VGA Monitor Interface with ARM7 Microcontroller

Deepali P. Sarag
Department of Electronics Engineering, H.I.T., Aurangabad, India

Abstract:
In this paper, we presented the embedded mini motherboard which control VGA monitor text mode operation. This design is used for displaying monitoring parameter on VGA screen where graphics and color requirements are minimum; we need not to use CPU for only displaying parameter on screen. We can connect this board directly to any system using RS232 port. It will accept data serially and display it on the VGA screen.

Keywords: VGA signal, LPC 2148, Serial Communication

1. Introduction
System has provided RS-232 port for interfacing this board to external world. In this system ARM7 microcontroller is act as graphics controller, which will accept data serially through any system and store this data on internal RAM in the frame format and send that data to VGA monitor, it controls VGA operation. In this system an ARM7 microcontroller board is used for driving VGA display. This design is used for displaying different parameter and indicators, where cost is constrain.
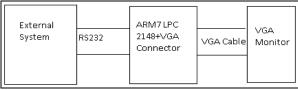
2. System Overview

Figure 1

There is no shortage of VGA graphics driver chips available but many are here today and gone tomorrow in this fast-paced world. Many are designed for PCI buses, not something that can be considered low-cost embedded. All of these chips require external memory and so on and besides we don't need all that fancy stuff that it can do. Most of the time there small requirement of only write text, draw a few lines, and change a few colors.

In this ARM7 microcontroller board, RS-232 port, VGA connector circuit is used as shown in block diagram, PC as External System is used for testing this system. RS-232 port of ARM7 board is connected to RS-232 port of PC through DB-9 cable secondly we will connect VGA port of ARM7 board to VGA monitor by DB-15 VGA cable. Data is to be display on the screen must send from PC to ARM7 Board through serial terminal. ARM7 Board will Process that data for displaying it on VGA monitor Screen.

With a little bit of software magic the LPC2148 ARM7 Core Microcontroller can be pressed into generating the VGA signals needed to drive a monitor plus handle all the font and graphics rendering. Memory requirement for instance 16 million colors sounds great but that means 3 bytes per pixel, 8-bits for red, 8 for green, and 8 for blue. If we tried to display 640x480 pixels that would be 307,200 pixels, which would then be multiplied by 3 bytes/pixel which then requires almost 1MB of RAM. This requires a larger package to handle the address/data lines to interface to external RAM. So, we will use 32K onboard RAM which means we could do 192 pixels wide by 240 lines overlaid on standard 640x480 VGA timing. If we fore-go any video DAC arrangements that would be necessary for analog color we could devote 1-bit per color, which would result in 8 colors. In this implementation we use 4-bits per pixel which requires around 23K of the 32K RAM. This is do-able, simple and a heck of a lot better than a tiny monochrome LCD.[1,2]
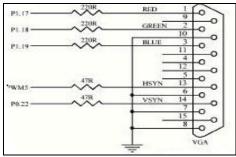
3. Circuit Description

Figure 2

- **Power Supply:** ARM7 board is power by +5V/2A.ARM7 Baord Components Discription: 1)RS-232 port 2)VGA port 3)Reset button 4)ISP button 5)RS-232 cable
- **RS-232 port:** This is UART0 port of ARM7 microcontroller. This Port is use for Programming LPC2148 along with Reset button and ISP button. This port is also use for communicating with PC. This port will accept data through PC and display that data on VGA monitor through VGA port.
- **VGA port:** This is a VGA port of LPC2148 microcontroller. This port is connected to VGA monitor.
- **Reset button:** This is reset button of LPC2148 microcontroller. This button is used to reset LPC2148.This button is also used program LPC2148 along with RS-232 port and ISP button.
- **ISP button:** This is ISP button of LPC2148 microcontroller. This button is used for programming lpc2148 along with RS-232 and Reset button.
- **RS-232 port Circuit Description:** In this circuit TXD0 (pin 19)and RXD0(pin 21) of LPC2148 microcontroller is connected to TXD0 (pin 10)and RXD0(pin 9) of MAX3232. PC_TX0 (pin 7) amd PC_RX0 (pin 8) of MAX3232 is connected to PC_TX0 (pin 3) amd PC_RX0 (pin 2) of DB-9 female connector. pin 5 of DB-9 female connector is connected to Ground Signal.[1]

