

EE 151 Final Project:

Get The Fuck Away

Written By:

Samuel Garcia

California Polytechnic State University

San Luis Obispo, CA

sgarc171@calpoly.edu

Leo Pineda

California Polytechnic State University

San Luis Obispo, CA

lpenapin@calpoly.edu

Kelvin Shi

California Polytechnic State University

San Luis Obispo, CA

kshi04@calpoly.edu

## Abstract

Our final project was one that was heavily influenced by the idea to create an alert system and defensive mechanism for individuals who have an aspiration for personal space between them and people. It took a great deal of time, but eventually, we were able to generate and produce the required code and hardware to make our idea a reality. Following the integration of the hardware and code, our project was able to run its functionality. It is capable of detecting when a subject would bypass a given proximity, and activate its alerts and defensive mechanism subsequently. After witnessing the operation of our project, there was much to discuss whether that be in the form of a detailed explanation of our project’s functionality and different phases, or improvements and weaknesses that were notably on display after seeing our project in action. In the end, our group was successful in manufacturing a device that would satisfy the needs of those who seek space between them and society, and is something that we are truly proud of.

**Figure 1.1** The Masterpiece

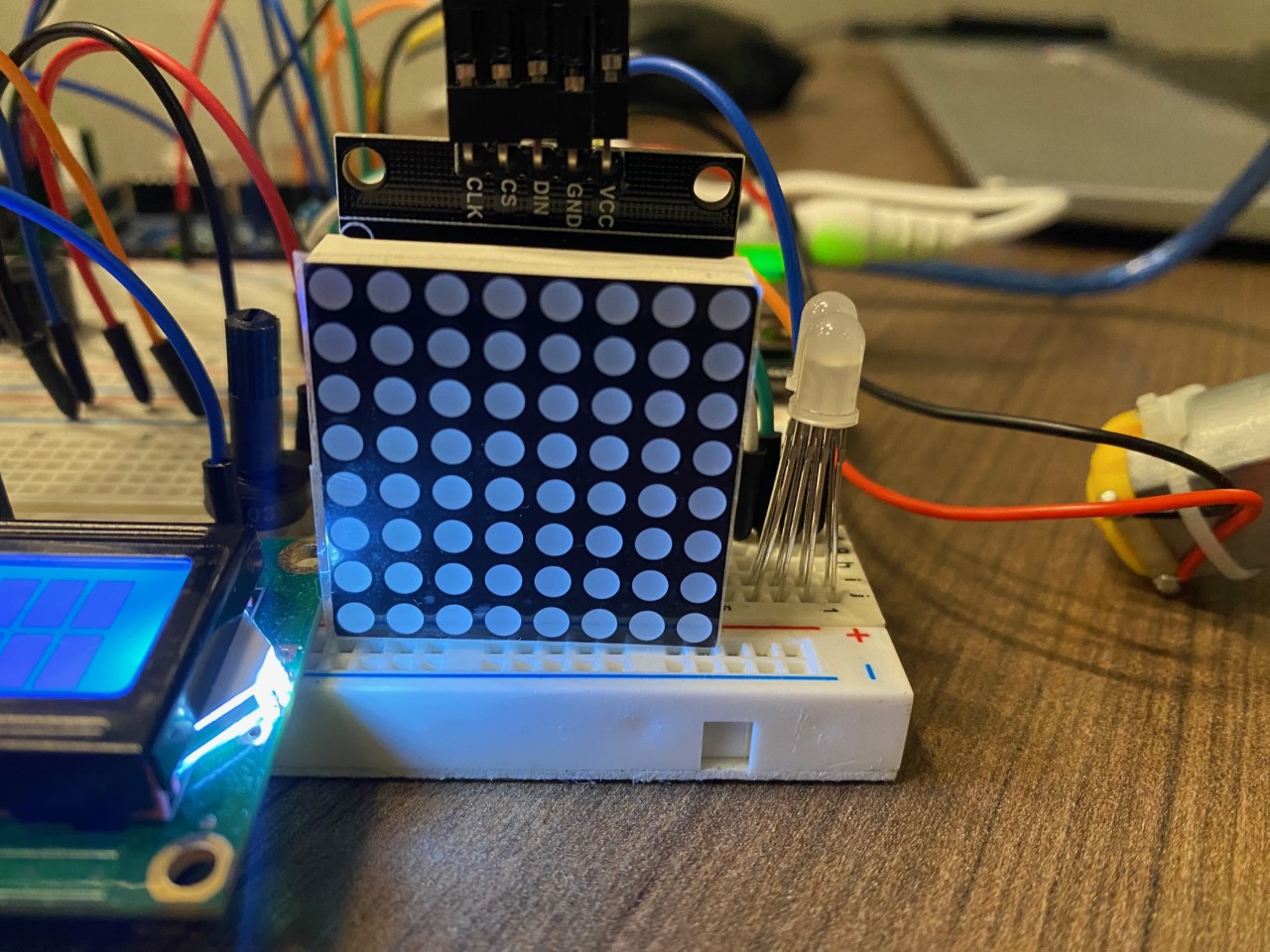
## Introduction

The overall purpose of our project was to come up with a defensive mechanism for people to use when space is desired between them and an object or person, or as the projects itself is named, to get the fuck away. To deliver and create this mechanism, heavy inspiration from Lab #5: Logic, Power, and Control, specifically the base functionality section of the lab, was taken. In the base functionality portion of the lab, the main component that was the focus of the schematic was the Ultrasonic sensor. The Ultrasonic sensor’s function was to emit an ultrasound and wait for the soundwaves to be reflected or echoed back after bouncing off an object. Once that event occurred, the distance from the object was measured and sent to the Arduino Board to trigger a detection and certain action using a fan and LED light. With this in mind, the Ultrasonic sensor was the main component in our final project, considering it would be able to detect when an object or person would cross a set proximity, triggering multiple defensive mechanisms as a result.

