MotorDriverBoard

Scratch Graphical Programming Tutorial

V.1.0

Revision

Date	Revision No	Description	Author
2018-12-20	**V.1.0**	**Create Document**	**Twisted**

目录

Revision 2
Chapter 1 Learn About Motor Driver Board And Its Programming Environment....5
1.1 Introduction....5
1.2 Overview....5
1.3 MotorDriverBoard Fuction Module List 8
1.4 MotorDriverBoard Graphical programming software introduction 8
1.4.1 Software Installation 8
1.4.2 Introduction of the compilation environment 10
1.5 Label area introduction 11
1.6 MotorDriverBoard Building Block Introduction 13
Chapter 2 First-time programming 16
2.1 Hello word 16
2.1.1Add the MotorDriverBoard library 16
2.1.2 First-time experience of programming 20
2.2 mBlock and MotorDriverBoard Connection steps....21
2.3 Upload a program to the MotorDriverBoard robot 22
Chapter 3 Let robot move 25
3.1 DC Motor....25
3.1.1 DC Motor Principle....25
3.1.2 DC Motor Test 25
3.2 Code Motor 26
3.3 Stepper Motor 26
3.4 Ultrasonic module....28
3.4.1 Ultrasonic principle....28
3.4.2 How to use the ultrasonic module....28
3.4.3 Ultrasonic obstacle avoidance module test 29
3.4.4 Servo working principle....29
3.4.5 Servo Test 30
3.5 RGB light module....31
3.6 Buzzer module 31
3.6.1 Introduces through life....31
3.6.2 Learn about modules and use them....32
Chapter 4 Infrared Remote Control Robot....33

4.1 Infrared remote control principle 33
4.2 How to use the remote control 33
4.3 Infrared remote control test 33
Chapter 5 PS2 Module 34
5.1 PS2 Remote control handle principle 34
5.2 PS2 Remote control test 36
Chapter 6 Nrf24L01 Module 37
6.1 Nrf24L01 Introduction 37
6.2 Drive NRF24L01+ module 37
6.3 NRF24L01+test demo 38

Chapter 1 Learn About Motor Driver Board And Its Programming Environment

1.1 Introduction

Hello everyone, welcome to the world of Arduino robots!In the arduino world, we often need to make the robot move. So we need a special motor to drive. We have integrated many motor scenes, sensor connections, wireless control, etc., and finally designed the Moto rDriverBoard V3.0 that can drive multiple motors and connect multiple sensors, support PS2X port and NRF24L01+ module.

1.2 Overview

The PS2X&Motor Driver Board driver can drive 4 DC motors, 2 encoder motors, 2 stepper motors, 6 servos (two external power supplies), and drive current up to 2A. The driver board is specially designed for the Arduino uno R3 mainboard. It can be directly plugged into the Arduino Uno. The mainboard integrates a passive buzzer, 2 RGB LED lights, and 1 infrared receiver. Besides it also reserved for PS2 socket, Uart interface, I2C interface, ultrasonic obstacle avoidance module socket and other sensor interfaces, it is very convenient to externally connect various sensor modules. 1.2.1 Motor Driver Board is described as below.

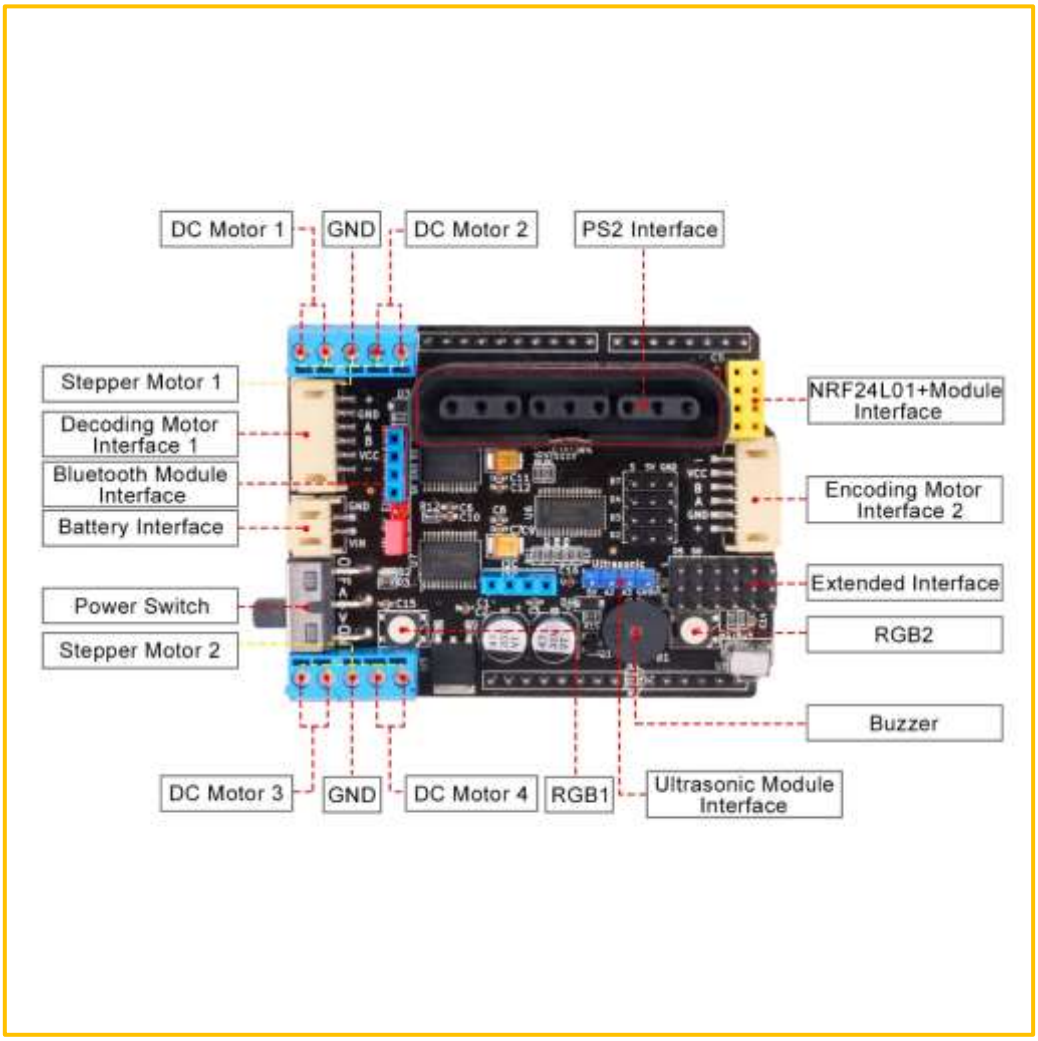


Figure1.2.1 Motor Driver Board is described

1.2.2 Common Questions

Q：How to power Emakfun Motor Shield?

A：The Emakfun Motor Shield is powered by a lithium battery and has a voltage range of 6 to 12V.

Q：How many motors does Emakfun Motor Shield can drive?

A：The Emakfun Motor Shield can drive 2 DC motors or 2 code motors.

Q：Can Emakfun Motor Shield drive other servo equipments?

A：The Emakfun Motor Shield can also drive the servo and drive directly by connecting the servo to the servo interface using the DuPont line.

Q：How does Arduino UNO board conncet with Emakfun Motor Shield?

A：Emakfun Motor Shield stacked design can directly insert into Arduino and use without additional wiring.

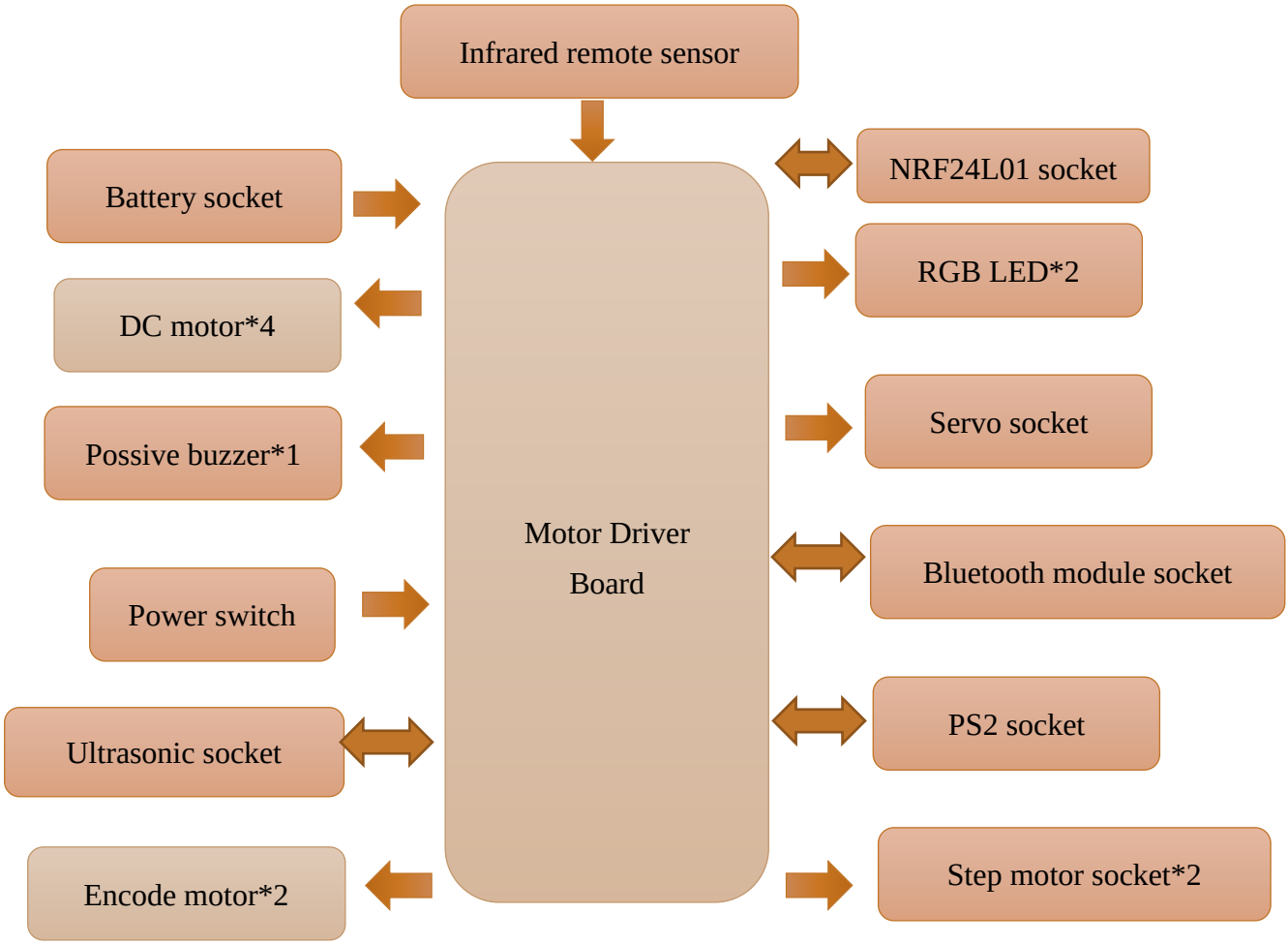
Q：I want to upload the sample program to the Arduino board, suggesting that the upload failed. Why?

A：Before uploading the sample program to the Arduino board, you need to check if the board and the computer are properly connected, then install the driver and try again.

Q：I want to upload the program to the Arduino board, and then the motor does not work after turning on the power. Why?

A：Firstly, check if the green indicator on the ArduinoUNO board is on. If it is not on, it means the power supply is not normal. Check whether the voltage of the battery is above 6v. Then check whether the wiring port of the motor is the same as the port set in the program. Turn it on again.

1.3 MotorDriverBoard Fuction Module List



1.4 MotorDriverBoard Graphical programming software introduction

1.4.1 Software Installation

1) According to your computer system model, choose to download the corresponding software version, software software download address: mBlock.cc/download
2) Double-click the installation package, as shown in Figure 1-5, then the language 'Select Chinese (Simplified)', click OK, as shown in Figure 1-6，

Figure1-5

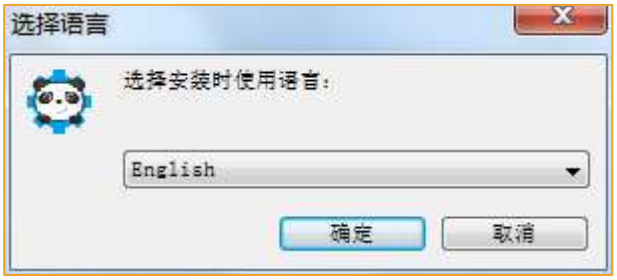


Figure1-6

3) Click “I accept the agreement (A)” as shown in Figure 1-7, and then click “Next” to complete the installation, as shown in Figure 1-8.

