Automatic Light Dimming Control System Based on PID Control Design

L. Souder and D. Boorstein
Section 1

May 8, 2018

1 Abstract

Demand for light dimming control systems is predicted to grow in the coming years with the rise in green buildings and green engineering. Current light dimming systems do not address the issue of high fluctuations in light level which can have damaging effects on occupants' eyes. This can also affect occupants' mental health, stress level, and productivity in the work environment. This report presents a design for a novel light dimming system based on software-implemented PID control theory. A central MSP430FR2433 is programmed to implement PID control for a negative feedback system consisting of an array of LEDs and current-output light sensors. PCBs were designed and printed. However, a final product was not fully built or tested due to several roadblocks and constraints. A fully designed system could definitely be built with additional time and effort.

2 Introduction

Sick building syndrome (SBS) is a condition often related to decreased productivity and increased stress, supposedly caused by deterioration of one's interior environment [1]. This can be caused by a number of factors, including indoor lighting. Indoor lighting is an often overlooked design consideration in modern residential and work spaces, and its impact is significant on productivity and health. The Occupational Safety and Health Administration (OSHA) measures general lighting in the workplace using foot-candles as units, requiring physical plants maintain 10 foot-candles of brightness and offices at least 30 foot-candles [2]. In labor-intense environments like manufacturing plants, factories, or workshops, proper lighting ensures accident prevention, hazard avoidance, and physical safety. In the office, where these dangers are often not present, lighting contributes directly to mental health which influences productivity [1],[3],[4].

In 2009, Samsung World Headquarters in Seoul, Korea launched a new Energy Saving Campaign and earned the highest ranking Green Building Certification System certification by the Korean Green Building Council (KGBC). During daylight hours, half of the interior lights in the building were to be turned off to reduce power consumption [1]. During sunny days, this was no issue; during cloudy days, however, the offices were dark. This significantly affected employee mood and contributed to their annoyance ratio of their workplace [1]. Daylight can certainly improve mood and has even shown to prevent certain diseases and influence window design in residential and office buildings [3]. However, visual distractions, variation in illuminance, and high stimulation caused by improper lighting may contribute to higher amounts of annoyance and stress [4]. Higher stress can lead to decreased productivity.

As green engineering and green buildings become more popular in the coming years, the demand for light dimming control systems should also increase. Current light dimming control systems offer illumination at a fraction of the energy cost. A recent Zigbee-based indoor light control system was developed using a network of light-sensitive diodes and a central MSP430F2619 microprocessor and CC2430 RF chip. Mapping and modulation of the network was determined by the Zigbee standard, which is based on the IEEE 802.15.4 standard [5]. Novel systems like these should take into account not only the level of brightness, but also the uniformity of brightness, how often brightness in the room varies during the day, and how quickly the system responds to change. The focus of these systems should be to achieve a comfortable medium between energy savings and occupant's comfort.

Therefore, a control system with proper transient behavior which addresses this issue must be developed in order to ensure the well-being of the occupants is secured while still maintaining proper light level and also saving energy. This report presents a design for a novel light dimming control system based on classical PID control. A system was to be constructed inside of a 1 cubic foot box and would contain an array of LED, motorized blinds, and a feedback sensor array. The primary function of the system would be to maintain a constant brightness in the room, using first and foremost the blinds to allow daylight into the room and compensating with the LED until a target brightness was achieved. A second-order design goal for the system would be to demonstrate proper compensator design to achieve a transient behavior which would not cause eye strain for the room's occupants. Compensation of the system was to be implemented using a central microcontroller with its own independent power supply for ease of transportation and setup.

3 Standards And Constraints

This section discusses the standards, regulations, and constraints of the project.

3.1 Standards and Regulations

Code for the software control system, which is written in C, conforms to the current standard for the C programming language, the ISO/IEC 9899:2011 or C11 [6].

This system is intended for demonstration only. However, the principals employed here could be up-scaled to a consumer product in a house or office space. Therefore, this design adopts several standards to ensure occupant safety. This project adheres to IEEE 1789-2015 Recommended Practices for Modulating Current in High Brightness LEDs for Mitiagting Health Risks to Viewers [7].

3.2 Constraints

It was originally intended for this system to be self-contained inside of a 1 cubic foot box. Completely independent of any external power, it was to posses its own transportable power supply. This was much less a constraint and more a criterion, however it did provide direction for the design.

