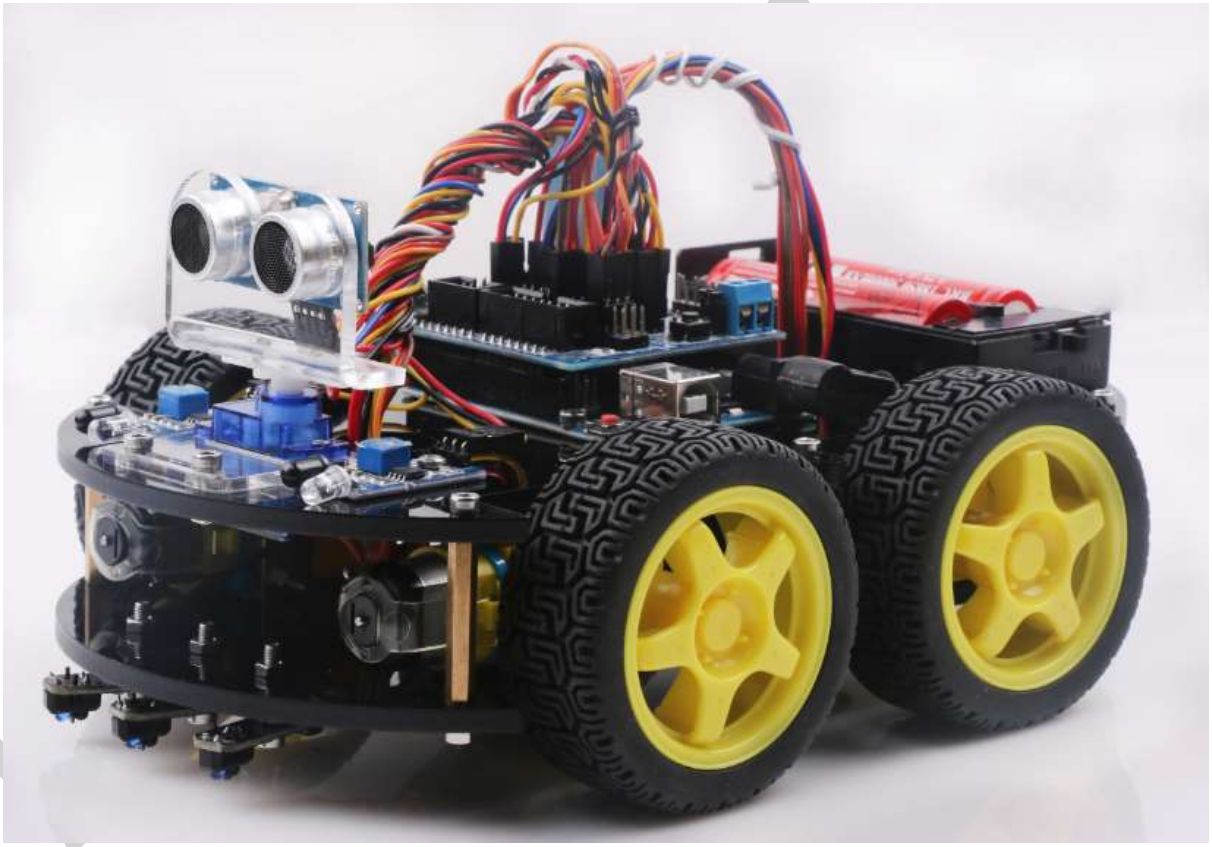
Hummer-bot-2.0 Scratch Programming Tutorial V.1.3



Revision

Date	Version No.	Description	Author
2018-11-9	**V.1.0**	**Create document**	**Abbott.Chen**
2018-11-9	**V.1.1**	**Delete some content, update the catalog**	**Ken.chen**
2018-11-30	**V.1.2**	**Proof translation**	**Ruby**
2018-12-1	**V.1.3**	**Optimization of pictures**	**Abbott.Chen**

Catalog

Chapter 1 Learning about Hummer-bot and its programming environment

1.1 Life introduction

Hello everyone, welcome to the world of Hummer-bot robot!

What is a robot?

Robot doesn't have to look like human that can called robot, A machine that can automatically perform work, accept both human command and pre-programmed programs then it can be called robot.

1.2 Overview

1.2.1 Hummer-bot Introduction

Through the study of this tutorial, we will experience the interactive fun of electromechanical and software. The Hummer-bot consists of three components: the Hummer-bot smart body, the Arduino and the mBlock graphical programming software.

mBlock is one of the world's most popular graphical programming learning software for STEAM education. It is based on MIT open source software Scratch2.0. It not only allows users to create interesting stories, games, animations, etc., but also supports Arduino hardware and a variety of types of sensors, actuators, displays, motors, communication modules, storage modules and other extended applications, just drag and drop when used Sensor control can be achieved by setting the pins. In this software, programming is no longer a boring and cumbersome code. The graphical logic and drag-and-drop interaction complete the core logic of programming. The whole programming is like playing a fun building block toy. You can program it yourself through mBlock's arduino mode. To control the behavior of the Hummer-bot, you can visually see the actual effects of various script designs.

1.2.2 Hummerbot function list

1) Ultrasonic obstacle avoidance function;
2) Ultrasonic + infrared obstacle avoidance function;
3) Automatic tracking function;
4) Infrared remote control function;
5) Bluetooth remote control function;
6) PS2 remote control function (optional);
7) Bluetooth remote mode switching function;

1.3 Hummer-bot Hardware introduction

1.3.1 Analogy introduction

The hardware part of the Hummer-bot robot we use is a body model built of acrylic plates, various sensors, motor wheel combinations, control boards, and Hummer-bot robot with powered batteries, as shown in Figure 1-1.

Figure 1-1

In this robot, each part of it is similar to the structure of our human body and plays a vital role. The classes of humans and humans are shown in Table 1-2:

Robot	**Humanity**	**Function**
control board	brain	accept instructions, process information, and issue control limb and organ commands
motor and wheel	arms and legs	responsible for limb movement
battery	heart	provide power
sensor	facial features	accepting sensory information
electronic circuit	nerve	connect each part of the limb so that each part is connected as a whole

Figure 1-2

1.3.2 Main control board introduction

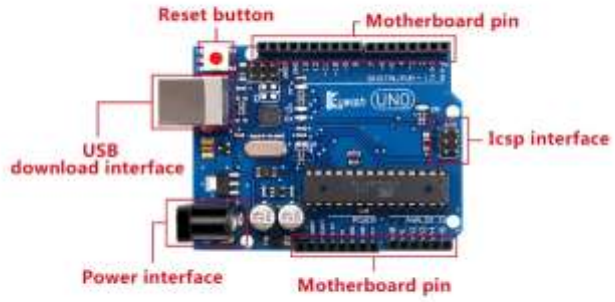


Figure 1-3

1.3.3 Expansion board introduction

Figure 1-4

1.4 Hummer-bot Software compilation environment introduction

1.4.1 Software Installation

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

as shown in Figure 1-6.

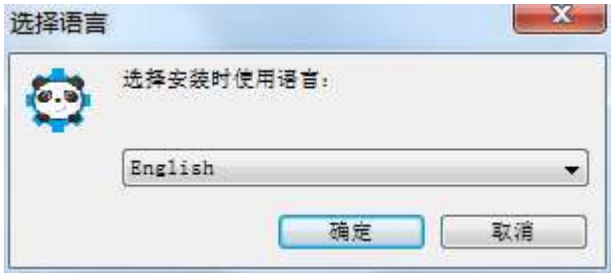


Figure 1-5 Figure 1-6

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.

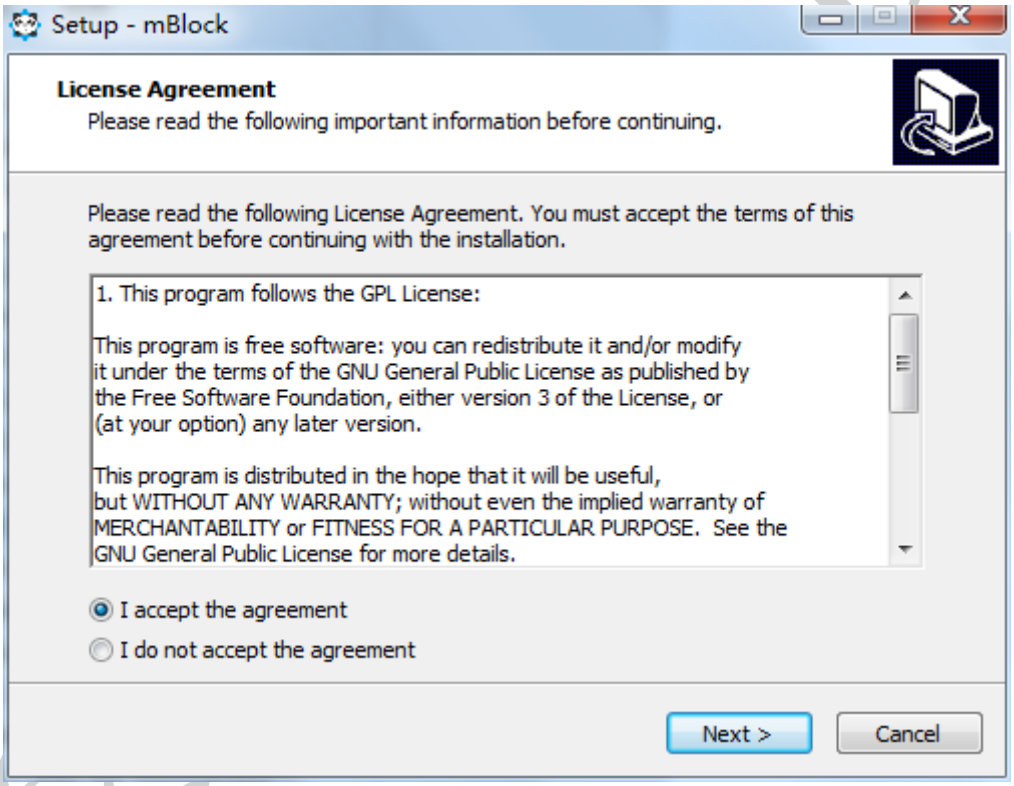


Figure 1-7

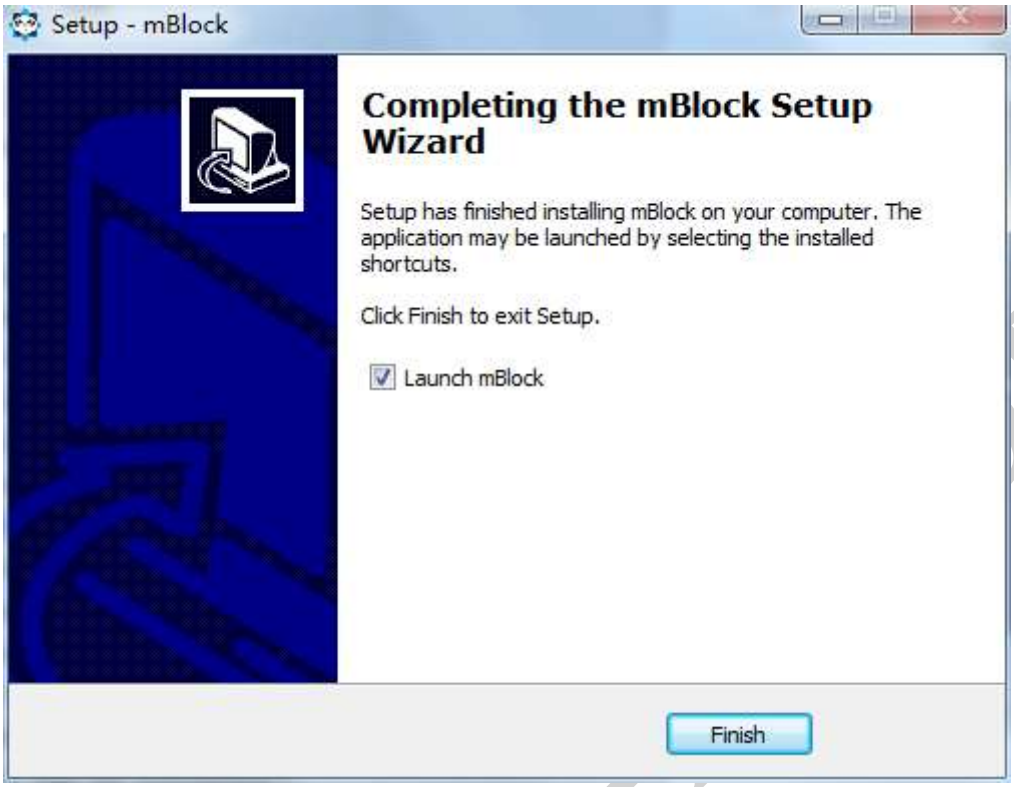


Figure 1-8

1.4.2 The compilation environment introduction

We know that in order to move the robot, in addition to the hardware cooperation, it is more important to program it. We have already understood the hardware part of the robot. Nextly we will understand the software part of it. The software part of Hummer-bot 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 1-13.

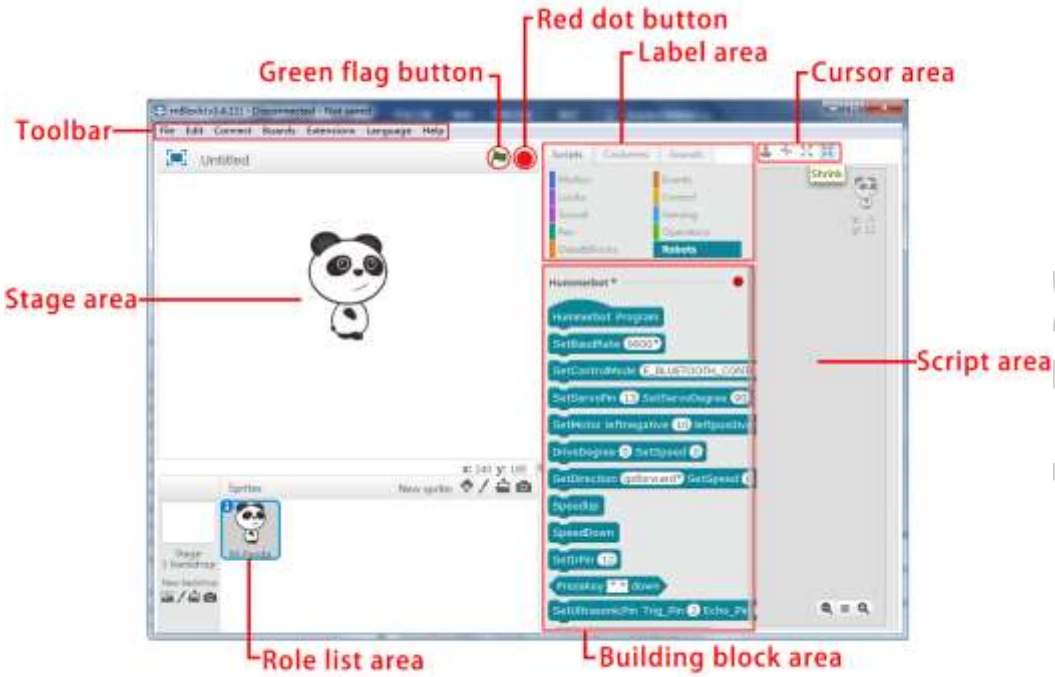


Figure 1-9

- **Toolbar:** The area where the project file, software interface mode, serial port connection and upload program, control panel selection, software usage 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 runs.
- **Green flag button：** Click the green flag button to start the program.
- **Red dot button:** Click the red button to stop the program.
- **Role list area:** The area where all the character prototypes are displayed. Here you can see the name of the character, the direction of rotation, the position and so on.
- **Label area:** Contains script tabs, styling tabs, and sound tabs for scripting, styling, and sounding of characters.
- **Building 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:** Contains copy, delete, reduce, zoom in buttons, which are used to manipulate the characters on the stage.
- **Script area:** The area where the program is written, the building 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 scripts (as shown in Figure 1-10), styling (as shown in Figure 1-11), and sound (as shown in Figure 1-12). The script is mainly used to perform actions by some control building blocks; Under the styling label, you can draw some graphics to add to the program; The sound label can record some sounds yourself or you can put them in the program;Hummer-bot mainly uses the control blocks under the script label.

