CyberPi Series User Manual

Describes the features, electronic parts, and packages of CyberPi series products

Foreword

Thank you for choosing CyberPi series products!

The CyberPi series includes a powerful main control board, abundant and optimized structural and electronic parts, which can be used in multiple education scenarios including large-class teaching, community teaching, and online/offline education & training. The series covers multiple teaching fields including programming, makers, and robots and thus can meet the diversified education needs, such as AI, IoT, data science, and UI design.

This user manual describes the structural and electronic characteristics, supporting software, key electronic parts, classic example programs, and available suites of CyberPi series products, which helps you explore how you can use the CyberPi series more flexibly and effectively to learn or teach programming.

Reading suggestions:

To know about the product characteristics

Structural Characteristics

Electronic Characteristics

To learn programming

Programming Software

To know about the key electronic parts

CyberPi

Pocket Shield

mBuild Modules

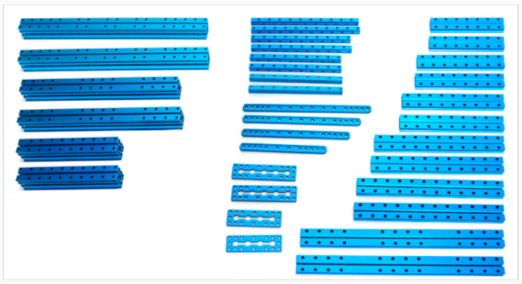
The content in this manual may be changed with the upgrade or modification of the products, without prior notice.

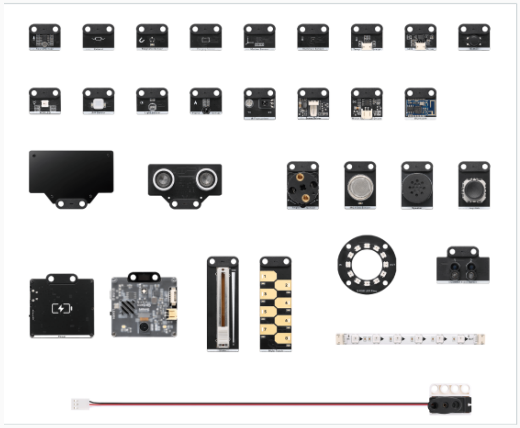
If you find any errors or have any ideas or suggestions on how to improve the product or this manual, contact us through cyber.list@makeblock.com.

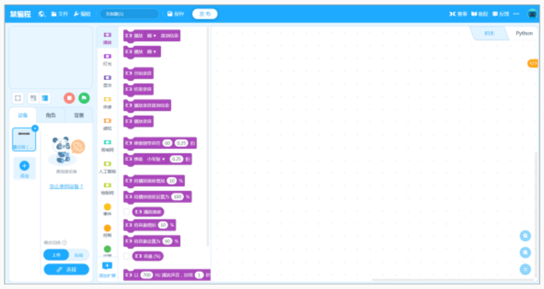
About the CyberPi Series

CyberPi is a new-generation networkable microcomputer designed for AI programming education. With the compact structure and powerful functions, it can be used to popularize AI education and advance your programming learning.

With the powerful programming software mBlock 5, extension boards, mechanical metal parts, and mBuild modules, CyberPi series products can be used in multiple education scenarios including large-class teaching, community teaching, and online/offline education & training. The products cover multiple teaching fields including programming, makers, and robots and thus can meet the diversified education needs, such as AI, IoT, data science, and UI design.





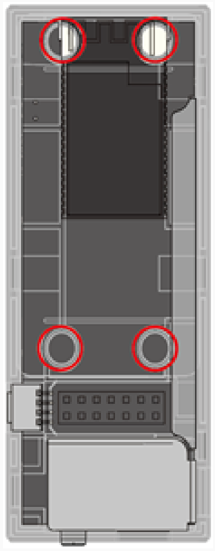
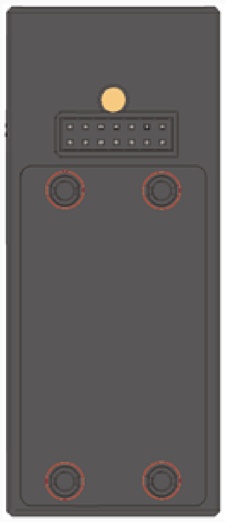


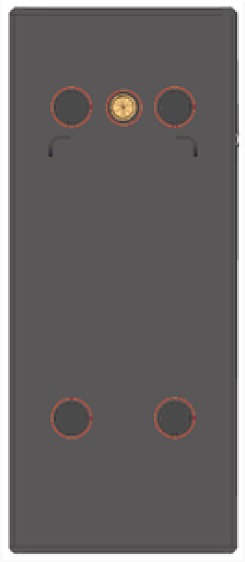
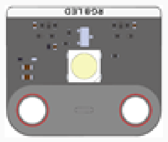
This chapter describes the following of the CyberPi series:

- Structural Characteristics
- Electronic Characteristics
- Programming Software
- Related Education System

Structural Characteristics

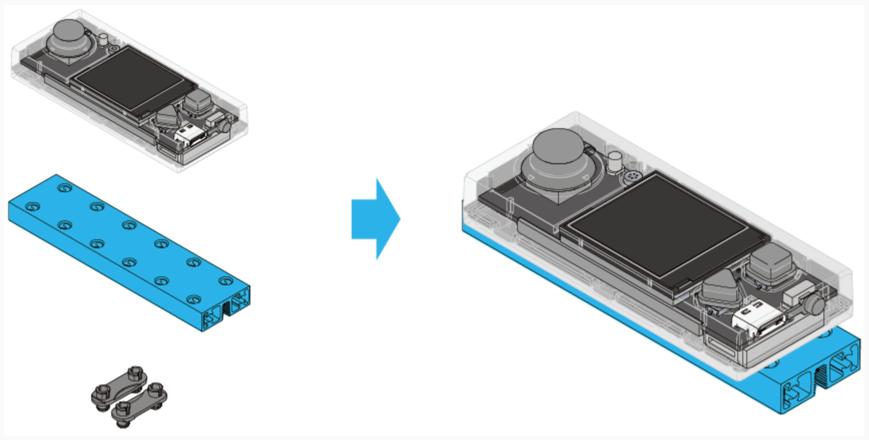
CyberPi series products are design with special appearances, such as holes, through-holes, pins, and tapped holes, which enable them to be extended with other parts or components.

Appearance design		Description
CyberPi		Holes on the back, used to: • connect to extension boards • connect to Lego blocks or Makeblock M4 metal parts through pins
Pocket Shield		Pins on the front, used to connect to CyberPi
		Holes and a tapped hole on the back. • The holes are used to connect to Lego blocks or Makeblock M4 metal parts through pins. • The tapped hole is used to connect to Makeblock M4 metal parts through screws.

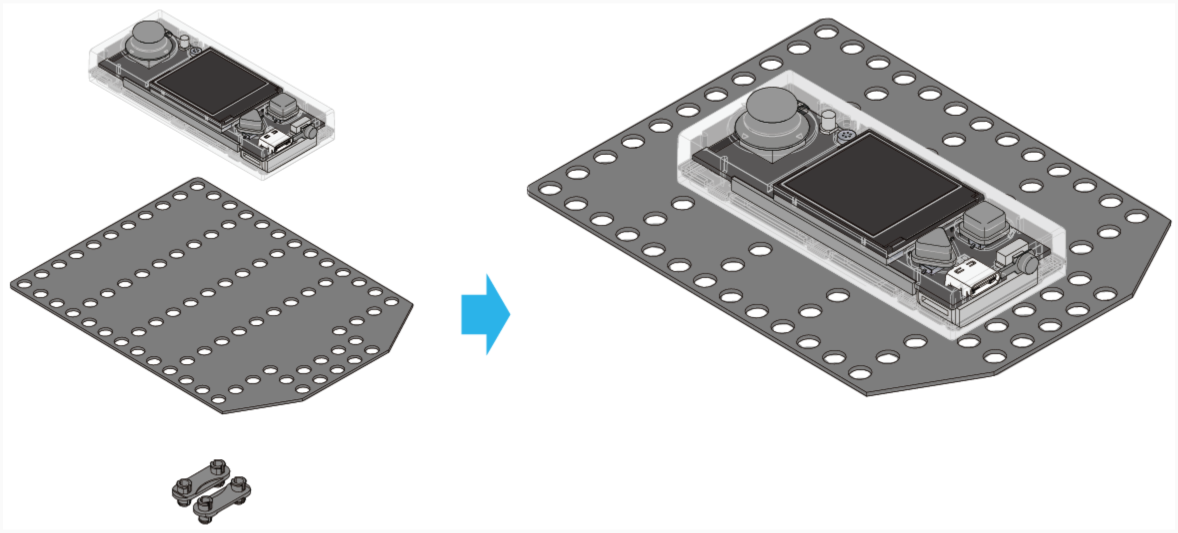
		
mBuild modules		Through-holes, used to connect to Makeblock M4 metal parts through pins or screws or Lego blocks through pins

The following provides some common connection examples:

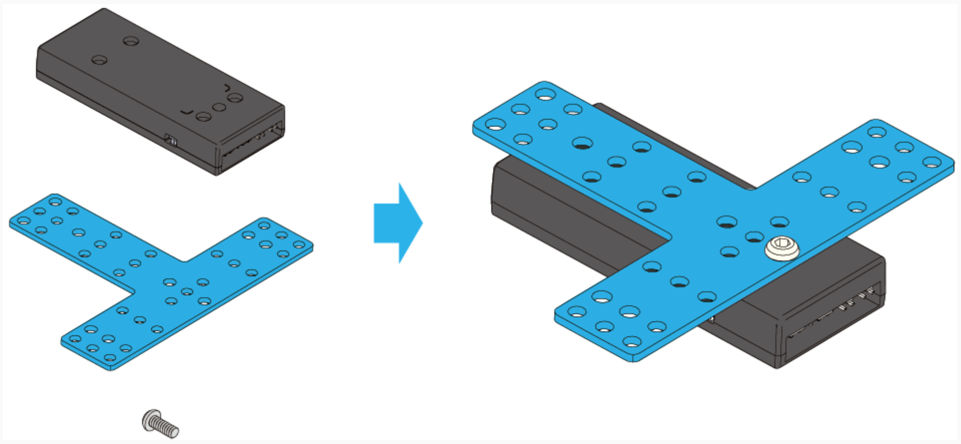
Example 1: CyberPi + one beam + one pin



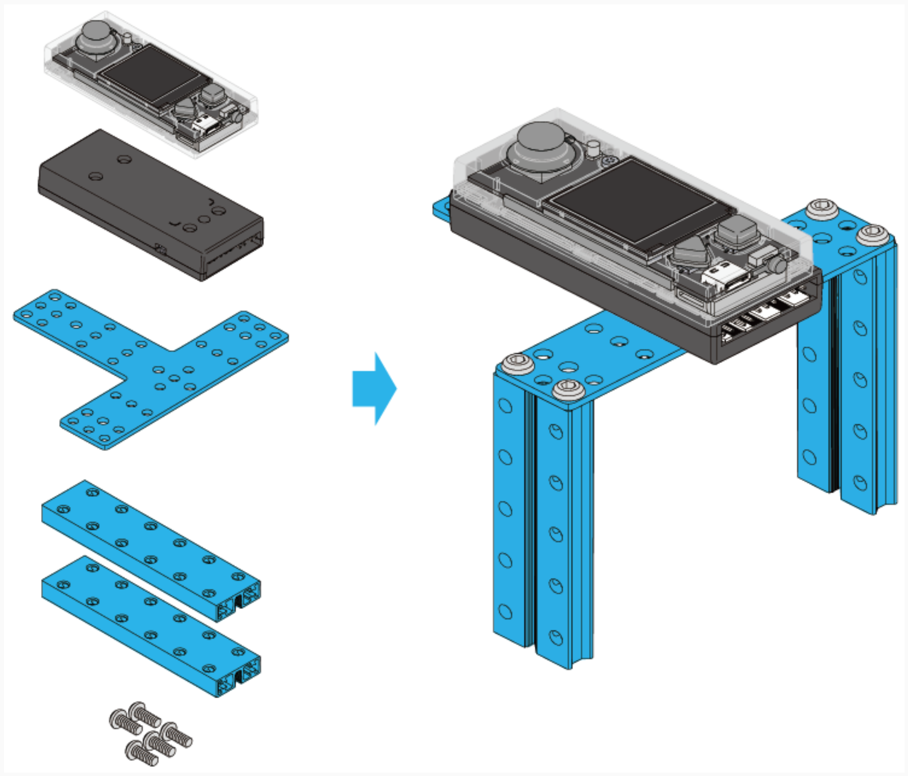
Example 2: CyberPi + one cut board + two pins



Example 3: Pocket Shield + one T-shaped plate + one M4 x 14 screw



Example 4: CyberPi + Pocket Shield + one T-shaped plate + two beams + five screws

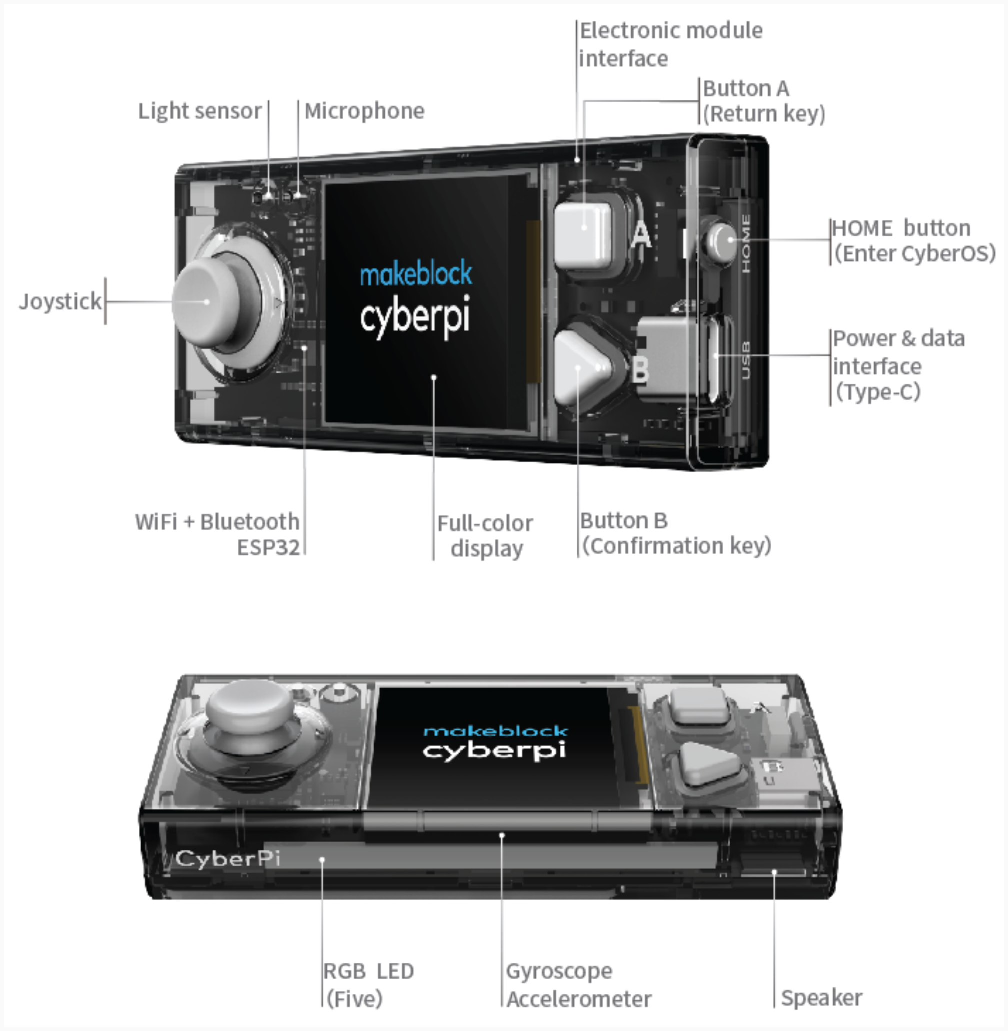


For more information about Makeblock metal parts, see Mechanical Parts on Maker's Platform.

Electronic Characteristics

Main control board

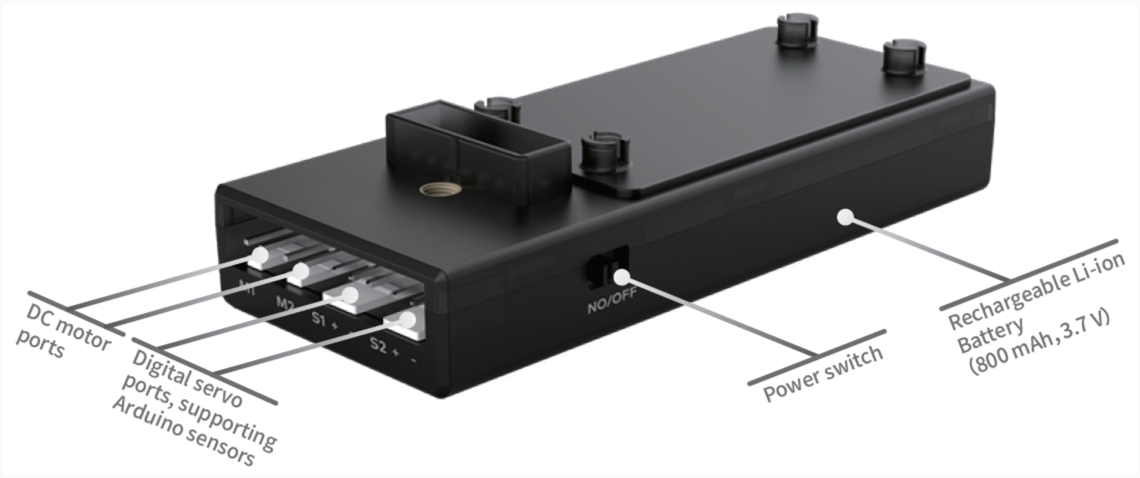
CyberPi



Extension boards

CyberPi can work with multiple extension boards to meet diversified education needs.

Pocket Shield


DC motor ports
Digital servo ports, supporting Arduino sensors
M1
M2
S1
S2
NO/OFF
Power switch
Rechargeable Li-ion Battery (800 mAh, 3.7 V)

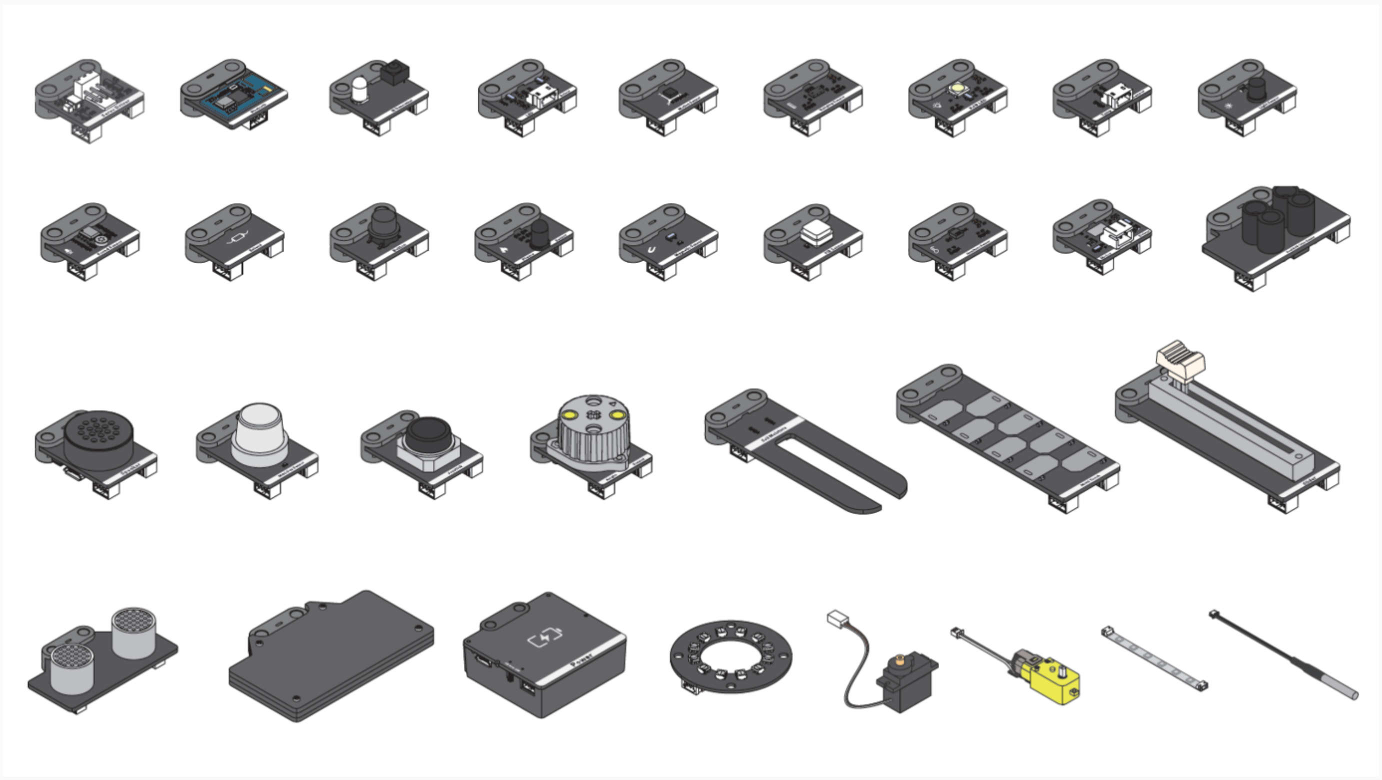
For more information about Pocket Shield, see Pocket Shield.

More extension boards are being developed. Stay tuned!

Electronic modules

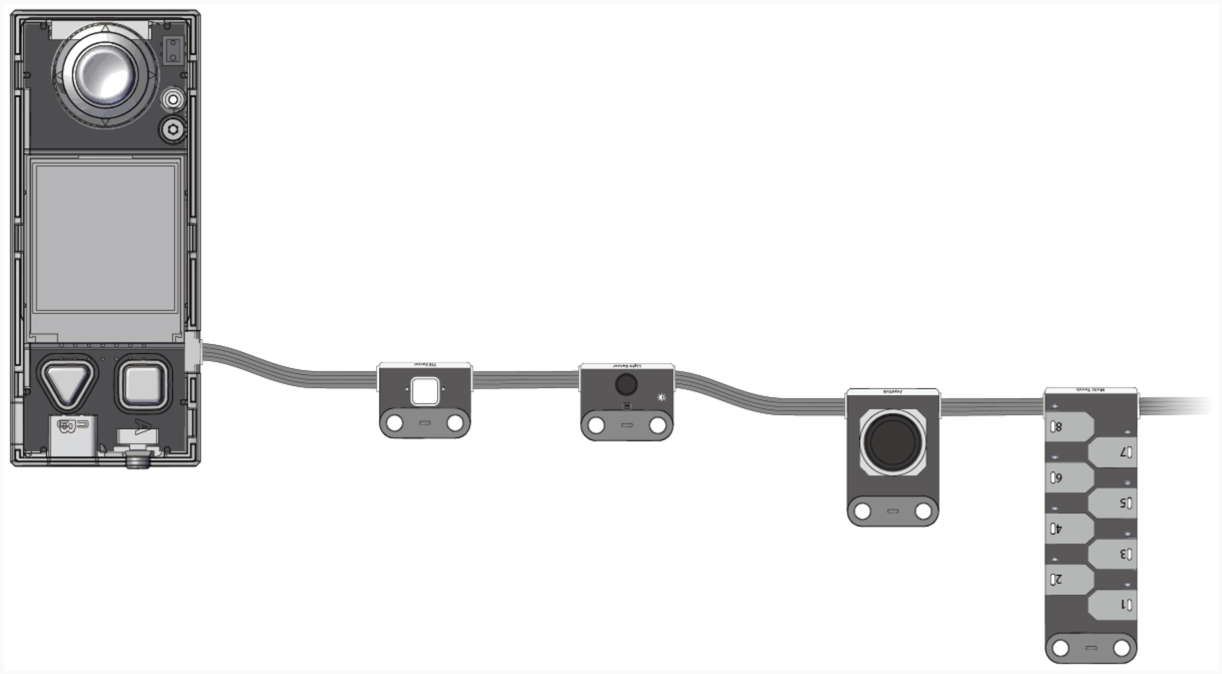
In addition to the mBuild electronic modules, the CyberPi series supports third-party electronic components and parts, for example, Arduino modules.

mBuild electronic modules



Currently, over 30 mBuild electronic modules have been developed, and more modules will be available.

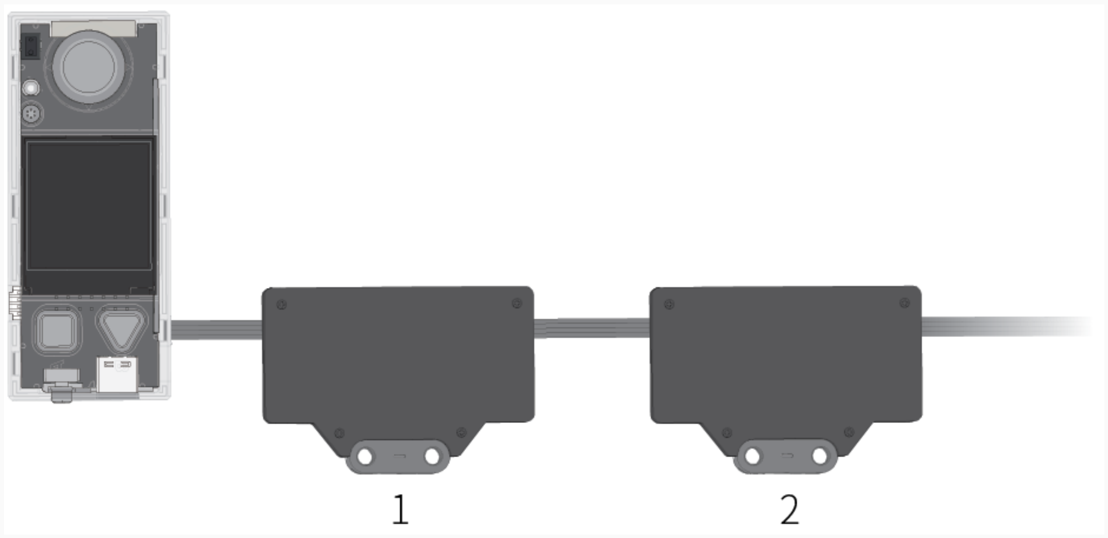
Each mBuild module is equipped with a micro-processing chip, which enables multiple modules to connect to one port in series, as shown in the following figure.



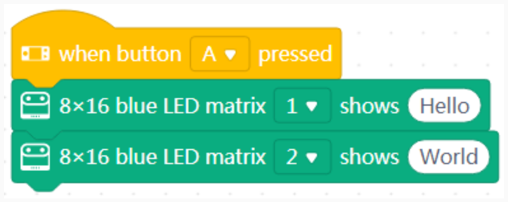
In addition, CyberPi can intelligently identify the positions of the modules, which simplifies your programming. You don't have to set the information about the positions of the modules when you add or remove a module.

Intelligent position identification

Example:
Connect CyberPi to multiple LED matrixes



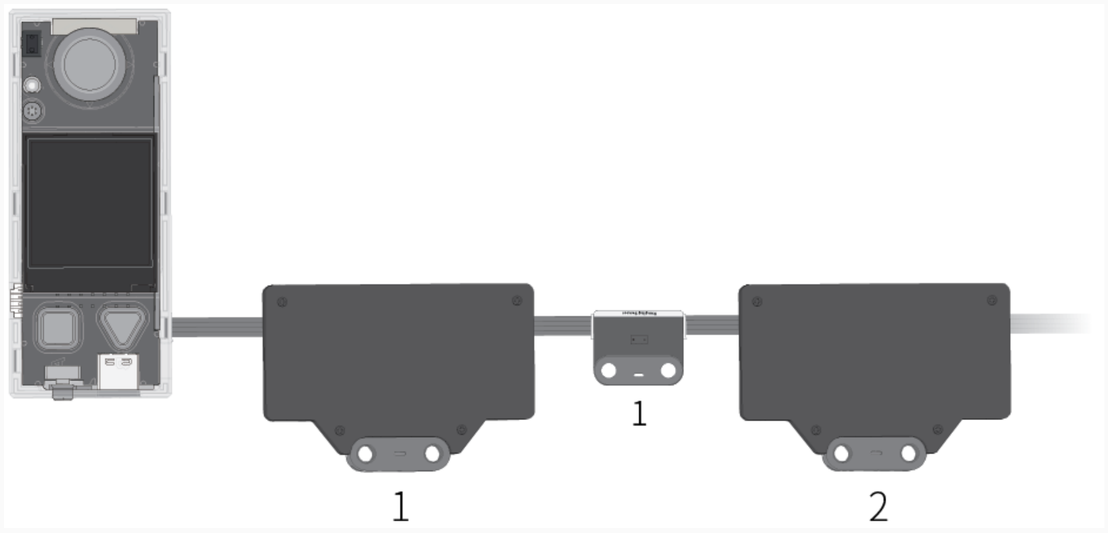
After connecting CyberPi to multiple LED matrixes, you need only to specify the place of an LED matrix among the ones connected instead of specifying the port to which the LED matrix is connected when compiling a program. As shown in the preceding figure, the first LED matrix connected to CyberPi is numbered 1, the second one numbered 2, and so on.



```
when button A ▾ pressed
8×16 blue LED matrix 1 ▾ shows Hello
8×16 blue LED matrix 2 ▾ shows World
```

When you press button A on CyberPi, the first LED matrix displays "Hello," and the second one displays "World."

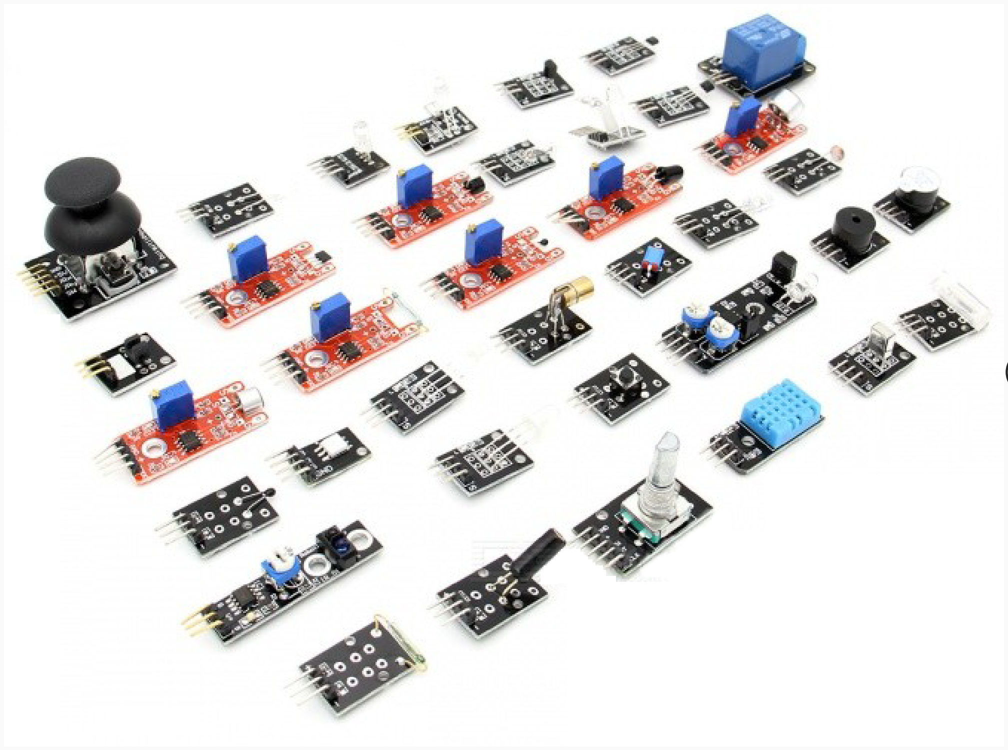
Change the positions of the modules, as shown in the following figure.



The preceding program still works after you add a ranging sensor. When you press button A on CyberPi, the first LED matrix displays "Hello," and the second one displays "World."

For more information, see mBuild Electronic Modules.

Third-party sensors


(Source: Internet webpage)

The CyberPi series is compatible with multiple third-party sensors. You can read Open-source Materials to understand how CyberPi series products are connected to third-party electronic components or parts.

Motors

Working in combination with the corresponding electronic modules or extension boards, CyberPi can drive multiple types of motors. The following table describes the motors supported by CyberPi.

Supporting	Through mBuild modules	Through Pocket Shield	Through the mBot2 extension board	Through the Challenge

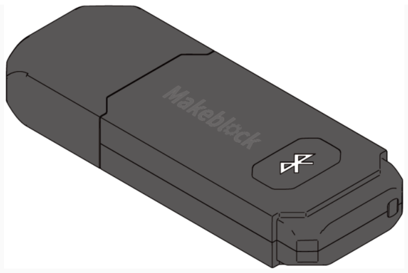
				extension board
5V TT motor	Yes	Yes	Yes	Yes
5V 9g servo	Yes	Yes	Yes	Yes
6—12V smart servo	Yes	Yes	Yes	Yes
12V encoder motor			Yes	Yes
Brushless motor				Yes
Stepper motor				Yes

Other accessories

Bluetooth Controller

For more information about Makeblock Bluetooth Controller, see Bluetooth Controller Online Help.

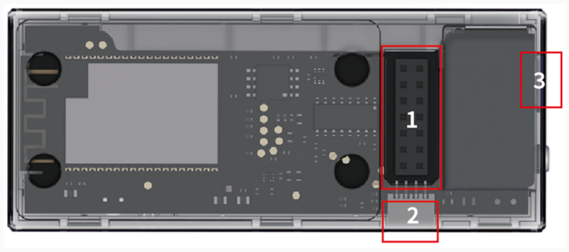
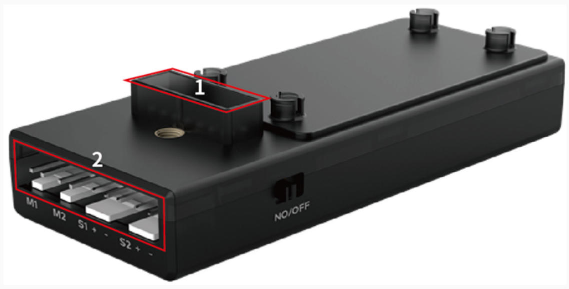
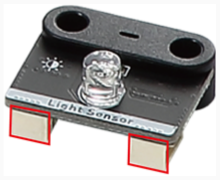
Bluetooth Dongle



For more information about Makeblock Bluetooth Dongle, see Bluetooth Dongle Quick Start Guide.

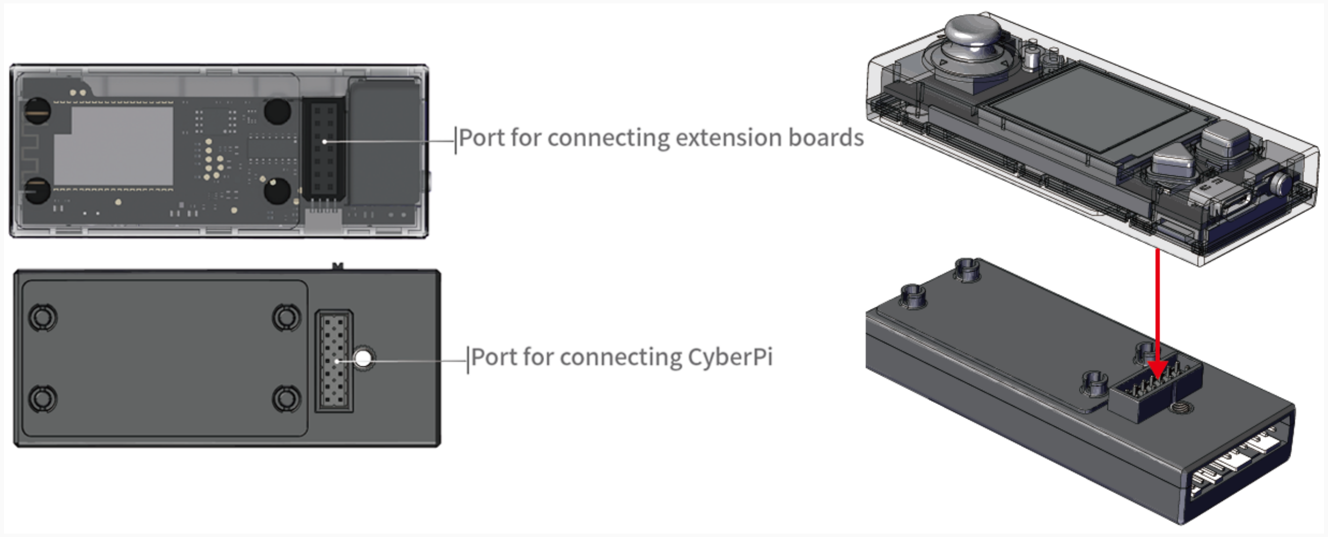
Interfaces

CyberPi series products are designed with multiple interfaces, which enable them to easily connect to other electronic components and parts, facilitating the extension of abundant functions.

Interface		Description
CyberPi		1: used to connect extension boards 2: used to connect mBuild electronic modules 3: Type-C USB cable, used to connect a PC
Pocket Shield		1: used to connect CyberPi 2: DC motor ports M1 and M2, used to connect motors; Digital servo ports S1 and S2, used to connect servos or LED strips
mBuild electronic module		Used to connect a main control board or other mBuild electronic modules

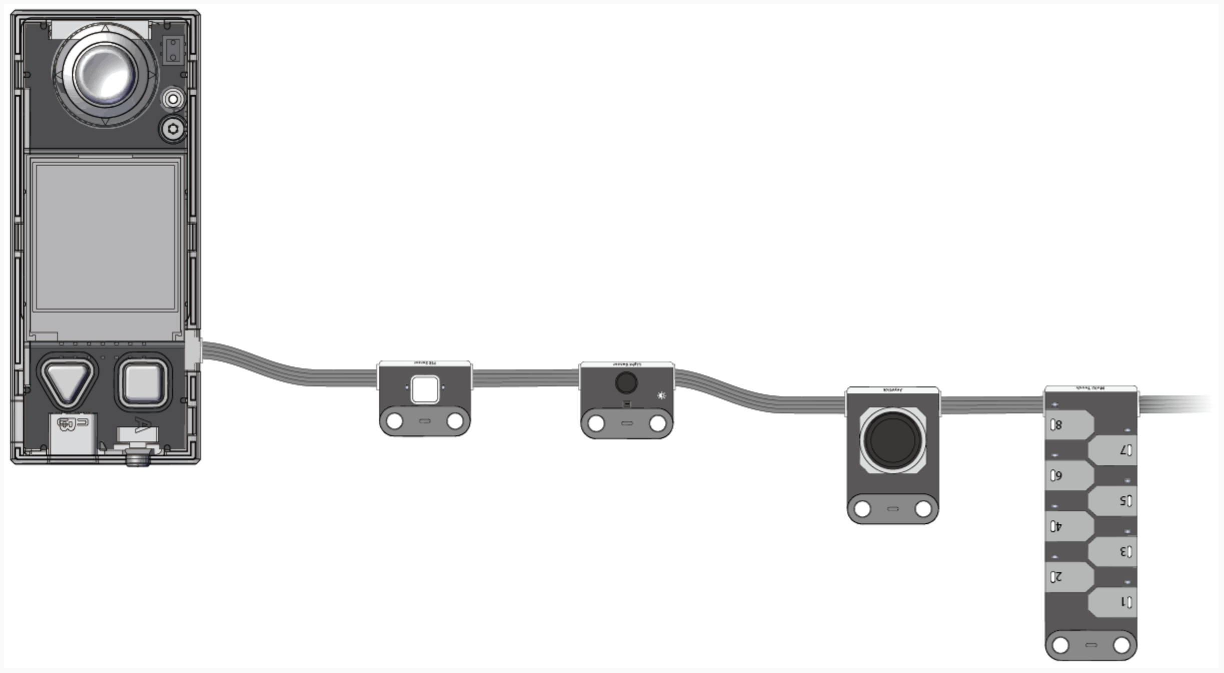
The following provides some common function extension examples:

Example 1: Connecting Pocket Shield to CyberPi


Port for connecting extension boards
Port for connecting CyberPi

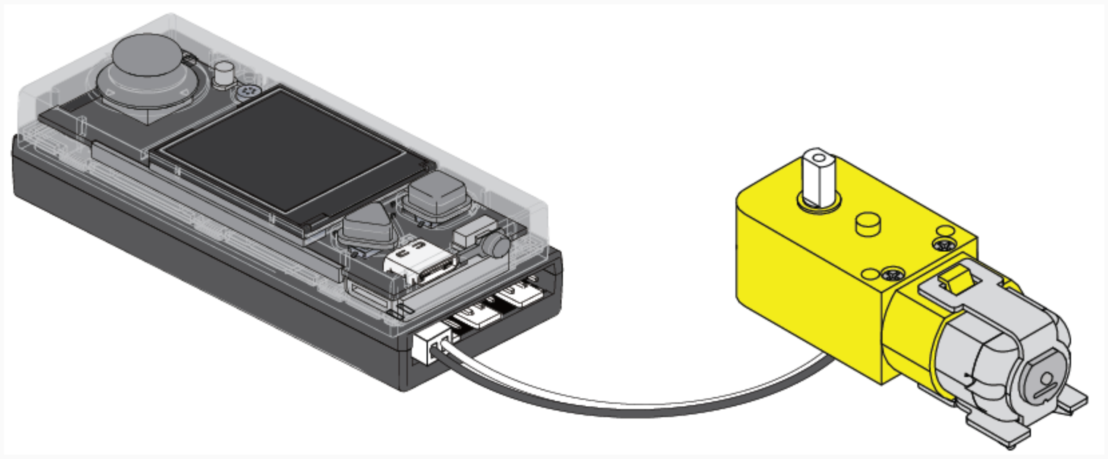
Pocket Shield is equipped with a built-in rechargeable battery that can supply power for CyberPi and provides 2-pin and 3-pin interfaces that can be used to connect servos, LED strips, and motors, which significantly improves the extensibility of CyberPi.

Example 2: Connecting mBuild modules to CyberPi



mBuild modules are small in size but rich in functions. CyberPi can be connected to multiple mBuild modules in series.

Example 3: Connecting a motor to CyberPi through Pocket Shield



CyberPi can connect to multiple types of motors through other modules or extension boards.

Cables

CyberPi series products supports multiple types of connection cables, as described in the following table.

Name	Description
Type-C USB cable	Used to connect CyberPi to PCs for power supply or program/command transmission
4-pin cable	Used to connect mBuild modules
3-pin cable	Used to connect Pocket Shield or mBot 2 to LED strips
	Used to connect mBuild LED drivers to LED strips
2-pin cable	Used to connect Pocket Shield or mBot 2 to DC motors
	Used to connect mBuild motor drivers to DC motors

Programming Software

You can use mBlock 5 to program CyberPi series products. mBlock 5 provides two editors, namely the block-based graphical editor (the default editor, referred to as mBlock 5) and Python editor (referred to as mBlock-Python Editor).

mBlock 5

Obtaining mBlock 5

Select the version applicable to your device. For example, if you use a PC, you can download the version for PCs or use mBlock on the web.

- For PCs
- On the Web (mLink Required)
- On Chromebook (mLink Required)
- For Android and iOS

Learning how to use mBlock 5

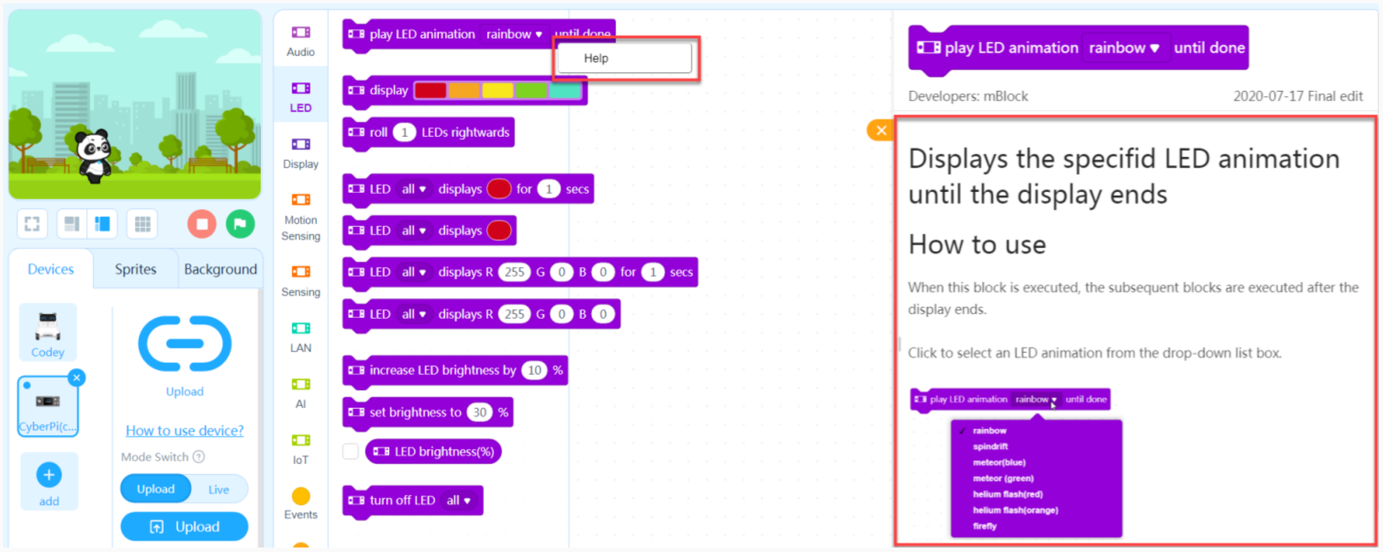
Before using mBlock to program CyberPi series products, you need to know about some basic operations.

- Connect Devices
- Device Library and Extension Center
- Live and Upload Modes
- Programming Languages
- Interact with Sprites

For more information and help about mBlock 5, see the *mBlock 5 Online Help*.

Block help

mBlock 5 provides multiple types of block for CyberPi series products, and you can right-click a block to see the help for it.



Developers: mBlock

2020-07-17 Final edit

Displays the specifid LED animation until the display ends

How to use

When this block is executed, the subsequent blocks are executed after the display ends.

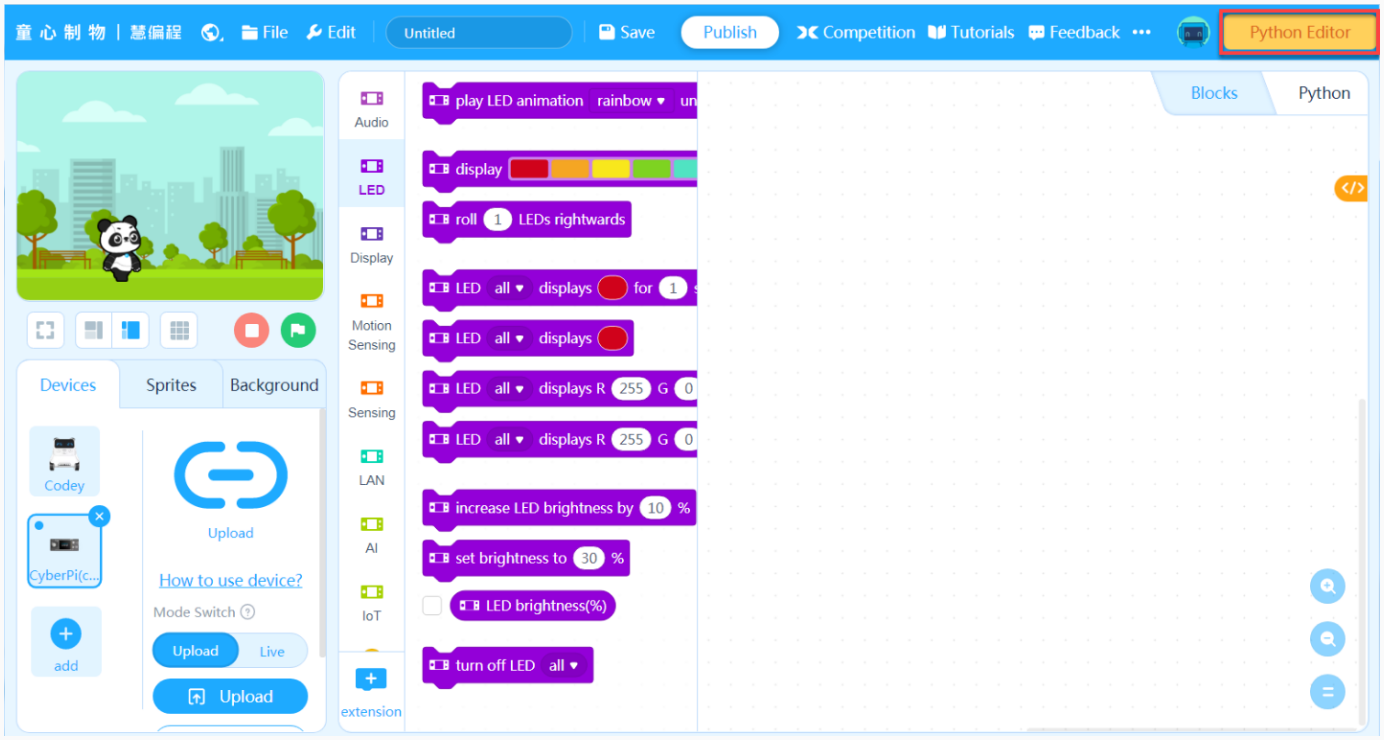
Click to select an LED animation from the drop-down list box.

mBlock-Python Editor

Open mBlock-Python Editor

Way 1

You can click the editor switching button on the upper-right of mBlock 5 on the web to go to mBlock-Python Editor.



Way 2

Open mBlock-Python Editor by entering its URL address: https://python.mblock.cc

For more information, see "Open mBlock-Python Editor."

Python API

Python API Documentations

The Python library `cyberpi` is provided for CyberPi series products. For details, see "Python API Documentation for CyberPi."