Another constraint of the project was the use of the TI MSP430 line of microprocessors as opposed to an Arduino. The Arduino microprocessors are arguably much easier to program when compared to the C-code structure required for MSP430 microprocessors. What initially appeared as a time constraint eventually became a boon. This inevitably lead to the creation of a much more robust library of functions for the purpose of PID tuning written in C.

The foremost constraint of the project was the number of printed circuit boards that needed to be assembled for the system to work. There were, in total, 14 separate PCB with surface mount components to be hand-soldered. Without these components, there was no system, because these boards contained the lights, light sensors, and central processing unit with power supply. The LEDs only came in a surface mounted package, which made PCB design necessary. Because these components were not assembled, the system itself was not realized nor tested. This was a major hindrance of the project, as it prevented actual PID implementation and demonstration of the design.

The system was also planned to have an opening on one of the sides with controllable vertical blinds, simulating daylight for the room. The complexity of this design became a hassle, especially with the limited time of the project, and this idea was ultimately put aside. In order to maintain a controlled environment inside the room, the cutout which would house the motorized blinds was also done away with. This constrained the design in two ways: (1) there was now no way to see inside the box to tell if the control system was working, and (2) there was no way to add disturbance to the system to identify proper PID compensation. Some ideas which were generated but unfortunately never implemented included:

- adding cameras to the inside of the box to be able to monitor the interior,

- cutting small holes in multiple parts of the box with slider shutters to simulate disturbance into the system, and
- manually changing the brightness on certain lights and forcing the other lights to compensate for this disturbance (essentially using a ”broken” light)

4 Control Design

This section discusses the characterization of the control system. It includes a discussion on the development of the system thus far, the design goals for the system, and control loop implementation. Listed here are some of the original design goals for the system despite any constraints that arose during development of the project, such as time and material constraints.

4.1 System Characterization

The system is characterized by an open loop interior light level determined by a feedback light sensor array, producing a closed-loop control system. A central microprocessor is to receive the input, output, and error signal of the closed-loop system. The design is to be compensated using a software-based PID compensator through the microprocessor.

4.2 Design Goals

The goal of the interior light dimming system is to achieve, first and foremost, a target brightness and uniformity in the room. This is to be accomplished through separate communications to each LED based on feedback through light sensors located on the floor of the box. The transient behavior of the system is then to be tuned using PID control implemented through software on the microcontroller's internal processor. The overall transient behavior of the system is to be slow so that the rate of change of the system does not irritate or strain the eyes.

4.3 Control Loop Implementation

An MSP430FR2433 serves as the control loop in this system. Control functions are to be implemented through software. A library of functions was written in C to calculate all necessary constants and variables for a PID controller, including errors and gains.

For example, the error signal was calculated by subtracting the the output reading from the target variable of the controller on line 17 of `Controller.c`. This value is then assigned to the an error variable of the controller.

```
int calcError(struct &Controller controller, int
    outputReading) {
        int error = controller.target - outputReading;
```

```
        controller.e = error;
        return error;
}
```

The integral error is calculated by multiplying the error by a user-defined time step.

```
int calcIntError(struct &Controller controller, int
    outputReading, int timeStep, int error) {
        int int_error = error * timeStep;
        controller.int_e += int_error;
        return int_error;
}
```

Derivative error is found by taking the rate of change of the error signal over the time step.

```
int calcdError(struct &Controller controller, int
    outputReading, int timeStep, int error) {
        int deltaError = error - controller.y;
        int rate = deltaError/timeStep;
        controller.de = rate;
        return rate;
}
```

These functions allow for calculation of total PID error in a single function.

```
void calcPID(struct &Controller controller int
    outputReading, int timeStep) {
        int error = calcError(controller, outputReading);
        calcIntError(controller, outputReading, timeStep,
            error);
        calcdError(controller, outputReading, timeStep,
            error);
}
```

Finally, an update function was created to automatically recalculate PID error each time step.

```
void updateDriver(struct &Controller controller, int
    outputReading, int timeStep) {
        calcPID(controller, outputReading, timeStep);
        controller.driver =     (controller.Kp *
            controller.e) +
        (controller.Ki * controller.int_e) -
    (controller.Kd * controller.de);
}
```

