# MULTI-SENSOR MOTION DETECTOR FOR RELIABLE LIGHTING CONTROL

By

Siddarth Boinpally

Joseph Paxhia

Lukas Ping

Final Report for ECE 445, Senior Design, Fall 2025

TA: Shiyuan Duan

10 December 2025

Project No. 21

## Abstract

This project, ILLUMINATE, is an automated room-occupancy lighting system that uses mmWave radar, PIR sensing, and a servo-driven pan/tilt scan to detect people and control an LED module. We built a custom two-board PCB setup that handles AC-DC power conversion, sensor connections, and closed-loop LED current control with an LED driver. An STM32 microcontroller reads the sensors, runs the scan routine, and uses a state machine to decide how the light should behave. In the lab, we confirmed that the 5 V and 3.3 V rails were stable, the LED current followed our commands, UART communication with the mmWave module was solid, and the servos moved to the expected angles. With everything integrated, the system detected occupancy within the scan area and turned the LED on with consistent timing. The final prototype provides semi-reliable presence detection for small indoor spaces and controls the light without storing personal information.

# Contents

# 1. Introduction

ILLUMINATE is a system designed to automate room-occupancy lighting using a combination of sensing and mechanical scanning. The following section outlines the motivation behind the project, the limitations of existing lighting solutions, and how ILLUMINATE addresses those challenges. From there, the report describes each subsystem, including sensing, power, actuation, and control, and explains how they work together to enable reliable occupancy detection. Cost, timing, ethics, and potential future improvements are also discussed.

## 1.1 Problem

In offices, classrooms, and lecture halls, motion sensors run the lights. They are convenient, but there is a clear flaw, the lights turn off when people are still, typing, reading, or watching a presentation. That is frustrating, it breaks focus, and it hurts productivity. Most systems rely on PIR sensors, which look for changes in infrared from warm bodies. They work for big movements, but they miss micromotions, they can false trigger, and they depend on fixed timeout settings. The result is simple, they do not consistently recognize that people are still in the room.

## 1.2 Solution

Our approach uses a two stage check to boost reliability while keeping what already works. Stage one uses a PIR sensor for fast entry detection and larger body movements, it captures when someone first walks into the room. Stage two adds a millimeter wave radar sensor that sees micromotions like breathing or small hand shifts, so presence is still detected during quiet periods. Together, PIR wakes the system on entry, and mmWave maintains accurate presence until everyone has left. This split lets each sensor do what it does best and keeps response time quick without losing precision.

We also add a multi level lighting setup to save energy and improve comfort. On detection, the lights ramp up smoothly to the target level, no harsh jumps. While people remain in the room, brightness holds at the right level, guided by ongoing checks from the mmWave sensor. If no presence is seen for a set interval, the system dims to an in between level to signal a possibly empty room. After one more quick verification, the lights turn fully off, and the timeouts are adjustable so teams can tune comfort and savings for each space.

## 1.3 Visual Aid

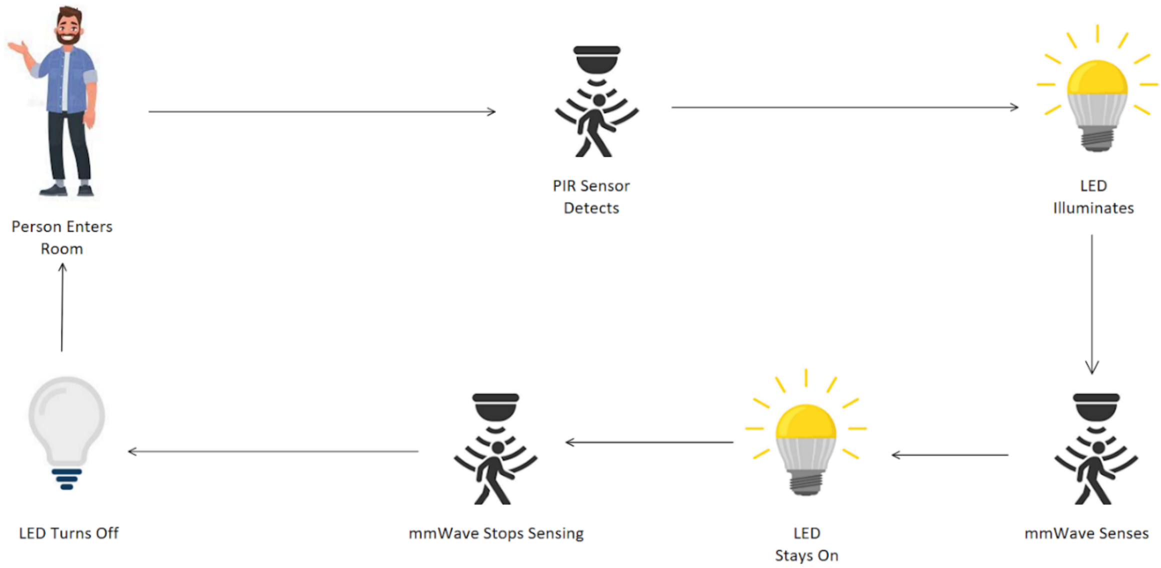


**Figure 1 Overview of the Multi Sensor Lighting System**

We have attached a high-level diagram for our sensing system in Figure 1. The PIR Sensor is used to turn on the light, while the mmWave sensor is used to either keep the light on or turn off the light.

## 1.4 High-Level Requirements

To be considered successful, our project must meet the following objectives:

1. The light must begin to illuminate within 1 second of someone entering a room, stay on for 30 minutes with someone being in the room, and begin to turn off after 1 minute of not detecting anyone
2. Capable of detecting people in all parts of a 25'x40' Room (or smaller).
3. The motor that controls the direction of the sensor should be able to sweep 180 degrees horizontall9oy and 90 degrees vertically at the same time.

