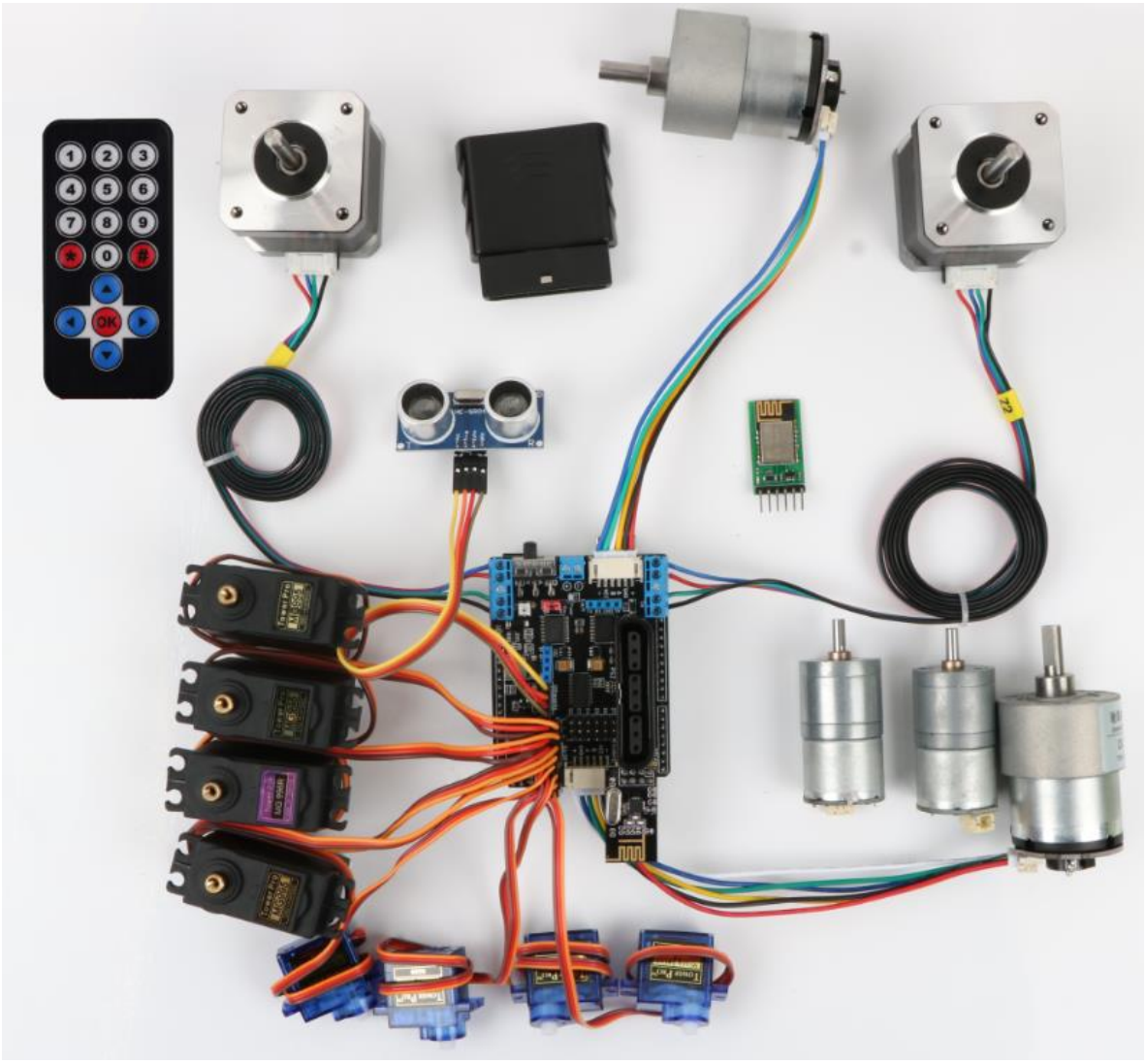
MotorDriverBoard MagicBlock Graphical Programming Course V.1.0



Revision of edition

date	Version number	description	author
2019-9-20	**V.1.0**	**create documents**	**Twisted**

Chapter 1 Understanding MotorDriverBoard and Its Programming Environment

1.1 Introduction of life

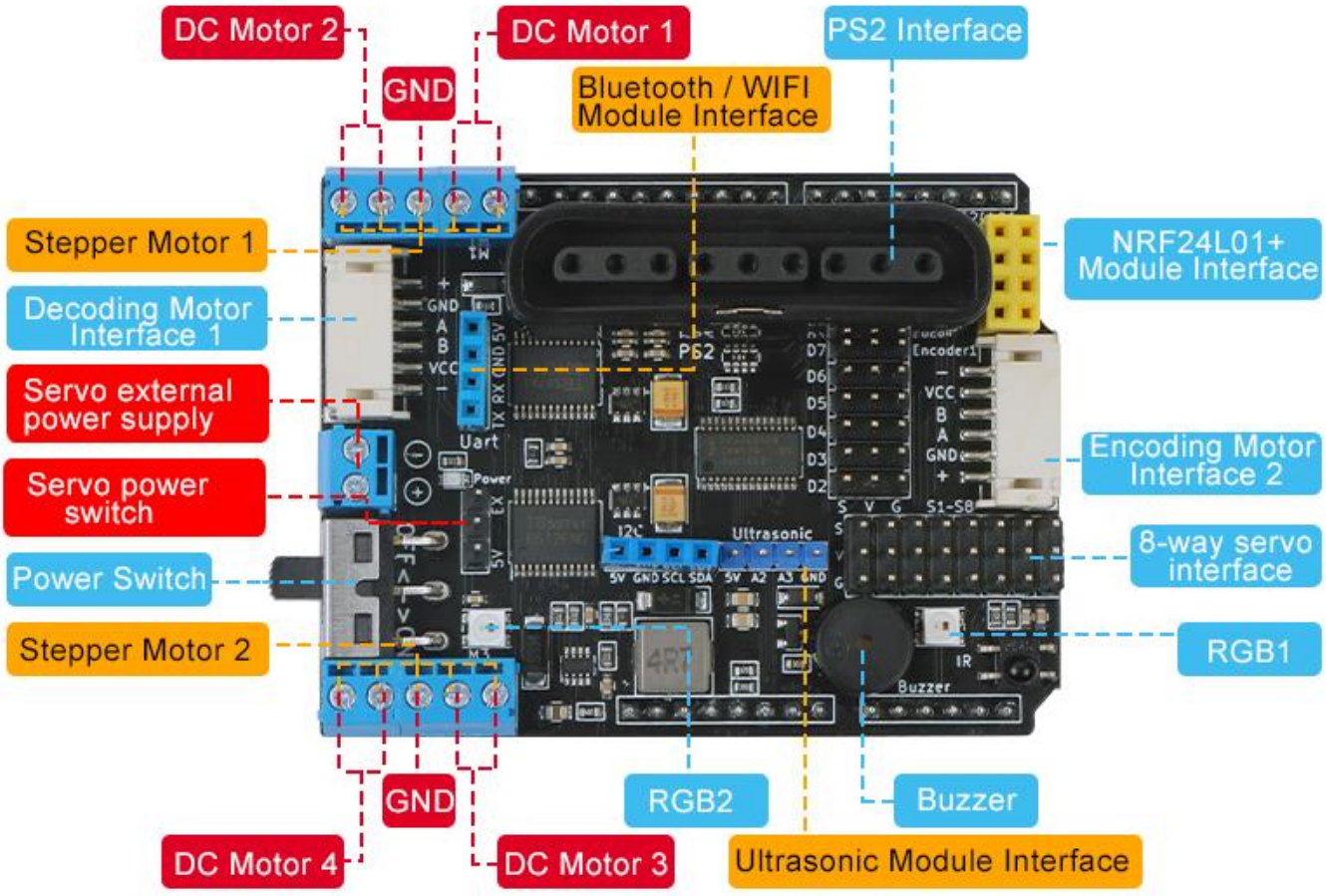
Hello, everyone. Welcome to the world of Arduino robots!

In our Arduino world, robots often need to move, so we need a special motor driver. We integrate many motor scenarios, sensor connections, wireless control, etc. Finally, we design a MotorDriverBoar 4.0 which can drive multiple motors, connect multiple sensors, support PS2X port, NRF24L01 + module.

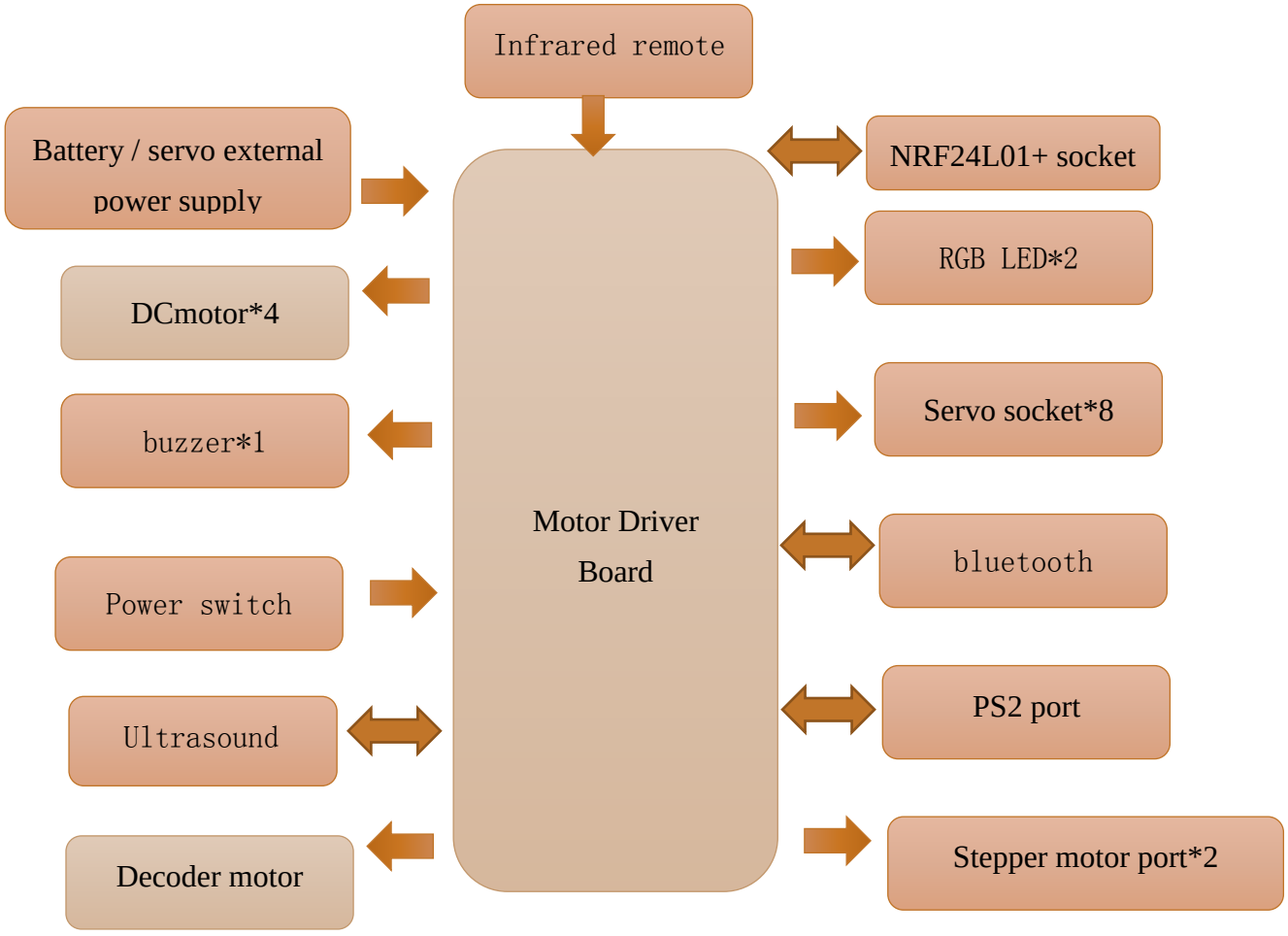
1.2 Summary

。The MotorDriverBoard can drive four DC motors, two coding motors, two stepping motors, eight steering motors (two external power supplies), and the driving current is up to 3A. The drive board is designed for Arduino UNO R3 motherboard and can be directly plugged into Arduino Uno. The motherboard integrates a passive buzzer, two RGB LED lights and one infrared receiver. Besides, PS2, Uart, I2C and ultrasonic obstacle avoidance module are reserved. It is very convenient to connect various sensor modules.

1.2.1MotorDriverBoard Introduction


DC Motor 2
DC Motor 1
PS2 Interface
GND
Bluetooth / WIFI Module Interface
Stepper Motor 1
NRF24L01+ Module Interface
Decoding Motor Interface 1
Servo external power supply
Encoding Motor Interface 2
Servo power switch
8-way servo interface
Power Switch
RGB1
Stepper Motor 2
GND
RGB2
Buzzer
DC Motor 4
DC Motor 3
Ultrasonic Module Interface

1.2.1MotorDriverBoard List of Function Modules



1.4 Introduction of Graphical Programming Software

1.4.1 Software installation

The installation package of MagicBlock, a graphical programming software based on scratch 2.0, has been placed in the MotorDriverBoard product data. It can be installed directly.