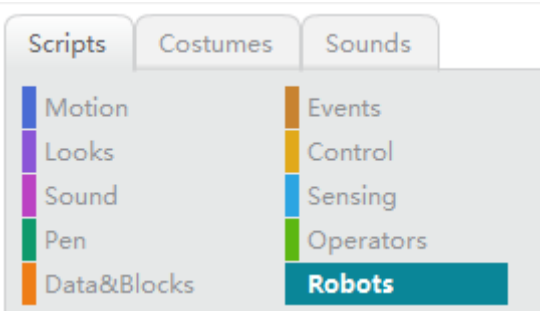


Figure 1-10

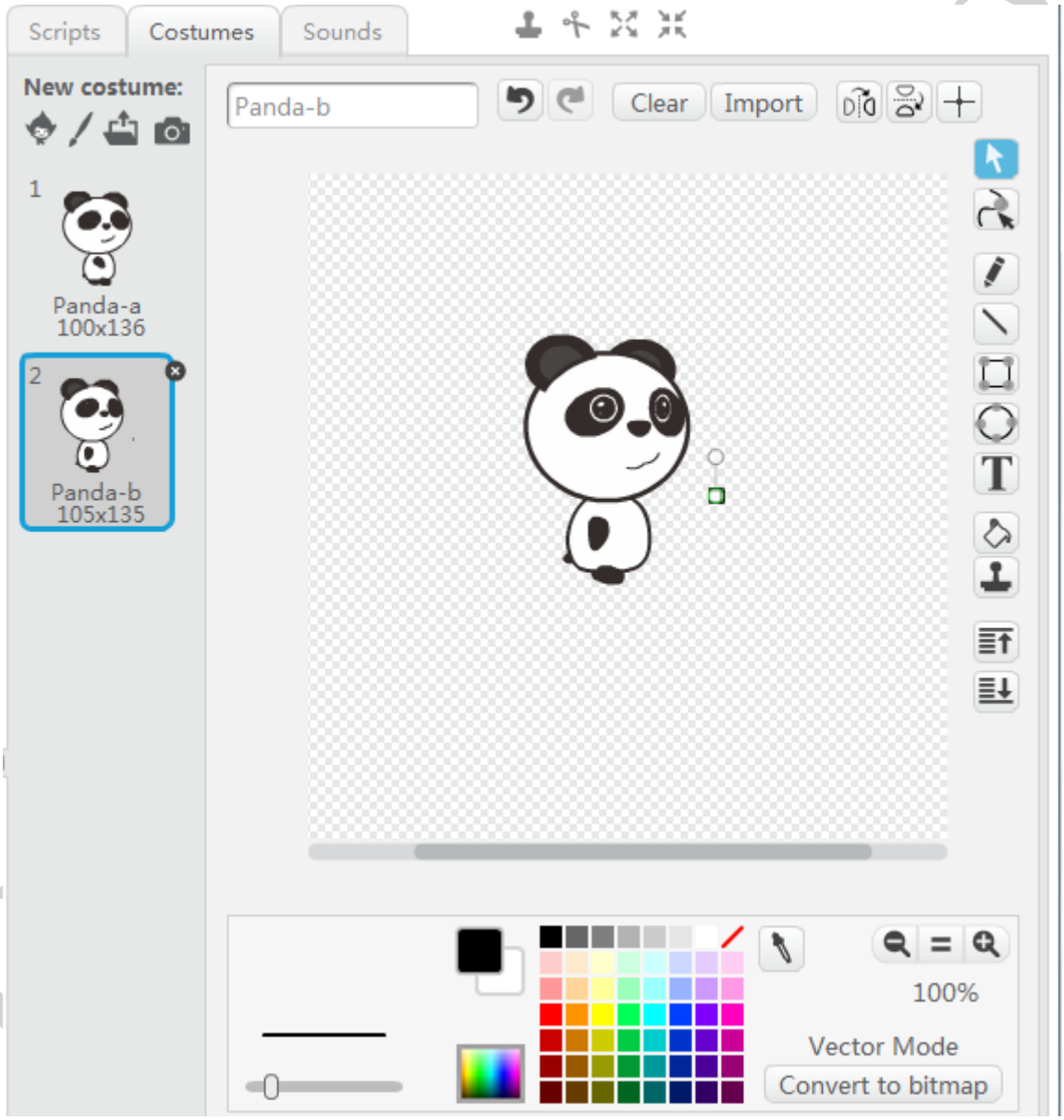


Figure 1-11

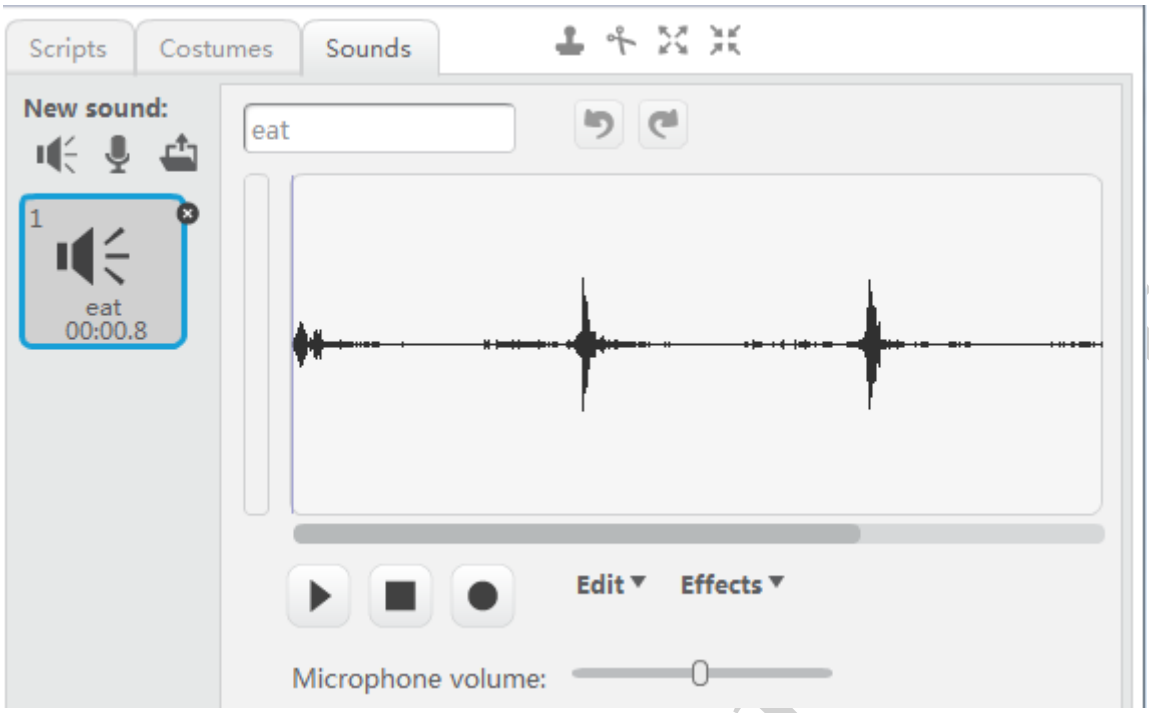


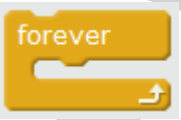
Figure 1-12

1.6 Hummerbot building block introduction

There are 10 types of building blocks under the script tag of Mblock. The 7 building block types of action, appearance, sound, brush, data and command, event, and detection are all common logic blocks that do not involve hardware. They are not used in Hummerbot. If you are interested, you can try your own practice. We won't go into details here. We mainly understand the three building block types of control, digital and logical operations and robot modules.

⑴Control building blocks are the building blocks that control the execution flow of the program (Figure 1-13), leading the program.

Figure 1-13

For example 

This building block means that the program inside the building block is repeatedly executed and such as this 

is to let the robot always print the" hello" on the serial port;

⑵The 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.

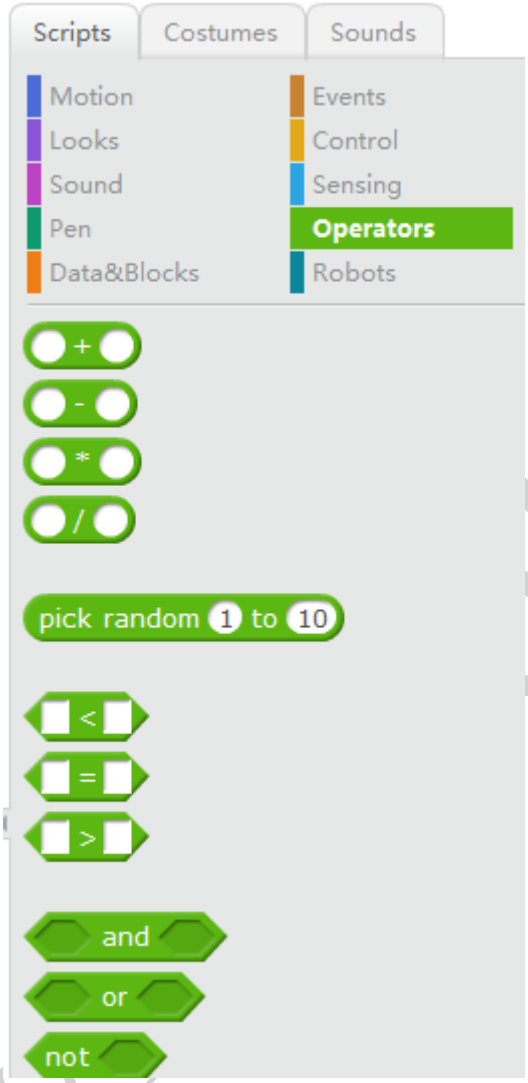
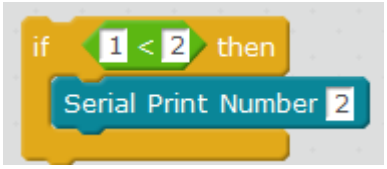
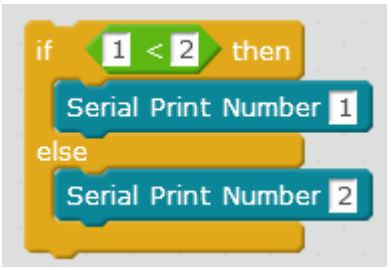


Figure 1-14

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.

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.

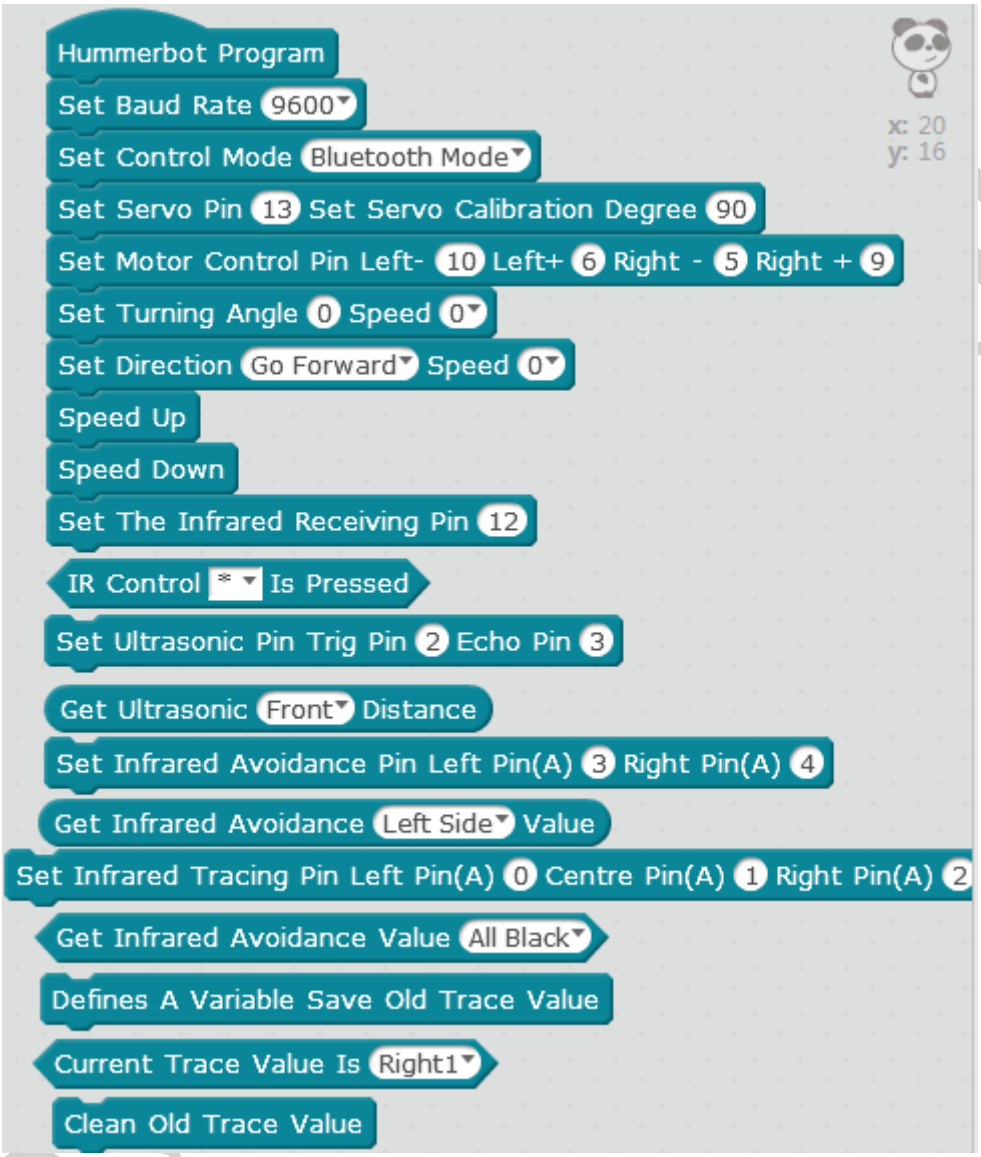


Figure1-15

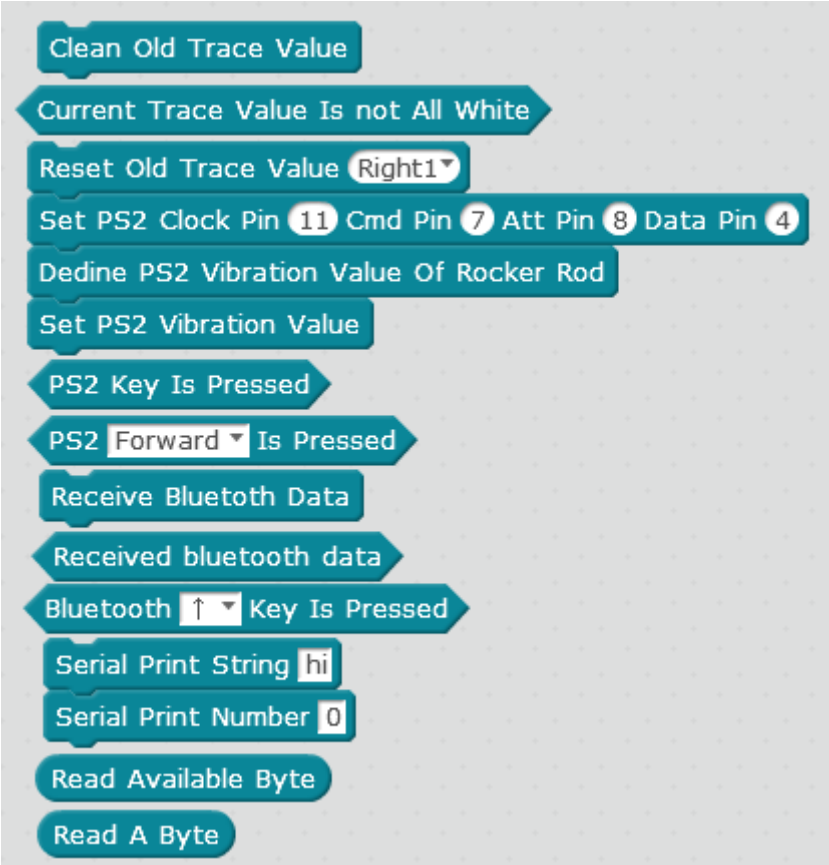


Figure 1-16

Chapter 2 First-time programming

2.1 Hello word

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

2.1.1 Add the Hummerbot library

Before programming, we need to add the Hummerbot library first, as described below:

1) Download the Hummer-bot library and save it on your computer. File name: Hummerbot.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.