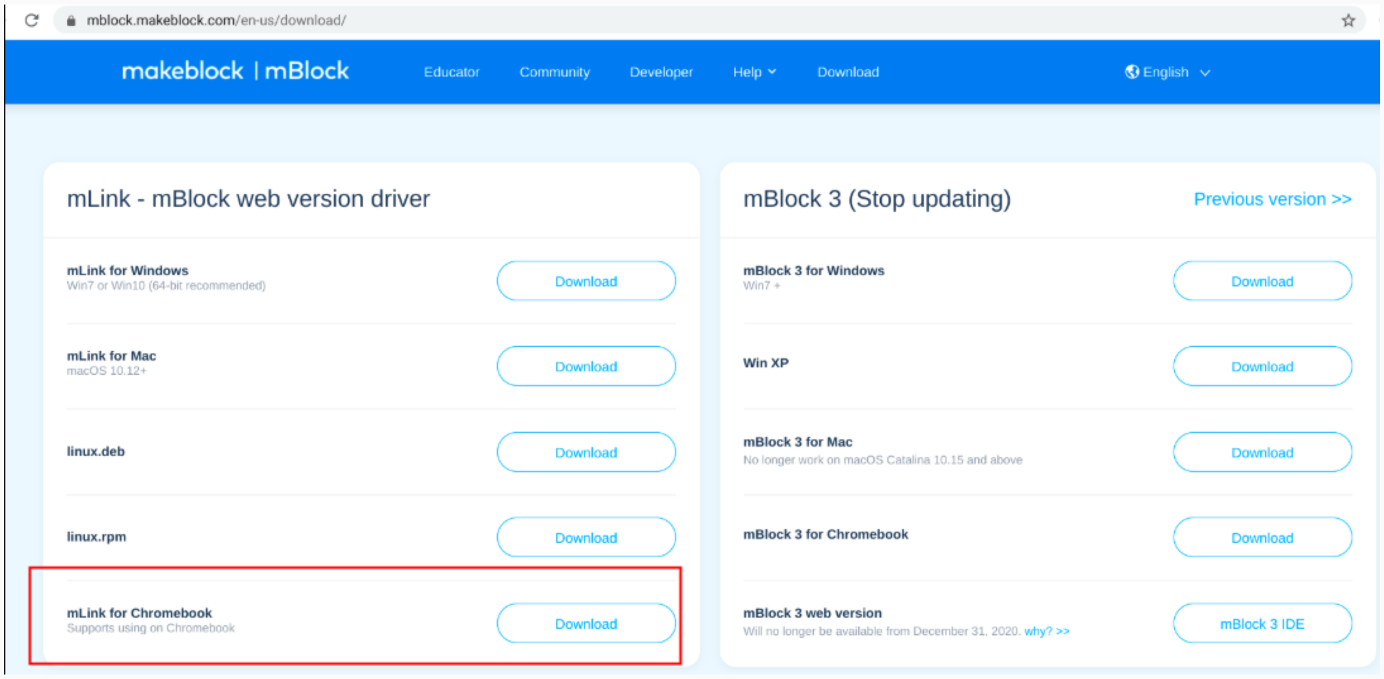
Quick access to the Python API Documentation for CyberPi

On mBlock-Python Editor, click **Tutorials** on the toolbar, and then click **Python API Documentation for CyberPi** in the title list.

Programming CyberPi on Chromebook

Install mLink on Chromebook and Start mBlock 5

Step1: Please access this link and find mLink for Chromebook (need pull down the page).



mblock.makeblock.com/en-us/download/

mLink - mBlock web version driver

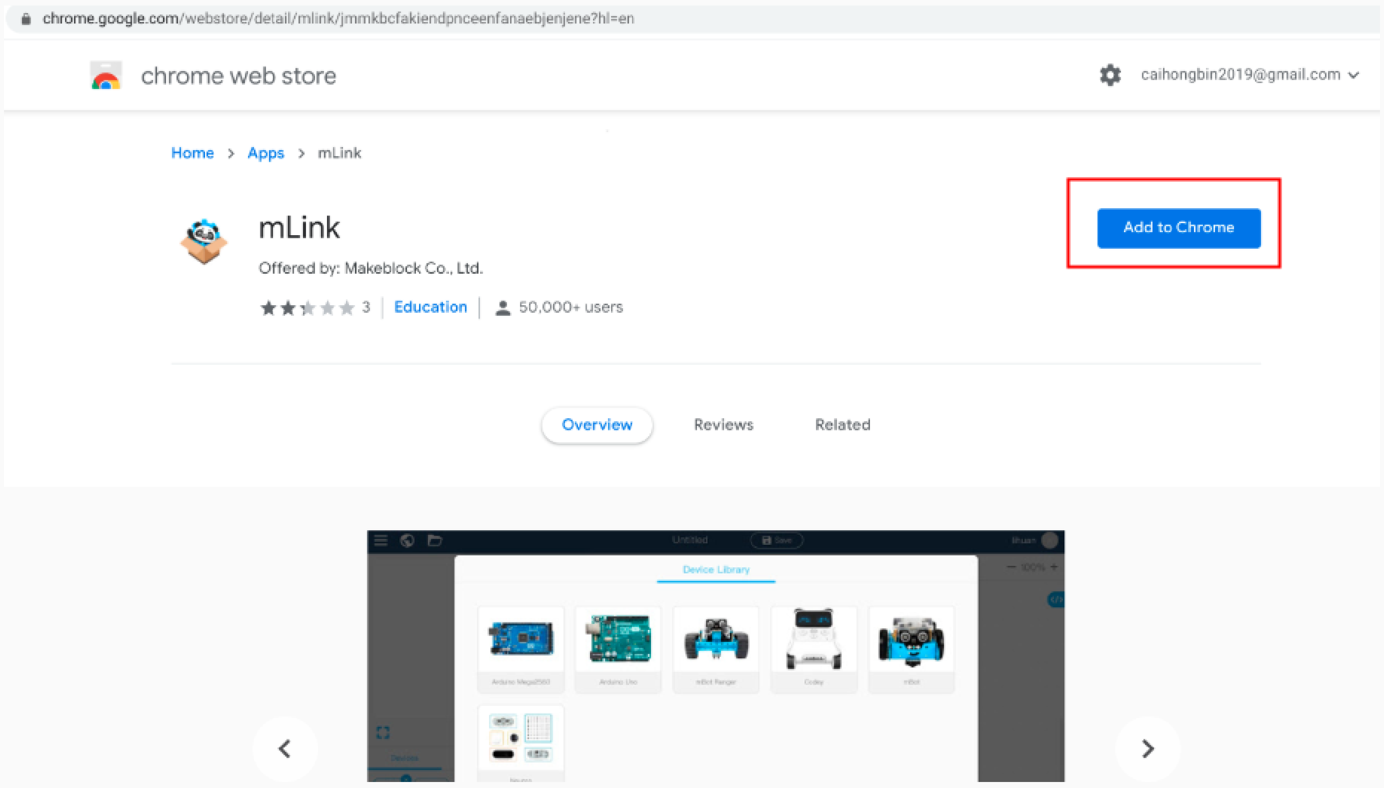
mLink for Windows Win7 or Win10 (64-bit recommended)	Download
mLink for Mac macOS 10.12+	Download
linux.deb	Download
linux.rpm	Download
mLink for Chromebook Supports using on Chromebook	Download

mBlock 3 (Stop updating) — Previous version >>

mBlock 3 for Windows Win7 +	Download
Win XP	Download
mBlock 3 for Mac No longer work on macOS Catalina 10.15 and above	Download
mBlock 3 for Chromebook	Download
mBlock 3 web version Will no longer be available from December 31, 2020. why? >>	mBlock 3 IDE

Step 2: Please click on "**Download**" to enter the chrome web store, then please click on "**Add to Chrome**" to install mLink.

Note: If you have already installed mLink before, you will see the option "Launch app" directly and you can bypass step2.


chrome.google.com/webstore/detail/mlink/jmmkbcfakiendpnceenfanaebjenjene?hl=en

chrome web store

caihongbin2019@gmail.com

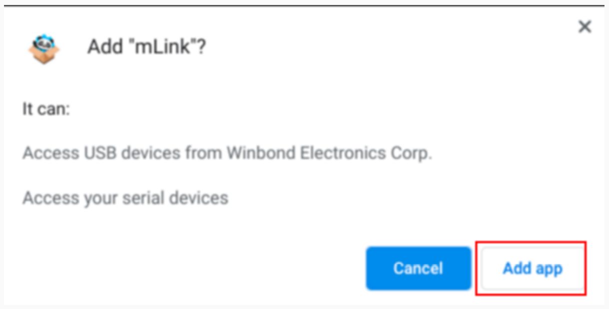
Home > Apps > mLink

mLink

Offered by: Makeblock Co., Ltd.

3 | Education | 50,000+ users

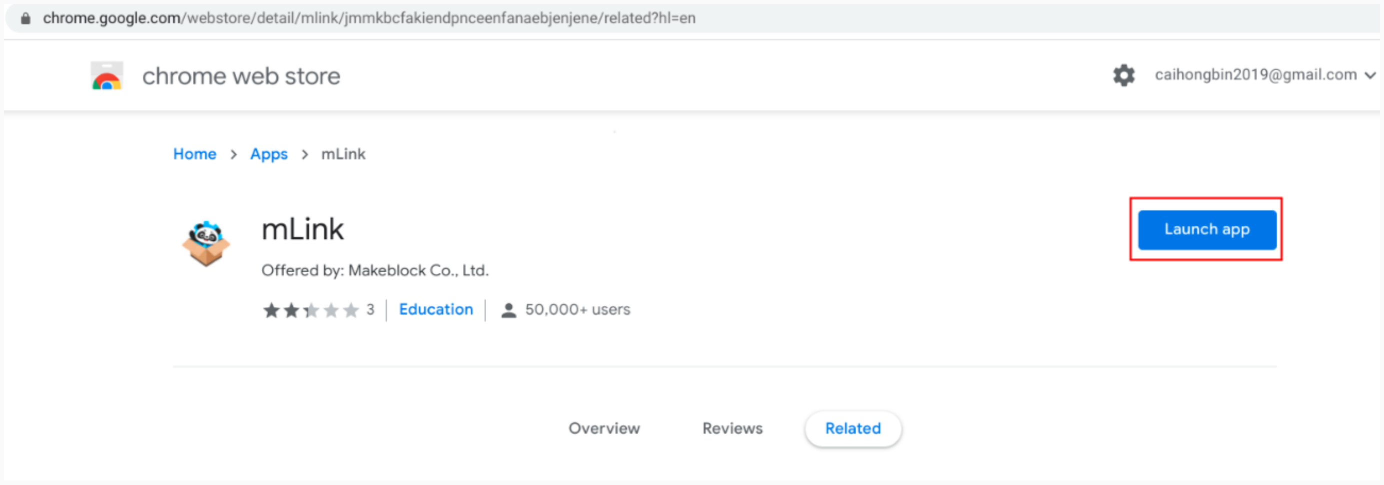
Then you will see a window pops up, just click on "**Add app**", and the option "**Launch app**" will be available.


Add "mLink"?

It can:

Access USB devices from Winbond Electronics Corp.

Access your serial devices


chrome.google.com/webstore/detail/mlink/jmmkbcfakiendpnceenfanaebjenjene/related?hl=en

chrome web store

caihongbin2019@gmail.com

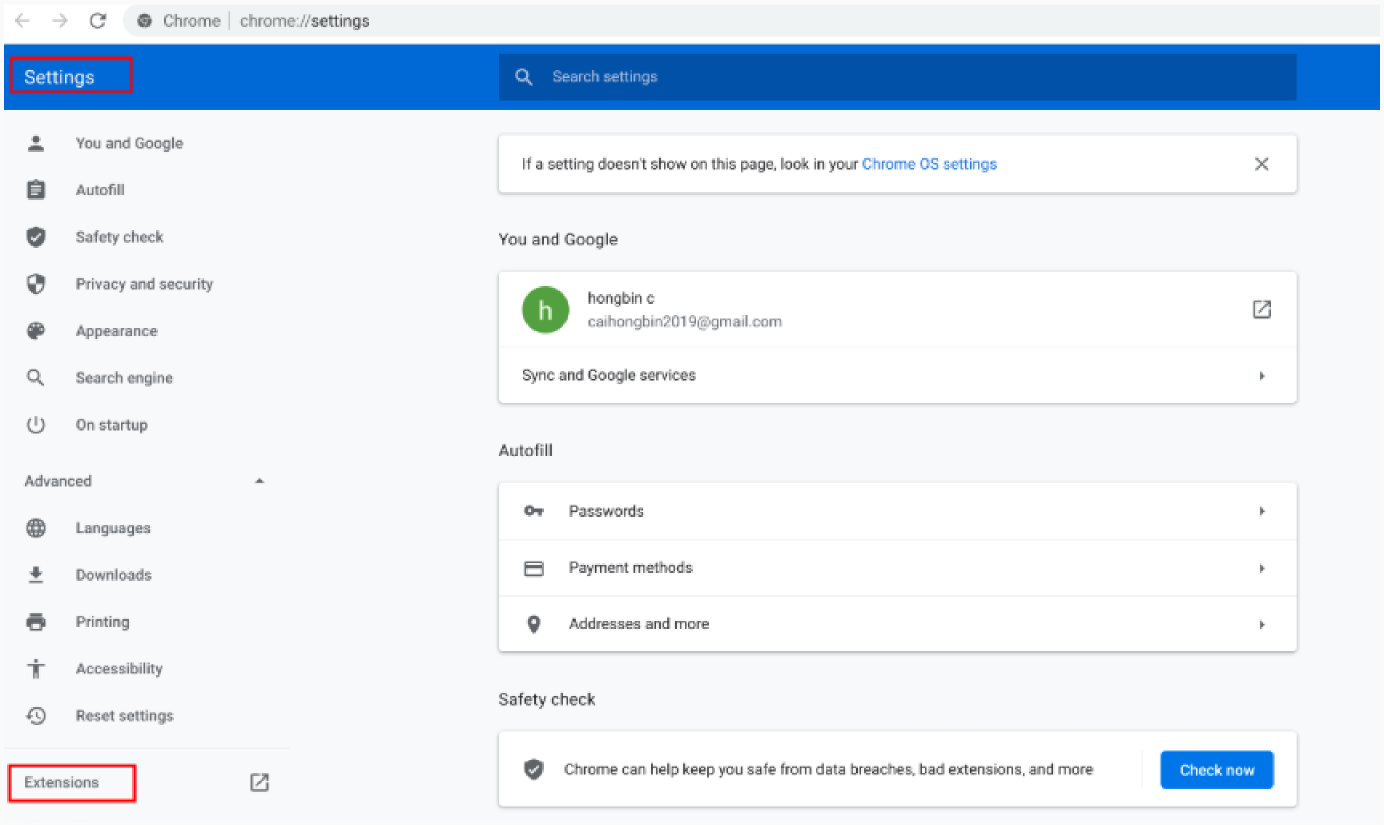
Home > Apps > mLink

mLink

Offered by: Makeblock Co., Ltd.

3 | Education | 50,000+ users

Note: Once you can't find the "**Launch app**" option after installed mLink, please check if mLink is enabled by going to "**Settings**"->"Extension".**Extension**".



If a setting doesn't show on this page, look in your Chrome OS settings

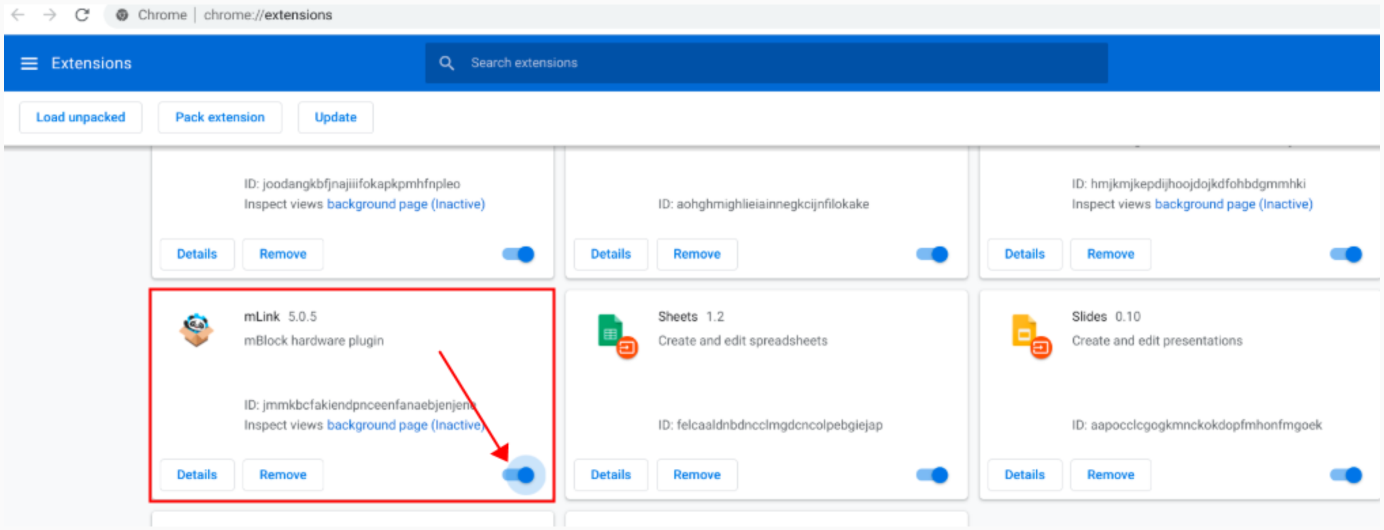
You and Google

hongbin c
caihongbin2019@gmail.com

Autofill

Safety check

Chrome can help keep you safe from data breaches, bad extensions, and more



ID: joodangkbfjnajiiifokapkpmhfnpleo
Inspect views background page (Inactive)

ID: aohghmighlieiainnegkcijnfilokake

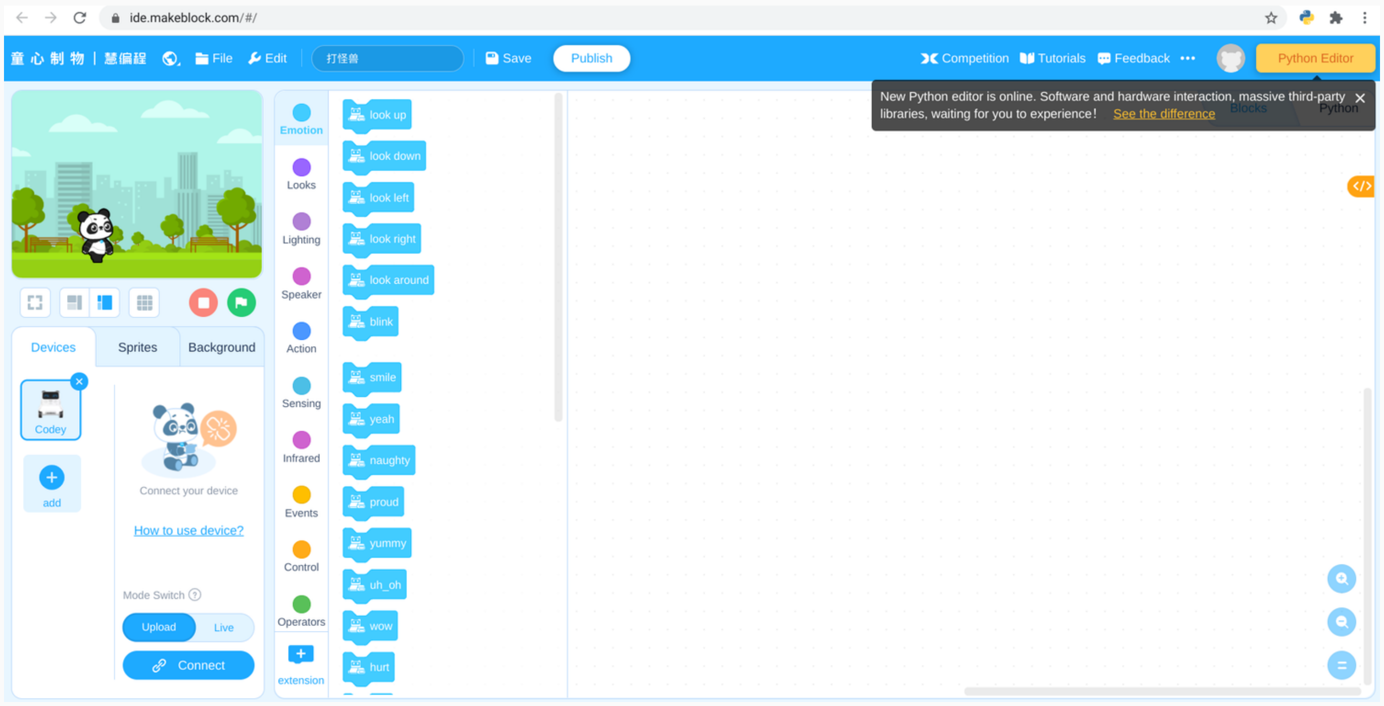
ID: hmjkmjkepdijhoojdojkdfohbdgmmhki
Inspect views background page (Inactive)

mLink 5.0.5
mBlock hardware plugin
ID: jmmkbcfakiendpnceenfanaebjenjen
Inspect views background page (Inactive)

Sheets 1.2
Create and edit spreadsheets
ID: felcaaldnbdncclmgdcncolpebgiejap

Slides 0.10
Create and edit presentations
ID: aapocclcgogkmnckokdopfmhonfmgoek

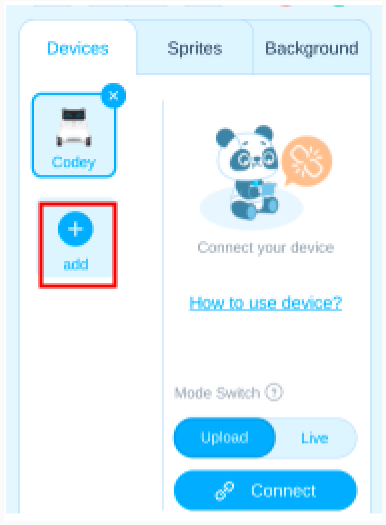
Step 3: Click on "**Launch app**" and you will enter the web version mblock 5 software:



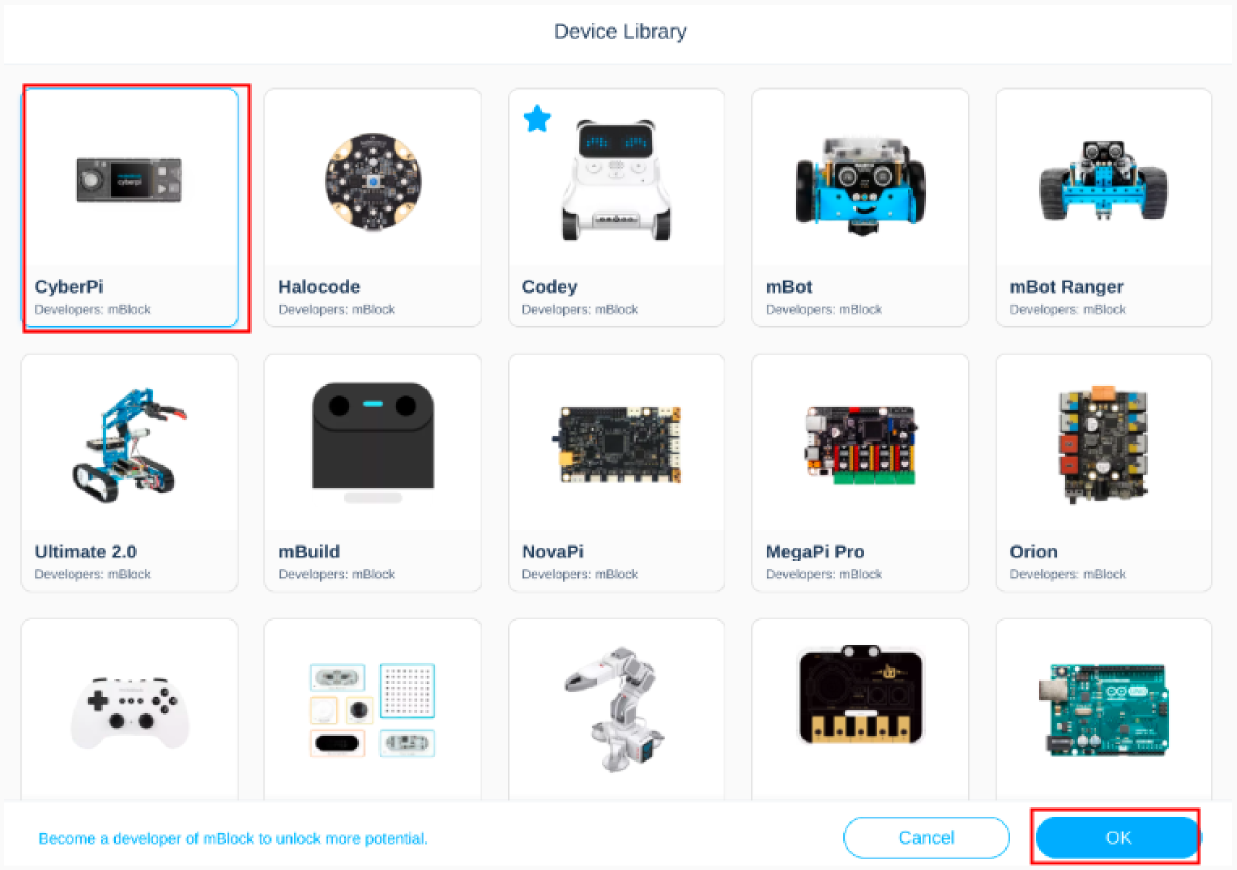
Connect CyberPi to mBlock 5 on Chromebook

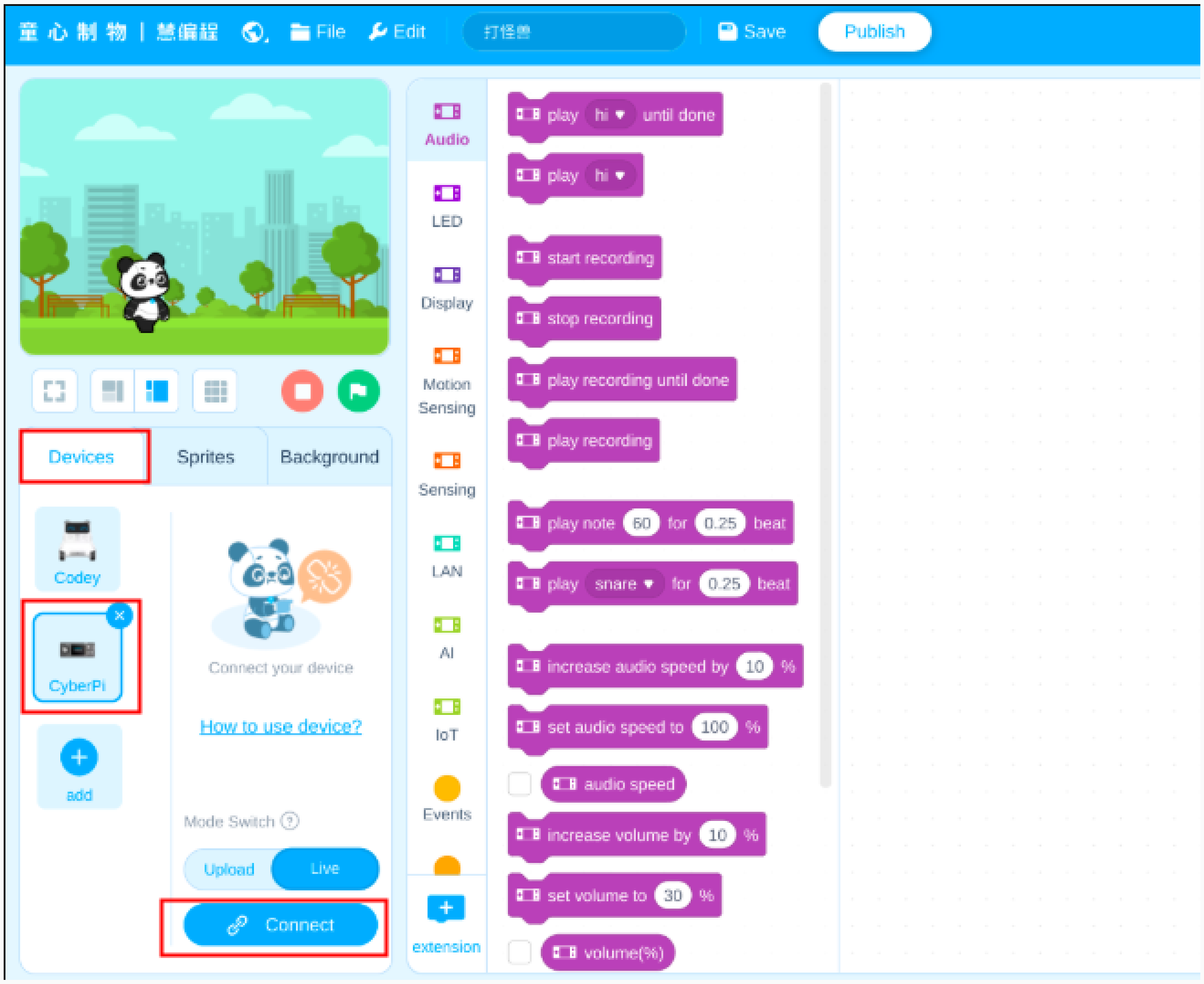
Step 1: Connect CyberPi to the Chromebook with a USB cable.

Step 2: Click on the "**+**" icon to open the Device Library:

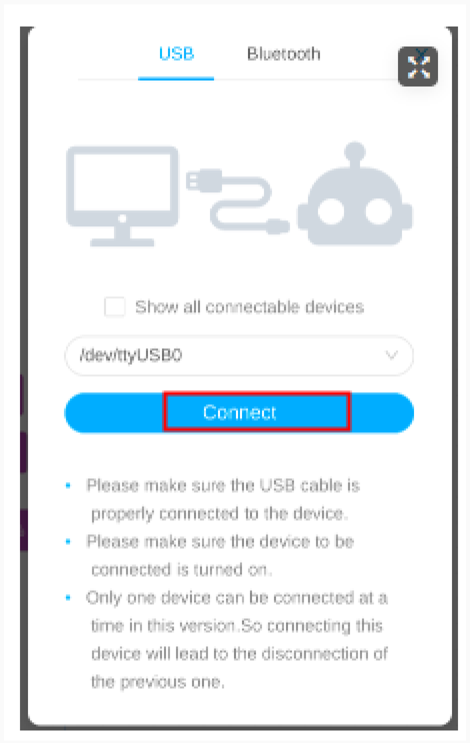


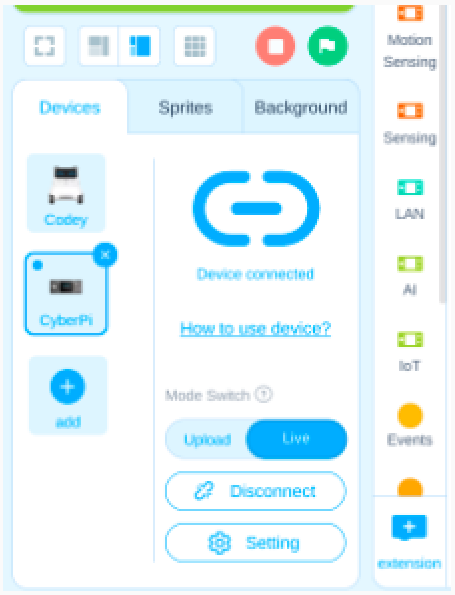
Step 3: Find the device "**CyberPi**" and select it, then click on "**Ok**" and you will see device CyberPi added under "**Devices**":


Device Library
CyberPi
Developers: mBlock
Halocode
Developers: mBlock
Codey
Developers: mBlock
mBot
Developers: mBlock
mBot Ranger
Developers: mBlock
Ultimate 2.0
Developers: mBlock
mBuild
Developers: mBlock
NovaPi
Developers: mBlock
MegaPi Pro
Developers: mBlock
Orion
Developers: mBlock
Become a developer of mBlock to unlock more potential.
Cancel
OK



Step 4: Click on the icon "**Connect**" and a connection window pops up, normally, the serial port for CyberPi will be available automatically, please just clink on "**Connect**" and you will see the status after connected.

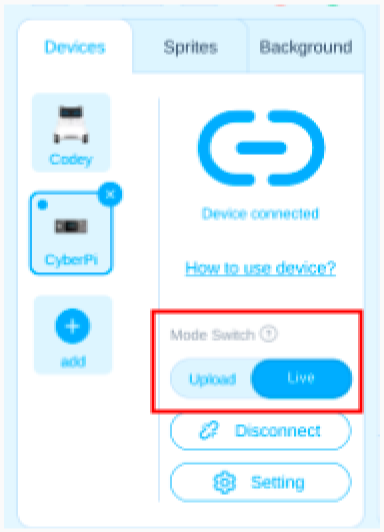

USB
Bluetooth
Show all connectable devices
/dev/ttyUSB0
Connect
- Please make sure the USB cable is properly connected to the device.
- Please make sure the device to be connected is turned on.
- Only one device can be connected at a time in this version.So connecting this device will lead to the disconnection of the previous one.



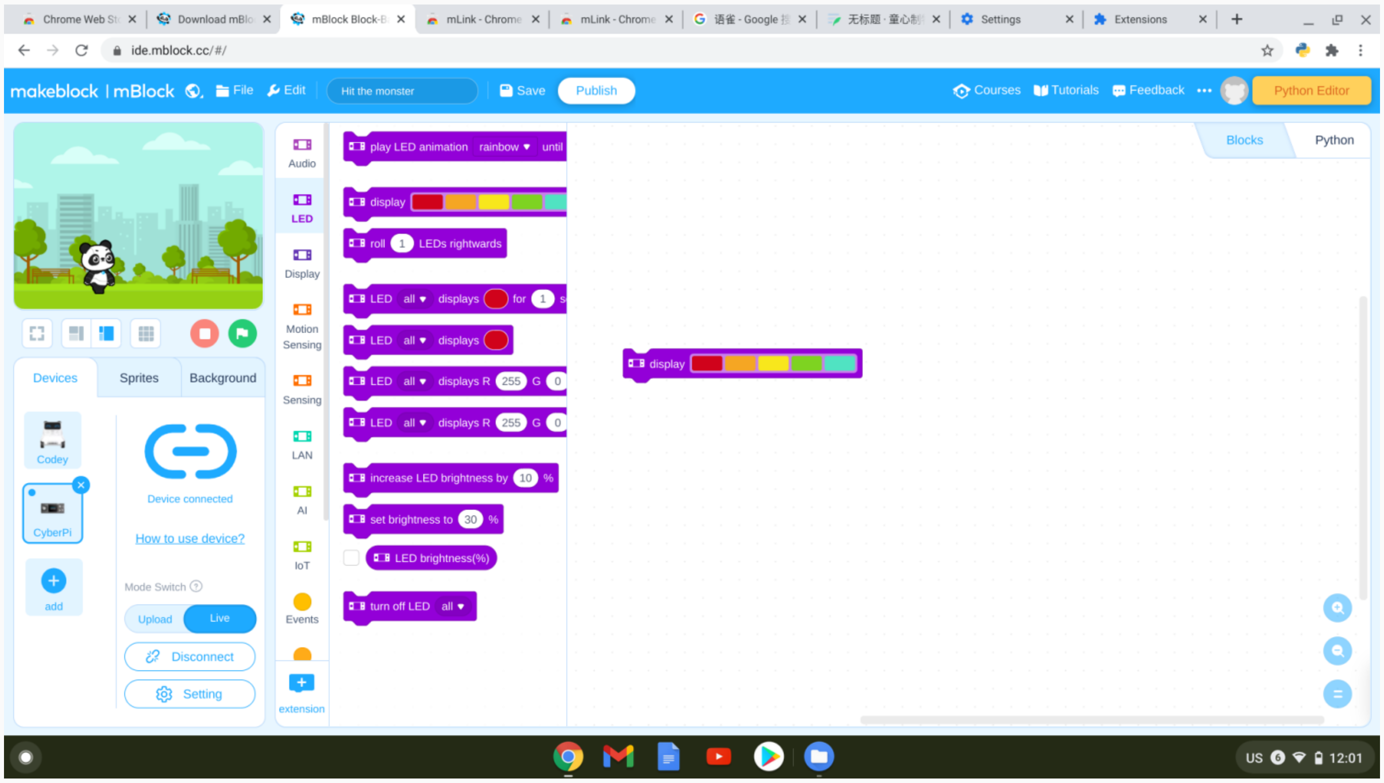
Programming CyberPi with mBlock 5 on Chromebook

live mode

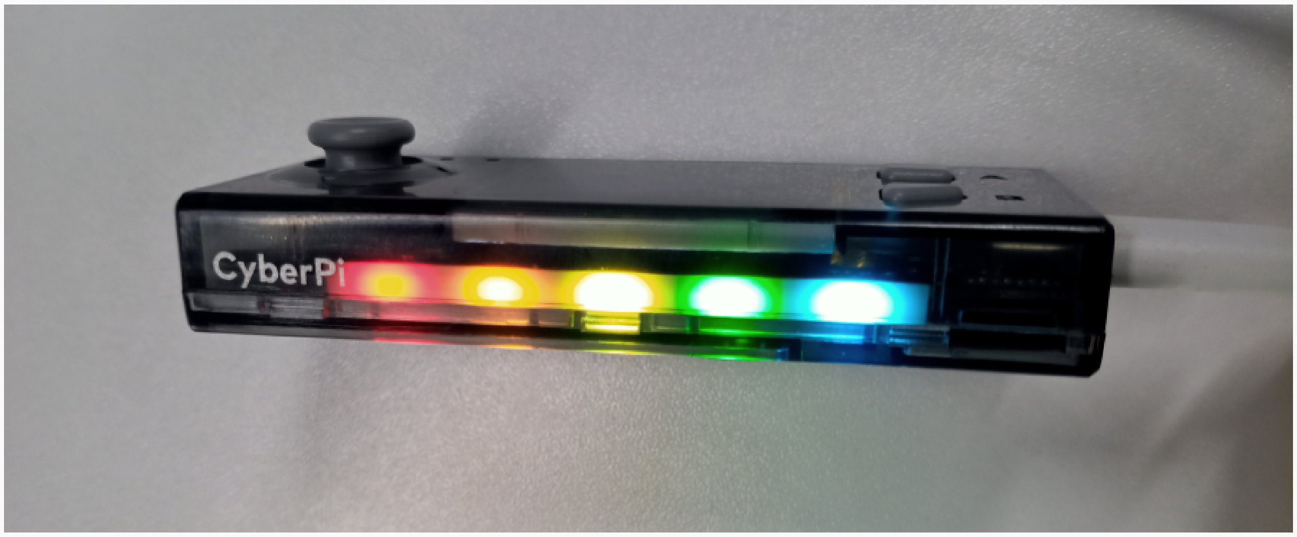
Step 1: Make sure the **Mode Switch** is pulled to "**Live**"



Step 2: Drag out a simple program to test CyberPi (here I just drag out the display[][][][][] block to test the LEDs on CyberPi as example).

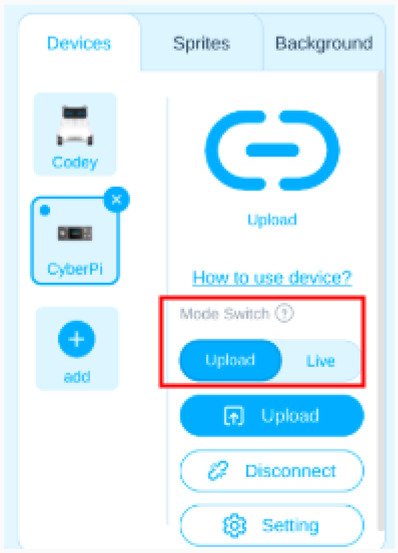


Step 3: Click on the program block and you will see the LED reaction on the CyberPi device.

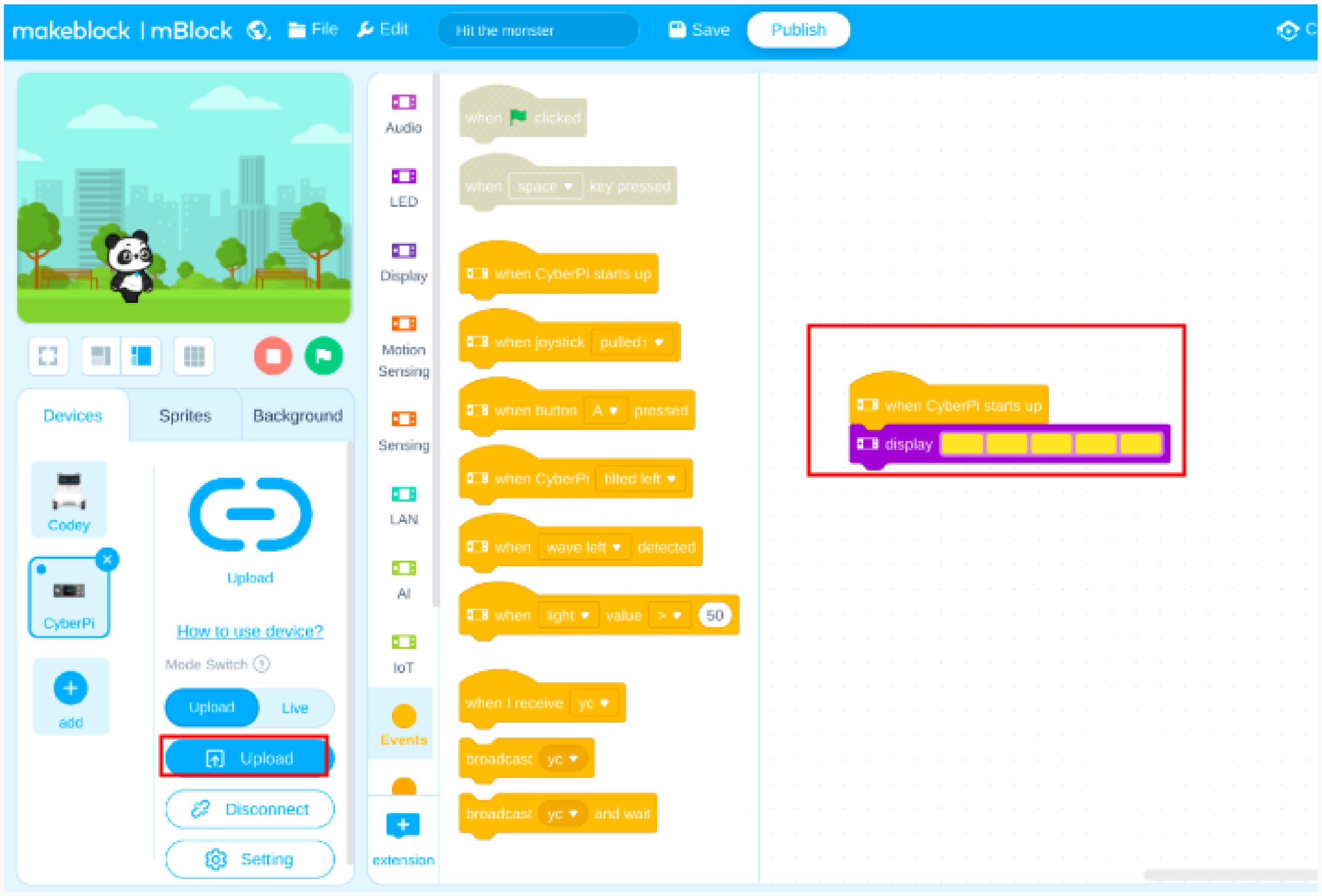


Upload mode

Step 1: switch the Mode Switch to "**Upload**".



Step 2: Edit a simple program (a program head from event catagory is necessary) and click on "**Upload**"



Step 3: Now you will see the LED reaction on the CyberPi device after the program uploaded successfully.

CyberPi

Related Education System

Coming soon!

Electronic Parts and Components

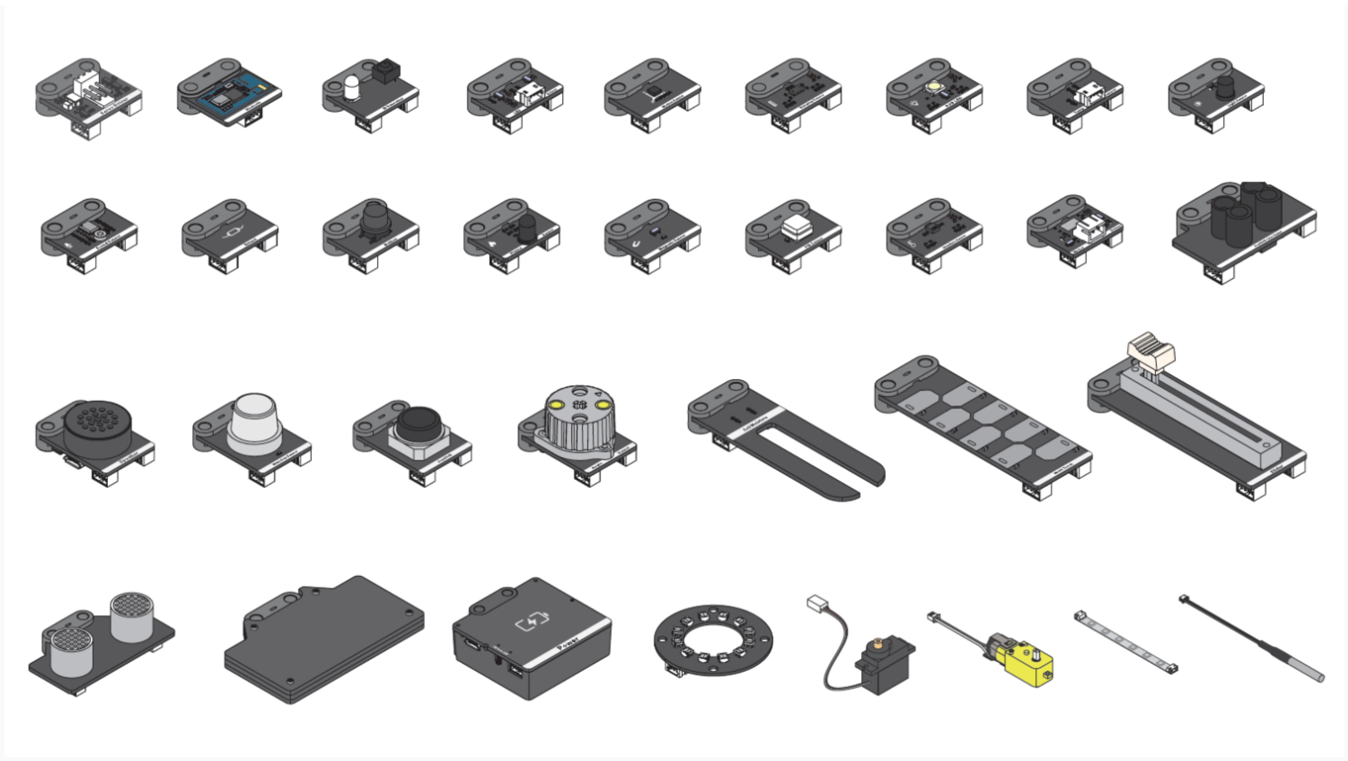
Main control board

CyberPi

Extension boards

Pocket Shield

Electronic modules


mBuild modules

Motors

5V TT motor	5V 9g servo
6~12V smart servo	12V encoder motor
Brushless motor	Stepper motor

Cables

Type-C USB cable

4-pin cable

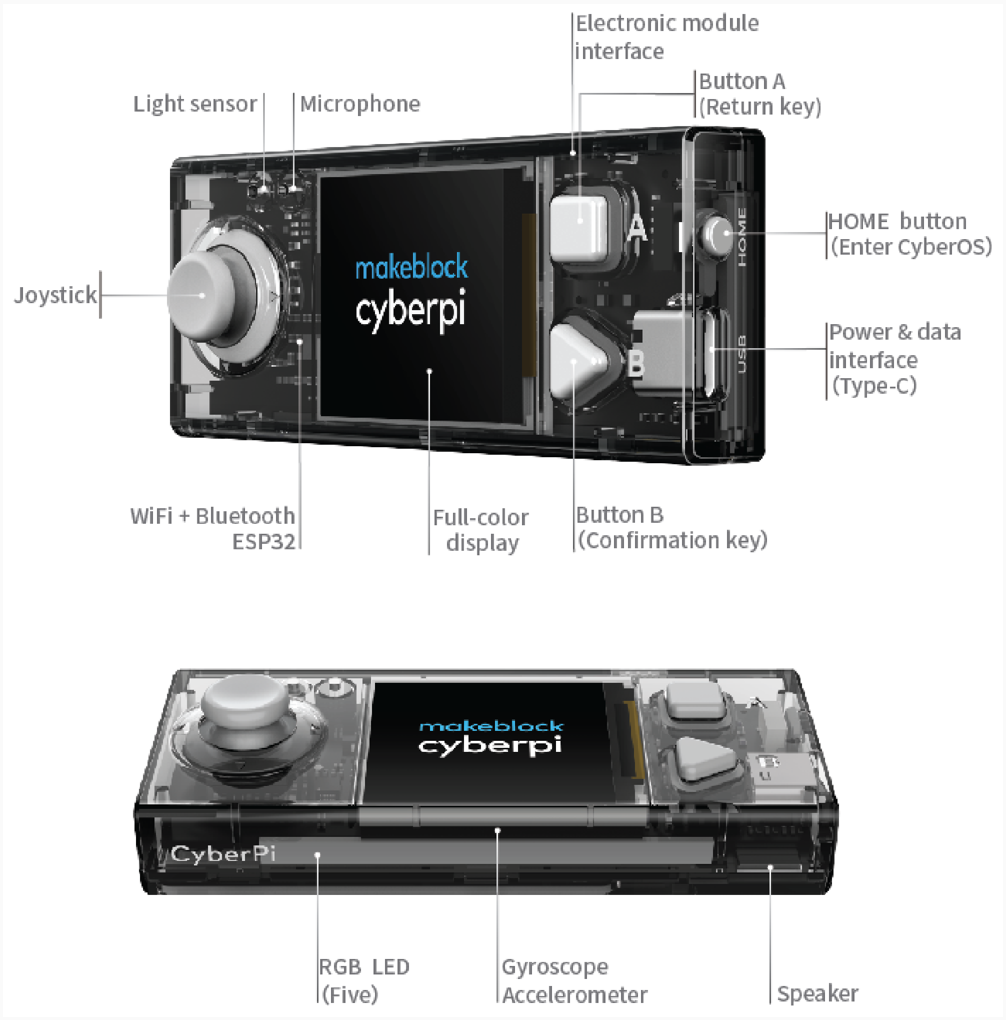
3-pin cable

2-pin cable

CyberPi

Overview

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



Technical specifications

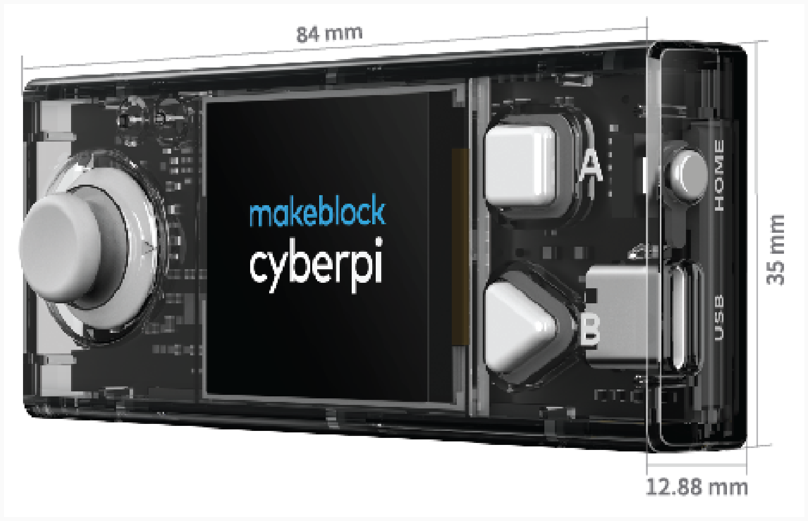
Name		CyberPi
Chip		ESP32-WROVER-B
Processor	Main processor	Xtensa® 32-bit LX6 dual-core
	Clock frequency	240 MHz
Onboard memory	ROM	448 KB
	SRAM	520 KB
Extended memory	SPI Flash	8 MB
	PSRAM	8 MB
Operating system		CyberOS, developed independently by Makeblock

Wireless communication	Wi-Fi Dual-mode Bluetooth
Physical interfaces	Micro USB port (Type-C) Port for connecting extension boards Port for connecting electronic modules (serial communication)
Hardware version	V1.0
Dimensions	84 mm × 35 mm × 13 mm (width × height × depth)
Weight	36 g

Features

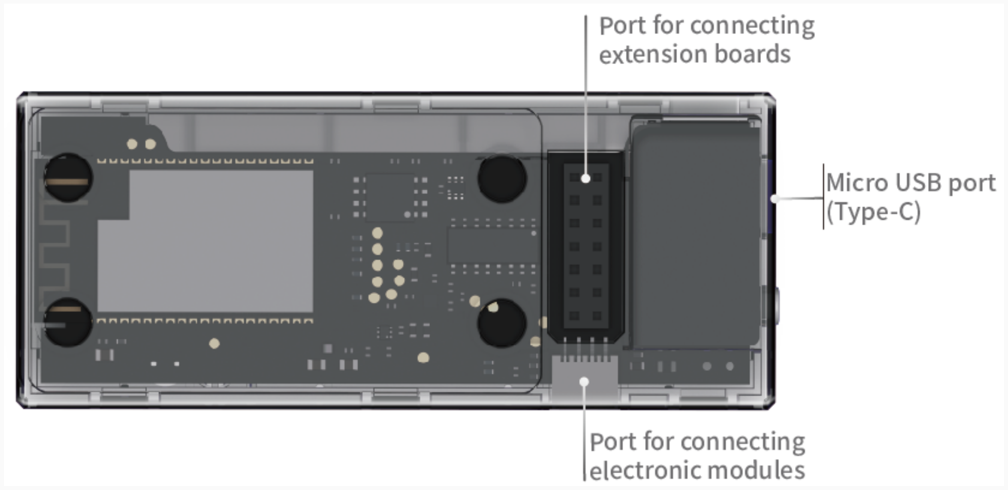
- Full-color display, providing user-friendly UIs for human-machine interaction
- CyberOS system, allowing you to execute the predefined programs, set the system language, and update the system through the onboard joystick and buttons
- One Micro USB port for connecting to PCs for power supply and communication
- 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
- 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

Dimensions



Interface description

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


Port for connecting extension boards
Micro USB port (Type-C)
Port for connecting electronic modules

Micro USB port (Type-C)

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

Port for connecting to extension boards

You can easily connect CyberPi to an extension board through the extension board port. Currently, the extension board Pocket shield is 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.

