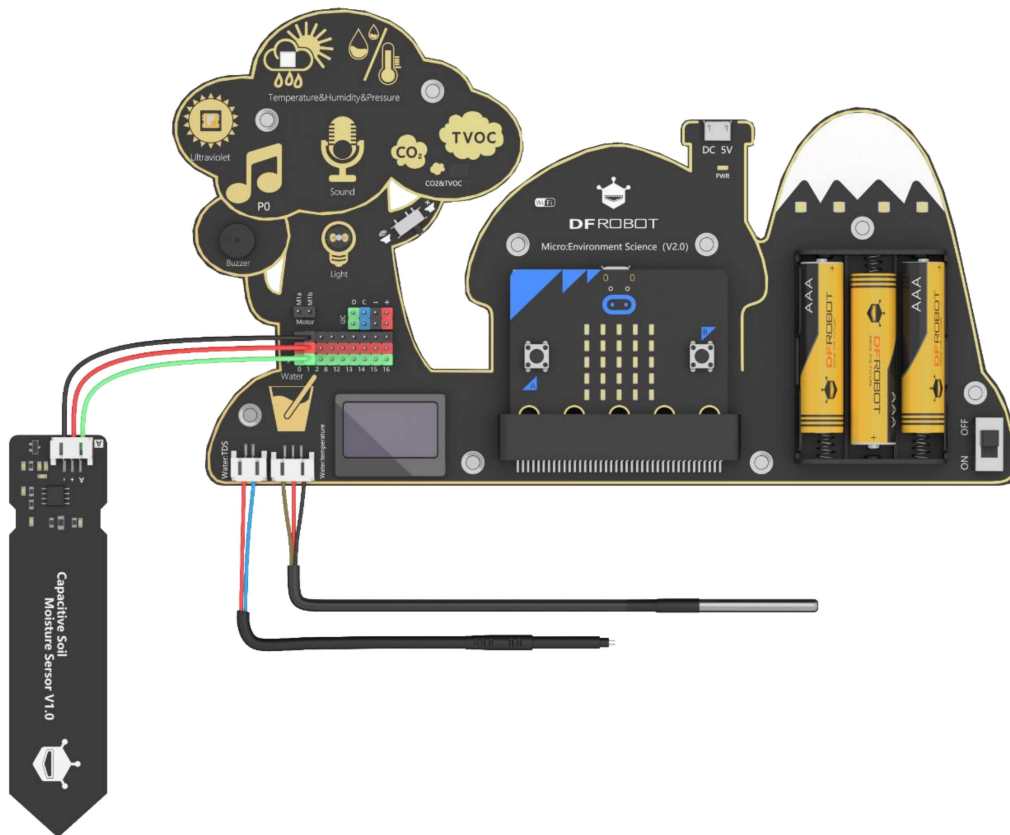


# Tutorial of Environment Science Expansion

## Board for micro:bit -V2.0

Based on MakeCode

(SKU: MBT0034)



[www.DFRobot.com](http://www.DFRobot.com)

# Content

<b>Chapter1: Introduction to Environment Science Expansion Board V2.0.....</b>	<b>3</b>
Introduction.....	3
V2.0 Update.....	3
Specification.....	3
Board Overview.....	4
Parameters.....	4
<b>Chapter2: MakeCode graphical programming.....</b>	<b>7</b>
Link and Library.....	7
Module Function Instruction.....	7
1. Measurement of Temperature and Humidity and The Comfort Level.....	17
2. Measurement of Atmospheric Pressure.....	18
3. Measurement of Sound Intensity.....	18
4. Measurement of Light Intensity.....	20
5. Measurement of Ultraviolet Rays.....	21
6. Measurement of TVOC.....	21
7. Measurement of CO2.....	22
8. Measurement of Water Temperature and Application (Conversion between Celsius and Fahrenheit).....	24
9. Measurement of Water Quality (TDS).....	25
<b>Chapter3: Applications of IoT Platform.....</b>	<b>26</b>
Example Project 1: IoT Environment Detection System (Based on Easy IoT).....	26
Sample Project 2: Light-Sensitive Automatic IoT Clock (Based on Easy IoT).....	28
Sample Program 3: IoT Greenhouse Environment Alarm (Based on IFTTT).....	30
Sample Project 4: IoT Sunshine Intensity Recorder (Based on ThingSpeak).....	38

# Chapter1: Introduction to Environment Science Expansion Board V2.0

**Product Name:** Micro: Environment Science ( V2.0 )

**SKU:** MBT0034

**Product Link:**

## Introduction

This micro:bit-based expansion board, specially designed for Maker education, allows students to measure environmental conditions for scientific experiments by using rich on-board sensors. It aims to provide a platform for students to learn theory with practices and bring science education closer to daily life!

Integrated sensors include UV sensor, temperature sensor, humidity sensor, air pressure sensor, sound sensor, light sensor, water temperature sensor, and a TDS (Total Dissolved Solids) water quality sensor, soil moisture sensor, etc.

## V2.0 Update

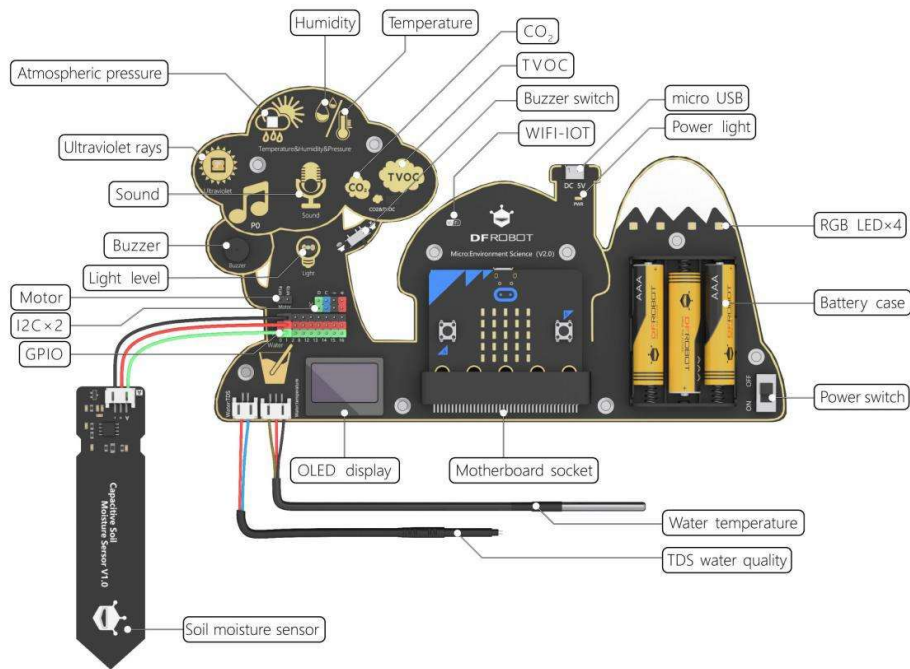
1. Comes with WiFi-IoT card, support IoT platforms such as IFTTT, ThingSpeak, EasyIoT
2. Supports for the measurements of TVOC and CO<sub>2</sub>
3. MicroUSB power supply port, more convenient for classroom teaching
4. Onboard 4-way RGB lights
5. 1-way motor driver
6. More powerful chip and more IO ports
7. Greatly optimized OLED display program, more convenient and flexible to use
8. The color sensor is removed from the list

## Specification

1. High Integration Density(14 function modules): atmospheric pressure, temperature, humidity, harmful gases, carbon dioxide, ultraviolet rays, light, sound, water quality, water temperature, soil humidity, buzzer, RGB light, OLED display, and motor driver.

- IoT Function: able to be accessible to famous IoT platforms such as IFTTT, ThingSpeak, and EasyIoT through the WIFI-IoT card.
- The interesting shape of the tree makes the classroom teaching more vivid and fun.

## Board Overview



## Parameters

**Power Supply:** MicroUSB (5V) / Three 1.5V AAA Batteries (4.5V)

Because the power consumption of WIFI-IoT card is relatively large, it may not work properly if the AAA battery is low. Please use MicroUSB port to supply power to ensure normal operation of the product.

**Digital Port Output Voltage:** 3.3V

**GPIO:** P0 P1 P2 P8 P12 P13 P14 P15 P16 I2C×2

### ML8511 UV Sensor

Operating Temperature: -20°C~70°C

Sensitive Area: UV-A, UV-B

Sensitivity Wavelength: 280-390nm

### BME280 Environment Sensor



Operating Current: 2mA

Operating Temperature: -40°C~+85°C

Temperature Measuring Range: -40°C~+85°C ,Resolution 0.1°C ,Deviation±0.5°C

Humidity Measuring Range: 0~100%RH, Resolution 0.1%RH , Deviation±2%RH

Response Time of Humidity Measurement: 1S

Atmospheric Pressure Measuring Range: 300~1100hPa

### **Waterproof Temperature Sensor**

Temperature Display Range: -10°C~+85°C ( Deviation±0.5°C )

Operating Temperature Range: -55°C~125°C

Query Time: less than 750ms

### **TDS Water Quality Sensor**

The TDS probe should not be used in water above 55°C.

The TDS probe should not be placed too close to the edge of the container, as this will affect the accuracy.

