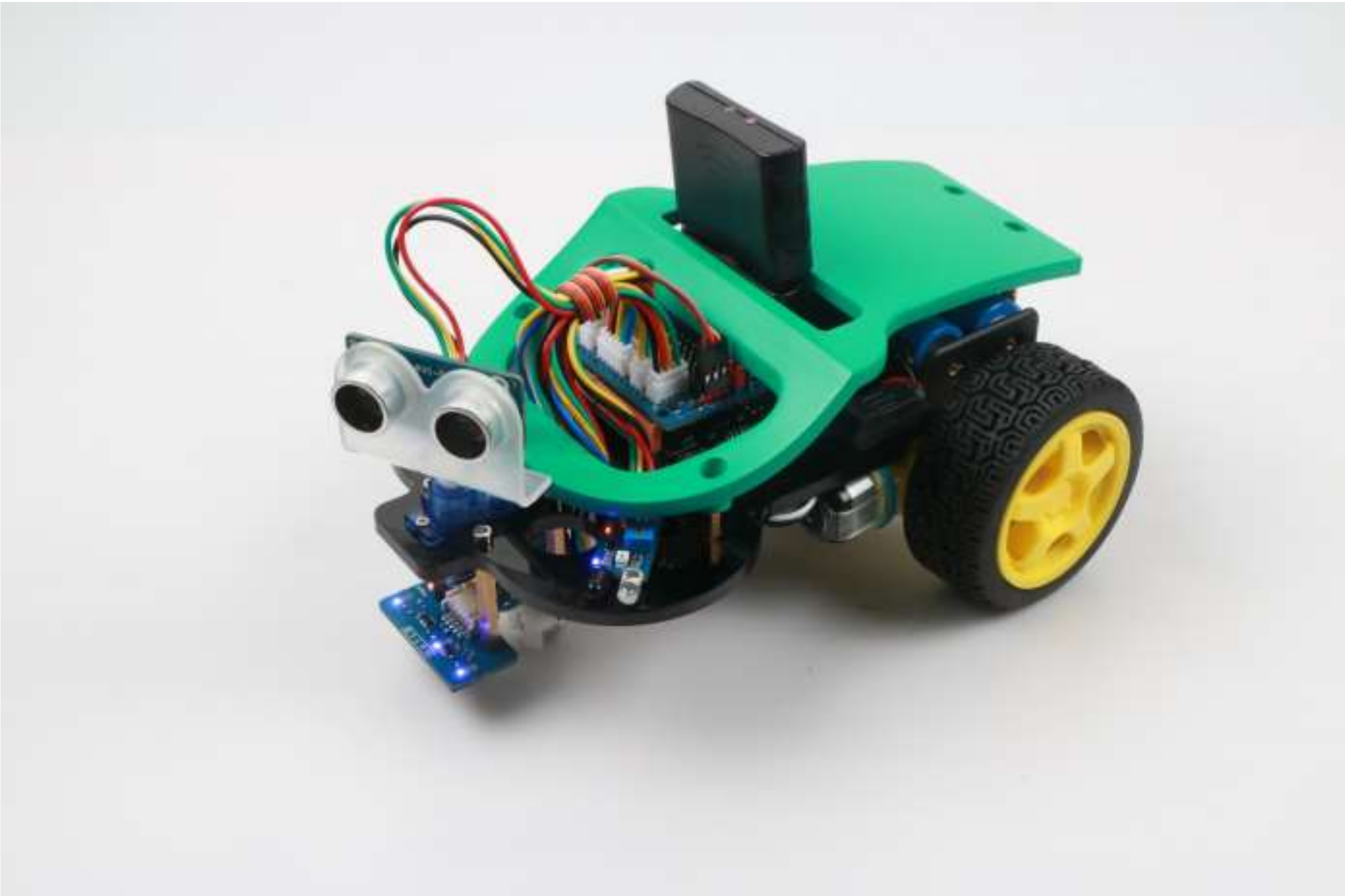
# Starter-bot MagicBlock Graphical Programming Course V.1.0



# Revision of edition

| date | version number | description | author |
|---|---|---|---|
| **2019-5-18** | **V.1.0** | **created documents** | **Twisted** |
| | | | |
| | | | |
| | | | |
| | | | |

# content

# Chapter 1 Understanding Starter-bot and Its Programming Environment

## 1.1 Introduction of life

Hello, everyone. Welcome to the world of Starter-bot robots!

What is a robot? Robots don't necessarily have to be long as human beings call them robots. Generally, robots can perform their work automatically. They can be commanded by human beings and run pre-programmed programs. Such things can be called robots.

What are the responsibilities of robots? Robot, originally Robo, means slave, that is, human servant, so the duty of robot is to help human work.

## 1.2 Summary

### 1.2.1 Starter-bot introduction

Through the study of this tutorial, we will experience the interactivity of the physical world and software. Starter-bot consists of three components: Starter-bot smart car, Arduino UNO master board and MagicBlock graphical programming software. MagicBlock is a graphical programming learning software for STEAM education, which is developed based on Scratch 2.0. It not only enables users to create interesting stories, games, animations and other works, but also supports Arduino hardware programming. We can use this software to control the Arduino UNO motherboard on Starter-bot, so that Starter-bot can issue various commands and control it. In this software, programming is no longer tedious code, using graphical representation and drag-and-drop interaction to complete the core logic of programming. The whole programming is like building an interesting building block toy. Through Magic Block's stage area or Starter-bot's behavior, we can intuitively see the actual effect of various script designs.

### 1.2.2 Starterbot Function list

1) Obstacle avoidance function of ultrasound
2) Ultrasound + Infrared Barrier Avoidance Function
3) Automatic Tracking Function
4) Infrared remote control function
5) Bluetooth remote control function
6) PS2 remote control function
7) Bluetooth Remote Mode Switching Function
8) Ligh Finding Function

# 1.3 Starter-bot Hardware introduction

## 1.3.1 Introduction to analogy

The hardware part of the Starter-bot robot we use is a Starter-bot robot, which consists of a body model built by acrylic board, various sensors, motor wheels, control panel and power supply battery, as shown in Figure 1-3-1-1.

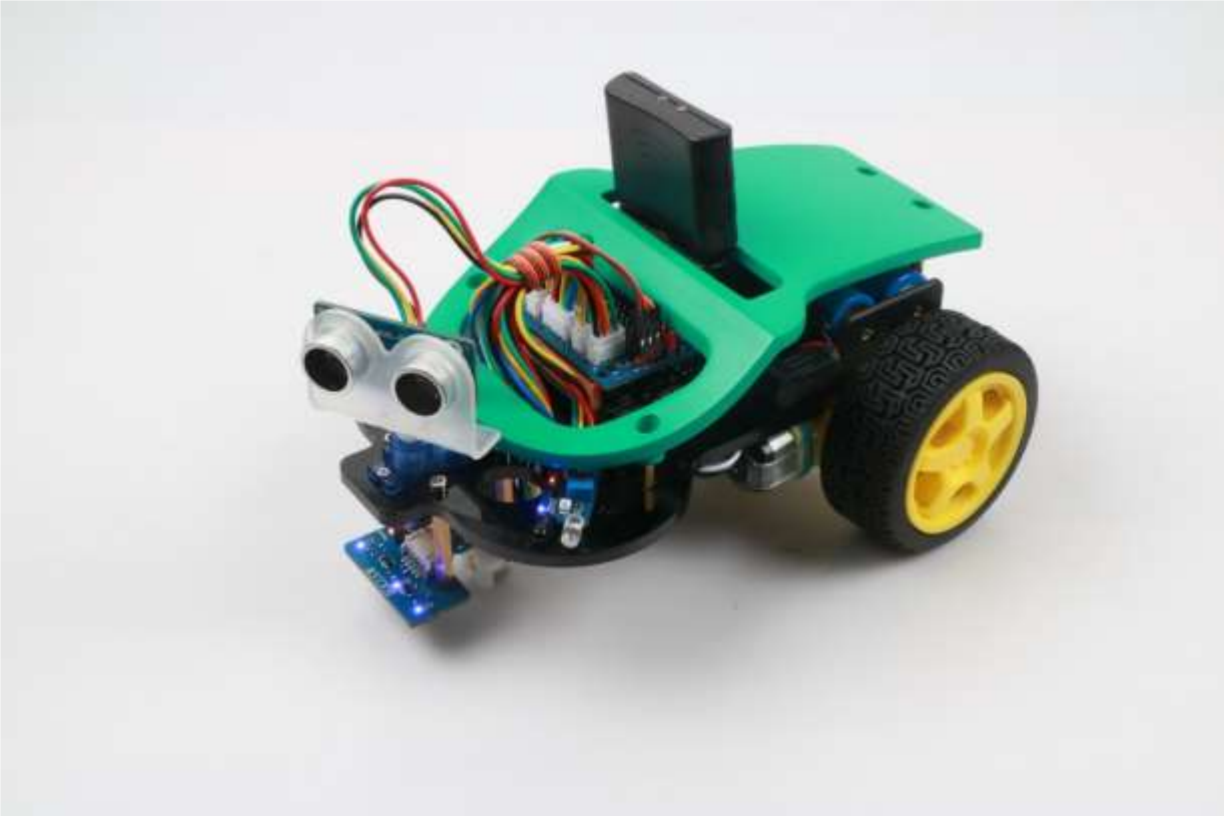


Figure 1-3-1-1

In this robot, every part of it is similar to the structure of our human body and plays a vital role. It is similar to human beings, as shown in Table 1-3-1-2.

| **Robot** | **Human** | **Function** |
|---|---|---|
| Control panel | Brain | Accept instructions, process information, and give instructions to control limbs and organs |
| Motors and wheels | limb | Responsible for limb movements |
| Battery | heart | Provide motivation |
| sensor | Facial features | Acceptance of perceived information |
| electronic circuit | nerve | Connect each part of the limb to form a whole. |