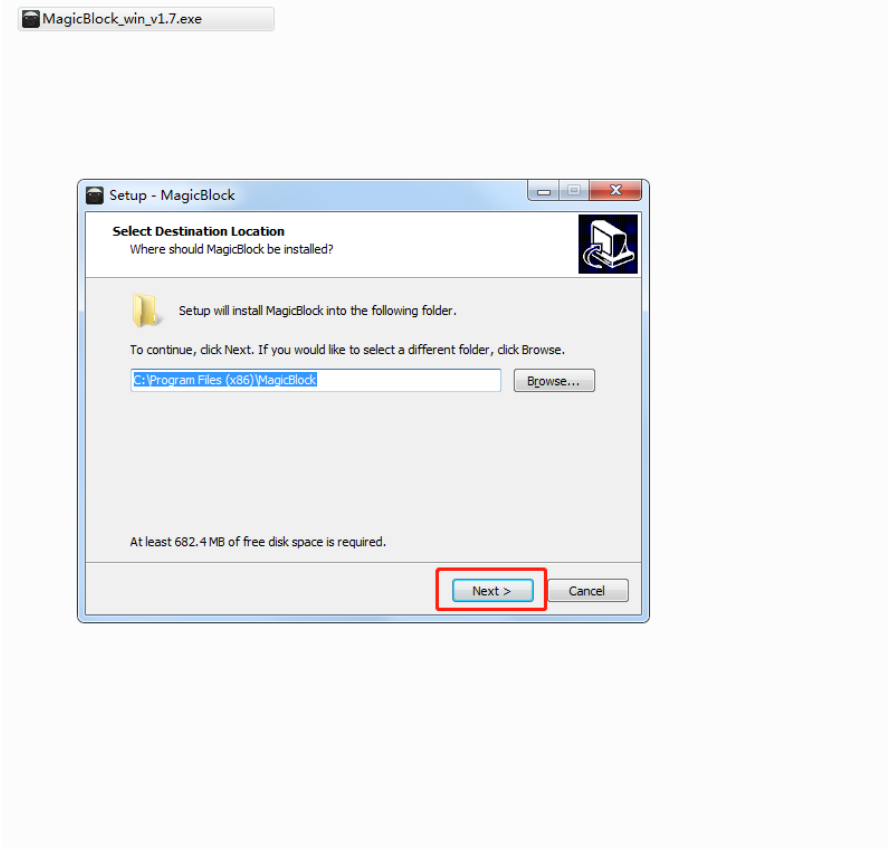


Figure 1-4-1-1

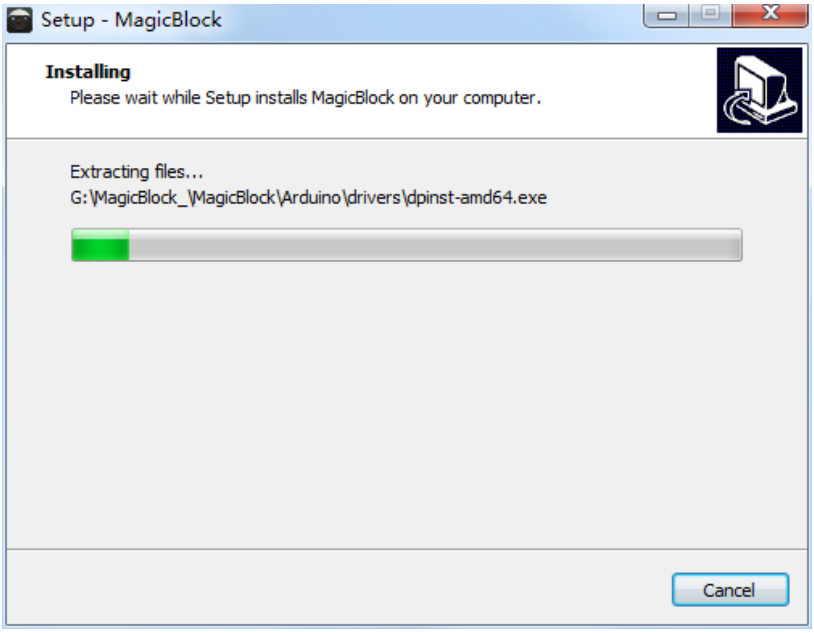


Figure 1-4-1-2

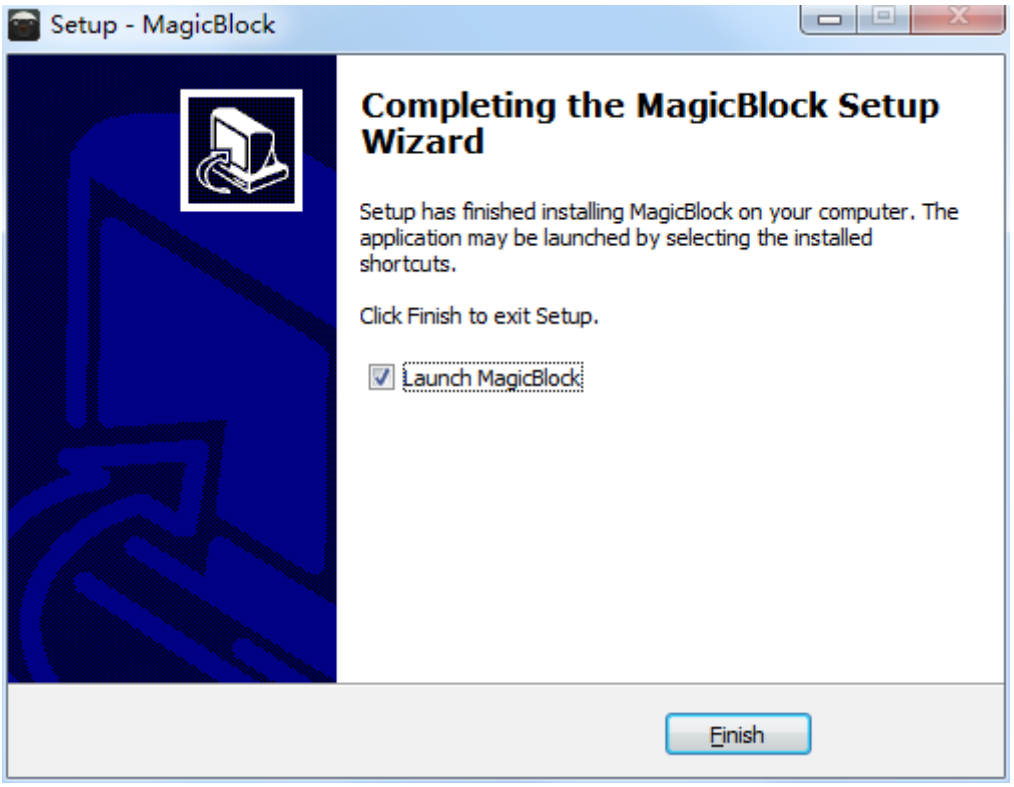


Figure 1-4-1-3

1.4.2 Introduction to Compiler Environment

We know that in order for a robot to move, besides the hardware cooperation, the more important thing is to write a program for it. We already know the hardware part of the robot, and then we will know its software part. MotorDriverBoard's software is programmed on MagicBlock, a graphical programming software. With this software, we can control the robot by writing all kinds of commands we want it to execute. MagicBlock's software interface is shown in Figure 1-4-2-1.

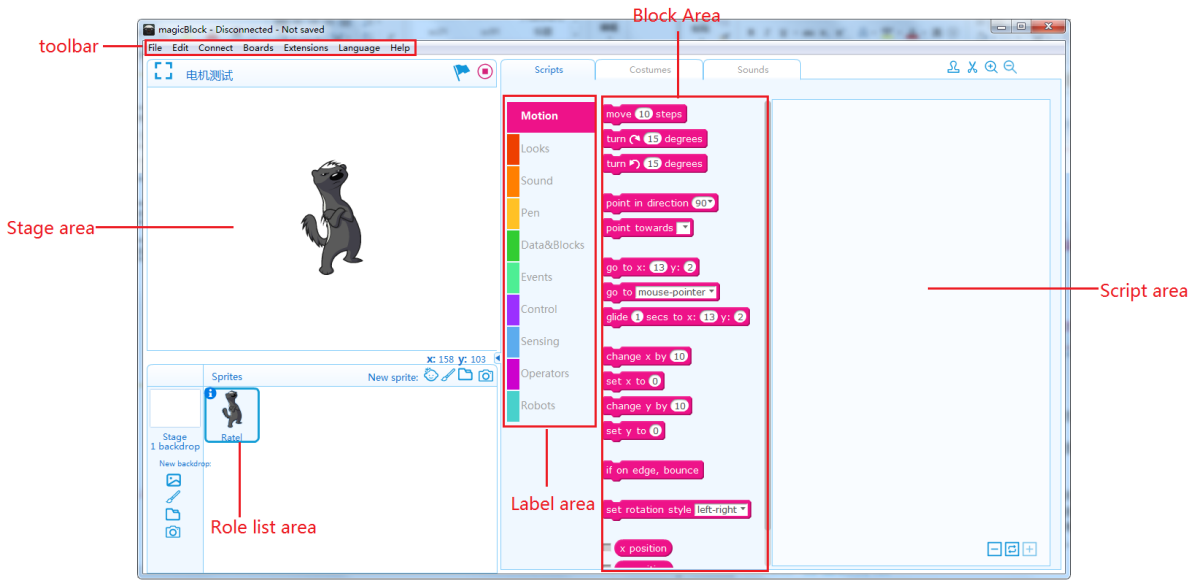


Figure 1-4-2-1

- **Toolbar：** For project files, software interface mode, serial connection and upload program, control board selection, software using language, software update operation area.
- **Stage area：** The role and role in the work, the interaction area between the role and the user, is the place to show the running effect of the program.
- **Roles list area：** All the role prototypes display areas, where you can see the name of the role, rotation direction, location, and so on.
- **Label area：** Contains script tab, styling tab, voice tab, which can operate on script, styling and voice of characters.
- **Block area：** The same type of building blocks are divided into the same module and given the same color. Each building block represents a control instruction.
- **Script area：** Programming area, the building blocks stacked in the script area can be programmed.

1.5 Introduction to Label Area

MagicBlock's label area is mainly divided into scripts (as shown in Figure 1-5-1), shapes (as shown in Figure 1-5-2) and sounds (as shown in Figure 1-5-3). The scripts are mainly control blocks to perform actions; under the model label, you can draw some graphics to add to the program; under the sound label, you can record some sounds by yourself or put them into the program. The main purpose of MotorDriverBoard is to control blocks under script labels.

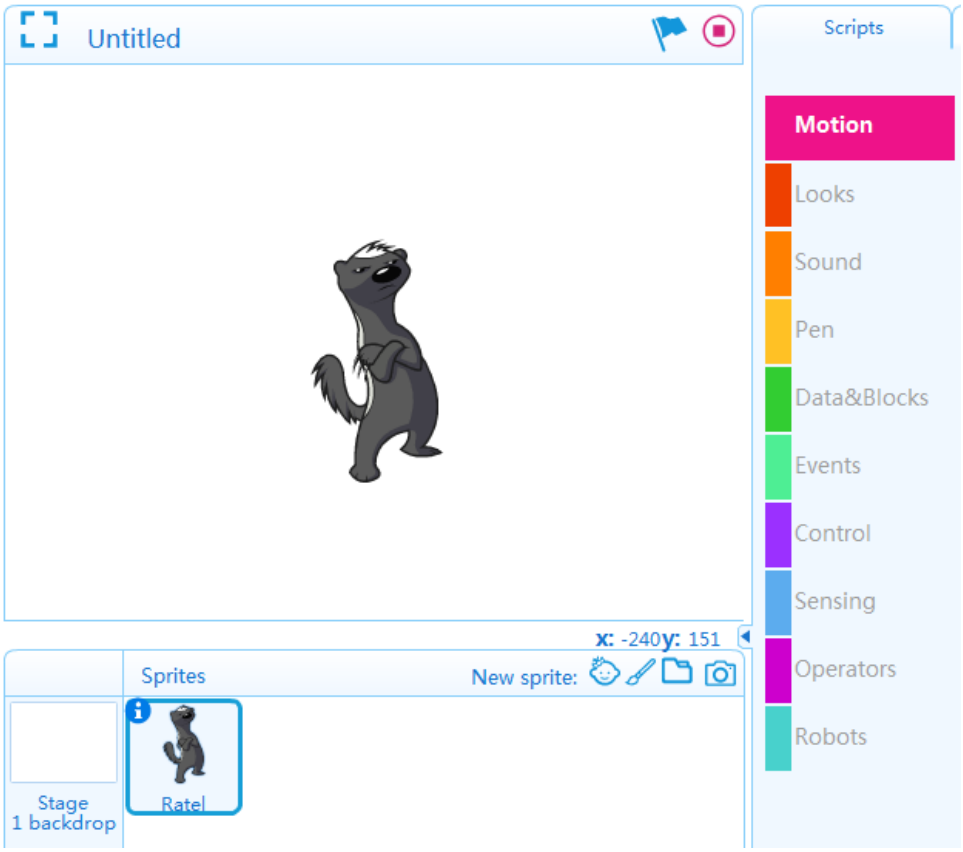

Untitled
Scripts
Motion
Looks
Sound
Pen
Data&Blocks
Events
Control
Sensing
Operators
Robots
x: -240 y: 151
Sprites
New sprite:
Stage
1 backdrop
Ratel

Figure 1-5-1

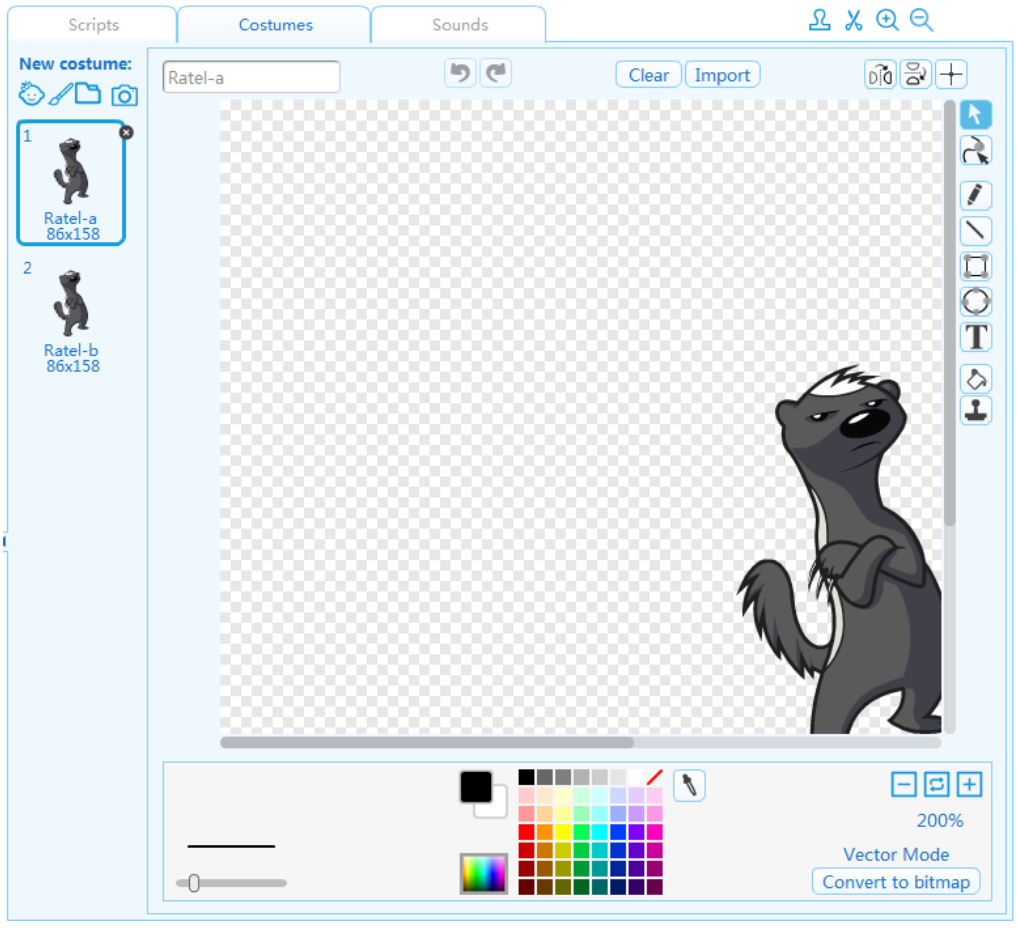

Scripts
Costumes
Sounds
New costume:
Ratel-a
Clear
Import
1
Ratel-a
86x158
2
Ratel-b
86x158
200%
Vector Mode
Convert to bitmap

