Contents

[**Abstract** 1](#_Toc127828379)

[**Problem Identification** 1](#_Toc127828380)

[**Sensors Selection** 2](#_Toc127828381)

[**Signal Processing** 5](#_Toc127828382)

[**Physical Considerations** 8](#_Toc127828383)

[**Logic Implementation/ADC** 8](#_Toc127828384)

[**Sensor Fusion Algorithms** 12](#_Toc127828385)

[**Internet of Things** 13](#_Toc127828386)

[**Results** 13](#_Toc127828387)

[**Resources** 14](#_Toc127828388)

[**Appendix** 14](#_Toc127828389)

[Section A - Data 14](#_Toc127828390)

[Section B - Physical Considerations 15](#_Toc127828391)

[Section C - Code 18](#_Toc127828392)

**Abstract**

The goal of this project was to build a project that utilizes three sensors and solves some sort of problem in the world around. This project is a lamp that dims and brightens depending on the amount of light present and focuses on making life more convenient for the user. The current result is a very archaic version which uses an LED as the lamp light source, an LDR to measure light, an ultrasonic sensor to measure the distance from the light to the nearest object, and a potentiometer to control the brightness at which the light stabilizes. The LDR and ultrasonic sensor worked as expected while the potentiometer had to go through some changes in the digital signal processing to account for large amounts of noise.

**Problem Identification**

Engineering is not only something to invent new things or analyze existing systems but also something that is used to make life more convenient. Oftentimes lamps and lights are considered harmful because they are too bright and can cause general upset. By having a lamp that caters to those needs, it makes life less upsetting.

Since this is for the general public, it needs to be easy to use and require minimal user input. This also means that precise measurements are not necessary and provide more information than needed. As long as there is visible change in the output light, the process works and can be implemented on a larger structure. Another aspect to think about is to keep cost minimal so it is easy to produce. The system should be as simple as possible to reduce costs and only require simple user inputs.

**Sensors Selection**

The first step to this project is to measure the light in the environment. To do so, a LDR will be used.

Diagram

Description automatically generated

**Figure:** LDR inside parts

An LDR is a photoresistor so that when light falls on it, the resistance decreases. Since the sensor is made of a semiconductor with high resistance, most of the electrons are locked inside the crystal lattice with only a few electrons available for current to travel through.

Chart, scatter chart

Description automatically generated

**Figure:** Photoelectric Effect

 As light shines on the semiconductor, the electrons inside the crystal lattice get excited from light photons and become free allowing for more current to flow through (photoelectric effect as shown by the figure above). This decreases the resistance of the material. The longer light shines on the material, the more electrons will become free and the lower the resistance becomes. This can also serve as a timer, allowing for the lamp to dim over time.

Other possible sensors:

|  |  |
| --- | --- |
| Sensor | Explanation Against |
| Photodiode | A diode has a constant voltage drop while the current changes. For this setup (based off of code and voltage measurements taken by the arduino), it would be harder to implement. If the setup was more hardware based, the photodiode would be a better option as it would control current flow. |
| Phototransistor | The constant voltage drop and the varying current flow poses a similar problem like the photodiode. While other voltage drops can be measured (like the drop across the collector resistor), this makes the hardware more complicated than a simple LDR. |
| Others | Other sensors are used to measure radiation specifically while this project focuses on ambient visible light. |

The end reasoning for choosing the LDR is that it is a simple sensor that is easy to implement so it does not pose any additional complications to the setup.

Diagram

Description automatically generated

**Figure:** Potentiometer inside

In addition to the LDRs, a potentiometer is used as a dial. By turning the potentiometer, the resistance changes as the wiper moves across the resistive wire and the voltage drop across the potentiometer changes proportionally. This voltage can be taken and used in calculations in the microprocessor code to change the brightness the circuit will stabilize at. This allows the user to change the brightness manually to a degree.

Other possible sensors:

|  |  |
| --- | --- |
| Sensor | Explanation Against |
| FSR | The sensor gives a simple HIGH or LOW reading which is what the potentiometer ended up being for. However, the reasoning for the potentiometer was so the user can give more specific inputs as to how bright they want their light to be. The FSR is a better sensor to measure a binary input. |

The reasoning for choosing a potentiometer is that it is a simple sensor that can give specific measurements. While it was not possible in this case, if a newer potentiometer was used, the noise would be significantly less allowing for better precision.

Diagram

Description automatically generated

**Figure:** Ultrasonic sensor workings

The last sensor chosen was an ultrasonic sensor. This sensor works by using a “speaker” and a “microphone”. The “speaker” or transmitter sends out a signal of a specific frequency (in this case 40Hz) and the “microphone” waits for the signal to bounce back. The amount of time it takes for the signal to be read can be used to calculate distance. The time it takes to come back is the time it takes for the signal to travel the distance twice (to and from). By multiplying this by the speed of sound, the distance traveled is calculated.

