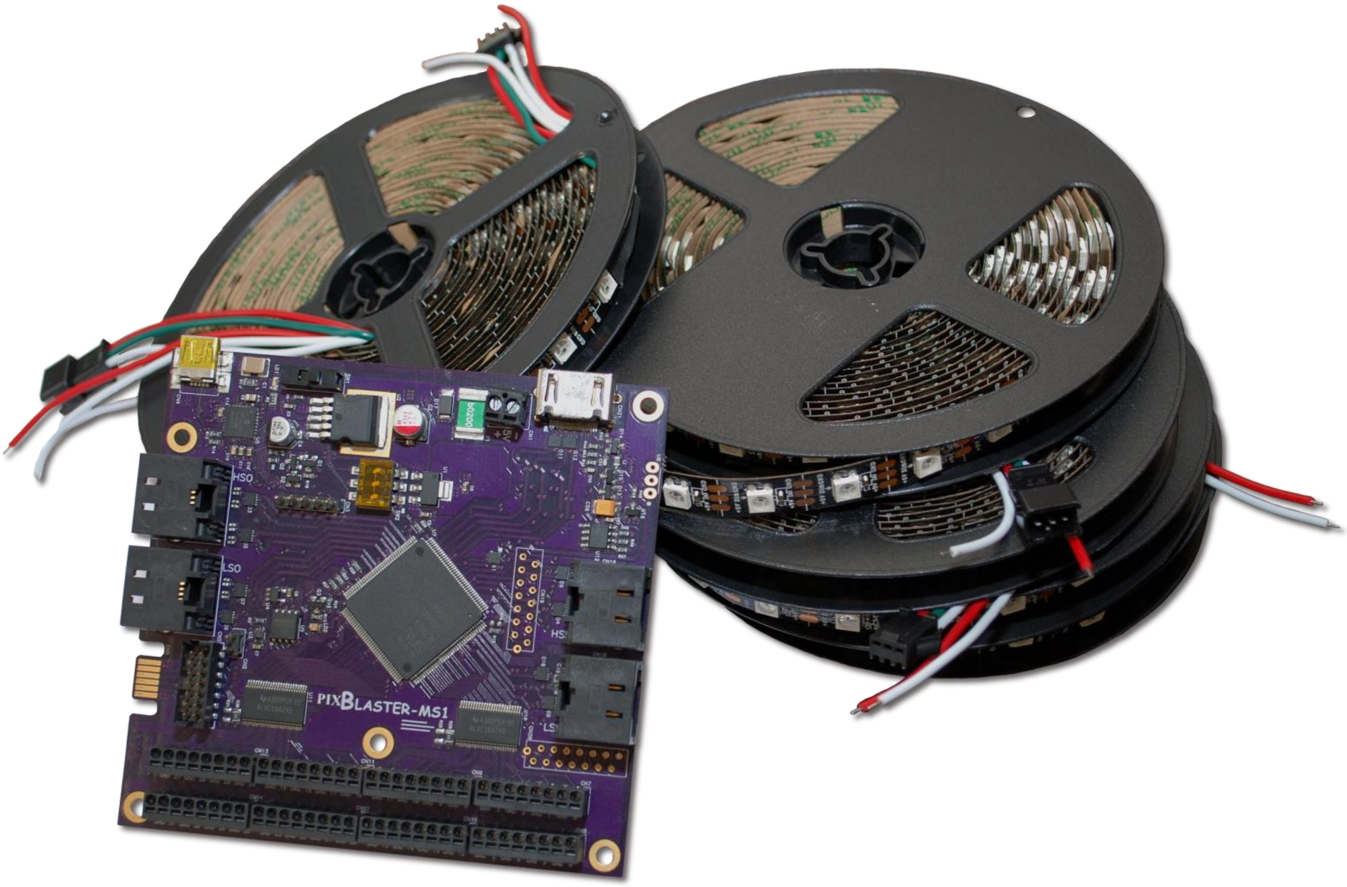
Pixblasters™ MS1 Video LED Controller User's Guide

March 2020



This publication has been carefully checked for accuracy. However, Pixblasters does not assume any responsibility for the contents or use of any product described herein. Pixblasters reserves the right to make any changes to product without further notice. Our customers should ensure to take appropriate action so that their use of our products does not infringe upon any patents.

ℹ *The pink colored PCB showcased on the cover page and several infographics through the document is the Ready-for-Production prototype of the Pixblasters MS1 Video LED Controller. The serial production board have black solder mask -* ***Figure 1.1***

1 INTRODUCTION

The Pixblasters™ MS1 video LED controller connects to any computer and any operating system as an ordinary monitor to display the user-selected portion of the monitor image on up to 16,384 RGB LEDs at 60 frames per second (FPS). This controller enables DIY enthusiasts and signage professionals, even those with minimal technical skills, to turn a bunch of addressable RGB LED strips into immense video LED displays. The Pixblasters MS1 controller enables new levels of professional-grade digital signage that cannot be attained by standard LED modules. The LED strips can be curved and glued to different surfaces to form giant yet economically viable video installations that can span entire buildings.

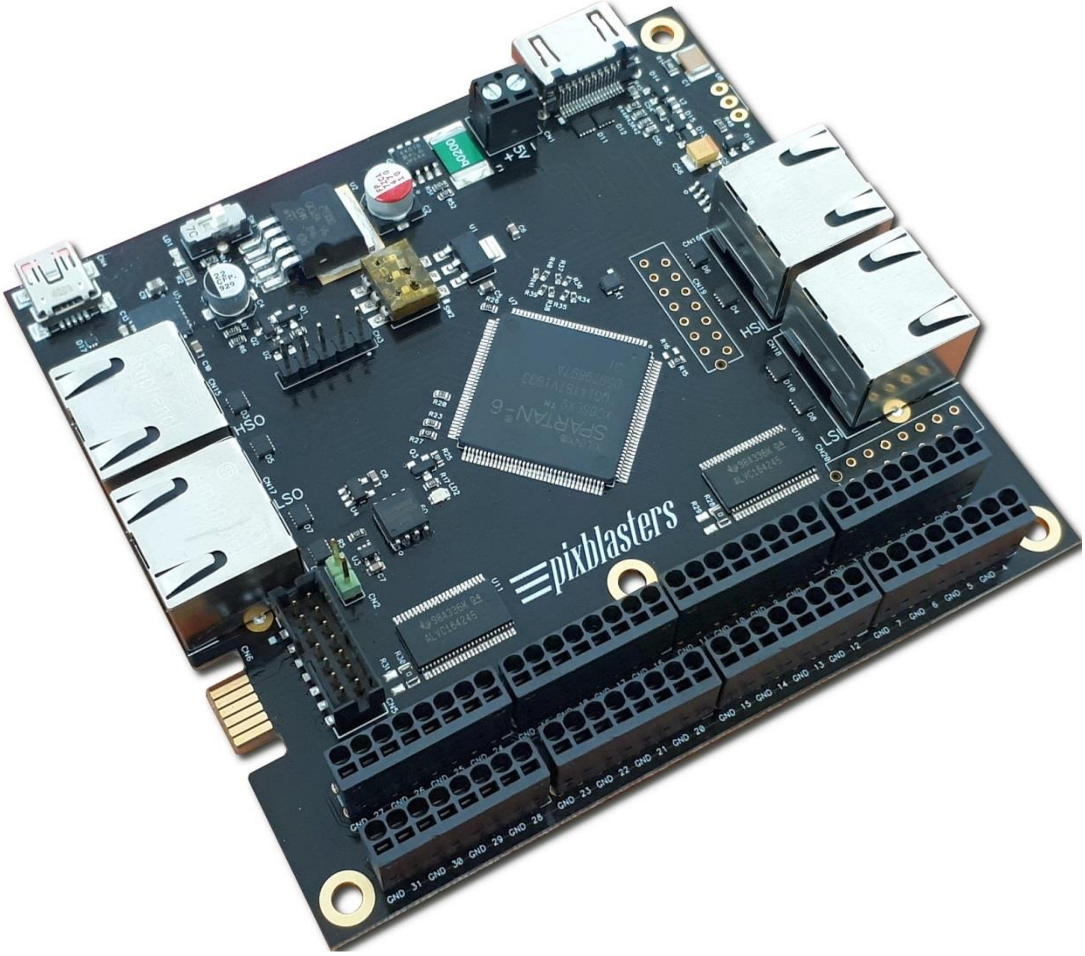


Figure 1.1 The Pixblasters MS1 Video LED Controller

Multiple Pixblasters MS1 controllers can be easily chained to drive immense video displays built of hundreds of thousands of perfectly synchronized LEDs. The Pixblasters MS1 displays any visual content with absolutely no programming required and with no burden on the driving computer that is free to run digital signage players, media players, and other software at the full speed. The LED displays controlled by the Pixblasters MS1 can be remotely managed anywhere in the world by virtually any digital signage software. With the MS1 LED controller, even those with minimal technical skills can build giant LED displays with no soldering and by using only simple tools like pliers, wire cutters and screwdrivers.

1.1 Display Video from Any Computer

The Pixblasters MS1 video LED controller connects to any computer as an ordinary monitor and puts no burden on the driving computer, which is free to run digital signage players, media players, and other video software at full speed. The LED displays controlled by the Pixblasters MS1 can be remotely managed by the digital signage software of your choice from anywhere in the world.

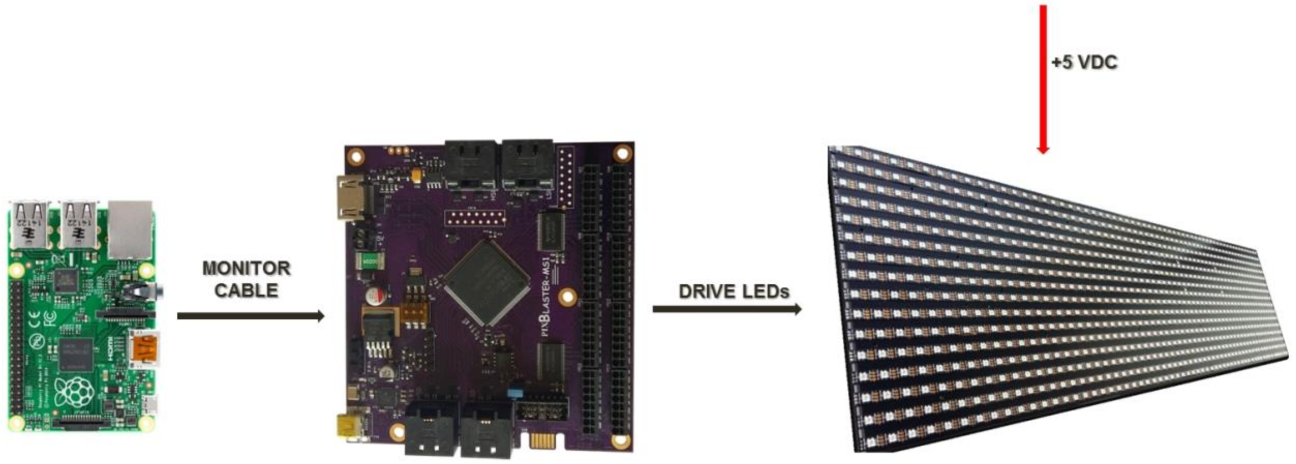


Figure 1.2 Pixblasters Controller Inserts between the Video Source and the LED Matrix

The Pixblasters Video LED Controller has been designed to support professional-looking and easy to use video LED displays. To display any graphics and video with no limitations, Pixblasters users just need to plug in the monitor cable. Stills and animations, AVI, FLV, WMV, MP4, etc. - what you see on the computer monitor connected to the selected computer is what you get on the Pixblasters MS1 controller-driven LED display with absolutely no coding required!

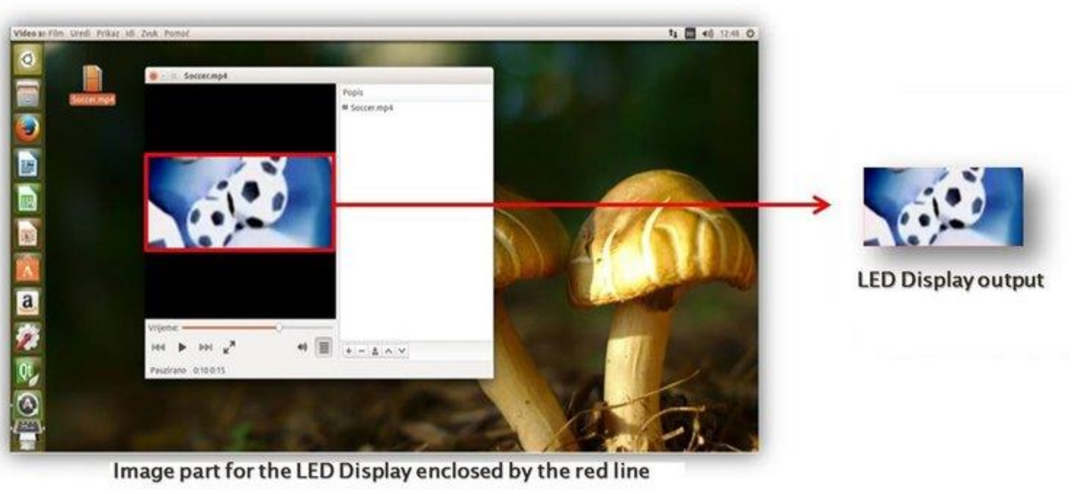


Figure 1.3 Selecting and Cropping of the Image Part for the LED Display

A single Pixblasters Video LED Controller supports up to 16,384 LEDs that can be arranged in LED display matrices with various resolutions. Although respectable, such LED driving capacity allows for the LED display of a smaller portion of standard monitor image. The Pixblasters LED controller receives the video formatted for the monitor, crops the image part selected (see **Video Input**

Cropping) for the LED display (**Figure 1.3**), and drives the LED strips by properly formatted video data. The complexity of LED driving is hidden from the driving computer and the WS2812B LEDs can be refreshed at the maximum 60 Hz refresh rate!

To keep the promise of supporting professional and easy-to-use video LED displays, the Pixblasters LED integrates several important features. For example, it includes an actual EDID chip to smoothly connect to any operating system as a standard monitor. Most consumer monitors these days carry inside an EEPROM chip called the EDID. It contains monitor's name and technical info about its capabilities. This information enable any standard video source, i.e. the PC computer, to properly detect the monitor and automatically format the video image suitable for the specific monitor.

The Pixblasters really shines in big video installations, which may be oversized and hardly portable. Such installations accept only minimal maintenance and must work automatically when powered. Consequently, the LED controller requires retentive memory to store important configuration data, such as the coordinates of the image window for cropping. It also needs a "house-keeping" device to take care on SRAM-based FPGA chip with no internal storage for its configuration data, PC connectivity, multi-controller display configurations, etc. All these functions are integrated in on-board Microchip PIC18F26J50 microcontroller that connects to the PC and runs a simple User's Interface. Besides the display configuration, it also enables firmware updates with new features.

1.2 System Connections

The Pixblasters Video LED Controller is designed to be expandable and to support really big video LED displays. While a single Pixblasters MS1 video LED controller drives up to 16,384 LEDs, multiple daisy-chained Pixblasters MS1 controllers can potentially drive hundreds of thousands of LEDs. To provide the simplest possible video display's architectures, each Pixblasters controller can operate either in Master or Slave modes. The **Figure 1.4** explains the Pixblasters daisy-chain system connections and an example architecture of 18.8 m^2 (202 ft^2) video display built of 60 LEDs/meter addressable RGB LED strips. Paragraph **Multi-controller Demo Display Architecture** showcases the Pixblasters demo display built with multiple LED controllers.

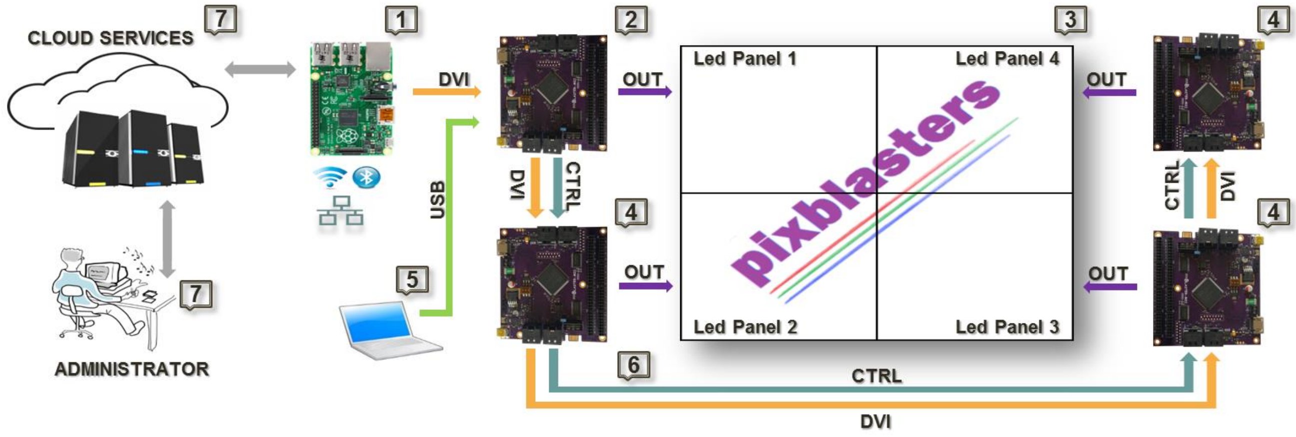


Figure 1.4 Architecture of Video LED Display Driven by Daisy-chained Pixblasters Controllers

- Control computer (1) generates video for the complete LED display, and with the standard monitor cable, connects to the DVI compatible video input of the Master Pixblasters MS1 video LED controller (2).

- Optionally, the computer (1) can be locally or remotely controlled through different network interfaces. It can run any video display application, including digital signage players that automatically connect to online digital content services (7).

- The Master Pixblasters controller (2) controls Slave controllers (4) through the DVI video link and an optional control link*. Each slave controller displays the exact portion of the video image on the attached LED panels (3) built of up to 16,384 RGB LEDs.

- The video and the control links (6) between the chained Pixblasters MS1 controllers use ordinary Ethernet cables to enable simple and flexible display wiring. The links guarantee automatic and perfect LEDs synchronization.

- Each controller (2, 4) is configured for Master/Slave mode by on-board DIP switches. Other configuration parameters are setup only once for each board through a simple configuration menu on the PC (5) connected with the USB cable.

- Display's configuration parameters, such as LED timing parameters and the video input image part for display on the attached LEDs, are permanently stored in the on-board microcontroller that also controls the LED system's boot-up.

- Separate +5V DC power supplies, which are not shown in **Figure 1.4**, must provide sufficient power for the complete display.

** Full control link capabilities are TBD.*

1.3 Flexible Enough for DIYers, Powerful Enough for Professionals

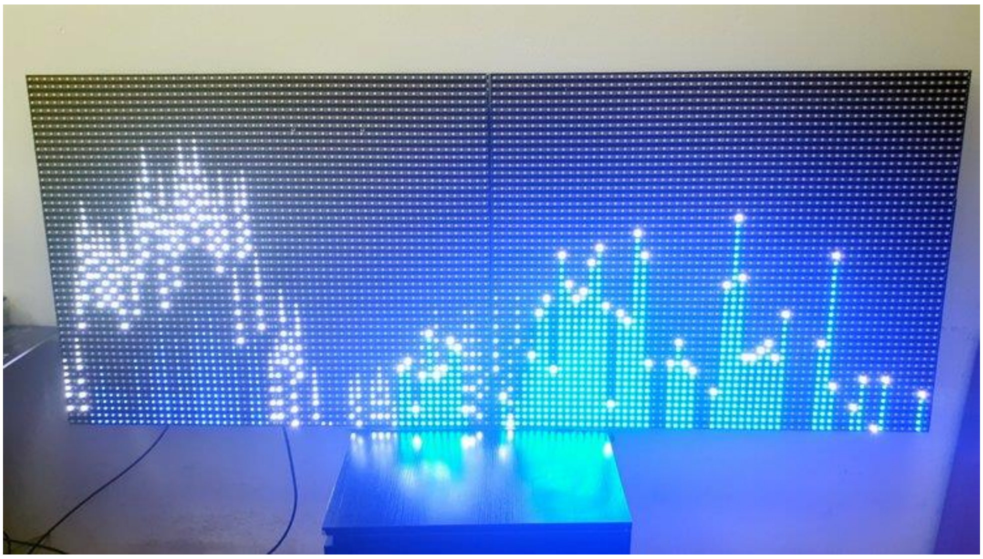


Figure 1.5 Pixblasters Demo Made Using LED Strips Glued to Aluminum Plates

The Pixblasters Video LED Controller is flexible enough to enable DIYers and developers ideas and at the same time, to enable established professionals to take their signage business to the next, maybe a nationwide or a global level.

With the Pixblasters controller, it is finally possible to make professionally looking LED displays with addressable LED strips, without any of usual constraints on the size, graphics and video content, display management, storage capacity, or other limitation common to the competing LED controllers.

The Pixblasters video LED controller enables DIY makers to tackle any LED signage project in a cost-effective and flexible way - with no programming, no soldering, and with no more than a simple screwdriver.

The possibilities are endless; here are just a few we thought of to get you started:

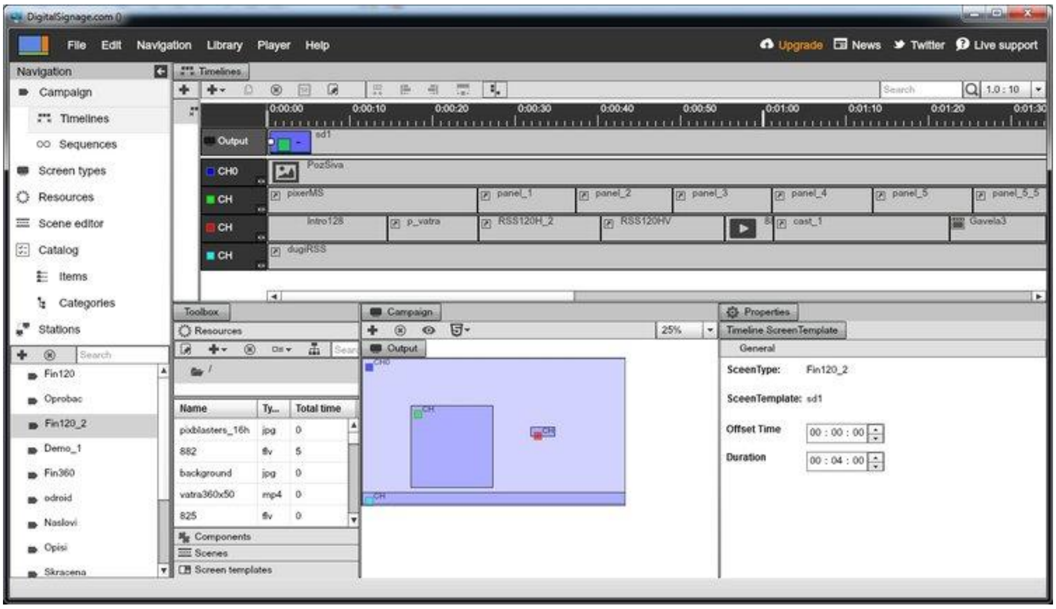
- Build a scoreboard for your school gym
- For musicians - incorporate a bold light feature into your stage set
- Holiday home decorations - elevate and customize!
- Artistic light installations or exhibitions
- Entrepreneur showing off a new idea or project

Signage professionals can use the Pixblasters MS1 Video LED Controller in many different applications. Small signage shops that make non-illuminated signs can expand their portfolio and offerings by incorporating the Pixblasters controller into their existing production workflow. Adding video LEDs can take any sign to the next level and make it more memorable.

Pixblasters can also be implemented (with small changes) within housings of existing illuminated signs. The installed video signs can be remotely controlled, so companies have more control over what kind of digital content they're displaying to their customers, and can make any changes needed quickly. Using the special LED strip's features to deploy very large and very cost-effective display installations will upgrade and improve your professional signage.

1.4 Easy Video Content Management

Pixblasters makes it easy to manage your video content. Instead of browsing the Internet in search of code snippets for specific LED patterns or odd software libraries that require guru-level software skills to format into an LED display image, Pixblasters users just need to plug in the monitor cable to display any graphics and video with no limitations. Stills and animations, AVI, FLV, WMV, MP4, etc. - what you see on the computer monitor connected to the selected computer is what you get on the Pixblasters MS1 controller-driven LED display with absolutely no coding required!



Name	Ty...	Total time
pixblasters_16h	jpg	0
882	flv	5
background	jpg	0
vatra360x50	mp4	0
825	flv	0

SceenType: Fin120_2

SceenTemplate: sd1

Offset Time 00 : 00 : 00

Duration 00 : 04 : 00

Figure 1.6 Example Digital Signage Software from MediaSignage, Inc. (aka DigitalSignage.com)

Pixblasters users can generate the LED display content in many different ways. For example, one can use the Windows Media Player animated visualizations to play music synchronized with video effects, or the Raspberry Pi Omxplayer video player to play MP4 video clips on the LED display. Those with special display applications (such as big real-time timetables and scoreboards) can develop custom software like graphics for a computer monitor.

The Pixblasters MS1 controller works with virtually any digital signage software, which allows users to operate and manage display content on the same level as high-end professional LED displays. This makes the generation of display content very easy. Either open source or fully commercial, digital signage software lets users quickly define scheduled playlists, combine video and graphics, incorporate news feeds and social networking posts, etc. Additionally, users can define the operating hours of a remote display, check what is currently playing, change the stored signage campaign on a single remote displays, or update multiple connected displays. Quality software enables the preview function and local campaign checkup prior to downloading to the video LED installation.

1.5 Key Features

- **Easy to use with any computer, any OS**:

  Pixblasters connects to any computer as an ordinary external monitor, with no programming needed. You can use it with Raspberry Pi, PC, media boxes, phones, etc. Simply plug in the monitor cable and the controller will smoothly drive the LEDs at the maximum 60fps frame-rate

- **Straightforward display content and management:**

  Anything that shows in the selected part of monitor image shows on the LEDs as well. No need for special software for anti-aliasing or fonts. Use any digital signage software and add layered screen divisions, text, animations, video, RSS... LEDs work at their maximum speed and are perfectly synchronized, no matter the number of LEDs and chained MS1 controllers, or the display size

- **Display size and resolution:**

  A single MS1 can control 4.7/17.7 m² (~ 50.6/190.5 ft²) while daisy-chained MS1s can control an LED display of hundreds of square meters using thirty to sixty LED strips. Daisy-chained controllers that drive 16,384 LEDs can support HD LED displays

- **Remote controllable and customizable:**

  Control it remotely from anywhere in the world through the network interfaces of the driving computer. Do not worry about the LED display's architecture. Users with special display applications (such as big, real-time timetables and scoreboards) can develop custom software for a computer monitor

- **Open source FPGA demo:**

  Use the hardware platform for experimenting with, and learning about, driving RGB LEDs.

