Adaptive Lighting

ECE 445: Senior Design, Team 17

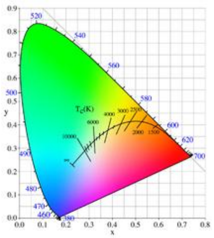
Jered Greenspan, Madhav Khanna, Sichao Wang
TA: Ryan May

Introduction

- Objectives
- System Overview
- Hardware Modules and Testing
- Recommendations
- Future Work
- Ethical Considerations

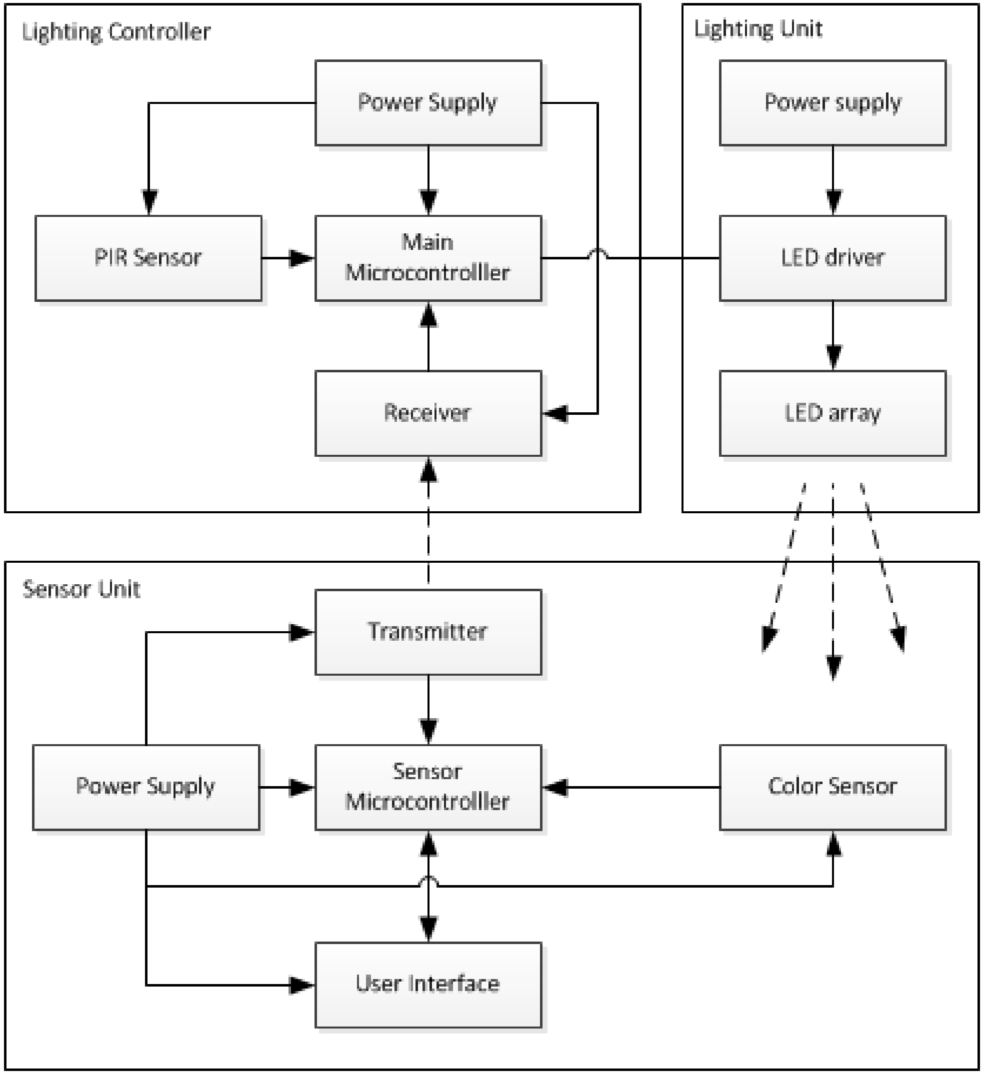
Objectives

- Goal: An adaptive lighting system in which feedback from a color sensor is used to drive an RGB LED array to achieve the CCT and brightness set by the user.
- Benefits:
  - Reduces overall lighting consumption
  - Maintains a lighting environment that is
    - Constant – even with changing environment
    - Set by user – more pleasing lighting experience

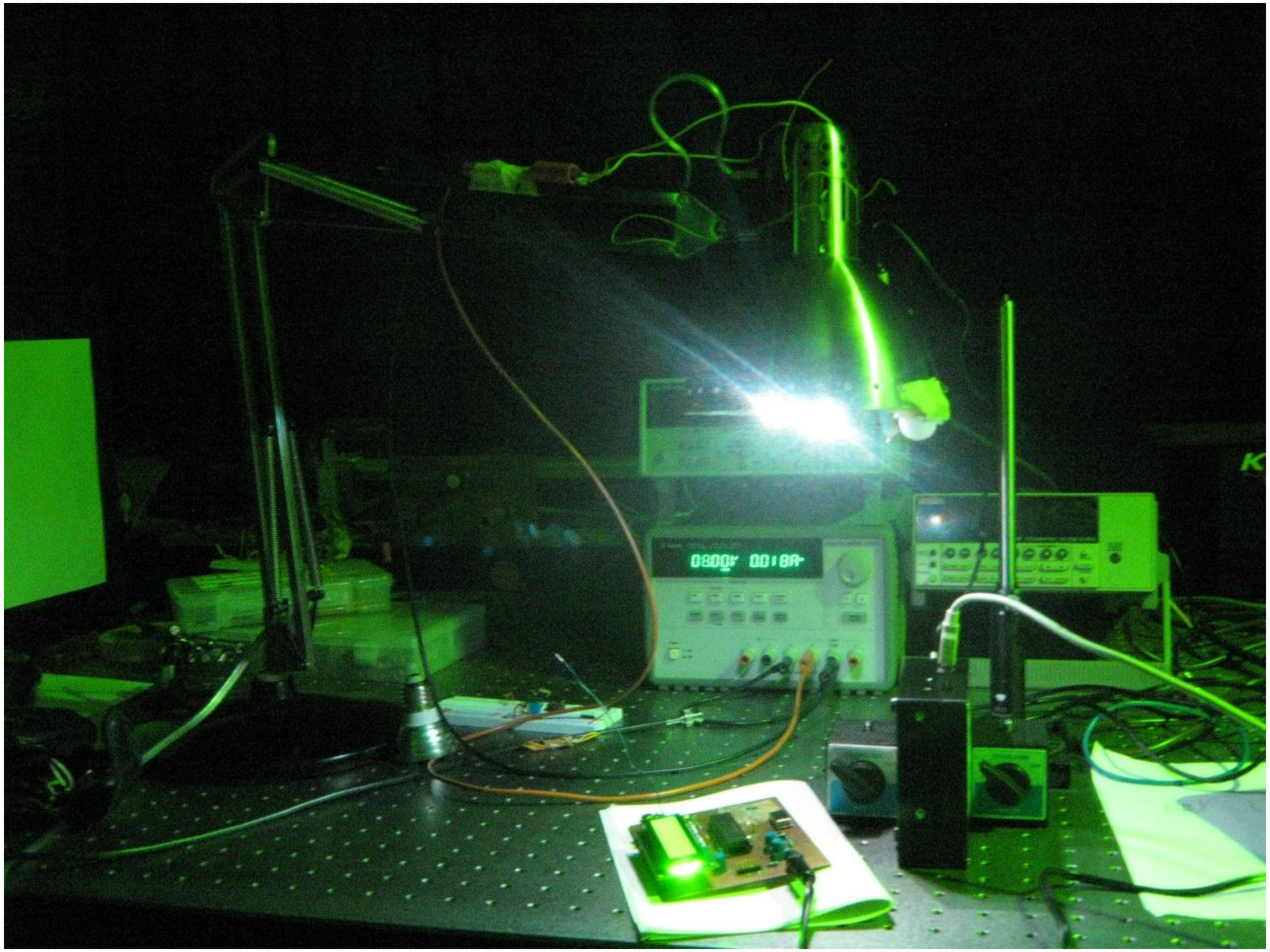


System Overview

- Sensor Unit
  - Manages user interface
  - Reads color sensor
  - Transmits sensor data, and user setpoints
- Lighting Controller
  - Receives data
  - Control algorithm
  - Outputs PWM signals
- Lighting Unit
  - Provides constant current to drive LED array from AC input