### **CCS811 Air Quality Sensor**

Operating Temperature Range: -40°C~85°C

Operating Humidity Range: 10%RH~95%RH

CO2 Measuring Range: 400ppm~8000ppm

TVOC Measuring Range: 0ppb~1100ppb

### **Capacitive Soil Humidity Sensor**

Operating Voltage: 3.3V-5.5V DC

Output Voltage: 0-3.0V DC

Connector: PH2.0-3P

### **RGB Light**

RGB Light Model: WS2812

Port: P15

### **Light Sensor**

Output Date Type: analog value

Data Range: 0-1023

### **Buzzer**

Dimension: 9mm in diameter

Model: passive buzzer

Port: P0

### **Sound Sensor**

Output Date Type: analog value

Data Range: 0-1023

### **OLED Display**

Dimension: 0.96"

Display Color: blue

Pixels: 128 × 64

Full-screen Lighting Consumption: about 22.75mA

### **Motor Driving**

Driving Mode: PWM

Compatible Motor: Low Power DC Motors such as N20 Gear Motor and 130 Gear Motor

### **WIFI IoT**

Wireless Mode: IEEE802.11b/g/n

Encryption Type: WPA WPA2/WPA2-PSK

Frequency: 2.4GHz

Built-in Protocol: TCP/IP protocol stack

Supported IoT Platform: EasyIoT, IFTTT, ThingSpeak, SIoT

Status Indicator:

Red: disconnected

Blue: connecting

Green: connected

**Dimension:** 196mm\*110.6mm/7.72\*4.35"

**Programming Platform:** MakeCode, Mind+

## Chapter2: MakeCode graphical programming

Here we assume that you have mastered how to use MakeCode to program the micro:bit board. Therefore, we mainly introduce the functions and programming methods of the McQueen Mechanic. The basic use of MakeCode will not be repeated.



### Link and Library

MakeCode Programming Platform: <https://makecode.microbit.org>





Natural and Science IoT Library :






[https://github.com/DFRobot/pxt-DFRobot\\_Environment\\_Science](https://github.com/DFRobot/pxt-DFRobot_Environment_Science)


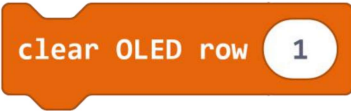

### Module Function Instruction




	<p><b>Request Data</b> <b>Instruction:</b> Request to read the sensor data once. This module is generally placed in the loop module to execute repeatedly to keep data up-to-date.</p>
	<p><b>Read the Ultraviolet Intensity</b> <b>Instruction:</b> Read the value of the ultraviolet intensity. The output value is of string type. If the value is required to participate in the programming, it should be transferred to the number type by the specific block first. <b>Data Type :</b> If the data is</p>

	<p>required to participate in the programming, transfer it to number first.</p>
<p>light level</p>	<p><b>Read the Ambient Light Intensity</b>  <b>Instruction:</b> Read the light intensity via the onboard light sensor,  <b>Data Type:</b> (Number )</p>
<p>water temperature(°C)</p>	<p><b>Read the Water Temperature</b>  <b>Instruction:</b> Read the water temperature in unit Celsius, correct to 1 decimal place. Plug the water temperature sensor into the board before using it.  <b>Data Type:</b> If the data is required to participate in the programming, transfer it to number first.</p>
<p>temperature(°C) ▼</p>	<p><b>Read the Environment Temperature</b>  <b>Instruction:</b> Read the Environment Temperature, correct to 1 decimal place.  <b>Data Type :</b> If the data is required to participate in the programming, transfer it to number first.</p>



	<p><b>Read the Environment Humidity</b></p> <p><b>Instruction :</b> Read the environment humidity in unit %, correct to 1 decimal place.</p> <p><b>Data Type :</b> If the data is required to participate in the programming, transfer it to number first.</p>
	<p><b>Read the Atmospheric Pressure</b></p> <p><b>Instruction:</b> Read the atmospheric pressure in unit kPa, correct to 1 decimal place.</p> <p><b>Data Type:</b> If the data is required to participate in the programming, transfer it to number first.</p>
	<p><b>Read the TDS Value</b></p> <p><b>Instruction:</b> Measure the TDS, and plug the TDS probe before using it.</p> <p><b>Data Type:</b> Number</p>
	<p><b>Set the K Value of the TDS</b></p> <p><b>Instruction:</b> If the TDS value is not accurate, this module can be used to adjust the</p>

	<p>measured value. This block is not obligatory.</p>
	<p><b>Set the K Value of the TVOC &amp; CO2</b></p> <p><b>Data Type:</b> If the TVOC and CO2 values are not accurate, this module can be used to adjust the measured value. This block is not obligatory.</p>
	<p><b>Read the CO2 Value</b></p> <p><b>Instruction:</b> Read the value of CO2 in the environment.</p> <p><b>Data Type:</b> Number</p>
	<p><b>Read the TVOC Value</b></p> <p><b>Instruction:</b> Read the value of TVOC in the environment.</p> <p><b>Data Type:</b> Number</p>
	<p><b>OLED Displays Characters at Specified Positions</b></p> <p><b>Instruction:</b> OLED Displays Characters at Specified Positions. The data type must be character. If it is number, it should be transferred to character first.</p>
	<p><b>OLED Displays number at Specified Positions</b></p> <p><b>Instruction:</b> OLED</p>

	<p>Displays number at Specified Positions. The data type must be number. If it is character, it should be transferred to number first.</p>
	<p><b>Clear the contents of the specified location of the OLED</b></p> <p><b>Instruction:</b> Clear the specified columns and rows in OLED. It is usually used to refresh the displayed content at a fixed location. A clear operation could cause the contents of the cleared location to blink at a fixed frequency.</p>
	<p><b>Clear the contents of the specified row of the OLED</b></p> <p><b>Instruction:</b> Clear the specified row in OLED. It is usually used to refresh the display in a row. A clear operation could cause the contents of the cleared location to blink at a fixed frequency.</p>
	<p><b>Control the Direction and Speed of the</b></p>

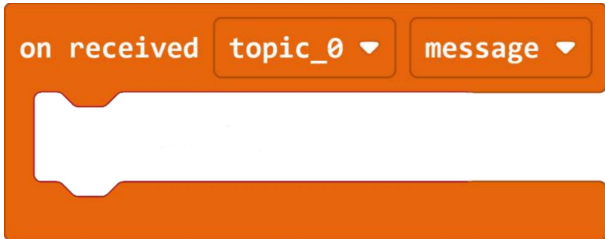
	<p><b>Motor's Rotation</b></p> <p><b>Instruction:</b> Control the direction and speed of rotation of the DC motor connected to the Motor port. The speed range is 0~255. When the speed is 0, the motor will not rotate.</p>
	<p><b>Control the Stop of the Motor</b></p> <p><b>Instruction:</b> Stop the DC motor connected to the Motor port.</p>
	<p><b>Adjust the Brightness of the RGB Light</b></p> <p><b>Instruction:</b> Set the brightness of the 4 RGB lights. The range is 0~255. During use, if the brightness of the RGB light is too high, you can embed this block into the "on start" block to lower the display brightness.</p>
	<p><b>Set the Three Primary Colors of The RGB Light</b></p> <p><b>Instruction:</b> Accurately adjust the color displayed by the RGB light by setting the values of the three primary colors of red, green, and blue.</p>





 <p>range from 1 with 4 leds</p>	<p><b>Specify the Range of Serial Numbers RGB Lights keep ON</b></p> <p><b>Instruction:</b> Specify the serial number range of the RGB LEDs turning ON. 1~4 is optional. For example, select 2-3, NO. 2 and NO.3 RGB LED will light up.</p>
 <p>RGB light 1 show color</p>	<p><b>Specify the Color of a specified RGB Light</b></p> <p><b>Instruction:</b> Specify the color of a specified RGB led in the 4 lights. 1-4 is optional. You can directly select the color block, or use the "Set Three Primary Colors of the RGB Lights" block.</p>
 <p>RGB show color</p>	<p><b>All RGB Lights Display the Same Color</b></p> <p><b>Instruction:</b> Set all the RGB LEDs light up in a specified color. You can directly select the color block, or use the "Set Three Primary Colors of the RGB Lights" block.</p>
 <p>set RGB show rainbow color from 1 to 360</p>	<p><b>Set the RGB Lights to Display a Rainbow Effect.</b></p> <p><b>Instruction:</b> Set 4 RGB lights to display with rainbow color effect. Fill</p>

	<p>in the degree of the color difference in the number box. The number range is 1 to 360. The larger the interval, the greater the difference. And different data intervals have different color effects.</p>
	<p><b>Turn off All RGB Lights</b>  <b>Instruction:</b> Turn off all the RGB lights.</p>
	<p><b>Configure the WIFI Parameters</b>  <b>Instruction:</b> Set the WIFI parameters. In one program, it only needs to be set once. Place it in "on start" block.  <b>Name:</b> Full in the WIFI name  <b>Password:</b> Full in the WIFI password</p>
	<p><b>MQTT Configuration</b>  <b>Instruction:</b> If the IoT platform uses MQTT protocol (EasyIoT, SloT and ect.), then this block is required for MQTT configuration.  <b>IOT_ID(ID):</b> Full in the ID of the IoT platform  <b>IOT_PWD(password):</b> Full in the password of</p>

