makeblock

# Educator Guide
# for mBot2 Educational Kit

# Table of Contents

## Welcome

Welcome to use the mBot2 Educational Kit.

mBot2 is a robot designed for programming education. By operating and programming it, children can explore mechanics, electronics, and software engineering. This guide aims to help educators to organize and carry out robotics-related classes. Whether a programming beginner or an experienced educator, you can find the right educational resources for you and your students.

This guide will walk you through mBot2 assembly, programming, and application, covering both basic and advanced knowledge.
Let mBot2 be your powerful assistant in helping you and your students explore the technology world. Ready? Go!

## Read before use

- Not suitable for children under 3.
- Children under 8 need to be accompanied by educators when using this product.
- To avoid damage as a result of falling down, please don't put the product on the edges of high places.
- To avoid product damage, please don't disassemble, repair or modify this product yourself.
- To avoid product damage or safety incidents, don't place this product in water, fire, or environments of high humidity, temperature, or altitude.
- Don't use this product or charge this product in any environment with a temperature above the operating temperature range of this product.
- Fully charge the product for storage and recharge it every 3 months when the product is left unused.
- Use the adapter included or a 5 V/2 A adapter to charge the product.
- The adapter used to charge the product cannot be used as a toy.
- Make sure that the product is powered off and disconnected from any external power supply when liquid is used to clean the product.
- The product should be connected to the power supply in strict accordance with the limit on the number of power supplies.
- Check the wires, plugs, housings or other parts periodically and stop using them if any damage has been found until they are well repaired.

# 1. Meet mBot2 Educational Kit

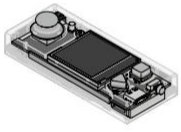
## Overview

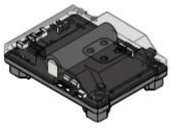
mBot2 Educational Kit is an education solution designed for educators to carry out programming classes effectively. Boasting a storage box and charging function, this kit is a good assistant for educators.
A highly integrated and extensible robot, mBot2 is powered by CyberPi. mBot2 supports both graphical and Python programming. With mBot2, students can:

- Learn graphical programming and Python programming, leveling up programming capabilities.
- Learn how to collect data from the many sensors of mBot2, such as the ultrasonic sensor, quad RGB sensor, and onboard sensors.
- Showcase creativity and ideas on the full-color display with different audio and visual effects.
- Launch AIoT projects by leveraging mBlock 5 and mBot2.

## Parts list

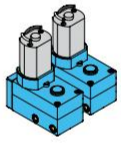
### Parts in mBot2 storage box

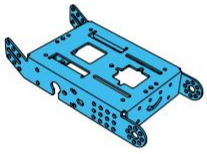

CyberPi


mBot2 shield


Ultrasonic sensor 2


Quad RGB sensor


Encoder motor


Chassis


USB cable


Wheel


Mini wheel


Bluetooth dongle


Motor cable


mBuild cable (10 cm)


mBuild cable (20 cm)


Screwdriver


Storage box


Screw M4*25 mm

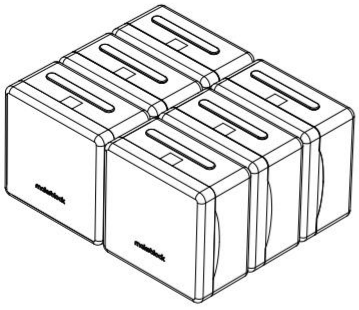

Screw M4*14 mm


Screw M4*8 mm

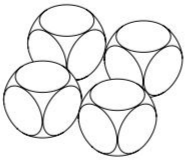

Screw M2.5*12 mm


Quick start guide

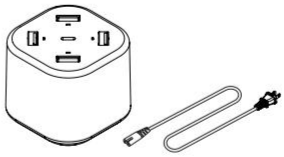
## Parts in mBot2 educational kit storage box



mBot2 storage box

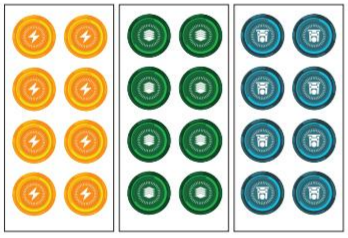


Energy block

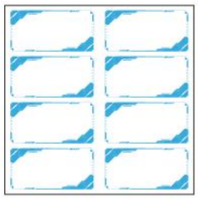


Power charger & cable

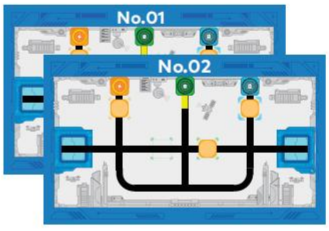

Educator guide



Energy sticker

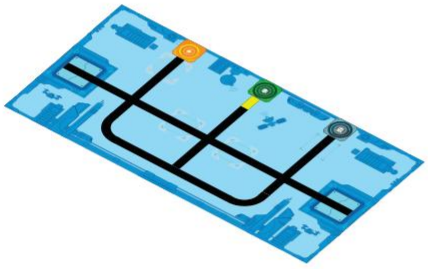


mBot2 sticker

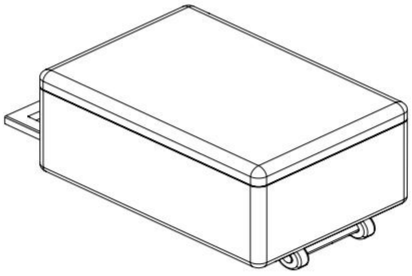



Task card

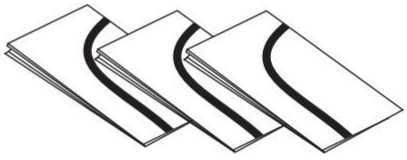

Programming Projects



Future City map



mBot2 educational kit storage box



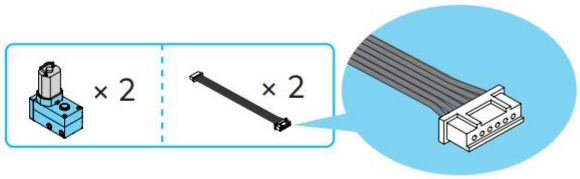
Line-following map

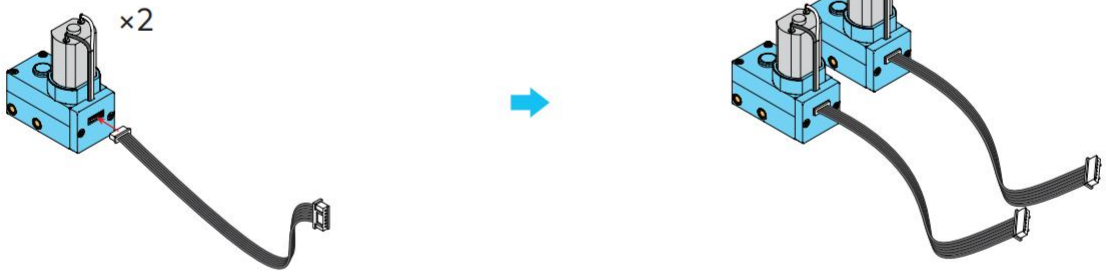
# 2. Quick guide

## Assemble mBot2

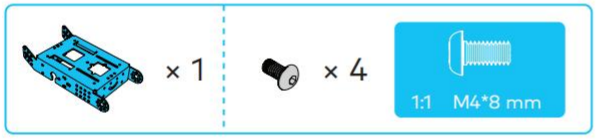
In your first class, help your students assemble mBot2 by performing the following steps.