Lighting Unit



LED Driver

- The LED driver circuit involves the following:
  - Bridge rectifier to convert 120VAC to 160VDC.
  - Controller chip which we operate as a buck converter
  - The controller will facilitate PWM dimming for our high power LED's.

LED Features

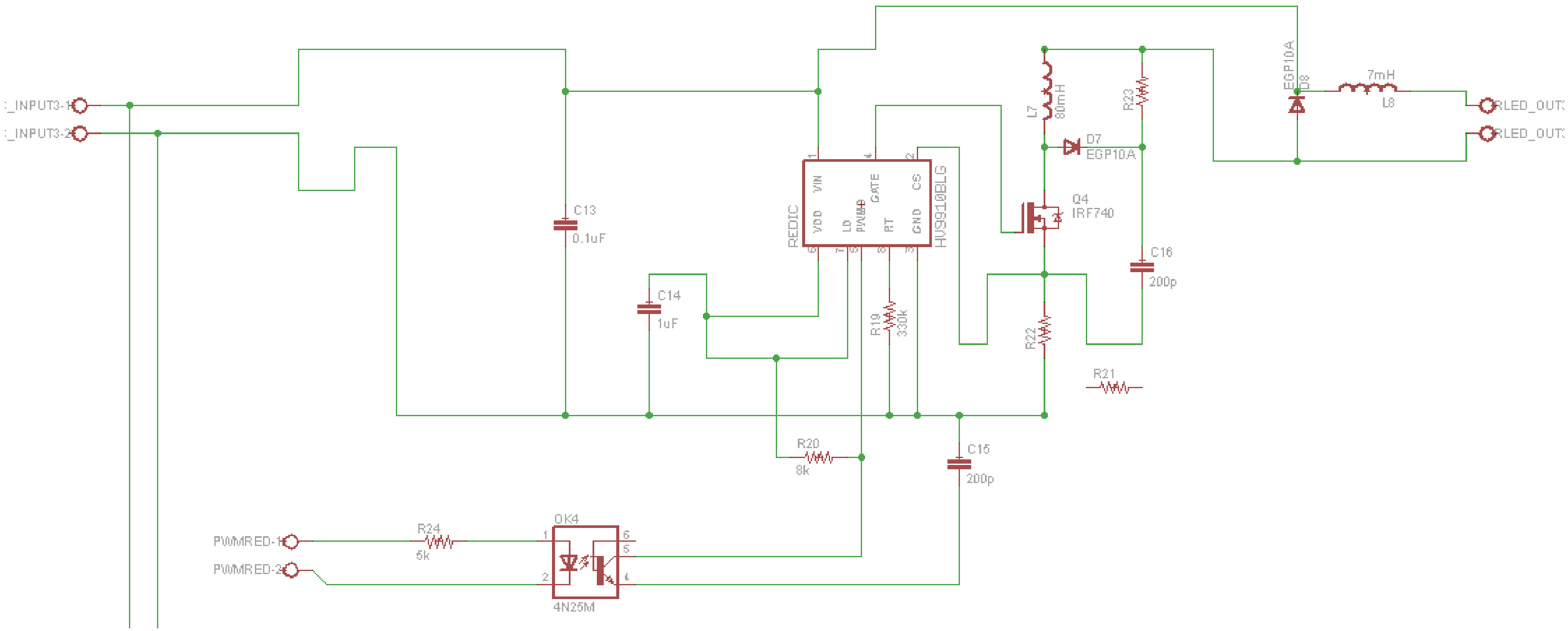
- The LED driver circuit drives 9 high power LED's.
- Each of the green, blue and red channels in the LED's are driven by individual drivers.
- LED Characteristics:
  - 3 W LED's.
  - Forward Voltage:
    - Red – 3.4 V ; Blue/Green – 2.5V
  - Forward Current:
    - Red – 330mA; Blue/Green- 300 mA

LED Driver Specifications

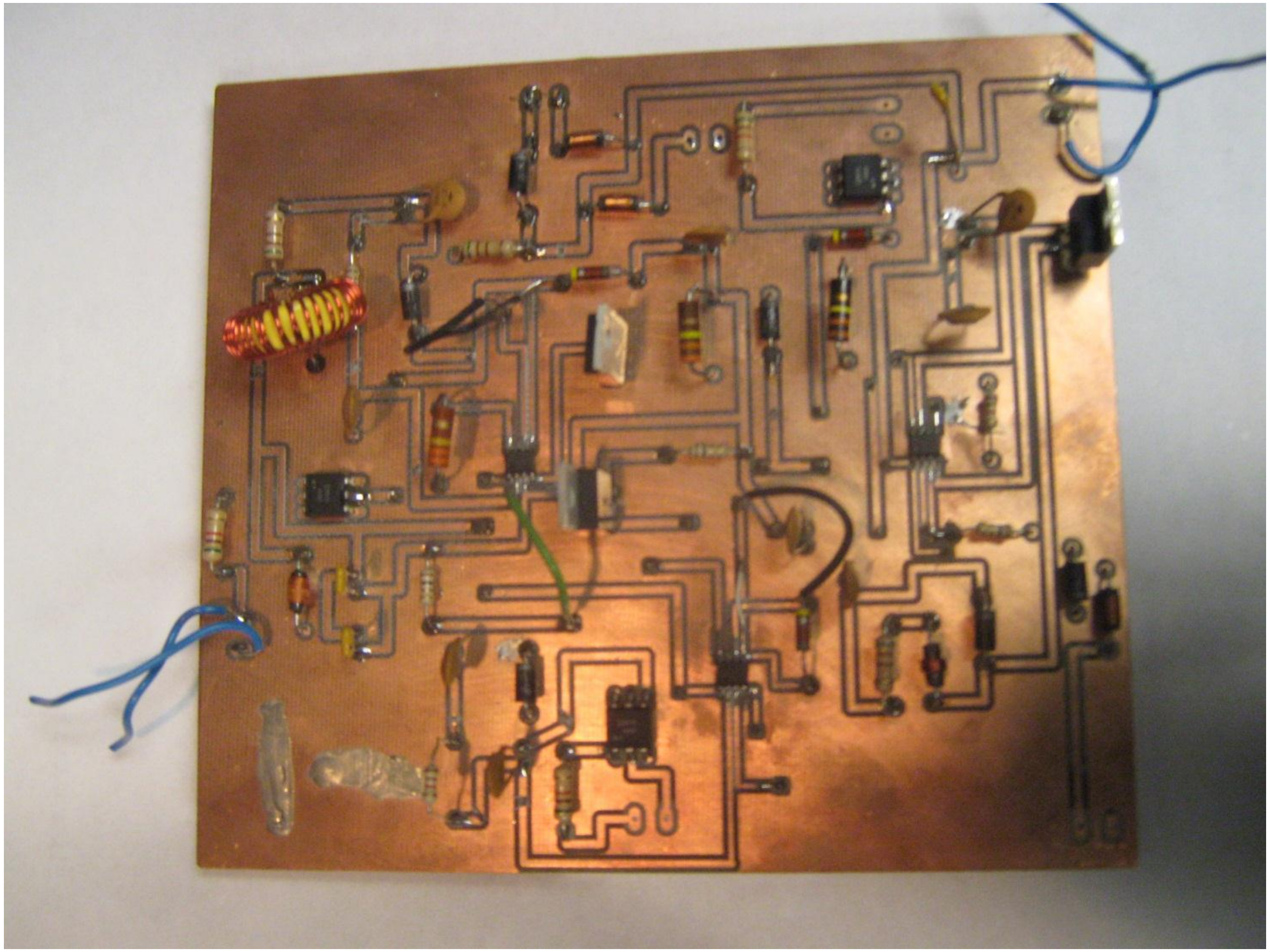
Driver consists of:

- Bridge Rectification circuit: Converting 120VAC to 160VDC.
- Buck Converter Circuit
- Lossless snubber circuit to take care of current spikes in the LED’s.
- Current across the LED driver circuit is controlled using a current sense resistor and a PWM input from the microcontroller.
- Increases PWM switching frequency from 500 Hz to 80 kHz

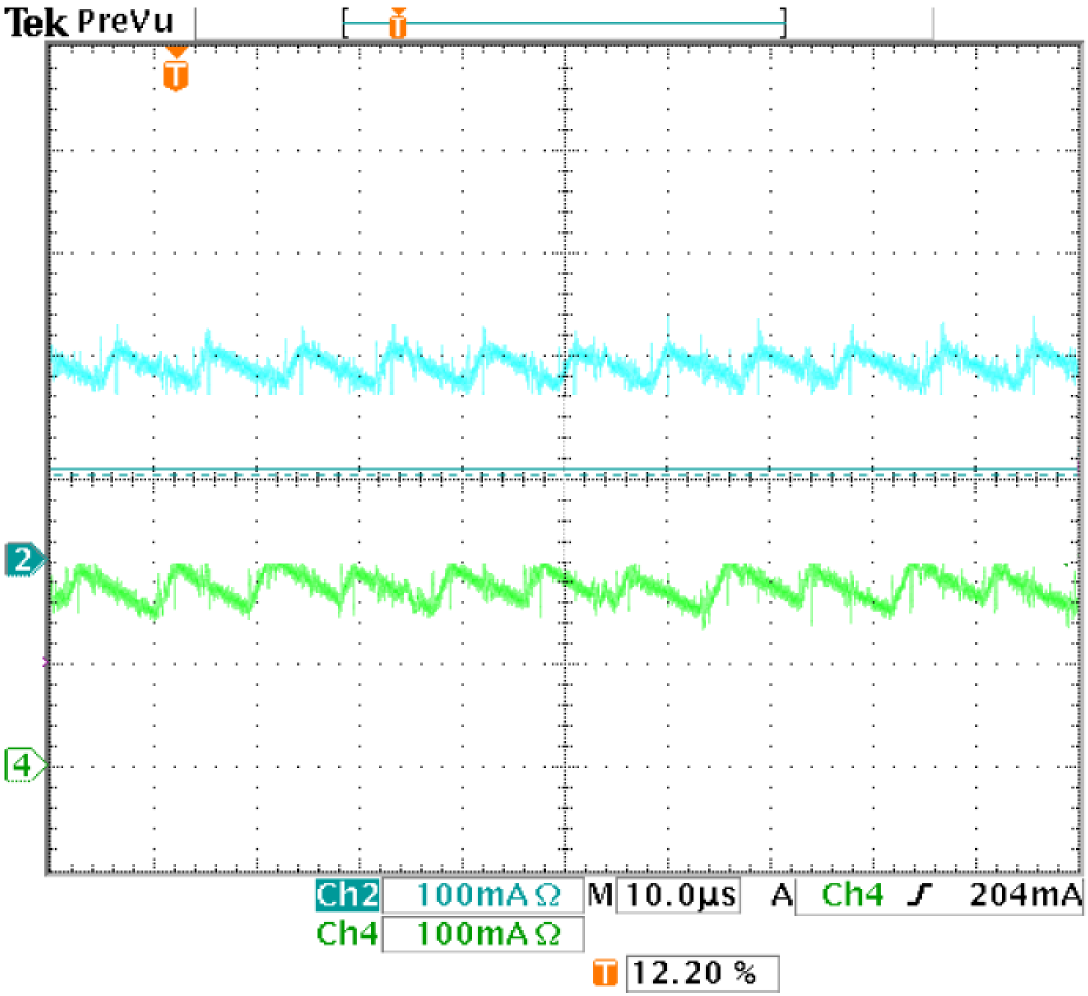
LED Driver


_INPUT3-1
_INPUT3-2
C13
0.1uF
C14
1uF
REDIC
HV9910BLG
VIN
VDD
LD
PWMD
GATE
RT
CS
GND
R19
330k
R20
8k
C15
200p
L7
80mH
R23
D7
EGP10A
Q4
IRF740
C16
200p
R22
R21
EGP10A
D8
7mH
L8
RLED_OUT
RLED_OUT
PWMRED-1
PWMRED-2
R24
5k
OK4
4N25M