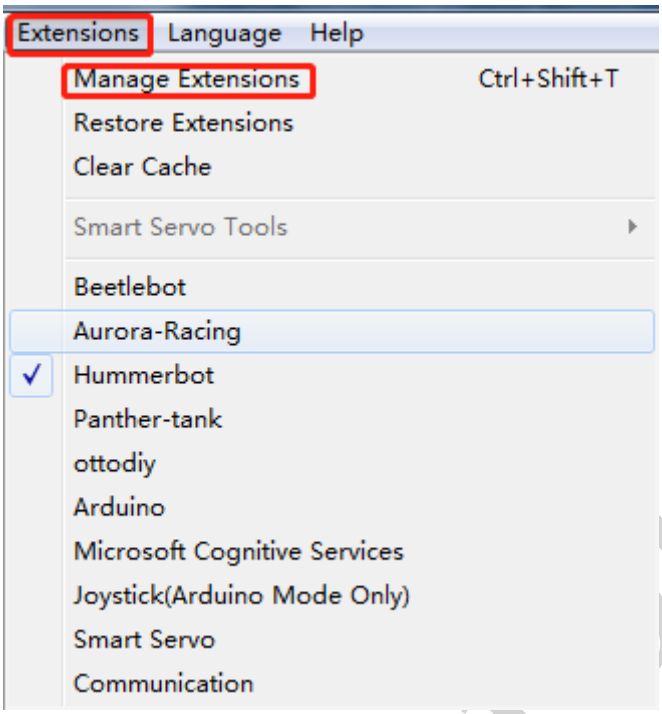


Figure 2-1

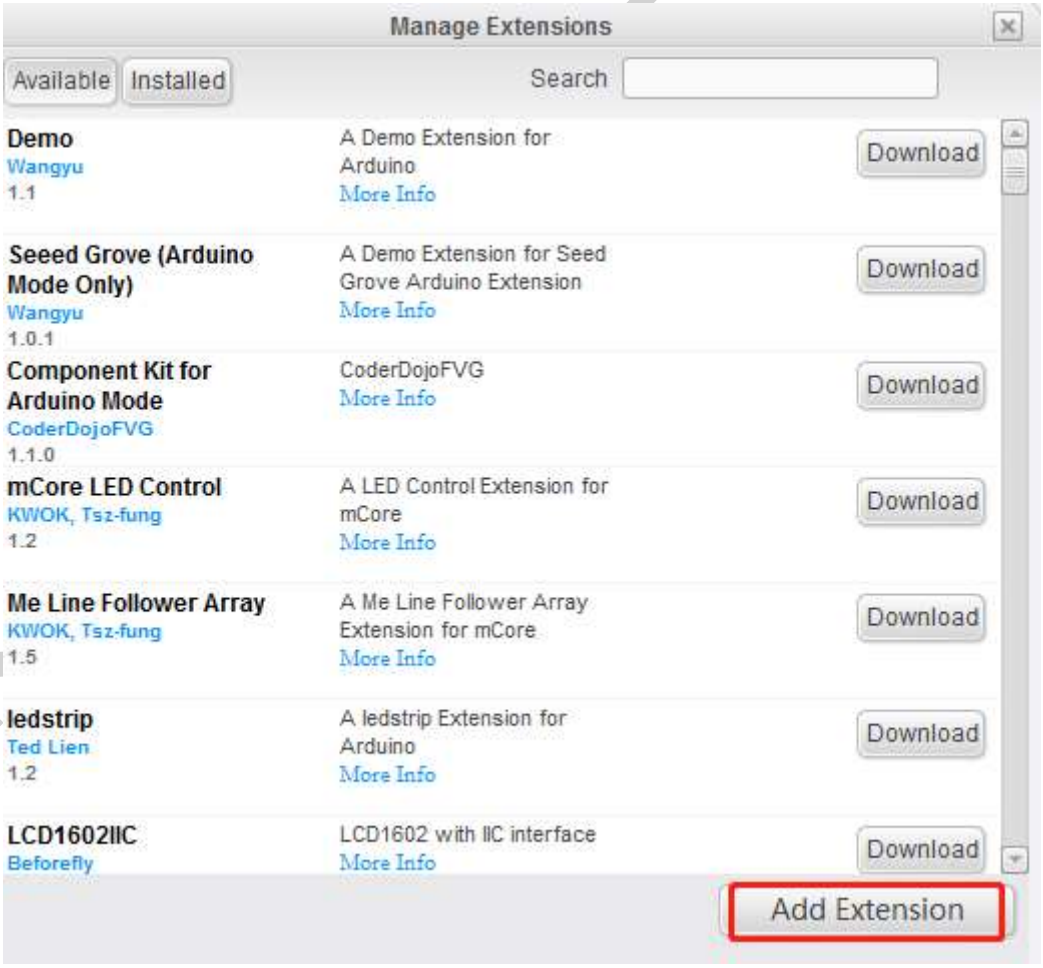


图 2-2

4) Select "Zip.file" for the file type, then select "Hummerbot.zip" and click "Open", as shown in Figure 2-3.

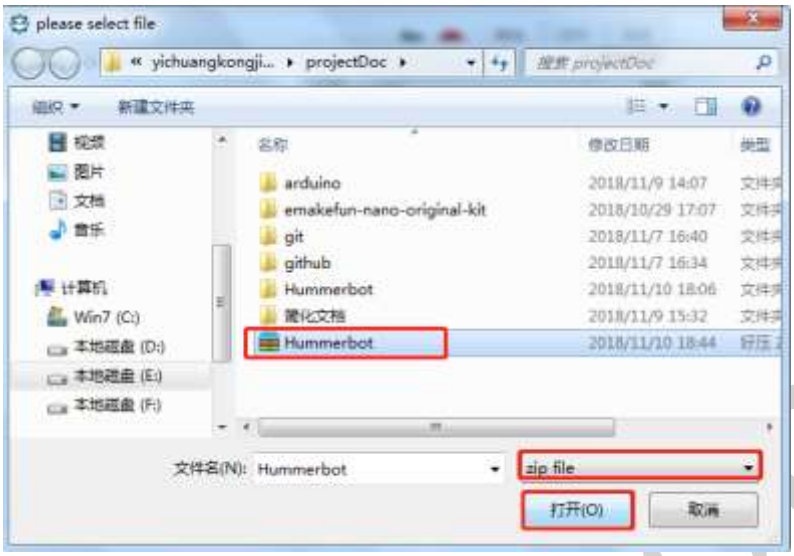


Figure 2-3

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

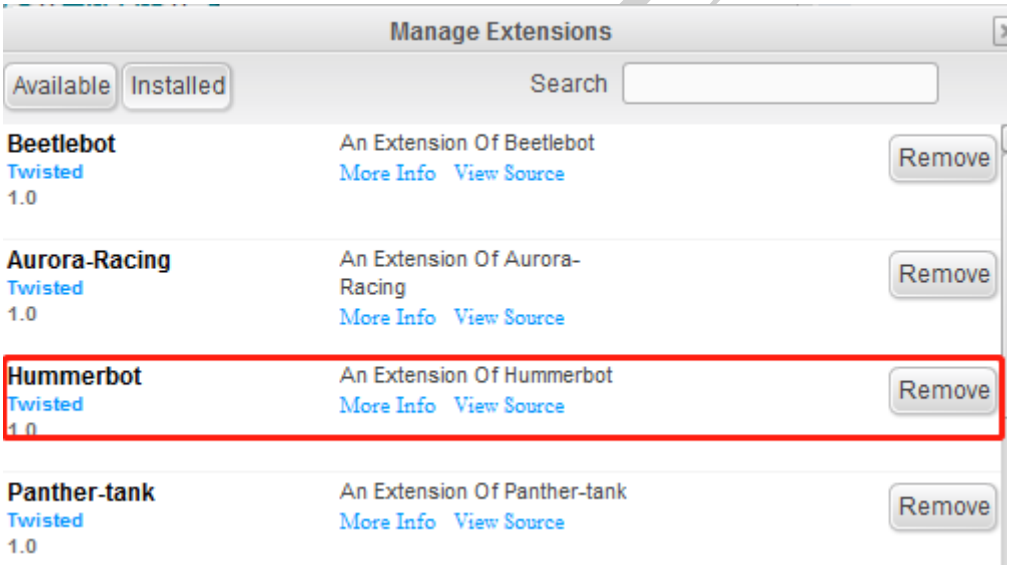


Figure 2-4

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

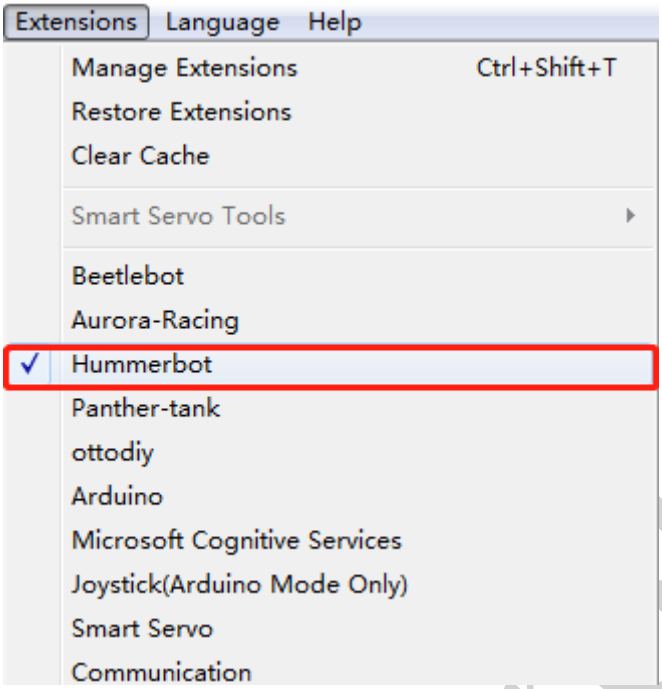


Figure 2-5

2.1.2 Programming experience

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

1) Firstly, drag the "Hummerbot 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 "Hummerbot main program" building block, as shown in Figure 2-6.

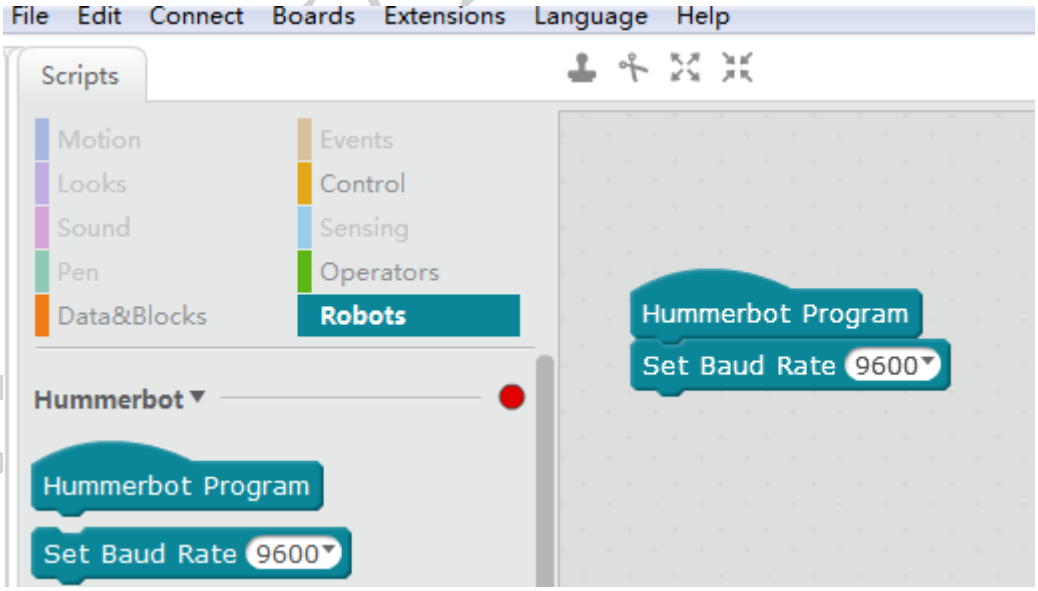


Figure 2-6

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

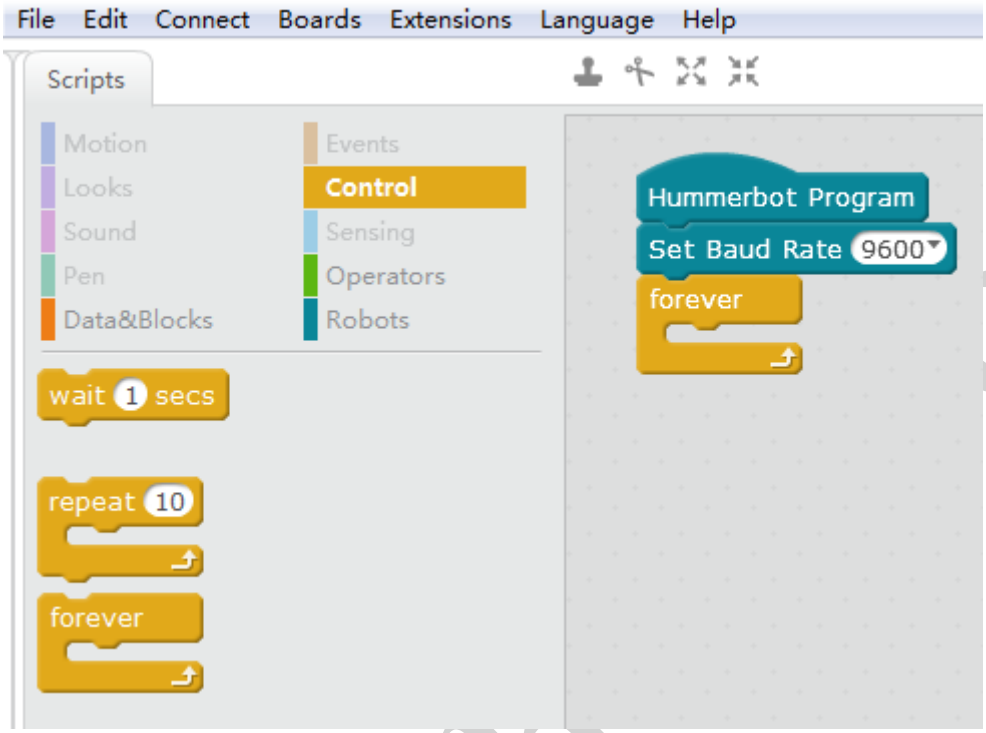


Figure 2-7

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

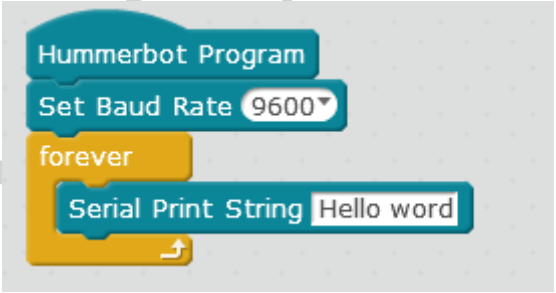


Figure 2-8

The above is the program that Hummerbot robot prints Hello Word. After the program is written, we need to transfer the program to the brain of the Hummerbot robot (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 Hummerbot robot (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 Hummerbot robot (control board). The following describes the connection method of the mBlock and the robot main control board.

2.2 mBlock and Hummer-bot Connection steps

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

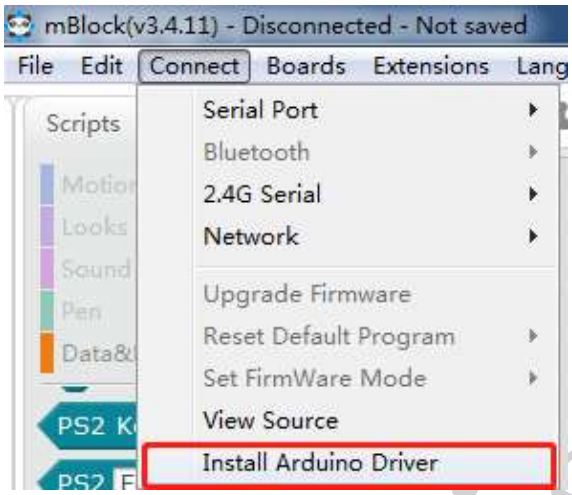


Figure 2-9

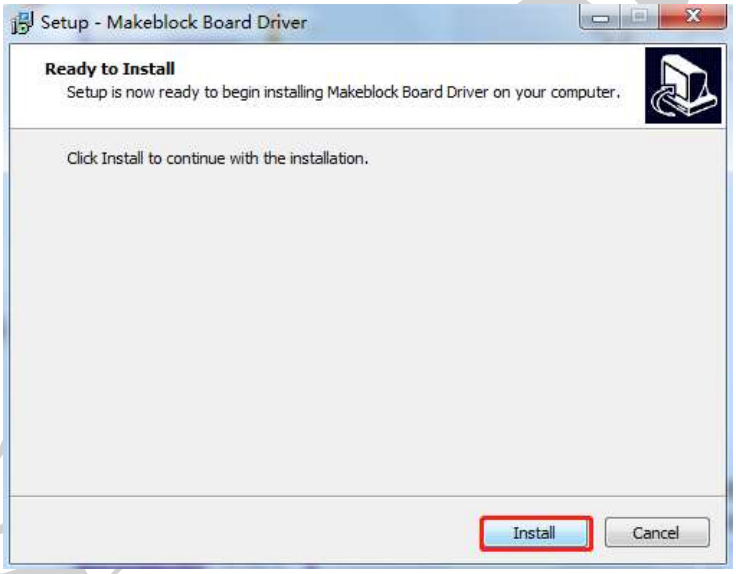


Figure 2-10

3) 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 Hummer-bot are successfully connected, as shown in Figure 2-12.

