



SMART HOME IOT

# MAKER 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
- How to build a House Model?..... P. 14-32
- Getting started: Add the Extension..... P. 8-11
- Case 01: Smart Saving Light Bulb..... P.15-20
- Case 02: Security Vanguard Alarm System..... P. 21-26
- Case 03: Environmental Motor Fan..... P. 15-18
- Case 04: Toilet Water Leakage Detection System..... P. 19-22
- Case 05: Smart Fridge Alert..... P. 23-26
- Case 06: Mechanical Password Switch Door..... P. 27-30
- Case 07: Smart Remote Control Musical Light..... P. 31-36

# CONTENT

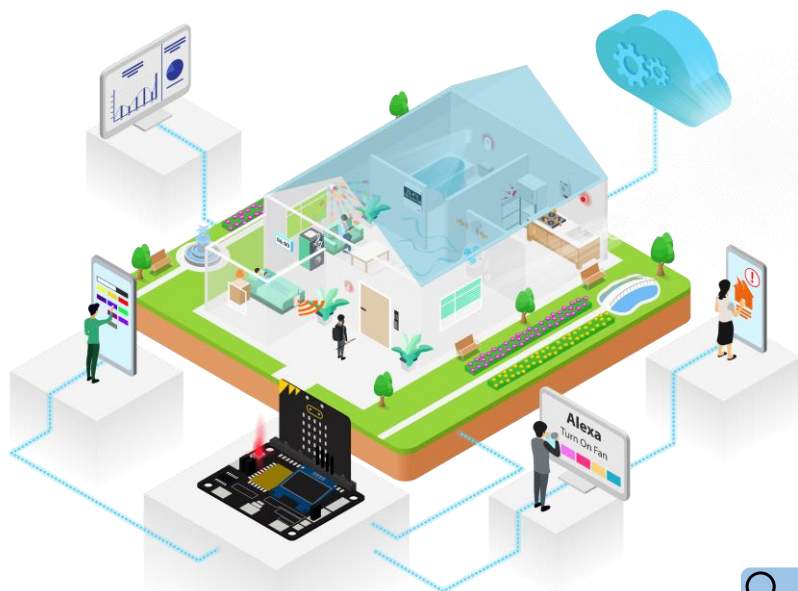
---



- Case 08: Automatic Sunlight Detecting Curtain..... P. 37-44
- IoT Case 09: Smart Digital Morning Clock..... P. 45-51
- IoT Case 10: Home Health Data Monitoring..... P. 52-60
- IoT Case 11: Kitchen Safety Flame Monitoring Alert..... P. 61-68
- IoT Case 12: Room Smart Colourful Light App ControlIT..... P. 69-75
- Adv IoT Case 13: Smart Fan Voice Control..... P. 76-83
- Scenario Example 1-4..... P. 84-86
- Appendix I: Electronic Modules on Smarthon – Further Exploration..... P. 87

# SMARTHON SMART HOME IOT MAKER KIT

---



**SMARTHON Smart Home 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 home creator and build a unique IoT system in the home. Based on Smarthon IoT board, which is compatible with multiple sensors and actuators, you can design home features; for example: using sensors to detect domestic status and upload home information to the internet.



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

# WHAT IS SMART Home?

Smart home 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 – personal wellness, entertainment, Internet of things, home automation, energy management and safety & Security.



## HOME ENTERTAINMENT

Using smart devices to improve the life quality and it can change the home-style easily and time-by-time



## INTERNET OF THINGS

Making use of the numerous internet resources, smart home can be monitored and controlled by internet.



## HOME AUTOMATION

Using automation technology to improve the convenience of life and reduce the redundant task in our daily life



## ENERGY MANAGEMENT

Monitoring the house energy consumption and saving energy by using smart control of electric appliances



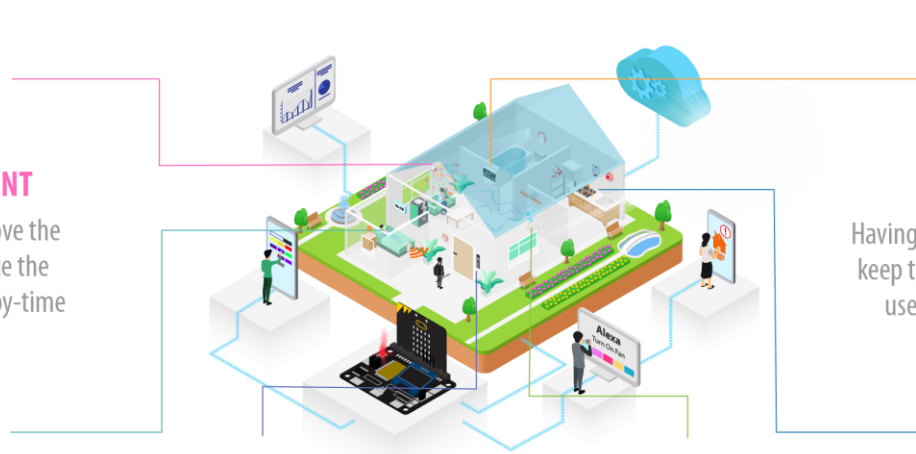
## PERSONAL WELLNESS

Having a health living environment can keep the personal wellness by making use of the technology/IoT device



## SAFETY & SECURITY

Protecting the property, life security and reducing the loss of potential dangers



# 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

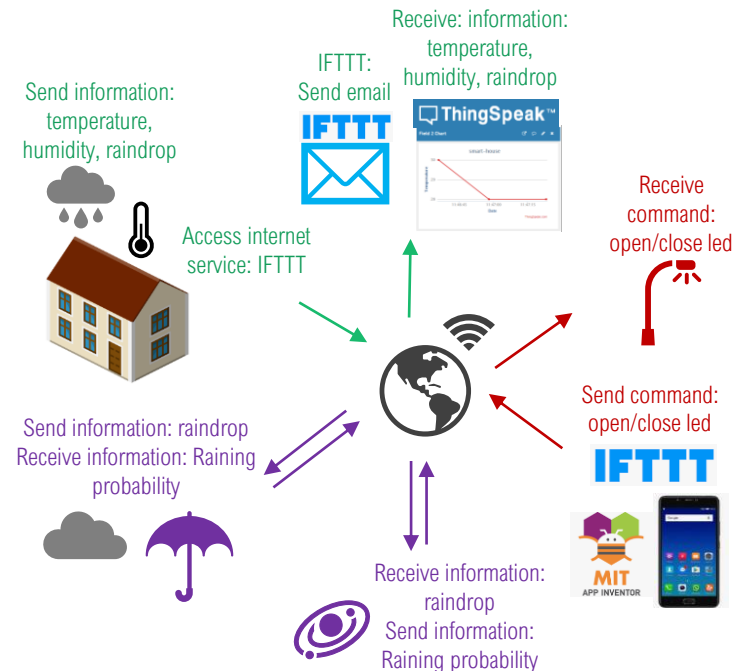
Thing that wait for the command for an action

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

---

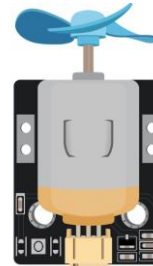
## **SMARTHON IOT BIT:**

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



## **MOTOR FAN:**

It converts electrical energy into mechanical energy. When providing power to the motor, the axis will start spin to drive the connected components.



## **ULTRASONIC DISTANCE SENSOR:**

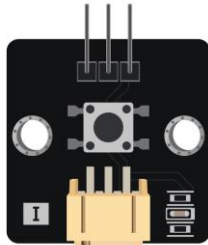
It has two ultrasonic sensors to detect the distance between current position and object.



(1) ULTRASONIC DISTANCE SENSOR

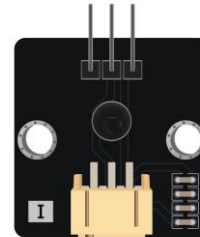
## **BUTTON:**

It is a sensor that allow user to physical switch the circuit open or close, so it is commonly use as control component.



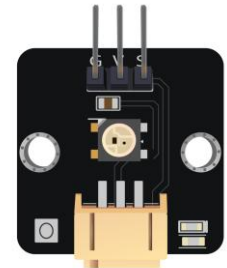
## **FLAME SENSOR:**

It uses the infrared(IR) sensor which sensitive to the spectrum of flame to detect the flame source.



## **MULTI-COLOR LED (WS2812B):**

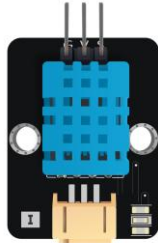
It is an integrated LED chip which can create different RGB color by programming



# PART LIST

## TEMPERATURE AND HUMIDITY SENSOR (DHT11):

It contains two components to detect humidity and temperature



## PIR MOTION SENSOR:

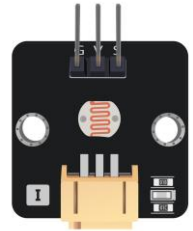
It use a pyroelectric infrared sensor to detect any moving object at front.



(1) PIR MOTION SENSOR

## LIGHT SENSOR:

It uses one photoresistor to detect the luminance of environment



## 360° SERVOS:

It is a tiny and lightweight motor which can perform rotation control (360° servo).



(1) 360° SERVO

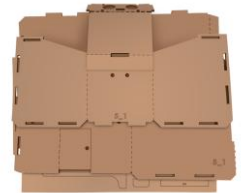
## 180° SERVOS:

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



## 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 · (1) BLU TACK · (1) RUBBER BEND · (1) PAPER CURTAIN · (1) CURTAIN ROD · (2) CROCODILE CLIP · (1) USB CABLE · (1) BATTERY HOLDER ·

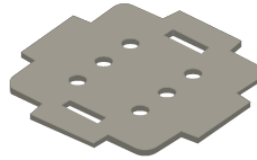
# MODEL LIST

---

## **MODEL F:**

It is attached motor fan to the Model F using M4 \* 10mm screws and nuts.

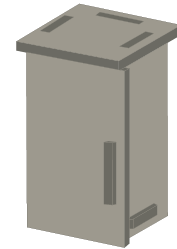
To build a fan stand, put Model F into the B3, B4 model.



Fan Stand

## **MODEL G:**

It is a fridge that has an extra sensor to help it determine the door close state, when the user forgets to close the door, it will warn the user to close it to reduce waste of energy and prevent food poisoning.



Fridge

## **MODEL H:**

It is installed as a decoration display on model B3.



Decoration display

## **MODEL I:**

It is a flat surface furniture which is placing on the living room for the decoration.



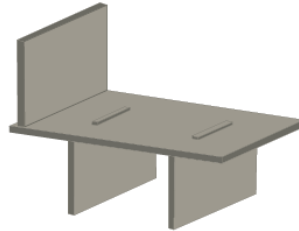
Table

# MODEL LIST

---

## **MODEL J:**

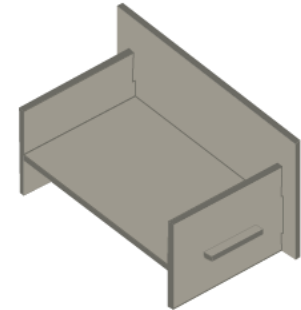
It is a large piece of furniture which is used for sleeping on the bedroom.



Bed

## **MODEL K:**

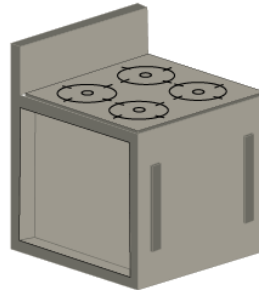
It is a sofa model which is long seat with a back and arms for decoration.



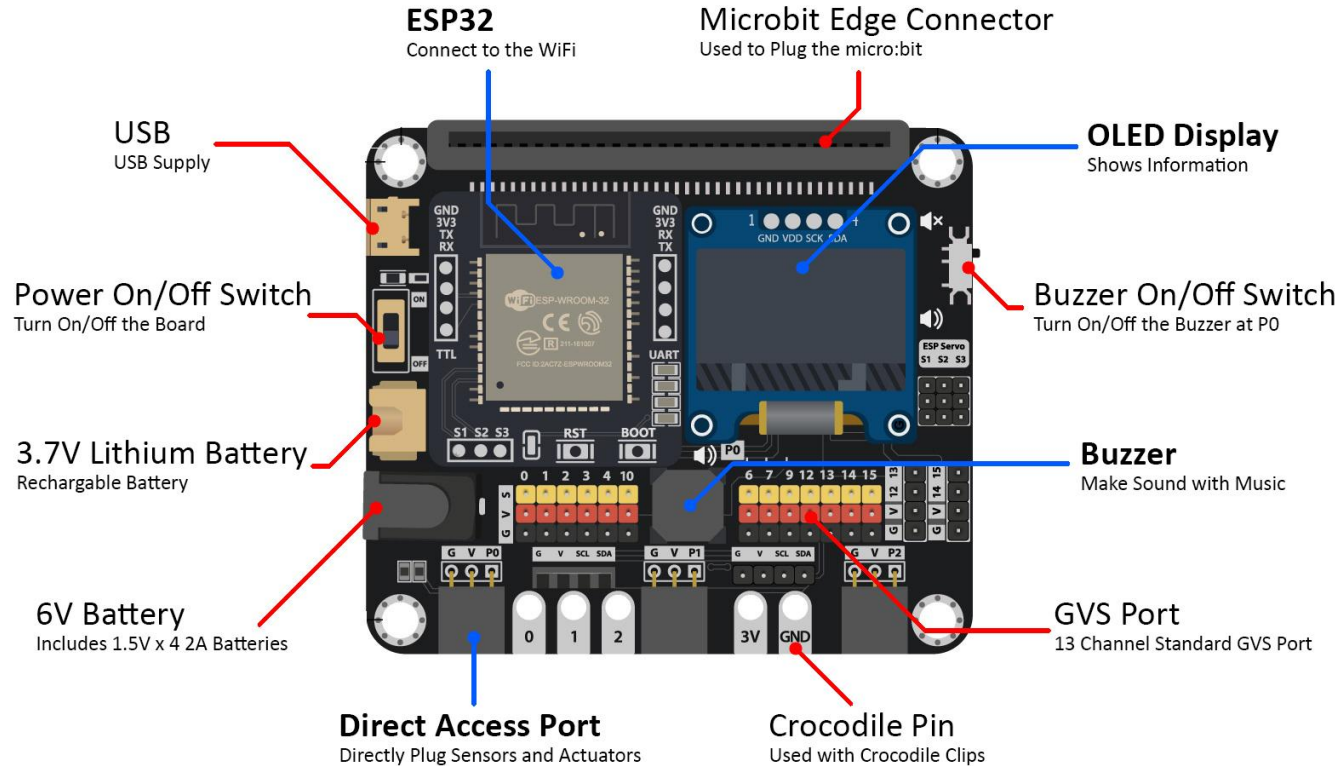
Sofa


## **MODEL L:**

It is installed an IOT system into cooking stove in the kitchen or living room to alert the user on his/her phone when he/she is sleeping or in the office to prevent accident happened.



Cooking stove



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

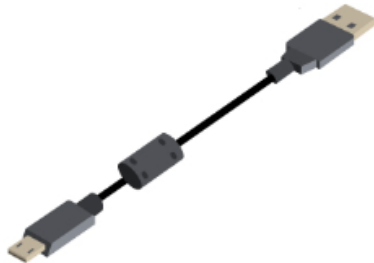
# POWER SUGGESTION

---

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

## OPTION 1

- USB



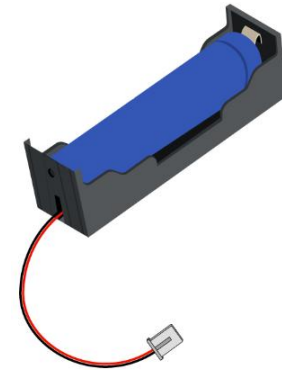
## OPTION 2

- 4 XAA Batteries (Battery Not Included)



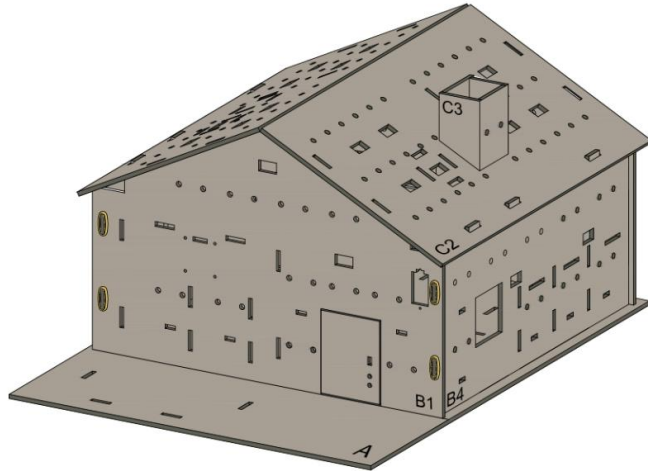
## OPTION 3

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

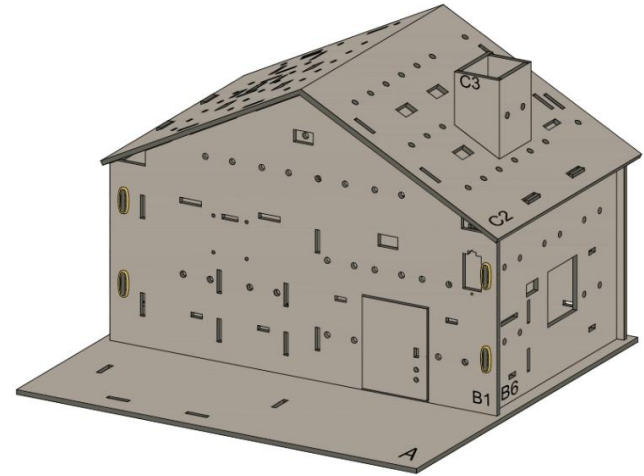


# HOW TO BUILD A HOUSE MODEL?

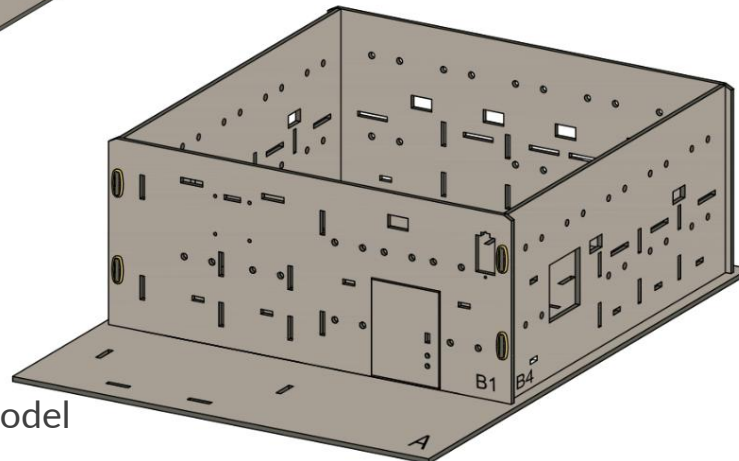
There are 3 type of house you can build



1. Big House Model



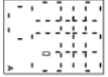




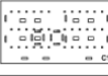
2. Small House Model

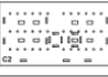


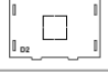





3. Open Style Model

# BIG HOUSE MODEL

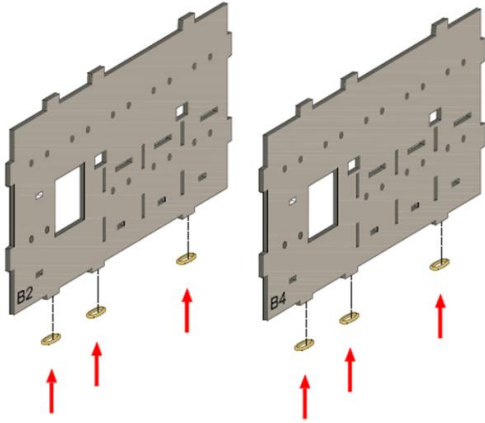
## Part list

NO.	CODE	DESCRIPTION	QTY	IMAGE
1.	A	House Base	X 1	
2.	B1	House Body 1	X 1	
3.	B2	House Body 2	X 1	
4.	B3	House Body 3	X 1	
5.	B4	House Body 4	X 1	
6.	C1	House Roof 1	X 1	

NO.	CODE	DESCRIPTION	QTY	IMAGE
7.	C2	House Roof 2	X 1	
8.	C3	House Roof 3	X 1	
9.	D1	Extra Room 1	X 1	
10.	D2	Extra Room 2	X 1	
11.	D3	Extra Room 3	X 1	
12.	D4	Extra Room 4	X 1	
13.	RB	Rubber Bend	X 28	

# ASSEMBLY STEP

Step 1. Add the 6 rubbers bends to the bottom of the model B4 and B2.

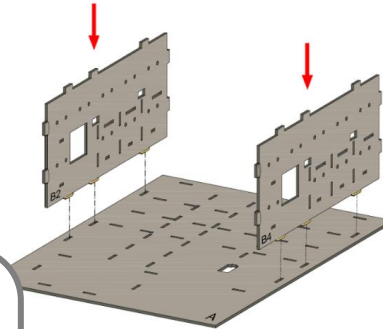


Part list:

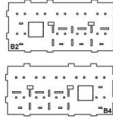
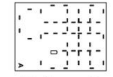


• Rubber Bend x6

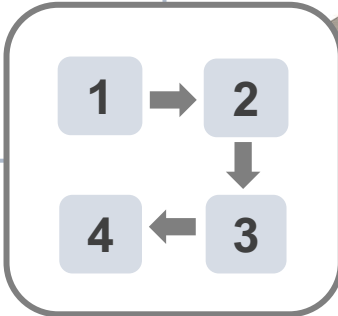
Step 2. Insert the model B2 and B4 to model A.



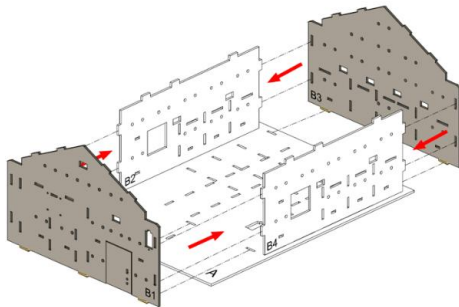
Part list:



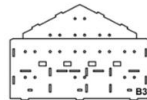
- Model A
- Model B2
- Model B4



Step 4. Put the model B1, B3 to model A, and then insert to the model B2 and B4.



Part list:

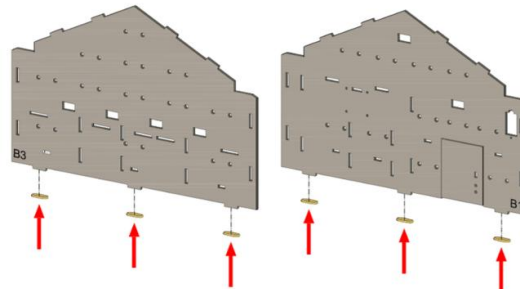


• Model B3



• Model B1

Step 3. Add the 6 rubbers bends to the bottom of the model B1 and B3.



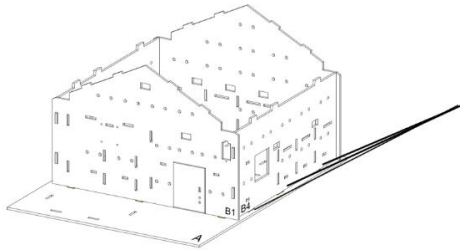
Part list:



• Rubber Bend x6

# ASSEMBLY STEP

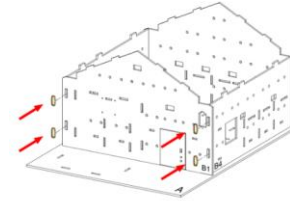
Step 5. Make sure all the holes of model A wrap the rubber from the model B1 to B4 properly and completely.



Make sure the holes of model A wrap the rubber properly and completely.

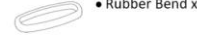


Step 6. Add the 4 rubbers bends to the front side (Model B1) and back side (Model B3) of the house.



Part list:

• Rubber Bend x8



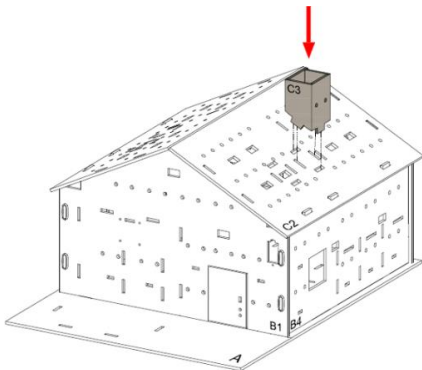
5

6

8

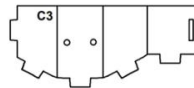
7

Step 8. Insert the model C3 on the model C2.

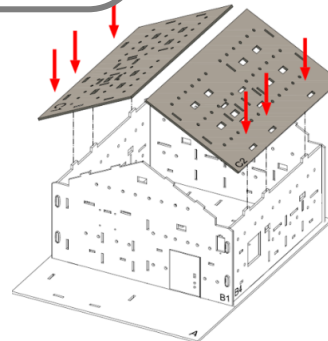


Part list:

• Model C3



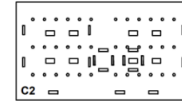
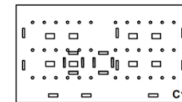
Step 7. Insert the model C1 and C2 on model B1, B2, B3, B4, align with the holes on them.



Part list:

• Model C1

• Model C2

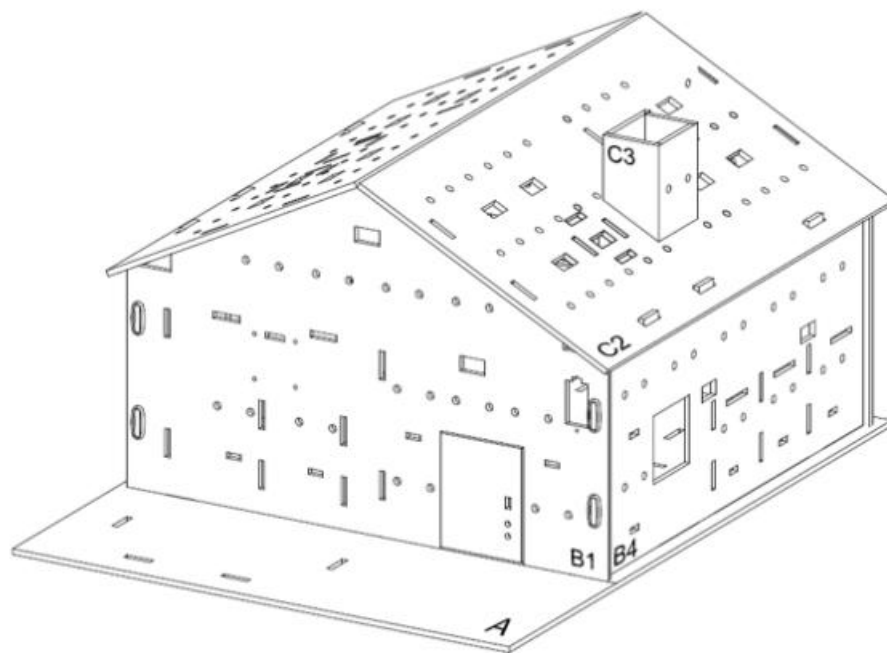


# ASSEMBLY STEP

---

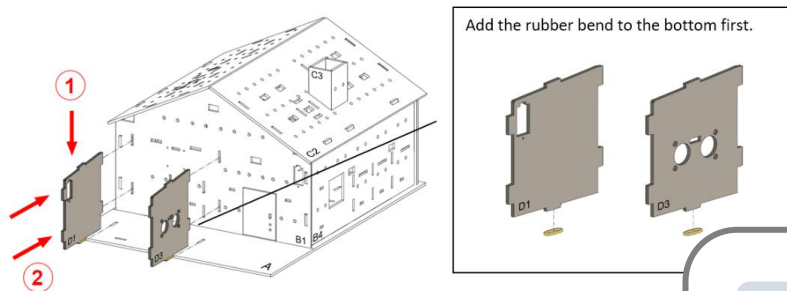
Step 9

Completed!

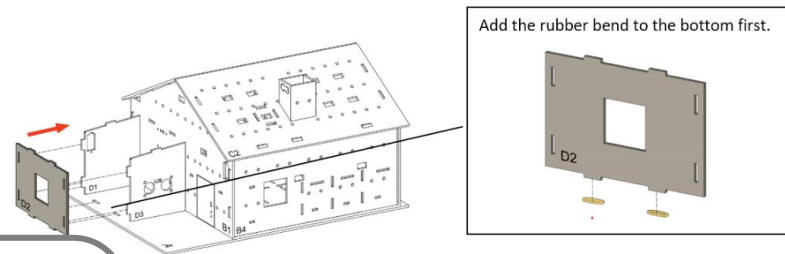


# OPTIONAL: EXTRA ROOM

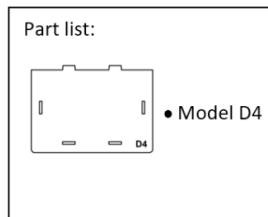
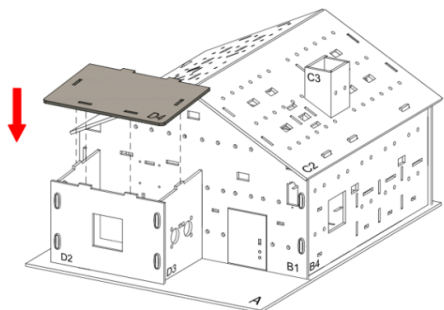
Step 1. For extra room installation, Insert the model D1, D3 on model B1 align with the holes on model A.



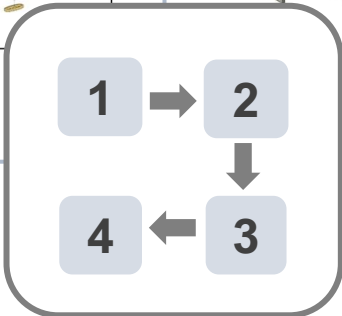
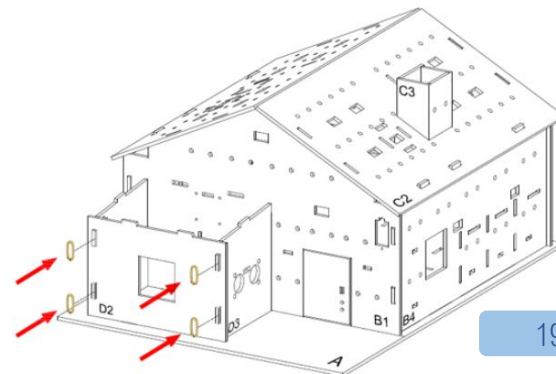
Step 2. Put the model D2 on the holes of model A, and then insert it to model D1 and D3.



Step 4. Insert the model D4 to the model B1, and then attach it to the model D1, D2 and D3.



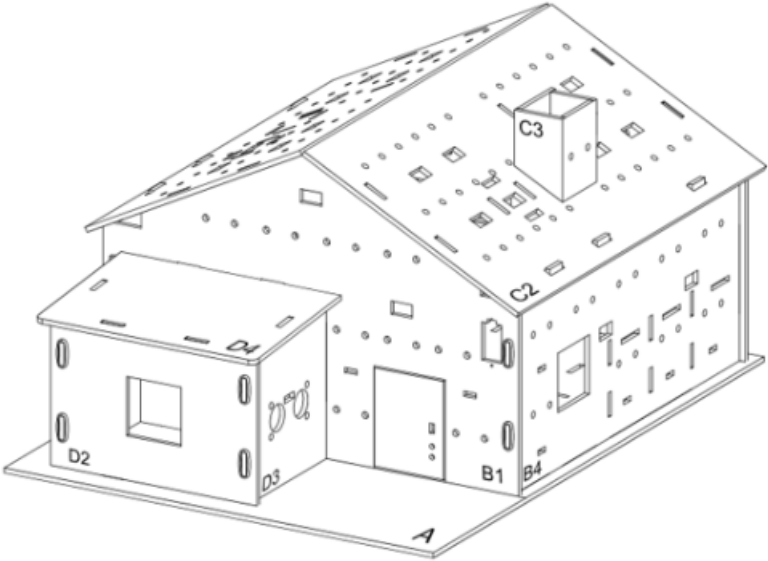
Step 3. Add the 4 rubbers bends to the model D2 so that the house will not loose.



# OPTIONAL: EXTRA ROOM

---

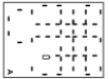
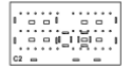


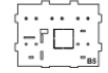





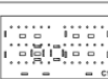


Step 5  
Completed!



# SMALL HOUSE MODEL

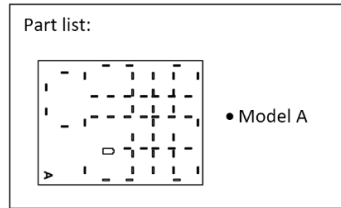
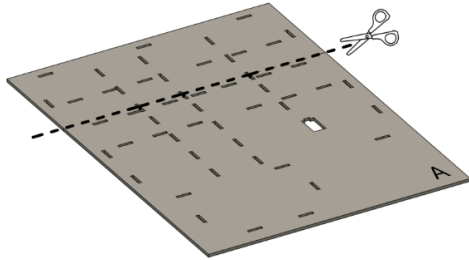
Before we build a small house model, we need to cut the cardboard as instruction on cardboard.

## Part list

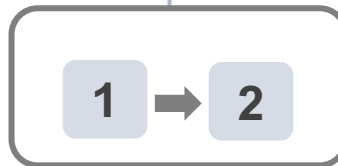
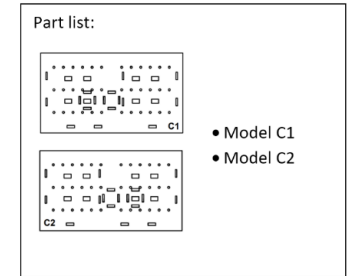
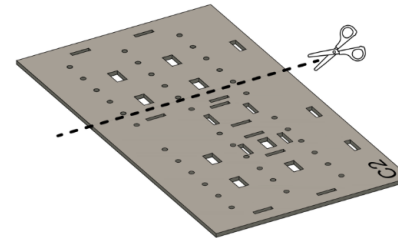
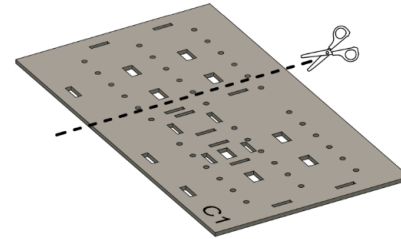
NO.	CODE	DESCRIPTION	QTY	IMAGE	NO.	CODE	DESCRIPTION	QTY	IMAGE
1.	A	House Base	X 1		7.	C2	House Roof 2	X 1	
2.	B1	House Body 1	X 1		8.	C3	House Roof 3	X 1	
3.	B5	Small House Body 2	X 1		9.	D1	Extra Room 1	X 1	
4.	B3	House Body 3	X 1		10.	D2	Extra Room 2	X 1	
5.	B6	Small House Body 4	X 1		11.	D3	Extra Room 3	X 1	
6.	C1	House Roof 1	X 1		12.	D4	Extra Room 4	X 1	
					13.	RB	Rubber Bend	X 22	

# Cut the cardboard

Step 1. Cutting with the dashed line on A cardboard.

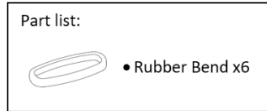
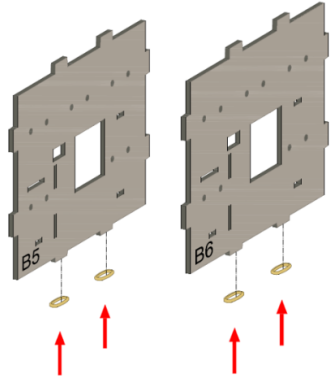


Step 2. Cutting with the dashed line on C1 and C2 cardboard.

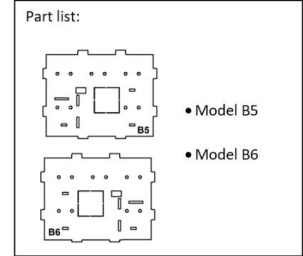
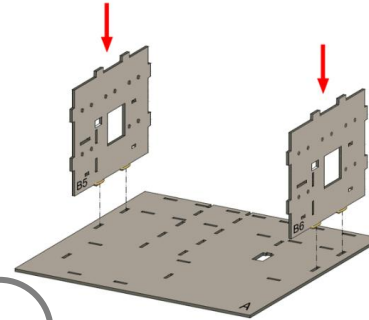


# ASSEMBLY STEP

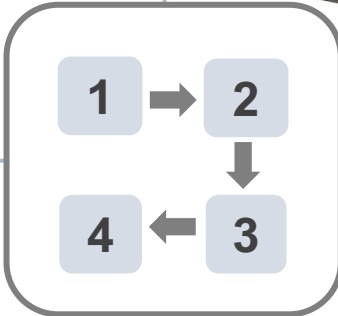
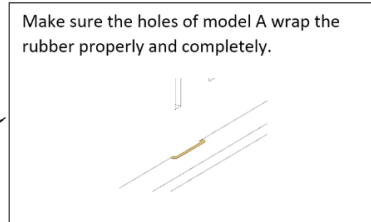
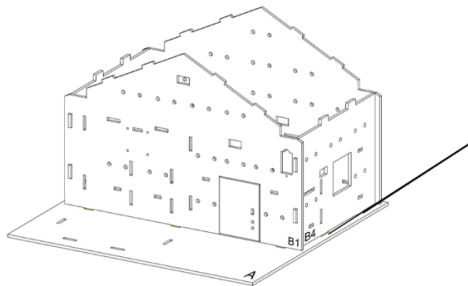
Step 1. Add the 6 rubbers bends to the bottom of the model B5 and B6.



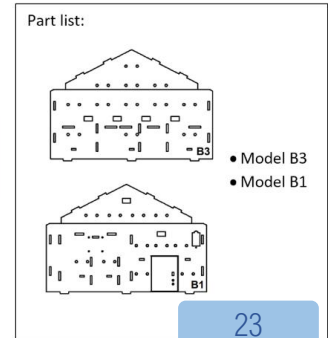
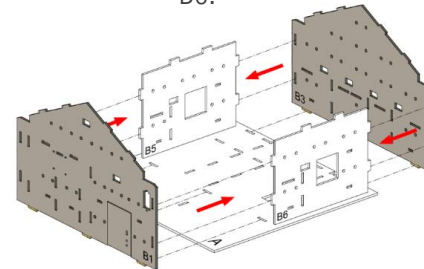
Step 2. Insert the model B5 and B6 to model A.



Step 4. Make sure all the holes of model A wrap the rubber from the model B1 to B4 properly and completely.

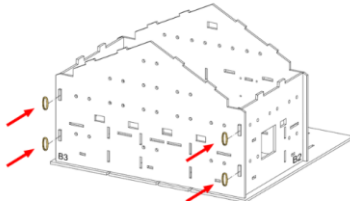
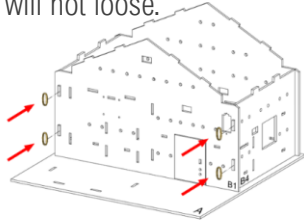


Step 3. Put the model B3 to model A, and then insert to the model B5 and B6. put the model B1 on the model A, and then insert it to the model B5 and B6.

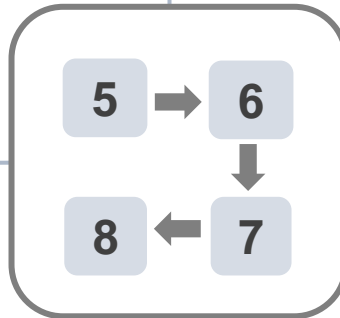
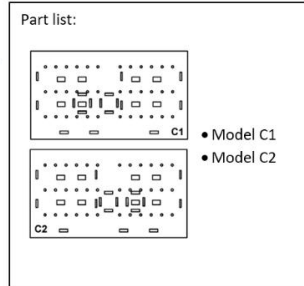
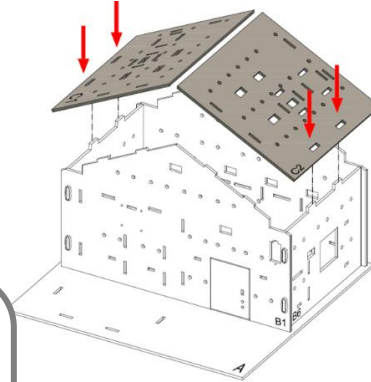


# ASSEMBLY STEP

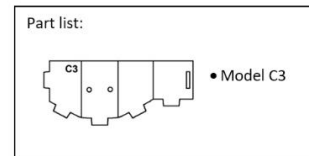
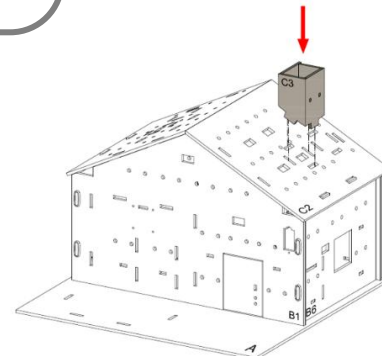
Step 5. Add the 8 rubber bends to the model B1 and B3 so that the house will not loose.



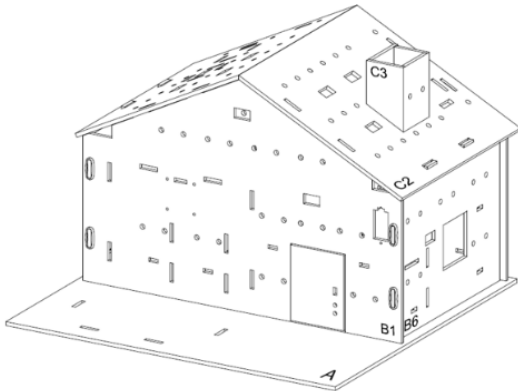
Step 6. Insert the model C1 and C2 on model B1, B5, B3, B6, align with the holes on them.



Step 7. Insert model C3 on the mode C2.

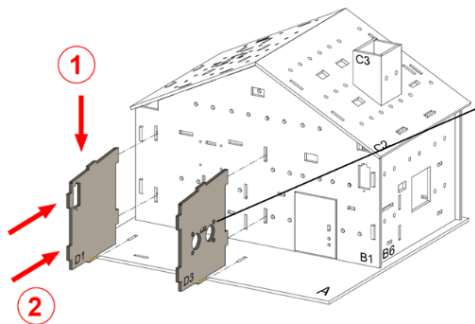


Step 8. Completed!

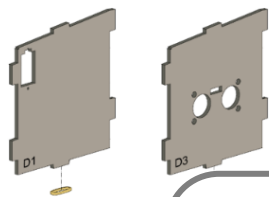


# OPTIONAL: EXTRA ROOM

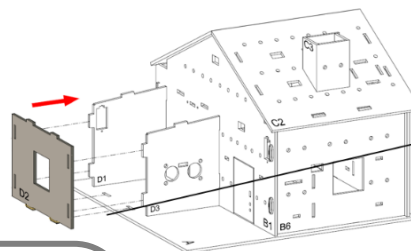
Step 1. For extra room installation, Insert the model D1, D3 on model B1, align with the holes on model A.



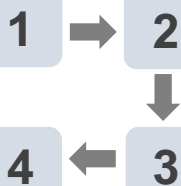
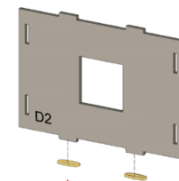
Add the rubber bend to the bottom first.



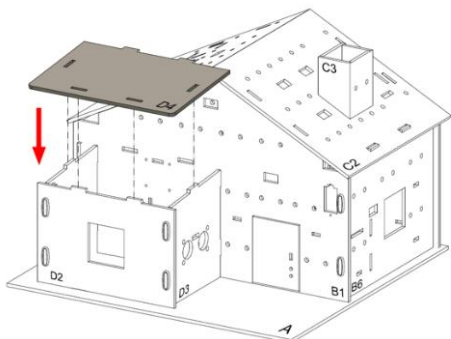
Step 2. Put the model D2 on the holes of model A, and then insert it to model D1 and D3.



