

2023-2024 Quality Education Fund Thematic Network - Tertiary Institutes

STEAM Education with Self-directed and Progressive Learning of
Engineering Design Process for Problem-solving
透過STEAM教育自主及循序漸進學習以工程設計流程解難

Smart Shelter

迦密中學 Carmel Secondary School

TEMPERATURE
SENSING

The text "TEMPERATURE SENSING" is written in a bold, rounded, yellow font. The word "TEMPERATURE" is on the top line, and "SENSING" is on the bottom line, with a horizontal yellow underline beneath it. The text is surrounded by various hand-drawn decorative elements: red stars and diamonds, blue stars and diamonds, blue circles, blue wavy lines, and pink wavy lines. The overall style is whimsical and colorful.

<https://makecode.microbit.org/>

Microsoft | micro:bit

Blocks JavaScript

Search...

- Basic
- Input
- more
- Music
- Led
- Radio
- Loops
- Logic
- Variables
- Math
- Octopus
- OLED
- RTC1307
- ESP8266_IoT
- Extensions
- Advanced

forever

show number temperature (°C)

Show data Simulator

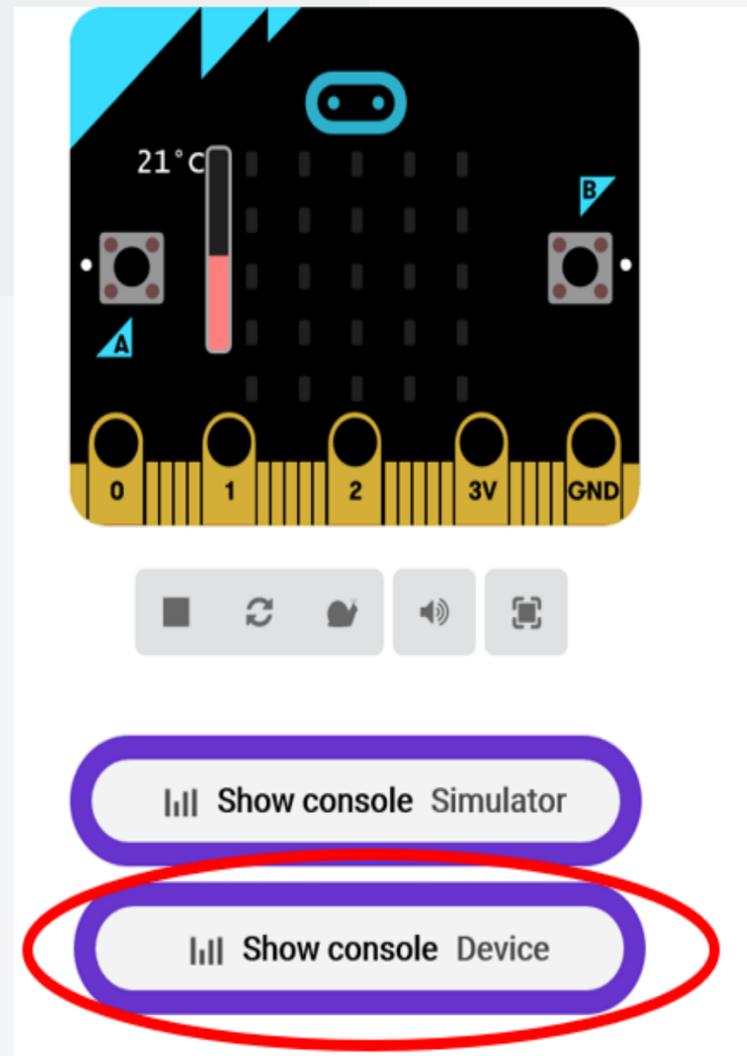
Download

dht11 #2

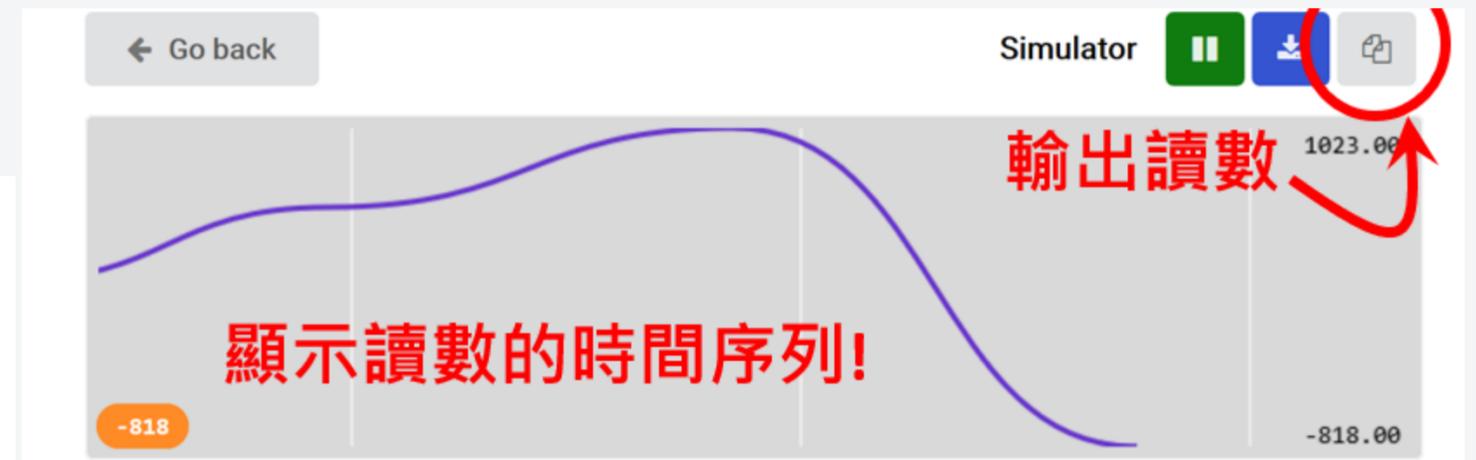
```
1 from microbit import *
2
3 while True:
4     if button_a.was_pressed():
5         display.scroll(temperature())
```

micro:bit Datalogger

```
forever
  show number temperature (°C)
  serial write line temperature (°C)
  pause (ms) 1000
```



- Show console



- Show console device → Export data → Copy text



RADIO

The word "RADIO" is written in a bold, blue, sans-serif font. It is surrounded by decorative elements: wavy pink lines radiate from the left and right sides, and short, straight yellow lines radiate from the top and bottom. The entire graphic is centered between two horizontal black lines.