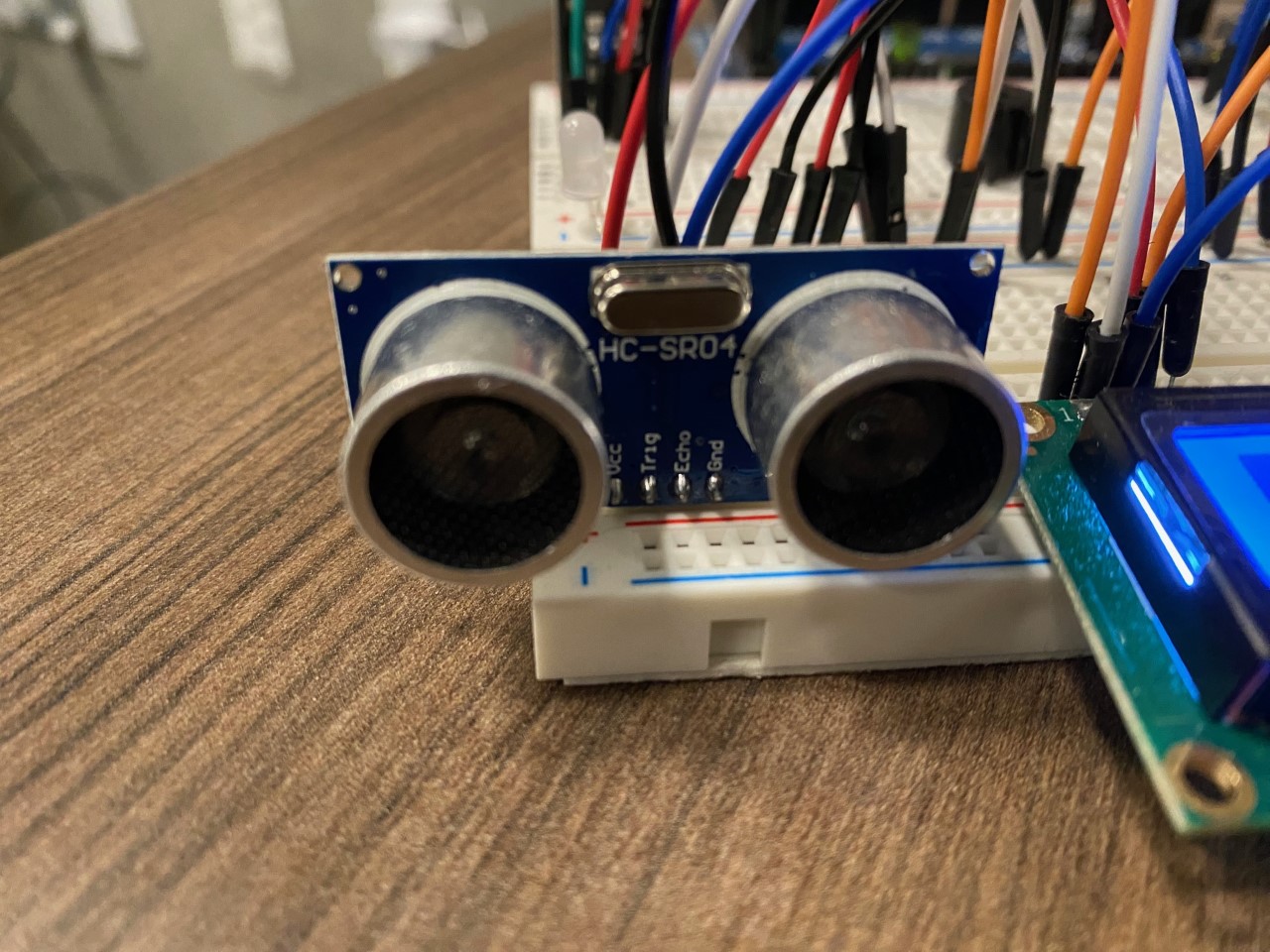
## Methods and Materials

* 830 Tie-Points Breadboard (2)
* MEGA 2560 Controller Board
* USB Cable
* LCD102 Module
* Power Supply Module

**Figure 3.1** LCD102 Module

* L293D
* MAX7219 Module
* 9V1A Adapter
* Google
* Coffee
* Power of Friendship (if applicable)
* Potentiometer 10k
* Passive Buzzer

**Figure 3.2** MAX7219 Module

* Ultrasonic Sensor
* Fan Blade and 3-6V Motor
* IR Receiver Module
* Remote Control
* RGB LED (3)
* 10k Resistor
* Female-to-Male Dupont Wire
* Breadboard Jumper Wire

