Crowd Monitoring Device

By

Armando Juresic, juresic2

William Schellhorn, schellh2

Final Report for ECE 445, Senior Design, Fall 2016

TA: Luke Wendt

7 December 2016

Project No. 20

Abstract

By the time of demonstration, our crowd monitoring device fulfilled the functionality listed in the Requirements and Verifications Appendix with slight difficulty in the power module. We showed functionality of essential modules: our transmitter array, our receiver array, our analog to digital circuit, our control module, our power module, and our central server. This progress offers a step toward future development, which is discussed in the Conclusion section of the report.

Contents

1 Introduction

1.1 Motivation

Computer vision is limited by a camera viewing objects from an angle, not top-down. Because of this, targets may overlap, preventing distinction. Spherical cameras are available for viewing nearby, cramped scenes. They are commercially available, but expensive. They also require a heavy software and mathematical effort to map a spherical projection to a plane. Faces and shapes of bodies, a big part of some computer vision, can get distorted.

We would like to create a modular tile that can hook together on a ceiling and save researchers money they would have otherwise spent on chaining together Microsoft Kinects. Ideally we could provide information on human clusters moving through space and tell computer vision algorithms running on a professor's laptop where to look in his high definition camera's frame.

For our demo, we shoot for a more attainable Demo Goal given the time we have to work – counting entrance and exit across one of our panels.

1.2 Objectives

We seek to make an array of sensors which will be directly above our targets. This way we can track motion across a space on a 1:1 scale without worrying about targets overlapping. This could be used in conjunction with installed computer vision to determine boundaries between people in a frame. Should we accomplish our Reach Goals specified as part of our design review revision, we will group activated pixels into blobs to follow the trajectory of a human blob through space. Ideally we could incorporate data collected by our array and processed with machine learning into computer vision.

Our project is a step toward sensor fusion of a ceiling-mounted sensor array and cameras placed at multiple viewing angles in a room. The goal of such a system is to track a target better than any individual sensing.

By the end of the project, we want to accurately count the number of people passing underneath one of our panels. We are then moving in the right direction toward our Reach Goals, which would include accurately counting the number of people within an area as long as we cover all entrances and exits. If a user were to tile an entire ceiling, they could map motion of specific individuals throughout a room.

2 Design

In revising our design review, we have devised Demo Goals and Reach Goals for each module. They are listed in Table 1. By time of demo, we will accomplish Step 1, the Demo Goals, for all modules. Reach Goals, Steps 2 and 3, may be met based on our success with Step 1 and continued research.

The prototype which will accomplish Step 1 for all modules has also been designed to accommodate Steps 2 and 3 should we have success in meeting Step 1 and time to further explore Steps 2 and 3. This will save us time spent building prototypes.

Our research toward accomplishing each step for each module is organized following the table below.

Table 1: Outline for Revision

Module	Step
IR Receiver Array with Receiver Cells	1. phototransistor array with tubes setting resolution 2. phototransistor array with pinhole aperture tubes setting resolution 3. phototransistor array with lenses setting resolution
IR Illuminator	1. LED array programmable by shift registers controlling switches; transmitted power calibration so received reflected power is above ambient power 2. LED array programmed with special patterns for finer distance sensing at Receiver Array
Sampling and Processing	1. sample all pixels into microcontroller memory; detect entrance and exit over panel area (if pixels are not saturated) 2. cluster on pixels with raster scan to identify finer features (head and shoudlers)
WiFi	1. send person count over Receiver Array to central server 2. send cluster information over Receiver Array to central server
Power (DC Supply and Buck Converters)	1. AC/DC, DC/DC

As described in our design review, we seek to take snapshot of the surface below a square area, detect whether distinct human figures are walking through the area, and report this information to assist in other computer vision projects. Our design effort is comprehensive. At a broad scale, it covers signal processing, digital systems, circuit design, and machine learning. We see it as a refreshment for past coursework and a culmination of our undergraduate understanding of electrical engineering. As the reader will notice, evolution from our initial Piazza post and Request for Approval to our current design review is evidence of a continually evolving design process.

2.1 Block Diagram

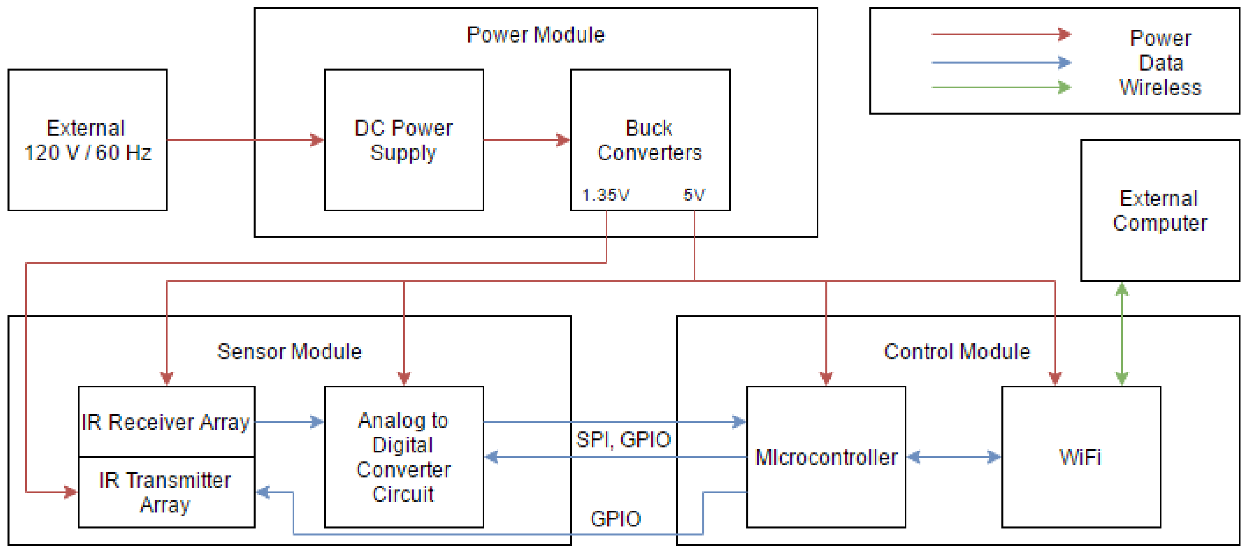

Figure 1: Block Diagram

2.2 Block Procedure, Details, and Verification

2.2.1 Sensor Module Overview

The sensor module is an active device meant to sense the height of a surface above the ground. It consists of two submodules: the infrared (IR) illuminator and the receiver array. The purpose of the IR illuminator is to flood the area beneath the receiver array with infrared light at an intensity such that received power $S_{received}$ seen at the receiver array reaches above the ambient IR power $S_{ambient}$ of the room. We define this threshold as $S_{received} = S_{ambient} + S_{flood,reflected}$. A large dynamic range between $S_{received}$ and $S_{ambient}$, the ambient noise, ensures we can sense the distance to objects between the floor and the array. An object between the floor and array will increase $S_{received}$ above $S_{ambient}$ since it causes $S_{splash,reflected}$ to increase in magnitude. The purpose of the receiver array is to sense an increase in $S_{received}$ relative to the initial $S_{ambient} + S_{splash,reflected}$ when no object is beneath the array. The increase is proportional to the square of the distance to the floor $R_{ambient}$ divided by the distance to the object $R_{reflected}$. That is,

$$S_{recieved} = (S_{ambient} + S_{splash,reflected})(\frac{R_{ambient}}{R_{reflected}})^2 \tag{1}$$

Because the module uses infrared light as an indicator, it should be deployed indoors, away from direct sunlight which would increase $S_{ambient}$ and limit our dynamic range. It is housed on a panel which should be mounted parallel to the floor to maximize sensing of reflected power. Figure 2 presents a drawing of what the sensor module might look like.

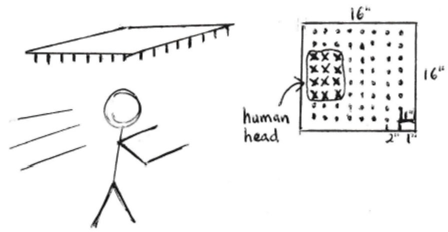

Figure 2: Person under sensor module and bottom view of sensor module

2.2.2 IR Illuminator

Step 1: Every receiver cell in the receiver array has a 940nm LED to splash the ground beneath it. Preliminary experimentation with free LTE-5228 LEDs from the ECE Service Shop demonstrates 16 LEDs at 2 inch spacing and 1.25V forward voltage already increase $S_{received}$ above $S_{ambient}$ at a height of 7 feet.

For Step 1, we must address the following question: Do the IR LEDs produce enough IR light to provide readings discernible above ambient conditions?

We utilize shift registers to create different levels of intensity by selectively turning on LEDs. We will turn off all IR LEDs to record the ambient IR conditions. Then, we will sequentially increase the levels until we see a voltage higher than ambient. Each level of intensity will correspond to a pattern of spacing between the LEDs.

Calibration of the illuminator's splash is discussed in Tolerance Analysis and a flowchart for calibration is included in Figure 3.