Indoor-outdoor temperature



micro:bit A (transmitter, outdoor)

```
on start
  radio set group 23
```

```
forever
  radio send number temperature (°C)
  pause (ms) 5000
```

micro:bit B (receiver, indoor)

```
on start  
  radio set group 23  
  
on radio received receivedNumber  
  set outdoorTemp to receivedNumber
```

The code for micro:bit B consists of two event-driven blocks. The first block, 'on start', is a blue block containing a pink 'radio set group' block with the value '23'. The second block, 'on radio received', is a pink block with a red 'receivedNumber' input field, containing a red 'set outdoorTemp' block with a dropdown menu and a red 'to receivedNumber' block.

```
on button A pressed  
  show number temperature (°C)  
  
on button B pressed  
  show number outdoorTemp
```

The code for micro:bit B also includes two button-press event blocks. The first block, 'on button A pressed', is a pink block containing a blue 'show number' block with a pink 'temperature (°C)' input field. The second block, 'on button B pressed', is a pink block containing a blue 'show number' block with a red 'outdoorTemp' input field.

```
1 from microbit import *
2 import radio
3 radio.config(group=23)
4 radio.on()
5
6 while True:
7     radio.send(str(temperature()))
8     sleep(5000)
```

```
1 from microbit import *
2 import radio
3 radio.config(group=23)
4 radio.on()
5 outdoorTemp = '-'
6
7 while True:
8     message = radio.receive()
9     if message:
10         outdoorTemp = message
11     if button_a.was_pressed():
12         display.scroll(str(temperature()))
13     if button_b.was_pressed():
14         display.scroll(outdoorTemp)
```

Task 1

Micro:bit A:

Measure temperature of the room

Micro:bit B:

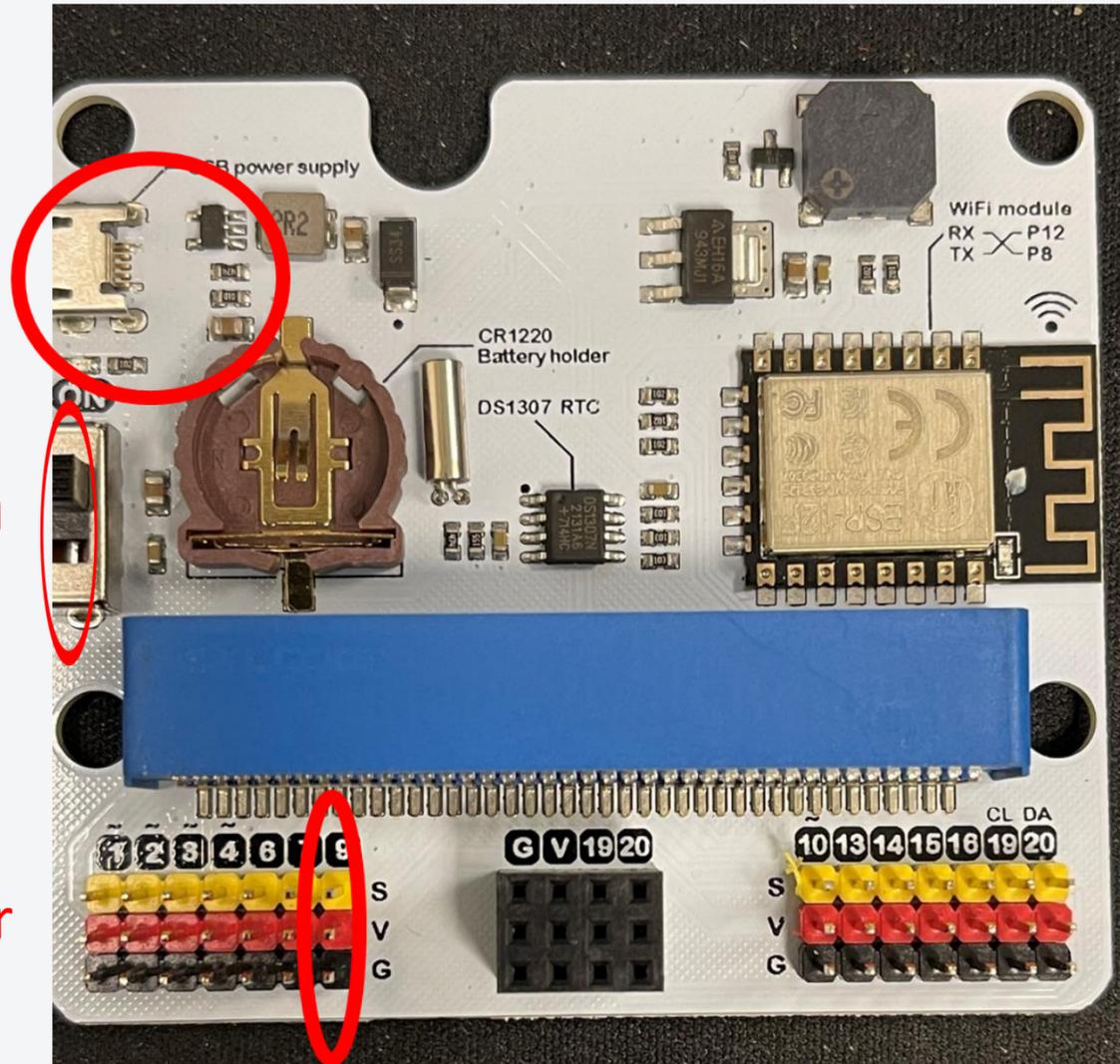
Receive temperature of the room by Radio
and display the display automatically in
your computer