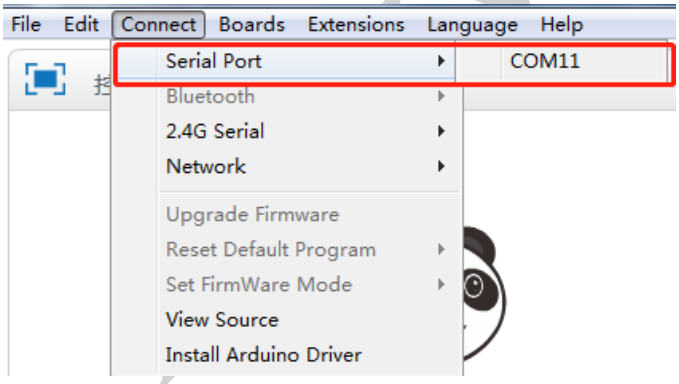


Figure 2-11

Figure 2-12

2.3 Upload a program to the Hummer-bot robot

When the program is written and the mBlock and Hummerbot robots are properly connected, we can transfer the program we wrote to the brain (main control board) of the Hummerbot 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.

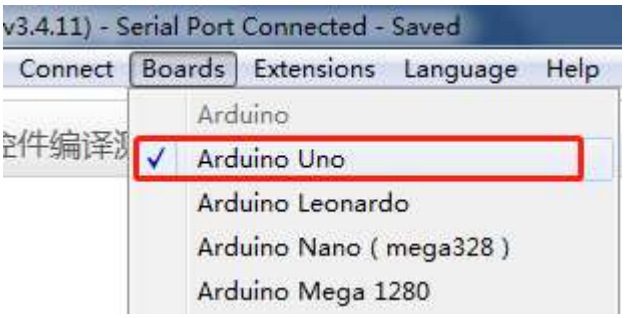


Figure 2-13

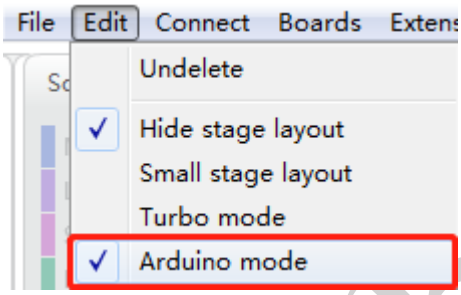


Figure 2-14

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 Arduino" to start generating the offline code of the building block and upload it to the Hummerbot 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

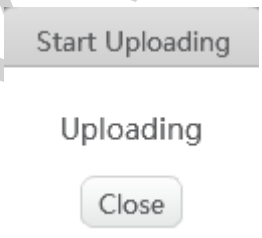


Figure 2-16

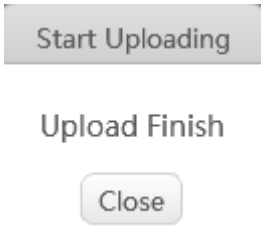


Figure 2-17

After completing the above steps, the Hummerbot 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 Arduino IDE" in the upper right corner, as shown in Figure 2-18.

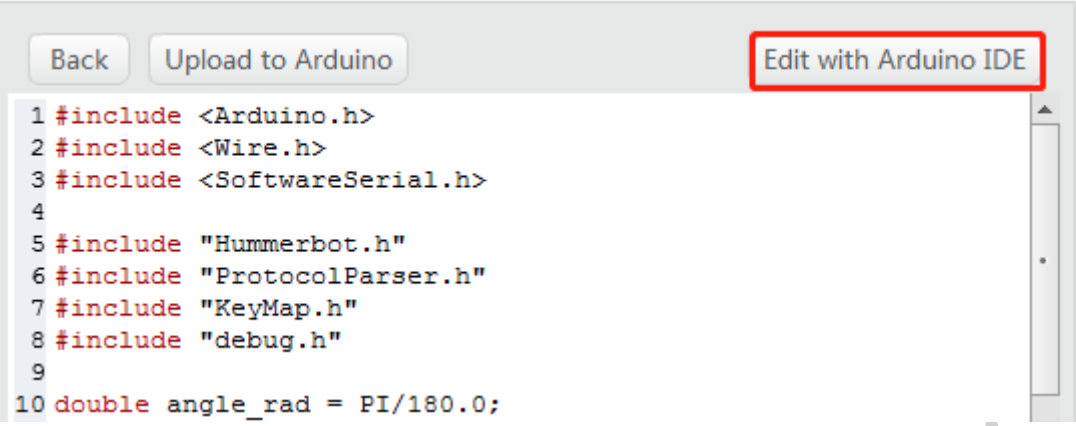


Figure 2-18

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

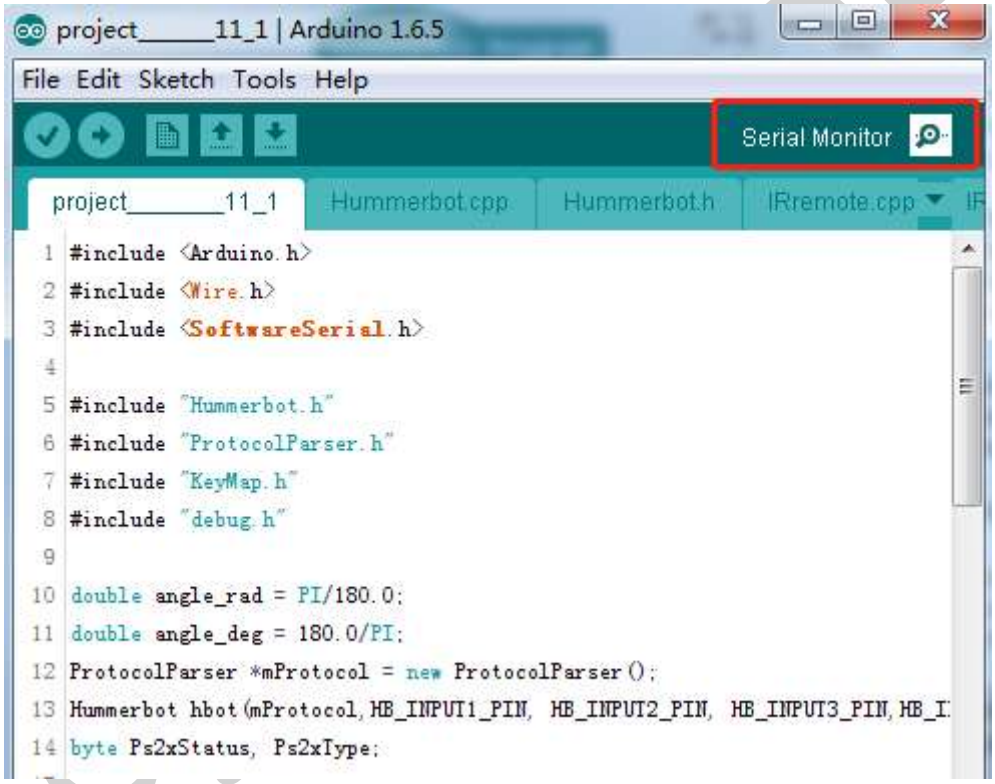


Figure 2-19

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.

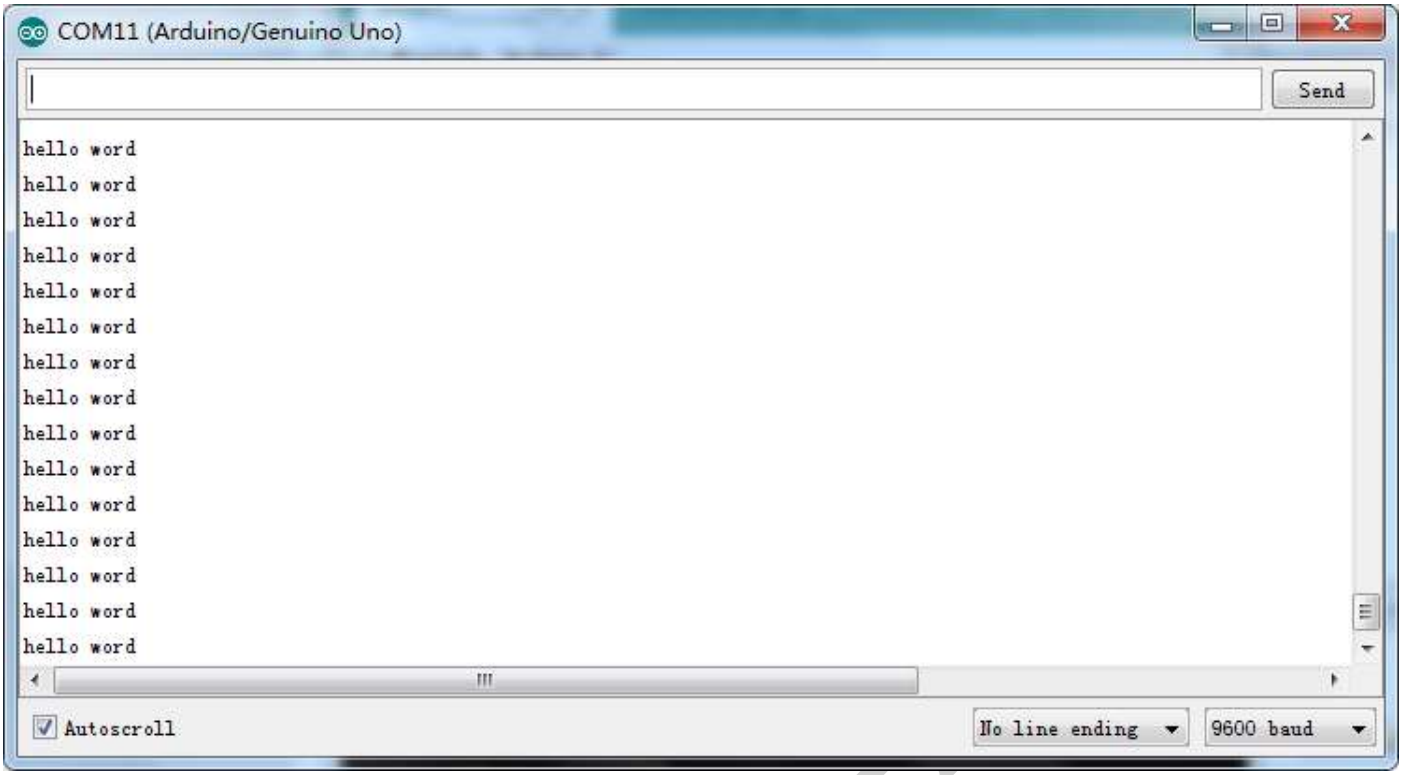


Figure 2-20

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 Hummer-bot robot has four DC motors. We can use a DC motor to control the building blocks to turn the DC motor up, which will drive the Hummer-bot 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.

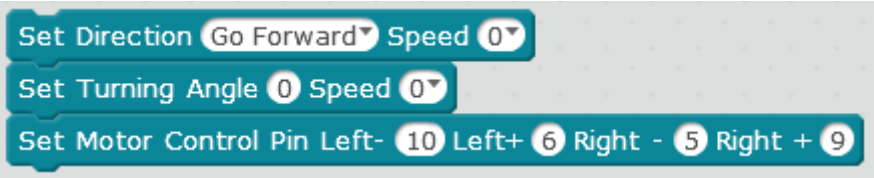


Figure 3-1

- Set the motor pin block: Set the connection pins of the four motors of the control robot and the main control board. This tutorial selects the default pin;
- Set direction and speed, speed building blocks: Set the direction of the robot movement and the speed in that direction;
- Set the turning angle and speed: Set the turning angle and speed when the robot moves.

Write the following four programs on the mblock, they can control the motor forward and reverse respectively, and then let the robot move forward, backward, left and right, run the program separately and carefully observe the robot moving in each direction. The direction of rotation of a DC motor, the programming in four directions is shown in Figure 3-2;

Figure 3-2

Through the above actual operation, we should have already understood the programming method of moving the robot in one direction by controlling the motor. Let's explore the programming method of making the robot go square.

3.1.3 Programmatically control and debug the robot to take the square route.

The way the robot moves in the square is to move forward→ turn left→ move forward → turn left → move forward →turn left→ move forward → turn left (also can reverse) the route of walking is shown in Figure 3-3, the reference procedure is shown in Figure 3- 4.

Figure 3-3

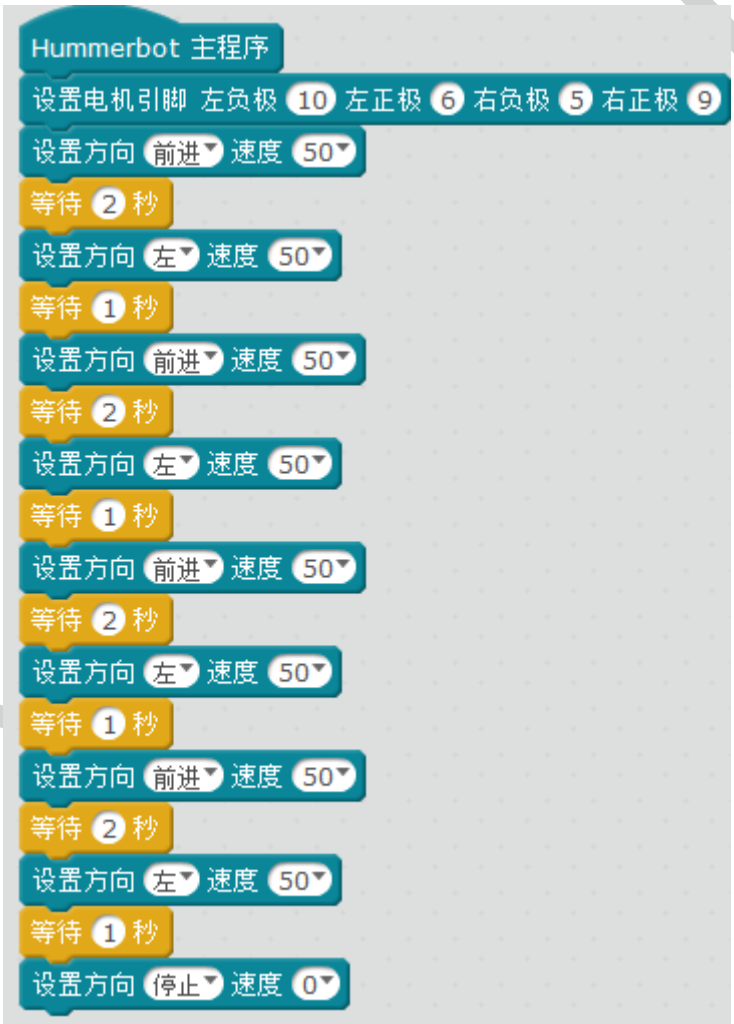


Figure 3-4

Note:

1) The time delay building block under the “forward” building block is to control the forward distance of the robot. The delay building block under the left turn is to control the turning angle of the robot. When programming, the delay time should be modified according to the actual walking condition of the robot.

2) At the end of the program, we're sure to stop the motor from rotating. Otherwise, the robot will keep the original motion and keep turning left.

3.1.4 Discover program running rules and optimize programs

By observing the procedure in Figure 3-4, we will find that the robot's route to the square is actually to control the wheel to repeat four times and turn left 90°, the optimization procedure is shown in Figure 3-4;

Figure 3-5

3.2 Ultrasonic module

3.2.1 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.

3.2.2 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 3-6, 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.

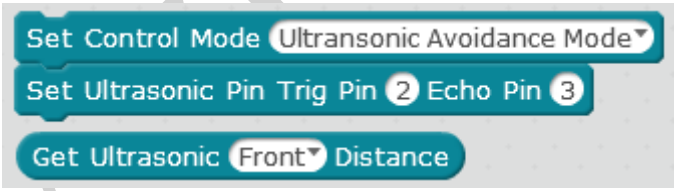


Figure 3-6

3.2.3 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 firstly.

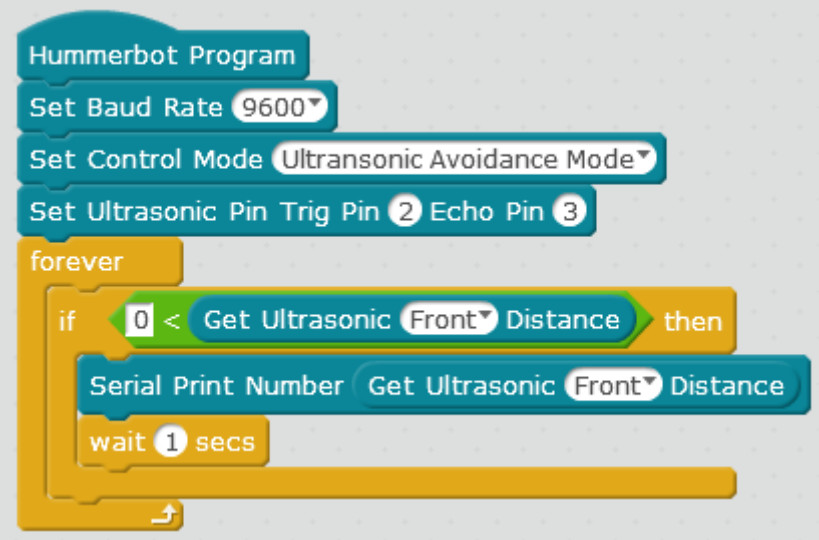


Figure 3-7

We write the program as shown in Figure 3-7, 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.