4. VGA Circuit Description

In this circuit pin 12 (P1.17),pin 8 (P1.18),pin 4 (P1.19) ,pin 1 (P0.21),pin 2 (P0.22) are connected to pin 1,pin 2,pin3,pin 13 ,pin 14 of DB-25 female VGA connector. pin 6,pin 7,pin 8,pin 10 of DB-25 female connector is connected to ground signal.

VGA analog monitors require 3 channels of RGB video data driven from 0 to 0.7V. As the monitor inputs are terminated to 75ohms we can place a series resistor in each line to achieve the correct level without overdrive. The horizontal and vertical sync lines are driven directly.

4.1. Signal Generation

- ➢ A VGA video signal contains 5 active signals horizontal sync: digital signal, used for synchronization of the video.
  - • vertical sync: digital signal, used for synchronization of the video-
  - • red (R): analog signal(0-0.7v),used to control the color.
  - • green (G): analog signal(0-0.7v),used to control the color.
  - • blue (B): analog signal(0-0.7v),used to control the color.
- ➢ By changing the analog levels of the three RGB signals all other colors are produced.
- ➢ The electron beam must be scanned over the viewing screen in a sequence of horizontal lines to generate an image. The RGB color information in the video signal is used to control the strength of the electron beam.
- ➢ The screen refresh process begins in the top left corner and paints 1 pixel at a time from left to right. At the end of the first row, the row increments and the column address is reset to the first column. Once the entire screen has been painted, the refresh process begins again.
- ➢ The video signal must redraw the entire screen 60 times per second to provide for motion in the image and to reduce flicker: this period is called refresh rate. Refresh rate higher than 60Hz are used in PC monitors.
- ➢ In 640 by 480-pixel mode, with a 60Hz refresh rate, this is approximately 40ns per pixel. A 25 MHz clock has a period of 40ns.
- ➢ The vertical sync signal tells the monitor to start displaying a new image or frame, and the monitor starts in the upper left corner with pixel(0,0)
- ➢ The horizontal sync signal tells the monitor to refresh another row of 640 pixels.
- ➢ After 480 rows of pixels are refreshed with 480 horizontal sync signals, a vertical sync signal resets the monitor to the upper left corner and the processor continues.
- ➢ During the time when pixel data is not being displayed and the beam is returning to the left column to start another horizontal scan, the RGB signals should all be set to black color(all zero).

➢ In a PC graphics card, a dedicated memory location is used to store the color value of every pixel in the display. This memory is read out as the beam scans across the screen to produce the RGB signals.
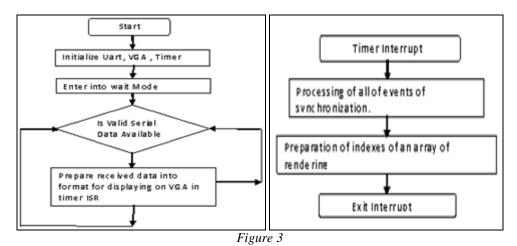
On the LPC2148 a PWM channel is setup to generate the horizontal sync pulse and also interrupt the processor. The interrupt routine then outputs the video data for that line and keeps track of which line it's on for blanking and vertical sync purposes. The vertical sync pulse only has to occur once a frame at 60 frames a second so it is generated from the interrupt routine. Normally, all this video timing generation would keep most processors too busy to do anything else, not so with the LPC2148 for accurate timing generation write code in assembly language.

In between all the actual video data there are lots of blanks, so the processor doesn't really have to do anything except respond to the interrupt, do a little housekeeping, and then it can get back to what it was doing. This allows enough time to service all the normal interrupt and run foreground processes such as font rendering and running actual user code.[2,3]

5. Working

Initially when power is on, embedded board initialize the UART, VGA, and waits for interrupt for serial communication a dock light software is used. After interrupt, µc checks valid serial data if data is valid checksum is calculated if matches then generates a VGA signal for each character & sends it to VGA monitor for displaying.