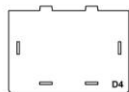
Add the rubber bend to the bottom first.



Step 4. Insert the model D4 to the model B1, and then attach it to the model D1, D2 and D3.

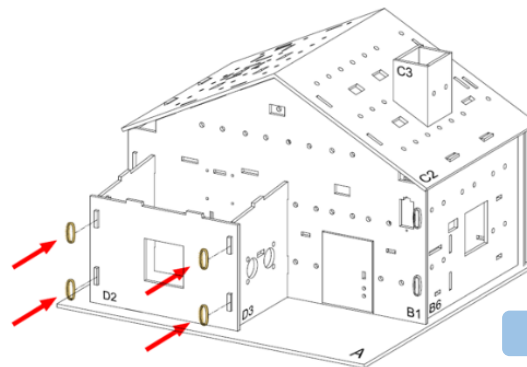


Part list:



• Model D4

Step 3. Add the 4 rubbers bends to the model D2 so that the house will not loose.

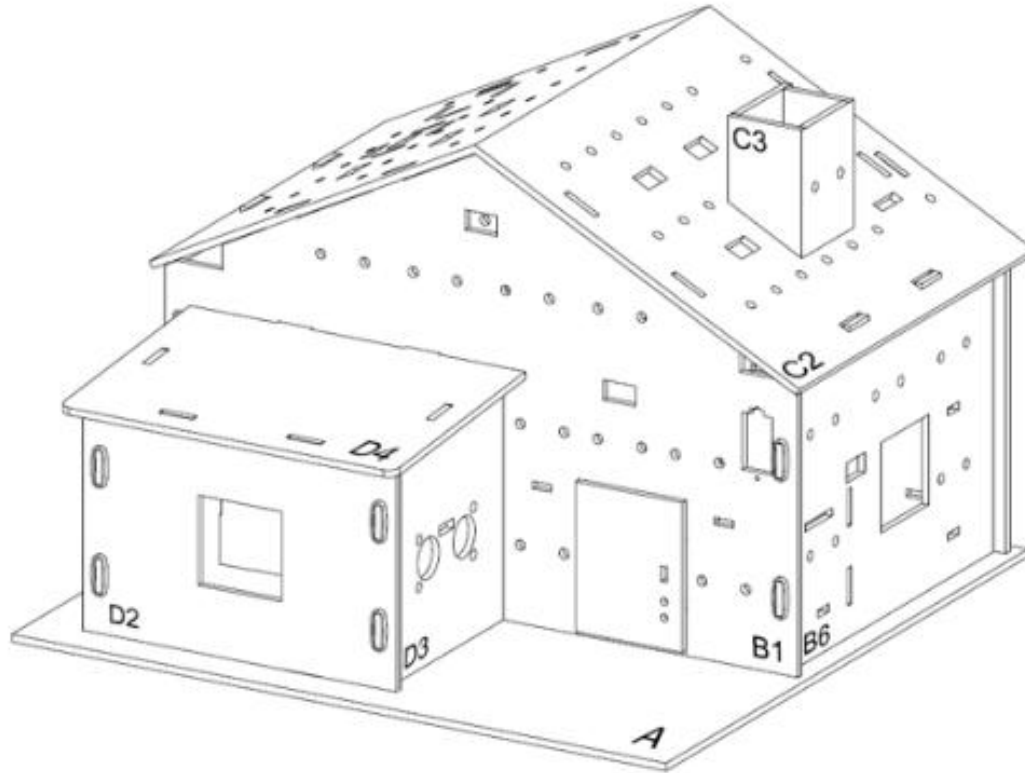


# OPTIONAL: EXTRA ROOM

---

Step 5

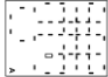
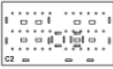











Completed!



# OPEN STYLE MODEL

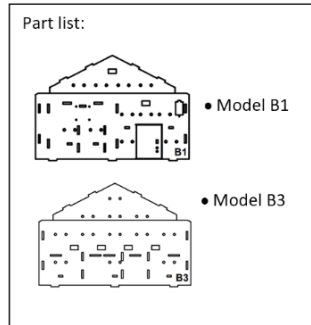
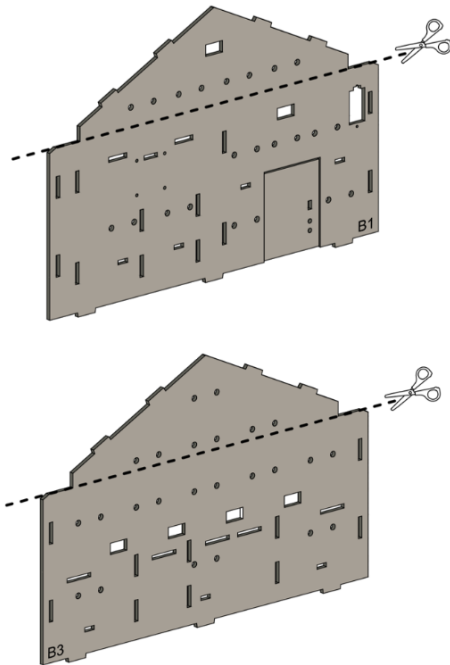
Before we build a open model, we need to cut the cardboard as instruction on cardboard.

## Part list

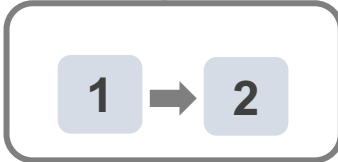
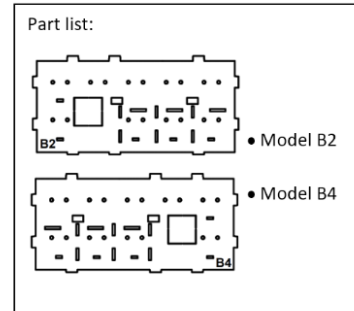
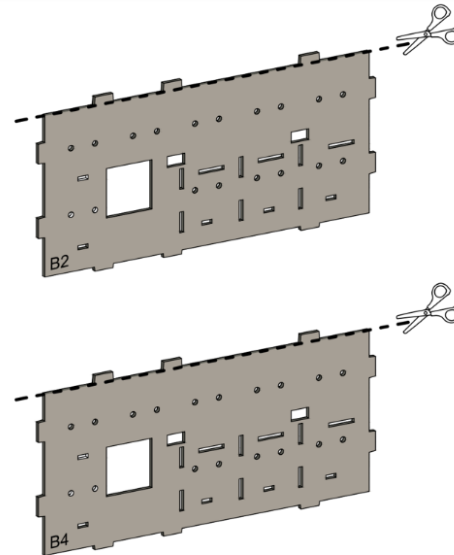
NO.	CODE	DESCRIPTION	QTY	IMAGE	NO.	CODE	DESCRIPTION	QTY	IMAGE
1.	A	House Base	X 1		7.	C2	House Roof 2	X 1	
2.	B1	House Body 1	X 1		8.	C3	House Roof 3	X 1	
3.	B2	House Body 2	X 1		9.	D1	Extra Room 1	X 1	
4.	B3	House Body 3	X 1		10.	D2	Extra Room 2	X 1	
5.	B4	House Body 4	X 1		11.	D3	Extra Room 3	X 1	
6.	C1	House Roof 1	X 1		12.	D4	Extra Room 4	X 1	
					13.	RB	Rubber Bend	X 28	

# Cut the cardboard

Step 1. Cutting with the dashed line on B1 and B3 cardboard.

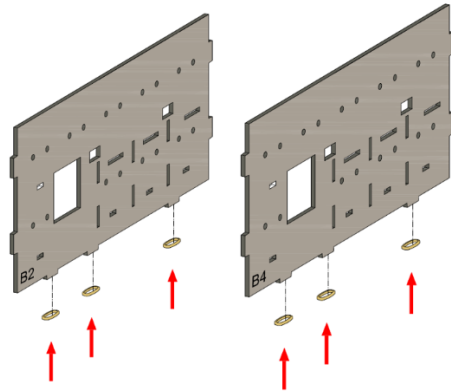


Step 2. Cutting with the dash line on B2 & B4 cardboard.




# ASSEMBLY STEP

Step 1. Add the 6 rubbers bends to the bottom of the model B4 and B2.

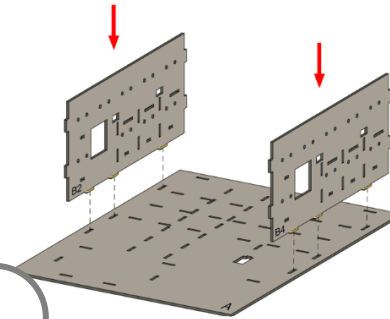


Part list:

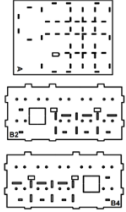


- Rubber Bend x6

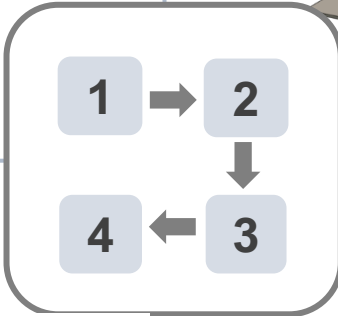
Step 2. Insert the model B2 and B4 to model A.



Part list:

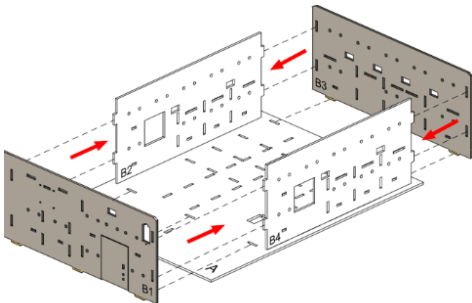


- Model A
- Model B2
- Model B4

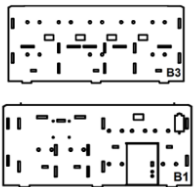


Step 4. Put the model B3 on the model A, and then insert to the model B2 and B4. Put the model B1 on the model A, and then insert it to the model B2 and B4.

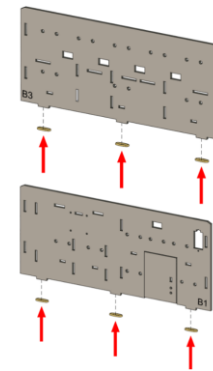
Step 3. Add the 6 rubber bends to the bottom of the model B1 and B3.



Part list:



- Model B3
- Model B1



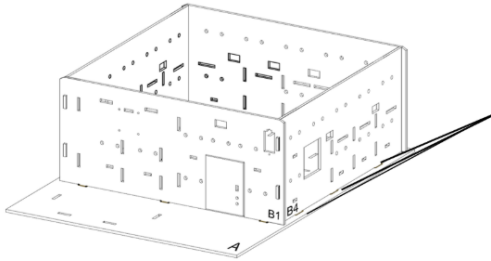
Part list:



- Rubber Bend x6

# ASSEMBLY STEP

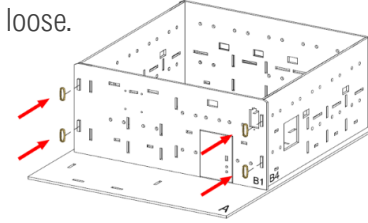
Step 5. Make sure all the holes of model A wrap the rubber from the model B1 to B4 properly and completely.



Make sure the holes of model A wrap the rubber properly and completely.



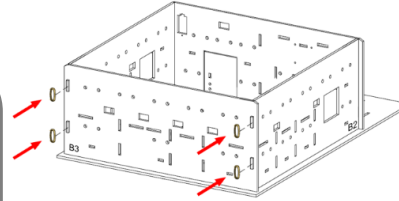
Step 6. Add the 4 rubber bends to the model B1 so that the house will not loose.



Part list:



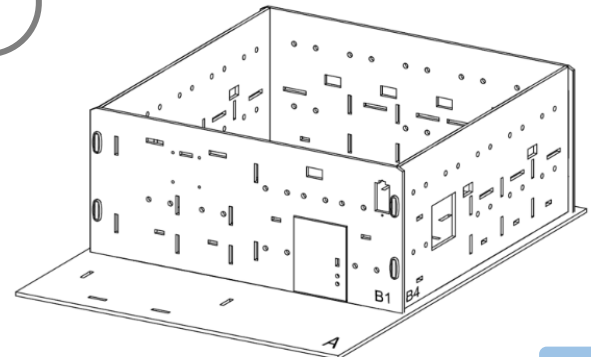
• Rubber Bend x8



5 → 6

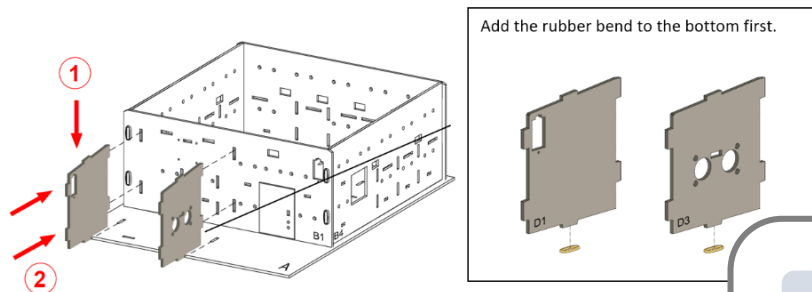
↓  
7

Step 7. Completed!

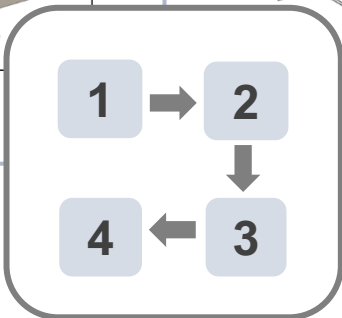
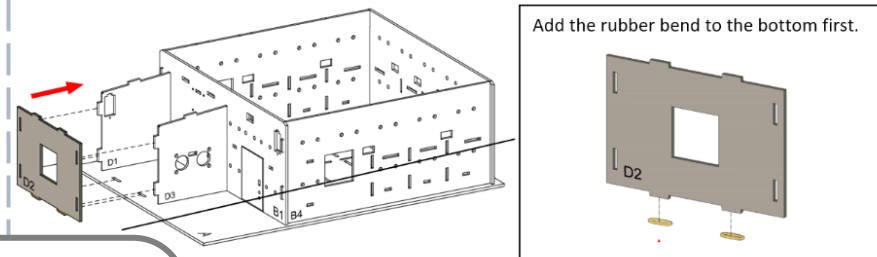


# OPTIONAL: EXTRA ROOM

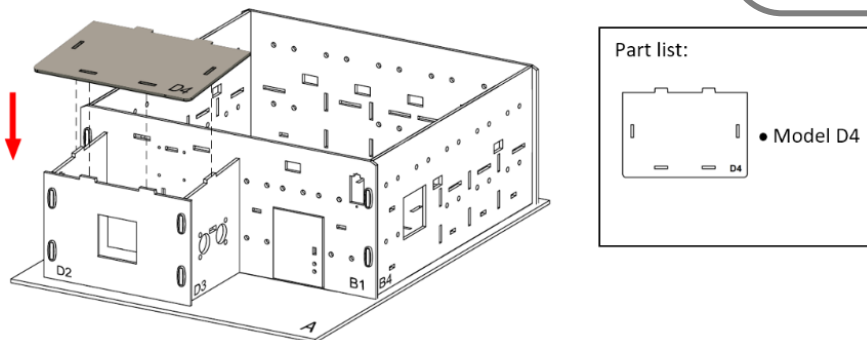
Step 1. For extra room installation, Insert the model D1, D3 on model B1, align with the holes on model A.



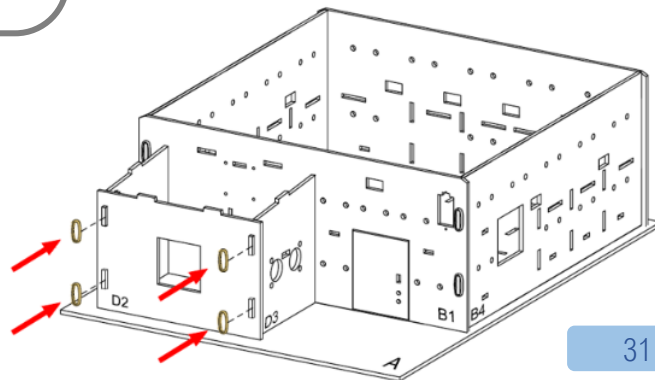
Step 2. Put the model D2 on the holes of model A, and then insert it to model D1 and D3.



Step 4. Insert the model D4 to the model B1, and then attach it to the model D1, D2 and D3.



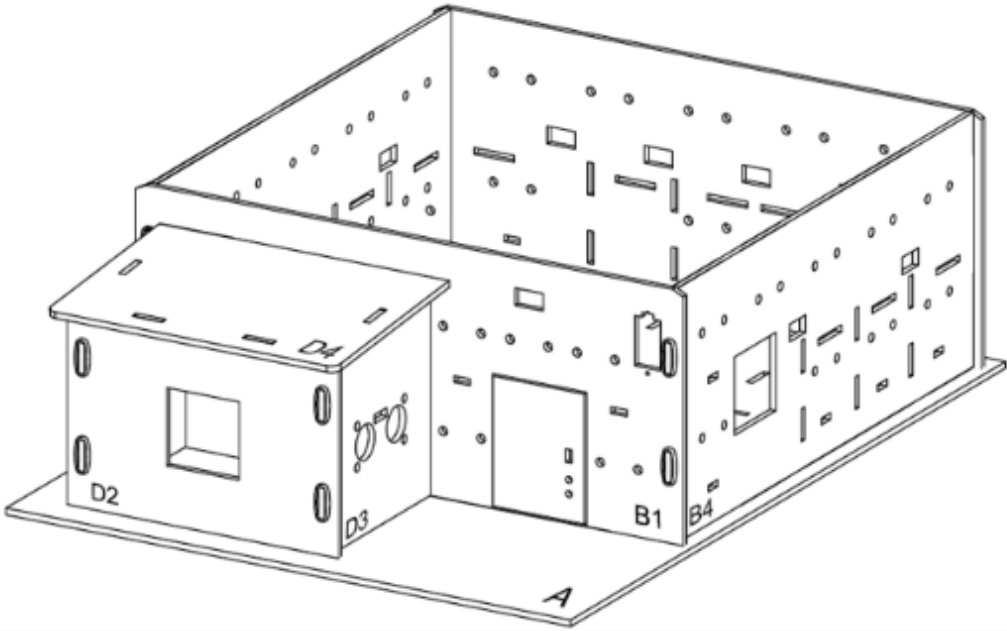
Step 3. Add the 4 rubbers bends to the model D2 so that the house will not loose.



# OPTIONAL: EXTRA ROOM

---

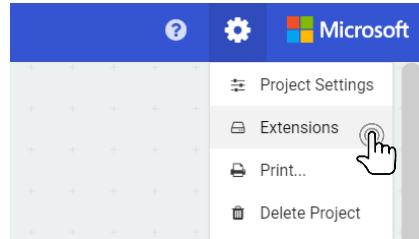
Step 5  
Completed!



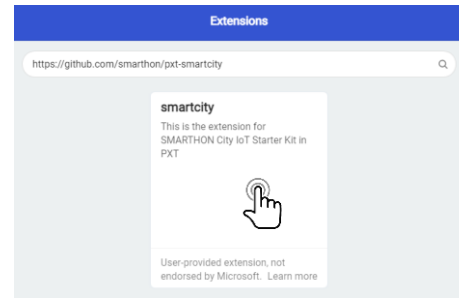
# GETTING STARTED: ADD THE EXTENSION ON MICRO:BIT

---

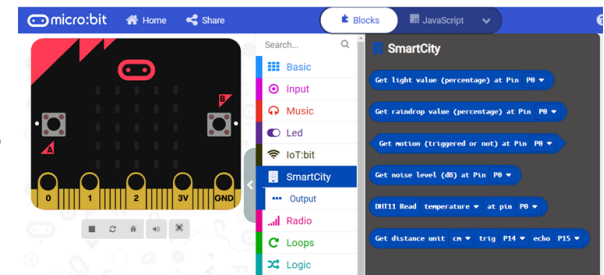
1. Go to <https://makecode.microbit.org/>,  
create a new project



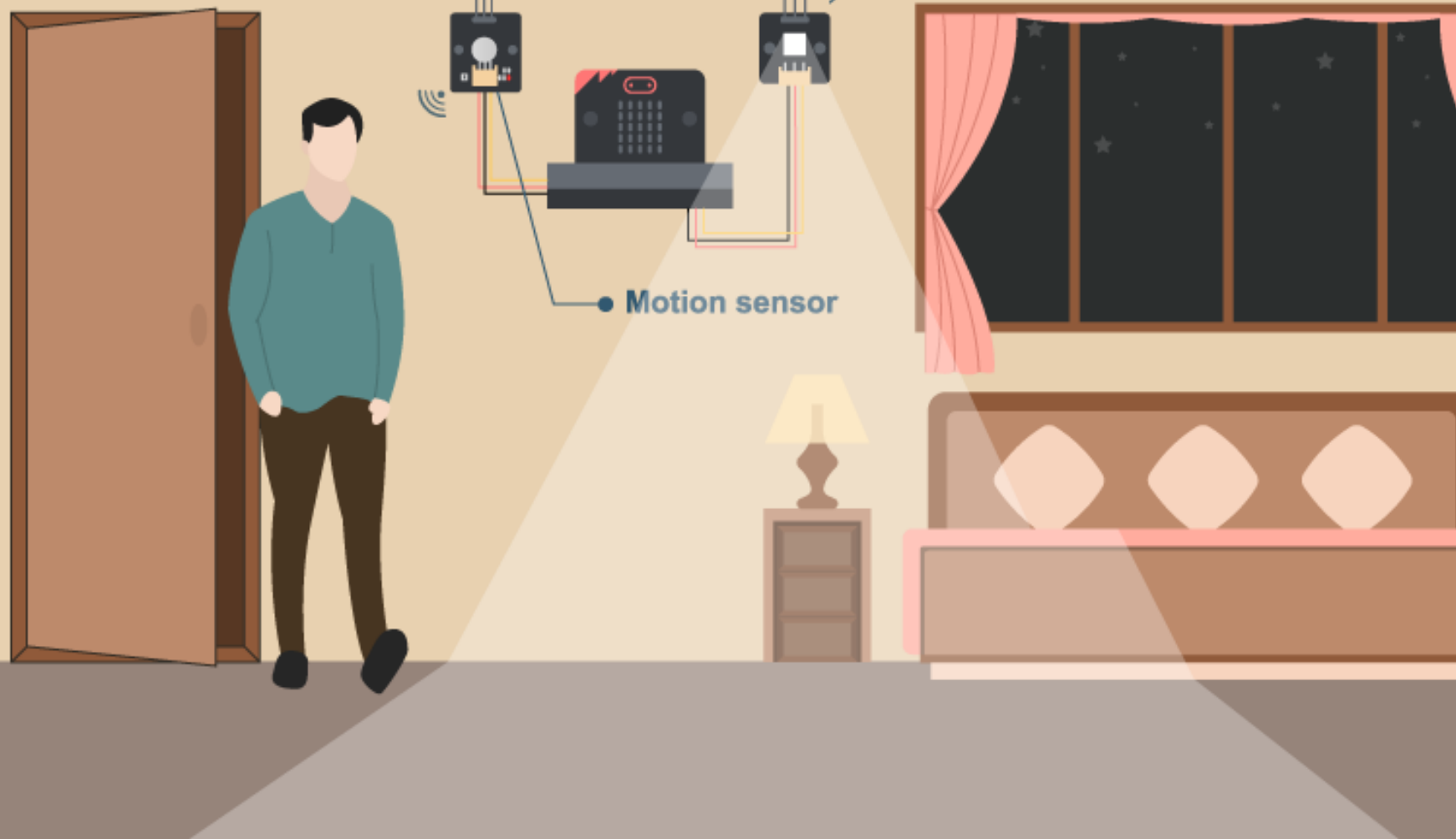
2. Search “<https://github.com/SMARTHON/pxt-smarthome>”  
and select “smarthome”



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



## ● Smart Saving Light Bulb



● Multi-colour LED

● Motion sensor

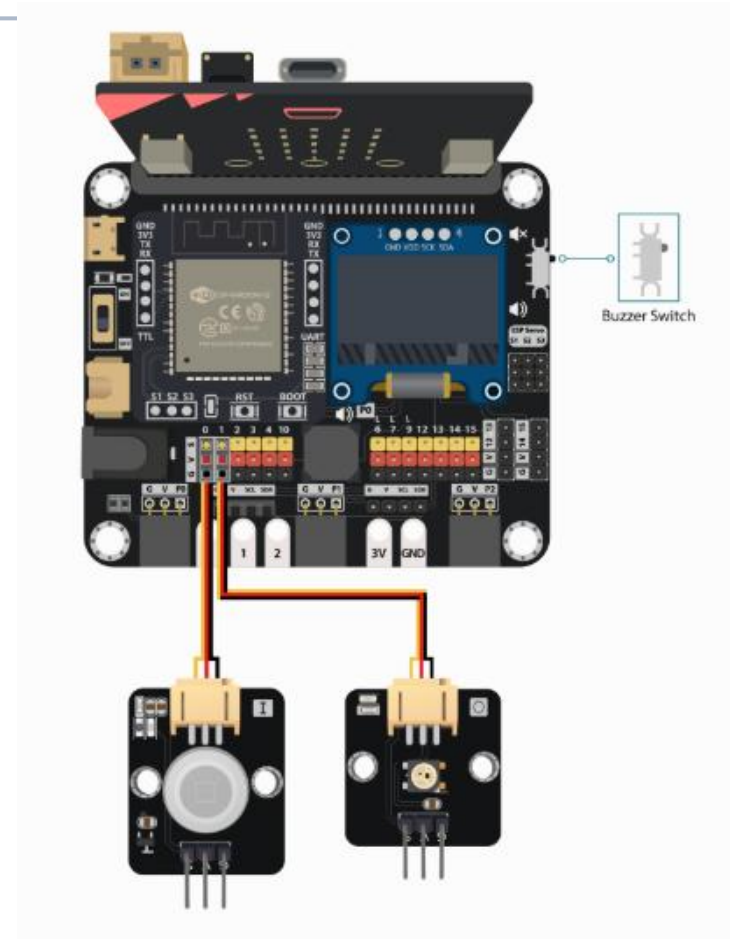
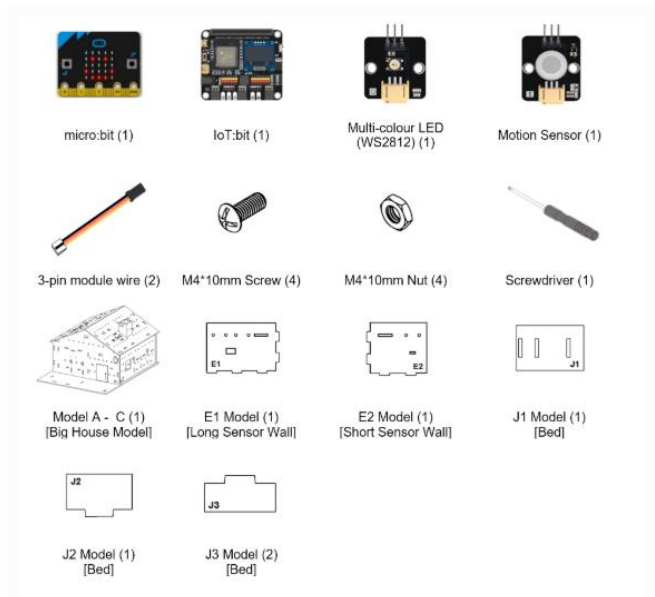
# 1. Smart Saving Light Bulb

LEVEL: ★ ☆ ☆ ☆ ☆

## Introduction

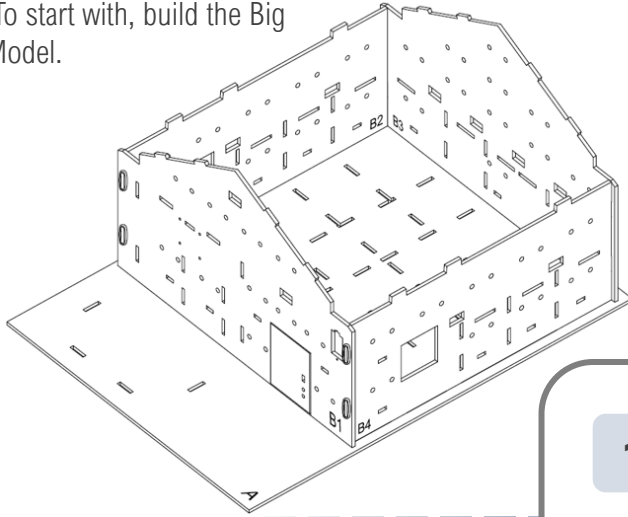
Smart light bulb is a smart light bulb that helps automation and saving electricity. When there is people coming in, the bulb will be turned on automatically. People always forget to turn off the light when he/she leave. After 10 mins, if there are no any motions inside the room, it will close the LED to save energy.

## PART LIST



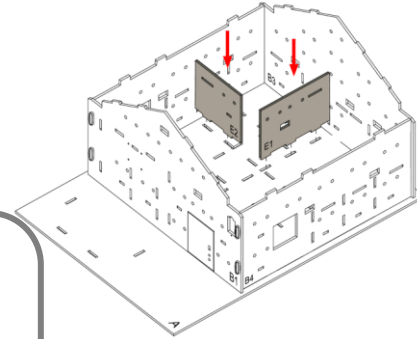
# ASSEMBLY STEP

Step 1. To start with, build the Big House Model.

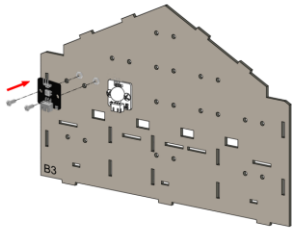
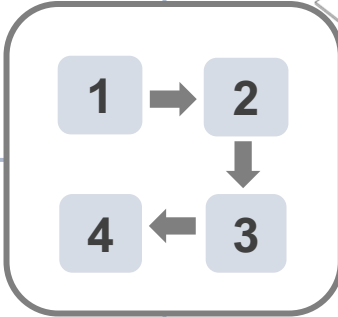
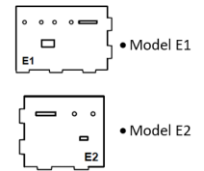


Step 2

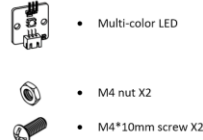
Pick one corner of the house. In this example, we pick the left upper corner, install the 2 walls (E1 & E2) to build a small room.



Part list:

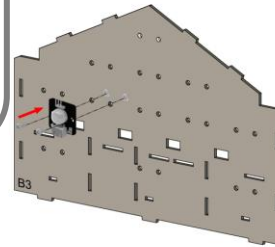


Part list:

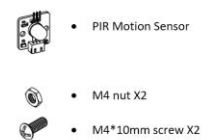


Step 4

Attach the Multi-color LED to the model B3 using M4 \* 10mm screws and nuts.



Part list:

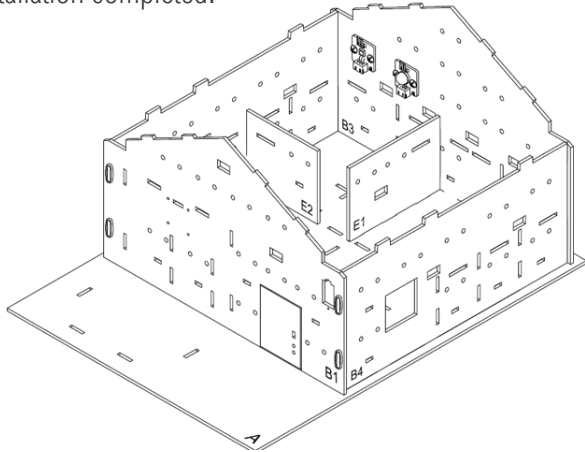


Step 3

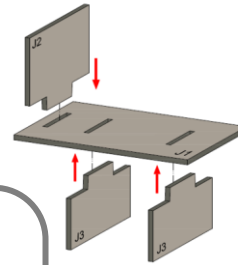
Attach the PIR Motion Sensor to model B3 using M4 \* 10mm screws and nuts.

# ASSEMBLY STEP

Step 5  
Installation completed!

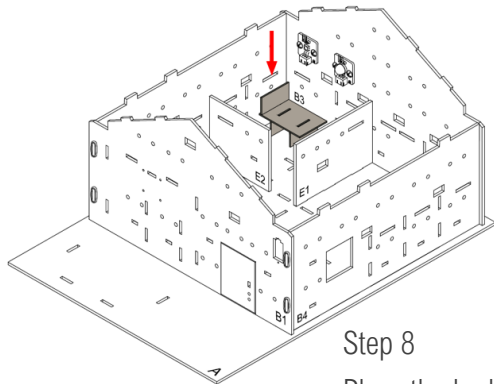
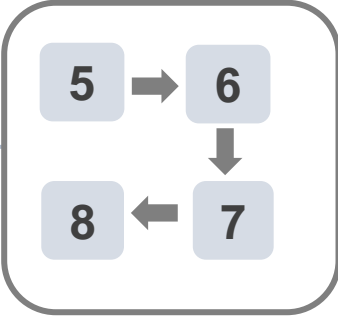


Step 6  
To build a Bed. Put model J2 onto the model J1, then put together all the cardboard parts (J1-J3).

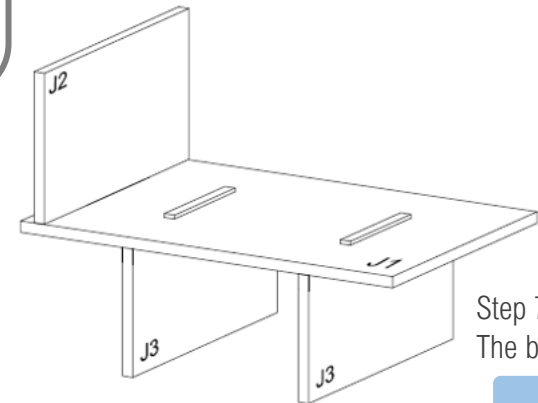


Part list:

- Model J1 X1
- Model J2 X1
- Model J3 X2



Step 8  
Place the bed inside the bedroom.



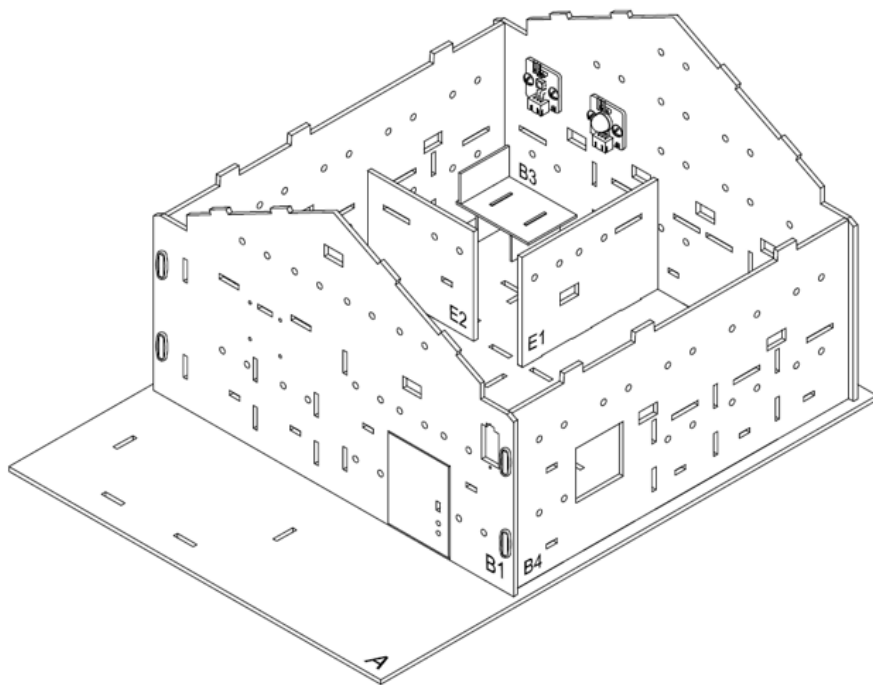
Step 7  
The bed completed!

# ASSEMBLY STEP

---

Step 9

Assembly completed!



# CODING



When the people are moving in the room, the motion sensor will trigger and keep the LED turned on. When there is no one moving, the LED will turn off.

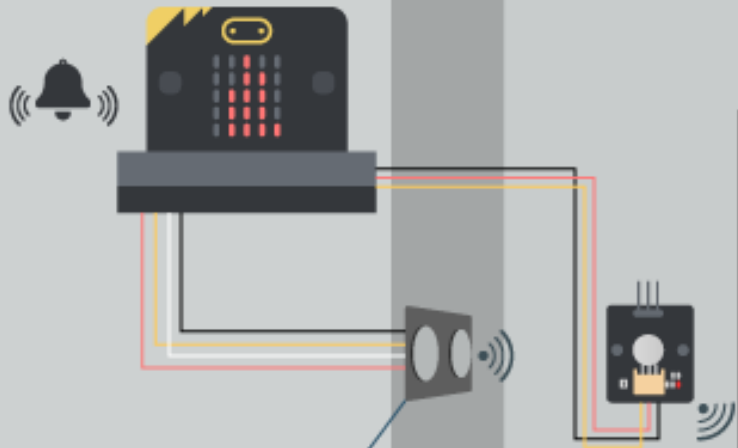
The image shows a Scratch code editor with the following blocks:

- 1 on start** (blue block)
- 2 set strip to NeoPixel at pin P1 with 1 leds as RGB (GRB format)** (red block)
- 3 forever** (blue loop block)
- 4 if Get motion (triggered or not) at Pin P0 = true then** (teal block)
- strip show color white** (red block)
- 5 pause (ms) 10000** (blue block)
- 6 else strip show color black** (teal block)

## STEPS:

1. Block **on start** only runs at the beginning
2. **Set strip to NeoPixel at pin P1 with 1 leds as RGB (GRB format)**
3. Block **forever** runs as a loop
4. If **Get motion (triggered or not) at Pin P0 = true** then, **strip** show color **white**.
5. **Pause for 10 seconds**.
6. Else, **strip** show color **black**.

# ● Security Vanguard Alarm System



Ultrasonic Distance Sensor

Motion sensor



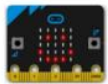
# 2. SECURITY VANGUARD ALARM SYSTEM

LEVEL: ★ ★ ☆ ☆ ☆

## Introduction

Security vanguard alarm system is a smart defense system that triggers an alert automatically when detecting an unexpected visit. It is good to have security system to prevent from the theft coming in.

## PART LIST



micro:bit (1)



IoT:bit (1)



Motion Sensor (1)



Distance Sensor (1)



3-pin module wire (1)



4-pin module wire (1)



M4\*10mm Screw (6)



M4\*10mm Nut (6)



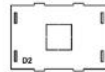
Screwdriver (1)



Model A - C (1)  
[Big House Model]



Model D1 (1)  
[Extra Room]



Model D2 (1)  
[Extra Room]



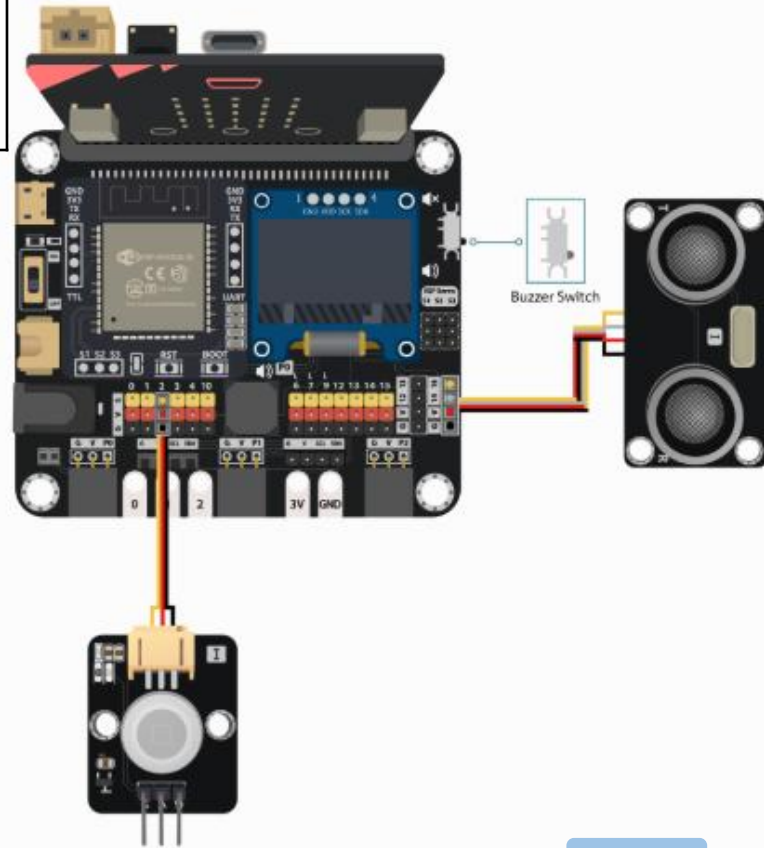
Model D3 (1)  
[Extra Room]



Model D4 (1)  
[Extra Room]



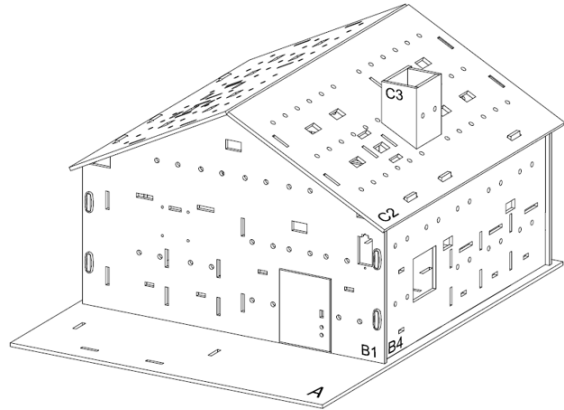
Rubber Bend (8)



# ASSEMBLY STEP

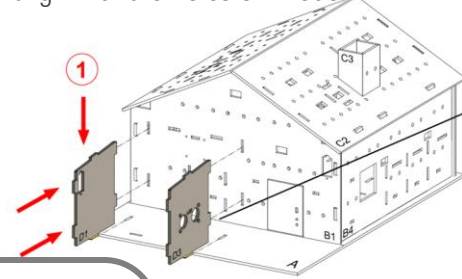
## Step 1

To start with, build the Big House Model.

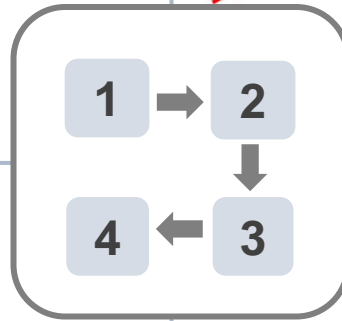
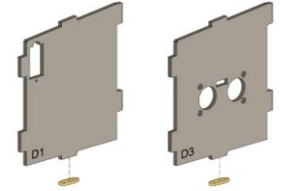


## Step 2

For extra room installation, Insert the model D1, D3 on model B1, align with the holes on model A.

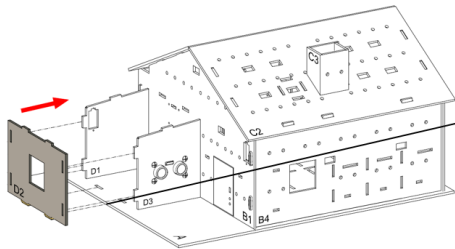


Add the rubber bend to the bottom first.

