

# Microbit Robotics Beginner Level 1

## Lesson 8

### Color recognition and Thermometer Modules

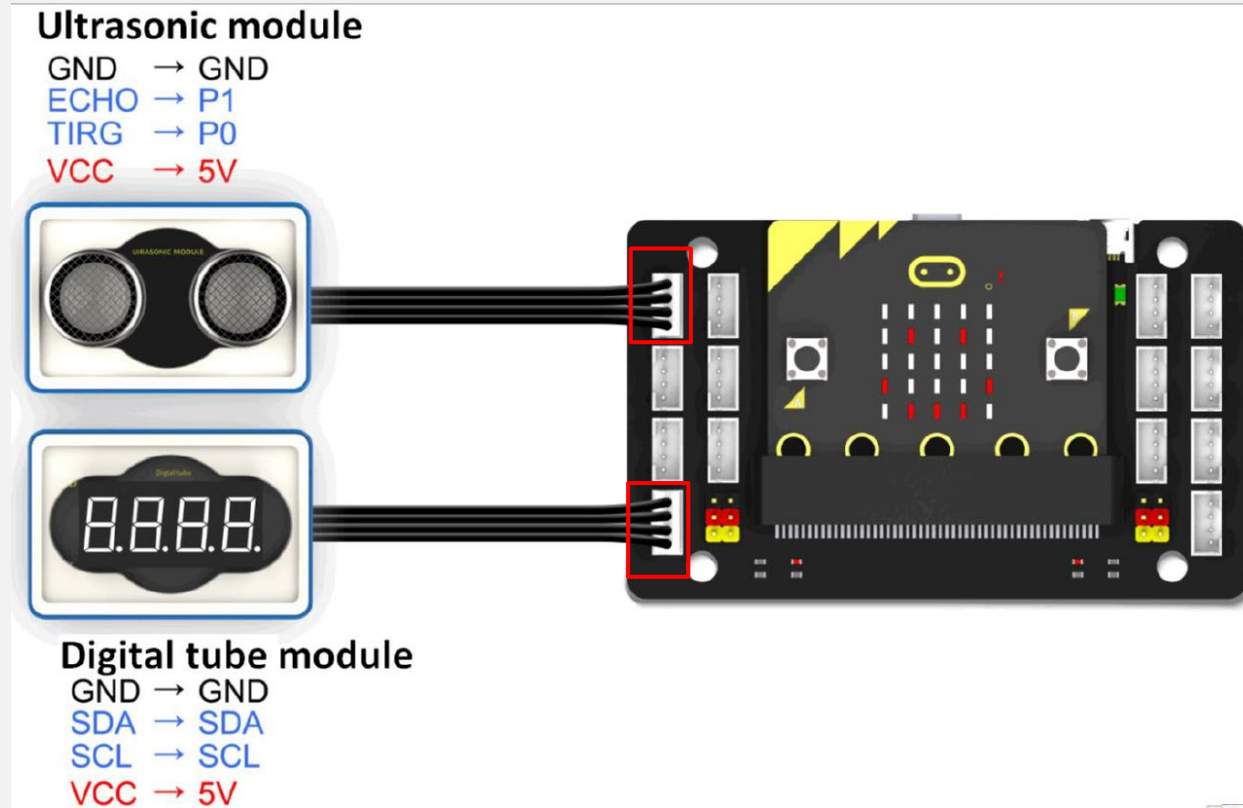
Presented by Advanced Superlogic Team

# Hand-held range finder



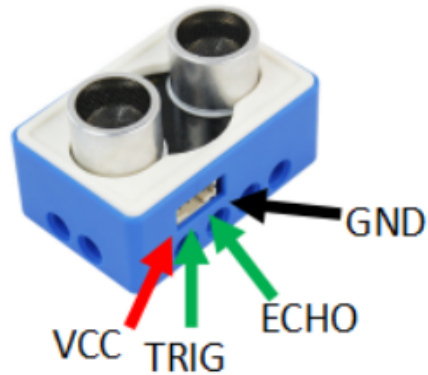
61

# Connect the modules



**Let's connect the module like this.**

# Introducing the Ultrasonic Module



GND: connect GND	VCC: connect 3.3V, 5V
TRIG: Transmit a signal	ECHO: Receive signal
Working Voltage: 3.3V/5V	Size of module: 44.7mm*28.8mm
Accuracy: 0.5cm	Range: 2cm~500cm

Ultrasonic sensor is designed to take advantage of the characteristics of ultrasonic waves. When working normally, the TRIG pin sends out ultrasonic signal. When encountering obstacles, the ultrasonic signal will be returned. After the ECHO pin receives the returned ultrasonic signal, the signal will be transmitted to the MCU. The MCU can calculate the time for ECHO to receive the signal to judge the current distance. The frequency of sound waves that can be heard by human ears is 20HZ ~ 20KHz, and the frequency of sound emitted by the ultrasonic module is greater than 20KHz, so the sound of the ultrasonic module cannot be heard by human ears.

# Introducing the Digital Tube Module



GND: connect <u>gnd</u>	VCC: Power supply interface, can be connected to 3.3V, 5V
SCL: IIC clock line	SDA: IIC data line
Working voltage: 3.3V/5V	Size: 44.7mm*28.8mm
Digital I2C base address: 0x34	Digital register I2C base address: 0x24

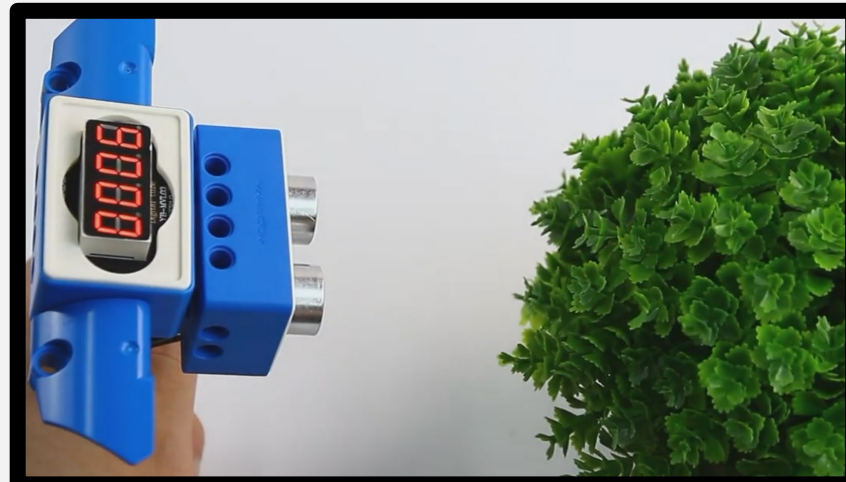
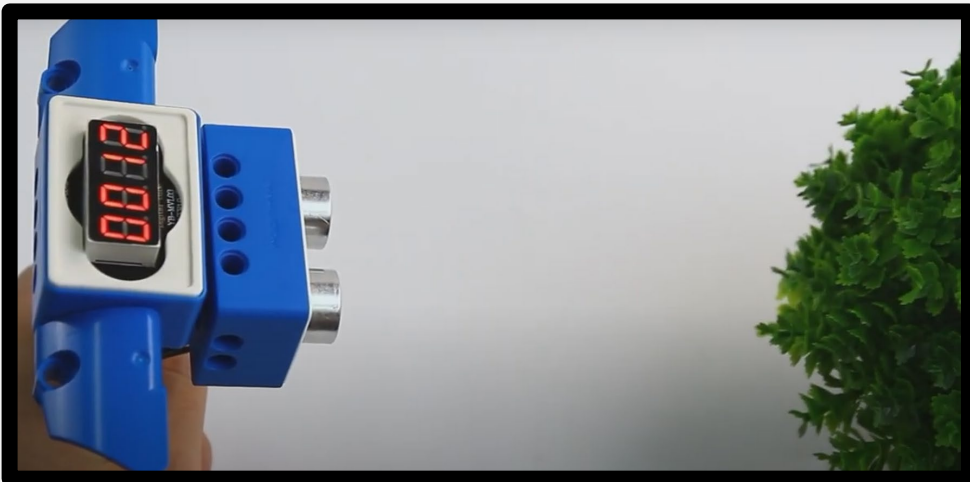
The module integrates the TM1650 chip, which can directly drive the digital tube display through I2C. It greatly reduces the difficulty of using the digital tube and saves the CPU occupancy rate of the single-chip microcomputer. The operation is convenient and flexible.

# Programming

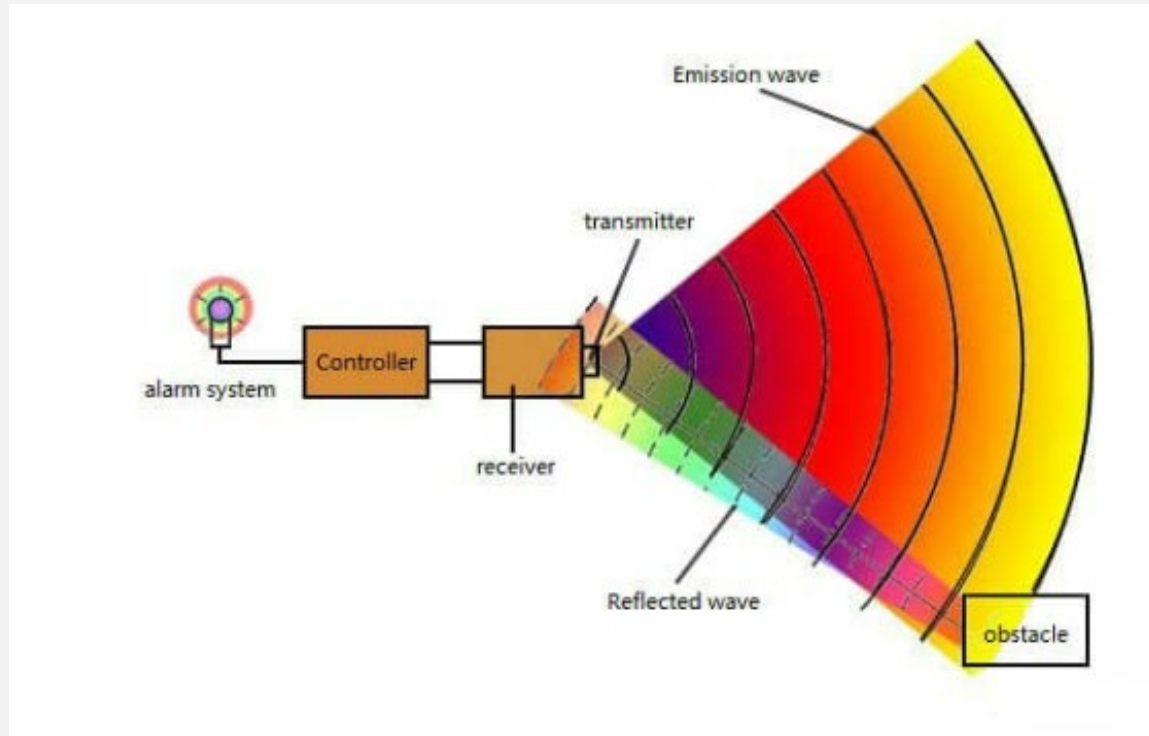


# Phenomenon

After the program is downloaded successfully.  
The digital tube will display the **distance**.



# Car Parking Sensor



How does a car parking sensor work?  
 Image result for car parking sensor  
 How does parking sensor work? The **Ultrasonic sensors** – just like bats – use **high-frequency sound waves** to detect objects. These sensors **emit sound pulses** that reflect off of nearby objects. A receiver detects the **reflected waves** and **calculates the distance** from your vehicle to the object.

## **Do a quick self-check of your learning outcome...**

- 1. Is Digital display module coding block in the Module-World extension?**
- 2. How many string or number can the digital tube module show at once?**
- 3. Is Ultrasonic sensor module giving digital or analog signal?**
- 4. Can you give 2 real life examples for the application of Ultrasonic sensor?**

# L7 -

## Parking sensor:

After the program is downloaded successfully.