Figure 1-5-2

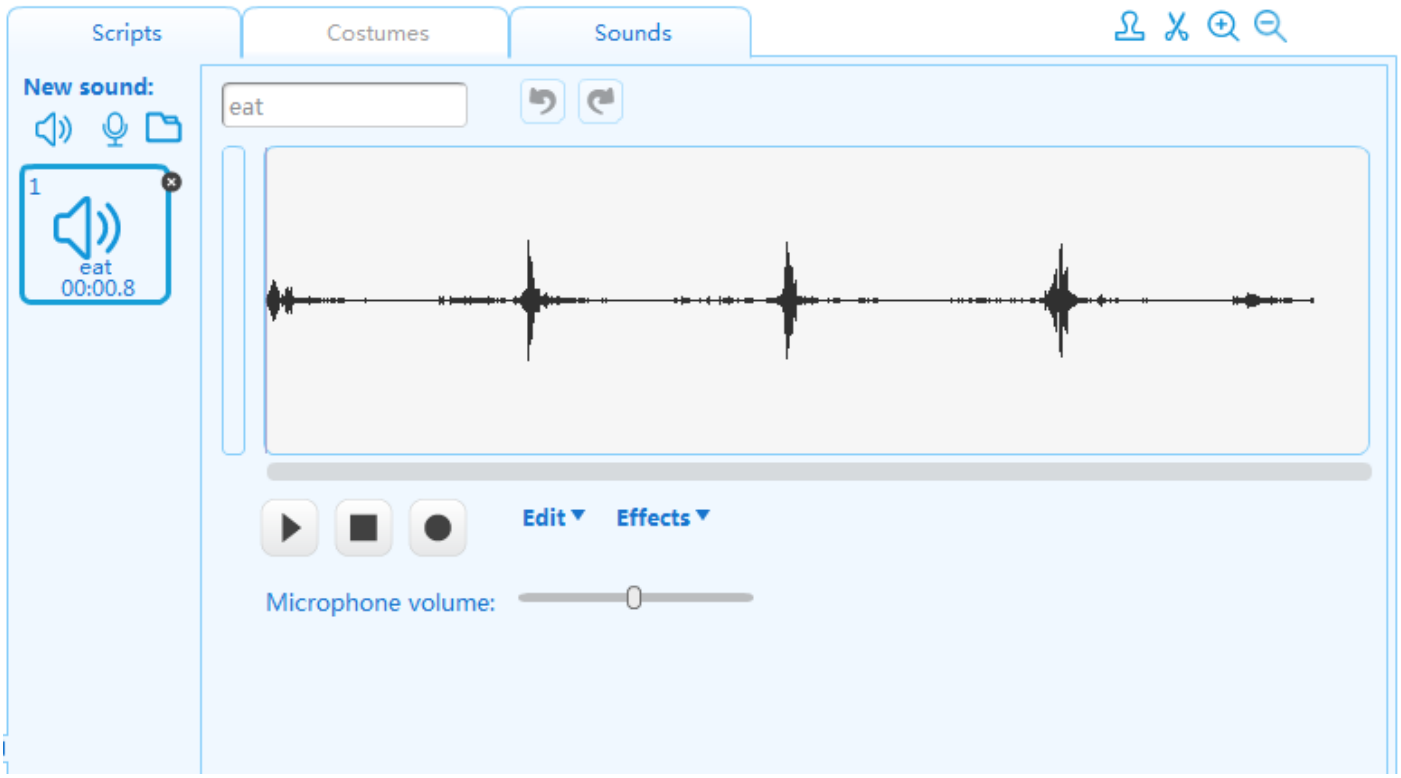


Figure 1-5-3

1.6 MotorDriverBoard Introduction of building blocks

MagicBlock's script label has 10 types of building blocks, including action, appearance, sound, brush, event, detection, which are some effects and data. Starterbox will not be used. If you are interested, you can try it out by yourself. Here we do not go into any more details. We mainly understand the three types of building blocks: control, digital and logical operations, and robotic module.

1) The control blocks are all the building blocks that control the execution process of the program (Fig. 1-6-1), the main program.

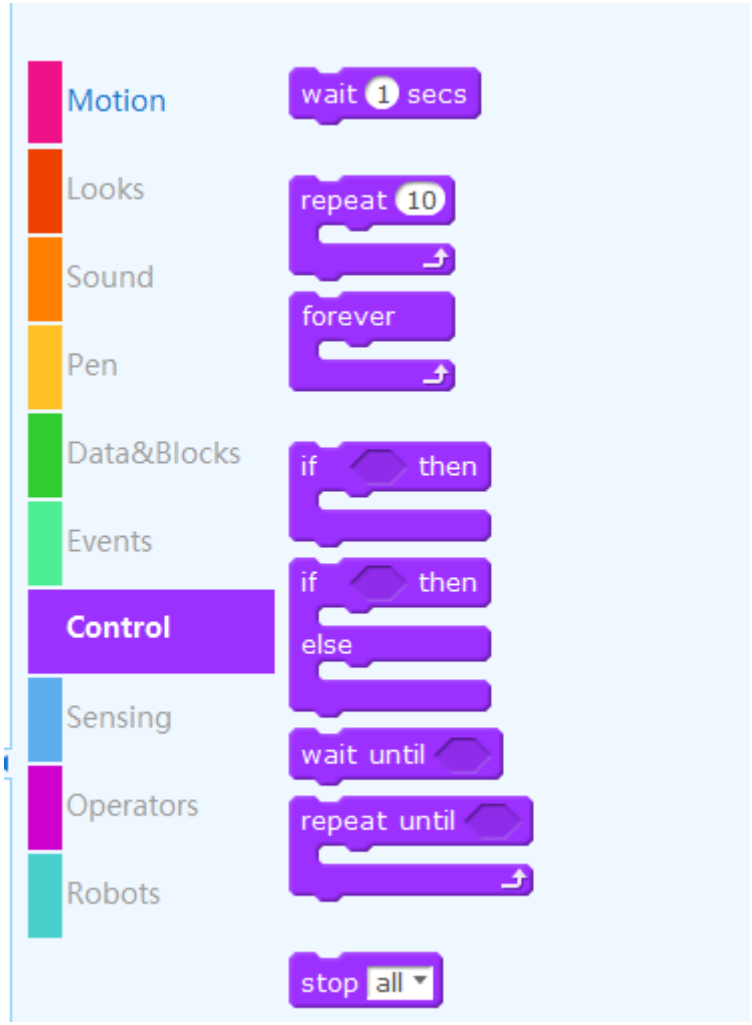
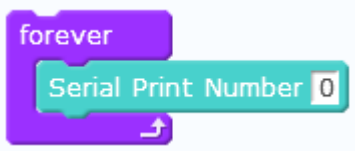


Figure 1-6-1

For example， 

This building block means to repeat the procedure in the building block all the time. Another example 

That is, the robot has been printing the number 0 in the serial port.

2) The main function of building blocks of numeric and logical operation types is to do mathematical operation as a condition of judgment, and to compare the size and logical judgment with, or with, or without, as shown in Figure 1-6-2.

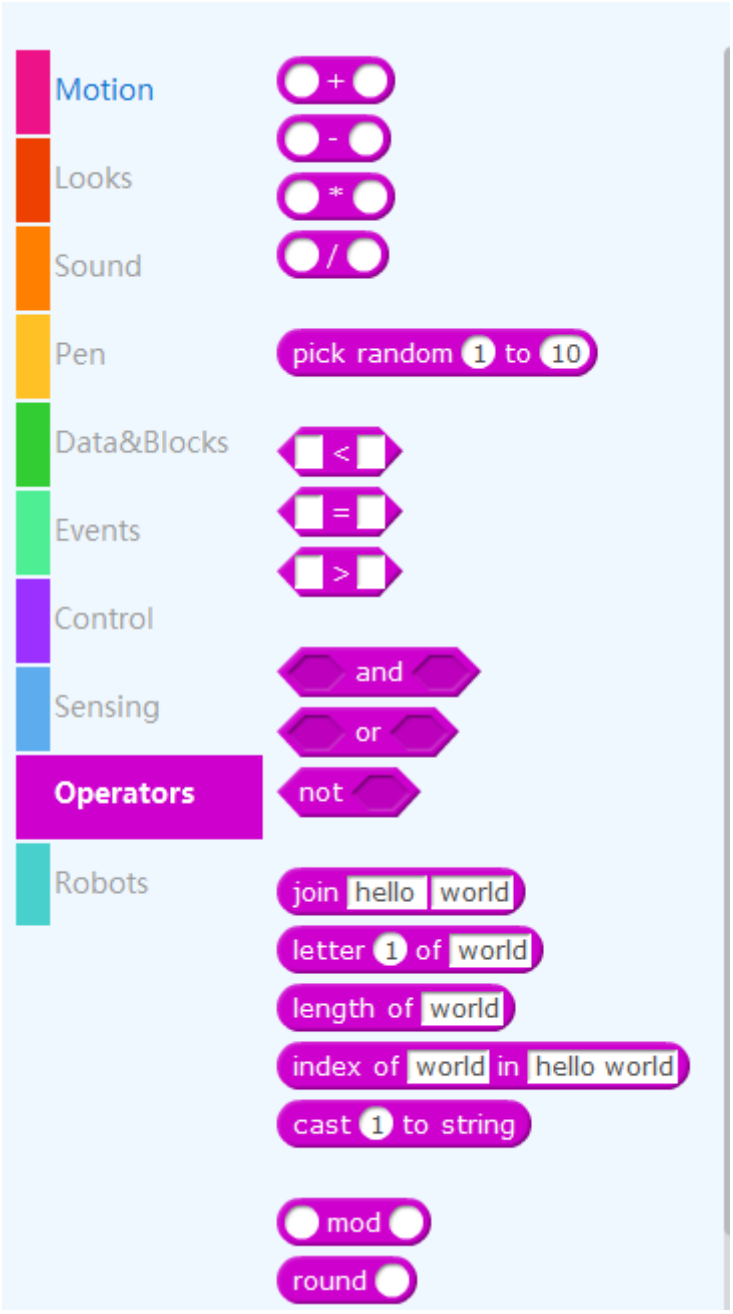
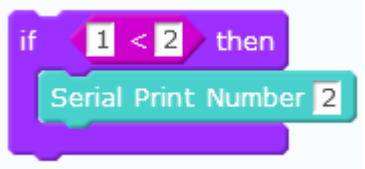
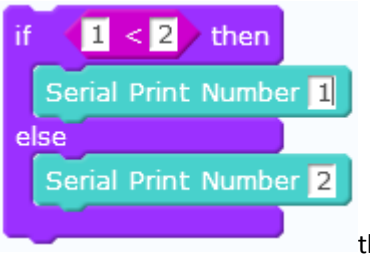


Figure 1-6-2

For example，  are the operations of judging size, when they are used together with control building blocks, digital and logical operations and robotic module building blocks.， You can write such a program.



， It means to judge  Is this condition valid? If the condition holds,

then the serial port printing number 2 is executed. Expanding, 

this program means 

Is this condition valid? If the condition is valid, the serial port print number 1 is executed, and if the condition is not valid, 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 and so on, as shown in Fig. 1-6-3 and Fig. 1-6-4.

Chapter 2 Initial Knowledge Programming

2.1 Hello word

If we want the robot controlled by MotorDriver Board to move, we need to store instructions (programs) in MotorDriver Board's brain (control board) beforehand. How to compile instructions for MotorDriver Board? Now let's take you through the experience of writing a Motor Driver Board print Hello word program.

