1. Overview

Cobit is a mini robot car designed for beginners, hobbyists, engineers, STEM educators, pro electronic enthusiast kids and adults.

It is driven by two stepper motors and can run at precise distances. It integrates an ultrasonic module, two photosensitive sensors, two 5mm white LED lights, four RGB LED light, a buzzer, an infrared receiving sensor, an infrared line tracking sensor, and it can be used with a Bluetooth module and an ESP-01 wifi module.

The Cobot robot is compatible with Arduino, Microbit, ESP wifi, so you can use C, C++, JavaScript, Python, Makecode block language to program it.

There are Lego mounting holes and IO ports on the car body, you can expand its functionality by adding other sensors or actuator modules.

Support

☛Quality Issues

☛Instructions for Use

☛Technical Exchange

☛Optimization Solution

☛Creativity Exchange

✉ : cokoino@outlook.com

In order to better serve you, please clearly state the SKU of the product when you ask a question, and describe your problem in detail as much as possible. We will reply to your question within 12 hours

Colophon

Date: 2021-08-03

Version: V1.0

Web: http://cokoino.com

Legal Disclaimer Notice

Cobit has benn applied for related patents. For any individual or organization imitating or pirating related technologies, we have the right to pursue related legal liabilities.

The codes and circuits involved in this product are published on GitHub: https://github.com/cokoino/CKK0008, you can use them in part or all of your own derivative works, as long as you also use the same license. The LK COKOINO brand and logo copyright belong to the Cokoino creative team and cannot be used for any commercial purposes without formal permission.

Table of Content

2. Tech specs

Power supply: two 18650 batteries (not included in the kit, you need to prepare by yourself)

Speed rating:

0 level: 0 revolutions per minute.

1 level: 15 revolutions per minute.

2 level: 30 revolutions per minute.

3 level: 60 revolutions per minute.

4 level: 120 revolutions per minute.

Wheel diameter: 65mm.

Walking precision: 0.51mm.

Precision of rotation angle: 0.9°.

Drive power: 110mN.m REF. per stepper motor.

Buzzer can be set frequency range: 20-- 1K Hz, volume: 0-5 level.

Maximum measurement distance of ultrasonic module is 255CM with an accuracy of +/-1CM.

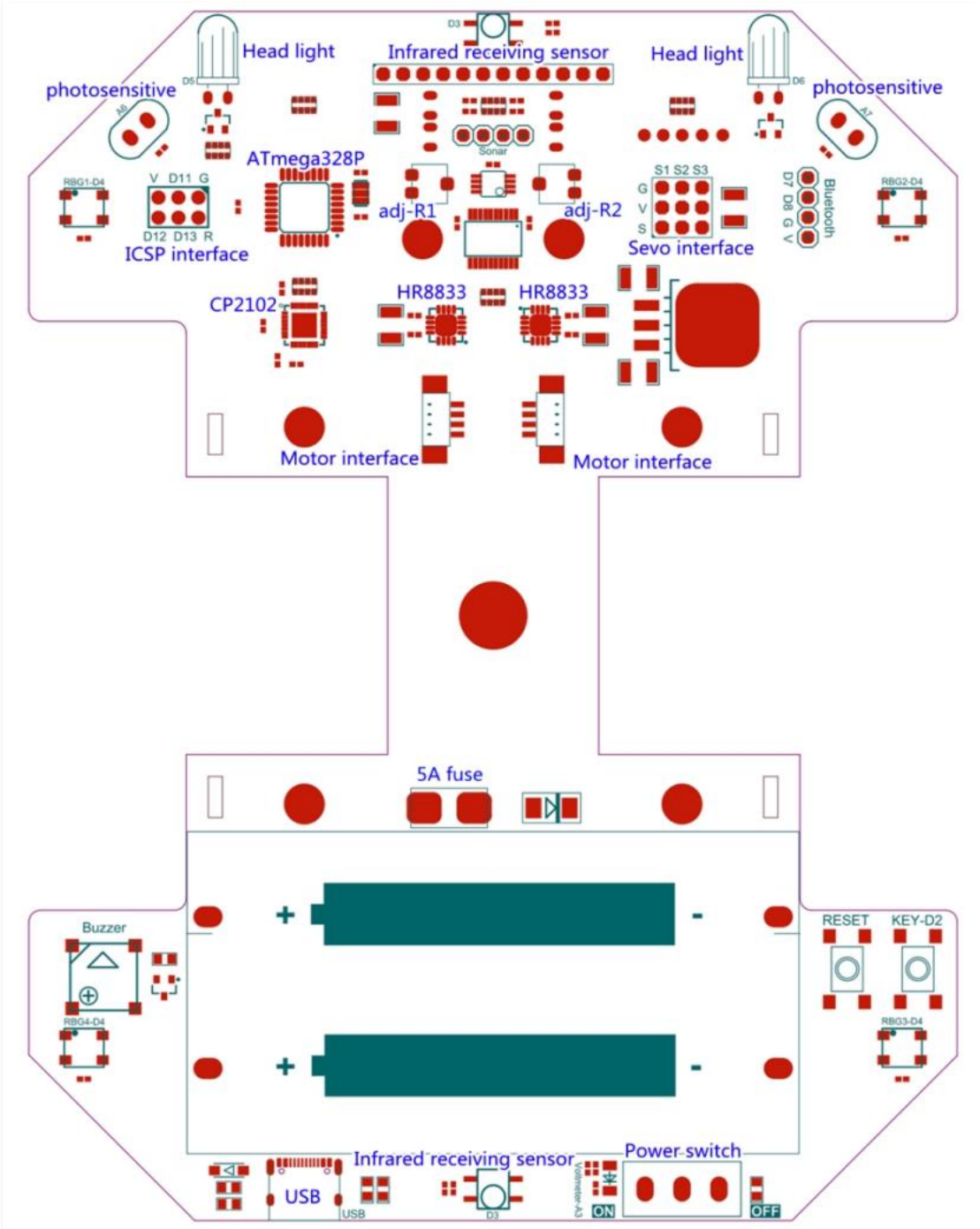
Programming: Arduino, Micro:bit, ESP-01

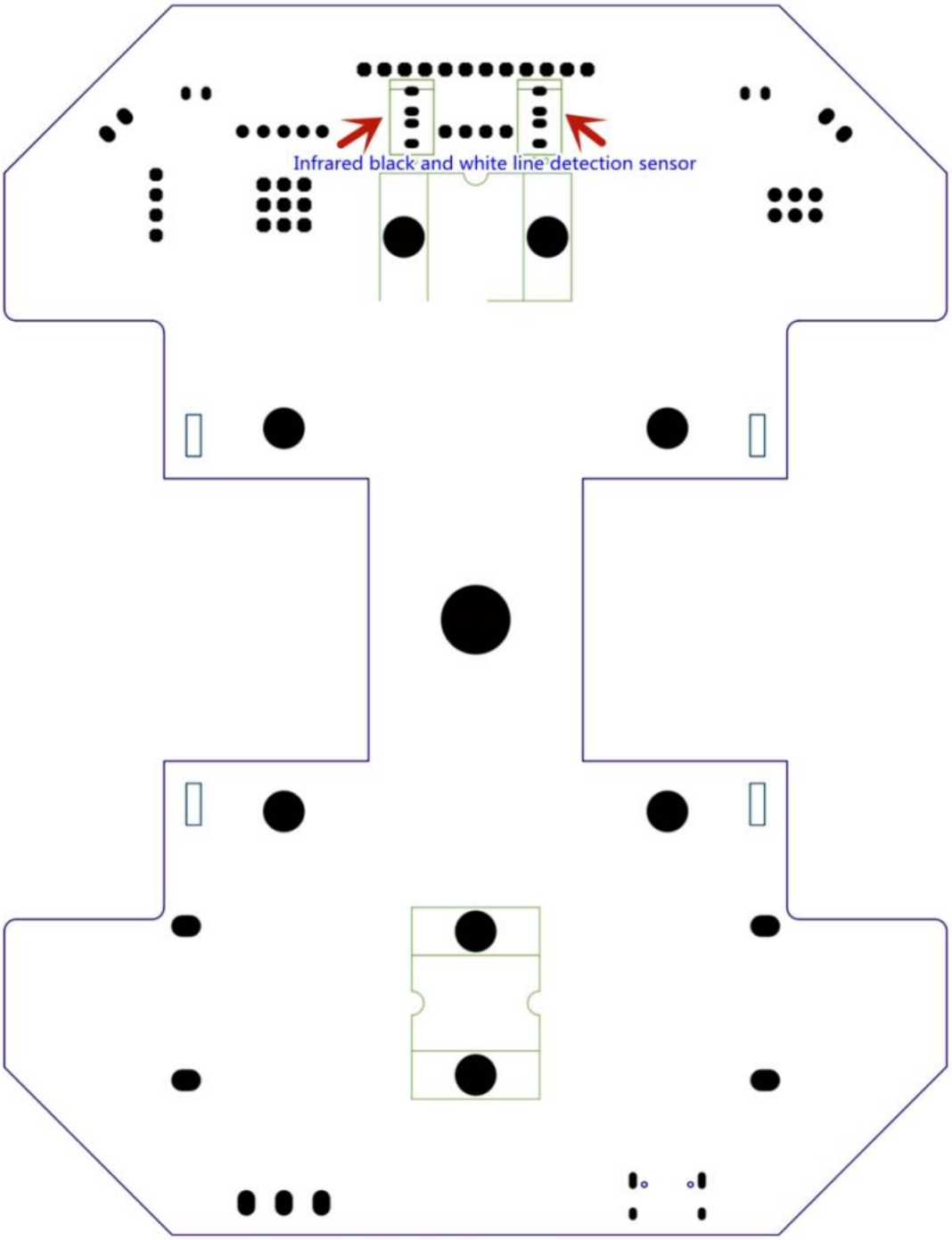
Distance between two wheels: 96mm

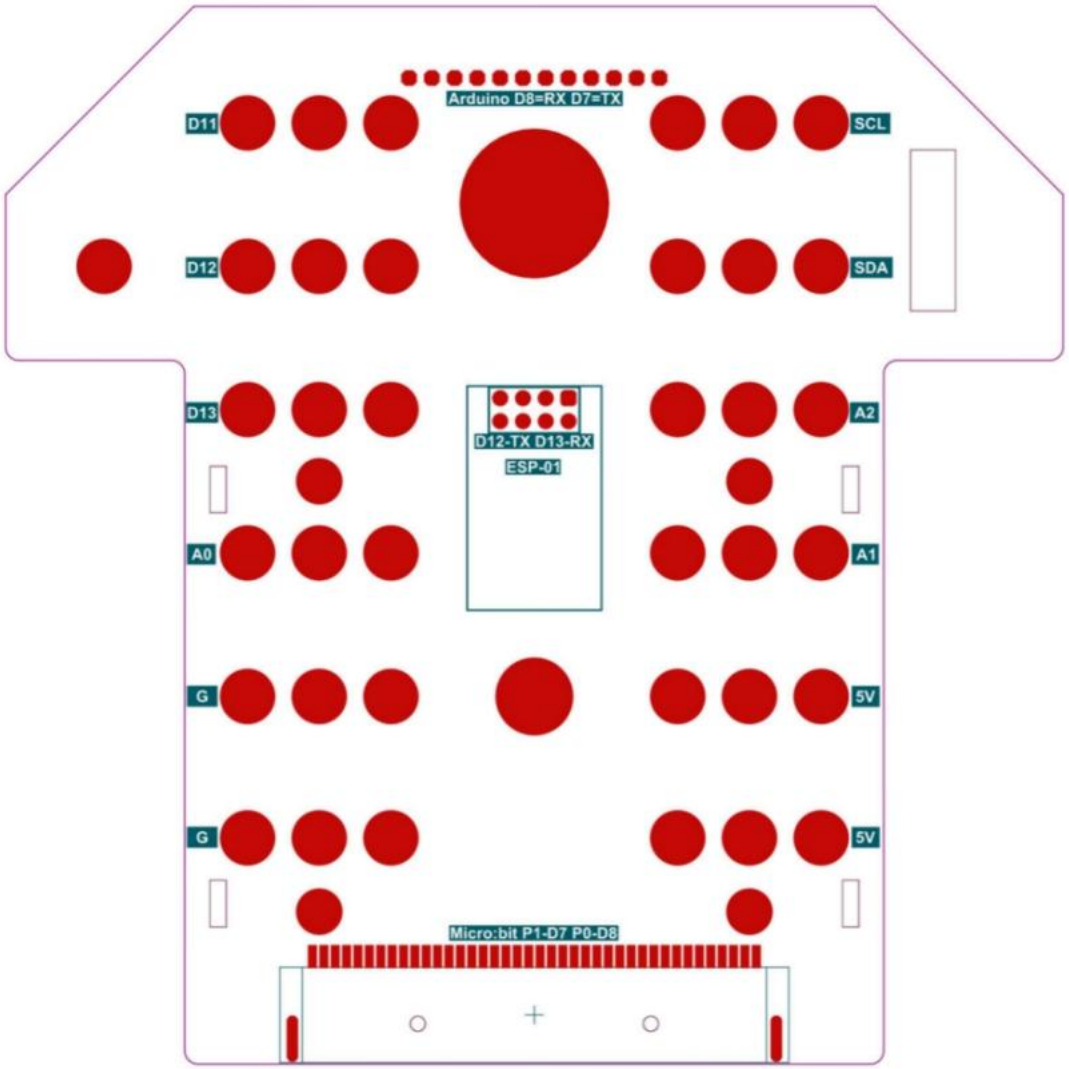
Length and width: 154*118mm

3. Hareware introduction

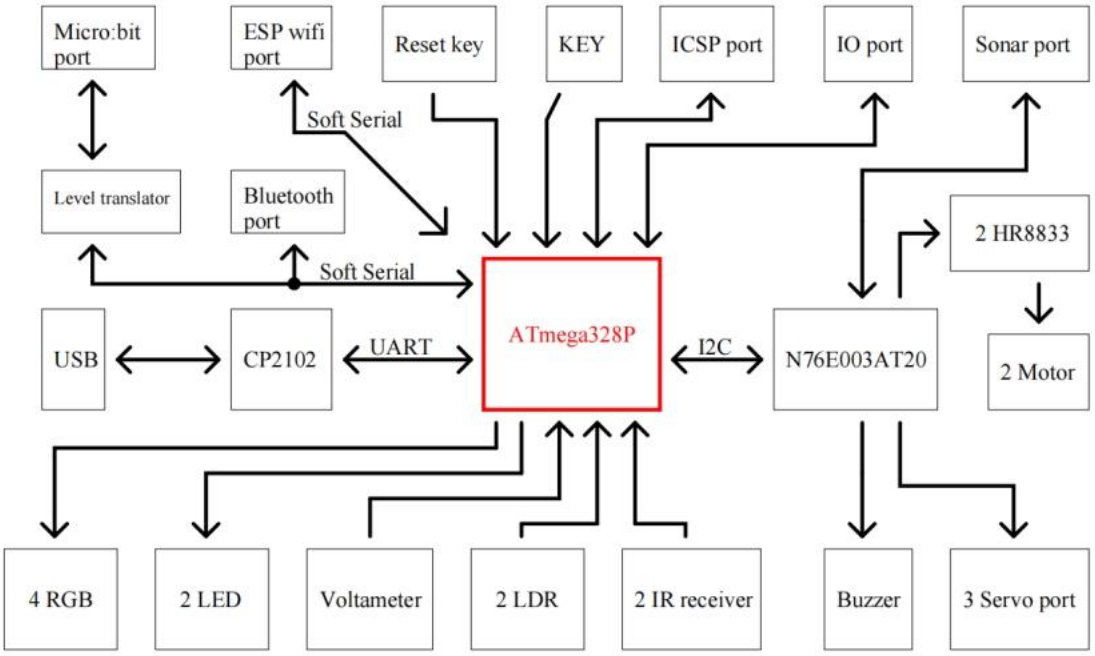
3.1 Overview




Infrared black and white line detection sensor


Arduino D8=RX D7=TX
D11
SCL
D12
SDA
D13
A2
D12-TX D13-RX
ESP-01
A0
A1
G
5V
G
5V
Micro:bit P1-D7 P0-D8

3.2 Hardware block diagram



3.3 Hardware Description

ATmega328P： Master Control Chip

The core controller can update the code through the USB port.

Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz
Bootloader	Arduino UNO R3

N76E003AT20： Auxiliary chip

Its function is to burn specific code, communicate with ATmega328P through I2C protocol, drive DRV8833RTYR, Servo, ultrasonic module and buzzer.

CP2102： USB to serial chip

Convert USB signal to TTL Serial level signal.

Max Baud Rate	921600
USB protocol	USB Specification 2.0 compliant; full-speed (12 Mbps)
Driver Support	Windows 10/8/7/Vista/XP/Server 2003/2000 Windows CE

HR8833：Motor Driver Chip

The double H-bridge motor driver can be used to drive 2 DC motors or 1 4-wire 2-phase stepper motor. It is controlled by the internal program of N76E003AT20. The I2C port of ATmega328P can send related commands to N76E003AT20 to control it.

Power Voltage Range: 2.7 V – 15 V	
Output Current	1.5-A RMS per H-Bridge
MOSFET On-Resistance:	HS + LS 400 mΩ

Motor port：

To connect 4-wire 2-phase stepper motor, one on each side, controlled by the HR8833 chip, please see the description for specific parameters as above.

Infrared Receiving Sensor：

Center frequency: 38K Hz

Receiving distance: 5-10 meters

NEC infrared communication protocol: default

Signal pin: connected to the D3 pin of ATmega328P.

Passive Buzzer：

It can produce sound up to 1K Hz , which is controlled by the internal program of N76E003AT20. The I2C port of ATmega328P can send related commands to N76E003AT20 to control it.

Infrared Line Tracking Sensor：

The Infrared Line Tracking Sensor is located under the robot and consists of two sensors Sensor1 and Sensor2. Each sensor is composed of an infrared transmitter

and an infrared receiver. Because it is often used to control the robot to walk along the line, it is also called line-following sensor. The black lines are sensed by the principle of infrared reflection. The signal pins are connected to the D9 and D10 pins of the ATmega328P. The infrared transmitter continuously emits infrared light to the ground while the Cobit is driving:

If the infrared light is reflected (for example, it runs on a white or other light-colored plane), the receiver receives an infrared signal, the signal pin outputs low power Level. If the infrared light is absorbed or cannot be reflected (for example, it runs on a black or other dark surface), the receiver cannot receive the infrared signal, the signal pin outputs high power Level. The sensitivity of the sensor can be adjusted through adj-R1 and adj-R2. It has been adjusted at the factory, and it is not recommended to adjust it again if it is not necessary.

Photoresistor Sensor：

Its function is to read the intensity of light, one on the left and right of the car body, which can output analog values, and the signal pins are respectively connected to the A6 and A7 pins of ATmega328P. The stronger the detected light, the smaller the resistance of the photoresistor, and the greater the voltage output by the internal circuit.

RGB LED：

WS2812 RGB light can emit controllable red, green and blue base colors. There are 4 lights in total. The serial numbers are divided into RGB1-RGB4. Various colors of light can be formed by controlling the intensity of the three primary colors. The signal input pin is connected to the D4 pin of ATmega328P.

Head light：

There are one 5mm white LED lights on the left and right sides of the car body, which are controlled by the D5 and D6 pins of the ATmega328P. The PWM signals of D5 and D6 pins can make the LED emit white light of different intensities.

Type C USB Port：

It is the USB communication port that sends the code and data from the PC to the cobit, it can also be used to update the program of ATmega328P.

Power Switch：

To control the power on and off of the battery box, the battery box circuit has a 5A fuse and an anti-connection diode for safety.

Voltmeter：

It is used to detect the voltage of the battery box and output the analog value. The signal pin is connected to the A3 pin of ATmega328P. Because the voltage obtained by A3 pin of ATmega328P is the battery voltage divided by 1/10, the value obtained by A3 must be multiplied by 10 to be the real battery voltage.

KEY-D2 Button：

The signal pin is connected to the D2 pin of ATmega328P. You can press the button to output low level, otherwise output high level.

RESET Button：

It is signal pin is connected to the reset pin of ATmega328P, which triggers ATmega328P to reset when the button is pressed.

Port of Ultrasonic Sensor ：

V=5V, G=GND, it is used to connect the external HC-SR04 ultrasonic module,and controlled by the internal program of N76E003AT20, the ATmega328P I2C port sends related commands to N76E003AT20 to read HC-SR04 data. The maximum test distance of the HC-SR04 ultrasonic module is 255CM, and the accuracy is +/-1CM.

Port of Servo：

V=5V, G=GND, S=control signal of servo, it is used to connect 3 channels of servos, controlled by the internal program of N76E003AT20, ATmega328P I2C port can send related commands to N76E003AT20 to control the external servos.

Port of Bluetooth：

To connect the external Bluetooth module, V=5V, G=GND, the TX pin is directly connected to the D8 of ATmega328P and the RX pin is directly connected to the D7 pin of ATmega328P.

ICSP Port：

V=5V, G=GND, it is mainly used to burn the bootloader of ATmega328P.

Port for ESP-01 wifi Module：

Used to connect ESP-01 wifi module, the TX signal pin of it is connected to D12 of ATmega328P and the RX signal pin of it is connected to D13 of ATmega328P.

Port for Micro:bit：

Used to connect the external Micro:bit board, the P0 pin of it it is connected to the D8 pin of the ATmega328P and the P1 pin of it is connected to the D7 pin of the ATmega328P through a 3V-5V level conversion circuit.

Digital Port：D11, D12, D13

These are the digital IO port of ATmega328P, among which D11 can output PWM signal.

Analog Port：A0, A1, A2

It is the analog port of ATmega328P, which can read analog values and can also be used as a digital port.

Power Output Port：G, 5V

GND = G, power supply = 5V, maximum output power is 10W, which can be used to power peripherals.

I2C Port：SCL, SDA

It is the I2C communication interface of ATmega328P, SDA=A4, SCL=A5, it can communicate with other I2C devices.

3.4 Further information

I2C Protocol：

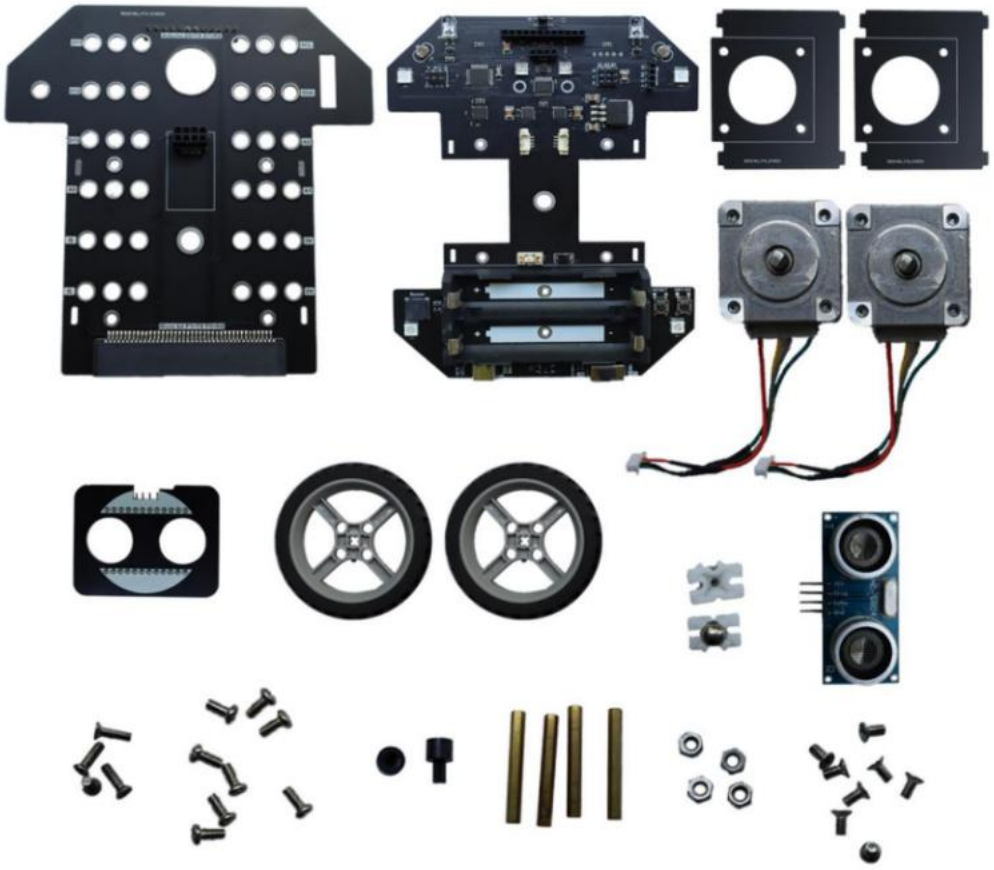
https://www.nxp.com/docs/en/user-guide/UM10204.pdf

NEC Infrared Transmission Protocol：

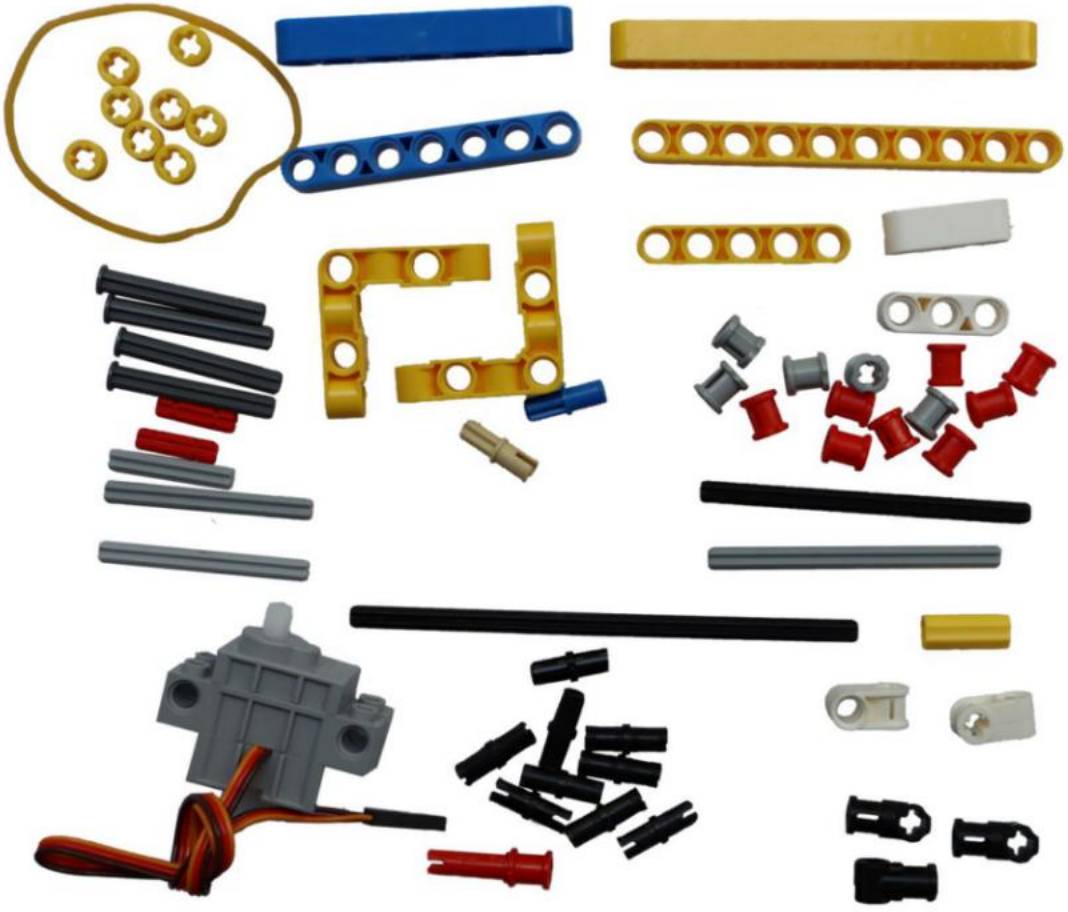
https://techdocs.altium.com/display/FPGA/NEC+Infrared+Transmission+Protocol

4. List(what's in the package)

Basic Parts



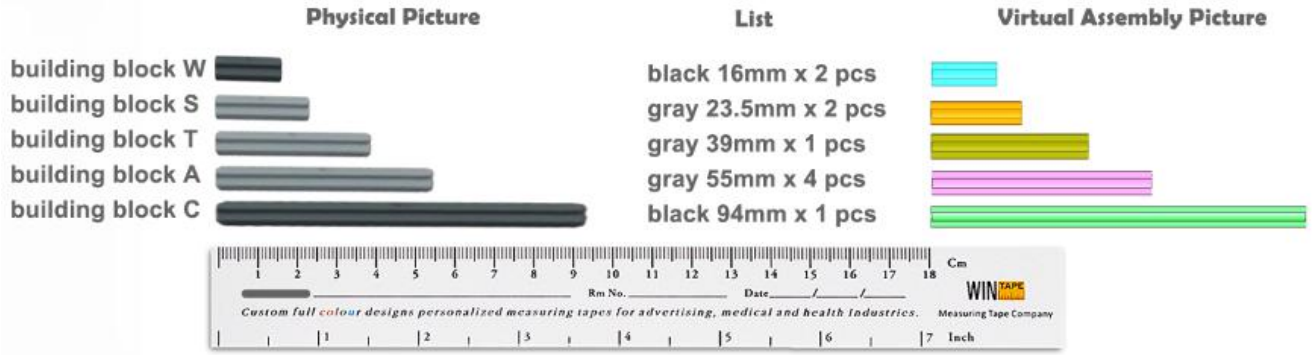
Lego Parts

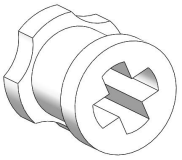
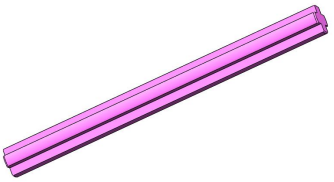


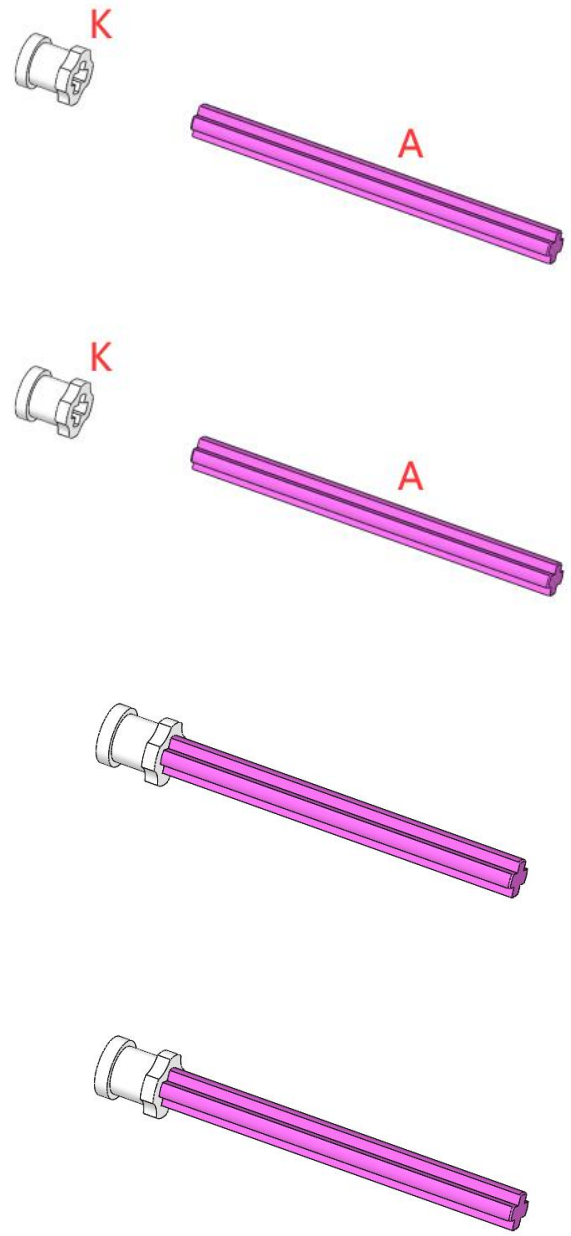
5. Assemble the Cobit Robot

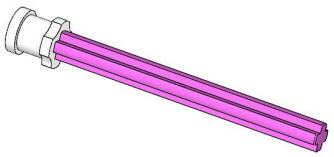
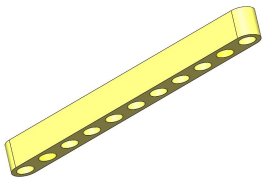
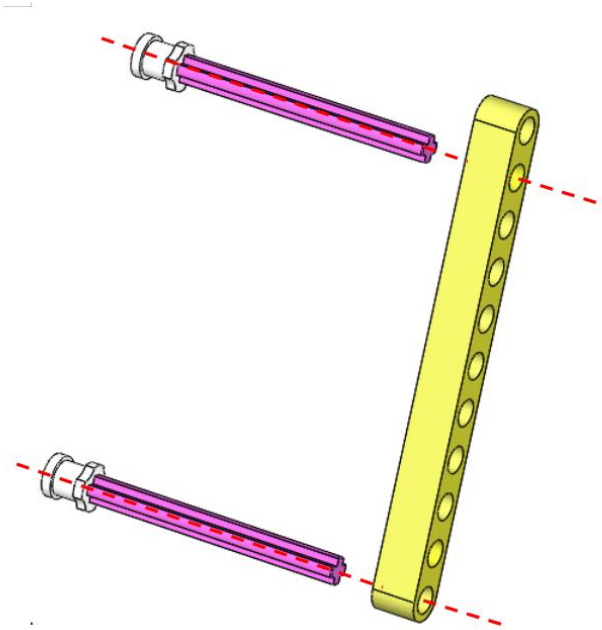
Note：

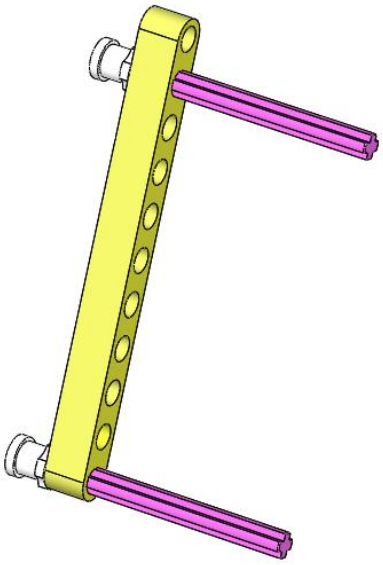
Before assembling, please take out the paper ruler from the product packaging box to measure and find the building block of the size you need.
When assembling, you may find blocks of different colors from the actual product. That is to make it easier for you to see and understand the assembly steps. The color of the block is subject to the actual product you received. The following is a list:

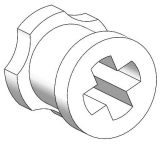
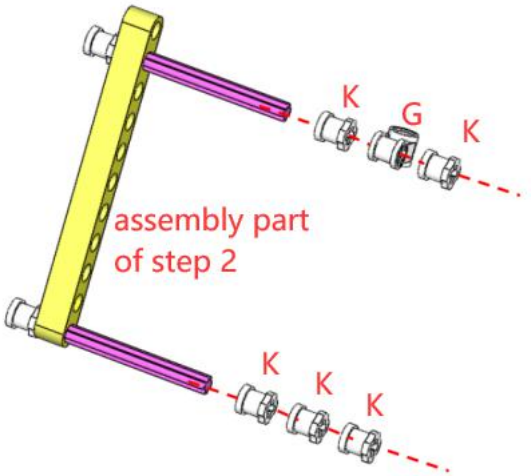


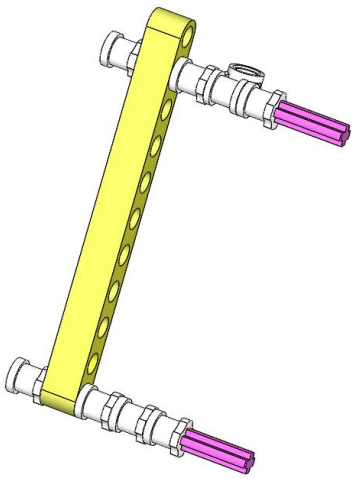
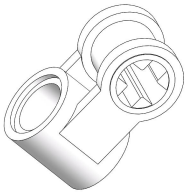
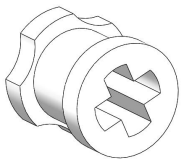
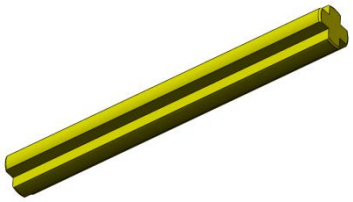
Step	1	Assemble building blocks		Tools needed	
Parts List	Part Name	Quantity	Unit	Picture	
	building block K	2	PCS		
	building block A（55mm)	2	PCS		
	Assembly Details			Assembly Demo	

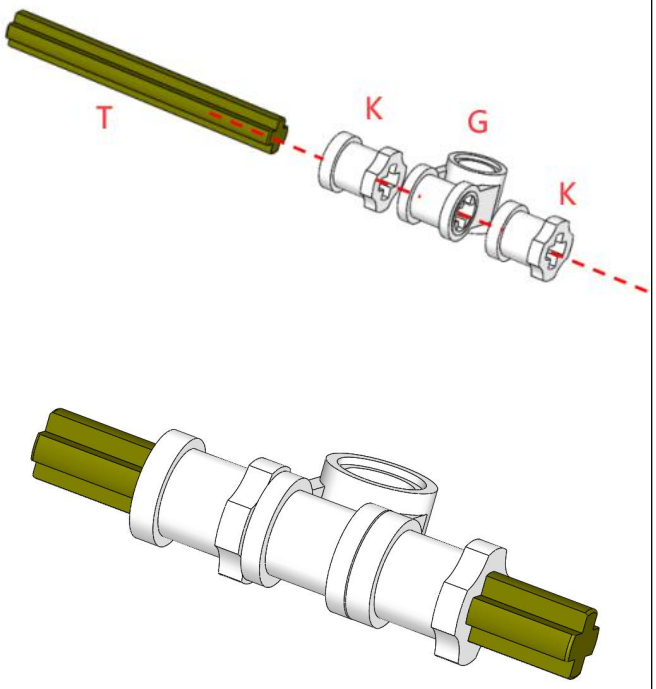
A	Take building block K and install it on building block A;			
Step	2	Assemble building blocks	Tools needed	
Part List	Part Name	Quantity	Unit	Picture

	assembling part of step 1	2	PCS	
	building block Y	1	PCS	
	Assembly Details			Assembly Demo
A	Connect the assembly part of step 1 with building block Y, as shown in the picture on the right;			

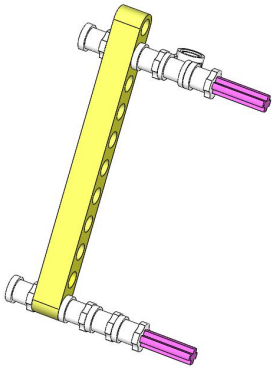
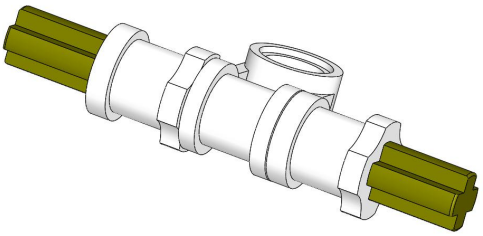
		

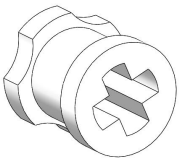
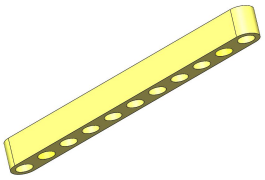
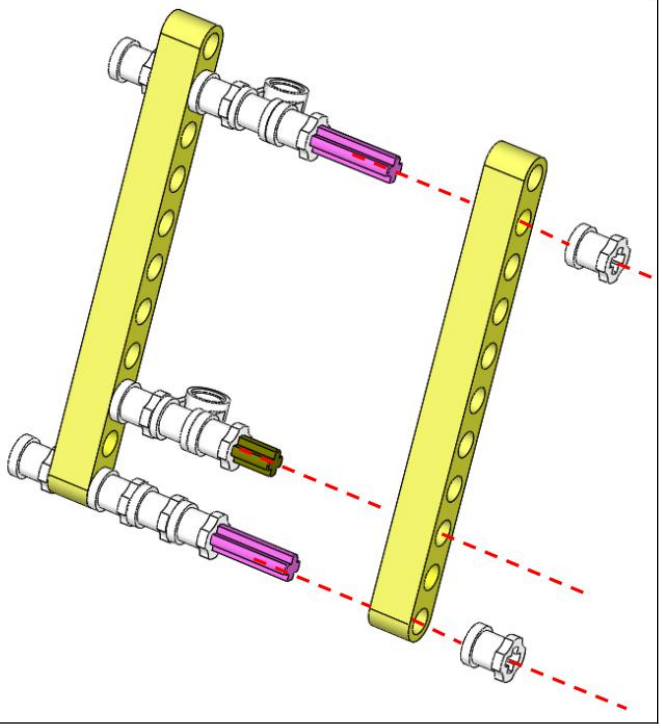
Step	3	Assemble building blocks		Tools needed	
Part List	Part Name	Quantity	Unit	Picture	
	building block K	5	PCS		
	building block G	1	PCS		
	Assembly Details			Assembly Demo	
A	Connect the assembly part of step 2 with building block K and building block G, as shown on the right；				

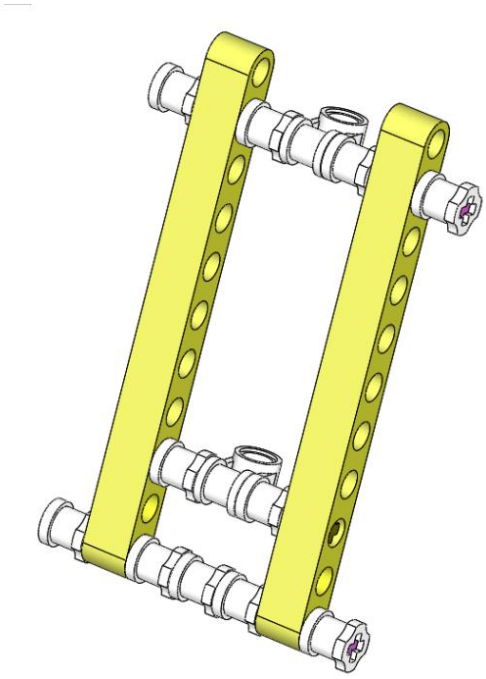
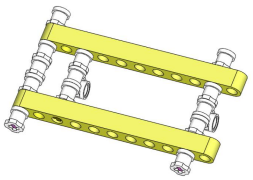
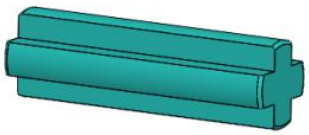
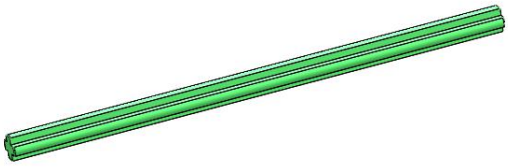
					
Step	4	Assemble building blocks		Tools needed	
Parts List	Part Name	Quantity	Unit	Picture	
	building block G	1	PCS		
	building block K	2	PCS		
	building block T (39mm)	1	PCS		
	Assembly Details			Assembly Demo	

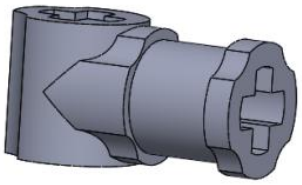
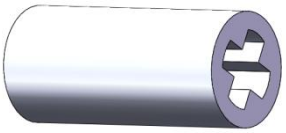
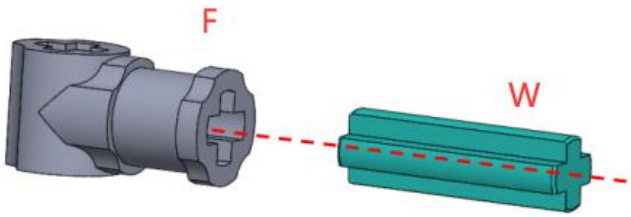
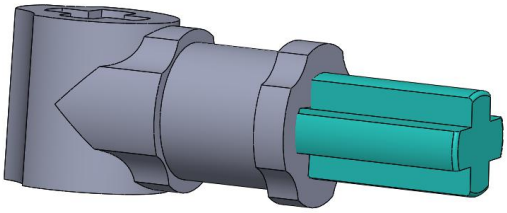
A	Install the building block K and the building block G on the building block T;	

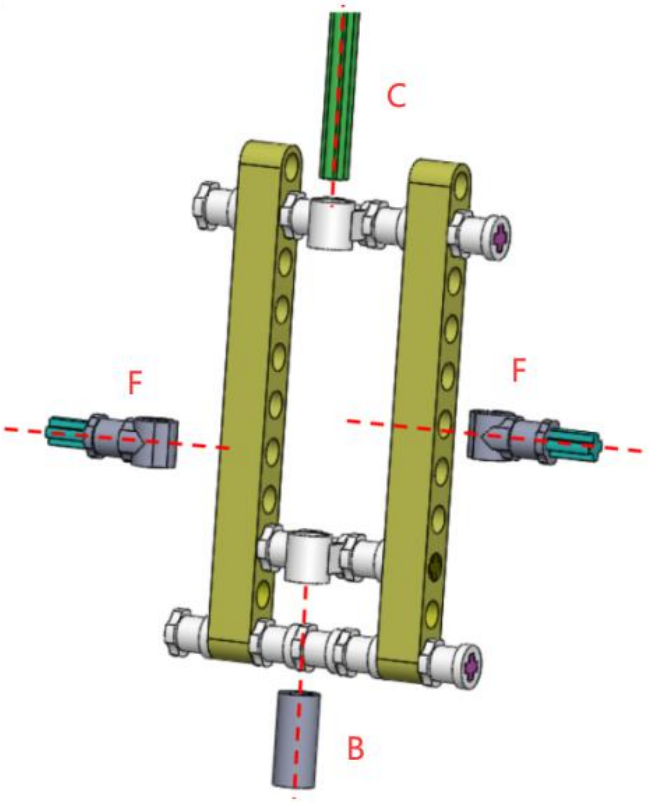
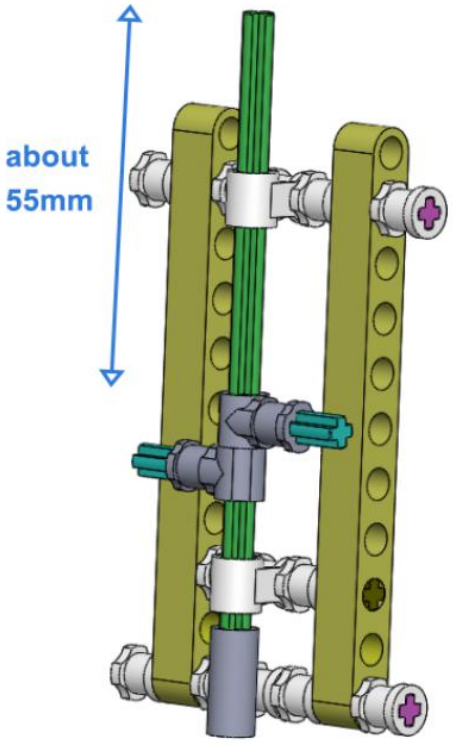
Step	5	Assemble building blocks	Tools needed	

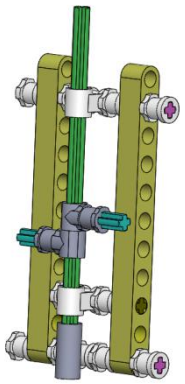
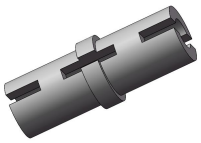
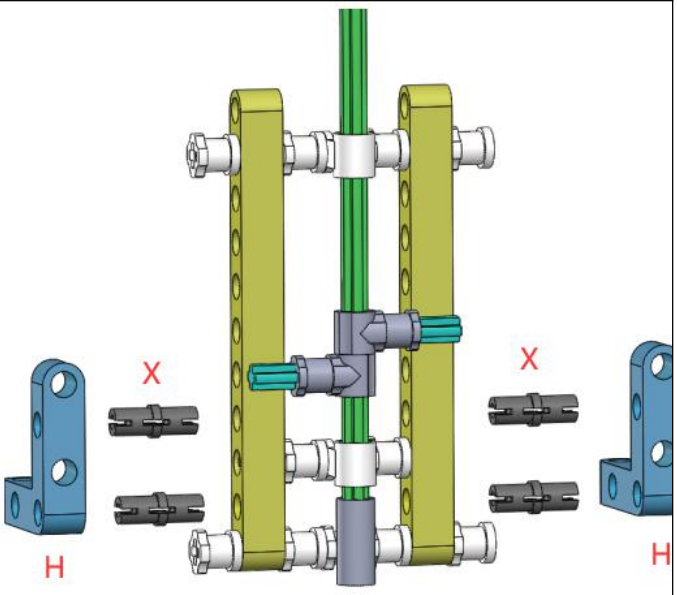
Part List	Part Name	Quantity	Unit	Picture
	the assembly part of step 3	1	PCS	
	the assembly part of step 4	1	PCS	

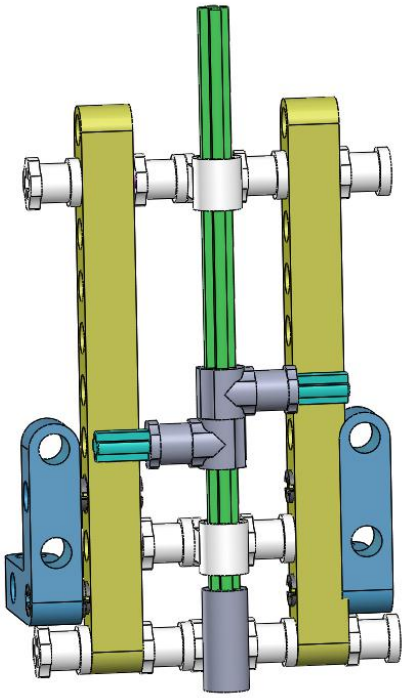
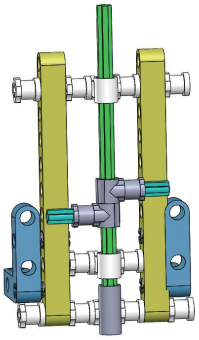
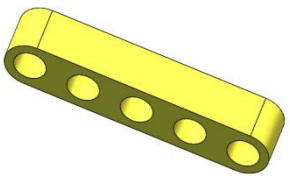
	building block K	2	PCS	
	building block Y	2	PCS	
	Assembly Details			Assembly Demo
A	Use building block Y and building block K to install the assembly part of step 4 on the assembly part of step 3; Note: The direction of building block G is as shown in the figure			

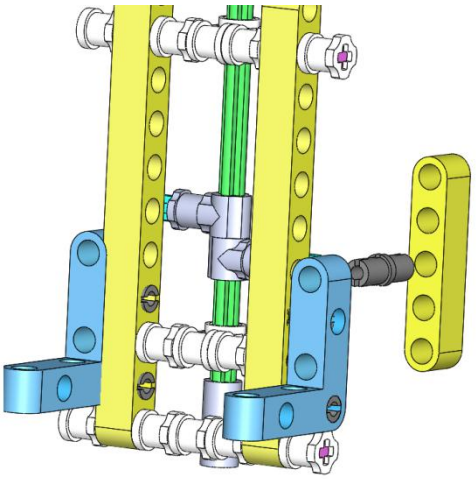
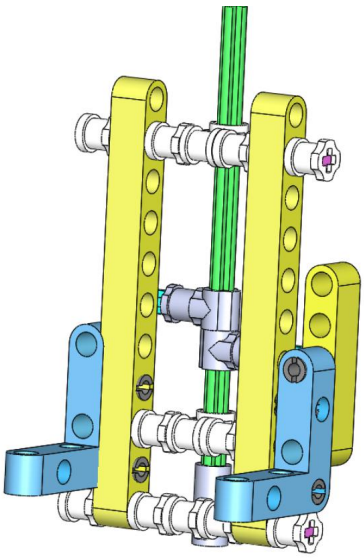
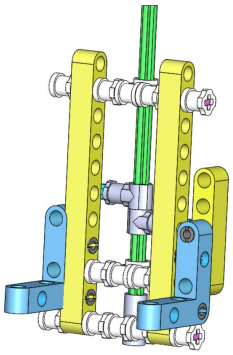
					
Step	6	Assemble building blocks		Tools needed	
Parts List	Part Name	Quantity	Unit	Picture	
	the assembly part of step 5	1	PCS		
	building block W（16mm）	2	PCS		
	building block C（94mm）	1	PCS		

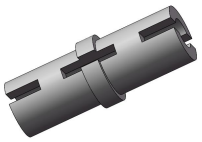
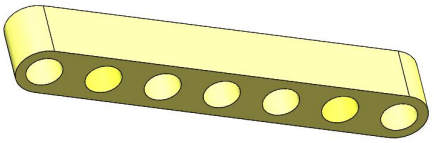
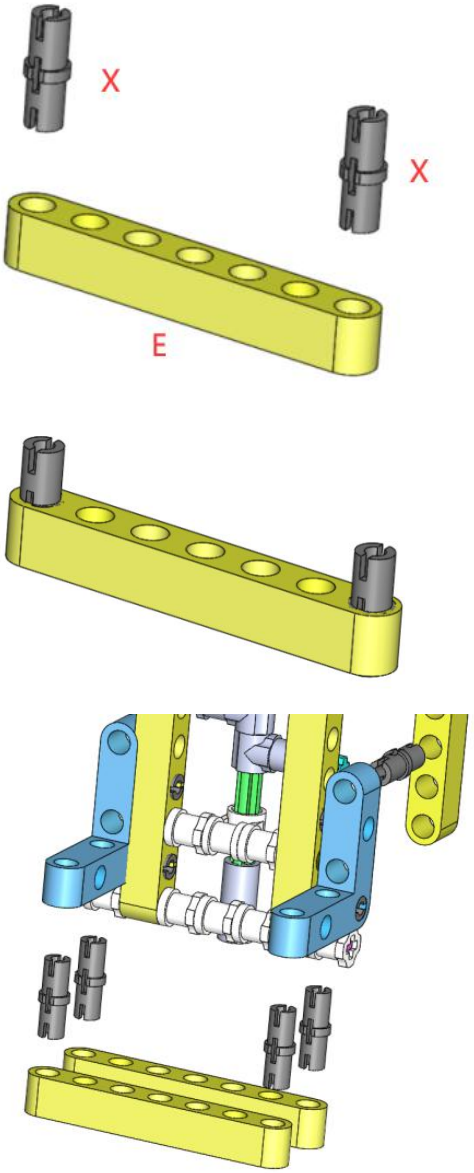
	building block F	2	PCS	
	building block B	1	PCS	
	Assembly Details			Assembly Demo
A	First assemble building block W and building block F; Then install them together with building block C and building block B on the assembly part of step 5; Note: Building block F is installed about 55mm away from the top of building block C;			 

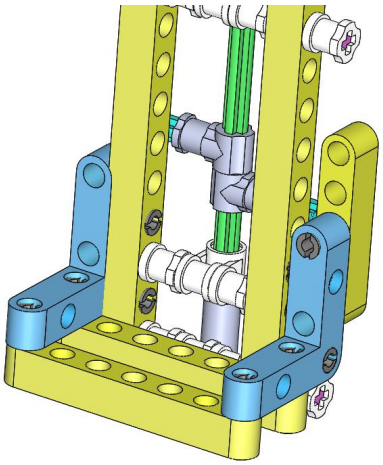
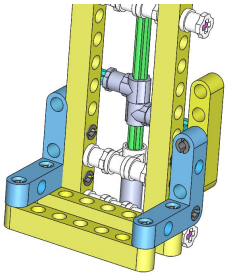
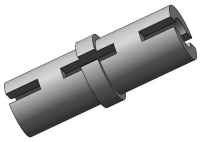
				 
Step	7	Assemble building blocks		Tools needed	
Parts	Part Name	Quant	Unit	Picture	

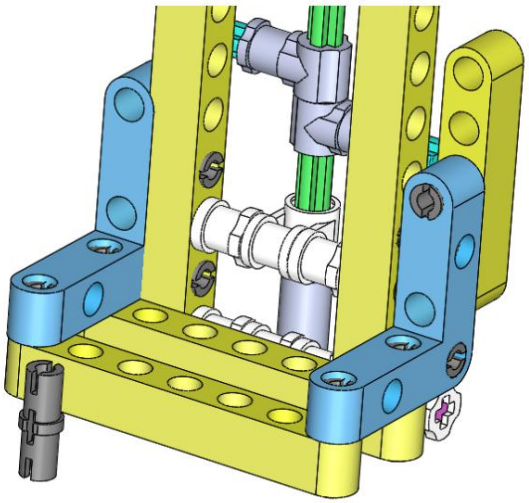
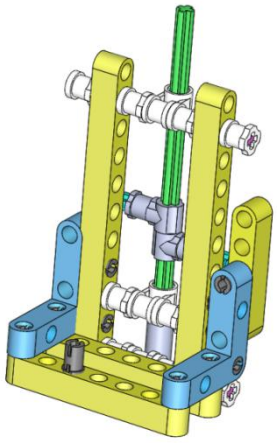
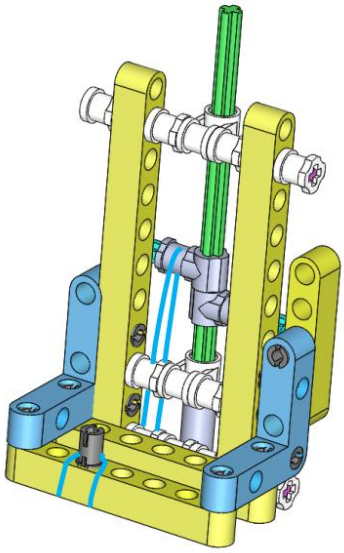
List		ity		
	the assembly part of step 6	1	PCS	
	building block X	4	PCS	
	building block H	2	PCS	
	Assembly Details			Assembly Demo
A	1. First install the building block X on the assembly part of step 6; 2. Install the building block H on the assembly part of step 6 through the building block X;			

					
Step	8	Assemble building blocks		Tools needed	
Parts List	Part Name	Quantity	Unit	Picture	
	the assembly part of step 7	1	PCS		
	building block X	1	PCS		
	building block Z	1	PCS		
	Assembly Details			Assembly Demo	

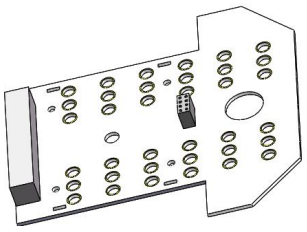
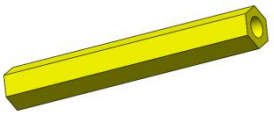
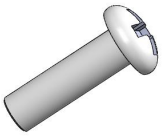
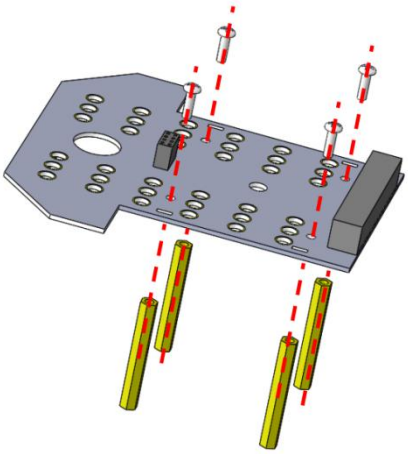
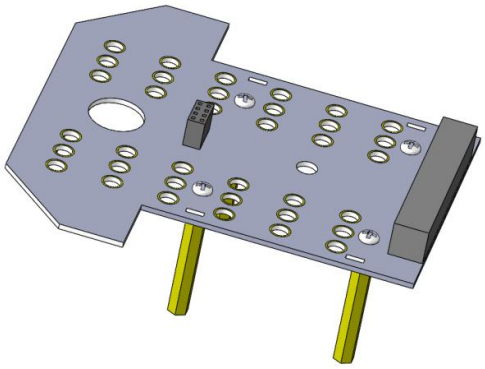
<table>
<tr><td>A</td><td>1. First install the building block X on the assembly part of step 7;
2. Install the building block Z on the assembly part of step 7 through the building block X;</td><td colspan="4"> 
  </td></tr>
<tr><td>Step</td><td>9</td><td colspan="2">Assemble building blocks</td><td>Tools needed</td><td></td></tr>
<tr><td rowspan="2">Parts List</td><td>Part Name</td><td>Quantity</td><td>Unit</td><td colspan="2">Picture</td></tr>
<tr><td>the assembly part of step 8</td><td>1</td><td>PCS</td><td colspan="2">  </td></tr>
</table>

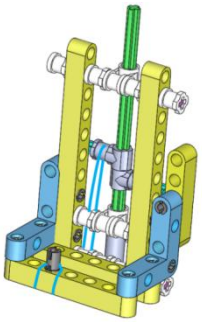
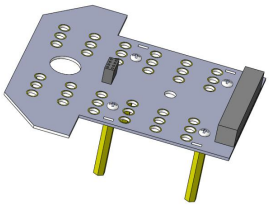
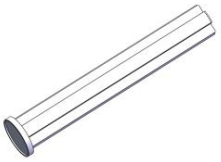
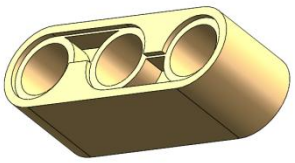
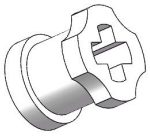
	building block X	4	PCS	
	building block E	2	PCS	
	Assembly Details			Assembly Demo
A	1. Assemble block X and block E together; 2. Install them on the assembly part of step 8;			

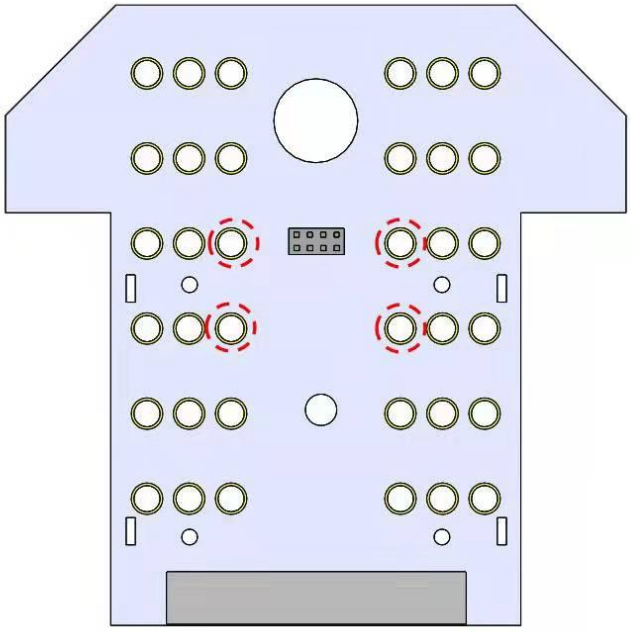
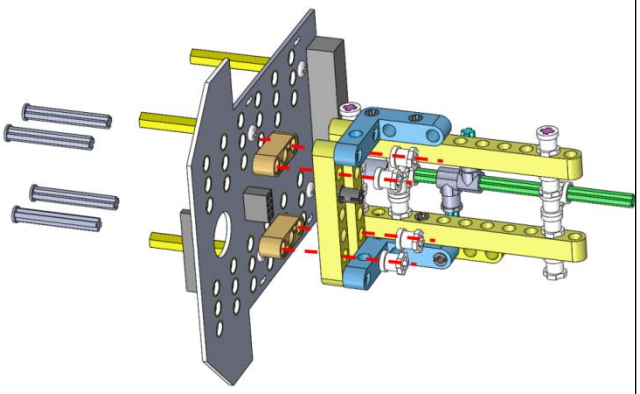
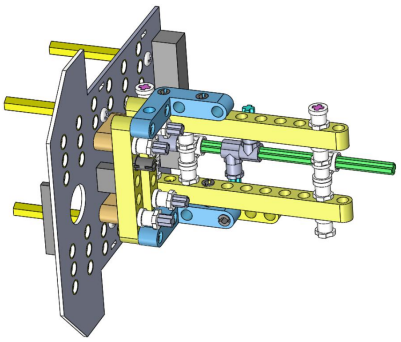
<table>
<tr><td></td><td colspan="3"></td><td>  </td></tr>
<tr><td>Step</td><td>10</td><td colspan="2">Assemble building blocks</td><td>Tools needed</td><td></td></tr>
<tr><td rowspan="4">Parts List</td><td>Part Name</td><td>Quantity</td><td>Unit</td><td colspan="2">Picture</td></tr>
<tr><td>the assembly part of step 9</td><td>1</td><td>PCS</td><td colspan="2">  </td></tr>
<tr><td>White rubber band</td><td>1</td><td>PCS</td><td colspan="2">  </td></tr>
<tr><td>building block X</td><td>1</td><td>PCS</td><td colspan="2">  </td></tr>
<tr><td></td><td colspan="3">Assembly Details</td><td colspan="2">Assembly Demo</td></tr>
</table>

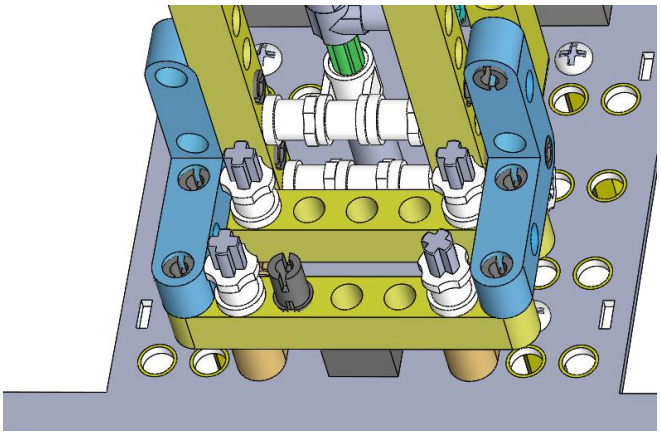
A	1. Install the building block X on the assembly part of step 9; 2. Install the rubber band on the assembly part of step 9; Note: The actual rubber band is white. The blue rubber band in the picture is convenient for you to see and understand the assembly details.	  