5 Design Discussion

The microprocessor chosen for this design was the MSP430FR2433. This is one of the latest microprocessors in the MSP430 line from TI and features optimized ultra-low power sleep modes and ferroelectric RAM for increased energy efficiency. It also has support for an external 32kHz crystal oscillator and more than enough I/O pins [8]. This makes the FR2433 practical for this design. Nine ASMB-MTB1-0A3A2 LEDs and four PNJ4K01F current-output light sensors were used for this design. The PCA9685 LED driver was chosen to drive the LEDs. A 6-switch Dip-Switch package was included in the design for selection of the LED driver address. This served as a safety mechanism to safeguard against accidentally permanently selecting an address during the design process and then finding out it was a reserved address later on.

An overall UML for the light controller system is given in Fig. 1. The FR2433 serves as the central control system where software PID control is to be implemented.

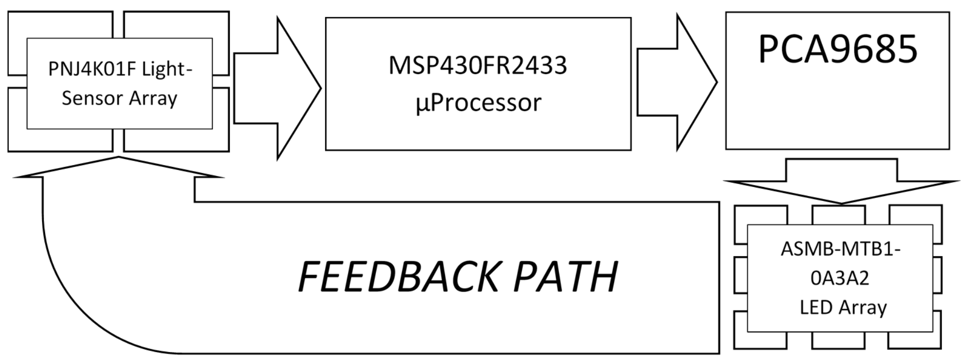

Figure 1: UML for the Light Controller

A 9V battery supply was chosen as the primary method of supplying power to all circuits. This was selected as 9V batteries are easily obtainable and replaceable in case of failure. The light sensors and LEDs required a +5V power supply, which necessitated a voltage regulator. The LM317 was chosen due to its frequent use in other circuits and the authors' experience with the device. Similarly with the FR2433, which required a supply of +3.3V supply, another linear regulator – namely the NCP551 – was selected. All components purchased for this design were surface-mount, which prompted PCB design.

Open source PCB design software, KiCAD, was used to design the PCB for the controller circuit and individual light boards. Overall circuit diagrams for the light controller and individual light boards are given in Section 7 in Fig. 5 and Fig. 6, respectively. The central PCB is intended to serve as a Launchpad^{TM} daughter card for convenient programming of the off-board microprocessor chip. The design for the controller board can be seen in Fig. 2. For the design of a single light board, see Fig. 3. You will notice that the light boards include all of the necessary pads and connections for both an LED and an ambient light sensor. This was done solely to cut back on costs, since each individual design adds to the price. In practice, we would use 4 boards with *just* a sensor for our sensor array, and 9 boards with *just* an LED for our LED array.