> The open source FPGA demo design supports a subset of the fully-featured Pixblasters MS1 LED controller. It is available from the Pixblasters GitHub: https://github.com/PixiGreen/Pixblasters-MicroDemo.

1.6 Technical Characteristics

- **Xilinx® Spartan®-6 XC6LX9-3 FPGA chip**
  - Max number of different FPGA configurations is seven (7)
- **On-board Microchip PIC18F26J50 microcontroller:**
  - The microcontroller connects to the PC, runs the configuration software, and permanently stores the configuration data. It also enables simple firmware updates with any new features.
- **Output resolutions (H x V):** 512 x 32 (native), 256 x 64, 128 x 128, 180 x 96, etc.
- **LED details:**
  - 32 LED digital outputs: up to max. 512 LEDs per output
  - Maximum WS2812B/SK6812 LED display frame rate: 60 fps
  - Currently supported LED types: WS2811, WS2812, WS2812B, SK6812
  - Supports RGB666 (256K colors) color format
  - Maximum driving capacity of the single MS1 controller is 16,384 RGB LEDs
  - 5 VDC 24 mA digital outputs compatible with the most popular RGB LED types
- **Connectivity:**
  - DVI (HDMI mechanically) video input allows for easy connection from the computer - it also works with adapter cables should your computer have a DVI® output
  - Push-in strip power and signal connectors enable easy wiring without soldering
  - Validated with different computers and different OS's: Raspberry Pi Linux, Microsoft® Windows® PC, media boxes, phones and tablets…
  - On-board EDID flash enables automatic connection to any computer and any OS
- **Display controls:**
  - Cropping image window
  - Display formatting
  - LED timing parameters
  - Supports different video resolutions (max. 720p - 1280 x 720 at 60 fps)
  - Integrated video input cropping and mirroring
  - Multiple lines per output mode
  - LED gamma correction
- **Daisy-chain multiple controllers:**
  - Each MS1 board can work either as the Master or the Slave video controller
  - Up to 48 controllers chained together
  - Daisy-chained controllers can drive more than 200,000 perfectly synchronized LEDs differently arranged in HD LED displays
  - The high-speed video link between MS1 boards assures max. fps and synchronicity
  - low-speed control link interface (TBD)
  - Video and control links use common UTP cables (Ethernet)
  - On-board DIP switches allow for an easy selection of the Master, Slave and six other FPGA configurations, including user-defined FPGA configurations

- A single digital output has the ability drive multiple output video horizontal lines
- Configuration is fully customizable through a menu on the PC connected via the USB serial cable

- **Power and General Info:**

  - Protected 5 VDC ±5% power supply input
  - The LED power supplies need to be wired separately to fulfill the power and current requirements of giant LED displays
  - Board dimensions: 95 x 100 mm
  - Operating temperature: -40°C to +85°C

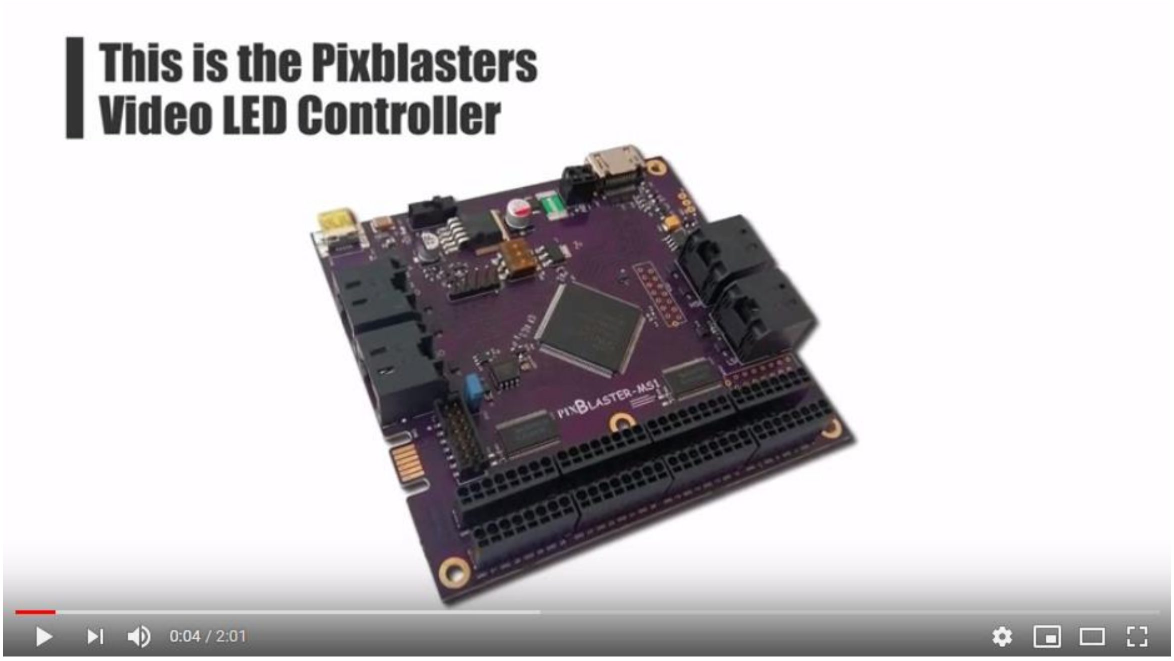
1.7 Pixblasters Support

For more information on the Pixblasters MS1 controller, please visit our official website www.pixblasters.com, or contact us at Pixblasters@gmail.com.

Follow us on social media:

Facebook: https://www.facebook.com/Pixblasters/
GitHub: https://github.com/PixiGreen/Pixblasters-MicroDemo
YouTube: https://www.youtube.com/channel/UCVbDHOoknUAESraVPJBDHkQ

Our YouTube channel brings comprehensive and illustrative how-to video clips that can help you to quickly start building great LED display installations:



This is the Pixblasters Video LED Controller

Figure 1.7 Example Pixblasters Video Clip

(https://www.youtube.com/watch?v=_aaxm1IZmkc&t=5s)

2 BOARD OVERVIEW

Figure 2.1 shows elements of the Pixblasters MS1 Video LED Controller board.

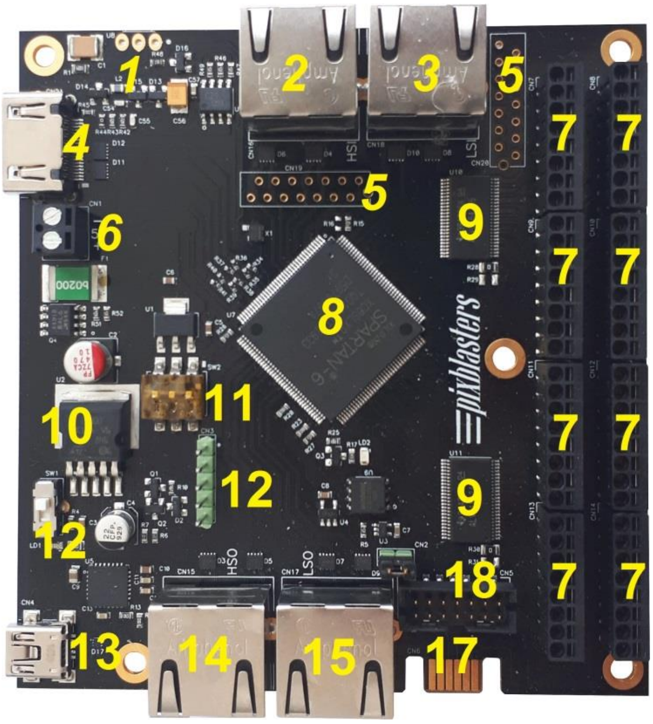


Figure 2.1 Pixblasters Controller Board Elements

1. IR receiver
2. High-Speed Input (HSI) port
3. Low-Speed Input (LSI) port
4. HDMI input
5. Expansion ports
6. Power supply input
7. LED strip ports
8. FPGA
9. Strip drivers
10. ON/OFF switch
11. FPGA configuration select
12. PIC µC (microcontroller) and its programming port
13. USB port
14. High-Speed Output (HSO) port
15. Low-Speed Output (LSO) port
16. FPGA program/run selector
17. FPGA JTAG & µC programming port
18. FPGA JTAG port.

2.1 IR Receiver (TSOP38238)

An optional IR receiver (not mounted) serves as a front end interface in IR remote control applications. IR remote control is not supported by default. If IR remote capability is required, user should mount IR receiver chip (TSOP38238) and provide the FPGA functionality to support targeted IR protocol.

This is USER MOUNTED OPTION! The IR receiver is not supported by Pixblasters production firmware and can be used in user-defined projects only.

2.2 High-Speed Input (HSI) Port

In a system comprised of multiple Pixblasters controllers chained together, the HSI port receives the video signal from the neighboring "upstream" device. For more information go to chapter **Multi-Controller Topology**.

2.3 Low-Speed Input Port

In a system comprised of multiple Pixblasters controllers chained together, the LSI port receives the control signals from the neighboring "upstream" device. LSI port functionality is currently not supported. For more information go to chapter **Multi-Controller Topology**.

Future expansion. Currently unused.

2.4 HDMI Video Input Port

This is the main, DVI-compatible video input port that uses the HDMI compatible connector. In a system comprised of multiple Pixblasters controllers chained together, the main video signal source (**Figure 1.4**) must be connected to this video input port of the uppermost device (Master board). For more information please read the chapter **Multi-Controller Topology**.

2.5 Expansion Ports

Expansion ports (CN19 and CN20) provide direct access to a number of FPGA pins. User may use these pins for connecting its own expansion hardware. The CN19 and the CN20 footprints are intended to accept the TE Connectivity's 1-215079-4 connectors.

This is USER MOUNTED OPTION! IMPORTANT: CN19 and HSI port share the same FPGA pins! This feature is not supported by Pixblasters production firmware and can be used in user-defined projects only.

2.6 Power Supply Input

The main 5V power supply input for the Pixblasters MS1 controller.

⚠	To prevent the short circuit, please respect the PCB voltage markings during wiring.

2.7 LED Strip Ports

These are LED strips wiring connections. While the ground (GND) connections are available on these ports, the power must be connected outside of the Pixblasters board. It is necessary due to possibly very high power requirements that surpass conducting capabilities of electronics PCB boards.

2.8 FPGA

Xilinx XC6SLX9 Spartan-6 FPGA chip with custom Pixblasters designs.

2.9 LED Strip Drivers

SN74ALVC164245DGGT FPGA LED strip driver ICs.

2.10 ON/OFF Switch

The main power switch for the complete Pixblasters MS1 board.

2.11 FPGA Configuration Selector

The Pixblasters MS1 Video LED Controller boards come with several FPGA configuration files stored in the on-board non-volatile flash memory. Use this switch (**Figure 8.8**) to select FPGA configuration to load into the FPGA chip. Different FPGA configurations are used to support different kinds of LED strips. Up to 7 (1 - 7) different configurations may be used. Any of these 7 configurations may be updated by user. Configuration '0' serves for updates of stored FPGA configurations and cannot be modified by user.

2.12 PIC Microcontroller and its Programming Port

The Microchip PIC18F26J50 microcontroller connects to the PC, runs the configuration software, and permanently stores the configuration data. It also enables simple firmware updates with new features. The programming port for the on-board PIC18F26J50 microcontroller accepts 2.54 mm pin header connector.

🚫	The original configuration of the on-board Microchip PIC18F26J50 microcontroller **MUST NOT** be changed under any circumstances and any change will permanently cease the operation of the Pixblasters firmware. The PCB board and all user-defined FPGA designs will continue to work with no limitations!

2.13 USB Port

The USB device port enables Pixblasters controller connection to the PC USB host port.

2.14 High-Speed Output (HSO) Port

In a system comprised of multiple Pixblasters controllers chained together, the HSO port sends the video signal to the connected neighboring "downstream" device. For more information go to chapter **Multi-Controller Topology**.

ℹ Future expansion. Currently unused.

2.15 Low-Speed Output (LSO) Port

In a system comprised of multiple Pixblasters controllers chained together, the LSO port sends control signals to the connected neighboring "downstream" device. The LSO port functionality is currently not supported. For more information please read the chapter **Multi-Controller Topology**.

ℹ Future expansion. Currently unused.

2.16 FPGA Program/Run Selector

The on-board reset generator (U3) requires constant refresh from the FPGA. In normal operation, when the FPGA provides reset generator refresh, the CN2 should be left open.

⚠ If JTAG is used to program the FPGA, or if user utilizes the FPGA configuration that does not provide reset generator refresh, the program/run selector (CN2) should be shorted.

2.17 FPGA JTAG & µC Programming Port

This PCB connector provides access to both, JTAG and µC JTAG programming pins. Port accepts RBB06DHHN connector.

2.18 FPGA JTAG Port

The FPGA JTAG port for direct FPGA JTAG pins access (configuration).

3 BUILDING LED STRIPS DISPLAY USING PIXBLASTER CONTROLLER

3.1 About LED Strips

The current Pixblasters MS1 Video LED Controller version supports WS2811, WS2812, WS2812B and SK6812 RGB LEDs. In the future we plan to add more control functions, like support for the APA102 LED strips. Adding new features to field deployed Pixblasters controllers is already possible and very easy. We will distribute new configuration files with the upgraded features sets and the Pixblasters users will update the controller through a simple procedure that does not require special tools or knowledge. Just connect the Pixblasters controller to the PC via a serial USB cable, use on-board DIP switches to select the firmware flasher FPGA configuration and configure the controller through a simple user interface.

Our preferred LED strip model is the WS2812B (**Figure 3.1**) because it enables the fastest LED operation, and combined with the Pixblasters MS1 video LED controller, full 60 frames per second (fps) video refresh rate. This is much faster than it is possible with the majority of competing controllers and enables smooth video reproduction regardless of the LED display's size.

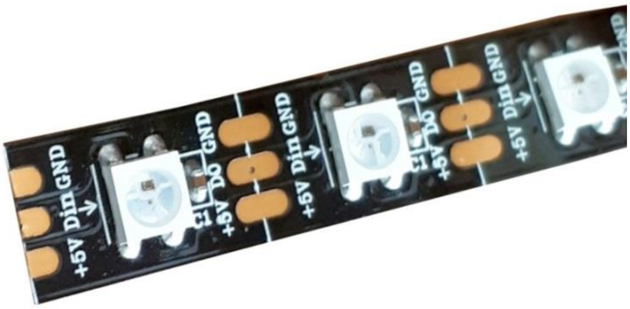


Figure 3.1 The WS2812B LED Strip Interface Pins

The WS2812B RGB LED strips have the 3-pin interface: +5V power input, GND ground pin, and the DIN control data input.

Note the small arrow sign on the LED strip. It shows the data flow direction and it **must be respected at all times**, or the LED strip will not light. The LED strip must be always connected to the Pixblasters video controller with the arrow pointing away from the board's digital output.

The **Figure 3.2** shows a simple LED video display formatting tool – the scissors. Use it to cut the video display line to the requested length.

The WS2812B LED strip can be cut between any two LEDs. The cutting positions for some other strips can be less flexible, i.e. some can be cut behind groups of three LEDs.

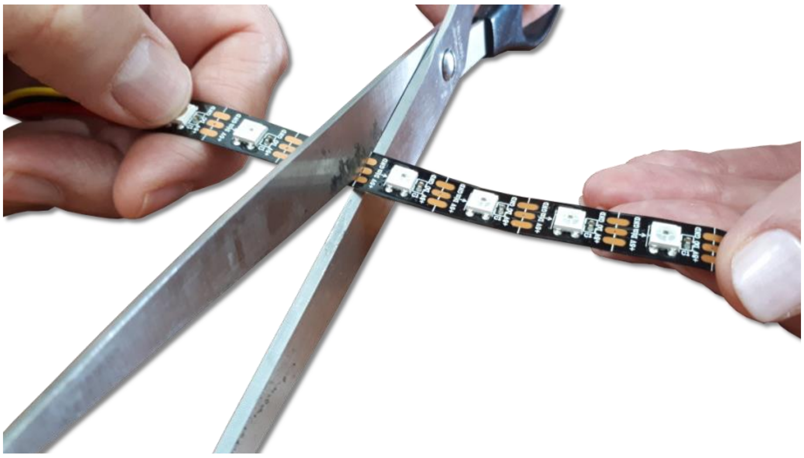


Figure 3.2 Define the LED Strip's Size and Display's Format

LED strips can be used without soldering. While being skilled with the soldering iron is definitely helpful, one can use the locking connectors, like the one shown on Figure 3.3, to connect the wire leads to the strip without soldering. The wire leads with the removed isolation simply insert into the Pixblasters controller's high-quality snap-action connectors. This way, the reliable contact between the strip and the controller can be established without soldering.

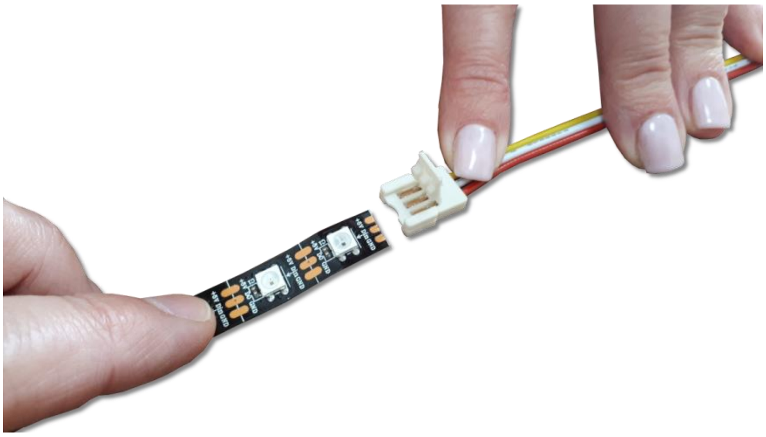
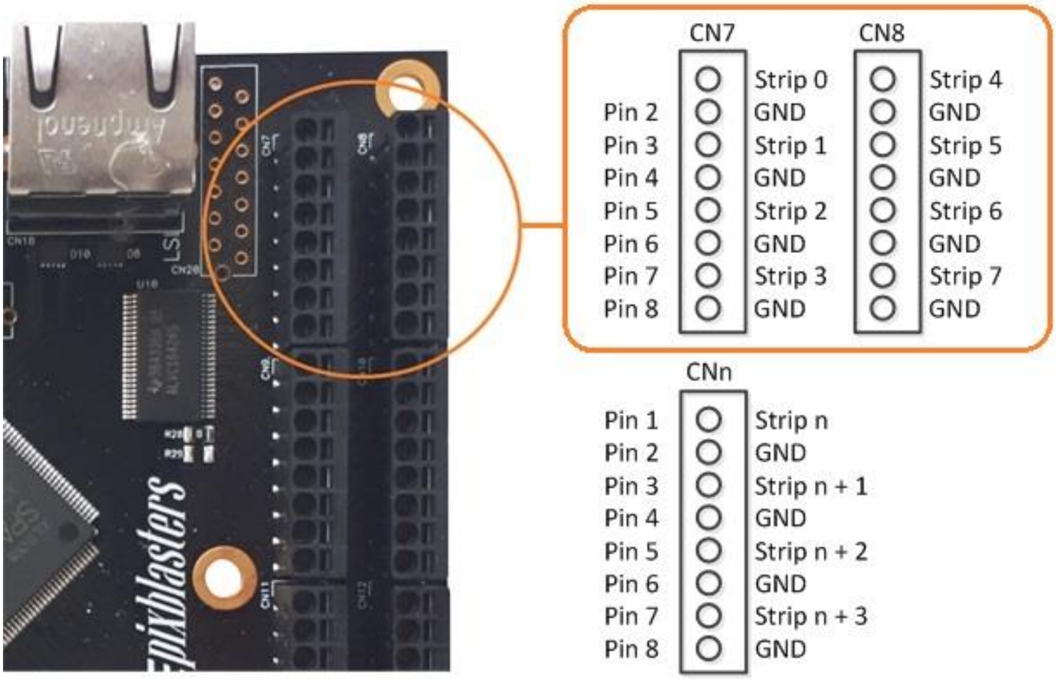


Figure 3.3 An Example Locking Connector for LED Strips

Besides the connector type shown on Figure 3.3, there are different types locking connectors that connect two pieces of the LED strip.

3.2 Connecting LED strips

The Pixblasters MS1 controller has 8 LED strip connectors CN7 - CN14. Figure 3.4 shows LED strip connector layout.



CN7			CN8	
	○	Strip 0	○	Strip 4
Pin 2	○	GND	○	GND
Pin 3	○	Strip 1	○	Strip 5
Pin 4	○	GND	○	GND
Pin 5	○	Strip 2	○	Strip 6
Pin 6	○	GND	○	GND
Pin 7	○	Strip 3	○	Strip 7
Pin 8	○	GND	○	GND

CNn		
Pin 1	○	Strip n
Pin 2	○	GND
Pin 3	○	Strip n + 1
Pin 4	○	GND
Pin 5	○	Strip n + 2
Pin 6	○	GND
Pin 7	○	Strip n + 3
Pin 8	○	GND

Pixblasters MS1 Board Detail

LED Strip Connector Labeling

Figure 3.4 LED Strip Connector Layout

shows the pinout of all eight LED strip connectors.

Each LED strip is connected to Pixblasters controller using one "Strip" and one "GND" pin. LED strip power supply is not connected to the Pixblasters controller directly. Separate Power and GND lines must be used for applying power to the LED strip (**Figure 5.1**).

Each RGB LED draws approximately 50 mA when it is set to full brightness white color and powered at V_{dd} = +5V, and this information helps the proper power supply selection. An example calculation for a 2,000 LEDs display:

Max. current: 2000 x 0,05A = 100A;

Max. DC power: P = V_{dd} * I = 5V * 100A = 500W

500W is valid for the worst case when all LEDs display white color. Since displays shows changing and colorful video, it enables use of power supplies of lower nominal power. This underpowered operational mode enables cost savings. Depending on the video content, the above 2,000 LEDs example display would comfortably work powered by 350W power supply.

Table 3.1: Pinout of LED Strip Connectors

Connector	Pin	Description	Pin	Description	32/512	16/512*
CN7	1	Strip 0	2	GND	Active	Active
CN7	3	Strip 1	4	GND	Active	Active
CN7	5	Strip 2	6	GND	Active	Active
CN7	7	Strip 3	8	GND	Active	Active
CN8	1	Strip 4	2	GND	Active	Active
CN8	3	Strip 5	4	GND	Active	Active
CN8	5	Strip 6	6	GND	Active	Active
CN8	7	Strip 7	8	GND	Active	Active
CN9	1	Strip 8	2	GND	Active	Active
CN9	3	Strip 9	4	GND	Active	Active
CN9	5	Strip 10	6	GND	Active	Active
CN9	7	Strip 11	8	GND	Active	Active
CN10	1	Strip 12	2	GND	Active	Active
CN10	3	Strip 13	4	GND	Active	Active
CN10	5	Strip 14	6	GND	Active	Active
CN10	7	Strip 15	8	GND	Active	Active
CN11	1	Strip 16	2	GND	Active	-
CN11	3	Strip 17	4	GND	Active	-
CN11	5	Strip 18	6	GND	Active	-
CN11	7	Strip 19	8	GND	Active	-
CN12	1	Strip 20	2	GND	Active	-
CN12	3	Strip 21	4	GND	Active	-
CN12	5	Strip 22	6	GND	Active	-
CN12	7	Strip 23	8	GND	Active	-
CN13	1	Strip 24	2	GND	Active	-
CN13	3	Strip 25	4	GND	Active	-
CN13	5	Strip 26	6	GND	Active	-
CN13	7	Strip 27	8	GND	Active	-
CN14	1	Strip 28	2	GND	Active	-
CN14	3	Strip 29	4	GND	Active	-
CN14	5	Strip 30	6	GND	Active	-
CN14	7	Strip 31	8	GND	Active	-

** FPGA configurations with support for 16 LED outputs do not use connectors CN11 – CN14*

Twisting wires that connect LED strips to the Pixblasters MS1 is highly recommended. It removes the electromagnetic interference between the wires, assures more reliability and flicker-free LED display operation. Connecting all GND pins between the Pixblasters LED controller and LED strips is also highly recommended.

3.3 Building LED Matrix

LED strips can be arranged in LED display's matrix driven by the Pixblasters controller in two main ways. The **Figure 3.5** shows the LED strips arrangement without strip segmentation. Each of 32 Pixblasters MS1 controller outputs drives a display line of maximally 512 LED pixels.

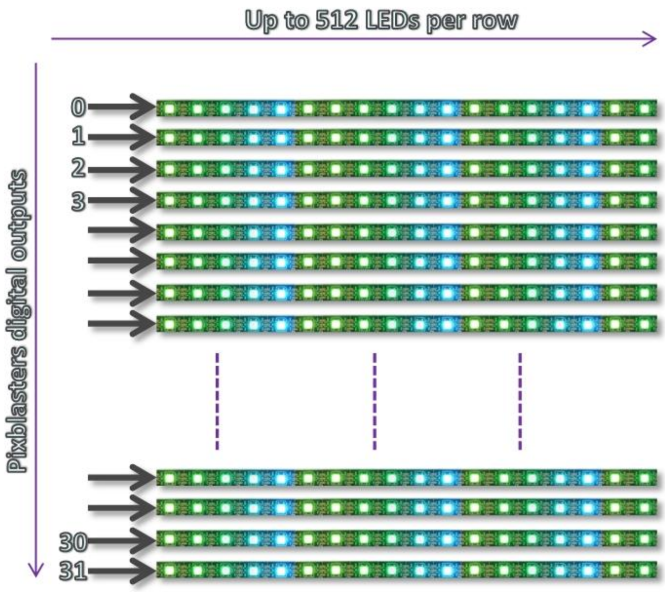


Figure 3.5 LED Strips Matrix with no Segmentation

Figure 3.6 shows the LED strips arrangement with the line segmentation, which enables more flexibility in display's resolution setup. Each Pixblasters MS1 controller's digital output can drive max. 512 pixels, but those pixels can be arranged in multiple display lines. The **Figure 3.6** example shows 4 lines driven by a single Pixblasters digital output and this arrangement enables the maximum resolution 128 x 128 (Horizontal x Vertical) pixels.

The Pixblasters controller needs to be configured, through the configuration menus, for such LED matrix arrangement and the further display control is fully automatic. There is no need for any input image manipulation.

The LED matrix with the embedded segmentation requires a bit different LED strips wiring and it is absolutely necessary to take care on the data flow direction. Take for example the four horizontal lines controlled by the digital output 0 from the **Figure 3.6** example. The first LED strip segment must have the data arrow (**Figure 3.1**) pointing from left to right, the second LED segment must have the data arrow pointing from right to left, the third one from left to right, and the fourth from right to left. Such LED arrangements assures an uninterrupted serial data flow.