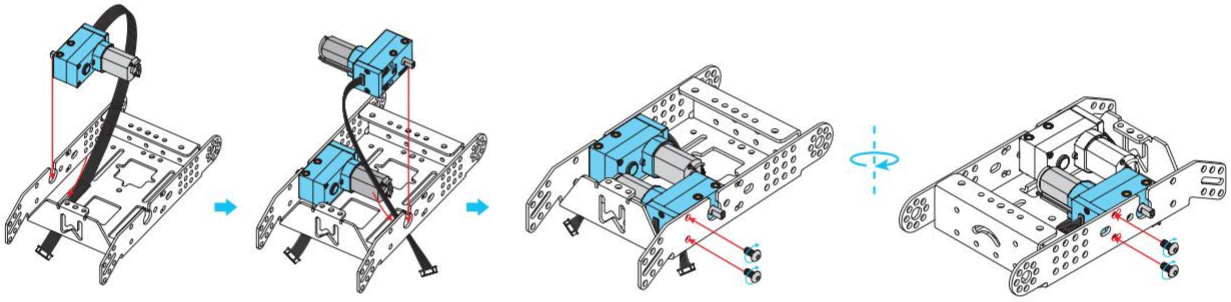
①



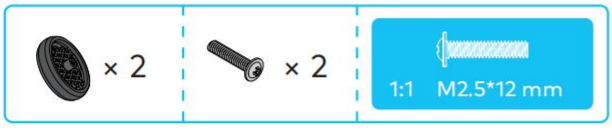



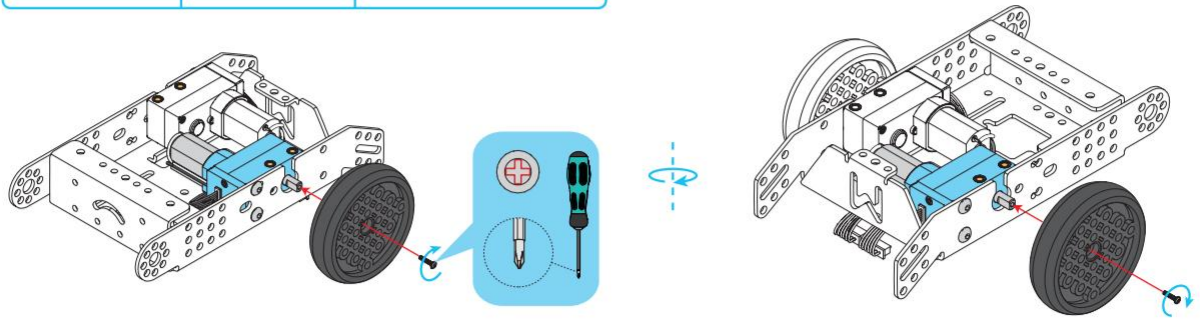

②



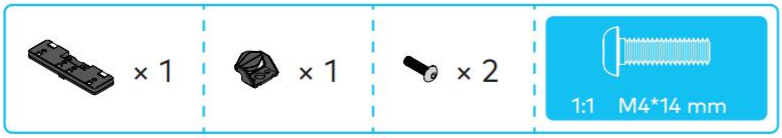



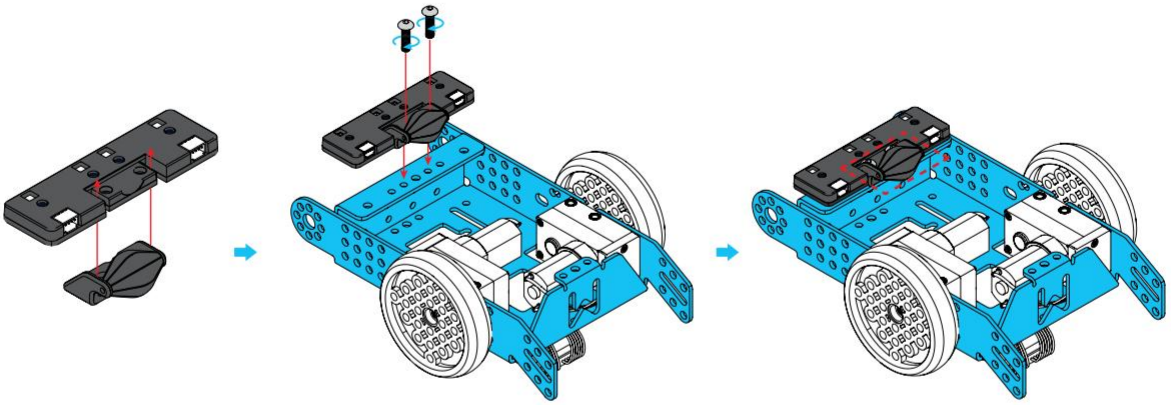
③



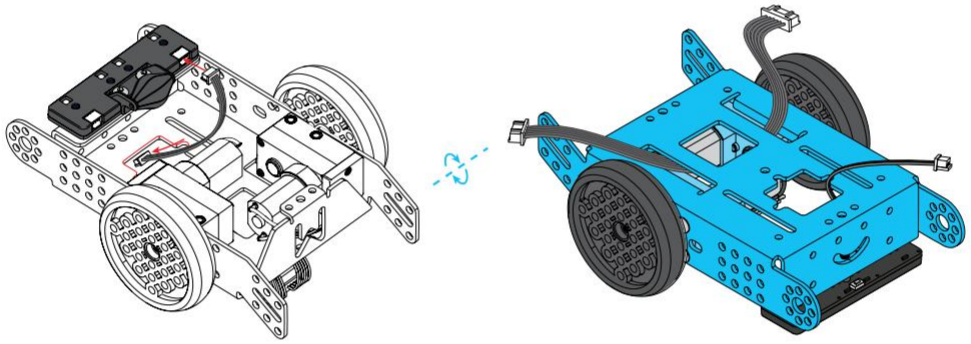



4

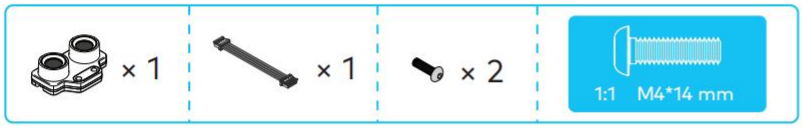


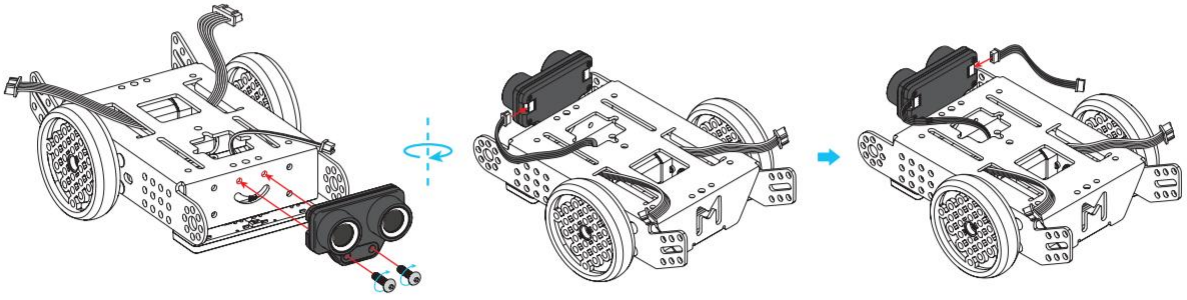



5

6



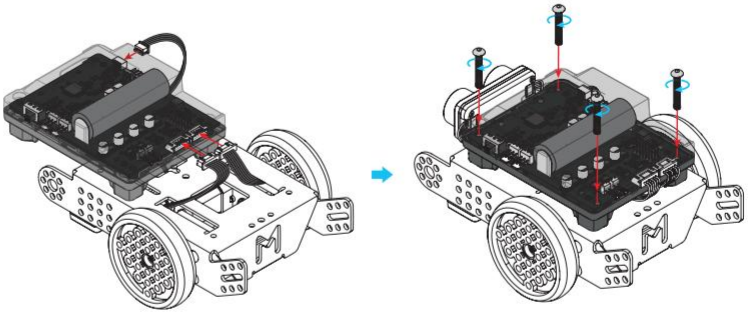



⑦

 × 1

 × 4

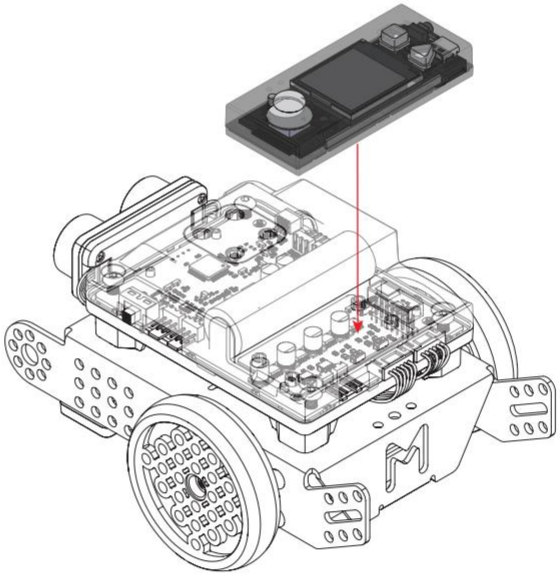

1:1 M4*25 mm



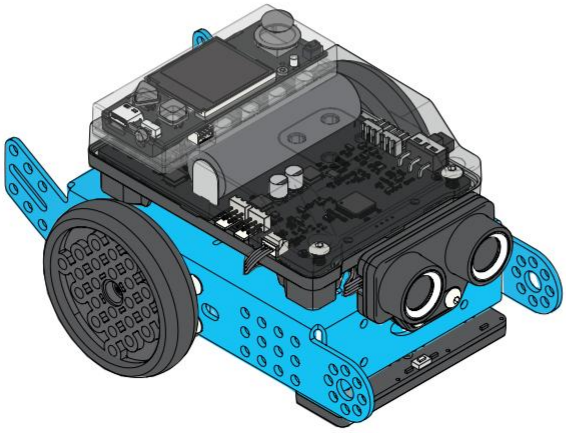
Before tightening the screws, make sure that the connection cables are not pressed by mBot2 shield.

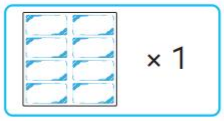
⑧

 × 1

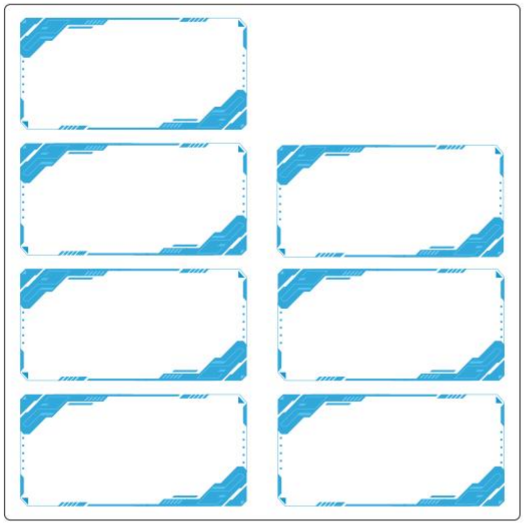


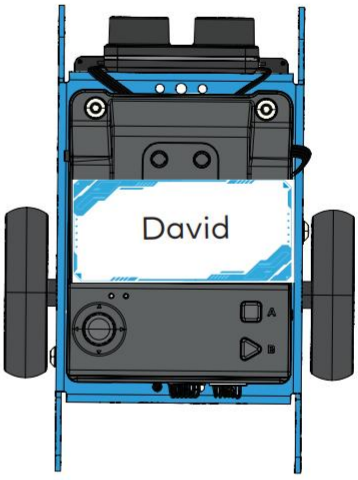
⑨



⑩  × 1

(Optional) Ask six students to write their names on the stickers, and stick the sticker on the top of mBot2.



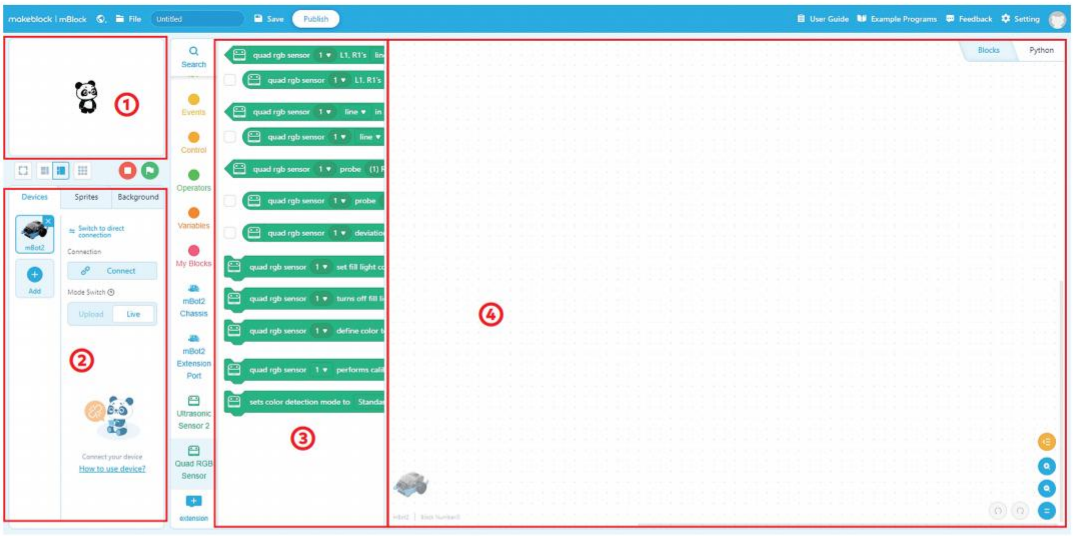


## Program mBot2 using mBlock 5

**1.** Go to www.mblock.cc and click "Download"


**2.** Download mBlock based on the device you use

## 3. Understand mBlock 5




① **Stage area:** Presents the effect of sprite movement.

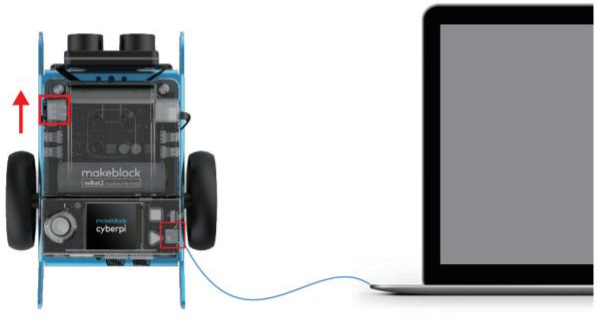
② **Device & Sprite & Background area:** Allows you to select characters, set background, and connect hardware devices.

③ **Block area:** Provides blocks for programming by classification and color.

④ **Script area:** Allows you to write your program. Drag and drop blocks to this area and arrange them in a certain order to form a program that can control devices and sprites.

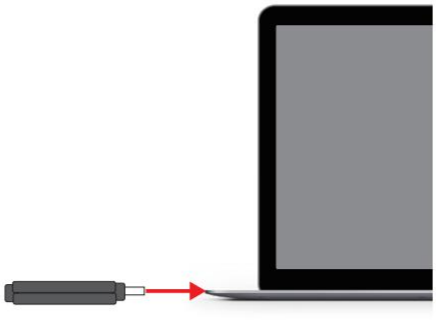
## 4. Connect mBot2 to a computer

- Using a USB cable




- Using a Bluetooth dongle

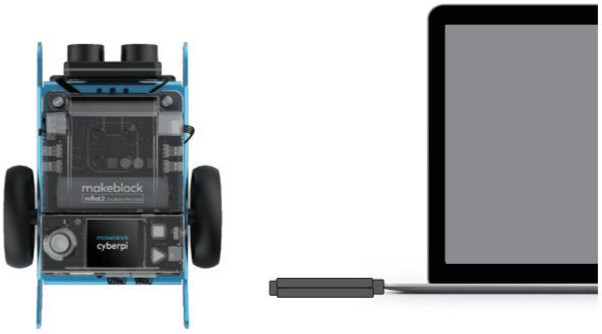
① Open the protection cap.



② Insert the Bluetooth dongle to a USB port of your computer.

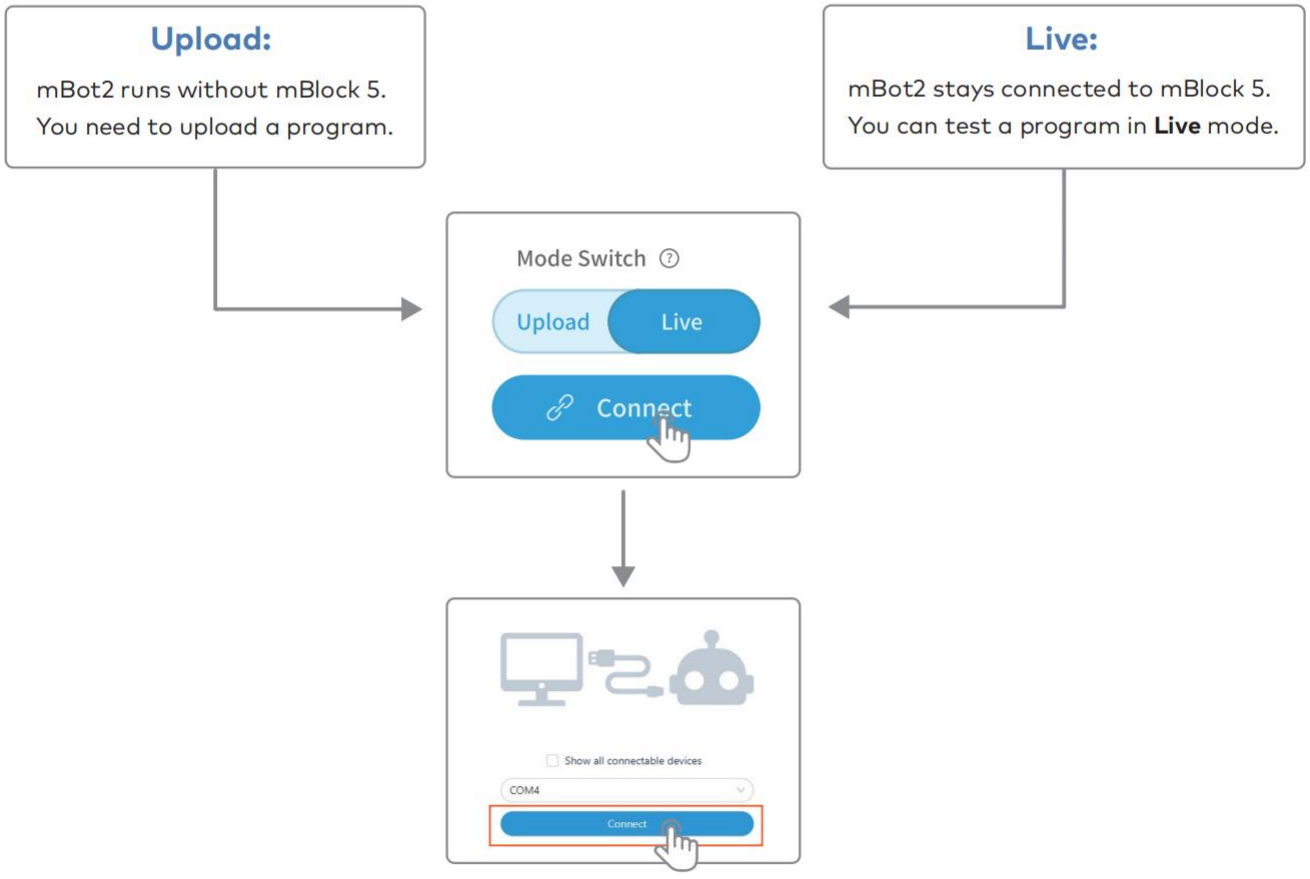

③ Turn on mBot2.