	<p>the IoT platform</p> <p><b>Topic ( the default is topic_0 ):</b> Fill in the code generated in the Topic on the MQTT platform.</p> <p><b>Server Options:</b> EasyIoT_CN, Easy IoT_EN, SloT</p> <p><b>IP Address:</b> You need to click the "+" to check this content, Easy IOT_CN and Easy IOT_EN do not need to be filled in and modified, the default is okay. SloT needs to fill in the IP address correctly.</p>
 <p>The image shows a Scratch 'subscribe additional' block. It has a dropdown menu set to 'topic_0' and a text input field containing the string 'yourIotTopic'.</p>	<p><b>New MQTT Subscription</b></p> <p><b>Instruction:</b> On the IoT platform of MQTT protocol such as Easy IoT, it is possible to create multiple topics. Besides, different Topics can be selected for data operation. When using this block, we need to create a corresponding Topic on the IoT platform. We can create 5 Topics at most.</p> <p>Fill the text box with the code corresponding to the Topic column.</p> <p><b>Options:</b> topic_0,</p>

	<p>topic_1, topic_2, topic_3, topic_4</p>
 <p>The image shows a Scratch 'on received' block. The 'topic' dropdown menu is set to 'topic_0' and the 'message' dropdown menu is set to 'message'. The block is orange with a white text area.</p>	<p><b>The Program to Be Executed after Receiving Message Sent by MQTT Platform (Event Trigger Mode)</b></p> <p><b>Instruction:</b> Execute the program after receiving the message sent by MQTT IoT platform. This Block is an event-triggered mode. When the event is triggered, a string type data with variable name "message" will be received.</p> <p><b>Option:</b> topic_0, topic_1, topic_2, topic_3, topic_4</p>
 <p>The image shows a Scratch 'IFTTT configure event' block. The 'event' dropdown menu is set to 'yourEvent' and the 'key' dropdown menu is set to 'yourKey'. The block is orange with a white text area.</p>	<p><b>Configure the Event Name and Key of IFTTT</b></p> <p><b>Instruction:</b> First, you need to create an event on IFTTT, and then fill in the corresponding event name and key in this block.</p>
 <p>The image shows a Scratch 'IFTTT send' block. The 'value1' dropdown menu is set to 'Hi', the 'value2' dropdown menu is set to 'DFRobot', and the 'value3' dropdown menu is set to '2020'. The block is orange with a white text area.</p>	<p><b>Send A Message to IFTTT Platform</b></p> <p><b>Instruction:</b> Send string information to the IFTTT</p>

	platform, with a total of 3 values
 <p>ThingSpeak configure key: "yourKey"</p>	<p><b>Configure the ThingSpeak Key</b></p> <p><b>Instruction:</b> Fill in the key of ThingSpeak platform. The key need to be generated on the ThingSpeak platform first.</p>
 <p>ThingSpeak send value1: "2020" +</p>	<p><b>Send Message to ThingSpeak</b></p> <p><b>Instruction:</b> Send string information to ThingSpeak. You can click the "+" to send multiple string messages simultaneously.</p>

## 1. Measurement of Temperature and Humidity and The Comfort Level

### Introduction

The example codes below are used to read temperature and humidity values, and display the real-time temperature and humidity on the OLED. When the temperature and humidity exceed the comfort level of human body, the micro:bit LED screen will show a "x", and if the comfort level is within the range of human body, it will show a "heart".

### Sample Program: Measurement of Temperature and Humidity and Estimation of Comfort Level

**Program Link:** [https://makecode.microbit.org/\\_ifULzwcC8U4y](https://makecode.microbit.org/_ifULzwcC8U4y)

**Program Screenshot:**

```

forever
  request data
  set temperature to parse to number temperature(°C)
  set humidity to parse to number humidity(%)
  OLED from column 1 to 16 in row 2 display string join "Temp:" temperature(°C) "°C"
  OLED from column 1 to 16 in row 4 display string join "Humidity:" humidity(%) "%"
  if temperature > 30 or humidity > 80 then
    show icon [LEDs on]
  else
    show icon [LEDs off]

```

## 2. Measurement of Atmospheric Pressure

### Introduction

Read the value of atmospheric pressure and display it on the OLED in unit kPa.

### Sample Program: Atmospheric Pressure Detection

**Program Link:** [https://makecode.microbit.org/\\_7A5HkkUJdXsd](https://makecode.microbit.org/_7A5HkkUJdXsd)

### Program Screenshot:

```

forever
  request data
  OLED from column 1 to 16 in row 1 display string join pressure(kPa) "kPa"

```

## 3. Measurement of Sound Intensity

### Introduction

In this part, we are going to learn the measurement of sound intensity through two samples.

### Sample Program 1: Sound Dynamic Display

Convert the acquired sound to the information that how many LEDs of micro:bit panel would light up. The louder the sound is, the more LEDs will be turned on at the LED screen from left to right.

**Program Link:** [https://makecode.microbit.org/\\_Ky7VPWaY15wT](https://makecode.microbit.org/_Ky7VPWaY15wT)

### Program Screenshot:

```
forever
  request data
  OLED from column 1 to 16 in row 1 display string convert sound level to text
  clear OLED row 1
```

```
forever
  request data
  set N to 0
  clear screen
  repeat 5 times
    do
      set Sound to sound level
      plot x map Sound from low 0 high 500 to low 0 high 4 y N
      change N by 1
```

### Sample Program 2: Clapping Counter

Record clapping times(1~9) and display the corresponding number on the micro:bit LED Screen, meanwhile, display the sound intensity on the OLED display in real-time.

Program Link: [https://makecode.microbit.org/\\_0bdDDRv8pDXz](https://makecode.microbit.org/_0bdDDRv8pDXz)

### Program Screenshot:

```

on start
  set N to 0

forever
  request data
  OLED from column 1 to 16 in row 1 display string convert sound level to text
  clear OLED row 1

forever
  request data
  if sound level > 200 then
    change N by 1
    show number N
    if N > 8 then
      set N to 0
  
```

## 4. Measurement of Light Intensity

### Introduction

In this chapter, we are going to learn the measurement of light intensity value and make an interesting light-controlled clock.

### Sample Program: Interesting Light-Controlled Clock.

The program simulates a scenario in which sunshine hits the sensor in the morning and the loudspeaker plays music to remind you that it is time to get up. Then you press button A, the music stops, and the bedside lamp lights up. When you press button B, the lamp goes out and the program enters the next round of waiting.

**Program Link:** [https://makecode.microbit.org/\\_TsVbYTEXj6Vj](https://makecode.microbit.org/_TsVbYTEXj6Vj)

### Program Screenshot:

```

on button B pressed
  request data
  set N to 0
  clear all RGB

on button A pressed
  request data
  stop melody all
  set N to 1
  RGB light range from 1 with 4 leds show color

on start
  set N to 0
  set RGB brightness to 100

forever
  request data
  OLED from column 1 to 16 in row 1 display string convert light level to text
  clear OLED row 1
  if light level < 500 and N = 0 then
    set N to 1
    start melody birthday repeating once
  
```



## 5. Measurement of Ultraviolet Rays

### Introduction

Ultraviolet radiation, which is directly invisible to our eyes, is the general term for radiation of wavelengths in the electromagnetic spectrum from 10nm to 400nm. Excessive ultraviolet ray intensity can damage human's skin and harm health.

In this chapter, the program will use a UV sensor to measure the UV intensity and determine whether it exceeds the specified value. If so, the servo will rotate to simulate the electric curtain closing to block the UV.

### Sample Program: Measurement of Ultraviolet Rays and Automatic Curtain-Closing System

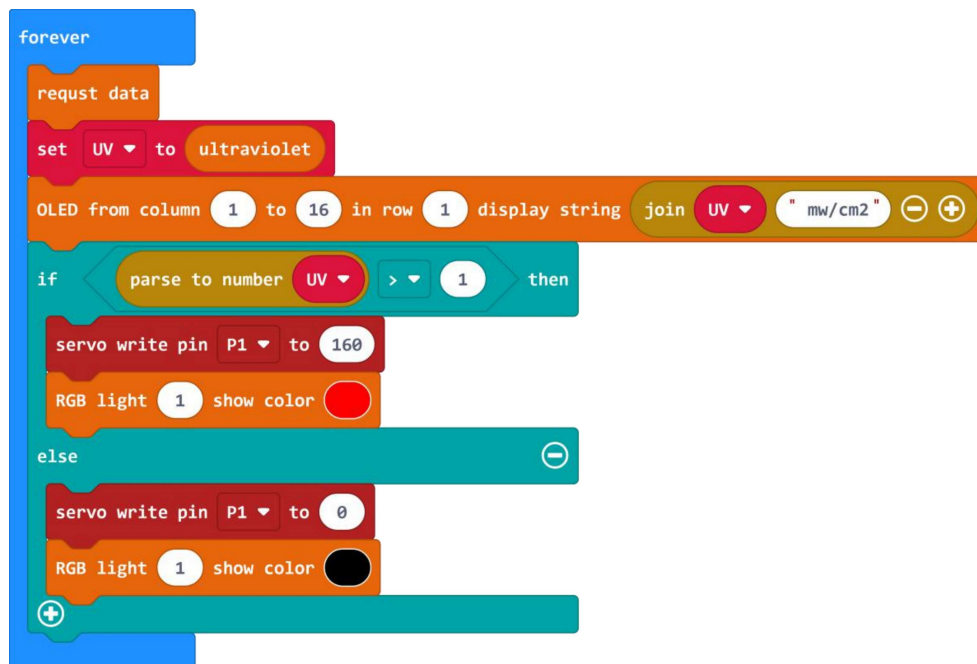
This program will read the UV intensity and display it on the OLED. The unit of UV intensity is mw/cm2. Besides, when the intensity reaches 1.0 or above, the servo will be activated to simulate closing the curtain.

Note: The servo in this sample should be purchased separately.