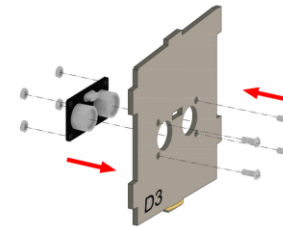
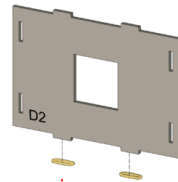


## Step 4

Put the model D2 on the holes of model A, and then insert it to model D1 and D3.



Add the rubber bend to the bottom first.

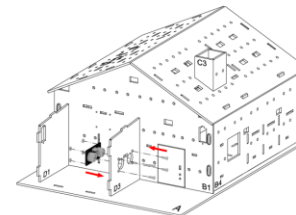


Part list:

-  Distance Sensor
-  M4 nut X4
-  M4\*10mm screw X4

## Step 3

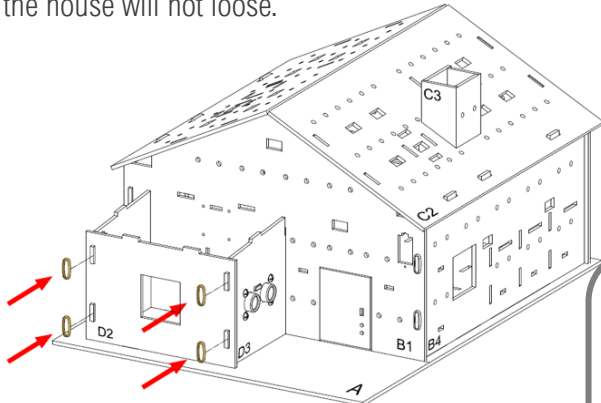
Attach the distance sensor to the Model D3 using M4 \* 10mm screws and nuts.



# ASSEMBLY STEP

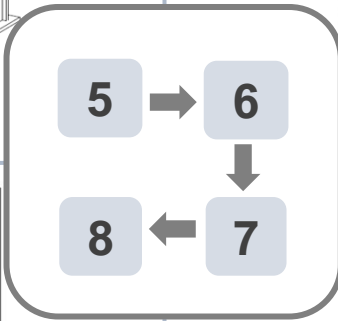
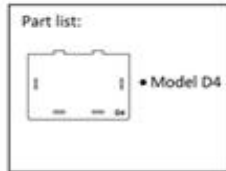
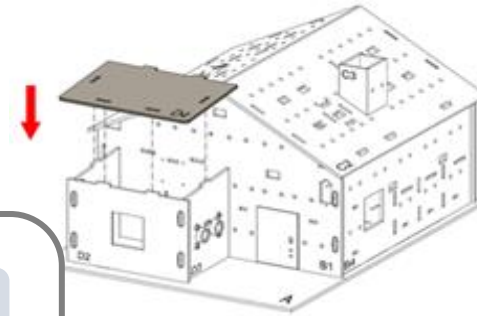
## Step 5

Add the 4 rubber bends to the model D2 so that the house will not loose.



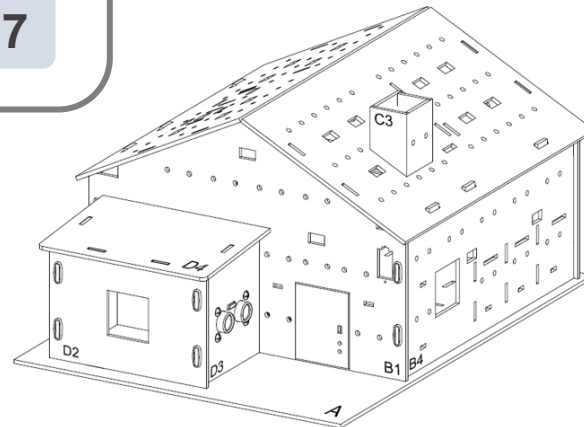
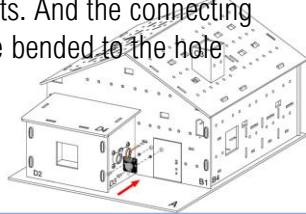
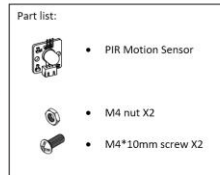
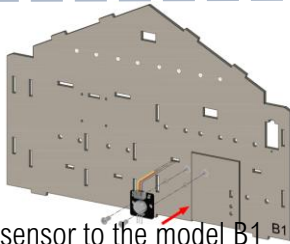
## Step 6

Insert the model D4 to the model B1, and then attach it to the model D1, D2 and D3.



## Step 8

Attach motion sensor to the model B1 (near the door) using M4 \* 10mm screws and nuts. And the connecting wire should be bended to the hole above it.



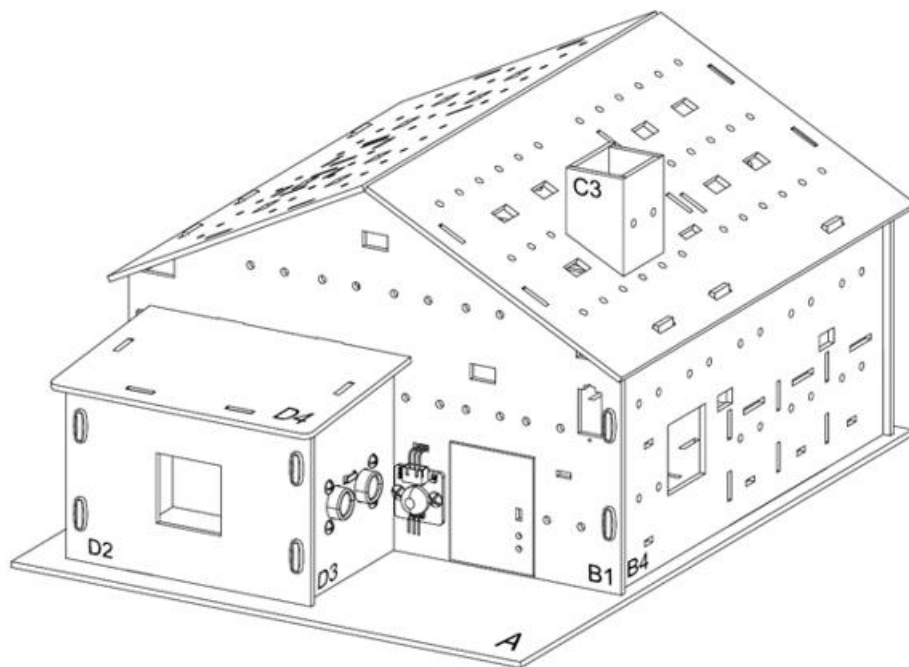
Step 7 Completed.

# ASSEMBLY STEP

---

Step 9

Assembly completed!



# CODING



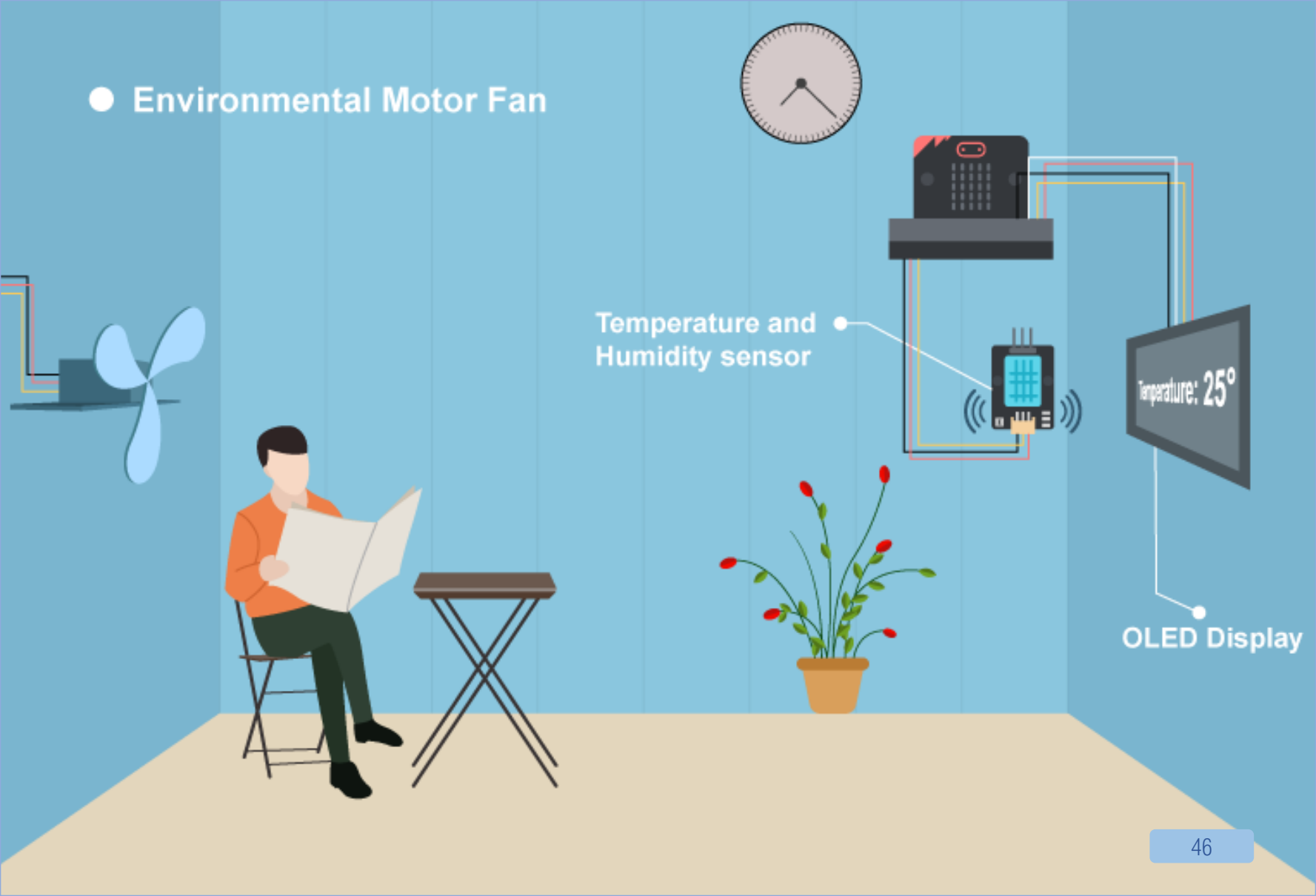
When someone is gotten close to the door, either distance sensor or motion sensor will detect it and warn alert to house owner

The code is a Scratch script on a grid background. It starts with a 'forever' loop block (blue). Inside the loop, there are two 'set' blocks (red): 'set distance to Get distance unit cm trig P14 echo P15' and 'set motion to Get motion (triggered or not) at Pin P2'. These two blocks are enclosed in a dashed yellow box. Below them is an 'if' block (teal) with a condition 'motion = true or distance < 10'. The 'if' block has a 'then' branch containing a 'show icon' block (blue) with an angry face icon and a 'play melody' block (red) with a musical note icon and 'at tempo 120 (bpm)'. It also has an 'else' branch (teal) containing a 'show leds' block (blue) with a 4x4 grid of 16 dots. The 'if' block is enclosed in a dashed yellow box. At the bottom of the 'forever' loop is a 'pause (ms) 1000' block (blue), also enclosed in a dashed yellow box. Numbered circles 1 through 6 are placed to the left of the code blocks to indicate the sequence of steps.

## STEPS:

1. Block **forever** runs as a loop.
2. Set **distance** to **Get distance unit cm trig P14 echo P15**.
3. Set **motion** to **Get motion (triggered or not) at Pin P2**.
4. If **motion** = **true** or **distance** < 10 then, **show icon of angry face** and **play melody at tempo 120 (bpm)**.
5. Else, show leds to clear up.
6. **Pause 1 second**.

## ● Environmental Motor Fan

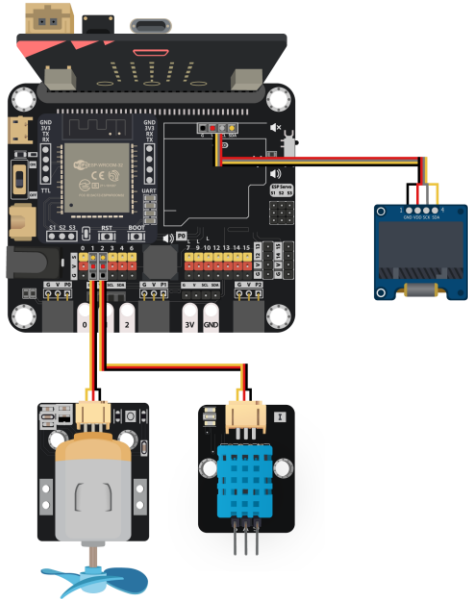


# 3. ENVIRONMENTAL MOTOR FAN

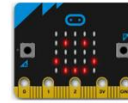
LEVEL: ★ ★ ☆ ☆ ☆

## Introduction

Smart motor fan is a fan that can adjust the power, turn on and off automatically to save electricity when not needed. Nowadays, for example, the temperature in the room is very low and cold, the air-conditioner also keeps turned on at high intensity. If there is a temperature sensor to detect surrounding parameters and change the power automatically, it can save much energy.



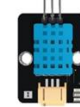
## PART LIST



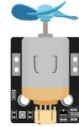
micro:bit (1)



IoT:bit with OLED (1)



Temperature and Humidity Sensor (1)



Motor Fan(1)



3-pin module wire (2)



4-pin extension wire (1)



M4\*10mm Screw (4)



M4\*10mm Nut (4)



M2\*10mm Screw (4)



M2 Nut (4)



Screwdriver (1)



Model A - C (1)  
[Big House Model]



Model E4 (1)  
[Sensor Display Wall]



Model F (1)  
[Motor Stand]



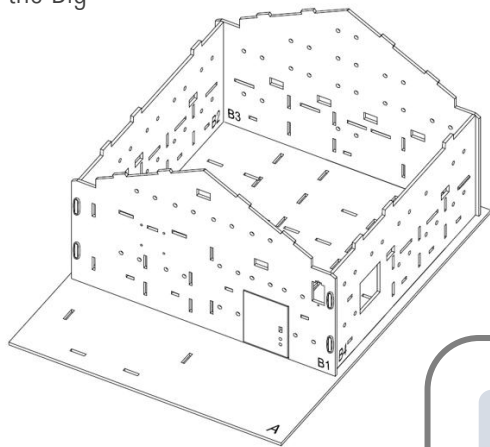
Model I1 (1)  
[Table]



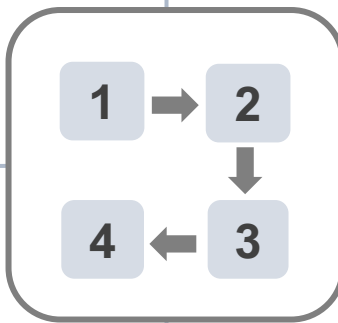
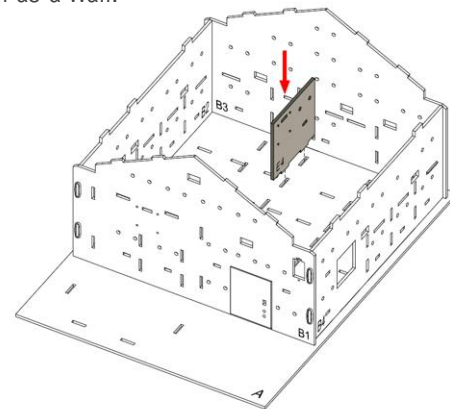
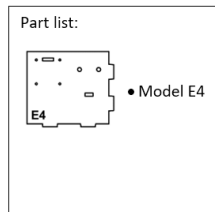
Model I2 (3)  
[Table]

# ASSEMBLY STRUCTIONS

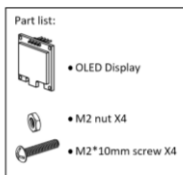
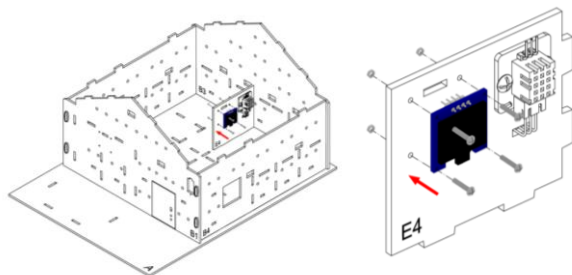
To start with, build the Big House Model.



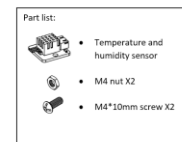
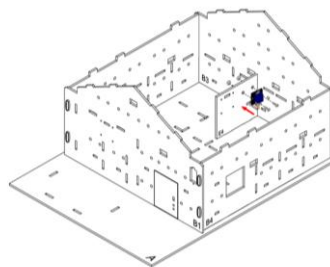
To build a small living room at the right side in the home, put Model E4 onto A model as a wall.



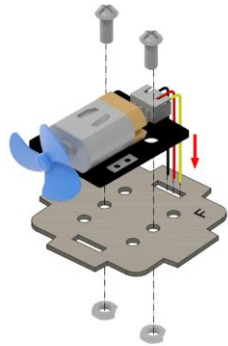
Attach OLED Display to the Model E4 using M2 \* 10mm screws and nuts.



Attach the Temperature and humidity sensor to Model E4 using M4 \* 10mm screws and nuts.

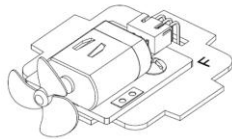


# ASSEMBLY STRUCTIONS

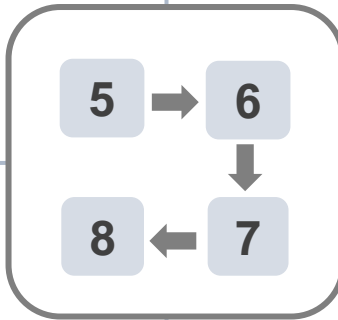
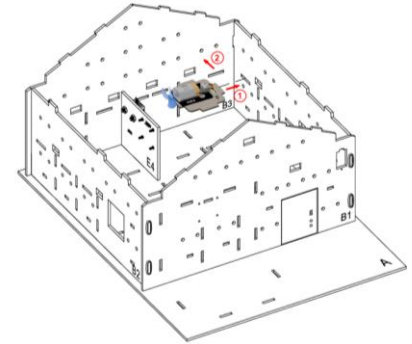
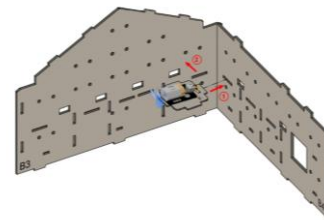


- Part list:
- Motor Fan
  - M4 nut X2
  - M4\*10mm screw X2

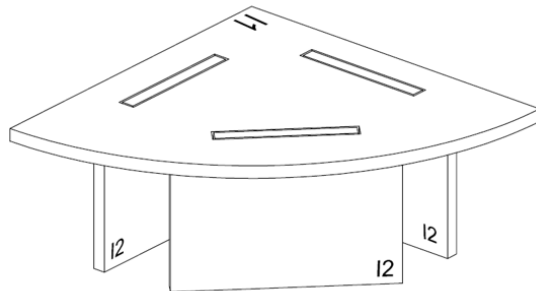
Attach motor fan to the Model F using M4 \* 10mm screws and nuts. And the connecting wire could be bended to the hole next to it.



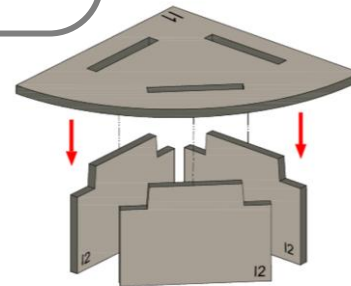
To build a fan stand, put Model F into the B3, B4 model.



The table completed!



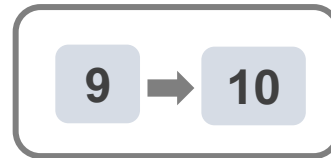
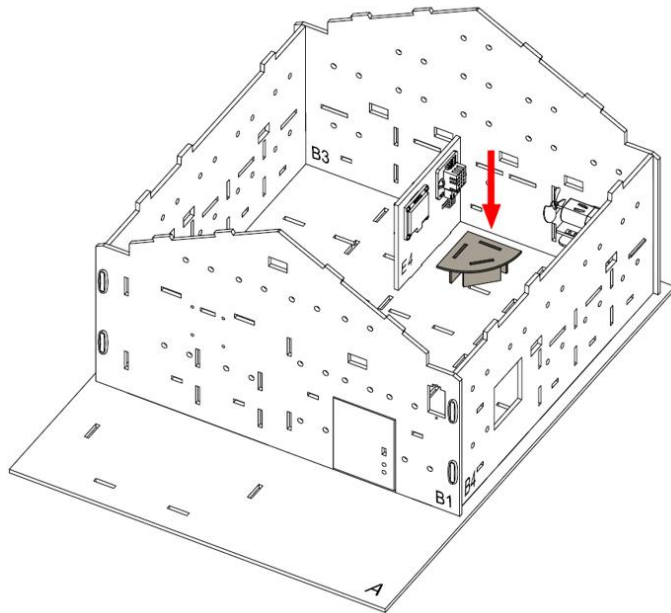
To build a table, put the Model I1 onto the model I2.



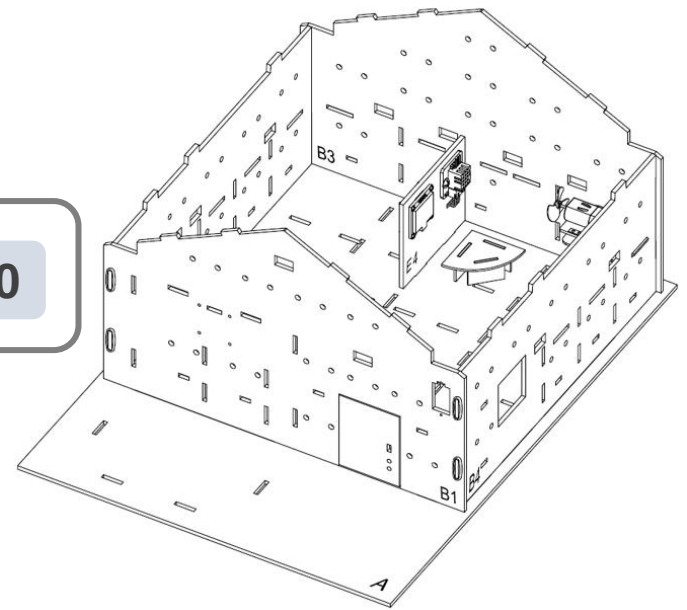
- Part list:
- Model I1
  - Model I2 X3

# ASSEMBLY STRUCTIONS

Place the table at the living room.



Assembly Completed!



# CODING



Apart from open the fans, any other method to reduce the temperature?

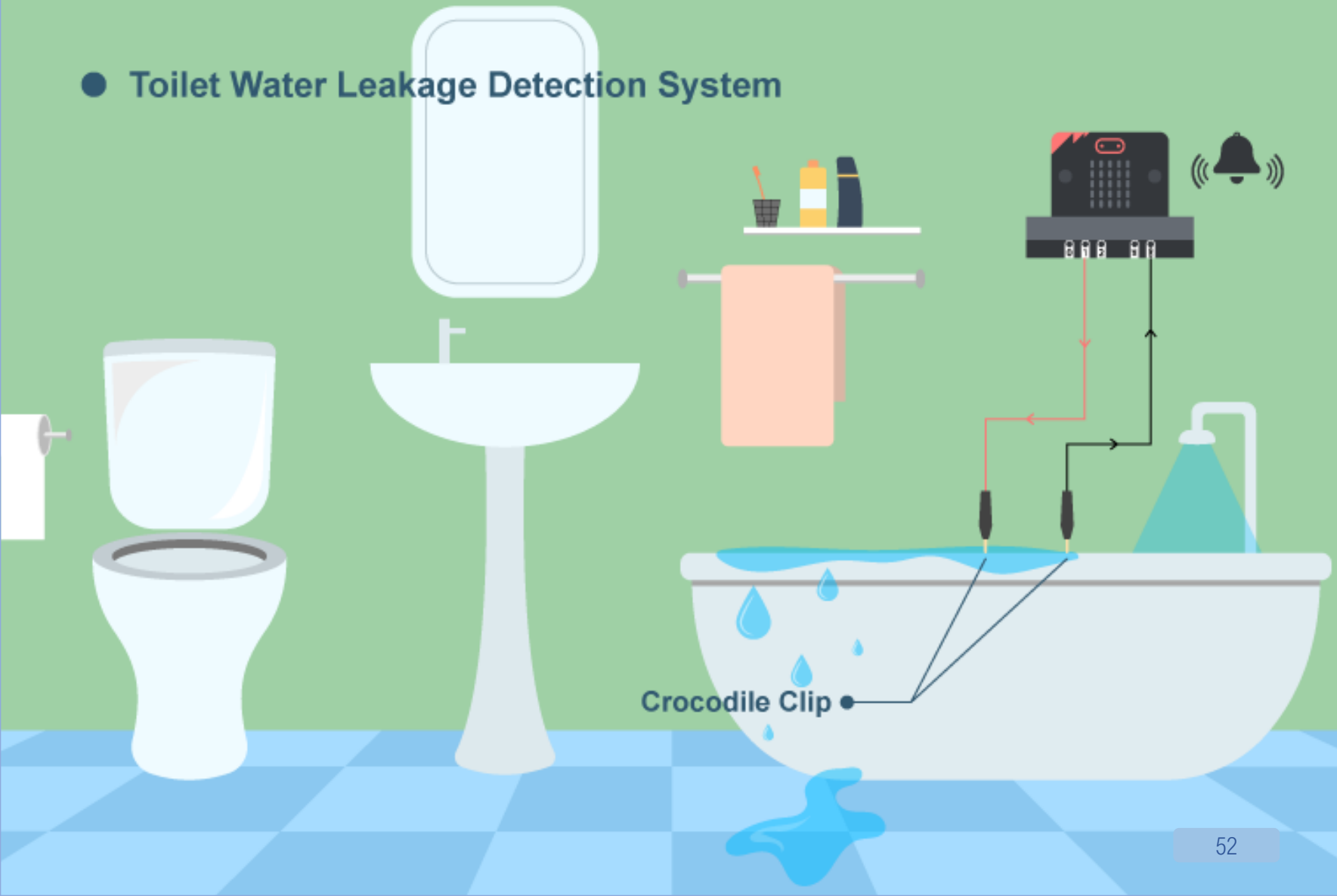
```
1 on start
  initialize OLED with width 128 height 64

2 forever
  3 read temperature & humidity sensor at pin P2
  4 set temperature to get temperature °C
  5 clear OLED display
  6 show string join "Temperature: " temperature
  7 if temperature < 22 then
    8 set motor fan with speed 0 at P1
  9 else if temperature < 26 then
    10 set motor fan with speed 600 at P1
  11 else if temperature >= 26 then
    set motor fan with speed 1023 at P1
```

## STEPS:

1. In **on start**, put **initialize OLED with width 128 height 64** to initialize the OLED display
2. In **Forever**, read the data by **Read Temperature & Humidity Sensor at pin P2**
3. Get the temperature value from data, and save it to variable by **set temperature to Get Temperature °C**
4. Use **clear OLED display** to clear the display
5. Then use **show string join Temperature: temperature** to show the temperature value with formatted text on display
6. In **Forever**, put a nested **if-else** statement. Use the **temperature < 22** as first condition.
7. In the first if segment, turn off the fan by **Set Motor fan with speed 0 at P1**
8. Use the **temperature < 26** as second condition
9. In the second if segment, turn on the fan with half speed by **Set Motor fan with speed 600 at P1**
10. Use the **temperature >= 26** as third condition
11. In the third if segment, turn on the fan with full speed by **Set Motor fan with speed 1023 at P1**

## ● Toilet Water Leakage Detection System



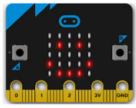
# 4. TOILET WATER LEAKAGE DETECTION SYSTEM

LEVEL: ★ ★ ☆ ☆ ☆

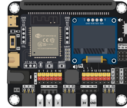
## Introduction

Toilet water leakage detection system is a system that can trigger alert upon unwanted water leakage situations.

## PART LIST



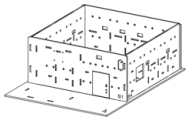
micro:bit (1)



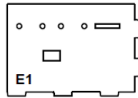
IoT:bit with OLED (1)



Crocodile Clip (2)



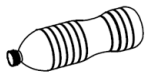
Model A - C (1)  
[Open Style Model]



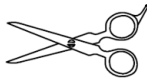
Model E1 (1)  
[Long Sensor Wall]



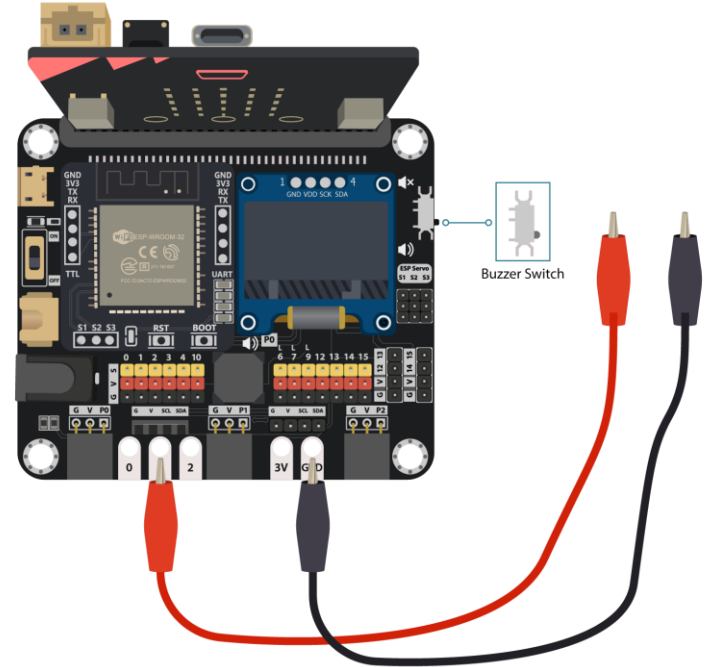
Model E2 (1)  
[Short Sensor Wall]



Water Plastic Cup  
[NOT Provided]

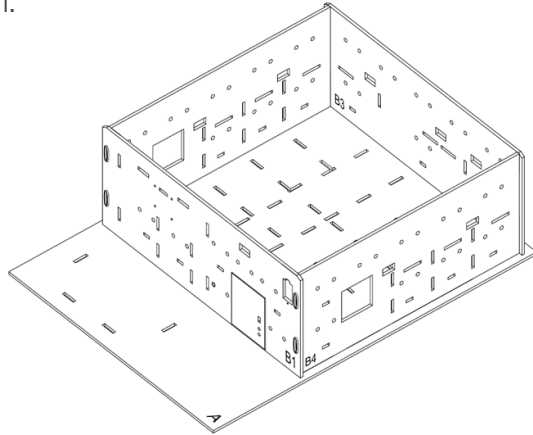


Scissor (1)  
[NOT Provided]

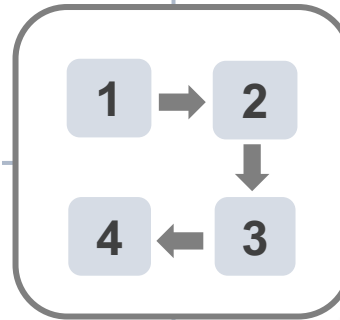
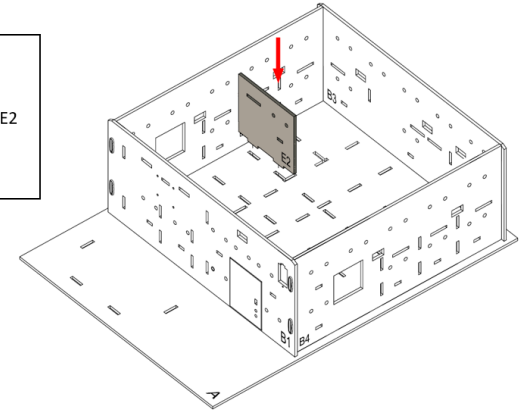
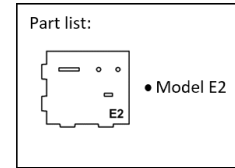


# ASSEMBLY STRUCTIONS

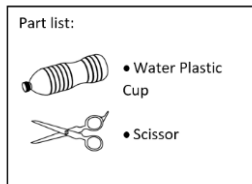
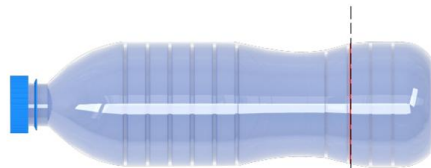
In this case, “Open Style Model” is used as a home base to start with.



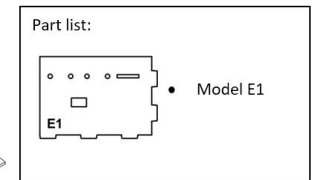
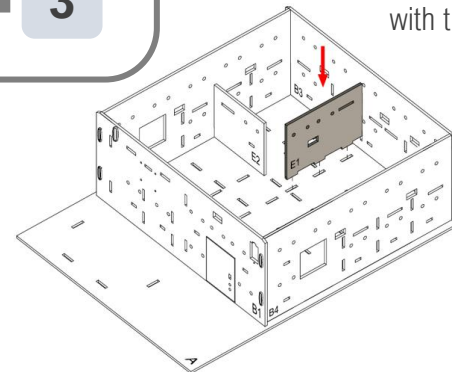
Let's build a toilet room. Put Model E2 onto model A, align with the holes at it.



Put Model E1 onto model A, align with the holes at model A, B3.

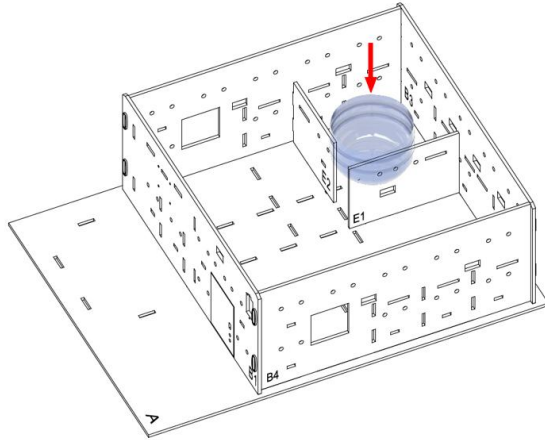


Find a plastic bottle and cut it to become a water tank. Cut the bottle at red line as shown.

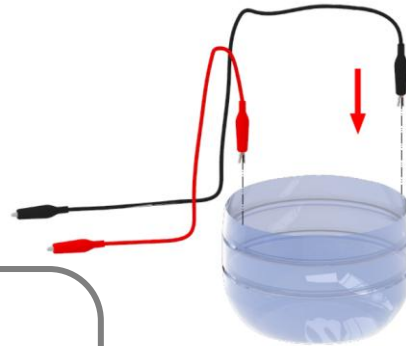


# ASSEMBLY STRUCTIONS

Place the water tank in the toilet.



Place two crocodile clips on the both sides of the water tank to detect water flowing.



Part list:



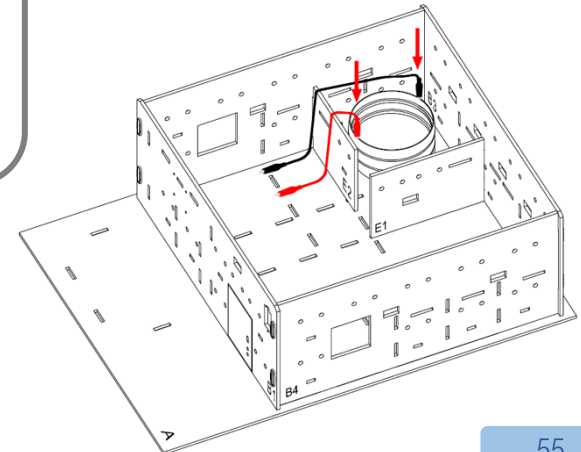
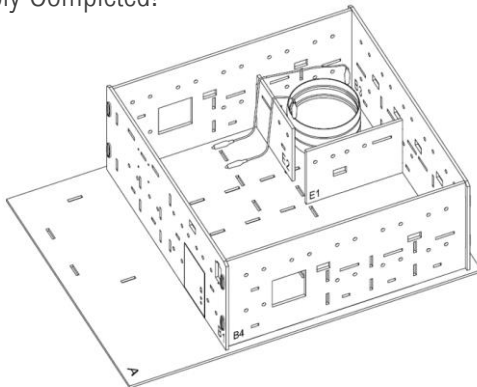
• Crocodile Clip X2

5

6

7

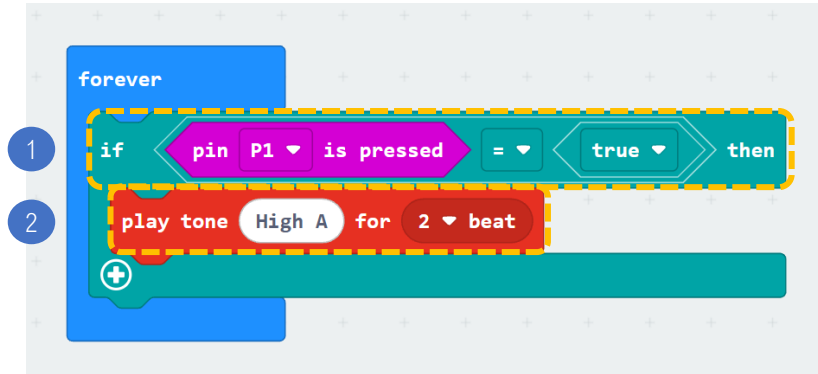
Assembly Completed!



# CODING



Apart from the warning by buzzer, any other method can be used to notify the house owner? i.e. showing red LED

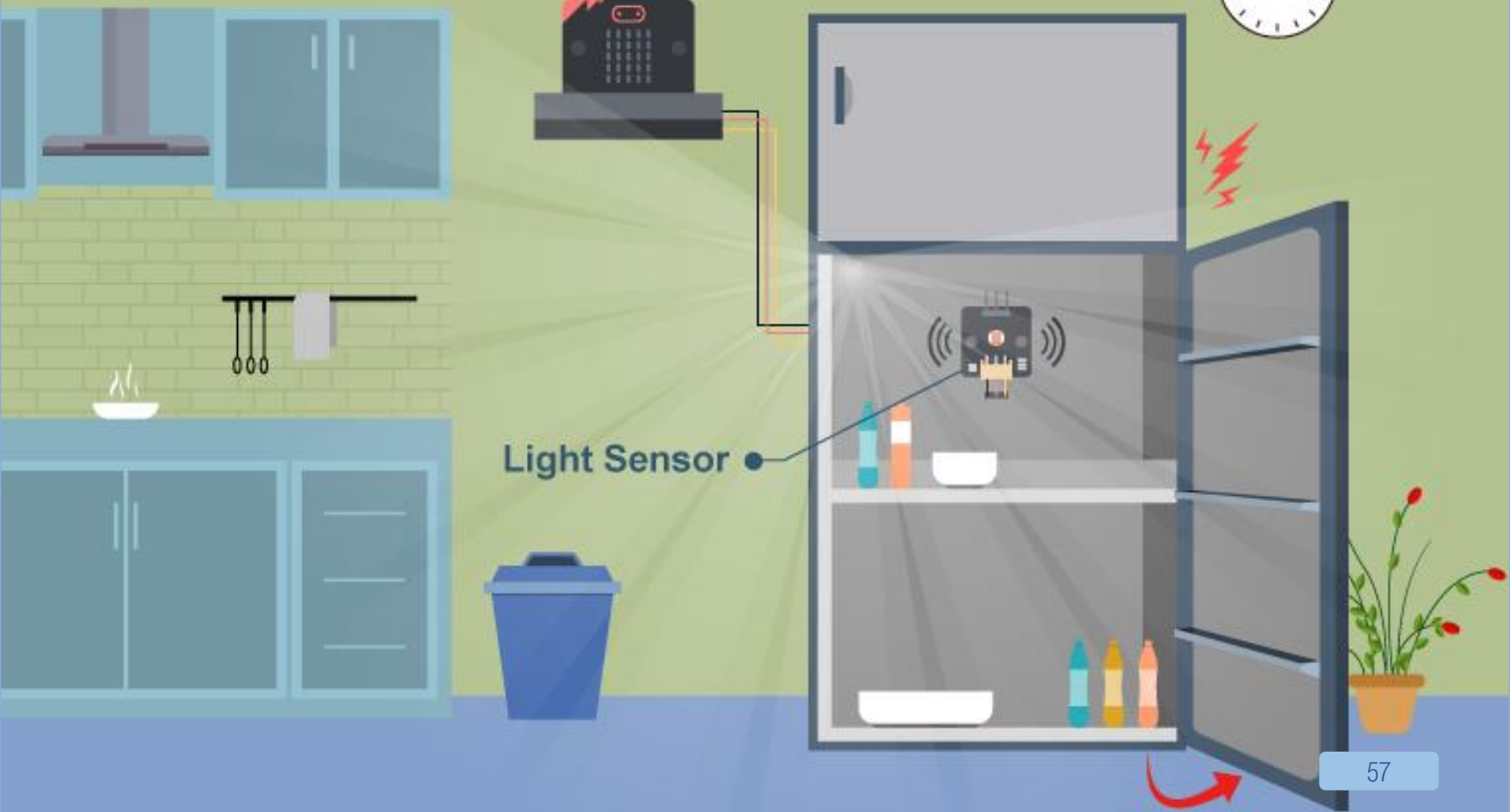


## STEPS:

1. In Forever, put a **if** statement. Put **pin P1 is pressed = true** as condition
2. Use **play tone High A for 2 beat** to play warning sound

## ● Smart Fridge Alert

10 mins



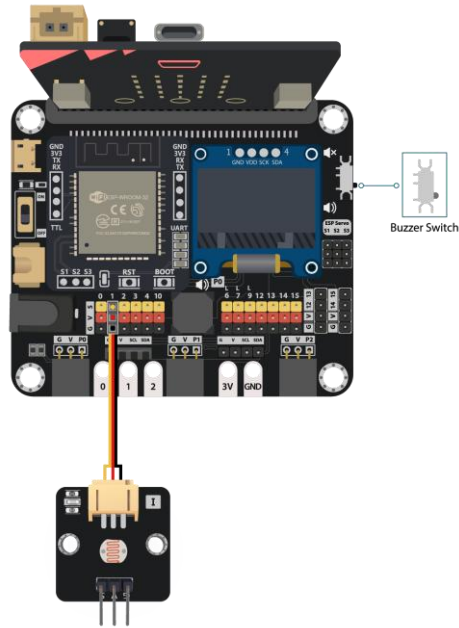
Light Sensor ●

# 5. SMART FRIDGE ALERT

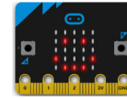
LEVEL: ★ ★ ☆ ☆ ☆

## Introduction

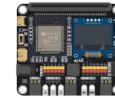
Smart Fridge is a Fridge that has an extra sensor to help it determine the door close state, when the user forgets to close the door, it will warn the user to close it to reduce waste of energy and prevent food poisoning.