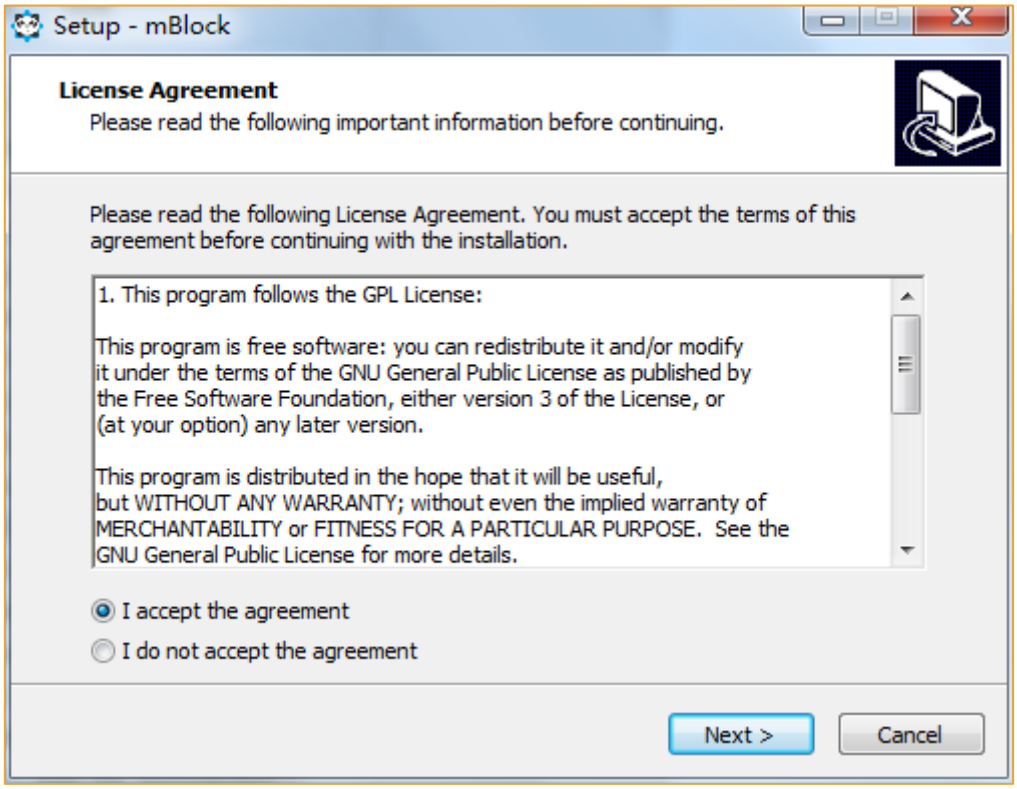


Figure 1-7

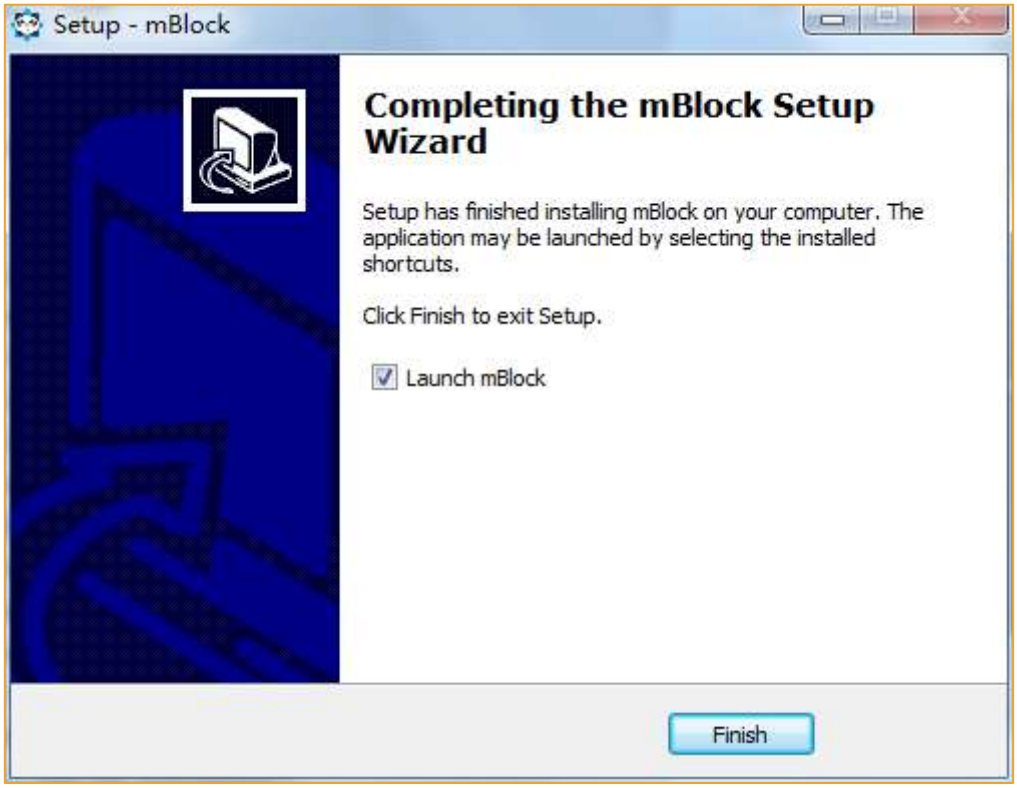


Figure 1-8

1.4.2 Introduction of the compilation environment

We know that in order to move the robot, in addition to the hardware cooperation, it is more important to write a program for it. We have already understood the hardware part of the robot. We will then understand the software part of it. The software part of MotorDriverBoard is programmed on the graphical programming software mBlock. With this software, we can control the robot by writing various commands we want it to execute! The software interface of mblock is shown in Figure 1-9.

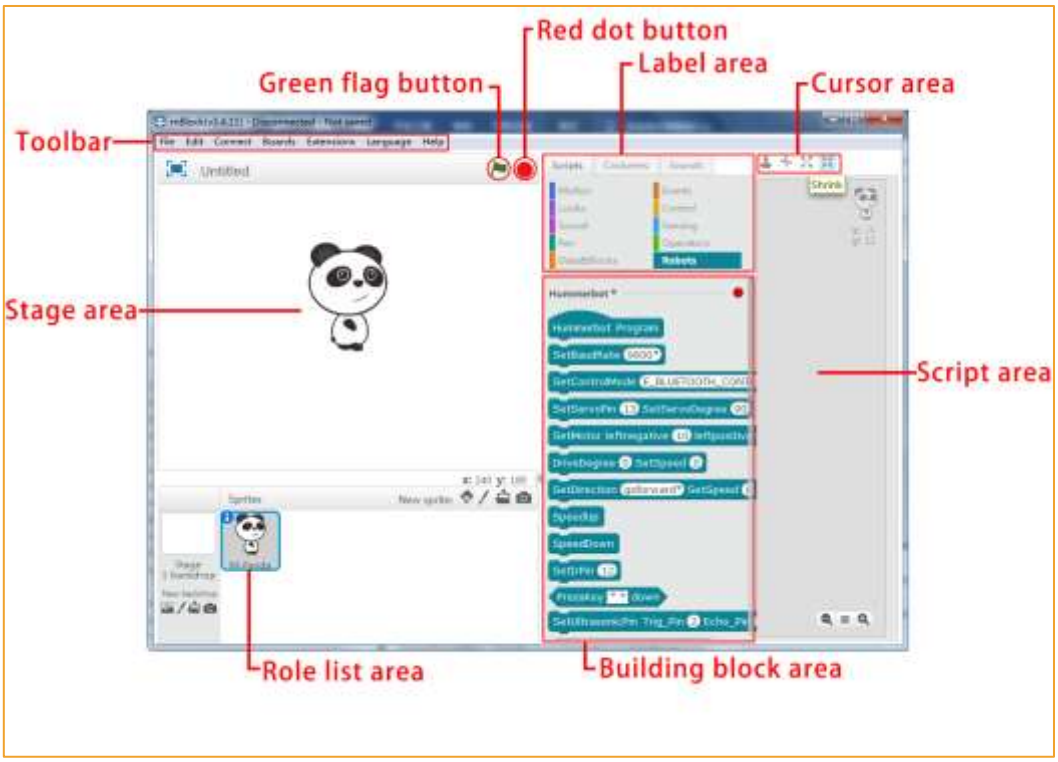


Figure 1-9

- **Toolbar：** The area where the project file, software interface mode, serial port connection and upload program, control panel selection, software language, and software update are operated.
- **Stage area：** The area where the roles and roles in the work, the interaction between the character and the user and the program display the running effect.
- **Green flag button：** Click the green flag button to start the program.
- **Red button：** Click the red button to stop the program.
- **Role list area：** The area where all the character prototypes are displayed, where you can see the name of the character, the direction of rotation, the position, and so on.
- **Label area:** The area where contains script tabs, styling tabs, and sound tabs can operate the script, styling, and sound.
- **Block area：** The same type of building blocks are grouped in the same module and given the same color, and each building block represents a control command.
- **Cursor area:** The area where contains copy, delete, reduce, zoom in buttons, which are used to operate the characters on the Stage.
- **Script area:** The area where the program is written, the blocks can be programmed by stacking them in the script area.

1.5 Label area introduction

The label area of Mblock is mainly divided into script (as shown in figure 1-10), modeling (as shown in figure 1-11) and sound (as shown in figure 1-12). The script is mainly some control blocks to perform the action; Modeling label is able to draw some of their own graphics to add to the program; The sound tag can record some sounds by itself or put them into the program. The MotorDriverBoard mainly USES the control blocks in the script tag.