Link: <https://www.dfrobot.com/product-1338.html>

Program Link: [https://makecode.microbit.org/\\_gDAaf5VpX1Ht](https://makecode.microbit.org/_gDAaf5VpX1Ht)

### Program Screenshot:



## 6. Measurement of TVOC

### Introduction

TVOC refers to the organic compounds whose saturated vapor pressure exceeds 133.32 Pa at room temperature. Its boiling point is between 50 to 250°C at room temperature, and it exists in the air in the form of evaporation. Its toxicity, irritation, carcinogenicity and special odor, will affect the skin and mucous membrane, and produce acute damage to human body.

TVOC Reference Value:

TVOC Concentration ( ppd)	Human Physiological Reaction
<50	Normal
50-750	May be irritable
750-6000	Uncomfortable and headaches may occur
>6000	Headaches and other neurological problems

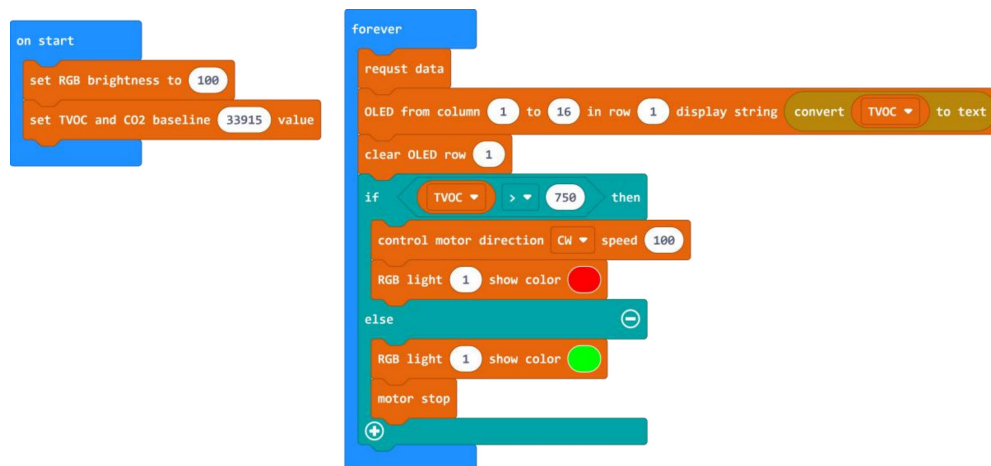
### Sample Program: TVOC Gauge

This program will read the TVOC value and display it on the OLED. When it exceeds 750, the program will turn on the motor to ventilate the room.

Note: the motor in this program should be purchased separately.

Program Link: [https://makecode.microbit.org/\\_Dm6PbThoYT77](https://makecode.microbit.org/_Dm6PbThoYT77)

### Program Screenshot:



## 7. Measurement of CO2

### Introduction

Carbon dioxide is a colorless and odorless gas at room temperature. It is denser than air and can dissolve in water. The chemical formula is CO<sub>2</sub> and it is one of the main components of the air.

Too much carbon dioxide in the air can cause breathing difficulties and even carbon dioxide poisoning.

CO<sub>2</sub> Reference Value:

<b>CO<sub>2</sub> Concentration</b>	<b>Human Physiological Reaction</b>
<500	Normal
500-1000	Feel the air cloudy
1000-2500	Feel sleepy
2500-5000	Bad for health
>5000	Rick of poisoning

#### **Sample Program: Carbon Dioxide Tester**

This program will read the value of carbon dioxide and display it on the OLED. When the value is below 500, the RGB light shows green; 500-1000 shows yellow; 1000-2500 shows orange; 2500-5000 shows red; above 5000 shows purple. Because the gas we breathe out also contains a lot of carbon dioxide, we can blow to the sensor to observe the significant change of the value.

**Program Link:** [https://makecode.microbit.org/\\_6JmgDeKRTEMT](https://makecode.microbit.org/_6JmgDeKRTEMT)

**Program Screenshot:**

```

on start
  set RGB brightness to 100

forever
  request data
  OLED from column 1 to 16 in row 1 display string convert CO2 to text
  clear OLED row 1
  if CO2 < 500 then
    RGB light range from 1 with 4 leds show color green
  if CO2 > 500 and CO2 < 1000 then
    RGB light range from 1 with 4 leds show color yellow
  if CO2 > 1000 and CO2 < 2500 then
    RGB light range from 1 with 4 leds show color orange
  if CO2 > 2500 and CO2 < 5000 then
    RGB light range from 1 with 4 leds show color red
  if CO2 > 5000 then
    RGB light range from 1 with 4 leds show color magenta

```

## 8. Measurement of Water Temperature and Application (Conversion between Celsius and Fahrenheit)

### Introduction

In daily life, many devices need to measure the water temperature, such as water heater, coffee maker, smart cup, etc. in this chapter, we will use a sensor to measure the water temperature: Water Temperature Sensor.

In the program, the default output temperature unit is Celsius. Another unit called Fahrenheit is also widely used. the conversion formula between them is:  $Fahrenheit = 32 + Celsius * 1.8$

### Sample Program: Water Temperature Detector

Write a program to read the water temperature and convert it to Fahrenheit. Then display the value of two units in the OLED.

**Program Link:** [https://makecode.microbit.org/\\_1YyJtubr6YJx](https://makecode.microbit.org/_1YyJtubr6YJx)

**Program Screenshot:**

```

forever
  request data
  set C to parse to number water temperature(°C)
  set F to truncate C x 1.8 + 32
  OLED from column 1 to 5 in row 1 display number C
  OLED from column 1 to 5 in row 2 display number F
  OLED from column 8 to 9 in row 1 display string "C"
  OLED from column 8 to 9 in row 2 display string "F"
  clear OLED from column 1 to 5 in row 2

```

## 9. Measurement of Water Quality (TDS)

### Introduction

TDS refers to the total dissolved solids. The unit is mg/L, which indicates how many milligrams of the soluble solid are dissolved in 1 liter of water. The higher the TDS value is, the more dissolved substance there is in the water.

TDS value partly reflects the purity of the water and grades the water quality. The lower the TDS value, the purer the water; the higher the TDS value, and the more soluble solids in the water. However, it cannot be judged from this that the water with a high TDS value is harmful.

### Sample Program: TDS Water Quality Monitor

Monitor the water quality of different water sources and classify the water quality. TDS value below 20 is considered as pure water, 20-200 as tap water, and above 200 as sewage. The OLED will show the classification and the different colors of the RGB lights represent different grades.

**Program Link:** [https://makecode.microbit.org/\\_7k2Xx4MJLsk](https://makecode.microbit.org/_7k2Xx4MJLsk)

**Program Screenshot:**

```

forever
  request data
  set TDS to TDS
  OLED from column 1 to 5 in row 1 display number TDS
  if TDS < 20 then
    RGB light range from 1 with 4 leds show color green
    OLED from column 1 to 16 in row 2 display string "Purified water"
  +
  if TDS > 10 and TDS < 200 then
    RGB light range from 1 with 4 leds show color yellow
    OLED from column 1 to 16 in row 2 display string "Tap water"
  +
  if TDS > 200 then
    RGB light range from 1 with 4 leds show color red
    OLED from column 1 to 16 in row 2 display string "sewage"
  +
  clear OLED from column 1 to 5 in row 1
  clear OLED row 1

```

## Chapter3: Applications of IoT Platform

In this chapter, we will use WIFI-IoT card, and connect to the IoT platform via WIFI. Due to the high consumption of WIFI-IoT card, if the power of AAA battery is low, The WIFI will not work well. Please use USB port to supply power.

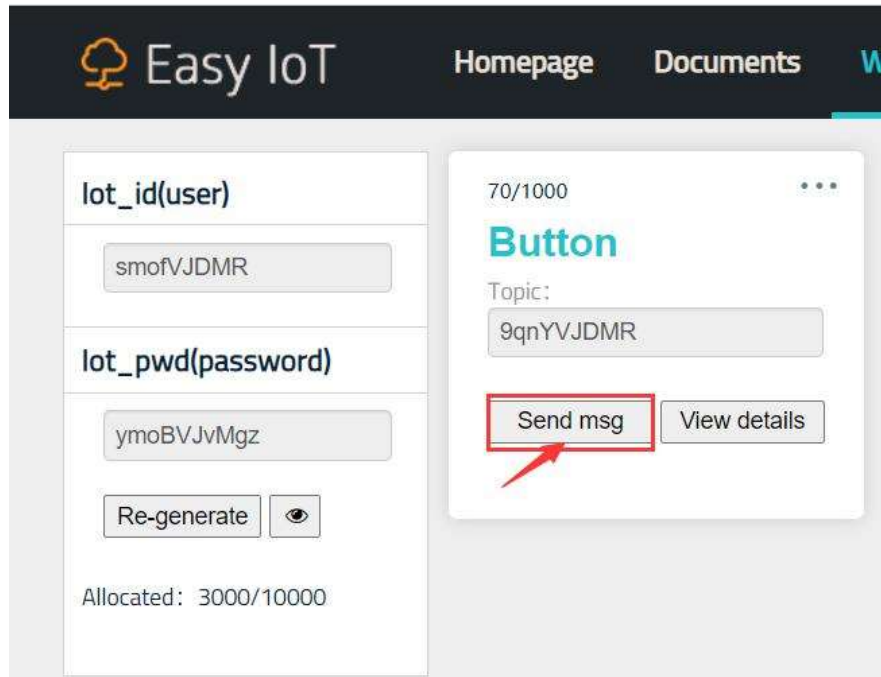
There are three states of the WIFI-IoT card indicator: red means WIFI disconnected; blue means WIFI connecting; green means connected successfully. Only when WIFI connection is successful, can we connect to and communicate with the IoT platform.