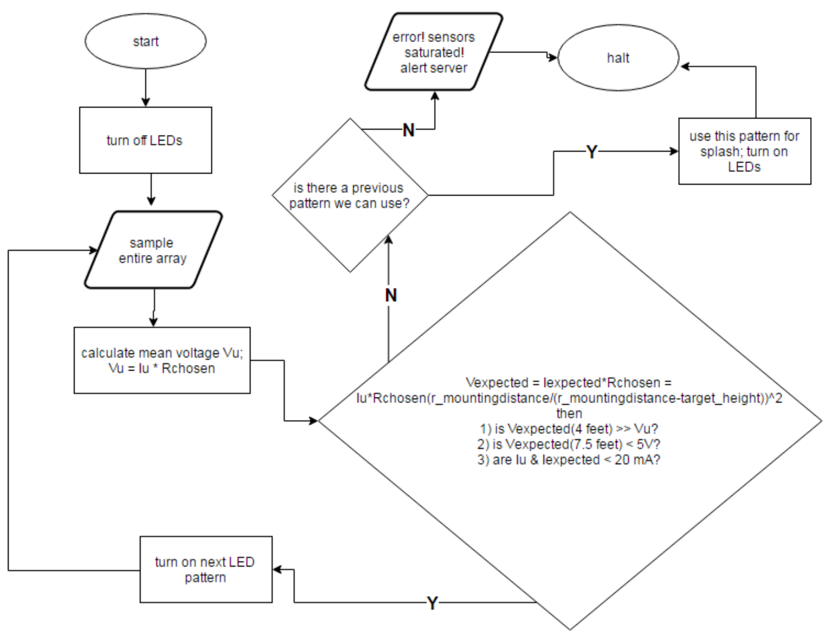

Figure 3: Calibration Flowchart

64 LEDs spaced 2 inches apart on a transmitter array can be individually turned on and off through the use of power logic shift registers. Reference [1] demonstrates the synchronized input of a vector to Serial In with control signals SRCK, SRCLR', RCK, and G' will allow us to pull any of 8 NMOS drains to ground or leave them floating. We attach IR LED cathodes to VDD and anodes to these drains. Since we want to control 64 LEDs, we need to chain 8 of these power logic shift registers together. To do this, we connect the Serial Out pin of the previous chip to the Serial In pin of the next. The pinout for our shift register is shown in Reference [1]. Finally, we breadboard the shift register chain, connect it to our Raspberry Pi general purpose input and output pins, and write code included in the Appendix to guarantee we can turn on or off 64 LEDs independently. Our test was successful. We can load a custom 64 bit pattern. This meets Requirement 6 of Appendix A. During final demonstration, we also sent a test vector of 8 LEDs into the transmitter array and verified they were on by noting a current draw of approximately 169 mA, an expected current from previous testing of LED test vectors, at 1.35 V.

Step 2: For Step 2, we must address the next question: When the waves constructively interfere does this amount of IR LEDs provide a consistent decrease in intensity as the distance from the array increases?

As you increase the distance from the array, the number of LEDs contributing intensity on a single point increases due to a 40° conical emission. Figure 4 shows this constructive interference between two LEDs as an example. To calculate how intensity changes on single point as that point increases in distance from the array, first we found how the relative intensity changes due to the angle. We extracted this information from the datasheet for the IR LEDs (LTE-5228) and plotted the data. Then we were able to fit an equation to the curve so we could use it in our calculations. We utilized that equation by writing a python script (Appendix E) to cycle through different spacings between the LEDs to determine which spacings have a consistent decreases in intensity as you increase the distance from the array.

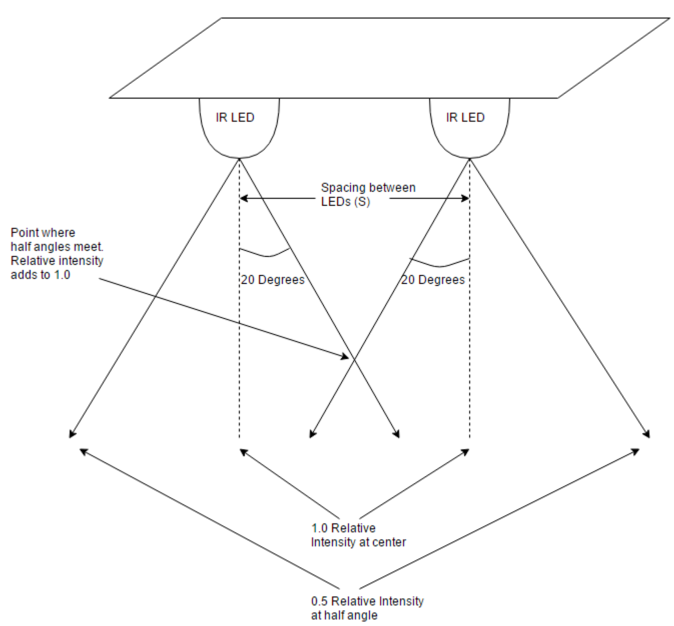

Figure 4: IR Illuminator Interference

2.2.3 Receiver Array

Step 1, 2, 3: The array is a square grid. We choose a square grid for accuracy in production. We can measure the distance between rows and columns instead of measuring the distance between each pixel, which could introduce human error.

To detect a human head, we want to see a rectangle where length is not equal to width. Then we can determine the line from forehead to back of head, the head length line. The line across shoulders will be perpendicular or oblique to the head length line. This gives a good starting point for looking for shoulders. Certain activations do not provide a rectangle: 1 pixel is a point, 2 pixels is a vector, 3 pixels is a triangle. With 4 pixels, we have a square. With 6 pixels we have a clear indication of a shape with length greater than width.

We choose 2 inch spacing between receivers so the smallest anticipated human head will span at least a 6 pixel rectangle. In the first percentile of the population, the human head is 5 inches wide by 7 inches long by Reference [2]. When the head length line is parallel to our grid's columns, we want a 2 by 3 pixel activation. Minimum spacing for width is $\frac{5inch}{2} = 2.5$ inches while minimum spacing for length is $\frac{7inch}{3} = 2.33$ inches. We round down to 2" spacing to satisfy both requirements. As Figure 5 demonstrates, we can also detect a head length line if the head is oblique – in the worst case, diagonal – to the grid's columns.

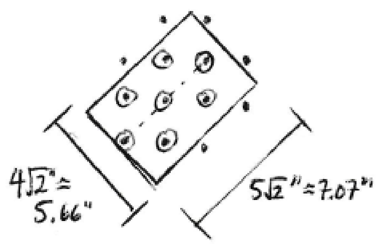

Figure 5: Head length line still detectable in diagonal case

A single array is 16 by 16 inches, containing 8 by 8 pixels. This is small enough to be portable. The design is modular so two arrays could be connected for a 32 by 16 inch configuration, which fits between a standard 34 inch door frame with the door open.

2.2.4 Receiver Cell

Step 1: Each receiver is an NPN phototransistor which conducts its highest current when exposed to 940nm light. The collector of the phototransistor is connected to Vdd, the emitter is connected to a resistor R_E, and the R_E is connected to ground. This constitutes a common collector configuration. Resistor sizing is discussed in the ADC Circuit subsection. By Reference [3], $S_{received}$ causes a directly proportional collector-emitter current. Current through the emitter resistor creates a voltage drop. This voltage $V_{received}$ is the signal we measure to represent $S_{reflected}$ so

$$V_{received} = (V_{ambient} + V_{splash,reflected})(\frac{R_{ambient}}{R_{reflected}})^2 \tag{2}$$

In order to minimize ambient infrared which would contribute to a higher $V_{ambient}$ reading, we first considered covering each phototransistor with a Plastidip-coated straw. This is too difficult to standardize.

To restrict the viewing angle of the phototransistor, we consider using a tube aperture, cone aperture, or a lens. A cone aperture would be difficult to fabricate. Lenses would be expensive and hard to mount to individual cells. For Step 1, we consider the tube aperture. It has two variables for changing viewing angle – diameter W and length L.

Figure 6 demonstrates the optics behind the tube. We see the following equation,

$$tan(\Theta) = \frac{\frac{W}{2}}{\frac{L}{2}} = \frac{\frac{R}{2}}{D + \frac{L}{2}} \tag{3}$$

so

$$D = \frac{L}{2}(\frac{R}{W} - 1) \tag{4}$$

determines the distance D at which we can observe our intended resolution R for a length L and width W of the tube.