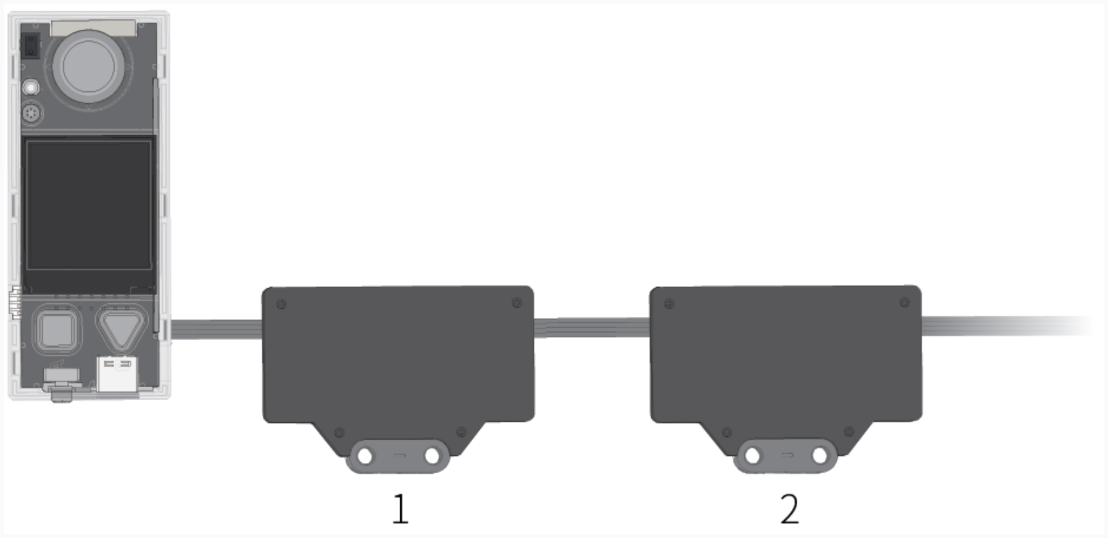
For more information, see "Pocket Shield."

Port for connecting electronic modules

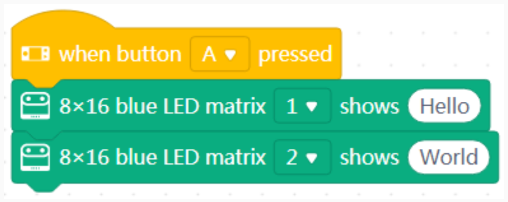
You can connect CyberPi to multiple electronic modules in series through the electronic module port.

CyberPi can intelligently identify the positions of the modules, which simplifies your programming. You don't have to set the information about the positions of the modules when you add or remove a module.

Example:
Connect CyberPi to multiple LED matrixes



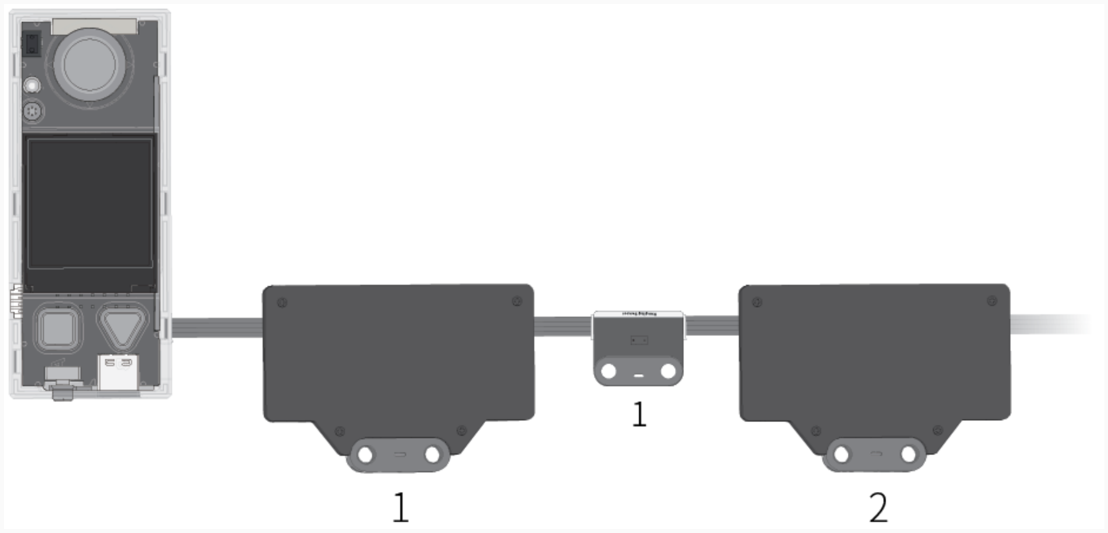
After connecting CyberPi to multiple LED matrixes, you need only to specify the place of an LED matrix among the ones connected instead of specifying the port to which the LED matrix is connected when compiling a program. As shown in the preceding figure, the first LED matrix connected to CyberPi is numbered 1, the second one numbered 2, and so on.



```
when button A pressed
8×16 blue LED matrix 1 shows Hello
8×16 blue LED matrix 2 shows World
```

When you press button A on CyberPi, the first LED matrix displays "Hello," and the second one displays "World."

Change the positions of the modules, as shown in the following figure.



The preceding program still works after you add a ranging sensor. When you press button A on CyberPi, the first LED matrix displays "Hello," and the second one displays "World."

Programming

You can use mBlock 5 to program CyberPi. mBlock 5 provides two editors, namely the block-based graphical editor (the default editor, referred to as mBlock 5) and Python editor (referred to as mBlock-Python Editor).

For details about programming, see "Programming Software."

Take CyberPi home

1. Contact the local dealer to purchase CyberPi series products and their educational packages.
2. Contact us to become our dealer.

More information

CyberPi Operation Guide

Pocket Shield Operation Guide

CyberPi Series User Manual

Python API Documentation for CyberPi

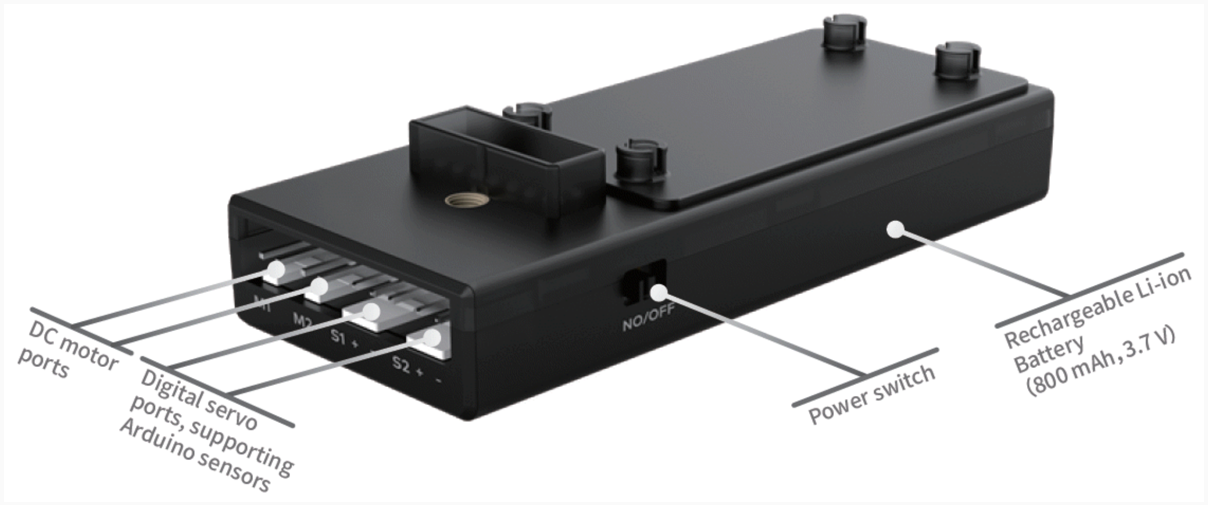
mBlock 5 Online Help

mBlock–Python Editor Online Help

Pocket Shield

Overview

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.


DC motor ports
Digital servo ports, supporting Arduino sensors
Power switch
Rechargeable Li-ion Battery (800 mAh, 3.7 V)
M1
M2
S1 +
S2 + -
NO/OFF

Features

- Built-in rechargeable Li-ion battery, used to supply power for CyberPi
- Two DC motor ports, used to connect and drive DC motors

- Two digital servo ports, used to connect and drive servos or LED strips
- One main control board port, allowing you to easily connect Pocket Shield to CyberPi
- 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

Programming

You can use mBlock 5 to program Pocket Shield. mBlock 5 provides two editors, namely the block-based graphical editor (the default editor, referred to as mBlock 5) and Python editor (referred to as mBlock-Python Editor).

For details about programming, see "Programming Software."

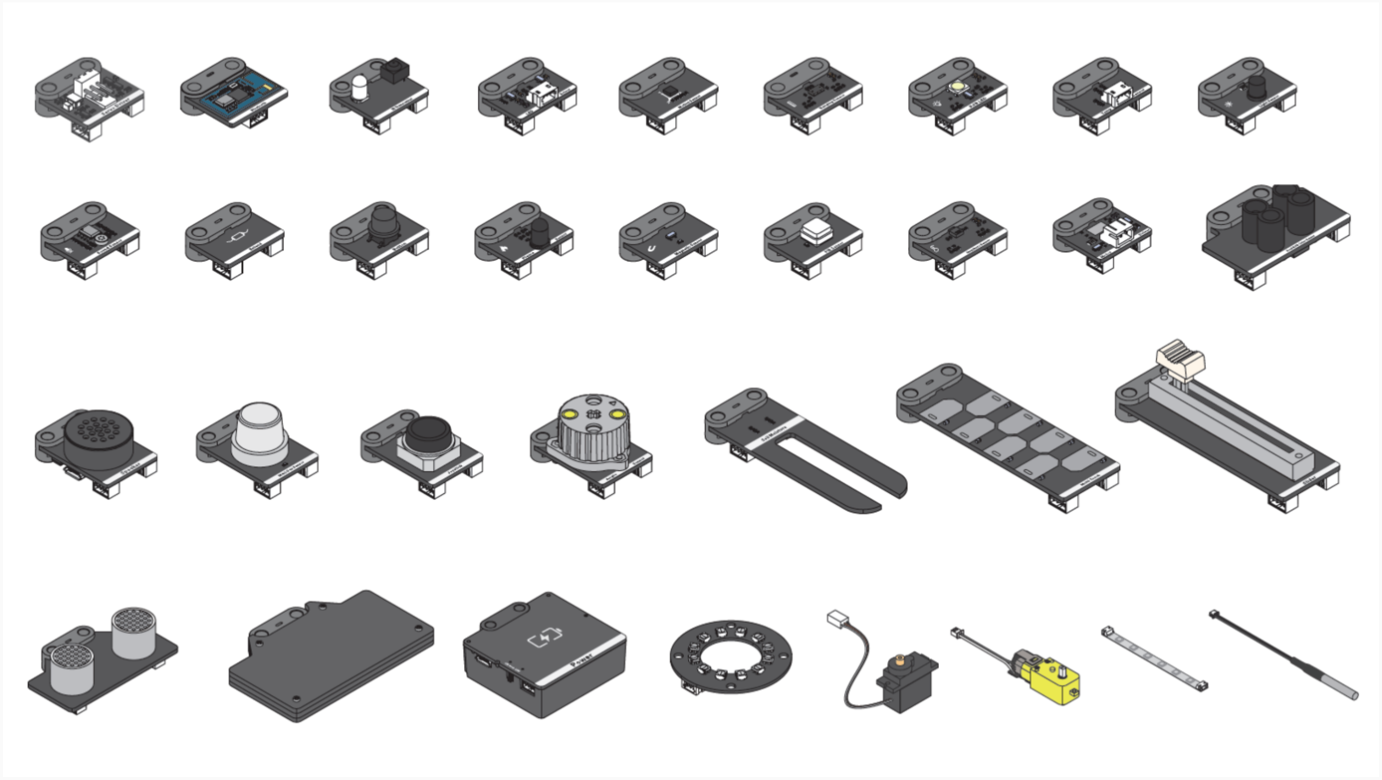
Take CyberPi home

1. Contact the local dealer to purchase CyberPi series products and their educational packages.
2. Contact us to become our dealer.

More information

CyberPi Operation Guide
Pocket Shield Operation Guide
CyberPi Series User Manual
Python API Documentation for CyberPi
mBlock 5 Online Help
mBlock-Python Editor Online Help

mBuild Modules



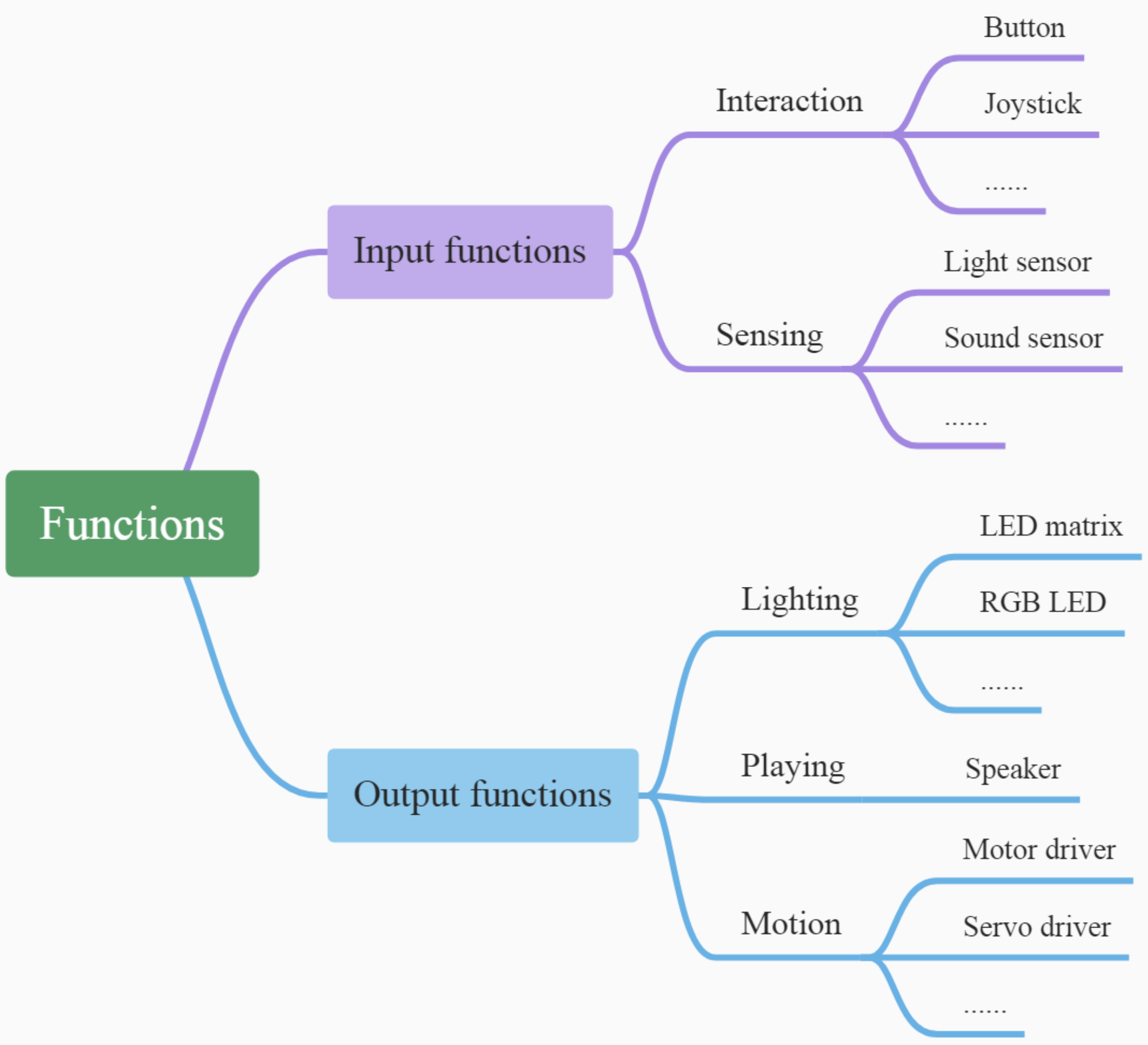
Developed by Makeblock, mBuild modules are small in size but highly intelligent in functions. They are compatible with almost all mainstream open-source hardware and are easy to use. Programming is not a necessity, but more complicated functions can be achieved through programming, for instance, using mBlock 5. Currently, over 30 mBuild electronic modules have been developed, and more modules will be available. You can mBuild modules to create projects based on your ideas, popularize programming education, teach programming, promote AI education, or participate in robotic competitions.

For details about each module, see Hardware Guide.

Functions and Features

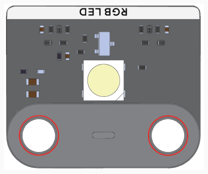
Functions

mBuild modules provide abundant input and output functions, allowing you to easily extend the functions of the supported main control boards, such as CyberPi and Halocode.



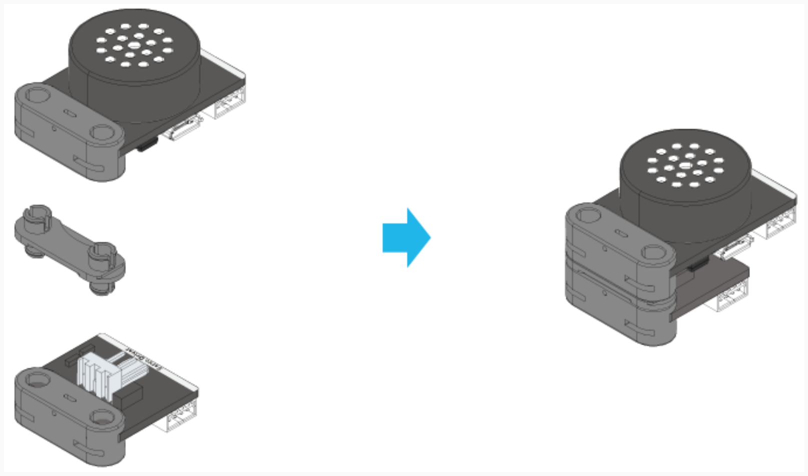
Structural characteristics

All mBuild modules are developed with two through-holes.

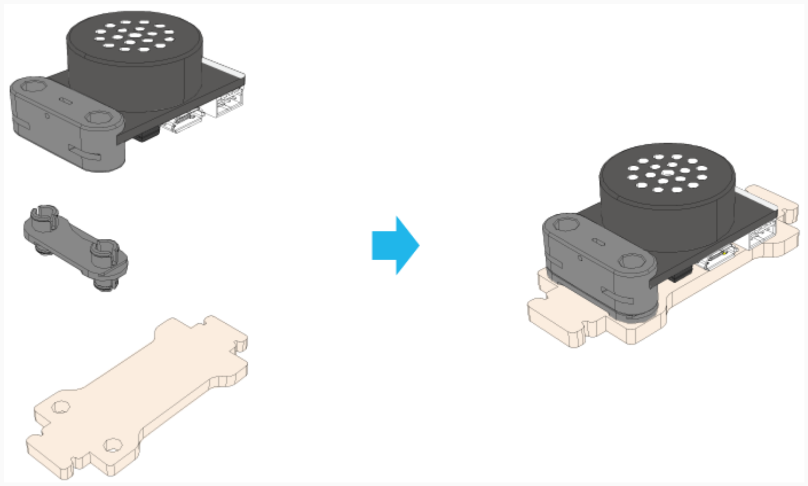


With the through-holes, mBuild modules can be mounted on each other with connectors, such as pins and screws. In addition, they can be connected to mechanical parts (M4), building blocks, and other structural parts through connectors.
The following provides some connection examples:

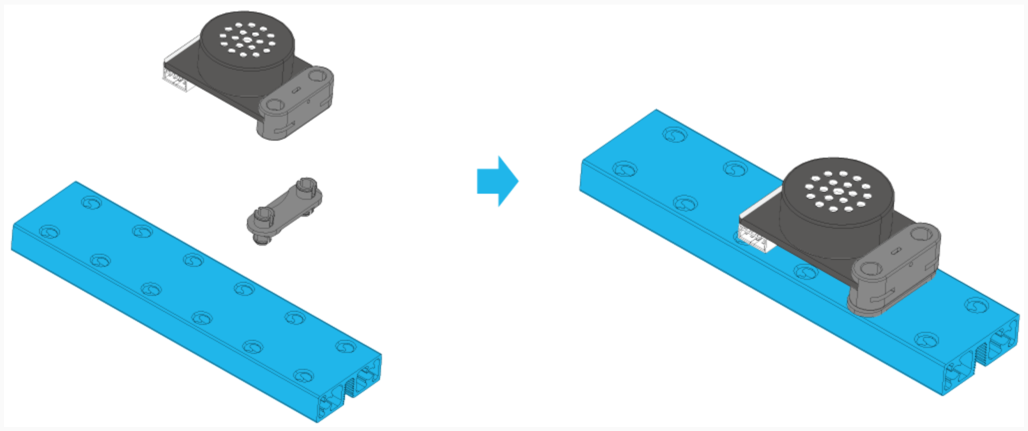
Example 1: Speaker + one pin + servo driver



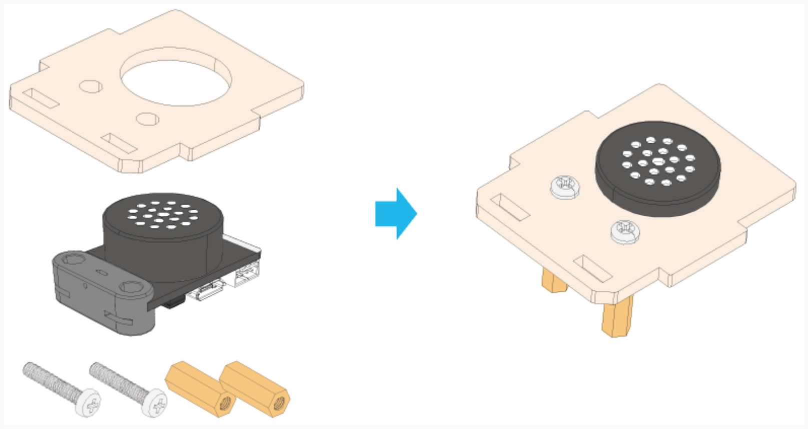
Example 2: Speaker + one pin + one cut board



Example 3: Speaker + one pin + one beam (M4)



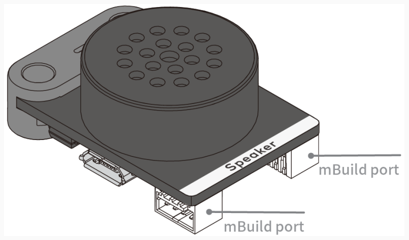
Example 4: Speaker + one cut board + two screws + two brass studs



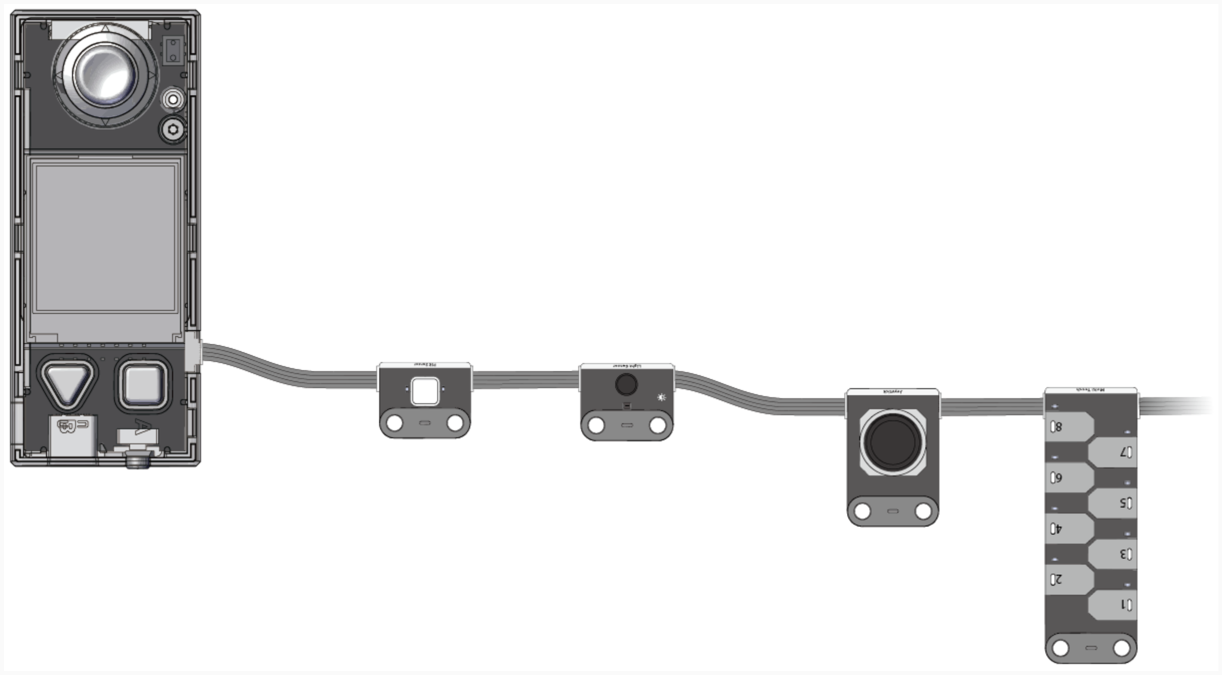
Electronic characteristics

mBuild ports

Most of the mBuild modules provide two mBuild ports.



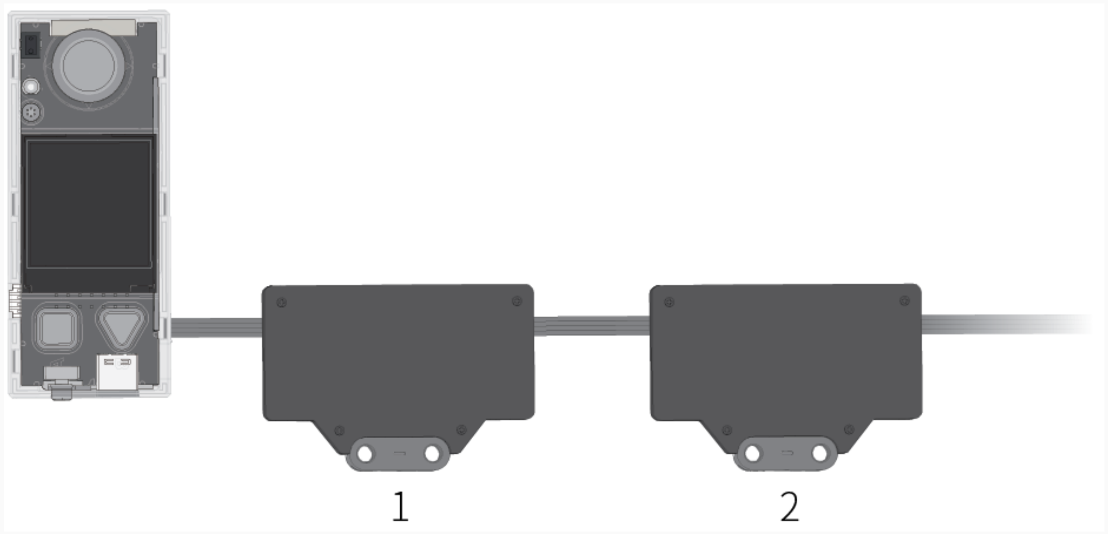
With the mBuild ports, multiple mBuild modules can be connected in series.



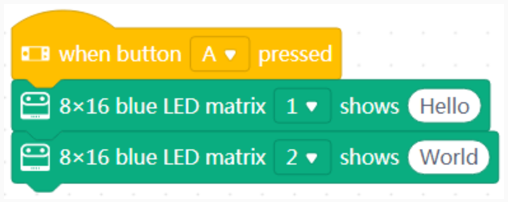
In addition, the positions of the modules can be intelligently identified when you program them on mBlock 5, which simplifies your programming. You don't have to set the information about the positions of the modules when you add or remove a module.

Intelligent position identification

Example:
Connect CyberPi to multiple LED matrixes



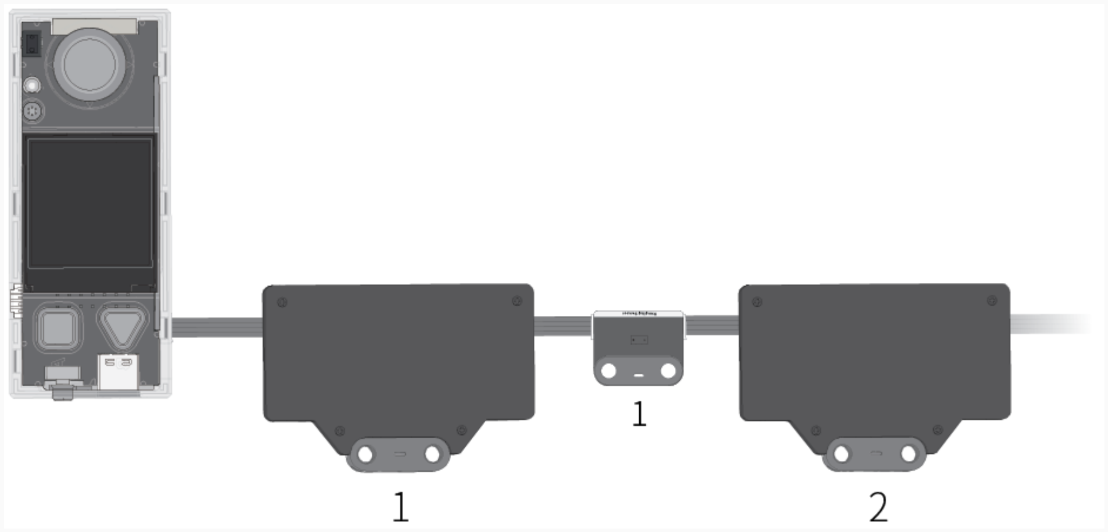
After connecting CyberPi to multiple LED matrixes, you need only to specify the place of an LED matrix among the ones connected instead of specifying the port to which the LED matrix is connected when compiling a program. As shown in the preceding figure, the first LED matrix connected to CyberPi is numbered 1, the second one numbered 2, and so on.



```
when button A ▾ pressed
8×16 blue LED matrix 1 ▾ shows Hello
8×16 blue LED matrix 2 ▾ shows World
```

When you press button A on CyberPi, the first LED matrix displays "Hello," and the second one displays "World."

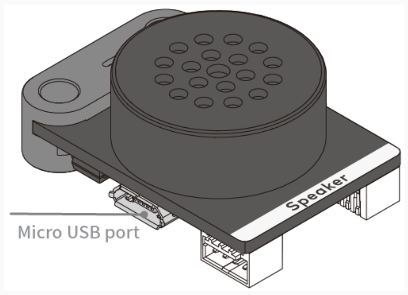
Change the positions of the modules, as shown in the following figure.



The preceding program still works after you add a ranging sensor. When you press button A on CyberPi, the first LED matrix displays "Hello," and the second one displays "World."

Micro USB port

Some of the mBuild modules, such as Speaker, Power Module, and Smart Camera, provide a Micro USB port, which allows a module to directly connect to a computer for power supply or communication.



For details about how to get started with mBuild modules, see Getting Started.

For details about the technical specifications and functions of each module, see mBuild Modules.

Getting Started

Thank you for choosing mBuild modules!

If you use mBuild modules for the first time, read this page carefully and follow the instructions so that you won't miss their functions.

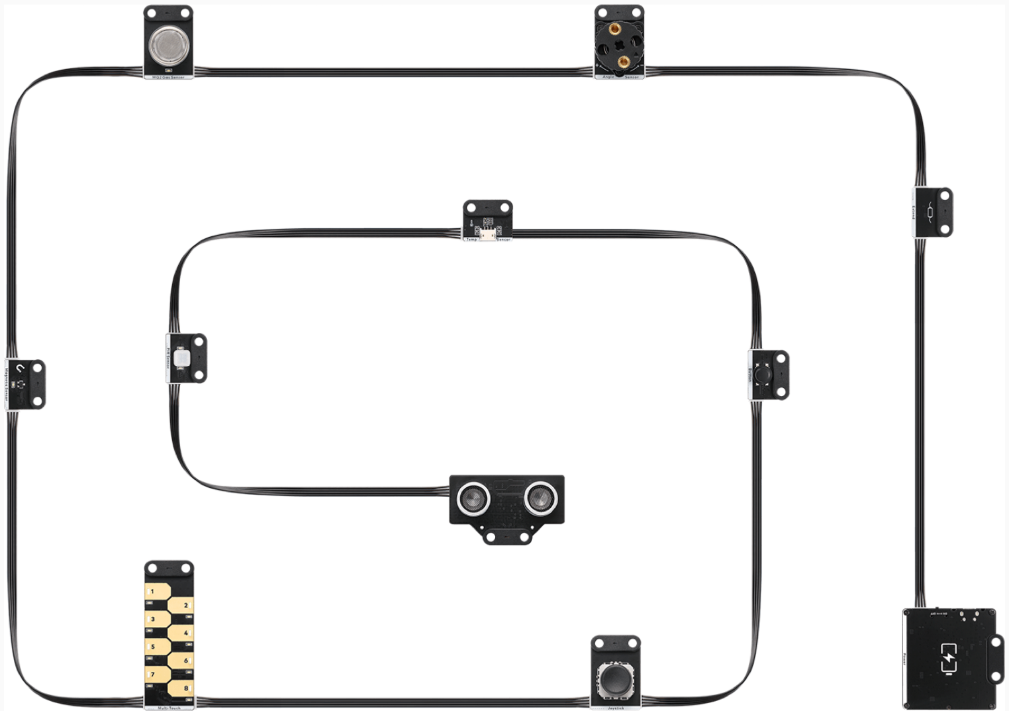
Before you use mBuild modules

Know about the wiring

Most of the mBuild modules provide two mBuild ports and therefore can be connected to main control boards or other mBuild modules through 4-pin cables. Some also provide a Micro USB port, which allows a module to directly connect to a computer for power supply or communication. For example, after connecting the speaker module to a computer, you can store audio files on it.

The following figure shows a 4-pin cable.

The following figure shows a connection example.

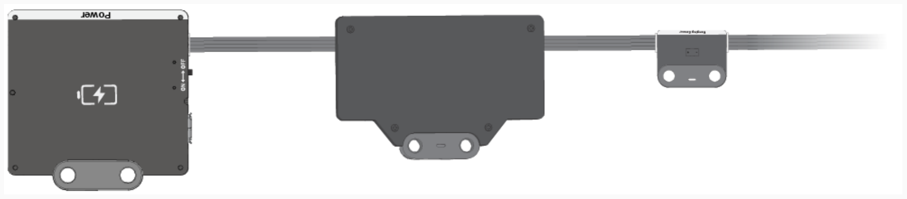


Power on mBuild modules

You can power on mBuild modules in one of the following ways:

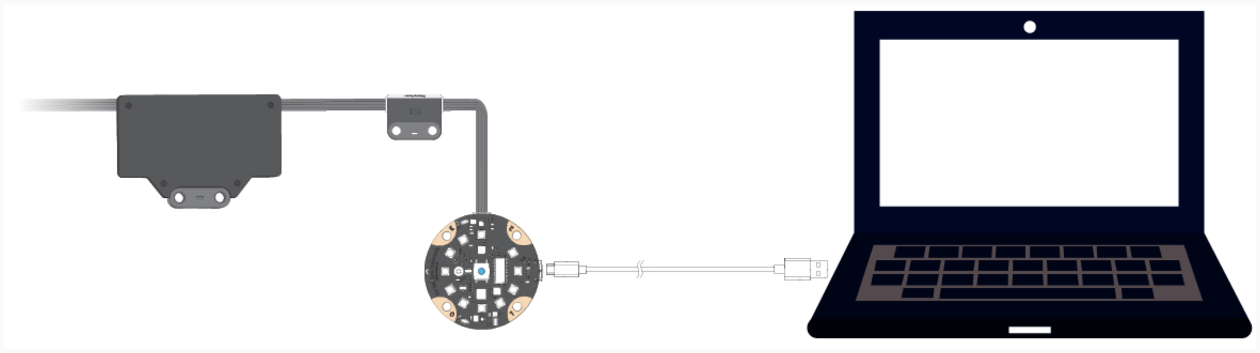
Way 1: use the power module

Connect the power module to the other modules through 4-pin cables. The power module is rechargeable. You can charge it by connecting it to a computer or charger.



Way 2: use a computer

Connect the mBuild modules to a main control board, such as Halocode or CyberPi, through 4-pin cables and then connect the main control board to a computer.



Start programming

With mBlock 5, you can program mBuild modules to implement their input or output functions. mBuild modules can be programmed as a device or as extension modules of a main control board.

Obtain mBlock 5

Select the version applicable to your device. For example, if you use a PC, you can download the version for PCs or use mBlock 5 on the web.

- For PCs

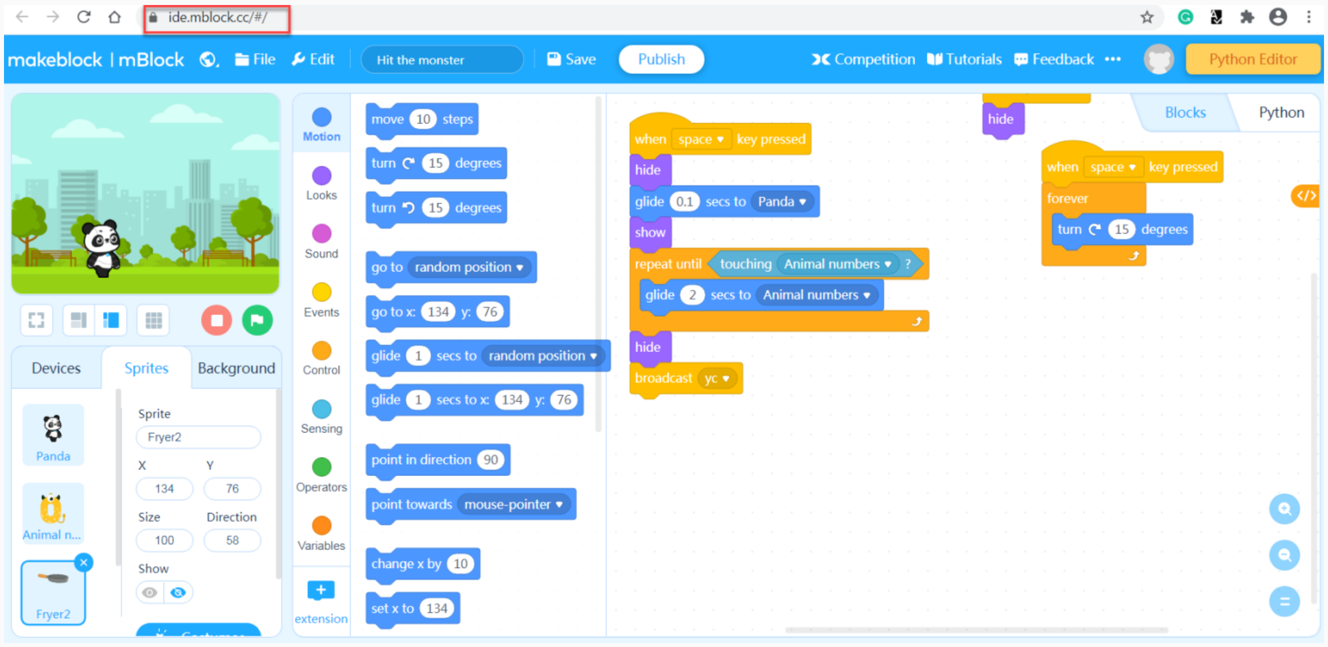
- On the Web (mLink Required)
- On Chromebook (mLink Required)
- For Android and iOS

Block-based programming

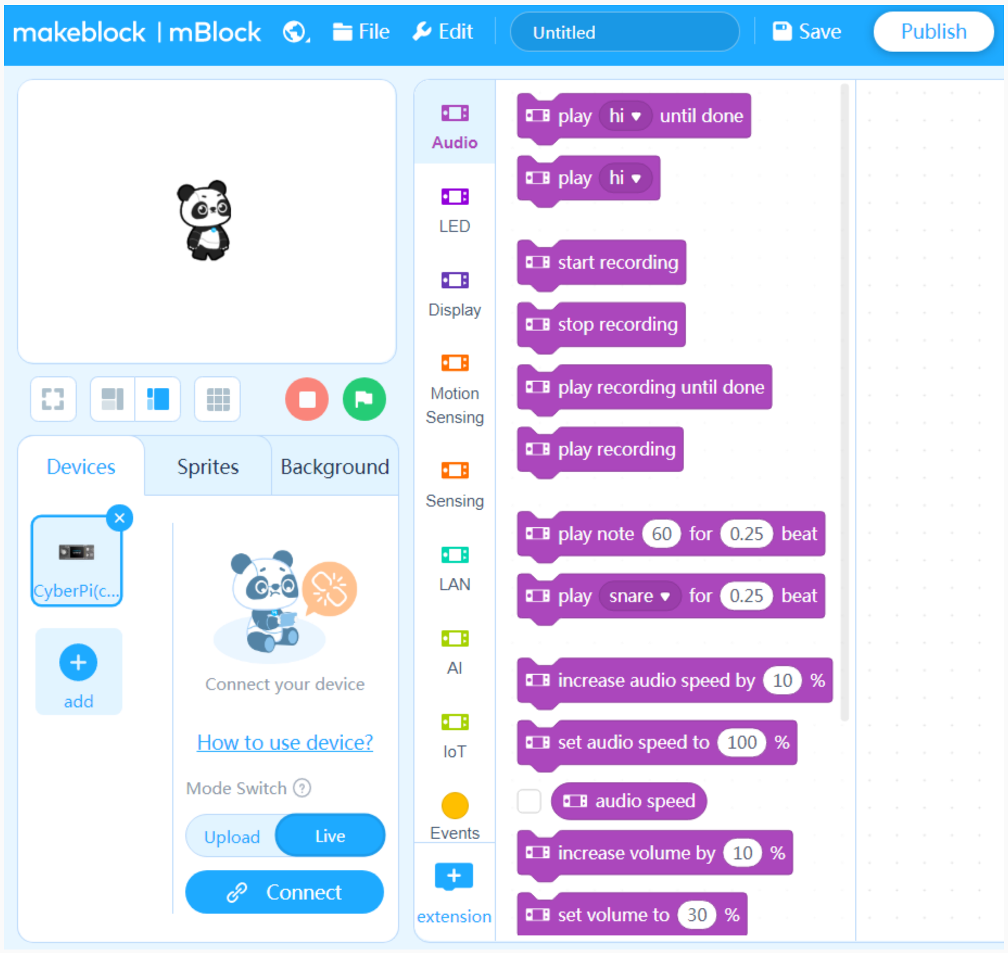
Program mBuild modules as a device

In the following steps, mBlock 5 on the web is used.

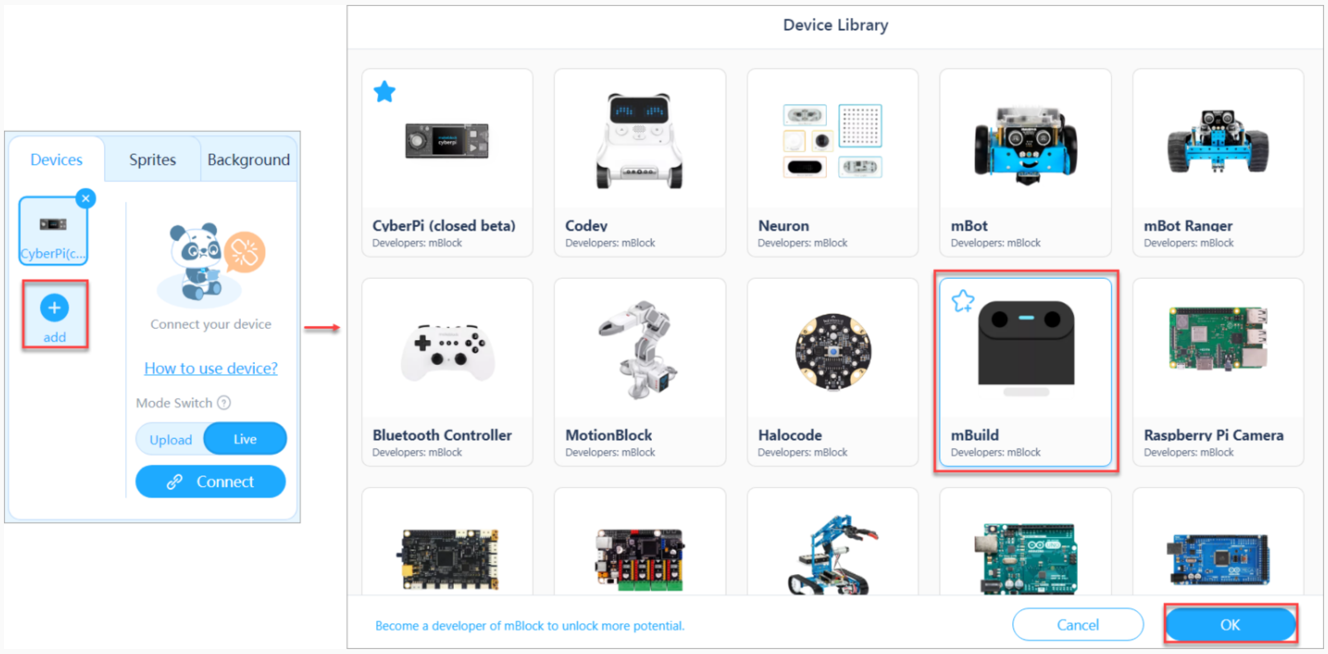
1. Open the web version of mBlock 5 on your PC.



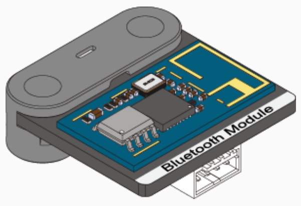
Choose **File** > **New** to start a new project.



2. Click **+ add** on the **Devices** tab and add **mBuild** from the device library.



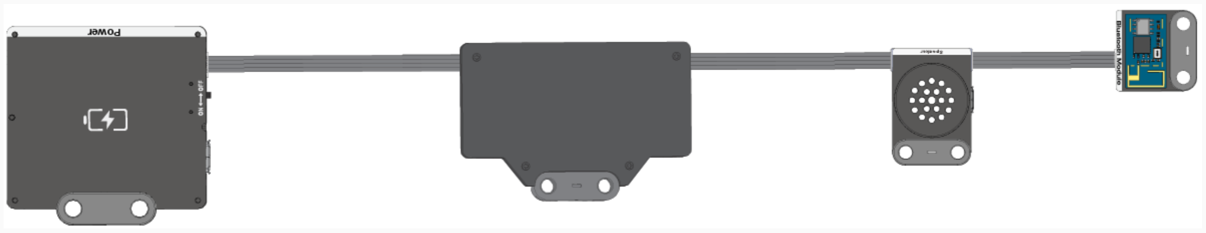
3. Connect the mBuild modules to mBlock 5 through the Bluetooth module.



Note:

To program mBuild modules as a device, you must connect them to mBlock 5 through the Bluetooth module.

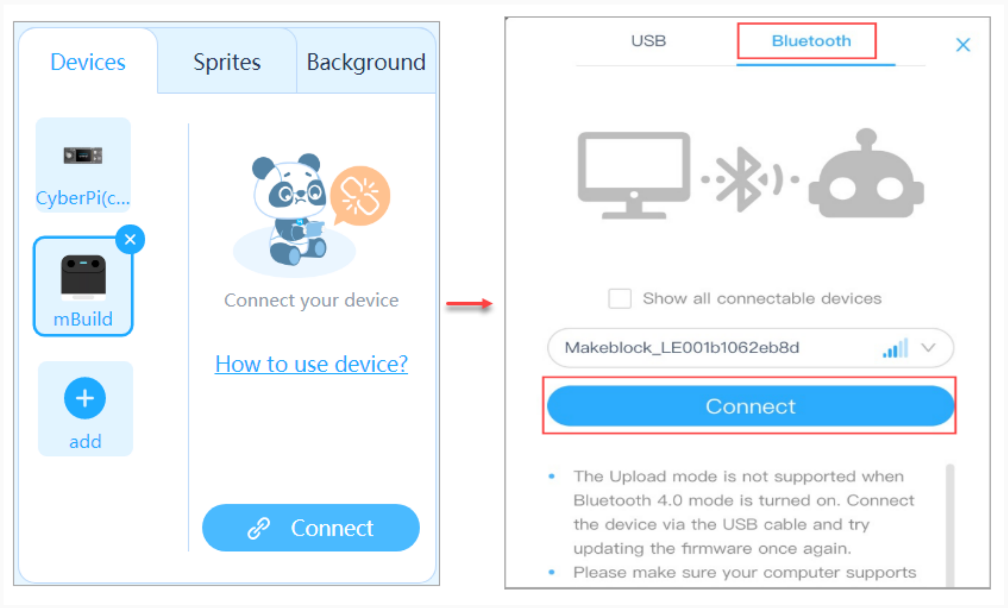
(1) Connect the modules to be programmed in series and connect the Bluetooth module last.



