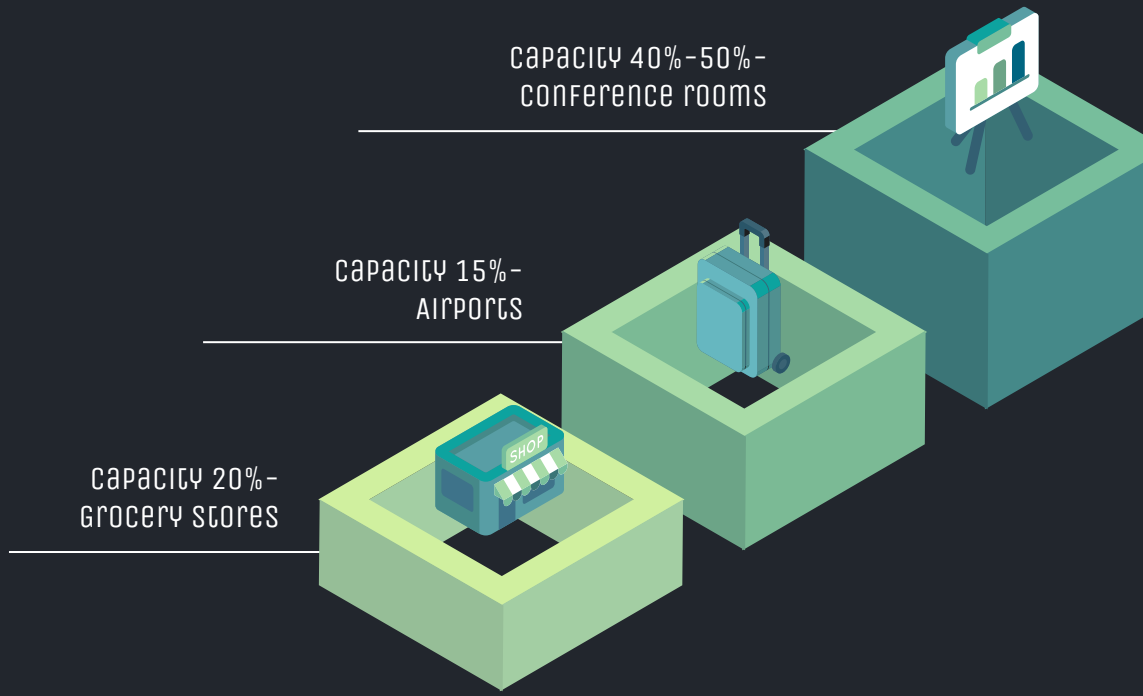
Track Pack

Jackson Teel and Vicky Vo

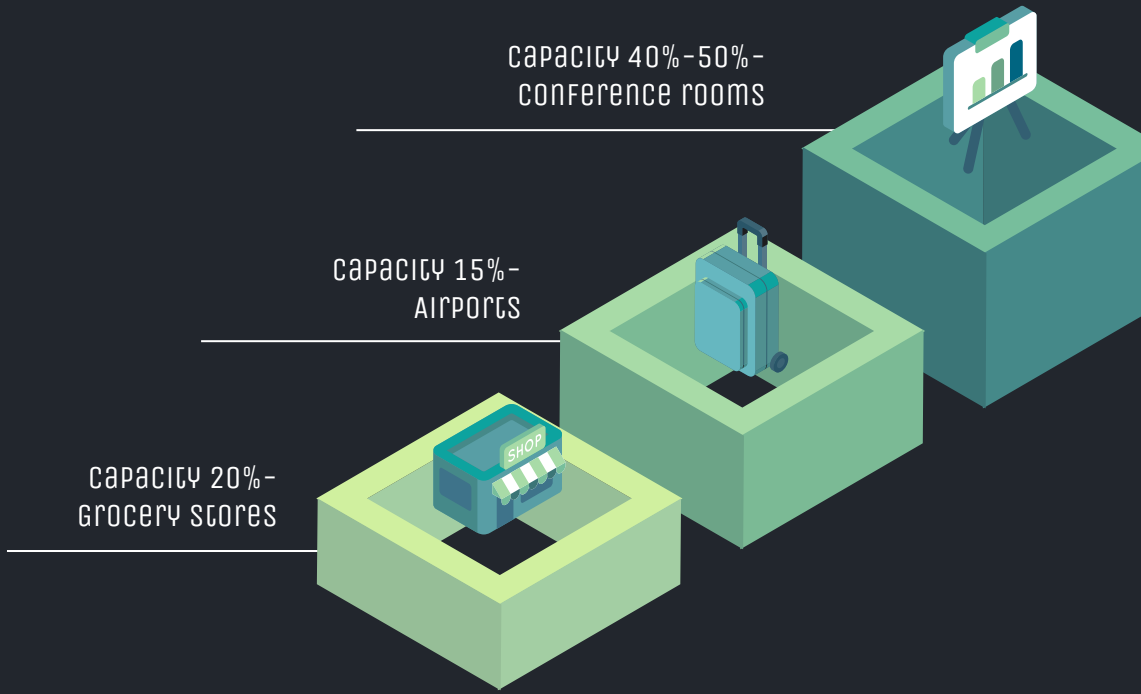
Application of Project

- Coronavirus (COVID-19) is an illness caused by a virus that can spread from person to person.
- The World Health Organization and other experts have said COVID-19 is spread mainly by large droplets sprayed when people cough or sneeze. Suggesting that 6ft (2 meters) might not be enough since a sneeze can travel up to 8 meters.
- Normal room capacity of 135,000ft is 11232 people. Now: with 7 feet apart 2,755 people.



