



Application Note

AN_200

Vinculum-II Scrolling Text Application Using LED dot matrix display

Version 1.0

Issue Date: 2012-02-09

This application note provides an example of using the FTDI Vinculum-II (VNC2) to communicate with a USB Flash drive to display text and graphics held on the Flash drive, on a LED dot matrix display. Drivers and source code are also provided (downloadable from the FTDI website)

Use of FTDI devices in life support and/or safety applications is entirely at the user's risk, and the user agrees to defend, indemnify and hold harmless FTDI from any and all damages, claims, suits or expense resulting from such use.

Future Technology Devices International Limited (FTDI)

Unit 1, 2 Seaward Place, Glasgow G41 1HH, United Kingdom

Tel.: +44 (0) 141 429 2777 Fax: + 44 (0) 141 429 2758

Web Site: <http://ftdichip.com>

Copyright © 2012 Future Technology Devices International Limited

Table of Contents

1	Introduction	3
1.1	Overview	3
1.2	Hardware Requirements.....	4
2	Operation	5
2.1	Using VPlayer Display Software	5
2.2	Updating the SEQUENCE.TXT file.....	5
2.3	VPlayer application demonstration	6
3	Hardware	7
3.1	Chipset	7
3.2	Interfaces	7
4	Firmware	9
4.1	File Types.....	9
5	Source Code for the VNC2 Application Firmware.....	10
5.1	vPlayerFlashDisk.C	10
5.1.1	SPI_Master_Init.....	10
5.1.2	main	10
5.1.3	Thread Flash Disk	11
5.1.4	Thread displayOut.....	13
6	Byte ordering in the frame buffer.....	15
6.1	2 bits per pixel colour palette	15
6.2	Display buffer format for 2 blocks	16
6.3	Display buffer format for a LED module	16
6.4	Display buffer format for a LED panel.....	17
6.4.1	LED panel stacking	17
6.4.2	LED column bit shifting direction.....	18
6.4.3	rowData Array.....	18
6.4.4	Bitmap file format.....	20
7	VPlayer Display Software	22
7.1	Features supported.....	23
8	Contact Information.....	24



Appendix A – References	25
Document References.....	25
Acronyms and Abbreviations	25
9 Appendix B – LED module schematics (control board and dot matrix board)	26
Appendix C – List of Tables & Figures	32
List of Tables	32
List of Figures	32
Appendix D – Revision History	33

1 Introduction

This application note describes an application that uses Vinculum-II to display text and graphics, obtained from a bitmap file in a USB Flash drive, on a LED dot matrix display. This solution is better than the conventional method of implementing the features in the firmware. In this method all the features are implemented in the PC software which creates the bitmap file. The features implemented in the PC software are; fonts, languages and graphics support. The firmware passes the raw data in the USB Flash drive to the display with no processing on the data. For a given LED panel the firmware is fixed and does not change for supporting multiple fonts, languages and graphics. So it is easy to update the display by replacing the content in the USB Flash drive. The figure below shows a Vinco Development Module used to display text on a LED dot matrix display

The source code for the application firmware is provided as an example and is neither guaranteed nor supported by FTDI. The source code for the firmware can be downloaded from the following location on the FTDI website: <http://www.ftdichip.com/Support/FTSwExamples.htm>

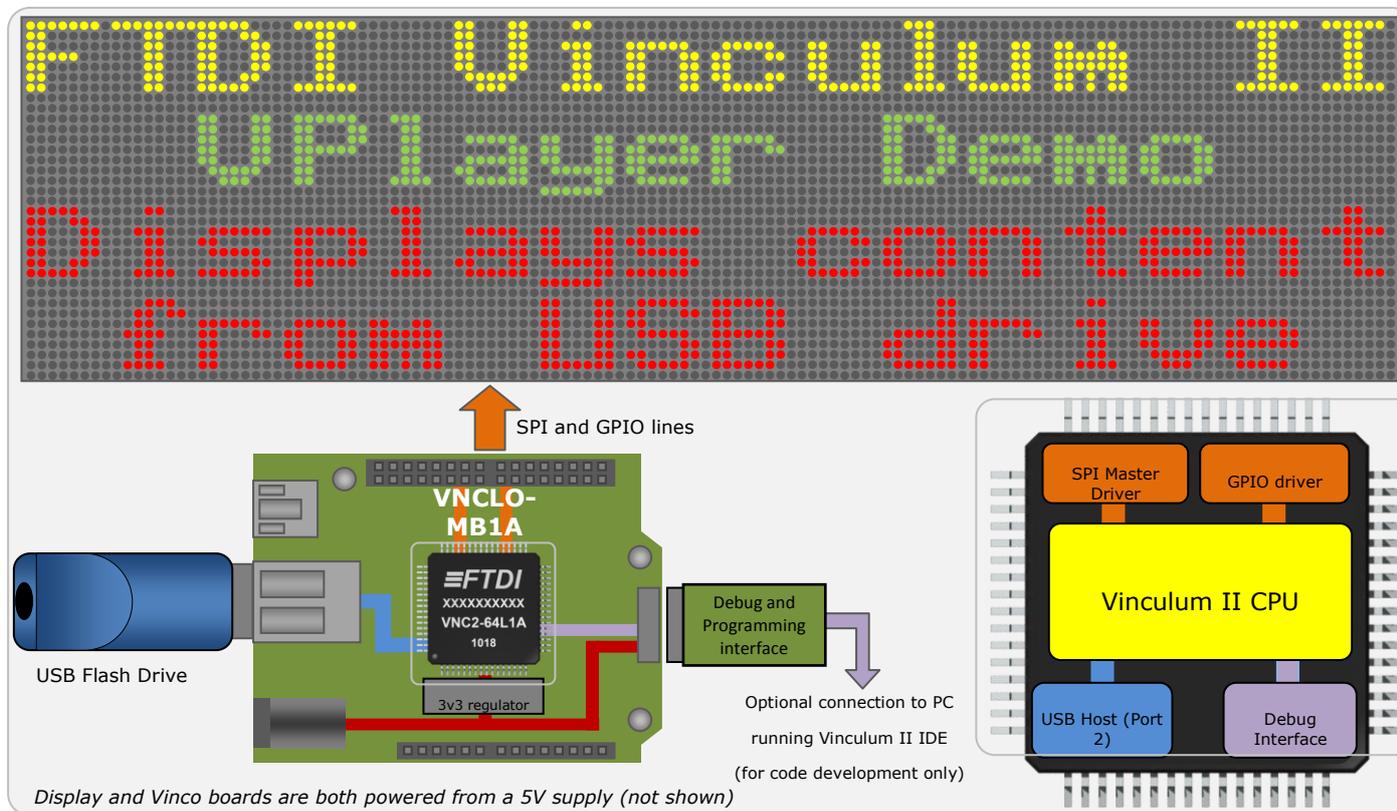


Figure 1 : Vinco Development Module with LED dot matrix display

1.1 Overview

The VPlayer consists of a PC application used to generate the bitmap file, Vinculum-II firmware and hardware interface to read data from the USB Flash drive and display on the LED dot matrix display panel.

The firmware is written in C and developed using the Vinculum-II IDE.

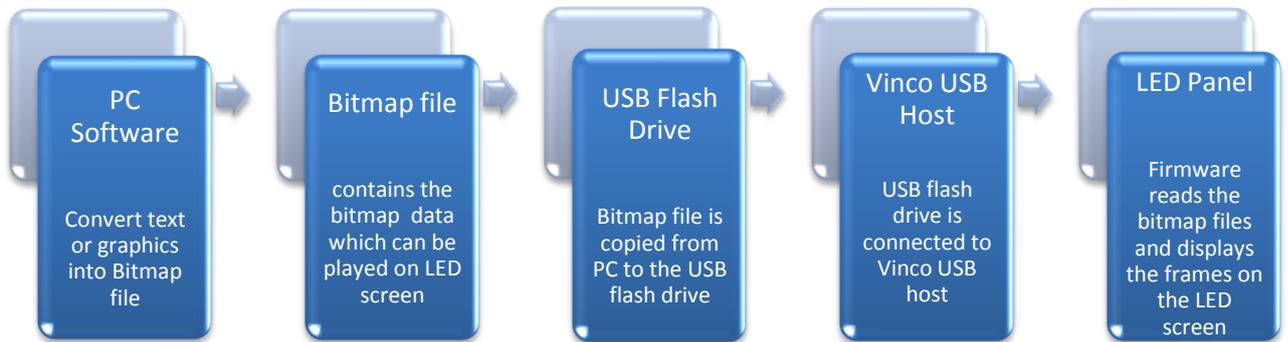


Figure 2 : VPlayer Overview

The VPlayer is a project designed to demonstrate the capabilities of FTDI's Vinculum-II (VNC2) series of embedded USB host controllers. The PC application converts text messages, applying optional effects to the text and saves it as a bitmap file on a USB Flash drive. The firmware reads the bitmap data from the USB Flash drive and sends it to the dot matrix LED panel for display. This can be used for advertising or displaying applications that requires various displaying effects.