(2) Turn on the power module, enable the Bluetooth function of your computer, and then place the module close to the computer until the indicator on the Bluetooth module keeps on.

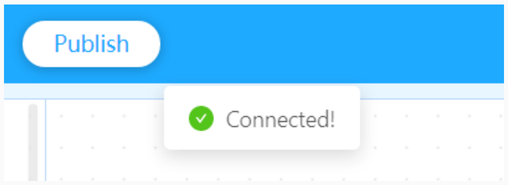
Note: Currently, the mBlock 5 web version on Windows doesn't support the connection of the mBuild Bluetooth module to it through system Bluetooth. If your PC runs Windows or if it runs another system but the Bluetooth module can't be connected to mBlock 5 properly, you can use the mBlock 5 PC client (mBlock V5.2.0 recommended) or Makeblock Bluetooth dongle. For details about how to use Makeblock Bluetooth dongle, see the section "Connection through Makeblock Bluetooth dongle" in the mBlock 5 Online Help.

(3) Click **Connect**, choose **Bluetooth** in the dialog box that appears, and then click **Connect**.

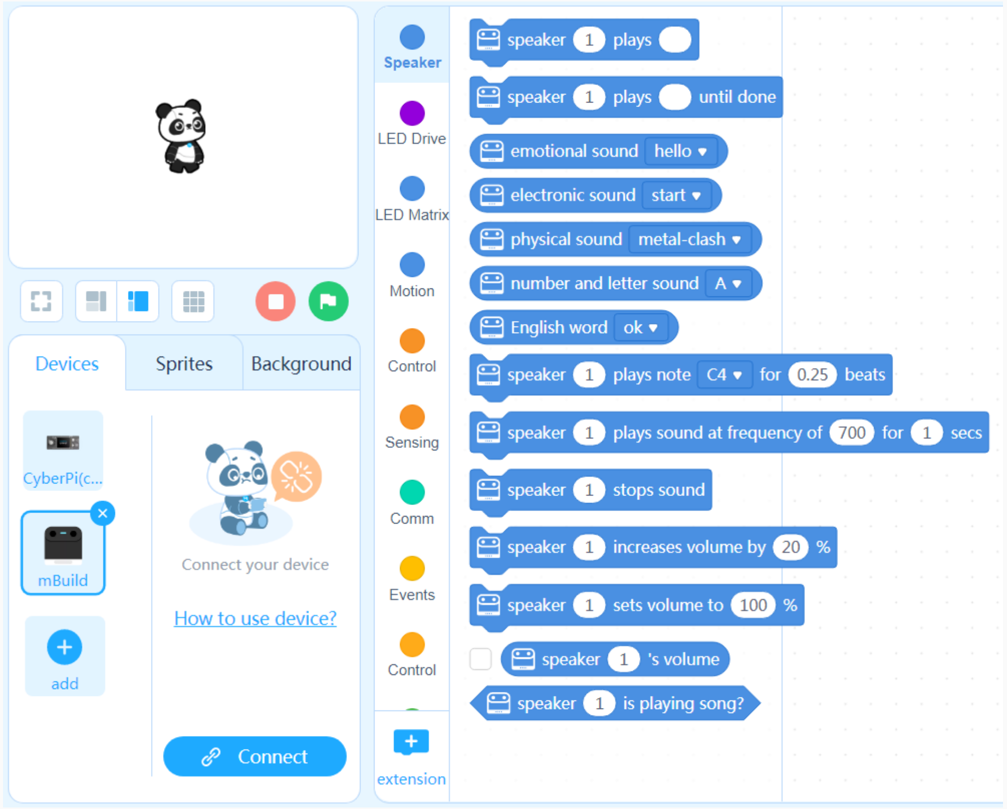


- The Upload mode is not supported when Bluetooth 4.0 mode is turned on. Connect the device via the USB cable and try updating the firmware once again.
- Please make sure your computer supports

A message is displayed after the mBuild modules are connected, indicating that the connection is successful.



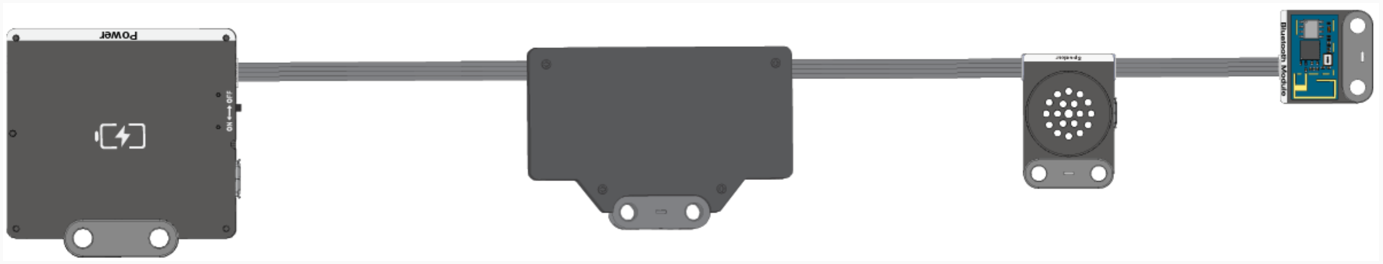
Now, you can drag and drop blocks to the scripts area to compile your program!

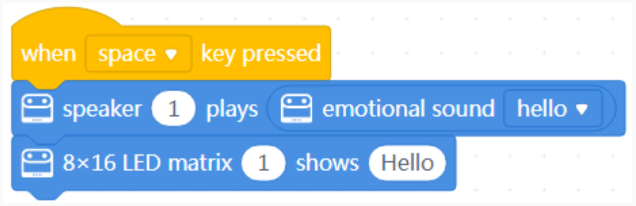


speaker (1) plays ( )
speaker (1) plays ( ) until done
emotional sound [hello ▾]
electronic sound [start ▾]
physical sound [metal-clash ▾]
number and letter sound [A ▾]
English word [ok ▾]
speaker (1) plays note [C4 ▾] for (0.25) beats
speaker (1) plays sound at frequency of (700) for (1) secs
speaker (1) stops sound
speaker (1) increases volume by (20) %
speaker (1) sets volume to (100) %
speaker (1) 's volume
speaker (1) is playing song?

Example programs

Example 1

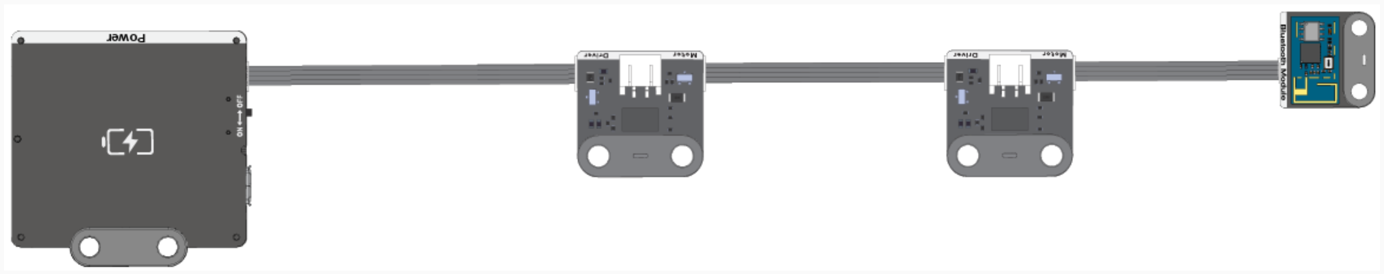


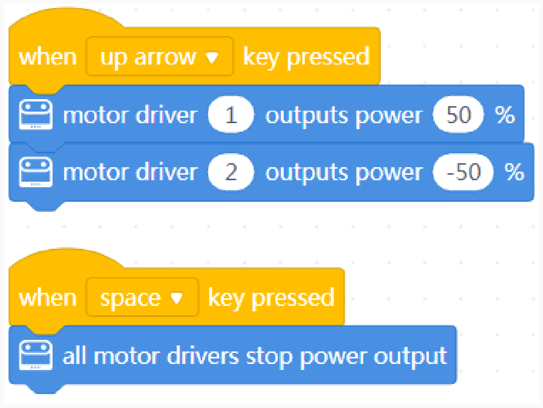


```
when [space ▾] key pressed
speaker (1) plays (emotional sound [hello ▾])
8×16 LED matrix (1) shows (Hello)
```

When you press the space key, the speaker says "hello," and the LED matrix displays **Hello**.

Example 2





```
when [up arrow ▾] key pressed
motor driver (1) outputs power (50) %
motor driver (2) outputs power (-50) %

when [space ▾] key pressed
all motor drivers stop power output
```

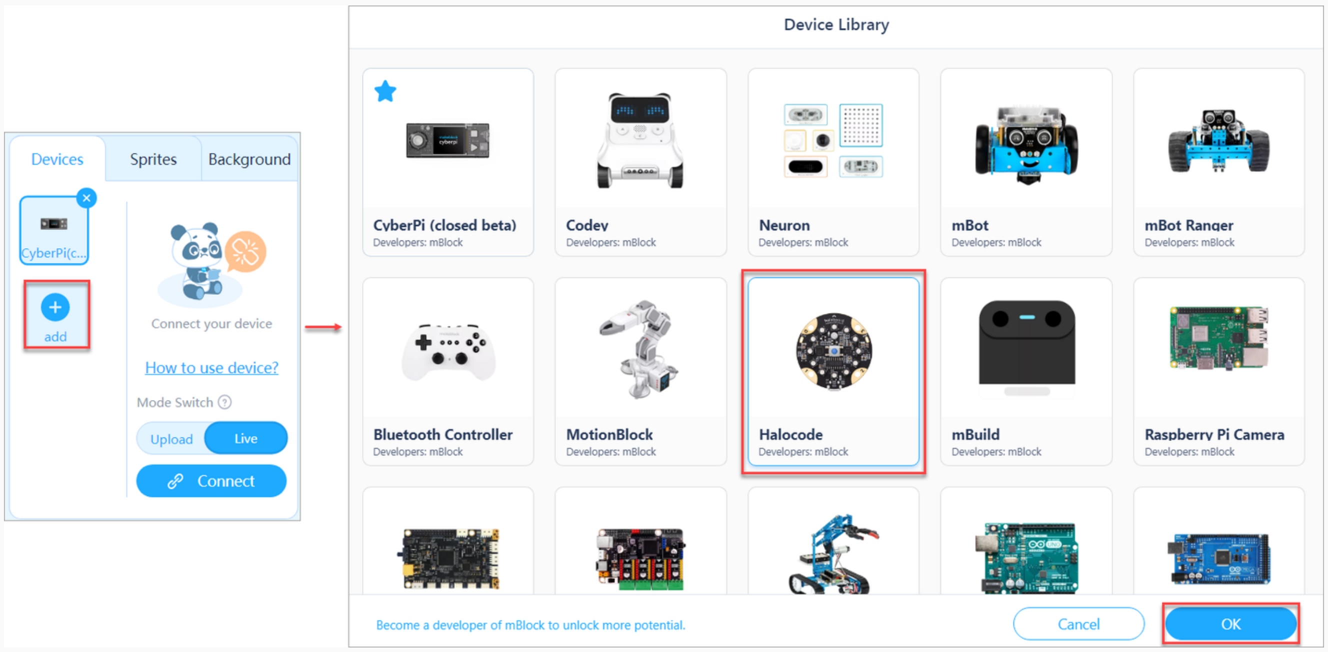
When you press the key, the two motor drivers provide power; and when you press the space key, they stop providing power.

Tips: After connecting the motor drivers to motors, you can control the motion of the motors through programming.

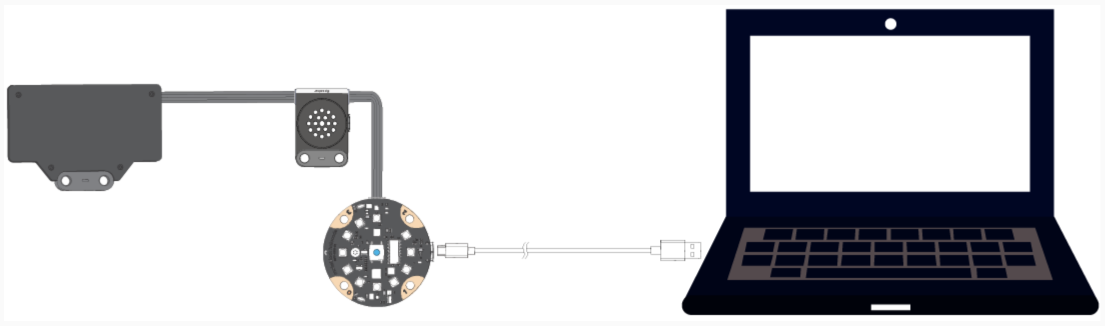
Program mBuild modules as extension modules of a main control board

In the following steps, Halocode is used as the main control board.

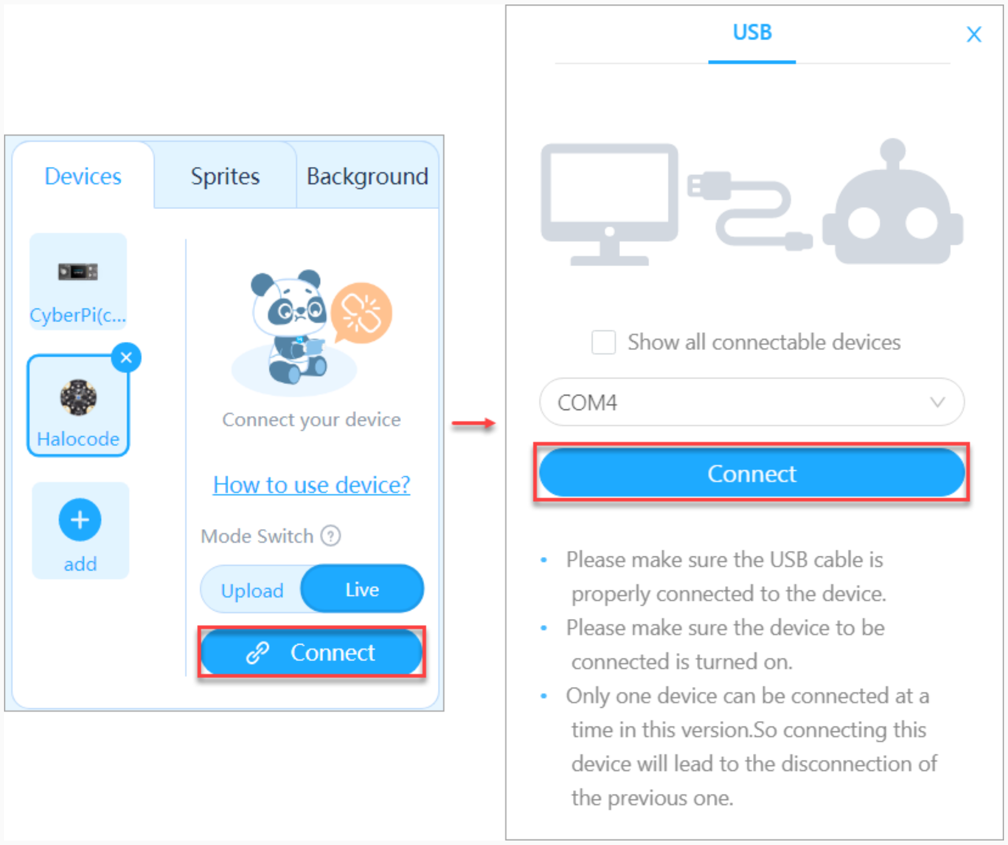
1. Click **+ add** on the **Devices** tab and add **Halocode** from the device library.



2. Connect Halocode and the mBuild modules to your computer through a Micro USB cable.

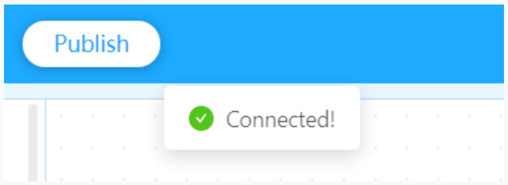


3. Click **Connect** and then click **Connect** in the dialog box that appears.



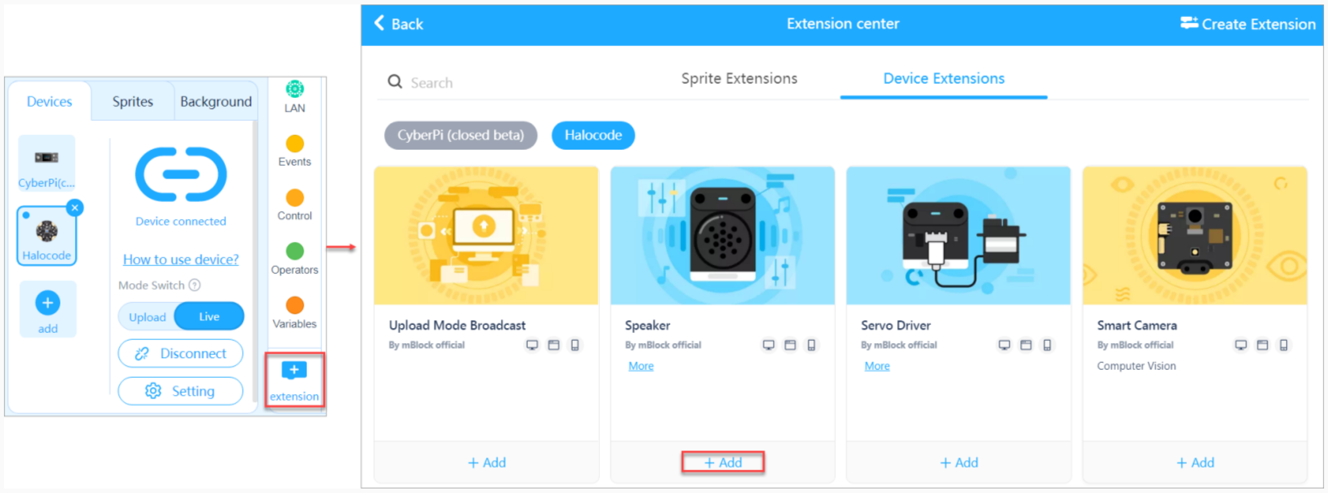
- Please make sure the USB cable is properly connected to the device.
- Please make sure the device to be connected is turned on.
- Only one device can be connected at a time in this version.So connecting this device will lead to the disconnection of the previous one.

A message is displayed after Halocode is connected, indicating that the connection is successful.

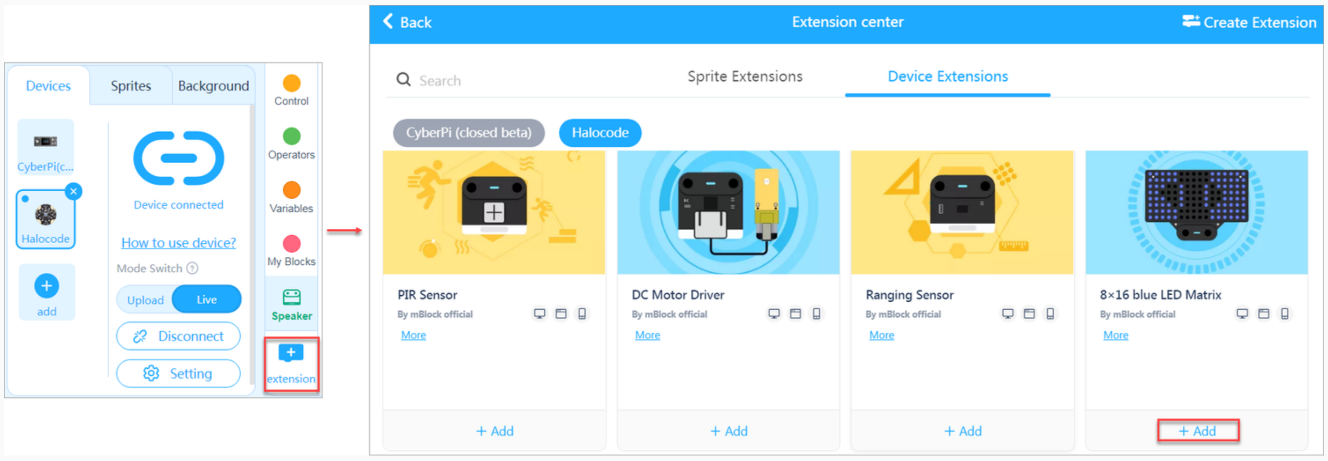


4. Add the Speaker and LED matrix extensions.

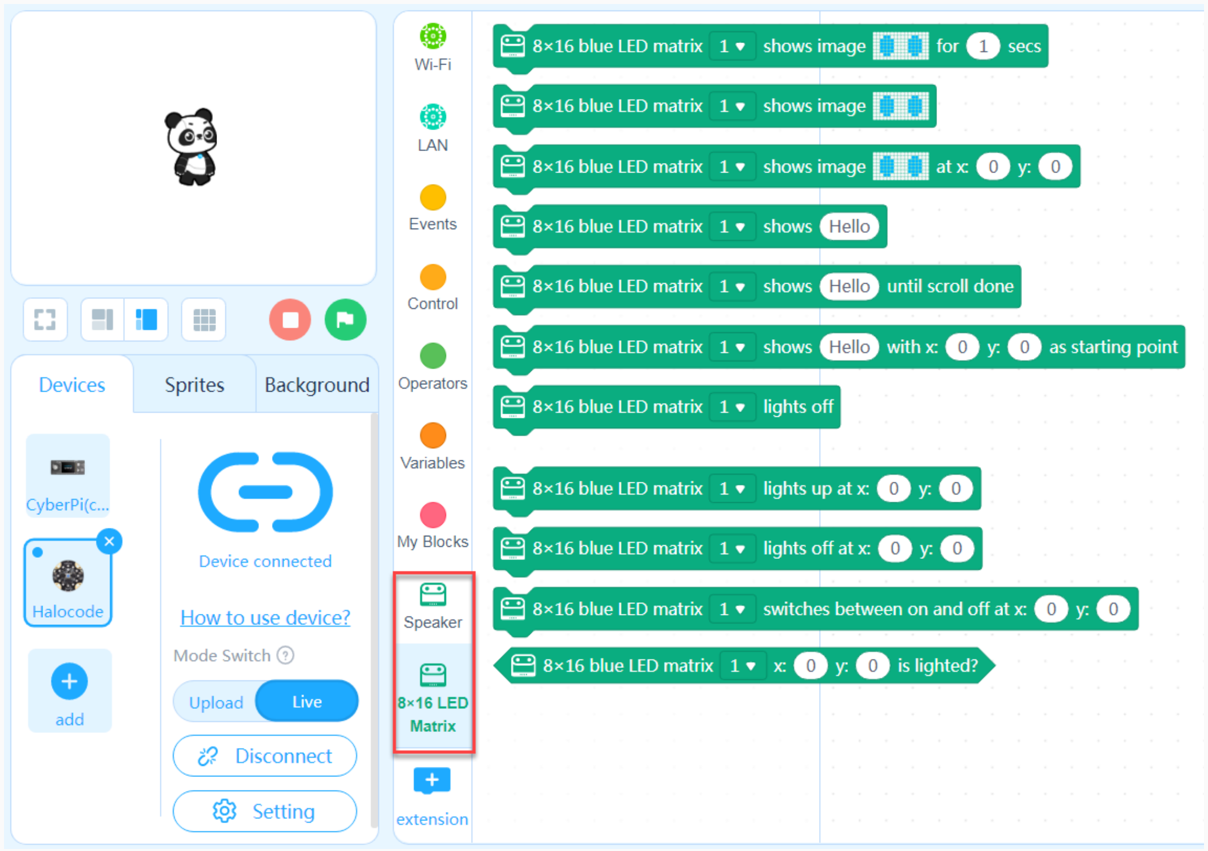
(1) Add the Speaker extension.



(2) Add the LED Matrix extension.



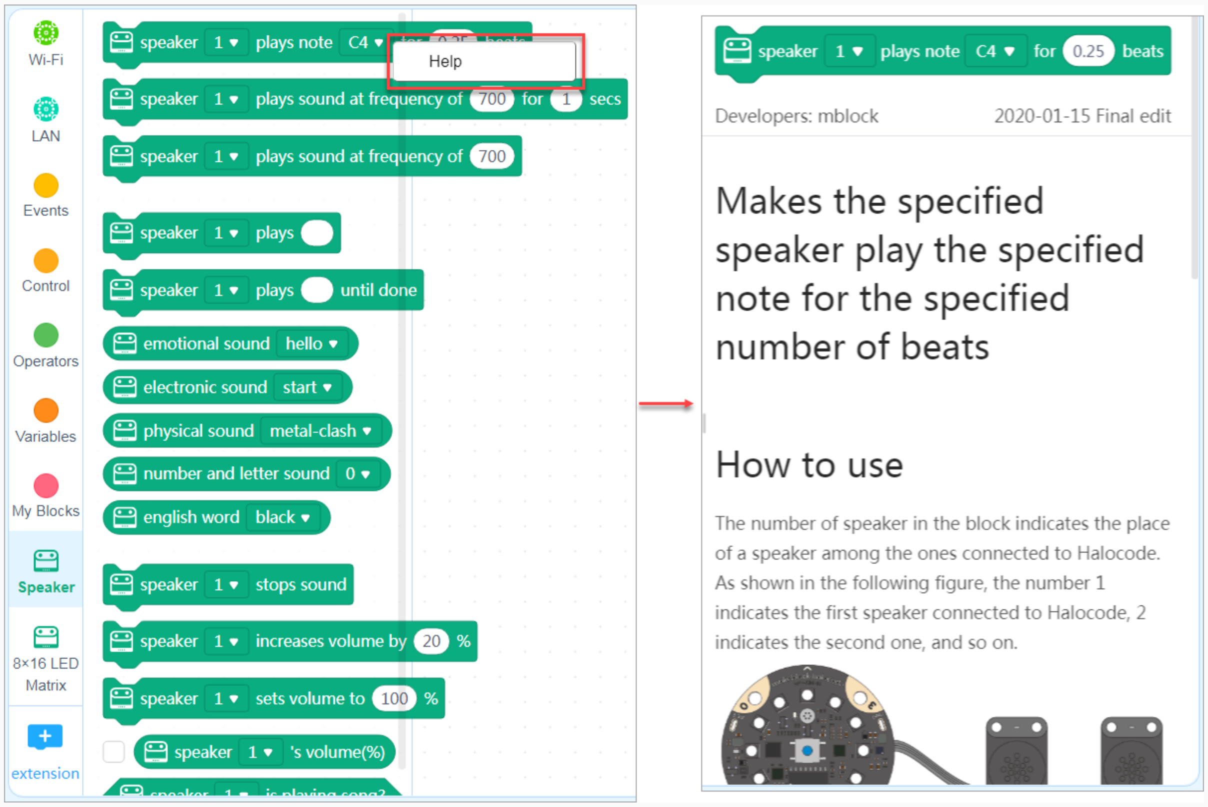
Now, you can drag and drop blocks to the scripts area to compile your program!



```
8×16 blue LED matrix 1 ▾ shows image [ ] for 1 secs
8×16 blue LED matrix 1 ▾ shows image [ ]
8×16 blue LED matrix 1 ▾ shows image [ ] at x: 0 y: 0
8×16 blue LED matrix 1 ▾ shows Hello
8×16 blue LED matrix 1 ▾ shows Hello until scroll done
8×16 blue LED matrix 1 ▾ shows Hello with x: 0 y: 0 as starting point
8×16 blue LED matrix 1 ▾ lights off
8×16 blue LED matrix 1 ▾ lights up at x: 0 y: 0
8×16 blue LED matrix 1 ▾ lights off at x: 0 y: 0
8×16 blue LED matrix 1 ▾ switches between on and off at x: 0 y: 0
8×16 blue LED matrix 1 ▾ x: 0 y: 0 is lighted?
```

Block help

If you don't understand a block when using it, you can right-click it and click **Help** that appears to view its help information.


Wi-Fi
LAN
Events
Control
Operators
Variables
My Blocks
Speaker
8×16 LED Matrix
extension
speaker 1 plays note C4
Help
speaker 1 plays sound at frequency of 700 for 1 secs
speaker 1 plays sound at frequency of 700
speaker 1 plays
speaker 1 plays until done
emotional sound hello
electronic sound start
physical sound metal-clash
number and letter sound 0
english word black
speaker 1 stops sound
speaker 1 increases volume by 20 %
speaker 1 sets volume to 100 %
speaker 1 's volume(%)
speaker 1 plays note C4 for 0.25 beats
Developers: mblock
2020-01-15 Final edit

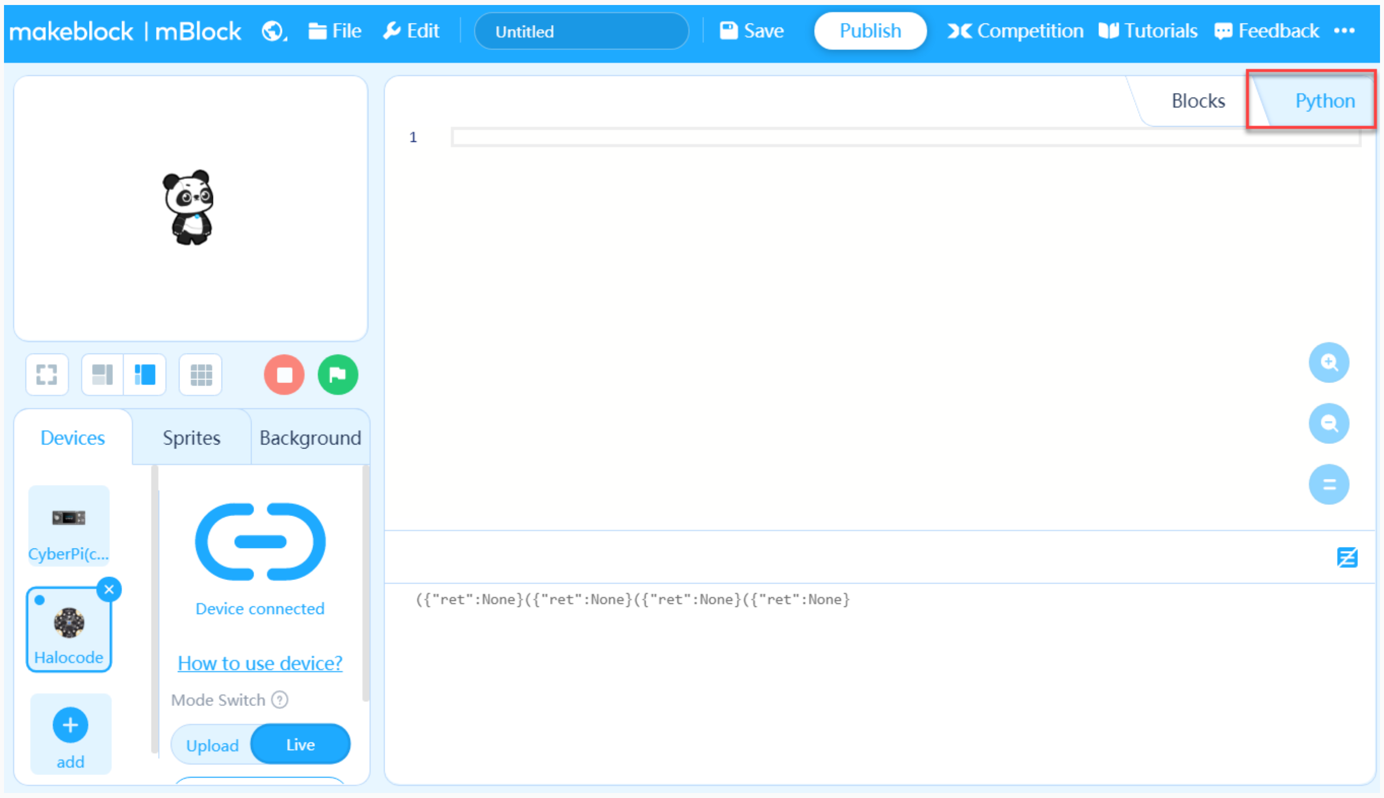
Makes the specified speaker play the specified note for the specified number of beats

How to use

The number of speaker in the block indicates the place of a speaker among the ones connected to Halocode. As shown in the following figure, the number 1 indicates the first speaker connected to Halocode, 2 indicates the second one, and so on.

Python programming

You can also use MicroPython to program mBuild modules.
Set the programming language to Python.



```
({"ret":None}({"ret":None}({"ret":None}({"ret":None}
```

For the Python API documentation of mBuild modules, see Python APIs for mBuild Modules.

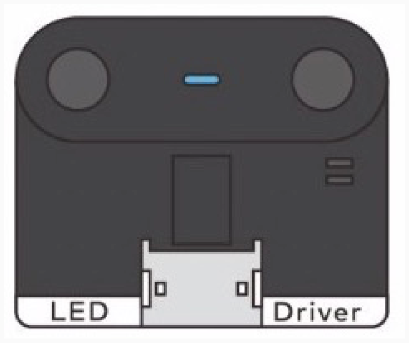
> **Tips:**
>
> You can select mBuild modules and mount or install them on structures made of cut boards or mechanical parts as required. For example, you can install the motor driver modules on a car made of mechanical parts.

More information

For the functions and features of the mBuild modules, see Functions and Features.
For details about the technical specifications and functions of each module, see mBuild Modules.

LED Driver

The LED driver can drive various light accessories, including LEDs, LED strips, and LED rings.



Parameters

Dimensions: 24 mm × 20 mm

Reference current: depending on the load

Supported accessories: Mini LED Strip and Ring

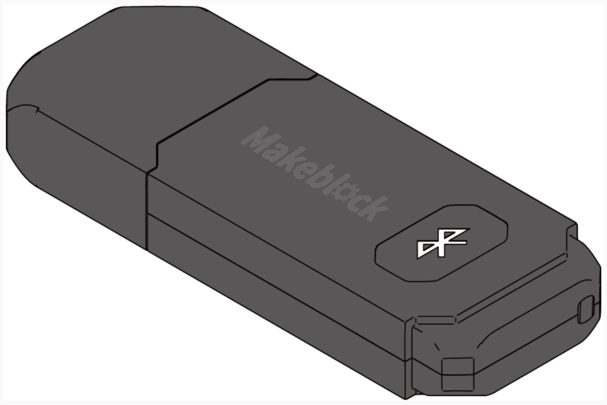
Wireless Adapter

Application scenarios

Your Makeblock device can't connect to mBlock 5 directly through Bluetooth if:

- you use mBlock 5 on a desktop computer without a Bluetooth adapter; or
- you use mBlock 5 on a laptop that is equipped with a Bluetooth adapter earlier than Bluetooth 4.0 or is not compatible with Makeblock devices due to driver exceptions.

To solve this problem, we provide the Makeblock wireless adapter for you. With it, you can connect your Makeblock devices, such as CyberPi, Codey Rocky, mBot of the Bluetooth version, and mBot Ranger, to mBlock 5 wirelessly, and thus debug the programs for your devices online and upload programs to CyberPi, Halocode, and Codey Rocky wirelessly.



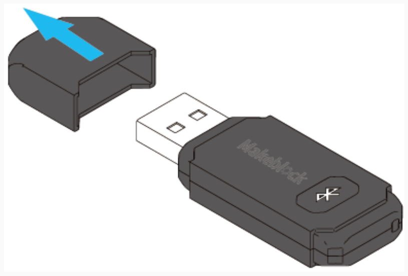
If you need to check whether your computer is equipped with the Blutooth adapter required for connecting Makeblock devices directly to your computer, see "Bluetooth Compatibility."

Indicator description

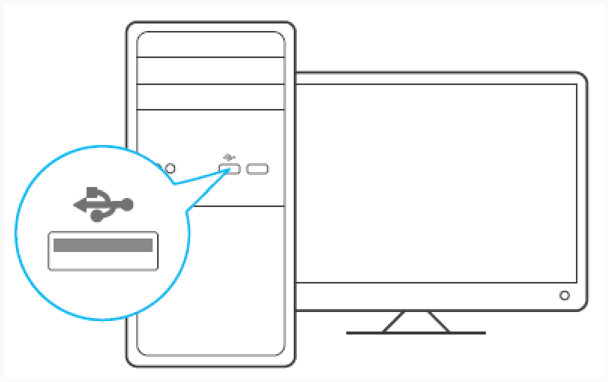
Blinks slowly	Standby	searches for the device it was paired with last time and automatically performs pairing
Blinks fast	Pairing	searches for a new device to be paired with it
Keeps on	Paired	

Hardware connection

1. Open the protection cap.



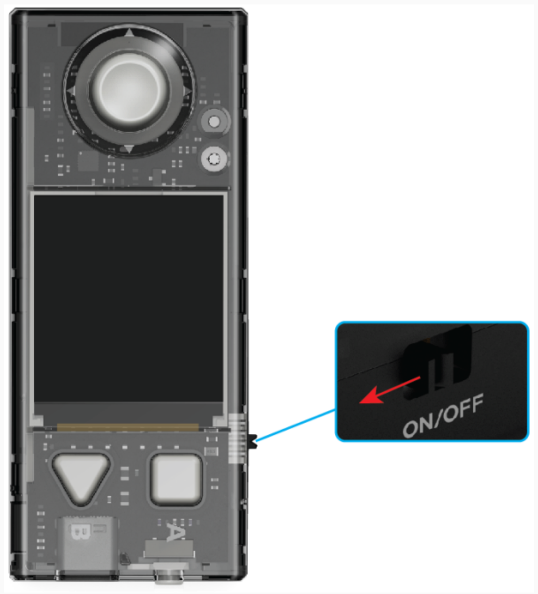
2. Insert the wireless adapter to a USB port of your computer.
   The indicator on it blinks slowly.



Pairing

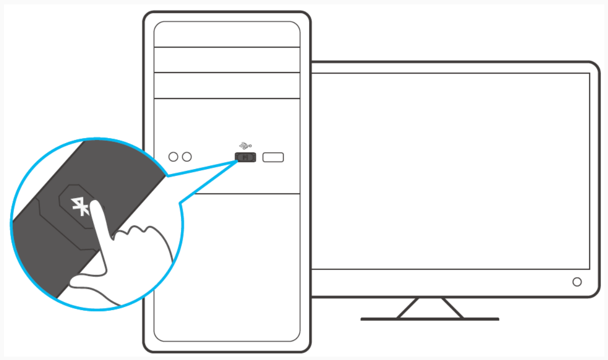
Use CyberPi as an example (in this example, Pocket Shield supplies power for CyberPi)

1. Turn on CyberPi.



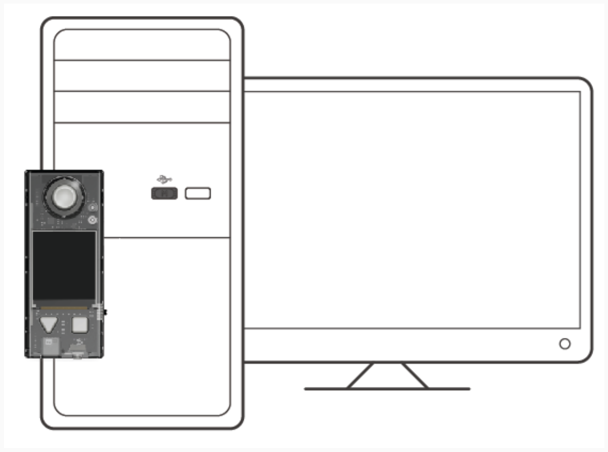
2. Press the button on the wireless adapter.

   The indicator on it blinks fast. The wireless adapter enters the pairing state.

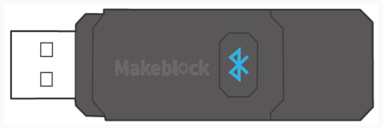


3. Place CyberPi close to the wireless adapter.

   The pairing is automatically performed.



The indicator on the wireless adapter keeps on, indicating that the pairing is successful.

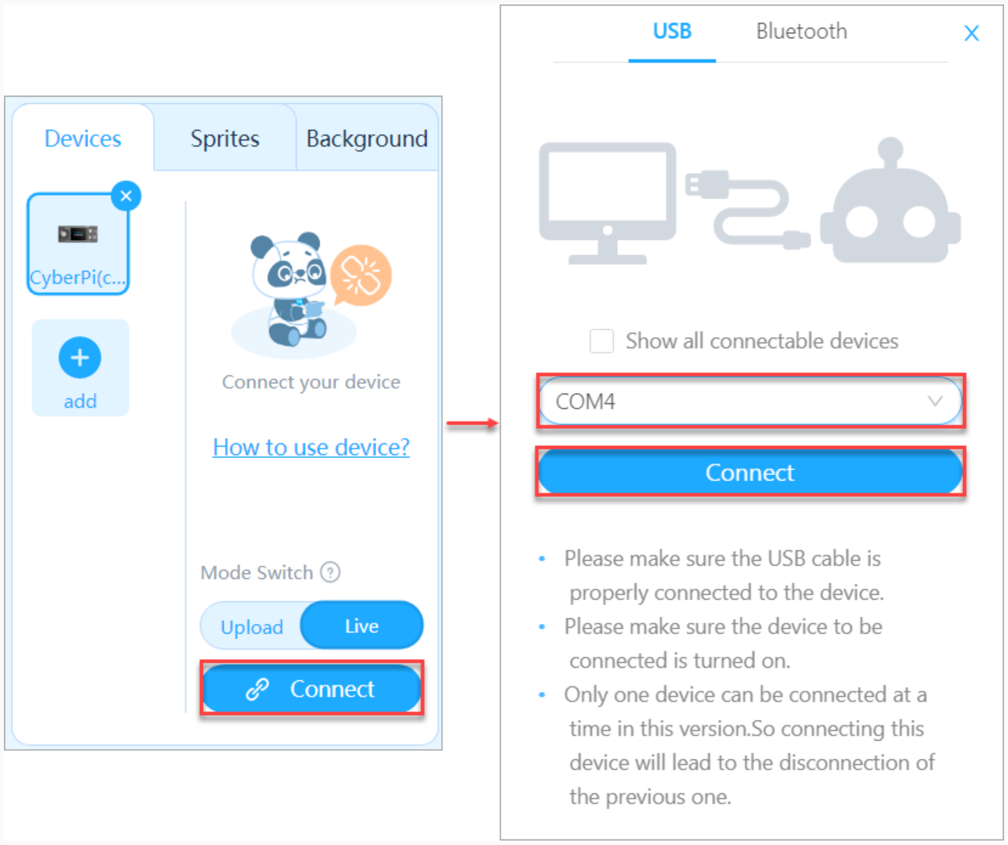


Software connection

After a device is successfully paired with the wireless adapter, you need to connect it to mBlock 5 before you program it.

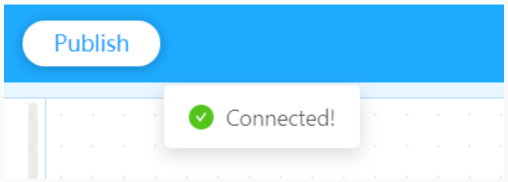
In this example, CyberPi is the device to be programmed.

Click **Connect**, select the serial port used by the wireless adapter in the dialog box that appears, and click **Connect**.



- Please make sure the USB cable is properly connected to the device.
- Please make sure the device to be connected is turned on.
- Only one device can be connected at a time in this version.So connecting this device will lead to the disconnection of the previous one.

A message is displayed after the device is connected, indicating that the connection is successful.



Now, you can start to program CyberPi on mBlock 5!

Bluetooth Compatibility

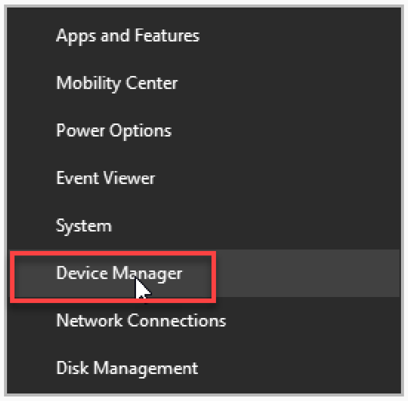
If you are to use mBlock 5 on a PC that runs Windows and is equipped with a non–Bluetooth 4.0 protocol, you need to use a Bluetooth 4.0 dongle to connect a device to mBlock 5. Check the Bluetooth version of your PC before performing other operations.

If you are to connect CyberPi, Halocode, or Codey Rocky to your computer directly through Bluetooth, the Bluetooth protocol on your computer must be Bluetooth 4.0 or later. If your computer is equipped with no Bluetooth adapter or the Bluetooth protocol is earlier than 4.0, consider purchasing the Makeblock wireless adapter or a kit including the Makeblock wireless adapter.

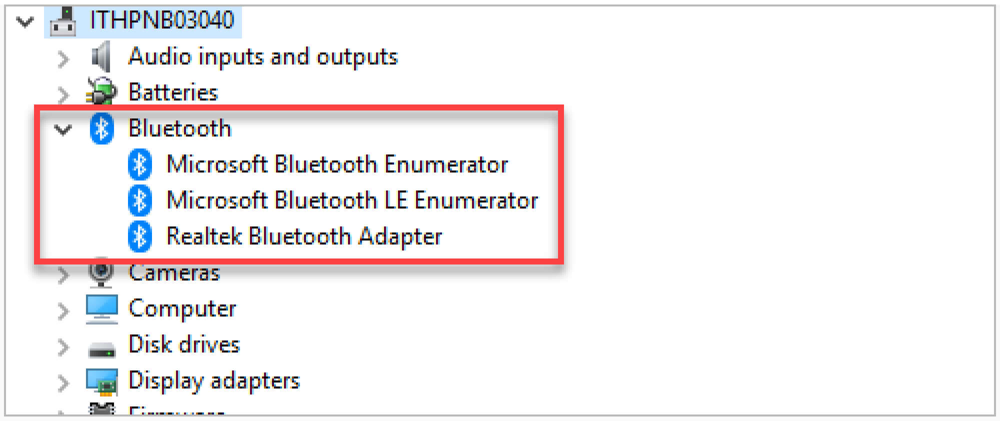
Check the Bluetooth version of a PC that runs Windows

To check the Bluetooth version, perform the following steps:

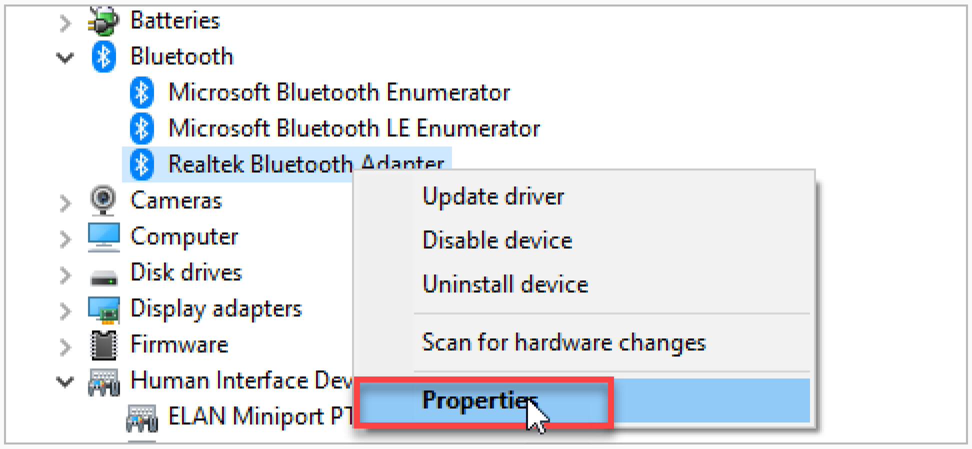
1. Press Win+X to open the Start Menu and select Device Manager.



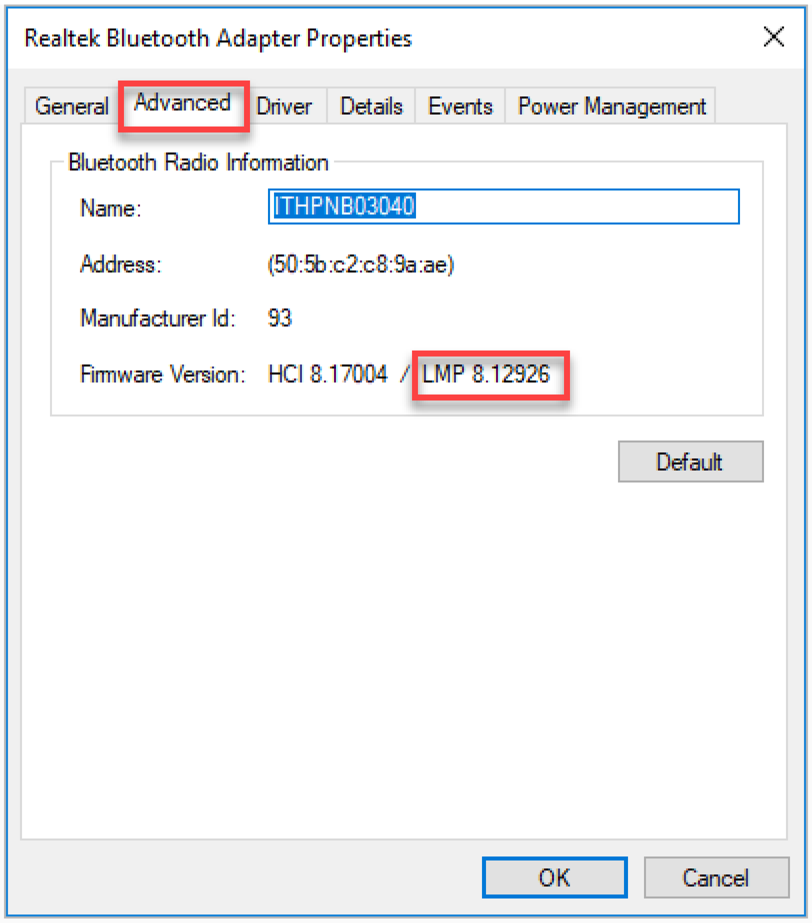
2. Click **Bluetooth**. You can see multiple Bluetooth devices.



3. Select your Bluetooth brand, right-click it, and choose **Properties**.



4. Click the **Advanced** tab and check the firmware version. The LMP number shows the version of Bluetooth your PC is using.


Realtek Bluetooth Adapter Properties

General | Advanced | Driver | Details | Events | Power Management

Bluetooth Radio Information

Name:	ITHPNB03040
Address:	(50:5b:c2:c8:9a:ae)
Manufacturer Id:	93
Firmware Version:	HCI 8.17004 / LMP 8.12926

The following table describes the LMP versions:

LMP No.	Bluetooth Version
LMP 9.x	Bluetooth 5.0
LMP 8.x	Bluetooth 4.2
LMP 7.x	Bluetooth 4.1
LMP 6.x	Bluetooth 4.0
LMP 5.x	Bluetooth 3.0 + HS
LMP 4.x	Bluetooth 2.1 + EDR
LMP 3.x	Bluetooth 2.0 + EDR
LMP 2.x	Bluetooth 1.2
LMP 1.x	Bluetooth 1.1
LMP 0.x	Bluetooth 1.0b

Alternatives

Alternatively, you can consider purchasing a third-party Bluetooth 4.0 adapter to connect Makeblock devices to mBlock 5. Due to the complexity of Windows and false labeling of some Bluetooth adapters on the market, however, the wireless connection may still fail.