When something is too **close** with the ultrasonic,

Digital Tube will **display** "STOP"

Buzzer will make a '**ba ding**' sound.



## Parking sensor level 2:

Add **button** module, to act as an **on/off** switch.

## Parking sensor level 3:

The **closer** the obstacle, the **louder** the buzzer sound.

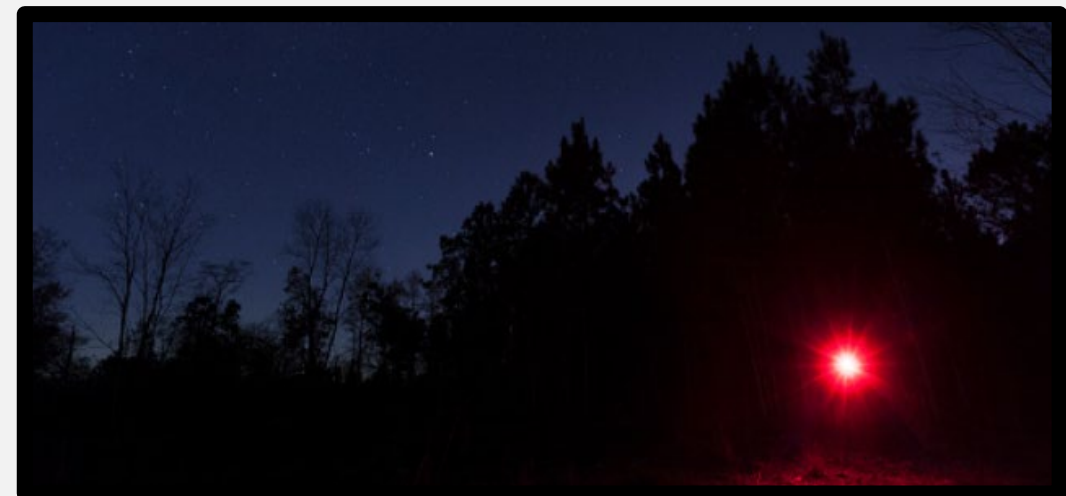
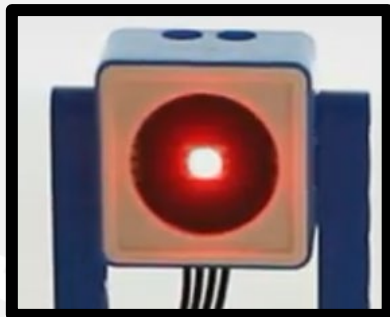
# L7 - Mission

Using 3 modules Ultrasonic , Button, and RGB:

**Danger Lamp Sensor:**

Make a remote for light switch that **turns red and make a sound** when an object is **close** to ultrasonic. When the object get **far** away, the light changes to **white**.

The button functions as an **on/off** switch.



# Today's Topic

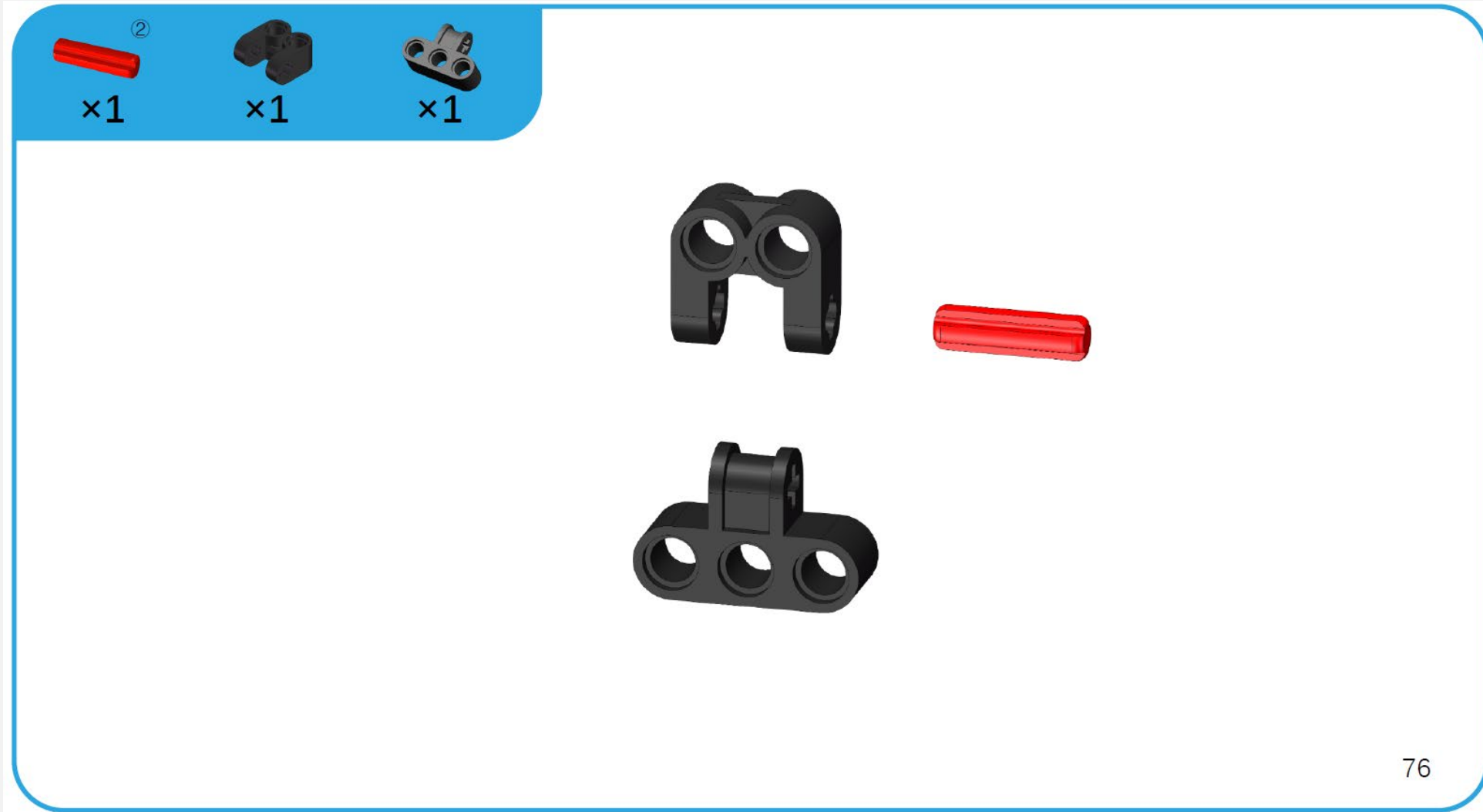
- 1. Build color recognition and thermometer device with World of Modules**
- 2. Color recognition and Temperature-humidity module connection method**
- 3. Color recognition and Temperature-humidity programming**

# Learning Outcome

- 1. Able to build color recognition and thermometer device with instruction manual**
- 2. Understand how color recognition programming work**
- 3. Understand how temperature-humidity programming work**
- 4. Able to program temperature-humidity module**