1.2 Hardware Requirements

- [Vinco Development Module](#)
- [VNC2 Debug Module](#)
- LED panel, 32x128 pixels (8 LED modules)
- AC:DC converter & regulator (supplies 5V to LED panel)
- Windows PC with Vinculum-II Tool chain installed
- a USB 2.0 compliant A-mini-B cable
- a FAT-formatted USB Flash drive

(See Appendix A for references to the datasheets and downloadable firmware and Appendix B for LED module schematics)

2 Operation

The VPlayer has a PC application (VPlayer Display software) which is used to generate the bitmap files. The display text is converted to a bitmap file in the VPlayer Display software. The bitmap files are copied to a USB flash drive. A master file (SEQUENCE.TXT) is created in the USB flash drive. The master file is a text file which has the sequence of the bitmap file names to be displayed. The master file is edited to include all the bitmap files to be displayed in the sequence. After the files are updated in the USB flash drive, the flash drive is removed from the PC and connected to the Vinco USB host port. Vinculum-II firmware and hardware interface read data from the USB Flash drive and display on the LED dot matrix display panel.

2.1 Using VPlayer Display Software

The following steps are required to create new bitmap files using VPlayer Display Software

1. Connect the USB Flash drive to the PC
2. Open the VPlayer Display Software
3. Enter the text in the text box and select the text colour.
4. Select the display type and click on the "Save Data File" button.
5. Save the bitmap data file to the USB Flash drive. The filename should be in 8.3 format

2.2 Updating the SEQUENCE.TXT file

The following steps are required to update the SEQUENCE.TXT file

1. Open the SEQUENCE.TXT file in the USB Flash drive using a text editor on PC (e.g. wordpad, notepad).
2. Add the bitmap filename to the SEQUENCE.TXT.
3. If the bitmap filename is less than 8 characters then the filename in SEQUENCE.TXT has to be padded with blank spaces. There should be only 1 bitmap filename in each line.
4. The content of the SEQUENCE.TXT file is provided as an example.

```
TCHISTATDAT<NEW LINE>
TCHIUP<SPACE><SPACE>DAT<NEW LINE>
TCHIBLNKDAT<NEW LINE>
SCHISTATDAT<NEW LINE>
SCHIUP<SPACE><SPACE>DAT<NEW LINE>
SCHIBLNKDAT<NEW LINE>
ENGSTAT<SPACE>DAT<NEW LINE>
ENGSTAT<SPACE>DAT<NEW LINE>
ENGUP<SPACE><SPACE><SPACE>DAT<NEW LINE>
ENGLLEFT<SPACE>DAT<NEW LINE>
ENGLLEFT<SPACE>DAT<NEW LINE>
ENGLBLNK<SPACE>DAT<NEW LINE>
```

Note: The Period "." Is not included in the string to separate the name from .DAT.

2.3 VPlayer application demonstration

The following steps are required for operation of VPlayer application demonstration

1. Apply power to the Vinco Development Module and the LED panel
2. Connect the USB Flash drive to USB Host port of the Vinco Development Module
3. The VNC2 will enumerate the USB Flash drive connected to the Vinco Development Module
4. The LED panel will display the text stored in the bitmap file in the USB Flash drive

3 Hardware

3.1 Chipset

The Vinculum-II VNC2 series of embedded USB Host / Slave controllers provide USB Host capability for a variety of USB device classes including support for BOMS (bulk only mass storage), Printer, HID (human interface devices). The VNC2 provides SPI, parallel FIFO and UART interfaces to be active on the available pins through the use of an I/O Multiplexer. The IO Multiplexer also configures the VNC2 IO pins as outputs which will be connected to the Dot Matrix 3mm/5mm bi-colour LED Display panel. The GPIO and SPI Master drivers of Vinculum-II are used to control the LED dot matrix display panel. The USB host port is used to attach a USB Flash drive. The figure below shows Vinco Development Module and VNC2 Debug Module.



Figure 3 : Vinco Development Module with VNC2 Debug Module

3.2 Interfaces

- SPI Master and GPIO – Communications between the Vinculum-II and LED display panel
- USB Host – Reads display data from USB Flash drive
- Debugger Interface – Program and debug the Vinculum-II with VPlayer firmware.

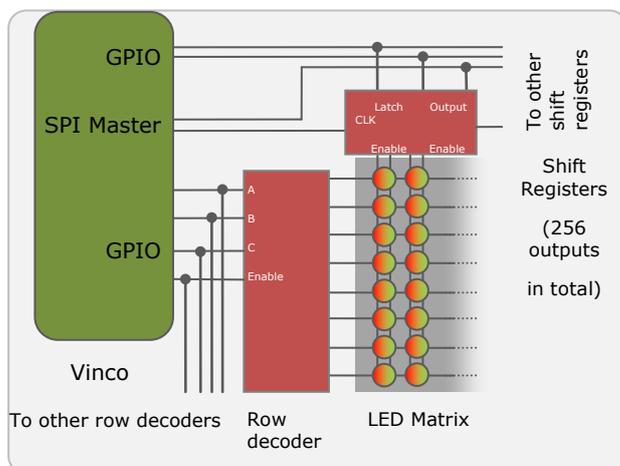


Figure 4 : LED Display driver and multiplexing

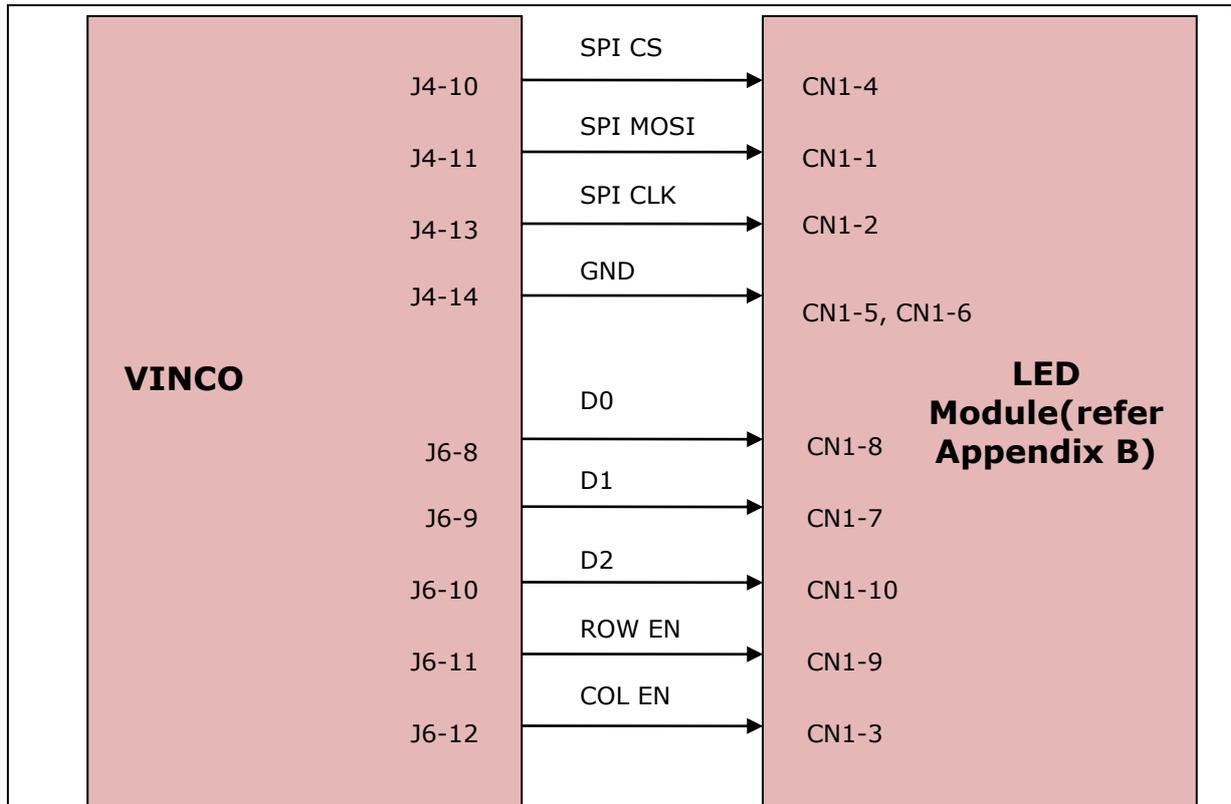


Figure 5 : Interfacing Vinco with LED dot matrix panel

4 Firmware

The firmware corresponds to the VNC2 firmware model defined in [4]. It consists of two threads: **Flash Disk** is responsible for enumerating the USB Flash drive attached to the USB host and reading bitmap data files; **displayOut** is responsible for sending the bitmap data to the LED display panel. Dual buffers are used for storing data and access to the buffer is synchronised using mutexes. The LED display works on the persistence of vision, the data has to be continuously refreshed to the different rows. This is done using the dual buffer, when the Flash disk thread is reading data into buffer1 the displayOut thread writes data to the LED panel from buffer2 and vice versa.

The VPlayer firmware data flow is shown in Figure 6.

