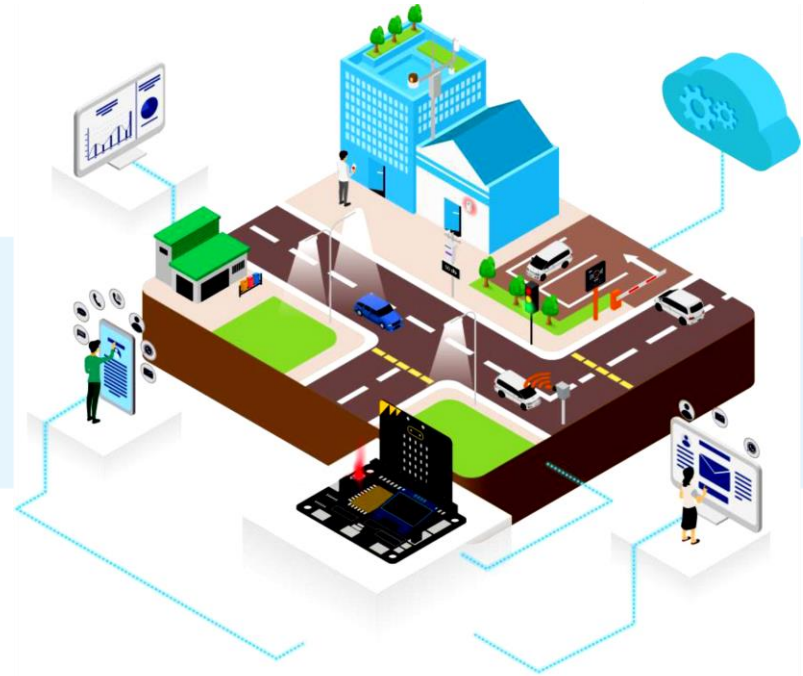




SMART CITY IOT

STARTER KIT FOR MICRO:BIT

Copyright © Smarthon Limited. All rights reserved.



ABOUT SMARTHON



SMARTHON LIMITED, based in Hong Kong, design, manufacture, and sell friendly products for Educators and Creatives focusing on regular course like General Studies. Our mission and vision are to encourage technology in education for all people with high quality of education and learning.

The establishment of SMARTHON is to design simplified devices and apply the ever-changing technology development to teaching. For example, teachers can use Micro:bit and sensors to teach students to design a small greenhouse system and then use the Internet of Things technology to quickly transmit relevant values to smart devices. Teachers can change the temperature or humidity in real-time to allow students to observe different changes, record them, and then use them for study, analysis, and understanding of the operation of the greenhouse system.



CONTENT



- Introduction
- What is Smart City & IoT?..... P. 5-7
- Component List: Parts List & Model List..... P. 8-11
- Smarthon IoT:bit..... P. 12-13
- Getting Started: Add the extension on micro:bit..... P. 14
- Case 1: Automated Smart Playground Lamp..... P. 15-18
- Case 2: Smart Car Park Access Barrier..... P. 19-22
- Case 3: Garbage disposal System..... P. 23-26
- Case 4: Urban Noise Detection..... P. 27-30
- Case 5: Car speed monitoring..... P. 31-36

CONTENT



- IoT Case 6: Weather Station..... P. 37-44
- IoT Case 7: Smart Defense System..... P. 45-51
- IoT Case 8: Automated Traffic Light..... P. 52-60
- IoT Case 9: Smart House Door Control..... P. 61-68
- IoT Case 10: Smart Street Light..... P. 69-75
- IoT Case 11: Roof Garden Clothes Rack..... P. 76-83
- Scenario Example 1-3..... P. 84-86
- Appendix I: Electronic Modules on Smarthon – Further Exploration..... P. 87

SMARTHON SMART CITY IOT STARTER KIT



SMARTHON Smart City IoT Starter kit is designed to introduce Internet of things (IoT). With basic knowledge of computing knowledge and electronics provided in the kit, you can be a city creator and build a unique IoT system in the city. Based on Smarthon IoT board, which is compatible with multiple sensors and actuators, you can design city features; for example: using sensors to detect traffic status and upload city information to the internet.

 <https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/index.html>

WHAT IS SMART CITY?

Smart city is a framework, integrating the information and communications technology (ICT) to improve the city's sustainability, efficiency and quality of life, in 6 major area - mobility, living, environment, people government and economy.



SMART LIVING

Using IoT/technology to improve the convenience / standard of living



SMART GOVERNMENT

government that support the adoption and development of technology in public/ private sectors and city infrastructure



SMART ENVIRONMENT

Monitor pollution and reduce waste, develop a green community



SMART ENCONOMY

strengthen the current economic pillars and develop new pillars with technology



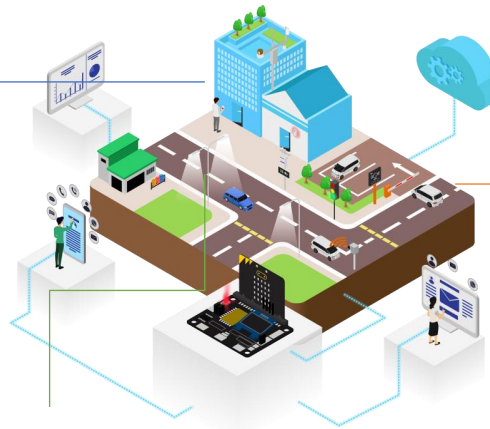
SMART MOBILITY

Enhance the travel experience by providing real time information



SMART PEOPLE

Provide training programming to nurture IT professions



WHAT IS IOT?

Internet of things (IoT) is a network connecting various objects (tangible or intangible objects, including computer devices, systems) via internet. By adding sensors, software or other technology on the object, it allows data connecting and exchange without requiring human-to-human or human-to-computer interaction. There are three ways of information sharing in IoT:

1. Collect and share information

Sensors on the things collect information (temperature, humidity, raindrop, etc) and upload to the central system for further analysis

Example: Upload data to ThingSpeak

Example2 : Send Email by IFTTT

2. Receive information and react

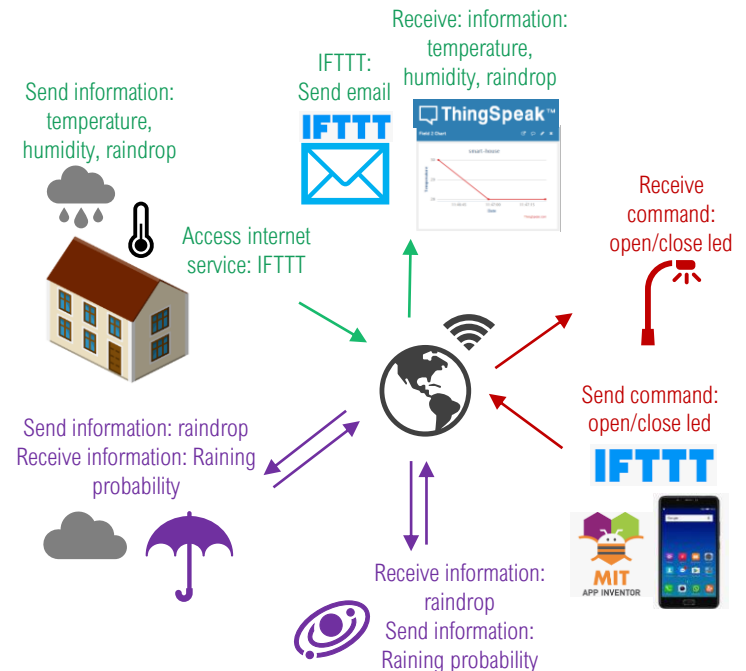
Thing that wait for the command for an action

Example: Control micro:bit by App Inventor 2/IFTTT Services

3. Doing both

Collect information and share via internet, it will react for further action

Example: Object to Object communication
A smart umbrella that collect raindrop information and send to the climate system. it will receive the raining probability from the system and then open/close the umbrella



PART LIST

SMARTHON IOT BIT:

It is an IoT orientated expansion board for Micro:bit. It can connect to different sensors or actuator



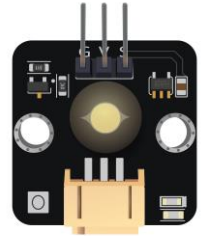
TRAFFIC LIGHT MODULE:

It includes three different colors LED which are red, yellow and green color, like the real-life traffic light



WHITE LED MODULE:

It can emit white LED light with different brightness by user input



RAINDROP SENSOR:

It uses conductivity to determine if the raindrops are on the board



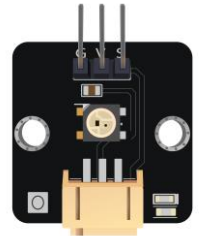
DISTANCE SENSOR:

It uses ultrasonic to measure the distance between the sensor and object



MULTI-COLOR LED (WS2812B):

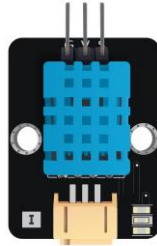
It is an integrated LED chip which can create different RGB color by programming



PART LIST

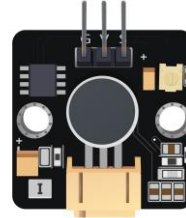
TEMPERATURE AND HUMIDITY SENSOR (DHT11):

It contains two components to detect humidity and temperature



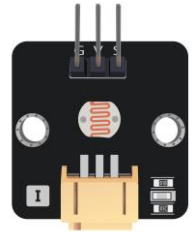
NOISE SENSOR:

use a microphone with an amplifier to detect the sound level of the environment



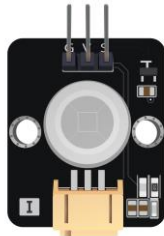
LIGHT SENSOR:

It uses one photoresistor to detect the luminance of environment



MOTION SENSOR:

It use a pyroelectric infrared sensor to detect any moving object at front



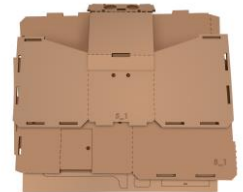
180° SERVOS:

It allows the shaft to be positioned between 0 and 180 degrees



CARDBOARD AND WOODEN MODEL:

It allows sensors/actuators to be placed on it



CITY MAP:

It is approximately 594mm*420mm, cardboard model can be placed on it



OTHERS:

(9) MODULE WIRE · (8) EXTENSION WIRE · (12) SCREW M2*10mm · (6) SCREW M3*10mm · (30) SCREW M4*10mm · (12) NUT M2 · (6) NUT M3 · (30) NUT M4 · (3) PAPER CLIP · (2) L-SHARPED STAND · (1) SCREWDRIVER · (2) SANDPAPER · (1) BLU TACK · (1) USB CABLE · (1) BATTERY HOLDER · (1) SMART CITY KIT MANUAL BOOK.

MODEL LIST

MODEL A:

It is a smaller size playground lamp with slope Light. There is a sensor port on the side (e.g. motion sensor can be attached to detect movement)



Playground Lamp

MODEL C:

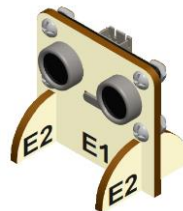
It is a garbage bin with a distance sensor to detect the amount of garbage inside. There is a port in front of the bin. You can attach multi-color to indicate the garbage level.



Garbage Bin

MODEL E:

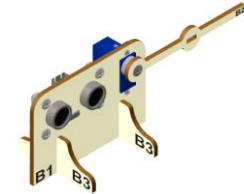
In this model, distance sensor can be attached to measure the approximate distance between the model and the target object. It can be used with Model D to display the distance/speed.



Car speed monitoring

MODEL B:

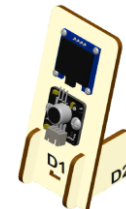
In this model, distance sensor is to detect if there are cars in front of the model, while servo control the angle of the access barrier.



Car Park Access Barrier

MODEL D:

It is an OLED stand used to display useful information. To collect information, user can attach sensors below OLED (e.g. sound sensor or temperature sensor).

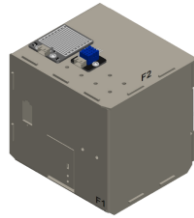


OLED Display Stand

MODEL LIST

MODEL F:

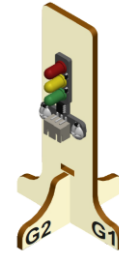
It is small building model. There are several ports on the top of the model. You can put sensors/actuators for different purpose. (e.g. rain drop sensor, temperature and humidity sensor)



Small Building
(Apartment, communal building)

MODEL G:

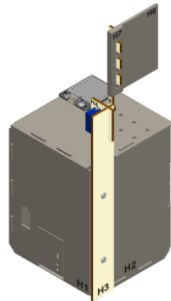
Traffic light module can be attached to this model. It is placed on the city street with 3 LED to indicate current traffic state.



Traffic Light

MODEL H:

It is a tall building model with an automatic cloth rack controlled by the servo. Sensors/actuators can be attached on top for different purpose. (e.g. rain drop sensor, temperature and humidity sensor)



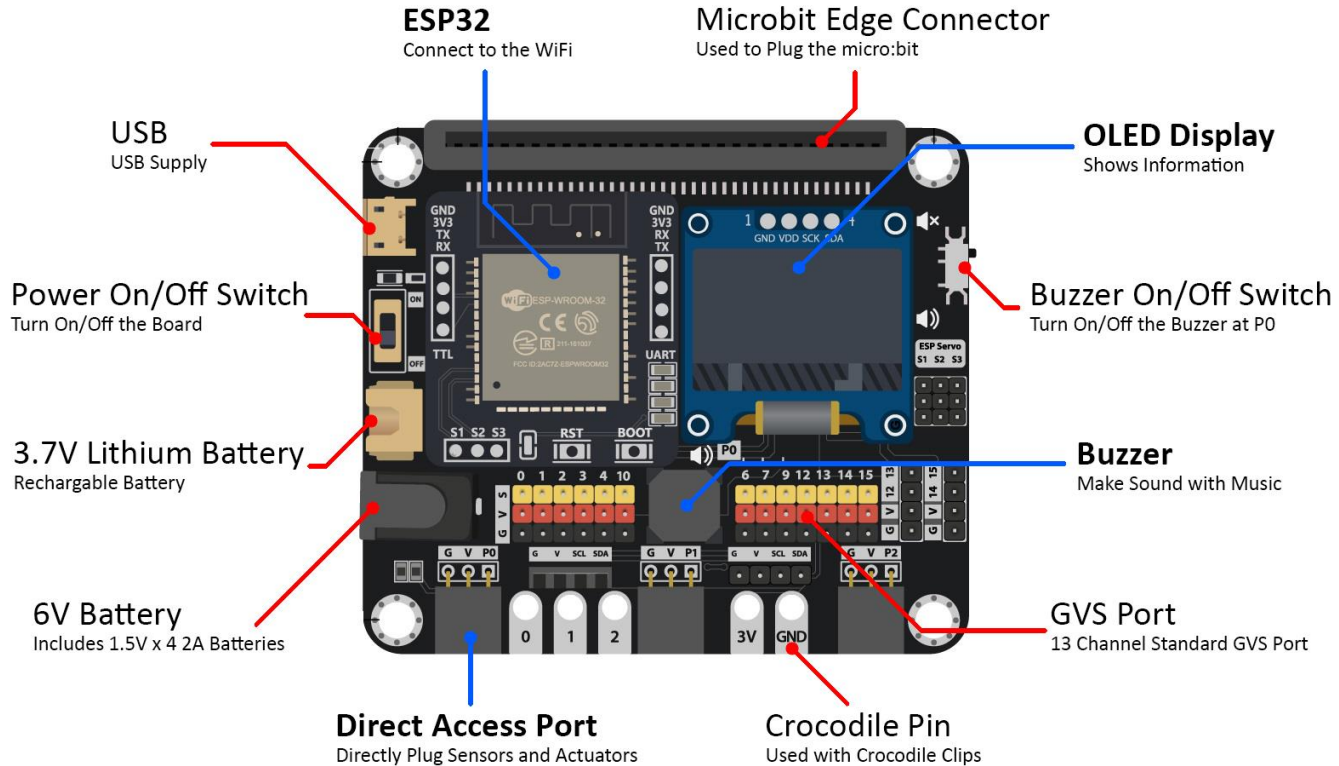
Tall Building
(Skyscrapers, high-rise housing)

MODEL I:

It is a taller sized lamp with slope light on the city street. It supports smart lighting system in the city. In your project, you can implement IoT connection in the lighting system.



Streetlight



 <https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/index.html>

POWER SUGGESTION

IoT:bit Voltage Range: 3.7V DC-6V DC. There are 3 options for the power supply

OPTION 1

- USB



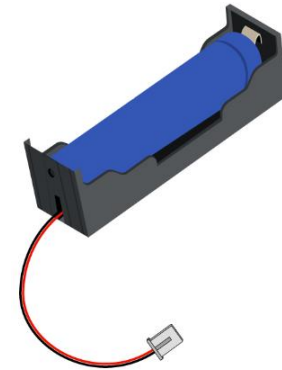
OPTION 2

- 4 XAA Batteries (Battery Not Included)



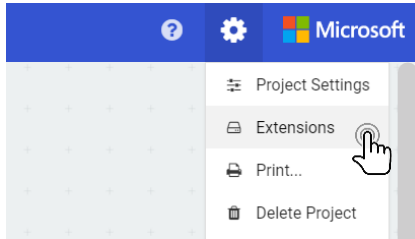
OPTION 3

-18650 3.7V Battery (Battery and holder are not included) with standard 2.54 interface

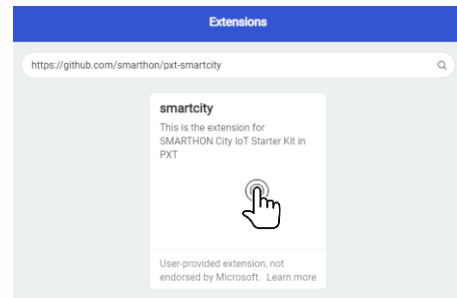


GETTING STARTED: ADD THE EXTENSION ON MICRO:BIT

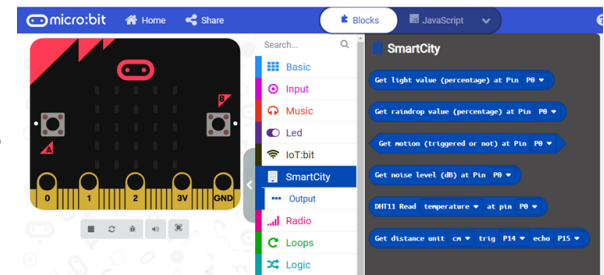
1. Go to <https://makecode.microbit.org/>,
create a new project



2. Search “<https://github.com/smarthon/pxt-smartcity>”
and select “smartcity”

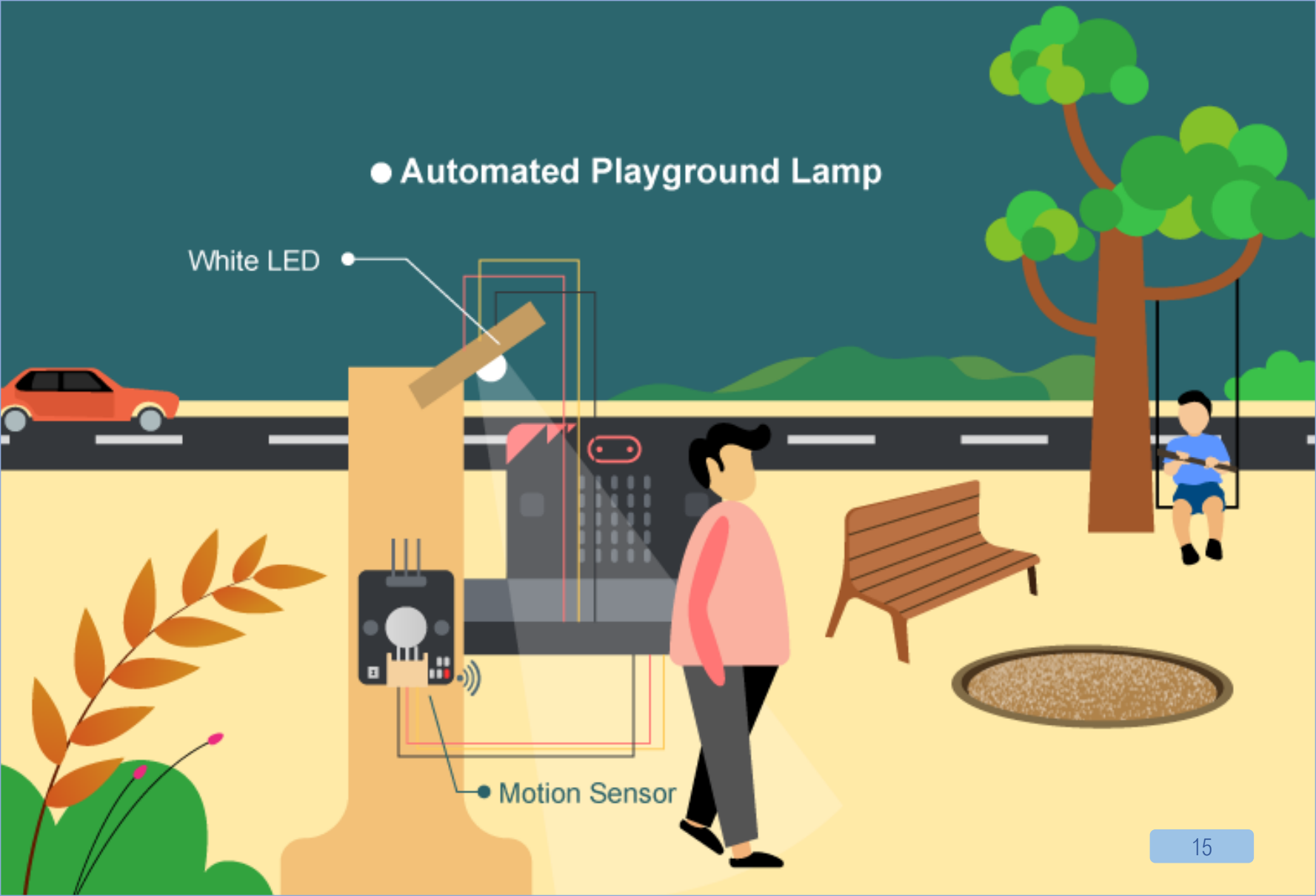


3. Once loading completed, [SmartCity](#)
will be appeared.



● Automated Playground Lamp

White LED



● Motion Sensor

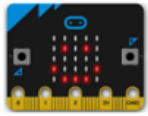
1. AUTOMATED SMART PLAYGROUND LAMP

LEVEL: ★ ☆ ☆ ☆ ☆

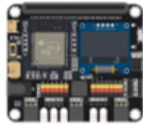
Introduction

Smart playground lamp is a lamp which can open automatically when someone passes by. Installing an auto-light can help the earth save electricity. When no one passes by, the light will automatically turn off