You can refer to the following to install or uninstall the Bluetooth 4.0 driver:

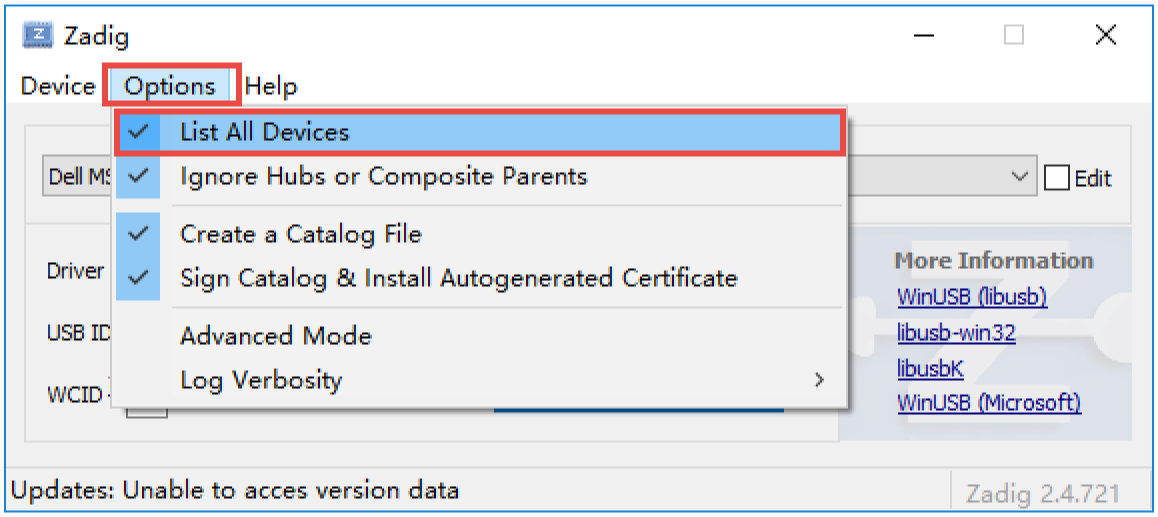
Note: After you install the Bluetooth 4.0 driver on Windows, the system Bluetooth is disabled. To restore the system Bluetooth, you need to uninstall the Bluetooth 4.0 driver. See "Uninstall the Bluetooth 4.0 driver to restore the system Bluetooth."

Install the Bluetooth 4.0 driver on a PC that runs Windows

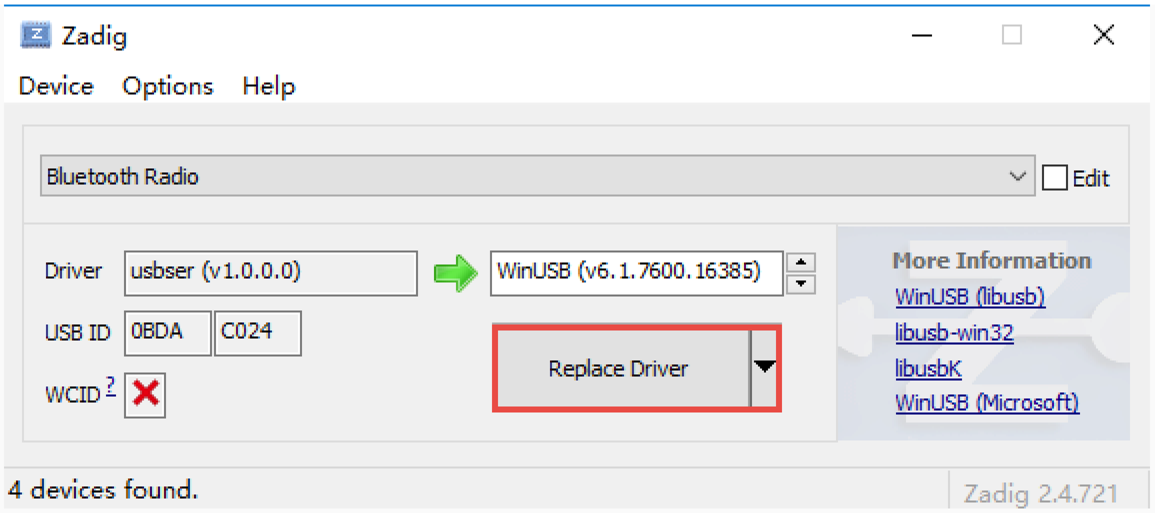
To install the Bluetooth 4.0 driver, perform the following steps:

1. Download Zadig tool.

2. Run Zadig tool, choose **Options** and click **List All Devices**.



3. Choose your device from the drop-down list and click **Replace Driver**.



Bluetooth Radio ☐ Edit

Driver usbser (v1.0.0.0) → WinUSB (v6.1.7600.16385)

USB ID 0BDA C024

WCID ? ✗

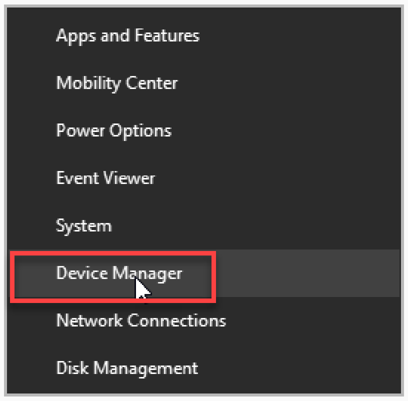
Replace Driver

More Information
WinUSB (libusb)
libusb-win32
libusbK
WinUSB (Microsoft)

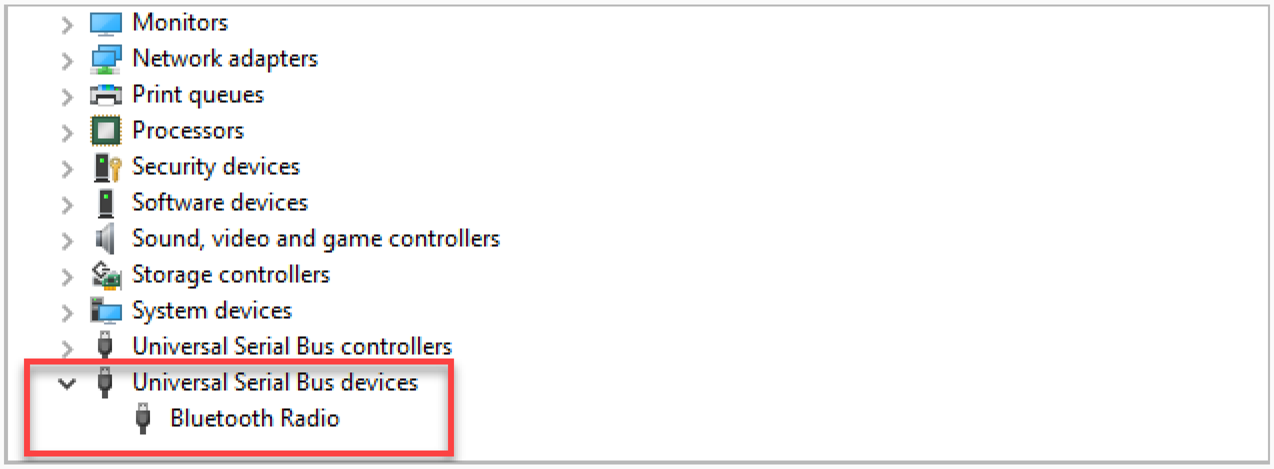
4 devices found. Zadig 2.4.721

Follow the following instructions to check Hardware information:

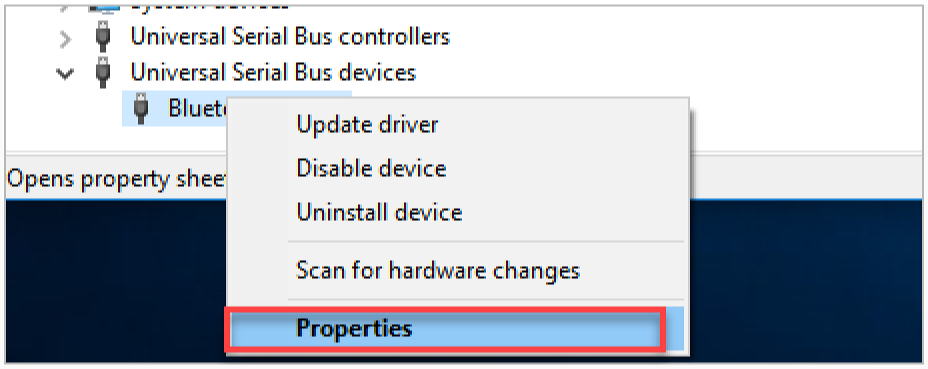
1. Press Win+X to open the Start Menu and choose **Device Manager**.



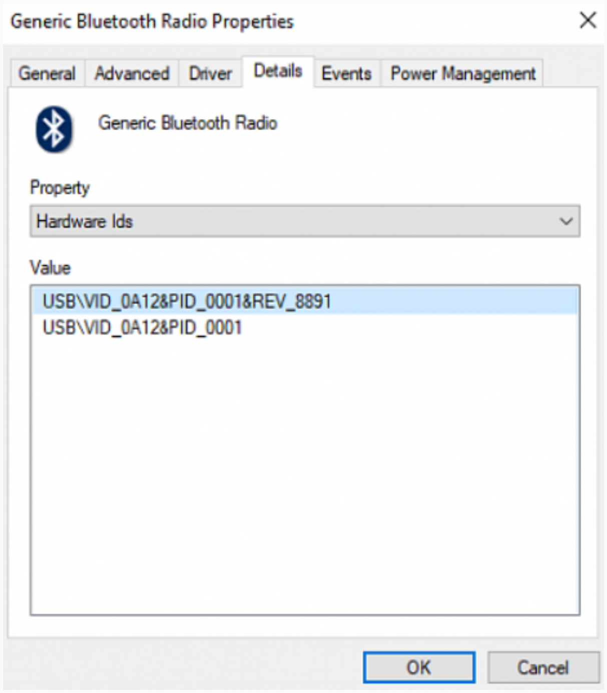
2. Under **Universal Serial Bus devices**, find **Bluetooth Radio**.



3. Right-click **Bluetooth Radio**, and choose **Properties**.


Universal Serial Bus controllers
Universal Serial Bus devices
Bluet
Opens property shee
Update driver
Disable device
Uninstall device
Scan for hardware changes
Properties

4. Select **Details**, and choose **Hardware Ids** from the **Property** drop-down list.

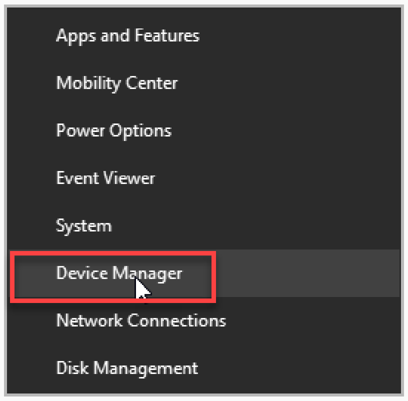


Property
Hardware Ids

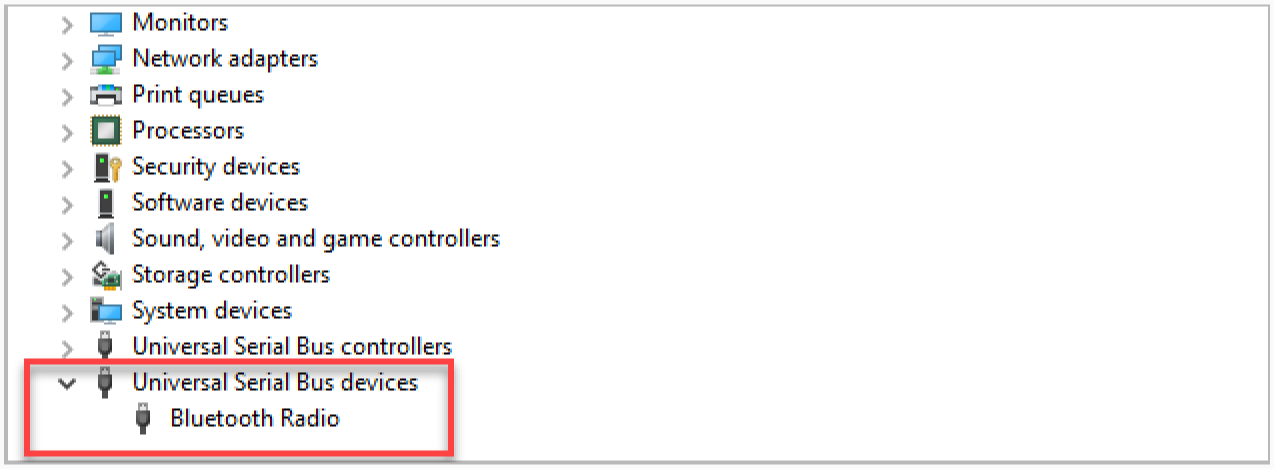
Value
USB\VID_0A12&PID_0001&REV_8891
USB\VID_0A12&PID_0001

Uninstall the Bluetooth 4.0 driver to restore the system Bluetooth

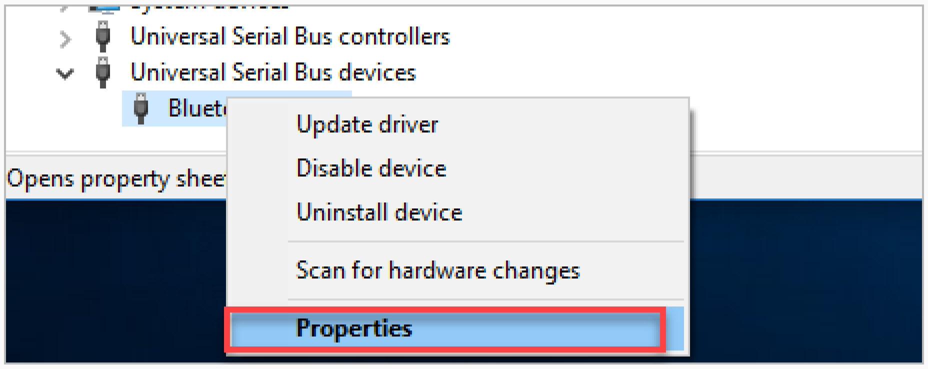
1. Press Win+X to open the Start Menu and choose **Device Manager**.



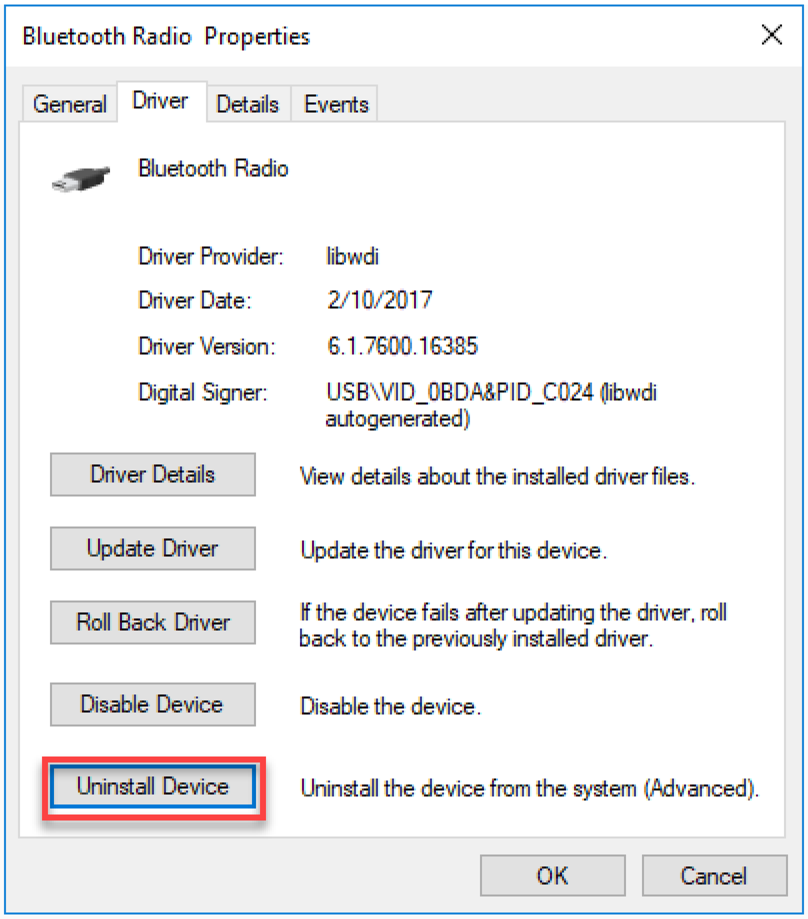
2. Under **Universal Serial Bus devices**, find **Bluetooth Radio**.



3. Right-click **Bluetooth Radio**, and choose **Properties**.


Universal Serial Bus controllers
Universal Serial Bus devices
Bluet
Opens property shee
Update driver
Disable device
Uninstall device
Scan for hardware changes
Properties

4. Choose **Driver** and then click **Uninstall Device**.


Bluetooth Radio Properties

General | Driver | Details | Events

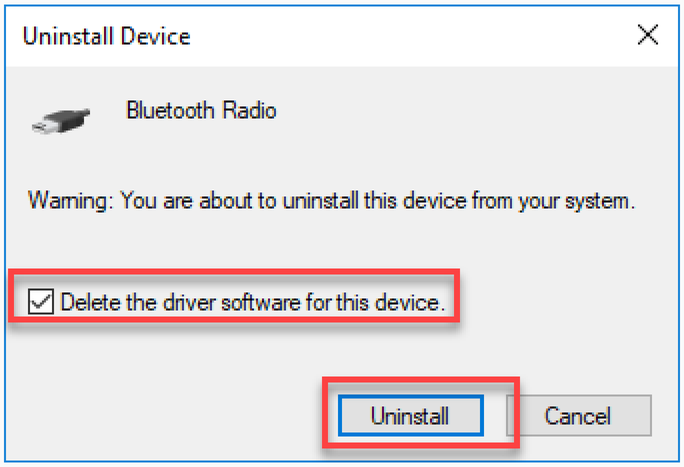
Bluetooth Radio

Driver Provider:	libwdi
Driver Date:	2/10/2017
Driver Version:	6.1.7600.16385
Digital Signer:	USB\VID_0BDA&PID_C024 (libwdi autogenerated)

Driver Details	View details about the installed driver files.
Update Driver	Update the driver for this device.
Roll Back Driver	If the device fails after updating the driver, roll back to the previously installed driver.
Disable Device	Disable the device.
Uninstall Device	Uninstall the device from the system (Advanced).

OK | Cancel

5. Check the box of **Delete the driver software for this device**, and then click **Uninstall**.


Uninstall Device

Bluetooth Radio

Warning: You are about to uninstall this device from your system.

☑ Delete the driver software for this device.

Uninstall | Cancel

6. After the uninstallation is complete, restart your computer.

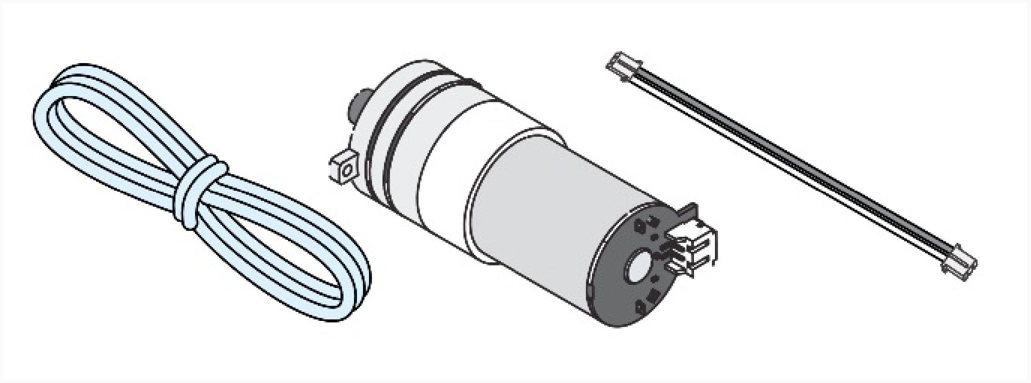
Water Pump

With the specified connection cables, the water pump can work together with Pocket Shield, mBot2, and DC motor drivers.

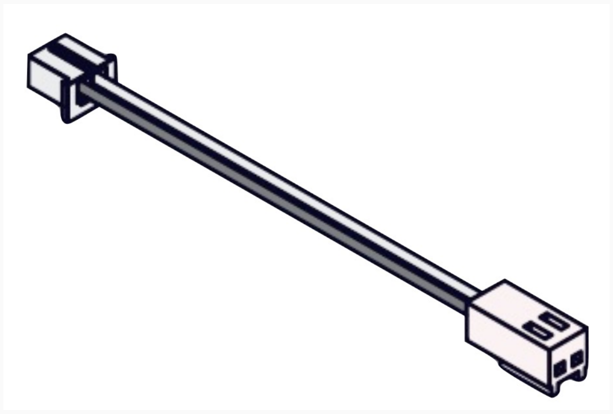
Driven by Pocket Shield, mBot2, or a DC motor driver, the water pump draws water or blows air to control the flow of water.

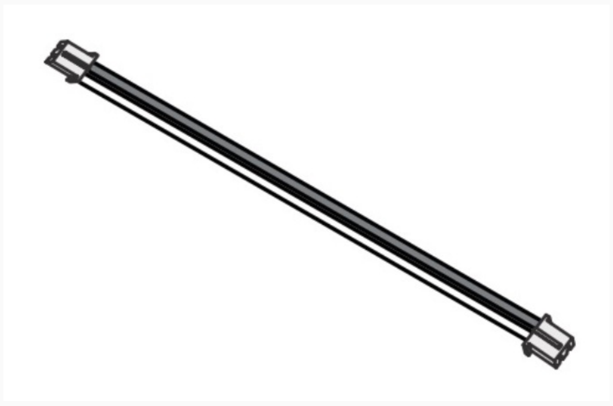
Note:

- The water pump must be used in combination with a tube that supports it.
- The water pump can't be used in water.



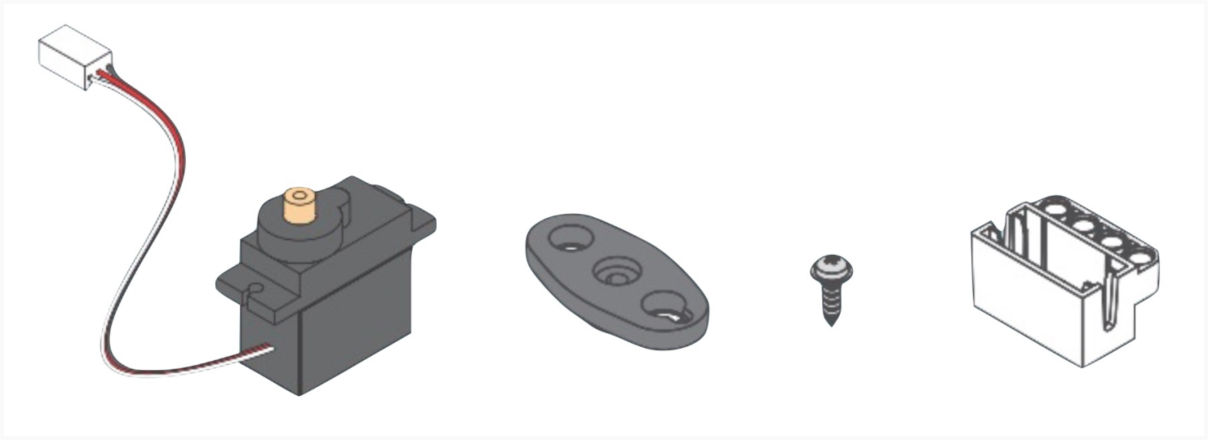
Compatibility description

Driven by	With
DC motor port—Pocket Shield	
DC motor port—mBot2 Shield	

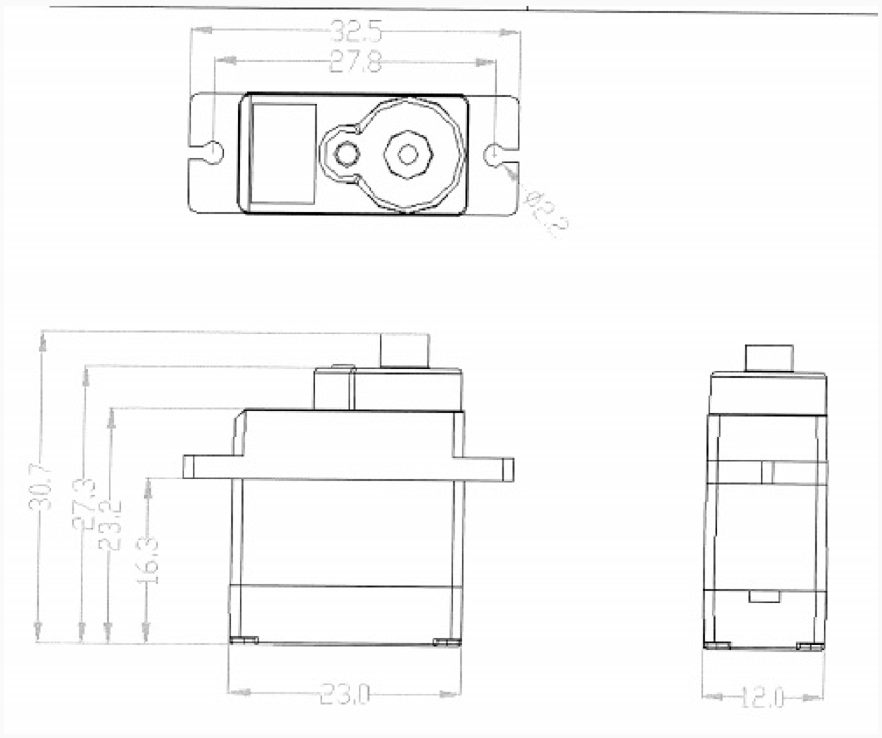
mBuild DC motor driver	
DC motor driver port that supports the voltage output of 5 V and peak current output of higher than 1 A	Connect them through welding or use a connection cable that supports the driver

9g Micro Servo (Metal Gear)

Makeblock 9g Micro Servo (metal gear) is a high-quality servo customized by Makeblock. It may be delivered assembled or unassembled in a pack, including the following four parts: 9g micro servo (metal gear), servo horn, screw, and servo holder.



Dimensions



Parameters

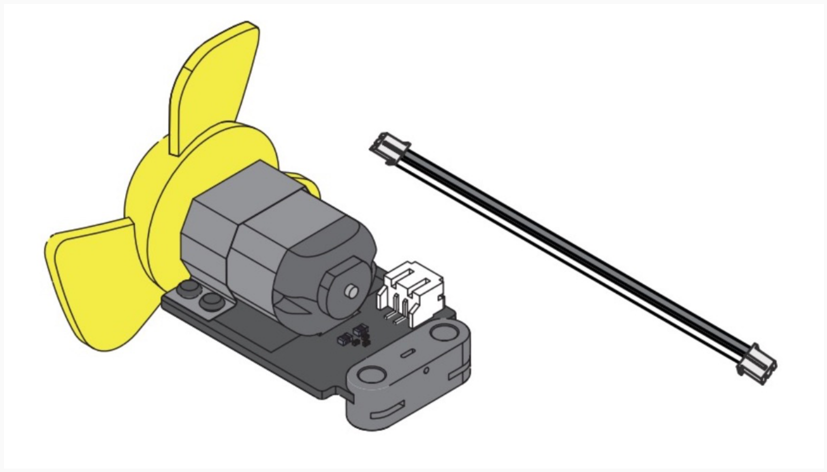
Electrical specifications		
Item	**4.8 V**	**6.0 V**
Operating speed (without load)	0.15±0.015 sec/60°	0.13±0.015 sec/60°
Running current (without load)	80±35 mA	90±35 mA
Max. load in operation	1.0±0.05kg·cm	1.1±0.1kg·cm
Stall torque (locked)	1.5±0.05kg·cm	1.8±0.1kg·cm
Stall current (locked)	680±40 mA	750±50 mA
Idle current (stopped)	6±1 mA	6±1 mA
Temperature drift (at 25°C)	⩽5°	⩽7°
Mechanical specifications		
Overall dimensions	See "Dimensions."	
Limit angle	180°±10°	
Weight	13.5±0.5g	
Connector wire gauge	#28 PVC, black-red-white, plug: 2510-3P, white	
Connector wire length	260±5 mm	
Horn gear spline	40T	
Horn type	Double-arm	
Reducation ratio	1/324	
Excessive play	⩽1°	
Screw	PM2×5 mm	
Control specifications		
Control system	Changing the pulse width	
Amplifier type	Analog controller	
Operating travel	180°±10° (600 to 2400 μ sec)	
Left & right travelling angle deviation	⩽7°	
Centering deviation	⩽1°	
Neutral position	1500 μ sec	

Dead band width	≤7 μ sec
Rotating direction	Counterclockwise (1500 to 2000 μ sec)
Pulse width range	500 to 2500 μ sec
Max. travel	About 190° (500 to 2500 μ sec)

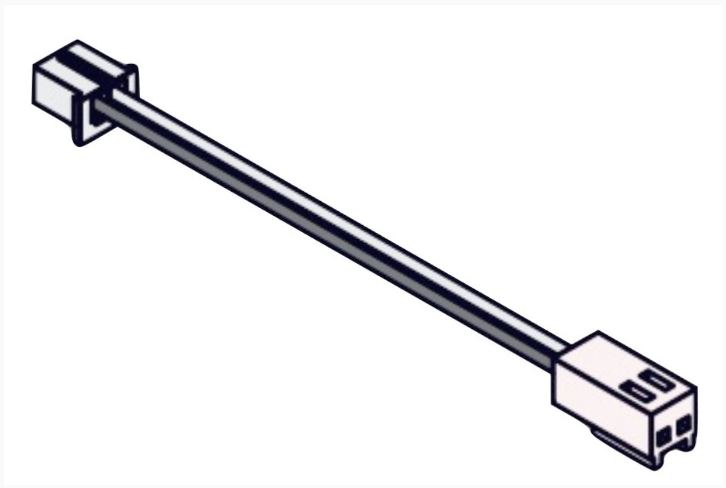
Fan

With the specified connection cables, the fan can work together with Pocket Shield, mBot2, and DC motor drivers.

Driven by Pocket Shield, mBot2, or a DC motor driver, the fan can run clockwise or counterclockwise to control the wind power and direction.



Compatibility description

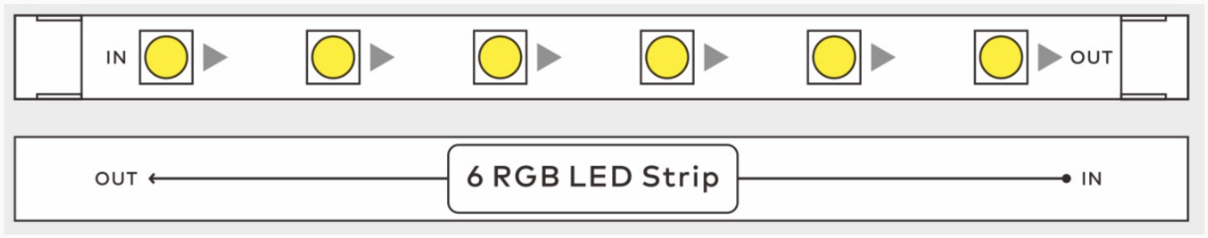
Driven by	With
DC motor port—Pocket Shield	
DC motor port—mBot2 Shield	
mBuild DC motor driver	

DC motor driver port that supports the voltage output of 5 V and peak current output of higher than 1 A	Connect them through welding or use a connection cable that supports the driver

Mini LED Strip and Ring

Due to the physical port and control protocols, the mini LED strip and ring can work properly only with the mBuild LED driver. Color deviation may be caused if you use a third-party driver to drive the mini LED strip or ring.

LED strip



The LED strip can be used to make a light sword or glowing words, or used to create an atmosphere.

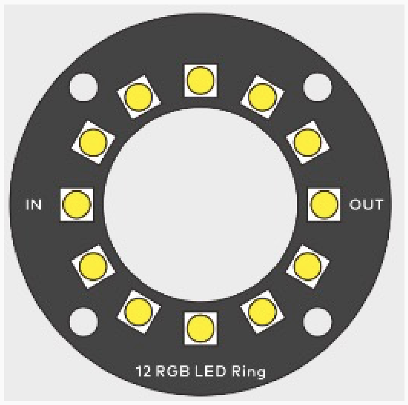
When using the LED strip, connect the IN plug to the LED driver or the OUT plug of another LED, LED strip, or LED ring.

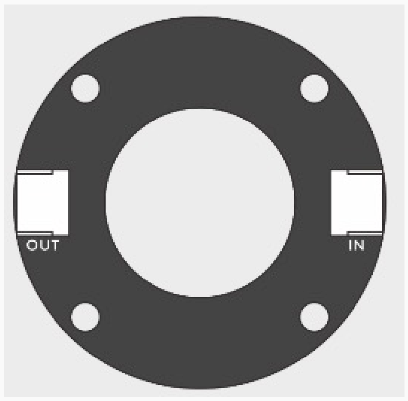
The LED strip is not waterproof. Using it in water may cause damage to it, and such damage is not covered in the warranty.

Parameter

- Dimensions: 8 mm × 112 mm

12-LED ring

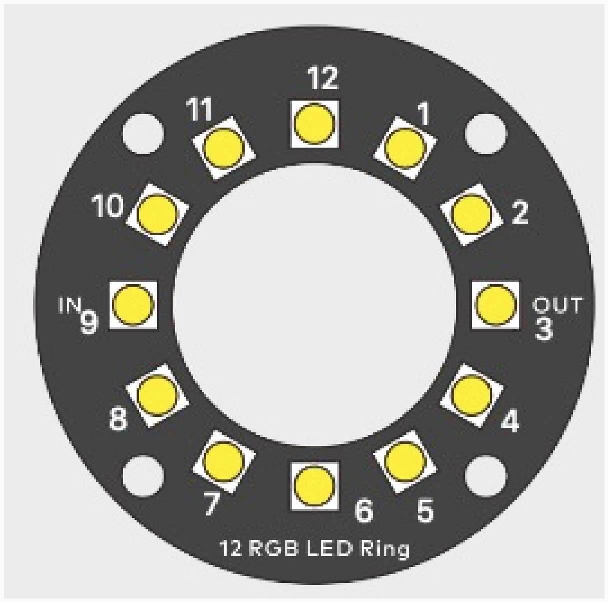




When using the LED ring, connect the IN plug to the LED driver or the OUT plug of another LED, LED strip, or LED ring.

The LED strip is not waterproof. Using it in water may cause damage to it, and such damage is not covered in the warranty.

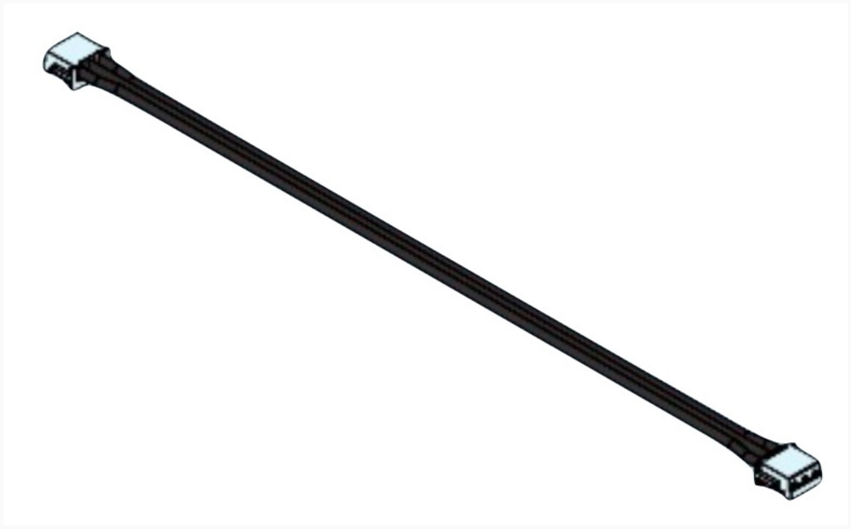
The places of the LEDs on the 12-LED ring are defined in the same way as a clock, as shown in the following figure.



Parameter

- Dimensions: diameter of 45 mm

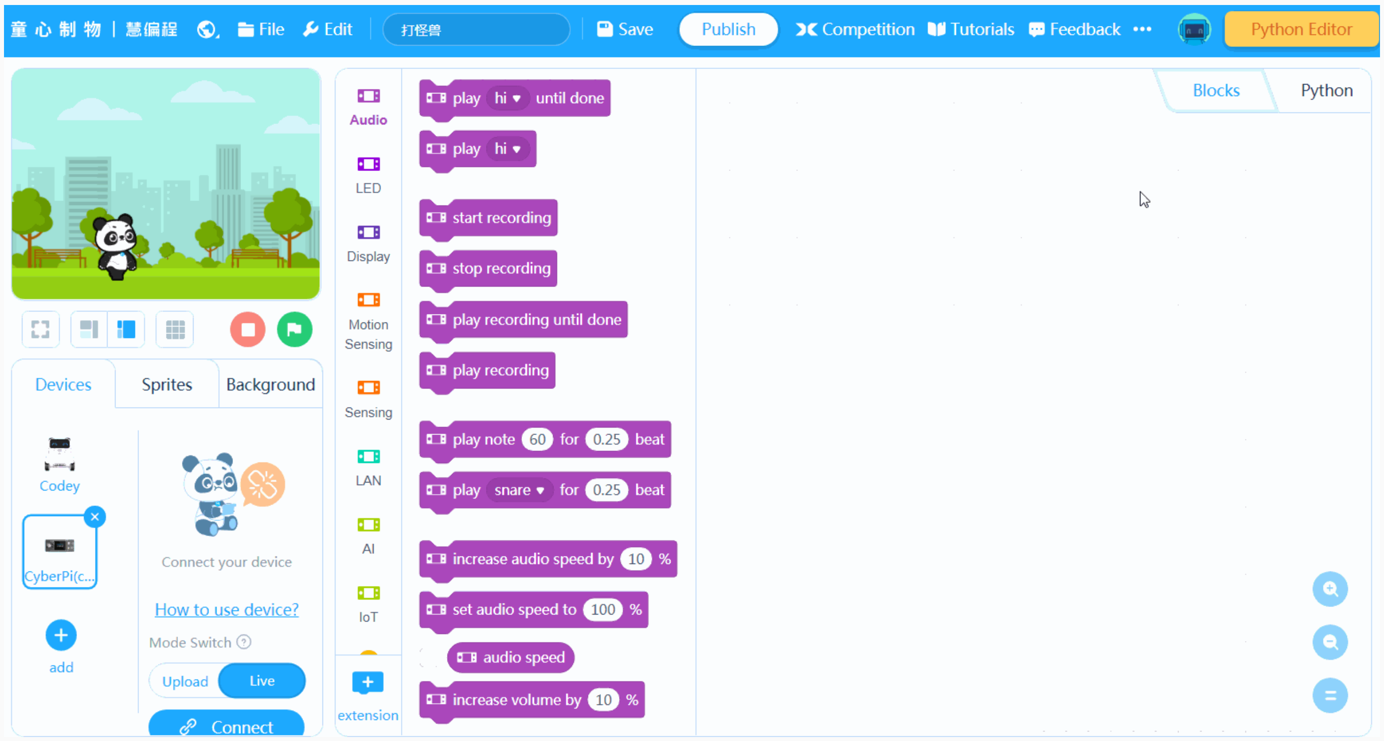
Compatibility description

Driven by	With
mBuild LED driver	

Classic Example Programs

mBlock 5 provides you with detailed example programs to help you get started quickly and learn about CyberPi series products.

On mBlock 5, choose **Tutorials** > **Example Programs**, and then click **CyberPi** to view all example programs provided for CyberPi.



In addition to the example programs, you can visit the mBlock community to find some programs you are interested in.

CyberPi Series Packages and Extensions

Currently, the following CyberPi series packages are provided:

CyberPi

You can purchase CyberPi to use the main functions of CyberPi series products.
For information about CyberPi and how to use CyberPi, see CyberPi and CyberPi Operation Guide.

Pocket Shield

You can purchase Pocket Shield to use it in combination with CyberPi.
Pocket Shield can connect to motors, servos, LED strips and third-party sensors as well as supplying power for CyberPi.

For information about Pocket Shield and how to use Pocket Shield, see Pocket Shield and Pocket Shield Operation Guide.

CyberPi AI & IoT Innovation Add-on Kit

This kit includes some mBuild modules and required electronic and structural parts, which allows the creation of a dozen of science and technology projects. Students can learn programming and the applied technologies in these projects.

For details about the mBuild modules, see mBuild Hardware Guide.

mBot2 AI Engineering Robot Kit

This kit includes mBot2 and the second-generation smart camera, which allows the creation of a dozen of science and technology projects. Students can learn the cutting-edge AI technologies as well as robotics and engineering.

You can contact the local dealer to purchase the kit or obtain the information and quote of the kit.

To become our dealer

Contact us to become our dealer or business partner.

CyberPi Operation Guide

English

日本語

Thank you for choosing CyberPi!

If you use CyberPi for the first time, read this guide carefully and follow the instructions so that you won't miss its functions.

1. Before you use CyberPi

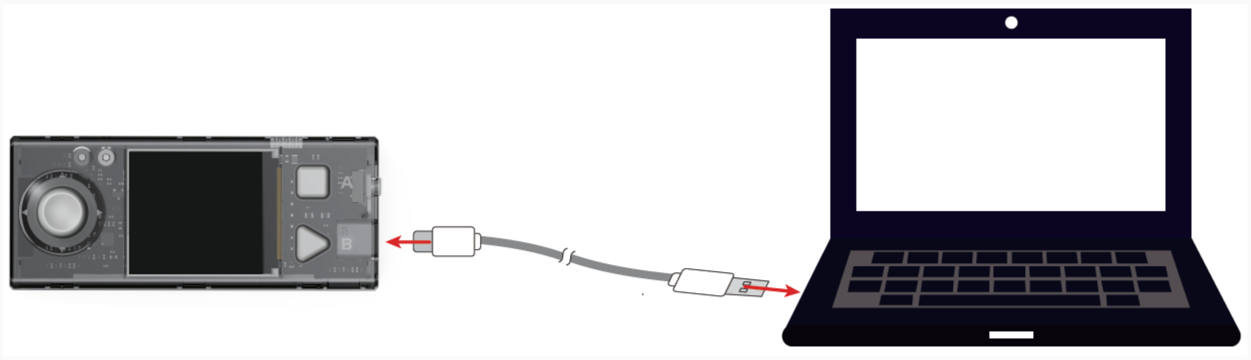
1.1 Power on CyberPi

Use a Micro USB cable (Type-C) to connect CyberPi to your PC.

The PC can supply power for CyberPi.

Note:

- The Micro USB cable is not included in the package. You need to purchase one separately.
- You can also use a power bank to supply power for CyberPi.

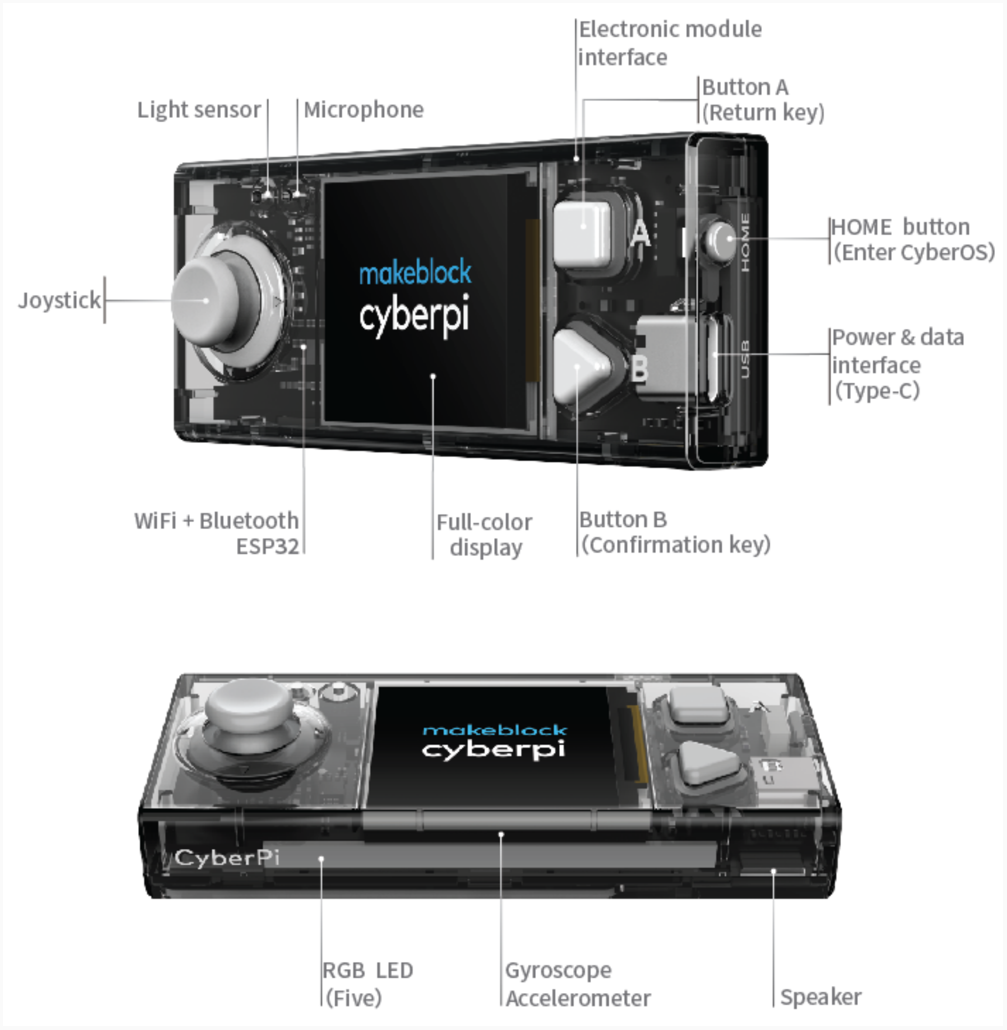


Alternatively, you can purchase Pocket Shield.

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.

1.2 Know about your CyberPi

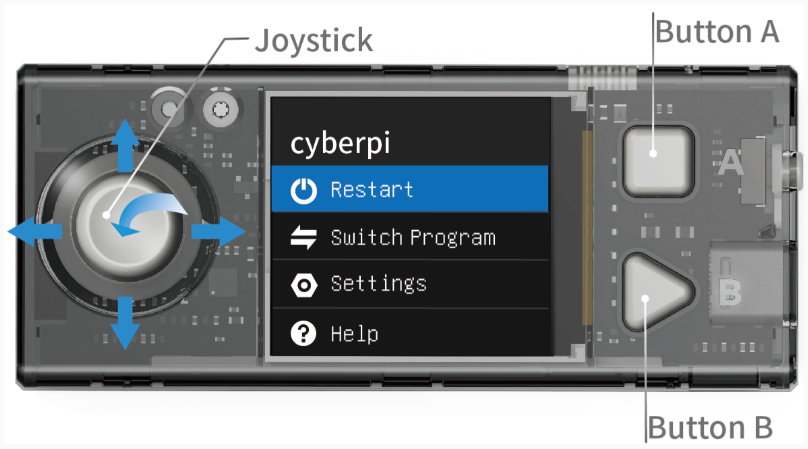
1.2.1 Hardware functions



1.2.2 Default functions of the joystick and buttons

CyberPi is equipped with one joystick and two buttons.

On system UIs, the default functions of the joystick and buttons are described as follows.



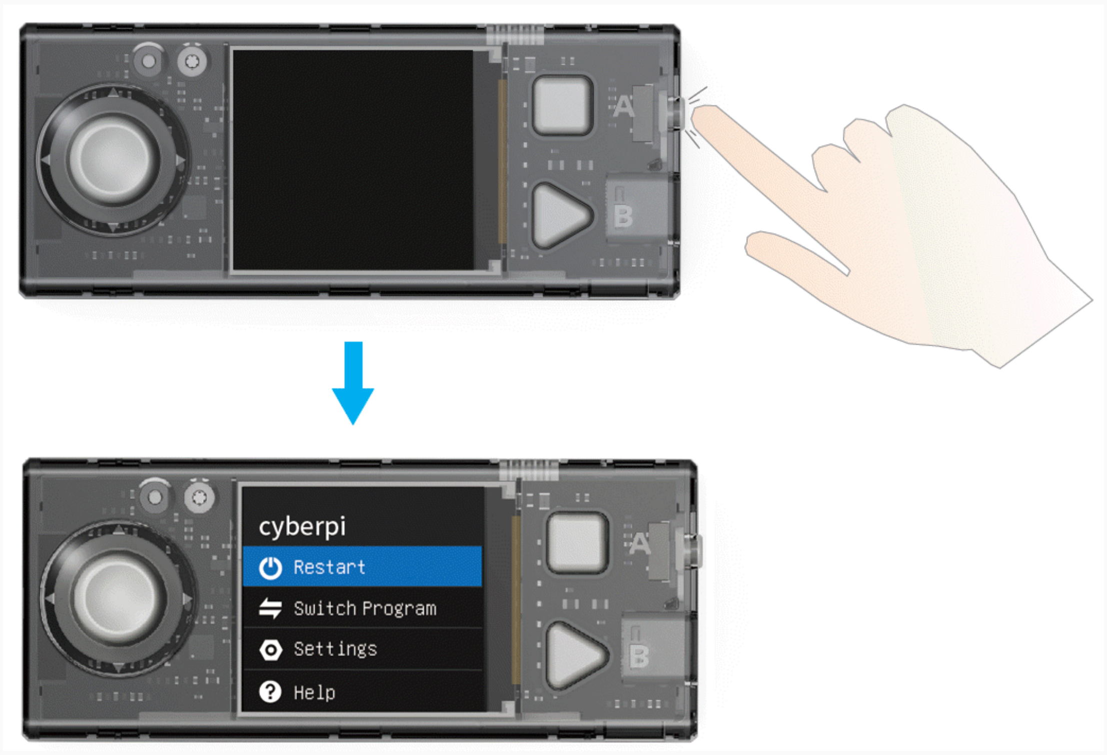
- **Move the joystick up:** to select the previous item
- **Move the joystick down:** to select the next item
- **Move the joystick to the left:** to select the left item
- **Move the joystick to the right:** to select the right item
- **Press the joystick in the middle:** to confirm the selection and enter the lower-level menu, if any
- **Button A:** to return to the upper-level menu
- **Button B:** to confirm the selection and enter the lower-level menu, if any

2. Enter the CyberOS homepage

CyberPi runs CyberOS.

After being powered on, CyberPi displays the system homepage.

If it doesn't display the homepage, you can press the **Home** button on the right side of CyberPi to enter the homepage.



After starting CyberOS, you can use the predefined programs, set the system language, and update the system through the Internet.

You can exit from the homepage by choosing "Restart Program."