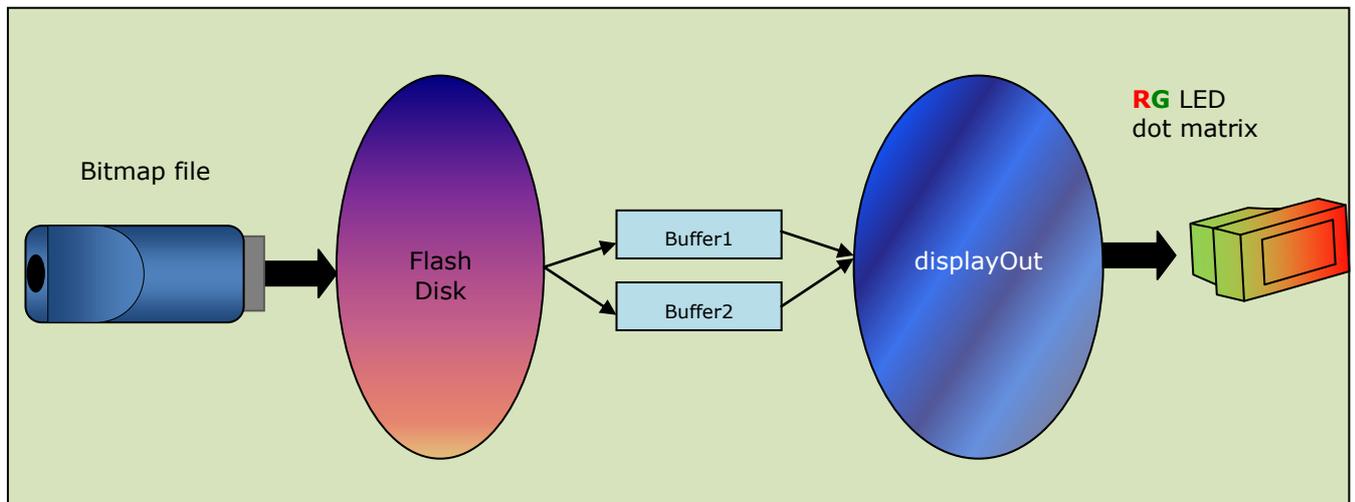


Figure 6 : VPlayer Firmware data flow

The firmware is developed using the Vinculum-II IDE software. It is available free from the FTDI website and has drivers, libraries and examples to help with application development. The SPI Master and GPIO drivers are used to interface the Vinculum-II to the LED display panel.

4.1 File Types

There are 2 types of files to update the display content. The sequence of the display content can be easily manageable without recreating each of the content.

Master file:

All the bitmap files are controlled by a master file (i.e. SEQUENCE.TXT) which has the information on the sequence of displaying the different bitmap files.

Bitmap file:

The display data is stored in a bitmap file which has all the information. The bitmap file has the information for switching ON and OFF all the LEDs in the LED panel in a single frame data. The bitmap file size depends on the number of frame data to be displayed. 24 frames are displayed every second.

5 Source Code for the VNC2 Application Firmware

This section is aimed at those wanting to learn about coding VNC2 devices or modifying the existing project.

The firmware basic steps are:

1. Initialise device drivers
2. Define pin-outs
3. Open ports to be used
4. Configure ports to be used
5. Read/write data
6. Close ports

The VNC2 source code for this project can be viewed in appendix A and can also be downloaded free from the FTDI website.

5.1 vPlayerFlashDisk.C

vPlayerFlashDisk.C is the main firmware file. This file is split into multiple functions.

```
void SPI_Master_Init(void);  
void main(void);  
void flashDisk();  
void displayOut(void);
```

5.1.1 SPI_Master_Init

Configure the SPI master interface. The SPI interface is used to send the data to the LED column drivers. The SPI clock is configured at 3MHz. The SPI clock frequency can be increased to 6 MHz, above which the clock signal is distorted in the cables connecting the LED modules.

5.1.2 main

Configure the IO pins. Initialize the GPIO, SPI, USB host and BOMS class driver. Threads Flash Disk and displayOut are created. Thread Flash Disk is linked to the function flashDisk and thread displayOut is linked to the function displayOut. The VPlayer sample requires the following FTDI provided components: USBHOST driver, BOMS driver, FAT driver, SPI Master driver, Timers driver, GPIO driver and the VOS kernel.

The GPIO pins are used to control the row selection, row enable and the column enable of the LED drivers. To select a particular row D0, D1, D2 has to be written with the row value and the ROWEN is driven LOW.

5.1.3 Thread Flash Disk

- This thread enumerates the USB Flash drive, attached to the USB host port and loads the BOMS class driver and the FAT file system driver.
- It then checks for the master file SEQUENCE.TXT. This file has the sequence of the bitmap files to display.
- The bitmap files is opened and the bitmap data is read into buffer1 and buffer2.
- Wait for data in buffer1 to be completely sent to the hardware. Then read the next block of data into buffer1
- Wait for data in buffer2 to be completely sent to the hardware. Then read the next block of data into buffer2
- Close the bitmap file after reaching End of File.
- Read the next bitmap file in the SEQUENCE.TXT until all the files are read.
- After reaching the end of file in the SEQUENCE.TXT the control transfers to the beginning of the SEQUENCE.TXT

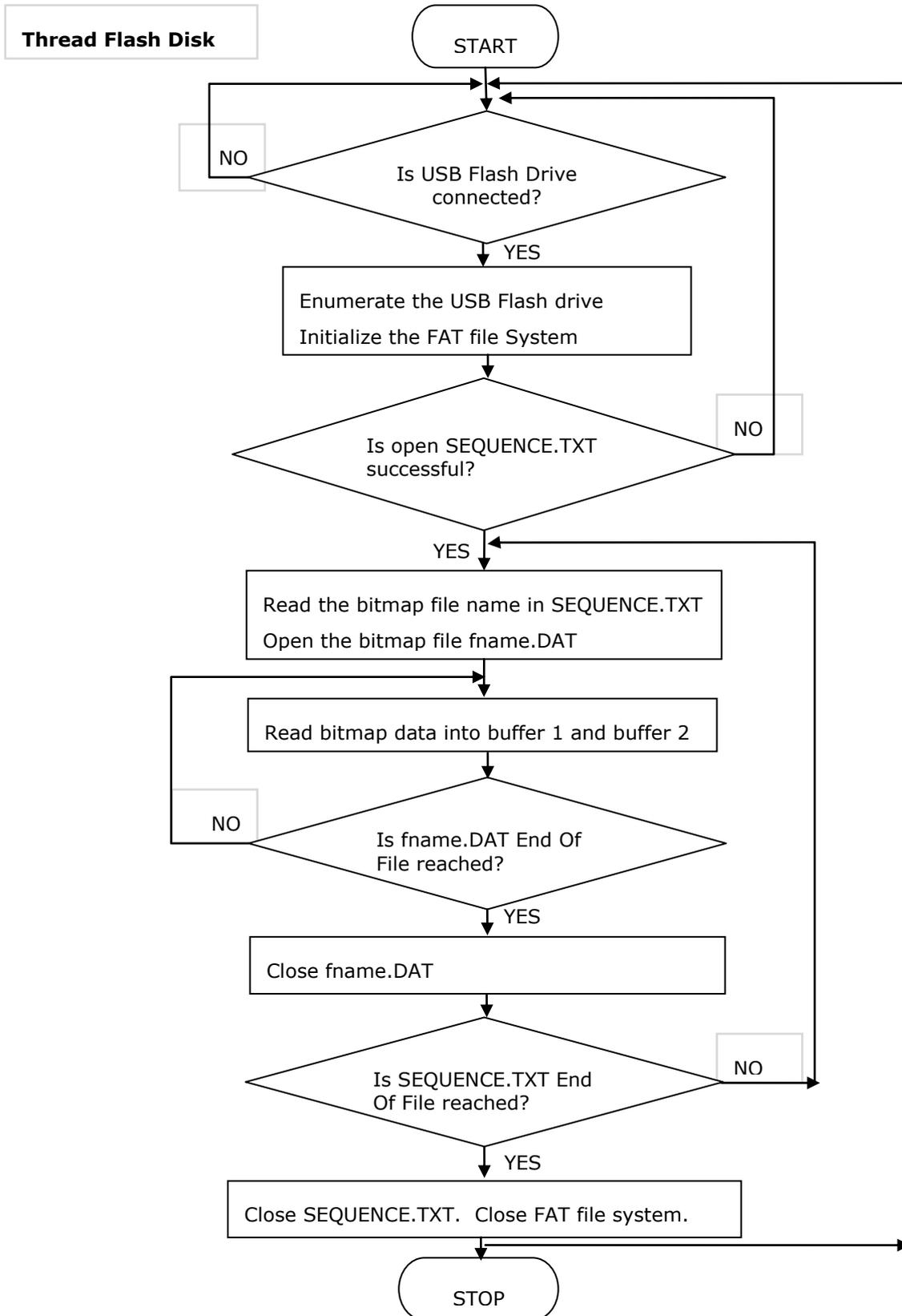


Figure 7 : Thread Flash Disk Flow Chart

5.1.4 Thread displayOut

- This thread initializes the SPI master driver and timer driver.
- It waits for the Flash Disk thread to finish reading data to buffer1.
- It reads the data from the buffer1 and sends it to the dot matrix display panel through the SPI Master write function.
- It then turns the corresponding row driver, row enable and column enable using the GPIO driver.
- It waits for the Flash Disk thread to finish reading data to buffer2.
- It reads the data from the buffer2 and sends it to the dot matrix display panel through the SPI Master write function.
- It then turns the corresponding row driver, row enable and column enable using the GPIO driver.