Other possible sensors:

|  |  |
| --- | --- |
| Sensor | Explanation Against |
| Others | Most other distance sensors (such as LVDT and the Bourdon tube) use motion to measure distance or in other words position/displacement. This system requires the measurement of distance without displacement. Optical based sensors could also be used, however those add additional complications since the system only requires a distance measurement from one axis and just generally. |

The reasoning for choosing the ultrasonic sensor is to enhance the system without adding additional complications. This is further explained below.

The initial thought process for the system was to use two LDRs and a potentiometer. The potentiometer is for user input on their preferred brightness level. Each LDR would be facing a different axis (one vertical and one horizontal) so that it would look like this:

Diagram

Description automatically generated

**Figure:** Circuit Design 1 (potentiometer excluded for simplicity)

However, this design is flawed because the LDR can only measure from the front facing direction (the direction the arrows are pointing). This means the LED would be hitting the back of the horizontal LDR so that one would be rendered useless.

To combat this the horizontal LDR could be flipped around as shown below:

Diagram

Description automatically generated with medium confidence

**Figure:** Circuit Design 2 (potentiometer excluded for simplicity)

While the LDR can now take the correct measurements, it is also now away from the rest of the system which would make the lamp more inconvenient for the user. So in the end, one LDR is used to measure the light existing in the space. Since the lamp is intended for public usage, specific measurements of light are not necessary and a general understanding of the amount of light in the area would suffice.

Diagram, schematic

Description automatically generated

**Figure:** Circuit Design 3 (potentiometer excluded for simplicity)

This is where the ultrasonic sensor comes into play. If the user wants to light up a certain object and holds it up to light, this would cast a shadow onto the LDR and mess up the readings. By measuring how close the closest object is, the system can determine, simply, if the user is trying to add more light. If the object is within a certain distance, then the object would be considered “targeted” and the amount of light emitted by the LED would be increased regardless of the LDR. The ultrasonic sensor will also be able to pick up clear objects (which optical sensors would not be able to pick up) that the user wants “targeted”.

**Signal Processing**

To process the LDR, a wheatstone bridge was used (one is shown below for reference). At room lighting, the LDR (R4) had a resistance of about 1100 Ω. By making all the other resistors 1100 Ω, then at room lighting (defined as 22 as shown in the data in the Appendix or in the graph further below) would create a balanced bridge where the difference between C and D would be 0 (zero) or null. Using a 5 V input, Vout had a range from [-2.5, 2.5] and Vd had a range of about [0, 5]. Since the input voltage ranges (nodes C and D) fall in the range for the Arduino, nodes C and D are directly taken into the code.

Diagram, schematic

Description automatically generated

**Figure:** Wheatstone bridge for reference.

The data points collected are put into a graph below:

Chart, scatter chart

Description automatically generated

**Figure:** Graph for voltage vs light level after wheatstone bridge.

As shown on the graph, 22 (light level) serves as the basis for all the data. This is because this measurement is chosen to be null (when choosing the earlier resistors). While the points on the graph are in a wave formation, they still roughly follow a linear trend so no linearizing resistor was put into place for the actual system. Also only general measurements are required per system requirements. If linearizing was required, a resistor could be used by implementing the following formula:

R = R2(R1+R3)-2R2R3R1+R3-2R2 where RP = RRSR+RS and RP1-RP2=RP2-RP3

The second equation states that a resistor is placed in parallel to the resistive sensor (parallel to R4). The third equation expands that statement so that the overall resistance of the sensor and added resistor are parallel as defined where RP1 and RP3 are two endpoints and RP2 is a middle point. If the two end points and the null measurement are chosen, R would be calculated as 1104.86 Ω. (This calculation uses approximate resistances pulled from the datasheet where R1 is .5 MΩ, R3 is 0 Ω, and R2 is 1100 Ω ).

The potentiometer is placed into a half-bridge which only has nodes A, D, and B from the wheatstone bridge, with the potentiometer acting as both R3 and R4. So when the wiper is centered, the resistance is evenly split from pin 1 and 2 and pin 2 and 3, so the voltage measured at pin 2 (node D) would be half of the input voltage (in this case 2.5). Through calculations, the voltage range across the potentiometer(R4)  is [0, 5]. Since this is an additional simple measurement to roughly shift the brightness, there is no need for any biasing or calibration.