LED Driver

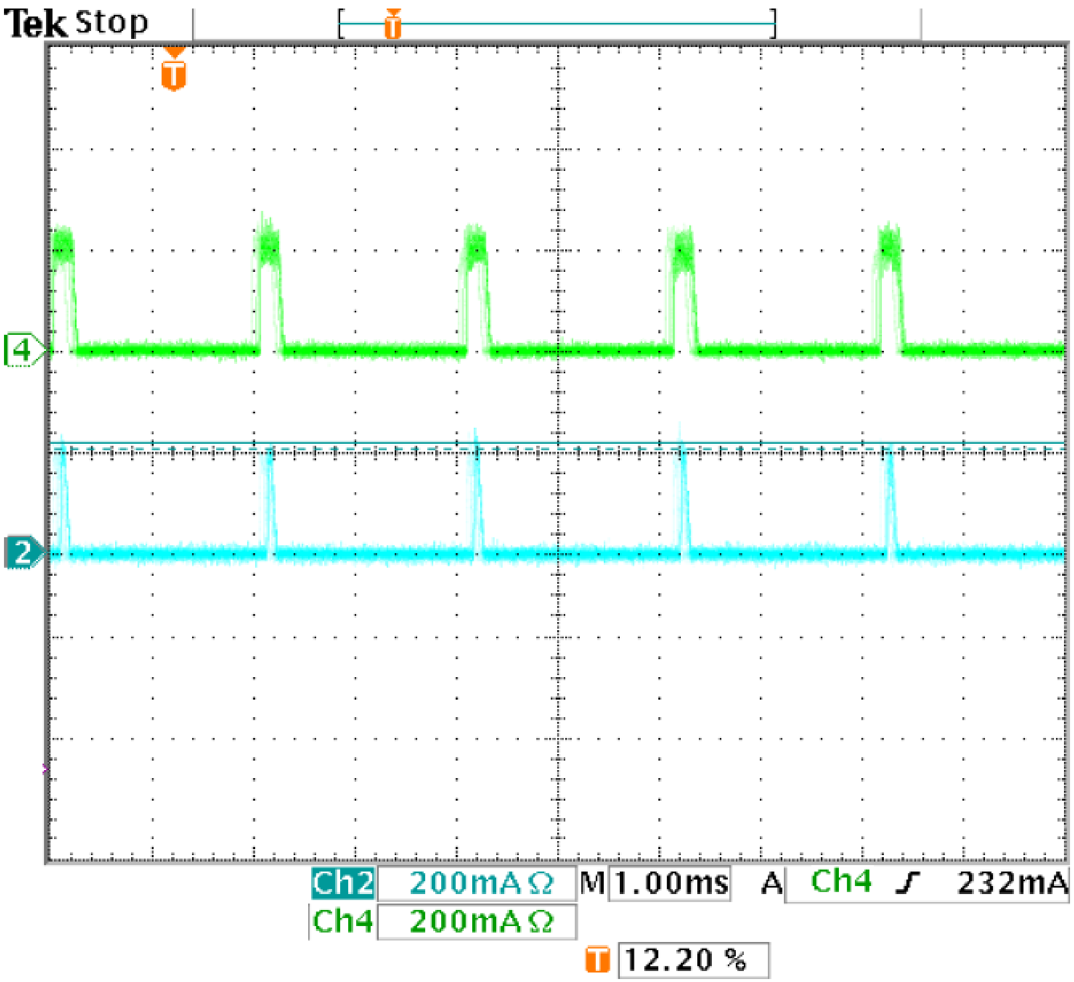


Driver Testing



LED Current in driver after snubber.

Driver Testing

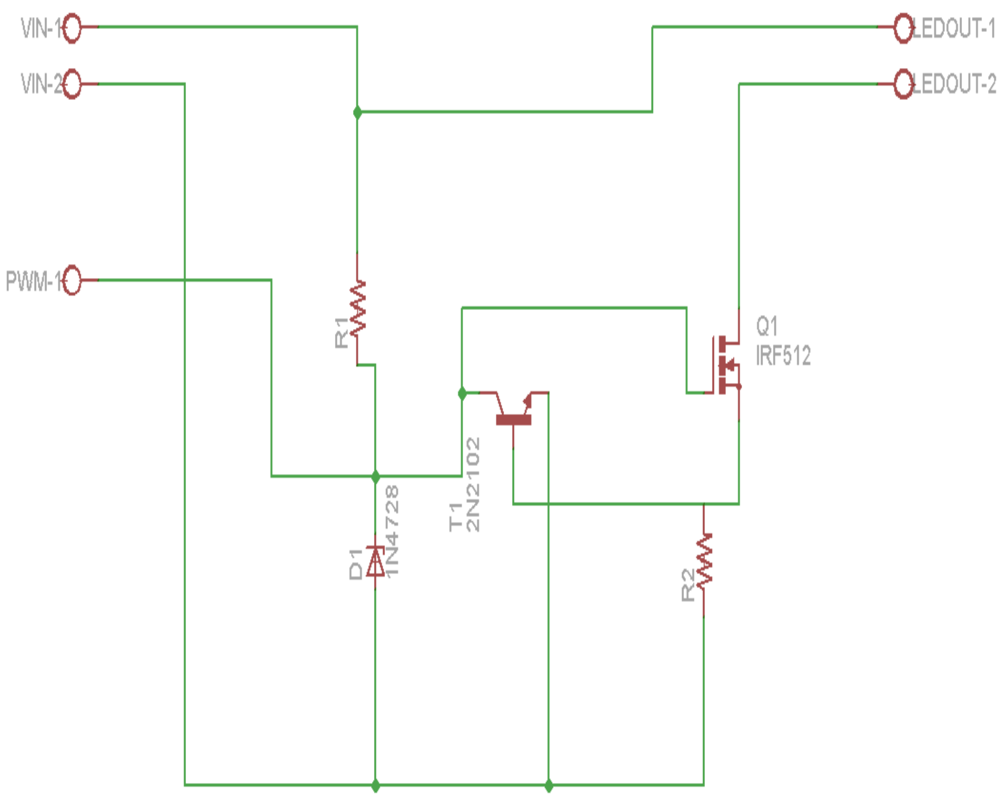


LED Current with PWM dimming

LED Driver: Challenges

- The LED driver had worked exactly according to specifications on our breadboard.
- During our PCB design we had mistakenly connected the snubber capacitor to the current sense pin on the LED driver chip.
- The fast current sensing pin detected a capacitance and began oscillating.
- These oscillations caused the internal circuitry of the chip to malfunction.

Replacement Circuit

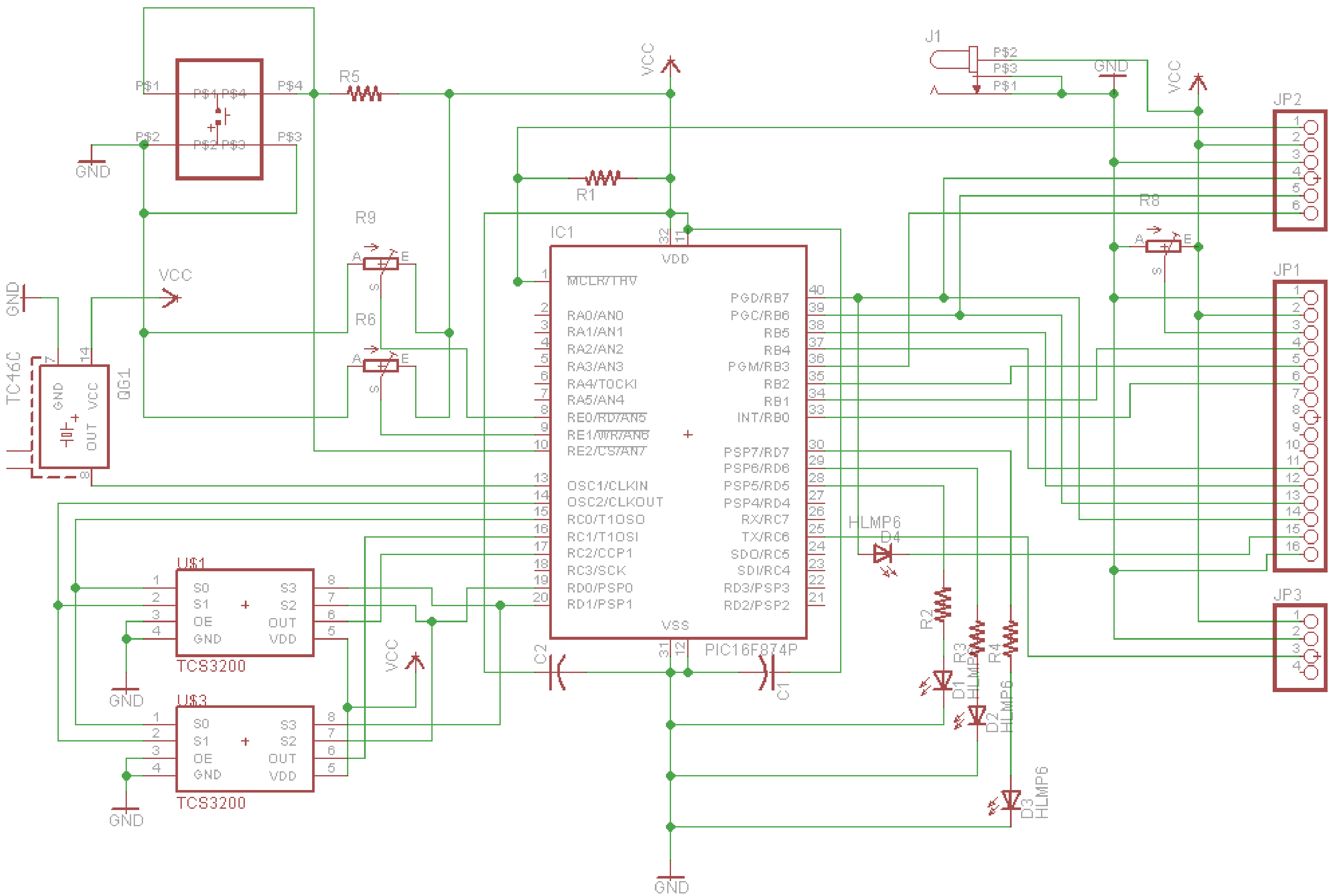


- A current sense resistor R2 to limit current across LED's.
- A NPN transistor and a NFET.
- Zener Diode to regulate voltage.