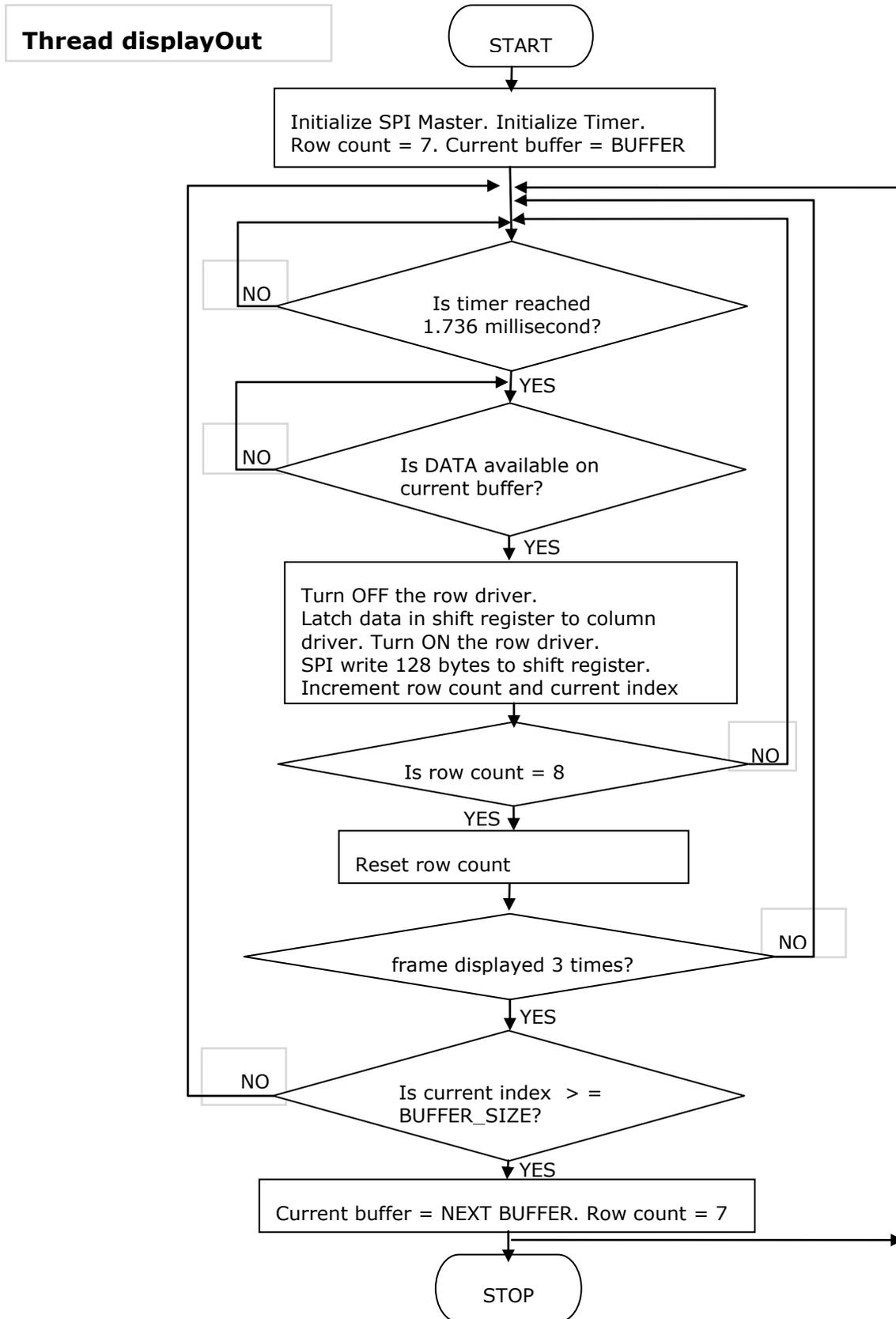


Figure 8 : Thread displayOut Flow Chart

6 Byte ordering in the frame buffer

The VPlayer support displays with 2-bit colour. For all display buffer formats, the order of pixels on the display is from left to right and from top to bottom. For example, pixel (0, 0) appears at the upper left corner of the display, and pixel (width - 1, height - 1) appears at the lower right corner.

6.1 2 bits per pixel colour palette

The 2-bpp format is typically used for 4-entry palette colour displays. The following table shows the bits for the associated palette entry.

Pixel (m,n), Green	Pixel (m,n), Red	Colour palette
0	0	Black
1	0	Green
0	1	Red
1	1	Yellow

Table 1 : Pixel colour palette

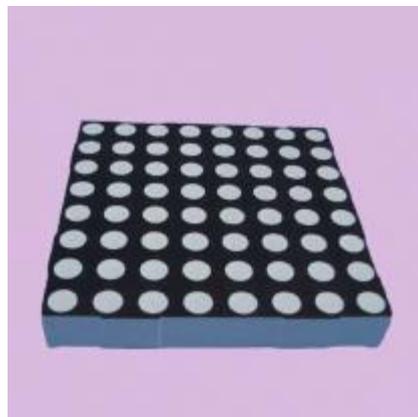


Figure 9 : LED block (8x8 pixels)

A LED block has 8x8 pixels with 2 LED (RED and GREEN) in each pixel position. The number of LEDs in a LED block is 128 (8*8*2).

6.2 Display buffer format for 2 blocks

The following table shows the **byte** arrangement of memory for each row in 2 blocks.

Byte 0			Byte 1			Byte 2			Byte 3		
Bit 7		Bit 0									
P00,	----->	P07,	P08,	----->	P0F,	P00,	----->	P07,	P08,	----->	P0F,
Green		Green	Green		Green	Red		Red	Red		Red

Table 2 : Pixel arrangement in the display buffer format for 2 blocks

6.3 Display buffer format for a LED module

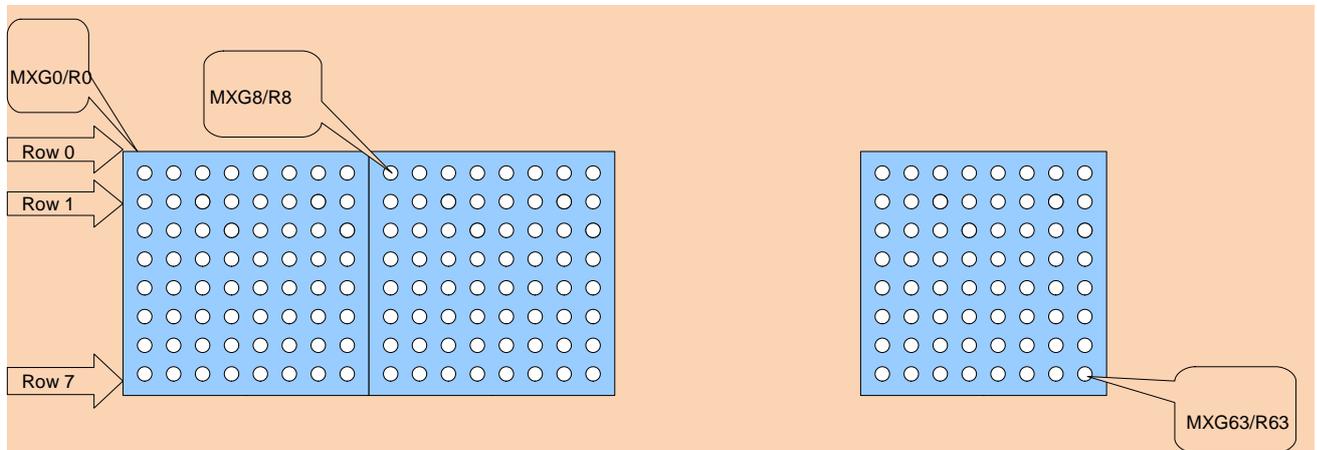
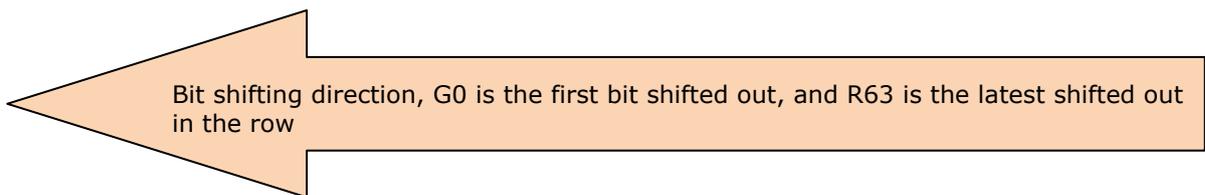


Figure 10 : LED module Pixel numbering



M0	M0	M0	M0	M0	M0	M0	M0
G0-15	R0-15	G16-31	R16-31	G32-47	R32-47	G48-63	R48-63
Green LED	Red LED						

M0 - module 0 (refers to the LED module 0)

M0R0 - module 0 left most pixel red LED

M0G0 - module 0 left most pixel green LED

M0G63 - module 0 right most pixel green LED

M0R63- module 0 right most pixel red LED.