3.2.4 Write a robot program that can dance tango

Tango dance is a kind of double dance. The dance step is when one person approaches and the other person stays away. To let the robot dance with our palms, the hand is close to it and the hand moves away from it. Then the ultrasonic module is used to measure the hand and the robot. The distance thus determines whether the robot is moving forward or backward. The measurement distance of the ultrasonic becomes larger, indicating that the hand is far away from the robot. After a certain distance, the robot advances; the measurement distance of the ultrasonic becomes smaller, indicating that the hand is close to the robot, and after approaching a certain distance, the robot retreats; We set the intermediate value to 20CM firstly. Then let's start programming.

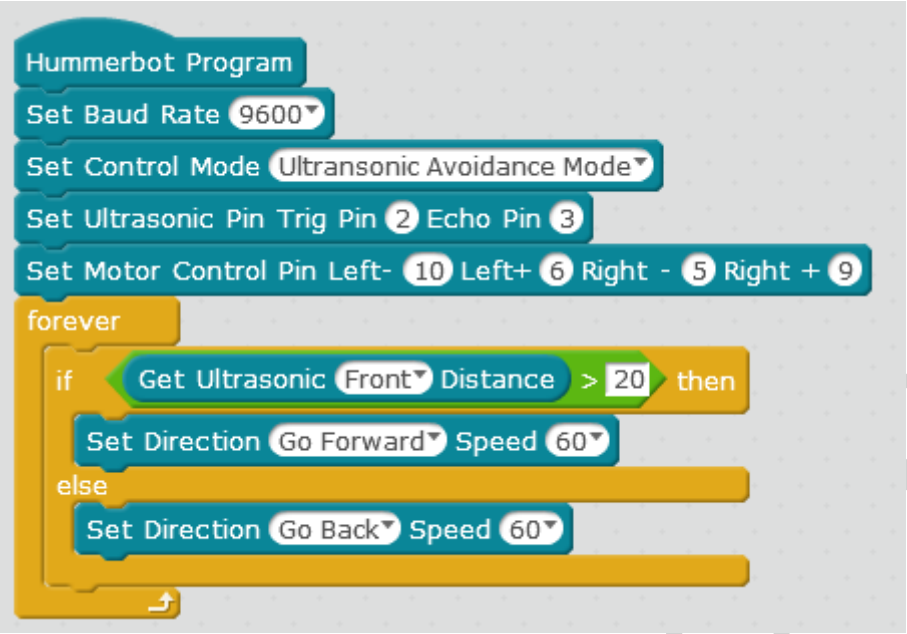


Figure 3-8

Running the program in Figure 3-8 above. When the hand is away, the robot advances; When the hand approaches, the robot moves backward. But when the hand is stationary, the robot will move back and forth repeatedly because we did not set a stop motion for the robot. So how do you stop the robot in an interval?

Value measured by the ultrasonic module	Robot's motion state
>20	move advance
12~20	still
<12	move backward

Table 3-9

As shown in Table 3-9 above, for a situation between 12 and 20, a new judgment statement is needed, as shown in Figure 3-10. When the distance of the ultrasonic measurement is greater than 12CM and less than 20CM, the robot stops.

Figure 3-10

3.2.5 Optimized dancing robot program

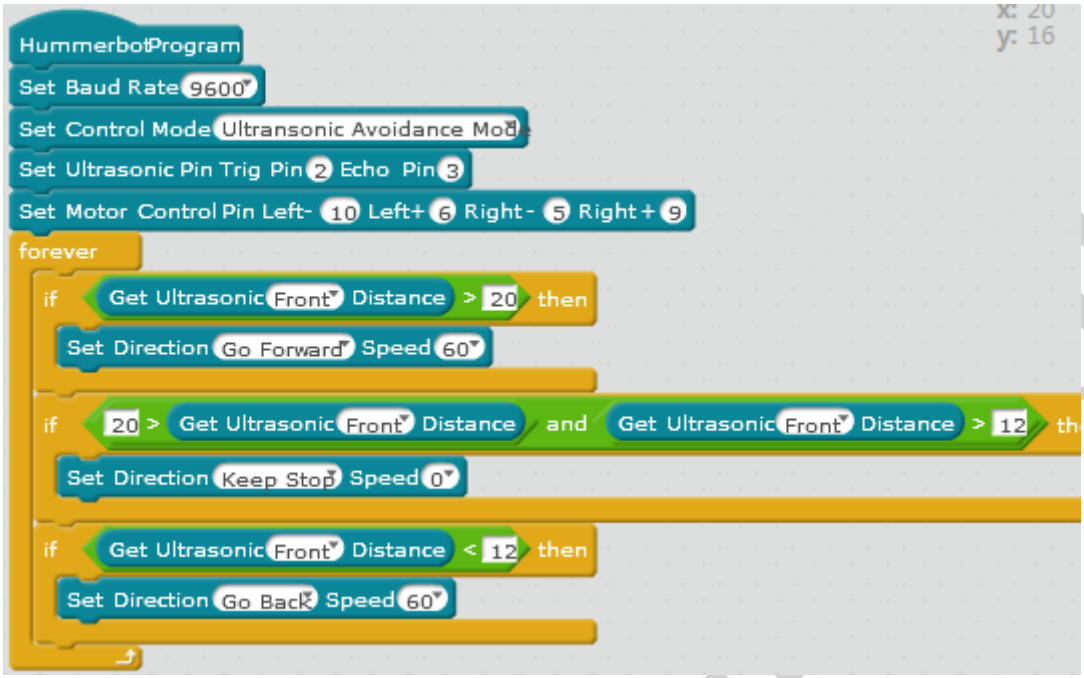


Figure 3-11

3.3 Ultrasonic obstacle avoidance robot

We have already learned about the role of robot forward, backward, left turn, right turn and ultrasonic module in robot movement. Let's explore how robots use ultrasonic to avoid obstacles in motion. The so-called robot obstacle avoidance is to let the robot when exercising, if there is an obstacle in front of it, then the obstacle will continue to move around (left turn or right turn or backward), the robot should turn left and right and judge the distance in the left and right direction, then the ultrasonic module will rotate left and right to explore"Road".At this time the servo is used to realize the ultrasonic module to turn around the road. How does the servo work? Let’s firstly introduce the use of the servo.

3.3.1 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 3-11. 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 3-12

shows the physical map of the most commonly used SG90 servo at present.

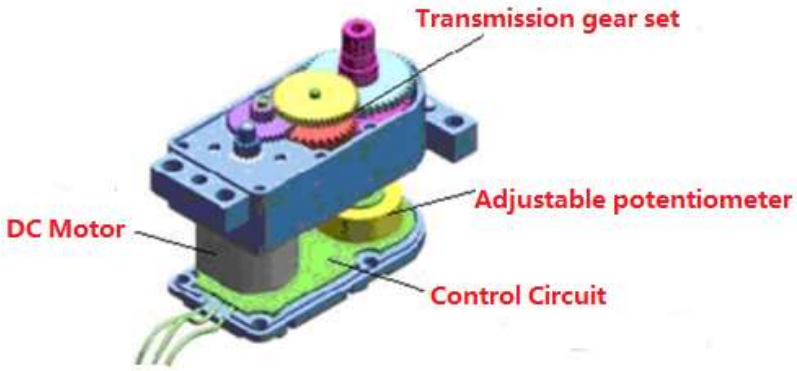


Figure 3-11 Schematic diagram of the servo

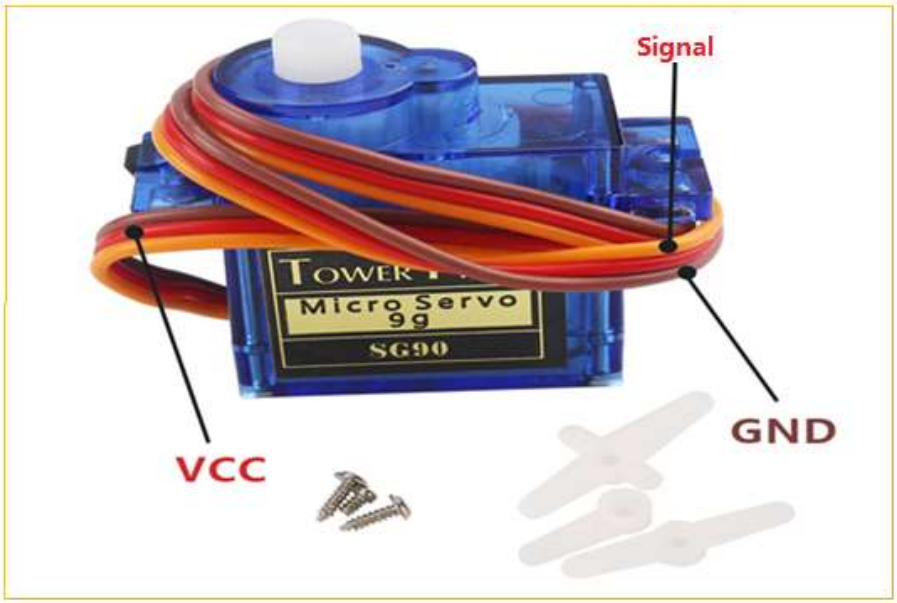


Figure 3-11 Physical map of the servo of SG90

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.

3.3.2 Servo Test

The servo on the Hummerbot robot is installed on the front, and the ultrasonic module is fixed on the servo. When the ultrasonic module measures the distance between the robot and the front obstacle, the servo is 90°. When the ultrasonic module measures the distance between the robot and the obstacle on the left, the servo rotates 90° to the left; When the ultrasonic module measures the distance between the robot and the

right obstacle, the servo rotates 90° to the right. Let's write a program that uses the servo to maket the ultrasonic firstly go forward, then turn left, finally turn right.

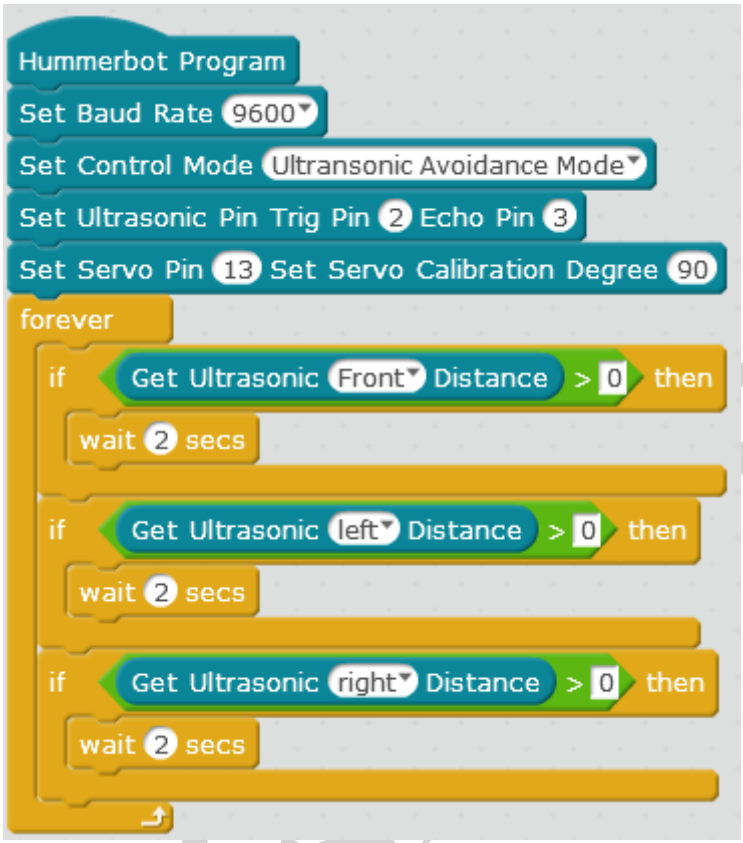


Figure 3-12

The above is the test procedure of the servo. After we upload the program to the robot, turn on the power and you will see the servos make the ultrasonic module turn forward and then to the left, nextly to the right.

3.3.3 Writing obstacle avoidance robot programs

The main working process of the ultrasonic obstacle avoidance robot is: after power-on, the servos automatically turn the ultrasonic module to 90 degrees (aligned directly in front of the robot), and the ultrasonic module measures the distance of the obstacle in front, if the value is greater than the safe distance which is set (12CM), the robot continues to move forward, otherwise stops. At this time, the servo rotates the ultrasonic 90 degrees to the right, and the ultrasonic module measures the distance from the obstacle on the right side. Then the servo rotates the ultrasonic to the left 180 degrees, the ultrasonic module measures the distance from the left obstacle, and then the servo returns the ultrasonic module to 90 degrees. The robot compares the measured distances. If the left is greater than the right, the robot turns to the left and then drives, otherwise turns right, if the distance measured on both the left and right sides is less than the safe distance, the robot will turn around and walk. Following this situation, let's start to write a program for obstacle avoidance robots.

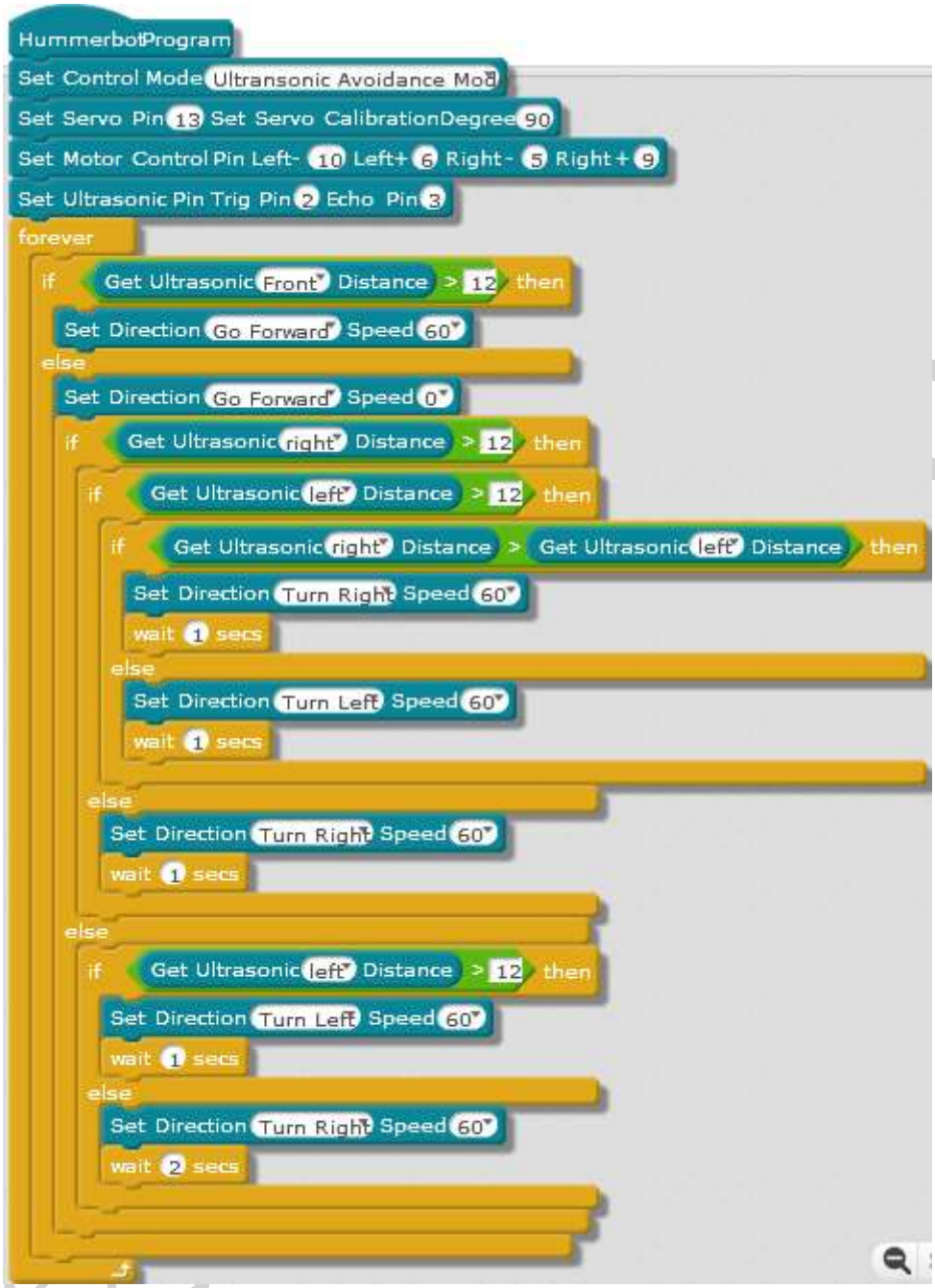


Figure 3-13

3.4 Infrared obstacle avoidance robot

3.4.1 Infrared obstacle avoidance module principle

The principle of the infrared obstacle avoidance module and the ultrasonic module is similar. The difference is that the ultrasonic module measures the distance by transmitting ultrasonic and the infrared obstacle avoidance module measures the distance by emitting infrared rays. The infrared obstacle avoidance module has a pair of infrared emitting and receiving tubes, and the transmitting tube emits Infrared. When the emitted infrared ray encounters an obstacle (reflecting surface), the infrared light is reflected back and

received by the receiving tube, the green indicator light is on and the received infrared signal is processed by the robot's control board to calculate the detection distance, and the infrared avoidance is performed. The infrared barrier module is shown in Figure 3-14.