2.1.1 Add MotorDriverBoard Library

Before programming, we need to add the MotorDriverBoard library. The steps are as follows:

1) Download the MotorDriverBoard library and save it on your computer. File name: MotorDriverBoard.zip
2) Open MagicBlock software and click "Extension Manager", as shown in Figure 2-1;
3) Add extensions in the lower right corner of the extension manager center, as shown in Figure 2-2;

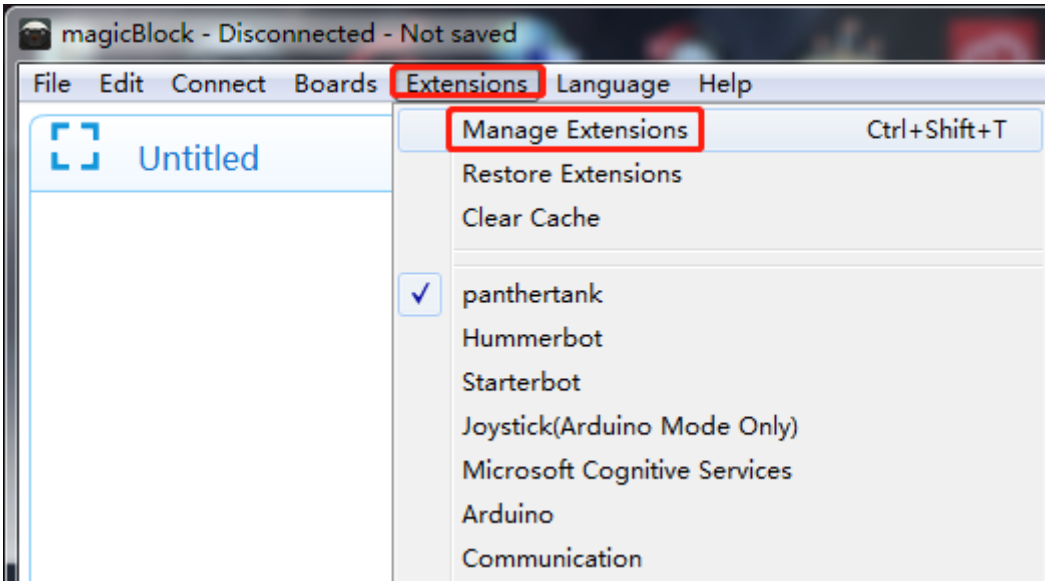


Figure 2-1

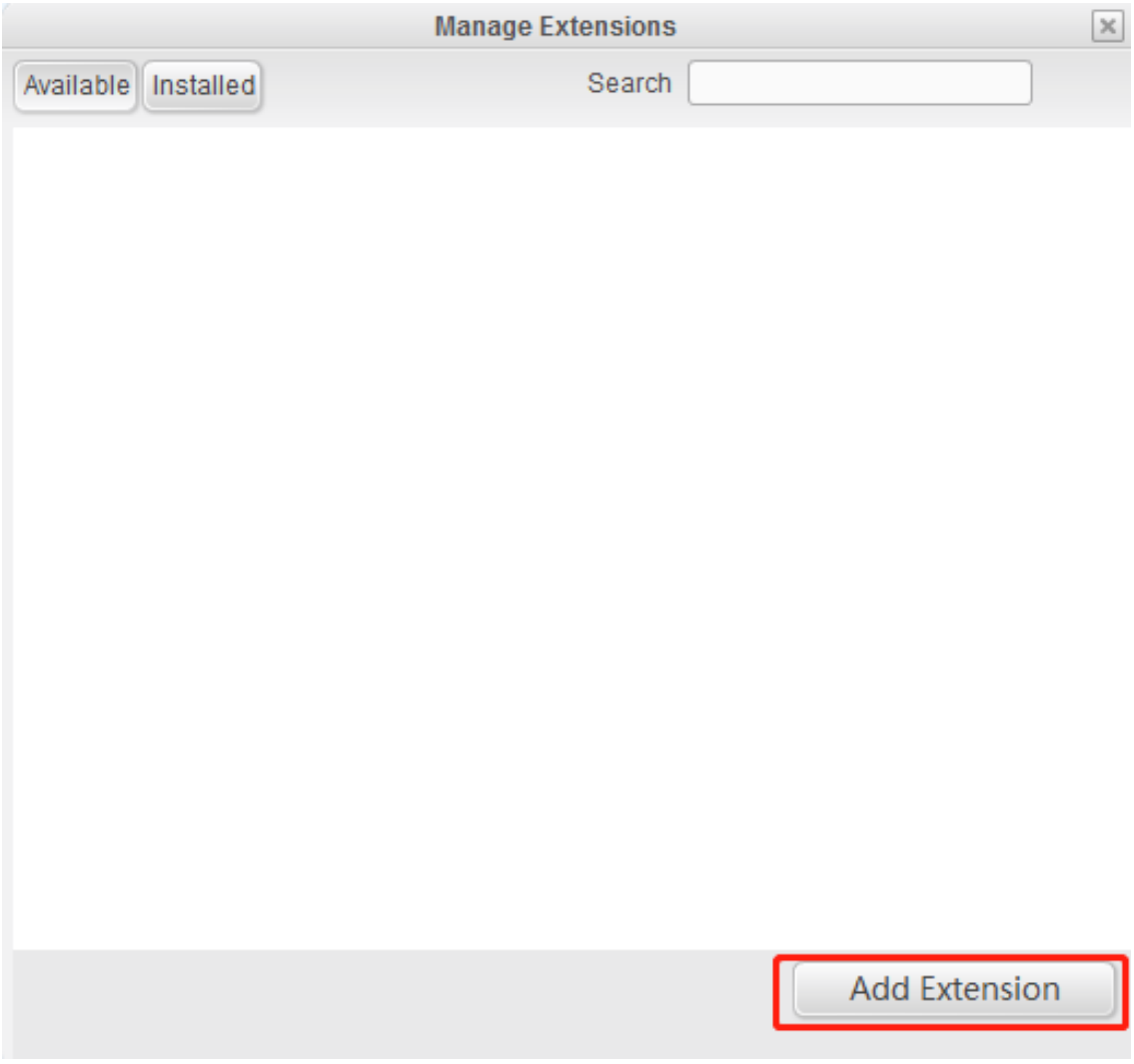


Figure 2-2

4) Select "Zip. file" for the file type, then select "MotorDriverBoard. zip" and click "Open",

as shown in Figure 2-3.

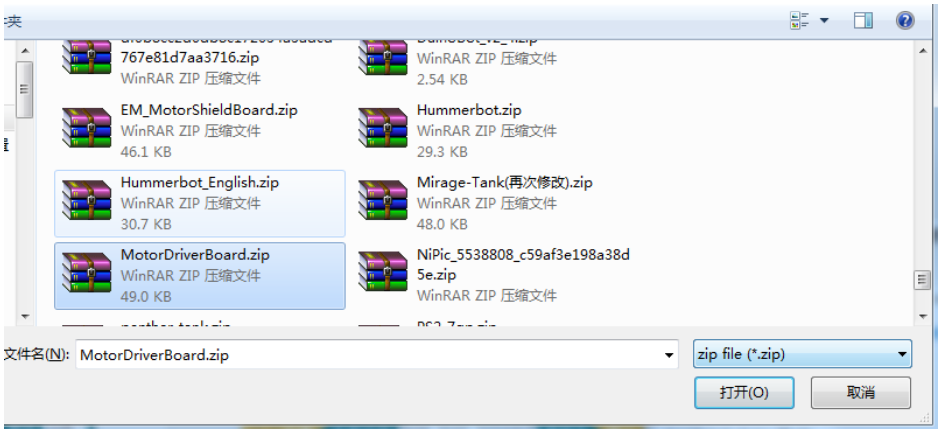


Figure 2-3

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

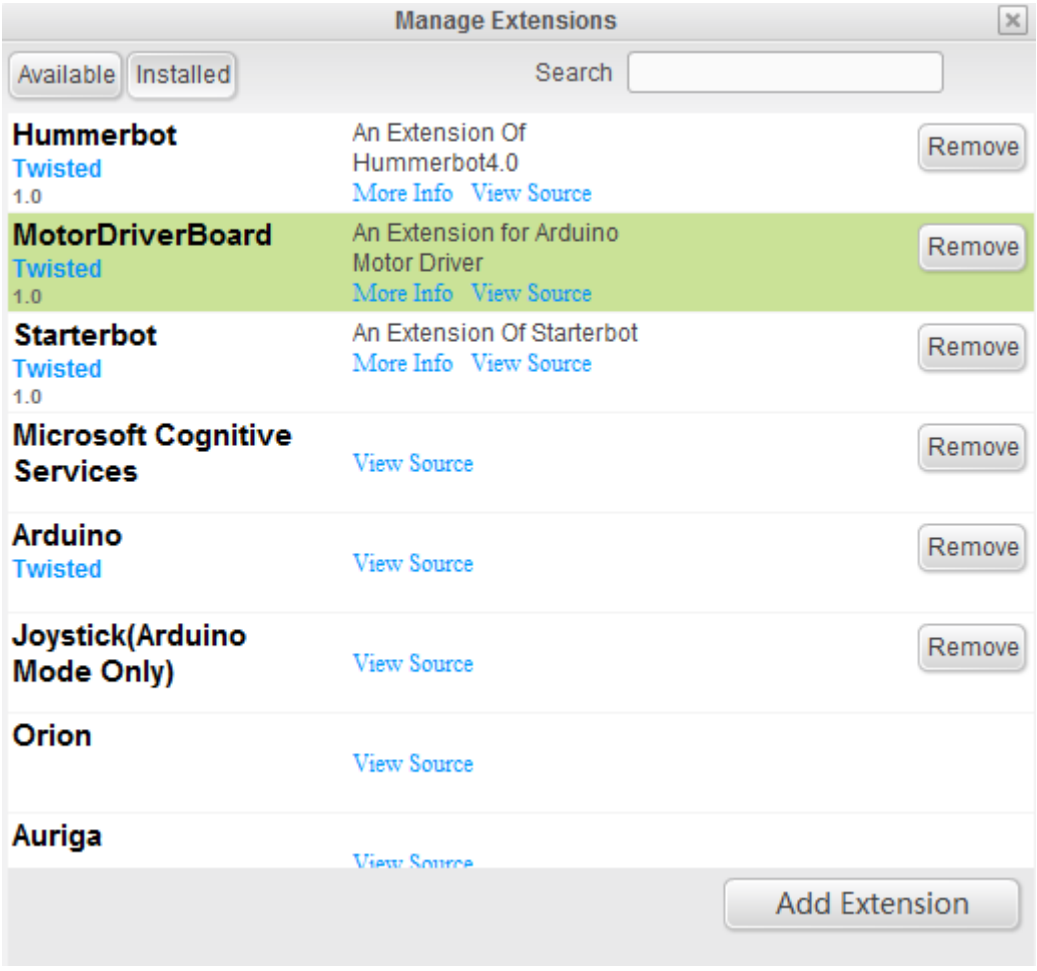


Figure 2-4

6) Click "Extension", select "Motor Driver Board", and then click "Script -> Robot Module" to display the Motor Driver Board graphics programming block in the building block area, as shown in Figure 2-5.

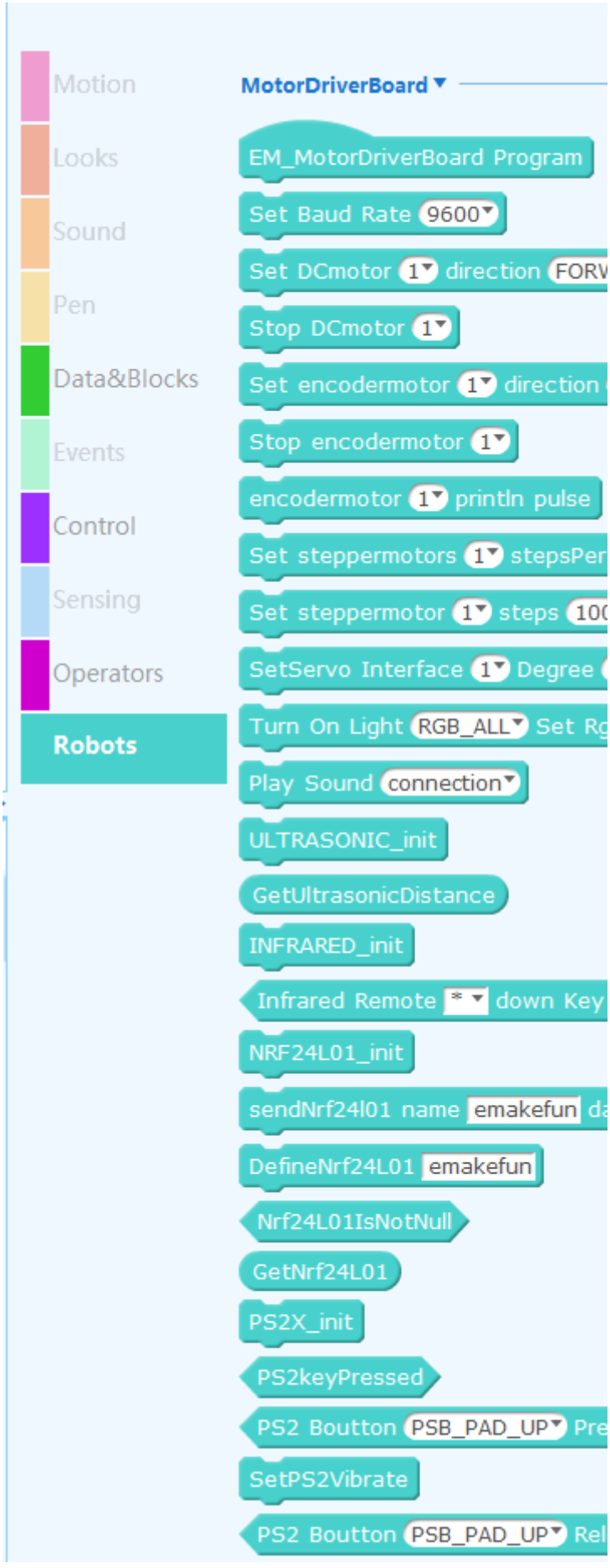

Motion
Looks
Sound
Pen
Data&Blocks
Events
Control
Sensing
Operators
Robots
MotorDriverBoard
EM_MotorDriverBoard Program
Set Baud Rate 9600
Set DCmotor 1 direction FORV
Stop DCmotor 1
Set encodermotor 1 direction
Stop encodermotor 1
encodermotor 1 println pulse
Set steppermotors 1 stepsPer
Set steppermotor 1 steps 100
SetServo Interface 1 Degree
Turn On Light RGB_ALL Set Rg
Play Sound connection
ULTRASONIC_init
GetUltrasonicDistance
INFRARED_init
Infrared Remote * down Key
NRF24L01_init
sendNrf24l01 name emakefun da
DefineNrf24L01 emakefun
Nrf24L01IsNotNull
GetNrf24L01
PS2X_init
PS2keyPressed
PS2 Boutton PSB_PAD_UP Pre
SetPS2Vibrate
PS2 Boutton PSB_PAD_UP Rel

Figure s2-5

2.1.2 First Experience of Programming

After adding the MotorDriverBoard library, let's experience MotorDriverBoard programming. Let's first write a program that lets MotorDriverBoard print Hello world.

1) First, drag the EM_MotorDriverBoard main program in the building block area to the script area with the mouse, and then drag the "Setting Serial Port Baud Rate 9600" to the bottom of the building block of the "EM_MotorDriverBoard main program", as shown in Figure 2-6.

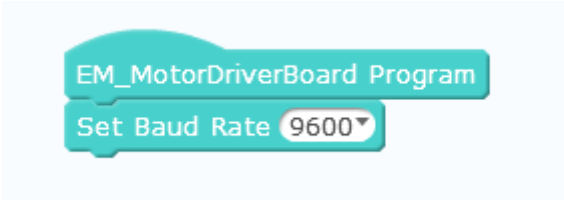


Figure 2-6

2) Click on the "Control" tab in the label area and drag the "Repeated Execution" block under the MotorDriverBoard main program in the script area, as shown in Figure 2-7.