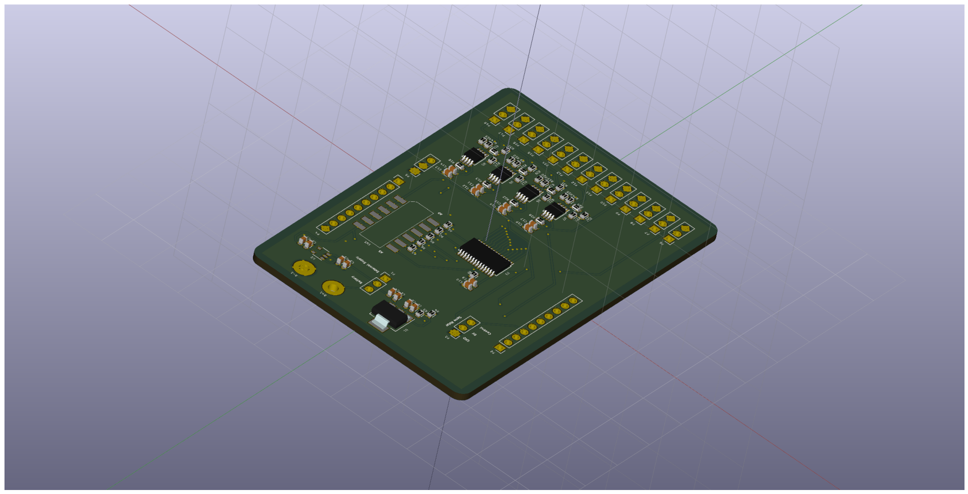

Figure 2: Light controller daughter card model

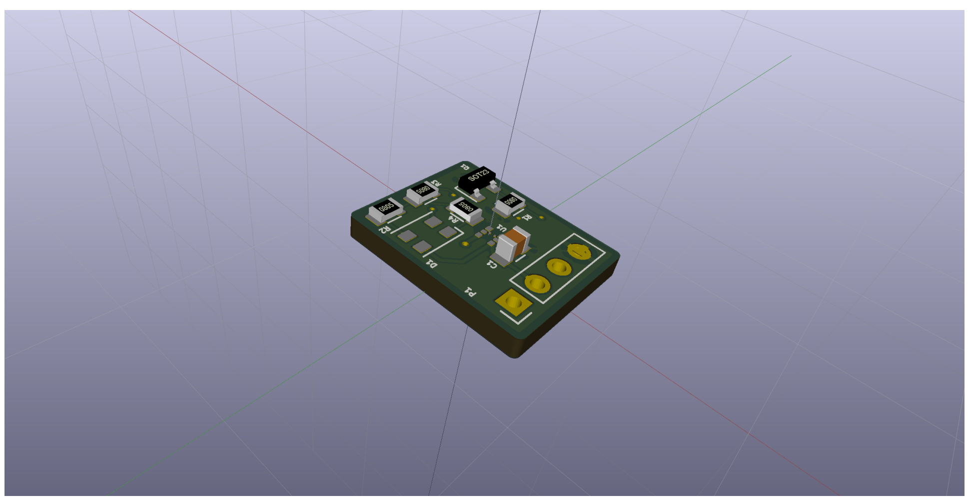

Figure 3: LED light board model

Design of the system began with software implementation of PID control. This was completed using TI's dedicated IDE for MSP430 processors, Code Composer Studio. Code for the control system is discussed in Section 4.3. Further coding was required for microprocessor register initialization, such as the FR2433's internal ADC.

A full block diagram for the system is given in Fig. 4. Gains K_P, K_i, and K_d are parameter gains determined through PID tuning. The system must first convert a target light level to a voltage through use of a transimpedance amplifier. After PID compensation, the system then converts this voltage signal into light through the LED circuit. This light is fed back through the sensor array, producing a negative feedback control.

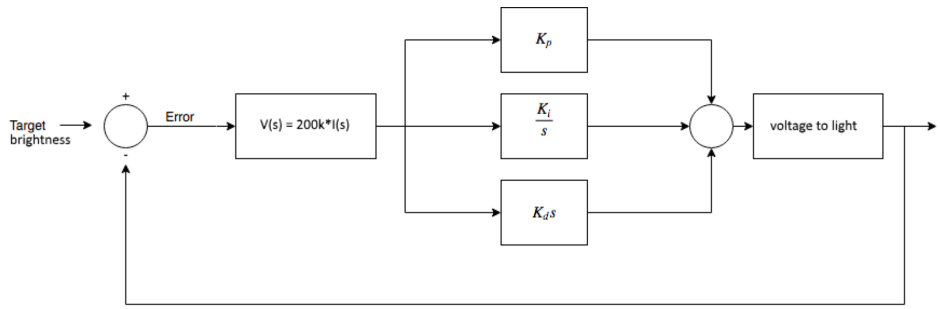

Figure 4: Block diagram of the system

6 Results and Conclusions

Code for the software control loop and a library of functions was written. Several PCB were printed and parts delivered. At the current time of writing, the project is unfinished. The system was never tested, unfortunately, because the PCB have not been assembled due to scheduling conflicts within the Tech Office. However, all the parts are currently in the possession of the authors and can be assembled. The authors anticipate that this system will operate correctly without much more work and that all that is left to do is manually tune each parameter gain of the PID controller.