## PART LIST



micro:bit (1)



IoT:bit with Buzzer (1)



Light Sensor (1)



3-pin module wire (1)



M4\*10mm Screw (2)



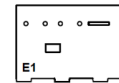
M4\*10mm Nut (2)



Screwdriver (1)



Model A - C (1)  
[Big House Model]



Model E1 (1)  
[Long Sensor Wall]



Model E3 (1)  
[Long Sensor Wall]



Model G1 (1)  
[Fridge]



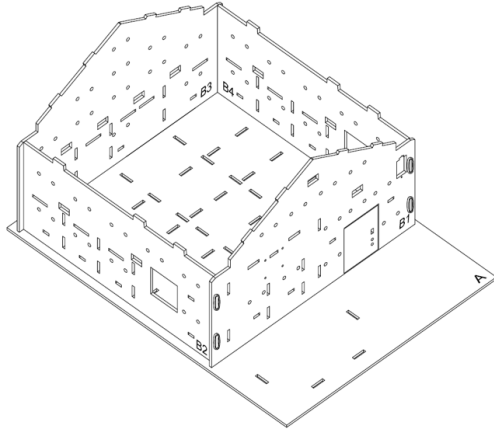
Model G2 (1)  
[Fridge]



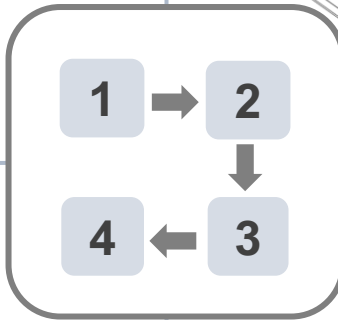
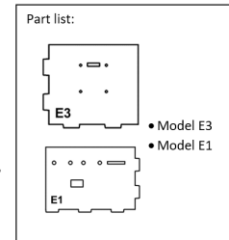
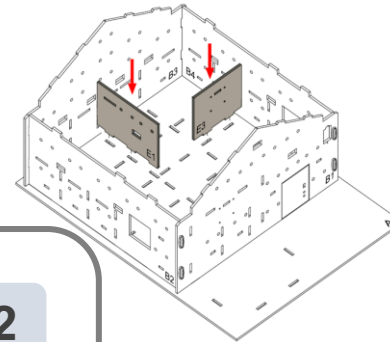
Model G3 (1)  
[Fridge]

# ASSEMBLY STRUCTIONS

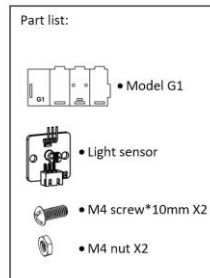
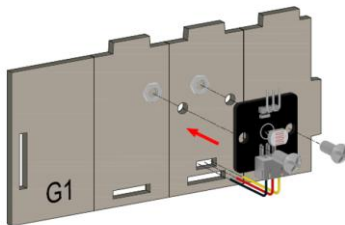
In this case, “Big House Model” is used as a home base to start with.



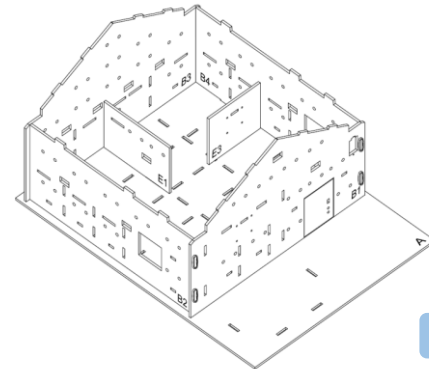
Let's build a kitchen. Put Model E3 onto the Model A and B4, align with the holes at it. Put Model E1 onto the Model A, align with the holes at Model A and B3.



To build a fridge, attach light sensor onto model G1 using M4 \* 10mm screws and nuts. And the connecting wire could be bended to the hole below it.

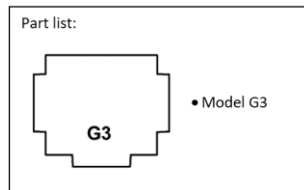
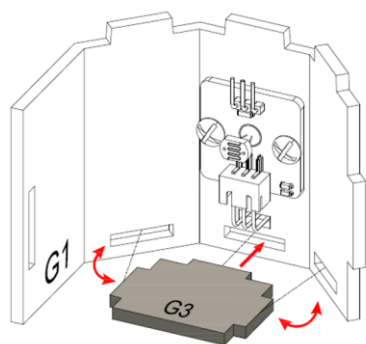


Completed.

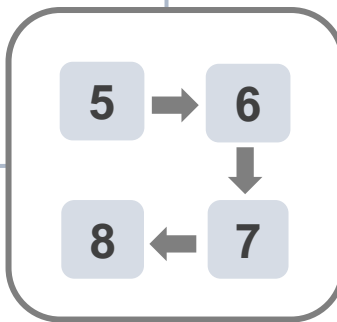
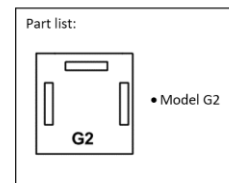
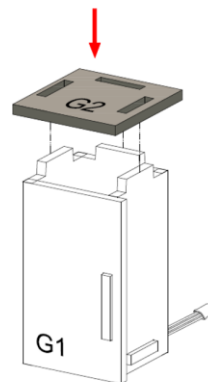


# ASSEMBLY STRUCTIONS

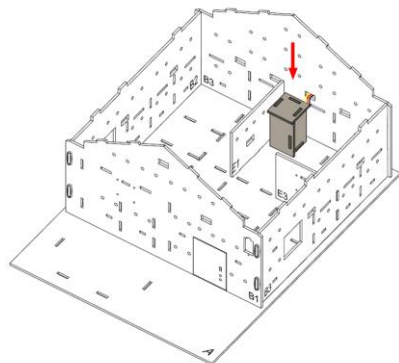
Fold and bend the Model G1 accordingly and put Model G3 into the Model G1, align with the holes.



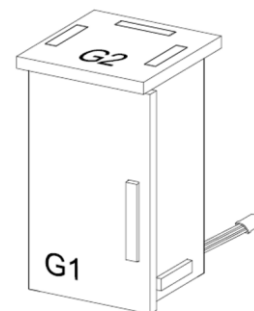
Put Model G2 onto the Model G1, align with the holes.



Place the fridge at the corner.



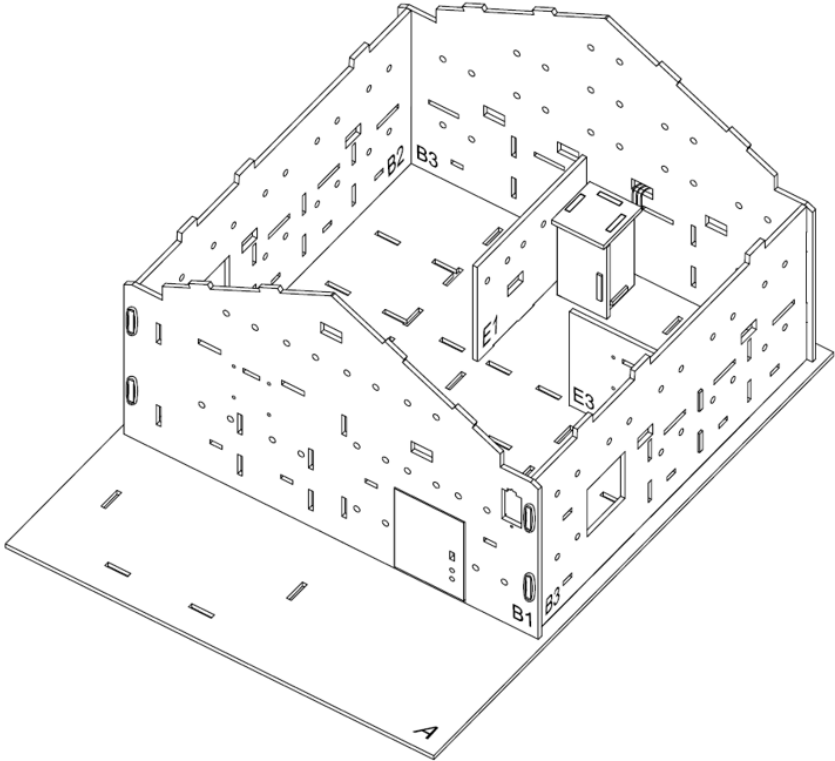
The fridge completed!



# ASSEMBLY STRUCTIONS

---

Assembly Completed!



# CODING



Apart from making the sound from the buzzer, another method to warn the user?

The image shows a Scratch code editor with the following blocks:

- 1** **on start** block containing a **set counter to 0** block.
- forever** loop containing:
  - 2** **if** block: **get light value (percentage) at pin P1** **>** **30** **then**
  - 3** **change counter by 1** block (inside the first if's then branch).
  - 4** **if** block: **counter** **≥** **60** **then** (inside the first if's then branch).
  - 5** **start melody** block with **jump down** and **repeating once** (inside the second if's then branch).
  - 6** **else** block: **set counter to 0** (inside the first if's else branch).
  - 7** **pause (ms) 1000** block (at the end of the forever loop).

## STEPS:

1. On **start**, create a “counter” variable and **set to 0**
2. In **Forever**, put a **if-else** statement. Use **Get light value (percentage) at Pin P1 > 30** as condition.
3. Add the counter with 1 by **change counter by 1** to sum up the time of door opened
4. Put the second **if** statement with **counter >= 60** condition to examine when should issue warning sound
5. In the second **if** segment, put **start melody jump down repeating once** to issue the warning sound
6. Back to the first **if-else** level, in the **else** segment, that's means light intensity is weak, door was closed, so reset the counter to 0 by **set counter to 0**
7. Put a **pause (ms) 1000** to check the door each second

## ● Mechanical Switch Password Door



# 6. MECHANICAL PASSWORD SWITCH DOOR

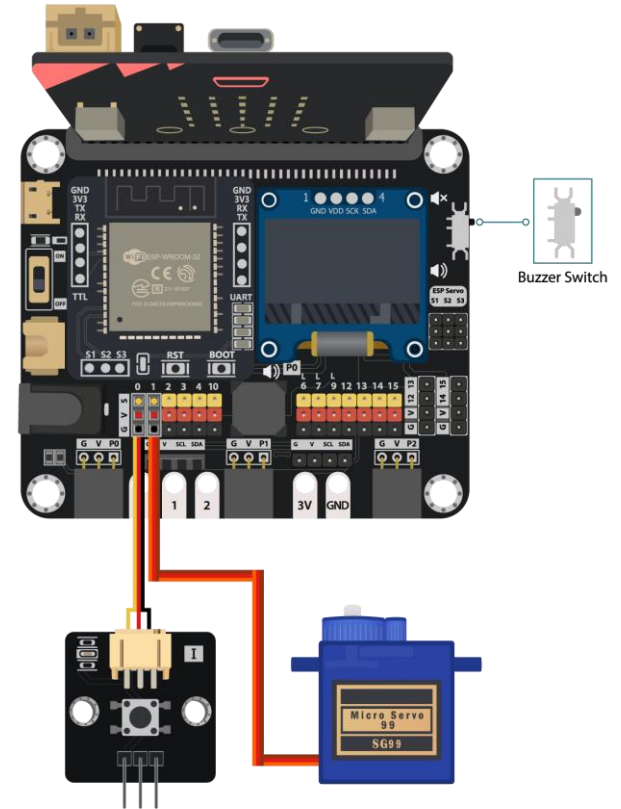
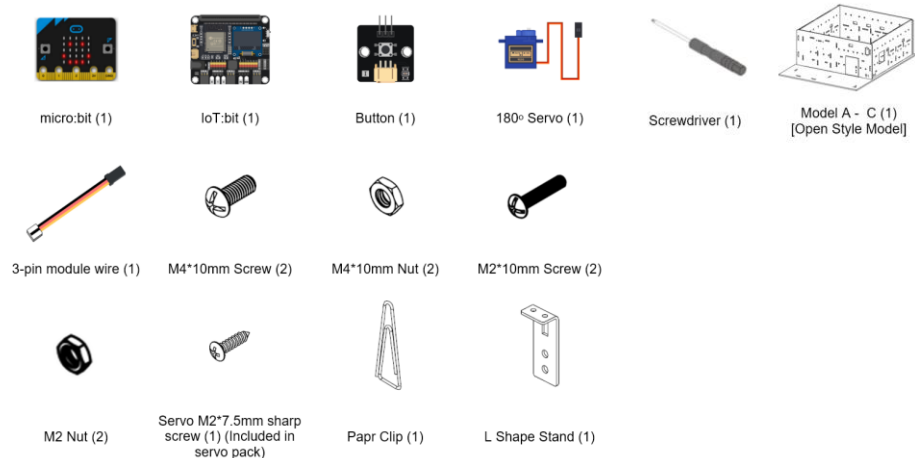
LEVEL: ★ ★ ★ ☆ ☆

## Introduction

Nowadays, digital locks are common, apart from preventing theft going in, it also provides convenience. For instance, home owner no need to bring the physical key, just use the password or Biometric information, the door can be opened or closed without pushing or pulling it.

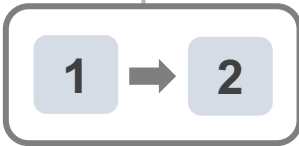
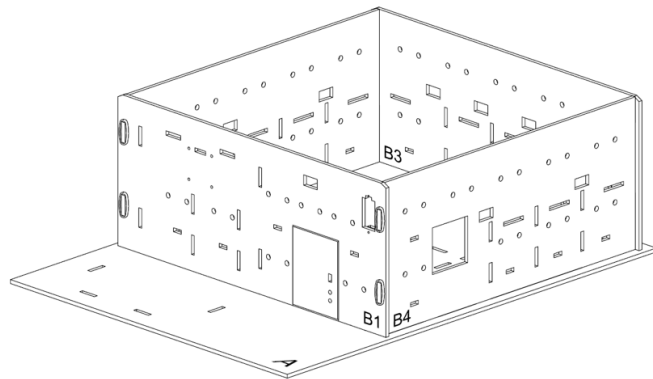
In this case, it will demonstrate the use of the basic switches with micro:bit to perform a mechanical password switch door.

## PART LIST

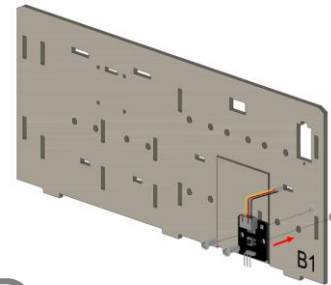


# ASSEMBLY STRUCTIONS

In this case, build the “Open Style Model” as a home base.

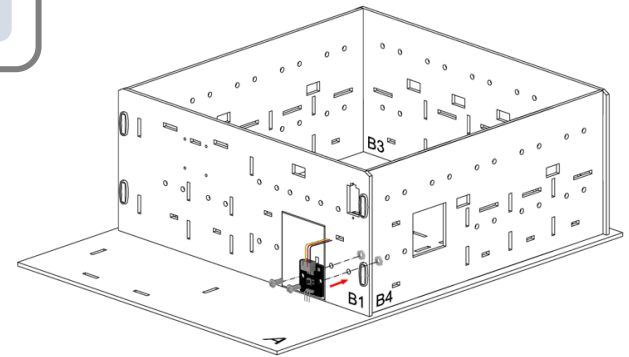


To build a smart door, attach button module onto Model B1 using M4 \* 10mm screws and nuts, align with the holes nearby the door on model B1. And the connecting wire should be bended to the hole above.



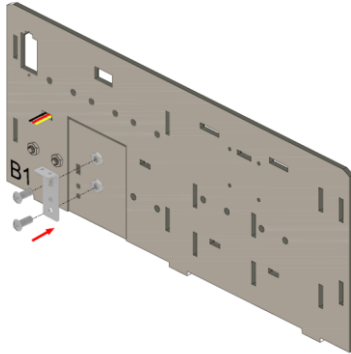
Part list:

- Button module
- M4 nut X2
- M4\*10mm screw X2



# ASSEMBLY STRUCTIONS

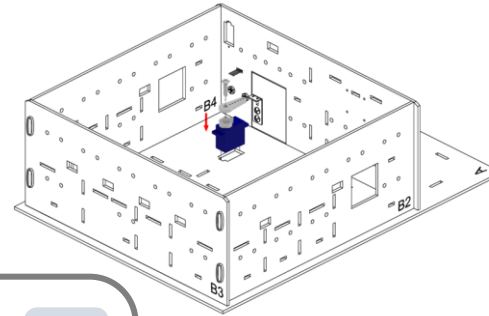
Attach the L-shape stand to the back of Model B1 using M3 \* 10mm screws and nuts.



Part list:

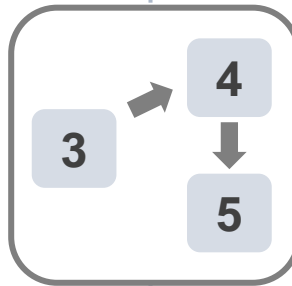
- L Shape Stand
- M3 nut X2
- M3\*10mm screw X2

Install 180° servo by using M2 \* 7.5mm sharp screw and then attach it to model A.

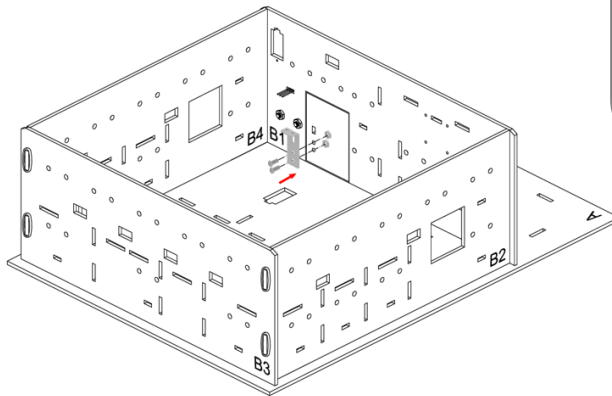


Part list:

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

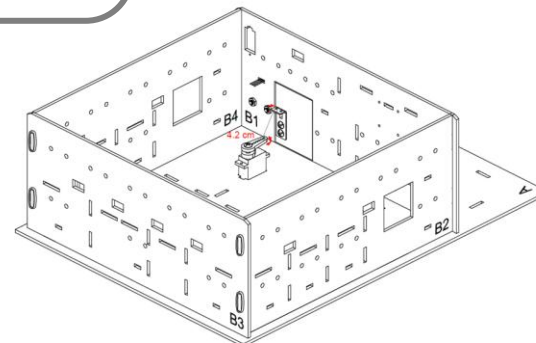


Use a paper clip as a hook to connect the L shape stand and 180° servo.



Part list:

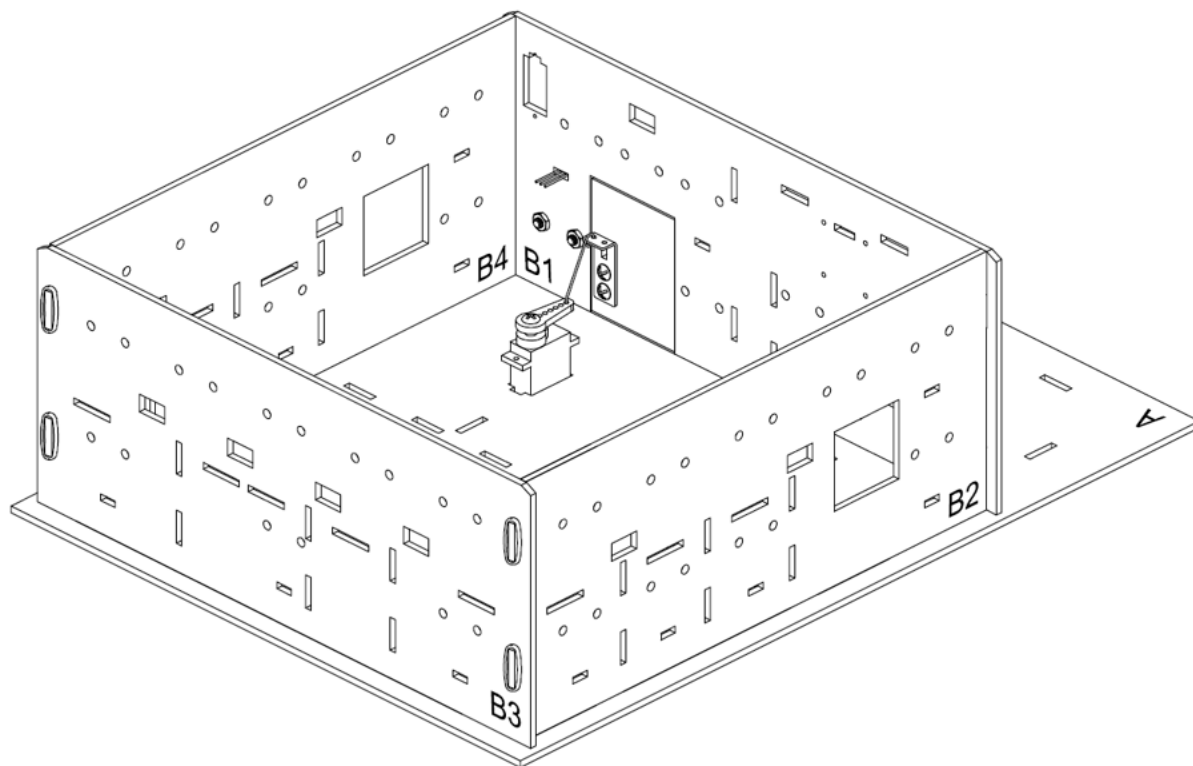
- Paper Clip \*1



# ASSEMBLY STRUCTIONS

---

Assembly Completed!



# CODING



Can it make a doorbell to produce sound when the door is opening?

1. **on start**

- set **unlock** to **false**
- set **count\_A** to **0**
- set **count\_B** to **0**
- Turn 180° Servo to **45** degree at **P1**

2. **When Button at P0 pressed**

3. **if** **unlock** = **true** **then**

4. **Turn 180° Servo to 180** degree at **P1**

**pause (ms) 5000**

**Turn 180° Servo to 45** degree at **P1**

**set unlock** to **false**

## STEPS:

1. In **on Start**, initialize the state of door to closed by **set unlock to false** and **Turn 180 Servo to 45 degree at P1**, also set the count variable **count\_A** and **count\_B** to 0.
2. Snap the **When Button at P0 pressed** block to editor
3. Put a **if** statement in the **When Button at P0 pressed** block. Set the condition to **unlock = true**
4. When **unlock = true**, that's means it is unlock successfully, the servo should open the door by **Turn 180 servo to 180 degree at P1**, after that, wait for 5 second by **pause (ms) 5000**, then close and lock the door again by **Turn 180 servo to 45 degree at P1** and **set variable unlock to false**

# CODING



Other than the door, can the switch apply to other usage? (e.g Turn on/off LED)

```
5 on button A pressed
   change count_A by 1

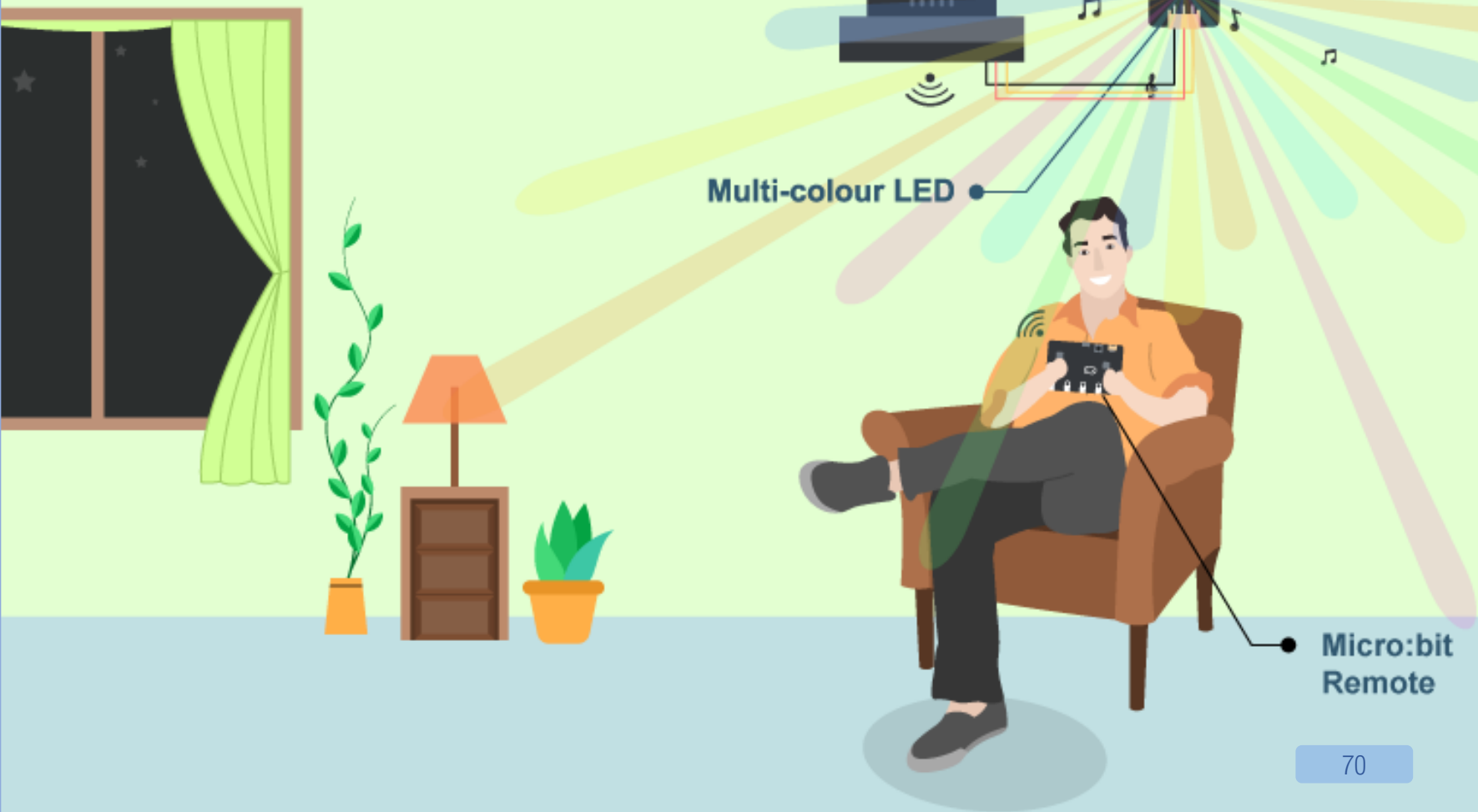
7 on button B pressed
   change count_B by 1
```

```
8 on button A+B pressed
9   if count_A = 2 and count_B = 3 then
10    show icon tick
11    set unlock to true
12  else
13    show icon cross
14    set unlock to false
15  set count_A to 0
16  set count_B to 0
```

## STEPS:

5. Snap **on button A pressed** to editor
6. Use **change count\_A by 1** to increase the count variable by 1
7. Repeat the step with minor different on target to let button B also can increase **count\_B**
8. Snap **on button A+B pressed** to editor
9. Put a **if-else** statement inside. Use **count\_A = 2** and **count\_B = 3** as condition, you may also change it to your favourite password.
10. In the **if** segment, that's means the input is correct, **show icon tick**, and **set unlock to true** to allow open the door when press the extended button
11. In the **else** segment, that's means the input is not correct, **show icon cross**, and **set unlock to false** to denied open the door when press the extended button
12. Reset the two count variable **count\_A** and **count\_B** to 0 for next input

## ● Smart Remote Control Musical Light



# 7. SMART REMOTE CONTROL MUSICAL LIGHT (SENDER)

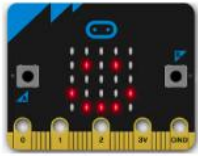
---

LEVEL: ★ ★ ★ ☆ ☆

## Introduction

Nowadays, different electronic furniture can also be controlled remotely. In this case, it illustrates the concept of remote control in Smart Home. The home owner can control the room atmosphere in the living room by pressing the remote. (Change of music and light)

## PART LIST



micro:bit (1)



Battery holder  
(Optional)

# CODING



Can you control other hardware like a motor fan using radio control?

1

on start

radio set group 1

2

on button A pressed

3

radio send string "funny"

4

on button B pressed

radio send string "excited"

on button A+B pressed

radio send string "stop\_music"

## STEPS:

1. In **on Start**, put a **radio set group 1** to join the radio group 1
2. Snap **on button A pressed** to editor
3. In **on button A pressed**, put a **radio send string funny** to send funny message to group 1 micro:bits
4. Repeat the steps with minor changes on the trigger button and string to create other message sender for excited and stop\_music

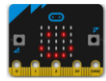
# 7. SMART REMOTE CONTROL MUSICAL LIGHT (RECEIVER)

LEVEL: ★ ★ ★ ☆ ☆

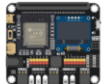
## Introduction

Nowadays, different electronic furniture can also be controlled remotely. In this case, it illustrates the concept of remote control in Smart Home. The home owner can control the room atmosphere in the living room by pressing the remote. (Change of music and light)

## PART LIST



micro:bit (1)



IoT:bit (1) with Buzzer



Multi-colour LED (WS2812) (1)



3-pin module wire (2)



Model H (1)  
[Decoration Display]



M4\*10mm Screw (2)



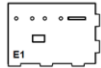
M4\*10mm Nut (2)



Screwdriver (1)



Model A - C (1)  
[Big House Model]



Model E1 (1)  
[Long Sensor Wall]



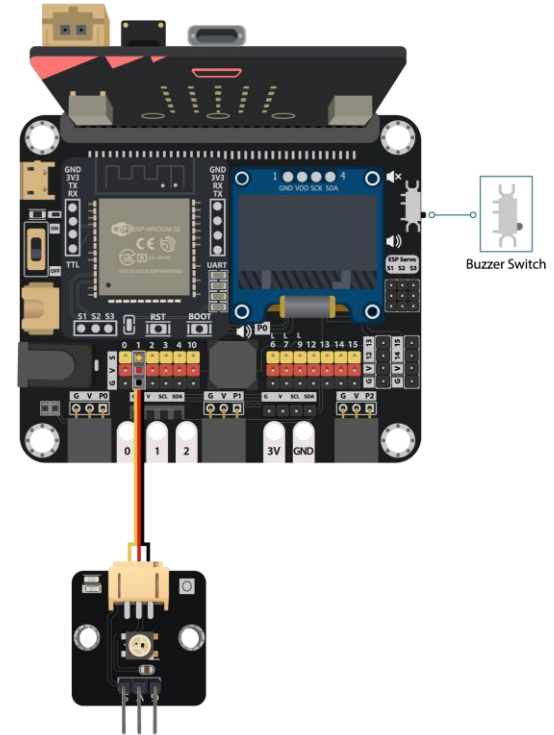
Model K1 (1)  
[Sofa]



Model K2 (1)  
[Sofa]

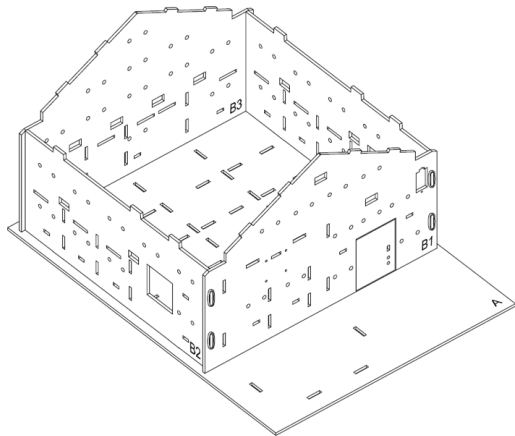


Model K3 (2)  
[Sofa]

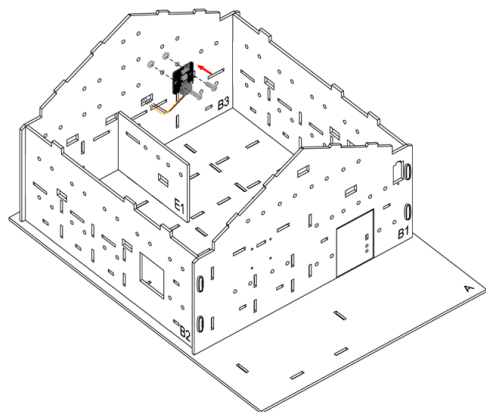
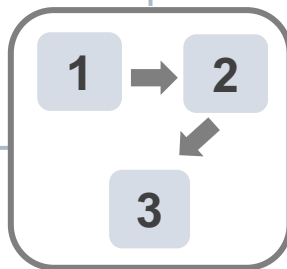
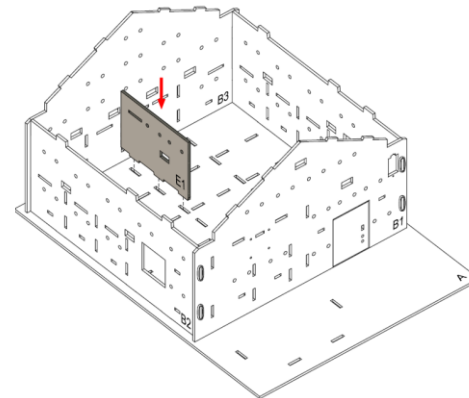
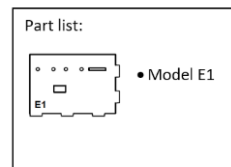


# ASSEMBLY STRUCTIONS

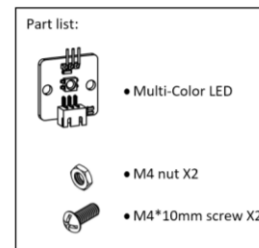
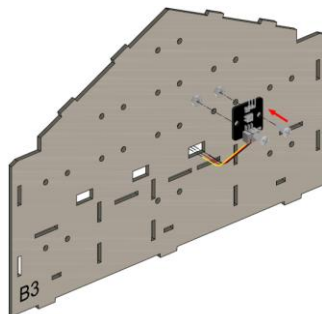
In this case, build the “Big Style Model” as a home base.



To build a living room, put model E1 onto model A, align with the holes at model A and B3.

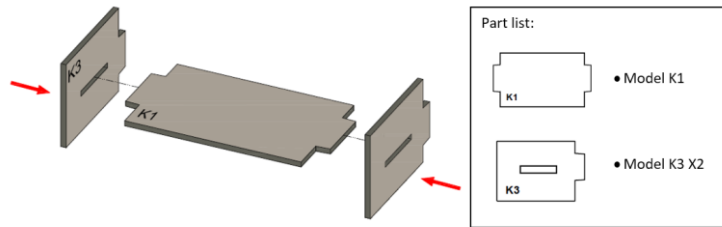


Attach the multi-color LED to the model B3 using M4\*10mm screws and nuts. And the connecting wire should be bended to the hole nearby.

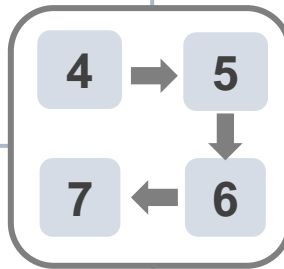
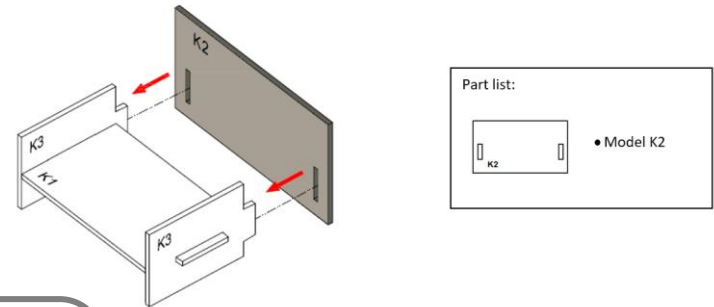


# ASSEMBLY STRUCTIONS

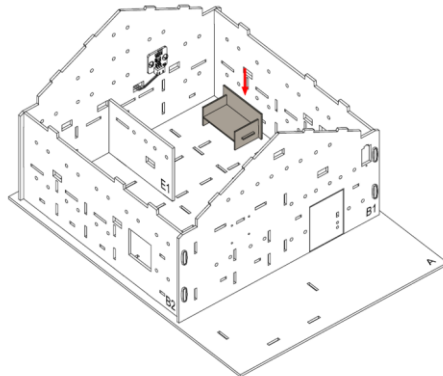
To build a sofa model. Put the model K3 to the two sides of model K1.



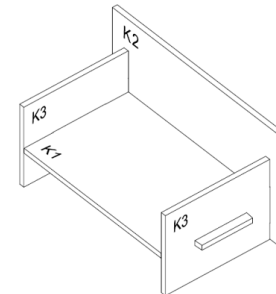
Put model K2 all together to the cardboard parts (K1-K3).



Place the sofa at the living room.

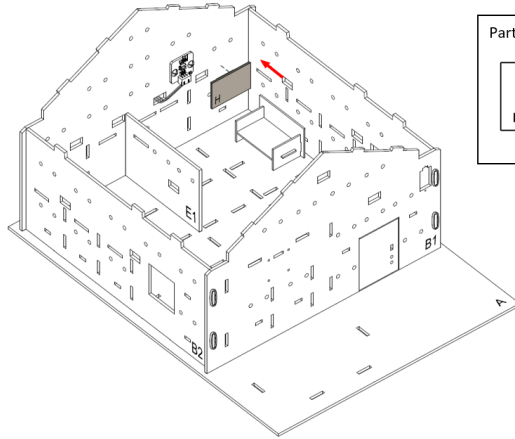


The sofa completed!




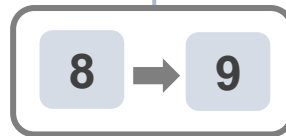
# ASSEMBLY STRUCTIONS

Put the model H as a decoration display on model B3.

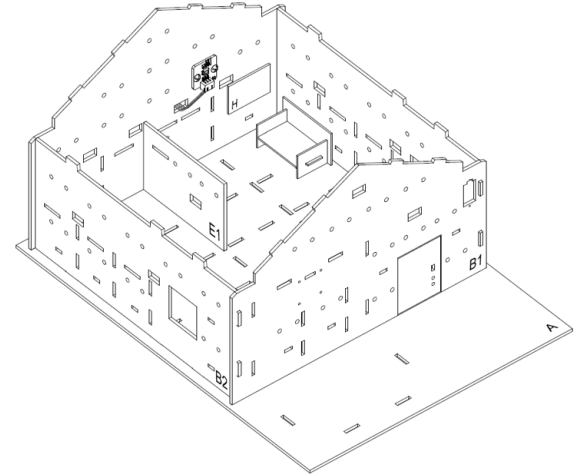


Part list:

	• Model H (Decoration Display Model)
---	--



Assembly completed!



# CODING



Can you control other hardware like a motor fan using radio control?

The code is organized into two main sections. The first section, labeled 'on start', contains three blocks: 'radio set group 1', 'set strip to NeoPixel at pin P1 with 1 leds as RGB (GRB format)', and 'set mode to 0'. The second section, labeled 'on radio received receivedString', contains a nested 'if-else' structure. The first condition checks if 'receivedString' is 'stop\_music', and if true, sets 'mode' to 0. The second condition checks if 'receivedString' is 'funny', and if true, sets 'mode' to 1. The third condition checks if 'receivedString' is 'excited', and if true, sets 'mode' to 2. A plus sign icon is visible at the bottom left of the 'on radio received' block, indicating it can be expanded.

```
on start
  radio set group 1
  set strip to NeoPixel at pin P1 with 1 leds as RGB (GRB format)
  set mode to 0

on radio received receivedString
  if receivedString = "stop_music" then
    set mode to 0
  else if receivedString = "funny" then
    set mode to 1
  else if receivedString = "excited" then
    set mode to 2
```

## STEPS:

1. Create a variable called **mode**
2. In **on Start**, put a **radio set group 1** to join the radio group 1
3. Initialize Multi-Color LED by **set strip to NeoPixel at pin P1 with 1 leds as RGB (GRB format)**
4. Set the variable mode value to 0 by **set mode to 0**
5. Snap **on radio received receivedstring** block to editor
6. Put a nested **if-else** statement inside that block
7. In the first condition, use **receivedstring = stop\_music** to filter out the stop\_music message, change the **mode** flag variable to 0
8. In the second condition, use **receivedstring = funny** to filter out the funny message, change the **mode** flag variable to 1
9. In the third condition, use **receivedstring = excited** to filter out the excited message, change the **mode** flag variable to 2

# CODING



Can you control other hardware like a motor fan using radio control?

10 function Flash

- strip show color red
- pause (ms) 500
- strip show color black
- pause (ms) 500
- strip show color red
- pause (ms) 500
- strip show color black
- pause (ms) 500
- strip show color red
- pause (ms) 500
- strip show color black
- pause (ms) 500
- strip show color red
- pause (ms) 500
- strip show color black
- pause (ms) 500

11 function Rainbow

- strip show color red
- pause (ms) 500
- strip show color orange
- pause (ms) 500
- strip show color yellow
- pause (ms) 500
- strip show color green
- pause (ms) 500
- strip show color blue
- pause (ms) 500
- strip show color indigo
- pause (ms) 500
- strip show color violet
- pause (ms) 500
- strip show color purple
- pause (ms) 500

## STEPS:

10. Create two function called **rainbow** and **flash**
11. For each function, use **strip show color XXX** and **pause(ms) XXX** to fill in the pattern of color changes as you want

# CODING



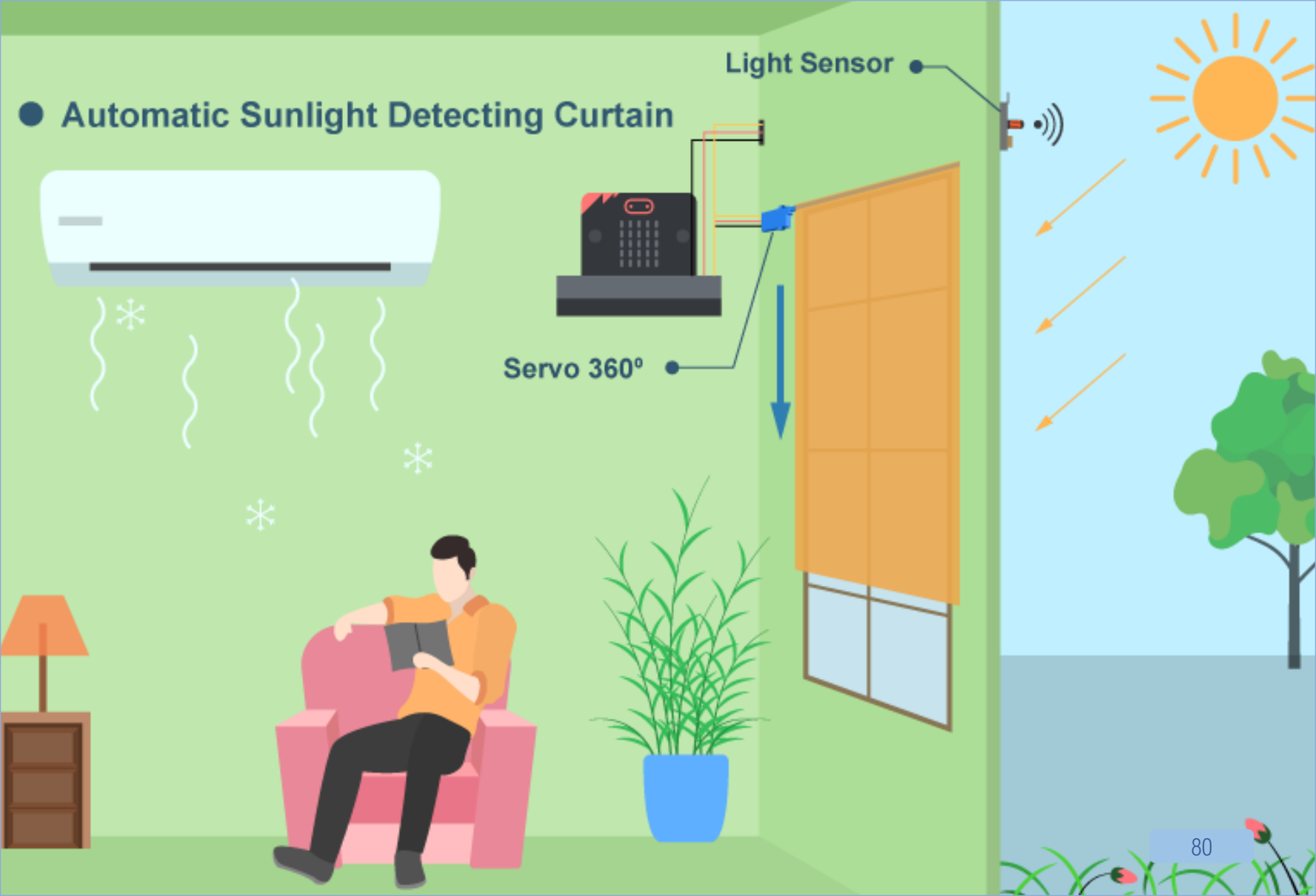
Apart from the built in melody, can you make your own music melody?

```
forever
13 if mode = 0 then
14   stop all sounds
15   strip show color black
16 else if mode = 1 then
17   stop all sounds
18   start melody prelude repeating once
19   call Rainbow
20 else if mode = 2 then
21   stop all sounds
22   start melody chase repeating once
23   call Flash
```

## STEPS:

12. Put a nested if-else statement inside the Forever
13. Use **mode = 0** as the first condition
14. In the first if segment, that's means stop\_music, stop the playing sound by **stop all sounds**
15. Turn off the LED by **strip show color black**
16. Use **mode = 1** as the second condition
17. In the second if segment, that's means funny, stop the playing sound by **stop all sounds**, then play a funny music with **start melody prelude repeating once**
18. Execute the function to change to rainbow pattern by **call Rainbow**
19. Use **mode = 2** as the third condition
20. In the third if segment, that's means excited, stop the playing sound by **stop all sounds**, then play a excited music with **start melody chase repeating once**
21. Execute the function to change to flash pattern by **call flash**

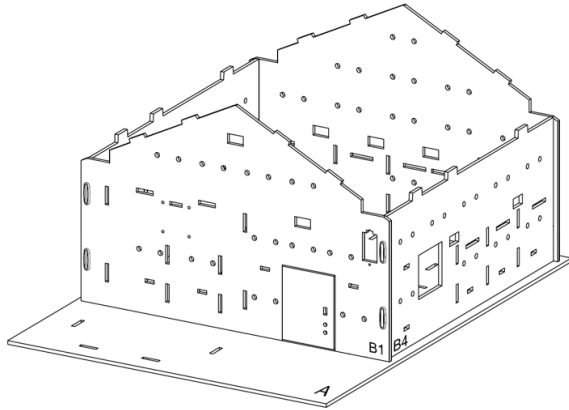
# ● Automatic Sunlight Detecting Curtain



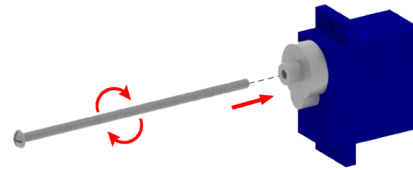


# ASSEMBLY STRUCTIONS

In this case, build the “Big Style Model” as a home base.