④ Press the Bluetooth button. The indicator on it blinks fast. The Bluetooth dongle enters the pairing state.

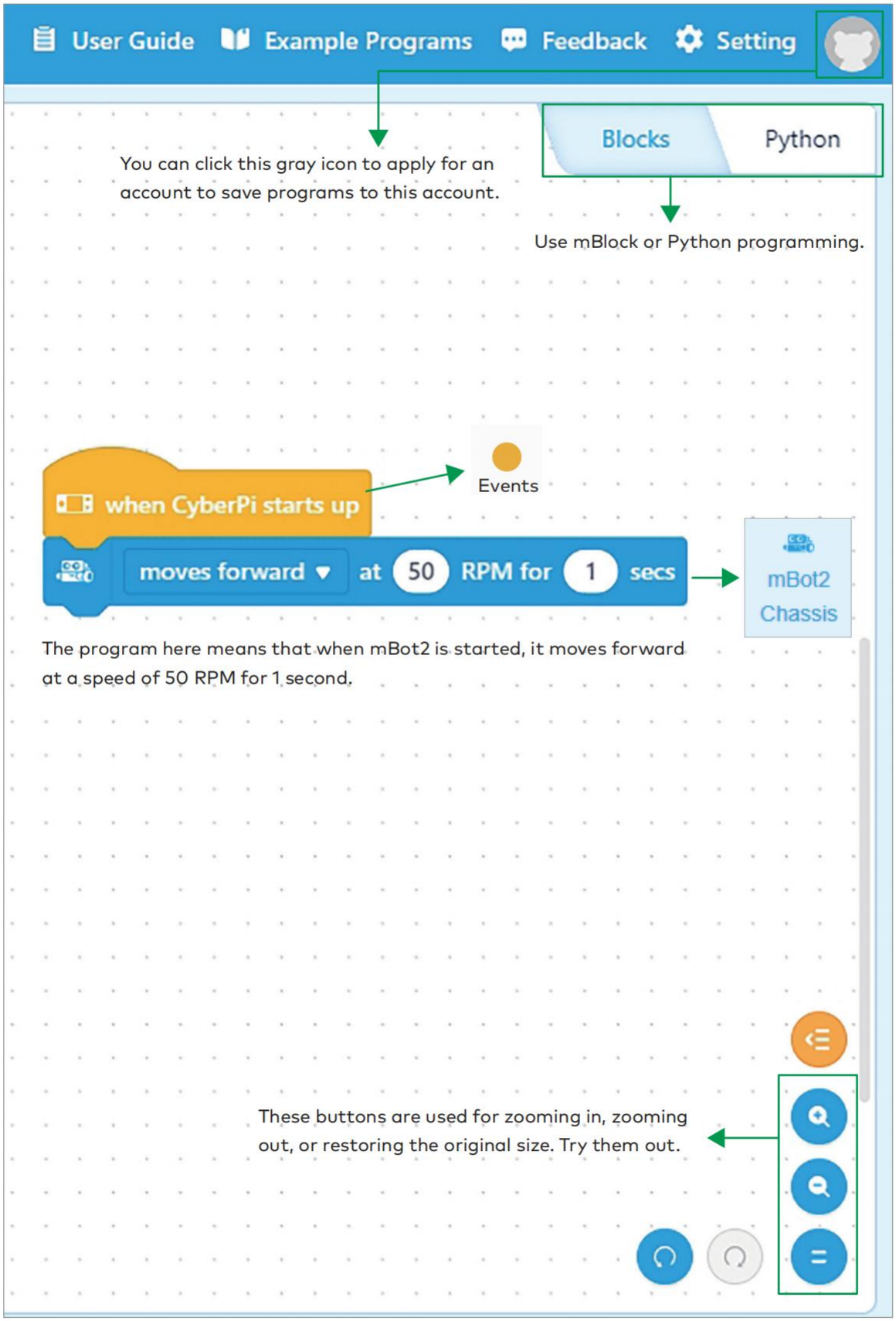



⑤ Place mBot2 close to the Bluetooth dongle and wait for the pairing to succeed. When the indicator on the Bluetooth dongle keeps on, the pairing is successful.

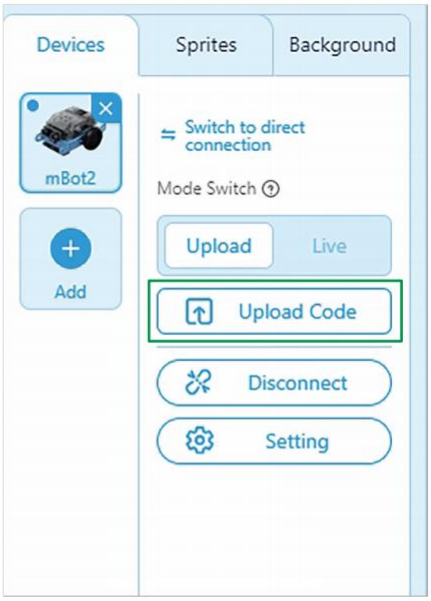
## **5.** Connect mBot2 on mBlock 5



**6.** Use mBlock 5 to write a simple program

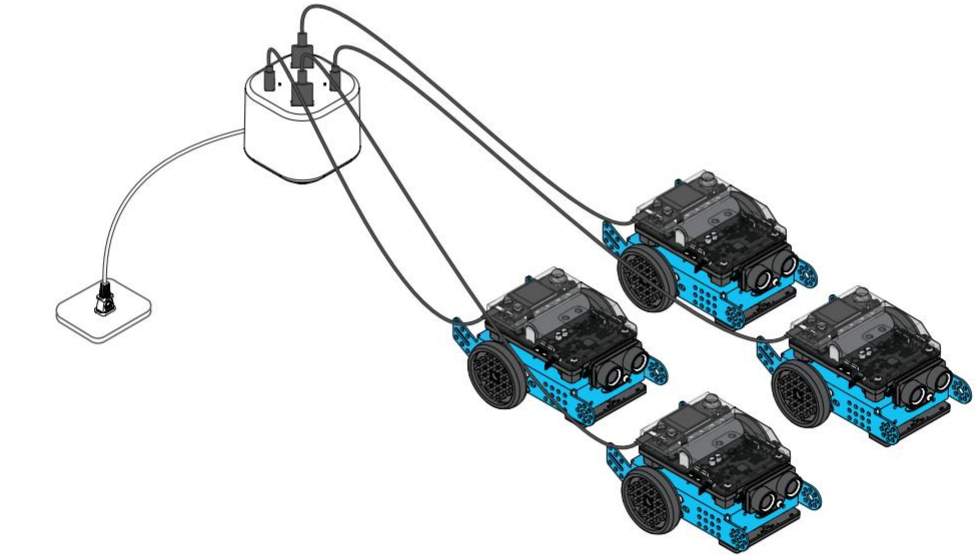


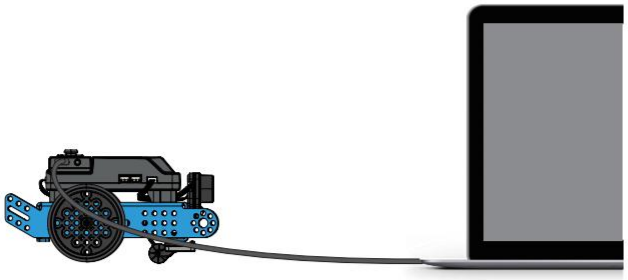
**7.** Upload the program to mBot2.
After finishing writing the program, click **Upload Code**.
The program is uploaded to mBot2 for execution.




## Charge mBot2

mBot2 Educational Kit provides a 5-port 10 A charger. To use the Type-C port in the middle, you need to use a Type-C to Type-C cable, which is not included in the kit. You can also charge mBot2 by connecting it to your computer using a USB cable.

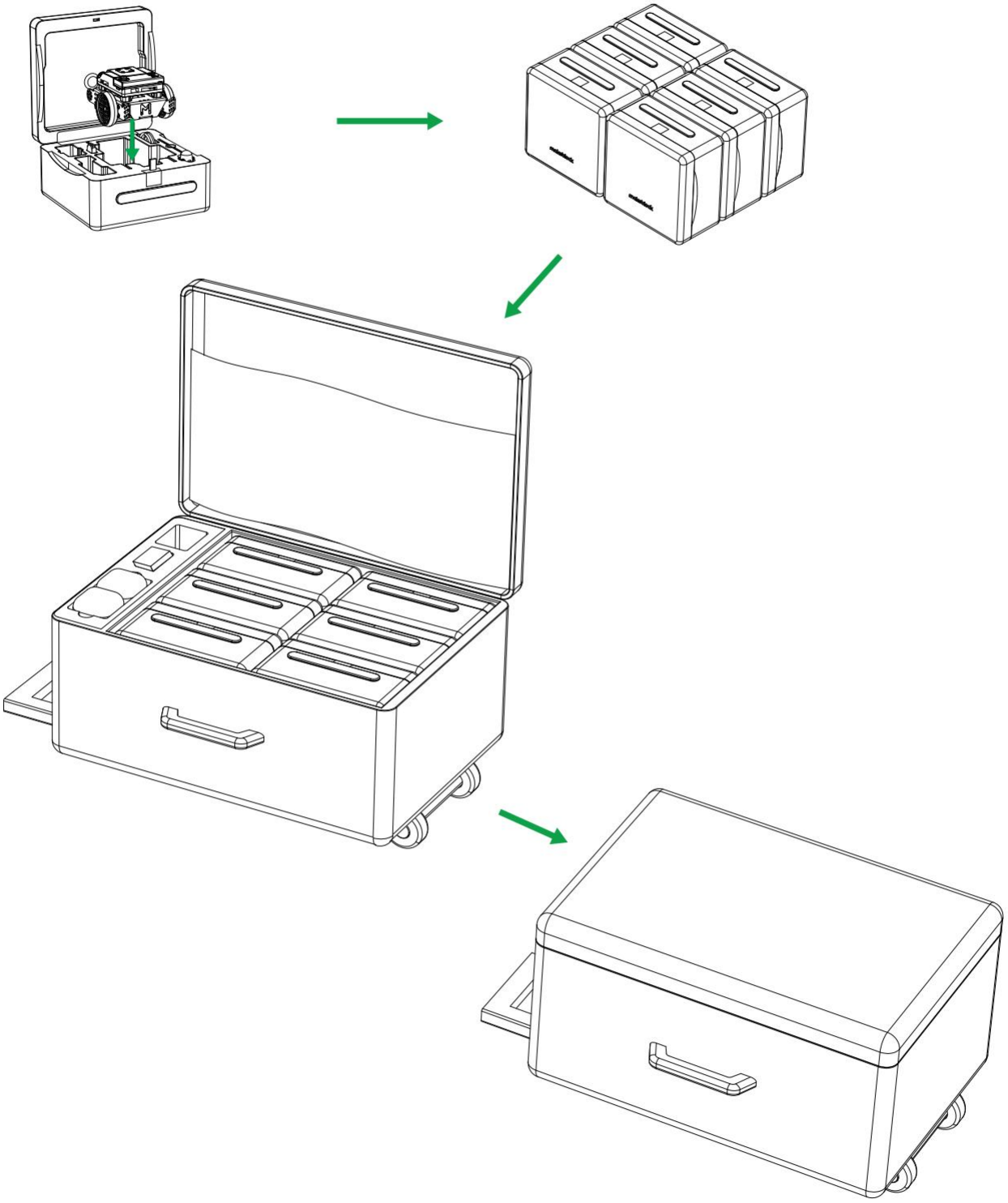




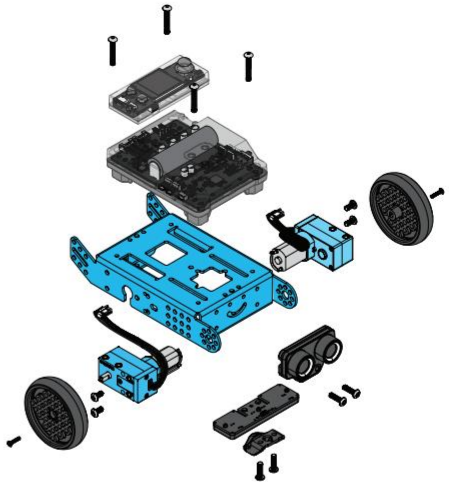
# Store mBot2 and the kit

## mBot2 storage

Each mBot2 is individually packed in a storage box. You can also put an assembled mBot2 and assembly tools in the box for storage.



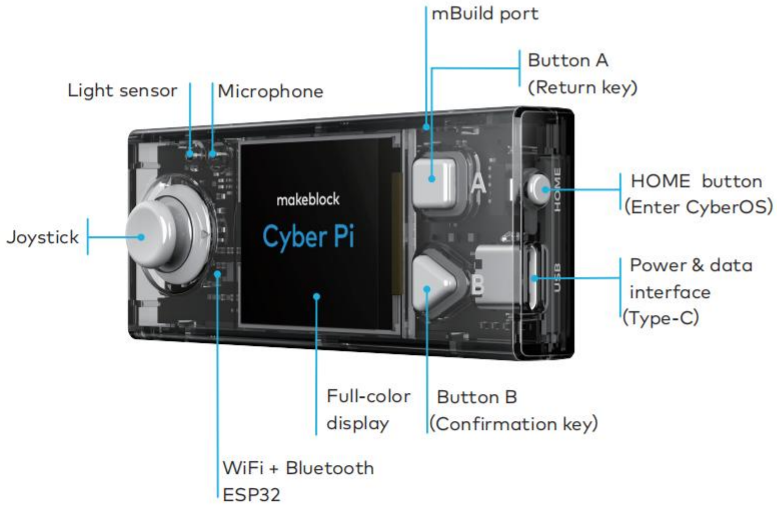
# 3. About mBot2



mBot2 uses CyberPi as its main control board and is equipped with multiple electronic modules, including mBot2 shield, ultrasonic sensor 2, quad RGB sensor, and encoder motors.