iot:bit

Power supply for
iot:bit

Switch

S = Data pin
V = 3.3v power
G = GND

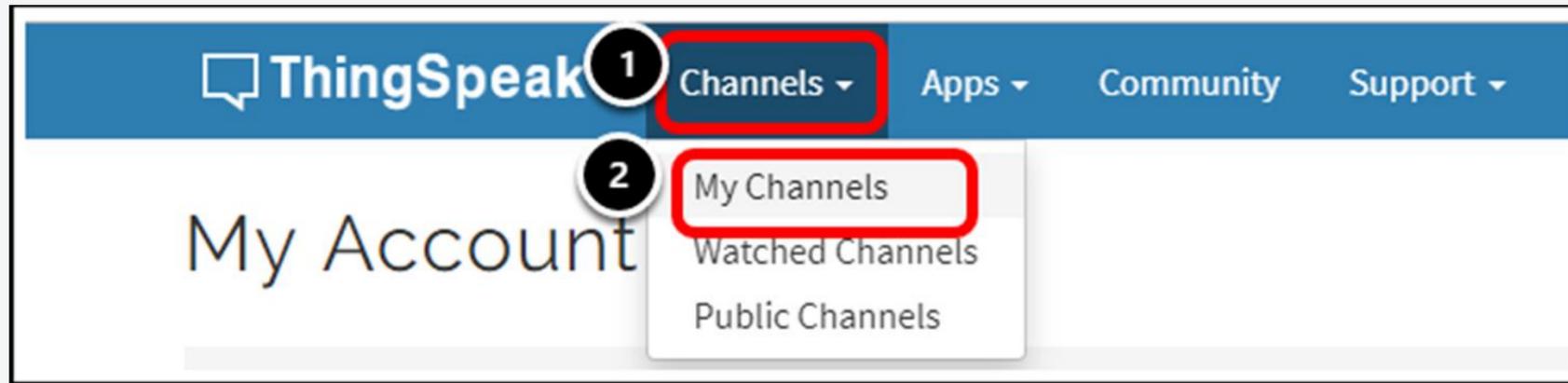


ThingSpeak Set Up

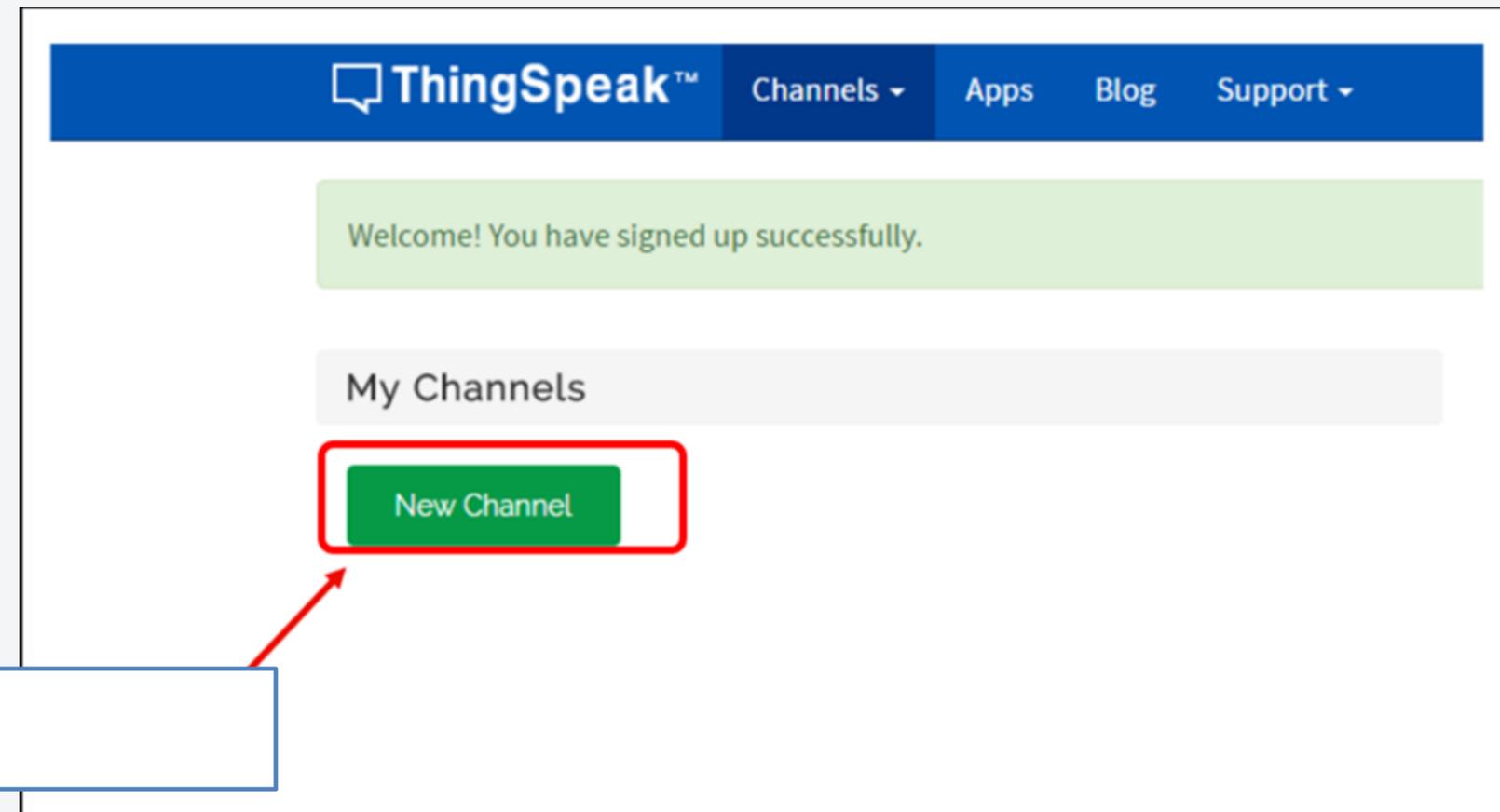
<https://thingspeak.com>

The screenshot shows the ThingSpeak website's sign-up page. The navigation bar includes 'ThingSpeak™', 'Channels', 'Apps', 'Community', 'Support', 'Commercial Use', 'How to Buy', 'Sign In', and 'Sign Up'. The main heading is 'Sign up for ThingSpeak'. Below it, a paragraph explains that free accounts have limitations and that users must create a MathWorks account. The 'Create MathWorks Account' section contains a form with the following fields: 'Email Address' (with a red error message 'Missing required information'), 'Location' (set to 'United States'), 'First Name', and 'Last Name'. A note states: 'To access your organization's MATLAB license, use your school or work email.' To the right, a diagram illustrates the data flow: 'SMART CONNECTED DEVICES' send data to a cloud labeled 'DATA AGGREGATION AND ANALYTICS ThingSpeak™', which then connects to a 'MATLAB' computer for 'ALGORITHM DEVELOPMENT SENSOR ANALYTICS'.

ThingSpeak Set Up



“My Channels”



