

Use of Arduino with IFTTT to build an IoT system | IFTTT Maker

This post describes how to use Arduino with IFTTT. For sure you know that Arduino is a prototyping board widely used to build IOT projects because Arduino is very easy and it has an affordable price. IFTTT is a cloud platform that provides several services to automate some tasks and making different systems working together. IFTTT stands for *If This Then That*. It is a powerful cloud platform that helps us to connect to countless services like social services, email, SMS and so on. We will exploit these IFTTT features to build an Arduino IOT system that sends alerts. To make things simple and focus our attention on the integration between IFTTT and Arduino, we will build an Arduino IOT system that monitors the gas leakage. To explore how to connect Arduino and IFTTT, we will connect Arduino to a gas sensor. When there is an anomaly revealed by the sensor, Arduino will send an alert through IFTTT. This Arduino IOT project can be further extended adding new features or you can use a different sensor to monitor another physical property (such as temperature, humidity and so on).

Components Required

- Arduino Uno + internet shield (or MKR1000 as shown in the project)
- IFTTT account
- MQ-4 sensor
- Jump wirings

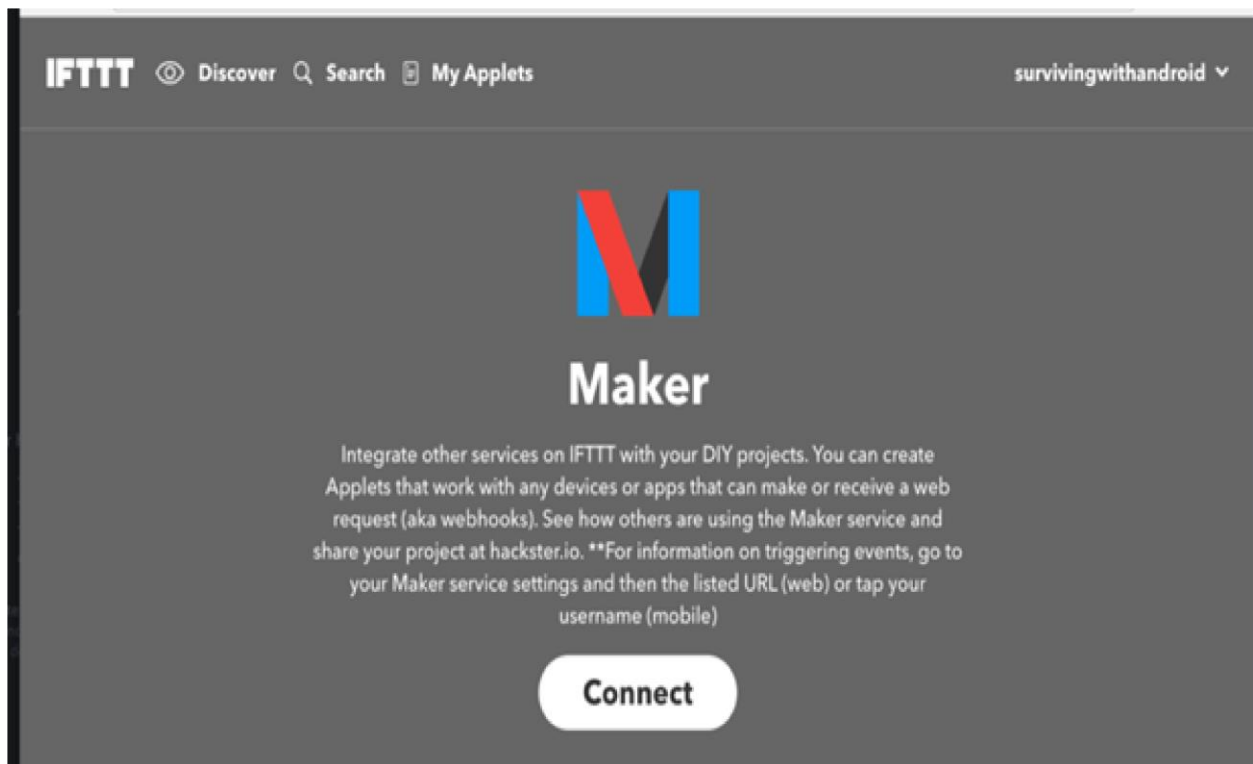
MQ-4 sensor is a fast and reliable sensor very sensitive to natural gas and CH₄ (Methane). It has a long life and is commonly used in the gas leakage detection. When you will turn on the gas sensor at the beginning it could smell a bit. This is normal do not worry. MQ-4 Arduino sensor has four pins:

- VCC (+5V)
- Ground GND
- Digital output
- Analog output

Configuring the IFTTT Maker channel to work with Arduino

In this step, we will configure the IFTTT maker channel, so that it connects to Arduino. The first step is creating an account if this is the first time you use this service. The services provided by IFTTT are free.

Once you have your account on IFTTT, then click on the search on the top and looks for maker service. This service is made for makers that want to integrate external services with Arduino or similar boards:



Now go to Applet and create a new Applet. Click on the plus sign and add the service. In this case, you should add Maker service the one we created before. Click on Receive a web request and configure your trigger. This is the event that triggers the notification process, in other words, as soon as IFTTT receives a web request it sends an email:

Receive a web request

This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap your username (mobile)

Event Name *

Alert



The name of the event, like "button_pressed" or "front_door_opened"

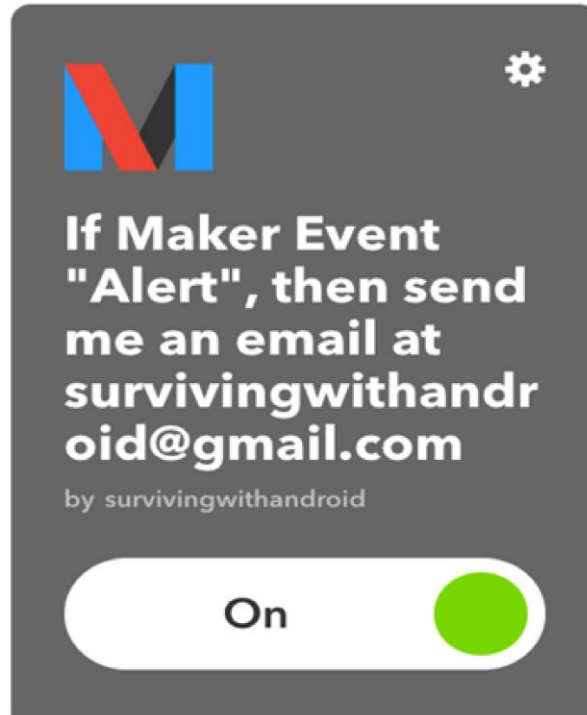
Create trigger

Create the trigger. Now we have to configure the other step: the service that should be executed when the trigger is fired. Click on the plus sign again (in the then part) and add the email service. You can add other services too:

Send me an email

This Action will send you an HTML based email. Images and links are supported.

Add email details and confirm. At the end you have your service configured in IFTTT:



That's all. Now to trigger this event we have to call the URL as shown in the service configuration. The Arduino system will call this URL when the sensor detects an anomaly in the gas concentration.

That's all. In this first step, we have configured IFTTT. Now Arduino can use IFTTT to send notifications. The first part of this Arduino IOT project is complete.

