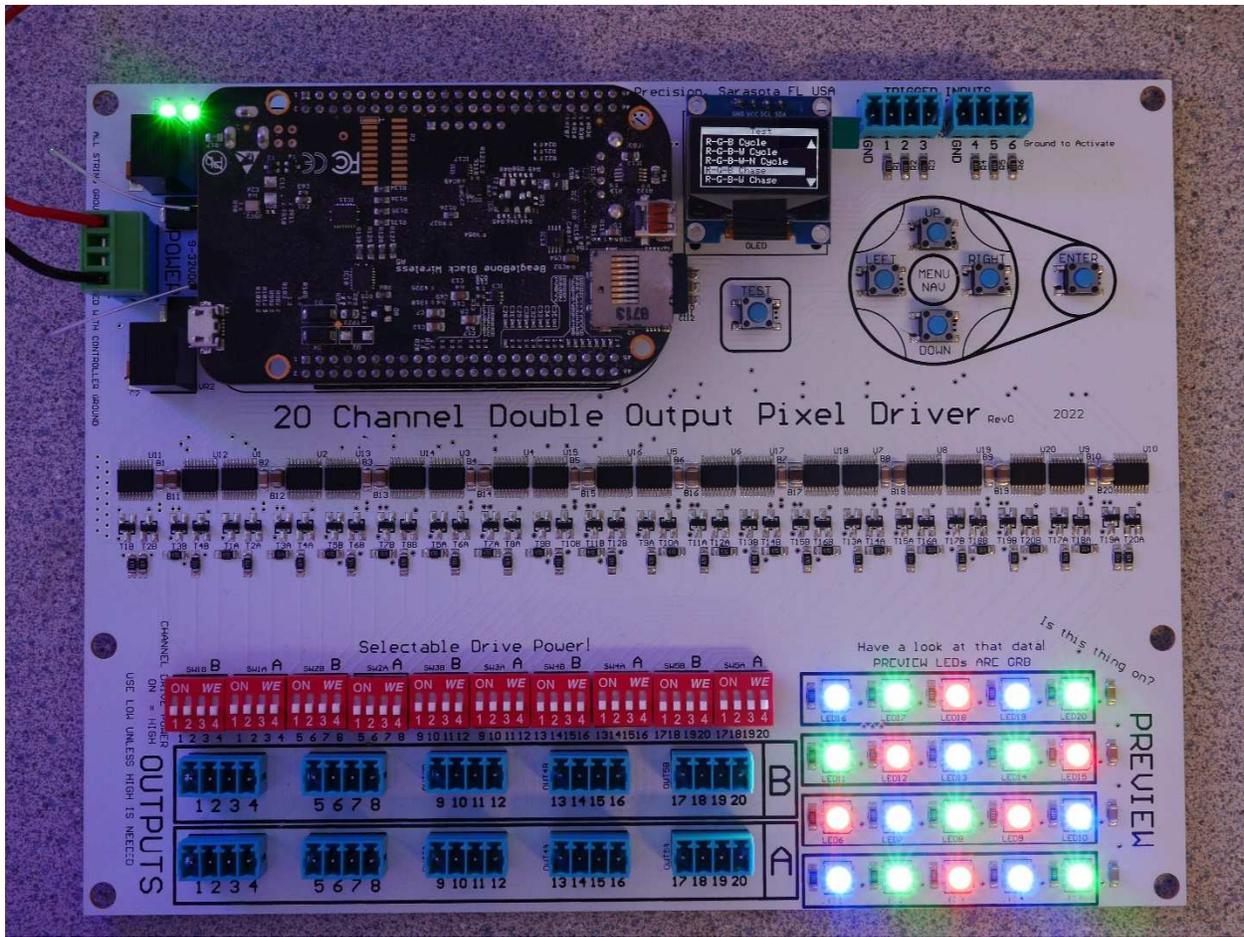


# 20 Channel Double Output Beagle Board Black Pixel Driver

Oct. 2022





## **Thank you for your purchase of our pixel driver board!**

The 20 channel double output BBB pixel driver was designed to offer the maximum amount of reliable data ports from the beagle board black SBC.

