GREEN TECHNOLOGY

Build an IoT soil moisture monitor using Arduino with an IFTTT alert system

[1]SUMEDHA(RA1611004010874)
Department of ECE, Team SRMSAT, SRM IST Kattankulathur (603203),
sumedha580@gmail.com
sumedha_ravinderkumar@srmuniv.edu.in
[2]ANUSH BHATIA (RA1711003010645)
Department of CSE, SRM IST Kattankulathur (603203)
anushbhatia1234@gmail.com

ABSTRACT

Food and water are a very precious and driving force in sustaining our lives. Production in agriculture and optimal use of water both are need of the hour. Efficient irrigation watering helps in saving water, getting better plant yields, reduce dependency on fertilizers and improve crop quality. Various methods, both laboratory, and field including remote sensing are available to measure soil moisture content, but the quickest and better one is with the use of soil moisture sensor electronic devices. For successful irrigation, it is necessary to monitor soil moisture content continuously in the irrigation fields. The selection of soil moisture probes is an important criterion in measuring soil moisture at different soil moisture sensors. They have their own advantages and disadvantages. The soil moisture sensors are used intensively at present because it gives real-time readings. With that use of renewable resources like solar cells will reduce cost.

INTRODUCTION

Agriculture plays a major role in economics as well as the survival of people in India. The purpose of this project is to provide an embedded based system for soil monitoring and irrigation to reduce the manual monitoring of the field and get the information via a mobile application. The system is proposed to help the farmers to keep a virtual eye on their crop and that might help in increase in production as well as a decrease in labor.

This is the project well simulated in Altair SmarCore

using various sensors such as

- Temperature sensors,
- Humidity,
- Tensiometer: Pressure,
- PH detection,
- Light sensor,
- Solar cells,
- Chlorophyll sensors

Based on the result,

- The applied sensors detecting for all parameters, combining the calculation to a helpful conclusion. That suggests different agriculture tactics and requirements of the crop according to the analyzed situation
- Watering the plant automatically whenever required, adding the automatic system to drip irrigation helps in saving water as well.
- PH detection and soil sensors help in nutrient requirements.

- The farmers can cultivate the appropriate crop that suits the soil.
- The automatic irrigation system is carried out when the soil temperature is high and moisture content reaches the threshold limit.
- Crop image is captured and it is sent to the field manager to suggest pesticides. Reducing the cost even further by solar power.
- We can implant a chlorophyll sensor to get it more precise. It may help in online disease inquiry as well.
- We can use a tensiometer to judge the absorbing capacity of the soil.
- Integrating this with drip irrigation will reduce the water consumed.
- The obtained sensor values are sent to the field manager through the internet and the crop suggestion is made through the mobile application.

SIMULATIONS :

To build this project we need:

- Arduino MKR1000 (or any compatible board)
- Moisture sensor- VH400
- Carriots free account – now known as Altair SmarCore
- IFFT account
- Google API
- Chlorophyll sensor 6025
- Camera

The determination of soil temperature is done using the DS18B20 sensor working on the Dallas one-wire protocol.