Create New Channel

ThingSpeak Set Up

Private View

Public View

Channel Setting

Sharing

API Keys

Data Import / Exp

Channel Settings

Percentage Complete 30%

Channel ID 2139603

Name dht11

Description

Field 1 temp



Field 2 humid



Save Channel

He

Chan
can h
in a c

Cha

**Input Channel & Field
information**

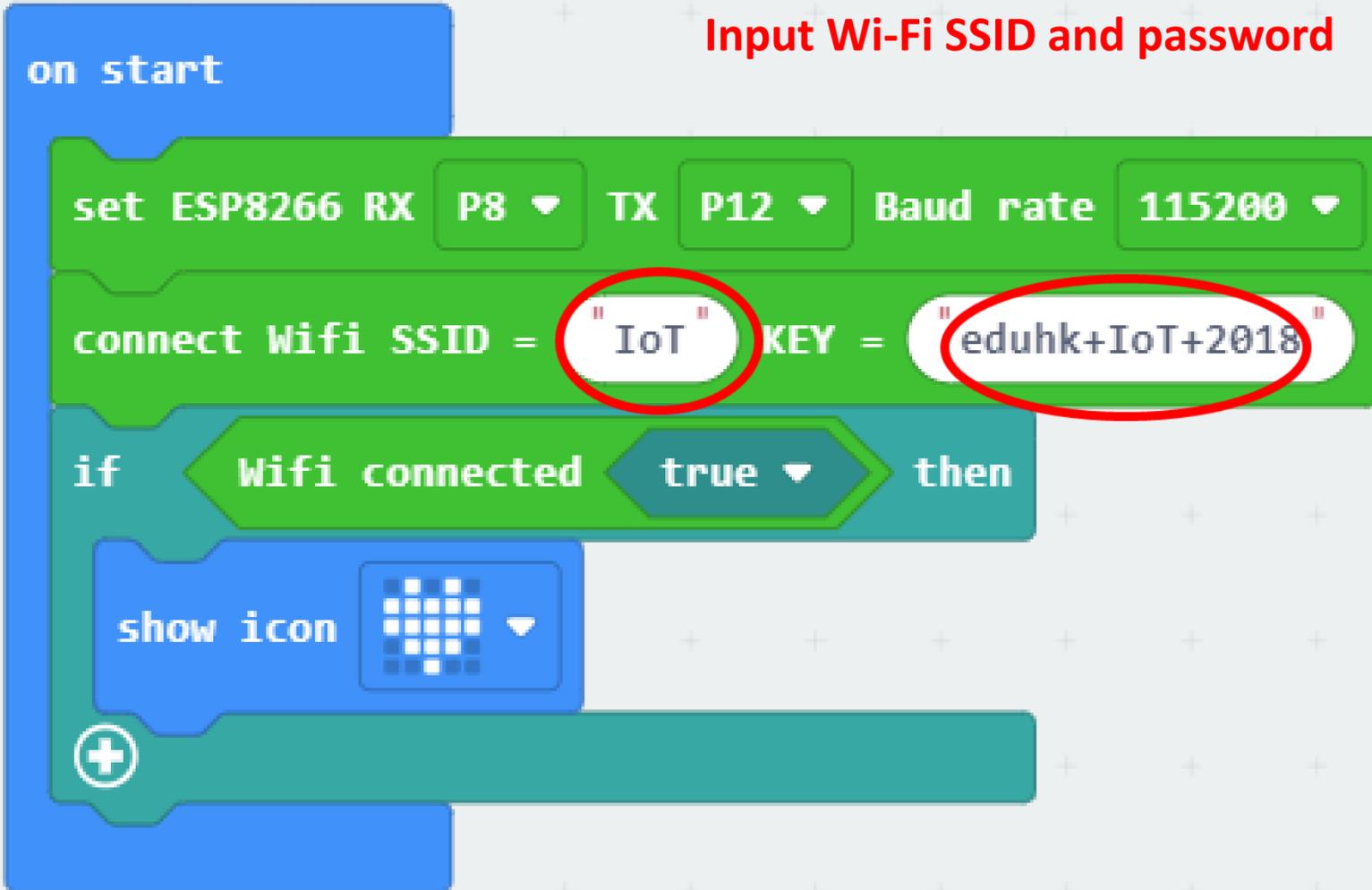
MakeCode extensions

- Basic
- Input
- Music
- Led
- Radio
- Loops
- Logic
- Variables
- Math
- Extensions**

The screenshot shows the MakeCode Extensions marketplace interface. A search bar at the top contains the text "iot-environment-kit". Below the search bar, two extension cards are displayed. The first card, titled "iot-environment-kit", features a green background with an illustration of various IoT devices and a micro:bit. Below the illustration, the text reads "iot-environment-kit" and "Environment and Science IoT Kit for micro:bit". A "Learn more" button is located at the bottom right of this card. The second card, titled "minode", has a white background with the "mi:node" logo in colorful letters. Below the logo, the text reads "minode" and "mi:node kit(micro:bit IoT Starter Kit by element14) driver package for". A "Learn more" button is also present at the bottom right of this card. An orange arrow points from the search bar to the "iot-environment-kit" card. Another orange arrow points from the "Learn more" button of the "iot-environment-kit" card to the "minode" card. A red circle highlights the "Extensions" menu item in the left sidebar.

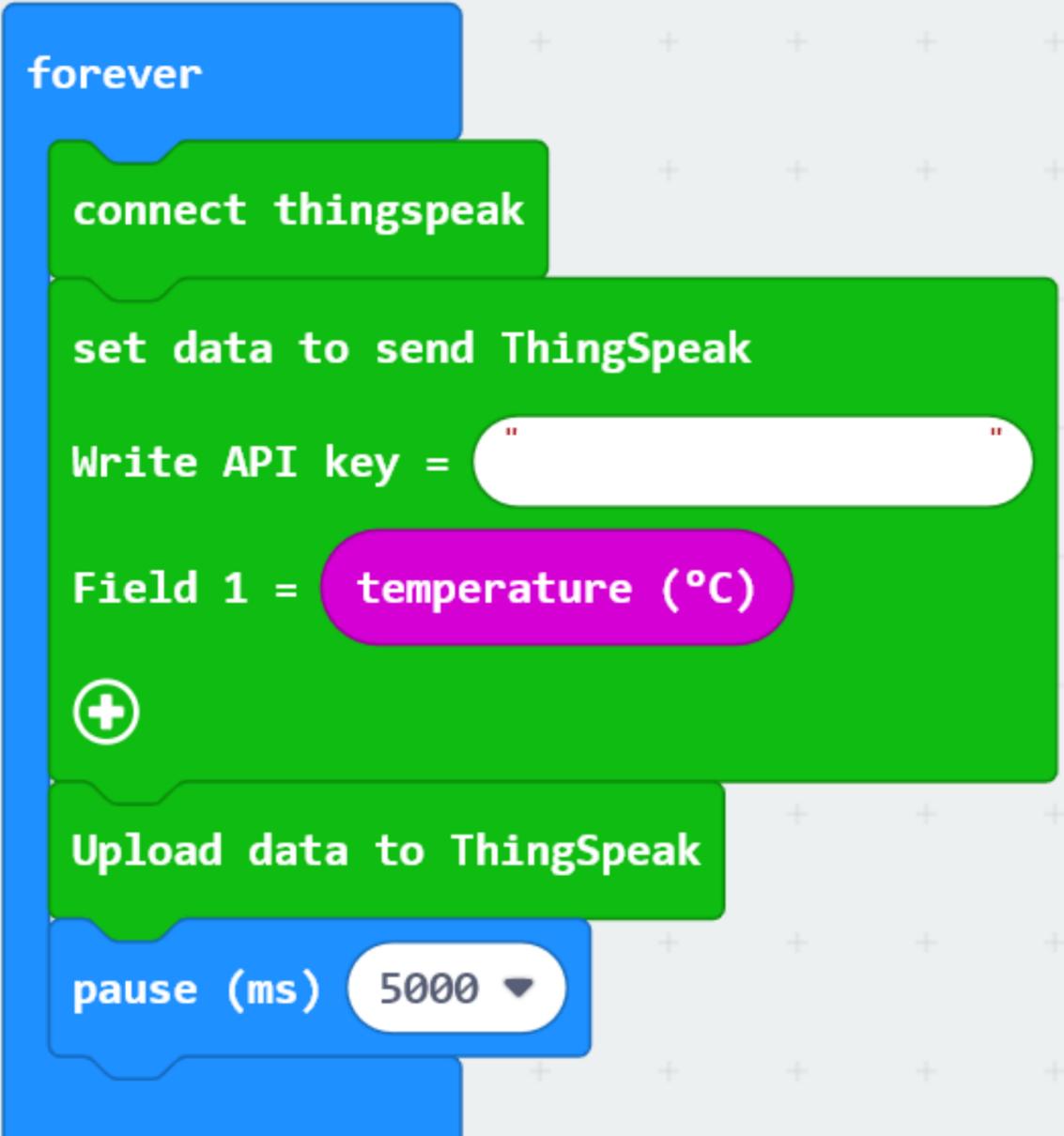
iot:bit WiFi setting

Input Wi-Fi SSID and password



The code starts with an 'on start' block. It then sets the ESP8266 RX pin to P8 and the TX pin to P12, with a baud rate of 115200. Next, it connects to a WiFi network with the SSID 'IoT' and the key 'eduhk+IoT+2018'. Both the SSID and key are circled in red. An 'if' block checks if the WiFi is connected (set to 'true'). If so, it shows an icon of a grid.

```
on start
  set ESP8266 RX P8 TX P12 Baud rate 115200
  connect Wifi SSID = "IoT" KEY = "eduhk+IoT+2018"
  if Wifi connected true then
    show icon [grid icon]
```