# Color recogniton

# Step 1



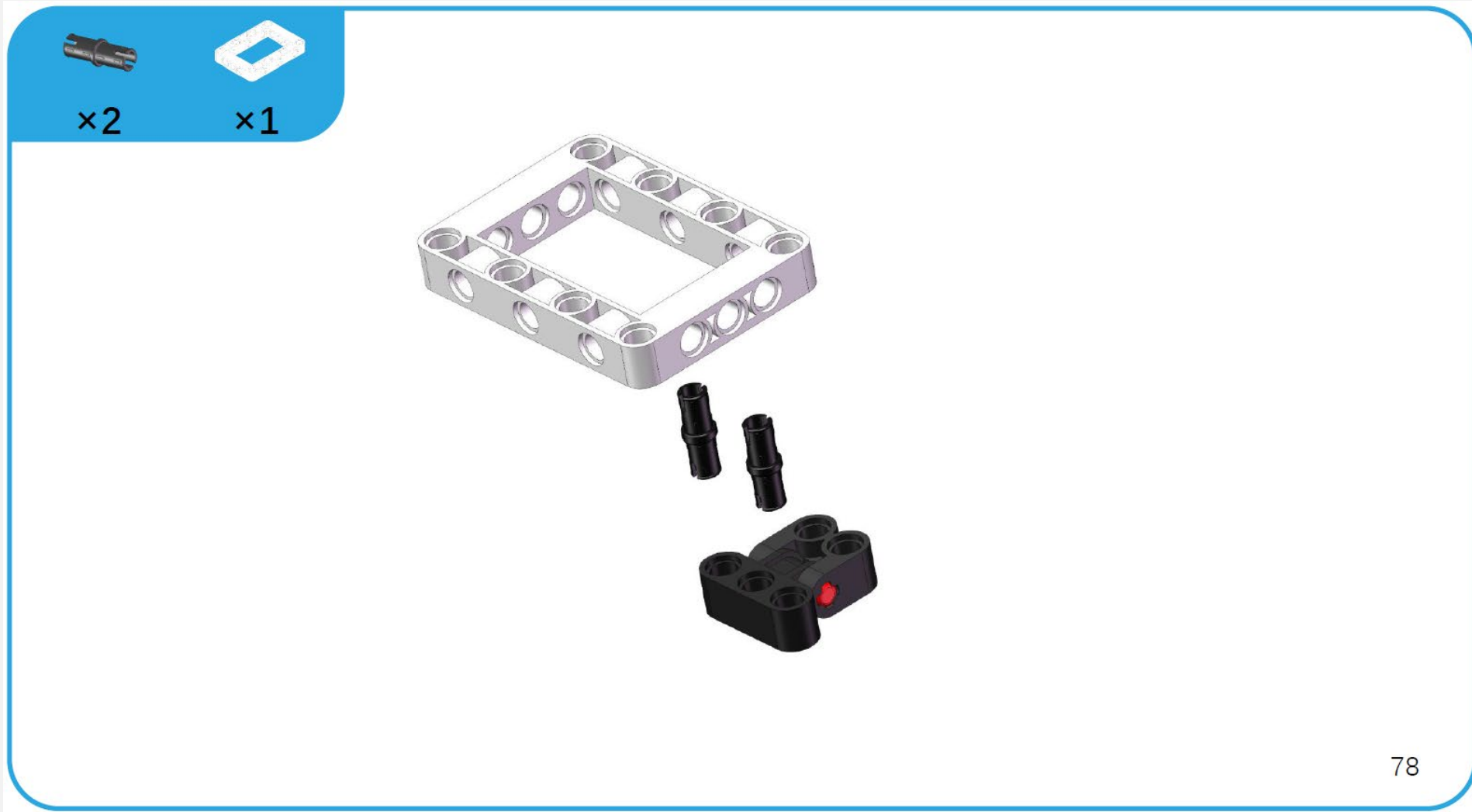
76

## Step 2



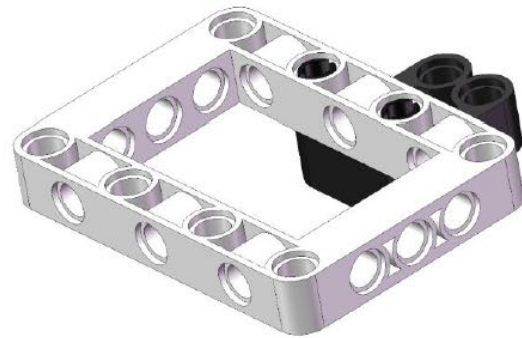
77

# Step 3



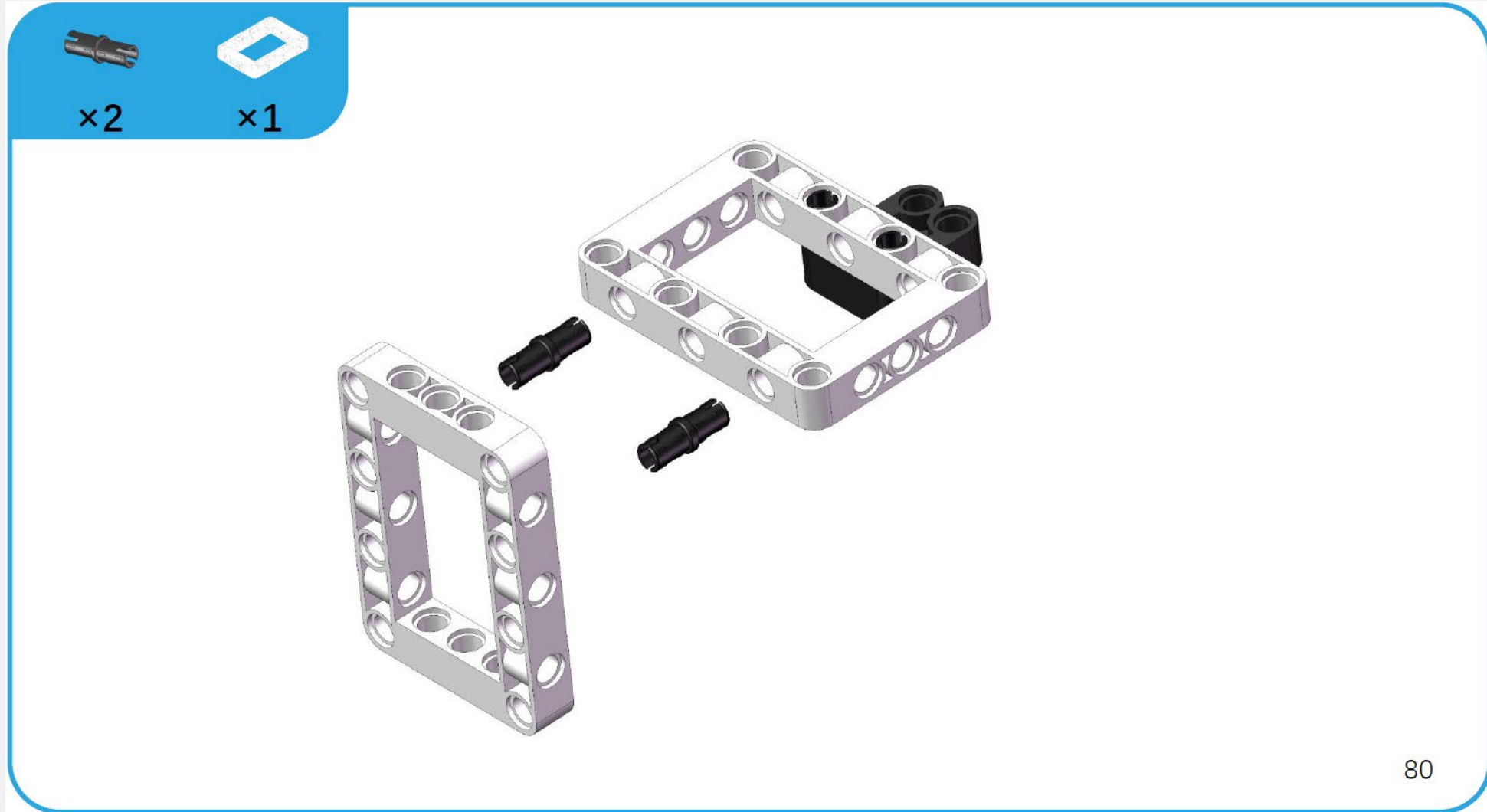
78

## Step 4



79

# Step 5



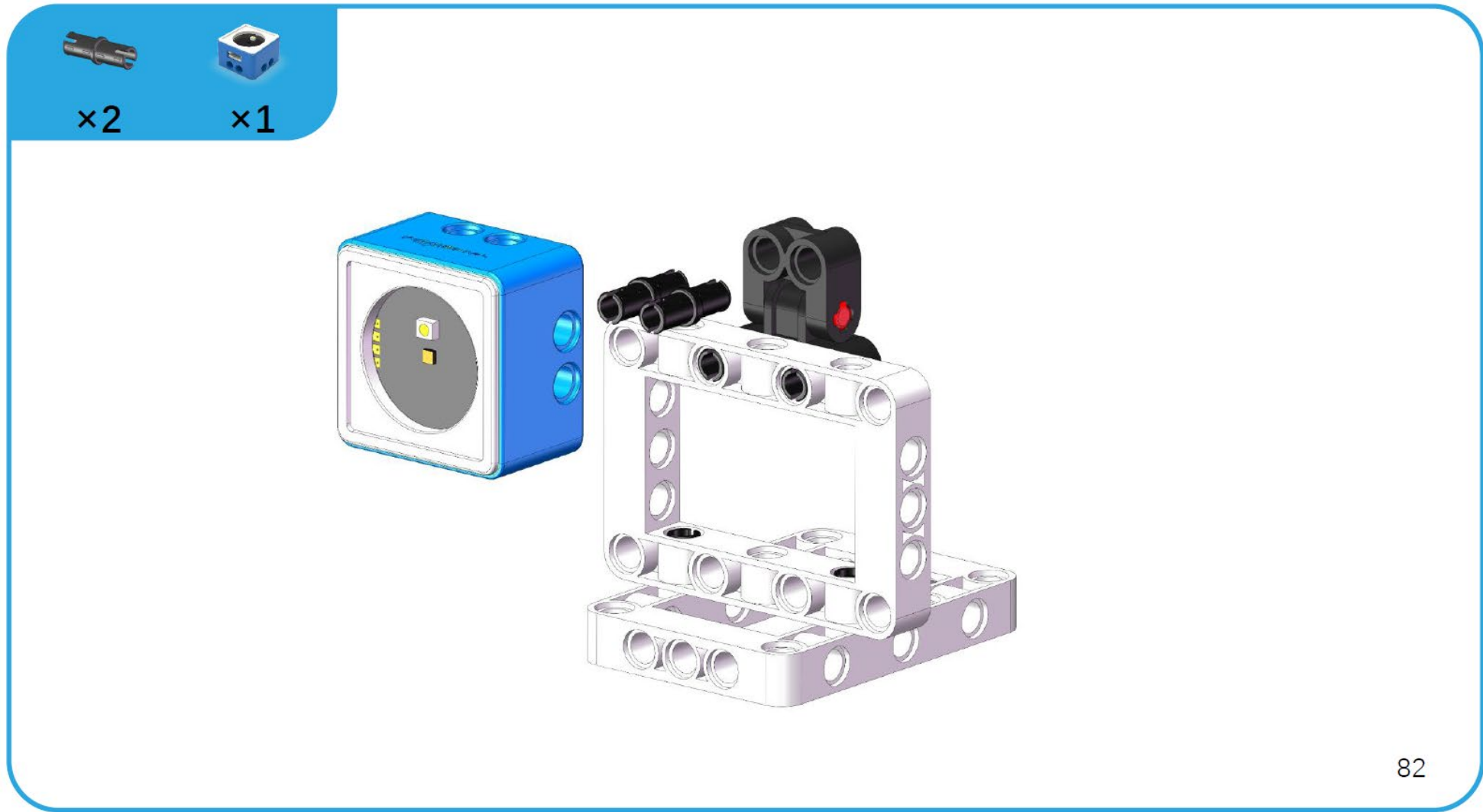
80

## Step 6



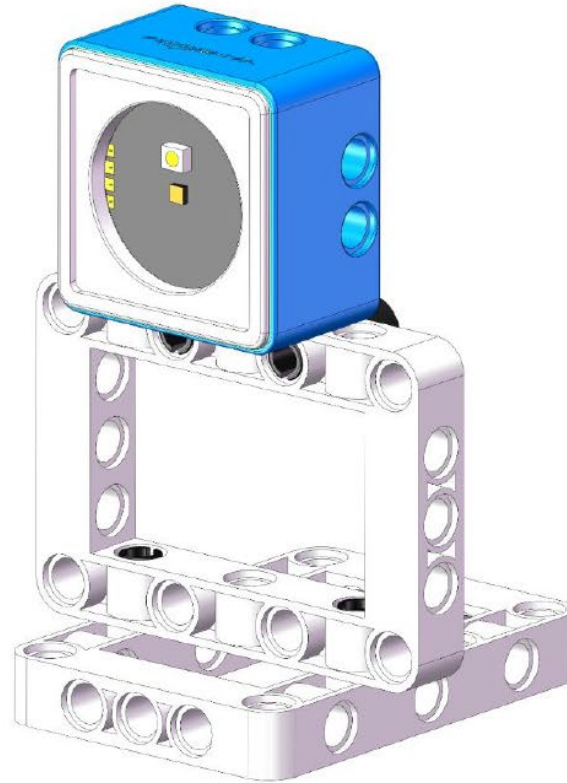
81

# Step 7



82

## Step 8



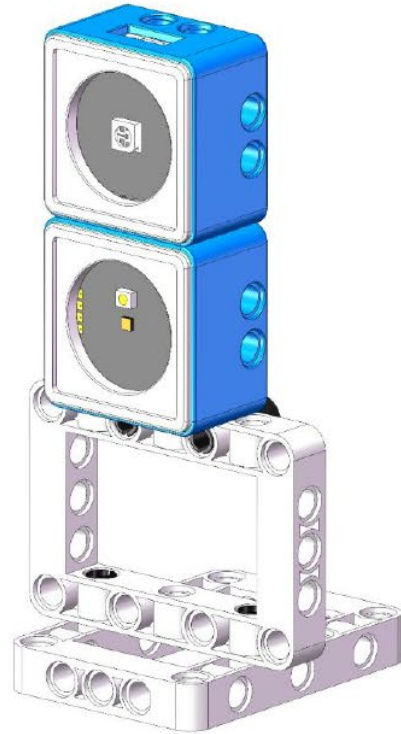
83

# Step 9



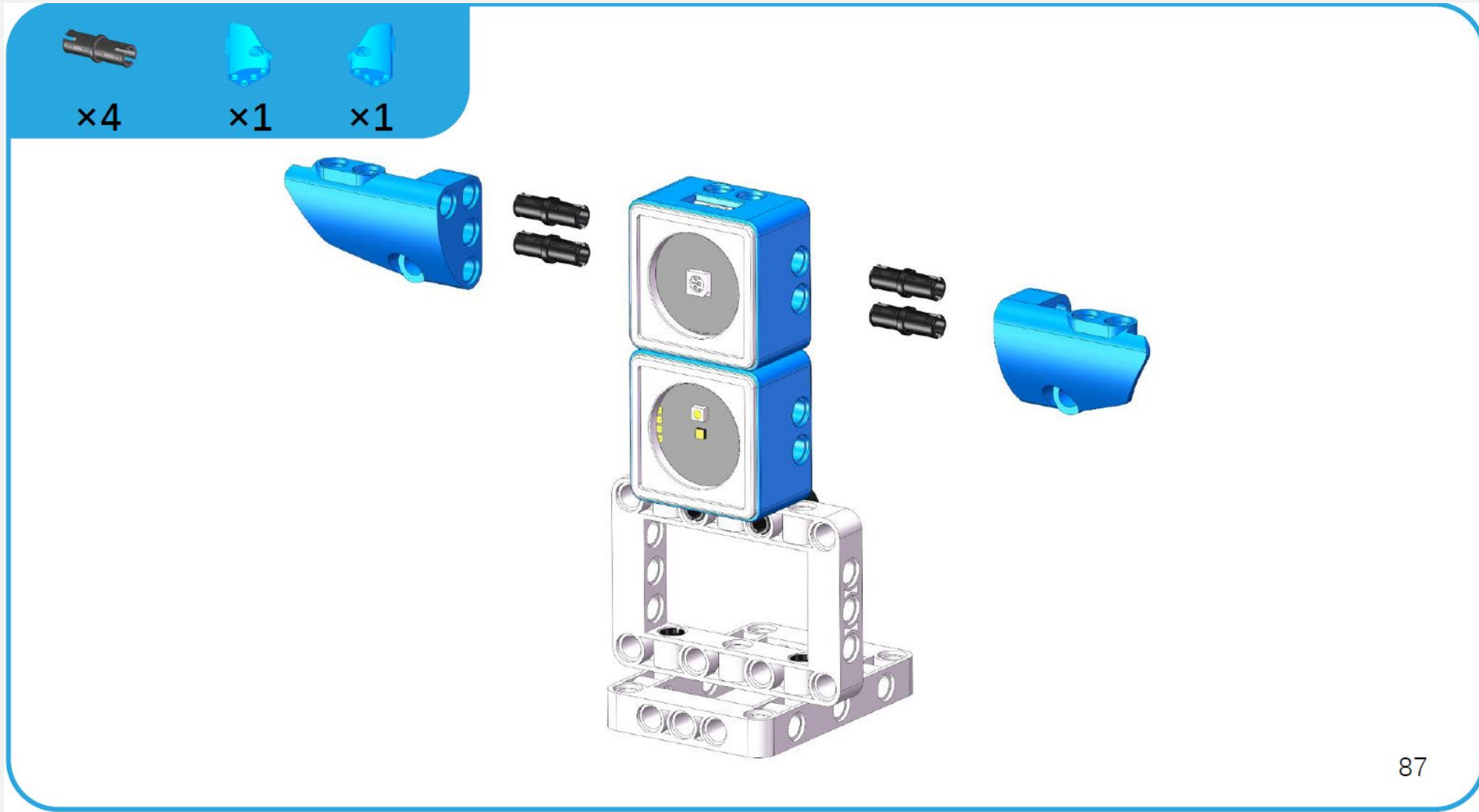
84

# Step 10

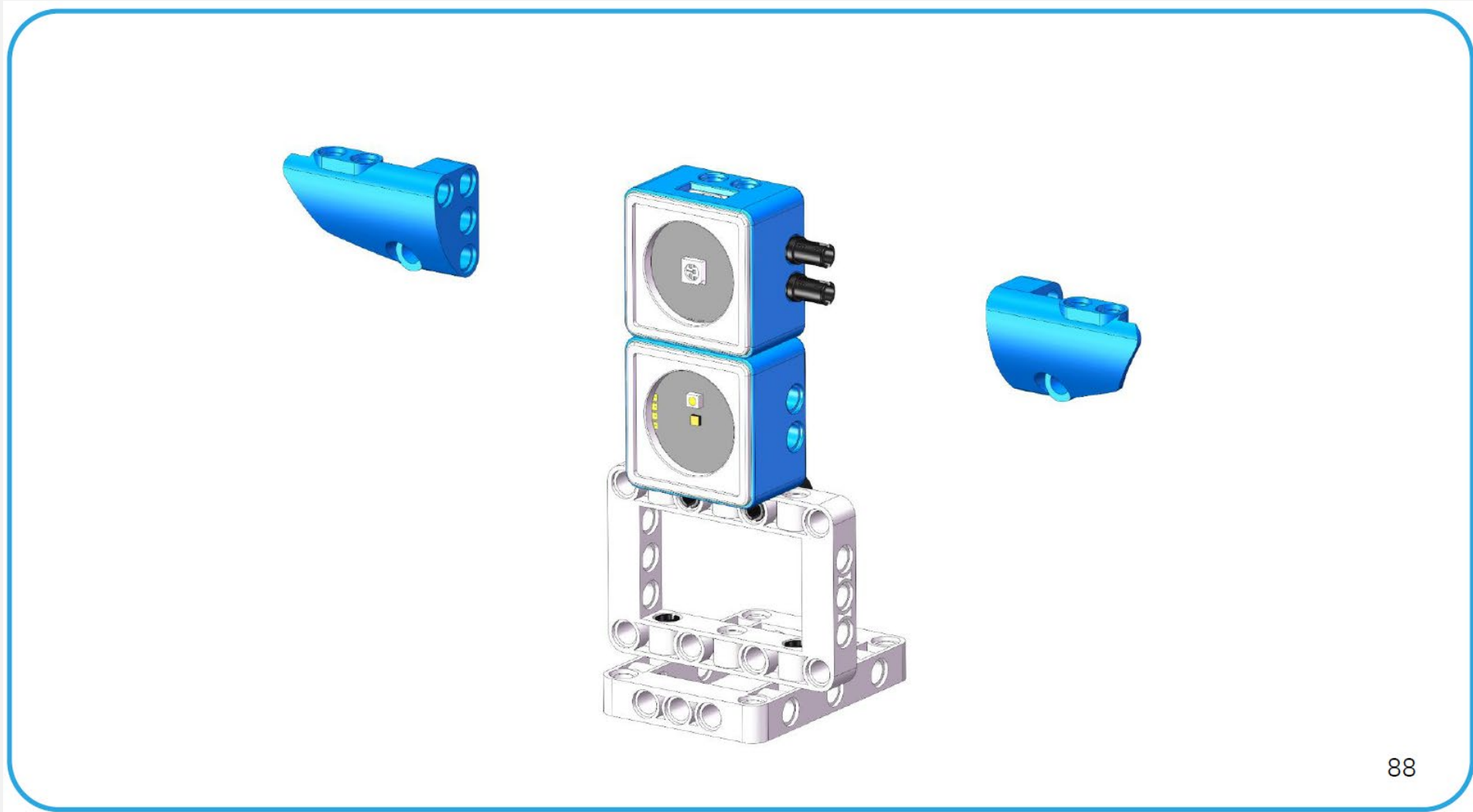


86

# Step 11

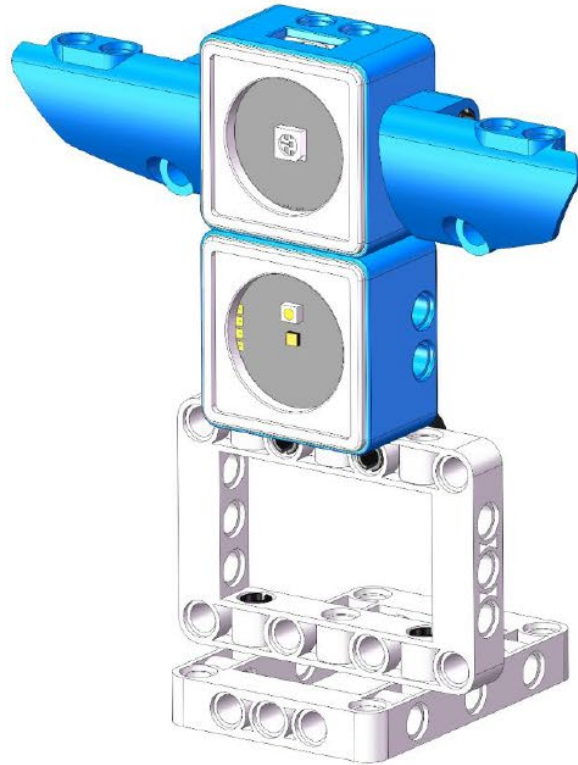


## Step 12



88

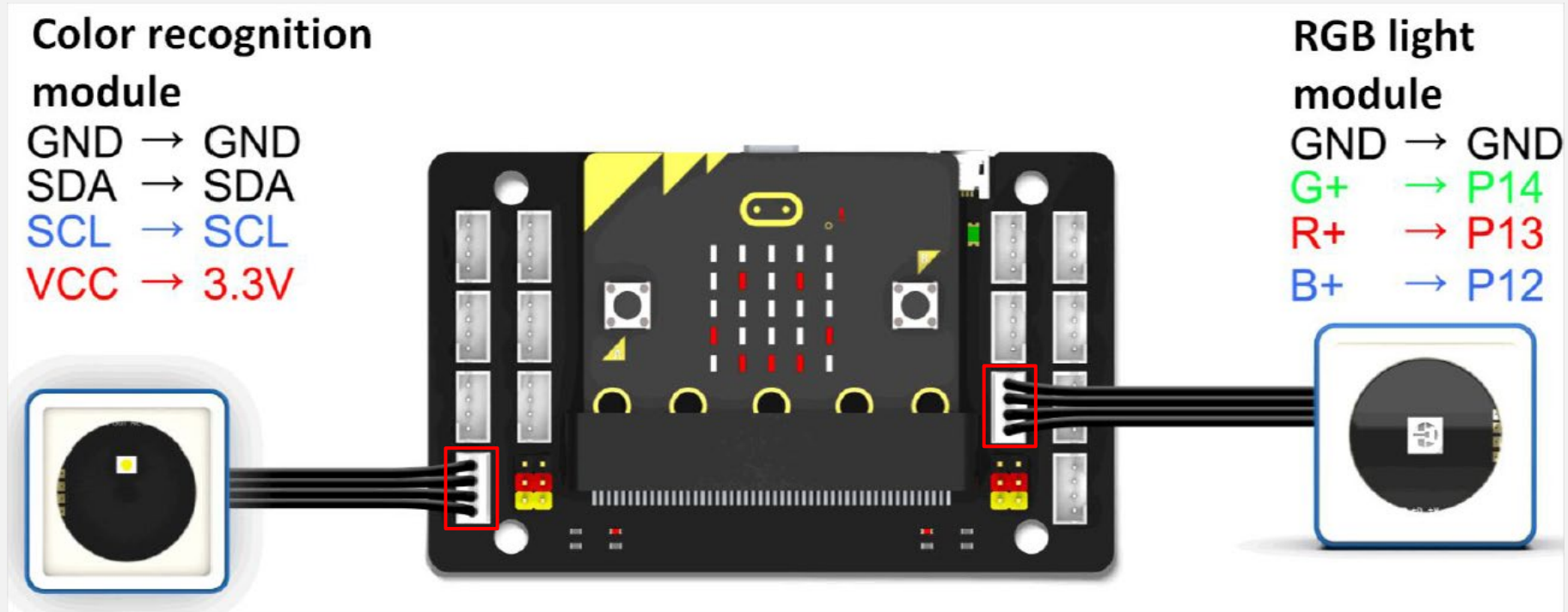
## Step 13



89

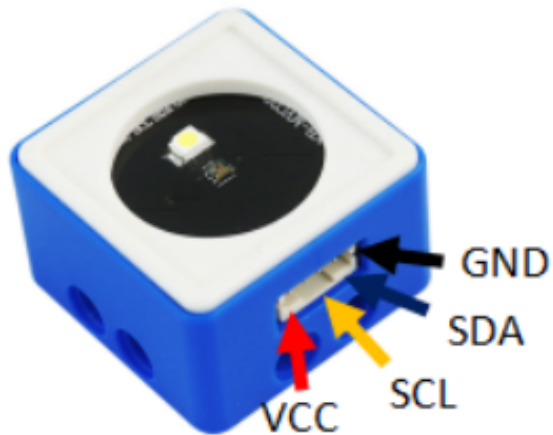
# Wire Connection

# Connect the modules



Let's **connect** the module like this.

# Introducing the Color Recognition Module



GND: connect GND	VCC: connect 3.3V, 5V
SDA: I2C Data transmission port	SCL: I2C sequential transmission port
Working Voltage: 3.3V/5V	Size of module: 29.4mm*28.8mm

Color recognition sensor, also known as color sensor, is a sensor that detects the color by comparing the color of the object with the reference color that has been set previously. When the two colors match within a certain error range, it outputs the detection result. The detection result is output through I2C communication.

# MakeCode Programming

<https://github.com/YahboomTechnology/Module-World>

And

[https://github.com/YahboomTechnology/Module\\_World\\_Color](https://github.com/YahboomTechnology/Module_World_Color).