Figure 3-14

3.4.2 How to use the infrared obstacle avoidance module

We found the control blocks of the infrared obstacle avoidance module in the control module of the robot, as shown in Figure 4-2, which can be used to detect the distance between the robot and the obstacles on the left and right. Note: When using the infrared obstacle avoidance module, first we need to select the mode control block and set the mode as the infrared obstacle avoidance mode. When the obstacle completely blocks the ultrasonic module or the ultrasonic module faces far away, the infrared obstacle avoidance module sends out the infrared rays cannot be received because they are too far or too close, and the distance between the robot and the obstacle cannot be measured. Therefore, when using the infrared obstacle avoidance module to measure the distance, the distance from the obstacle is preferably between 5 and 15.

Figure 3-15

3.4.2 Infrared obstacle avoidance module test

We can firstly write a program to test the infrared obstacle avoidance module, use the serial port to print. When we are close to the infrared obstacle avoidance module and away from the infrared obstacle avoidance module, we observe the serial print content, you can see the infrared obstacle avoidance module more intuitively of the process of measuring the distance. Let’s write a test program first.

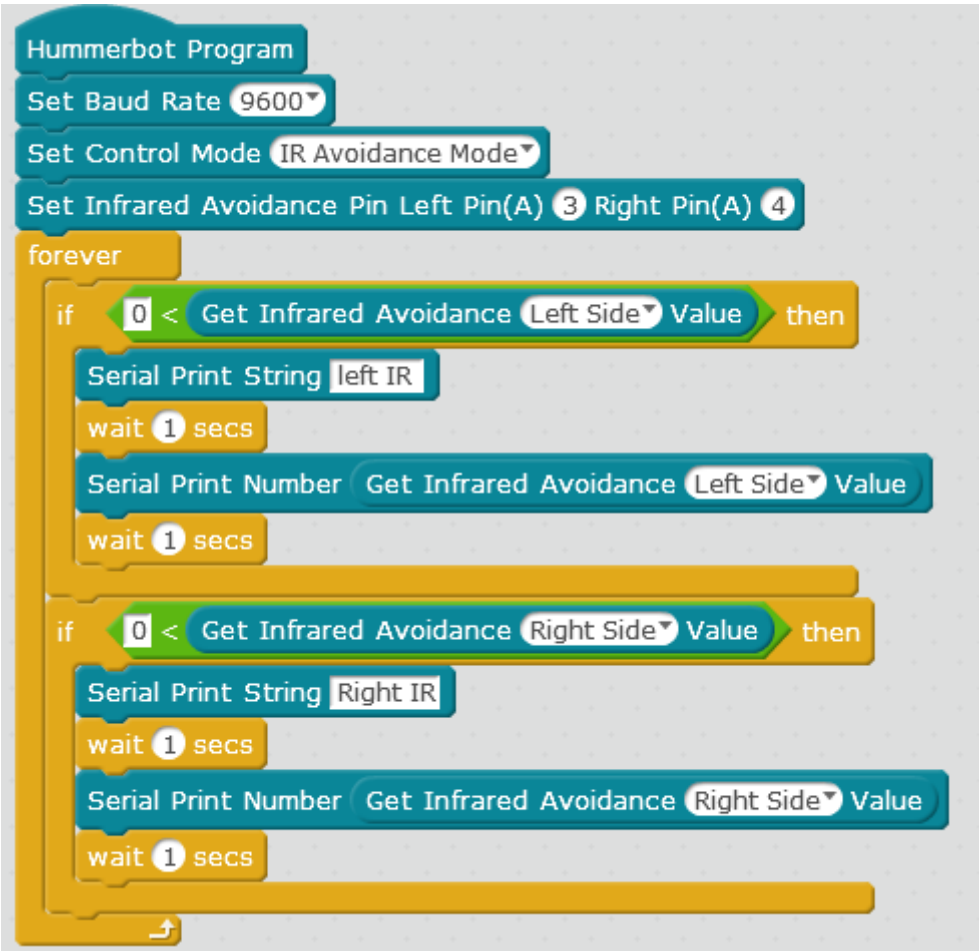


Figure 3-14

Firstly, we write the program as shown in Figure 3-14, and then upload the program successfully, open the serial port, and nextly move the hand close to or away from the infrared obstacle avoidance module, the serial port will print the corresponding distance.

3.4.4 Write a robot running the track

If we want the robot to run smoothly on the track to avoid hitting the track. It is necessary to keep the robot at a certain distance from both sides of the track during the movement. At this time, we can use the infrared obstacle avoidance module to achieve this effect. The basic idea is to firstly set the safe distance of the infrared obstacle avoidance module, the robot walks forward, the infrared obstacle avoidance module measures the distance from the edge of the right track, and then the infrared obstacle avoidance module measures the distance from the edge of the left track. Then the robot compares the distances measured twice. If the left is greater than the right, the robot deflects to the left and then travels, otherwise turns right. If the distance measured on both the left and the right is less than the safe distance, the robot will stop walking. Follow this thought, let's learn how to write a robot that runs the track.

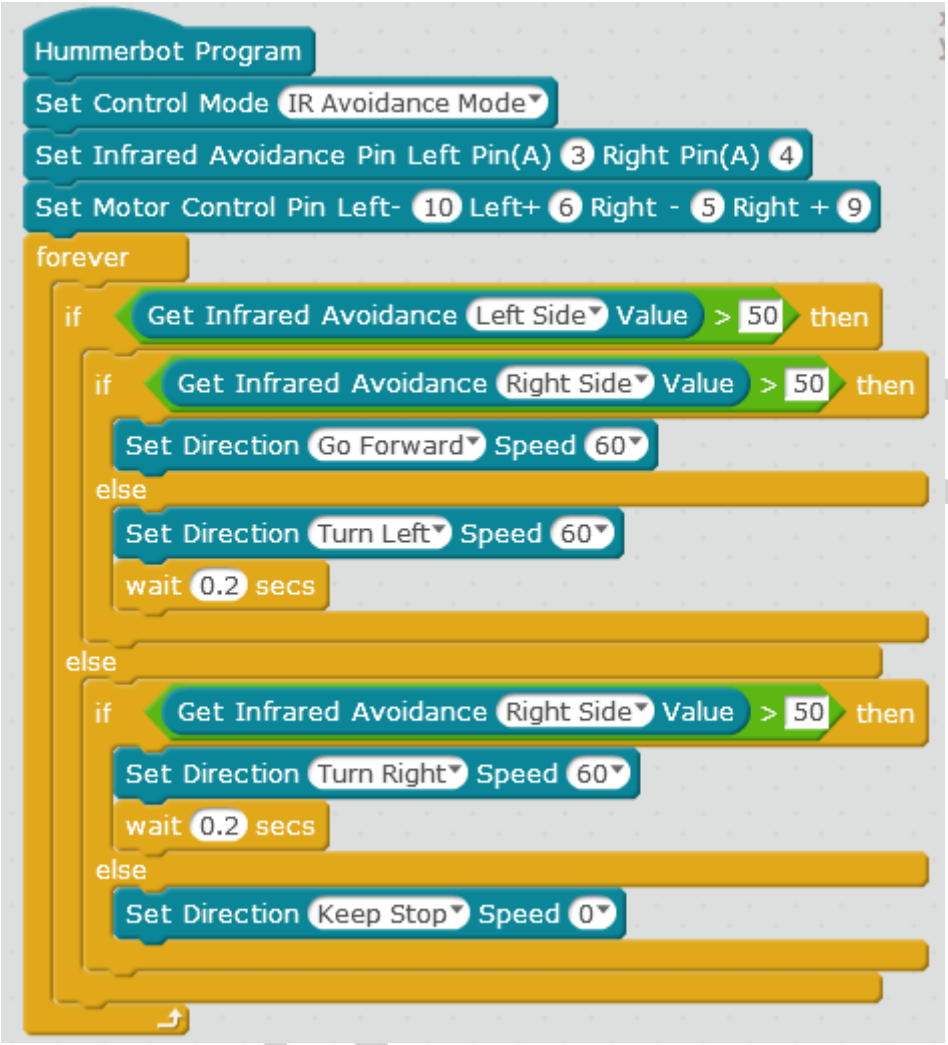


Figure 3-16

Run the program in Figure 3-16 above and the robot advances; when the left side of the robot is near the edge of the runway, the robot turns right. When the right side of the robot is near the edge of the runway, the robot turns left. When the left and right sides of the robot are close to the obstacle, the robot stops.

3.5 Ultrasonic + infrared obstacle avoidance robot

We have already learned about the separate use and programming of the ultrasonic module and the infrared obstacle avoidance module. Nextly let's we combine the ultrasonic module and the infrared obstacle avoidance module together. The programming idea is to set the mode to ultrasonic infrared obstacle avoidance mode, and then set ultrasonic module, safe distance of infrared obstacle avoidance module, then ultrasonic detection of obstacles in front of the robot, infrared obstacle avoidance module to detect whether there are obstacles on the left and right sides of the robot.Let's write the program of ultrasonic infrared obstacle avoidance robot together.

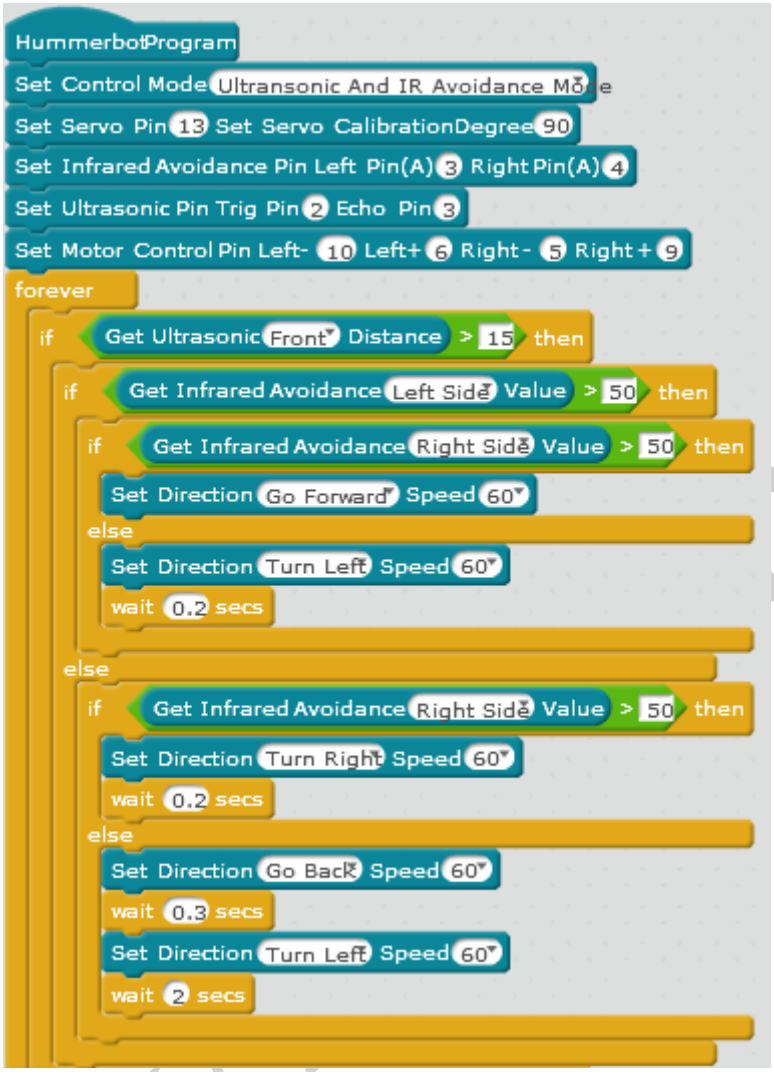


Figure 3-17

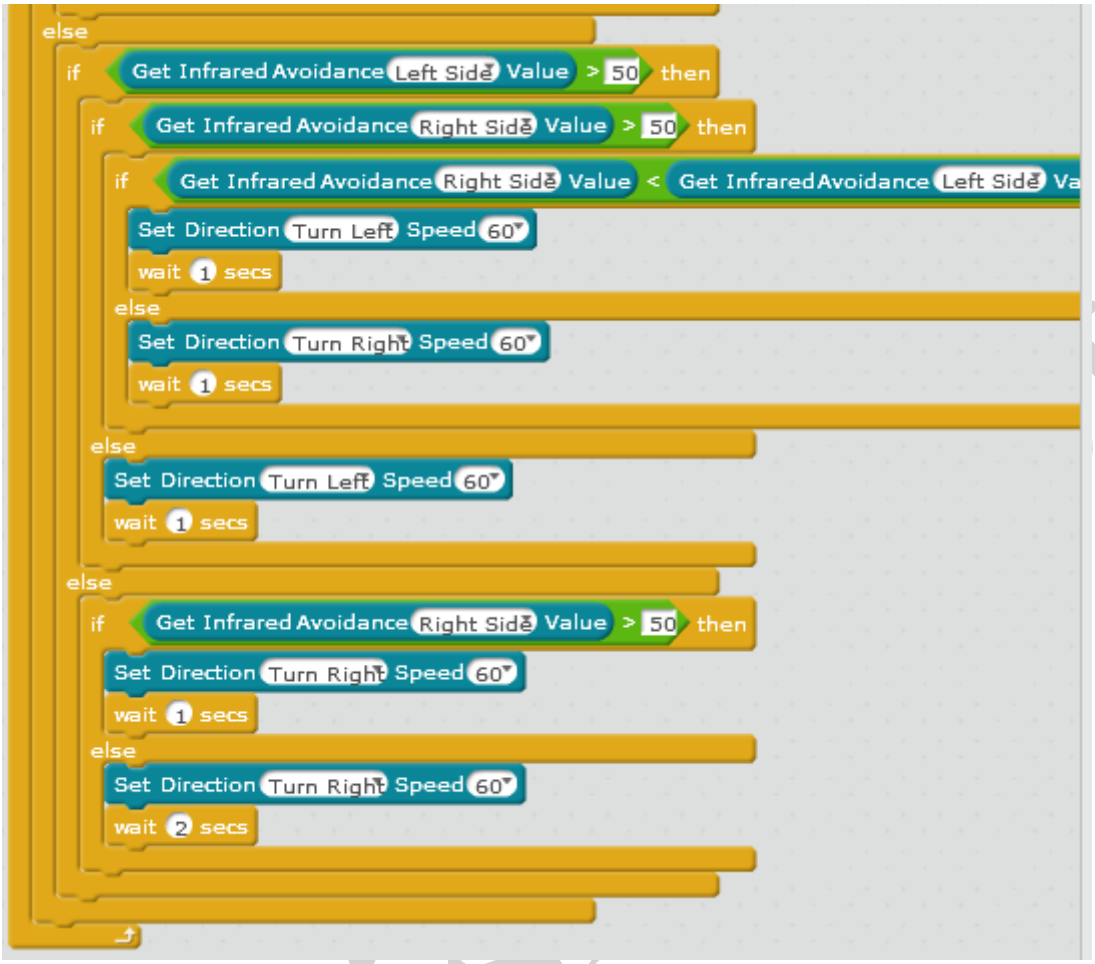


Figure 3-18

Figure 3-17 and Figure 3-18 show the procedure of the ultrasonic infrared obstacle avoidance robot. In the above procedure, the delay time after steering needs to be adjusted according to the speed of the robot. This should be noted during programming.

Chapter 4 Tracking Robot

4.1 Tracking module principle

The tracking module works by using infrared rays to have different reflective properties on the surface of objects of different colors. During the movement of the robot, the infrared light is continuously emitted to the ground, and the emitted infrared rays are not reflected back or reflected back while the intensity is not large enough, the robot thinks that the area is the route it wants to take; When the infrared light is reflected back and recognized by the robot, the robot thinks that the area is not the route it takes. The robot determines the position of the black line and its walking route based on the reflected infrared light. The Hummerbot robot is equipped with three tracking modules on the left, middle and right sides. And the tracking principle

is shown in Figure 4-1.