The code is enclosed in a 'forever' loop. It connects to 'thingspeak' and sets the data to send to 'ThingSpeak'. The 'Write API key' field is empty. The first field is set to 'temperature (°C)'. There is a plus sign icon below the field. The code then uploads the data to ThingSpeak and pauses for 5000 ms.

```
forever
  connect thingspeak
  set data to send ThingSpeak
  Write API key = ""
  Field 1 = temperature (°C)
  Upload data to ThingSpeak
  pause (ms) 5000
```

ThingSpeak Set Up

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

Write API Key

Key

[Generate New Write API Key](#)

Read API Keys

Key

Note

[Save Note](#) [Delete API Key](#)

Step 5: Copy API Key

ThingSpeak Set Up

```
forever
  connect thingspeak
  set data to send ThingSpeak
  Write API key = your_write_api_key
  Field 1 = temperature (°C)
  Upload data to ThingSpeak
  pause (ms) 5000
```

Paste API Key in Makecode

ThingSpeak Set Up

The screenshot displays the ThingSpeak web interface. At the top, a navigation bar contains several tabs: 'Private View' (highlighted with a red box), 'Public View', 'Channel Settings', 'Sharing', 'API Keys', and 'Data Import / Export'. Below the navigation bar are several action buttons: '+ Add Visualizations', '+ Add Widgets', 'Export recent data', 'MATLAB Analysis', and 'MATLAB Visualization'. On the right side of the navigation bar, it says 'Channel 4 of 4 < >'. Below the navigation bar, there is a 'Channel Stats' section with the following information: 'Created: 7.months.ago', 'Last entry: 22.days.ago', and 'Entries: 0'. At the bottom of the interface, there are two empty chart widgets. The left widget is titled 'Field 1 Chart' and has a y-axis labeled 'temp' and an x-axis labeled 'Date'. The right widget is titled 'Field 2 Chart' and has a y-axis labeled 'humid' and an x-axis labeled 'Date'. Both charts are connected to a device labeled 'dht11'. A white box with a black border and the text 'Click Private View' is overlaid on the right side of the interface, pointing to the 'Private View' tab.

ThingSpeak Set Up

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

Import

Upload a CSV file to import data into this channel.

File No file chosen

Time Zone

Export

Download all of this Channel's feeds in CSV format.

Time Zone

Click Download

ThingSpeak Set Up

Want to clear all feed data from this Channel?

Clear Channel

Want to delete this Channel?

Delete Channel

ThingSpeak Public View Sharing

testing

Channel ID: 1281894

Author: mwa0000020852618

Access: Public

Private View

Public View

Channel Settings

Sharing

API Keys

Data Import / Export

Channel Sharing Settings

- Keep channel view private
- Share channel view with everyone
- Share channel view only with the following users:

Email
Address

Enter email here

Add User

Help

ThingSpeak allows you to control who can view the data in your channel. Irrespective of the settings on this tab, reading data from or writing data to the fields of a channel requires the appropriate API key for the channel.

Channel Sharing Settings

- **Keep channel view private:** Selecting this option keeps your channel private. Only you will be able to see the channel view.
- **Share channel view with everyone:** Selecting this option makes the public view of your channel viewable by anyone browsing the ThingSpeak website.
- **Share channel view only with the following users:** Selecting this option shares the private view of your channel only with specific ThingSpeak users.

Task 2

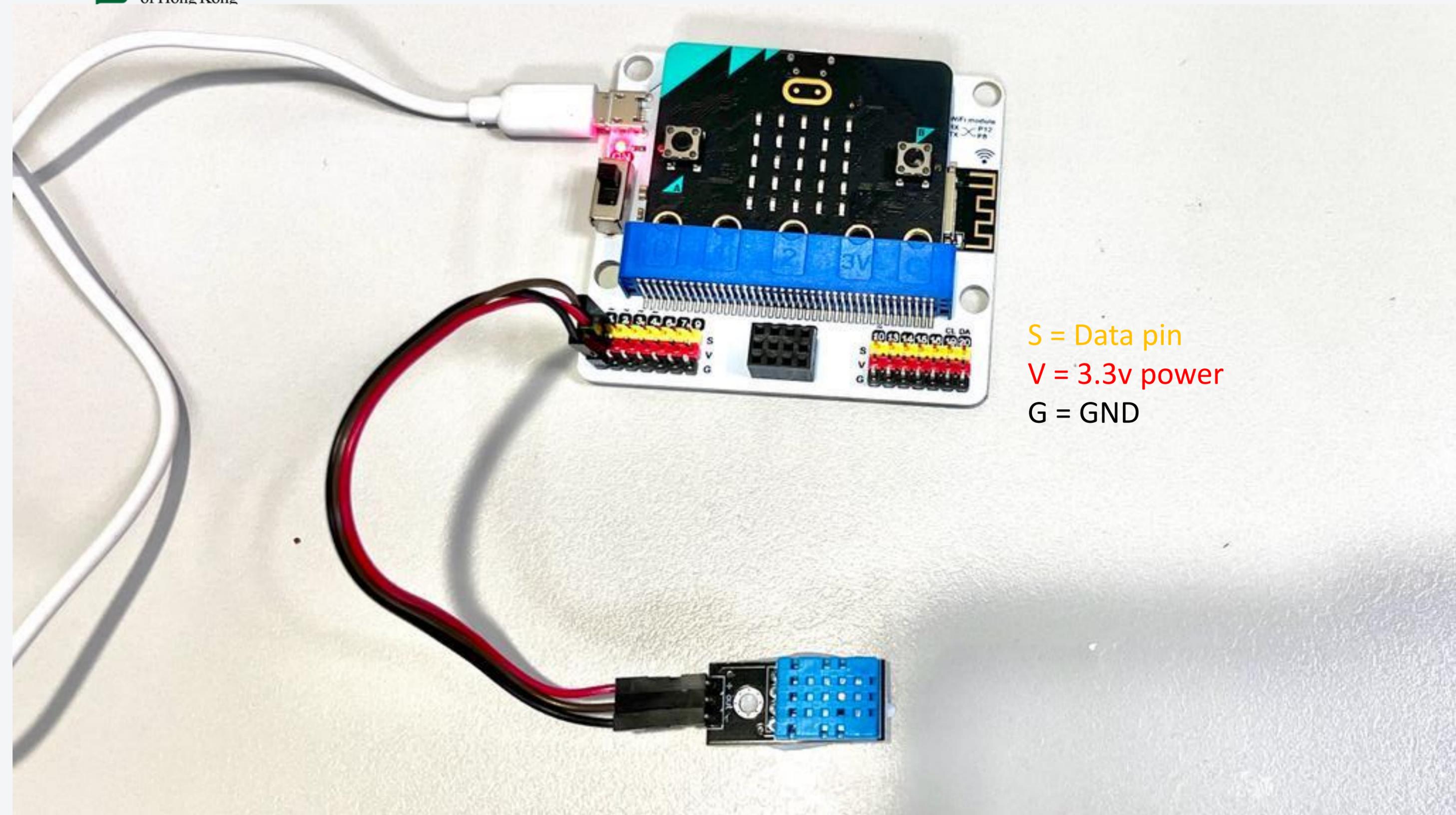
Micro:bit:

Measure temperature of the room and upload the reading to ThingSpeak

ThingSpeak:

Display the reading in the ThingSpeak platform

DHT11



S = Data pin

V = 3.3v power

G = GND

iot:bit WiFi setting

Input Wi-Fi SSID and password

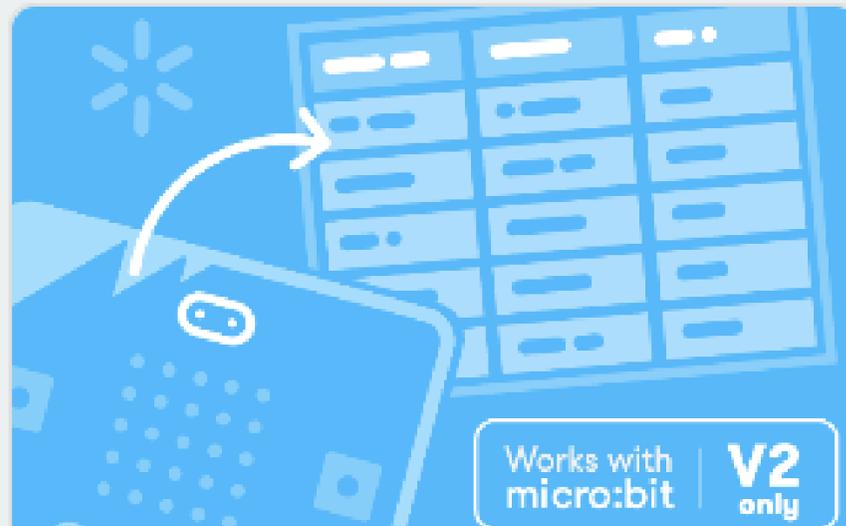
```
on start
  set ESP8266 RX P8 TX P12 Baud rate 115200
  connect Wifi SSID = "IoT" KEY = "eduhk+IoT+2018"
  if Wifi connected true then
    show icon
```

```
forever
  connect thingspeak
  set data to send ThingSpeak
  Write API key = ""
  Field 1 = value of dht11 temperature(°C) at pin P1
  Field 2 = value of dht11 humidity(0~100) at pin P1
  Upload data to ThingSpeak
  pause (ms) 5000
```

Datalogger

Add this Extension

Recommended



datalogger

Data logging to flash memory.
micro:bit (V2) only.

[Learn More](#)

A Scratch script for the Datalogger extension. The script is set on a blue "on start" block. It contains two dark green blocks: "set timestamp" with a dropdown menu set to "seconds", and "set columns" with a text input field containing "Temp" and a plus sign icon. To the right, there is a dark green "on log full" block containing a blue "show icon" block with a dropdown menu showing a grid icon. Below the "on start" block is a green "every" block set to "5000 ms". Inside the "every" block, there are two blocks: a blue "show number" block with a pink text input field containing "temperature (°C)", and a dark green "log data" block with a dropdown menu set to "column", a text input field containing "Temp", the word "value", a pink text input field containing "temperature (°C)", and a plus sign icon.

Datalogger

睇data

> MICROBIT (E:) ▼ ↻ Search MICROBIT (E:)

Name	Date modified	Type	Size
 ASSERT.TXT	22/03/2016 16:30	Text Document	1 KB
 DETAILS.TXT	22/03/2016 16:30	Text Document	1 KB
 MICROBIT.HTM	22/03/2016 16:30	Chrome HTML Do...	1 KB
 MY_DATA.HTM	22/03/2016 16:30	Chrome HTML Do...	124 KB

micro:bit

micro:bit data log

Download Copy Update data... Clear log... Visual preview

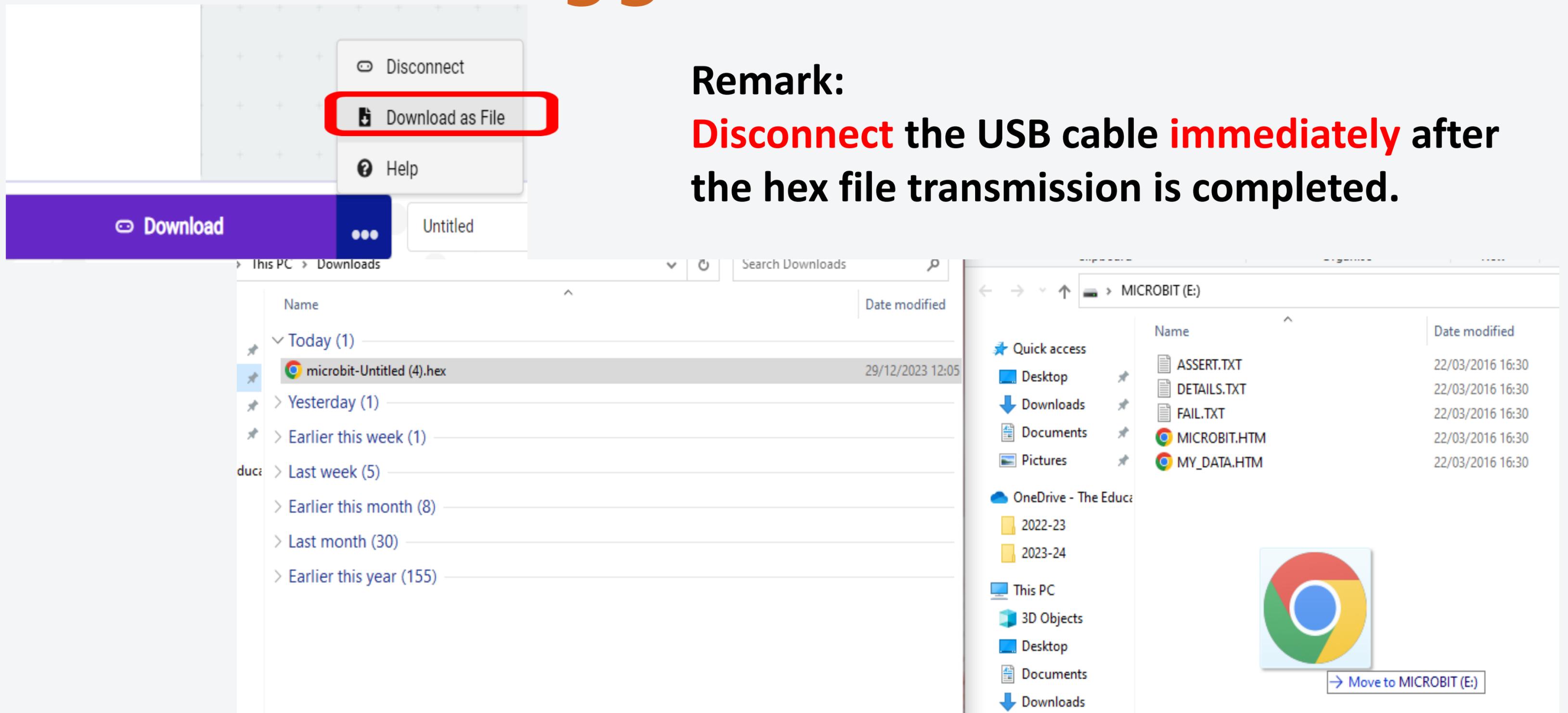
This is the data on your micro:bit. To analyse it and create your own graphs, transfer it to your computer. You can copy and paste your data, or download it as a CSV file which you can import into a spreadsheet or graphing tool. [Learn more about micro:bit data logging.](#)

