# 1. iot:kit Introduction


## 1.1. micro:bit Smart Science IoT kit

ELECFREAKS Smart Science IoT Kit is developed based on IoT:Bit,a new breakout board compatible with science sensors such as theultrasonic sensor, dust sensor, light sensor and water level sensor,.including RTC Timing and WIFI module. You can gather data via thesensors and send data to the cloud with more stable and accuratedata analysis.

## 1.2. pack list

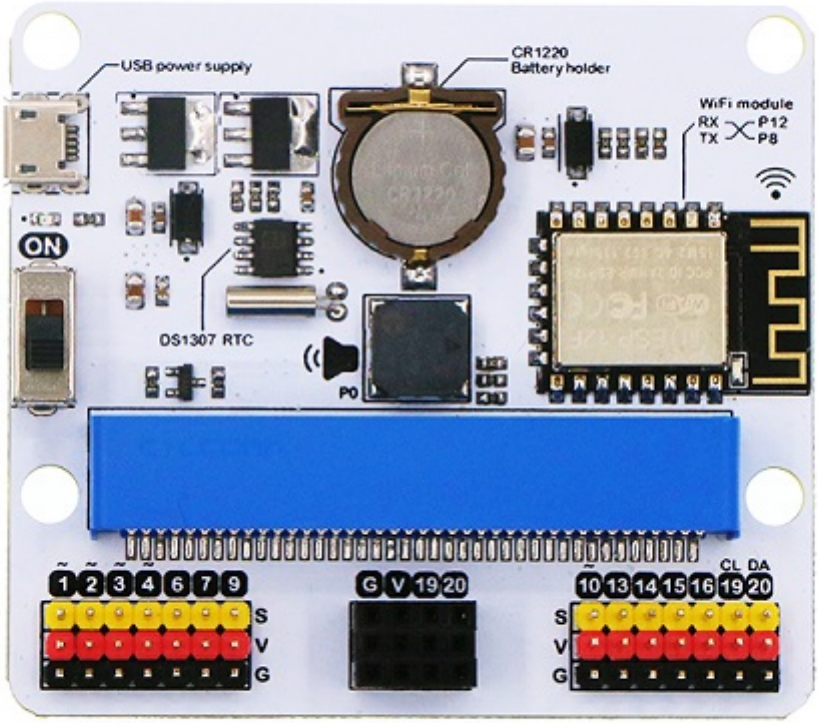
| Component | number |
|---|---|
| micro:bit | option |
| IoT:bit | 1 |
| OLED | 1 |
| BME 280 | 1 |
| PIR sensor | 1 |
| Light sensor | 1 |
| Noise sensor | 1 |
| Water level sensor | 1 |
| soil moisture sensor | 1 |
| Sonar:bit | 1 |

| Component | number |
| --- | --- |
| Dust sensor | 1 |
| 180° servo | 1 |

# 2. iot:bit

## 2.1. Introduction

- IoT:bit is an expansion board based on IoT for micro:bit, it can support the sensors in 3-pins, the actuator and the buzzer, it can also give a specific time telling even the micro:bit powers off with the RTC onboard.
- 
- It uses ESP8266 as WIFI expansion board and serial port to communicate with micro:bit. It also has extended all available IO port of the micro:bit which is led by GVS and you can extent various 3V E-blocks as LED, photosensitive and servo by using it. At the same time, the IOT:bit with an on-board buzzer for outside sound and an on-board RTC clock for timing without power supply.The codebase for makecode can connect to Thingspeak conveniently and create your IoT project quickly.




### Features

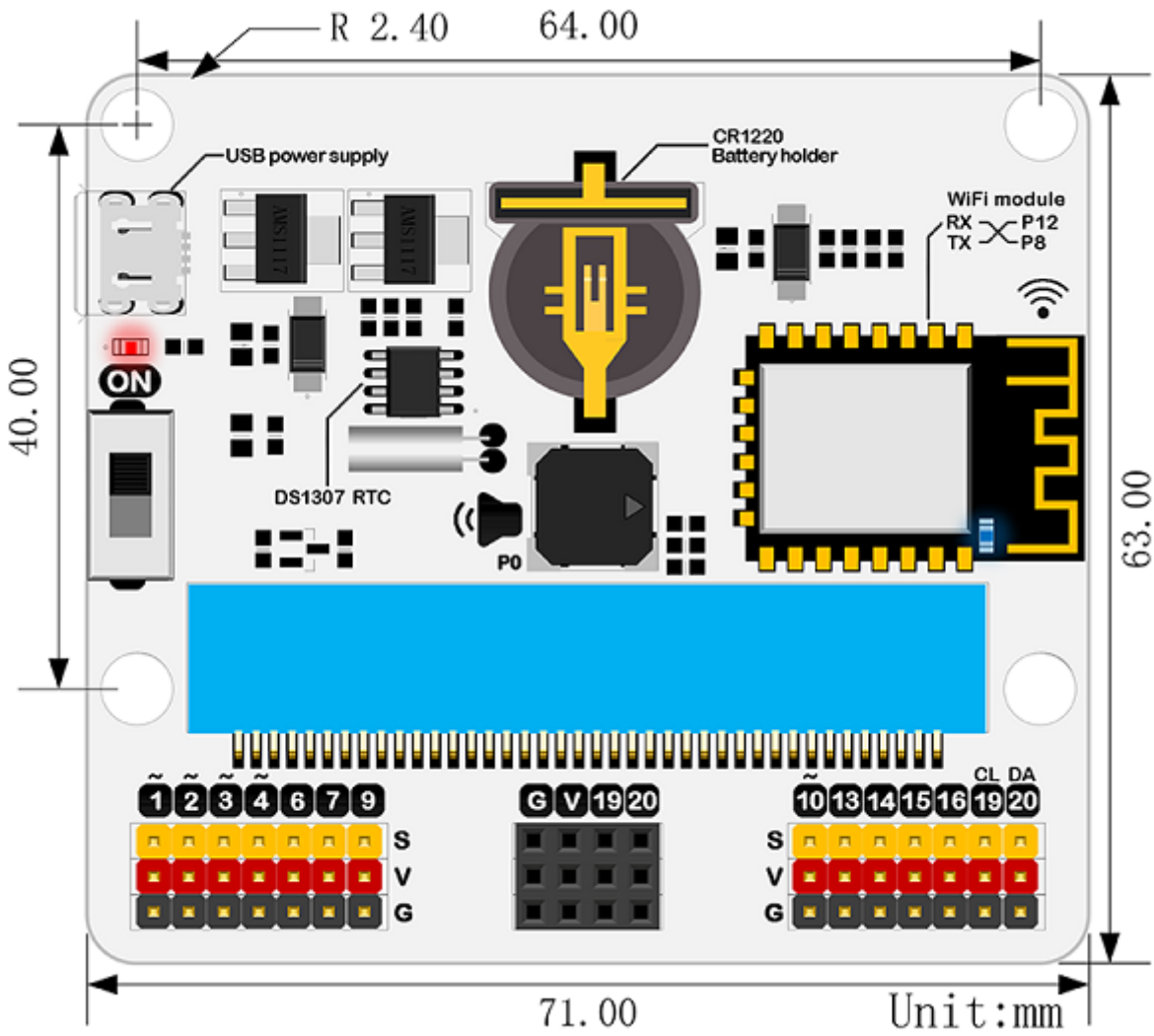
- Intergrated ESP12F WiFi、RTC and passive buzzer module
- Extended most IO ports by GVS
- Silk-screen indication onboard main components
- Separately lead IIC interface, directly plug in OLED, BME280 and IIC,etc
- Intergrated buzzer and earphone jack
- Adaptive for LEGO (4 standard spacing Lego fixed holes)

## 2.2. Hardware apperance & Parameters

### Size & Installation

- Product size：71mm x 63mm x 23mm
- PCB thickness：1.5mm

- Hole diameter： 2.4mm

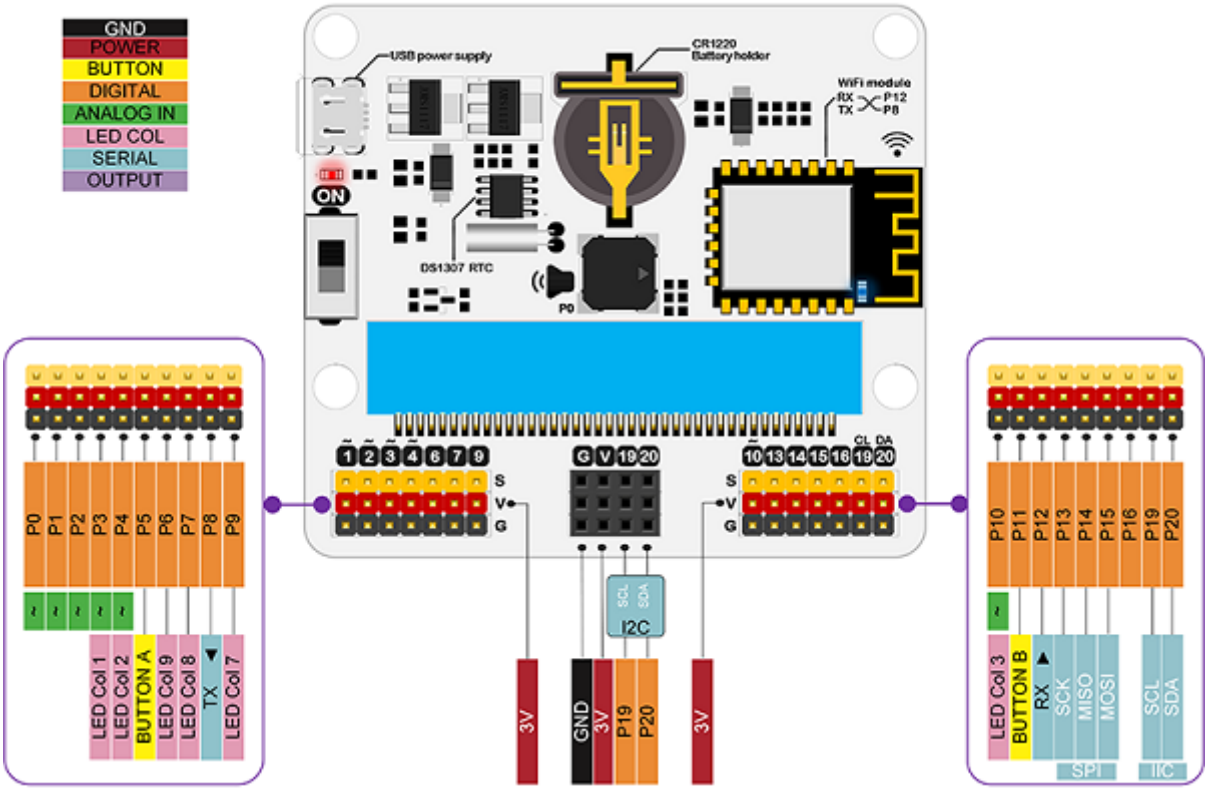



## Parameters

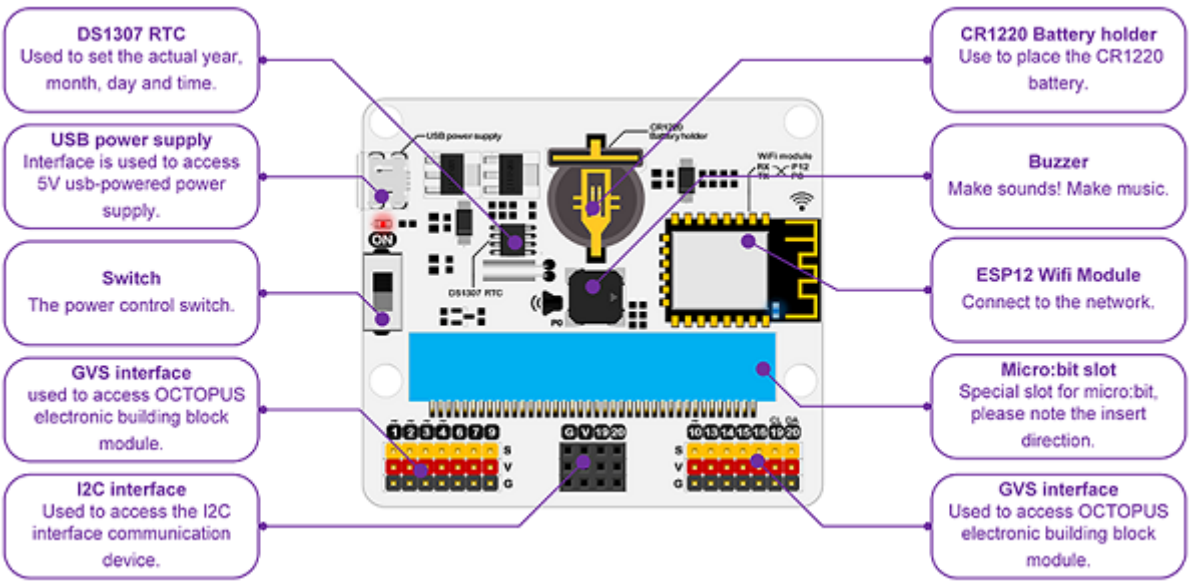
| Items | Parameters | Remarks |
|---|---|---|
| DC supply | USB-5V | - |
| Maximum power current | 800mA | - |
| Operation temperature | -25~80°C | - |
| WIFI module | ESP8266 | ESP12F |
| Buzzer | Passive buzzer | - |
| RTC timing | DS1307 RTC | - |
| RTC timing battery | CR1220 button cell | Equipped by yourself |
| Lead out | Not all | - |
| port lead | Serial port can map the IO port | Coding |
| I2C port lead | 19、20 pin | Only for I2C pin |

| Items | Parameters | Remarks |
|---|---|---|
| SPI port lead | 14、15 pin | for common IO |
| Size | 71.00mm X 63.00mm | Without packing |
| Net weight | 30.00g | Without packing |

## Pin interface drawing




## Introduction of main module

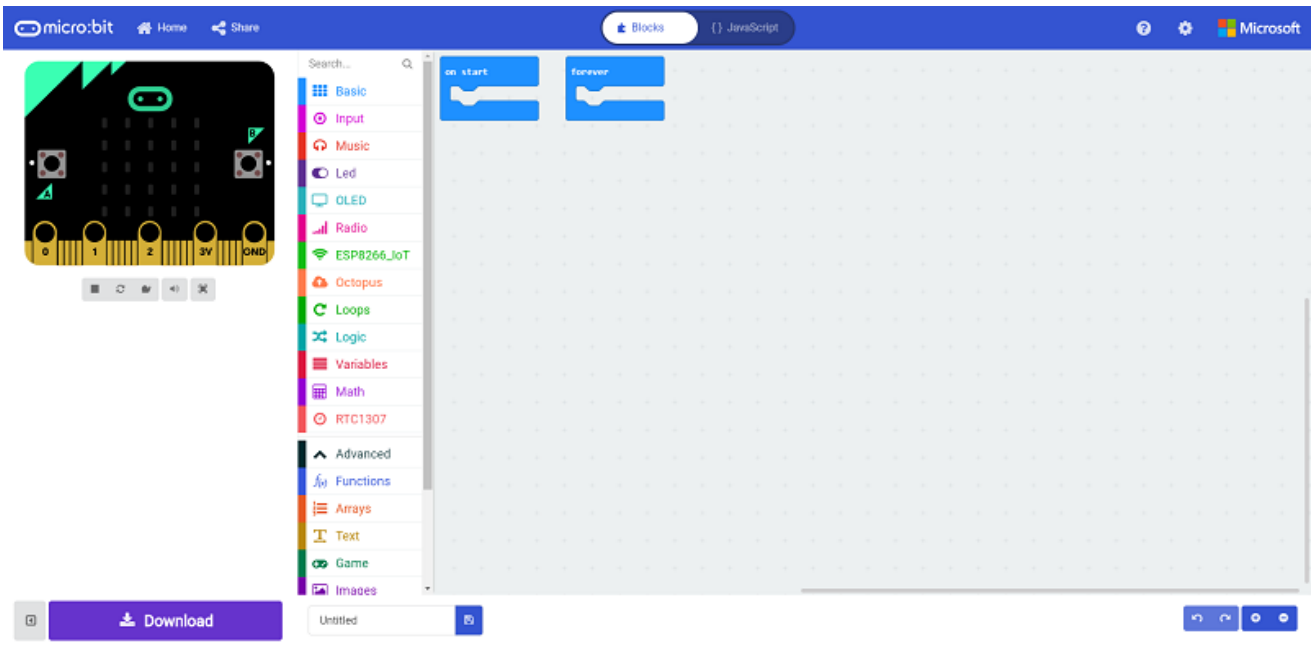


## 2.3. Software support

- Coding： Makecode/Micropython/JavaScript/

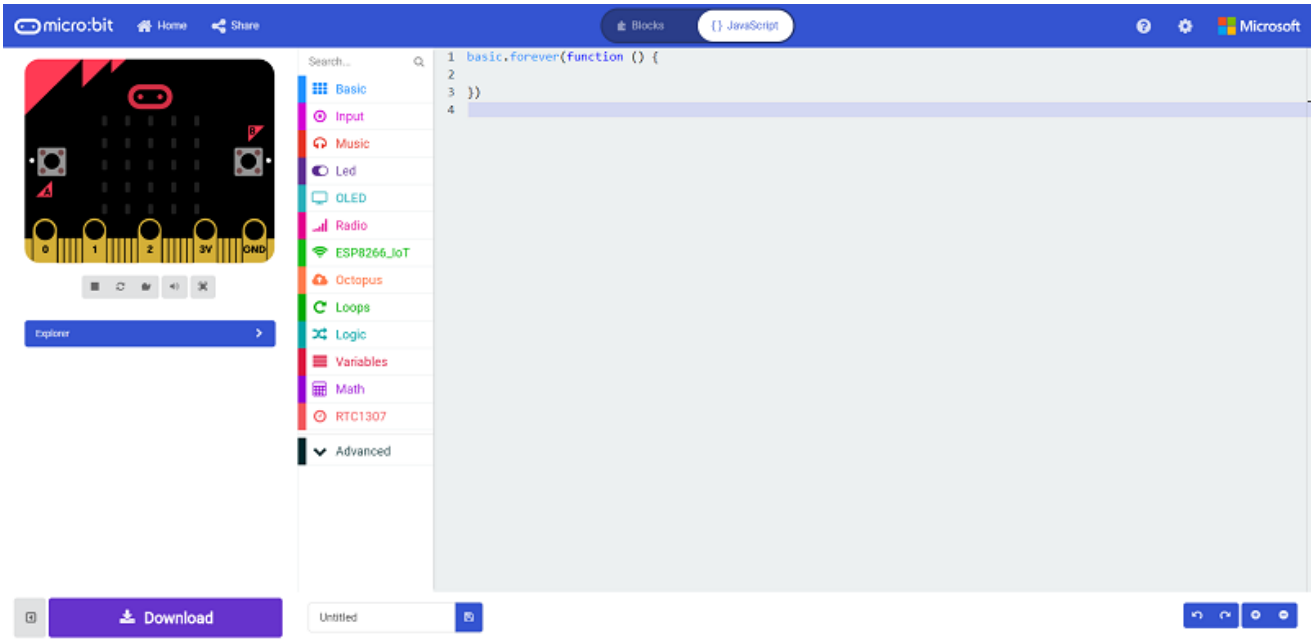
### Makecode block

- Coding developed by Microsoft and mainly publiced by micro:bit offical.
- https://makecode.microbit.org



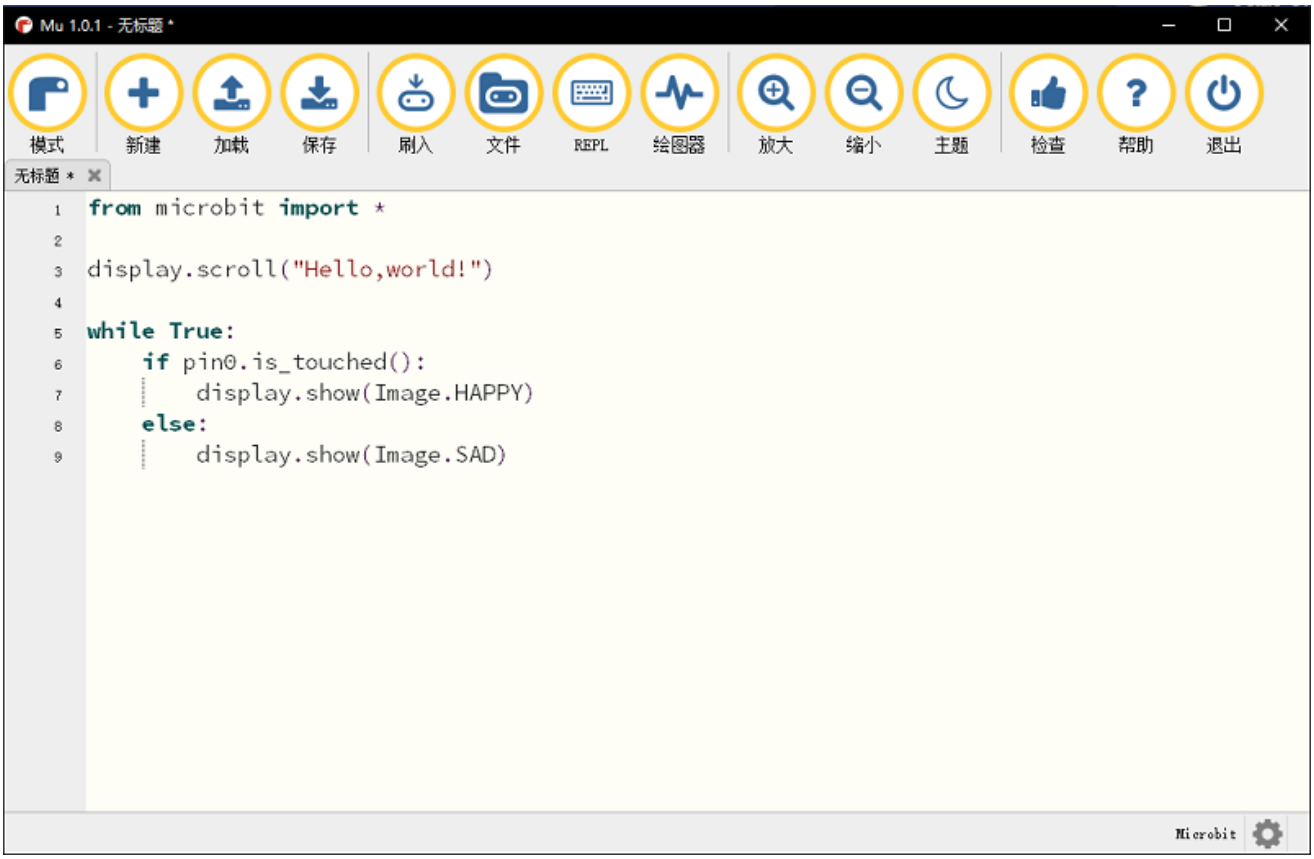

### JavaScript

- Click on `JavaScript` of the `makecode` for `JavaScript` coding.
- https://makecode.microbit.org




## MicroPython

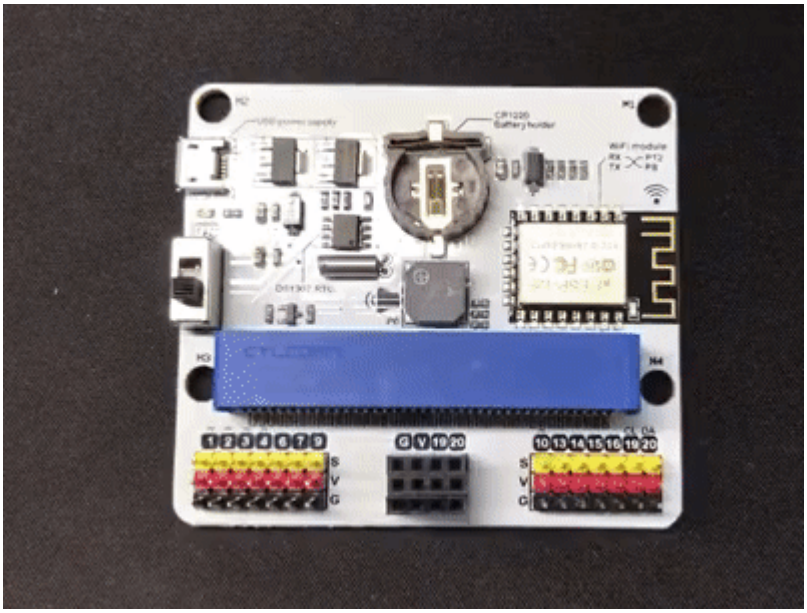
- Using `MU` for advanced coding method `MicroPython`
- https://codewith.mu/



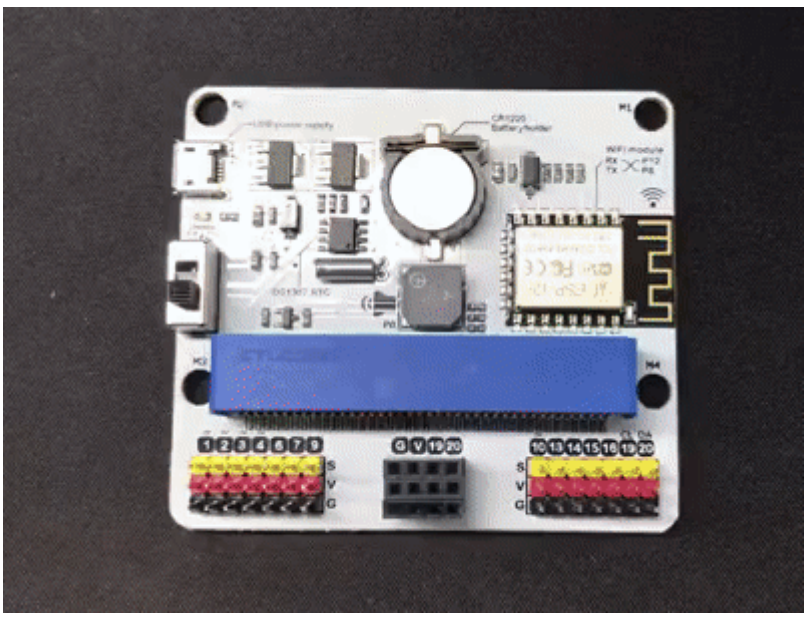

# 2.4. Easy start

## Hardware connection

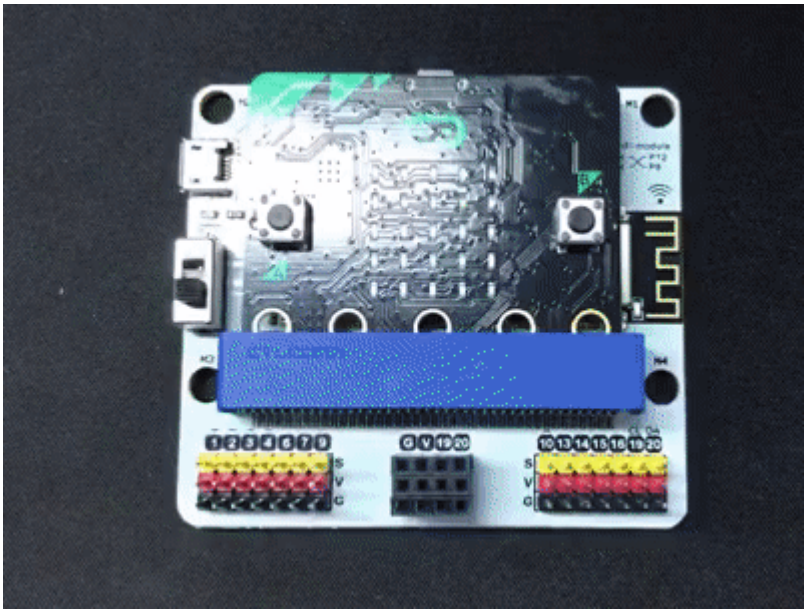
- Fisrt, install CR1220 button cell for power supply to the RTC timing.



- Plug the micro:bit to the Iot:bit.



- Using single USB for power supply to expansion board and switch on.



## Coding

---

### Coding Platform

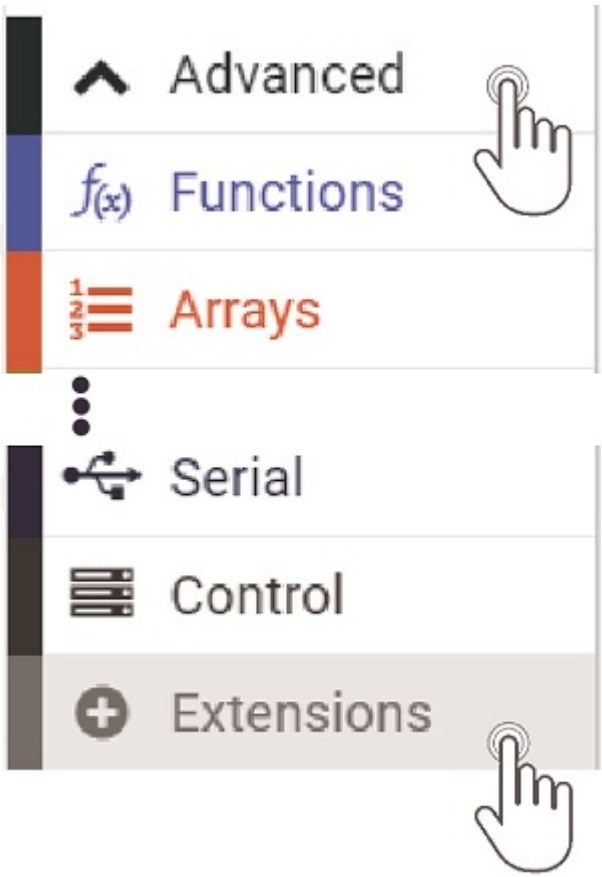
---

https://makecode.microbit.org

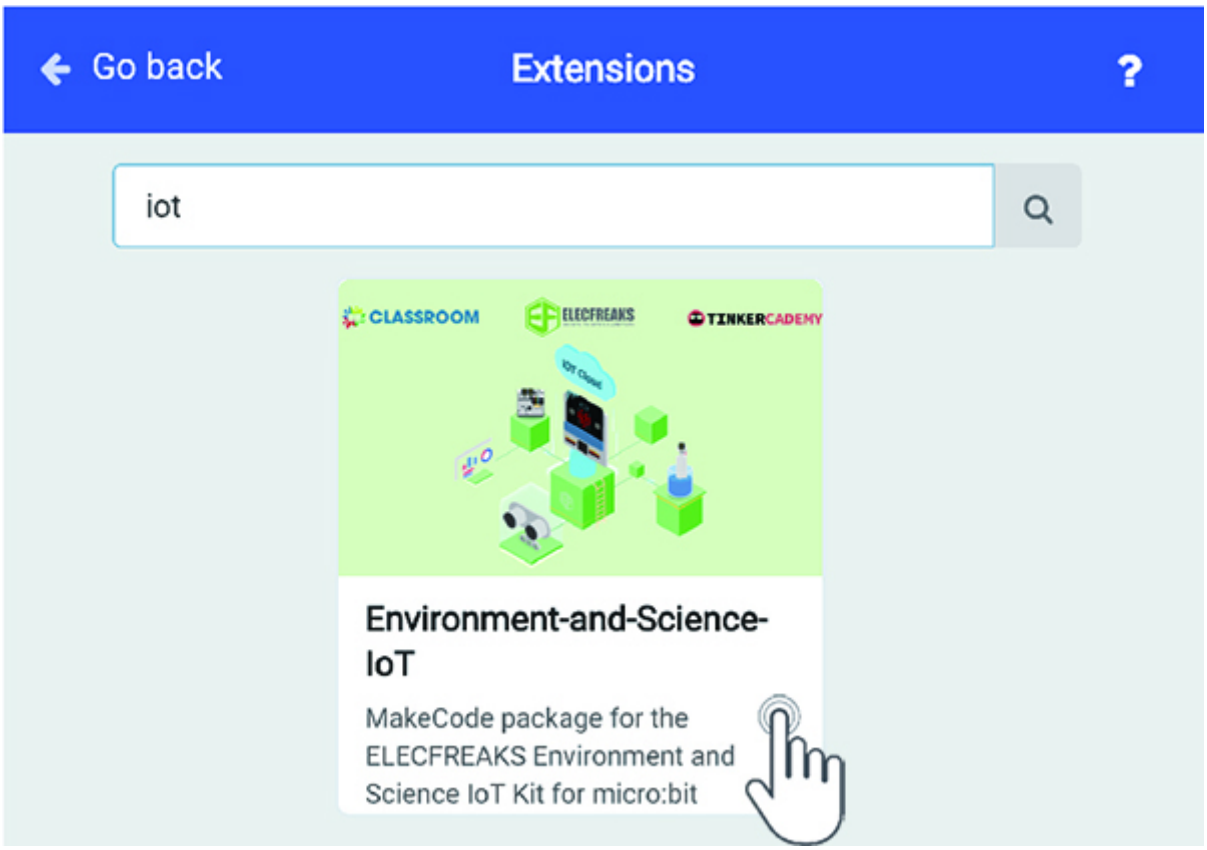
### Add codebase

---

- Click on “Advanced” in the MakeCode Drawer to see more code sections.



- We need to add an extension for coding to the IOT. Click on the “Extension” at bottom of coding drawer, then Search for “IOT” and click on the IOT package to add it to your project. (As below picture)

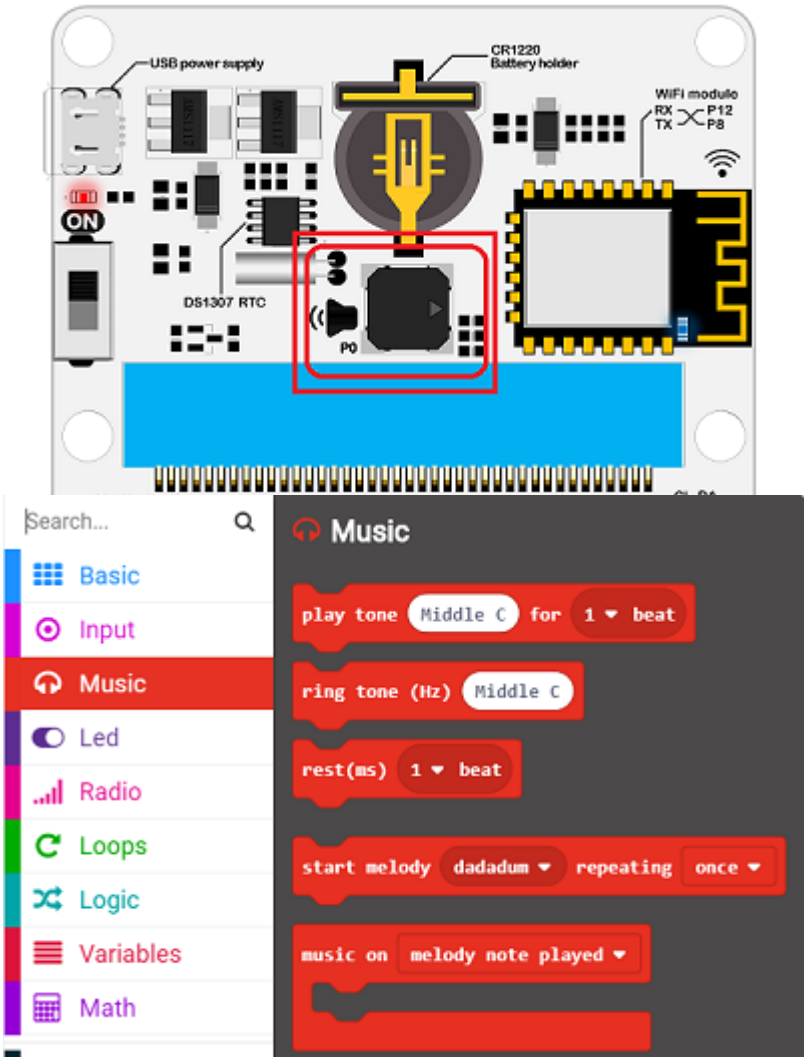



Note： If you get a warning telling you some packages will be removed because of incompatibility issues, either follow the prompts or create a new project in the Project file menu.

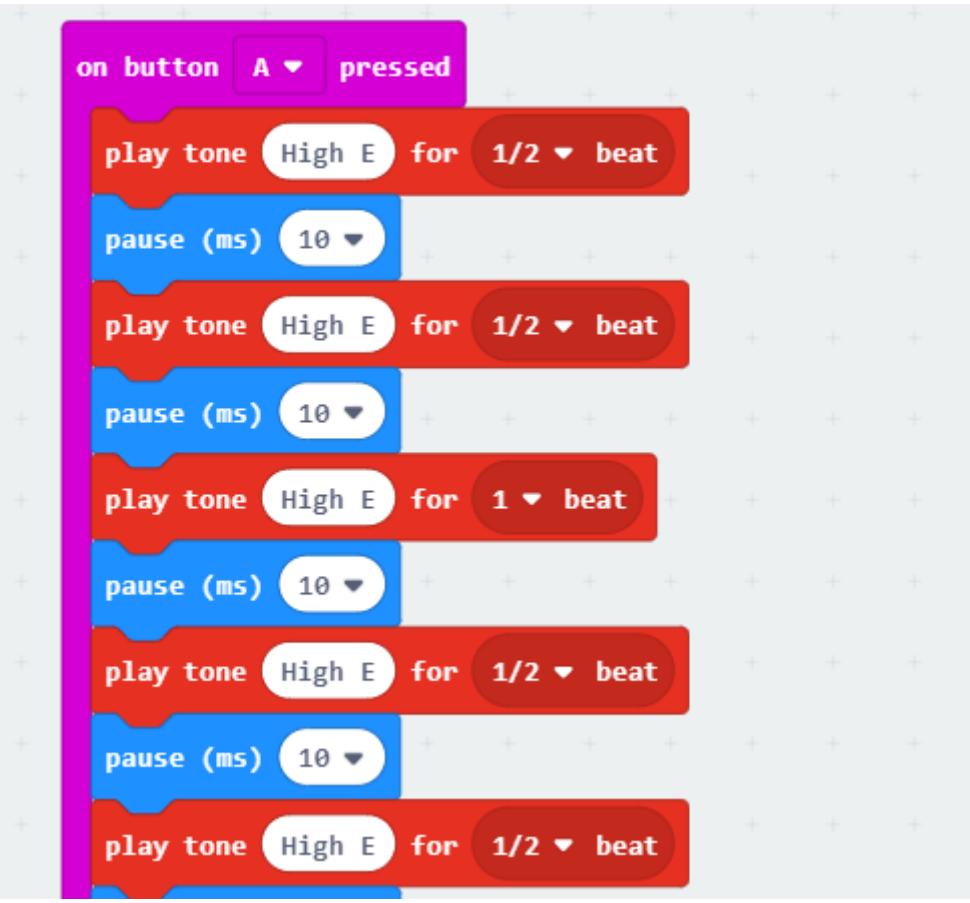
## How to drive the buzzer

---

- There is on-board buzzer on the IOT:bit as below picture. Let the buzzer be connected to the PO of the micro:bit to play music by the `music` of the makecode.



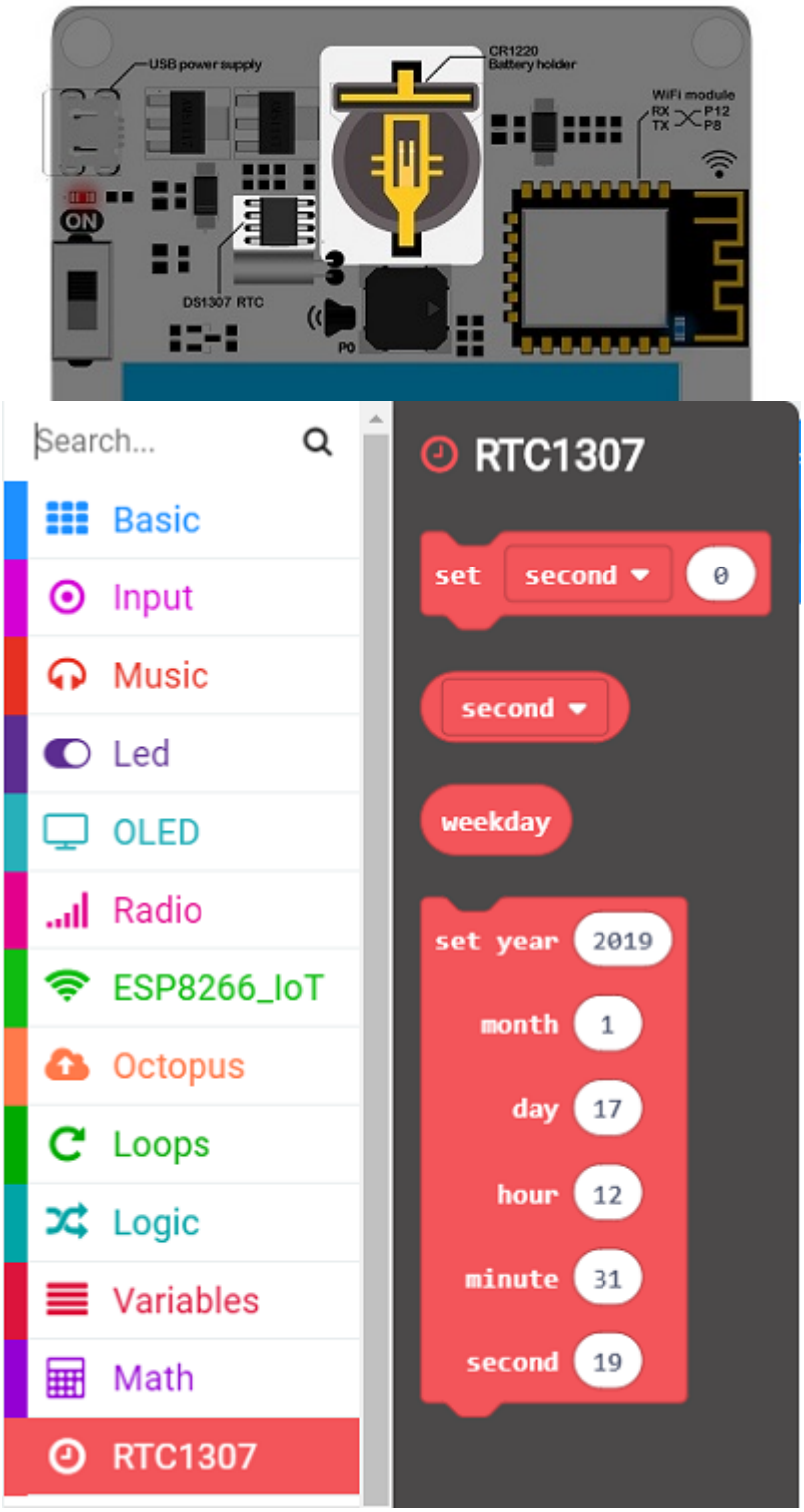

- When button A be pressed in the “input”, play a bit of music.