### Example Project 1: IoT Environment Detection System (Based on Easy IoT)

Based on Easy IoT platform, the project saves the temperature and humidity data in the IoT platform via WIFI. When the digital instruction "1" is sent to the Easy IoT platform, the temperature and humidity value will be read and uploaded to Easy IoT platform via WIFI. If you need to read other values of other sensors, such as UV, harmful gases, etc., the principle and method are the same.

#### Settings of Easy IoT Platform:

1. Open the Easy IoT official website: <http://iot.dfrobot.com/>
2. Sign up and log in the Easy IoT platform. Three sequence number will be generated automatically: ID, PWD, and Topic. Fill in the corresponding number in the program.



3. Click "Send msg" to enter the interface. Input number "1" in the box, and click "Send".

### Button



4. Click "View details", enter the detailed interface and you can see the data of temperature and humidity values returned.

Latest news	
Time	Message
2021/1/4 8:53:41	28.2
2021/1/4 8:53:32	51.4
2021/1/4 8:53:22	1

**Program Link:** [https://makecode.microbit.org/\\_Woh7HuMdD6hz](https://makecode.microbit.org/_Woh7HuMdD6hz)

**Program Screenshot:**

```

on start
  Wi-Fi configure name: "dfrobotOffice" password: "dfrobot2011" start connection
  MQTT configure
  IOT_ID(user): "3g6H14GZg"
  IOT_PHD(password): "qR6H1VGWgz"
  Topic(default topic_0): "cYEDJVMWR"
  server: EasyIOT_CN
  set N to 0

on received topic_0 message
  set M to parse to number message
  if M = 1 then
    request data
    send message temperature(°C) to topic_0
    send message humidity(%) to topic_0
  
```

## Sample Project 2: Light-Sensitive Automatic IoT Clock (Based on Easy IoT)

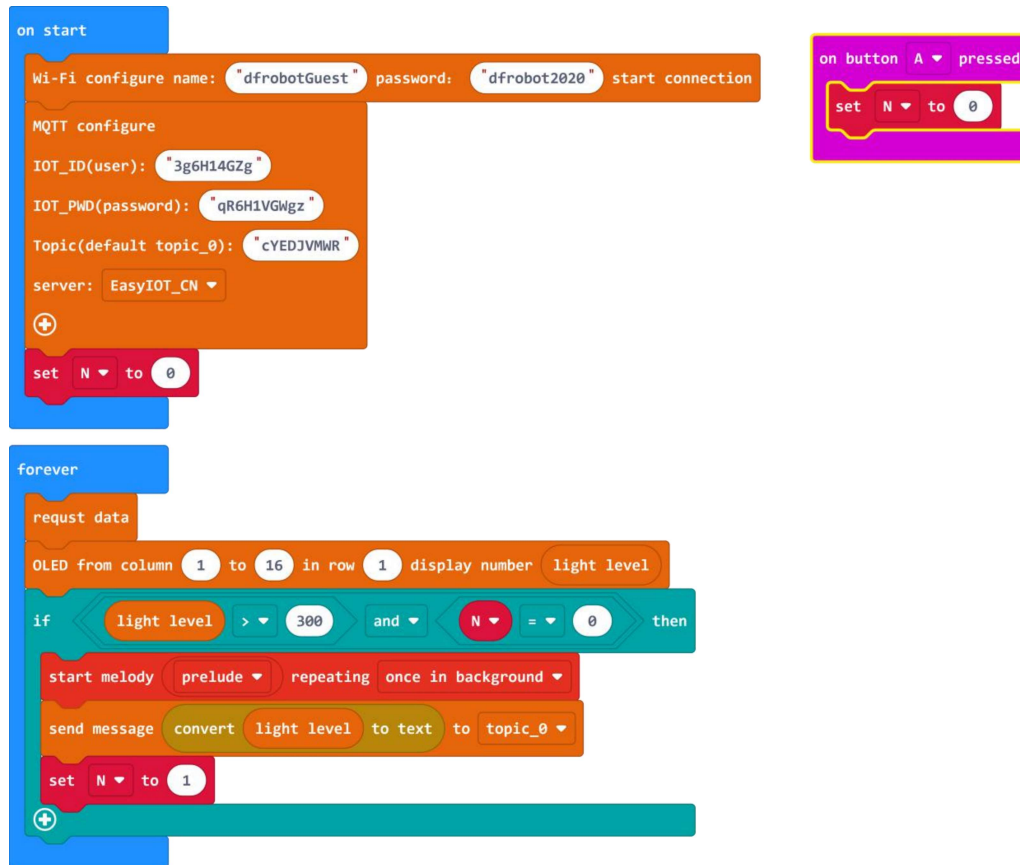
### Introduction:

Sunshine hits the sensor in the morning and the loudspeaker plays music. The light intensity value will be upload to Easy IoT. Press button A on the main board, the program will enter the next round of waiting.



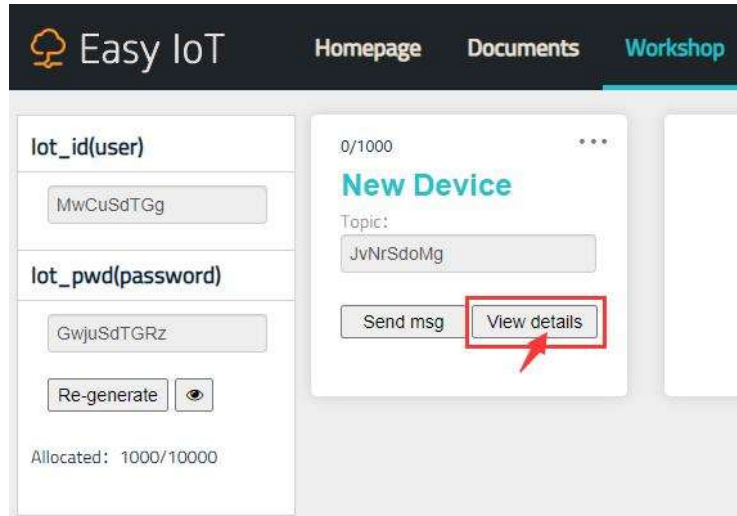
Program Link: [https://makecode.microbit.org/\\_Eda4TbaAFXqg](https://makecode.microbit.org/_Eda4TbaAFXqg)


### Program Screenshot:



### Program Execution Result:

- 1) When you shine a flashlight at the sensor, the buzzer will play music and the current value will be sent to the Easy IoT platform via WIFI.
- 2) Click "View details" at the position that the following figure shows to enter the detailed website to view the received data.



 **Query result**

Time	Message	Operate
2020/11/16 11:54:52	32	
2020/11/16 11:54:52	32	

3) Press button A on the main board, the program will restore the waiting state, wait for the light to reach a certain level before playing music and sending data again.

4) Refresh the page on the IoT platform to view the latest data.

### Sample Program 3: IoT Greenhouse Environment Alarm (Based on IFTTT)

#### 1-1 Introduction

This project simulates the environment monitoring system of the greenhouse. After starting up, the system will automatically detect the temperature, humidity and soil humidity in the current greenhouse, and display them on the OLED. If the values exceed a certain range, the data will be sent to the mailbox set before via IFTTT platform.

#### 1-2 Settings of Sending Message from IFTTT End

Open the IFTTT official website <https://ifttt.com/> click sign in to log in.

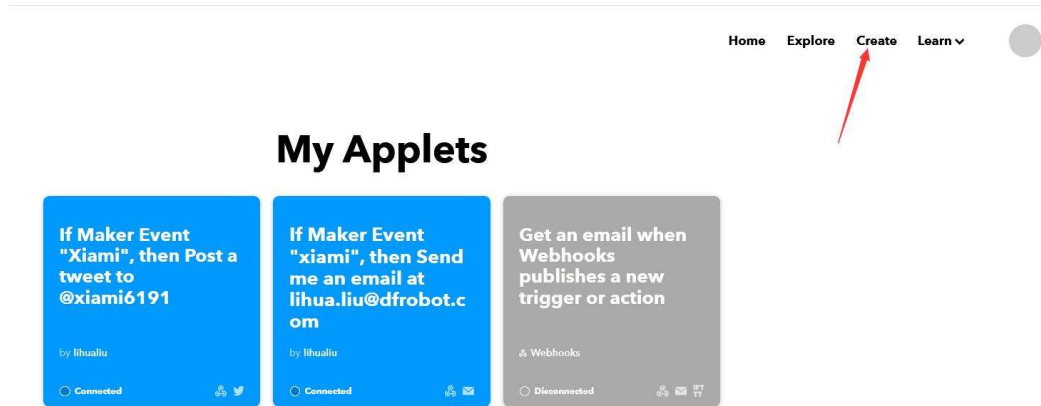
# Make your work more productive

Enter your email  **Get started**

or

Continue with Apple Continue with Google Continue with Facebook

After logging in, operate as the following picture shows



Click "Create your own" and you will see the interface like this:



# If This Then That

**Build your own service**

Click "This" to configure it and search "webhooks" in the box.

## Choose a service

Step 1 of 6



## Choose a service

Step 1 of 6



If it is the first time, the page will display as shown below. Click "Connect", and choose "Receive a web request", fill in the "Event Name" to finish the creation of "This".



## Connect Webhooks

Step 1 of 6

Integrate other services on IFTTT with your DIY projects. You can create Applets that work with any device or app that can make or receive a web request. If you'd like to build your own service and Applets, check out the IFTTT platform.