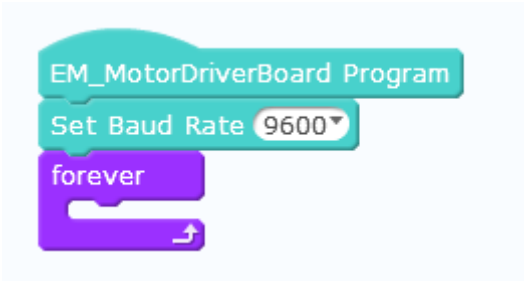


Figure 2-7

3) Drag the "serial port print string" building block into the repeated execution building block in the Motor DriverBoard building block and enter "Hello world" as shown in Figure 2-8.

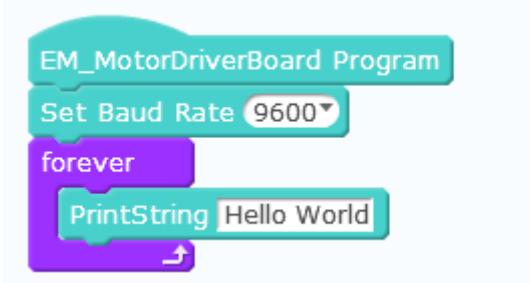


Figure 2-8

This is the MotorDriverBoard program for printing Hello World. After the program is written, we need to transfer the program to the brain of the Motor Driver Board Robot (control motherboard) so that the robot can do the desired action according to the program we write. How to transfer the program to the brain of the Motor Driver Board Robot (control motherboard)? Only when MagicBlock is connected with the robot motherboard, can we transfer the written

program from the computer to the brain of the MotorDriver Board robot (control motherboard). Here is the connection method between MagicBlock and the robot motherboard.

2.2 Connection steps for MagicBlock and MotorDriverBoard

1) Using a USB data line, one end is inserted into the computer, and the other end is inserted into the robot master board, which connects the robot master board with the computer.
2) Install the Motor Driver Board driver, click "Connect to Install Motor Driver Board driver" as shown in Figure 2-9, and click "Install" as shown in Figure 2-10.

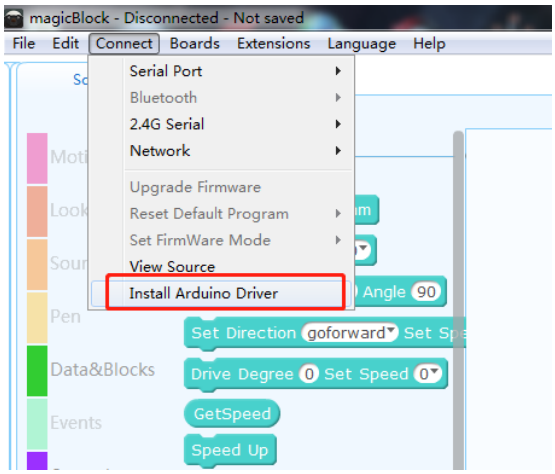


Figure 2-9

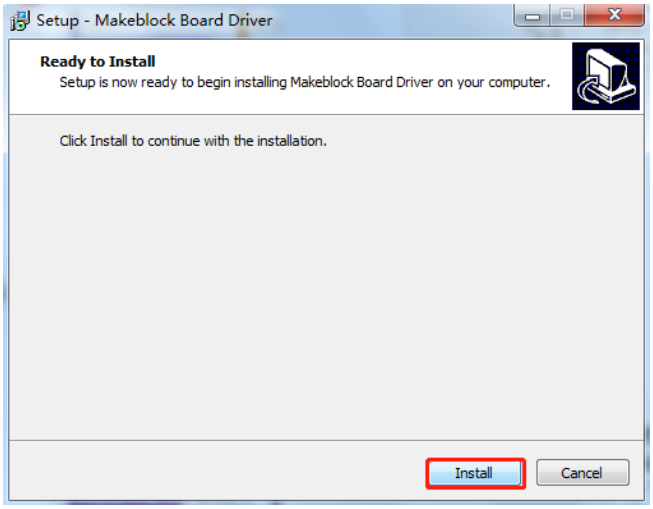


Figure 2-10

3) Click on "Connect to COM7 (different computers have different number of COM port)", as shown in Figure 2-11. After the correct connection, the software will have the "serial port has been connected" prompt at the top, at which time MagicBlock and MotorDriver Board successfully connected, as shown in Figure 2-12.

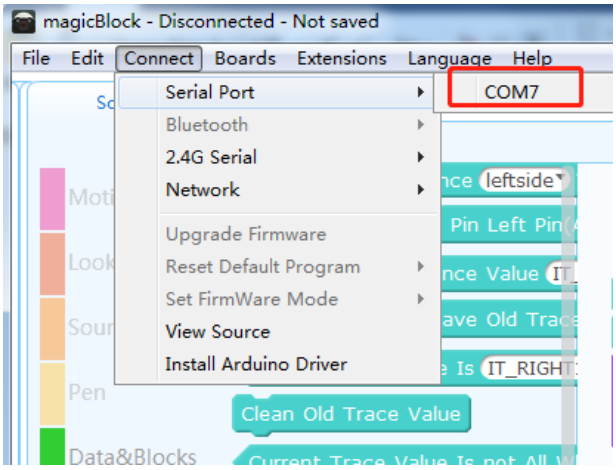


Figure 2-11

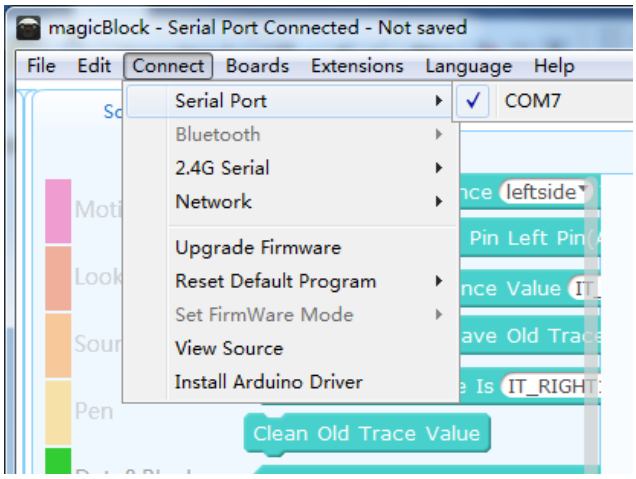


Figure 2-12

2.3 Upload Program to MotorDriverBoard

When the program is finished and Magic Block and Motor Driver Board are connected correctly, we can transmit the program to the brain of Motor Driver Board. The specific steps are as follows:

1) Select the type of control board for program transmission, select "Arduino Uno" as shown in Figure 2-13, and select "Edit Arduino Mode" as shown in Figure 2-15.

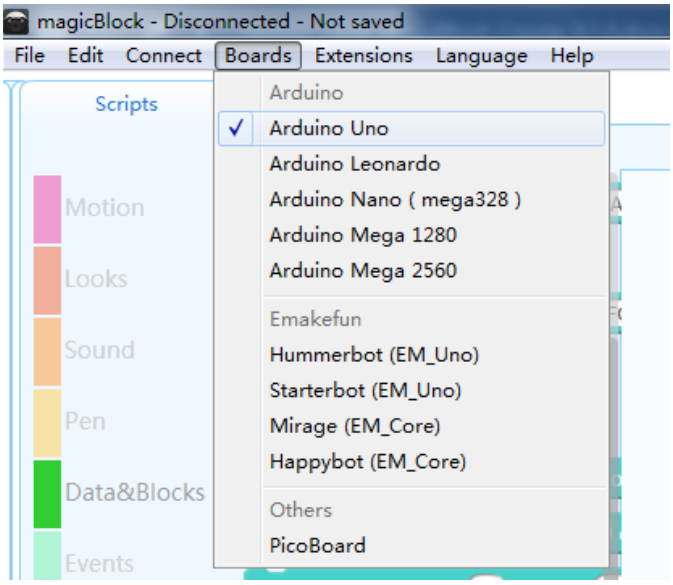


Figure 2-13

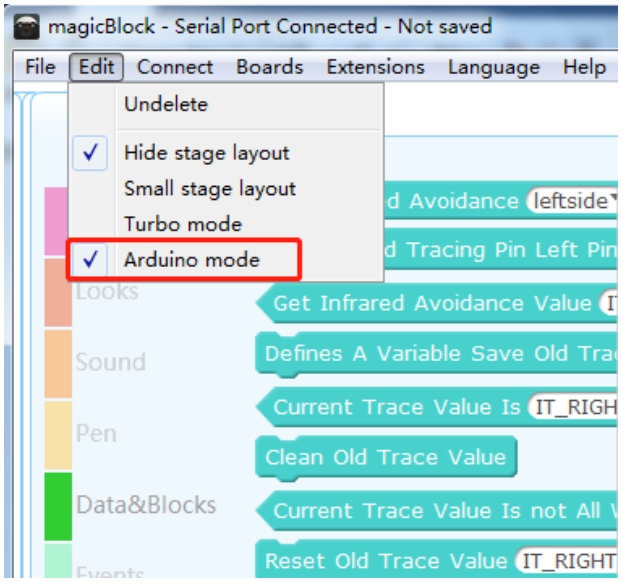


Figure 2-14

2) Click the mouse on any building block in the script area. On the right side of the building block area, the corresponding building block code will appear. Click "Upload to MotorDriverBoard", start to generate the off-line building block code and upload it to the MotorDriverBoard Robot. On the screen, the prompt window "Upload in", as shown in Figure 2-16, will appear after the upload is completed. "Upload completed", as shown in Figure 2-17;

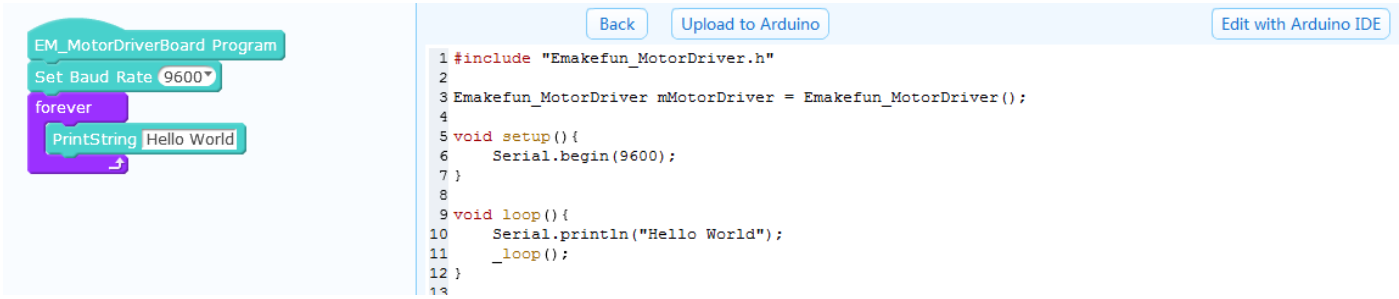


Figure 2-15

Figure 2-16 Figure 2-17

After completing the above steps, the Motor Driver Board robot's brain (master board) already has the program we wrote, so how can we see the Hello Word printed by Motor Driver Board? At this point, we will use a serial monitor to see, click the "Edit with MotorDriver Board IDE" in the upper right corner, as shown in Figure 2-18.

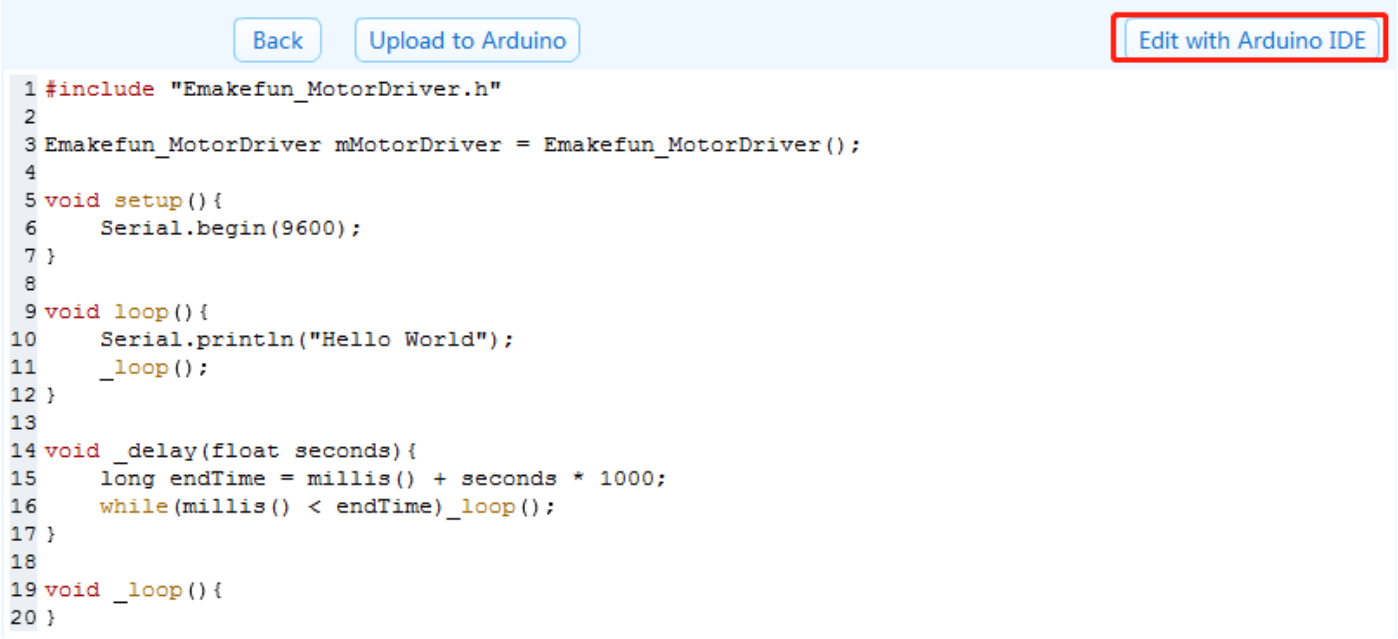


Figure 2-18

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

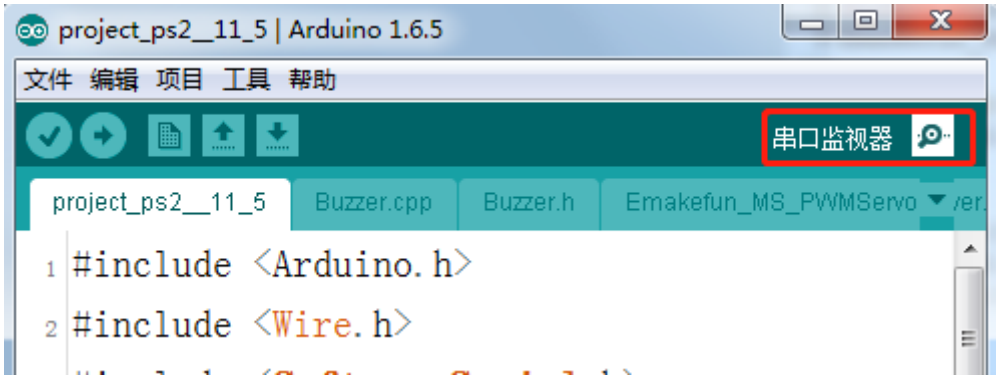


Figure 2-19

Turn on the serial monitor and we will see Hello Word printing continuously on the serial monitor, as shown in Figure 2-10.