## CyberPi

CyberPi is a main control board developed independently by Makeblock. With the compact structure and built-in ports, it can be easily extended. It supports mBlock 5 and mBlock-Python Editor, and applies to multiple education scenarios including large-class teaching, community teaching, and online/offline education & training. It covers teaching fields including coding, makers, and robots, and thus it can meet diversified education needs, such as AI, IoT, data science, and UI design.





CyberOS is the operating system that runs on CyberPi. It is the core component that supports all the functions of CyberPi.

## Specifications

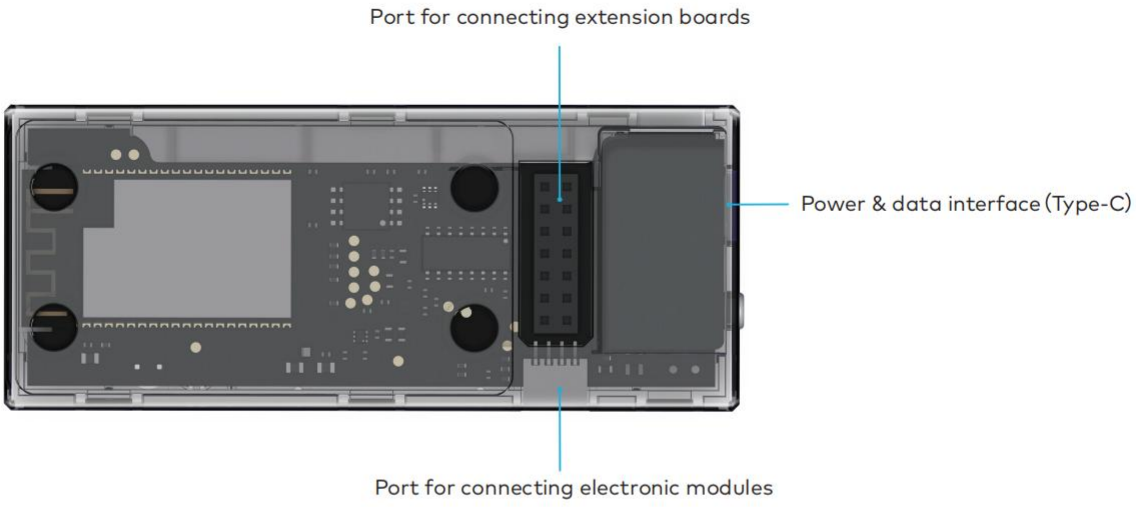
| | | |
|---|---|---|
| Name | | CyberPi |
| Chip | | ESP32-WROVER-B |
| Processor | Main processor | Xtensa® 32-bit LX6 dual-core |
| | Clock frequency | 240 MHz |
| Onboard memory | ROM | 448 KB |
| | SRAM | 520 KB |
| Extended memory | SPI Flash | 8 MB |
| | PSRAM | 8 MB |
| Operating system | | CyberOS, developed independently by Makeblock |
| Wireless communication | | Wi-Fi |
| | | Dual-mode Bluetooth |
| Ports | | Micro USB port (Type-C)<br>Port for connecting extension boards<br>Port for connecting electronic modules (serial communication) |
| Hardware version | | V1.0 |
| Dimensions | | 84 mm × 35 mm × 13 mm (height × width × depth) |
| Weight | | 36 g |

## Features

- Full-color display, providing user-friendly UIs for human-machine interaction
- CyberOS system, allowing you to execute the predefined programs, set the system language, and update the system through the onboard joystick and buttons
- One Micro USB port (Type-C) for connecting to PCs for power supply and communication
- One electronic module port for connecting electronic modules
- One extension board port for connecting to extension boards
- Multiple onboard sensors, such as light sensor and gyroscope, which provides multiple types of data output
- Five LEDs, allowing you to present abundant light effects
- One onboard Bluetooth and Wi-Fi module, enabling wireless communication
- Supporting mBlock 5 programming, which is intended for users of all ages, including those without any programming experience
- Supporting Python programming, for which the cyberpi library is provided

## Port description

CyberPi is equipped with a Type-C USB port, an electronic module port, and an extension board port, which allow it to easily and quickly connect to various types of electronic modules and extension boards.




**Micro USB port (Type-C)**

The Micro USB port allows CyberPi to connect to various types of computer devices for power supply and communication.

> ! When charging CyberPi, ensure that the voltage is lower than 6 V. The hardware of CyberPi may be damaged if the voltage is equal to or higher than 6 V.

**Port for connecting to extension boards**

You can easily connect CyberPi to an extension board through the extension board port. Currently, the extension boards Pocket shield (not included in the kit) and mBot2 shield are available for CyberPi.
Pocket shield is equipped with a built-in rechargeable battery that can supply power for CyberPi and provides 2-pin and 3-pin interfaces that can be used to connect servos, LED strips, and motors, which significantly improves the extensibility of CyberPi.

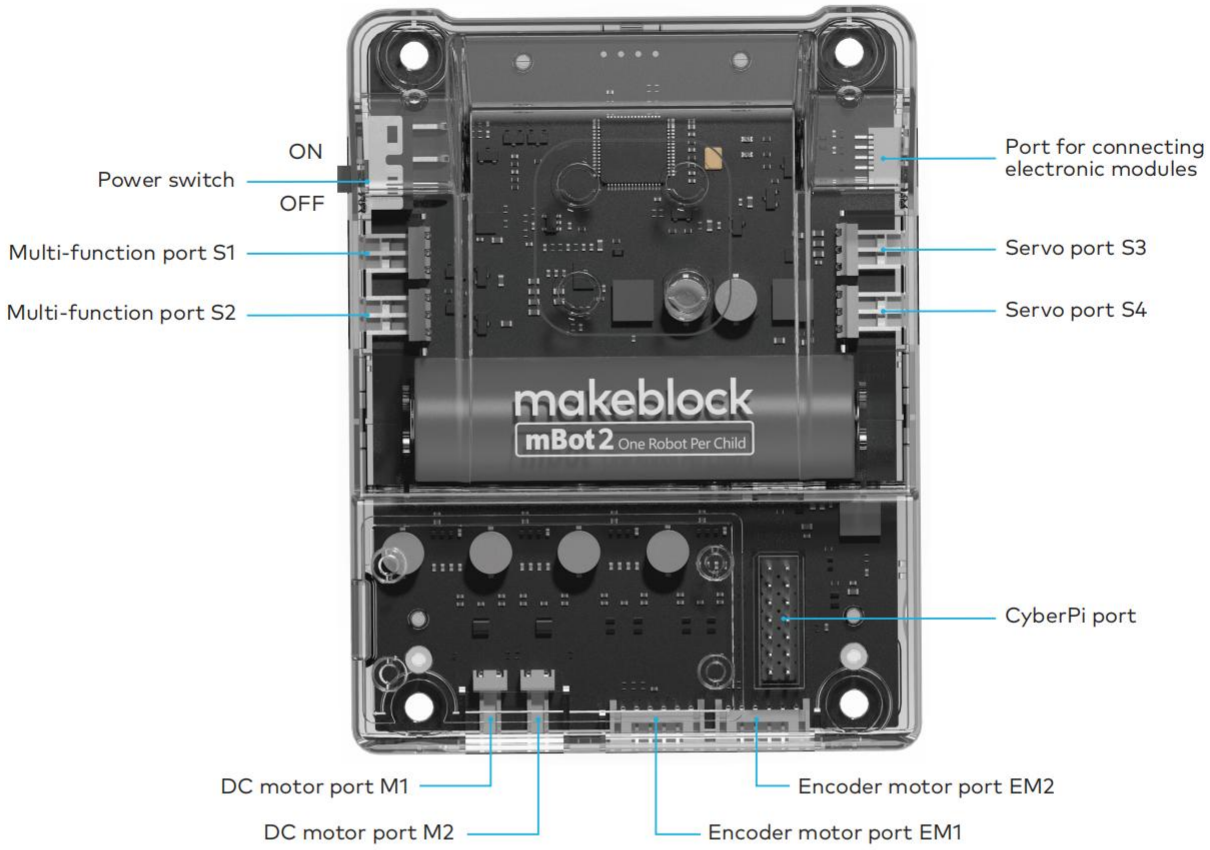
**Port for connecting electronic modules**

You can connect CyberPi to multiple electronic modules in series through the electronic module port.
CyberPi can intelligently identify the addresses of the modules, which simplifies your programming.
You don't have to set the information about the addresses of the modules when you add or remove a module.

For more details about CyberPi, visit this link: support.makeblock.com/hc/en-us/sections/6973045213719.

# mBot2 shield

mBot2 shield is equipped with a built-in rechargeable lithium-ion battery that can supply power for CyberPi. With the multi-function, servo, and motor ports, it can drive motors, servos, and LED strips.




## Features

- Built-in rechargeable lithium-ion battery, used to supply power for CyberPi
- Two multi-function ports, used to connect and drive not only servos but also LED strips
- Two servo ports, used to connect and drive servos
- Two DC motor ports, used to connect and drive DC motors
- Two encoder motor ports, used to connect and drive encoder motors
- One CyberPi port, allowing you to easily connect mBot2 to CyberPi

## Specifications

| Specifications | Description |
|---|---|
| Microprocessor | GD32F403 |
| Battery | 3.7 V, 2500 mAh<br>Packaging: Bundle (battery included)<br>Battery type: Lithium-ion battery<br>Battery weight: 44.6 g<br>Watt hours per battery (wh): 9.25 wh<br>Lithium content: 1.07 g<br>Lithium battery voltage: 3.7 V |
| Input voltage and current | 5 V, 2000 mA (fast charging) |
| | 5 V, 500 mA (charging in operation) |
| Output voltage and current | 5 V, 6 A |
| Battery life | 3–6 hours (in general application scenarios, just for reference) |
| Charging time | 80 minutes (in fast charging mode) |
| Battery endurance | The capacity of the battery is remained in 70% or higher after it is charged and used for 800 times (at 20±5℃, 0.2 C discharging). |
| Communication mode | Serial communication: between the main control board and extension board<br>Digital signals: at the digital servo port<br>PWM signals: at the DC motor port |
| Hardware version | V1.0 |

## Note:

- Self-discharge occurs in the lithium-ion battery. If you store mBot2 shield with a battery voltage lower than 3.6 V for a long time, the battery will be over-discharged and its internal structure may be damaged, which reduces the endurance of the battery. Therefore, to store mBot2 shield for a long time while keeping the battery intact, you need to charge the battery once every three to six months to 3.8–3.9 V (the best voltage for storage is 3.85 V), which allows the discharge depth of 40% to 60%.
- Store mBot2 shield at 4℃ to 35℃ in a dry place or keep it away from moisture through packaging.
- Keep it away from heat or direct sunlight.

# Ultrasonic sensor 2

The ultrasonic sensor 2 can be used to detect the distance between an obstacle and it. The transmitter on the left transmits ultrasonic waves, and the receiver on the right receives the ultrasonic waves reflected.


## Specifications

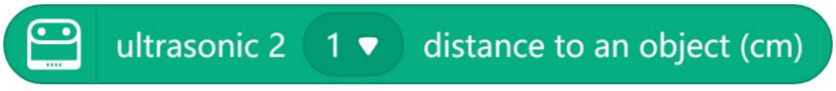
- Output range: 5–300 cm (The output value is 300 when the distance detected is out of the output range.)
- Output value error: ±5%

## Working principles

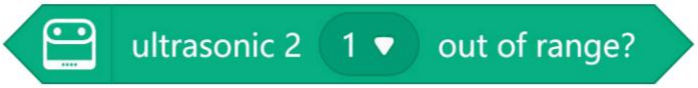
Human beings can hear sounds of 20 to 20,000 Hz. Sound waves with frequencies higher than 20,000 Hz are called ultrasonic waves. The sound waves are reflected back by the obstacles they encounter and are received by the receiver of the ultrasonic sensor. Based on the time between transmission and receiving, the distances between the ultrasonic sensor and the obstacles can be calculated.