Or search **YahboomTechnology/Module-World** in the extension block

Or search **YahboomTechnology/Module\_World\_Color** in the extension block

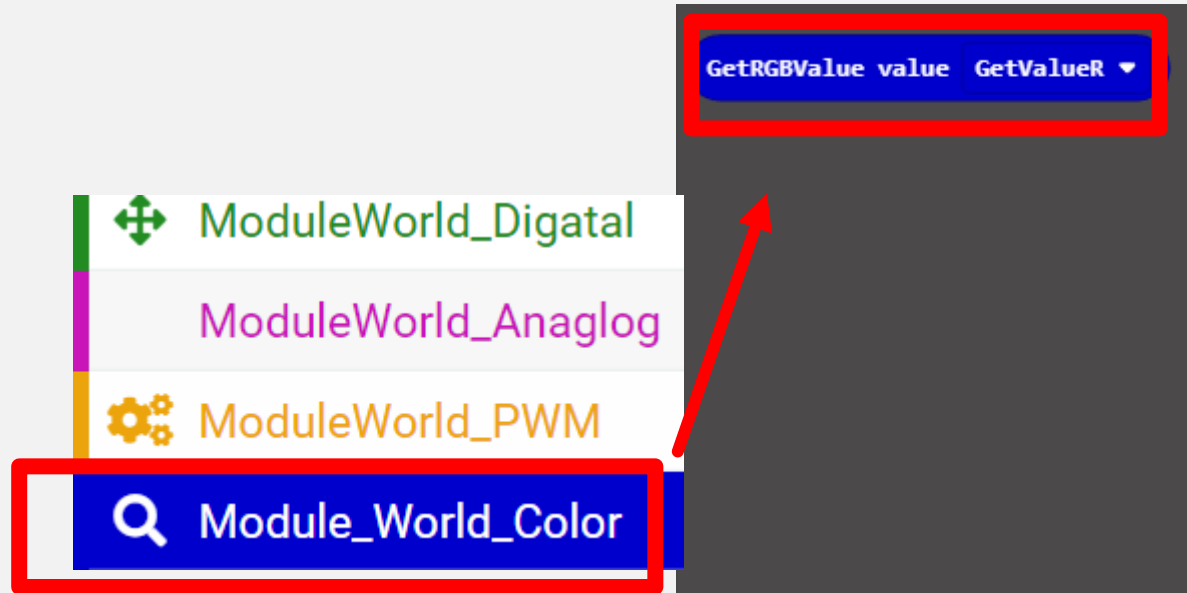
# Coding - Basic

The image shows the Microbit coding interface. On the left is a sidebar with various categories: Basic (highlighted with a red box), Input, Music, Led, Radio, Loops, Logic, Variables, Math, ModuleWorld\_Digital, ModuleWorld\_Analog, and ModuleWorld\_PWM. A red arrow points from the Basic category in the sidebar to the 'pause (ms)' block in the main workspace. The main workspace contains a script with the following blocks: 'show icon' (with a dropdown menu), 'show string' (with the text 'Hello!'), 'clear screen', 'forever' (loop block), 'on start' (event block), and 'pause (ms)' (with the value '100' and a dropdown menu, highlighted with a red box).

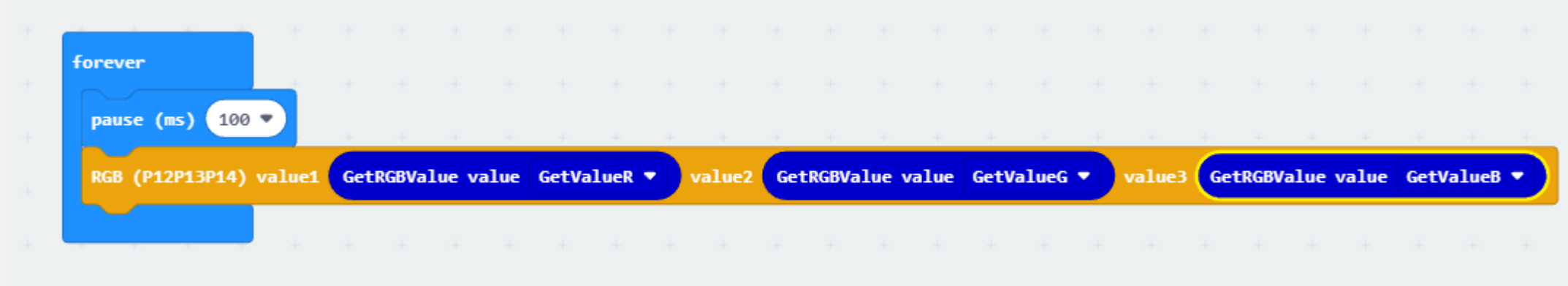
# Coding - ModuleWorld\_PWM



# Coding - Module\_World\_Color

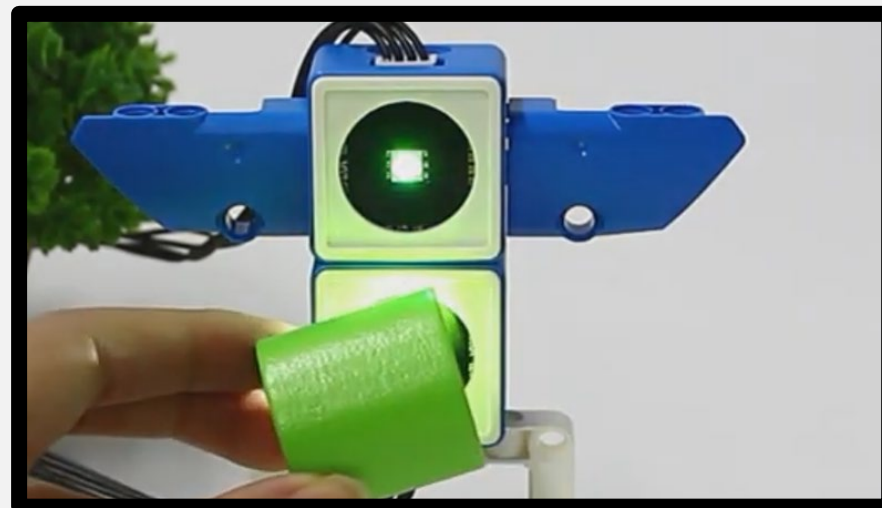
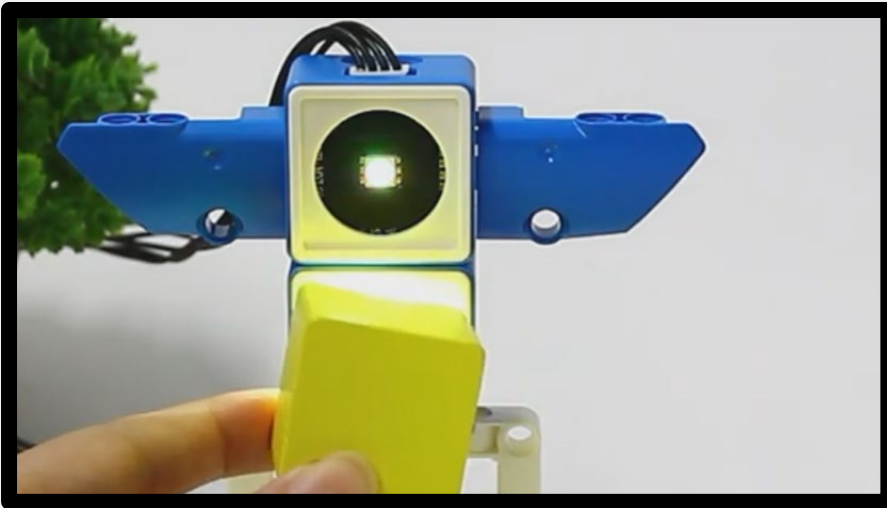


# Coding - Combine Blocks



# Phenomenon

After the program is downloaded successfully. When the color sensor recognizes **different** colors, the RGB lights display the **corresponding colors**.