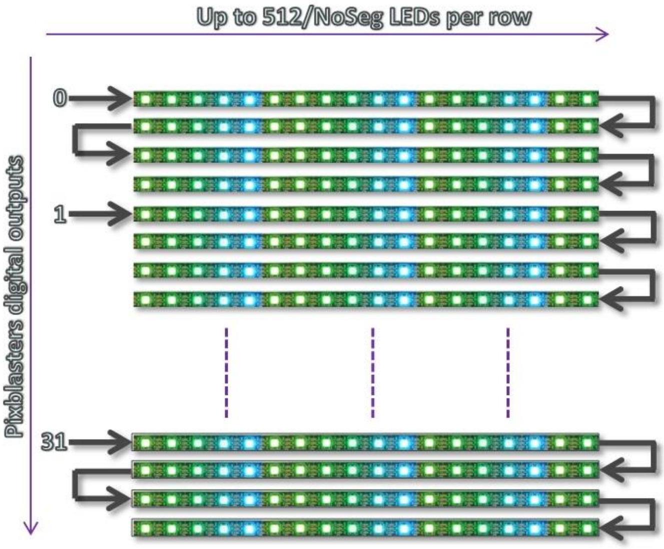
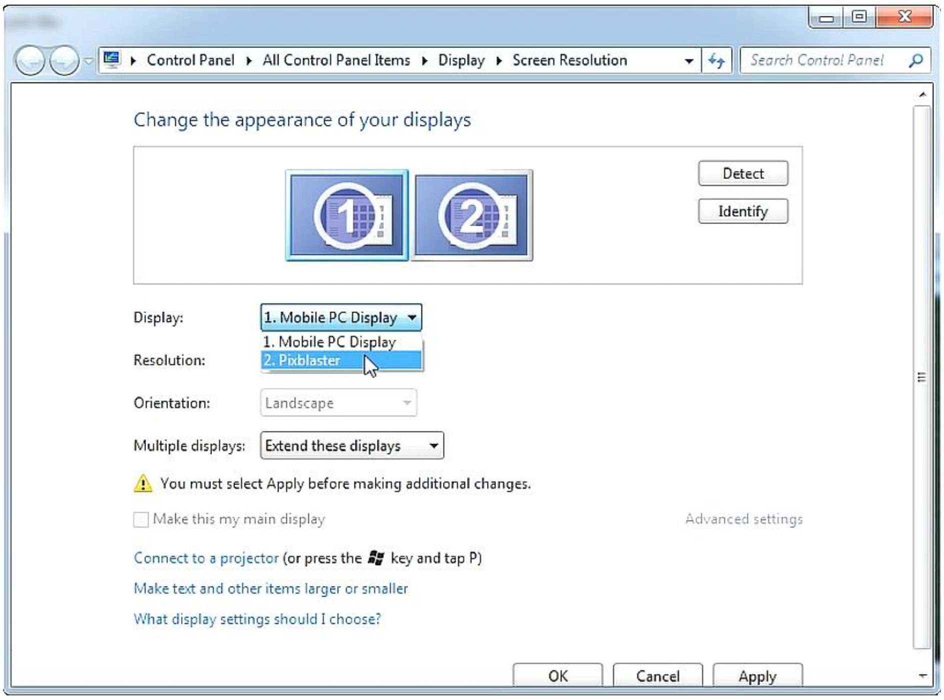


Figure 3.6 LED Strips Matrix with Segmentation

3.4 Connecting Video Source to the Pixblasters MS1 Controller

Most consumer monitors these days carry inside an EEPROM chip called the EDID. It contains monitor's name and technical info about its capabilities. This information enable any standard video source, i.e. the PC computer, to properly detect the monitor and automatically format the video image suitable for the specific monitor. The Pixblasters MS1 Video LED Controller also includes Extended Display Identification Data (EDID) chip to smoothly connect to any operating system as a standard monitor.


Control Panel ▸ All Control Panel Items ▸ Display ▸ Screen Resolution

Change the appearance of your displays

Display: 1. Mobile PC Display
1. Mobile PC Display
2. Pixblaster

Resolution:

Orientation: Landscape

Multiple displays: Extend these displays

You must select Apply before making additional changes.

□ Make this my main display

Search Control Panel
1
2
Detect
Identify
Advanced settings
Connect to a projector (or press the key and tap P)
Make text and other items larger or smaller
What display settings should I choose?
OK
Cancel
Apply

Figure 3.7 The Pixblasters Controller Recognized as a Monitor by Microsoft Windows OS

The recommended length of the input video cable is up to 1 meter. In case of video sources with weaker driving capabilities and longer cables between the video source and the Pixblasters MS1 LED controller, a noticeable flickering of the LED display is possible. In such situations we recommend use of active signal conditioners, such as the active HDMI video cable, HDMI splitters or HDMI signal repeaters. Please note that the Pixblasters MS1 is DVI video compatible and the listed HDMI equipment should smoothly pair with it.

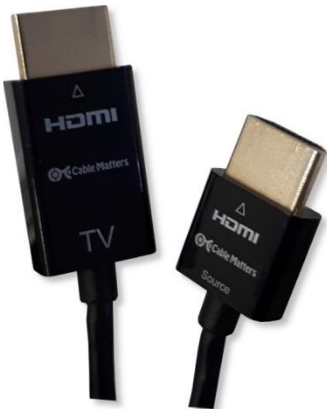


Figure 3.8 Active HDMI Cable

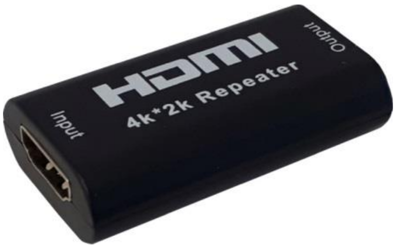


Figure 3.9 HDMI Repeater (Cable Extender)

Please respect the required signal direction as labeled on video equipment. For example, the Source side of the active video cable should be connected to the video source, while the TV side of the cable connects to the Pixblasters MS1 video input.

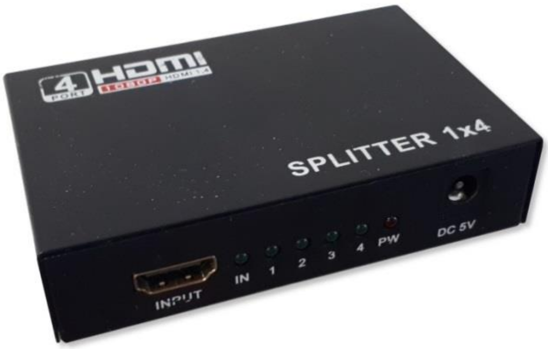


Figure 3.10 Example HDMI Splitter – Input

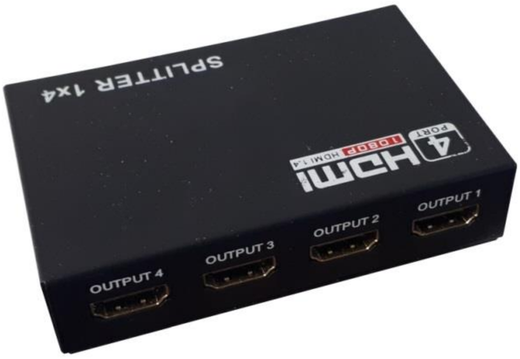


Figure 3.11 Example HDMI Splitter – Outputs

4 SYSTEM TOPOLOGIES

4.1 Single Controller Topology

There are currently two main topologies in which Pixblasters controllers may be arranged – Single controller topology and Multi-Controller topology. Each Pixblasters MS1 controller, depending on the loaded FPGA configuration, supports either up to 32 strips/512 pixels per strip, or 16 strips/512 pixels per stripe combinations. If the 16/512 combination is used, only controller outputs 0 - 15 are active. (See chapter **3.2 - Connecting LED strips**)

As its name suggests, the single controller topology uses only one Pixblasters MS1 controller to drive LED strips. In that topology, depending on the loaded FPGA configuration, the LED display may consist of up to 32/512 or 16/512 LED strips. Single controller topology is further subdivided into "Non-segmented" and "Segmented" topologies.

4.1.1 Non-Segmented Single Controller Topology

Figure 4.1 shows the Pixblasters MS1controller in "Non-segmented single controller" topology. "Non-segmented" means that each LED strip contains a single video line. When "Non-segmented" mode is used, vertical resolution is limited to the number of strips connected to Pixblasters controller.

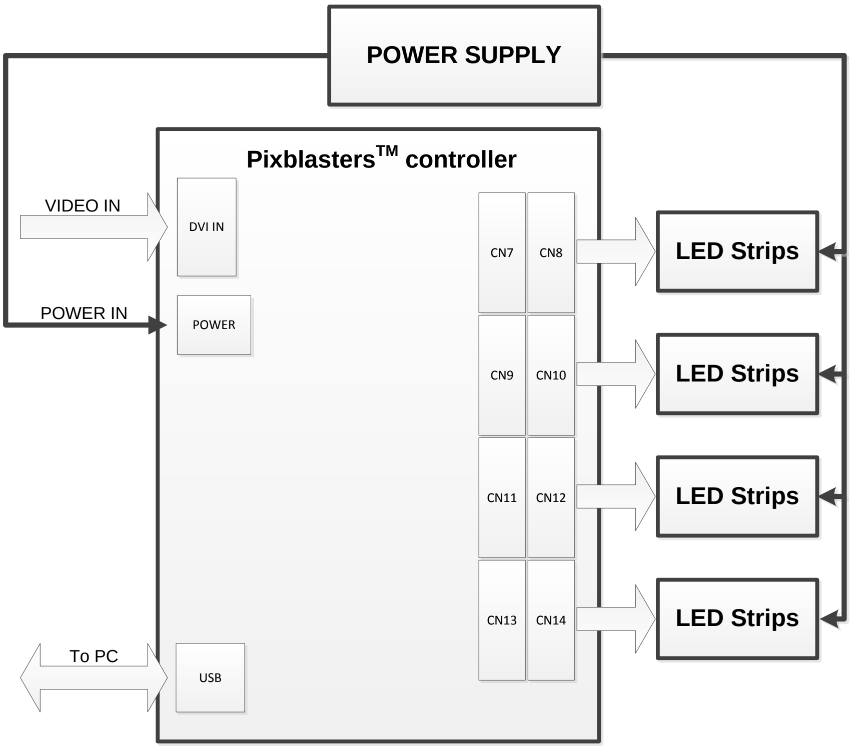


Figure 4.1 Single Controller Topology

4.1.2 Segmented Single Controller Topology

In terms of how LED strips are connected to the Pixblasters MS1 controller, "Segmented" and "Non-segmented" topologies are identical. Therefore, block schematic shown in the **Figure 4.1** still applies. What is different is the way how LED strips are used and configured. In "Segmented" topology each LED strip may contain several video lines (see **Figure 3.6**). A number of lines depend on the value stored in the **SegmentsNO** register. This way, vertical resolution is no longer limited to the number of connected LED strips. Since LED strip is treated by Pixblasters controller as segmented, it has to be physically configured (broken into segments) as shown in **Figure 3.6** and **Figure 4.2**.

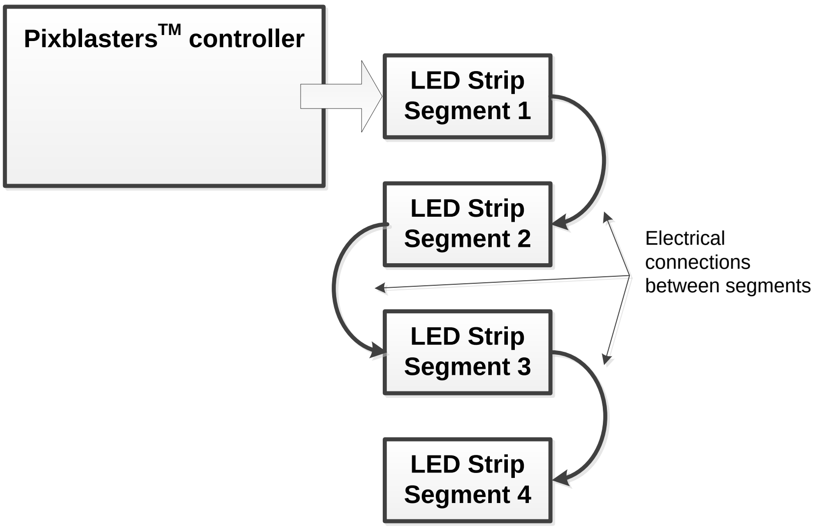


Figure 4.2 LED Strip Configuration for "Segmented" Topology

4.2 Multi-Controller Topology

Multi-controller topologies use more than one Pixblasters MS1 controller to drive LED strips. In that case, system has one "Master" controller to which the video source is connected, and up to 48 downstream "Slave" controllers chained together using standard Ethernet cable. Please, see the chapter **Powering the Pixblasters controller based LED display** for more information.

Each Pixblasters MS1 controller, depending on the loaded FPGA configuration, supports up to 32 strips/512 pixels per strip, or 16 strips/512 pixels per stripe combinations. If the 16/512 combination is used, only controller outputs 0 - 15 are active. (See chapter **Connecting LED strips**). As in the "Single controller topology", LED strips may also be "Segmented" or "Non-segmented". Multi-controller topology is further subdivided into "Non-mirrored" and "Mirrored" topologies.

4.2.1 Non-Mirrored Multi-Controller Topology

Figure 4.3 and **Figure 4.4** show how Pixblasters LED controllers may be connected to build a "Non-mirrored" multi controller display. The only difference between these two examples is related to a power supply configuration. More details about the power supply connections are given in the chapter **Powering the Pixblasters controller based LED display**.

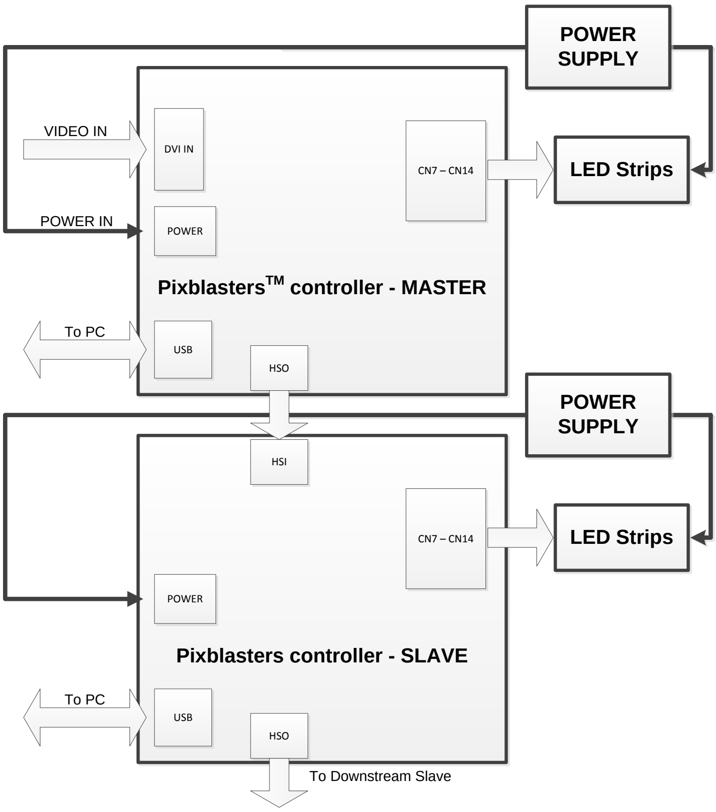


Figure 4.3 Multi-Controller Topology – Separate Power Supplies

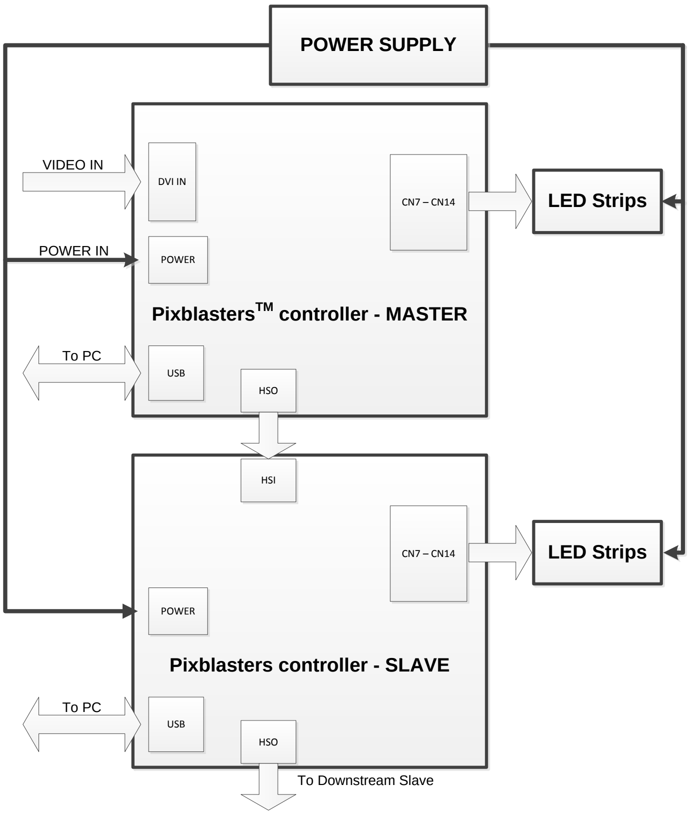


Figure 4.4 Multi-Controller Topology – Common Power Supply

4.2.2 Mirrored Multi-Controller Topology (TBD)

When used in the "Non-mirrored topology", depending on the loaded FPGA configuration, display's horizontal resolution is limited to 512 pixels. The "Mirrored" mode is used to overcome that limit and effectively double the horizontal resolution. Figure 4.5 shows how Pixblasters controllers should be connected when "Mirrored" mode is used. It can be seen that controllers are arranged in two adjacent columns - Figure 4.5.

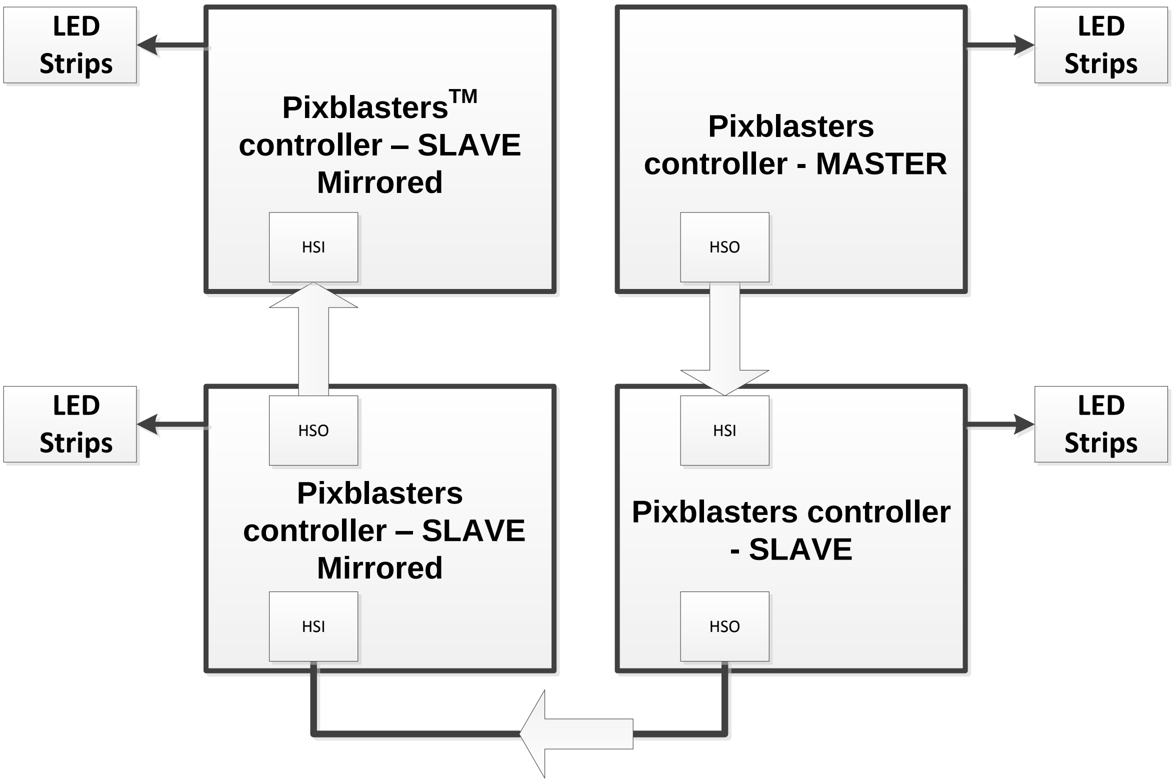

LED Strips
Pixblasters™ controller – SLAVE Mirrored
HSI
Pixblasters controller - MASTER
HSO
LED Strips
LED Strips
HSO
Pixblasters controller – SLAVE Mirrored
HSI
HSI
Pixblasters controller - SLAVE
HSO
LED Strips

Figure 4.5 Mirrored Multi-Controller Topology

Pixblasters controllers in the right column are "non-mirrored" and should be configured to handle right portion of the video lines. Pixblasters controllers in the left column should be "mirrored" and configured to handle the left portion of the video lines. Figure 4.6 shows display structure when "Mirrored" multi-controller topology is used

LED Strips L1	LED Strips R1
LED Strips L2	LED Strips R2

Figure 4.6 Display Structure in the Mirrored Multi-Controller Topology

IMPORTANT: The mirroring feature will be supported with the firmware release planned in 2Q2020!

4.3 Multi-controller Demo Display Architecture

Figure 4.7 shows Pixblasters demo display developed for designing and testing purposes. This small maquette contains all parts necessary for building big and segmented LED displays.

There is a single master (M) Pixblasters MS1 controller that receives, through the monitor cable (1), video from the Raspberry Pi video source (VS). Slaves (S1) and (S2) MS1 LED controllers are interconnected by UTP Ethernet (2) cables that propagate the input video image. LED outputs (3) are connected to LED strips, which are placed on the opposite side of a mechanical structure and invisible on the photo.

The complete system is powered by (4) power supplies.

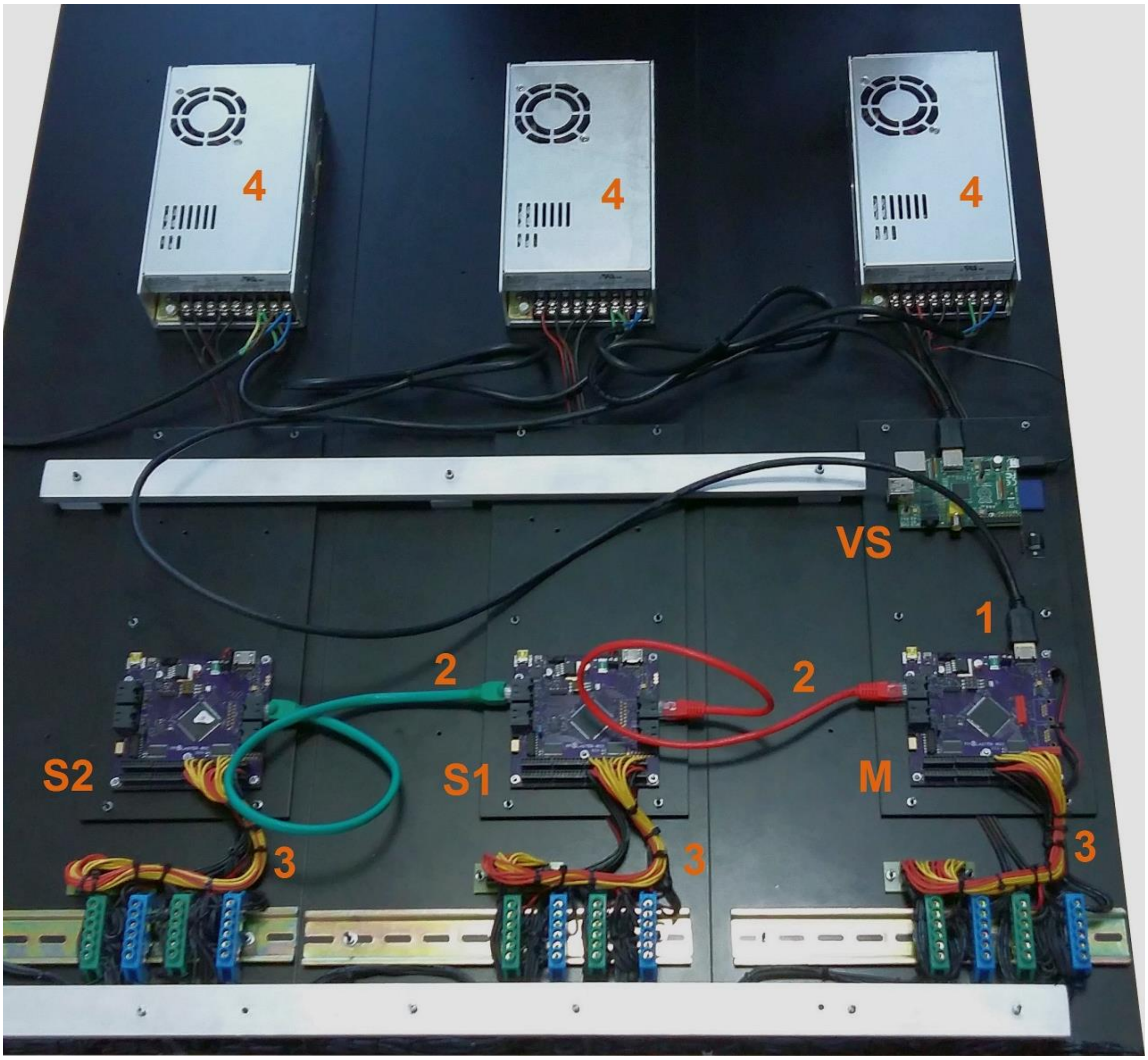


Figure 4.7 Demo Display Structure in the Multi-Controller Topology

5 POWERING THE PIXBLASTERS CONTROLLER BASED LED DISPLAY

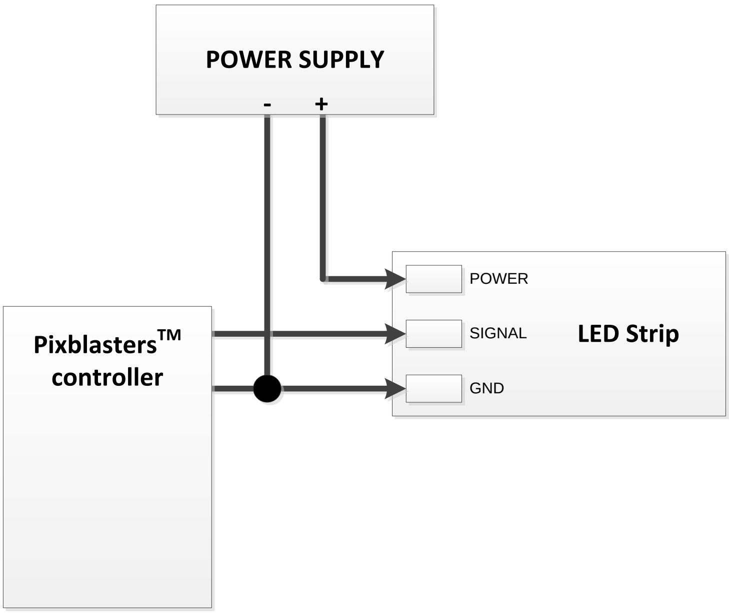


Figure 5.1 Connecting LED Strip to Power Supply and Controller

Power supply topology will mainly depend on the current consumption of the connected LEDs and on the power supply's current rating. In a single controller topology, one power supply may be enough to provide power for the whole system. If, however, current consumption of the connected LEDs requires multiple power supplies, than the scheme shown in **Figure 5.2** should be used.