Table 1-3-1-2

## 1.3.2 Introduction of Main Control Board

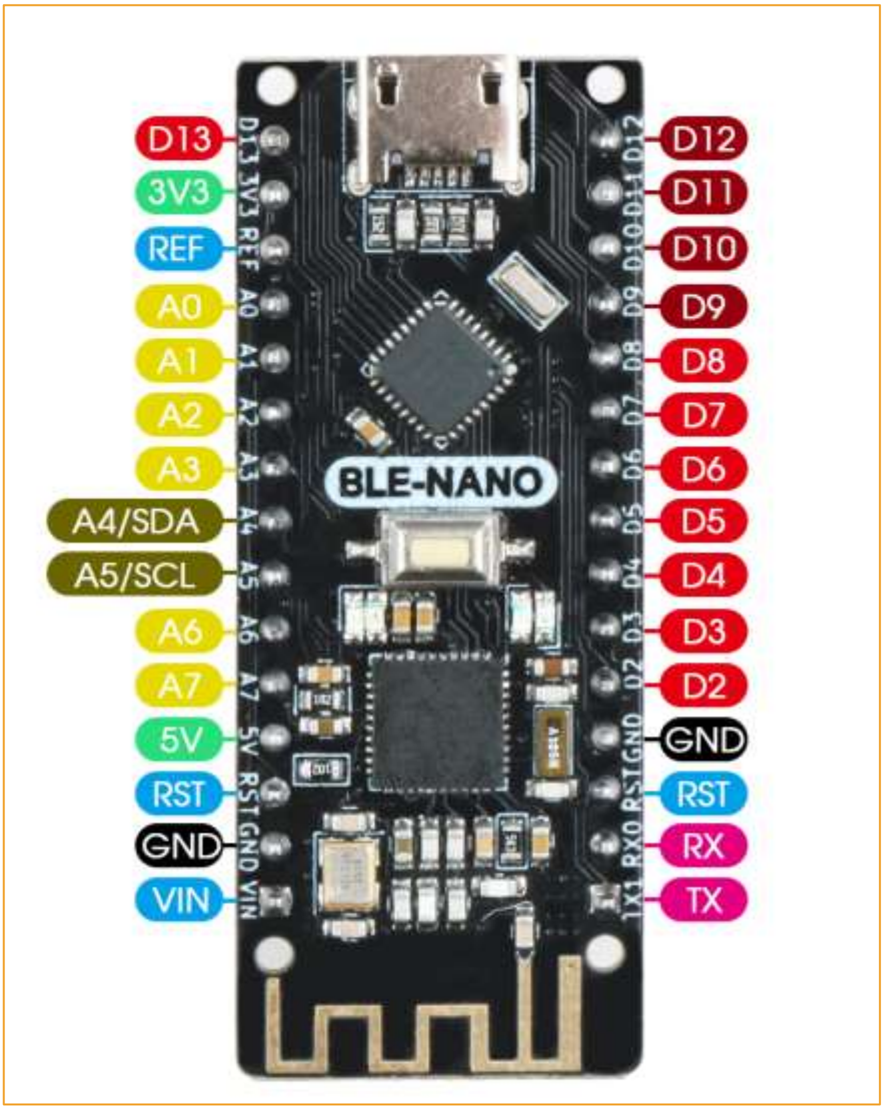



Figure 1-3-2-1

## 1.3.3 Introduction to Extension Board

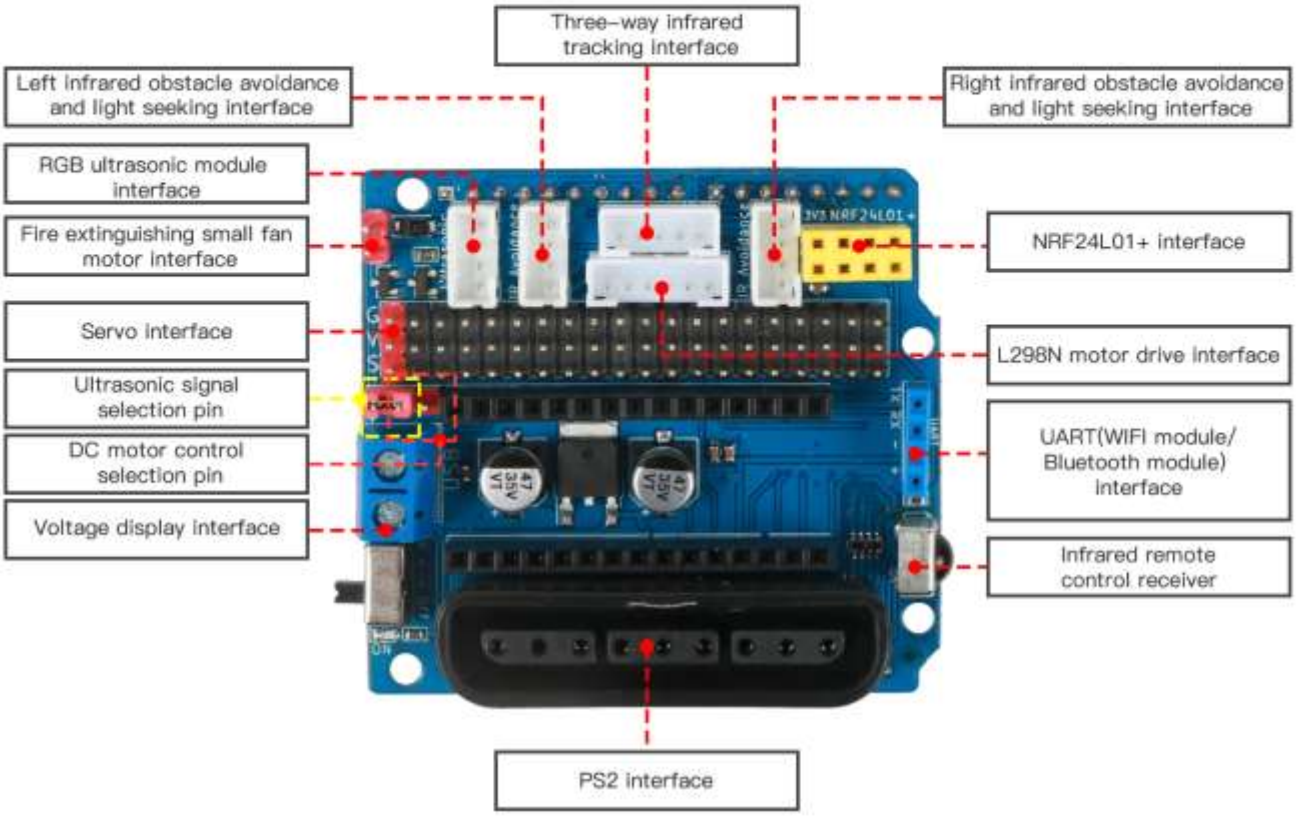



Figure 1-3-3-1

# 1.4 Starter-bot Introduction to Software Compiling Environment

## 1.4.1 Software installation

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

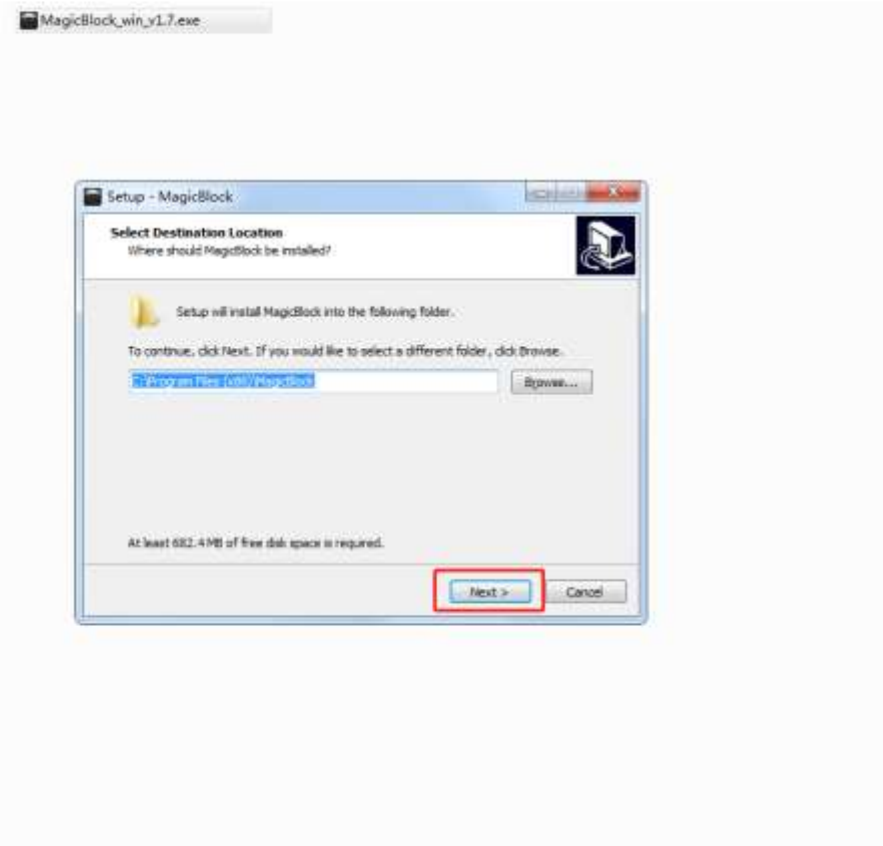



Figure 1-4-1-1

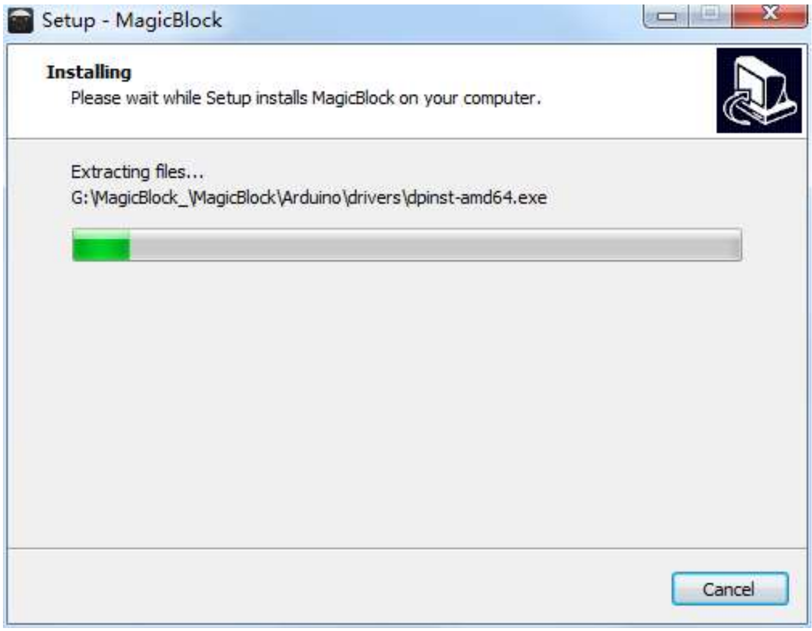



Figure 1-4-1-2

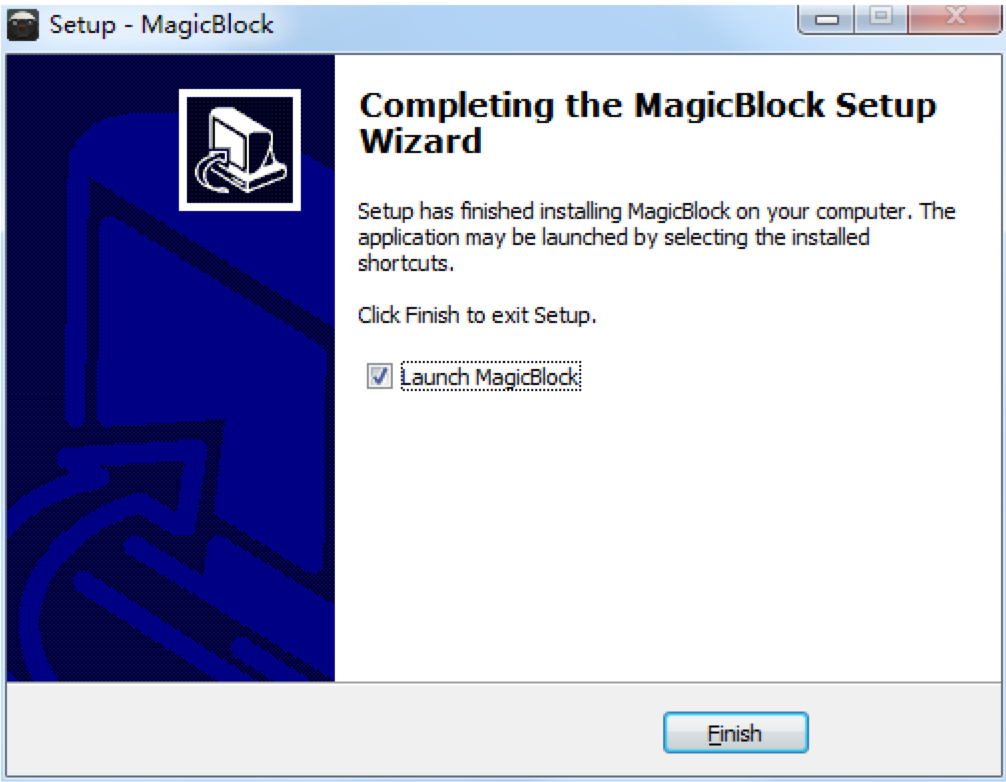



Figure 1-4-1-3

## 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. Starter-bot'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.

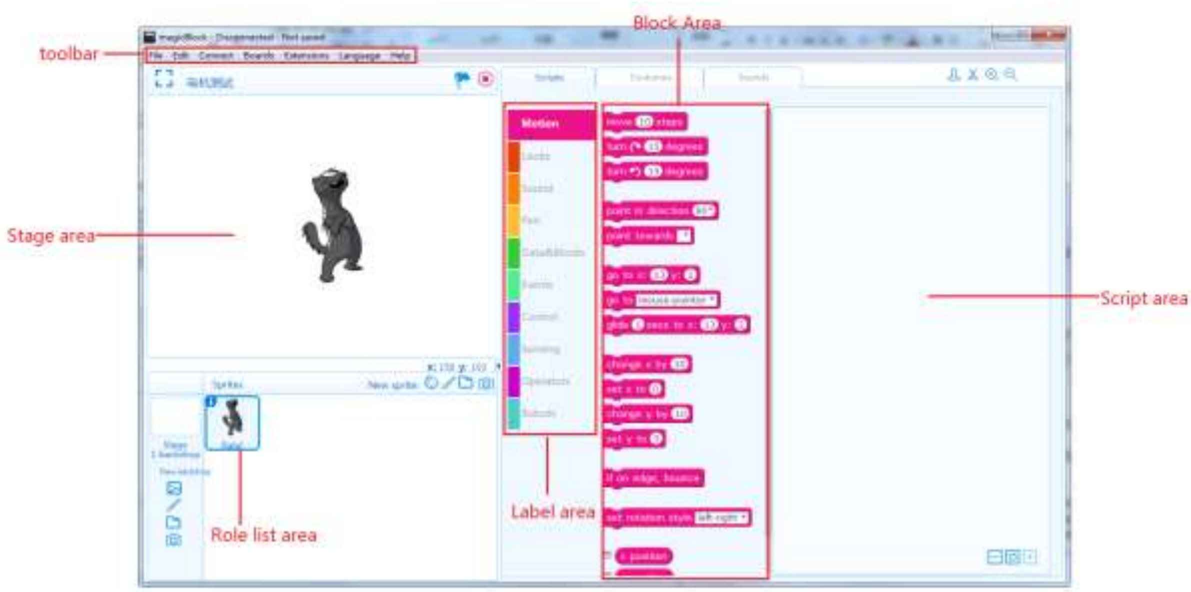



Figure 1-4-2-1

- **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 Starterbot is to control blocks under script labels.

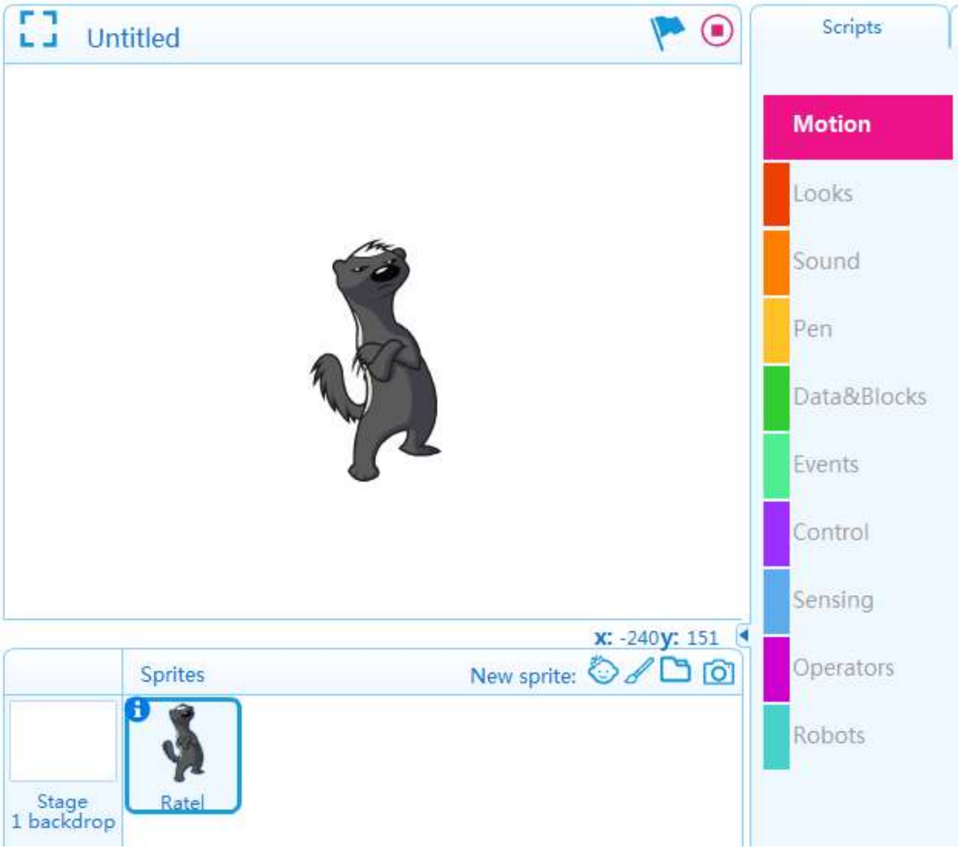

Untitled
Scripts
Motion
Looks
Sound
Pen
Data&Blocks
Events
Control
Sensing
Operators
Robots
x: -240 y: 151
Sprites
New sprite:
Stage
1 backdrop
Ratel

Figure 1-5-1

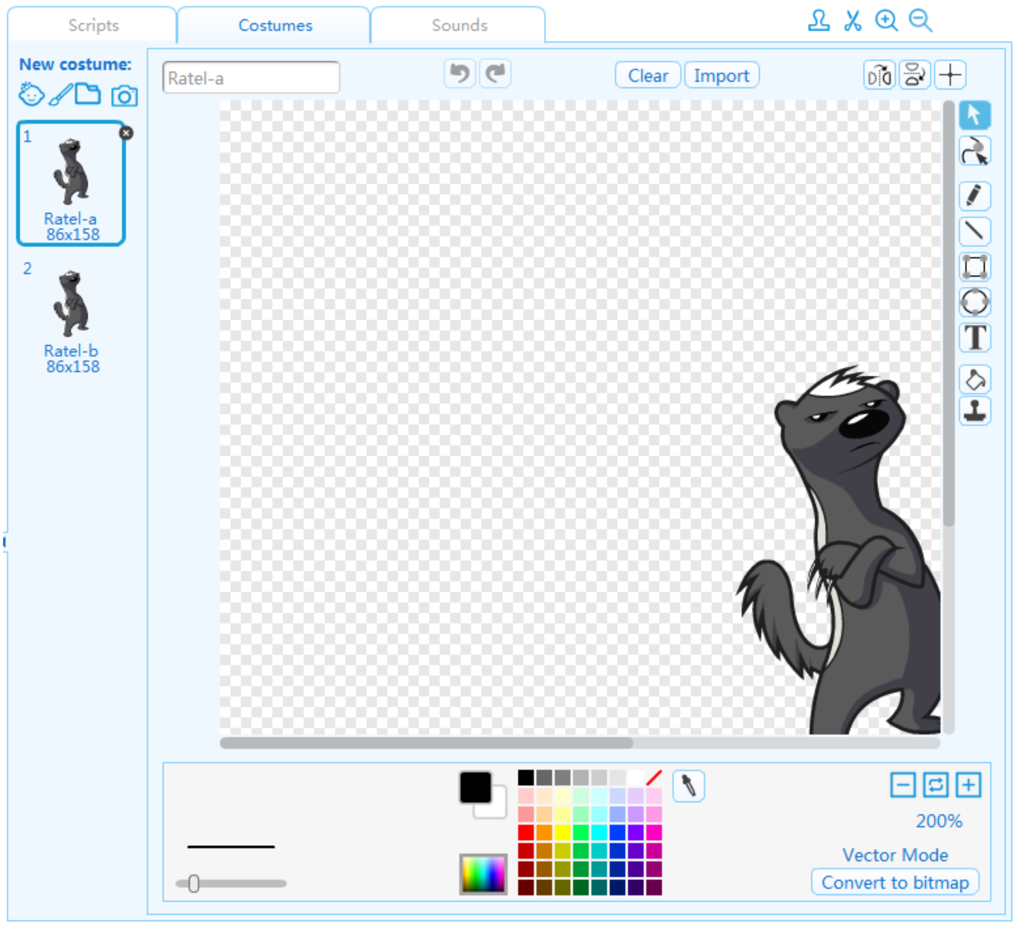

Scripts
Costumes
Sounds
New costume:
Ratel-a
Clear
Import
1
Ratel-a
86x158
2
Ratel-b
86x158
200%
Vector Mode
Convert to bitmap