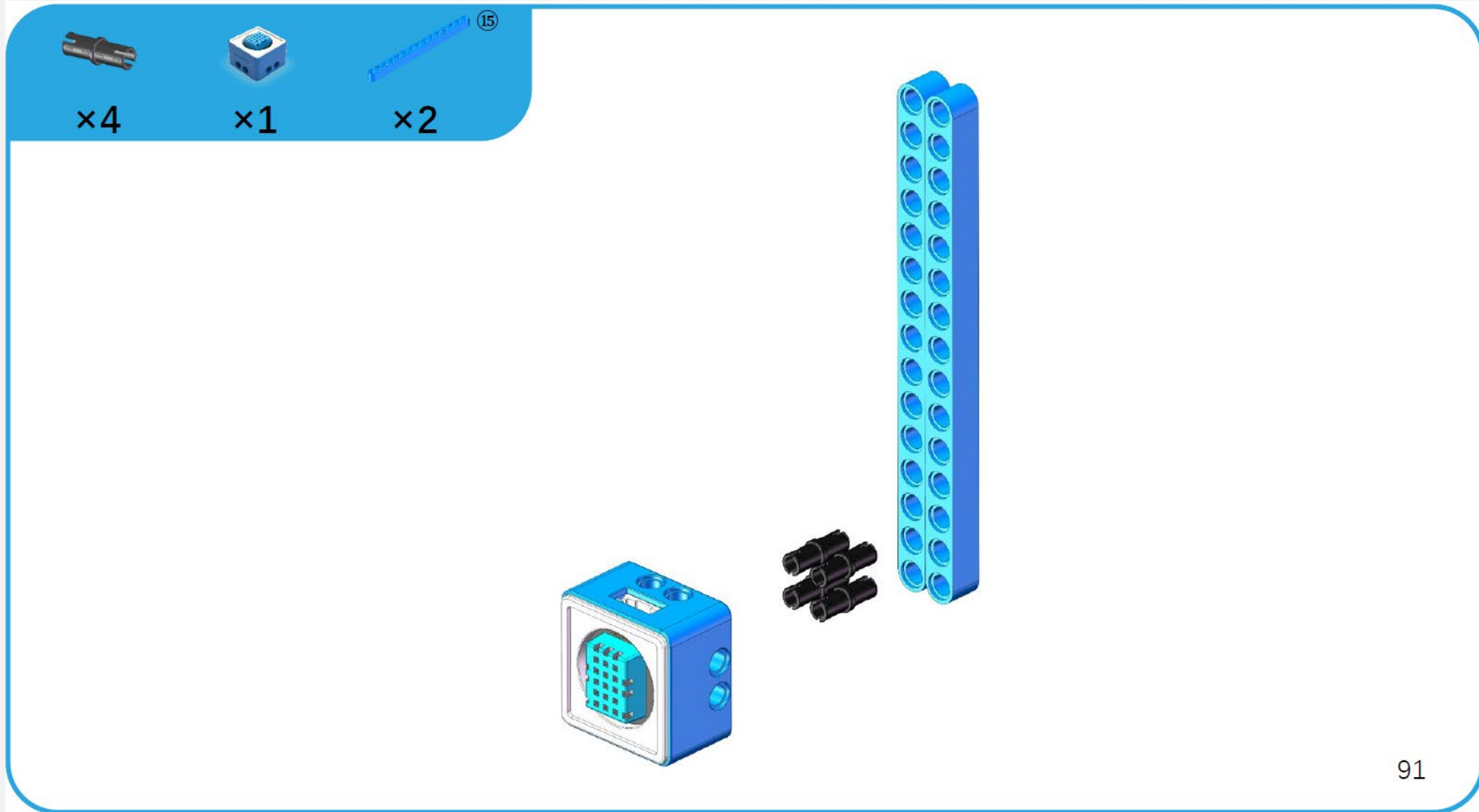


# Real life application

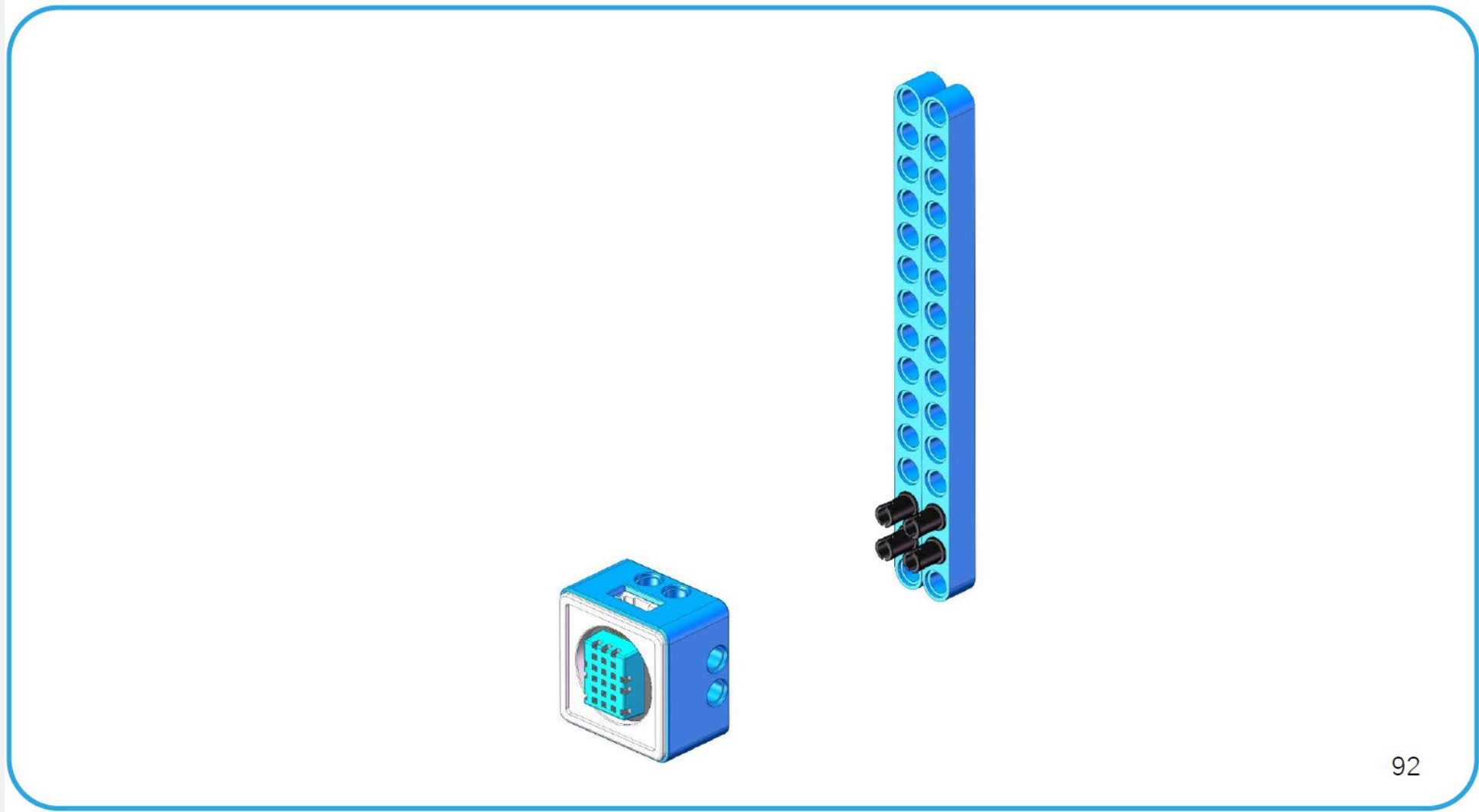


# Thermometer

# Step 1

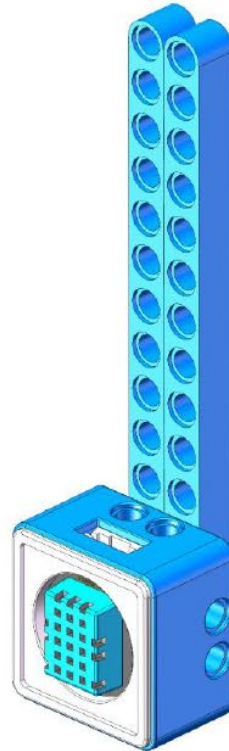


## Step 2



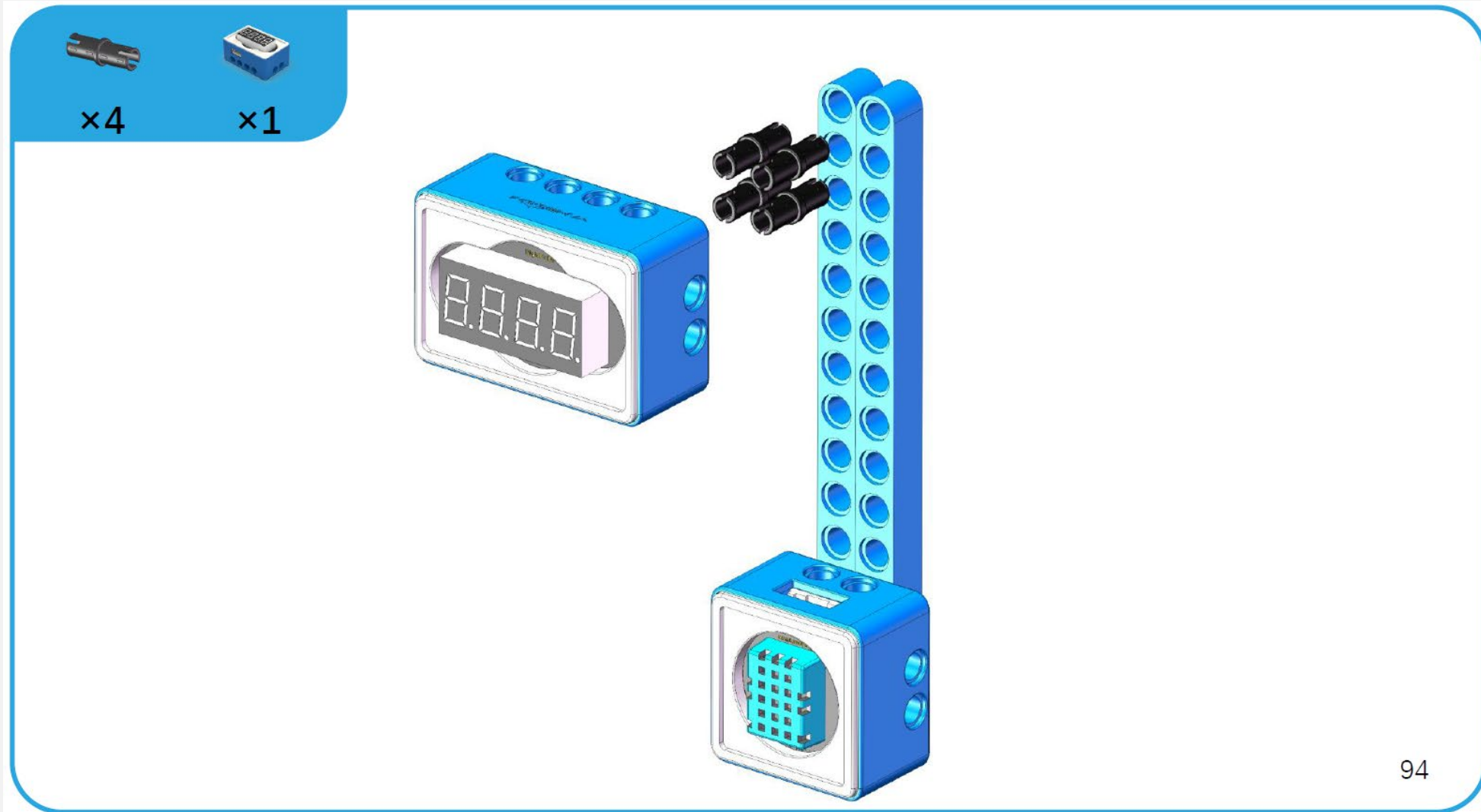
92

## Step 3



93

# Step 4



## Step 5

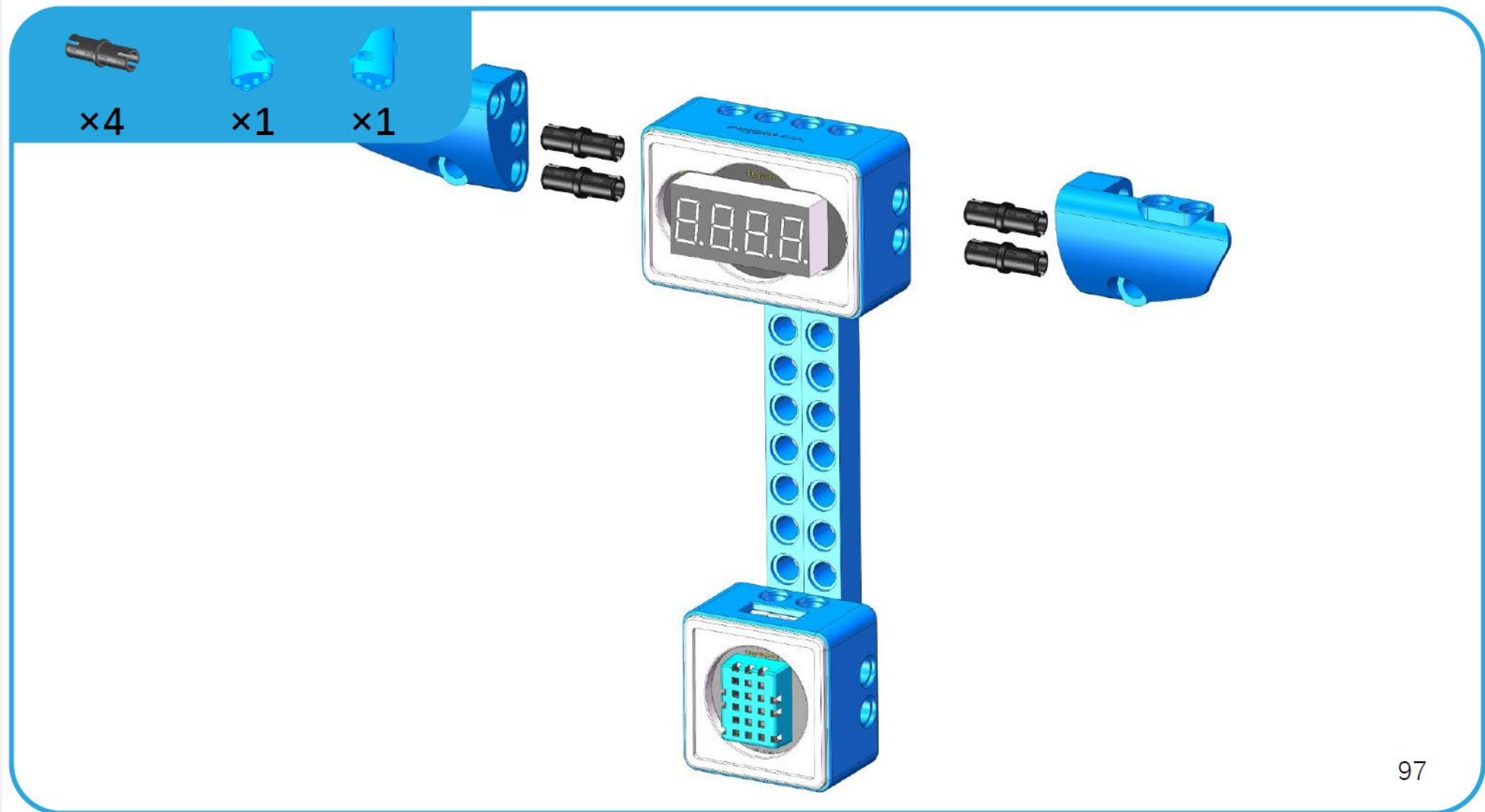


95

## Step 6



# Step 7



## Step 8



98

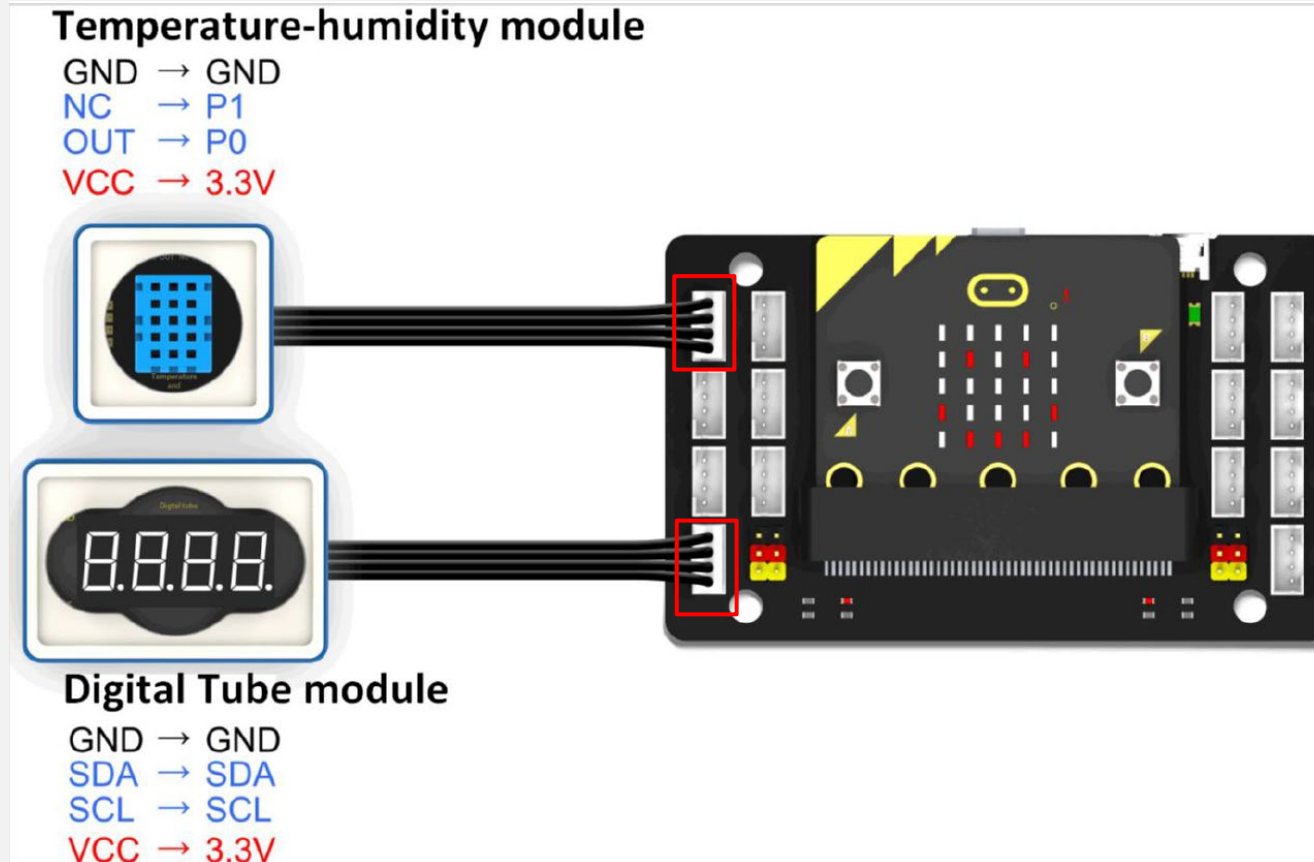
## Step 9



99

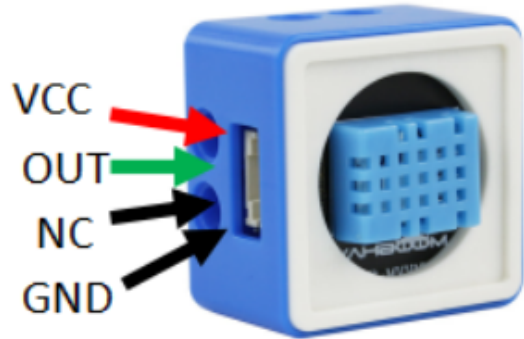
# Wire Connection

# Connect the modules



**Let's connect the module like this.**

# Introducing the Humidity and Temperature module



GND: connect <u>gnd</u>	OUT: Single bus digital signal output	+: Connect 3.3V, 5V
Range	Humidity: 20-90%RH	Temperature: 0~50 °C
Accuracy	Humidity: ±5%RH	Temperature: ±2 °C
Resolution	Humidity: 1%RH	Temperature: 1 °C

DHT11 is a temperature and humidity sensor. Its accuracy range, humidity: + -5% RH, temperature: + -2 °C. Measuring range, humidity: 20-90% RH, temperature: 0 ~ 50 °C.

# MakeCode Programming

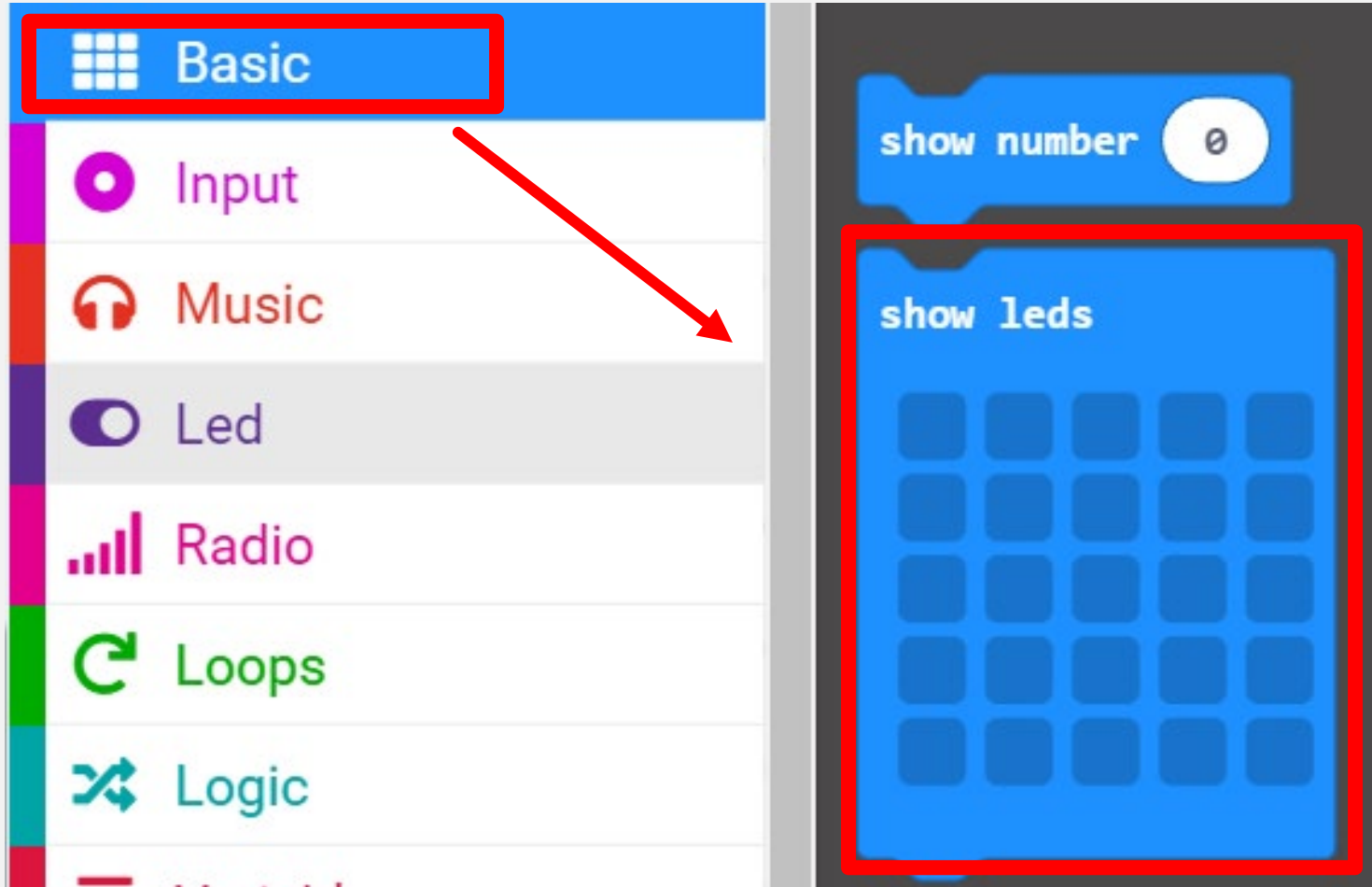