To make a curtain. Attach the curtain rod to 360° servo by using the screwdriver to assist.



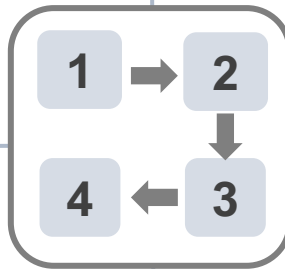
Part list:



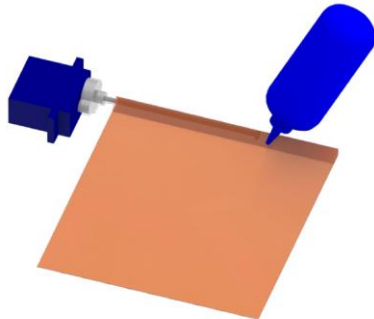
• 360° Servo



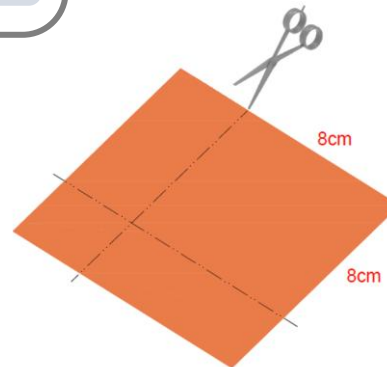
• Curtain Rod



Stick the cutted paper on the curtain rod by glue.



Cut the curtain paper into a 8cm\*8cm square.



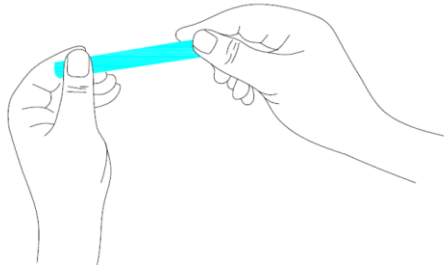
Part list:




• Curtain Paper

# ASSEMBLY STRUCTIONS

Shape the blu tack into a rectangle by hand.

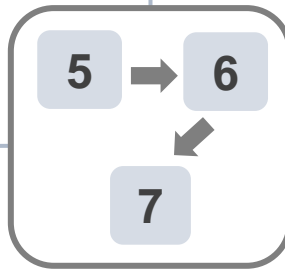
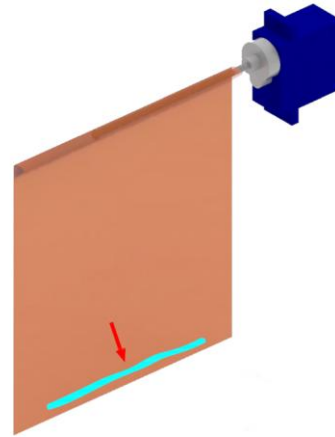


Part list:

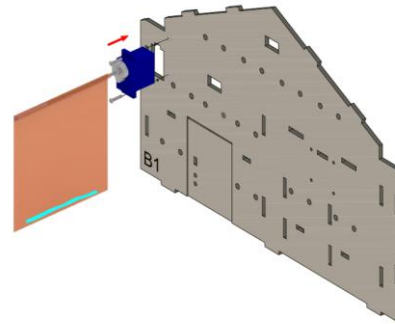
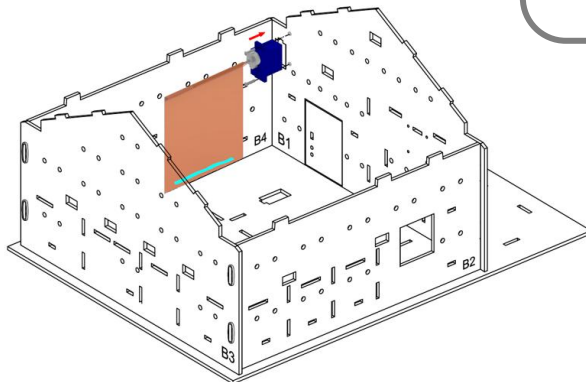


- Blu Tack


Stick the shaped blu tack at the bottom of the paper curtain.




Attach the completed curtain (servo) onto Model B1 using M2 \* 10mm screws and nuts.



Part list:



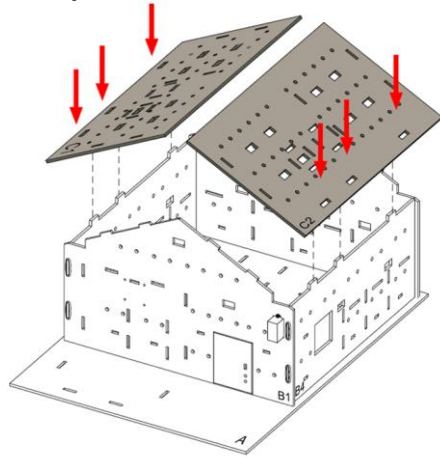
- M2\*10mm screw X2



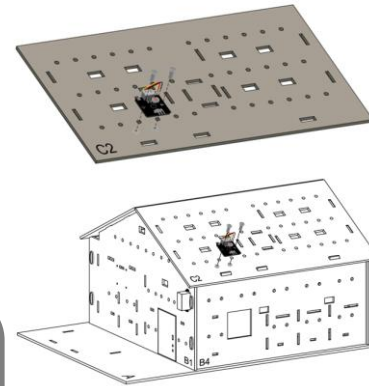
- M2 nut X2

# ASSEMBLY STRUCTIONS

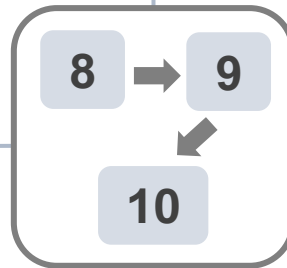
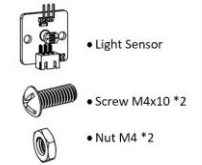
Close the house by model C1 and C2.



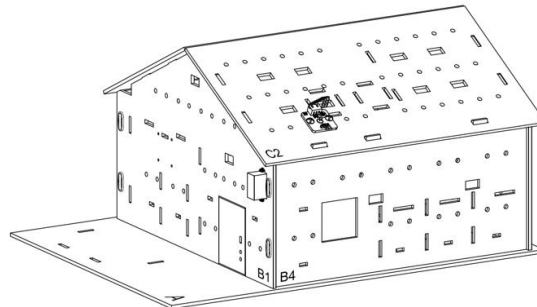
Attach the light sensor onto model C2 using M4 \* 10mm screws and nuts.



Part list:



Assembly Completed!



# CODING

💡 How can we use distance sensor to detect traffic status?

```
on start
  set curtainOn to false
  initialize OLED with width 128 height 64
```

```
function OpenCurtain
  Turn 360° Servo with clockwise direction speed Level 3 at P2
  pause (ms) 6000
  Turn 360° Servo with clockwise direction speed Stop at P2
  set curtainOn to true
```

## STEPS:

1. On start, initialize the OLED display by **initialize OLED with width 128 height 64**
2. Create the **curtainOn** variable and set it to false
3. Create function **openCurtain**
4. Inside the function, control the speed and direction of the 360 degree servo at connected pins, such as **Turn 360 Servo with clockwise direction speed level 3 at P2**
5. Add **pause** to wait it rotate for few second (depend on your model setup)
6. Stop the 360 servo with same method, such as **Turn 360 Servo with clockwise direction speed level 0 at P2**
7. set the **curtainOn** variable to true



```
function CloseCurtain  
  Turn 360° Servo with anti-clockwise direction speed Level 3 at P2  
  pause (ms) 8000  
  Turn 360° Servo with anti-clockwise direction speed Stop at P2  
  set curtainOn to false
```

## STEPS:

8. Create function `closeCurtain`
9. Inside the function, following the previous function, but in reversed direction and state
10. control the speed and direction of the 360 degree servo at connected pins, such as `Turn 360 Servo with anti-clockwise direction speed level 3 at P2`
11. Add `pause` to wait it rotate for few second (depend on your model setup)
12. Stop the 360 servo with same method, such as `Turn 360 Servo with anti-clockwise direction speed level 0 at P2`
13. set the `curtainOn` variable to `false`

# CODING



How can we use distance sensor to detect traffic status?

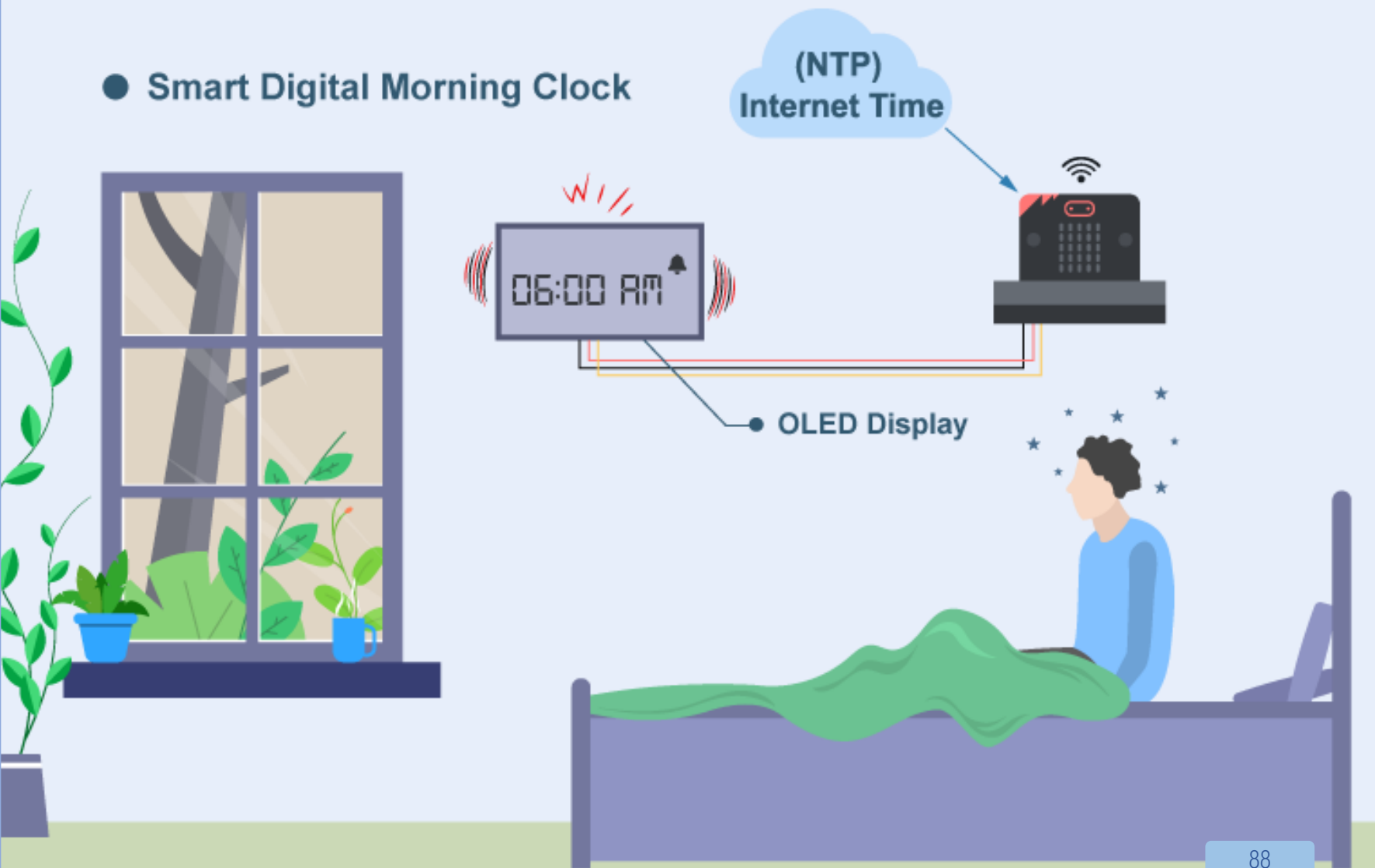
## STEPS:

14. In **Forever**, reading the value by **set light to Get light value (percentage) at Pin P1**
15. Clear the OLED display before each update by **clear OLED display**
16. Show the number of value on display by **show number light**
17. Snap a nested **if** statement to **Forever**
18. Set **light2 >= 70** and **curtainOn = true** as first condition
19. In the **if** segment, that's means the sunlight is strong, need to close the curtain, **call closeCurtain**
20. In the second condition, use **light2 < 40** and **curtainOn = false**
21. In the second **if** segment, that's means the sunlight is weak, need to open the curtain, **call openCurtain**

```
forever
  set light to Get light value (percentage) at Pin P1
  clear OLED display
  show number light
  if light >= 70 and curtainOn = true then
    call CloseCurtain
  else if light < 40 and curtainOn = false then
    call OpenCurtain
```

The code block is a 'forever' loop containing several steps. Step 14 is 'set light to Get light value (percentage) at Pin P1'. Step 15 is 'clear OLED display'. Step 16 is 'show number light'. Step 17-18 is an 'if' statement with two conditions: 'light >= 70 and curtainOn = true'. Step 19 is 'call CloseCurtain'. Step 20 is an 'else if' statement with two conditions: 'light < 40 and curtainOn = false'. Step 21 is 'call OpenCurtain'. The code blocks are numbered 14 through 21 on the left side.

## ● Smart Digital Morning Clock



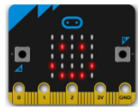
# 9. [IOT] SMART DIGITAL MORNING CLOCK

LEVEL: ★ ★ ★ ★ ☆

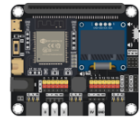
## Introduction

Smart digital clock is a clock that shows time as accurate as to atomic clock by connecting to the Internet.

## PART LIST



micro:bit (1)



IoT:bit with OLED (1)



4-pin extension wire (1)



M2\*10mm Screw (4)



M2 Nut (4)



Screwdriver (1)



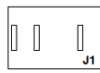
Model A - C (1)  
[Home Model Base]



Model E4 (1)  
[Sensor Display Wall]



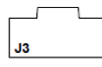
Model E3 (1)  
[Display Wall]



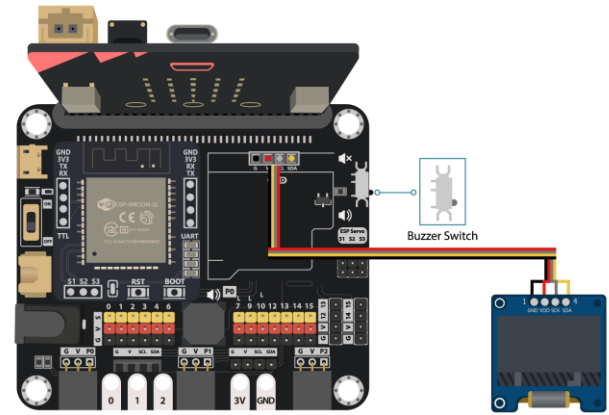
Model J1 (1)  
[Bed]



Model J2 (1)  
[Bed]

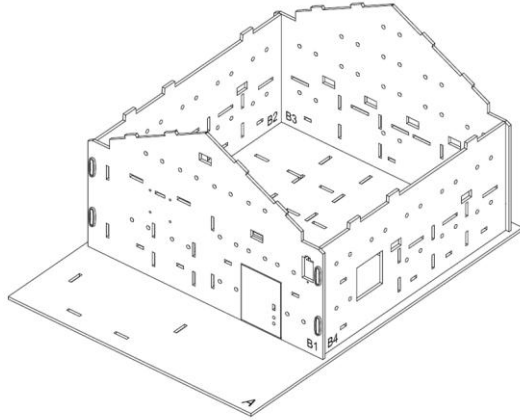


Model J3 (2)  
[Bed]

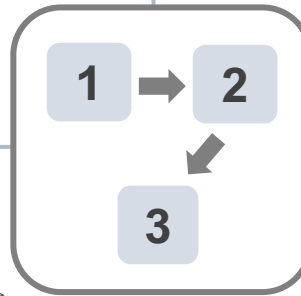
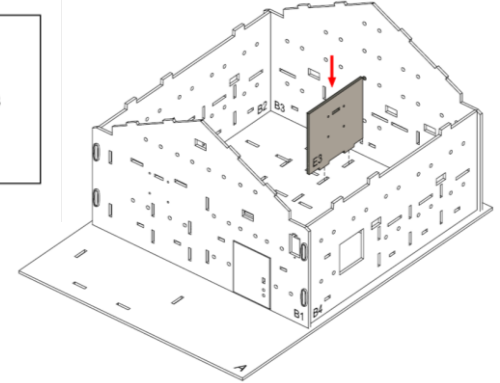
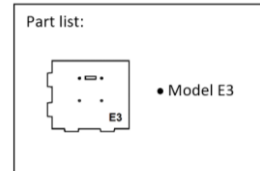


# ASSEMBLY STRUCTIONS

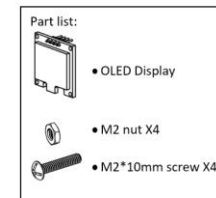
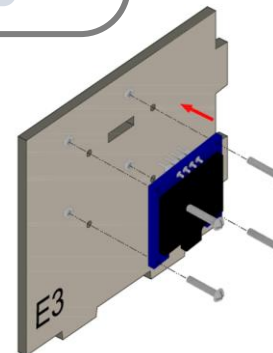
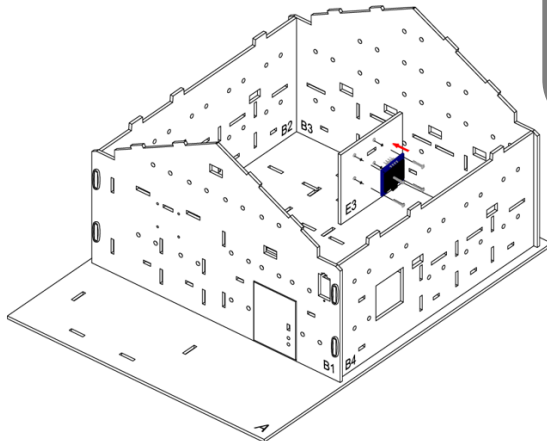
In this case, build the “Big Style Model” as a home base.



Let's build a bed room to install a digital clock. Put the model E3 onto cardboard A, align with the holes at model A and B3.

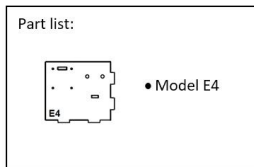
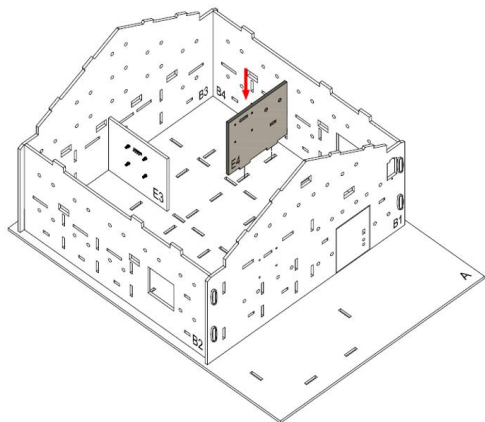


Attach the OLED display as a digital clock to the model E3 using M2 \* 10mm screws and nuts.

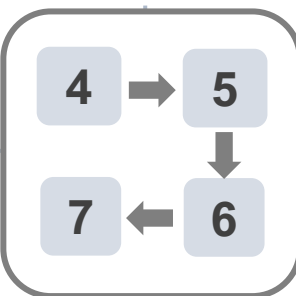
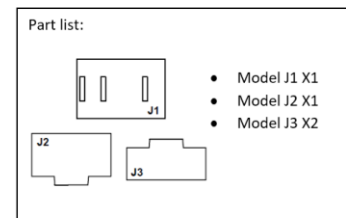
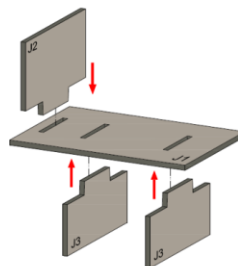


# ASSEMBLY STRUCTIONS

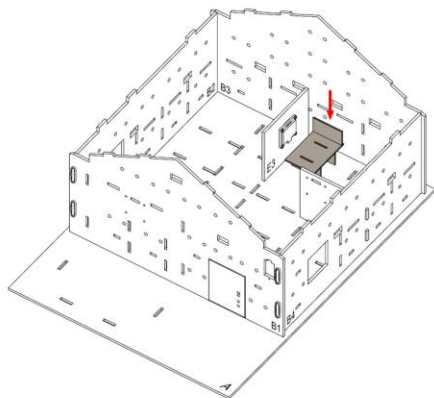
Put the model E4 onto cardboard A, align with the holes at model A and B4.



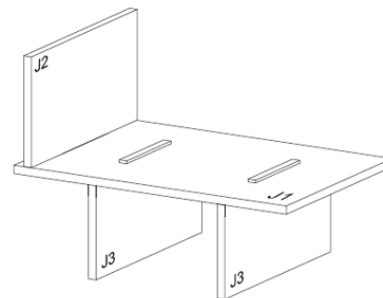
To build a bed. Put model J2 onto the model J1, then put together all the cardboard parts (J1-J3).



Place the bed next to the digital clock at the room.



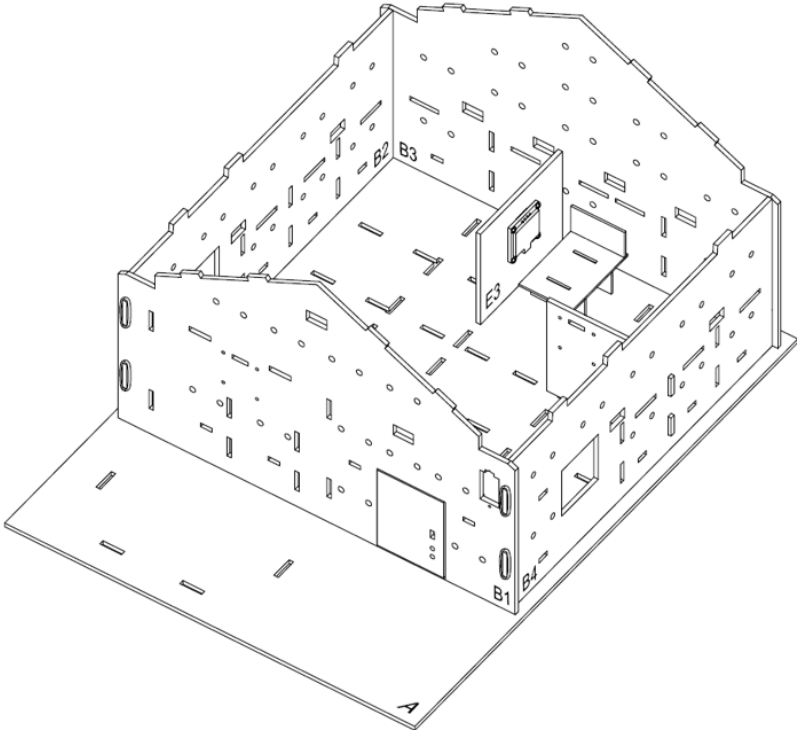
The bed completed!



# ASSEMBLY STRUCTIONS

---

Assembly Completed!



# CODING



Apart from showing the time and alarm, any other tasks can be done when we know the time?

```
on start
1 initialize OLED with width 128 height 64
2 Initialize IoT:bit TX P16 RX P8
3 Set WiFi to ssid "smarthon_ss8" pwd "Qw64811051"
5 set alarmTrig to false

6 On WiFi connected IP_Address Device_ID
7 show icon

forever
8-9 if WiFi connected? then
10 Get NTP Current Time at city HongKong (UTC+8)
11 pause (ms) 500
```

## STEPS:

1. Snap **Initialize OLED with width 128, height 64** to **on start**
2. Snap **Initialize IoT:bit TX P16 RX P8** to **on start**
3. Snap **Set Wi-Fi to ssid pwd** to **on start** and enter the SSID and password
4. Create the variable **alarmTrig**
5. Set the **alarmTrig** variable to **false**
6. Snap the **On WiFi connected** block to editor
7. Put a **show icon tick** inside the block to show tick icon after connected
8. Put a **if** statement in **Forever**
9. Use **WiFi connected?** as condition
10. In **if** segment, put **Get NTP Current Time at city HongKong(UTC+8)** to get NTP time
11. **Pause 500ms** to get the NTP time with each 0.5 second

# CODING



Apart from showing the time and alarm, any other tasks can be done when we know the time?

The code editor shows the following sequence of blocks:

- 12** On NTP received (Event block) with sub-blocks: Year, Month, Day, Hour, Minute, Second.
- 14** clear OLED display (Control block).
- 15** show string (Display block) with:
  - join (Text block) containing "Date: "
  - Day (Text block)
  - ":" (Text block)
  - Month (Text block)
  - ":" (Text block)
  - Year (Text block)
  - minus and plus (Text block)
- insert newline (Control block).
- 16** show string (Display block) with:
  - join (Text block) containing "Time: "
  - Hour (Text block)
  - ":" (Text block)
  - Minute (Text block)
  - ":" (Text block)
  - Second (Text block)
  - minus and plus (Text block)

## STEPS:

12. Put **On NTP received** to editor
13. When get the NTP time, the program in this block will running
14. Clear the display before each update by **Clear OLED display**
15. Show the formatted time string with **show string join Date: day.Month.Year**
16. Show the formatted time string with **show string join Time: Hour:Minute:Second**

# CODING



Apart from showing the time and alarm, any other tasks can be done when we know the time?

17. if Hour = 6 and Minute = 0 and Second = 0 then

18. set alarmTrig to true

19. forever

20-21. if alarmTrig = true then

22. play melody at tempo 500 (bpm)

else

23. stop all sounds

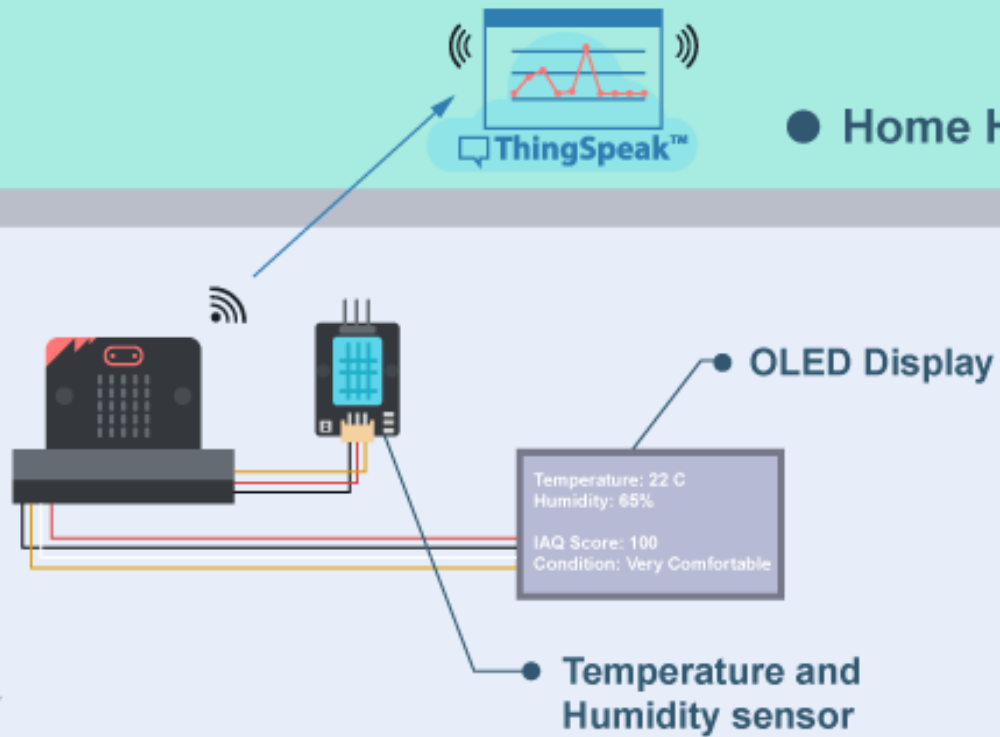
24. on button A pressed

25. set alarmTrig to false

## STEPS:

17. Put a **if** statement with multiple conditions  
**Hour = 6 and Minute = 0 and Second = 6**
18. Put **set alarmTrig to true** in if segment
19. Put another **Forever** to editor
20. Put a **if-else** statement inside the **Forever**
21. Use **alarmTrig = true** as condition
22. Play the alarm by **melody at tempo 500 bpm**
23. Otherwise, stop the alarm
  
24. Put **on Button A pressed** to editor
25. Put **set alarmTrig to false** into the block

## ● Home Health Data Monitoring



# 10. [IOT] HOME HEALTH DATA MONITORING

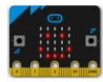
LEVEL: ★ ★ ★ ★ ☆



## Introduction

Health monitoring is the concept of tracking the conditions of different parameters inside the user's house. For indoor, the important parameters to analyze the home comfortability are temperature, humidity, pm2.5, CO2 and VOC. In this system, we can base on temperature and humidity to measure the comfortability of the home. With the too high temperature or too low humidity will also affect the comfortability.

## PART LIST



micro:bit (1)



IoT:bit with OLED (1)



Temperature and Humidity Sensor (1)



Screwdriver (1)



3-pin module wire (1)



4-pin extension wire (1)



M4\*10mm Screw (2)



M4\*10mm Nut (2)



M2\*10mm Screw (4)



M2 Nut (4)



Model A - C (1)  
[Big House Model]



Model E4 (1)  
[Sensor Display Wall]



Model E1 (1)  
[Long Sensor Wall]



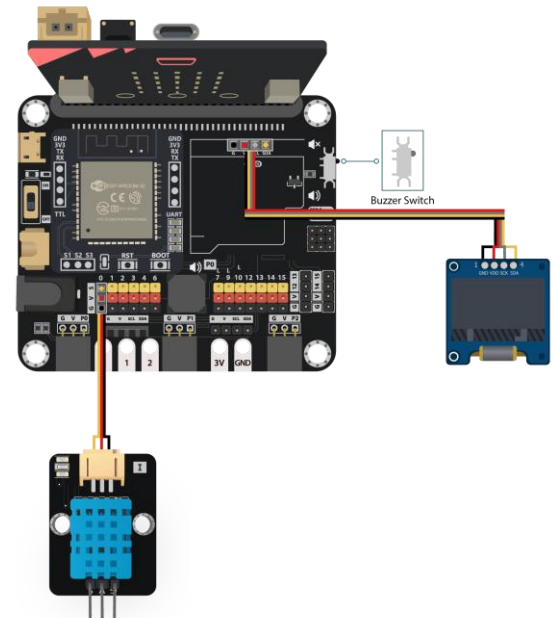
Model K1 (1)  
[Sofa]



Model K2 (1)  
[Sofa]

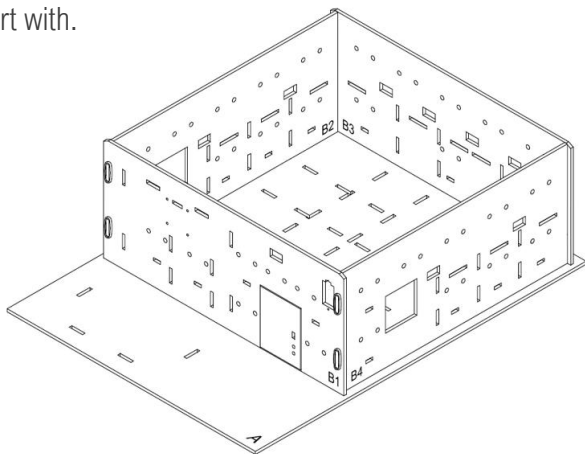


Model K3 (2)  
[Sofa]

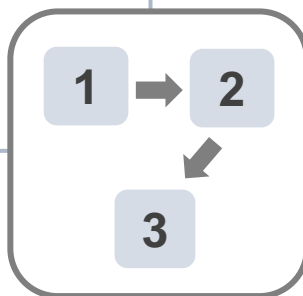
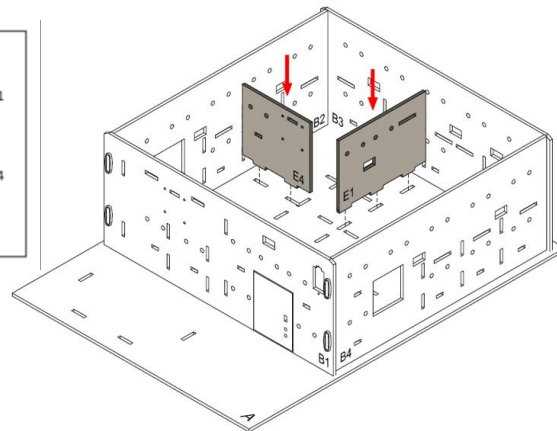
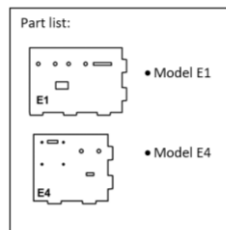


# ASSEMBLY STRUCTIONS

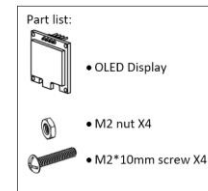
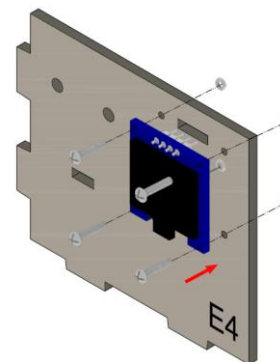
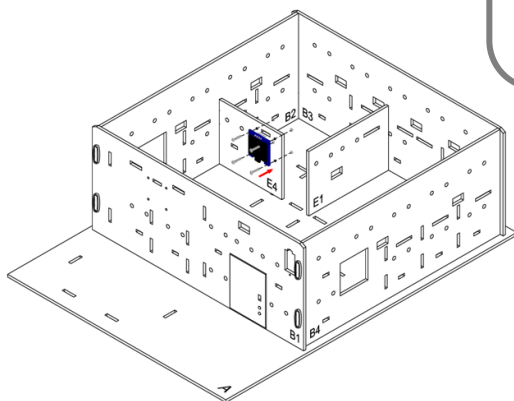
In this case, "Open Style Model" is used as a home base to start with.



Put the model E4 & E1 onto model A, align with holes at model A, B2 and B3.

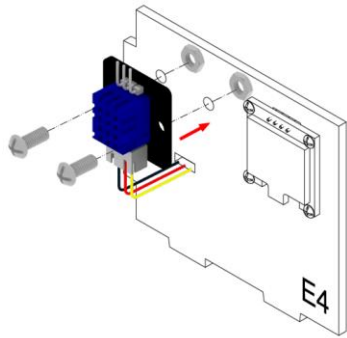


Attach the OLED display onto model E4 using M4 \* 10mm screws and nuts.



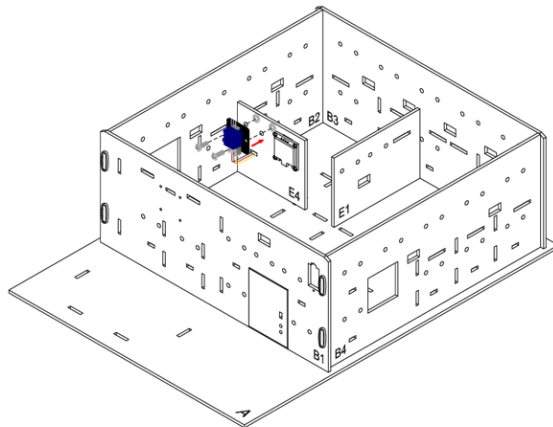
# ASSEMBLY STRUCTIONS

Attach the DHT11 temperature and humidity sensor onto model E4 using M2 \* 10mm screws and nuts. And the connecting wire should be bended to the hole below it.

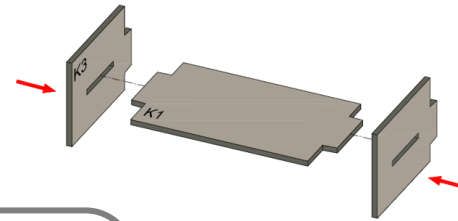


Part list:

- Temperature and Humidity Sensor
- M4\*10mm screw X2
- M4 nut X2

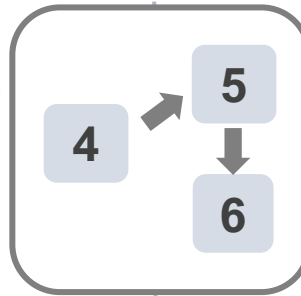


To build a sofa model. Put the model K3 to the two sides of model K1.

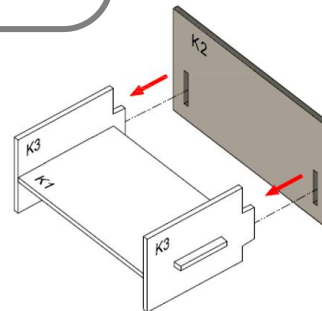


Part list:

- Model K1
- Model K3 X2



Put model K2 all together to the cardboard parts (K1-K3).

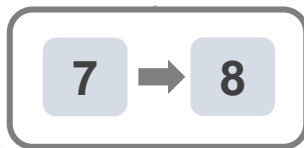
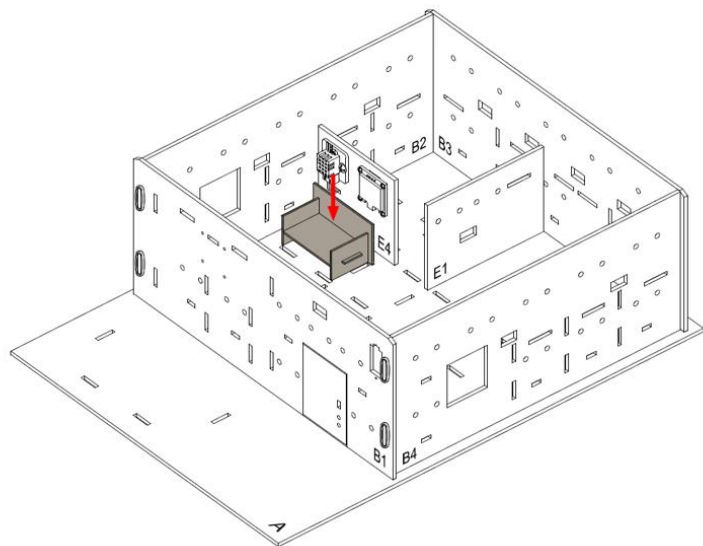


Part list:

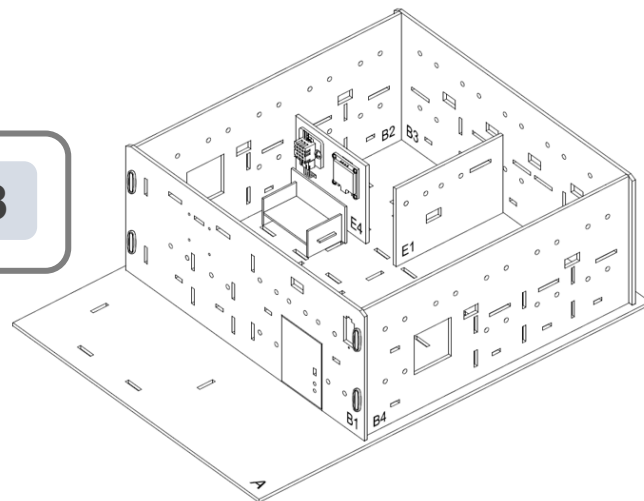
- Model K2

# ASSEMBLY STRUCTIONS

Plase the sofa model behind the model E4.



Assembly Completed!

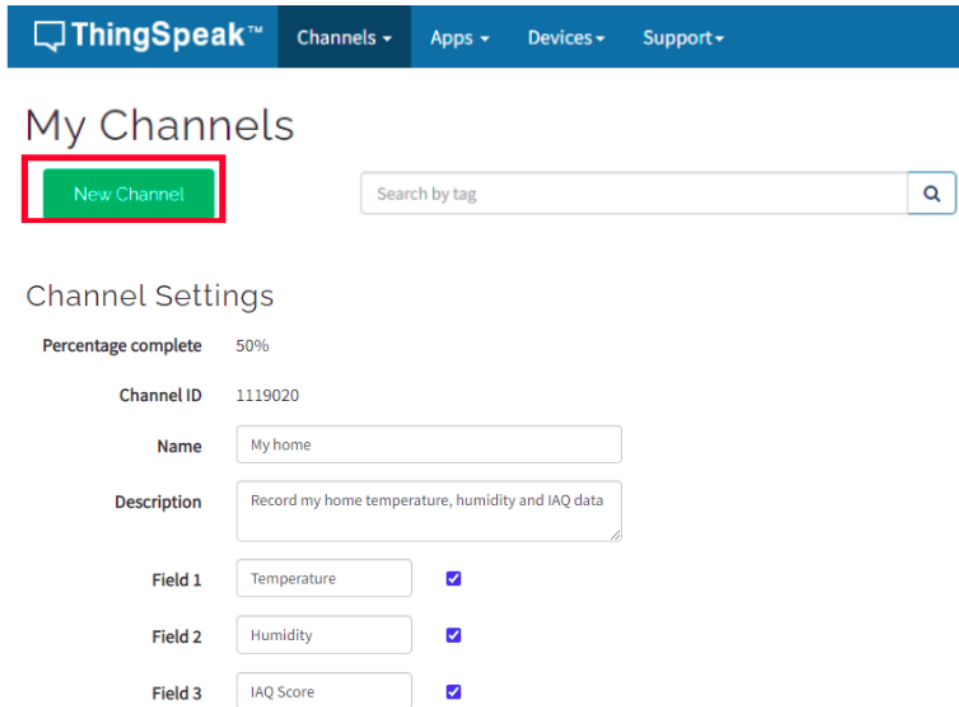


# IOT – THINGSPEAK

---

Step 1. Create Thingspeak channel

Go the Thingspeak create an account and create a channel



The screenshot shows the Thingspeak user interface. At the top is a dark blue navigation bar with the Thingspeak logo and menu items: Channels, Apps, Devices, and Support. Below the navigation bar is the 'My Channels' section. A green 'New Channel' button is highlighted with a red rectangular box. To its right is a search bar labeled 'Search by tag' with a magnifying glass icon. Below this is the 'Channel Settings' section. It displays 'Percentage complete' at 50%. The 'Channel ID' is 1119020. The 'Name' field contains 'My home'. The 'Description' field contains 'Record my home temperature, humidity and IAQ data'. There are three 'Field' settings, each with a text input and a checked checkbox: 'Field 1' is 'Temperature', 'Field 2' is 'Humidity', and 'Field 3' is 'IAQ Score'.