## 1.5 Block Diagram

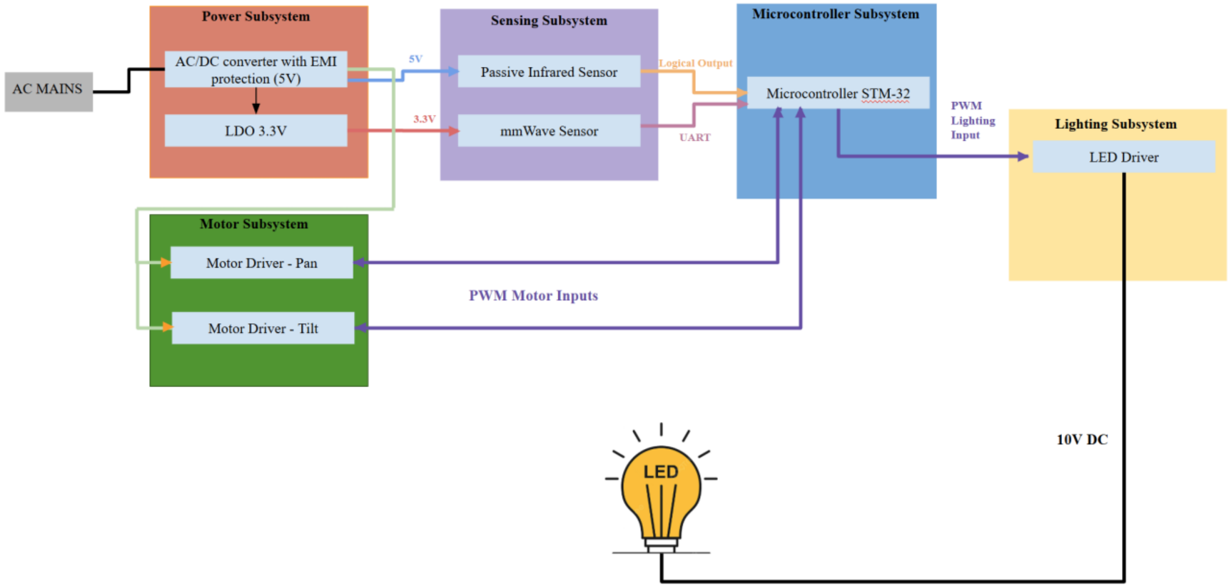


**Figure 2 Block Diagram for Multi-Sensor Detection Lighting System**

Our block diagram consists of 5 subsystems: Power, Sensing, Microcontroller, Lighting, and Motor. The power subsystem is responsible for powering all components within the scope of our project, and consists of taking power from 120 VAC AC mains. The sensing subsystem senses whether people are in the room, and outputs this data to the STM32 MCU. The microcontroller outputs PWMs for the lighting, and motor. The PWM for the lighting tells how much light, and if the lights should be on, and the PWMs for the motor tell the motors where to go. The Lighting subsystem drives the LEDs to ensure the correct power is sent to the LEDs so that they are ON, OFF, dimming, etc. The motor subsystem moves the mmWave sensor to ensure that the entire room is scanned.

# 2 Design

## 2.1 Power Subsystem

Our entire system will be powered by AC mains. We will need 5V for the HS-318 motors, 5V for the ADAFruit 189 PIR Sensor, and 3.3V for the C4001 DFRobot mmWave sensor. In addition, the STM32F103C8 requires a 3.3V input. First, one will have to plug in AC Mains. These AC mains will feed to an Inlet Board, which will contain a fuse. Our AC/DC converter is rated for around 0.75A, so we will use a 2A fuse. We will use a 2A fuse so that the fuse does not trip during sudden spikes in current. This board also contains a varistor which will provide overvoltage protection. Our AC/DC Converter is rated for an input voltage of around 264V. So, the varistor we have chosen is 250V, which is sufficient to protect the AC/DC Converter. We will feed the output of this Inlet board into an AC/DC 5V converter with EMI Protection. EMI Protection is necessary since we deal with AC mains.

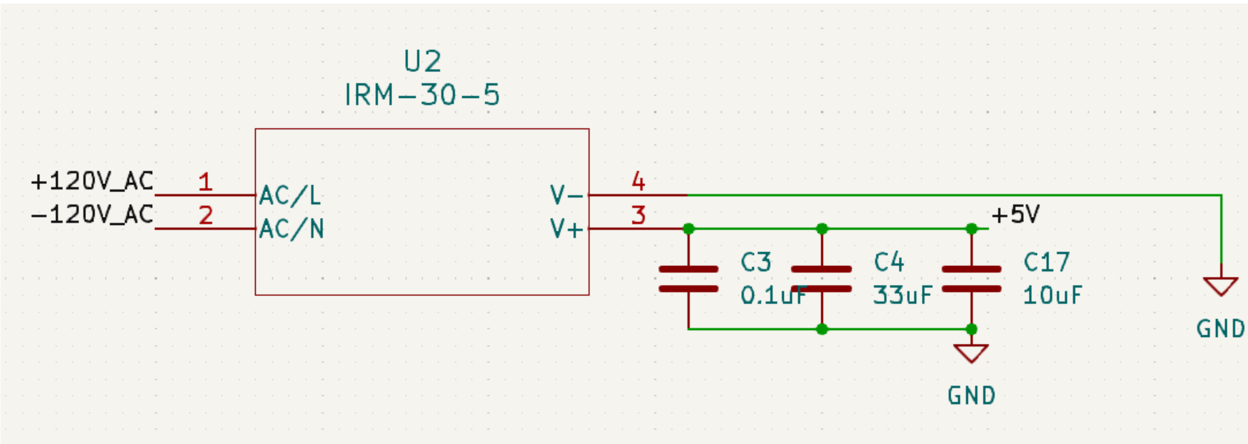


**Figure 3  AC/DC Converter Schematic**

We will be using the IRM-30-5, which is rated at 6A output (our current will not exceed 3A at worst case scenario) [8]. It is also class II, meaning that it is double-insulated and does not require a ground connection. The circuit configuration for this chip is simply the chip and some decoupling capacitors. Capacitor C3 serves as a ceramic decoupling capacitor, used as a high-frequency bypass, and placed very close to the output V+ pin of the IRM-30-5. Capacitors C4 and C17 serve as bulk capacitors, which prevents voltage sagging when there are big current surges. Their values come from the IRM-30-5 Datasheet [10], which recommends a 0.1uF ceramic + 47 uF bulk capacitor for decoupling. The electronic services shop does not have 47 uF capacitors, so we have chosen to use a 33 uF and 10 uF capacitors in parallel to get close to the targeted value.

This 5V will be fed into a 5V to 3.3V LDO. This LDO is rated for 1A [9], which does not exceed our worst case current that the mmWave and LED Driver require. Rather than using a buck converter, an LDO is sufficient for a drop of 1.7V without avoiding too much power loss, as Texas Instruments states that 2V drops are acceptable [6].

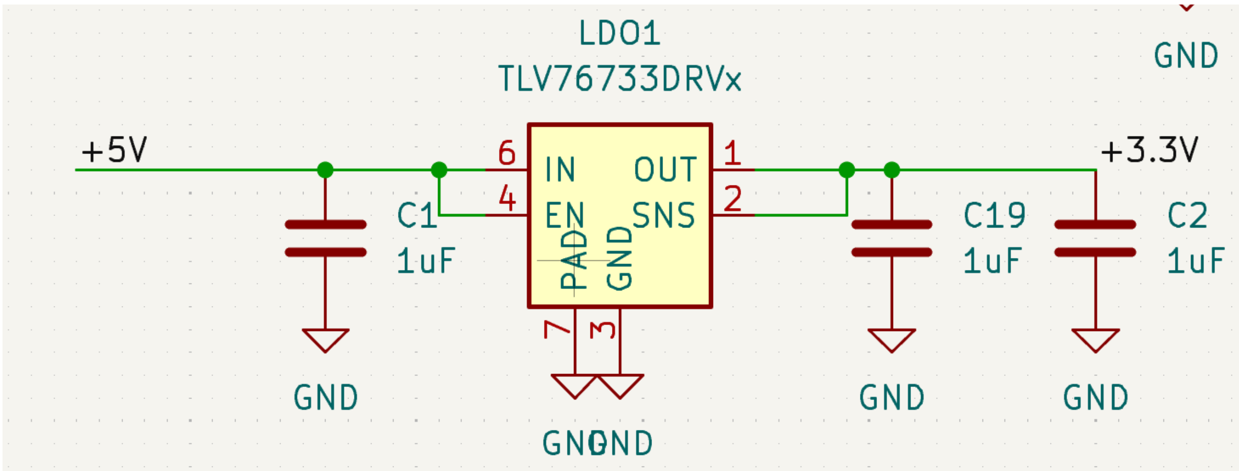


Figure 4 5V-3.3V LDO Schematic

Here is a schematic of our LDO circuit. It contains 2 1uF capacitors on the output of the LDO (3.3V side) and 1uF capacitor on the input side (5V side). The input capacitor is there to stabilize the input voltage to the LDO. The output capacitors are also there to stabilize the output voltage. This circuit comes from a typical application provided from the datasheet [9].

