Using Hand Movements: The Ideal Method to Turn on Electricity

Anna Tsatsos
(Dated: April 30th, 2021)

To minimize the work needed to switch on and off certain household lights, my project proposal is to create a switch that bridges the lights with the outlet, allowing the homeowner to keep them plugged in. Using microcontrollers, sensors, and relays, I apply the method of motion activation to toggle the switch on and off. For this project, I implement a staged approach strategy: I start with applying the switch to small voltages, thereby serving as a rough draft and allowing me to refine my switch. I will then graduate to voltages found in household products, achieving the main objective of this project.

I. INTRODUCTION

There are certain lighting products where in order to switch on the light bulbs, one must manually plug it into the outlet. This method proves to be tedious and to give the homeowner unnecessary work every time they want to use the product. To minimize the work needed, my project proposal is to create a switch that bridges the lights with the outlet, allowing the homeowner to keep the lights plugged in. Additionally, I used microcontrollers, sensors, and relays to provide a convenient method of activating the switch; by moving the hand near the motion sensor, one can turn on or turn off the lights as they need. This serves useful for the rising trend of string lights used around the house for ambient lighting, as well as lights used for the purposes of seasonal decoration.

II. CIRCUIT DESIGN

Figure 1 shows a detailed flowchart of the circuit. The circuit starts with the motion sensor on the left side, which emits and receives ultrasonic waves in order to measure the distance of objects via time delay. The sensor sends this information to the Arduino, which converts the time delay in microseconds to a distance in centimeters by using the speed of sound and dividing by 2, since the wave had to go there and back. Then, if the distance is under a certain threshold, this is assumed to be due to a hand movement in front of the sensor. The Arduino toggles the relay, and then waits a second before taking in more information to prevent any residual hand swiping from messing with the switch.

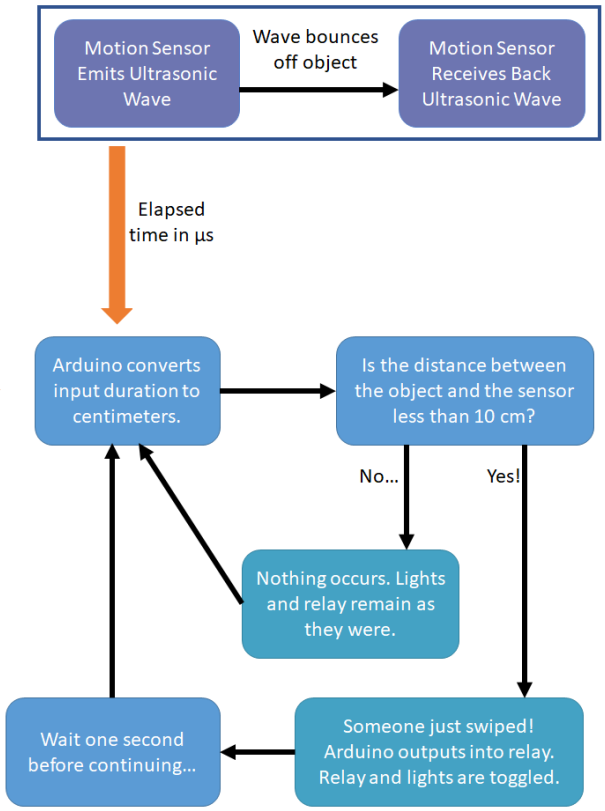

FIG. 1. The circuit design presented as a flowchart, with boxes colored to coordinate with circuit parts: motion sensor (dark blue), Arduino microcontroller (light blue), and SPDT relay (teal).

III. CIRCUIT SCHEMATIC AND PARTS

All of the tables and figures for this section are on the next page. Figure 2 shows the planned schematic for turning on a set of string lights at 120V. Table 1 lists the various parts that will be needed for the process of creating this circuit, including creating it for a 5V LED and then replicating it for 120V products. Figure 3 shows a more technical version of the schematic, making clearer the pins within each part and the wiring of the circuit. Several of the parts are further elaborated in this section.

Motion Sensor: The motion sensor I will use is the HC-SR04 module included in the Arduino kit.[3] The module contains 4 pins: two for the power supply (+5V and Ground), a trigger pin, and an echo pin. The trigger pin first sends out a 10 microsecond pulse, which signals the sensor to emit 8 ultrasonic pulses. Because ultrasonic waves are directional, the only way for a motion sensor to receive its wave back is for the wave to have been bounced off of a certain object. The echo

TABLE I. List of Parts Needed and Order Status

Part Needed	Intent	Availability
Motion Sensor Module	Senses close and far distances	Included in Arduino kit
5V SPDT Relay	Switches the lights on and off based on input	Included in Arduino kit
SPDT Relay for 120V wires	Switches the lights on and off based on input	Delivered via Amazon
Arduino Uno R3	Takes sound input and makes the necessary implementations	Included in Arduino kit
Wires	Connects circuit parts together	Included in Arduino kit
Resistors	Alleviates current and voltage throughout circuit parts	Included in Arduino kit
Breadboard	Serves as platform for circuitry	Included in Arduino kit
Wall Outlet	Serves as the power source for the product	Included in household
String Lights	Serves as the household product	Included in household

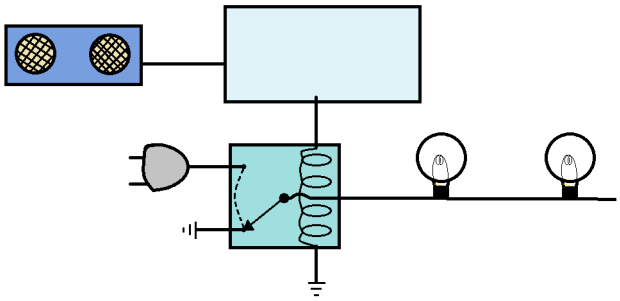

FIG. 2. The final circuit schematic, complete with a motion sensor (blue) with the ultrasonic wave transmitter and receiver, Arduino microcontroller (light blue), SPDT relay (teal), wall outlet (yellow), and string lights, which are currently turned off.