Time (hours)	temp
0.00	24
0.00	24
0.00	24
0.00	24

Datalogger Clear data

Remark:

Disconnect the USB cable **immediately** after the hex file transmission is completed.



micro:bit data log

[Download](#)[Copy](#)[Update data...](#)[Clear log...](#)[Visual preview](#)

This is the data on your micro:bit. To analyse it and create your own graphs, transfer it to your computer. You can copy and paste your data, or download it as a CSV file which you can import into a spreadsheet or graphing tool. [Learn more about micro:bit data logging.](#)

Time (seconds)	Temp
3.88	25
8.89	26
13.89	26
18.89	26
23.89	26
28.89	26
33.89	26
38.89	26

Task 3

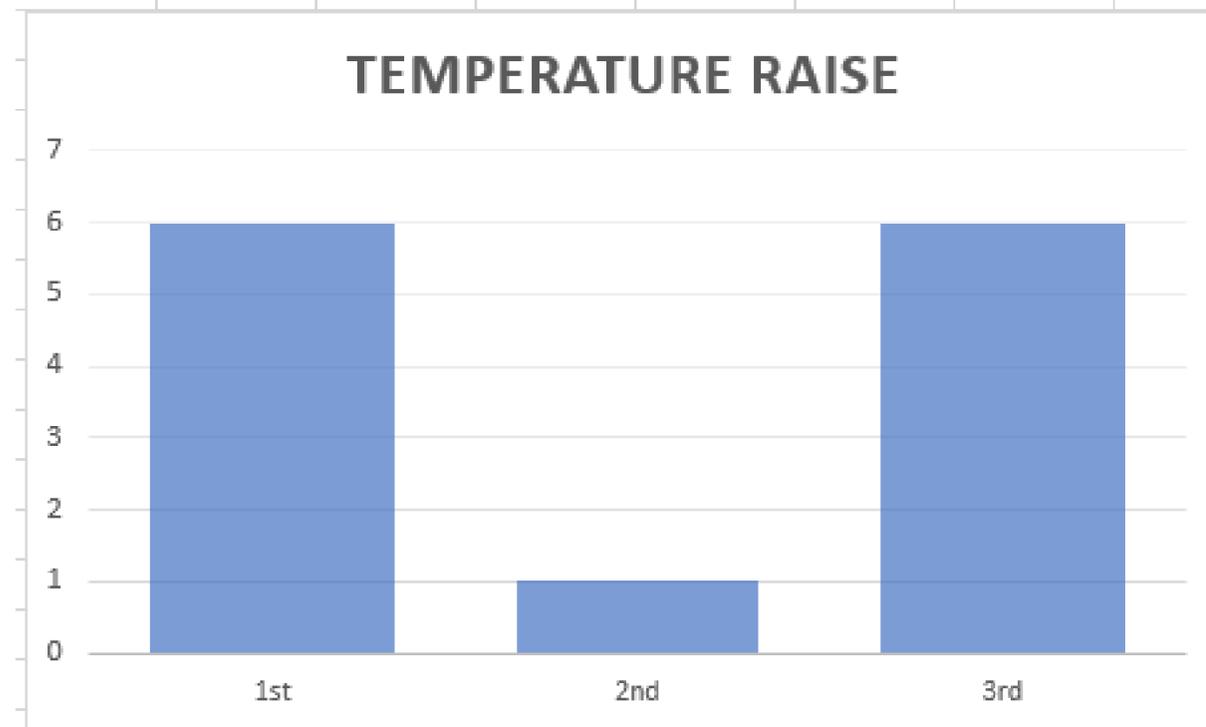
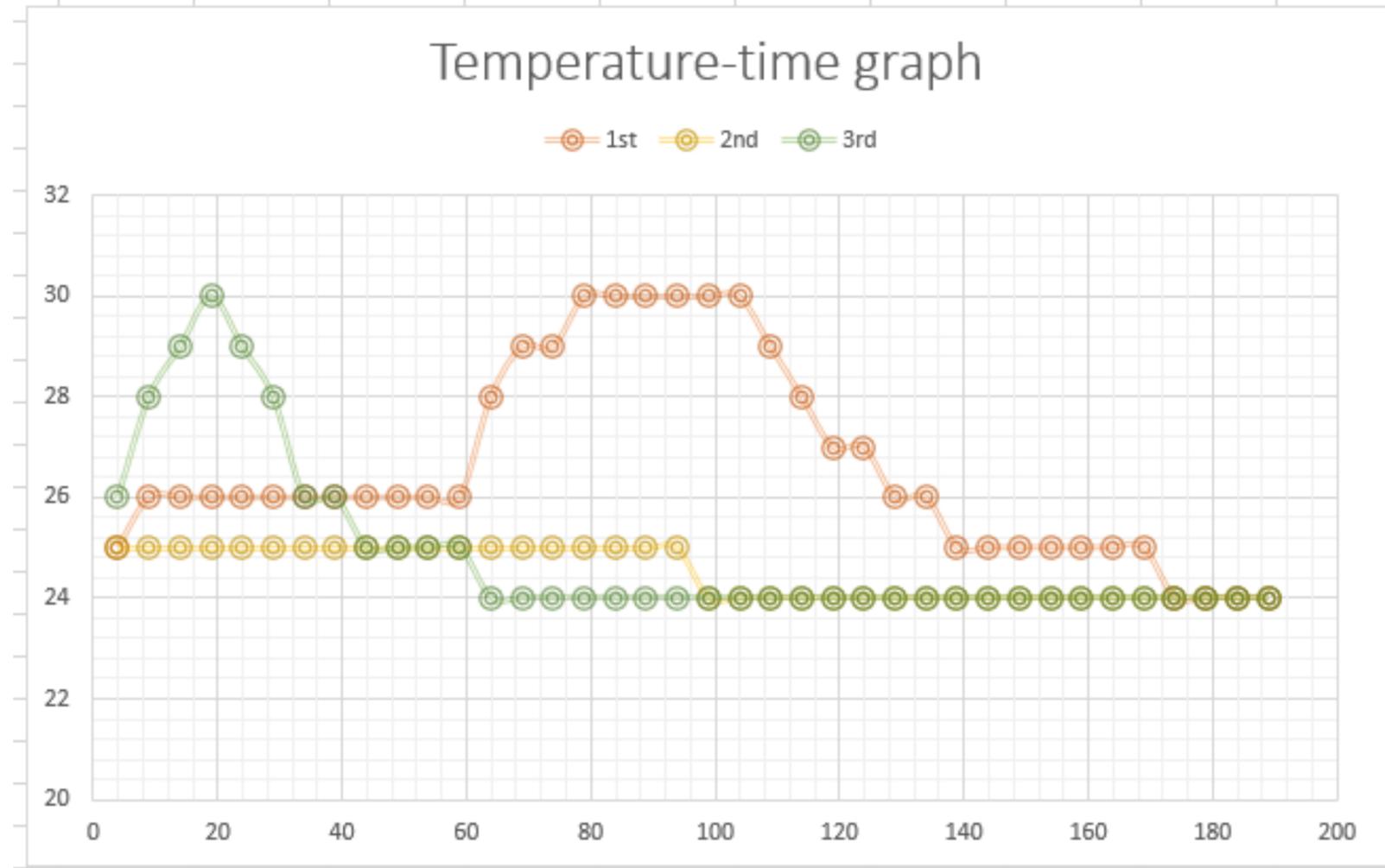
Micro:bit datalogger