MXG0 - first bit to be shifted out in a row,
MXR63 - last bit to be shifted out in a row,
MX - LED module Number
RY - Red LED column number
GY - Green LED column number

Figure 11 : LED module , data shift direction

6.4 Display buffer format for a LED panel

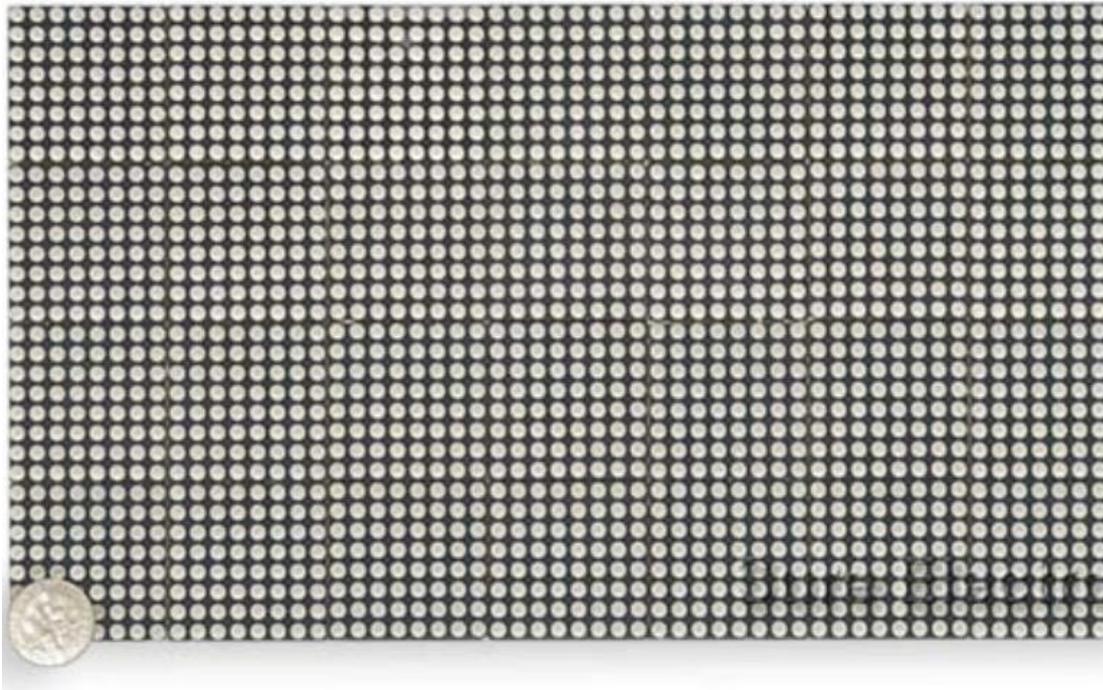


Figure 12 : LED Panel (32 x 64 pixels)

The size of the LED display panel of 32x128 pixels is 59.5cm x 16 cm x 7cm

6.4.1 LED panel stacking

LED block: consists of 8*8 dot pixel, each pixel has 1 red LED and 1 green LED

LED module: consists of 8 LED blocks in a row. It has row and column control signals designed on control PCB.

LED panel: built with 8 LED modules in 4 * 2 arrays. Effectively, the LED panel consists of 32 (row) * 128 (column) dual LED (1 red and 1 green LED in each pixel).

6.4.2 LED column bit shifting direction

In the LED panel, first row of all LED modules will be turned on at same time. The controller will shift the data of row 0 of all LED modules before turn on the corresponding row output.

M0R0/G0 corresponding to top most left pixel on LED module 0, M7G63/R63 corresponding to the right most bottom pixel. Module position can be referred to the diagram below.

The shifting operation will shift all LED column bits of the same row of all LED modules from module M7 to M0. Module 0 green LED 0 (M0G0) is the first bit to shifted out. Detailed shifting sequence is in the frame bit map table below.

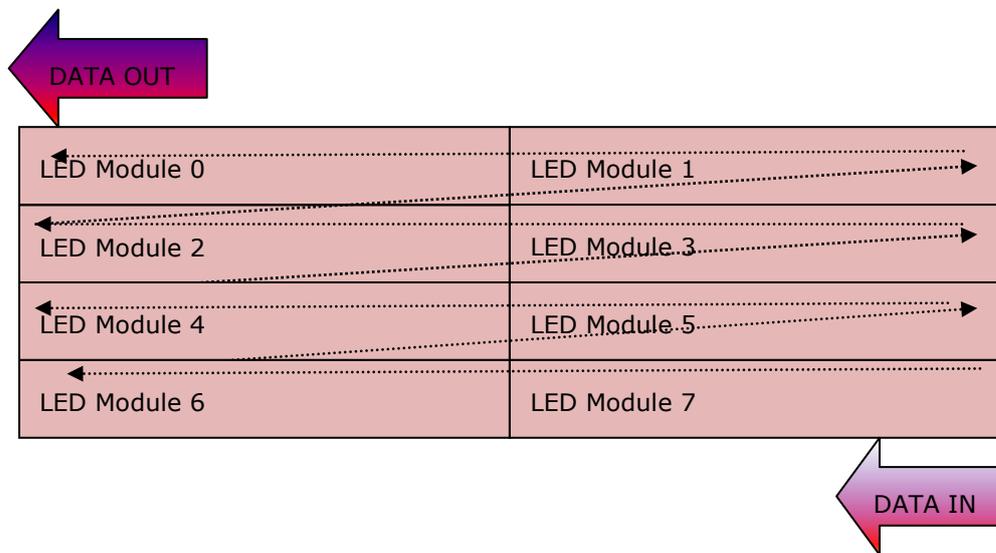


Figure 13 : LED panel stacking

6.4.3 rowData Array

Byte 0 (bit 7 corresponding to M0G0 – the first bit to be shifted out)	M0G[0-7] (M0G0 at bit 7, M0G1 at bit 6..)
Byte 1	M0G[8-15] (M0G8 at bit 7, M0G9 at bit 6..)
Byte 2	M0R[0-7]
Byte 3	M0R[8-15]
Byte 4	M0G[16-23]
Byte 5	M0G[24-31]
Byte 6	M0R[16-23]
Byte 7	M0R[24-31]
Byte 8	M0G[32-39]
Byte 9	M0G[40-47]

Byte 10	M0R[32-39]
Byte 11	M0R[40-47]
Byte 12	M0G[48-55]
Byte 13	M0G[56-63]
Byte 14	M0R[48-55]
Byte 15	M0R[56-63]
Byte 16 (starts module 1 bit map)	
Byte 17 – byte 111	LED bit map for module 1 to 6
.....	
Byte 112	M7G[0-7]
Byte 113	M7G[8-15]
Byte 114	M7R[0-7]
Byte 115	M7R[8-15]
Byte 116	M7G[16-23]
Byte 117	M7G[24-31]
Byte 118	M7R[16-23]
Byte 119	M7R[24-31]
Byte 120	M7G[32-39]
Byte 121	M7G[40-47]
Byte 122	M7R[32-39]
Byte 123	M7R[40-47]
Byte 124	M7G[48-55]
Byte 125	M7G[56-63]
Byte 126	M7R [48-55] (M7R48 at bit 7, M7R47 at bit 6..)
Byte 127	M7R [56-63] (M7R56 at bit 7, M7R57 at bit 6..)

Table 3 : Byte arrangement in rowData Array

The table above shows the data ordering in rowData array.

6.4.4 Bitmap file format

The LED panel bitmap file format based on 8 LED modules per panel, stacking in 4 by 2. The frame buffer contains the data for Row 0 to Row 7. The frame buffer has 128*8 bytes.

Structure BitmapFileFormat

```
{
    Structure framebuffer Data_Frame0
    {
        Structure rowBuffer Data_Row 0
        {
            rowData[128];// include the array in section 6.4.3 here
        }
        Structure rowBuffer Data_Row 1
        {
            rowData[128];// include the array in section 6.4.3 here
        }
        Structure rowBuffer Data_Row 2
        {
            rowData[128];// include the array in section 6.4.3 here
        }
        Structure rowBuffer Data_Row 3
        {
            rowData[128];// include the array in section 6.4.3 here
        }
        Structure rowBuffer Data_Row 4
        {
            rowData[128];// include the array in section 6.4.3 here
        }
        Structure rowBuffer Data_Row 5
        {
            rowData[128];// include the array in section 6.4.3 here
        }
        Structure rowBuffer Data_Row 6
        {
            rowData[128];// include the array in section 6.4.3 here
        }
        Structure rowBuffer Data_Row 7
```

```
        {
            rowData[128]; // include the array in section 6.4.3 here
        }
    } /* end of Structure framebuffer Data_Frame0 */

    Structure framebuffer Data_Frame1
    {
        ...
    }
    Structure framebuffer Data_Frame2
    {
        ...
    }
    .
    .
    .

    Structure framebuffer Data_FrameN
    {
        ...
    }

} /* end of Structure BitmapFileFormat */
```

7 VPlayer Display Software

Bitmap creation is done by VPlayer Display software in the computer.

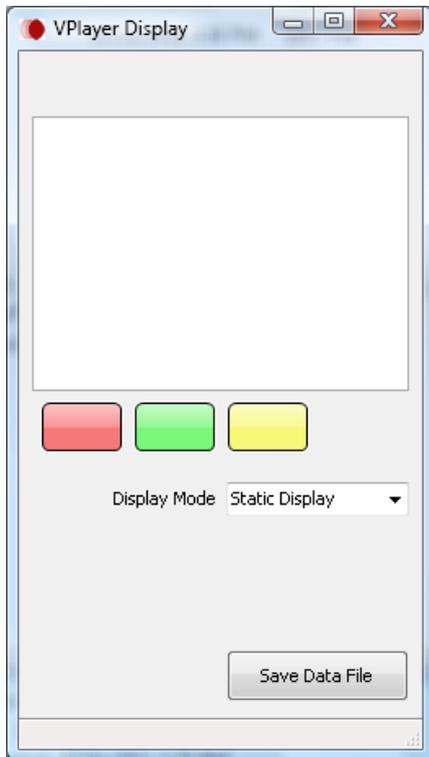


Figure 14 : VPlayer Display Software

FTDI provide an example PC utility called VPlayer Display Software, which is used to create the messages for the LED display panel. The text to be displayed is entered into the VPlayer text box. Individual characters can be coloured Red, Green or Yellow and the display mode (static, shift left, shift up or blink) can also be selected.

