NAME- SWAMYNATHAN VIGNESH

REG NO-RA1611008010011

BRANCH- INFORMATION TECHNOLOGY

SUBJECT- IOT PROJECT

FACULTY-M SAFA

PROJECT TITLE-HOME INTRUSION SYSTEM WITH ALARM SYSTEM

Githublink:

Youtubelink:

HOME INTRUSION SYSTEM::

Home security is becoming necessary nowadays as the possibilities of intrusion are increasing day by day .Lot of Security companies are available towards protecting house from getting vandalized or so. But still there is no much guarantee that the house be safe or even if the house vandalized, security personnel safe and sound to report to police nearby. This leads to property loss and damage. Lot of research been carried out employing sensors like PIR, Sensor Camera, GSM towards detecting the intruder at home. But the drawback in all these systems is that they are all expensive to be deployed integrated with LCD panel or Camera. Again with PIR sensor or Ultrasonic Sensor integrated with GSM, there is good possibility of false intruder detection based on line of sight cut by any entity and not necessarily an intruder. Currently homes in India still rely on security service personnel and no home security system been employed so far. So taking the above mentioned aspects into consideration, we here have developed an economical and affordable Home security system which have integrated the security component by making use of sensors like PIR, Temperature, humidity etc to sense the motion, change in temperature and humidity in room from normalcy rather relying on PIR sensor for change in motion only . The owner of the house is informed about any intruder by sending a text message by using of GSM module. All these activities are controlled by AtMega microcontroller of Arduino.

COMPONENTS USED:

LDR(LIGHT DEPENDENT RESISTOR)

Light dependent resistors, LDRs or photoresistors are often used in circuits where it is necessary to detect the presence or the level of light.

They can be described by a variety of names from light dependent resistor, LDR, photo-resistor, or even photo cell, photocell or photoconductor.

Although other devices such as photodiodes or photo-transistor can also be used, LDRs or photo-resistors are a particularly convenient electronics component to use. They provide large change in resistance for changes in light level.

In view of their low cost, ease of manufacture, and ease of use LDRs have been used in a variety of different applications. At one time LDRs were used in photographic light meters, and even now they are still used in a variety of applications where it is necessary to detect light levels.

DRs are made from semiconductor materials to enable them to have their light sensitive properties. Many materials can be used, but one popular material for these photoresistors is cadmium sulphide, CdS, although the use of these cells is now restricted in Europe because of environmental issues with the use of cadmium. Similarly cadmium CdSe is also restricted. Other materials that can be used include lead sulphide, PbS and indium antimonide, InSb. Although a semiconductor material is used for these photoresistors, they are purely passive devices because they do not possess a PN junction, and this separates them from other photodetectors like photodiodes and phototransistors.

NODE MCU::

The NodeMCU (Node MicroController Unit) is an open source software and hardware development environment that is built around a very inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains all crucial elements of the modern computer: CPU, RAM, networking (wifi), and even a modern operating system and SDK. When purchased at bulk, the ESP8266 chip costs only $2 USD a piece. That makes it an excellent choice for IoT projects of all kinds.

However, as a chip, the ESP8266 is also hard to access and use. You have to solder wires, with the appropriate analog voltage, to its PINs for the simplest tasks such as powering it on or sending a keystroke to the "computer" on the chip. And, you have to program it in low-level machine instructions that can be interpreted by the chip hardware. While this level of integration is not a problem when the ESP8266 is used as an embedded controller chip in mass-produced electronics, it is a huge burden for hobbyists, hackers, or students who want to experiment with it in their own IoT projects.

Borrowing a page from the successful playbooks of Arduino or a Raspberry Pi, the NodeMCU project aims to simplify ESP8266 development. It has two key components.

1. An open source ESP8266 firmware that is built on top of the chip manufacturer's proprietary SDK. The firmware provides a simple programming environment based on eLua (embedded Lua), which is a very simple and fast scripting language with an established

developer community. For new comers, the Lua scripting language is easy to learn.

2. A DEVKIT board that incorporates the ESP8266 chip on a standard circuit board. The board has a built-in USB port that is already wired up with the chip, a hardware reset button, wifi antenna, LED lights, and standard-sized GPIO (General Purpose Input Output) pins that can plug into a bread board. Figure 1 shows the DEVKIT board, and Figure 2 shows the schema of its pins.

