// new arduino user system

// includes

#include <Adafruit\_NeoPixel.h>

// constants

#define PIN 6 // LED on digital pin 6

#define NUMLEDS 1 // Use total of 1 LED

#define FALSE 0

#define TRUE 1

#define BuzzerPin 13

// Keypad Things

char ourCode[4] = "1974"; // Set the required PIN code.

int index = 0; // Keeps track of the position of the code entered.

char userEnteredPin[4];

// Ultrasonic Sensor Things

const int trigPin = 9;

const int echoPin = 10;

int uss\_value;

// RGB LED Things

Adafruit\_NeoPixel rgb\_led = Adafruit\_NeoPixel(NUMLEDS, PIN, NEO\_GRB + NEO\_KHZ800);

// define variables

long duration;

bool alarm\_state;

// serial.printline

void setup() {

Serial.begin(9600);

// pin modes

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

pinMode(echoPin, INPUT); // Sets the echoPin as an Input

pinMode(BuzzerPin, OUTPUT);

// initialize the rgb led

rgb\_led.begin();

//Make sure LED is off to demonstrate start of sketch

rgb\_led.setPixelColor(0, rgb\_led.Color(0, 0, 0)); // Turn off LED

rgb\_led.show(); // Set new value

Serial.println("Setup Complete");

}

void loop() {

// read sensor1 value; display to A

uss\_value = getSensorData();

Serial.print("UltraSonic Sensor Value: ");

Serial.println(uss\_value);

// delay(10);

if (uss\_value <= 800) {

Serial.println("alarm on");

alarm();

}

else if (uss\_value > 800) {

Serial.println("alarm off");

disarmAlarm();

}

}

// display system on/off state in L

// display alarm off in J

/\* if (sensors have triggered past threshhold) {

alarm();

// even if sensors change to below threshhold, the alarm doesn't off until the pin is entered

}

if(Serial.available()){

communication = Serial.readStringUntil('\n');

Serial.println(communication);

if(communication[0] == 'M'){

communication.remove(0,1);

key\_value = communication.toInt();

getPinCode(key\_value);

}

}

void getPincode(int key) {

if (isAlarmOn == True) {

userEnteredPin[index] = key;

++index;

if (index == 4) {

index = 0;

if (strcmp(userEnteredPin, ourCode) == 0 ) {

disarmAlarm();

}

else {

invalidCode();

}

}

else {

return;

}

}

long getSensorData() {

// Clears the trigPin

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds

duration = pulseIn(echoPin, HIGH);

return duration;

}

void alarm() {

// change the display in J to be alarm ON

// buzzer on

digitalWrite(BuzzerPin, HIGH);

// Red Light On

rgb\_led.setPixelColor(0, rgb\_led.Color(255, 0, 0)); // Color mix for LED

rgb\_led.show(); // Set the LED

delay(5); // Delay for visual effect

}

void invalidCode()

{

// display code incorrect

}

void disarmAlarm()

{

alarm\_state = FALSE;

// turn the lights off

rgb\_led.setPixelColor(0, rgb\_led.Color(0, 0, 0)); // Turn off LED

rgb\_led.show(); // Set new value

delay(5);

// buzzer off

digitalWrite(BuzzerPin, LOW);

// maybe temporarily freeze uss\_value while the person moves away?

}