## Distance detection and determination

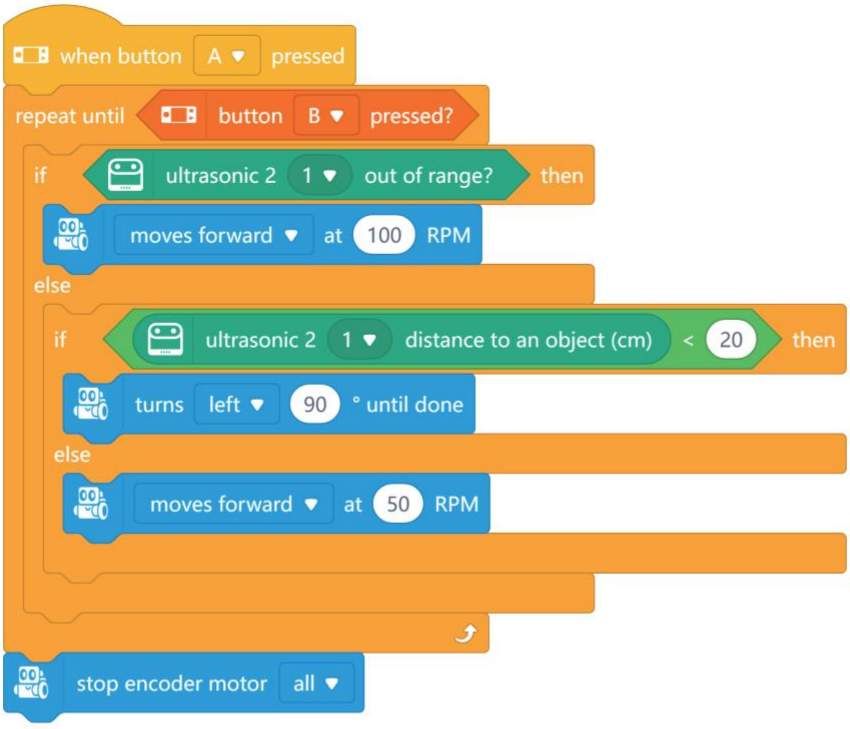
To obtain the distance between the obstacle and the ultrasonic sensor 2, use:




To determine whether an obstacle is found within the output range of 5 cm to 300 cm, use:



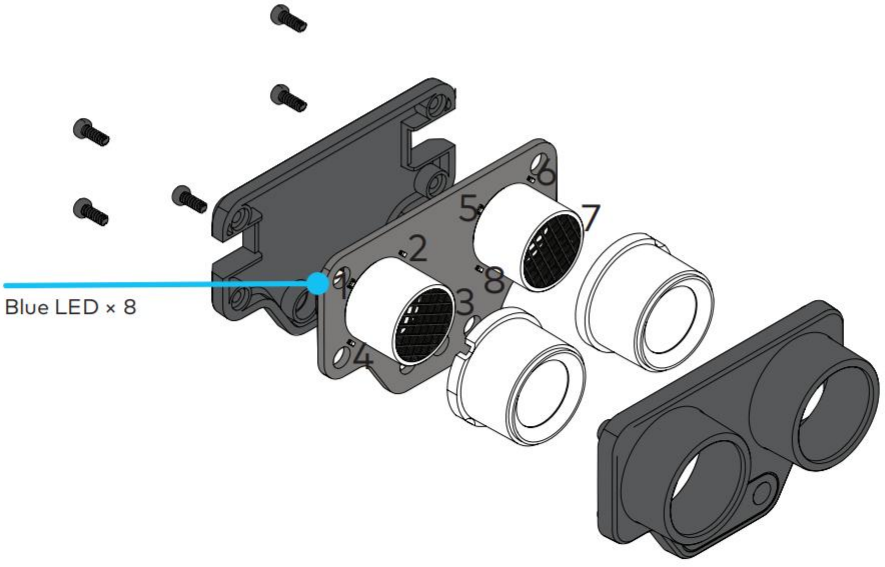
Generally, the distance detection and determination functions are used to implement obstacle avoidance.
For example:



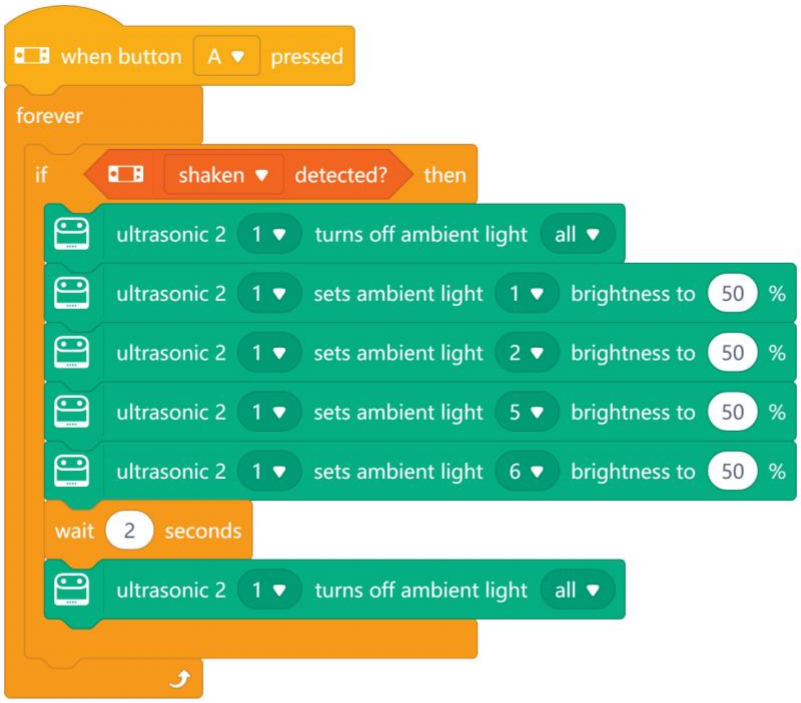

When no obstacle is within the detection range of the ultrasonic sensor 2, mBot2 moves forward at the speed of 100 RPM; when an obstacle is detected, it moves forward at the speed of 50 RPM; and when the distance between the obstacle and it is shorter than 20 cm, it turns left.

## Emotion and information expression

The ultrasonic sensor 2 is equipped with eight blue LEDs. Through programming their brightness, you can use the ultrasonic sensor 2 to express various types of emotions and information.



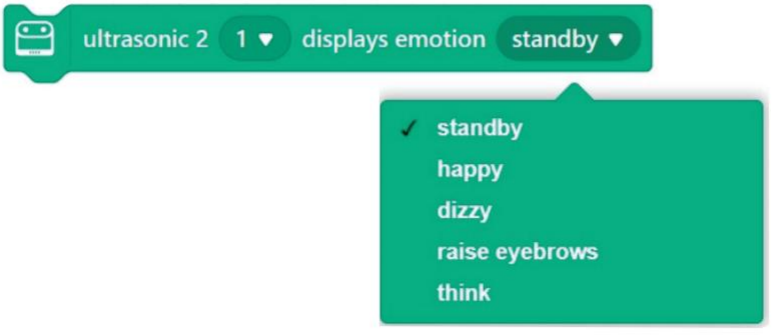
## Example 1:



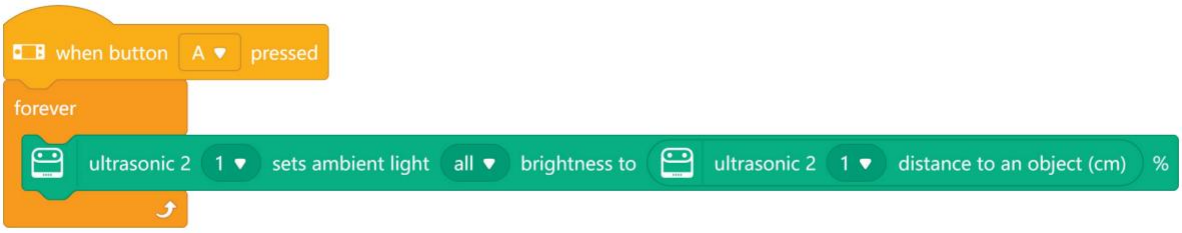

After you press button A, the ultrasonic sensor 2 expresses the emotion of happiness by "smiling" when you shake it.

⚠ Note: In this example, "smiling" is performed by lighting up the blue LEDs in positions 1, 2, 5, and 6 and turning off those in positions 3, 4, 7, and 8

Alternatively, you can use the emotion block provided on mBlock 5 to express an emotion:



### Example 2:




After you press button A, place an obstacle close to the ultrasonic sensor 2, and then move the obstacle slowly away from it.

You can see that the blue LEDs become brighter as you move the obstacle. A higher brightness indicates a larger distance.

## Quad RGB Sensor


The quad RGB sensor uses visible light as fill lights, which significantly reduces the interference of ambient light. In addition, it provides the function for recognizing colors. The new ambient light calibration function also reduces the interference of the ambient light in line following.

| | **Quad RGB sensor** |
|---|---|
| Plastic casing to improve the durability and quality | Yes |
| Line-following sensor | 4 |
| Color sensor | 4 (also serve as line-following sensors) |
| Light sensor | 4 (also serve as line-following sensors) |
| Fill light | Visible light |
| Ambient light calibration to significantly reduce the interference of ambient light | Yes |

## Specifications

Detection range: 5–15 mm from the object to be detected

## Button description

- Double-press: When the button is double-pressed, the quad RGB sensor starts to learn the background and line for line following.

  Place the light sensors on the background of the line-following track map and double-press the button. When you see the LEDs indicating the line-following state blink quickly, sway the sensors from side to side above the background and line until the LEDs stop blinking. It takes about 2.5 seconds. The parameter values obtained are automatically stored. If the learning fails, the LEDs blink slowly, and you need to start the learning again.

- Long-press: When the button is long-pressed, the quad RGB sensor switches the color of the fill lights. Generally, you don't need to change the color. The color is set automatically after the learning is complete.

For more details about the quad RGB sensor, visit support.makeblock.com/hc/en-us/articles/24279693845527.

# 4. mBlock 5 guide

You can program mBot2 using mBlock 5.
For details about how to program using mBlock 5, visit support.makeblock.com/hc/en-us/sections/360001829013
After you have mastered the use of mBlock 5, you can try the following programming projects on mBot2. You can also introduce these programming projects to your classes to make teaching fun.


Note: Besides the programming projects, you can also use the three line-following maps for other teaching projects.

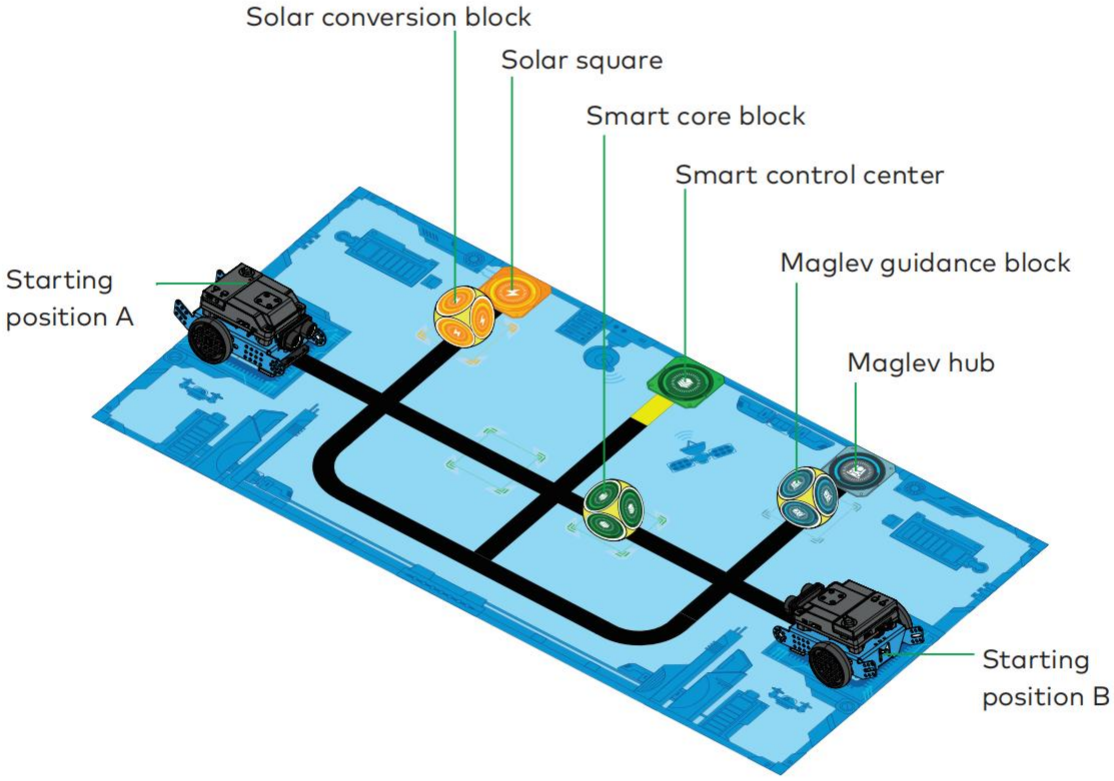
# 5. Classroom activities

In this section, you will lead the students to explore a world of technology and innovation through the Future City program. By programming mBot2, students are going to accomplish a series of tasks with great fun.

This Future City program can help students:

- enhance programming skills by programming mBot2 using graphical blocks and leveraging sensor and control features.
- foster problem-solving and logical thinking by tackling problems encountered in operating mBot2.
- cultivate team collaboration by cooperating with others to accomplish the tasks.

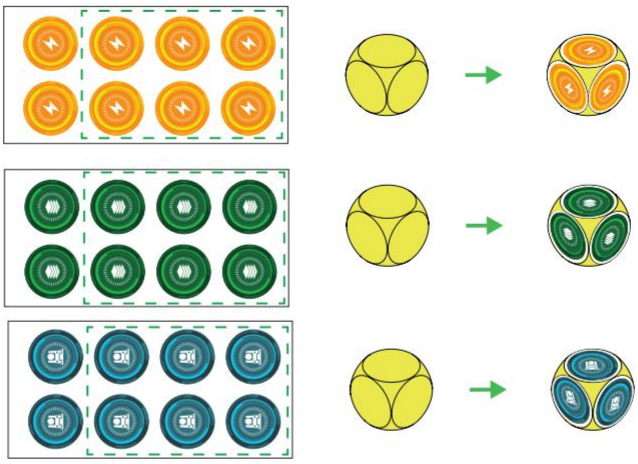
## Introduction



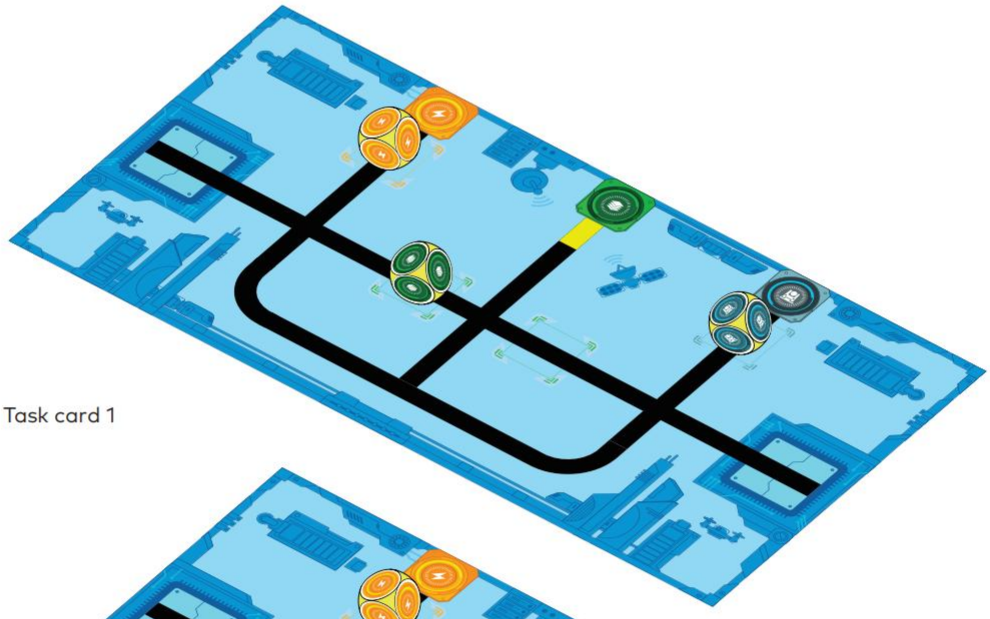

- Solar conversion block: converts solar energy into clean energy that continuously powers the city.
- Smart core block: functions as the brain of the Future City and controls all systems within the city.
- Maglev guidance block: provides transportation services to the Future City.
- Solar square: provides energy to the solar conversion block.
- Smart control center: provides energy to the smart core block.
- Maglev hub: provides energy to the maglev guidance block.
- Starting positions A and B: The starting positions of the two mBot2s.