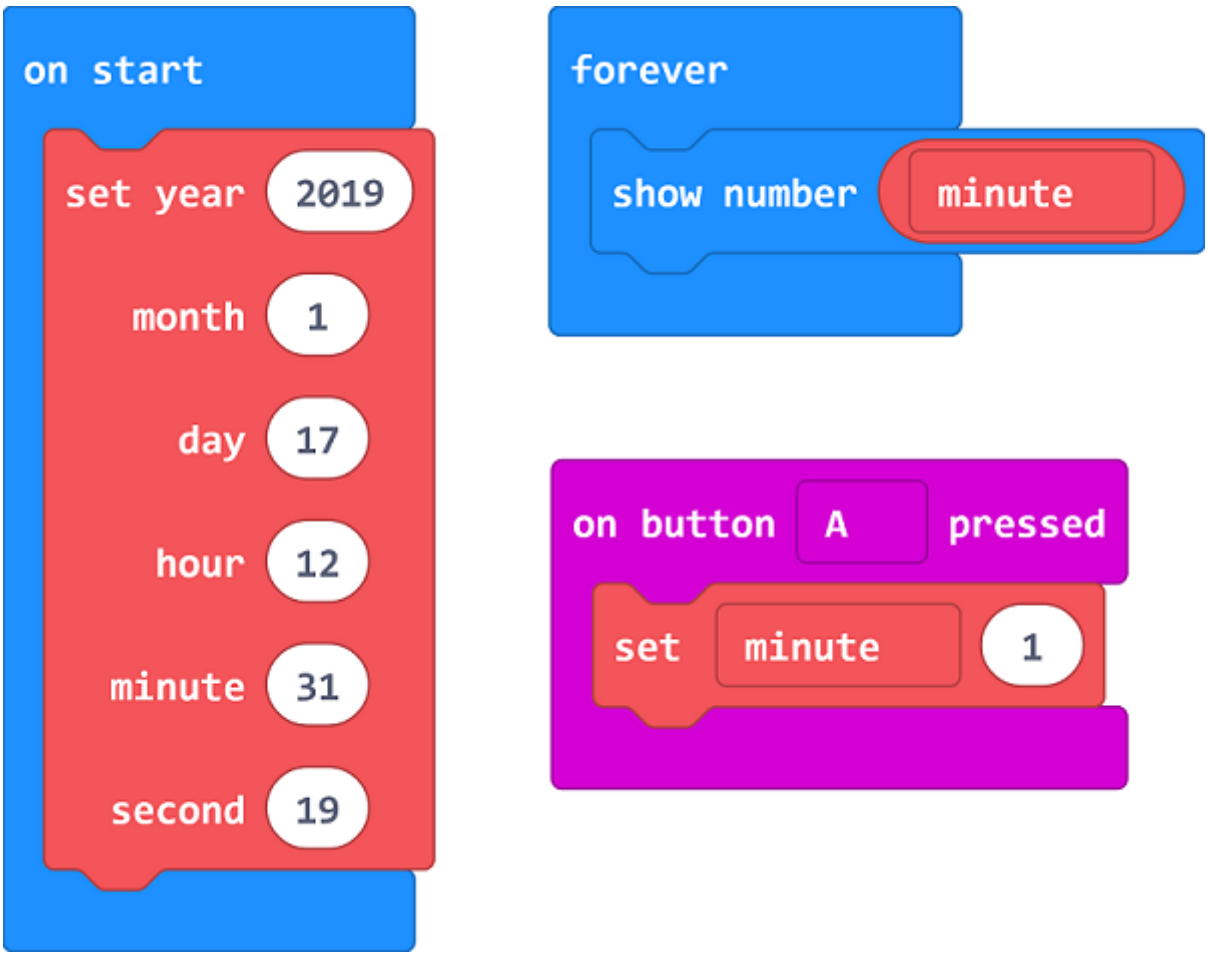
- Program link：https://makecode.microbit.org/_4j6PCeV087AW

## How to use RTC

- There is on-board DS1307RTC timing on the IOT:bit as below picture. RTC timing need a CR1220 button cell for lasting power supply. We need the RTC timing keep accuracy in the event of a power outage.



- Press button A to set the time to the set time. On start with RTC function and the minutes be displayed on the 5*5 allay.
- Turn off the power after turning off the power for one minute, and the dot matrix display will show the number of minutes after another 1 minute.

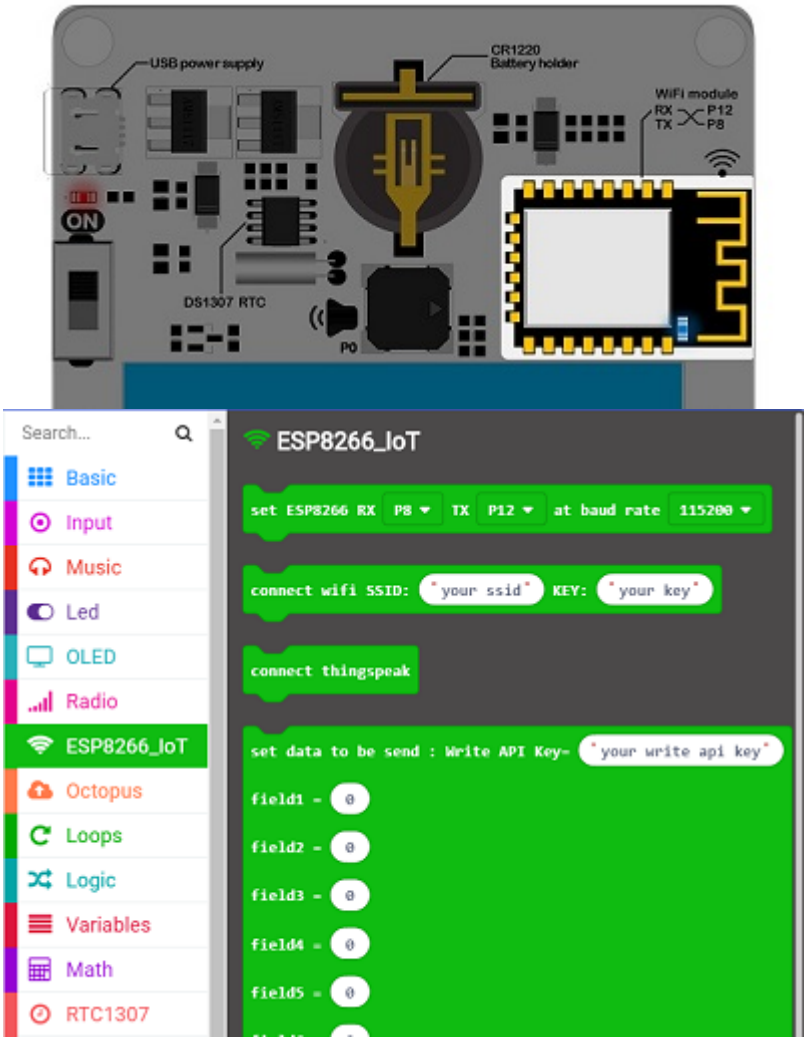



- Program link：https://makecode.microbit.org/_e9d3vW96bPe2

## How to use the internet function

---

- The most important function of IOT:bit is WIKI. The onboard `ESP-12F` WIFI module can connect WIFI and send information. To use port for communication with the micro:bit and the pin `RX-P8`, `TX-P12` is special for IoT.
- Using thingspeak as cloud to coding and data uploding.thingspeak guidebook




- On start, initialize ESP8266 to default connection with P8 & P12.
- Connect your own WiFi, input the key and the password.
- Under foever loop, connect the thinkspeak and set date to send, then pause.

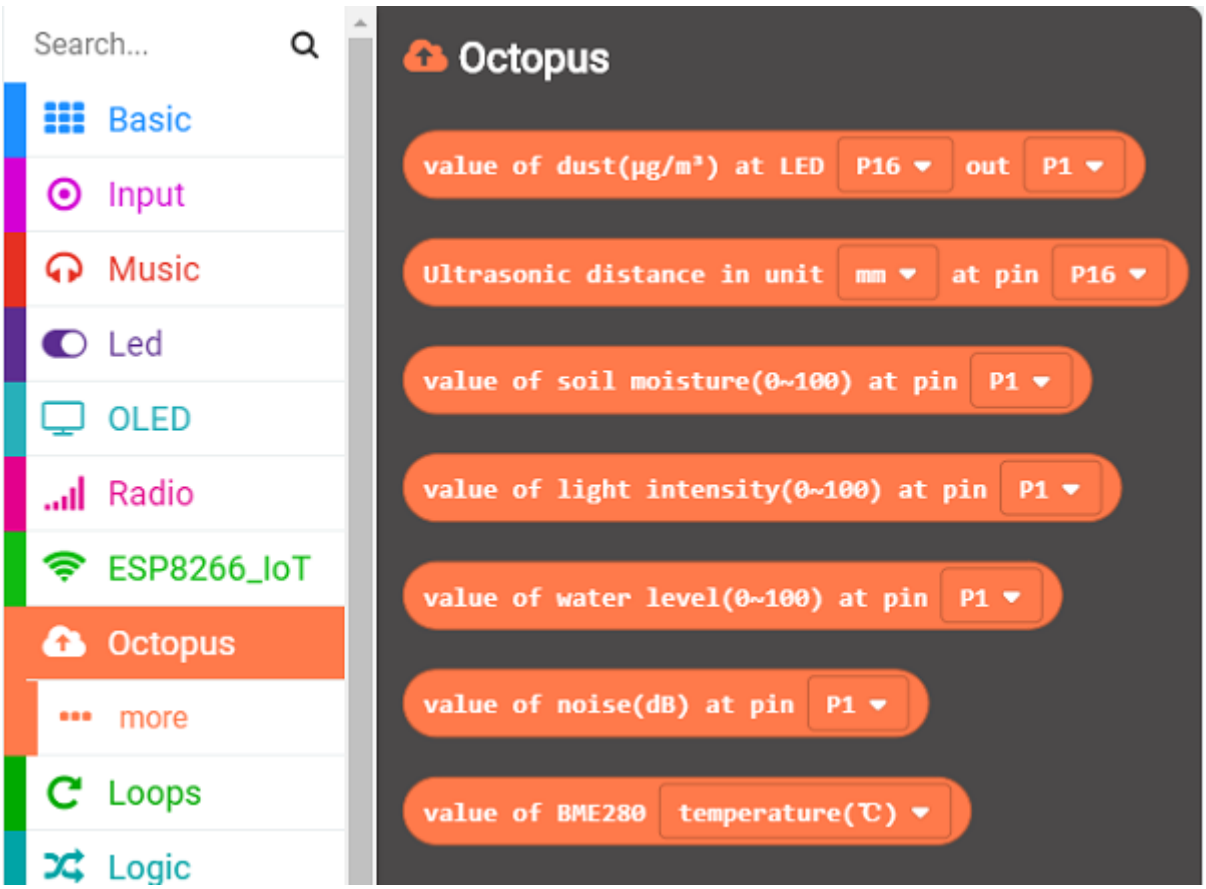
```
on start
  set ESP8266 RX P8 TX P12 at baud rate 115200
  connect wifi SSID: your ssid KEY: your key
```

```
forever
  connect thingspeak
  set data to be send : Write API Key= your write api key
  field1 = 0
  field2 = 0
  field3 = 0
  field4 = 0
  field5 = 0
  field6 = 0
  field7 = 0
  field8 = 0
  send data to thingspeak
  pause (ms) 1000000
```

- Program link：https://makecode.microbit.org/_JAXAmmHq4FhW

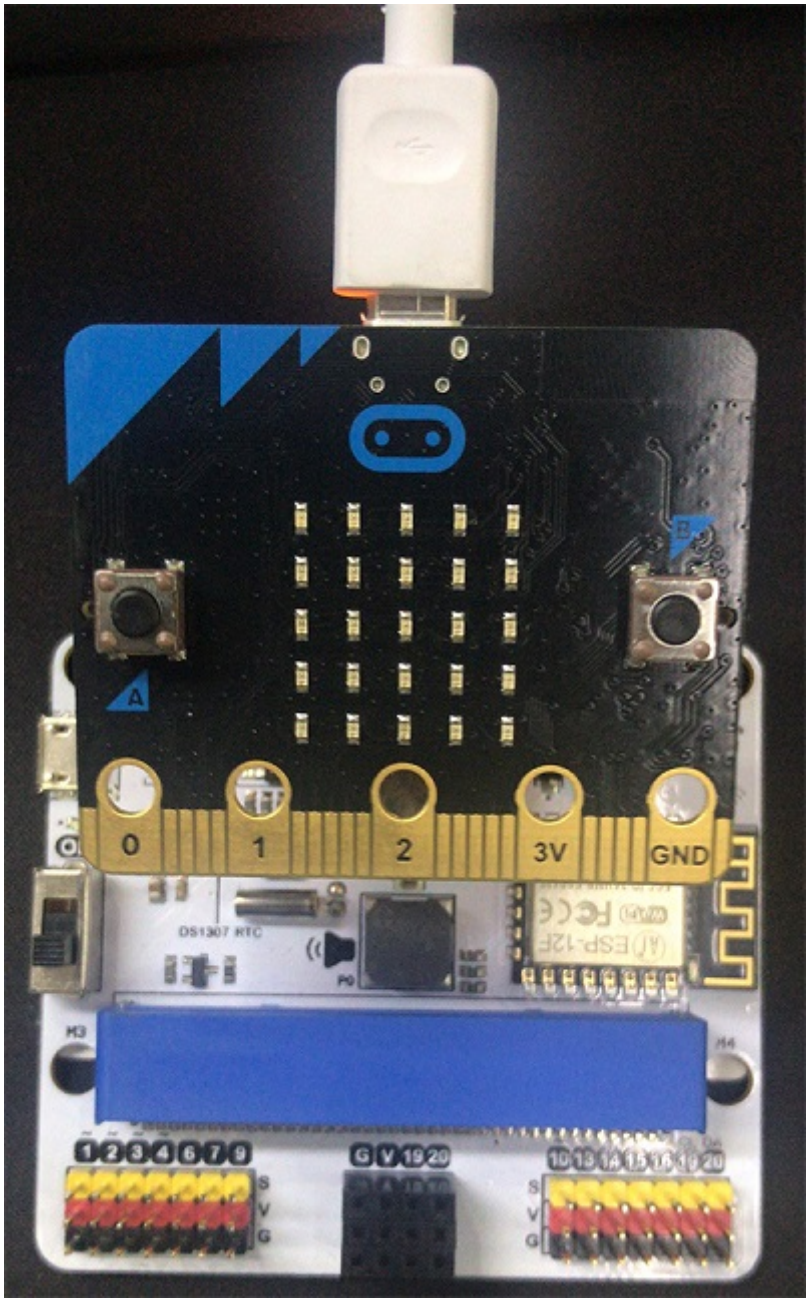
####Codebase for other sensors

-IoT:bit also supports the other ELECFREAKS sensors can the new Octopus blocks are added in the package.



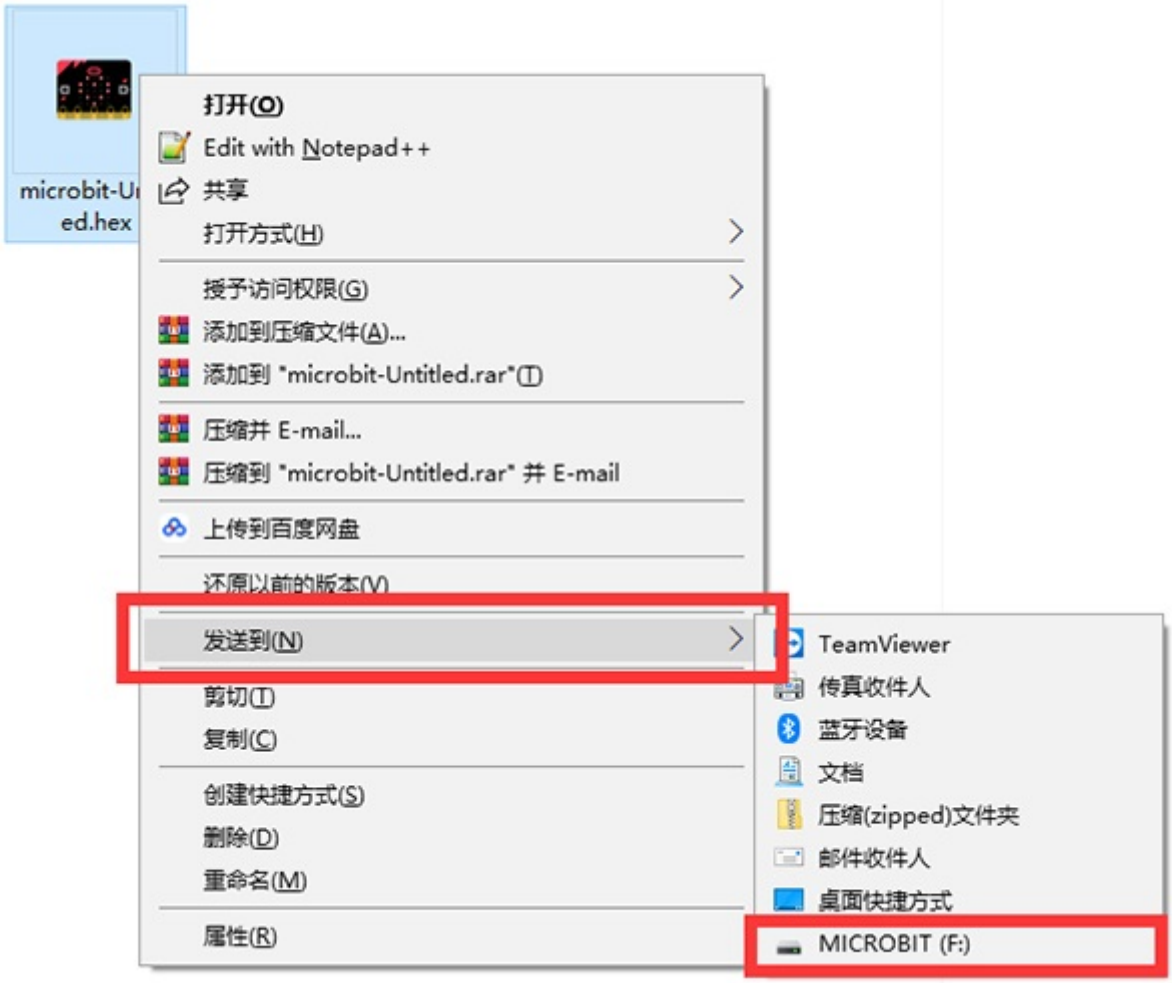

## Down load code

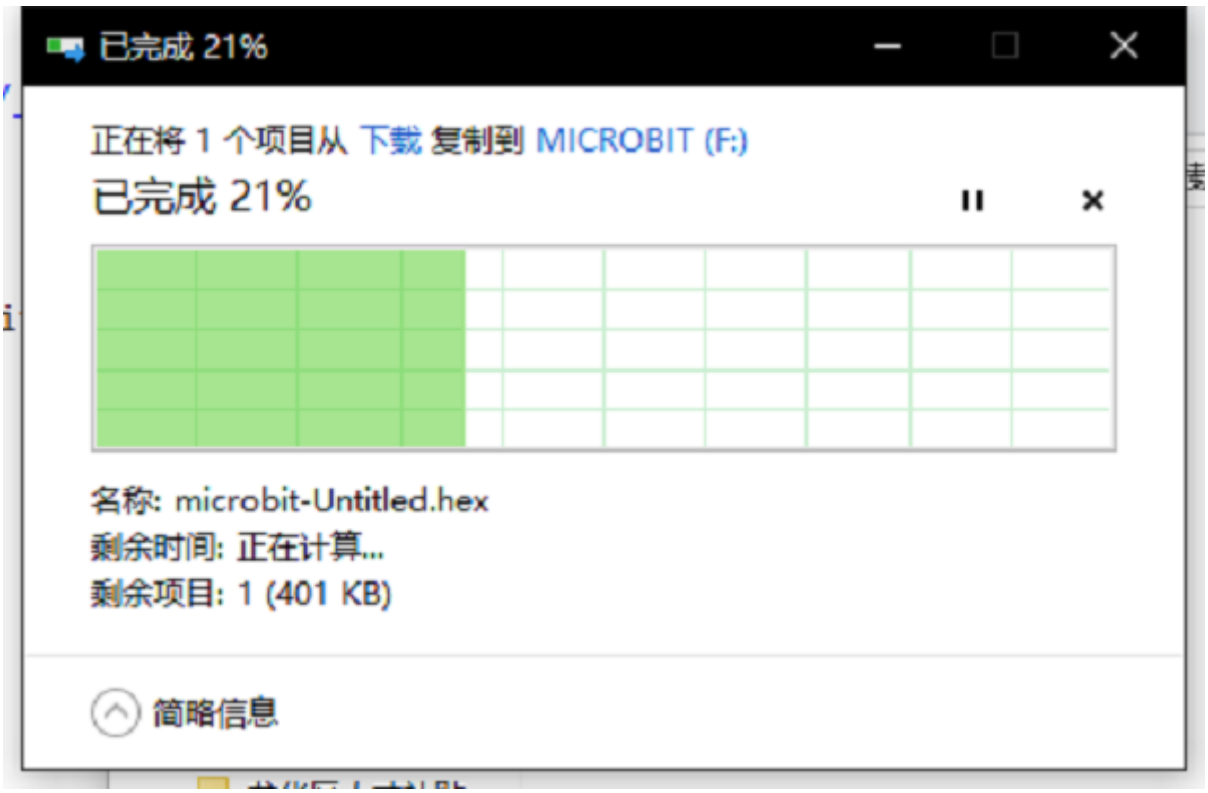
- Let your micro:bit be connected to one side of the USB, the other side to your computer. （Inserting on the expansion board may cause the micro:bit connection to be abnormal or damaged.）




- Then copy your documents to the micro:bit.

- Now, here is your observing time!



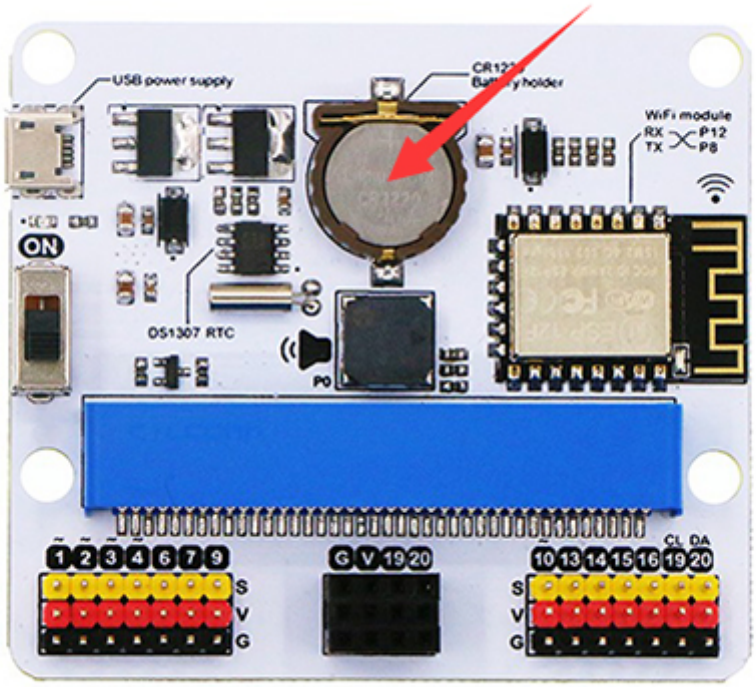

## Documents

---

## FAQ

####Question: After installing the button cells, why can't I get the correct data for RTC ?

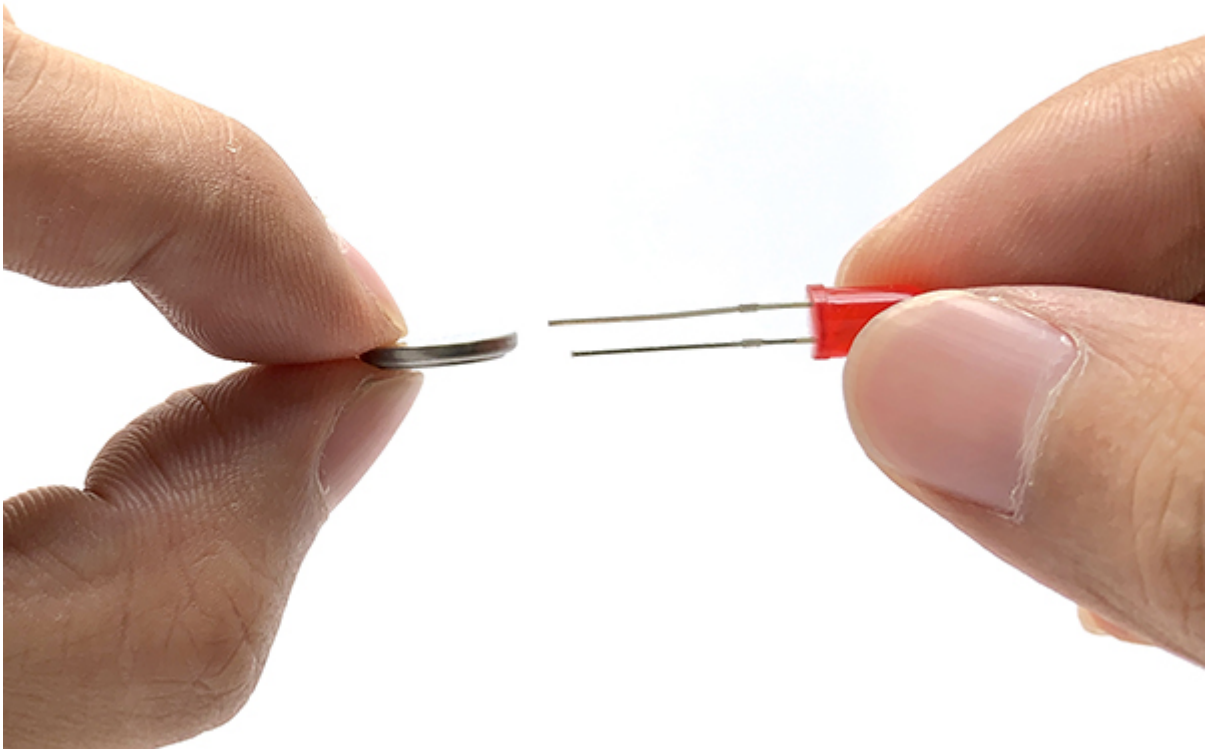
Solution: The nominal voltage for CR1220 button cells is 3V, while some of the cells in the market may have a higher voltage than 3V. The high voltage will make abnormal operation of the RTC happen, to solve this, you can use the used cells or discharge the brand new cells by yourself.



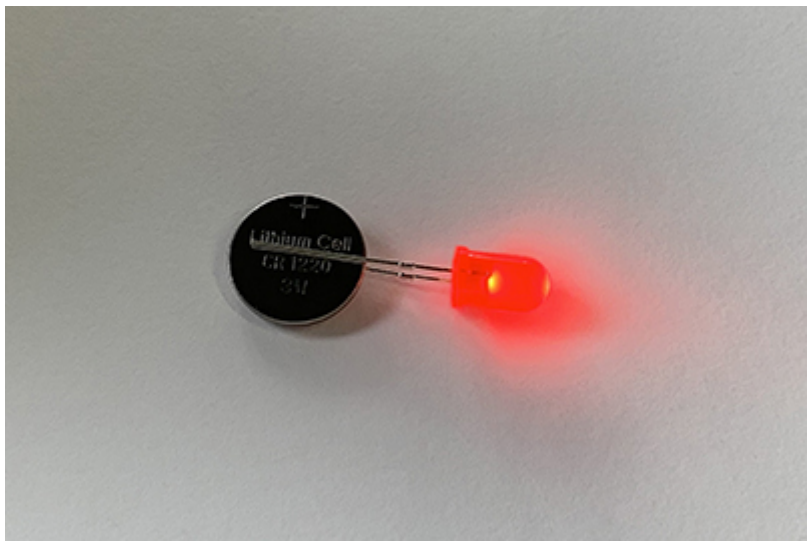


Cells Discharge Steps

1. Connect the pins of the LED as the picture shows(Make sure the positive and negative anodes connect in the right way)



1. The LED lights up after connection, the button cells in 3.4V would come down to 3V after around 10 seconds.




2. Detect the cells voltage with a multimeter and it works well after deducing the voltage to about 3V.


UNI-T
UT51 MULTIMETER
200
2k
20k
200k
2M
20M
200M
200m

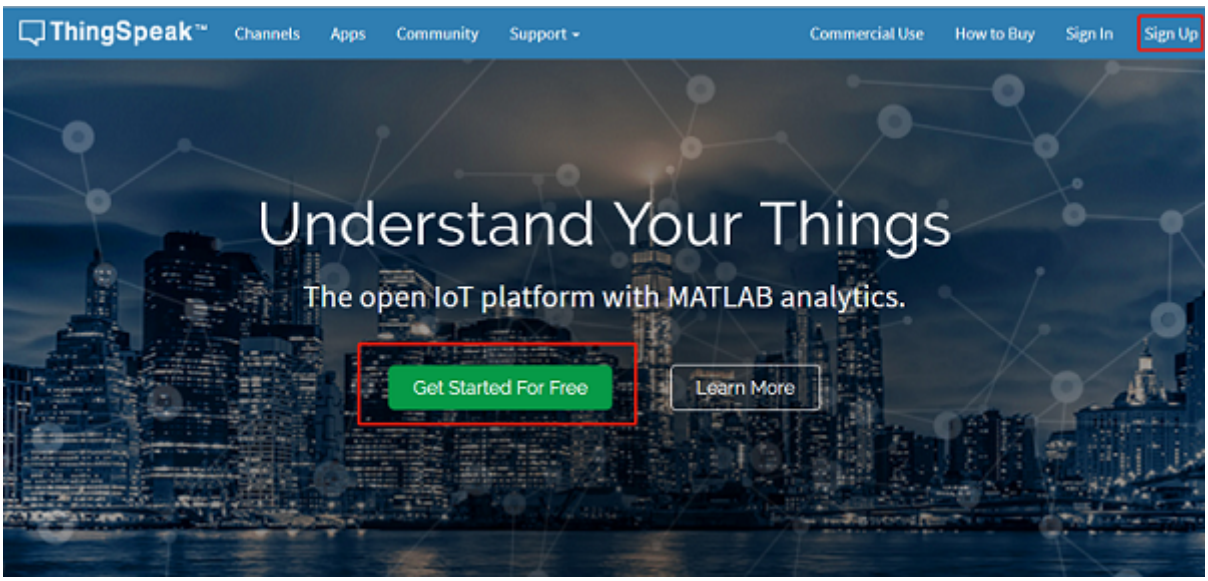
# 3. The first part: Upload the data collected by IoT kit to the Thingspeak IoT platform.

ThingSpeak can deal with HTTP request, save and process data. This platform has function as open application, real-time data collection, location data, data processing, visualization, device status information and plug-in. It can integrate many hardwares and software platforms like Arduino, Raspberry Pi, ioBridge/RealTime.io、Electic Imp, mobile & internet application, social network and MATLAB data processing. It also porvide hosted service apart from open source edition.

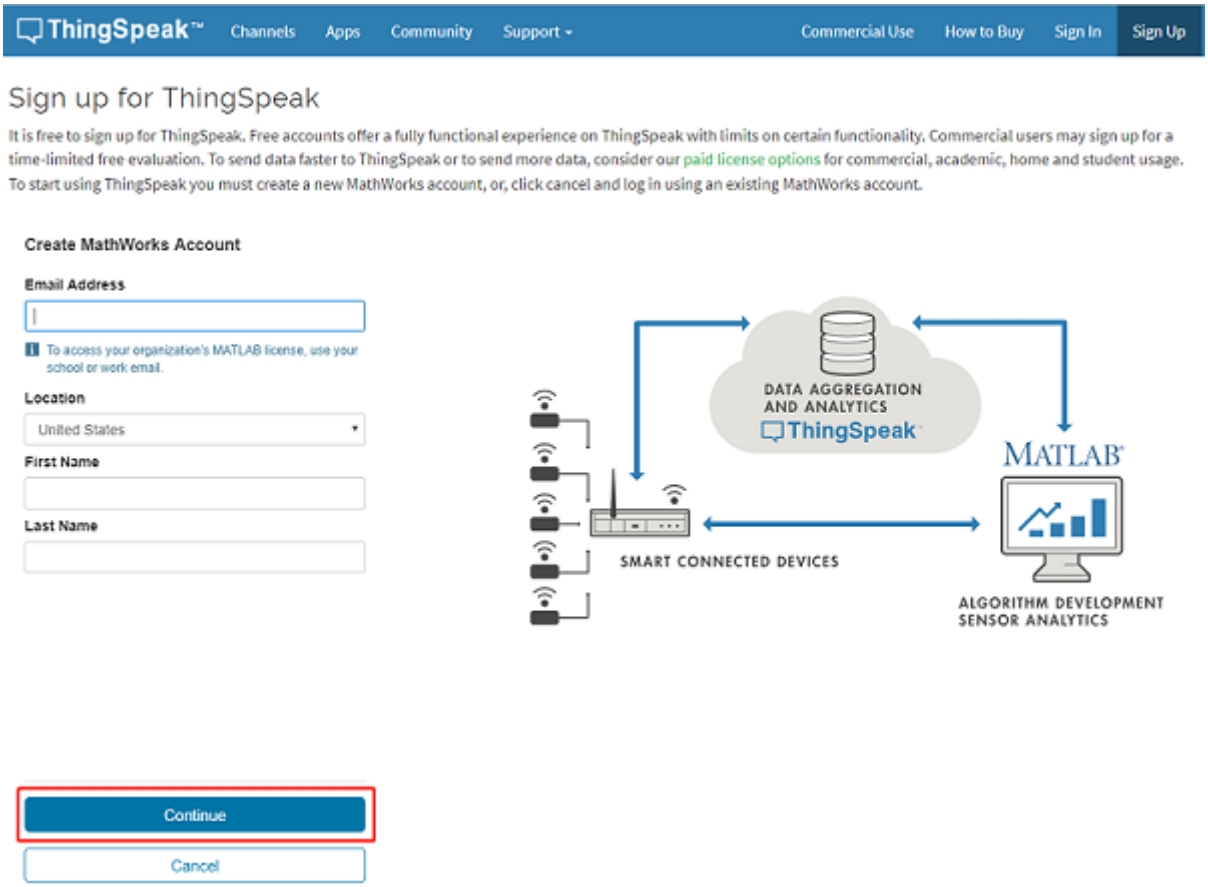
Thingspeak link: thingspeak

## 3.1. Thingspeak Registration

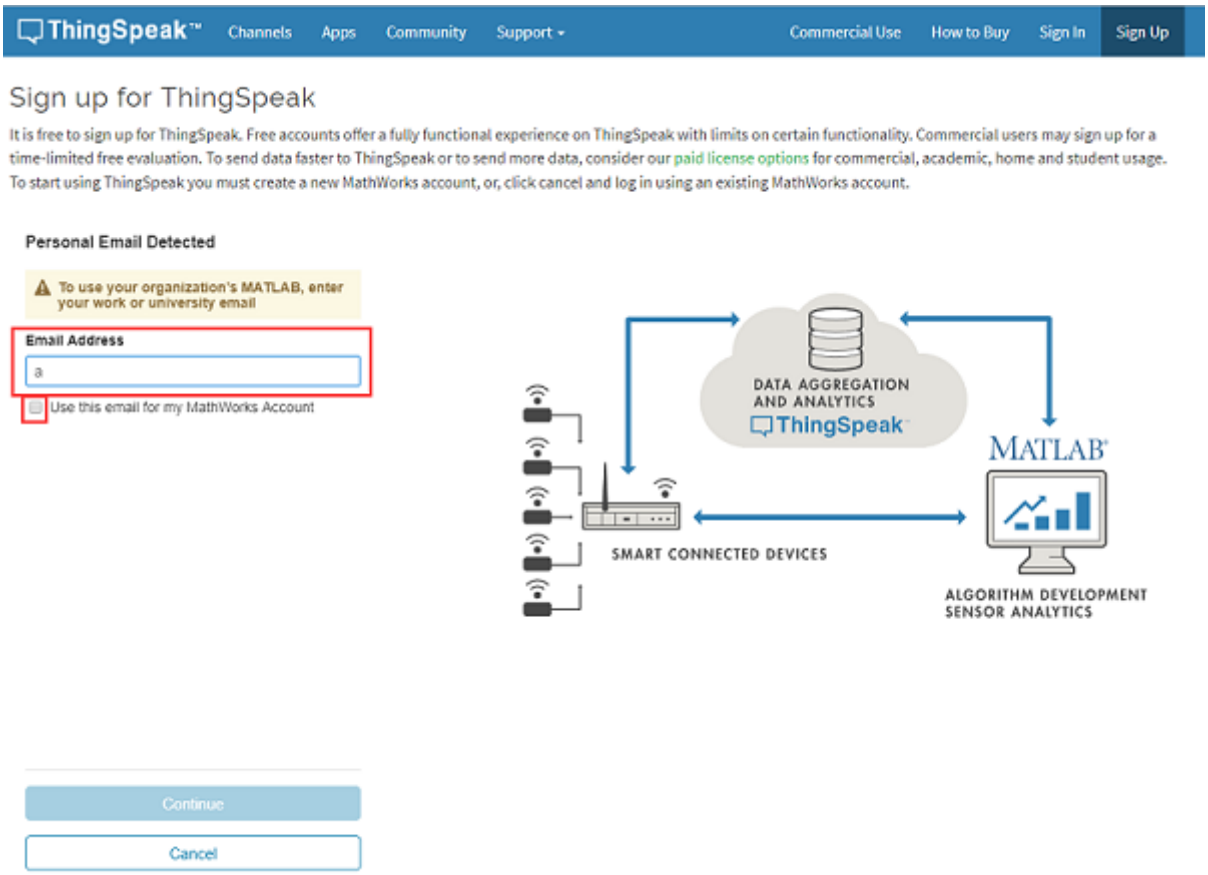
- Head to thingspeak website, click on Get Started For Free and in create MathWorks Account package.



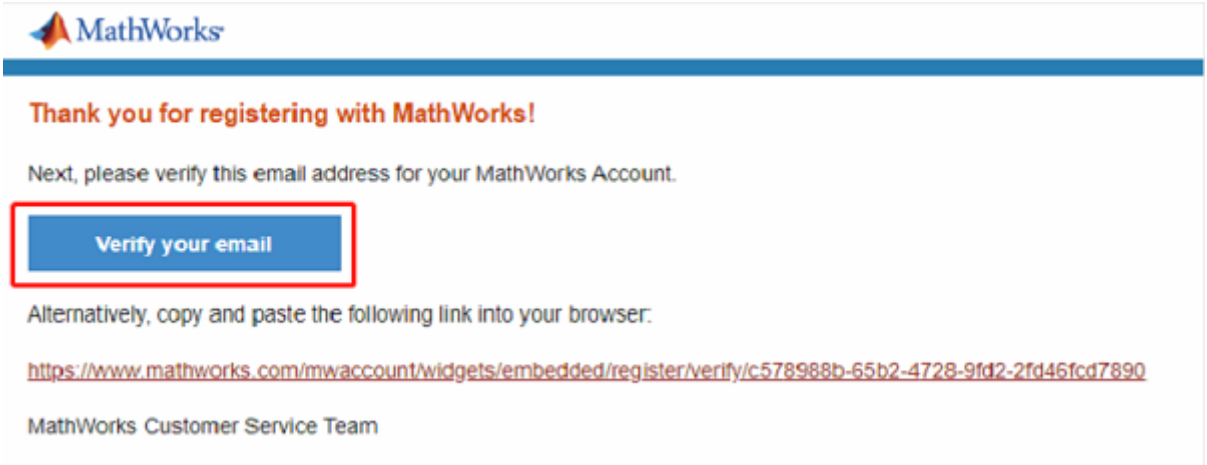

- Fill in the registration information and click continue to go on.（As below）




-Confirm the email address, click it as the sign in account to continue.

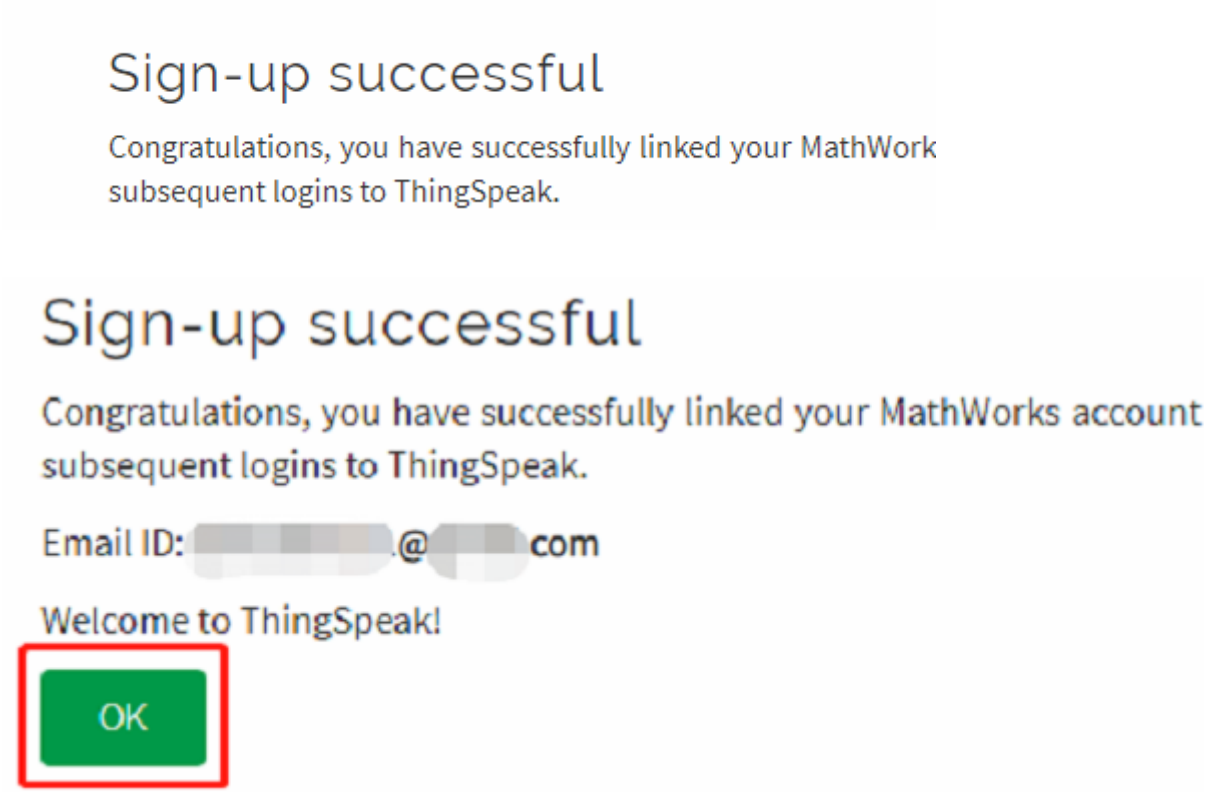



- Verify your MathWorks account and the thingspeak will send an email to your email box. You need to click the lin of the email to verify and continue.




- Then set your own user ID and password(Note: both Captital and lower case words are required)，then click on continue to go on.

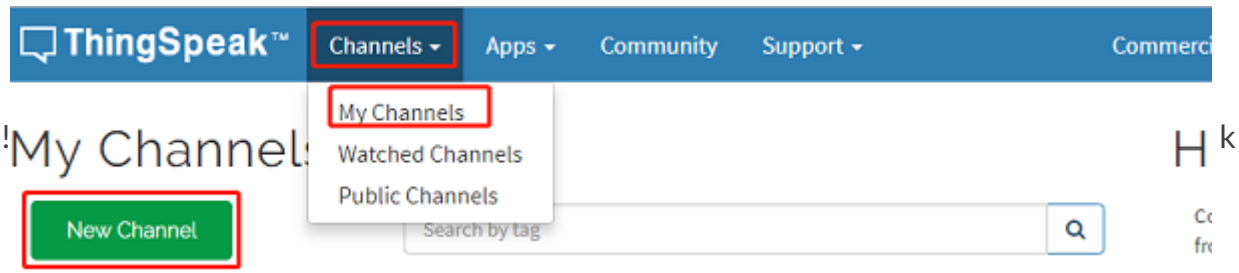

- Then, Sign-up successful!



