Team Name/Number: Team 14

Team Member Names (Last, First): Choudhury, Arnav; Shaw, Harrison; Chon, Kris

Date of submission: March 16, 2021

Remote Controlled Safe!

Design Summary:

Beginning the prototyping process was largely just throwing out ideas until things stuck. An initial design prototype that our team brainstormed was a cardboard safe that used a DC motor to open and close the safe door. Additionally, there would be a piezo speaker mounted on the top of the safe, to create an alarm effect. This would have had an infrared receiver module directly behind the safe door, with the arduino and battery being seated in the back of the safe; the infrared receiver module would take input from a user's remote, and turn on the DC motor if the correct code was input. The final component would be an ultrasonic sensor mounted on the bottom of the safe, which would detect when the safe is being moved. This would prevent the safe from being moved or destroyed, as if the ultrasonic sensor detected movement, it would cause the speaker to blare noise.

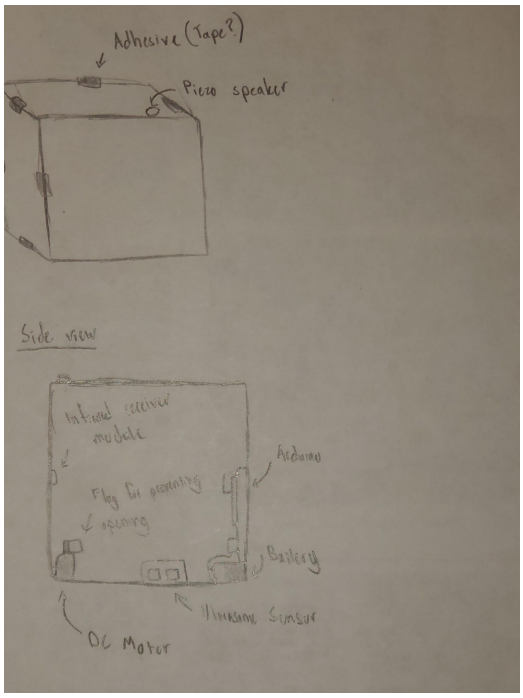

Figure A1

The second design implemented a somewhat simple design change, however, it was integral to the function of the safe as far as the door was concerned. After deliberation, we decided that a Servo motor would be better suited towards a safe's function than a DC motor, as Servo motors turn sequentially, whereas DC motors simply spin, and would need more coding to produce the same result. This would necessitate a change in the door of the safe: it would need to be separate from the front of the safe.

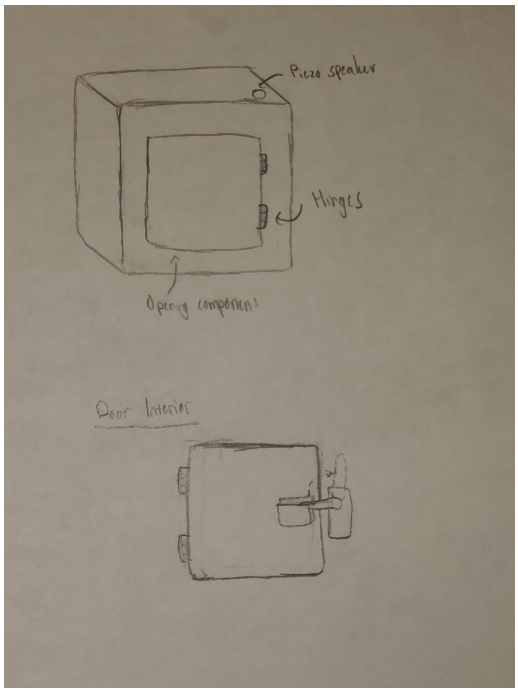

Figure A2

Another seemingly simple change came from the material of the safe. Because the safe simply needed to be a rectangular container, the team opted to use a shoebox instead of cardboard, as this would be a significantly simpler structure, and easily accessible to boot. Furthermore, the shoebox would be much more secure than cardboard, which would require adhesive to hold its structure. Also, we decided to implement a serial monitor in the safe door, to provide visual stimulus for when the safe received the right or wrong code. The serial monitor would be mounted in the center of the safe door, thus requiring the infrared receiver module to be moved to the top of the safe, next to the piezo speaker.

Admittedly, the position of the receiver module is largely irrelevant, so long as it can receive the infrared signal.

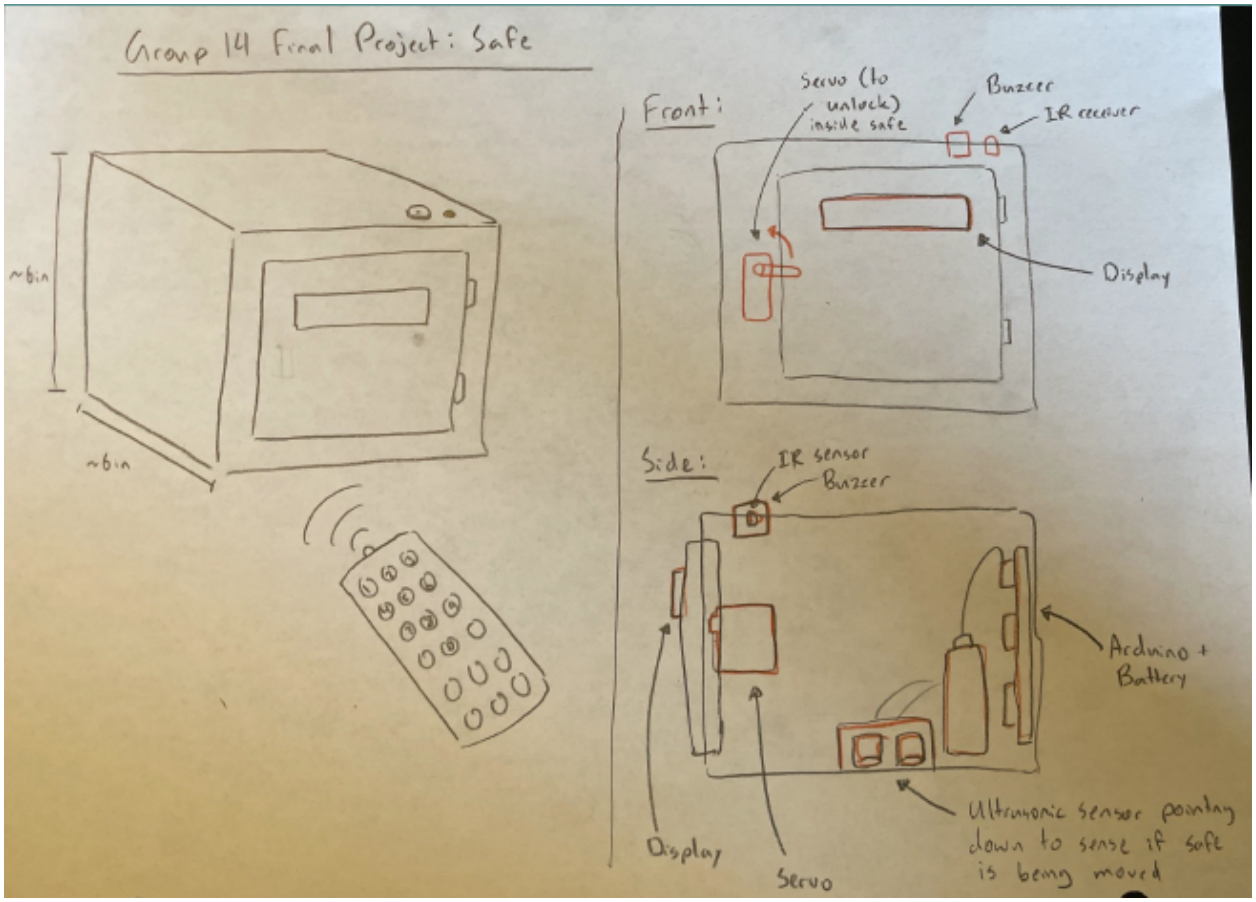

Figure A3

In finalizing the design, a great many changes were made. Due to coding and complexity constraints, as well as a general lack of necessity, the ultrasonic sensor was removed from the safe. We were still able to include the LCD display and buzzer features, to provide some visual and auditory feedback to enhance the experience. Unfortunately, due to a lack of male-to-female jumper cables, we were unable to place the arduino and general wiring on the inside of the box; consequently, the electronics were placed on the outside of the shoebox. As one would expect, this decreased the safety of the safe almost entirely, but its purpose of locking and unlocking storage was kept.

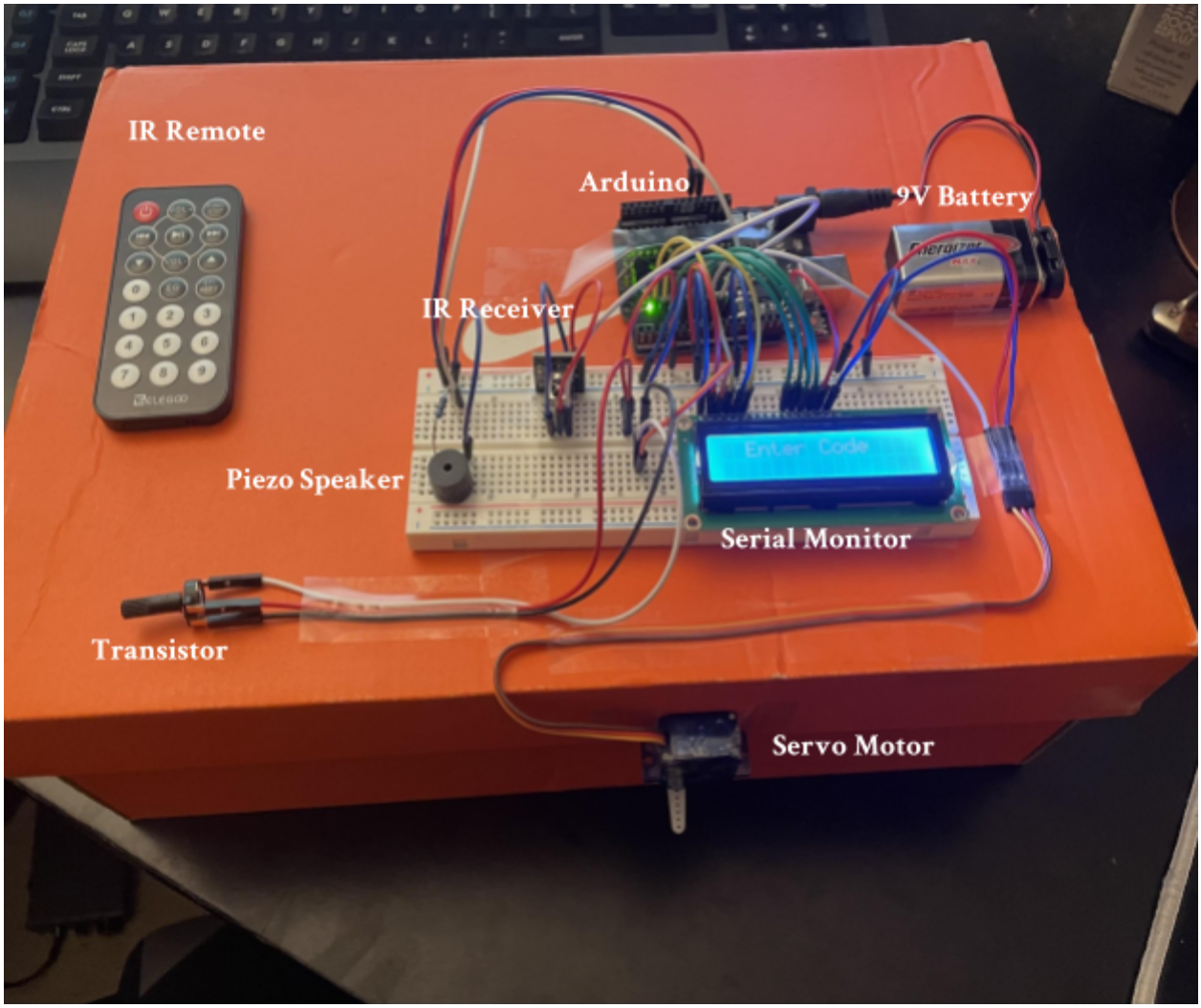

Figure A4

	Ease of Construction (1-5)	Cost Efficiency (1-5)	Stability/Security (1-5)
Prototype 1	3	4	2
Prototype 2	2	5	4
Prototype 3	4	5	4

Appendix:

Table B1: Bill of Materials for xx robot project

Item	Quantity	Description	Cost per item	Total Cost
01	1	Shoebox	$5.80*	$5.80
02	1	Infrared Receiver Module	N/A - ENGR 1 kit	
03	1	Servo Motor	N/A - ENGR 1 kit	
04	1	Potentiometer	N/A - ENGR 1 kit	
05	1	100 Ohm Resistor	N/A - ENGR 1 kit	
06	31(?)	Jumper Cable	N/A - ENGR 1 kit	
07	1	Serial Monitor	N/A - ENGR 1 kit	
08	1	Active Buzzer	N/A - ENGR 1 kit	
09	1	Transistor	N/A - ENGR 1 kit	
10	1	Breadboard	N/A - ENGR 1 kit	
		Sum of all parts - Total Cost		$5.80

*The shoebox was sitting around, so it cost nothing

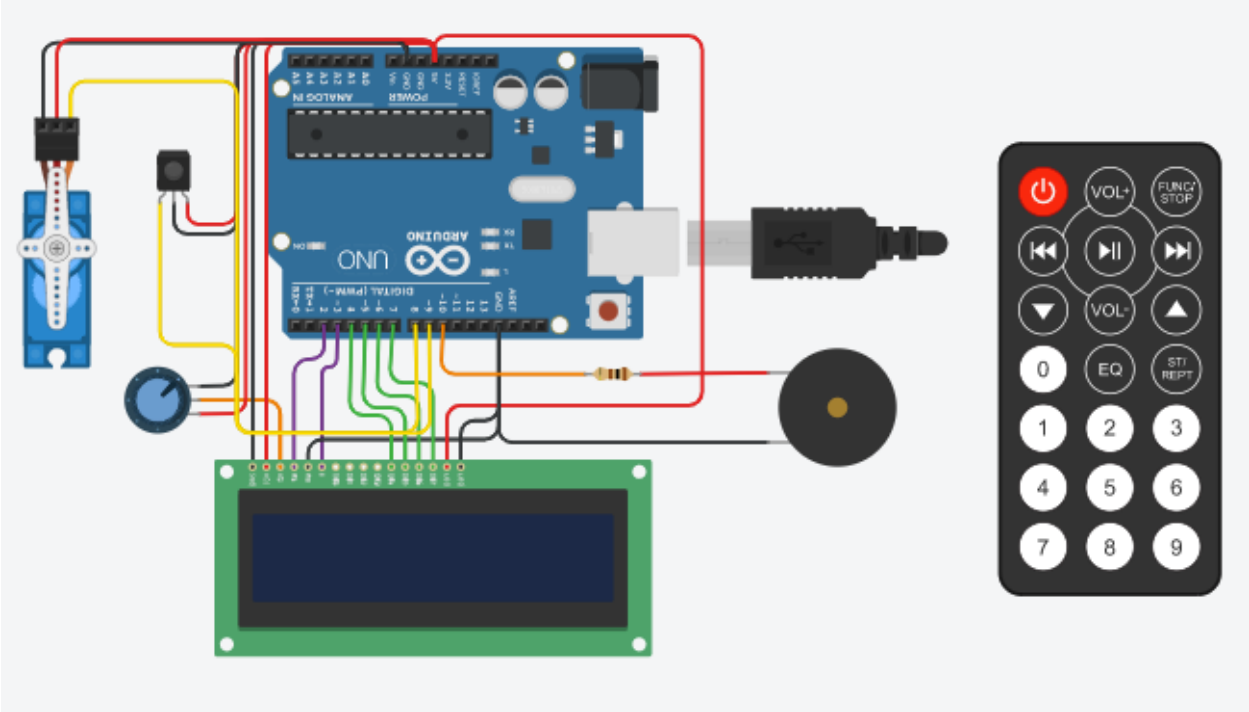

Figure B2: TinkerCAD Electronic Safe Schematic

Table A2: Team member project time log

Team Member	**Time spent on project (include individual and team time)**	
	In-class (any lab periods attended)	Out-of-class (non-lab periods)
Arnav Choudhury	6 hrs	7 hrs
Harrison Shaw	6 hrs	7 hrs
Kris Chon	6 hrs	7 hrs

Commented computer code for Remote Safe project:

The code for the safe takes an infrared input of 4 pin numbers and validates if that is correct to open the safe. The code also shows the pins used for all of the components in the project.

```
/*
Team 14
by Arnav Choudury
This program regcognizes a password and opens the lock with a key.
*/
#include<LiquidCrystal.h>
#include<IRremote.h>
#include<Servo.h>



// all the button codes from the elegoo remote
#define NUM0  0xFF6897
#define NUM1  0xFF30CF
#define NUM2  0xFF18E7
#define NUM3  0xFF7AB5
#define NUM4  0xFF10EF
#define NUM5  0xFF38C7
#define NUM6  0xFF5AA5
#define NUM7  0xFF42BD
#define NUM8  0xFF4AB5
#define NUM9  0xFF52AD
#define REST  0xFFB04F
#define HELD  0xFFFFFF

#define SERVO_PIN 8

#define IR_PIN 9

#define P_PIN 11

IRrecv irrecv(IR_PIN);
decode_results results;

Servo servo;
LiquidCrystal lcd = LiquidCrystal(2, 3, 4, 5, 6, 7);
int password[] ={0,1,2,3};
int entered[]={-1,-1,-1,-1};

void setup() {
  //serial setup
  Serial.begin(9600);
  //servo setup
  servo.attach(SERVO_PIN);
  servo.write(0);
  //screen set up
  lcd.begin(16,2);
  //set up ir sensor
  irrecv.enableIRIn();
  irrecv.blink13(true);

}
int i=0;
```

```
void loop()
{
  //Serial.println(NUM0,HEX);
  lcd.setCursor(2,0);
  lcd.write("Enter Code");
  if (irrecv.decode(&results))// Returns 0 if no data ready, 1 if data ready.
  {
    Serial.println("RECIEVED Button");
    if(nextPin(results.value)>-1){
      Serial.println("Got a number");
      entered[i]=nextPin(results.value);
      i++;
      delay(100);
    }
    else if(nextPin(results.value)==-2){
      servo.write(0);
      entered[0]=-1;
      entered[1]=-1;
      entered[2]=-1;
      entered[3]=-1;
      i=0;
    }
    irrecv.resume();

  }
  if(i==4){//length of password has to equal currentConditon
      if(passWordCheck(entered,password)){
        servo.write(100);
        lcd.setCursor(2,1);
        lcd.print("Correct Code!");
        entered[0]=-1;
        entered[1]=-1;
        entered[2]=-1;
        entered[3]=-1;
        i=0;
        delay(1500);
        lcd.clear();
      }
      else{
        lcd.setCursor(2,1);
        lcd.print("Wrong Code");
        entered[0]=-1;
        entered[1]=-1;
        entered[2]=-1;
        entered[3]=-1;

        i=0;
        delay(1500);


        delay(500);
        lcd.clear();
      }
  }
}
bool passWordCheck(int currentCondition[],int password[]){
  //checks if the password matches
  for(int i=0;i<4;i++){
    if(currentCondition[i]!=password[i]){
      return false;
    }

  }
 return true;
}
```

```
int nextPin(int nextEntry){
  //matches the ircodes with the number for a pin
  switch(nextEntry){
    case NUM0:
    Serial.println(0);
    return 0;
    break;
    case NUM1:
    Serial.println(1);
    return 1;
    break;
    case NUM2:
    Serial.println(2);
    return 2;
    break;
    case NUM3:
    Serial.println(3);
    return 3;
    break;
    case NUM4:
    Serial.println(4);
    return 4;
    break;
    case NUM5:
    Serial.println(5);
    return 5;
    break;
    case NUM6:
    Serial.println(6);
    return 6;
    break;
    case NUM7:
    Serial.println(7);
    return 7;
    break;
    case NUM8:
    Serial.println(8);
    return 8;
    break;
    case NUM9:
    Serial.println(9);
    return 9;
    break;
    case HELD:
    Serial.println("HELD");
    return -1;
    break;
    case REST:
    Serial.println("Reset");
    return -2;
    break;
    default:
    Serial.println("Not a button");
    return -1;
    break;
  }
}
```