PART LIST



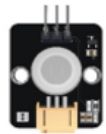
micro:bit (1)



IoT:bit (1)



White LED Light (1)



Motion Sensor (1)



3-pin module wire (2)



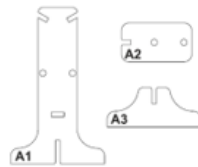
M4*10mm Screw (4)



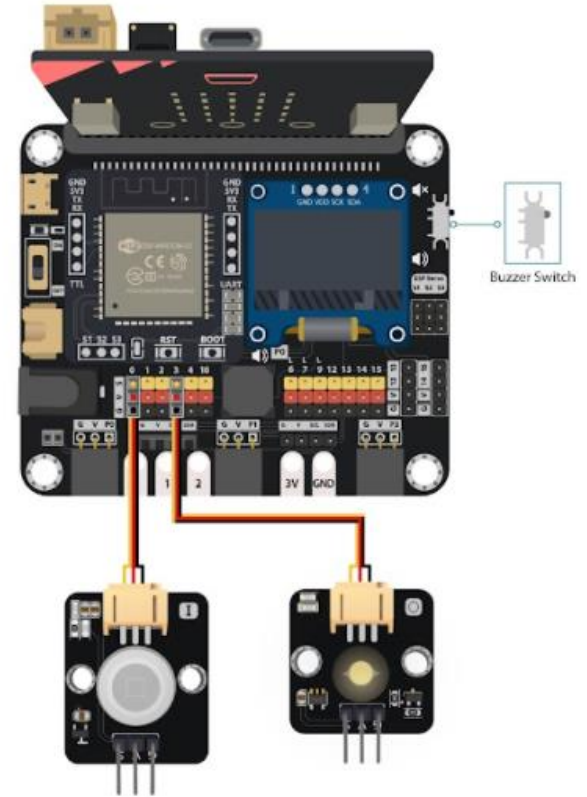
M4*10mm Nut (4)



Screwdriver (1)

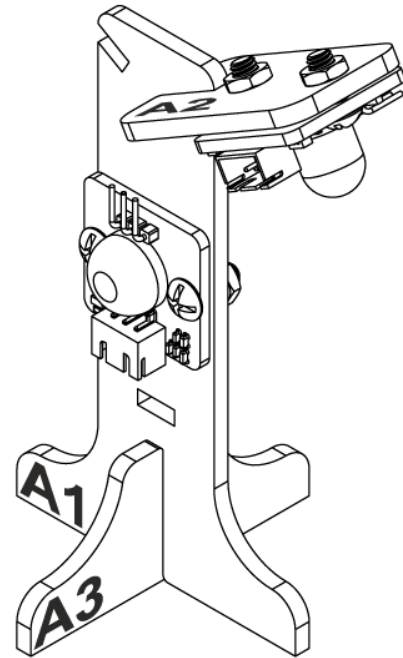
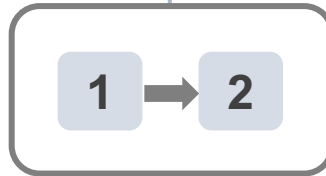
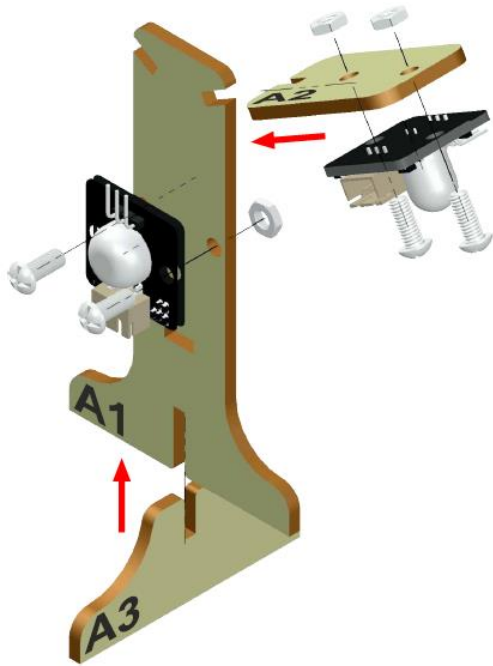


A1-A3 Model (1)



ASSEMBLY STRUCTIONS

A1-A3 model	x1
PIR Motion Sensor	x1
White LED	x1
M4 Nut	x4
M4 x 10mm Screw	x4




CODING



Other than turning on the light, how can you use with motion sensor? (e.g. people counting)

1 on start

2 led enable false



3 forever

4 if Get motion (triggered or not) at Pin P0 = true then

5 Turn White LED to 1023 at P3

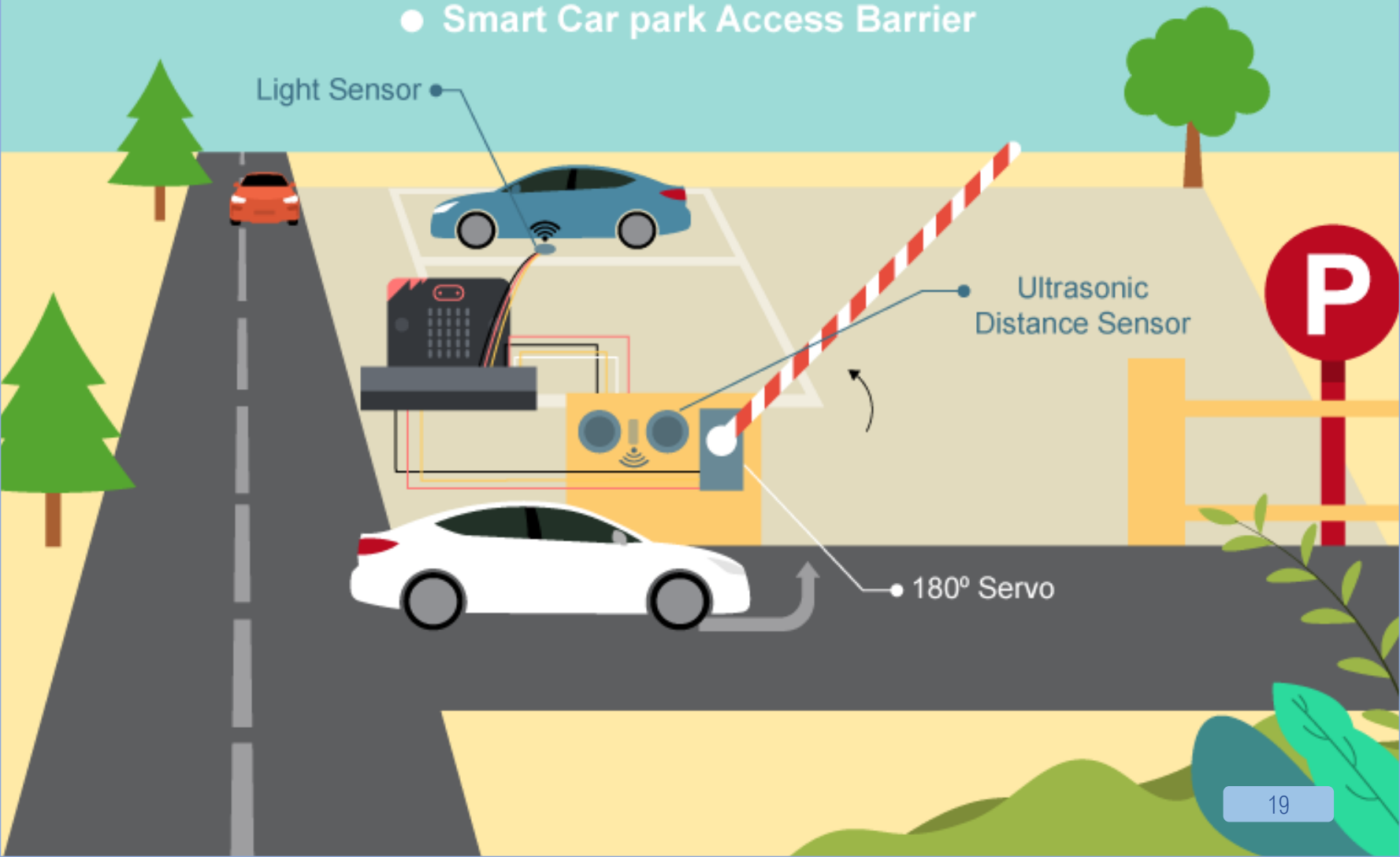
6 pause (ms) 10000

7 else Turn White LED to 0 at P3

STEPS:

1. Block **on start** only runs once at the beginning.
2. **Led enable** is **false** (as P3 is used for micro:bit LED by default, LED need to be disable before using P3)
3. Block **forever** runs as a loop.
4. If **motion sensor** triggered (i.e. someone passes by) is **true**,
5. **White LED light at P3** is turned on.
6. It pauses **10 seconds** and the light will up for 10 seconds.
7. **White LED light at P3** is set to be turned off.

● Smart Car park Access Barrier



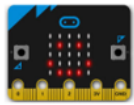
2. SMART CAR PARK ACCESS BARRIER

LEVEL: ★ ★ ☆ ☆ ☆

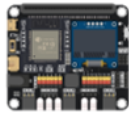
Introduction

Smart Car park Access Barrier is used to allow people to live conveniently. It can reduce manpower and time in controlling the gate and manage information (e.g. car park vacancies). It will be opened automatically if there are vacancies in the car park and there are cars coming in.

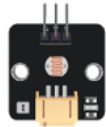
PART LIST



micro:bit (1)



IoT:bit (1)



Light Sensor (1)



Distance Sensor (1)



180° Servo (1)



3-pin module wire (1)



4-pin module wire (1)



Servo M2*7.5mm sharp screw (1) (Included in servo pack)



M2*10mm Screw (2)



M2 Nut (2)



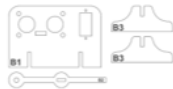
M4*10mm Screw (4)



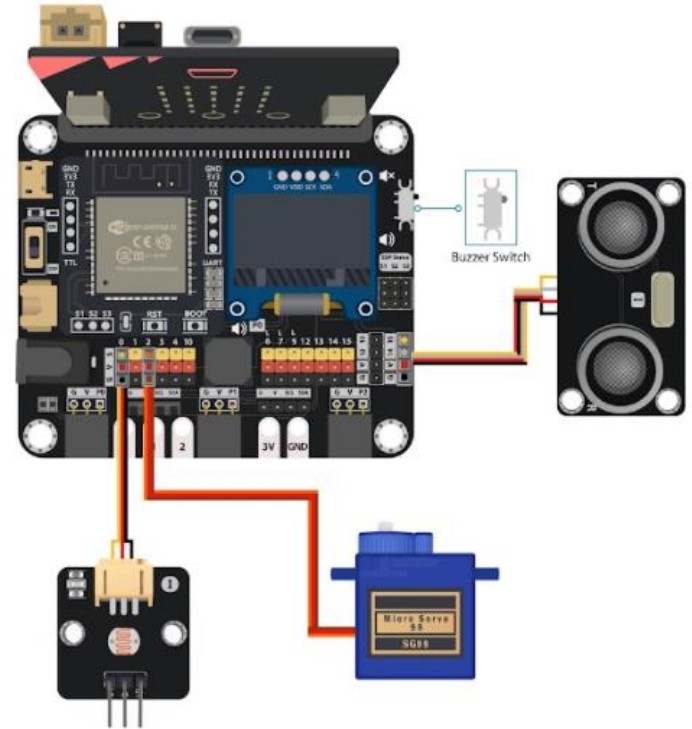
M4 Nut (4)



Screwdriver (1)

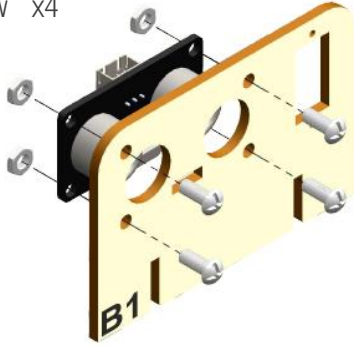


B1-B3 Model (1)

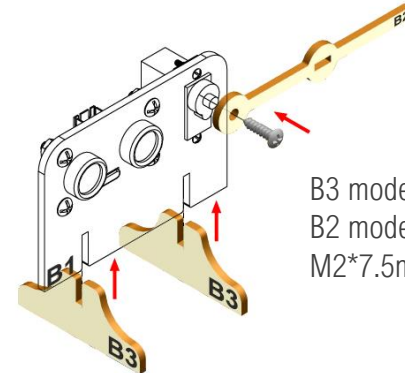
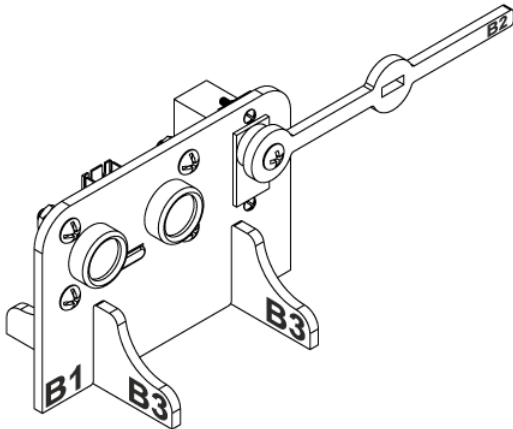
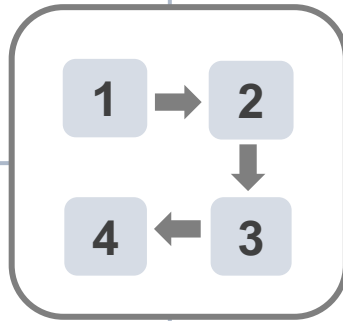
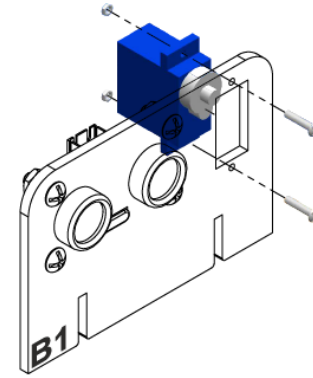


ASSEMBLY STRUCTIONS

B1 model x1
 Distance Sensor x1
 M4 Nut x4
 M4*10mm Screw x4



180° servo X1
 M2 Nut X2
 M2*10mm Screw X2



B3 model X2
 B2 model X1
 M2*7.5mm Sharp Screw X1

CODING

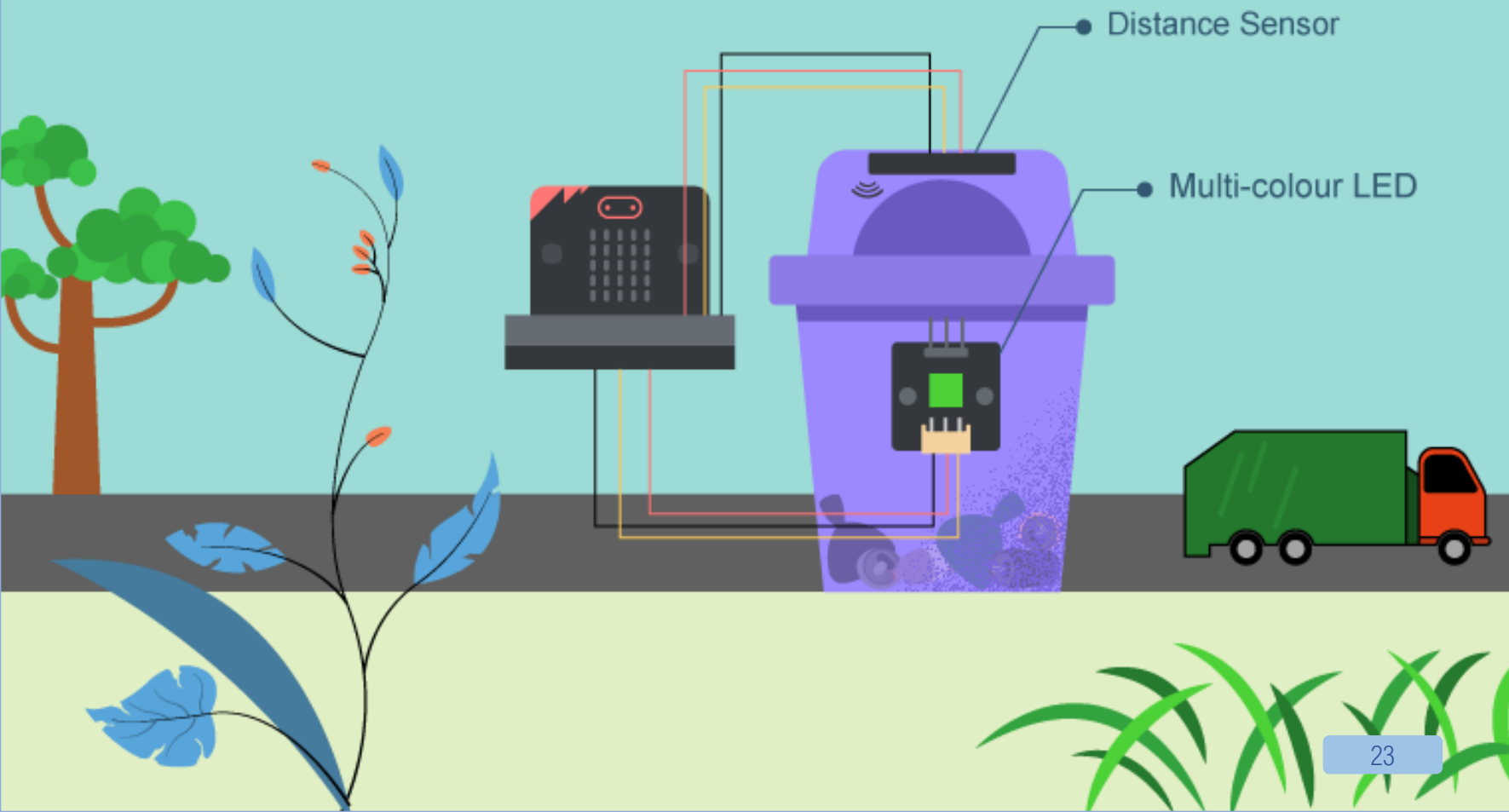
💡 How can you show the light and distance value on OLED?

```
1 on start
2 set distance to 0
  set light to 0
  Turn Servo to 0 degree at P2
3 forever
4 set distance to Get distance unit cm trig P14 echo P15
5 set light to Get light value (percentage) at Pin P0
6 if distance <= 5 then
7   if light > 20 then
8     Turn Servo to 90 degree at P2
     pause (ms) 5000
     Turn Servo to 0 degree at P2
```

STEPS:

1. Block **on start** only runs once at the beginning.
2. Set **distance** and **light** to 0 and **turn servo at P2 to 0 degree**.
3. Block **forever** runs as a loop.
4. Set **distance** to get value from the distance sensor connected to P14 and P15
5. Set **light** to get light value from the light sensor at P0.
6. If there is car coming near (**distance \leq 5**) then,
7. If there is vacancy inside the car part (**light value $>$ 20**),
8. Car park gate is opened (i.e. **servo is turned for 90°**). It **pauses 5 seconds** and the gate will open for 5 seconds. Then the car park gate is closed (i.e. **servo is returned to 0°**).