Application of Project

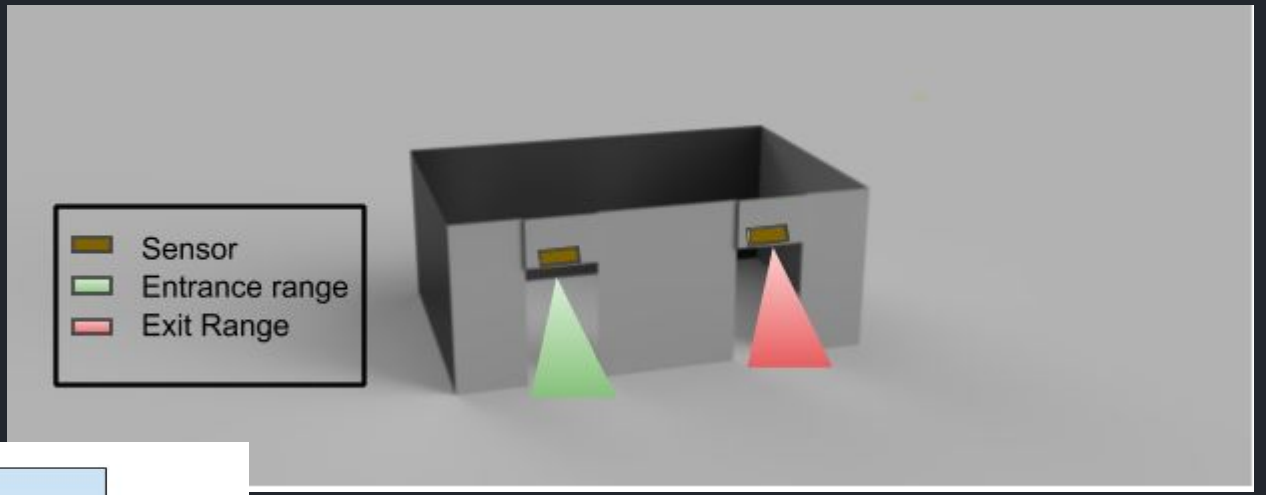
For stores that are opening back up and allowing customers inside, they will most likely be implementing some sort of limit to the amount of people allowed within the store. A tracker can help count the number of people within the store at all times and effectively notify managers and staff when the capacity has been reached. This with help business continue while considering the health and safety of customers and workers.