## 2.2 Motor Subsystem

The motor subsystem is powered by 5V. This will power two servo motors that form a double axis motor rotation for the mmWave sensor. The pan servo motor rotates around the vertical axis, allowing horizontal sweeps across the room. The tilt servo will rotate around the horizontal axis, allowing for vertical sweeps. By using horizontal and tilt motors together, this gives our mmWave sensor full detection in the room.

To ensure reliable operation, the motors are powered through a dedicated supply rail capable of handling their peak current draw. Mechanically, the servos are integrated into a compact mount that supports the mmWave sensor while minimizing vibration.

We are going to use two HS-318 servo motors, which support 0.5 degrees of movement increments. These can by controlled by sending PWMs sent by the microcontroller. One can control the pan and tilt by sending PWMs to control the position of the motors. This will then be changed over time to complete a horizontal and vertical sweep of the room. The pan is going to have to go slower than the tilt motor to make sure that the tilt motor is able to tilt fully enough times to scan the room.

## 2.3 Sensing Subsystem

Our sensing subsystem uses both a PIR (Passive Infrared) sensor and a mmWave sensor (C4001), in addition to the necessary surrounding components to make sure the sensing operates correctly. The PIR sensor allows a quick response to the initial movement of a person entering a room, while the mmWave sensor detects continuous motion by seeing micromotions like breathing and typing while people are present in a room.

This sensing subsystem will require a 5V supply for the PIR sensor and a 3.3V supply for the mmWave. This will come from the power subsystem. The PIR sensor will output a digital logic signal that is sent to the STM32 microcontroller's GPIO input. The mmWave sensor communicates with the microcontroller through UART transmission.

For the initial configuration of the mmWave sensor, we will be using an Arduino Uno to demo the use of this sensor before integrating it into our final design to ensure desired specifications.

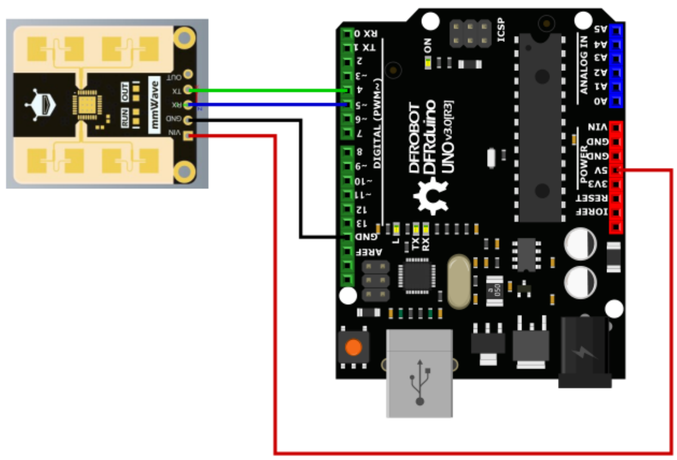


Figure 5  Arduino UNO Configuration for C4001 mmWave sensor

At first, we did not want to use Arduino UNO in our final product. We thought that the mmWave could be programmed using Arduino UNO, but we realized that this was not possible. The mmWave sensor does not get flashed, and it is very challenging to use the STM32 to program the mmWave. This is because DFRobot contains a library inside of Arduino UNO that allows us to set certain parameters, such as the trigger sensitivity. This library is not in the STM32 programmer, so we were unable to program the mmWave sensor without using the Arduino UNO with our project.

We will be placing the mmWave sensor in the middle of the room. We are assuming a 25x40(ft) room, but we will be placing it at the 20ft mark (middle), at the front of the room. Thus, the mmWave must be able to detect 20ft (left to right), and 25 front to back. Thus, we use the Pythagorean theorem and find that the max distance that the mmWave will have to detect from.

$$\max distance^2 = 25^2 + 20^2, \max distance = 32\ ft \tag{1}$$

Thus, the mmWave must be able to detect a human at a distance of 32 ft or less.

## 2.4 Microcontroller Subsystem

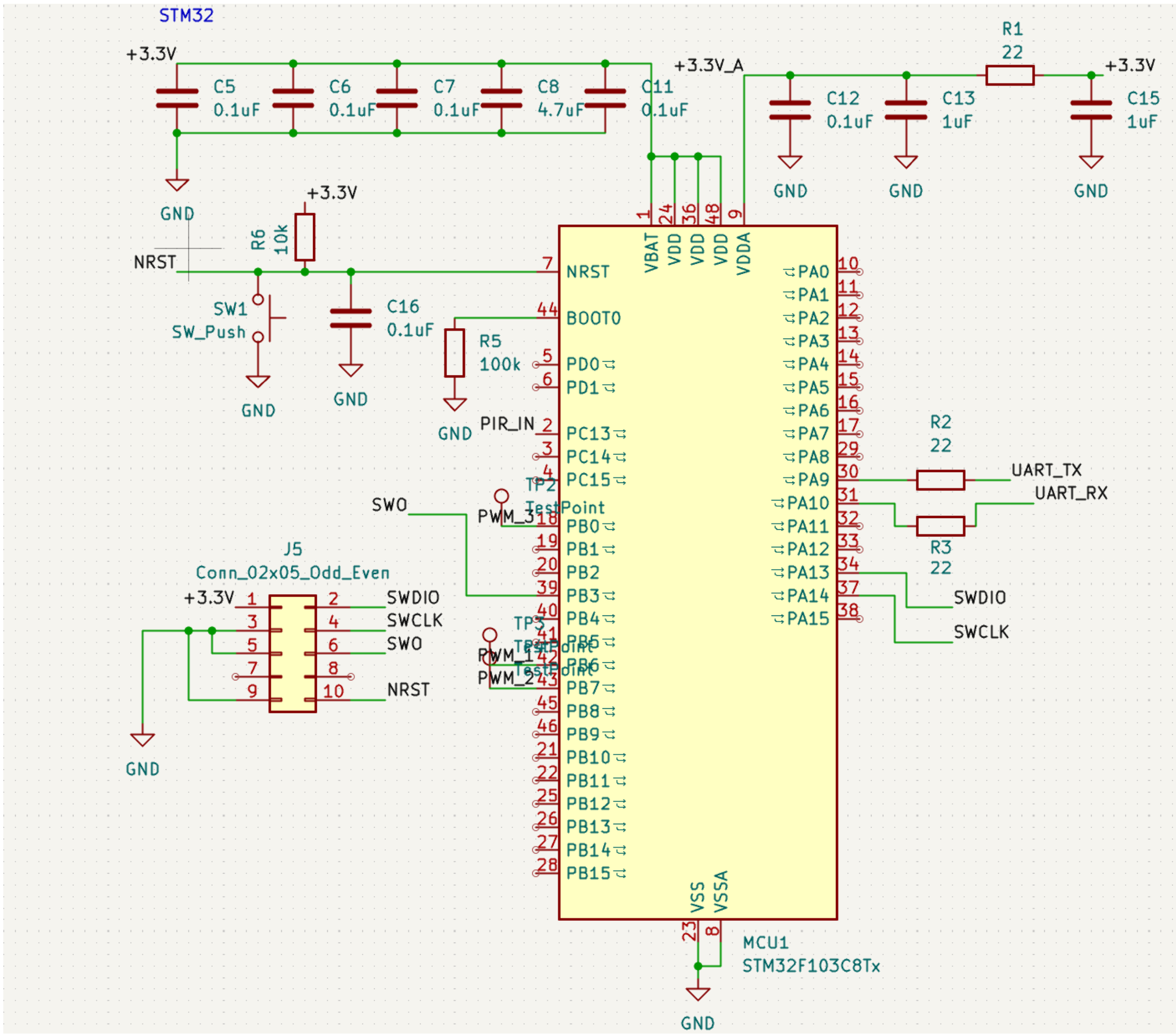


**Figure 6 STM32 Microcontroller Schematic**

The microcontroller controls the movement of the two Servo motors by generating PWM signals to the motor drivers, which translate these signals into motor currents.