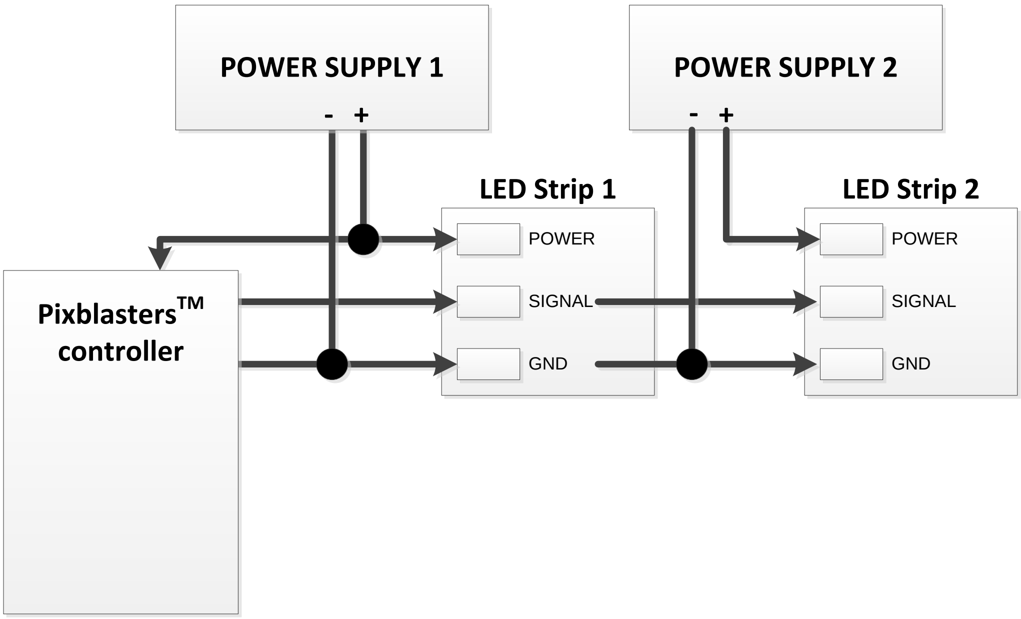


Figure 5.2 Multiple Power Supplies in a Single Controller Topology

Figure 5.2 assumes that LED strip and Pixblasters controller use the same power supply voltage (5V). In that case, the Pixblasters controller should be connected to the power supply which powers the strip directly connected to the Pixblasters controller.

In the multi-controller topology, power supply connecting scheme will again mainly depend on current consumption of the connected LEDs and on the power supply's current rating. If power supply can provide enough current for all display elements (LEDs and controllers), than scheme shown in Figure 5.3 should be used. Again, this scheme assumes that LED strip and Pixblasters controller use the same power supply voltage (5V).

🚫	If the Pixblasters controllers use the same power supply, video link HSO and HSI ports should be connected by Unshielded-Twisted-pair (UTP) cable! Notice the metal shield shown in Figure 5.5 that explains differences between STP and UTP cables.

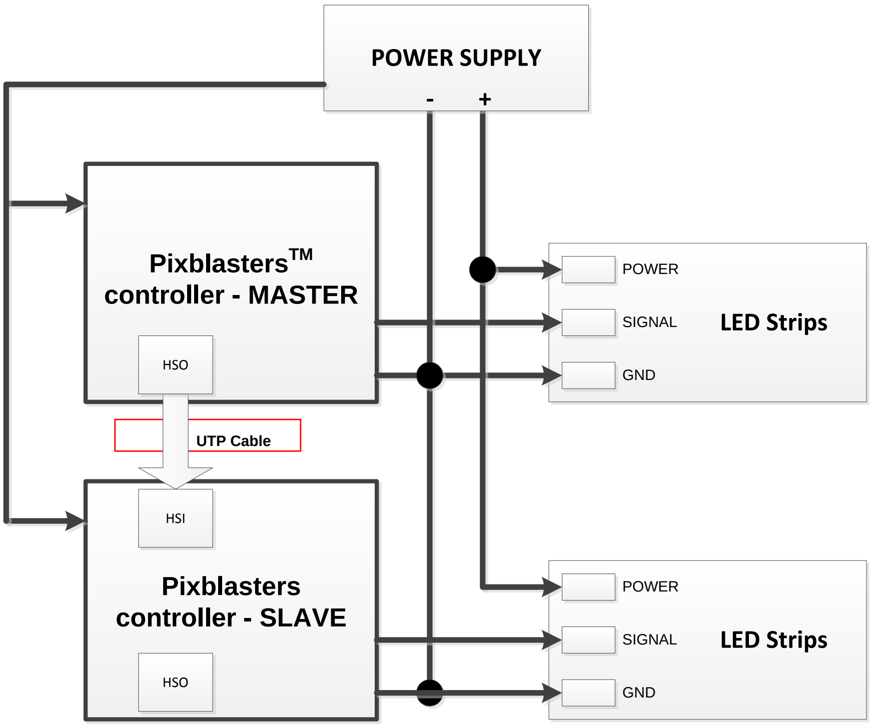


Figure 5.3 Single Power Supply in a Multi-Controller Topology

In larger systems, one power supply will most likely be insufficient to power all LEDs and MS1 video controllers. In that case, the power supply scheme may be as shown in Figure 5.4. Again, there is a rule covering HSO-HSI connections.

🚫	If the Pixblasters controllers do not use the same power supply, video link HSO and HSI ports should be connected by the Shielded-Twisted-pair (STP) cable! Notice the metal shield shown in Figure 5.5 that explains differences between STP and UTP cables.

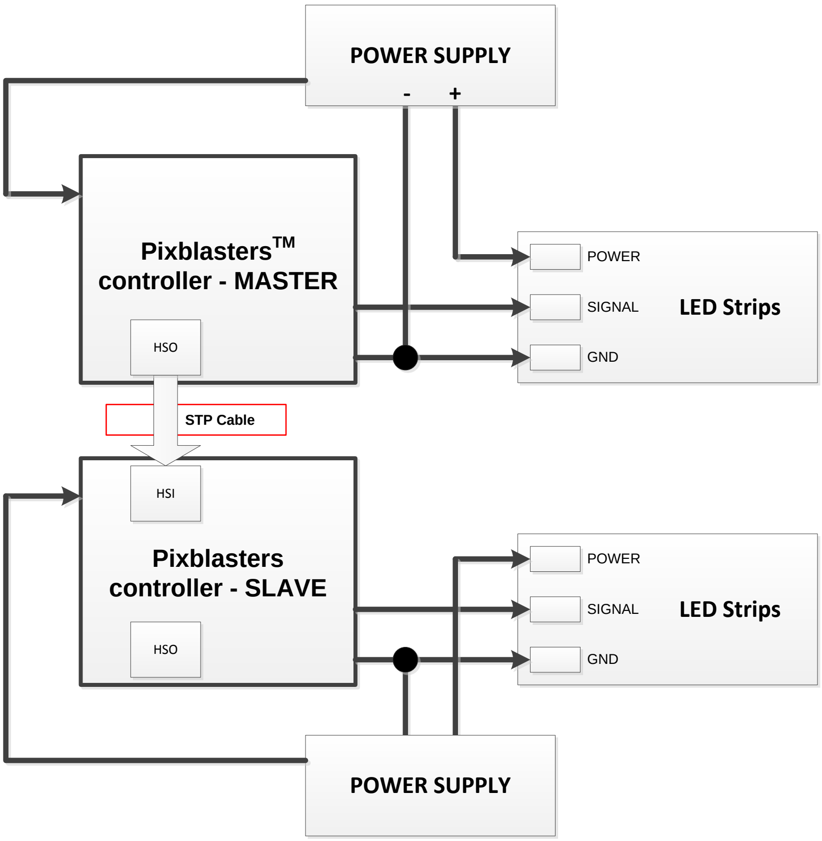


Figure 5.4 Multiple power supplies in Multi-controller topology

The power supply scheme shown in Figure 5.2 is also applicable for multi-controller topologies. Again, if LED strips and Pixblasters controller use the same power supply, the Pixblasters controller should be connected to the power supply which powers the strip directly connected to the Pixblasters controller.

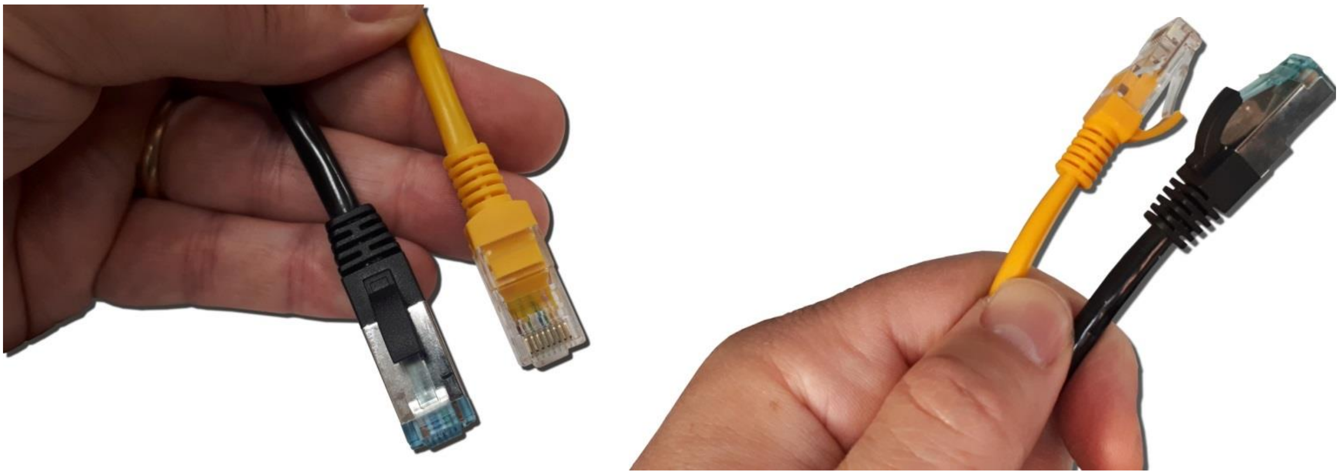


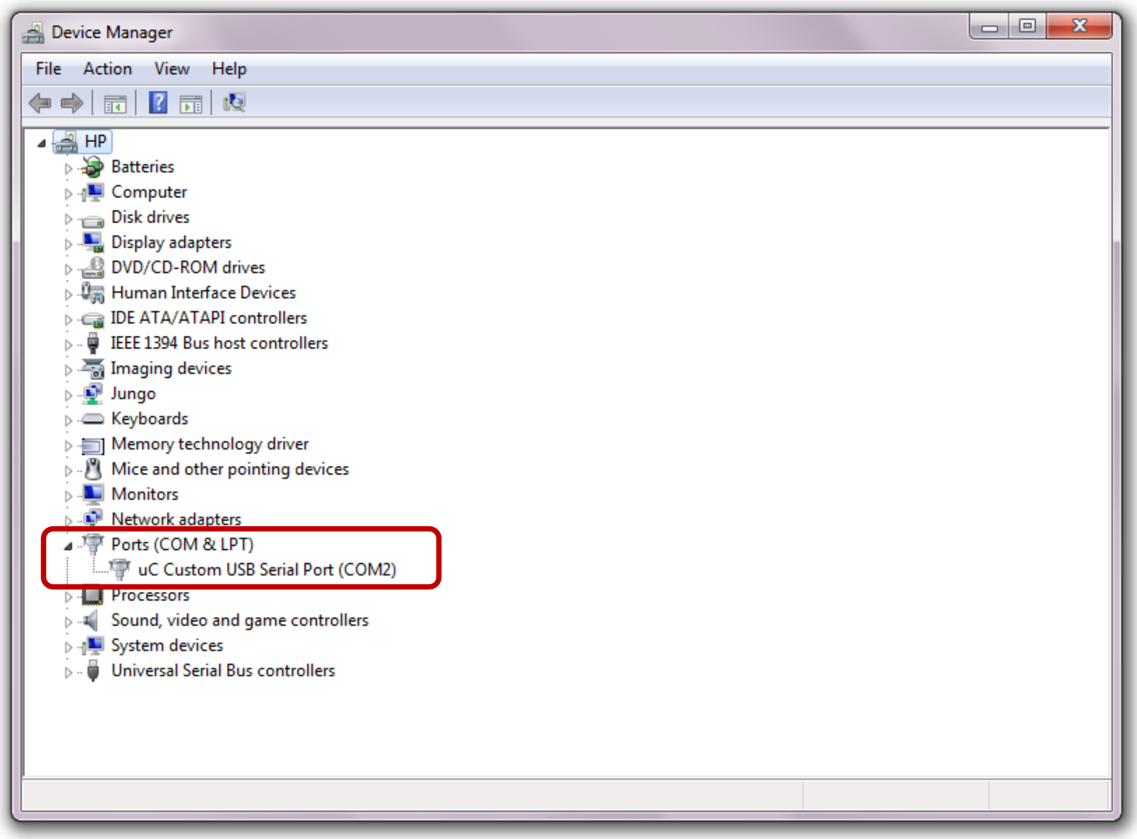
Figure 5.5 The STP (black) and the UTP (yellow) Cables

6 USER CONTROL INTERFACE

6.1 PC Connection to the Pixblasters MS1 Controller

To adjust the Pixblasters MS1 Video LED Controller to a specific display configuration, various controller parameters have to be set and permanently stored in the non-volatile flash memory. This procedure requires the USB serial connection between the Pixblasters controller and a PC.
If the LED display utilizes multiple MS1 controllers, each controller has to be configured separately. Of course, if the configuration of the LED display never changes, the configuration process must be executed only once!

On Microsoft Windows PC machines, upon the Pixblasters controller connection, a new COM port will be visible in the **Control Panel\Device Manager**.

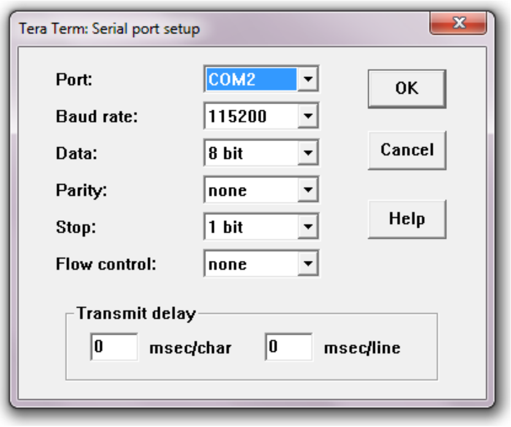


- HP
  - Batteries
  - Computer
  - Disk drives
  - Display adapters
  - DVD/CD-ROM drives
  - Human Interface Devices
  - IDE ATA/ATAPI controllers
  - IEEE 1394 Bus host controllers
  - Imaging devices
  - Jungo
  - Keyboards
  - Memory technology driver
  - Mice and other pointing devices
  - Monitors
  - Network adapters
  - Ports (COM & LPT)
    - uC Custom USB Serial Port (COM2)
  - Processors
  - Sound, video and game controllers
  - System devices
  - Universal Serial Bus controllers

Figure 6.1 New USB Serial Connection Visible in the Windows Device Manager

The next step requires use of any terminal emulator application – the Pixblasters development team preference is the Tera Term, an open-source, free, software implemented, terminal emulator program. The Tera Term project home: https://ttssh2.osdn.jp/.

The Tera Term is used in all examples described within this document. Start the Tera Term (or equivalent program of your choice) and connect the Pixblasters controller through the related COM port (COM2 in this instance). The terminal emulator settings must be setup as shown in **Figure 6.2**.



Port: COM2

Baud rate: 115200

Data: 8 bit

Parity: none

Stop: 1 bit

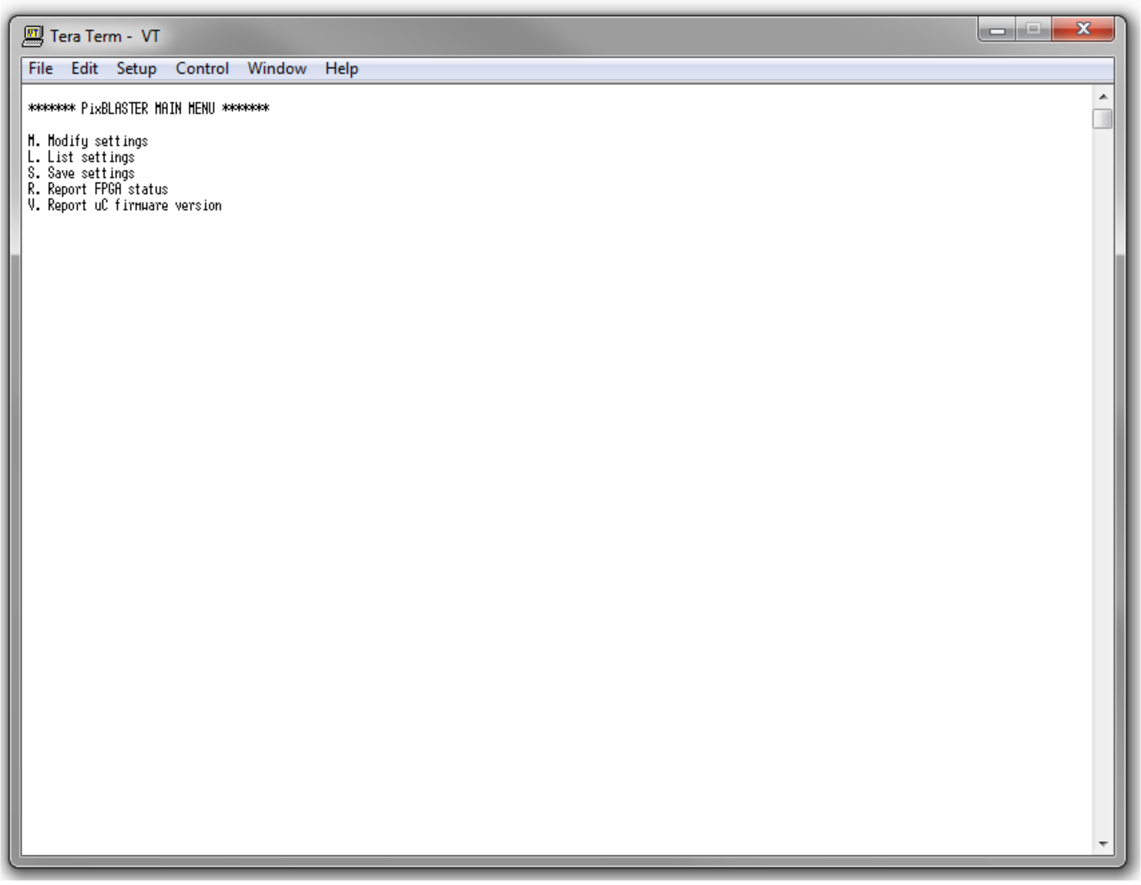
Flow control: none

Transmit delay

0 msec/char 0 msec/line

Figure 6.2 Serial Connection Parameters

After the established connection, hit the **Enter** key and the following menu should appear in the terminal emulator window:

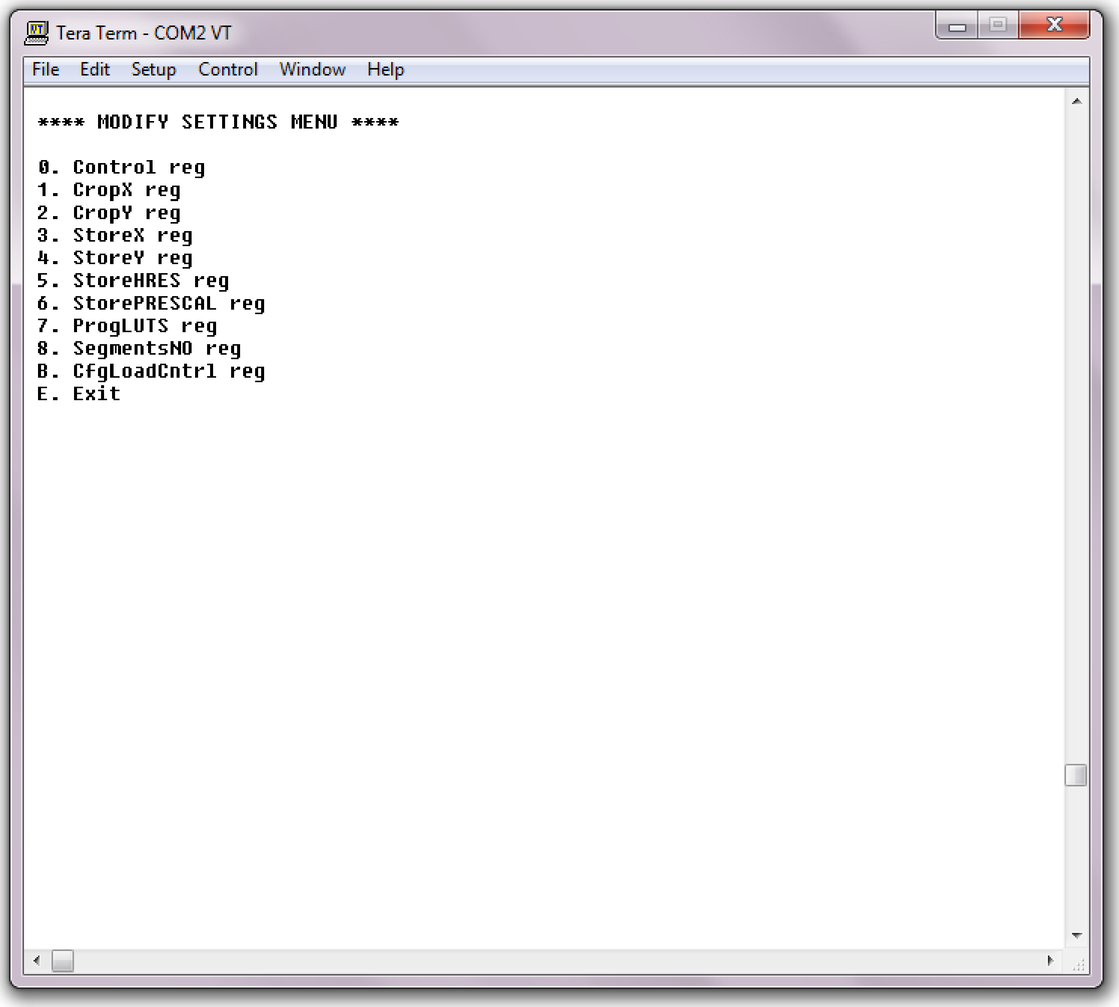


```
******* PixBLASTER MAIN MENU *******

M. Modify settings
L. List settings
S. Save settings
R. Report FPGA status
V. Report uC firmware version
```

Figure 6.3 The Pixblasters Main Configuration Menu

6.2 Adjusting Pixblasters Controller Settings

Hit **M** or **m** key to enter the “Modify settings” menu, hit **E** or **e** to return to the main menu (Figure below).



```
**** MODIFY SETTINGS MENU ****

0. Control reg
1. CropX reg
2. CropY reg
3. StoreX reg
4. StoreY reg
5. StoreHRES reg
6. StorePRESCAL reg
7. ProgLUTS reg
8. SegmentsNO reg
B. CfgLoadCntrl reg
E. Exit
```

Figure 6.4 The Sub-menu for Internal Registers’ Value Changes

IMPORTANT: New register values will be active after the exiting from the “Modify settings” menu!

7 INTERNAL REGISTERS

7.1.1 Control Register Settings

Table 7.1: Control Register Description

Bits	Name	Type	Reset value	Description
0	CONTROLLER ENABLE	R/W	0	'0' – Controller disabled '1' – Controller enabled This bit is automatically handled by the Pixblasters controller's firmware accordingly to the presence status of the valid input video signal. When changing the Control register's values, it is recommended to keep the current value of this bit.
1	MIRROR*	R/W	0	'0' – Non-mirrored image '1' – Mirrored image For more information see the chapter **Multi-Controller Topology**.
3-2	PADDING*	R/W	0	Used to set a number of LED strip pixels at the beginning of each strip that will not be used. First pixel of the video line will be sent to the LED strip pixel that follows the skipped LED strip pixels. This feature can be used in waterproof installations where strips are coming out from the sealed box. In such installations, the first strip pixels can be skipped to properly position the image on the visible display's part.
5-4	LSEL	R/W	00	Pixel color order (low byte) "00" – Blue, "01" – Green, "10" – Red, "11" – Blue
7-6	MSEL	R/W	01	Pixel color order (middle byte) "00" – Green, "01" – Red , "10" – Blue, "11" – Green
9-8	HSEL	R/W	01	Pixel color order (high byte) "00" – Red, "01" – Green, "10" – Blue, "11" – Red
15-10	RESERVED	-	-	Do not change

⚠ * IMPORTANT: The mirroring and the padding features will be supported with the firmware release planned in 2Q2020!

HSEL, MSEL and LSEL bits determine the output pixel color order. Example: In order to change pixel color order from R (high byte), G (middle byte), B (low byte) to G (high byte), R (middle byte), B (low byte) pixel data, HLSEL, MSEL and LSEL bits must be programmed as follows:
LSEL[1:0] – "00"; MSEL[1:0] - "01"; HSEL[1:0] - "01"

7.1.2 CropX Register Settings

Table 7.2: CropX Register Description

Bits	Name	Type	Reset value	Description
9-0	CROPX	R/W	0x032F	Defines the exact X coordinate of the upper left corner of the cropping video window.
15-10	RESERVED	-	-	Do not change

For more information see chapter **8.1 Video Input Cropping**.

7.1.3 CropY Register Settings

Table 7.3: CropY Register Description

Bits	Name	Type	Reset value	Description
9-0	CROPY	R/W	0x014A	Defines the exact Y coordinate of the upper left corner of the cropping video window.
15-10	RESERVED	-	-	Do not change

For more information see chapter **8.1 Video Input Cropping**.

7.1.4 StoreX Register Settings

Table 7.4: StoreX Register Description

Bits	Name	Type	Reset value	Description
9-0	STOREX	R/W	0x0194	Defines the width of the cropping video window. The width is defined as STOREX + 1.
15-10	RESERVED	-	-	Do not change

For more information see chapter **8.1 Video Input Cropping**.

7.1.5 StoreY Register Settings

Table 7.5: StoreY Register Description

Bits	Name	Type	Reset value	Description
9-0	STOREY	R/W	0x000F	Defines the height of the cropping video window. The width is defined as STOREY + 1.
15-10	RESERVED	-	-	Do not change

For more information see chapter **8.1 Video Input Cropping**.

