

DR-10-CV94 Operating Guide

DESIGN GOALS

This is a tile-able design to drive a 25-pixel matrix of Cree CV-94 LED clusters. These are typically used in outdoor and longer distance applications, so the visual separation of the R,G, and B LEDs is not noticeable. The board is laid out such that four can be clustered together and maintain a regular pixel spacing. The onboard driver is I2C addressable but not pre-programmed so an external controller is necessary. Example code for the Arduino environment (easily portable to other frameworks) on a Raspberry Pi Pico (RP2040) is available.

HOOKUP

LED power is supplied on the terminal block (J2). This is a 4.5-5.5V input.

Controller power is supplied on J1, as marked. This is the I/O voltage and should be supplied by the external controller (ie, the Pi Pico PCB).

Jumpers JP1-JP4 configure the I2C address (8-bit, unshifted) of each board:

Jumper	I2C Address
JP1	0xA0
JP2	0xAA
JP3	0xB4
JP4	0xBE

OPERATION

The external controller must be programmed initially; the instructions below are for our demo firmware.

1. power to matrix board first, 5V nominal (demo is jumpered to USB power)
2. plug in rp2040 board to USB
3. there is a 5 second delay to allow host OS to enumerate USB before starting the array
4. a COM port should show up
5. solid green LED on rp2040, flashes if I2C doesn't configure the board (check connections)
6. led matrix should be all-on
7. Run terminal emulator (puTTY, Tera Term, etc) at 115,200 baud 8,N,1 connected to the serial port assigned in step 3a

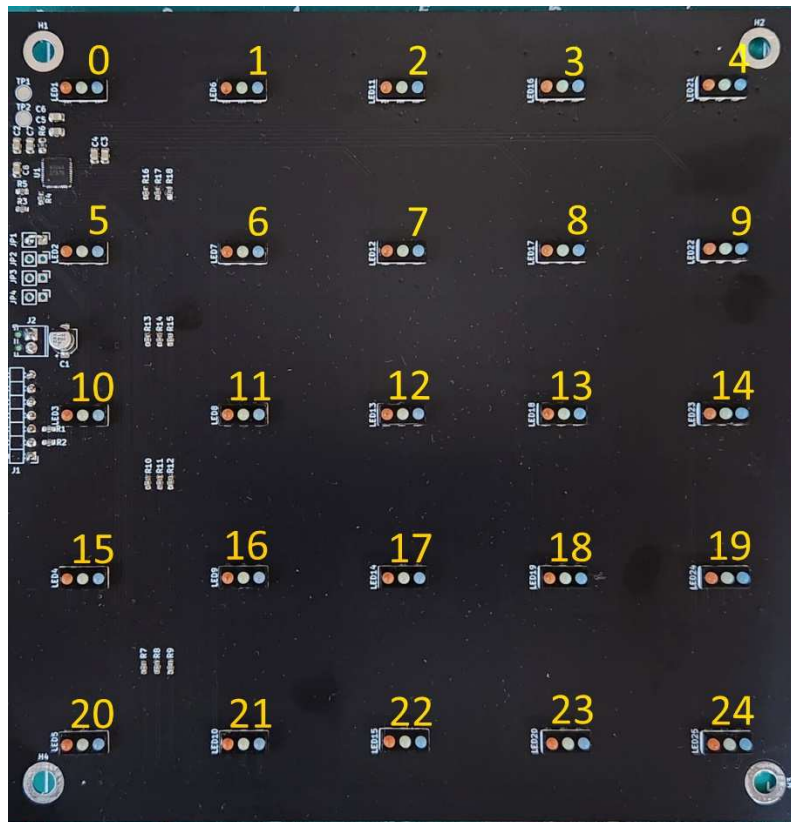
COMMAND INTERFACE

All commands are three characters followed by a value, then press [enter]:

- `cur nnn` – set global current, scale from 0-255
- `red nnn` – set all red pixels, PWM value 0-255
- `grn nnn` – set all green pixels, PWM value 0-255
- `blu nnn` – set all blue pixels, PWM value 0-255
- `pix nn rrr ggg bbb` – set a pixel (0-24) at RGB values given
- `rbw n` – set rainbow mode, 0 (off) or 1 (on)

Successful commands will echo back to the terminal, unknown commands will error.

Pixels numbered as shown:



FIRMWARE NOTES

The IS31FL3737.c/.h API is lightly edited from Lumissil's example code. I2C transactions via this API are wrapped around the Arduino Wire library in I2C.c/.h. Porting to another framework should therefore be started in the I2C interface.

IS31FL3737RES.hpp contains a lookup table to map the IS31FL3737 PWM registers to the numbered pixels. In an effort to keep the PCB fanout simple and the stackup to 2 layers, pins were chosen for the most convenient layout. Thus, the pixel mapping is not fully sequential with the PWM registers.

Some example usage is included in the application code in `IS31FL3737_Controller.ino`. Methods to change individual pixels and entire buffers are shown. Low framerate video may be possible using the buffer writes, but is still somewhat limited by the 400kHz I2C interface.

A simple “rainbow” function is included to show a constant brightness color sweep. An indexed value through the HSV color space is converted to RGB values and written to the pixel buffer.