The microcontroller controls the LED through 6 states: OFF, Illuminate, Bright, dim1, Standby, and dim2. OFF is self-explanatory: the LED is off. Illuminate means that the LED is gradually turning on after detecting motion. In the Bright State, the LED stays at its max brightness. In dim1, the LED begins to gradually turn off. In Standby, the microcontroller analyzes whether any extra movement is seen before fully turning the light off, and once it finds that there is still no movement, it sends a signal to turn off the light. Appendix B shows a flow chart of this process.

The microcontroller will need to have a GPIO input to receive data from the PIR sensor. It will also need a UART_RX and UART_TX signal to receive and communicate with the mmWave sensor. It also will need to send out 2 PWMs to the motors, which will use 2 more outputs on the microcontroller. Lastly, it will need to send out PWMs to the LED Driver. The microcontroller that we will be using is the STM32F103C8. This is an STM based microcontroller, and has 48 pins. It also has enough timers, PWMs, and GPIOs for our implementation.

The schematic of the microcontroller subsystem was designed with three necessities in mind: the signals we needed, programming capabilities, and decoupling/noise management. Our STM32 has 3 VDD lines, and 1 VBAT line. To ensure our MCU is powered with a clean 3.3V, we decided to use 1 0.1 uF decoupling capacitor per line as well as one 4.7uF bulk capacitor. There is also a VDDA line, which powers the Analog functions of the MCU. Even though we do not use any of the analog functions in our project, it is necessary to power the VDDA for proper chip function. Thus, we also include decoupling capacitors for this line, as well as a small resistor (which would function the same as a ferrite bead).

The signals portion of our schematic is pretty straightforward; maintaining clear signal names and choosing the right pins on the STM32 to use was pretty much the bulk of it. For two signals specifically, the UART signals connected to our mmWave, we decided to include 22 Ω termination resistors in series with the signals. The primary purpose of this was to prevent any line impedance or reflection issues as these signals were vital for our project.

The final connections on our MCU Subsystem are intended for programming the STM32. This includes SWDIO, SWCLK, SWO, NRST, +3.3V, and GND. These lines were connection to a single 2x5 connector, which was then connected to an external ST-LINKV2 (STM32 USB Programmer). The NRST was specifically important initially; we had it unconnected as it was deemed an “optional” pin. Once we ran into initial issues with programming, we added a push-button to allow for manual resetting; this solved our programing issues and allowed us to communicate properly with the STM32 CubeIDE Software.

## 2.5 Lighting Subsystem

Our lighting subsystem’s is supposed to keep the lights on, gradually turn lights off and on, and keep the lights off. Our microcontroller will generate a PWM that is proportional to the desired brightness, and will use a LED driver to use this PWM. For example, a 90% duty cycle should translate to 90% brightness of the LED. The LED Driver will output the correct current and voltage based on this PWM.

The LED Driver is able to set the correct voltage at the FB pin by sensing this voltage. If the voltage sensed at the FB pin is too high, then the duty cycle of the switch at the SW pin goes down, so that the inductor current has less time to build up. If the voltage at the FB pin is too low, then the duty cycle of the switch at the SW pin goes up so that the inductor current has more time to build up.

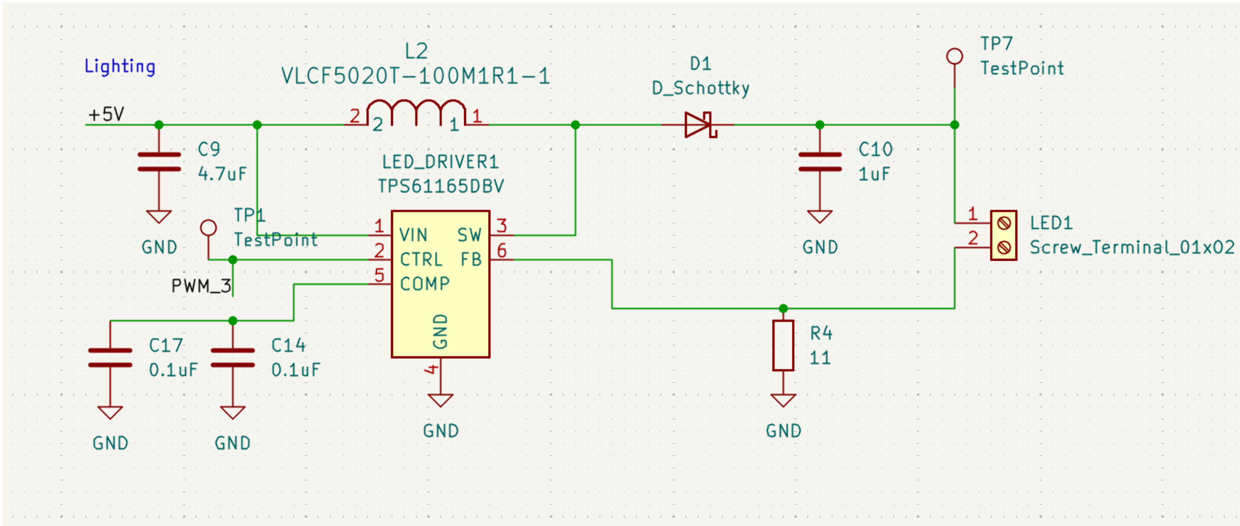


**Figure 7 LED Driver Schematic**

The lightning system centers around the TPS61165DBVR chip from Texas Instruments, which is a high brightness white LED driver. The circuit was designed based on the typical application circuit found in the chip's data sheet [11]. The input and output capacitors, C9 and C10 respectively, are used for decoupling purposes; they help to meet the chip's requirements for ripple and loop stability. Their values were chosen based on recommendations made by the datasheet. Capacitors C17 and C14 are compensation capacitors, and are used to stabilize the feedback loop of the chip. The recommended value of the compensation capacitor is 220 nF, which isn't a capacitance we have access to through the Electronic Services shop. Thus, we have chosen to use two 0.1 uF capacitors in parallel to get close to the recommended value. The inductor is arguably the most important component in power regulator design; it affects steady state operations as well as transient behavior and loop stability. The inductor was chosen from the table of recommended inductors on the data sheet. The Schottky Diode helps fulfill the high speed rectification that is required because of the high switching frequencies of the TPS61165. The resistor is a current-sense resistor, whose value was determined by the feedback regulated voltage and the LED Current Ratings. Put together, this circuit allows our microcontroller to control 3 LEDs through PWM outputs.

We measured the current going through the LEDs by using a lab power supply, and supplying a sufficient voltage, and reading the current on the lab power supply. This current was around 20mA, so we needed to set the current going through the LEDs to be around 20mA. The chip's datasheet states that FB is set to 0.2V, which gives the current through the LEDs. The following equation can be used to find the correct Rset value (R4 from the schematic):

$$\mathrm{I} = \frac{0.2}{R_{set}} \tag{2}$$

Using this equation, and 20mA as the current, the value for Rset should be 10Ω. However, the electronic services shop's lowest resistor was 22Ω, so we decided to use two 22Ω resistors in parallel, to get to 11Ω.