● Garbage Disposal System



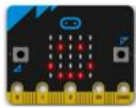
3. GARBAGE DISPOSAL SYSTEM

LEVEL: ★ ★ ☆ ☆ ☆

Introduction

LED light on the garbage bin can show people the amount of garbage inside the garbage bin so that garbage truck can easily determine if the garbage is full or not. This can minimize the wastage of the garbage bags and become a more environmental-friendly city.

PART LIST



micro:bit (1)



IoT:bit (1)



Multi-colour LED (WS2812) (1)



Distance Sensor (1)



3-pin module wire (1)



4-pin module wire (1)



M4*10mm Screw (6)



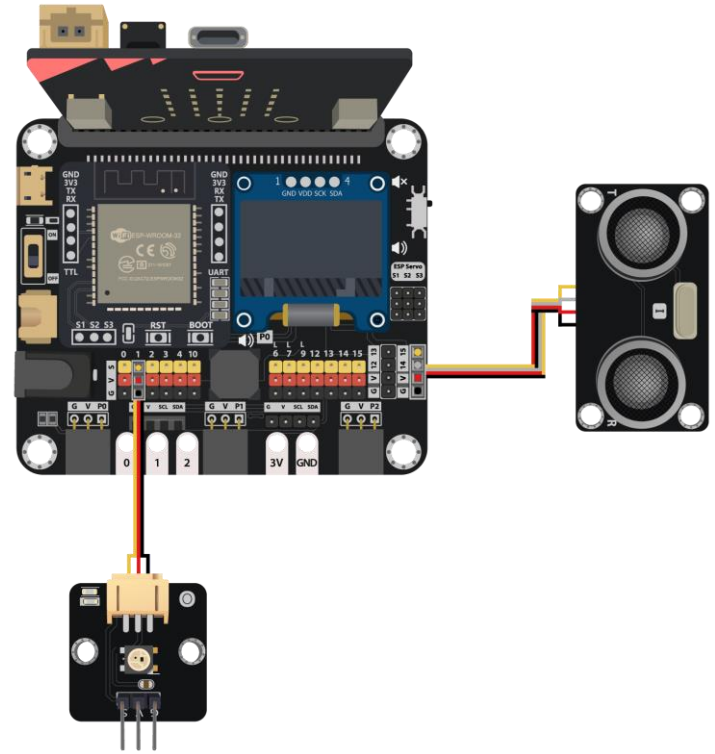
M4 Nut (6)



Screwdriver (1)

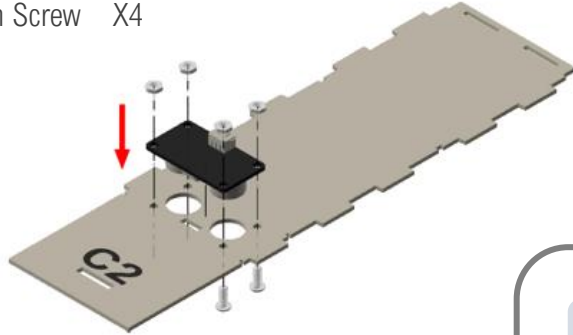


C1-C2 Model (1)

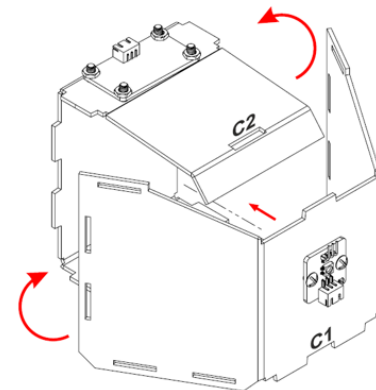
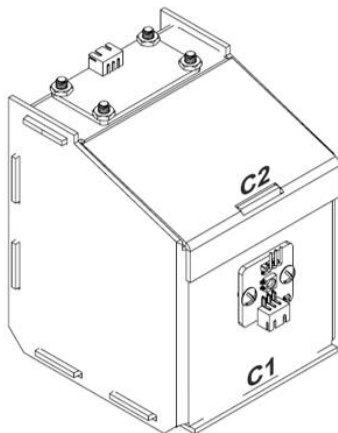
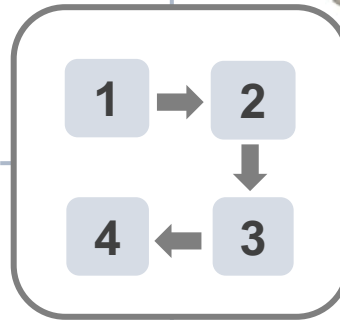
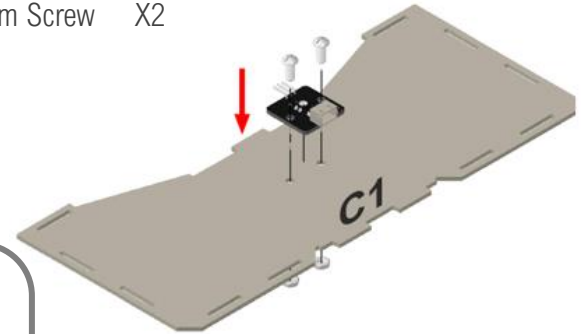


ASSEMBLY STRUCTIONS

C2 model X1
Distance sensor X1
M4 Nut X4
M4*10mm Screw X4



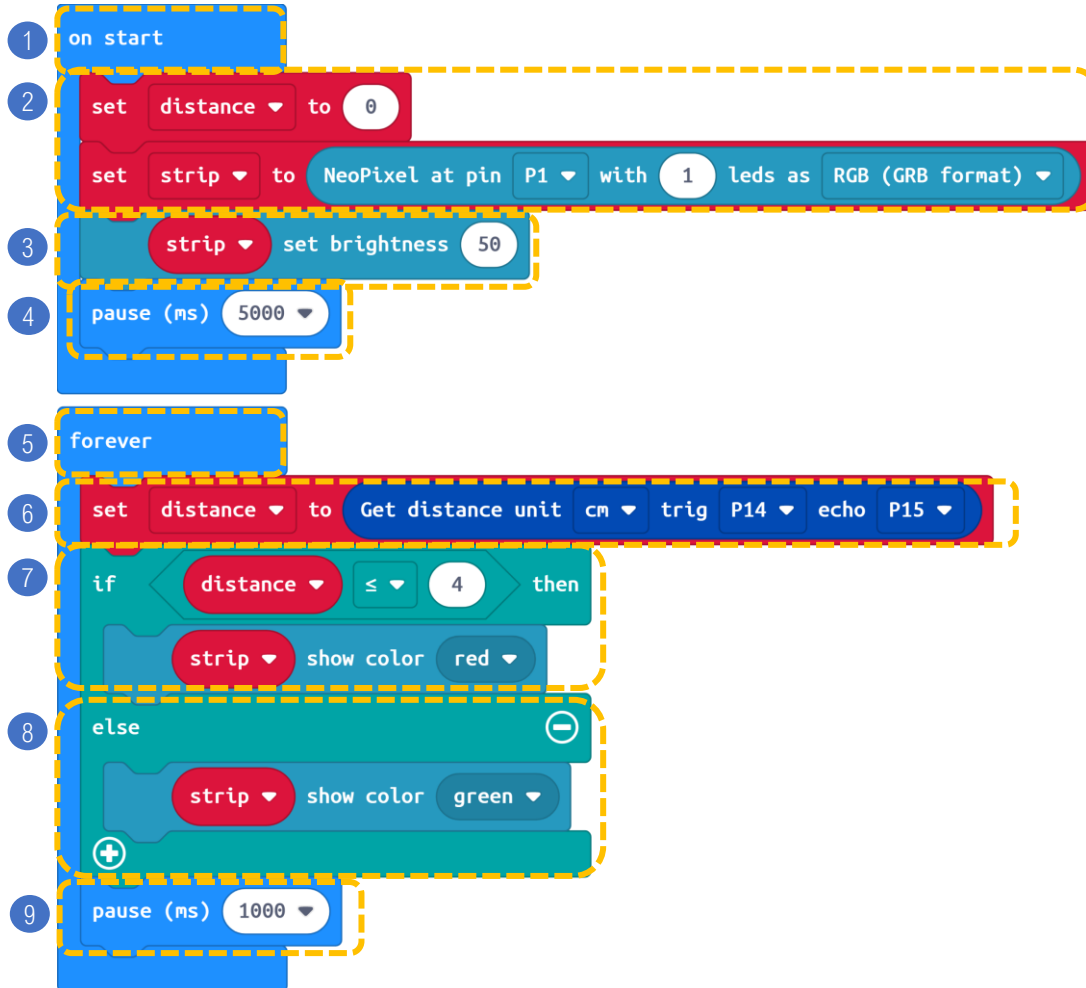
C1 model X1
Multi-colour LED X1
M4 Nut X2
M4*10mm Screw X2



CODING



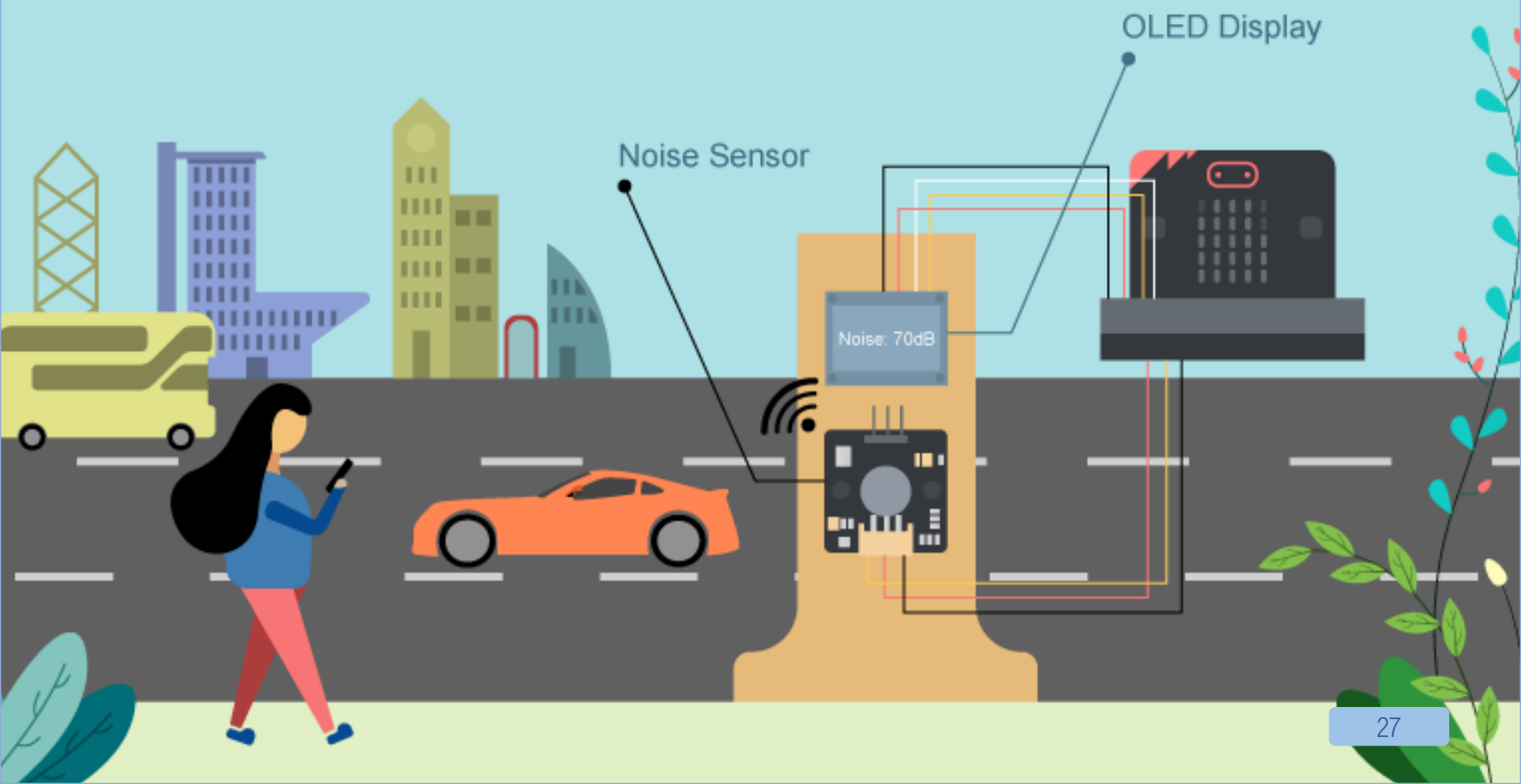
How to make sound notification if there are full of garbage (e.g. using buzzer)?



STEPS:

1. Block **on start** only runs once at the beginning.
2. **Set distance to 0** and **set strip to NeoPixel at pin P1 with 1 leds as RGB (GRB format)**
3. **Strip set brightness to 50**
4. **Pause for 5 seconds.**
5. Block **forever** runs as a loop.
6. **Set distance to get distance unit cm trig P14 echo P15.**
7. If **distance** \leq 4 then, strip show colour **red**.
8. Else, strip show colour **green**.
9. **Pause 1 second** for the corresponding colour lighting up.

● Urban noise detection



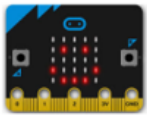
4. URBAN NOISE DETECTION

LEVEL: ★ ★ ☆ ☆ ☆

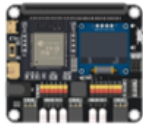
Introduction

It is a system to detect noise near the road as noise pollution caused by cars on the road seriously affect the living standard of people. By installing a monitor to detect the noise volume near the roadside can help engineer to gather noise information and find solution to solve the problem in the future.

PART LIST



micro:bit (1)



IoT:bit with OLED (1)



Noise Sensor (1)



3-pin module wire (1)



4-pin extension wire (1)



M2*10mm Screw (2)



M2 Nut (2)



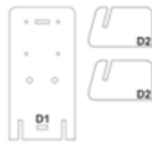
M4*10mm Screw (4)



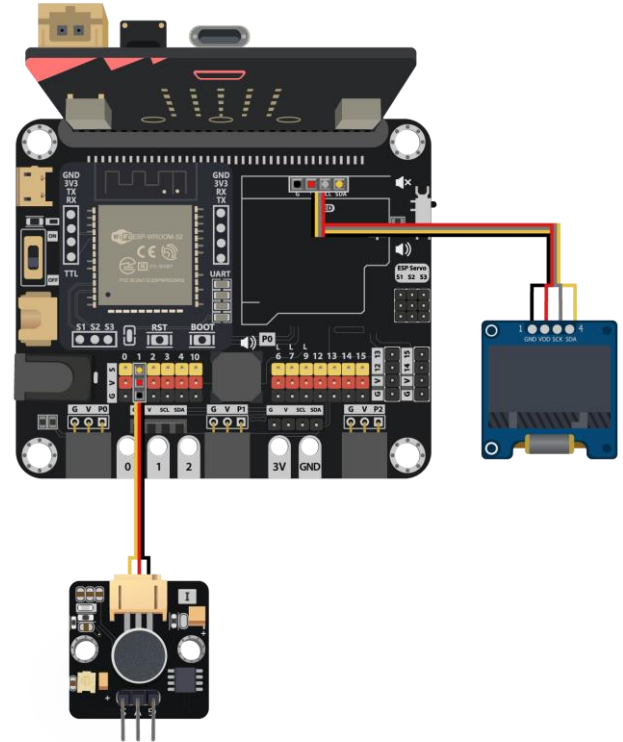
M4 Nut (4)



Screwdriver (1)

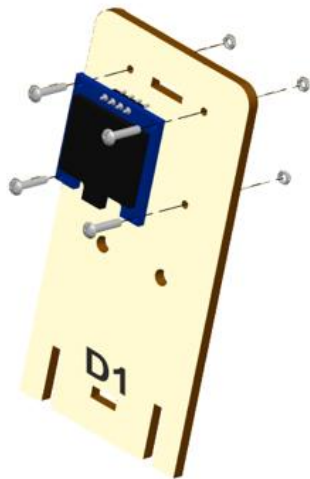


D1-D2 Model (1)

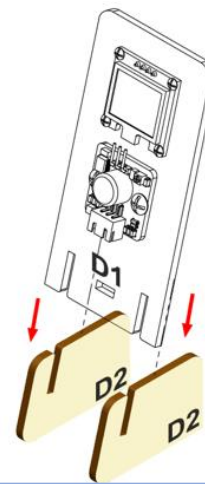
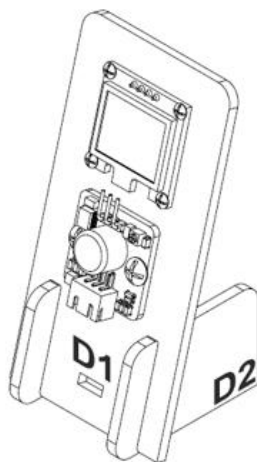
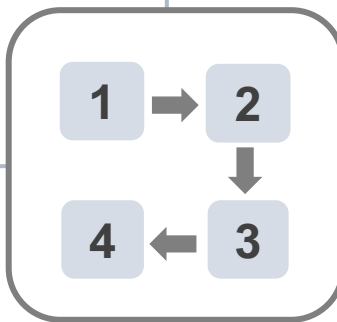
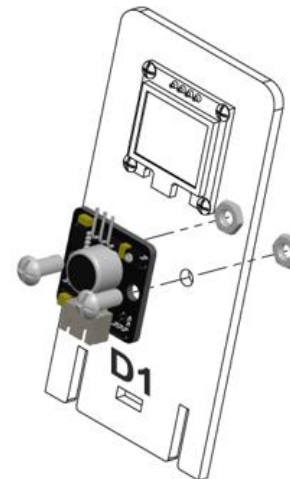