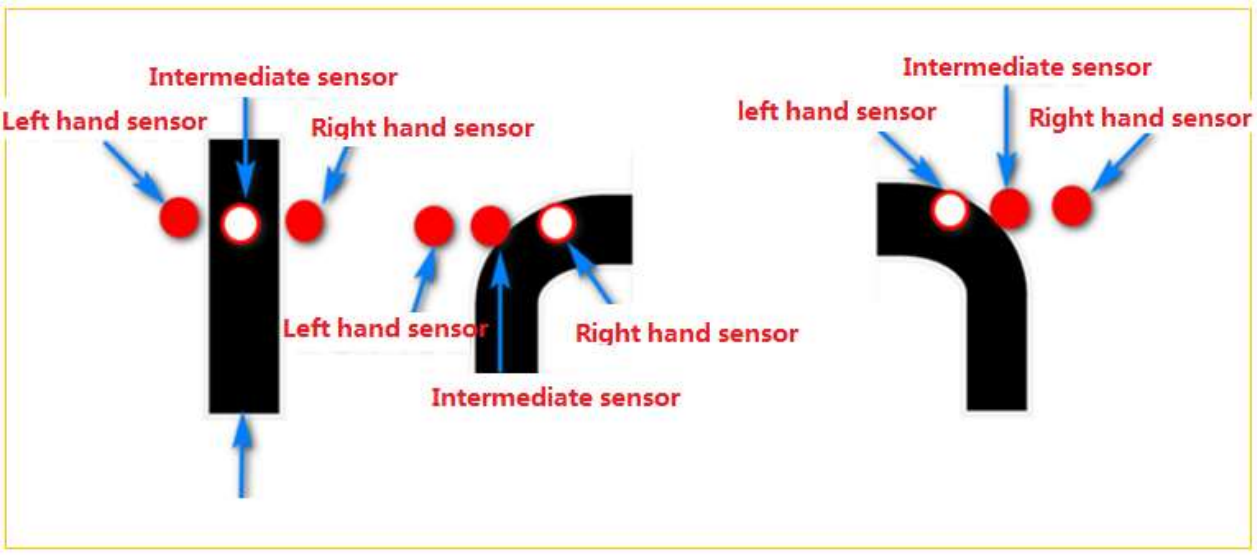


Figure 4-1

4.2 Use of tracking module

Find the control blocks of the infrared tracking module in the robot's control module, as shown in Figure 4-2. Using this brick allows the robot to automatically run along its own track line. Note: When using the infrared tracking module, firstly we need to select the mode control block, set the mode to infrared tracking mode, and then select the color of the tracking value according to the color of the track line.

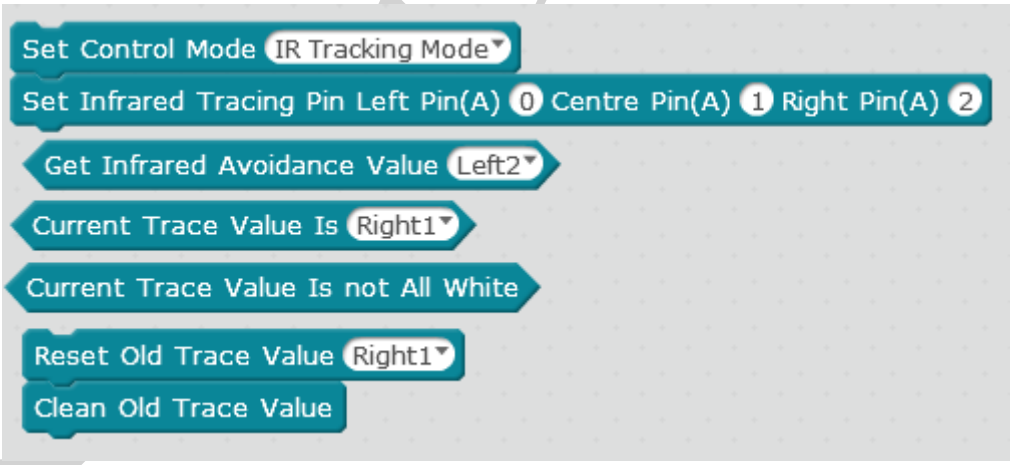


Figure 4-2

4.3 Infrared tracking module test

We can firstly write a program to test the infrared tracking module, use the serial port to print, use the black tape to close to the infrared tracking module and stay away from the infrared tracking module. We can observe the content of the serial port printing, so that we can see the infrared tracking more intuitively of the tracking effect of each module when the module tracks. Let’s write a test program firstly.

Figure 4-2

As shown in Figure 4-2 above, when we open the serial port, move the black strip to any one of the tracking modules, and the serial port will print the corresponding number.

4.4 Writing a tracking robot program

The programming idea of the tracking robot is to set the mode to the infrared tracking mode firstly. Nextly set the connection pin of the infrared tracking module according to its actual wiring mode.And then the robot performs the corresponding detection according to the results of the three left, middle and right tracking modules to move to the left, forward, and right. Let's write the program for the tracking robot.


HummerbotProgram
Set Control Mode IR Tracking Mode
Set Motor Control Pin Left- 10 Left+ 6 Right - 5 Right + 9
Set Infrared Tracing Pin Left Pin(A) 0 Centre Pin(A) 1 Right Pin(A) 2
forever
DefinesA VariableSave Old Trace Value
if Get InfraredAvoidance Value All Black then
Set Direction Keep Stop Speed 0
else
if Get InfraredAvoidance Value All White then
if Current Trace Value Is Right1 then
repeat until Current Trace Value Is not All White
Set Turning Angle 174 Speed 70
Clean Old Trace Value
else
if Current Trace Value Is Left1 then
repeat until Current Trace Value Is not All White
Set Turning Angle 6 Speed 70
Clean Old Trace Value
else
if Current Trace Value Is Right2 then
repeat until Current Trace Value Is not All White
Set Turning Angle 150 Speed 70
Clean Old Trace Value
else
if Current Trace Value Is Left2 then
repeat until Current Trace Value Is not All White

Figure 4-3

Chapter 5 Infrared Remote Control Robot

5.1 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 5-1.

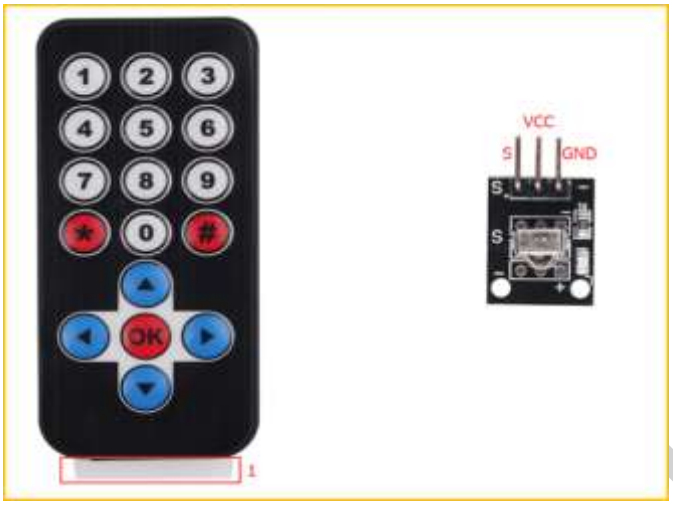


Figure 5-1

5.2 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 5-2

5.3 Infrared remote control test

We can firstly write a program to test the infrared remote control, use the serial port to print, when the remote control button is pressed, the serial port prints the corresponding button key value.We observe the serial port printing content, we can see which button of the infrared remote control is pressed.Let's write a test program.

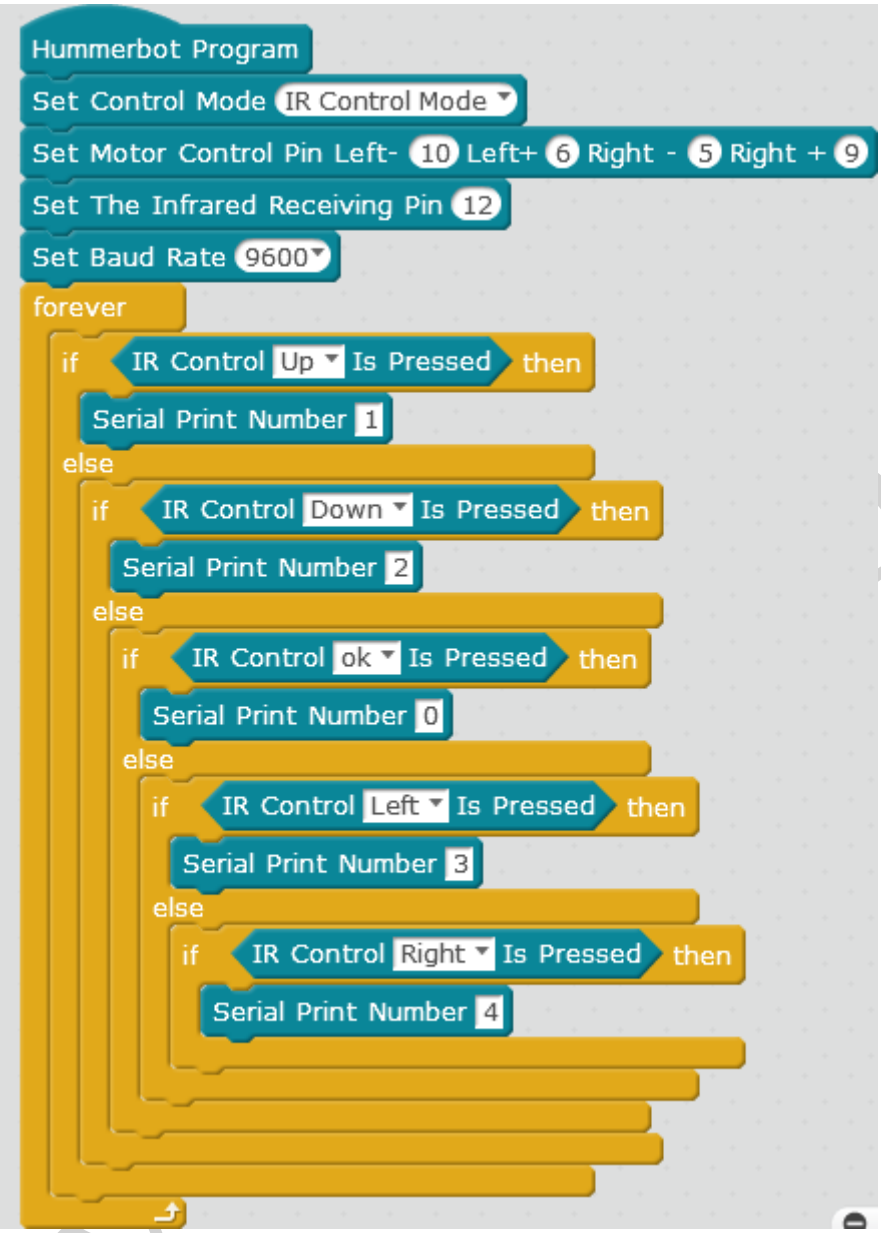


Figure 5-3

When we write the program as shown in Figure 5-3 and upload the program successfully.Then open the serial port and press the remote control to press the 1, 2, 3 and 4 keys to the robot, the serial port will print the corresponding button number.

5.4 Remote robot programming

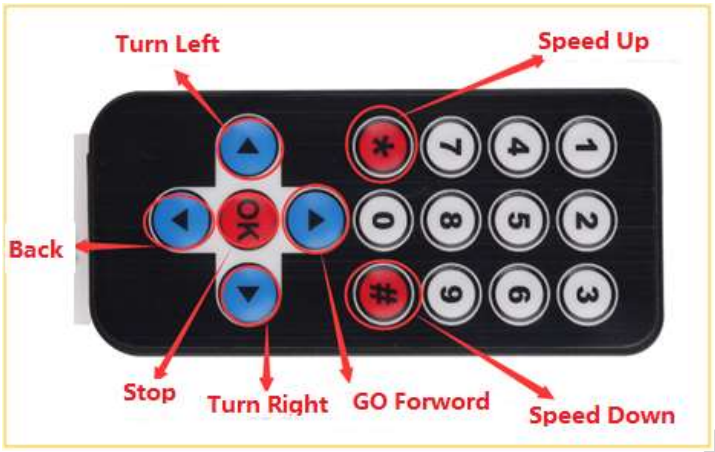


Figure 5-3

We write a remote control button function as defined in Figure 5-3, and set speed adjustment 1 (*) as acceleration and speed adjustment 2 (#)as deceleration.

Figure 5-4

Note: When setting the speed of steering, the speed value can't be set too small. Otherwise, the car will be stuck in place because of the force is not enough. Therefore, the speed value has better to be set above 40.

Chapter 6 Bluetooth Remote Control Robot

6.1 Bluetooth control principle

Bluetooth is a kind of remote communication method. We send the command from the mobile phone APP to the Bluetooth module on the robot through the mobile phone Bluetooth, and then the Bluetooth module on the robot sends the command to the robot brain (main control board).After that the robot will do

corresponding action by controlling its four limbs (four wheels). At the same time, the robot will also send the action being made to the mobile phone APP through the Bluetooth module installed by itself, so that the mobile APP can control the robot through Bluetooth.

6.2 The connection method of Bluetooth module and Hummerbot robot APP KeywishBot

The Hummerbot robot is equipped with a Bluetooth module (as shown in Figure 6-1). After the power is turned on, the blue light on the Bluetooth module will flash. Then open the APP and select Hummerbot (as shown in Figure 6-3), Hummerbot robot APP KeywishBot (shown in Figure 6-2) of the Bluetooth module and the phone is connected via Bluetooth. After the connection is successful, the blue light is on, and nextly select “Remote Control Interface” (Figure 6-4) to enter the gravity sensing control interface, as shown in Figure 6-5. You can also switch the handle control mode as shown in Figure 6-6.

Figure 6-1

Figure 6-2

Figure 6-3

Figure 6-4

Figure 6-5

Figure 6-6

6.3 Bluetooth module test

We can firstly write a program to test the Bluetooth module, use the serial port to print, when the Bluetooth module of the robot receives the data, the serial port prints the data received by Bluetooth;We observe the content of the Bluetooth transmission and the content printed by the serial port, we can see the effect of the data of the Bluetooth transmission.Let’s write a test program.

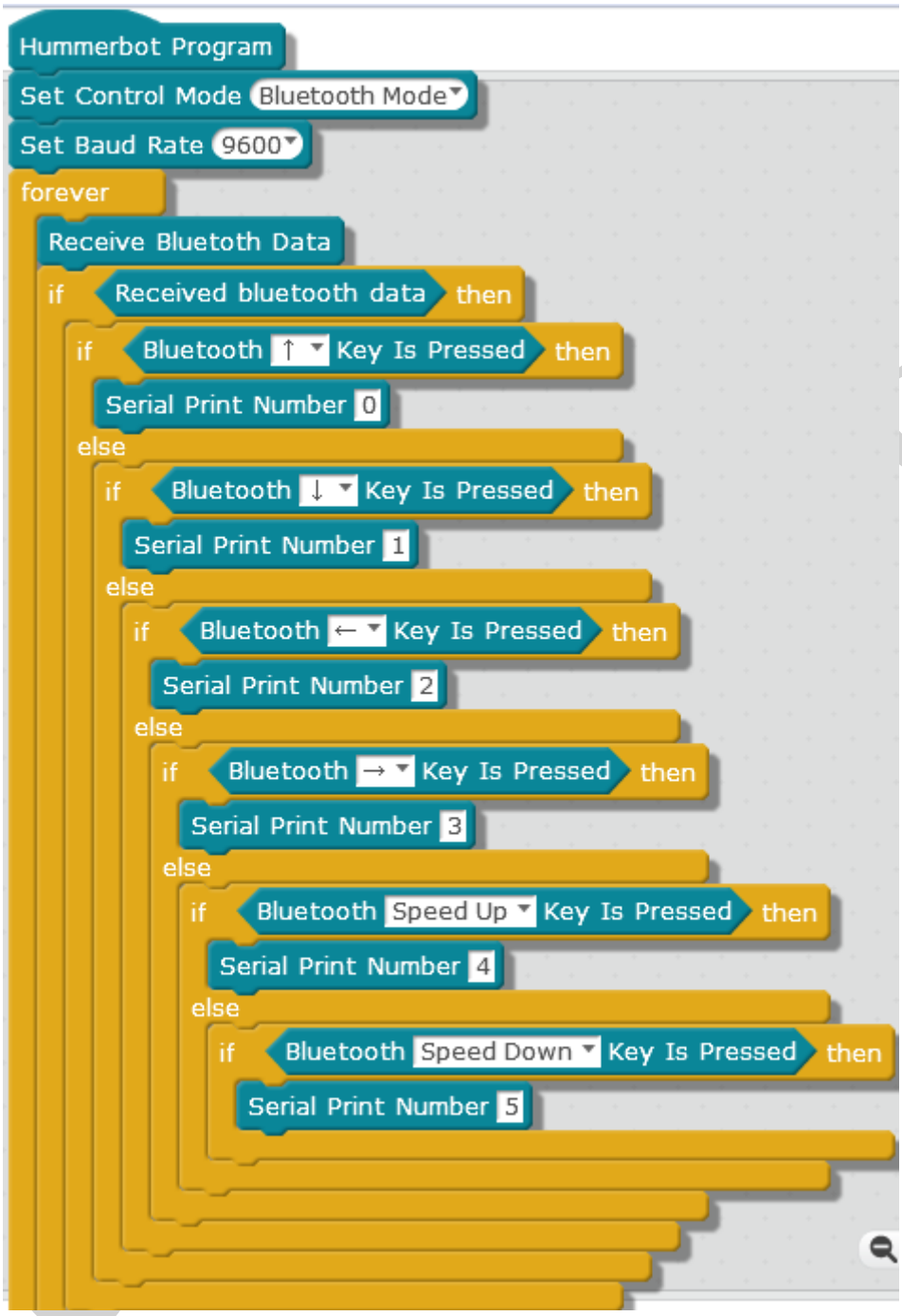


Figure 6-7

When we write the program as shown in Figure 5-3 and upload the program successfully, open the serial port, then use the mobile phone APP and the robot Bluetooth connection, press the up, down, left and right buttons, the serial port will print the corresponding button number.