Then µc remains into wait mode till further interrupt. As limitation of RAM of µc if I tried to display 640x480 pixels that would be 307,200 pixels, which would then be multiplied by 3 bytes/pixel which then requires almost 1MB of RAM. So, this requires a larger package to handle the address/data lines to interface to external RAM. Therefore for simplicity we design screen of [23 * 13] matrix, with specifically seven foreground color & seven backgrounds color is done. [1]

Figure 3

Intel Hex Format for VGA
- " : " Start of String
- 2-bytes ROW address - 2-byte ROW address is 00-21 (00,01,02,03,04,05, 06,07,08,09,10,11, 12, 13, 14,15,16,17, 18, 19, 20, 21)
- 2-byte COLUMN address - 2-byte COLUMN address is 00-12 (00,01,02,03,04,05,06,07,08,09,10,11,12)
- 1-byte for- ground color is 0-7
- 1-byte back-ground color is 0-7
- Number of character to be display on line it must be 2-byte.
- n-Character is character to be displayed on one ROW Max 22 Character you can display Display On single ROW you have such 13-ROW,So you can Display Max 22 X 13 =330 Character on VGA Monitor.
- Enter or 0Dh or ' \n '(end of Command)[4]

Example:
- :00001007414243444546470D
- :00002007414243444546474FD
- :000030074142434445464ED

Calculation of Check Sum:
add all 2-byte hex pair for calculating Check-Sum and ignore carry bit from result ,subtract result from 0x00,you will get checksum value in hex format then convert 8-bit result in 2-byte ASCII. (upper nibble ASCII 1-byte and lower nibble ASCII 1-byte.)

Example:
To send following string on VGA monitor "ABCDEFG" in green latter then string will be
: 000020074142434445464647

Calculation of check sum for this string
00+00+20+07+41+42+43+44+45+46+47=0x203
Ignore carry byte 2 now check sum byte is=0x03
Now subtract check sum byte from 0x00 =0x00-0x03
Now check sum is=0xFD (actual result 0x00-
0x03=FFFD, but consider only first byte FD
String to be send VGA Monitor is=:000020074142434445464 7FD
Add Enter(0Dh) at end of string =:00002007414243444546 47FD (0Dh or <CR>)
Example triangle wave :000230160203020302030203020302030203020302030203020381<CR> [4]

6. Result and Testing

- HEX format is designed in the software
- VGA monitor layout is designed for 22rows and 13 columns.
- Addressing maximum resolution of 192*240 pixels over standard 640 *420 pixels
- Initialize serial link and only valid data is send verification is done by check sum byte

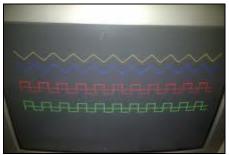Following waves will be generated for the above example

Photo 1: Wave Generated

7. Conclusion

One feature of ARM graphics processing is very low power and, based on the graphics quality of the end products mentioned, its performance is good as well.
The integrated display controller is capable of directly driving almost any LCD display with an RGB or STN/CSTN interface, which includes a wide range of TFT, STN, and some OLED displays. These features effectively create a complete graphical subsystem that is fully integrated on the same chip as the MCU, driving up to 640 x 480 (VGA) display resolution.

8. References

1. ARM system developers guide by Andrew N. Sloss
2. http://microvga.com/conio-lib
3. http://www.webopedia.com/TERM/V/VGA.html
4. http://en.wikipedia.org/wiki/Video_Graphics_Array
5. http://www.docklight.de/pdf/docklight_manual.pdf
6. http://www.rickard.gunee.com/projects/video/pic/howto.php
7. http://www.digikey.com/enUS/articles/techzone/2012/nov/microcontrollers-with-graphics-display-capabilities-are-in-high-demand