A and B mirrored outputs with independent driver circuitry allow two redundant data outputs per channel. The common use for this is driving two sides of a prop where only one side is visible from any given position. This simplifies the amount of channels needed to run a greater amount of lights.

### **Power Specs and info**

The Beagle Board and driver circuitry supply is reverse input and overvoltage protected.

The board can run on 9-32VDC. It is designed for a nominal voltage of 12 or 24VDC. There is NO option to run on 5V at this time (this is to maintain a robust controller with onboard regulation while still keeping it simple). 5v pixels can be used by simply having a separate small 12 or 24v power supply for the board.

Under normal conditions, the driver board consumes a little less than an amp at 12v or a little under half an amp on 24v. The more outputs that are "loaded", the more power the board will consume with its high power data drivers to maintain signal integrity.

The blue terminal blocks and headers are of the 3.5mm pitch variety.

It is important to note that this board is meant to be paired with nearby power distribution blocks. The ground connection of the board's power supply is the sole grounding point for the pixel data and thus the pixels strings all need to be bonded to the controller's ground. The quality of this ground topology has major effects on data distance. Two connections on the power input block are for ground wires. It is advisable to use both to provide a solid connection since 40 channels of high speed data add up to a significant load.

## Pixel Data Outputs

The 40 data outputs on the board provide the standard 5V “TTL” signal level used by almost all pixels. All data timing and protocol comes from the FPP software. This board is strictly a data driver in that it simply provides level translation and robust chips to send the high speed data through as much wire length as possible without glitching.

The “special feature” of our boards is the data power level DIP switch. The normal position is “OFF” or “LOW”. Through many hours of research we arrived at a driver output impedance that provided the cleanest signal over the greatest amount of wire scenarios. Nevertheless, sometimes things aren’t perfect when trying to string wire all over the place and you need a little more juice to push that data out there. Now all you have to do is switch the affected channels to “ON” or “HIGH” to lower the output impedance and likely eliminate any glitching. This isn’t a miracle cure that will make bad wiring work every time but we found that it solves problems often enough to make it worthwhile adding to every board.

All channels are TVS protected from static electricity or wiring mishaps. We’ve made this as strong as was reasonable but please don’t try to blow them up on purpose!

## Preview LEDs

The built-in RGB LEDs are in parallel with the B data outputs (through additional current limiting resistors to ensure that a fluke failure in the pixel does not kill your whole string). This allows them to show the contents of the first pixel in your string without affecting any addresses or needing null pixels or any other nonsense! Note that the preview LEDs are G-R-B color order (the only way they’re available) so if you have RGB pixels the red and green would be swapped on the preview. The LEDs are mostly on the board to determine if data is actively being sent. If they are not lighting up, one of three things is going on; no data being sent from FPP on that channel, null data being sent by FPP for that address, or the string is shorted externally. Technically, an output driver could be blown up but we have taken many precautions to make that as difficult as possible to do!

## Opto-isolated Trigger Inputs

We've included the sought after circuitry of opto-isolated and hardware debounced easy to actually connect and use trigger inputs. There are two pluggable terminal blocks provided for a total of 6 inputs.

To use these inputs, simply connect your button between the input and the provided ground connection. Finally there is no need to worry about wire distance or GPIO voltages or static electricity.... We've taken care of it all for you! Just hook up some buttons!