Measure the temperature of the room (3 times) using DHT11 sensors and log the values in the Micro:bit.

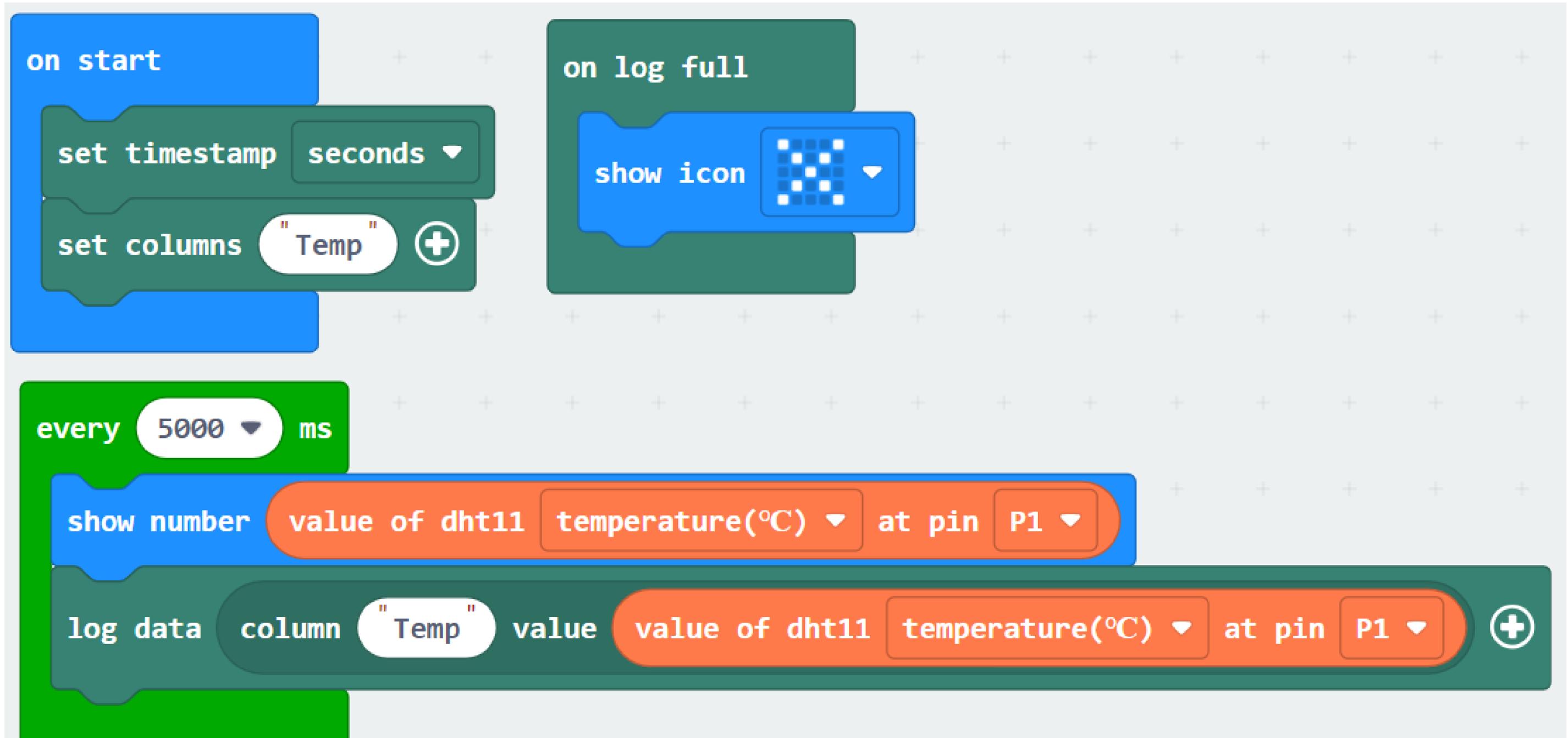
Using Microsoft Excel

Organise the data and create the following graphs.

Submit your file to Google Classroom



Data Collection



The image shows a Scratch script for data collection. It consists of three main blocks: 'on start', 'on log full', and a loop block.

- on start** (blue block):
 - set timestamp to seconds
 - set columns to "Temp"
- on log full** (green block):
 - show icon (grid icon)
- every 5000 ms** (green loop block):
 - show number value of dht11 temperature(°C) at pin P1
 - log data column "Temp" value value of dht11 temperature(°C) at pin P1

Black	White	Aluminium
Group 1,4,7	Group 2,5,8	Group 3,6,9

Task 4: Micro:bit Datalogger Experiment

Objective:

Measure and record the temperature changes during a testing period using DHT11 sensors and log the values using a Micro:bit.

Procedure:

Use the Micro:bit to measure the temperature every 5 seconds for 30 minutes.

(Light on: 15 minutes; Light off: 15 minutes)

Log the temperature values in a file on the Micro:bit.

Download the data from the Micro:bit to a computer.

Open the data file in Microsoft Excel.

Remove any unnecessary data from the file.

Save the file as a .xlsx file, and submit your file to Google Classroom