**Figure 3.3** Ultrasonic Sensor

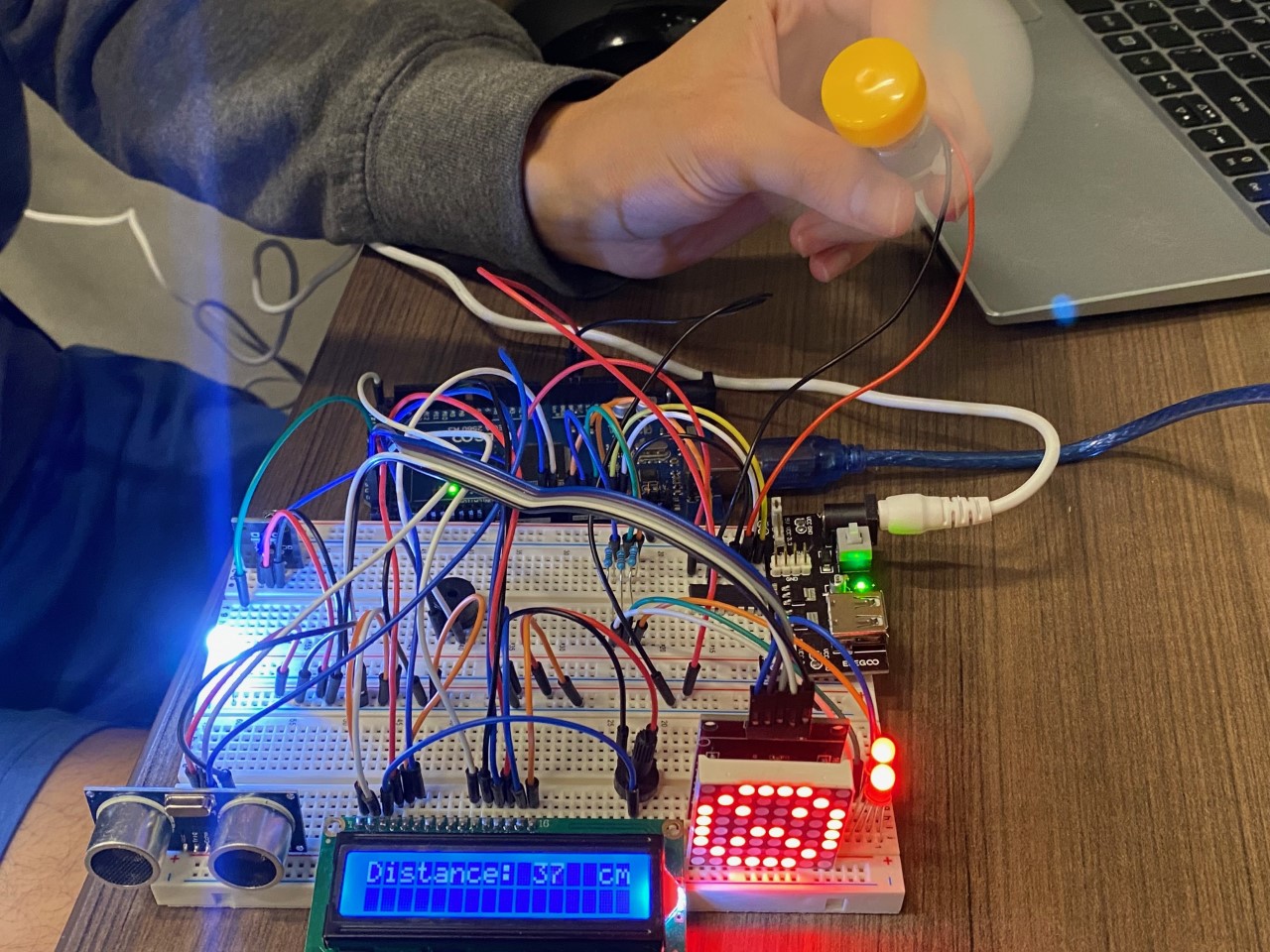
## Experimental Procedure

We started our project by brainstorming potential ideas that would be applicable and practical to everyday life. After countless hours of research on current events to find a topic, we were compelled to make a machine that enforces social distancing in the wake of the COVID-19 pandemic. Immediately after acquiring our idea, we moved onto our development phase, designing a schematic that layouts all the components and their connections needed for the functionality of the project, framing the logic with the use of a flowchart, and creating a scrum board to keep our project organized. From there, the flowchart helped massively to facilitate the software development as well as referencing past lab reports and the Arduino Kit User Manual. Following the code, a few more hours were invested wiring up the breadboard, carefully establishing connections to the right pins of the control board and to the pins of the components. After the initial prototype was built, we went through a process of testing and troubleshooting to ensure that all parts of the project, both hardware and software, were working as intended. In the end, we had successfully designed and constructed a machine that alerts surrounding people if they are too close and invading personal space.

## Results

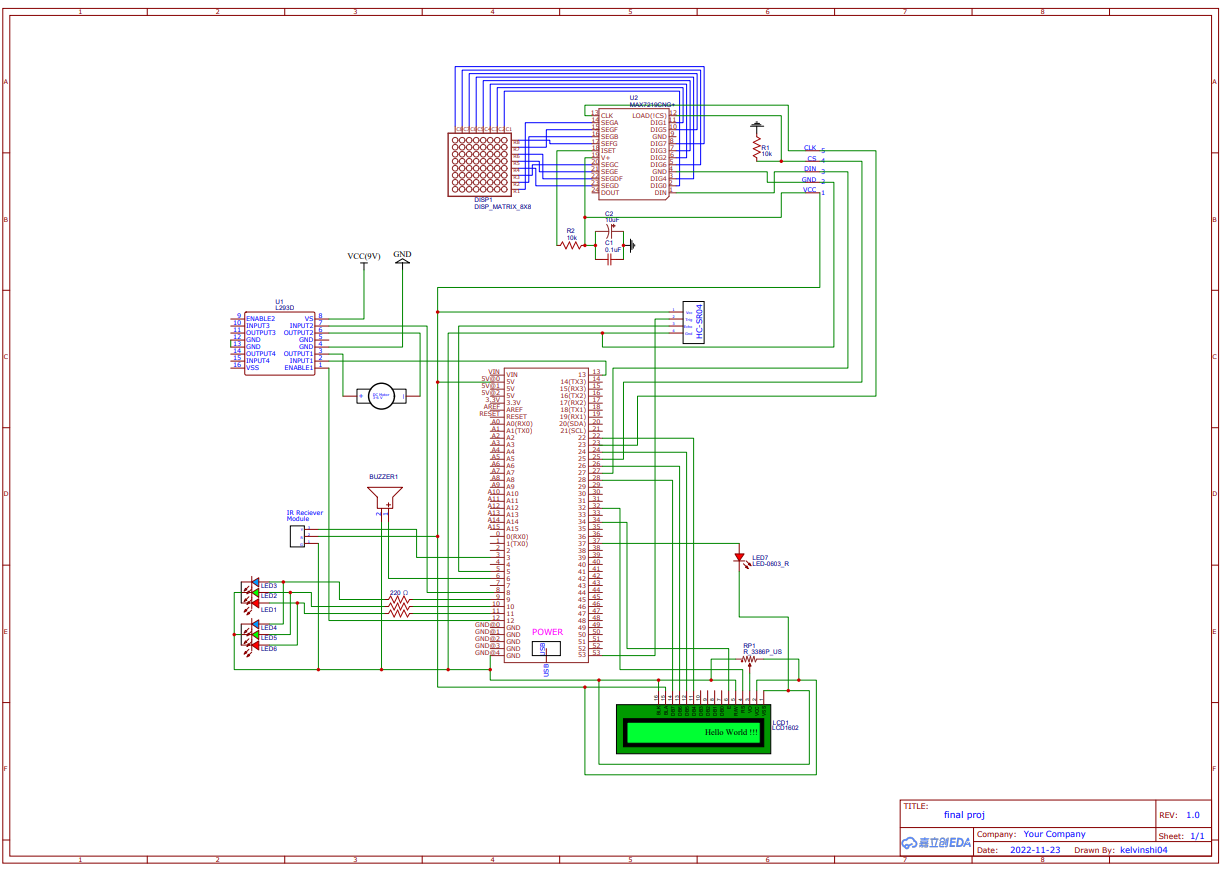
**Figure 5.1**

Project when no subject was detected.



**Figure 5.2**

Project when a subject was detected.



**Figure 5.3**

Schematic of project.

## Discussion

**Explanation:**

After all was said and done with the final project, the end result was one that one can be greatly proud of. All in all, we were able to achieve the overall purpose of our project as we were able to successfully create a defensive mechanism for people to use when space is desired between them and an object or person. The main components of our project focused around the Ultrasonic sensor and , which served as our indication and detection of subjects, while the Passive Buzzer, Fan, 2 of the RGB LEDs, and MAX7219 Module played the role of our defensive mechanisms when a set proximity was passed (refer to V. Results for visual aid). It should be noted as well that the project could turn off and on, while also being able to adjust the volume of the Passive Buzzer with the use of the Remote Control. Once all the components were put together, the only thing left to do was to power on our project and see it in use. To serve as an indication for both ourselves and the given user, once powered on and connected, a set RGB LED would light up, the LCD102 would start displaying distance from the closest object in the proximity, the MAX7219 Module would project a smiley face being that the coast was clear, and the other 2 RGB LEDs would display green for the same reason. When powered off, all the RGB LEDs and MAX7219 Module would be turned off, while the LCD102 would display that the project was off. Bringing the focus back to when the project itself was in use, the process would begin with the Ultrasonic sensor. The Ultrasonic sensor would constantly be sending out ultrasounds for detection. When there was no detection, all remained the same as it did when powered on. However, in the event when there was detection, stuff went down. First and foremost a very unpleasant Passive Buzzer would start to emit a very loud and annoying frequency. Think of your alarm to wake up, just five times worse. Moving on, the MAX7219 Module would go from having a smiley face to a frowny face. Along with the frowny face, the 2 RGB LEDs that were once green to indicate everything was sunshine and rainbows, turned red to indicate everything was indeed no longer sunshine and rainbows. Last but not least, the fan would turn on in an attempt to blow the enemy away. All these actions would continue to take place until the given enemy finally would get the fuck the away.

**Reflection:**

In regard to the general efficiency of our project, we were able to identify both notable limitations and possible improvements, while also being able to identify the key strengths of our project. The limitations and improvements are both related to the Ultrasonic Sensor and Fan. Focusing on the Ultrasonic sensor, the distance it would be able to cover/detect efficiently, without error, was limited to about 100 cm. It was until we reached over 100 cm that indication would not be able to be made. So in essence, if a consumer had the desire for someone to stay more than 100 cm away, our project would not be able to effectively do so. However, to be able to counteract this issue, a more advanced and powered Ultrasonic Sensor could come to solve it. We notably found an Ultrasonic Sensor known as the LiDAR Lite V3, which has a range of 40 m, that could serve as that improvement in Ultrasonic Sensor role. With reference to the fan, obviously given its size and power, it would not be able to successfully blow someone anyway. A much more powerful and bigger fan would be desired to satisfy this function. Focusing on the key strengths of our project, the project as a whole and its components all served their functionality. When identifying a subject within 100 cm, the Ultrasonic sensor proficiently detected the subject immediately and activated all the alerts and defensive mechanisms accordingly. The alerts and mechanisms would be activated instantaneously, and once a subject would no longer be detected, all would recuperate back to its original state. Throughout that process, none of the components ever malfunctioned, became out of sync, or diminished in productivity. With that in mind, it would be indiscreet to not proclaim that the project overall was strong in every component with respect to its max potential.

## Closing Statement

In closing, given the resources that were available to us, as a team we can comfortably say that we accomplished the goal we sought. That being to create a mechanism that’d get someone the fuck away. Our group designed and manufactured both the code and hardware needed to satisfy our objective, much of which could not have occurred if not for the knowledge we learned from EE 151 and the G.O.A.T. himself, Professor RJ Macaranas.

**Figure 7.1** The Final Project

### References

Elegoo Incorporation. (2020, October 21). *THE MOST COMPLETE STARTER KIT TUTORIAL FOR MEGA2560*. Shenzhen, the Silicon Valley of China.

Macaranas, R. J. (2022, October 18). *EE 151. Introduction to Electrical Engineering*. San Luis Obispo; California Polytechnic State University.

GitHub Inc. (2022). *EE-Final-Project.* <https://github.com/LeonardoPenaPineda/EE-151-Final-Project.git>

### Appendix

|  |
| --- |
| /\*  file name: final\_proj.ino  authors: Kelvin Shi  revision date: 11/15/2022  \*/  #include "LedControl.h"  #include <LiquidCrystal.h>  #include <IRremote.h>  LedControl lc = LedControl(27,23,25,1);  LiquidCrystal lcd(32, 34, 22, 24, 26, 28);  #define LED 37  #define ENABLE 12  #define DIRA 8  #define DIRB 13  #define echoPin 5  #define trigPin 53  long duration;  int distance;  #define BLUE 9  #define GREEN 10  #define RED 11  int redValue = 0;  int greenValue = 0;  int blueValue = 0;  int buzzerValue = 1000;  bool on = false;  int IRPIN = 3;  IRrecv irrecv(IRPIN);  decode\_results results;  #define buzzer 6  void setup() {  lc.shutdown(0, false);  lc.setIntensity(0, 5);  lc.clearDisplay(0);  lcd.begin(16, 2);  lcd.setCursor(0, 0);  lcd.print("Distance:");  pinMode(LED, OUTPUT);  pinMode(ENABLE,OUTPUT);  pinMode(DIRA,OUTPUT);  pinMode(DIRB,OUTPUT);  pinMode(trigPin, OUTPUT);  pinMode(echoPin, INPUT);  pinMode(RED, OUTPUT);  pinMode(GREEN, OUTPUT);  pinMode(BLUE, OUTPUT);  irrecv.enableIRIn();  }  void statusLED() {  if (on == true) {  digitalWrite(LED, HIGH);  }  if (on == false) {  digitalWrite(LED, LOW);  }  }  void translateIR() {  switch(results.value) {    case 0xFFA25D: on = true; statusLED(); break;  case 0xFF629D: if (buzzerValue < 1000) {buzzerValue += 200;}; break;  case 0xFFA857: if (buzzerValue > 0) {buzzerValue -= 200;}; break;  case 0xFF02FD: on = false; statusLED(); break;  default: lcd.setCursor(0, 1); lcd.write(" "); delay(200); lcd.setCursor(0, 1); lcd.write(" ");  }  }  void writeHappy() {  byte happy[8] = {B00111100, B01000010, B10101001, B10000101, B10000101, B10101001, B01000010, B00111100};  lc.setRow(0,0,happy[0]);  lc.setRow(0,1,happy[1]);  lc.setRow(0,2,happy[2]);  lc.setRow(0,3,happy[3]);  lc.setRow(0,4,happy[4]);  lc.setRow(0,5,happy[5]);  lc.setRow(0,6,happy[6]);  lc.setRow(0,7,happy[7]);  }  void writeSad() {  byte sad[8] = {B00111100, B01000010, B10100101, B10001001, B10001001, B10100101, B01000010, B00111100};  lc.setRow(0,0,sad[0]);  lc.setRow(0,1,sad[1]);  lc.setRow(0,2,sad[2]);  lc.setRow(0,3,sad[3]);  lc.setRow(0,4,sad[4]);  lc.setRow(0,5,sad[5]);  lc.setRow(0,6,sad[6]);  lc.setRow(0,7,sad[7]);  }  void checkDistance() {  digitalWrite(trigPin, LOW);  delayMicroseconds(2);  digitalWrite(trigPin, HIGH);  delayMicroseconds(10);  digitalWrite(trigPin, LOW);  duration = pulseIn(echoPin, HIGH);  distance = duration \* 0.034 / 2;  lcd.setCursor(0, 0);  lcd.print("Distance:");  lcd.setCursor(10,0);  lcd.print(" cm");  lcd.setCursor(10,0);  lcd.print(distance);  delay(200);  }  void loop() {    while (on == true){  checkDistance();  lcd.clear();  while (distance < 50) {  analogWrite(GREEN, 0);  redValue = 255;  blueValue = 3;  //led go up  for (int i = 0; i < 201; i+=3) {  redValue -=3;  blueValue +=4;  analogWrite(RED, redValue);  analogWrite(BLUE, blueValue);  delay(10);  }  if (irrecv.decode(&results)) {  translateIR();  irrecv.resume();  }    analogWrite(buzzer, buzzerValue);  digitalWrite(ENABLE, HIGH);  digitalWrite(DIRA, LOW);  digitalWrite(DIRB, HIGH);  lc.clearDisplay(0);  writeSad();  delay(100);  checkDistance();  }  analogWrite(buzzer, 0);  digitalWrite(ENABLE, LOW);  lc.clearDisplay(0);  writeHappy();  analogWrite(BLUE, 0);  analogWrite(RED, 0);  analogWrite(GREEN, 255);  if (irrecv.decode(&results)) {  translateIR();  irrecv.resume();  }  }  if (on == false) {  lc.clearDisplay(0);  analogWrite(buzzer, 0);  digitalWrite(echoPin, LOW);  digitalWrite(ENABLE, LOW);  lcd.clear();  lcd.setCursor(6, 0);  lcd.write("OFF!");  analogWrite(GREEN, 0);  if (irrecv.decode(&results)) {  translateIR();  irrecv.resume();  }  }  } |