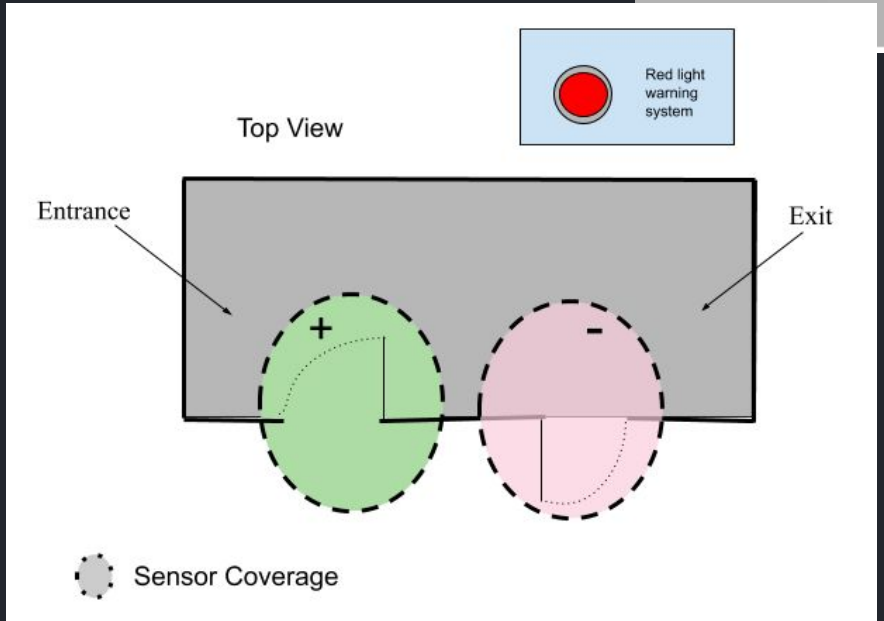


our visions

We will use a tracking device to monitor the number of people congregating within the store. Then we will proceed to use the device to issue a warning for when the capacity of the store is being reached. This will help to stop overcrowding, a very real concern given the circumstances.

our visions





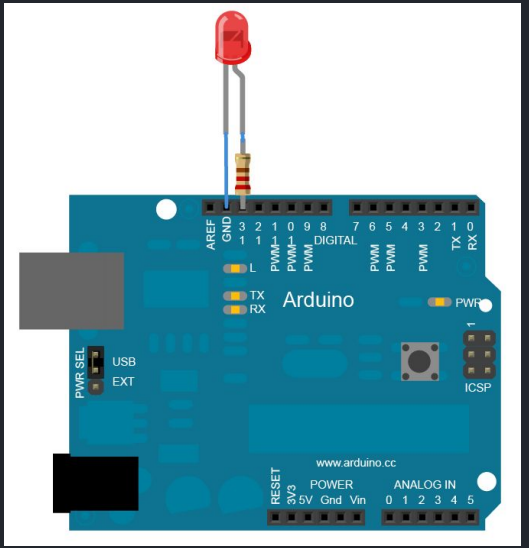
Top View

Materials

1. (2) Motion Detection Sensors (HC-SR501) + Motion Sensor Holders
2. (3) LED Lights
3. Arduino Circuit Board
4. (2) Sets of : VCC Wire, GND Wire, OUT Wire