The Save Data File button creates the bitmap data file for this particular message. This processed bitmap contains all of the individual frames required to display the entire message with the selected effect. The bitmap file is then copied to the USB Flash drive. The master file (SEQUENCE.TXT) on the flash drive should also be updated to add the name of the bitmap file. This file allows multiple messages (each with their own bitmap file) to be shown in sequence and defines the order in which they should be displayed.

The SEQUENCE.TXT file has to be manually edited using a text editor. Please refer to the section "Updating the SEQUENCE.TXT file".

The USB Flash drive is then inserted into the FTDI Vinco USB Host port (port 2). The firmware reads the SEQUENCE.TXT file to determine the name of the first bitmap to display. It reads though the bitmap, taking each frame in turn, and sending this to the display at a rate of 24 frames per second. It then moves onto the next bitmap file in the SEQUENCE.TXT file. Once the end of the sequence is reached, it will start at the beginning again.

7.1 Features supported

- **Text Edit box**
The user enters the text in the text box.
- **Colour Section buttons**
Select the text and click on the colour buttons to choose the text colour.
- **Display Mode**
Select the different display modes from the list.
- **Save Data file**
Click on the "Save Data File" button will open a File Save dialog box. Choose the folder path, enter the filename and click on the save button. The bitmap data is stored to a bitmap file. This bitmap file will be copied to USB Flash drive which will be connected to the Vinco USB host port.

8 Contact Information

Head Office – Glasgow, UK

Future Technology Devices International Limited
Unit 1, 2 Seaward Place, Centurion Business Park
Glasgow G41 1HH
United Kingdom
Tel: +44 (0) 141 429 2777
Fax: +44 (0) 141 429 2758

E-mail (Sales) sales1@ftdichip.com
E-mail (Support) support1@ftdichip.com
E-mail (General Enquiries) admin1@ftdichip.com

Branch Office – Hillsboro, Oregon, USA

Future Technology Devices International Limited
(USA)
7235 NW Evergreen Parkway, Suite 600
Hillsboro, OR 97123-5803
USA
Tel: +1 (503) 547 0988
Fax: +1 (503) 547 0987

E-Mail (Sales) us.sales@ftdichip.com
E-Mail (Support) us.support@ftdichip.com
E-Mail (General Enquiries) us.admin@ftdichip.com

Branch Office – Taipei, Taiwan

Future Technology Devices International Limited
(Taiwan)
2F, No. 516, Sec. 1, NeiHu Road
Taipei 114
Taiwan, R.O.C.
Tel: +886 (0) 2 8791 3570
Fax: +886 (0) 2 8791 3576

E-mail (Sales) tw.sales1@ftdichip.com
E-mail (Support) tw.support1@ftdichip.com
E-mail (General Enquiries) tw.admin1@ftdichip.com

Branch Office – Shanghai, China

Future Technology Devices International Limited
(China)
Room 408, 317 Xianxia Road,
Shanghai, 200051
China
Tel: +86 21 62351596
Fax: +86 21 62351595

E-mail (Sales) cn.sales@ftdichip.com
E-mail (Support) cn.support@ftdichip.com
E-mail (General Enquiries) cn.admin@ftdichip.com

Web Site

<http://ftdichip.com>

System and equipment manufacturers and designers are responsible to ensure that their systems, and any Future Technology Devices International Ltd (FTDI) devices incorporated in their systems, meet all applicable safety, regulatory and system-level performance requirements. All application-related information in this document (including application descriptions, suggested FTDI devices and other materials) is provided for reference only. While FTDI has taken care to assure it is accurate, this information is subject to customer confirmation, and FTDI disclaims all liability for system designs and for any applications assistance provided by FTDI. Use of FTDI devices in life support and/or safety applications is entirely at the user's risk, and the user agrees to defend, indemnify and hold harmless FTDI from any and all damages, claims, suits or expense resulting from such use. This document is subject to change without notice. No freedom to use patents or other intellectual property rights is implied by the publication of this document. Neither the whole nor any part of the information contained in, or the product described in this document, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder. Future Technology Devices International Ltd, Unit 1, 2 Seaward Place, Centurion Business Park, Glasgow G41 1HH, United Kingdom. Scotland Registered Company Number: SC136640

Appendix A – References

Document References

Firmware source code

http://www.ftdichip.com/Support/SoftwareExamples/VinculumIIProjects/AN_200_Source_Code.zip

- [1] FTDI Application Note 142, [Vinculum-II Tool Chain Getting Started Guide](#), FTDI, 2010.
- [2] FTDI Application Note 144, [Vinculum-II IO Mux Config Utility User Guide](#), FTDI, 2010.
- [3] FTDI Application Note 139, [IO MUX Explained](#), FTDI, 2010.
- [4] FTDI Application Note 151, [Vinculum-II User Guide](#), FTDI, 2010.
- [5] FTDI Application Note 159, [Vinculum-II Firmware Flash Programming](#), FTDI, 2011
- [6] FTDI [Vinc Development Module](#)
- [7] FTDI [VNC2 Debug Module](#)

Acronyms and Abbreviations

Terms	Description
GPIO	General Purpose Input / Output pin
IDE	Integrated Development Environment
IOMux	Input Output Multiplexer – Used to configure pin selection on different package types of the VNC2.
LED	Light Emitting Diode
LED block	It consists of 8*8 dot pixel, each pixel has 1 red LED and 1 green LED
LED panel	It is built with 8 LED modules in 4 * 2 array. Effectively, the LED panel consists of 32 (row) * 128 (column) dual LED (1 red and 1 green LED in each pixel)
LED module	It consists of 8 LED block in a row. It has row and column control signals designed on control PCB
SPI	Serial Peripheral Interface Bus
VNC2	Vinculum II
VOS	Vinculum Operating System