# 3. Design Verification

## 3.1 Power Subsystem

Requirements 1-2 in Table 10 pertain to the Power Subsystem (Appendix A). Requirement 1 pertains to the ability of our AC/DC Converter to generate a stable 5V output from the 120VAC AC Mains. This 5V output needs to be stable to ensure that components (such as our HS-318 motors) receive a stable voltage input for power. We found that the largest value was 5.01V, and the smallest value was about 4.94V, which is within our 5% tolerance.

Requirements 2 pertains to the ability of our 5V to 3.3V LDO to generate a stable 3.3V output from the 5V output that comes from the AC/DC Converter. Similar to the 5V output, this 3.3V output needs to be stable to ensure that components (like our STM32 Microcontroller) receive a stable input for power. We found that the largest value was about 3.32V, and the lowest was about 3.28V, which is within our 5% tolerance.


Figure 8 5V and 3.3V readings

## 3.2 Motor Subsystem

Requirements 3-5 in Table 10 pertain to the Motor Subsystem (Appendix A). Requirement 3 is there to ensure that our sensor can do a full sweep of the room. While we cannot put a video showing that the motors rotate 180 degree horizontally, and 30 degrees vertically, Table 2 gives a few starting and ending angles (we completed the test multiple times) that verify that the motors were able to sweep 180 degrees. Table 1 can be used to prove that the tilt motor is able to tilt 30 degrees. A protractor can be used to accurately measure up to a single degree, so this is what we used to measure.

**Table 1 Starting and Ending Angle Measurements (Tilt)**

| Test Number | Starting Angle (Vertical) | Ending Angle (Vertical) |
|---|---|---|
| 1 | -31° | -1° |
| 2 | -31° | -2° |
| 3 | -32° | -2° |

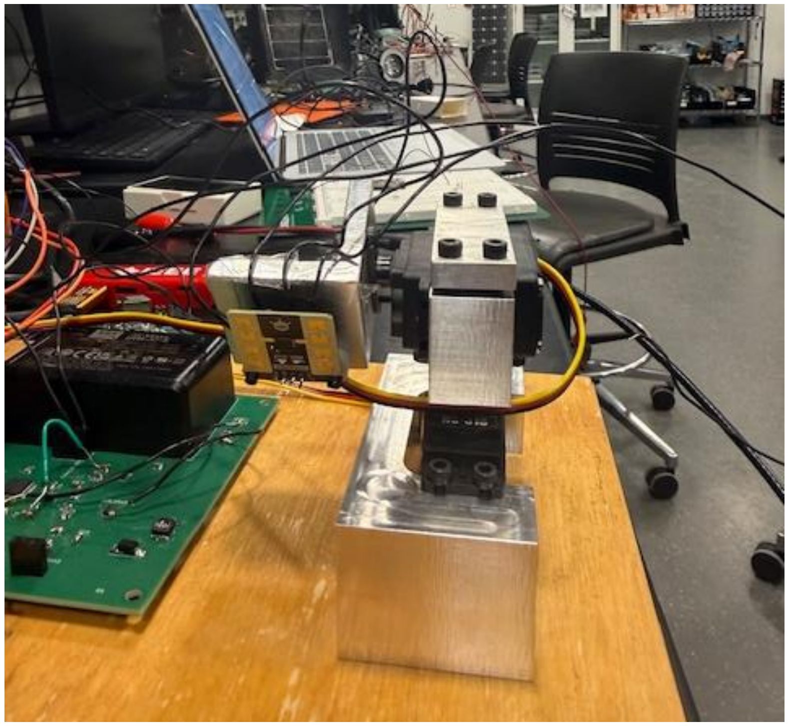

Figure 9  Motor Ending Point

Table 2 can be used to prove requirement 5 as well, which is used to prove that the settled point at each end is within 5 degrees each time. Lastly, this table can be used to also prove requirement 4. This requirement is present to ensure that no matter where the motor ends, the motor can start in the same position.

**Table 2   Starting and Ending Angle Measurements**

| Test Number | Starting Angle (Horizontal, Vertical) | Ending Angle (Horizontal, Vertical) |
|---|---|---|
| 1 | -2°, -31° | 178°, -31° |
| 2 | -2°, -31° | 179°, -31° |
| 3 | -1°, -31° | 178°, -32° |

## 3.3 Sensing Subsystem

Requirements 6-8 in Table 10 pertain to the Sensing Subsystem that features a dual-sensor approach consisting of an mmWave and PIR sensor. The motivation behind our 6$^{th}$ requirement is to ensure high PIR accuracy (low error rate) and ensure that it can detect up to the specified range of our design. In order to verify this we conducted movement tests from various distances to identify the drop rate of the PIR. In other words, the rate at which the PIR was not detecting movement when it should have been.

**Table 3   PIR Presence Detection Error Rate**

| Distance from Target | Drop Rate (Out of 20 Trials) |
|---|---|
| 5 ft | 0% |
| 10 ft | 0% |

| | |
|---|---|
| 15 ft | 5% |
| 20 ft | 10% |

When attempting to verify Requirement 7, we ran into a few issues with respect to the reliability of the mmWave sensor, especially at longer distances. Upon conducting our detection testing, we came to realize that the mmWave sensor had a very high false positive rate (detecting movement when there is no moving target in frame) and negative rate (detecting no movement when there is a target in frame).

**Table 4 mmWave Presence Detection False Positive Rate**

| Distance from Target | False Positive Rate (Out of 20 Trials) |
|---|---|
| 10 ft | 25% |
| 20 ft | 25% |
| 25 ft | 35% |
| 32 ft | 40% |

**Table 5 mmWave Presence Detection False Negative Rate**

| Distance from Target | False Negative Rate (Out of 20 Trials) |
|---|---|
| 10 ft | 20% |
| 20 ft | 25% |
| 25 ft | 45% |
| 32 ft | 50% |

Finally, with respect to the sensor subsystem, our final requirement was to ensure that there was at most a 1 second difference between the rising edge of the PIR channel and the rising edge of the PWM sent to the LED Driver. When making this requirement, we thought that the propagation delay would be much larger than it was. In reality, it was basically instant.

## 3.4 Microcontroller Subsystem

Requirements 9-11 in Table 10 pertain to the Microcontroller Subsystem (Appendix A). Requirement 9 pertains to our MCU’s ability to generate stable PWM waves with an increasing duty cycle from 0% to 100%. This PWM is meant to be sent to the LED Driver to allow for stable and gradual illumination of our LEDs. This requirement was validated with the respective procedure, and the correct PWM build-up was observed on an oscilloscope. Since we cannot insert a video, three pictures are used instead:

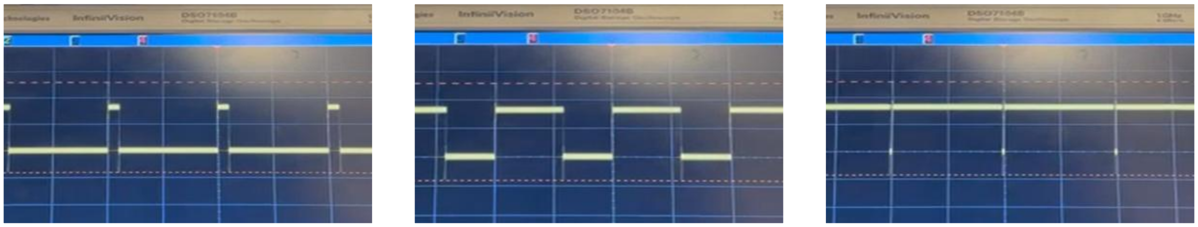

Figure 10  MCU PWM Duty Cycle

Requirements 10 and 11 are meant to show functionality between the mmWave/PIR sensors, MCU, and LED Drivers. Through separate validation, we were able to prove that the LED portion of these requirements worked: the LEDs were able to hold their full brightness for 30 minutes and were able to be dimmed to 50% duty cycle, and then OFF. However, the mmWave sensor portion of these requirements were unsuccessful. Due to trouble with our physical mmWave sensor, we could not consistently detect humans enough to reliably have the LEDs respond properly. The PIR sensor did reliably turn the LEDs on, but the mmWave did not properly shut the LEDs off.

