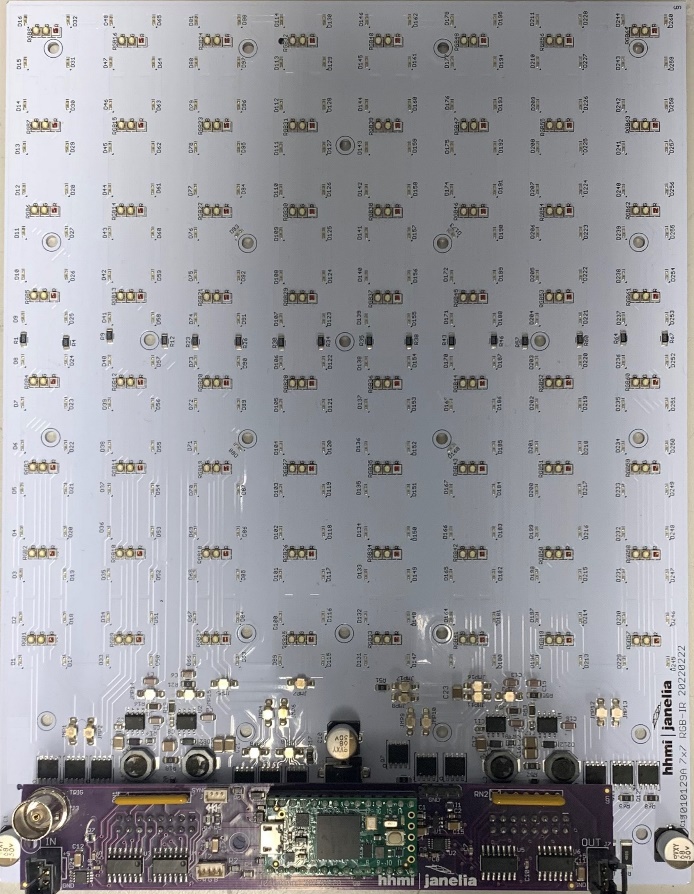
7” x 7” RGB-IR LED Panel

J0010129 and J010130

****

Steve Sawtelle  
Electrical Engineer, ID&F  
(571) 209-4000

Contents

[System Overview 2](#_Toc151466519)

[Hardware Development 2](#_Toc151466520)

[Connectors 3](#_Toc151466521)

[Setup and Deployment 4](#_Toc151466522)

[Command Set 6](#_Toc151466523)

[Firmware Description 6](#_Toc151466524)

# System Overview

The RGB-IR LED Panel is used to provide three colors and IR lighting of various intensities. The standard design uses red, green, and blue (RGB) LEDs, but since separate LEDs are used for each color, any three colors can be used that are available in the Luxeon C series palette. A built- in micro-processor controls the panel via a USB-Serial interface. This allows the panel to directly control pulsing of the LEDs, offloading the task from the host computer, and ensuring accurate timing. The panel is divided into four quadrants that can be independently set up with different colors and intensities. The lit area of the panel is 7” x 7” (122mm x 122mm). The overall size of the panel is 6” x 6” (152.6mm x 152.6mm). Four panels can be combined to form a lit area of 9.6” x 9.6” (178 mm x 178 mm). These panels can be daisy-chained so that all panels can have synchronized control through a single USB connection. A three-pin connector can be used to drive an indicator LED and/or perform sync in and out functions. A four-pin connector provides individual marker signals for the RGB light. A five-pin connector provides four digital outputs. The hardware is divided between two printed circuit boards; the LEDs and power circuit are located on an aluminum-backed board to provide optimal heat sinking and the control electronics are located on a plug-in, standard, printed circuit board.

# Hardware Development

The board incorporates a Teensy 4.0 processor with code developed under the Teensyduino environment to make code updates and changes easy. The RGB LEDs are controlled via linear drivers to provide constant levels with no PWM artifacts. The linear control driver is through a standard FET with a low-side current sense resistor driving the feedback control via an op-amp. Since there are two series strings of LEDs in each quadrant, a dual, matched FET is used to passively regulate the second string. Using linear regulators does increase power dissipation, which should be low if the LEDs are pulsed at low duty cycles. The IR LEDs are PWM controlled to lower heat dissipation, as they are typically always on. The IR PWM artifacts are also out of the detection range of most test animals. Multiple panels are controlled by a serial communications link that is daisy-chained from board to board. The RGB LEDs are on 0.6” (15.24mm) grid and the IR LEDs are on a 0.3” (7.62mm) grid.

*Schematic -* See project file

*Printed Circuit Board* - See project file

*Materials-* See project file

# Connectors

**J1**– (located on LED board) power barrel 2.5 x 5.5 mm

Power In – 28VDC @ 5 Amp minimum

**Mini-USB** on Teensy

Connects to host computer for control and firmware updates.