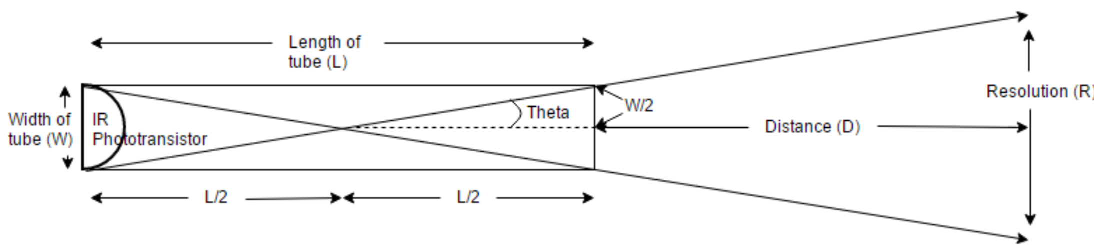

Figure 6: Receiver Cell with tube

As Equation (4) and Figure 7 demonstrate, we can only maintain resolution R at a distance D. We set $W =$ .125" to hug the 3mm = 0.11811" diameter phototransistor and $L =$ 12". This gives us $R =$ 2" resolution at $D =$ 7.5', or 90". Because we're mounting the panel at 8' above the ground, we'll see 2" resolution therefore at $8' - 7.5' = 0.5'$ or 6" above the ground. Pixels overlap past D. We want to prevent pixel overlap between the sensor array and the ground because distances where pixels overlap will cause larger blobs to form, i.e. magnification. We set L to a large value to minimize magnification close to the ground. By time of final demonstration, we met Requirement 1 from Appendix A by demonstrating the dimensions of the tube guide with a ruler.

When a person walks under the array, we will see small area of high intensity activations corresponding to their head and shoulders and a surrounding aura of lower intensity activations corresponding to their magnified feet and ankles. It is possible the intensity of reflection from feet and ankles (6" above the ground) will be indiscernible from ambient noise. We verify this with experimentation.

We then have to think about tube construction. What material should we use? How should we mount the tubes?

When we consider heat transfer, we refer to Kirchhoff's law of thermal radiation [4], which states an object

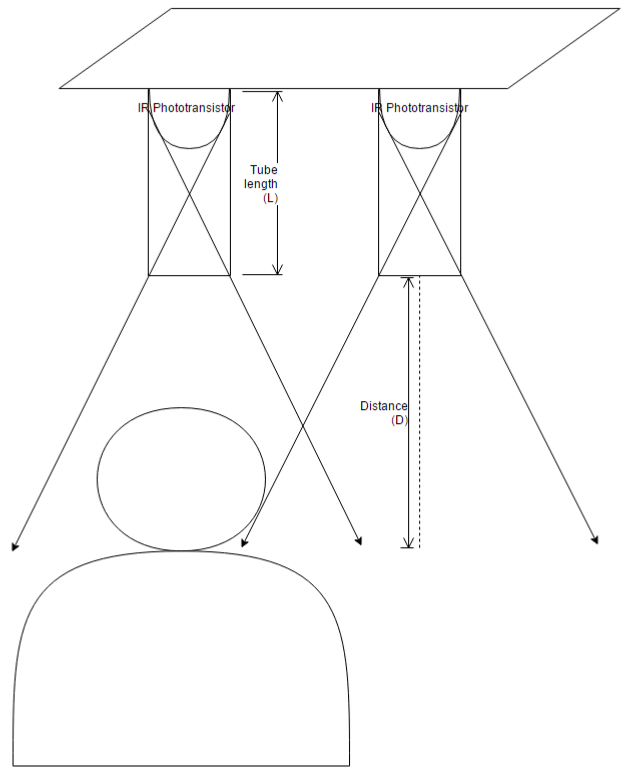

Figure 7: Receiver Cell Resolution

at a given temperature has an absorptivity equal to its emissivity. This only holds when an object is in thermodynamic equilibrium. Regarding material, we want something with a high solar absorptivity, which we will refer to as emissivity assuming Kirchhoff's law of thermal radiation holds. Emissivity, ranging from 0 to 1, is a measure of how well a material absorbs radiation of a particular wavelength. An emissivity value of 1 represents a perfect black body radiator. The solar band of radiation ranges from 380nm to 2.5um. Our infrared LEDs emit at 940nm within the solar band of radiation. We want any of this infrared light which enters the tube at an angle greater than Θ to be absorbed when it hits the tube's inner wall. PVC is a candidate for tubing because of its high emissivity (0.91 to 0.93) and the ease at which we can cut it to a desired length L and choose its diameter W.

For any tubing, we have to limit its curvature. We can do this by framing the tubes with a tube guide structure. Foam board is a good candidate for creating this structure. In fact, if we use a solid foam board to frame the tubes, we can eliminate the PVC tubes completely. They will only have a more reflective surface than the foam itself and likely a lesser emissivity. Foam has a jagged surface, especially if it's been cut, so boring holes straight into a block of foam meant for home insulation should creates integrated tubes which will absorb infrared that hits their walls.

Step 2: For Step 2, we will investigate tubes with a pinhole aperture. Figure 8 demonstrates the advantage these offer. This is equivalent to taking Figure 6, cutting it in half, and blocking all light out of the receiving end, except for the vertex of the height $\frac{L}{2}$ triangle. In the ideal case, when the pinhole is a single point, this would cut our tube length in half.

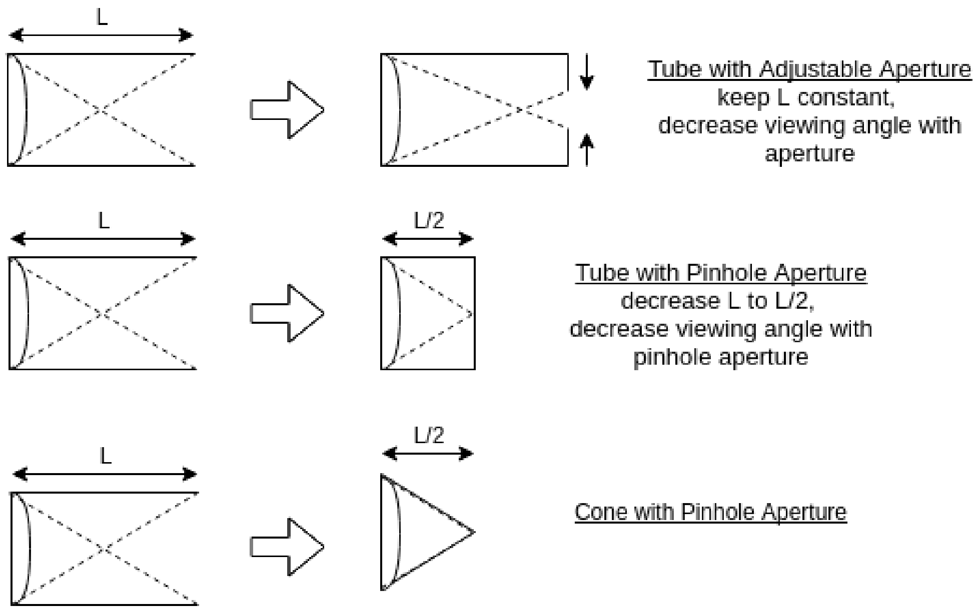

Figure 8: Changing length and aperture size to adjust viewing angle

Step 3: For Step 3, we will investigate lenses on each receiver. The advantage is an even shorter distance from the phototransistor than the tube or tube with pinhole aperture approaches.

2.2.5 ADC Circuit

Step 1: The array has 8 rows and 8 columns. To route $V_{received}$ at a single pixel to the ADC for sampling, we need to select pixel at a particular row and column. To select a row, we use eight 8:1 analog multiplexers. We index each of these muxes with the 3 bit row address. Next we have to choose a particular column of this row. We could use another 8:1 analog multiplexer, but we instead use an 8 channel ADC to save cost. Figure 9 simulates the receiver cell with a current source I1 substituted for the phototransistor and an emitter resistor R2. Next, two levels of analog multiplexing (one on an external mux and one inside the ADC) are loaded by the $G\Omega$ resistance of the ADC's sample/hold unit. Nearly all of the phototransistor current flows through the emitter resistor due to R_E being much smaller than the sample/hold resistance with which it's in parallel. Therefore, R_E determines the Vout.

The CD74HCT4051 mux is chosen because it's an analog mux with 8 channels. This makes for a logical arrangement of 8 muxes feeding into our 8 channel ADC to sample all 64 receiver cells. The same 3 select bits are routed in parallel to 8 of these muxes. Reference [5] shows how, for a single mux, select bits S0, S1, and S2 turn on only 1 of 8 transmission gates, connecting only 1 of 8 inputs to the common output. The node between the phototransistor emitter and the resistor of an individual receiver cell will be connected to one of these inputs for a given select bit combination. Therefore we can connect 8 of our receiver cells to a single 4051 mux for sampling the node voltage of one these receiver cells at a time. The 4051 mux will preserve the voltage across the receiver cell resistor for sampling. Before testing 8 muxes operating in parallel with outputs feeding to the ADC's channel inputs, we test an individual 4051 mux to ensure it functions how we expect. According to the pinout, we ground V_{EE} and the enable bit E', apply 5V and 0V to channels 0-7, apply address select to select channels where we apply 0V and 5V, and check the voltage we see at pin 3

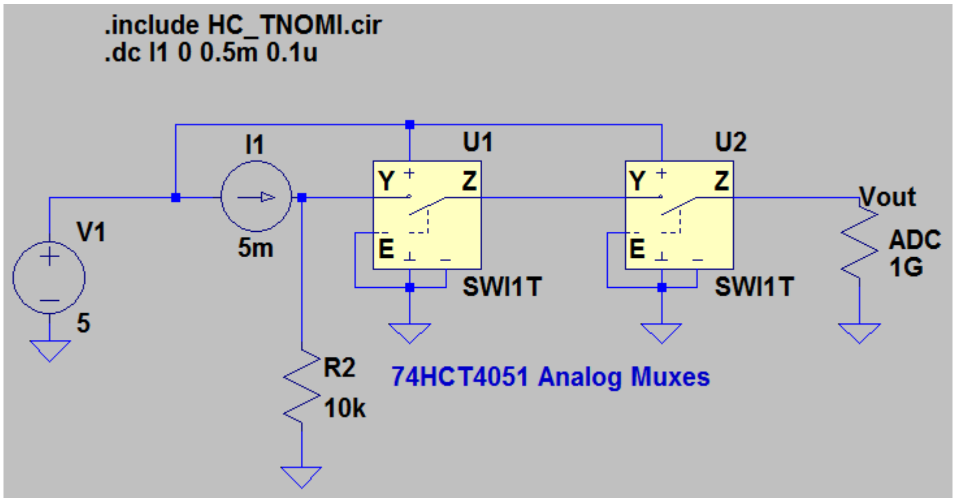

Figure 9: Individual IR Sensor Cell Receiver Circuit

with a DMM. Our test confirms the mux functions as we expect. It is ready for integration.