9 Appendix B – LED module schematics (control board and dot matrix board)

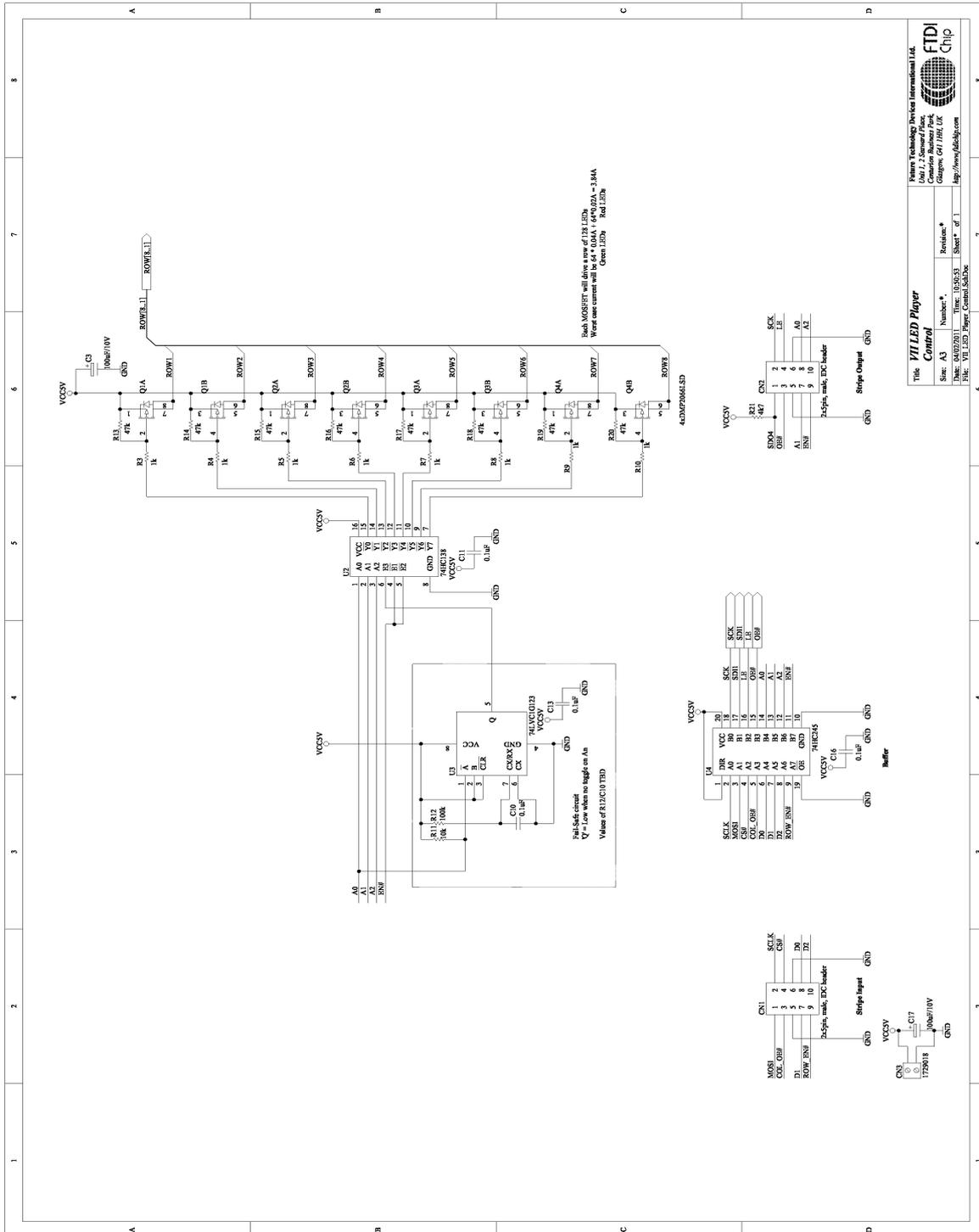
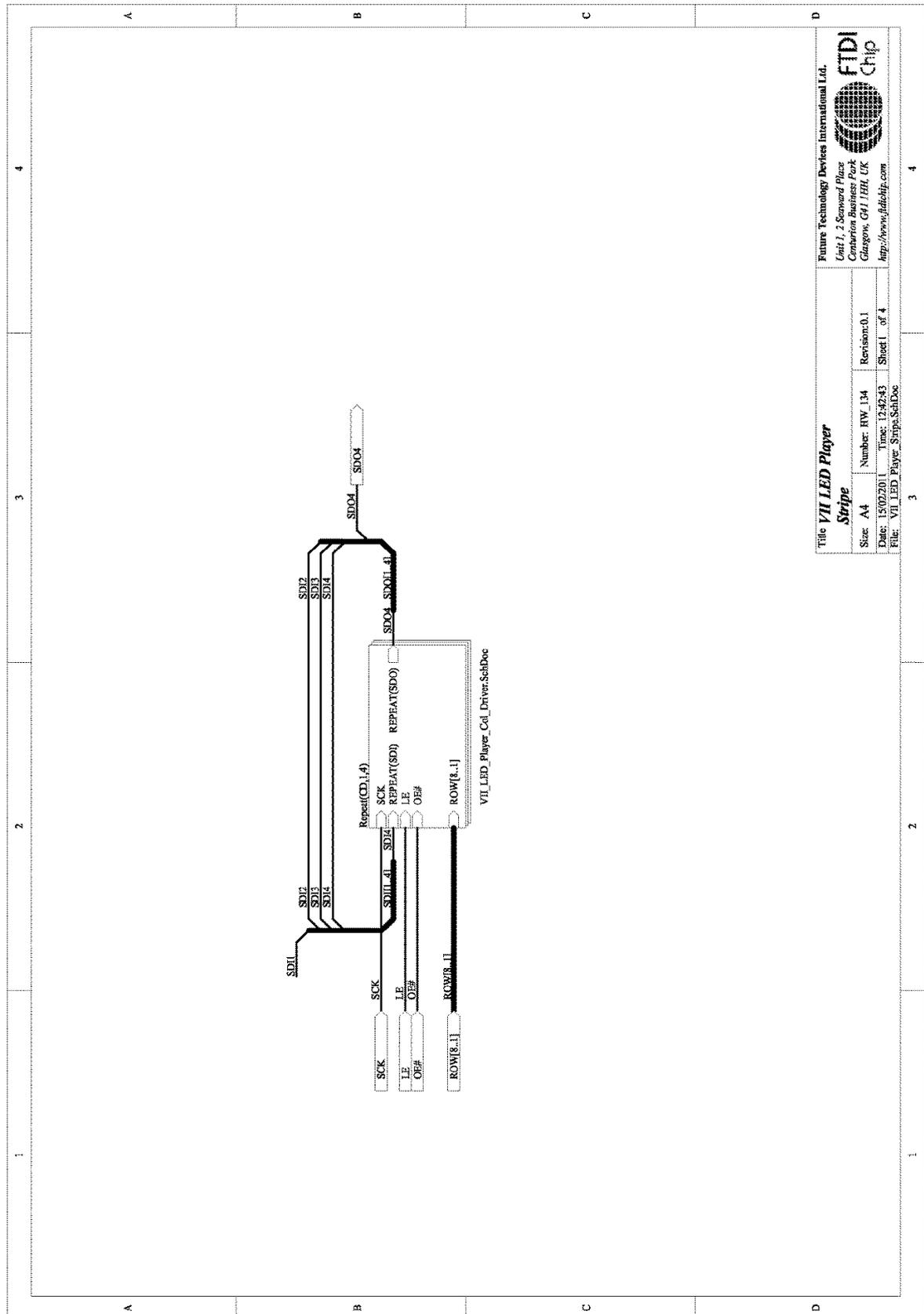


Figure 15 : LED control board with row drivers



Title VII_LED_Player_Strip

Future Technology Devices International Ltd.
 Unit 1, 2 Seaward Place
 Canberron Business Park
 Glasgow, G47 1RH, UK
<http://www.ftdchip.com>

Size: A4	Number: RW_134	Revision: 0.1
Date: 15/02/2011	Time: 12:42:43	Sheet 1 of 4
File: VII_LED_Player_Strip_SetDoc		

Figure 16 : LED column driver

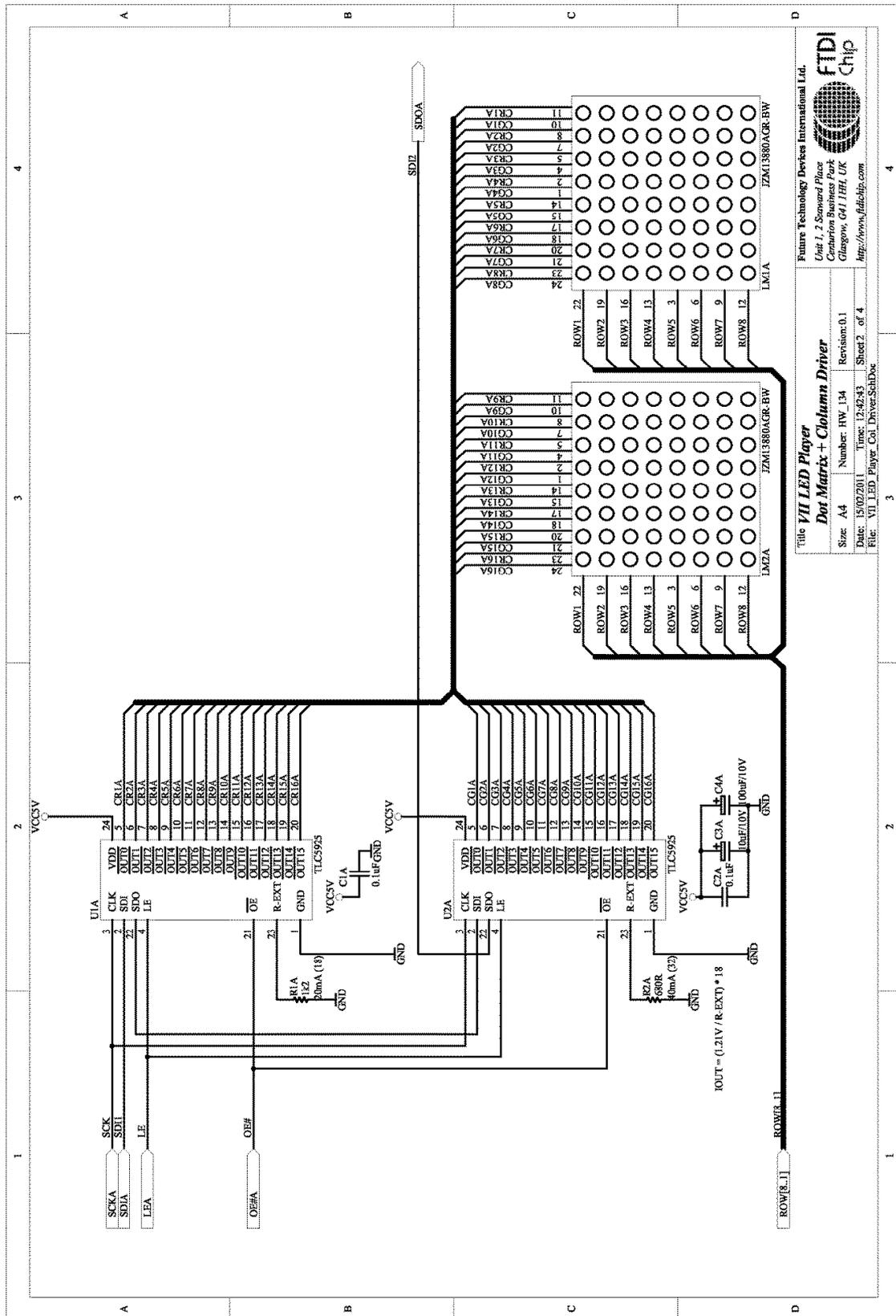
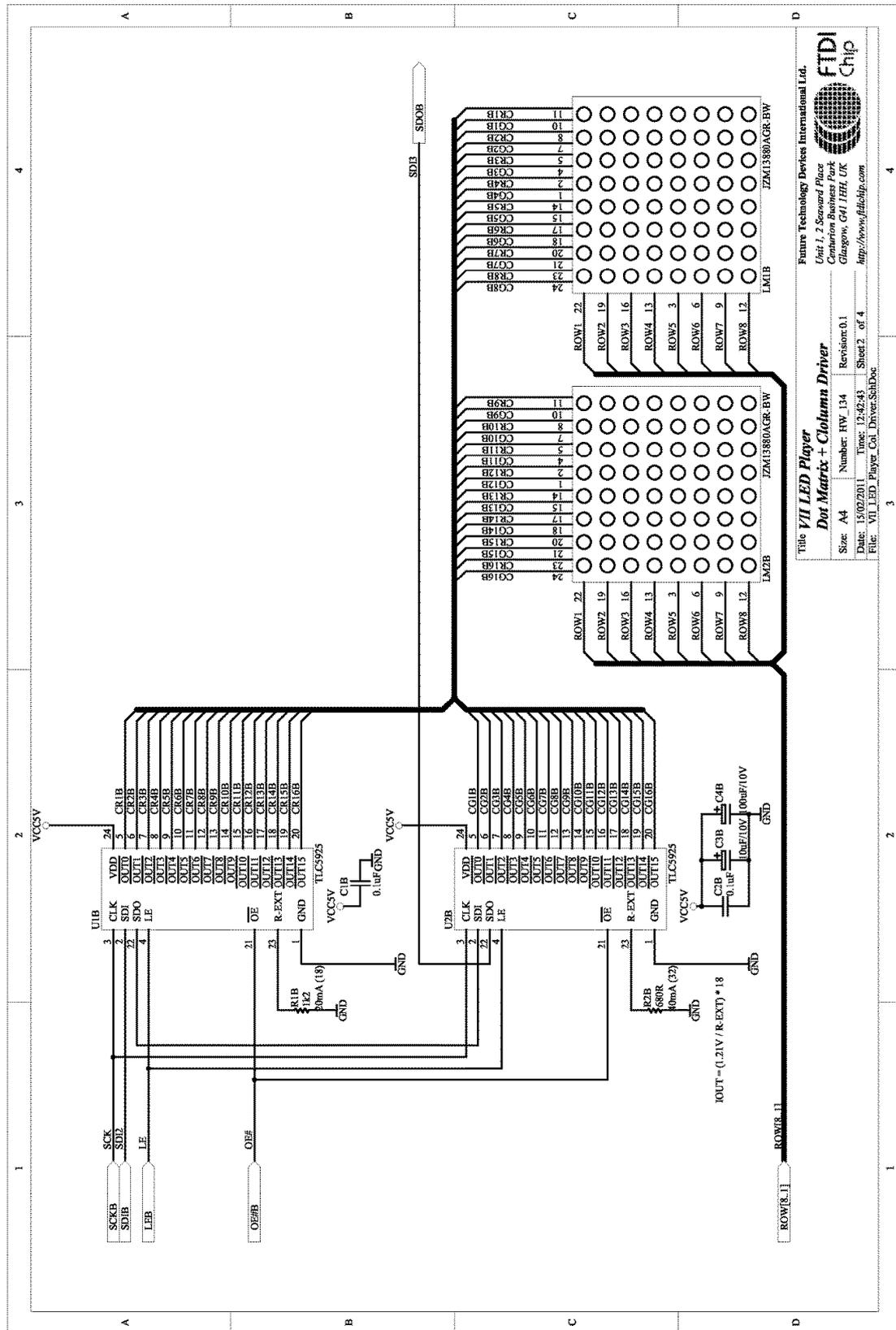