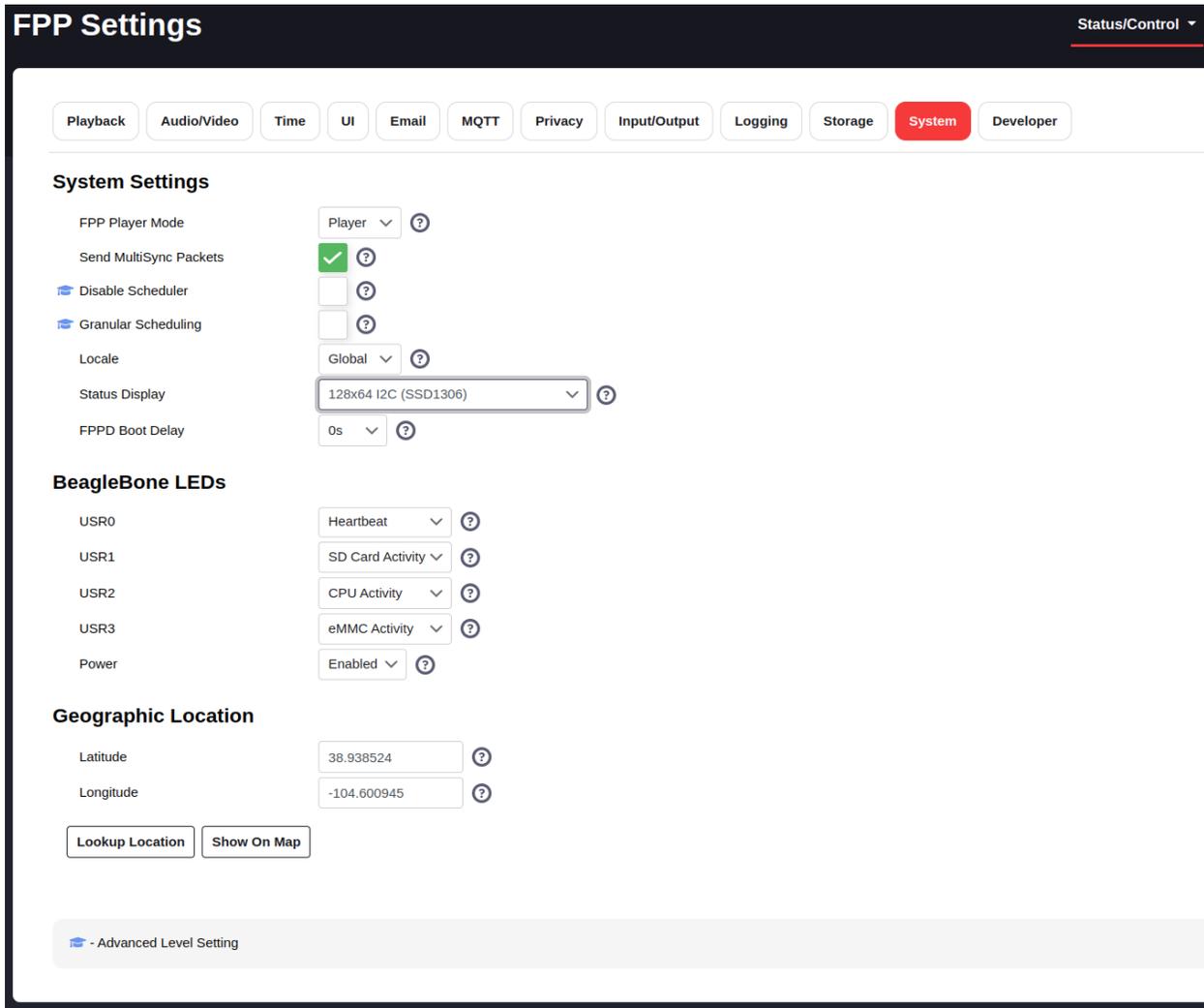
When an input is activated, our circuit produces a "rising edge" trigger to the FPP software. You can configure an action for this on the GPIO inputs page of FPP.

The GPIO pin assignments are P9 11-16

## Status OLED and navigation buttons

This model board includes the standard status OLED screen with 6 buttons for easy use. To make the display work, it must be enabled in the FPP settings > (system).

