International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol. 3, Issue. 4, Jul - Aug. 2013 pp-2489-2491 ISSN: 2249-6645

Colour Rendering For True Colour Led Display System

N. Venkata Narendra[1], Dr. T. C. Sarma[2]
(Department of Electronics, Sreenidhi Institute of Science and Technology, India)

ABSTRACT: *This paper deals finding primary colours (Red, Green, Blue) depth values for each pixel in selected image or text to display the true colour of an image on the RGB LED display panel and transmitting those values to the controller via Ethernet. True colour having the colour depth of 2^8 (=256) for each primary colour in RGB colour space. So, we are able to display 256X256X256=1.6 billion colours on the RGB LED display panel.*

Keywords: *Ethernet, MATLAB, MBI5030 LED Driver, RGB LED panel.*

I. Introduction

Light Emitting Diodes (LEDs) are commonly used as indicator lights in electronic devices such as televisions and computers. LEDs, as with all semiconductors devices, have material and process variation which yields products with corresponding variation in performance. LEDs are binned and packed to balance the nature of manufacturing process with needs of the lighting industry. Color rendering is essentially an experiment where secondary color is generated by adding the appropriate proportion of primary colors.This paper describes how we are displaying the image or the video on RGB LED panel. Firstly we are generating the RGB values of image or video for each pixel, then we perform the gamma correction for RGB pixels separately, then after the operation we transfer each RGB values of each pixel via the frames using Ethernet to the STM controller. In the controller it processes and gives each pixel to LED driver. From the driver each pixel values are passed to the RGB LED panel and the image is displayed.

II. Accompanying Hardware and Software

The firmware describes in this paper is included with the STM32F207V Evaluation board. This board is specifically developed for hardware circuitry that is exhibited on the board. This board also includes a PC software that displays color output produced by the board by using Kiel software.

III. High Level Firmware

This firmware performs four tasks (refer to figure 1).The chart in this figure represents the procedural execution of the associated project.

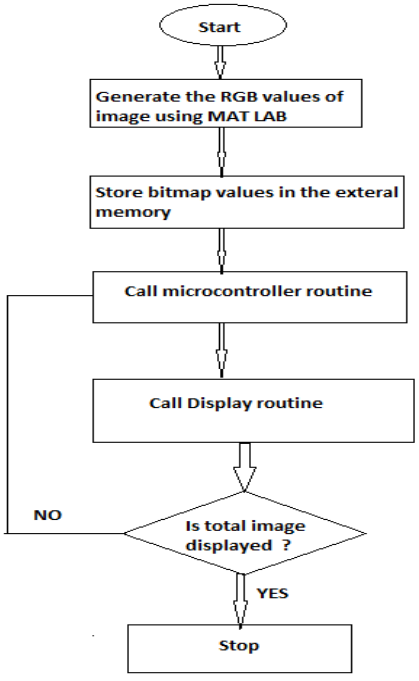


Fig 1.High Level Flow Chart of Firmware Execution

The important tasks are generating the RGB values for each pixel of an image and displaying it on the LED panel. And the other primary tasks are storing the bit map values and performing the gamma correction. Give three bit inputs (8 different combinations) to the decoder on display board to select each row in top & bottom 8 rows of one brick (display board consist of two active bricks).Activate first row LEDs of two bricks at a time and load internal buffers with respective gray values in LED drivers using D latch of LED driver .Set Global clock pin to glow all LEDs at a time with respective

gray values. Activate second, 3rd up to 8th row of two bricks and repeat process of glowing LEDs with the row refresh time of 312 micro seconds.

Before exploring the displaying of image, we must know more about the inputs and outputs of the system.