We have three considerations when sampling. First, we need to index and sample all 64 pixels at least once before a person moves 2 inches on the grid. This allows us to take a moving average. The mean human running speed is 6.7 m/s, approximately 7 m/s, according to Reference [6]. It would take 7.3 milliseconds to cover a 2 inch distance. If we want at least one sample at a pixel before the person moves to another – a form of aliasing – we must index and sample the entire array in 7.3 milliseconds, or $\frac{7.3ms}{64pixels} = 0.114$ milliseconds per pixel. This yields a sampling rate of at least $\frac{1sample}{0.114ms} \geq 8.772$ kHz. Second, the samples we store in microcontroller memory should not prevent us from also normalizing pixel readings as we calibrate S_{splash} above $S_{ambient}$. According to Reference [7], the MCP3008 eight channel ADC can sample up to 200 kHz, satisfying our requirement, at 10 bit resolution. The memory footprint due to sampling, if we take 3 samples per frame as is mentioned later in Step 2 of the Microcontroller design, is $3samples \times \frac{10bits}{sample} \times 64pixels = 1920$ bits $= 240$ bytes. Other memory will be used to keep a running count of objects crossing the panel area and house code necessary to complete sampling and processing. Memory allocated to store samples will not limit the count and processing effort since the Raspberry Pi Zero offers 512 MB of RAM. Third, we want our ADC to have a high impedance for minimal impact on $V_{received}$ when sampling. The chosen ADC has a high impedance sample/hold unit.

10 bit resolution at a 5V swing provides a 4.88 mV step. Knowing this is our finest voltage step for detecting a change in distance, we consider the power difference over 1 inch at the furthest intended distance from our array. A door is 8 feet tall, or 96 inches. Moving from 96 to 95 inches increases $V_{received}$ by 102.1% by Equation (2). If we expect an ambient current near 1 uA a 2.1% change means roughly $\triangle I = .02$ uA. Then the R_E needed to detect a 1 inch change in distance from the top of the door frame to the ground is $\frac{4.8mV}{.02uA} \geq 0.24\text{M}\Omega$. The precise resistor value will be found in the course of building and testing our Step 1 prototype. It will be dependent upon the ambient noise of the room $S_{ambient}$, its standard deviation, and how much $S_{splash,received}$ we choose to reliably detect a change above the noise with our ADC's 4.88 mV step. The chosen R_E value was 22.4 MΩ, which satisfied our constraint and provided greater amplification of current generated from the little light received through our foot-long tube guide. By time of final demonstration, we

demonstrated functionality of Requirements 2, 3, and 4 from Appendix A. For Requirement 2, we explained the choice of resistor, as done here, and we also showed the 5V rail leading to a bank of receiver cells with a digital multimeter to verify 5V operation. For Requirement 3, we ran test code which showed our first level muxing by printing the mux select bits and ran another test code which printed the time it took to sample the entire array (the printed value was consistently less than 7.3 milliseconds). For Requirement 4, we wrote test code to print out all 64 pixel readings from the ADC once they entered microcontroller memory.

Step 2: For Step 2, we will run a raster scan algorithm for clustering pixels on our array (see Figure 10). This should help us gain insight not only into entrances and exits across a panel, but also distinct human features, like the height gap between head and shoulders. Step 2 requires finer precision than Step 1 because we have to ensure we can sense differences in distance beneath the array. Without our foot-long tube guide, which drastically limited receiver array resolution as well as the intensity of light that could be received, this was difficult to guarantee by time of demo. Therefore, our raster scan algorithm for clustering provided limited groupings by time of demo.

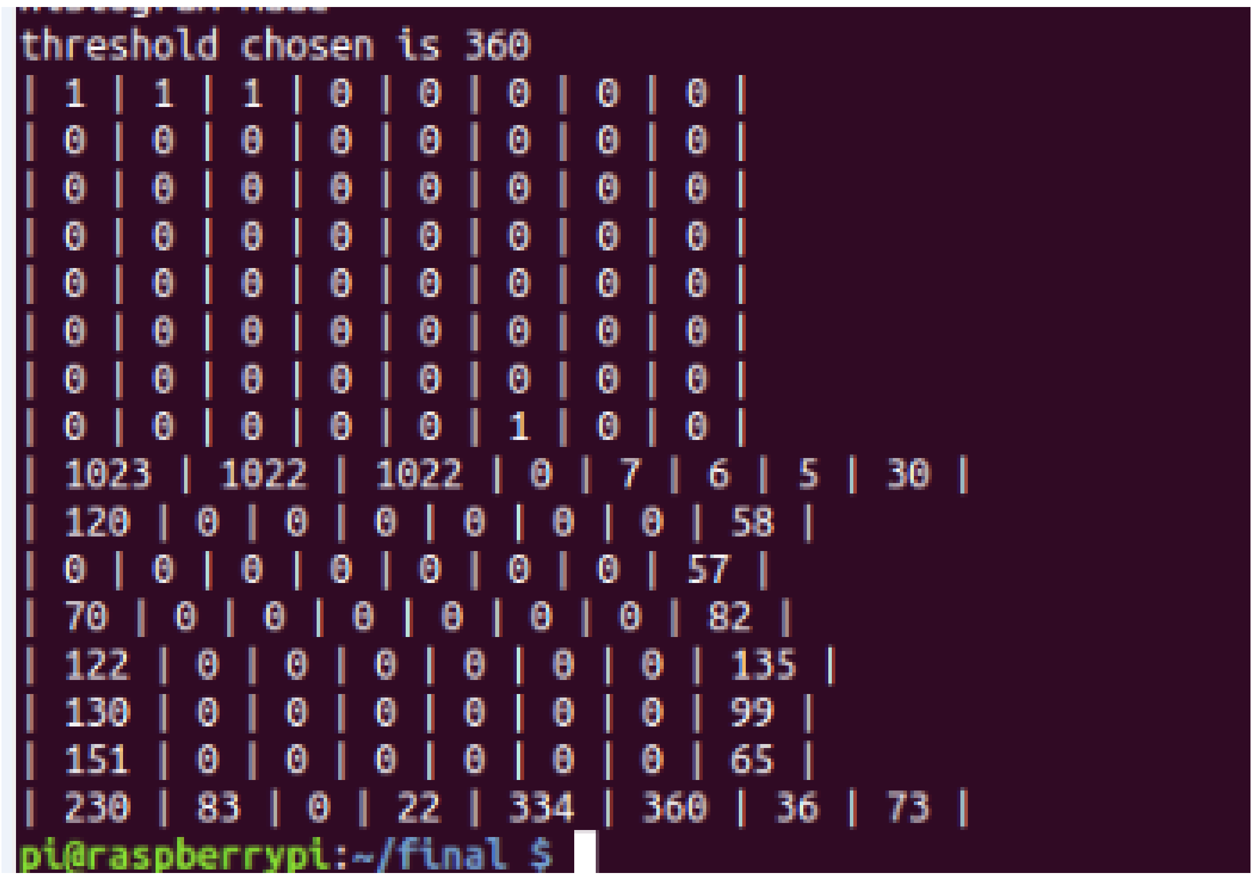

Figure 10: Preliminary Grouping in Early Attempt at Raster Scan

2.2.6 DC Power Supply

Step 1: This power supply will convert the standard 120Vac 60Hz signal to 5A 10Vdc. Its purpose is to provide a DC voltage that can be stepped down with a Buck converter (DC/DC power converter). The maximum power dissipation of all the components in one sensor module is 20W. To accommodate modularity, which would be a Reach Goal, we choose transformer rated for a maximum power of at least 40W, which is double the power we anticipate for a single panel.

The DC supply will be encased in a NEMA 1 enclosure for safe operation. It operates off a three-prong NEMA power plug connected to a wall outlet.