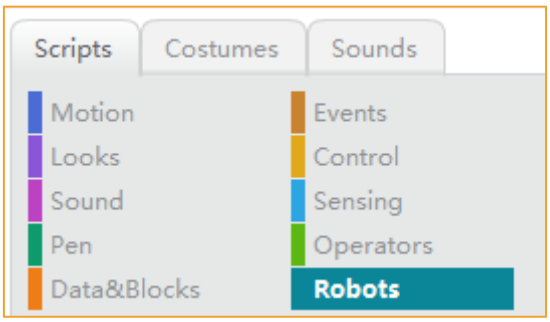


Figure 1-10

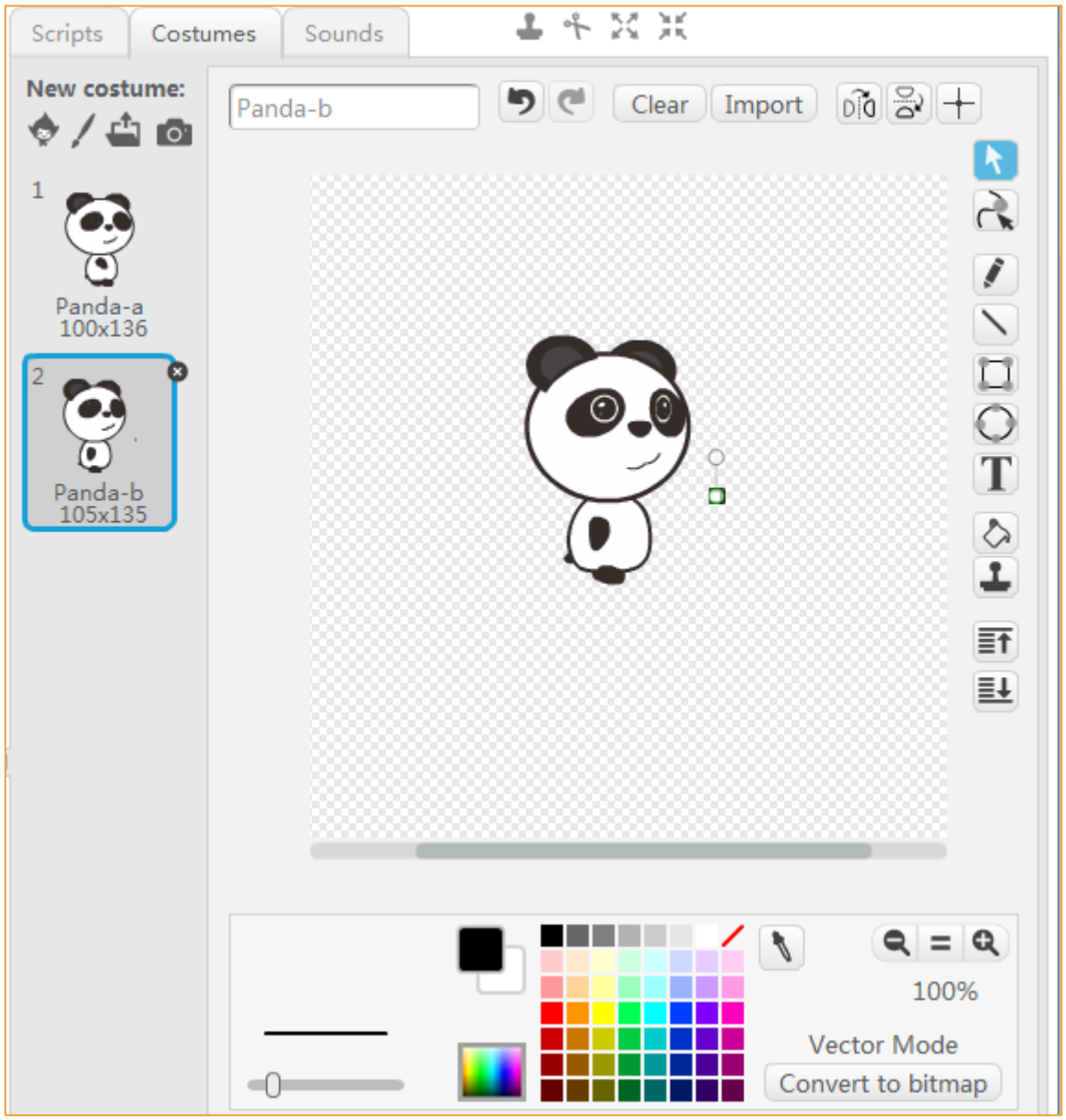


Figure 1-11

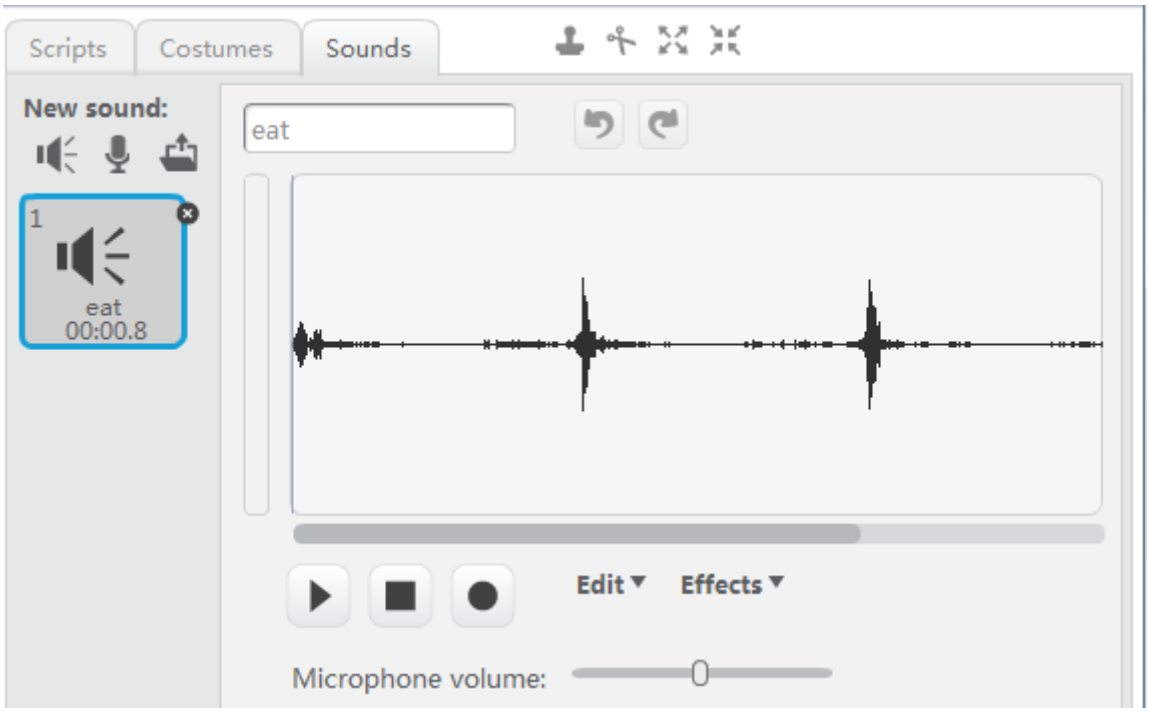


Figure 1-12

1.6 MotorDriverBoard Building Block Introduction

There are 10 types of building blocks under Mblock's script tag. The 7 building block types of action, appearance, sound, brush, data and command, event, and detection are some effects and data. It is not used in MotorDriverBoard. If you are interested You can try it yourself. We mainly know about the three building block types of control, digital and logical operations, and robot modules.

1) The control building blocks are all the blocks that control the execution flow of the program (Figure 1-13), leading the program.

Figure 1-13

For example,  this block means to Repeat the program inside the building block all the time,

for exmaple  this block means to make the robot print the number 0 on the serial port all the time；

he main function of the building blocks of numbers and logic types is to do mathematical operations, as a condition for judgment, comparing size and logical judgments with, is or not, as shown in Figure 1-14.

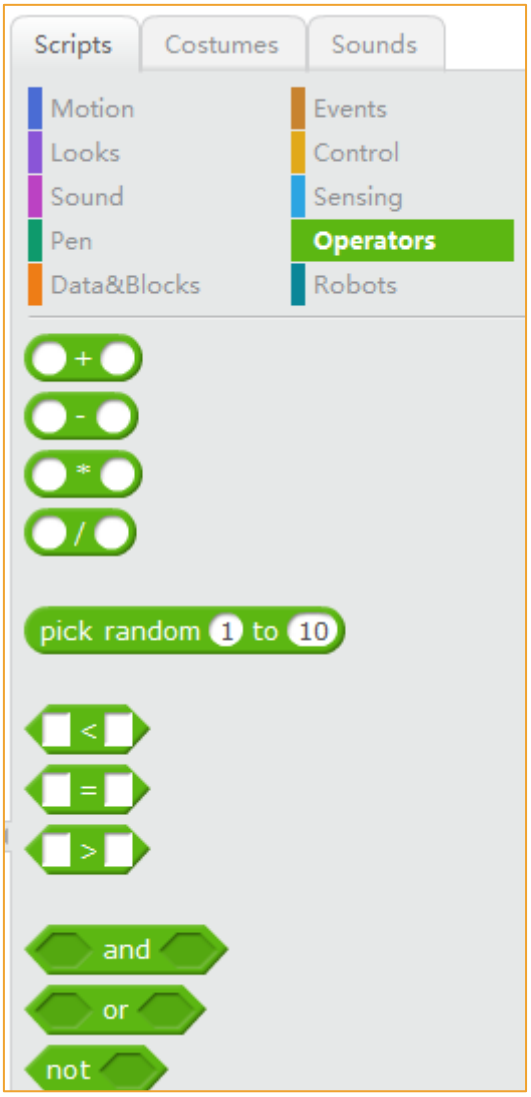
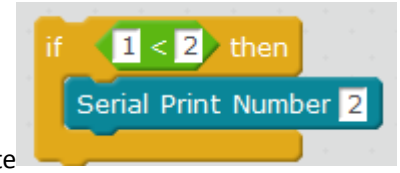
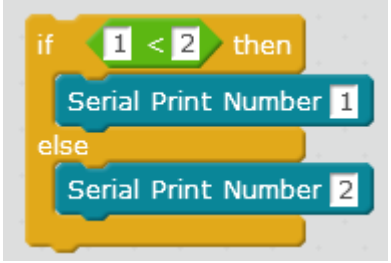


Figure 1-14

1) For example  building blocks are operations that judge size, when using control blocks, numbers, and logic operations with robotic building blocks,then can wirte  such a program, which means to determine whether this condition  is true, If the condition is met, then execute the serial port to print the number 2; Expand it,  Such a program means to

judge  whether this condition is ture, If the condition is met, then the serial port print number 1 is executed, and if the condition is not established, the serial port print number 2 is executed.

3）The main function of the robot module is to control the robot to perform corresponding actions, such as forward, backward, steering steering, etc., as shown in Figure 1-15 and Figure 1-16.

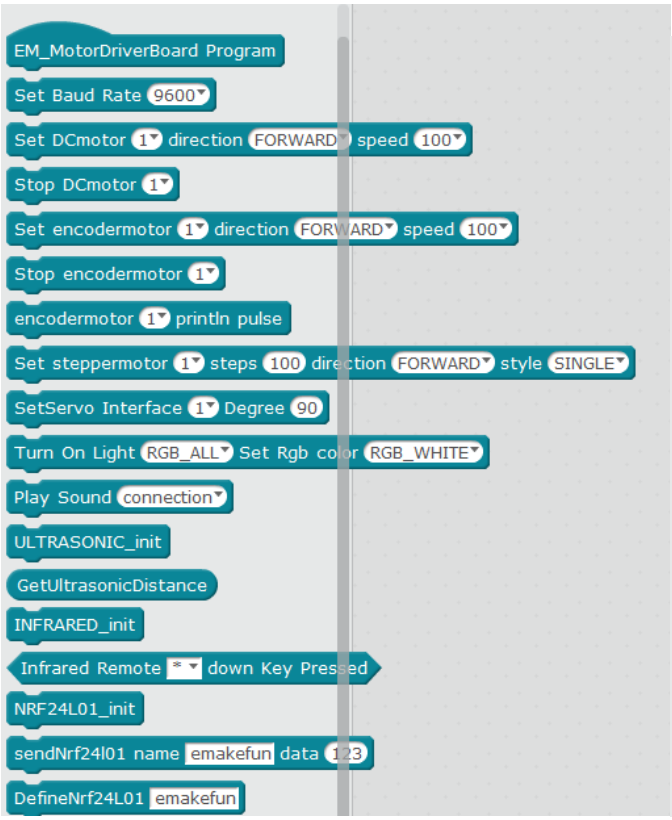


Figure 1-15

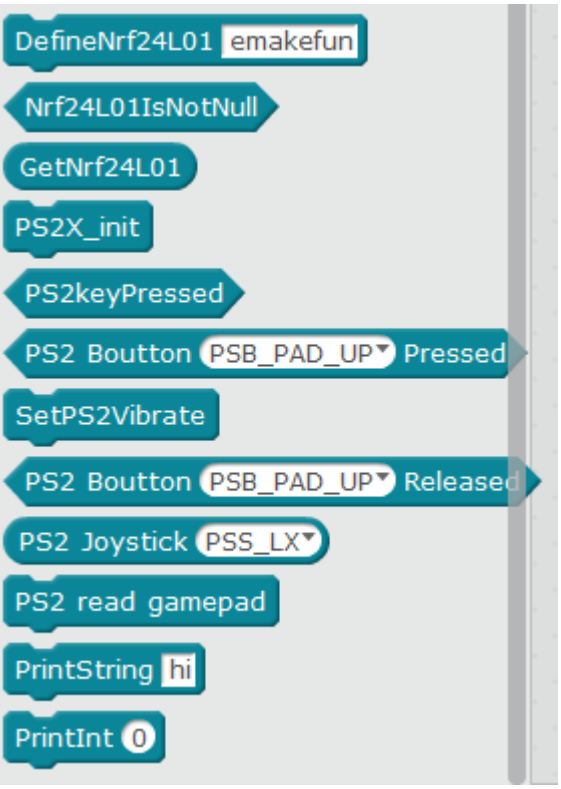


Figure 1-16

Chapter 2 First-time programming

2.1 Hello word

If we want the MotorDriverBoard to move, we need to store the MotorDriverBoard in advance in its brain (control board). How do we compile instructions for the MotorDriverBoard? Let's take a look at the process of writing a Hummer-bot robot to print Hello word.

2.1.1Add the MotorDriverBoard library

Before programming, we need to add the MotorDriverBoard library firstly, as described below:

1) Download the Hummer-bot library and save it on your computer. File name: MotorDriverBoard.zip
2) Open the mBlock software and click “Extension →Extension Manager”, as shown in Figure 2-1;

3) In the Extension Manager, click "Add Extension" in the lower right corner, as shown in Figure 2-2.

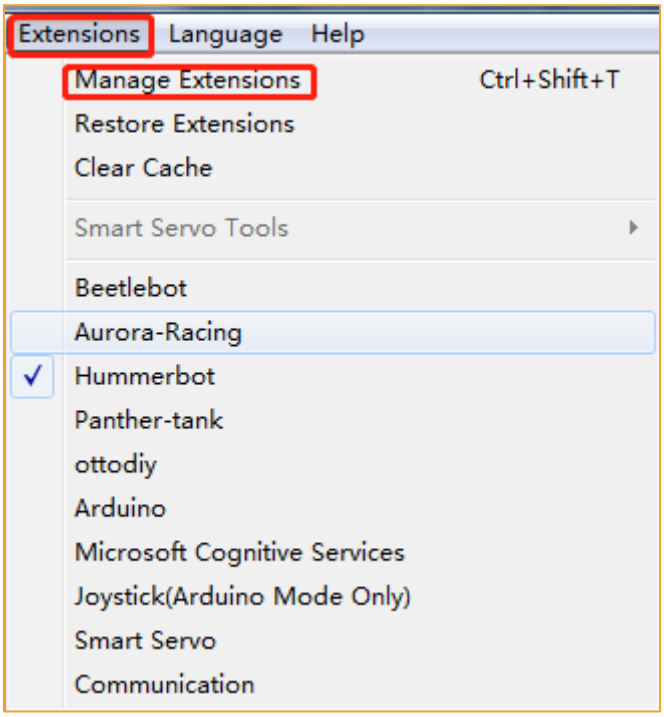


Figure 2-1

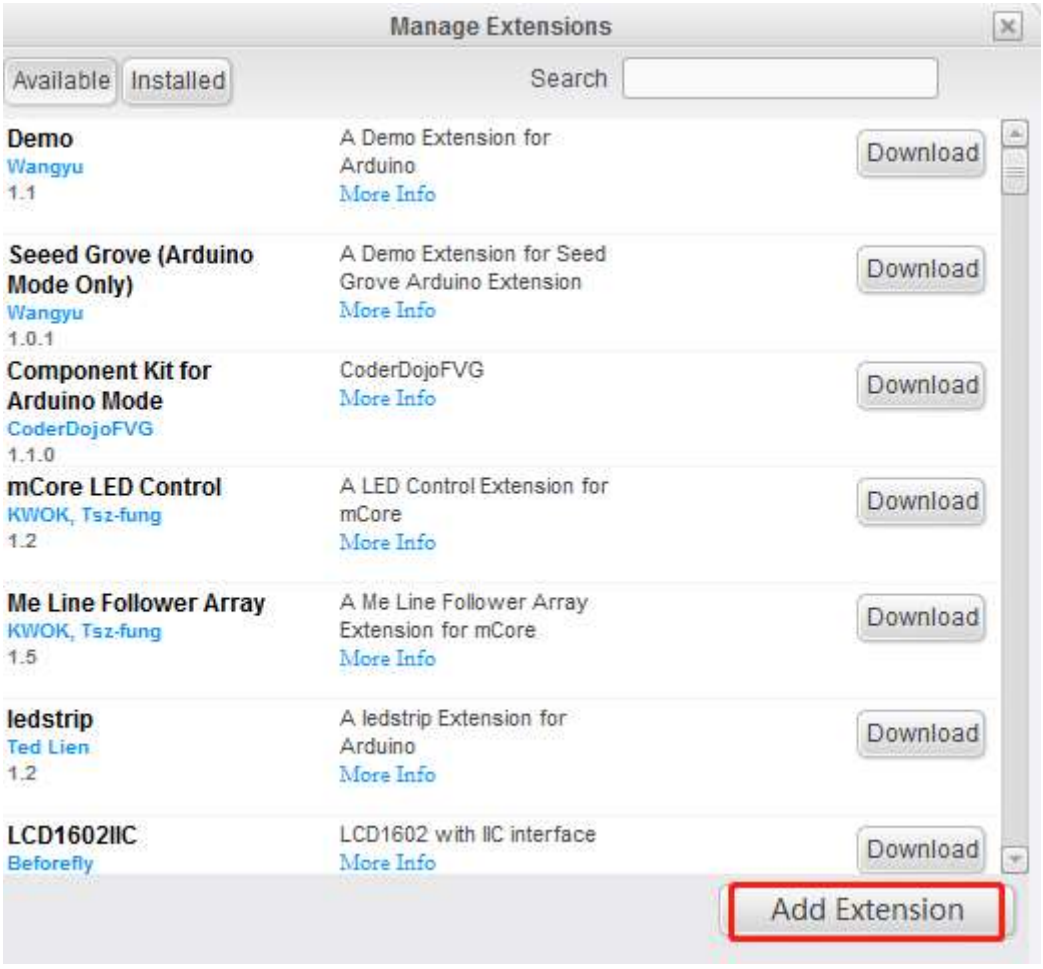


Figure 2-2

4) Select “Zip.file” for the file type, then select “MotorDriverBoard” and click “Open”, as shown in Figure 2-3.