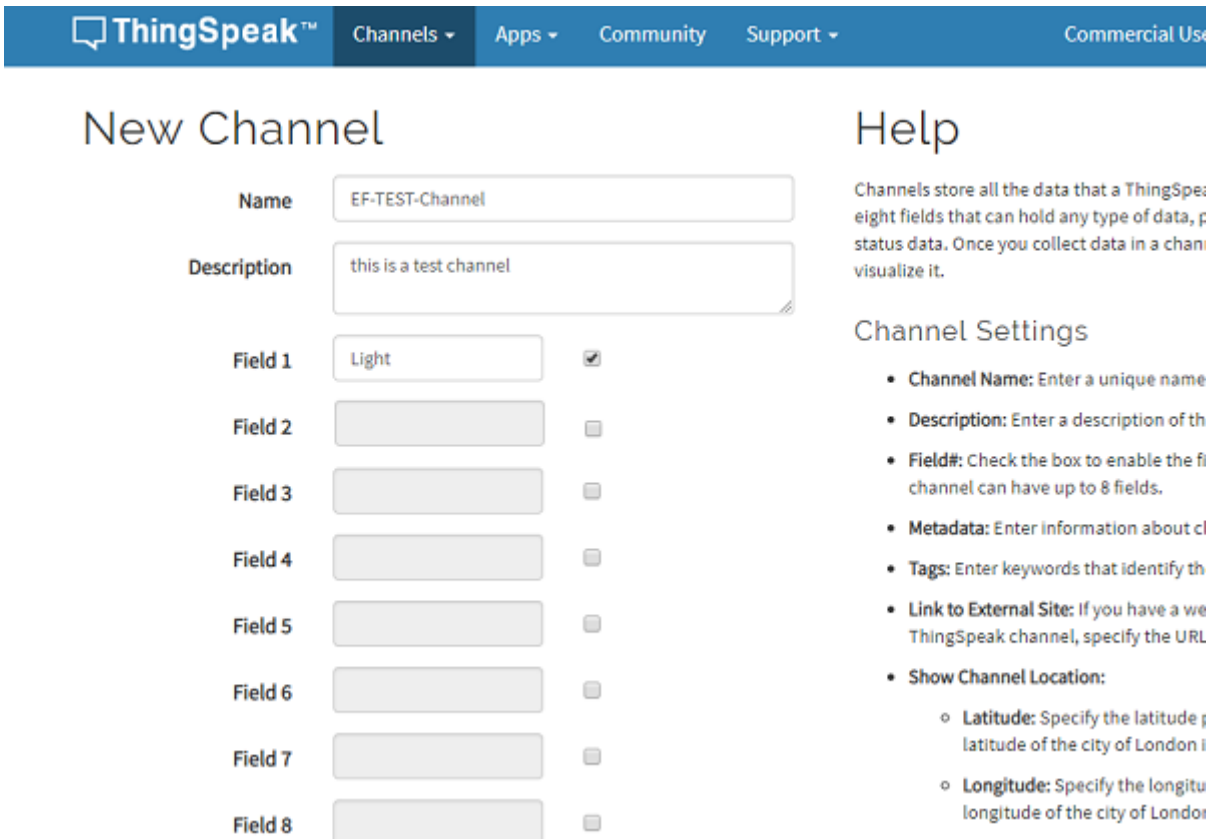

---

##Thingspeak guide:

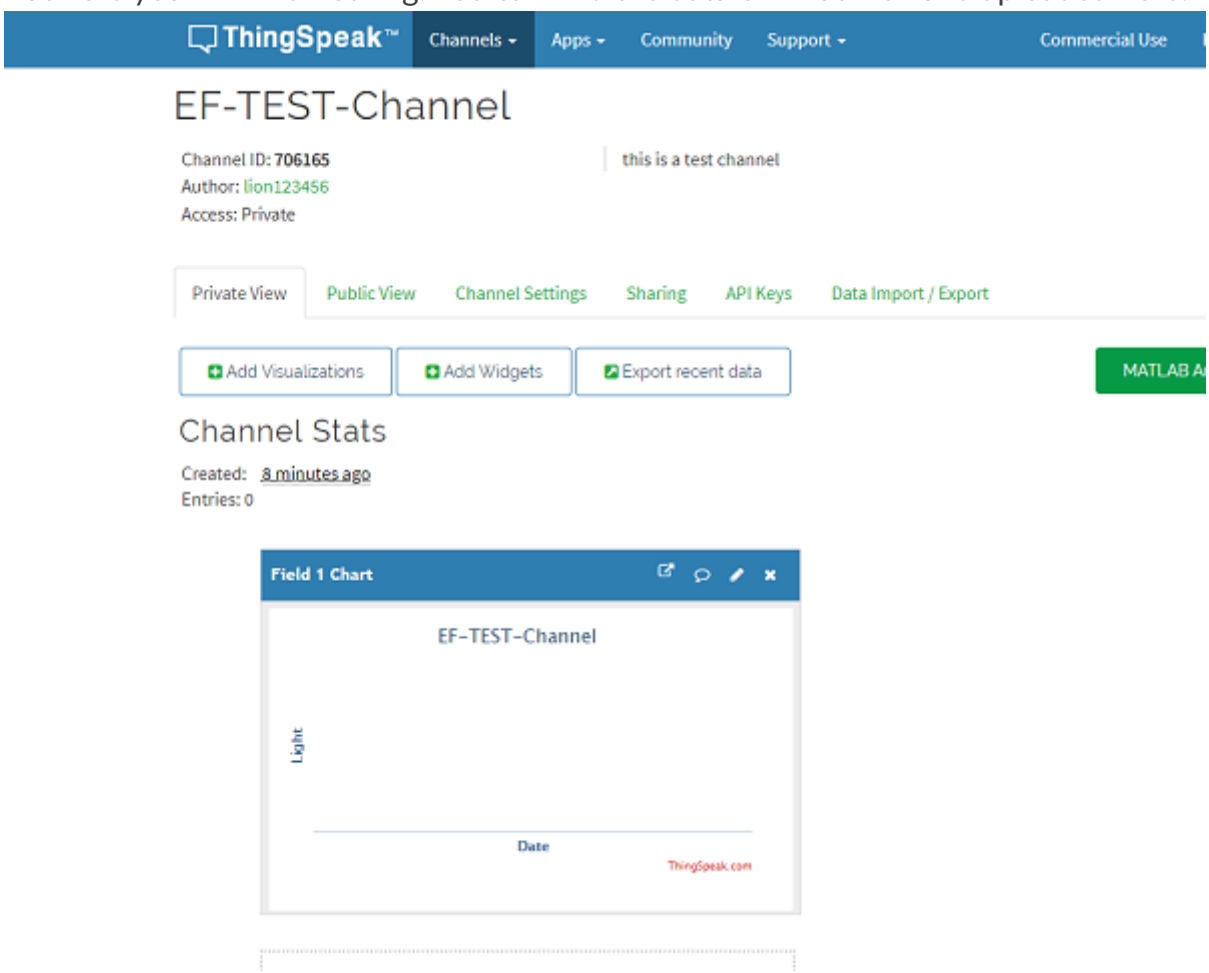
- Here we will upload luminous intensity for testing.
- Click on “MY Channels” and click on **New Channel** for your new project.



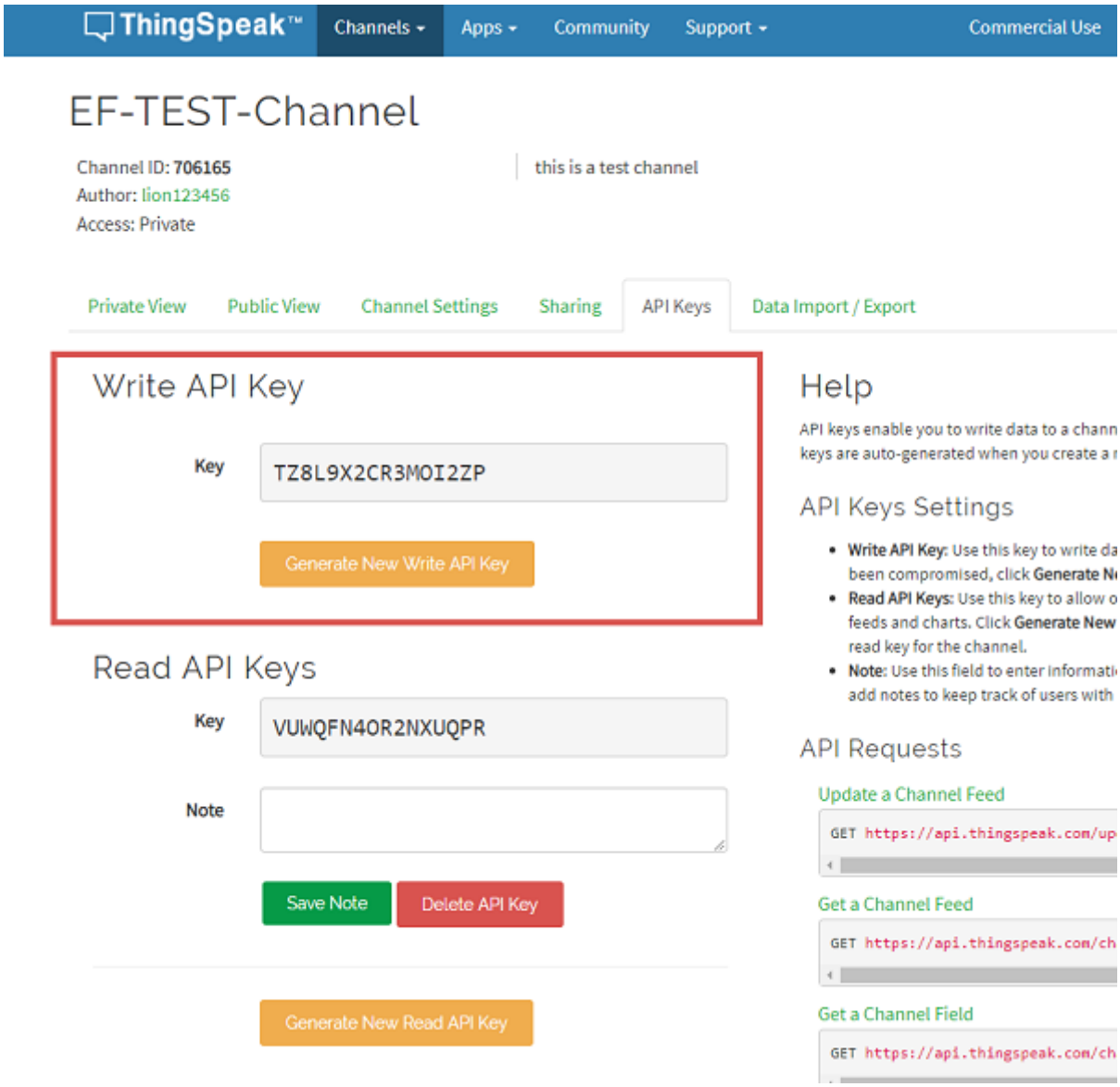
- An project parameter design list be showed. If there are more parameters, you need to choose more fields.




- Then, click on save channel.
- But here you will find nothing. You can find the data of what we have uploaded here:



- Click on API KEY, parameters as below be showed.

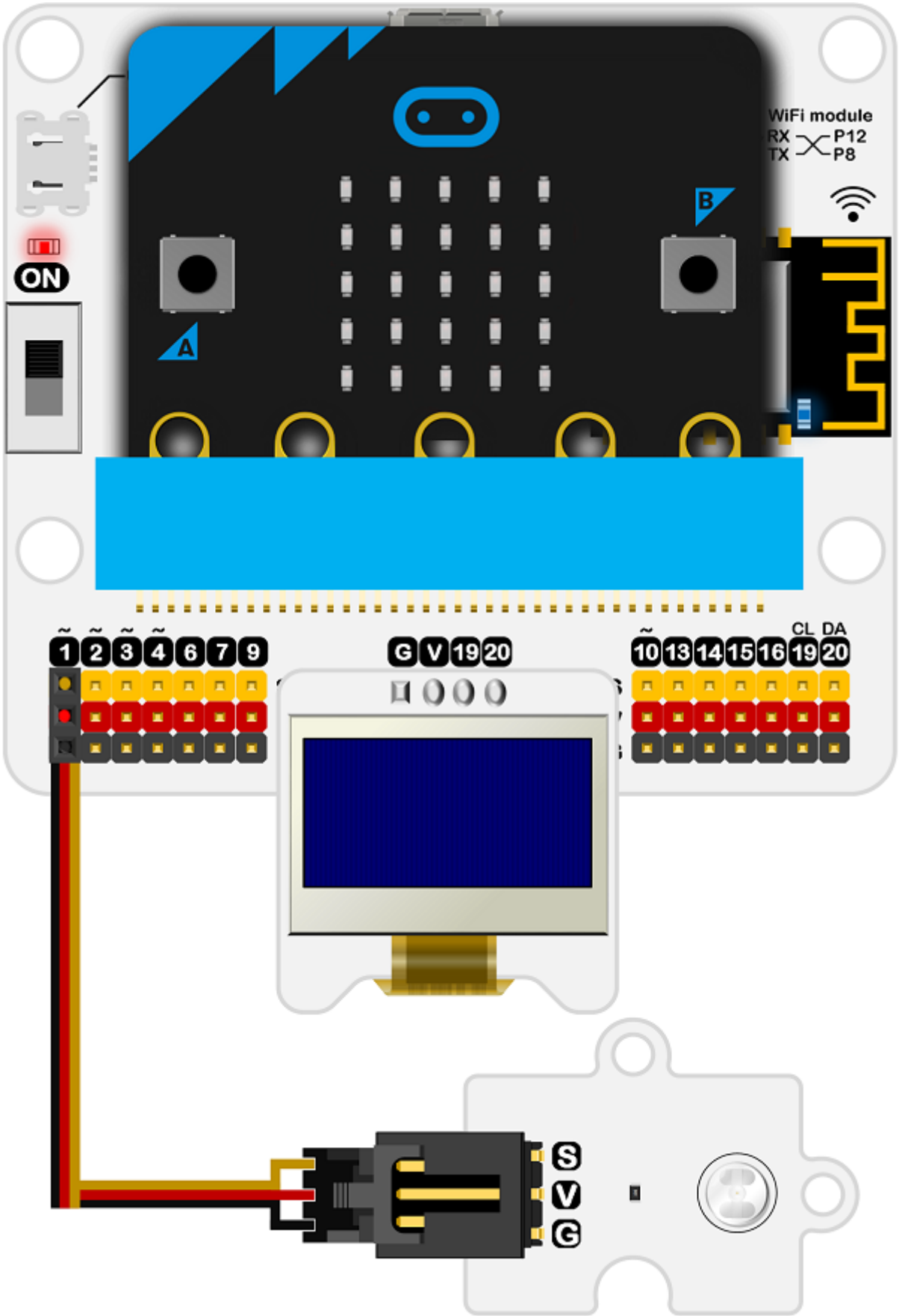



## 3.2. Coding

---

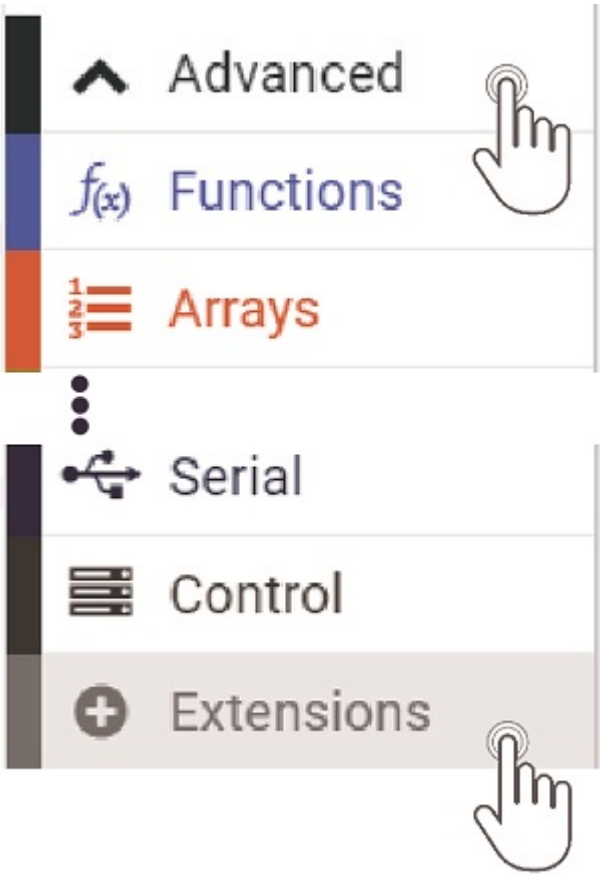
### Connection diagram:

-Connect the light sensor to P1 port.

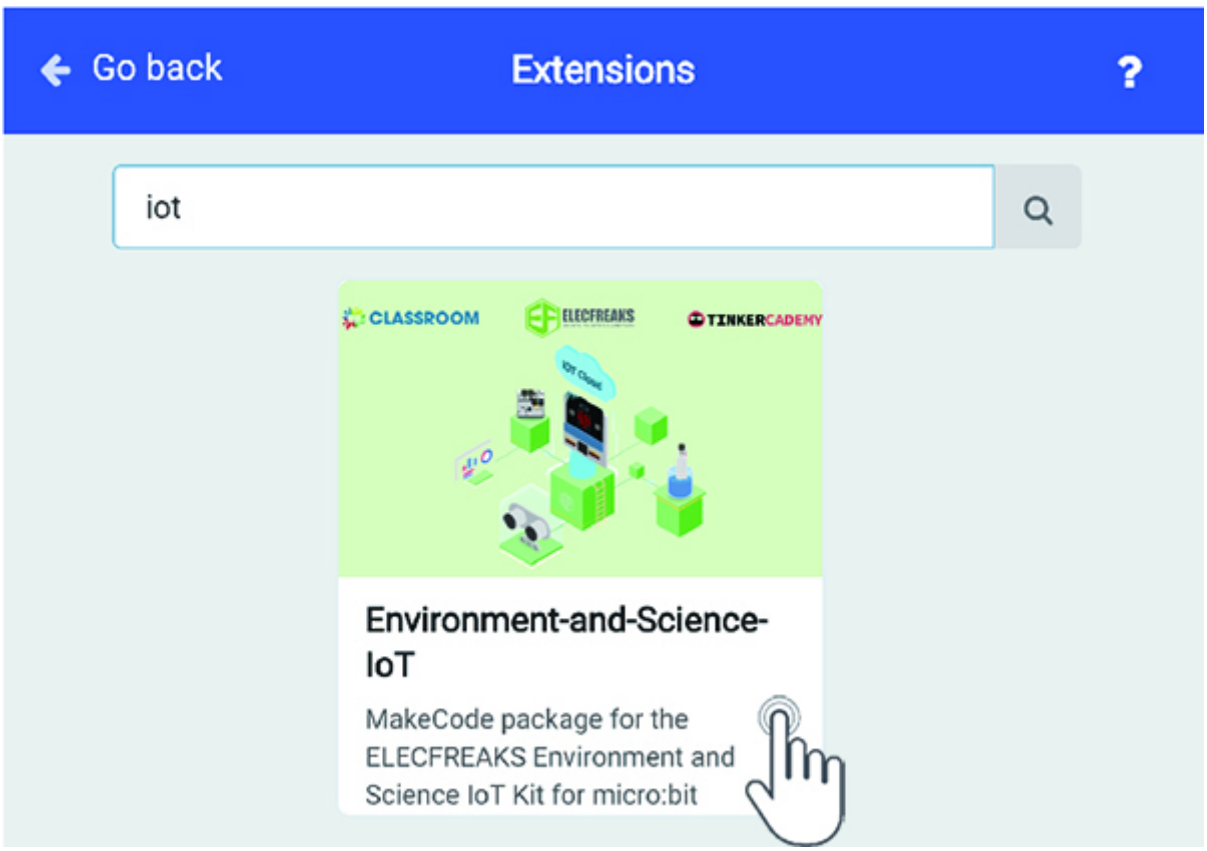



## Step 1

- Click on “Advanced” in the MakeCode Drawer to see more code sections.




- We need to add an extension for coding to the IOT. Click on the “Extension” at bottom of coding drawer, then Search for “IOT” and click on the IOT package to add it to your project. (As below picture)

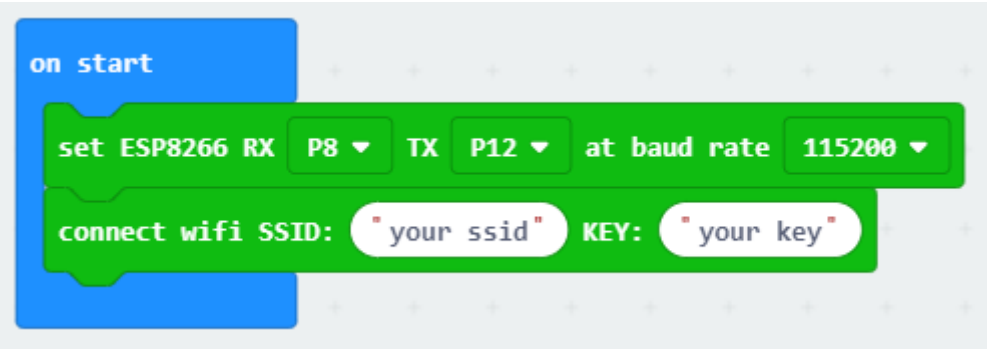



Note：If you get a warning telling you some packages will be removed because of incompatibility issues, either follow the prompts or create a new project in the Project file menu.

## Step 2

Snap the `set 8266` block into the `on start` and choose RX `P8` TX `P12` `baud rate` 115200`.

Snap into the `connect wifi` block， write in your wifi name and the key.



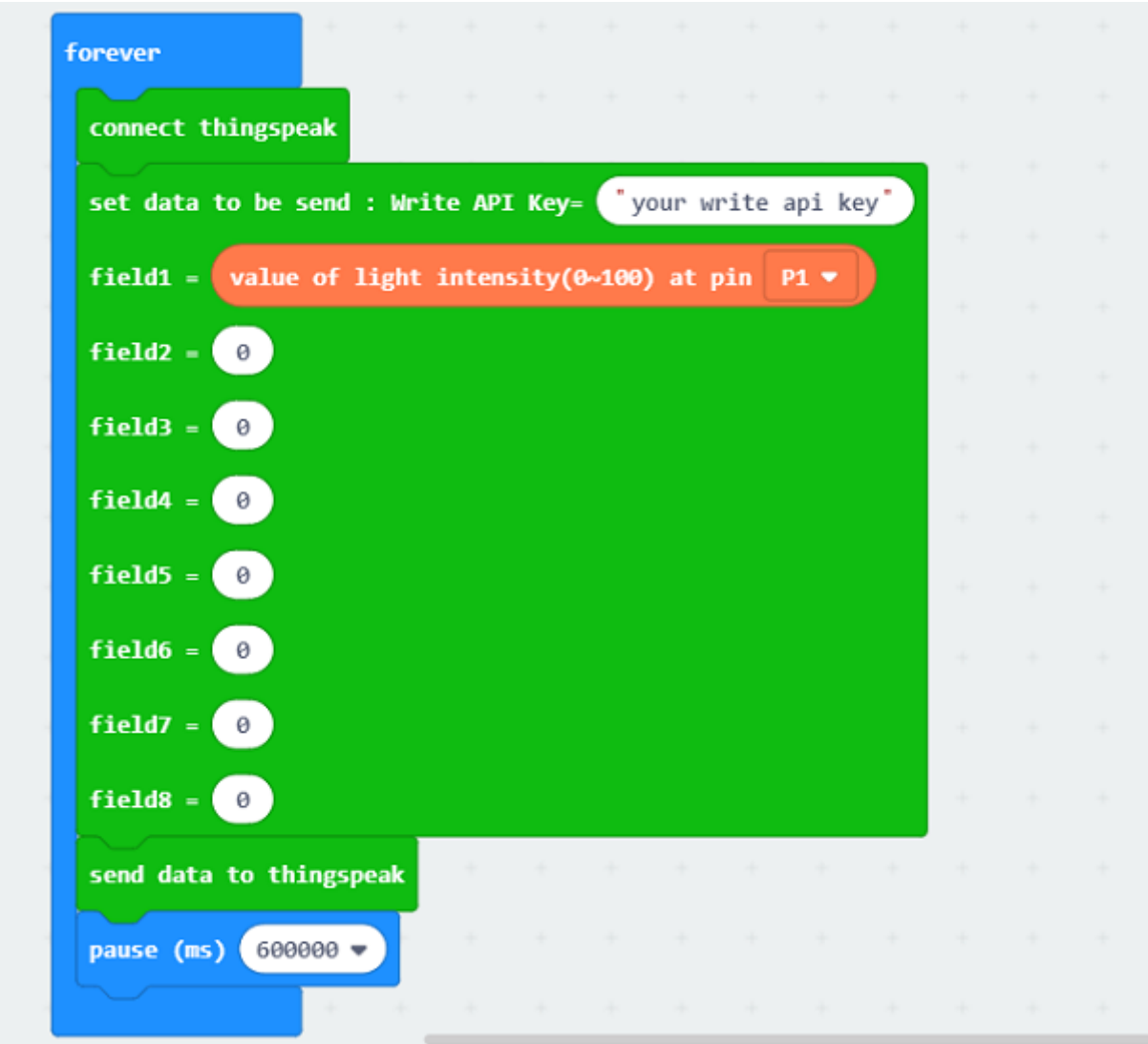

*Note： The 8266 module can't support the 5G, please conenct it to 2.4G*

## Step 3

Snap `connect thinkspeak`, `set data` and `pause` blocks into the `forever`.

Write `write api key` into `set data` and snap light intensity to the field1.

Pause `60s` and sending every minute.




## Step 4

Initialize OLED `initialize OLED` in `On start'to display the IoT working state.

In `serial on data received', when the serial port receives the data, insert the display building blocks, and display the serial port to read characters until it encounters a newline.


## Program

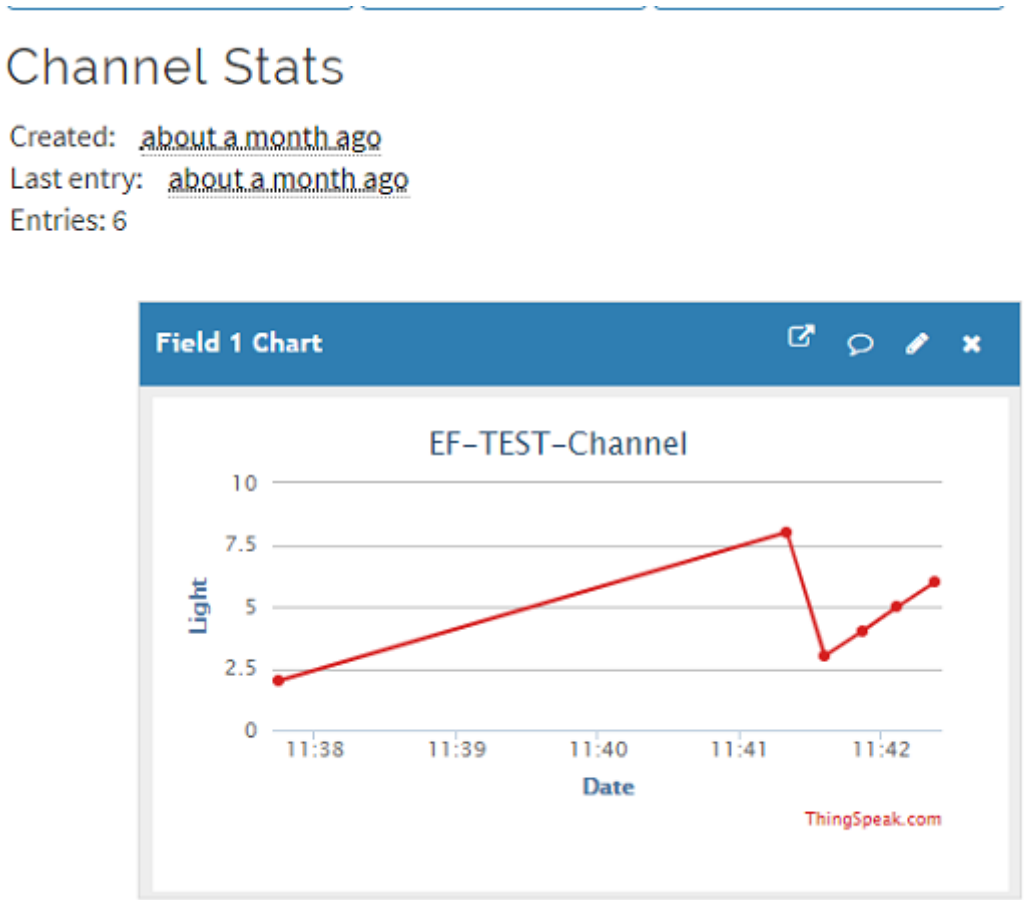
program link：https://makecode.microbit.org/_WukWe0VL3bJX

If you don't want to type these code by yourself, you can directly download the whole program from the link below:

## Result

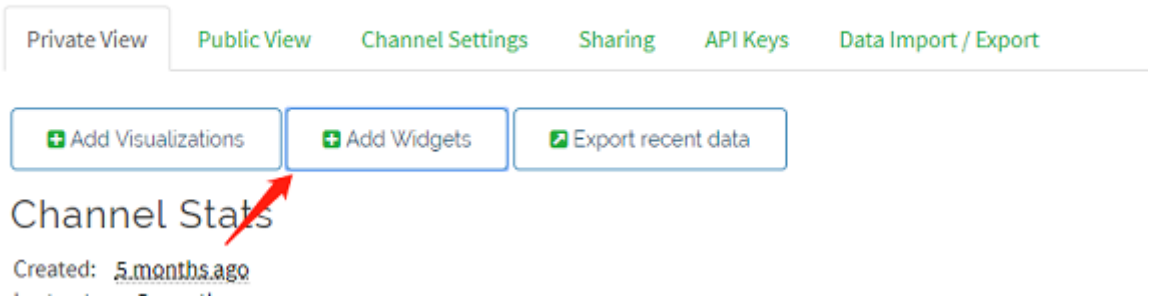
Uploading the light intensity to the thingspeak every minute.

You can check all datas on the thingspeak.

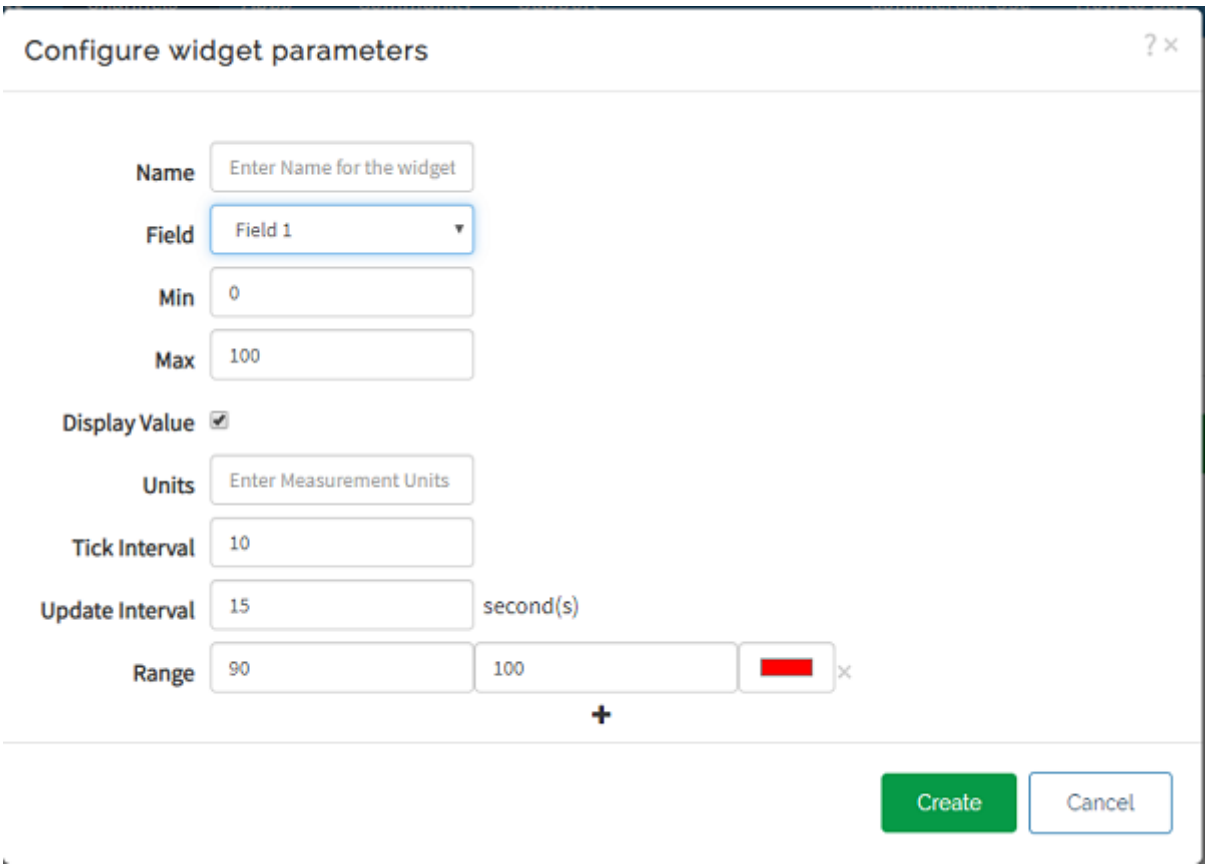



## Thingspeak Add visulization

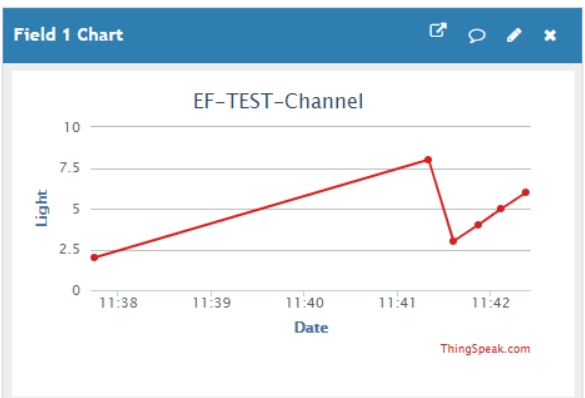
—Click the visulization button to choose the three choices: [stopwatch],[Numeric Display], [Threshold indicator light].

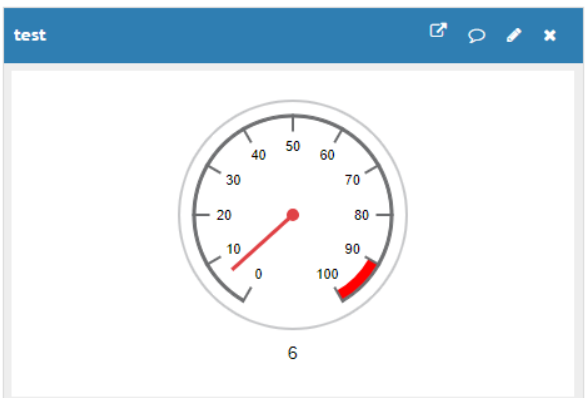



-Choose types of the stopwatch and click NEXT, write the name, filed, the maximum and minimum value of the chart, then click Create to create a new chart.

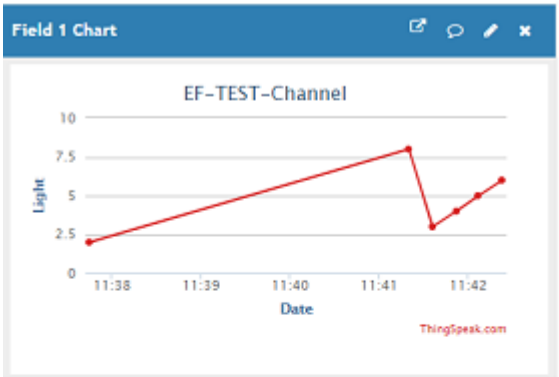


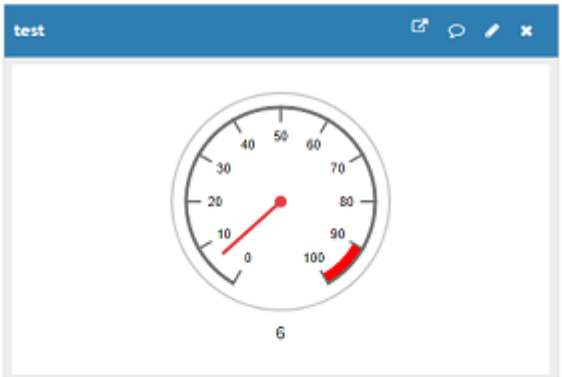

Both the chart and the value of Field 1 changed.

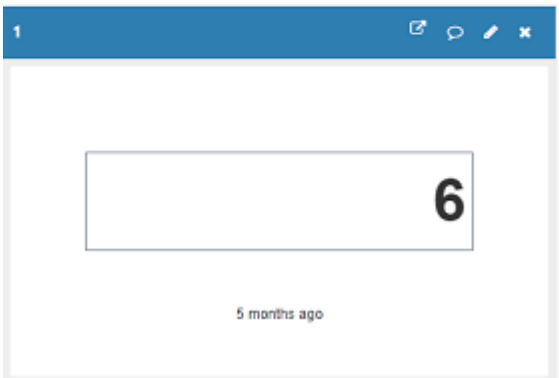


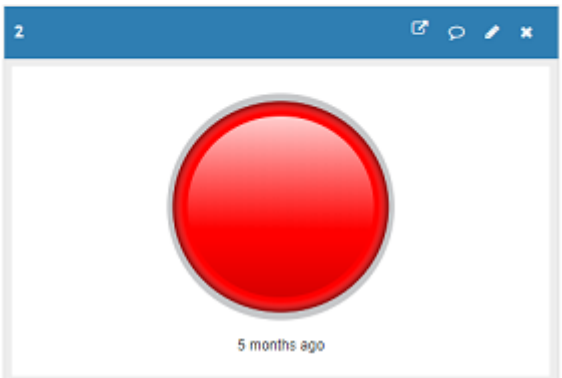



-Add the numeric display and Threshold indicator light in the same way for the chart.



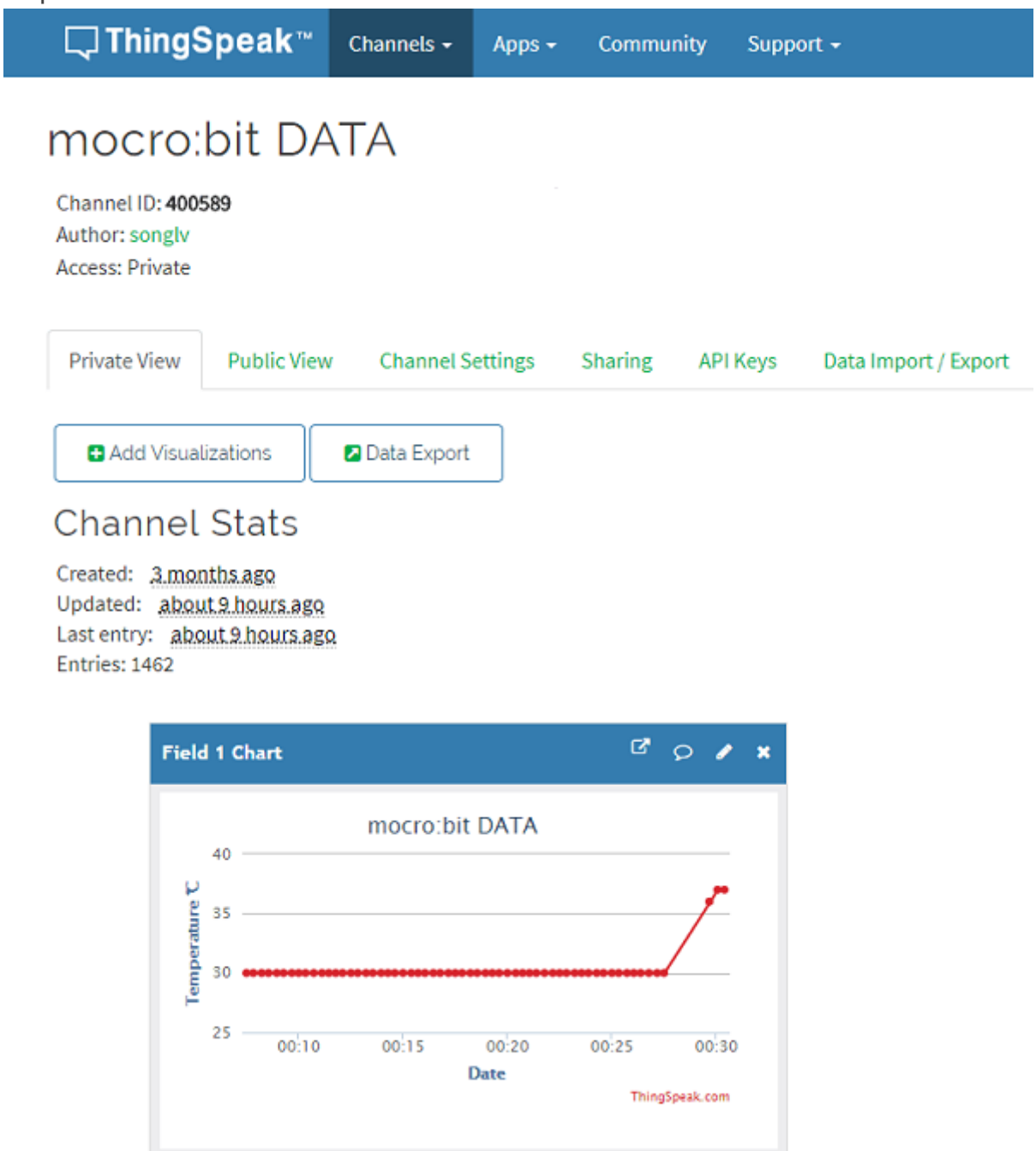



# 4. The second part: Use the IoT kit to male a temperature alarm through Thingspeak and IFTTT.

## 4.1. How to Send Temperature Threshold Value Alarm Email via IFTTT


- In the article How to Send Micro:bit Data to ThingSpeak IoT Platform, we have talked about how to use micro:bit to upload data to Thingspeak IoT platform. In this article, we will talk about how to use IFTTT to send micro:bit temperature thresholod value alarm email.
- What is IFTTT?
- IFTTT is the abbreviation of “if this then that”. In fact, it causes a series of chain reaction to your website behaviour with the goal of “Put the internet to work for you”, which brings you more convinience in usage. IFTTT aims to help people take advantage of public APIs of different websites in order to link websites(like Facebook、Twitter,etc.) or Apps together to complete your task. Thus, everyone can become a programer of the whole Internet without writing a program. IFTTT connects all kinds of information through process and then centrally present your desired information to you, which solves the problem of miscellaneous information and receive or focus on important information. According to IFTTT, the operation of “this” is called “Trigger”, that is to say your behaviour in a certain website; while “that” means another behaviour “action” caused by the chain

reaction. Those triggers and actions all based on a certain website, which is called “channel” in IFTTT. The whole “if this then that” action is defined to be “Task”. Let me explain it to you with an example. In IFTTT, users can realize website chain reaction through creating and implementing a “Task”. For example, if you have just uploaded your temperature data to Thingspeak using micro:bit, when the temperature arrives at a threshold value, it will activate the trigger so that to implement your designated action: send an email to your mail box. Send micro:bit Temperature Alarm Email via IFTTT First of all, please make sure that you have uploaded your temperature data from micro:bit to Thingspeak successfully. If you don’t know how to do that, you can read this article for help.