Figure 1-5-2

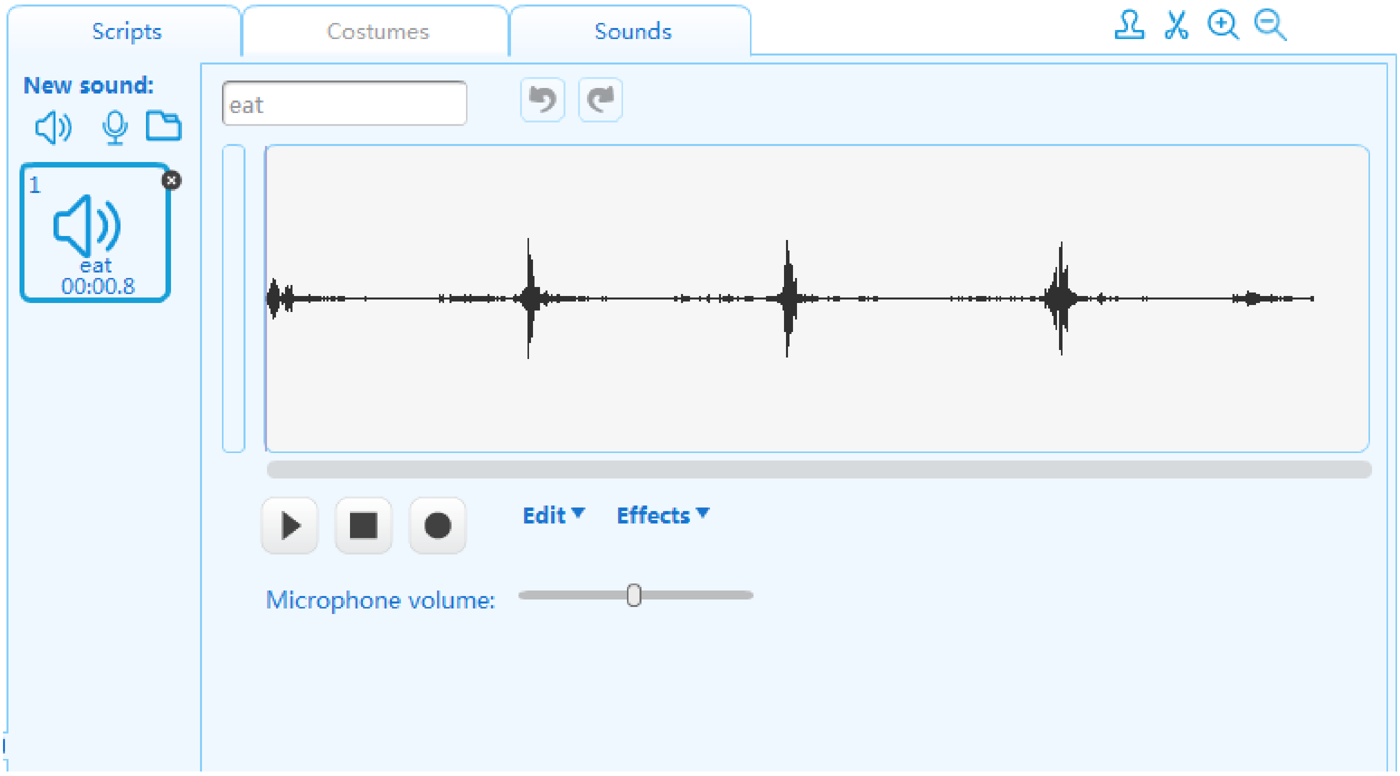



Figure 1-5-3

## 1.6 Starterbot 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.

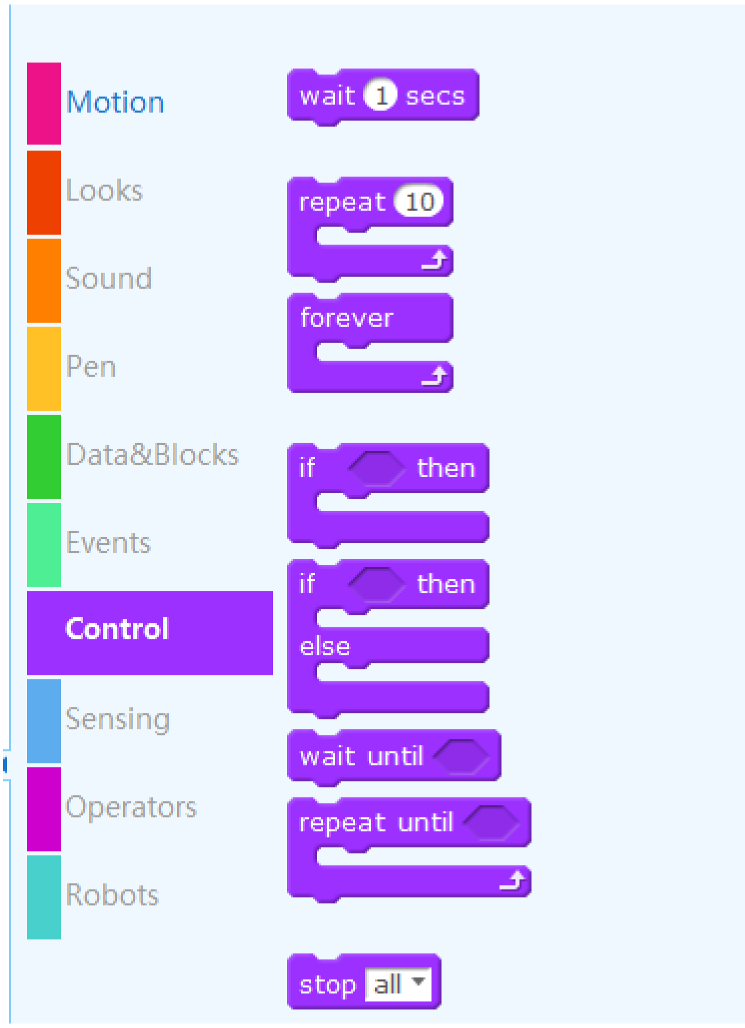
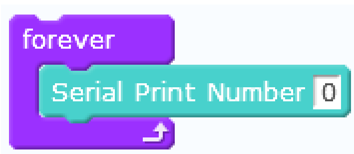



Figure 1-6-1

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.

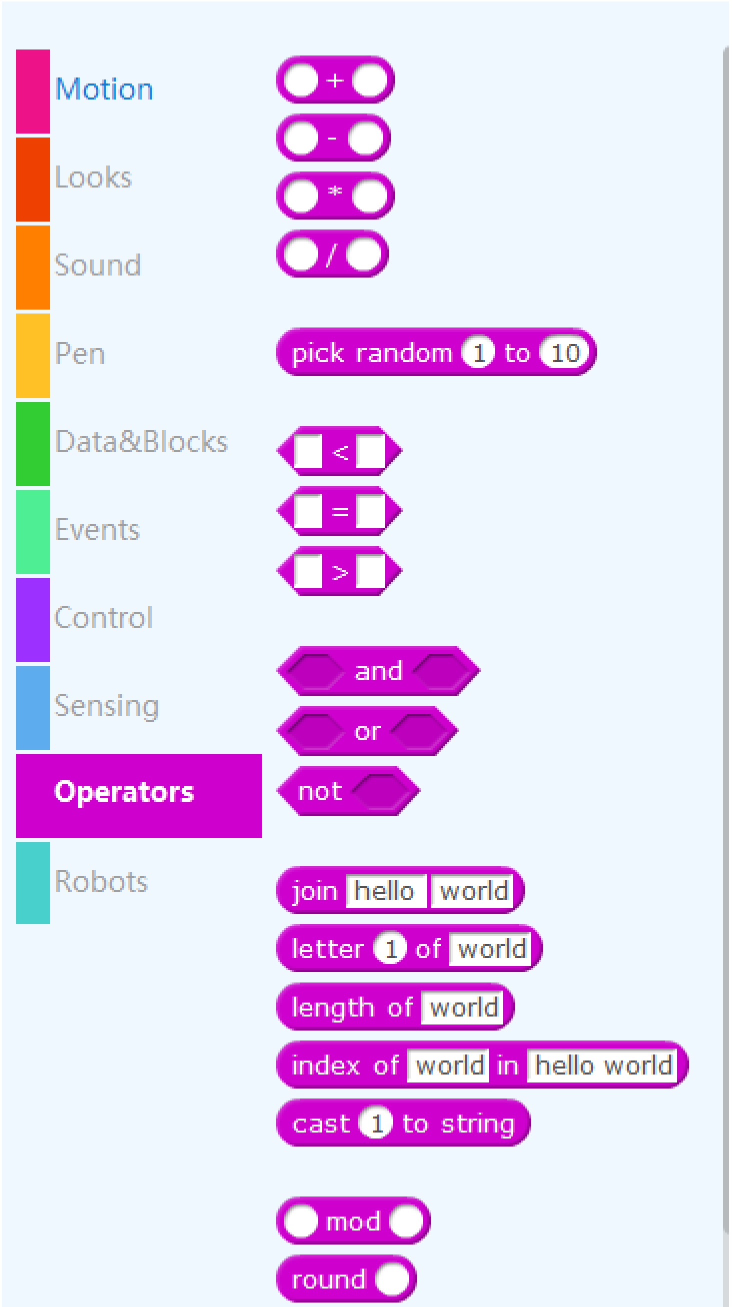
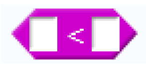
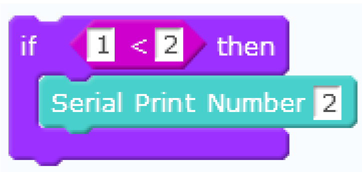


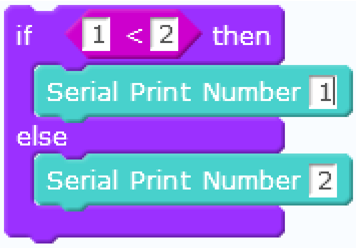

Figure 1-6-2

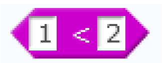
For example， 

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.

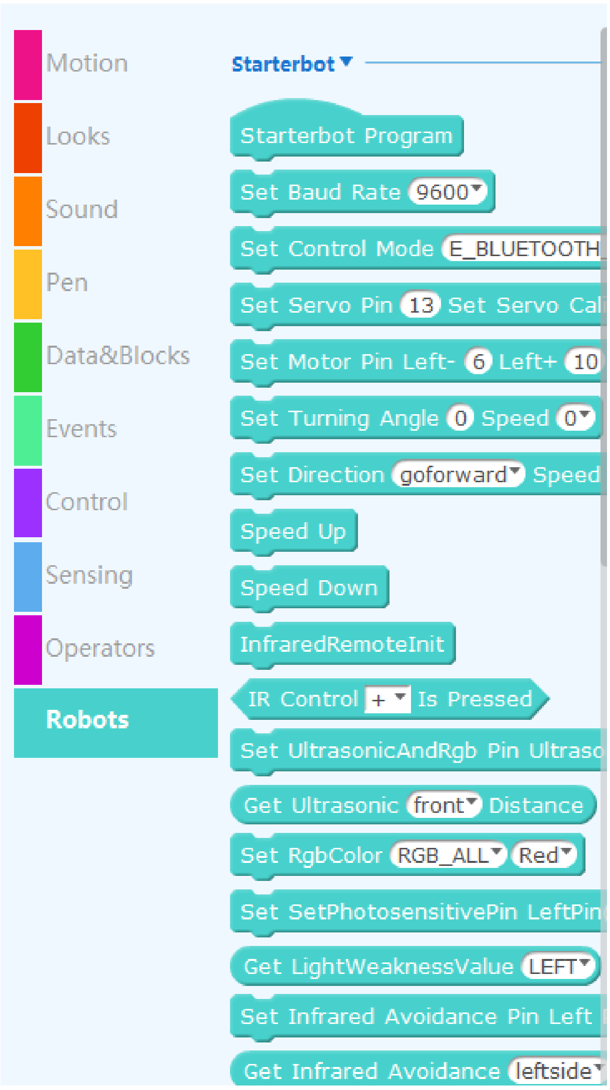



Figure 1-6-3

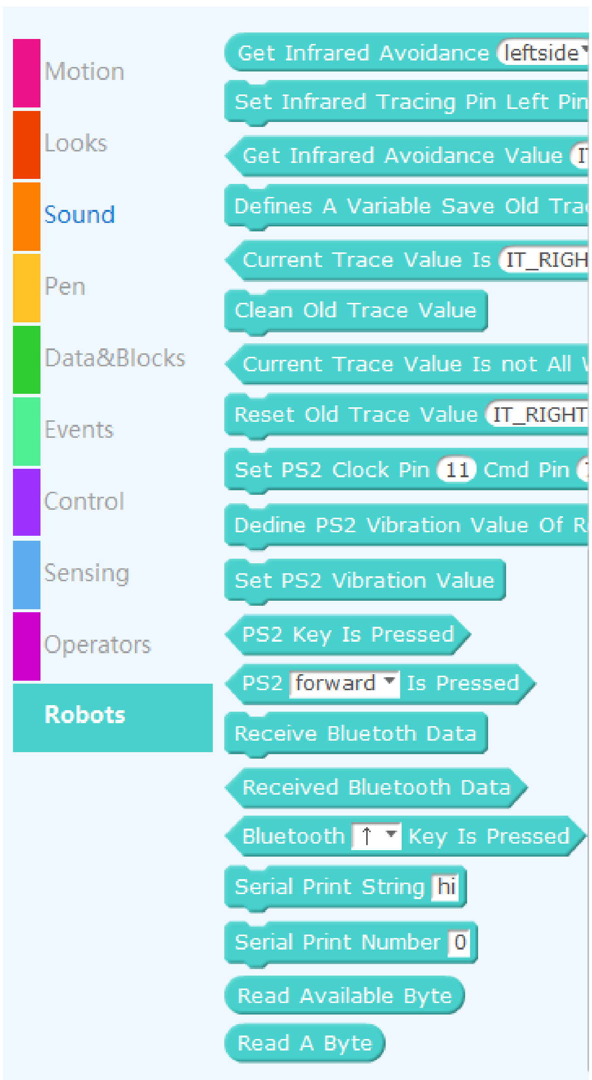



Figure 1-6-4

# Chapter 2 Initial Knowledge Programming

## 2.1 Hello world

If we want the Starter-bot robot to move, we need to store instructions (programs) in its brain (control board) beforehand for the Starter-bot robot. How to compile instructions for the Starter-bot robot? Now let's take you through the experience of writing a Starter-bot robot print Hello World program.

