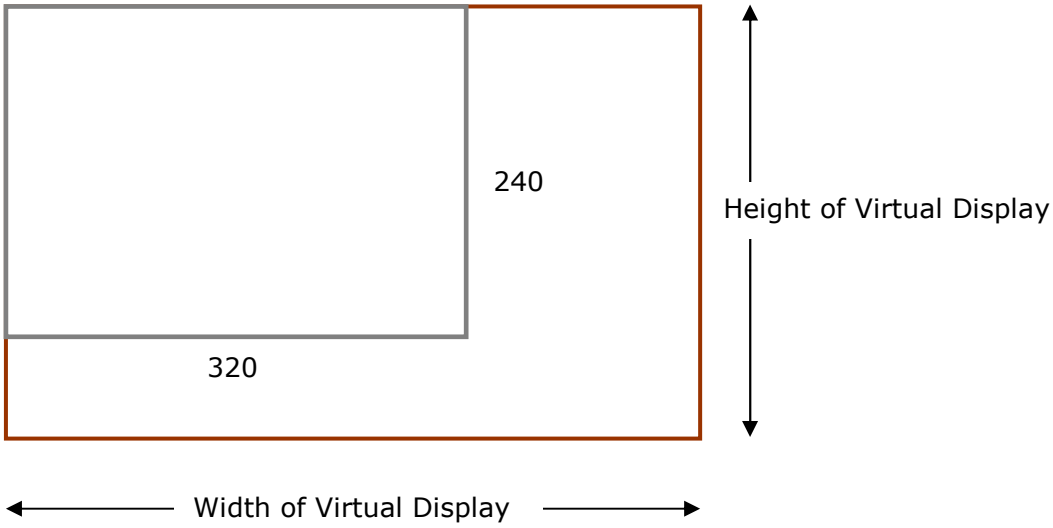
Image Processor Solomon SS1928 has a memory of 256KB for display buffer. That means the maximum display resolution in 16-bit per pixel is calculated as follows:

SCREEN_WIDTH*SCREEN_HEIGHT*2 < 256000

If we have a LCD panel of screen height 240, the maximum width of display we could use is:

SCREEN_WIDTH = 256000/(240*2) = 533.

Because we are using a QGA display of 320x240, we may configure the memory area larger than the physical display for screen panning and scrolling. The scenario is illustrated as below.



In the driver version 240309, the following files have been revised for this feature.

SSD1298.h
SSD1928.c
GraphicsConfig.h

It is the file GraphicsConfig.h that allows us to enable or disable the virtual display feature. Physical display area and virtual display area are also defined here.

```
#define DISP_HOR_RESOLUTION                         320
#define DISP_VER_RESOLUTION                         240
//Newly defined for SSD1928, virtual display width and height
//For Main Window Line Address Offset REG[78h] and REG[79h]
//Initialization under ResetDevice, ssd1928.c
//For visual page scroll & pan
#define USE_VIRTUAL_DISPLAY

#ifdef USE_VIRTUAL_DISPLAY
#define DISP_VIRTUAL_HOR_RESOLUTION         480
#define DISP_VIRTUAL_VER_RESOLUTION         272
#else
#define DISP_VIRTUAL_HOR_RESOLUTION         DISP_HOR_RESOLUTION

#define DISP_VIRTUAL_VER_RESOLUTION         DISP_VER_RESOLUTION
#endif
```

Comment this statement to disable virtual display.

New macros for virtual display:

GetVirtualMaxX()
GetVirtualMaxY()

API:

void SetPageOrigin(BOOL clipLeftRgn, BOOL clipTopBottom, SHORT x, SHORT y)

Heavy comments have been put on files SSD1928.h and .c files. User must refer to them for instruction in details.

References:

1. CMOS camera module (OV9650 CMOS sensor) http://www.techtoys.com.hk/Components/OV9650_MOD/OV9650%20CMOS%20Camera%20Module.htm
2. Wiki on YUV standard. http://en.wikipedia.org/wiki/YUV
3. Compressed Image File Formats JPEG, PNG, GIF, XBM, BMP by John Miano Addison Wesley
4. Double Buffering / Back Buffer: http://java.sun.com/docs/books/tutorial/extra/fullscreen/doublebuf.html