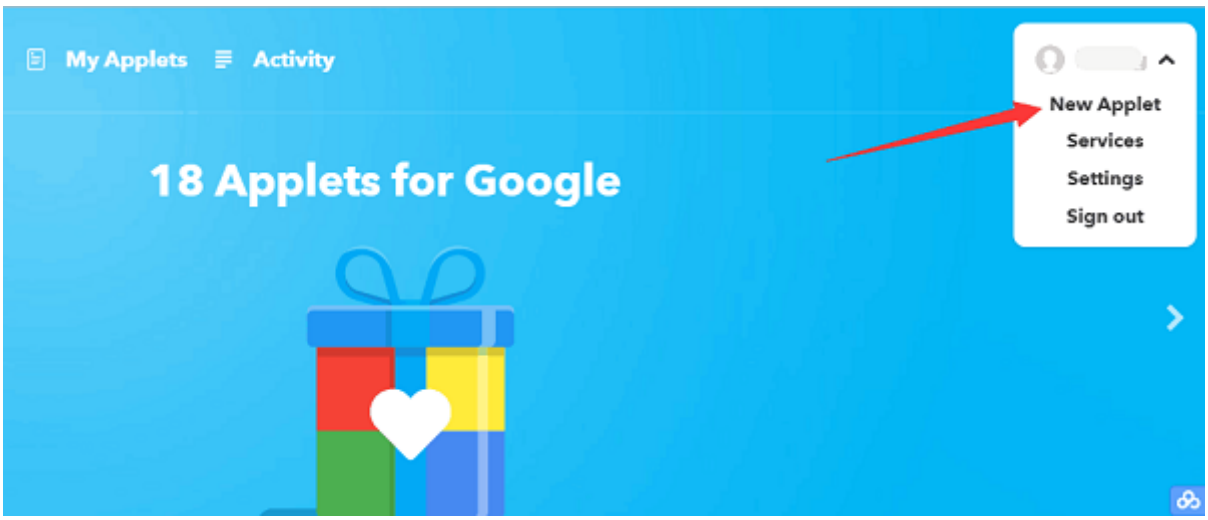



## 4.2. Step 1: Register an IFTTT Account

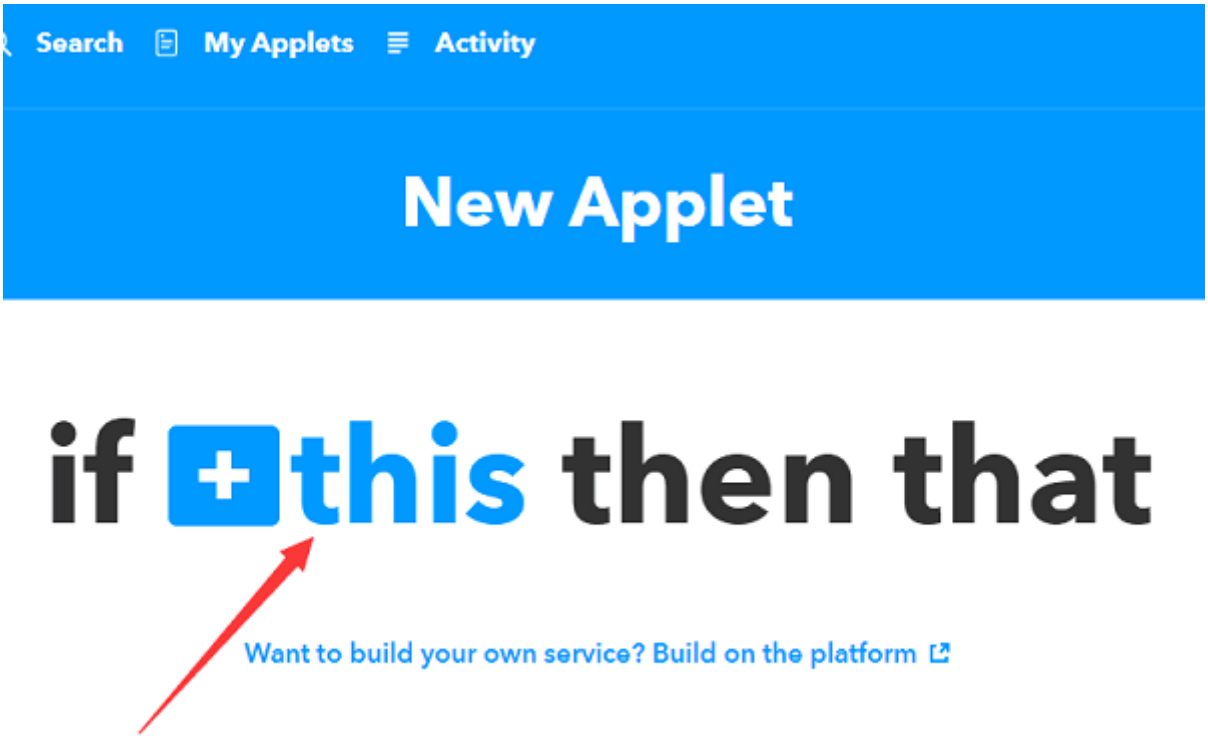
Log in IFTTT. If you don't have an account, please register one first.


## 4.3. Step 2: IFTTT Webhooks Setting

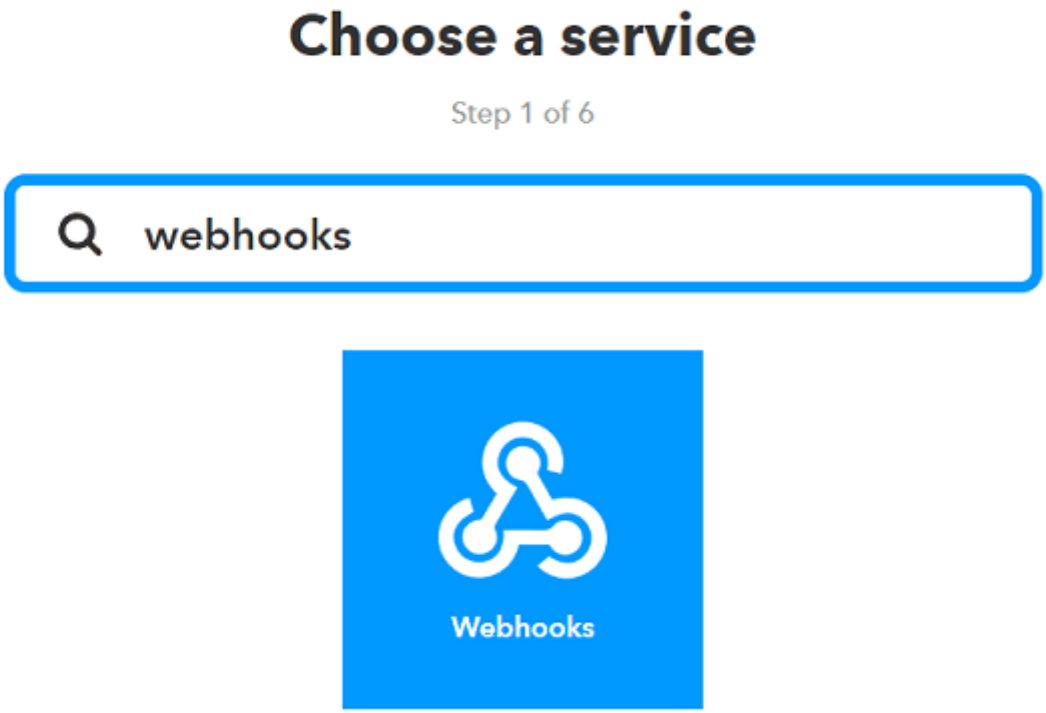
## 4.4. Create an Applet.



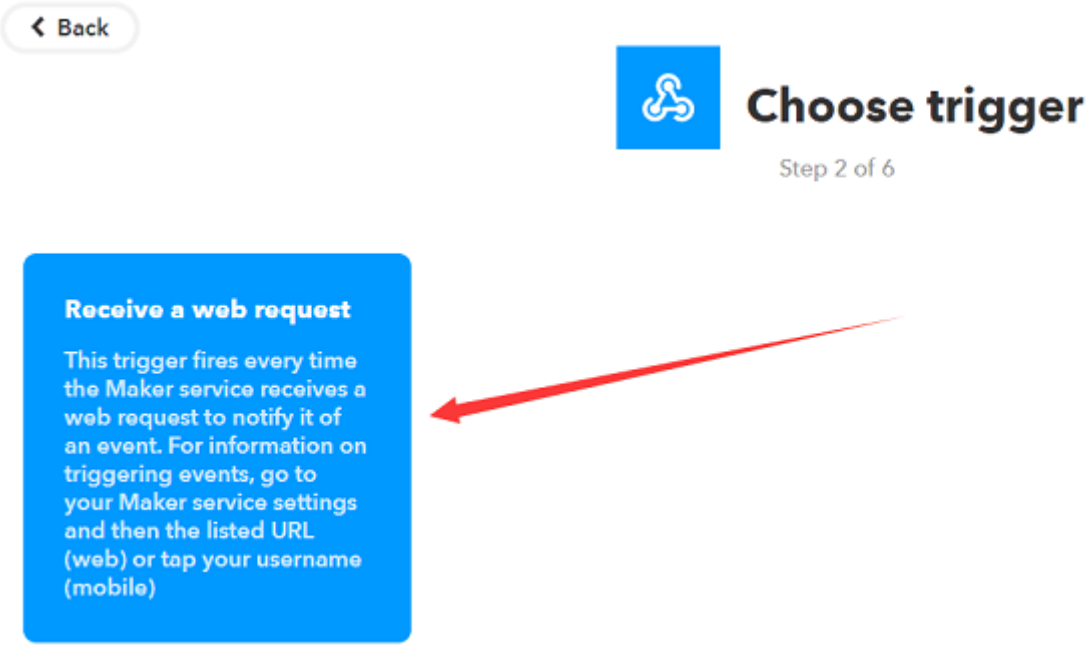

Click on "this".



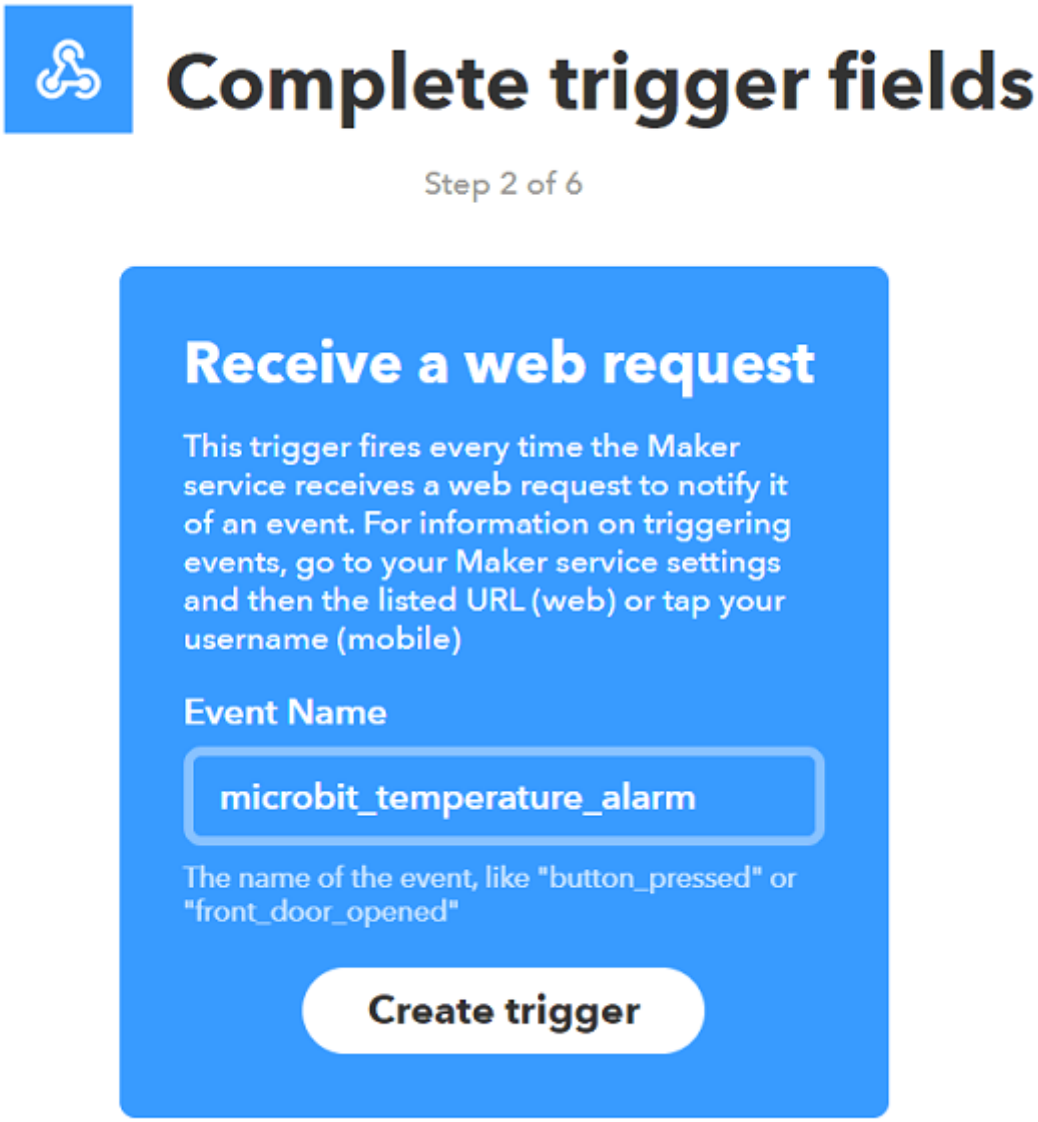

Search for “webhooks”.




Choose trigger.

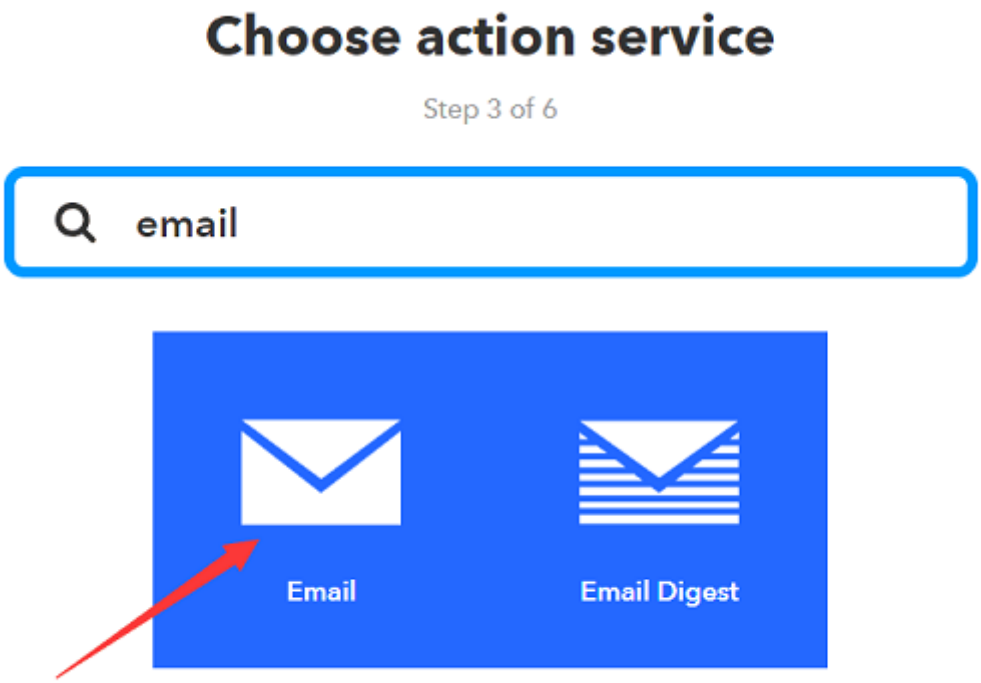



Name this task. Here we call it “microbit_temperature_alarm”.

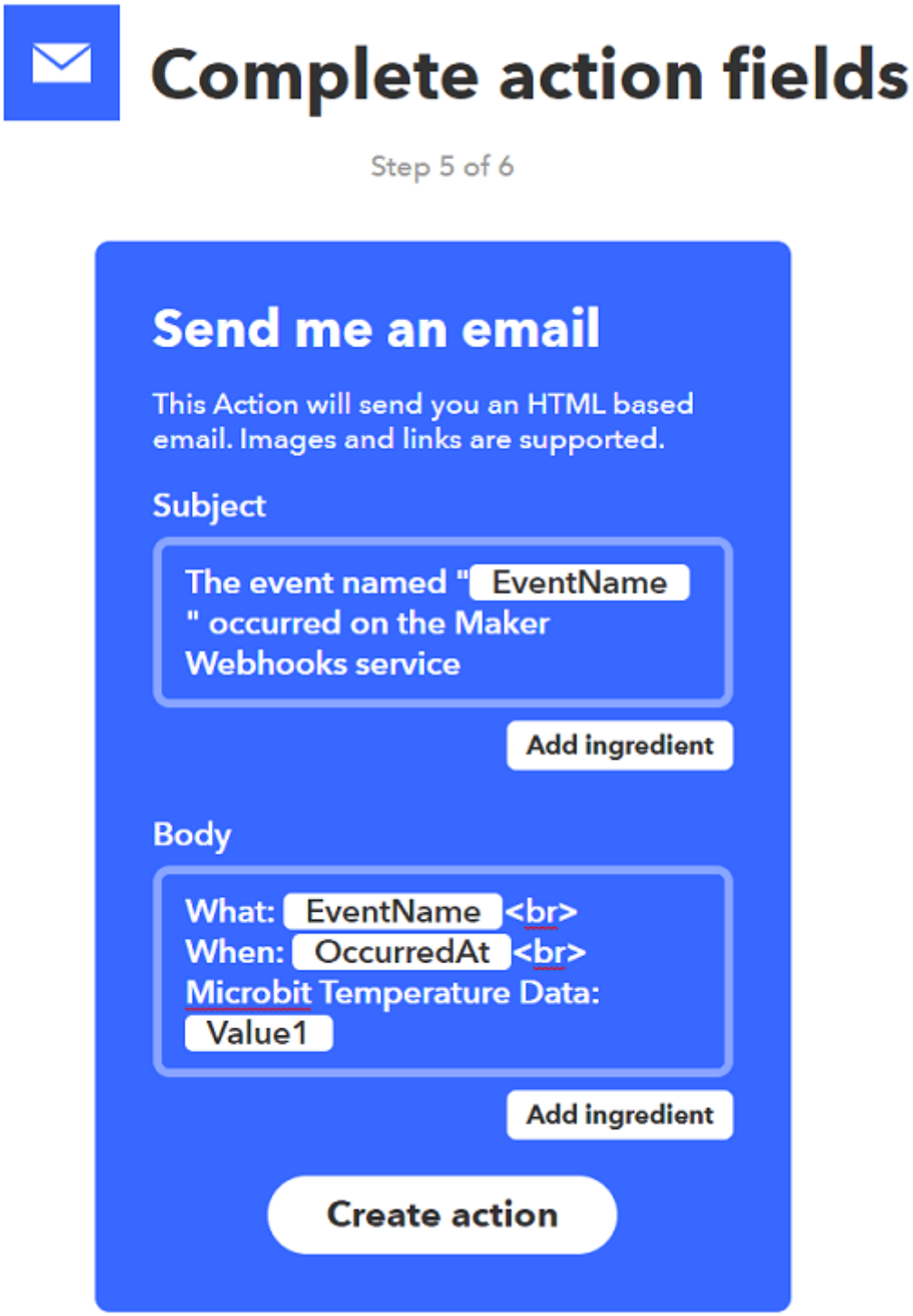


Once we have completed trigger setting, click on “that”.

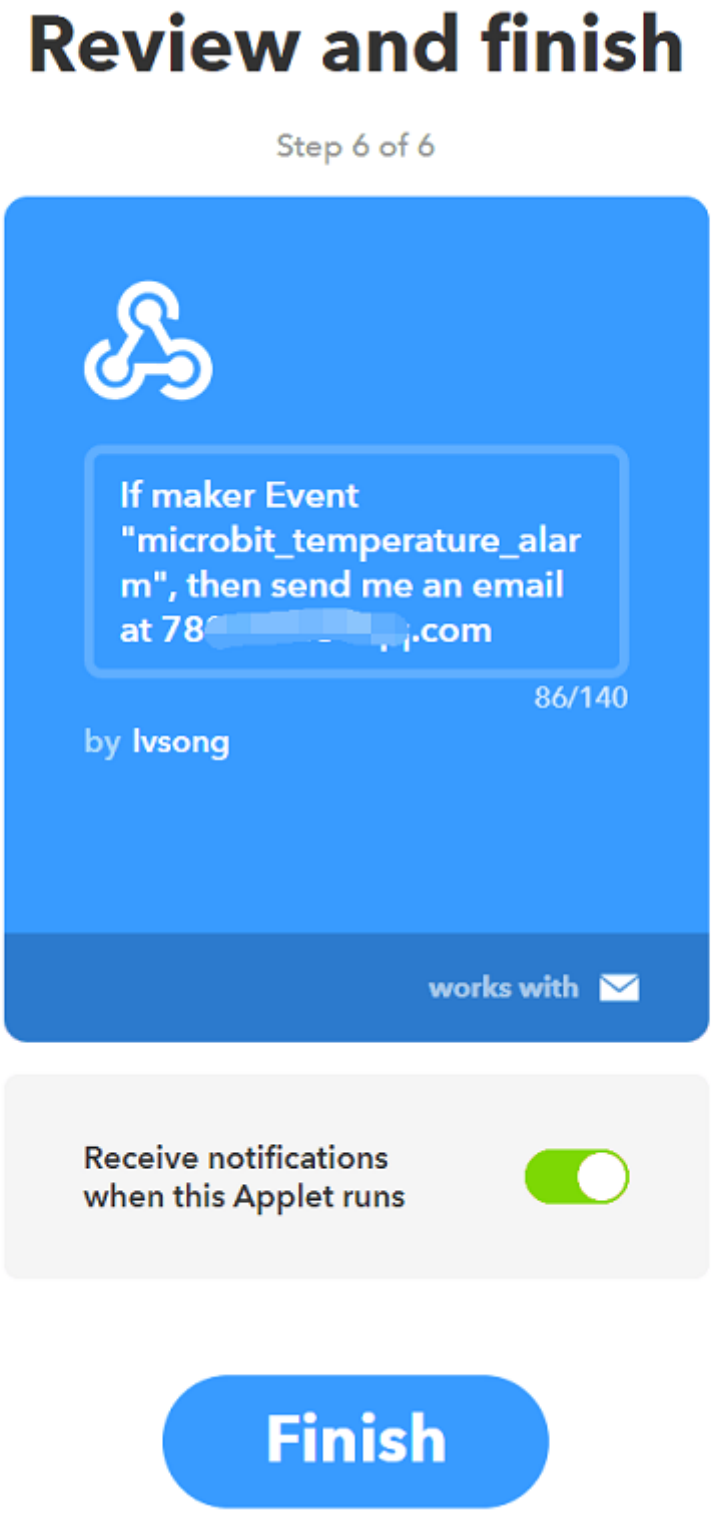

Search for “email”.

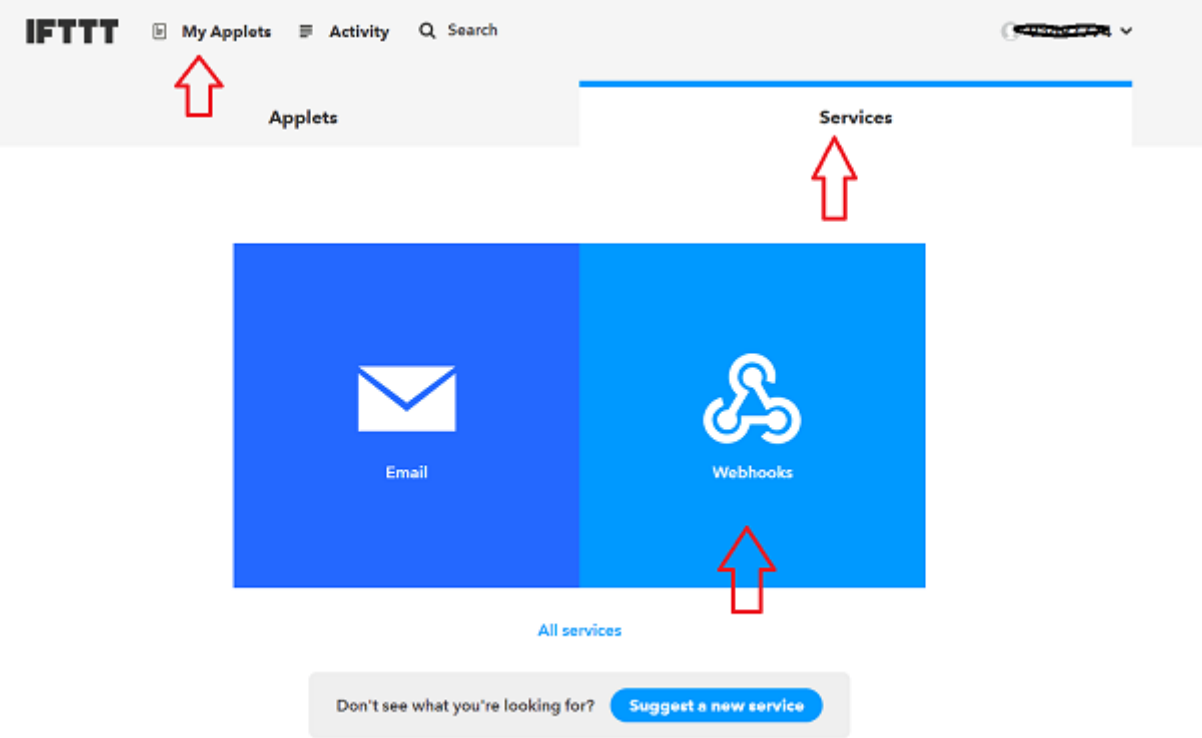


Fill in your email contents and pay attention to the format showed, among which {{}} allows us to extract datas with the same name from Web Request and then forward it to the email.

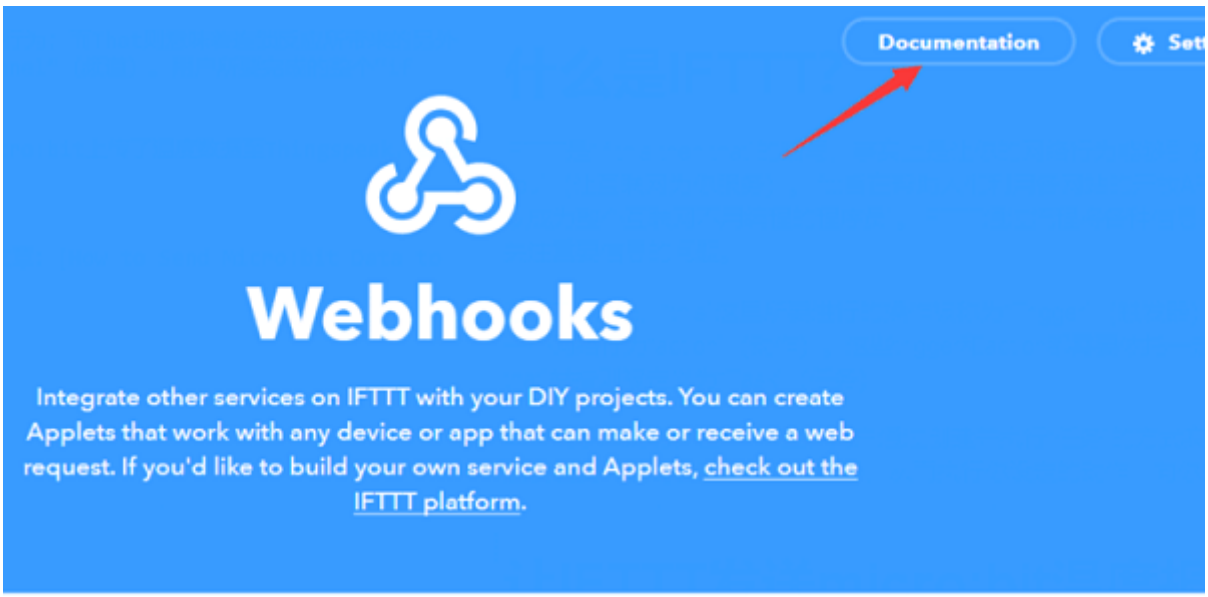


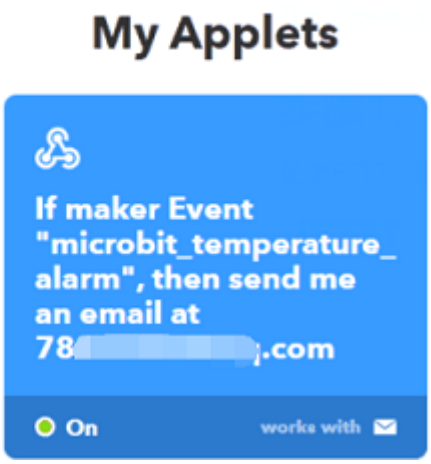
Completed.


Review and finish
Step 6 of 6
If maker Event "microbit_temperature_alarm", then send me an email at 78[redacted].com
86/140
by lvsong
works with
Receive notifications when this Applet runs
Finish

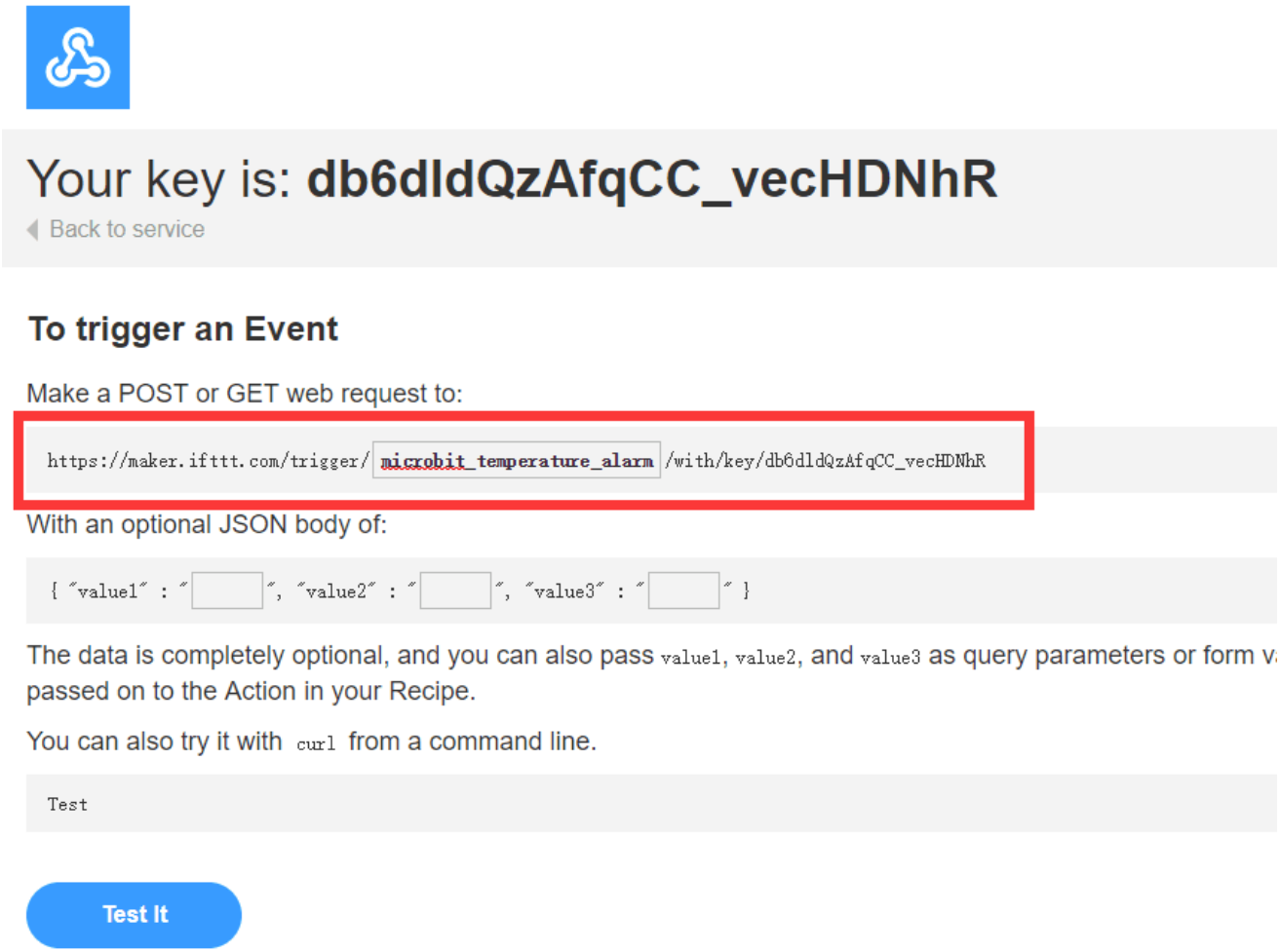



Click on “Documentation”.



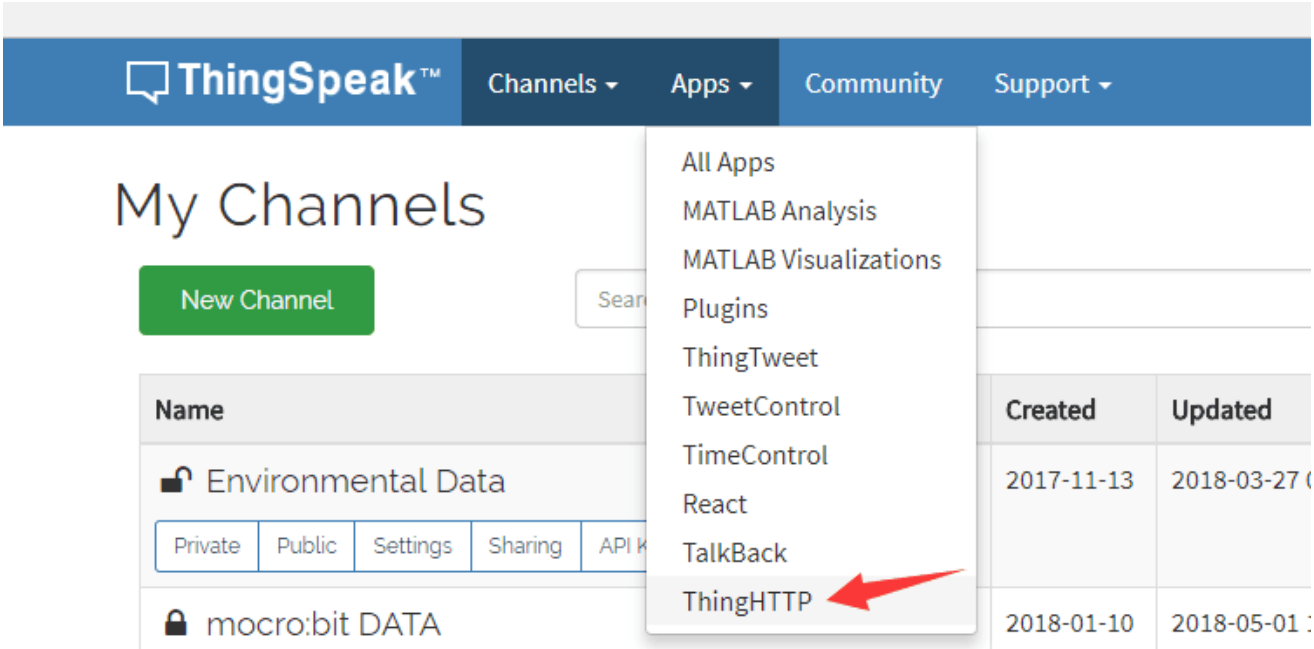




This link is the link of web request. It is very important in the Thingspeak setting later on.




## 4.5. Step 3: Thingspeak Setting

Before you do this, you must have uploaded your temperature data from micro:bit to Thingspeak. If you don't know how to do it, just read this acticle. Firstly, create a a new ThingHTTP service.




Here's the setting of connection with IFTTT:

Apps / ThingHTTP / temperature_alarm / Edit

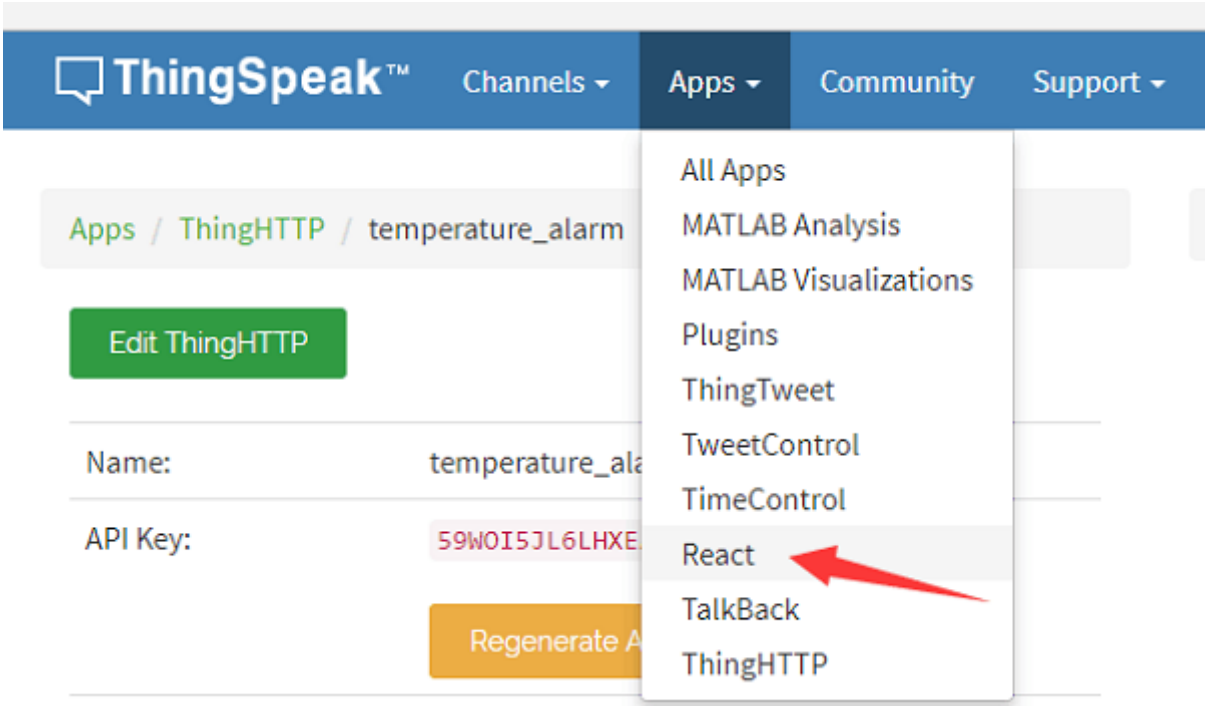
| | |
|---|---|
| Name | temperature_alarm |
| API Key | 59WOI5JL6LHXEAGL |
| URL | https://maker.ifttt.com/trigger/microbit_temperature_alarm/\ |
| HTTP Auth Username | |
| HTTP Auth Password | |
| Method | POST |
| Content Type | application/json |
| HTTP Version | 1.1 |
| Host | |
| Headers | Name<br>Value<br>remove header |

add new header

| | |
|---|---|
| Body | {"value1":"%%channel_400589_field_1%%"}<br>channel ID |
| Parse String | |

Save ThingHTTP

Note: URL is the link of web request, which has to include a Private Key provided by IFTTT. Content type must be JSON, because the expected format od IFTTT Maker Channel is JSON. Within Body, you can invoke any data in Channel. This is the data that is going to be sent to IFTTT with the format as follow: {“value1”:”%%channel_138112_field_1%%”} For more details about ThingHTTP App, please refer to https://ww2.mathworks.cn/help/thingspeak/thinghttp-app.html. In the last, create a React service.




For more details about React APP, please refer to https://ww2.mathworks.cn/help/thingspeak/react-app.html.

## Step 4: Test

Till this step, you have already completed all of settings. Now let’s test it! If the temperature has not arrived 30 degrees yet, you can hold micro:bit with your hands to improve its temperature.

Apps / React / alarm / Edit

**React Name** alarm

**Condition Type** Numeric

**Test Frequency** On Data Insertion

**Condition** If channel

mocro:bit DATA (400589)

field

1 (Temperature ℃)

is greater than

30

**Action** ThingHTTP

then perform ThingHTTP

temperature_alarm

**Options**

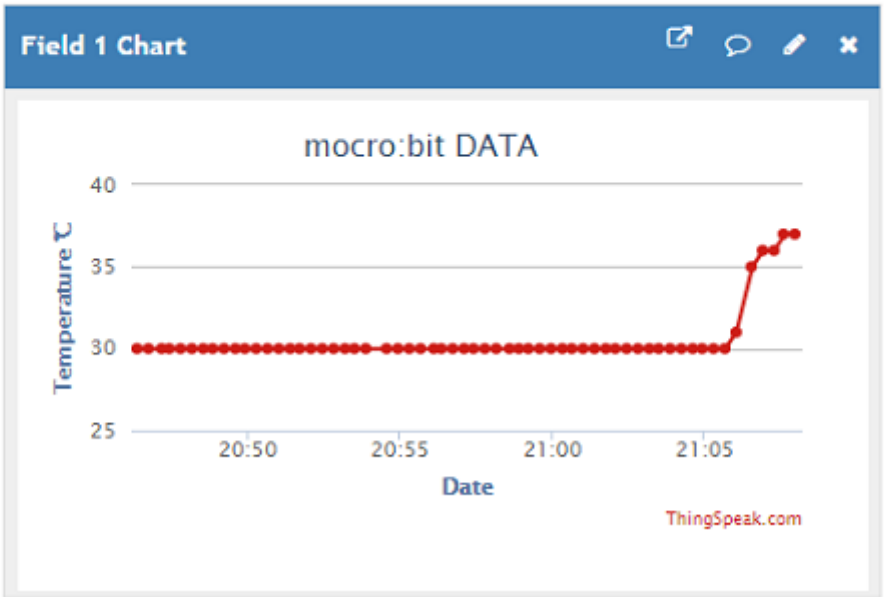
- ○ Run action only the first time the condition is met
- ◉ Run action each time condition is met

Save React

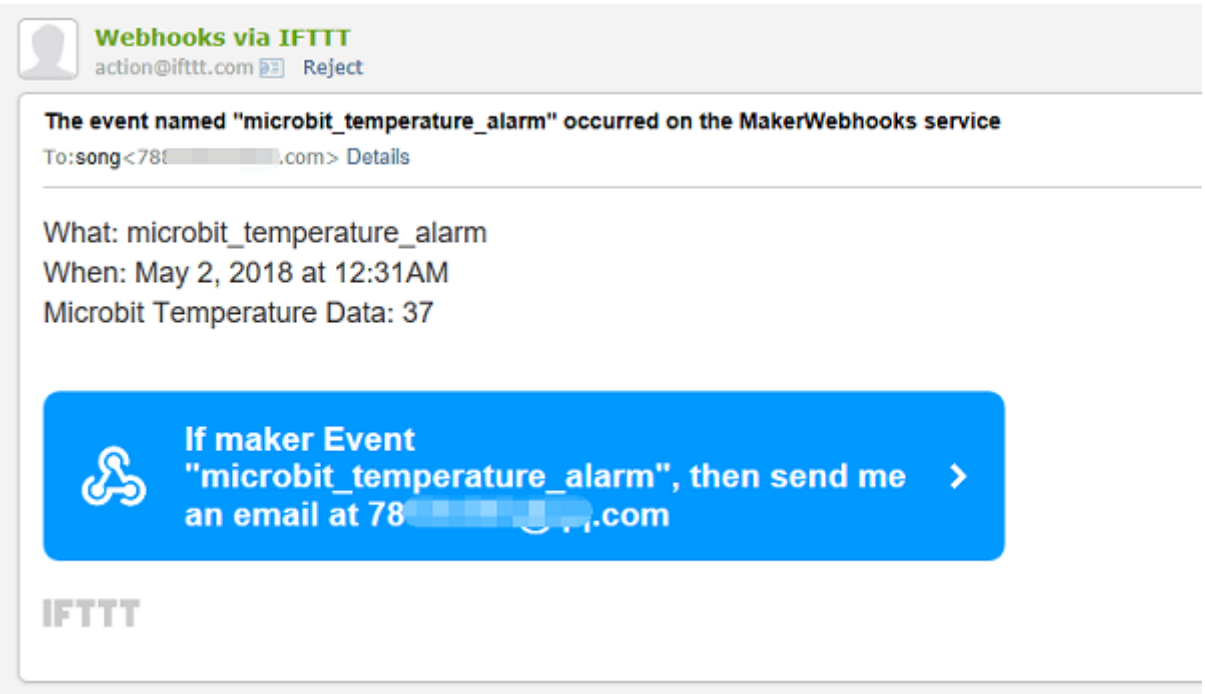
We can see from the data of Thingspeak channel that the temperature has surpassed 30m degrees.