## 3.5 Lighting Subsystem

Requirements 12-14 in Table 10 pertain to the Lighting Subsystem (Appendix A).  Requirement 12 ensures that a current of 18.18 mA (with tolerance) is going through the LEDs, to simulate full brightness. By measuring the voltage at the FB pin, and using a resistance of 11Ω, we can find the current going through the LEDs. We got a max voltage of about 0.205V, and a low voltage of 0.193V, which is within our 10% tolerance for 18.18 mA.

Requirement 14 can also be proven based on the max and min voltage being between 0.205V and 0.193V. These values ensure a current ripple of 18.18mA with a tolerance of 20%. In addition, the voltage at the input of the LED Driver did not differ too much from the 5V at the AC/DC. It had a little bit of a different max and min value for the voltage (5.042V max, 4.983V min), but this also passed this requirement.


Figure 11  Voltage reading at FB Pin for 100% Duty Cycle

Requirement 13 ensures that the other duty cycle values correspond to the correct brightness. For example, a duty cycle of 50% should give 50% brightness, which corresponds to a FB voltage of about 0.1V. Table 6 gives the average voltage and corresponding current that we found during measurements when we sent duty cycles of 10%, 50%, and 90%.

**Table 6   Current through LEDs at various Duty Cycles**

| Duty Cycle | Voltage (Average) | Current |
|---|---|---|
| 10% | 0.022 V | 2 mA |
| 50% | 0.106 V | 9.63 mA |
| 90% | 0.185 V | 16.82 mA |

# 4. Costs

## 4.1 Labor and Parts

In our labor cost estimation, we will use the formula present on the design document overview on the ECE 445 website and assume 6 hours worked per week. We will also assume that an average ECE grad from UIUC makes $40/hr.

$$\text{hourly rate} * \text{actual hours spent} * 2.5 = \text{Labor Costs} \quad (3)$$

**Table 7 Labor Costs**

| Team Member | $/hr | Hours Worked per Week | Weeks Worked | Total Cost ($) |
|---|---|---|---|---|
| Joseph Paxhia | 40 | 6 | 14 | 8400 |
| Lukas Ping | 40 | 6 | 14 | 8400 |
| Sid Boinpally | 40 | 6 | 14 | 8400 |
| **Total** | | | | **25200** |

In addition, we also have added costs based on the components that we plan to use in our project. We have included the costs of everything non-standard and have not included parts such as resistors and capacitors that will be sourced from the Electronics Services Shop.

**Table 8 Parts Costs**

| Part | Manufacturer | Retail Cost ($) | Bulk Purchase Cost ($) | Quantity | Actual Cost ($) |
|---|---|---|---|---|---|
| C4001 mmWave Sensor | DFRobot | 13.90 | 13.90 | 1 | 13.90 |
| PIR Sensor (ID 189) | ADAfruit | 9.95 | 7.96 | 1 | 9.95 |
| HS-318 Motor | ServoCity | 11.99 | 11.99 | 2 | 23.98 |
| Micro Controller - STM32F103C8 | STMicroelectronics | 3.26 | 3.26 | 1 | 3.26 |
| IRM-30-5 AC/DC Converter | MEAN WELL USA Inc. | 12.50 | 11.10 | 1 | 12.50 |
| NEMA 5-15P to Standard Roj Power Cord | CableWholesale | 9.21 | 9.21 | 1 | 9.21 |
| 2-pos terminal block (1715721) | Phoenix Contact | 1.59 | 0.92 | 4 | 6.36 |
| TLV76733DRVR LDO 5 to 3.3V | Texas Instruments | 0.94 | 0.46 | 1 | 0.94 |
| TPS61165DBVR LED Driver | Texas Instruments | 2.15 | 1.14 | 1 | 2.15 |
| ED1543-ND Connector ST (02x05) | On Shore Technology Inc. | 0.26 | 0.14 | 1 | 0.26 |
| PPTC041LFBN-RC | Sullins Connector | 0.37 | 0.19 | 1 | 0.37 |

| | | | | | |
|---|---|---|---|---|---|
| Connector ST (4 pos) | Solutions | | | | |
| PPPC031LFBN-RC 3 Connector ST (3 pos) | Sullins Connector Solutions | 0.31 | 0.16 | 3 | 0.93 |
| VLCF5020T-100M1R1-1 10uH Inductor | TDK Corporation | 0.59 | 0.40 | 1 | 0.59 |
| STLink Programmer | AITrip | 5.59 | 5.59 | 1 | 5.59 |
| **Total** | | | | | **89.99** |

Therefore, the total cost, with labor included, is 89.99+25200 = $25289.99.

## 4.2 Schedule

**Table 9 Schedule**

| Week | Actions |
|---|---|
| Week of 10/13 | 1. Design Schematics and PCB in time for 2nd PCB Design Review (For inlet board) **(ALL)**<br>2. Complete all orders for parts (including STLink programmer for microcontroller) **(ALL)**<br>3. Complete teamwork evaluation **(ALL)**<br>4. Finalize revisions for assembly needed from machine shop (servo motor fixture) **(ALL)** |
| Week of 10/20 | 1. Begin testing and interfacing with mmWave sensor to set up configuration as desired using Arduino Uno **(Sid)**<br>2. Setup breadboard demo to accommodate servo motors along with mmWave sensor through microcontroller interface **(ALL)** |
| Week of 10/27 | 1. Breadboard demo first half of the week **(ALL)**<br>2. Begin Soldering on Components onto PCB **(Lukas and Sid)**<br>3. Verify current draws of individual components aligns with expected values that were used in trace calculations **(Lukas)**<br>4. Program microcontroller to interface with servo motors, sensors, and LED driver **(Sid)**<br>a. Start testing basic functionality of software (UART with mmWave, PWM outputs to motors and LED driver) **(Sid)**<br>5. Test and debug board with 120V AC power supply and modify PCB design to address any discrepancies **(Joey)** |
| Week of 11/3 | 1. Finalize PCB modifications of main board **(Sid)**<br>2. Test LED driver functionality with external LED breadboard with properly configured load parameters **(Joey)**<br>3. Complete individual progress reports **(ALL)**<br>4. Look into better LEDs **(Lukas)** |
| Week of 11/10 | 1. Start full design testing and ensure that high level requirements are met (they were not) **(ALL)**<br>2. Adjust mechanical setup in accordance with room layout and configuration **(ALL)**<br>3. Finalize lab notebooks and prepare for Mock Demo **(ALL)** |

| | |
|---|---|
| Week of 11/17 | 1. Prepare for Final Demos and Mock Final Presentation **(ALL)** |
| Week of 11/24 | **FALL BREAK** |
| Week of 12/1 and 12/8 | 1. Present Final Presentation **(ALL)**<br>2. Write Final Paper **(ALL)**<br>3. Submit Lab Notebooks **(ALL)** |

# 5. Conclusion

## 5.1 Accomplishments

Over the semester, we were able to design and bring up a functioning two board PCB system with correct power electronics behavior. The inlet board in our design safely converts 120 VAC mains to regulated 5 V and 3.3 V rails, and our lab measurements showed that our 6W load had less than 50 mVpp ripple on the 5 V rail. On the main board, the STM32 microcontroller, LED driver, servo headers, and sensor interfaces all behaved the way we expected, which let us run closed loop control of the light level based on motion sensing. In the end, we tied together AC to DC conversion, DC to DC regulation, an LED driver, PIR and mmWave sensing, and the pan/tilt servo scan into a single prototype that powered up cleanly and ran our code without brownouts or random resets.