Step	11	Assemble the copper column on the circuit board	Tools needed	M3 slotted screwdriver 

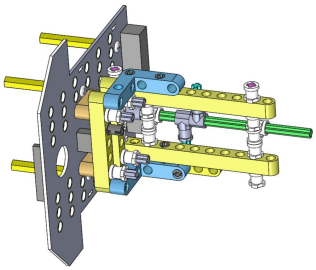
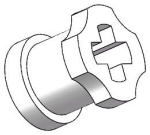
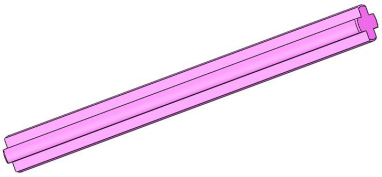
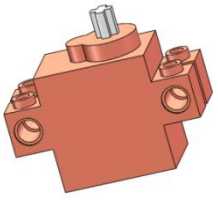
	Part Name	Quantity	Unit	Picture
Parts List	the circuit board	1	PCS	
	M3*40 copper pillar	4	PCS	
	M3*8 round head screw	4	PCS	
	Assembly Details			Assembly Demo
A	1. Use M3*8 round head screws to fix the M3*40 copper column on the circuit board;			 
Step	12	The building block part is assembled with the circuit board	Tools needed	

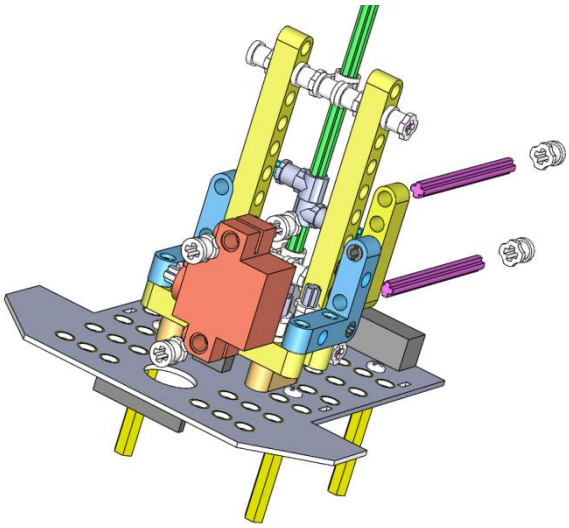
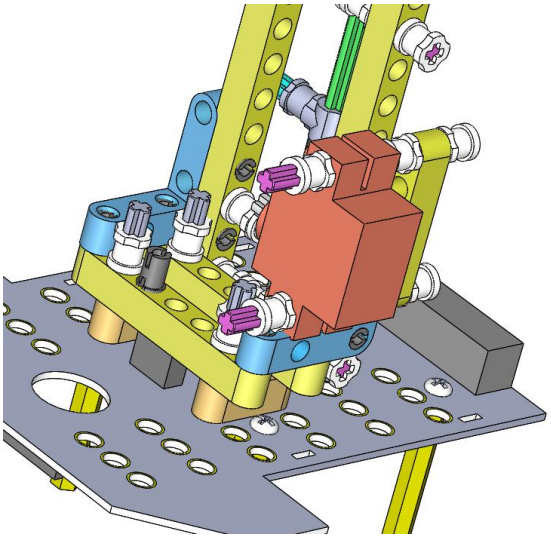
	Part Name	Quantity	Unit	Picture
Parts List	the assembly part of step 10	1	PCS	
	the assembly part of step 11	1	PCS	
	building block P	4	PCS	
	building block D	2	PCS	
	building block K	4	PCS	
	Assembly Details			Assembly Demo

A	1. The four building block P will pass through the 4 holes of the circuit board (as shown in the picture on the right) 2. Then they passes through the holes of two building block D 3. The four building block P passes through the four holes of the assembly part of step 10 4. Use four building blocks K to fix the assembly part of step 10 and the assembly part of step 11 together.	  

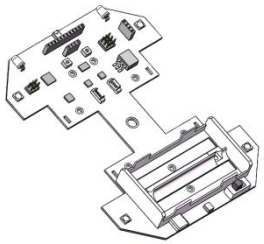
		

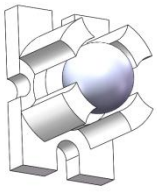
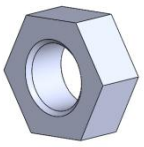
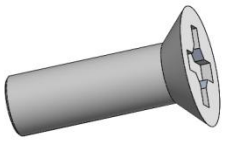
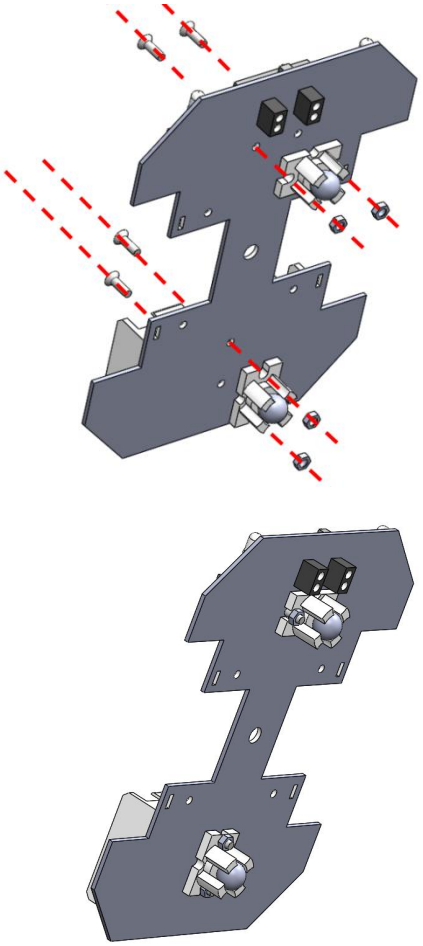
Step	13	Assemble the servo	Tools needed	

Parts List	Part Name	Quantity	Unit	Picture
	the assembly part of step 12	1	PCS	
	building block K	5	PCS	
	building block A （55mm）	2	PCS	
	Servo	1	PCS	
	Assembly Details			Assembly Demo

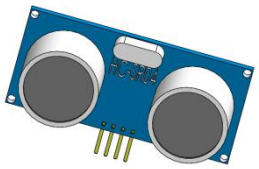
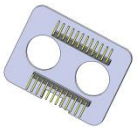
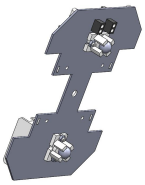
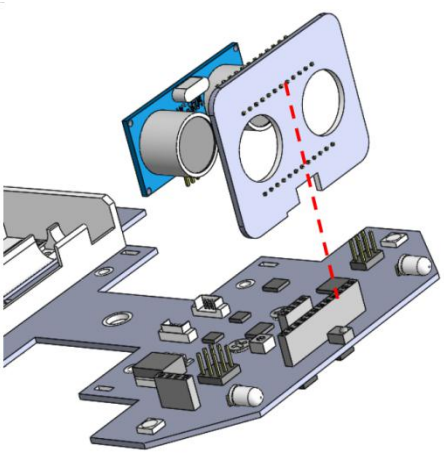
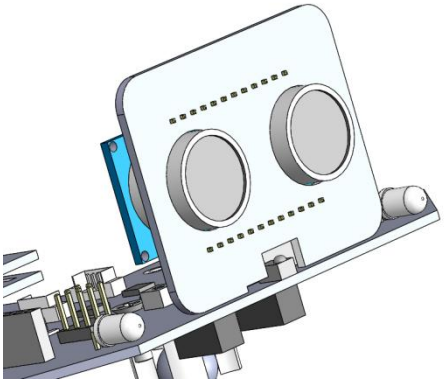
A	Use building block K and building block A to install the servo on the assembly part of step 12; Pay attention to the installation direction of servo;	 

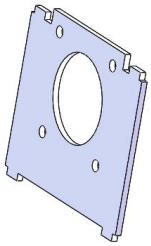
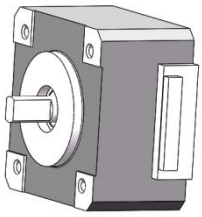
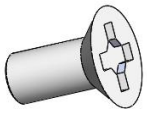
Step	14	Assemble universal wheel	Tools needed	M3 flat-blade screwdriver and wrench 

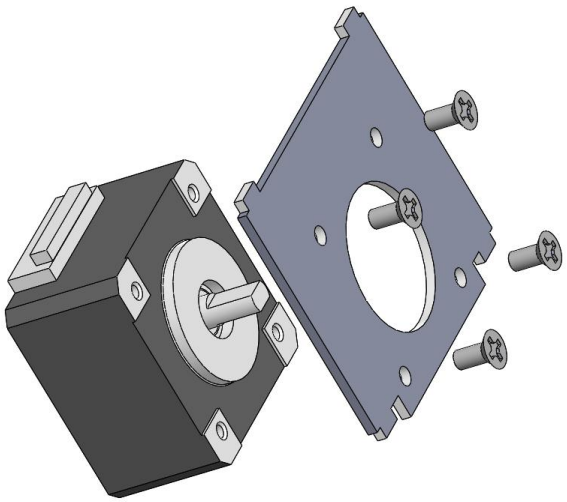
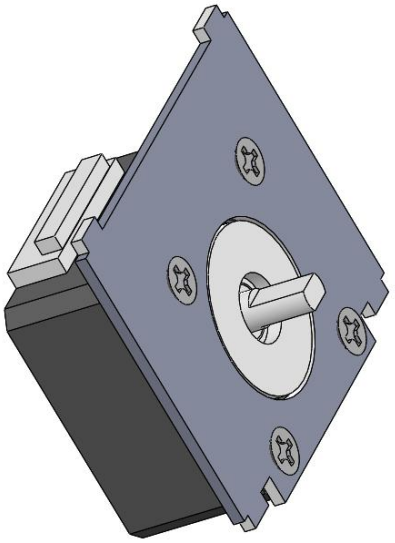
Parts List	Part Name	Quantity	Unit	Picture
	control board	1	PCS	

	universal wheel	2	PCS	
	nut	4	PCS	
	M3*10 countersunk screw	4	PCS	
	Assembly Details			Assembly Demo
A	Use M3*10 countersunk screws and M3 nuts to fix the universal wheel on the control board;			

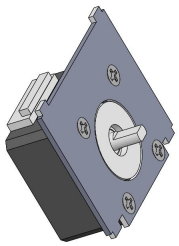
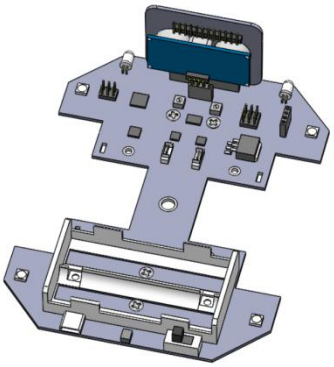
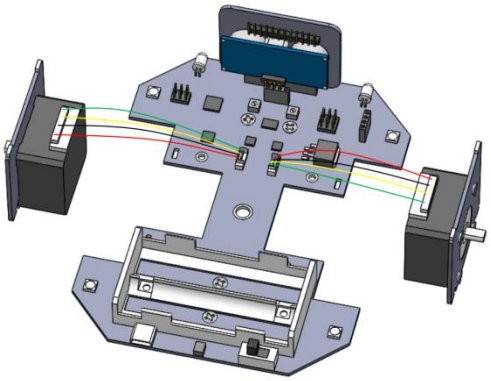
Step	15	Assemble the ultrasonic bracket and ultrasonic module on the control board		Tools needed	
Parts List	Part Name	Quantity	Unit	Picture	

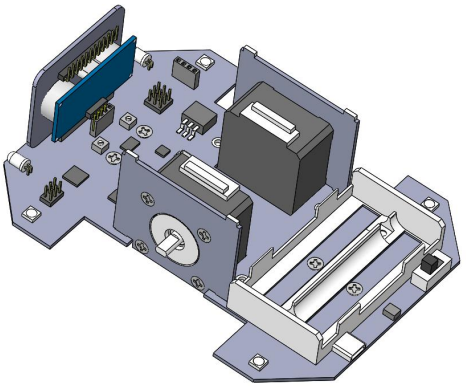
	ultrasonic module	1	PCS	
	ultrasonic bracket	1	PCS	
	the assembly part of step 14	1	PCS	
	Assembly Details			Assembly Demo
A	1. Assemble the ultrasonic module in the two holes of the ultrasonic bracket, 2. Then insert them into the corresponding pins of the control board; Pay attention to the installation direction of the ultrasonic bracket;			 
Step	16	Assemble the motor	Tools needed	M3 flat-blade screwdriver 

	Part Name	Quantity	Unit	Picture
Parts List	motor	2	PCS	
	motor bracket	2	PCS	
	M3*6 countersunk screw	8	PCS	
	Assembly Details			Assembly Demo

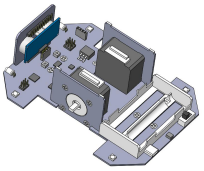
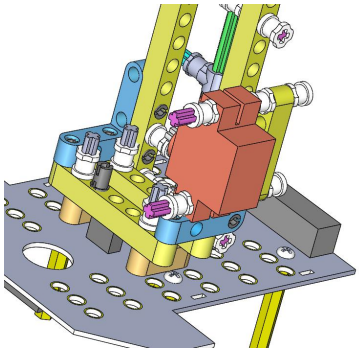
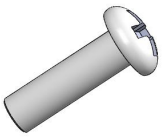
A	Use M3*6 countersunk head screws to install the motor on the motor bracket; Pay attention to the installation direction of the motor	 

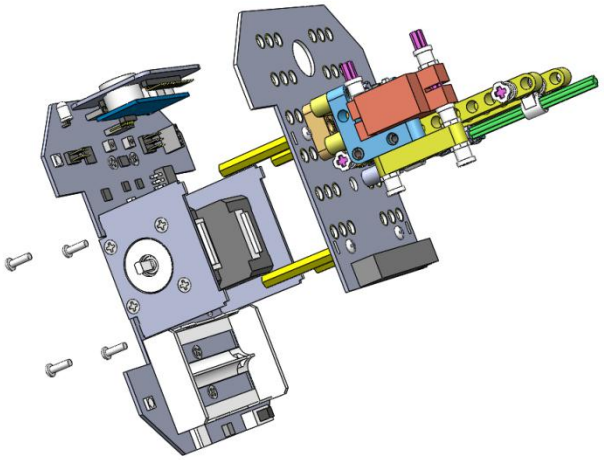
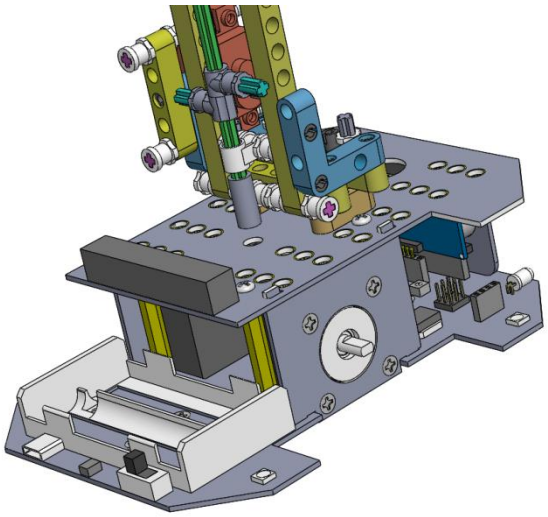
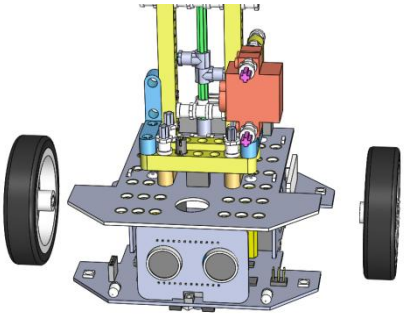
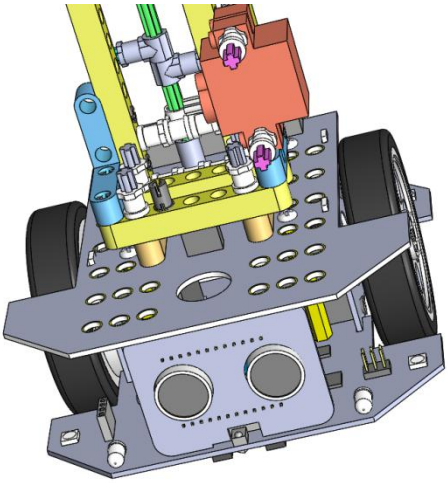
Step	17	Assemble the motor on the control board	Tools needed	

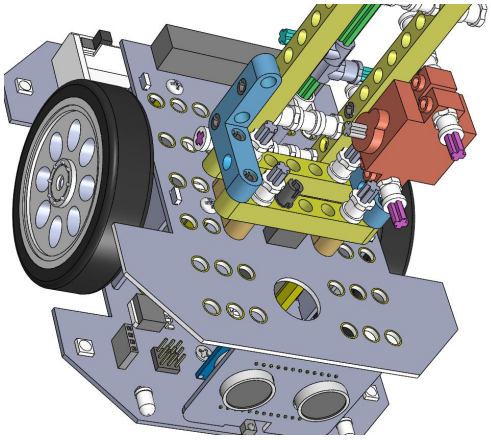
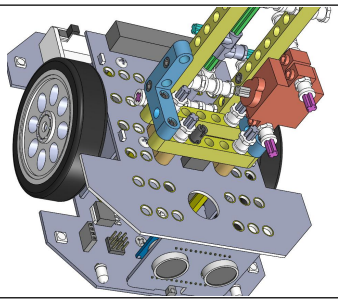
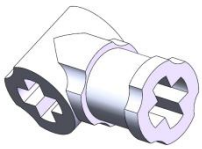
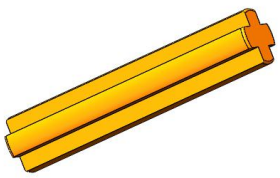
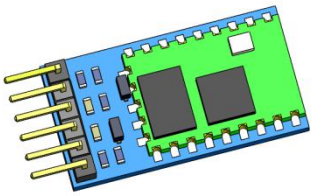
	Part Name	Quantity	Unit	Picture
Parts List	the assembly part of step 16	2	PCS	
	the assembly part of step 15	1	PCS	
	motor connection wire	2	PCS	
	Assembly Details			Assembly Demo
A	1. Connect one end of the cable to the motor and the other end to the control board Pay attention to the installation direction of the connecting line; 2. Fix the motor bracket to the control board;			

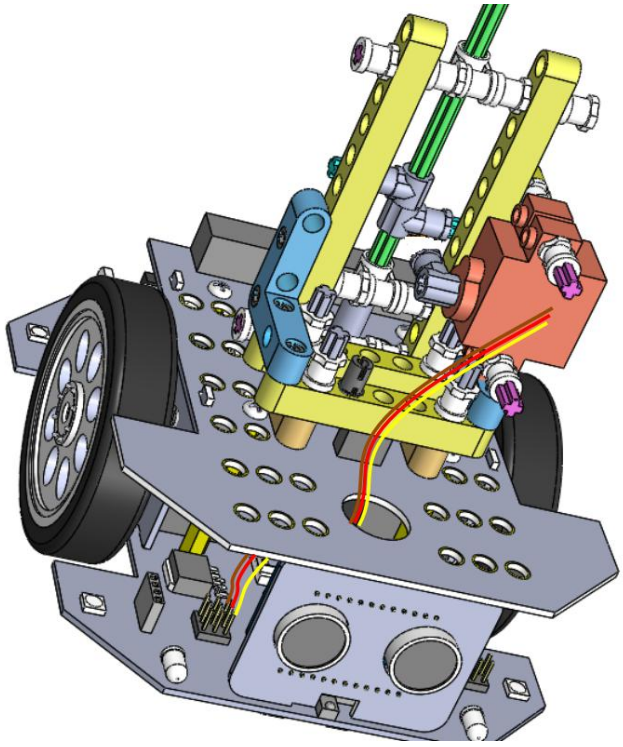
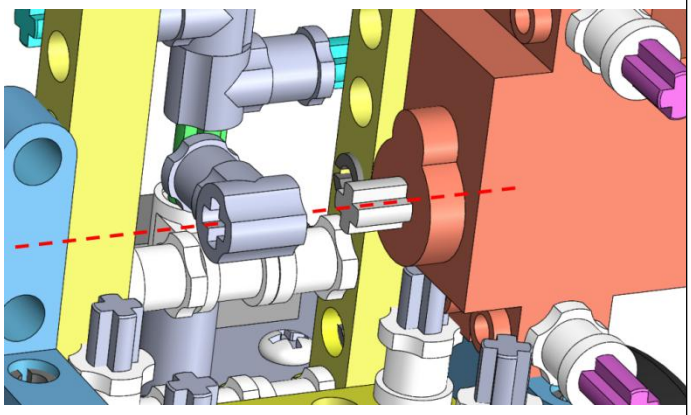
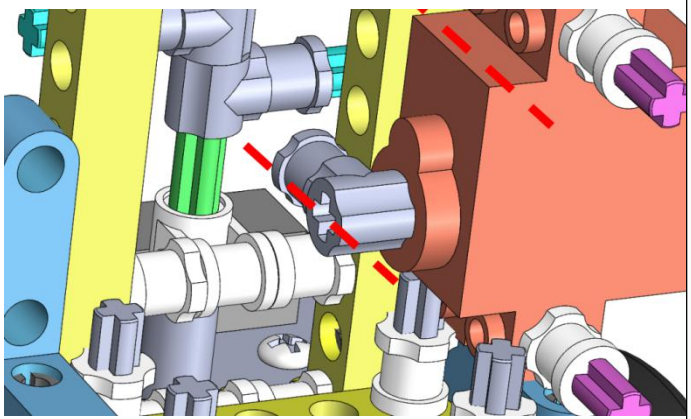
		

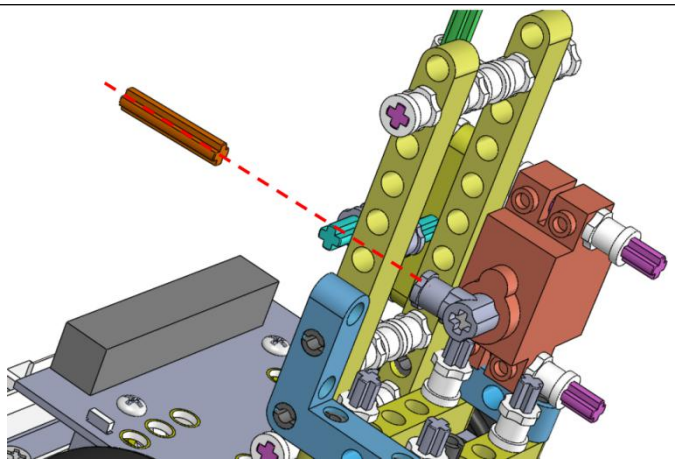
Step	18	Assemble the wheels	Tools needed	M3 slotted screwdriver 

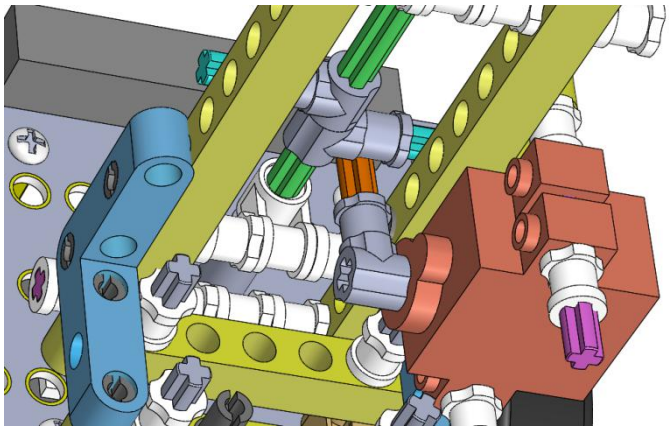
Parts List	Part Name	Quantity	Unit	Picture
	the assembly part of step 17	1	PCS	
	the assembly part of step 13	1	PCS	
	M3*8 round head screw	4	PCS	
	wheel	2	PCS	
	Assembly Details			Assembly Demo

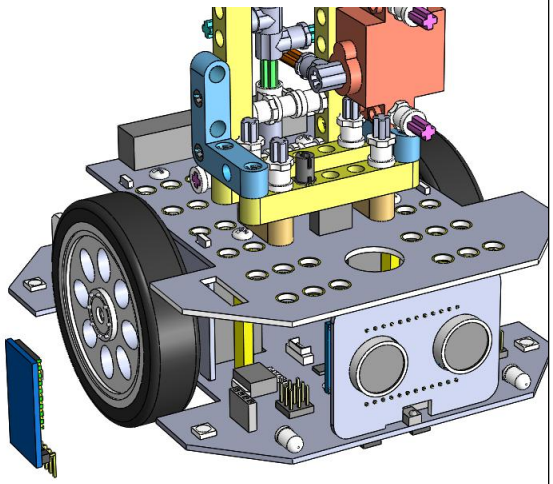
A	1. Use M3*8 round head screws to fix the assembly part of step 13 on the assembly part of step 17; 2.Both the upper circuit board and the lower control board have holes for securing the ultrasonic bracket and the motor bracket So at the same time you should fix the ultrasonic bracket and the motor bracket between assembly part of step 13 and the assembly part of step 17 3. Install the wheels;	   

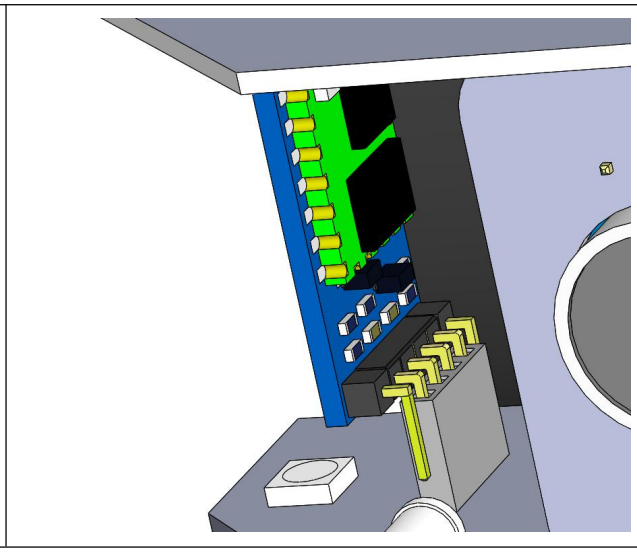
					
Step	19	Assemble building blocks/ Assemble bluetooth module		Tools needed	
Parts List	Part Name	Quantity	Unit	Picture	
	the assembly part of step 18	1	PCS		
	building block F	1	PCS		
	building block S （23.5mm）	1	PCS		
	bluetooth module	1	PCS		
	Assembly Details			Assembly Demo	

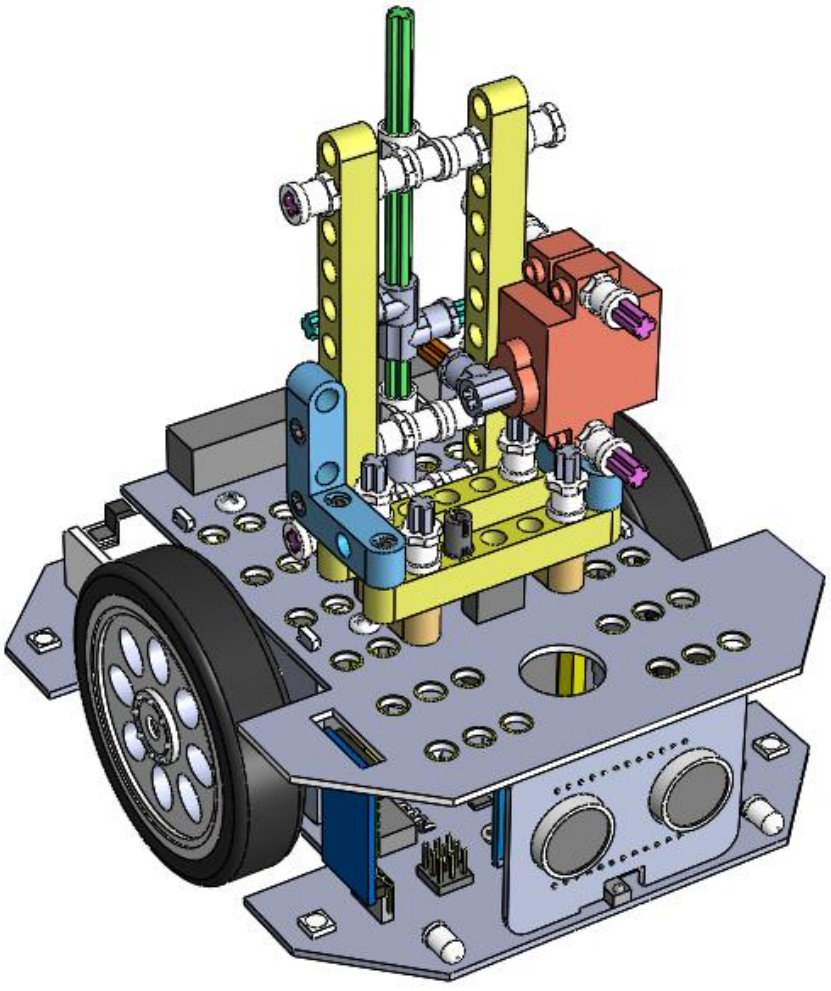
A	1.Connect the wire of the servo to the corresponding interface of the control board; The wiring method: The gray wire is connected to S1-G; The red wire is connected to S1-V; The yellow wire is connected to S1-S; 2.Install the building block F on the servo shaft at a horizontal angle; (note that before Installing the block F, you need to adjust the servo to # degrees according to the project #.) 3. Assemble the building block S and the building block F together; 4. Install the bluetooth module on the bluetooth interface on the control board; Note: Although Bluetooth is equipped with 6 pins, only 4 pins are used in actual assembly. The connection method is: VCC connects to pin V of the control board GND connects to pin G of the control board TXD connects to pin D8 of the control board RXD connects to pin D7 of the control board	  

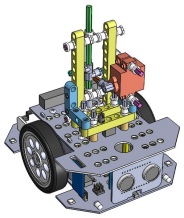
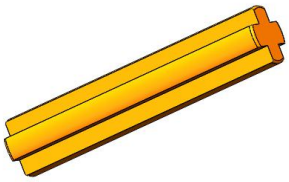
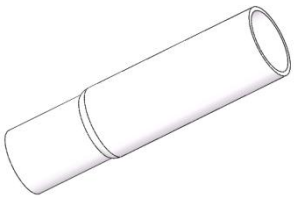
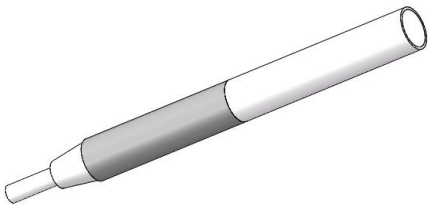
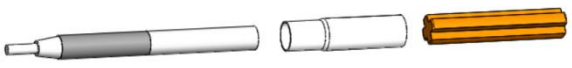


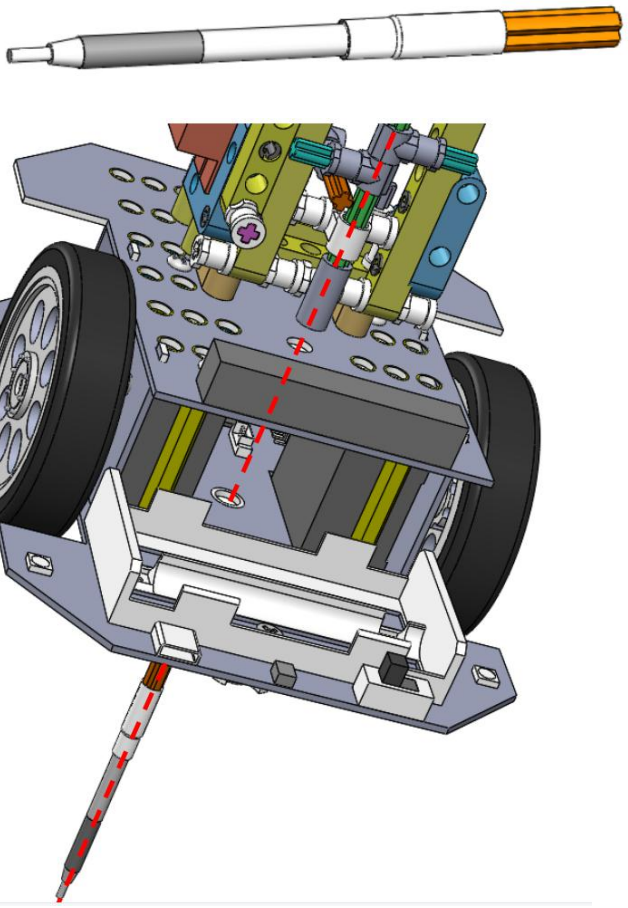
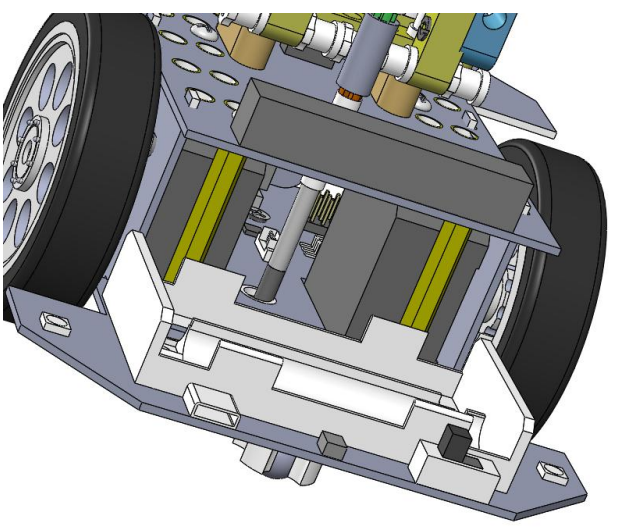




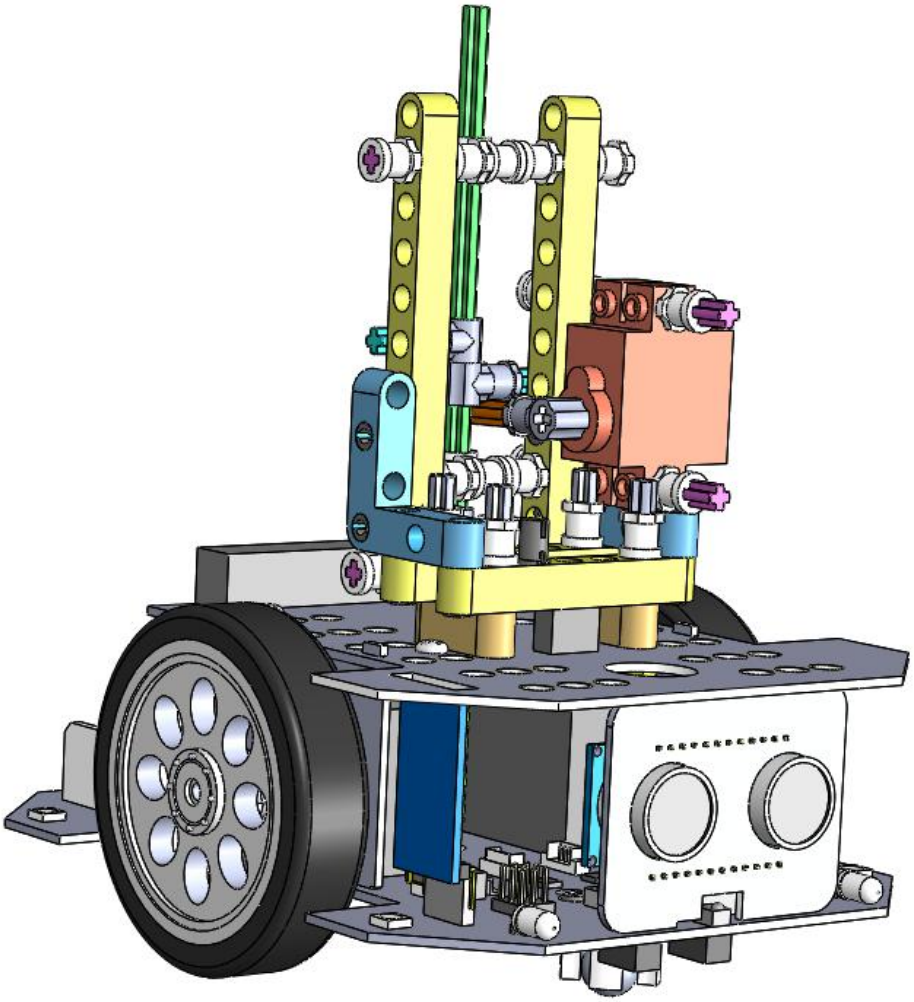




Note :	**The next step is to assemble the pen, but don't install the pen yet because the pen will touch the ground when assembled. We'll do other experiments first, and then install the pen when we're working on a writing project.**				
Step	20	Assembly the pen		Tool	
Parts List	Part Name	Quantity	Unit	Picture	
	the assembly part of step 19	1	PCS		
	building block S（23.5mm）	1	PCS		
	rubber tube	1	PCS		
	pen	1	PCS		
	Assembly Details			Assembly Demo	
A	1.As shown in the picture on the right, connect the pen to the narrow end of the rubber tube, and connect the wide end of the rubber tube to the building block S (note that the				

	pen and the building block S are in contact with each other in the rubber tube) 2. Pass the assembled pen through the holes of the control board and the upper circuit board and connect it with the building block B;	 

Congratulations, an interesting robot has been assembled!

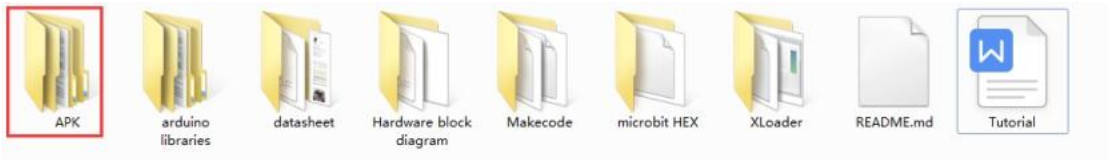


6. Bluetooth BitApp introduction

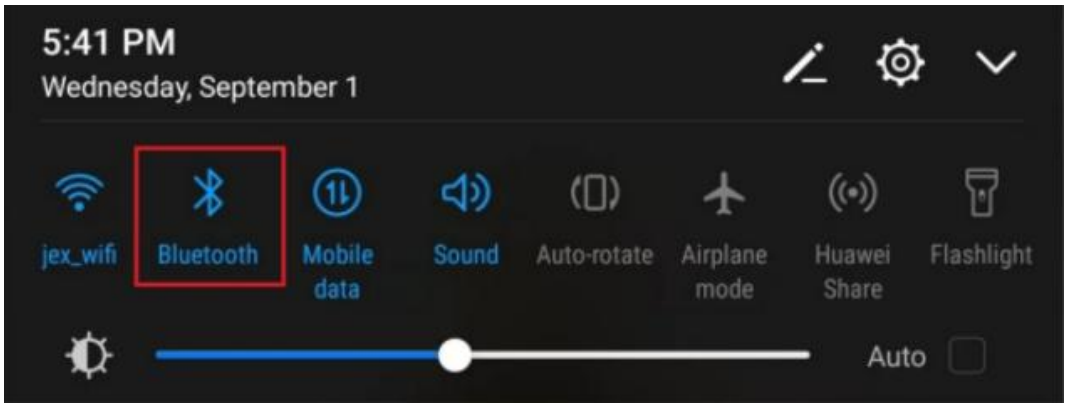
We design an Bluetooth APP (BitApp) which compatible with Android System for the Bluetooth module we provided in the kit and the integrated Bluetooth module of the micro:bit V1/2 board. It is very convenient to control Cobit with the Android mobile phone. The explanation and installation in this chapter are preserved for the subsequent Bluetooth control examples.

6.1 Install the Bitapp

The installation package is stored in the "APK" folder. Please copy the installation package to your phone and install it. If there is a risk prompt during the installation process, please continue to install it. This is because the security monitoring of the mobile phone judges the App as an unknown application. The application software itself does not carry any viruses or poses security risks. Please feel free to use it.



After the installation is complete, you need to turn on the Bluetooth function of your mobile phone, as shown in the figure below:



Then please click thee APP Icon 

, The interface of the Bluetooth APP is as follow:

6.2 Bluetooth APP Interface Introduction

1 Drop Down Menus

Click this button  , the drop-down menu has two option buttons, one navigates to the Bluetooth pairing page(

) and the other navigates to the App information page(

).

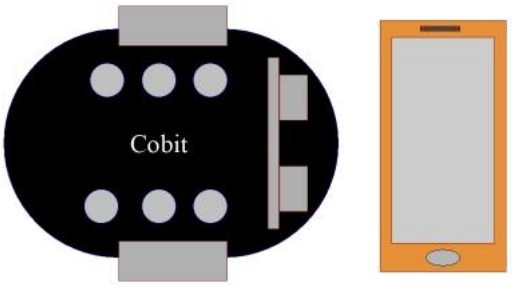
1.1 Bluetooth Pairing Page

Click this button 

to open the Bluetooth pairing page, you can search, select and pair Bluetooth. When the Bluetooth of the mobile phone is turned on, the App will automatically search for the Bluetooth nearby and display their name, UUID and RSSI. This App is only compatible with the HM-10 Bluetooth, and the Bluetooth on the Micro:bit V1/2 motherboard.

There are two ways to connect the bluetooth.

Automatic Connect Mode： Move the phone close to the Bluetooth module that needs to be connected, and then click the Bluetooth pairing icon. The App will automatically search for Bluetooth and judge the distance between the phone and the connected Bluetooth based on the RSSI signal strength. When the distance is less than or equal to the preset strength, it will automatically connect the Bluetooth.



Manually Connect Mode： Click into the Bluetooth pairing page, the App will automatically search for Bluetooth, and click the name of the Bluetooth you need to connect, and then pair the Bluetooth.

A successful Bluetooth pairing does not mean that the app and the Bluetooth module can communicate normally. The app can’t communicate with the Bluetooth until you can see the "Bluetooth ok" that pops up on the APP interface, as shown in the following figure:

1.2 App Information Page

Click this button 

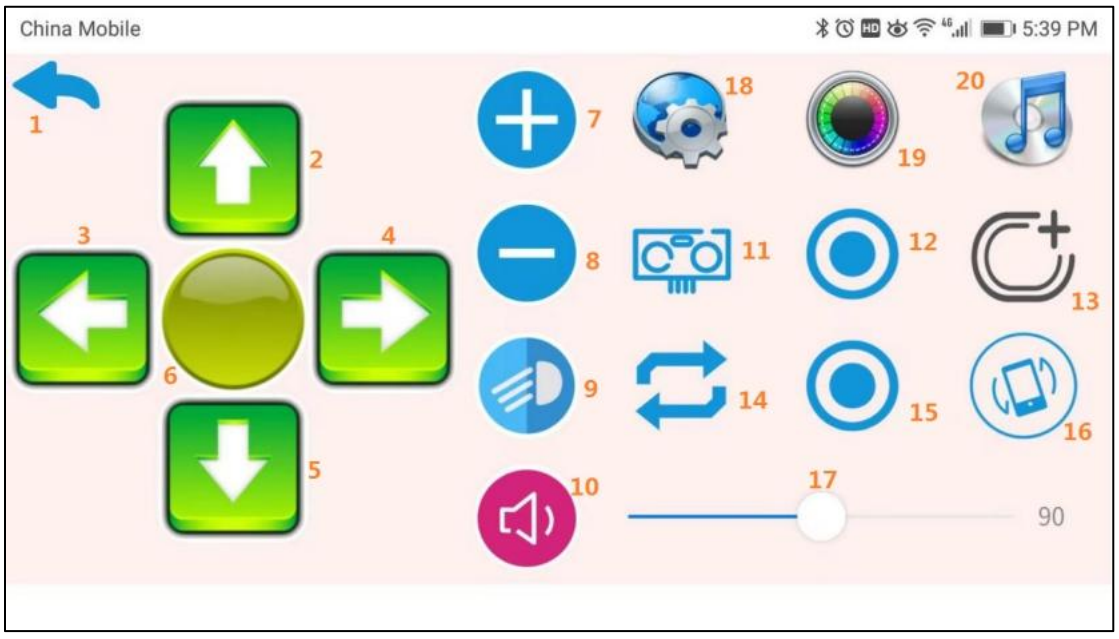
to open app information page. This page contains the internal information of the App, such as the design team, company, data communication format, version number, and update records.

4： Product Selection Button

This App can be applied to multiple products. In the middle of the app, you can see one picture of our products. If it is not the product you want to select and control, please slide the picture to the left or right to find the corresponding product.

Here we click on the picture of the cobit robot  .

Cobit Bluetooth Control Page Introduction - Home Page



Introduce the button of home page

1： Exit this page/Return to the previous page

2： button, button value： S-0x00-0x01-P

3： button, button value： S-0x00-0x02-P

4：  button, button value：S-0x00-0x04-P

5：  button, button value：S-0x00-0x05-P

6：  button, button value：S-0x00-0x03-P

7：  button, button value：S-0x00-0x06-P

8：  button, button value：S-0x00-0x07-P

9：  button, button value：S-0x00-0x08-P

10：  button, button value：S-0x00-0x09-P

11：  button, button value：S-0x00-0x0a-P

12：  button, button value：S-0x00-0x0b-P

13：  button, button value：S-0x00-0x0c-P

14：  button, button value：S-0x00-0x0d-P

15：  button, button value：S-0x00-0x0e-P

16：  button, button value：S-0x14-dataX-dataY-dataZ-P

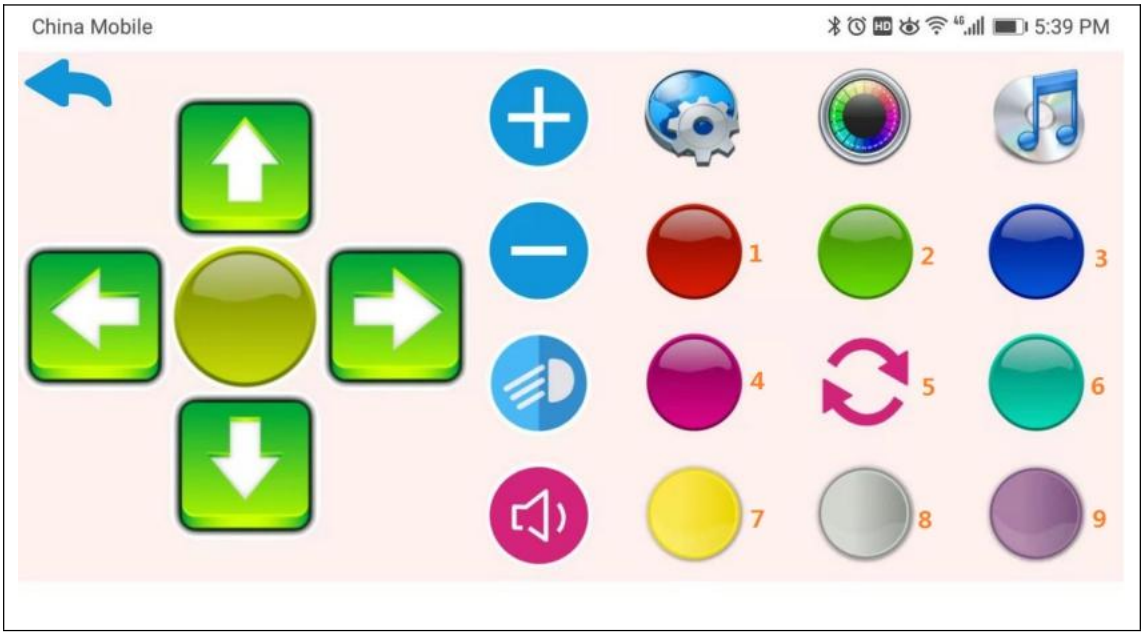
17：  Slider button, button value：S-0x0a-data-P

18：  Settings button

19：  Color switch button

20：  Music button

Page B： click the color switch button 



1：Color Red button, button value：S-0x00-0x0f-P

2：Color Green button, button value：S-0x00-0x10-P

3：Color Blue button, button value：S-0x00-0x11-P

4：Color Purple button, button value：S-0x00-0x12-P

5：Toggle button, button value：S-0x00-0x13-P

6：Color Sea Green button, button value：S-0x00-0x14-P

7：Color Yellow button, button value：S-0x00-0x15-P

8：Color Grey button, button value：S-0x00-0x16-P

9：Color Dark Purple button, button value：S-0x00-0x17-P

Page C： click the Music button 

1：do button，button value：S-0x00-0x18-P

2：re button，button value：S-0x00-0x19-P

1：mi button，button value：S-0x00-0x1a-P

3：fa button，button value：S-0x00-0x1b-P

4：sol button，button value：S-0x00-0x1c-P

5：la button，button value：S-0x00-0x1d-P

6：ti button，button value：S-0x00-0x1e-P

7. Geting start with Cobit for arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. You can use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Official Website： https://www.arduino.cc/

Arduino UNO R3 is a main board based on the ATmega328P control chip. Cobit is also

used an ATMEGA328P chip as the core processor and is burned the Arduino UNO R3 boot program at the factory. Therefore, it is fully compatible with Arduino UNO R3. When it is used on the arduino platform, it can be set in accordance with the UNO R3 parameters.

When configuring arduino IDE, select "arduino uno" for board type, and select com port assigned by driver of CP2102 in the device manager.

About Arduino IDE

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

Note：

If you have already installed Arduino IDE on your PC and have some experience with arduino, you can skip this chapter.

Install Arduino IDE

Install the Arduino Software (IDE) on Windows computer

Refer to: https://www.arduino.cc/en/Guide/Windows

Install arduino IDE on macOS System computer

Refer to: https://www.arduino.cc/en/Guide/macOS

Install arduino IDE on Linux System computer

Refer to: https://www.arduino.cc/en/Guide/Linux

Portable IDE (Windows and Linux)

Refer to: https://www.arduino.cc/en/Guide/PortableIDE

More information for Using Arduino IDE

Environment：https://www.arduino.cc/en/Guide/Environment

Official tutorials：https://www.arduino.cc/en/Tutorial/HomePage

Language Reference：https://www.arduino.cc/reference/en/

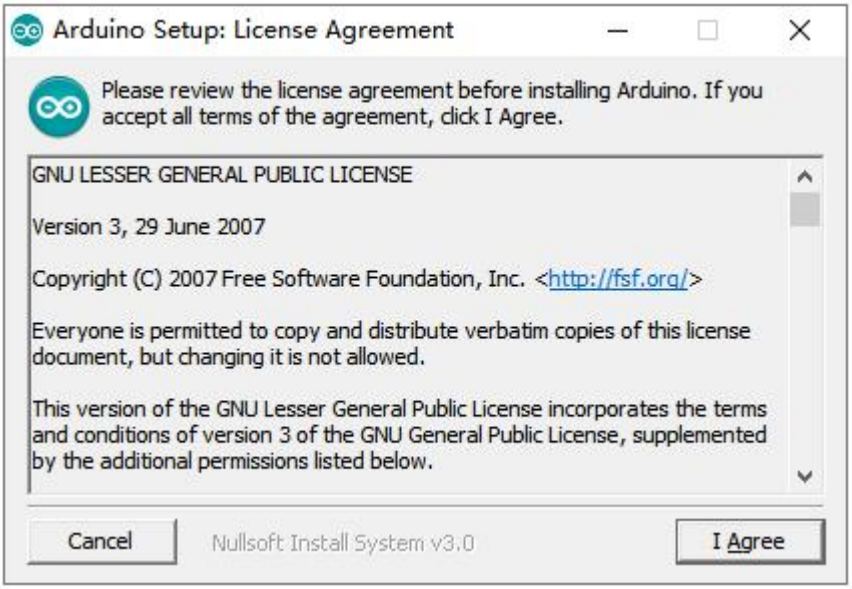
Here we detail how to Install the Arduino Software (IDE) on Windows PCs

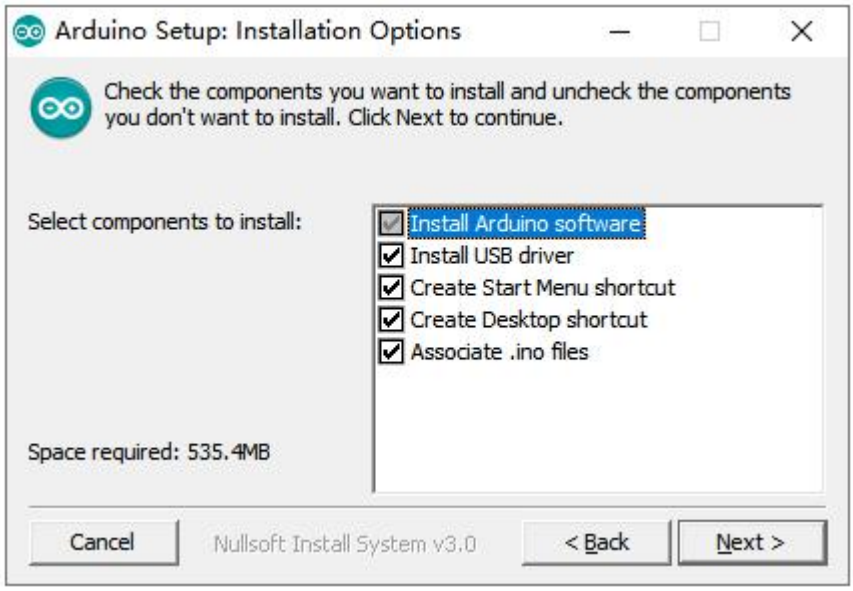
Get the latest version from this link (https://www.arduino.cc/en/Main/Software). You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually. The Zip file is also useful if you want to create a portable installation.

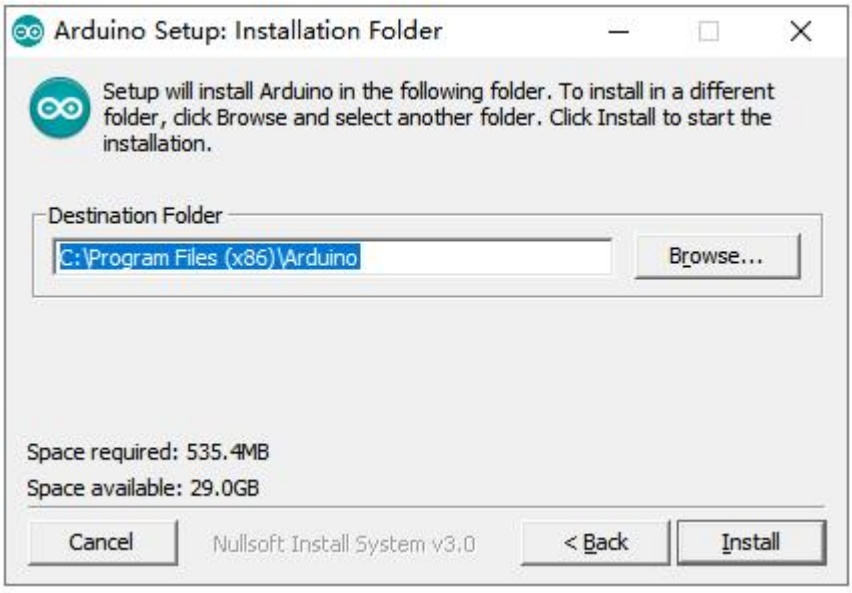
If we choose Installer (.exe) for Windows.

Click Windows Win 7 and newer and JUST DOWNLOAD.

Double-click the arduino IDE (.exe) file
Click "I Agree"
Click "Next"
Click "Install" to initiate installation.


Arduino Setup: License Agreement
Please review the license agreement before installing Arduino. If you accept all terms of the agreement, click I Agree.
GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007
Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.
This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.
Cancel
Nullsoft Install System v3.0
I Agree


Arduino Setup: Installation Options
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.
Select components to install:
Install Arduino software
Install USB driver
Create Start Menu shortcut
Create Desktop shortcut
Associate .ino files
Space required: 535.4MB
Cancel
Nullsoft Install System v3.0
< Back
Next >


Arduino Setup: Installation Folder
Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.
Destination Folder
C:\Program Files (x86)\Arduino
Browse...
Space required: 535.4MB
Space available: 29.0GB
Cancel
Nullsoft Install System v3.0
< Back
Install

The functions of each button on the Toolbar are listed below:

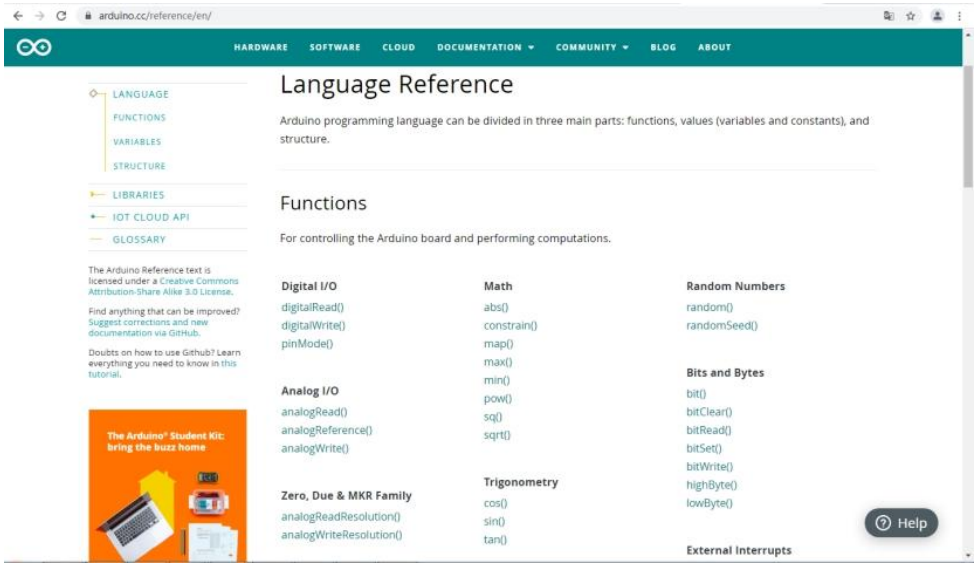
Verify/Compile	Check the code for errors
Upload	Upload the current Sketch to the Arduino
New	Create a new blank Sketch
Open	Show a list of Sketches
Save	Save the current Sketch
Serial Monitor	Display the serial data being sent from the Arduino

Arduino Programming Language

Arduino has a set of programming syntax and API interface based on C and C++, which can realize various functions. The Arduino programming language can be divided into three main parts: function, value (variable and constant) and structure.

For specific learning materials, please refer to：

https://www.arduino.cc/reference/en/

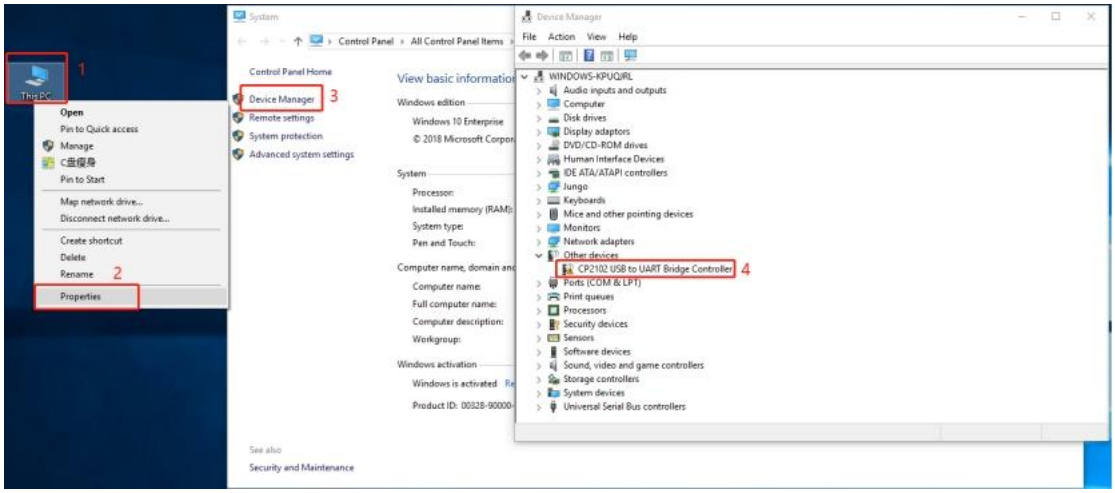


7.1 Install the USB-to-UART COM Port Driver

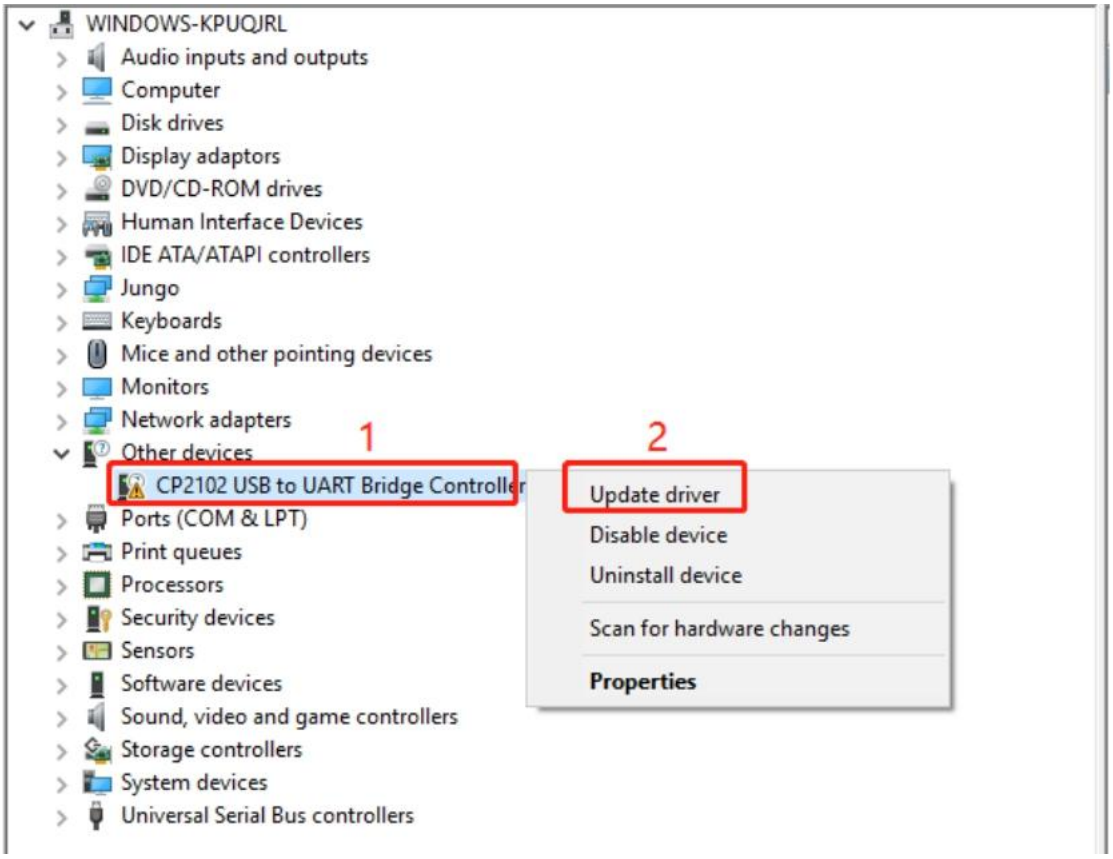
There is a Type-C USB port on the Cobit, and it integrates a CP2102 USB-to-serial chip. It can communicate with the serial port of ATmega328P. Before using the Cobit robot, you must install the driver of CP2102 chip, otherwise it will not communicate with computer.