Figure 11 shows our DC power supply configuration. We started by finding a transformer in the ECE Service Shop that was rated for 50W or above to accommodate all of the components used for this project. We use MBR1045 diodes for our full wave rectification since they are available at the ECE Service Shop (MBR745 shown for simulation purposes). Finally, we use a linear regulator (LT1084) to output 10V. Figure 12 shows the output voltage of the DC power supply. We were able to verify Requirement 8 of Appendix A at demo by probing the output voltage with a DMM and showing that it is 10V ± 0.2V.

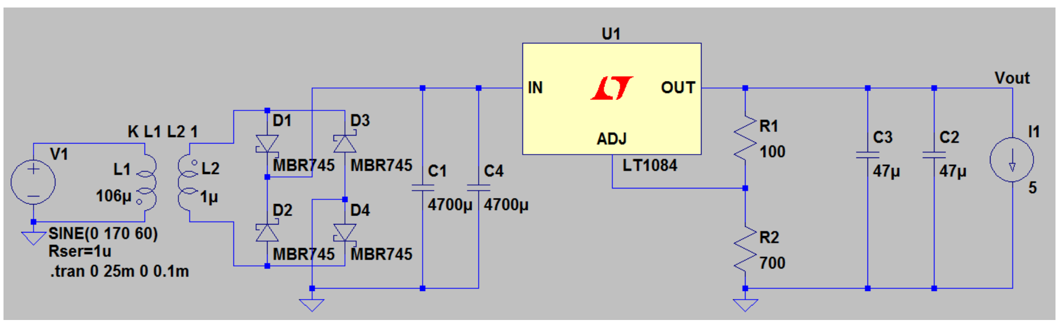

Figure 11: DC Power Supply

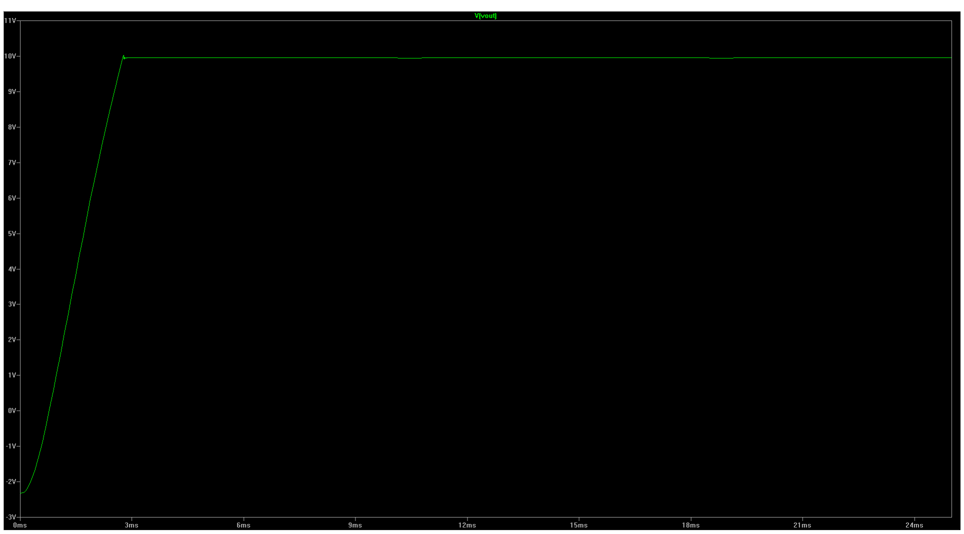

Figure 12: DC Power Supply - Output Voltage

2.2.7 Buck Converters

Step 1: Two buck converters will be used to convert the 10V produced by the DC power supply to 1.35V and 5V. The 1.35V rail will be used to power the IR illuminator submodule while the 5V rail will be used to power the control module, IR receiver submodule, and ADC circuit submodule. The purpose of the buck converters is to efficiently step down the DC power supply voltage to the operating component level voltages.

We decided the use a buck converter (LM2679) from Texas Instruments[8]. Texas instruments provides a detailed design procedure for choosing the components that make up the circuit shown in Figure 13.

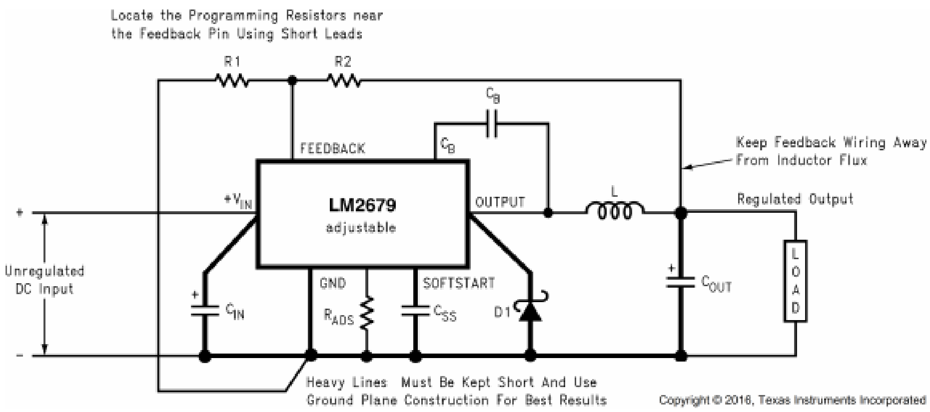

Figure 13: Buck Converter Circuit[8]

The calculations for our 5V buck converter are shown below. The same calculations were used for the 1.35V buck converter.

$$OperatingConditions : Vin, max = 10.2V, Vout = 5V, Iload, max = 5A \tag{5}$$

$$R2 = R1(\frac{Vout}{Vfb} - 1), Vfb = 1.21V(given), R1 = 1k\Omega(recommended) \tag{6}$$

Therefore:

$$R2 = 3.132k\Omega \tag{7}$$

$$E * T = (Vin, max - Vout - Vsat) * \frac{Vout + Vd}{Vin, max - Vsat + Vd} * \frac{1000}{260}(V * \mu s) \tag{8}$$

Vsat is given as 0.12Ω * Iload,max = 0.6V and Vd is the forward voltage drop across the diode (0.55V), therefore:

$$E * T = 9.67(V * \mu s) \tag{9}$$

Then the nomograph they provided is used along with E*T to find L, therefore:

$$L = 10\mu H \tag{10}$$

Then using the parameters we just calculated and their tables provided to us to find Cout, therefore:

$$Cout = 470\mu F \tag{11}$$

Finally Cboost is provided as 0.01μF and Radj is calculated using the following equation.

$$Radj = \frac{37125}{Iload, max} = 7425\Omega \tag{12}$$

We decided to increase the voltage requirement (Requirement 7 of Appendix A) of the buck converter for the IR LEDs from 1.2V to 1.35V to be able to drive the LEDs at a higher intensity. The buck converter circuits were able to provide the desired output voltages when constructed during the final demo but they become unstable when applying different loads. Since we designed the bucks for a max output current of 5A, we believe the converters are operating in discontinuous-conduction-mode (DCM) when drawing small currents. This is confirmed by calculating the critical inductance required for output current by using Equation (13).

$$L = \frac{Vout * (1 - D)}{2 * Iout * fsw} \tag{13}$$

where D is the duty cycle and f_{sw} is the switching frequency of the buck converter. For I_{out} of a single LED (20mA), we would need an inductance of 112.4μH which we do not have from the design procedure.

2.2.8 Microcontroller

Step 1: The microcontroller provides a clock signal to the ADC to set the sample rate. It communicates on an SPI bus with the ADC to read each receiver cell's $V_{received}$ sample and write it to microcontroller memory. Processing on the microcontroller determines whether an object has entered or exited the array area and stores this information as a count variable. It transfers this information to the WiFi chip. The function of the microcontroller is described in Figure 14.

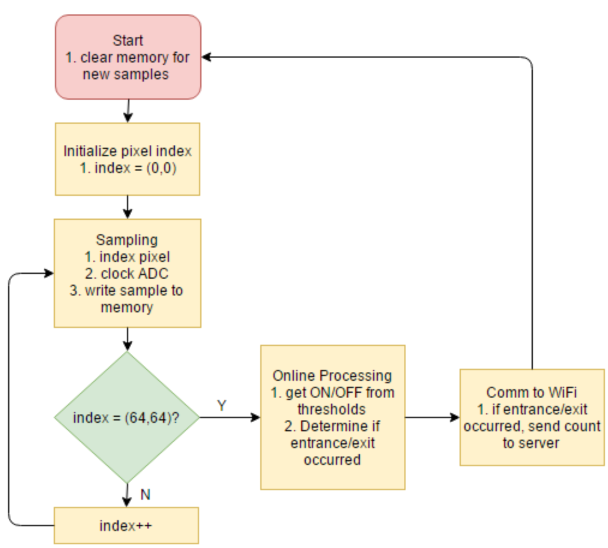

Figure 14: Microcontroller Software Flowchart for Step 1

Step 2: For Step 2, the microcontroller uses the raster scan algorithm to identify clusters of pixels corresponding to a person's head and shoulders. It transmits this cluster data to the Wi-Fi chipset. Figure 15 demonstrates the software procedure necessary to accomplish sampling and clustering on the microcontroller. To provide better data to the clustering algorithm, we run a moving average on pixel activations, sampling each pixel three times. The automatic thresholding and raster scan algorithm for grouping pixels is covered thoroughly in Reference [9].