In implementation, the potentiometer was not very accurate. As seen in the plot below, the readings kept returning values around 2.5 despite the dial being turned at either maximum. From time 0 to shortly before 100, the dial was turned all the way right so the resistance from pin 2 and 3 would be about 0 as with the voltage reading. From about time 100 and onwards, the dial was turned all the way to the left so the resistance between pin 2 and 3 would be about 1K and all of the voltage drop would be across it (5V). However, in both of these cases, the readings would occasionally peak at the correct value before returning to the incorrect 2.5V.

Chart

Description automatically generated

**Figure:** Arduino voltage readings for 1K potentiometer

To combat this, a Kalman filter could be used, but that would be both overkill and would not lead to the wanted results. This is because the peaks would be read as “noise” since the Kalman filter favors the more stable value so the results would stay at 2.5 at all times.

Instead a different approach was taken. If the readings kept returning to values around 2.5 then those values could be taken out completely. This was done through the code and will be further explained in logic implementation. Essentially the previous value would be stored and if the actual readings went back to 2.5, then it would use the previous value. If it changed to a value not around 2.5, then the used value would change. This is shown in the figure below. (During the video portion of the results the potentiometer got worse so the noise increased from 2 to 4.8, so only the maximum parts of the potentiometer were used (when voltage was either 0 or 5)).

Chart

Description automatically generated

**Figure:** Readings after code was implemented. Red is the readings, green is the modified readings.

Diagram

Description automatically generated

**Figure:** Ultrasonic signal from datasheet

Resources could not be found on the exact signal processing used for ultrasonic sensors. What was found were that transducers are used to emit and receive a specific frequency. These transducers convert electrical pulses to ultrasonic waves and vice versa. When a pulse is sent to the emitter, it triggers a sonic burst. The receiving end waits for the signal to come back and emits a pulse. The output gives the time it took for the signal to go and come back in μs. From this information it can be safely assumed the sensor works purely in highs and lows and includes a clock. There could be some calibration measure to make sure highs stay high and lows stay low.

**Physical Considerations**

The Bill of Materials and Sensors Checklist can be found in Section B of the Appendix. Since this is very simple and a prototype of a product, most parts of the Sensors Checklist do not apply to this project. An additional point is that the system is meant to be kept in a room inside shelter so extreme situations would rarely be encountered and could only be encountered in devastating conditions.

**Logic Implementation/ADC**

The microcontroller chosen was an Arduino because of its simplicity and availability. The initial process was to be code heavy and with Arduino’s IDE already in place, this makes the microcontroller easy to use. This choice of microcontroller also led to some limitations. The initial thought was to use an actual lightbulb. However, since a lightbulb requires an AC source and Arduino only outputs DC voltage this would not be plausible without additional parts.

Diagram, schematic

Description automatically generated

**Figure:** Triac dimmer switch

The use of a microcontroller also simplifies the hardware. Originally, to change the brightness of the lightbulb, a triac lamp dimmer switch was to be used. [1] This circuit works by essentially taking an AC signal and applying a duty cycle to it. There is a variable resistor to change the cutoff voltage. As the signal pulses, capacitors charge and discharge according to the cutoff voltage which effectively decreases the energy given to the light source. However, this process also makes the light flicker since the voltage changes drastically. This requires additional measurements and overcomplicates the process. Instead by using the microcontroller, the voltage given to the lightsource can be changed through the code according to the calculations from the LDR, potentiometer, and ultrasonic sensor. While increasing the complexity of the code, the small increase drastically decreases the complexity in hardware.

The basic flow chart for the code can be found below. The actual code can be found in the appendix.

A picture containing text

Description automatically generated

**Figure:** Code flow chart.

Chart, diagram

Description automatically generated

**Figure:** Signal flow for sensor reading and processing.

The transfer function stated above for the LDR was derived by taking data points (found in the Appendix) of different levels of light and dark while the LDR was in the wheatstone bridge. These data points were taken by measuring the distance between the shadow/light source and the sensor. The lower the number, the closer the shadow was. There can be some error due to the light source also casting shadows. The graph is shown below. This calculated function is then inverted (solved for x) so the equation calculates for light level based on voltage. This new equation is the one used in the actual Arduino code in the Appendix.

Text, letter

Description automatically generated

**Figure:** Snippet of code used to calibrate potentiometer

The shift voltage goes through calibration as shown in the flow chart. A snippet of the code is provided above. If the readings are in between 2 and 3, then those readings are determined as noise so a stored prior reading is used instead. If the reading is not between 2 and 3, then the reading is not considered noise and is used while also stored in another value in case noise comes up again.