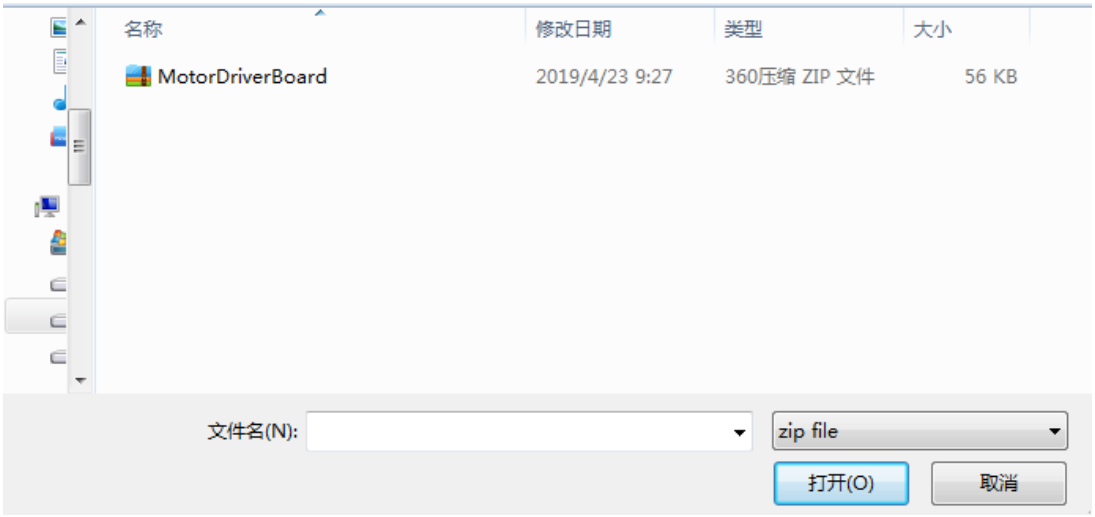


Figure 2-3

5) In the "Extension Manager", click "Installed", you will see that the MotorDriverBoard library has been successfully added, as shown in Figure 2-4;

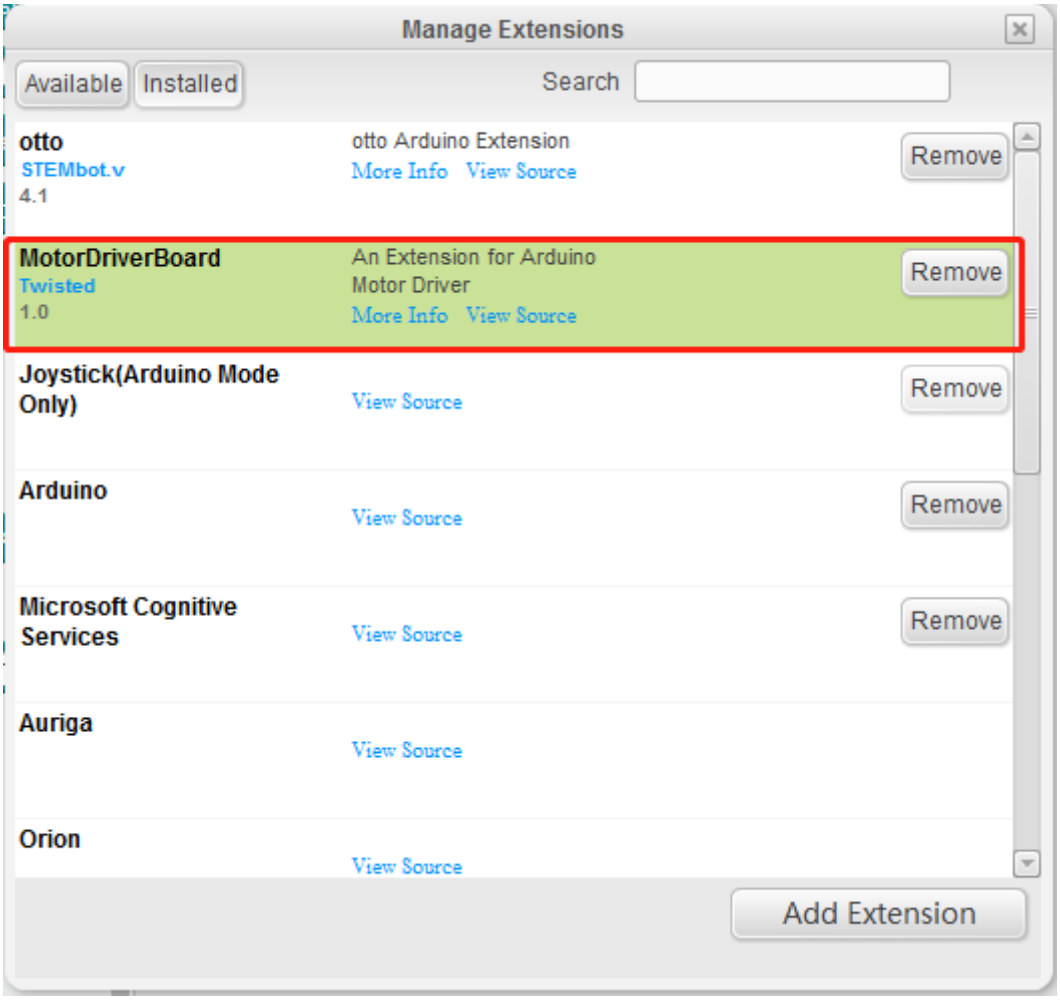


Figure 2-4

6)Click "Extension", select "MotorDriverBoard", and then click "Script —> Robot Module", the MotorDriverBoard block graphic programming block will be displayed in the building block area, as shown in Figure 2-5.

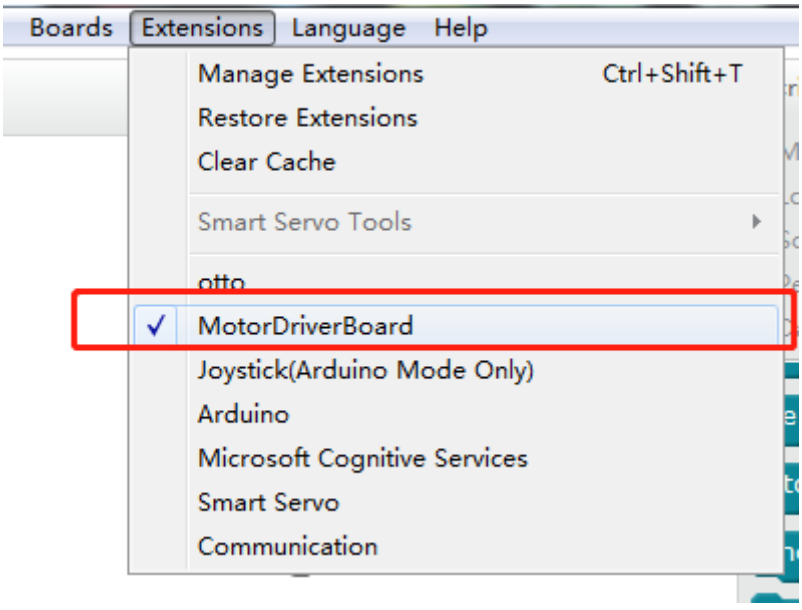


Figure 2-5

2.1.2 First-time experience of programming

After adding the MotorDriverBoard library, let's experience MotorDriverBoard programming! Let's start by writing a program that makes the MotorDriverBoard print the Hello word;

1) Firstly, drag the "EM_MotorDriverBoard main program" of the building block area to the script area with the mouse, and then drag "Set serial port baud rate 9600" to the "EM_MotorDriverBoard main program" building block, as shown in Figure 2-6.

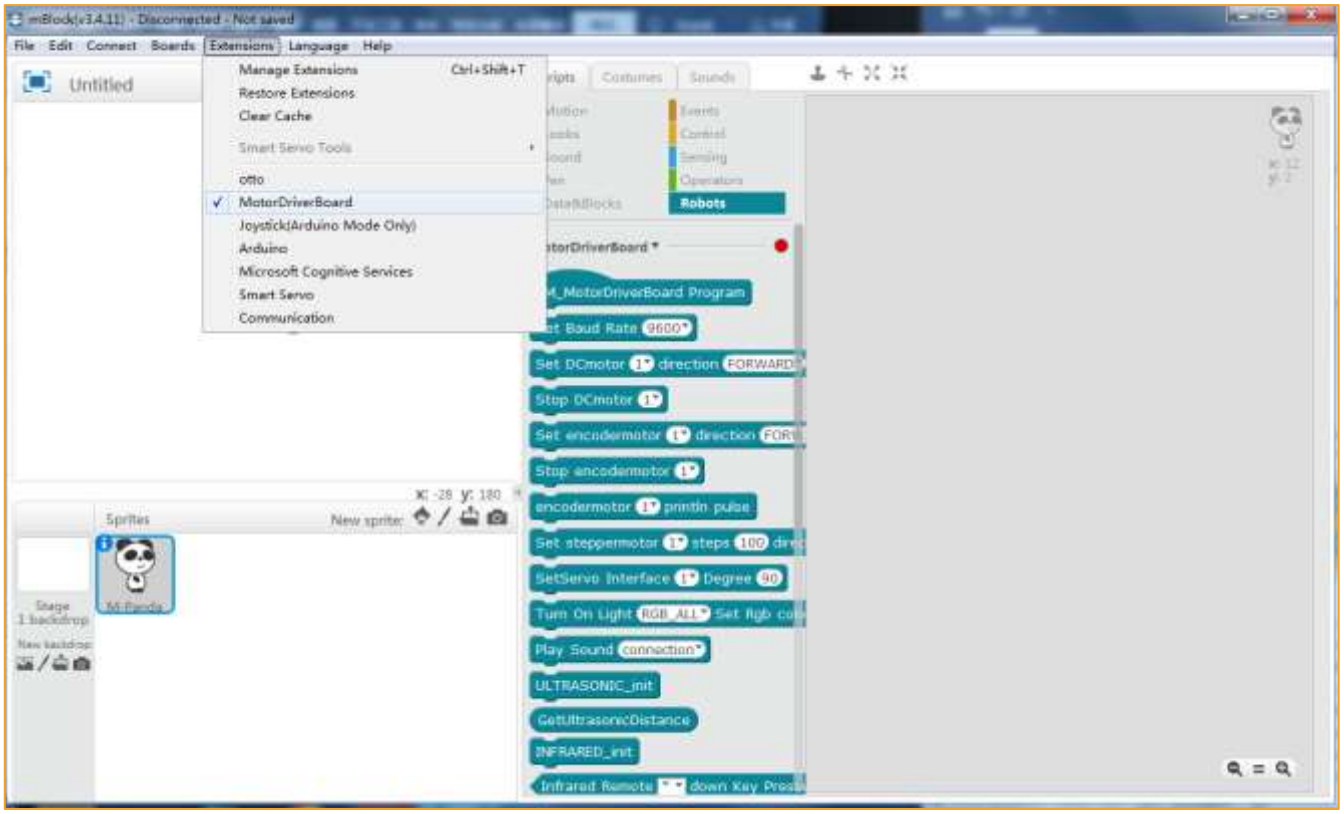


Figure 2-6

2) Click the "Control" tab in the tab area and drag the "Repeat" building block to the bottom of the MotorDriverBoard main program in the script area, as shown in Figure 2-7.

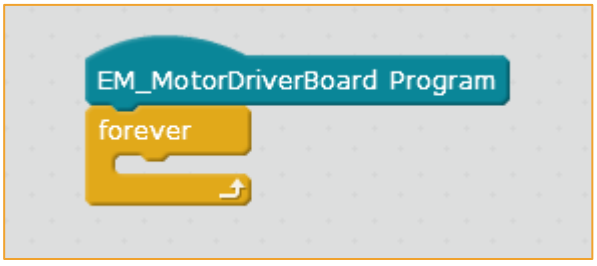


Figure 2-7

3) Drag the "Serial Print String" block into the repeating building block in MotorDriverBoard building block and enter "Hello word" as shown in Figure 2-8.