ASSEMBLY STRUCTIONS

- D1 model X1
- OLED X1
- M2 Nut X4
- M2*10mm screw X4



- Noise sensor X1
- M4 Nut X2
- M4*10mm screw X2

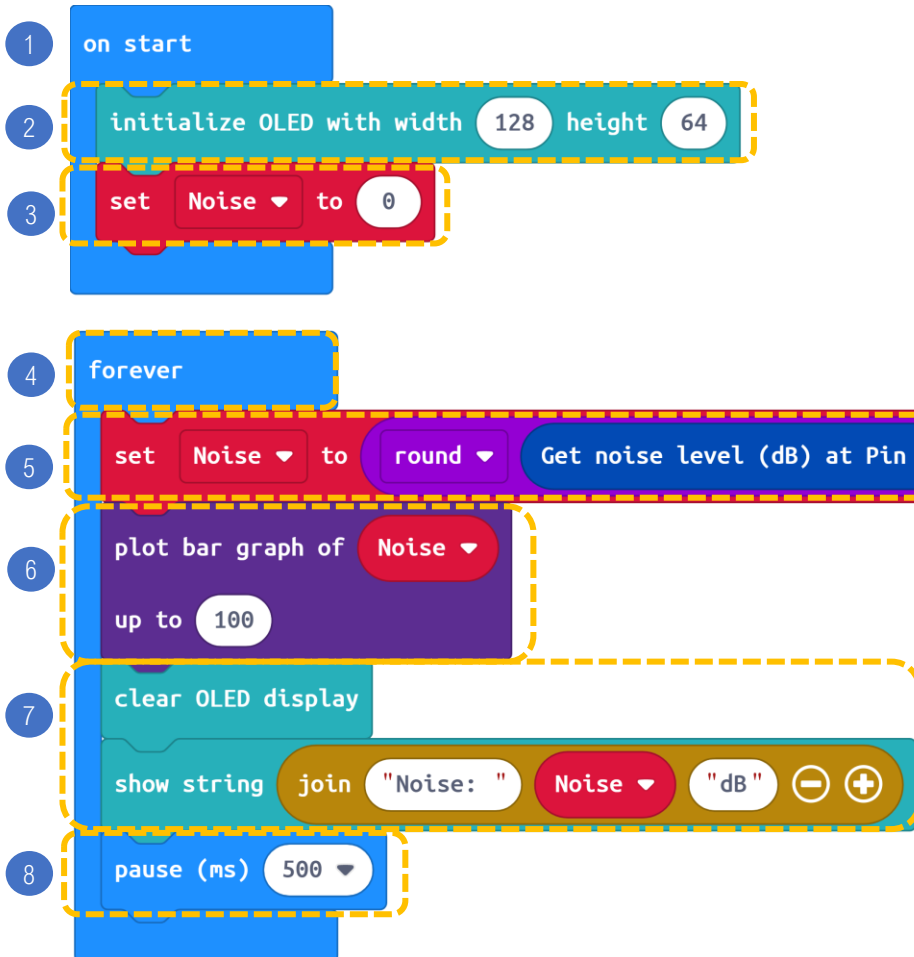


- D2 model X2

CODING



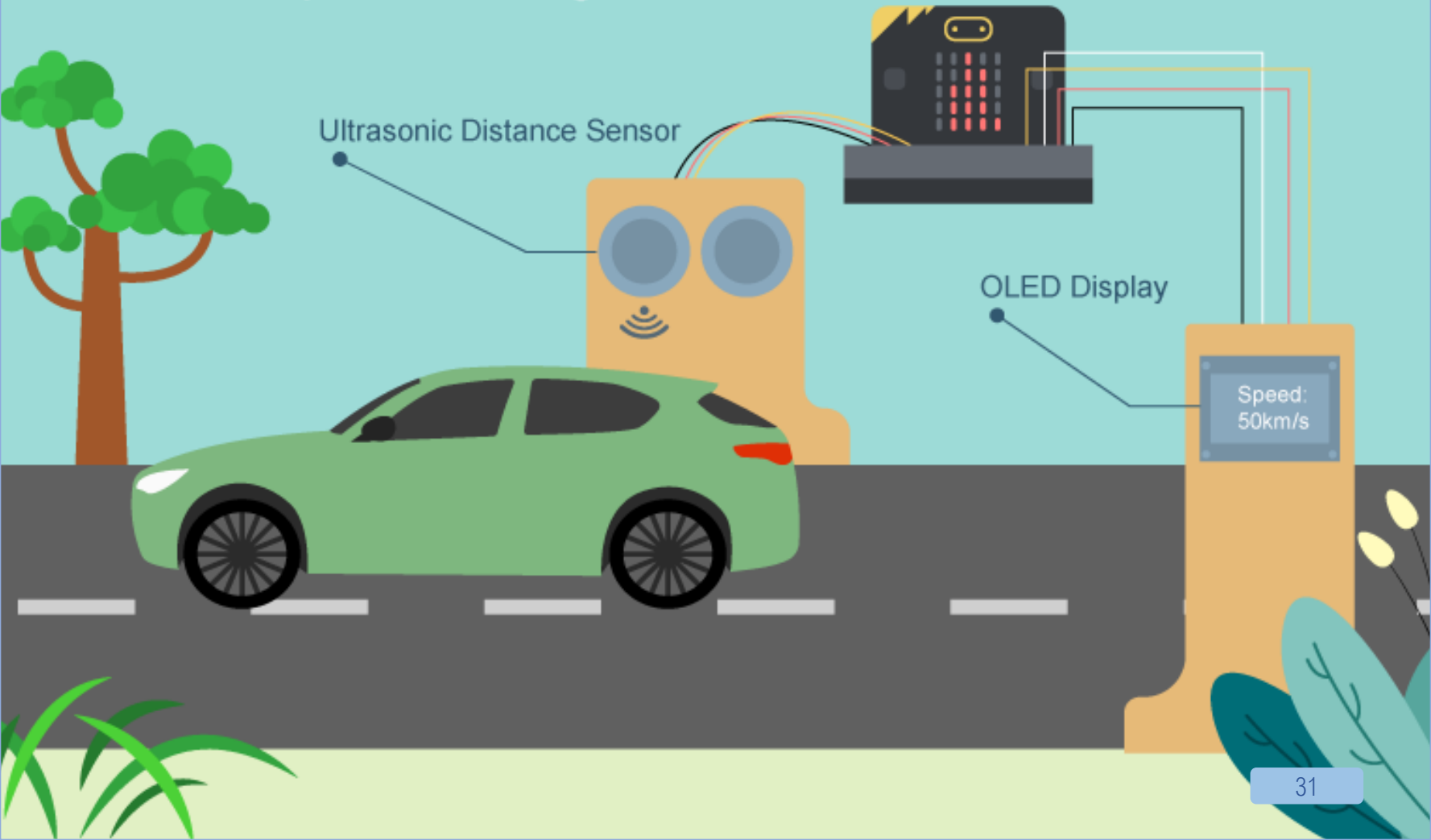
How to make a notification if noise pollution problem is serious? e.g. showing red LED



STEPS:

1. Block **on start** only runs once at the beginning.
2. Initialize OLED with width:128, height: 64
3. Set **noise** to 0
4. Block **forever** runs as a loop.
5. Set **noise** to **round get noise level (dB) at Pin P1**
6. Plot bar graph of **noise** on the micro:bit LED.
7. Clear OLED display, then show string text "Noise:" & noise level & text "dB:" on the OLED.
8. Pause 0.5 second for the corresponding noise bar graph shown.

● Car Speed Monitoring



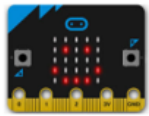
5. CAR SPEED MONITORING

LEVEL: ★ ★ ★ ☆ ☆

Introduction

It is an automatic system to check car speed on the road at certain time interval. There are cars often over-speed causing traffic accidents, therefore installing a car speed monitoring is a must to minimize the chances of traffic accidents.

PART LIST



micro:bit (1)



IoT:bit with OLED (1)



Distance Sensor (1)



4-pin module wire (1)



4-pin extension wire (1)



M2*10mm Screw (4)



M2 Nut (4)



M4*10mm Screw (4)



M4 Nut (4)



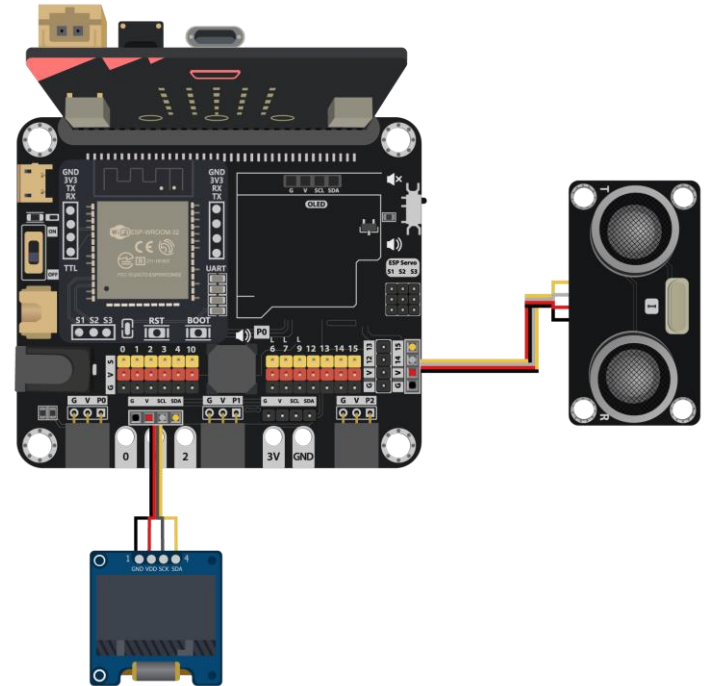
Screwdriver (1)



D1-D2 Model (1)

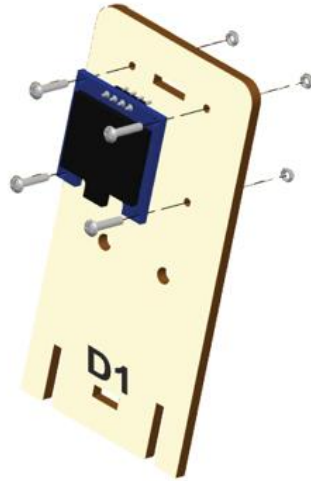


E1-E2 Model (1)

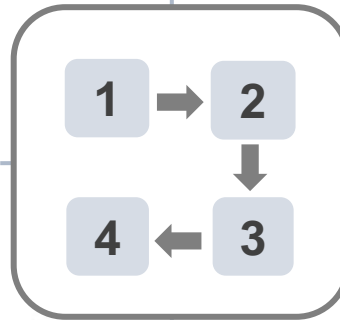
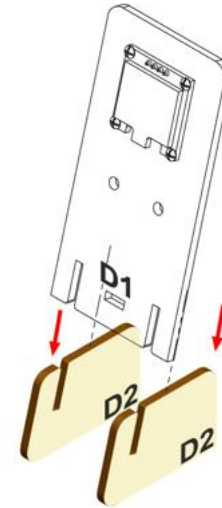


ASSEMBLY STRUCTIONS

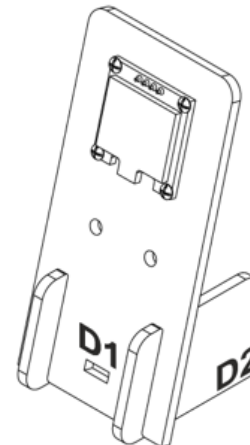
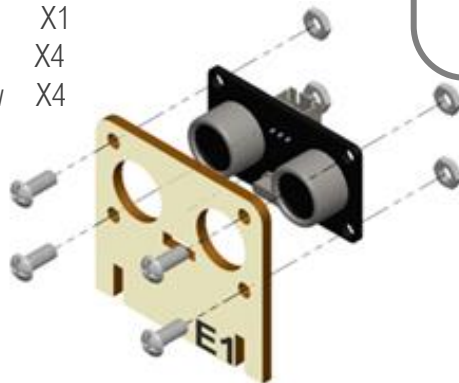
- D1 model X1
- OLED X1
- M2 Nut X4
- M2 *10mm screw X4



- D2 model X2

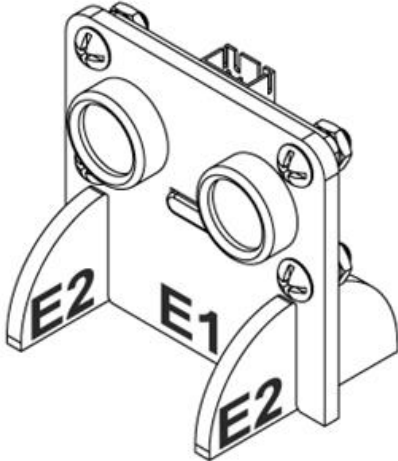
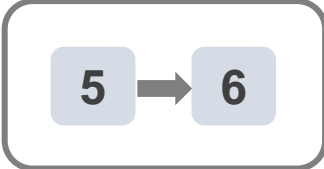
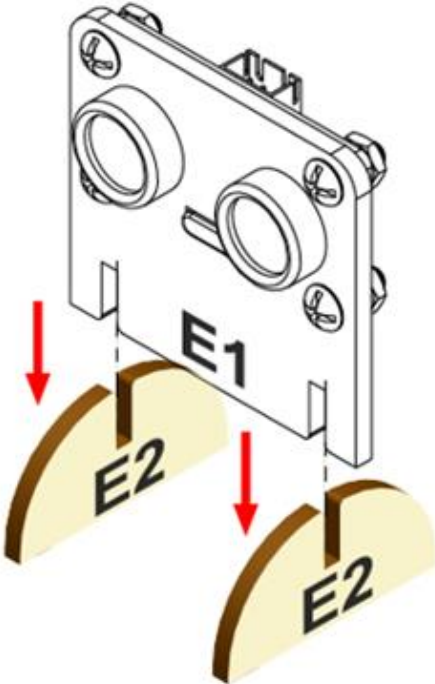


- E1 model X1
- Distance sensor X1
- M4 Nut X4
- M4 *10mm Screw X4



ASSEMBLY INSTRUCTIONS

E2 model X2



CODING



How can we set a sound alert to notify that there is car over-speeding?

```
1 on start
2 initialize OLED with width 128 height 64
3 set distance1 to 0
  set distance2 to 0
  set speed to 0
4 forever
5 call calculate_Speed
6 if speed >= 0 then
7   plot bar graph of speed
   up to 20
8 clear OLED display
9 show string join "Distance1: " distance1
  show string join "Distance2: " distance2
  show string join "Speed: " speed
```

STEPS:

1. Block **on start** only runs once at the beginning.
2. **Initialize OLED with width:128, height:64**
3. Set **distance 1**, **distance 2** and **speed** to 0.
4. Block **forever** runs as a loop.
5. Call function **calculateSpeed** (it will calculate distance1, distance2 and the speed)
6. If there is car coming near (**speed \geq 0**) then,
7. **Plot bar graph of speed** on the micro:bit LED.
8. **Clear OLED display**
9. **Show string text "Distance1:" & distance 1, text "Distance2:" & distance2 and text "Speed:" & speed** on the OLED.

CODING



How can we set a sound alert to notify that there is car over-speeding?

The image shows a Scratch script with four numbered steps:

- 10. A function block named `calculate_Speed`.
- 11. A `set distance1` block with a dropdown menu, followed by a `Get distance unit` block with dropdowns for `cm`, `trig`, `P14`, and `echo P15`.
- 12. A `pause (ms)` block with a dropdown menu set to `500`.
- 13. A `set distance2` block with a dropdown menu, followed by a `Get distance unit` block with dropdowns for `cm`, `trig`, `P14`, and `echo P15`.
- 14. A `set speed` block with a dropdown menu, followed by a `distance1` dropdown, a `-` operator dropdown, a `distance2` dropdown, a `÷` operator dropdown, and a `0.5` input field.

STEPS:

10. Set function (`calculate_Speed`)
11. Set **distance 1** to get **distance** from the distance sensor connected to P14 and P15.
12. **Pause 0.5 second** for the checking the next distance
13. Set **distance 2** to get **distance** from the distance sensor connected to P14 and P15.
14. Set **speed** to $(\text{distance1} - \text{distance2}) / 0.5$.

● Weather Station



Temperature and Humidity Sensor

Raindrop Sensor



6. [IOT] WEATHER STATION

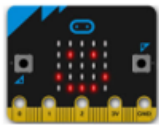
LEVEL: ★ ★ ☆ ☆ ☆



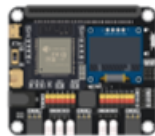
Introduction

Collecting temperature, humidity and raindrop consistently and uploading the data by using Thingspeak. This can help us to do analytical work more conveniently as we can refer to the automatically plotted graphs.

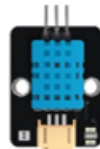
PART LIST



micro:bit (1)



IoT:bit with OLED (1)



Temperature and humidity Sensor (1)



Raindrop Sensor (1)



3-pin module wire (2)



M4*10mm Screw (4)



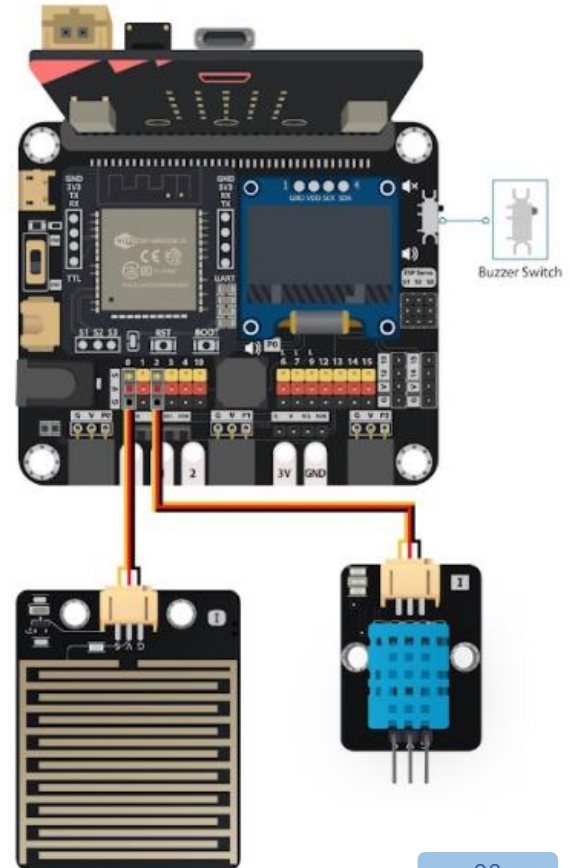
M4 Nut (4)



Screwdriver (1)



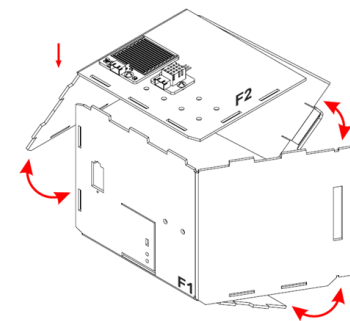
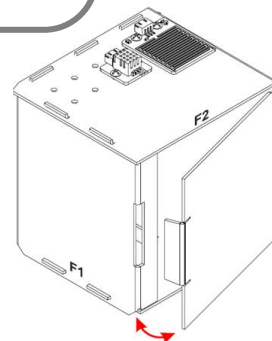
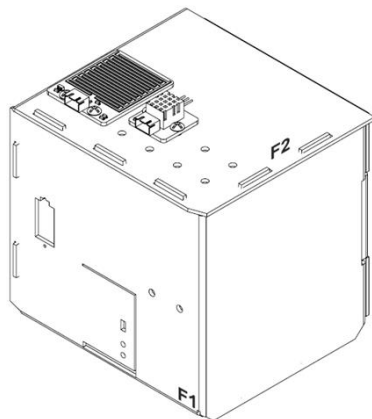
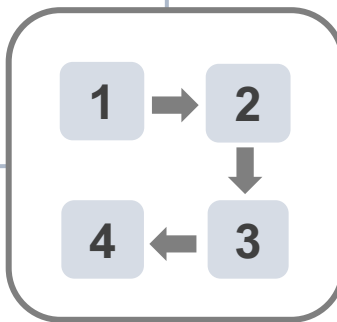
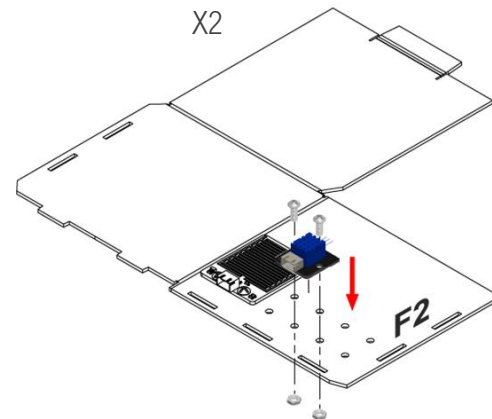
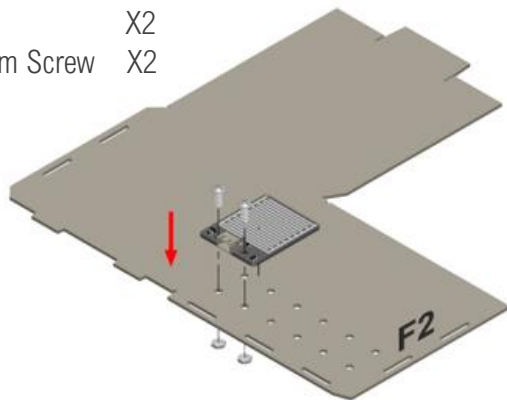
F1-F2 Model (1)



ASSEMBLY STRUCTIONS

F2 model X1
Raindrop sensor X1
M4 Nut X2
M4*10mm Screw X2

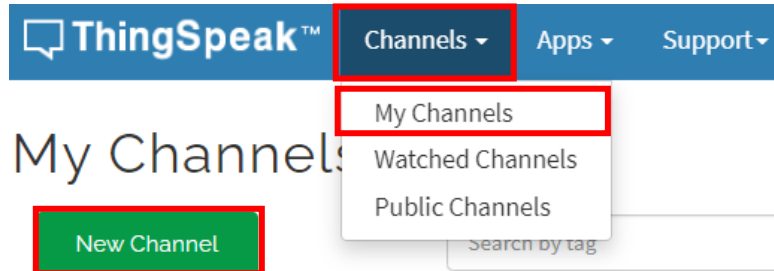
Temperature and humidity sensor X1
M4 Nut X2
M4*10mm Screw X2



F1 model X1

IOT – THINGSPEAK

Step 1. Go to <https://thingspeak.com/> , Choose Channels -> My Channels -> New Channel



Step 2. Input Channel name, Field1 and Field2 , then click “Save Channel”

- Channel name: Smart Weather Station
- Field 1: temperature
- Field 2: humidity
- Field 3: raindrop



<https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/chapter1.html>

IOT – THINGSPEAK

Step 3. Select your channel > “API Keys”, copy the write API key as follows:

The screenshot shows the Thingspeak interface. At the top, there are navigation tabs: 'Private View', 'Public View', 'Channel Settings', 'Sharing', and 'API Keys'. The 'API Keys' tab is highlighted with a red box. To the right of the tabs is a yellow button labeled '1. Select API Keys'. Below the tabs, the 'Write API Key' section is visible, with the key '5G4I...' highlighted in a red box. Below the key is a button labeled 'Generate New Write API Key'. To the right, the 'Help' section is partially visible, with a yellow button labeled '2. Copy this key'.



