DiSTRiBUTiON
LED PLATFORM

A REMIX ON THE D14 SIGN

BY BRAD OSGOOD

Source: https://github.com/bosgood/d14

Greetings DOer!

If you are reading this, you are either trying to repair the D14 sign or in some other way prepare it for use at a Disorient event. This guide will attempt to explain its construction and provide a rough operational manual.

ANATOMY

Each letter contains the following elements:

- **Frame** - the wood frame, created in 2014
- **LED strips** - each letter has up to 4 WS2812 LED strips. They are often powered from both ends, but the data signal only goes in one direction.
- **DiStribution board** - a Pixelblaze breakout board custom-designed in 2023 for this purpose
- **Pixelblaze V3 standard** - an ESP32-based LED control module created by Ben Hencke, also a burner
- **Pixelblaze Output Expander** - expands Pixelblaze to be able to control up to 8 LED strips per module

Each letter can operate completely standalone of the other 8. However, they are configured to connect to a WiFi network broadcast by the (included) Raspberry Pi access point if it is available. The Raspberry Pi runs an instance of Firestorm (https://github.com/Simap/firestorm), which synchronizes pattern timing, and can also be used to control all connected letters as a unit.

HARDWARE REPAIR

Some spare hardware may be available with the sign, but more can be purchased with these instructions:

- **Pixelblaze V3 Standard** - https://shop.electromage.com/products/pixelblaze-v3-standard-wifi-led-controller
- **Pixelblaze Output Expander** - https://shop.electromage.com/products/pixelblaze-output-expander-serial-to-8x-ws2812-apa102-driver
- **DiStribution board** - order from OSHPark (standard 2-layer service), allow at least 2 weeks lead time. Find PCB files in `hardware/` directory in the d14 Github repository.
- **Connectors** - order terminal block connectors from Digi-Key, allow about 2 weeks lead time
  - 3-pin female - https://www.digikey.com/en/products/detail/cui-devices/TBP01R2-508-03BE/10238400

- 3-pin male - https://www.digikey.com/en/products/detail/cui-devices/TBP01P1-508-03BE/10238368
- 2-pin male - https://www.digikey.com/en/products/detail/cui-devices/TBP01P1-508-02BE/10238367
- 2-pin female - https://www.digikey.com/en/products/detail/cui-devices/TBP01R2-508-02BE/10238399

New Pixelblaze chips will need to be initialized with LED mapping data, which can be found in the `led-mapping/` directory in the Github repository.

For miscellaneous issues, see the Troubleshooting section below for tips on triaging common problems.

SOFTWARE CONTROL

Firestorm WiFi network

- **SSID**: `d22_firestorm`
- **Password**: `disorient`

Resources

- Pixelblaze language reference: https://electromage.com/docs/language-reference
- Pixelblaze V3 documentation: https://electromage.com/docs/quickstart-v3-standard
- Firestorm REST API reference: https://github.com/simap/Firestorm#firestorm-api

Each letter contains its own copy of each LED pattern, and can operate completely standalone. When orchestrated by Firestorm, each letter does not receive LED pattern control data; it only receives time synchronization data. However, Firestorm has a control API that can be used to propagate pattern data written on one Pixelblaze to the others, allowing an edit-then-deploy workflow for new patterns.

Pattern live editing and deploy

Choose a letter to develop on. Through the Firestorm UI, connect to the Pixelblaze for this letter. A language reference is available on the same page as the patten editor, below the text input. Each keystroke will cause Pixelblaze to reload the pattern. Have fun programming LED patterns live on Disorient art!

Once your change is ready to be deployed to the other letters, clone the d14 Github repo, and locate the `scripts/` directory. There is a helper script (`d14.py`) available to help make controlling Firestorm easier, but consider the Firestorm REST API reference above to be the source of truth in case of any issues.

Using d14.py

```
python d14.py scan
```
List name/IDs of all online Pixelblaze nodes

```
python d14.py discover
```
List complete information available for all online Pixelblaze nodes

```
python d14.py deploy --source='distribution-d' --dest='distribution-
i1,distribution-s,distribution-o,distribution-r,distribution-
i2,distribution-e,distribution-n,distribution-t'
```
Overwrite all Pixelblaze nodes with the contents of the 'D' letter. This should be your primary deployment command when live-coding.

TROUBLESHOOTING

Ultimately, the sign is a fragile piece of electronic art subjected to harsh environmental conditions, over a long period of time - it will continue to break and require repairs.

Random colors flashing on LED strip

This issue is happening on some of the LED strips that have been damaged over the years. First check to see that the LED strip is properly powered from both ends, if connectors are available. Low voltage levels can cause an LED strip to be unable to properly light up, especially for brighter colors (right => low power, white => high power).
If only individual pixels at the end of a strip are flashing randomly, it is possible that those pixels need to be disabled in the Pixelblaze Settings page. Locate the channel that strip is connected to, and decrease the Count value by the number of pixels that need to be hidden.

Pixelblaze WiFi troubleshooting

If a letter does not show on the Firestorm GUI, first try restarting it by power-cycling the letter. If its LEDs are working, but it still is not discoverable by Firestorm, you can try disconnecting/reconnecting it to the Firestorm WiFi network. Be sure the Raspberry Pi is online and accessible (with a phone, or laptop) before continuing, if you intend to continue to use Firestorm.

To cause an individual Pixelblaze to enter AP search mode, press and hold the small round button on the controller while providing power to the letter. A new WiFi network will appear and using the captive portal that opens when joining the network, the letter can be joined to another WiFi network or used in AP mode at: http://192.168.4.1

Firestorm WiFi troubleshooting

The Raspberry Pi is also a sensitive piece of electronics being exposed to harsh environmental conditions. If its WiFi network is not broadcasting, you can attempt to troubleshoot it by power-cycling the device by unplugging its USB cable.

If the AP is present but Firestorm is not available, it can be connected to with a monitor and keyboard. The credentials are username: `pi`, password: `disorient`. Firestorm is a NodeJS server running as a systemd service. Logs for Firestorm can be read via: `journalctl -u firestorm`.

Alternatively, disconnect the existing Raspberry Pi and connect another one included with the sign. Each Raspberry Pi has been flashed with identical contents to its SD card so they should be interchangeable in the case of hardware failure.