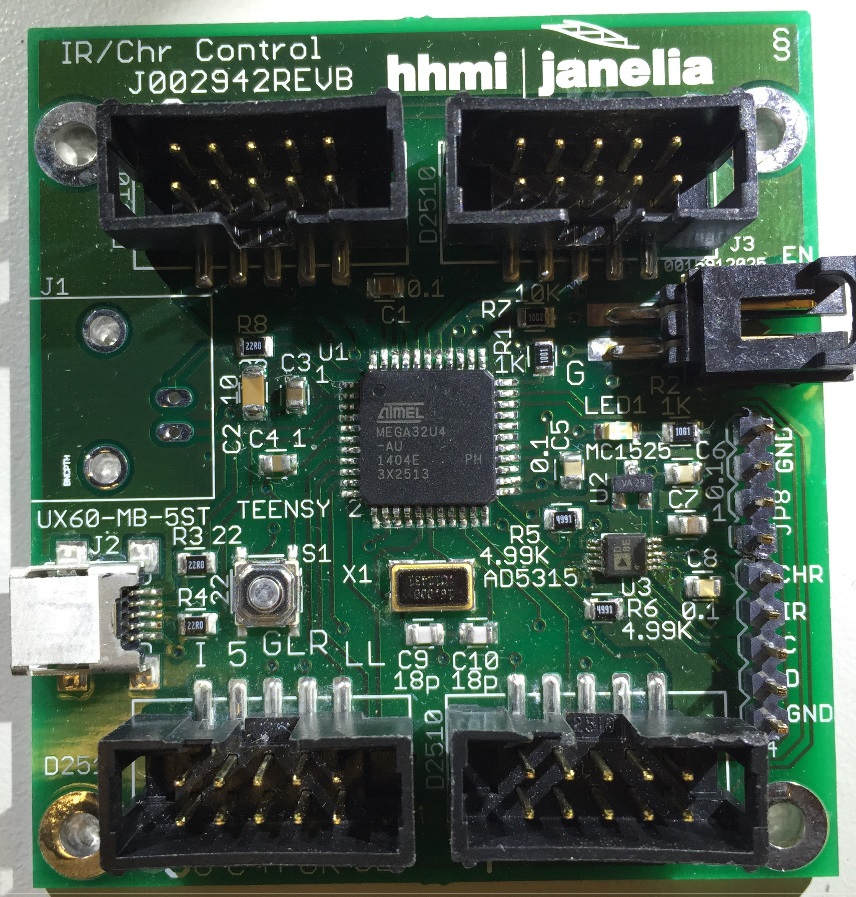
IR-Optogenetic LED Panel Controller

Vault Folder: Fly Bowl LEDs

****

Steve Sawtelle  
Lead Electrical Engineer, ID&F  
(571) 209-4158

Contents

[System Overview 2](#_Toc430158833)

[Hardware Development 2](#_Toc430158834)

[Connectors 3](#_Toc430158835)

[Firmware Development 4](#_Toc430158836)

[Command Set 5](#_Toc430158837)