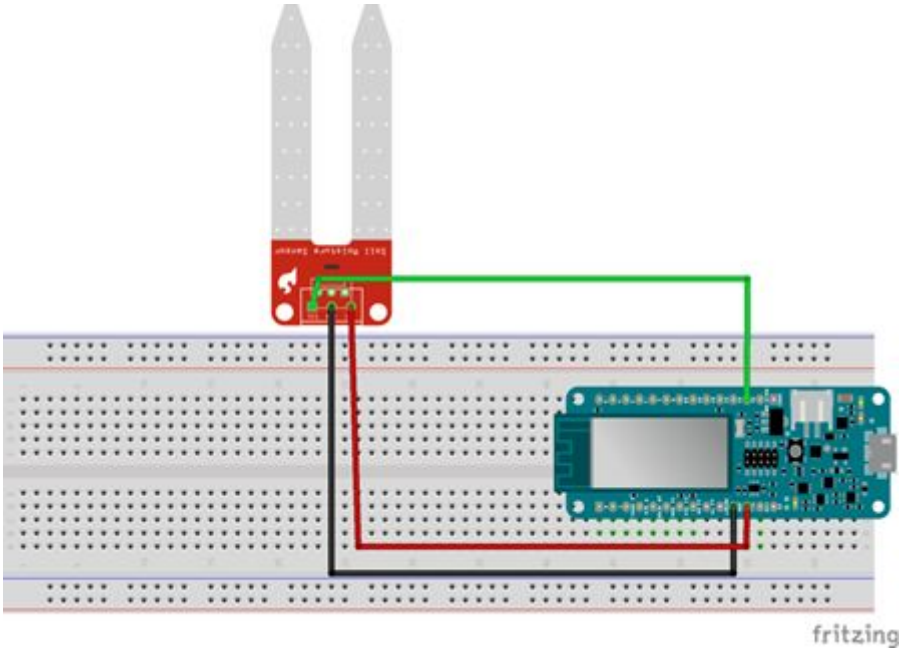
Tensiometers are simple soil moisture tension measuring devices used frequently in irrigation scheduling. This instrument does not measure soil moisture content directly, instead, it measures soil water tension. Generally, the response time of a tensiometer is 2 to 3 hours. There are tensiometers available which can be automated with the irrigation system with the help of pressure gauge.

VH400 Soil Moisture Sensor

VH400 soil moisture sensor is resistive-based soil moisture which measures the dielectric constant of soil. It helps in precise low-cost monitoring of soil water content. It has a rapid response time, can take reading in under one second and is much sensitive at higher volumetric water content. The soil moisture probe is inserted into the ground, preferably in a horizontal position at the root level. This sensor is small in size, rugged, waterproof, and consumes less power. It is also insensitive to the salinity of the water, does not corrode over time, and is sensitive to even small changes in water content. This type of sensor is sensitive to temperature changes in wet conditions, thus temperature measurement would always be required. The probe is usually attached to soil moisture reading device to form a wireless sensor network, such networks are extensively used in precision agriculture and smart irrigation. One such device is soil moisture data logger which displays moisture content readings on the digital screen. There are two means of communication between the system and the far user; first, the readings are sent via Short Messaging Service (SMS) on GSM network and second the readings can be stored in the memory card which can be transferred to a computer for analysis.

IoT soil moisture project overview

The idea that stands behind this project is building an IoT system that monitors the soil moisture detecting when it gets too dry. The Arduino MKR1000 controls the sensor sending the data to the Carriots IoT platform (now Altair SmartCore). This platform, in turn, stores the data coming from the sensor and detects when the values stored are under a threshold level. We will see later how to analyze the data. By now, we can assume that the Carriots IoT platform is able somehow to invoke an IFFT service that will send a short message to the user alerting him. Building this IoT system we can explore how to use several components of the IoT ecosystem. Moreover, this project displays the humidity soil status using a LEDs matrix. Let's see how to build it.



Retrieving data from the soil moisture sensor connected to Arduino

In this first step, we have to read the sensor data. This IoT project uses YL-38 + YL-69 sensor. This is an analog sensor that can be inserted into the soil we want to check. The picture below shows how to connect the sensor to Arduino:

The code is very simple, we read data from the A1 pin and then we calculate the humidity:

```
float moistureHum = analogRead(A1);
moistureHum = (1023 - moistureHum) * 100 /1023;
```

This is very simple, not much to explain about it. The value stored in `moistureHum` is the one we will send to IoT cloud platform. Moreover, to do it, Arduino MKR1000 has to connect to the internet so that it can send data. The code below describes how to connect Arduino to the WiFi:

```
#include "WiFi101.h"
WiFiClient client;
void setup() {
  Serial.begin(9600);
  Serial.print("Starting...");

  if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("WiFi shield not present");
    while (true);
  }
  connectToWifi();
```

```
}
```