6.4 Programming ideas of Bluetooth remote control robot

The programming idea of the Bluetooth remote control robot is to set the mode as the Bluetooth control mode firstly, initialize the serial port, and then the program Bluetooth module waits to receive the command

sent by the mobile APP, and the robot will perform the corresponding action according to the instruction. Let's write program of.the Bluetooth remote control robot together.

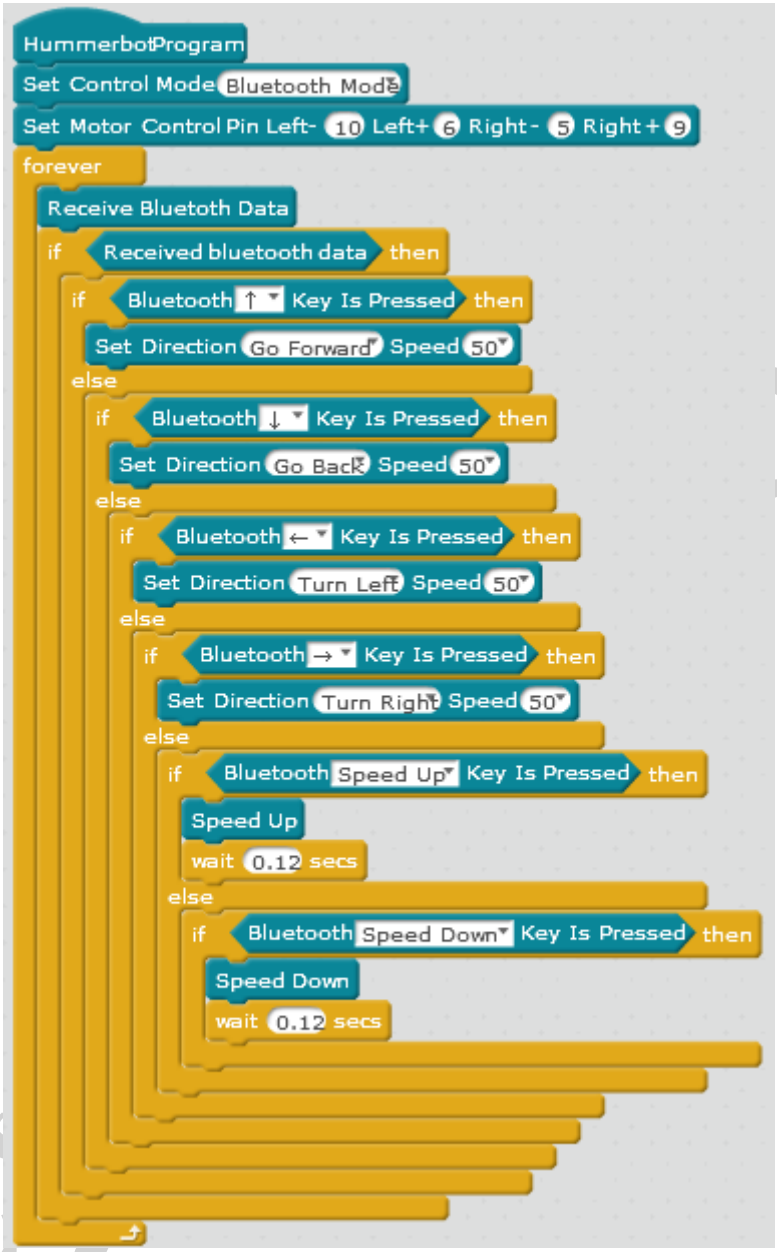


Figure 6-8

In the above programming, note that we have made a 0.12S delay after the acceleration and deceleration program. This is to make the robot accelerate and decelerate slowly. If there is no delay, the speed will directly change to the maximum value or the minimum.

Chapter 7 PS2 Remote Control Robot

7.1 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 7-1

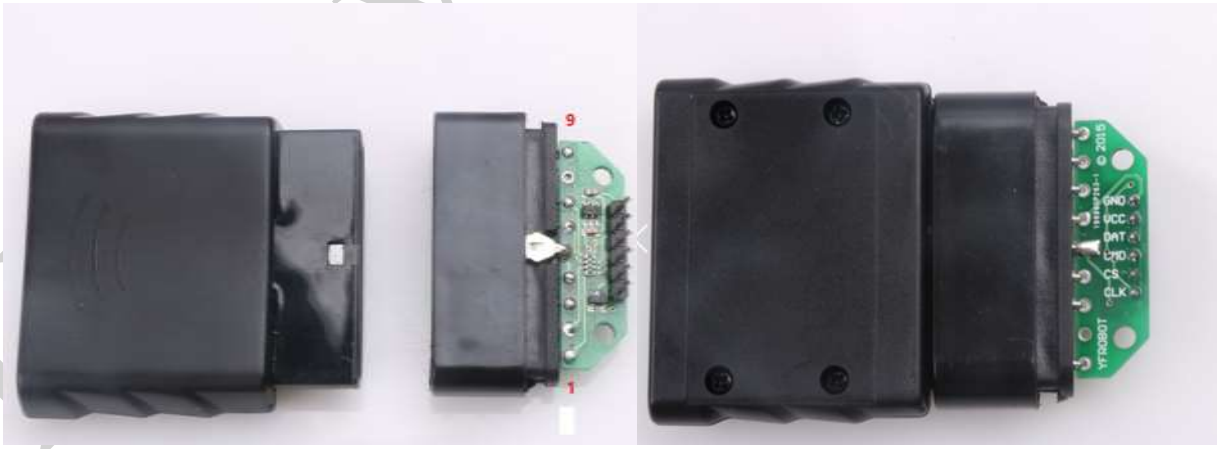


Figure 7-2

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.

7.2 PS2 Remote control test

We can firstly write a program to test the PS2 remote control, use the serial port to print, when the PS2 remote control button is pressed, the serial port prints the corresponding button key value, we observe the serial port printing content and we can see which button of the PS2 remote control is pressed..Let's write a test program firstly.

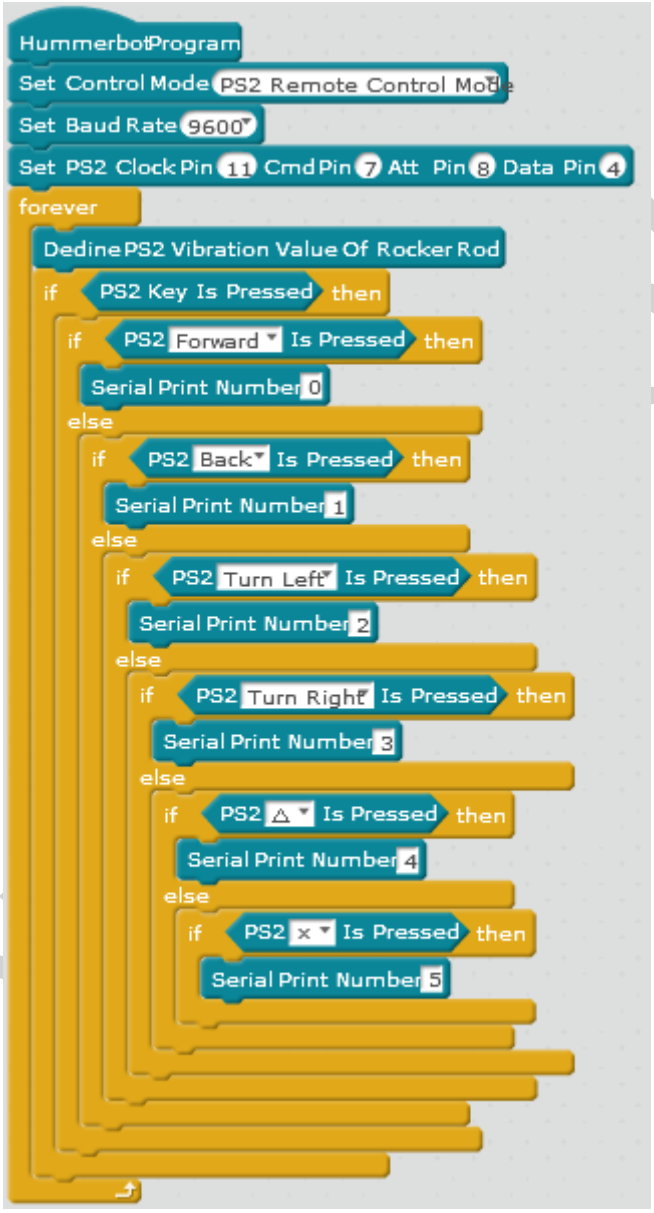


Figure 7-3

As shown in Figure 7-3 PS2 test program, after uploading the program to the robot, open the serial port monitor, press the PS2 up and down buttons, the serial port will print the corresponding number.

7.3 PS2 Remote control robot programming ideas

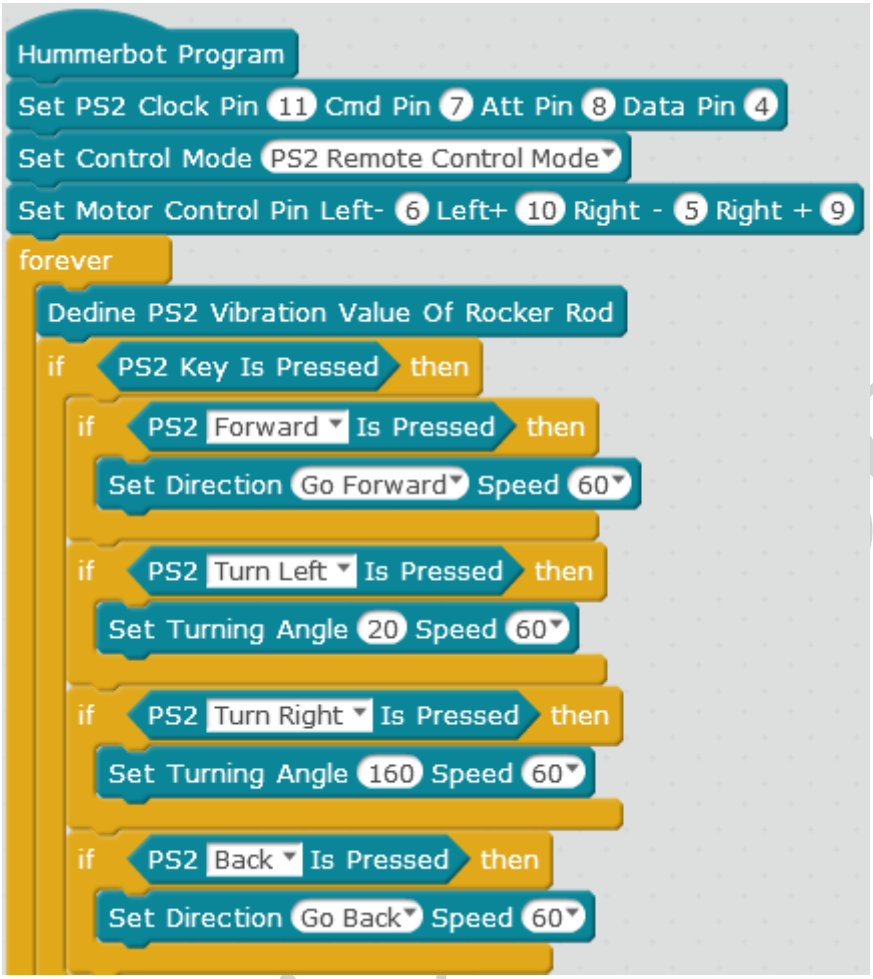
PS2 remote control robot programming idea is to set the mode to PS2 control mode, and then define the button function on the PS2 handle according to your own needs. Our next programming is programmed

according to the button definition shown in Figure 7-3. Then the robot performs corresponding leftward, forward, and rightward movement according to the received handle button value and it can also accelerate, decelerate and operate the servo etc. The PS2 handle remote control car defines all the button functions as shown in Figure 7-4

Figure 7-4

- Mark UP：Advance
- Mark DOWN：Backward
- Mark LEFT：Turn left
- Mark RIGHT：Turn right
- Mark A：Acceleration
- Mark B：Servo turns left
- Mark C：Deceleration
- Mark D：Servo turns right

Let's write the program for the tracking robot together.


Hummerbot Program
Set PS2 Clock Pin 11 Cmd Pin 7 Att Pin 8 Data Pin 4
Set Control Mode PS2 Remote Control Mode
Set Motor Control Pin Left- 6 Left+ 10 Right - 5 Right + 9
forever
Dedine PS2 Vibration Value Of Rocker Rod
if PS2 Key Is Pressed then
if PS2 Forward Is Pressed then
Set Direction Go Forward Speed 60
if PS2 Turn Left Is Pressed then
Set Turning Angle 20 Speed 60
if PS2 Turn Right Is Pressed then
Set Turning Angle 160 Speed 60
if PS2 Back Is Pressed then
Set Direction Go Back Speed 60

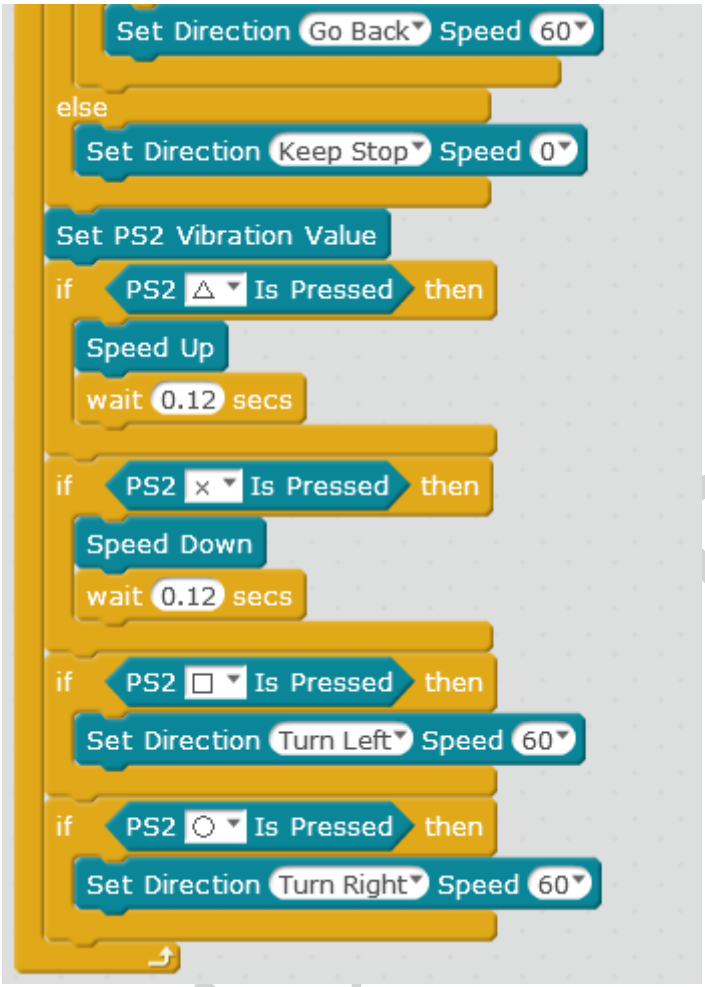


Figure 7-5

Transfer the program in Figure 7-5 to the robot. We turn on the power of the robot and connect the PS2 handle to the receiver. When we press the button on the PS2 handle, the robot will perform the corresponding action.