Select "128x64 I2C (SSD1306)"



The screenshot shows the FPP Settings web interface. At the top, there's a navigation bar with tabs for Playback, Audio/Video, Time, UI, Email, MQTT, Privacy, Input/Output, Logging, Storage, System (highlighted in red), and Developer. The main content area is titled "System Settings" and includes several configuration options:

- FPP Player Mode: Player (dropdown)
- Send MultiSync Packets:  (checkbox)
- Disable Scheduler:  (checkbox)
- Granular Scheduling:  (checkbox)
- Locale: Global (dropdown)
- Status Display: 128x64 I2C (SSD1306) (dropdown, highlighted)
- FPPD Boot Delay: 0s (dropdown)

Below this is the "BeagleBone LEDs" section with dropdown menus for USR0 (Heartbeat), USR1 (SD Card Activity), USR2 (CPU Activity), USR3 (eMMC Activity), and Power (Enabled).

The "Geographic Location" section has input fields for Latitude (38.938524) and Longitude (-104.600945), along with "Lookup Location" and "Show On Map" buttons.

At the bottom, there's a link for "Advanced Level Setting".

To make the buttons work, the GPIO pins must be assigned for menu navigation.

[Status/Control](#) | [Content Setup](#) | [Input/Output Setup](#) | [Help](#)

## GPIO Input Triggers

Save

EN.	HDR.PIN	GPIO# - GPIOD	PULL UP/DOWN	DESCRIPTION	COMMANDS: RISING EDGE	COMMANDS: FALLING EDGE
<input type="checkbox"/>	P8-03	38 - 1/6	None/External		Command: <input type="text"/>	Command: <input type="text"/>
<input type="checkbox"/>	P8-04	39 - 1/7	None/External		Command: <input type="text"/>	Command: <input type="text"/>
<input type="checkbox"/>	P8-05	34 - 1/2	None/External		Command: <input type="text"/>	Command: <input type="text"/>
<input type="checkbox"/>	P8-06	35 - 1/3	None/External		Command: <input type="text"/>	Command: <input type="text"/>
<input type="checkbox"/>	P8-07	66 - 2/2	None/External		Command: <input type="text"/>	Command: <input type="text"/>
<input type="checkbox"/>	P8-08	67 - 2/3	None/External		Command: <input type="text"/>	Command: <input type="text"/>
<input type="checkbox"/>	P8-09	69 - 2/5	None/External		Command: <input type="text"/>	Command: <input type="text"/>
<input type="checkbox"/>	P8-10	68 - 2/4	None/External		Command: <input type="text"/>	Command: <input type="text"/>
<input checked="" type="checkbox"/>	P8-11	45 - 1/13	None/External		Command: OLED Navigation Multicast: <input type="text"/> Action: Up	Command: <input type="text"/>
<input checked="" type="checkbox"/>	P8-12	44 - 1/12	None/External		Command: OLED Navigation Multicast: <input type="text"/> Action: Down	Command: <input type="text"/>
<input checked="" type="checkbox"/>	P8-13	23 - 0/23	None/External		Command: OLED Navigation Multicast: <input type="text"/> Action: Enter	Command: <input type="text"/>
<input checked="" type="checkbox"/>	P8-14	26 - 0/26	None/External		Command: OLED Navigation Multicast: <input type="text"/> Action: Back	Command: <input type="text"/>
<input checked="" type="checkbox"/>	P8-15	47 - 1/15	None/External		Command: OLED Navigation Multicast: <input type="text"/> Action: Enter	Command: <input type="text"/>
<input checked="" type="checkbox"/>	P8-16	46 - 1/14	None/External		Command: OLED Navigation Multicast: <input type="text"/> Action: Test	Command: <input type="text"/>
<input type="checkbox"/>	P8-17	27 - 0/27	None/External		Command: <input type="text"/>	Command: <input type="text"/>
<input type="checkbox"/>	P8-18	65 - 2/1	None/External		Command: <input type="text"/>	Command: <input type="text"/>
<input type="checkbox"/>	P8-19	22 - 0/22	None/External		Command: <input type="text"/>	Command: <input type="text"/>
<input type="checkbox"/>	P8-20	63 - 1/31	None/External		Command: <input type="text"/>	Command: <input type="text"/>
<input type="checkbox"/>	P8-21	62 - 1/30	None/External		Command: <input type="text"/>	Command: <input type="text"/>
<input type="checkbox"/>	P8-22	37 - 1/5	None/External		Command: <input type="text"/>	Command: <input type="text"/>
<input type="checkbox"/>	P8-23	36 - 1/4	None/External		Command: <input type="text"/>	Command: <input type="text"/>

For user convenience, the GPIO pin assignments for both the trigger inputs as well as the menu buttons are printed on the back of the PCB.