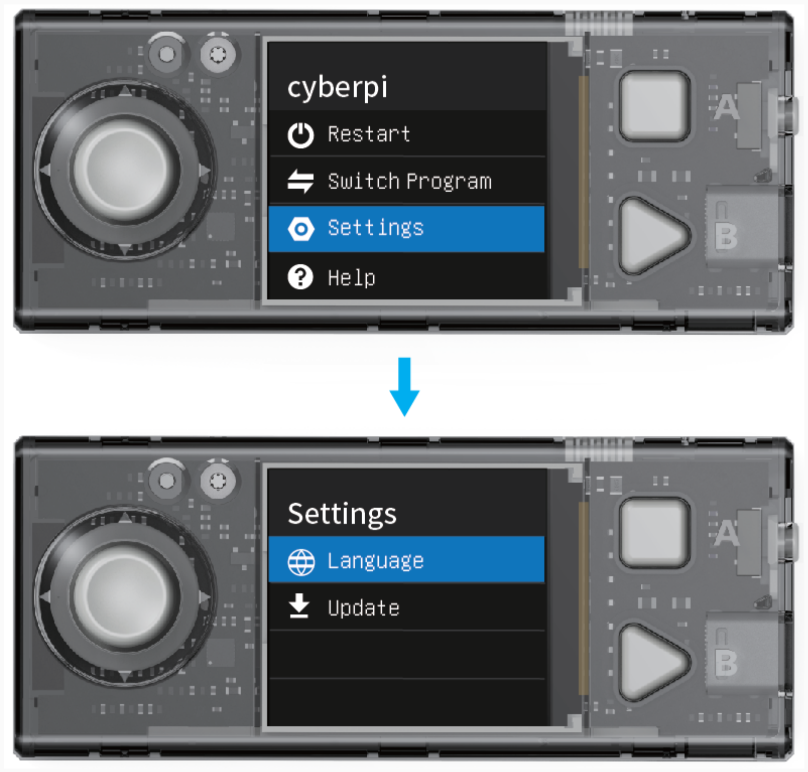
Note: When CyberPi is connected to mBlock 5 and being programmed in **Live** mode, the **HOME** button doesn't take effect and you can't enter the homepage by pressing it.

2.2 Set the system language

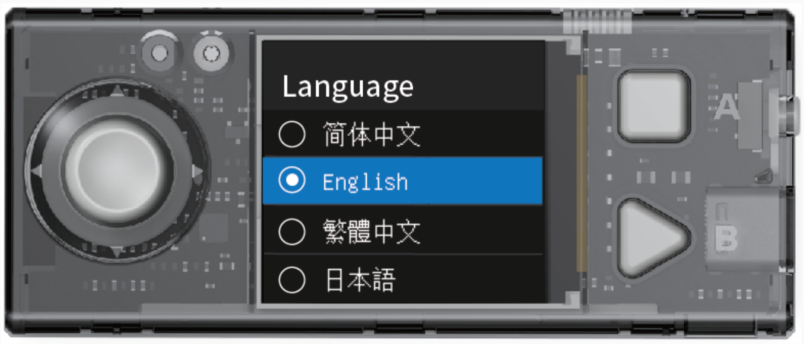
If the default system language is not your native language, you can change it.

The following takes the steps for changing the language from English to Chinese as an example:

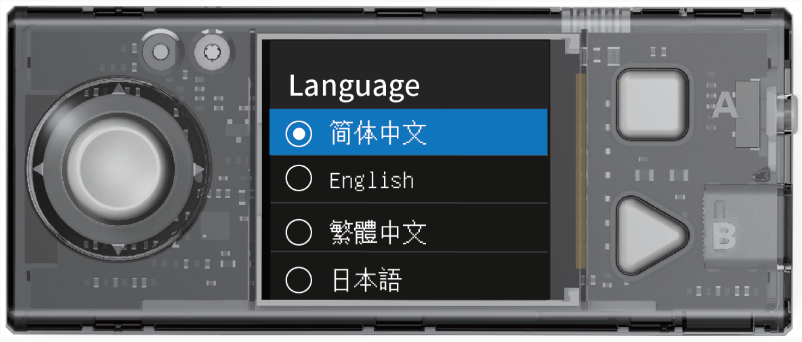
(1) Enter the homepage, select **Settings** by moving the joystick up or down, and press B to enter the **Settings** page.



(2) On the **Settings** page, select **Language** by moving the joystick up or down and press button B to enter the **Language** page.



(3) On the **Language** page, select **简体中文** by moving the joystick up or down and press button B to complete the setting.



After you set a language, the system switches to the language and returns to the homepage.

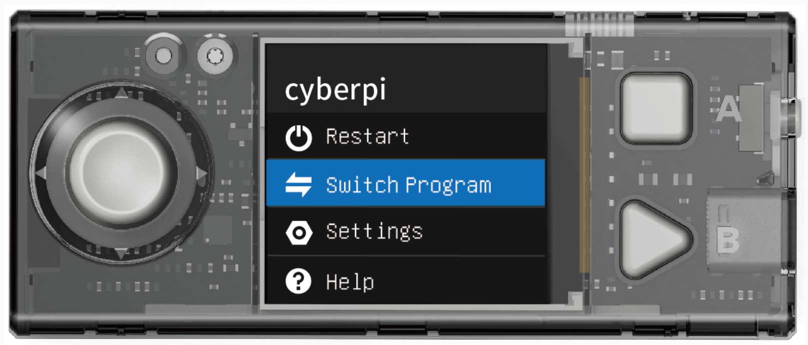
2.3 Switch programs

CyberPi can store multiple programs and you can switch them. In addition, CyberPi is delivered with several default example programs on it to help you understand its main functions.

Note:

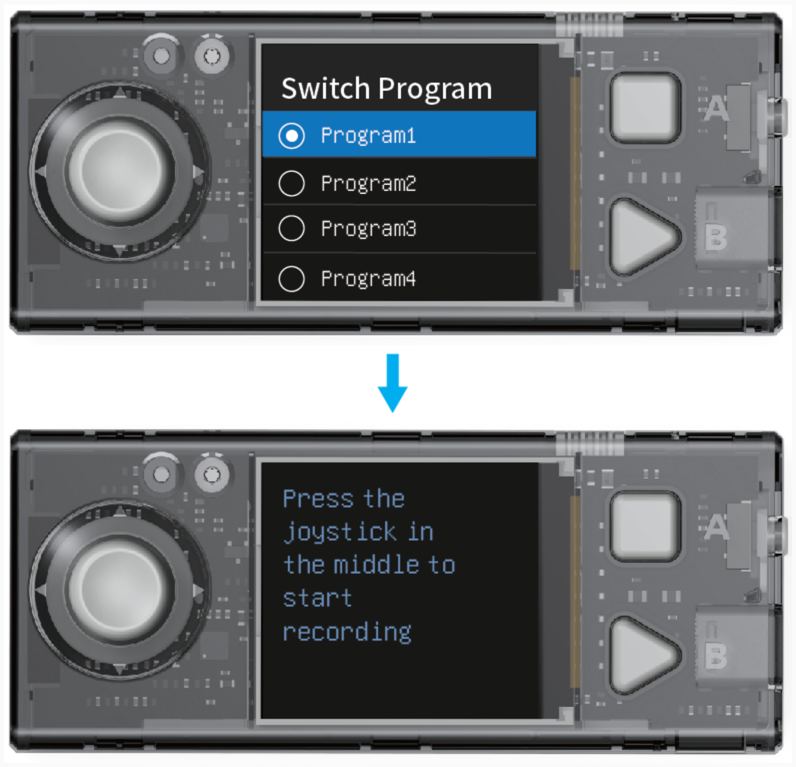
- Internet access is required for AI and IoT programs, and therefore they are not included in the default example programs and are provided in the example programs on mBlock 5. Find them by choosing **Tutorials** > **Example Programs** on mBlock 5.
- The program you upload to CyberPi on mBlock 5 replaces the one you have used last time. The default example programs are provided, and you can restore them by uploading them to your CyberPi.

(1) Enter the homepage, select **Switch Program** by moving the joystick up or down, and press B to enter the **Switch Program** page.



(2) On the **Switch Program** page, select a program, for example, **Program1**, by moving the joystick up or down, and then press button B.

CyberPi restarts and executes Program 1.



CyberPi displays the name of the program first and then prompts you to perform the steps required for executing the program.

3. Start programming

This section describes how to program CyberPi on mBlock 5. You can use the powerful functions of CyberPi through programming.

3.1 Download and install the required software

Currently, CyberPi supports the block-based graphical programming and Python programming. Make sure that you have downloaded and installed the required software.

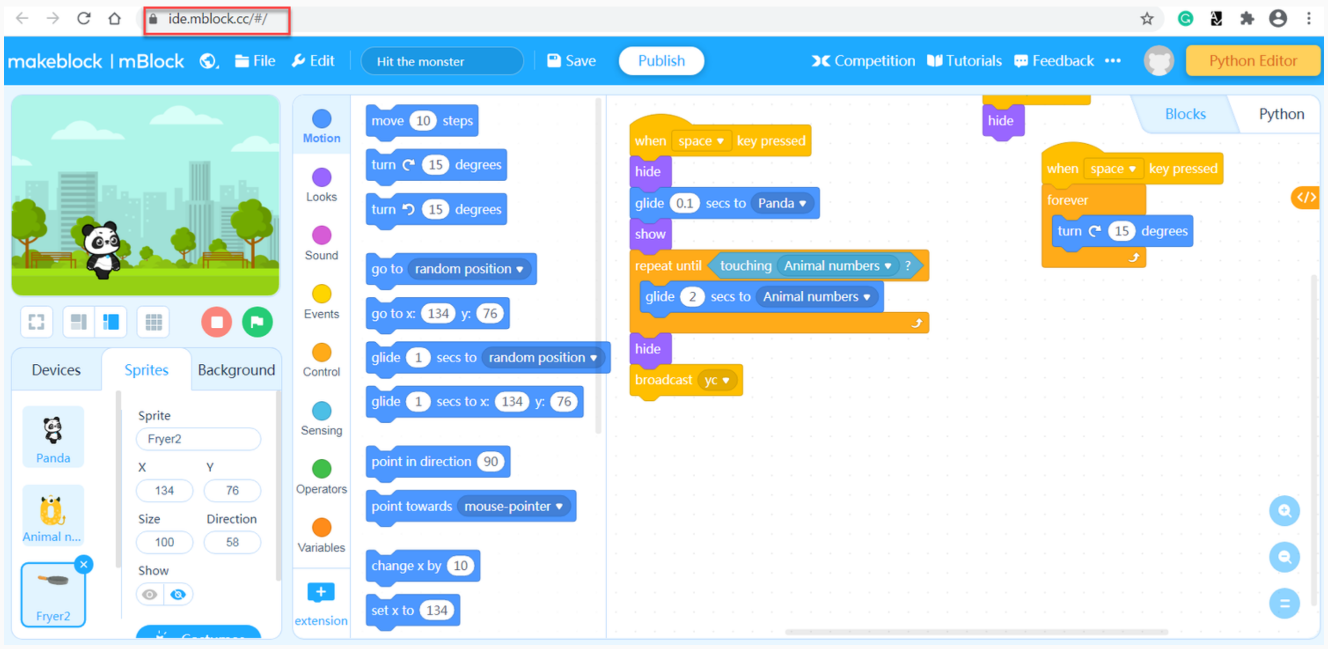
anguage	Editor	mBlock 5 version	Required software	URL address
Scratch, MicroPython	mBlock Block-based Editor	PC client on Windows/Mac	mBlock 5 for Windows mLink2 for Windows mBlock 5 for Mac mLink2 for Mac	N/A

		Web version on Windows/Mac	Google Chrome mLink2 for Windows mLink2 for Mac	https://ide.makeblock.com/
Python, MicroPython	mBlock-Python Editor	Web version on Windows/Mac	Google Chrome mLink2 for Windows mLink2 for Mac	https://python.makeblock.com/

3.2 Block-based programming

3.2.1 Add and connect CyberPi

(1) Open the web version of mBlock 5 on your PC.



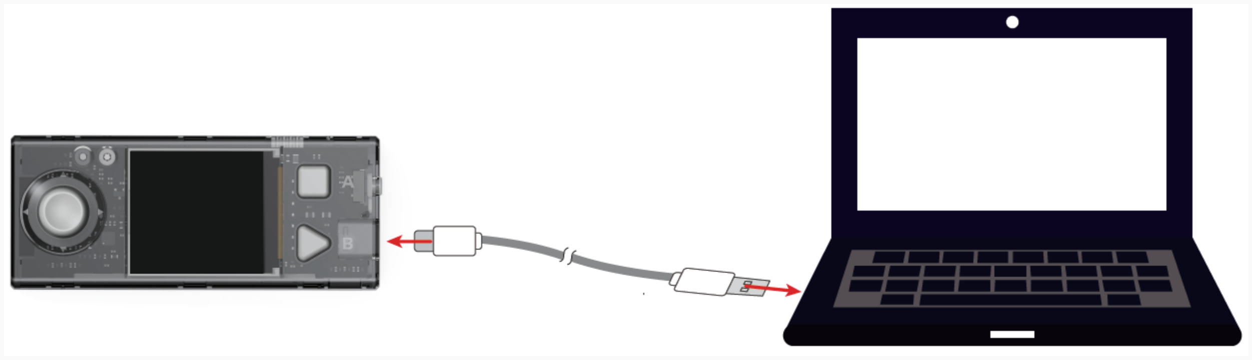
```
when space key pressed
hide
glide 0.1 secs to Panda
show
repeat until touching Animal numbers ?
    glide 2 secs to Animal numbers
hide
broadcast yc
```

```
hide
```

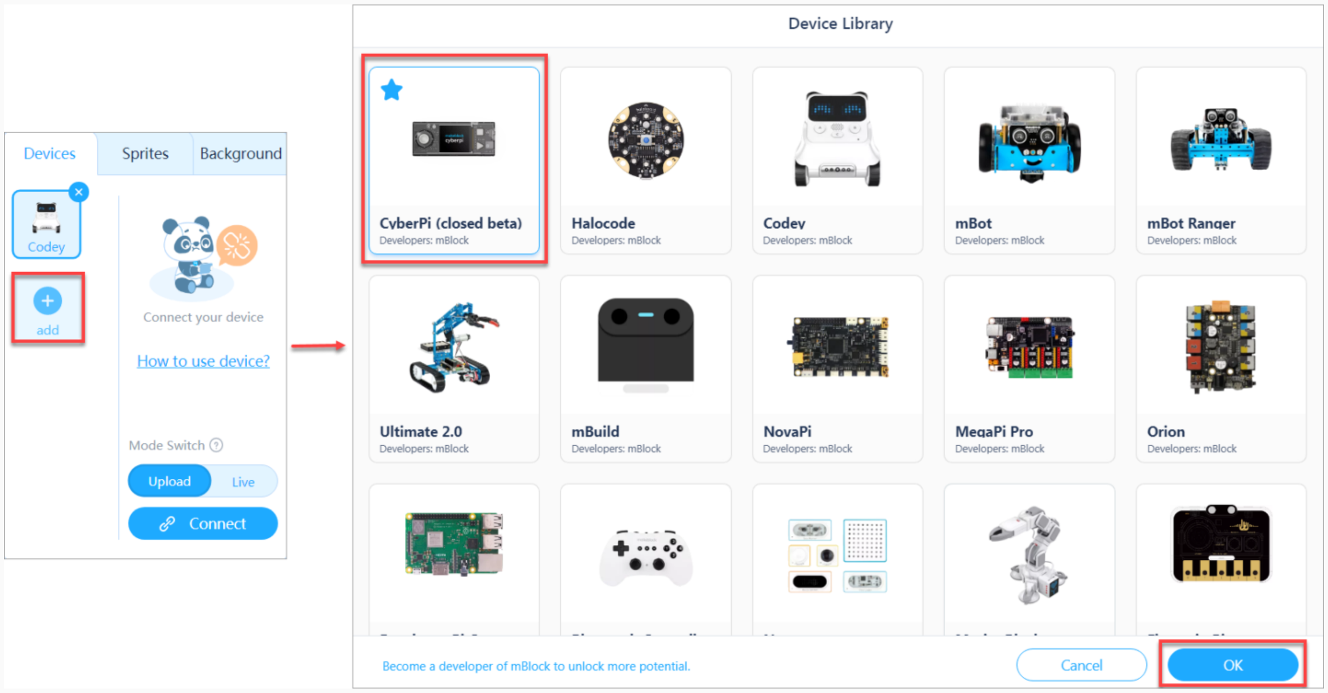
```
when space key pressed
forever
    turn ↻ 15 degrees
```

Note: To use some cloud service functions, you need to sign in to your mBlock 5 account. Sign up an account if you haven't got one.

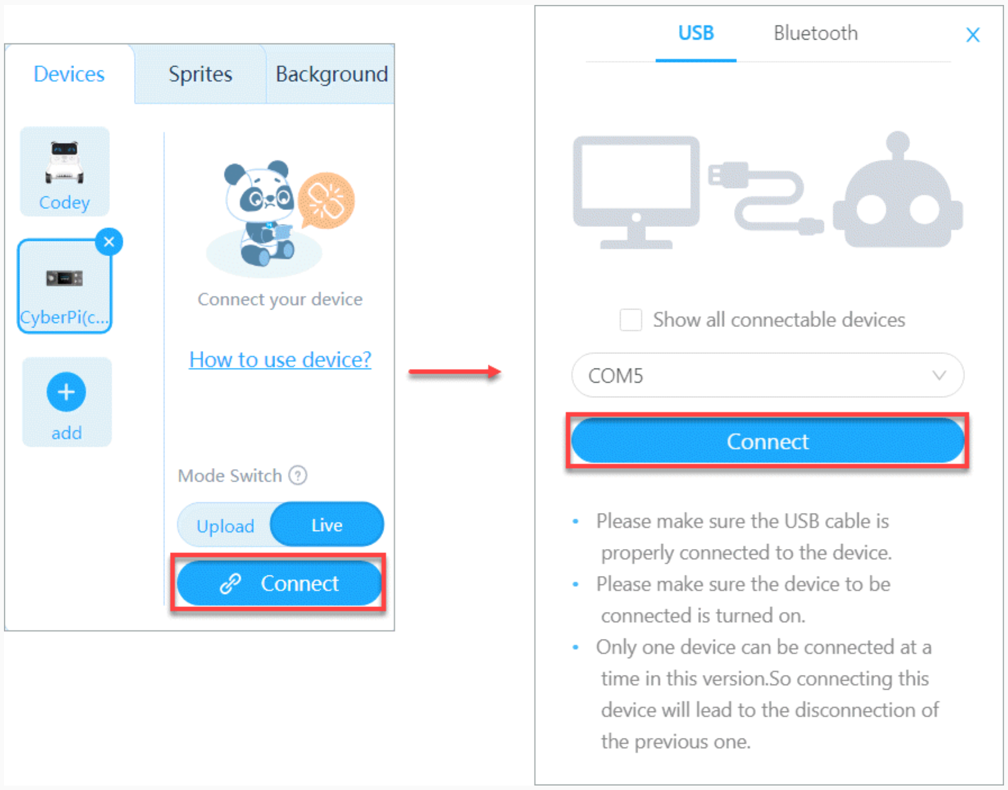
(2) Use a Micro USB cable (Type-C) to connect CyberPi to your PC.



(3) Click **+ add** on the **Devices** tab, select **CyberPi** in the device library, and click **OK**.

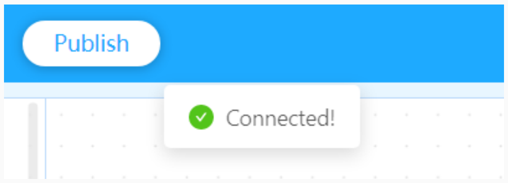


(4) Click **Connect** to connect CyberPi to mBlock 5.



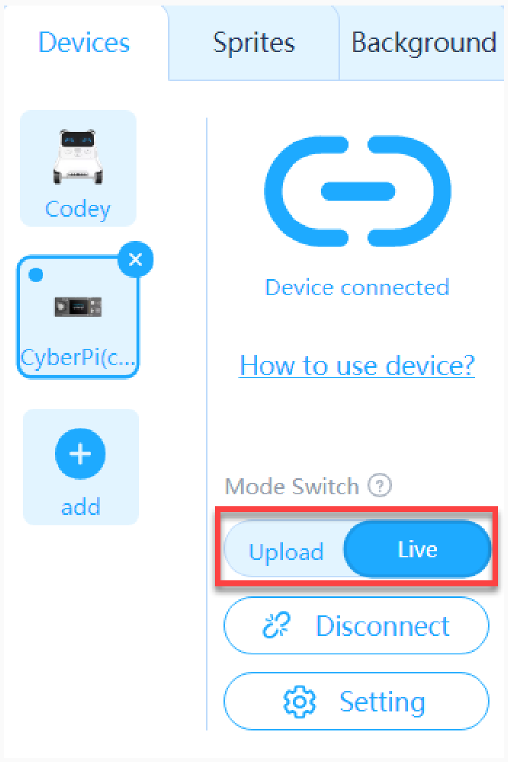
- Please make sure the USB cable is properly connected to the device.
- Please make sure the device to be connected is turned on.
- Only one device can be connected at a time in this version.So connecting this device will lead to the disconnection of the previous one.

A message is displayed after CyberPi is connected, indicating that the connection is successful.



(5) Set the programming mode.

mBlock 5 provides two programming modes, namely **Live** and **Upload**. You can click to switch the modes.



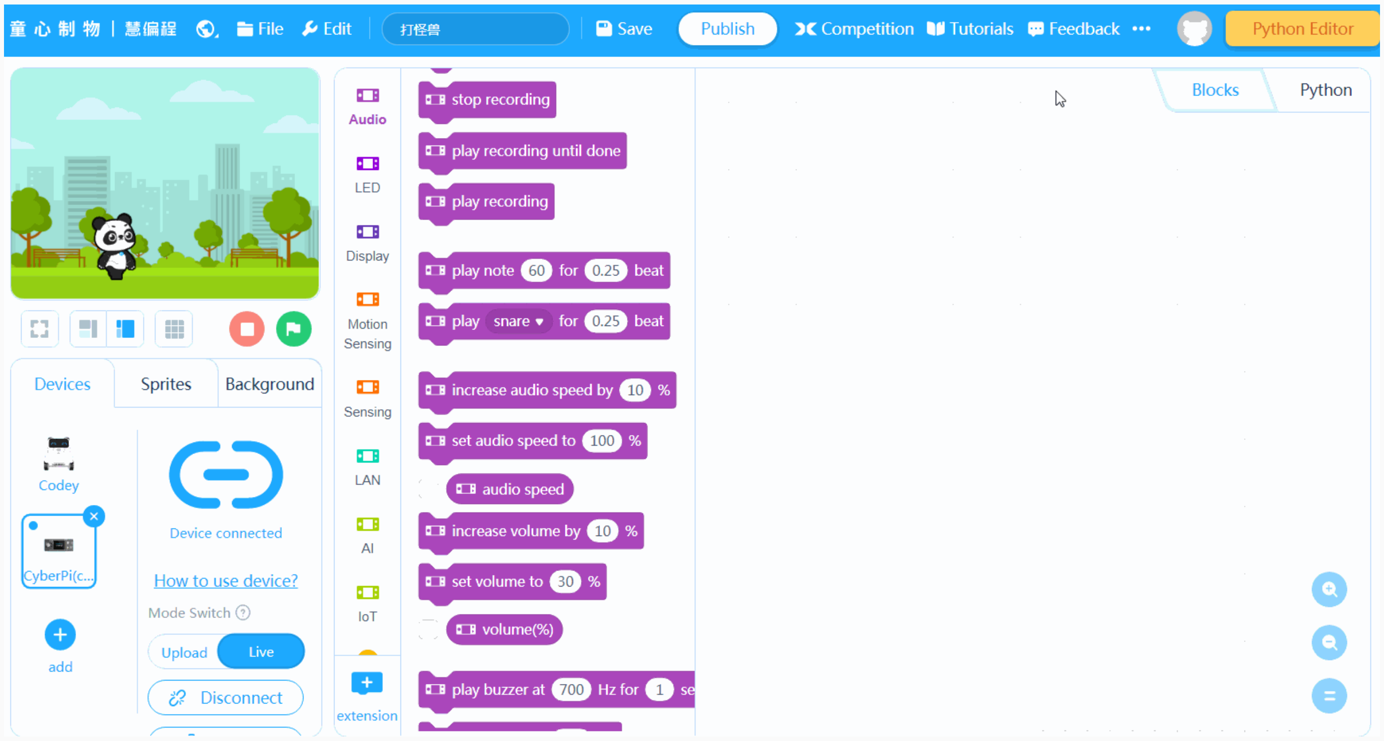
Live: In this mode, you can view the program execution effect in real time, which facilitates the debugging of the program. In this mode, you must keep CyberPi connected to mBlock 5. If they are disconnected, the program cannot be executed.

Upload: In this mode, you need to upload the compiled program to CyberPi. After being successfully uploaded, the program can still run properly on CyberPi when it is disconnected from mBlock 5.

3.2.2 Example programs

mBlock 5 provides a large number of example programs.

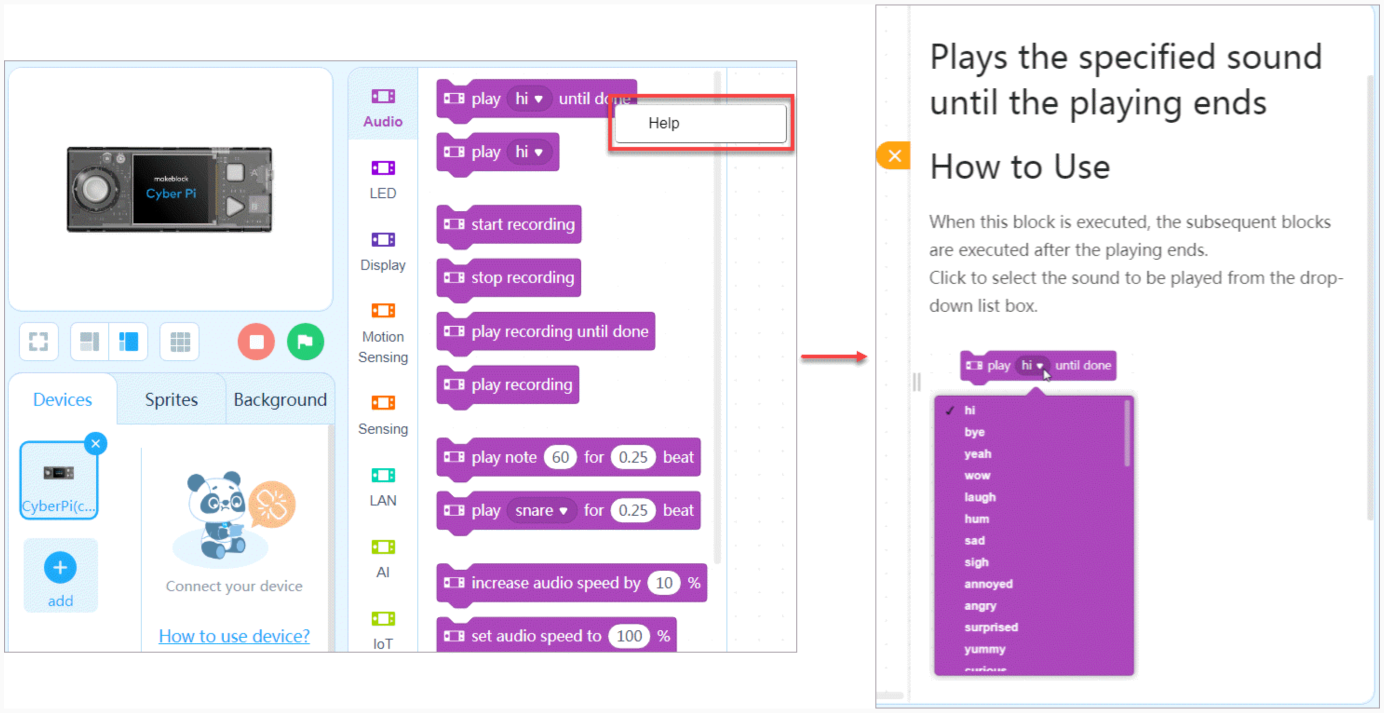
Choose **Tutorials** > **Example Programs** and click **CyberPi** to view example programs provided for CyberPi.



You can select and load a program to view, execute, or upload it to CyberPi.

3.2.3 Block help

mBlock 5 provides multiple types of blocks for CyberPi. If you don't understand a block when using it, you can right-click it and click **Help** that appears.



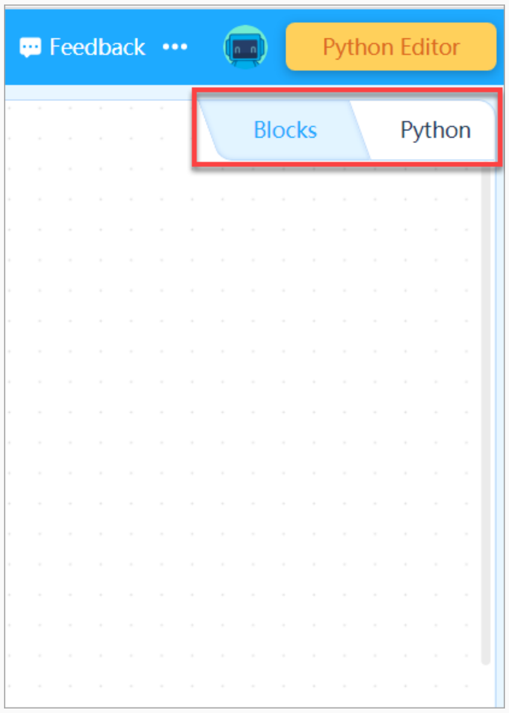
Plays the specified sound until the playing ends

How to Use

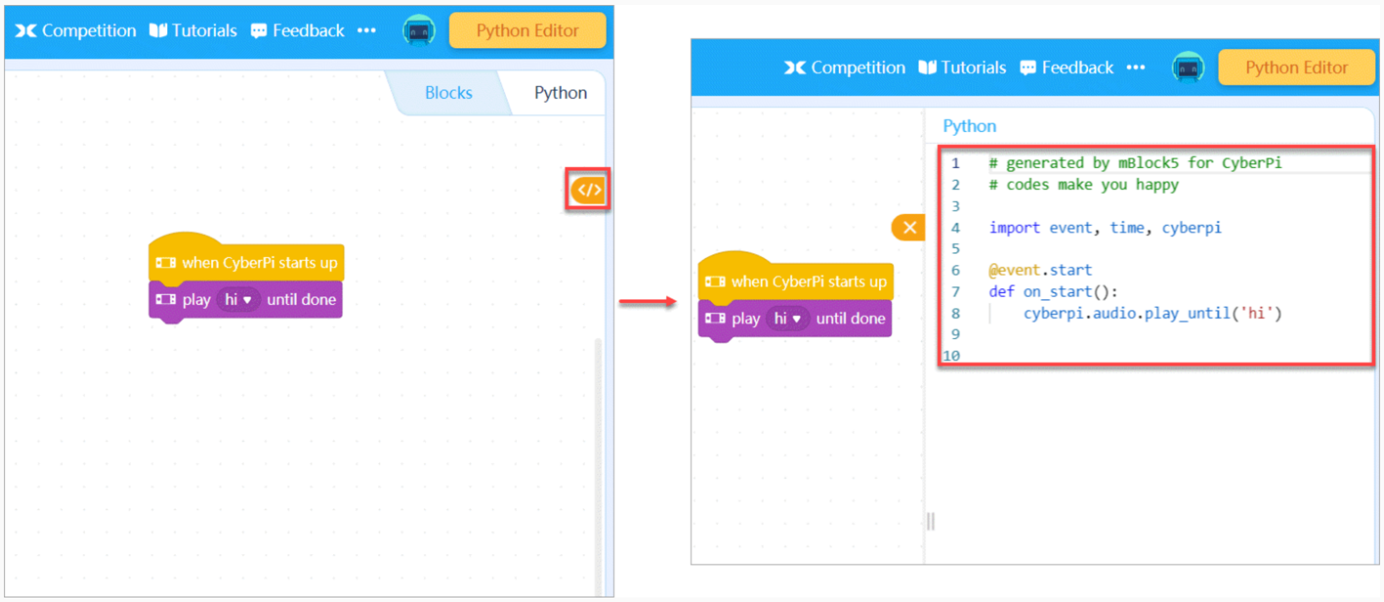
When this block is executed, the subsequent blocks are executed after the playing ends.
Click to select the sound to be played from the drop-down list box.

3.2.4 Programming languages

mBlock 5 provides two programming languages for CyberPi, namely block-based programming and Python. In **Upload** mode, you can click the buttons on the right to switch the programming languages.



In addition, when programming CyberPi in the block-based language, you can click the switching button on the right to view the corresponding Python statements (obtained by converting the blocks).



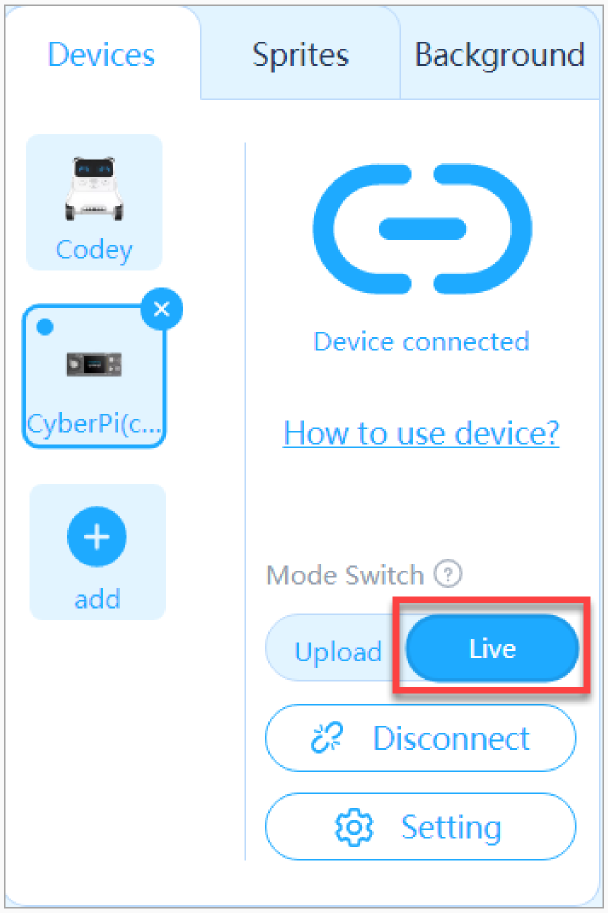
```
# generated by mBlock5 for CyberPi
# codes make you happy

import event, time, cyberpi

@event.start
def on_start():
    cyberpi.audio.play_until('hi')
```

Note: You can see Python API Documentation for CyberPi to know about more functions of CyberPi.

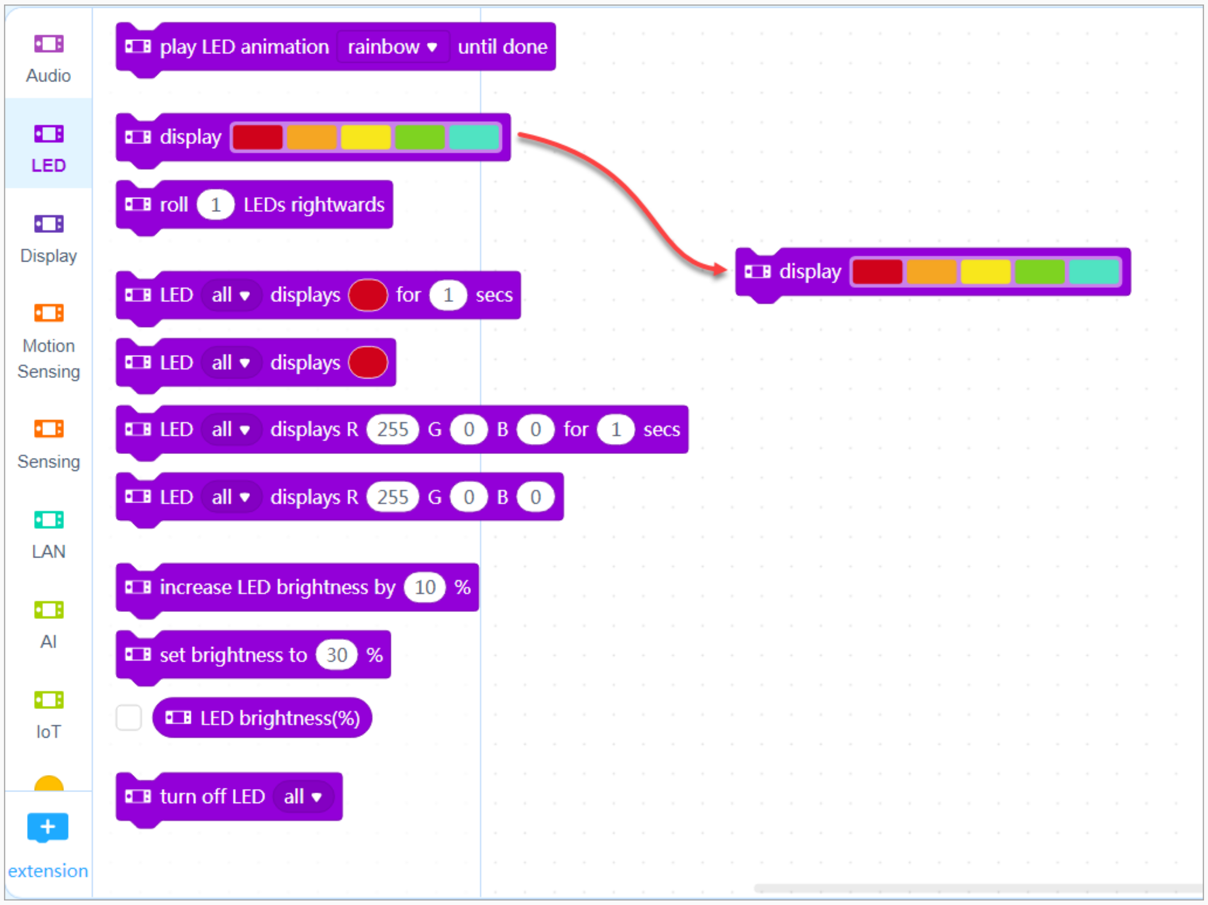
3.2.5 Example: Compile the “Marquee” program

Perform the following steps to compile a program using blocks in **Live** mode to implement the marquee effect of the LEDs on CyberPi.

(1) Set the programming mode to **Live**.

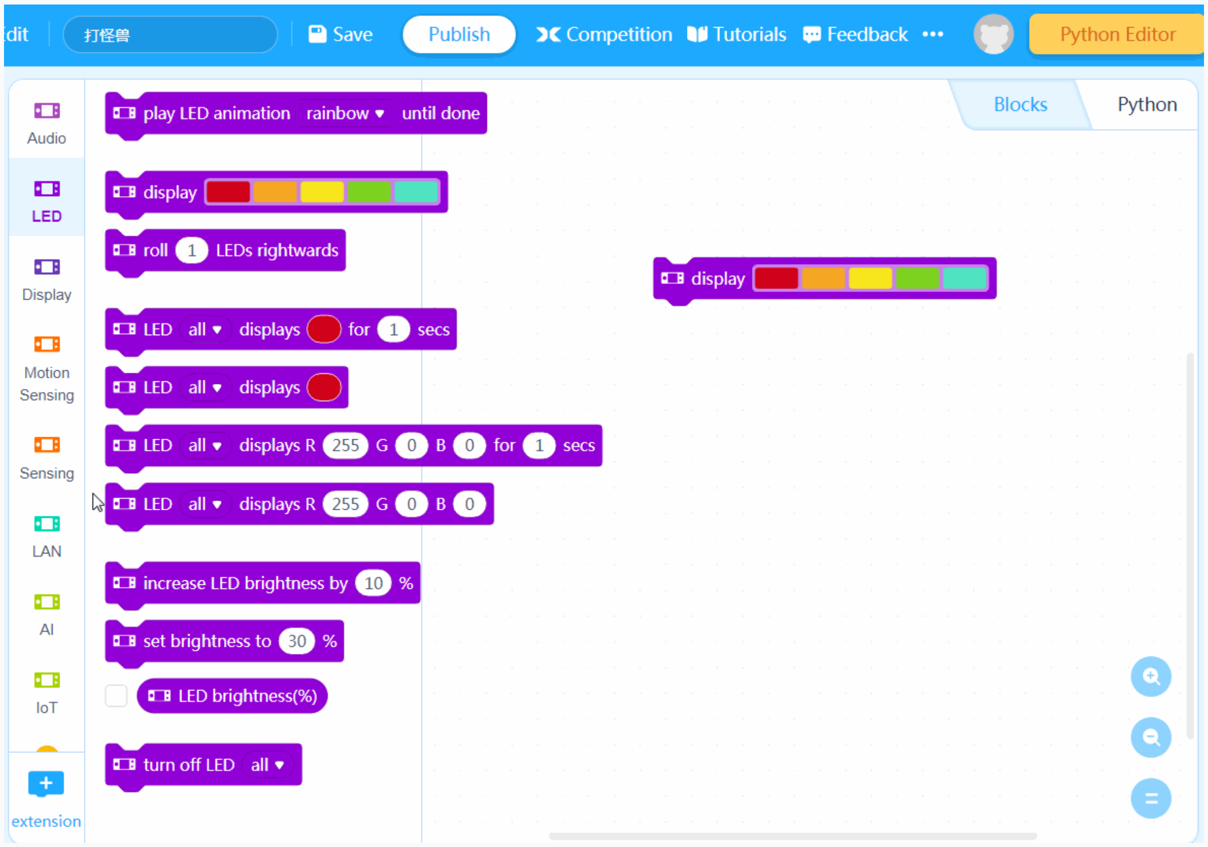


(2) Set the initial color(s) of the LEDs.


Audio
LED
Display
Motion Sensing
Sensing
LAN
AI
IoT
extension
play LED animation rainbow until done
display
roll 1 LEDs rightwards
LED all displays for 1 secs
LED all displays
LED all displays R 255 G 0 B 0 for 1 secs
LED all displays R 255 G 0 B 0
increase LED brightness by 10 %
set brightness to 30 %
LED brightness(%)
turn off LED all
display

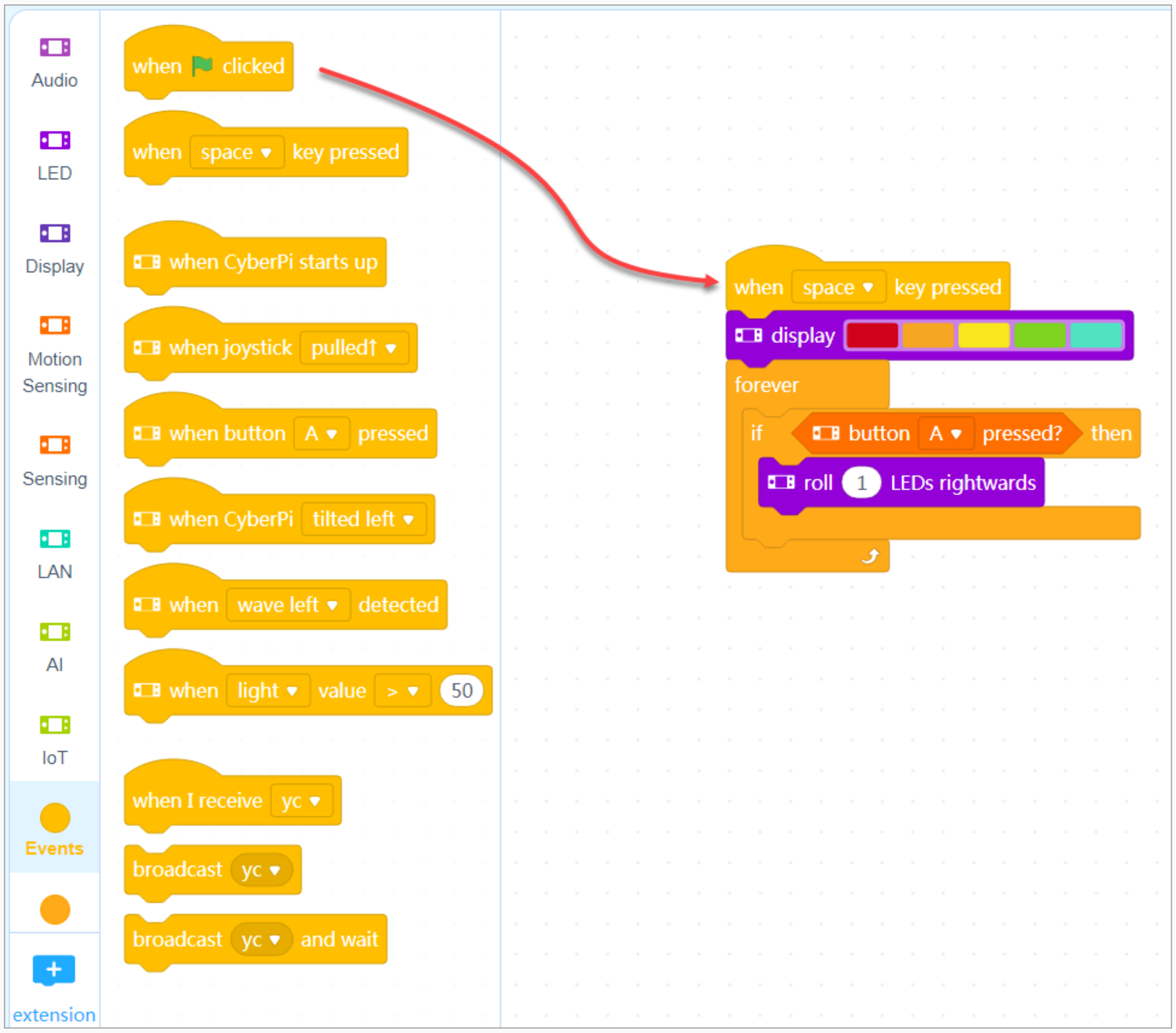
Set the initial colors to red, orange, yellow, green, and cyan.

(3) Set a condition for rolling the colors.



Each time button A on CyberPi is pressed, the colors roll from left to right by one position.

(4) Add an event that triggers the program.



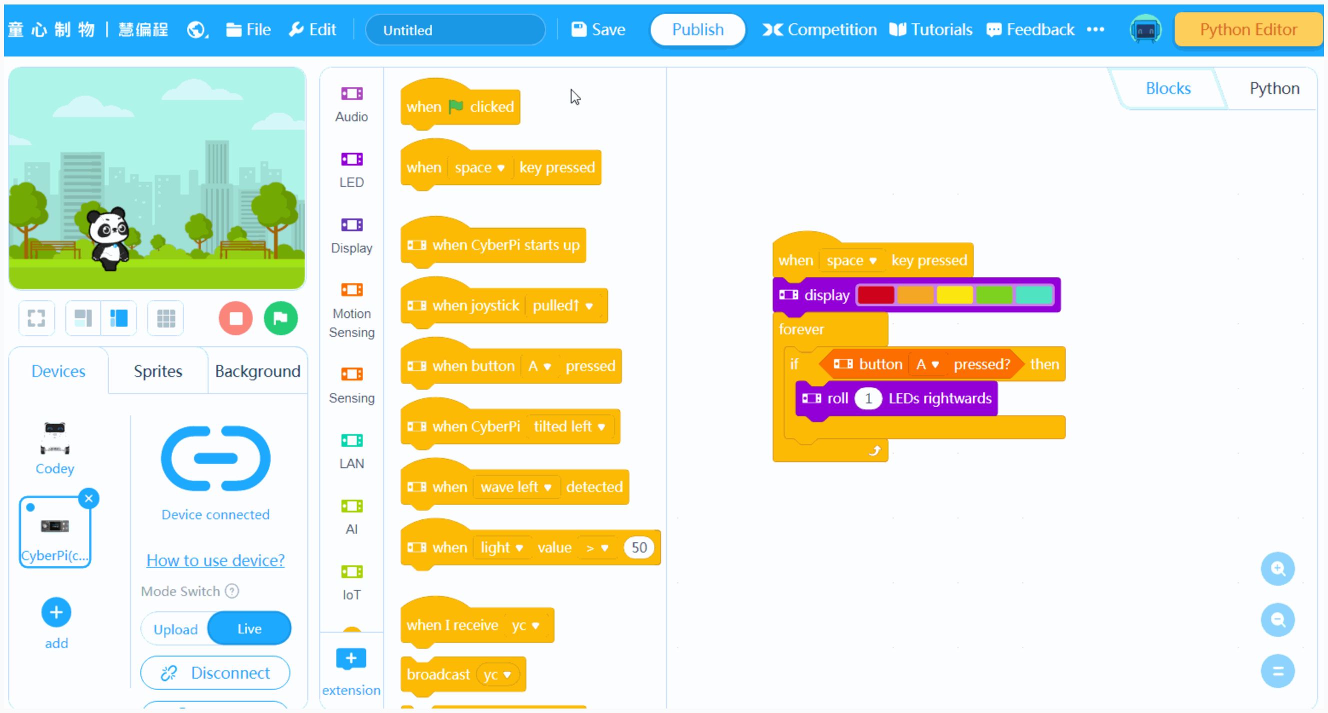
The program is to be executed when you press the space key.

The Marquee program is complete.

Press the space key, and then keep pressing button A to see the execution result of the program.

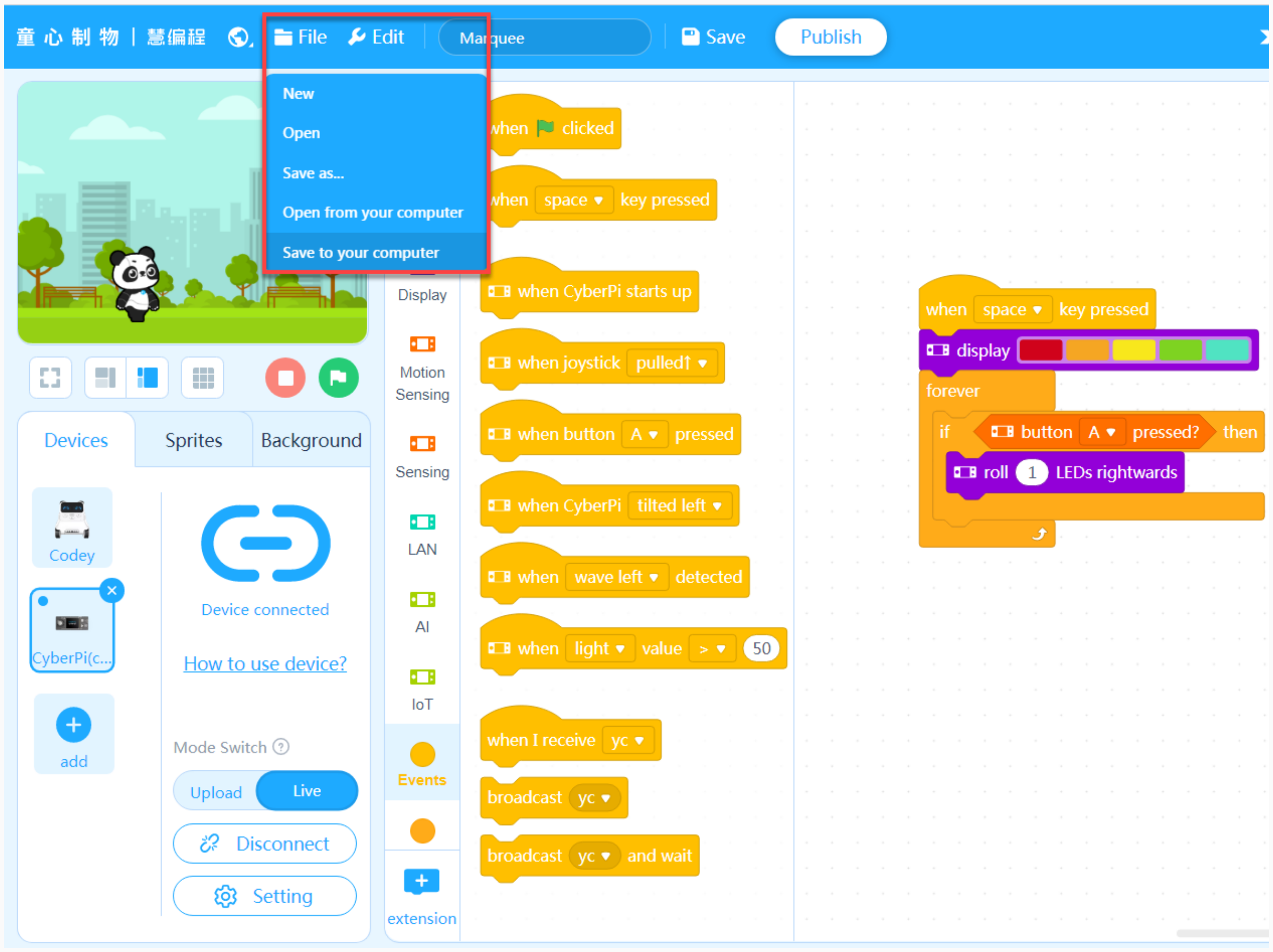
(5) Name the project and save it.