NANO ARDUINO BOARD::

- **Arduino Nano** is a small, compatible, flexible and breadboard friendly Microcontroller board, developed by Arduino.cc in Italy, based on ATmega328p (Arduino Nano V3.x) / Atmega168 (Arduino Nano V3.x).
- It comes with exactly the same functionality as in Arduino UNO but quite in small size.
- It comes with an operating voltage of 5V, however, the input voltage can vary from 7 to 12V.
- **Arduino Nano Pinout** contains 14 digital pins, 8 analog Pins, 2 Reset Pins & 6 Power Pins.
- Each of these Digital & Analog Pins are assigned with multiple functions but their main function is to be configured as input or output.
- They are acted as input pins when they are interfaced with sensors, but if you are driving some load then use them as output.
- Functions like pinMode() and digitalWrite() are used to control the operations of digital pins while analogRead() is used to control analog pins.

- The analog pins come with a total resolution of 10bits which measure the value from zero to 5V.
- Arduino Nano comes with a crystal oscillator of frequency 16 MHz. It is used to produce a clock of precise frequency using constant voltage.
- There is one limitation using Arduino Nano i.e. it doesn’t come with DC power jack, means you can not supply external power source through a battery.
- This board doesn’t use standard USB for connection with a computer, instead, it comes with Mini USB support.
- Tiny size and breadboard friendly nature make this device an ideal choice for most of the applications where a size of the electronic components are of great concern.
- Flash memory is 16KB or 32KB that all depends on the Atmega board i.e Atmega168 comes with 16KB of flash memory while Atmega328 comes with a flash memory of 32KB. Flash memory is used for storing code. The 2KB of memory out of total flash memory is used for a bootloader.

ABSTRACT::

A laser alarm system operates by projecting a beam of invisible laser light across a doorway or window opening. When the light is broken, it activates a buzzer or alarm. The principles are very similar to those of lower tech burglar alarms. A laser alarm requires only slightly more sophisticated electronics and can be put together by anyone with a soldering gun and a knack for tinkering with basic circuits and transistors.

Theory::

This system for security uses the combination of LASER light and LDR. The LDR module has an onboard potentiometer to adjust the sensitivity of LDR, so that it only senses laser light falling onto it. The concept is quite simple and similar to what we see in movies where antique, priceless ornaments are protected under laser lights. As someone crosses these lights, an alarm runs on to indicate unauthorised presence. This project works similarly. In normal conditions, where there is always laser light falling on the LDR, the LDR module always gives a high signal to microcontroller. When someone crosses this laser light, it will behave as an obstruction between the LDR module and laser light, resulting in no light falling on LDR. In such cases LDR module gives a low signal to the microcontroller, which indicates it to switch on an alarm.

WORKING::

1. Take the serial wires and connect the LDR with the buzzer and the arduino.

2 once the connections are made, fix the LDR at a certain point in a wall and use the battery powered laser to incident a beam of light on the LDR directly .

3. now that the physical setup is done we need to reset the arduino and the nodemcu.

4.once the values are reset we can see the message HOUSE IS PROTECTED appear on the adafruit dashboard.

5. once this is done we connect our project to the wifi.(NOTE::the wifi can be a mobile hotspot also)

6.once the wifi has been connected the setup for the project is complete.

7. Now try disrupting the laser beam of light by creating an obstruction

8. we see that the moment the beam of light is disrupted the alarm buzzes .the alarm buzzes for 5 sec before automatically coming to a stop

9. once the alarm starts ringing a change in pin is detected by the node mcu which sends a corresponding status message to the adafruit server.

10. The node mcu and the arduino need to be reset for them to work again

PRECAUTIONS::

Cheap laser pointers that we find in most stores are generally restricted to 5mW or less. These are generally considered safe. However, it is still possible to damage your eyes if you are not careful. When working with lasers, it is a good idea to wear the appropriate eye protection. Avoid looking directly at the laser diode. Also never point lasers at aircraft.