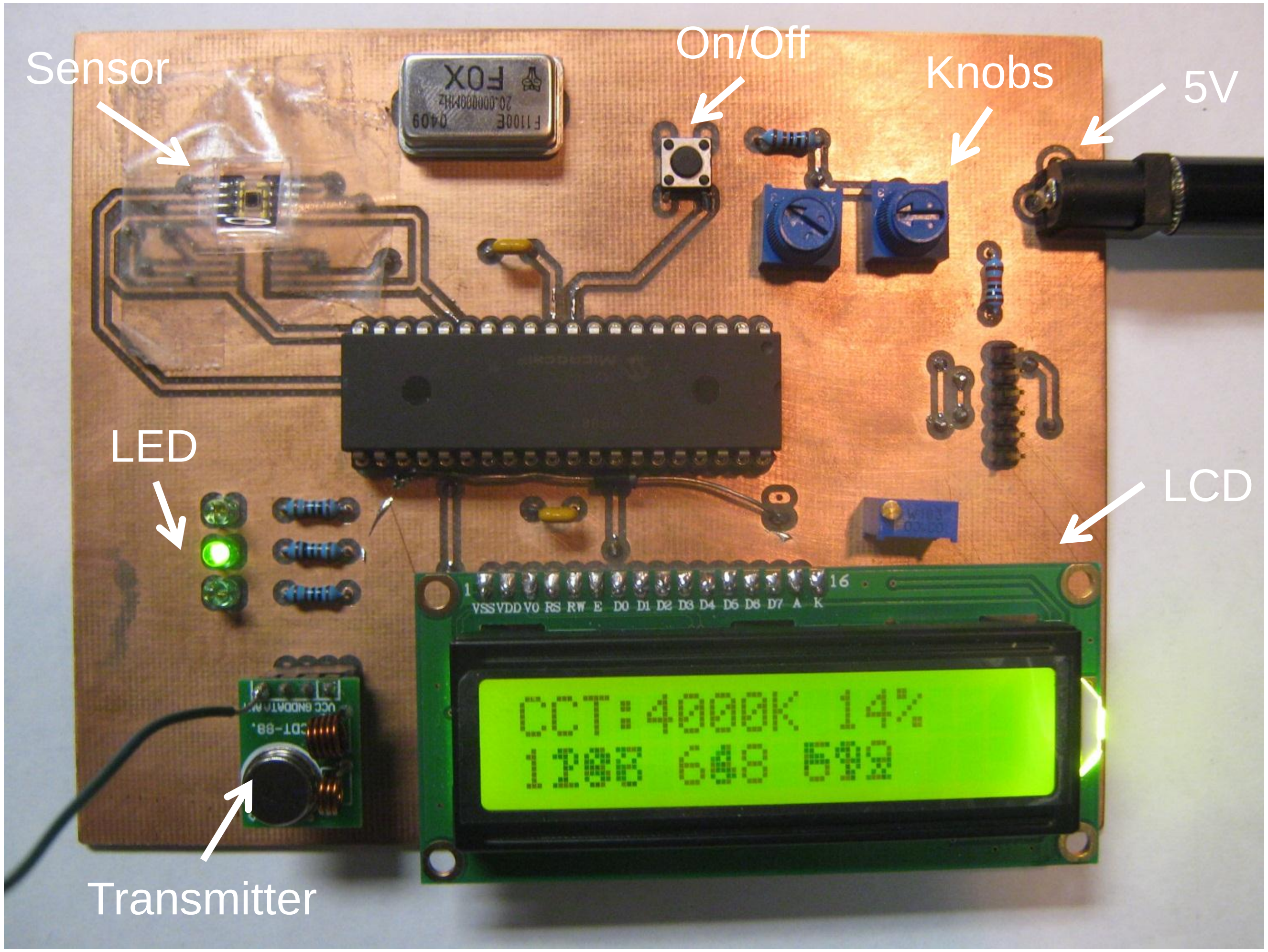
Sensor Unit

- User interface
  - Adjust CCT and brightness setpoints
- Reads color sensor using frequency capture
- Wirelessly sends packet over UART using RS232
- All supplied at 5V

Sensor Unit Schematic



Sensor Unit



Color Sensor

- 4x4 array of R,G,B,C photodiodes
- Frequency output: 2Hz-150kHz
- Output frequencies captured by microcontroller