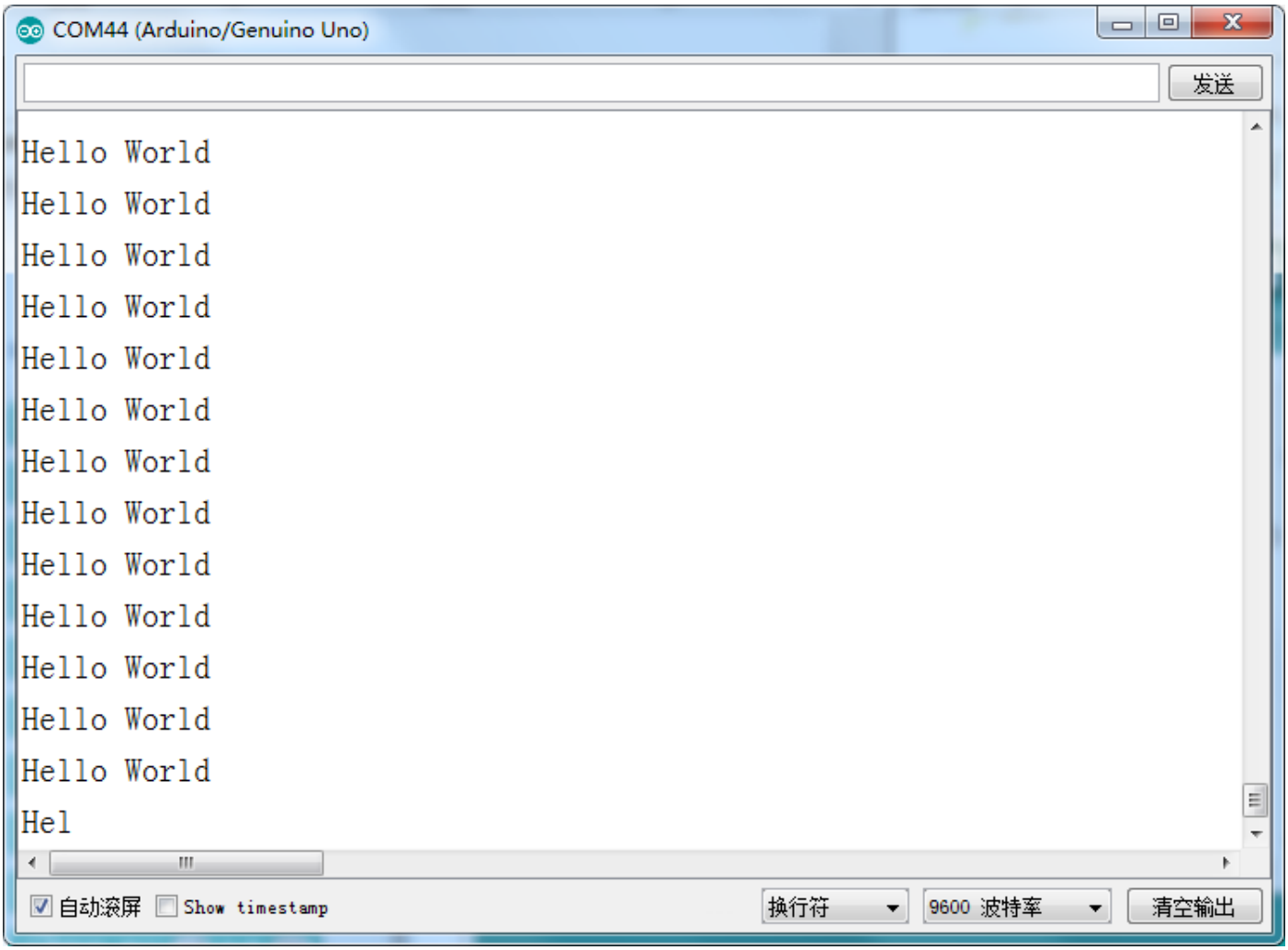


Figure 2-20

Chapter Three Robots Move

3.1 DC motor

3.1.1 Principle of DC Motor

The reason why cars are active is that they have engines to power them. Robots also have DC motor modules that allow them to move, so what is a motor? In our science textbook, there is an introduction of electromagnetic induction. The motor is rotated by electromagnetic induction. It has an iron core winding copper wire inside and a rotor outside. When the iron core is electrified, there is electromagnetic induction to make the rotor move. This is the motor.

Motor Driver Board has two DC motor interfaces. We can use DC motor to control the building blocks to make the DC motor rotate, thus driving the Motor Driver Board robot to move.

3.1.2 DC motor test

In the robot module, we can find two control blocks of the DC motor module, one is to set the motor pin blocks, as shown in Figure 3-1.

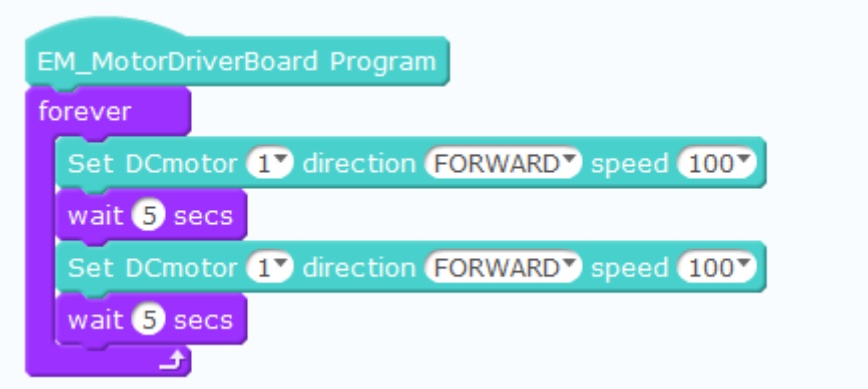
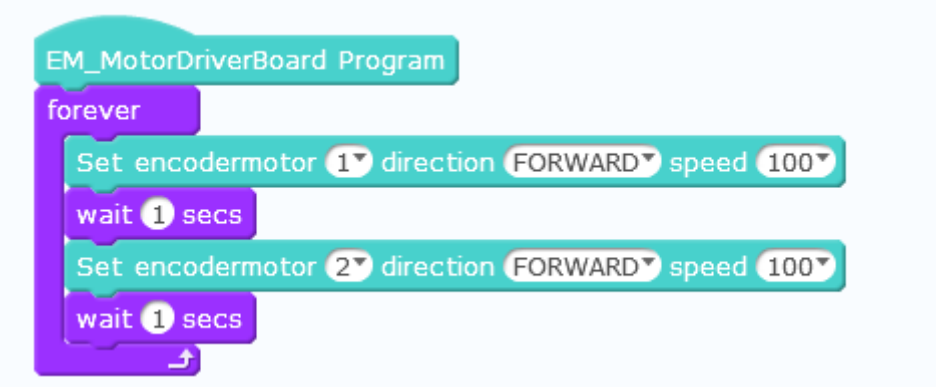


Figure 3-1

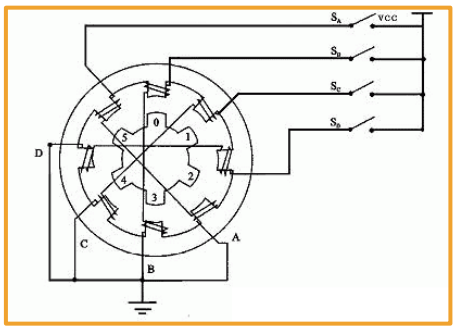
3.2 Coding motor

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

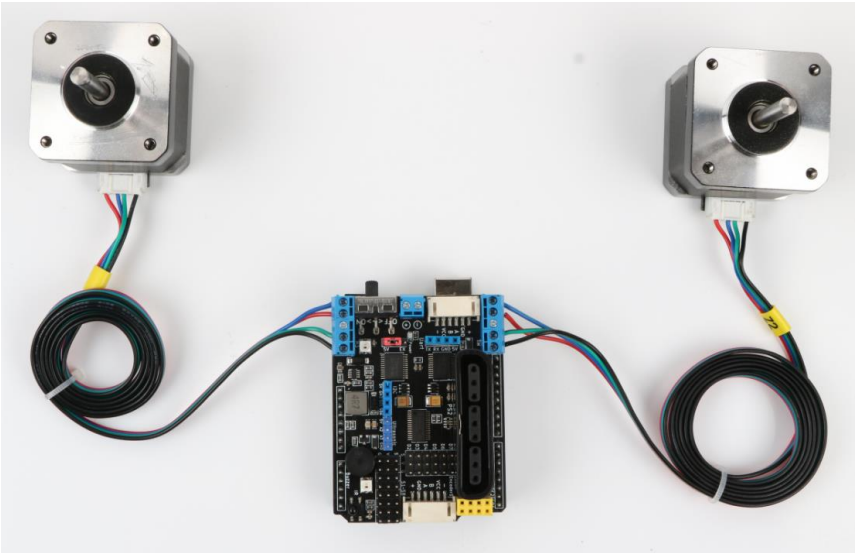


3.3 Stepper motor

Stepping motor is an open-loop control motor that converts electric pulse signal into angular displacement or linear displacement. It is the main executive component in modern digital program control system and is widely used. In the case of non-overload, the speed and stop position of the motor only depend on the frequency and number of pulses of the pulse signal.



There are many pairs of magnetic poles in the stepper motor. If the electrified state remains unchanged, the stepper motor will remain in a fixed state. Only by constantly changing the electrified state of each pair of magnetic poles can the stepper motor rotate continuously. Therefore, stepper motors can not work directly on DC or AC power supply, so special driving power supply (stepper motor driver) must be used. The controller (pulse signal generator) can control the angular displacement by controlling the number of pulses, so as to achieve the purpose of accurate positioning. At the same time, the speed and acceleration of motor rotation can be controlled by controlling the pulse frequency, so as to achieve the purpose of speed regulation.



3.4 Ultrasound module

3.41 Principle of Ultrasound

Ultrasound sensor is a device that detects distance by transmitting ultrasound. Ultrasound is an inaudible sound wave, which has the characteristics of returning when it touches an object. Ultrasound sensors have two "eyes", one of which emits ultrasound, and the other "eyes" receive the ultrasound emitted from obstacles. When one eye emits ultrasound, it begins to time, and when the other eye receives the returned ultrasound, it stops the time. Mathematically, we have learned to go through the road. Range = speed * time, then the distance measured by ultrasound = the speed * of ultrasound (timing time 2); thus distance can be calculated.

Figure 3-3

3.4.2 Usage of Ultrasound Module

We find the control building blocks of the ultrasonic module in the control module of the robot. As shown in Figure 3-4, the distance between the robot and the obstacle ahead can be detected by using the building blocks. Note: In order to use the ultrasonic module, we first need to choose the mode to control the building blocks and set the mode to the ultrasonic obstacle avoidance mode. When the obstacles completely block the ultrasonic module or face the ultrasonic module far away, the ultrasonic wave emitted by the ultrasonic module can not be received, so the robot and obstacle can not be detected. The detection distance of the ultrasonic module installed on the robot is 5CM ~ 400CM.

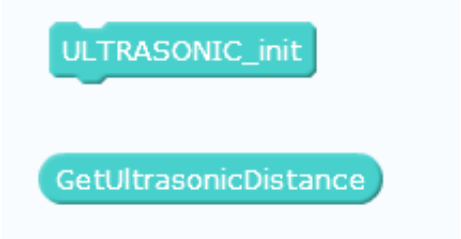


Figure 3-4

3.4.3 Testing of the Ultrasound Obstacle Avoidance Module

We can first write a program to test the ultrasonic obstacle avoidance module, using serial port printing, when we hand close to the ultrasonic obstacle avoidance module and far away from the ultrasonic obstacle avoidance module, we observe the distance measured by the serial port printing, we can more intuitively see the process of measuring the distance of the ultrasonic obstacle avoidance module. Let's write a test program first.

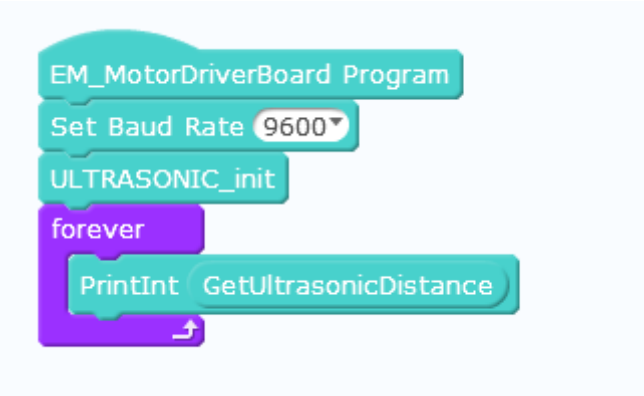


Figure 3-5

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

3.4.4 Operating Principle of Steering Engine

The steering gear is mainly composed of the following parts: steering wheel, deceleration gear set, position feedback potentiometer, DC motor, control circuit and so on, as shown in Figure 3-6. The Bumblebee and Optimus Prime joints we see in the movies need to be controlled by the steering gear. Especially when the robot is walking, it makes the mechanical sound of clicking and clicking. That is the steering gear rotates on the robot. Figure 3-6 is the most commonly used SG90 steering gear physical diagram at this stage.