The IDE of Arduino version 1.8 or higher has included the CP201X driver inside. When you connect the cobot robot to the USB port of the PC, the PC may automatically recognize and install the CP201X driver.

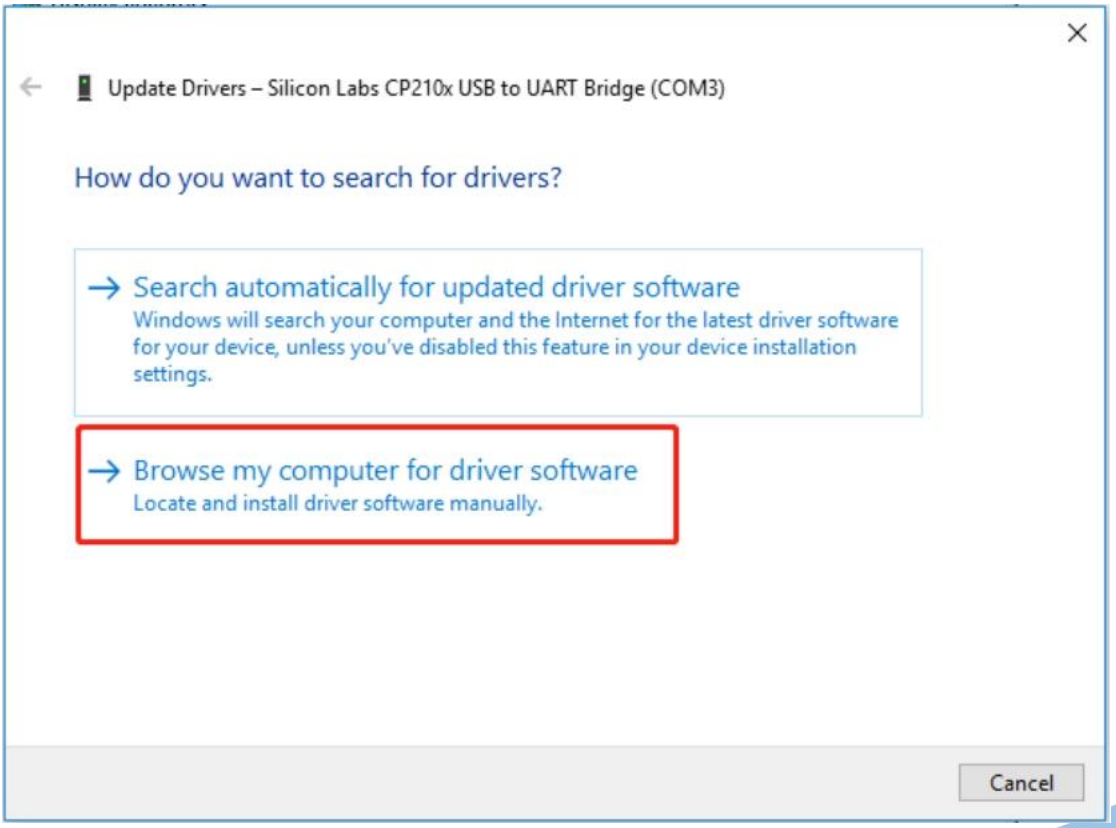
If the driver installation fails or you want to manually update the driver, select the "This PC" menu on the PC desktop, then right-click and select "Properties"--->"Device Manager".



A yellow exclamation mark means that the CP2102 driver installation failed, as shown above. Select this device, right click and select "Update driver".

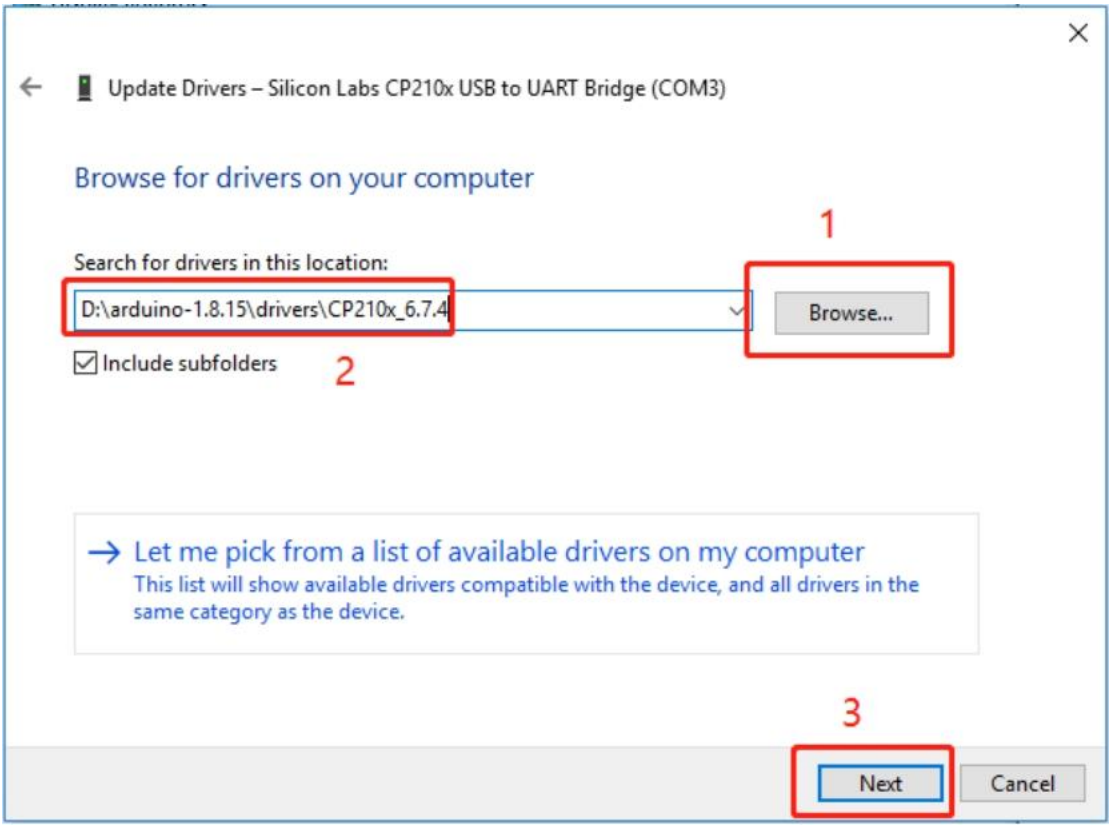


Click“browse my computer for updated driver software”。

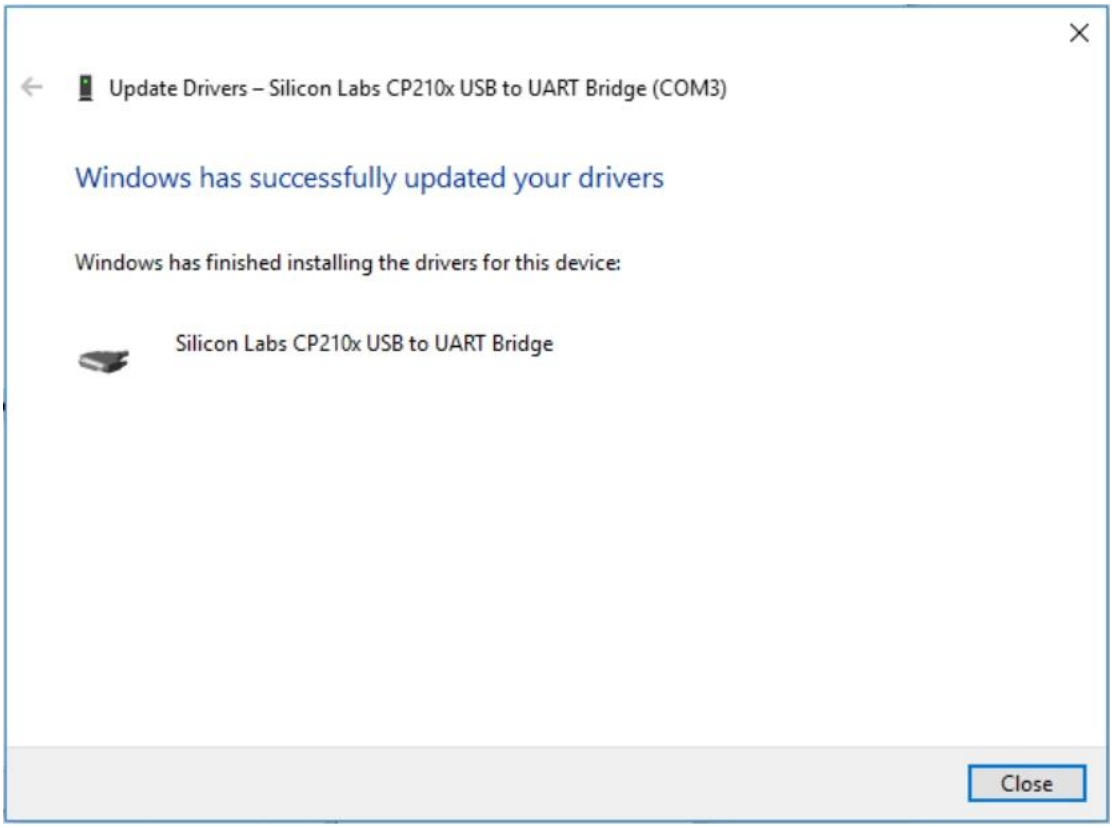


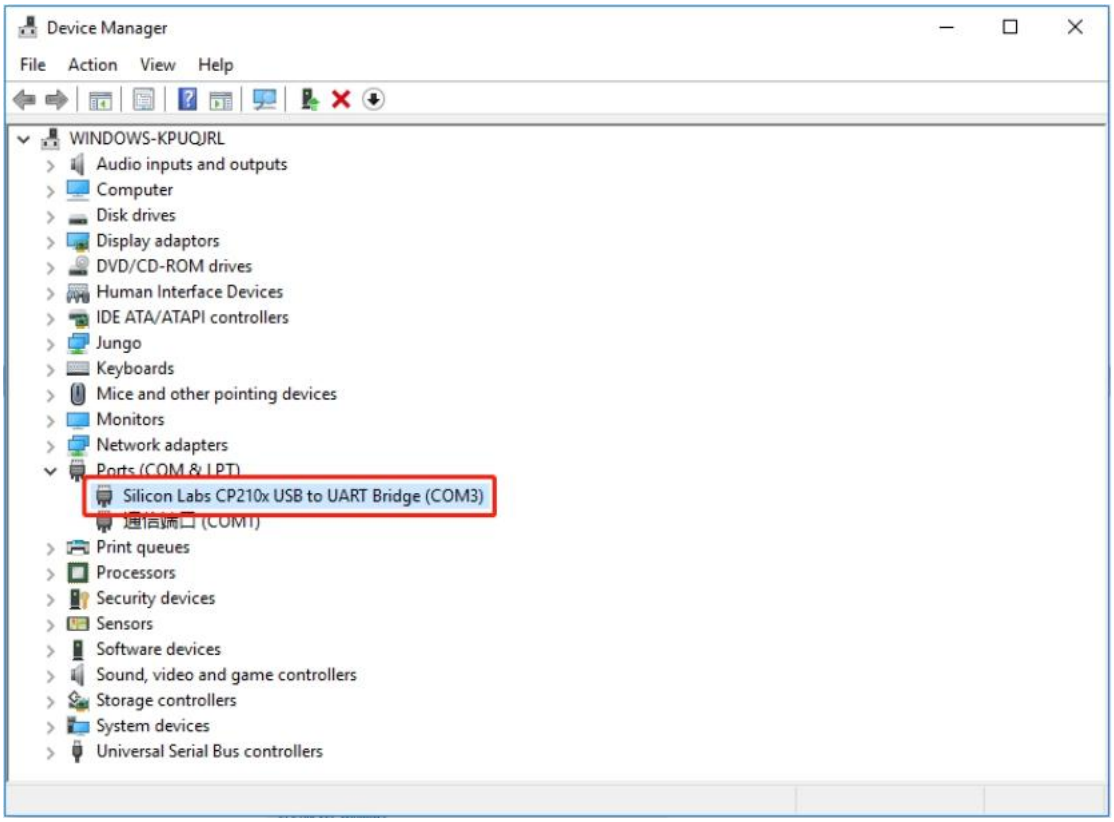
The IDE of Arduino version 1.8 or higher has included the CP201X driver inside. Please navigate to the drives folder of the Arduino Software you have download, open the driver folder and you can see the driver of CP210X drive. As showed in the below picture：

Click “Browse”, then click “Next”, the driver will be installed.



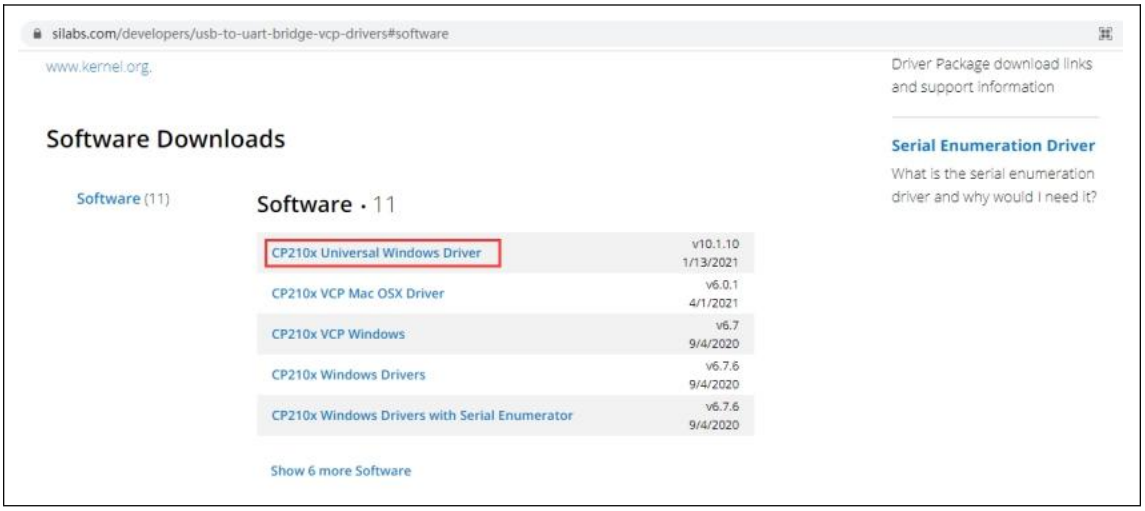
Open device manager again, we will see the yellow exclamation mark disappear. The driver of CP2102 is installed successfully.





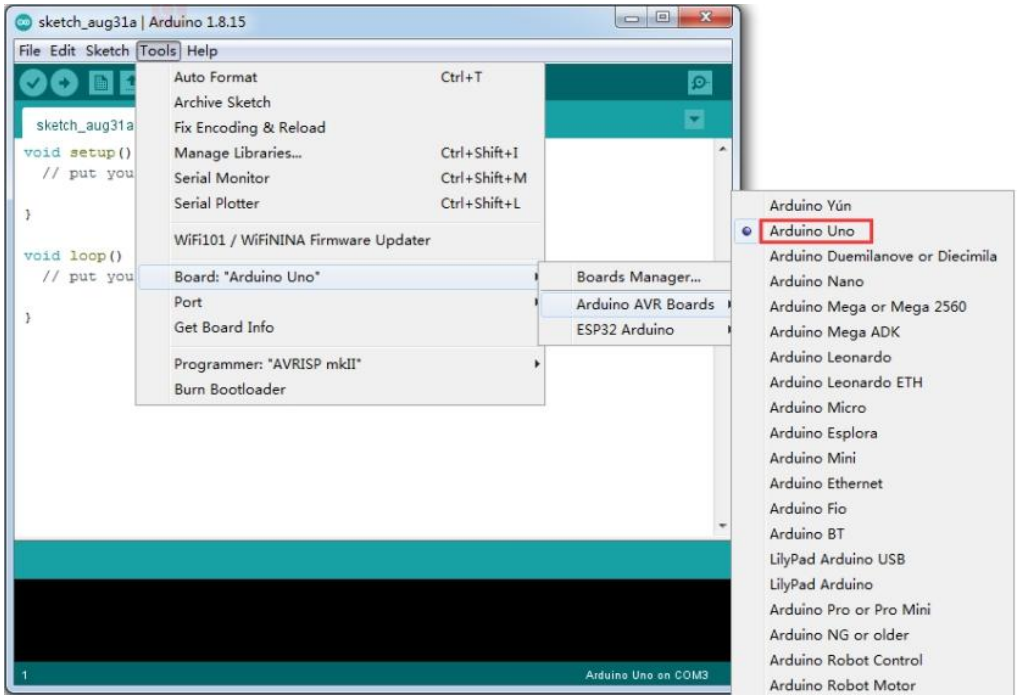
The driver of CP2102 can also be downloaded from the original official website, you can manual install it if your Arduino IDE version is not 1.8 or higher.

https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers#software



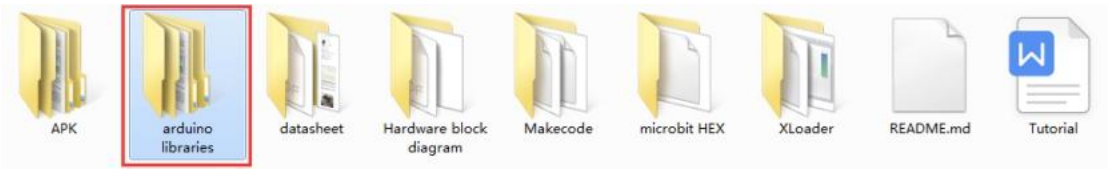
Cobit for Arduino Platform

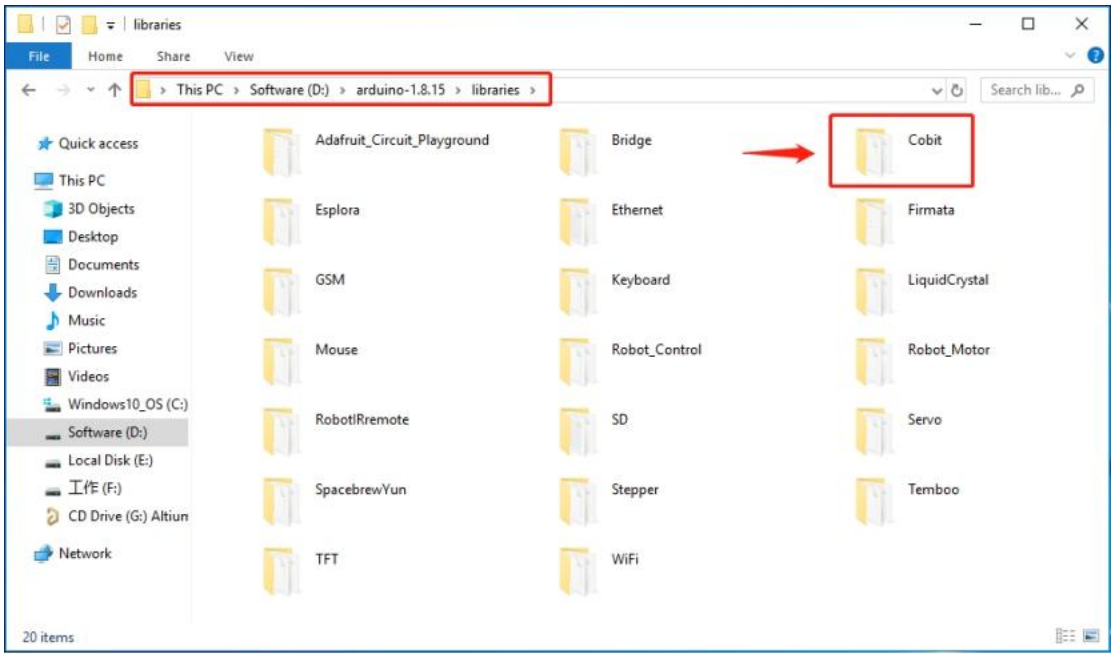
Cobit is used an ATMEGA328P chip as the core processor and is burned the Arduino UNO R3 boot program at the factory. When configuring arduino IDE, select "arduino uno" for board type, and select com port assigned by driver of CP2102 in the device manager.



7.2 Arduino library for Cobit

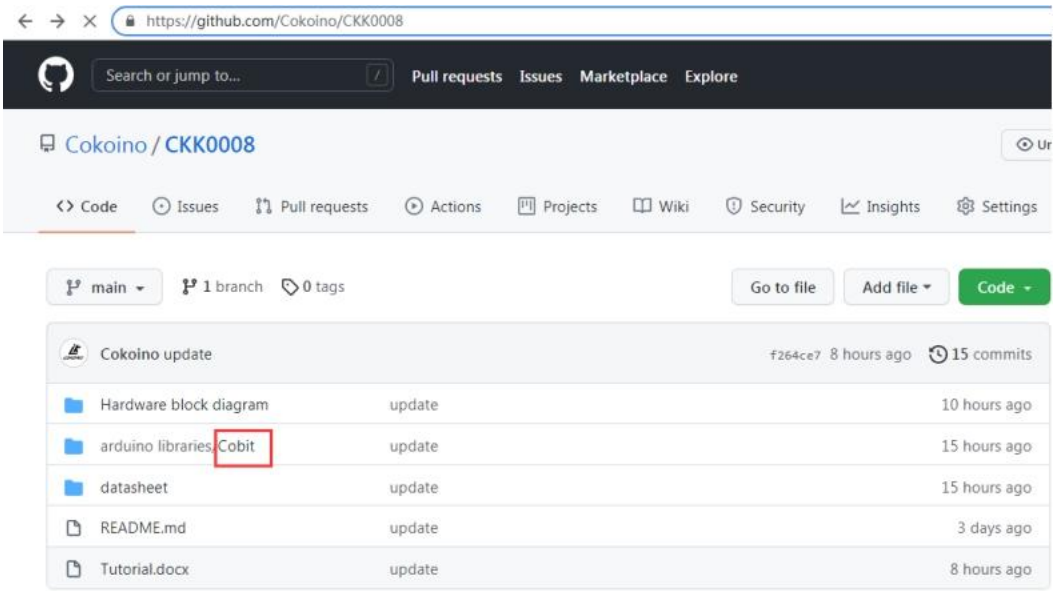
We have used the arduino platform to develop a library file for Cobit. This library contains all the functions of Cobit, making you to program the Cobit more easily and efficiently. The Cobit library files are saved in the arduino libraries folder provided by us, and you need to copy it to the libraries folder of the arduino installation folder.





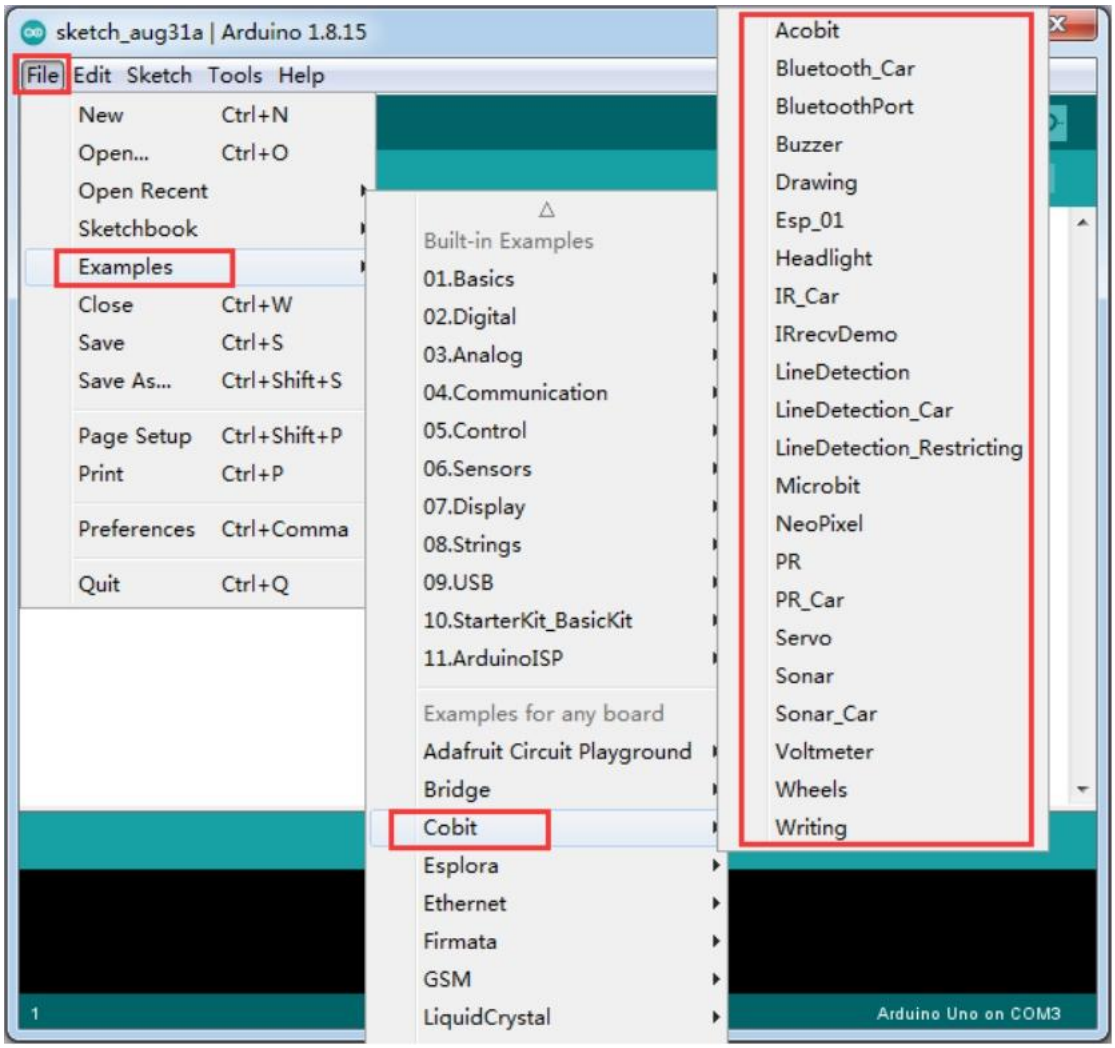
Note: This library is based on UNO R3 development board.

Cobit is also stored in： https://github.com/Cokoino/CKK0008



Arduino Example of the Cobit

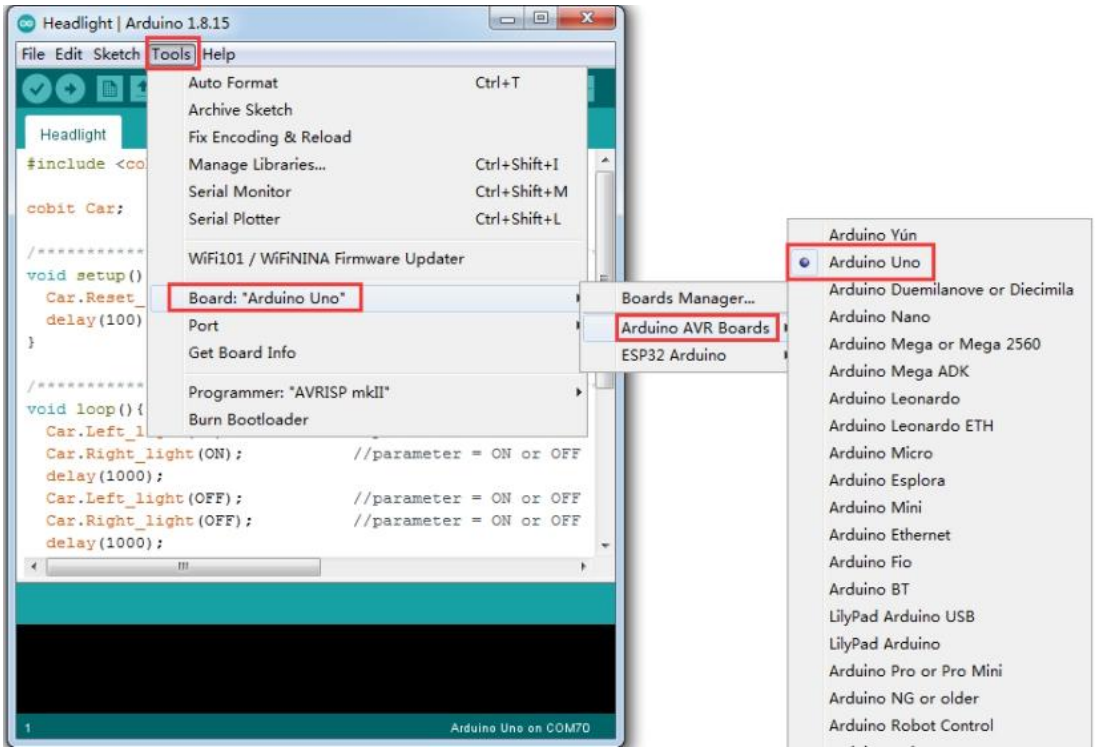
All functions we designed are integrated in the arduino library name “Cobot”. In the previous step, we have decompressed and copy the “Cobot” library file to the "libraries" folder of the arduino IDE. Now we can find all the sample codes of cobit in the arduino IDE, as following picture:



Note: After decompressing and copy the Cobit library file to the "libraries" folder under the Arduino installation directory, you need to close the IDE and restart to find the Cobit example in the Example drop-down menu of IDE.

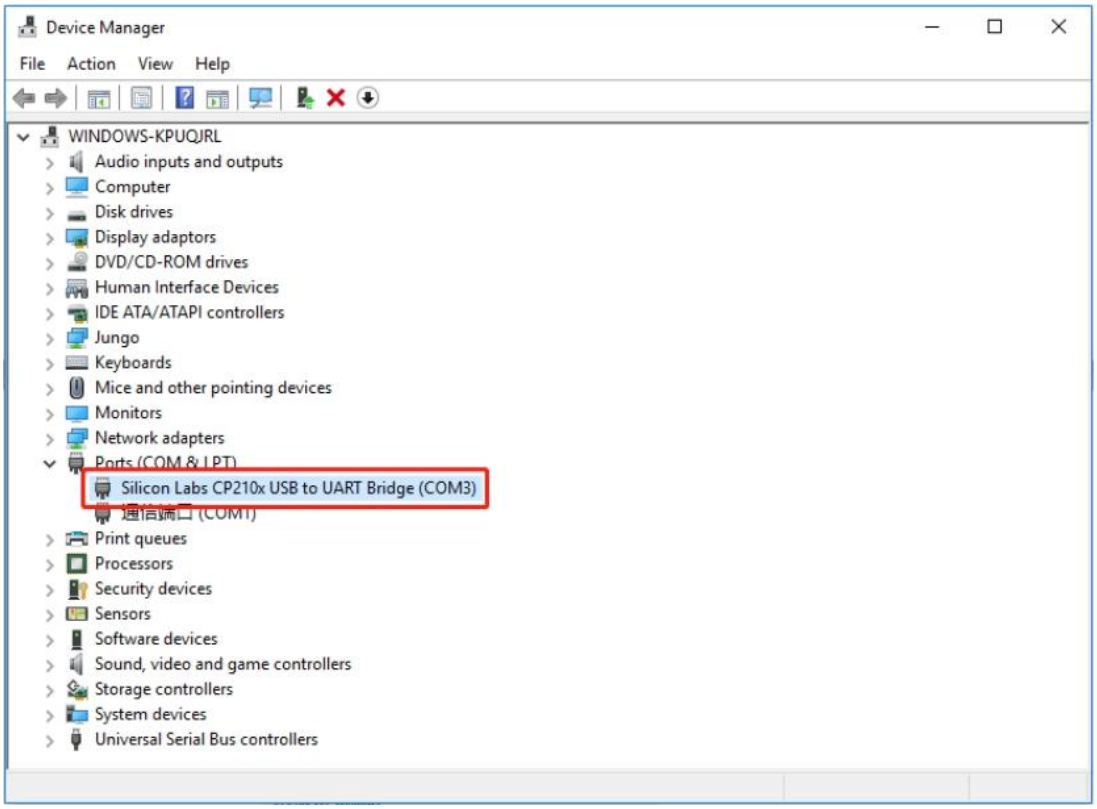
A simple example here:

Select board:



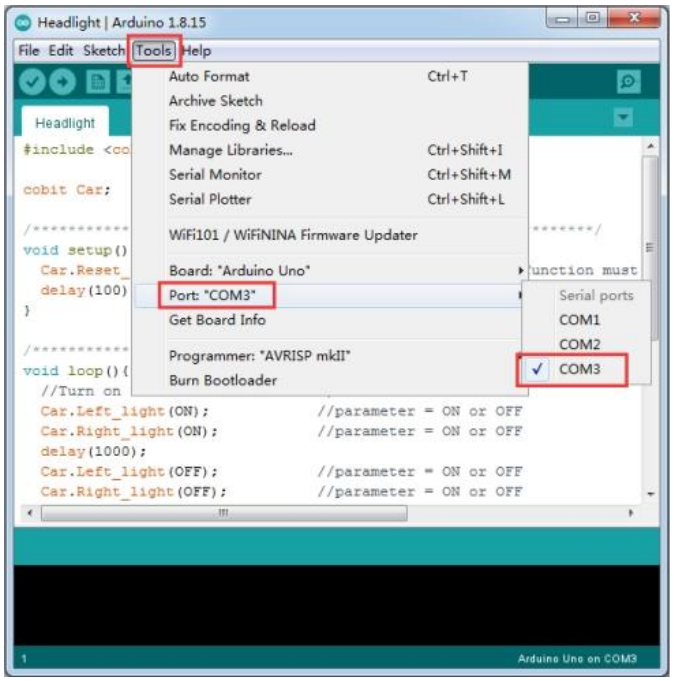
Select port:

Use the Type C USB cable to connect the Cobit to the PC (open the device manager of the PC, if the device displays an exclamation mark, please refer to the previous chapter to install the driver).

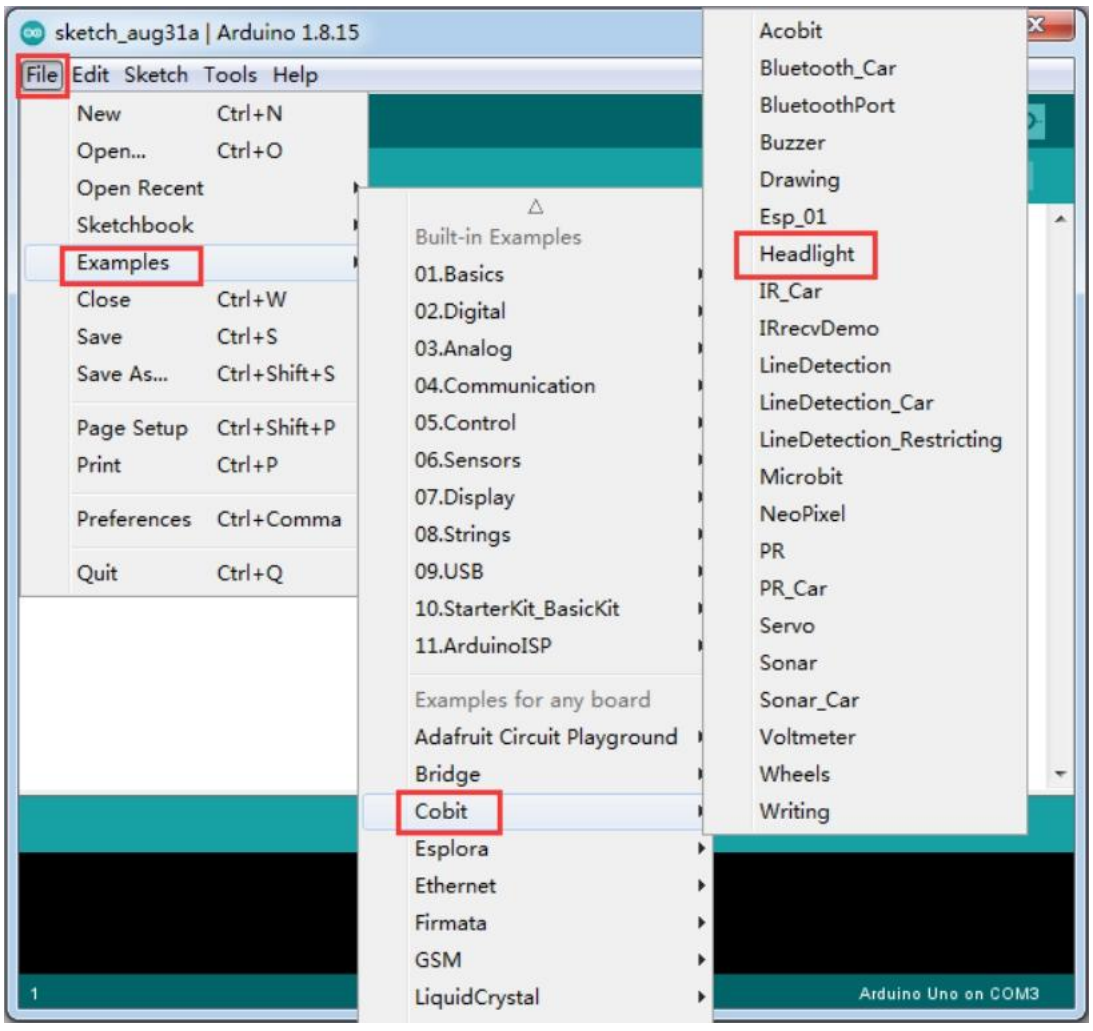


Select the corresponding port in the arduino IDE according to the com port the

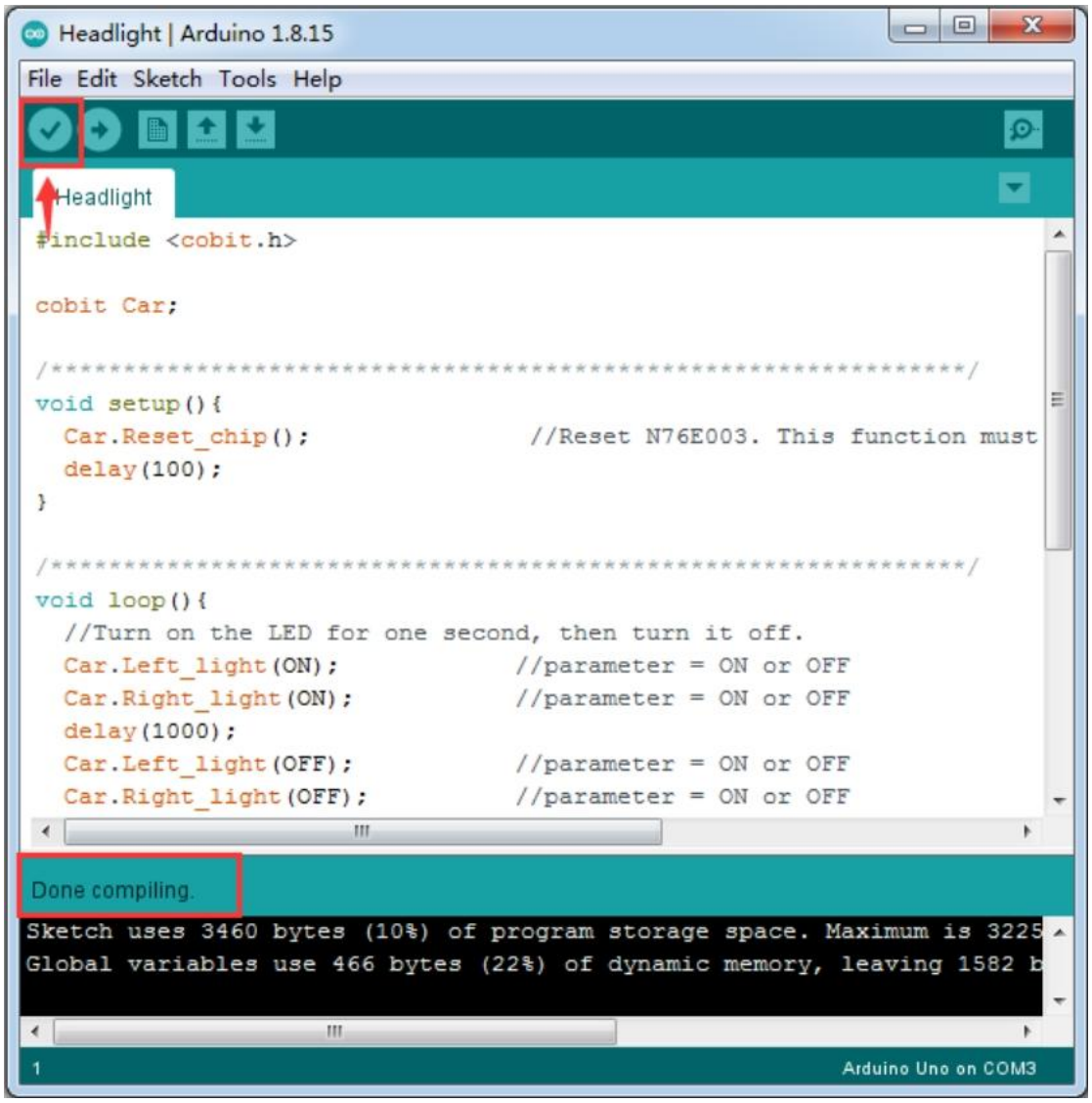
device manager displayed.



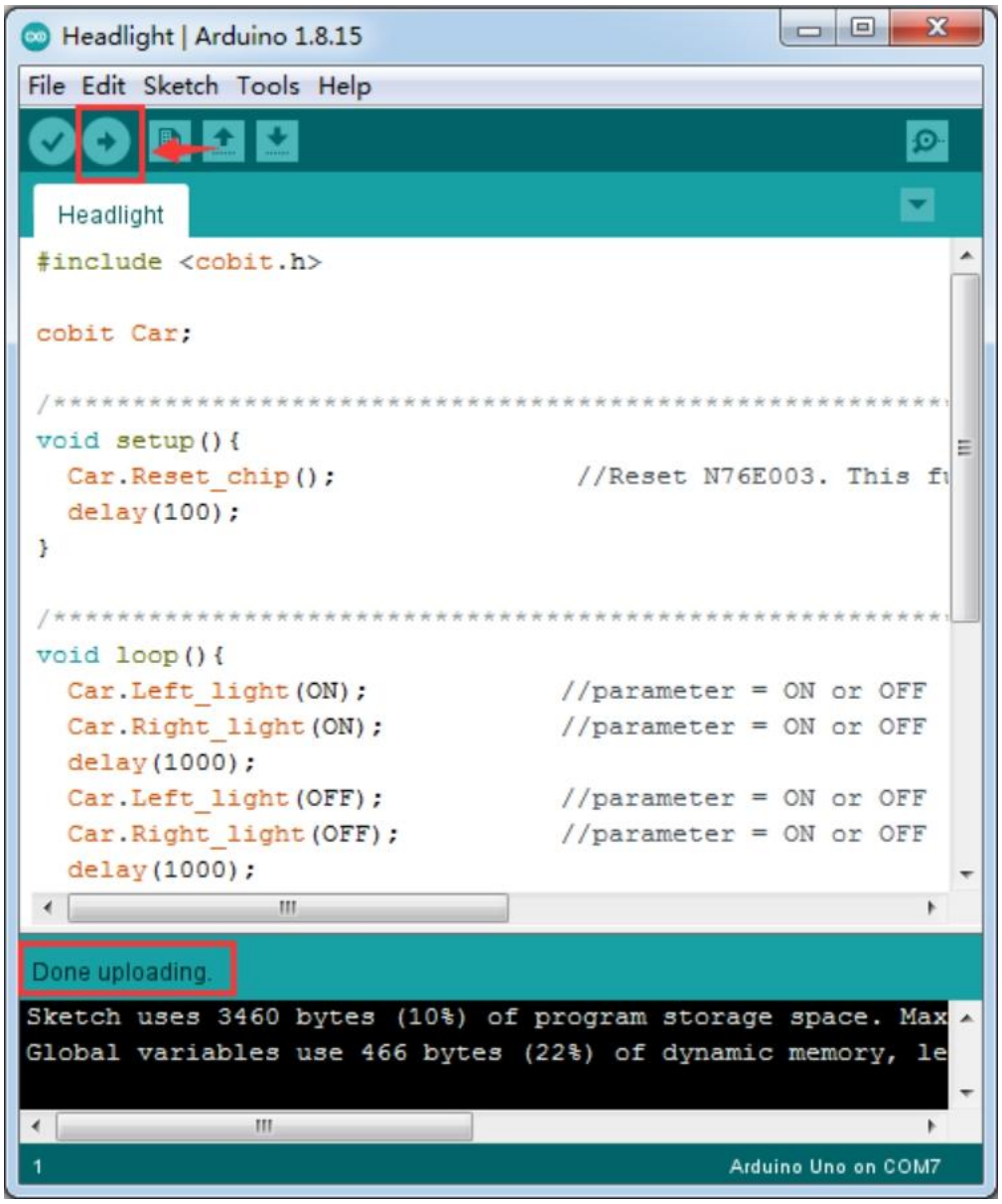
Choose code：



Check the code (if there is an error in the code, an orange error message will be printed in the information window at the bottom of the IDE).



Upload code:



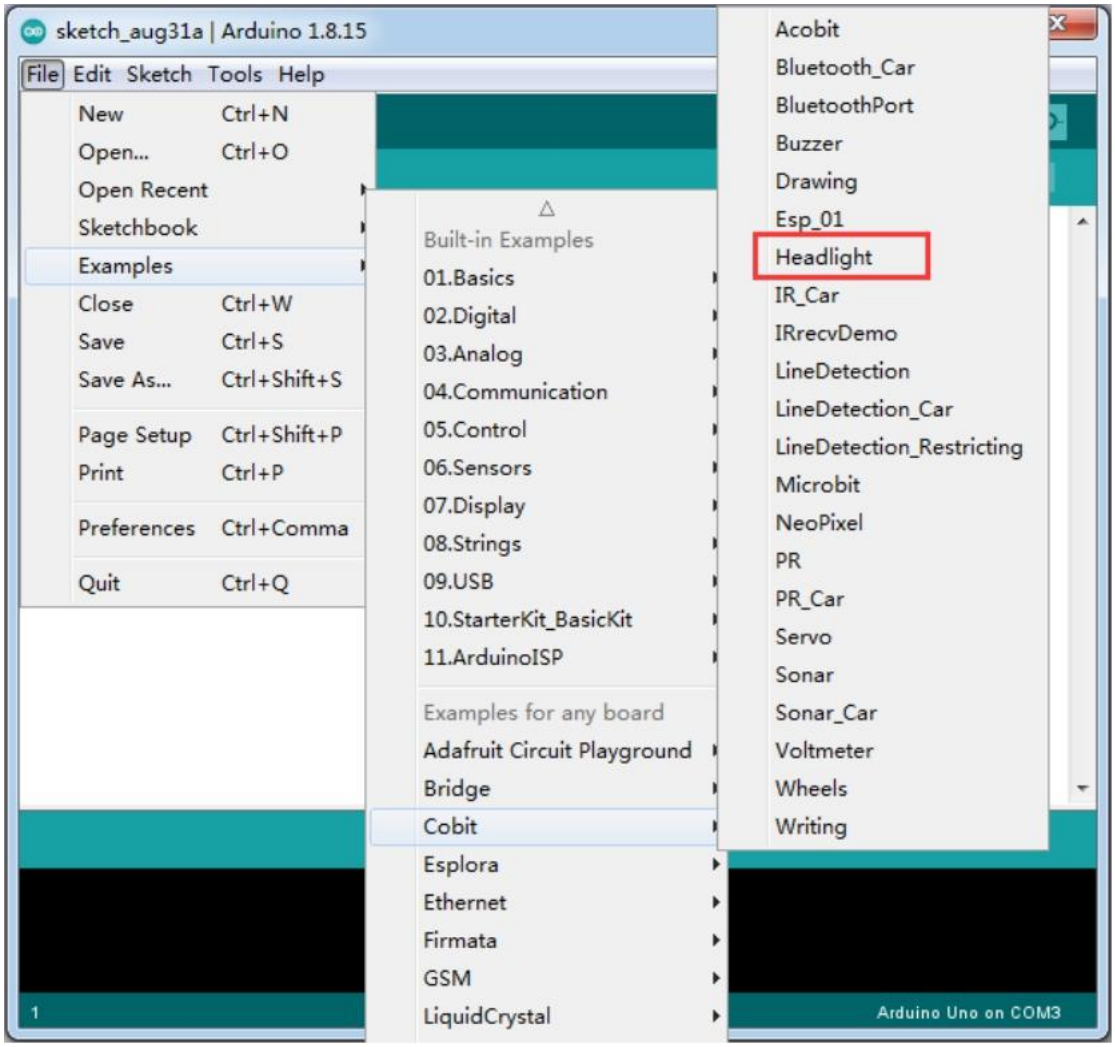
Result:

The two headlights of the Cobit will be on for 1 second and then off; then the breathing light will be on, and then off; Keep the loop like this

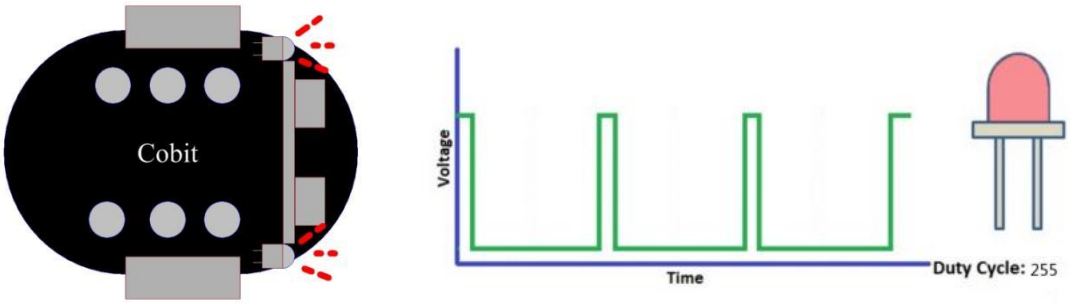
7.3 Example Tutorial

The sample courses of the Cobit have been integrated in the Cobit library, it is very convenient for you to make the robot functional and avoids errors caused by copying and pasting the code. Please carefully check the comments after the "//" in the code when using the example, it will be very helpful for you to understand the function of the code.

1_Examples of headlights



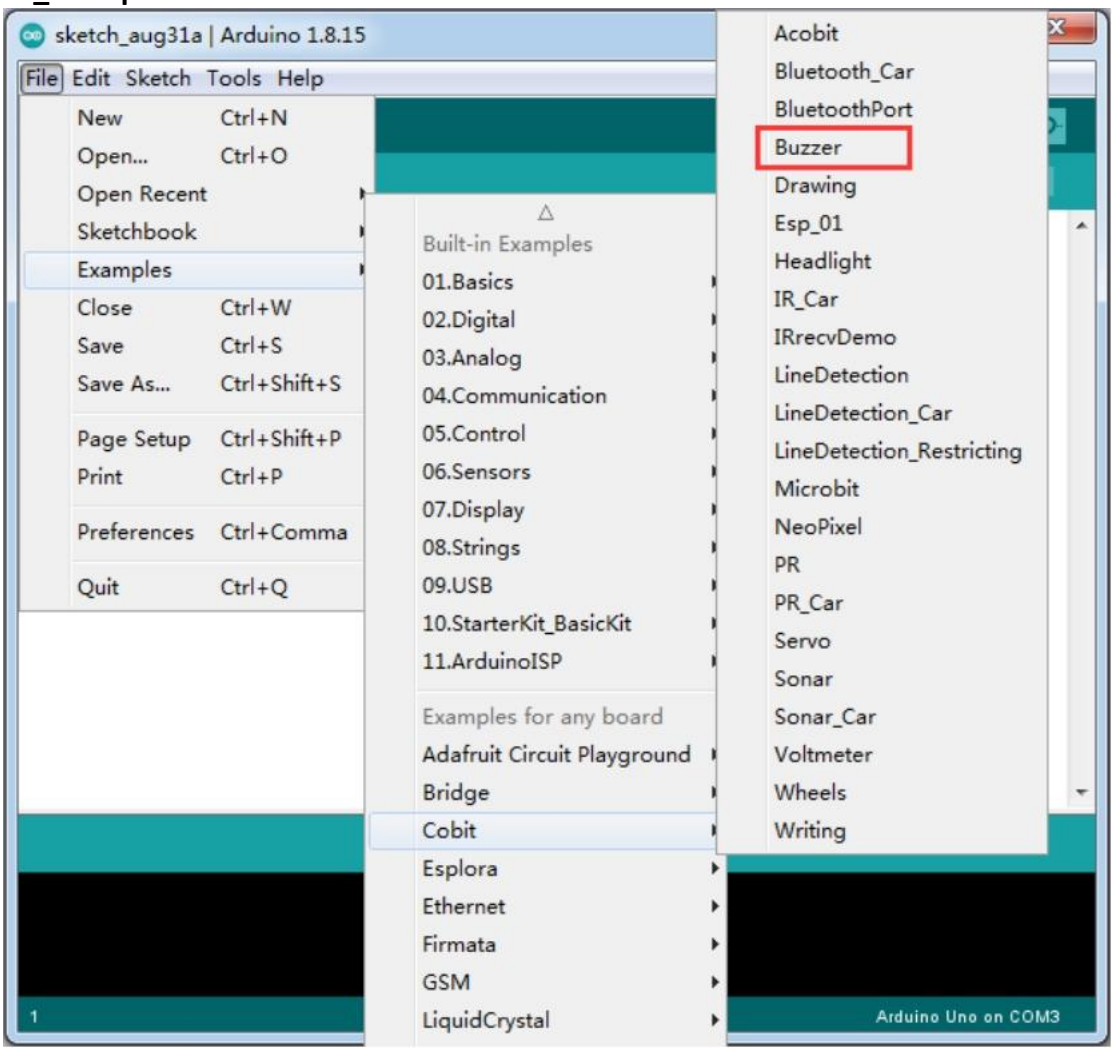
Cobit Integrats two white 5MM headlights, the "Headlight" example provides control of their on and off states. You can also control their brightness, this function can achieve the effect of breathing light.

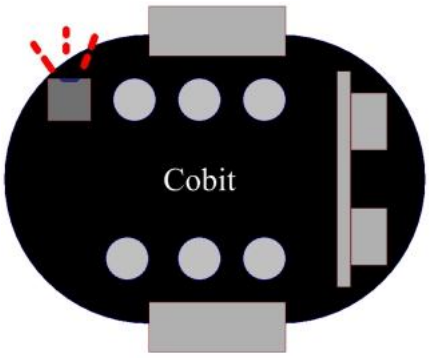


The brightness of the light is controlled by a PWM signal. PWM means pulse width modulation, that is, pulses with variable high and low time widths are periodically generated. When the high level is high, the headlights will be on, and when the low level is low, the headlights will be off. For example, the period of a pulse is 255 milliseconds (period = high level time + low level time), when the high level time width is 255 milliseconds , The low-level time width is 0, and the headlights are the brightest at this time; when the high-level time width is 0 and the low-level time width is 255 milliseconds, the headlights are the darkest at this time.

After uploading the code, the two headlights of Cobit will light up for 1 second and then turn off; then the breathing light will turn on, and then the breathing light will turn off; and keeps working like this.

2_Example of Buzzer



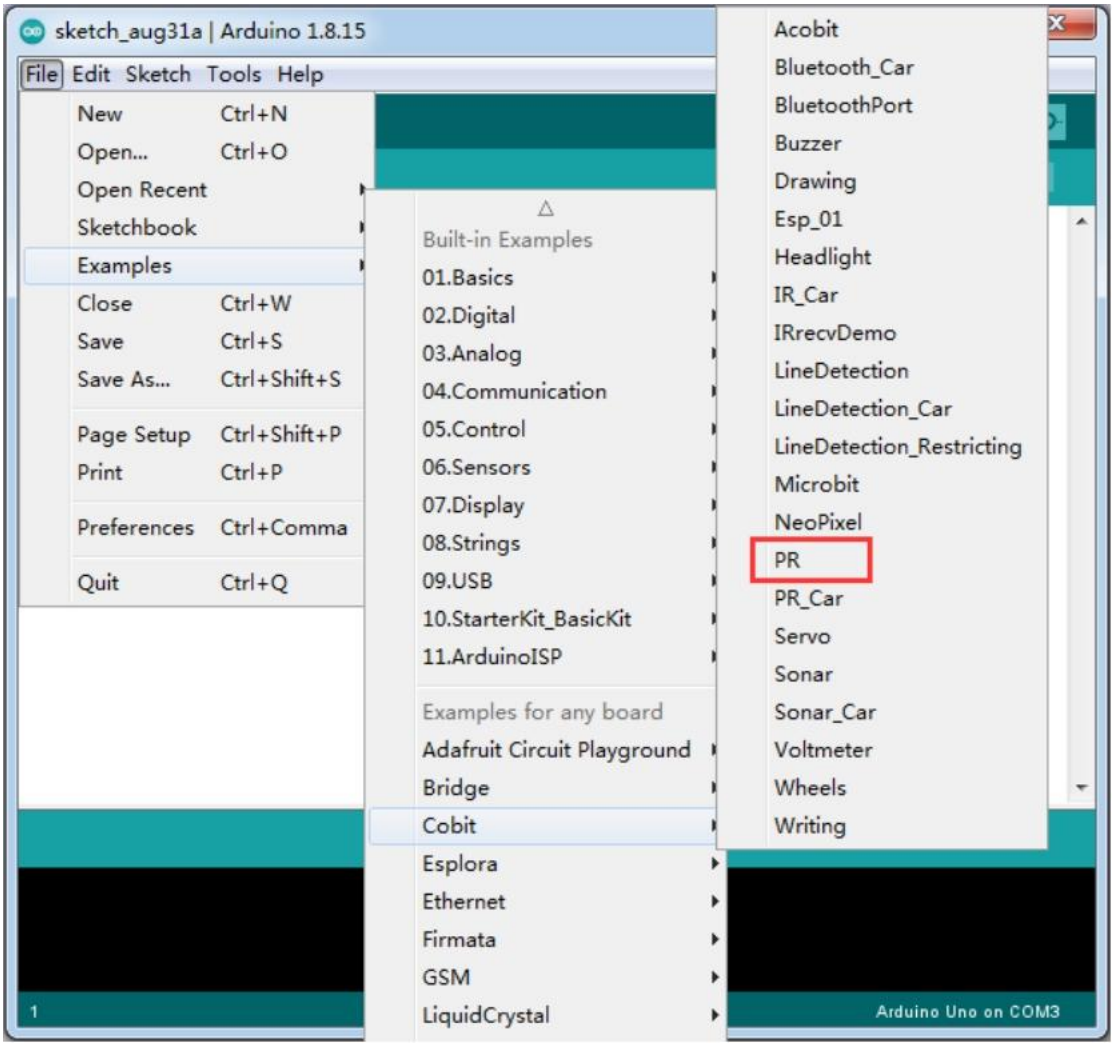


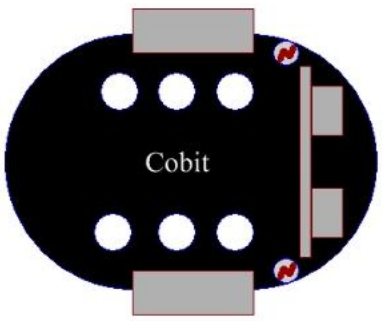
Cobit has a passive buzzer. When the passive buzzer has been supplied with direct current, the buzzer will not make sound, and sometimes it will cause the buzzer to become hot. In severe cases, the buzzer will be damaged. Please give the passive buzzer a alternating current with frequency to make it produce a sound. The "Buzzer" example provides the control of the passive buzzer on and off, and you can also

adjust its volume and sound frequency (20---1000 Hz).

After uploading the code, the buzzer will emit a sound of 20---1000 Hz, and keeps working like this.

3_Examples of Photoresistors

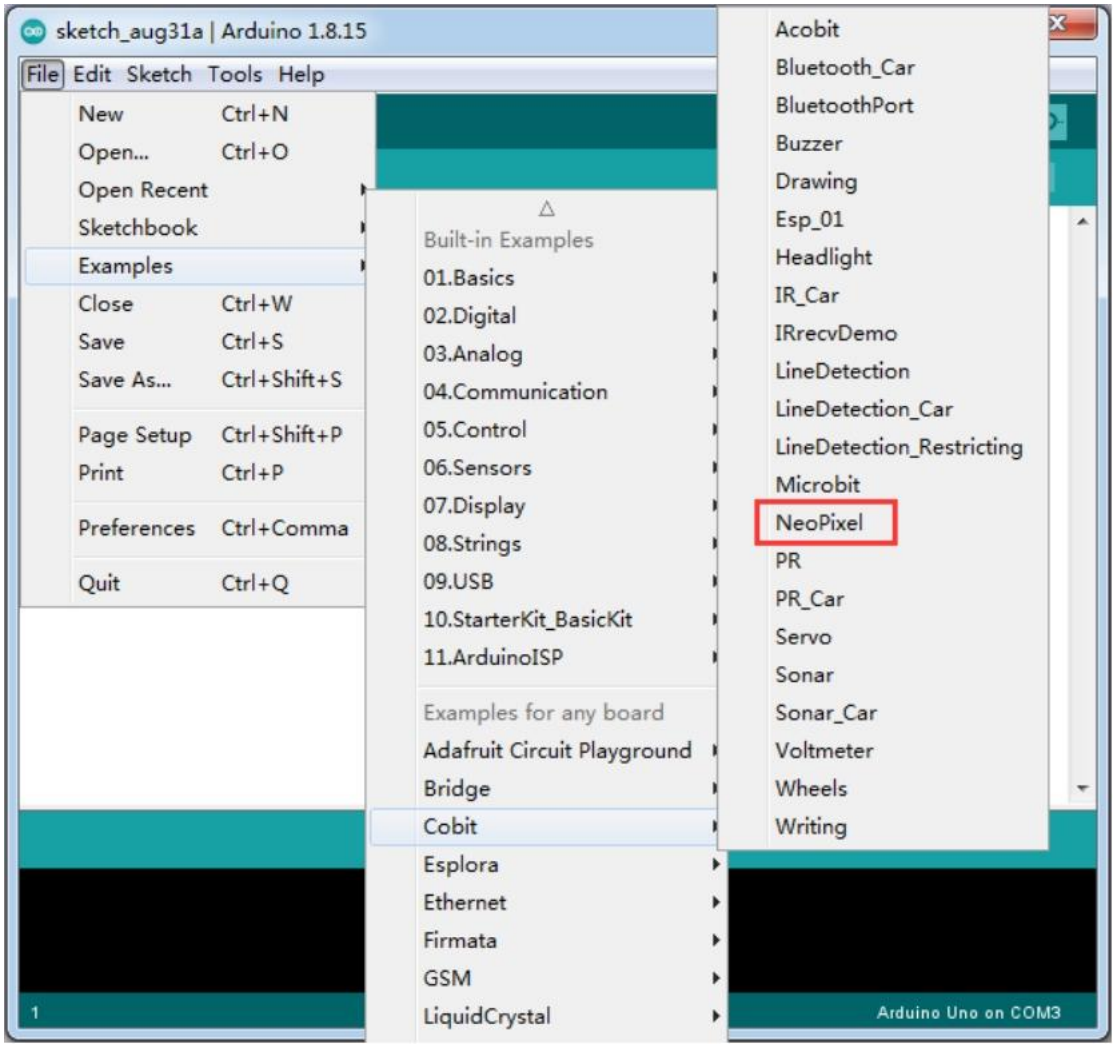




The front end of Cobit integrates two photoresistors, and uploading the "PR" example can read their analog values.