## Preparations

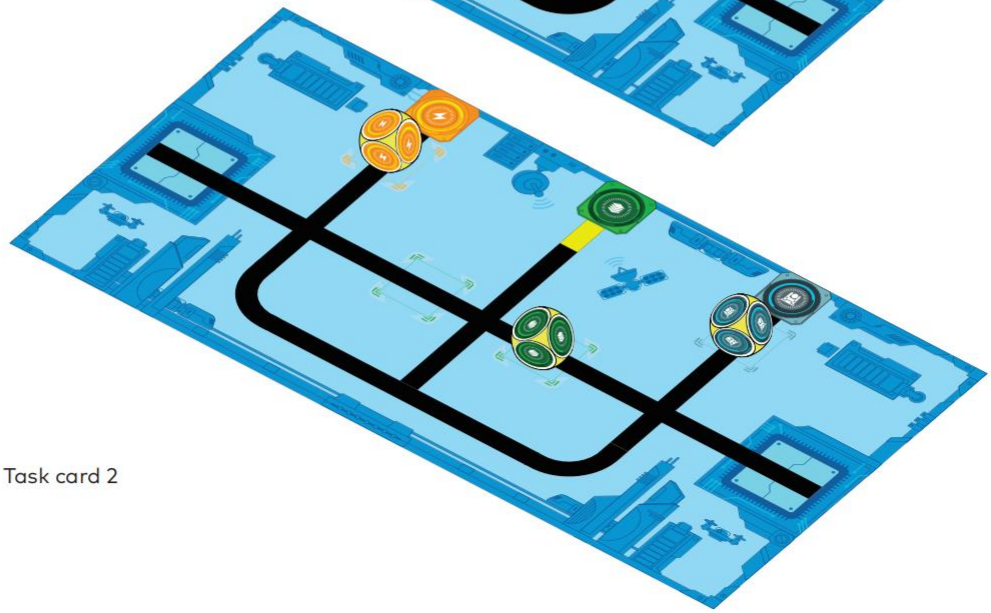
**1.** Attach energy stickers to three yellow blocks, with six stickers of the same color attached to the six sides of each block. Keep extra energy stickers properly for future use.



**2.** Place the three energy blocks on the map.



Task card 1



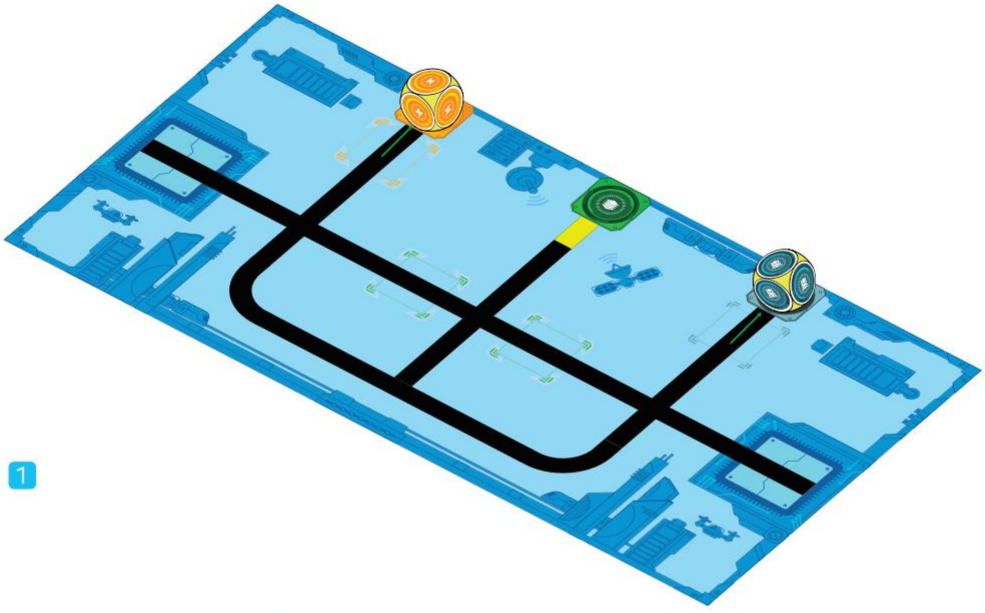
Task card 2

# Task description

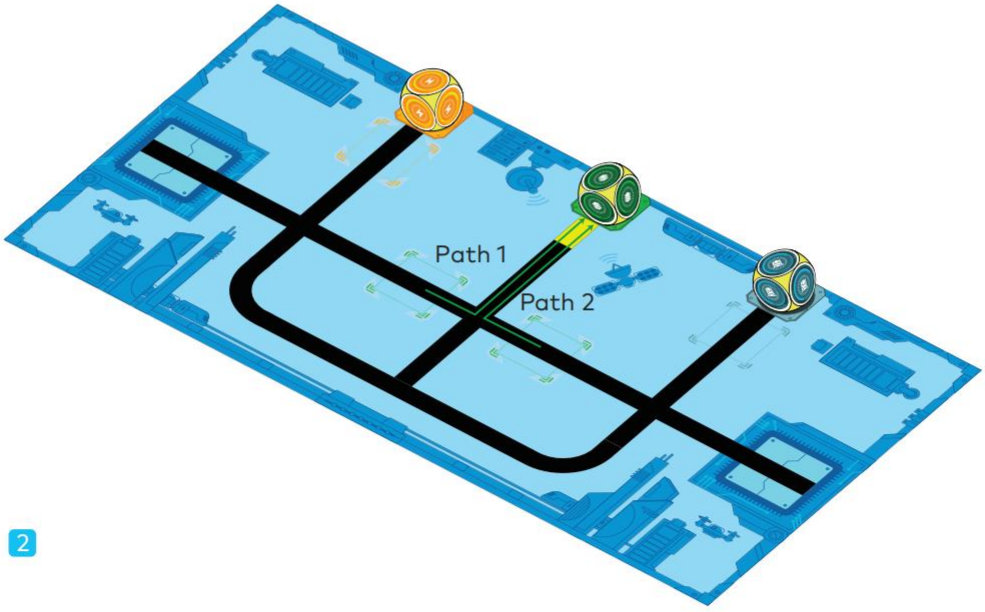
## Objective

The objective is to program and control two mBot2s to accomplish the following tasks using line-following, ultrasonic, and color recognition technologies:

1. Push the solar conversion block to the solar square and the maglev guidance block to the maglev hub.
2. Push the smart core block to the smart control center.



1




2

## Instructions

- Explain the activity tasks to the students, including the task principles and winning conditions.
- Group students into teams of two. Ask one member to draw a task card.
- Instruct the students to write programs based on the task cards they have drawn.
- Ask the students to place two mBot2s in the starting area and make a signal after they finish placing mBot2s.
- Give the command of "Start" and start the timer. Two students in one team push the start button of their respective mBot2.
- End the timer when the task is completed.
- Check the score and time used by all teams, and announce the winner.

## Activity principles

During the activity:

- Students are not allowed to touch the energy blocks throughout the activity. However, if mBot2 breaks down, pauses, or drives out of the map, the team members can manually move mBot2 to the starting position and try again. However, the timing continues.
- In the collaborative tasks, the two mBot2s should not touch each other.
- Students score points only when the smart core block, solar conversion block, or magnetic guidance block touches its designated destination.

After the activity:

- At the end of each round of the activity, the students need to put the energy blocks back to their original position.

## Score and winning conditions

| Activity | Illustration | Score |
|---|---|---|
| The solar conversion block is pushed to the solar square. |  | 100 |
| The maglev guidance block is pushed to the maglev hub. |  | 100 |
| The smart core block is pushed to the smart control center. |  | 200 |

Winning: At the end of the task, the team with the highest score wins. In the case of a tie, the team that takes the least amount of time wins.

# Carry out the tasks

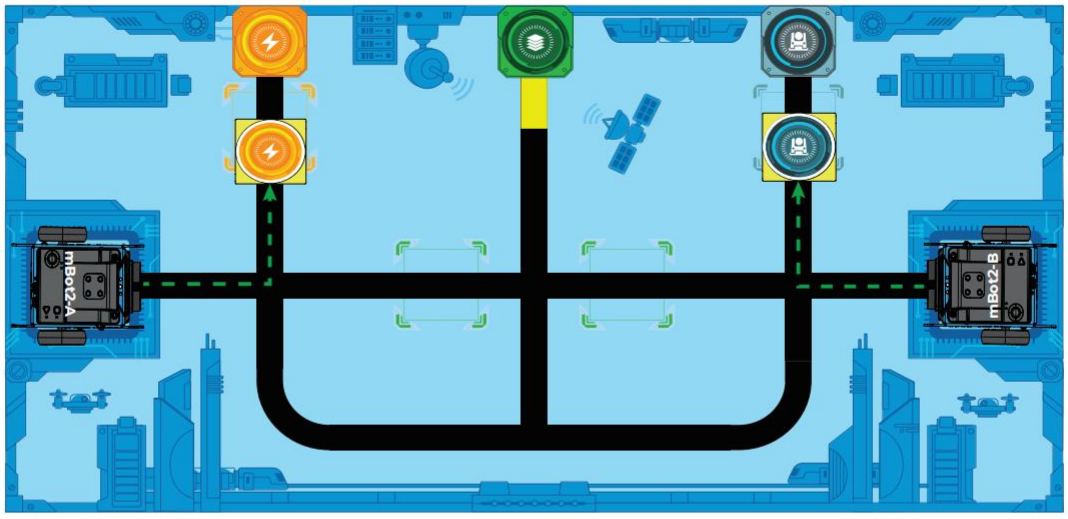
## Task card 1 --Independent tasks

**mBot2-A**

Push the solar conversion block to the solar square.

**mBot2-B**

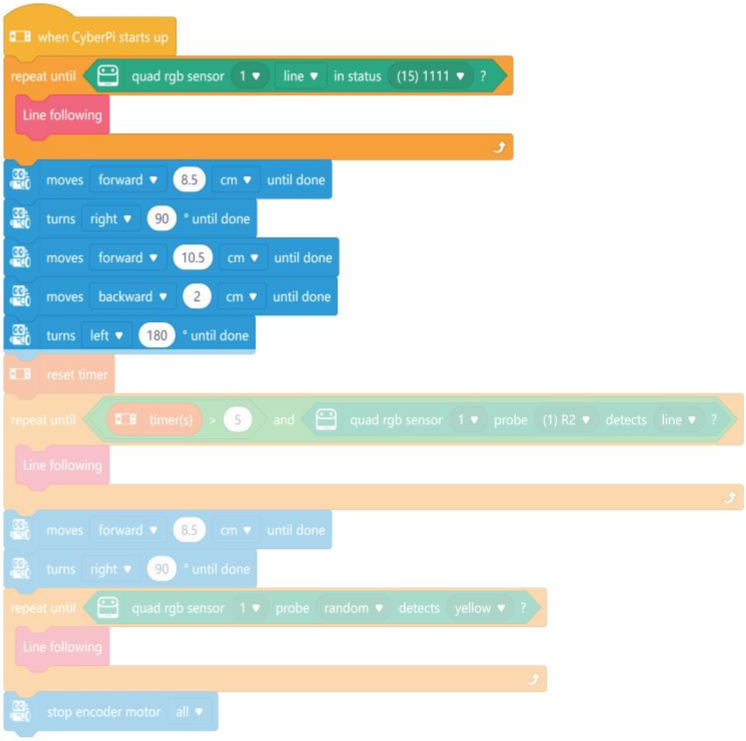
Push the maglev guidance block to the maglev hub.