<https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/chapter1.html>

CODING



How can we upload other module values (e.g. noise) to ThingSpeak?

1 on start

2 initialize OLED with width 128 height 64

3 Initialize IoT:bit TX P16 RX P8

4 Set WiFi to ssid "smarthon" pwd "12345678"

5 set raindrop to 0

set humidity to 0

set temperature to 0

6 On WiFi connected IP_Address Device_ID

7 show icon

STEPS:

1. Block **on start** only runs once at the beginning.
2. **Initialize OLED with width:128, height:64**
3. Initialize IoT:bit at P16 and P8
4. Set WiFi connection by entering WiFi name and password.
5. Set **temperature, humidity** and **raindrop** to 0.
6. When WiFi is connected
7. Micro:bit LED will **show icon "tick"**

CODING



How can we upload other module values (e.g. noise) to ThingSpeak?

```
8 forever
9 if WiFi connected? then
10 set humidity to DHT11 Read humidity at pin P2
   set temperature to DHT11 Read temperature at pin P2
11 set raindrop to Get raindrop value (percentage) at Pin P0
12 clear OLED display
13 show string join "Temperature:" temperature
   show string join "Humidity:" humidity
   show string join "Raindrop:" raindrop
14 Send Thingspeak key* "5G4...89V"
   field1 value temperature
   field2 value humidity
   field3 value raindrop
15 pause (ms) 15000
```

Paste Thingspeak key here

STEPS:

- Block **forever** runs as a loop.
- Set if **WiFi is connected** then
- Set **humidity** and **temperature** to get reading from the DHT11 connected to P2.
- Set **raindrop** to get reading from the raindrop sensor connected to P0.
- Clear **OLED display**
- Show string text "Temperature" & **temperature**, text "Humidity" & **humidity** and text "Raindrop:" & **raindrop** on the OLED.
- Send data to **Thingspeak**.
- Pause **15 seconds**.

CODING



How can we upload other module values (e.g. noise) to ThingSpeak?

```
16 On Thingspeak Uploaded Status Error_code
17 clear OLED display
18 show string join "ThingSpeak:" Status
   show string join "Error:" Error_code
```

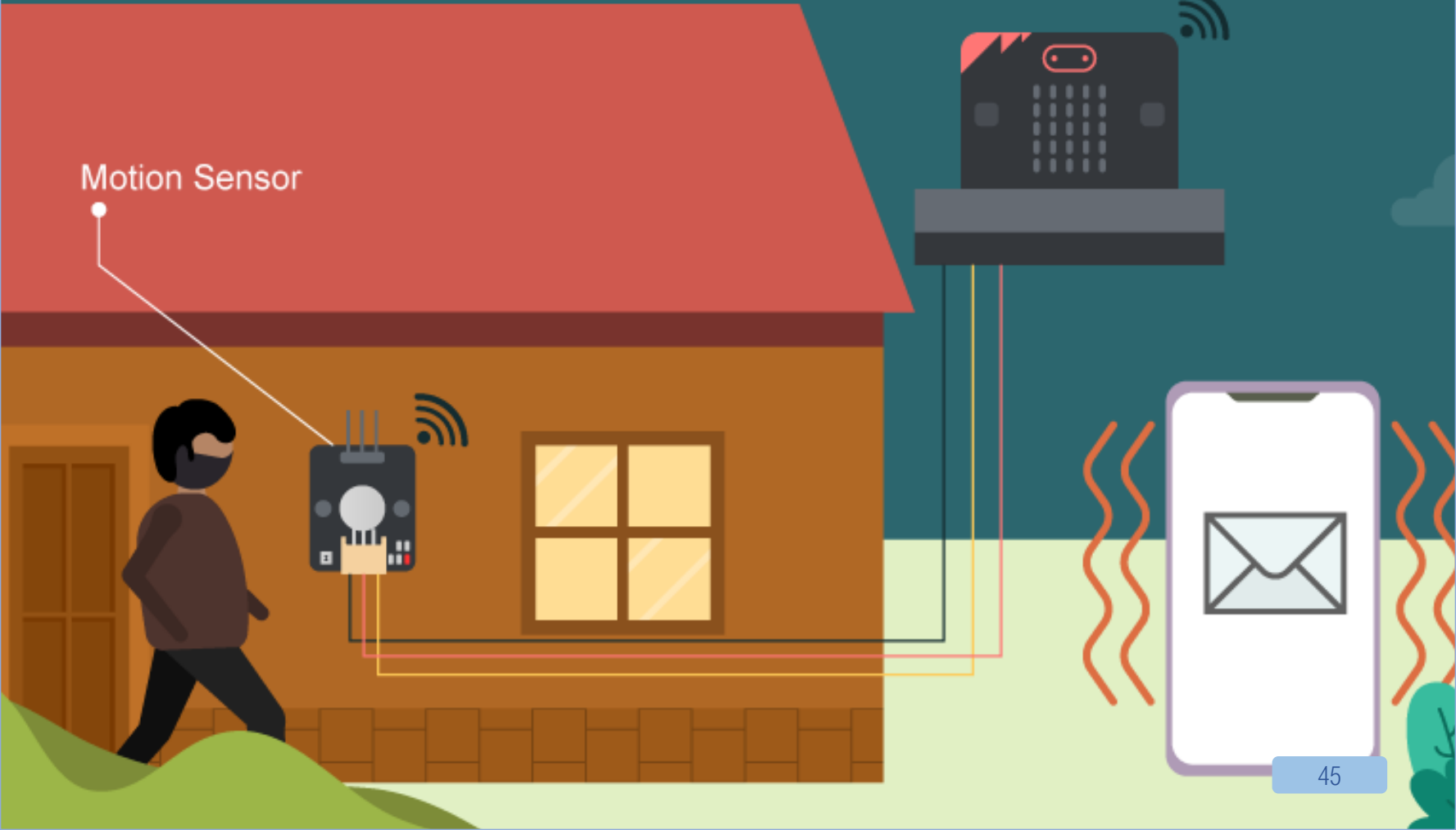
STEPS:

16. When Thingspeak uploaded
17. Clear OLED display
18. Show string text "ThingSpeak:" & status and text "Error:" & error code on the OLED.

● Smart defense system

IFTTT

Motion Sensor



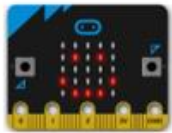
7. [IOT] SMART DEFENSE SYSTEM

LEVEL: ★ ★ ★ ☆ ☆

Introduction

The motion sensor can deliver a motion signal to the micro:bit. When the micro:bit detects the signal, the buzzer will emit sound and send an email to specific email account through IFTTT. Also, a monster icon will be shown on the micro:bit if there are suspicious people passes by.

PART LIST



micro:bit (1)



IoT:bit with OLED (1)



Motion Sensor (1)



3-pin module wire (1)



M4*10mm Screw (2)



M4 Nut (2)

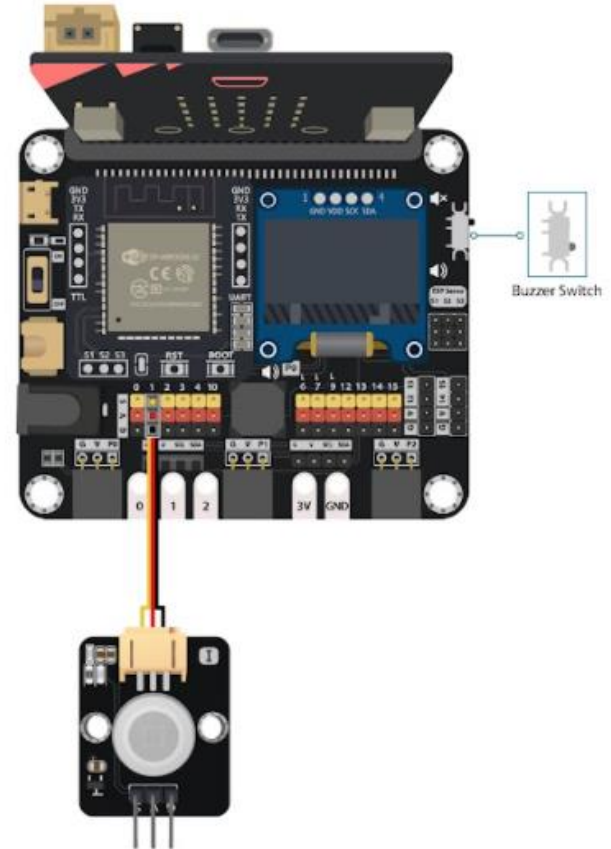


Screwdriver (1)



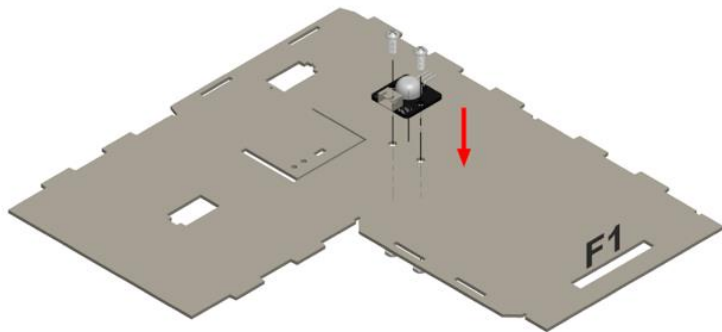
F1-F2 Model (1)

IFTTT

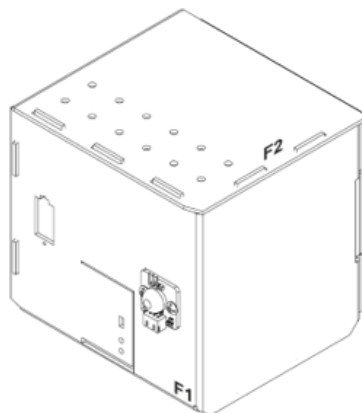
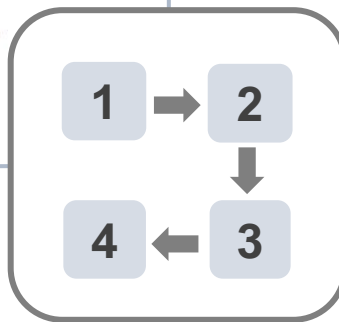
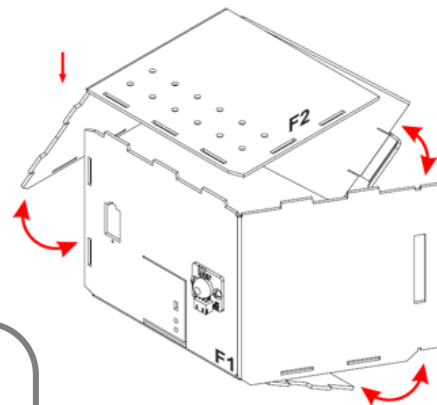


ASSEMBLY STRUCTIONS

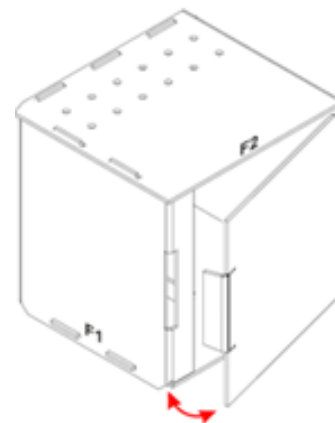
F1 model X1
Motion sensor X1
M4 Nut X2
M4*10mm Screw X2



F2 model X1

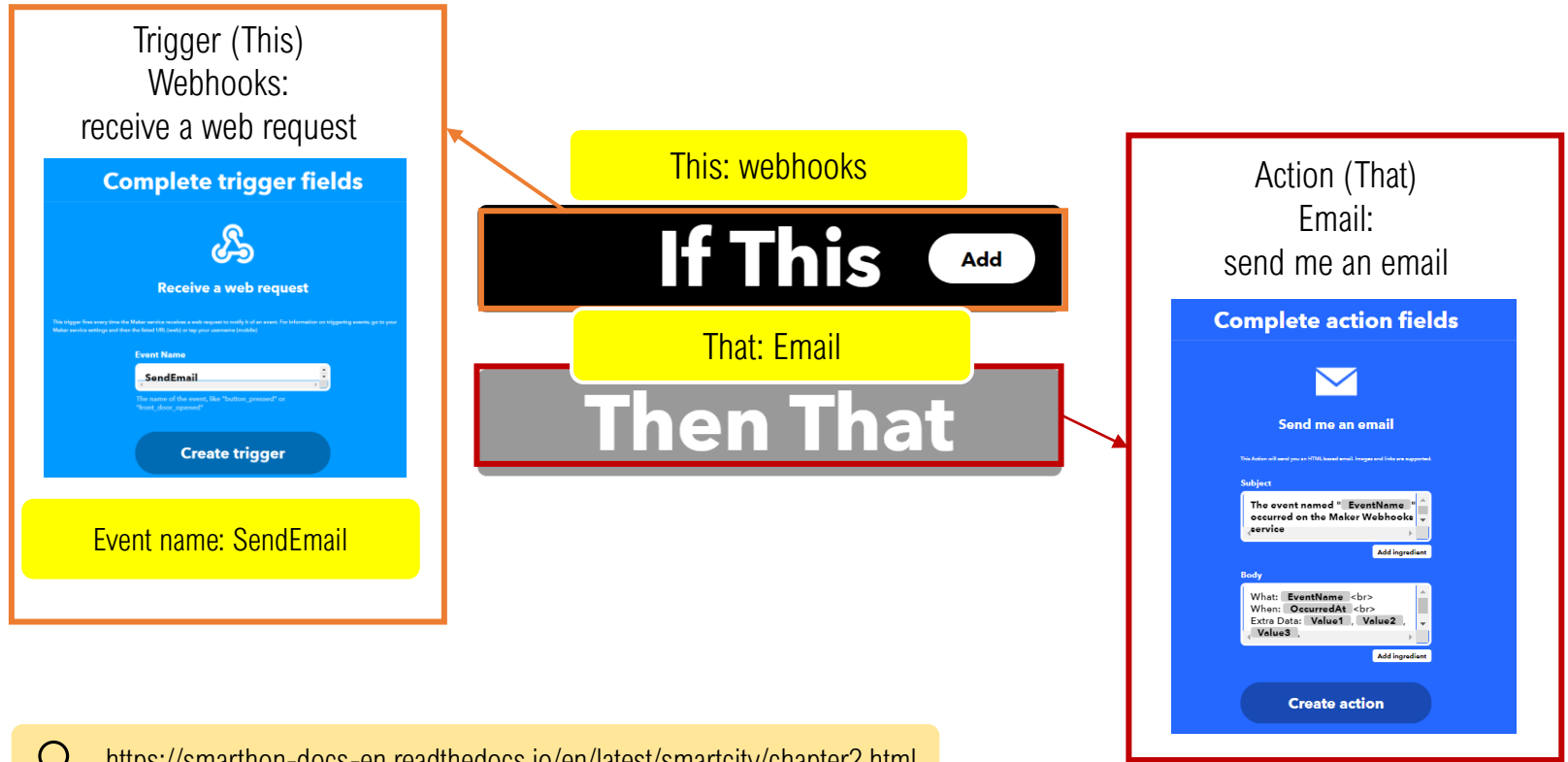


F2 model X1



IOT – IFTTT

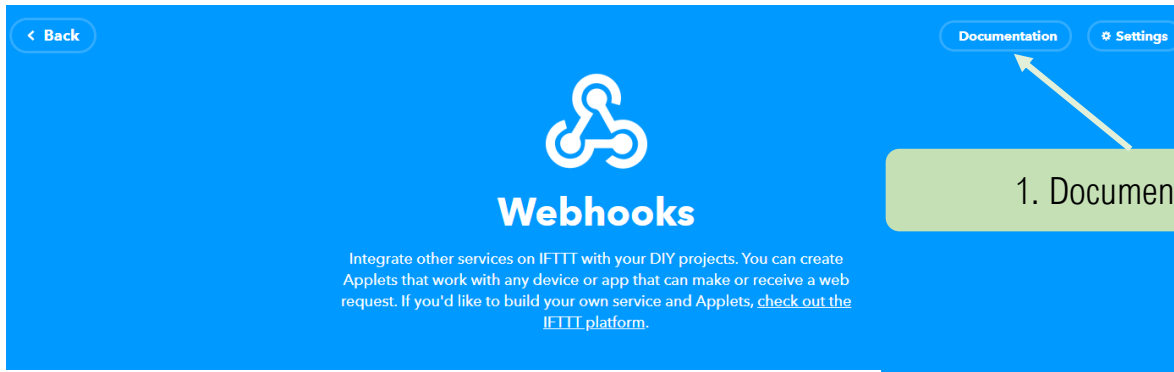
Step 1. Go to <https://ifttt.com/> , create applet (if webhooks then Email)



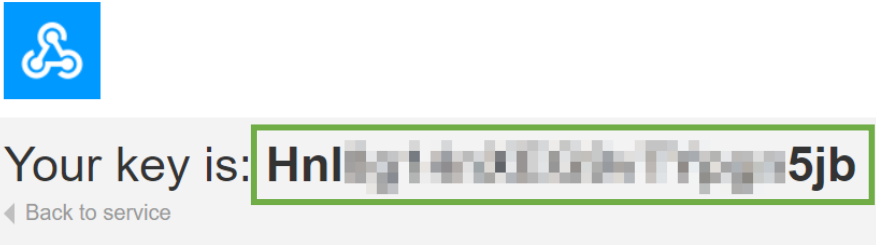
<https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/chapter2.html>

IOT – IFTTT

Step 2. Go to “My services” > “Webhooks”, select “Documentation” .
Copy your Webhooks Key as follows:



1. Documentation



2. Copy this IFTTT key

 <https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/chapter2.html>

CODING



How to avoid sending duplicate emails? (tips: using variable)?

```
1 on start
2 initialize OLED with width 128 height 64
3 Initialize IoT:bit TX P16 RX P8
4 Set WiFi to ssid "smarthon" pwd "12345678"
5 forever
6 if WiFi connected? then
7   if Get motion (triggered or not) at Pin P1 = true then
8     play tone Middle C for 1 beat
9     show icon monster
10    Send IFTTT key* "HnL...5jb"
11    event_name* "SendEmail"
12  else
13    show icon smile
14    pause (ms) 1000
```

STEPS:

1. Block **on start** only runs once at the beginning.
2. **Initialize OLED with width:128, height:64**
3. **Initialize IoT:bit at P16 and P8**
4. **Set WiFi connection by entering WiFi name and password**
5. Block **forever** runs as a loop.
6. If **WiFi** is connected and if **motion sensor** is triggered then
7. Set buzzer to play tone **Middle C for 1 beat** and **show a monster icon** on micro:bit.
8. **Send IFTTT event "SendEmail"** with provided IFTTT key.
9. Else **show a smile icon** if there is no suspicious movement near the door.
10. **Pause 1 second**

CODING



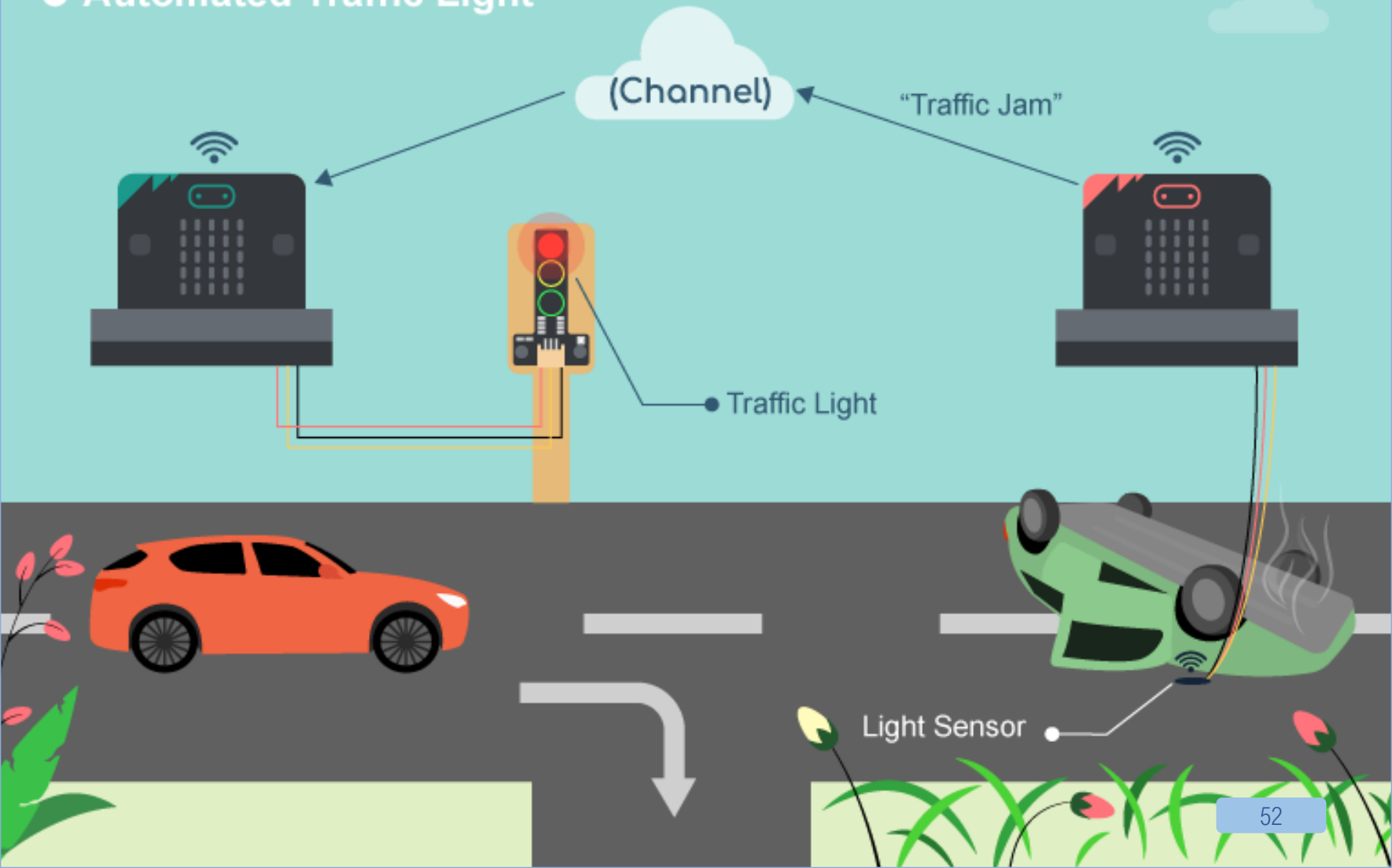
How to avoid sending duplicate emails? (tips: using variable)?

```
11 On WiFi connected IP_Address Device_ID
12 show icon [WiFi icon]
13 On IFTTT Uploaded Status Error_code
14 clear OLED display
15 show string join "IFTTT: " Status
16 show string join "Error: " Error_code
```

STEPS:

11. When WiFi is connected
12. Micro:bit LED will show icon "tick"
13. When IFTTT uploaded
14. Clear OLED display
15. Show string text "IFTTT" & Status
16. Show string text "Error" & Error_code

● Automated Traffic Light



8. [IOT] AUTOMATED TRAFFIC LIGHT (SENDER)

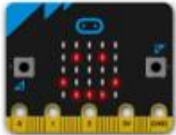
LEVEL: ★ ★ ★ ★ ★

Introduction

When the light value detected is too low, this would represent there is a traffic jam and a 'trafficjam' wifi message to another microbit. When the light value detected is high, this would represent there is no traffic jam and send a "nojam" wifi message to another micro:bit.