Photoresistor is a component that converts light signals into electrical signals. The stronger the luminosity, the smaller the resistance of the photoresistor. After uploading the code, open the serial monitor of the arduino IDE, set the baud rate to 9600, and the monitor will print the data from the two photoresistors.

4_RGB light example

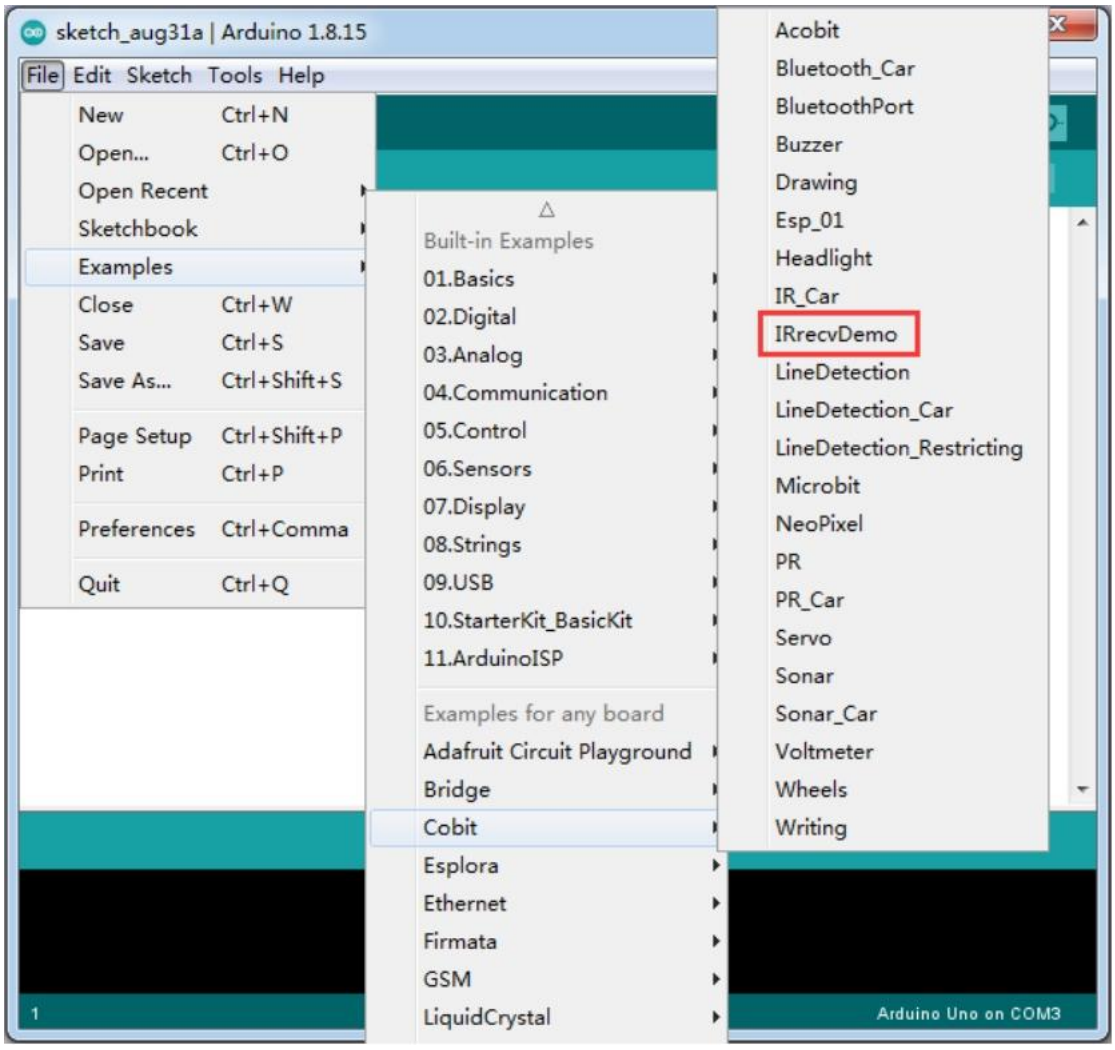


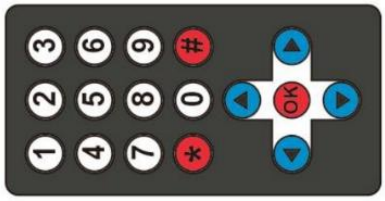
RGB

Cobit has 4 RGB LED lights, which can emit red, green, and blue lights. By controlling the brightness of the three lights, and then combining with each other, the RGB light can produce light of various colors.

After uploading the code, 4 RGB lights will turn on (white light), and then turn off.

5_Example of Infrared Receiver





There is an infrared receiving sensor on the front and rear of the Cobit robot body. This design allows Cobit to receive data more accurately, sensitively, and in more directions.

After uploading the code, open the serial monitor, set the baud rate to 9600, and then use the infrared remote control provided by us or the remote control compatible with NEC protocol to send data of the key to the Cobit, and the serial monitor will print the value of the key of the remote control.

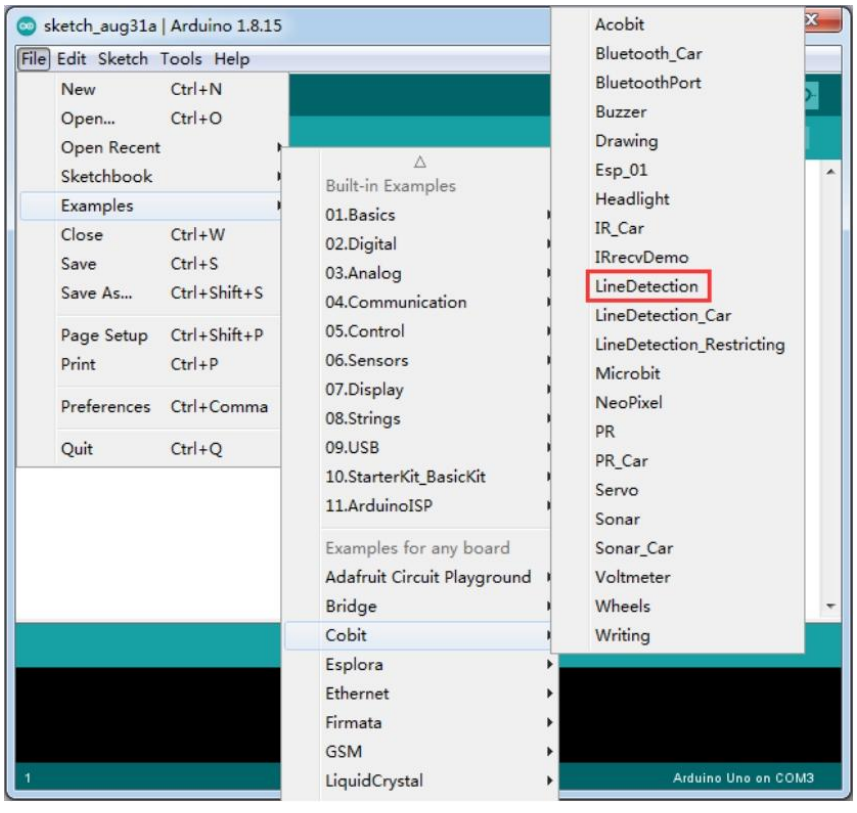
Infrared Remote Control Key Value:

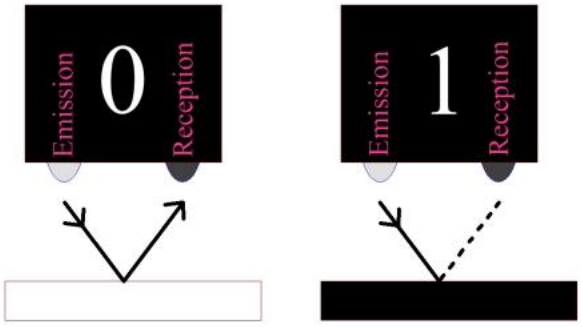
Key values for the Arduino code

0	1	2	3	4	5	6	7	8
FF9867	FFA25D	FF629D	FFE21D	FF22DD	FF02FD	FFC23D	FFE01F	FFA857
9	*	#	▲	▼	◀	▶	OK	
FF906F	FF6897	FFB04F	FF18E7	FF4AB5	FF10EF	FF5AA5	FF38C7	

Note: Use the arduino IRremote library to get the above values.

6_Example of Black and White Line Detection



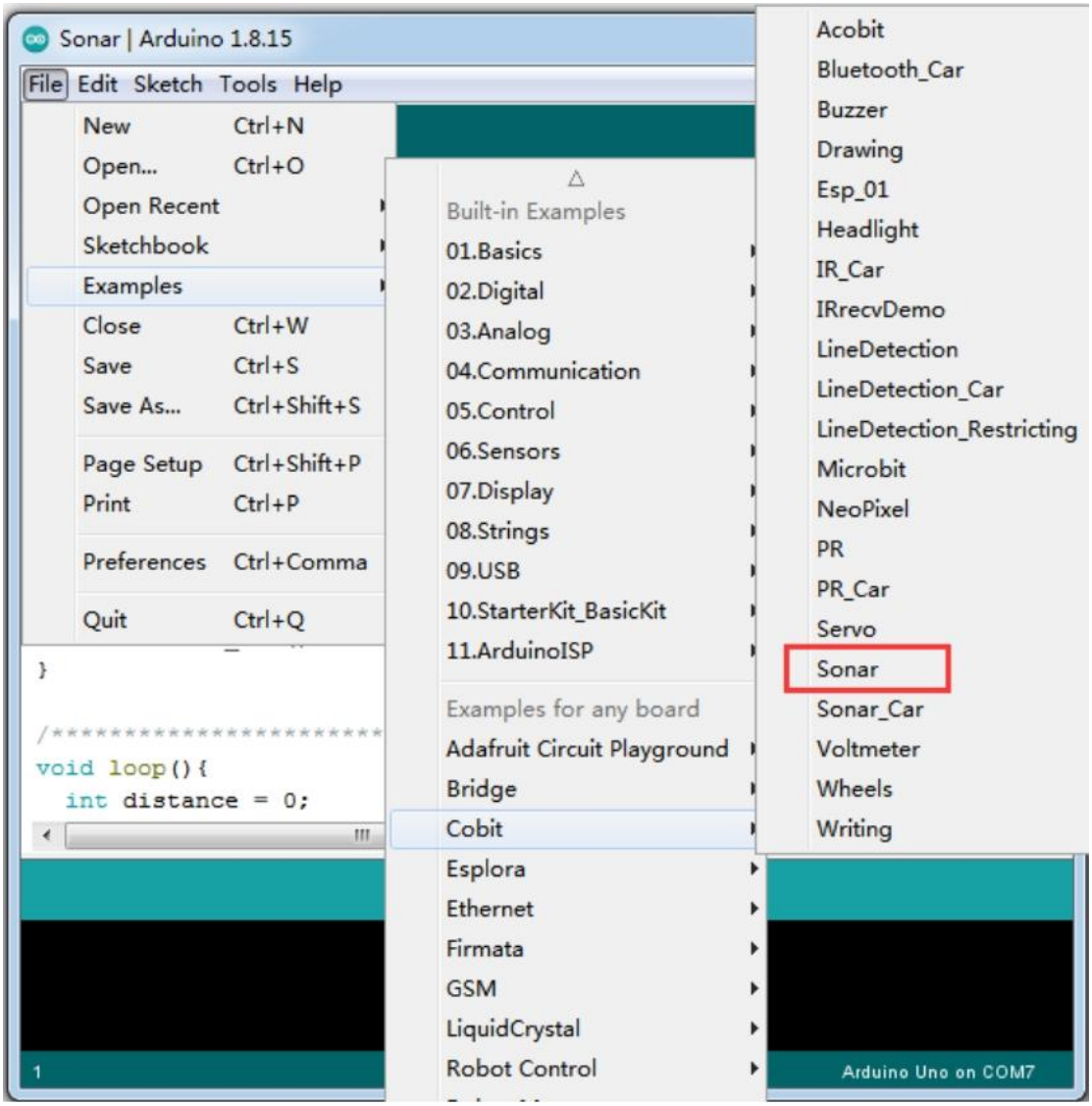


Two infrared line tracking sensors are integrated at the bottom of the Cobit, and the initial default output of it is high level. The infrared pair tube of the line tracking sensor is divided into two parts, one is the infrared transmitting end, the other is the infrared receiving end.

The black line or black object has a good absorption effect on infrared rays. When the infrared rays are emitted from the transmitting end of sensor to the black line, the infrared rays are absorbed without forming a reflected signal, the receiving end of the sensor does not receive a signal, the line tracking sensor outputs a high level (1); The white line or white objest has no absorbing effect on infrared rays, the infrared signal from the transmitting end of the sensor will be reflected back to the receiving end, and the sensor outputs low level (0).

After uploading the code, open the serial monitor, set the baud rate to 9600, use white paper to approach the two infrared sensors. If the left infrared sensor detects detects infrared rays that are reflected back by the white paper, the serial monitor will print "left detected". If the left infrared sensor detects nfrared rays that are reflected back by the white paper, the serial monitor will print "right detected". If the left and right infrared sensors both detect the nfrared rays that are reflected back by the white paper, the serial monitor will print "left detected" and "right detected".

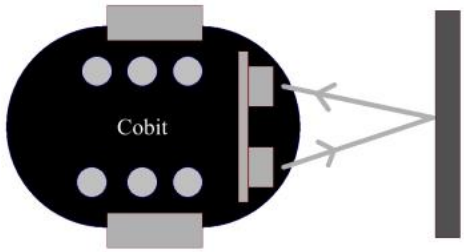
7_Example of Ultrasonic Sensor



Ultrasonic sensor is a mechanical wave with an extremely short wavelength. The wavelength in the air is generally shorter than 2 cm (centimeters). It must rely on the medium to propagate and cannot exist in a vacuum (such as space). But because of its short wavelength, it is easily lost and scattered in the air. It is not as far as audible sound and infrasound waves. However, the short wavelength is easier to obtain

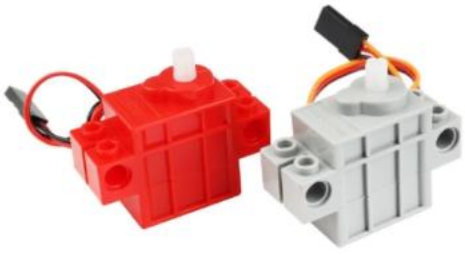
anisotropic sound energy, which can be used for cleaning, ranging, etc.

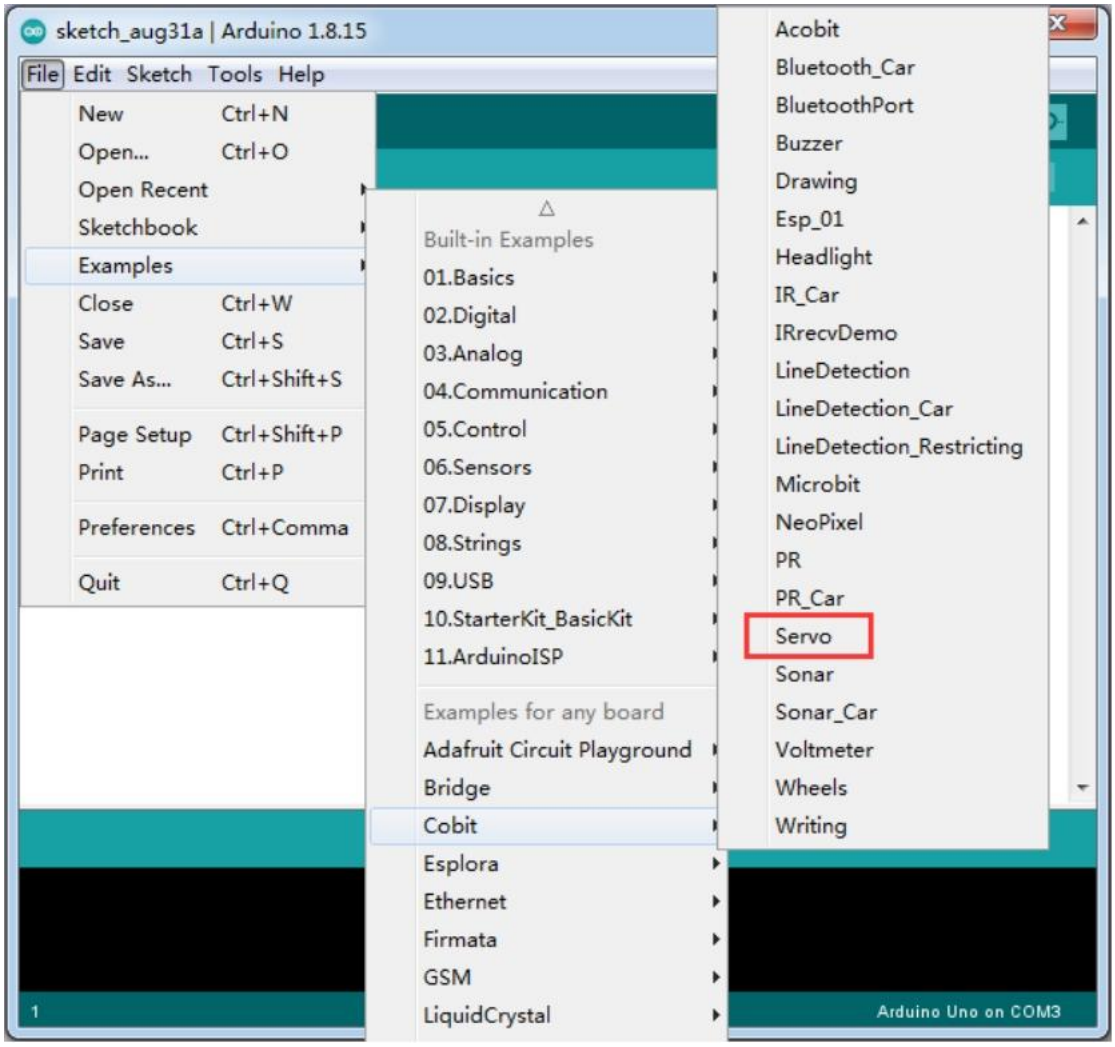
An interface for the ultrasonic sensor is reserved on the Cobit, which can be connected to the HC-RS04 ultrasonic module for distance measurement. Cobit is equipped with the HC-RS04 ultrasonic module by default, and integrates the API of the module in the Cobit library. The measurement data of the module can be read by using the API, which greatly reduces the difficulty of use. The maximum measuring distance of it is 255cm, and the accuracy is +/-1CM.



Since ultrasonic sensors work much like sonar sensors, we named this example "Sonar". Select the Sonar example in the "Cobit" menu. After uploading the code, open the serial monitor, set the baud rate to 9600, place an obstacle in front of the cobit, and the serial port will print out the distance between the cobit and the obstacle (Unit : cm).

8_Example of the Servo





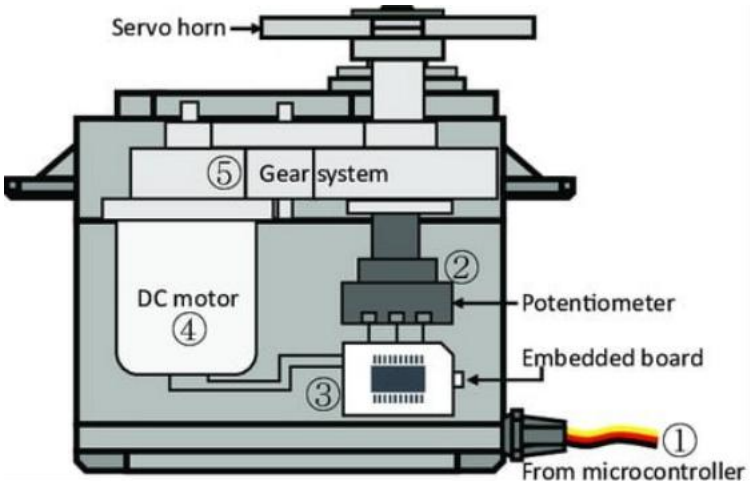
There are 3 ports for the servo reserved on the Cobit, which can be connected with Lego or SG90 servos. Cobit comes with a Lego servo, and integrates the API of the module into the Cobit library. The API can be used to drive the servo simply and quickly, which is easier to use.

Notice:

The power from the USB is not enough to drive the servos, please make sure that the two 18650 batteries have been installed and the power switch next to the battery case is "ON" status.

After uploading the code, connect the servo to any servo interface, and the servo will swing back and forth between 0---180 degrees.

About Servo:



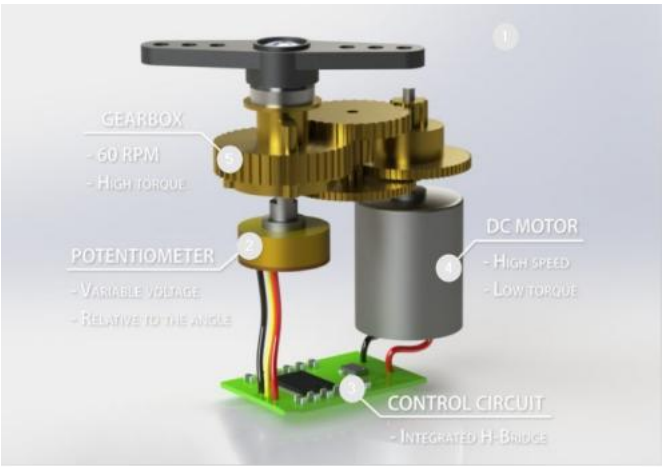
①Power negative (black), power positive (red), signal (yellow);

②Potentiometer: can measure the position of the output shaft, which belongs to the feedback part of the entire servo mechanism;

③ Internal controller: processing signals from external control, driving motors and processing feedback position signals are the core of the entire servo mechanism;

④Motor: as an actuator, how much speed, torque, and position are output by it;

⑤Transmission mechanism / servo system: This mechanism scales the stroke of the motor output to the angle of the final output according to a certain transmission ratio.



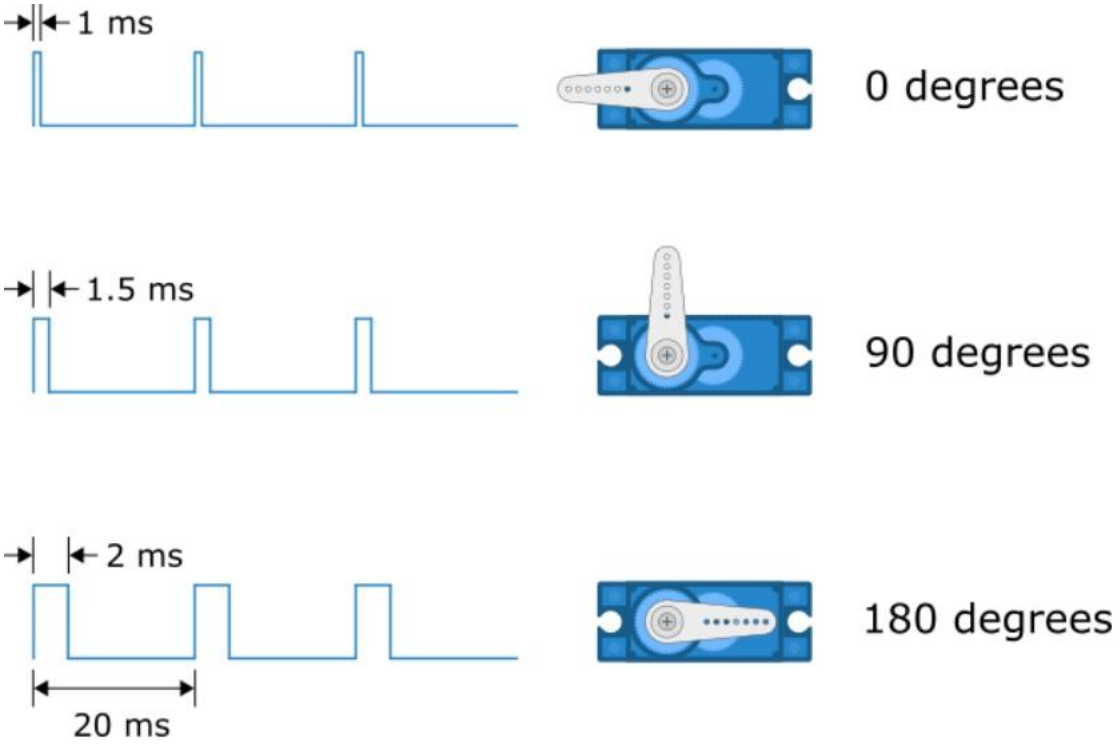
The output of the servo is controlled by sending a PWM signal to the signal line of the servo. The period and duty ratio of the PWM are controllable, so the duty ratio of the PWM pulse directly determines the position of the output shaft.

Example:

When the pulse width is 1ms, the output shaft of the servo will move to the smallest position (0 degrees);

When the pulse width is 1.5ms, the output shaft of the servo will move to the middle position (90 degrees);

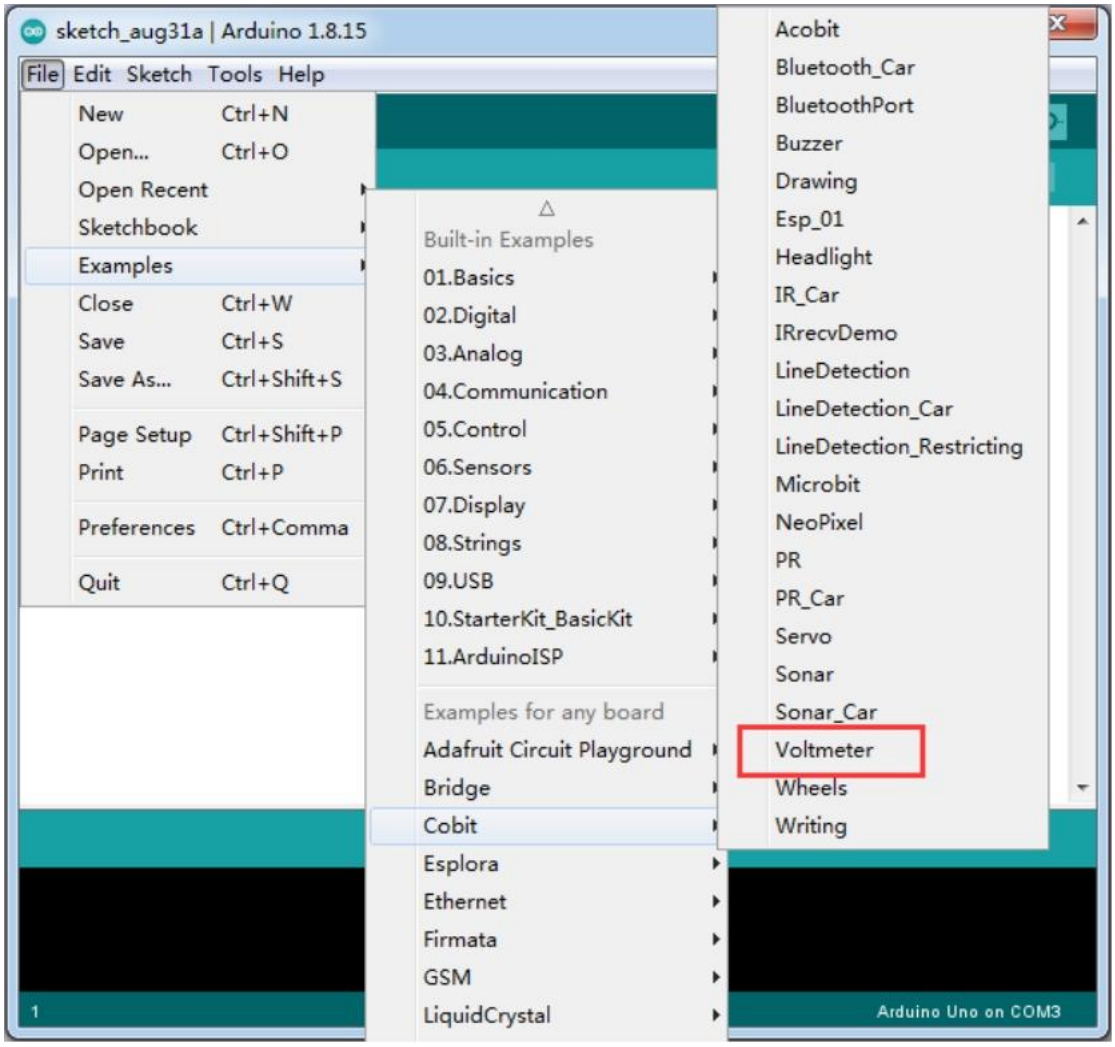
When the pulse width is 2ms, the output shaft of the servo will move to the smallest position (180degrees).



9_Example of Voltmeter

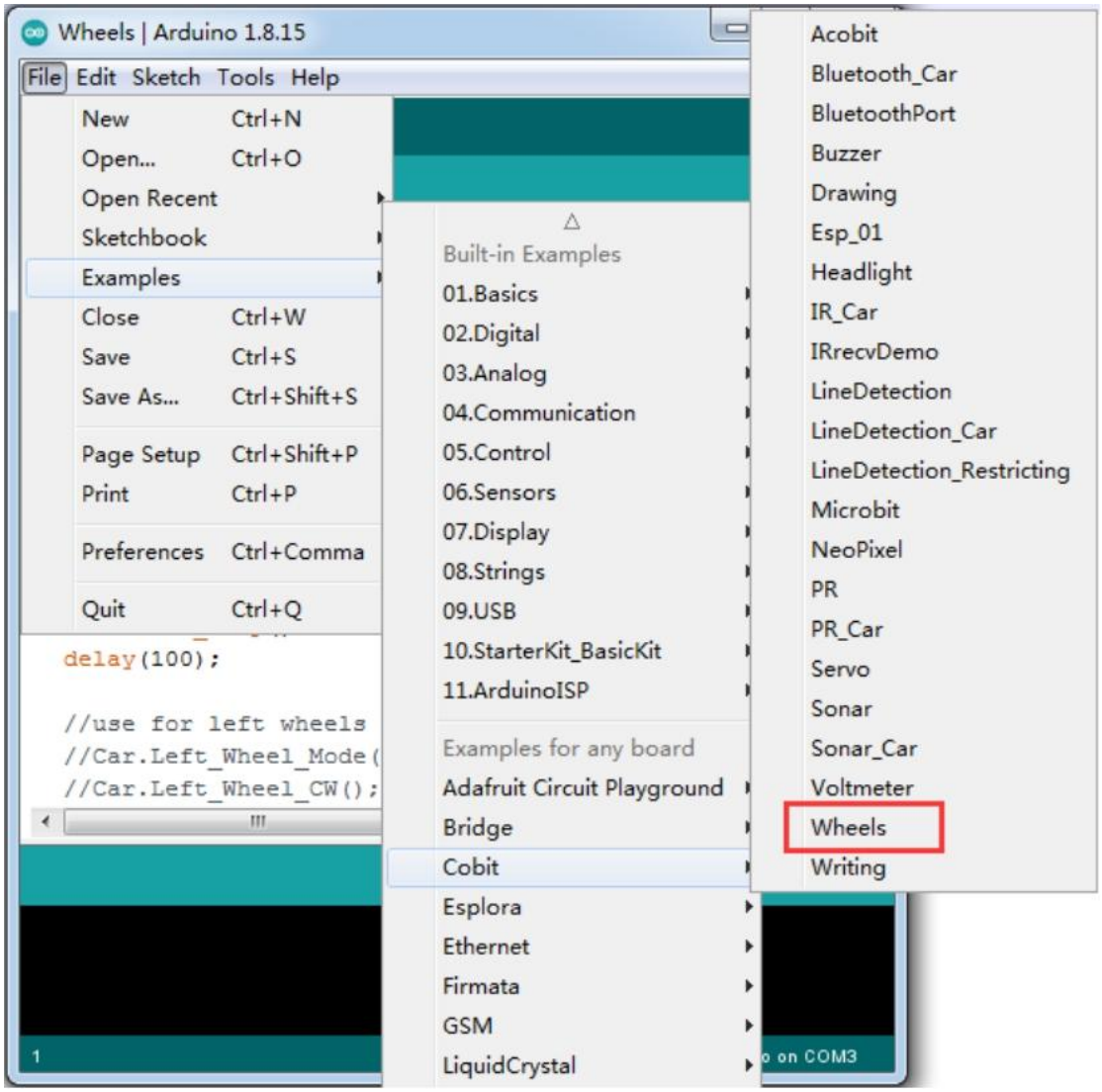
A voltmeter is designed on Cobit to read the voltage of the battery. The API in the Cobit library can be used to quickly read the battery voltage and voltage percentage. The voltage percentage is a mapping value:

Range of Voltage	Mapped Voltage Percentage
6.5V---8.4V	0---100%



After uploading the code, put two 18650 lithium batteries into the battery box of the Cobit, turn the power switch to the "ON" state, open the serial port monitor, set the baud rate to 9600, and the serial port will printed the value and voltage percentage of two 18650 lithium batteries.

10_Example of Wheels

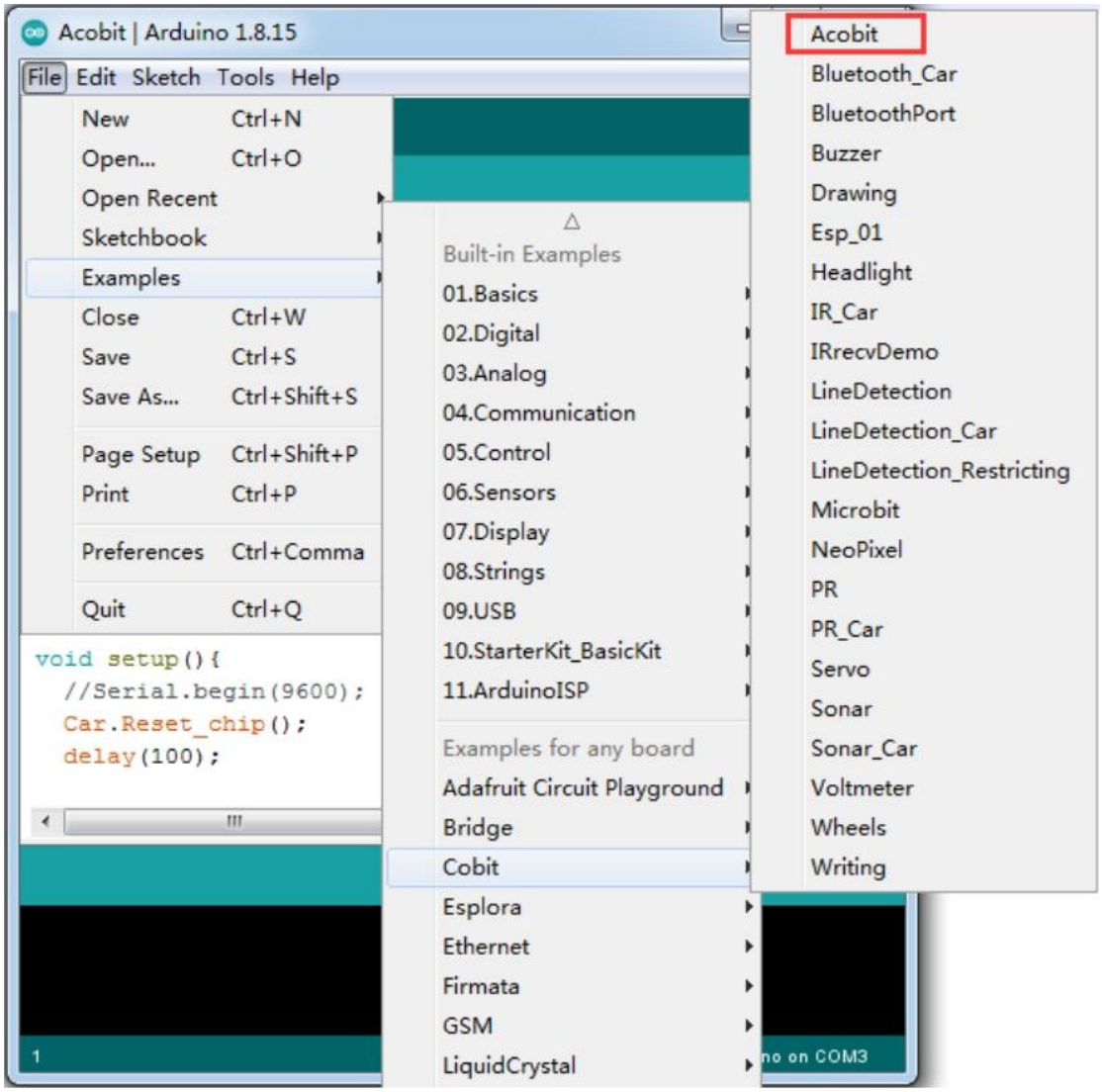


There is a wheel with a diameter of 65mm on the left and right sides of the Cobit. Two stepper motors are used as the power of the wheel. The rotation angle and the rotation arc of the wheel can be controlled. The rotation angle accuracy is +/-0.9 degrees, and the rotation speed is divided into 0-- -4 gears (0 = 0rpm, 1 = 15rpm, 2 = 30rpm, 3 = 60rpm, 4 = 120rpm).

We have compiled the complex code to drive the stepper motor into a library file, and reserve a wealth of AIP interfaces. Using AIP interfaces, you can implement various applications very simply and efficiently. For detailed usage, please refer to the sample code of "Wheels", there are comments after the " //" symbol.

After uploading the code, the two motors on the left and right of Cobit first rotate 360 degrees clockwise, and then rotate counterclockwise for 3 seconds, and then keep repeating this movement.

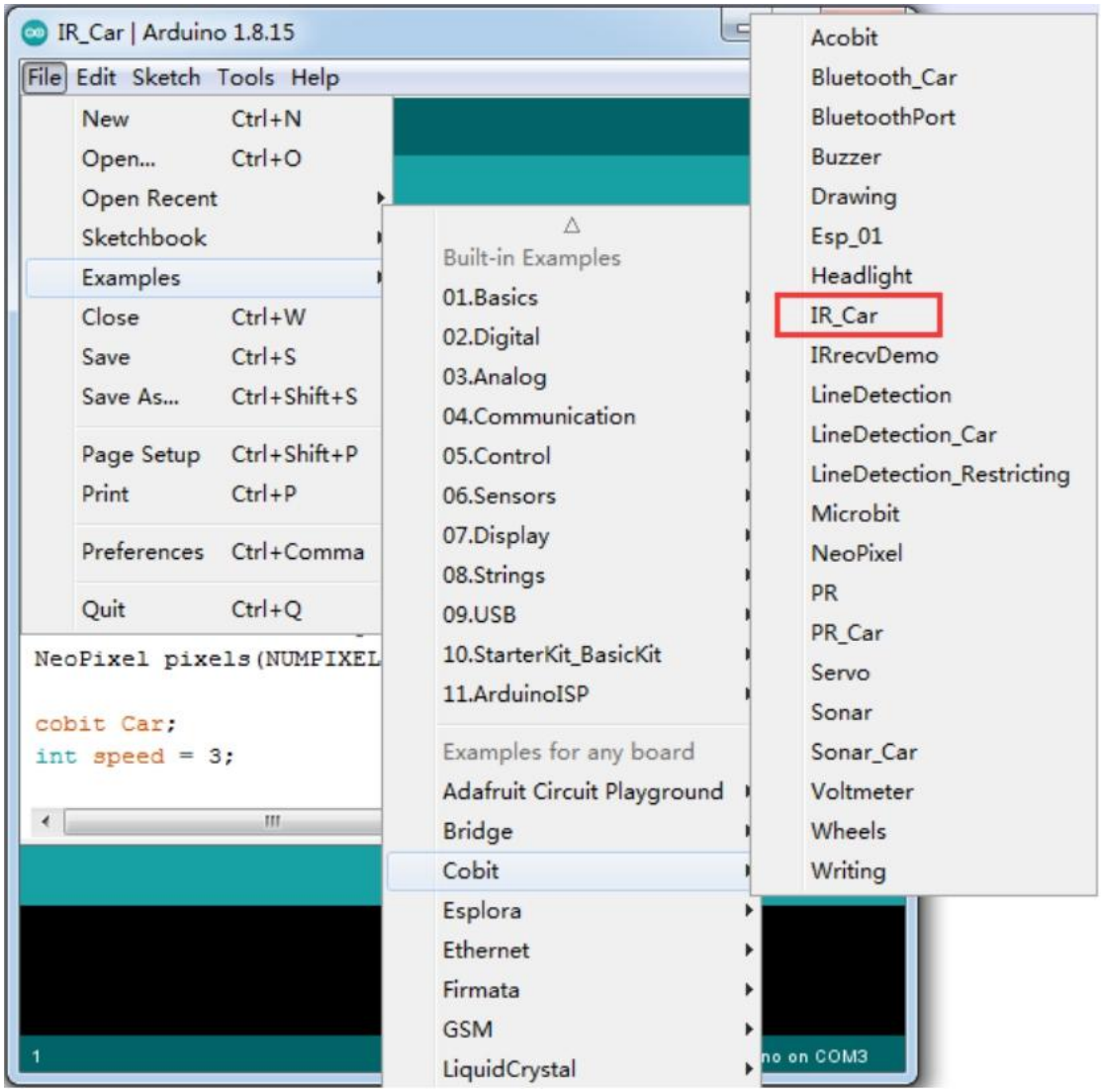
11_Basic operation of Cobit

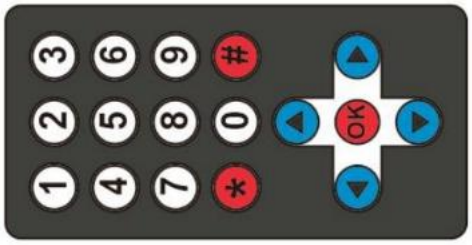


This example includes the basic driving control example of Cobit, such as the driving direction of the car and their corresponding delay functions.

After uploading the code, the Cobit robot will turn left 360 degrees, turn right 360 degrees, and then goes 200mm forward, next reverse 200mm; then turn left continuously for 2 seconds, turn right continuously for 2 seconds, and goes forward for 1 second, then back for 1 second, keep moving like this.

12_Infrared remote control car



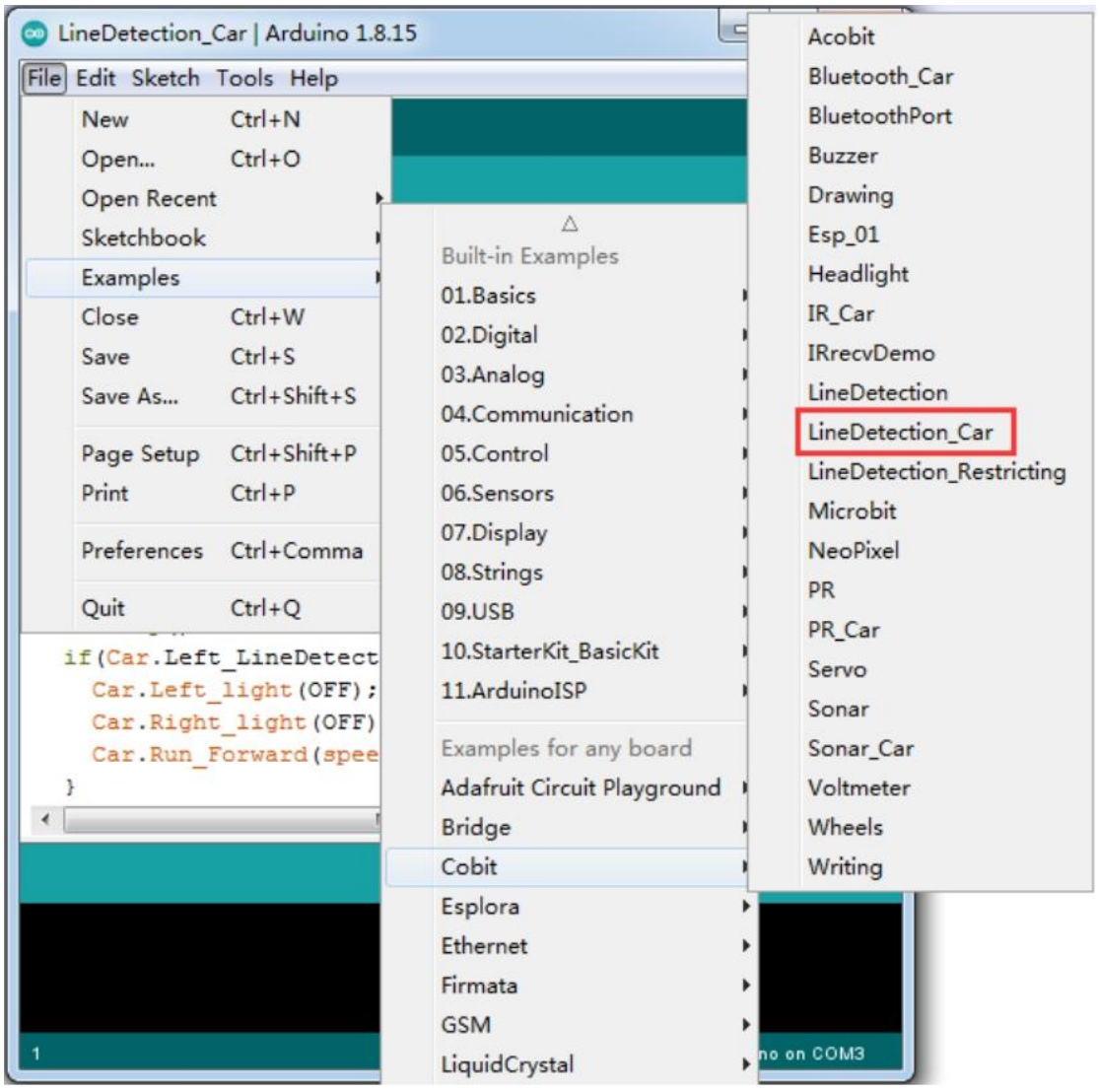


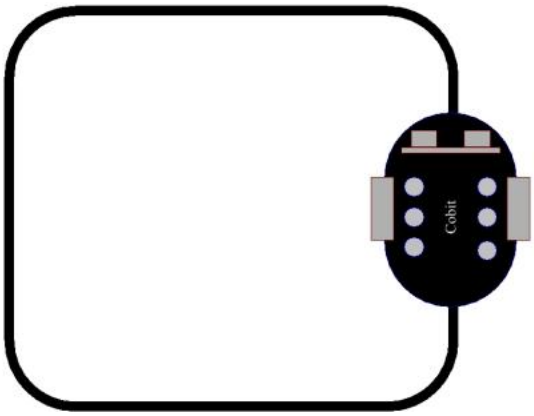
Cobit comes with an infrared remote control, and each button of the remote control has a different key value. We give each button a different function, and then send the key value to Cobit through the remote control, when Cobit obtains the key value, it will implement the function corresponding to the key value.

Arduino Key value and function:

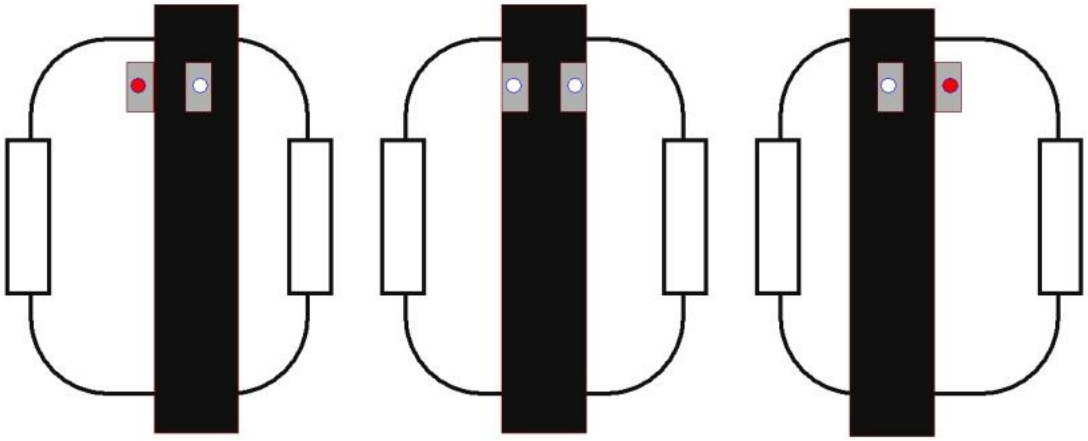
Key	0	1	2	3	4	5	6	7	8
Key Value	FF9867	FFA25D	FF629D	FFE21D	FF22DD	FF02FD	FFC23D	FFE01F	FFA857
Function	null	Turn on the buzzer	Turn off the buzzer	Turn on the RGB LED	Turn off the RGB LED	Turn on the Car light	Turn off the Car light	null	null
Key	9	*	#	▲	▼	◀	▶	OK	
Key Value	FF906F	FF6897	FFB04F	FF18E7	FF4AB5	FF10EF	FF5AA5	FF38C7	
Function	null	null	null	forward	back	turn left	turn right	stop	

13_Line Detection Car



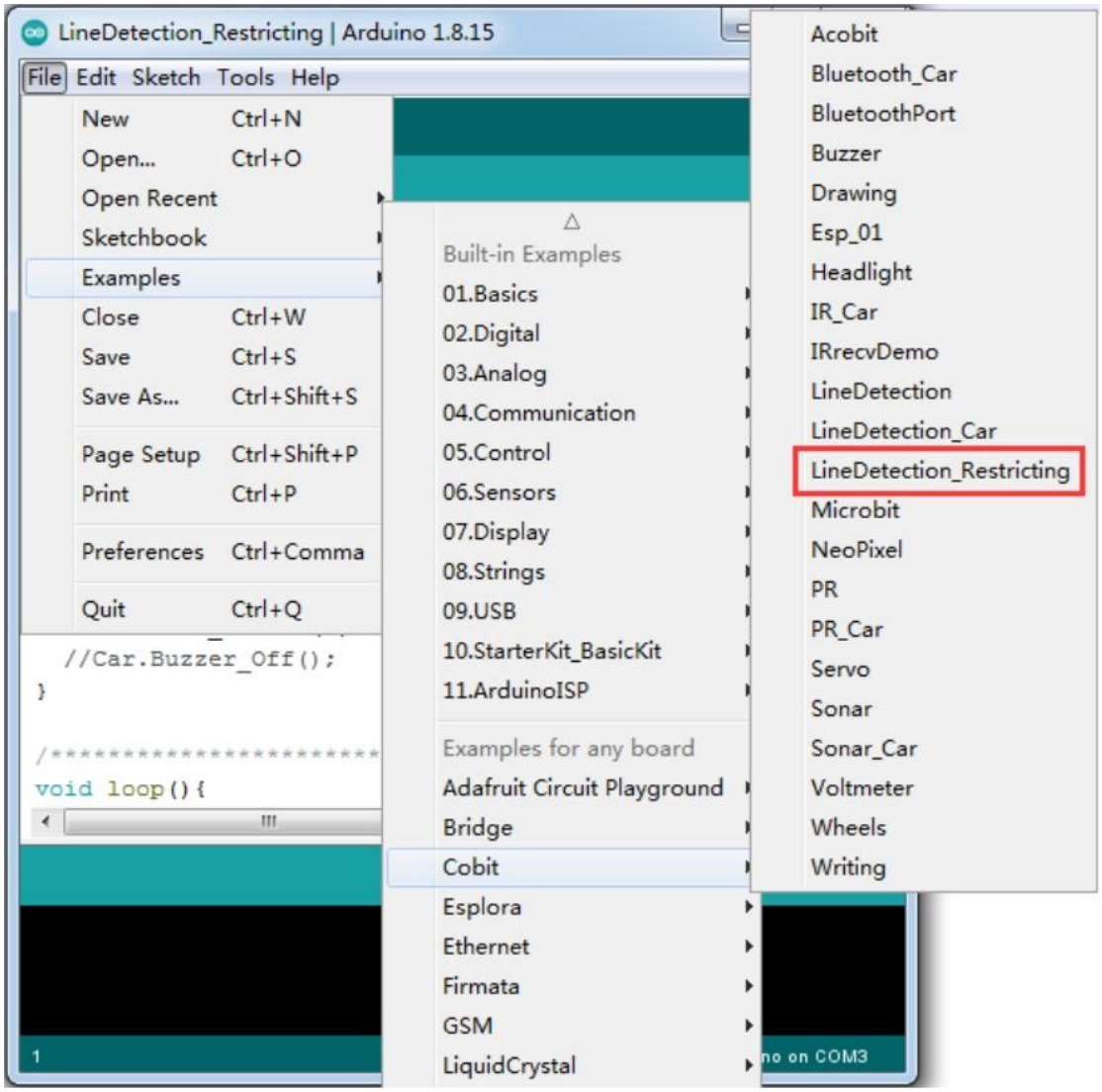


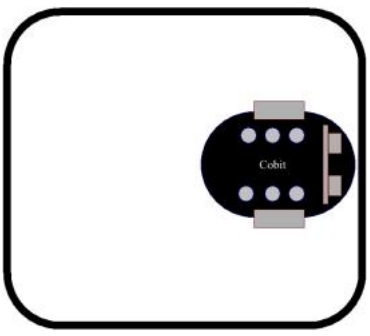
Two infrared line tracking sensors are integrated at the bottom of the Cobit, makes the cobit robot can walk along the black line. Place the cobit on the white paper with the black track that came with the package. Black objects have the effect of absorbing infrared rays, and white objects have the effect of reflecting infrared rays. Use this function you can keep the car walk along the black line.



is “0”, is “1”.

14_Line Detection Restricting

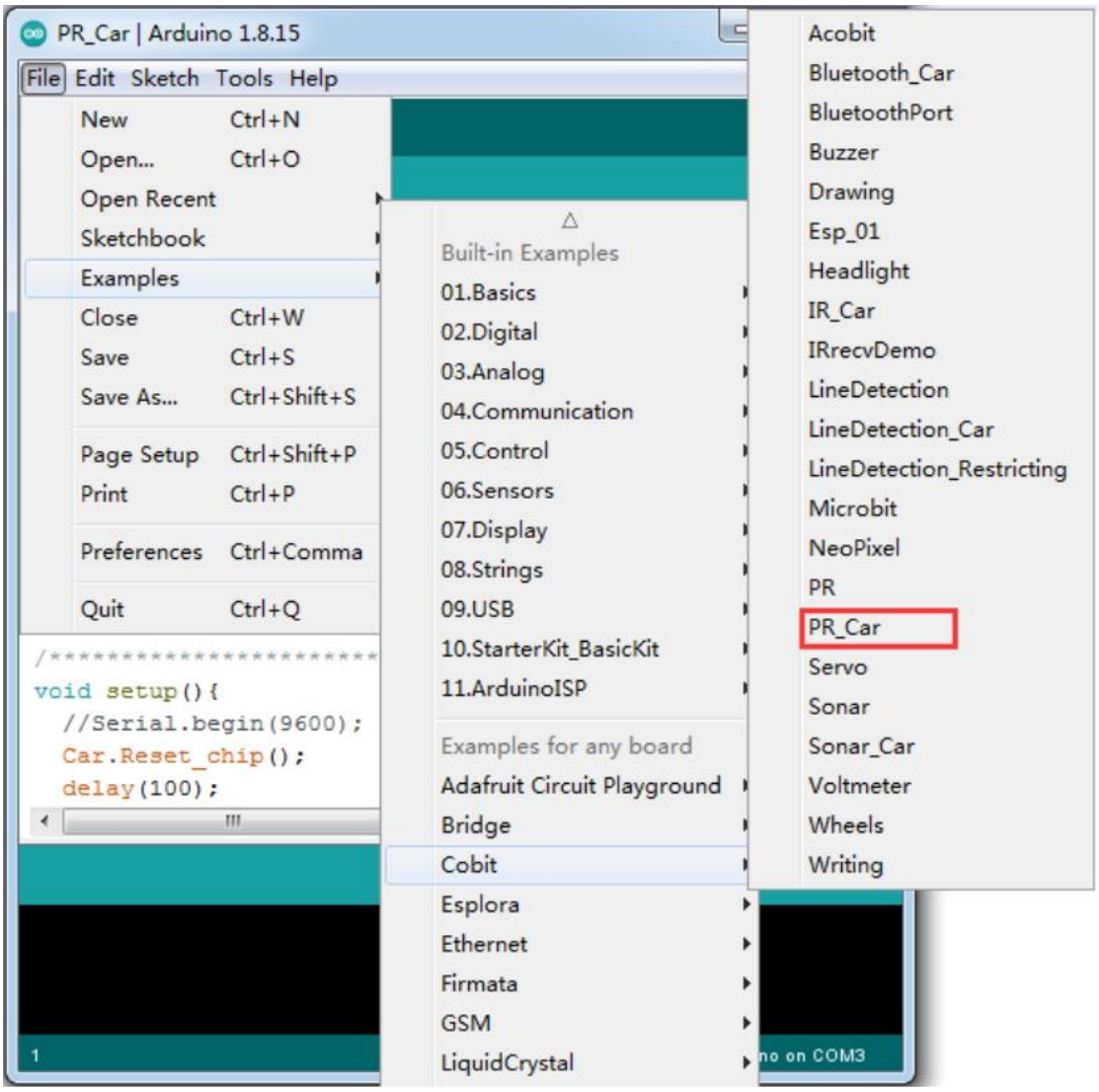


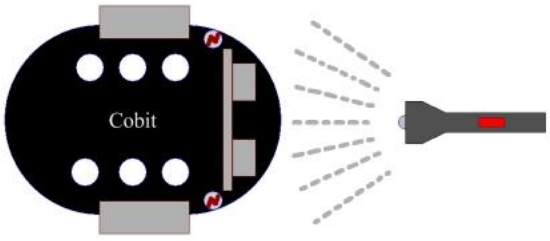


Using the cobit robot's ability to detect black lines and white areas, we can program the cobit robot to confine itself to the black circle. After uploading the code, place the cobit robot on the white paper with the black circle that came with the package. When the car is moving inside the black cycle, the two infrared line tracking sensors

keep receiving the reflected signal. When the car hits the black line, the line-following sensor does not form a reflection signal, and the car immediately stop, and then turns direction to keep inside the black cycle.

15_PR Car

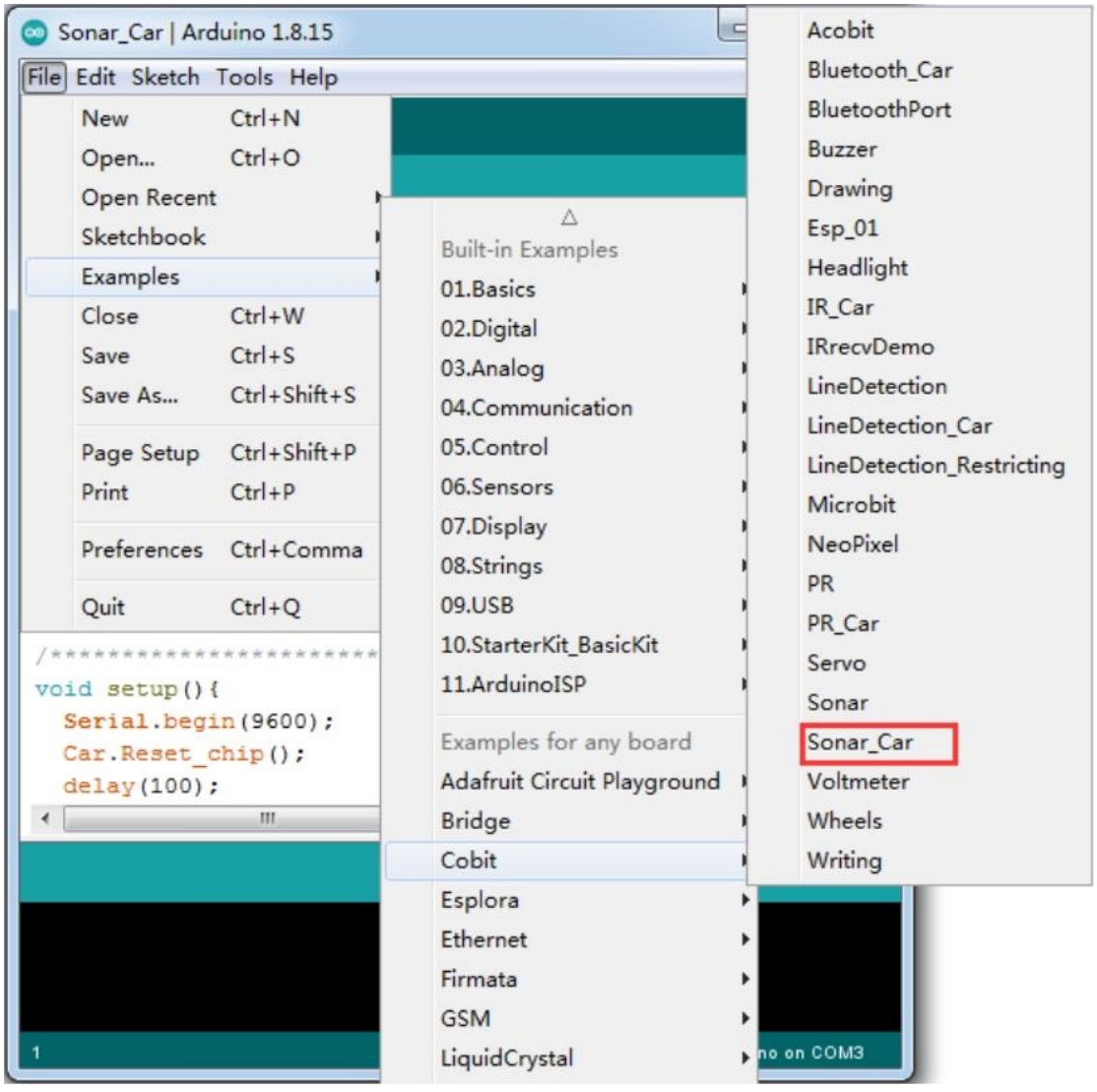


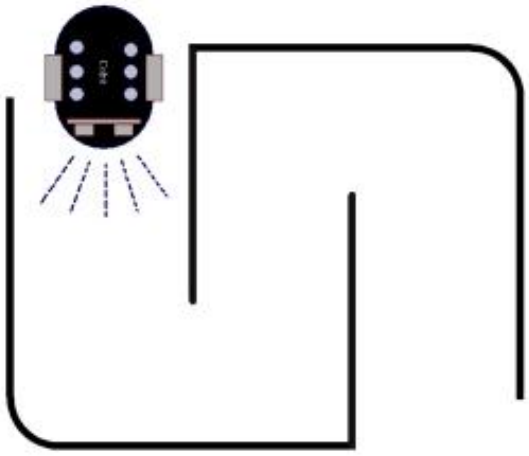


A photoresistor sensor is integrated on the left and right sides of the Cobit, which can be used to determine the brightness of the light on both sides of the Cobit. We set the program to make the Cobit move to make the same brightness obtained by the photoresistor sensors on both sides. This way of working enables the Cobit to face the light source and follow it.