7.1.6 StoreHRES Register Settings

Table 7.6: StoreHRES Register Description

Bits	Name	Type	Reset value	Description
9-0	STOREHRES	R/W	0x0194	Defines the width of the LED display. The width is defined as HRES + 1.
15-10	RESERVED	-	-	Do not change

7.1.7 StorePRESCAL Register Settings

Defines the clock signal used to control the LED display. For more information see chapter **8.2 LED Display Timing**.

Table 7.7: StorePRESCAL Register Description

Bits	Name	Type	Reset value	Description
9-0	STOREPRESCAL	R/W	0x0000	Defines the T_{SLOT} period which is equal to the T_{STRIP}*(PRESCAL_REG + 1).
15-10	RESERVED	-	-	Do not change

7.1.8 ProgLUTS Register Settings

Used for bit encoders LUT programming. For more information see chapter **8.3 LED Pixel Bit Value Encoding**.

Table 7.8: ProgLUTS Register Description

Bits	Name	Type	Reset value	Description
0	LUT_DATA	R/W	0x0000	Data to be written into the Look-Up Tables (LUT) implemented in FPGA chip memories.
1	LUT_CLK	R/W		CLK signal for the LUT. Must respect timing requirements
2	LUT_ZERO_EN	R/W		Enable signal for the LUT encoding 0 bit value
3	LUT_UNO_EN	R/W		Enable signal for the LUT encoding 1 bit value
8-4	LUT_ADR(4:0)	R/W		5 bit address for 32x1 LUTs
15-10	RESERVED	-	-	Do not change

7.1.9 SegmentsNO Register Settings

Table 7.9: SEGMENTS Register Description

Bits	Name	Type	Reset value	Description
9-0	SEGMENTS	R/W	0x0000	Defines the number of output display lines driven by a single LED output. This number must comply with the following constraint: StoreHRES_REG * SEGMENTS_REG <= 512. Number of LED strip segments (display lines) = SEGMENTS + 1. For more information see chapter **Segmented Single Controller Topology**.
15-10	RESERVED	-	-	Do not change

7.1.10CfgLoadCntrl Register Settings

Table 7.10: CfgLoadCntrl Register Description

Bits	Name	Type	Reset value	Description
0	LOADSRCSEL	R/W	0	'0' – FPGA configuration to load is defined by position of the "Configuration select" switch '1' - FPGA configuration to load is defined by register bits 3-1.
3-1	CFGSELECT	R/W	001	"000" – Update configuration "001" – User configuration 1 "010" – User configuration 2 "011" – User configuration 3 "100" – User configuration 4 "101" – User configuration 5 "110" – User configuration 6 "111" – User configuration 7
15-4	RESERVED	-	-	Do not change

7.1.11Aux1 Register Settings

Table 7.11: Aux1 Register Description

Bits	Name	Type	Reset value	Description
9-0	Aux1	R/W	0x0000	Currently not used. Reserved for future expansion.
15-10	RESERVED	-	-	Do not change

7.1.12Aux2 Register Settings

Table 7.12: Aux2 Register Description

Bits	Name	Type	Reset value	Description
9-0	Aux2	R/W	0x0000	Currently not used. Reserved for future expansion.
15-10	RESERVED	-	-	Do not change

8 OPERATIONAL DETAILS

8.1 Video Input Cropping

Cropping window position and sizing are programmable through the following registers: X_CROP, Y_CROP, X_STORE and Y_STORE. The following figure explains how to define the video cropping window.

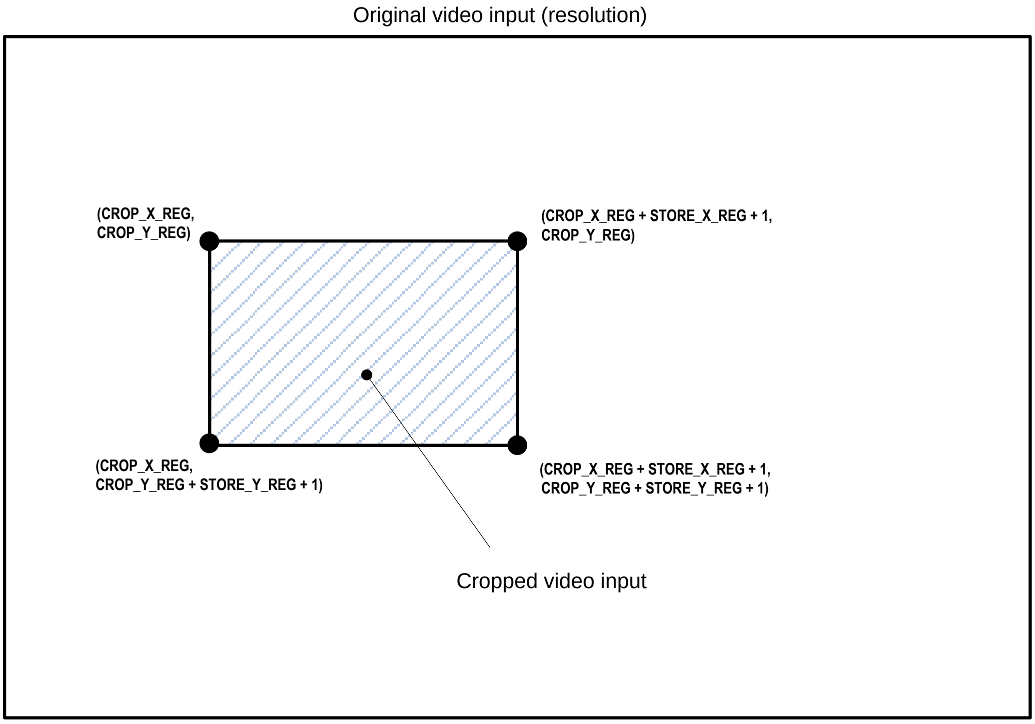

Original video input (resolution)
(CROP_X_REG, CROP_Y_REG)
(CROP_X_REG + STORE_X_REG + 1, CROP_Y_REG)
(CROP_X_REG, CROP_Y_REG + STORE_Y_REG + 1)
(CROP_X_REG + STORE_X_REG + 1, CROP_Y_REG + STORE_Y_REG + 1)
Cropped video input

Figure 8.1 Definition of the Cropping Window Programming

Select **M. Modify Settings** (Figure 6.4) and setup the Cropping window position and sizing through the following registers: X_CROP, Y_CROP, X_STORE and Y_STORE. Once programmed, the configuration data will be permanently stored in the PIC18F26J60 microcontroller's internal flash memory, and at the next power up, the LED display will show the programmed image window.

8.2 LED Display Timing

Figure 8.2 explains relationship between the Pixblasters controller's internal clock (f_{STRIP}), and the timing at the LED control outputs. f_{STRIP} is fixed to 25.6MHz. StorePRESCAL register should be set such that correct T_{SLOT} interval for the specific LED strip type is generated (T_{SLOT} = T_{STRIP}*(PRESCAL_REG + 1) where T_{STRIP} = 1/f_{STRIP}).

The NEW_PIXEL is fixed to 24 T_{SLOT} periods. This setup enables serial shift of the RGB888 or similar pixel formats to correctly drive the LED pixel.

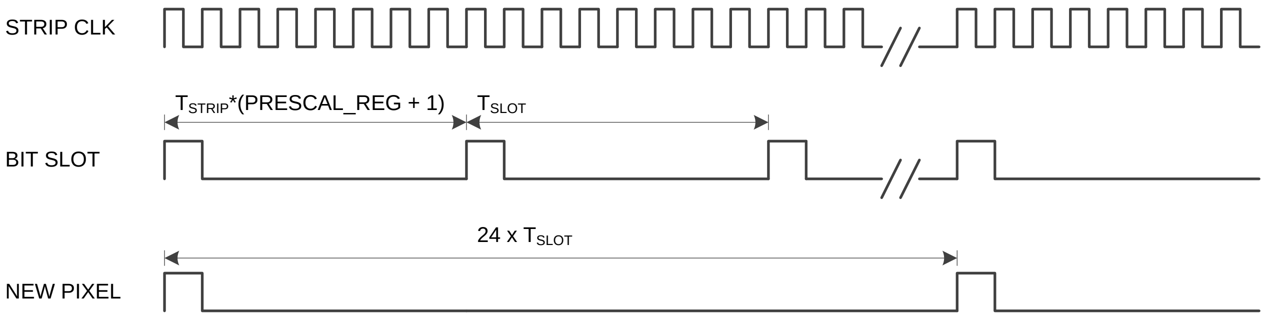


Figure 8.2 LED display clocking

8.3 LED Pixel Bit Value Encoding

The Pixblasters MS1 controller currently supports only Worldsemi compatible 1-wire drivers that require bit encoding presented in the **Figure 8.3**.

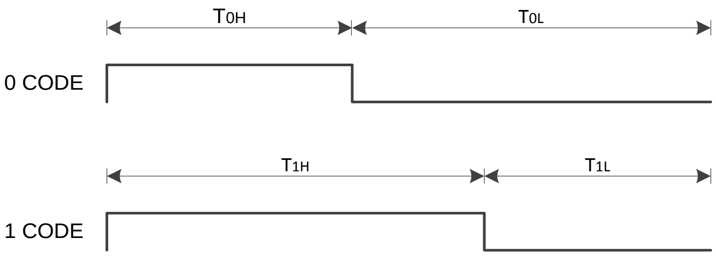


Figure 8.3 Control 0 and 1 data encoding

In order to achieve appropriate bit value encoding user has to set up "zero" and "one" codes by writing into LUTS_REG. Each code (zero and one) is consisted of 32 samples (S0 – S31) where each sample has duration of T_{STRIP} = 1/f_{STRIP}..

LUT_ADR bits define sample, number, LUT_ZERO_EN and LUT_ONE_EN define to which code (zero or one) sample value will be assigned to, LUT_DATA is the actual sample value that is being written, and LUT_CLK is "Write strobe" flag.

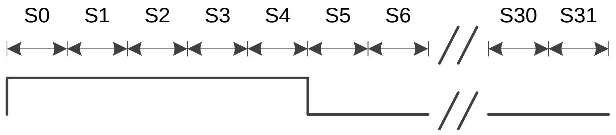


Figure 8.4 Control 0 and 1 data encoding can be adjusted

If, for example, bit encoding has to look like the one shown in **Figure 8.4**, and if that encoding defines code for the "zero" bit, than user must perform the following sequence of writes into LUTS_REG.

Table 8.1: SEGMENTS Register Description

WR NUM	LUT_ADR	LUT_ONE_EN	LUT_ZERO_EN	LUT_CLK	LUT_DATA
1	0	0	1	1	1
2	0	0	1	0	1
3	1	0	1	1	1
4	1	0	1	0	1
5	2	0	1	1	1
6	2	0	1	0	1
7	3	0	1	1	1
8	3	0	1	0	1
9	4	0	1	1	1
10	4	0	1	0	1
11	5	0	1	1	0
12	5	0	1	0	0
13	6	0	1	1	0
14	6	0	1	0	0
61	30	0	1	1	0
62	30	0	1	0	0
63	31	0	1	1	0
64	31	0	1	0	0

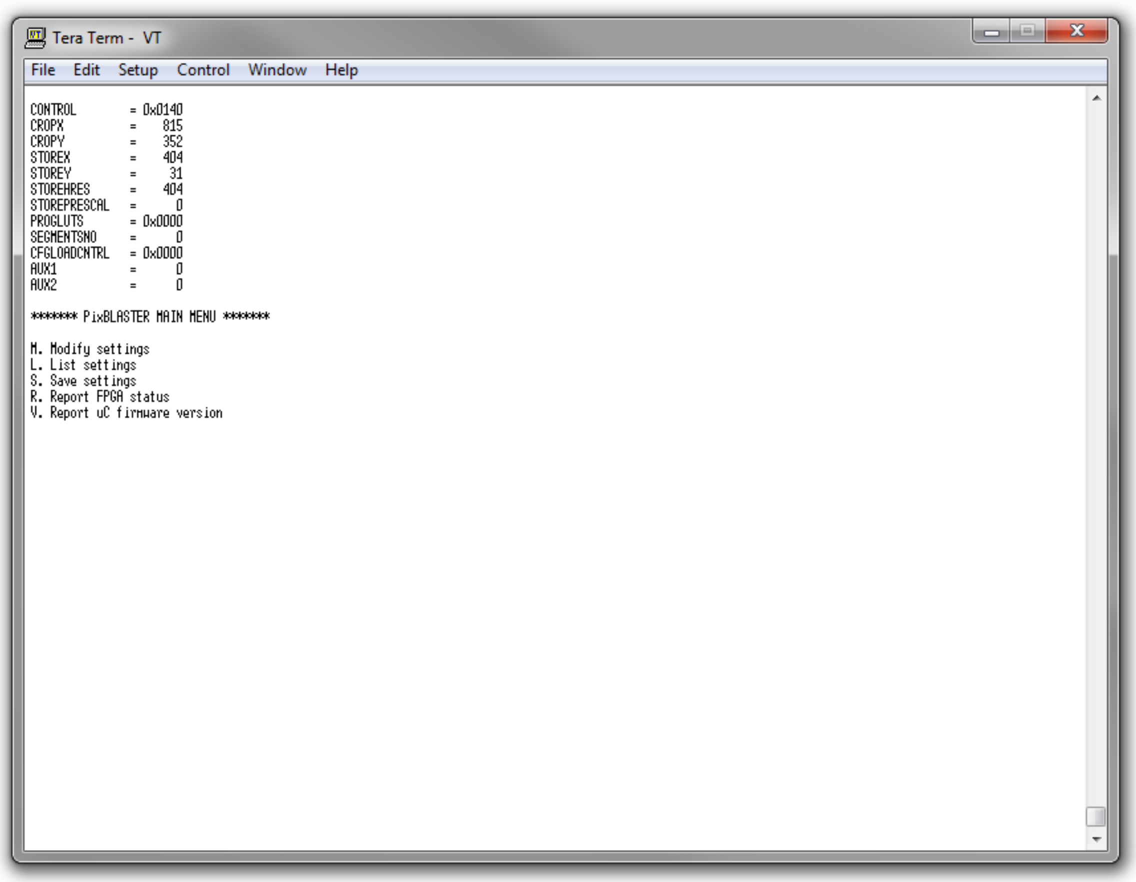
If the same encoding had belong to the "one" bit, than the following sequence should have been used

Table 8.2: SEGMENTS Register Description

WR NUM	LUT_ADR	LUT_ONE_EN	LUT_ZERO_EN	LUT_CLK	LUT_DATA
1	0	1	0	1	1
2	0	1	0	0	1
3	1	1	0	1	1
4	1	1	0	0	1
5	2	1	0	1	1
6	2	1	0	0	1
7	3	1	0	1	1
8	3	1	0	0	1
9	4	1	0	1	1
10	4	1	0	0	1
11	5	1	0	1	0
12	5	1	0	0	0
13	6	1	0	1	0
14	6	1	0	0	0
61	30	1	0	1	0
62	30	1	0	0	0
63	31	1	0	1	0
64	31	1	0	0	0

8.4 Listing Pixblasters Controller Settings

To check configured controller’s settings, hit **L** or **l** key and the following should appear:



```
CONTROL       = 0x0140
CROPX         =    815
CROPY         =    352
STOREX        =    404
STOREY        =     31
STOREHRES     =    404
STOREPRESCAL  =      0
PROGLUTS      = 0x0000
SEGMENTSNO    =      0
CFGLOADCNTRL  = 0x0000
AUX1          =      0
AUX2          =      0

******* PixBLASTER MAIN MENU *******

M. Modify settings
L. List settings
S. Save settings
R. Report FPGA status
V. Report uC firmware version
```

Figure 8.5 Check-list of Programmed Internal Registers’ Values

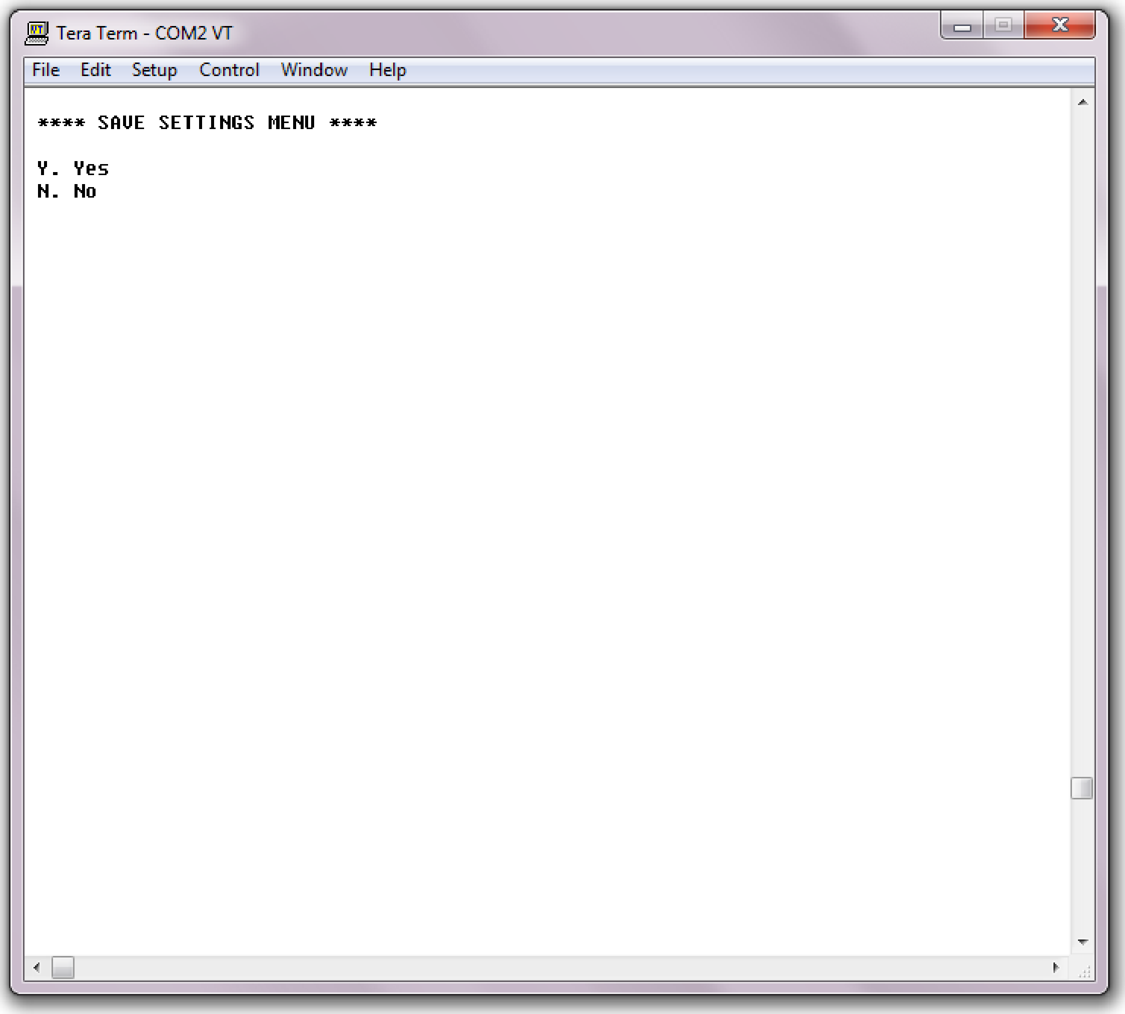
Please note that the actual values may be different from those shown in **Figure 8.5** since they reflect a specific controller’s configuration.

CONTROL	0x00
CROPX	0x00
CROPY	0x00
STOREX	0x00
STOREY	0x00
STOREHRES	0x00
STOREPRESCAL	0x00
PROGLUTS	0x00
SEGMENTSNO	0x00
CFGLOADCTRL	0x00
AUX1, AUX2	Reserved

Table 8.3: Default (Pre-Programmed) Registers Values

8.5 Saving Pixblasters Controller Settings

"Save settings" menu is accessed by **S** or **s** key.



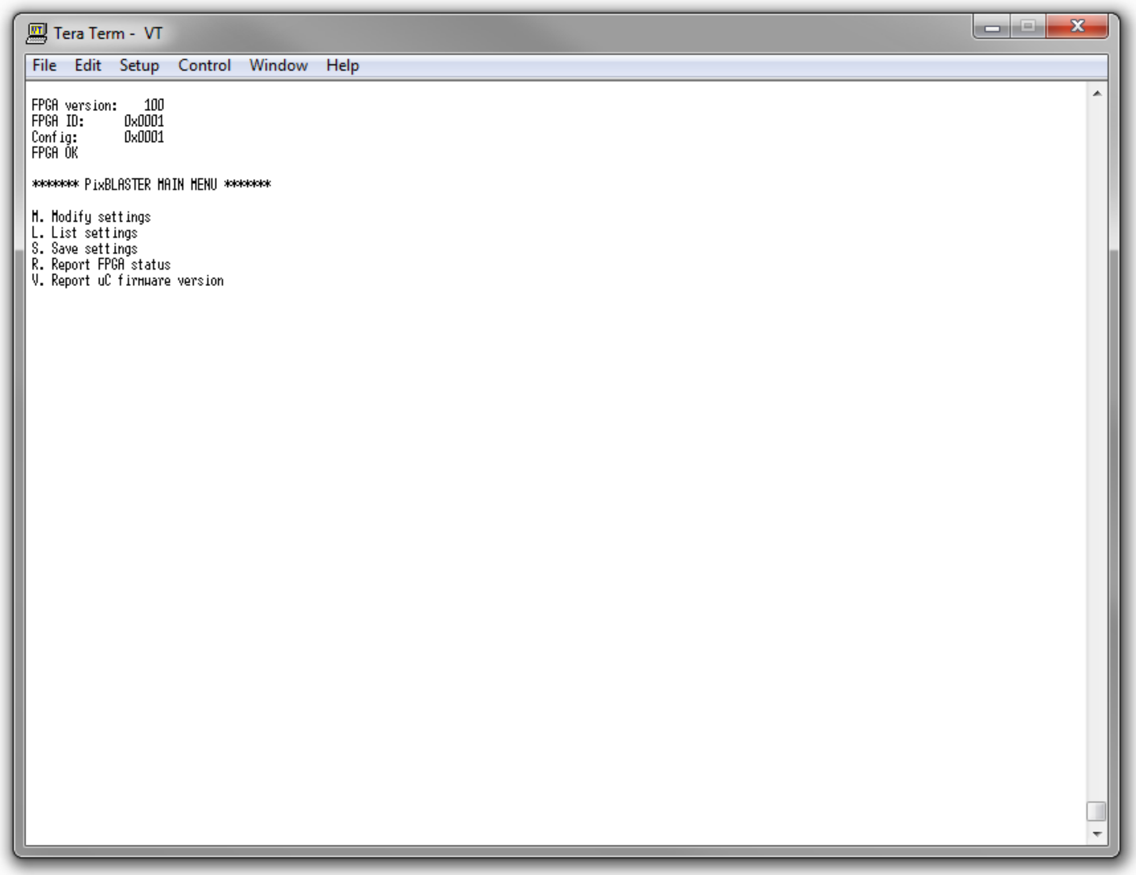
```
**** SAVE SETTINGS MENU ****

Y. Yes
N. No
```

Figure 8.6 Permanently Save the Settings Into Pixblasters MS1 Controller

Hit **Y** or **y** to save settings, or **N** or **n** to exit.

8.6 Report FPGA Status

This menu is used for checking the FPGA status. Hit **R** or **r** to access the FPGA status. The following log shall appear:



```
FPGA version:    100
FPGA ID:      0x0001
Config:       0x0001
FPGA OK

******* PixBLASTER MAIN MENU *******

M. Modify settings
L. List settings
S. Save settings
R. Report FPGA status
V. Report uC firmware version
```

Figure 8.7 Check the FPGA Status

8.7 Updating FPGA Configuration

In order to be able to perform FPGA configuration update, the FPGA configuration '0' should be loaded first. That is accomplished by setting the **CfgLoadCntrl** bits to appropriate values (0x01 for firmware configuration load control, or 0x00 for controlling configuration load by using FPGA configuration select switch. Figure 8.8 shows the DIP switches setup to 0x00 and explains the applied configuration number coding.

If the firmware load control is selected, new configuration will be loaded immediately upon the **CfgLoadCntrl** register update.

⚠ New register value will become active upon exiting "Modify settings" menu!