## Channel Stats

Created: 3 months ago
Updated: 2 minutes ago
Last entry: 2 minutes ago
Entries: 1333




Check your email box and see if you have received an email from IFTTT!



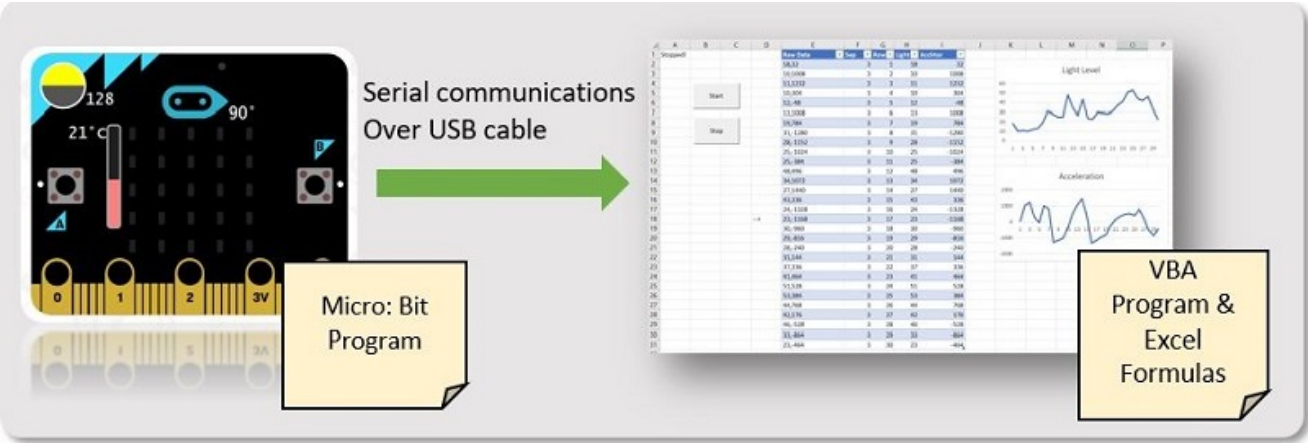
# 5. The third part: Use IoT and Excel to make date visualized.

## 5.1. Excel and micro:bit - Hacking for fun and creativity!

Our goal is to collect some basic data from sensors with the micro:bit and make the data visualized in Excel.

For this purpose, we will do with the below three steps:

1. We will program the controller to collect some sensor data and send it over its' built in serial communication port.
2. We will connect the micro:bit to the PC's serial port.
3. We will write a small program in Excel that reads the data from the serial port into the grid.




### The Experiment-from micro:bit to Excel

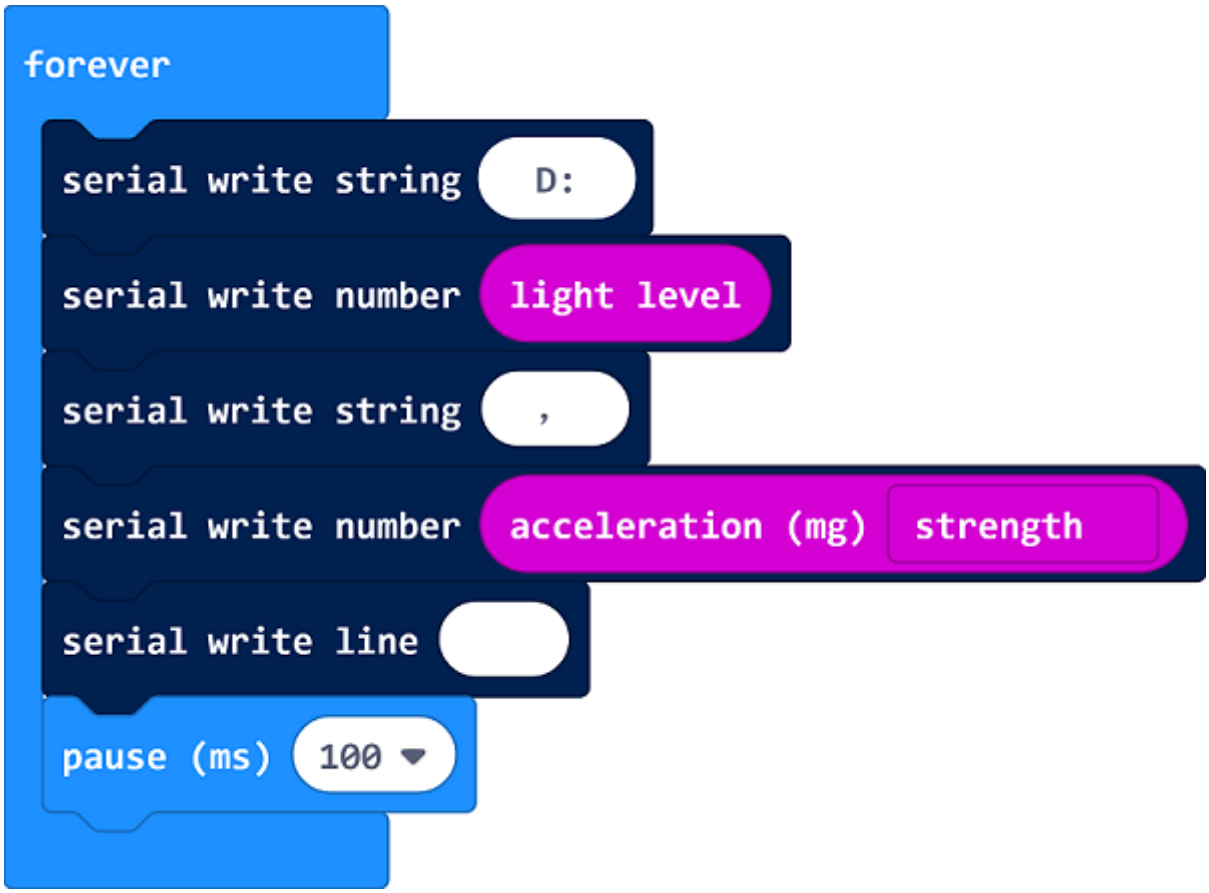
### The first step: programming the micro:bit

Programming the micro:bit is the easiest thing you'll ever do. Microsoft actually has a web-based development environment ready for you (Microsoft is one of the founding partners of the micro:bit).

All you have to do is, go to www.makecode.com, select the micro:bit as your device, and write a little program using a visual "Block-based" programming language.

The program we'll write for this experiment will simply collect data from two sensors that we can easily play with – acceleration and light level, and send a sampling of the sensors over the serial communication port every 100ms.

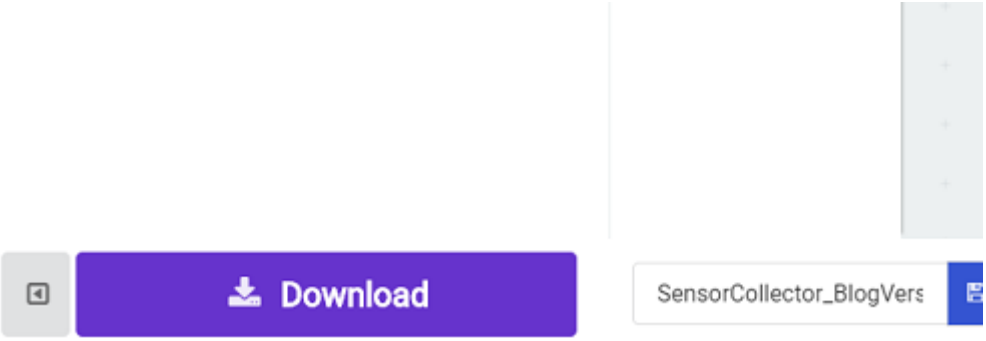
5.2.




## The second step:download the program to the micro:bit

All you have to do now is load the program into your Micro:Bit. To do that, connect your micro:bit to the PC using a USB cable, which will make it appear like a removeable disk drive.


Then download the HEX file from within the MakeCode environment and save it onto the device. A few seconds later, the program will start running.

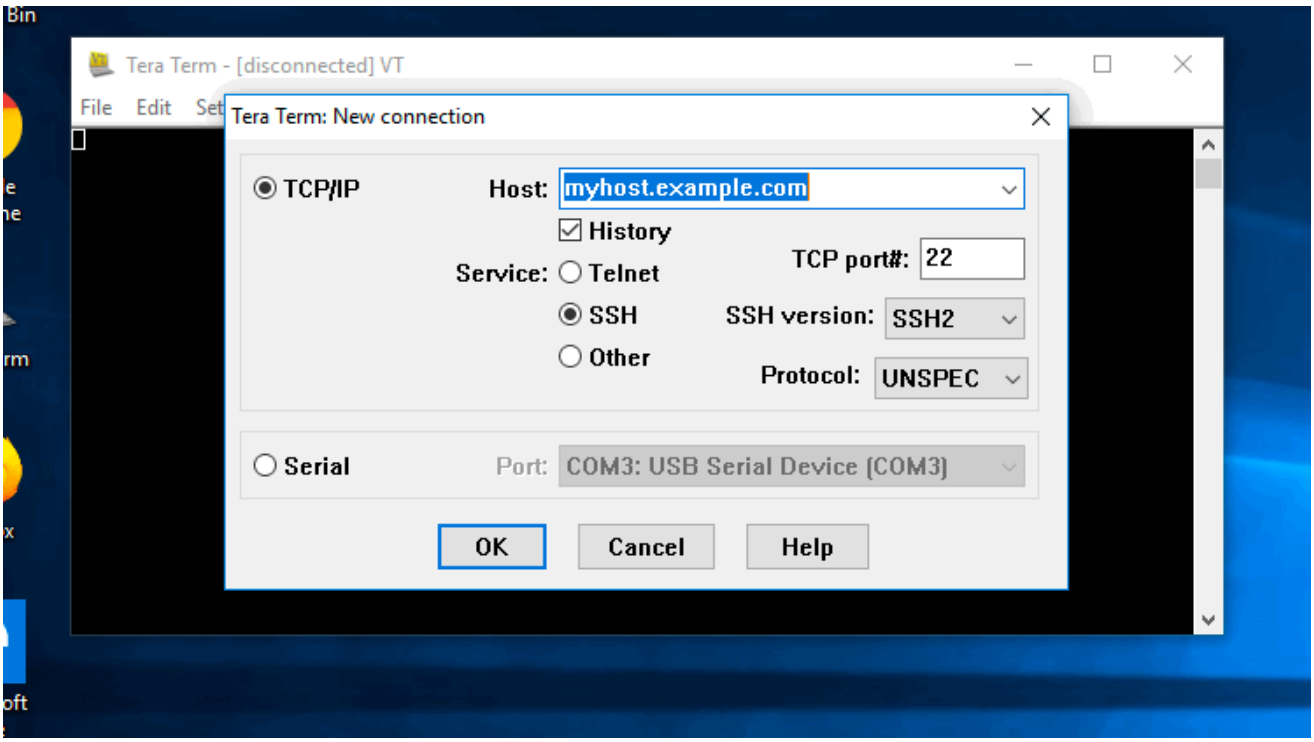



You can also download it here: (https://makecode.microbit.org/_cjvC4RU1CVUD)

## Connecting it to the PC

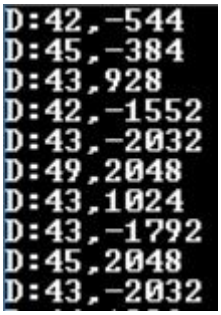
Now that we have our controller running and sending data, and before we try it out in Excel, it's a good idea to verify that the PC can indeed see the incoming data stream. To do that, you'll need to follow the instructions on https://www.microbit.co.uk/td/serial-library which basically means you need to do two things:

1. Install a driver, which will make the micro:bit "appear" as a serial port on your PC.Youcan download it here: Download drive program

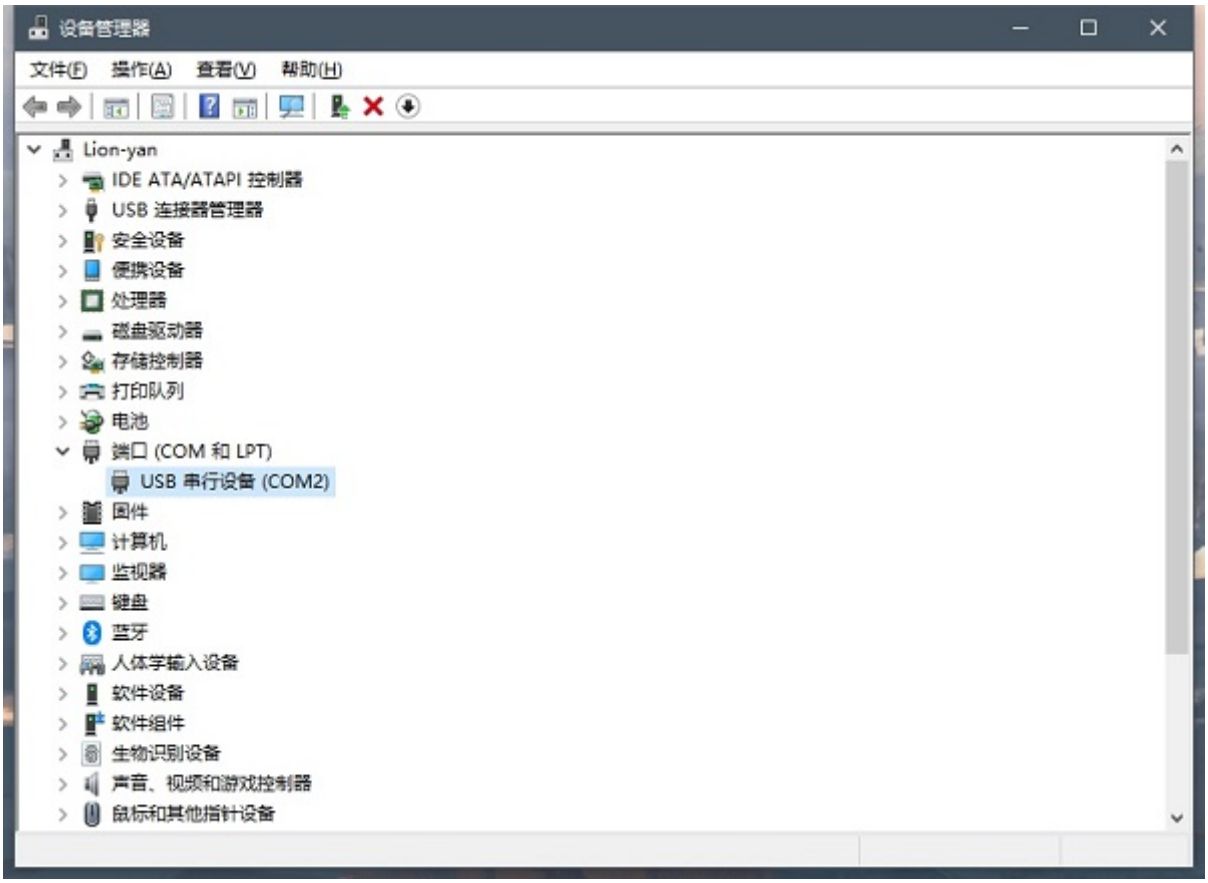



1. The serial communication terminal simulator has to be used for the test. You will need to configure the right COM port. On my environment, it was configured to COM3. The sample code in Excel assumes that, so if yours is different, you'll need to modify the Excel code later on to reflect the right port.

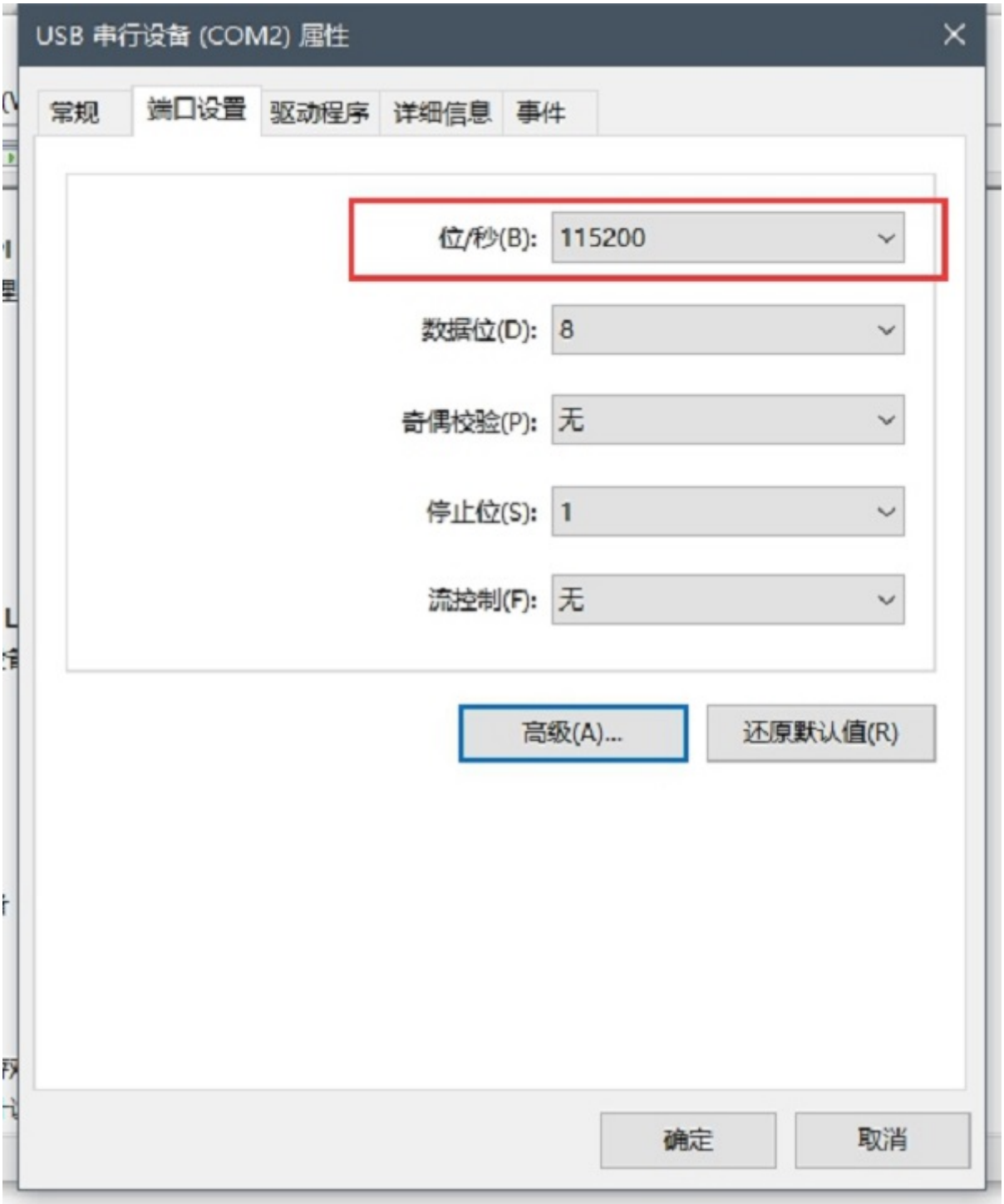
Once you do that, you should see a stream of data that looks something like this inside your emulator:



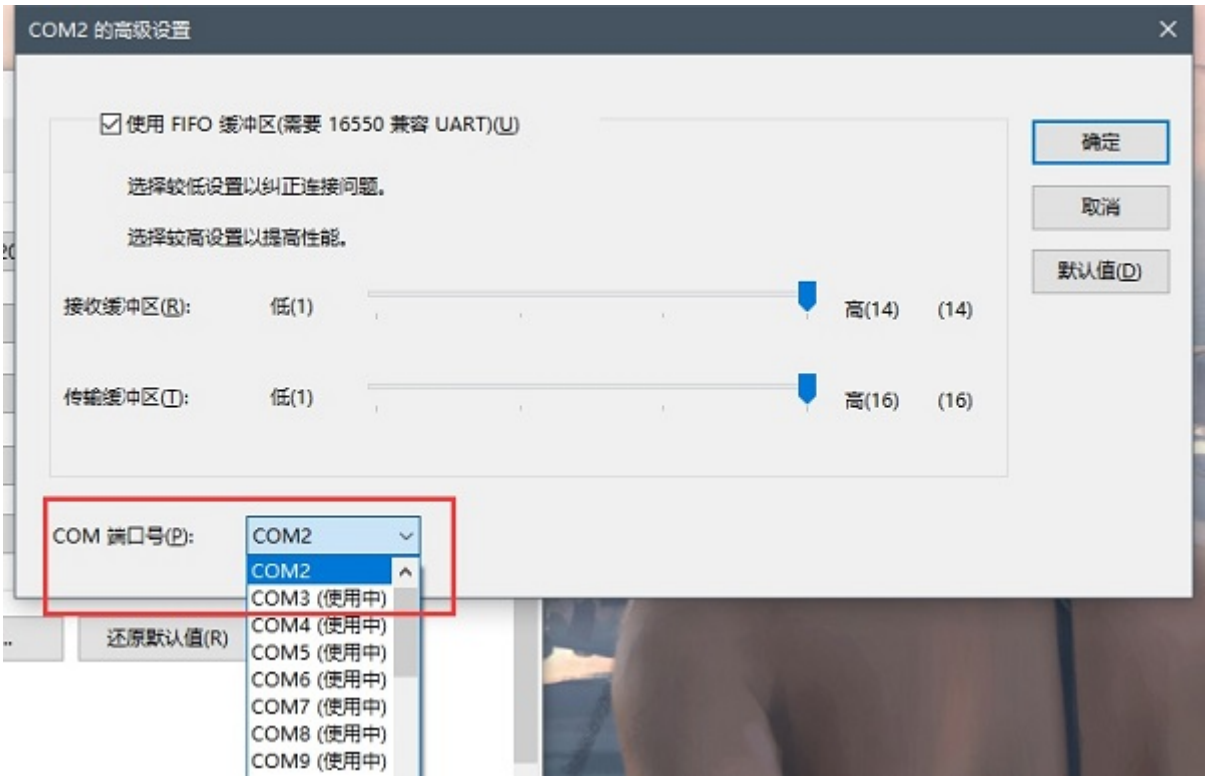

3.If it passes from the terminal simulator, go to “device manager”-choose “port”, find “USB serial device”.



Right click the mouse to find propertities to choose the port setting, change the first data to “115200”.



Click “Advanced”, set the “COM port” to “1” or “2”, click to confirm the setting in turns.




**The incoming data flow - light level and acceleration**

## The fourth step: programming in the Excel!

Now that we have a stream of incoming data, lets get it into Excel. The spreadsheet comprises of two parts :

1. First part: some VBA code that is used to communicate with the micro controller
2. Second part: some basic grid data manipulation functions which are used to break the data points and charts it.

You can find a copy of the working spreadsheet:https://techcommunity.microsoft.com or find it here: check here to download

While you open the copy of the Excel, it reminds” The macro is forbidden”, clicking “start macro” is important.

Because this is an endless stream of data, for the purpose of this experiment, we will iterate thru the last 30 data samples collected.

One more point: When reading from the serial communication port in VBA, the most reliable way to do that is to read byte-by-byte and not whole lines. There’s also some chance of losing some data (depending on speed of communications, speed of VBA execution, etc.), which is why I’ve added the “D:” prefix for each line. If the line we read doesn’t start with it, the line gets ignored as garbage data.

With no further ado, here’s a snippet from the main loop in the VBA code:

```
On Error GoTo ErrorHandler
Open "COM3:115200,N,8,1" For Random As #1 Len = 1

PrevRow = 0
Row = 0
While (StopReading = False)
    EOL = False
    Data$ = ""
    While (Not EOL And Not StopReading)
        Get #1, , COM_Byte
        If COM_Byte = 13 Then
            EOL = True
        ElseIf (COM_Byte <> 10) And (COM_Byte <> 0) Then
            Data$ = Data$ & Chr(COM_Byte)
        End If
        DoEvents
    Wend
    If Not StopReading And Left(Data$, 2) = "D:" Then
        Row = (Row + 1) Mod 30
        Cells(PrevRow + 2, 4) = "" ' Clear previous location indicator
        Cells(Row + 2, 4) = "-->" ' Display new location indicator
        Cells(Row + 2, 5) = "'" + Right(Data$, Len(Data$) - 2) ' Enter data into the grid as a string
        PrevRow = Row
    End If
Wend
```

Few things to note in this code snippet:

1. We open the COM3: port at 115,200 baud (the speed at which the Micro:Bit sends data).
2. Reading happens a byte at a time, until end of line (char(13)) is detected.
3. Whenever a line is read, it gets pushed into the grid into the next row in a fixed column. Row numbers are fixed between 2-31 to keep this example simpler.
4. There is a flag used to stop reading. It's triggered by a stop button (from a different Macro).

The best way to understand this code it is to run it in debug mode and step thru it, so go ahead and download the demo workbook and experiment!

Now that we have the data coming into the grid, we're in the plain old good Excel formula and charting territory. Time to do something with the data we are collecting!

To keep things generic, the VBA scripts reads the data as-is into the grid, so Column "E" contains the actual data, as it arrived over the wire. In our case, it's two numbers, comma-separated.

So, first thing we want to do is to break it up into two distinct values per row. The light level, and the acceleration value. I did that on purpose in the easy to read way – used the FIND formula to find the location of the "," separator inside the incoming data, and then used NUMERVALUE and LEFT and RIGHT formulas to break the string apart and convert it into two numeric values.

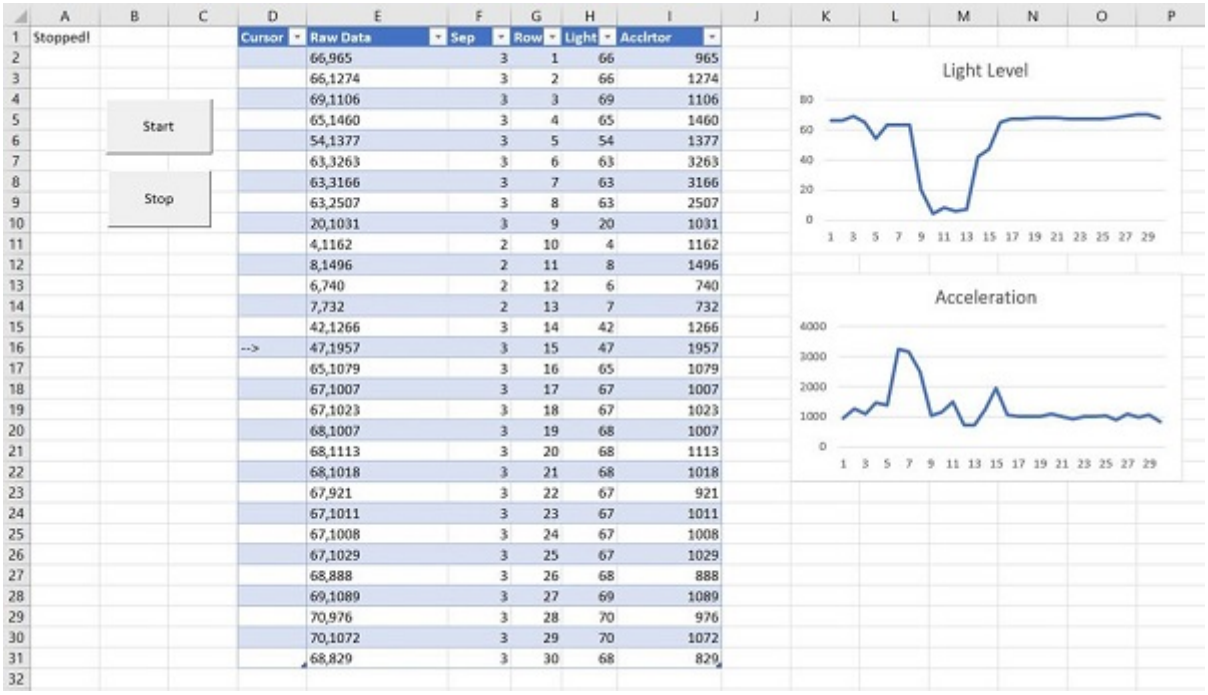
Here is a bit more about the formulas I used to break down the values from the input data string:

`=FIND(",",E2,1)` : Finds the location of the first comma separator inside cell E2 (which contains the raw incoming string of comma-separated values). `=NUMBERVALUE(LEFT(E2,F2-1))` : Takes the left side of the string, up to the comma location, and convert it to a number value.

This gives us a number representing the light sensor value. Light values in the Micro:Bit range from 0 to 255. `=NUMBERVALUE(RIGHT(E2,LEN(E2)-F2))` : Similar to the previous formula, only taking the right side number, which is the acceleration value. Values can be on the X,Y or Z axis, or combined, and explained here.

I also added a fixed “Row” column numbered 1-30, so that we will have an X axis for our charts.

The final piece of the puzzle – create two charts from the values. In both of them, the X axis is the row number, and the Y axis is the data coming in from the sensor (either light or acceleration). This is what it looks like:

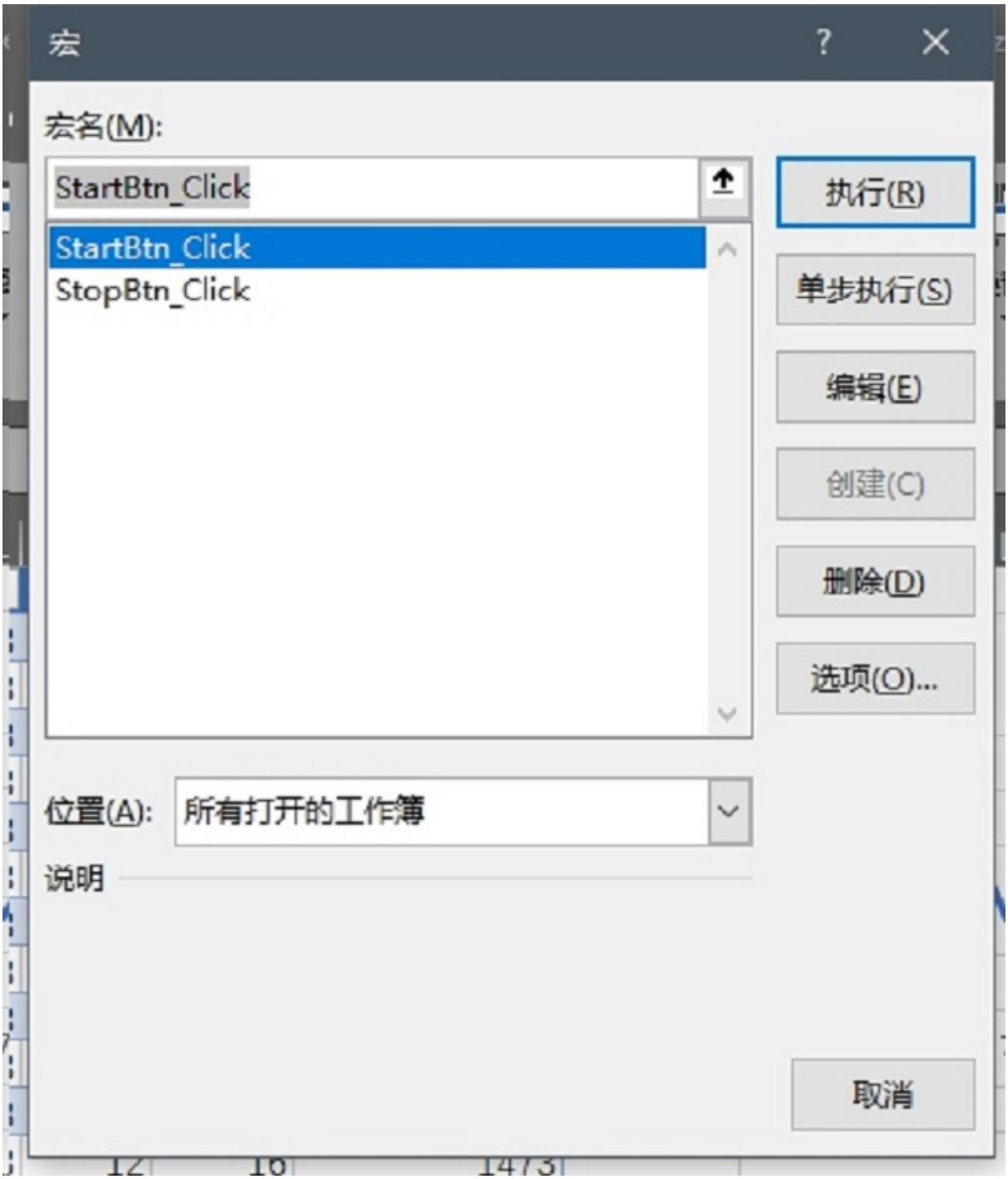



| Cursor | Raw Data | Sep | Row | Light | Acclrtor |
|---|---|---|---|---|---|
| | 66,965 | 3 | 1 | 66 | 965 |
| | 66,1274 | 3 | 2 | 66 | 1274 |
| | 69,1106 | 3 | 3 | 69 | 1106 |
| | 65,1460 | 3 | 4 | 65 | 1460 |
| | 54,1377 | 3 | 5 | 54 | 1377 |
| | 63,3263 | 3 | 6 | 63 | 3263 |
| | 63,3166 | 3 | 7 | 63 | 3166 |
| | 63,2507 | 3 | 8 | 63 | 2507 |
| | 20,1031 | 3 | 9 | 20 | 1031 |
| | 4,1162 | 2 | 10 | 4 | 1162 |
| | 8,1496 | 2 | 11 | 8 | 1496 |
| | 6,740 | 2 | 12 | 6 | 740 |
| | 7,732 | 2 | 13 | 7 | 732 |
| | 42,1266 | 3 | 14 | 42 | 1266 |
| --> | 47,1957 | 3 | 15 | 47 | 1957 |
| | 65,1079 | 3 | 16 | 65 | 1079 |
| | 67,1007 | 3 | 17 | 67 | 1007 |
| | 67,1023 | 3 | 18 | 67 | 1023 |
| | 68,1007 | 3 | 19 | 68 | 1007 |
| | 68,1113 | 3 | 20 | 68 | 1113 |
| | 68,1018 | 3 | 21 | 68 | 1018 |
| | 67,921 | 3 | 22 | 67 | 921 |
| | 67,1011 | 3 | 23 | 67 | 1011 |
| | 67,1008 | 3 | 24 | 67 | 1008 |
| | 67,1029 | 3 | 25 | 67 | 1029 |
| | 68,888 | 3 | 26 | 68 | 888 |
| | 69,1089 | 3 | 27 | 69 | 1089 |
| | 70,976 | 3 | 28 | 70 | 976 |
| | 70,1072 | 3 | 29 | 70 | 1072 |
| | 68,829 | 3 | 30 | 68 | 829 |

The fifth step: Revise VBA code

Click”view”to choose “macro”.

Choose”StartBtn-click” to”edit”.



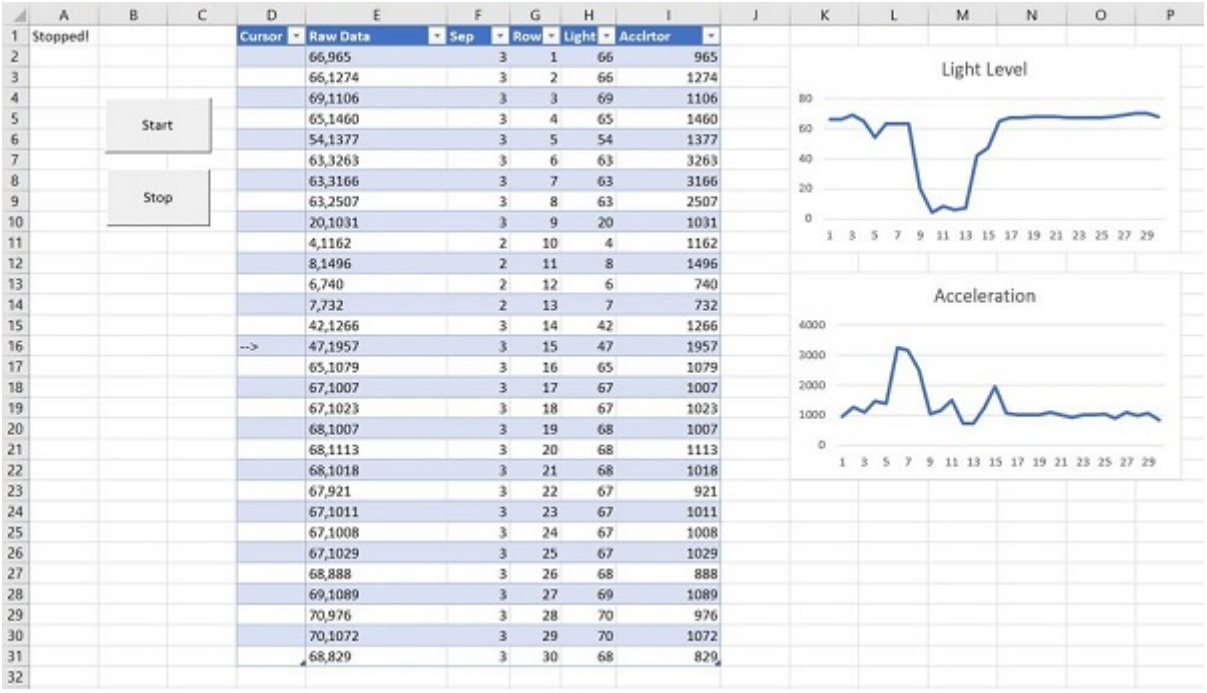

Revise the value of “COM” to the former port value, here the example is “2” in this code.

```
' StartBtn - Starts reading from the Micro:Bit into the grid

    Dim COM_Byte As Byte
    StopReading = False

    Cells(1, 1) = "Reading..."
    On Error GoTo ErrorHandler
    Open "COM2:115200,N,8,1" For Random As #1 Len = 1
    Cells(1, 1) = "Test..."
    PrevRow = 0
    Row = 0
    While (StopReading = False)

        EOL = False
        Data$ = ""
        While (Not EOL And Not StopReading)
            Get #1, , COM_Byte
            If COM_Byte = 13 Then
                EOL = True
            ElseIf (COM_Byte <> 10) And (COM_Byte <> 0) Then
                Data$ = Data$ & Chr(COM_Byte)
            End If
        Wend
        If Not StopReading And Left(Data$, 2) = "D:" Then
            Row = (Row + 1) Mod 30
            Cells(PrevRow + 2, 4) = "" ' Clear previous location indicator
            Cells(Row + 2, 4) = "—>" ' Display new location indicator
            Cells(Row + 2, 5) = "" " + Right(Data$, Len(Data$) - 2) ' Enter data into the grid as a string
            PrevRow = Row
            DoEvents
            DoEvents    ' Hack - Apparently, two DoEvents are required to get the charts to update live
```

Clict to save the Excel in the right upper corner and reopen it.