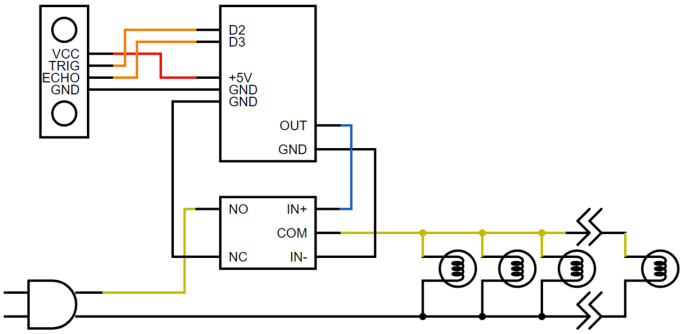

FIG. 3. A more detailed version of the circuit schematic, showing the pins used in each circuit component, and the 24 string lights in parallel each taking in the same amount of voltage.

pin measures how long it take for the pulses to come back. It outputs high after the pulses have been sent out, and is set to low once the pulses have returned. By measuring the time delay from emission to reception, the ultrasonic sensor can sense how far away certain objects are from it. Should there be no object in front of the sensor or an object that is too far away, the echo pin will output a time delay of 38ms by default. For my project, this default output will not serve as an issue, as I plan to toggle the switch at distances corresponding to time delays that are in the order of microseconds.

Arduino Microcontroller: The microcontroller I will be using is the Arduino One R3 that comes with the Elegoo complete starter kit. This Arduino will be the brains of the operation, taking the input sound and outputting the appropriate response to the relay. The Arduino is complete with 14 digital pins, 4 analog pins, and its own power supply and ground pins that are used for 3.3V or 5V supply. The kit also comes with a wire that can be used to connect the Arduino to a computer, allowing one to insert a program into the Arduino, which is then committed to memory in the microcontroller. The wiring also provides a power supply to the Arduino, which can then be further wired throughout the circuit.

SPDT Relay: The relay I will use is the Songle SRD-05VDC-SL-C.[1] The relay consists of 5 pins – three of which are dedicated to the switch within the relay, and the other two connected by an inductor which triggers the switch.[2] The inductor voltage can handle 5 volts of direct current, while the switch works for up to 10 amps of current and 250 volts. Since our household outlets run at 120Vac, the relay can handle that as well, but the wiring would need to be modified. For this reason, I have ordered a part which consists of the same relay, but it is connected to a platform with wiring pins that can be used on 120 Vac zip cord wires.

IV. TIMELINE OF IMPLEMENTATION

This step-by-step implementation plan uses an LED connected in series to a resistor as the demo version of the final product:

1. Test out Arduino motion sensor by running code with distance measurement.

2. Ensure parts properly function to circuit specifics.

3. Characterize motion sensor sensitivity, establishing threshold for switch activation.

4. If parts do not properly function, order other circuit parts via Amazon for easy delivery.

5. Assemble schematic using a Breadboard, using resistor and LED to simulate lights.

6. Improve code to trigger sensor or switch.

7. Prepare slides for the open house presentation.

8. Apply schematic to string lights and other household products that use an outlet (note the voltage difference!).

9. Disconnect the microcontroller from the computer, using a battery as the power supply and a drive to hold the programming for the circuit.

V. CHALLENGES AND AREAS FOR IMPROVEMENT

The main issue with using the motion sensor is the constraint of switch location relative to the household product. Not only does the motion sensor have to be in an area where there is little to no interference of household objects, but the sensor also has to be in close proximity to the lights. One way to improve upon this is to find a way to wirelessly transmit information. This way the motion sensor can be placed anywhere without much constraint.

Another way to improve on the circuit is to consider the temperature and humidity in the homeowner's environment, as both of those factors affect the speed of sound, which in turn affects the conversion from time delay to distance.[3] I can account for this in the Arduino code by inserting sensors for temperature and humidity, which can transmit their information to the Arduino for use in calculating the conversion. This is more of an issue for the homeowner who is using this switch in an outside environment, where the seasonal weather can affect the temperature and humidity from one day to the next.

REFERENCES

[1] *Grove - SPDT Relay(30A)*. 2021. URL: https://wiki.seeedstudio.com/Grove-SPDT_Relay_30A/.

[2] P. Marian. *Single Pole Double Throw SPDT Relay*. 2014. URL: https://www.electroschematics.com/spdt-relay-switch/.

[3] *Using the HC-SR04 Ultrasonic Distance Sensor with Arduino - Everything you need to know!* 2017. URL: https://www.youtube.com/watch?v=6F1B_N6LuKw.