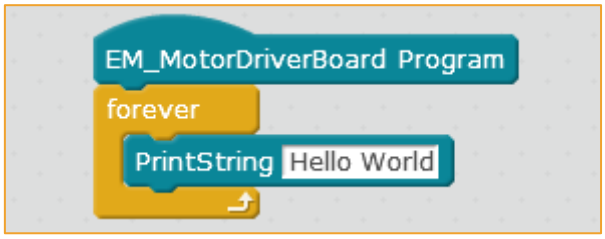


Figure 2-8

The above is the program that MotorDriverBoard prints Hello Word. After the program is written, we need to transfer the program to the brain of the MotorDriverBoard (control board) to let the robot do the action we want according to the program we wrote. How to transfer the program into the brain of the MotorDriverBoard (control board)? Only when the mBlock and the robot main control board are connected together, we can transfer the program written on the computer to the brain of the MotorDriverBoard (control board). The following describes the connection method of the mBlock and the robot main control board.

2.2 mBlock and MotorDriverBoard Connection steps

Use a USB data cable, one end is plugged into the computer, the other end is inserted into the robot main control board, and the robot motherboard is connected to the computer;

1) Install the MotorDriverBoard main control board driver, click "Connect→ Install MotorDriverBoard Driver" as shown in Figure 2-9; click "Install", as shown in Figure 2-10, drive installation;

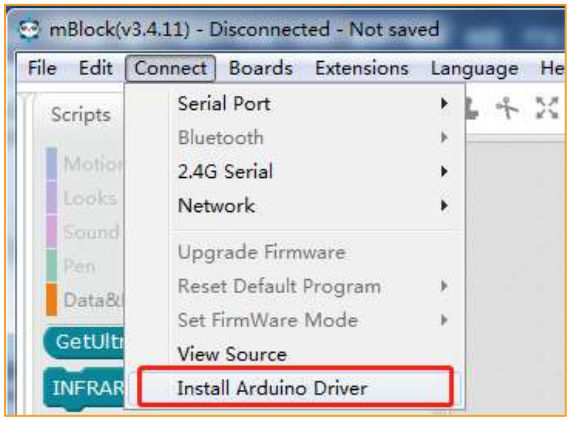


Figure 2-9

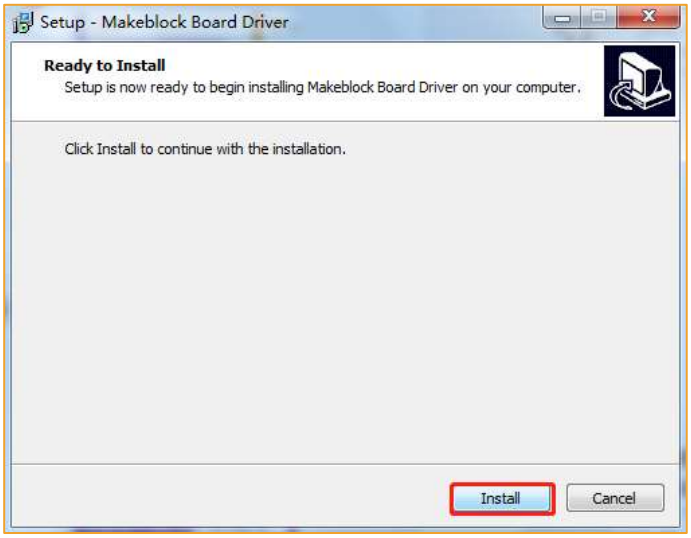


Figure 2-10

1) Click "Connect→ Serial port→COM4 (the number of COM port is different for different computers)", as shown in Figure 2-11. After the correct connection, there will be a "Serial port connected" prompt at the top of the software. The mblock and

MotorDriverBoard are successfully connected, as shown in Figure 2-12.

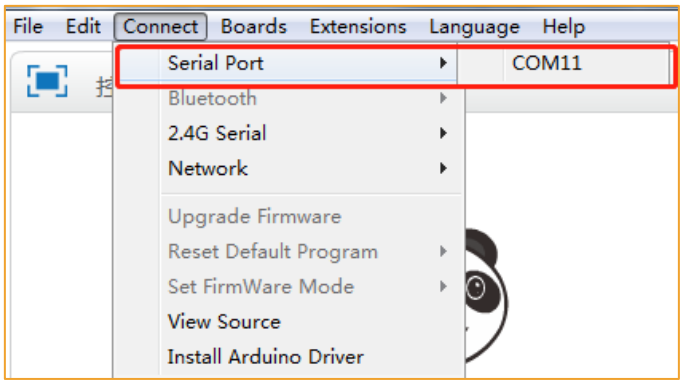


Figure 2-11

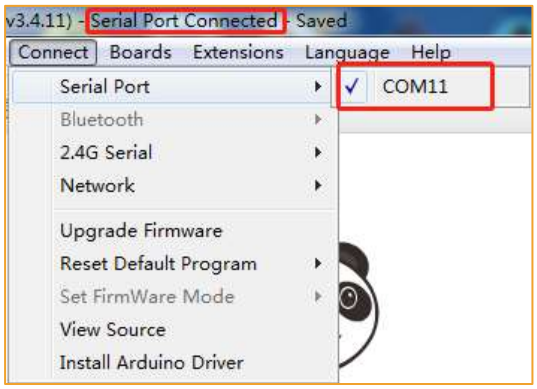


Figure 2-12

2.3 Upload a program to the MotorDriverBoard robot

When the program is written and the mBlock and MotorDriverBoard robots are properly connected, we can transfer the program we wrote to the brain (main control board) of the MotorDriverBoard robot. The specific steps are as follows:

1) Select the type of control board to transfer the program, select "Control board→Arduino Uno", as shown in Figure 2-13; select "Edit→Arduino Mode", as shown in Figure 2-15.

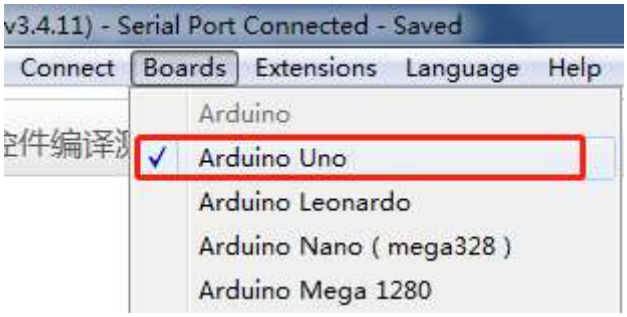


Figure 2-13

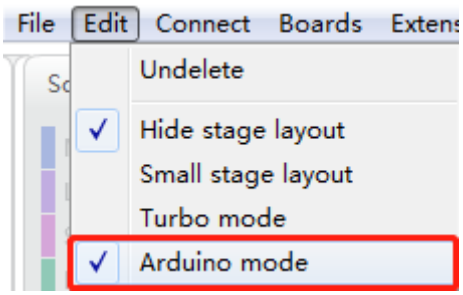


Figure 2-14

2) Click on the mouse block in the script area. The program code of the corresponding building block will appear on the right side of the building block. Click "Upload to MotorDriverBoard" to start generating the offline code of the building block and upload it to the MotorDriverBoard robot. The prompt window "Uploading" will appear on the screen, as shown in Figure 2-16; After the upload is complete, prompting "Upload Complete", as shown in Figure 2-17;

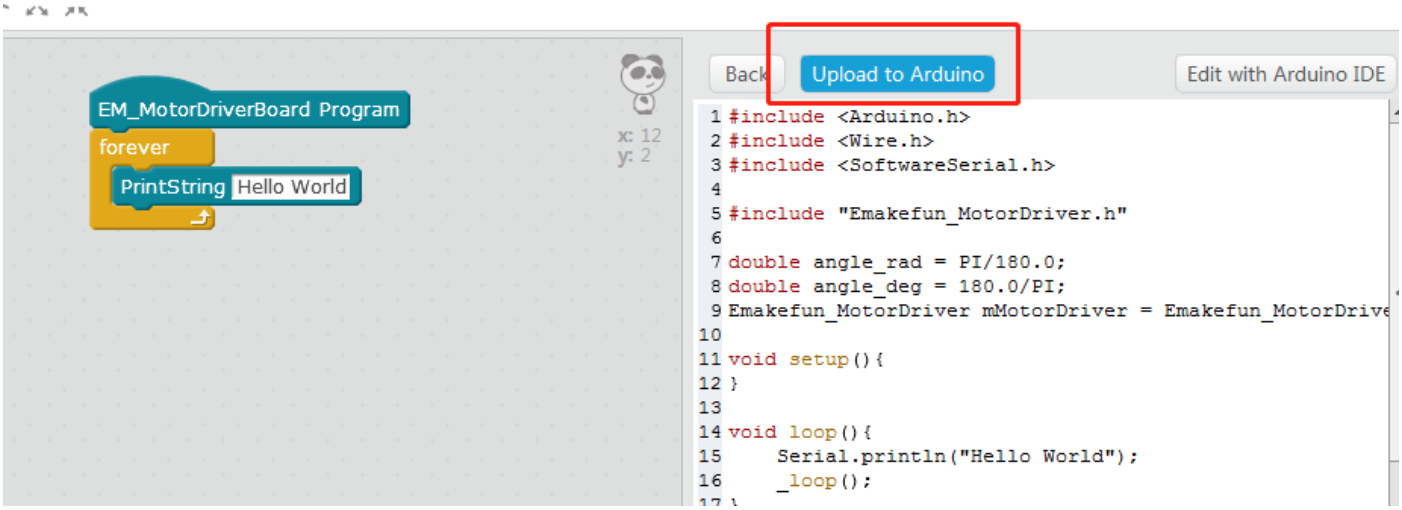


Figure 2-15

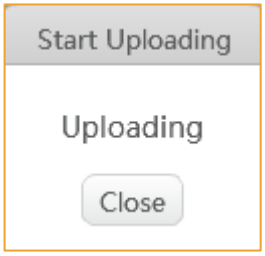


Figure 2-16

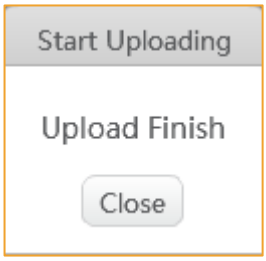


Figure 2-17

After completing the above steps, the MotorDriverBoard robot's brain (main control board) already has the program we wrote, then how do we see the hello word printed by the robot? At this point we will use a serial monitor to see, click on the "Edit with MotorDriverBoard IDE" in the upper right corner, as shown in Figure 2-18.

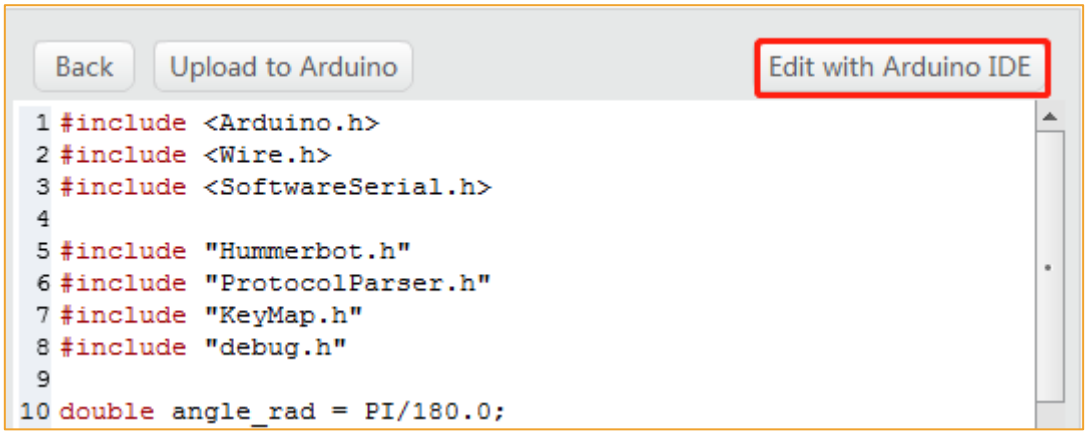
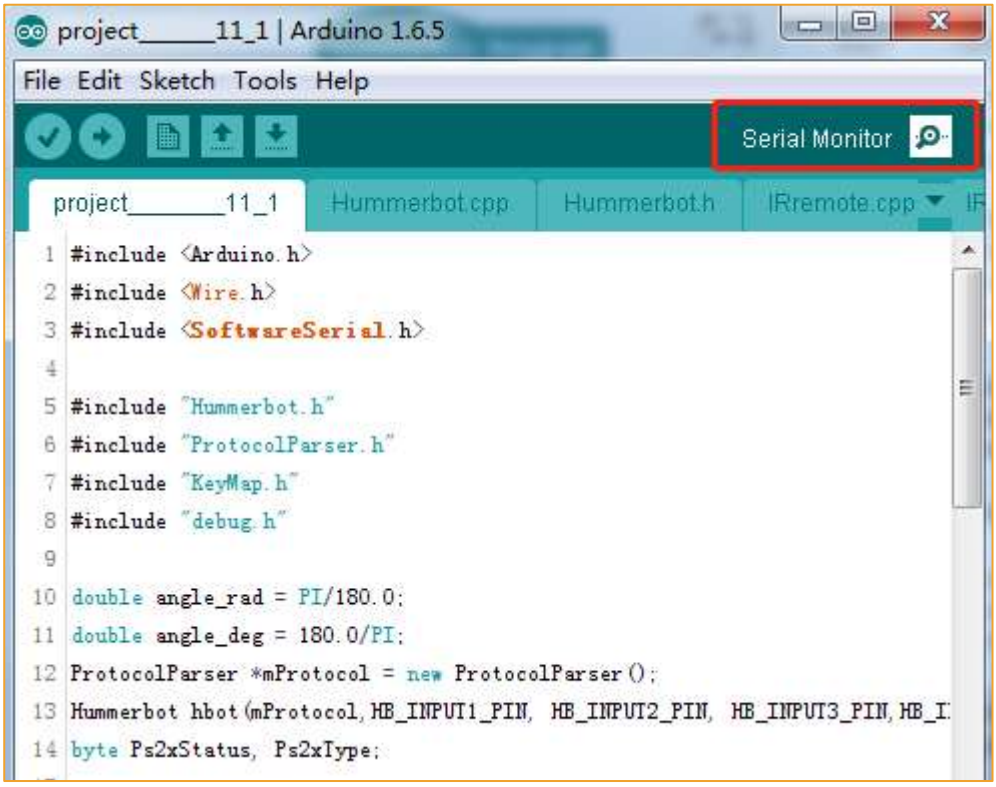


Figure 2-18

After opening the Auduino software, click on the serial monitor in the upper right corner, as shown in Figure 2-19.