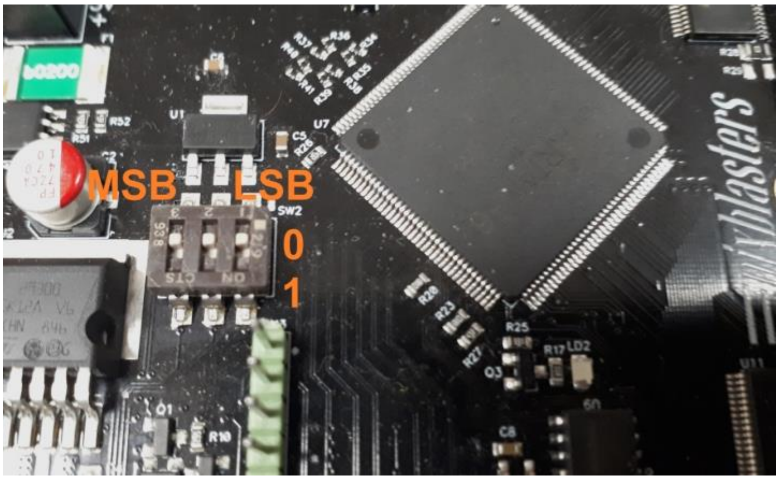


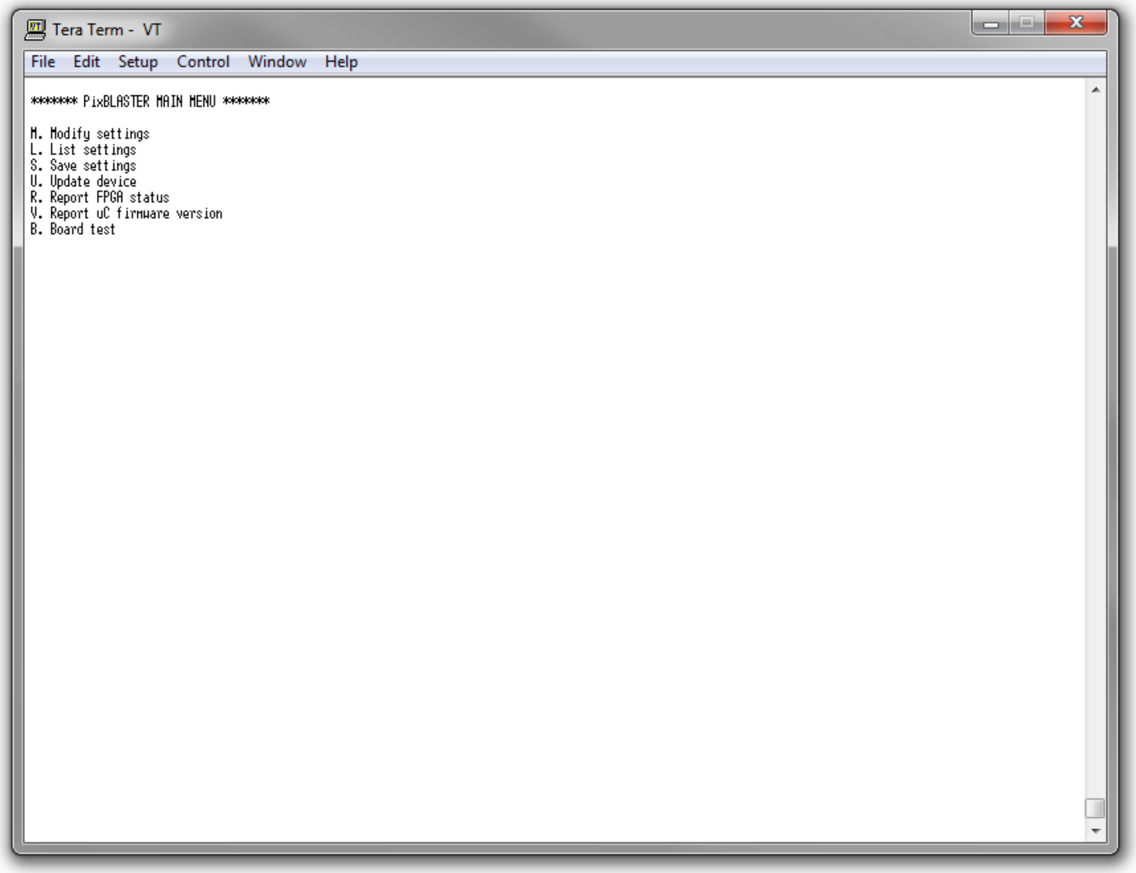
Figure 8.8 DIP switch – FPGA Configuration Selector

Table 8.4: Selecting FPGA Configuration by DIP Switches

Config. No.	DIPs [MSB, b1, LSB]	Description
Config. 0	000	Load configuration that enables FPGA configurations change
Config. 1	001	Master Pixblasters MS1 configuration with 16 active LED strip outputs. Connect the video input to HDMI connector.
Config. 2	010	Slave Pixblasters MS1 configuration with 16 active LED strip outputs. Connect the video input to HIS RJ45 connector.
Config. 3	011	Master Pixblasters MS1 configuration with 32 active LED strip outputs. Connect the video input to HDMI connector.
Config. 4	100	Slave Pixblasters MS1 configuration with 32 active LED strip outputs. Connect the video input to HIS RJ45 connector.
Config. 5	101	Reserved.
Config. 6	110	Reserved.
Config. 7	111	Reserved.

The latest versions of the User's Manual and FPGA configuration files can be found at: www.pixblasters.com/downloads/deliverables.

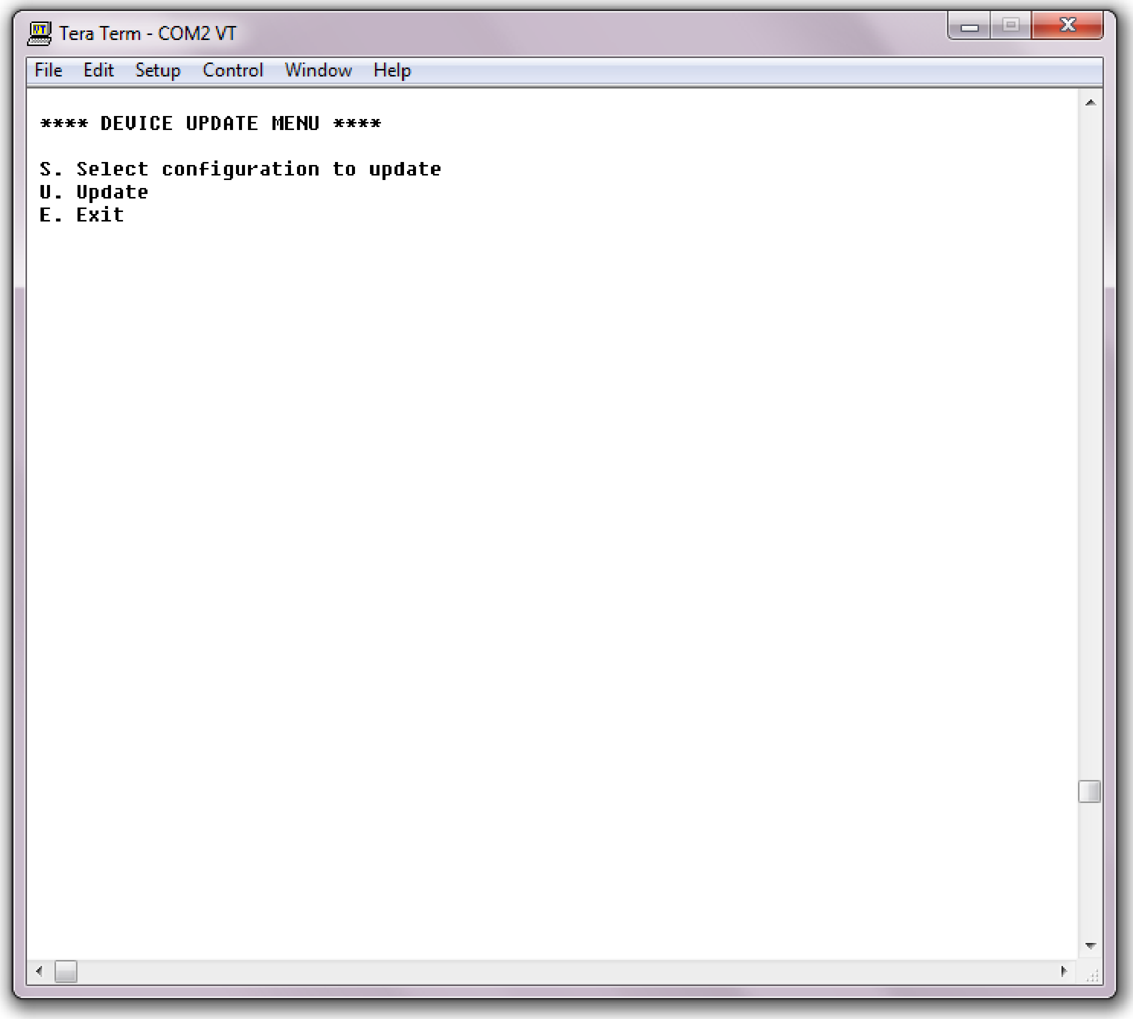
When the configuration '0' is loaded, the following screen should appear:



```
****** PixBLASTER MAIN MENU ******

M. Modify settings
L. List settings
S. Save settings
U. Update device
R. Report FPGA status
V. Report uC firmware version
B. Board test
```

Figure 8.9 Prepare for the FPGA Configuration Update

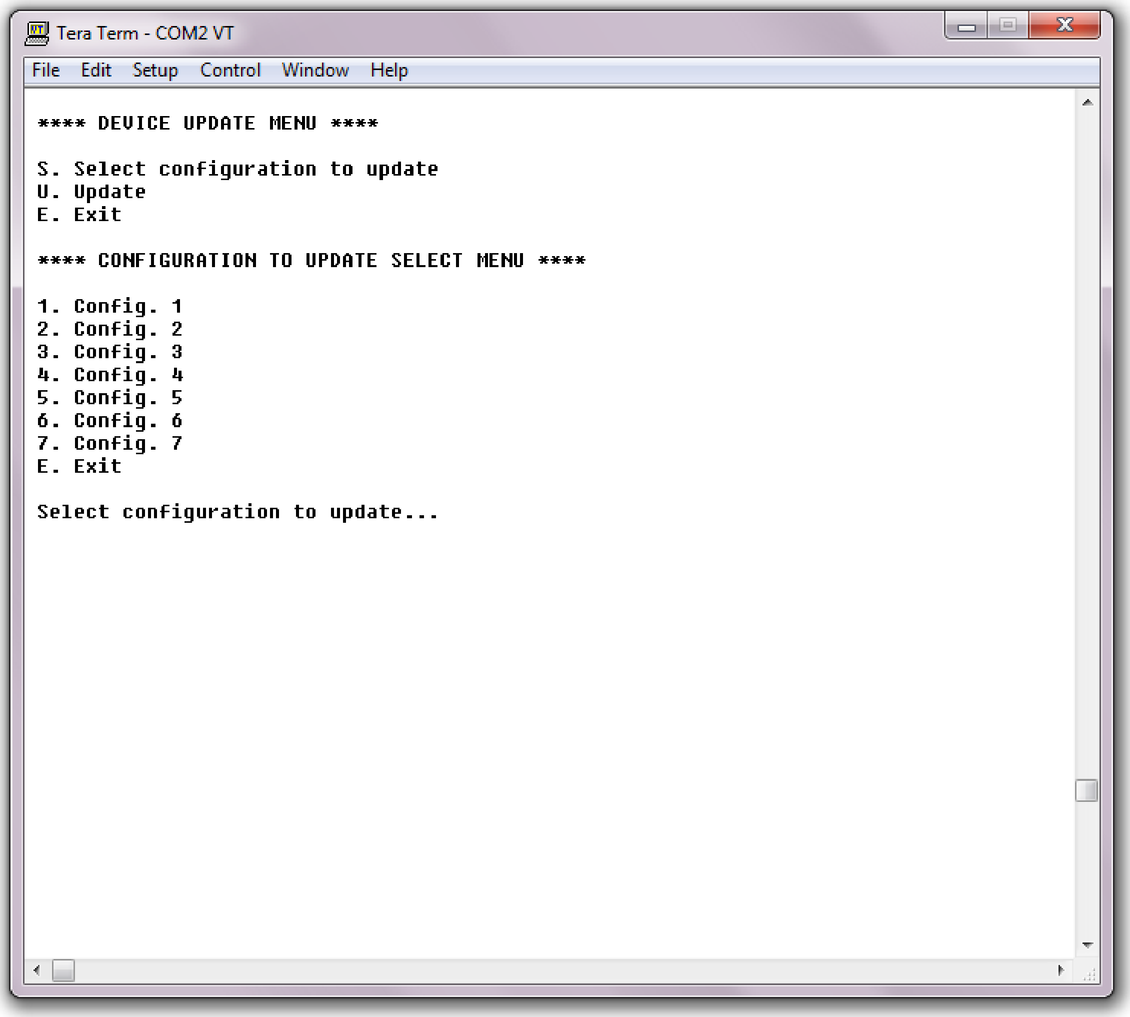
Hit **U** or **u** key to enter the "Update device" menu. The following shall appear:



```
**** DEVICE UPDATE MENU ****

S. Select configuration to update
U. Update
E. Exit
```

Figure 8.10 The FPGA Configuration Update Menu

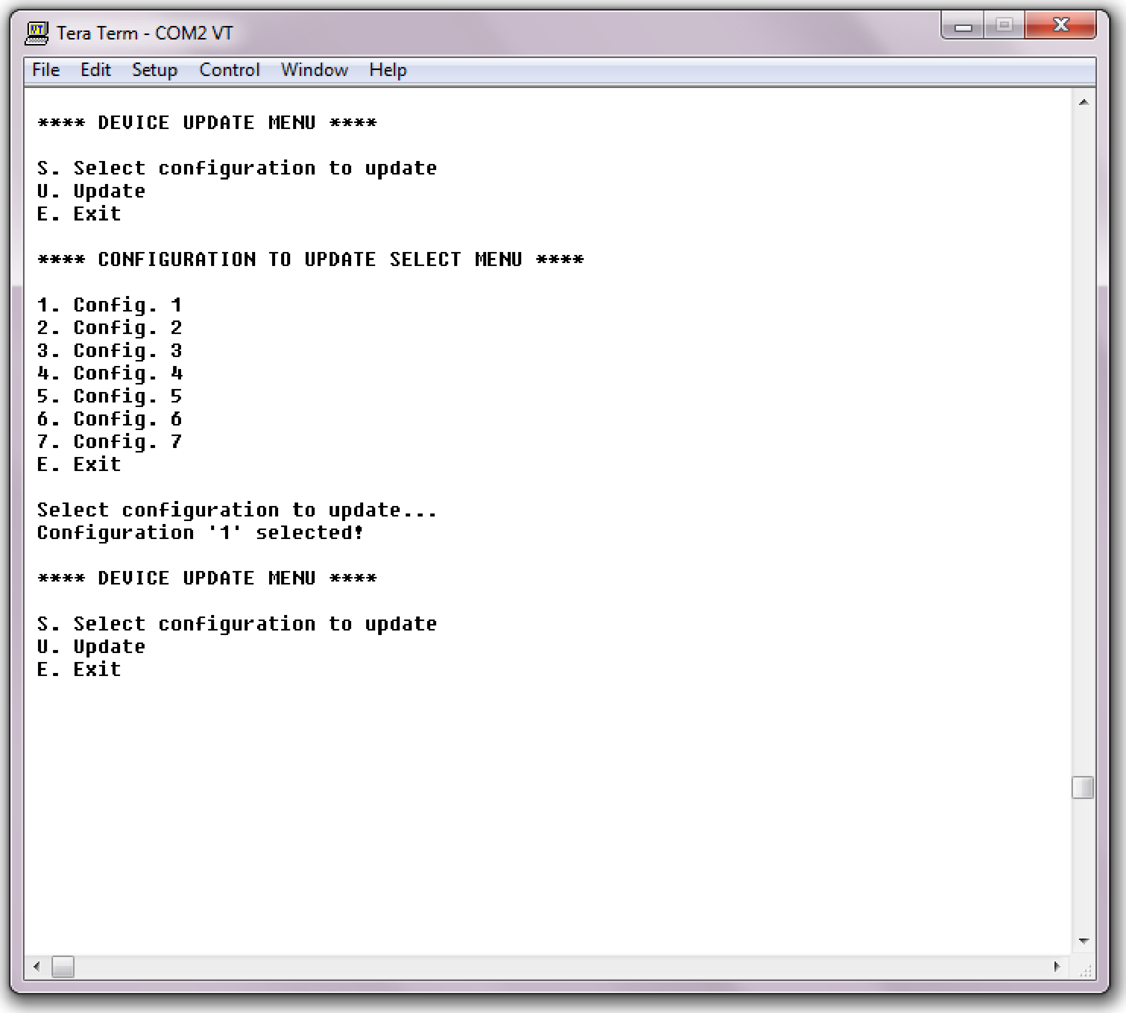
Hit **S** or **s** to select configuration you wish to update. The following will appear:



```
**** DEVICE UPDATE MENU ****

S. Select configuration to update
U. Update
E. Exit

**** CONFIGURATION TO UPDATE SELECT MENU ****

1. Config. 1
2. Config. 2
3. Config. 3
4. Config. 4
5. Config. 5
6. Config. 6
7. Config. 7
E. Exit

Select configuration to update...
```

Figure 8.11 Select the FPGA Configuration for Update

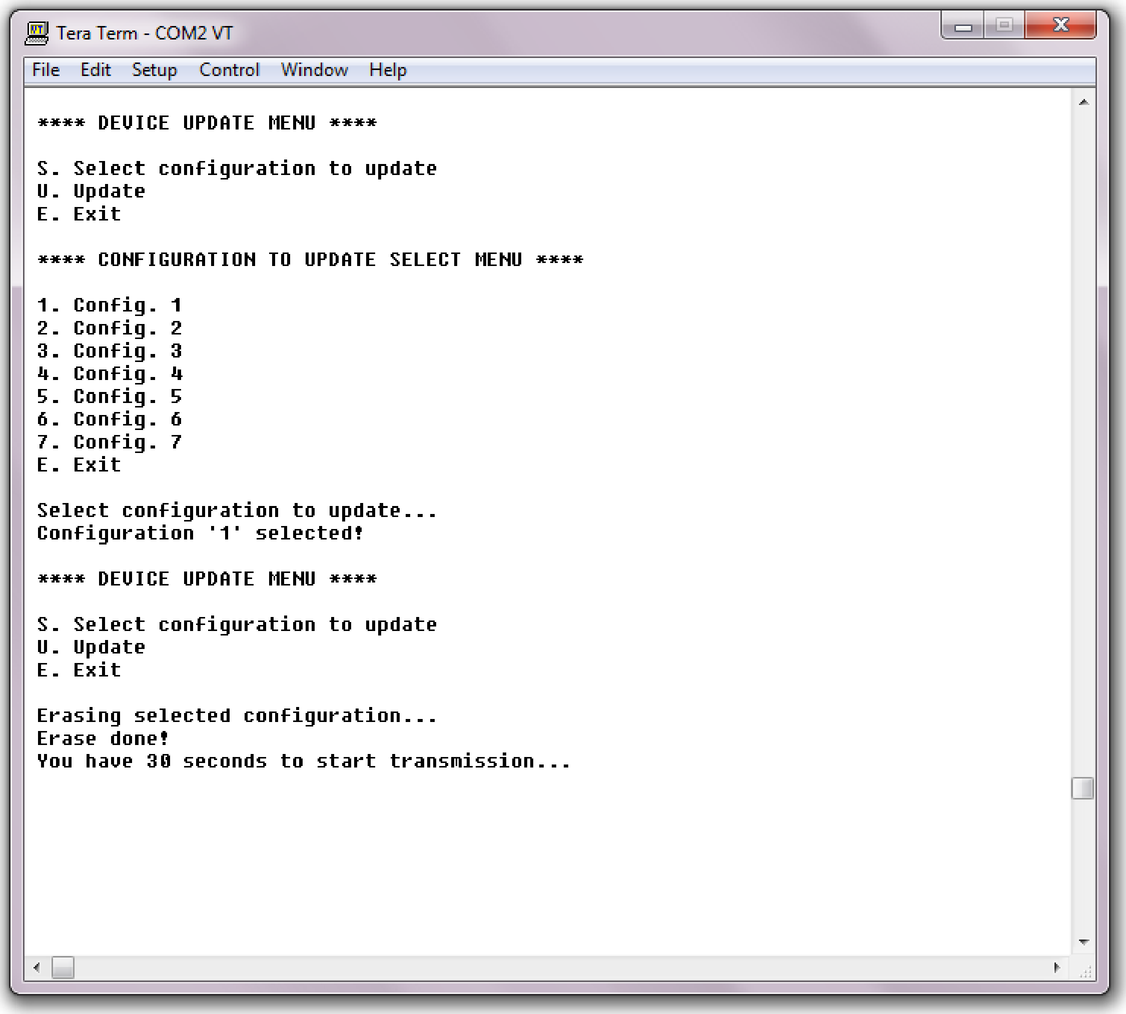
Hit a number corresponding to a configuration you wish to update. If, for example, you select configuration ‘1’ you should see the following:



```
**** DEVICE UPDATE MENU ****

S. Select configuration to update
U. Update
E. Exit

**** CONFIGURATION TO UPDATE SELECT MENU ****

1. Config. 1
2. Config. 2
3. Config. 3
4. Config. 4
5. Config. 5
6. Config. 6
7. Config. 7
E. Exit

Select configuration to update...
Configuration '1' selected!

**** DEVICE UPDATE MENU ****

S. Select configuration to update
U. Update
E. Exit
```

Figure 8.12 The FPGA Configuration for Update Selected

Hit **U** or **u** to start update procedure. The following shall appear:

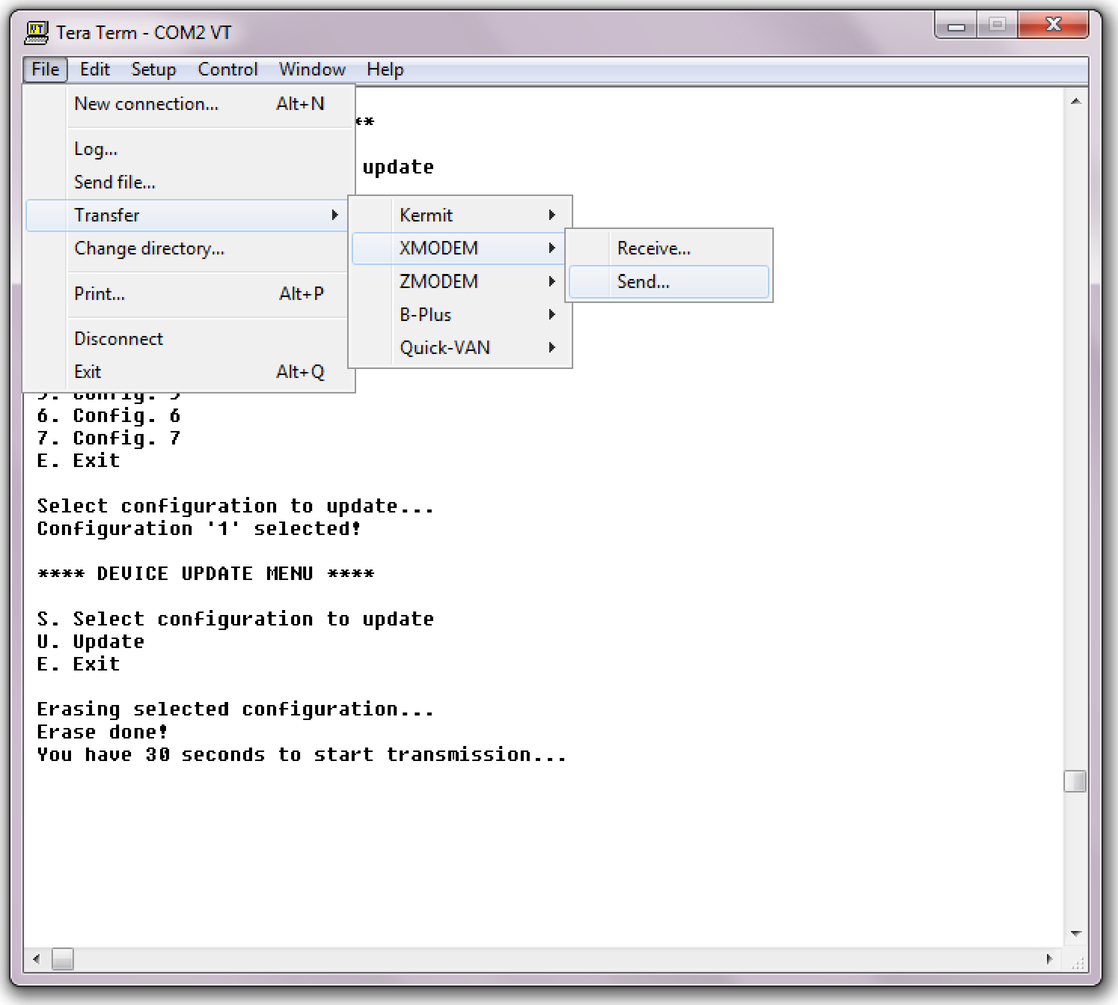


```
**** DEVICE UPDATE MENU ****

S. Select configuration to update
U. Update
E. Exit

**** CONFIGURATION TO UPDATE SELECT MENU ****

1. Config. 1
2. Config. 2
3. Config. 3
4. Config. 4
5. Config. 5
6. Config. 6
7. Config. 7
E. Exit

Select configuration to update...
Configuration '1' selected!

**** DEVICE UPDATE MENU ****

S. Select configuration to update
U. Update
E. Exit

Erasing selected configuration...
Erase done!
You have 30 seconds to start transmission...
```

Figure 8.13 Erasing the FPGA Configuration for the Update

This message means that selected configuration has been erased, and that you have 30 seconds to select and start transmitting a file containing a new configuration.