**The end result - incoming data is visualized live!**

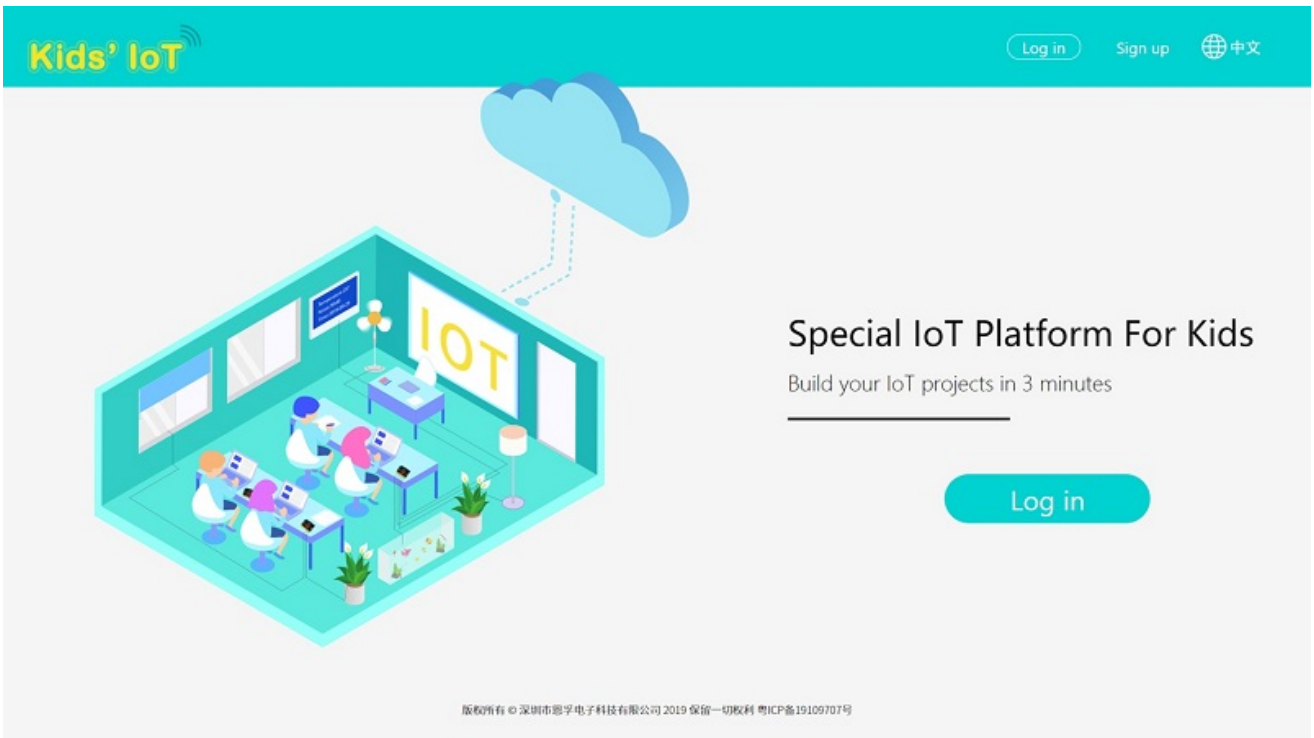
All you have to do now is click “Start” and see the data coming in and being charted live!




| Cursor | Raw Data | Sep | Row | Light | Acclrtor |
|---|---|---|---|---|---|
| | 66,965 | 3 | 1 | 66 | 965 |
| | 66,1274 | 3 | 2 | 66 | 1274 |
| | 69,1106 | 3 | 3 | 69 | 1106 |
| | 65,1460 | 3 | 4 | 65 | 1460 |
| | 54,1377 | 3 | 5 | 54 | 1377 |
| | 63,3263 | 3 | 6 | 63 | 3263 |
| | 63,3166 | 3 | 7 | 63 | 3166 |
| | 63,2507 | 3 | 8 | 63 | 2507 |
| | 20,1031 | 3 | 9 | 20 | 1031 |
| | 4,1162 | 2 | 10 | 4 | 1162 |
| | 8,1496 | 2 | 11 | 8 | 1496 |
| | 6,740 | 2 | 12 | 6 | 740 |
| | 7,732 | 2 | 13 | 7 | 732 |
| | 42,1266 | 3 | 14 | 42 | 1266 |
| --> | 47,1957 | 3 | 15 | 47 | 1957 |
| | 65,1079 | 3 | 16 | 65 | 1079 |
| | 67,1007 | 3 | 17 | 67 | 1007 |
| | 67,1023 | 3 | 18 | 67 | 1023 |
| | 68,1007 | 3 | 19 | 68 | 1007 |
| | 68,1113 | 3 | 20 | 68 | 1113 |
| | 68,1018 | 3 | 21 | 68 | 1018 |
| | 67,921 | 3 | 22 | 67 | 921 |
| | 67,1011 | 3 | 23 | 67 | 1011 |
| | 67,1008 | 3 | 24 | 67 | 1008 |
| | 67,1029 | 3 | 25 | 67 | 1029 |
| | 68,888 | 3 | 26 | 68 | 888 |
| | 69,1089 | 3 | 27 | 69 | 1089 |
| | 70,976 | 3 | 28 | 70 | 976 |
| | 70,1072 | 3 | 29 | 70 | 1072 |
| | 68,829 | 3 | 30 | 68 | 829 |

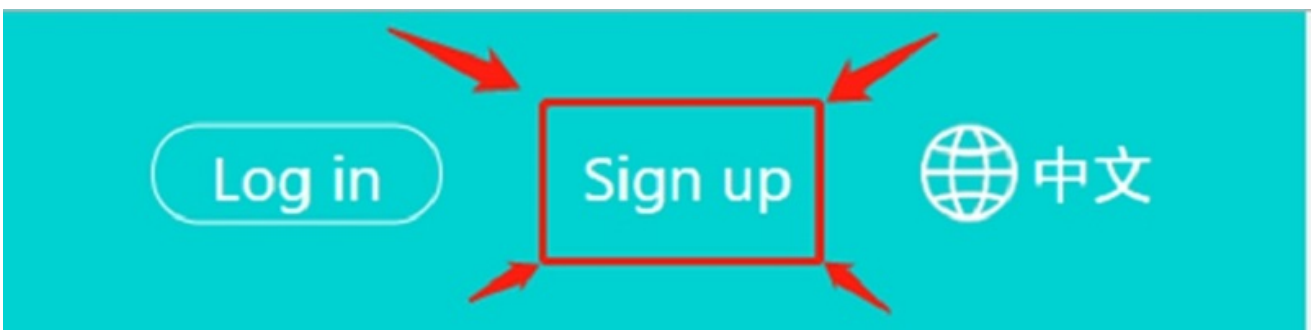
# 6. The fourth part: Guidance of KidsIoT

- ) KidsIoT is a platform for IoT(Internet of Things) produced by Elecfreaks with only three minutues to get connected, it can achieve a remote control to the micro:bit. (Currently in English version only) link: KidsIoT Cloud Platform: https://www.kidsiot.cn/

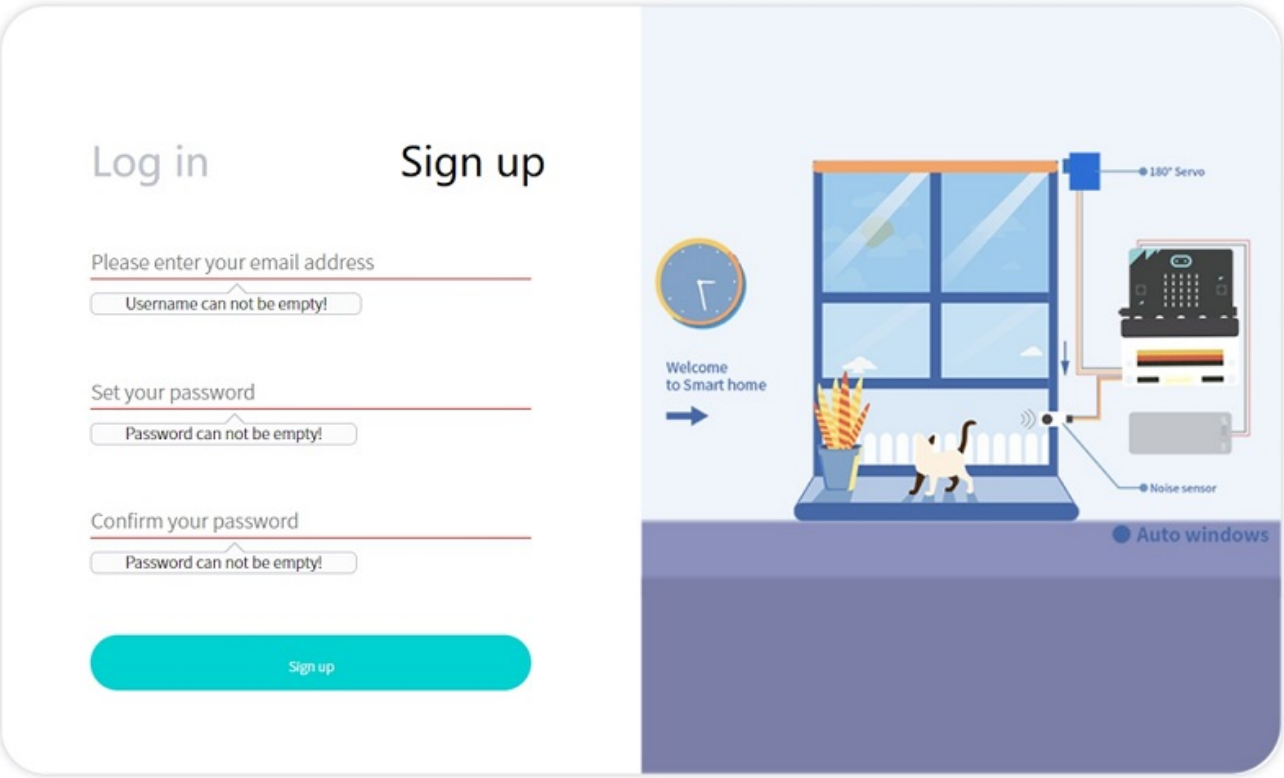



## 6.1. Registration

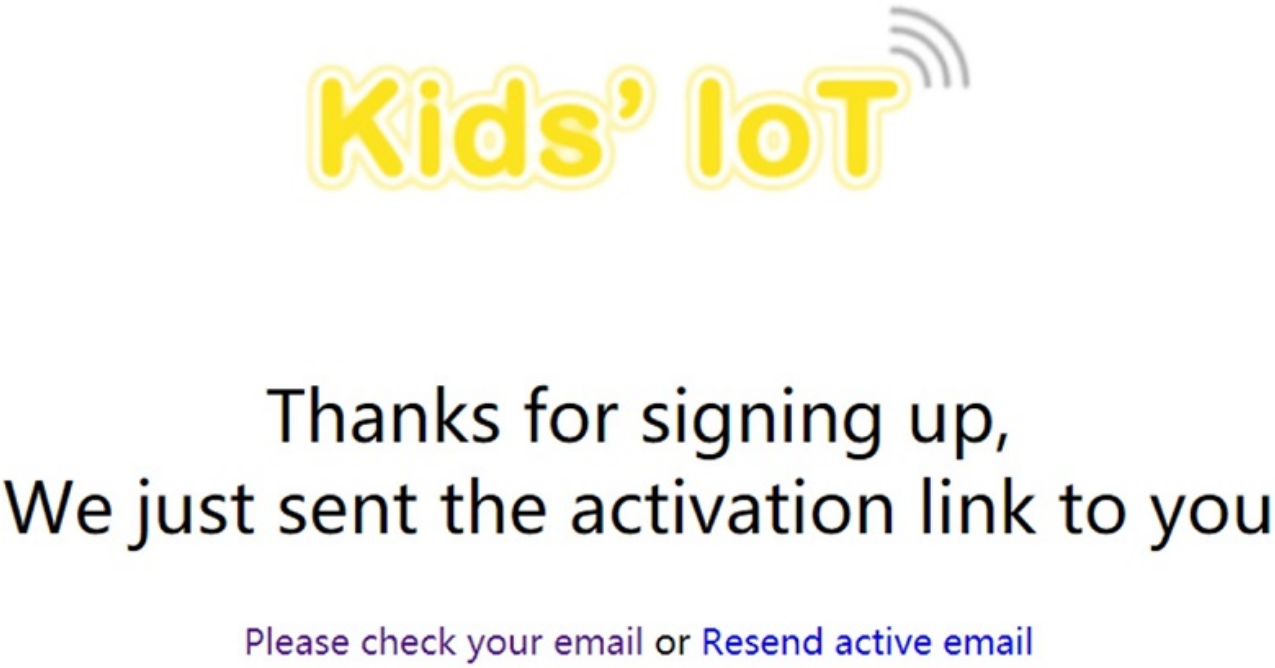
- Sign up in the website to get a new account.




- Fill in with your email address and password, and click Sign Up.

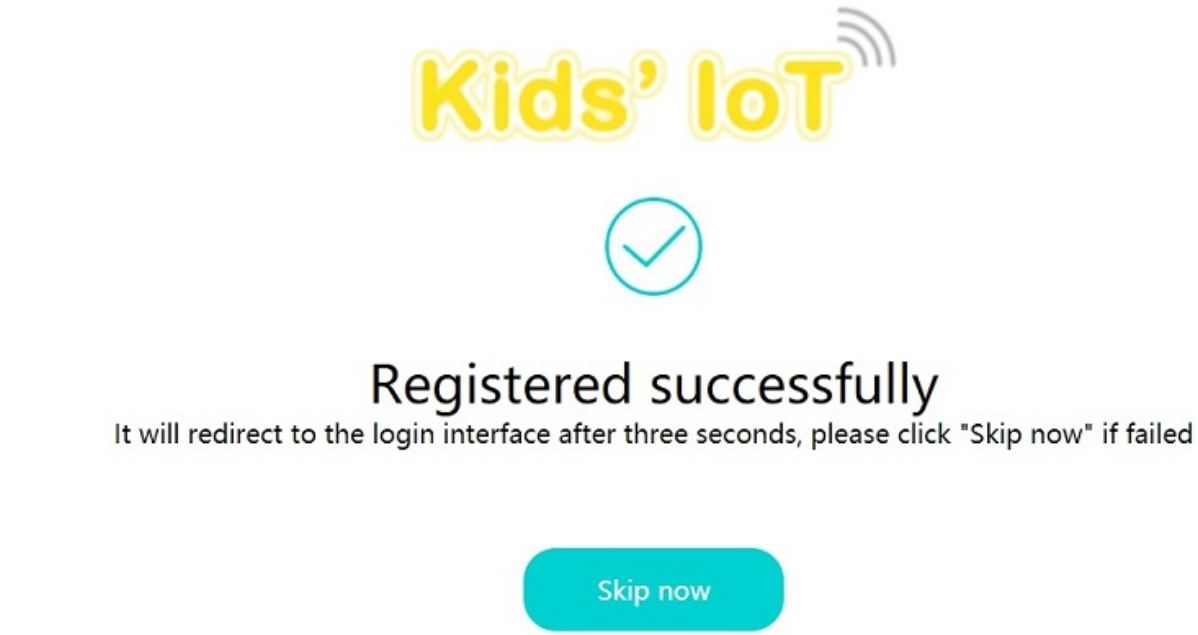



- After signing up, an email should be sent to your mailbox.



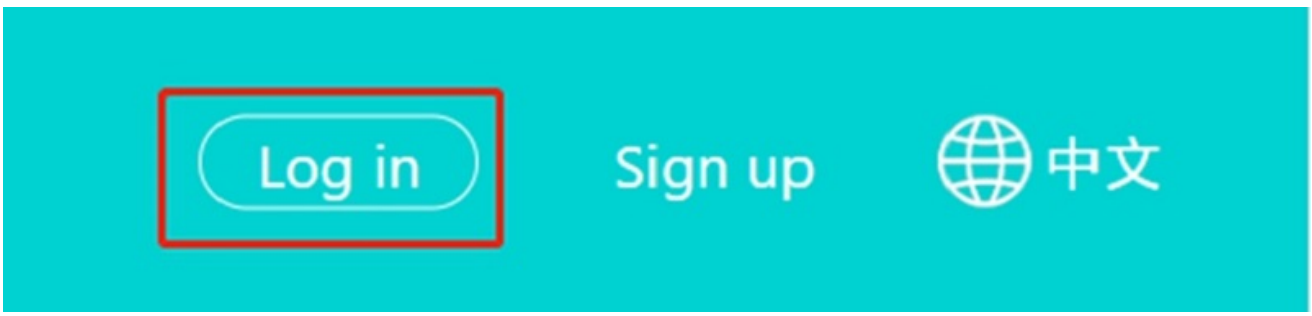

- Activate your account with the links sent to your mailbox, note the letter might be in your junk email for the first time.

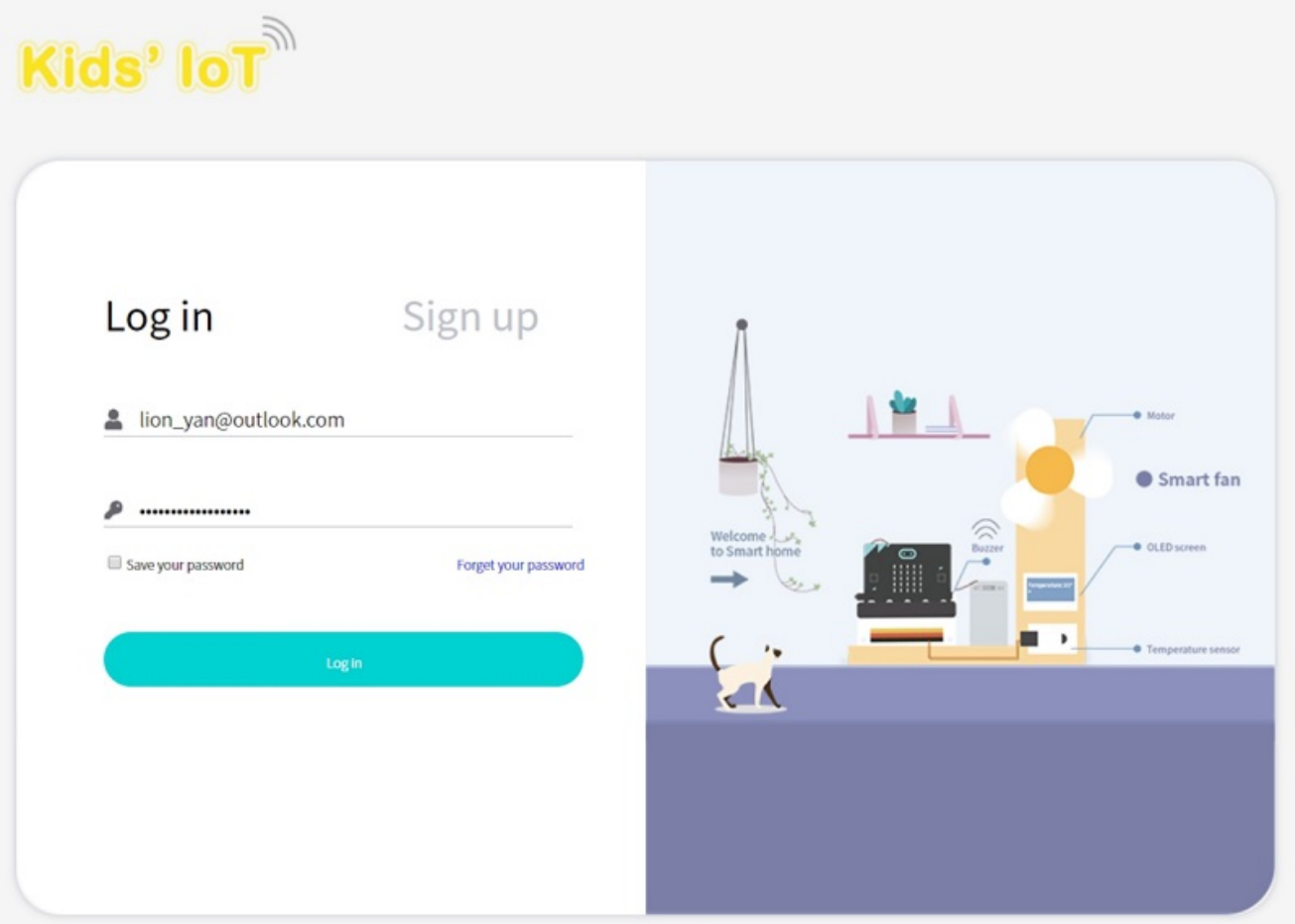

- Sign Up successfully!




## 6.2. Guidance of the KidsIoT

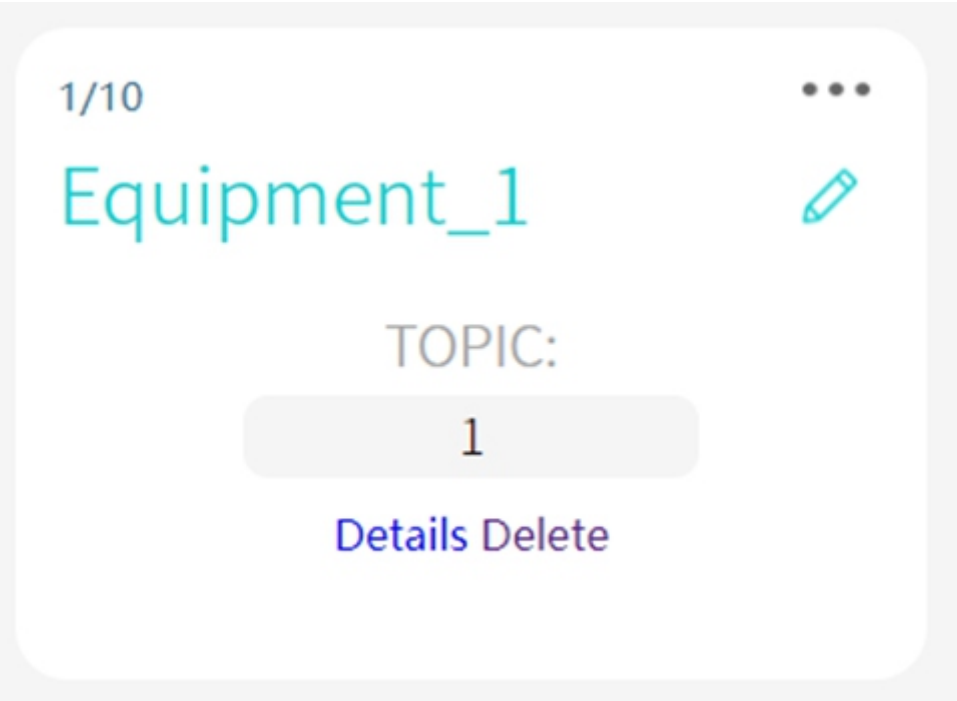
- Go to the log in interface by clicking Sign in.



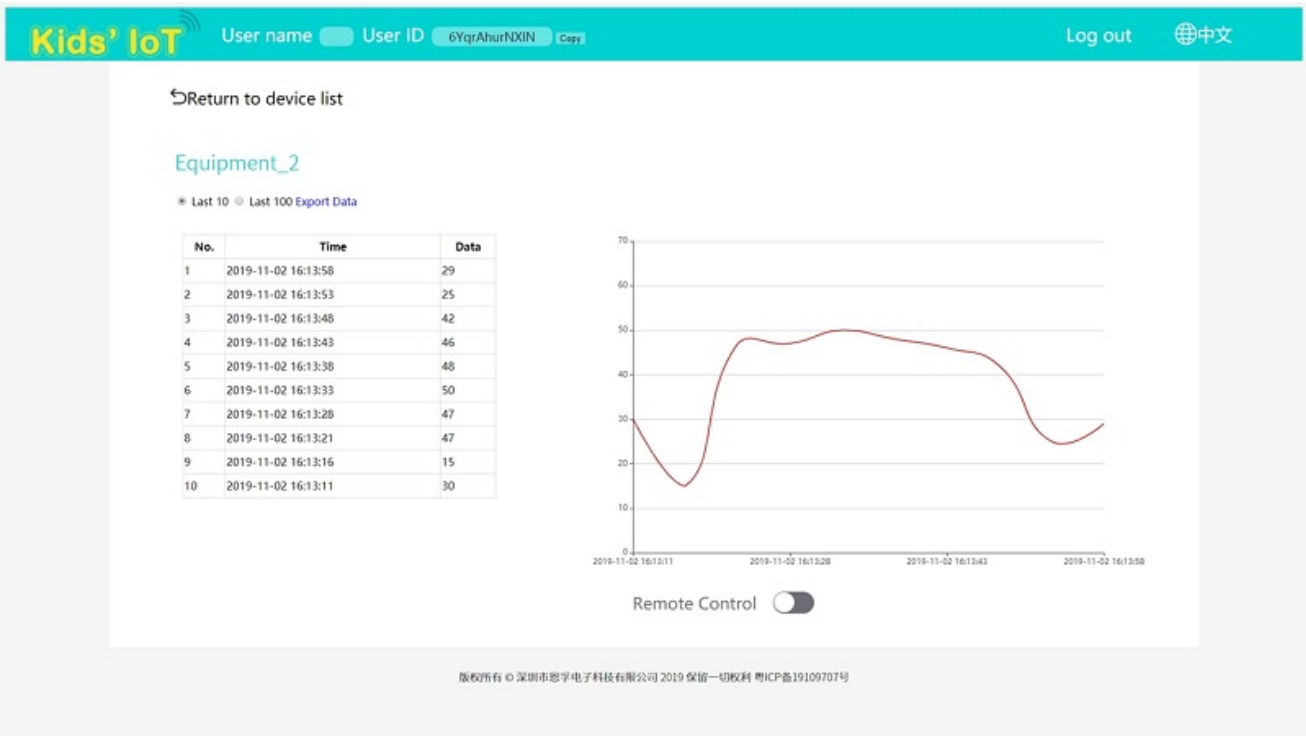



- Click log in to enter the device manage interface, the “User Name” on the top left corner is your email address, the unique “User Token”(Currently miswritting as ID, we will correct to Token later) on the top right corner is the only indentification code for this platform which is corresponding to your account.


- Create new device, “Topic” is the only identification code(the only device in the account), and you can revise the device name(only 10 devices can be created).
- Click “Details”to see the information, click “Delete” to delete the device.




- The upload data will be shown on the left, what on the right is a two-dimension line chart for data and time. You can choose the data that you want to see or export the data.
- You can use “Remote Control” to give instruct to the micro:bit.



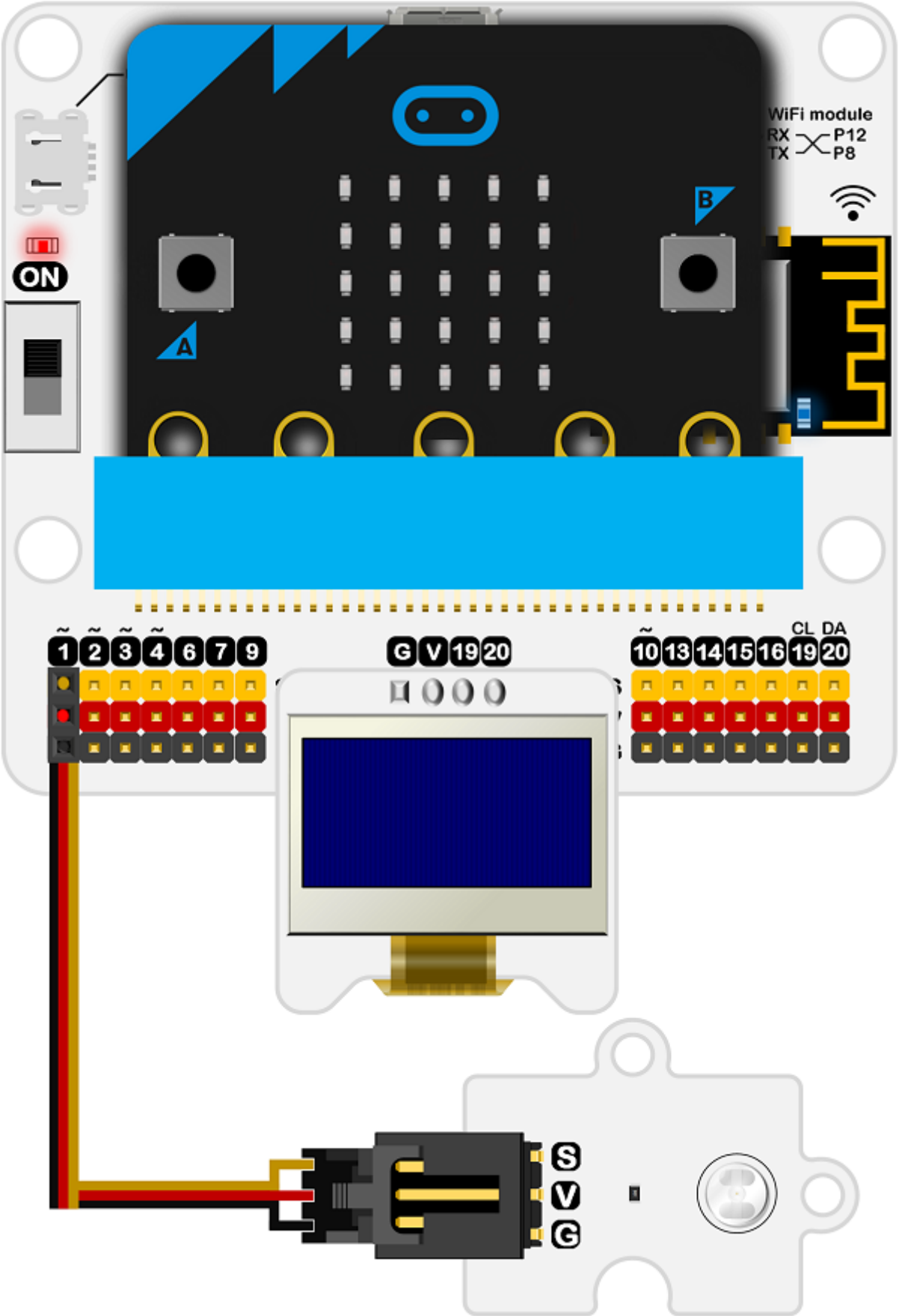

## 6.3. Write Code

MicroSoftmakecode

## 6.4. Programme

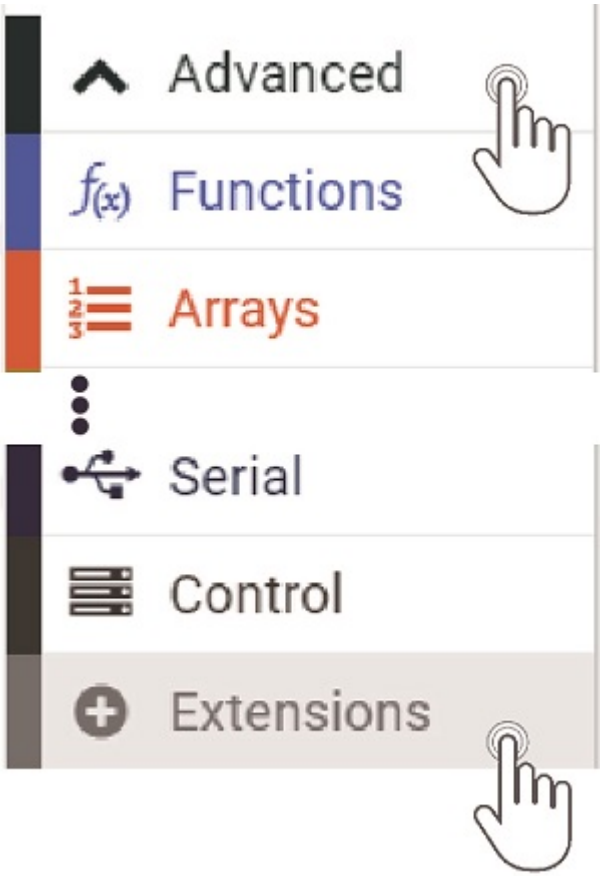
## Modules Connection Diagram

- Connect the Light sensor to P1.

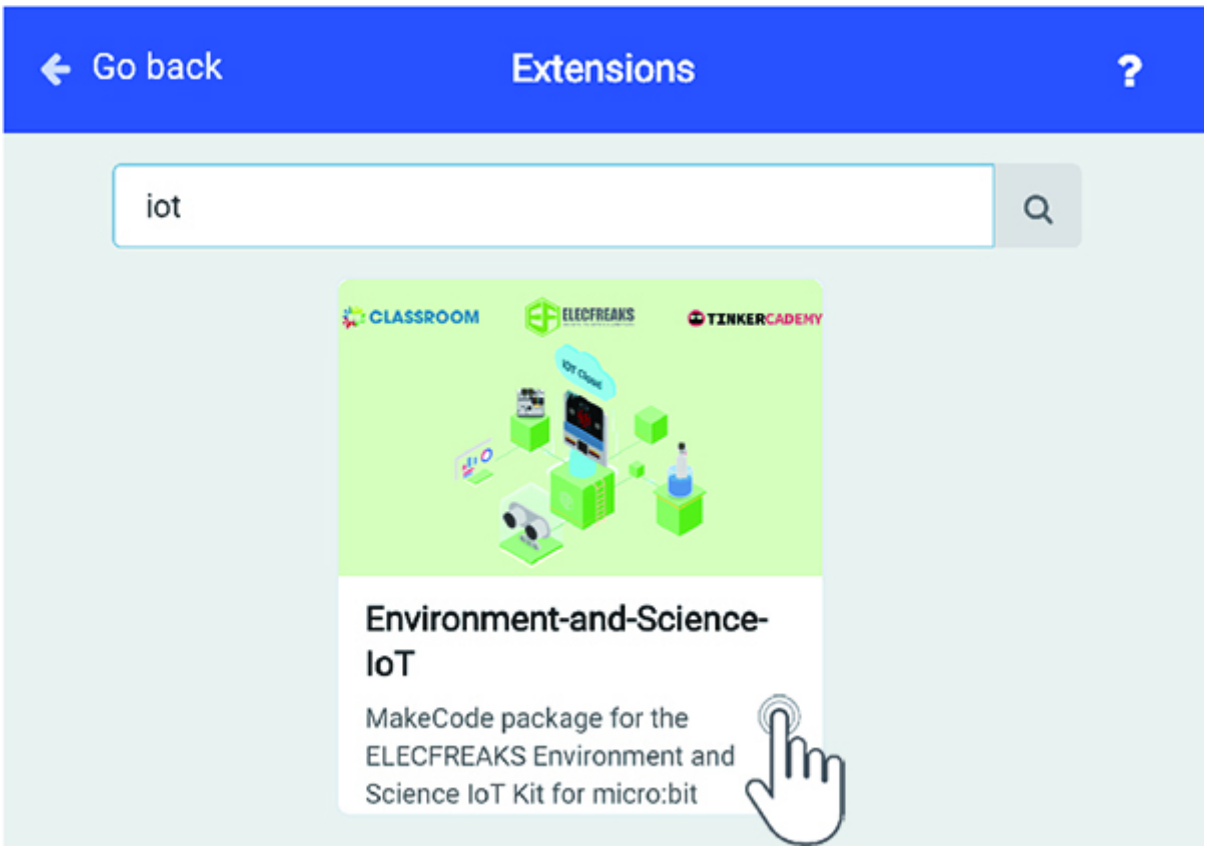


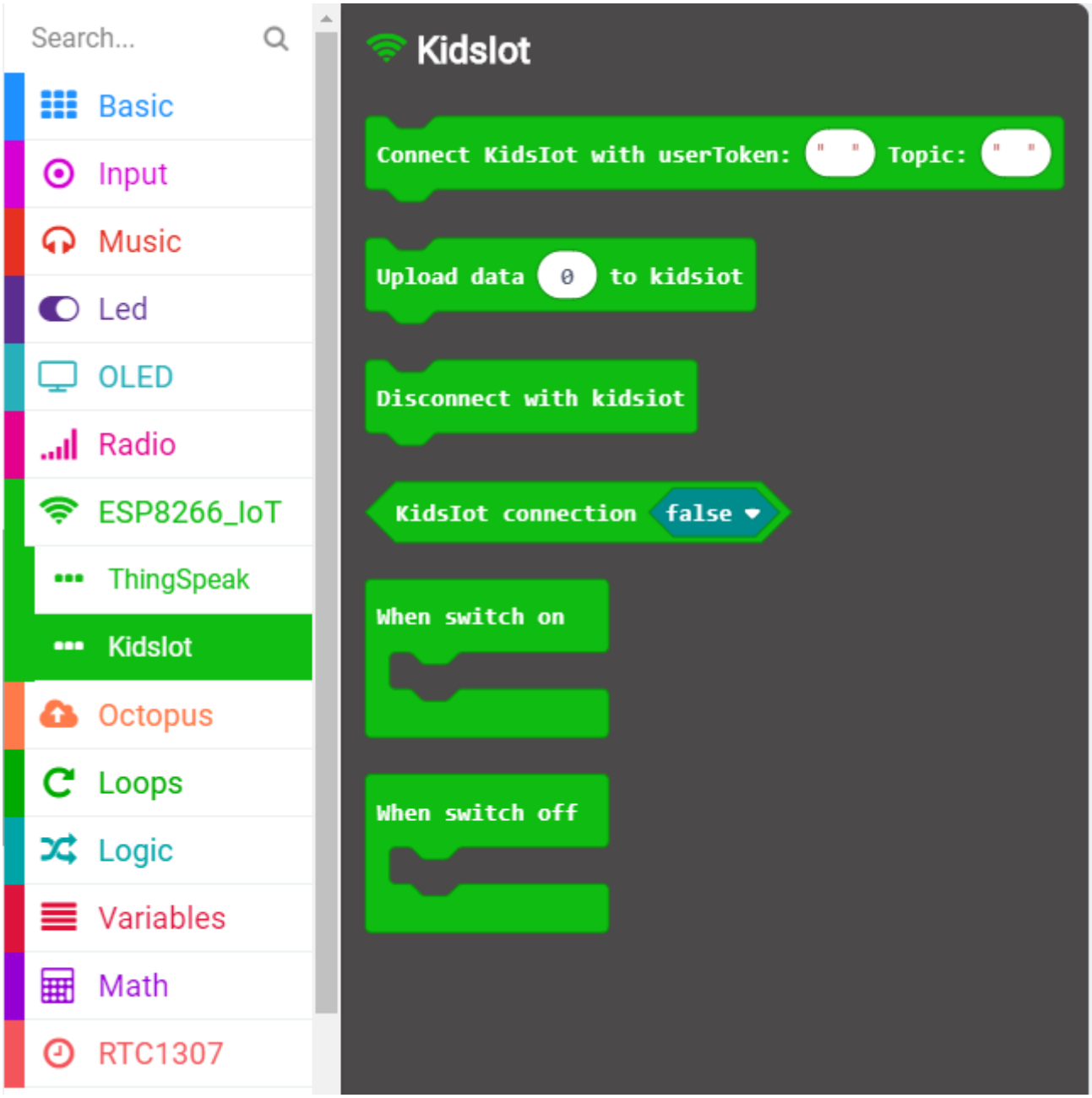

## Add Package

- Click “Advanced” to see more choices in MakeCode drawer.



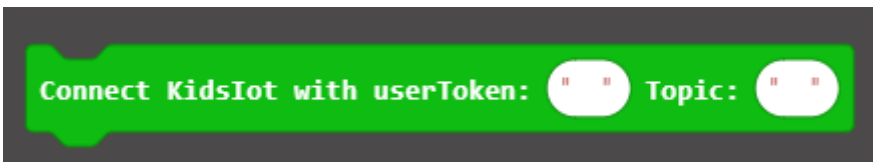

- We need to add a package for programming the IoT kit. Click “Extension” in the drawer and search “IoT” in the dialogue box to download it.



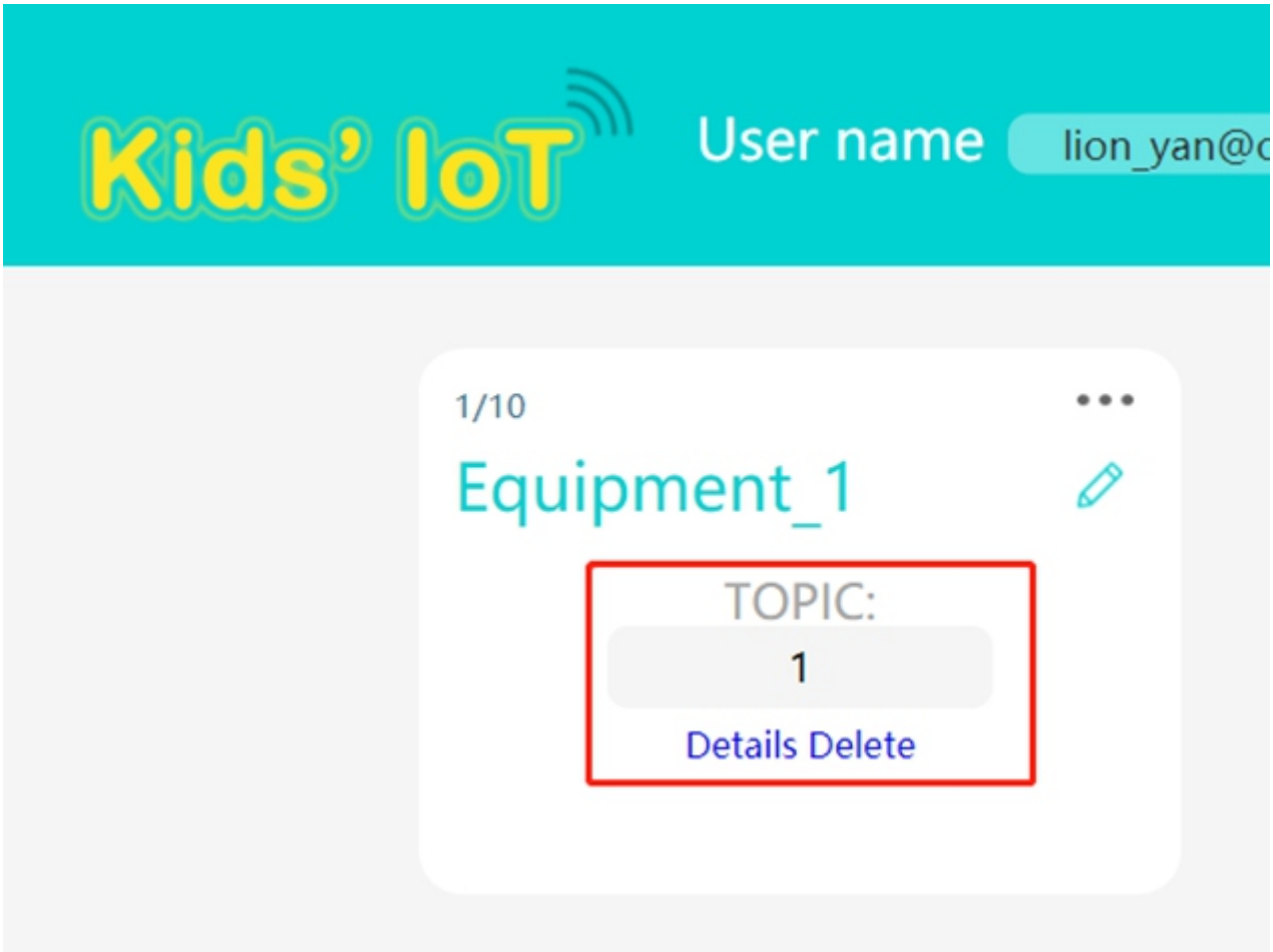



***Note：If you get a warning indicating some packages will be removed because of incompatibility issues, you can follow the prompts or create a new project in the menu.

## Code Details




Connect to Kids'IoT platform, User Token(Currently miswritting as ID) is the only indentification code for the account which can not be revised.




Topic is the only indentification code and you have to appoint the updload device when connecting. The device number(Topic) will be in sequential order.

---

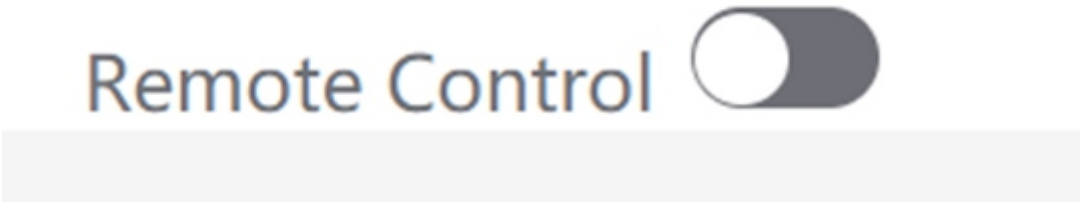

Upload an integer to KidsIoT and it shows here.

| No. | Time | Data |
|---|---|---|
| 1 | 2019-11-02 16:13:58 | 29 |
| 2 | 2019-11-02 16:13:53 | 25 |
| 3 | 2019-11-02 16:13:48 | 42 |
| 4 | 2019-11-02 16:13:43 | 46 |
| 5 | 2019-11-02 16:13:38 | 48 |
| 6 | 2019-11-02 16:13:33 | 50 |
| 7 | 2019-11-02 16:13:28 | 47 |
| 8 | 2019-11-02 16:13:21 | 47 |
| 9 | 2019-11-02 16:13:16 | 15 |
| 10 | 2019-11-02 16:13:11 | 30 |


---

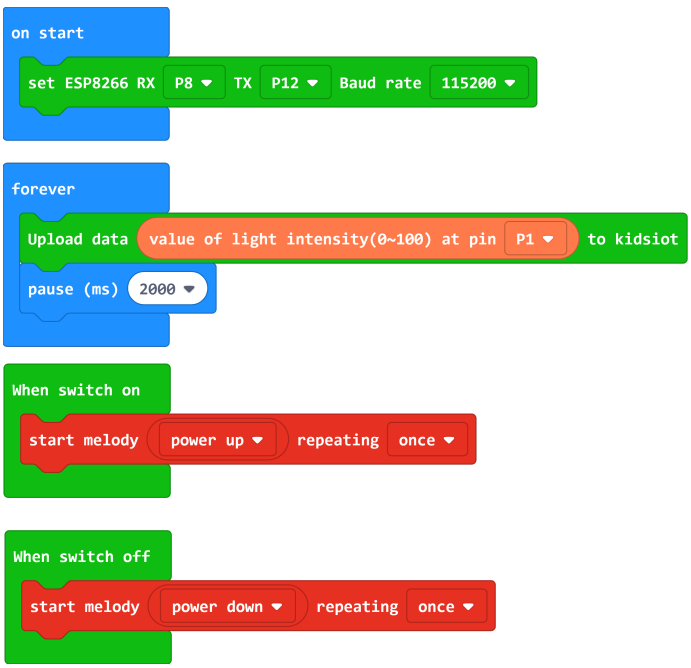

Judge if the conection status is successful, “Ture” for success, “False” for failure. You can edit the reconnection system to ensure the stable connection.