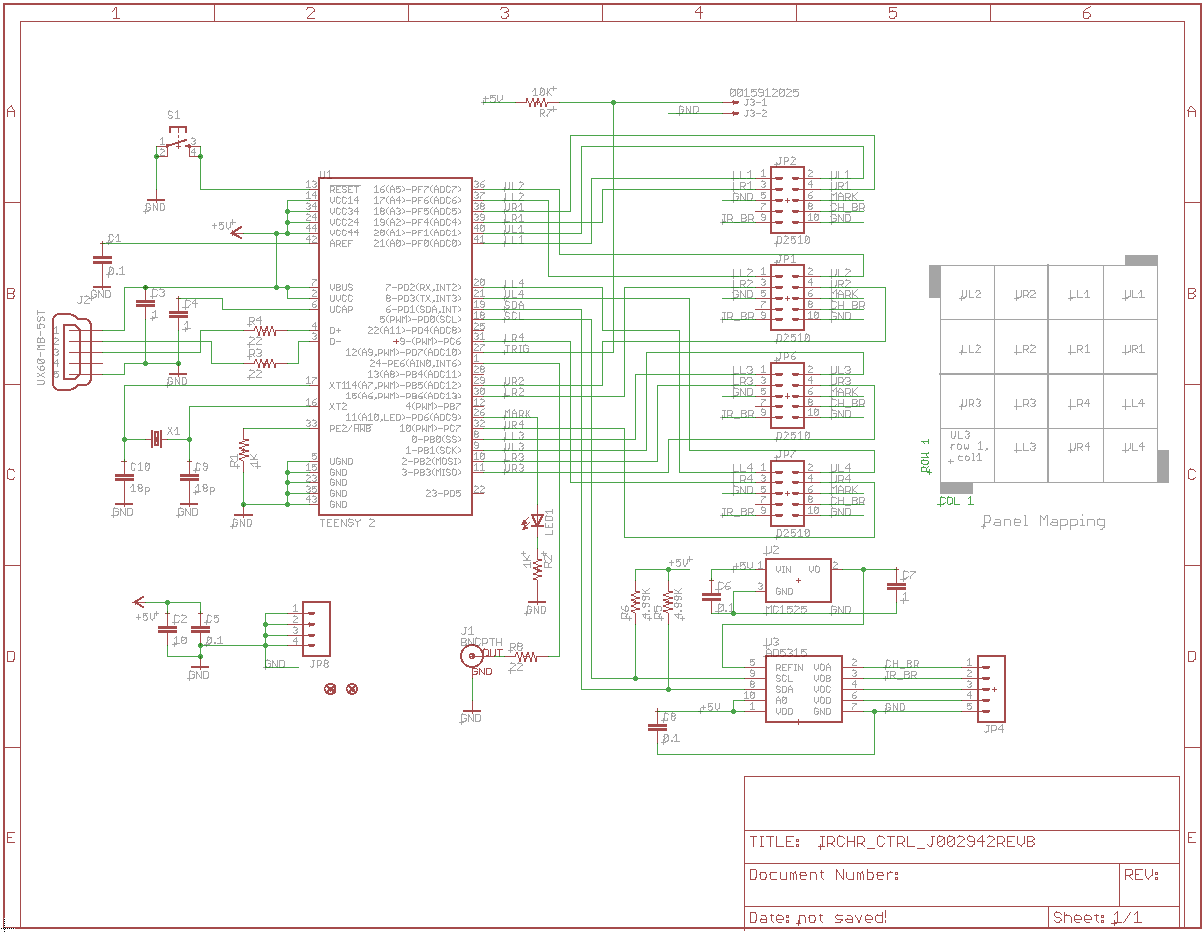
# System Overview

The IR-Optogenetic Panel Controller is used to control light intensity, pulsing, and pattern generation for up to four LED IR (Infrared) backlight/optogenetic panels. The controller utilizes USB-serial communications to the host to receive commands and return status. A pulse pattern can be downloaded and run by the controller with timing independent from the host to allow precise pulse generation.

# Hardware Development

The board incorporates a Teensy 2.0 processor to make code updates and changes easy. Four ten-pin ribbon cable connectors are used to connect to up to four LED panels. The IR intensity and optogenetic intensity (0-100%) can be set independently, but are each common to all boards. IR on/off control is accomplished by setting the IR intensity to zero. The optogenetic LEDs are arranged in four quadrants on each panel, with independent on/off control. In this way, up to sixteen different areas of light activation can be turned on and off independently. Each panel connector also has a Marker LED that can be used to drive an IR LED off to the side of the arena so that optogenetic LED status can be monitored by an optogenetic-blocking camera. The board has four mounting holes on 2” centers which facilitates attaching the board to a standard one-inch-center optical breadboard.

*Schematic*



*Printed Circuit Board*

See project file

*Materials*

See project file

# Connectors

**J1** BNC

Available for custom use as input or output.

**J2** Mini-USB

Connects to host computer for control and firmware updates

**J3** – 2 pin latching header (TE Connectivity 103635-1; Mates with Receptacle 104257-1 and terminals 1-104480-5)

Connection for either trigger in or sync out.

**JP1, JP2, JP6, JP7** – 2x5 ribbon cable header

These headers connect up to four panels to the controller.

**Pinout**

|  |  |
| --- | --- |
| Pin | Function |
| 1 | Optogenetic lower left on/off |
| 2 | Optogenetic upper left on/off |
| 3 | Optogenetic lower right on/off |
| 4 | Optogenetic upper right left on/off |
| 5 | Ground |
| 6 | Maker LED out |
| 7 | +5V from panel (not used on this board) |
| 8 | Optogenetic brightness control (0-2.5V out analog) |
| 9 | IR brightness control (0-2.5V out analog) |
| 10 | Ground |

# Firmware Development

The firmware is written under the Arduino programming environment with Teensy extensions added in. There are three compiler defines that affect the operation of the firmware:

USE\_SCALING - normally undefined

If undefined, the brightness of the optogenetic LED is set in a linear fashion. If defined the brightness is set with pseudo-log scaling, such that up to 40%, the LED brightness increases in a small step linear fashion, and above 40%, it increases logarithmically.