REMARKS: This project requires 2 micro:bit, one for SENDER and one for RECEIVER.

PART LIST



micro:bit (1)



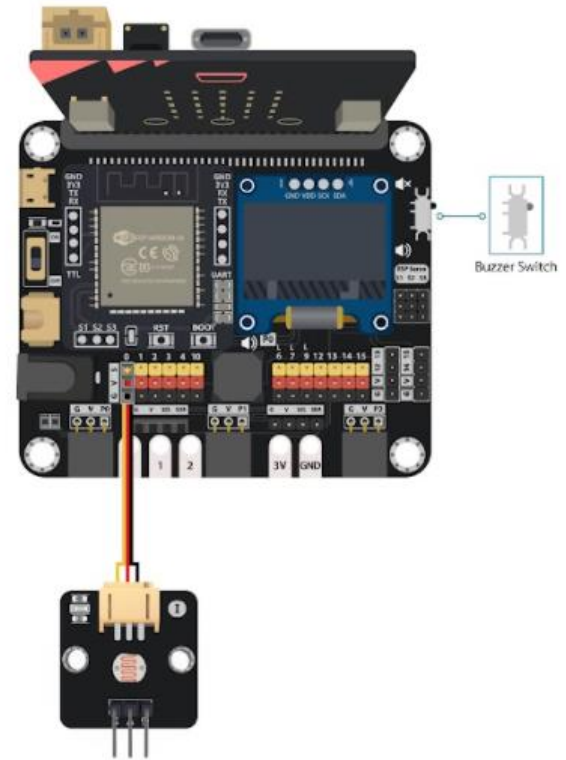
IoT:bit with OLED (1)



Light Sensor (1)



3-pin module wire (1)



CODING



How can we use distance sensor to detect traffic status?

```
1 on start
2 initialize OLED with width 128 height 64
3 Initialize IoT:bit TX P16 RX P8
4 Set WiFi to ssid "smarthon" pwd "12345678"
5 set light2 to 0

6 On WiFi connected IP_Address Device_ID
   show icon [LED icon]
```

STEPS:

1. Block **on start** only runs once at the beginning.
2. **Initialize OLED with width:128, height:64**
3. **Initialize IoT:bit at P16 and P8**
4. **Set WiFi connection** by entering WiFi name and password.
5. Set **light2** to 0.
6. **On WiFi connected**, micro:bit LED will **show icon "tick"**

CODING



How can we use distance sensor to detect traffic status?

```
7 forever
8 if WiFi connected? then
  set light2 to Get light value (percentage) at Pin P0
9 clear OLED display
10 show string join "Light: " light2
11 if light2 < 10 then
  WiFi Sender send channel "tsuenwan" message "trafficjam"
12 else
  WiFi Sender send channel "tsuenwan" message "nojam"
13 pause (ms) 6000
```

STEPS:

7. Block **forever** runs as a loop.
8. Set if **WiFi** is connected, set **light2** to get value from **P0**
9. **Clear OLED display**
10. **Show string text "Light:" & light2**
11. If **light2** < 10 then **WiFi** send message **trafficjam** in channel **"tsuenwan"**.
12. Else, **WiFi** send message **nojam** in channel **"tsuenwan"**.
13. **Pause for 6 seconds.**

8. [IOT] AUTOMATED TRAFFIC LIGHT (RECEIVER)

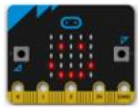
LEVEL: ★ ★ ★ ★ ★

Introduction

When a wifi message “trafficjam” is received, it means there is traffic jam forward. The traffic LED Module will turn red. When a wifi message “nojam” is received, it means there is no traffic jam forward. The traffic LED Module will turn green. By using smart traffic light, the problem of traffic jam can be reduced as automatic traffic control is used.

REMARKS: This project requires 2 micro:bit, one for SENDER and one for RECEIVER.

PART LIST



micro:bit (1)



IoT:bit with OLED (1)



Traffic Light Module (1)



3-pin module wire (1)



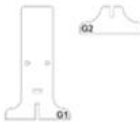
M4*10mm Screw (2)



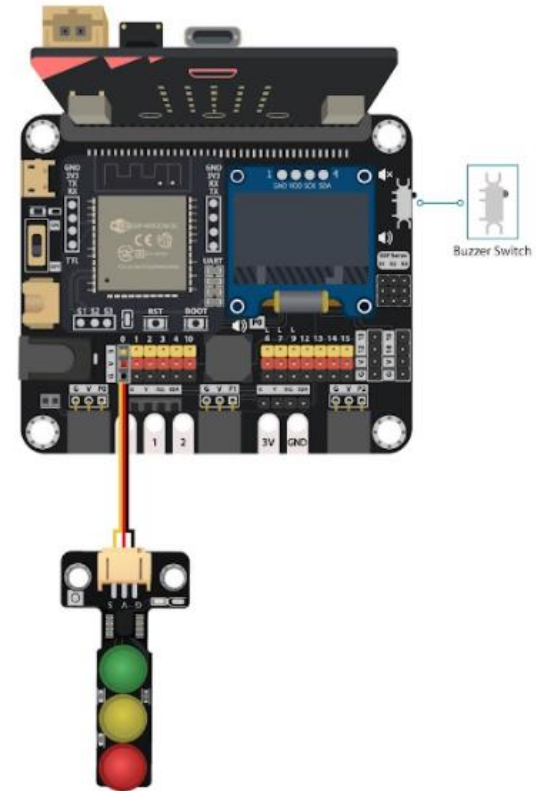
M4 Nut (2)



Screwdriver (1)

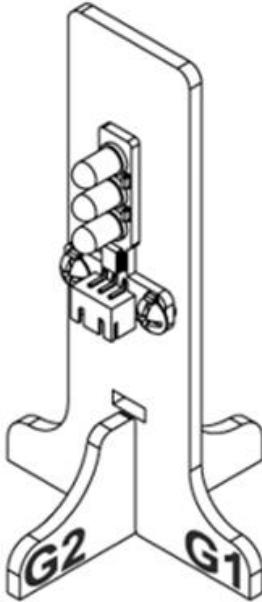
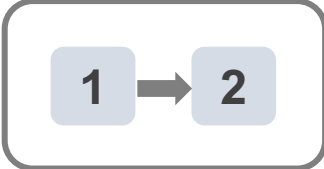
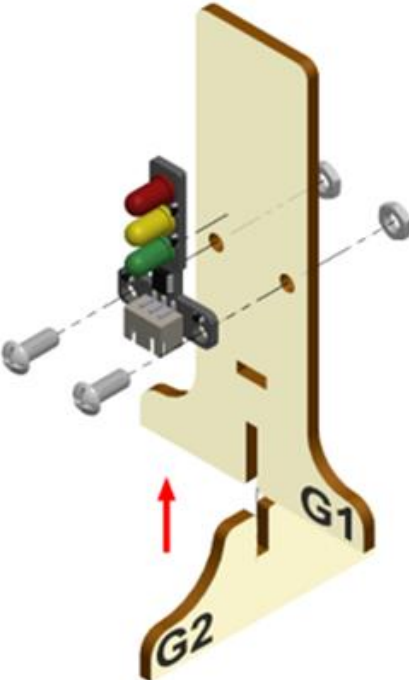


G1-G2 Model (1)



ASSEMBLY STRUCTIONS

- G1-G2 model X1
- Traffic light module X1
- M4 Nut X2
- M4*10mm Screw X2



CODING



How can we add sound effect to the traffic LED Module according to the corresponding color?

1 on start

2 initialize OLED with width 128 height 64

3 Initialize IoT:bit TX P16 RX P8

4 Set WiFi to ssid "smarthon" pwd "12345678"

5 set oldmsg to ""

6 call TurnGreen

7 On WiFi connected IP_Address Device_ID

8 WiFi Receiver join channel "tsuenwan"

STEPS:

1. Block **on start** only runs once at the beginning
2. **Initialize OLED with width:128, height:64**
3. Initialize IoT:bit at P16 and P8
4. Set WiFi connection by entering WiFi name and password.
5. Set **oldmsg** = ""
6. Call **TurnGreen**
7. On WiFi connected, Micro:bit LED will **show icon "tick"**
8. WiFi Receiver join channel "tsuenwan"

CODING



How can we add sound effect to the traffic LED Module according to the corresponding color?

```
9 On WiFi Receiver received Channel receivedMessage
10 clear OLED display
11 show string join "Message: " receivedMessage
12 if receivedMessage ≠ oldmsg then
   set oldmsg to receivedMessage
13 if receivedMessage = "trafficjam" then
   call TurnRed
14 else if receivedMessage = "nojam" then
   call TurnGreen
```

STEPS:

9. When WiFi received message from channel
10. Clear OLED display
11. Show string text "Message:" & receivedMessage
12. If receivedMessage ≠ oldmsg, then
Set oldmsg = receivedMessage
13. If receivedMessage = "trafficjam", then
call TurnRed
14. Else if receivedMessage = nojam, then
call TurnGreen.

CODING



How can we add sound effect to the traffic LED Module according to the corresponding color?

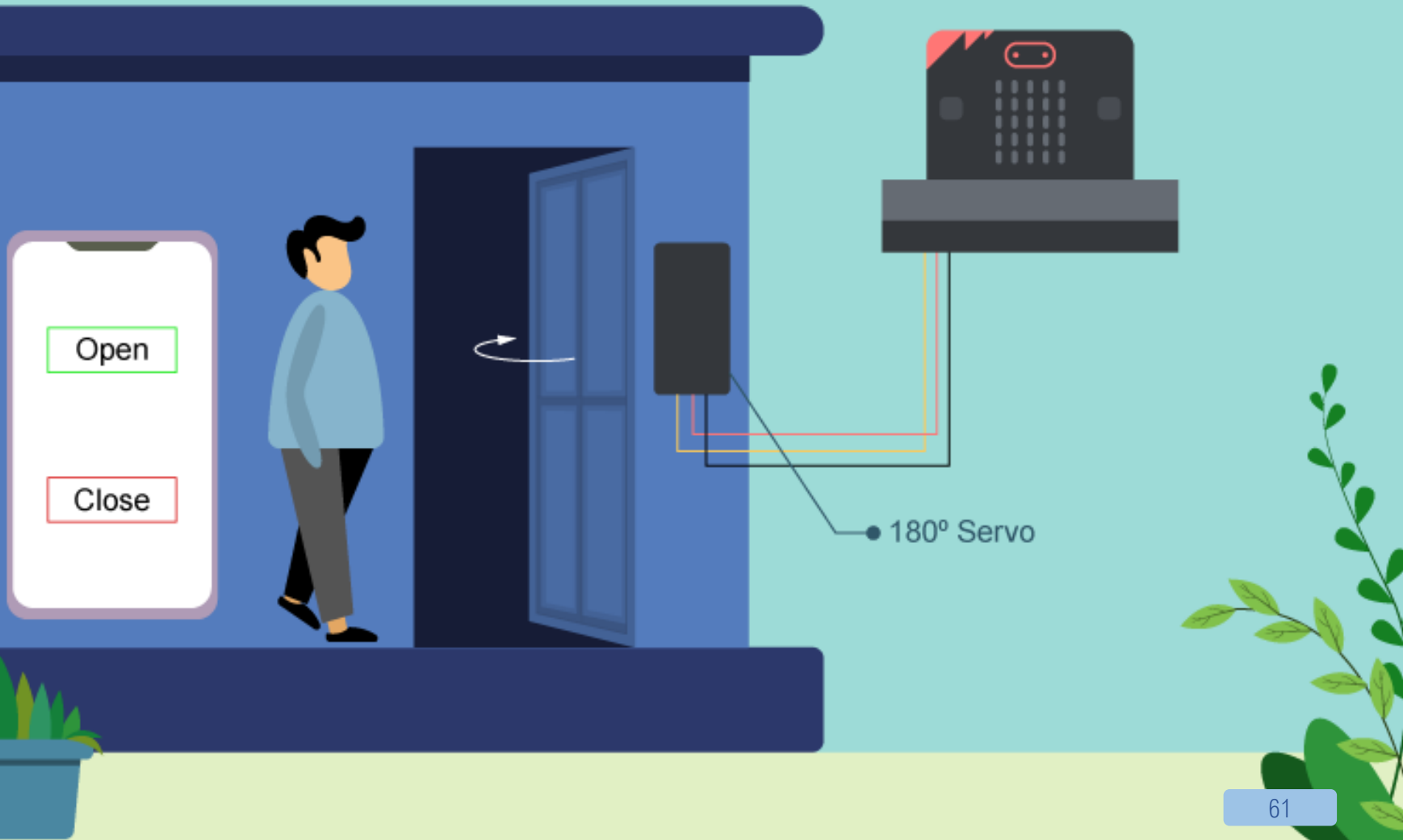
15

The image shows two columns of Scratch code blocks, each enclosed in a dashed yellow rounded rectangle. The left column is titled 'function TurnGreen' and the right column is titled 'function TurnRed'. Each column contains three 'Control traffic light' blocks, each followed by a 'pause (ms) 2000' block. In the 'TurnGreen' function, the first block has Red ON, Yellow OFF, and Green OFF; the second block has Red ON, Yellow ON, and Green OFF; and the third block has Red OFF, Yellow OFF, and Green ON. In the 'TurnRed' function, the first block has Red OFF, Yellow OFF, and Green ON; the second block has Red OFF, Yellow ON, and Green OFF; and the third block has Red ON, Yellow OFF, and Green OFF.

STEPS:

15. Set up two **new functions** (**TurnRed** & **TurnGreen**). In function **TurnRed**, control traffic light to turn from green, then yellow and finally red, each colour pauses for 2000ms. In function **TurnGreen**, control traffic light to turn from red, then red and yellow at the same time and finally green, each colour **pauses for 2s**.

● Smart House Door Control



9. [IOT] SMART HOUSE DOOR CONTROL

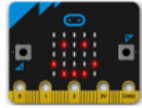
LEVEL: ★ ★ ★ ★ ☆



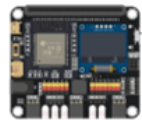
Introduction

When the micro:bit receives the signal “opendoor” from the app, the 180° servo will turn for 180° to open the door. When the micro:bit receives the signal “closeddoor” from the app, the 180° servo will turn back 180° to close the door.

PART LIST



micro:bit (1)



IoT:bit with OLED (1)



180° Servo (1)



Servo horn (1)



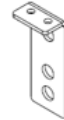
Servo M2*7.5mm sharp screw (1) (Included in servo pack)



M3*10mm Screw (2)



M3 Nut (2)



L-shaped stand (1)



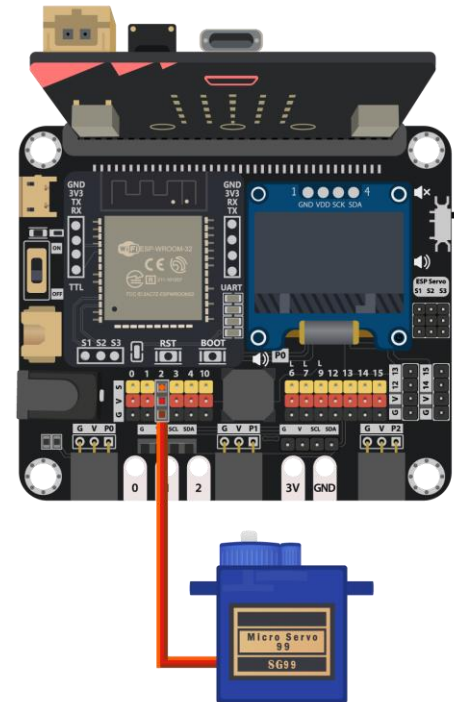
Paper clip (1)



Screwdriver (1)

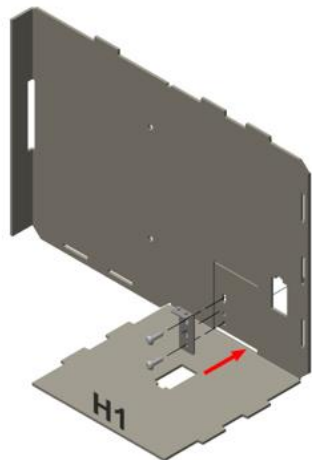


H1-H2 Model (1)

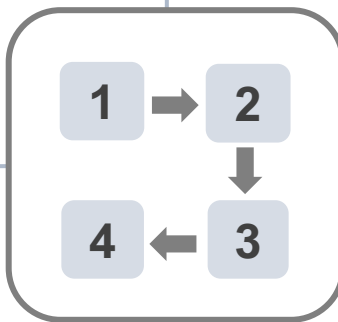
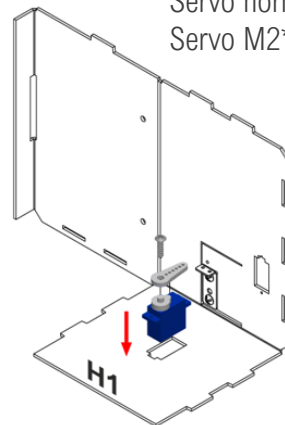


ASSEMBLY STRUCTIONS

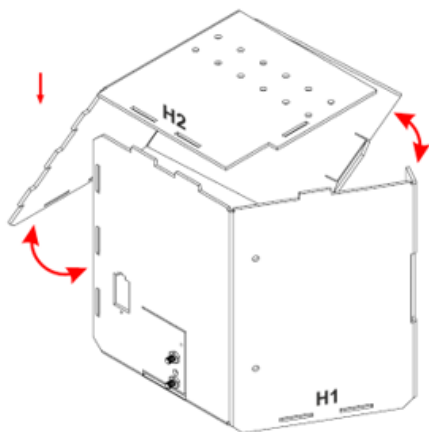
- L-shaped stand X1
- M4 Nut X2
- M4*10mm Screw X2
- H1 model X1