IV. Colour Rendering Inputs

This firmware uses RGB values of each pixel as the input to be serviced. First we should select an image, for that image we are going to generate the RGB values for the each pixel depends on the image resolution. This is done by using the Mat lab software. By using the image read function we are going to read the data of the particular image and by image pixel format we are going to generate R value separately, G value separately and B value separately. Then for this values gamma correction is performed. The gamma correction is a name of a nonlinear operation used to code and decode luminance or tristimulus values in video or image systems i.e it is used to display the text, image in accurate manner. The gamma correction is done to each pixel value of R,G,B by using the expression:

$\Gamma = R * G\_LUT[R] / 0xFFFF$ (For Red Pixels) .....(1)

And similarly in case of the green and blue colors. After the gamma correction these values are stored in external memory .By using the Ethernet we are going to the bitmap values from memory to STM controller via frames.The Ethernet peripheral enables the STM32F207V to transmit and receive data over Ethernet in compliance with the IEEE 802.3-2002 standard. Ethernet frames read from the system memory are pushed into the FIFO by the DMA. The frames are then popped out and transferred to the MAC core. When the end-of-frame is transferred, the status of the transmission is taken from the MAC core and transferred back to the DMA. The Transmit FIFO has a depth of 2 Kbyte. FIFO-fill level is indicated to the DMA so that it can initiate a data fetch in required bursts from the system memory, using the AHB interface. The data from the AHB Master interface is pushed into the FIFO. When the SOF is detected, the MAC accepts the data and begins transmitting to the MII. The time required to transmit the frame data to the MII after the application initiates transmission is variable, depending on delay factors like IFG delay, time to transmit preamble. Then it gives the bitmap data to controller via the com port.

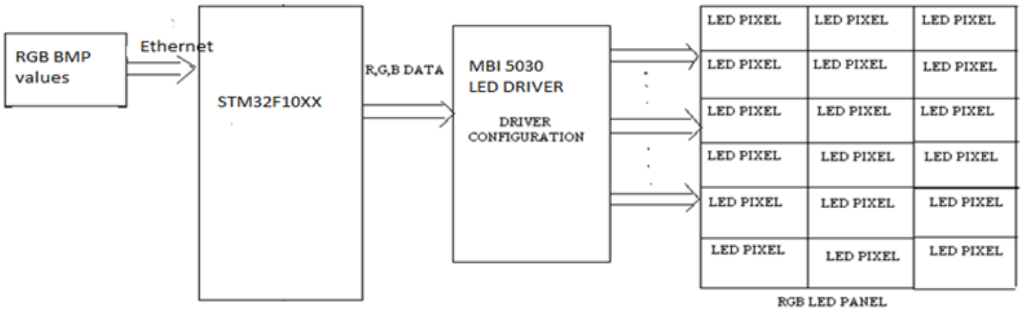


Fig 2.Firmware Input and Output

V. Colour Rendering Outputs

The challenge in developing this system lies in controlling LEDs .Here the controller used is STM32F207V.The features available in this controller are JTAG,LCD , Ethernet ,Decoder. The bitmap data from the controller is MBI5030 LED driver bit by bit.MBI 5030 is designed for LED video applications using internal pulse width modulation control with selectable 12-bit color depth. This features a 16-bit shift register which converts serial input data into each pixel gray scale of output. Suppose in order to display a 8*8 resolution image, their is need of 64 LEDs to drive 64 RGB LEDs .If we are using total 64 LED drivers in system it will become too complex as well as cost will be more, so in order to this drawback we are going use a decoder so that number of LED driver should use will be less. Give three bit inputs (8 different combinations) to the decoder on display board to select each row in top & bottom 8 rows of one brick (display board consist of two active bricks).Activate first row LEDs of two bricks at a time and load internal buffers with respective gray values in LED drivers using D latch of LED driver .Set Global clock pin to glow all LEDs at a time with respective gray values. Activate second, 3rd up to 8th row of two bricks and repeat process of glowing LEDs on RGB LED panel with the row refresh time of 312 micro seconds. so that image or video can seen by us clearly as refresh rate is very low.