Fgiure 2-19

Open the serial monitor, at this point we will see the continuous printing of Hello Word on the serial monitor, as shown in Figure 2-10.

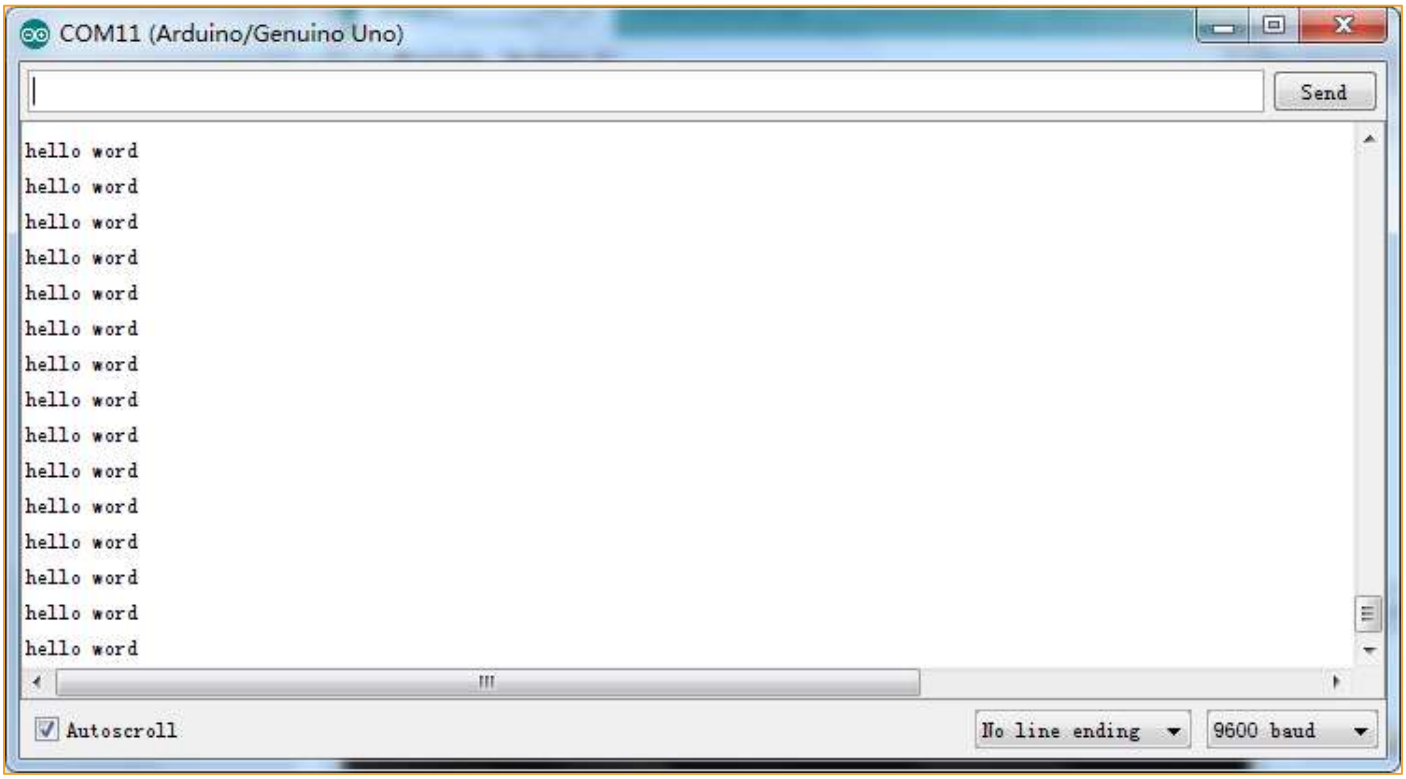


Figure 2-20

Chapter 3 Let robot move

3.1 DC Motor

3.1.1 DC Motor Principle

The reason why cars can move is because there are engines that power the cars. The robot also has a DC motor module that allows it to move. What is a motor? In our scientific textbook, there is an introduction to electromagnetic induction. The motor is turned by electromagnetic induction. It has a core with a copper wire inside and a rotor. When the core is energized, there is electromagnetic induction for the rotor to move it up, this is the motor.

The MotorDriverBoard has two DC motor interfaces, We can use a DC motor to control the building blocks to turn the DC motor up, which will drive the MotorDriverBoard robot to move.

3.1.2 DC Motor Test

In the robot module, we can find the three control blocks of the DC motor module, which are to set the motor pin blocks, set the direction, speed building blocks, set the turning angle and speed building blocks, as shown in Figure 3-1.

Figure 3-1

Write the following program on mblock, they can control the motor forward and reverse respectively. Four DC motors can be connected to the MotorDriverBoard. The M1, M2, M3, and M4 on the board correspond to the motor numbers 1, 2, 3, and 4.

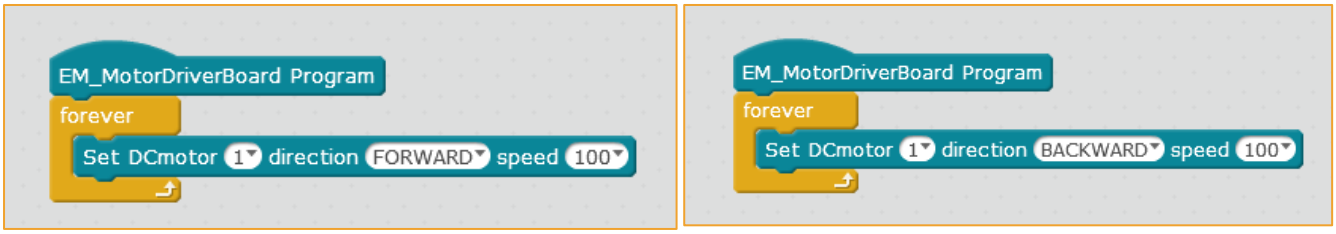


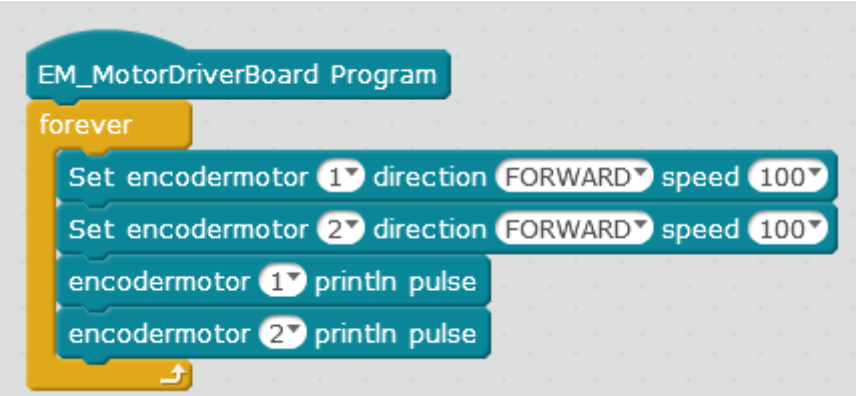
Figure 3-2

This demo can be opened and tested in MotorDriverBoard\Scratch\MotorDriverBoard_demo\DCmotor_demo.sb2.

3.2 Code Motor

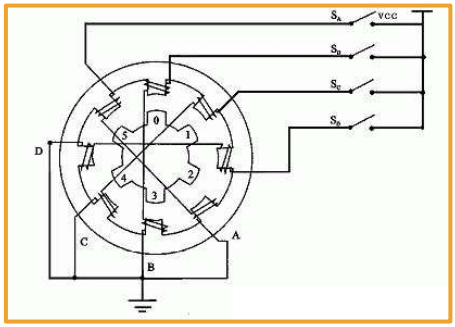
An encoder is a rotary sensor that converts angular or angular velocity into a series of electrical digital pulses. We can measure the displacement or velocity information through the encoder. Write the following program to test whether the forward rotation of the encoder motor is normal.

This demo can be opened and tested in :MotorDriverBoard \Scratch\ MotorDriverBoard_demo\EncoderMotor_demo.sb2.

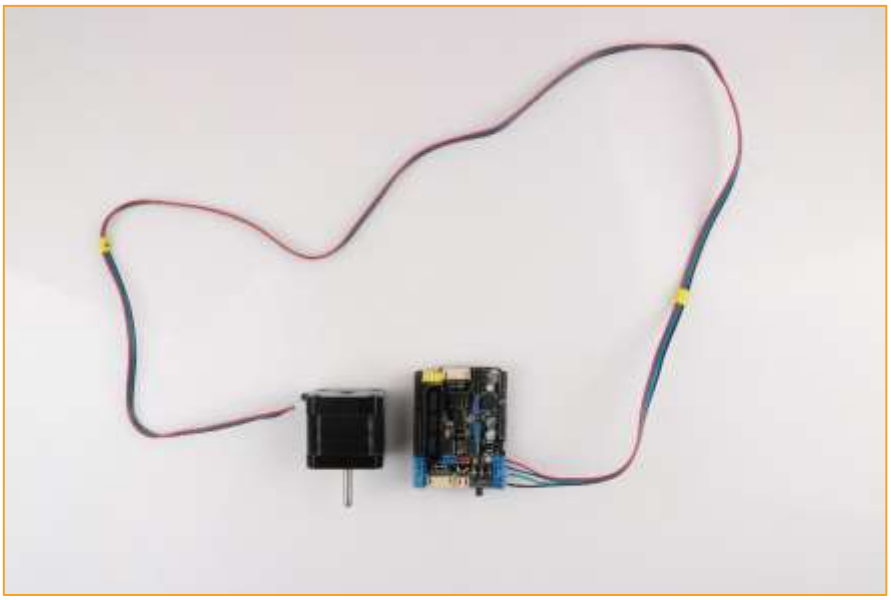


3.3 Stepper Motor

Stepper motor is an open-loop control motor that converts electrical pulse signals into angular displacement or line displacement. It is the main actuator in modern digital program control systems and is widely used. In the case of non-overloading, the speed and stop position of the motor depend only on the frequency of the pulse signal and the number of pulses.



There are many pairs of magnetic poles inside the stepping motor. If the energization state remains unchanged, the stepping motor will remain in a certain fixed state. Only when the energization state of each magnetic pole pair is constantly changed, the stepping motor can be continuously rotated. Therefore, the stepping motor cannot be directly connected to the DC or AC power supply, and a dedicated driving power supply (stepping motor driver) must be used. The controller (pulse signal generator) can control the angular displacement by controlling the number of pulses to achieve the purpose of accurate positioning. At the same time, the speed and acceleration of the motor rotation can be controlled by controlling the pulse frequency, thereby achieving the purpose of speed regulation.



Write a graphical program as shown below to test the operation of the stepper motor .This demo can be opened and tested inMotorDriverBoard\Scratch\MotorDriverBoard_demo\StepperMotor_demo.sb2

3.4 Ultrasonic module

3.4.1 Ultrasonic principle

Ultrasonic sensors are devices that detect the distance by transmitting ultrasonic waves. Ultrasound is an inaudible sound wave that has the property of returning to an object. The ultrasonic sensor has two "eyes", one of which "excites" the ultrasonic while the other "eye" receives the ultrasonic that is emitted by the obstacle. When one eye emits the ultrasonic, it starts timing and waits for the other one. When the eye receives the returned ultrasonic, it stops counting. In mathematics, we learned the distance = speed × time, then the ultrasonic measured distance = the speed of the ultrasonic × (time counting time ÷ 2); this can calculate the distance.

Figure 3-3

3.4.2 How to use the ultrasonic module

We found the control block of the ultrasonic module in the control module of the robot, as shown in Figure 3-4, which can be used to detect the distance between the robot and the obstacle ahead. Note: When using the ultrasonic module, firstly we need to select the mode control block and set the mode to the ultrasonic obstacle avoidance mode. When the obstacle completely blocks the ultrasonic module or the ultrasonic module faces far away, the ultrasonic emitted by the ultrasonic module can't receive so the distance between the robot and the obstacle cannot be measured. The ultrasonic module installed on the robot has a detection range of 5CM~400CM.