Text

Description automatically generated

**Figure:** Distance threshold in code

A threshold was placed on distance as well. If the value calculated from the ultrasonic sensor was less than 6 cm, it was considered as “targeted” so the light would brighten faster than any other changes could happen. As seen below, under normal circumstances, the duty cycle is only increased or decreased by 1. This is so the brightening and dimming of light is less of an inconvenience to the user and their pupils can adjust accordingly.

Graphical user interface, text

Description automatically generated

**Figure:** Changes to duty cycle to brighten or dim LED

During implementation, the LED would consistently restart, so to combat this if the duty cycle had reached its ultimate high (255) or low (0), it would be forced to stay there. This solved the circulation problem.

**Sensor Fusion Algorithms**

A picture containing electronics

Description automatically generated

**Figure:** General system set up

The overall system is set up like a hub. Each sensor’s output is individually put into the Arduino and then processed to change the brightness of the LED.

There are no specific sensor fusion algorithms used. Each sensor is individually used to alter the brightness of the LED according to the user’s interests as specified above. The potentiometer shifts the wanted brightness level (altering the brightness) and the ultrasonic sensor increases the brightness depending on how far the nearest object is.

If the distance of the light were being measured, then by using the light level calculated from the LDR distance can be calculated (since the light level is was initially made by changing how far a constant light source was). This can be combined with the distance measured by the ultrasonic sensor using a complementary filter so the LDR distance would be used to straighten out noise from the ultrasonic sensor.

However, this system does not require the usage of complicated fusion algorithms since precise measurements of light or distance are not required but just enough to please the common user.

**Results**

A picture containing text

Description automatically generated

**Figure:** Final setup

The system worked as expected with a hiccup with the potentiometer. This is due to hardware issues and was solved through some digital signal processing for the most part. The changes in the brightness of the LED was successfully done with some simple thresholds placed on the values taken from the three sensors. The LED was able to react to the changes as wanted and overall the results were favorable indicating a successful simple system.

**Resources**

[1] home.howstuffworks.com/dimmer-switch5.htm

[2] app.code2flow.com

[3] [www.smartdraw.com/flowchart/flowchart-maker.htm](http://www.smartdraw.com/flowchart/flowchart-maker.htm).

[4] <https://www.enago.com/academy/abstract-versus-introduction-difference/>

[5] <https://www.watelectronics.com/light-dependent-resistor-ldr-with-applications/>

[6]<https://www.electronicsforu.com/technology-trends/learn-electronics/photodiode-working-applications#:~:text=The%20photodiode%20is%20a%20special,to%20work%20in%20reverse%20bias>.

[7] [https://instrumentationtools.com/phototransistor-working-principle/#](https://instrumentationtools.com/phototransistor-working-principle/)

[8] <https://www.maxbotix.com/articles/how-ultrasonic-sensors-work.htm>

[9] <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>

[10] <https://www.kth.se/social/files/54ef17dbf27654753f437c56/GL5537.pdf>

[11]<https://www.deviceplus.com/arduino/the-basics-of-arduino-adjusting-led-brightness/#:~:text=You%20can%20easily%20switch%20an,use%20the%20%E2%80%9CPWM%E2%80%9D%20output>.

[12] <https://www.digikey.com/en/articles/understanding-ultrasonic-sensors>

[13] <https://docs.arduino.cc/static/23df5232176092b0b45986d954d95fc0/A000066-datasheet.pdf>

[14]<https://create.arduino.cc/projecthub/user16726/control-your-lights-with-arduino-and-a-relay-3dcfc0>

**Appendix**

Section A - Data

**A.1** *Data points for voltage vs light level after the sensor was put into a wheatstone bridge.*

|  |  |
| --- | --- |
| Light Level | Voltage |
| 0 | 2.4 |
| 4 | 1.26 |
| 8 | 0.55 |
| 12 | 0.29 |
| 15 | 0.26 |
| 22 | 0 |
| 29 | -0.47 |
| 32 | -0.58 |
| 36 | -0.8 |
| 40 | -1.6 |
| 44 | -2.3 |

**A.2** *Data points for potentiometer across pin 2 and 3 when the potentiometer was configured as a half-bridge.*

|  |  |
| --- | --- |
| Angle of Dial | Voltage across pin 2 and 3 (2nd resistor) Vout |
| 0° | 4.996 |
| 20° | 4.883 |
| 40° | 4.512 |
| 60° | 4.166 |
| 80° | 3.712 |
| 100° | 3.191 |
| 120° | 2.803 |
| 140° | 2.531 |
| 160° | 2.119 |
| 180° | 1.775 |
| 200° | 1.424 |
| 220° | .995 |
| 240° | .493 |
| 260° | .175 |
| 280° | .00377 |
| 290° | .002765 |