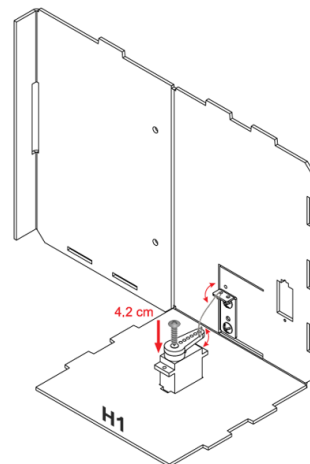
- 180° Servo X1
- Servo horn X1
- Servo M2*7.5mm sharp screw X1



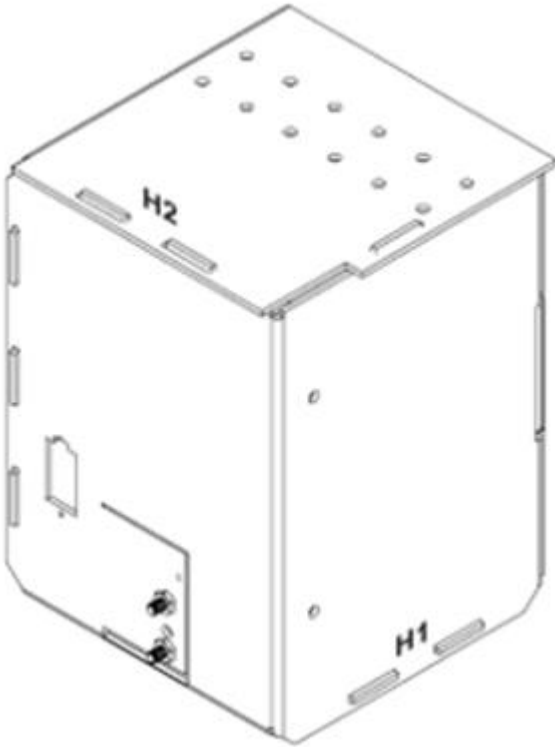
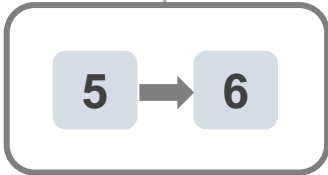
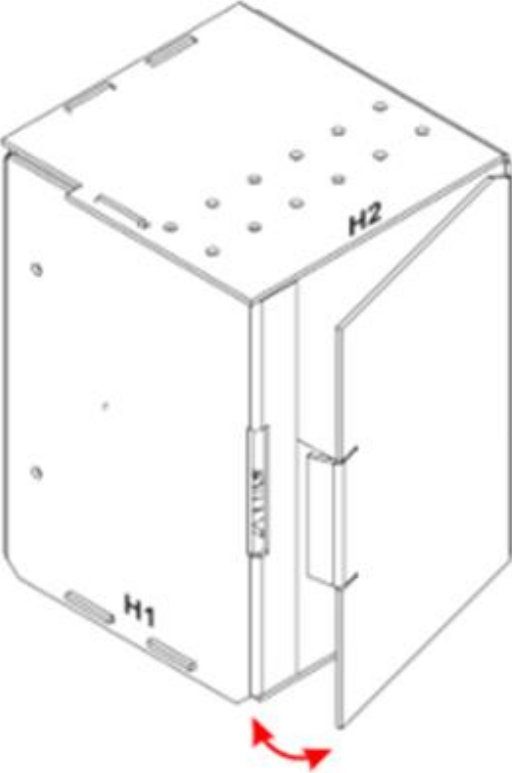
- H2 model X1



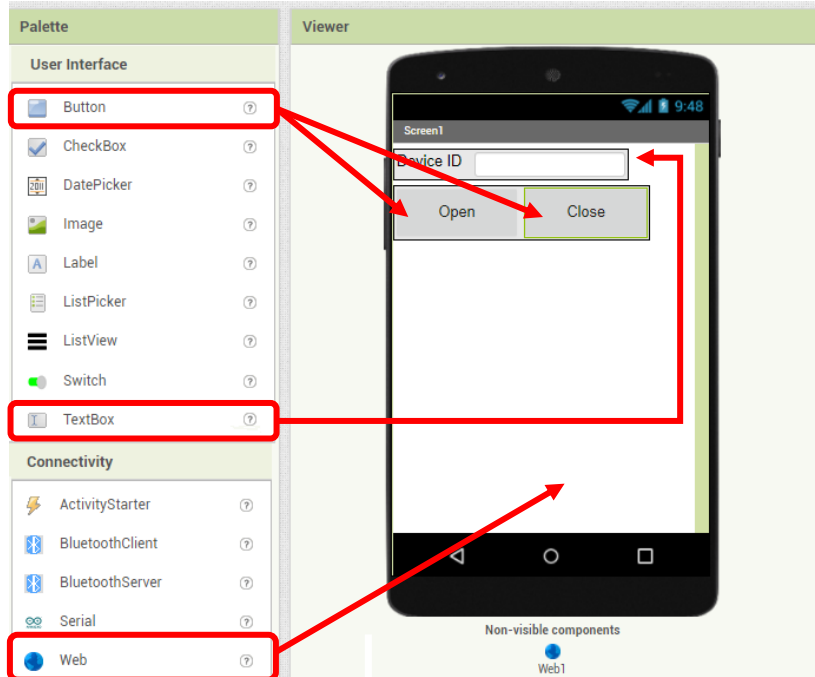
- Paper Clip X1



ASSEMBLY INSTRUCTIONS



IOT – APP INVENTOR 2



Step 1. Create the page with the components

On Designer:

Drag the components from the left menu – 2 “Buttons” (to open or close the door) and a “Textbox” (to input the Device ID number).

Then, add the invisible component “Web” under connectivity, we will need it for WAN connection.



<https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/chapter3.html>

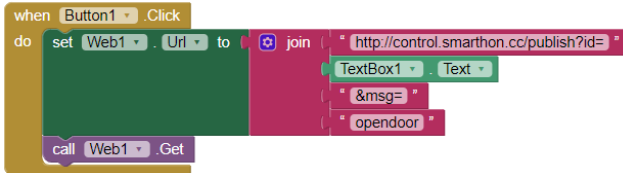
IOT – APP INVENTOR 2

Step 2. Make the programming part

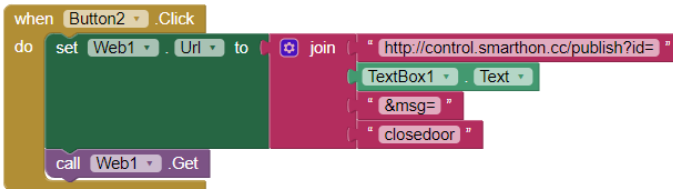
- **On Blocks:**
- The WAN control command URL is:

<http://control.smarthon.cc/publish?id=DeviceID&msg=ControlCommand>

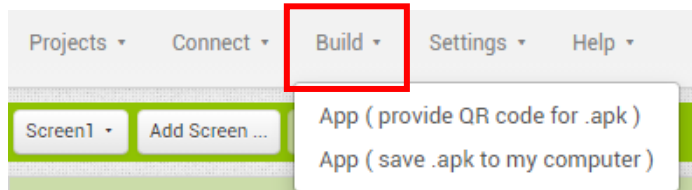
- When Button1 is clicked, it will direct to the URL with given device ID and command “opendoor”.



- When Button2 is clicked, it will direct to the URL with given device ID and command “closedoor”.



- Download the App in your phone to by .apk or scan QR code.



CODING



How can we add password authentication to open the door?

```
1 on start
2 initialize OLED with width 128 height 64
3 Initialize IoT:bit TX P16 RX P8
4 Set WiFi to ssid "smarthon" pwd "12345678"
5 Turn Servo to 180 degree at P2

6 On WiFi connected IP_Address Device_ID
   show icon tick
7 show string Device_ID
```

STEPS:

1. Block **on start** only runs once at the beginning.
2. **Initialize OLED with width:128, height:64**
3. **Initialize IoT:bit at P16 and P8**
4. Set **WiFi connection** by entering WiFi name and password.
5. **turn servo to 180 degree at P2.**

6. On WiFi connected, micro:bit LED will **show icon "tick"**
7. **Show string Device_ID**

CODING



How can we add password authentication to open the door?

```
8 On WiFi received WAN_Command
9 clear OLED display
10 show string join "Command: " WAN_Command
11 if WAN_Command = "opendoor" then
    Turn Servo to 45 degree at P2
12 else if WAN_Command = "closedoor" then
    Turn Servo to 180 degree at P2
```

STEPS:

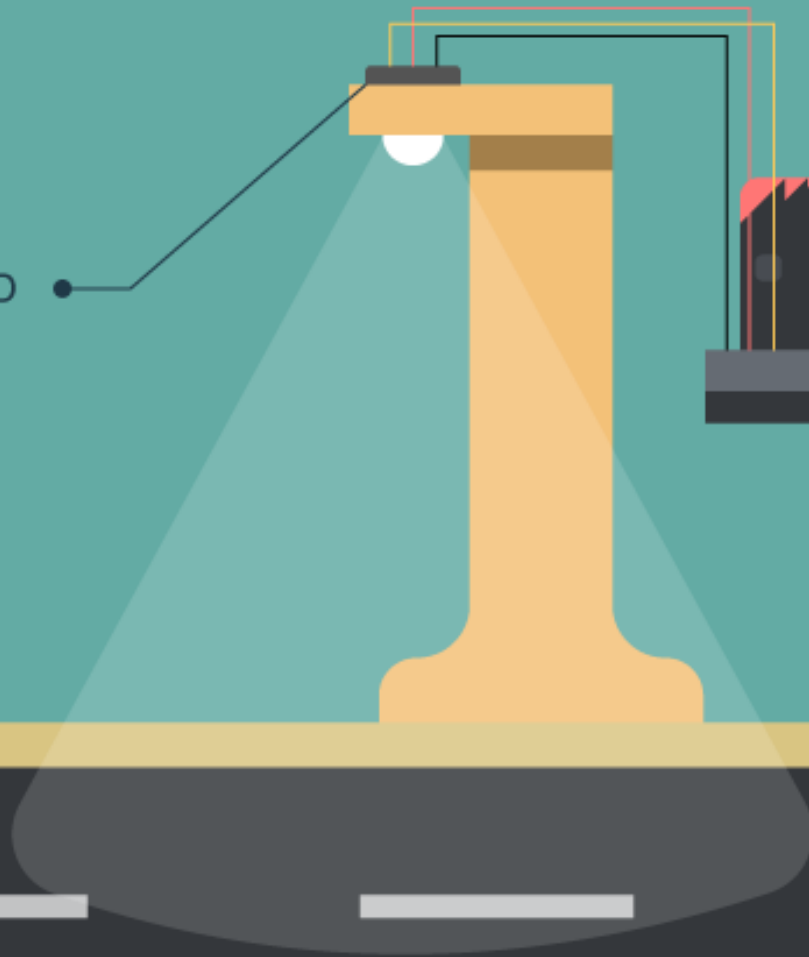
8. When WAN command is received,
9. Clear OLED display
10. Show string text "Command:" & WAN_Command
11. If WAN_Command = "opendoor" then turn servo to 45 degree at P2
12. Else if WAN_Command = "closedoor" then, turn servo to 180 degree at P2.

● Smart Street Light

06:00 PM

IFTTT

White LED



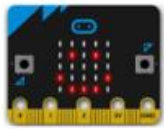
10. [IOT] SMART STREET LIGHT

LEVEL: ★ ★ ★ ★ ☆

Introduction

To improve the living stand of citizen and to save electricity, smart street light can be automatically turned on at night (e.g. 6pm) and turned off in the morning (e.g. 6am).

PART LIST



micro:bit (1)



IoT:bit with OLED (1)



White LED Light (1)



3-pin module wire (1)



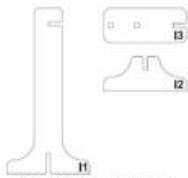
M4*10mm Screw (2)



M4 Nut (2)

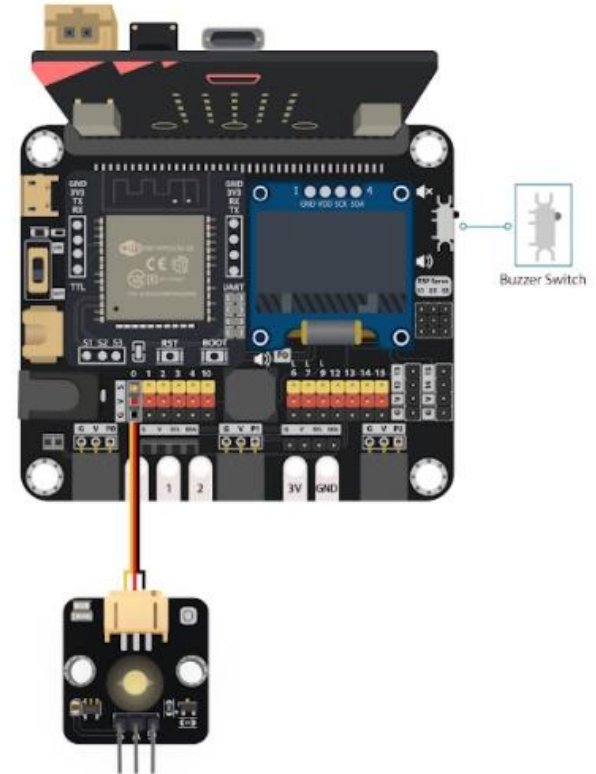


Screwdriver (1)



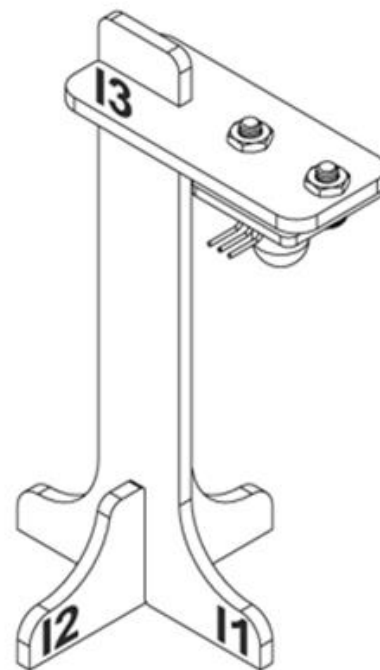
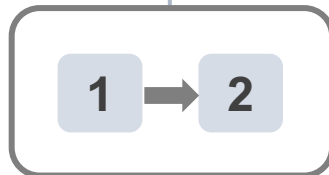
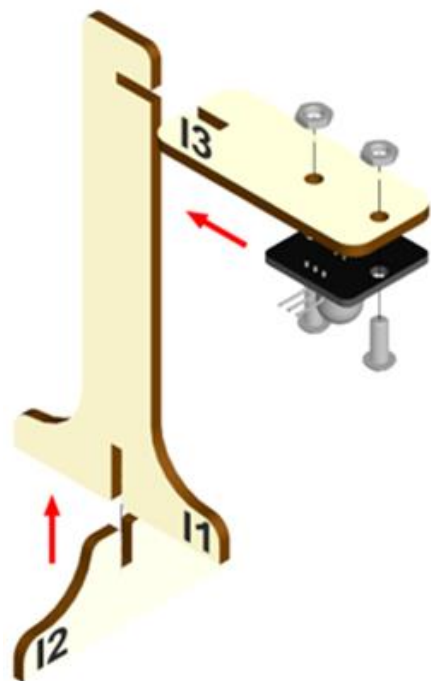
I1-I2 Model (1)

IFTTT



ASSEMBLY STRUCTIONS

I1-I3 model	X1
LED light	X1
M4 Nut	X2
M4*10mm Screw	X2



CODING



How to turn on the light automatically if the today's weather is cloudy reported by IFTTT?

```
1 on start
2 initialize OLED with width 128 height 64
3 Initialize IoT:bit TX P16 RX P8
4 Set WiFi to ssid "smarthon" pwd "12345678"

5 On WiFi connected IP_Address Device_ID
   show icon [LED icon]
6 show string Device_ID
```

STEPS:

1. Block **on start** only runs once at the beginning.
2. **Initialize OLED with width:128, height:64**
3. **Initialize IoT:bit at P16 and P8**
4. Set WiFi connection by entering WiFi name and password.
5. On WiFi connected, micro:bit LED will **show icon "tick"**
6. **Show string Device_ID**

CODING



How to turn on the light automatically if the today's weather is cloudy reported by IFTTT?

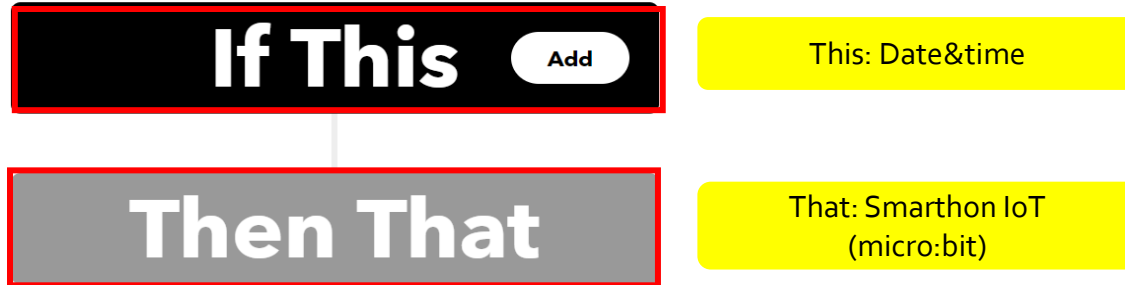
```
7 On WiFi received WAN_Command
8 clear OLED display
9 show string join "Command: " WAN_Command
10 if WAN_Command = "lighton" then
    Turn White LED to 1023 at P1
11 else if WAN_Command = "lightoff" then
    Turn White LED to 0 at P1
```


STEPS:

7. When WAN command is received,
8. Clear OLED display
9. Show string text "Command:" & WAN_Command
10. If WAN_Command = lighton then, turn White LED to 1023 at P1.
11. If WAN_Command = lightoff then, turn White LED to 0 at P1.

IOT – CLOUD CONTROL BY IFTTT

Step 1. Create applet (If Date&time then Smarthon IoT (micro:bit))

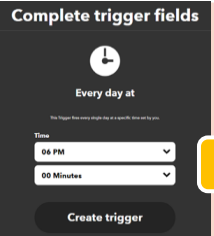
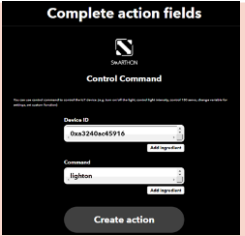


 <https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/chapter4.html>

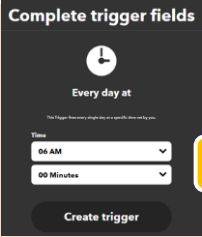
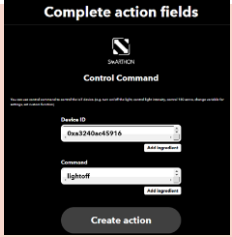
IOT – CLOUD CONTROL BY IFTTT

Step 2.

- If it is at 18:00, send WAN control command: light on to the micro:bit

Trigger (This): Date&time Every day at	Action (That): Smarhon IoT (micro:bit) Control Command
 <p>6:00pm</p>	 <p>Device ID</p> <p>lighton</p>

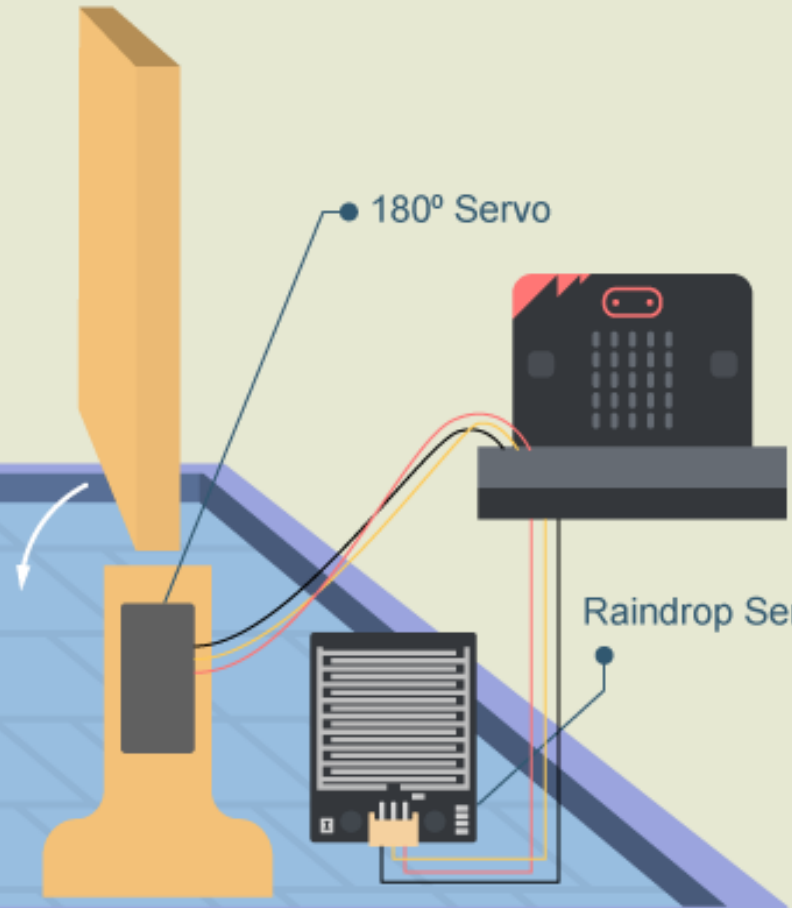
- If it is at 06:00, send WAN control command: light off to micro:bit

Trigger (This): Date&time Every day at	Action (That): Smarhon IoT (micro:bit) Control Command
 <p>6:00am</p>	 <p>Device ID</p> <p>lightoff</p>

* Load the program to micro:bit and get your device ID



● Roof Garden Clothes Rack



● 180° Servo

● Raindrop Sensor

11. [IOT] ROOF GARDEN CLOTHES RACK

LEVEL: ★ ★ ★ ★ ☆

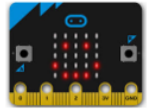
Introduction

When raindrop sensor sensed that it is raining, it will send a signal to micro:bit and therefore the clothes rack will be opened. When there is no rain, the clothes rack will be closed.