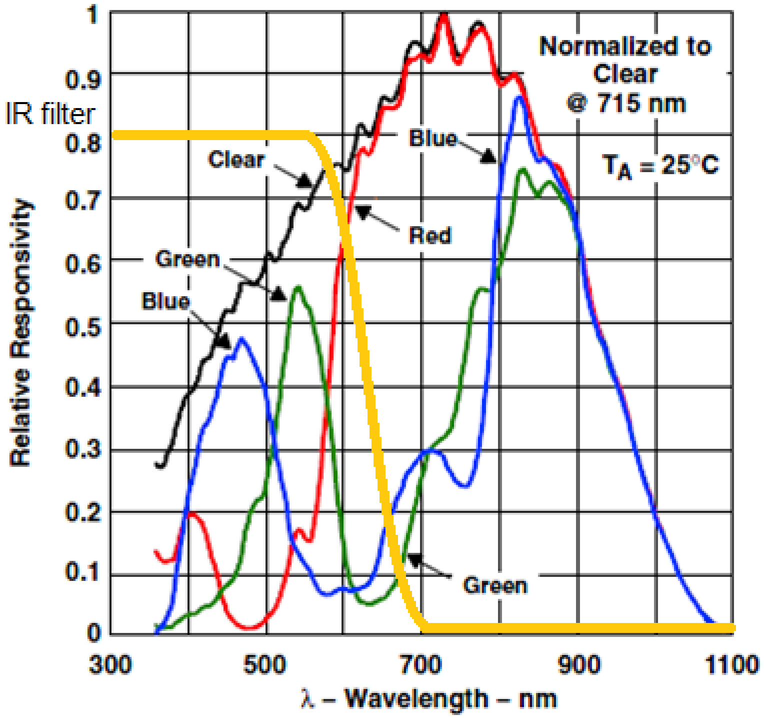
TAOS TCS3200

Challenges of Color Sensor

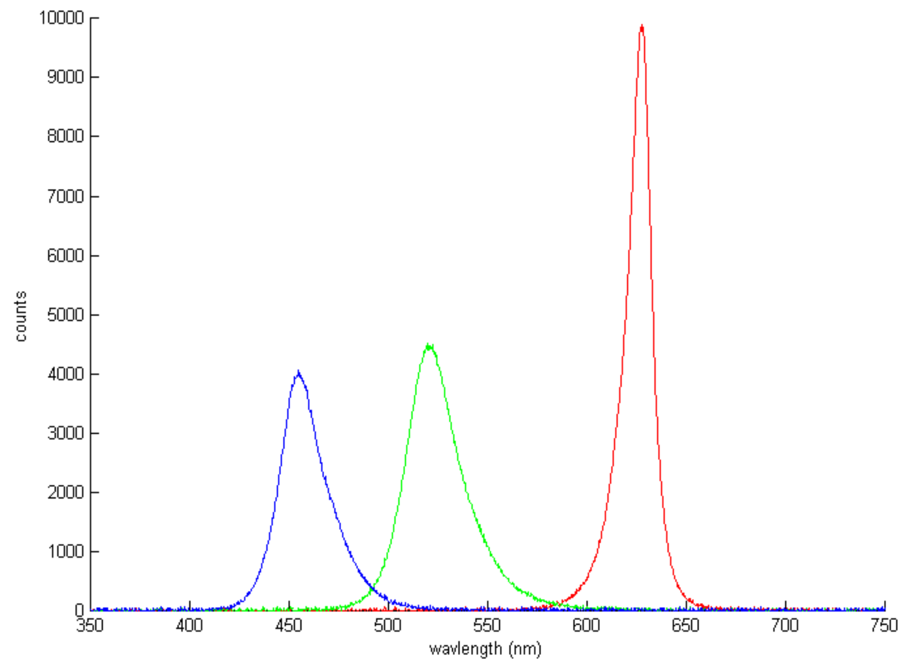
- Frequency output can be unstable, causing oscillations in control
  - solved by using lower resolution and powerful LEDs for a wider frequency output range
- Each color channel sensitive to nonvisible spectrum. Covered sensor with IR-block filter
- Sensor is sensitive to the angle of incident light
- Nonlinear in dark condition (SNR low)

Color Sensor Spectrum

Color Sensor Response



Spectrum of LEDs



Peak response	Sensor	LED
R	630	630
G	525	522
B	460	455

Sensor Microcontroller

- PIC16F887
- Captures RGB frequencies, collects user settings, constantly transmits serial RS232 data at baud rate=1000 through transmitter

- Wireless transmission frequency is dependent on the sensor output frequency
  - 2 Hz in dark -> 0.5 s x 16 x 3 chan. = 24s

Frequency Capture

- Using the internal timer of the PIC 16, we measure the time it takes between 16 periods of the output pulse of the sensor.
- Calculate the frequency by dividing by the period (with appropriate scaling).
- Initially, tried a software timer to count frequency, but was slow and narrow range

User Interface

- 16x2 Character LCD Module
- Displays the correlated color temperature (CCT) 2000-9000K and brightness from 0%~100%
- User sets by adjusting two knobs

- A button that turns the lights on/off

Wireless Communication

- Sensor unit contains transmitter module and lighting controller contains receiver
- Transmitter data pin connected to sensor microcontroller
- Transmitter modulates serial data and transmits wirelessly using ASK at 315 MHz
- PIC has UART driver. We use RS232.

Wireless Communication

- UART requires one byte at a time
- Package is 13 bytes long
- RGB values 2-bytes long, must split
- First two bytes are header used to identify beginning of package

Byte	Data
1	255
2	255
3	R[7:0]
4	R[15:8]
5	G[7:0]
6	G[15:8]
7	B[7:0]
8	B[15:8]
9	CCT (2-9)
10	0
11	Brightness (0-100)
12	0
13	On/Off

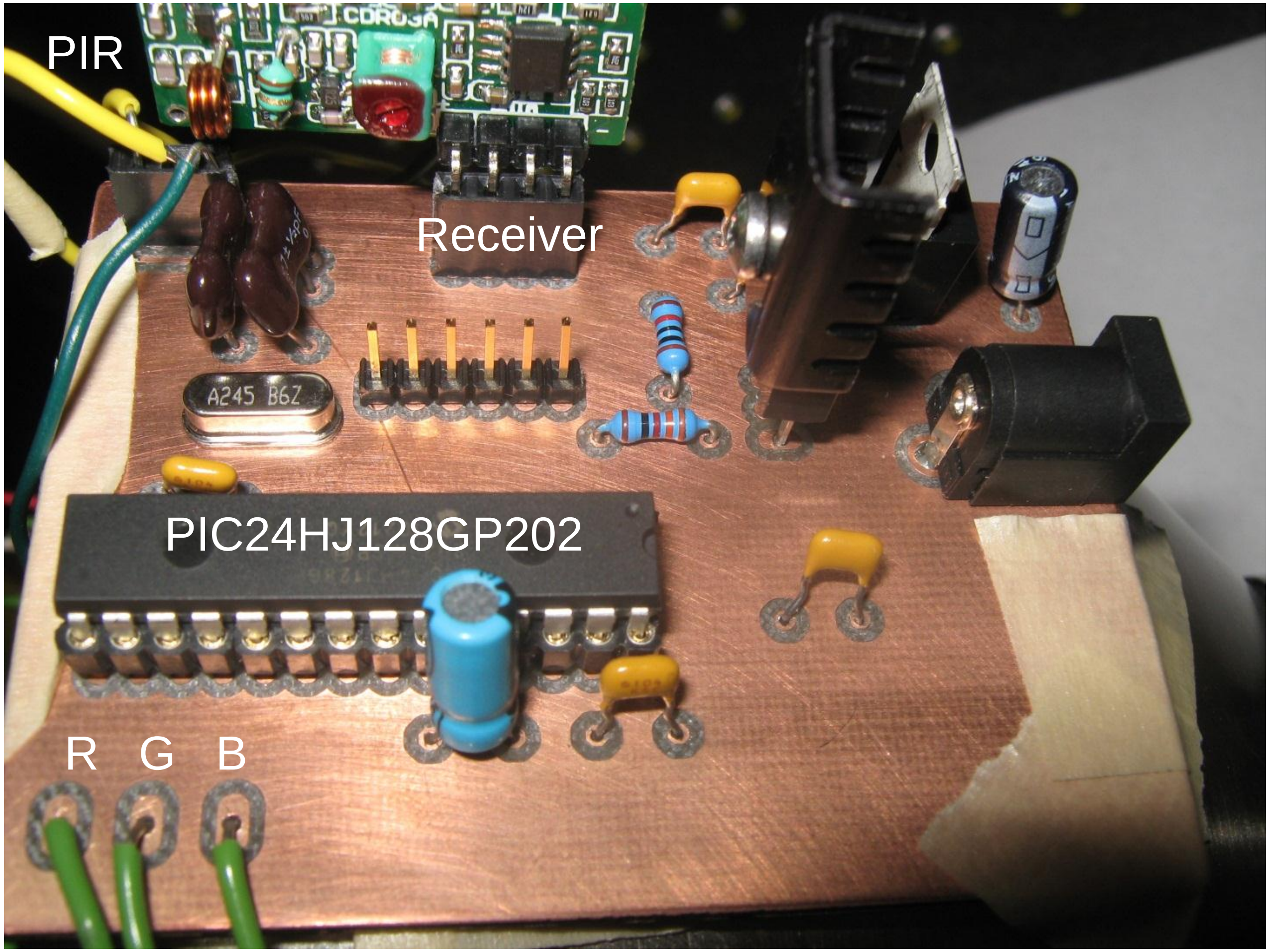
Sensor Unit: Testing

- Tested sensor frequency capture by comparing with oscilloscope.
- Verified data received using PICkit2 UART tool
- We confirmed rate of transmission met specification by counting packets ~300/min = ~5 packets/s