**Connect**

Click "Receive a web request"

< Back

## Choose trigger

Step 2 of 6

### 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)

Don't see what you're looking for?

Suggest a new trigger

Fill out the "Event Name" to finish the creation of "This".



## Complete trigger fields

Step 2 of 6

Event Name

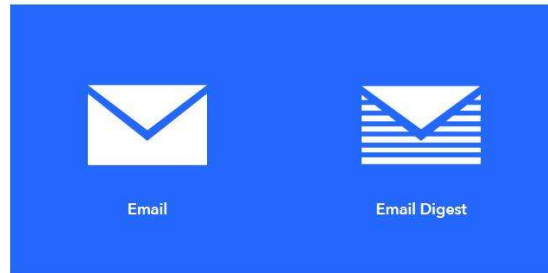
The name of the event, like "button\_pressed" or "front\_door\_opened"

Create trigger

When finished, it will return automatically. Then click "That" to configure it, search "Email" and click the "Email" on the left.

## Choose action service

Step 3 of 6



Click "Connect" and fill your email address in "Email address" box. Click "Send PIN" to send a PIN code to your own email.



## Connect Email

Step 3 of 6

Send and receive important information when you need it, automatically, with this service. The "send me an email" action has a daily limit of 750 messages per day, at which point Applets will be paused until the limit resets at 12:00 AM GMT.

**Connect**



## Connect Email

Enter the email address you would like to use for all of your Email Applets.

Email address

**Send PIN**

Check the PIN code in your email, fill it in and click "Connect".



## Connect Email

Enter the email address you would like to use for all of your Email Applets.

Email address

lihua.liu@dfrobot.com

Please enter the 4-digit PIN you received below.  
PIN

8823

Connect

Retry

Enter the mailbox setting step, click "Send me an email"



## Choose action

Step 4 of 6

### Send me an email

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

Don't see what you're looking for?

Suggest a new action

In this interface, it is to write the content of the email that needs to be sent to you. Here we directly choose the default. Then click "Create action" to complete the creation.



## Complete action fields

Step 5 of 6

Subject

The event named "**EventName**" occurred on the Maker Webhooks service

Add ingredient

Body

What: **EventName** <br>  
When: **OccurredAt** <br>  
Extra Data: **Value1**, **Value2**, **Value3**,

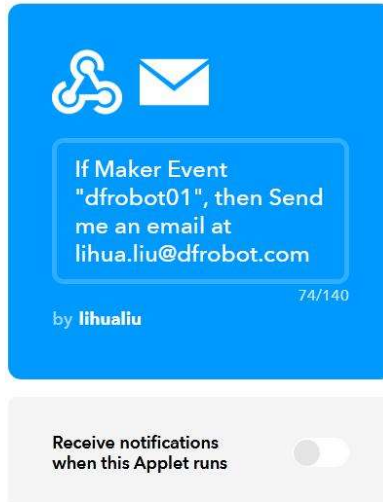
Add ingredient

Create action

Click "Create action" to enter the following interface:

## Review and finish

Step 6 of 6

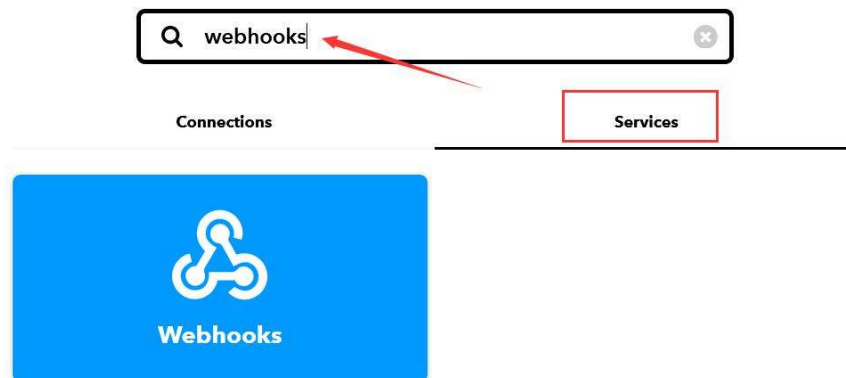


The interface shows a blue card with a Maker Event icon and an envelope icon. The text on the card reads: "If Maker Event 'dfrobot01', then Send me an email at lihua.liu@dfrobot.com". Below the card, it says "by lihualiu" and "74/140". At the bottom, there is a toggle switch for "Receive notifications when this Applet runs", which is currently turned off.

**Finish**

View the key: Click the "Explore" on the upper right corner, search "webhooks" in that page and switch to "Services", then click "Webhooks" icon to enter

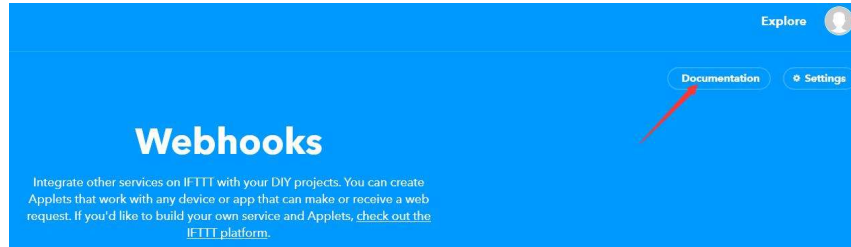
## Explore



The interface shows a search bar with the text "webhooks" and a red arrow pointing to it. Below the search bar, there are two tabs: "Connections" and "Services". The "Services" tab is selected and highlighted with a red box. Below the tabs, there is a blue card with a Maker Event icon and the text "Webhooks".

After entering webhooks, click "Documentation" to view the key, as shown below:





Your key is: **cjDOSJrDqYwwSTWTMZVtG4**

◀ Back to service

### To trigger an Event

Make a POST or GET web request to:

```
https://maker.ifttt.com/trigger/{event}/with/key/cjDOSJrDqYwwSTWTMZVtG4
```

With an optional JSON body of:

```
{ "value1" : "[ ]", "value2" : "[ ]", "value3" : "[ ]" }
```

The data is completely optional, and you can also pass `value1`, `value2`, and `value3` as query parameters or form variables. This content will be passed on to the Action in your Recipe.

You can also try it with `curl` from a command line.

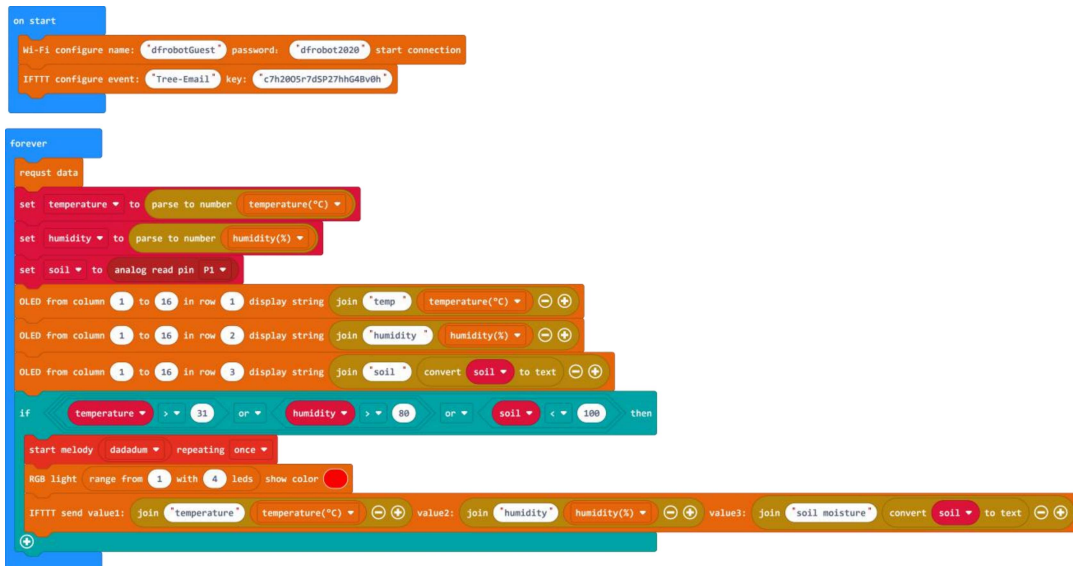
Test

### 1-3 Program Link

[https://makecode.microbit.org/\\_b3kUxjRTfeU](https://makecode.microbit.org/_b3kUxjRTfeU)

Note: when in use, please modify the WIFI configuration in the program as well as the ID and key in IFTTT to yours, so that it can be used.

### 1-4 Program Screenshot



### 1-5 Program Execution Result

After downloading the program, turn on the power, and the sensor will start to detect the current temperature, humidity, soil humidity, and other values related to the greenhouse environment. All the values will be displayed on the LOED. When the temperature exceeds 31°C, or the humidity exceeds 80%, or the soil humidity is below 100, a message will be sent to your default mailbox via WIFI-IoT and IFTTT platform. All the RGB lights will turn on in red and the buzzer will buzz. The mail you received will be similar like the one shown below:



## Sample Project 4: IoT Sunshine Intensity Recorder (Based on ThingSpeak)

### Introduction

This project will read the local light and UV intensity at regular intervals, and upload the data to ThingSpeak and generate a curve graph. Through the analysis of the graph, we can basically inform the relationship between sunlight and UV intensity.

Note: The clock module in this project should be purchased separately.