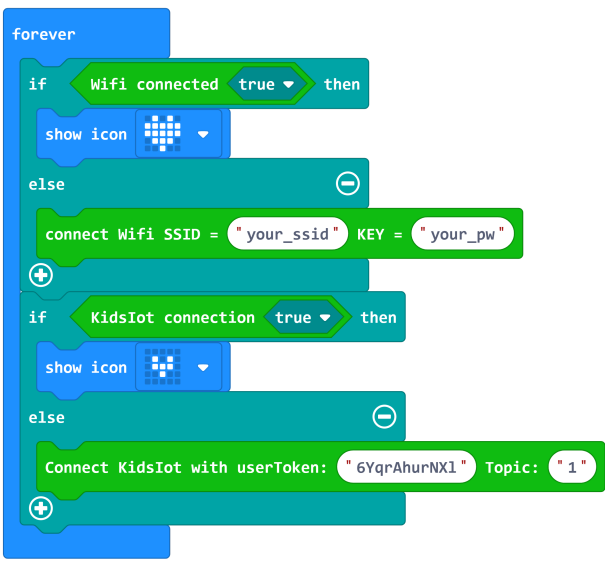
---


When getting connected well, you can click this switch to operate these two bricks.

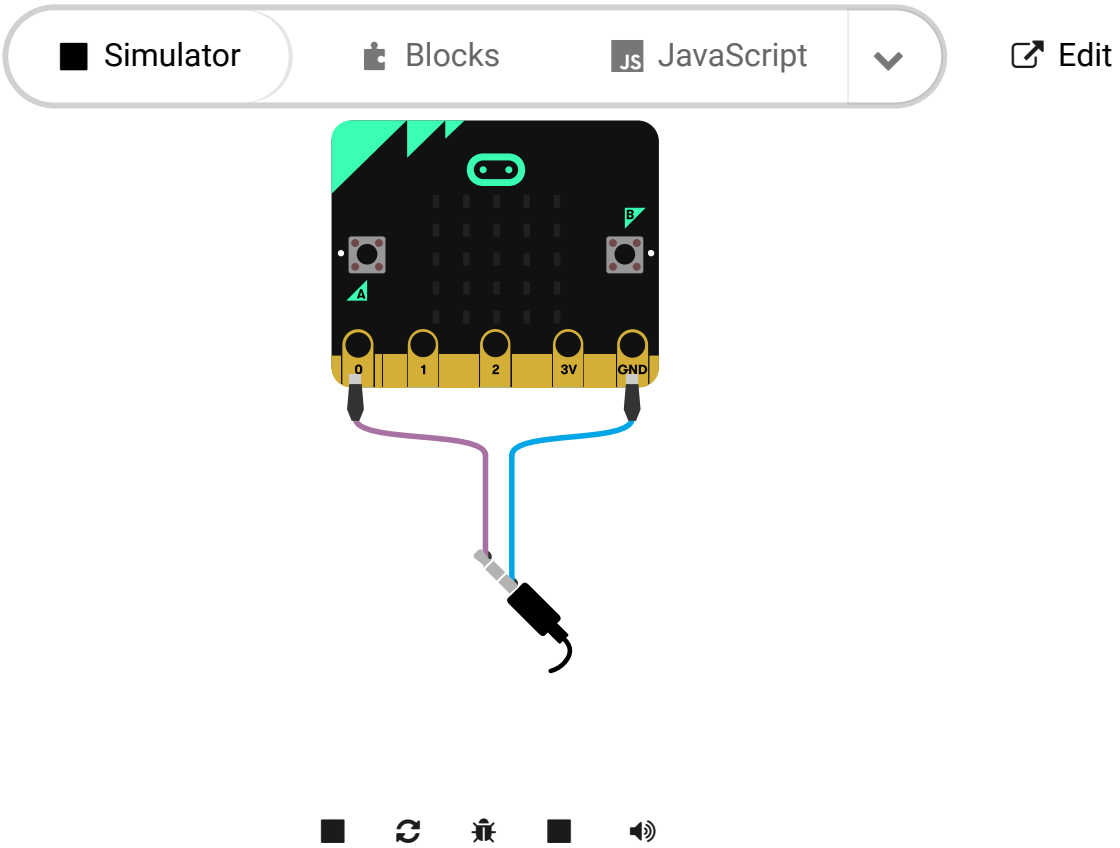
## Reference




Link： https://makecode.microbit.org/_d1fifJTomKq2

You can also revise the code with the below page:




## Result

Connect WIFI when on start. Continuously judge if the WIFI is connected successfully, if yes, a big icon will be shown or it continues connecting. Continuously judge if the connection to KidsIoT is ready, if yes, a small icon will be shown or it continues connecting. Continuously judge if the connection to KidsIoT is a success, if yes, the data(given by the light sensor connected to P1) will be uploaded. If the switch of the platform is on, it plays a music of “Power Up” If the switch of the platform is off, it plays a music of “Power Down”

# 7. iot:kit case01:Air quality monitoring

## 7.1. Our goal

- Let's make an air quality monitoring.

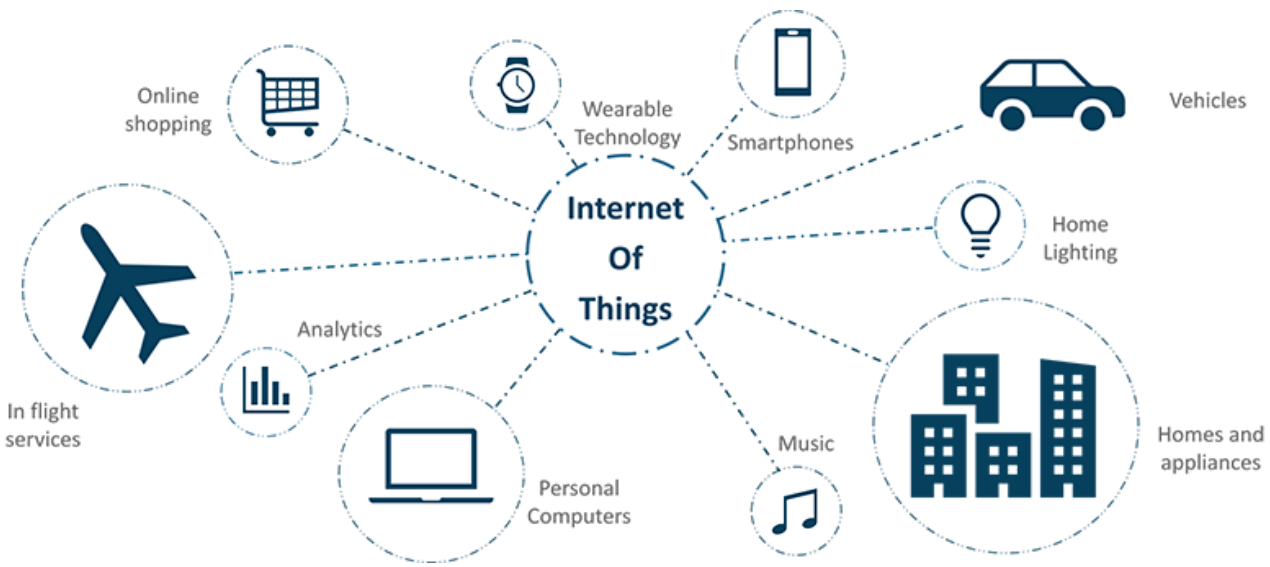
## 7.2. Required materials

- 1 x IOT:kit https://www.elecfreaks.com/store](https://www.elecfreaks.com/store/micro-bit-smart-science-iot-kit-with-micro-bit.html)

## 7.3. Background

### What is the IOT

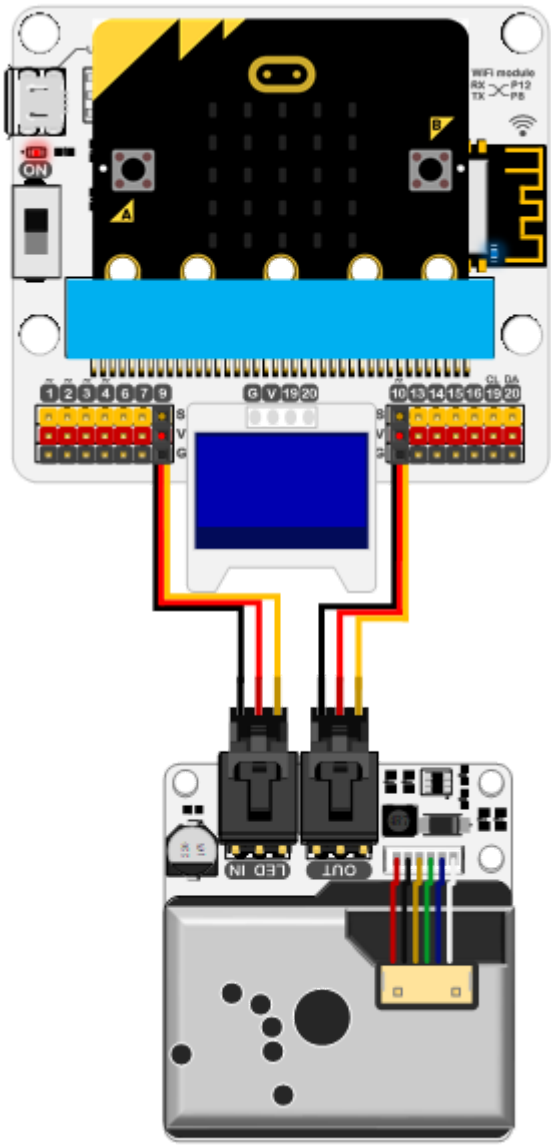
- IoTThe Internet of things (IoT) is the network of devices such as vehicles, and home appliances that contain electronics, software, sensors, actuators, and connectivity which allows these things to connect, interact and exchange data.The IoT involves extending Internet connectivity beyond standard devices, such as desktops, laptops, smartphones and tablets, to any range of traditionally dumb or non-internet-enabled physical devices and everyday objects. Embedded with technology, these devices can communicate and interact over the Internet, and they can be remotely monitored and controlled.



## 7.4. Hardware connection

As below picture, connect `LED IN` of the dust sensor to `P9` and `OUT` to `P10`.

Connect the OLED screen to the `IIC` interface.



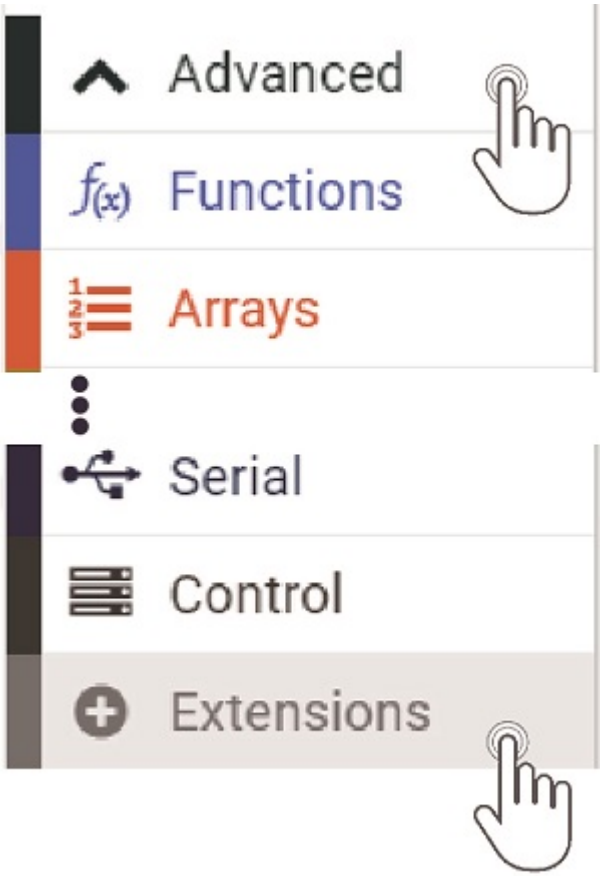

## 7.5. Software

makecode

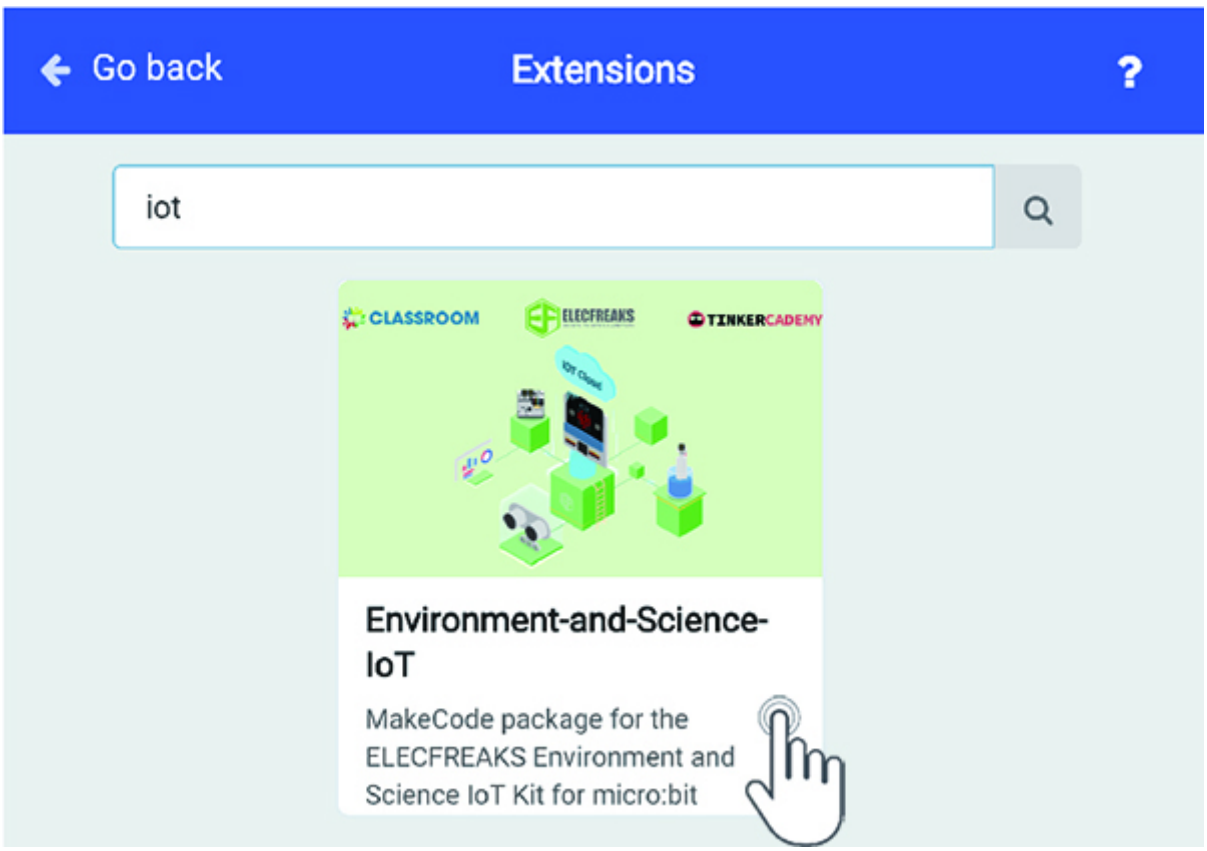
## 7.6. Coding

### Step 1

- Click on “Advanced” in the MakeCode Drawer to see more code sections.




- We need to add an extension for coding to the IOT. Click on the “Extension” at bottom of coding drawer, then Search for “IOT” and click on the IOT package to add it to your project. (As below picture)




Note：If you get a warning telling you some packages will be removed because of incompatibility issues, either follow the prompts or create a new project in the Project file menu.

## Step 2

Snap the `initialize OLED` block into the `on start` and write parameters `64*128`.

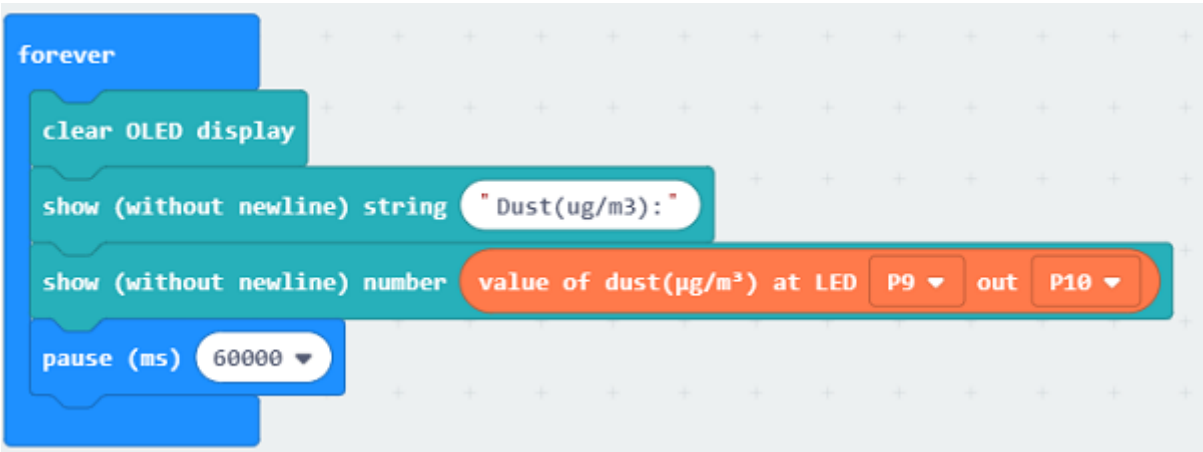
Initialize the OLED screen to `64` * `128` pixel.


## Step 2

Snap the `clear OLED display`, `show string` and `show number` blocks into the forever as below picture.

Showing string `Dust(ug/m3):` to display returned value of the dust sensor.

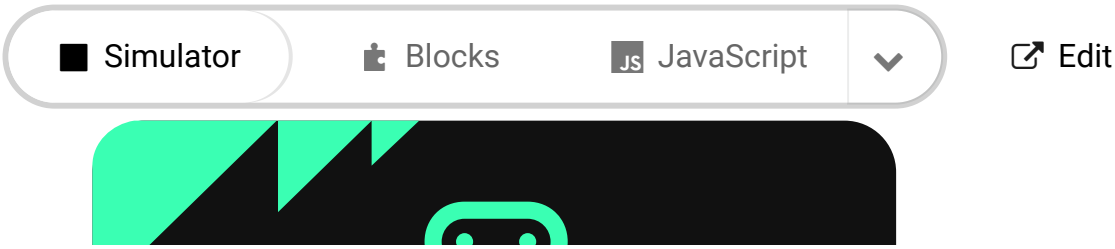
Pause `60s`, once a minute.

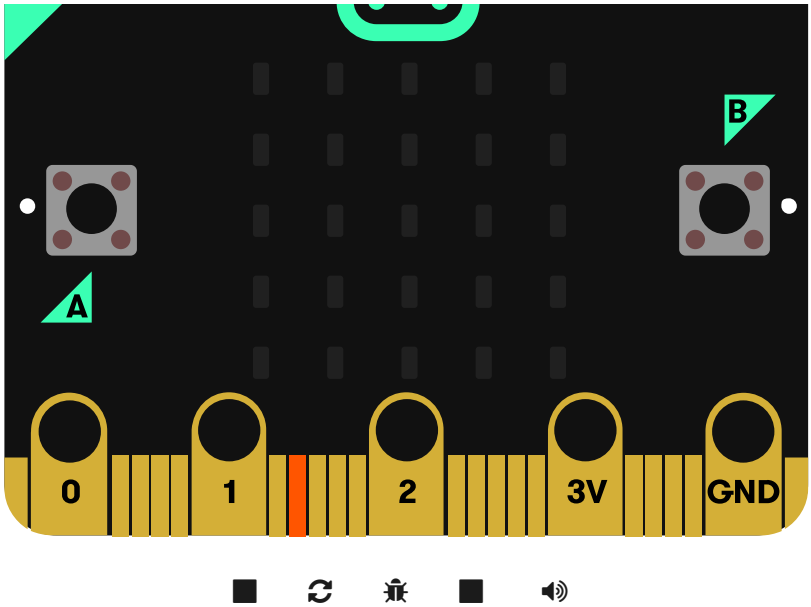



## Program

Program link: https://makecode.microbit.org/_YFCE0bc7vbVC

If you don't want to type these code by yourself, you can directly download the whole program from the link below:






### Result

The dust particle number will be displayed every minute.

## 7.7. Think

How do you do to make it alarm when the air is in bad quality?

## 7.8. Questions

## 7.9. More Information

# 8. iot:kit case02: Environmental noise detection

## 8.1. Our goal

- Let's make an environmental noise detection.

## 8.2. Required materials

- 1 x IOT:kit https://www.elecfreaks.com/store](https://www.elecfreaks.com/store/micro-bit-smart-science-iot-kit-with-micro-bit.html)

## 8.3. Background

### What is the environmental noise?

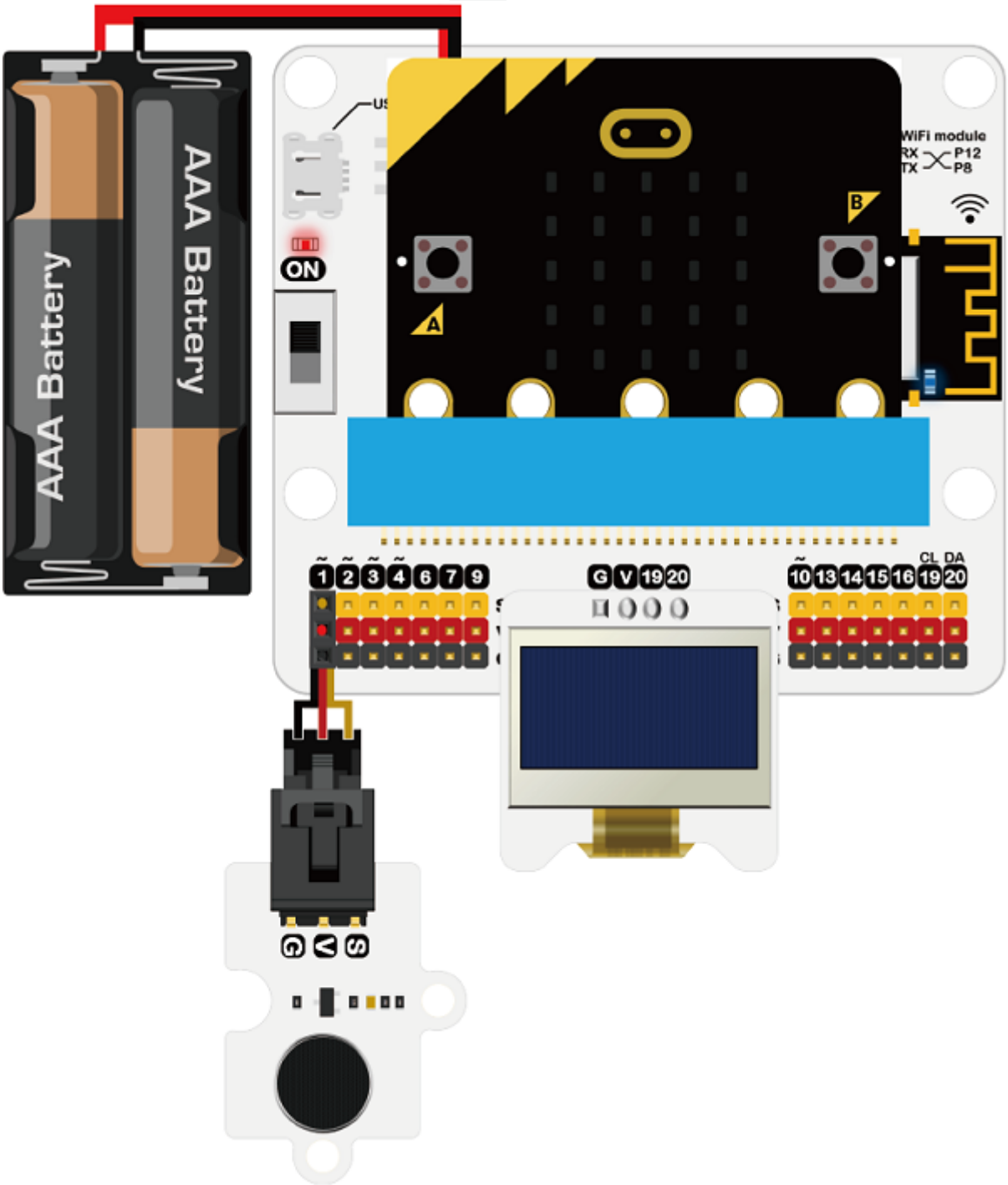
- Environmental noise is the summary of noise pollution from outside, caused by transport, industrial and recreational activities.Noise is frequently described as 'unwanted sound', and, within this context, environmental noise is generally present in some form in all areas of human, animal, or environmental activity. The effects in humans of exposure to environmental noise may vary from emotional to physiological and psychological.

## 8.4. Hardware connection

As below picture, let the noise sensor be connected to `P1`.

Let the OLED screen be connected to `IIC`.

Let the onboard RTC be connected to `IIC` Bus.



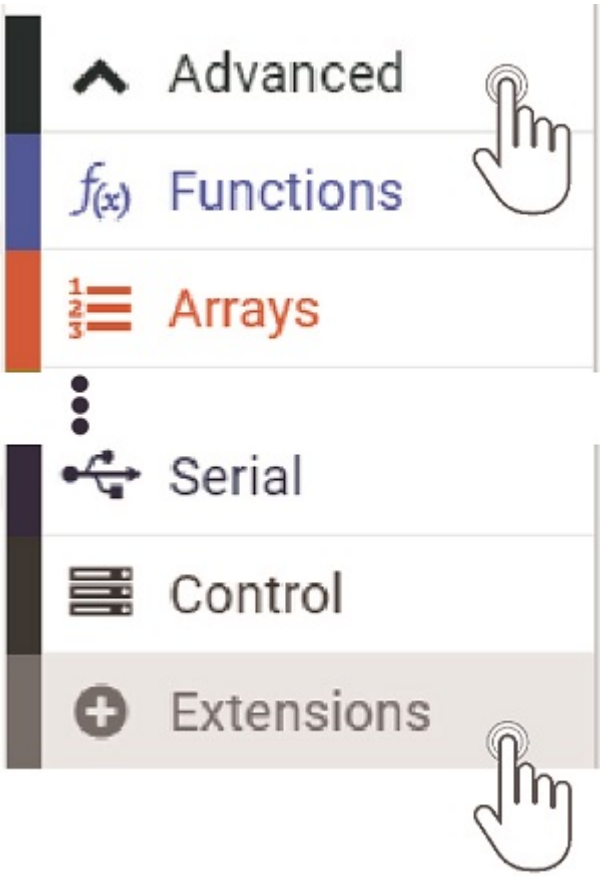

## 8.5. Software

makecode

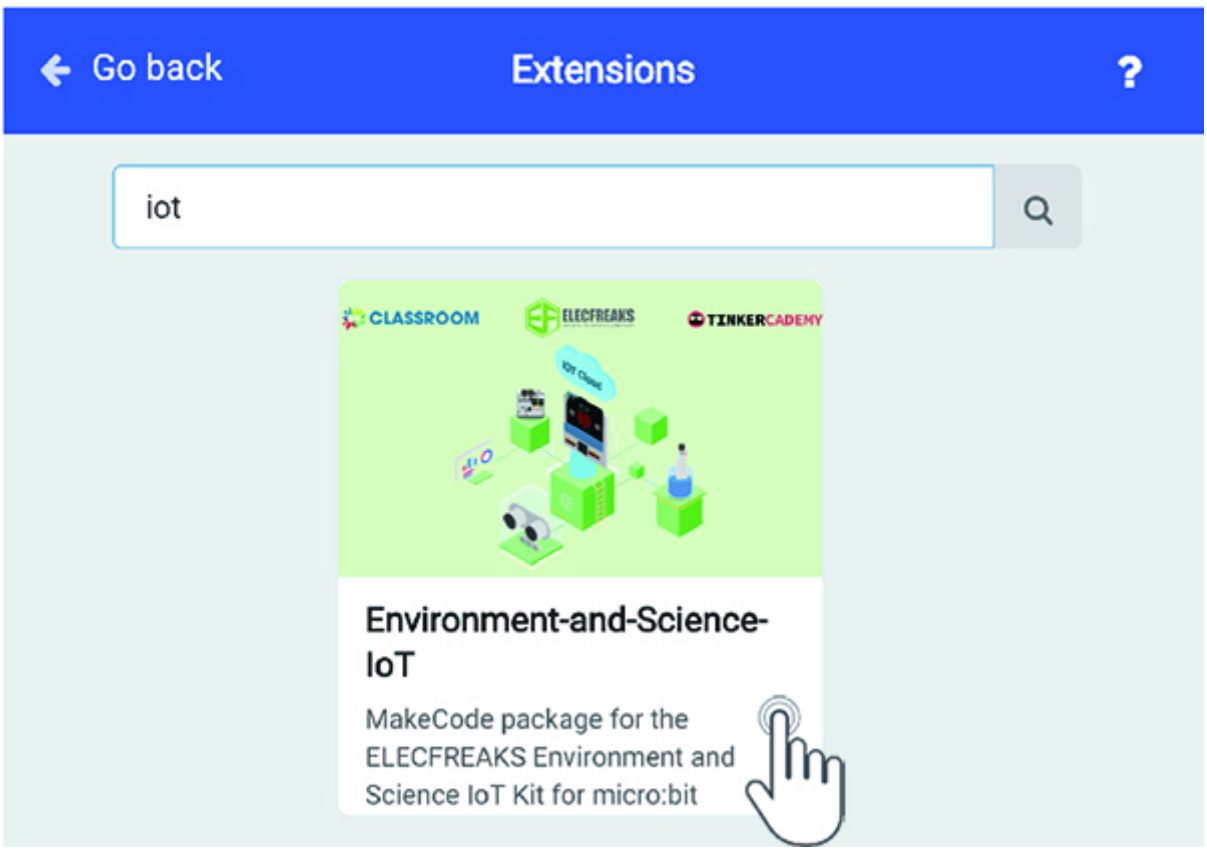
## 8.6. Coding

### Step 1

- Click on “Advanced” in the MakeCode Drawer to see more code sections.




- We need to add an extension for coding to the IOT. Click on the “Extension” at bottom of coding drawer, then Search for “IOT” and click on the IOT package to add it to your project. (As below picture)




Note：If you get a warning telling you some packages will be removed because of incompatibility issues, either follow the prompts or create a new project in the Project file menu.

## Step 2

Snap the `initialize OLED` in to the `on start`, initialize OLED screen pixels to `64*128`.

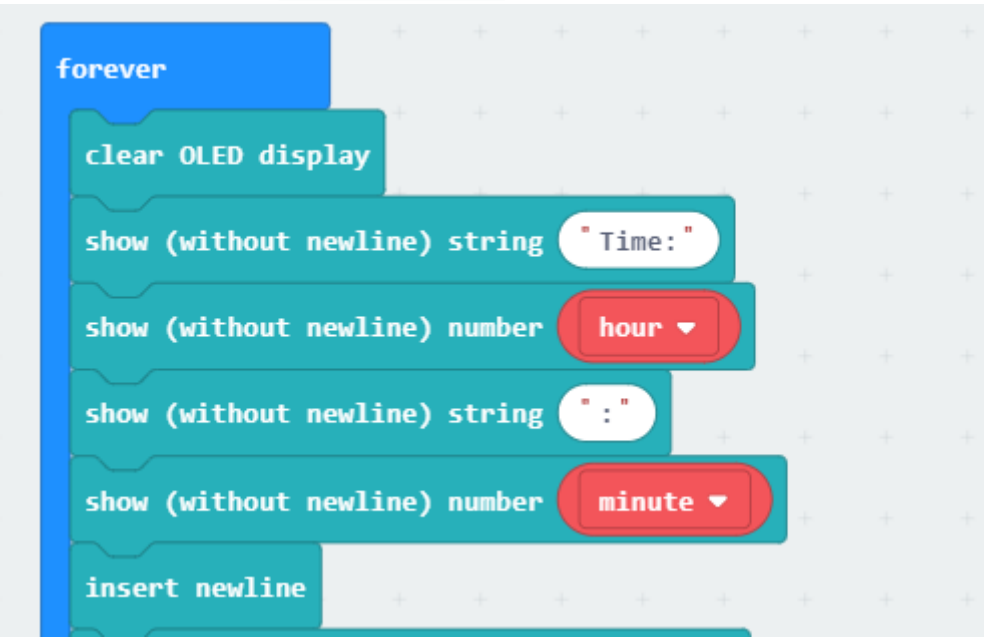
Then snap the RTC timing block after the `initialize OLED`, set current time to 14:15.


## Step 2

Snap the `clear OLED display`, `show string` and `show number` blocks into the `forever` in turn.

Display string: `Time:` for current hours and minutes.

Then, snap into the `insert newline`.

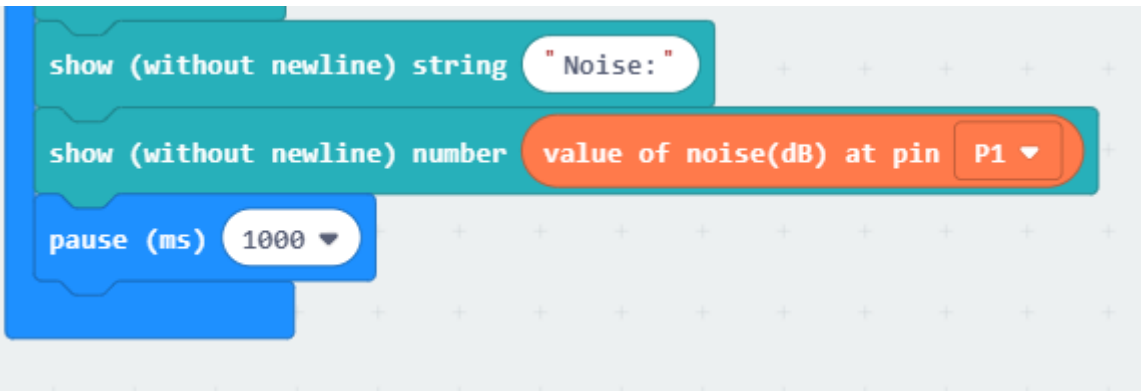



## Step 3

Now snap intp the `show string` and `show number` blocks.

Display string `Noise:` and returned noise value.

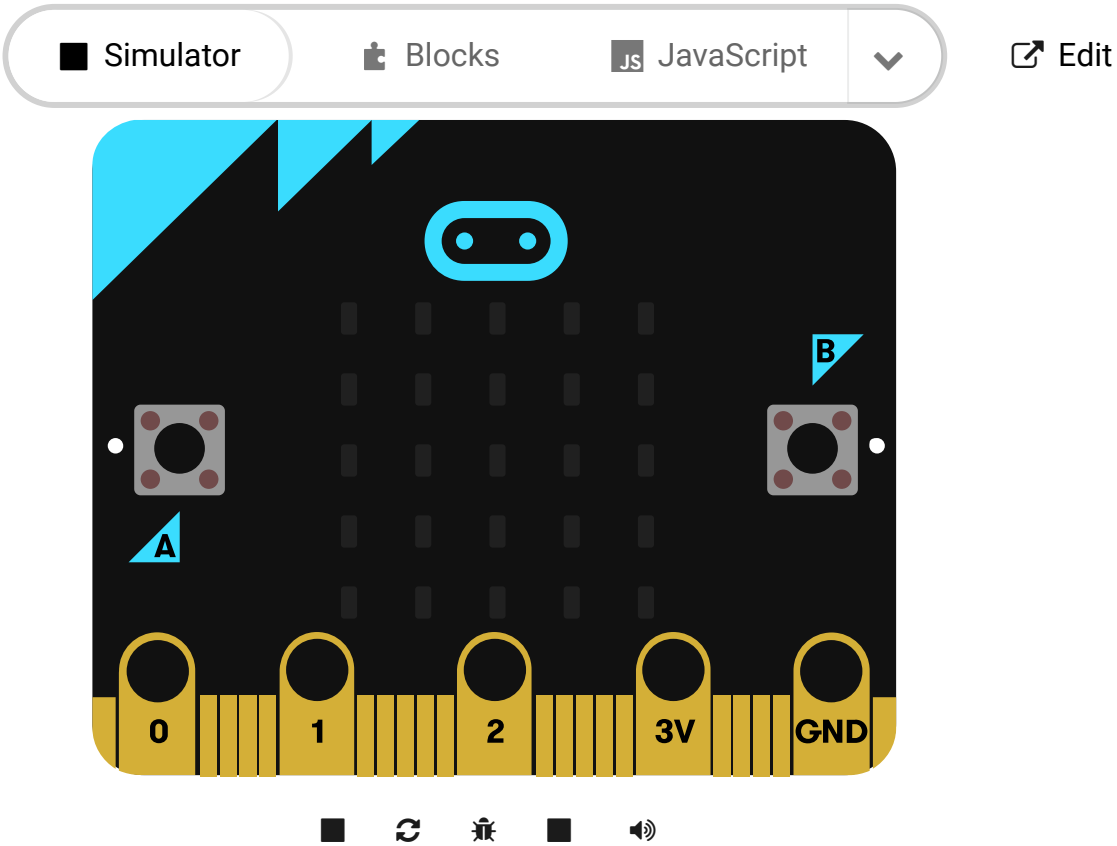
Pause 1 second.




## Program

Program link：https://makecode.microbit.org/_2jvctXPa0heW

If you don't want to type these code by yourself, you can directly download the whole program from the link below:




## Result

The noise db be displayed every second.

# 8.7. Think

How can you count the average noise db in a minute?

## 8.8. Questions

## 8.9. More Information

# 9. iot:kit case03: An environmental quality monitoring station

## 9.1. Our goal

- Let's make an environmental quality monitoring station.

## 9.2. Required materials

- 1 x IOT:kit IOT:kit:https://www.elecfreaks.com/store

## 9.3. Background

### What is the environmental monitoring

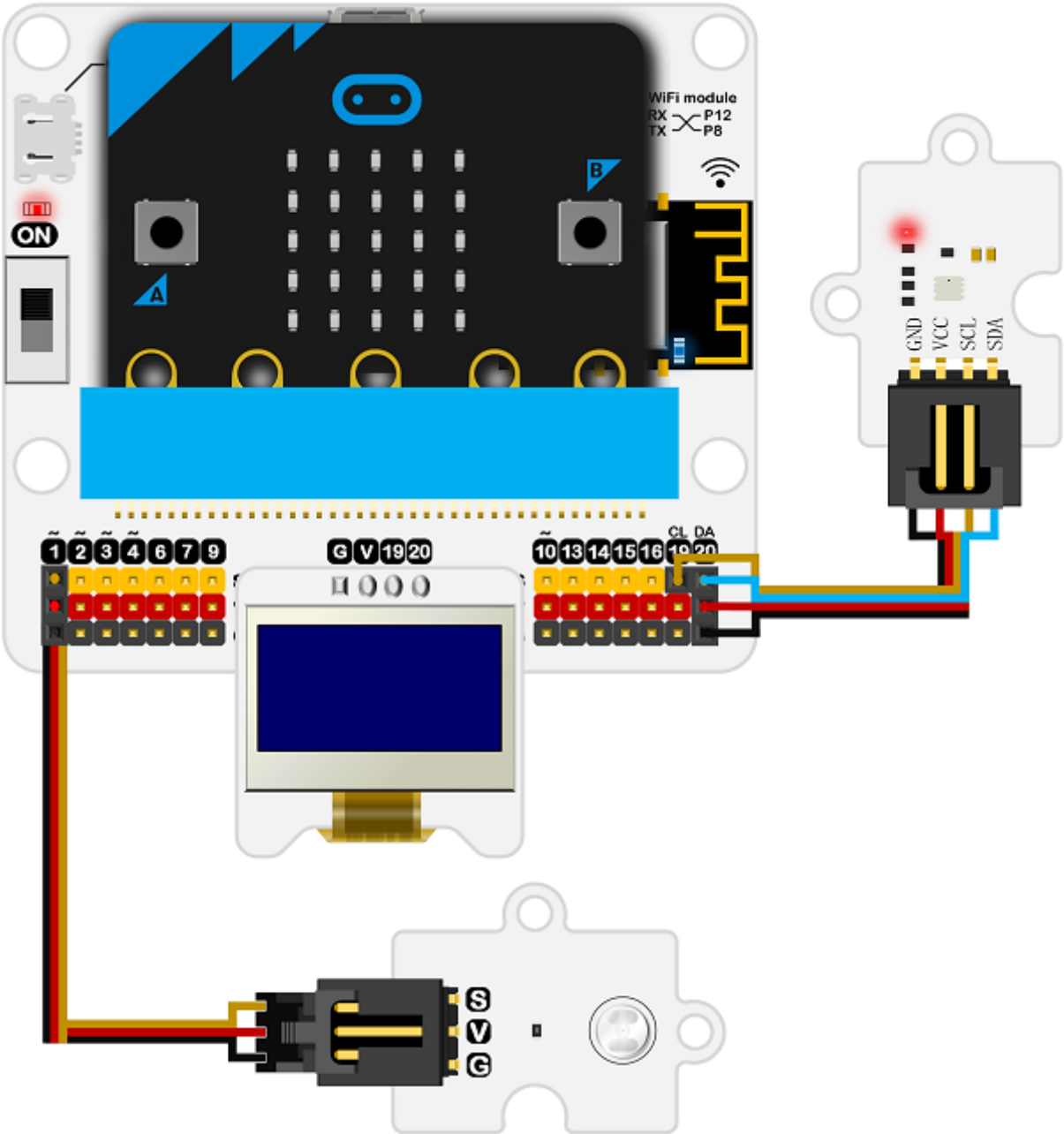
- Environmental monitoringn Environmental monitoring is to design environemntal monitoring and display & save the collected information by using of GIS, it can also give a detailed analysis towards it's monitoring spot.

## 9.4. Hardware connection

As below picture, let the light sensor be connected to `P1`.

Let the BME280 module be connected to `SCL-P19` `SDA-P20` of the `IIC`.

Let the onboard RTC be connected to `IIC`.