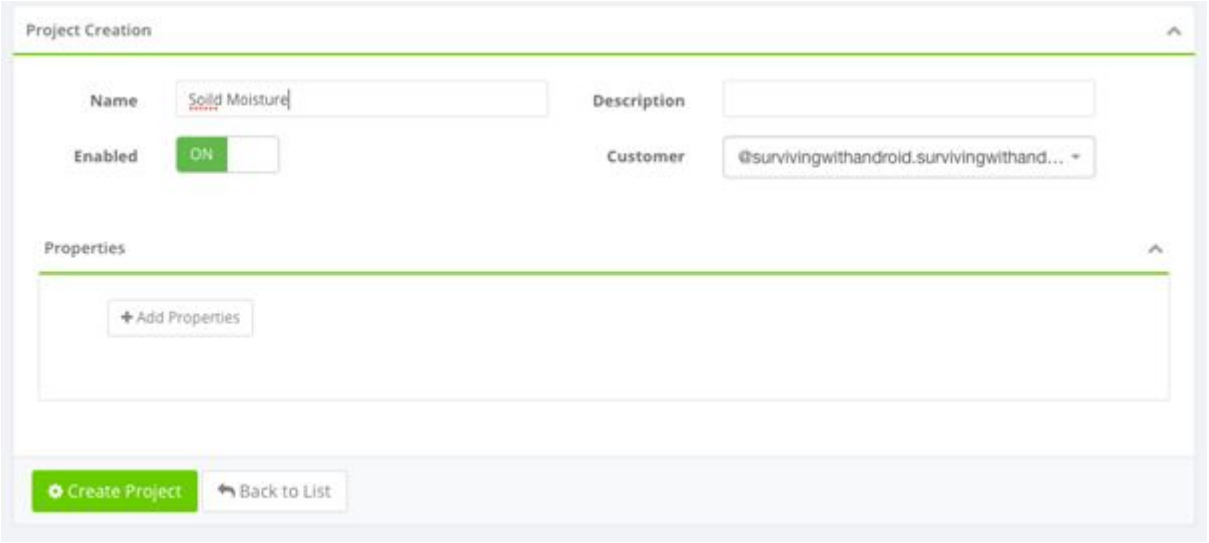
where the connectToWifi() is:

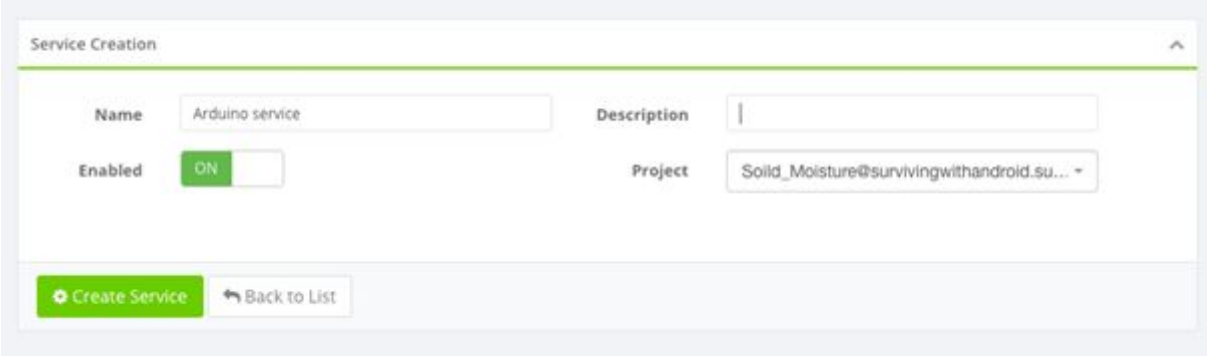
```
void connectToWifi() {
  while ( status != WL_CONNECTED) {
   Serial.print("Attempting to connect to WPA SSID: ");
   Serial.println(ssid);
   // Connect to WPA/WPA2 network:
   status = WiFi.begin(ssid, pass);
   // wait 10 seconds for connection:
  delay(10000);
 }
}
```

In this code, you have to provide the ssid and the pass (WiFi password).