Figure 3-4

3.4.3 Ultrasonic obstacle avoidance module test

We can firstly write a program to test the ultrasonic obstacle avoidance module, use the serial port to print. When we are close to the ultrasonic obstacle avoidance module and away from the ultrasonic obstacle avoidance module, we observe the ultrasonic measurement distance of the serial port printing, you can see the ultrasound more intuitively. The process of measuring the distance by the obstacle avoidance module, let us write a test program firstl.This democan be opened and tested in MotorDriverBoard\Scratch\MotorDriverBoard_demo\ultrasonic_demo.sb2.

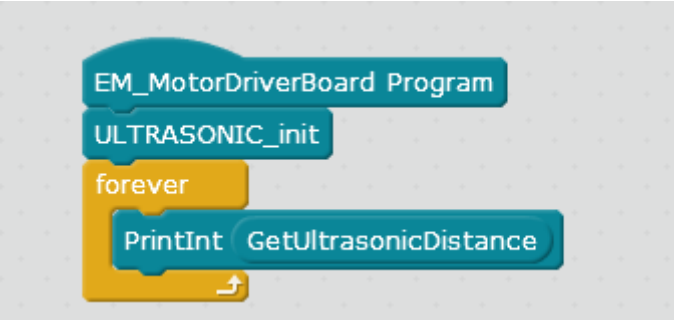


Figure 3-5

We write the program as shown in Figure 3-5, and then upload the program successfully, open the serial port, and then move the hand close to or away from the ultrasonic module, the serial port will print the corresponding distance.

3.4.4 Servo working principle

The servo is mainly composed of the following parts: steering wheel, reduction gear set, position feedback potentiometer, DC motor, control circuit, etc., as shown in Figure 3-6. The Bumblebee, Optimus Prime we saw in the movie, the joints of these robots are all need the servo to control, especially the mechanical sound of the clacking when the robot is walking is made by the robot when its servo rotates. And Figure 3-6 shows the physical map of the most commonly used SG90 servo at present.

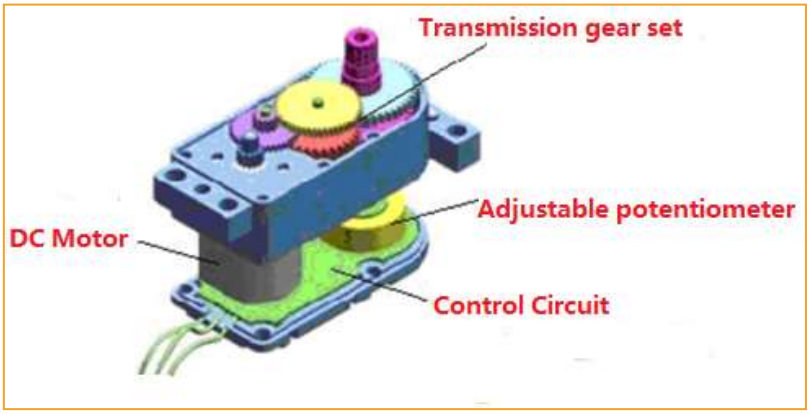


Figure 3-6 Schematic diagram of the servo

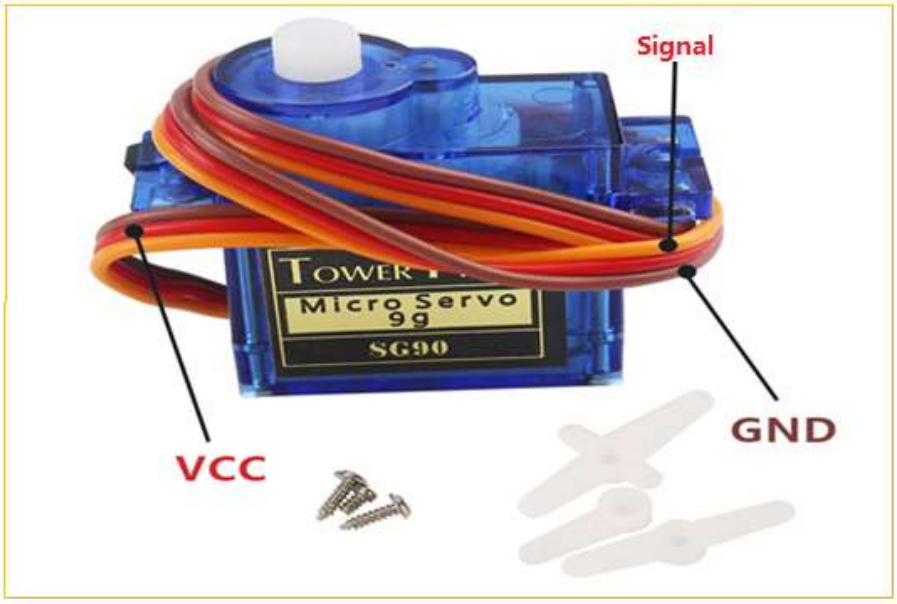


Figure 3-7 Physical map of the servo of SG90

When the servo control circuit board receives the control signal from the signal line, the motor is controlled to rotate, and the motor drives a series of gear sets, and then decelerates and drives to the output steering wheel. The workflow is: control signal → control circuit board → motor rotation → gear set deceleration → servo rotation → position feedback potentiometer → control circuit board feedback.

3.4.5 Servo Test

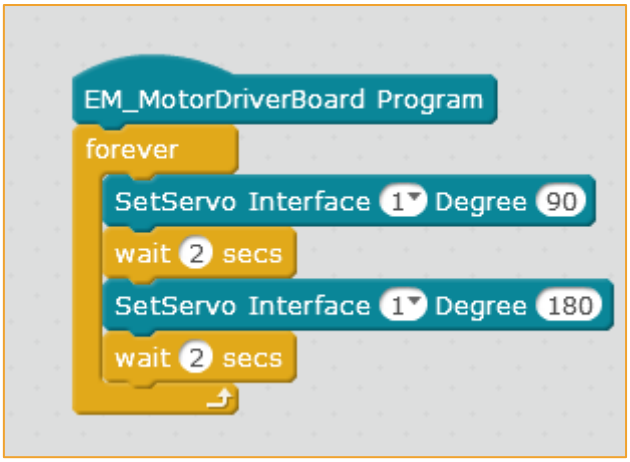


Figure 3-8

The above is the test program of the servo. Connect the servo interface 1. After uploading the program to the MotorDriverBoard, connect the power supply. You can see that the servo repeats the left and right turns every two seconds. (180 degrees is right and 0 degrees is left.) This demo can be opened and tested in MotorDriverBoard\Scratch\MotorDriverBoard_demo\servo_demo.sb2.

3.5 RGB light module

MotorDriverBoard is equipped with two left and right RGB lights. These two modules are called RGB modules. R is the abbreviation of Red, G is the abbreviation of Green, B is the abbreviation of Blue. Red, green and blue are what we call the three primary colors. The RGB module is capable of emitting different colors of light through the control of the building blocks. We can find the control blocks of the RGB module in the robot module, as shown in Figure 3-9.

Figure 3-9

We can use the RGB module to write a program that keeps changing the flashing light as shown in Figure 3-16.This demo can be opened and tested in MotorDriverBoard\Scratch\MotorDriverBoard_demo\RGB_demo.sb2.

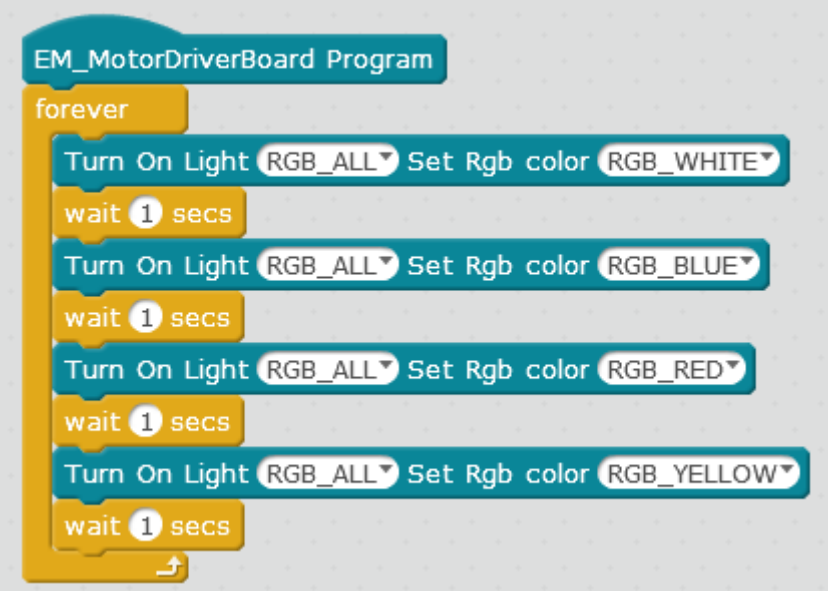


Figure 3-10

3.6 Buzzer module

3.6.1 Introduces through life

By thinking about the way common Musical Instruments are played, we can learn that sound is the sound wave generated by the vibration of objects and the surrounding air. Different frequencies of vibration will make different sounds

3.6.2 Learn about modules and use them

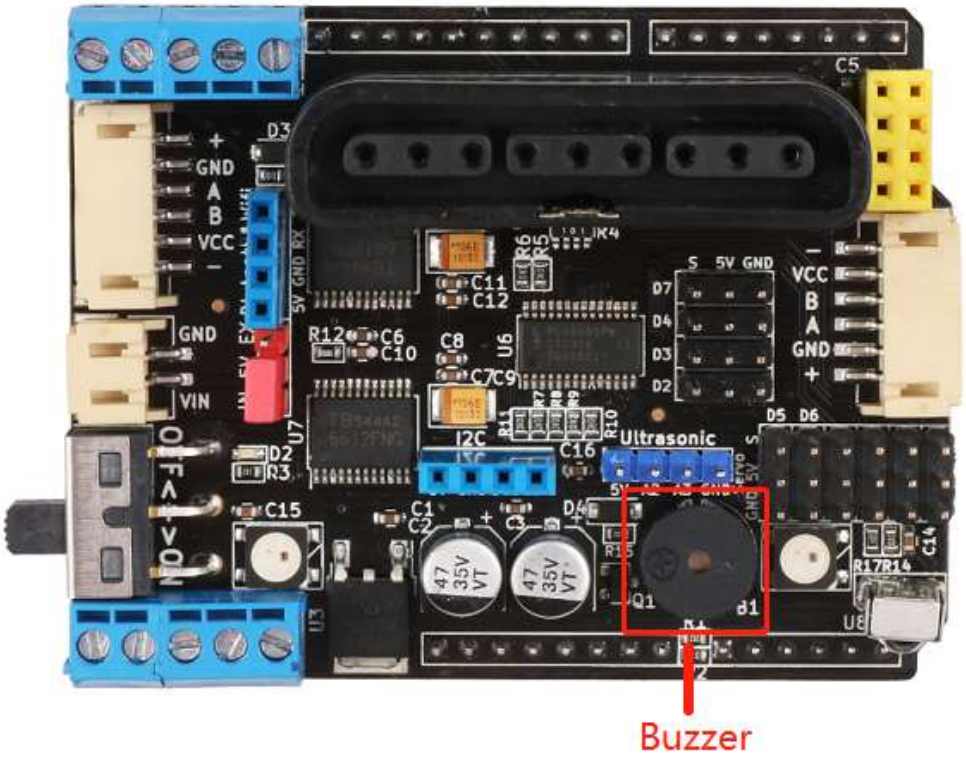


Figure 3-11

We can find the control blocks of the buzzer module in the robot module, as shown in Figure 3-11.

Figure 3-12

We can use the buzzer module to write a simple sounding program as shown in Figure 3-13. Of course, we can also select various desired sounds in the drop-down box.This demo can be opened and tested in MotorDriverBoard\Scratch\MotorDriverBoard_demo\buzzer_demo.sb2.

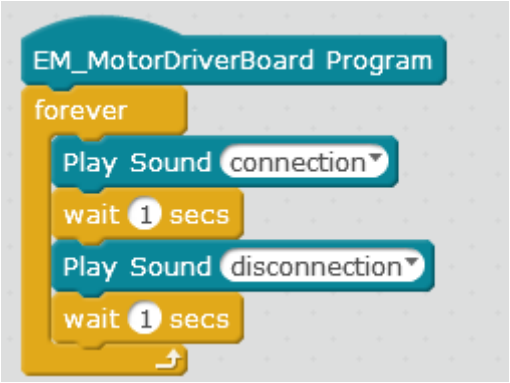


Figure 3-13

Chapter 4 Infrared Remote Control Robot

4.1 Infrared remote control principle

The remote control system generally consists of a remote control (transmitter) and a receiver. When you press any button on the remote control, the remote control will issue an instruction. After receiving the command from the remote control, the receiver will send the command and pass to the robot's brain, the robot thinks about what kind of action to do according to the remote command, and then controls its limbs (four wheels) to do the corresponding action, the remote control and receiver are shown in Figure 4-1

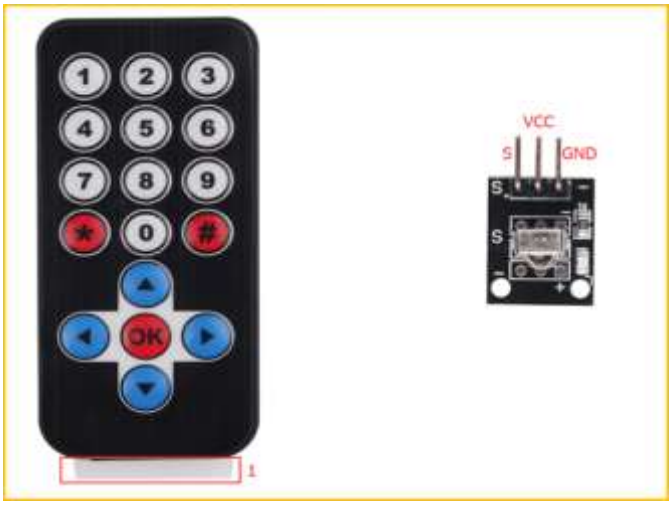


图 4-1

4.2 How to use the remote control

To control the robot through the remote control, we firstly need to set the control mode to the infrared remote control mode. Nextly set the infrared remote control receiving pin according as the actual wiring port, and then define the effect of pressing each button of the remote control, so when we press the button of the remote control, the robot will do the actions defined in our program.