CODE(MQTT)::

```
#include "config.h"

#include <ESP8266WiFi.h>

#include <DNSServer.h>

#include <ESP8266WebServer.h>

#include <WiFiManager.h>

#include <SoftwareSerial.h>

SoftwareSerial swSer(D3, D4, false, 256);

String response;

String st11 = "House is Protected";


int current = 0;

int last = -1;
```

```
AdafruitIO_Feed *st1 = io.feed("st1");

void setup()
{
  Serial.begin(9600);
  swSer.begin(9600);
  WiFiManager wifiManager;
  wifiManager.autoConnect("AutoConnectAP");
  Serial.println("connected...:)");
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.print("Connecting to Adafruit IO");
  io.connect();
  while(io.status() < AIO_CONNECTED) {
   Serial.print(".");
   delay(500);
  }
  Serial.println();
  Serial.println(io.statusText());
  delay(2000);
  pinMode(LED_BUILTIN, OUTPUT);
  for(int i=0;i<5;i++)
  {
    digitalWrite(LED_BUILTIN, LOW);
    delay(100);
    digitalWrite(LED_BUILTIN, HIGH);
    delay(100);
  }
  st1->save(st11);
}

int ESPwait(String stopstr, int timeout_secs)
{
  bool found = false;
  char c;
  long timer_init;
  long timer;

  response="";
```

```
timer_init = millis();
while (!found) {
  timer = millis();
  if (((timer - timer_init) / 1000) > timeout_secs) { // Timeout?
    Serial.println("!Timeout!");
    return 0;  // timeout
  }
  if (swSer.available()) {
    c = swSer.read();
    //Serial.print(c);
    response += c;
    if (response.endsWith(stopstr)) {
      found = true;
      delay(10);
      swSer.flush();
      Serial.flush();
      Serial.println();
    }
  } // end Serial1_available()
} // end while (!found)
```

```
  return 1;
}

int ESPwait1(String stopstr, int timeout_secs)
{
  bool found = false;
  char c;
  long timer_init;
  long timer;

  response="";
  timer_init = millis();
  while (!found) {
    timer = millis();
    if (((timer - timer_init) / 1000) > timeout_secs) { // Timeout?
      Serial.println("!Timeout!");
      return 0;  // timeout
    }
    if (Serial.available()) {
      c = Serial.read();
      //Serial.print(c);
      response += c;
      if (response.endsWith(stopstr)) {
        found = true;
        delay(10);
        Serial.flush();
        Serial.println();
      }
    } // end Serial1_available()
  } // end while (!found)
  return 1;
}


void loop()
{
 char c;
 io.run();
if(swSer.available())
  {
```

```
c=swSer.read();
Serial.println(c);
if(c=='*')
  {
    if(ESPwait("#",3))
    {
    char * strtokIndx;
    response.remove(response.length()-1);
     response="";
     Serial.println("sending -> ");
    }
  }
}
if(Serial.available())
{

c=Serial.read();
Serial.println(c);
if(c=='*')
  {
      if(ESPwait1("#",3))
      {
      char * strtokIndx;
     //response.remove(response.length()-1);
     Serial.println(response);
     if(response.endsWith("1#"))
     {
  st11 = "Intrusion Detected";
  st1->save(st11);
  Serial.println("Sending");
     }
   response="";
  }
 }
}
}
```

CODE(ARDUINO BOARD)::

```
#include <SoftwareSerial.h>

SoftwareSerial esp8266(2, 3);

int a1=0;

void setup()

{

 pinMode(12, OUTPUT);

 digitalWrite(12, LOW);

 esp8266.begin(9600);

 Serial.begin(9600);

 Serial.println("Getting In..");

}


void loop()

{

a1=analogRead(A0);

//Serial.println(a1);

if(a1>400)

{

digitalWrite(12, HIGH);

esp8266.println("*1#");
Serial.println("*1#");
delay(5000);
digitalWrite(12,LOW);
while(1);
}
char c;
 if(esp8266.available())
 {
 c=esp8266.read();
 Serial.print((String)(c));
 }
 }
```

APPLICATION::