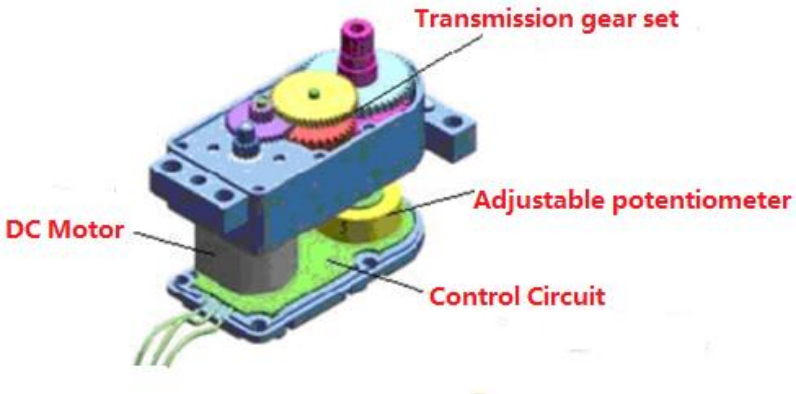


Figure 3-6

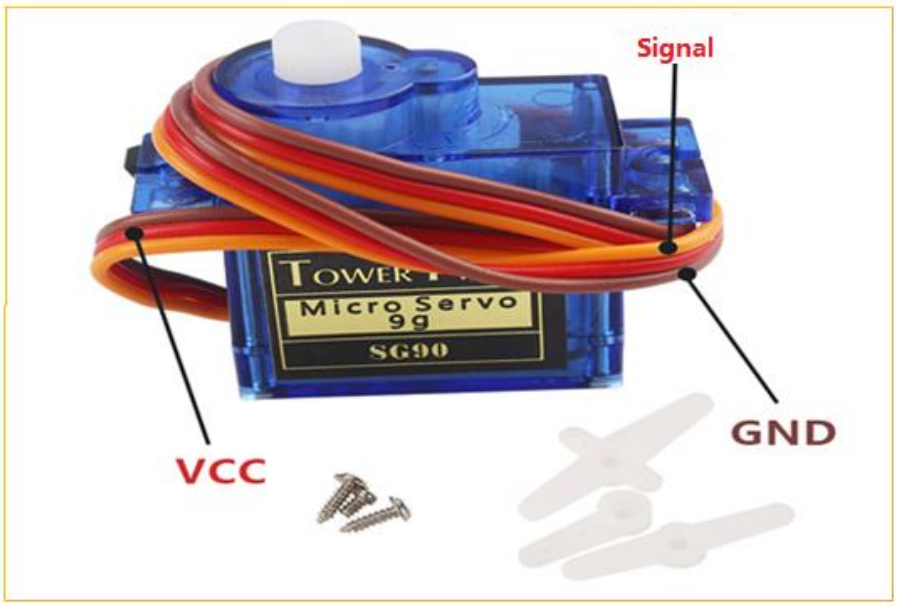


Figure 3-7

When the control circuit board receives the control signal of the self-confidence signal line, it controls the motor to rotate, and the motor drives a series of gears, then reduces the speed and drives them to the output steering wheel. Its work flow is: control signal control circuit board motor rotation gear set deceleration steering wheel rotation position feedback potentiometer control circuit board feedback.

3.4.5 Servor Test

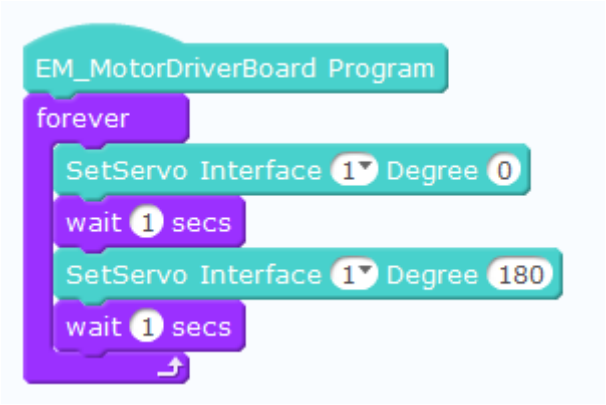


Figure 3-8

Above is the test procedure of the steering gear. Connect the interface 1 of the steering gear, upload the program to the Motor Driver Board, connect the power supply, and you can see that the steering gear repeats left and right turns every two seconds. (180 degrees is right, 0 degrees is left.)

3.4 RGB module

Motor Driver Board is equipped with two RGB lights. These two modules are called RGB module. R is the abbreviation of English Red, G is the abbreviation of English Green, B is the abbreviation of English Blue, red, green and blue is what we call the three primary colors of color and light. RGB module can make it emit different colors of light through building block control. We can find the control building blocks of RGB module in the robot module, as shown in Figure 3-9.

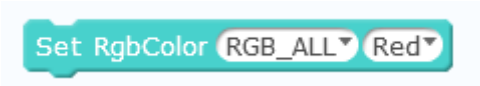


Figure 3-9

They can use the RGB module to write a program of continuously changing and flickering lights as shown in Figure 3-16.

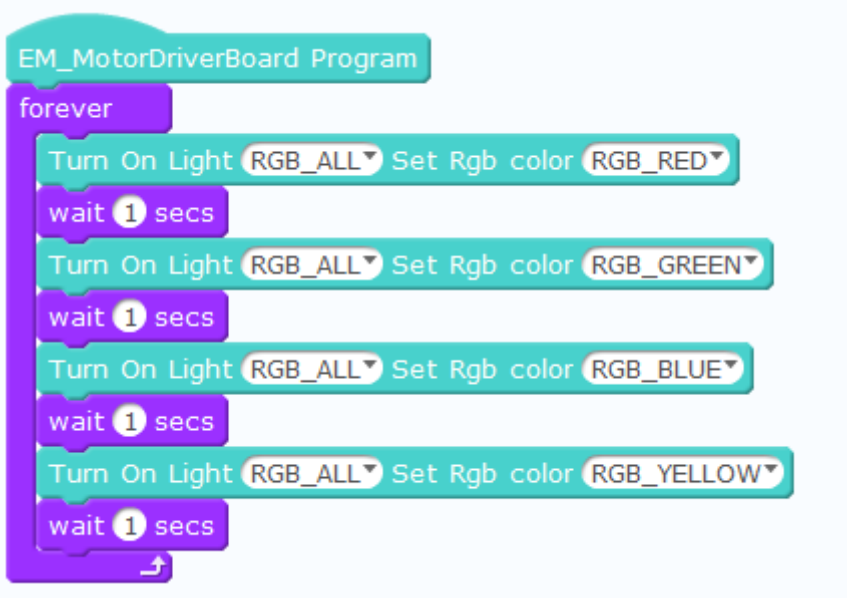


Figure 3-10

3.5 Buzzer module

3.5.1 Introduction of life

By associating the common way of playing musical instruments, we can understand that sound is the sound wave produced by the vibration of the right object and the surrounding air. Different frequencies of vibration make different sounds.

3.5.2 Cognitive Module and Its Use

We can find the buzzer module control building blocks in the robot module, as shown in Figure 3-12.

Figure 3-12

We can use the buzzer module to write a simple voice program as shown in Figure 3-13. Of course, we can also choose all kinds of desired sounds in the drop-down box.

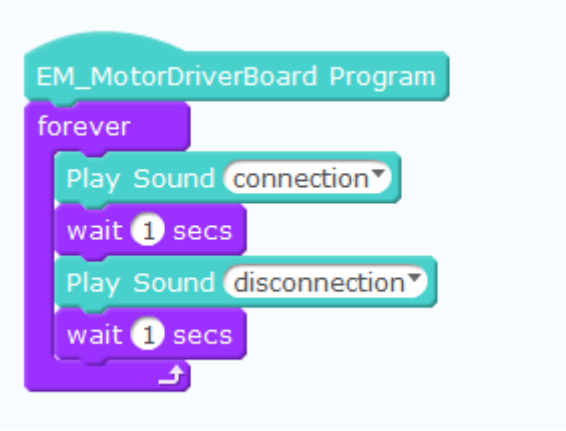


Figure 3-13

Chapter IV Infrared Telecontrol Robot

4.1 Principle of infrared remote control

The remote control system is generally composed of a remote controller (transmitter) and a receiver. When you press any key on the remote controller, the remote controller will issue an instruction. When the receiver receives the instruction from the remote controller, it will transmit the instruction to the robot's brain. The robot will think about what kind of action to do according to the remote control instruction. After controlling their limbs (four wheels) to do the corresponding action, the remote controller and receiver are shown in Figure 4-1.

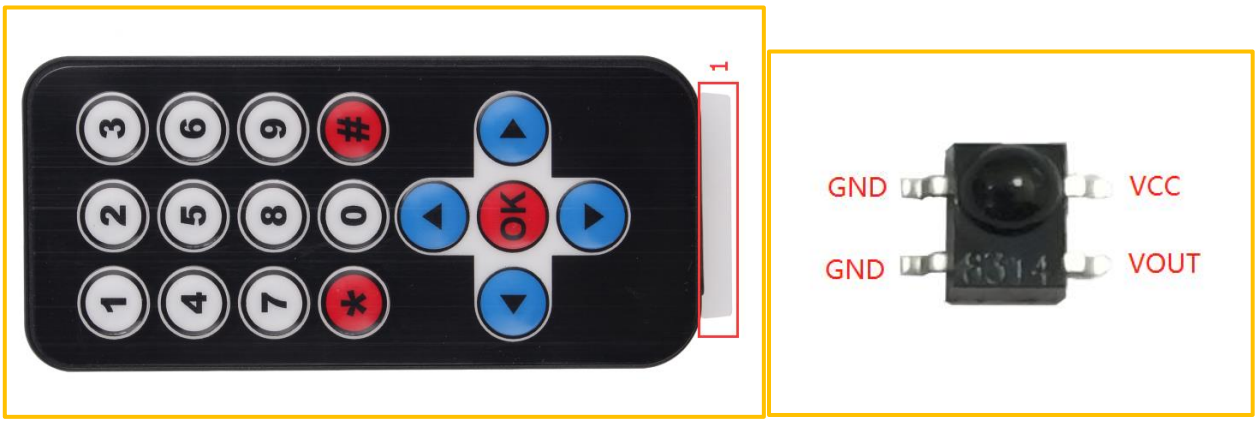


Figure 4-1

4.2 The Use of Remote Controller

To control the robot through the remote control, we first set the operation mode to infrared remote control mode, and set the infrared remote control receiving pin according to the actual connection port, then define the effect of each key press of the remote control, so that when we press the key of the remote control, the robot will do the settings in our program.

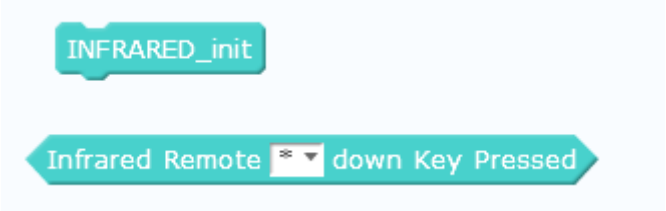


Figure 4-2

4.3 Infrared remote control test

We can write a program to test the infrared remote control, using the RGB light color, when the remote control button is pressed, we can observe the color of the RGB light, we can see which button of the infrared remote control is pressed. Now let's write a test program first.

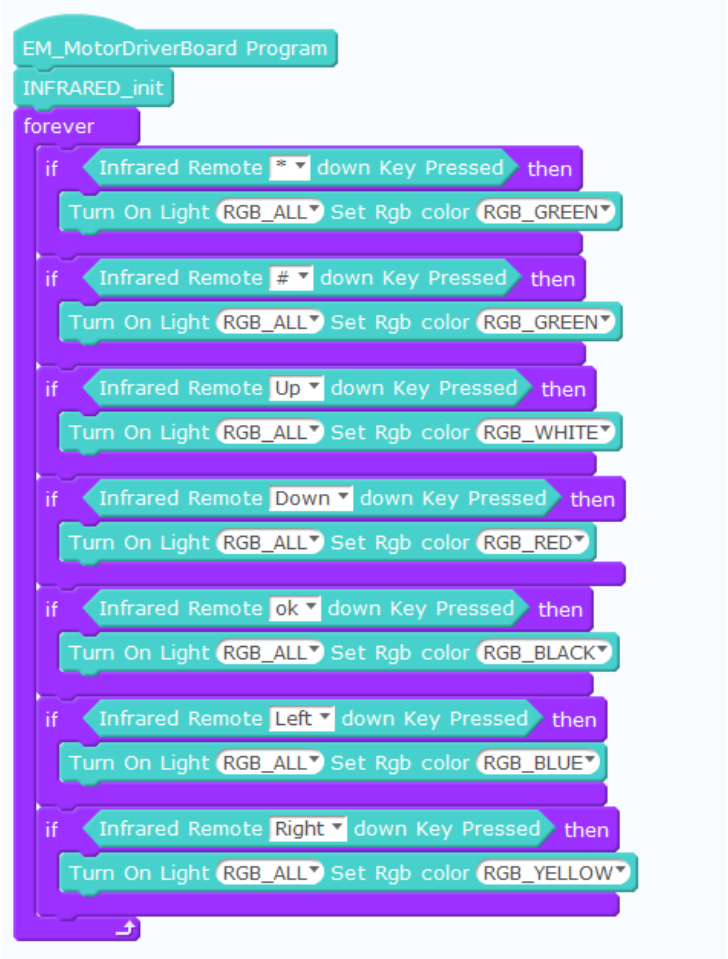


Figure 4-3

We have written the program as shown in Figure 4-3, and then after the program is uploaded successfully, open the serial port, then hold the remote control and press the key to the robot, and the corresponding RGB light color will be displayed.

Chapter 5 PS2 Module

5.1 Principle of PS2 remote control handle

The PS2 handle consists of a handle (Fig. 5-1) and a receiver (Fig. 5-2). The handle needs two sections of No. 7 1.5V power supply. Put the handle switch on ON. Without searching for the receiver, the light on the handle will flash continuously. For a certain period of time, the receiver has not been searched, and the handle will enter standby mode. The light on the handle will be extinguished. At this time, press the "START" button to wake up the handle.