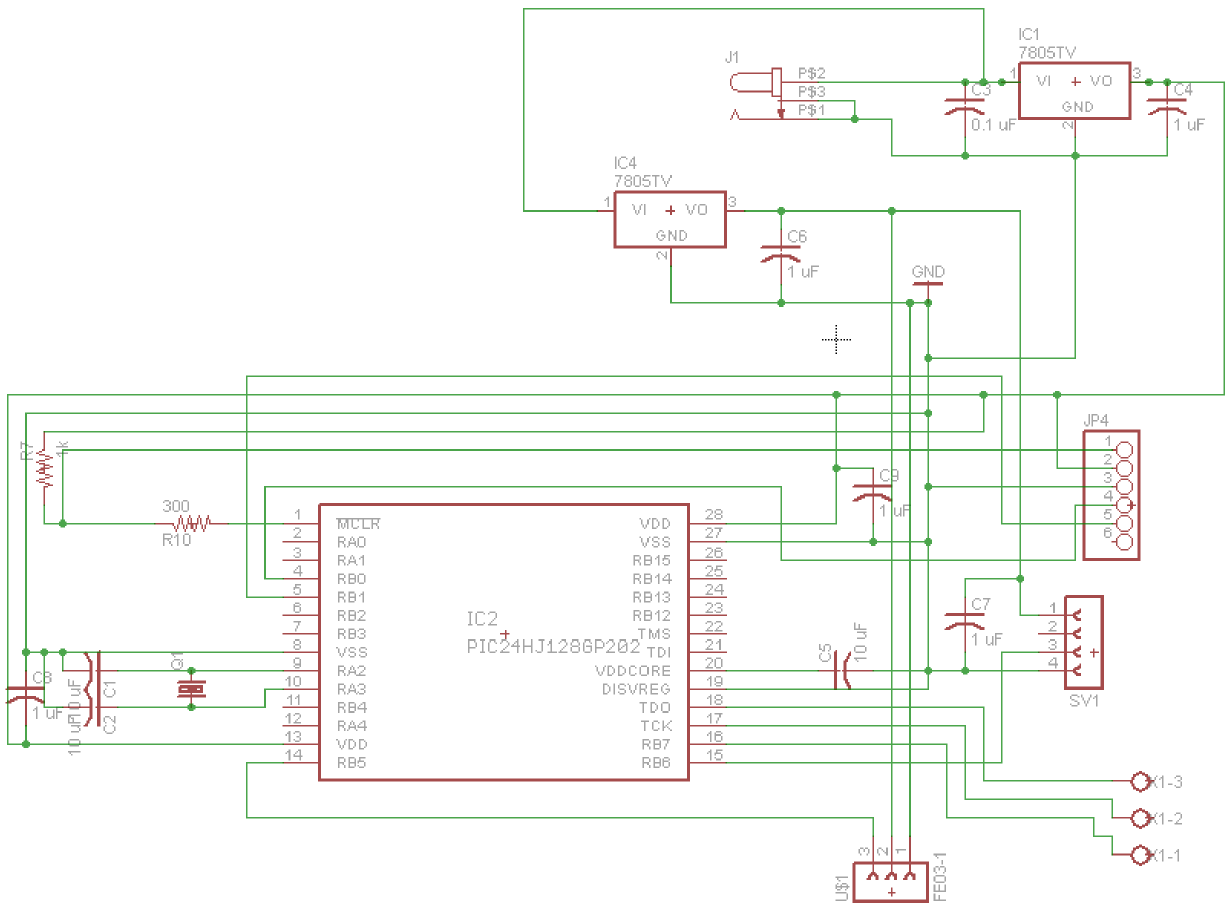
Lighting Controller

- Receives sensor data
  - R,G,B,CCT,brightness,on/off
- Run PI controller on R,G,B variables
- Output PWM signals
- Connect to PIR sensor
- PIC24HJ128GP202

Lighting Controller



Lighting Controller



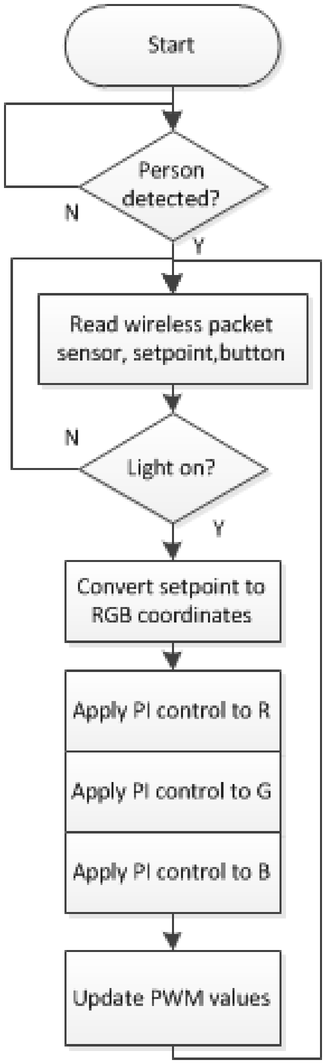
PIR sensor

- The sensor is highly integrated, but we had to test to get right on time and sensitivity
- Simply outputs high (3.3V) if a person is present (moving)
- If not detected, system will be in low power mode
  - Not process any data
  - LEDs off

5V / Out / GND

Lighting Controller: Flow

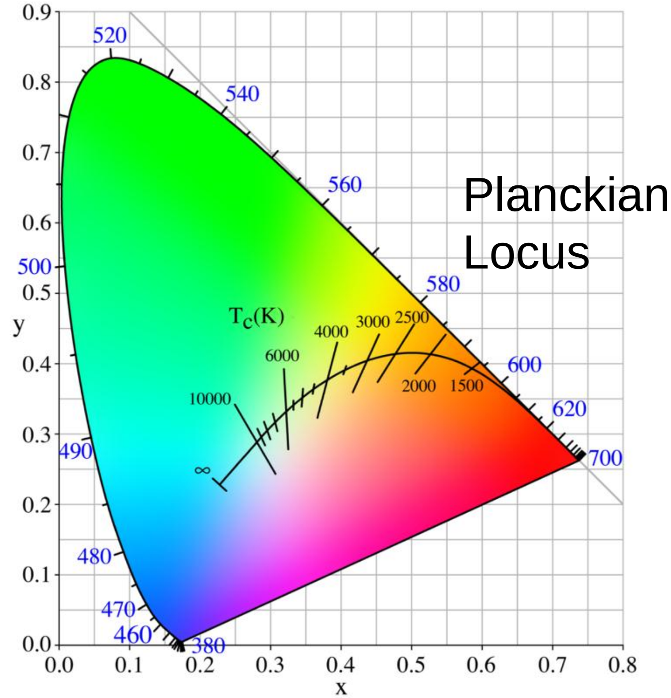
- Control is done in R,G,B coordinates
- Every new received packet
  - Update setpoint
  - PI control iteration done for each color channel
  - Update PWM signal



RGB Target Calculations

- We use CIE 1931 XYZ color space
- Cubic approximation to Planckican Locus to go from CCT to (x,y)
- Scale brightness to get Y
- $X = \frac{x}{y}Y,\ Z = \frac{1-x-y}{y}Y$
- Matrix N maps X,Y,Z to R,G,B

$$\begin{array}{ccc} \text{CCT} \longrightarrow \text{x,y} & \searrow & \\ & & \text{X,Y,Z} \\ \text{Brightness} \rightarrow \text{Y} & \nearrow & \\ & & \downarrow \text{N} \\ & & \text{R,G,B} \end{array}$$