Note: In order to avoid the influence of other light, please carry out this experiment in a dark place.

16_Sonar Car(Ultrasonic Module)





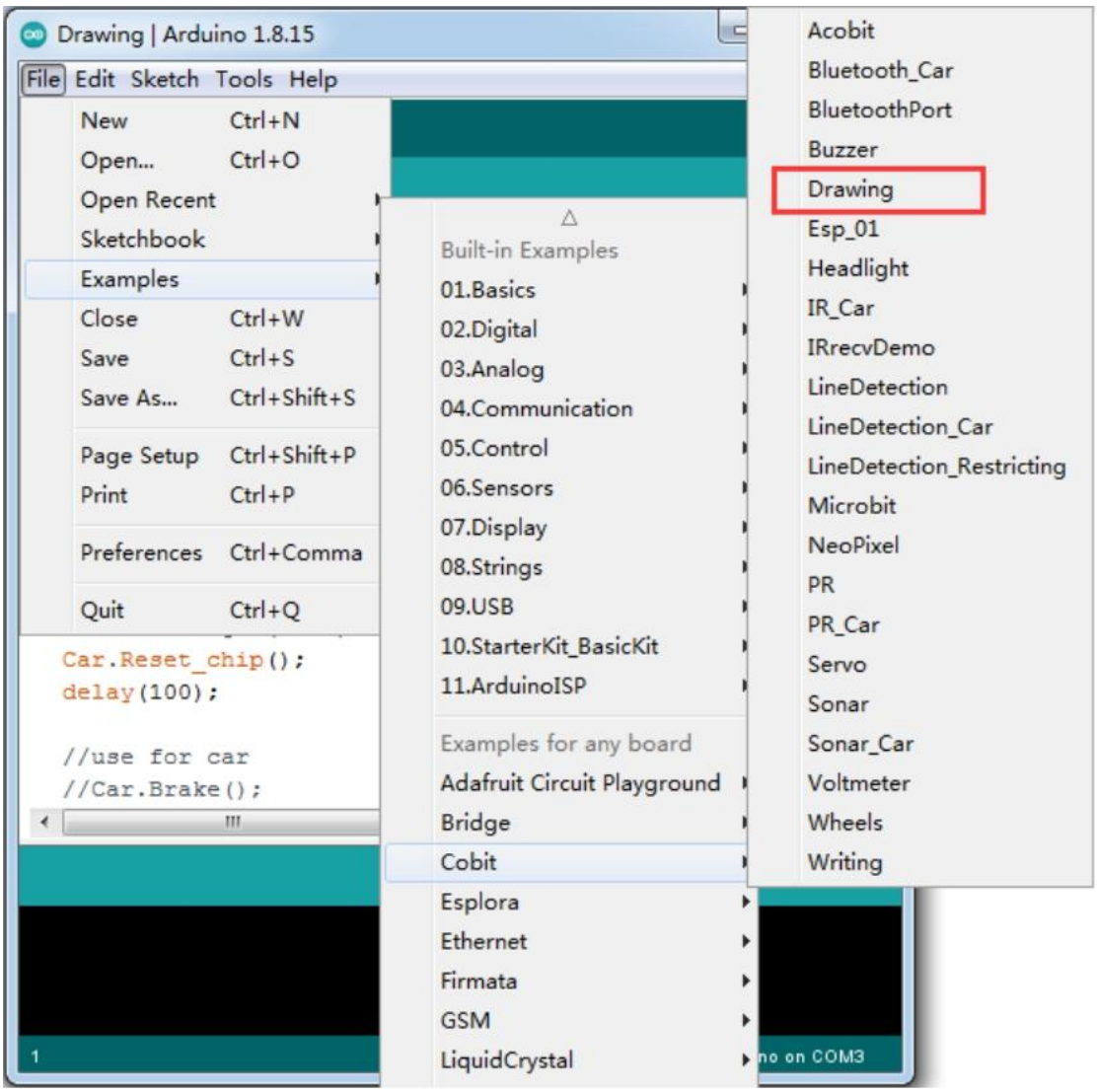
The Cobit robot comes with a Ultrasonic Module. As long as the module is inserted into the Ultrasonic Module interface of the Cobit, the Cobit has the function of measuring distance.

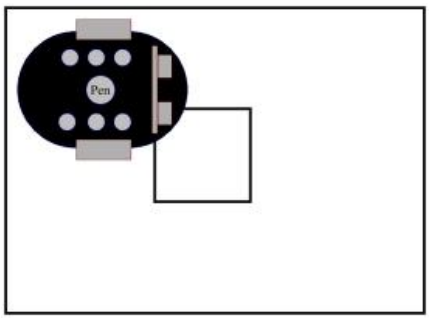
Since ultrasonic sensors work much like sonar sensors, we named this example "Sonar Car".

The function of this code is to use the ultrasonic sensor on the front end of the Cobit to measure the distance of the obstacle in front. If the distance is less than the distance set by the program, the car will stop immediately, and then turn to measure the distance of the obstacles on the left and right sides. If the distance between the left and right obstacles is less than the programmed distance, the robot will turn around. f the distance between the left and right obstacles is greater than the programmed distance, the robot will move to the side farther from the obstacle.

17_Drawing Car

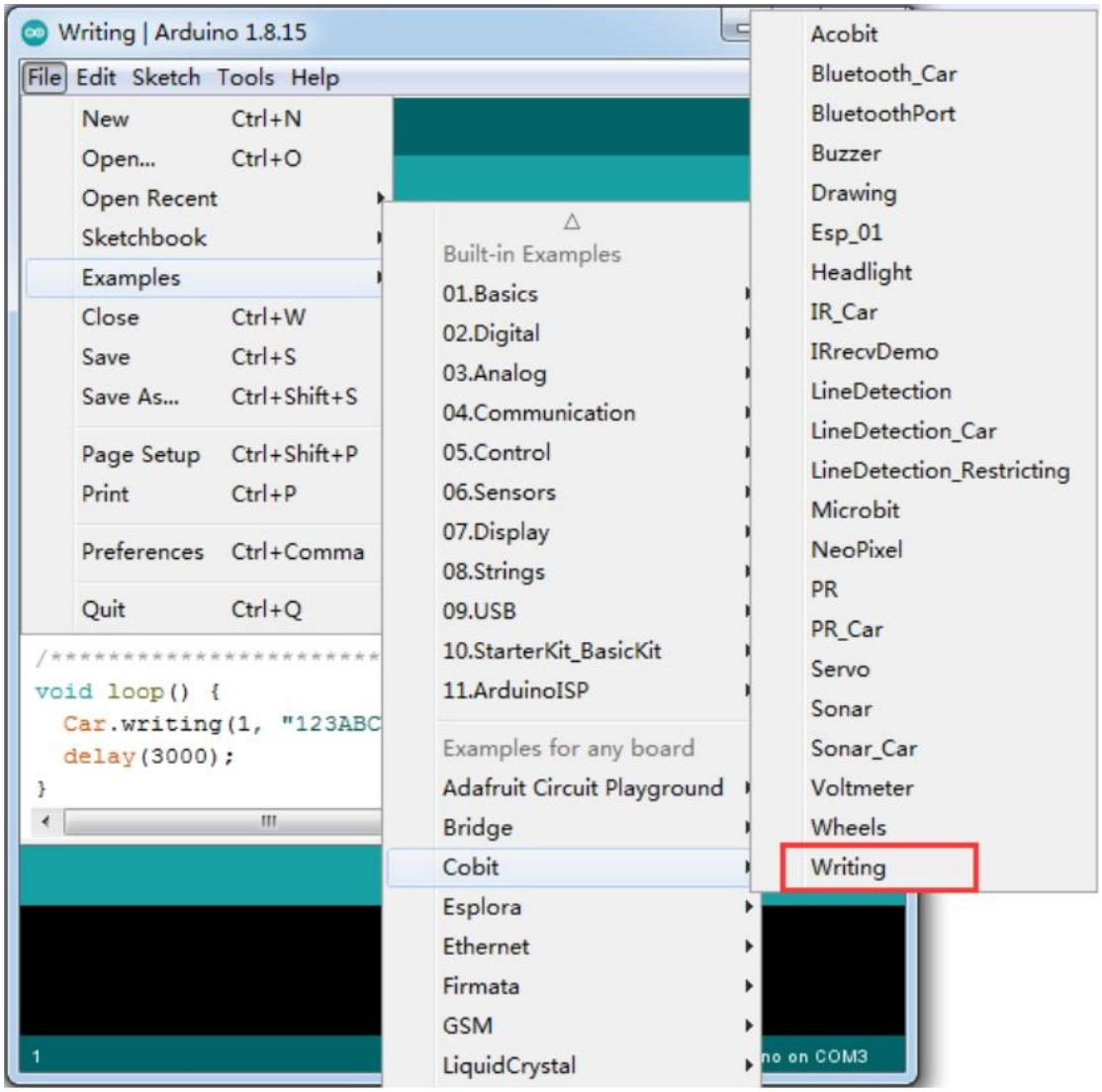
This function of the Cobit is completed in conjunction with Lego accessories. There are standard Lego mounting holes reserved on the car body. As long as the pen holder structure is built according to the assembly instructions, the Cobit can paint then.





Burn the code into the Cobit and place the Cobit on the upper left corner of the A4 paper. The cart will draw a rectangle on the paper. If you need more drawing shapes, you can refer to the code programming.

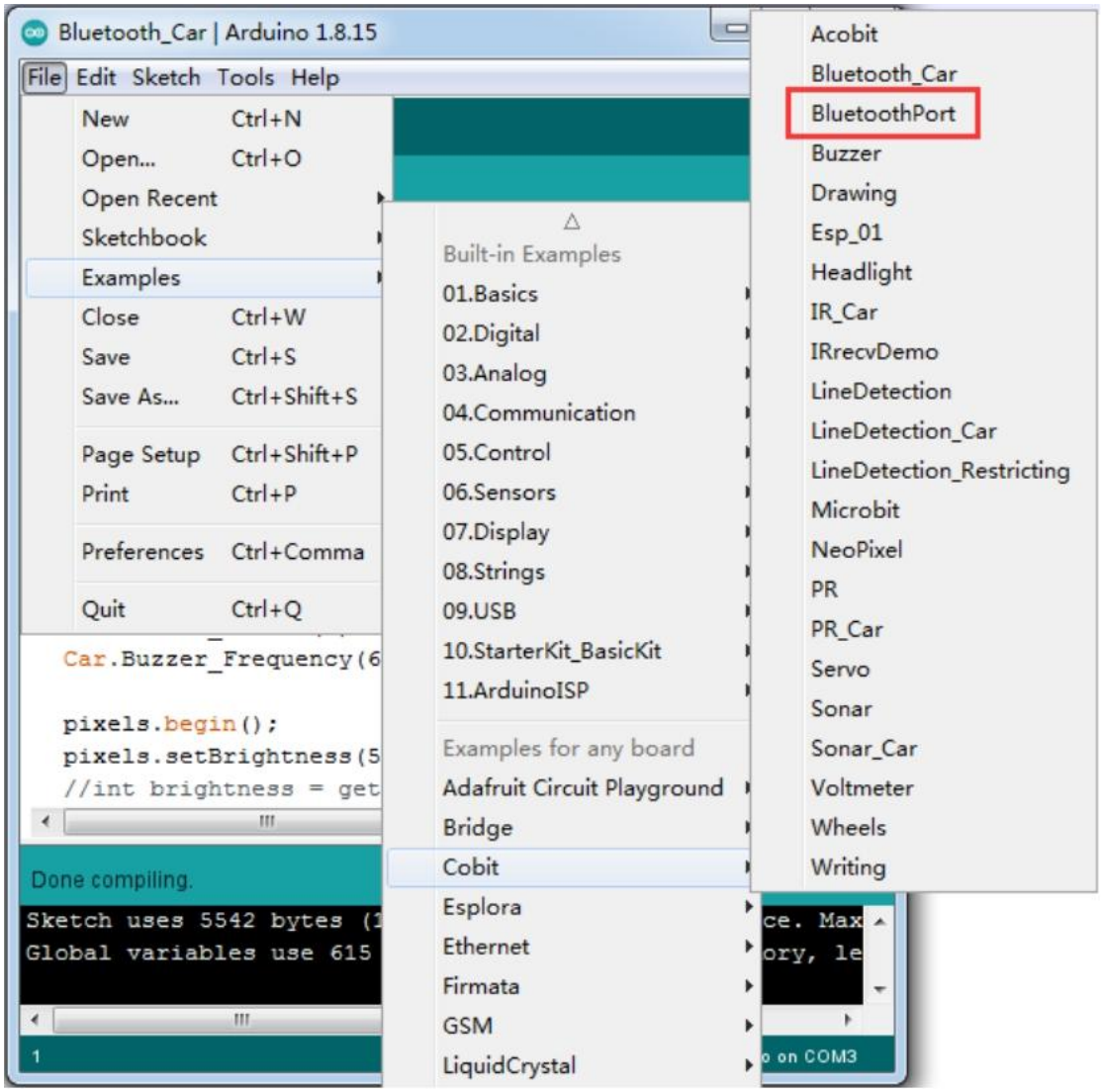
18_Writing Car



You need to build the Lego pen holder structure according to the assembly instructions to enable the Cobit to have the writing function.

Burn the code into the Cobit, stitch together 3 sheets of A4 paper, put the Cobit on the upper left corner of the first A4 paper, the car will write "123ABCabc" on the paper, you can also change "123ABCabc" to whatever number and text you want in the code.

19_Bluetooth Port Example



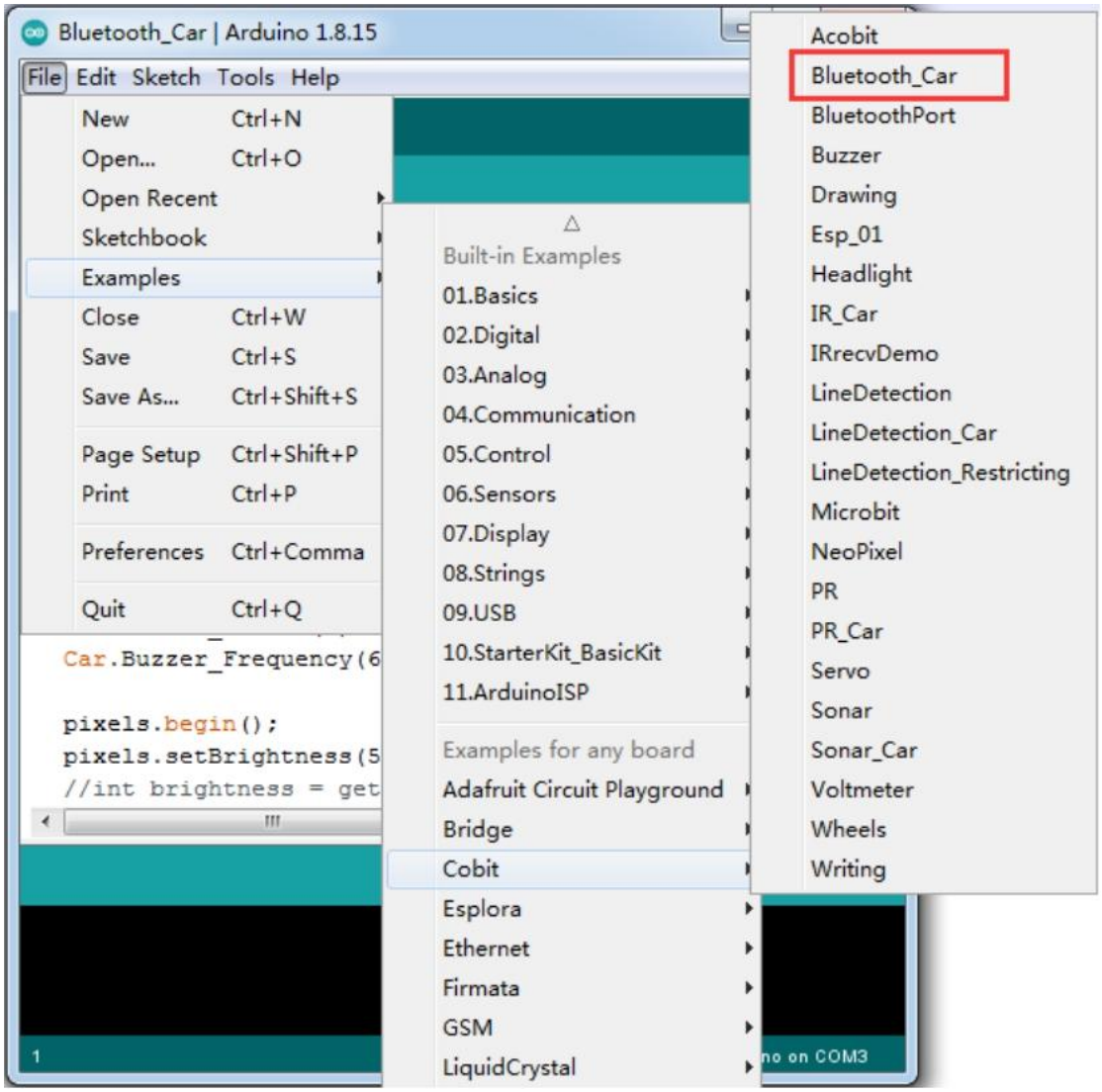
A Bluetooth port is reserved on Cobit. This port is a Soft serial port with a fixed baud rate of 115200. The connection method with the Bluetooth module is as follows:

Bluetooth port	V（5V）	G	D8	D7
Bluetooth module	VCC	GND	TX	RX

After uploading the code, connect the Bluetooth to the port, then use the Bluetooth Make the APP of your mobile phone to pair the Bluetooth, open the serial monitor, set the baud rate to 9600, press any button on the Bluetooth APP, the serial monitor of the Arduino IDE will print the buttons value of the Bluetooth APP.

Note: Bluetooth must be in serial port transparent transmission mode to transmit data to Cobit, and the module defaults to transparent transmission mode.

20_Bluetooth Car

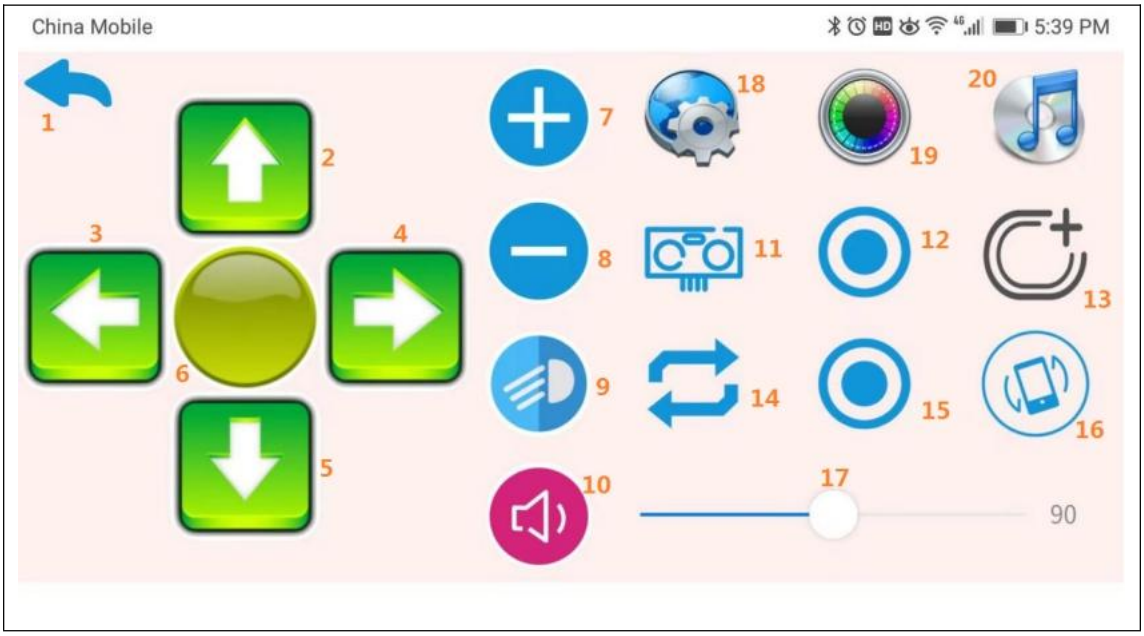


A Bluetooth port is reserved on Cobit. It is a Soft serial port with a fixed baud rate of 115200. The connection method with the Bluetooth module is as follows:

Bluetooth port	V（5V）	G	D8	D7
Bluetooth module	VCC	GND	TX	RX

We have developed a Bluetooth APP for Android phones. Each button has a different button value. In the program we set, we assign different functions to each button, and send the button value to Cobit through the mobile phone Bluetooth APP, and Cobit can realize different functions.

Button value and corresponding function:



SN	1	2	3	4	5
Function	return to previous menu	forward	turn left	turn right	back
SN	6	7	8	9	10
Function	stop	reserved for use	reserved for use	on/off head light	control the buzzer
SN	11	12	13	14	15
Function	ultrasonic obstacle avoidance robot	reserved for use	reserved for use	reserved for use	reserved for use
SN	16	17	18	19	20
Function	Mobile phone tilt angle control	headlight brightness	switch button	switch button	switch button

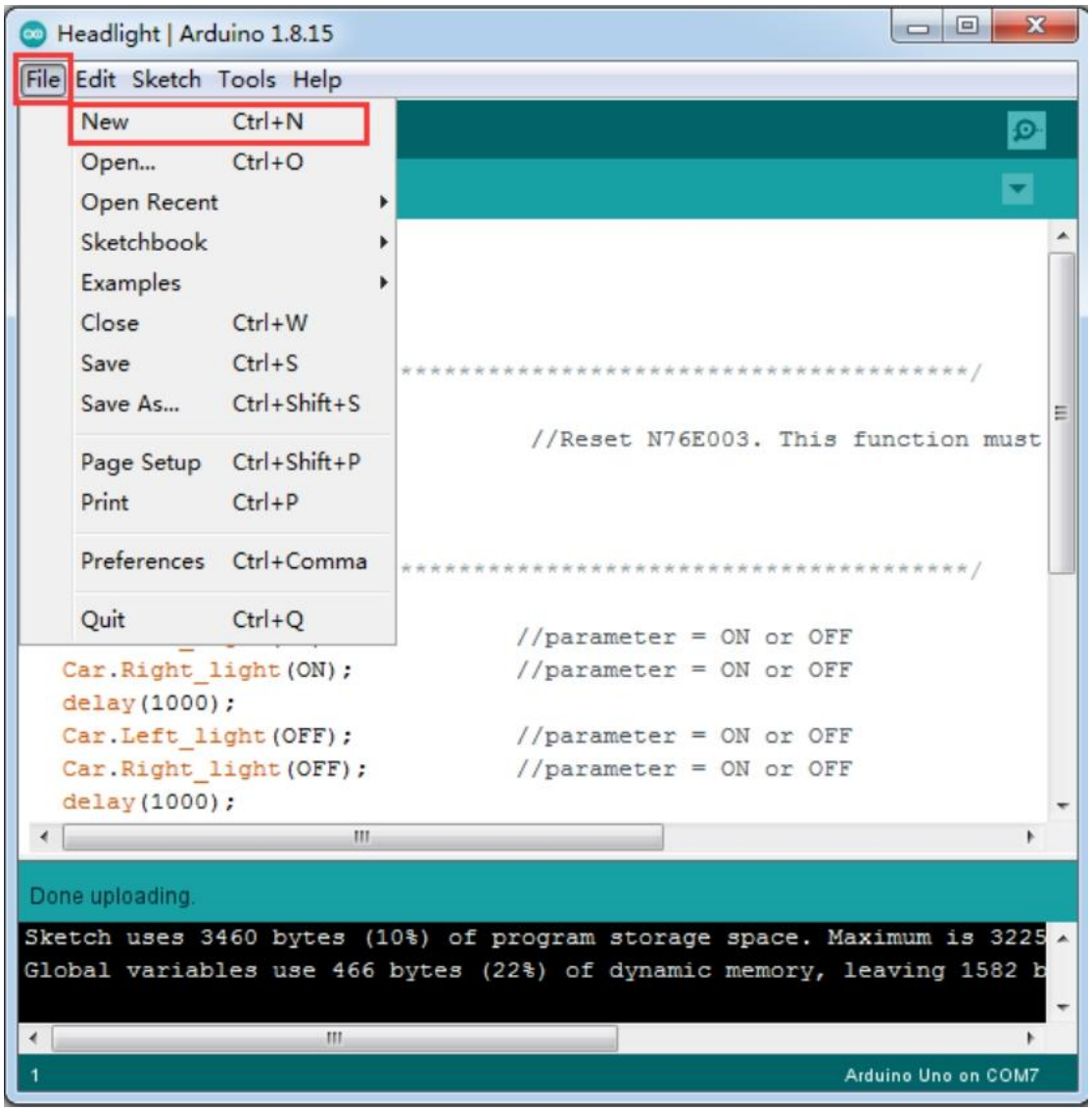
Change the color of RGB LED lights

SN	1	2	3
Function	Red	Green	Blue
SN	4	5	6
Function	Purple	Close all RGB lights	Sea Green
SN	7	8	9
Function	Yellow	Grey	Dark purple

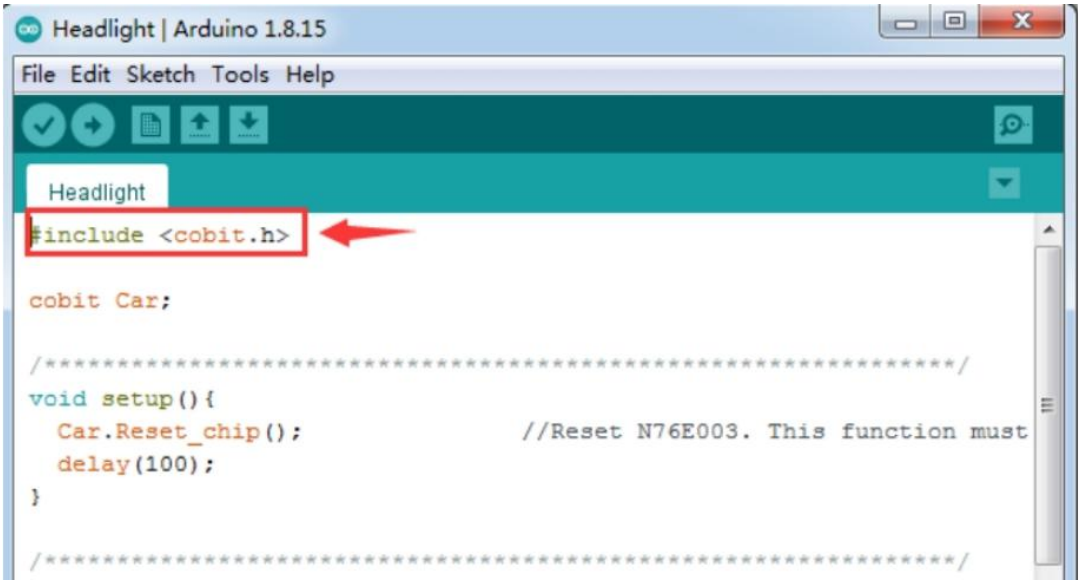
New Cobit Project

If you want to use Cobit for implement of custom functions, you only need to create a new project, include the necessary head files in the code, and then use the API interface reserved by the sample or refer to the analysis of the library file to select the required functions, as follows picture.

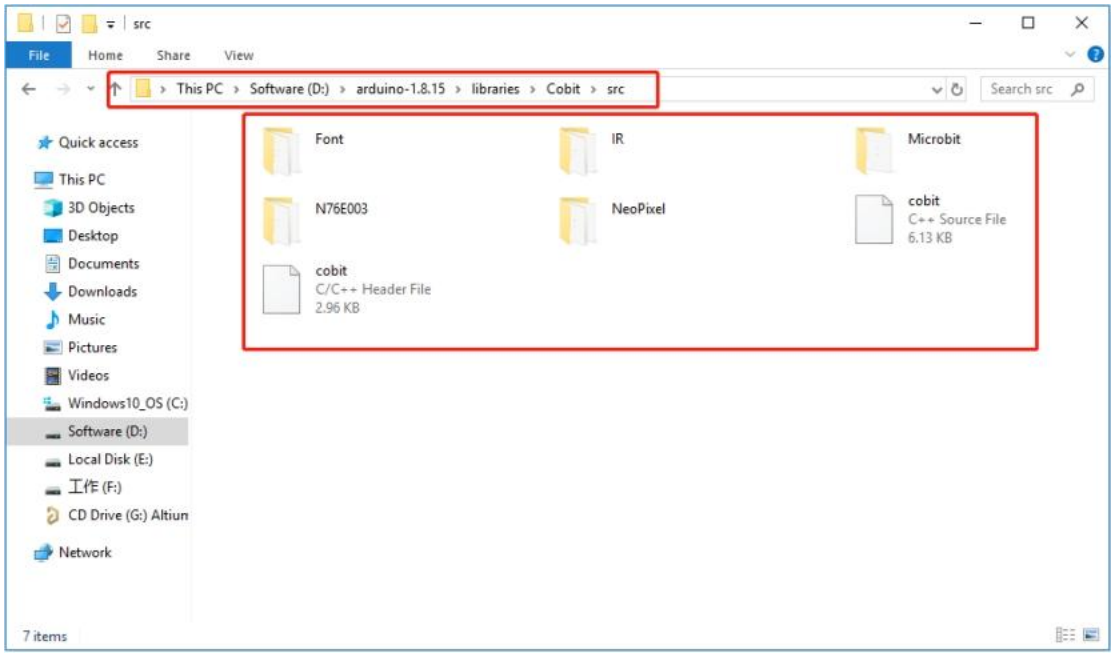
Create New Cobit Project:

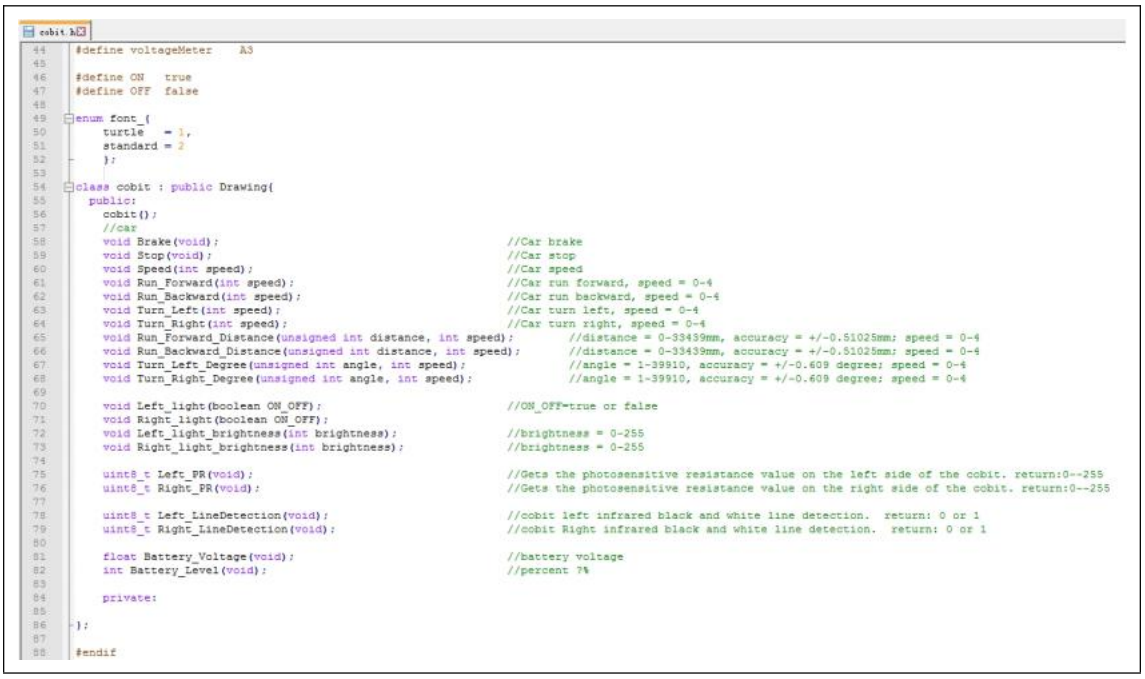


Add head files:



Check the function usage method of the library file:





```
cobit.h
#define voltageMeter    A3

#define ON   true
#define OFF  false

enum font_{
    turtle   = 1,
    standard = 2
    };

class cobit : public Drawing{
  public:
    cobit();
    //car
    void Brake(void);                                   //Car brake
    void Stop(void);                                    //Car stop
    void Speed(int speed);                              //Car speed
    void Run_Forward(int speed);                        //Car run forward, speed = 0-4
    void Run_Backward(int speed);                       //Car run backward, speed = 0-4
    void Turn_Left(int speed);                          //Car turn left, speed = 0-4
    void Turn_Right(int speed);                         //Car turn right, speed = 0-4
    void Run_Forward_Distance(unsigned int distance, int speed);        //distance = 0-33439mm, accuracy = +/-0.51025mm; speed = 0-4
    void Run_Backward_Distance(unsigned int distance, int speed);       //distance = 0-33439mm, accuracy = +/-0.51025mm; speed = 0-4
    void Turn_Left_Degree(unsigned int angle, int speed);               //angle = 1-39910, accuracy = +/-0.609 degree; speed = 0-4
    void Turn_Right_Degree(unsigned int angle, int speed);              //angle = 1-39910, accuracy = +/-0.609 degree; speed = 0-4

    void Left_light(boolean ON_OFF);                    //ON_OFF=true or false
    void Right_light(boolean ON_OFF);
    void Left_light_brightness(int brightness);         //brightness = 0-255
    void Right_light_brightness(int brightness);        //brightness = 0-255

    uint8_t Left_PR(void);                              //Gets the photosensitive resistance value on the left side of the cobit. return:0--255
    uint8_t Right_PR(void);                             //Gets the photosensitive resistance value on the right side of the cobit. return:0--255

    uint8_t Left_LineDetection(void);                   //cobit left infrared black and white line detection.  return: 0 or 1
    uint8_t Right_LineDetection(void);                  //cobit Right infrared black and white line detection.  return: 0 or 1

    float Battery_Voltage(void);                        //battery voltage
    int Battery_Level(void);                            //percent ?%

    private:

};

#endif
```

Reminder: It is recommended to use "Notepad++" software to view the above documents, download URL.

https://notepad-plus-plus.org/downloads/

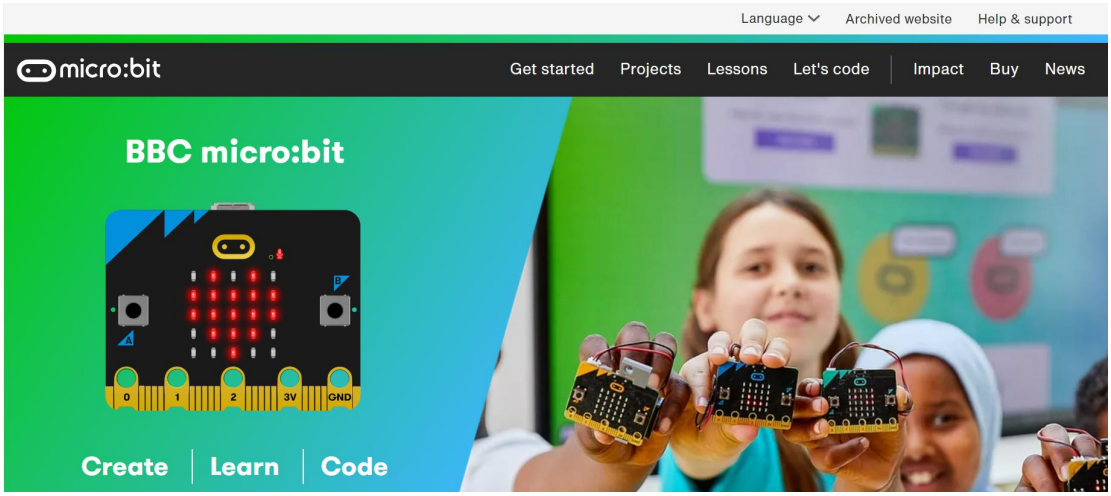
8. Geting start with Cobit for micro:bit

The Micro:bit Educational Foundation is a not-for-profit organisation founded in the UK in 2016, with the aim of inspiring every child to create their best digital future.

They do this by:

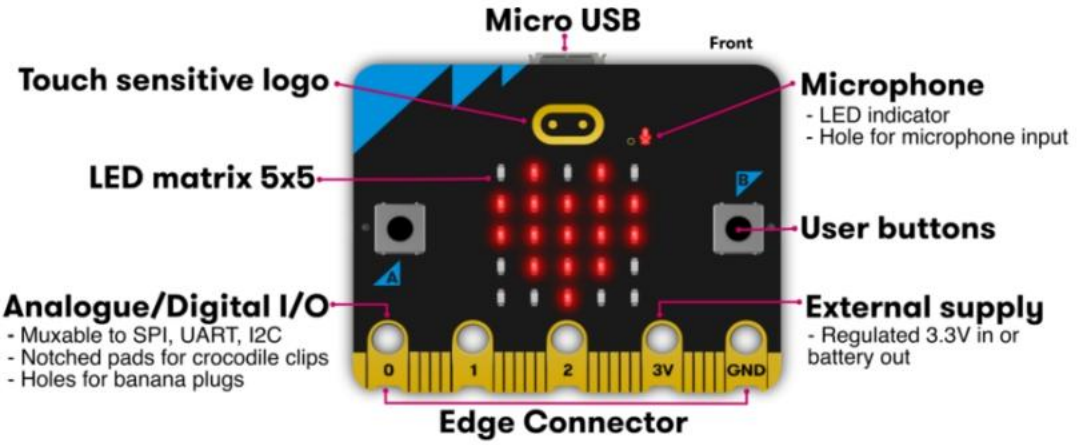
developing hardware and software that inspires young people to get excited about technology and the opportunities it presents for them creating free, user-friendly educational resources to support teachers in delivering engaging and creative lessons working with like-minded partners to deliver high-impact educational programmes across the globe.

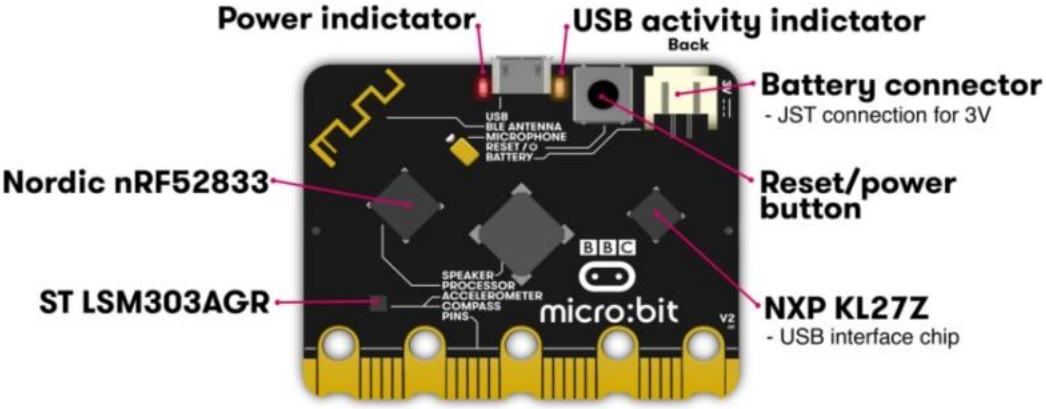
Micro:bit official website：https://microbit.org/

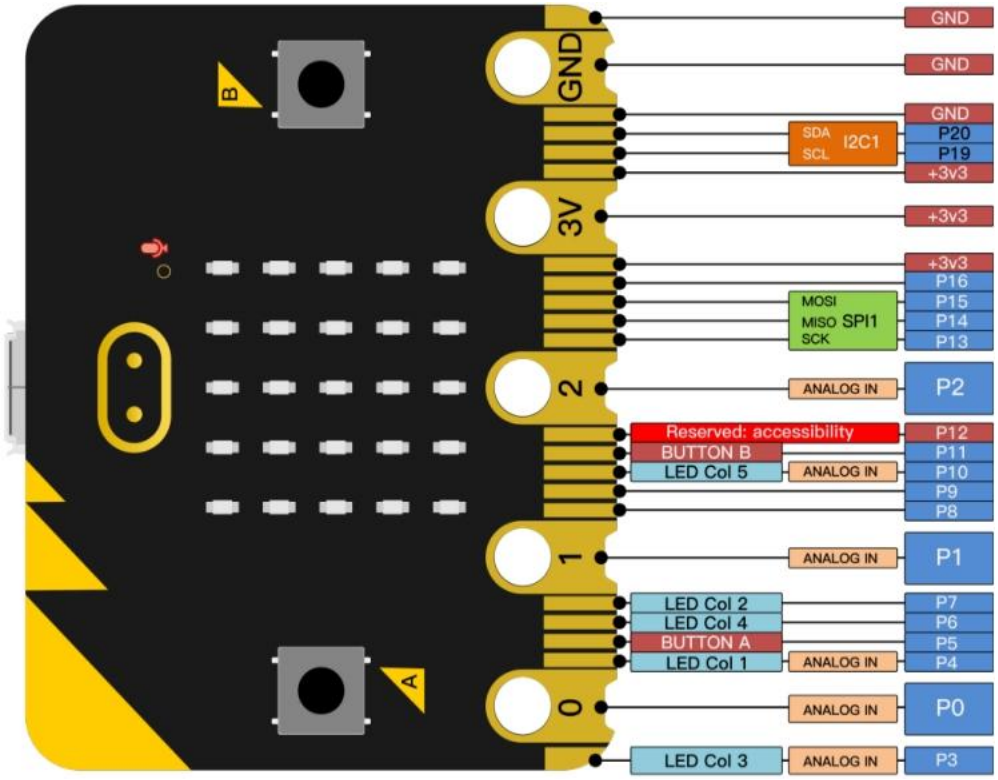


8.1 Micro:bit Development Board

Micro:bit has its own dedicated development board, which is divided into two versions, V1 and V2. It has powerful functions and integrates most commonly used sensors. It retains an externally expanded IO port for external Expand the experiment.







More details see here： https://tech.microbit.org/
https://microbit.org/get-started/user-guide/overview/

8.2 Micro:bit Editor and Programming Language

Editor selection：

You can program your micro:bit in the online MakeCode block or Python text editors.

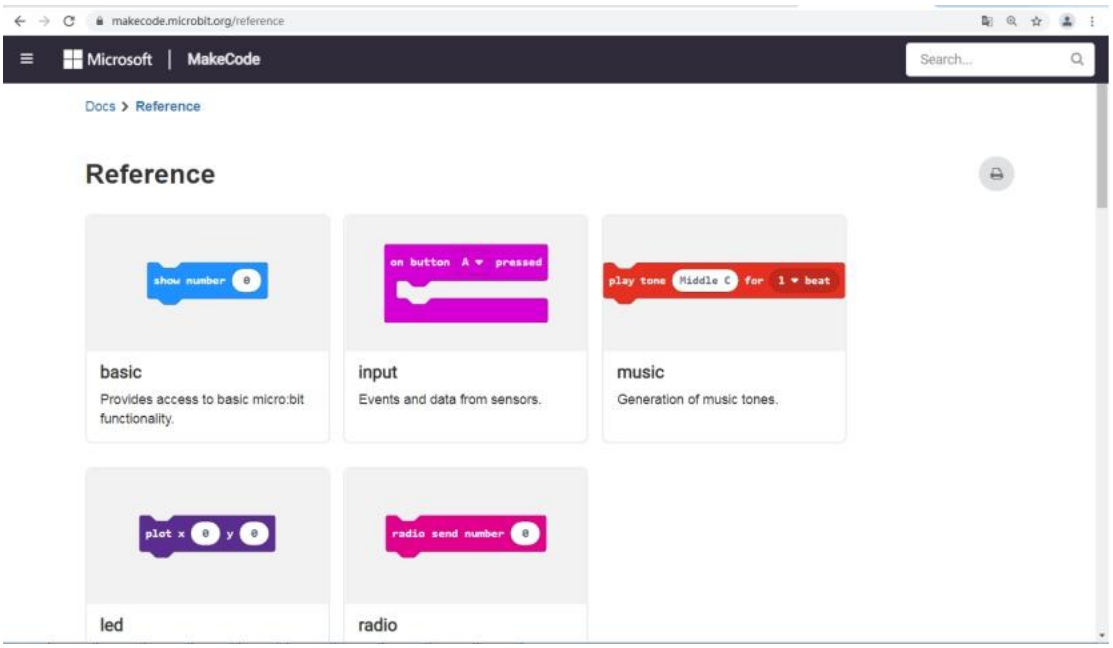
Our **Let's code** page helps you choose the one that's right for you.

Let’s code page: https://microbit.org/code/

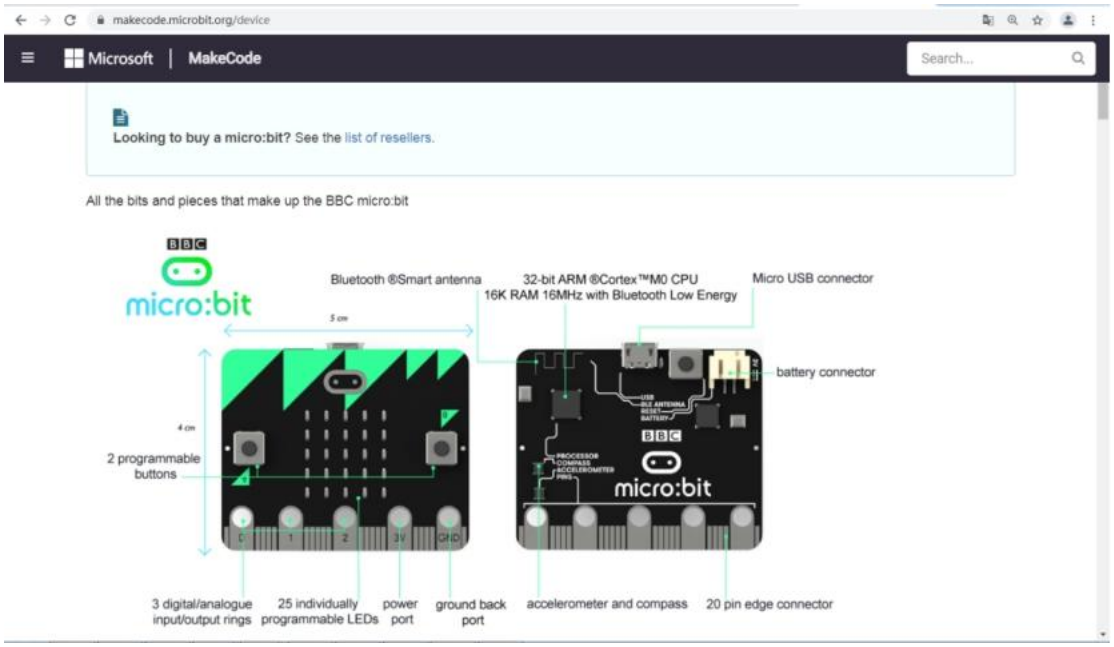
Grammar Tutorial:

The Micro:bit platform provides official API and device usage references, including detailed usage methods of makecode, python, and JavaScript.

APIs: https://makecode.microbit.org/reference

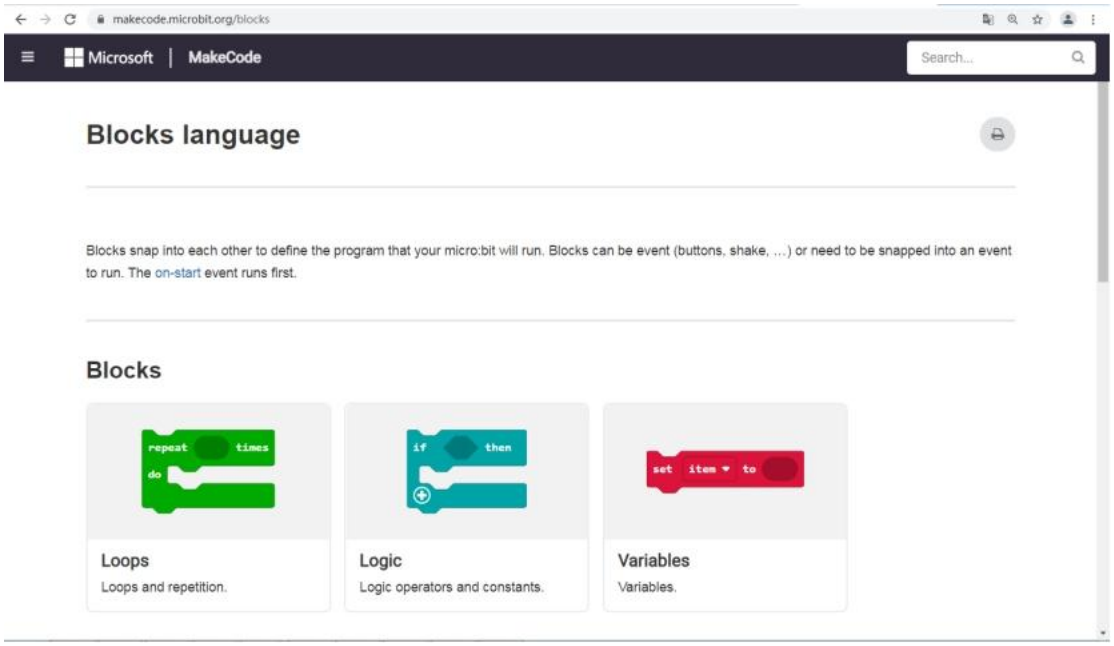


Device: https://makecode.microbit.org/device

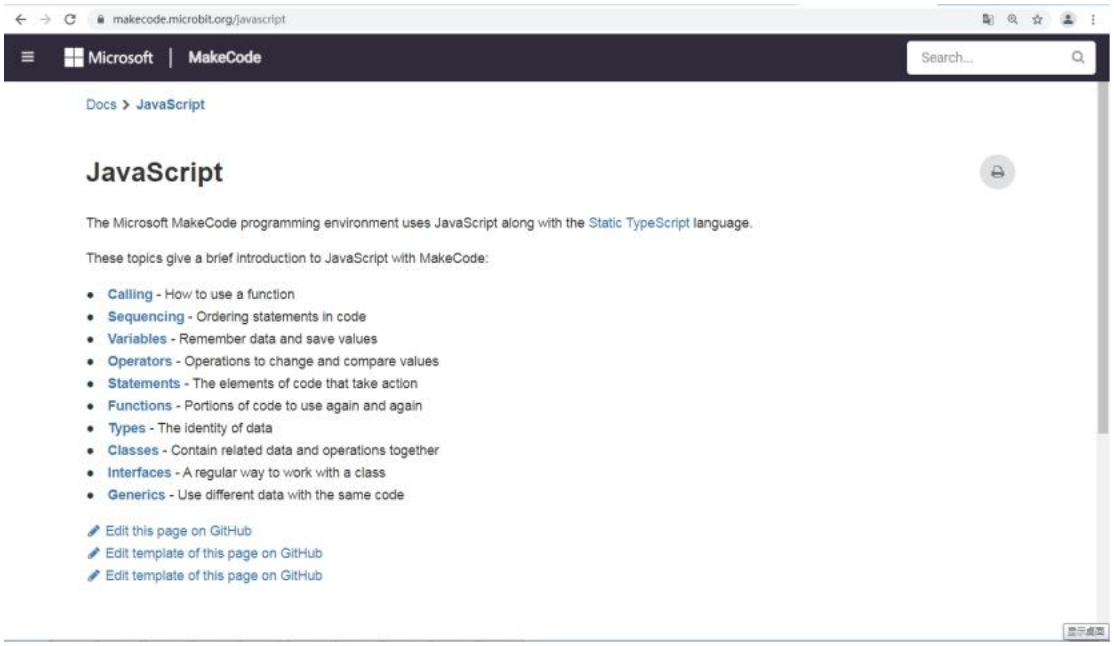


Logic and data type reference:

Blocks language: https://makecode.microbit.org/blocks



JavaScript language: https://makecode.microbit.org/javascript

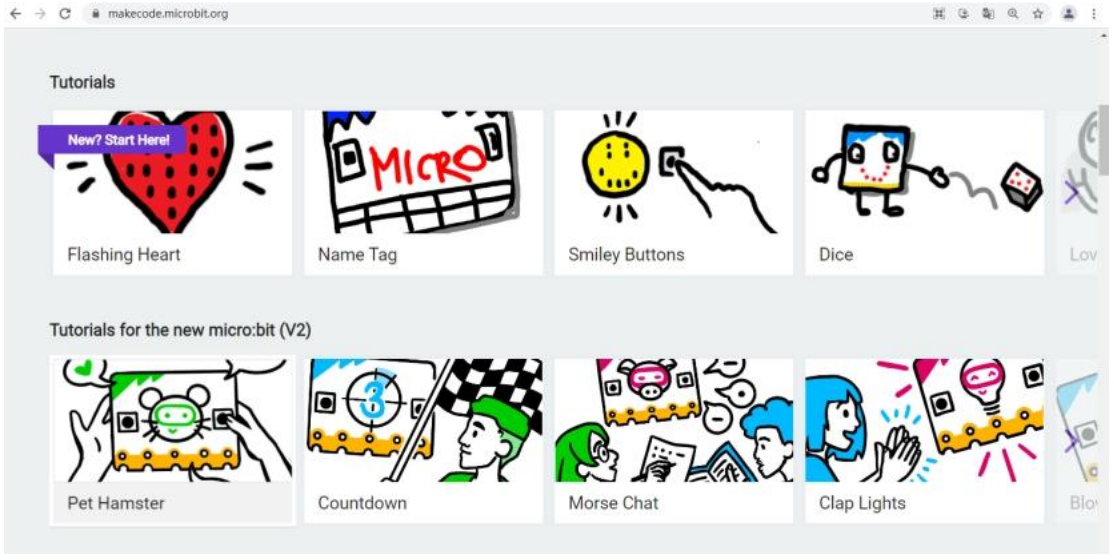


Get started micro:bit

The best way to learn is to do it. Start with imitation and follow the learning files to do it again. The official website of Micro:bit provides a wealth of video tutorials. We will follow the video to operate it again and be familiar with the basic functions of Micro:bit development. Please be sure to carefully review the video tutorials linked below.

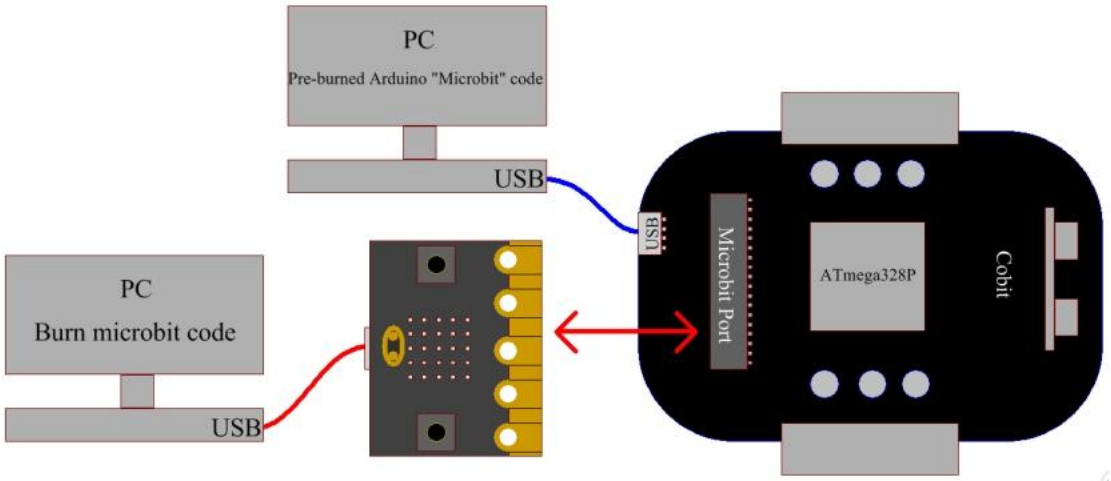
Video tutorial link:https://www.microbit.org/get-started/first-steps/introduction/

More video tutorials can also refer to the link： https://makecode.microbit.org/



Uses Micro:bit for Programming

We use makecode to develop a dedicated expansion package for Cobit, so that we can also use micro:bit to program Cobit. The use of this micro:bit expansion package is based on the burned arduino "microbit" code. Without the pre-burned arduino "microbit" code, the micro:bit extension package will not work.



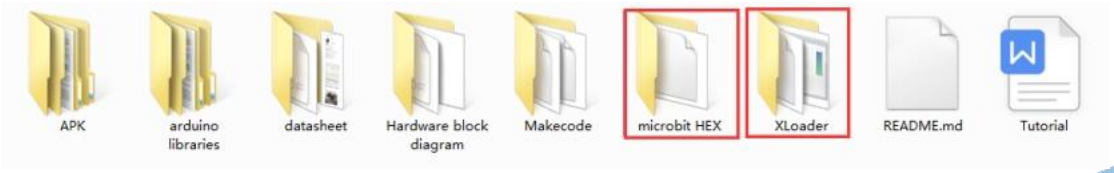
8.3 Burn Arduino Microbit Code

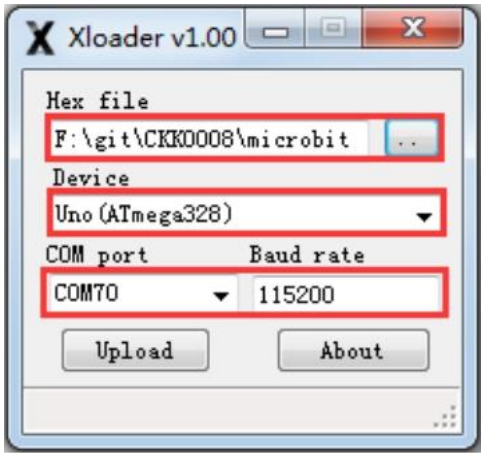
There are two ways to burn the Arduino microbit code, you just need to choose one of the two.

XLoader Burning

XLoader burning does not need to set up a programming environment. Just make sure to install the USB driver. If the USB driver is not installed, please refer to the "Install the USB-to-UART COM port driver" chapter of the Arduino section;

The burning application and the HEX file are stored in the "XLoader" and "microbit HEX", Open ”XLoader”folder, double click ” XLoader ” icon to start the application, select ”Microbit.hex”, and select device ”Uno (ATmega328)”, select COM port, and set the baud rate to 115200, click "Upload" to burn the code.

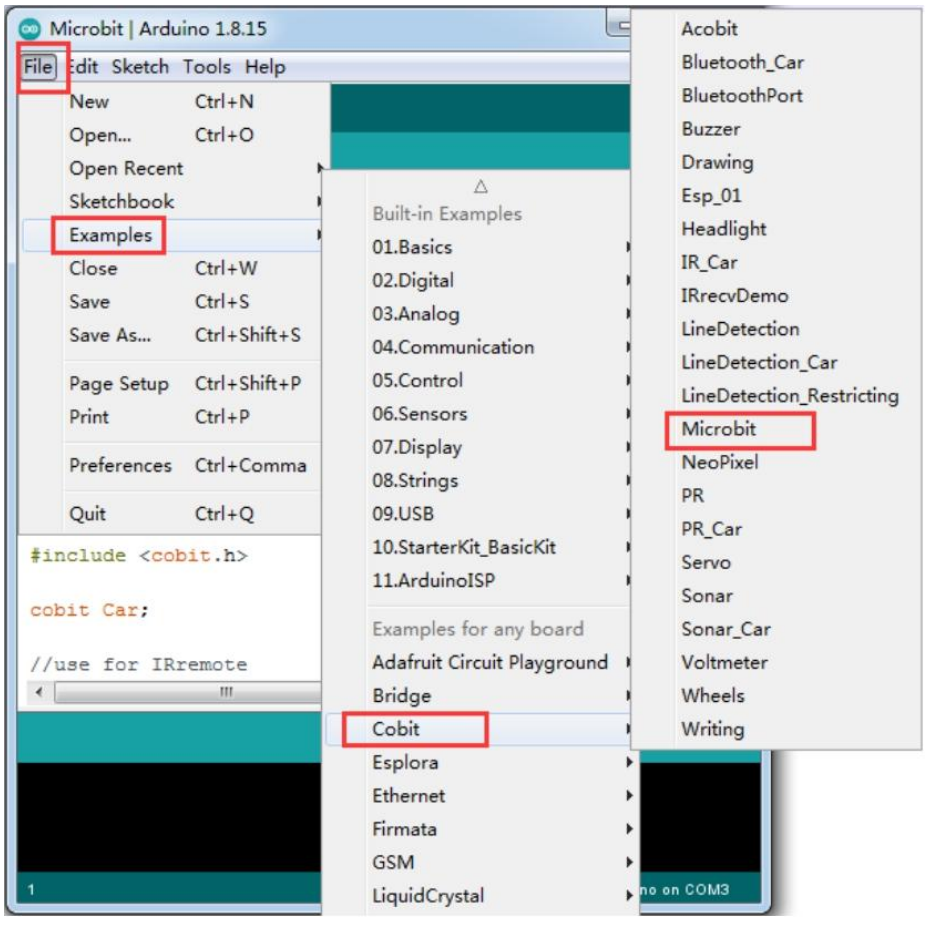




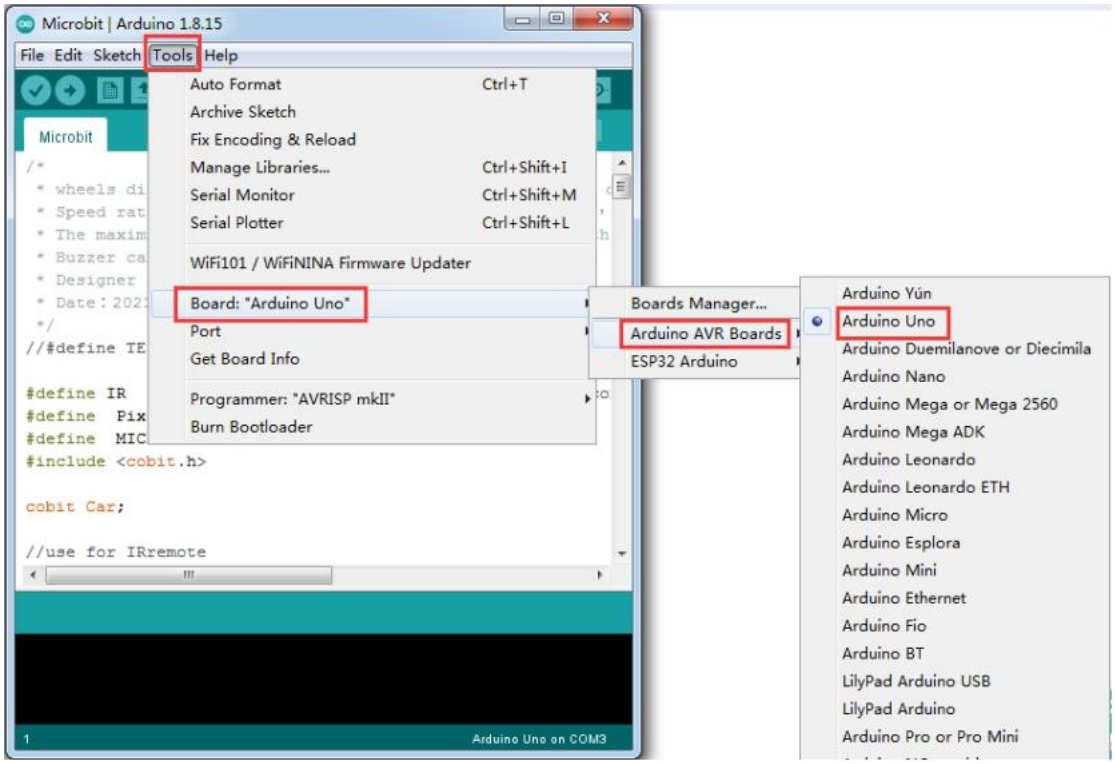
Arduino IDE Burning

Before uploading the arduino "microbit" code to the Cobit, you need to download arduino IDE, install Cobit library and USB driver. About the download method of the arduino IDE, installation of the Cobit library and installation method of the USB driver, please refer to the "Getting start with Cobit for arduino" chapter.