# IOT – THINGSPEAK

---

## Step 2. Get channel API

After created a new channel, get the write in API

### My home

Channel ID: **1119020**  
Author: **tester**  
Access: Public

Record my home temperature, humidity and IAQ data

[Private View](#) [Public View](#) [Channel Settings](#) [Sharing](#) **[API Keys](#)** [Data Import / Export](#)

### Write API Key

Key

[Generate New Write API Key](#)

# CODING



Apart from temperature and humidity, what elements do you think are essential to be included in a home health monitoring system? (E.G CO2 level? PM2.5 value?)

```
on start
  initialize OLED with width 128 height 64
  Initialize IoT:bit TX P16 RX P8
  Set WiFi to ssid "smarthon" pwd "12345678"
  set humidity to 0
  set temperature to 0
  set iaq_score to 0

On WiFi connected IP_Address Device_ID
  show icon
```

## STEPS:

1. Snap **Initialize OLED with width:128, height: 64** to **on start**
2. Snap **Initialize IoT:bit TX P16 RX P8** from IoT:bit to **on start**
3. Snap **Set WiFi to ssid smarthon pwd 12345678** to **on start**
4. Create and initialize the variables **temperature**, **humidity** and **iaq\_score** to 0
5. In **On WiFi connected**, put a **show icon** tick get notice after WiFi is connected.

# CODING



Apart from temperature and humidity, what elements do you think are essential to be included in a home health monitoring system? (E.G CO2 level? PM2.5 value?)

```
6 function check_condition
7-8 if iaq_score < 20 then
  show string "Very uncomfortable"
9 else if iaq_score < 40 then
  show string "Uncomfortable"
10 else if iaq_score < 60 then
  show string "Discomfortable"
11 else if iaq_score < 80 then
  show string "Comfortable"
12 else if iaq_score ≤ 100 then
  show string "Very Comfortable"
```

## STEPS:

6. Create a new function `check_condition`
7. Put nested `if-else` statement inside the function
8. The first condition is `iaq_score < 20`, then `show string Very uncomfortable` in the segment
9. The second condition is `iaq_score < 40`, then `show string Uncomfortable` in the segment
10. The third condition is `iaq_score < 60`, then `show string Discomfortable` in the segment
11. The fourth condition is `iaq_score < 80`, then `show string Comfortable` in the segment
12. The fifth condition is `iaq_score ≤ 100`, then `show string Very comfortable` in the segment

# CODING



Apart from temperature and humidity, what elements do you think are essential to be included in a home health monitoring system? (E.G CO2 level? PM2.5 value?)

## STEPS:

13. In **Forever**, put a **if** statement with condition **WiFi connected?** to make sure connected to internet before uploading to thingspeak
14. Use **Read Temperature & Humidity sensor at pin P0** to read the data from DHT11 Temperature and Humidity Sensor
15. After read the DHT11 data, use the three functions to get the result and set to the variable
16. **Set temperature to Get Temperature °C** for temperature
17. **Set humidity to Get Humidity** for humidity
18. **Set iaq\_score to Get IAQ Score** for iaq\_score

```
forever
  if WiFi connected? then
    Read Temperature & Humidity Sensor at pin P0
    set temperature to Get Temperature °C
    set humidity to Get Humidity
    set iaq_score to Get IAQ Score
```

# CODING



Apart from temperature and humidity, what elements do you think are essential to be included in a home health monitoring system? (E.G CO2 level? PM2.5 value?)

```
19. clear OLED display
21. show string join "Temperature: " temperature
22. show string join "Humidity: " humidity
23. show string join "IAQ Score: " iaq_score
24. call check_condition
25. Send Thingspeak key* "CQZ89"
    field1 value temperature
    field2 value humidity
    field3 value iaq_score
26. pause (ms) 15000
```

## STEPS:

19. Clear the display before each times update by **clear OLED display**
20. Show the three variables respectively, with some text explanation
21. **show string join Temperature: temperature** for temperature
22. **show string join Humidity: humidity %** for humidity
23. **show string Join IAQ Score: iaq\_score** for iaq\_score
24. Run the **check\_condition** function to summarize the IAQ Score result
25. Send the data to Thingspeak by **Send Thingspeak key XXXX field1 value XXX ...**, fill in the write API key from the Thingspeak channel and the values need to be upload
26. After uploading the data to Thingspeak, wait for 15 second to avoid upload too frequently by **pause(ms) 15000**, then start another Reading and uploading.

# CODING



Apart from temperature and humidity, what elements do you think are essential to be included in a home health monitoring system? (E.G CO2 level? PM2.5 value?)

```
27. On Thingspeak Uploaded Status Error_code
28. insert newline
30. show string join "Thingspeak:" Status
31. show string join "Error:" Error_code
```

## STEPS:

27. To check the uploading status, use **On thingspeak Uploaded** to get the uploading result
28. **Insert newline** for better visual effect
29. Use the **Status** and **Error\_code** from block placeholder respectively to showing some text explanation
30. **show string join Thingseak: Status** for Upload status
31. **show string join Error: Error\_code** for Error code if upload failed



# ● Kitchen Safety Flame Monitoring Alert

Flame Sensor ●

Buzzer ●

Multi-colour LED ●



# 11. [IOT] KITCHEN SAFETY FLAME MONITORING ALERT

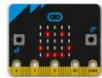
LEVEL: ★ ★ ★ ★ ☆

# IFTTT

## Introduction

Nowadays, fire accidents always happen when the homeowner is sleeping or outside the home. It is good to install an IOT system in the kitchen or living room to alert the user on his/her phone when he/she is sleeping or in the office so that he/she will wake up and rush out from the home to prevent a terrible accident happened.

## PART LIST



micro:bit (1)



IoT:bit with OLED (1)



Flame Sensor (1)



Multi-colour LED (WS2812) (1)



Light Sensor (1)



3-pin module wire (2)



M4\*10mm Screw (4)



M4\*10mm Nut (4)



Screwdriver (1)



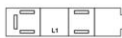
Model A - C (1)  
[Big House Model]



Model E1 (1)  
[Long Sensor Wall]



Model E3 (1)  
[Display Wall]



Model L1 (1)  
[Stove]



Model L2 (1)  
[Stove]



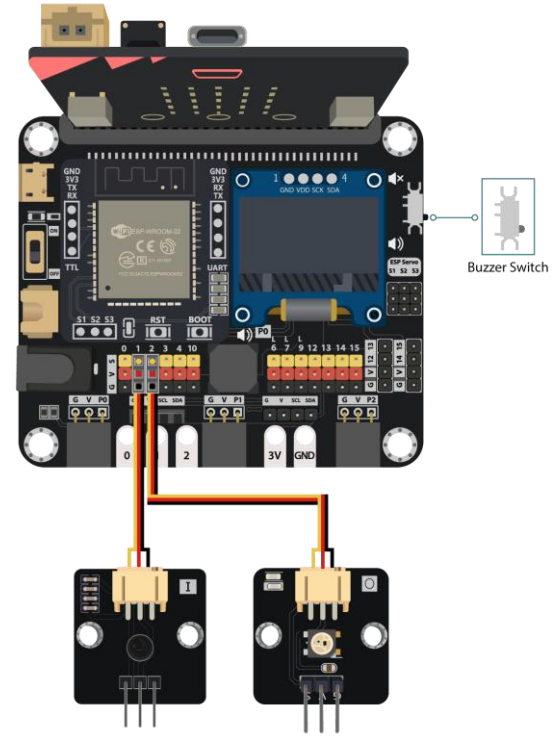
Model G1 (1)  
[Fridge]



Model G2 (1)  
[Fridge]

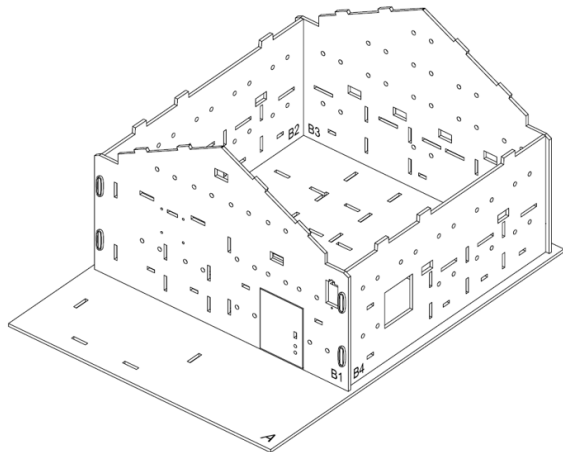


Model G3 (1)  
[Fridge]

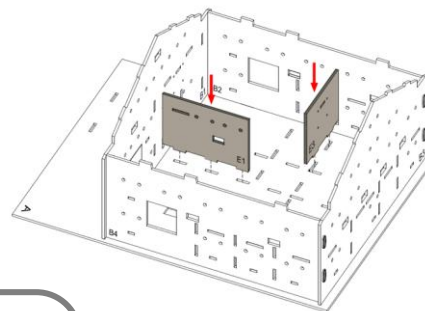


# ASSEMBLY STRUCTIONS

In this case, build the “Big Style Model” as a home base.



To build a kitchen, put model E3 & E1 onto model A, align with holes at model A and B2 & B1.



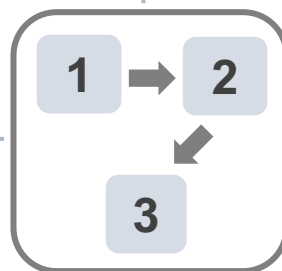
Part list:



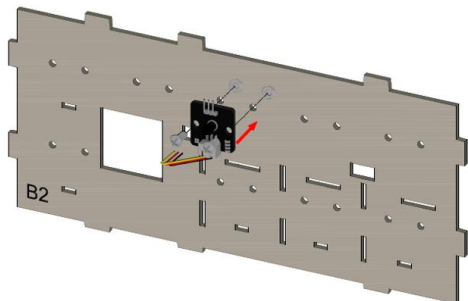
• Model E1



• Model E3



Attached the flame sensor to the model B2 using M4\*10mm screws and nuts. And the connecting wire should be bended to the hole nearby.



Part list:



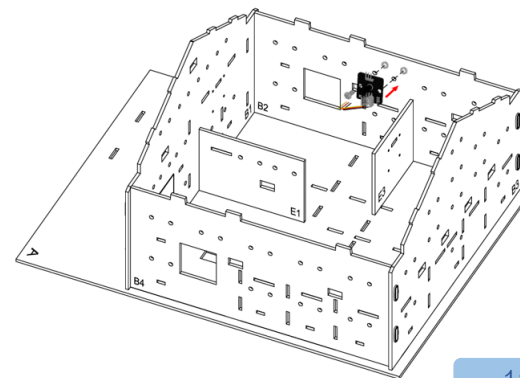
• Flame Sensor



• M4\*10mm screw X2

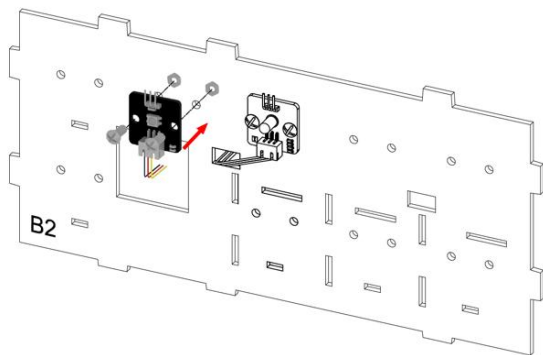


• M4 nut X2



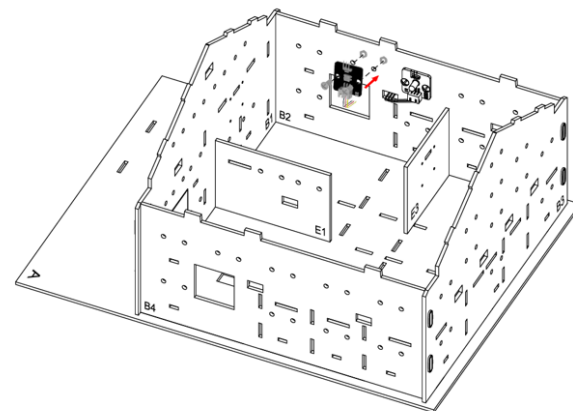
# ASSEMBLY STRUCTIONS

Attached multi-color LED to the model B2 using M4\*10mm screws and nuts. And the connecting wire should be bended to the hole below.

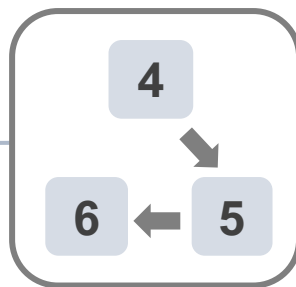
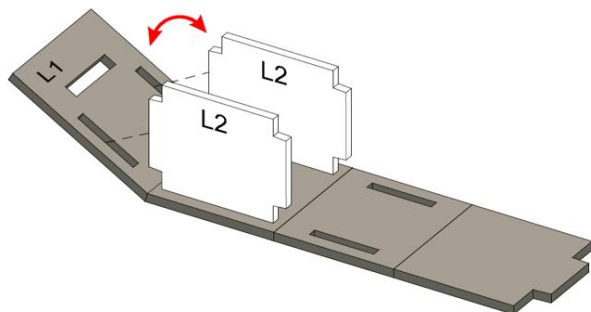


Part list:

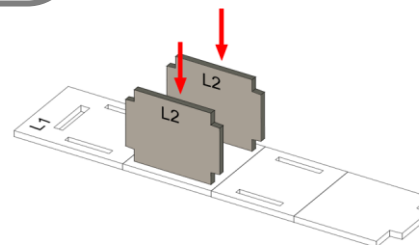
- Multi-color LED
- M4\*10mm screw X2
- M4 nut X2



Fold the model L1 into the model L2 according to the arrow.



To build a stove model. Put the model L2 to model L1 from the top to bottom.

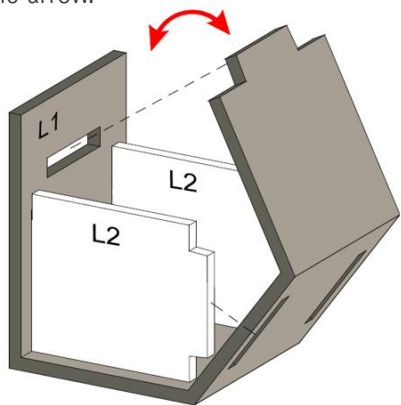


Part list:

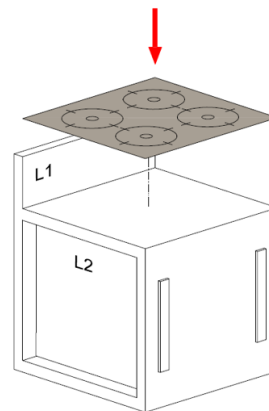
- L1 model
- L2 model X2

# ASSEMBLY STRUCTIONS

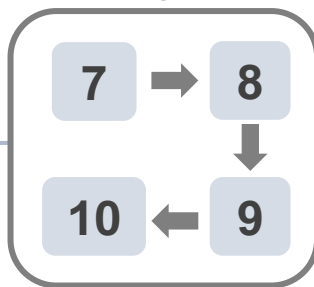
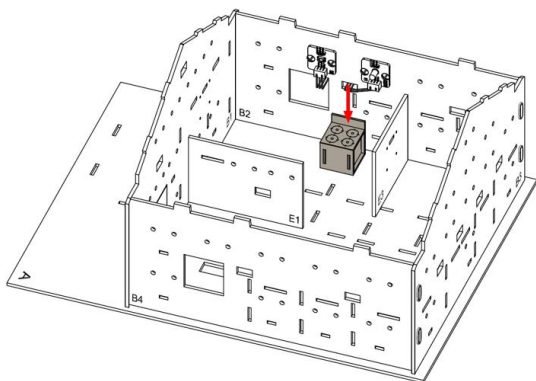
Fold the another side of model L1 into the model L2 according to the arrow.



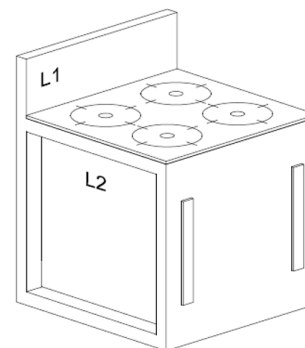
Stick the paper with a cooking stove pattern.



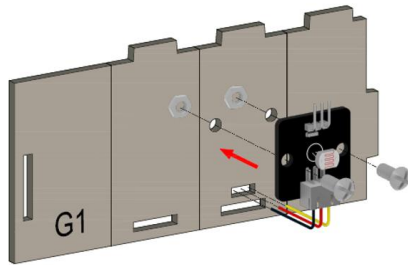
Place the cooking stove model at the kitchen.



The cooking stove Completed!

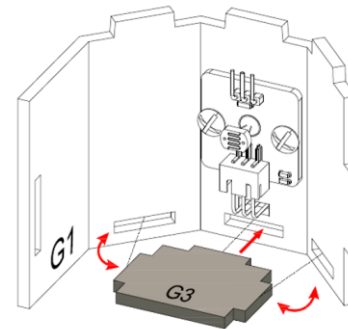


# ASSEMBLY STRUCTIONS



Part list:

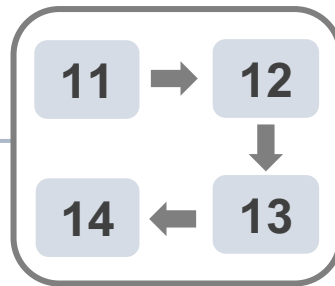
- Model G1
- Light sensor
- M4 screw\*10mm X2
- M4 nut X2



Part list:

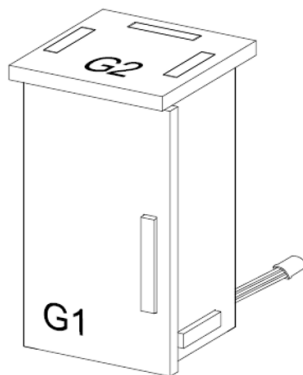
- Model G3

To build a fridge, attach light sensor onto model G1 using M4 \* 10mm screws and nuts. And the connecting wire could be bended to the hole below it.

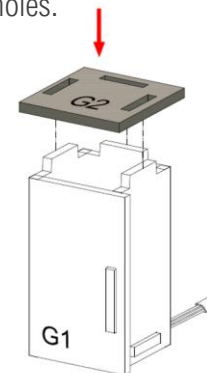


Fold and bend the Model G1 accordingly and put Model G3 into the Model G1, align with the holes.

The fridge completed!



Put Model G2 onto the Model G1, align with the holes.

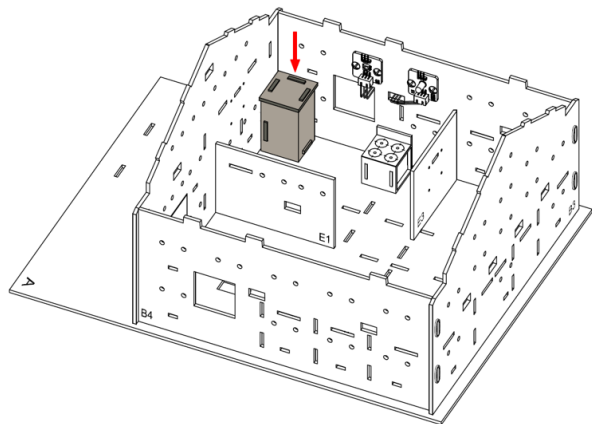


Part list:

- Model G2

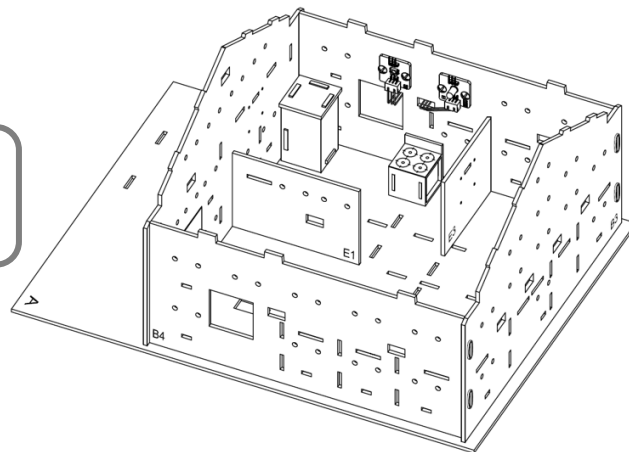
# ASSEMBLY STRUCTIONS

Place the fridge model at the coner of the kitchen.



The fridge Completed!

15 → 16



# CODING



Other than warning, any task we can do when we detect the flame? (e.g add a fan or sprinkler to extinguish the fire, call the police?)

```
on start
  initialize OLED with width 128 height 64
  Initialize IoT:bit TX P16 RX P8
  Set WiFi to ssid "smarthon" pwd "12345678"
  set strip to NeoPixel at pin P1 with 1 leds as RGB (GRB format)
```

- 1
- 2
- 3-4
- 5

## STEPS:

1. Snap **Initialize OLED with width:128, height: 64** to **on start**
2. Snap **Initialize IoT:bit TX P16 RX P8** from IoT:bit to **on start**
3. Snap **Set Wi-Fi to ssid pwd** from IoT:bit
4. Enter your Wi-Fi name and password. Here we set smarthon as SSID and 12345678 as password
5. Snap **Set strip to NeoPixel at pin P1 with 1 leds as RGB(GRB format)**
6. Snap **show icon** from basic to **On WiFi connected** and select icon tick
7. Draw the **Device ID** variable from **On WiFi connected** to the **show string** block placeholder

```
On WiFi connected IP_Address Device_ID
  show icon
  show string Device_ID
```

- 6
- 7

# CODING



Other than warning, any task we can do when we detect the flame? (e.g add a fan or sprinkler to extinguish the fire, call the police?)

The code is written in Scratch and consists of a 'forever' loop. Inside the loop, there is an 'if' statement with the condition 'Wifi connected?'. If true, another 'if' statement checks 'get flame detection at pin P2 = true'. If true, the following actions are performed: play a 'High B' tone for 2 beats until done; set the LED strip color to red; pause for 100ms; set the LED strip color to black; pause for 100ms; and send an IFTTT event named 'fire'. The code is annotated with blue circles containing numbers 8 through 17, corresponding to the steps on the right.

## STEPS:

8. In the Forever, put a **if** statement with condition **WiFi connected?** to check the connection status
9. Put another **if** statement with condition **Get flame detection at Pin P2 = true** to recognize the flame
10. Play a warning sound when the flame was detected
11. The LED should be blinking to telling dangerous
12. Put **strip show color red** to turn on the LED in red color
13. Add a **pause (ms) 100** to wait for 0.1 second
14. Put **strip show color black** to turn off the LED
15. Pause for 0.1 second again
16. Put a **Send IFTTT event\_name\* XXXXX** to send the event to IFTTT
17. Fill in the Applet's event\_name



```
18 On IFTTT Uploaded Status Error_code
20 clear OLED display
21 show string join "IFTTT: " Status
22 show string join "Error: " Error_code
```

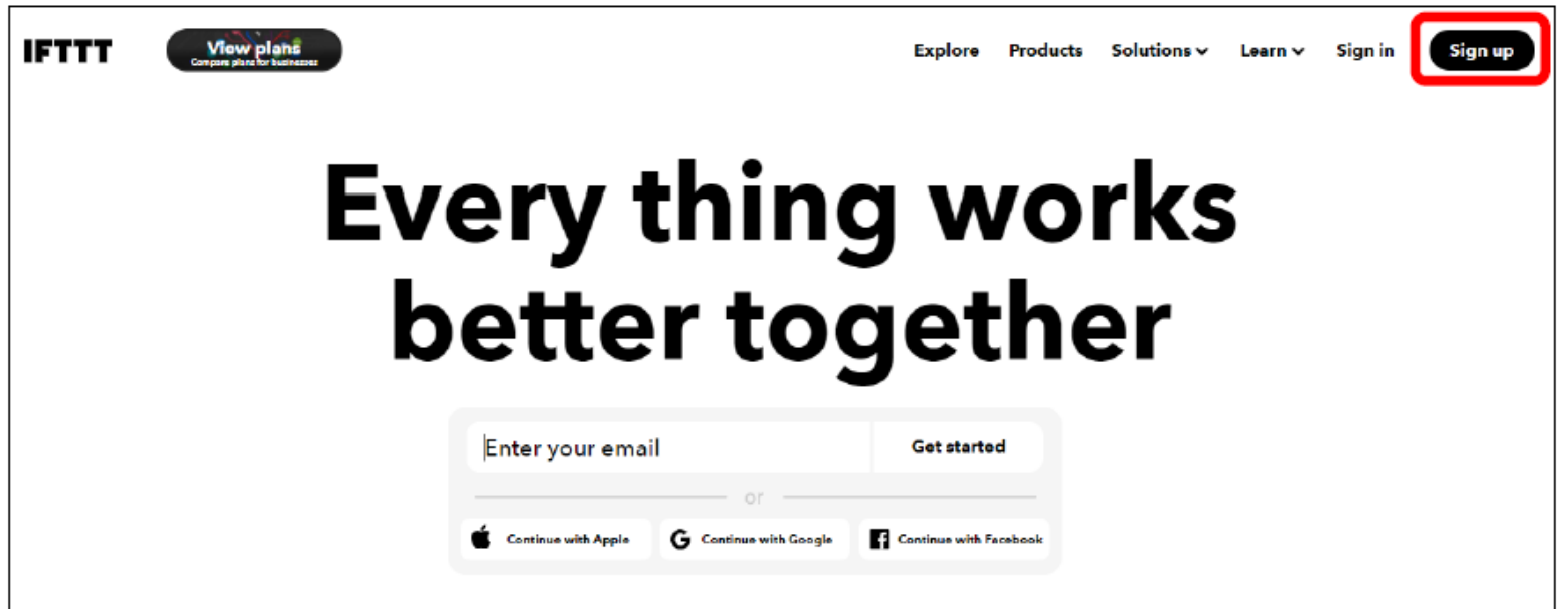
## STEPS:

18. To check the upload state, use the **On IFTTT Uploaded** to get the sending result
19. Inside the On IFTTT Uploaded, use OLED display to show the information
20. Clear the OLED display before each update by **clear OLED display**
21. Show upload state by **show string join IFTTT: Status**, the **Status** value is from the function's placeholder
22. Show error\_code by **show string join Error: Error\_code**, the **Error\_code** value is from the function's placeholder

# IOT – IFTTT

---

Step 1: Go to <http://www.ifttt.com>, register an account and login to the platform

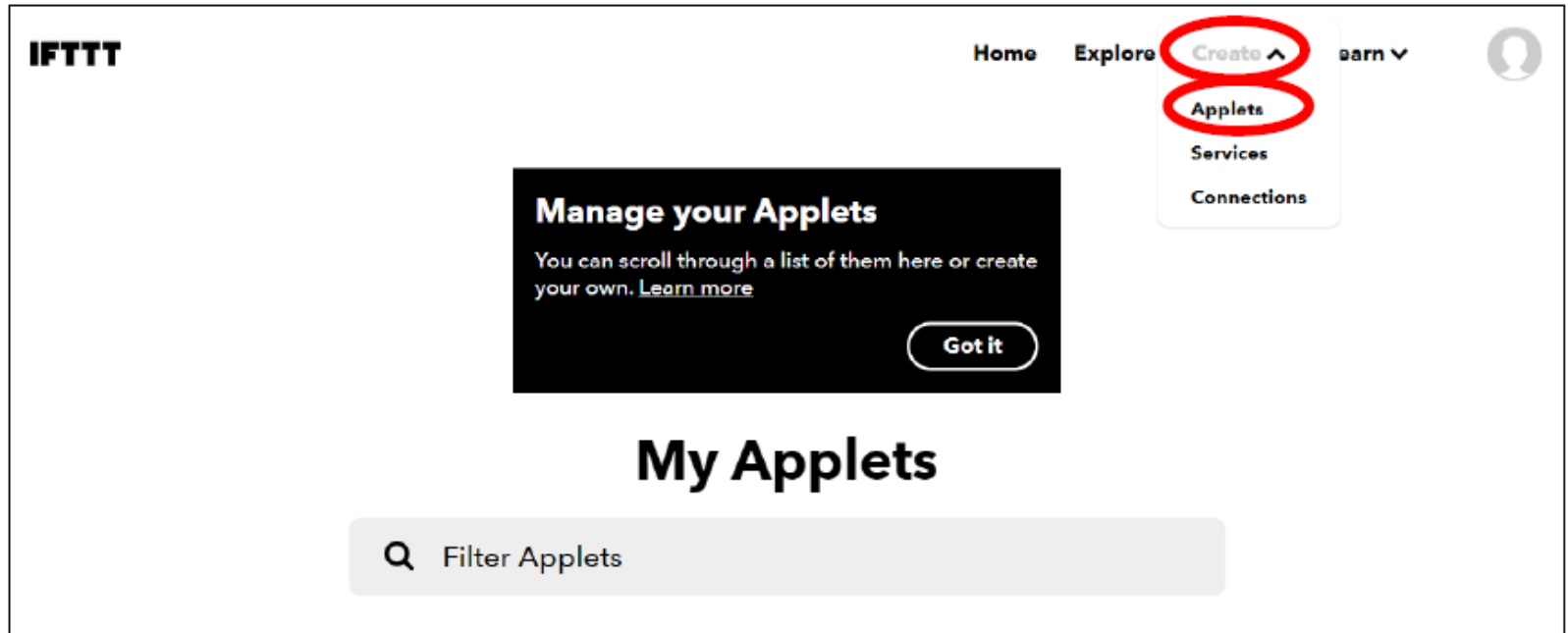


The screenshot shows the IFTTT website homepage. The IFTTT logo is in the top left. A navigation bar at the top right includes links for 'Explore', 'Products', 'Solutions', 'Learn', and 'Sign in'. A 'Sign up' button is highlighted with a red border. The main heading reads 'Every thing works better together'. Below the heading is a sign-up form with an email input field, a 'Get started' button, and social media login options for Apple, Google, and Facebook.

# IOT – IFTTT

---

Step 2: On the top right menu, click “Create” > “Applets”



# IOT – IFTTT

---

## Step 3

1. Select this
2. Select Smarthon IoT
3. Input Device ID and Event Name. (eg. Device ID: 0x55a842e3477a, Event Name: Fire)
4. Click “Create trigger”

## Step 4

1. Select “That”
2. Notifications
3. Send a notification from the IFTTT app
4. Message (eg. There is a fire in the house!), the click “Create action”



This: Smarthon IoT (micro:bit)



That: Notifications

# IOT – IFTTT

## Step 3

1. Select this
2. Select Smarthon IoT
3. Input Device ID and Event Name. (eg. Device ID: 0x55a842e3477a, Event Name: Fire)
4. Click “Create trigger”

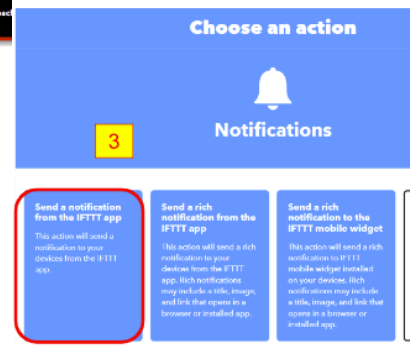
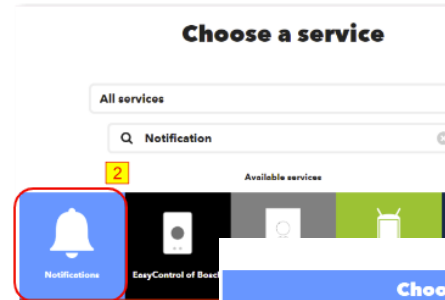
The image displays four sequential screenshots from the IFTTT application, illustrating the process of creating a trigger for a Smarthon IoT device. Each step is numbered 1 through 7, corresponding to the instructions in the text above.

- Step 1:** The 'Create' screen shows the 'If This' button highlighted with a red box and a yellow '1' in the top right corner. The 'Add' button on the 'If This' button is also highlighted with a red box.
- Step 2:** The 'Choose a service' screen shows a dropdown menu with 'All services' selected, highlighted with a yellow '2'. Below it, a search bar contains 'Smarthon IoT', highlighted with a yellow '3'. Underneath, the 'Smarthon IoT (Trigger Event) (micro:bit)' service is highlighted with a red box and a yellow '3'.
- Step 3:** The 'Choose a trigger' screen shows the 'Smarthon IoT (Trigger Event) (micro:bit)' trigger selected, highlighted with a red box and a yellow '4'.
- Step 4:** The 'Edit trigger fields' screen shows the 'Device ID' field with the value '0x55a842e3477a' highlighted with a yellow '5'. The 'Event Name' field with the value 'fire' is highlighted with a yellow '6'. The 'Update trigger' button is highlighted with a red box and a yellow '7'.

# IOT – IFTTT

## Step 4

1. Select “That”
2. Notifications
3. Send a notification from the IFTTT app
4. Message (eg. There is a fire in the house!), the click “Create action”



# IOT – IFTTT

Optional: Use email as notification method

In the THEN field, search for the “email” and use it to replace the “notifications” in previous step

**Create**

If Receive a micro:bit request

+

Then That **Add**

**Choose a service**

All services

Q Email

Available services

**Email**

**Choose an action**

**Email**

**Send me an email**

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

**+**

Suggest a new action

**Complete action fields**

**Send me an email**

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

Subject

Fire

**Add Ingredient**

Body

There is a fire in the house!

**Add Ingredient**

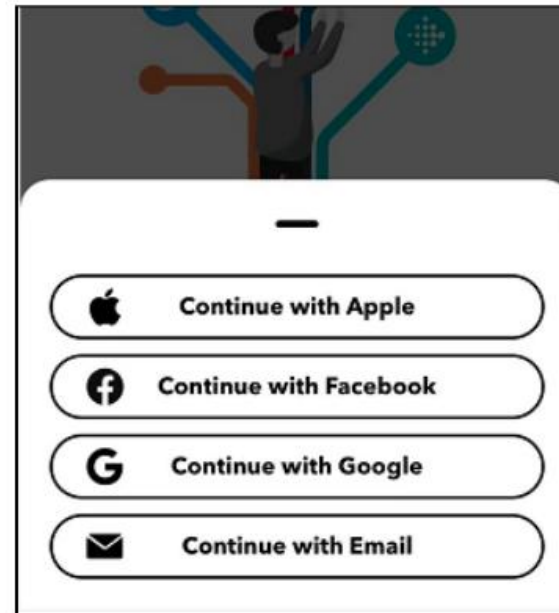
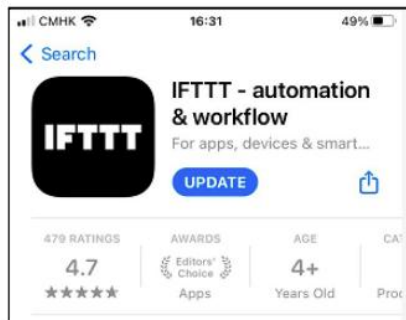
**Create action**

# IOT – IFTTT

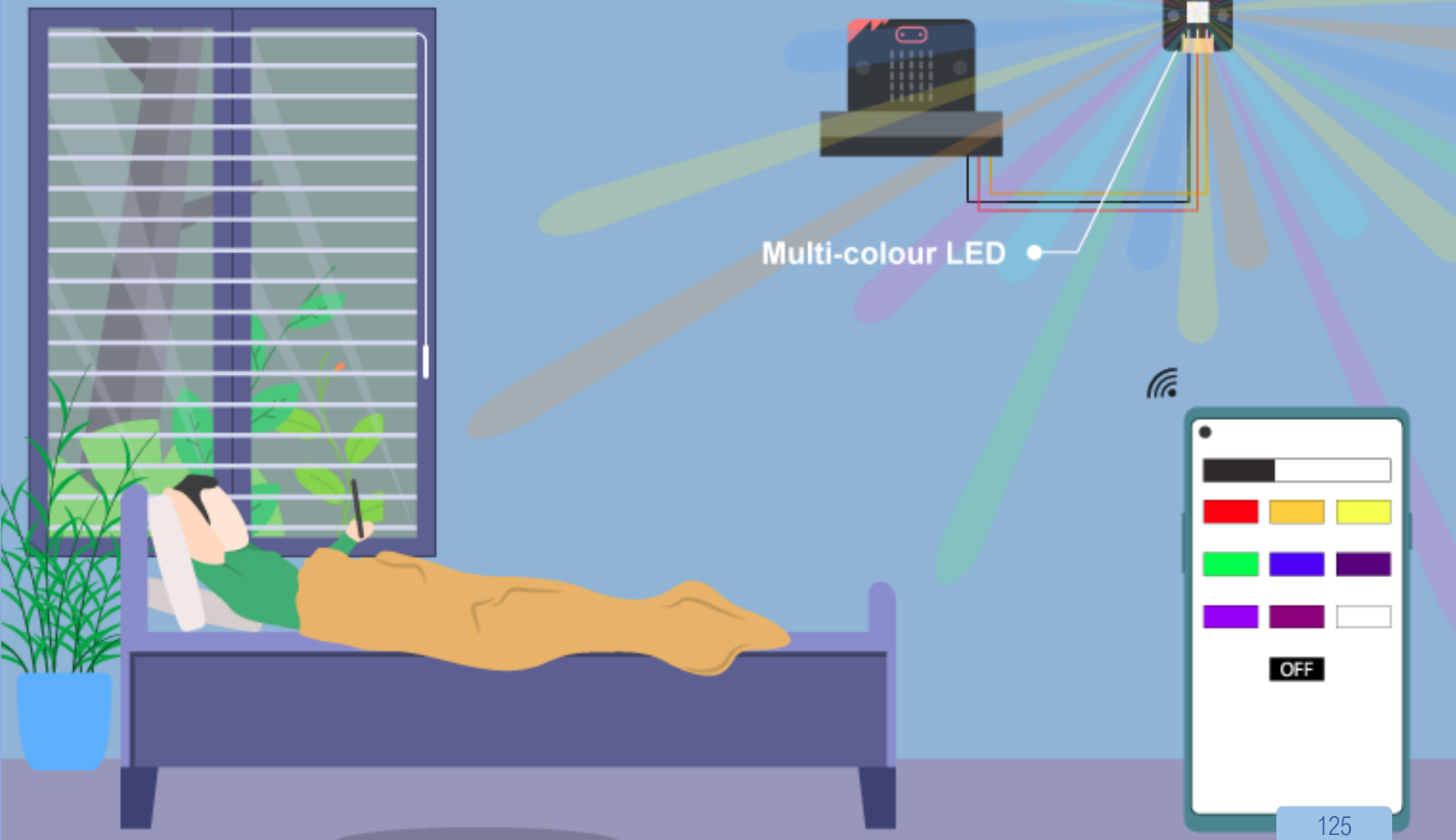
---

## Part 2: Install the IFTTT App on smartphone

1. Go to play store or app store to search and download the IFTTT App
2. Login to your IFTTT account



## ● Room Smart Colourful Light App Control



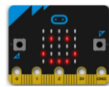
# 12. [IOT] ROOM SMART COLOURFUL LIGHT APP CONTROL

LEVEL: ★ ★ ★ ★ ☆

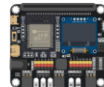
## Introduction

A smart colorful light is a multi-color LED light that allows switching color of light, usually being requested through remote control, for example a mobile app. The user can adjust the color based on different needs. For instance, the user might need dark light when he/she is sleeping, baby feel comfortable when the room in warm color, and user may pick a good color when he/she is reading the book.

## PART LIST



micro:bit (1)



IoT:bit with OLED (1)



Multi-colour LED (WS2812) (1)



3-pin module wire (1)



M4\*10mm Screw (2)



M4\*10mm Nut (2)



Screwdriver (1)



Model A - C (1)  
[Big House Model]



Model E1 (1)  
[Long Sensor Wall]



Model E3 (1)  
[Display Wall]



Model J3 (1)  
[Bed]



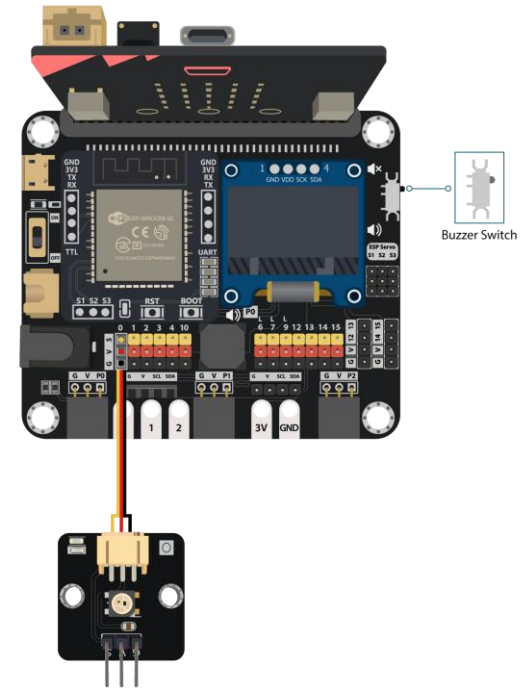
Model J2 (1)  
[Bed]



Model J3 (1)  
[Bed]

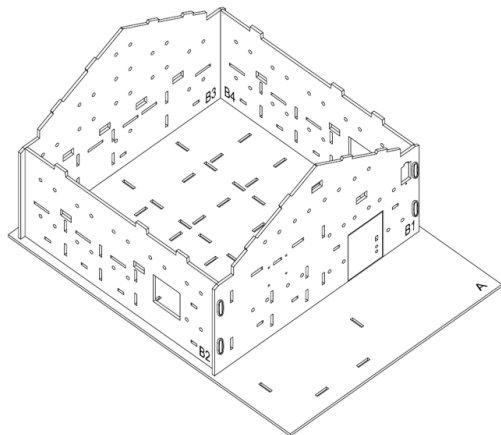


Model H (1)  
[Decoration Display]

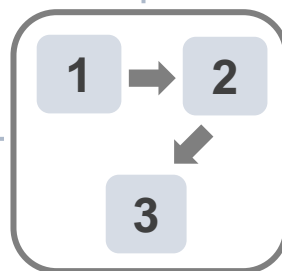
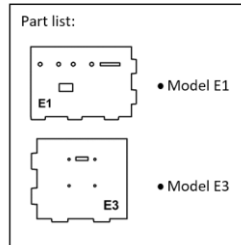
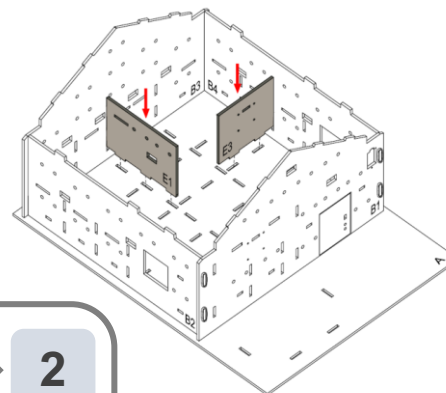


# ASSEMBLY STRUCTIONS

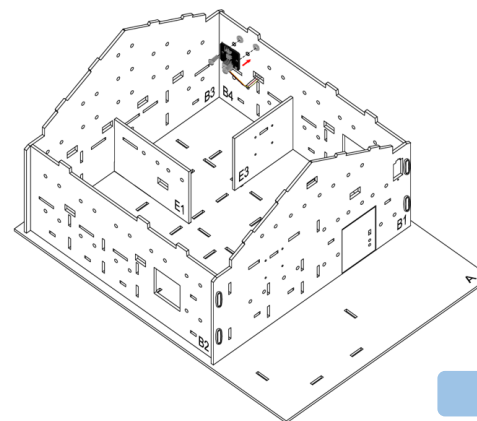
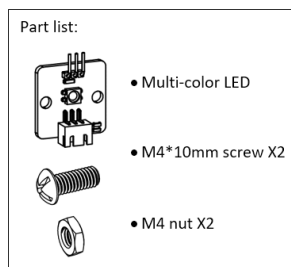
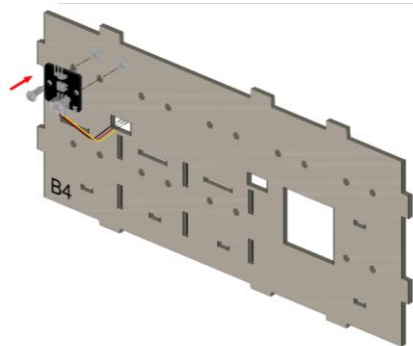
In this case, build the “Big Style Model” as a home base.