CIE 1931 color space chromaticity diagram

Color Sensor Calibration

- We want a relationship between R,G,B and X,Y,Z
- For red, green, and blue individually
  - Record sensor readings
  - Obtain X,Y,Z values using spectrometer
- Compute calibration matrix M
- $N = M^{-1}$
  - R,G,B of sensor and PWM only differ by constant

$$\underbrace{\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}}_{\text{Tristimulus Values}} = \underbrace{\begin{bmatrix} \mathrm{m}_{11} & \mathrm{m}_{12} & \mathrm{m}_{13} \\ \mathrm{m}_{21} & \mathrm{m}_{22} & \mathrm{m}_{23} \\ \mathrm{m}_{31} & \mathrm{m}_{32} & \mathrm{m}_{33} \end{bmatrix}}_{\text{Calibration Matrix}} \underbrace{\begin{bmatrix} R_s \\ G_s \\ B_s \end{bmatrix}}_{\text{Sensor Output}}$$

$$M = \begin{bmatrix} \mathrm{X}_r & \mathrm{X}_g & \mathrm{X}_b \\ \mathrm{Y}_r & \mathrm{Y}_g & \mathrm{Y}_b \\ \mathrm{Z}_r & \mathrm{Z}_g & \mathrm{Z}_b \end{bmatrix} \begin{bmatrix} \mathrm{R}_r & \mathrm{R}_g & \mathrm{R}_b \\ \mathrm{G}_r & \mathrm{G}_g & \mathrm{G}_b \\ B_r & B_g & \mathrm{B}_b \end{bmatrix}^{-1}$$

PI Controller

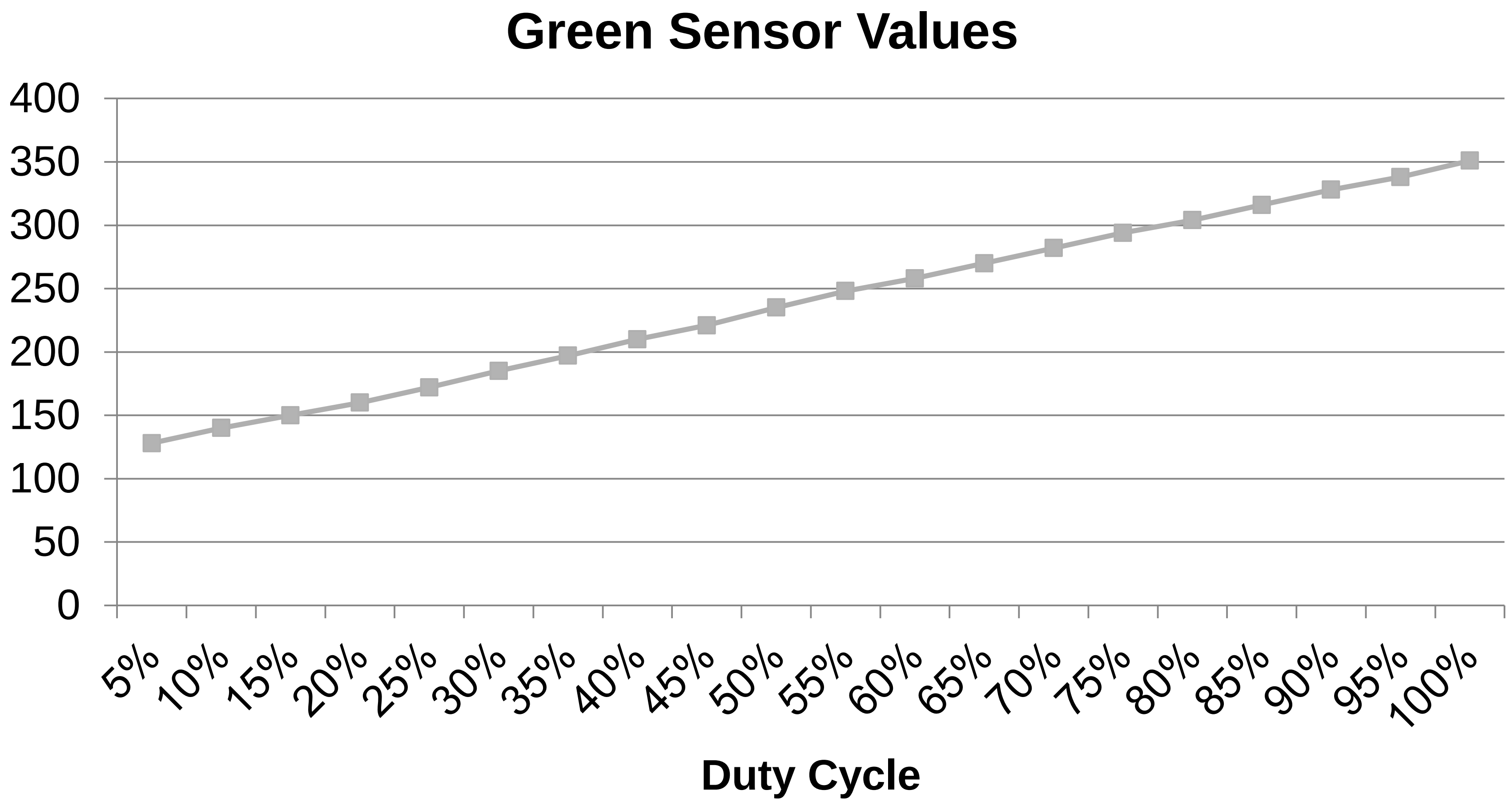
- For each color channel
  - $Error_R = R_{target} - R_{sensor}$
  - Error less than threshold? Stable. Else:
  - $\Delta R_{PWM} = k_{P,R} Error_R + k_{\mathrm{I},R} ErrorSum_R$
  - $R_{PWM} \mathrel{+}= \Delta R_{PWM}$
  - If PWMs saturate, set to maximum or minimum (100%,0%)

Control assumptions

This works because

- Sensor output is linear with incident power
  - Will reach the setpoint more slowly under nonlinear conditions
- LED output power is linear with PWM duty cycle
  - important because it influences PI control

Linearity of PWM



Lighting Controller: Testing

- Once system was integrated, only PI control needed testing
- We allow a settling time of less than 10s in the design review
- Some times are longer than 10s
  - Needs better tuning K_i and K_p
  - Tuning is needed for every change in LEDs relative position to the sensor to get the contribution of LEDs

Recommendations

- Brighter LED’s for larger controllable range
- More accurate calibration matrix would make setpoints more accurate (~10% error currently)
- Tuning PI factors would lead to lower settling time, lower steady state error
- A better sensor with stable output smooth, narrow peaks for spectral response
- Second color sensor with ND filter
  - Handle brighter light when other sensor saturates

Future Work

- The system is extendable for LEDs with more colors to refine the output light
- Users are able to adjust more factors, i.e. arbitrary color in (x,y) space
- More sensors can be used to evaluate the environment better
- Sensor board battery powered
- Time of day settings

Ethical Considerations

- Safety of device was considered and evaluated before implementation and before the demonstration.
- Precautions were taken while building and soldering to not put ourselves.
- We must not overstate our performance claims.

Thank You

- Ryan May
- Alex Suchko
- Kevin Colvary
- Dr. Dragic
- Dr. Carney
- Mark Smart
- Parts Shop Team

Questions