**J1** – 3 pin latching, Molex 1.25mm connector (533984003), marker and sync I/O

1. Teensy pin 20 through 332 ohms, intended to drive a marker LED.
2. Teensy pin 19 through 100 ohms, intended for a sync input
3. Ground

**J2** – 4 pin latching, Molex 1.25mm connector (533984004), marker LEDs connection

These markers are active when the corresponding color LED is active. When a pulse command is being run, the marker ‘envelopes’ the pulse train. These marker LEDs are generally IR LEDs to allow a camera with an IR high pass filter to record when the color LEDs are active.

1. Red marker output through 332 ohms, intended to drive a marker LED.
2. Green marker output through 332 ohms, intended to drive a marker LED.
3. Blue marker output through 332 ohms, intended to drive a marker LED.
4. Ground

**J3**– BNC, Sync Output

**J7 and J8** – 3 pin TE Connectivity 0.1” center (3-647167-3), Board daisy chain

1. ground
2. TX/RX
3. +5VDC

# 

# Setup and Deployment

LED Intensity

Power Supply

A power supply with a rating of 28 volts is required. The current required depends on which LEDs are illuminated and at what maximum intensity. A single color, at 100% intensity, will draw about 2 amps. The IR at 100% intensity will draw about 0.8 amps. Multiple boards can be run from a single power supply, but it must supply the total current needed by all boards. The Cincon TRH100A280-12E13-Level-VI power supply can supply 28V at up to 3.54 volts and can be plugged directly into the RGB board. A single board, with a power requirement of less than 1.2 amps, could use an XP Power ACM36US30, which also can plug directly into the board.

Mounting and heatsinking

The panel needs to be mounted on a minimum 7” x 9” (178 mm x 229 mm) heat sink. The mounting screws (4-40 or 3mm) are on 1” x 1” (25.4 mm x 25.4 mm) grid. This allows the board to mounted on a Thorlabs Imperial dimensioned breadboard. Adapters from the 4-40 (or 3mm) to ¼-20 threads will be needed. A diffuser positioned at least 0.5” (12.7mm) above the panel will provide uniform IR lighting over the lit area. The panel runs off a single 28-volt power supply, although up to 36 volts can be used if duty cycles are low or good heat sinking is employed. At lower duty-cycles the major heat source is the linear regulators (12 large flat ICs), especially the red drivers. If continuous RGB illumination is used then the LEDs become the major heat source. In general, at low illumination and/or duty cycles, a standard breadboard can be used. At higher power levels and longer on times, a liquid cooled breadboard will be needed along with a cooler such as a Koolance EXT-440CU computer cooler.

Panels and Quadrants

The firmware and the 5” x 5” RGB board has four quadrants, each of which can be set to different colors and intensities. The 7” x 7” does not have this feature (though it uses the same firmware).

The 7” x 7” does allow for multiple boards (panels) which can be combined to make a larger illuminated area, either 7” x 28” or 14” x 14”.

The Host USB connection is made to Panel 1 only. Cables are used to daisy-chain Panel 1 to Panel 2, Panel 2 to Panel 3, and Panel 3 to Panel 4. When the RESET command is set, a command is sent along the daisy-chain to enable each panel to understand where it lies in the arrangement. The daisy-chain setup can also be used if to control four separate rigs that only need one panel each. This is especially useful if the four rigs will run the same protocol.

# Command Set

See the separate document: RGB-IR LED Panel Command Set.docx for descriptions of the commands.

# Firmware Description

This section describes, in general terms, how the firmware works.

The firmware is written in Arduino with Teensyduino extensions. Several libraries are used:

Cmd.h – This has been modified from the version available on Github to add features needed by the firmware. It processes commands sent over USB-Serial and is used to set up parameters and create actions.

Watchdog.h – This is a library written by Peter Polidoro to set up and control a watchdog timer in case the firmware hangs up.

EEPROM.h – This library is used to save setups in on-volatile memory, such as offsets and gains for each quadrant to make the quadrants light intensity more even (especially at low levels).

Several Teensy varieties are supported. The preferred boards are the Teensy 3.2 and 4.0. The LC can be used but has limited memory for Experiment Command Steps. The firmware will compile correctly based on the chosen processor.

The RGB firmware is command driven. The processing of commands is handled by the cmd.h library, which will call the appropriate function once the command line is parsed. These functions end in ‘cmd’. The commands can set up parameters, query settings, or cause action. Commands are ASCII and case insensitive. Parameters are separated by spaces or commas. Each command is terminated with a linefeed and carriage return.

The timed functions of the board are based on a one millisecond timer, so the basic accuracy of the board is one millisecond. The processing of timing functions is controlled via a four-phase state machine, one phase for each color and a fourth for digital outputs. An initial state is called, which sets up parameters and timing for the next state. A flag is set to note a change in state and the next state will wait until the last state has completed before it executes its function and sets up the next state.

There are two main timed functions: Pulse and Experiment:

The Pulse function is used to create a train of pulses at a set rate. See the PULSE command for details. Each color (and digital output) can have its own pulse setup and they can run concurrently.

The Experiment function can be used to create an entire experiment setup consisting of various pulse setups that can be varied over an experimental run. This is especially useful in complex experiments as all the timing can be offloaded to the RGB Board from the host.