The design here nonetheless presents a framework for future designs. This report stresses the importance of choosing proper transient behavior for the system to prevent eye-strain and promote psychological wellness for indoor occupants. Design through PID compensation aids in this effort, as one can choose different gain values for each gain stage (proportional, integral, and derivative) to achieve a desired transient and steady-state behavior.

6.1 Work Distribution Summary

Damon Boorstein

- Parts shopping
- Register initialization code

- Main author of this report

Lonnie Souder II

- Wrote libraries for PID controller
- Designed light controller and main board schematics
- Designed light controller and main board PCBs

6.2 GitHub Link

A link to this project's Git repository can be found here:
`https://github.com/boorsteid4/snc-closedloop`

7 Appendix A: Code

```c
/*
 * Controller.c
 *
 * Lonnie L. Souder II
 * 04/13/2018
 *
 * This file defines a data structure which contains all
    the necessary
 * data for a PID controller including gain constants and
    rolling
 * sums for error, integral of error, and derivative of
    error.
 * In addition, this file will include methods used to
    derive and manipulate
 * all of data collected by the PID controller
 */

#include "Controller.h"

int calcError(struct &Controller controller, int
    outputReading) {
        int error = controller.target - outputReading;
        controller.e = error;
        return error;
}

int calcIntError(struct &Controller controller, int
    outputReading, int timeStep) {
        int error = calcError(controller, outputReading);
        int int_error = error * timeStep;
        controller.int_e += int_error;
        return int_error;
}

int calcIntError(struct &Controller controller, int
    outputReading, int timeStep, int error) {
        int int_error = error * timeStep;
        controller.int_e += int_error;
        return int_error;
}

int calcdError(struct &Controller controller, int
    outputReading, int timeStep) {
```

```
        int error = calcError(controller,
           outputReading);       //current error
        int deltaError = error - controller.y;
        int rate = deltaError/timeStep;
        controller.de = rate;
        return rate;
}

int calcdError(struct &Controller controller, int
    outputReading, int timeStep, int error) {
        int deltaError = error - controller.y;
        int rate = deltaError/timeStep;
        controller.de = rate;
        return rate;
}

void calcPID(struct &Controller controller int
    outputReading, int timeStep) {
        int error = calcError(controller, outputReading);
        calcIntError(controller, outputReading, timeStep,
           error);
        calcdError(controller, outputReading, timeStep,
           error);
}

void updateDriver(struct &Controller controller, int
    outputReading, int timeStep) {
        calcPID(controller, outputReading, timeStep);
        controller.driver =     (controller.Kp *
           controller.e) +
                                                (controller.Ki
                                                   *
                                                   controller.int_e)
                                                   -
                                                (controller.Kd
                                                   *
                                                   controller.de);
}


/***************************************
* Helper Functions for PWM Controller
***************************************/

int Clamp(&int subject, int min, int max) {
        if (subject > max)
```

```
                subject = max;
        else if (subject < min)
                subject = min;
        return subject;
}

void updateDriver(struct &Controller controller, int
    outputReading, int timeStep) {
        calcPID(controller, outputReading, timeStep);
        controller.driver =     (controller.Kp *
            controller.e) +
                                                (controller.Ki
                                                    *
                                                    controller.int_e)
                                                    -
                                                (controller.Kd
                                                    *
                                                    controller.de);
        Clamp(controller.driver, 0, controller.driverMax);
}
```