## Attaching the BeagleBoard

The BeagleBoard can only plug on one way. Other components on this board block it from being plugged on when turned 180 degrees. Make note of the board outline on the PCB and make sure to align all of the pins on the tall headers... all 92 of them!

## USB Precautions

The USB connector on the BeagleBoard is useful for connecting to a PC to configure network settings for the first time. Note that the USB port ground is bonded to the string negative of the power on the driver board (no reasonable way to avoid this). It is strongly recommended NOT to use the USB connection when the board is powered from an external power supply. If there is a ground differential between your computer and your pixel power supply, many unavoidable problems arise which result in blowing up the beagle board or USB port on the PC. This can be avoided by using a laptop computer (not plugged in to AC power), or by simply removing the BeagleBoard from the driver when using the USB tether. Fortunately USB tether is not needed once proper network settings are applied. WiFi tether can usually handle all setup needs anyway.

## Setting up FPP to use the Pixel Driver Board

If you have used FPP on BBB before, skip to step 5

1. Download the latest FPP image for BeagleBoneBlack from the Github page  
<https://github.com/FalconChristmas/fpp/releases>
2. Use “Balena Etcher” or similar software to “burn” the SD card image onto a new card. Note that this will erase anything that was on the card.
3. Insert the card into the BeagleBoard
4. Connect to the FPP web interface
  - Use USB cable to power and connect (192.168.7.2)
  - Power with driver board and use WiFi generated by the BBBMore info on these steps is available in the FPP manual  
[https://github.com/FalconChristmas/fpp/releases/download/5.5/FPP\\_Manual.5.5.pdf](https://github.com/FalconChristmas/fpp/releases/download/5.5/FPP_Manual.5.5.pdf)
5. Navigate to the FPP status page by typing in the Board’s IP address in the browser of your choice
6. Locate the “Input/Output Setup” tab near the upper right corner of the page
7. Select “Channel Outputs”
8. Select the “Pixel Strings” tab

For Revisions 0 and 1, there is no “cape” autodetect functionality built into the driver board. There are two solutions for configuring the pixel outputs.

### Solution 1

The driver board can be used as-is by selecting “Cape-Type”

F8-B (No Serial)    Version 2.x

Note that 20 channels are available. Take care to “zero out” all unused channels as erroneous information on this page will cause FPP to not run.

## **Solution 2**

**We are working on getting the driver board definition incorporated for auto detecting, in the mean time we have the cape's definition file available to be manually added to FPP on your SD card.**

*If you purchase an SD card from us, we can do this in advance for you*

- 1. Download the cape definition JSON file from the [ledpixeldriver.com](http://ledpixeldriver.com) page**
- 2. Plug the SD (with FPP already installed) into a card reader on a linux computer such as a raspberry pi with keyboard, mouse, and monitor which is already booting and running on its own SD card.**
- 3. Copy the JSON file into `"/opt/fpp/capes/bbb/strings"` on the BBB SD card**

**Now on the "Pixel Strings" page, you can select "Cape-Type" "ARBP-2040"**

**This way, the channels show up without the irrelevant notes about smart receivers or differential channels.**

**For further information about setting up the attributes for each string channel, consult the FPP manual or forum. This setup is the same for ALL FPP devices and is beyond the scope of the driver board.**



## Support

For all inquiries please contact

[support@ledpixeldriver.com](mailto:support@ledpixeldriver.com)

Please let us know if you would like anything added to or elaborated in the manual.