### 2.1.1 Add Starterbot Library

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

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

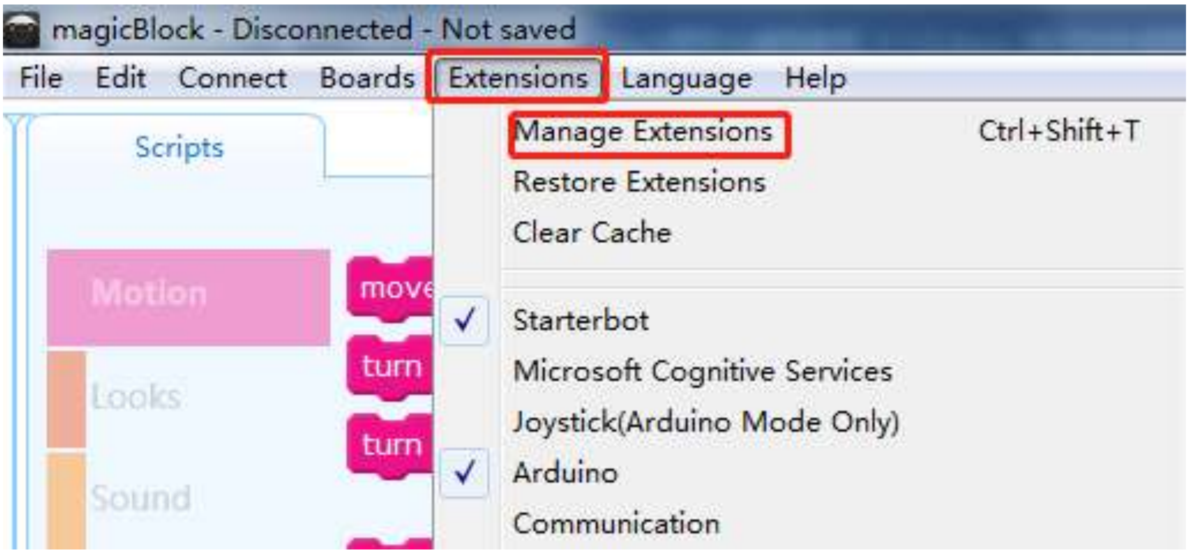



Figure 2-1-1-1

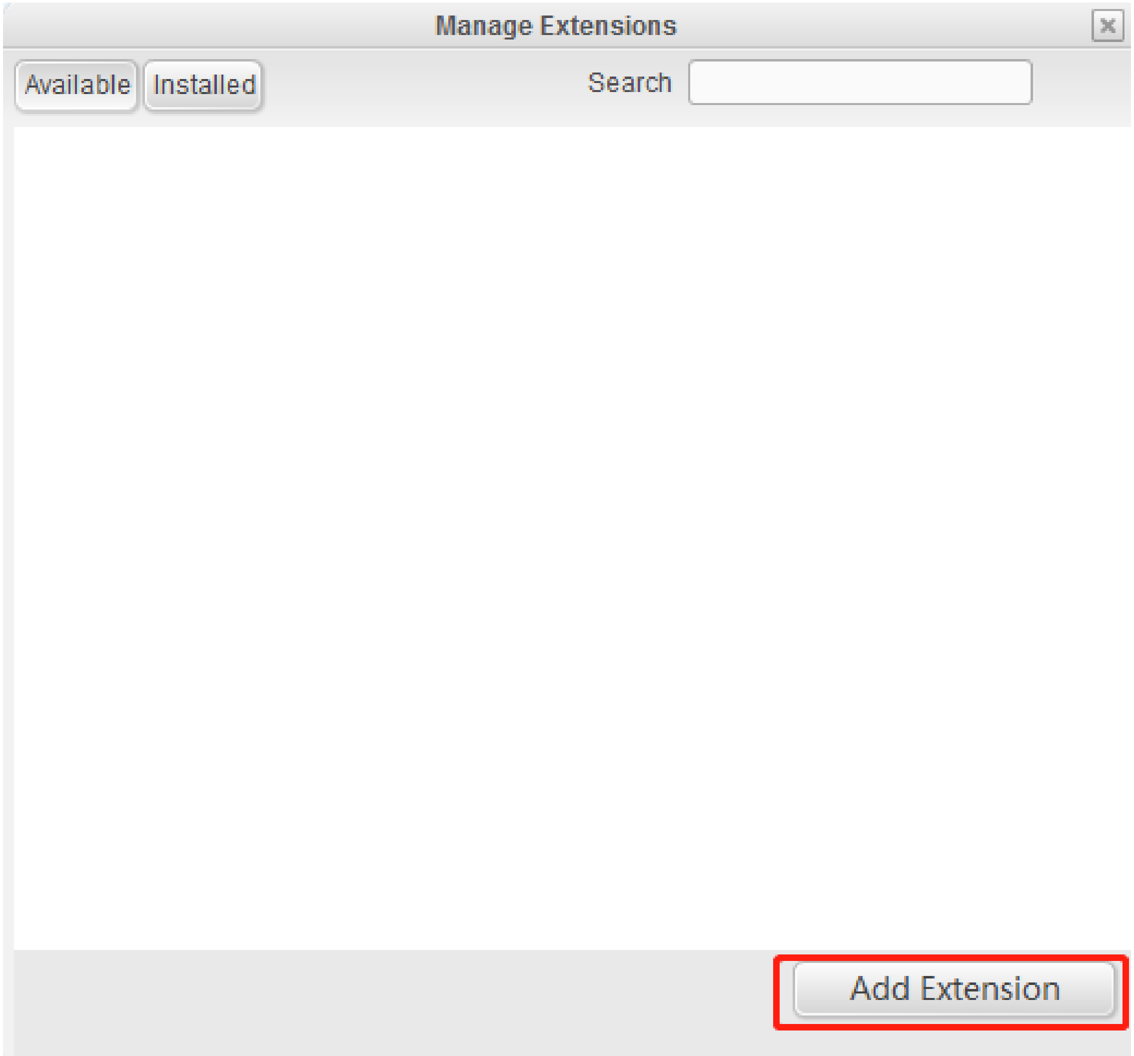



Figure 2-1-1-2

4) Select "Zip. file" for the file type, then select "Starterbot. zip" and click "Open", as shown in Figure 2-1-1-3.

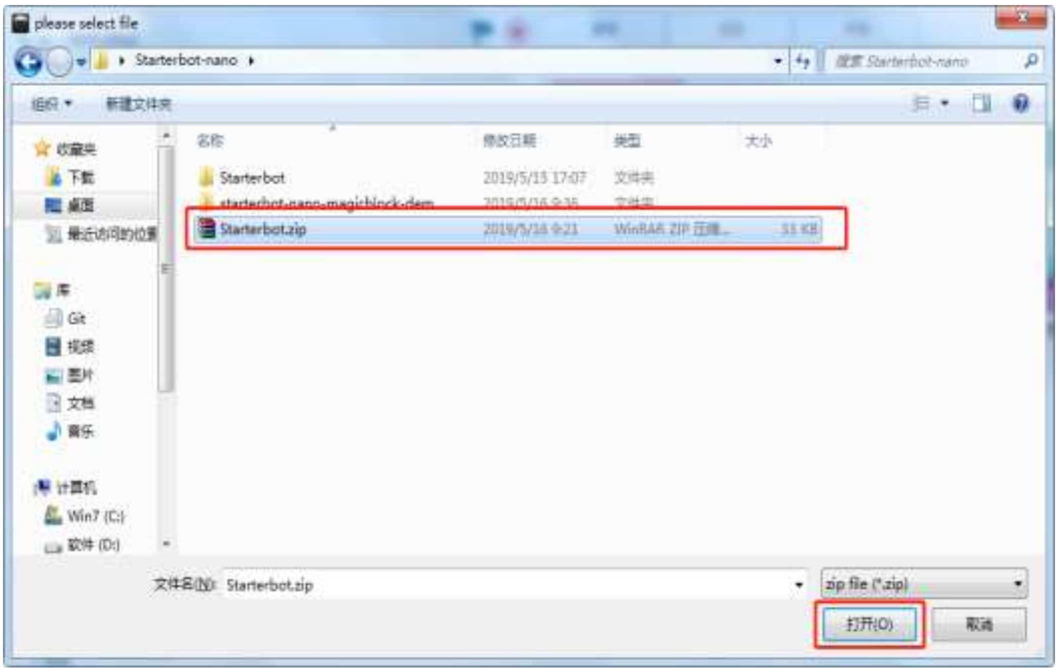


Figure 2-1-1-3

5) In "Extension Manager", click "Installed", and you will see that the Starterbot library has been successfully added, as shown in Figure 2-1-1-4;

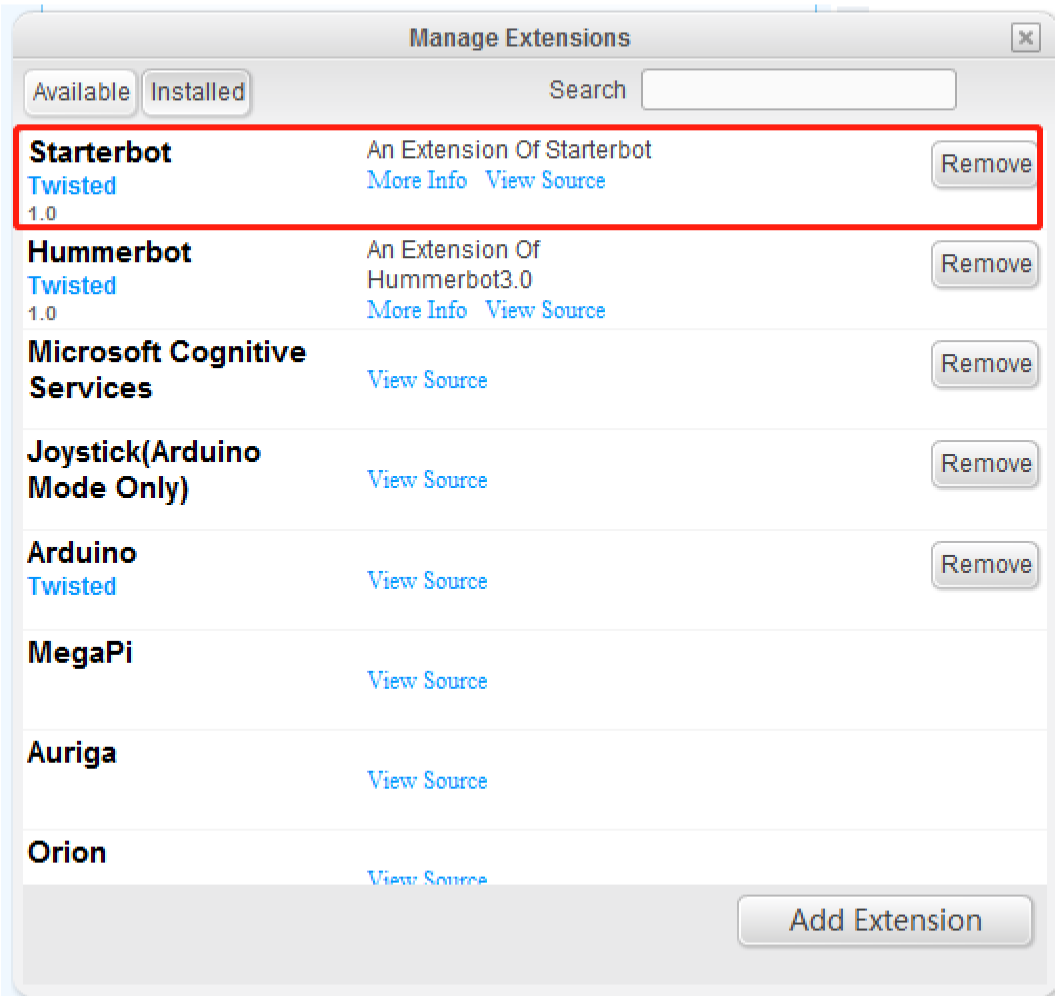



Figure 2-1-1-4

6) Click on "Extension", select "Starterbot", and then click "Script -> Robot Module", the Starterbot Block Graphics Programming Block will be displayed in the building block area, as shown in Figure 2-1-5.

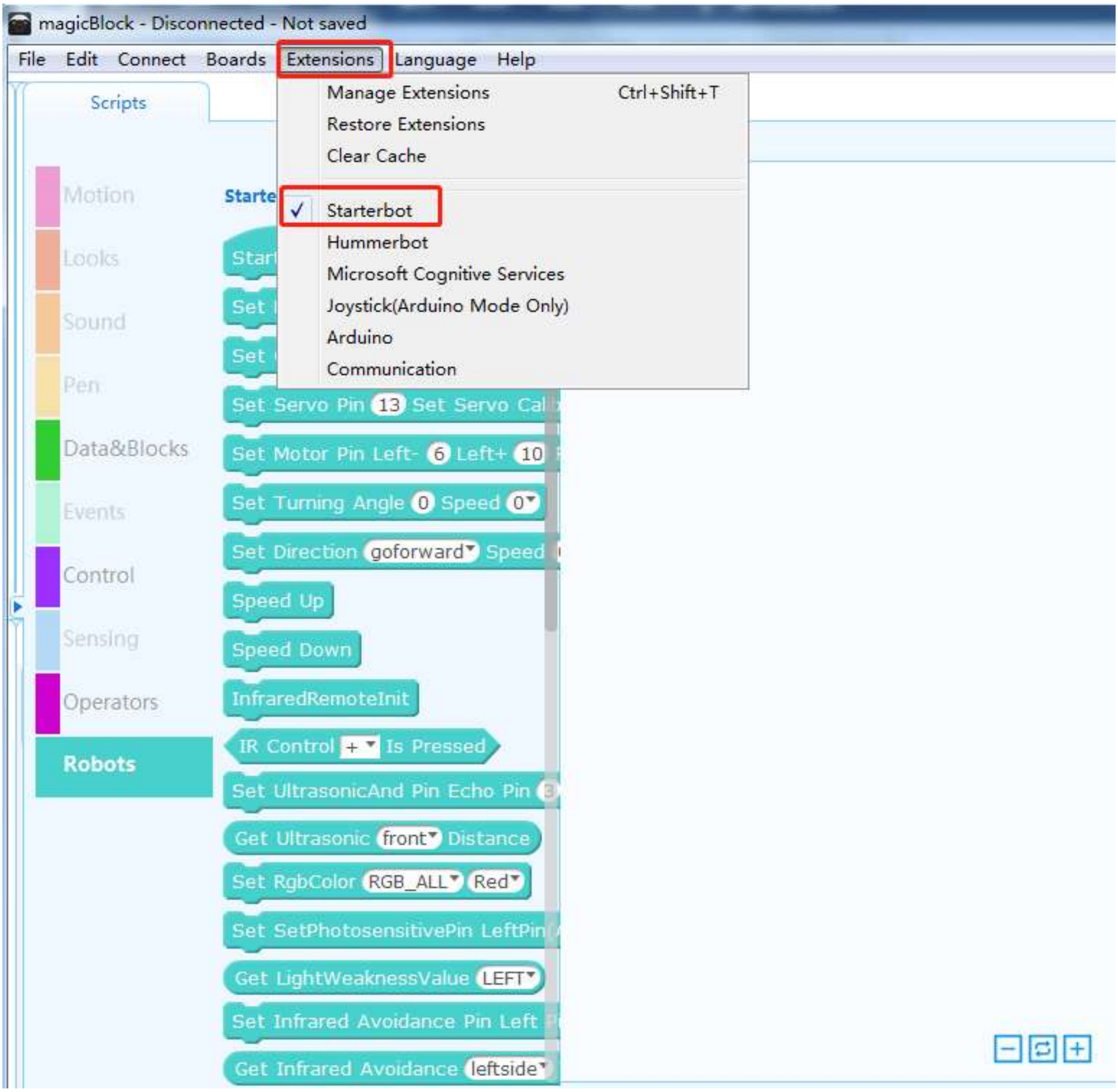



Figure 2-1-1-5

## 2.1.2 First Experience of Programming

After learning about Magic Block's graphics blocks, let's experience Starterbot programming. Let's first write a program for Starterbot robots to print Hello world.