Beyond the hardware itself, this project grew our hands on engineering skills a lot. We split up responsibilities so that power electronics, firmware, and mechanical work all moved in parallel, but we still stayed in sync with weekly integration goals. That made a big difference in our overall planning and execution. By the second PCB spin we went from multiple bring up issues, such as burned out components or missing pull ups, down to zero showing that our process improved. We also got comfortable with standard lab gear like oscilloscopes, multimeters, and programmable supplies, we practiced reflow and hand soldering fine pitch parts, and we learned how to program and debug the microcontroller in CubeIDE using the ST-Link programmer. Being flexible and clear about who owned what and working side by side in collaborative sessions allowed us to end up with a mostly working prototype by the end of the term.

## 5.2 Uncertainties

The biggest open technical question is still the behavior of the C4001 mmWave radar when it sits on the pan or tilt mechanism. On the bench, with the sensor fixed and the servos completely off, it did pretty well, it correctly reported motion versus no motion in roughly 88 percent of 50 trials at a distance of about 3 to 4 meters. Once we mounted it on the moving platform, things appeared differently. In about 40% of pauses within the pan/tilt process, it still reported motion even though there was nobody in view and the servos were supposed to be holding position. Our best guess is that small vibrations and micro steps from the servos, on the order of half to one degree of jitter, show up in the radar returns and look like micromotion. That makes the presence detection less reliable than we would like.

A smaller but still important uncertainty is the perceived brightness of our LED. Electrically, the LED driver did what we asked, the measured LED string current followed our duty cycle within plus or minus 5%. Even so, people told us the light looked dim from the side, and lux measurements at desk height came out around 120 to 150 lx at full output. Typical recommendations for comfortable task lighting are closer to 300 lx. So while we hit the goal of controllable dimming and proper driver behavior, the current optical setup might not feel bright enough for real users in all room layouts.

## 5.3 Ethical considerations

Future versions of this project should focus on separating the mmWave sensor from servo jitter and on improving the lighting so it is closer to something you could actually ship. One simple change would be to add a digital switch, for example a MOSFET load switch controlled by the MCU, that cuts power to the servos during a "stagnant observation" window. After the platform finishes a sweep, the controller could turn off servo power for one to two seconds while the radar looks for motion. That would remove motor jitter and let us retest the sensor's false alarm rate with truly still mechanics. Another option is to mount the sensor with some kind of damping material, or to change the scanning pattern so we only stop at a few fixed viewpoints that have time to settle before we read the radar.

On the lighting side, future work should look at brighter LED packages and maybe a driver that can source higher current. Right now we run at roughly 80 mA per string, bringing that up to 150 to 200 mA with proper thermal design could more than double our measured lux and push us closer to the 300 lx target. Finally, our user feedback so far is based on around 10 informal tests that weren't entirely conclusive given the limitations of our design.

## 5.4 Future work

Since our system is always checking whether a room is occupied, privacy is one of the main ethical issues. Following the IEEE Code of Ethics, we designed the system so it collects as little information as possible. The PIR and mmWave sensors only feed into a simple occupancy decision, occupied or vacant, and we never capture images, audio, or anything that can identify a specific person. All of the processing happens locally on the STM32, and occupancy is just a flag in RAM, it is not logged to flash and it is not sent off the device.

We also thought about how the system could be misused, so we kept what the sensors output as simple as possible. The mmWave sensor only talks to the microcontroller over UART and all it sends is whether it sees presence or not, there is no information about who the person is, how many people there are, or anything about their behavior beyond a basic motion state. The PIR is even simpler, it just provides a single digital logic signal that flips when it detects motion in its field of view. There are no images, no audio, and no raw radar data stored or streamed off the device. If this ever became a real product, we would make sure the documentation clearly explains that the sensors only feed into a basic occupancy decision and that no personal information is recorded. By keeping the data this limited, our goal is to get the energy savings and convenience of smarter lighting without collecting or exposing anything sensitive about the people in the room.

## References


[1] Association for Computing Machinery, “ACM Code of Ethics and Professional Conduct.” [Online]. Available: https://www.acm.org/code-of-ethics

[2] Institute of Electrical and Electronics Engineers, “IEEE Code of Ethics.” [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html

[3] UL Standards & Engagement, “UL 8750: Light Emitting Diode (LED) Equipment for Use in Lighting Products.” [Online]. Available: https://ulstandards.ul.com/standard/?id=8750

[4] American National Standards Institute, “ANSI C137.1-2022: Lighting Systems – 0 to 10 V Analog Control Protocol.” [Online]. Available: https://www.ansi.org/standards-activities/standards-coordinating-committees/c137-lighting-systems

[5] U.S. Federal Communications Commission, “47 CFR Part 15 – Radio Frequency Devices,” Electronic Code of Federal Regulations. [Online]. Available: https://www.ecfr.gov/current/title-47/part-15

[6] Texas Instruments, “SLUP239A (application report).” [Online]. Available: https://www.ti.com/lit/ml/slup239a/slup239a.pdf

[7] DFRobot, “C4001 24 GHz mmWave Human Presence Detection Sensor – Product Page.” [Online]. Available: https://www.dfrobot.com/product-2793.html

[8] Mean Well USA via Digi-Key, “IRM-30-5 – 5 V, 30 W AC-DC Converter (product page).” [Online]. Available: https://www.digikey.com/en/products/detail/mean-well-usa-inc/IRM-30-5/7704672

[9] Texas Instruments, “TLV767 Low-Dropout Linear Regulator – Datasheet.” [Online]. Available: https://www.ti.com/lit/ds/symlink/tlv767.pdf

[10] Mean Well, “IRM-30 Series – Specification Sheet (datasheet).” [Online]. Available: https://www.meanwellusa.com/upload/pdf/IRM-30/IRM-30-spec.pdf

[11] Texas Instruments, “TPS61165 – White LED Driver (product page/datasheet).” [Online]. Available: https://www.ti.com/lit/gpn/TPS61165