## Programs

**mBot2-A**

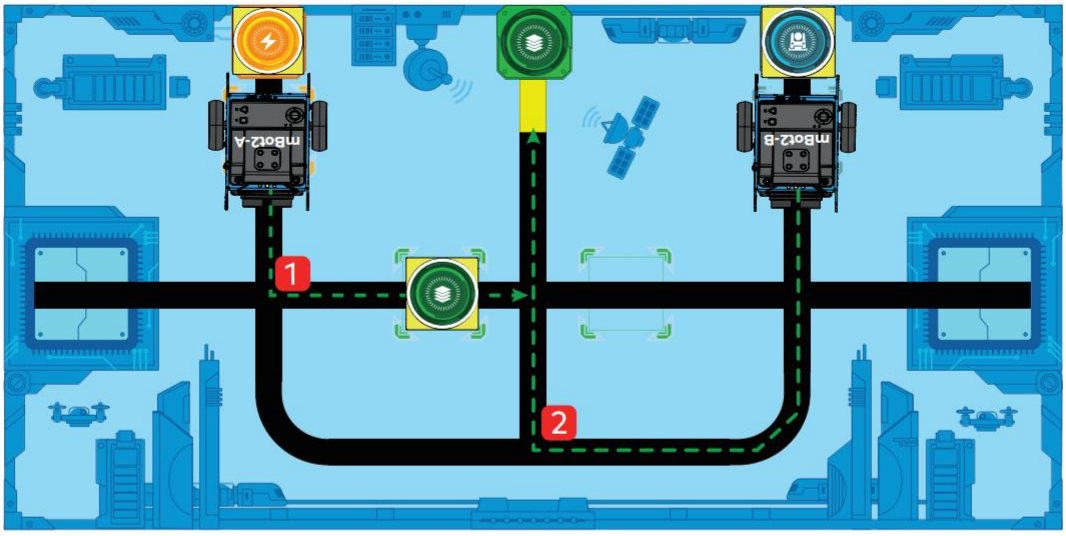

**mBot2-B**




You can change the distance unit from cm to inch. Note that the distance value must be changed accordingly. The actual distance and angle of mBot2s vary with the mBot2 speed and map settings. The values used in this guide are for reference only.

## Task card 1 -- Collaborative task

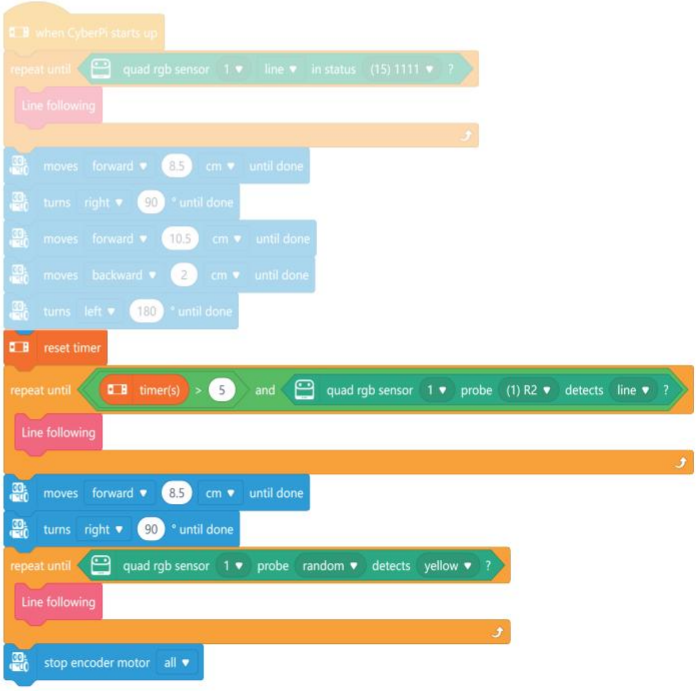
mBot2-A and mBot2-B cooperate to push the smart core block on the left to the smart control center.




## Programs

### mBot2-A


### mBot2-B



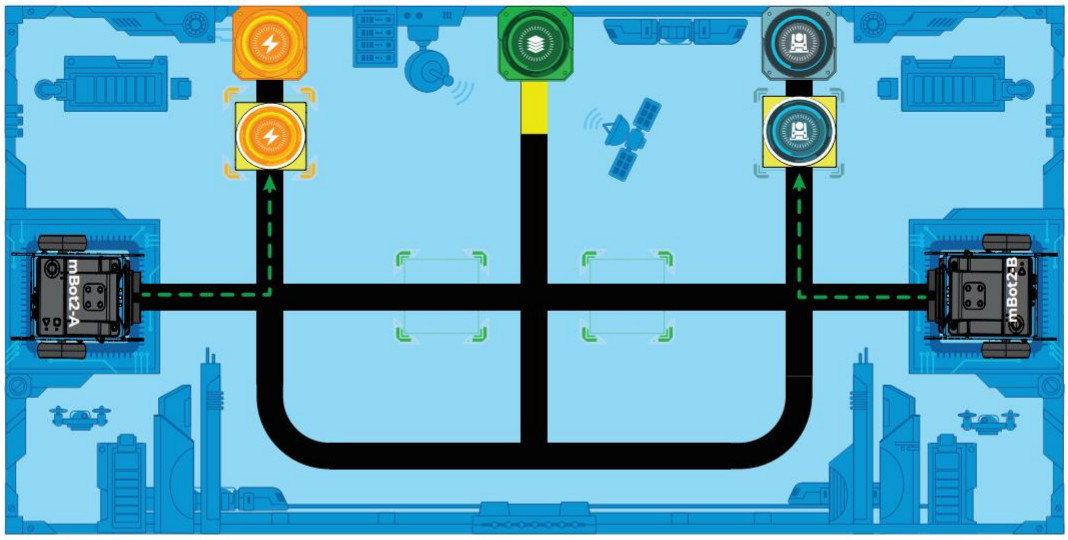
## Task card 2 -- Independent tasks

**mBot2-A**

Push the solar conversion block to the solar square.

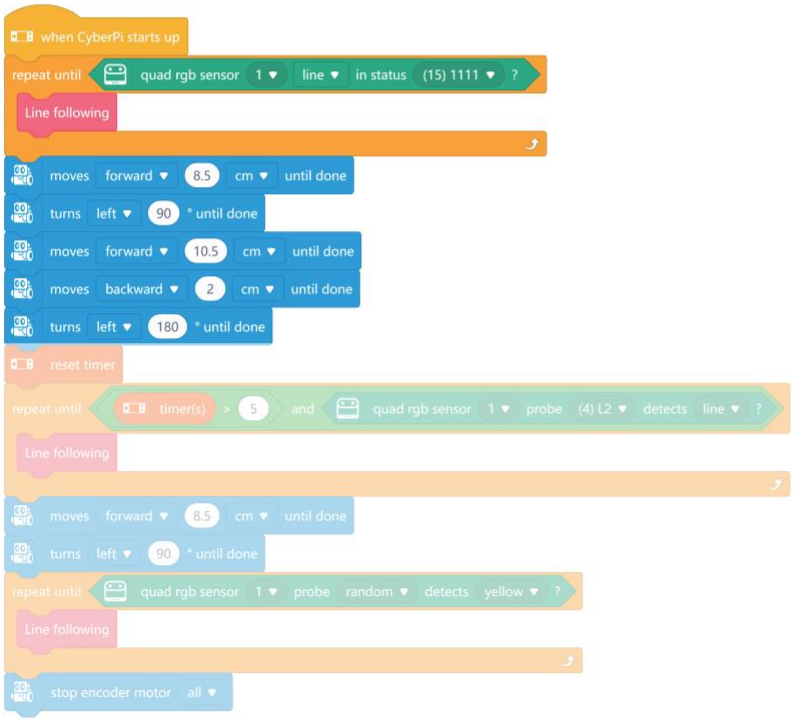
**mBot2-B**

Push the maglev guidance block to the maglev hub.

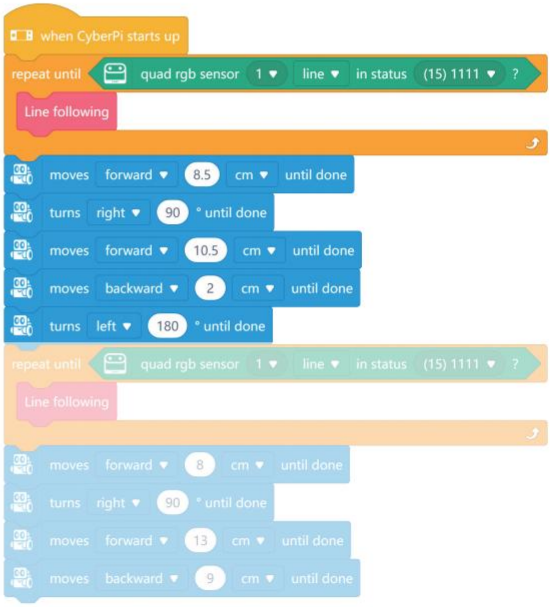



## Programs

### mBot2-A

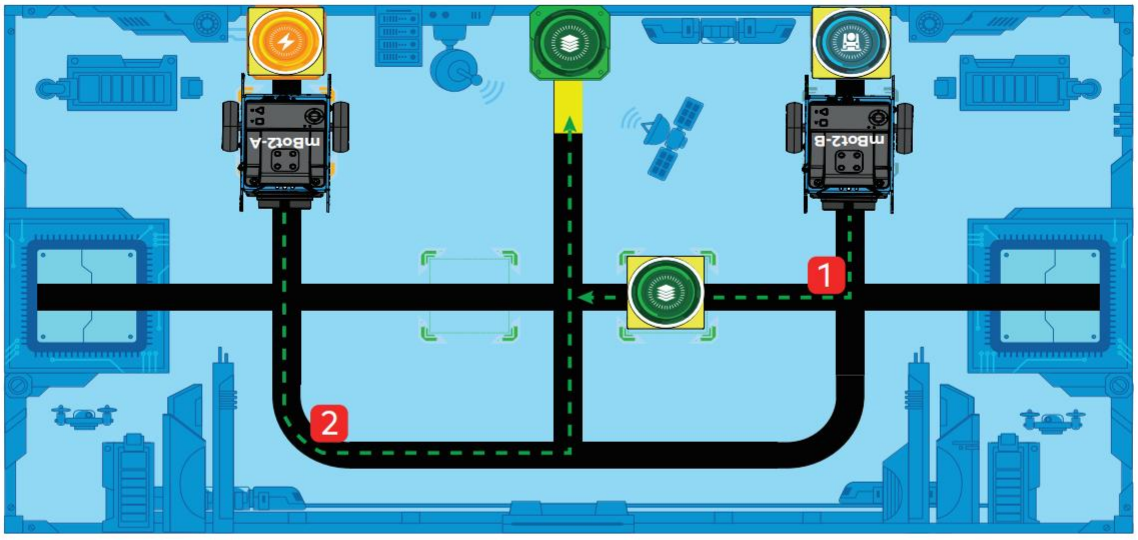



### mBot2-B



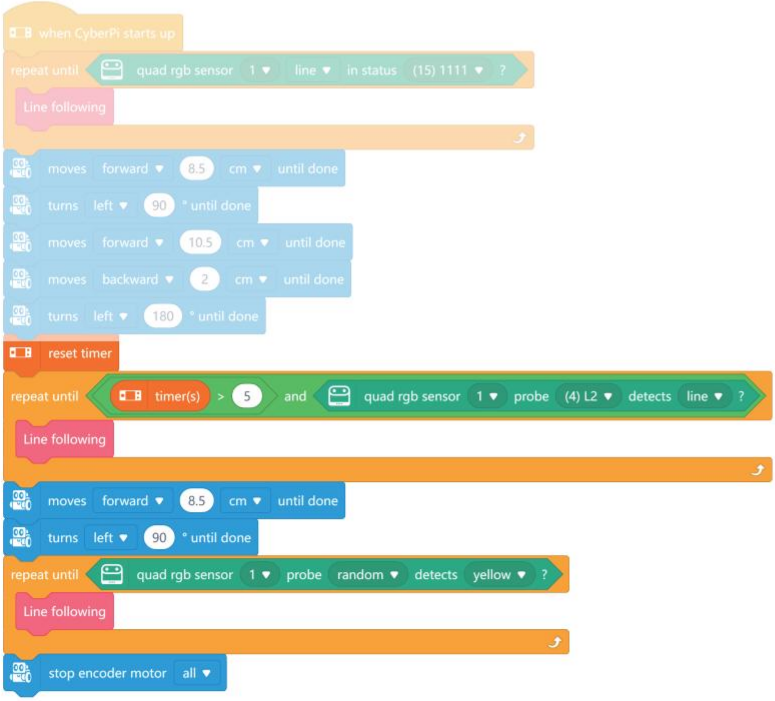
## Task card 2 --Collaborative task

mBot2-A and mBot2-B cooperate to push the smart core block on the right to the smart control center.