<https://github.com/YahboomTechnology/Module-World>

And

<https://github.com/YahboomTechnology/tm1650>

**Or search [YahboomTechnology/Module-World](https://github.com/YahboomTechnology/Module-World) in the extension block**

# Coding - Basic



# Coding - TM1650

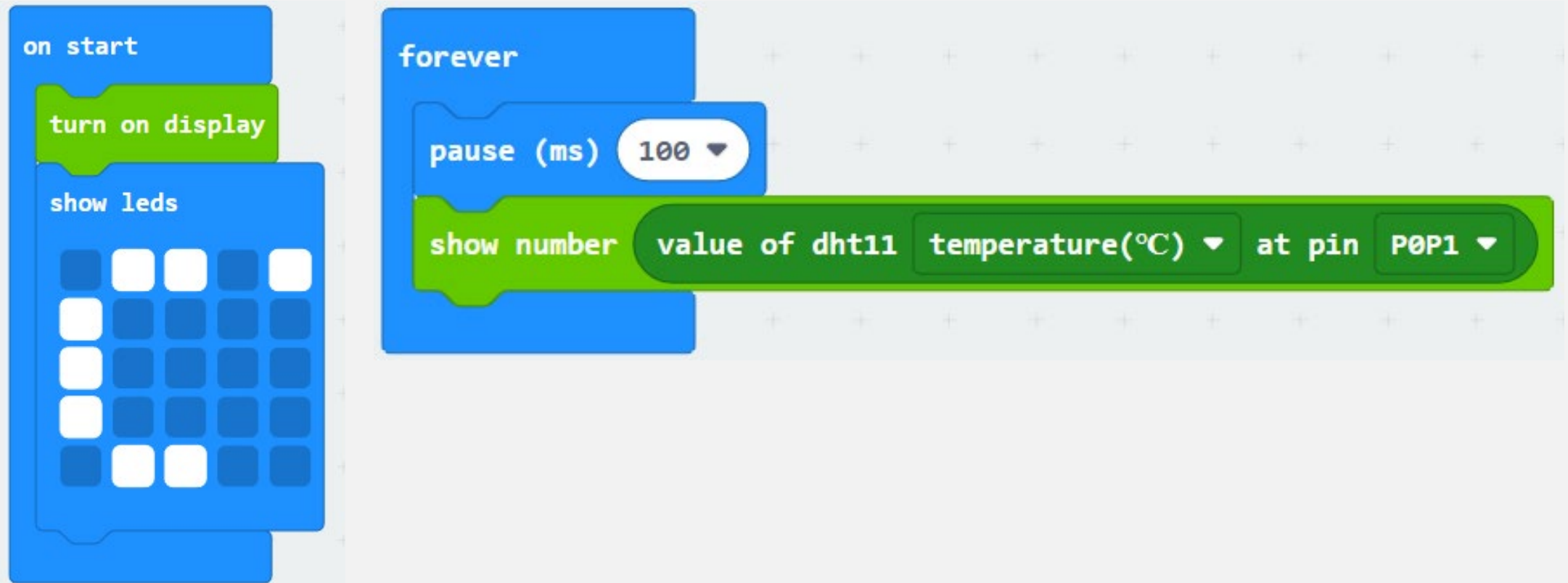
The screenshot displays the Advanced Superlogic coding interface. On the left, a vertical menu lists various modules: Led, Radio, Loops, Logic, Variables, Math, ModuleWorld\_Digital, ModuleWorld\_Analog, ModuleWorld\_PWM, Module\_World\_Color, and 8 TM1650. The '8 TM1650' module is highlighted with a red box, and a red arrow points from it to the 'turn on display' block in the script. The script itself consists of several green blocks: 'show number' (value 100), 'show hex number' (value 123), 'scroll display' (text "", rolling time 0), 'show dot point' (value 0, show true), 'show digit' (value 0 at 0), 'set intensity' (value 3), 'turn off display', and 'turn on display'. The 'show number' and 'turn on display' blocks are also highlighted with red boxes.