Select Code:

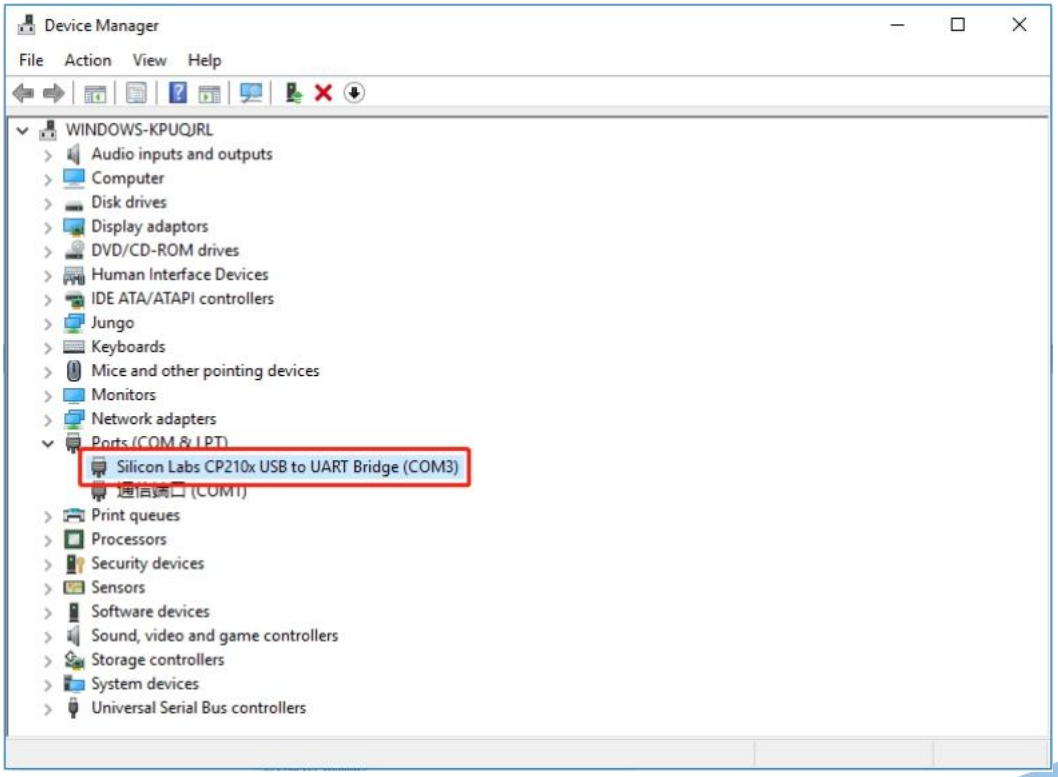


Select Board:

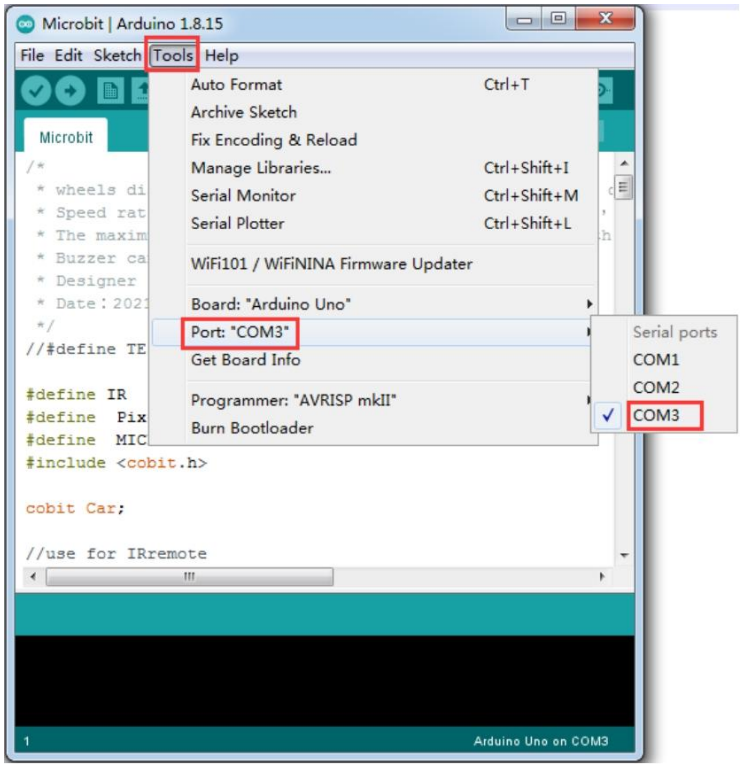


Select Port:

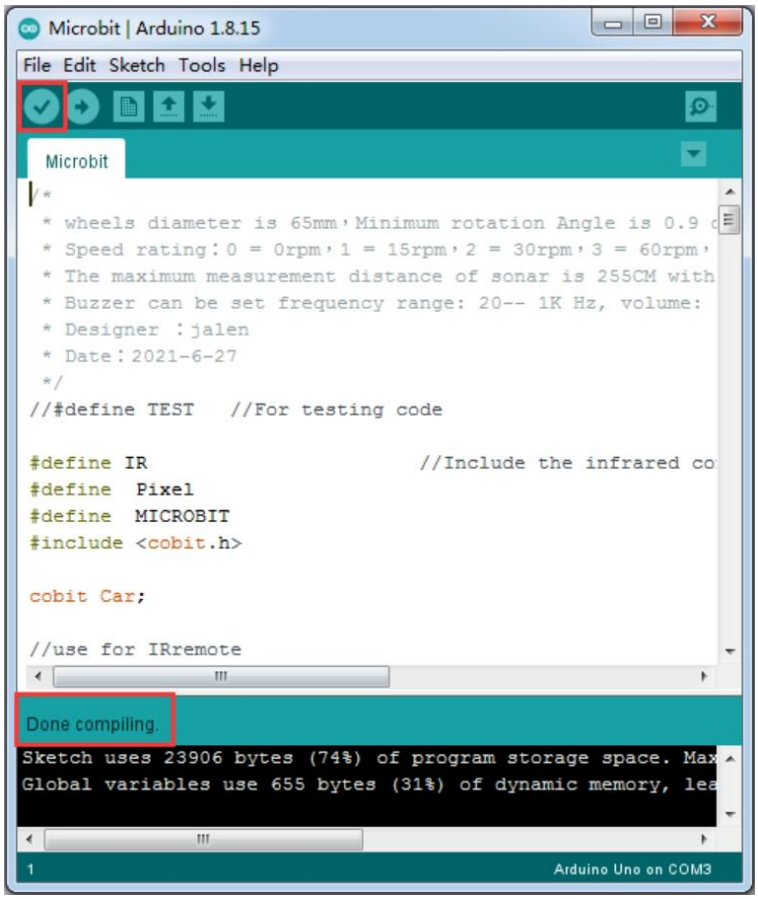
Use the Type C USB cable to connect the Cobit to the PC, and open the device manager of the PC (if the device displays an exclamation mark, please refer to the "Install the USB-to-UART COM port driver" chapter of the Arduino chapter to install the driver).



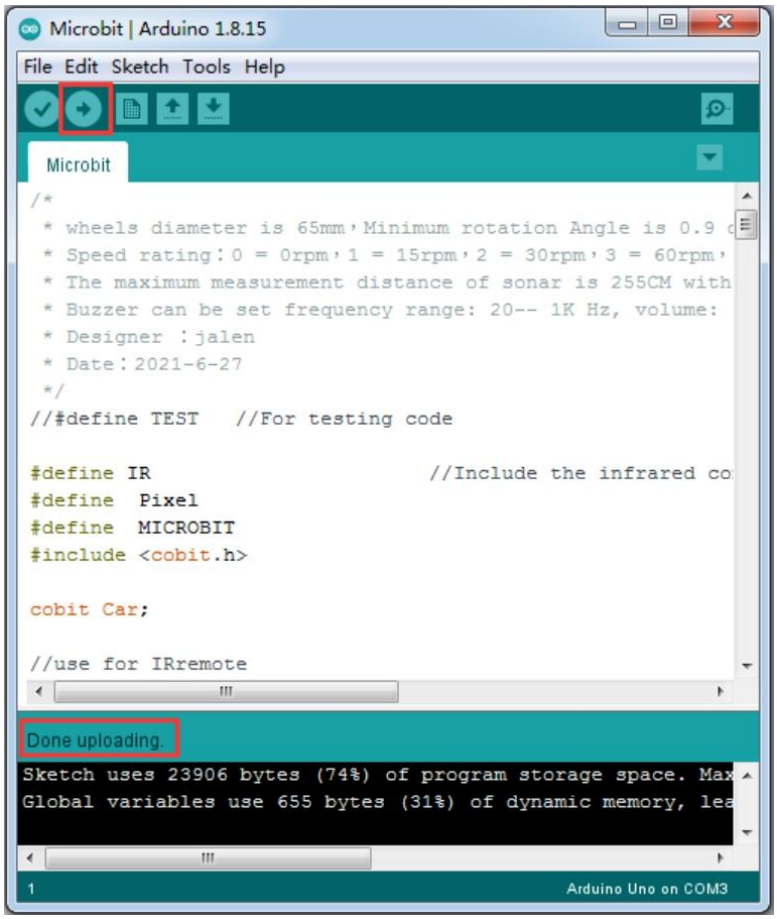
Select the corresponding port in the arduino IDE according to the port of the device manager.



Check the code (if there is an error in the code, an orange error message is printed in the information window at the bottom of the IDE).



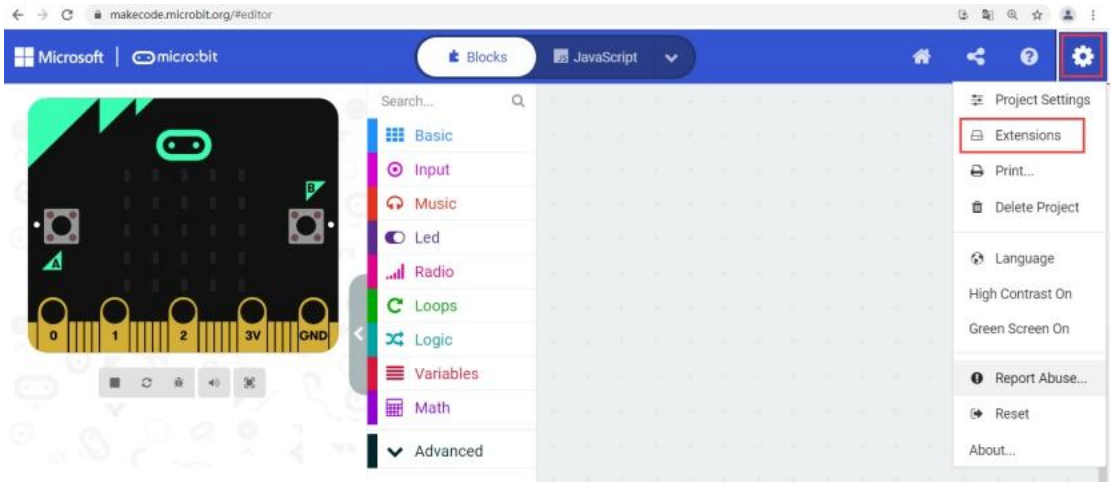
Upload the Code:

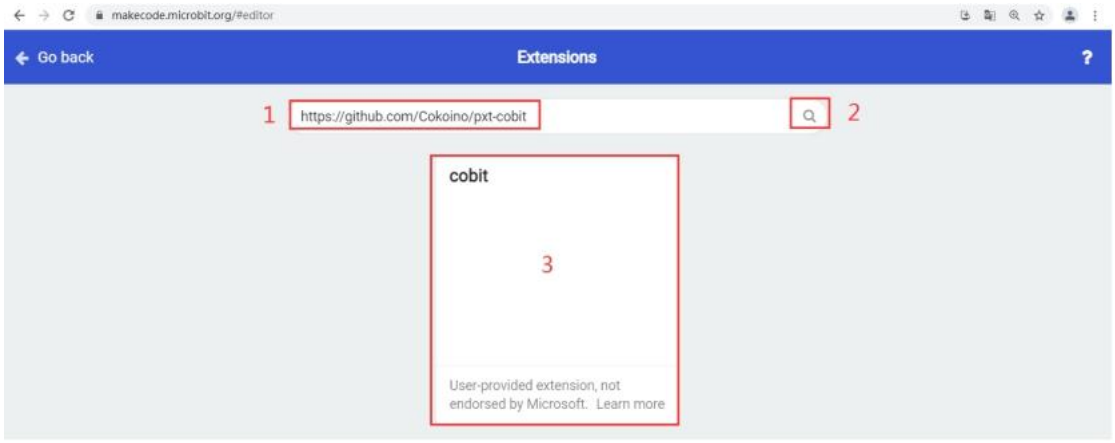


Successfully Burn Arduino Microbit Code to Cobit ！

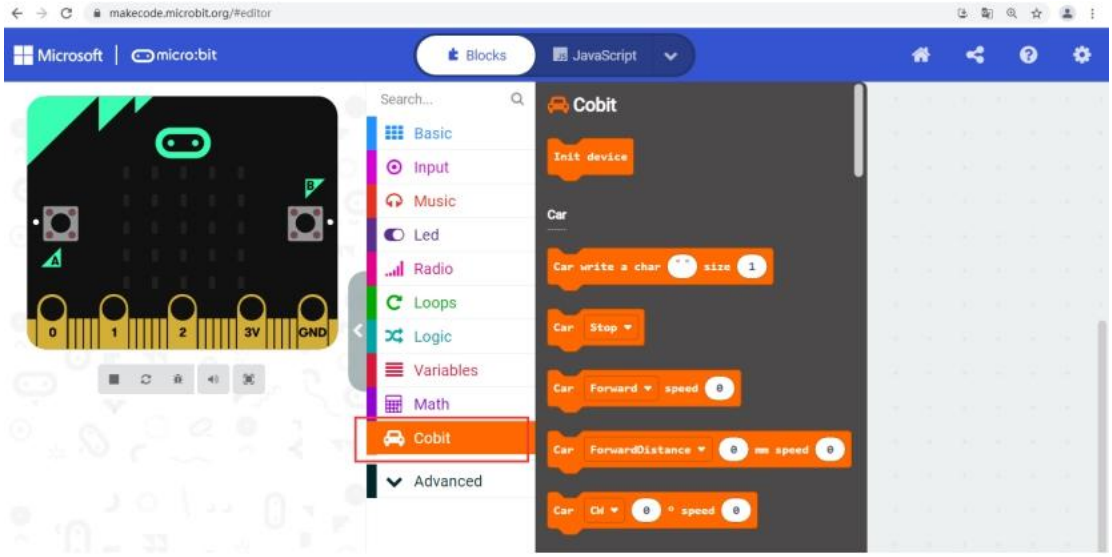
8.4 Add Micro:bit Cobit Expansion Package

Cobit extension package for Micro:bit needs to be loaded before use. The steps are as follows:





Link of Cobit expansion package based on microbit: https://github.com/Cokoino/pxt-cobit

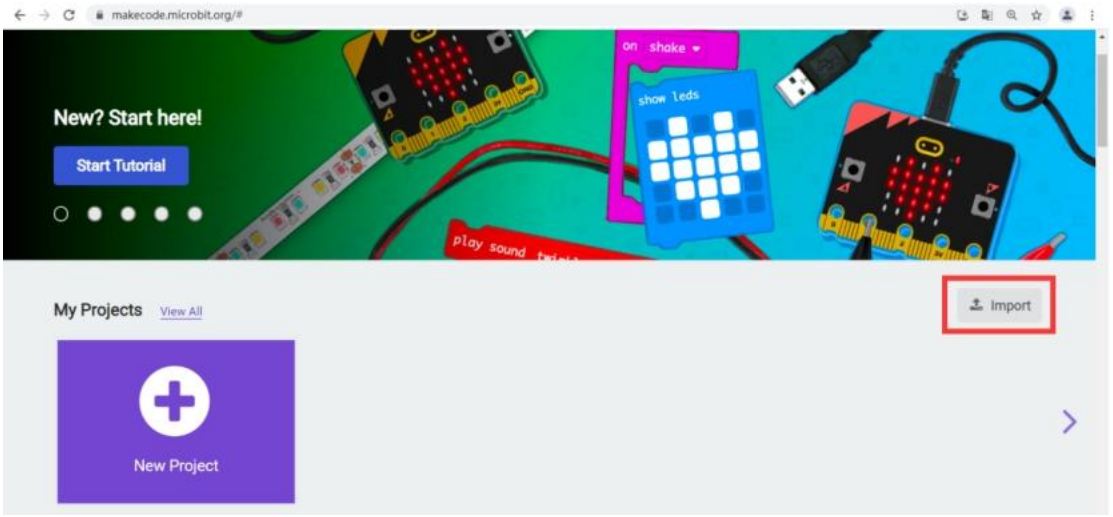


Now all preparations are complete, and you can use makecode to program Cobit.

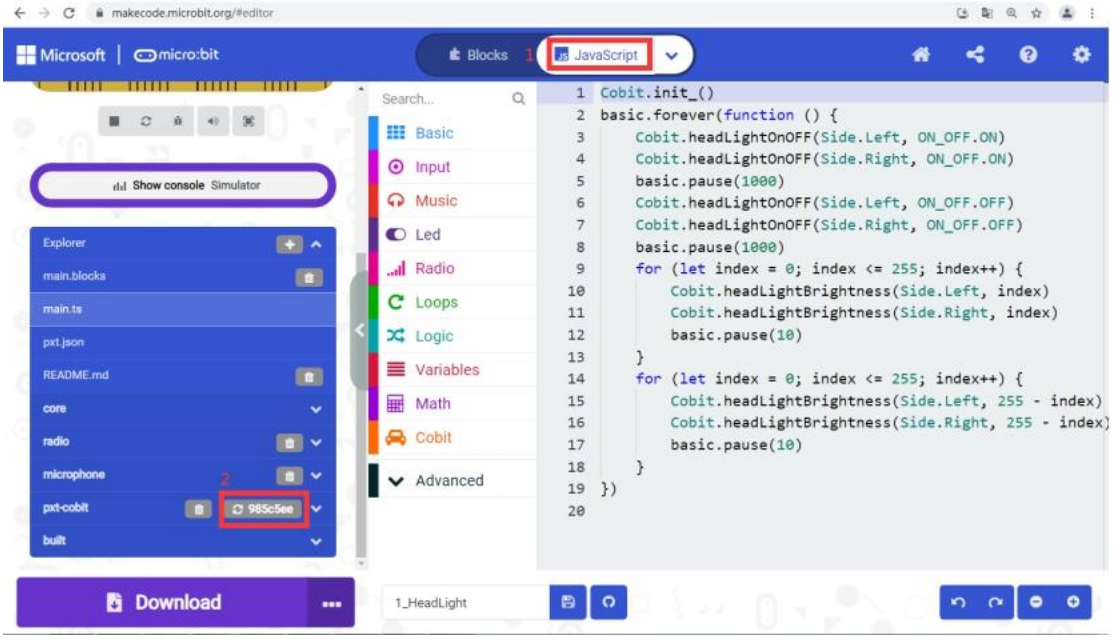
Refresh Micro:bit Cobit extension package

Micro:bit extension packages are all saved on the github website, and the modified information on the github website will have a version record on the website. For example, the expansion package V1 is added for the first time, and the project is created using the expansion package V1. After a period of time, we may optimize the expansion package and update the expansion package information on the github website. At this time, the github website will automatically The version of the expansion pack is upgraded to V2. If we use the makecode online programming webpage to import the previous project, we still use the V1 version of the expansion pack. If we want to use the latest expansion pack, we need to refresh the expansion pack manually at this time.

Import the previous project:

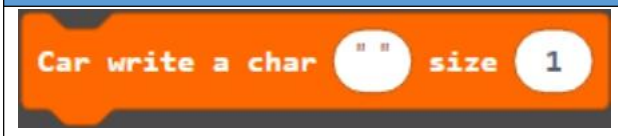
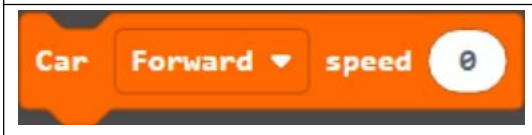
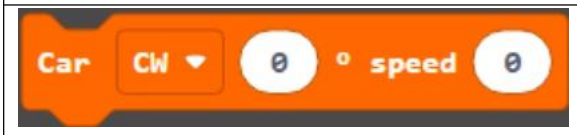
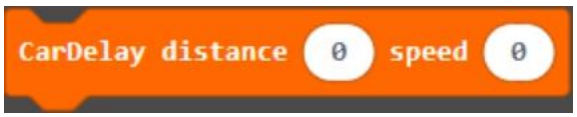


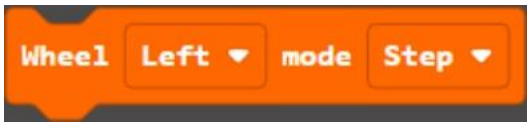
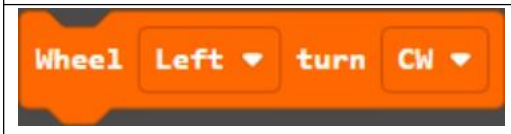
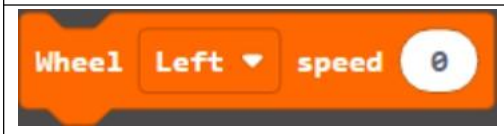
Switch to the JavaScript programming interface to refresh the extension package:

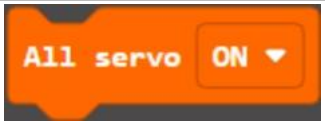
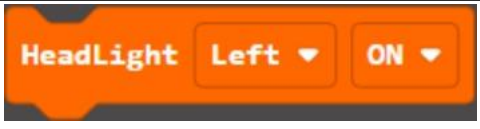


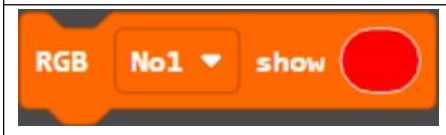
8.5 Analysis of Makecode Statement Based on Cobit

Cobit-based Makecode statements are all integrated in the Cobit extension package. Simply drag and drop to the micro:bit code editing area to use. The statement analysis is as follows:

Statement	Analysis
	The Cobit internal device initialization statement must be initialized once in "on start".
Car	
	Cobit writing statement, the first parameter is any character in the ASCII displayable characters, and the second parameter is the size of the font.
	Cobit stop running statement.
	Cobit car continuous driving statement, the first parameter is the direction, and the second parameter is the speed.
	Cobit car precise distance travel statement, the first parameter is the direction, the second parameter is the travel distance, and the third parameter is the speed.
	Cobit car precise steering statement, the first parameter is the direction, the second parameter is the rotation angle, and the third parameter is the speed.
	In the stepping mode, the car has a delay based on the travel distance and speed.

	In the stepping mode, the car has a delay based on the rotation angle and speed.
Wheels	
	Cobit wheel mode selection statement, the first parameter is wheel selection; the second parameter is mode selection, Step mode is stepping mode, which can carry out precise driving, Continue mode is continuous driving mode, and the driving distance, distance and rotation angle are not controllable .
	Cobit wheel steering selection statement, the first parameter is wheel selection, and the second parameter is steering selection.
	Cobit wheel speed selection, the first parameter is wheel selection, and the second parameter is speed selection.
	Cobit wheel precise rotation angle setting, the first parameter is wheel selection, and the second parameter is rotation angle setting.
	Cobit wheel precise rotation distance setting, the first parameter is wheel selection, and the second parameter is driving distance setting.
	Cobit wheel status selection, the first parameter is wheel selection, and the second parameter is stop or brake status selection.
	When the wheels are in stepping mode, the delay is based on the travel distance and speed.

	When the wheel is in step mode, the delay is based on the rotation angle and speed.
Sonar	
	The ultrasonic module switch state setting of the Cobit car.
	Return the ultrasonic module measurement distance data of the Cobit car.
IRremote	
	Return the data received by the infrared receiving sensor of the Cobit car.
	The key value of the infrared remote control.
Buzzer	
	The buzzer switch state setting of the Cobit car.
	Set the frequency of the buzzer.
	Set the volume of the buzzer.
Servo	
	Cobit car steering gear interface off state setting.
	Set the rotation angle of the servo. The first parameter is the selection interface, and the second parameter is the rotation angle.
Headlight	
	Headlight switch state setting, the first parameter is car light selection, and the second parameter is switch state selection.

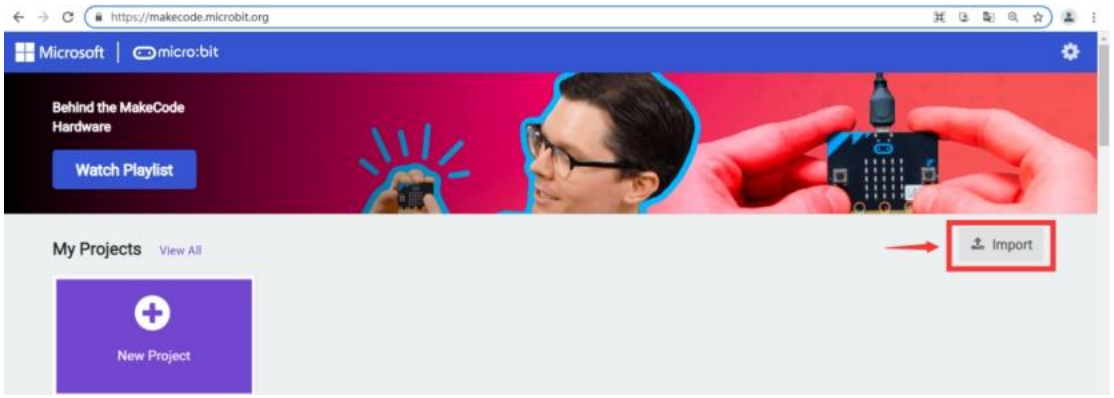
	Headlight brightness setting, the first parameter is car light selection, and the second parameter is brightness.
PR	
	Read the photoresistor data, the parameter is photoresistor selection.
LineDetection	
	Read the data of the line-detection sensor, the parameter is the line-detection sensor selection.
RGB	
	RGB light brightness setting.
	RGB color, the first parameter is RGB light selection, and the second parameter is color selection.
	RGB light arbitrary color settings, the first parameter is RGB light selection, the second 2---3 parameters are the corresponding red, green, and blue color settings, by adjusting the values of these three parameters, the RGB lights can emit different colors of light.
Voltmeter	
	Read the battery voltage of the Cobit car.
	Read the battery percentage of the Cobit car, and 0---100% is mapped to 6.5V---8.4V.
Port	

	Set the reserved IO port mode of the Cobit car. The first parameter is IO port selection, and the second parameter is IO mode selection.
	Read IO output level setting, the first parameter is IO port selection, and the second parameter is IO output level setting.
	D11 IO port outputs PWM signal, the parameter is 0---255.
	Read the digital value of the IO port, and the parameter is IO port selection.
	Read the analog value of the IO port, and the parameter is the IO port selection.

8.6 Example Tutorial

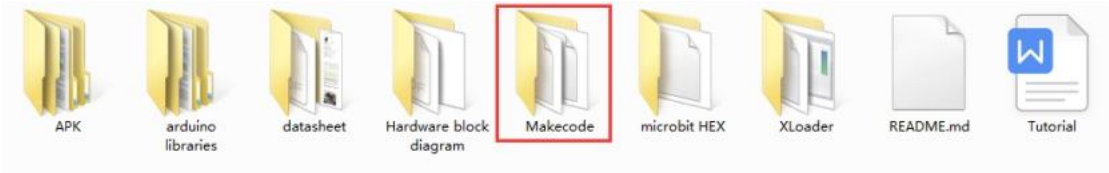
The source code of all Makecode examples is stored in the "Makecode" folder. Please open the makecode online programming webpage when you need to view it.

https://makecode.microbit.org/

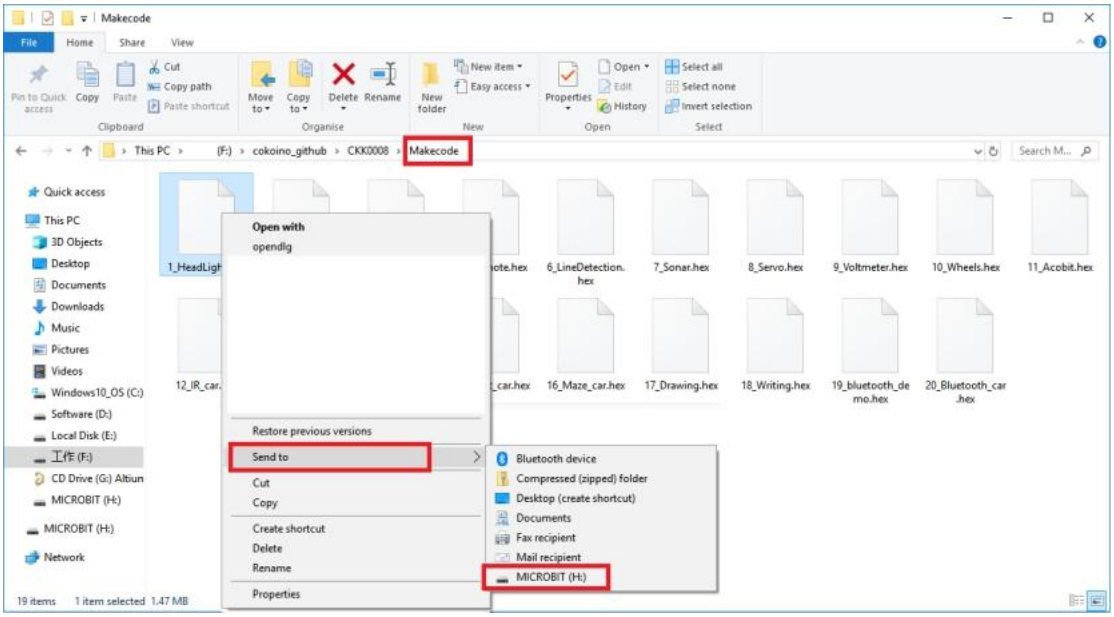


Then use the "Import" menu to import the required examples from the "Makecode"

folder.

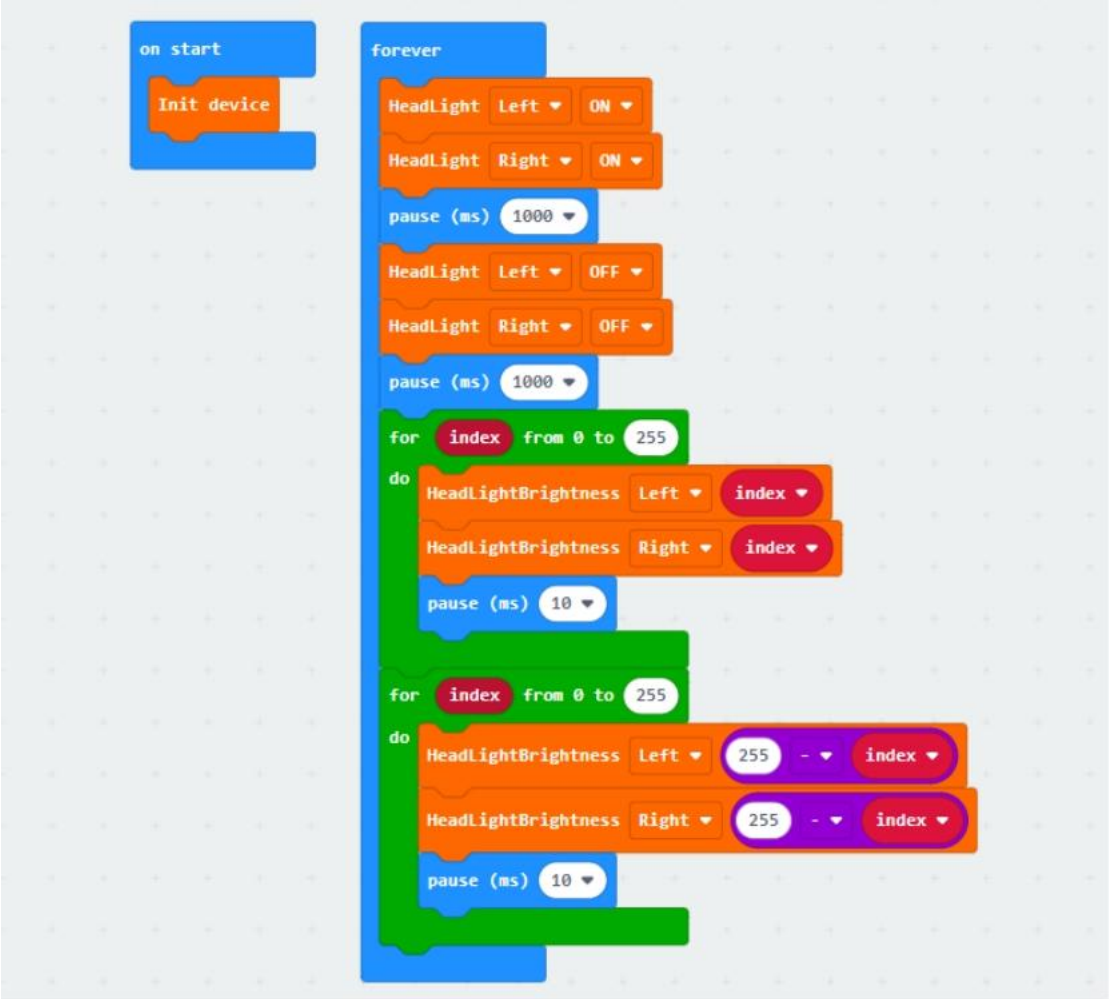


You can also open the "Makecode" folder, select the file in need, and then click the right mouse button to send the sample HEX file directly to the micro:bit board, and then check the RESULT.

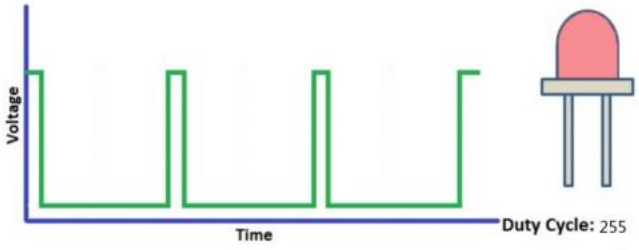


Please note that the following example is based on the pre-burned arduino "microbit" code, otherwise it will not work. For specific operations, please refer to the "Uses Micro:bit for Programming" chapter.

1_Examples of headlights

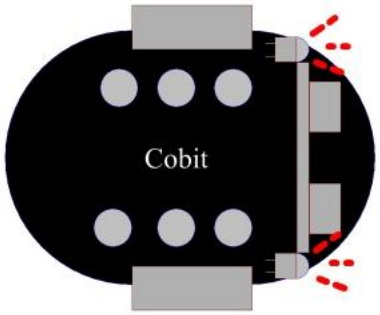


Cobit Integrats two white 5MM headlights, the "Headlight" example provides the control of their on and off states. You can also control their brightness to achieve the effect of breathing light.



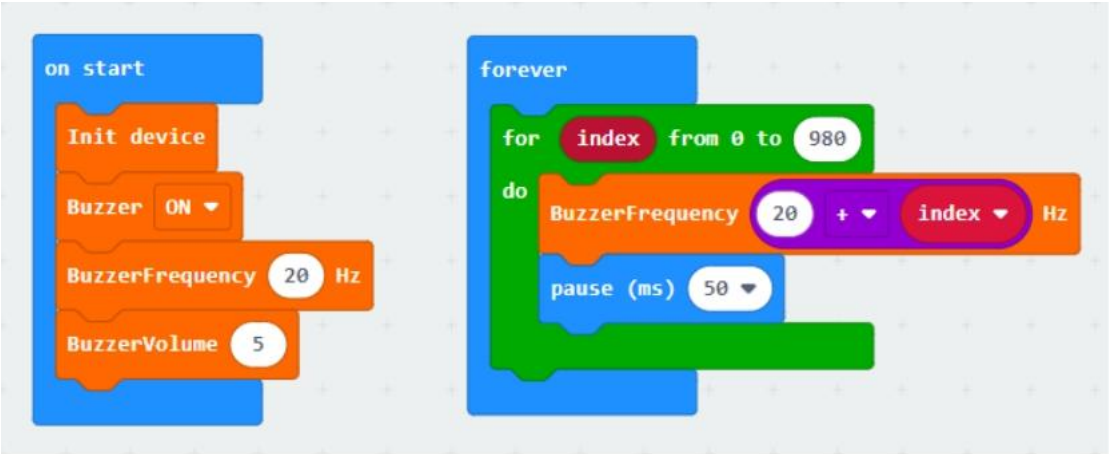
The brightness of the light is controlled by a PWM signal. PWM means pulse width modulation, that is, pulses with variable high and low time widths are periodically generated.When the high level is high, the headlights will be on, and when the low level is low, the headlights will be off. For example, the period of a pulse is 255

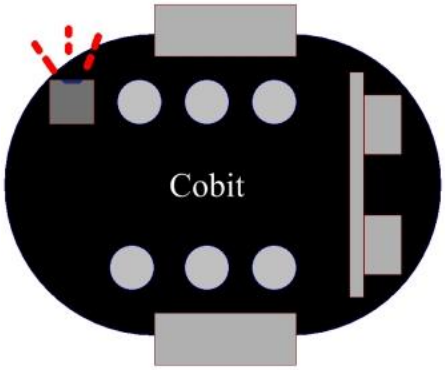
milliseconds (period = high level time + low level time), when the high level time width is 255 milliseconds , The low-level time width is 0, and the headlights are the brightest at this time; when the high-level time width is 0 and the low-level time width is 255 milliseconds, the headlights are the darkest at this time.



After uploading the code, the two headlights of Cobit will light up for 1 second and then turn off; then the breathing light will turn on, and then the breathing light will turn off; and keeps working like this.

2_Buzzer

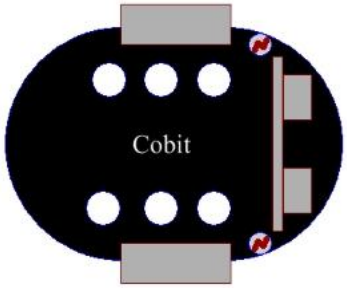




Cobit has a passive buzzer. When the passive buzzer has been supplied with direct current, the buzzer will not make sound, and some will cause the buzzer to become hot. In severe cases, the buzzer will be damaged. Give the passive buzzer a signal source with frequency to produce a sound. The "Buzzer" example provides the control of the passive buzzer on and off, and can also adjust its volume and sound frequency (20---1000 Hz).

After uploading the code, the buzzer will emit a sound of 20---1000 Hz, and keeps working like this.

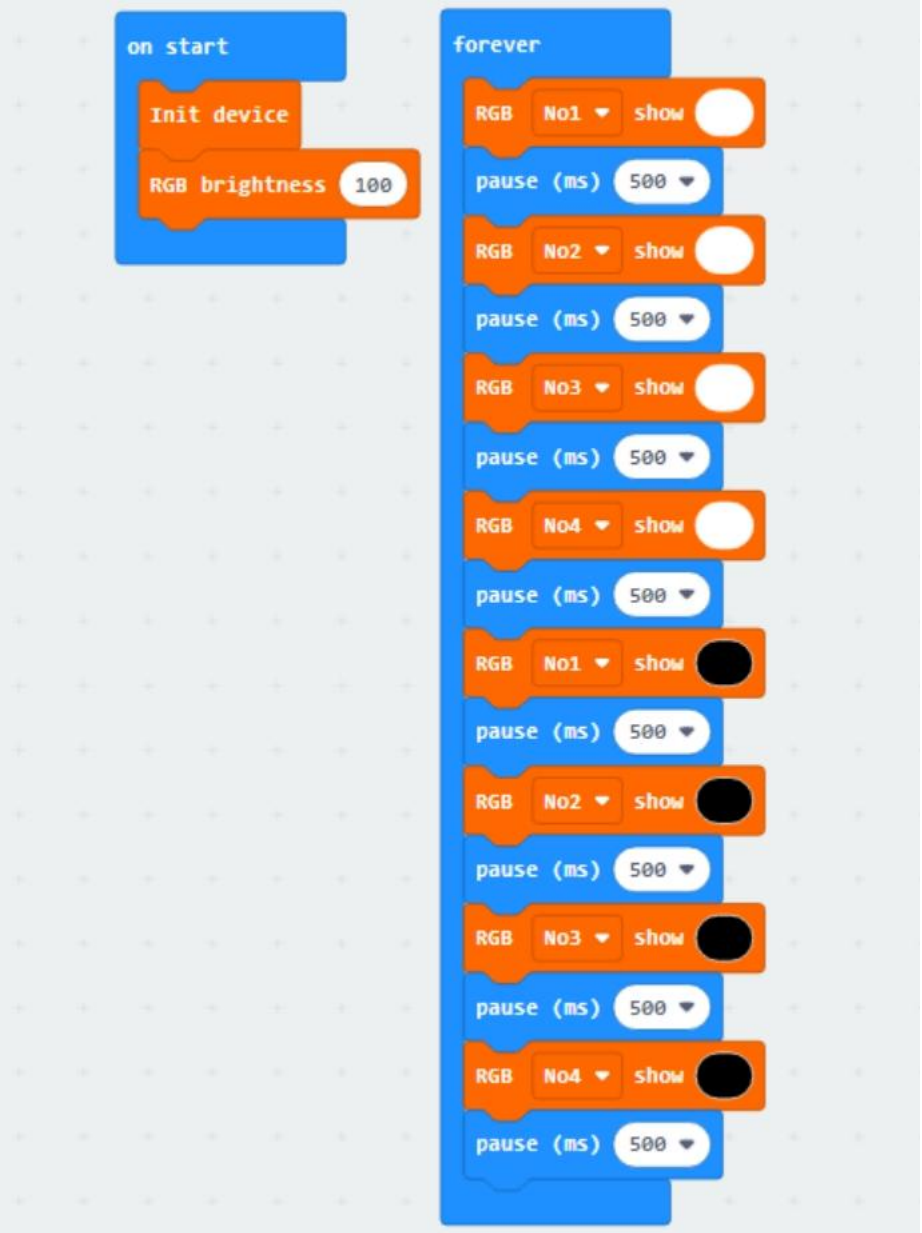
3_Examples of Photoresistors



The Cobit front end integrates two photoresistors, and the "PR" example can read their analog values.

Photoresistor is a component that converts light signals into electrical signals. The stronger the luminosity, the smaller the resistance of the photoresistor. After uploading the code, open the serial monitor of the arduino IDE, set the baud rate to 9600, and the monitor will print the data of the two photoresistors.

4_RGB light example

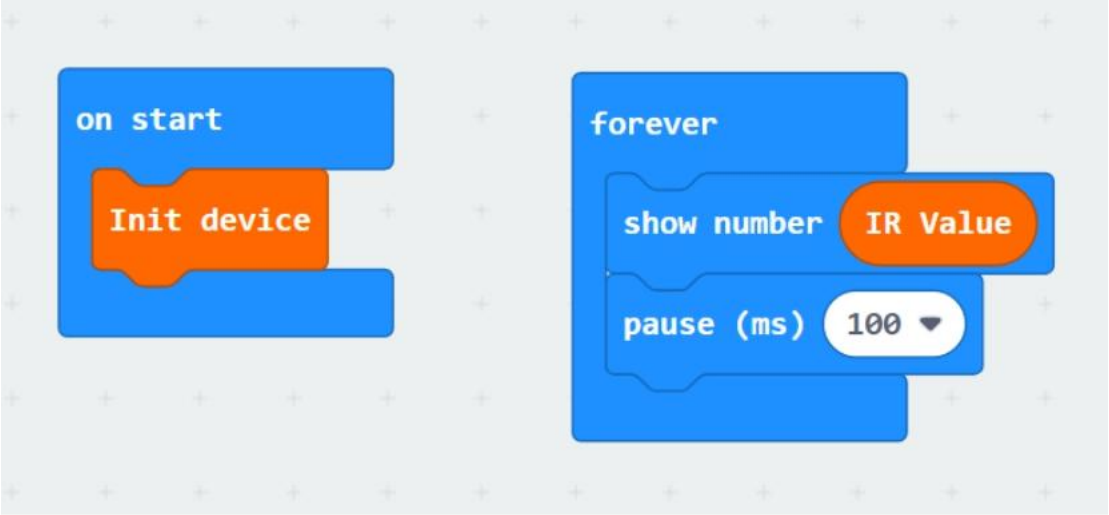


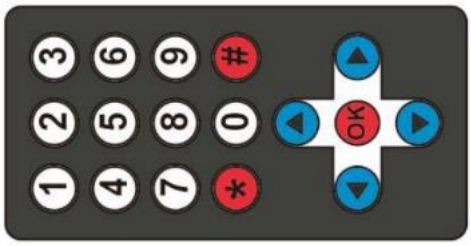
RGB

Cobit has 4 RGB lights, which can emit red, green, and blue lights. By controlling the brightness of the three lights, and then combining with each other can produce light of various colors.

After uploading the code, 4 RGB lights will turn on (white light), and then turn off.

5_Infrared receiver example





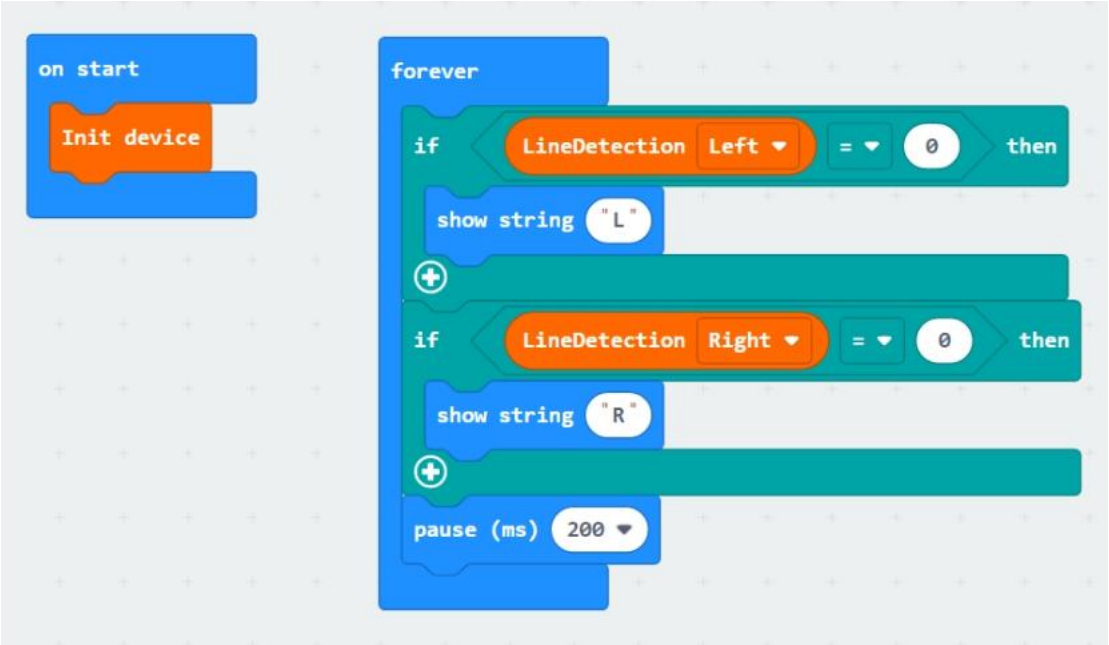
There is an infrared receiving sensor on the front and rear of the Cobit. This design makes the dead angle of the Cobit smaller when receiving data, and the received data is more accurate and more sensitive. The data received by "IR Value" is processed, only the lower 8 bits (0---255) of the data.

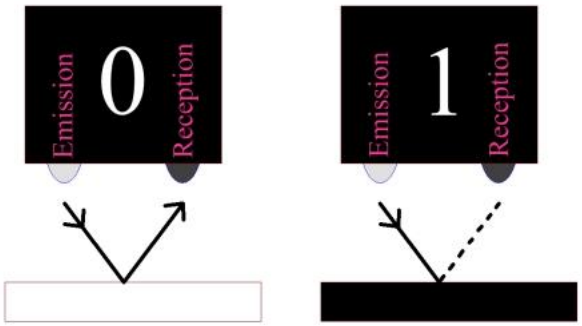
After uploading the code, the 5*5 dot matrix on the micro:bit motherboard will display the received key value of the infrared remote control.

Infrared remote control key value:

Key	0	1	2	3	4	5	6	7	8
Key value	0x67	0x5D	0x9D	0x1D	0xDD	0xFD	0x3D	0x1F	0x57
Key	9	*	#	▲	▼	◀	▶	OK	
Key value	0x6F	0x97	0x4F	0xE7	0xB5	0xEF	0xA5	0xC7	

6_Example of Infrared Black and White Line Detection





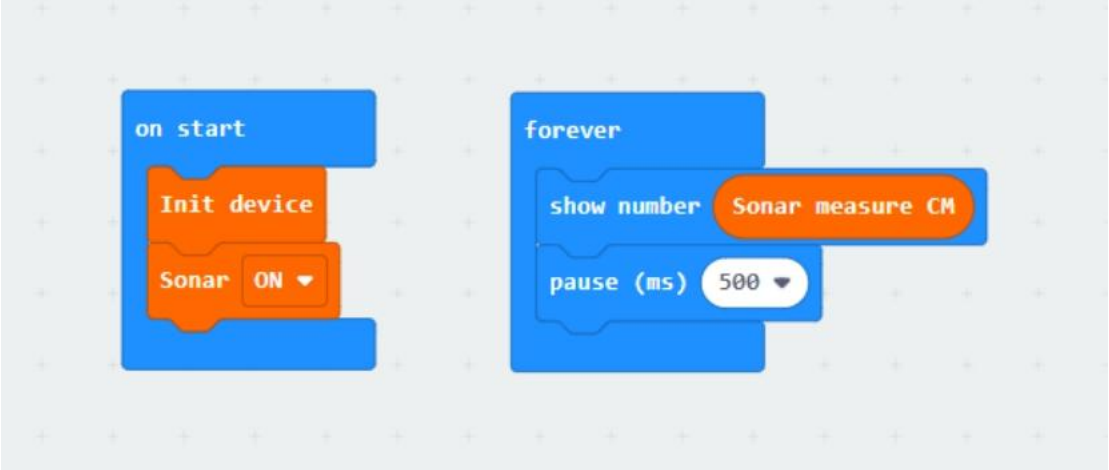
Two infrared pair tube line tracking sensors are integrated at the bottom of the Cobit, and the initial default output is high level. The infrared pair tube is divided into two parts, one is the infrared transmitting end, the other is the infrared receiving end, the infrared transmitting end transmits infrared signals, and the receiving end receives the infrared signals reflected back after hitting an object.

The black line has a good absorption effect on infrared rays. When the infrared rays are emitted from the transmitting end of the tube patrol sensor to irradiate the black line, the infrared rays are absorbed without forming a reflected signal, and the receiving end does not receive a signal, and the line patrol sensor outputs a high level (1); When the white line is irradiated, because the white line has no absorbing effect on infrared rays, the infrared signal is reflected back, the receiving end receives the signal, and the line-following sensor outputs low level (0).

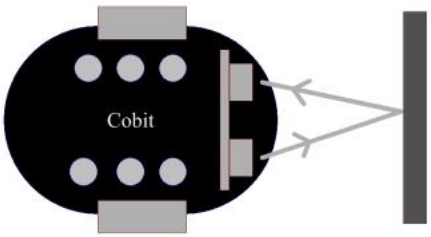
After uploading the code, the 5*5LED dot matrix of the microbit will display "R"

when a white object is used to block the line-following sensor on the right, and "L" when blocking the left.

7_Sonar Example(Ultrasonic Module)



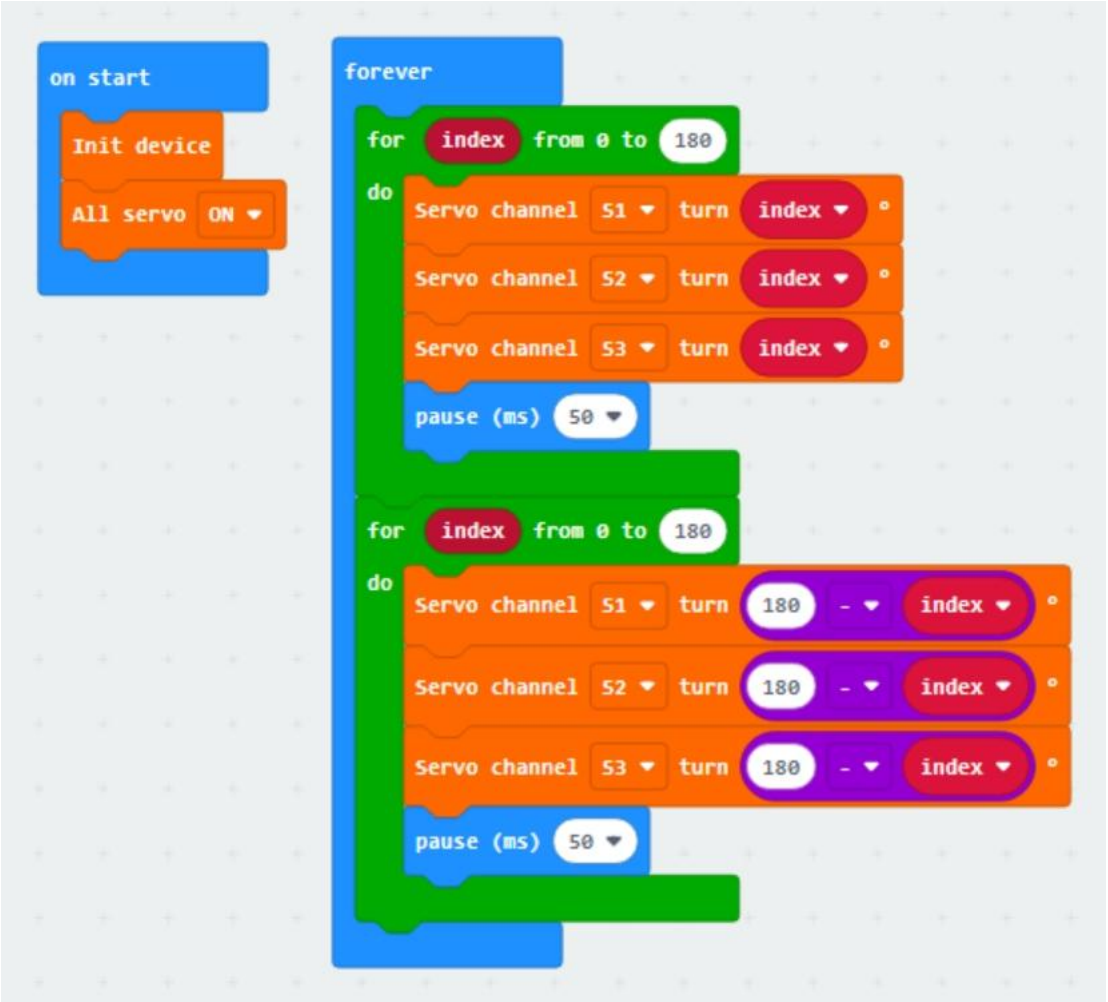
Ultrasonic module is a mechanical wave with an extremely short wavelength. The wavelength in the air is generally shorter than 2 cm (centimeters). It must rely on the medium to propagate and cannot exist in a vacuum (such as space). But because of its short wavelength, it is easily lost and scattered in the air. It is not as far as audible sound and infrasound waves. However, the short wavelength is easier to obtain anisotropic sound energy, which can be used for cleaning, ranging, etc.

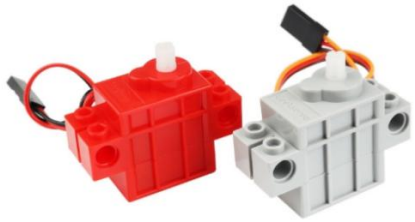


A ultrasonic module port is reserved on the Cobit, which can be connected to the HC-RS04 ultrasonic module for distance measurement. Cobit is equipped with the HC-RS04 ultrasonic module by default, and integrates the module's API in the Cobit library. The measurement data of the module can be read by using the API, which greatly reduces the difficulty of use. The maximum measuring distance is 255cm, and the accuracy is +/-1CM.

After uploading the code, place an obstacle directly in front of the Cobit, and the 5*5LED dot matrix of the microbit will display the distance between the Cobit and the obstacle, and the data unit is cm.

8_Servo Example

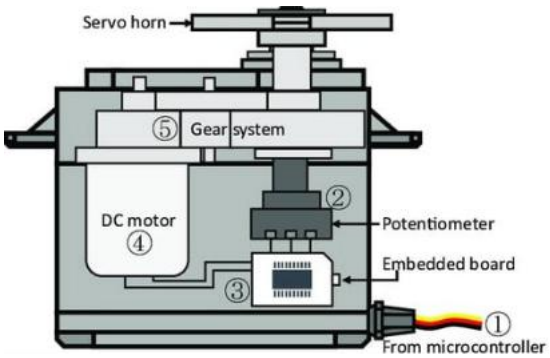




There are 3 servo drive ports reserved on the Cobit, which can be connected with Lego or SG90 servos.

After uploading the code, connect the servo to any servo port, and the servo will swing back and forth between 0---180 degrees.

About Servo:

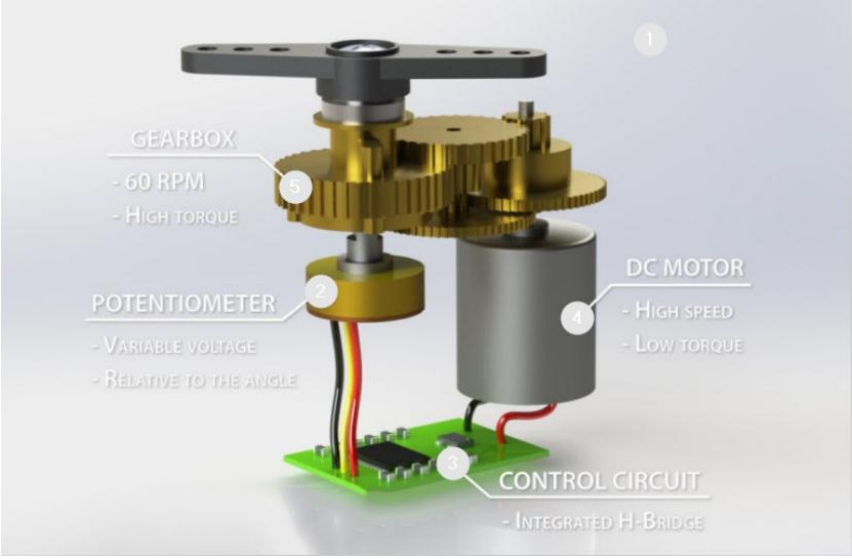


①Power negative (black), power positive (red), signal (yellow);

②Potentiometer: can measure the position of the output shaft, which belongs to the feedback part of the entire servo mechanism;

③ Internal controller: processing signals from external control, driving motors and processing feedback position signals are the core of the entire servo mechanism;④ Motor: as an actuator, how much speed, torque, and position are output by it;

⑤Transmission mechanism / servo system: This mechanism scales the stroke of the motor output to the angle of the final output according to a certain transmission ratio



The output of the servo is controlled by sending a PWM signal to the signal line of the servo. The period and duty ratio of the PWM are controllable, so the duty ratio of the PWM pulse directly determines the position of the output shaft.

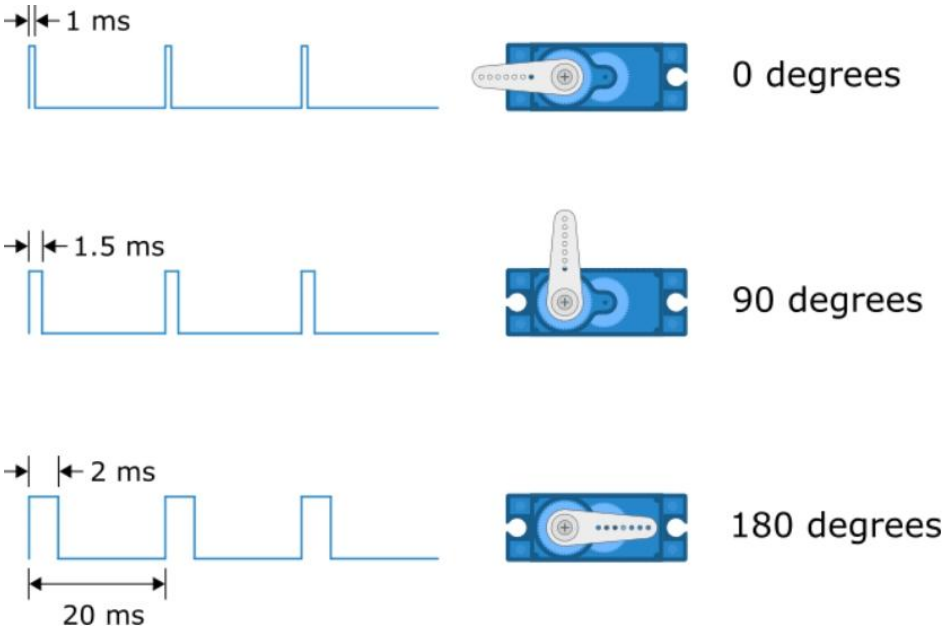
Example:

When the pulse width is 1ms, the output shaft of the servo will move to the smallest position (0 degrees);

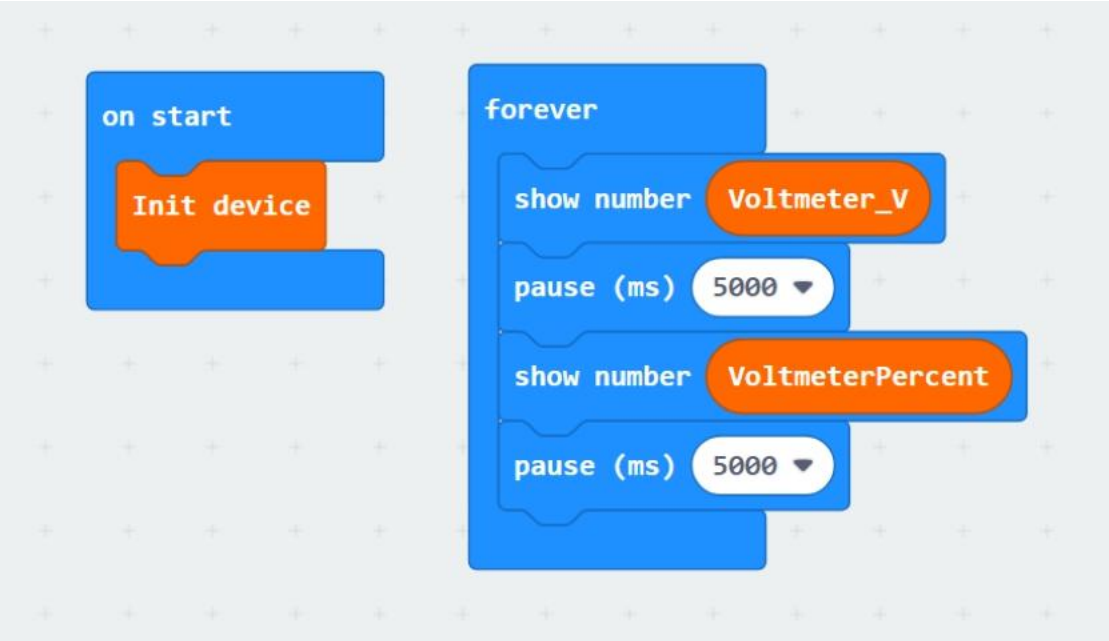
When the pulse width is 1.5ms, the output shaft of the servo will move to the middle position (90 degrees);

When the pulse width is 2ms, the output shaft of the servo will move to the smallest position

(180degrees)



9_Voltmeter example

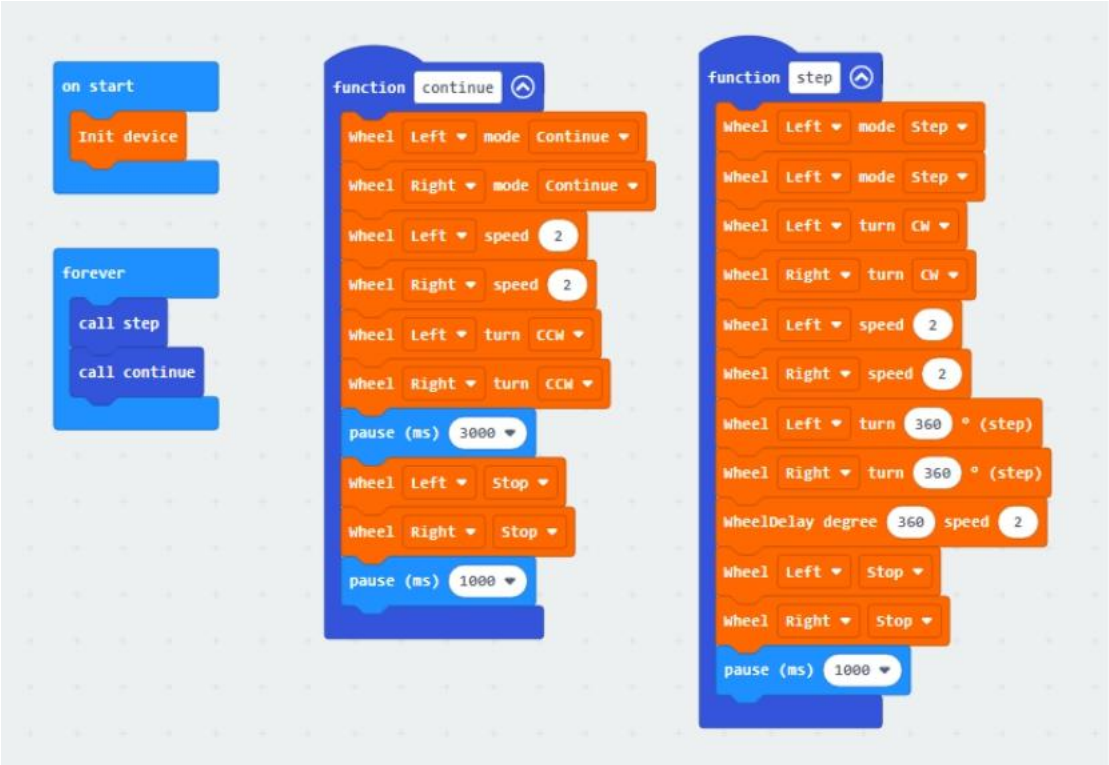


A voltmeter is designed on Cobit to read battery voltage and voltage percentage. The voltage percentage is a mapping value:

Range of Voltage	Mapped Voltage Percentage
6.5V---8.4V	0---100%

After uploading the code, put two 18650 lithium batteries into the battery box of the Cobit, turn the power switch to the "ON" state, the 5*5LED dot matrix of the microbit motherboard will print the voltage value and voltage percentage of the two lithium batteries in series out.