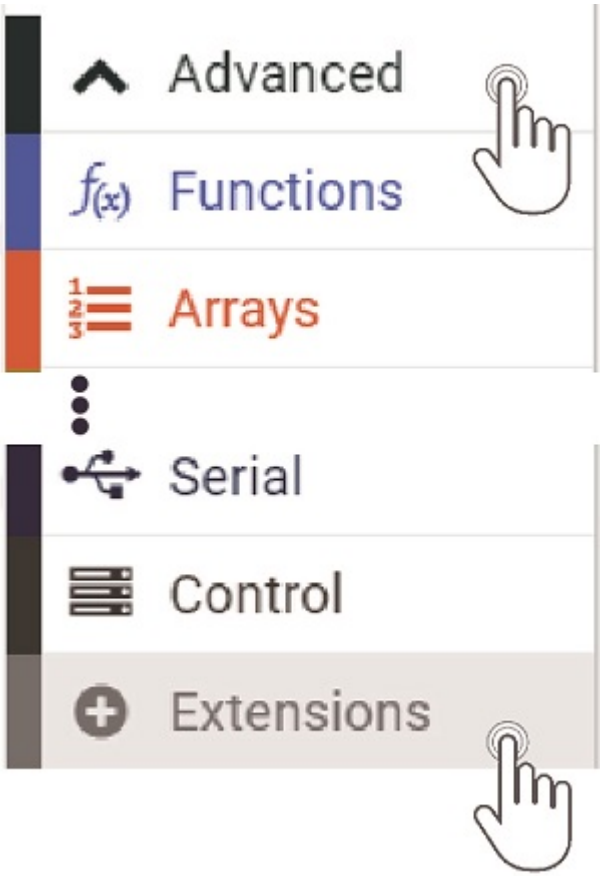

## 9.5. Software

makecode

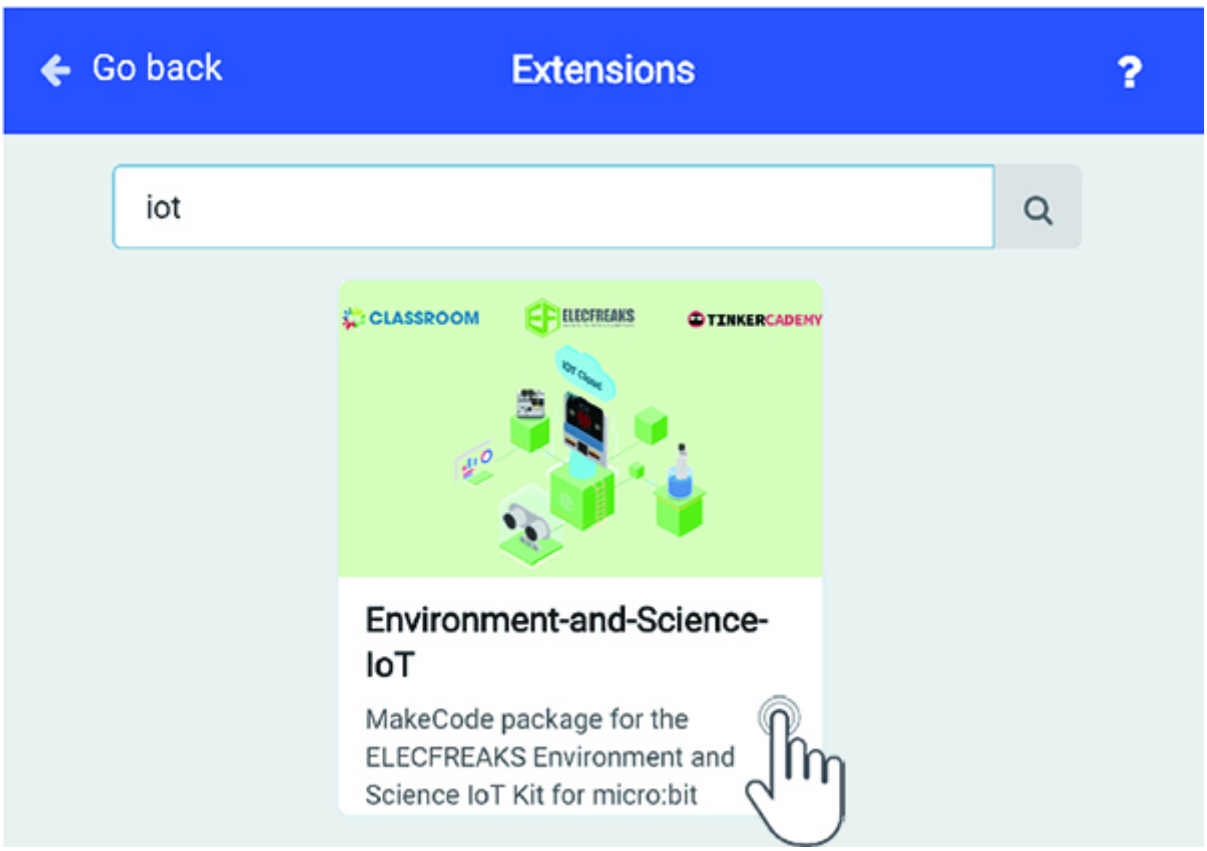
## 9.6. Coding

### Step 1

- Click on “Advanced” in the MakeCode Drawer to see more code sections.




- We need to add an extension for coding to the IOT. Click on the “Extension” at bottom of coding drawer, then Search for “IOT” and click on the IOT package to add it to your project. (As below picture)




Note：If you get a warning telling you some packages will be removed because of incompatibility issues, either follow the prompts or create a new project in the Project file menu.

## Step 2

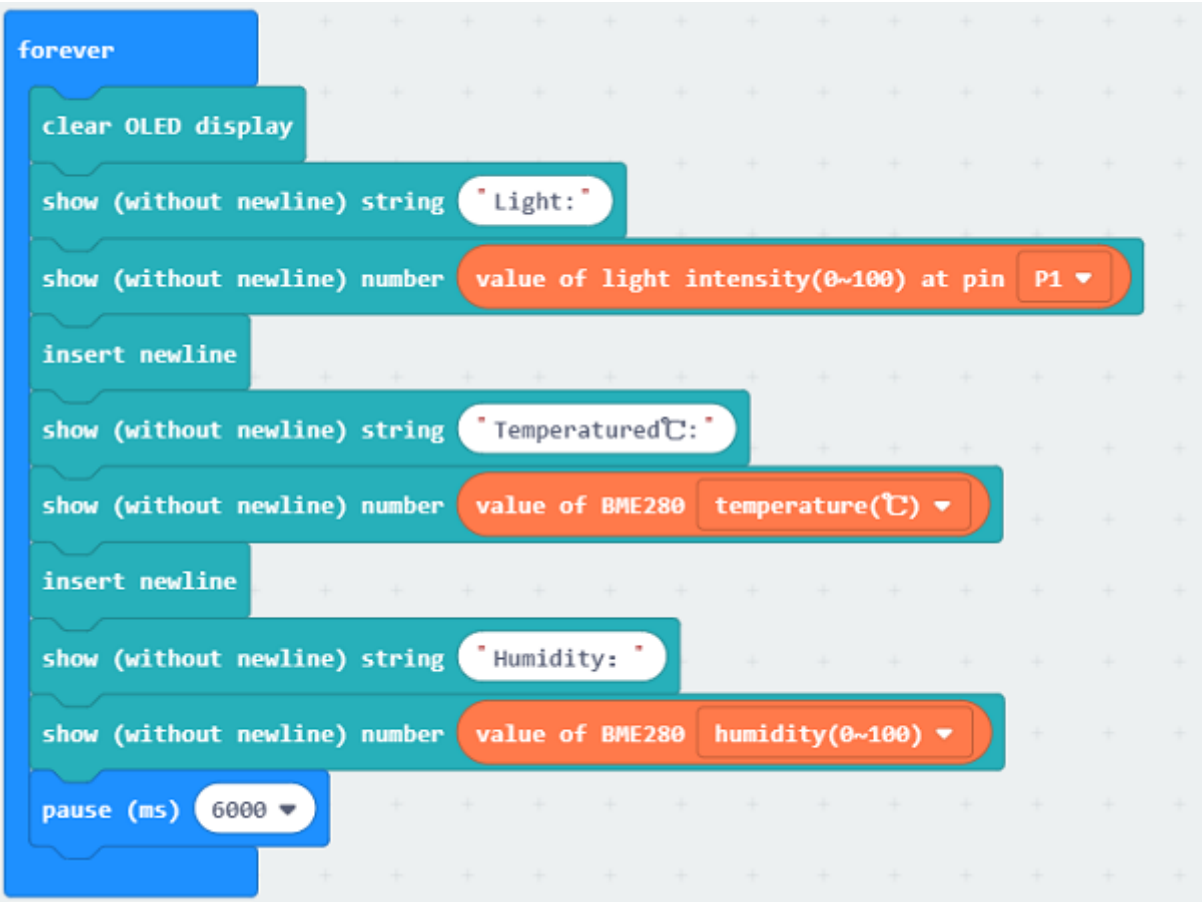
Snap the `initialize OLED` in to the `on start`, initialize OLED screen pixels to `64*128`.


## Step 2

Snap the `show string` and `show number` blocks into the `forever` in turn.

Display current light intensity, humidity and temperature.

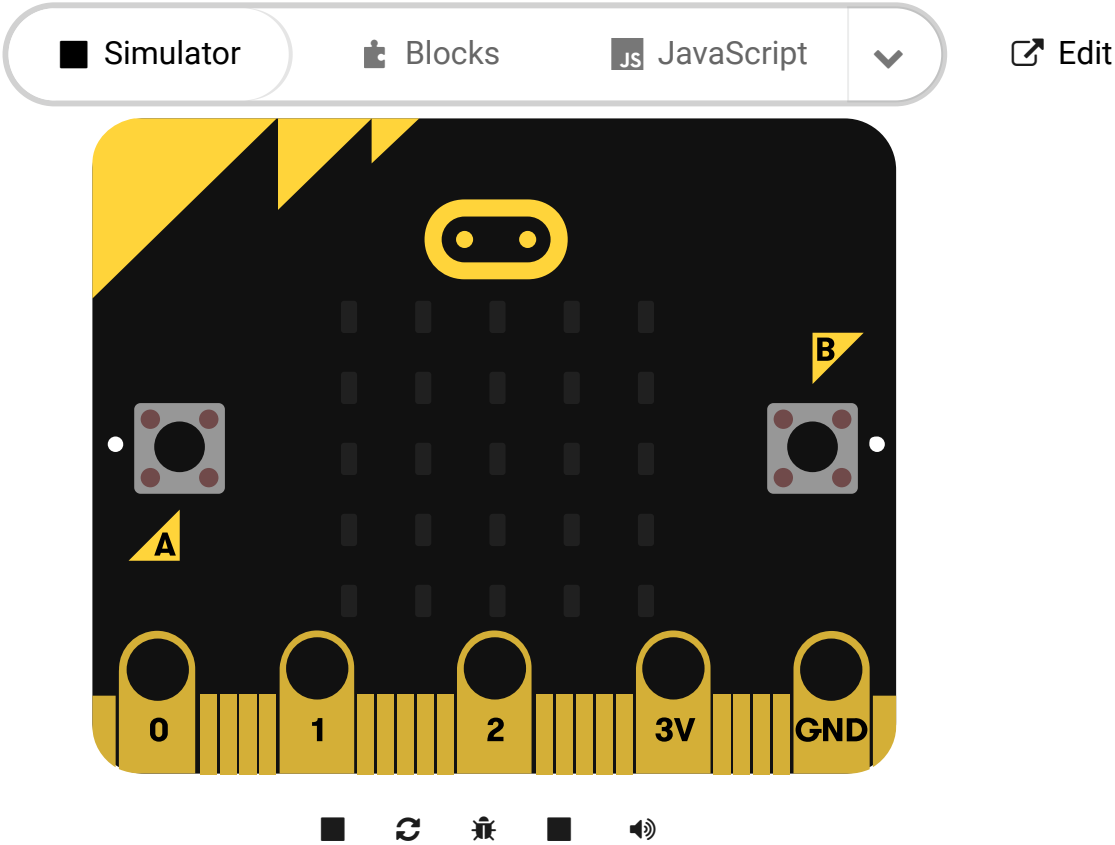
Then, snap into the `insert newline`.




## Program

Program link：https://makecode.microbit.org/_ePDLFuUwqUhK

If you don't want to type these code by yourself, you can directly download the whole program from the link below:




### Result

The light intensity, temperature and humidity are displayed every minute.

## 9.7. Think

How do you count the data in a day ?

## 9.8. Questions

## 9.9. More Information

# 10. iot:kit case04: An automatic irrigation detection

## 10.1. Our goal

- Let's make an automatic irrigation detection.

## 10.2. Required materials

- 1 x IOT:kit IOT:kit:https://www.elecfreaks.com/store

## 10.3. Background

### What is the automatic irrigation detection

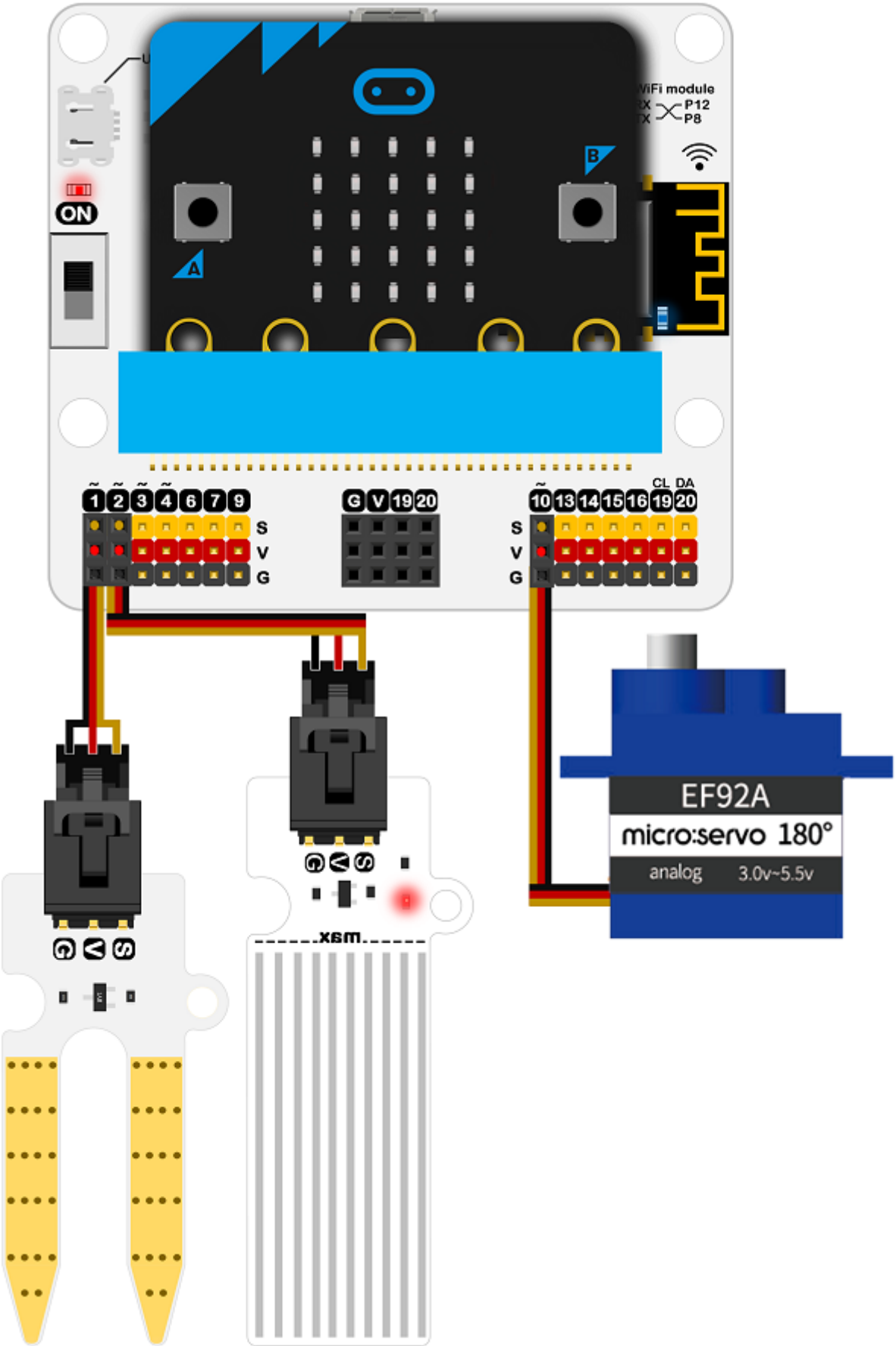
- automatic irrigation detection An automated irrigation system refers to the operation of the system with no or just a minimum of manual intervention beside the surveillance. Almost every system (drip, sprinkler, surface) can be automated with help of timers, sensors or computers or mechanical appliances. It makes the irrigation process more efficient and workers can concentrate on other important farming tasks. On the other hand, such a system can be expensive and very complex in its design and may needs experts to plan and implement it.

## 10.4. Hardware connection

As below picture, let the soil moisture sensor be connected to `P1`.

Let the water level sensor be connected to `P2`.

Let the 180° servo be connected to `P10`.




## 10.5. Software

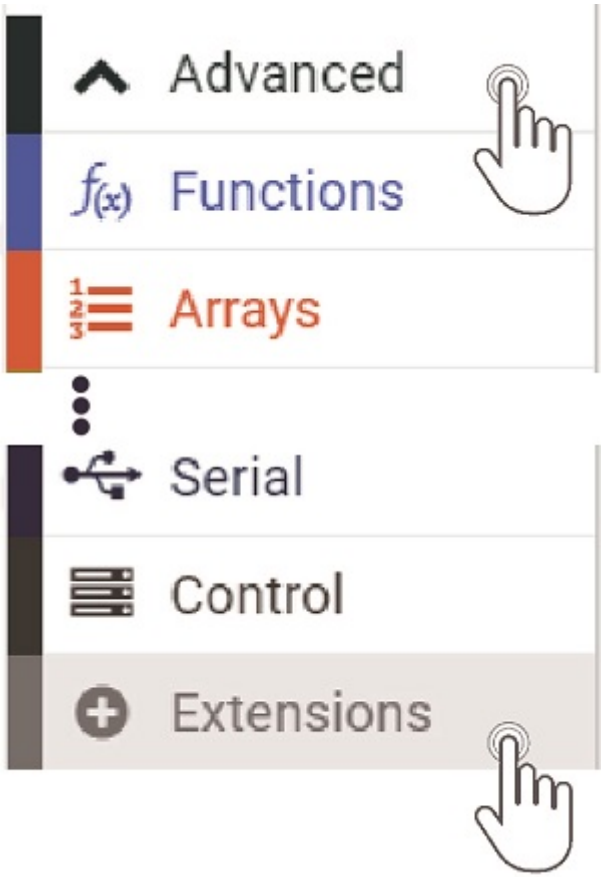
---

makecode

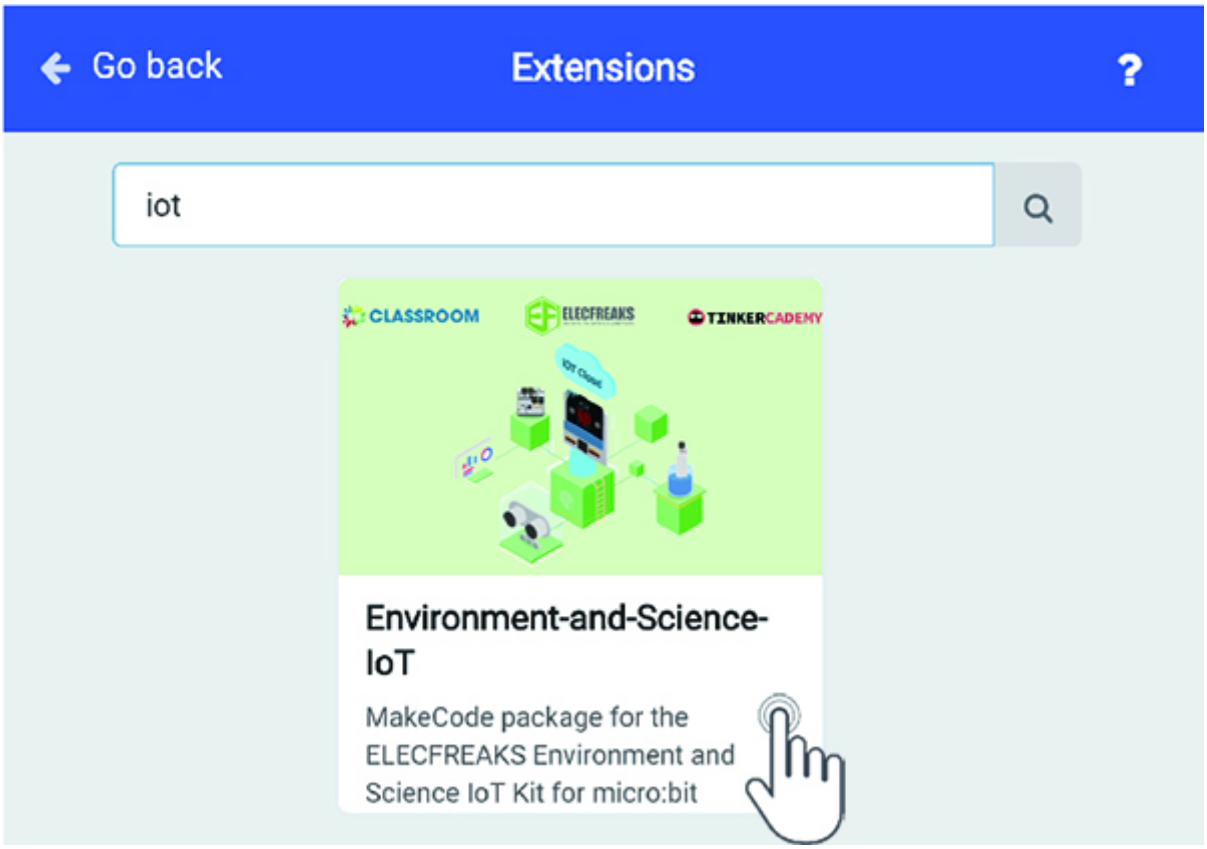
## 10.6. Coding

## Step 1

- Click on “Advanced” in the MakeCode Drawer to see more code sections.



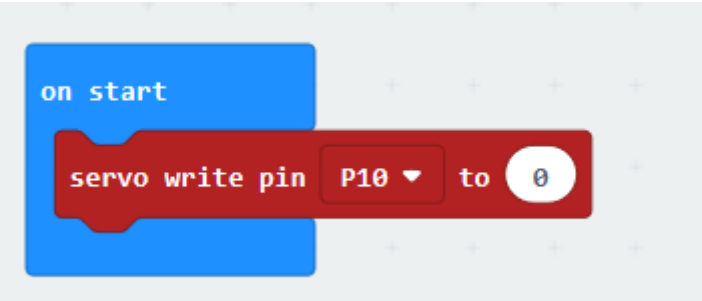

- We need to add an extension for coding to the IOT. Click on the “Extension” at bottom of coding drawer, then Search for “IOT” and click on the IOT package to add it to your project. (As below picture)




Note： If you get a warning telling you some packages will be removed because of incompatibility issues, either follow the prompts or create a new project in the Project file menu.

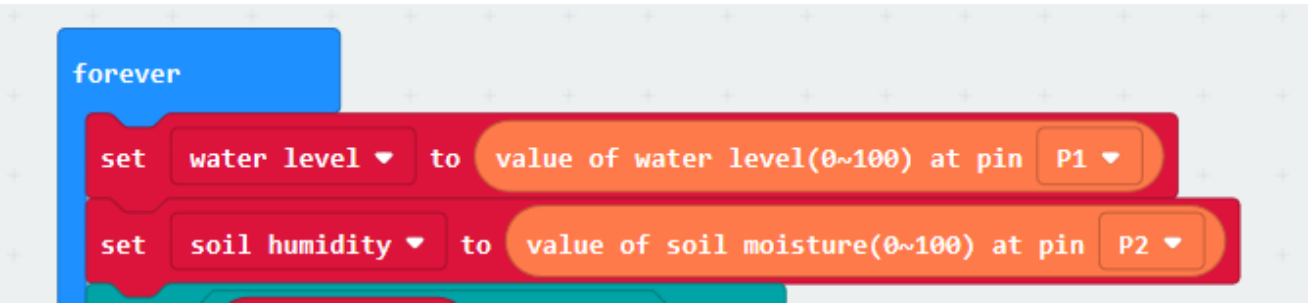
## Step 2

Snap the `servo write` block into the `on start`, write in `0` to turn the servo to 0 deg for shuting the valve.



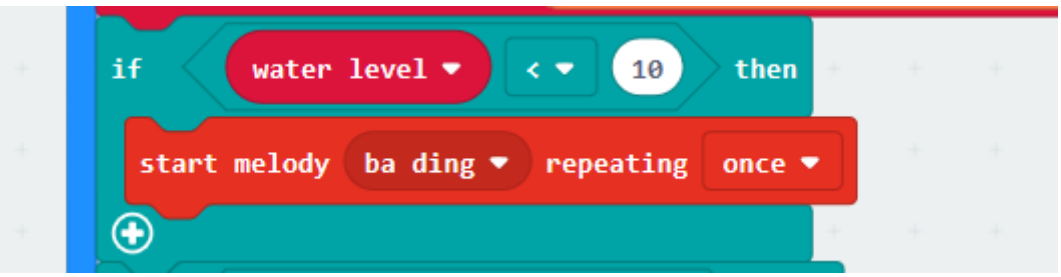

## Step 3

Snap the `set to` block into the `forever`. Get the value of water level and assignment it to the `water level` variable and the value of soil moisture value to the `soil humidity` variable.



## Step 4

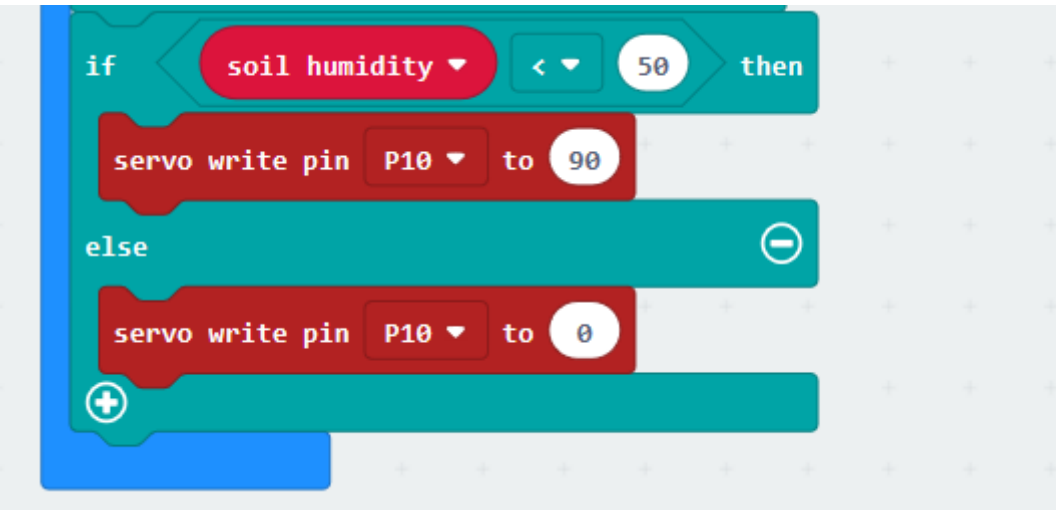
To judge if the water level is lower than 10, if it is, play a bading for adding water.




## Step 5

To judge if the soil humidity is lower than 30, that is if it is need to watering.

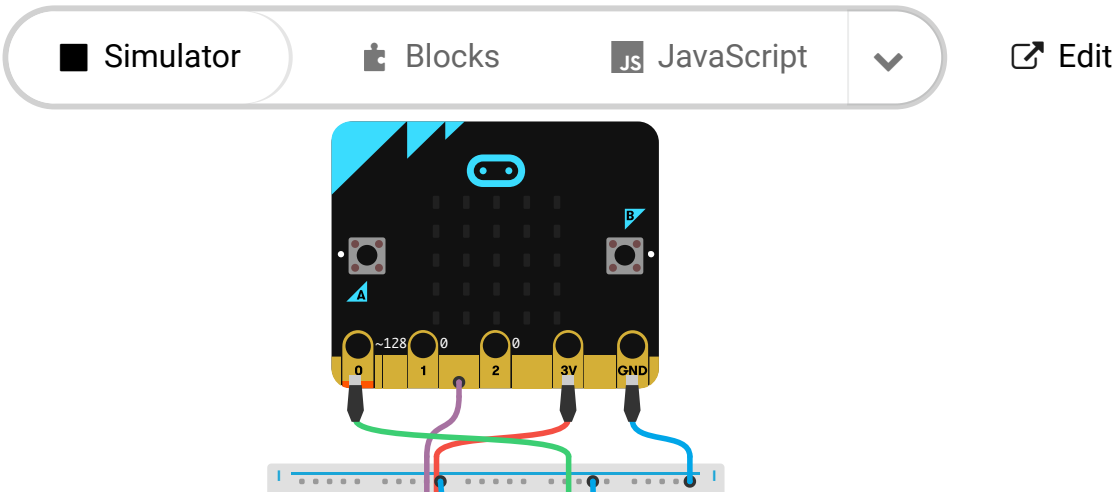
If it lowers than 30, valve opened and watering; if it not lowers than 30, valve shuted.

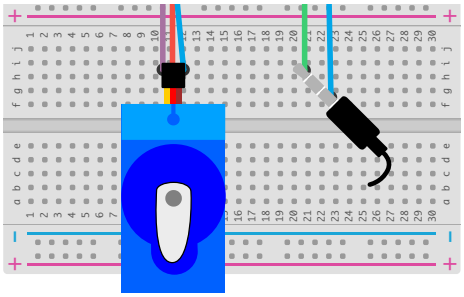



## Program

Program link：https://makecode.microbit.org/_dxTF0K1be0w0

If you don't want to type these code by yourself, you can directly download the whole program from the link below:





### Result

## 10.7. Think

## 10.8. Questions

## 10.9. More Information

# 11. iot:kit case05: A self defense monitoring station

## 11.1. Our goal

- Let's make a self defense monitoring station.

## 11.2. Required materials

- 1 x IOT:kit IOT:kit:https://www.elecfreaks.com/store

## 11.3. Background

### What is self defense?

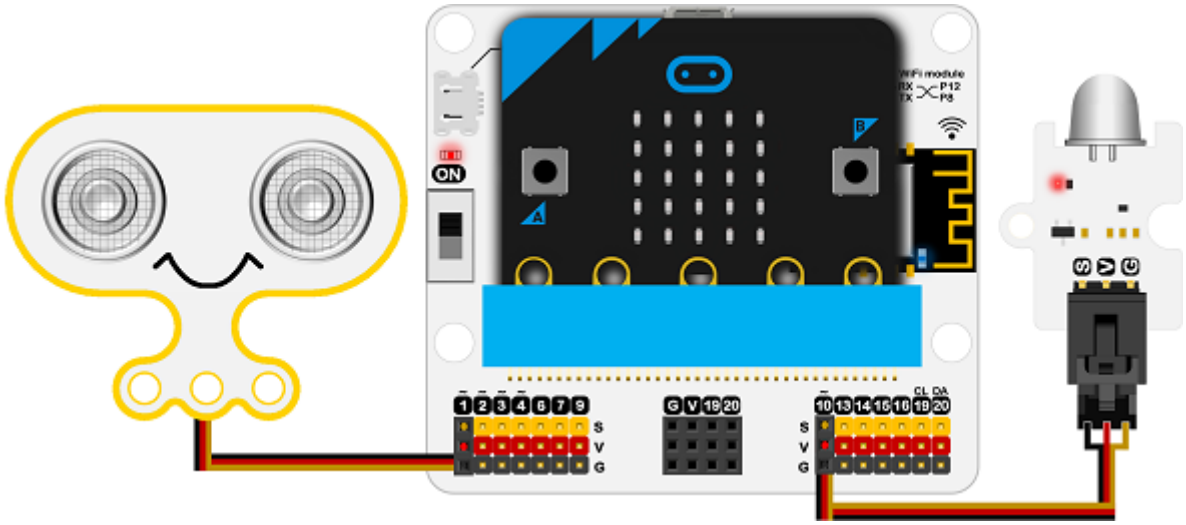
- You need to set a self defense device when your environmental monitor station is out of doors to prevent device damage and to warn people keep away from this device.

## 11.4. Hardware connection

As below picture, let the ultrasonic module be connected to `P1`.

Let the human infrared sensor module be connected to `P10`.

Let the onboard buzzer be connected to `P0`.



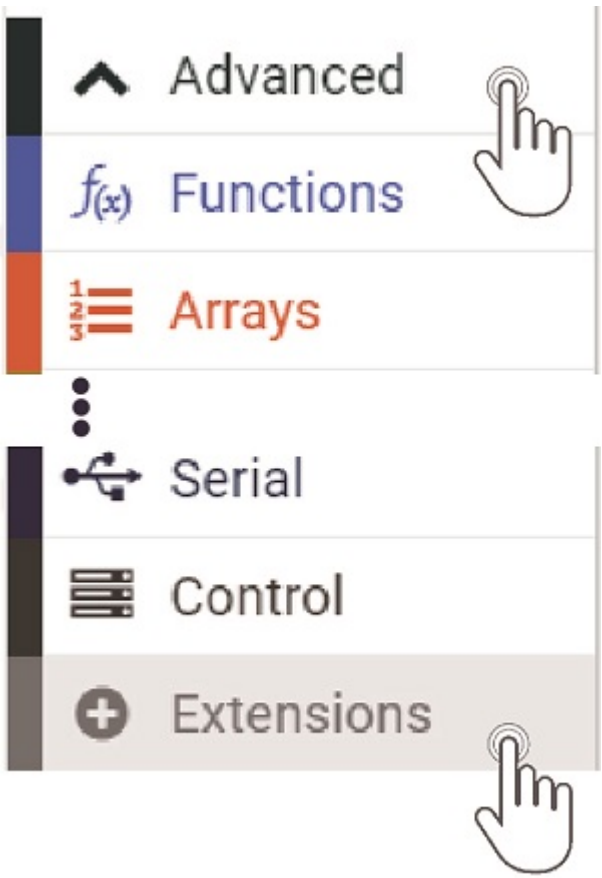

## 11.5. Software

makecode

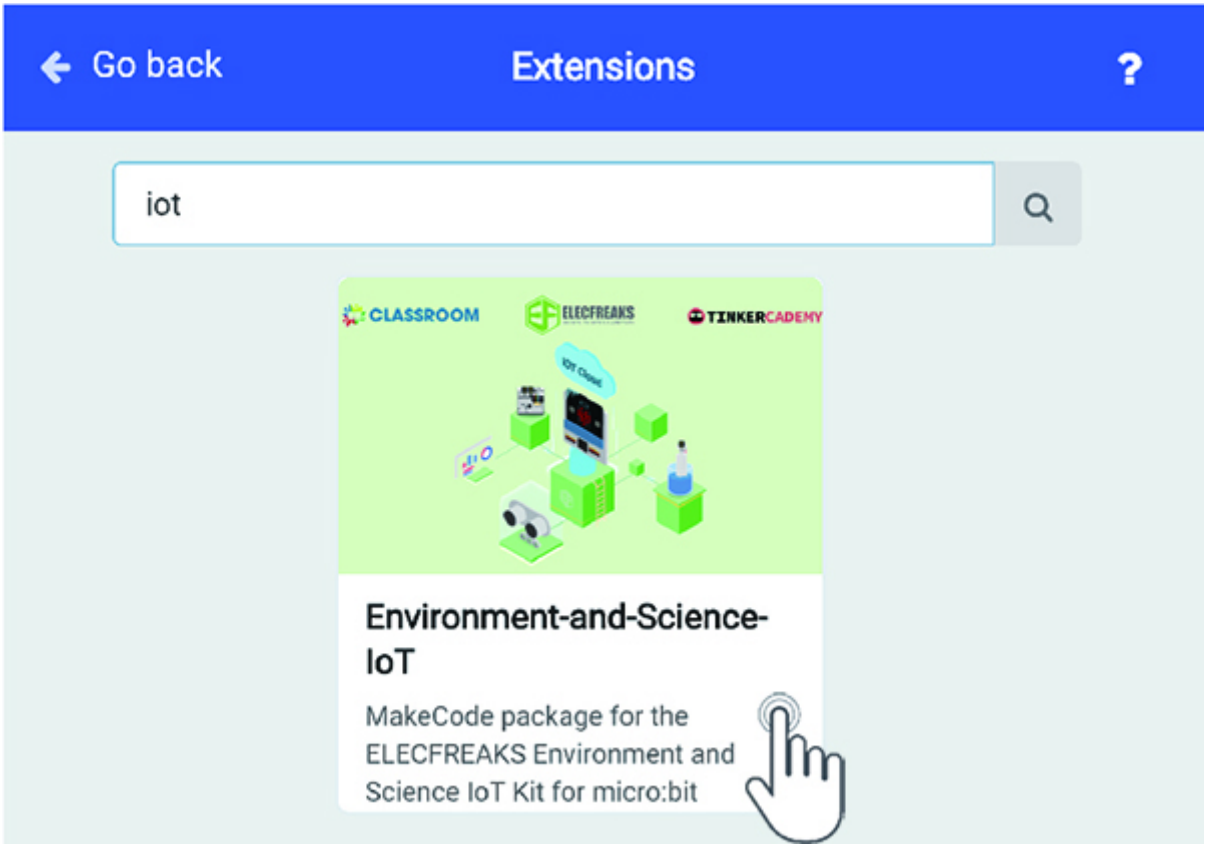
## 11.6. Coding

### Step 1

- Click on “Advanced” in the MakeCode Drawer to see more code sections.



- We need to add an extension for coding to the IOT. Click on the “Extension” at bottom of coding drawer, then Search for “IOT” and click on the IOT package to add it to your project. (As below picture)



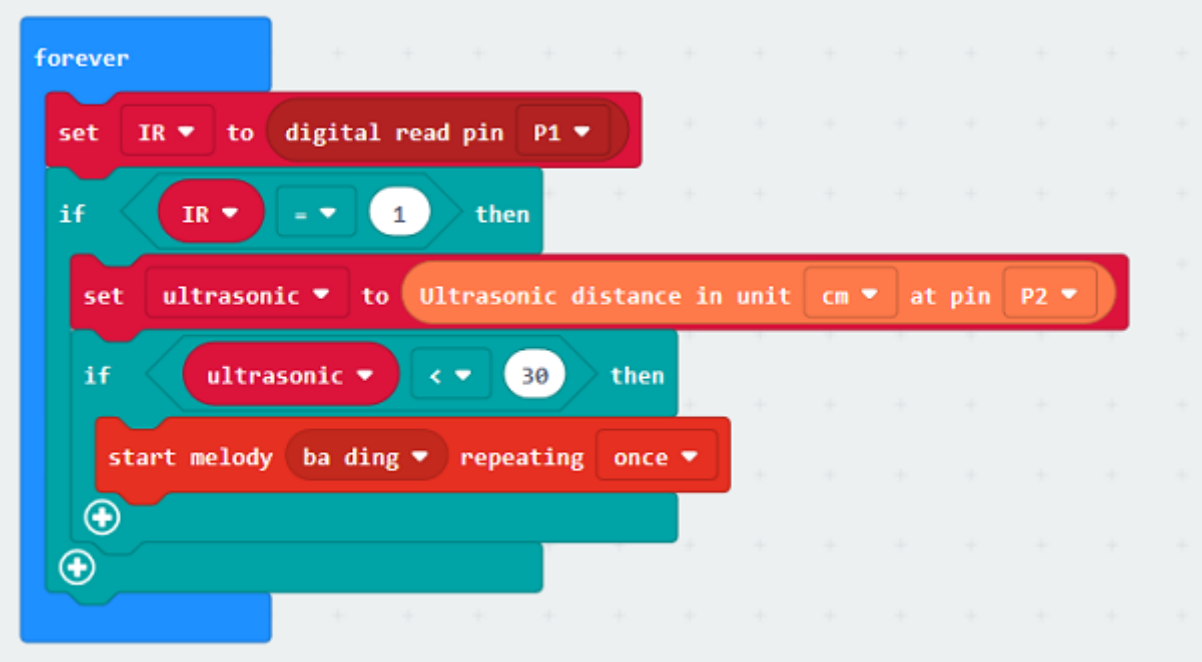

Note：If you get a warning telling you some packages will be removed because of incompatibility issues, either follow the prompts or create a new project in the Project file menu.

## Step 2

Get value from `P1` and assignment it to `IR` block uner `forever`, to judge if `IR` is equal to `1`.

If the `IR` parameter is equal to `1`, get the ultrasonic returned value again and assignment to the `ultrasonic` variable.

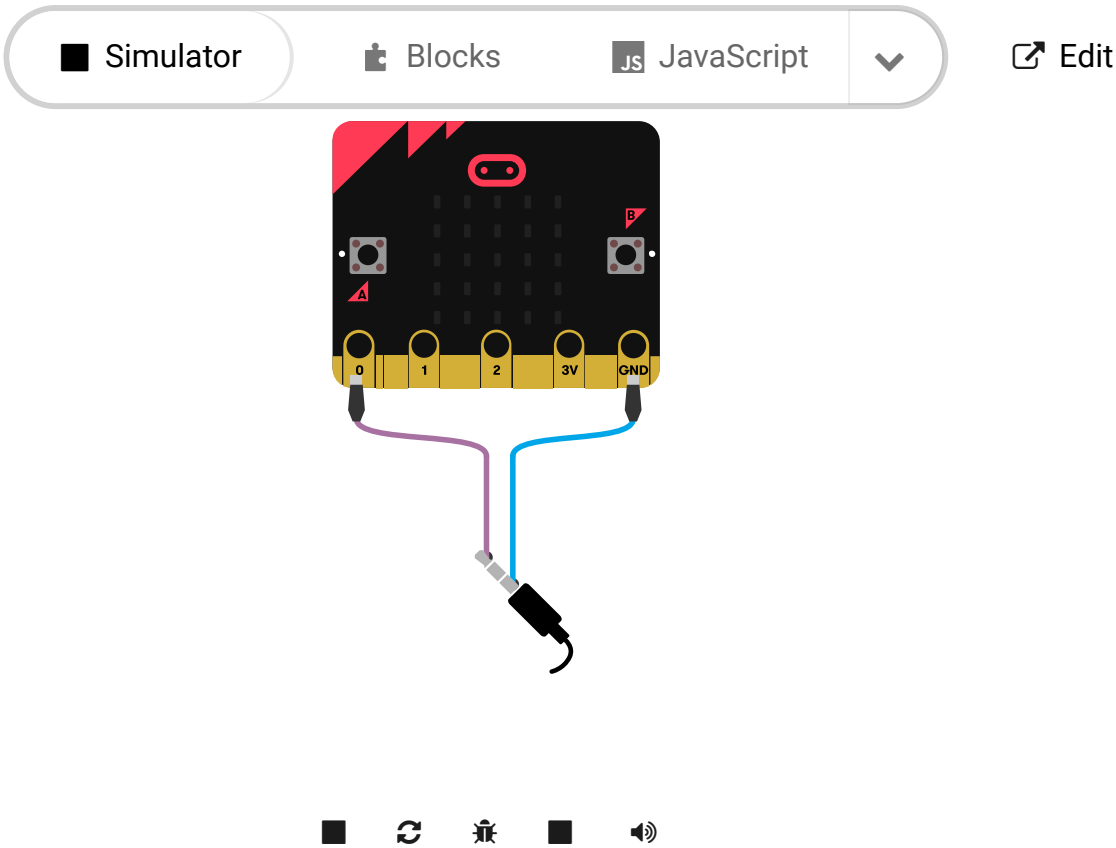
If the `ultrasonic` variable is lower than 30, play a `ba ding` to warning.




## Program

Program link：https://makecode.microbit.org/_05sYuyciH93g

If you don't want to type these code by yourself, you can directly download the whole program from the link below:




## Result

It will play a remind sound when detecting creatures approching in a near distance.

## 11.7. Think

## 11.8. Questions

## 11.9. More Information