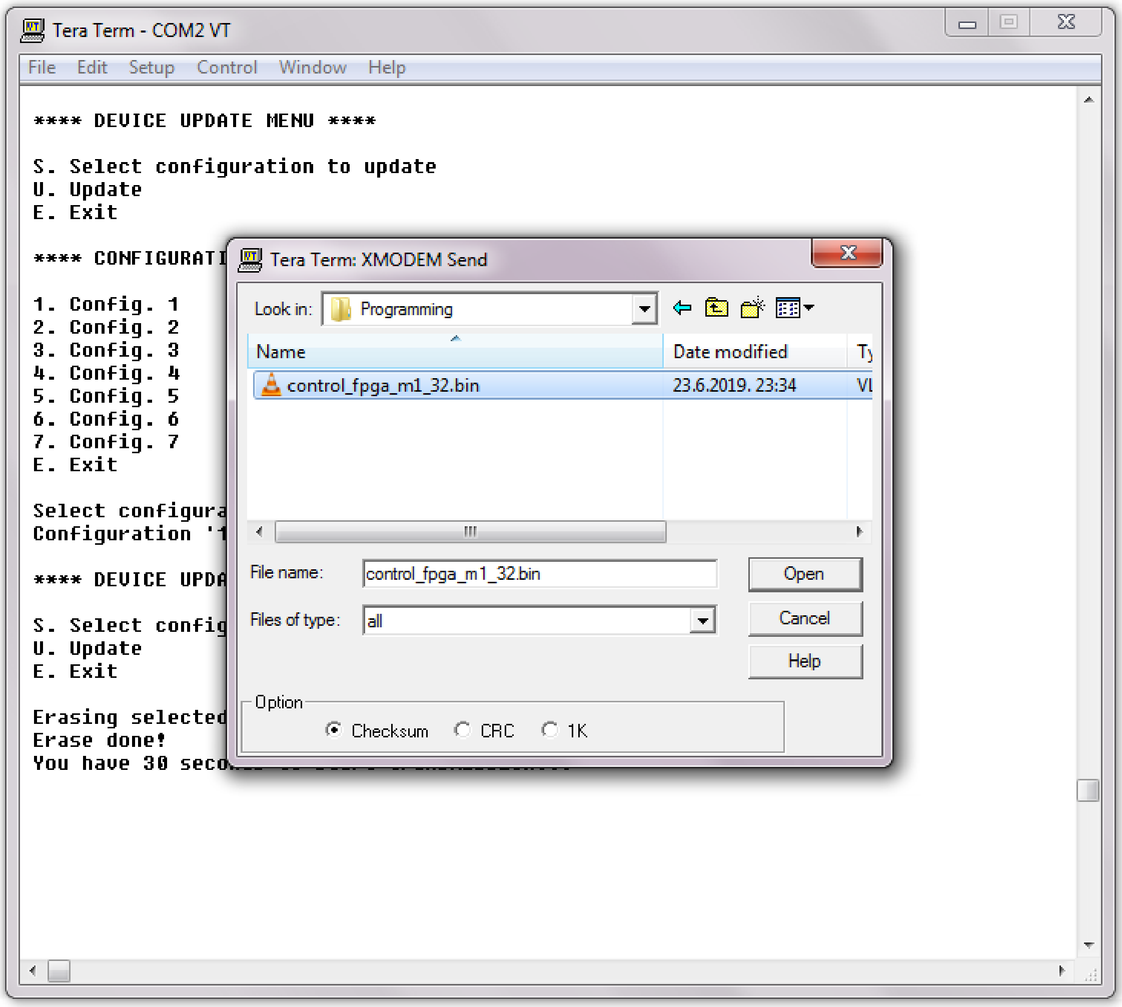
File transmission is accomplished through the XMODEM protocol. If the Tera Term emulator is used, the XMODEM transmission is selected in the following way:



```
                                     **

                                     update

5. Config. 5
6. Config. 6
7. Config. 7
E. Exit

Select configuration to update...
Configuration '1' selected!

**** DEVICE UPDATE MENU ****

S. Select configuration to update
U. Update
E. Exit

Erasing selected configuration...
Erase done!
You have 30 seconds to start transmission...
```

Figure 8.14 Setup the XMODEM Protocol in the Tera Term Serial Emulator

Then select the update .bin FPGA configuration file.



```
**** DEVICE UPDATE MENU ****

S. Select configuration to update
U. Update
E. Exit

**** CONFIGURATI

1. Config. 1
2. Config. 2
3. Config. 3
4. Config. 4
5. Config. 5
6. Config. 6
7. Config. 7
E. Exit

Select configura
Configuration '1

**** DEVICE UPDA

S. Select config
U. Update
E. Exit

Erasing selected
Erase done!
You have 30 seco
```

Look in: Programming

Name	Date modified	Ty
control_fpga_m1_32.bin	23.6.2019. 23:34	VL

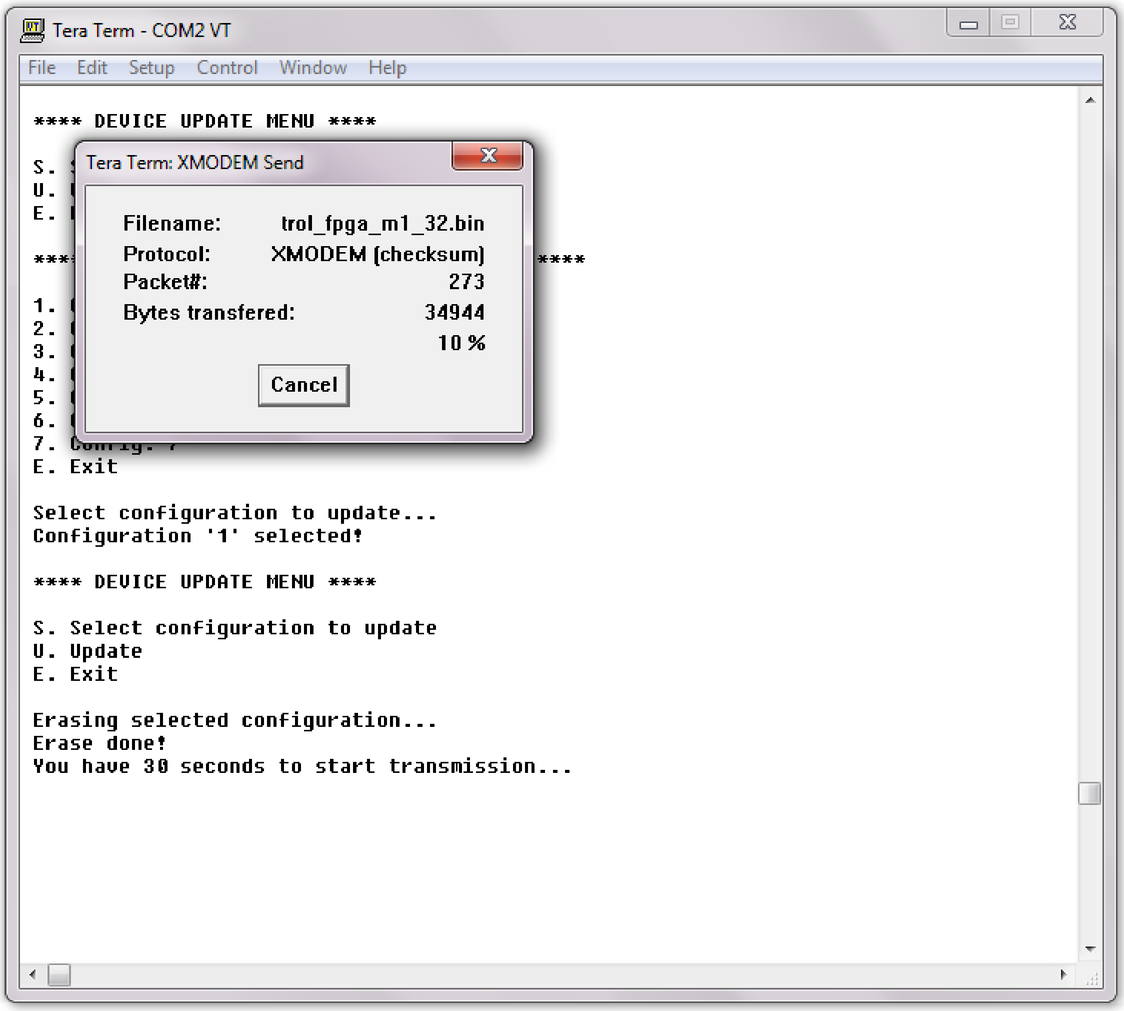
File name: control_fpga_m1_32.bin

Files of type: all

Option ● Checksum ○ CRC ○ 1K

Figure 8.15 New FPGA Configuration Update File Selected

If selection was done within 30 seconds, the file update will start.

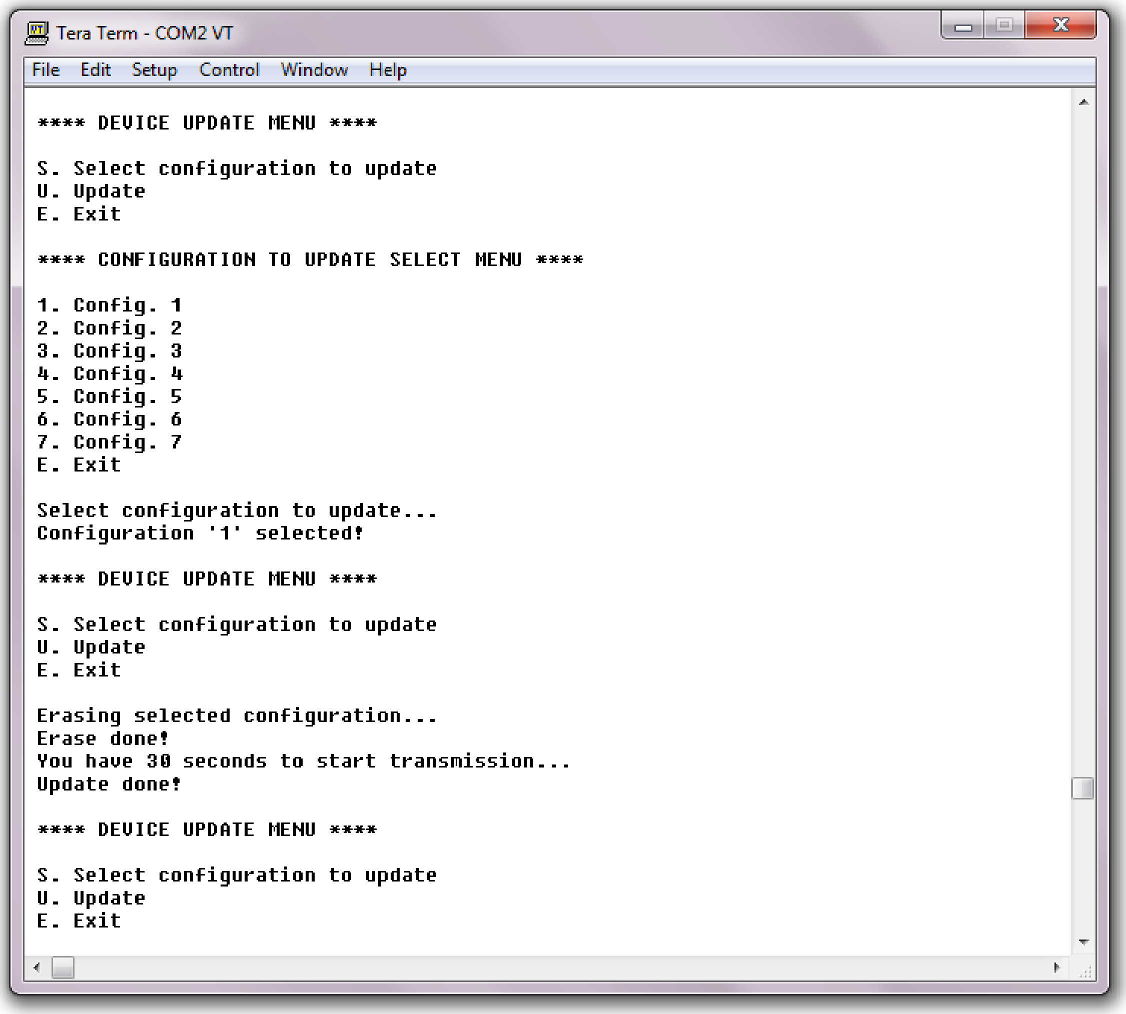


Filename: trol_fpga_m1_32.bin
Protocol: XMODEM (checksum)
Packet#: 273
Bytes transfered: 34944
10 %

```
**** DEVICE UPDATE MENU ****

S. S
U. U
E. E

***                                              ****

1. (
2. (
3. (
4. (
5. (
6. (
7. Config. 7
E. Exit

Select configuration to update...
Configuration '1' selected!

**** DEVICE UPDATE MENU ****

S. Select configuration to update
U. Update
E. Exit

Erasing selected configuration...
Erase done!
You have 30 seconds to start transmission...
```

Figure 8.16 The FPGA Update in Progress

If the update was successful, the following log text will appear:



```
**** DEVICE UPDATE MENU ****

S. Select configuration to update
U. Update
E. Exit

**** CONFIGURATION TO UPDATE SELECT MENU ****

1. Config. 1
2. Config. 2
3. Config. 3
4. Config. 4
5. Config. 5
6. Config. 6
7. Config. 7
E. Exit

Select configuration to update...
Configuration '1' selected!

**** DEVICE UPDATE MENU ****

S. Select configuration to update
U. Update
E. Exit

Erasing selected configuration...
Erase done!
You have 30 seconds to start transmission...
Update done!

**** DEVICE UPDATE MENU ****

S. Select configuration to update
U. Update
E. Exit
```

Figure 8.17 The FPGA Configuration Update Successful

9 ARRAY-S LED HUB BOARD

9.1 Array-S Board Specifications

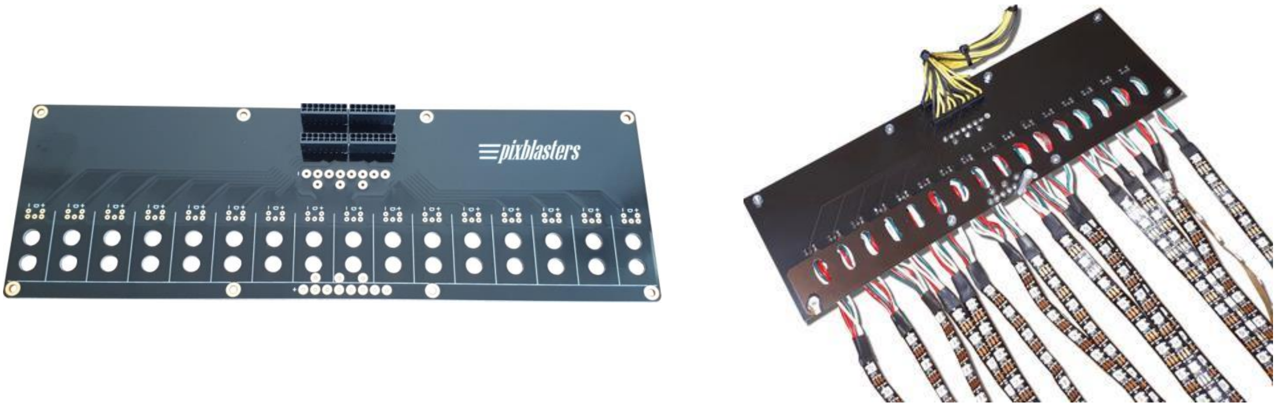


Figure 9.1 The Array-S Auxiliary LED HUB Board

The Array-S board enables releasable connections with the Pixblasters MS1 LED controller and easy building of the LED matrix. It distributes the video control signals and the power sourced from an external +5 VDC power supply. The board includes holes for easy wall mounting. Each controller supports up to two Array-S Boards for a total of 32 strips per controller.

> The Array-S board set is helpful, but not necessary for using the Pixblasters MS1 LED controller. Figure 9.2 shows the display with different wiring terminals used.

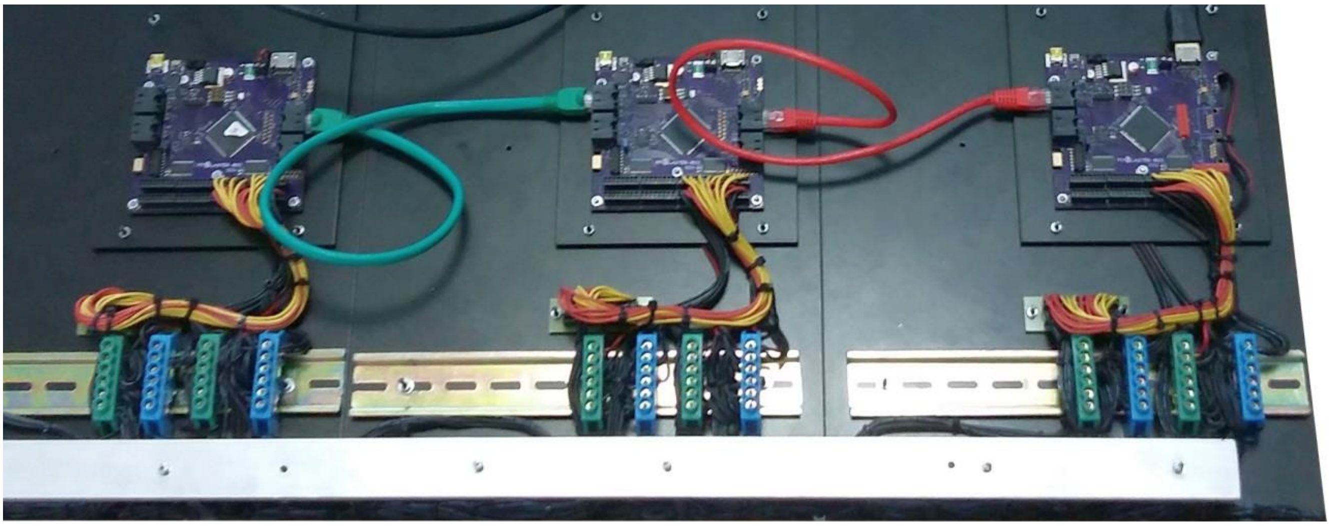
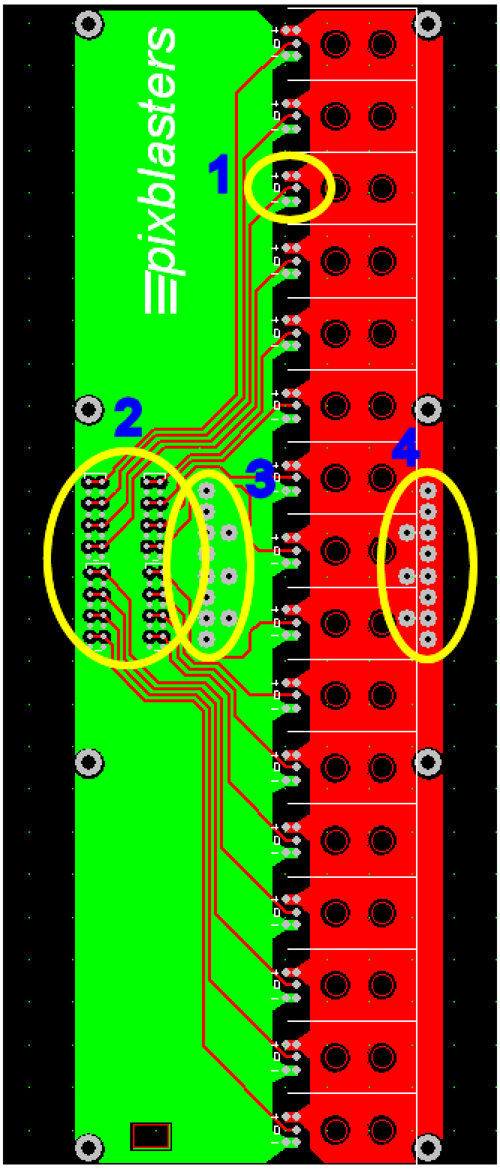
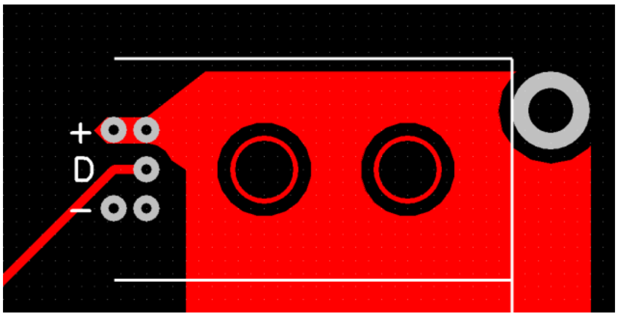


Figure 9.2 Example Wiring (Terminals) in the Multi-Controller Display Topology

LED HUB board is an optional component which has been designed to help users building their custom LED displays. It is a PCB that helps connecting LED strips, power supply, and Pixblasters controller together. Each LED HUB supports up to 16 LED strips.



1 – LED strip port



‘+’ - LED strip power
’D ’ – LED strip data
’-’ - LED strip ground

2 - Pixblasters controller port

Placeholder for connectors used for connecting to Pixblasters controller LED strip ports. Total of 4 Phoenix-1771017 connectors are supported.

3 – GND port

TH pads for connecting GND wires from power supply module. One pad may be used for connecting GND to Pixblasters controller.

4 – POWER port

The pads for connecting POWER wires from power supply module. One pad may be used for connecting POWER to Pixblasters controller.

Figure 9.3 Array-S LED HUB Board’s Layout

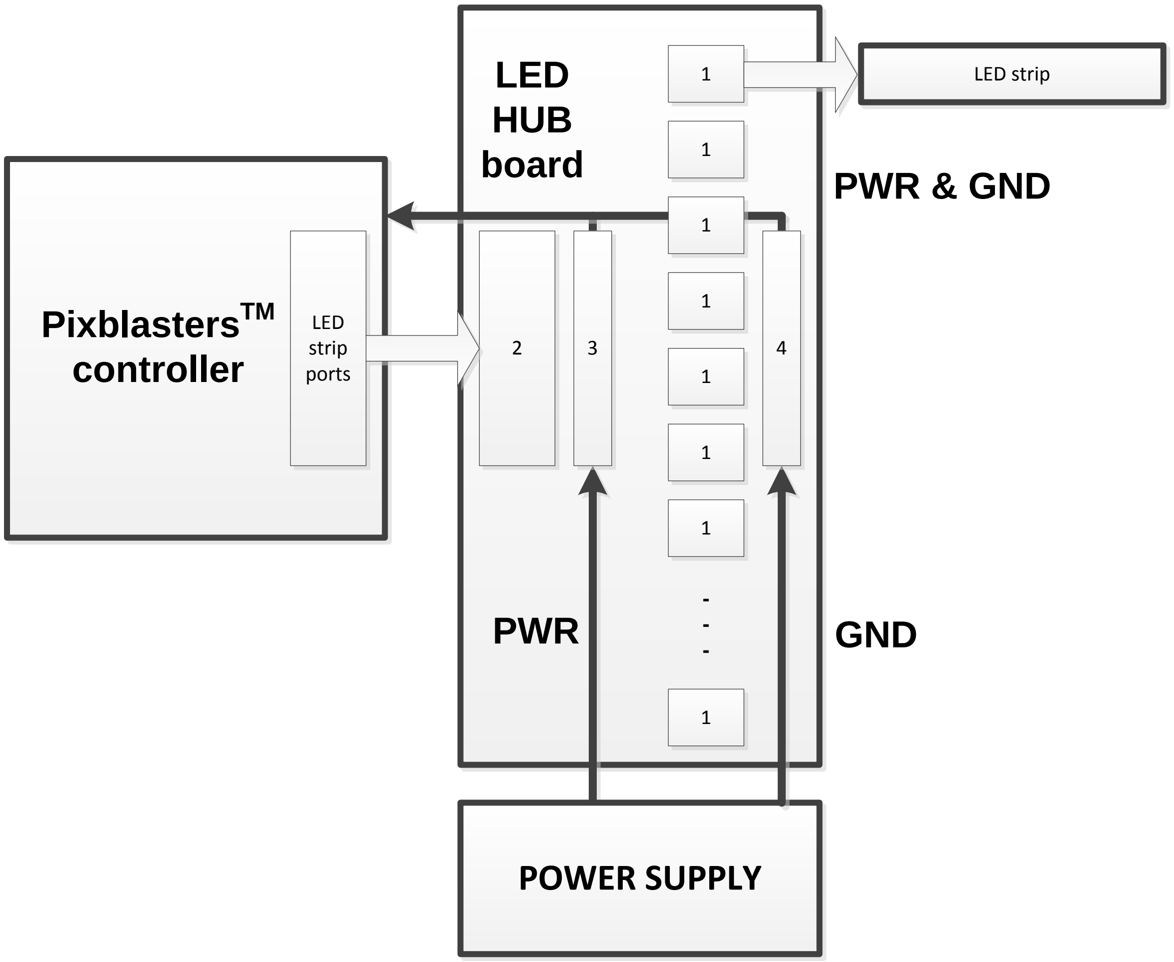


Figure 9.4 LED HUB Usage

10 EXAMPLES

10.1 Single Controller Topology, 32 LED Strips, Res - 256x64

This example shows how to configure the Pixblasters MS1 LED controller operating in the "Single controller topology", with 32 LED strips connected and with a supported resolution of 256 x 64. Since Vertical resolution exceeds the maximum number of LED strips that may be connected to a single controller, the LED strips need to be segmented. Each LED strip should have 512 LEDs, and each will "handle" two 256 pixels lines (segments). Content of the Pixblasters controller registers is given in the remaining of the text.

Control register

Read "Enable" bit first. Example assumes that controller is disabled (no valid video at the input).

0x00A0 – Controller disabled, No mirroring, No padding, GRB pixel order,

CropX

0x000F – 15 pixels at the beginning of each line of the original video connected to video input will be skipped. 16th pixel in each line will be the first pixel sent to LED stripes.

CropY

0x000F – 15 lines at the beginning of each frame of the original video connected to video input will be skipped. 16th line of each frame will be the first line sent to LED stripes,

StoreX

0x00FF – 256 pixels will be taken from the original video connected to video input (starting at pixel 16 as defined in CropX) and sent to LED strips,

StoreY

0x003F – 64 lines will be taken from the original video connected to video input (starting at line 16 as defined in CropY) and sent to LED strips,

StoreHRES

0x00FF – Each LED strip line will have 256 pixels,

StorePRESCAL

0x0000 – T_{SLOT} = T_{STRIP} = 1/25.6MHz,

ProgLUTS
Zero code – 0x0000003F,
For addresses "AAAAA" 31 – 6 write sequence is as follows:
0000000AAAAA0100
0000000AAAAA0110
For addresses "AAAAA" 5 – 0 write sequence is as follows:
0000000AAAAA0101
0000000AAAAA0111
One code – 0x0000FFFF,
For addresses "AAAAA" 31 – 16 write sequence is as follows:
0000000AAAAA0100
0000000AAAAA0110
For addresses "AAAAA" 15 – 0 write sequence is as follows:
0000000AAAAA0101
0000000AAAAA0111

SegmentsNO
0x0001 – Two segments per strip

CfgLoadCntrl
0x0003 – FPGA configuration 1 will be loaded. FPGA configuration to load is defined by register bits 3-1

10.2 "Mirrored" Multi-controller Topology, 32 LED strips, Res - 1024x64

This example shows how to configure Pixblasters controllers operating in "Mirrored multi controller topology", with 32 LED strips connected to each controller and with resolution of 1024 x 64 Figure 10.1. Mirrored topology is used to double the maximum horizontal resolution. Required vertical resolution also exceeds the maximum number of LED strips that may be connected to a single controller, so two Pixblasters controllers are used per vertical column. Each LED strip should have 512 LEDs, and each will "handle" half of the line.

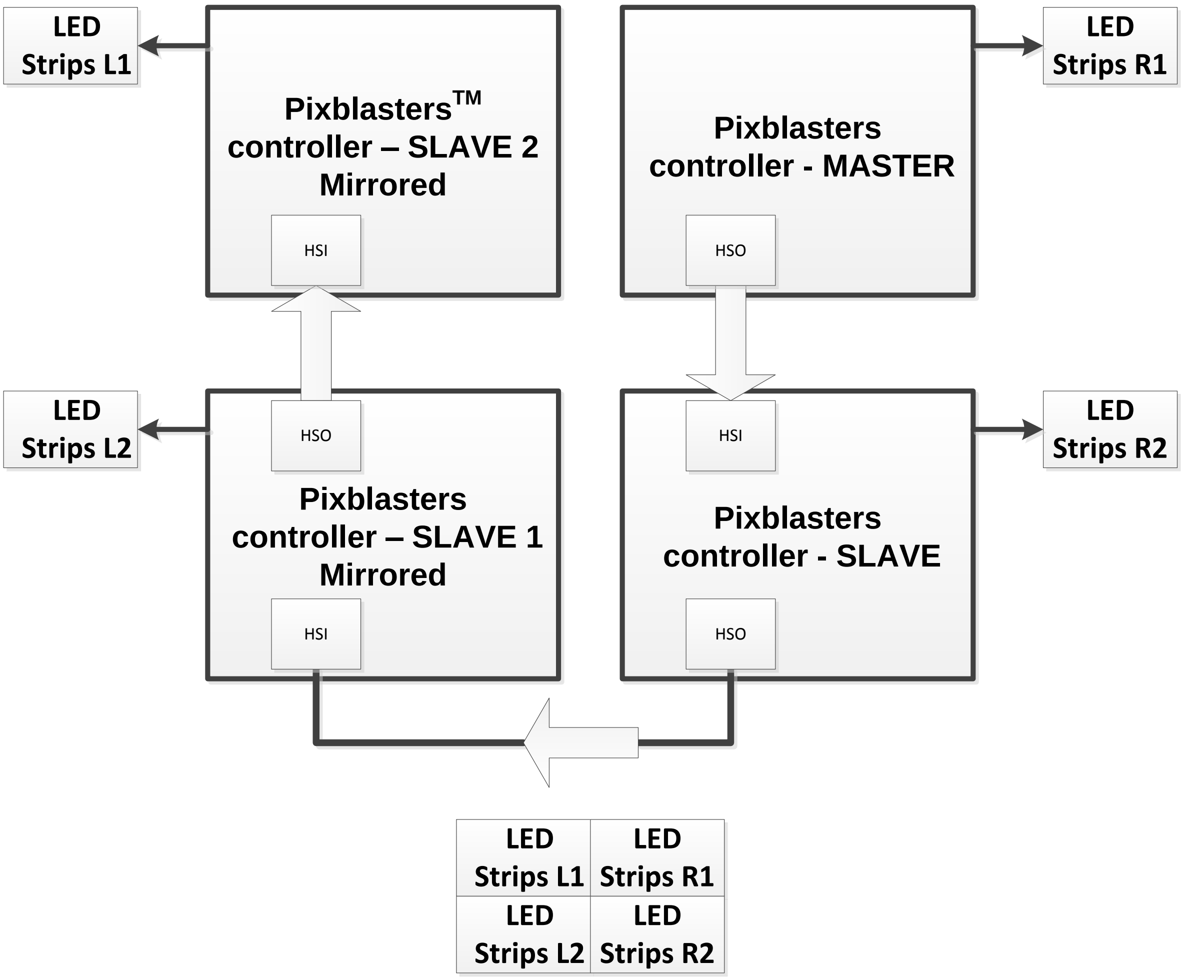

LED Strips L1
Pixblasters™ controller – SLAVE 2 Mirrored
HSI
Pixblasters controller - MASTER
HSO
LED Strips R1
LED Strips L2
HSO
Pixblasters controller – SLAVE 1 Mirrored
HSI
HSI
Pixblasters controller - SLAVE
HSO
LED Strips R2
LED Strips L1
LED Strips R1
LED Strips L2
LED Strips R2

Figure 10.1 Example of Mirrored Multi-Controller Topology

Register content for each of four Pixblasters controllers is given in the remaining of the text.