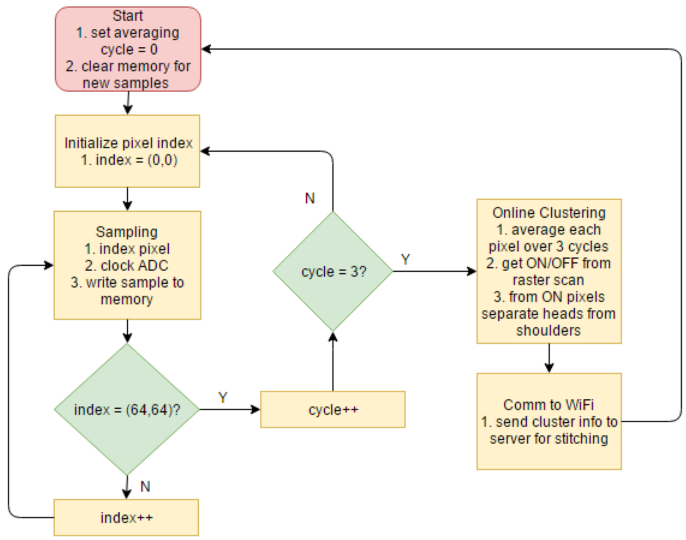

Figure 15: Microcontroller Software Flowchart for Step 2

2.2.9 Wi-Fi Chipset

Step 1: The Wi-Fi chipset will transmit data only if an entrance/exit has occurred. The central server will use that data to display a real time count of people that crossed the panel area. At the final demonstration, we showed functionality of the WiFi chipset according to Requirement 5 of Appendix A by running sampling and raster scan on the Pi and running a separate server terminal on a local machine to receive real time counts of groups in frame. The client and server code was borrowed heavily from Reference [10].

Step 2: The Wi-Fi chipset will transmit the head and shoulder cluster data received from the microcontroller to a central location. Since our device is modular in Step 2, the central server stitches the data received from multiple arrays to act as one larger array.

3 Cost

Since this project was designed to be modular and has the potential to cover a large area, the cost for one device needed to be low. The total cost of our device is show in Table 2. The parts and labor cost is explained in the below sections.

Table 2: Project Total

Labor Cost ($)	Parts Cost ($)	Grand Total ($)
42,500.00	56.59	42,556.59

3.1 Parts

The cost of all the parts used in our device is shown in Appendix C. The cost of miscellaneous supplies to manufacture our device came out to be $126.58 in addition to the parts cost. This miscellaneous cost could be distributed over 3 units, so the actual additional miscellaneous cost per unit is $42.19. If we were to buy additional supplies in bulk we could further decrease the cost. We see the benefit of producing several units to achieve a modular design – the miscellaneous supplies cost can be amortized over every three units.

A search on Amazon for security cameras yields cameras costing as low as $65. As a camera-based system, combined with depth-sensing elements, could be an alternative to our system, these options should be thoroughly explored for a particular use case.

3.2 Labor

Table 3: Labor Costs

Name	Hours Invested	Hourly Rate ($)	Total * 2.5 Engineering Factor ($)
Armando Juresic	250	34.00	21,250.00
William Schellhorn	250	34.00	21,250.00
Total	**500**	**68.00**	**42,500.00**

4 Conclusion

4.1 Accomplishments

By time of demonstration, we completed the requirements in our Requirements and Verifications table with a slight setback in our power module. Our progress demonstrated with continued development and access to a manufacturing facility, we would be able to significantly improve the functionality and the size of our product.

4.2 Tolerance Analysis

A critical function of our system is being able to detect changes in distance beneath the array despite operating in a noisy environment. We will use the same sampling procedure as described in Step 1 for the microcontroller submodule. However, we will use it to sample ambient noise. From our samples, we can determine the standard deviation and mean of the noise power. Noise is our first random variable. We turn on the LEDs selectively and contribute a second random variable. The distributions of the noise power $S_{ambient}$ and splashed, reflected power $S_{splash,reflected}$ add. An ON activation level will be several standard deviations above this joint distribution mean, which will correspond to a certain number of steps above a mean voltage on our ADC. We will use this logic when experimenting with our prototype array.

Every minute our microcontroller will analyze 15 seconds of sampled data. If it appears there is a uniform voltage across the array, the microcontroller will disable the splash for several milliseconds. It will resample ambient noise without the backgroound. This way we can determine changes in ambient noise and figure out how much power to splash from our array.

We believe the next most important part of our system is making sure we don't operate the panel in an environment flooded with infrared (IR) light. Ambient conditions could cause erroneous activations and therefore the data we process could produce false positives. Ambient conditions with high intensity IR could saturate our sensors.

We have also sized R_E with our ADC's minimum voltage step in mind to maximize our ability to see changes in distance to objects below the array.

4.3 Ethics and Safety

Our device directly addresses the first tenet of the IEEE Code of Ethics addressed in Reference [11]: "to accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment." One purpose of crowd monitoring is to perceive threats in a crowded environment and attend to them before they cause harm. Another purpose is to gauge congestion over an area for properly staffing this area. To address security and privacy concerns, the resolution is limited so it would be very difficult to profile any single person with information other than their motion.

One danger posed by our project is shining people with too high an intensity IR light. IR, though it is not detected by rods and cones in the retina, is still focused onto the back of the retina and can cause tissue damage if it is high power. Our chosen LEDs will not transmit a high enough intensity to damage the eye.

Another danger is our panel falling from its mount and hitting a passerby. We limit this risk by demonstrating our panel without mounting the heavy transformer we scavenged from the ECE Service Shop to our PCB. In fact, we create a NEMA enclosed DC power supply which will sit on the ground while we run wires with DC voltage to our panel. This way we lessen the weight on our mount.

When working on our transformer, we follow the course safety policy carefully. This includes never working with main voltage unless a TA is present or unless we have received TA approval. We have checked the transformer's turns ratio with TA oversight and guaranteed the transformer housing is at ground. Nevertheless, we enclose the entire DC power supply in a NEMA type 1 enclosure for added safety.

4.4 Future Work

To improve the sensor array's reception and create a more compact device, we can incorporate lenses into our design. The end goal is a compact, manufactured device. With additional time and resources put toward development, this device could be sold at market with surveillance and security applications in locations such as theme parks, shopping malls, entertainment venues, and transportation hubs. In the future, it could also be used in manufacturing to monitor personnel and parts included in assembly and architectural applications for designing the movement of groups of people throughout a structure.

References

[1] "TPIC6B595 Power Logic 8-Bit Shift Register (Rev. B)," Datasheet, Texas Instruments, 2015. [Online]. Available: http://www.ti.com/lit/ds/symlink/tpic6b595.pdf

[2] "Human Head," Web page, Wikimedia Foundation, accessed October 2016. [Online]. Available: https://en.wikipedia.org/wiki/Human_head

[3] *PT204-6B*, Datasheet, Everlight Electronics Co Ltd., 2016. [Online]. Available: http://www.everlight.com/file/ProductFile/PT204-6B.pdf

[4] "Kirchhoff's Law of Thermal Radiation," Web page, accessed December 2016. [Online]. Available: https://en.wikipedia.org/wiki/Kirchhoff's_law_of_thermal_radiation

[5] "High Speed CMOS 8-Channel Analog Multiplexer/Demultiplexer with TTL inputs," Datasheet, Texas Instruments, 2015. [Online]. Available: http://www.ti.com/lit/gpn/cd74hct4051

[6] "What Is The Average Human Running Speed," Web page, Wikimedia Foundation, accessed October 2016. [Online]. Available: https://www.reference.com/health/average-human-running-speed-7f0ef0953669fa1

[7] "2.7V 4-Channel/8-Channel 10-Bit A/D Converters with SPI Serial Interface," Datasheet, Microchip Technology, Inc., 2016. [Online]. Available: https://cdn-shop.adafruit.com/datasheets/MCP3008.pdf

[8] "LM2679 SIMPLE SWITCHER 5A Step-Down Voltage Regulator With Adjustable Current Limit," Datasheet, Texas Instruments, Inc., 2016. [Online]. Available: http://www.ti.com/lit/ds/symlink/lm2679.pdf

[9] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. John Wiley and Sons, 2006.