To build a room, insert the model E1 and E3 on model A, align with holes on model A, B3 and B4.

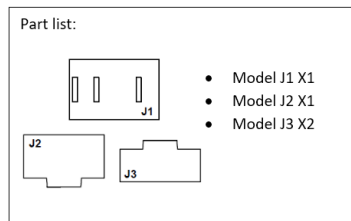
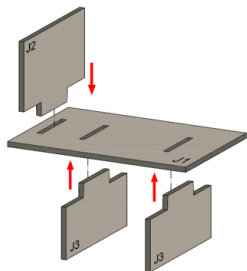


Attach the Multi-color LED to model B4 using M4 \* 10mm screws and nuts.

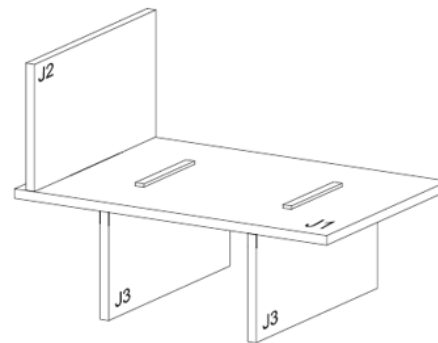


# ASSEMBLY STRUCTIONS

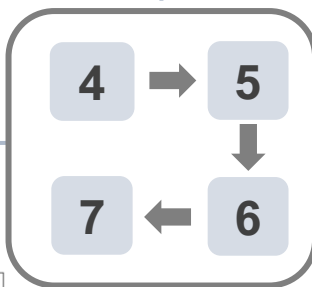
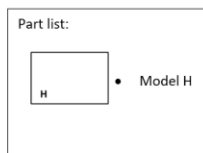
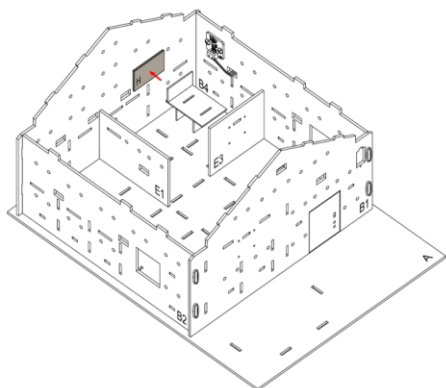
To build a bed. Put model J2 onto the model J1, then put together all the cardboard parts (J1-J3).



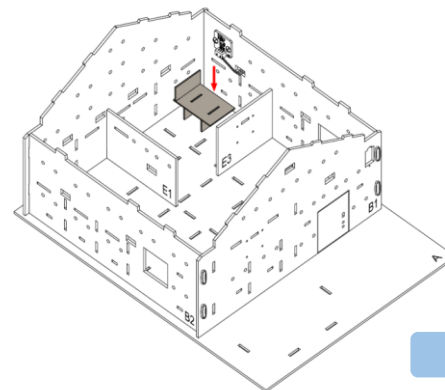
The bed completed!



Place the decoration board (Model H) on model B3.



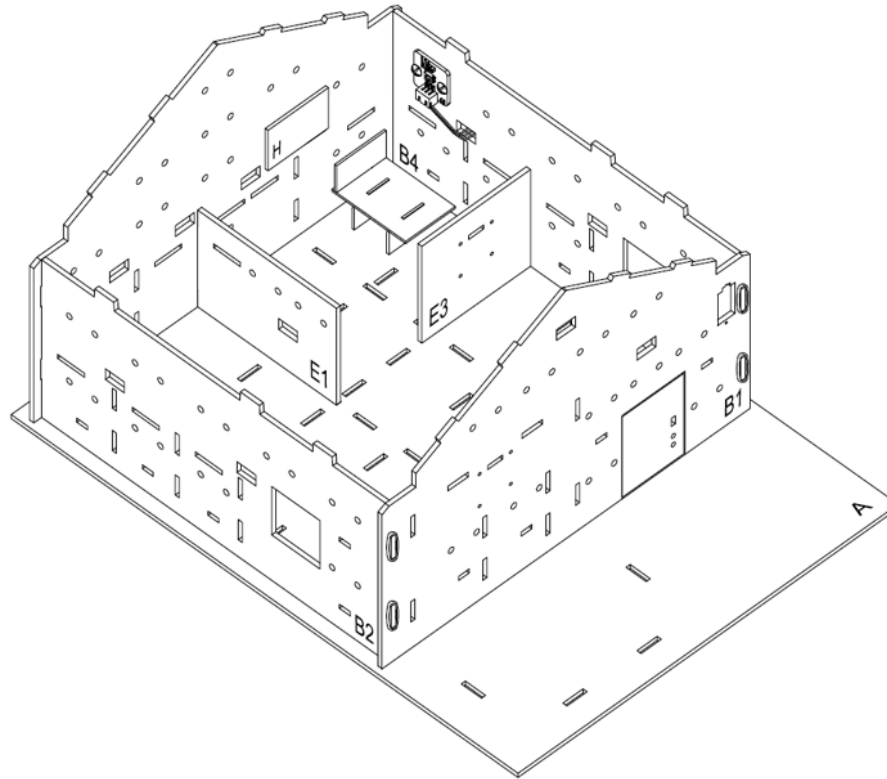
Place the bed inside the bedroom.



# ASSEMBLY STRUCTIONS

---

Assembly Completed!



# CODING



Can you think of a suggestion to make the colorful light even smarter?  
(e.g. add automatic decision rather than always wait for command)

1 initialize OLED with width 128 height 64

2 Initialize IoT:bit TX P16 RX P8

3-4 Set WiFi to ssid "smarthon\_ss8" pwd "Qw64811051"

5 set strip to NeoPixel at pin P0 with 1 leds as RGB (GRB format)

6 show icon

7 show string Device\_ID

## STEPS:

1. Snap **Initialize OLED with width:128, height: 64** to **on start**
2. Snap **Initialize IoT:bit TX P16 RX P8** from IoT:bit to **on start**
3. Snap **Set Wi-Fi to ssid pwd** from IoT:bit
4. Enter your Wi-Fi name and password. Here we set smarthon as SSID and 12345678 as password
5. Snap **set strip to NeoPixel at pin P0 with 1 leds as RGB(GRB format)** from Neopixel
6. Snap **show icon** from basic to **On WiFi connected** and select icon tick
7. Draw the **Device ID** variable from **On WiFi connected** to the **show string** block placeholder

# CODING



Can you think of a suggestion to make the colorful light even smarter?  
(e.g. add automatic decision rather than always wait for command)

8-9

10

11

12

13

```
On WiFi received WAN_Command
if WAN_Command = Red then
  strip show color red
else if WAN_Command = Orange then
  strip show color orange
else if WAN_Command = Yellow then
  strip show color yellow
else if WAN_Command = Green then
  strip show color green
else if WAN_Command = Blue then
  strip show color blue
else if WAN_Command = Indigo then
  strip show color indigo
else if WAN_Command = Violet then
  strip show color violet
else if WAN_Command = Purple then
  strip show color purple
else if WAN_Command = White then
  strip show color white
else if WAN_Command = Off then
  strip show color black
```

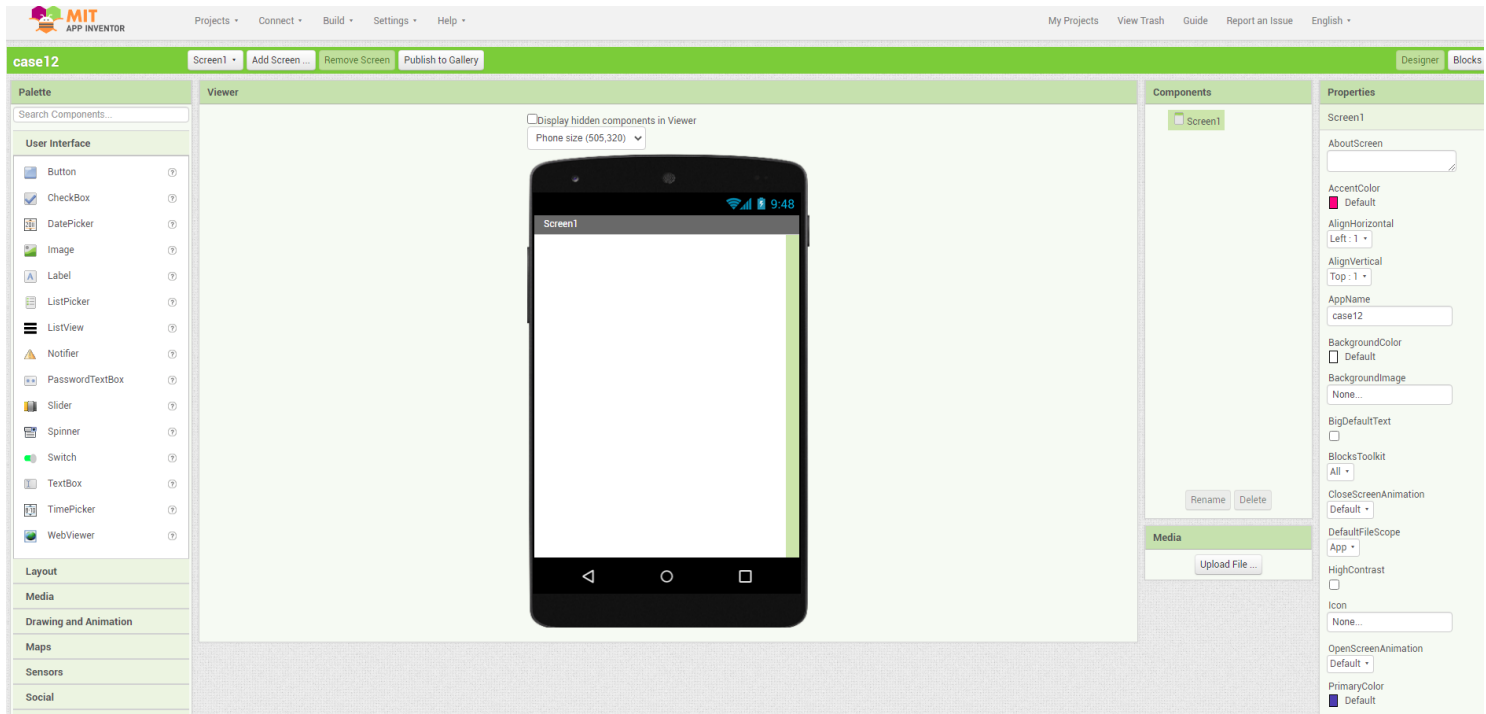
## STEPS:

8. Inside the **On WiFi Received**, put a nested **if-else** statement with different conditions
9. Set the first condition as **WAN\_Command = Red**
10. In the **if** segment, perform the change color action by **strip show color red**
11. In the next **if** condition, use **WAN\_Command = Orange**
12. Set the color to Orange by **strip show color orange**
13. Repeat the previous steps, with changes on the Color, to finish the setup of responding to each color.

# IOT – App inventor 2

## Step 1. Create an APP project

1. Create a APP inventor account at <http://ai2.appinventor.mit.edu/>
2. Create a new project

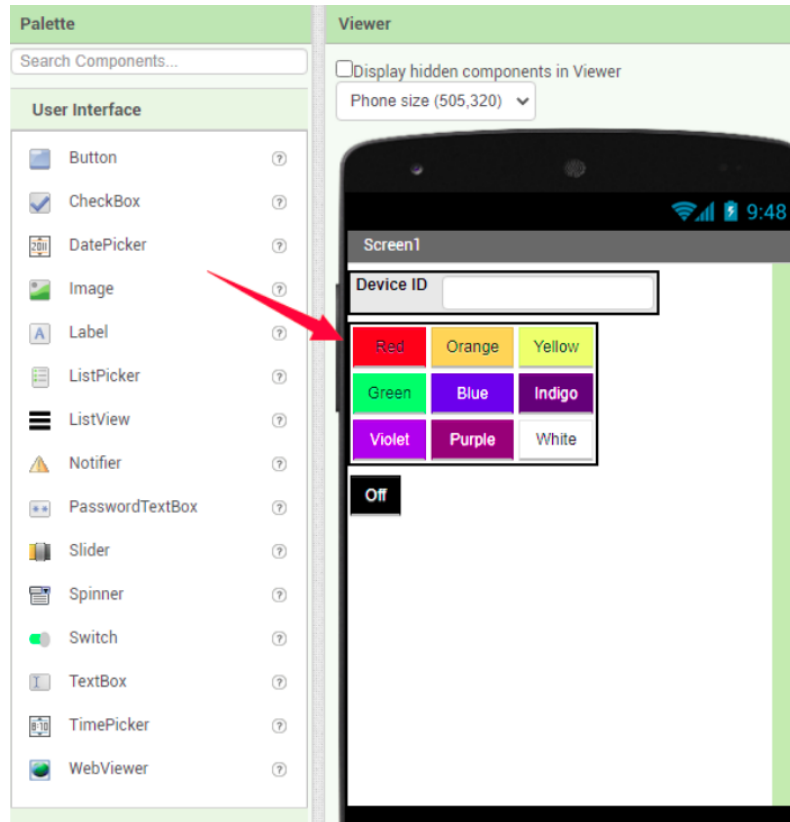


# IOT – App inventor 2

---

Step 2. Design the layout of APP

1. In the designer page, pull the layout element from the left side to the editor



# IOT – App inventor 2

2. In this example case, using different button, textbox, label and layout control
3. Place the element in your way
4. For each element has their own property, you may change it in your mind, such as the background color, font size, width, height, align.
5. In this example, you are required to change the Text to the Color that button representing to get the same result as documentation. For example, button who send “Red” command need to be “Red” as Textproperty
6. Remember to put Web element to editor, it will used for sending command



# IOT – App inventor 2

---

Step 3. Programming the elements in APP

1. Switch to Blocks page by click the button at top right corner





# IOT – App inventor 2

After finish the function for button1, repeat the steps for other buttons, with changes on target button element ( from button1 to button2, button3, ... )

```
when Button1 . Click
do
  set Web1 . Url to join * https://control.smarthon.cc/publish?id= *
  join TextBox1 . Text
  join * &msg= *
  Button1 . Text
call Web1 . Get
```

```
when Button2 . Click
do
  set Web1 . Url to join * https://control.smarthon.cc/publish?id= *
  join TextBox1 . Text
  join * &msg= *
  Button2 . Text
call Web1 . Get
```

```
when Button3 . Click
do
  set Web1 . Url to join * https://control.smarthon.cc/publish?id= *
  join TextBox1 . Text
  join * &msg= *
  Button3 . Text
call Web1 . Get
```

```
when Button4 . Click
do
  set Web1 . Url to join * https://control.smarthon.cc/publish?id= *
  join TextBox1 . Text
  join * &msg= *
  Button4 . Text
call Web1 . Get
```

```
when Button5 . Click
do
  set Web1 . Url to join * https://control.smarthon.cc/publish?id= *
  join TextBox1 . Text
  join * &msg= *
  Button5 . Text
call Web1 . Get
```

```
when Button6 . Click
do
  set Web1 . Url to join * https://control.smarthon.cc/publish?id= *
  join TextBox1 . Text
  join * &msg= *
  Button6 . Text
call Web1 . Get
```

```
when Button7 . Click
do
  set Web1 . Url to join * https://control.smarthon.cc/publish?id= *
  join TextBox1 . Text
  join * &msg= *
  Button7 . Text
call Web1 . Get
```

```
when Button8 . Click
do
  set Web1 . Url to join * https://control.smarthon.cc/publish?id= *
  join TextBox1 . Text
  join * &msg= *
  Button8 . Text
call Web1 . Get
```

```
when Button9 . Click
do
  set Web1 . Url to join * https://control.smarthon.cc/publish?id= *
  join TextBox1 . Text
  join * &msg= *
  Button9 . Text
call Web1 . Get
```

```
when Button10 . Click
do
  set Web1 . Url to join * https://control.smarthon.cc/publish?id= *
  join TextBox1 . Text
  join * &msg= *
  Button10 . Text
call Web1 . Get
```



# 13. [IOT] SMART FAN VOICE CONTROL

LEVEL: ★ ★ ★ ★ ★

## Introduction

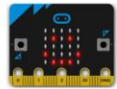
It is a fan that is connected to the internet and can be woken up by voice command using a voice assistant. The user can define different voice commands to control fan in different speeds or modes. In this case, the user can turn on and off from the voice command.



IFTTT



## PART LIST



micro:bit (1)



IoT:bit with OLED (1)



Motor Fan(1)



Screwdriver (1)



3-pin module wire (1)



M4\*10mm Screw (2)



M4\*10mm Nut (2)



Model A - C (1)  
[Big House Model]



Model E1 (1)  
[Long Sensor Wall]



Model K1 (1)  
[Sofa]



Model K2 (1)  
[Sofa]



Model K3 (2)  
[Sofa]



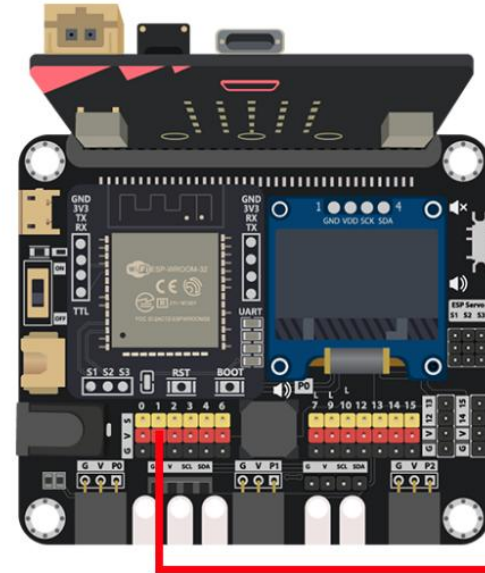
Model F (1)  
[Motor Stand]



Model H (1)  
[Decoration Display]

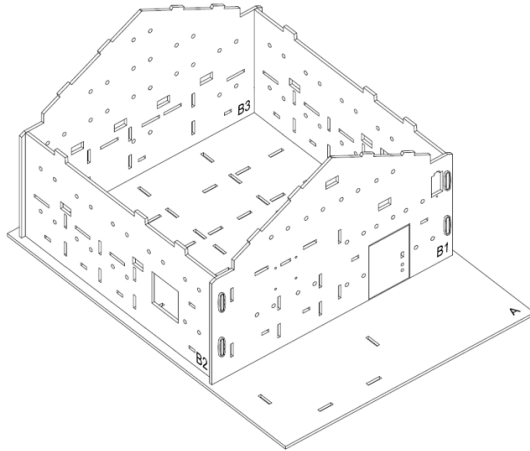


Amazon Alexa Device or  
Google Nest Mini or  
Apple Homepod mini (1)  
[Optional]

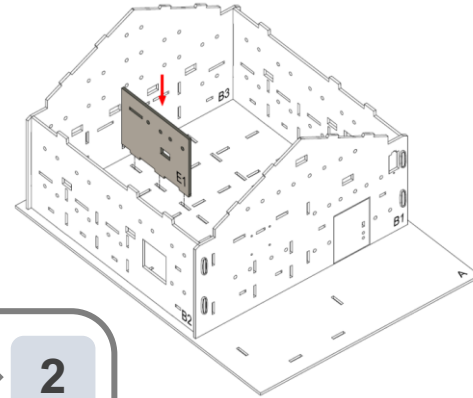


# ASSEMBLY STRUCTIONS

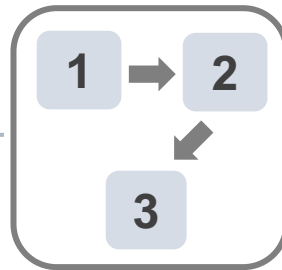
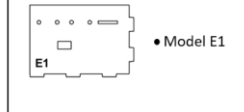
In this case, build the “Big Style Model” as a home base.



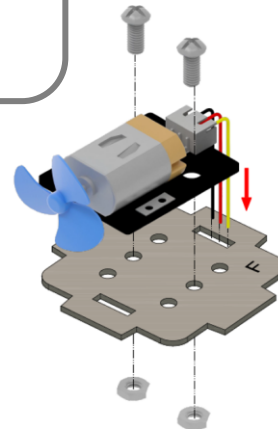
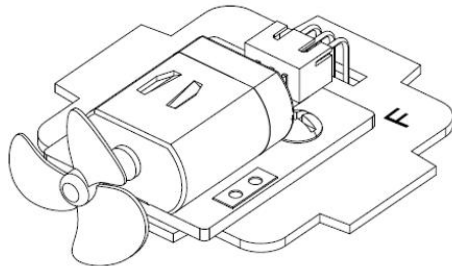
Insert the model E1 on model A, align with the holes on model A and B3.







Part list:



Attach the motor fan into the model F using M4 \* 10mm screws and nuts. And the connecting wire could be bended to the hole next to it.

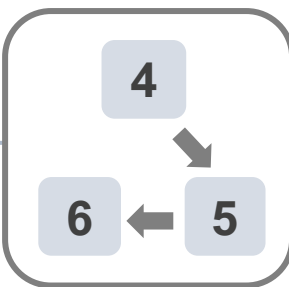
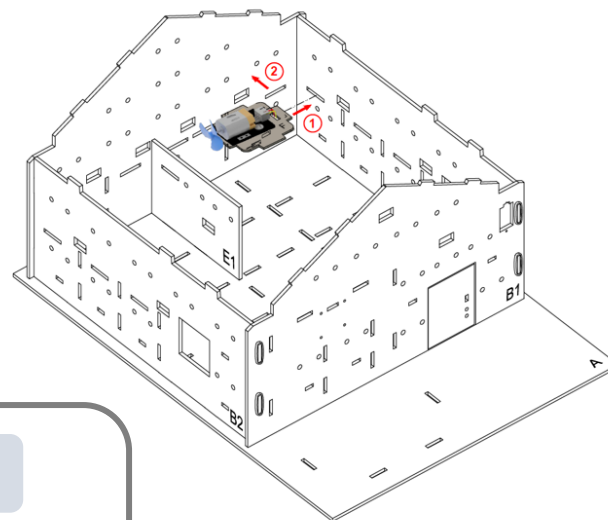
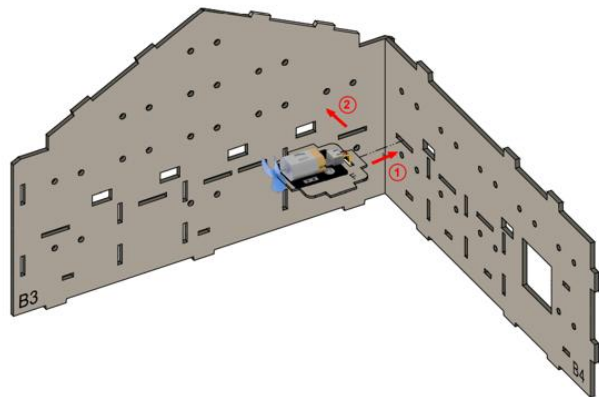


Part list:

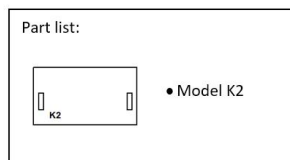
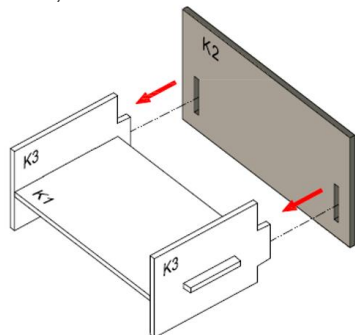
-  • Motor Fan
-  • M4 nut X2
-  • M4\*10mm screw X2
-  • Model F

# ASSEMBLY STRUCTIONS

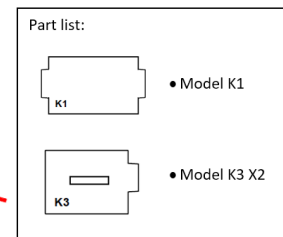
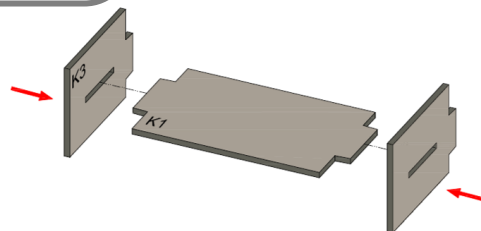
To build a fan stand, put model F into the model B3 and B4.



Put model K2 all together to the cardboard parts (K1-K3).

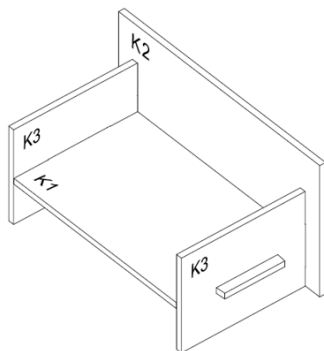


To build a sofa model. Put the model K3 to the two sides of model K1.

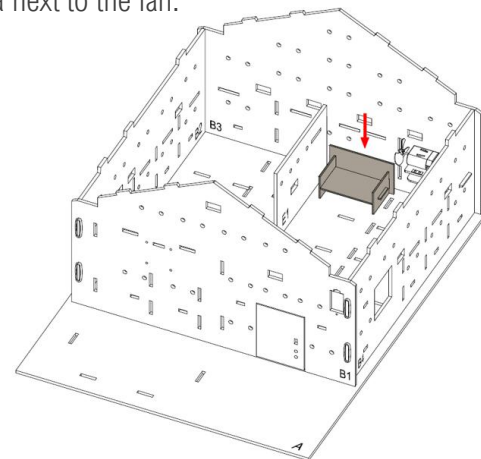


# ASSEMBLY INSTRUCTIONS

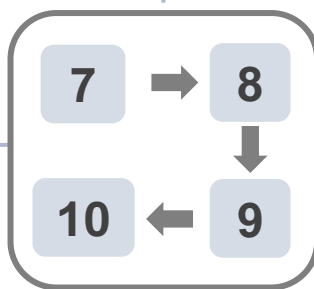
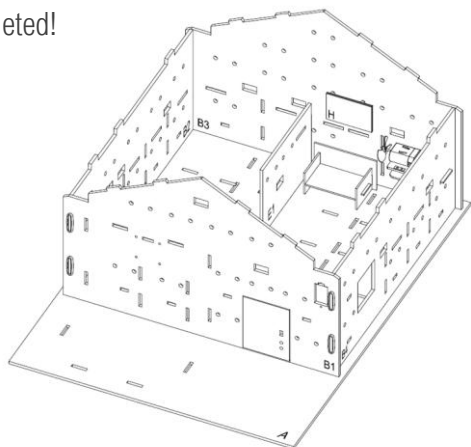
The sofa completed~



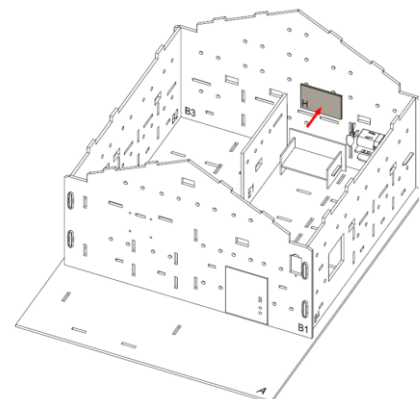
Place the sofa next to the fan.



Completed!

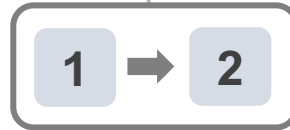
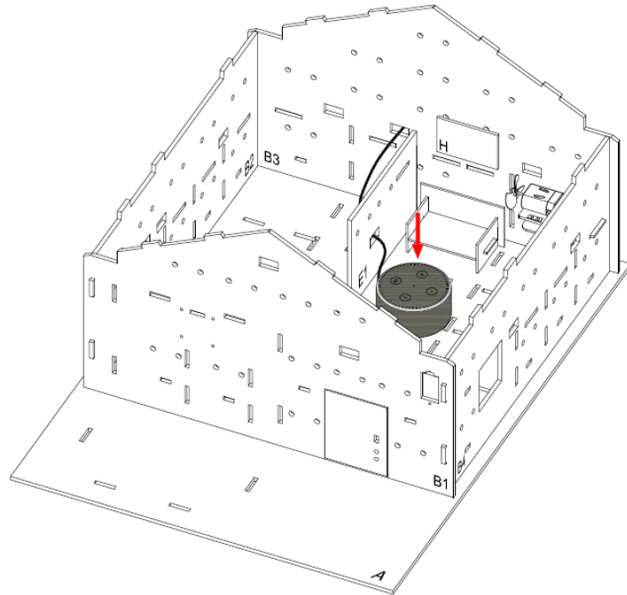


Place the decoration board (Model H) on model B3.

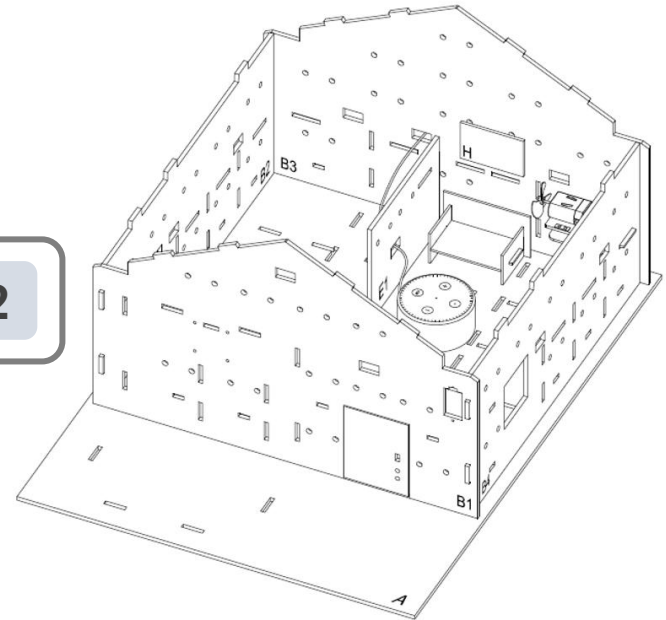


# ASSEMBLY STRUCTIONS (OPTIONAL)

Place the Amazon Alexa/Google Nest Mini/Apple Homepod mini at the living room. In this case, we put Amazon Alexa here.



Assembly Completed!



# CODING



Apart from “Say a specific phrase” Alexa IFTTT trigger, any other Alexa trigger can use in case?

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

```
On WiFi connected IP_Address Device_ID
  show string Device_ID
  show icon
  Device ID
```

## STEPS:

1. Snap **Initialize OLED with width:128, height: 64** to **on start**
2. Snap **Initialize IoT:bit TX P16 RX P8** from IoT:bit to **on start**
3. Snap **Set Wi-Fi to ssid pwd** from IoT:bit
4. Enter your Wi-Fi name and password. Here we set smarthon as SSID and 12345678 as password
5. Snap **show icon** from basic to **On WiFi connected** and select icon tick
6. Draw the **Device ID** variable from **On WiFi connected** to the **show string** block placeholder

# CODING



Apart from “Say a specific phrase” Alexa IFTTT trigger, any other Alexa trigger can use in case?

```
on WiFi received wanCommand
8 clear OLED display
9 show string join "Command: " wanCommand
10-11 if wanCommand = "turn_off_fan" then
12   Set Motor fan with speed 0 at P1
13 else if wanCommand = "turn_on_fan" then
14   Set Motor fan with speed 1023 at P1
```

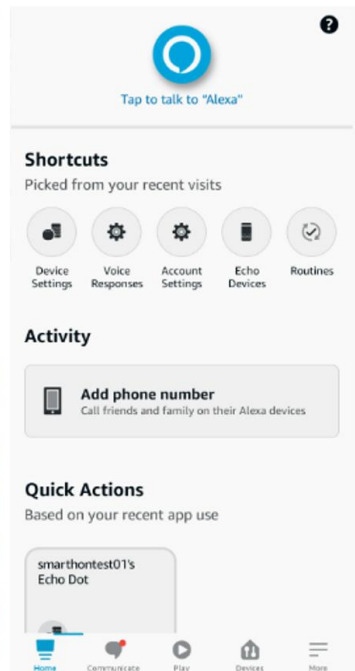
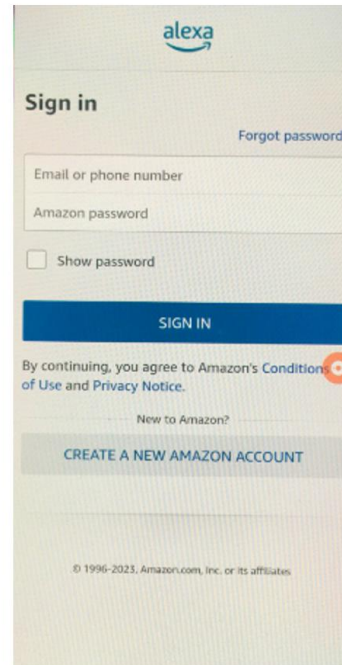
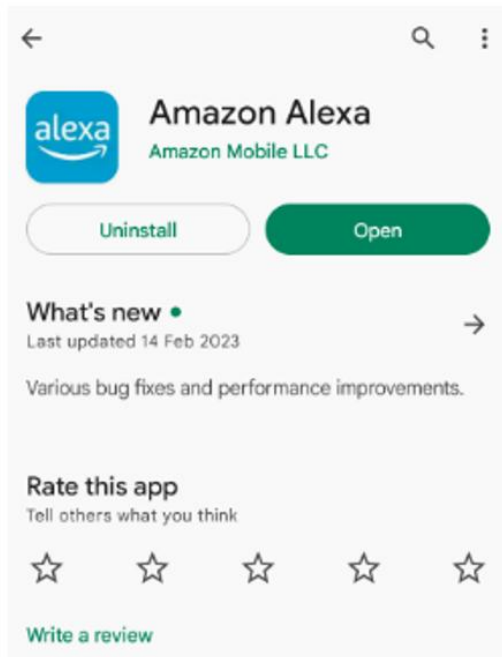
## STEPS:

7. Inside the **On WiFi Received**, show the command on OLED display
8. Clear the display before each update by **Clear OLED display**
9. Show the **WAN\_Command** with text explanation by **show string join Command: WAN\_Command**
10. put a nested **if-else** statement with different conditions
11. Set the first condition as **WAN\_Command = turn\_off\_fan**
12. In the **if** segment, turn off the fan by **set Motor fan with speed 0 at P1**
13. In the second **if** condition, use **WAN\_Command = turn\_on\_fan**
14. In the second **if** segment, turn on the fan by **set Motor fan with speed 1023 at P1**

# IOT – Alexa, Cloud Control, IFTTT

## Part 1. Setup Amazon Alexa

1. Get the amazon alexa apps.
2. Create Amazon Account (if not have amazon account) and login.
3. Optional: Setup the alexa device and connect to the amazon account.

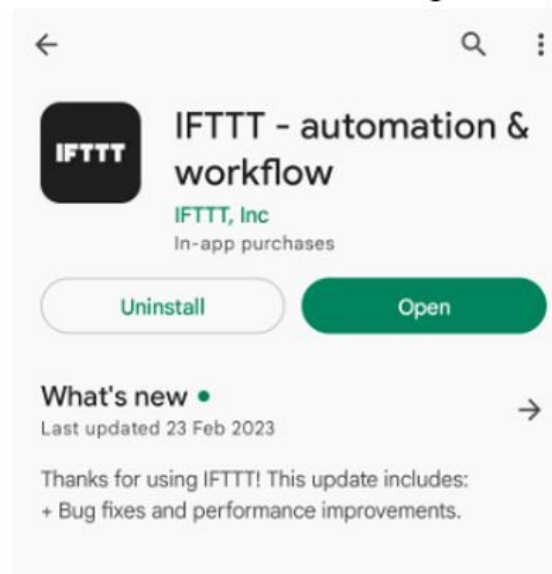


# IOT – Alexa, Cloud Control, IFTTT

---

## Part 2. Setup IFTTT

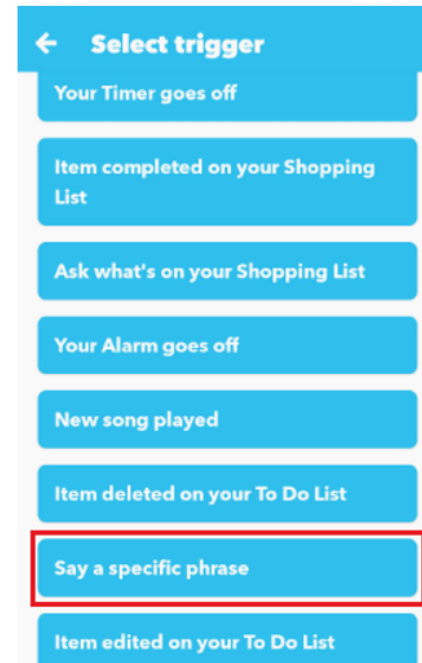
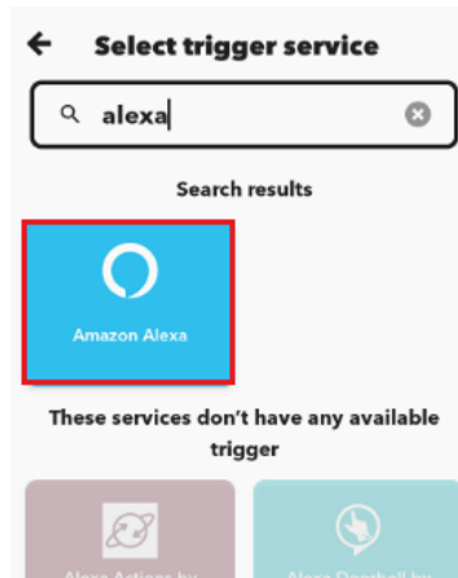
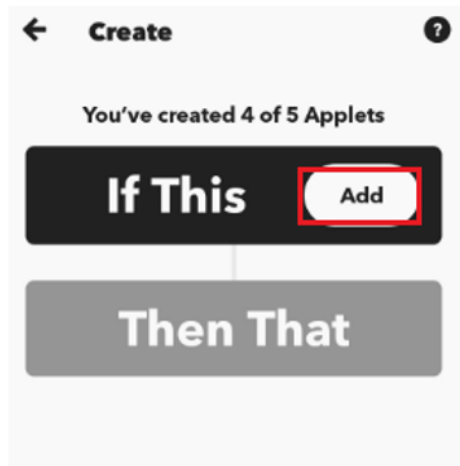
1. Install IFTTT from Google Play Store.



# IOT – Alexa, Cloud Control, IFTTT

---

2. Open and login to IFTTT.
3. Press create to create the applet.
4. At the IF, search “alexa “ and choose “Amazon Alexa” and choose “Say a specific phrase.”




# IOT – Alexa, Cloud Control, IFTTT

---

5. It will detect the login account from Amazon Alexa app. Then, type the phrase here. For this example, we type “turn on the fan”.

← Complete trigger fields



**Say a specific phrase**

This trigger fires every time you say "Alexa trigger" + the phrase that you have defined. For instance, if you set "party time" as the phrase, you can say "Alexa trigger party time" to have your lights loop colors.

Amazon Alexa account

smarthontest01

Pro Add more accounts

What phrase?

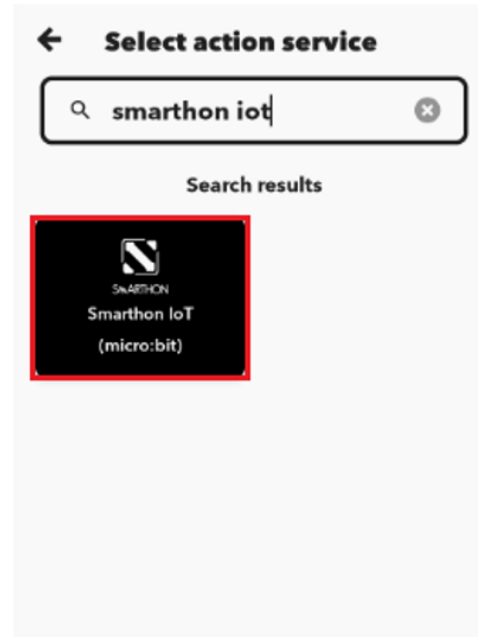
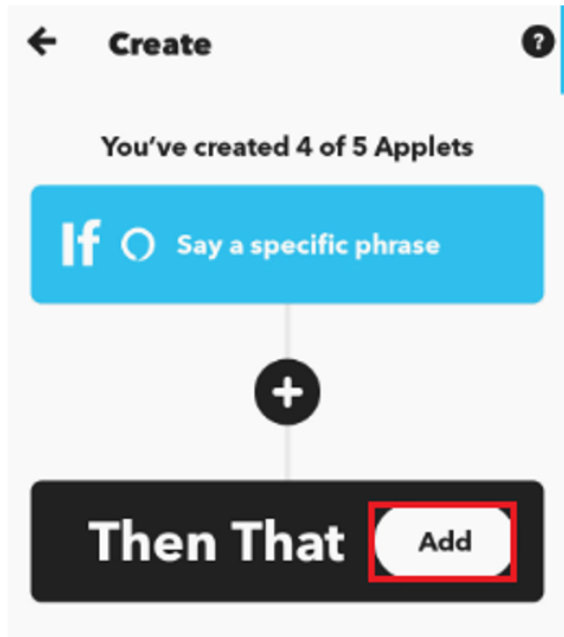
turn on the fan

Continue

# IOT – Alexa, Cloud Control, IFTTT

---

6. Click Add and type “smarthon iot” and choose Smarthon IoT as “then” service.



# IOT – Alexa, Cloud Control, IFTTT

---

7. In the service, choose “Control Command” and then fill in the Device ID, the command name that can read from the Micro:bit, and click “continue”, DONE!