Link: <https://www.dfrobot.com.cn/goods-535.html>

MakeCode Library of Clock Module:

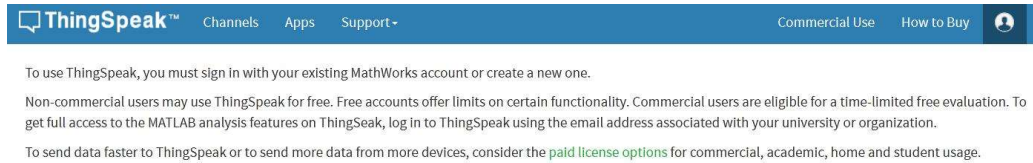
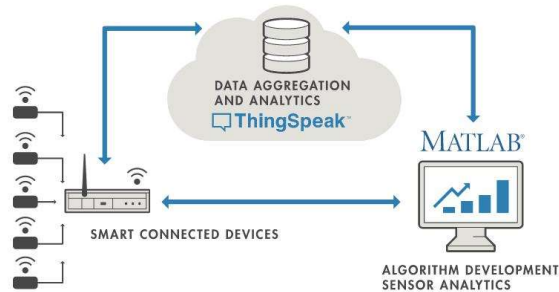
[https://github.com/tangjie133/pxt-DFRobot\\_SD1307](https://github.com/tangjie133/pxt-DFRobot_SD1307)

## Operation Steps

1) Open the ThingSpeak website: <https://thingspeak.com/> and click the profile at the upper right corner to enter the sign in/up interface.



2) Sign up in the following interface for the first use. You can sign up after filling the correct email address and verifying it.

The image shows a MathWorks sign-up form. It includes the MathWorks logo at the top left. Below the logo is a text input field labeled 'Email'. Underneath the input field, there is a link that says 'No account? Create one!' and a line of text that reads 'By signing in you agree to our privacy policy.' At the bottom right of the form is a blue button labeled 'Next'.

To use ThingSpeak, you must sign in with your existing MathWorks account or create a new one.

Non-commercial users may use ThingSpeak for free. Free accounts offer limits on certain functionality. Commercial users are eligible for a time-limited free evaluation. To get full access to the MATLAB analysis features on ThingSpeak, log in to ThingSpeak using the email address associated with your university or organization.

To send data faster to ThingSpeak or to send more data from more devices, consider the [paid license options](#) for commercial, academic, home and student usage.

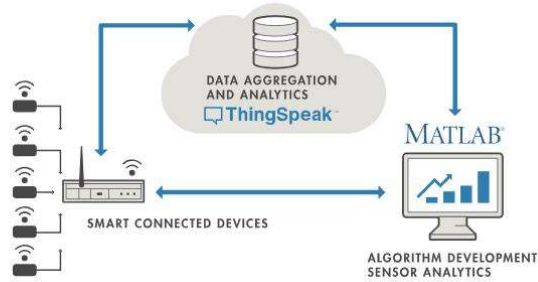
**Create MathWorks Account**

Email Address  
  
Missing required information  
To access your organization's MATLAB license, use your school or work email.

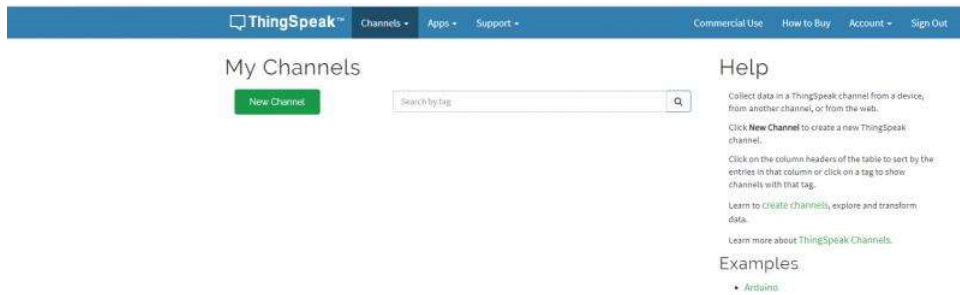
Location  
 Armenia

First Name  
  
Missing required information

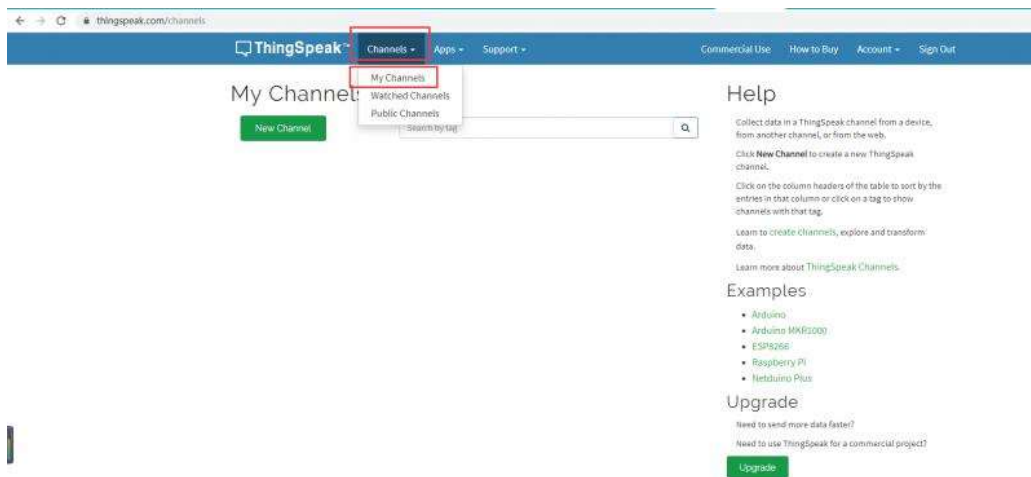
Last Name  
  
Missing required information



3) After signing up and logging in, you will find the following interface:



4) If not, you can click "Channels", there will be a drop-down menu. The above interface can also be found by Clicking "My Channels".



5) After the previous steps, we can create the channels. Click “New Channels”, you can see the following interface.

The screenshot shows the 'New Channel' form on the ThingSpeak website. The form has the following fields:

- Name:** A text input field.
- Description:** A text area.
- Field 1:** A text input field with 'Field Label 1' and a checked checkbox.
- Field 2:** A text input field with an unchecked checkbox.
- Field 3:** A text input field with an unchecked checkbox.
- Field 4:** A text input field with an unchecked checkbox.
- Field 5:** A text input field with an unchecked checkbox.
- Field 6:** A text input field with an unchecked checkbox.
- Field 7:** A text input field with an unchecked checkbox.
- Field 8:** A text input field with an unchecked checkbox.
- Metadata:** A text area.
- Tags:** A text area with a note '(Tags are comma separated)'. Below it is a 'Link to External Site' field with 'http://' and a 'Link to GitHub' field with 'https://github.com/'.
- Elevation:** A text input field.
- Show Channel Location:** A checked checkbox. Below it is a 'Latitude' field with '0.0'.

The 'Help' section on the right contains the following information:

**Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.**

**Channel Settings**

- Channel Name:** Enter a unique name for the ThingSpeak channel.
- Description:** Enter a description of the ThingSpeak channel.
- Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- Tags:** Enter keywords that identify the channel. Separate tags with commas.
- Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- Show Channel Location:**
  - Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
  - Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.
  - Elevation:** Specify the elevation position meters. For example, the elevation of the city of London is 35.052.
- Video URL:** If you have a YouTube™ or Vimeo® video that displays your channel information, specify the full path of the video URL.
- Link to GitHub:** If you store your ThingSpeak code on GitHub®, specify the GitHub repository URL.

**Using the Channel**

You can get data into a channel from a device, website, or another ThingSpeak channel. You can then visualize data and transform it using ThingSpeak Apps.

See Tutorial: ThingSpeak and MATLAB for an example of measuring dew point from a weather station that acquires data from an Arduino® device.

[Learn More](#)

6) Here we need to fill the name, Description, and tick the number of fields. In the MakeCode library, the field is the same as that in channels, so you just need to tick the corresponding numbers. In this project, we only upload the light intensity to the channel, so just tick one. The name of the field also can be customized, here we name it as “light level”. Once you have done that, you can save the channels. Here is the channel I created.

The screenshot shows the 'New Channel' form with the following filled-in data:

- Name:** Data reception
- Description:** Receive sensor data
- Field 1:** light level (checkbox checked)
- Field 2:** (checkbox unchecked)
- Field 3:** (checkbox unchecked)
- Field 4:** (checkbox unchecked)
- Field 5:** (checkbox unchecked)
- Field 6:** (checkbox unchecked)
- Field 7:** (checkbox unchecked)
- Field 8:** (checkbox unchecked)
- Metadata:** (empty)
- Tags:** (empty)

The 'Help' section on the right contains the following information:

**Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.**

**Channel Settings**

- Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete you channel.
- Channel Name:** Enter a unique name for the ThingSpeak channel.
- Description:** Enter a description of the ThingSpeak channel.
- Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- Tags:** Enter keywords that identify the channel. Separate tags with commas.
- Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- Show Channel Location:**
  - Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
  - Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.
  - Elevation:** Specify the elevation position meters. For example, the elevation of the city of London is 35.052.

7) After saving, the channel is created. The channel created is as follows:

## Data reception

Channel ID: **1082418** Receive sensor data  
Author: jietang133  
Access: Private

Private View **Public View** Channel Settings Sharing API Keys Data Import / Export

Add Visualizations  Add Widgets  Export recent data

### Channel Stats

Created: [about a minute ago](#)  
Entries: 0



8) If you want to upload more data, click "Channel Settings", and then tick the "box" behind the "Field" and save. You can also delete channel or clear data in this interface.