LINEAR\_DRIVER – normally defined

This should be defined when used with the current panel design. Earlier panels used a PWM brightness control for the optogenetic LEDs and required an offset in the brightness setting.

USE\_EXT\_TRIG - normally undefined

If defined, an external trigger can be used to turn the optogenetic LEDs on and off.

# Command Set

The host communications is via a USB-serial link. The Teensy serial driver must be loaded on the host (available from PJRC.com). The link is baud-independent. The communications protocol is based on an ASCII commands with various parameters. Commands are case insensitive and must be terminated with a carriage return character (‘\r’, 0x0D). Basic error checking is done and ‘cmderr n’ is returned with ‘n’ being a negative integer that can be used to determine the type of error. Commands may return other information to the Host to indicate status. These are still under development.

The following commands are supported:

**CHR p** – set optogenetic level to ‘p’ percent of full brightness

This does not turn LEDS on. If set to 0% LEDs will turn off, but it is recommended to use the TTL controls for that (via OFF or PULSE command). Very low levels (1 or 2%) may have more variability in individual LED brightness. NOTE: The command was named after ‘CHRimson’ optogenetic lighting as this was the original use. However, the panels can be assembled with a variety of LED colors, so the generic term ‘optogenetic’ is used in this document.

**IR p** – set IR level to ‘p’ percent of full brightness.

A setting below about 10% will turn off the IR LEDs. There is no separate TTL control. Values near 10% may have inconstant individual LED brightness. It is best to set brightness to 0% if the LEDs should be off. Use values above 10% to turn on the LEDs to a percent of full brightness.

**ON x, y** – turn on quadrant of optogenetic LEDs at position x and y

Each LED board is divided into four quadrants of optogenetic LED control. Up to four boards can be controlled, so there are sixteen quadrants total in a 4x4 arrangement. The board numbers are 1-4 in a rectangular coordinate system fashion (Quadrant 1 is to the upper right, 2 is upper left, 3 lower left, 4 lower right). If the boards are connected to the appropriate numbered connectors on the controller (also set up in the same order), then one easily select an LED (or groups of LEDs) to turn on. The control is row and column, with row 1 being the lowest and column 1 being the left most. So 1,1 references the lower left quadrant of board 3. Entire rows and columns can be changed at one time by setting the value to 0; so ON 1,0 turn on row 1, all columns. Setting ON 0,0 turns on all crimson LEDS, and is the general way the command is used when no patterns are needed.

**OFF x,y** – turn off quadrant of Optogenetic LEDs at position x,y

Turns off one or more quadrants of LEDs based on x and y; see the ON command for details.

**PATT bbbbbbbbbbbbbbbb** – download a Optogenetic ‘on’ pattern

The Pattern command allows an arbitrary pattern of LEDs to be set based on the 16 binary values ‘b’. ’0’ is off. This command does not turn on or off any LEDs directly. The pattern is used in the PUSLE command.

The bits control LEDs in the following pattern (row, column):

If the leftmost bit in the pattern is called ‘b15’ down to ‘b0’ for the rightmost bit:

|  |  |  |  |
| --- | --- | --- | --- |
| b3 | b2 | b1 | b0 |
| b7 | b6 | b5 | b4 |
| b11 | b10 | b9 | b8 |
| b15 | b14 | b13 | b12 |

**PULSE width, period, number, off, wait, iterations** - pulse setup command

The PULSE command sets up a pulse train to light Optogenetic LEDs that have been set up with an ON or PATT command. The pulse has on delay, pulse width, period, number of cycles, off delay, and number of iterations:

width: The pulse width (on time) in milliseconds from 1 to 30000

period: The period of the pulse (on time plus off time) in milliseconds from 1 to 30000. If the width and period are equal, then the LEDs are on constantly during the pulse train.

number: The number of pulses in a pulse train. One pulse train starts with an on time and ends ‘number’ of pulse later after the last off time.

off: The off time after the pulse train completes in milliseconds from 0 to 30000. This provides a means to have a ‘dead time‘ between pulse trains if iterations are used. Note that the total off time from when the last pulse on time ends to the next pulse on time starts is equal to: off + period – width.

wait: Delay the start of the pulse sequence in milliseconds from 0 to 120. This delays the start of the very first pulse after the RUN command is sent. It is not repeated again until RUN is resent.

iteratations: The number of time the pulse train is repeated from 0 to 30000. A ‘0’ value runs the pulse train continuously until a STOP command is sent.

**RUN** - Start running the pulse sequence

**STOP** - Stop running the pulse sequence and reset to the beginning

**???** - Returns the version number and other parameters