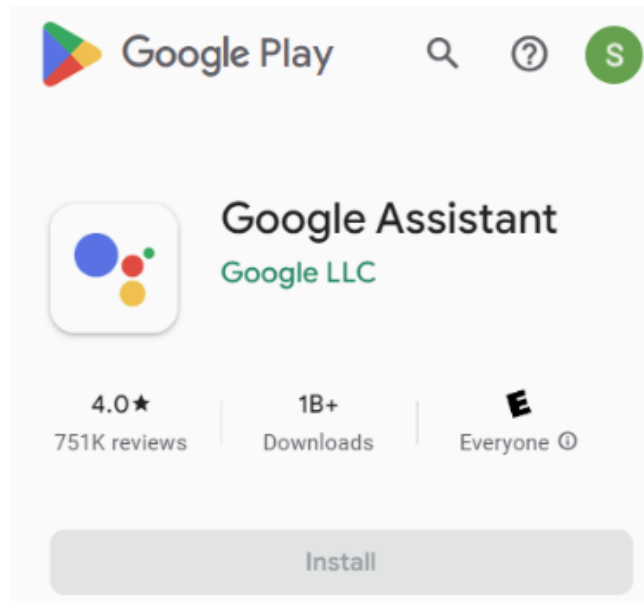
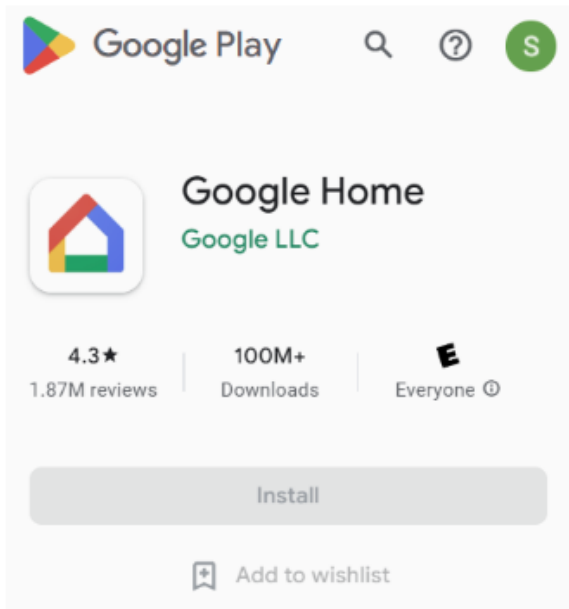


# IOT – Google Assistant

---

Part 1. Setup Google Home and Google Assistant App

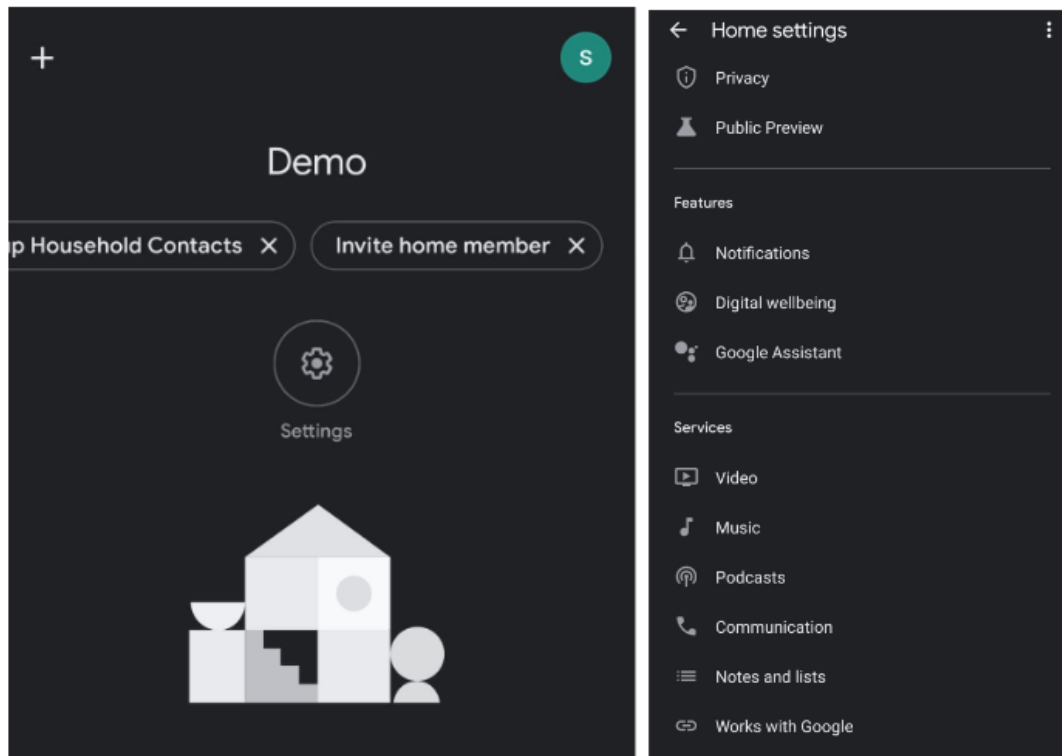
1. Install Google Home and Google Assistant app.



# IOT – Google Assistant

---

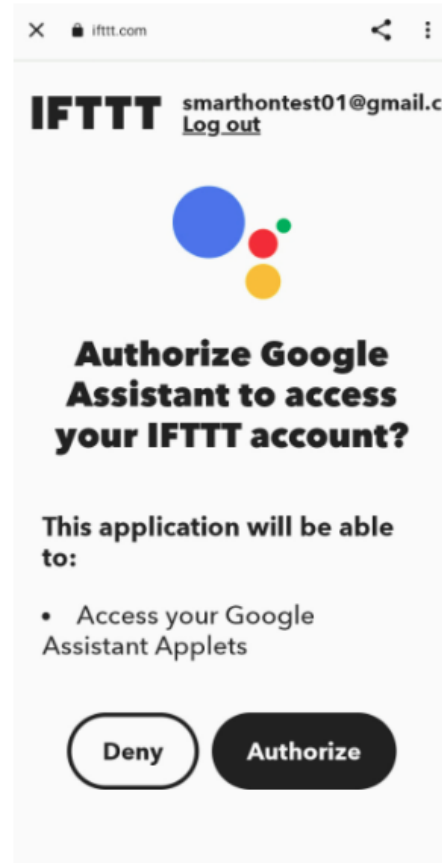
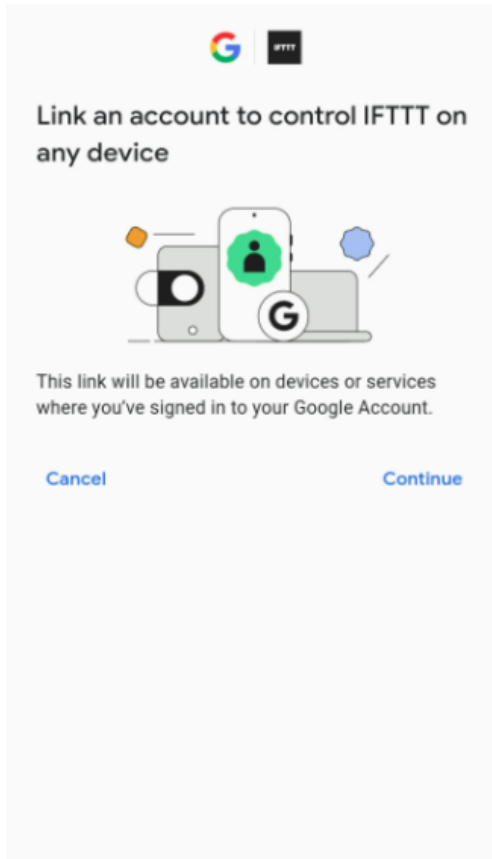
2. Login with your google account.
3. Open the google home app, switch to your google account.
4. Press the settings button, choose “Works with Google”.



# IOT – Google Assistant

---

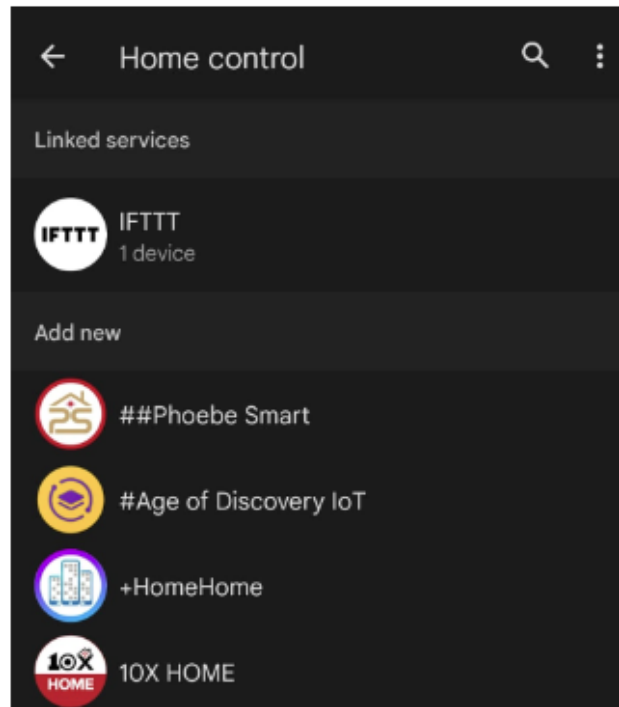
5. Search and select IFTTT, provide authorize of Google to access the IFTTT service.



# IOT – Google Assistant

---

6. After linked, the IFTTT will show on the list.

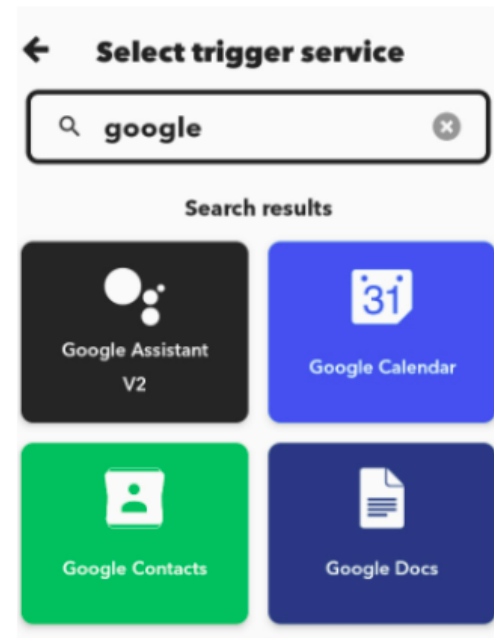
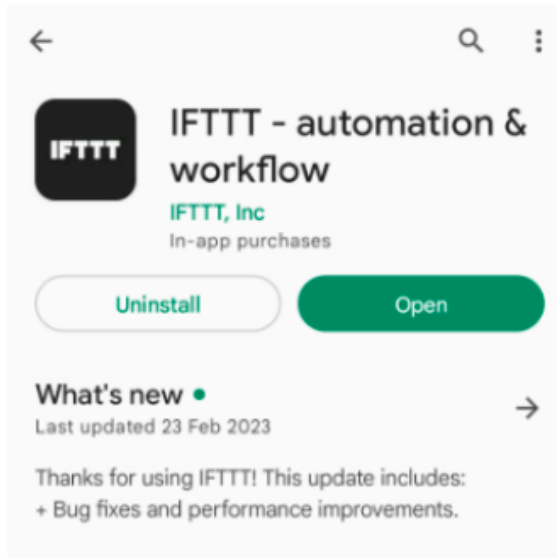


# IOT – Google Assistant

---

## Part 2. Setup IFTTT

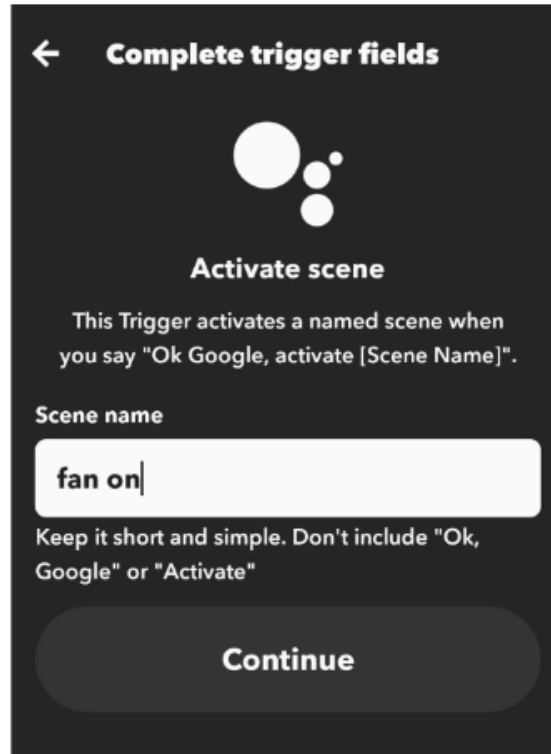
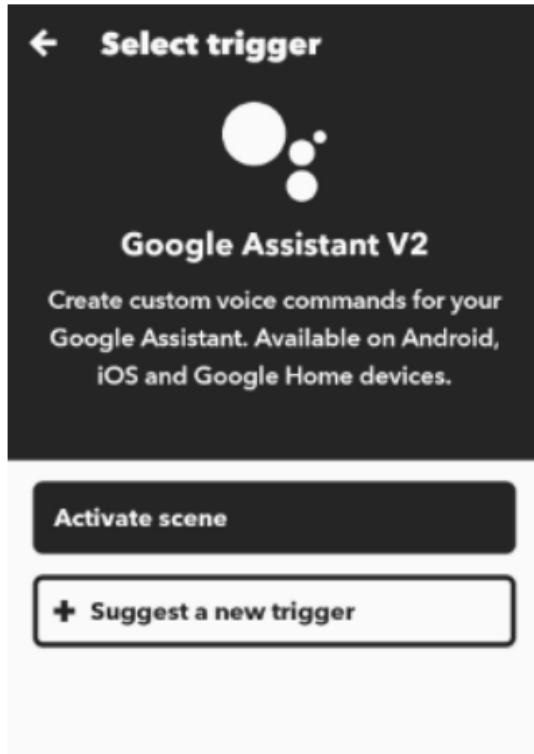
1. Install IFTTT from Google Play Store.
2. Open and login to IFTTT.
3. Press create to create the applet.
4. At the IF, search “google assistant “ and choose “Google assistant V2”.



# IOT – Google Assistant

---

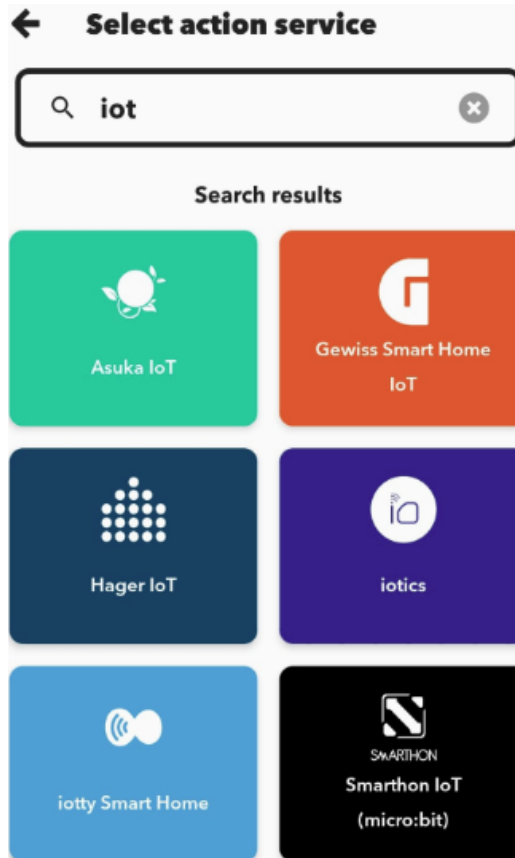
5. In “Activate scene” input the phrase for trigger the command. In this case, we input “fan on”.



# IOT – Google Assistant

---

- At the THEN, search “iot” and choose “Smarthon IoT (micro:bit)”.



# IOT – Google Assistant

---

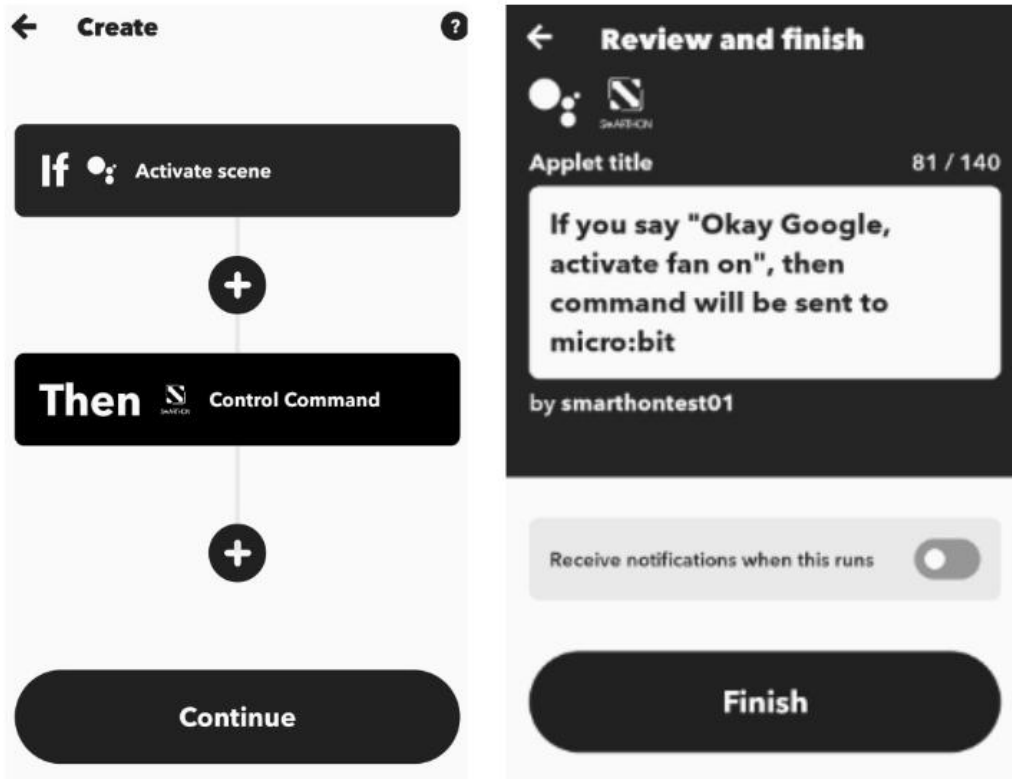
7. Press the “Control Command”, input the device ID of lot:bit and the command “turn\_on\_fan”.



# IOT – Google Assistant

---

8. Then click continue and finish button to save the applet.

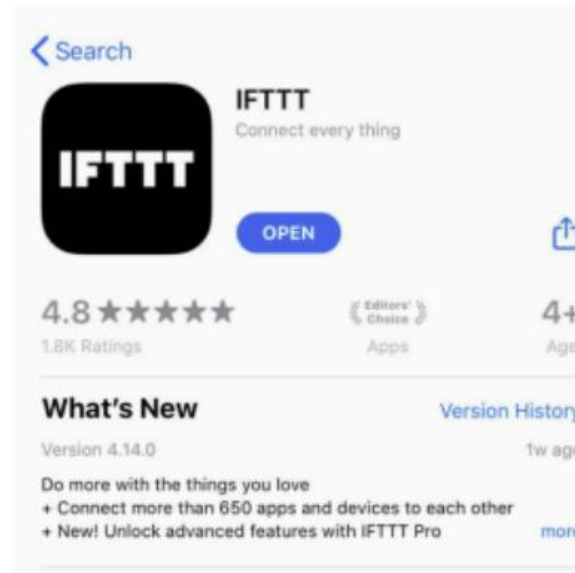


# IOT – Apple’s Siri

---

## Part 1. Setup IFTTT

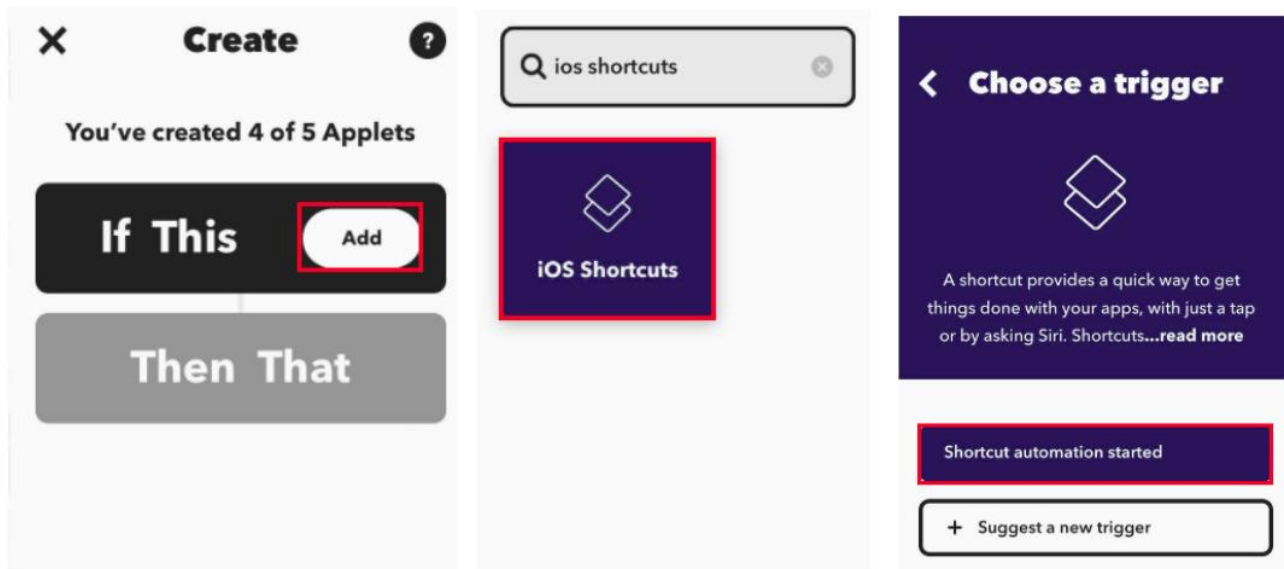
1. Install “IFTTT” App from Apple store.



# IOT – Apple’s Siri

---

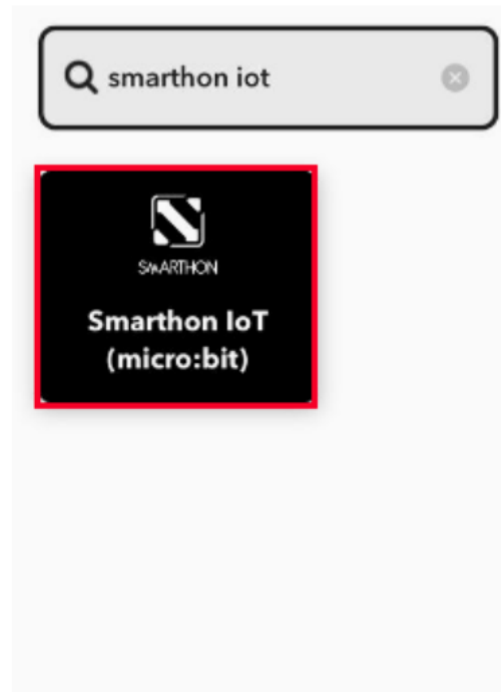
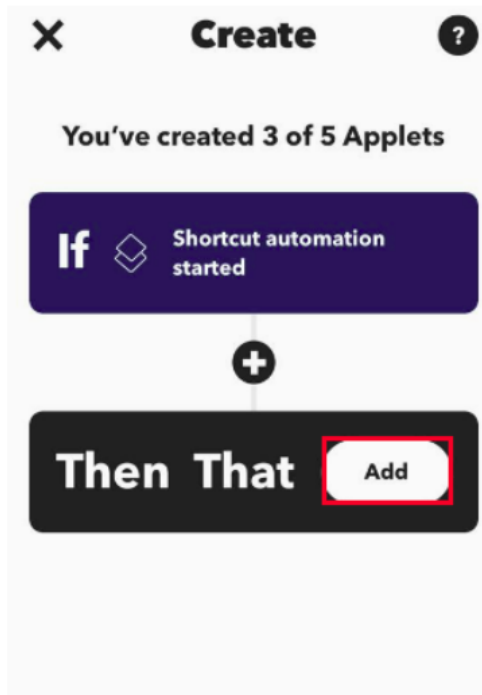
2. Open the APP, login into IFTTT.
3. Tap “Create” to create an applet.
4. Click If This “Add” button. Select the “iOS Shortcuts” and click “Shortcut automation started”.



# IOT – Apple’s Siri

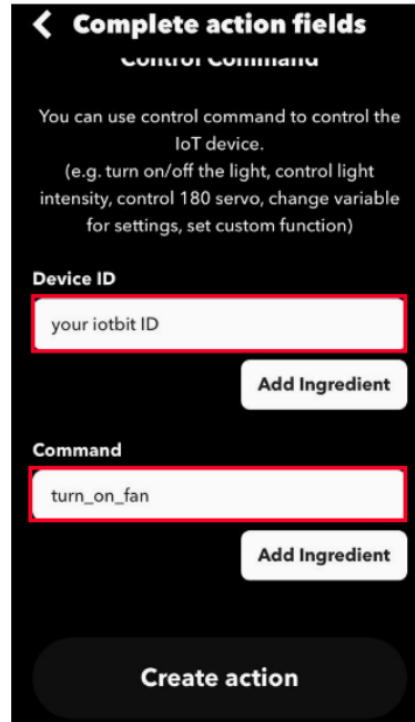
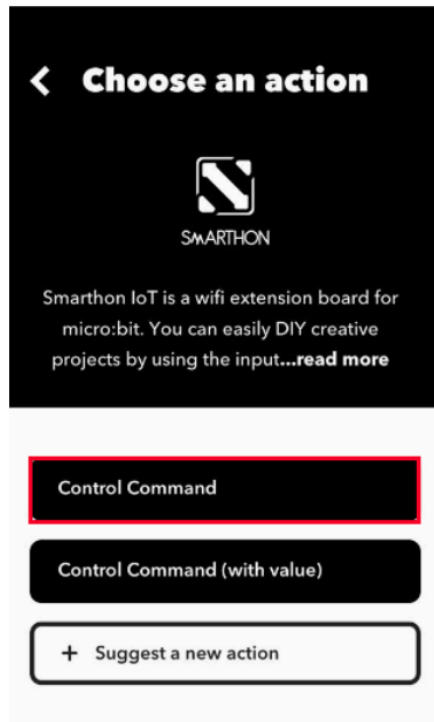
---

5. Click Then That “Add” button, search “Smarthon IoT” and select the “Smarthon IoT (micro:bit)”.



# IOT – Apple’s Siri

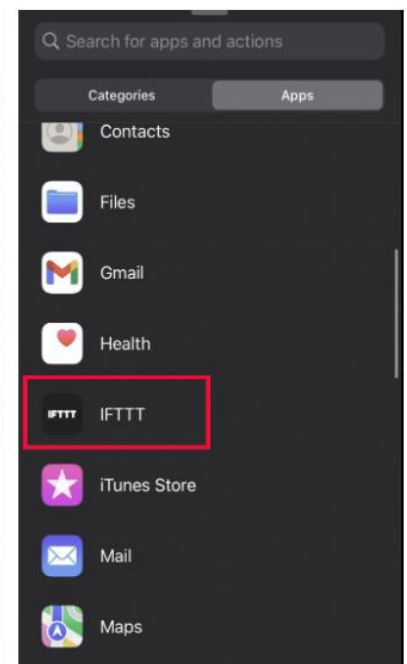
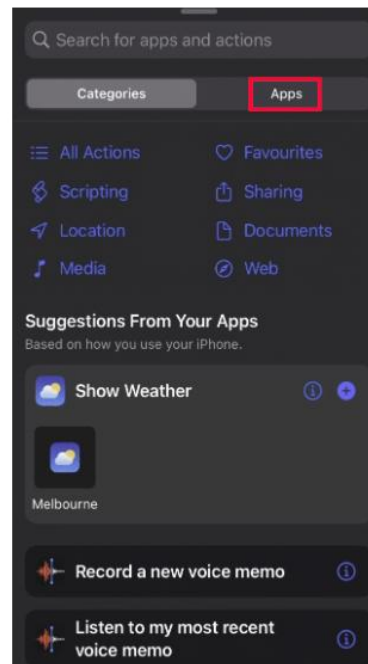
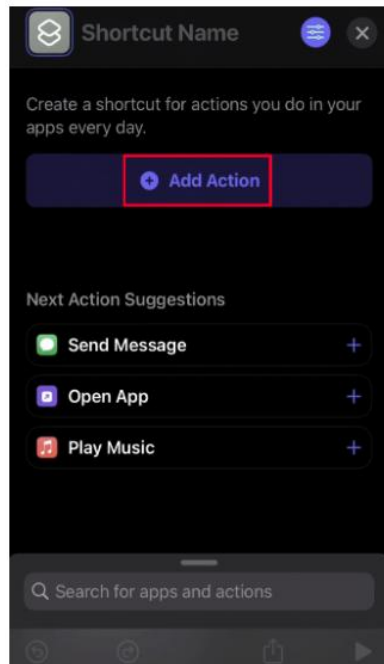
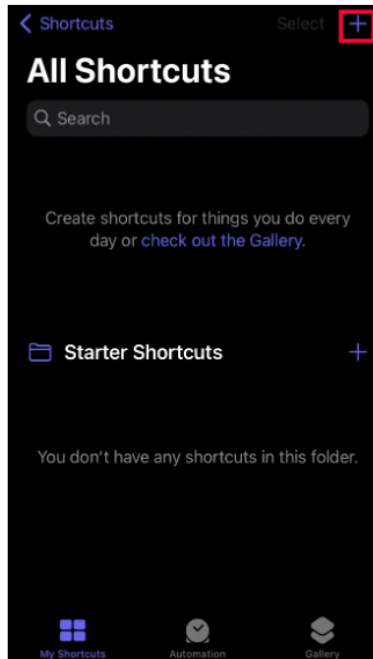
6. Choose the “Control Command”, and fill in the Device ID, and the command name that can read from the Micro:bit, and click “Continue”, then save the applet.



# IOT – Apple's Siri

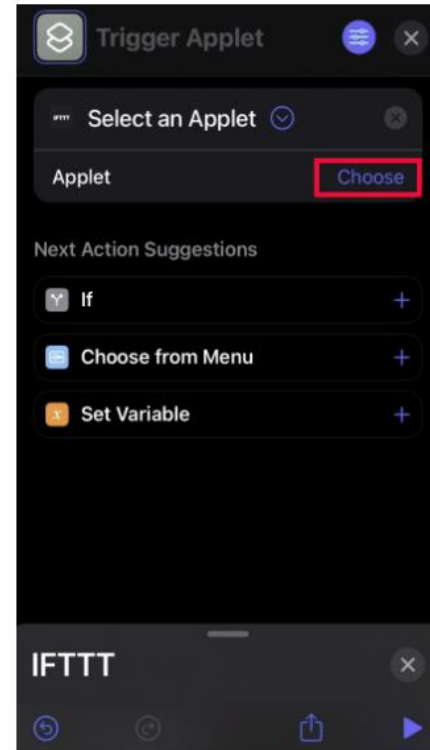
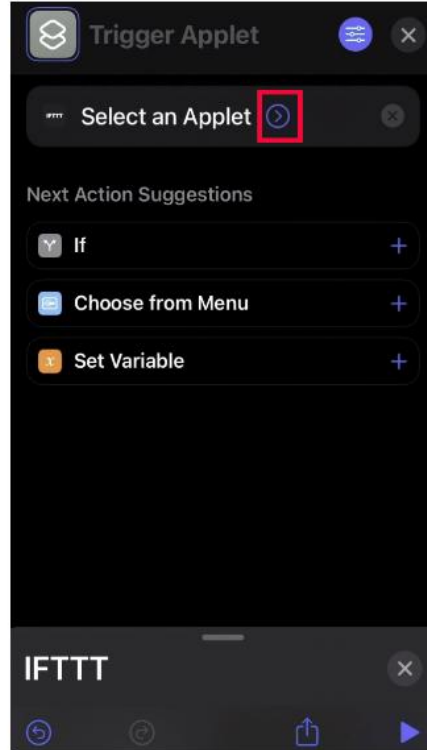
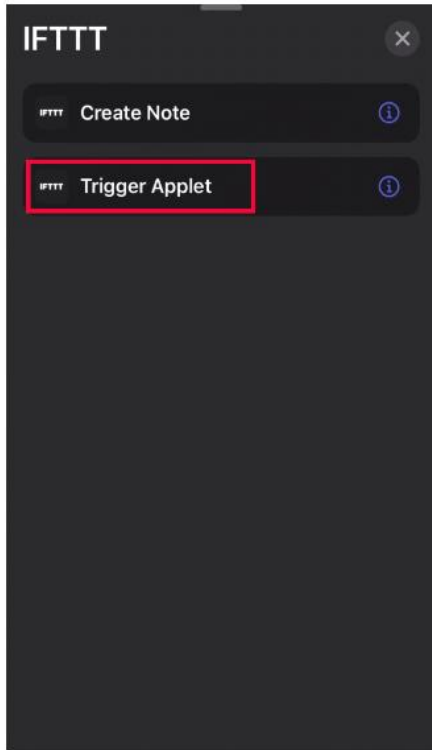
## Part 2. Setup IOS ShortCut on Iphone

1. Open the Shortcuts app and tap the plus button ( + ) in the top right corner.
2. Tap Add Action > Apps > IFTTT.



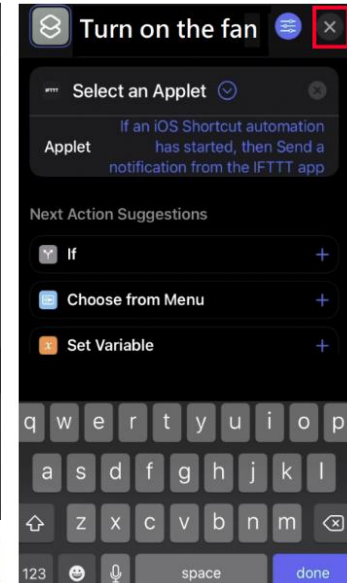
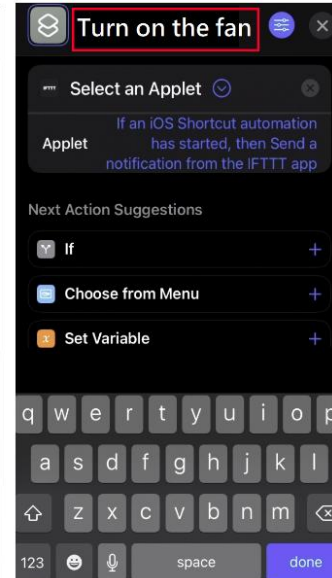
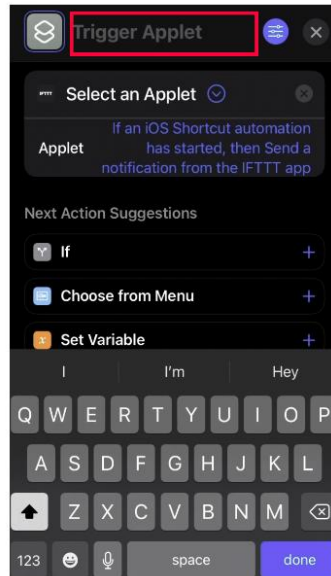
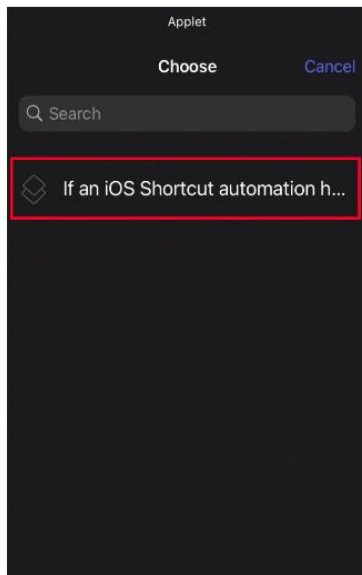
# IOT – Apple’s Siri

3. Select Trigger Applet, Tap the blue arrow ( > ) next to ‘Select an Applet’, then tap Choose.



# IOT – Apple's Siri

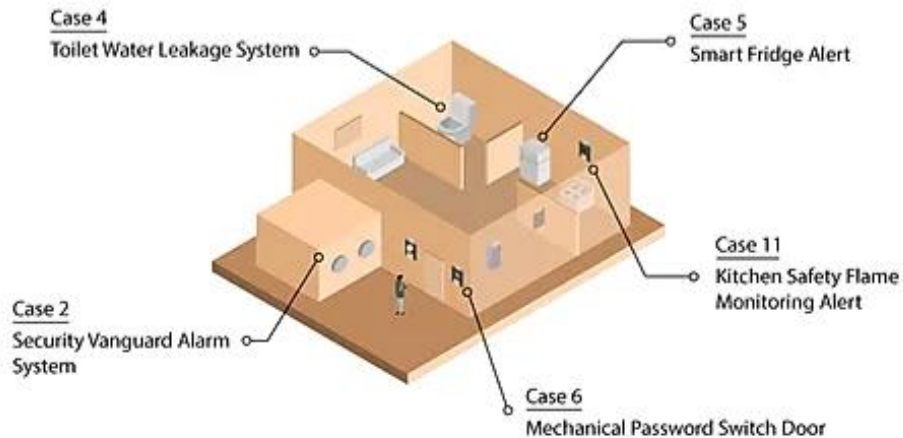
4. Select the Applet you'd like this Shortcut to run (please note that only Applets that use the iOS Shortcuts - Shortcut automation started trigger will appear in this list).
5. Name your Shortcut by tapping Trigger Applet at the top of the screen. This will be the custom phrase you will use to trigger your Applet.
6. Save the shortcut.



# SCENARIO EXAMPLE 1: HOUSING SAFETY AND SECURITY

---

Safety and Security is an important part of the smart home concept. With help from each sensor installed at home, it can reduce the chance of getting property damage, and even save a life.



## PARTS:

- Motion sensor
- Distance sensor
- Crocodile clip cables
- Light sensor
- Flame sensor
- Multi-color LED

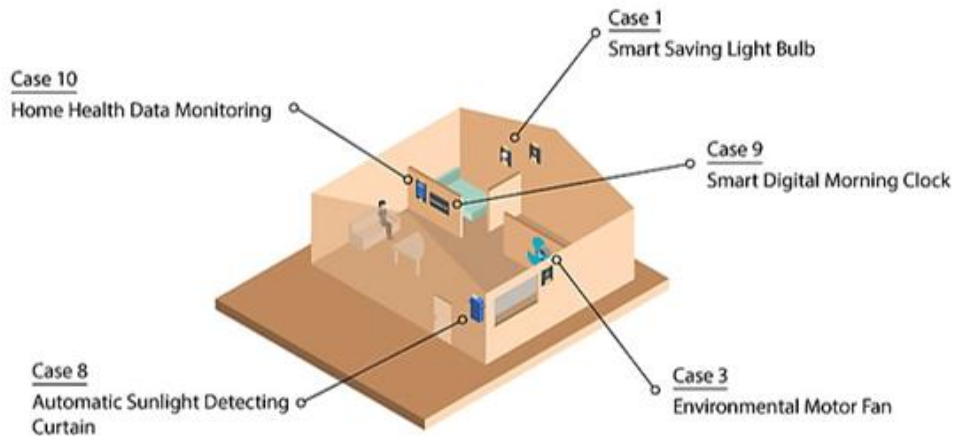
## IOT:

- IFTTT

# SCENARIO EXAMPLE 2: ENVIRONMENTAL HEALTH LIVING

---

Sustainability is a common topic of smart home, Environmental Health Living is to keep Personal Wellness and reduce the waste of energy at the same time.



## PARTS:

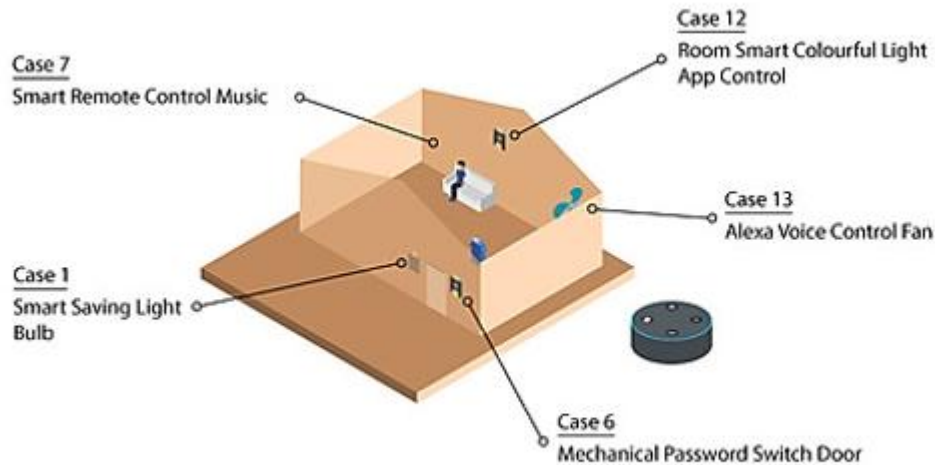
- Motion sensor
- Multi-color LED (WS2812)
- Temperature and Humidity sensor (DHT11)
- Motor fan module
- Light sensor
- 360° servo
- OLED

## IOT:

- ThingSpeak

# SCENARIO EXAMPLE 3: RELAXING HOME AUTOMATION

Relaxing Home Automation can bring convenience to the user, it reduces the redundant task in our daily life.



## PARTS:

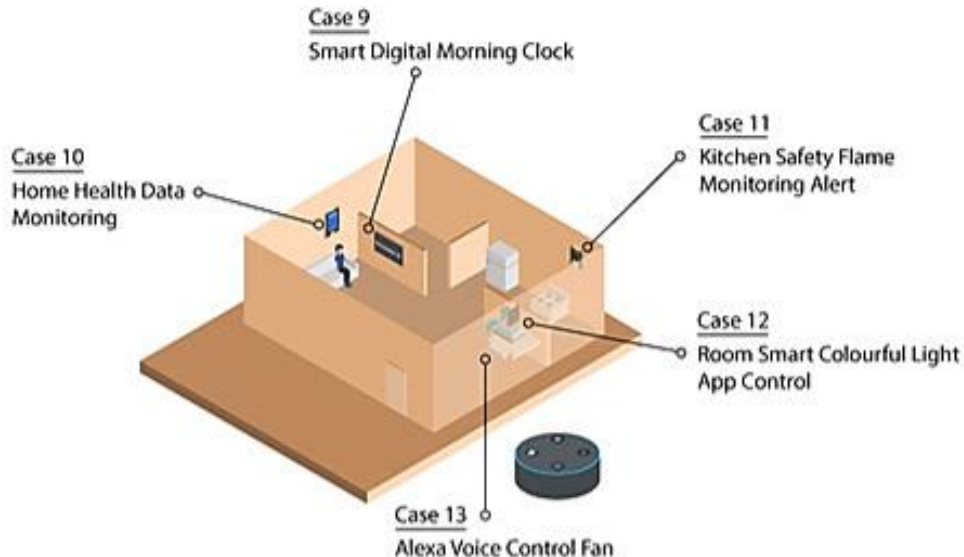
- Motion sensor
- Multi-color LED (WS2812)
- Button module
- 180° servo
- OLED
- Motor fan

## IOT:

- App Inventor
- Alexa
- IFTTT
- Google Assistant
- Siri

# SCENARIO EXAMPLE 4: SMART INTELLIGENT IOT HOME

Smart Intelligent IOT Home make use of the numerous internet resources is the advantage of IoT device, with that smart home can be monitoring and controlling by the internet.



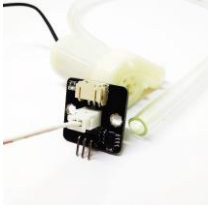


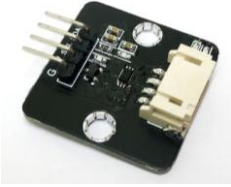

## PARTS:

- OLED
- Temperature and Humidity sensor (DHT11)
- Multi-color LED (WS2812)
- Flame sensor
- Motor fan

## IOT:

- ThingSpeak
- IFTTT
- App Inventor
- Alexa
- IFTTT
- Google Assistant
- Siri

# 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 <a href="http://www.smarthon.cc">www.smarthon.cc</a> .	

**FOR MORE INFORMATION  
PLEASE VISIT**

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



SMARTHON

[www.smarthon.cc](http://www.smarthon.cc)

**INGITE YOUR CREATIVITY**