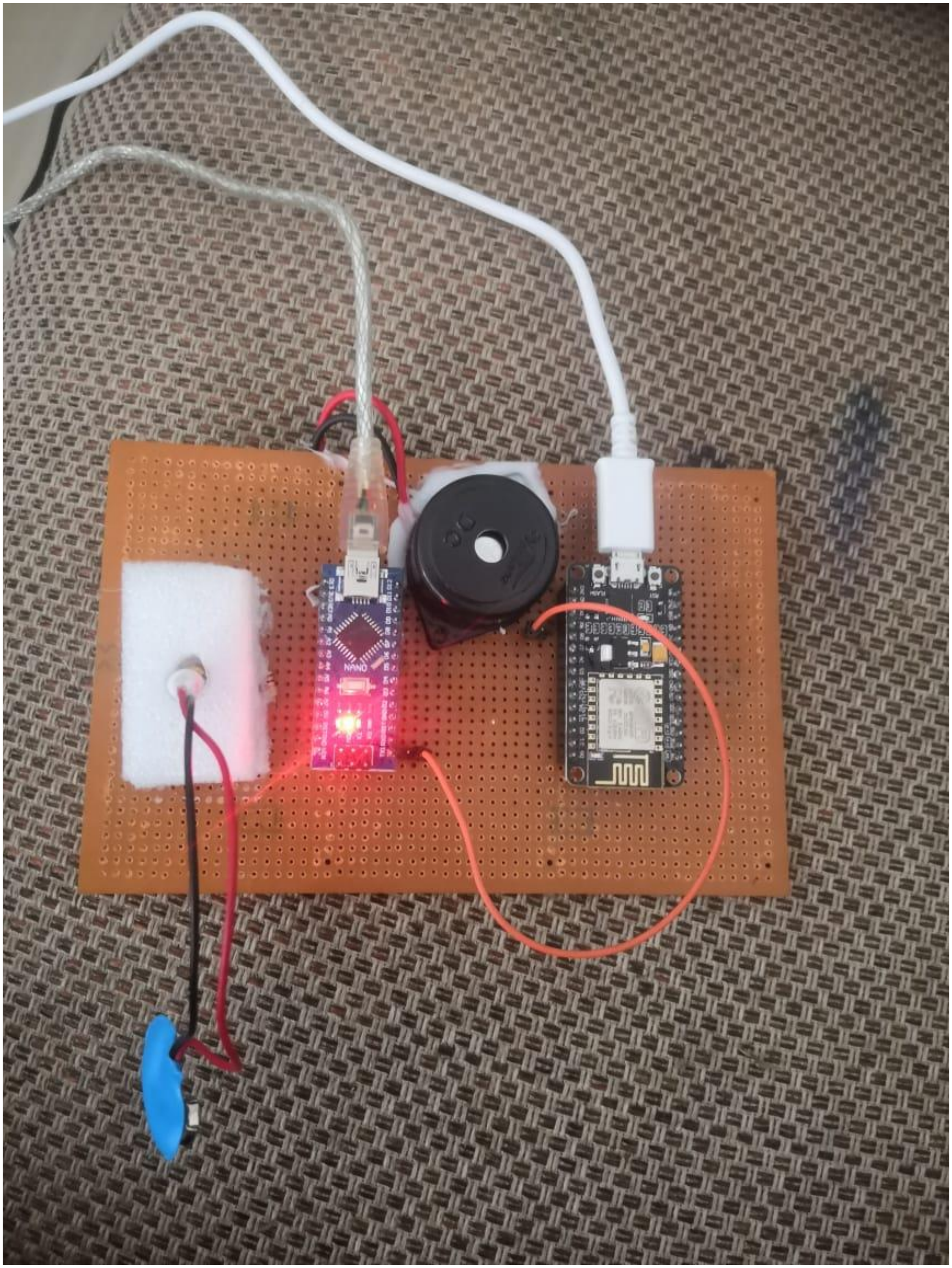
A laser security alarm is a system designed to detect intrusion – unauthorized entry – into a building or area. They are also called security alarms, security systems, alarm systems, intrusion detection systems, perimeter detection systems, and similar terms.

Burglar alarms are used in residential, commercial, industrial, and military properties for protection against burglary (theft) or property damage, as well as personal protection against intruders. Car alarms likewise protect vehicles and their contents. Prisons also use security systems for control of inmates.

Some alarm systems serve a single purpose of burglary protection; combination systems provide both fire and intrusion protection. Intrusion alarm systems may also be combined with closed-circuit television surveillance systems to automatically record the activities of intruders, and may interface to access control systems for electrically locked doors. Systems range from small, self-contained noisemakers, to complicated, multi-area systems with computer monitoring and control.

CONCLUSION::

Laser security systems are a high tech technology that used to be a part of home security only available to the wealthy. It is manually switch dependent sensors and a basic alarm unit. Laser security system a person moves in front of the sensor, that person triggers the system's alarm by cutting the laser. And the alarm signals the security monitoring company and local law enforcement. The basic alarm unit will also sound a loud alarm. Both analysis and experiment indicate that rather stringent requirements must be met in order to obtain efficient optical heterodyne detection. At some wavelengths it may provide the only means of overcoming noise and detect noise problems. The operation of LDR depends upon the photoconductivity process.



ADAFRUIT::

Adafruit.io is a *cloud service* - that just means we run it for you and you don't have to manage it. You can connect to it over the Internet. It's meant primarily for storing and then retrieving data but it can do a lot more than just that!