First, drag the Starterbot 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 "Starterbot Main Program" building block, as shown in Figure 2-1-2-1.

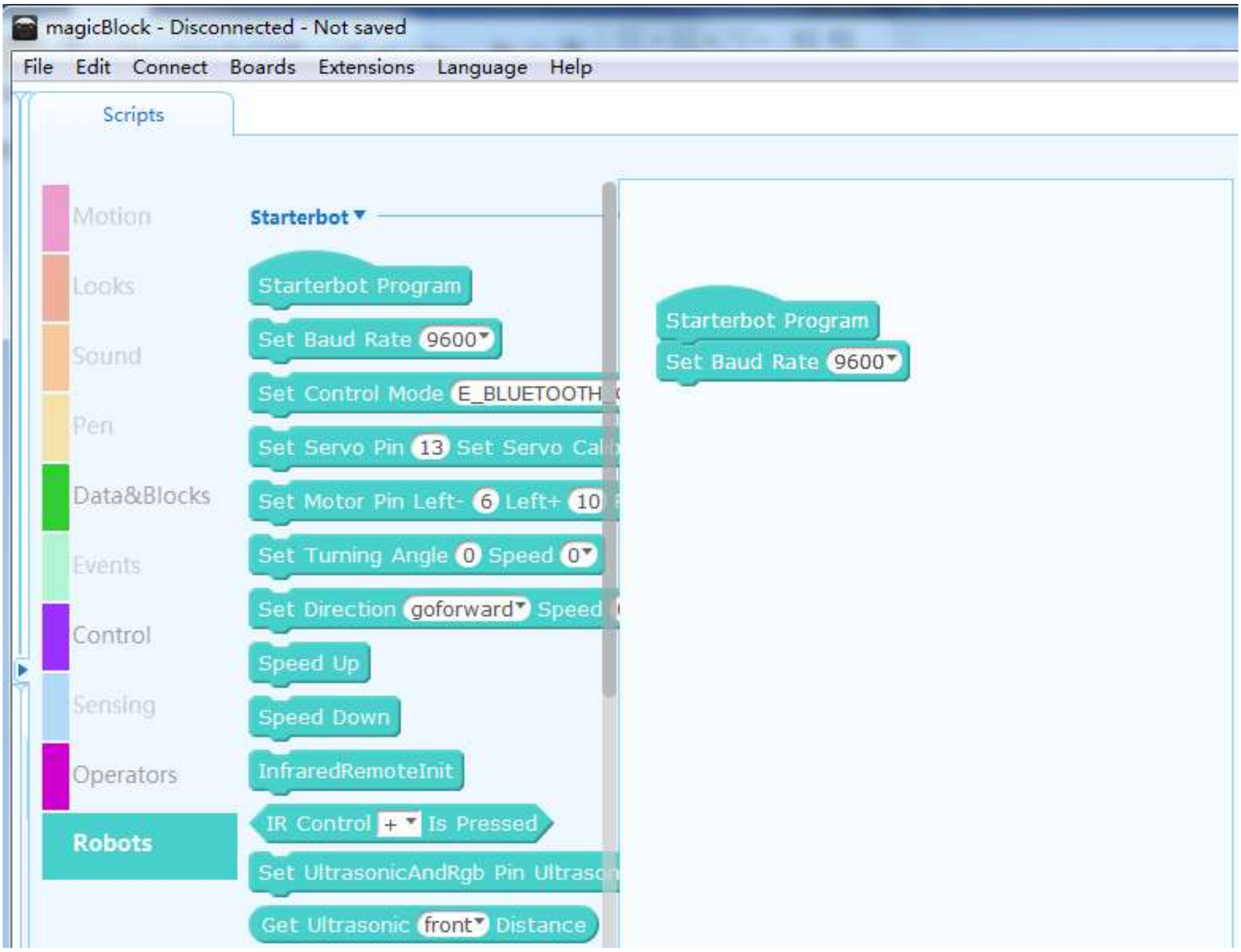



Figure 2-1-2-1

1) Click on the "Control" tab in the label area and drag the "Repeated Execution" block under the Starterbot main program in the script area.

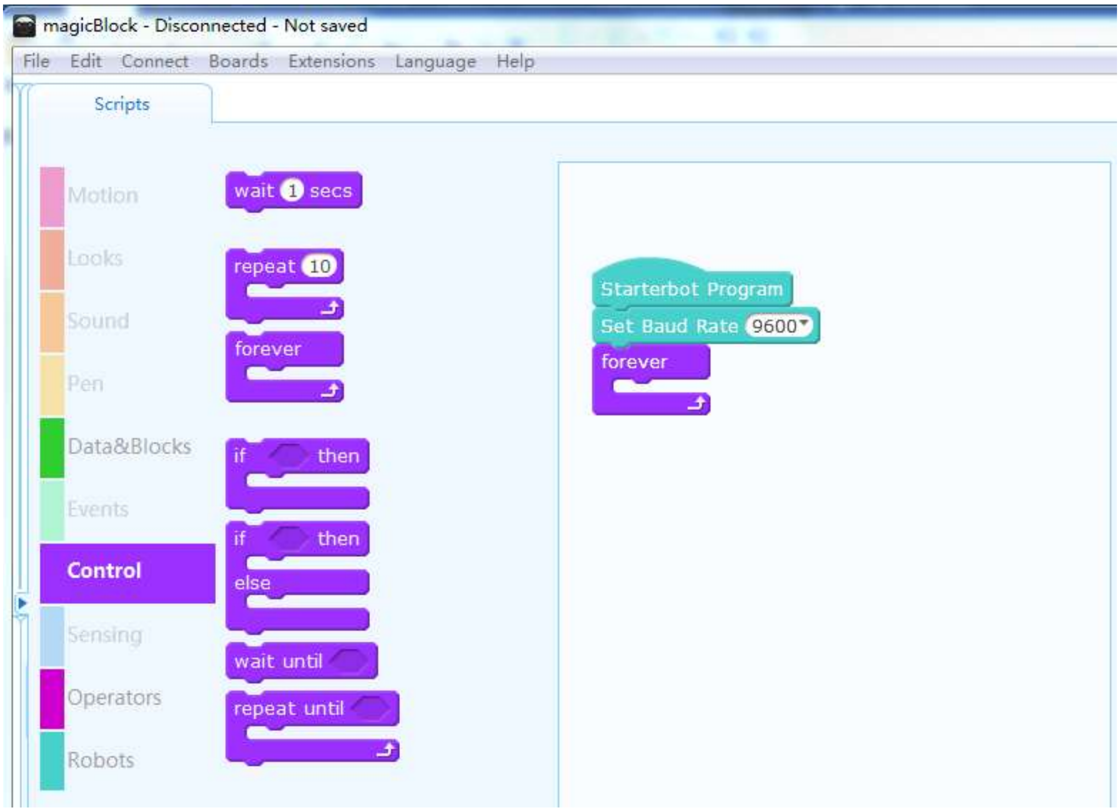



Figure 2-1-2-2

2) Drag the "serial port print string" building block into the repeated execution building block and enter "Hello world" as shown in Figure 2-1-2-3.

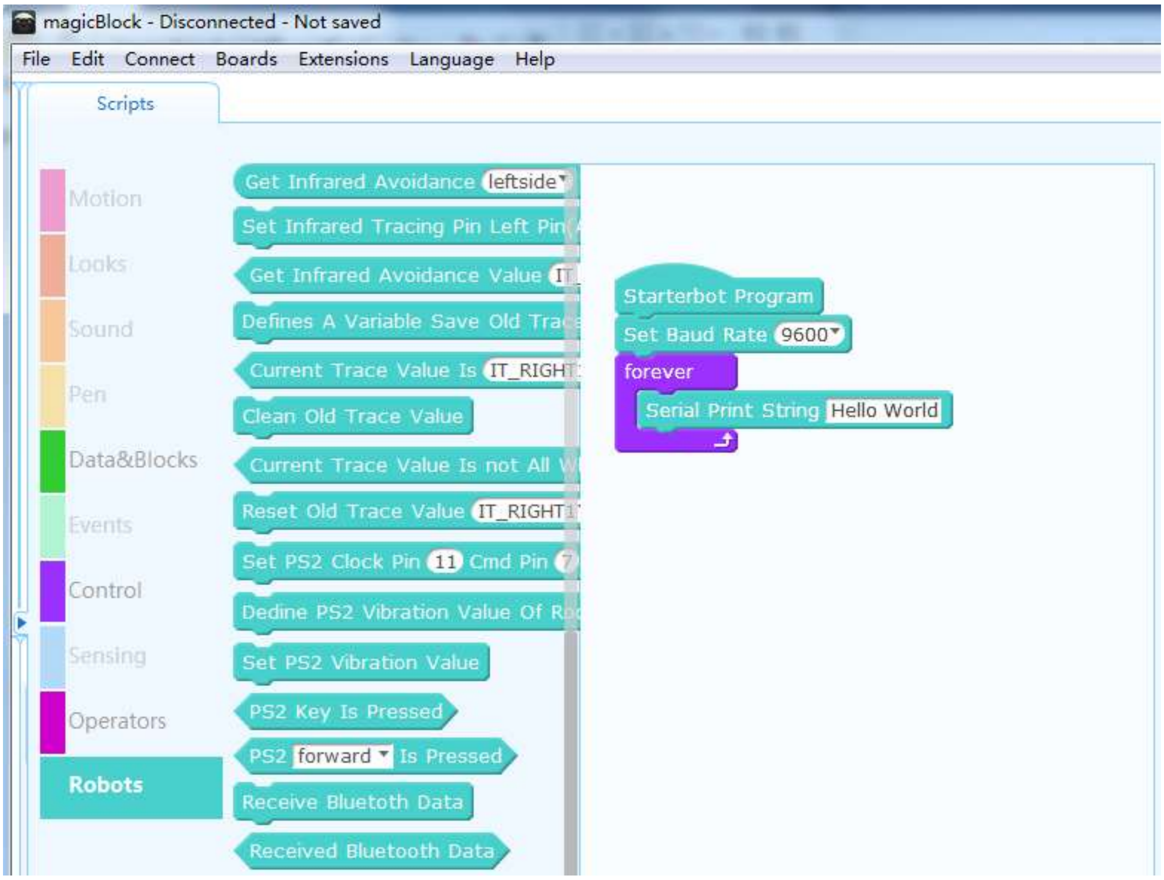


Figure 2-1-2-3

This is the Starterbot Robot Printing Hello World program. After the program is written, we need to transfer the program to the brain of the Starterbot robot (control motherboard) to do the desired action according to the program we wrote. How to transfer the program to the brain of the Starterbot 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 Starterbot robot (control motherboard). Here is the connection method between MagicBlock and the robot motherboard.

## 2.2 MagicBlock and Starter-bot Connection Steps

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 driver of Starter-bot Master Control Board, click "Connect to Install Arduino Driver" as shown in Figure 2-2-1, and click "Install" as shown in Figure 2-2-1.

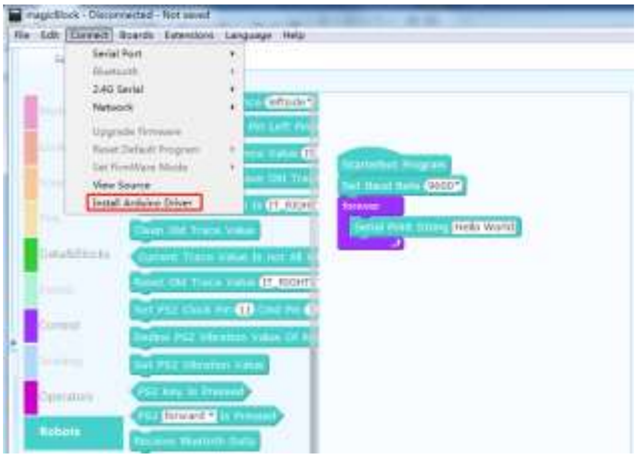


Figure 2-2-1

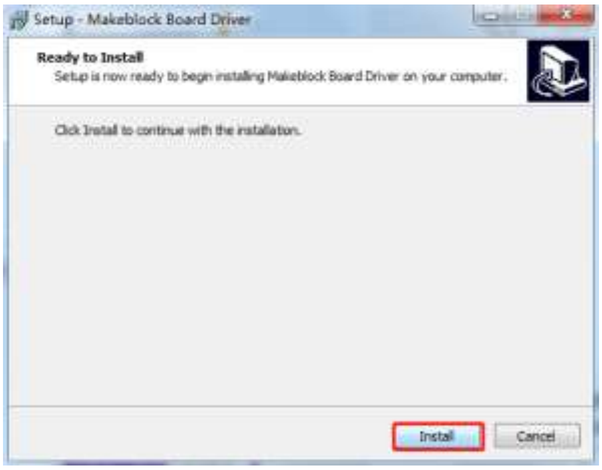



Figure 2-2-2

3) Click on "Connect to COM4 (different computers have different numbers of COM ports)", as shown in Figure 2-2-3. After the correct connection, there will be a "Serial Port Connected" prompt at the top of the software. At this time, MagicBlock and Starter-bot are successfully connected, as shown in Figure 2-2-4.