10.2.1MASTER Controller – Not Mirrored

Control register

Read "Enable" bit first. Example assumes that controller is disabled (no valid video at the input).

0x00A0 – Controller disabled, No mirroring, No padding, GRB pixel order,

CropX

0x0200 – 512 pixels at the beginning of each line of the original video connected to video input will be skipped. 513rd pixel in each line will be the first pixel sent to LED stripes,

CropY

0x0000 – 0 lines at the beginning of the frame of the original video connected to video input will be skipped. First line of each frame will be the first line sent to LED stripes,

StoreX

0x01FF – 512 pixels will be taken from the original video connected to video input (starting at pixel 513 as defined in CropX) and sent to LED strips,

StoreY

0x001F – 32 lines will be taken from the original video connected to video input (starting at first line as defined in CropY) and sent to LED strips,

StoreHRES

0x01FF – Each LED strip line will have 512 pixels,

StorePRESCAL

0x0000 – T_{SLOT} = T_{STRIP} = 1/25.6MHz,

ProgLUTS

Zero code – 0x0000003F,
For addresses "AAAAA" 31 – 6 write sequence is as follows:
0000000AAAAA0100
0000000AAAAA0110
For addresses "AAAAA" 5 – 0 write sequence is as follows:
0000000AAAAA0101
0000000AAAAA0111
One code – 0x0000FFFF,
For addresses "AAAAA" 31 – 16 write sequence is as follows:
0000000AAAAA0100
0000000AAAAA0110
For addresses "AAAAA" 15 – 0 write sequence is as follows:
0000000AAAAA0101
0000000AAAAA0111

SegmentsNO

0x0000 – No segments

CfgLoadCntrl

0x0003 – FPGA configuration 1 will be loaded. FPGA configuration to load is defined by register bits 3-1

10.2.2SLAVE Controller – Not Mirrored

Control register

Read "Enable" bit first. Example assumes that controller is disabled (no valid video at the input).
0x00A0 – Controller disabled, No mirroring, No padding, GRB pixel order,

CropX

0x0200 – 512 pixels at the beginning of each line of the original video connected to video input will be skipped. 513rd pixel in each line will be the first pixel sent to LED stripes,

CropY

0x0020 – 32 lines at the beginning of the frame of the original video connected to video input will be skipped. First line of each frame will be the first line sent to LED stripes,

StoreX

0x01FF – 512 pixels will be taken from the original video connected to video input (starting at pixel 513 as defined in CropX) and sent to LED strips,

StoreY

0x001F – 32 lines will be taken from the original video connected to video input (starting at line 33 as defined in CropY) and sent to LED strips,

StoreHRES

0x01FF – Each LED strip line will have 512 pixels,

StorePRESCAL

0x0000 – T_{SLOT} = T_{STRIP} = 1/25.6MHz,

ProgLUTS

Zero code – 0x0000003F,
For addresses "AAAAA" 31 – 6 write sequence is as follows:
0000000AAAAA0100
0000000AAAAA0110
For addresses "AAAAA" 5 – 0 write sequence is as follows:
0000000AAAAA0101
0000000AAAAA0111
One code – 0x0000FFFF,
For addresses "AAAAA" 31 – 16 write sequence is as follows:
0000000AAAAA0100
0000000AAAAA0110
For addresses "AAAAA" 15 – 0 write sequence is as follows:
0000000AAAAA0101
0000000AAAAA0111

SegmentsNO

0x0000 – No segments

CfgLoadCntrl
0x0003 – FPGA configuration 1 will be loaded. FPGA configuration to load is defined by register bits 3-1

10.2.3SLAVE 1 Controller – Mirrored

Control register
Read "Enable" bit first. Example assumes that controller is disabled (no valid video at the input).
0x00A2 – Controller disabled, Mirrored, No padding, GRB pixel order,

CropX
0x0000 – 0 pixels at the beginning of each line of the original video connected to video input will be skipped. First pixel in each line will be the first pixel sent to LED strips,

CropY
0x0020 – 32 lines at the beginning of the frame of the original video connected to video input will be skipped. First line of each frame will be the first line sent to LED strips,

StoreX
0x01FF – 512 pixels will be taken from the original video connected to video input (starting at first pixel as defined in CropX) and sent to LED strips,

StoreY
0x001F – 32 lines will be taken from the original video connected to video input (starting at line 33 as defined in CropY) and sent to LED strips,

StoreHRES
0x01FF – Each LED strip line will have 512 pixels,

StorePRESCAL
0x0000 – T_{SLOT} = T_{STRIP} = 1/25.6MHz,

ProgLUTS
Zero code – 0x0000003F,
For addresses "AAAAA" 31 – 6 write sequence is as follows:
0000000AAAAA0100
0000000AAAAA0110
For addresses "AAAAA" 5 – 0 write sequence is as follows:
0000000AAAAA0101
0000000AAAAA0111
One code – 0x0000FFFF,
For addresses "AAAAA" 31 – 16 write sequence is as follows:
0000000AAAAA0100
0000000AAAAA0110
For addresses "AAAAA" 15 – 0 write sequence is as follows:
0000000AAAAA0101
0000000AAAAA0111

SegmentsNO
0x0000 – No segments

CfgLoadCntrl
0x0003 – FPGA configuration 1 will be loaded. FPGA configuration to load is defined by register bits 3-1

10.2.4SLAVE 2 Controller – Mirrored

Control register
Read "Enable" bit first. Example assumes that controller is disabled (no valid video at the input).
0x00A2 – Controller disabled, Mirrored, No padding, GRB pixel order,

CropX
0x0000 – 0 pixels at the beginning of each line of the original video connected to video input will be skipped. First pixel in each line will be the first pixel sent to LED stripes,

CropY
0x0000 – 0 lines at the beginning of the frame of the original video connected to video input will be skipped. First line of each frame will be the first line sent to LED stripes,
StoreX
0x01FF – 512 pixels will be taken from the original video connected to video input (starting at first pixel as defined in CropX) and sent to LED strips,

StoreY
0x001F – 32 lines will be taken from the original video connected to video input (starting at first line as defined in CropY) and sent to LED strips,

StoreHRES
0x01FF – Each LED strip line will have 512 pixels,

StorePRESCAL
0x0000 – T_{SLOT} = T_{STRIP} = 1/25.6MHz,

ProgLUTS
Zero code – 0x0000003F,
For addresses "AAAAA" 31 – 6 write sequence is as follows:
0000000AAAAA0100
0000000AAAAA0110
For addresses "AAAAA" 5 – 0 write sequence is as follows:
0000000AAAAA0101
0000000AAAAA0111
One code – 0x0000FFFF,
For addresses "AAAAA" 31 – 16 write sequence is as follows:
0000000AAAAA0100
0000000AAAAA0110

For addresses "AAAAA" 15 – 0 write sequence is as follows:
0000000AAAAA0101
0000000AAAAA0111

SegmentsNO
0x0000 – No segments

CfgLoadCntrl
0x0003 – FPGA configuration 1 will be loaded. FPGA configuration to load is defined by register bits 3-1

11 APPENDIX A – ABOUT LED STRIPS BASED DISPLAYS

Pixblasters LED controller enables creation of LED display installations that are:

1. Flexible enough to support practically any method of generating digital content, including the use of professional digital signage software
2. Easy to use and with the hardware architecture fully transparent to end users
3. Creatively different from other common products on the market
4. Large, or possibly enormous in size, and
5. Affordable to build and operate

The Pixblasters MS1 video LED controller's ability to connect to any computer and any operating system as an ordinary monitor successfully fulfills the first two requirements from the list. The last three project requirements are linked to a common question regarding Pixblasters - **why do we support LED strips and not the fixed LED modules?**

The Pixblasters team has recognized specific situations and applications where LED strips-based video screens can offer certain benefits and fill the holes in the existing LED display feature sets.

11.1 LED Strips are More Affordable and Manageable

The high-end module based LED displays are huge "electronic monsters" that implement hundreds and thousands of driving chips, carefully calibrated LED lots, air-conditioners to keep the whole system thermally stable, complex internal networking based on proprietary communication standards, etc. Due to the complexity that comes at high costs, those huge, shiny, beautiful video displays have always been beyond the available price range and beyond the area of technical expertise of the vast majority of enthusiasts and small business owners.

The Pixblasters MS1 video LED controller wants to change it and to make big video LED displays affordable to everyone. In combination with LED strips, this video controller assures the most important LED display features like the smooth video reproduction, appealing color content and big displays size.

With the LED strip pricing in a range of $2 - 5 per (source: Alibaba.com), LEDs for the 8 meters long and 0.3 meters high (480 x 16 resolution) would cost between $256 – 640.

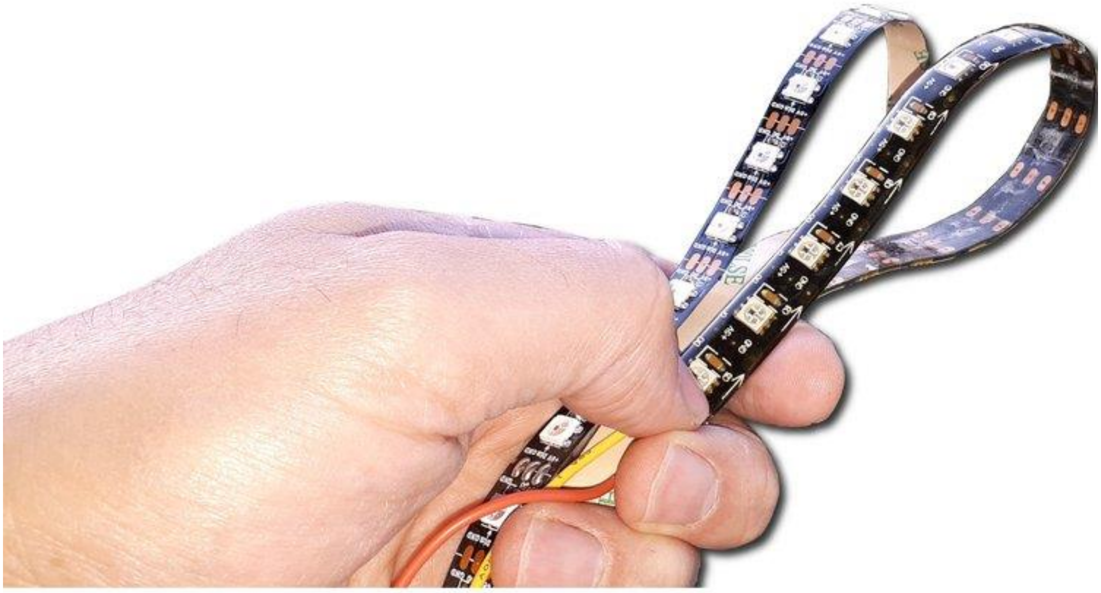


Figure 11.1 LED Strip (left) with Low Water Resistance and the Waterproof LED Strip (right)

11.2 LED Strips are Flexible, Lighter, and Fits to a Variety of Surfaces

Responsive and LED strips based video displays offer some features that cannot be attained by standard LED modules. The LED strips can be curved in different shapes. Wrap them around the round pillars, glue them as a transparent curtain in the shopping window glass, embed the animated strips in big printed jumbo posters, cover the room's ceiling or even the complete street vault by video LED strips display, and new video installations that are hard or impossible to build with LED modules become reality!

Lightweight LED strips do not need complex mechanical supports. They can be easily glued or stapled to different surfaces, inserted in LED pipes, and combined in various lattice structures, etc. Literally within minutes, LED strips can be glued to a plain smooth wall to convert it into a video screen, which can be wirelessly controlled through the Internet.

The strips come with different numbers of LEDs per meter/foot, with LEDs of different size, in different PCB colors, and with different waterproof ratings. The **Figure 11.1** shows both types, the RGB LED strip with a very low water resistance and the waterproof LED strip that is fully covered by the silicon coating and suitable for outdoors applications.

11.3 LED Strips are More Durable, and Easier to Repair if They do Break

It sounds a bit funny, but the **LED strips based video screens can be cut and repaired by ordinary scissors.** The non-functional part of the LED screen can be easily cut off and replaced with new pieces of the LED strip! Much easier than in more complex LED module based video displays that often have scrambled screen parts while waiting for repair.

11.4 LED Strips Can be Adjusted for Pixel Pitch and Density

Chained Pixblasters Video LED Controllers can drive LED displays with HD (High Definition) video resolutions, but we strongly believe that great and engaging video results can be also achieved by LED displays, which have smaller video resolution and are great in size. It is somehow related to the way humans see – our eyes and brain connect a smaller number of visible dots (pixels) in a smooth video image.

A bigger pixel pitch (distance between two pixels either horizontally or vertically) and the lower pixel density significantly decrease assembly and operational costs of such displays. With the LED strips, the pixel pitch and pixels density can be freely changed, which cannot be done by predefined and fixed pitch LED modules. A school gym scoreboard visible from a big distance may have a significantly bigger pixel pitch than a room's ceiling decoration that should be visible from a closer distance. Please note that the Pixblasters MS1 controller drives serially connected addressable RGB LEDs, which can be supplied as the off-the-shelf LED strips with a horizontally fixed pitch, or self-made LED strips (single pixels wired) with an arbitrary setup pixel pitch.

As a conclusion, a modules based video LED display that i.e. shows news with letters of human height size, requires much more LEDs than the LED strips based video display. Of course, the first one usually provides a higher quality image, but the strips based one generates a comparable video effect at a much lower cost!

A single Pixblasters controller directly drives 16,384 LEDs, which in displays built up from the 30 LEDs/m (pitch 33 mm), turns into an active video surface of approx. 17 square meters (~ 183 ft2). Multiple Pixblasters controllers can be chained to build economically viable video installations that can span entire buildings. Potentially it can turn the complete shopping mall's façade into responsive and remotely controlled video display that shows any multimedia content.

On the other side of the display size-spectrum are ordinary illuminated signs on shops, coffee bars, gas stations… that can be easily converted into video displays connected to the Internet. Such conversion only requires the exchange of the existing background LEDs by the addressable RGB LEDs driven by the Pixblasters and a small computer like the Raspberry Pi Zero, or similar.

A number of used LEDs is not the only significant cost driver. The LED modules require a higher number of driver chips and each LED module contains such driving electronics.

11.5 The Pixblasters MicroSign Demo

The LED strips based displays have significantly simpler construction and all electronic functions are contained in the Pixblasters MS1 video LED controller. It can be easily illustrated by our Pixblasters MicroSign (**Figure 11.2**) display, which was available only during the crowdfunding campaign for demonstration purposes.

The photo shows all necessary display parts. The Raspberry Pi computer generates the video output connected to the Pixblasters MS1 video LED controller. The demo uses only a half of available 32 digital outputs and drives 120 x 16 (1,920) LEDs on 16 LED strips attached through the Pixblasters Array-S terminal board. This board can be easily exchanged by any kind of wire terminals. There is also a power supply for the complete video display.

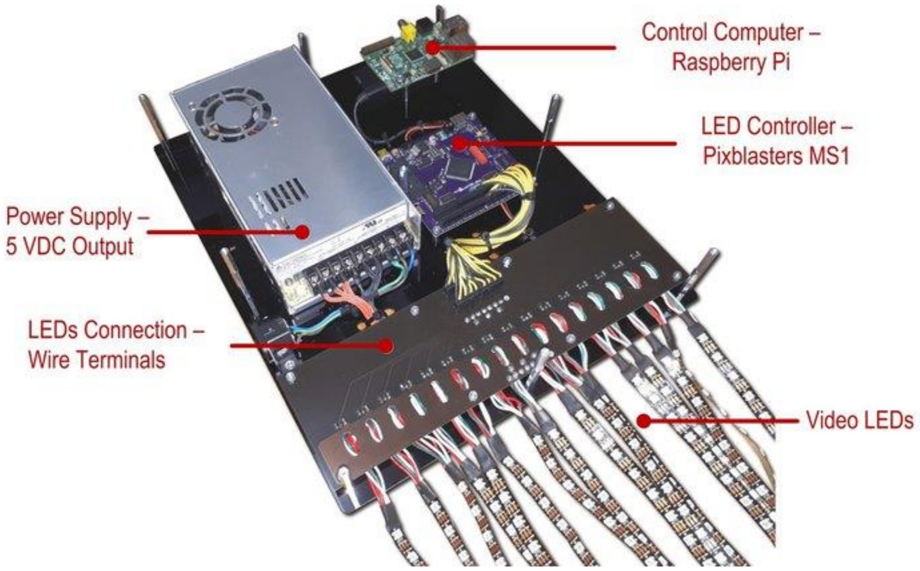


Figure 11.2 The Pixblasters MicroSign Demo (resolution 120x16, display size 2 x 0.3 m)

(not available as a product from Pixblasters!)

Display like this one can be built in one day! It was named the MicroSign due to a number of connected LEDs. But, the LED driving capabilities of this simple electronic assembly are much, much higher!

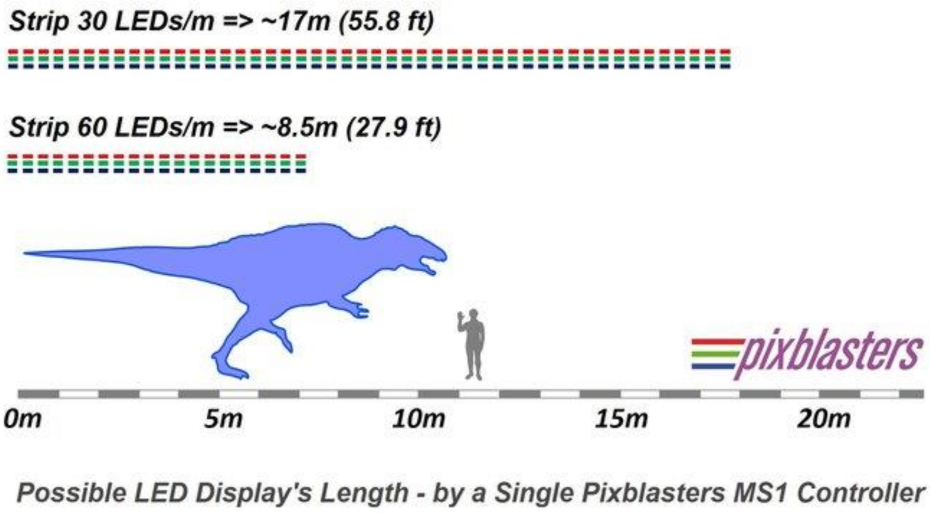

Strip 30 LEDs/m => ~17m (55.8 ft)
Strip 60 LEDs/m => ~8.5m (27.9 ft)
pixblasters
0m
5m
10m
15m
20m
Possible LED Display's Length - by a Single Pixblasters MS1 Controller

Figure 11.3 Achievable LED Display Lengths by a Single Pixblasters MS1 Video Controller

Figure 11.3 illustrates the length of LED displays that can be controlled by the MicroSign, and that length is simply impressive. Additionally, the unused digital outputs can effectively double the vertical display's resolution and the number of LEDs. Of course, the bigger number of LEDs requires more power and the power supply unit from the Figure 11.2 would not be sufficient for more than 2,000 LEDs. The MicroSign LED displays with higher resolutions would require additional power supply units, but the LED control electronic remains the same.

The following video clip shows the Pixblasters MicroDemo in action. We think that this simple display construction, with the activated Internet access and the digital signage software, can transform i.e. dull street shop kiosks into multimedia hot spots. Glue the waterproof LED strips around the kiosk's top and it will light up!

Please note that the MicroSign display demo use the Pixblasters open-source FPGA. The code is available from here: https://github.com/PixiGreen/Pixblasters-MicroDemo

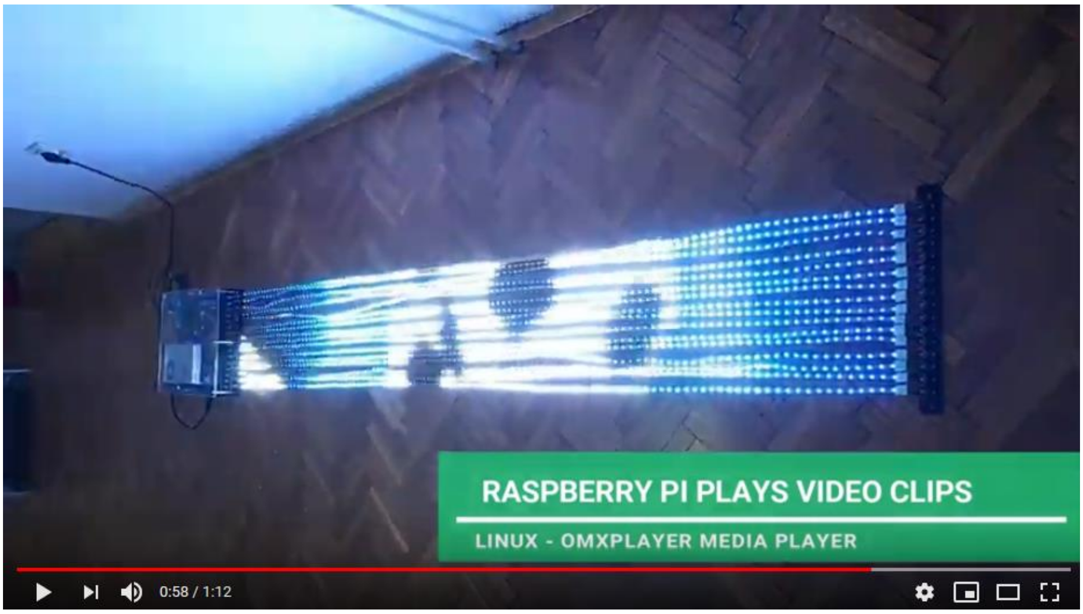

RASPBERRY PI PLAYS VIDEO CLIPS
LINUX - OMXPLAYER MEDIA PLAYER
0:58 / 1:12

Figure 11.4 Pixblasters MicroSign - Open Source LED Display Linux Test Controller

(https://www.youtube.com/watch?v=CLNaBz24IXY)

12 APPENDIX B - µC AND FPGA DIRECT PROGRAMMING

Programming steps described in this paragraph may **PERMANENTLY DISABLE** the Pixblasters functionality and should be executed only by skilled developers who want to use the MS1 board hardware in ways that differ from its original purpose.

Pixblasters **do not provide PIC18F26J50 firmware** for download to the end users. Once deleted, it becomes **PERMANENTLY LOST** to the end user! The same applies to the FPGA configuration 0. If the EEPROM containing FPGA configurations is completely erased, user will lose ability to update remaining FPGA configurations!

12.1 Direct µC Programming

Direct µC programming is accomplished by connecting the compatible Microchip programming tool to the µC programming port (CN3). Figure 12.1 shows the programming port's pinout.

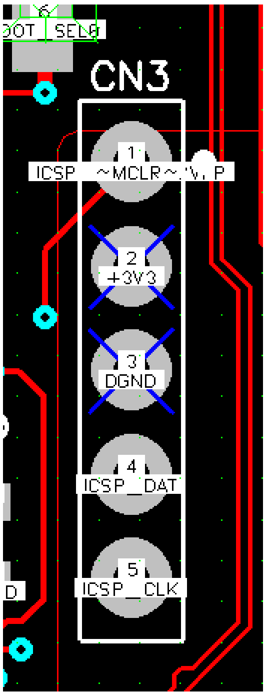


Figure 12.1 µC Programming Port Pinout

12.2 Direct FPGA Programming

Direct FPGA programming is accomplished by connecting the appropriate JTAG programmer to the FPGA JTAG programming port (CN5, Connector type - 87832-1420). **Figure 12.2** shows the JTAG programming port's pinout. It is compatible with the Xilinx Platform Cable USB programmers.

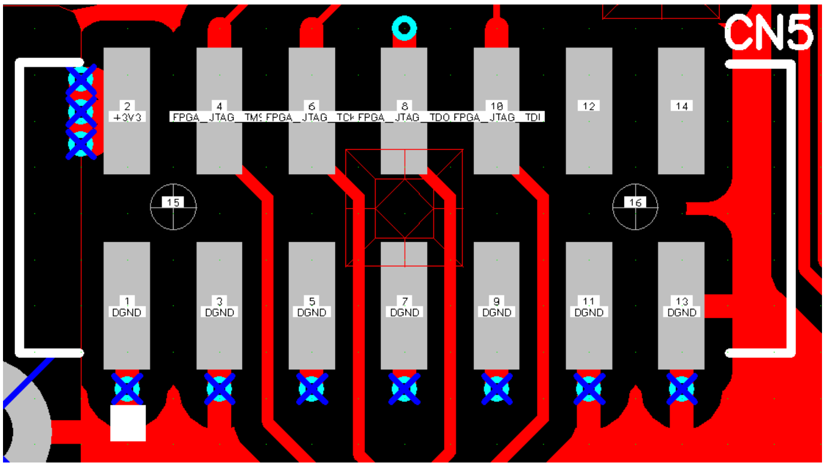


Figure 12.2 FPGA Programming Port Pinout

12.2.1 On-board Watchdog

During the programming, the onboard watchdog has to be disabled. This is accomplished by shorting the connector CN2 (FPGA program/run selector). If the watchdog function is required, the CN2 should be left open and FPGA functionality should be provided for performing watchdog refresh.

Revision History

Version	Date	Note
1.00	11.03.2020.	Initial release