What can Adafruit IO do for me?

- Display your data in real-time, online
- Make your project internet-connected: Control motors, read sensor data, and more!
- Connect projects to web services like Twitter, RSS feeds, weather services, etc.
- Connect your project to other internet-enabled devices
- The best part? All of the above is do-able for **free** with Adafruit IO

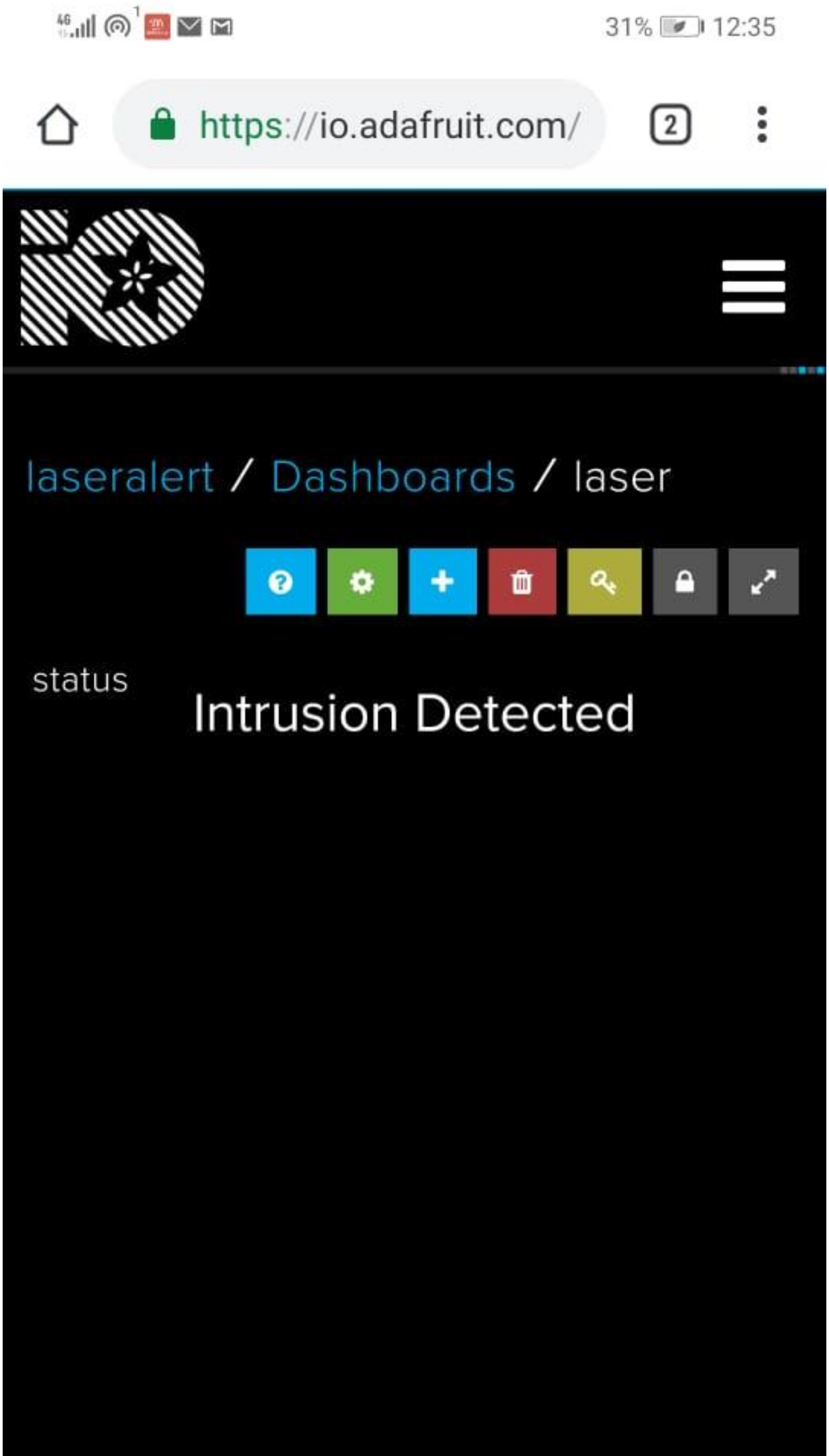
DASHBOARD::

Adafruit.io can handle and visualize multiple feeds of data. Want to display data from a temperature-humidity sensor alongside data from an air quality sensor and add a button to turn on the air-conditioner in your room?
No problem! Dashboards are a feature integrated into Adafruit IO which allow you to chart, graph, gauge, log, and display your data. You can view your dashboards from anywhere in the world.