VI. Mathematical Model

Fig.2 shows the inputs of the firmware and translated outputs .The mathematical functions in this section describe how the dimming value are obtained from CIE color space in order to a specific color. For some multi-LED applications, mixing white LEDs from a variety of bins is a cost effective way to achieve good color quality while minimizing LED costs. In this Illustration we show four LEDs can achieve the same perceived result as if four LEDs from one of the central sub-bins were used instead. Mathematically the results come because color and flux are additive. LEDs are typically characterized by chromaticity (x,y in the 1931 CIE color space) and flux (Φ =Y). Luminous flux is an additive metric just as perceived color is additive.

Tristimulus values, used in color mixing math, can be calculated as follows:

$X=x*(Y/y)$ ----------(2)

$Y=Y$ ----------(3)

$Z=(Y/y)*(1-x-y)$------(4)

The combined color is the result of the added tristimulus values:

Xmix=X1+X2+X3+X4
Ymix=Y1+Y2+Y3+Y4
Zmix=Z1+Z2+Z3+Z4

After finding Xmix, Ymix, Zmix, calculate xmix, ymix, zmix

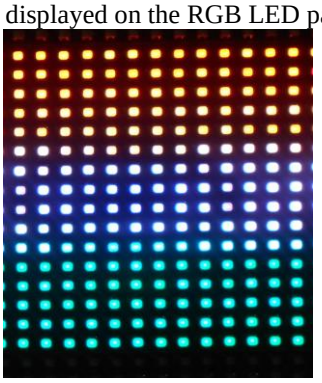
xmix = Xmix/(Xmix+ Ymix+Zmix)
ymix = Ymix/(Xmix+Ymix+Zmix)
Φmix = Y1+Y2+Y3+Y4

VII. Conclusion

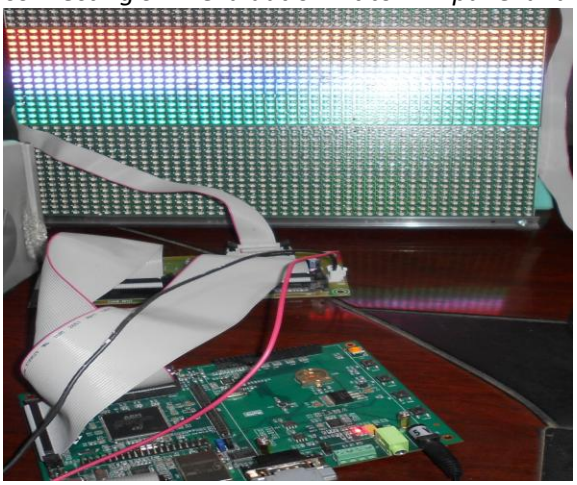
The final two primary tasks in fig 1 are generating the RGB values of the image with the dimming values for the each pixel and passing it to the STM controller on evaluation board via the Ethernet and gamma correction is performed and the bit by bit data is send to RGB LED panel via the LED driver and the image , video can be displayed on the LED panel which has true colour. This application mainly useful in Large LED Screens for Stadiums, Railway Stations, Bus stands etc.

VIII. Output

9.1 This is the one color rendered output, that is displayed on the RGB LED panel.



9.2 This is the procedure of connecting STM evaluation kit to LED panel and the output is display panel.



References

[1] Golbert A. and Israel I."Trends in microprocessor cache architectures" IEEE on March,1991
[2] Jaber Hasan and Simon S. Ang, "A High-Efficiency Digitally Controlled RGB Driver for LED Pixels" IEEE paper.
[3] Qin Song, Yan Sun "ARM9-based Control System for LED Large Screen Display" 2010 IEEE paper
[4] A guided tour of color space, Charles poynton,Newyork,1997
[5] Firmware for RGB color mixing .-EZ-color,AN16035,2010
[6] Cree's LED color mixing-Basics and Background,2010
[7] STM32F2XX datasheet "Doc ID 15818 Rev 8" www.stm.com.
[8] Yao Yingbiao, Zhang Jianwu, and Zhao Danying "Survey on microprocessor architecture and development trends" IEEE on November, 2008
[9] Microblock MBI5030 16-channel constant current LED driver
[10] http://www.colourbasics.com.