|  |
| --- |
| #include <Servo.h>  //Variables String inData; boolean amHome = true; boolean oGarage = false; boolean oHalls = false; boolean oDining = false; boolean oKitchen = false; boolean oLiving = false; boolean oBathroom = false; boolean oLock = true; boolean intruder = false; boolean oGDoor = false;  //Pin Defines #define echo 2 #define trigger 3 #define ledHalls 13 #define ledGarage 12 #define ledDining 6 #define ledKitchen 4 #define buzzer 9 #define ledLiving 7 #define ledBathroom 5  Servo servoGarage; Servo servoLock;  void alarm() { //detects change in distance for motion alarms  int duration, distance;  digitalWrite(trigger, HIGH);  delayMicroseconds(1000);  digitalWrite(trigger, LOW);  duration = pulseIn(echo, HIGH);  distance = (duration / 2) / 29.1;  Serial.println(distance);  if (distance >= 35) {  intruder = false;  } else {  intruder = true;  } }  void setup() {  // put your setup code here, to run once:  Serial.begin(9600);   //sets up pin modes  pinMode(trigger, OUTPUT);  pinMode(buzzer, OUTPUT);  pinMode(ledHalls, OUTPUT);  pinMode(ledGarage, OUTPUT);  pinMode(ledDining, OUTPUT);  pinMode(ledKitchen, OUTPUT);  pinMode(ledLiving, OUTPUT);  pinMode(ledBathroom, OUTPUT);  pinMode(echo, INPUT);   servoGarage.attach(10);  servoLock.attach(11);   //starting positions for servos  servoGarage.write(0);  servoLock.write(185);  }  void loop() {  // put your main code here, to run repeatedly:  //motion detection  alarm();  if (intruder == true && amHome == false) { //sets off alarm  digitalWrite(buzzer, HIGH);  } else {  digitalWrite(buzzer, LOW);  }    while (Serial.available() > 0) { //While connected to bluetooth  char received = Serial.read();   if (received == '\*')  {   switch (inData[0]) {   case 'F':  if (inData [1] == 'L') { //front door lock  if (oLock) {  servoLock.write(90);  oLock = false;  } else {  servoLock.write(185);  oLock = true;  }  }  break;   case 'H':  if (inData[1] == 'B') { //hallway lights  if (oHalls) {  digitalWrite(ledHalls, LOW);  oHalls = false;  } else {  digitalWrite(ledHalls, HIGH);  oHalls = true;  }  }  break;   case 'B':  if (inData[1] == 'B') { //bathroom light  if (oBathroom) {  digitalWrite(ledBathroom, LOW);  oBathroom = false;  } else {  digitalWrite(ledBathroom, HIGH);  oBathroom = true;  }  }  break;   case 'L':  if (inData[1] == 'B') { //living room light  if (oLiving) {  digitalWrite(ledLiving, LOW);  oLiving = false;  } else {  digitalWrite(ledLiving, HIGH);  oLiving = true;  }  }  break;   case 'K':  if (inData[1] == 'B') { //kitchen light  if (oKitchen) {  digitalWrite(ledKitchen, LOW);  oKitchen = false;  } else {  digitalWrite(ledKitchen, HIGH);  oKitchen = true;  }  }  break;   case 'D':  if (inData[1] == 'B') { //dining room light  if (oDining) {  digitalWrite(ledDining, LOW);  oDining = false;  } else {  digitalWrite(ledDining, HIGH);  oDining = true;  }  }  break;   case 'Z':  if (inData[1] == 'T' && amHome) { //security system  amHome = false;  } else {  amHome = true;  }  break;   case 'G':  if (inData[1] == 'D') { //open/close garage  if (oGDoor) {  servoGarage.write(0);  oGDoor = false;  } else {  servoGarage.write(180);  oGDoor = true;  }   } else if (inData[1] == 'B') { //garage light  if (oGarage) {  digitalWrite(ledGarage, LOW);  oGarage = false;  } else {  digitalWrite(ledGarage, HIGH);  oGarage = true;  }  }  break;   case 'A':  if (inData[1] == 'B') { //turns all lights off  digitalWrite(ledHalls, LOW);  digitalWrite(ledGarage, LOW);  digitalWrite(ledKitchen, LOW);  digitalWrite(ledDining, LOW);  digitalWrite(ledLiving, LOW);  digitalWrite(ledBathroom, LOW);   oGarage = false;  oHalls = false;  oKitchen = false;  oDining = false;  oLiving = false;  oBathroom = false;  }  break;   default:  //nothing  break;   }   String inData;   } else {  if (inData.length() >= 2) { //resets inData  inData = "";  }  inData += received; //adds what it recieved to the data   }  } } |