TRIGGERS::

Use triggers in Adafruit IO to control and react to your data. Configure triggers to email you when your system goes offline, react to a temperature sensor getting too hot, and publish a message to a new feed.

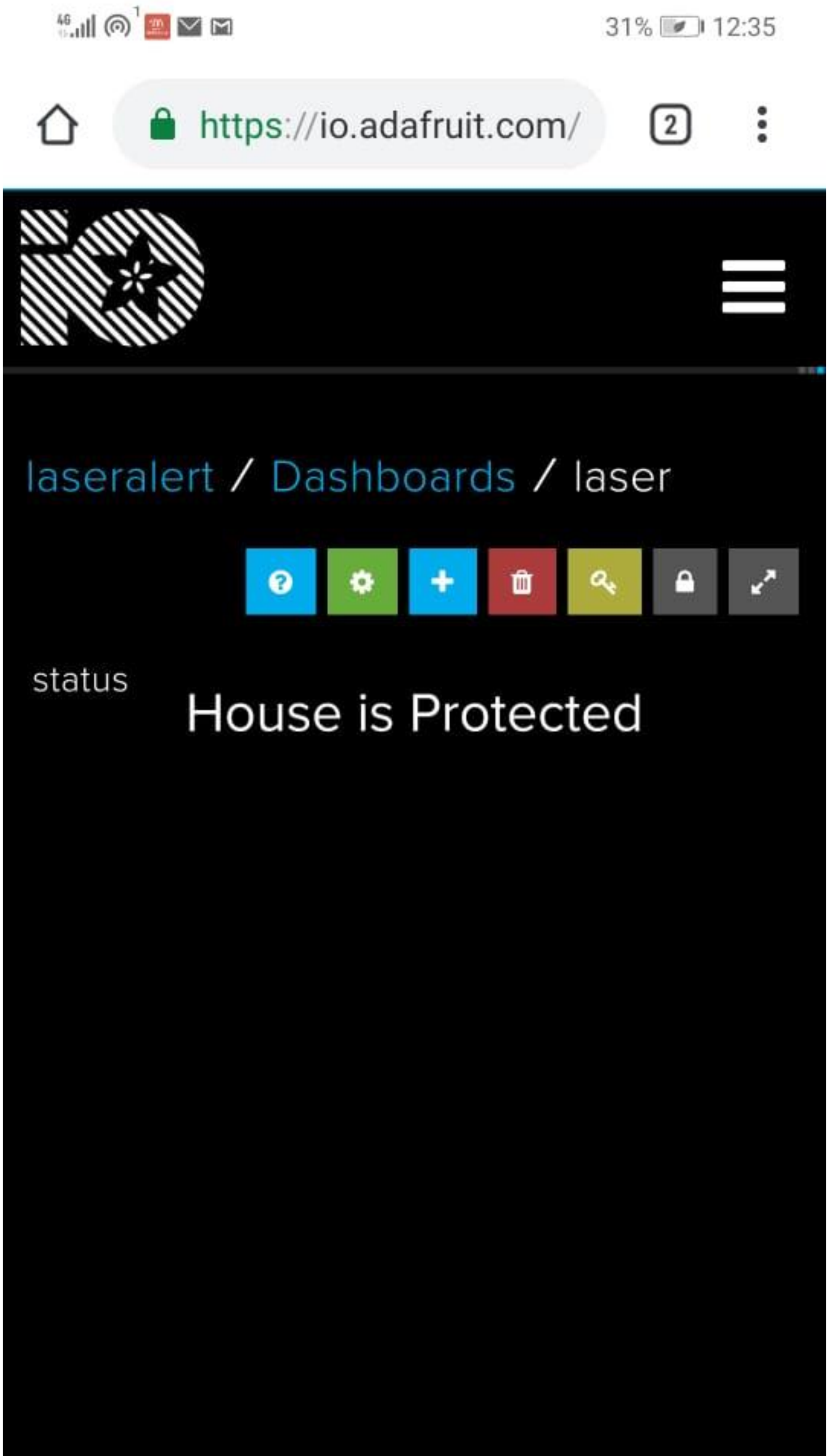
OUTPUT::



laseralert / Dashboards / laser

status

Intrusion Detected



status

House is Protected