[10] "Beej's Guide to Network Programming," Web page, accessed October 2016. [Online]. Available: http://beej.us/guide/bgnet/output/html/multipage/index.html

[11] "IEEE Code of Ethics," Web page, IEEE, accessed October 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html

Appendix A Requirement and Verification Table

Table 4: System Requirements and Verifications

Requirement	Verification	Verification status (Y/N)
1. Submodule: Receiver Array (a) Receiver rows are spaced 2 $\pm$ 0.2 inches apart. Columns are spaced 2 $\pm$ 0.2 inches apart. (b) Receiver tubes are 12 $\pm$ 0.5 inches in length. (c) Receiver tubes are 1/8 $\pm$ 1/16 inches in diameter.	1. Submodule: Receiver Array (a) Measure 2 $\pm$ 0.2 inches between rows and columns with tape measure. (b) Measure 12 $\pm$ 0.5 inches for lengths of tubes. (c) Measure 1/8 $\pm$ 1/16 inch for diameter of drill bit used for creating the tubes.	Y
2. Submodule: Receiver Cell (a) Resistor R_E is chosen so maximum current through phototransistor keeps $V_{received} \leq$ 5V $\pm$ 0.1V. (b) Cell has 5V $\pm$ 0.2V supply.	2. Submodule: Receiver Cell (a) Test current through phototransistor as object approaches array from 1 foot away to end of tubes. Test 2-wire resistance from output node to ground with DMM. Max current times resistance should be less than 5V $\pm$ 0.1V for entire range. (b) Probe voltage at collector of each cell's photoransistor with DMM. It should be 5V $\pm$ 0.2V.	Y
3. Submodule: ADC Circuit (a) By virtue of first level muxing, ADC can read output node at each of 64 pixels. (b) We must sample array data at $\geq$ 8772 Hz.	3. Submodule: ADC Circuit (a) Do this for every pixel with aid of microcontroller testbench. Apply 5V to cell output node, apply corresponding column index to mux select pins, output voltage reading with DMM on proper row mux output should be 5V $\pm$ 0.1 V. (b) In microcontroller code, write timestamp before sampling the 64 pixels and after sampling the pixels. The output time should be $\leq$ 8 ms.	Y

Continued on next page

Table 4 – continued from previous page

Requirement	Verification	Verification status (Y/N)
4. Submodule: Microcontroller (a) Microcontroller reads all 64 pixel samples from ADC into microcontroller memory.	4. Submodule: Microcontroller (a) Start sampling of array according to microcontroller flowchart. Write at least 240 characters to memory. Verify by visual inspection (print statements) memory has been written.	Y
5. Submodule: WiFi Chipset (a) WiFi chip needs sends count to central server.	5. Submodule: WiFi Chipset (a) Connect WiFi chip to central server on Wireless LAN. Send test byte from microcontroller to central server.	Y
6. Submodule: IR Transmitter (a) IR LEDs receive 1.2 - 1.6 V. (b) We can turn on/off each individual LED.	6. Submodule: IR Transmitter (a) Probe voltage across LED terminals with DMM. It must be greater than 1.2V and less than 1.6V. (b) Send serially a pattern to Power Logic Shift Registers to turn on 1 LED at a time. Use smartphone camera to visually verify LED is on.	Y
7. Submodule: Buck Converters (a) Vout = 1.2V ± 0.05V with output current limited to 5A (b) Vout = 5V ± 0.1V with output current limited to 5A	7. Submodule: Buck Converters (a) Place load resistor at output, use a multimeter to measure the output voltage. The voltage should be 1.2V ± 0.05V. Use an ammeter to measure the current drawn through the load. The max current should be 5A. (b) Place load resistor at output, use a multimeter to measure the output voltage. The voltage should be 5V ± 0.1V. Use an ammeter to measure the current drawn through the load. The max current should be 5A.	Y

Continued on next page

Table 4 – continued from previous page

Requirement	Verification	Verification status (Y/N)
8. Submodule: DC Power Supply (a) Vout = 10V ± 0.2V with output current limited to 5A	8. Submodule: DC Power Supply (a) Place load resistor at output, use a multimeter to measure the output voltage. The voltage should be 10V ± 0.2V. Use an ammeter to measure the current drawn through the load. The max current should be 5A.	Y