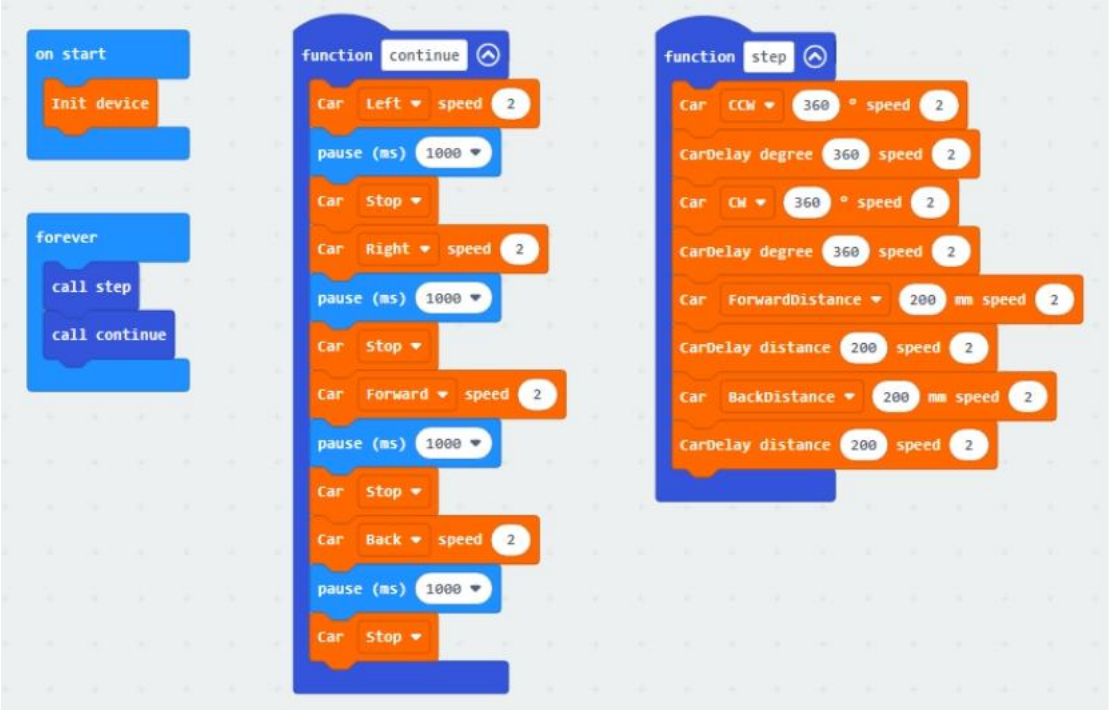
10_Wheels Example



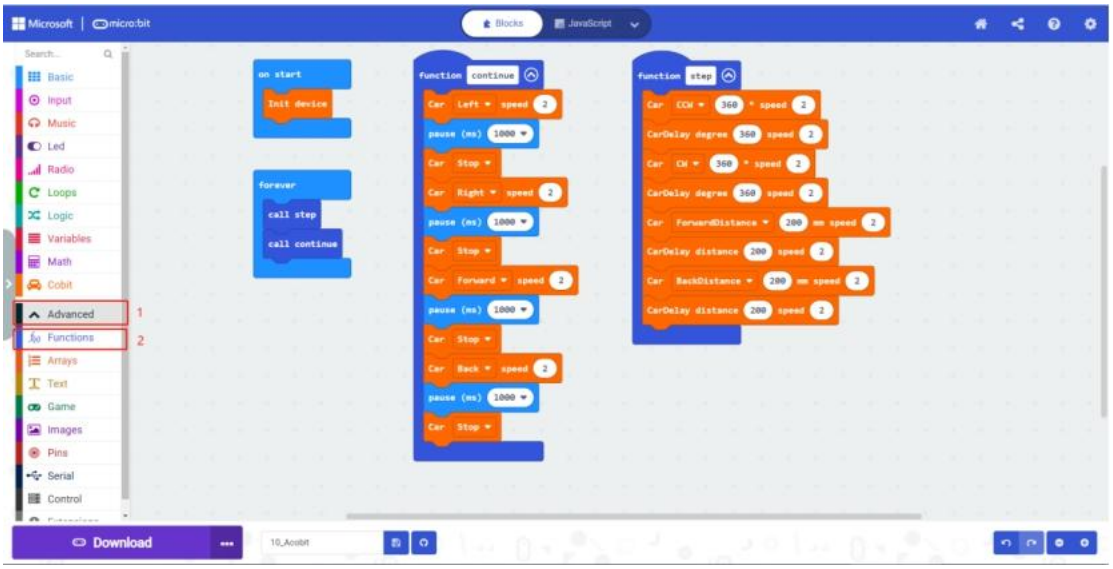
There is a wheel with a diameter of 65mm on the left and right sides of the Cobit. Two stepper motors are used as the power of the wheel. The rotation angle and the rotation arc of the wheel can be controlled. The rotation angle accuracy is +/-0.9 degrees, and the rotation speed is divided into 0-- -4 gears (0 = 0rpm, 1 = 15rpm, 2 = 30rpm, 3 = 60rpm, 4 = 120rpm).

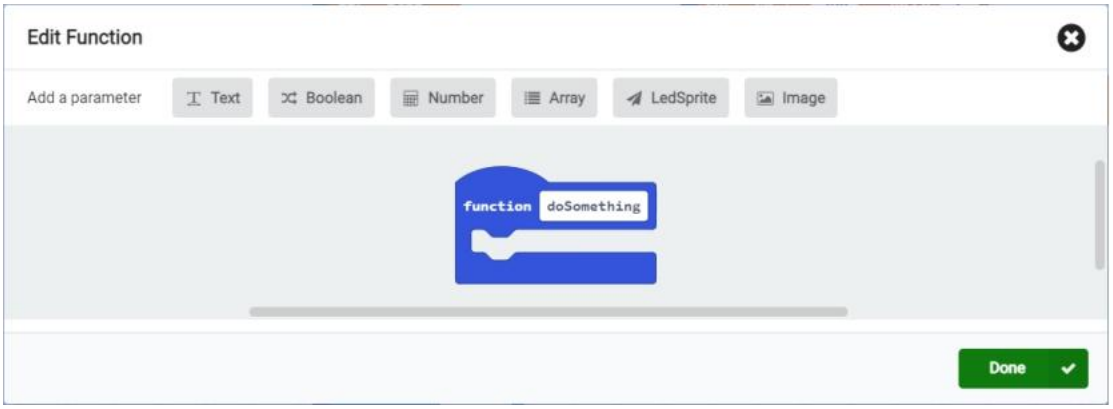
After uploading the code, the two motors on the left and right of Cobit first rotate 360 degrees clockwise, and then rotate counterclockwise for 3 seconds, and then continue like this.

11_Basic operation of Cobit



How to create the makecode function:：

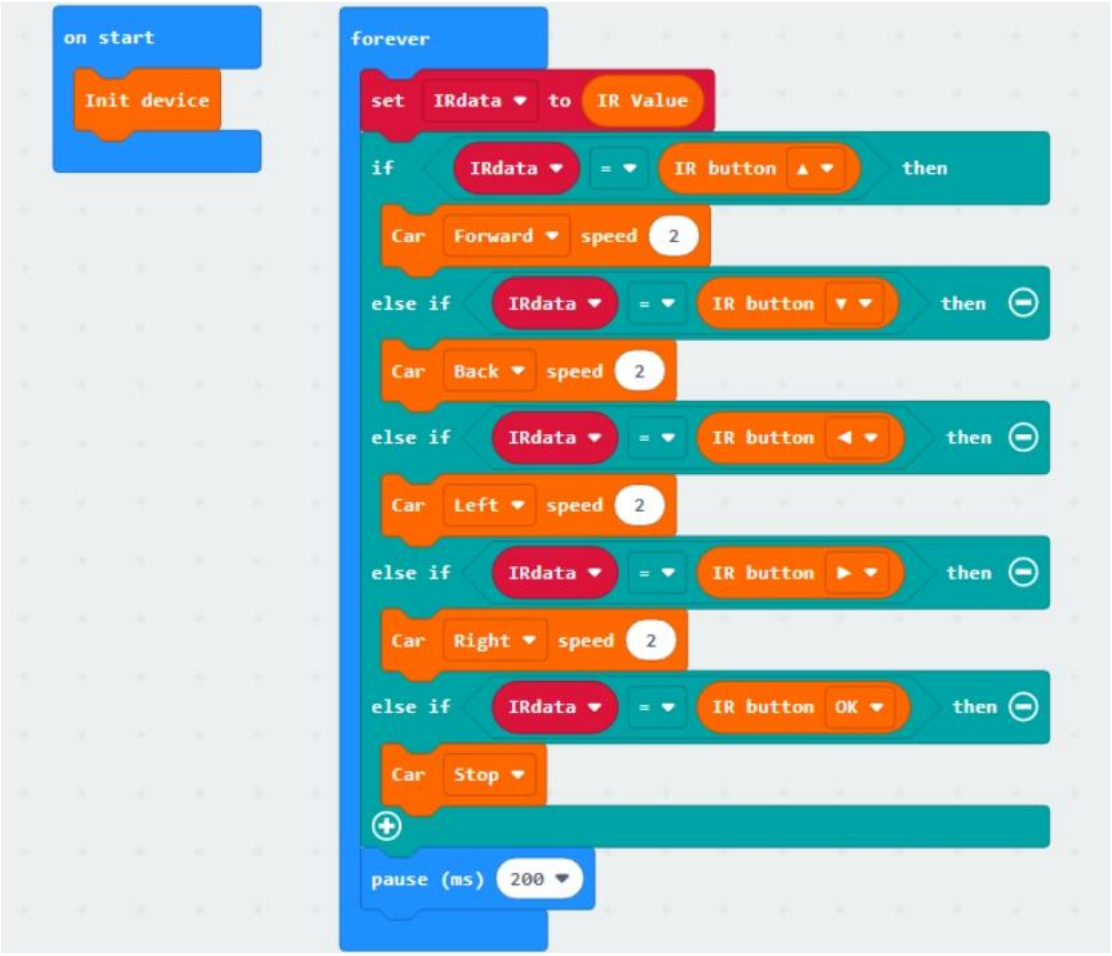


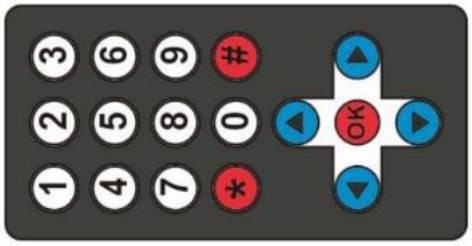


This example includes the basic driving operations of Cobit, such as the steering, driving direction of the car and their corresponding delay functions.

After uploading the code, Cobit will turn 360 degrees, turn right 360 degrees, drive 200mm forward, and then reverse 200mm; then turn left continuously for 2 seconds, turn right continuously for 2 seconds, and drive forward for 1 second, then Go back for 1 second, and then keeps like this.

12_Infrared remote control car



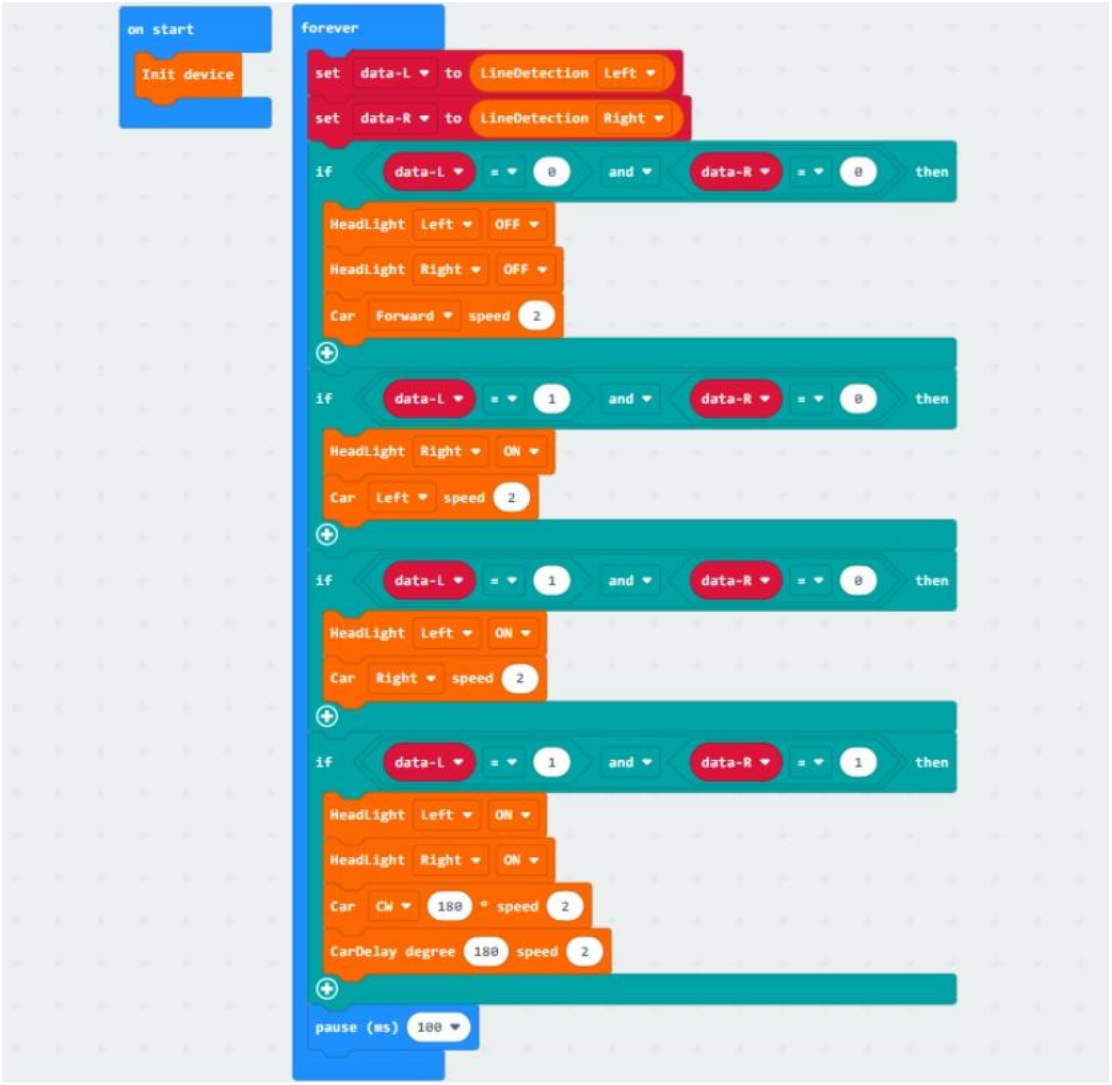


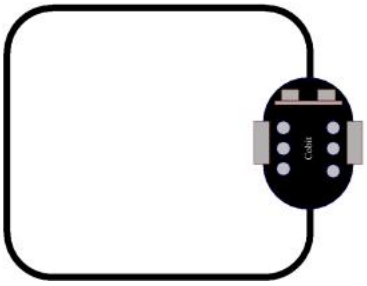
Cobit is equipped with an infrared remote control by default, and each button has a different key value. We give each key value a different function, and then send the key value to Cobit through the remote control, and Cobit obtains the key value and then implements the key Value function.

Infrared remote control key value:

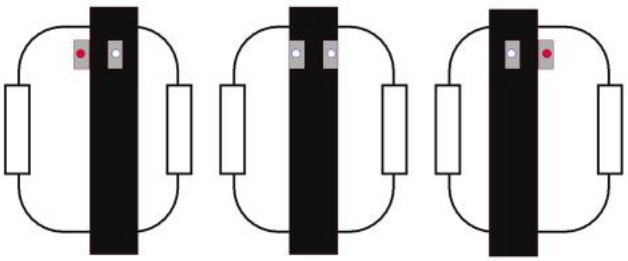
Key	0	1	2	3	4	5	6	7	8
Key Value	0x67	0x5D	0x9D	0x1D	0xDD	0xFD	0x3D	0x1F	0x57
Function	null	null	null	null	null	null	null	null	null
Key	9	*	#	▲	▼	◀	▶	OK	
Key Value	0x6F	0x97	0x4F	0xE7	0xB5	0xEF	0xA5	0xC7	
Function	null	null	null	forward	back	turn left	turn right	stop	

13_Line Detection Car



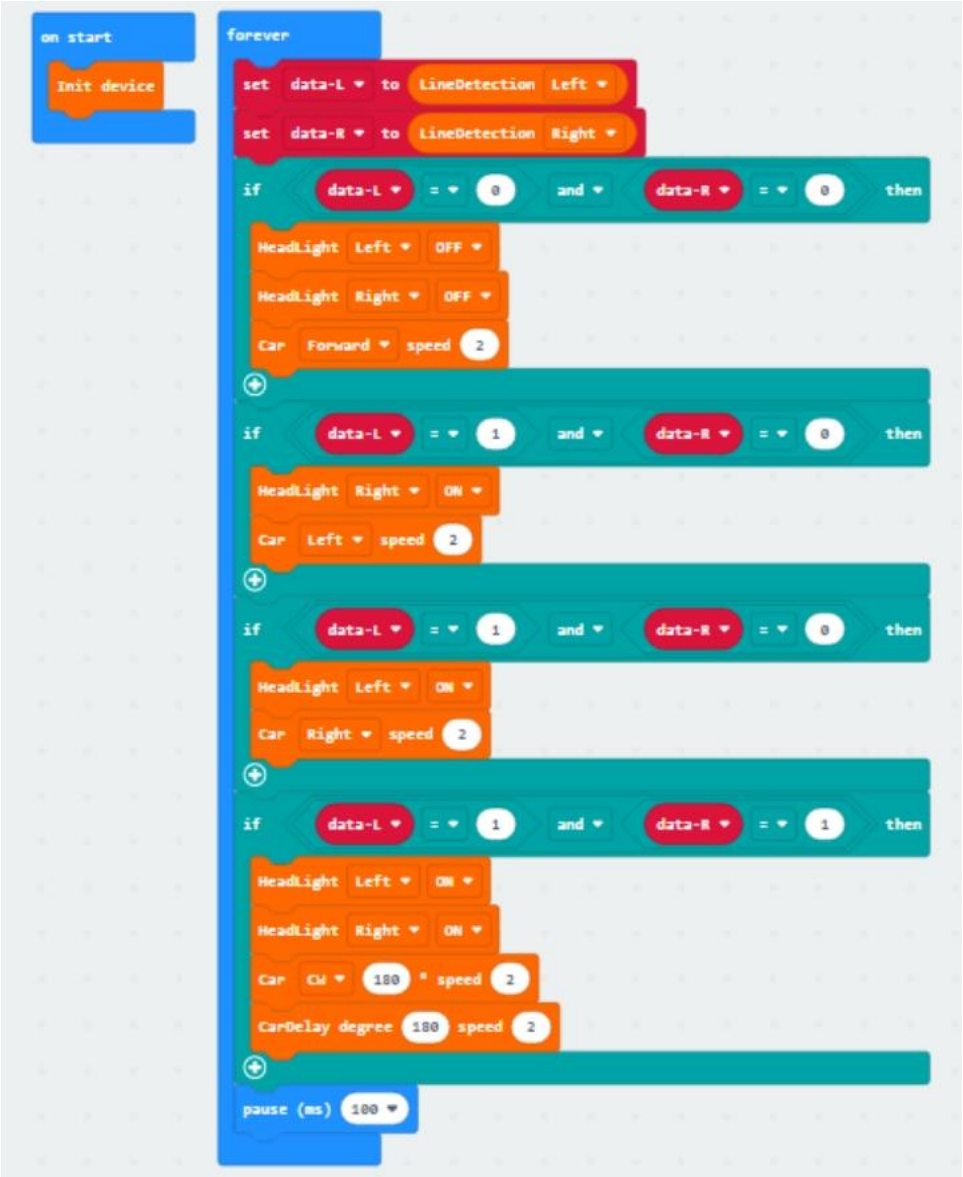


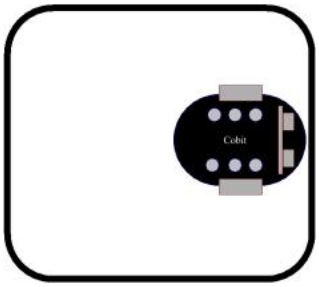
The car is with function of the infrared line Detection. Use 2 cm wide black paper cloth on the white black paper to form a line of track shape. Black objects have the effect of absorbing infrared rays, and white objects have the effect of reflecting infrared rays. Use this function You can keep the car walk along the black line.



is “0”, is “1” .

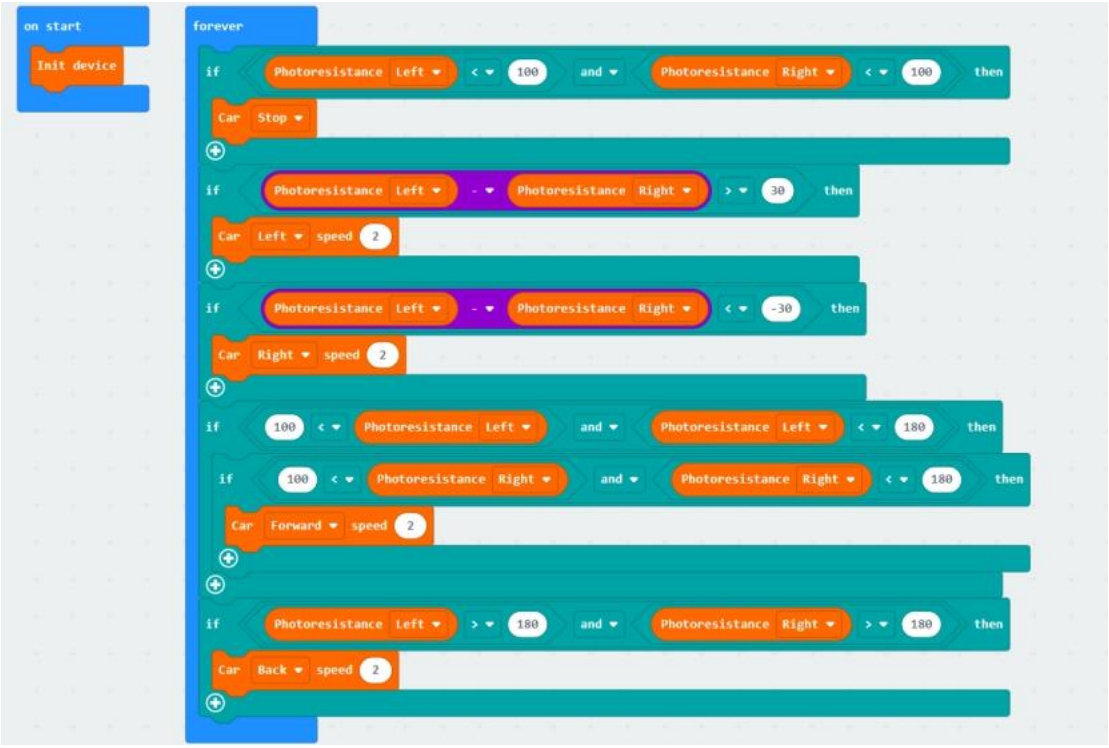
14_Line Detection Restricting

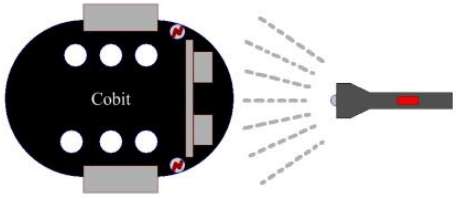




Line Detection Restricting function realized by infrared line Detection sensor. After burning the code, stick a square with black tape on the white paper, and put the car into the square. When the car is driving in the square, the two infrared line patrol sensors keep receiving the reflected signal, it means that the car is still on the desktop. When the car hits the black line, the line-following sensor does not form a reflection signal, and the car immediately brakes to stop, and then turns direction to travel.

15_PR Car

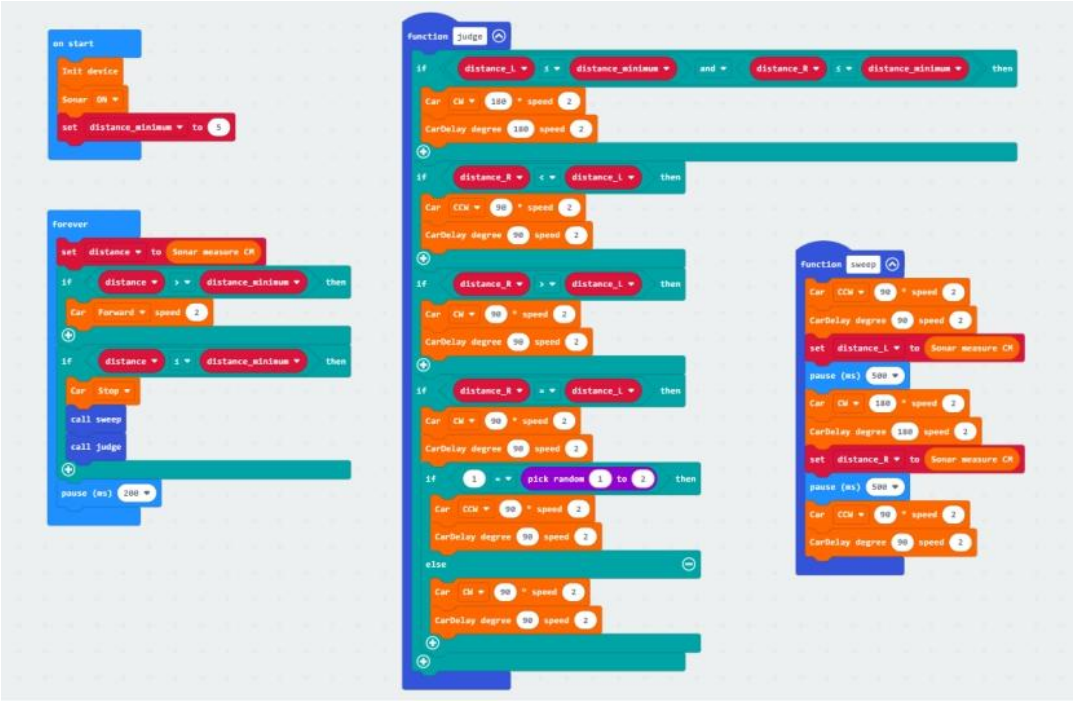




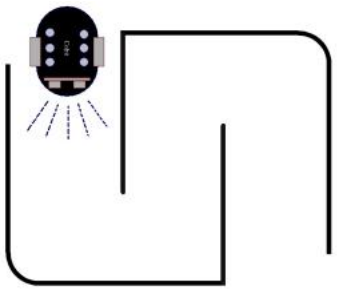
A photoresistor sensor is integrated on the left and right sides of the Cobit, which can be used to determine the brightness of the light on both sides of the Cobit, and then the motor rotates to make the light brightness obtained by the photoresistor sensors on both sides of the Cobit basically the same, so that the Cobit always keeps facing the light source. Follow the light source.

Note: In order to avoid the influence of other light, please carry out this experiment in a dark place.

16_Sonar Car(ultrasonic module)



Remarks: The examples use functions. For specific implementation, please refer to the 11th example in Chapter 8.7.

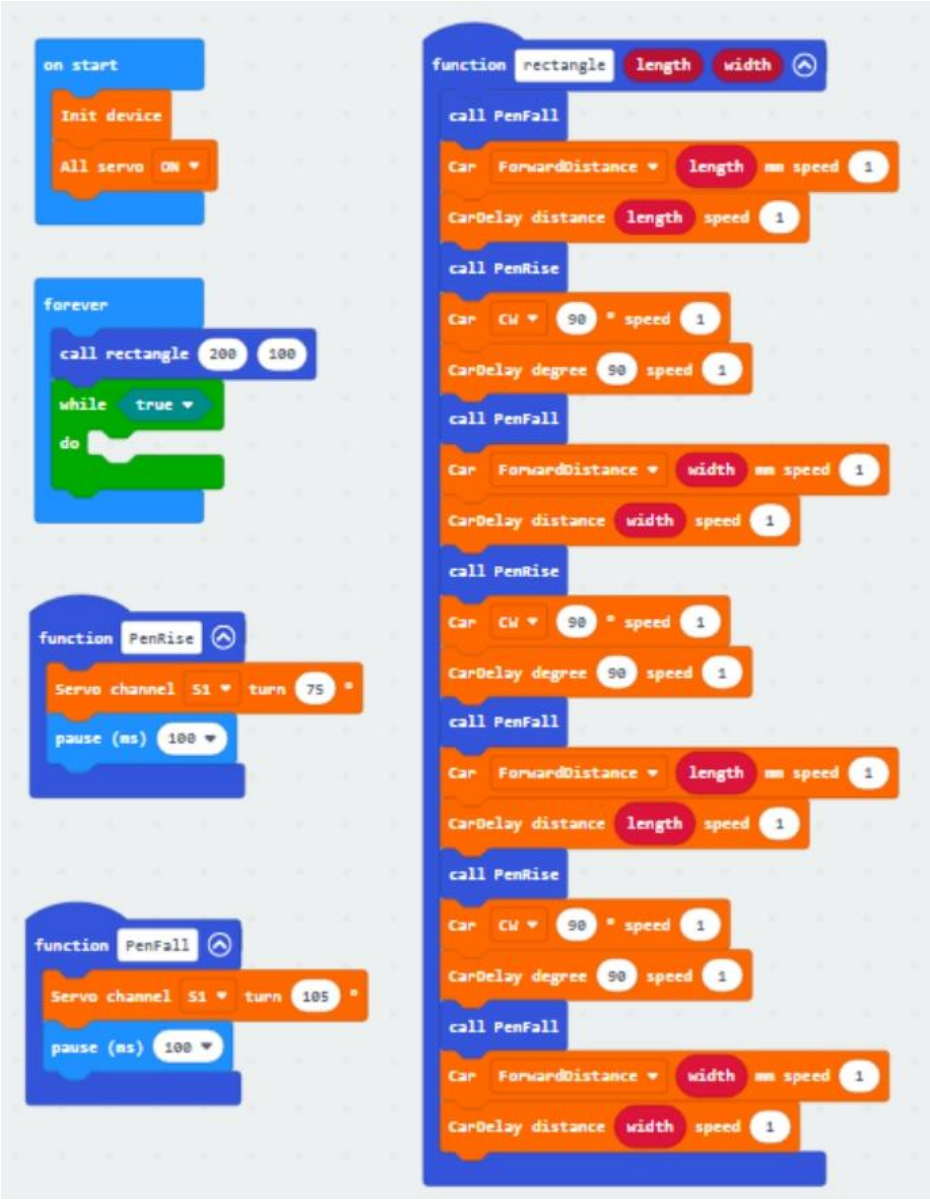


We provide the ultrasonic module for the Cobit car by default. As long as the module is inserted into the ultrasonic module interface of the Cobit, the Cobit has the

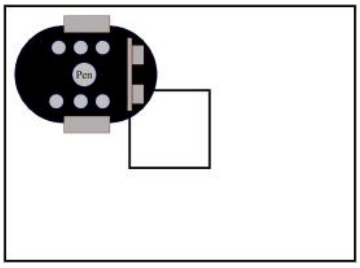
function of measuring distance.

The function of this code is to use the ultrasonic sensor on the front end of the Cobit to measure the distance of the obstacle in front. If the distance is less than the distance set by the program, the car will stop immediately, and then turn to measure the distance of the obstacles on the left and right sides, and select the farthest distance and greater than the minimum distance If the left and right distances are less than the minimum distance, the car will make a U-turn.

17_Drawing Car



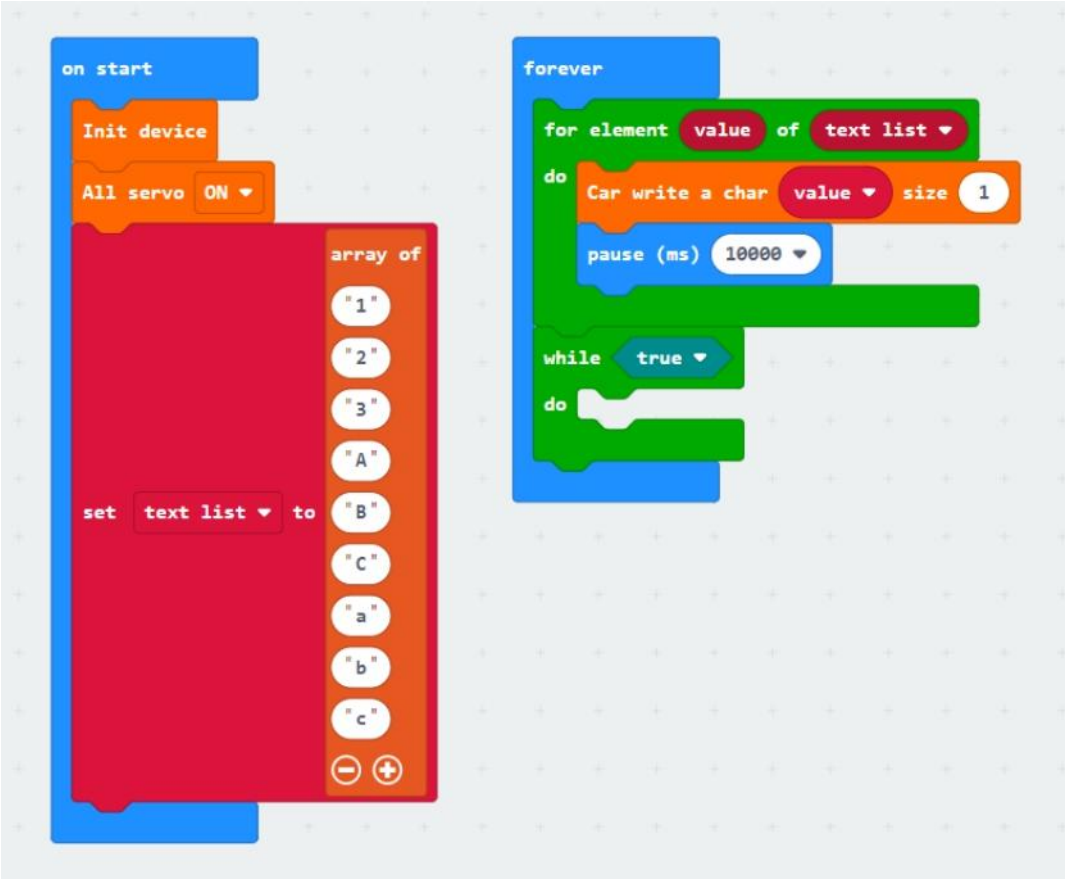
Remarks: The examples use functions. For specific implementation, please refer to the 11th example in Chapter 8.7.



This function of the Cobit is completed in conjunction with Lego accessories. There are standard Lego mounting holes reserved on the car body. As long as the pen holder structure is built according to the assembly instructions, the Cobit can paint then.

Burn the code into the Cobit and place the Cobit on the upper left corner of the A4 paper. The cart will draw a rectangle on the paper. If you need more shapes, you can refer to the code programming.

18_Writing Car



This example is completed with Lego accessories. The car body has standard Lego mounting holes. You need to build the Lego pen holder structure according to the assembly instructions to enable the Cobit to have the writing function.

Burn the code into the Cobit, stitch together 3 sheets of A4 paper, put the Cobit on the upper left corner of the first A4 paper, the car will write "123ABCabc" on the paper, you can also fill in the sample code string you need.

19_Bluetooth Port Example

Micro:bit V1 integrates a Bluetooth 4.1 stack, Micro:bit V2 integrates a Bluetooth 5.0 stack, and the communication frequency band is 2.4G---2.41G, which enables the micro:bit motherboard to communicate with the Bluetooth of the mobile phone to achieve wireless control functions.

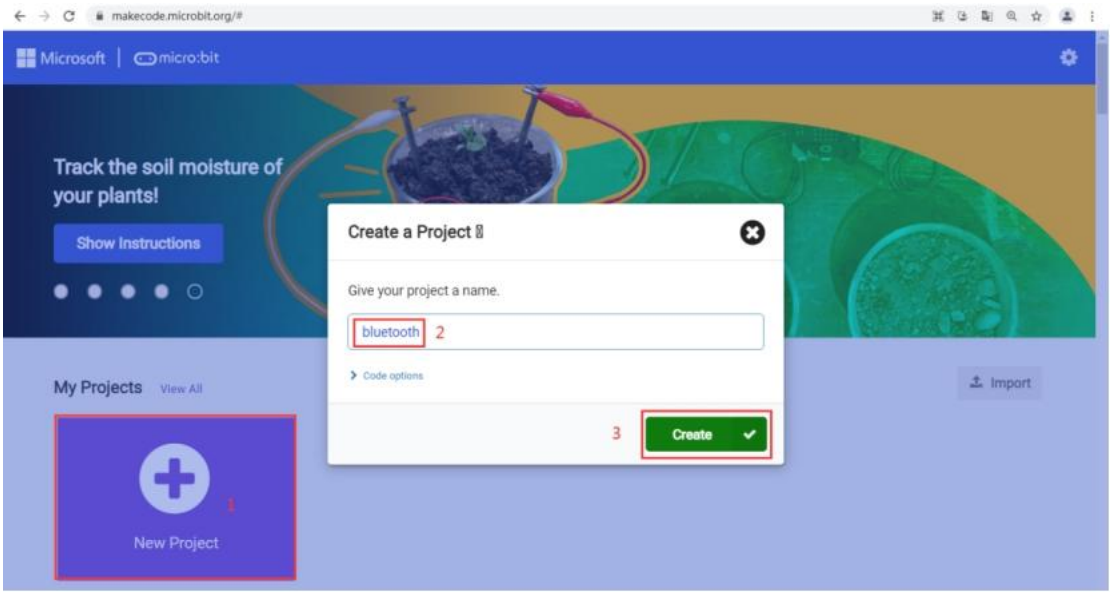
Bluetooth hardware technical documents:
https://tech.microbit.org/bluetooth/

Bluetooth syntax usage tutorial:
https://makecode.microbit.org/reference/bluetooth

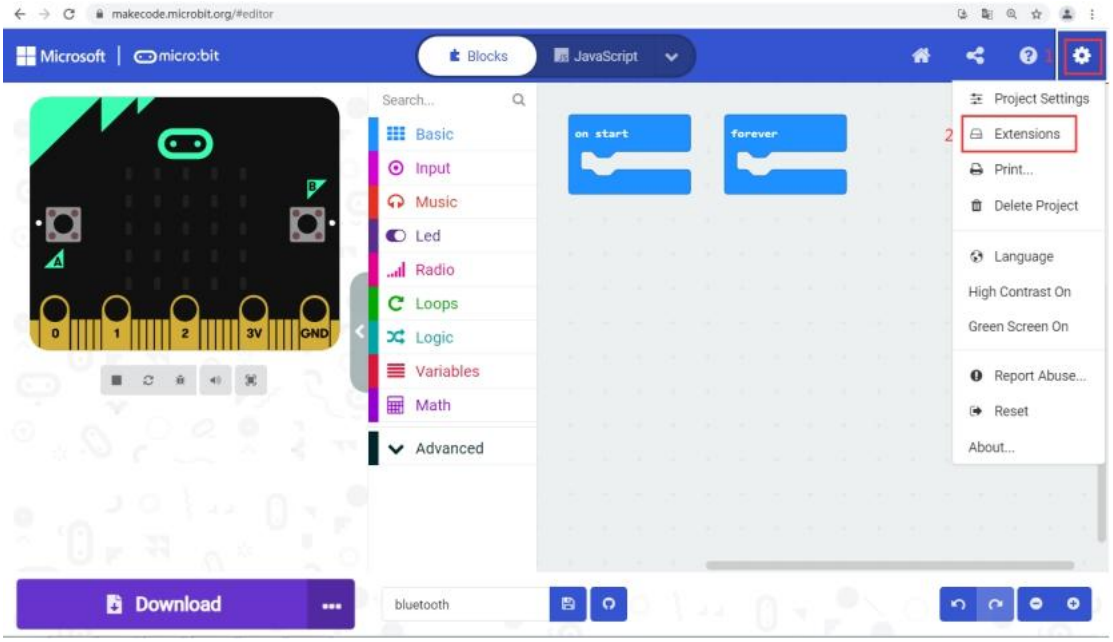
Tutorial video of connecting Bluetooth official website:
https://makecode.microbit.org/reference/bluetooth/on-bluetooth-connected

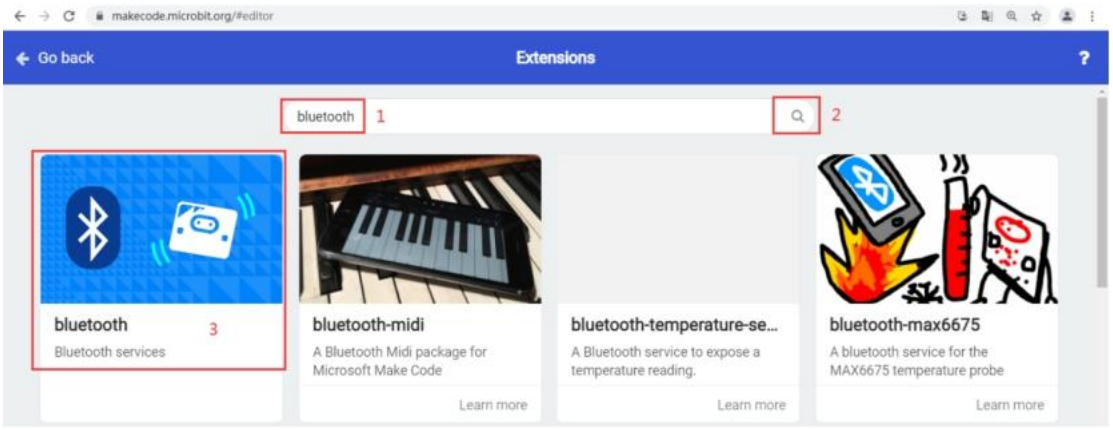
BitApp is paired with Micro:bit V1/2

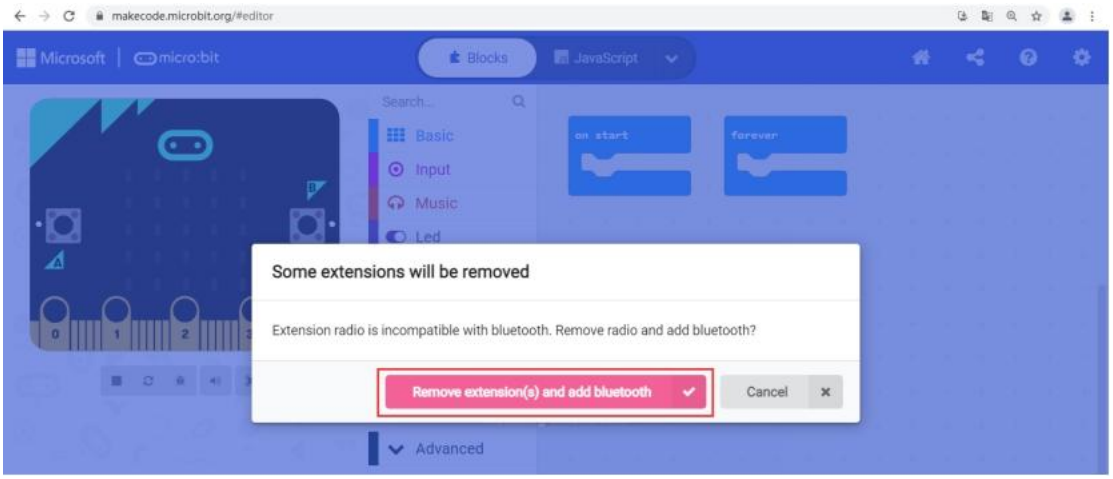
Open https://makecode.microbit.org Create a new online project:



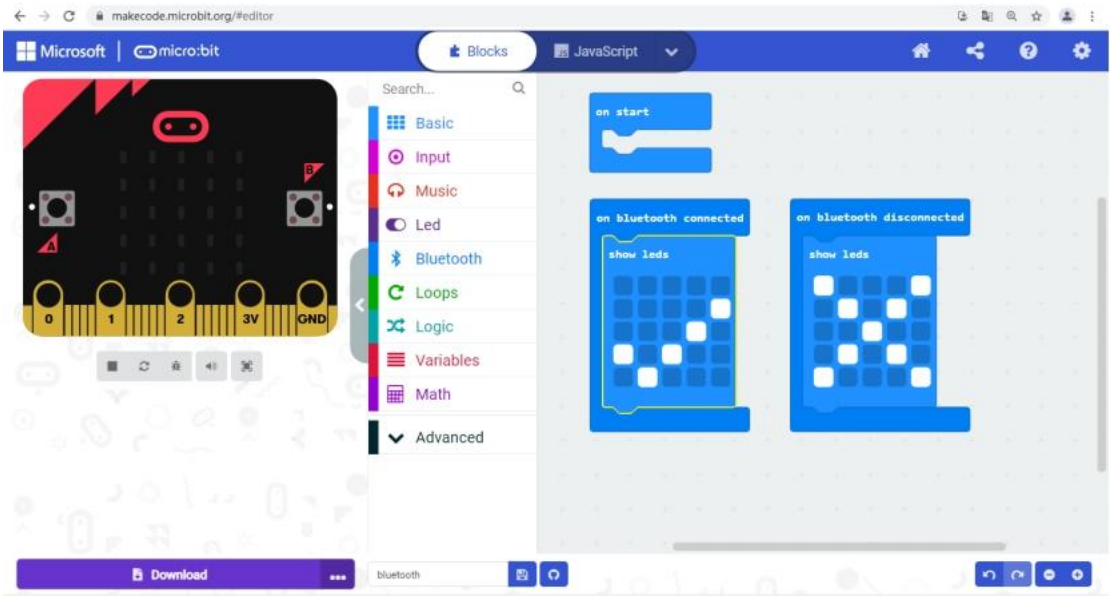
After creating a new project, you need to import the Bluetooth extension package. Because the Bluetooth expansion board and the micro:bit default "Radio" expansion are not compatible with the micro:bit motherboard at the same time, you will be prompted to delete the "Radio" expansion board during the process of adding the Bluetooth expansion package. Please confirm the deletion at this time (if you need to use the "Radio" expansion package can also be added back).



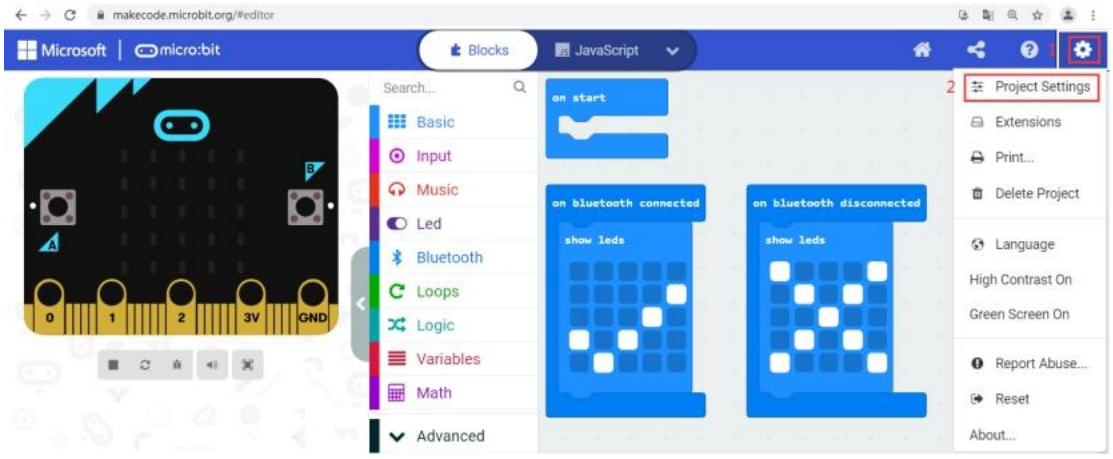


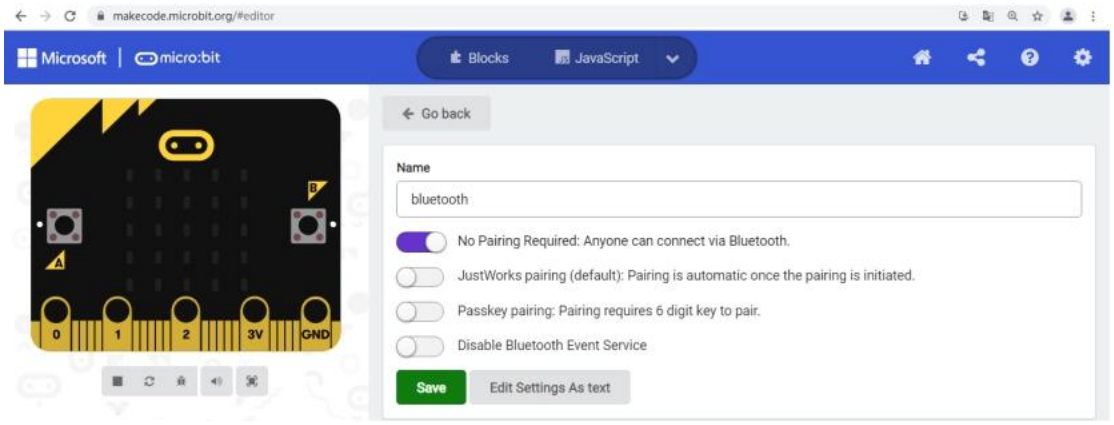


After installing the Bluetooth expansion package, you can use the Bluetooth function of the micro:bit normally. Here we then program to display " √ " on the micro:bit 5*5 LED dot matrix when the mobile phone is paired with the Bluetooth of the micro:bit.



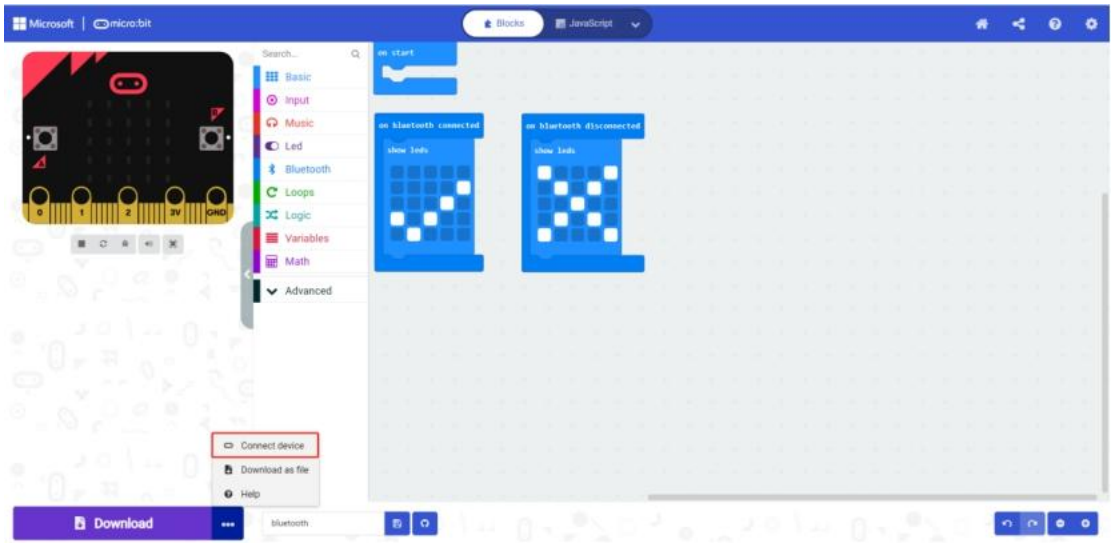
The project has 4 working modes, we need to select the required working mode according to the actual needs of the project, here we select the "No Pairing Required: Anyone can connect via Bluetooth." mode. and then save.

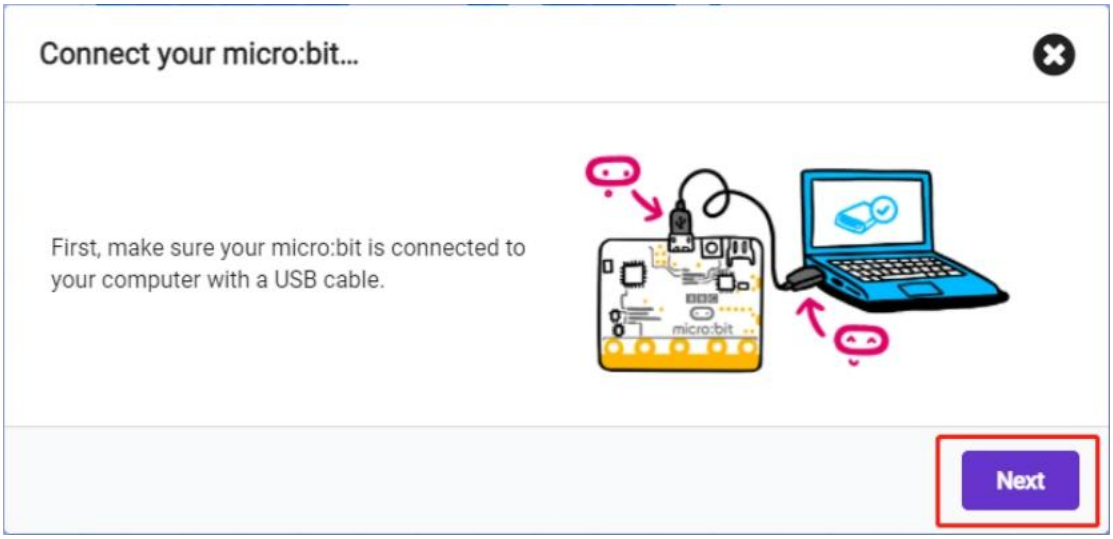


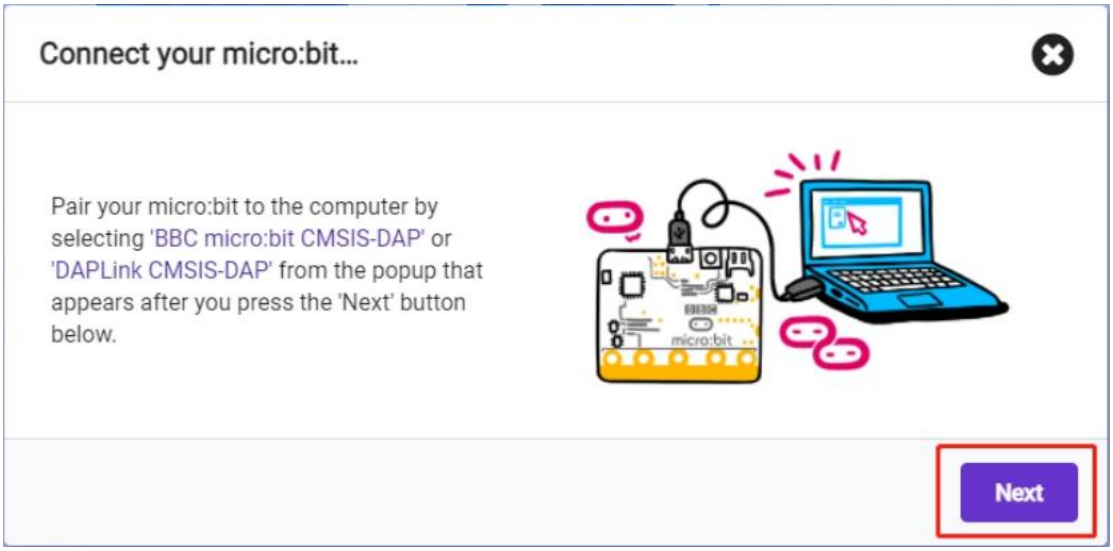


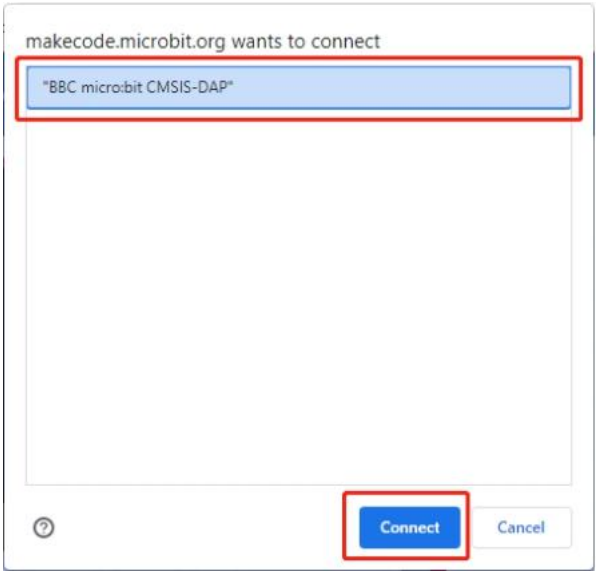
After setting the mode, connect the micro:bit device, then generate the HEX file from the project, and upload the HEX file to the micro:bit. There are two ways to upload the HEX file of the micro:bit.