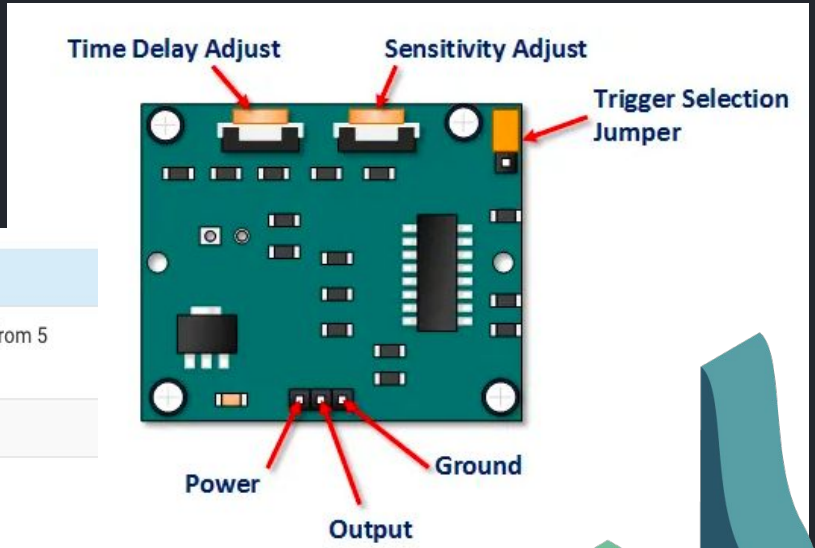
HC-SR501 Motion Sensor

This motion sensor module uses the LHI778 Passive Infrared Sensor and the BISS0001 IC to control how motion is detected.



HC-SR501 Controls

The SR501 will detect infrared changes and if interpreted as motion, will set its output low. What is or is not interpreted as motion is largely dependent on user settings and adjustments. The PIR Range (Sensitivity) Adjustment range is from approximately 3 to 7 meters.



Pin or Control	Function
Time Delay Adjust	Sets how long the output remains high after detecting motion.... Anywhere from 5 seconds to 5 minutes.
Sensitivity Adjust	Sets the detection range.... from 3 meters to 7 meters
Trigger Selection Jumper	Set for single or repeatable triggers.
Ground pin	Ground input
Output Pin	Low when no motion is detected.. High when motion is detected. High is 3.3V
Power Pin	5 to 20 VDC Supply input

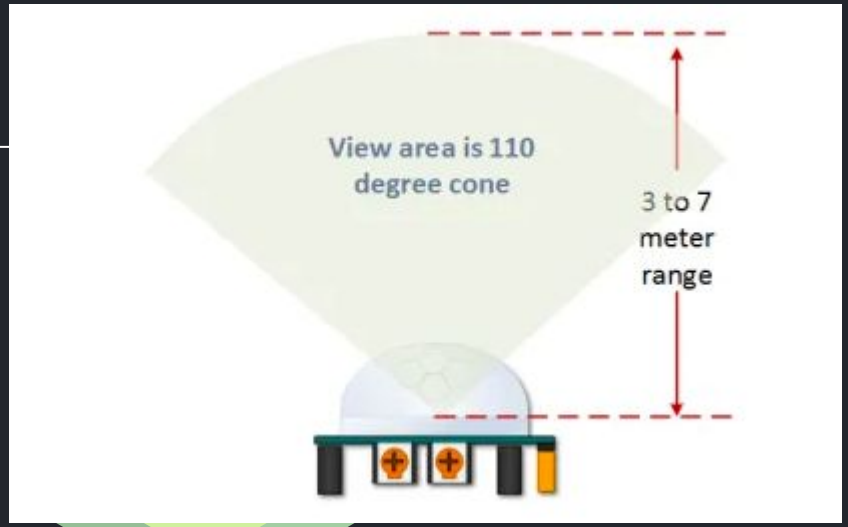
HC-SR501


Temperature Sensor
L = No Repeat
H = Repeat
Light Sensor
Time
Sensitivity
Ground
Output
VCC (4.5 -20 V)