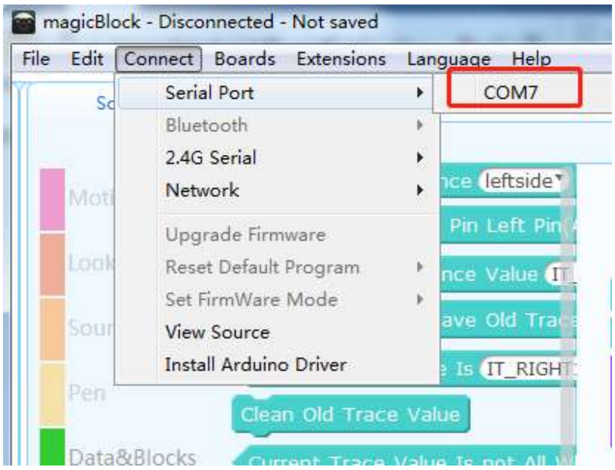



Figure 2-2-3

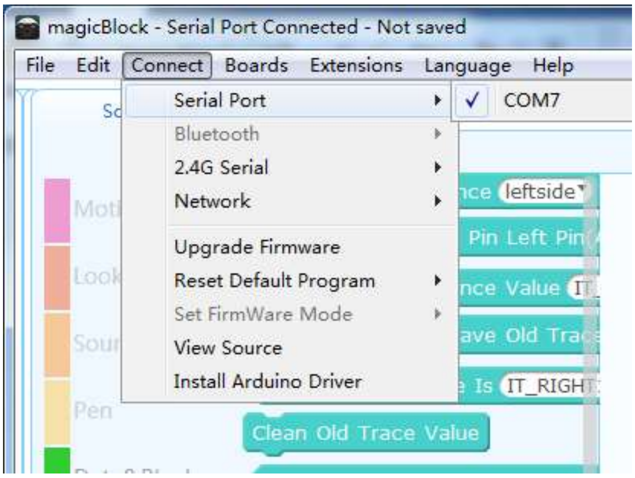



Figure 2-2-4

## 2.3 Upload program to Starter-bot robot

When the program is finished and MagicBlock and Starterbot robot are connected correctly, we can transmit the program to the brain of Starterbot robot (main control board). The specific steps are as follows:

1) Select the type of control board for program transmission, select "Arduino Uno" as shown in Figure 2-3-1, and select "Edit Arduino Mode" as shown in Figure 2-3-2.

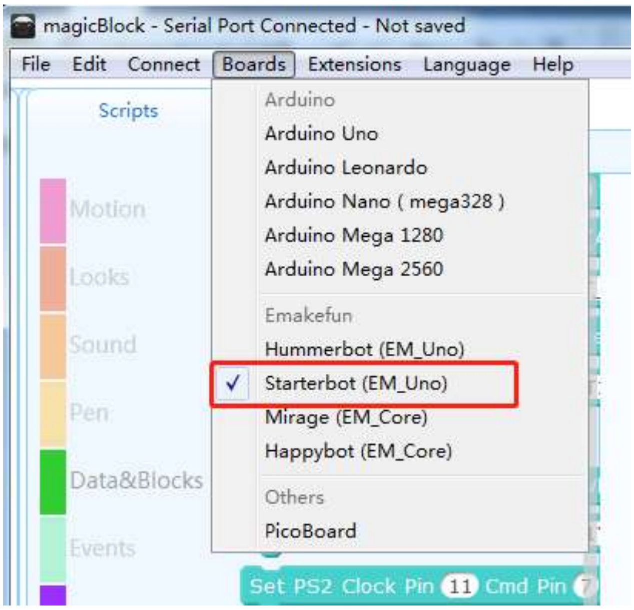



Figure 2-3-1

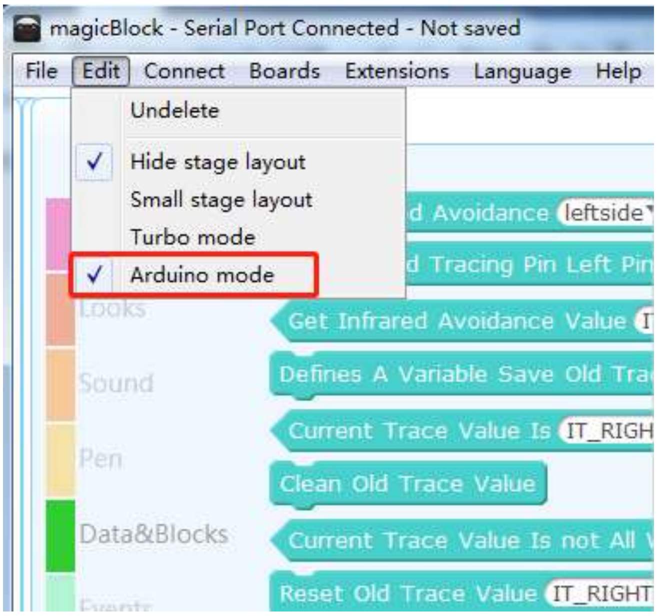



Figure 2-3-2

2) Click on any building blocks in the script area with the mouse. On the right side of the building blocks area, the corresponding building blocks program code will appear. Click "Upload to Arduino", start to generate the building blocks offline code and upload it to the Starterbot Robot. The screen will display a prompt window "Upload in", as shown in Figure 2-3-3; after the upload is completed, the prompt "Upload completed" will appear as shown in Figure 2-3-4.


Figure 2-15

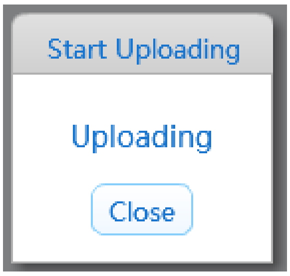



Figure 2-3-3

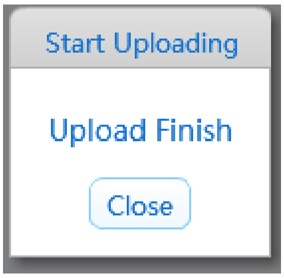



Figure 2-3-4

After completing the above steps, the Starterbot robot's brain (master board) already has the program we wrote, so how can we see the Hello Word printed by the robot? At this point, we will use a serial monitor to see, click the "Edit with Arduino IDE" in the upper right corner, as shown in Figure 2-3-5.

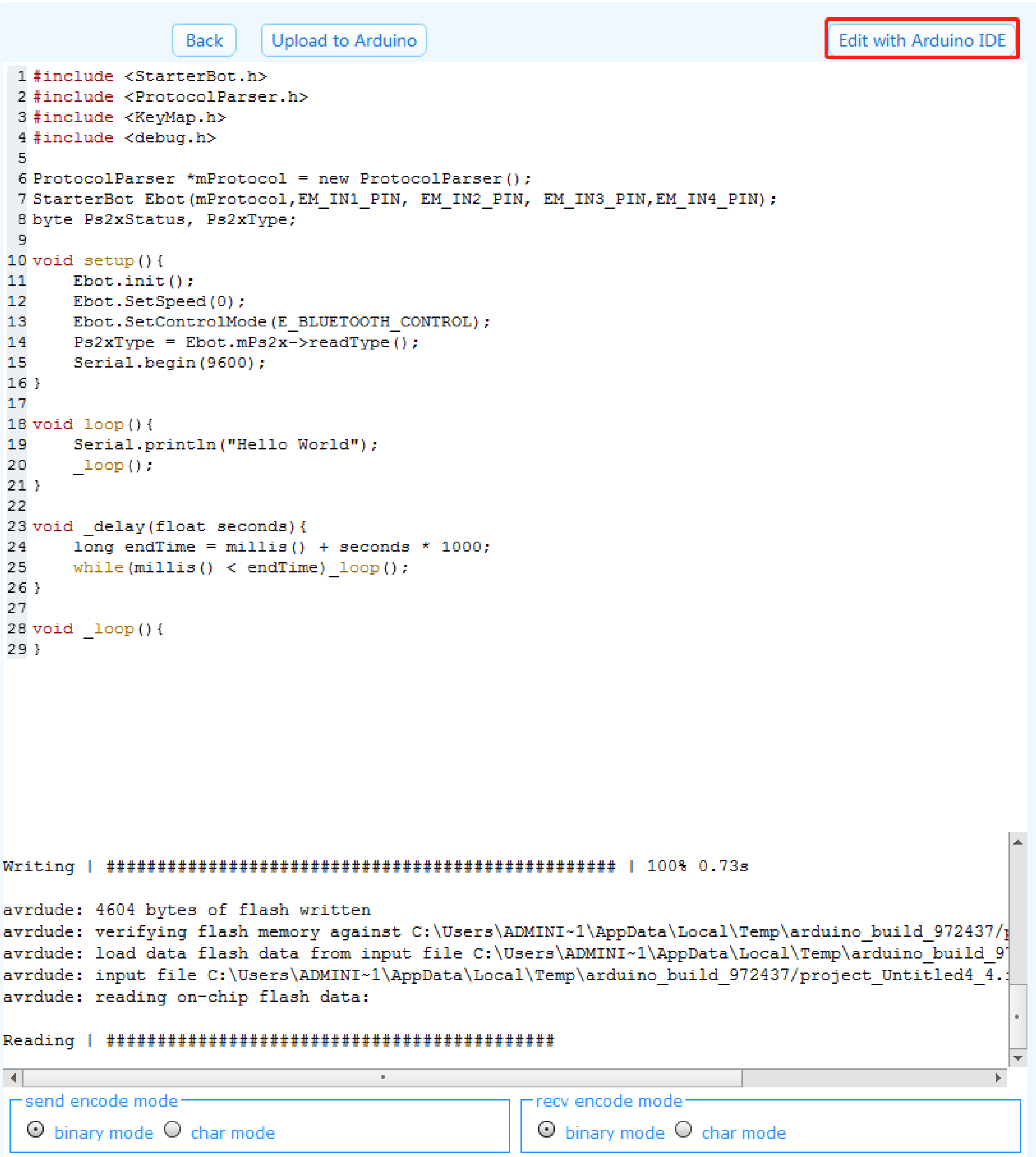



Figure 2-3-5

After opening the Auduino software, click on the serial monitor in the upper right corner, as shown in Figure 2-3-6.

Figure 2-3-6

Turn on the serial monitor and we will see Hello Word printing continuously on the serial monitor, as shown in Figure 2-3-7.

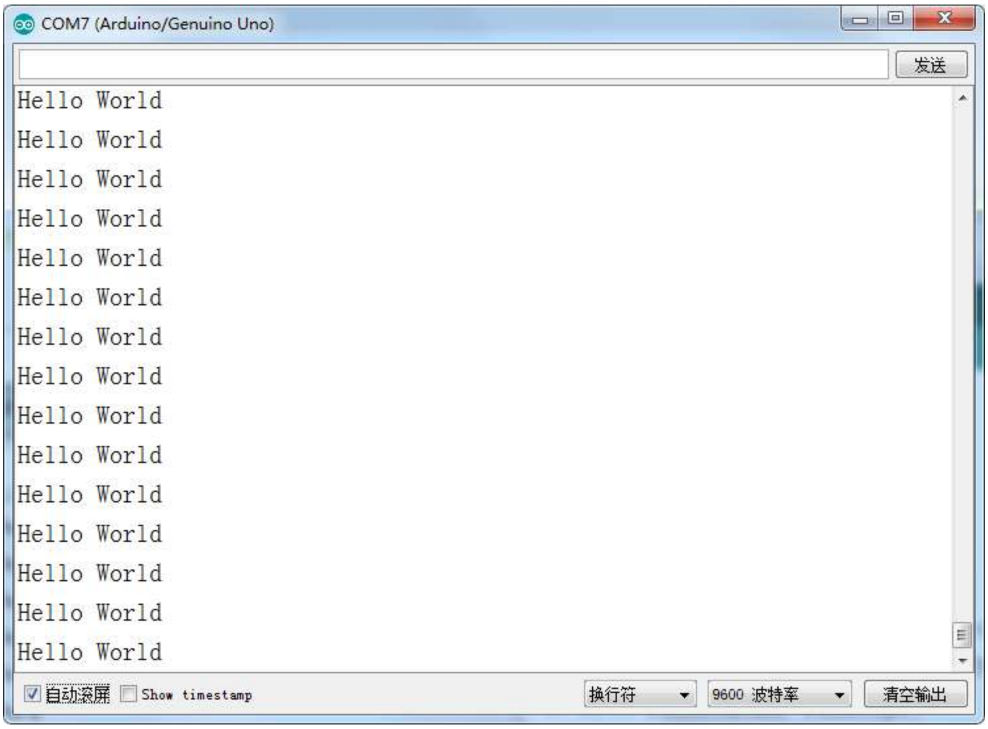



Figure 2-3-7

# Chapter 3 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.

Starter-bot robot has two DC motors. We can use DC motors to control building blocks to make DC motors turn, thus driving Starter-bot robot to move.

### 3.1.2 DC motor test

In the robot module, we can find three control blocks of DC motor module, which are motor pin building blocks, direction setting, speed building blocks, turning angle setting and speed building blocks, as shown in Figure 3-1-2-1.

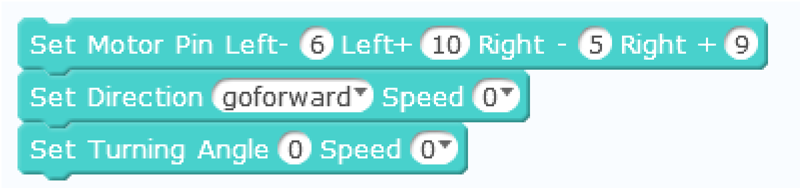



Figure 3-1-2-1

- Setting up the motor pin building blocks: Used to set the pins connecting the two motors of the control robot to the main control board. The default pins are selected in this tutorial.
- Set the 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

The following program is written on Magic Block. They can control the motor forward and reverse respectively, and then let the robot move forward, backward, left and right directions. They run the program separately and carefully observe the rotation direction of two DC motors when the robot moves in each direction. The programming of the four directions is shown in Figure 3-1-2-2.

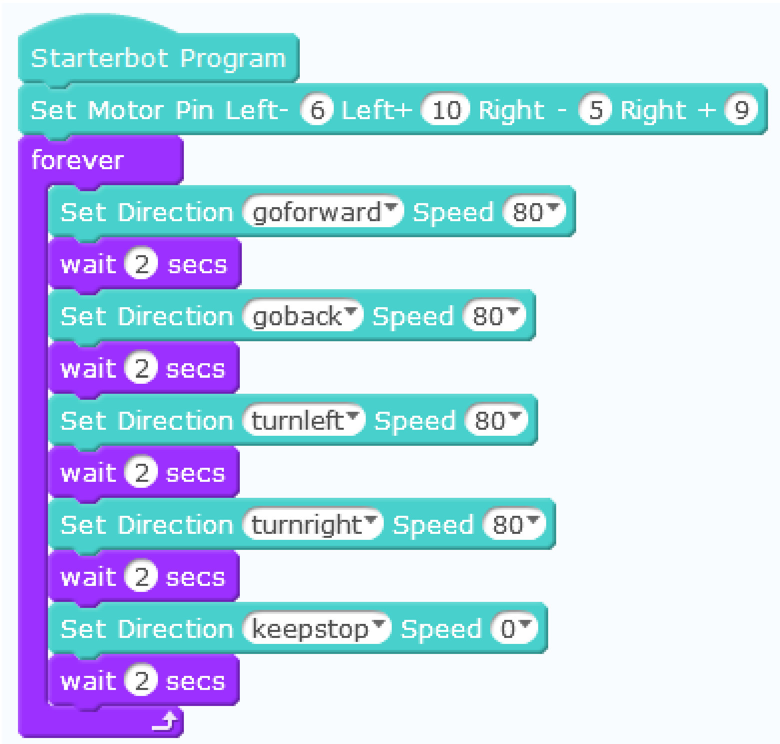



Figure 3-1-2-2

hrough the above practical operation, we should have understood the programming method of letting the robot move in a single direction by controlling the motor. Now let's explore the programming method of letting the robot walk in a square.

## 3.1.3 Programming control and debugging the robot to take the square route.

The way the robot moves in a square is to go forward, turn left, move forward, turn left, move forward, turn left, turn left (reverse can also be done). The path of the robot is shown in Figure 3-1-3-1, and the reference program is shown in Figure 3-1-3-2.

Figure 3-1-3-1

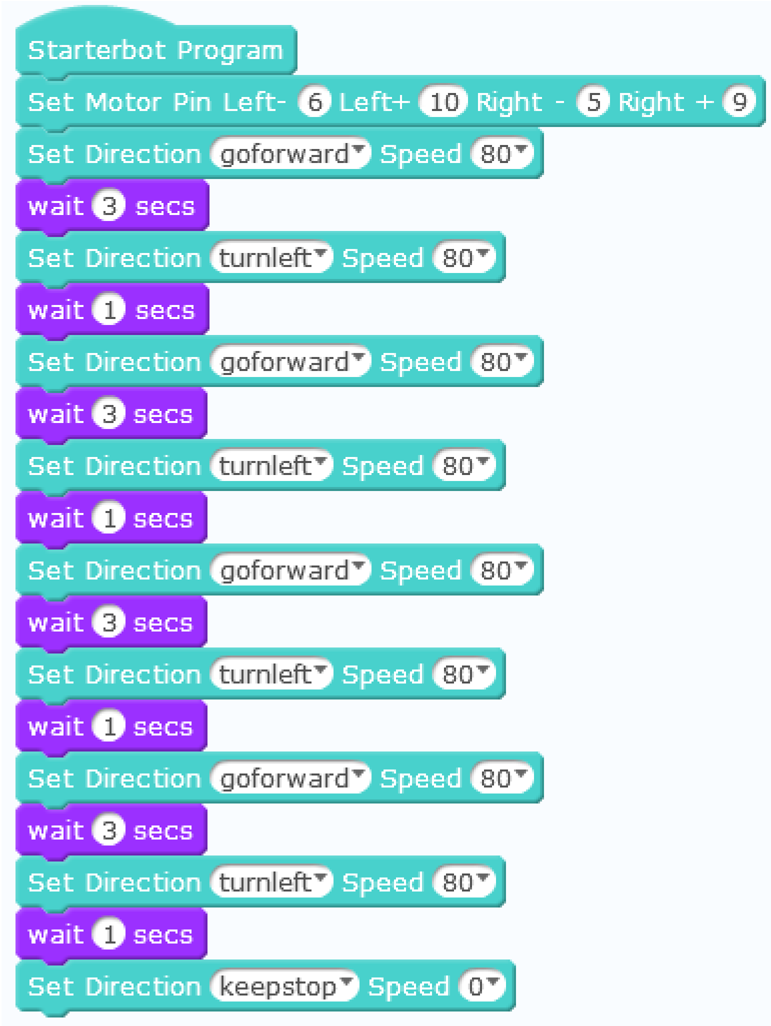



Figure 3-1-3-2

**Be careful:**

1) The delay blocks under the "forward" building blocks are to control the forward distance of the robot, while the delay blocks under the left turn are to control the turning angle of the robot. When writing the program, the delay time should be modified according to the actual walking condition of the robot.
2) t the end of the program, the motor must stop turning, otherwise the robot will remain in its original state of motion and turn left all the time.

### 3.1.4 Discover the Rule of Program and Optimize Program

By observing the program of Figure 3-1-2-2, we can find that the path of the robot walking square is actually to control the wheel to repeat four times forward and turn left 90 degrees. The optimization program is shown in Figure 3-1-4-1.

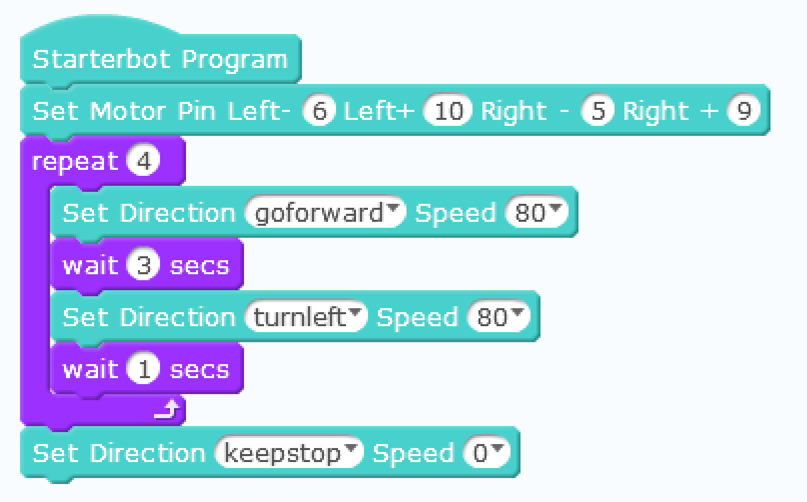
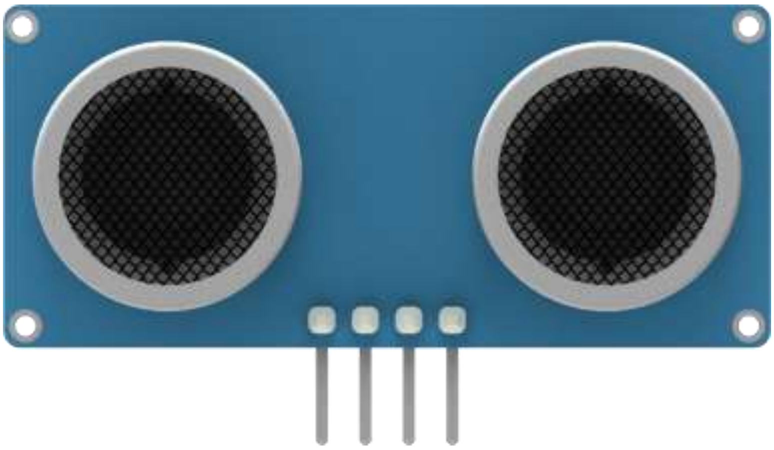



Figure 3-1-4-1

## 3.2 Ultrasound module

### 3.2.1 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 sensor has two "eyes", one of which emits ultrasound, and the other "eyes" receives 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 that distance = speed * time, then the distance is measured by ultrasound.= The speed of the ultrasonic wave (2) so that the distance can be calculated.



## 3.2.2 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 3-2-2-1, the building blocks can detect the distance between the robot and the obstacle in front. Note: In order to use the ultrasonic module, we should first 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 to the distant place, the ultrasonic wave emitted by the ultrasonic module can not be received, so the distance between the robot and the obstacle can not be measured. The ultrasonic module installed on the robot can not be detected. The detection range is 5CM ~ 400CM.

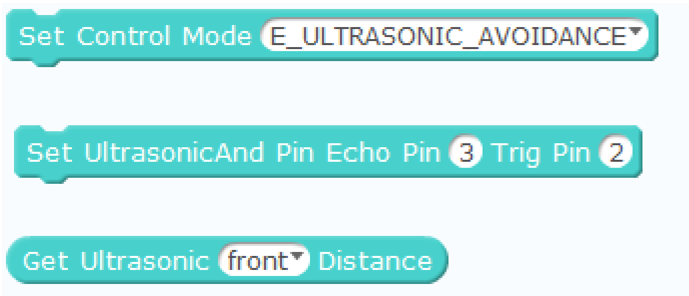



Figure 3-2-2-1

## 3.2.3 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 can observe the ultrasonic distance measured by serial port printing, and we can more intuitively see the process of measuring the distance of the ultrasonic obstacle avoidance module. Now we first write a test program.

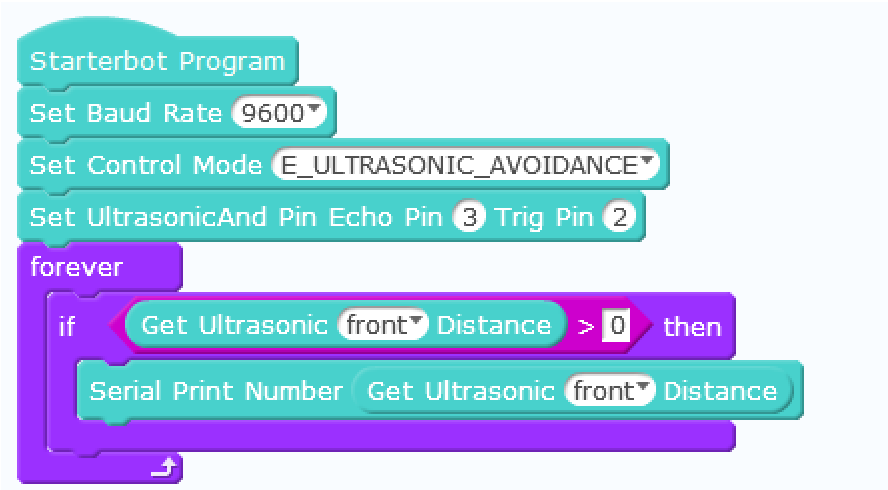



Figure 3-2-3-1

We write the program as shown in Figure 3-2-3-1, then upload the program successfully, open the serial port, and then close or far from the ultrasonic module, the serial port will print the corresponding distance.

## 3.2.4 Writing Robot Program for Tango Dance

Tango dance is a kind of double dance. The dance step is to let the robot dance with our palms when one person is close. If the hand is close to it and the hand moves away from it, the distance between the hand and the robot should be measured by the ultrasonic module to determine whether the robot moves forward or backward. The larger the measuring distance of ultrasound means that the hand is far away from the robot, and after a certain distance, the robot moves forward; the smaller the measuring distance of ultrasound means that the hand is close to the robot, and after a certain distance, the robot retreats; let's set the median value to 20CM first, then let's start programming.

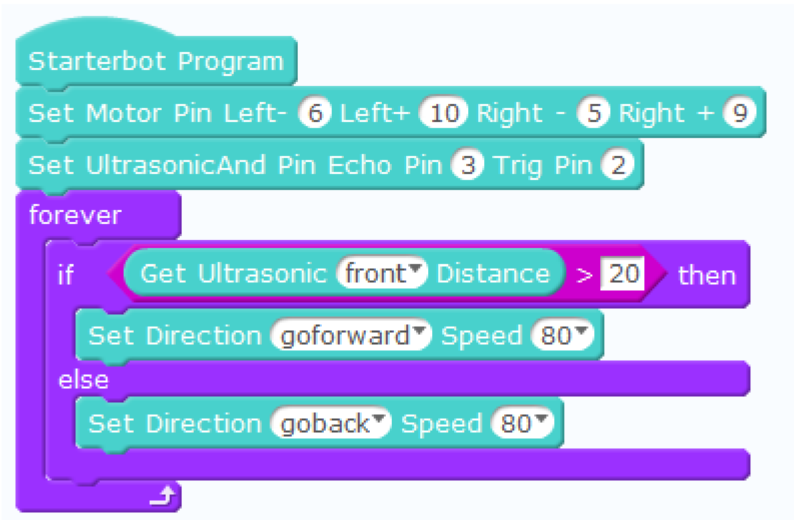



Figure 3-2-4-1

Run the program shown in Figure 3-2-4-1 above. When the hand is far away, the robot moves forward; when the hand is near, the robot retreats. But when the hand is still, the robot will move back and forth repeatedly, because we did not set a stop motion interval for the robot. How to stop the robot in an interval?

| The Value Measured by the Ultrasound Module | Motion State of Robot |
|---|---|
| Greater than 20 | Forward |
| Between 12~20 | Stop |
| Less than 12 | Back off |

Table3-2-4-2

As shown in Table 3-2-4-2 above, a new judgment statement is needed for this situation between 12 and 20. As shown in Figure 3-2-4-3, when the distance measured by ultrasound is greater than 12CM and less than 20CM, the robot stops.


Figure 3-2-4-3

## 3.2.5 Optimizing Dancing Robot Program

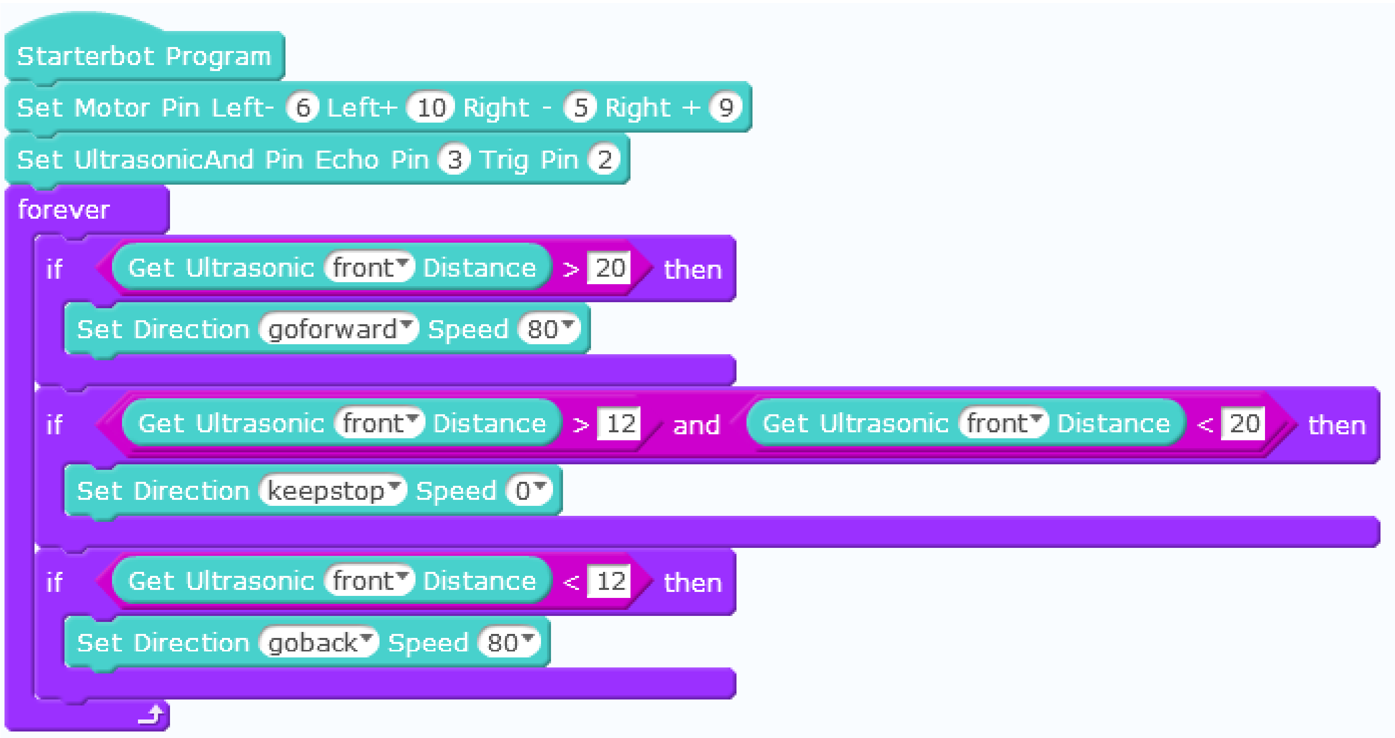



Figure 3-2-5-1

# 3.3 Ultrasonic obstacle avoidance robot

We've learned the roles of robots in moving forward, backward, left, right and ultrasonic modules. Now let's explore how robots use ultrasound to avoid obstacles in motion. The so-called obstacle avoidance of robots is to let robots move around obstacles (left or right or backward) if there are obstacles in front of them. If the object continues to move, the robot needs to turn left and right and judge the distance between left and right directions, then the ultrasonic module needs to rotate left and right to "explore the way". At this time, the steering gear is used to realize the left and right turn of the ultrasonic module. How does the steering gear work? Next we will introduce the use of the steering gear.