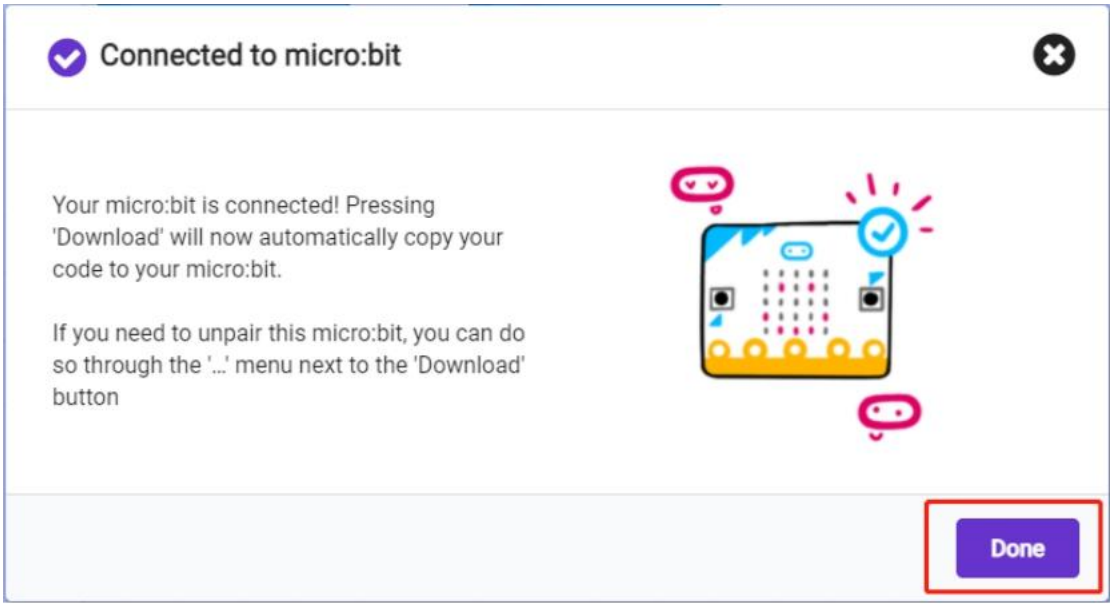
Method 1 (systems above windows8)

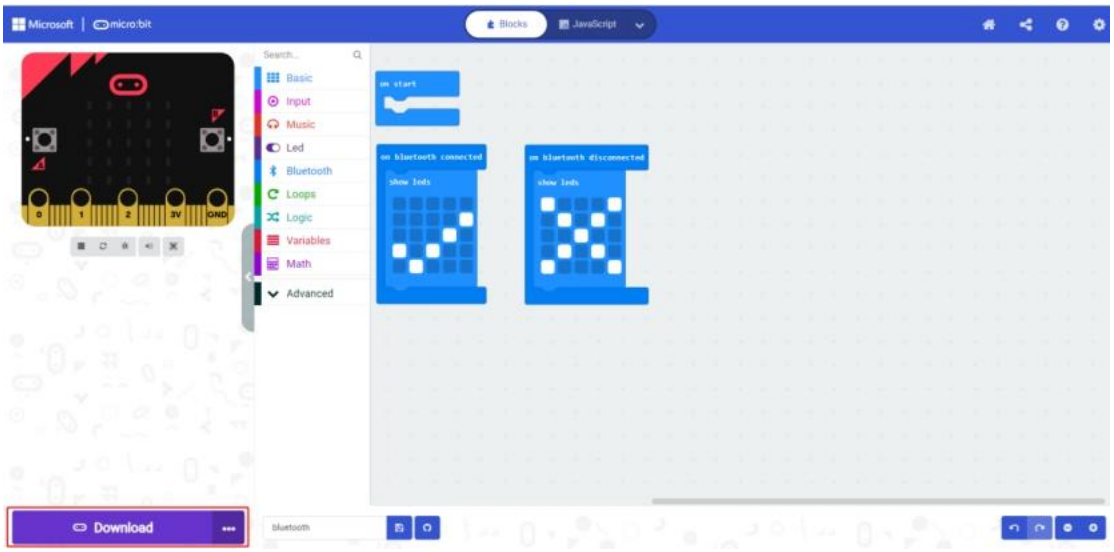




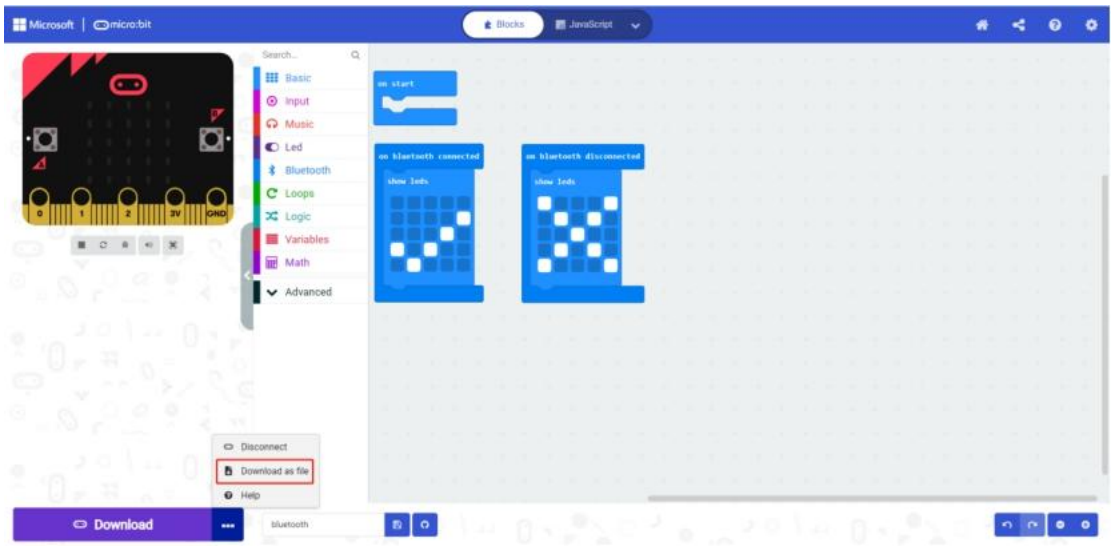


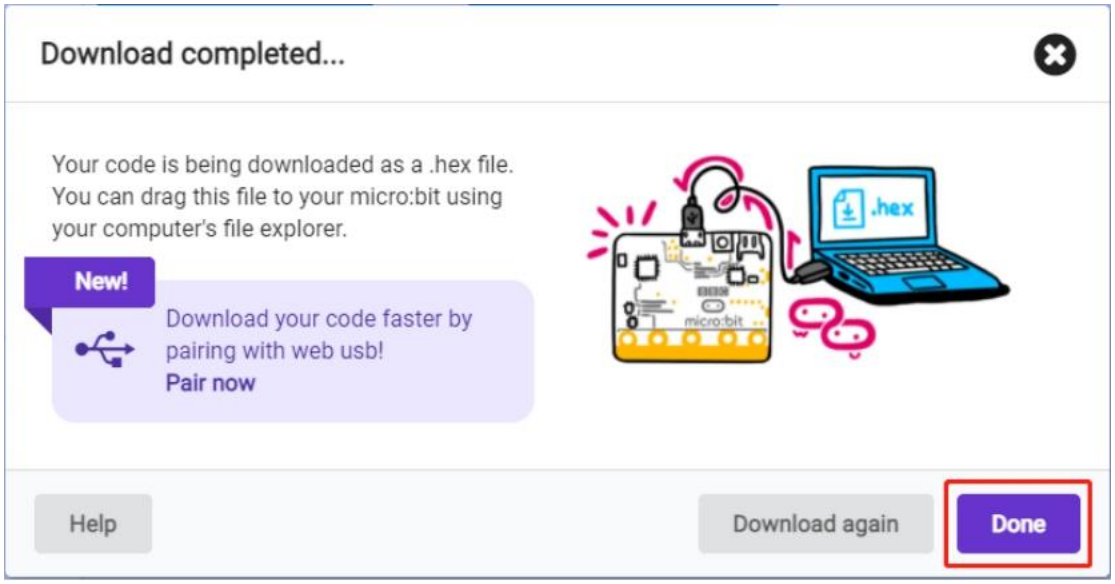

makecode.microbit.org wants to connect
"BBC micro:bit CMSIS-DAP"
Connect
Cancel

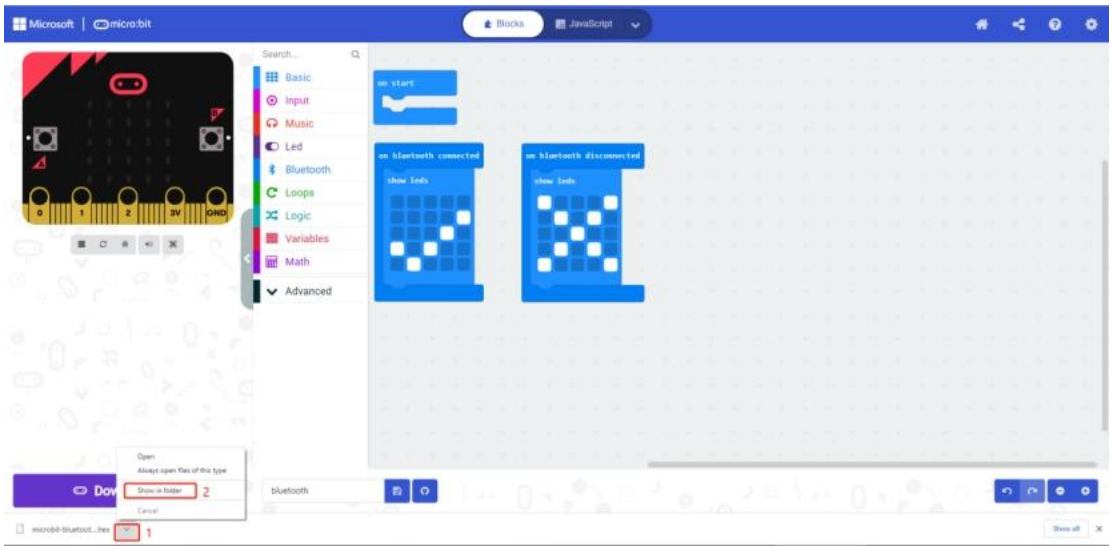

Connected to micro:bit
Your micro:bit is connected! Pressing 'Download' will now automatically copy your code to your micro:bit.
If you need to unpair this micro:bit, you can do so through the '...' menu next to the 'Download' button
Done

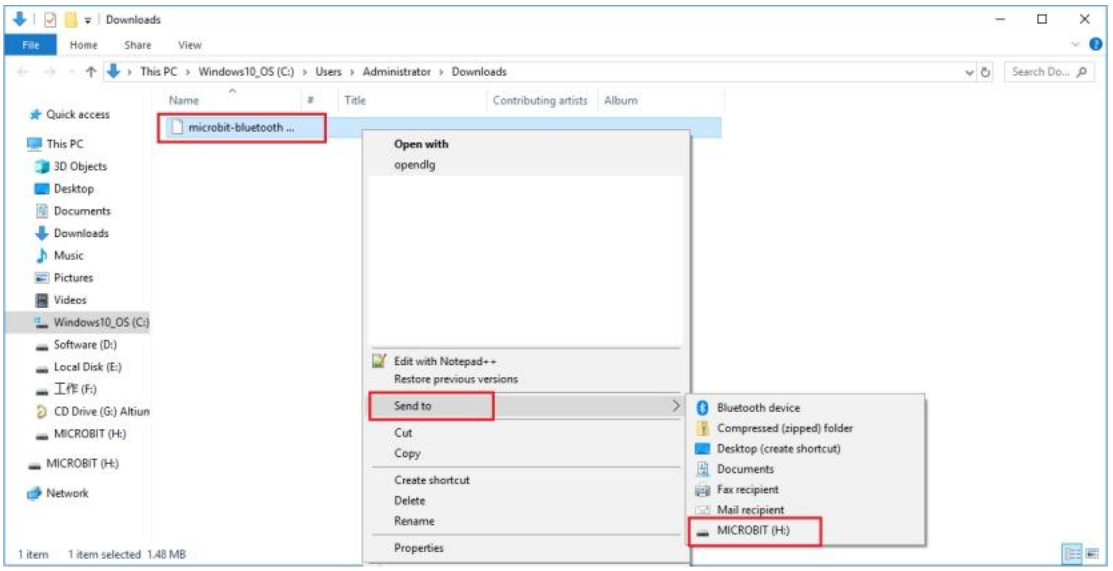

Microsoft
micro:bit
Blocks
JavaScript
Search...
Basic
Input
Music
Led
Bluetooth
Loops
Logic
Variables
Math
Advanced
on start
on bluetooth connected
show leds
on bluetooth disconnected
show leds
Download
bluetooth

Method 2


Microsoft | micro:bit
Blocks
JavaScript
Search...
Basic
Input
Music
Led
Bluetooth
Loops
Logic
Variables
Math
Advanced
on start
on bluetooth connected
show leds
on bluetooth disconnected
show leds
Disconnect
Download as file
Help
Download
bluetooth

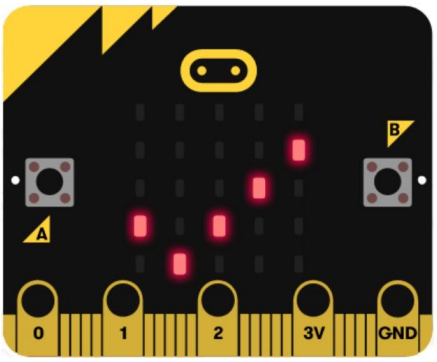

Download completed...
Your code is being downloaded as a .hex file.
You can drag this file to your micro:bit using your computer's file explorer.
New!
Download your code faster by pairing with web usb!
Pair now
.hex
micro:bit
Help
Download again
Done


Microsoft | micro:bit
Blocks
JavaScript
Search...
Basic
Input
Music
Led
Bluetooth
Loops
Logic
Variables
Math
Advanced
on start
on bluetooth connected
show leds
on bluetooth disconnected
show leds
bluetooth



Then open BitApp, search and pair Bluetooth (for more details, please refer to the "BitApp" chapter).

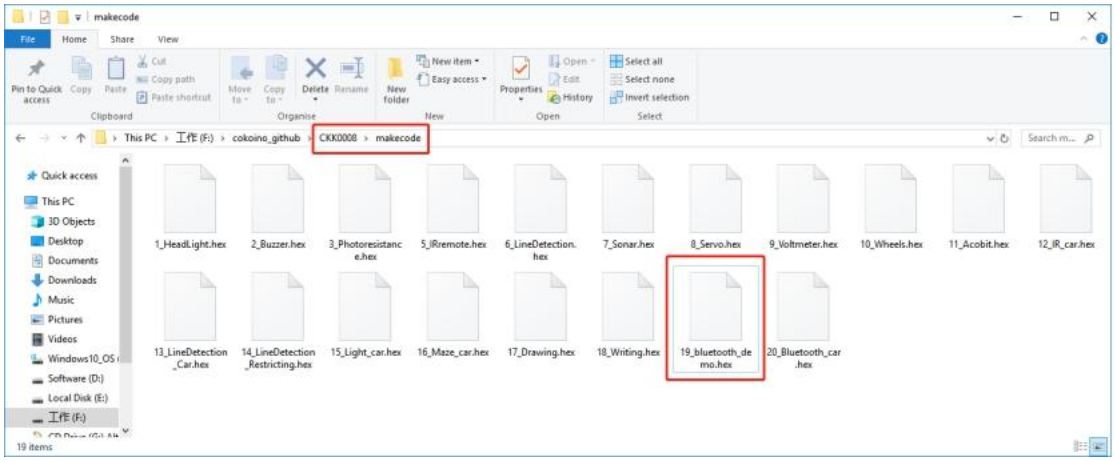
When the pairing is successful, the 5*5 LED dot matrix of the micro:bit displays " √ ".

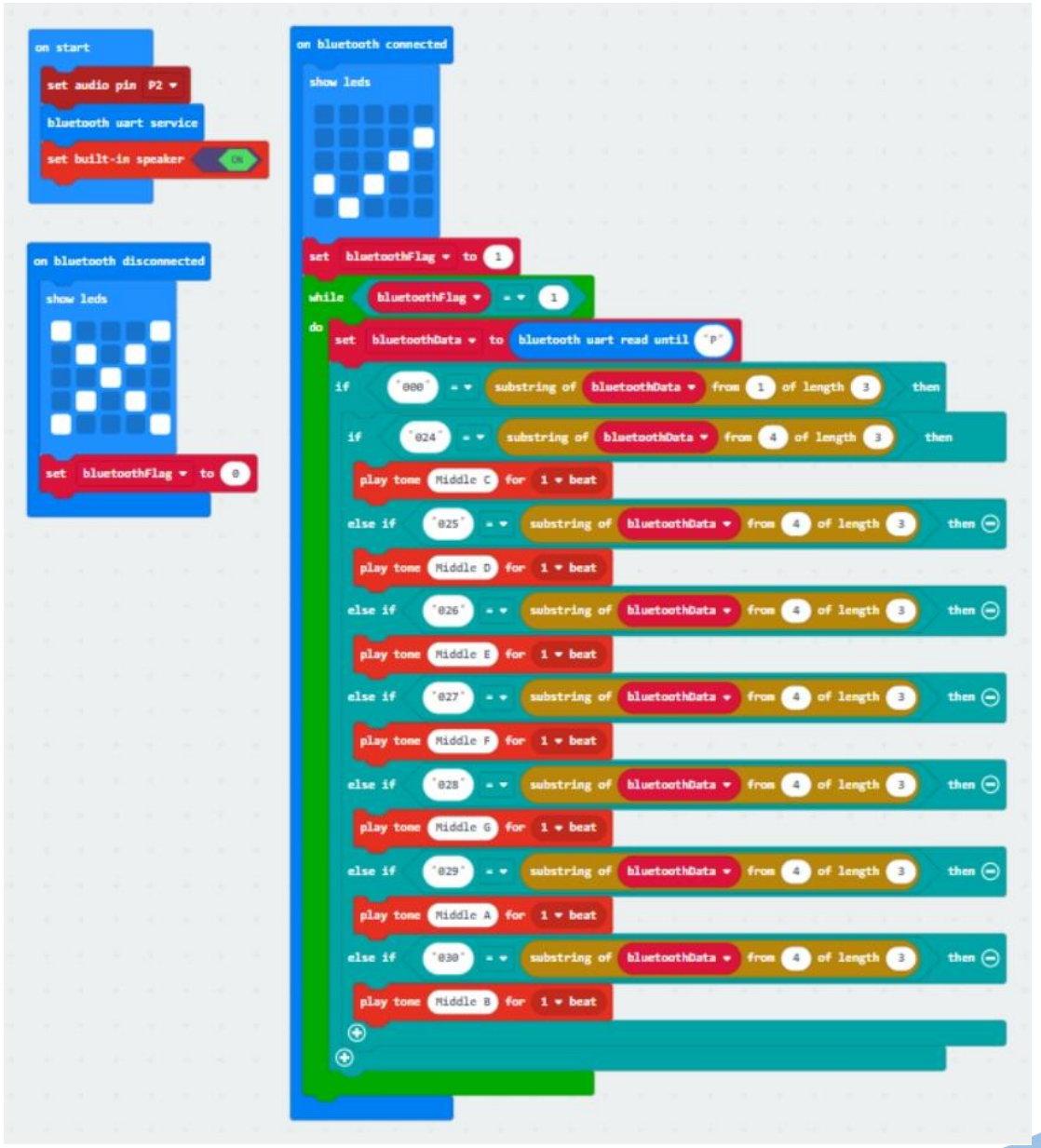


We also have a simple communication example. The code is stored in the "makecode" folder. Please copy the file to the micro:bit motherboard, then connect the Bluetooth of the micro:bit motherboard, and select the control method under

the Cobit image in BitApp. Click to switch to the music keyboard, and then click the music keyboard to make Cobit sing.

Sample code:





Connect to Bluetooth and select the operation mode of the Cobit icon:

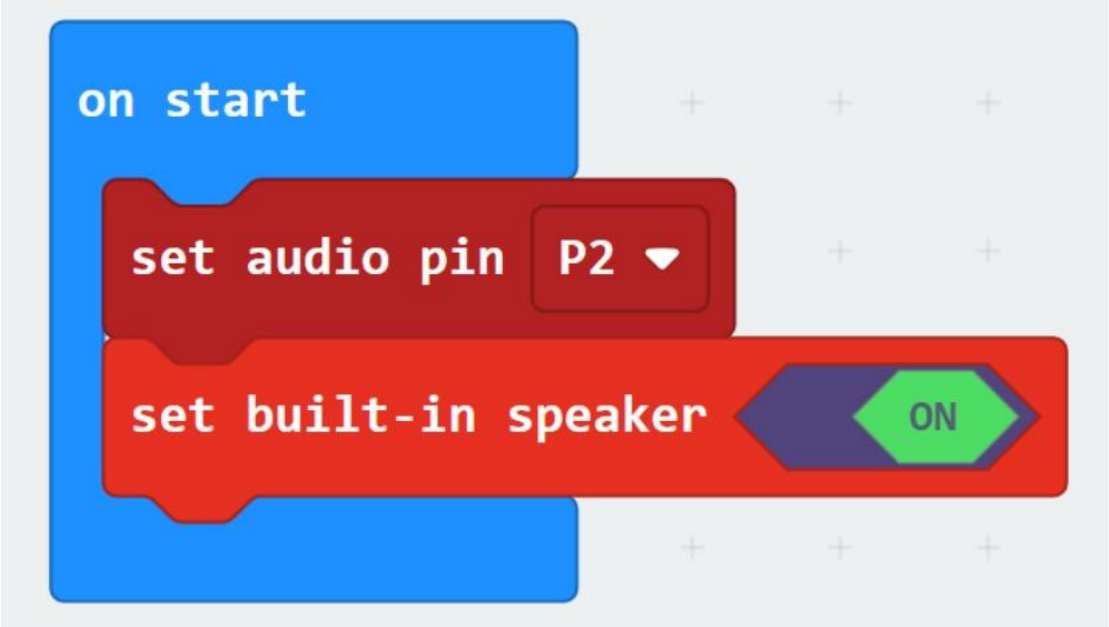
Send key value:

Music keyboard:

SN	1	2	3	4
Function	do	re	mi	fa
SN	5	6	7	
Function	sol	la	ti	

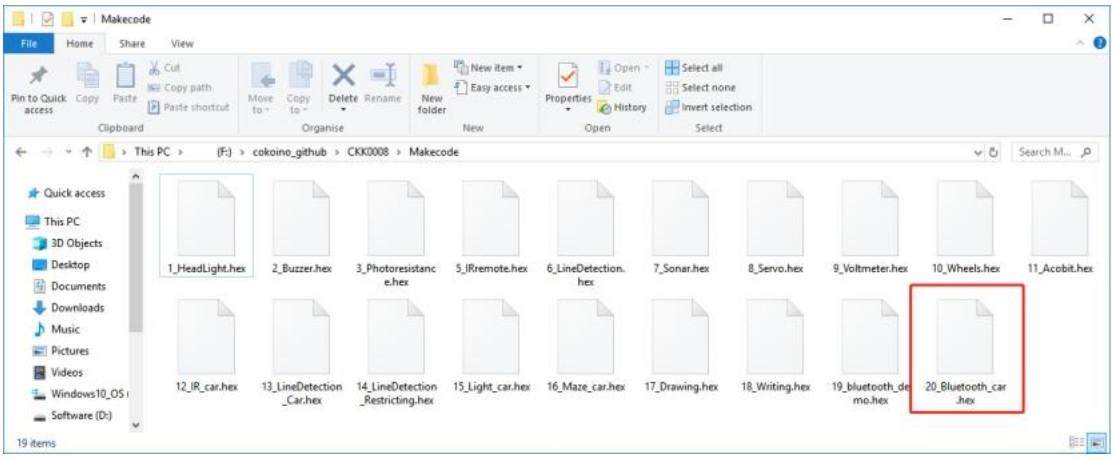
Precautions for using the buzzer on the micro:bit motherboard after adding the makecode cobit expansion package:

The expansion package is serial communication, using P0 and P1 ports, the pin of the buzzer is also the default P0 port, when using the buzzer, the pin of the buzzer must be mapped to other pins, otherwise it will affect the normal of the expansion package Use, and the buzzer switch is off by default, you need to turn it on, as shown in the figure below:

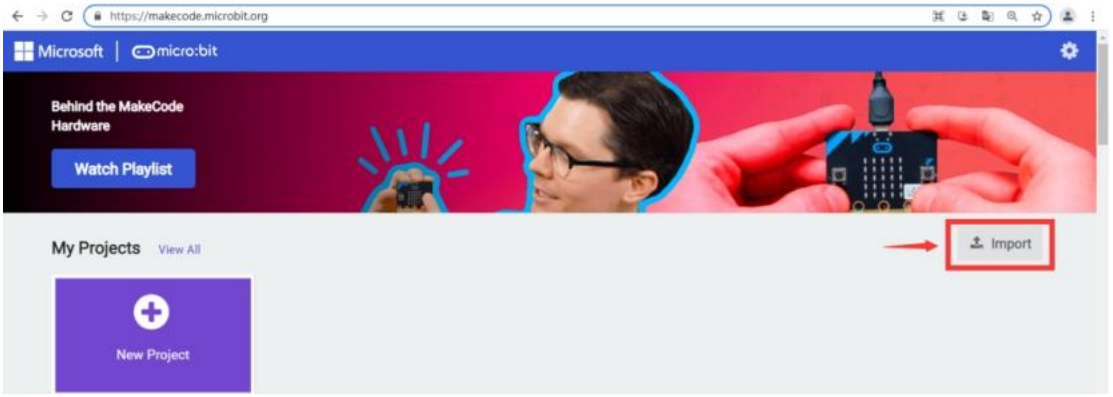


20_Bluetooth remote control car

This example is a comprehensive example. If the graphics code is too large, no screenshots will be taken. The code is stored in the "Makecode" folder:



To view the source code, please import the online programming webpage:



Use BitApp to connect to the Bluetooth on the micro:bit board, select the operation mode under the Cobit image, and then press the following key instructions to operate.

Key value and function:

SN	1	2	3	4	5
Function	back	forward	turn left	turn tight	return
SN	6	7	8	9	10
Function	stop	preserved	preserved	on/off headlight	speaker
SN	11	12	13	14	15
Function	on/off ultrasonic detecting car	preserved	preserved	preserved	preserved
SN	16	17	18	19	20
Function	preserved	Headlight brightness	toggle key	RGB color switch	music key

RGB color control page

SN	1	2	3
Function	red	green	blue
SN	4	5	6
Function	purple	Close all RGB lights	Sea green

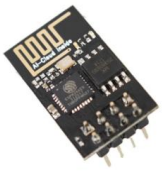
Music keyboard

SN	1	2	3	4
Function	do	re	mi	fa
SN	5	6	7	
Function	sol	la	ti	

9. Getting start with Cobit for esp wifi

The 21st century is the era of the Internet of Things. The Internet of Things has brought a lot of convenience to our lives. The esp-01 wifi port is reserved on the cobit, so that the cobit can access the Internet of Things.

9.1 ESP-01 wifi

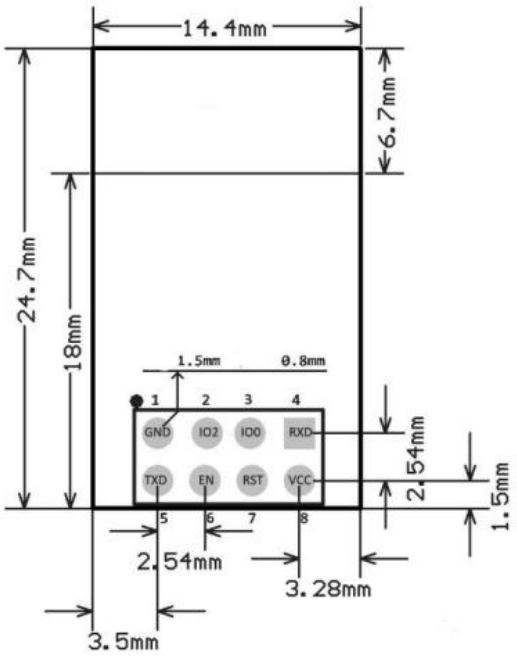


ESP-01 can be widely used in various IoT applications, suitable for home automation, industrial wireless control, baby monitors, wearable electronic products, wireless position sensing devices, wireless positioning system signals and other IoT applications. It's is ideal solution for the application of The Internet of Things.

Features

- The smallest 802.11b/g/n Wi-Fi SOC module

- Low power 32-bit CPU, can also serve as the application processor



- Up to 160MHz clock speed
- Supports UART/GPIO
- DIP-8 package for easy welding
- Integrated Wi-Fi MAC/BB/RF/PA/LNA
- Deep sleep current as low as 20uA
- Embedded lwIP protocol stack
- Supports STA/AP/STA + AP operation mode
- Support Smart Config/AirKiss technology
- UART baudrate up to 4Mbps
- General AT commands can be used quickly
- Supports remote firmware upgrade (FOTA)

9.2 The Usage of Arduino with ESP -01 wifi

The use of Arduino and ESP-01 is based on ESP-01 burning AT firmware (factory default), ATmega328P sends AT commands through software serial (D12=RX, D13=TX) to control ESP-01 to connect to wifi to send and receive data, so as to achieve Internet of things function.

Basic AT Commands

AT: Test AT startup.
AT+RST: Restart a module.
AT+GMR: Check version information.
AT+CMD: List all AT commands and types supported in current firmware.
AT+GSLP: Enter Deep-sleep mode.
ATE: Configure AT commands echoing.
AT+RESTORE: Restore factory default settings of the module.
AT+UART_CUR: Current UART configuration, not saved in flash.
AT+UART_DEF: Default UART configuration, saved in flash.
AT+SLEEP: Set the sleep mode.
AT+SYSRAM: Query current remaining heap size and minimum heap size.
AT+SYSMSG: Query/Set System Prompt Information.
AT+USERRAM: Operate user's free RAM.
AT+SYSFLASH: Query/Set User Partitions in Flash.

AT+RFPOWER: Query/Set RF TX Power.
AT+SYSROLLBACK: Roll back to the previous firmware.
AT+SYSTIMESTAMP: Query/Set local time stamp.
AT+SYSLOG: Enable or disable the AT error code prompt.
AT+SLEEPWKCFG: Query/Set the light-sleep wakeup source and awake GPIO.
AT+SYSSTORE: Query/Set parameter store mode.
AT+SYSREG: Read/write the register.

Wi-Fi AT Commands

AT+CWMODE: Set the Wi-Fi mode (Station/SoftAP/Station+SoftAP).
AT+CWSTATE: Query the Wi-Fi state and Wi-Fi information.
AT+CWJAP: Connect to an AP.
AT+CWRECONNCFG: Query/Set the Wi-Fi reconnecting configuration.
AT+CWLAPOPT: Set the configuration for the command AT+CWLAP.
AT+CWLAP: List available APs.
AT+CWQAP: Disconnect from an AP.
AT+CWSAP: Query/Set the configuration of an ESP SoftAP.
AT+CWLIF: Obtain IP address of the station that connects to an ESP SoftAP.
AT+CWQIF: Disconnect stations from an ESP SoftAP.
AT+CWDHCP: Enable/disable DHCP.
AT+CWDHCPS: Query/Set the IP addresses allocated by an ESP SoftAP DHCP server.
AT+CWAUTOCONN: Connect to an AP automatically when powered on.
AT+CWAPPROTO: Query/Set the 802.11 b/g/n protocol standard of SoftAP mode.
AT+CWSTAPROTO: Query/Set the 802.11 b/g/n protocol standard of station mode.
AT+CIPSTAMAC: Query/Set the MAC address of an ESP station.
AT+CIPAPMAC: Query/Set the MAC address of an ESP SoftAP.
AT+CIPSTA: Query/Set the IP address of an ESP station.
AT+CIPAP: Query/Set the IP address of an ESP SoftAP.
AT+CWSTARTSMART: Start SmartConfig.
AT+CWSTOPSMART: Stop SmartConfig.
AT+WPS: Enable the WPS function.
AT+MDNS: Configure the mDNS function.
AT+CWHOSTNAME: Query/Set the host name of an ESP station.
AT+CWCOUNTRY: Query/Set the Wi-Fi Country Code.

For more information, please refer to:

https://docs.espressif.com/projects/esp-at/en/latest/AT_Command_Set/index.html

9.3 Example tutorial

Course purpose

What this experiment wants to achieve is to obtain the light intensity data of the current environment through the photoresistor on the right side of the cobit, and then upload the data to the ThinkSpeak website on the Internet through the ESP-01, and we can view the data anytime and anywhere through the ThinkSpeak website .

Tool

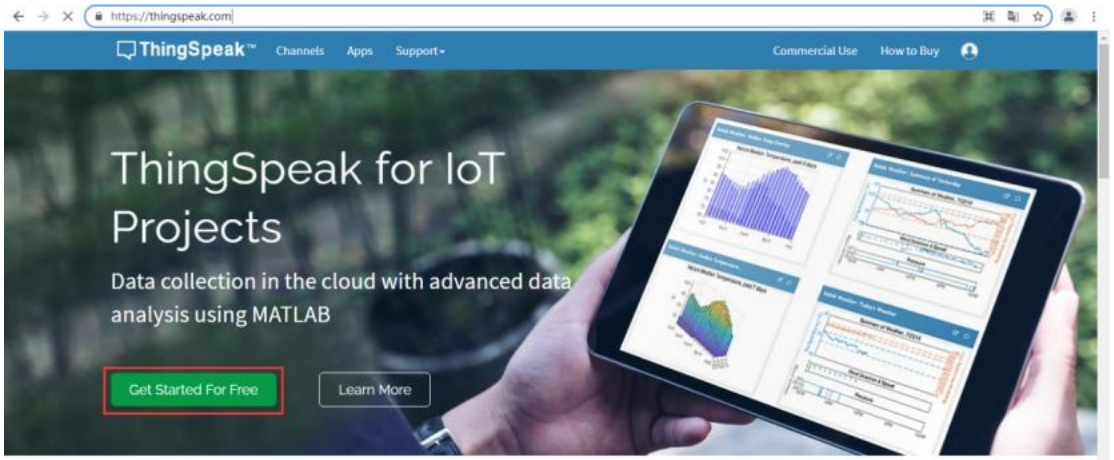
Wi-Fi network and computer with Internet access.

ThinkSpeak

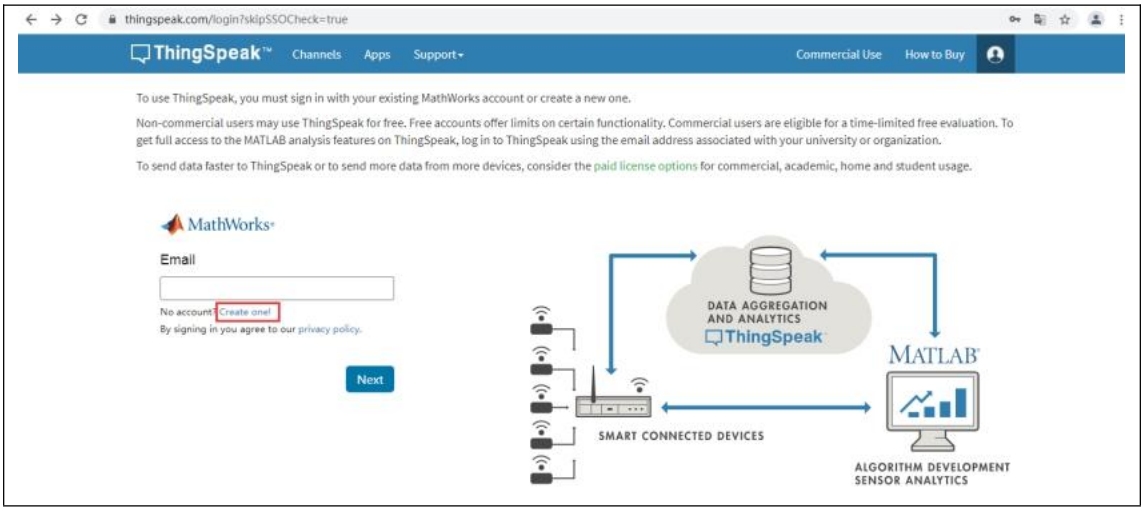
ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize and analyze live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by your devices to ThingSpeak. With the ability to execute MATLAB® code in ThingSpeak you can perform online analysis and processing of the data as it comes in. ThingSpeak is often used for prototyping and proof of concept IoT systems that require analytics.

Step1： register ThinkSpeak account

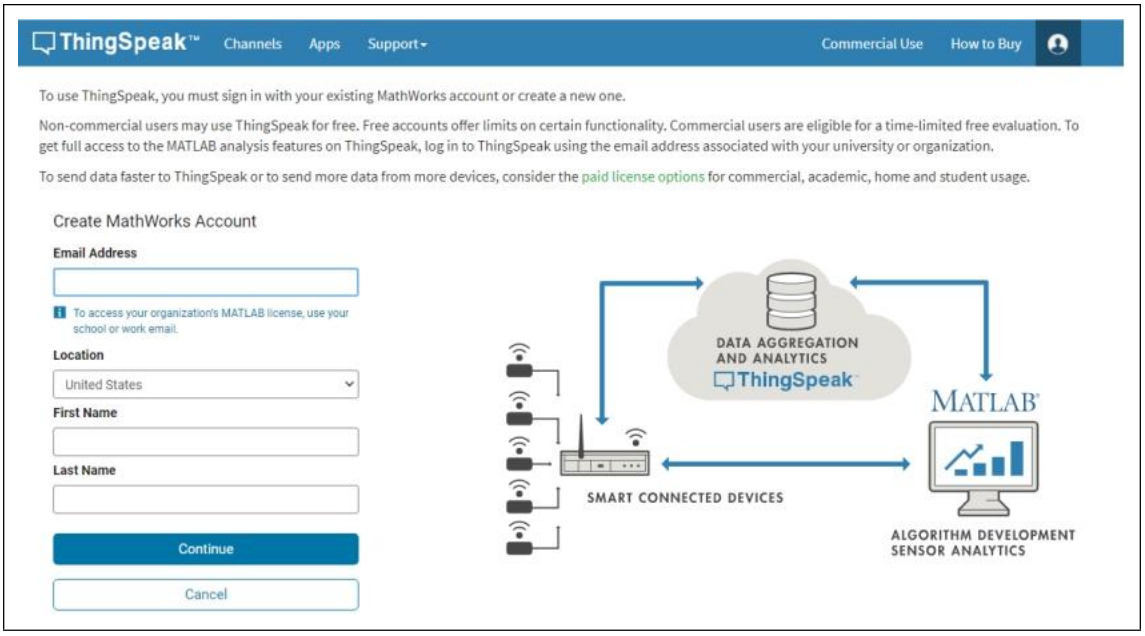
Visit https://thingspeak.com/, and then click “Get Started for Free”：

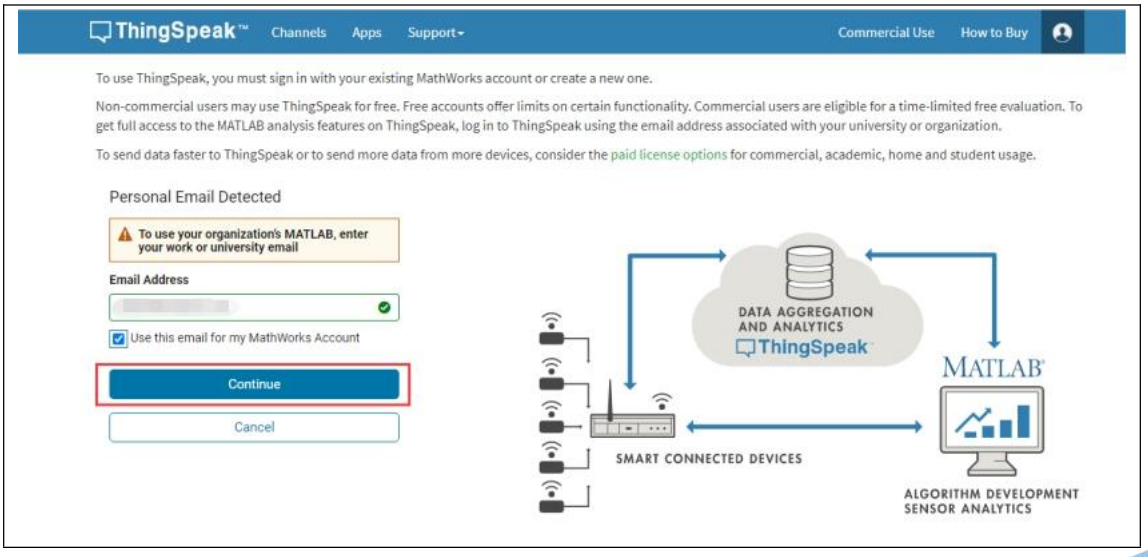


Click “Create one!” create a new account.

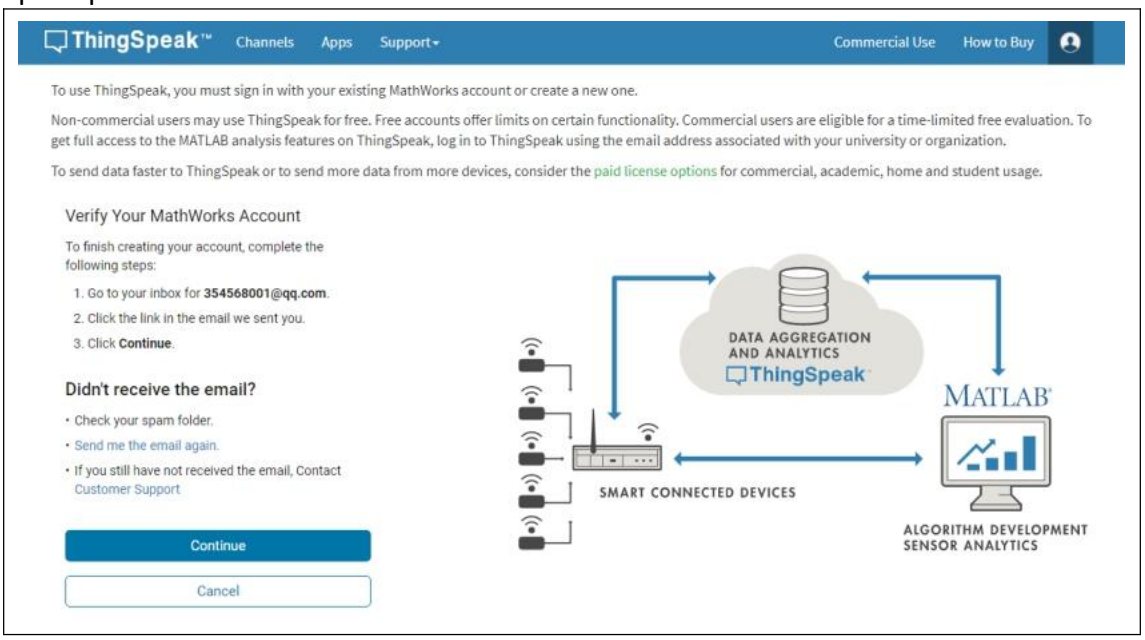


Fill in the relevant information as required, and then click "Continue",



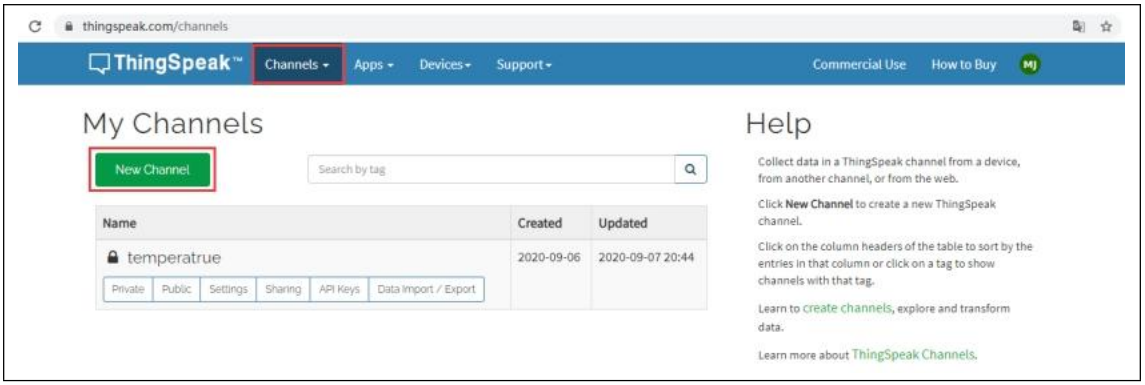


After that, you will receive a confirmation email from ThingSpeak. Just follow the prompts on the interface:

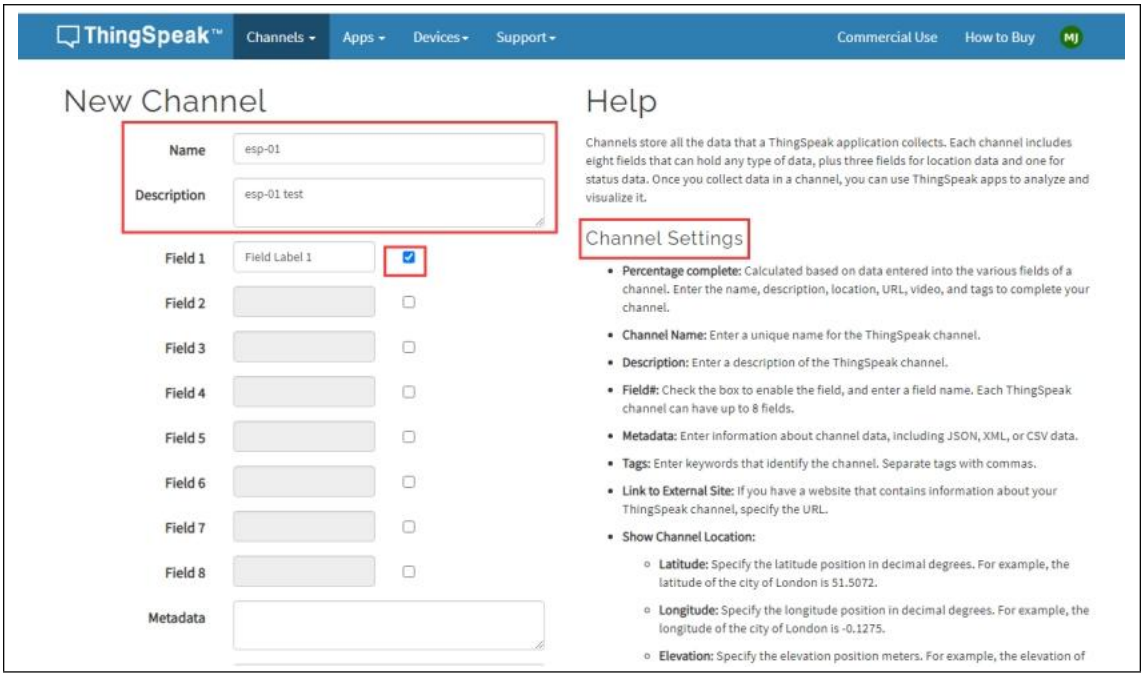


Step 2: Create a ThingSpeak channel

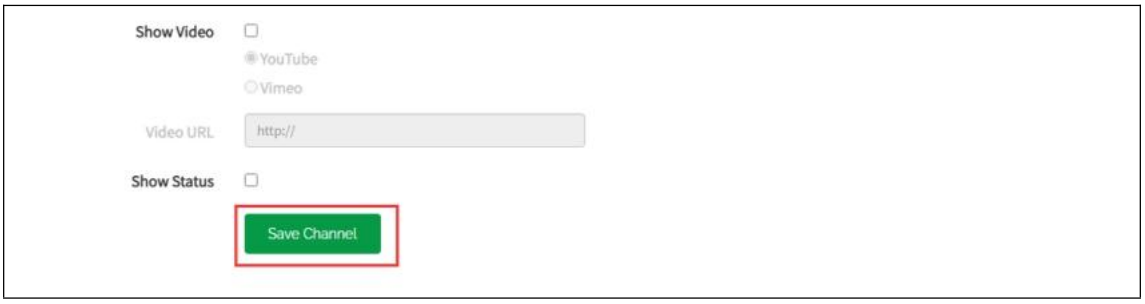
After the account is successfully created, click "channels" --> "new channel" to create a new channel:



Fill in the name and description of the channel, because we only use "field1", so here only check "field1", more setting instructions can also refer to the "Channel Settings" on the right side of the page:

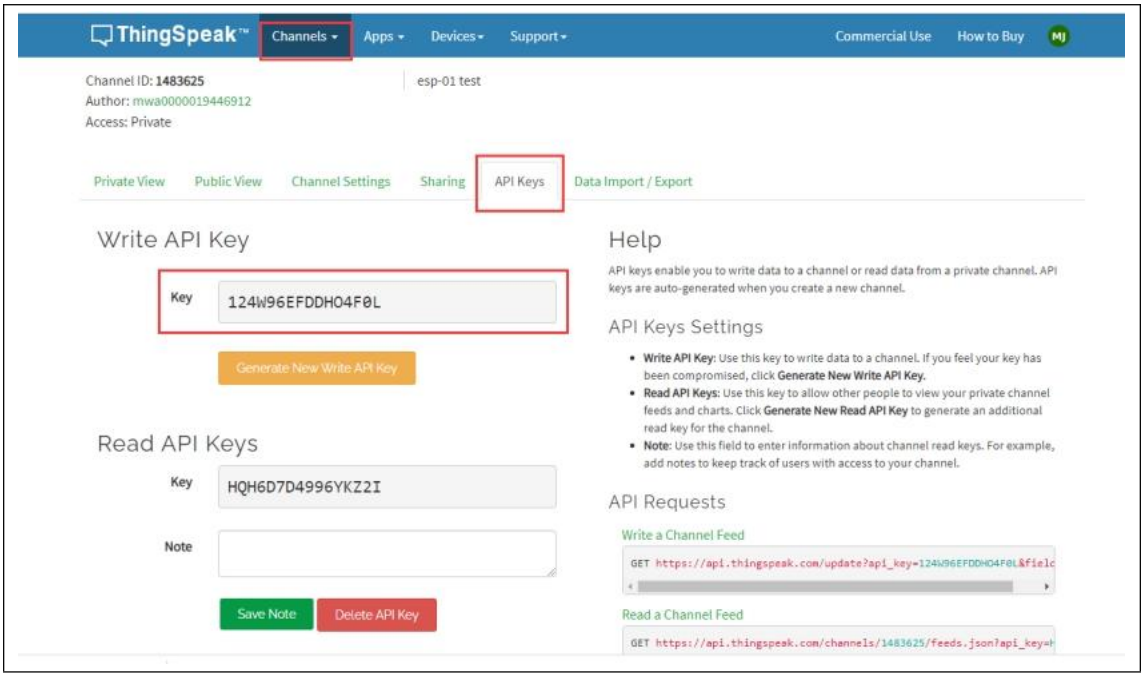


Then drag it to the bottom of the page and click "Save Channel" to complete the creation of the channel:



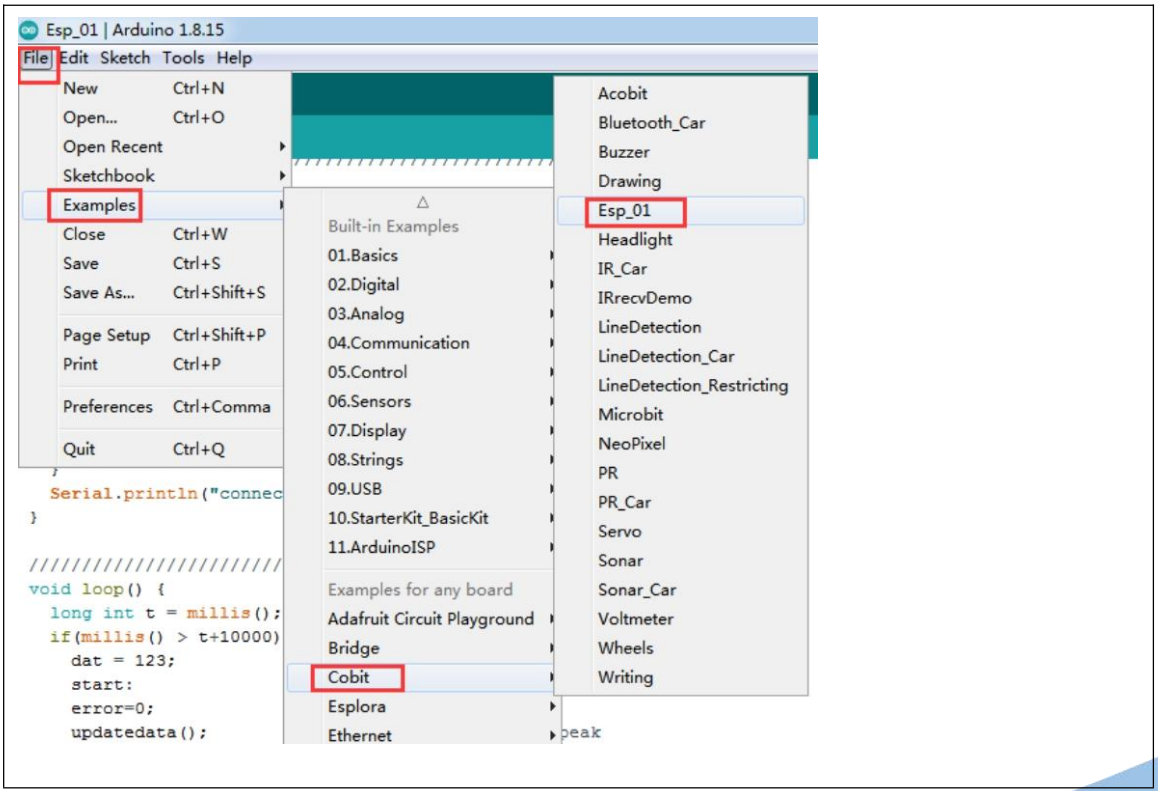
Step 3: Obtain API Keys of ThingSpeak channel

Immediately after the previous step, it will automatically jump to the channel created in the previous step on the "Channels" page. You can select "My Channels" through the "Channels" drop-down menu to select other channels you created (if you have created multiple channels) , And then select "API Keys", we only use "Write API Key" in the project, we need to copy it and record it, the following code uses it:

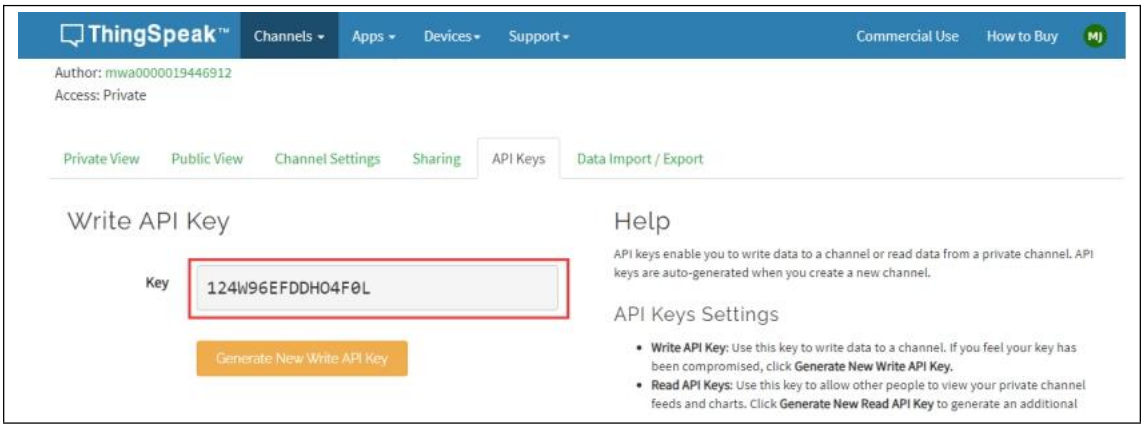


Code

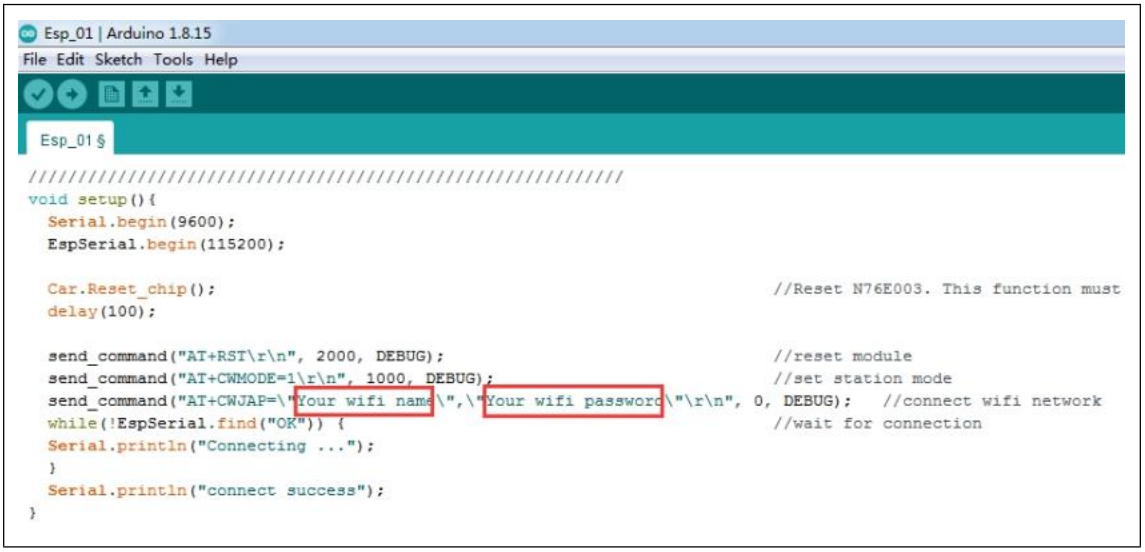
Arduino IDE editing code can easily upload data to ThingSpeak website through ESP-01. We have included the code in Cobit's Arduino library. For the installation method of the library, please refer to the chapter "Cobit uses arduino platform". After ensuring that the Cobit library has been installed in the arduino IDE, then open the arduino IDE and select "File" --> "Examples" --> "Cobit" --> "Esp-01" to open the sample code:



Modify the API Key of the sample code to the AIP Key of the corresponding channel on your account:

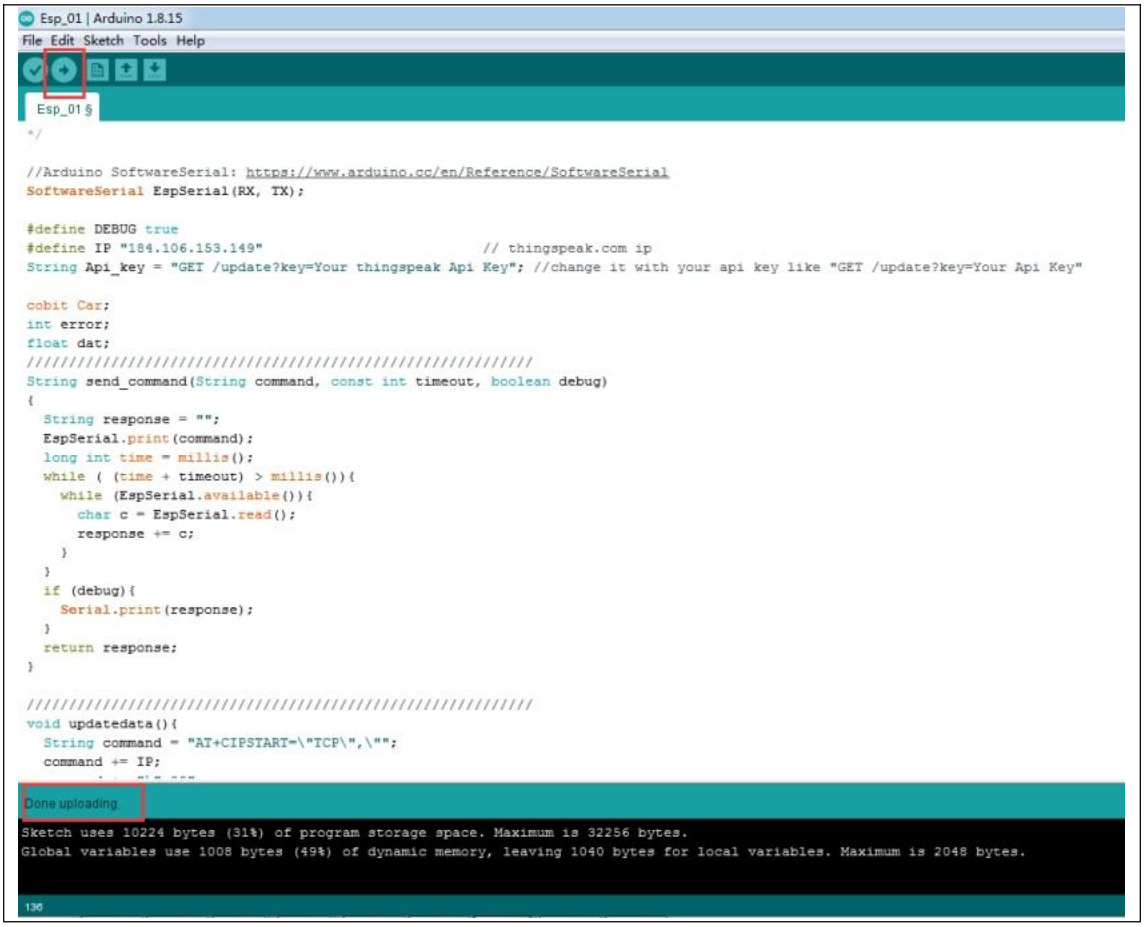


Change the wifi name and wifi password of the sample code to your own wifi name and password:

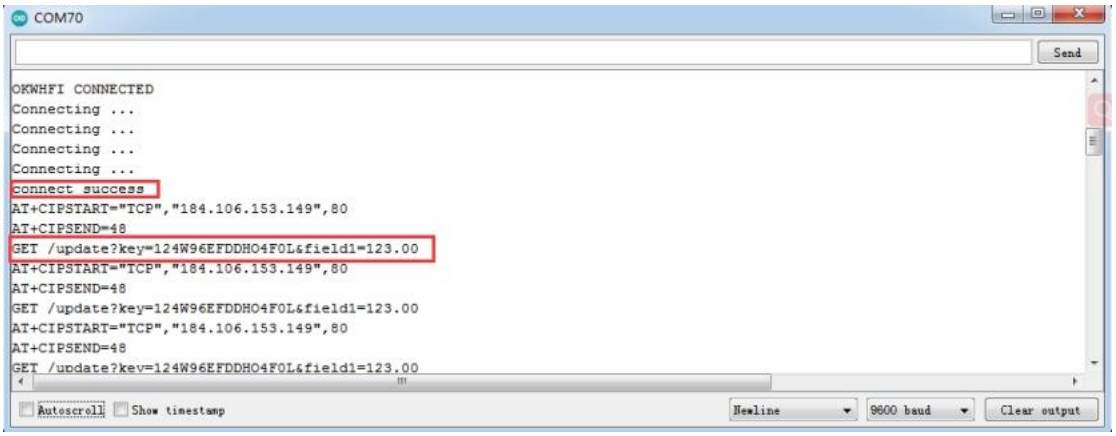


Upload code

After selecting the board type and port number, click the upload menu button (the detailed method of Arduino uploading code can be viewed in the "Cobit Apply with Arduino Platform" chapter):

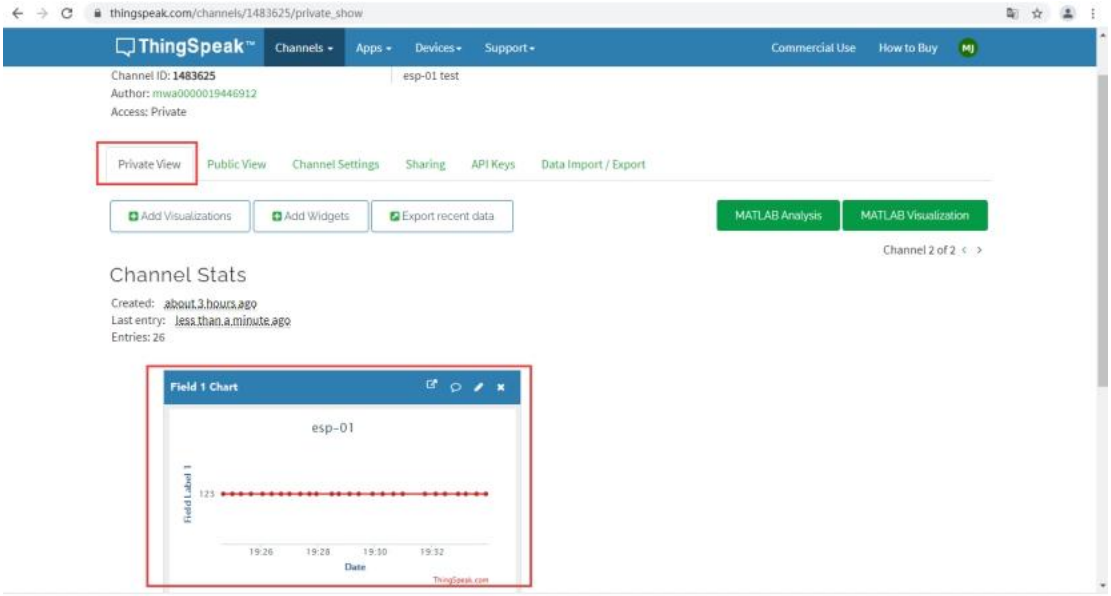


After the code is uploaded successfully, click the arduino IDE menu  , Open the serial monitor, if you successfully connect to the wifi serial port, it will print "connect success". The picture below shows the successful upload of the photoresistor data "123.00" to the ThingSpeak website.



Tip: ESP-01 must be plugged into the corresponding port of the Cobit!

Open the "Pricate View" menu of the corresponding channel to view the uploaded data:



End!

10. Trouble shooting

10.1 Arduino

Could not upload code?

☛Do you use a USB cable with data communication function?

☛If the USB data performance well?

☛Have you selected the correct board type and port?

Can't find port no?

☛Is the CP2102 driver installed?

10.2 Micro:bit

Could not upload code?

☛ At present, Makecode only supports the compilation of HEX code less than or equal to 7KB. When it is greater than 7KB, a compilation error will be reported.

☛Do you use a USB data cable with data communication function?

☛If the USB data performance well?

Drive letter displayed incorrectly?

☛Drive letter display: MAINTENANCE (Normal is MICROBIT)

This is due to accidentally pressing the micro:bit reset button and then powering on the micro:bit, which makes the micro:bit enter the firmware update mode. Re-update the firmware to return to normal, you can refer to the following link operation:
https://microbit.org/get-started/user-guide/firmware/

Not Work?

☛Please check if the micro:bit motherboard is plugged in the wrong way?

10.3 ESP-01

Not Work?

☛ The Cobit code is based on the AT firmware of ESP-01. If the firmware is re-recorded, it will not work. You need to re-burn the factory AT firmware.

☛Is the ESP-01 module with non-AT firmware used?

10.4 Other failures

☛Please check if the assembly is correct?

☛Please check whether the battery power is sufficient?

☛Please check whether the battery used meets the specifications?

☛ When the battery voltage is too low, it will cause walking, writing disorder or malfunction.

11. Develop and join us

We are committed to open source hardware maker education, and sincerely share our products with everyone, and hope to be supported by everyone. We hope that everyone can communicate with us more to find out the shortcomings of our products. We have open sourced all our products on the github website. We hope you can interact with us and welcome to our team.

 https://github.com/Cokoino

 cokoino@outlook.com

 http://cokoino.com