After saving a project, you can click your user image and choose **My Projects** to view it.



Note: To save projects, you need to sign in to mBlock 5 first.

You can also save your project to the local disk.



3.3 Python programming

Open mBlock-Python Editor on Google Chrome.

For details about how to program CyberPi on mBlock-Python Editor, see mBlock-Python Editor Online Help.

4. Courses for CyberPi

Coming soon!

5. Online help

After-sales services and technical support

If you encounter any product quality problems or find any parts missing or damaged when you open the package, or if you need any technical support, contact us for after-sales services through:

support@makeblock.com

(Service hours: 9:00—12:00 & 14:00—18:30 from Monday to Friday)

Feedback and suggestions

Should you have any feedback or suggestions on CyberPi, contact our R&D team through: cyber.list@makeblock.com

6. FAQs

How can I design a game with CyberPi?

To ensure user experience, only limited control permission on CyberPi's screen is provided for users currently. More functions are on the way.

Currently, you can use mBlock-Python Editor in combination with some Python libraries such as `pygame` to implement control over a Python game, using CyberPi as a remote control. You can find examples in the *mBlock-Python Editor Online Help*.

Alternatively, you can use the CyberPi Lab extensions to use the latest functions of CyberPi, for which you may need to update the firmware of CyberPi.

---

7. More information

CyberPi Operation Guide
Pocket Shield Operation Guide
CyberPi Series User Manual
Python API Documentation for CyberPi
mBlock 5 Online Help
mBlock-Python Editor Online Help

Pocket Shield Operation Guide

English

日本語

1. Know about Pocket Shield

Pocket Shield provides the following functions for CyberPi:

- Supplying power
- Extending functions

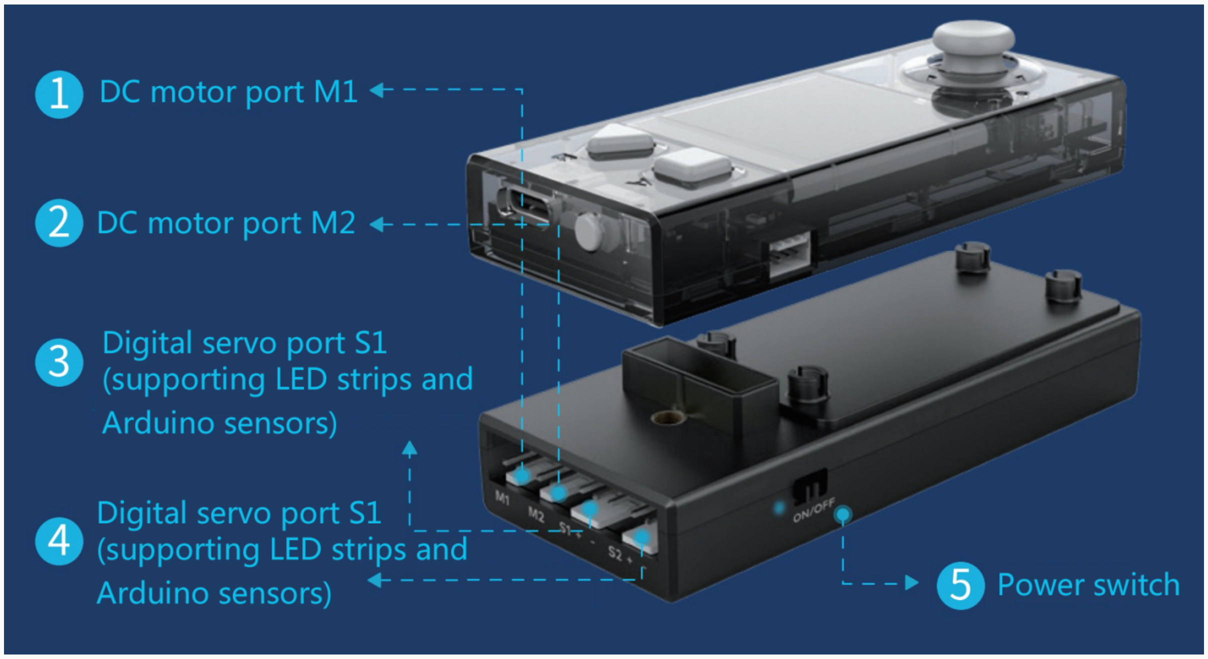
1.1 Supplying power

Pocket Shield is equipped with a rechargeable 800 mAh Li-ion battery that can be charged while supplying power for CyberPi. In general, Pocket Shield can supply power for CyberPi for more than four hours after being fully charged. The duration, however, depends on the load of the battery.

1.2 Extending functions

Pocket Shield provides two types of ports, namely 2-pin and 3-pin ports, which are designed with an anti-plug mechanism and the pin spacing of 2.54 mm. The 2-pin ports can be used to connect motors and the 3-pin ones can be used to connect servos or LEDs strips.

In addition, you can connect third-party sensors to Pocket Shield through DuPont wires.


1 DC motor port M1
2 DC motor port M2
3 Digital servo port S1 (supporting LED strips and Arduino sensors)
4 Digital servo port S1 (supporting LED strips and Arduino sensors)
5 Power switch
M1
M2
S1
S2
ON/OFF

Supported motors

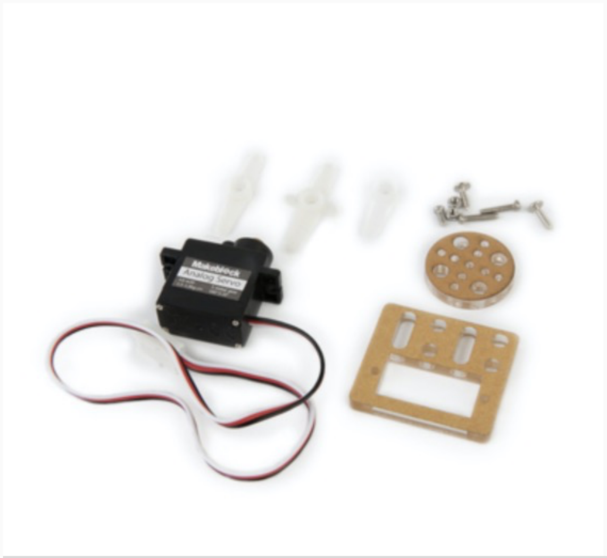
Motors developed by Makeblock are recommended.
Pocket Shield is compatible with third-party motors, but you need to pay attention to the start current and connector of a motor when you connect it to Pocket Shield.

Pin spacing	2.54 mm
Output voltage	5 V
Operating current	< 1.2 A
Max. transient current	2.4—4.8 A

Note: The TT motor developed by Makeblock for competition is too powerful that its start current is too high to work with Pocket Shield. Keep this in mind when you purchase motors.

Supported servos

Servos developed by Makeblock are recommended.
Pocket Shield is compatible with third-party servos, but slight angle deviations may occur due to the component protocols.



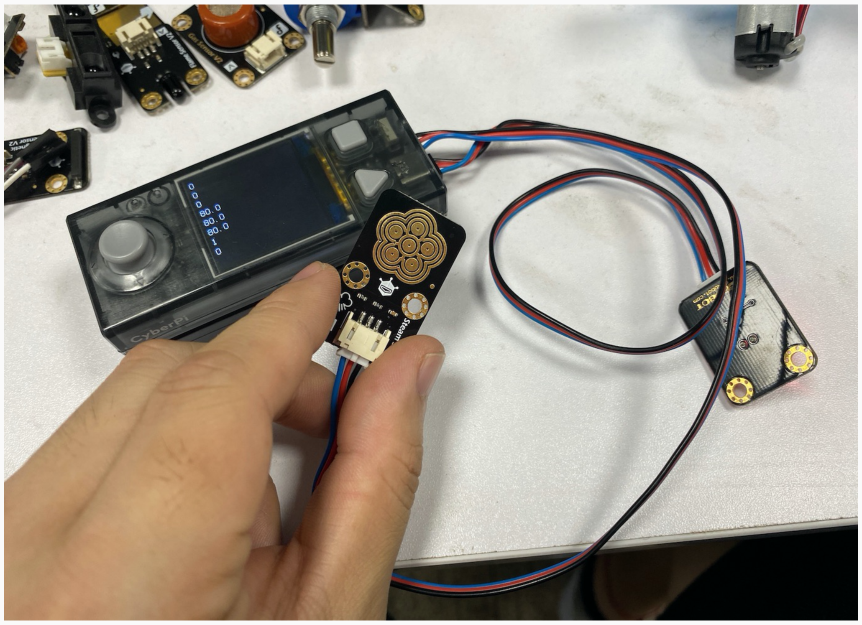
Pin spacing	2.54 mm
Operating voltage	5 V
Operating current	< 1.6 A

Supported LED strips

The digital servo ports can also be used to connect LED strips.
LED strips developed by Makeblock are recommended.
Pocket Shield is compatible with third-party LED strips, but control exceptions may occur due to the definition differences of high and low levels in the component protocols.

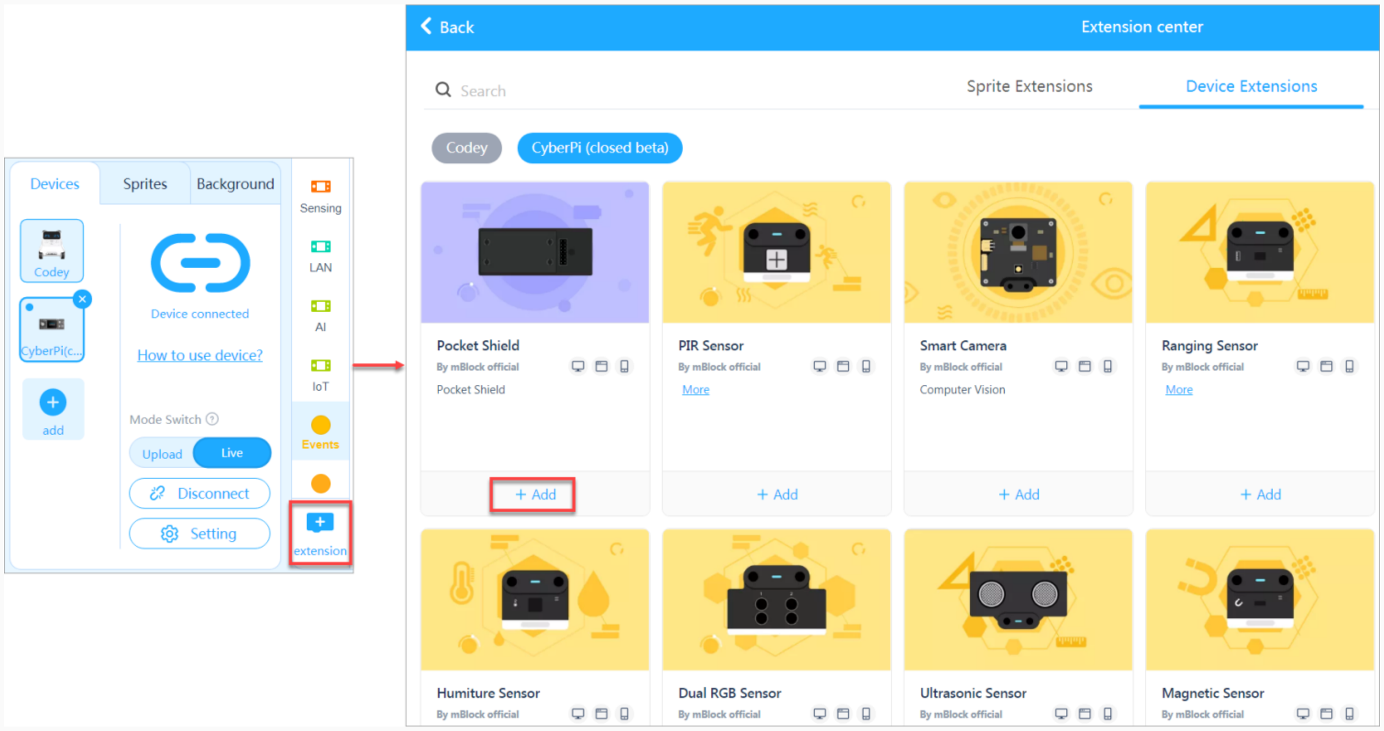
Supported third-party sensors

The output voltage of Pocket Shield is 5 V. It is well compatible with **Arduino digital and analog sensors and output modules**. The operating voltage of micro:bit sensors is 3.3 V, and therefore Pocket Shield may not work well with them.

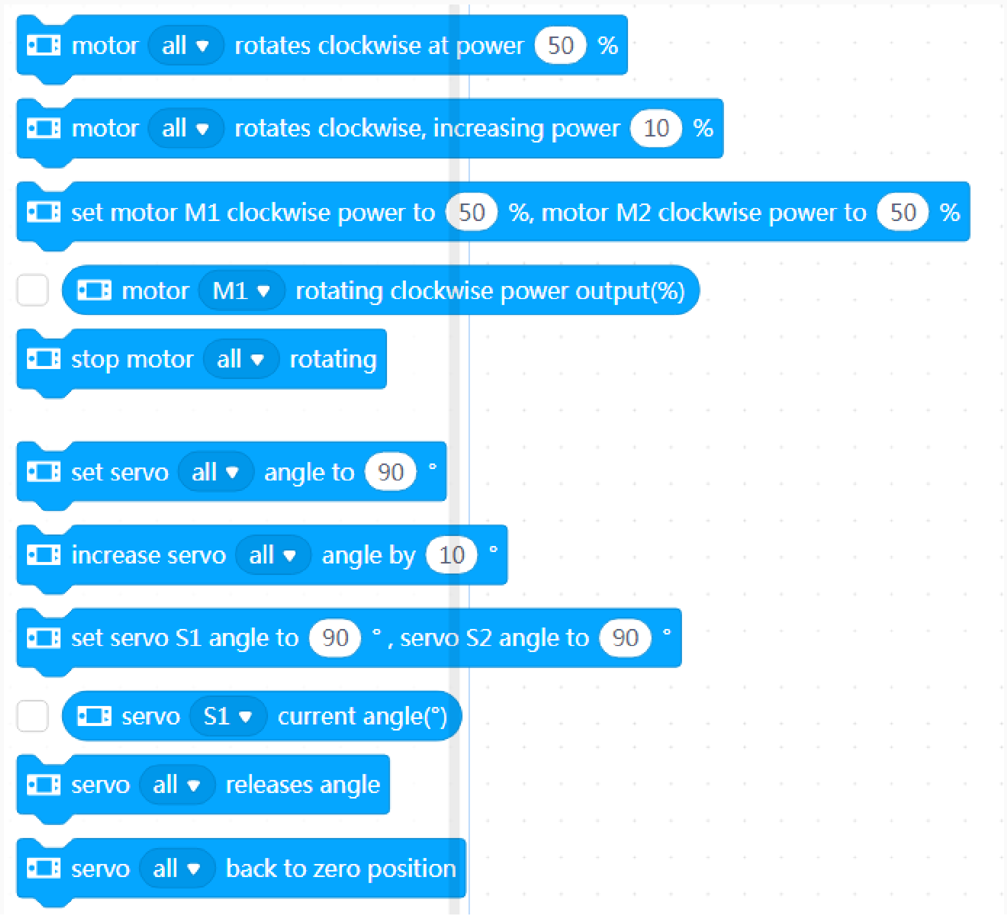

(DF sensor used)

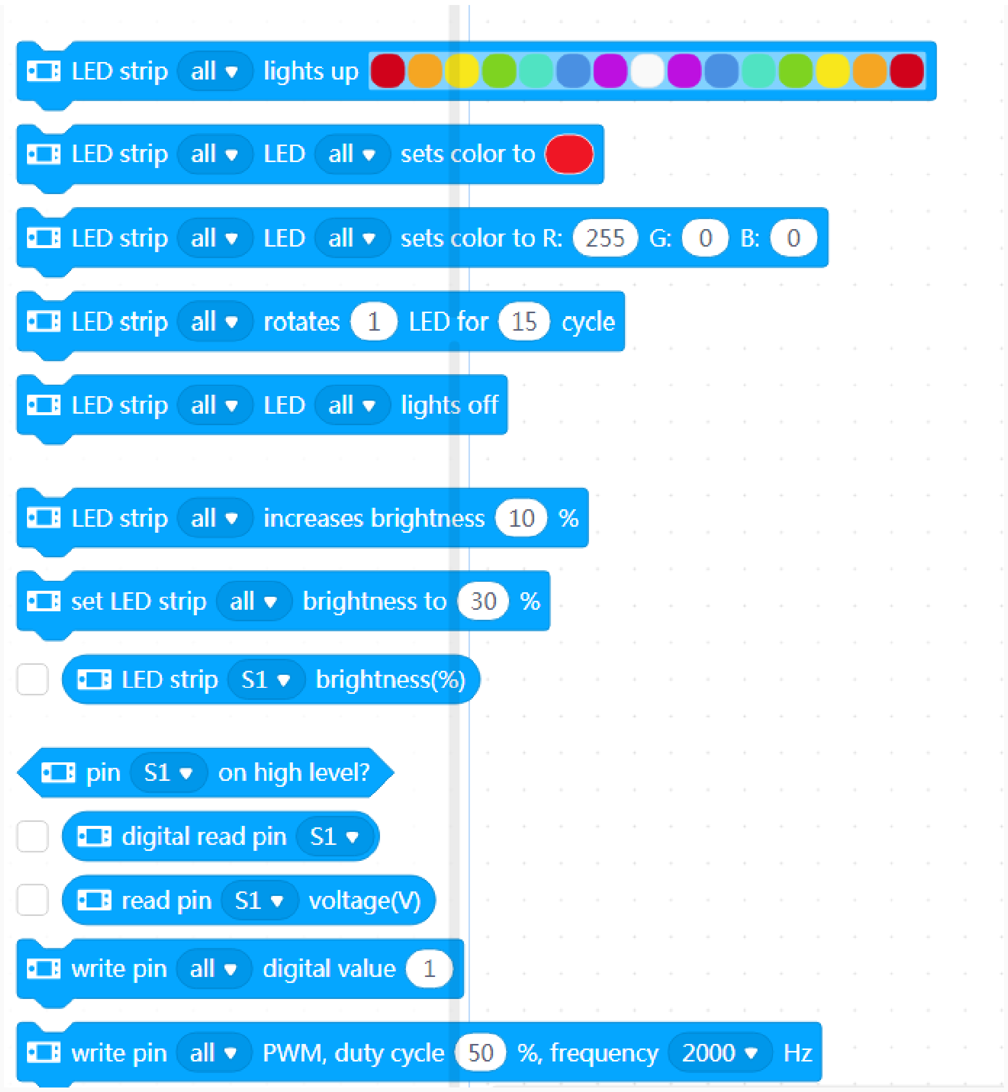
2. Add the Pocket Shield extension

Click **+ extension** and then click **+ Add** in the extension center to add the Pocket Shield extension.



The following blocks are provided in the Pocket Shield extension, indicating the functions that have been developed.


motor all ▾ rotates clockwise at power 50 %
motor all ▾ rotates clockwise, increasing power 10 %
set motor M1 clockwise power to 50 %, motor M2 clockwise power to 50 %
motor M1 ▾ rotating clockwise power output(%)
stop motor all ▾ rotating
set servo all ▾ angle to 90 °
increase servo all ▾ angle by 10 °
set servo S1 angle to 90 ° , servo S2 angle to 90 °
servo S1 ▾ current angle(°)
servo all ▾ releases angle
servo all ▾ back to zero position


LED strip all ▾ lights up ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●

LED strip all ▾ LED all ▾ sets color to ●

LED strip all ▾ LED all ▾ sets color to R: 255 G: 0 B: 0

LED strip all ▾ rotates 1 LED for 15 cycle

LED strip all ▾ LED all ▾ lights off

LED strip all ▾ increases brightness 10 %

set LED strip all ▾ brightness to 30 %

LED strip S1 ▾ brightness(%)

pin S1 ▾ on high level?

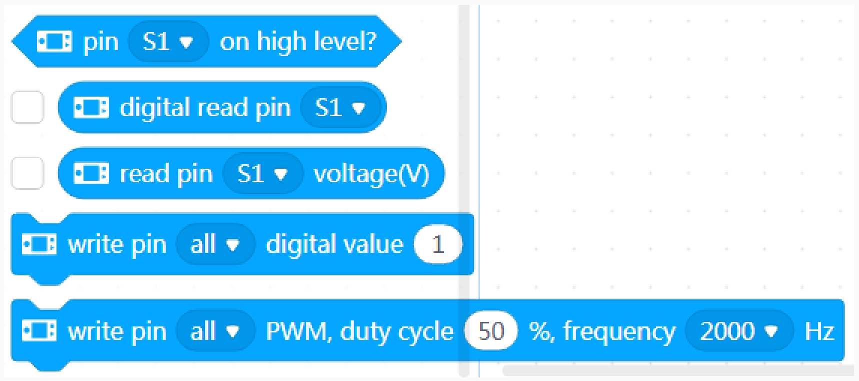
digital read pin S1 ▾

read pin S1 ▾ voltage(V)

write pin all ▾ digital value 1

write pin all ▾ PWM, duty cycle 50 %, frequency 2000 ▾ Hz

Among them, the following ones are developed for third-party sensors.


pin S1 ▾ on high level?

digital read pin S1 ▾

read pin S1 ▾ voltage(V)

write pin all ▾ digital value 1

write pin all ▾ PWM, duty cycle 50 %, frequency 2000 ▾ Hz

3. Python APIs

For information about the APIs for Pocket Shield, see "APIs for Pocket Shield" in the *Python API Documentation for CyberPi*.

4. Online help

After-sales services and technical support

If you encounter any product quality problems or find any parts missing or damaged when you open the package, or if you need any technical support, contact us for after-sales services through:
support@makeblock.com
(Service hours: 9:00—12:00 & 14:00—18:30 from Monday to Friday)

Feedback and suggestions

Should you have any feedback or suggestions on CyberPi, contact our R&D team through:
cyber.list@makeblock.com

5. FAQs

How long does it take to fully charge Pocket Shield?

Within two hours. It can be charged faster when powered off.

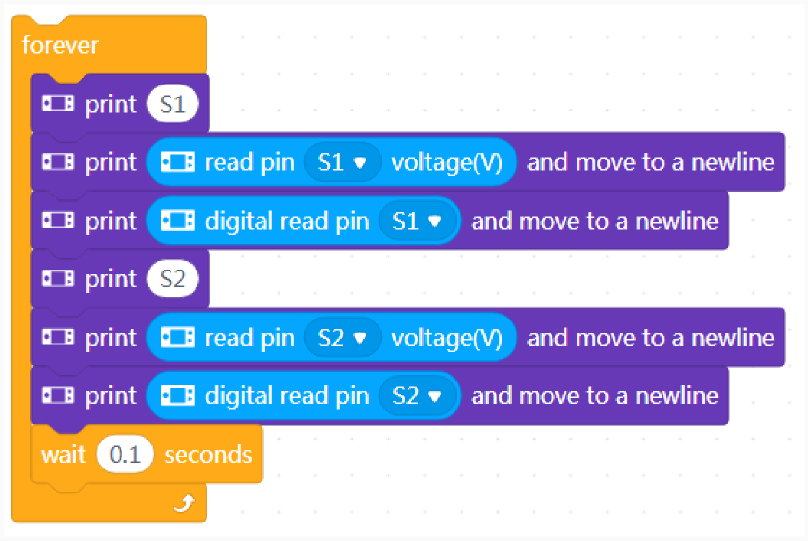
Can I use CyberPi and Pocket Shield to participate in the Arduino competition?

Working with Pocket Shield, CyberPi supports Arduino modules and parts, but it is a new-generation open-source device designed based on China-made chips instead of Arduino hardware.

CyberPi is designed with high compatibility and can communicate with Arduino main control boards through serial ports or Bluetooth. Therefore, you can use CyberPi as an input or output part for Arduino main control boards to implement powerful and abundant AI and IoT functions in your Arduino projects.

What if the voltage of pins can't be properly obtained?

This is caused by the errors that occur during the switching of pin modes. A new firmware version will be released in August to solve this problem. At present, you can refer to the following program to switch the pins to the correct states to ensure proper voltage output.



```
forever
    print S1
    print read pin S1 voltage(V) and move to a newline
    print digital read pin S1 and move to a newline
    print S2
    print read pin S2 voltage(V) and move to a newline
    print digital read pin S2 and move to a newline
    wait 0.1 seconds
```

6. More information

CyberPi Operation Guide

Pocket Shield Operation Guide

CyberPi Series User Manual

Python API Documentation for CyberPi

mBlock 5 Online Help

mBlock-Python Editor Online Help

Operation Guide of the "Display +" Extension for CyberPi

Foreword

Hello, I'm Alex Yu, the product manager of CyberPi. I'm glad to introduce the **Display +** extension to you. It is the key feature update in iteration 003 of CyberPi.

This page describes the details about how to use the **Display +** extension and provides some example programs, both simple and complicated ones. The content on this page will be continuously updated and improved to help you use the **Display +** extension more efficiently.

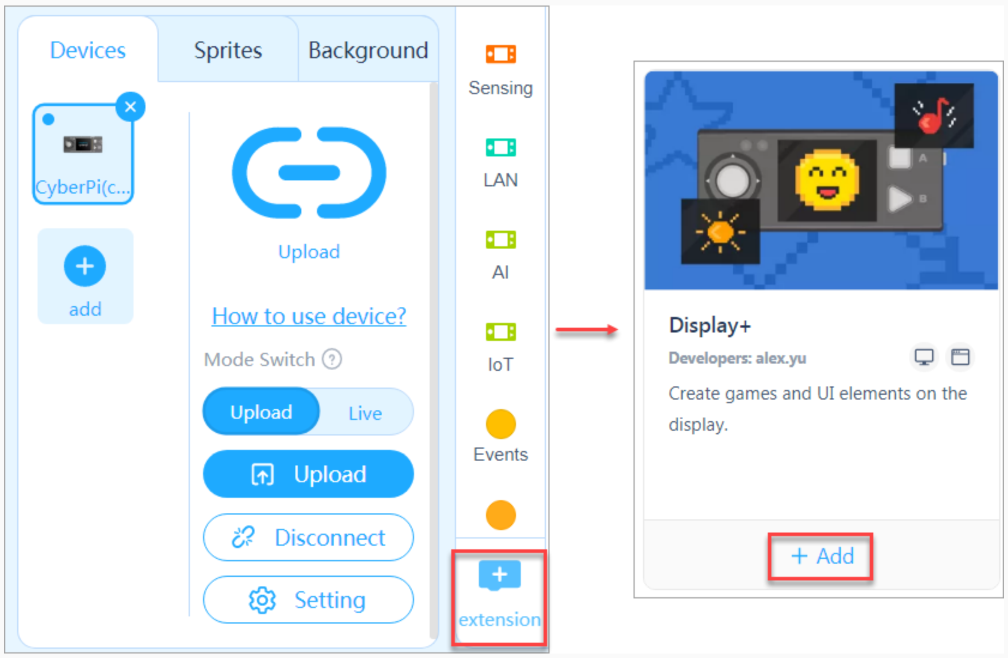
Notice

- The **Display +** extension is available only for firmware 003 or later of CyberPi. You can update the firmware of CyberPi in OTA mode or on mBlock 5.
- Currently, CyberPi supports only mBlock 5 on the web. The PC client and mobile app will be launched around the end of September.
- The current **Display +** extension“ is a beta version and provides only some basic functions. Exceptions may occur when you use it, and its design still needs some improvement. Contact us if you have any suggestions or feedback on the improvement of this extension.
- Efforts will be made to ensure the compatibility of later official versions with projects created in the earlier versions, but the compatibility can't be ensured.
- In the short term, this extension will support only the **Upload** mode and MicroPython programming due to the complexity of object-oriented programming.

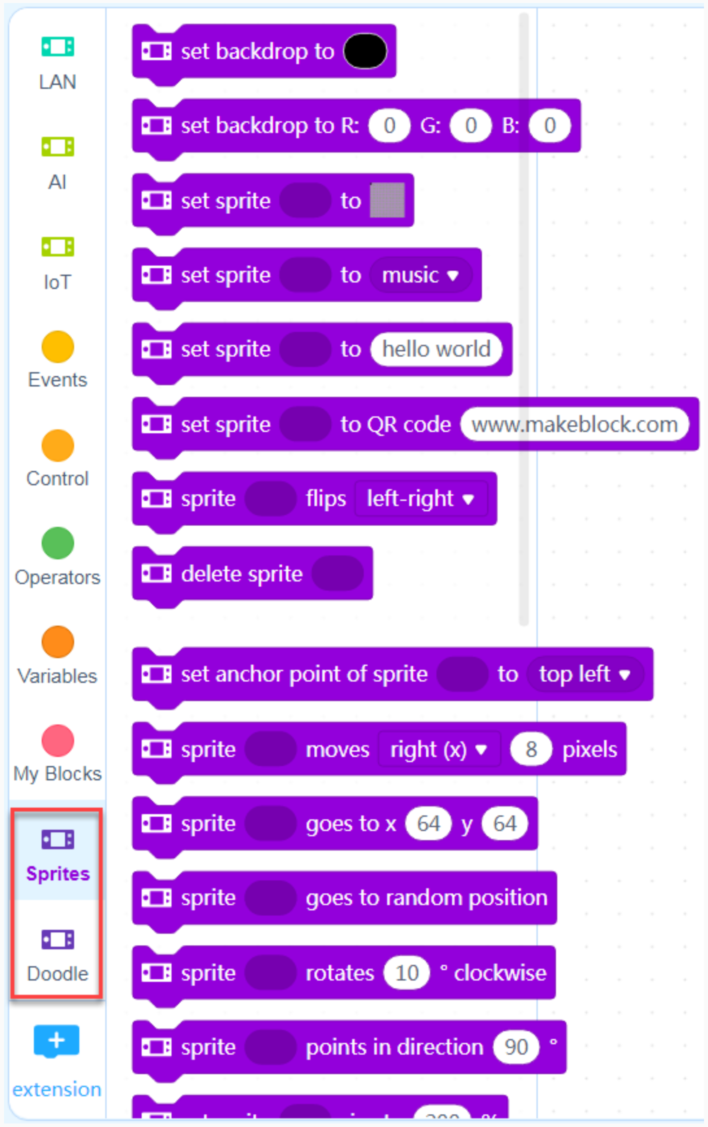
Use the "Display +" extension

Add the Display + extension

After connecting CyberPi to mBlock 5, click **+ extension** at the bottom of the blocks area, and then click **+ Add** at the bottom of **Display +**.



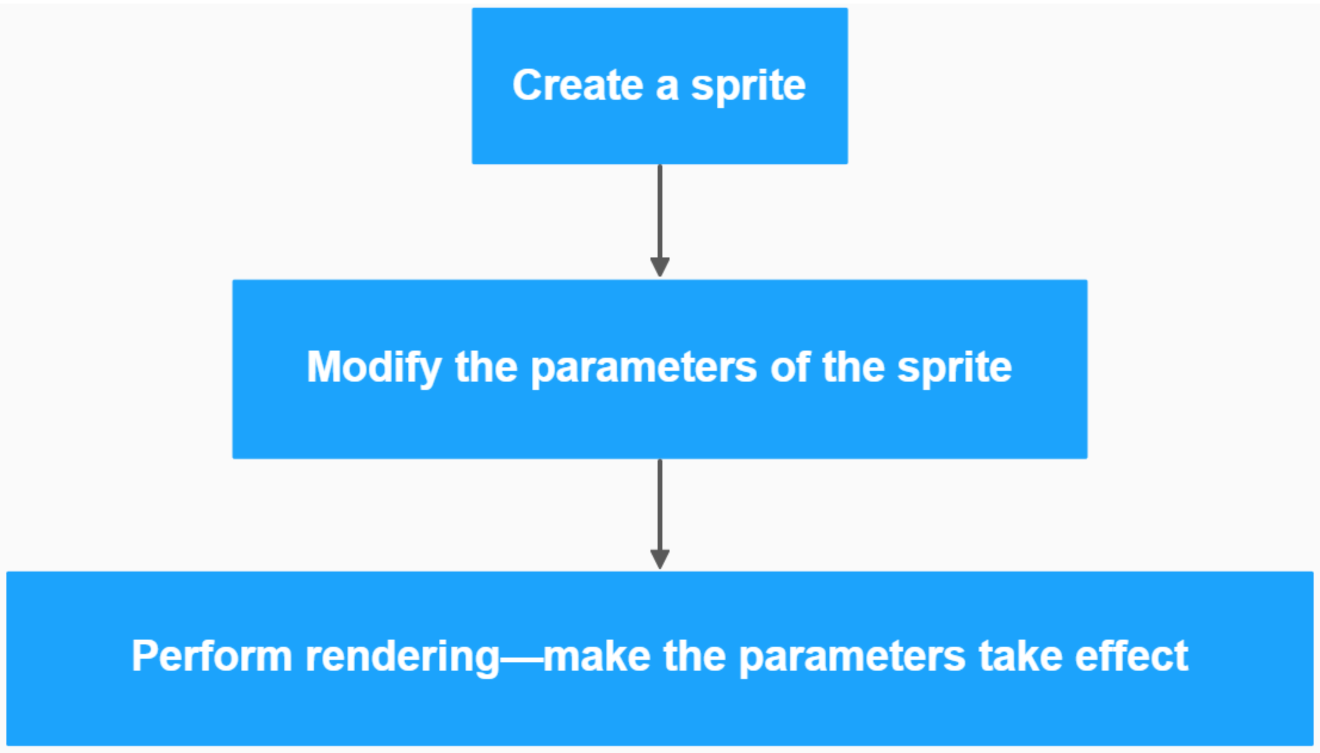
After adding the extension, you can see the **Sprites** and **Doodle** blocks.


LAN
AI
IoT
Events
Control
Operators
Variables
My Blocks
Sprites
Doodle
extension
set backdrop to
set backdrop to R: 0 G: 0 B: 0
set sprite to
set sprite to music
set sprite to hello world
set sprite to QR code www.makeblock.com
sprite flips left-right
delete sprite
set anchor point of sprite to top left
sprite moves right (x) 8 pixels
sprite goes to x 64 y 64
sprite goes to random position
sprite rotates 10 ° clockwise
sprite points in direction 90 °

The **Display +** extension extends the programmable functions of CyberPi's screen. With this extension, you can design apps, games, and more complicated charts on CyberPi.

The **Display +** extension adopts the concept of object-oriented programming (OOP), and therefore, when you use this extension for programming, you need to define an object first.

The following figure describes the three basic steps for using the **Display +** extension for programming.

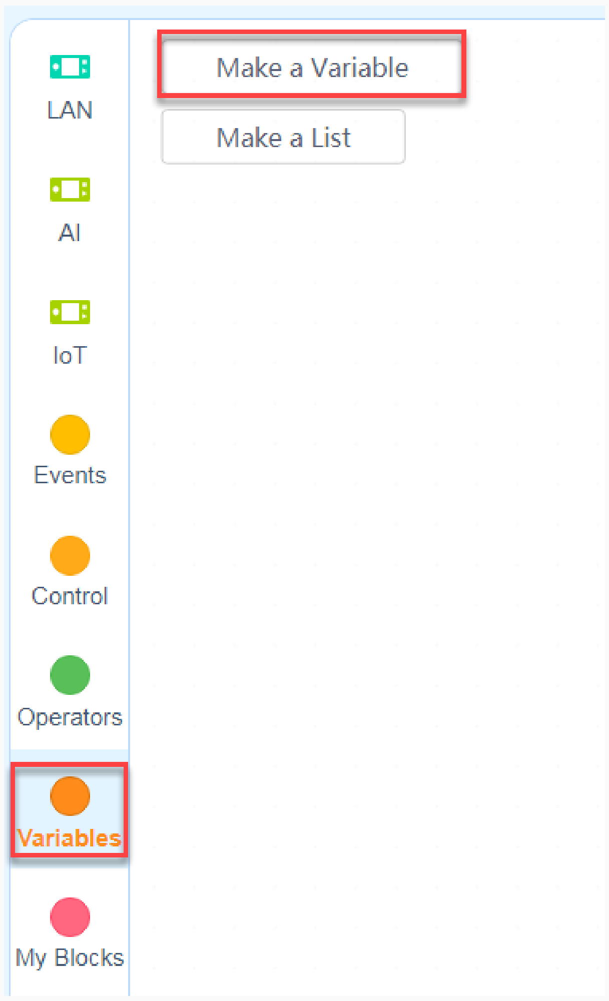

Create a sprite
Modify the parameters of the sprite
Perform rendering—make the parameters take effect

Create a sprite

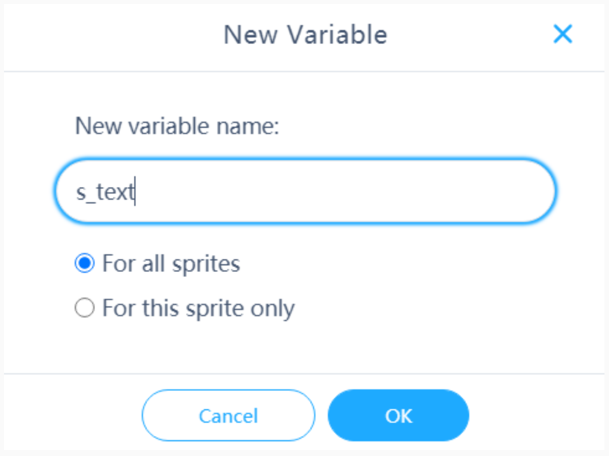
Create a variable in the Variables category, and then define the variable as a sprite.

Tips: You can name a variable to be defined as a sprite in the format of **s_*XXX*** to distinguish it from an ordinary one. For example, **s_text** indicates a text variable to be defined as a sprite, and **text** indicates an ordinary text variable.

1. Select the **Variables** category and click **Make a Variable**.



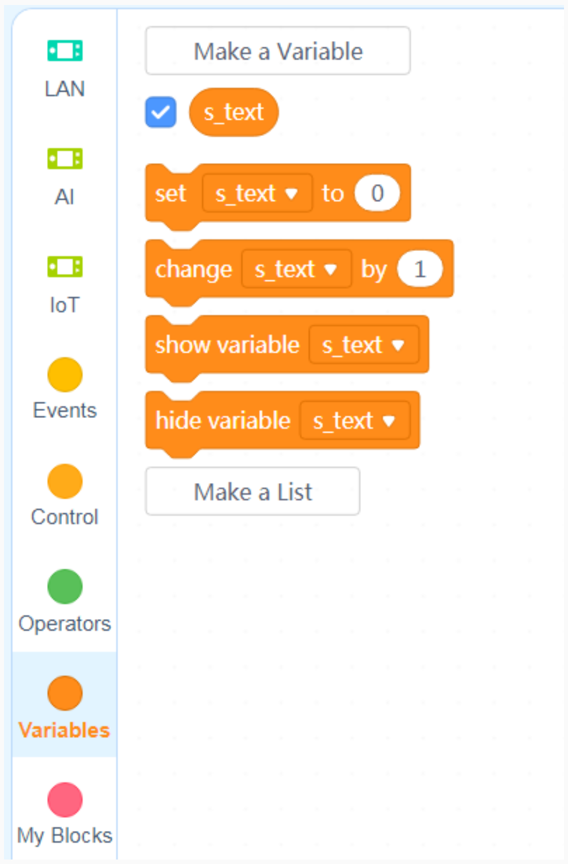
2. In the **New Variable** dialog box that appears, enter the variable name and click **OK**.



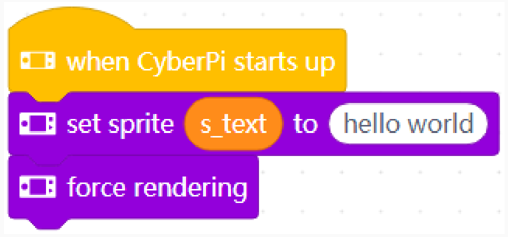
New variable name: s_text

[✓ For all sprites]
For this sprite only

The created variable and its related blocks appear in the **Variables** category.



3. Set the value of the created variable and define it as a sprite through blocks, as shown in the following example.



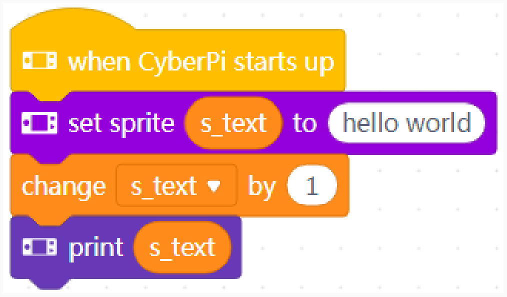
```
when CyberPi starts up
set sprite (s_text) to (hello world)
force rendering
```

Summary:
The steps for creating a sprite are as follows:
creating a variable -> assigning a value and defining it as a sprite -> performing redering to display the sprite

Note that the sprite can be displayed only when you use the force rendering block.

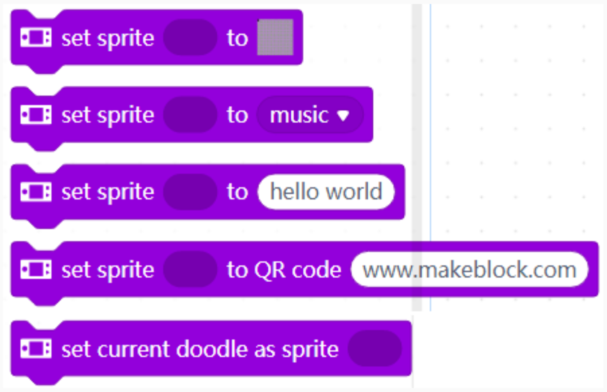
If a variable is defined as a sprite, you can no longer perform arithmetic operations on it because data of the sprite type does not support arithmetic operations. Errors may be reported if you do.

For example, an error is reported when you execute the following program.

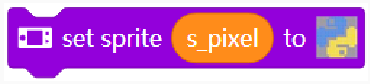
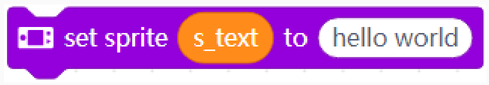
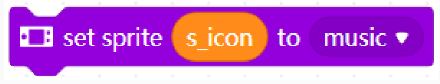

```
when CyberPi starts up
set sprite (s_text) to (hello world)
change [s_text ▾] by (1)
print (s_text)
```

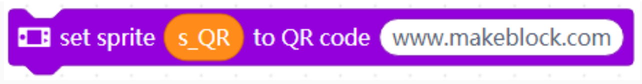
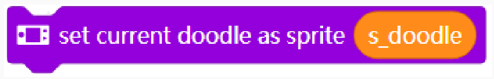
Types of sprites

We have defined multiple types of sprites to facilitate your creation of projects. Generally, you can use the following blocks to define sprites.


```
set sprite ( ) to [ ]
set sprite ( ) to [music ▾]
set sprite ( ) to (hello world)
set sprite ( ) to QR code (www.makeblock.com)
set current doodle as sprite ( )
```

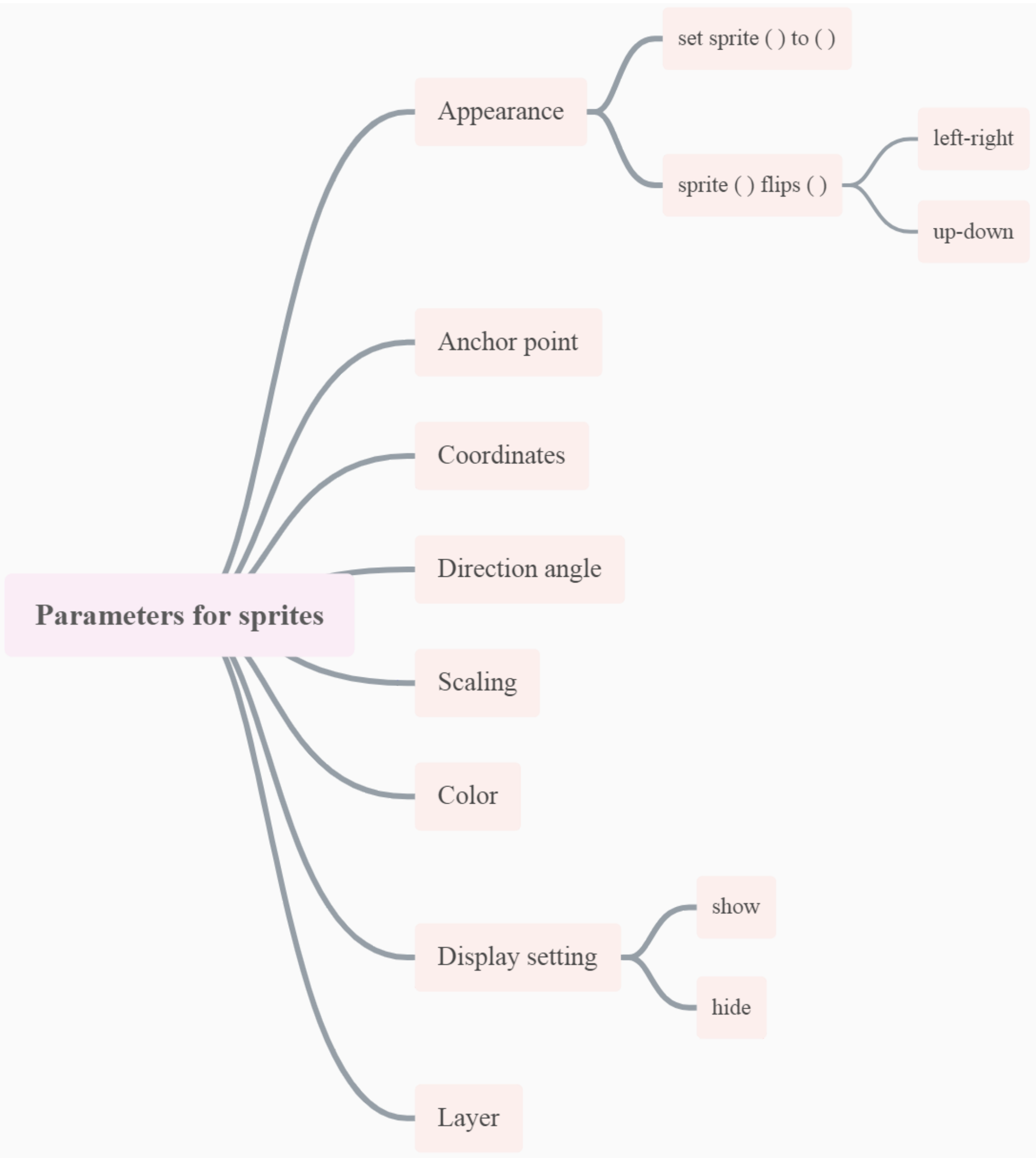
The following table describes the currently available types of sprites.

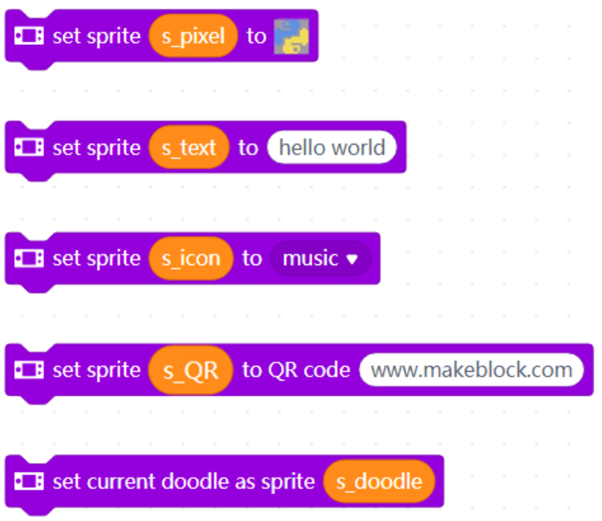
Type	Block	Description
Matrix	 set sprite (s_pixel) to	Defines a matrix pattern as a sprite **Tips:** We have restricted the aspect ratio of matrix patterns for MicroPython programming, allowing you to create larger images, for example, a 128 × 128 (pixel) image.
Text	 set sprite (s_text) to (hello world)	Defines a text as a sprite
Icon	 set sprite (s_icon) to [music ▾]	Defines a preset icon as a sprite

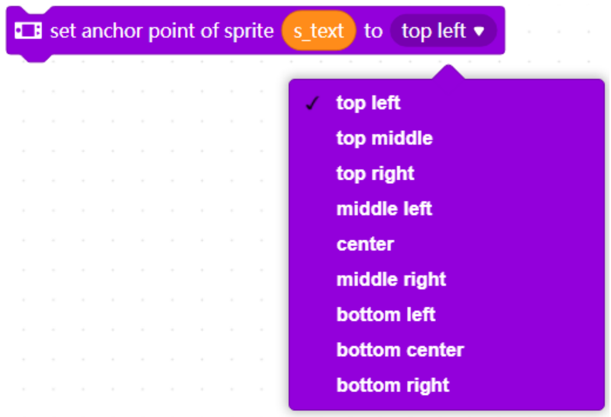
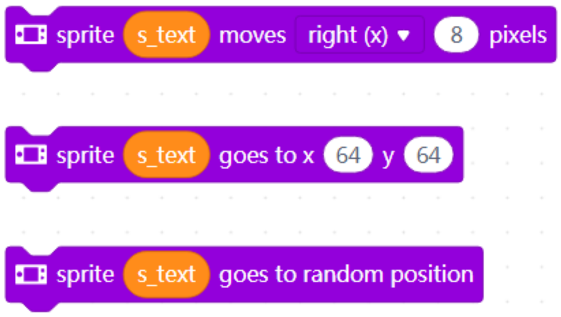
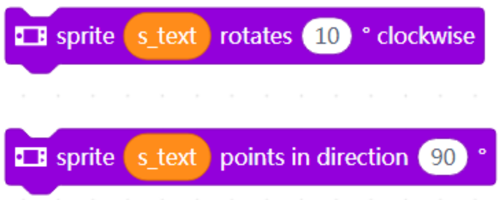
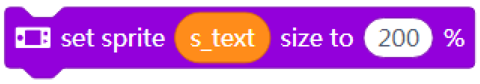
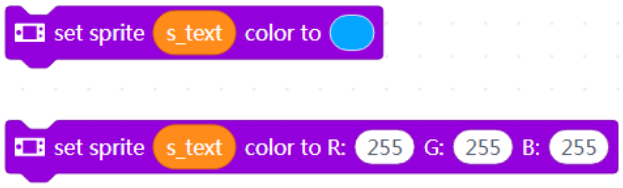
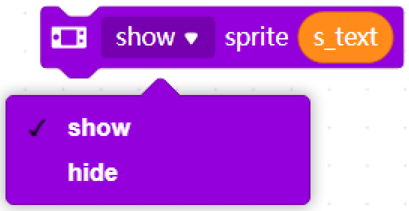
QR code	 set sprite s_QR to QR code www.makeblock.com	Defines a QR code as a sprite
Doodle	 set current doodle as sprite s_doodle	Defines a sketch as a sprite **Tips:** The **Doodle** blocks simulate the structures of Turtles drawing, which work best when you use them to draw geometric figures.