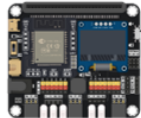
Also, micro:bit can receive WAN commands from WAN (eg. IFTTT), the the micro:bit will get the command name. the rack will be opened/closed.

IFTTT

PART LIST



micro:bit (1)



IoT:bit with OLED (1)



180° Servo (1)



3-pin module wire (1)



Servo M2*7.5mm sharp screw (1) (Included in servo pack)



M2*10mm Screw (2)



M2 Nut (2)



M4*10mm Screw (2)



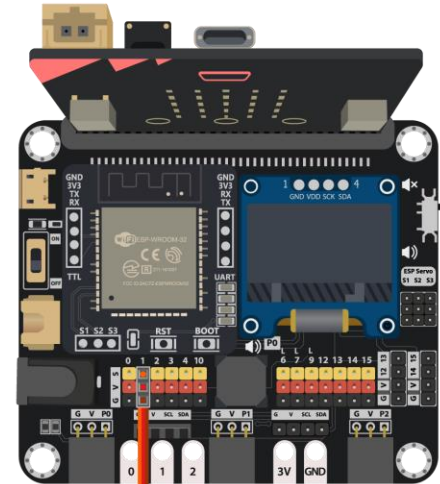
M4 Nut (2)



Screwdriver (1)

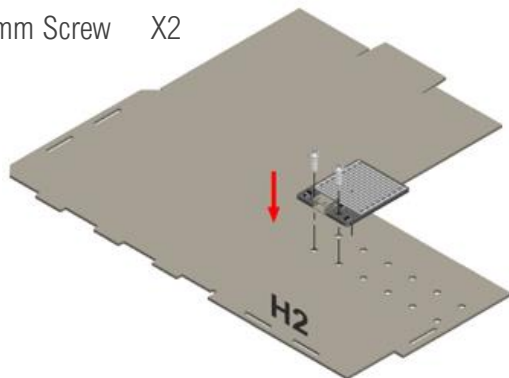


H1-H7 Model (1)

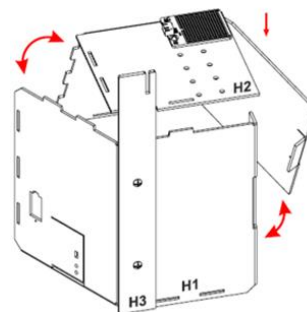
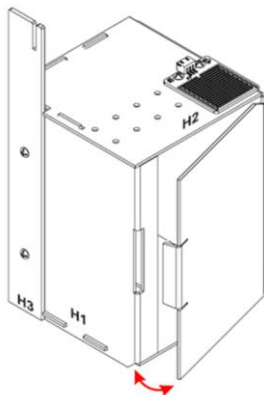
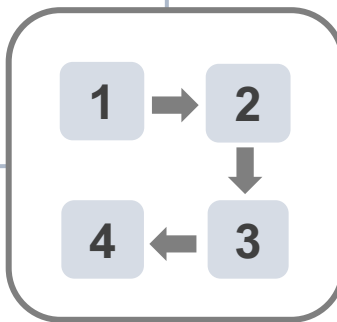
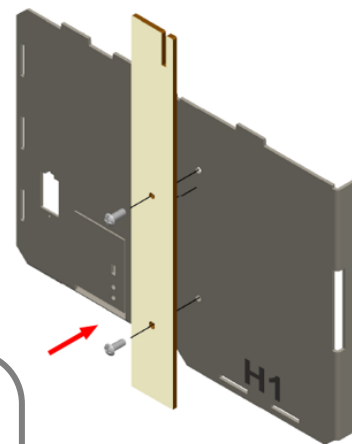


ASSEMBLY STRUCTIONS

H2 model	X1
Raindrop sensor	X1
M4 Nut	X2
M4*10mm Screw	X2

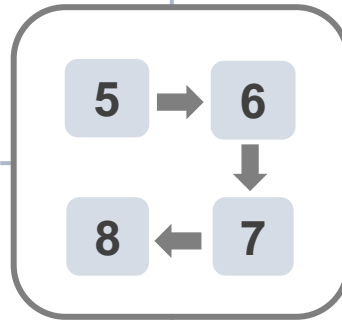
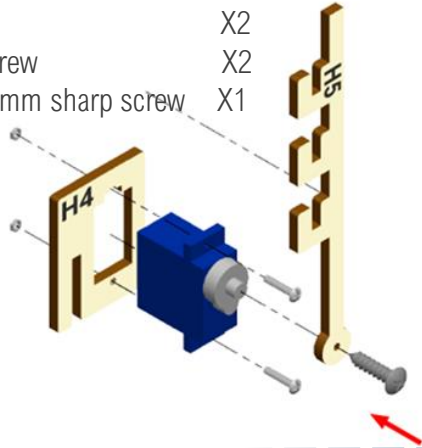


H1 & H3 model	X1
M4 Nut	X2
M4*10mm Screw	X2

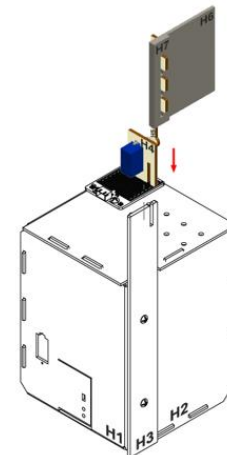
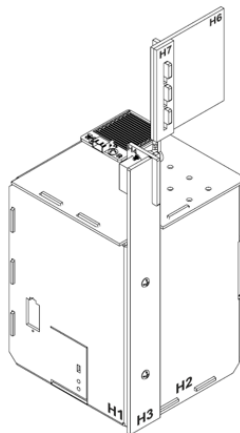
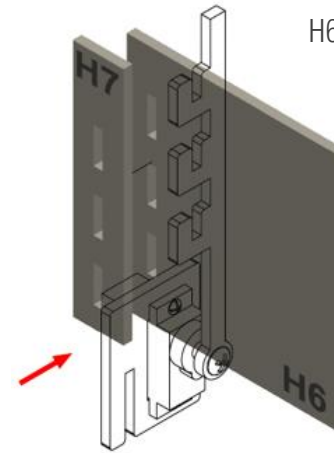


ASSEMBLY STRUCTIONS

- H4 - H5 model X1
- 180° Servo X1
- M2 Nut X2
- M2*10mm Screw X2
- Servo M2*7.5mm sharp screw X1



- H6-H7 model X1



CODING



Can you control the clothing rack by other weather conditions? (e.g. sunlight)

```
1 on start
2 initialize OLED with width 128 height 64
3 Initialize IoT:bit TX P16 RX P8
4 Set WiFi to ssid "smarthon" pwd "12345678"
5 On WiFi connected
   IP_Address Device_ID
   show icon [tick icon]
6 show string join "ID: " Device_ID
```

STEPS:

1. Block **on start** only runs once at the beginning.
2. **Initialize OLED with width:128, height:64**
3. **Initialize IoT:bit at P16 and P8**
4. **Set WiFi connection by entering WiFi name and password.**
5. **On WiFi connected, micro:bit LED will show icon "tick"**
6. **Show string Device_ID**

CODING



Can you control the clothing rack by other weather conditions? (e.g. sunlight)

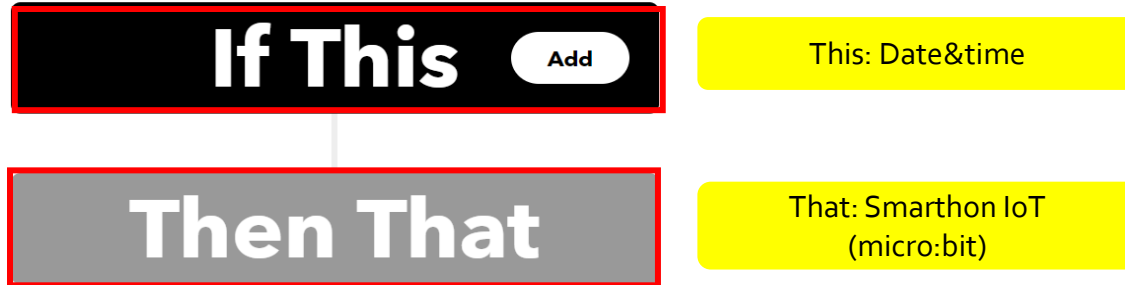
```
7 On WiFi received WAN_Command
8 clear OLED display
9 show string join "Command: " WAN_Command
10 if WAN_Command = "Rain" then
    Turn Servo to 90 degree at P1
11 else if WAN_Command = "Clear" then
    Turn Servo to 180 degree at P1
12 on button A pressed
13 Turn Servo to 90 degree at P1
14 on button B pressed
15 Turn Servo to 180 degree at P1
```

STEPS:

7. When WAN command is received,
8. Clear OLED display
9. Show string text "Command:" & WAN_Command
10. If WAN_Command = Rain then,
Turn Servo to 90 degree at P1.
11. If WAN_Command = Clear then, Turn Servo to 180 degree at P1.
12. When button A is pressed,
13. Turn Servo to 90 degree at P1.
14. When button B is pressed,
15. Turn Servo to 180 degree at P1.

IOT – CLOUD CONTROL BY IFTTT

Step 1. Create applet: If Weather Underground then Smarthon IoT(micro:bit)


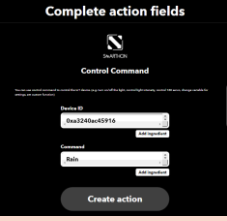


<https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/chapter4.html>

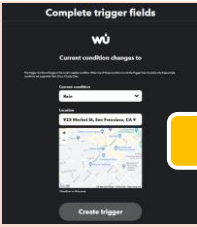
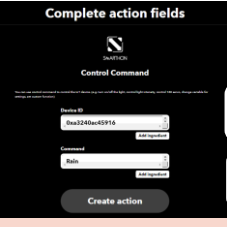
IOT – CLOUD CONTROL BY IFTTT

Step 2.

- If weather condition change to rain, send WAN control command: Rain to the micro:bit

Trigger (This): Weather Underground Current condition changes to	Action (That): Smarthon IoT (micro:bit) Control Command
 <p>Rain</p>	 <p>Device ID</p> <p>Rain</p>

- If weather condition change to clear, send WAN control command: Clear to micro:bit

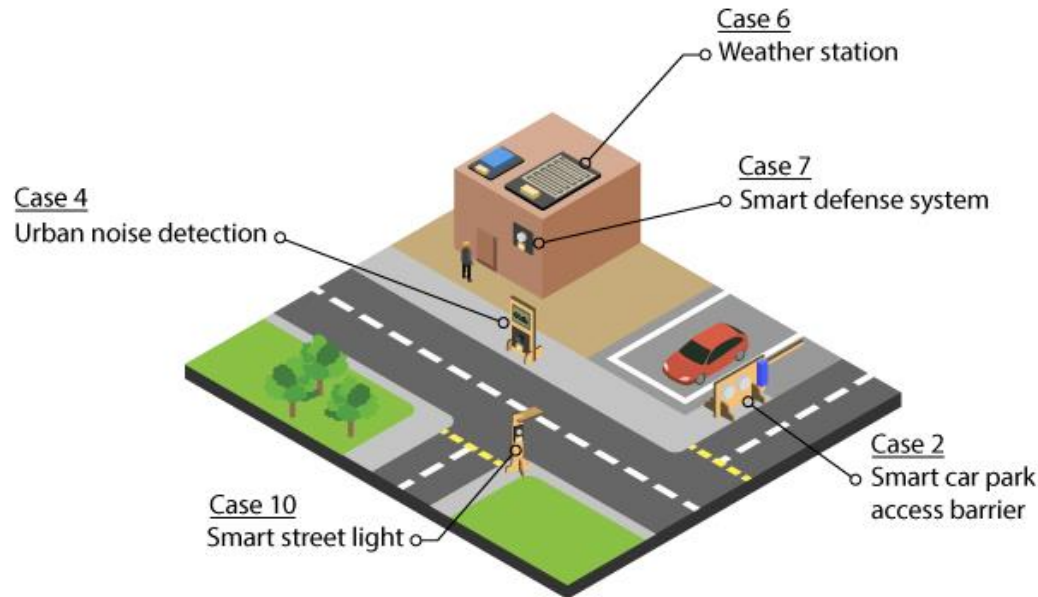
Trigger (This): Weather Underground Current condition changes to	Action (That): Smarthon IoT (micro:bit) Control Command
 <p>Clear</p>	 <p>Device ID</p> <p>Clear</p>

* Load the program to micro:bit and get your device ID



SCENARIO EXAMPLE 1: MODERN RESIDENTIAL HOUSE

Modern residential house focus on achieving a high living standard of the citizen. It includes automation system for parking and high security defence system. Noise and weather information are collected to evaluate the environmental comfortability of a city.



PARTS:

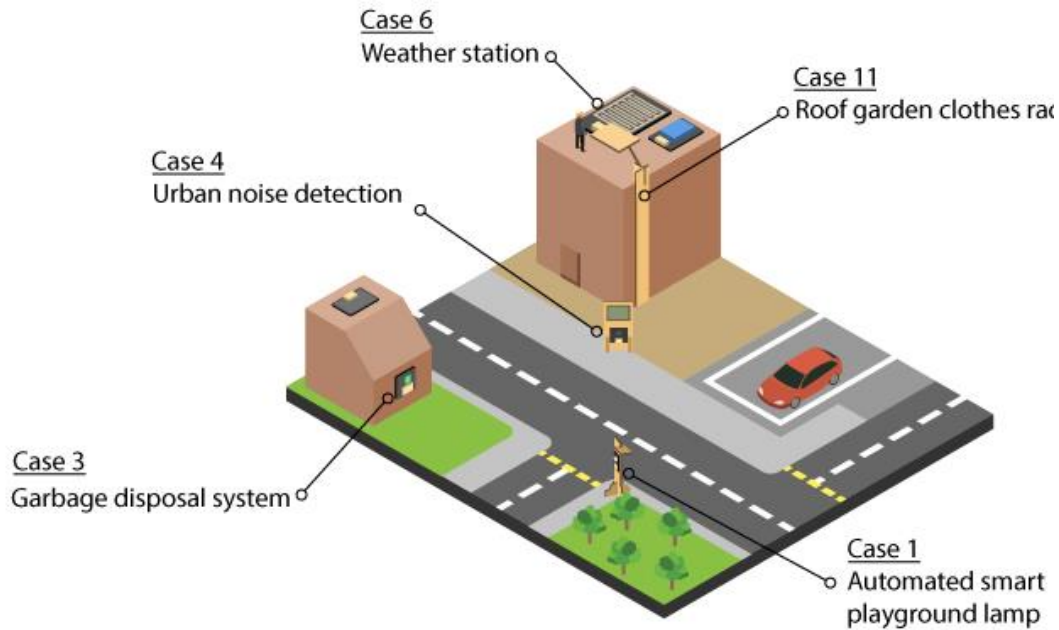
- Temperature and Humidity sensor (DHT11)
- Raindrop sensor
- Motion sensor
- Buzzer
- Light sensor
- 180° servos
- Distance sensor
- Noise sensor
- White LED module

IOT:

- ThingSpeak
- IFTTT

SCENARIO EXAMPLE 2: GREEN ENGINEERING

It is a city development, which focus on high sustainability It includes increasing green areas, energy efficiency, reducing wastes, monitoring climate information and pollution states (eg. Noise).



PARTS:

- Temperature and Humidity sensor (DHT11)
- Raindrop sensor
- 180° servos
- White LED module
- Motion sensor
- Noise sensor
- Distance sensor
- Multi-color LED (WS2812B)

IOT:

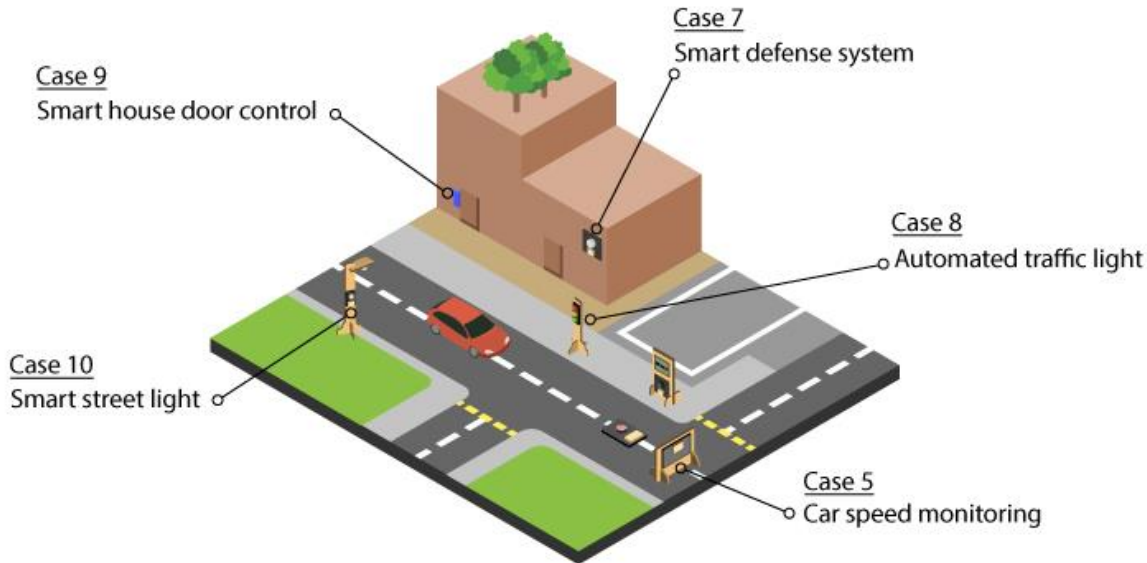
- ThingSpeak
- IFTTT



You can design your own scenario, simply combine different cases together or create dedicated cases in your city!

SCENARIO EXAMPLE 3: TRANSPORT SAFETY AND SECURITY

The transport safety and security is the primary concern of a city. Car speed and traffic condition are being monitored in the public transportation system. Only responsible person of the corporation can enter the building, alerts will be sent on the network if there are suspicious people nearby.



PARTS:

- 180° servos
- White LED module
- Buzzer
- Motion sensor
- Light sensor
- Traffic light module
- Distance sensor

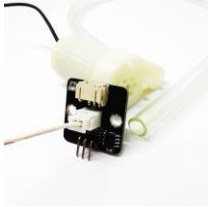

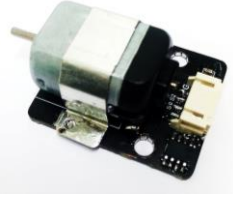


IOT:

- App Inventor 2
- IFTTT
- WiFi channel



You can design your own scenario, simply combine different cases together or create dedicated cases in your city!

APPENDIX I: ELECTRONIC MODULES ON SMARTHON – FUTHUR EXPLORATION

	Water Pump		Temperature, Humidity and Pressure Sensor
	Motor Fan		Digital Light Sensor
	Water Temperature Sensor	For more information, please visit www.smarthon.cc .	

**FOR MORE INFORMATION
PLEASE VISIT**

<https://smarthon-docs-en.readthedocs.io/en/latest/smartcity/index.html>



SMARTHON

www.smarthon.cc

INGITE YOUR CREATIVITY