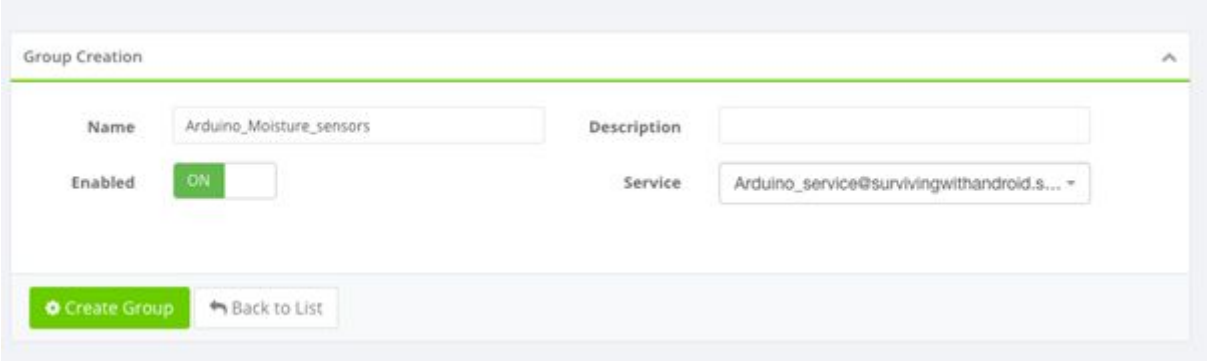
That's all. We can manage in this sketch the LEDs matrix showing the humidity level as described in the code attached to this article. Now we can focus our attention on the IoT cloud platform.

[bctt tweet="Learn how to build an IoT Arduino system to monitor the soil moisture. Step by step guide #IoT" username="survivingwithan"]



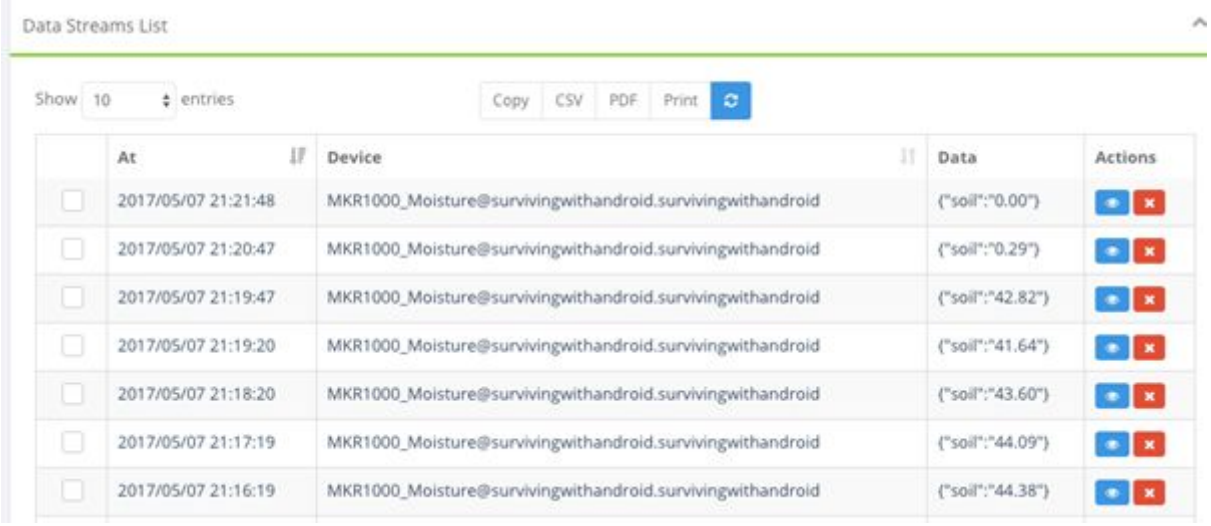


Finally, you have to create the group:



Once the arduino is configured. We have to connect it device to Carriots and start sending data. In this step, it is important the *developer id* shown in the picture above, it represents the unique device identifier we will use to bind the data coming from our Arduino device to the Carriots device. Another important parameter is your *API Key*. You can find it in Settings -> Api key menu. In order to send data, add this function to the sketch shown above:

Notice that the function sends a JSON payload containing the data read from the sensor. This function must be invoked in the loop() method. Running the sketch, we can notice that the device sends data to the Carriots as shown below:



Implementing an Arduino alert system using IFTTT

The next step is monitoring the data. Usually, in an IoT system, we are not only interested in acquiring data from the sensor but we want to monitor such information to take corrective actions when the values are out of a specific interval. There are several actions we can take, in this example we want to inform the user alerting him/her. Even if Carriots has a built-in email system, we prefer to integrate with Carriots other useful and interesting platforms

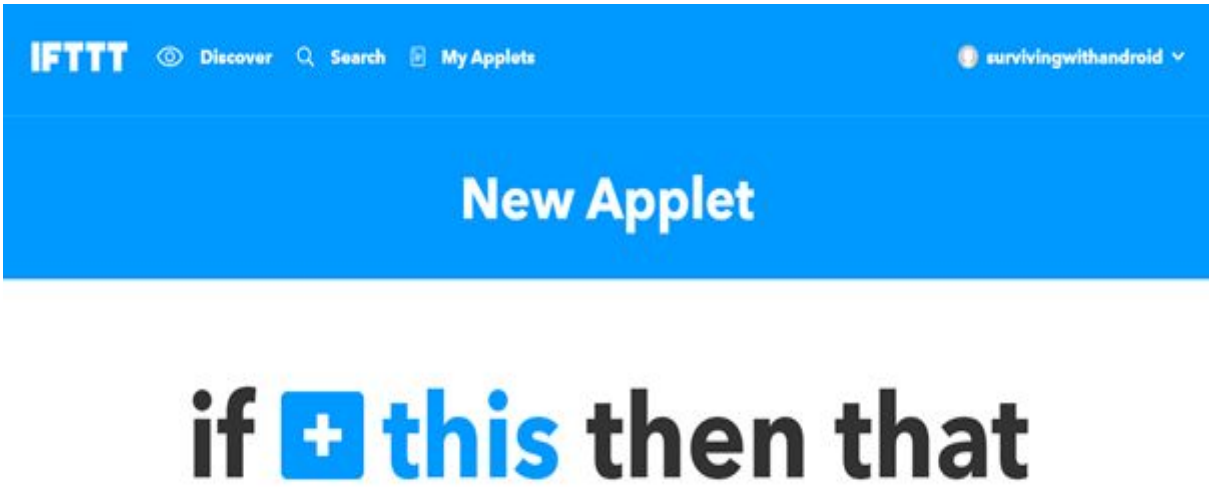
called IFTTT. This platform provides several integration services we can use and integrate into our IoT projects.

In order to alert the user we need two components:

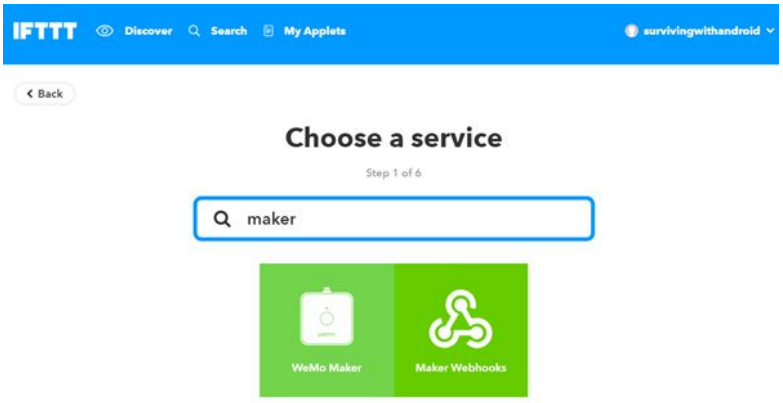
1. A monitoring data system
2. Alerting system

As monitoring data system, this IoT system uses Carriots listener. A listener is a process that analyzes the incoming values and applies a specific rule. When the rule is verified then it invokes a script. The interesting aspect of Carriots is that we can use Groovy as a scripting language to invoke external services.