Specifications

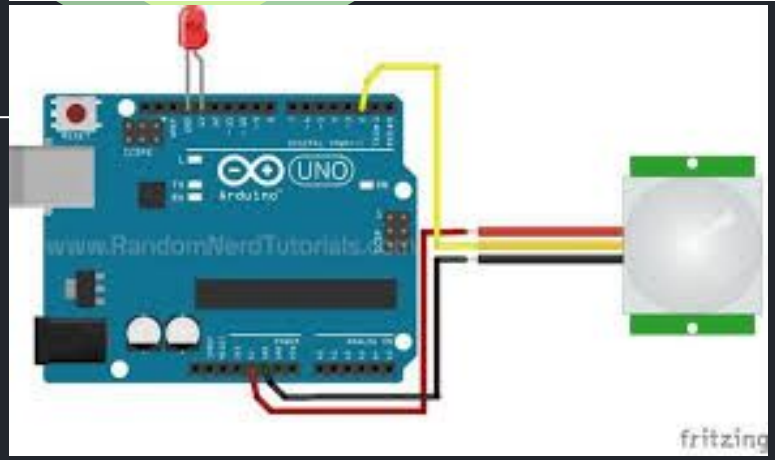
Little Sensitivity

We want to read the people as soon as they come in, we don't want to still be in the process of sensing the same person when they are far from the sensor



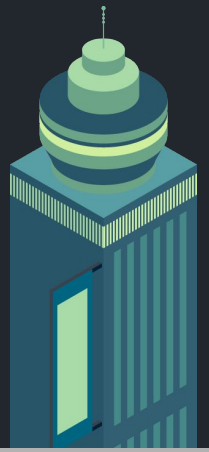
Little Time Delay

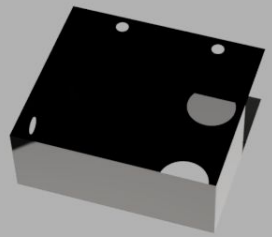
We want small time delay because we want the counter to be ready to count more people as they come in, and not be stuck on the same person for too long.

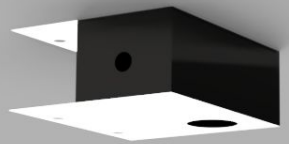


Placement of sensor

The placement of the sensor will be right above the doors of the establishment. For simplicity, we will assume that each time a person enters, they are using a door that says entrance, and for each person that exits, they are using a door that says exit.