Section B - Physical Considerations

**B.1** *Bill of materials*

|  |  |  |
| --- | --- | --- |
| Item | Price | Link |
| Arduino Uno | $28.50 | Arduino UNO REV3 [A000066] <https://a.co/d/gvsXQ9J> |
| LDR | $5.99/20 pcs | Chanzon 20pcs LDR Resistor 5516 GL5516 5mm 0.5 Mohm Photoresistor Light-Dependent Photoconductor 20pcs Photo Light Sensitive <https://a.co/d/4yvleA9> |
| 1.1k Resistor | $0.10/1 pc | <https://www.digikey.com/en/products/detail/stackpole-electronics-inc/CF14JT1K00/1741314> |
| LM324 | $0.49/1 pc | <https://www.jameco.com/z/LM324N-TI-IC-LM324N-Low-Power-Quad-Operational-Amplifier-32V-DIP-14_23683.html> |
| 1k Potentiometer | $1.47/1 pc | <https://www.digikey.com/en/products/detail/bourns-inc./PDB181-E415K-102B/4699101?utm_adgroup=General&utm_source=google&utm_medium=cpc&utm_campaign=PMax:%20Smart%20Shopping_Product_Zombie%20SKUS&utm_term=&utm_content=General&gclid=CjwKCAjw8JKbBhBYEiwAs3sxN_vkAs4OhQ5m7n5m474GGIOpqM5CdYrljKV-5pyfv2zIASpYTsEdpRoC7ewQAvD_BwE> |
| Ultrasonic Sensor (HC-SR04) | $7.99/3 pc | Excelity 3pcs Ultrasonic Module HC-SR04 Distance Sensor with 3pcs Mounting Bracket <https://a.co/d/4VgkyT0> |
| 1k Resistor | $0.10/1 pc | <https://www.digikey.com/en/products/detail/yageo/CFR-25JB-52-1K/96> |
| White LED | $0.26/1 pc | <https://www.digikey.com/en/products/detail/creeled,-inc./C512A-WNN-CZ0B0152/6138557?utm_adgroup=Optoelectronics&utm_source=google&utm_medium=cpc&utm_campaign=Shopping_Supplier_Cree%20LED_0090_Co-op&utm_term=&utm_content=Optoelectronics&gclid=Cj0KCQiA7bucBhCeARIsAIOwr--LSSPKkymQKaPMYhZ8sem6-sIuWAz0hXewgLx9DHfJl_MsxSWjxXcaApLmEALw_wcB> |
| Total Cost | $34.08 |  |

**B.2:** *Sensors Checklist*

Sensor (and the environment)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Area | Check | Pass/Fail | Environment Range | Sensor Range | Comments |
| Environment\* | Temperature Range | Pass | Room Temperature | Can work | Ultrasonic sensor changes due to change in speed of sound at different temperatures |
|  | Max shock and vibration | Fail | Any if not held |  | Rough construction, requires better building materials to stay together |
|  | Humidity | Pass | Room | Can work | Ultrasonic sensor changes due to change in speed of sound at different humidity levels |
|  | Pressure | Pass | Room | Can work |  |
|  | Acoustic Level | Pass |  |  | As long as sounds do not reach ultrasonic level |
|  | Corrosive Gases | Fail |  |  | All sensors need to be exposed |
|  | Magnetic and RF Fields | Pass |  |  | No magnetic fields measured |
|  | Nuclear Radiation | Fail |  |  | All sensors need to be exposed |
|  | Salt Spray | Pass |  |  | Ultrasonic sensor changes due to change in speed of sound in different materials |
|  | Transient Temperatures | Pass |  |  | Should be kept inside a room |
|  | Strain in the Mounting Surface | Pass |  |  | Components are light |

\*Overall accuracy is MOST affected by sensors characteristics such as environmental effects and dynamic characteristics.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Area | Check | Pass/Fail | Environment Range | Cable Range | Comments |
| Sensor Cable\* | Temperature Range | Pass | Room | Room | Should be kept in a room |
|  | Humidity Conditions | Pass | Room | Room | Should be kept in a room |
|  | Noise levels | Pass |  |  | Sounds do not affect readings |
|  | Size and weight | Pass |  |  | All parts are lightweight |
|  | Flexibility |  |  |  |  |
|  | Is a sealed connection req? | Fail |  |  | All sensors and wires are exposed |

\*often the weakest link in a measurement system chain

Section C - Code

**C.1** *Arduino code. The transfer function was shifted from 1.7 to 2.4 to better fit the expected [-2.5, 2.5] range.*

A picture containing text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated