

ENGINEERING PROJECT PORTFOLIO

Created By: Aditya Bangalore

Introduction

The following pages include documentation on the various electronics engineering projects I have completed. A table of contents is included below.

Github for Code Access: <https://github.com/adityabangalore>

Table of Contents

OBSTACLE AVOIDANCE ROBOT.....	2
SEVEN SEGMENT GAME.....	7
AUTOMATIC NIGHT LIGHT.....	19
SECURITY SYSTEM PANEL.....	24
REFERENCES & CITATIONS.....	36

OBSTACLE AVOIDANCE ROBOT

Objective/Description

Create a robot that drives forward until an object is detected, at which point it turns to avoid the object and continues moving forward. If the robot turns and there is still an object there, it will turn the other way for twice the amount of time, to, hopefully, avoid the other obstacle.

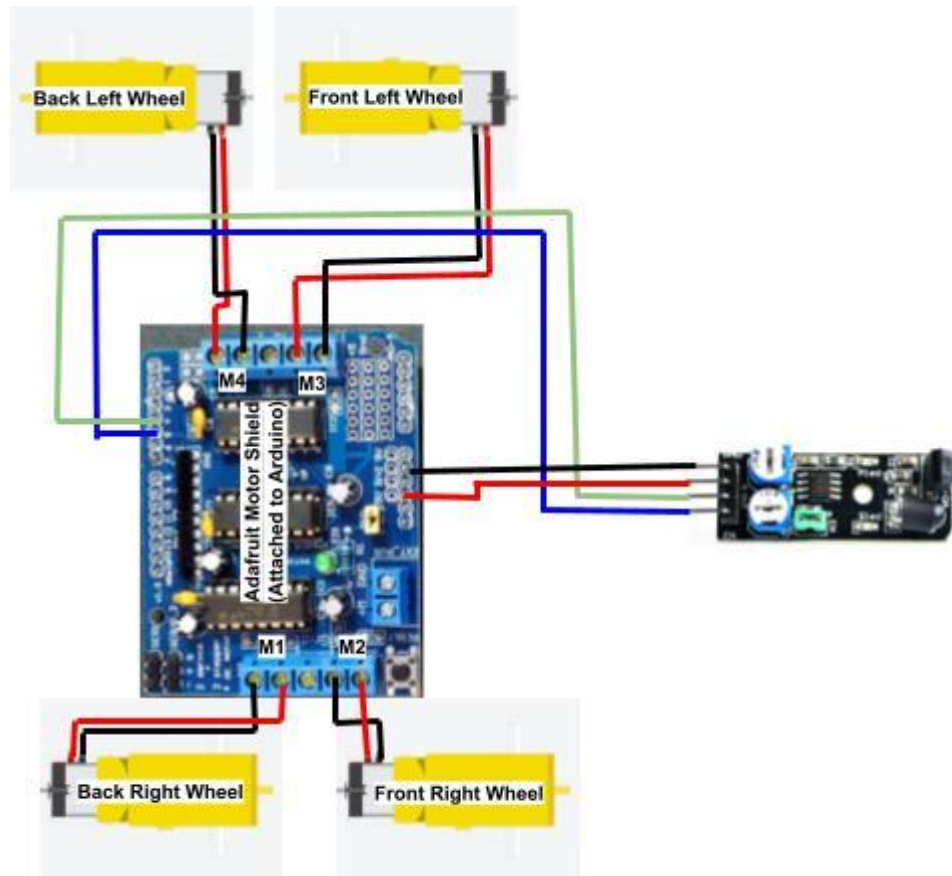
Materials

Arduino Uno
Adafruit Motor and Servo Shield
KY-032 Infrared Sensor Module
Portable Battery Pack
Robot Chassis w/ 4 DC Motors and Wires
Jumper Wires (M to F)
Screws and Nuts
Spacers

Assembly

1. Robot chassis was assembled as per instructions in the purchased robot chassis kit.
2. Arduino Uno was mounted using 3D printed spacers (refer to CAD section of this documentation), screws, and nuts.
3. Adafruit Motor Shield was attached to Arduino Uno through Arduino I/O pins
4. Duct tape was used to attach the KY-032 Infrared sensor to the front of the robot.
5. Duct tape was used to attach the portable battery pack to the top of the robot.
6. Motors and KY-032 sensor were wired as per the wiring diagram below.

Wiring Diagram



The table below indicates which pin on the KY-032 sensor module was connected to which pin on the motor shield.

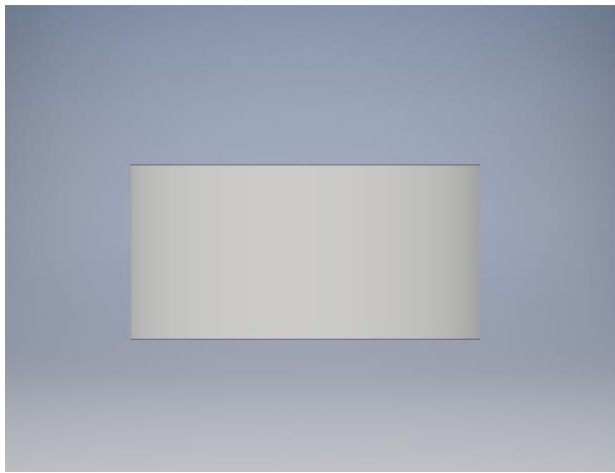
KY-032 Pin	Motor Shield Pin
Enable	5
Out	4
+	5V
Gnd	Gnd

(Continued on Next Page)

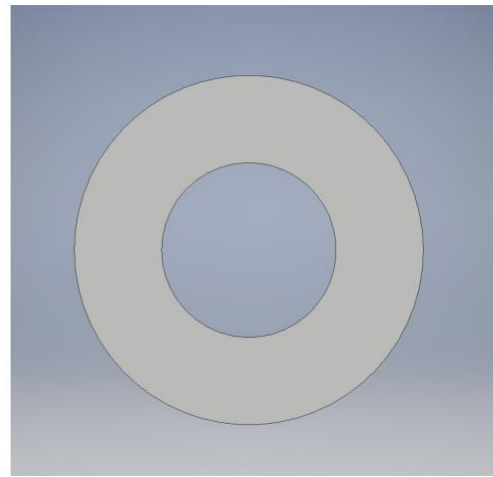
CAD

Below is an image of the design for the spacers that were used to mount the Arduino onto the robot chassis. It was produced using Autodesk Inventor.

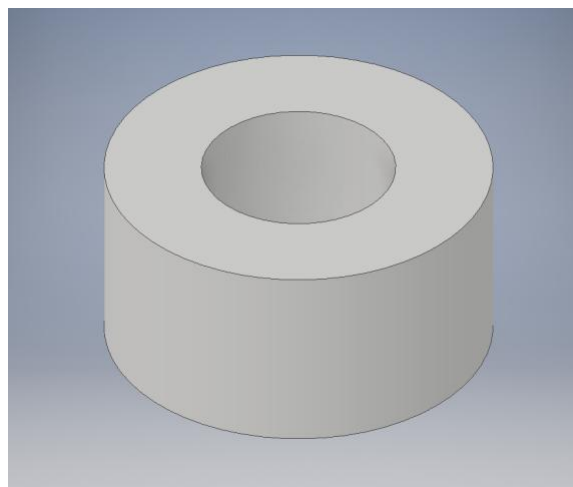
Front View



Top View



Isometric View



Code

```
//Created By: Aditya Bangalore
```

```
//Date: March 25, 2020
```

```
//This code was uploaded to the Arduino Uno for running the obstacle avoiding robot.
```

```

#include <AFMotor.h> //Imports library for running the motor shield

#define enablepin 5 //Defines the enable pin on the KY-032 as pin 5 on Arduino
#define IR 4        //Defines the output pin on the KY-032 as pin 4 on Arduino.
                    //Rename the pin to "IR".

AF_DCMotor backright(1); //Assigns the back right motor to the M1 pins on the motor shield
AF_DCMotor frontright(2); //Assigns the front right motor to the M2 pins on the motor shield
AF_DCMotor frontleft(3); //Assigns the front left motor to the M3 pins on the motor shield
AF_DCMotor backleft(4); //Assigns the back left motor to the M4 pins on the motor shield

void Stop() {          //Creates a Stop() function that stops the robot's wheels
    backright.run(RELEASE);
    backleft.run(RELEASE);
    frontright.run(RELEASE);
    frontleft.run(RELEASE);
}

void Forward() { //Creates a Forward() function that drives the robot forward
    backright.run(FORWARD);
    backleft.run(FORWARD);
    frontright.run(FORWARD);
    frontleft.run(FORWARD);
}

void Backward() { //Creates a Backward() function that drives the robot backward
    backright.run(BACKWARD);
    backleft.run(BACKWARD);
    frontright.run(BACKWARD);
    frontleft.run(BACKWARD);
}

void Right() { //Creates a Right() function that turns the robot to the right
    backright.run(BACKWARD);
    frontright.run(BACKWARD);
    backleft.run(FORWARD);
    frontleft.run(FORWARD);
}

void Left() { //Creates a Left() function that turns the robot to the left
    backright.run(FORWARD);
    frontright.run(FORWARD);
    backleft.run(BACKWARD);
    frontleft.run(BACKWARD);
}

```

```

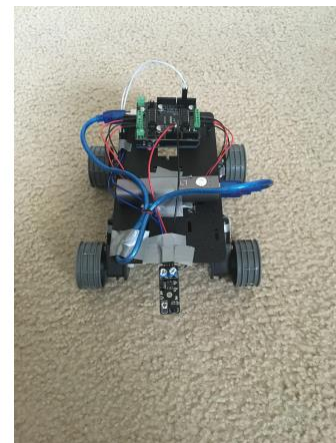
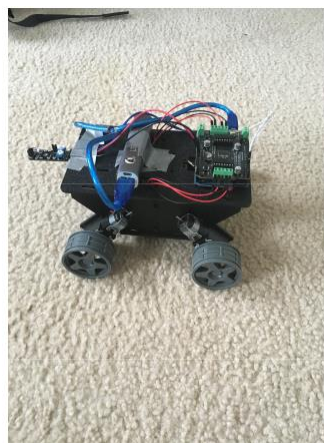
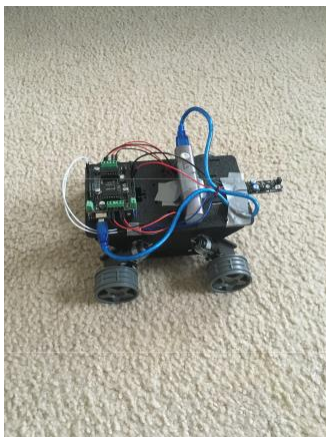
}

void setup() { //Setup function: sets the speed of each motor to the maximum value
  backright.setSpeed(255);
  frontright.setSpeed(255);
  frontleft.setSpeed(255);
  frontleft.setSpeed(255);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(enablepin, HIGH); //Send a HIGH signal to the enable pin of the KY-032
  if(digitalRead(IR) == LOW) { //If the reading on the IR pin is low...
    Serial.println("Object Detected"); //Print to the serial monitor that there was an object
    Stop(); //Stop the car
    Right(); //Turn right
    delay(2000); //Keep turning for two seconds
    if(digitalRead(IR) == LOW) { //If the reading on the IR pin is still low...
      Stop(); //Stop the car
      Left(); //Turn left
      delay(4000); //Keep turning for four seconds
    }
  }
  else { //If the reading on the IR pin is high or something else...
    Serial.println("No Object"); //Print to the serial monitor that there was no object detected
    Forward(); //Continue moving forward
  }
}

```

Pictures of Finished Design



SEVEN SEGMENT GAME

Objective/Description

Create a game involving a seven segment display in which a user has to press a button when a key number appears on the display (while the key number is not on the display, the display shows random numbers). If they press the button at the same time that the key number appears on the display, the user earns a point. If they press the button but the key number is not on the display, they lose a point. Points are kept by lighting LEDs. Once the user reaches three points (3 lit LEDs) they win.

Materials

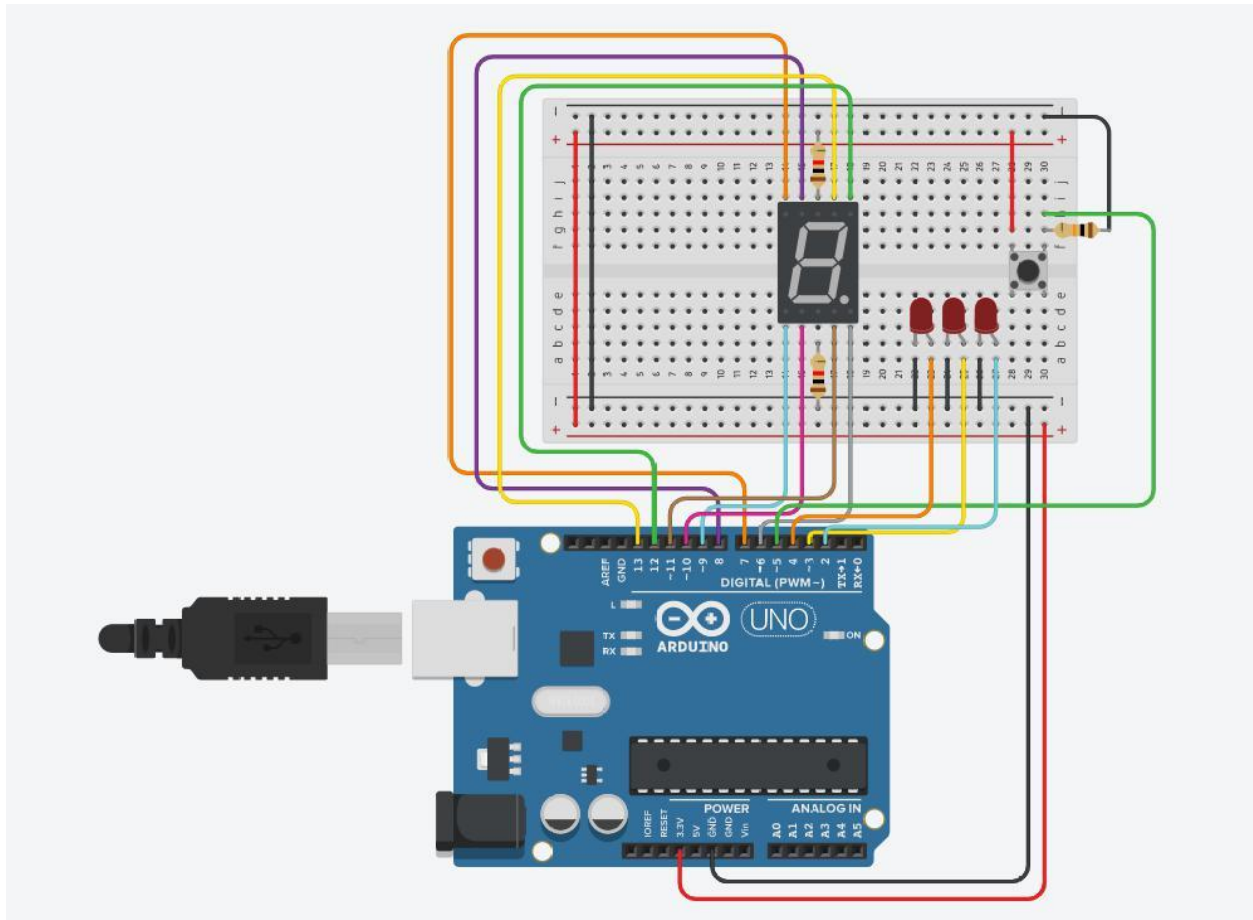
Arduino Uno
Seven Segment Display Module
3 LEDs (any color)
Button Module
1 k Ω Resistors (x2)
10 k Ω Resistor (x1)
Breadboard
Jumper Wires (M to M)

Assembly

1. Wire a circuit that matches the wiring diagram below.
2. Upload the included code onto the Arduino.

(Continued on Next Page)

Wiring Diagram



Code

//Created By: Aditya Bangalore

//Date: March 28, 2020

//This code was uploaded to the Arduino Uno to play the seven segment display game.

```
#include <Metro.h> //Include a timing library that will allow for the button pin to
read continuously while the seven segment display shows numbers.
```

```
#define A 13 //Assign appropriate pins to variables for each purpose.
```

```
#define B 12
```

```
#define C 11
```

```
#define D 10
```

```
#define E 9
```

```
#define F 8
```

```
#define G 7
```

```
#define DP 6
```

```
#define button 5
#define LED1 4
#define LED2 3
#define LED3 2
```

```
int present; //Create an integer that will change based on which number is displayed on the
seven segment.
```

```
Metro buttonAndCheckMetro = Metro(1); //Set the interval for reading and checking the
input from the button as 1 ms.
```

```
Metro gameMetro = Metro(450); //Set the interval for changing the number on the seven
segment.
```

```
void off() { //Create a function that turns off the display. Gives the variable present a value of -1.
```

```
    present = -1;
    digitalWrite(A, LOW);
    digitalWrite(B, LOW);
    digitalWrite(C, LOW);
    digitalWrite(D, LOW);
    digitalWrite(E, LOW);
    digitalWrite(F, LOW);
    digitalWrite(G, LOW);
    present = -1;
}
```

```
void zero() {
```

```
    present = 0; //Create a function that shows a 0 on the display. Gives the variable present a value
of 0.
```

```
    digitalWrite(A, HIGH);
    digitalWrite(B, HIGH);
    digitalWrite(C, HIGH);
    digitalWrite(D, HIGH);
    digitalWrite(E, HIGH);
    digitalWrite(F, HIGH);
    digitalWrite(G, LOW);
    present = 0;
}
```

```
bool checkZero() { //Creates a function to check if 0 is on the display. 0 is the key number.
```

```
    if(present == 0) {
        return true;
    }
    else {
        return false;
    }
}
```

```
}
```

```
void one() { //Creates a function to display a 1 on the display. Gives present val of 1
  present = 1;
  digitalWrite(A, LOW);
  digitalWrite(B, HIGH);
  digitalWrite(C, HIGH);
  digitalWrite(D, LOW);
  digitalWrite(E, LOW);
  digitalWrite(F, LOW);
  digitalWrite(G, LOW);
  present = 1;
}
```

```
void two() { //Creates a function to display a 2 on the display. Gives present val of 2
  present = 2;
  digitalWrite(A, HIGH);
  digitalWrite(B, HIGH);
  digitalWrite(C, LOW);
  digitalWrite(D, HIGH);
  digitalWrite(E, HIGH);
  digitalWrite(F, LOW);
  digitalWrite(G, HIGH);
  present = 2;
}
```

```
void three() { //Creates a function to display a 3 on the display. Gives present val of 3
  present = 3;
  digitalWrite(A, HIGH);
  digitalWrite(B, HIGH);
  digitalWrite(C, HIGH);
  digitalWrite(D, HIGH);
  digitalWrite(E, LOW);
  digitalWrite(F, LOW);
  digitalWrite(G, HIGH);
  present = 3;
}
```

```
void four() { //Creates a function to display a 4 on the display. Gives present val of
  4 present = 4;
  digitalWrite(A, LOW);
  digitalWrite(B, HIGH);
  digitalWrite(C, HIGH);
  digitalWrite(D, LOW);
  digitalWrite(E, LOW);
```

```
digitalWrite(F, HIGH);  
digitalWrite(G, HIGH);  
present = 4;  
}
```

```
void five() { //Creates a function to display a 5 on the display. Gives present val of  
5 present = 5;  
digitalWrite(A, HIGH);  
digitalWrite(B, LOW);  
digitalWrite(C, HIGH);  
digitalWrite(D, HIGH);  
digitalWrite(E, LOW);  
digitalWrite(F, HIGH);  
digitalWrite(G, HIGH);  
present = 5;  
}
```

```
void six() { //Creates a function to display a 6 on the display. Gives present val of 6  
present = 6;  
digitalWrite(A, LOW);  
digitalWrite(B, LOW);  
digitalWrite(C, HIGH);  
digitalWrite(D, HIGH);  
digitalWrite(E, HIGH);  
digitalWrite(F, HIGH);  
digitalWrite(G, HIGH);  
present = 6;  
}
```

```
void seven() { //Creates a function to display a 7 on the display. Gives present val of 7  
present = 7;  
digitalWrite(A, HIGH);  
digitalWrite(B, HIGH);  
digitalWrite(C, HIGH);  
digitalWrite(D, LOW);  
digitalWrite(E, LOW);  
digitalWrite(F, LOW);  
digitalWrite(G, LOW);  
present = 7;  
}
```

```
void eight() { //Creates a function to display an 8 on the display. Gives present val of 8  
present = 8;  
digitalWrite(A, HIGH);  
digitalWrite(B, HIGH);
```

```

digitalWrite(C, HIGH);
digitalWrite(D, HIGH);
digitalWrite(E, HIGH);
digitalWrite(F, HIGH);
digitalWrite(G, HIGH);
present = 8;
}

```

```

void nine() { //Creates a function to display a 9 on the display. Gives present val of
  9 present = 9;
  digitalWrite(A, HIGH);
  digitalWrite(B, HIGH);
  digitalWrite(C, HIGH);
  digitalWrite(D, LOW);
  digitalWrite(E, LOW);
  digitalWrite(F, HIGH);
  digitalWrite(G, HIGH);
  present = 9;
}

```

```

void earnPoint() { //Creates an animation on the display to show the user they earned a point.
  for(int a = 1; a <= 3; a += 1) {
    digitalWrite(A, HIGH);
    delay(50);
    digitalWrite(A, LOW);
    delay(50);
    digitalWrite(B, HIGH);
    delay(50);
    digitalWrite(B, LOW);
    delay(50);
    digitalWrite(C, HIGH);
    delay(50);
    digitalWrite(C, LOW);
    delay(50);
    digitalWrite(D, HIGH);
    delay(50);
    digitalWrite(D, LOW);
    delay(50);
    digitalWrite(E, HIGH);
    delay(50);
    digitalWrite(E, LOW);
    delay(50);
    digitalWrite(F, HIGH);
    delay(50);
    digitalWrite(F, LOW);
  }
}

```

```

    delay(50);
    digitalWrite(G, LOW);
    delay(50);
  }
}

void losePoint() { //Creates an animation on the display to show the user that they lost a point.
  for(int a = 1; a <= 3; a += 1) {
    digitalWrite(A, HIGH);
    delay(100);
    digitalWrite(A, LOW);
    digitalWrite(G, HIGH);
    delay(100);
    digitalWrite(G, LOW);
    digitalWrite(D, HIGH);
    delay(100);
    digitalWrite(D, LOW);
  }
}

void firstPoint() { //Create a function to turn the first LED on.
  digitalWrite(LED1, HIGH);
}

void secondPoint() { //Create a function to turn the second LED on.
  digitalWrite(LED2, HIGH);
}

void thirdPoint() { //Create a function to turn the third LED on.
  digitalWrite(LED3, HIGH);
}

void win() { //Blink all three LEDs when the user wins.
  for(int b = 1; b <= 5; b += 1) { digitalWrite(LED1,
    LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);
    delay(500);
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, HIGH);
    delay(500);
  }
  digitalWrite(LED1, LOW);
  digitalWrite(LED2, LOW);

```

```

    digitalWrite(LED3, LOW);
}

void game() { //Displays random numbers from 0 to 9.
    off();
    delay(10);
    int num = random(0,10);
    switch(num) {
        case 0:
            zero();
            break;
        case 1:
            one();
            break;
        case 2:
            two();
            break;
        case 3:
            three();
            break;
        case 4:
            four();
            break;
        case 5:
            five();
            break;
        case 6:
            six();
            break;
        case 7:
            seven();
            break;
        case 8:
            eight();
            break;
        case 9:
            nine();
            break;
    }
}

void setup() { //Define the function of each pin (output or input). Initialize other components.
    pinMode(A, OUTPUT);
    pinMode(B, OUTPUT);
}

```

```

pinMode(C, OUTPUT);
pinMode(D, OUTPUT);
pinMode(E, OUTPUT);
pinMode(F, OUTPUT);
pinMode(G, OUTPUT);
pinMode(button, INPUT);
pinMode(LED1, OUTPUT);
pinMode(LED2, OUTPUT);
pinMode(LED3, OUTPUT);
randomSeed(analogRead(0));
Serial.begin(9600);
}

void loop() {
  bool btnrdng = digitalRead(button); //tells if the button is on or off.
  bool check0 = checkZero(); //tells if 0 is on the display.

  if(buttonAndCheckMetro.check()) { //create a thread to run every 1 ms.
    Serial.println(btnrdng); //print the button reading.
    if(check0 && btnrdng) { //If 0 is on the display and the button is pressed,
      off(); //turn off display.
      delay(50); //wait 50 ms.
      earnPoint(); //display animation for earning a point.
      delay(100); //wait 100 ms.
      for(int a = 1; a <= 3; a += 1) {
        switch(a) {
          case 1: //when a is 1,
            if(digitalRead(LED1) == LOW) { //if LED 1 is
              low, firstPoint(); //turn it on.
              break;
            }
          else {
            firstPoint(); //if LED 1 is high,
            a = 2; //change a to 2.
          }
          case 2: //if a is 2,
            if(digitalRead(LED2) == LOW) { //if LED 2 is
              low, firstPoint(); //turn LED 1 on.
              secondPoint(); //turn LED 2 on.
              break;
            }
          else { //if LED 2 is high,
            firstPoint(); //turn LED 1 on.
            secondPoint(); //turn LED 2 on.
            a = 3; //change a to 3.
          }
        }
      }
    }
  }
}

```



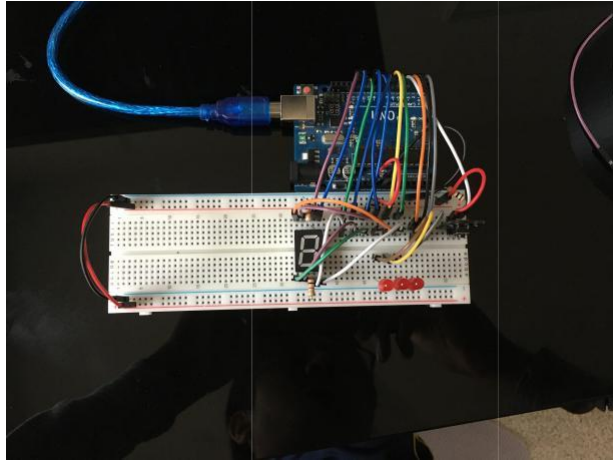
```

    }
    case 3: //if a is 3,
    if(digitalRead(LED3) == LOW) { //if LED 3 is
        low, firstPoint(); //turn LED 1 on.
        secondPoint(); //turn LED 2 on.
        thirdPoint(); //turn LED 3 on.
        delay(500); //wait 500 ms.
        win(); //blink LEDs three times. The user won!
        break;
    }
    }
    break;
}
}
else if (check0 == false && btnrdng){ //if the button reading is high but 0 is not
    displayed, off(); //turn off the display.
    delay(50); //wait 50 ms.
    losePoint(); //show animation that user lost.
    off(); //turn off display.
    delay(50); //wait 50 ms.
    if(digitalRead(LED1) && digitalRead(LED2) == LOW && digitalRead(LED3) == LOW)
{ //if LED 1 is on, but LED 2 and 3 are off,
    digitalWrite(LED1, LOW); //turn off LED 1.
    }
    else if(digitalRead(LED1) && digitalRead(LED2) && digitalRead(LED3) == LOW) { //if
LED 1 and 2 are on, but LED 3 is off,
    digitalWrite(LED2, LOW); //turn off LED 2.
    }
    }
}
}
if(gameMetro.check()) { //every 450 ms,
    game(); //run the game() function (display a random number).
    }
}
}

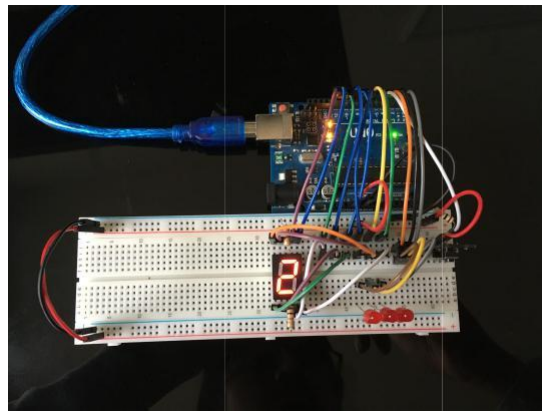
```

Pictures of Finished Design

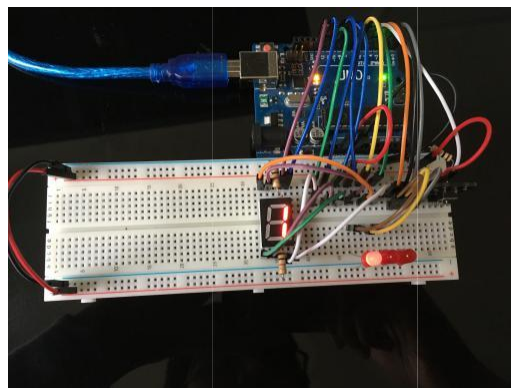
(Pictures on Next Page)



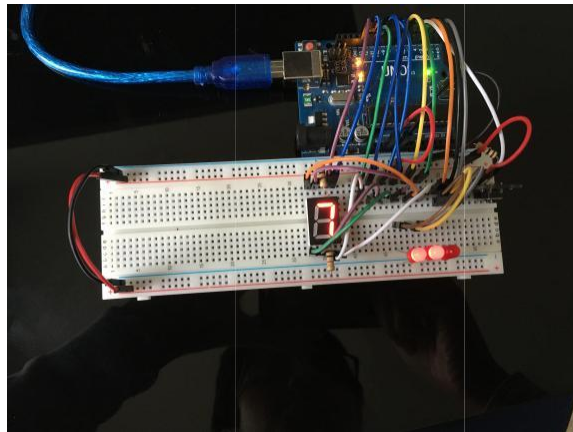
Caption: Game is off.



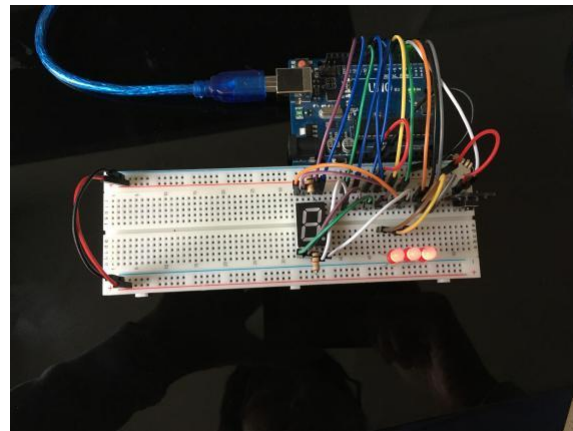
Caption: Game is on, but the user has not earned any points. Display is showing random numbers.



Caption: The user has earned one point.



Caption: The user has earned two points.



Caption: The user has won and earned all three points. The display now turns off.

AUTOMATIC NIGHT LIGHT

Objective/Description

Create a self-timing night light that will automatically turn on at 9 PM and automatically turn off at 1 AM. At any point, the night light is able to toggle between on and off, but regardless of which toggle state it is in, it should automatically turn on at 9 PM and turn off at 1 AM. After the automatic process, the night light should be able to be toggled again.

Materials

Raspberry Pi 3 Model B
5 VDC Relay Module
Capacitive Touch Sensor Module
6 ft Extension Cord
Night Light/Desk Lamp
Wire Caps
Jumper Wires (M to F)
Breadboard
Exacto Knife/Scissors
Wire Stripper
Wire Cutter
Screwdriver
Extra Extension Wires

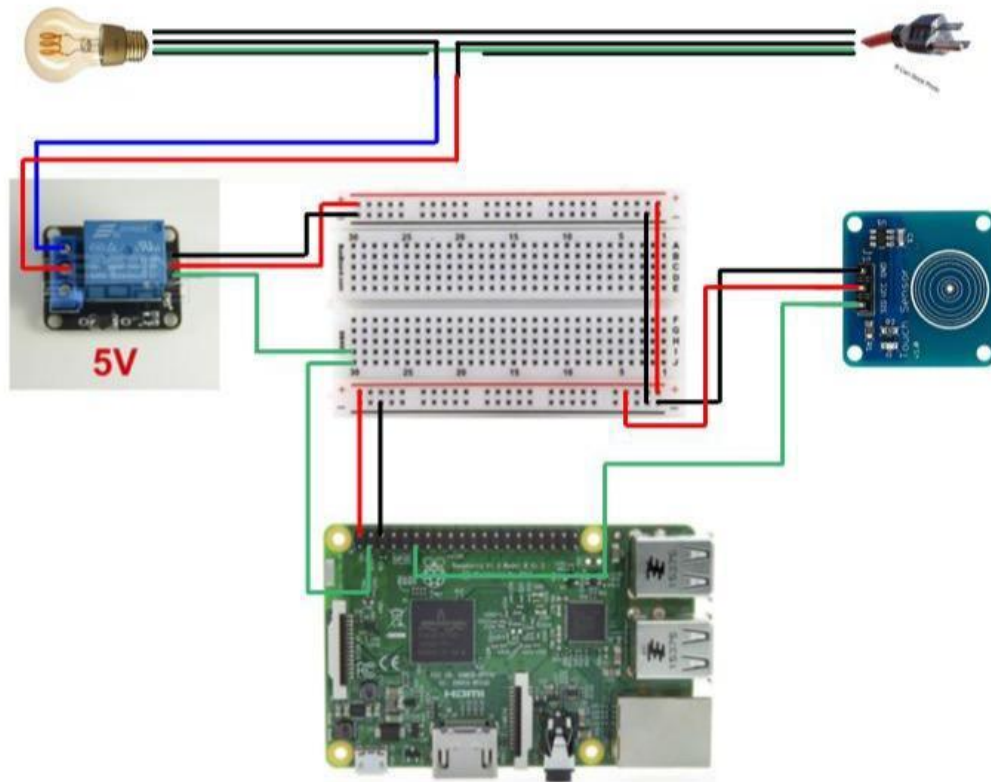
Assembly

1. WARNING: Make sure that the extension cord and night light are UNPLUGGED during the assembly, unless otherwise specified.

2. Use the knife/scissors to slit the extension cord's outermost layer, revealing the three wires inside (green - ground, white - neutral, and black - hot).
3. Cut the black wire with the wire cutters.
4. Strip each cut end of the black wire by about half an inch.
5. Use the wire caps to join each cut end of the black wire to a separate extension wire.
6. Using the screwdriver, plug one of the extension wires (each of which should now be capped with the black wire) into the COM port of the relay. Plug the other extension wire into the NO port of the relay.
7. Wire up a circuit as shown in the Wiring Diagram section.
8. Create a Python file on the Raspberry Pi to copy and paste the code from the Code section into.
9. Plug the night light's cord into the extension cord socket, plug the extension cord into the wall outlet, and run the code as root on the Raspberry Pi. DO NOT TOUCH THE EXTENSION WIRES/EXTENSION CORD WHILE THE EXTENSION CORD IS PLUGGED INTO THE WALL. If you need to change wiring/code, make sure to UNPLUG THE EXTENSION CORD FROM THE WALL first.

Wiring Diagram

(Wiring Diagram on Next Page)



Code

#Created By: Aditya Bangalore

#Date: July 4, 2020

#This code was uploaded to the Arduino Uno to run the automatic night light.

```
import RPi.GPIO as gpio #import the Raspberry Pi GPIO library
```

```
import time #import the time library
```

```
light = 3          #declare a variable for the light/relay (pin 3)
```

```
ts = 11           #declare a variable for the touch sensor (pin 11)
```

```
gpio.setmode(gpio.BOARD) #tell the RPi to read the pins in the BOARD mode
```

```
gpio.setup(light, gpio.OUT) #declare the light/relay as an output
```

```
gpio.setup(ts, gpio.IN) #declare the touch sensor as an input
```

```
def buttonpush():      #declare a function responsible for toggling the touch sensor
```

```
    read_old = True    #declare an "old reading" as True #set the initial light state
```

```
    light_state = False as False
```

```
    while True:
```

```

t = time.asctime()    #store data about the local time in a variable called t
hour = int(t[11:13])  #set the hour of the time as the 11th through 13th char of the t var
minute = int(t[14:16]) #set the minute of the time as the 14th through 16th char of the t var
second = int(t[17:19]) #set the second of the time as the 17th through 19th char of the t var

read_new = gpio.input(ts) #tell the RPi to read the sensor in a var called read_new

    if read_old == False and read_new == True: ““if the old reading is False and the new
reading is True (the sensor has been pressed)...””
        if light_state == False:            #if the light is off...
            gpio.output(light, True)         #turn the light on
            light_state = True               #set the light_state variable to True
        else:                               #otherwise...
            gpio.output(light, False)        #turn the light off
            light_state = False              #set the light_state variable to False
    elif hour == 1 and minute < 1 and second < 1: #if it just passed 1 AM...
        break                               #break out of the loop
    elif hour == 21 and minute < 1 and second < 1: #if it just became 9 PM...
        break                               #break out of the loop
    read_old = read_new                     #set the old reading as the new reading for the new loop

def nightlight():                #declare a function responsible for automating the light
    while True:
        t = time.asctime()        #declare the necessary time variables like before
        hour = int(t[11:13])
        minute = int(t[14:16])
        second = int(t[17:19])

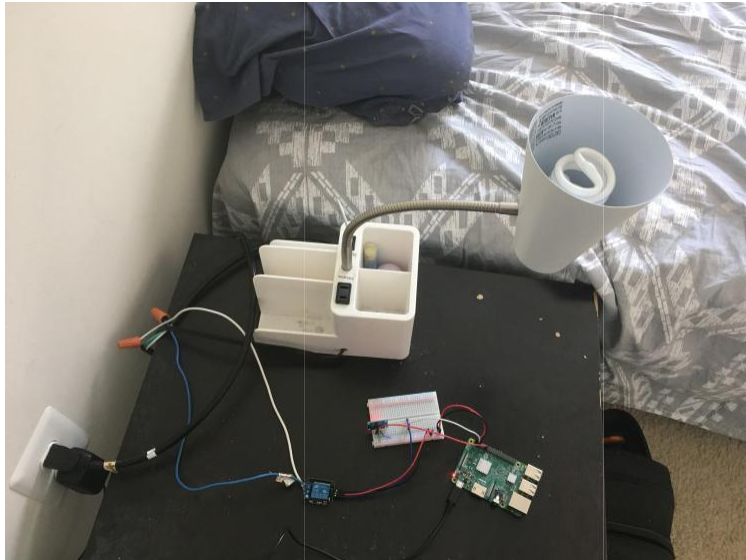
        if hour == 1 and minute < 1 and second < 1: #if it just passed 1 AM...
            gpio.output(light, False)              #turn the light off
        elif hour == 21 and minute < 1 and second < 1: #if it just became 9 PM...
            gpio.output(light, True)               #turn the light on
        else:                                       #otherwise...
            buttonpush()                          #do the commands in the buttonpush() function

nightlight()    #run the nightlight() function

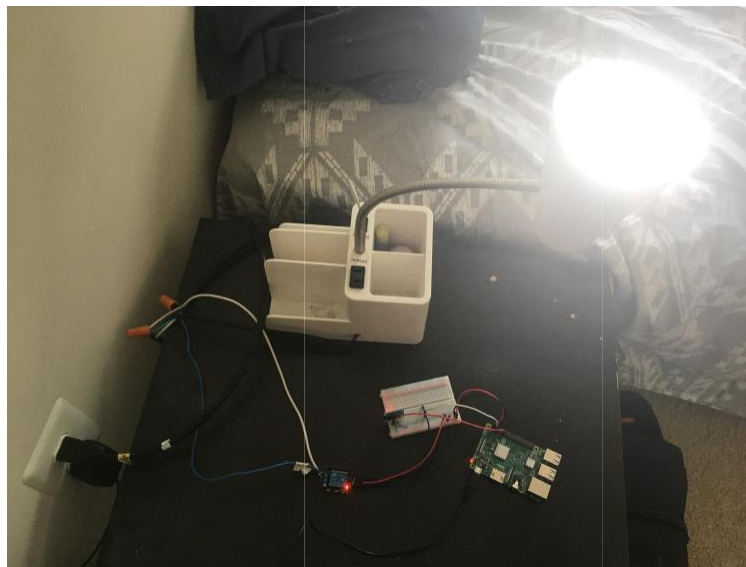
```

Pictures of Finished Design

(Pictures and Captions on the Next Page)



Caption: Default state, sensor not toggled and light is off. If the light had been on and it just passed 1 AM, this state would be achieved.



Caption: Secondary state, sensor is toggled and light is on. If the light had been off and it just turned 9 PM, this state would be achieved.

SECURITY SYSTEM PANEL

Objective/Description

Design a makeshift security system panel using an LCD display. The system should notify the user of the status of the system (Armed/Unarmed), allow the user to use a code to change the status of the system, and allow the user to change the code. While the user is changing the code, the system should prompt the user to enter the old code and the new code. If the user incorrectly confirms the old code, the system will return back to the main menu. If a user incorrectly inputs a code five times in a row (at the main menu screen while trying to change system status), the system should lock for 60 seconds.

Materials

Arduino Uno
Velleman LCD & Keypad Shield for Arduino

Assembly

1. Plug the Velleman LCD & Keypad Shield for Arduino onto the Arduino, according to product instructions.
2. Upload code from “Code” to Arduino.

Code

//Created By: Aditya Bangalore

//Date: July 8, 2020

#include <LiquidCrystal.h>

//Import LiquidCrystal library

LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
parameters

//Create LiquidCrystal object lcd with necessary

#define None 0

//Assign pin numbers to keypad function

```

#define Right 1
#define Up 2
#define Down 3
#define Left 4
#define Select 5

String user_code = "";                                //Create instance variables for the codes and system
status
String master_code = "URDLLDRU";
String system_status = "Unarmed";
String asterisks = "";

byte count = 0;                                       //Create instance variables that track certain variable states
byte old_count;
byte wrongpwd_counter = 0;
int correctpwd_counter = 0;
int changeCodeActive = 0;

int readButtons() {                                  //Create function that takes analog input from buttons and
interprets them
    int btnRead = analogRead(0);
    if(btnRead == 0) {
        return Right;
    }
    if(btnRead == 99) {
        return Up;
    }
    if(btnRead == 258) {
        return Down;
    }
    if(btnRead == 411) {
        return Left;
    }
    if(btnRead == 640) {
        return Select;
    }
    else {
        return None;
    }
}

void homeScreen() {                                  //Create home screen defaults
    lcd.setCursor(0,0);
    lcd.print("Code:");
    lcd.setCursor(0,1);

```

```

lcd.print("Status:");
if(system_status == "Unarmed") {
    system_status = "Unarmed";
    lcd.setCursor(11,1);
    lcd.print("  ");
    delay(100);
    lcd.setCursor(9,1);
    lcd.print(system_status);
}
else if(system_status == "Armed") {
    system_status = "Armed";
    lcd.setCursor(9,1);
    lcd.print("  ");
    delay(100);
    lcd.setCursor(11,1);
    lcd.print(system_status);
}
}

void typeCode(int col, int row) {           //Function that allows user to type a code
String uc = user_code.substring(user_code.length() - 1); //Codes are typed w/ *, but stored as
vals int reading = readButtons();
lcd.setCursor(col,row);
switch(reading) {
    case None:
        break;
    case Right:
        if(uc == "R") {
            if(old_count != count) {
                delay(300);
                if(readButtons() == Right) {
                    user_code = user_code + "R";
                    asterisks = asterisks + "*";
                    lcd.print(asterisks);
                    break;
                }
            }
            else {
                break;
            }
        }
    }
}
if(uc != "R") {
    user_code = user_code + "R";
    asterisks = asterisks + "*";
    lcd.print(asterisks);
}

```

```

    old_count = count;
    count += 1;
    break;
}
case Up:
    if(uc == "U") {
        if(old_count != count) {
            delay(300);
            if(readButtons() == Up) {
                user_code = user_code + "U";
                asterisks = asterisks + "*";
                lcd.print(asterisks);
                break;
            }
            else {
                break;
            }
        }
    }
    if(uc != "U") {
        user_code = user_code + "U";
        asterisks = asterisks + "*";
        lcd.print(asterisks);
        old_count = count;
        count += 1;
        break;
    }
case Down:
    if(uc == "D") {
        if(old_count != count) {
            delay(300);
            if(readButtons() == Down) {
                user_code = user_code + "D";
                asterisks = asterisks + "*";
                lcd.print(asterisks);
                break;
            }
            else {
                break;
            }
        }
    }
    if(uc != "D") {
        user_code = user_code + "D";
        asterisks = asterisks + "*";

```

```

    lcd.print(asterisks);
    old_count = count;
    count += 1;
    break;
}
case Left:
    if(uc == "L") {
        if(old_count != count) {
            delay(300);
            if(readButtons() == Left) {
                user_code = user_code + "L";
                asterisks = asterisks + "*";
                lcd.print(asterisks);
                break;
            }
        }
        else {
            break;
        }
    }
    if(uc != "L") {
        user_code = user_code + "L";
        asterisks = asterisks + "*";
        lcd.print(asterisks);
        old_count = count;
        count += 1;
        break;
    }
case Select:
    changeCode();
    break;
}
}

void checkCode() { //Check if the input code matches master code.
    if(user_code.length() == 8 && user_code != master_code) {
        user_code = "";
        asterisks = "";
        wrongpwd_counter += 1;
        delay(200);
        if(wrongpwd_counter < 5) {
            lcd.setCursor(0,0);
            lcd.print("      ");
            lcd.setCursor(0,1);
            lcd.print("      ");

```

```

    lcd.setCursor(8,0);
    lcd.print("    ");
    for(byte n = 0; n <= 1; n++) {
        lcd.setCursor(1,0);
        lcd.print("Incorrect Code");
        delay(500);
        lcd.setCursor(1,0);
        lcd.print("    ");
        delay(500);
    }
    homeScreen();
}
else if(wrongpwd_counter == 5) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("System Locked");
    lcd.setCursor(0,1);
    lcd.print("Time Left:");
    for(int n = 60; n >= 0; n--) {
        if(n >= 10) {
            lcd.setCursor(14,1);
            lcd.print(n);
            delay(1000);
        }
        else if(n < 10 && n > 0) {
            lcd.setCursor(14,1);
            lcd.print("0");
            lcd.setCursor(15,1);
            lcd.print(n);
            delay(1000);
        }
        else if(n == 0) {
            lcd.clear();
            wrongpwd_counter = 0;
            homeScreen();
        }
    }
}
}
if(user_code.length() == 8 && user_code == master_code) {
    delay(200);
    user_code = "";
    asterisks = "";
    wrongpwd_counter = 0;
    correctpwd_counter += 1;
}

```

```

if(correctpwd_counter > 0) {
  lcd.setCursor(8,0);
  lcd.print(" ");
  if(system_status == "Unarmed") {
    system_status = "Armed";
    lcd.setCursor(9,1);
    lcd.print(" ");
    delay(100);
    lcd.setCursor(11,1);
    lcd.print(system_status);
  }
  else if(system_status == "Armed") {
    system_status = "Unarmed";
    lcd.setCursor(11,1);
    lcd.print(" ");
    delay(100);
    lcd.setCursor(9,1);
    lcd.print(system_status);
  }
}
}
else if(user_code.length() > 8) {
  user_code = "";
  asterisks = "";
  lcd.print(asterisks);
}
}

void changeCode() {                                     //Provide method for user to change master code.
  String old_code = master_code;
  user_code = "";
  asterisks = "";
  wrongpwd_counter = 0;
  delay(300);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Current Code:");
  lcd.setCursor(0,1);
  while(HIGH) {
    typeCode(0,1);
    if(user_code.length() == 8 && user_code == master_code) {
      delay(200);
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("New Code:");
    }
  }
}

```

```

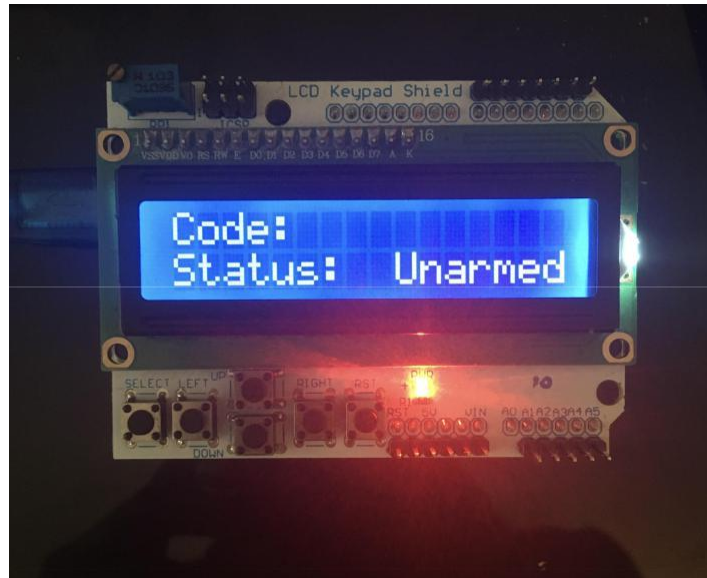
user_code = "";
asterisks = "";
while(HIGH) {
  if(user_code.length() == 8) {
    delay(200);
    master_code = user_code;
    lcd.clear();
    homeScreen();
    correctpwd_counter = -1;
    break;
  }
  else if(user_code.length() < 8) {
    typeCode(0,1);
  }
}
break;
}
else if(user_code.length() == 8 && user_code != master_code) {
  delay(200);
  lcd.clear();
  user_code = "";
  asterisks = "";
  for(byte n = 0; n <= 1; n++) {
    lcd.setCursor(1,0);
    lcd.print("Incorrect Code");
    delay(500);
    lcd.clear();
    delay(500);
  }
  homeScreen();
  break;
}
}
}

void setup() {                                     //Start the program
  lcd.begin(16,2);
  homeScreen();
  Serial.begin(9600);
  analogWrite(10, 100);
}

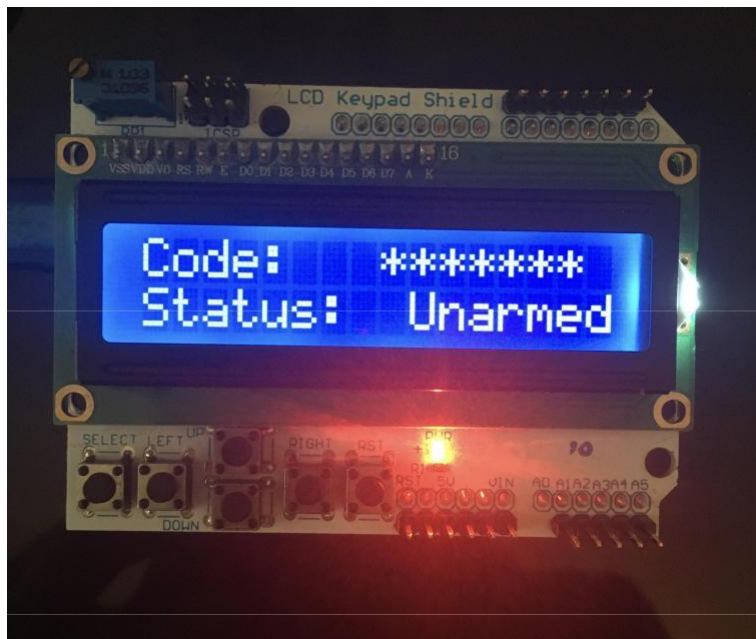
void loop() {                                     //Run necessary functions to allow program to work.
  typeCode(8,0);
  checkCode();  }

```

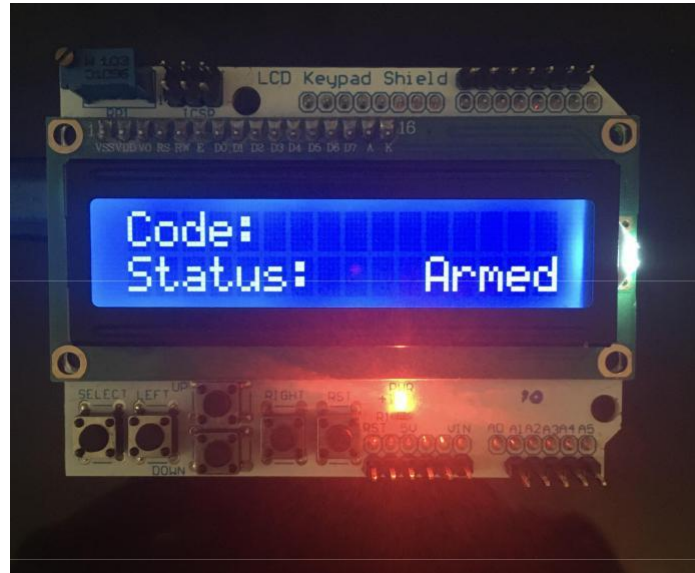

Pictures of Finished Design



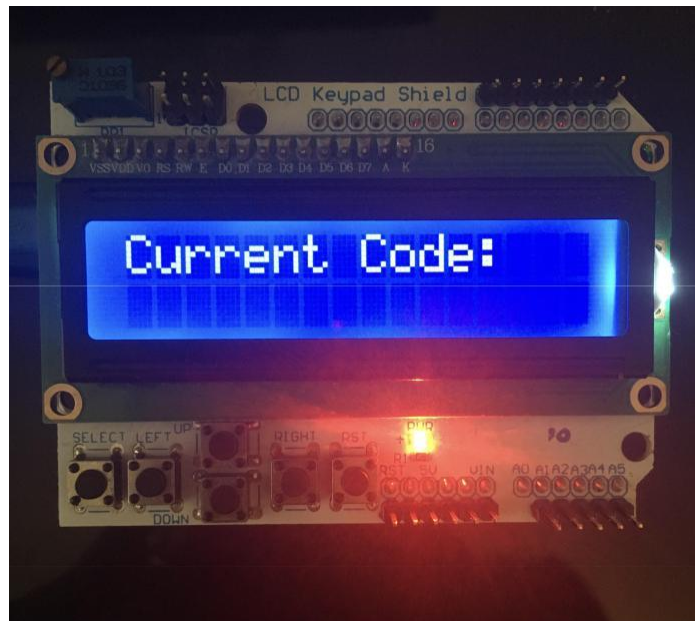
Caption: Default state (home screen) of security panel. System is unarmed



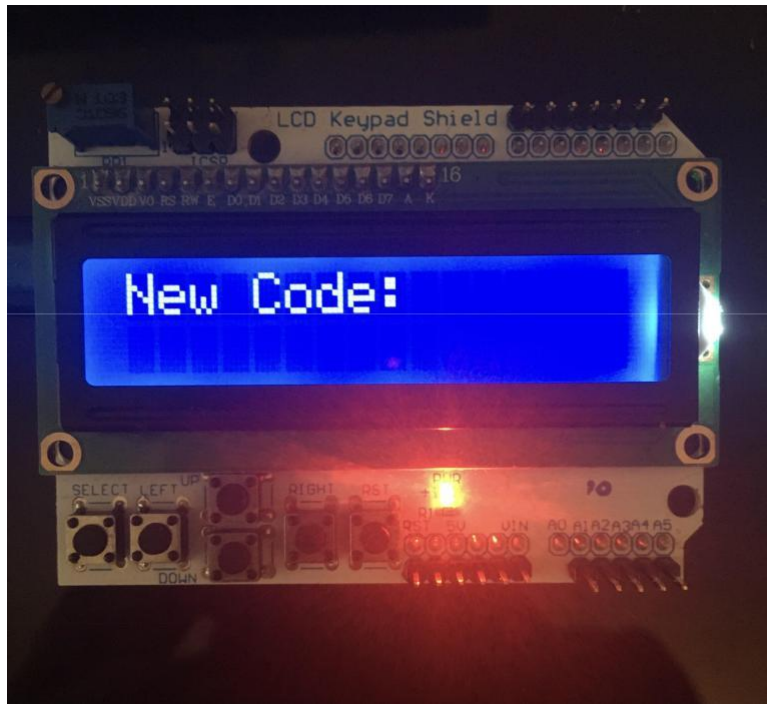
Caption: Security panel with code being typed. Code is displayed as asterisks but stored as a combination of values which are converted to Strings



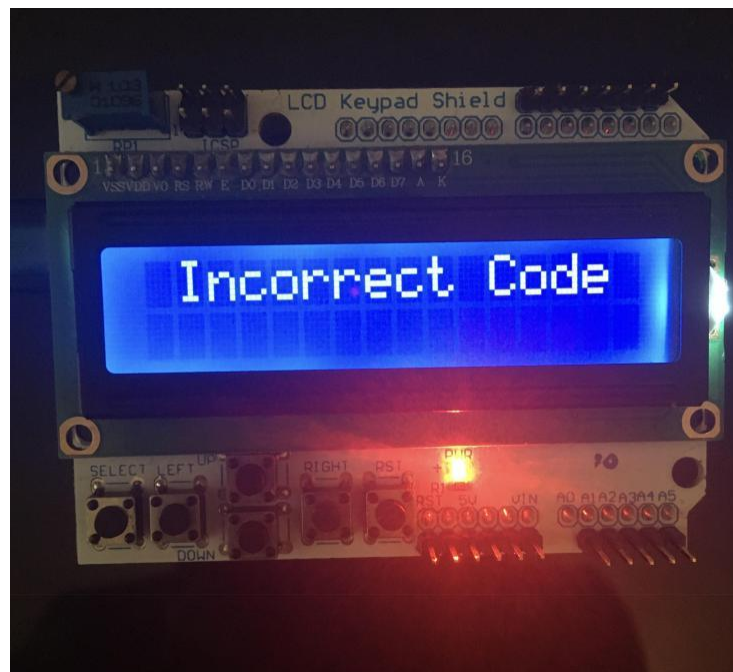
Caption: Home screen with Armed status of system



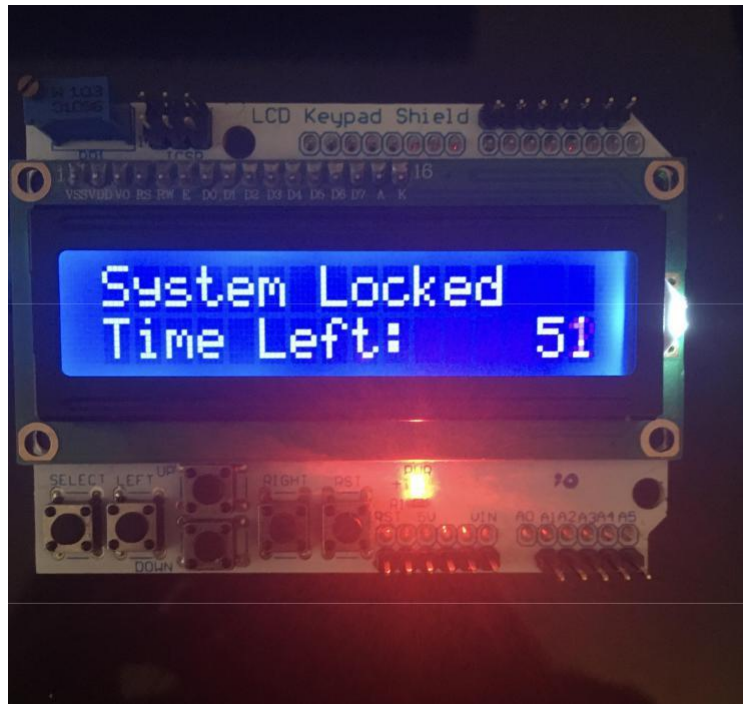
Caption: Prompt for users while they are changing the code; asks for confirmation of the old code



Caption: Prompt for users while they are changing the code; asks user for the new master code



Caption: Alert user that they have entered an incorrect code. Screen flashes three times.



Caption: Locked screen after user incorrectly enters code five times in a row. System counts down from 60 seconds. In the photo, the countdown is at 51 seconds.

REFERENCES & CITATIONS (APA)

References below contain resources used for research about project components, hardware and/or software.

- Backus, M. (Director). (2016, April 29). *7-Segment Display* [Video file]. Retrieved March 28, 2016, from <https://www.youtube.com/watch?v=kEVdZ1i4Uok>
- byte sized. (2017, May 14). DIY Smart Switch - Part 1 how to use a relay. Retrieved July 4, 2020, from <https://www.youtube.com/watch?v=5NxVmg8ZFec>
- DroneBot Workshop (Director). (2018, March 19). *Using LCD Displays with Arduino* [Video file]. Retrieved July 8, 2020, from <https://www.youtube.com/watch?v=wEbGhYjn4QI>
- ExplainingComputers (Director). (2014, July 17). *Raspberry Pi Robotics #1: GPIO Control* [Video file]. Retrieved July 4, 2020, from <https://www.youtube.com/watch?v=41IO4Qe5Jzw>
- ItKindaWorks (Director). (2016, July 22). *How to multi-task on an Arduino* [Video file]. Retrieved March 28, 2020, from https://www.youtube.com/watch?v=zhWV_D_9OCY
- logMaker260 (Director). (2015, October 16). *KY-032 Infrared obstacle sensors* [Video file]. Retrieved March 25, 2020, from <https://www.youtube.com/watch?v=QWCdcwxF3RE>
- Shovic, J. C., & Simpson, A. (2019). *Python all-in-one*. Hoboken, NJ: John Wiley & Sons.