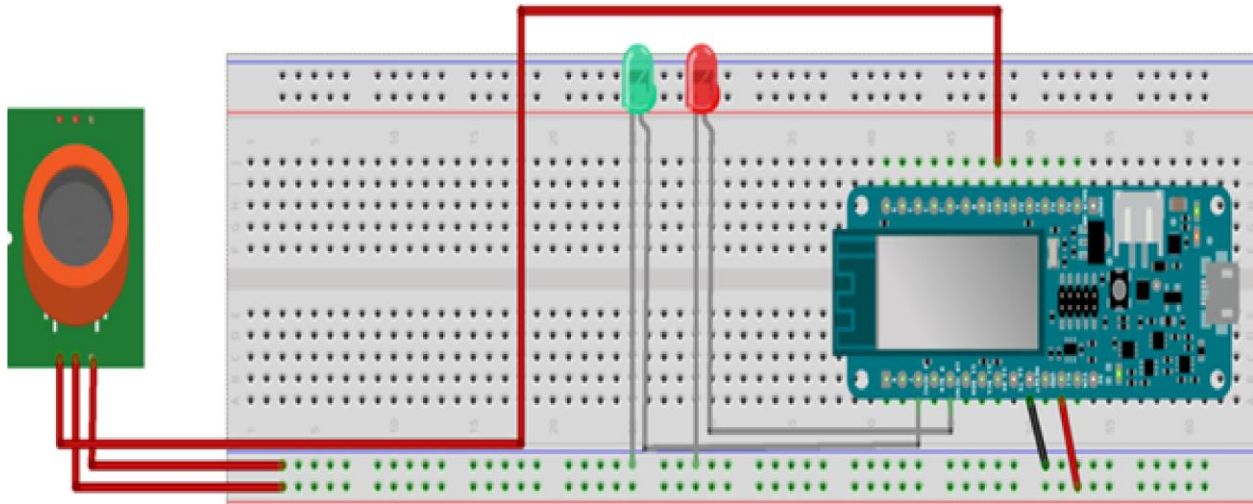
How to integrate Arduino with IFTTT

To trigger the event, Arduino will read the sensor values. To make things simple, this IOT system uses a MQ-4 sensor. You can use other sensors too like PIR, temperature, humidity and so on. The same steps can be applied to other sensors too. The process that stands behind this project doesn't depend on the sensor you will use.

What we want to do is creating a simple project that shows the gas concentration using two LEDs:

- Green Led: The gas concentration is under the threshold
- Red Led: Alert the gas concentration is above the threshold

We will use the analog output to check the gas concentration. The circuit below shows how to connect the gas sensor to Arduino:



Please notice that the connections to the gas sensor in the schematic above are not real, while the Arduino pin used to get data from the sensor is the real one (A5). In the picture above, the resistors are not showed.

In the schematic, there are two resistors that connect the Arduino digital pin and the LEDs. This resistor limits the current flowing through the LEDs. They are 220 Ohm. That's all. You are ready now to develop the sketch to monitor the Arduino sensor.

The sketch below describes how to implement the gas monitoring system:

```
int pinRedLed = 11;

int pinGreenLed = 8;

int pinSensor = A5;

int THRESHOLD = 250;

void setup() {

  pinMode(pinRedLed, OUTPUT);

  pinMode(pinGreenLed, OUTPUT);

  pinMode(pinSensor, INPUT);

  Serial.begin(9600);

}

void loop() {
```

```
int analogValue = analogRead(pinSensor);
Serial.println("Val: " + analogValue);
digitalWrite(pinGreenLed, HIGH);
if (analogValue <= THRESHOLD) {
digitalWrite(pinGreenLed, LOW);
digitalWrite(pinRedLed, HIGH);
}
else {
digitalWrite(pinRedLed, LOW);
}
delay(5000);
}
```

The code is very simple. In the beginning, it declares the pins we will use to connect to the sensor and LEDs. The green LED is always on while the red LED turns on only when the gas concentration is over the threshold.

Conclusion

In the end, you have implemented a simple Arduino IOT project in just a few steps. Integrating IFTTT maker channel and Arduino you can expand your Arduino and create interesting and useful IOT project with a few lines of code. In this tutorial, you learned how to create an Arduino IOT project that monitors sensor. The best part is the integration between Arduino and IFTTT so that this integration can be applied to other scenarios too.