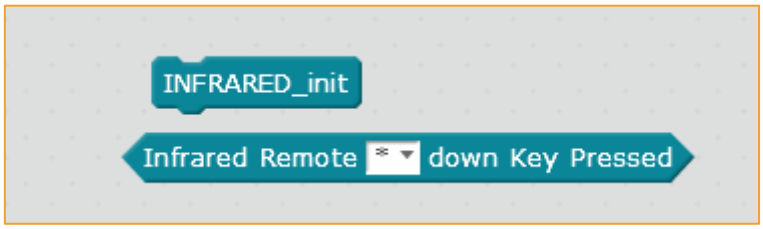


Figure 4-2

4.3 Infrared remote control test

We can firstly write a program to test the infrared remote control, using the RGB light color, when the remote control button is pressed, we observe the color of the RGB light, you can see which button of the infrared remote control is pressed, let us write a test program firstly .

Figure 4-3

We write the program as shown in Figure 4-3, and then upload the program successfully, open the serial port, and then press the remote control to press the button on the robot, the corresponding RGB light color will be displayed.This demo can be opened and tested in MotorDriverBoard\Scratch\MotorDriverBoard_demo\IrRemote_demo.sb2.

Chapter 5 PS2 Module

5.1 PS2 Remote control handle principle

The PS2 handle consists of two parts: the handle (as shown in Figure 7-1) and the receiver (Figure 7-2). The handle needs two 7-cell 1.5V dry battery power supply, and the handle switch is turned ON. Under the condition that the handle hasn't searched the receiver, the light on the handle will flash continuously. Within

a certain period of time, if the receiver has not been searched, the handle will enter the standby mode, and the light on the handle will be extinguished. At this time, press the "START" button to wake up the handle. The working power of the receiver is 3~5V, it can't be reversed, and it can't overvoltage, otherwise it will burn out the receiver.

After the normal power-on, the handle and the receiver are automatically paired. When the pairing is not successful, the green light of the receiver flashes, and the light on the handle also flashes. After the pairing is successful, the green light on the receiver is always on, and the light on the handle is also always on. The button "MODE" (the handle batch is different, the above logo may be "ANALOG", but it will not affect the use), you can choose "red light mode"or"green light mode".

Figure 5-1

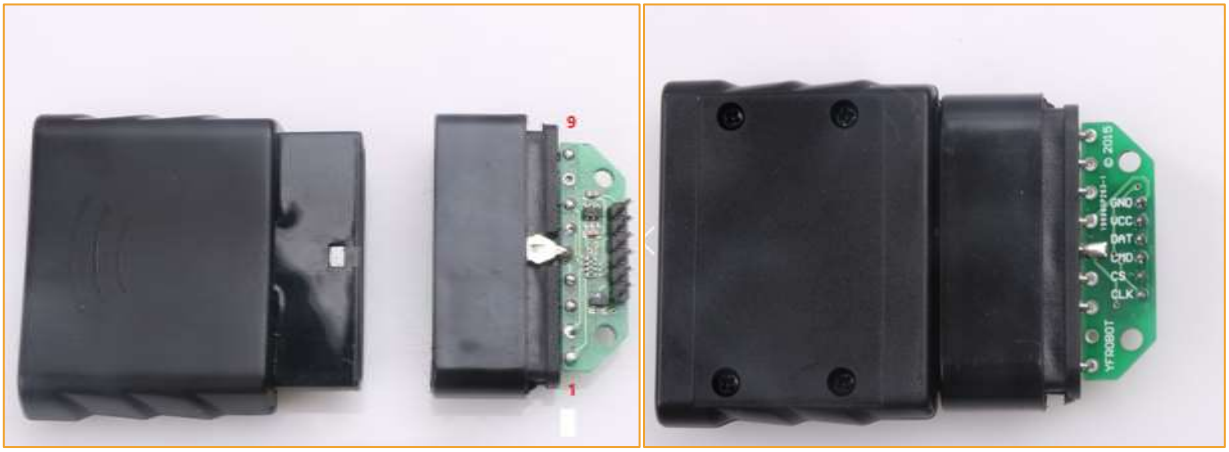


Figure 5-2

When the handle is connected to the receiver, we can use the handle to send the button commands. When the receiver receives these button commands, the robot's brain (the main control board) will follow the commands received to make his limbs (four wheels, the servo) and so on do the corresponding action.

5.2 PS2 Remote control test

We can first write a program to test the PS2 remote control. When the PS2 remote control button is pressed, the RGB light is on. When we observe the color of the RGB light, we can see which button of the PS2 remote control is pressed. The following figure is a test procedure.This demo can be opened and tested in MotorDriverBoard\Scratch\MotorDriverBoard_demo\PS2_demo.sb2.

Figure 5-3

Chapter 6 Nrf24L01 Module

6.1 Nrf24L01 Introduction

Nrf24L01 is a wireless transmission module for wireless data transmission. The NRF24L01+ module is described as below.

- The nRF24L01+ module is a 2.4G wireless communication module developed by Nordic based on the nRF24L01 chip.
- Adopt FSK modulation and integrate Nordic's own Enhanced Short Burst protocol.
- Point-to-point or 1-to-6 wireless communication can be achieved.
- Wireless communication speed can reach up to 2M (bps).
- The NRF24L01 has four operating modes: transceiver mode, configuration mode, idle mode, and shutdown mode.

6.2 Drive NRF24L01+ module

Insert the nRF24L01+ module into the corresponding interface on the PS2X&Motor Driver Board driver board (as shown in Figure 6-1). For the stable reception of Nrf24L01 data, it is recommended to connect 10uf capacitor between VCC and GUD as shown in Figure 6-2. (Example program file path: git\MotorDriverBoard\ArduinoDemo\nRF24l01+), which can send and receive data test to each other through two devices

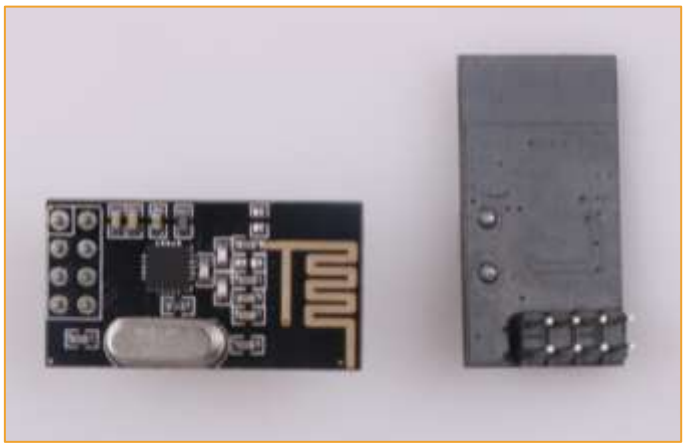


Figure 6-1: Nrf24l01+ physical map

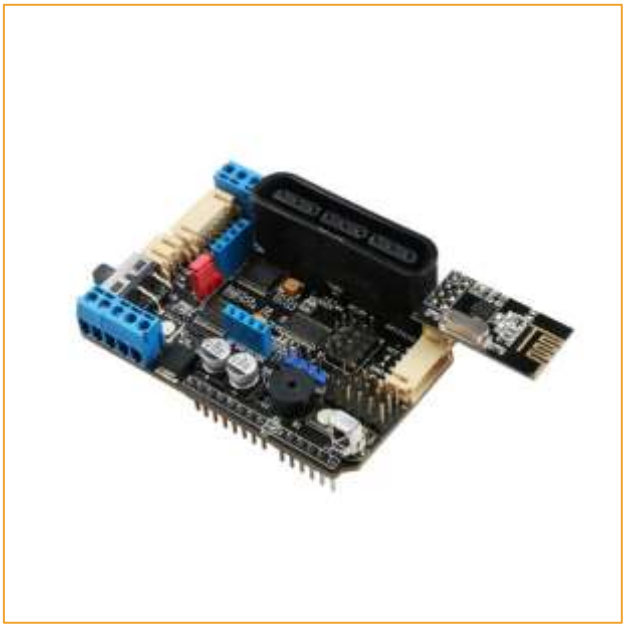


Figure 6-2

6.3 NRF24L01+test demo

When writing Nrf24L01+ test with mBlock, you need to use two MotorDriverBoard and insert two Nrf24L01+ modules into the corresponding excuses respectively, then use the building blocks to make two programs in mBlock: one sending data program and the other receiving data program, as shown below.This demo can be opened and tested in MotorDriverBoard\Scratch\MotorDriverBoard_demo\NRF24L01_send_demo.sb2 and MotorDriverBoard\Scratch\MotorDriverBoard_demo\NRF24L01_receive_demo.sb2.

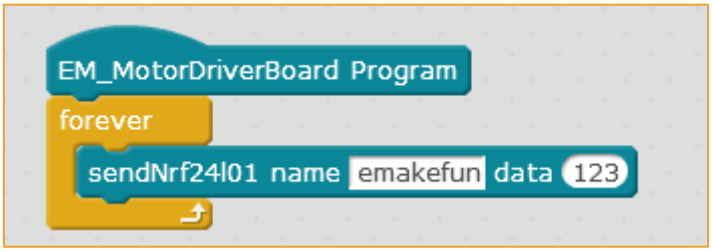


Figure 6-3 Send data

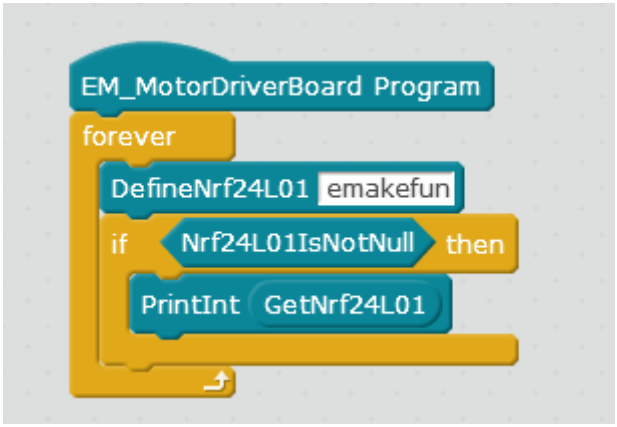


Figure 6-4 receive data

Compile with Arduino ide, open the serial monitor, you can observe that the data is successfully sent from Nrf24L01 to another Nrf24L01

Figure 6-5 send data“123”

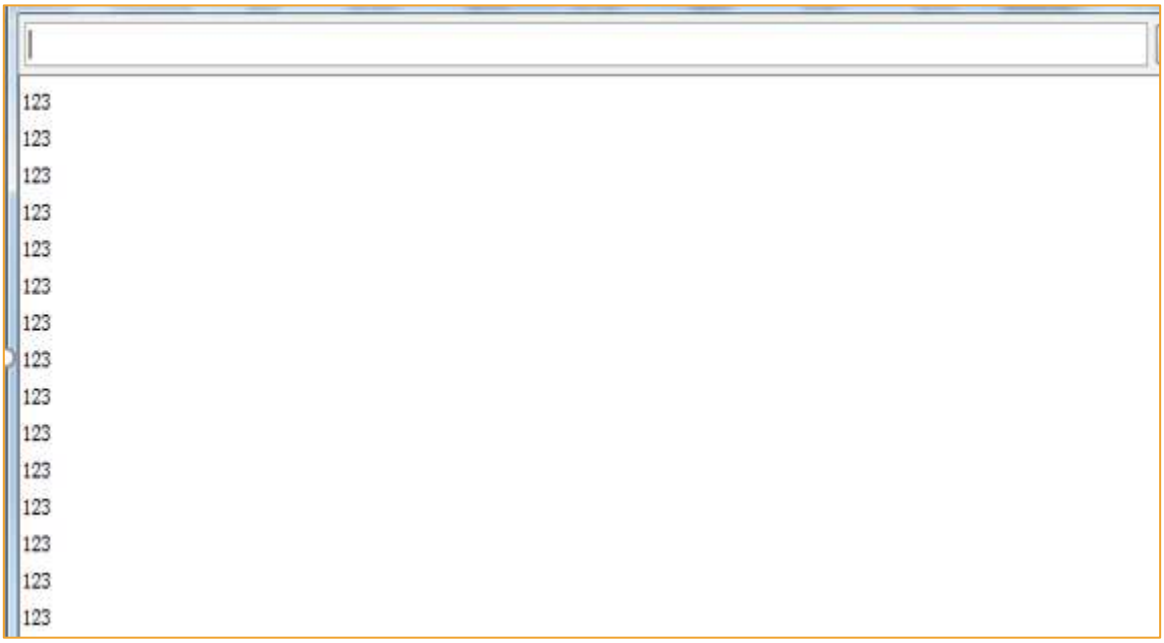


Figure 6-6 receive data “123”