## Data reception

Channel ID: 1082418  
 Author: jietang133  
 Access: Private

Receive sensor data

Private View Public View Channel Settings Sharing API Keys Data Import / Export

### Channel Settings

Percentage complete: 50%

Channel ID: 1082418

Name: Data reception

Description: Receive sensor data

Field 1: light level

Field 2:

Field 3:

Field 4:

Field 5:

Field 6:

Field 7:

Field 8:

Metadata:

Tags:   
(Tags are comma separated)

Link to External Site:

Link to GitHub:

Elevation:

Show Channel Location:

Latitude:

Longitude:

Show Video:

YouTube  
 Vimeo

Video URL:

Show Status:

[Save Channel](#)

Want to clear all feed data from this Channel?

[Clear Channel](#)

Want to delete this Channel?

[Delete Channel](#)

### Help

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

#### Channel Settings

- Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- Channel Name:** Enter a unique name for the ThingSpeak channel.
- Description:** Enter a description of the ThingSpeak channel.
- Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- Tags:** Enter keywords that identify the channel. Separate tags with commas.
- Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- Show Channel Location:**
  - Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
  - Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.
  - Elevation:** Specify the elevation position meters. For example, the elevation of the city of London is 35.052.
- Video URL:** If you have a YouTube® or Vimeo® video that displays your channel information, specify the full path of the video URL.
- Link to GitHub:** If you store your ThingSpeak code on GitHub®, specify the GitHub repository URL.

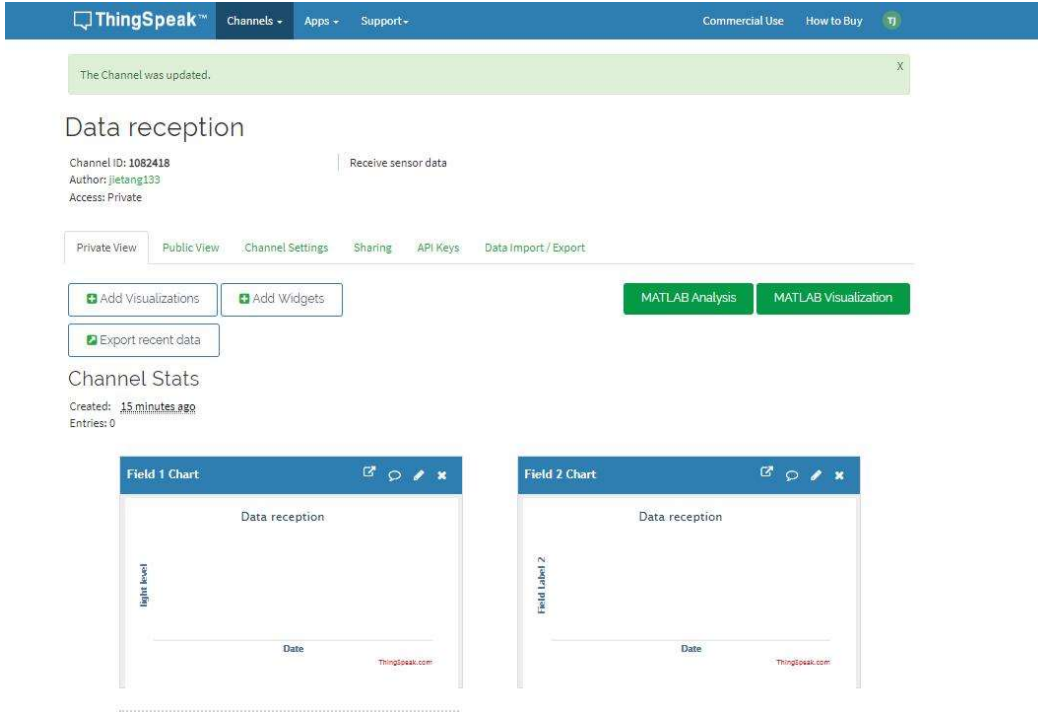
#### Using the Channel

You can get data into a channel from a device, website, or another ThingsSpeak channel. You can then visualize data and transform it using ThingSpeak Apps.

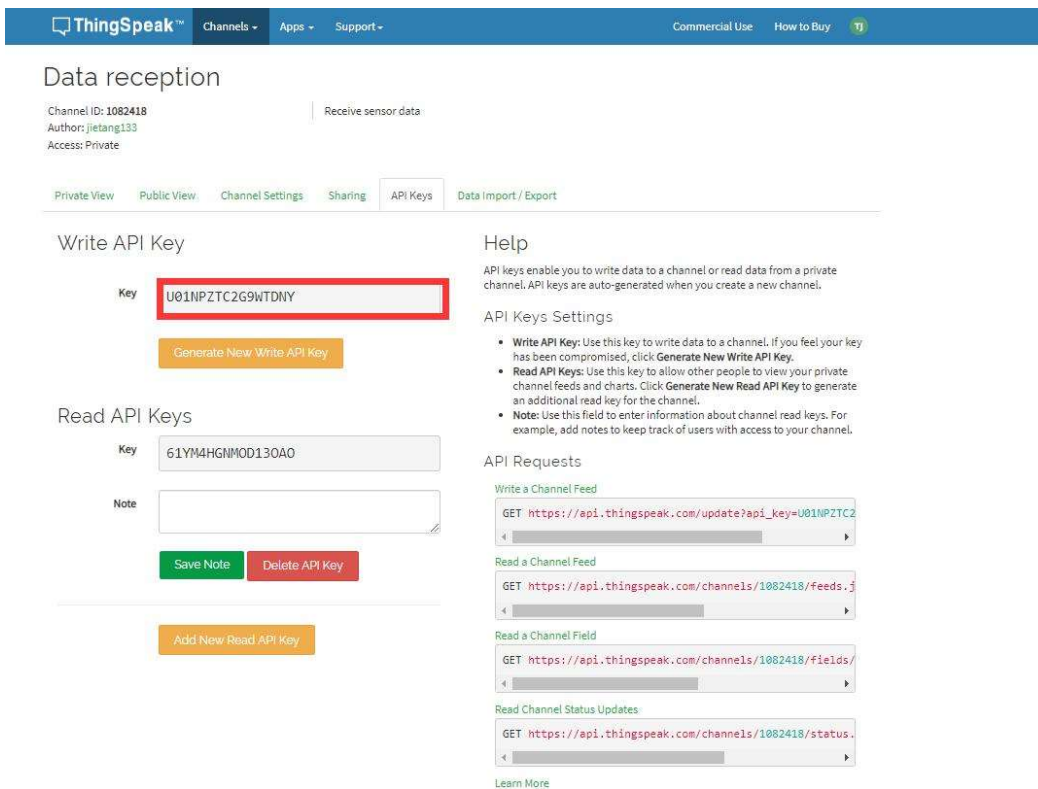
See [Get Started with ThingSpeak®](#) for an example of measuring dew point from a weather station that acquires data from an Arduino® device.

[Learn More](#)

The effect after adding channels:



9) Click "API Keys" and you can check the key of the corresponding channel. The key in the red box will be used in the program below.



**Program Link:** [https://makecode.microbit.org/\\_LdsAUxFKHH4h](https://makecode.microbit.org/_LdsAUxFKHH4h)

**Program Screenshot:**



```

on start
  Wi-Fi configure name: "dfrobotGuest" password: "dfrobot2020" start connection
  Thingspeak configure key: "U01NPZTC2G9WTD9Y"
  clock begins
  set 2020 year 7 month 31 day 5 weekday 23 hour 59 minute 0 second
  clear OLED row 1
  set beat to 1

forever
  if beat = get SD1307 time Second then
    change beat by 1
    if beat = 59 then
      set beat to 1
      clear OLED from column 3 to 3 in row 1
      clear OLED from column 6 to 6 in row 1
    else
      OLED from column 3 to 3 in row 1 display string ':'
      OLED from column 6 to 6 in row 1 display string ':'
    if get SD1307 time Hour < 10 then
      OLED from column 1 to 1 in row 1 display number 0
      OLED from column 2 to 2 in row 1 display number get SD1307 time Hour
    else
      OLED from column 1 to 2 in row 1 display number get SD1307 time Hour
    if get SD1307 time Minute < 10 then
      OLED from column 4 to 4 in row 1 display number 0
      OLED from column 5 to 5 in row 1 display number get SD1307 time Minute
    else
      OLED from column 4 to 5 in row 1 display number get SD1307 time Minute
    if get SD1307 time Second < 10 then
      OLED from column 7 to 7 in row 1 display number 0
      OLED from column 8 to 8 in row 1 display number get SD1307 time Second
    else
      OLED from column 7 to 8 in row 1 display number get SD1307 time Second

forever
  request data
  if if time Second equals 0 then
    clear OLED row 1
    Thingspeak send value1: ultraviolet value2: temperature(°C)
    RGB show color red
    pause (ms) 1000
    RGB show color black

```

**Program Execution Result:**

After the program is executed, the IoT platform will automatically generate a curve graph based on the date uploaded at regular intervals. In this case, the data will be uploaded every 5 seconds, and the length of time can be adjusted according to the actual situation.

## Data reception

Channel ID: **1082418** Receive sensor data  
Author: jietang133  
Access: Private

Private View Public View Channel Settings Sharing API Keys Data Import / Export

[+ Add Visualizations](#) [+ Add Widgets](#) [Export recent data](#)

[MATLAB Analysis](#) [MATLAB Visualization](#)

### Channel Stats

Created: [25 minutes ago](#)  
Last entry: [less than a minute ago](#)  
Entries: 15