# Coding -

## ModuleWorld\_Digital

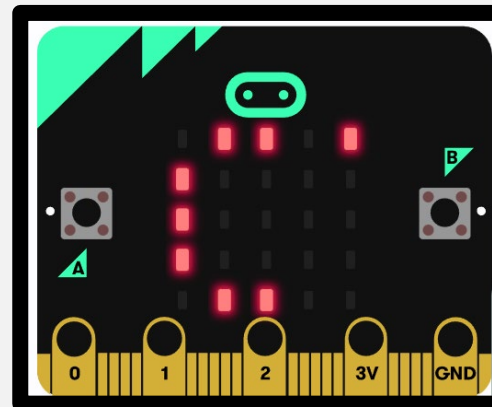
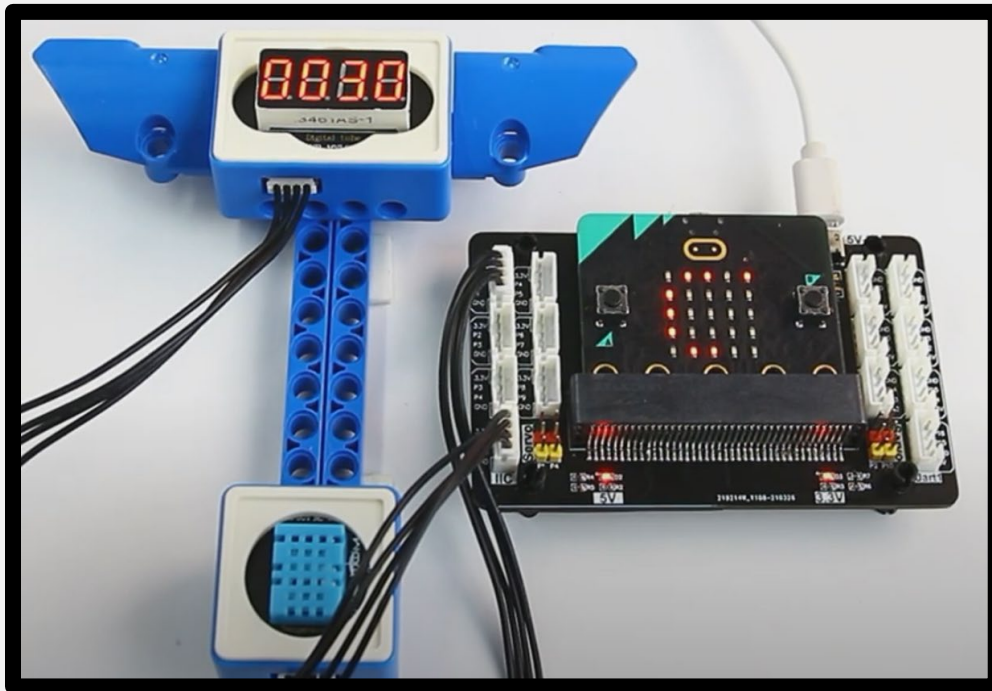
The screenshot displays the ModuleWorld\_Digital coding environment. On the left, a vertical sidebar lists various modules: Basic, Input, Music, Led, Radio, Loops, Logic, Variables, Math, and ModuleWorld\_Digital. The ModuleWorld\_Digital module is highlighted with a red box. A red arrow points from this box to the top block in the main workspace. The main workspace contains several green blocks for digital sensors, all configured with pin P0P1. The top block, also highlighted with a red box, is 'Dht11 temperature(°C) at pin P0P1'. Below it are blocks for 'Ultrasonic pin P0P1', 'PIR pin P0P1 value NoPIR', 'IR pin P0P1 value Obstacle', 'Button pin P0P1 value Press', 'Collision pin P0P1 value NoCollision', and 'Vibration pin P0P1 get'.

# Coding - Combine Blocks



# Phenomenon

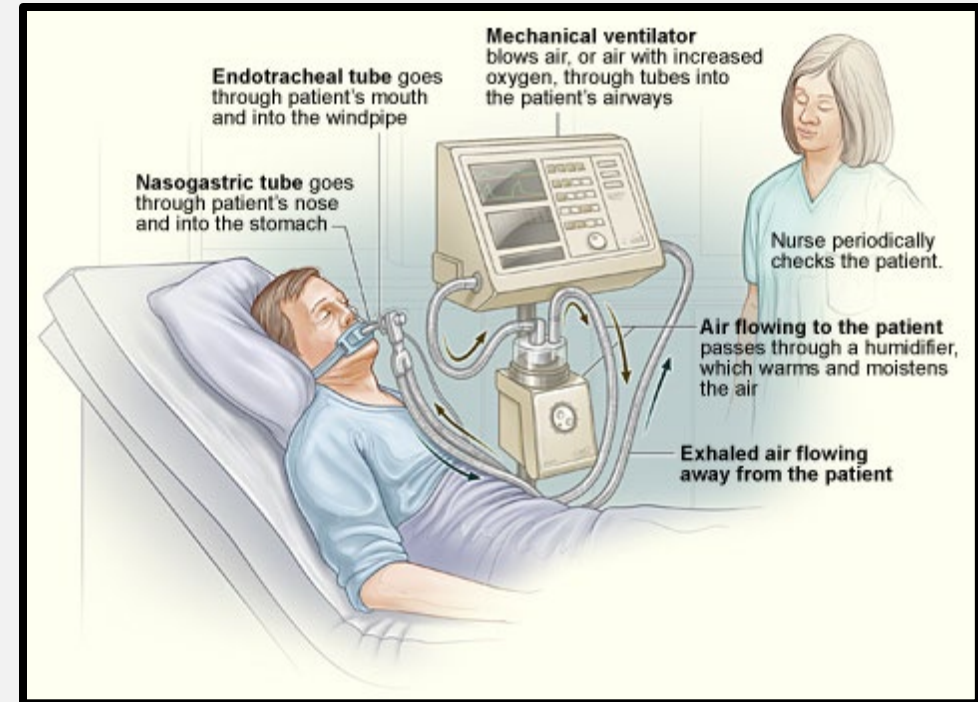
After the program is downloaded successfully. The **temperature** and **humidity** of the current environment will be **displayed** on the digital tube module.



# Real life application



**Air  
conditioner**



**Ventilat  
or**

**Do a quick self-check of your learning outcome...**

- 1. Which port should the color recognition module plug into?**
- 2. What application can the color recognition module do in real life?**
- 3. What functions are the temperature and humidity modules?**
- 4. Give 2 real life examples with the application of temperature and humidity modules.**

# CHALLENGE

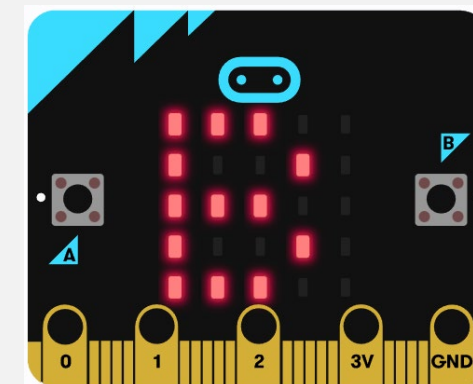
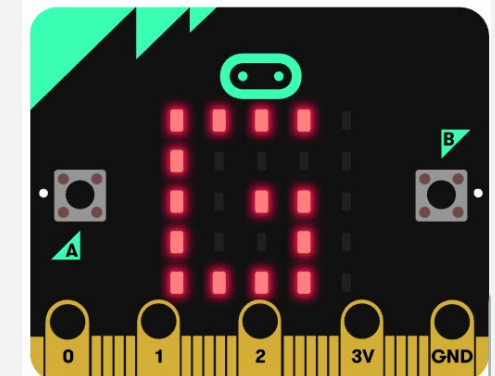
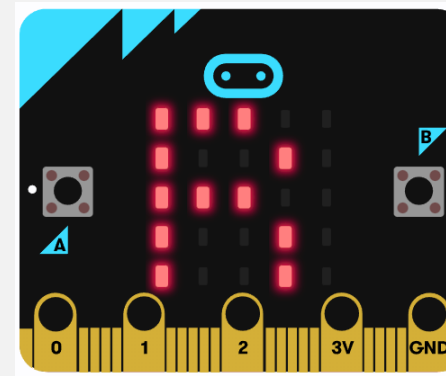
## for : Lesson 8

# L8 -

After the program is downloaded successfully.  
When the color sensor recognizes **different** colors, the RGB lights display the **corresponding** colors.

## Condition:

1. When RGB display **Red**, LED show **"R"**
2. When RGB display **Green**, LED show **"G"**
3. When RGB display **Blue**, LED show **"B"**

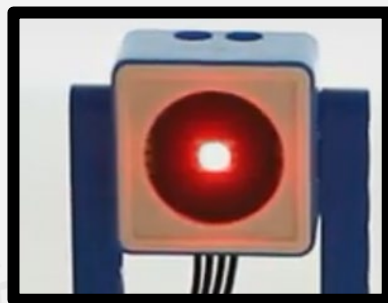


## L8 - Challenge level 2

Using 3 modules Color recognition, Button, and RGB:

### Chameleon Lamp:

Make a torch light that can **sense** and **changes to** the corresponding **color**. The use of a button serves as an **on/off** switch.

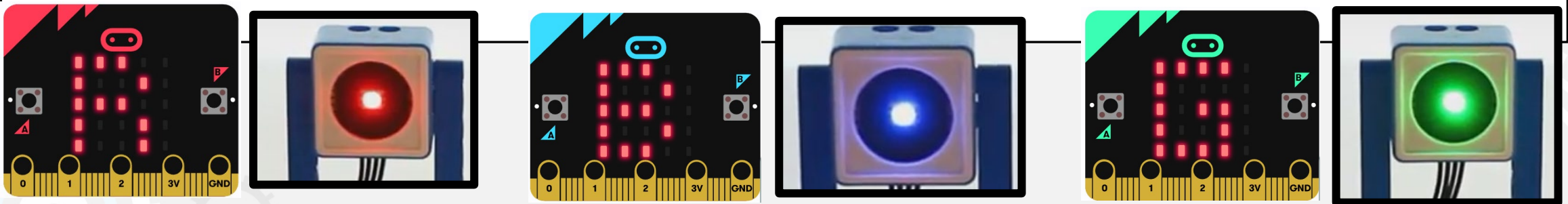


# L8 - Mission

After the program is downloaded successfully.

LED will **randomly** pick between **R(red)**, **G(green)**, **B(blue)**

1. When the LED display 'R'(red), if the color sensor does **not** recognize color red, the buzzer will make a sound.
2. If the color sensor **does** recognize color red, **RGB** will turns to red and LED will display **another** color (R/G/B).





Any  
Questions?

Thank you :)