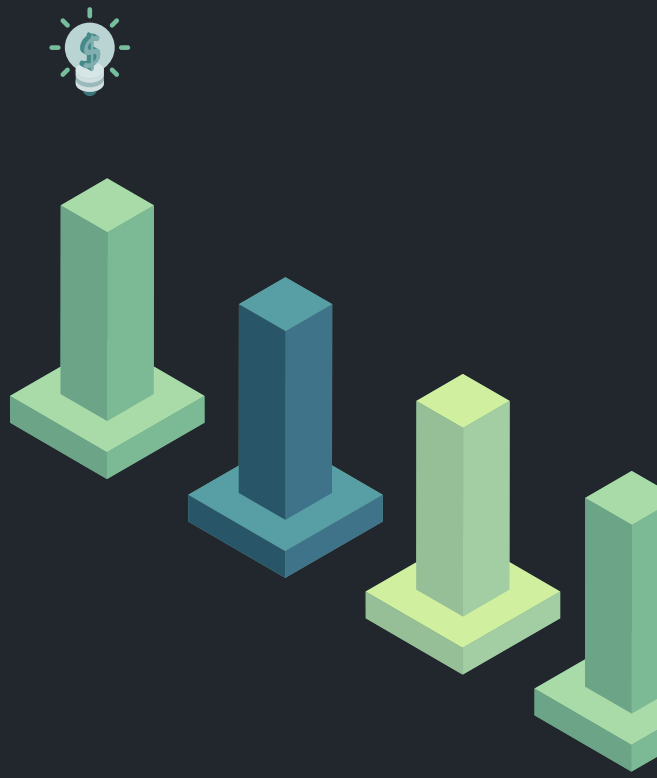






THE LED + counter

Everytime a body is detected by the motion sensor, we have the the LED light flash and update a counter variable. This variable is stored in the outer class and is updated each time a body is detected by either the entering sensor, or exiting sensor. When the counter reaches 10, the LED light will stay on, notifying that max capacity has been reached.



```java
package arduino;

import java.io.InputStream;
import java.io.OutputStream;
// import gnu.io.CommPortIdentifier;
// import gnu.io.SerialPort;
// import gnu.io.SerialPortEvent;
// import gnu.io.SerialPortEventListener;
import java.util.Enumeration;


public class TrackerProgram{

private int counter = 0;

public int getCounter(){
    return this.counter;
}

public void setCounter(int counter){
    this.counter = counter;
}


    public class EnteringSensor extends TrackerProgram{

        //these numbers will represent voltage
        private static final int LOW = 0;
        private static final int HIGH = 5;

        int counter = super.getCounter();
        private int pinNum = 13;
        private int LED = 2;
        private int value = 0;
        private int pirState = LOW;

        void setUp(){
            //sets the input and output to the corresponding pins.
            pinMode(this.pinNum, INPUT);
            pinMode(this.LED, OUTPUT);
            //initailizes serial comm at 9600 bits per second
            Serial.begin(9600);
        }

        void loop(){
            //will read the value associated with the given pin
            this.value = digitalRead(this.pinNum);

            //will handle the case when a body is present
            if(this.value == this.HIGH){

                //will send signal to turn on LED light
                digitalWrite(this.LED, this.HIGH);

                //will change the state of pirSensor
                if(this.pirState == this.LOW){
                    Serial.println("Detected body");
                    this.pirState = this.HIGH;
                }

            }

            //will turn off the LED
            else{
                digitalWrite(this.LED, this.LOW);
            }

            //will reset the state of the PIRsensor and add to the counter.
            if(this.pirState == this.HIGH){
                Serial.println("Body left");
                this.pirState = this.LOW;

                this.counter++;
                super.setCounter(this.counter);
```

```
        }

    }
}

public class ExitingSensor extends TrackerProgram{

    //these numbers will represent voltage
    private static final int LOW = 0;
    private static final int HIGH = 5;

    int counter = super.getCounter();
    private int pinNum = 12;
    private int LED = 4;
    private int value = 0;
    private int pirState = LOW;

    void setUp(){
        //sets the input and output to the corresponding pins.
        pinMode(this.pinNum, INPUT);
        pinMode(this.LED, OUTPUT);
        //initailizes serial comm at 9600 bits per second
        Serial.begin(9600);
    }

    void loop(){
        //will read the value associated with the given pin
        this.value = digitalRead(this.pinNum);

        //will handle the case when a body is present
        if(this.value == this.HIGH){

            //will send signal to turn on LED light
            digitalWrite(this.LED, this.LOW);

            //will change the state of pirSensor
            if(this.pirState == this.LOW){
                Serial.println("Detected body");
                this.pirState = this.HIGH;
            }

        }

        //will turn off the LED
        else{
            digitalWrite(this.LED, this.LOW);
        }

        //will reset the state of the PIRsensor and add to the counter.
        if(this.pirState == this.HIGH){
            Serial.println("Body left");
            this.pirState = this.LOW;

            this.counter--;
            super.setCounter(this.counter);
        }

    }
}

public class MaxCapacityLight extends TrackerProgram{
    //these numbers will represent voltage
    private static final int LOW = 0;
    private static final int HIGH = 5;
    private static final int maxCapacity = 10;

    int counter = super.getCounter();
    private int LED = 3;
    private int value = 0;

    //will keep the warning LED light on while store is at or above maxCap.
    public void maxLED(){
        while(this.counter >= this.maxCapacity){
            digitalWrite(this.LED, this.HIGH);
```

```
145         //will keep the warning LED light on while store is at or above maxCap.
146         public void maxLED(){
147             while(this.counter >= this.maxCapacity){
148                 digitalWrite(this.LED, this.HIGH);
149                 this.counter = super.getCounter();
150             }
151         }
152
153     }
154
155 }
156
```

Questions???

Sources:

http://henrysbench.capnfatz.com/henrys-bench/arduino-sensors-and-input/arduino-hc-sr501-motion-sensor-tutorial/

https://dronebotworkshop.com/using-pir-sensors-with-arduino-raspberry-pi/

https://www.cdc.gov/coronavirus/2019-ncov/downloads/2019-ncov-factsheet.pdf

https://supporthandbook.wordpress.com/2011/10/04/arduino-first-project-pir-java-motion-sensor/