8 Appendix B: Schematics

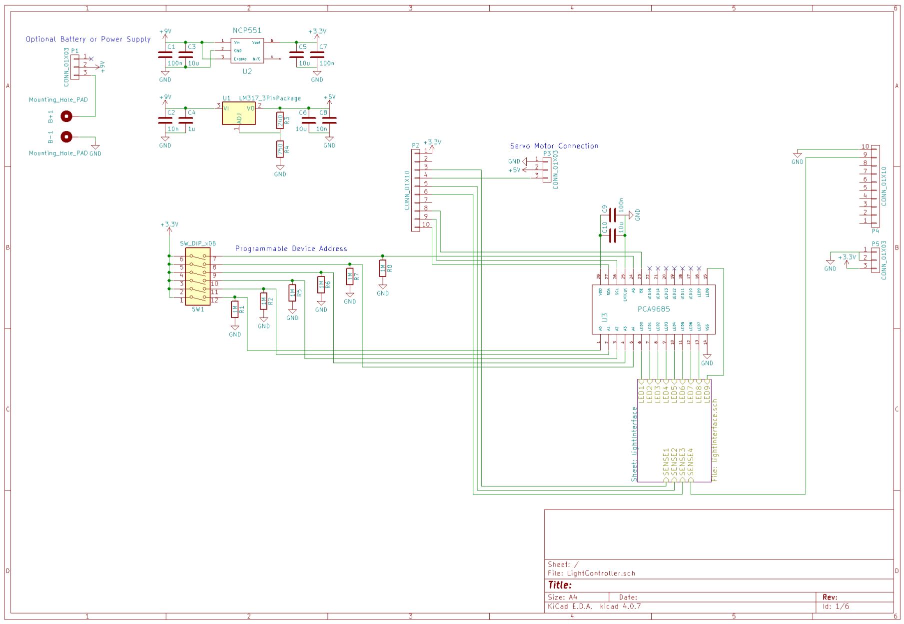

Figure 5: Schematic for the Light Controller

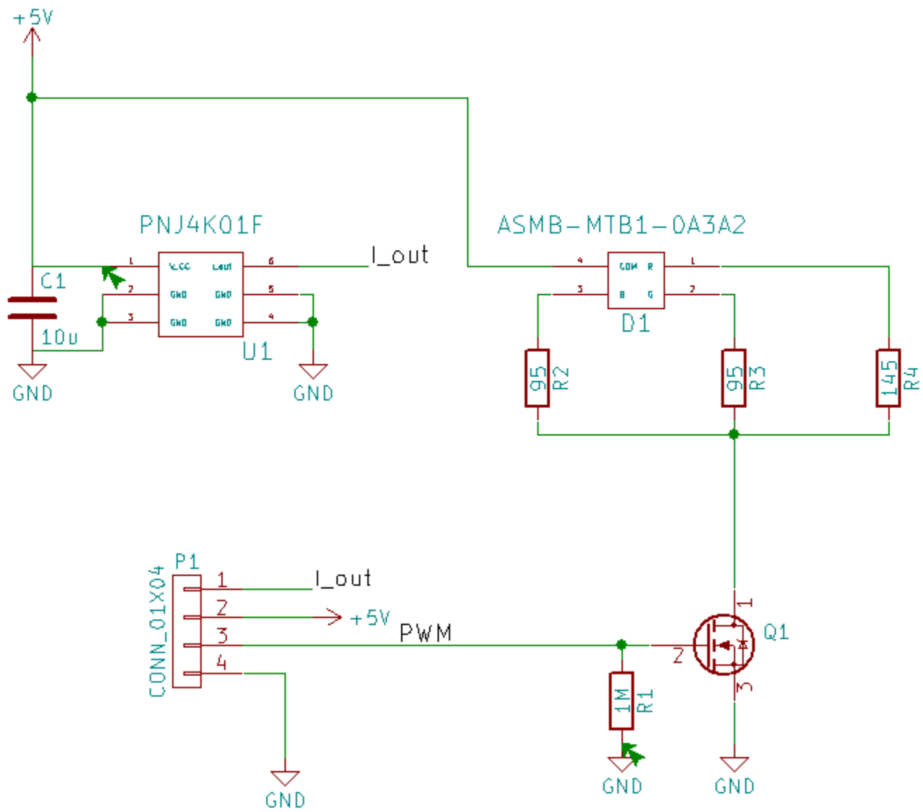

Figure 6: Schematic for Light Board

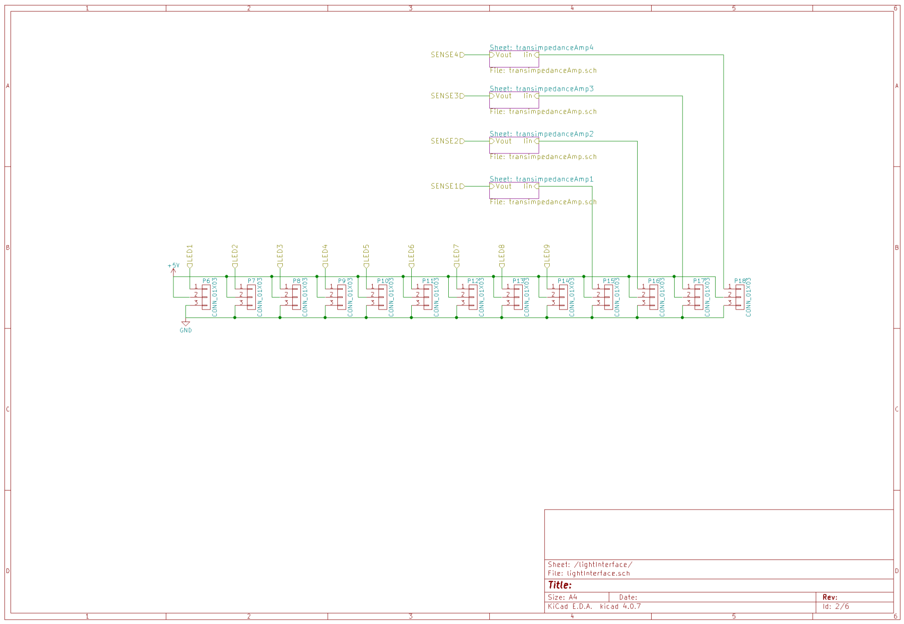

Figure 7: LED Sensor array schematic

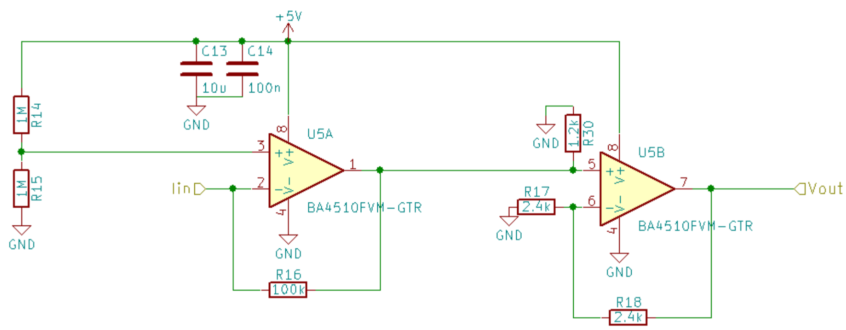

Figure 8: Example transimpedance amplifier design inside sensor array

References

[1] T. Hwang and T. K. Jeong, "Effects of indoor lighting on occupants visual comfort and eye health in a green building," *Indoor and Built Environment*, vol. 20, no. 1, pp. 75–90, 2011. doi: 10.1177/1420326X10392017; 05.

[2] J. Williams, "Requirements for workplace lighting," Aug 3, 2010.

[3] P. Xue, C. M. Mak, and H. D. Cheung, "The effects of daylighting and human behavior on luminous comfort in residential buildings: A questionnaire survey," 2014. ID: 271434.

[4] J.-H. Lee, Y. Yoon, Y.-K. Baik, and S. Kim, "Analyses on human responses to illuminance variations for resident-friendly lighting environment in a small office," *Indoor and Built Environment*, vol. 22, no. 3, pp. 535–550, 2013. doi: 10.1177/1420326X12446208; 05.

[5] C. Z. Lu, X. Yang, and W. Pan, "A zigbee-based indoor intelligent lighting system research and design," *Applied Mechanics and Materials*, vol. 71-78, p. 2276, 2011. Source type: scholarlyjournals; Object type: Article; Object type: Feature; Copyright: Copyright Trans Tech Publications Ltd. Jul 2011; DOCID: 3105622361; PCID: 81499262; PMID: 205230; ProvJournalCode: LMCM; DOI: 10.4028/www.scientific.net/AMM.71-78.2276; DocISBN: 978-3-03785-203-3; PublisherXID: ICATTPL_LMCM_V7178_20110701_B794667EB66B7DF972358.

[6] Wikipedia contributors, "C11 (c standard revision) — Wikipedia, the free encyclopedia," 2018. [Online; accessed 7-May-2018].

[7] IEEE, "Recommended practices for modulating current in high-brightness leds for mitigating health risks to viewers," *IEEE Std 1789-2015*, pp. 1–80, June 2015.

[8] TI, *MSP430FR2433 Mixed-Signal Microcontroller*, 6 2017.