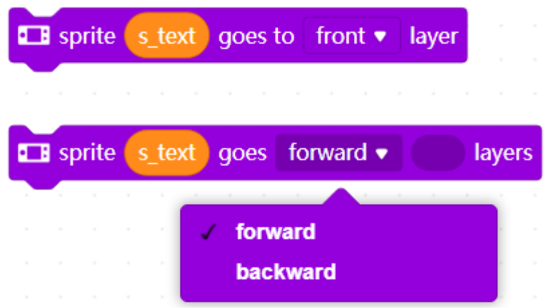
Modify parameters for sprites

You can modify the parameters for a sprite through blocks or APIs. The following figure shows the currently available parameters for sprites.



Parameter	Block	Description
Appearance	 set sprite s_pixel to set sprite s_text to hello world set sprite s_icon to music ▾ set sprite s_QR to QR code www.makeblock.com set current doodle as sprite s_doodle	The appearance of the sprite depends on the blocks set sprite ( ) to ( ) and sprite ( ) flips ( ).

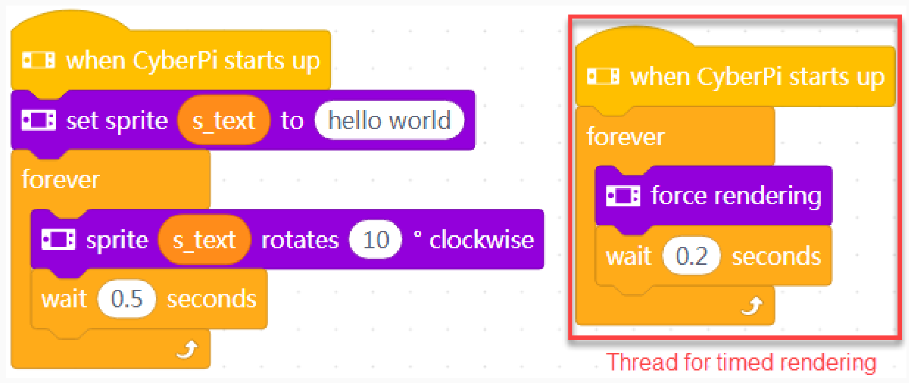
Anchor point	 set anchor point of sprite s_text to top left ▾ ✓ top left top middle top right middle left center middle right bottom left bottom center bottom right	The anchor point determines the origin of the sprite and its center of rotation.
Coordinates	 sprite s_text moves right (x) ▾ 8 pixels sprite s_text goes to x 64 y 64 sprite s_text goes to random position	The coordinates determine the position of the sprite. You can move a sprite by changing its coordinates.
Direction angle	 sprite s_text rotates 10 ° clockwise sprite s_text points in direction 90 °	The direction angle determines the direction the sprite faces. You can rotate a sprite by changing its direction angle.
Scaling	 set sprite s_text size to 200 %	This block determines the ratio of the to-be-displayed sprite to the original size of the sprite.
Color	 set sprite s_text color to set sprite s_text color to R: 255 G: 255 B: 255	You can put a color layer over a sprite. For example, you can change the color of a text sprite from white to red.
Display setting	 show ▾ sprite s_text ✓ show hide	This block can be used to show or hide a sprite.
Layer		These blocks can be used to place a sprite on the specified layer.

	 sprite s_text goes to front layer sprite s_text goes forward layers forward backward	

Tips: After a sprite is created, its parameters are set to the default values. When it is changed by using the block set sprite ( ) to ( ), all the other properties remain, except the reflecting effect. The reflecting effect is part of the appearance of the sprite.

Rendering

After modifying the properties of a sprite, you need to perform rendering on it to make the new properties take effect, that is, use the block force rendering.
Each time of rendering is similar to generate a frame in animation. Using the force rendering function properly can make better use of the hardware performance.
CyberPi can execute multiple threads simultaneously, and therefore, you can write a thread to implement timed rendering.
The following shows an example.



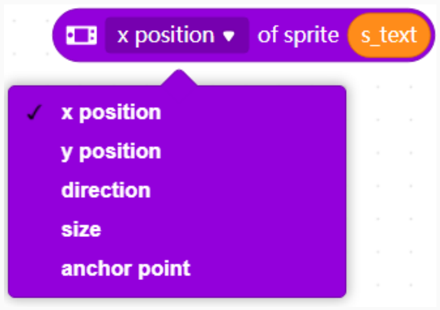
The implementation of the rendering mechanism may be different from other graphical programming logic. We have also tried to design an automatic rendering mechanism but haven't released it due to its stability.

Others

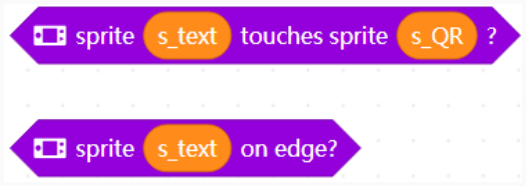
Detection

Some blocks are designed for users to obtain the status information of sprites. The following describes some of these blocks.

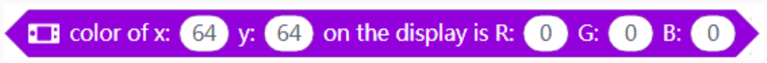
To obtain status information of a sprite, use:



To determine whether a sprite touches another one or the edge of the screen, use:

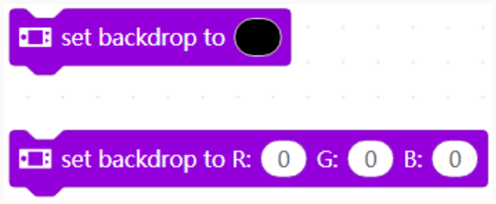


To determine the background color of the screen, use:



Background color setting

To set the color of the background, use:



Note: After setting the color, you need to use the block force rendering to make the setting take effect.

Example programs

Aircraft War

In this example, detection of touching and self-defined matrix patterns are used.

📎 Aircraft War V1.4.rar

Fun with Arithmetic

In this example, the function for defining sprites is used to implement the alternation of questions and answers.

📎 Fun with Arithmetic V3.1.rar

CyberPi Innovation Add-on Kit

Open-source Materials

CyberPi is an open-source device.
This section provides the open-source materials that can help you obtain the underlying permissions on CyberPi, allowing you to:

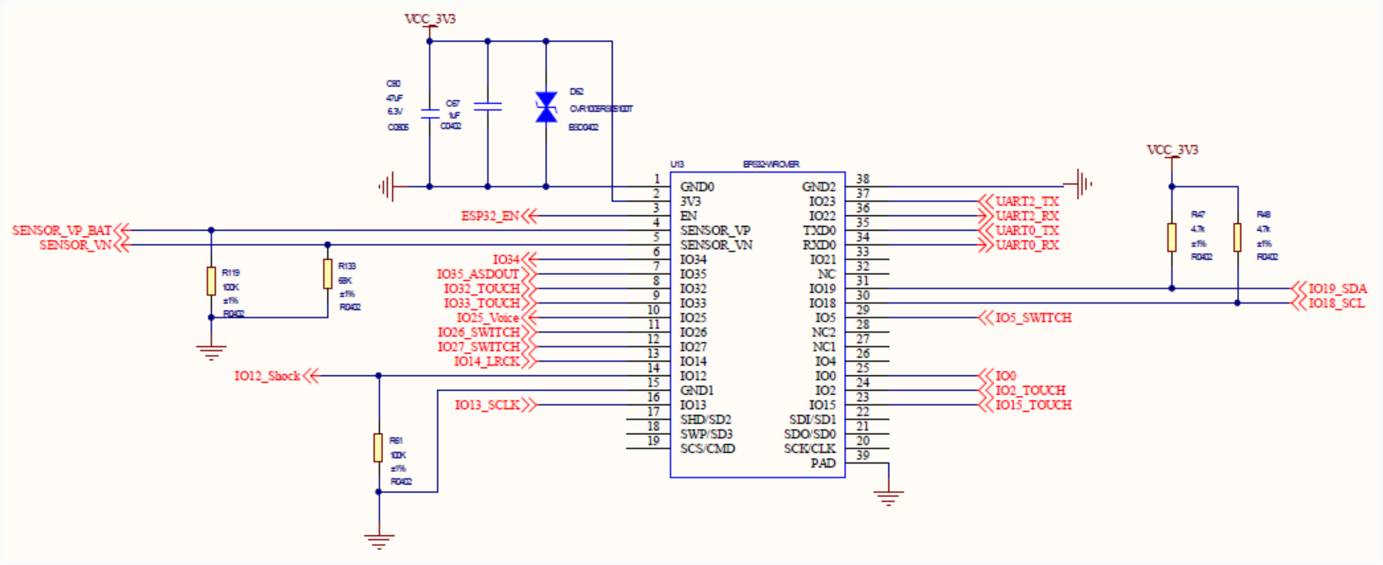
- Develop functions for CyberPi
- Design blocks in Extension Builder for CyberPi
- Enable CyberPi to work with third-party hardware
- Develop extension boards and electronic modules for CyberPi

Device schematic diagram

The schematic diagram of a device can help you understand its hardware design.

Schematic diagram of Halocode

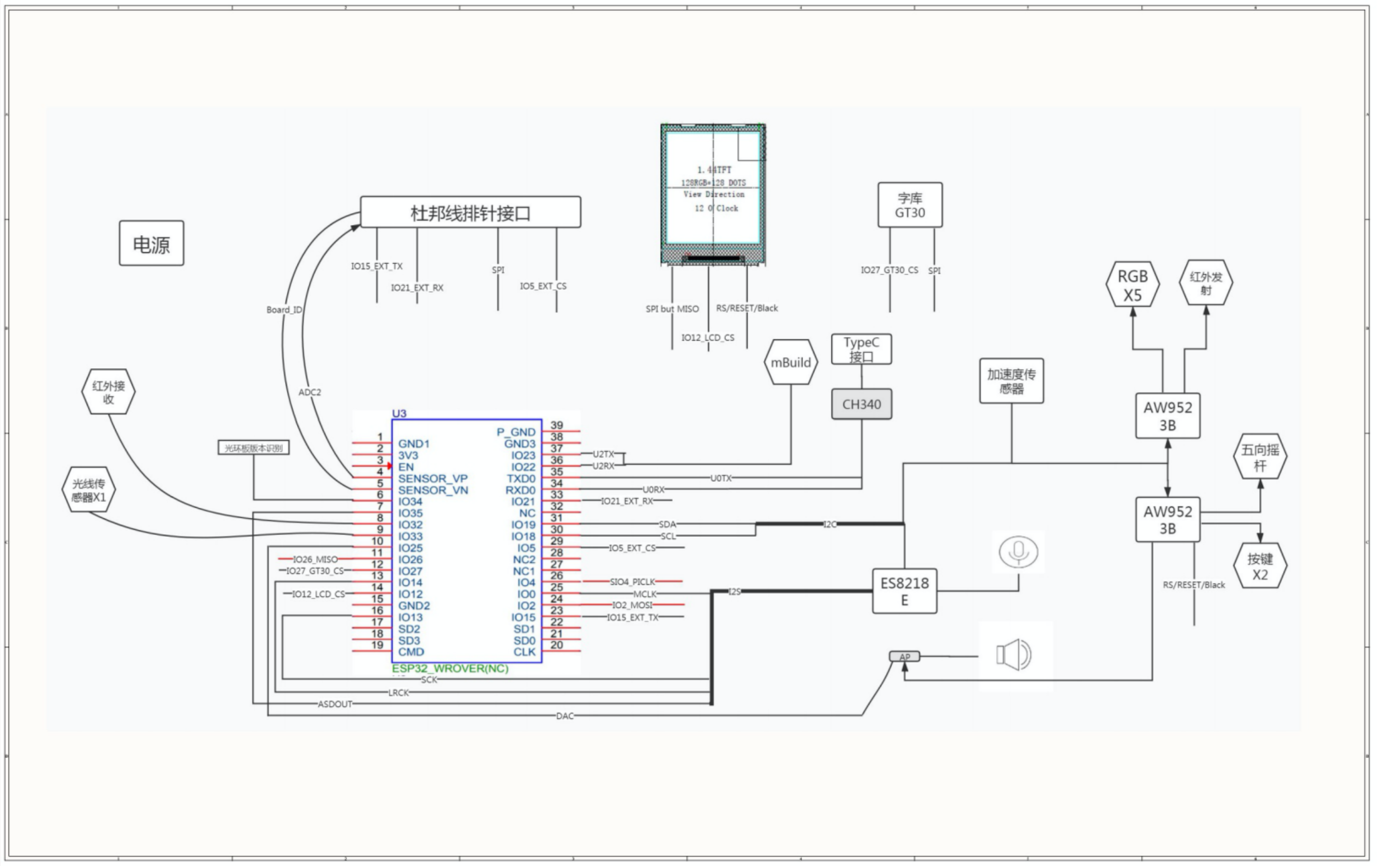
To understand the hardware design of Halocode, refer to its schematic diagram, as shown in the following figure.



📎 halocode_V1_0.pdf

[For non-Yuque users, click halocode_V1_0.]

Schematic diagram of CyberPi



📎 CYBERPI_V1_X (1).pdf [For non-Yuque users, click CYBERPI_V1_X.]

📎 CyberPi_V1_0_PCB.pdf [For non-Yuque users, click CyberPi_V1_0_PCB.]

📎 CyberPi_Gerber.rar [For non-Yuque users, click CyberPi_Gerber.]

Hardware License: 📎 License_cern_ohl_p_v2.pdf [For non-Yuque users, click License_cern_ohl_p_v2.]

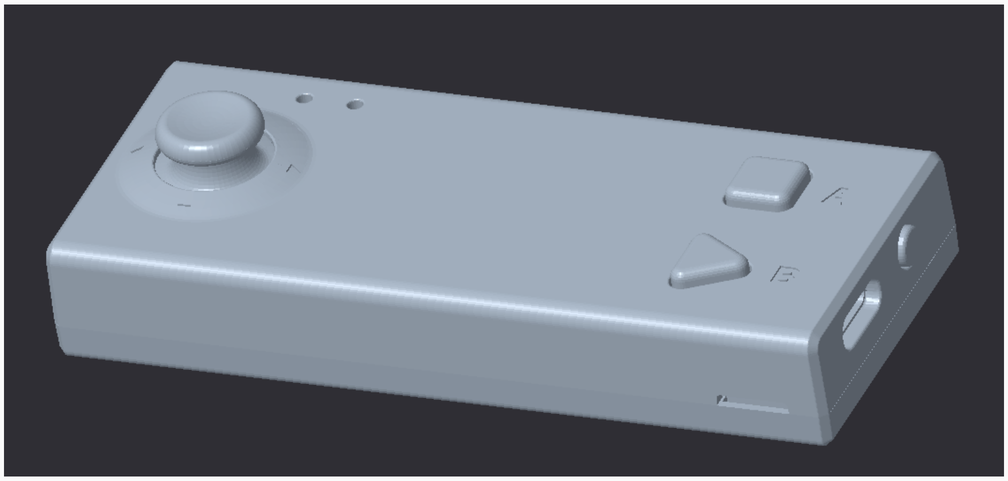
Software License: GPL v3

Documentation License: CC BY-SA 4.0 International

3D model

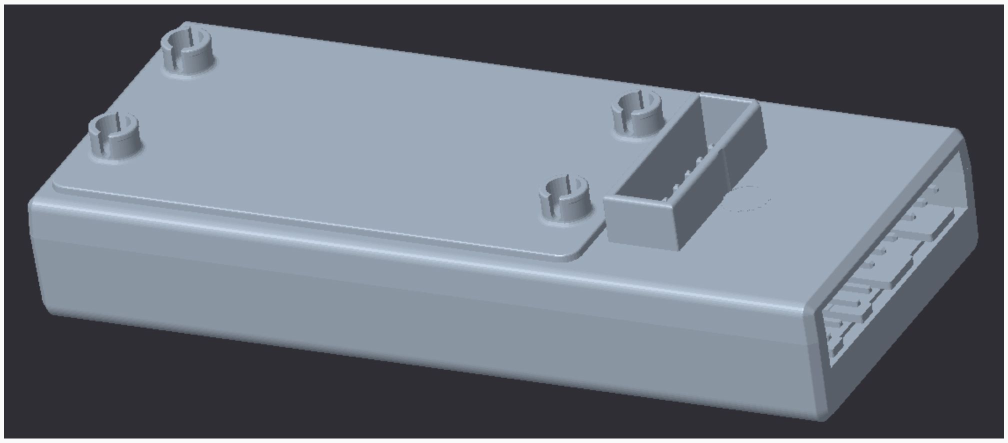
If you want to design and print some 3D models to work with CyberPi, refer to the 3D model of CyberPi.

3D model of CyberPi



📎 CyberPi.rar [For non-Yuque users, click CyberPi.]

3D model of Pocket Shield



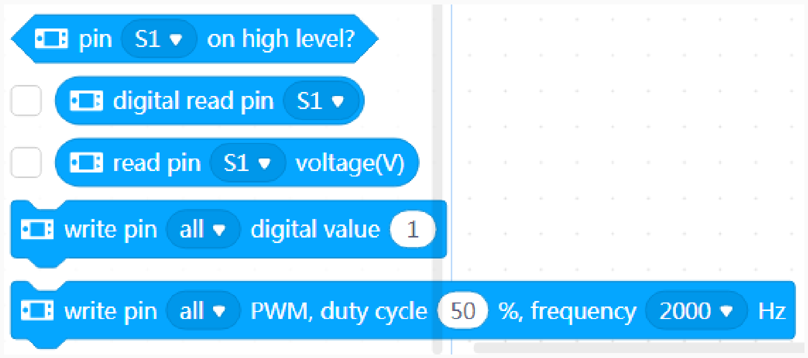
📎 Pocket Shield.rar [For non-Yuque users, click Pocket Shield.]

Third-party compatibility

You can use the 3-pin ports on Pocket Shield to connect and control third-party parts or modules, such as Arduino modules.

Related blocks and Python APIs are provided, as described in the following.

Blocks


pin S1 on high level?
digital read pin S1
read pin S1 voltage(V)
write pin all digital value 1
write pin all PWM, duty cycle 50 %, frequency 2000 Hz

Python APIs

`cyberpi.pocket.write_digital(val, port)`

Sets the digital input for the specified pin(s)

Parameters:

- *val:* digital input
- *port:* `int` or `str`, port where the pins are located

  Setting range:

  `all`

  `s1`

  `s2`

  `S1`

  `S2`

  `1`

  `2`

`cyberpi.pocket.read_digital(port)`

Obtains the digital input for the specified pin

Parameter:

- *port:* `int` or `str`, port where the pins are located

  Setting range:

  `s1`

  `s2`

  `S1`

  `S2`

  `1`

  `2`

An `int` value ranging from `0` to `1` is returned, where `0` indicates a low electrical level and `1` indicates a high electrical level.

`cyberpi.pocket.set_pwm(duty, frequency, port)`

Sets the specified pin(s) to output PWM signals with the specified frequency and duty cycle

Parameters:

- *duty:* `int`, duty cycle of the PWM signals to be output; setting range: `0-100`, in percentage
- *frequency:* `int`, frequency of the PWM signals to be output; setting range: `1-2000`, in Hz
- *port:* `int` or `str`, port where the pins are located

  Setting range:

  `all`

  `s1`

  `s2`

  `S1`

  `S2`

  `1`

  `2`

`cyberpi.pocket.read_analog(port)`

Obtains the voltage at the specified pin

Parameter:

- *port:* `int` or `str`, port where the pins are located

  Setting range:

  `s1`

  `s2`

  `S1`

  `S2`

  `1`

  `2`

A `float` value ranging from `0` to `5` is returned, in volts.

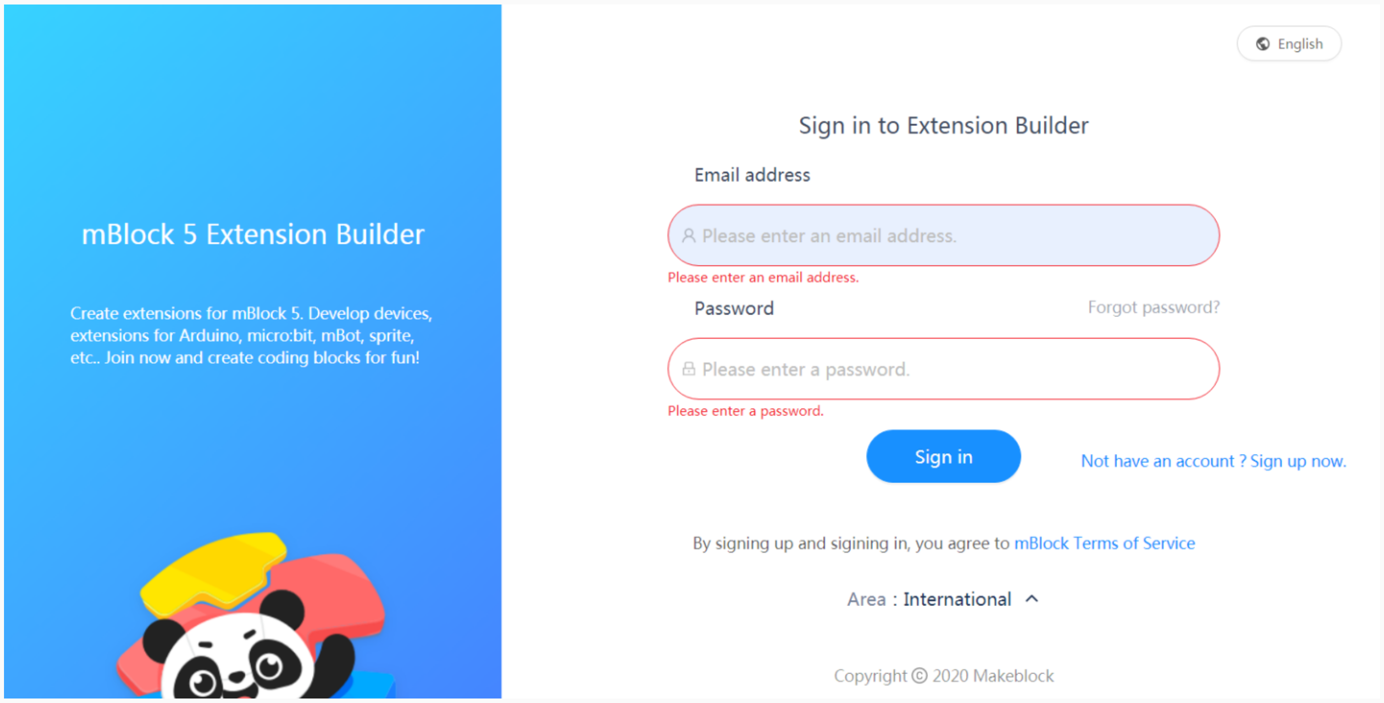
For more information, see the *Pocket Shield Operation Guide*.

Extension Builder (recommended)

With the SDK and Python APIs provided by Makeblock, you can design various block extensions to meet your needs.

Designing an extension is not as difficult as you think. Trust yourself! You can be a developer.

Open Extension Builder.


English
mBlock 5 Extension Builder
Sign in
Not have an account ? Sign up now.
Area : International
Forgot password?

Create extensions for mBlock 5. Develop devices, extensions for Arduino, micro:bit, mBot, sprite, etc.. Join now and create coding blocks for fun!

Sign in to Extension Builder

Email address

Please enter an email address.

Please enter an email address.

Password

Please enter a password.

Please enter a password.

By signing up and sigining in, you agree to mBlock Terms of Service

Copyright © 2020 Makeblock

For details about how to use Extension Builder, see the *Extension Builder Developer Documentation*.
For details about the Python APIs, see the *Python API Documentation for CyberPi*.
Courses for Extension Builder are on the way. Stay tuned!

SDK for developers

To modify the underlying firmware of CyberPi, you can view the corresponding Gitee project. Note that modifications on the firmware may cause the failure of some functions, but you can update the firmware of CyberPi on mBlock 5 to restore the functions, of course.

With the SDK for developers, you can develop CyberPi in another programming environment other than mBlock 5.

Arduino SDK

Arduino is a platform widely used in the open-source hardware field, and therefore we have designed an open Arduino SDK for CyberPi. You can click the following link to modify the firmware of CyberPi and program it based on the functions it provides by using Arduino.

Visit the Arduino SDK at https://github.com/Makeblock-official/CyberPi-Library-for-Arduino.

Currently, the firmware includes the source code of the drivers for the sensors, display, and LEDs, and available interfaces are encapsulated in the drivers.

Welcome to join us to make it more powerful! We will update this project continuously to develop more features for it.

Arduino examples

Some basic example programs are provided in the following. You can download the official Arduino IDE to update and use the firmware of CyberPi.

1. Button

```
#include "cyberpi.h"

CyberPi cyber;
void setup()
{
    Serial.begin(115200);
    cyber.begin();
}

void loop()
{
    Serial.print("a:");
    Serial.print(cyber.get_button_a());
    Serial.print(" b:");
    Serial.print(cyber.get_button_b());
    Serial.print(" menu:");
    Serial.println(cyber.get_button_menu());
    delay(500);
}
```

2. Joystick

```

```

```
#include "cyberpi.h"

CyberPi cyber;
void setup()
{
    Serial.begin(115200);
    cyber.begin();
}

void loop()
{
    Serial.print("x:");
    Serial.print(cyber.get_joystick_x());
    Serial.print(" y:");
    Serial.print(cyber.get_joystick_y());
    Serial.print(" pressed:");
    Serial.println(cyber.get_joystick_pressed());
    delay(500);
}
```

3. LED

```
#include "cyberpi.h"

CyberPi cyber;
void setup()
{
    cyber.begin();
}
float j, f, k;

void loop()
{
    for(uint8_t t = 0; t < 5; t++)
    {
        uint8_t red = 32 * (1 + sin(t / 2.0 + j / 4.0) );
        uint8_t green = 32 * (1 + sin(t / 1.0 + f / 9.0 + 2.1) );
        uint8_t blue = 32 * (1 + sin(t / 3.0 + k / 14.0 + 4.2) );
```

```
            cyber.set_rgb(t, red, green, blue);
        }
        j += random(1, 6) / 6.0;
        f += random(1, 6) / 6.0;
        k += random(1, 6) / 6.0;
        delay(10);
}
```

4. Light sensor

```
#include "cyberpi.h"

CyberPi cyber;
void setup()
{
    Serial.begin(115200);
    cyber.begin();
}

void loop()
{
    Serial.print("light:");
    Serial.println(cyber.get_light());
    delay(500);
}
```

5. Gyroscope

```
#include "cyberpi.h"

CyberPi cyber;
void setup()
{
    Serial.begin(115200);
    cyber.begin();
}

```

```
void loop()
{
    Serial.print("roll:");
    Serial.print(cyber.get_roll());
    Serial.print(" pitch:");
    Serial.println(cyber.get_pitch());
    delay(25);
}
```

6. Display

```
#include "cyberpi.h"

CyberPi cyber;
void setup()
{
    Serial.begin(115200);
    cyber.begin();
    for(int y=0;y<128;y++)
    {
        for(int x=0;x<128;x++)
        {
            int R = (128-x)*255/128;
            int G = x*255/128;
            int B = y*255/128;
            cyber.set_lcd_pixel(x,y,cyber.swap_color(cyber.color24_to_16((R<<16)+(G<<8)+B)));
        }
    }
    cyber.render_lcd();
}
void loop()
{
    cyber.set_lcd_light(false);
    delay(2000);
    cyber.set_lcd_light(true);
    delay(2000);
}
```

7. Points

```
#include "cyberpi.h"

CyberPi cyber;
#define POINTS_COUNT 200
Bitmap points[POINTS_COUNT];
float speed_x[POINTS_COUNT];
float speed_y[POINTS_COUNT];
void setup() {
    Serial.begin(112500);
    delay(1000);
    cyber.begin();
    for(int i=0;i<POINTS_COUNT;i++)
    {
        points[i].x = 64;
        points[i].y = 64;
        speed_x[i] = random(100)/20.0f-2.5f;
        speed_y[i] = random(100)/20.0f-2.5f;
        points[i].width = 1;
        points[i].height = 1;
        points[i].buffer = (uint16_t*)cyber.malloc(2);
        points[i].buffer[0] = 0xffff;
    }
}

void loop()
{
    cyber.clean_lcd();
    for(int i=0;i<POINTS_COUNT;i++)
    {
        cyber.set_bitmap(points[i].x,points[i].y,&points[i]);
        points[i].x+=speed_x[i];
        points[i].y+=speed_y[i];
        if(points[i].x<0||points[i].y<0||points[i].x>127||points[i].y>127)
        {
            points[i].x = 64;
            points[i].y = 64;
        }
    }
    cyber.render_lcd();
}
```

8. Text

```
#include "cyberpi.h"

CyberPi cyber;
uint8_t samples[128];
int idx = 0;

void setup()
{
    Serial.begin(112500);
    delay(1000);
    cyber.begin();
    int font_size = 16;
    Bitmap *bitmap1 = cyber.create_text(L"你好",0xffff,font_size);
    cyber.set_bitmap(4,4,bitmap1);
    Bitmap *bitmap2 = cyber.create_text(L"簡體",0xff00,font_size);
    cyber.set_bitmap(4,24,bitmap2);
    Bitmap *bitmap3 = cyber.create_text(L"hello",0x00ff,font_size);
    cyber.set_bitmap(4,44,bitmap3);
    Bitmap *bitmap4 = cyber.create_text(L"こんにちは",0x0ff0,font_size);
    cyber.set_bitmap(4,64,bitmap4);
    Bitmap *bitmap5 = cyber.create_text(L"여보세요",0x0f0f,font_size);
    cyber.set_bitmap(4,84,bitmap5);
    Bitmap *bitmap6 = cyber.create_text(L"Привет",0xf0f0,font_size);
    cyber.set_bitmap(4,104,bitmap6);
    cyber.render_lcd();
}
```

```

void loop()
{

}
```

9. Loudness detection

```
#include "cyberpi.h"

CyberPi cyber;
uint8_t samples[128];
int idx = 0;

void setup() {
    Serial.begin(112500);
    delay(1000);
    cyber.begin();
}

void loop()
{
    if(idx<128)
    {
        samples[idx] = cyber.get_loudness()>>5;
        idx++;
    }
    else
    {
        for(int i=0;i<128;i++)
        {
            samples[i] = samples[i+1];
        }
        samples[idx-1] = cyber.get_loudness()>>5;
    }
    cyber.clean_lcd();
    for(int i=0;i<128;i++)
    {
```

```
   if(i==0)
    cyber.set_lcd_pixel(i,127-samples[i],0xffff);
   else
   {
     int min_level = MIN(samples[i-1],samples[i]);
     int max_level = MAX(samples[i-1],samples[i]);
     for(int j=min_level;j<=max_level;j++)
     {
      cyber.set_lcd_pixel(i,127-j,0xffff);
     }
   }

  }
  cyber.render_lcd();
  delay(25);
}
```

10. Microphone data

```
#include "cyberpi.h"

CyberPi cyber;

int lo[7] = {48,50,52,53,55,57,59};
int mo[7] = {60,62,64,65,67,69,71};
int ho[7] = {72,74,76,77,79,81,83};
void mic_recv(uint8_t* samples,int len)
{
  cyber.clean_lcd();
  for(int i=0;i<len;i+=8)
  {
   int current = (int8_t)samples[i+1];
   if(current>-64&&current<64)
   {
    cyber.set_lcd_pixel(i/8,current+64,0xffff);
   }
  }
  cyber.render_lcd();
}

void setup()
{
    Serial.begin(112500);
    delay(1000);
    cyber.begin();
    cyber.on_microphone_data(mic_recv);
    for(int i=0;i<14;i++)
    {
        cyber.set_instrument(i);
        int idx = 0;
        while(idx<7)
        {
            cyber.set_pitch(0,lo[idx],100);
            delay(600);
            idx++;
        }
        idx = 0;
        while(idx<7)
        {
            cyber.set_pitch(0,mo[idx],100);
            delay(600);
            idx++;
        }
        idx = 0;
        while(idx<7)
        {
            cyber.set_pitch(0,ho[idx],100);
            delay(600);
            idx++;
        }
    }
}

void loop()
{

}
```

11. Music property

```
#include "cyberpi.h"

CyberPi cyber;

int lo[7] = {48,50,52,53,55,57,59};
int mo[7] = {60,62,64,65,67,69,71};
int ho[7] = {72,74,76,77,79,81,83};
void data_recv(uint8_t* samples,int len)
{
    cyber.clean_lcd();
    for(int i=0;i<len;i+=16)
    {
      int current = samples[i+1];
      cyber.set_lcd_pixel(i/16,current-64,0xffff);
    }
    cyber.render_lcd();
}
void setup()
{
    Serial.begin(112500);
    delay(1000);
    cyber.begin();
    cyber.on_sound_data(data_recv);
    for(int i=0;i<14;i++)
    {
        cyber.set_instrument(i);
        int idx = 0;
        while(idx<7)
        {
            cyber.set_pitch(0,lo[idx],100);
            delay(600);
            idx++;
        }
        idx = 0;
        while(idx<7)
        {
            cyber.set_pitch(0,mo[idx],100);
            delay(600);
            idx++;
        }
        idx = 0;
        while(idx<7)
        {
            cyber.set_pitch(0,ho[idx],100);
            delay(600);
            idx++;
        }
    }
}

void loop()
{
}
```

Iteration 003

Foreword

This page describes the details about iteration 003 of CyberPi. The outcome of each iteration is a major version of CyberPi.

Most of the iterations will upgrade the product incrementally while being compatible with the earlier versions, and therefore, your programs compiled for the earlier versions can be used in the latest version.

Understanding the information provided on this page may require some background information. If you have purchased the latest version of CyberPi or use an earlier version without questions about the function updates, you don't have to read this page.

Version description

The outcome of iteration 003 is the first major version of CyberPi. The following table describes the versions of its hardware, firmware, and block plugin.

Launch date	August 20, 2020
Hardware version	CyberPi PCB V1.1 Pocket Shield PCB V1.1
Firmware version	CyberPi 003 (NEW) Pocket Shield 006 (NEW)
Block plugin version	cyberpi 1.0.9 (NEW) sprite 0.01 (NEW) pocket 0.03 (NEW)

Firmware update

Visit mBlock on the web and update the firmware of CyberPi as prompted. If mBlock 5 doesn't prompt you to update the firmware, your CyberPi runs with the latest firmware.

Alternatively, connect your CyberPi to the Internet through Wi-Fi, enter CyberOS, and choose **Settings** > **Update**. The system checks the current version and updates the firmware if it is not the latest version.

Iteration summary

1. Rectification and improvement

Rectified the problems found in firmware version 002 to stabilize the blocks and functions of CyberPi, and made some improvement, as described in the following:

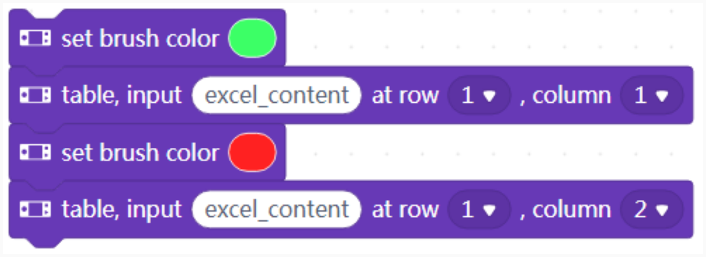
- Removed the tight coupling between the brush color and text settings. In the earlier version, the printing of texts changed immediately with the change of the brush color; in this version, after setting the brush color, it is applied to a text only after you program to print the text again.
- Modified the playing of sounds in **Live** mode to be consistent with that in **Upload** mode, enabling the playing in **Live** mode to be interrupted by another thread.
- Fine-tuned the mechanism of beats for the piano playing blocks, making the implementation logic consistent with that of the Music extension for sprites on mBlock 5.
- Optimized the playing mechanism of the buzzer, eliminating the playing interruption caused by frequency change.
- Rectified the errors that occurred when a bar chart was rotated on the screen.
- Rectified the problems that bar charts and line charts supported only integers.
- Improved the display of line charts, stopping it from starting at the zero-point.
- Improved the format of tables, replacing the graduated-color style with the black-and-white style
- Modified the execution logic the event headers in **Live** mode to be consistent with that in **Upload** mode on mBlock 5, that is, one event header corresponds to only one thread instead of multiple ones when it is triggered multiple times
- Updated the default programs provided on CyberPi. The original programs included some politically sensitive elements and might cause copyright issues. In this version, blocks and Python code of all the default programs can be provided to users.
- Rectified the problem that CyberPi failed to automatically connect to the Internet through the configured Wi-Fi name and password. In this version, after you set the Wi-Fi name and password for your CyberPi on mBlock 5, CyberPi attempts to connect to the Internet every time it is started.
- Rectified the problem that CyberPi prompted you to update the firmware even when it ran with the latest firmware.
- Modified the execution logic of the block restart CyberPi in **Live** mode to be consistent with that in **Upload** mode. In this version, after the block restart CyberPi is executed in **Live** mode,

all the blocks under the block when CyberPi starts up are executed again.

2. New features

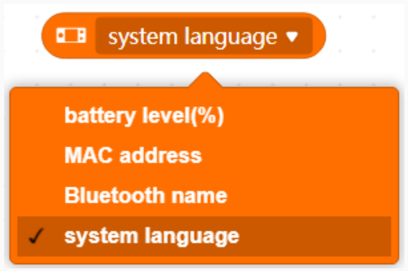
Added some new features to enhance the functions of CyberPi, as described in the following:

- Setting the color of texts in a table by setting the brush color



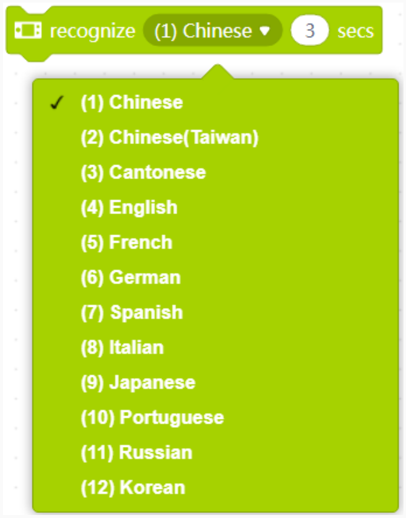
set brush color 🟩
table, input excel_content at row 1 ▾ , column 1 ▾
set brush color 🟥
table, input excel_content at row 1 ▾ , column 2 ▾

- Displaying texts in multiple fonts in any positions
- Displaying to-be-printed texts in multiple fonts
- Prompting you to set the system language when CyberPi is started for the first time
- Providing more system languages, including Simplified Chinese (简体中文), English, Traditional Chinese (繁體中文), Japanese (日本語), Korean (한국어), Spanish (Español), Italian (taliano), French (Français), and German (Deutsch)
- Providing the interface for obtaining the system language



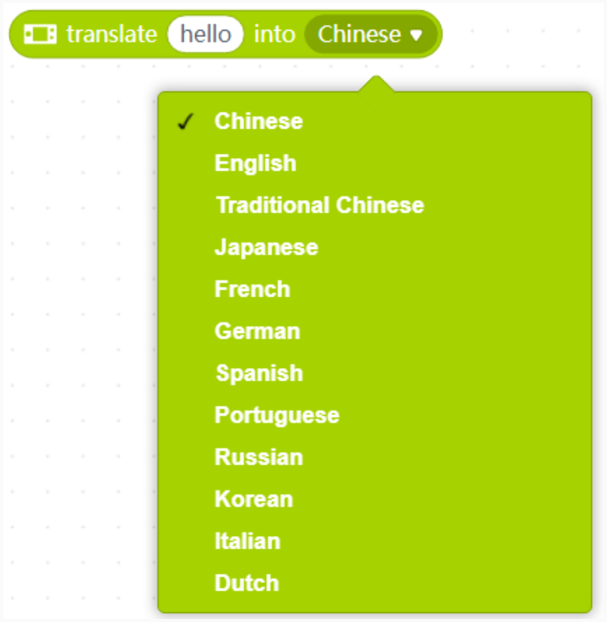
system language ▾
battery level(%)
MAC address
Bluetooth name
✓ system language

- Recognizing speeches of common countries and regions

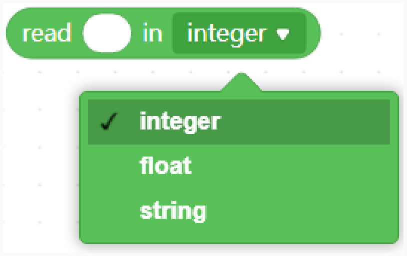


recognize (1) Chinese ▾ 3 secs
✓ (1) Chinese
(2) Chinese(Taiwan)
(3) Cantonese
(4) English
(5) French
(6) German
(7) Spanish
(8) Italian
(9) Japanese
(10) Portuguese
(11) Russian
(12) Korean

- Translating texts into more languages



- Providing a new charging mechanism, enabling faster charging. In this version, when Pocket Shield and CyberPi are connected to a PC or charger and you turn Pocket Shield off, CyberPi enters the charging mode, displaying the charging state in real time.
- Providing the block for numeric type conversion in the **Operators** category



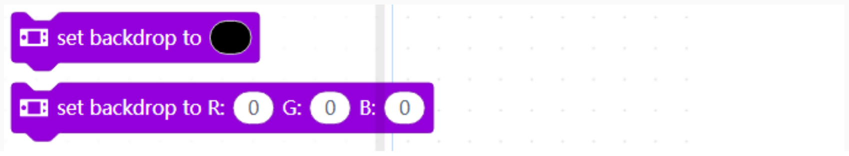
- Supporting lists in **Upload** mode
- Inserting code through blocks

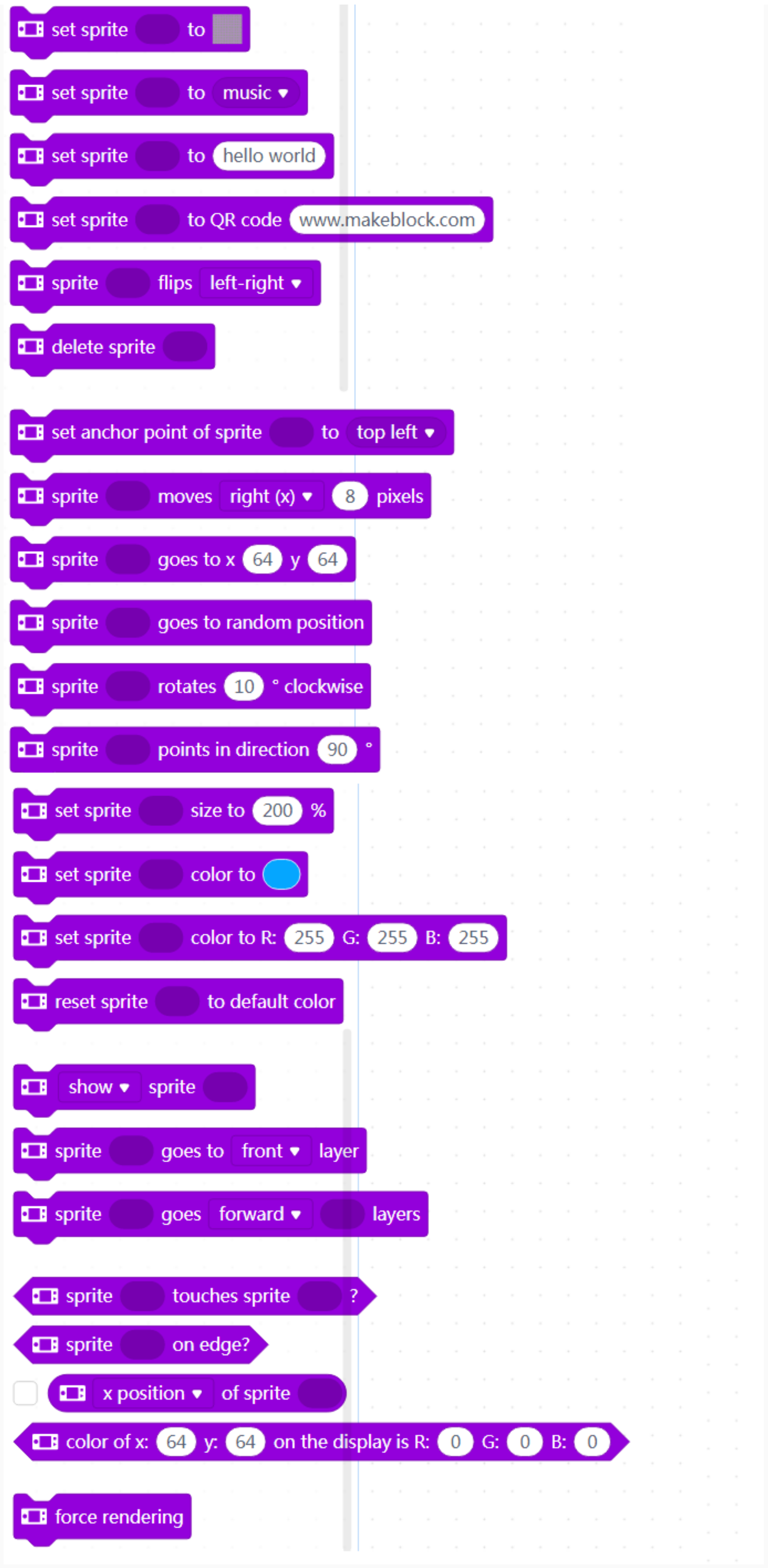
3. Unstable features

We have also provided some features that are not fully stabilized for teachers and students to create some basic design.

Efforts will be made to ensure the compatibility of later official versions with projects created in the earlier versions.

- Providing the **Sprites** extension and APIs (supporting only the **Upload** mode), enabling you to create and run apps and games on CyberPi




set sprite ( ) to ( )
set sprite ( ) to music ▾
set sprite ( ) to hello world
set sprite ( ) to QR code www.makeblock.com
sprite ( ) flips left-right ▾
delete sprite ( )

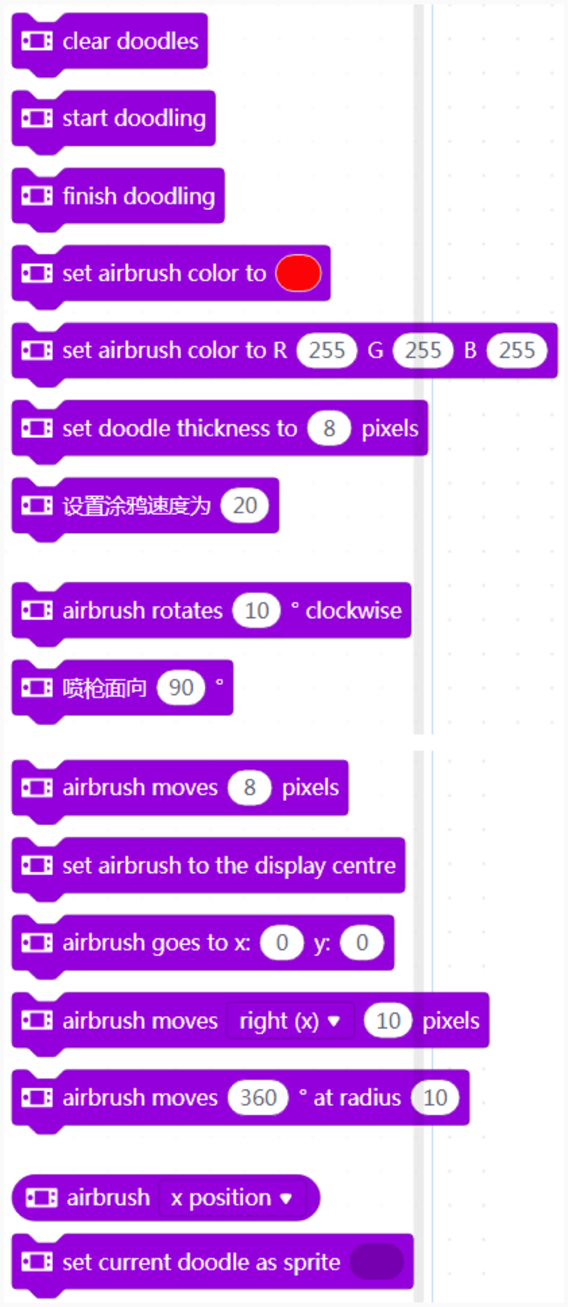
set anchor point of sprite ( ) to top left ▾
sprite ( ) moves right (x) ▾ 8 pixels
sprite ( ) goes to x 64 y 64
sprite ( ) goes to random position
sprite ( ) rotates 10 ° clockwise
sprite ( ) points in direction 90 °
set sprite ( ) size to 200 %
set sprite ( ) color to ( )
set sprite ( ) color to R: 255 G: 255 B: 255
reset sprite ( ) to default color

show ▾ sprite ( )
sprite ( ) goes to front ▾ layer
sprite ( ) goes forward ▾ ( ) layers

sprite ( ) touches sprite ( ) ?
sprite ( ) on edge?
x position ▾ of sprite ( )
color of x: 64 y: 64 on the display is R: 0 G: 0 B: 0

force rendering

- Providing the **Doodle** extension and APIs (supporting only the **Upload** mode), enabling you to draw sketches similar to Python Turtle drawings on CyberPi. In addition, the sketches you draw can be set as sprites and thus used in apps or games you create.


clear doodles
start doodling
finish doodling
set airbrush color to
set airbrush color to R 255 G 255 B 255
set doodle thickness to 8 pixels
设置涂鸦速度为 20
airbrush rotates 10 ° clockwise
喷枪面向 90 °
airbrush moves 8 pixels
set airbrush to the display centre
airbrush goes to x: 0 y: 0
airbrush moves right (x) ▾ 10 pixels
airbrush moves 360 ° at radius 10
airbrush x position ▾
set current doodle as sprite

- Cancelling the Bluetooth connection function. Currently, mBlock 5 on the web and mBlock 5 PC client don't support Bluetooth connection of CyberPi due to the implementation mechanism of mLink 2. We may enable this function in the future. To enable the wireless connection between CyberPi and mBlock 5, you can purchase Makeblock Bluetooth Dongle or use mBlock 5 mobile app.

Change History

Date	Modifications
2020/07/20	Initial draft
2020/08/26	Updated the UI-related figures according to the latest version (003 firmware)
2020/10/22	Added the **Accessories** group and added the page "Wireless Adapter"