# Appendix A Requirement and Verification Table

**Table 10 System Requirements and Verifications**

| Requirement | Verification | Verification status (Y or N) |
|---|---|---|
| 1. The power subsystem must be able to accept 120 VAC from AC Mains in a building and convert it to 5V, with a tolerance of ±5%, for a continuous current load of up to 3A. | 1. Wait enough time to ensure that the circuit has been properly discharged.<br>2. Plug in the AC Mains to start supplying power to the circuit. Wait a few seconds for components to power up.<br>3. Measure the voltage output at the 120VAC-5VDC converter to ensure that this is within ±5% of 5V. This will be measured via DMM.<br>4. Apply some sort of motion near the PIR sensor, and wait for the lights to turn on (fully), and ensure they stay on.<br>5. Next, wait for the motors to start moving, and while this is occurring, measure the voltage output at the 120-VAC-5VDC converter and ensure this is within ±5% of 5V (this is because the largest output current occurs when the LEDs are on, and the motors are moving). This will be measured via oscilloscope. | Y |
| 2. The power subsystem must be able to accept 5V (±5%) and output 3.3V with a tolerance of ±5% through the LDO during continuous current of 1A. | 1. Power up the circuit through the AC Mains. Ensure that the components are fully dissipated before powering back up to ensure safety.<br>2. Turn on the light by applying motion near the sensors.<br>3. Measure the output of the LDO through oscilloscope/DMM and ensure that it is within the requirement. | Y |
| 3. The pan motor must be able to rotate 180 degrees through mcu programming, and the tilt motor must be able to rotate 30 degrees (tolerance of 5%). | 1. Plug in the AC power and wait for the motor sequence to start.<br>2. Once the motor has stopped at one edge, unplug power to stop the sequence.<br>3. Measure the angle of the pan and tilt motor during the off period with | Y |

| | | |
|---|---|---|
| | a protractor.<br>4. Plug in AC Power again and wait for the motor to pan to the other side and stop.<br>5. Unplug power again, and measure the angles of both the pan and tilt motors to ensure that they are 180 and 90 degrees away from the starting position. | |
| 4. Both the pan and tilt motors must be able to start in the same position (within ±5 degrees) | 1. Measure the desired starting angles (in our case it should be 0 degrees horizontally and vertically, so there is really no need to measure this).<br>2. Plug in AC power, and wait for the motor rotation to begin.<br>3. Wait for the motor to sweep and tilt at an angle that is not the same as the starting angle.<br>4. Unplug power<br>5. Plug AC Mains power back in, and once the motors move back to the start, unplug power before it moves again. Then, measure the angle that it starts at. | Y |
| 5. For both servos (pan and tilt), the settled angle at each endpoint shall vary by no more than 5.0 degrees across repeated commands. | 1. Measure the desired starting angles (in our case it should be 0 degrees horizontally and vertically, so there is really no need to measure this).<br>2. Plug in AC power, and wait for the motor rotation to begin.<br>3. Wait for the motor rotation to sweep 180 degrees one way.<br>4. Once it has swept 180 degrees one way, unplug power and measure the angles to ensure they start and end at the same angle (within our stated tolerance). | Y |
| 6. The PIR sensor must be able to detect human presence at a distance of 20 ft. or less. (90% of the time) | 1. Stand 20 feet away from the PIR sensor.<br>2. Have somebody else plug in the AC power.<br>3. While staying around 20ft away, make motions to see if the PIR can detect and turn on the light (you need to move to turn on the light since PIR can't detect standstill motion).<br>4. If the light has turned on, that | Y |

| | | |
|---|---|---|
| | means that this is a successful test. You can also test this by probing the output from the PIR sensor and seeing if a high is output from the sensor. | |
| 7. The mmWave sensor must be able to detect human presence at a distance of 32ft or less. (90% of the time) | 1. Plug in the AC Power.<br>2. Stand near to the PIR sensor, and start moving, and wait for the light to turn on.<br>3. Once the light has turned on, move to a distance of 32 ft. away from the sensor.<br>4. Wait for 5 minutes. If the light has not turned off, this means that the mmWave is able to detect a human presence. One can also observe the output light on the mmWave.<br>Alternative:<br>1. Stand 32 feet away from the mmWave.<br>2. Plug in the mmWave to the Arduino Uno.<br>3. If the Arduino Uno prints "exist motion", it is able to detect presence at this range. | N |
| 8. There is at most a 1 second difference between the rising edge of the PIR channel and the rising edge of the PWM sent to the LED Driver. | 1. Make sure that the sensors are powered through AC power.<br>2. Connect an oscilloscope to probe the output of the PIR, and in channel 2, probe the output of the PWM that sends to the LED Driver. You can also probe with a DMM.<br>3. Standing near to the PIR sensor, do some sort of motion (walk, wave, etc.).<br>4. Once this has happened, look at the oscilloscope and ensure the time between the rising edge of the PIR channel and the rising edge of the PWM sent to the LED Driver is less than 1 second. | Y |
| 9. The MCU must be able to generate a PWM signal to send to the LED driver, and must vary from a duty cycle of 0-100% during the gradual illumination stage. | 1. Plug in AC power to the lighting project.<br>2. Probe the output that feeds to the LED Driver from the microcontroller with an oscilloscope.<br>3. Make motion near the PIR sensor, and observe the oscilloscope. | Y |

| | | |
|---|---|---|
| | 4. Observe to see if the output from the oscilloscope varies (gradually) from a PWM with a duty cycle of 0% to a duty cycle of 100%. | |
| 10. The MCU shall be able to dim and turn off an LED if no motion is detected after a full sweep. It should dim after 1 full sweep, and turn off after another full sweep. | 1. Plug in AC Mains<br>2. Turn on the LEDs by applying motion near to the PIR Sensor.<br>3. Once it has turned on, move away from the mmWave.<br>4. Observe the LEDs, while staying away from the mmWave sensor.<br>5. After 1 full pan, the LEDs should dim if no motion is detected.<br>6. After another full pan, the LEDs should turn off. | N |
| 11. The MCU shall be able to hold its Bright state for at least 30 minutes with people being in a room. | 1. Ensure that AC Power is plugged in to power the project.<br>2. Turn on the lights by applying motion near the PIR sensor.<br>3. Once this has occurred, start your stopwatch.<br>4. Stay in the room for 30 minutes, and do anything you really want, as long as you are not waiting.<br>5. If the LED stays bright for 30 minutes, the test has passed. | N |
| 12. When CTRL is held HIGH (100% duty), the LED current should be 18.18 mA ±10%. | 1. Plug in the AC Mains.<br>2. Set CTRL to 100% Duty. The LED current should be about 18.18 mA. Verify by measuring the feedback voltage V_FB across R4 and calculating I_LED = V_FB / 11 ohms.<br>3. The test passes if the calculated current is between 16.362 mA and 20 mA (±10%) and the converter is stable. | Y |
| 13. When CTRL is a 1 kHz PWM (0–3.3 V), the average LED current should scale with the duty cycle D. At D = 10%, 50%, 90%, the measured average current must be within ±10% of 0.1*I_SET, 0.5*I_SET, 0.9*I_SET respectively. | 1. Drive CTRL from a function generator and set each duty. Ensure that the MCU is not programmed to send a PWM and the LED Driver is still powered.<br>2. On the scope, smooth the FB node (use a bandwidth limit) to get V_FB,avg. Compute I_LED,avg = V_FB,avg / 11 Ω.<br>3. Pass if the results are within ±10% of the expected values at each duty.<br>4. Targets: | Y |

| | | | |
|---|---|---|---|
| | a. 10% duty → 1.818 mA (accept 1.64–2mA)<br>b. 50% duty → 9.09 mA (accept 8.18–10 mA)<br>c. 90% duty → 16.362 mA (accept 14.726–18 mA) | |
| 14. With CTRL = 100% and the normal load attached:<br>a. The LED current ripple must be ≤ 20% peak-to-peak of the targeted LED current I_SET.<br>b. The input-pin ripple (VIN at the IC) must be ≤ 100 mV peak-to-peak. | 1. Plug in AC Power, and activate the PIR to turn on the LED. Keep the LEDs on.<br>2. Attach an oscilloscope to measure the input voltage to the LED Driver.<br>3. Observe the oscilloscope or DMM, and ensure that the voltage is ≤ 100 mV peak-to-peak.<br>4. Now, attach an oscilloscope to the FB output, or probe FB through a DMM. Ensure that the FB voltage is within 20% of 0.2V. This ensures that the current ripple is ≤ 20% peak-to-peak. | Y |

# Appendix B Lighting Process Flow Chart