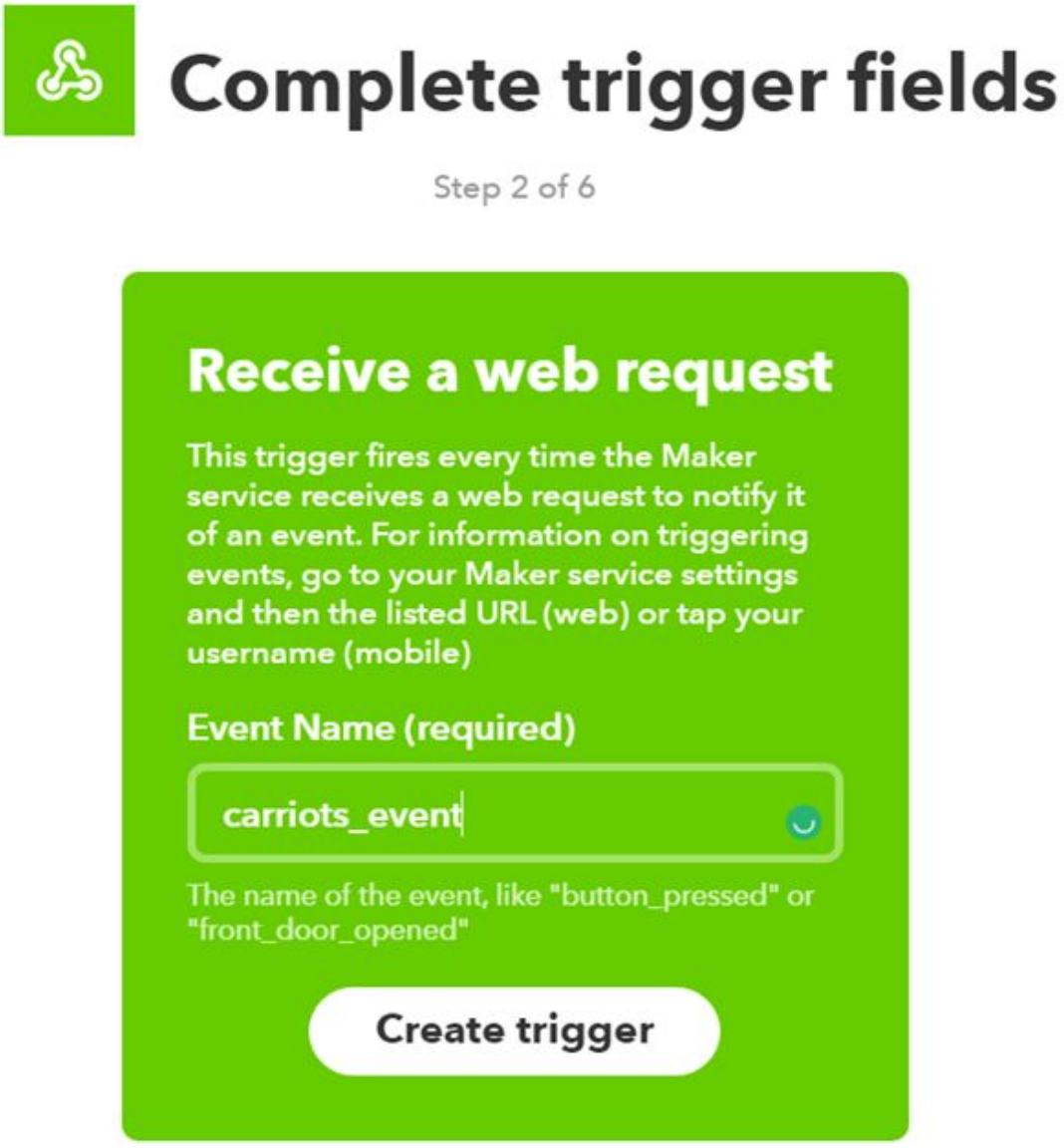
The alerting system is built on IFTTT. Before completing the work on Carriots, it is useful to configure the IFTTT. As stated before, we want to send a short message when the humidity hits a threshold level. In order to achieve it, we configure a short message service in IFTTT. It is required you have a free account in order to complete this task. As a first step, it is necessary to create a new Applet:



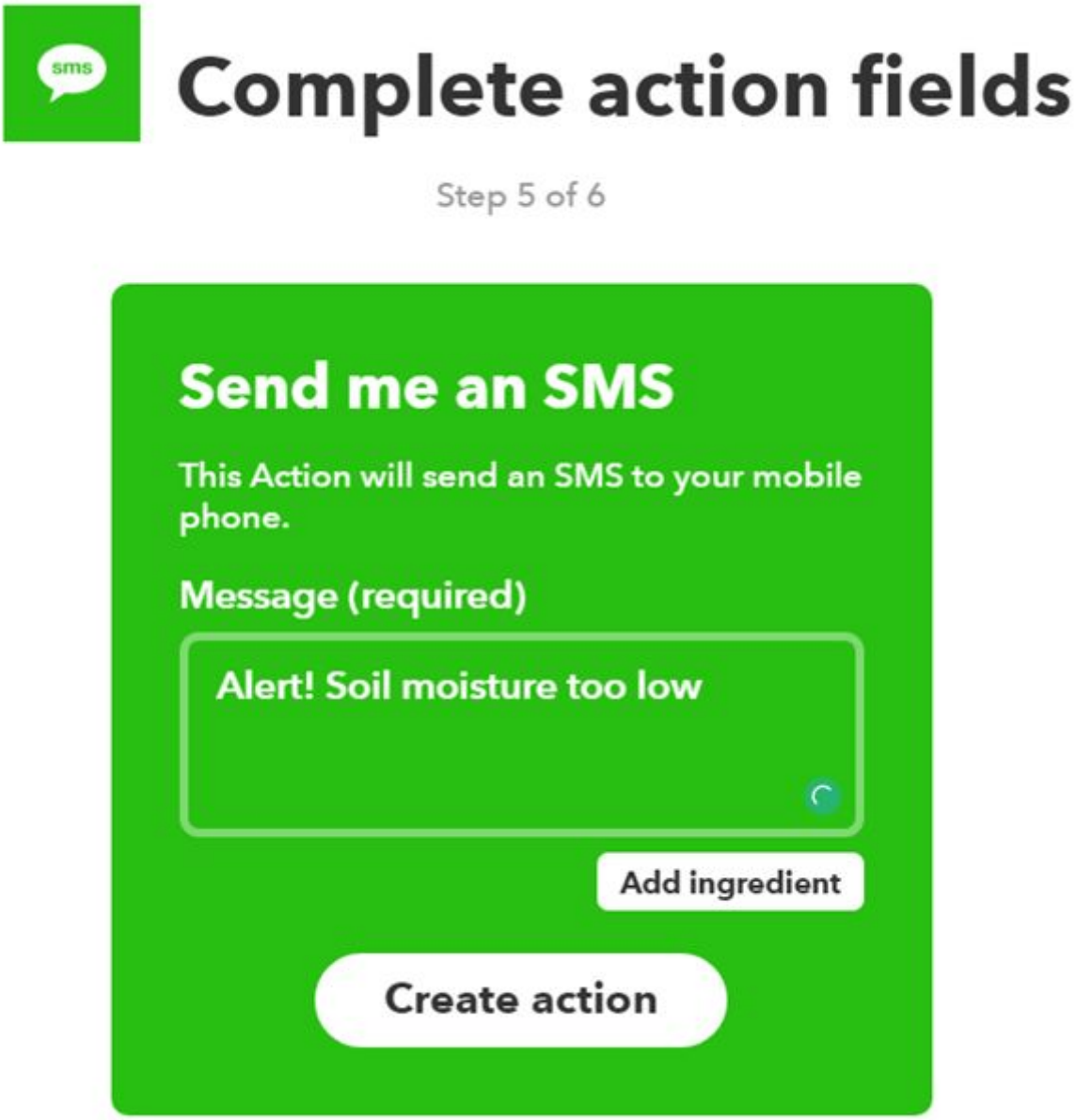
Now click on plus sign to add the service and search for Maker service:



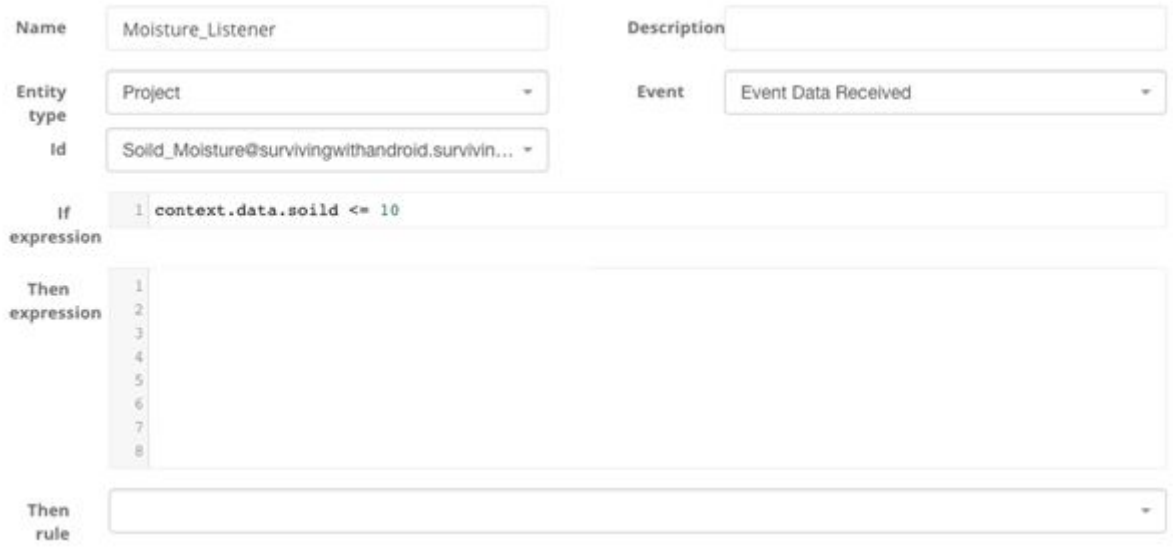
Select Maker webhooks to enable IoT maker. Now we have to configure the maker service adding the event name that triggers the sending message process:



Finally, enable the sending message service configuring all the required parameters like the destination number and the message body:



That’s all. Now we can focus our attention on the listener in Carriots platform. Let us create a new listener that invoke the URL related to the applet we have just created. When the listener invokes the URL, IFFT sends a short message. The picture below shows how to configure the listener:



The last step is configuring the expression. We can write it using Groovy as described in this **example**.That's all, now you can test the project verifying that when the soil moisture is lower than the threshold level, we will get a short message on our mobile.

CONCLUSION:

End Users can tailor the mote operation to a variety of experimental setups, which will allow farmers to reliably collect data from locations previously inaccessible on a micro-measurement scale. Such a system can be easily installed and maintained. This paper successfully applies wireless sensor networks on agro-ecology fields by investigating environmental situations. The complete real-time and historical environment information is expected to help the agro-ecological specialists achieve efficient management and utilization of agro-ecological resources.