The working power supply of the receiver is 3 ~ 5V, so it can not be connected back or overvoltage, otherwise the receiver will burn out.

After normal power-on, the handle and the receiver are automatically paired and connected. In the unsuccessful state, the green light of the receiver flashes, and the light on the handle also flickers. After matching success, the green light on the receiver is always on, the light on the handle is always on, and the button "MODE" (the batch of the handle is different, the above logo may be "AnalyOG", but not. Will affect the use of), you can choose "red light mode" and "green light mode".

Figure 5-1

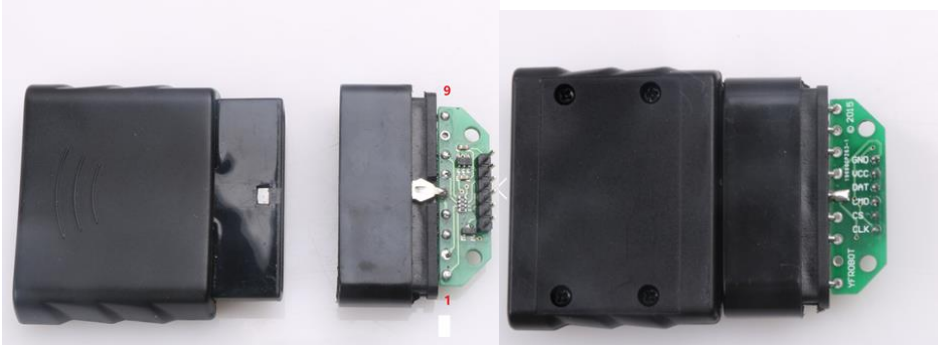
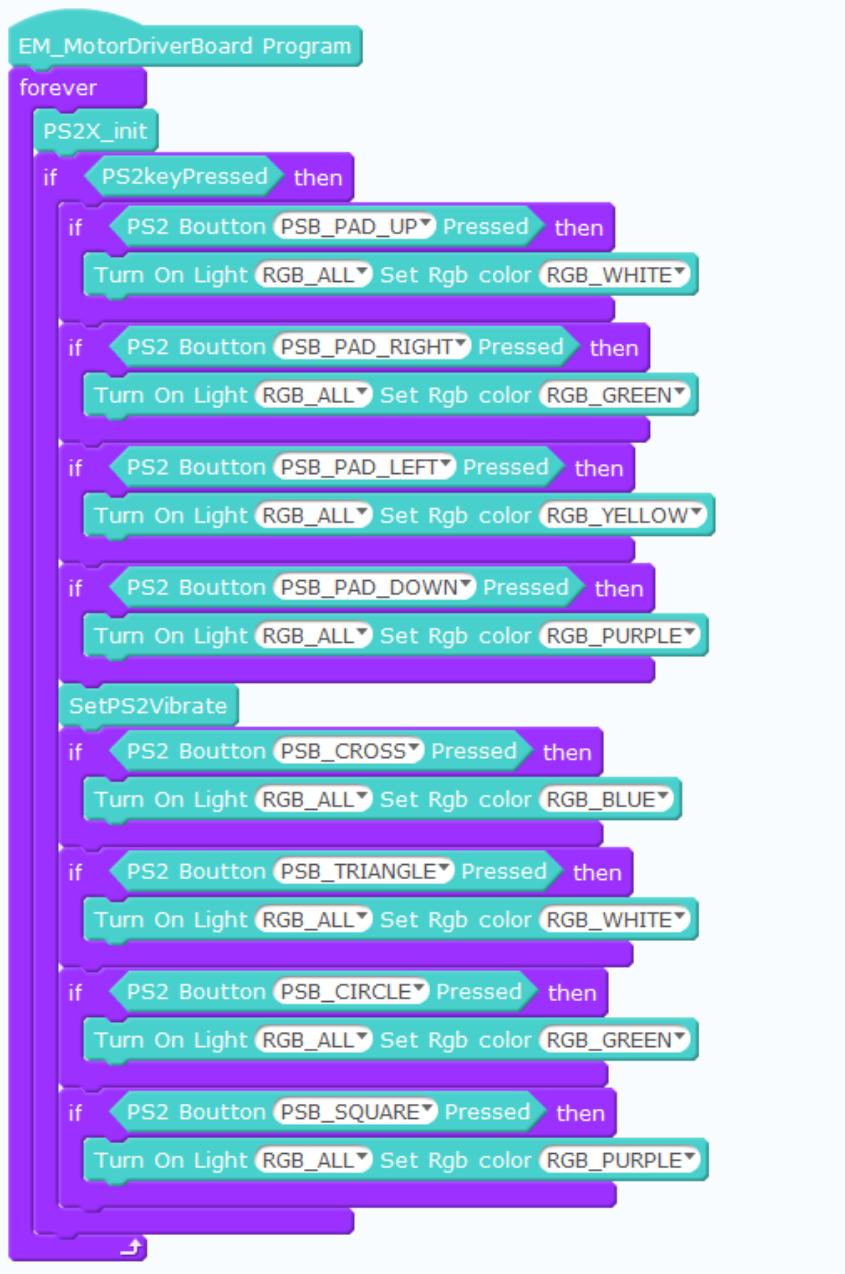


Figure 5-2

When the handle is connected to the receiver, we can use the handle to send button commands. When the receiver receives these button commands, the robot's brain (main control board) will let his limbs (four wheels, steering gear) do the corresponding actions according to the received commands.

5.2 PS2 test

We can write a program to test the PS2 remote control first. When the PS2 remote control button is pressed and the RGB light is on, we can see which button of the PS2 remote control is pressed. The following picture is a test program.



Chapter 6 Nrf24L01 Module

6.1 Introduction of Nrf24L01

Nrf24L01 is a wireless transmission module, which is used for wireless data transmission. Introduction of NRF24L01 + module

The nRF24L01 + module (shown in Figure 13-1) is a 2.4G wireless communication module developed by Nordic based on nRF24L01 chip. Using FSK modulation, Nordic's Enhanced Short Burst protocol is integrated internally. It can realize point-to-point or 1-to-6 wireless communication. The highest speed of wireless communication can reach 2M (bps). NRF24L01 has four working modes: transceiver mode, configuration mode, idle mode and shutdown mode.

Driving NRF24L01 + Module

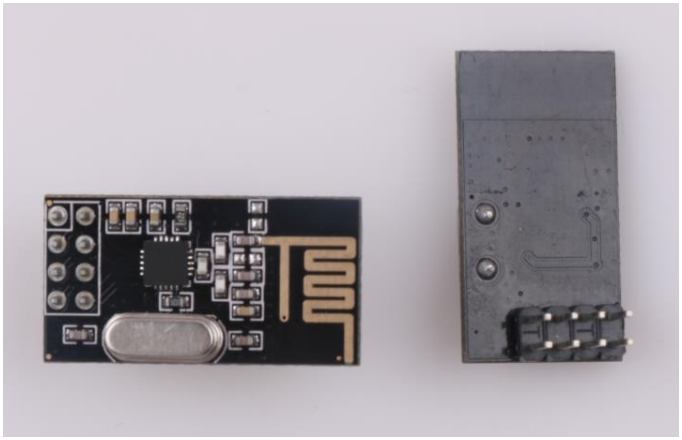


Figure 6-1

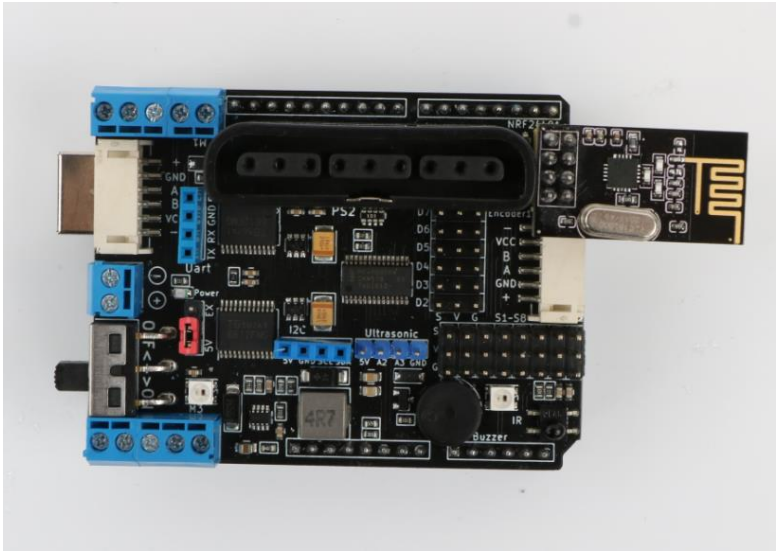
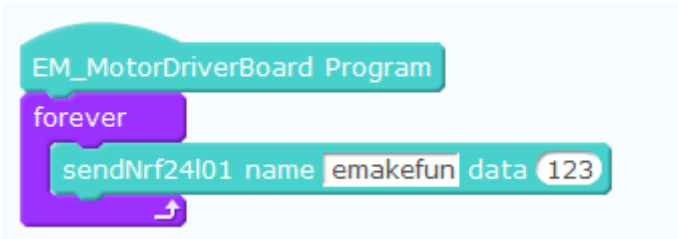


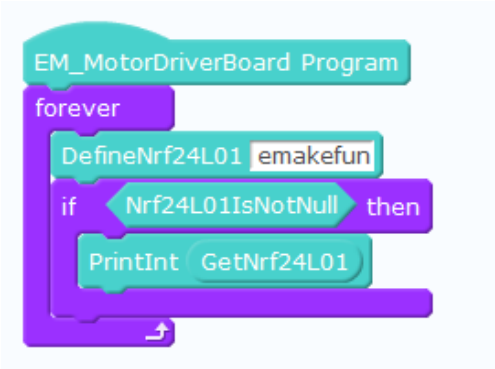
Figure 6-2

NRF24L01+ Test

When writing Nrf24L01 + test with MagicBlock, two MotorDriver Boards are needed and two Nrf24L01 + modules are inserted into corresponding excuses respectively. Then two programs are constructed with building blocks in MagicBlock: a data sending program and a data receiving program, as shown in the following figure.

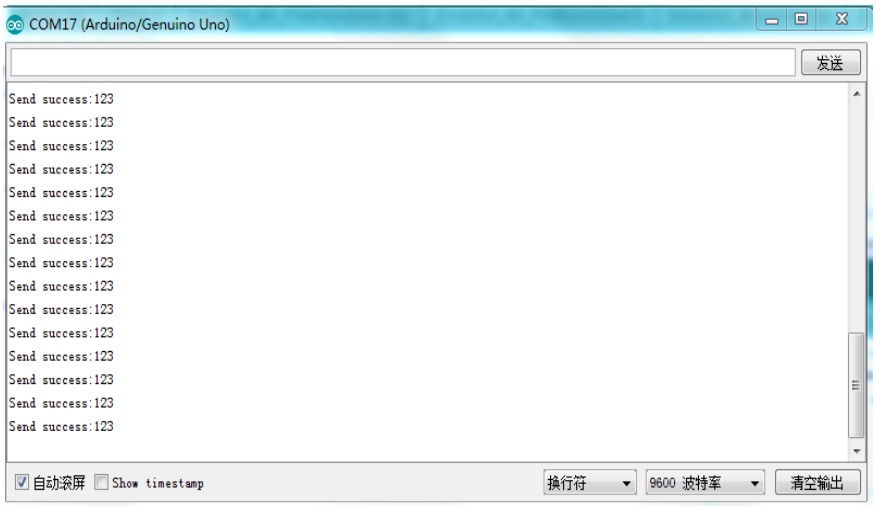


send data

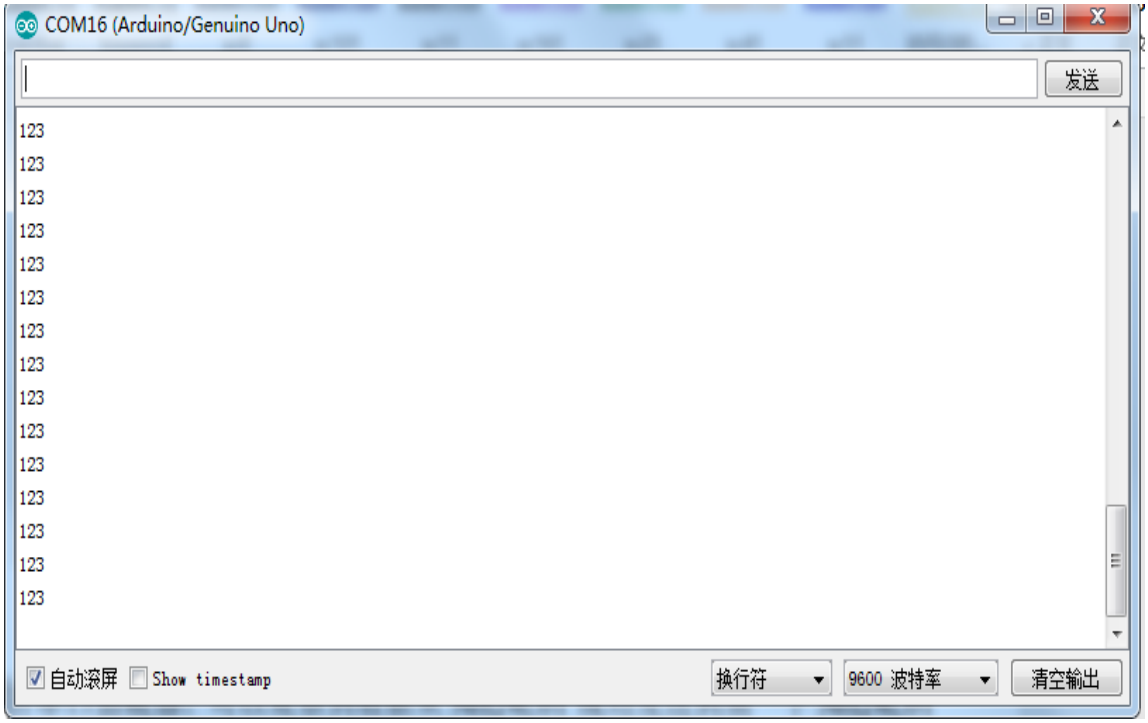


receive data

By compiling with Arduino IDE and opening the serial port monitor, we can observe that the data is successfully sent from Nrf24L01 to another Nrf24L01.



send data“123”



receive data “123”