Appendix B Overall Schematic

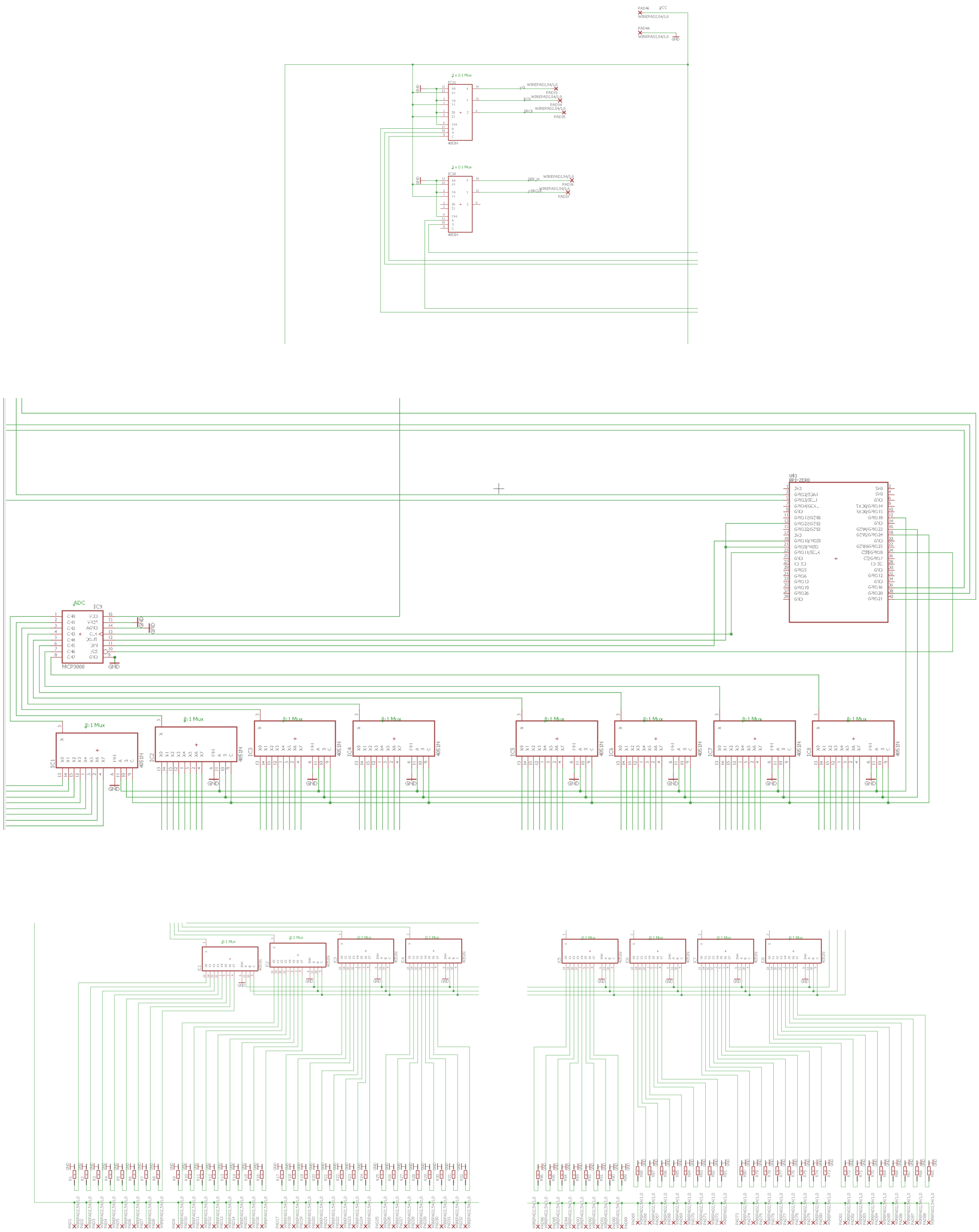

Figure 16: Control Schematic

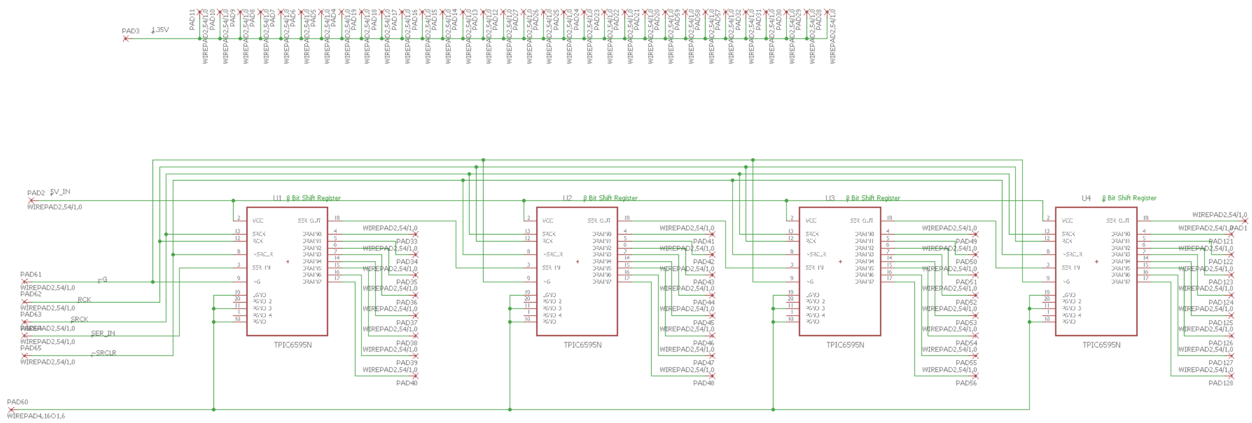

Figure 17: Shift Register Schematic

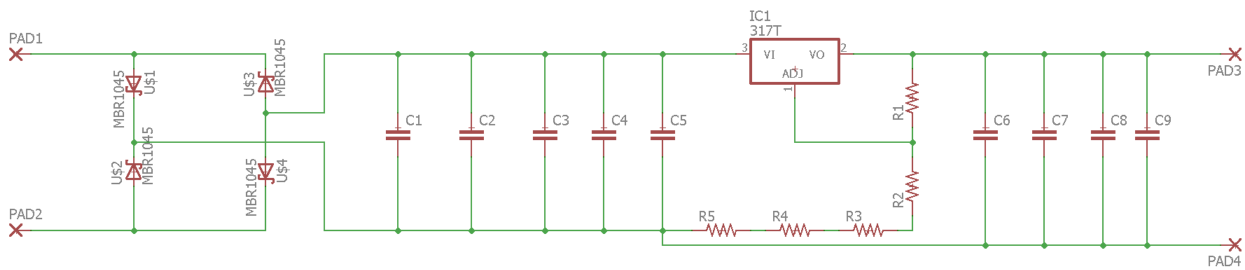

Figure 18: DC Power Supply Schematic

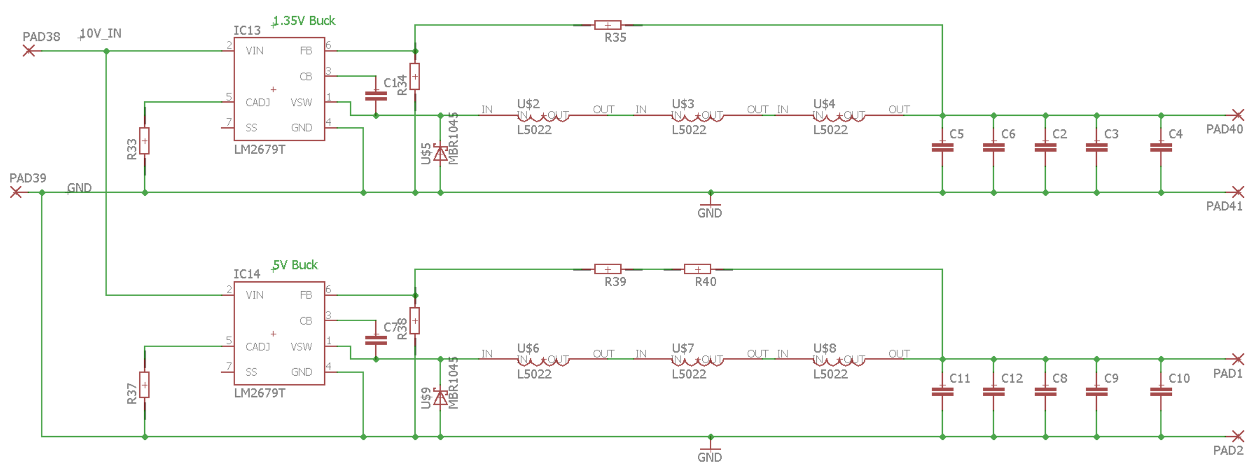

Figure 19: Buck Converters Schematic

Appendix C Parts Cost

Table 5: Parts Cost

Part	Retail Cost ($)	Bulk Purchase Cost ($)	Actual Cost ($)	Status
PT204-6B IR Phototransistor	1 @ 0.34	100 @ 0.1315	13.15	received
Raspberry Pi Zero	1 @ 5.00		5.00	received
LTE-5228A IR LED	1 @ 0.50	100 @ 0.28	free	received
CD74HCT4051E 8:1 Analog Mux	1 @ 0.63	10 @ 0.536	5.36	received
MCP3008 ADC	1 @ 2.19	25 @ 1.82	2.19	received
Resistors			free	received
PL-8188EUS WLAN USB adapter	1 @ 5.29		5.29	received
NEMA power plug			free	received
2x20-pin Strip Dual Male Header	1 @ 0.95		0.95	received
USB to OTG	1 @ 0.95		0.95	received
Wire			free	received
16G microSD			9.88	received
P6458 AC/DC Transformer			free	received
LM2679 Buck Converter	1 @ 5.96	10 @ 5.355	11.92	received
LT1084 Linear Regulator	1 @ 1.90	25 @ 1.75	1.90	received
MBR1045			free	received
Total			**56.59**	

Appendix D Schedule

Table 6: Schedule

Week	William	Armando
09/18/2016	Simulate movement through a grid using python.	Draft AC/DC, DC/DC power conversion and ADC circuits.
09/25/2016	Design IR Sensor Cell, surrounding circuit, and muxing	Design FSM for Control Unit
10/02/2016	Write machine learning algorithm	Build Sensor Cell prototype
10/09/2016	Prototype sensor array on breadboard, first run	Program control module with FSM, interact with WiFi chip
10/16/2016	PCB layout for sensor array and power module	Prototype sensor array on breadboard, second run
10/23/2016	Write code for central server to receive from WiFi Chipset	Assemble power module with NEMA enclosure (DC supply)
10/30/2016	Assemble array with power module	Revise PCB layout if needed
11/06/2016	Testing and Debugging Unforeseen Issues	Testing and Debugging Unforeseen Issues
11/13/2016	Testing and Debugging Unforeseen Issues	Testing and Debugging Unforeseen Issues
11/20/2016	Thanksgiving Break / Testing and Debugging Unforeseen Issues	Thanksgiving Break / Testing and Debugging Unforeseen Issues
11/27/2016	Demo	Demo
12/04/2016	Presentation	Presentation

Appendix E Software

```
from math import sqrt,atan

spacing = 0
dimension = 0

widest_angle = 0.756

start_spacing = 2.0
end_spacing = 2.1

max_intensity = 7.22

start_led_x = 0
start_led_y = 0

sensor_x = 7
sensor_y = 7

start_height = 6
stop_height = 72

def angle_to_relative_intensity(angle):
        return -5.2773*pow(angle,4) + 13.081*pow(angle,3) - 9.0559*pow(angle
           ↪ ,2) + 0.3827*angle + 0.9996

def intensity(x, y, led_x, led_y, height):
        distance_from_sensor = sqrt( pow( x - led_x, 2) + pow( y - led_y, 2 ))
        angle = atan(distance_from_sensor/height)
        if (angle > widest_angle):
                angle = widest_angle
        return ( max_intensity * angle_to_relative_intensity(angle) ) / pow(
           ↪ height,2)

def single_sensor(x, y, height):
        intensity_at_point = 0
        led_x = start_led_x
        led_y = start_led_y
        while led_x < (dimension-1)*spacing:
                while led_y < (dimension-1)*spacing:
                        intensity_at_point += intensity(x, y, led_x, led_y,
                           ↪ height)
                        led_y = led_y + spacing
```

```
            led_x = led_x + spacing
        return intensity_at_point

def sweep_height_single(start, stop, x, y):
        curr_height = start
        interval = 9
        while curr_height < stop:
                count = 0
                success = 1
                curr_intensity = max_intensity
                while (interval*count + curr_height < stop):
                        prev_intensity = curr_intensity
                        curr_intensity = single_sensor(x, y, interval*count + curr_height)
                        if (curr_intensity > prev_intensity):
                                print "prev_int ", prev_intensity, " curr_intensity ", curr_intensity
                                print "spacing ", spacing, "curr_height_is ", curr_height, " count_is", count, " study at ", interval*count + curr_height, dimension
                                success = 0
                        count = count + 1
                if (success == 1):
                        print "Success at ", curr_height
                curr_height = curr_height + 1

if __name__ == '__main__':
        spacing = start_spacing
        dimension = 16 / spacing
        while spacing < end_spacing:
                sweep_height_single(start_height, stop_height, sensor_x, sensor_y)
                spacing = spacing + 0.1
                dimension = 16 / spacing
```

```c
/*
isr4pi.c
Credit to D. Thiebaut, Drogon for WiringPi Library, and Beej at "Beej's Guide
    ↪ to Network Programming" for their contributions to this code...Provided
    ↪ is a sample of the code. Please contact Armando or Will for full body of
    ↪  code...
based on isr.c from the WiringPi library, authored by Gordon Henderson
https://github.com/WiringPi/WiringPi/blob/master/examples/isr.c

Compile as follows:
    gcc -o isr4pi isr4pi.c -lwiringPi

Run as follows:
    sudo ./isr4pi
*/

#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <stdlib.h>
#include <wiringPi.h>
#include <limits.h>
#include <time.h>
#include <pthread.h>
#include <unistd.h>
#include <netdb.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <math.h>

#include <arpa/inet.h>

#define PORT "3490"
#define MAXDATASIZE 100

// Use GPIO Pin..., which is Pin... for wiringPi library

#define TEST_ON_PI
#define PRINT_SAMPLES
```