## Programs

### mBot2-A




### mBot2-B

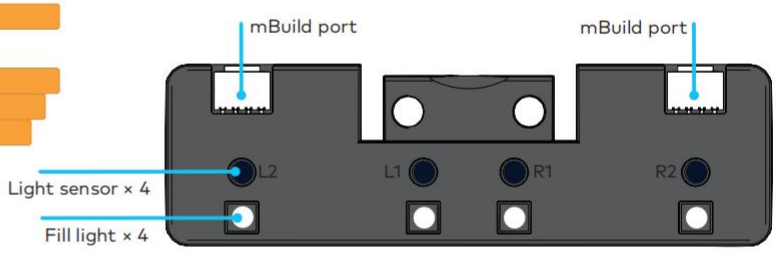
# Programming description

## Line following program

The line following feature is required throughout the two task cards. Therefore, we introduce a custom block "line following" to simplify the task. You can also create your own blocks by following the instructions in this link: www.yuque.com/makeblock-help-center-en/mblock-5/my-blocks

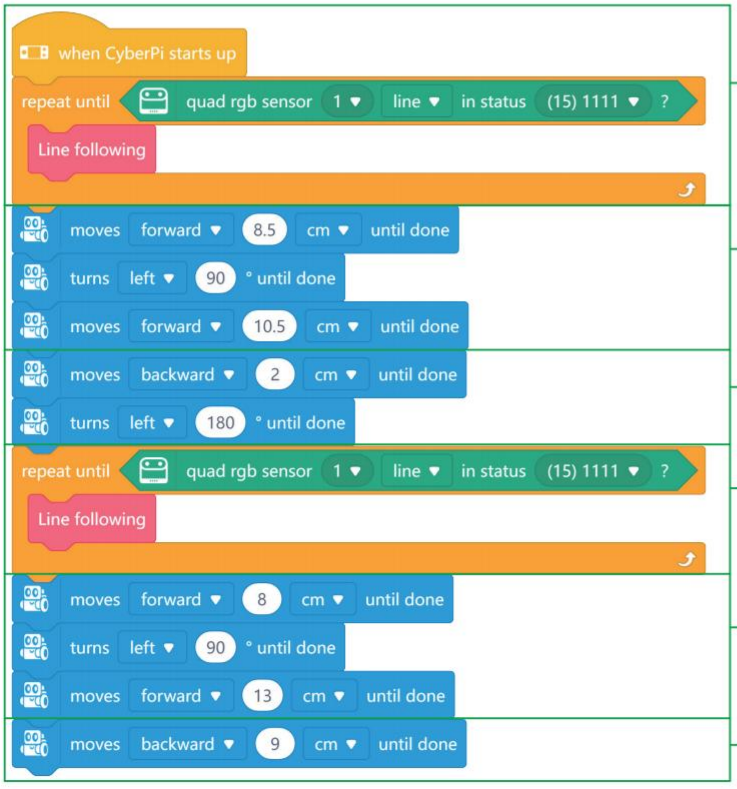

In the line-following program, value **1** of the quad RGB sensor indicates a black line detected while value **0** indicates a background detected. By checking whether the two probes "R1" and "L1" of the quad RGB sensor detect the black line or not, you can determine whether mBot2 is deviating from the black line.

For example, when **10** or **01** is returned for "R1" and "L1" probes, mBot2 deviates from the black line, and adjusts its position by turning left or right.




## Programs for task card 1

**mBot2-A**




Starts line following after being turned on and stops line following after detecting an intersection (quad RGB sensor value: **1111**).

**1.** Moves forward to accurately arrive at the intersection.
**2.** Turns 90° left at the intersection.
**3.** Moves forward, pushing the solar conversion block to the solar square.

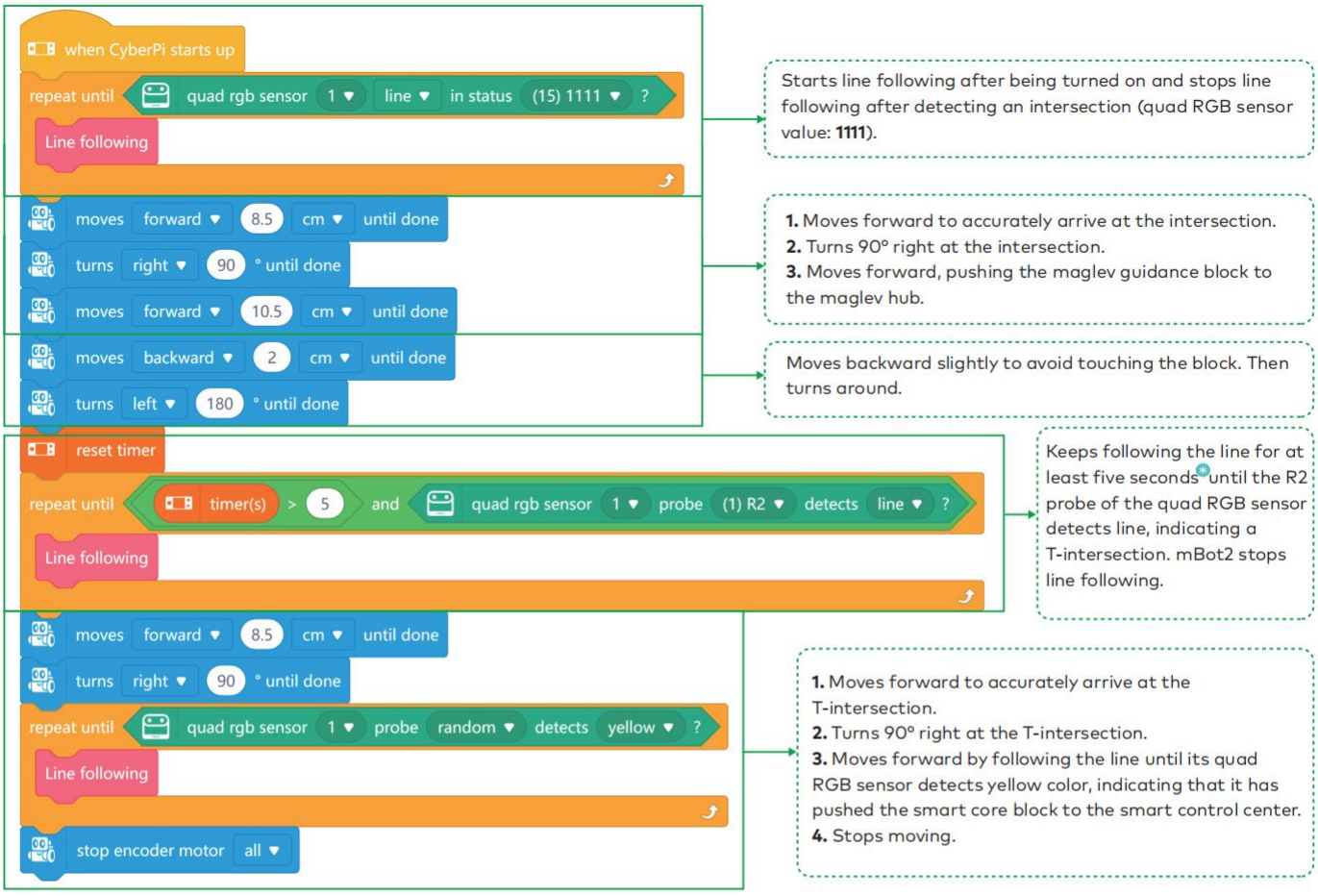
Moves backward slightly to avoid touching the block. Then turns around.

Keeps following the line until detecting an intersection (quad RGB sensor value: **1111**).

**1.** Moves forward to accurately arrive at the intersection.
**2.** Turns 90° left at the intersection.
**3.** Moves forward, pushing the smart core block to the center intersection of the map.

Moves backward for a distance to avoid colliding with mBot2-B.

**mBot2-B**



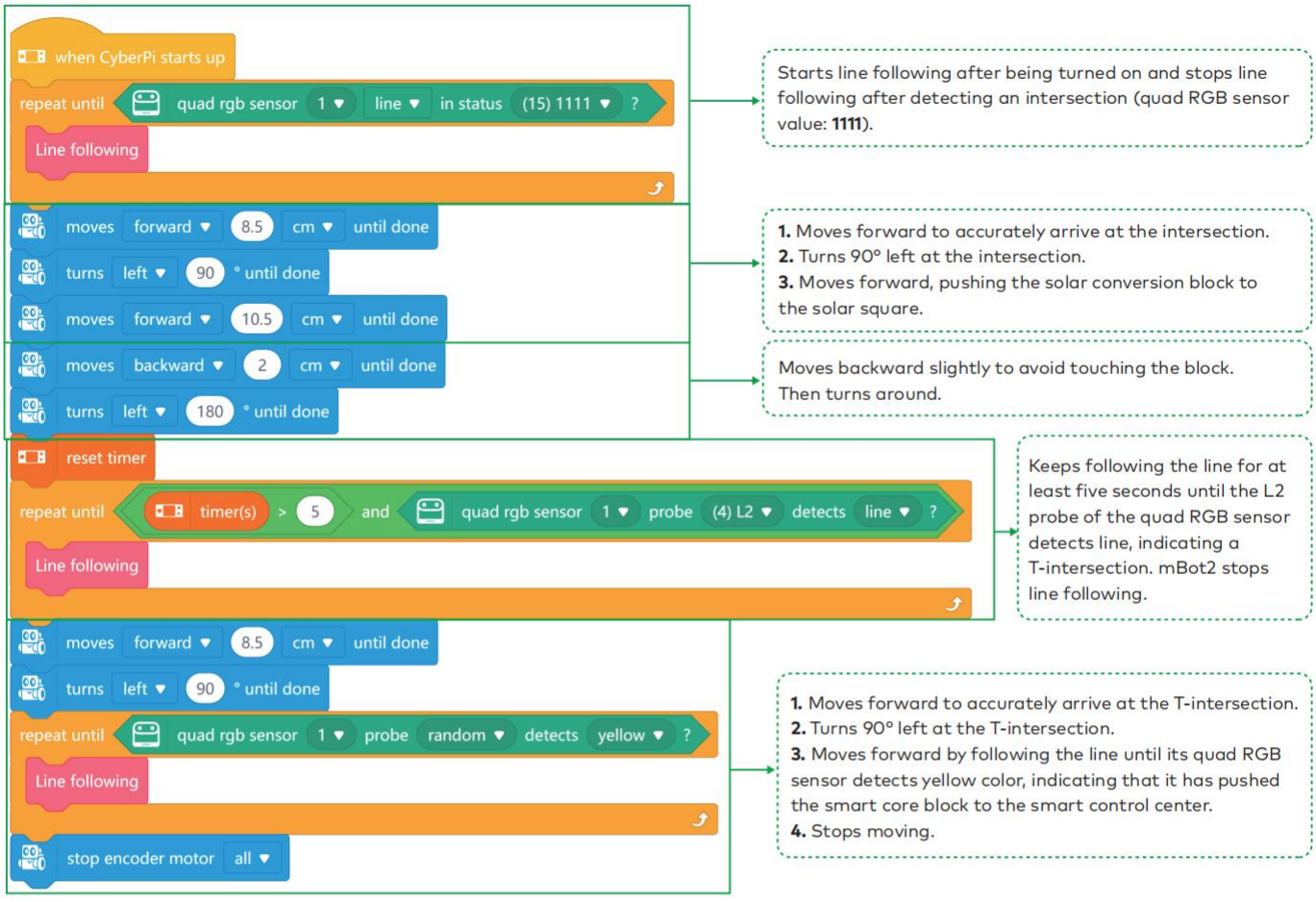

Why a timer used?

mBot2 stops line following when it detects a turn (the status value of its quad RGB sensor becomes **0111**).
To avoid this happening when mBot2 makes a turn, the timer function is introduced. It ensures that mBot2 detects a T-intersection only after it finishes line following and turning around. Pay attention that the timer period varies with the speed of mBot2.

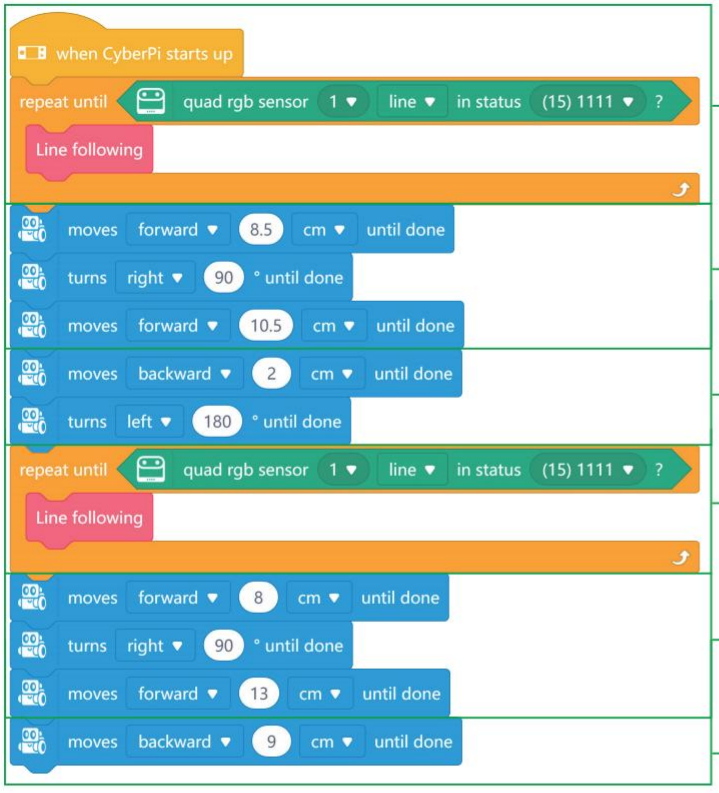
## Programs for task card 2

Note: The programs for mBot2s in task card 2 are reverse of those in task card 1.

### mBot2-A



**mBot2-B**




Starts line following after being turned on and stops line following after detecting an intersection (quad RGB sensor value: **1111**).

**1.** Moves forward to accurately arrive at the intersection.
**2.** Turns 90° right at the intersection.
**3.** Moves forward, pushing the maglev guidance block to the maglev hub.

Moves backward slightly to avoid touching the block. Then turns around.

Keeps following the line until detecting an intersection (quad RGB sensor value: **1111**).

**1.** Moves forward to accurately arrive at the intersection.
**2.** Turns 90° right at the intersection.
**3.** Moves forward, pushing the smart core block to the center intersection of the map.

Moves backward for a distance to avoid colliding with mBot2-A.

# Appendix

**1.** Where can I get more product information and troubleshooting suggestions?
support.makeblock.com/hc/en-us

**2.** Where can I access mBot2 classes?
STEM classes: makeblock.com/collections/stem-classes

**3.** How can I contact Makeblock technical support?
support@makeblock.com or service@makeblock.com

makeblock

Manufactured by: Makeblock Co., Ltd.
Address: Floor 4, Building C3, Nanshan iPark, No. 1001 Xueyuan Avenue,
Nanshan District, Shenzhen, Guangdong Province, 518000, China
www.makeblock.com
info@makeblock.com
T&C: https://www.makeblock.com/general-terms-conditions