Figure 17 : LED dot matrix with column driver for block 0 and block 1

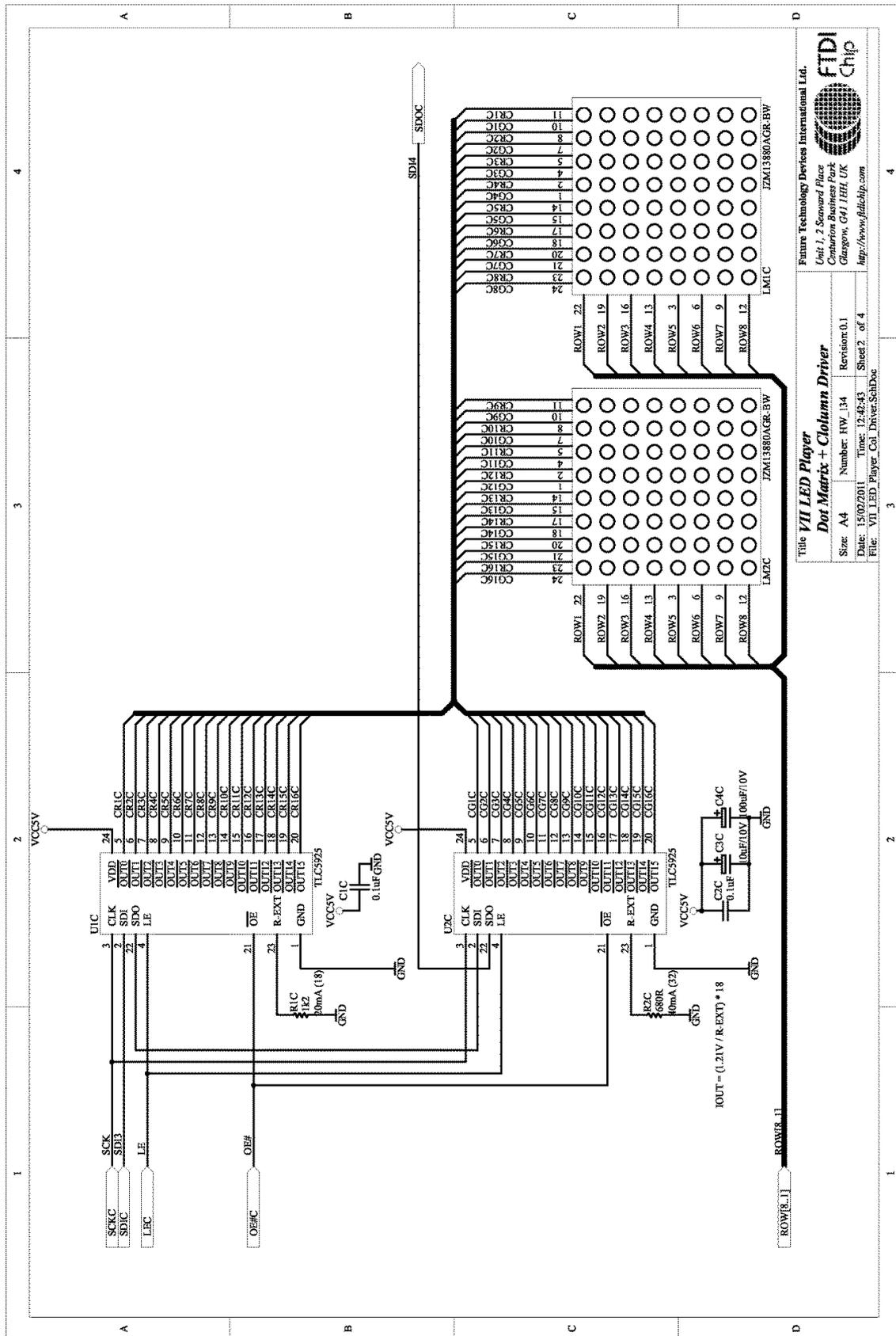


Future Technology Devices International Ltd.
Unit 1, 2 Second Place
Chimney Works, 81
Glasgow, G4 1HH, UK
<http://www.ftdi.com>

**Title VII LED Player
Dot Matrix + Coltium Driver**

Size: A4 Number: FW_134 Revision: 0.1
Date: 15/02/2011 Time: 12:42:43 Sheet 2 of 4
File: VII_LED_Player_Col_Driver_SchDoc

Figure 18 : LED dot matrix with column driver for block 2 and block 3



Future Technology Devices International Ltd.
Unit 1, 2 Seaward Place
Centurion Business Park
Glasgow, G4 1 1HH, UK
<http://www.ftdi.com>

Title: VII LED Player + Coltium Driver

Size: A4	Number: HW_134	Revision: 0.1
Date: 15/07/2011	Time: 12:42:43	Sheet 2 of 4
File: VII LED Player_Col_Driver_SchDoc		

Figure 19 : LED dot matrix with column driver for block 4 and block 5

Appendix C – List of Tables & Figures

List of Tables

Table 1 : Pixel colour palette	15
Table 2 : Pixel arrangement in the display buffer format for 2 blocks	16
Table 3 : Byte arrangement in rowData Array	19

List of Figures

Figure 1 : Vinco Development Module with LED dot matrix display	3
Figure 2 : VPlayer Overview	4
Figure 3 : Vinco Development Module with VNC2 Debug Module	7
Figure 4 : LED Display driver and multiplexing	7
Figure 5 : Interfacing Vinco with LED dot matrix panel	8
Figure 6 : VPlayer Firmware data flow	9
Figure 7 : Thread Flash Disk Flow Chart	12
Figure 8 : Thread displayOut Flow Chart	14
Figure 9 : LED block (8x8 pixels)	15
Figure 10 : LED module Pixel numbering	16
Figure 11 : LED module , data shift direction	17
Figure 12 : LED Panel (32 x 64 pixels)	17
Figure 13 : LED panel stacking	18
Figure 14 : VPlayer Display Software	22
Figure 15 : LED control board with row drivers	26
Figure 16 : LED column driver	27
Figure 17 : LED dot matrix with column driver for block 0 and block 1	28
Figure 18 : LED dot matrix with column driver for block 2 and block 3	29
Figure 19 : LED dot matrix with column driver for block 4 and block 5	30
Figure 20 : LED dot matrix with column driver for block 6 and block 7	31

Appendix D – Revision History

Document Title: AN_200 Vinculum-II Scrolling Text Application Using LED dot matrix display

Document Reference No.: FT_000632

Clearance No.: FTDI# 277

Product Page: <http://www.ftdichip.com/Products/ICs/VNC2.htm>

Document Feedback: [Send Feedback](#)

Revision	Changes	Date
1.0	First Release	2012-02-09