## 3.3.1 Operating Principle of Steering Engine

he 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 3-3-1-1. 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, the mechanical sound of clicking and clicking is generated by the steering gear rotating on the robot body. Figure 3-3-1-2 is the most commonly used SG90 steering gear physical diagram at this stage.

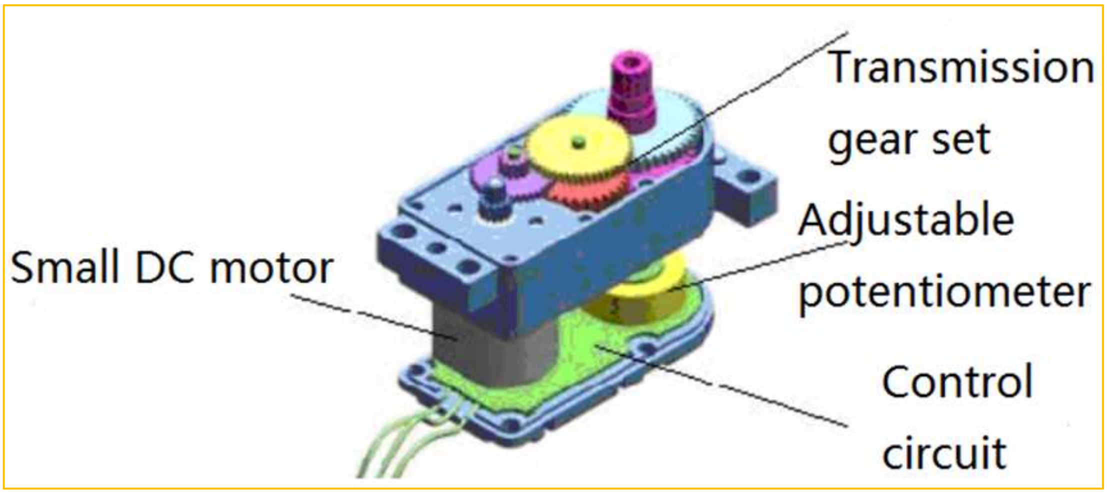



Figure 3-3-1-1

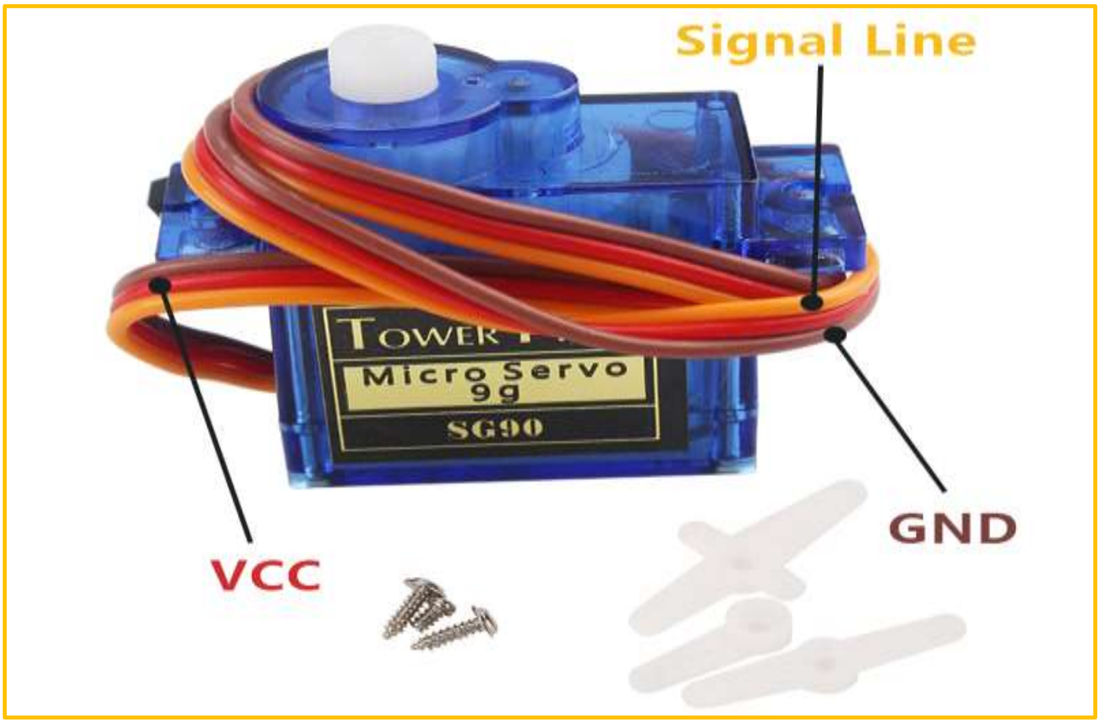



Figure 3-3-1-2

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.

### 3.3.2 Steering engine test

The steering gear on the Starterbot robot is installed in the front, and the ultrasonic module is fixed on the steering gear. When the ultrasonic module is to measure the distance between the robot and the front obstacle, the steering gear is 90 degrees. When the ultrasonic module measures the distance between the robot and the left obstacle, the steering gear rotates 90 degrees to the left. When the ultrasonic module measures the distance between the robot and the right obstacle, the steering gear rotates 90 degrees to the left. The steering gear rotates 90 degrees to the right. Next, we write a program to test the steering gear by using the steering gear to make the ultrasonic wave move forward, then left and then right.

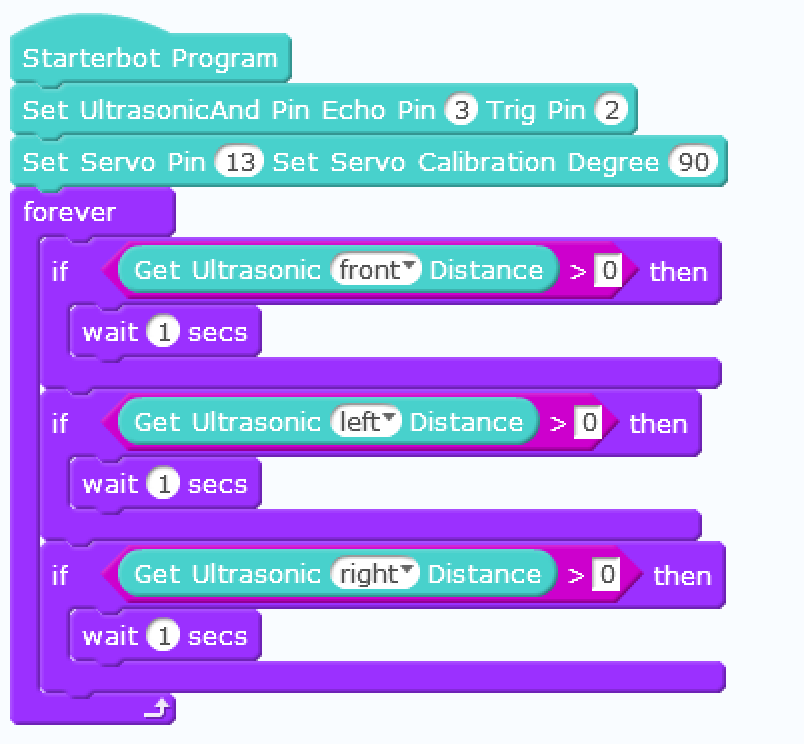
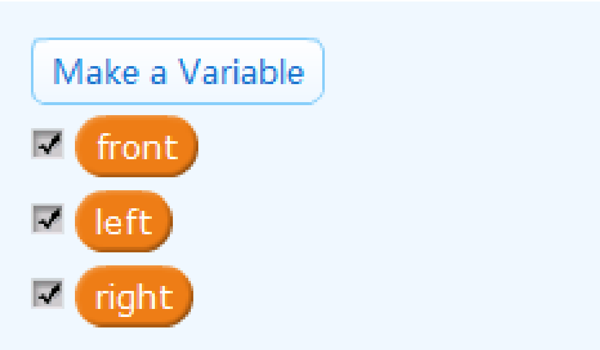



Figure 3-3-2-1

Above is the testing procedure of the steering gear. After uploading the program to the robot, turning on the power supply, we can see that the steering gear makes the ultrasonic module turn forward, then left, and then right.

## 3.3.3 Programming of Obstacle Avoidance Robot

The main work flow of the ultrasonic obstacle avoidance robot is as follows: after power-on, the steering gear makes the ultrasonic module automatically turn to 90 degrees (alignment in front of the robot). The ultrasonic module measures the distance of the obstacle in front. If the value is larger than the set safe distance (12CM), the robot continues to move forward, otherwise stops. At this time, the steering gear makes the ultrasound rotate 90 degrees to the right first, and the ultrasonic mode. The distance between the block and the right obstacle is measured. Then the steering gear rotates the ultrasonic wave 180 degrees to the left. The ultrasonic module measures the distance between the ultrasonic module and the left obstacle. Then the steering gear lets the ultrasonic module return to 90 degrees. The robot compares the two measured distances. If the left is larger than the right, the robot turns left, then drives, then turns right, and vice versa. When the distance is less than the safe distance, the robot will turn around and walk. Following this train of thought, let's start to write the program of obstacle avoidance robot. Note: In the following program, we use the variable module to declare the left and right distance before ultrasound.


Make a Variable
front
left
right

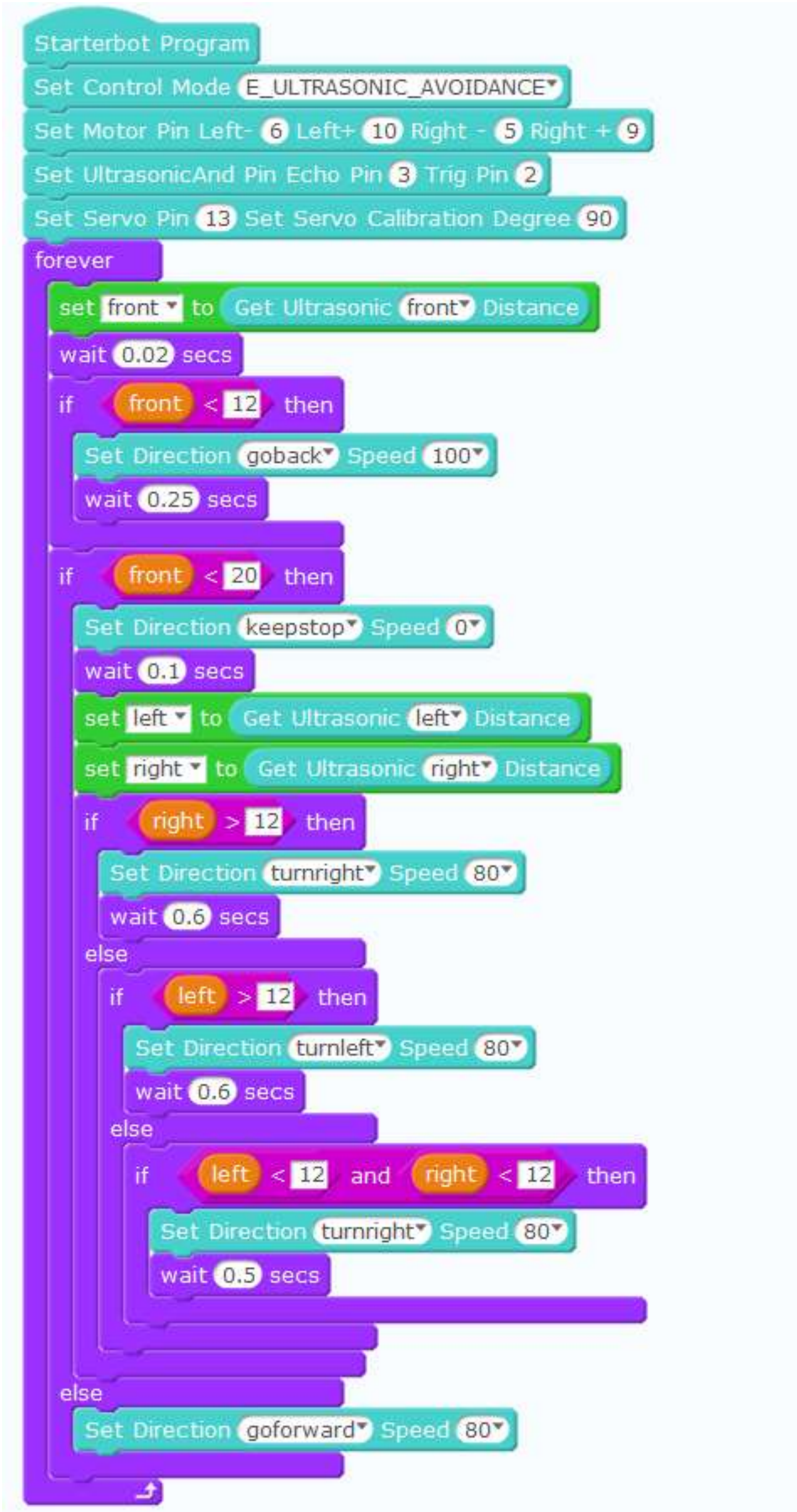

Starterbot Program
Set Control Mode E_ULTRASONIC_AVOIDANCE
Set Motor Pin Left- 6 Left+ 10 Right - 5 Right + 9
Set UltrasonicAnd Pin Echo Pin 3 Trig Pin 2
Set Servo Pin 13 Set Servo Calibration Degree 90
forever
set front to Get Ultrasonic front Distance
wait 0.02 secs
if front < 12 then
Set Direction goback Speed 100
wait 0.25 secs
if front < 20 then
Set Direction keepstop Speed 0
wait 0.1 secs
set left to Get Ultrasonic left Distance
set right to Get Ultrasonic right Distance
if right > 12 then
Set Direction turnright Speed 80
wait 0.6 secs
else
if left > 12 then
Set Direction turnleft Speed 80
wait 0.6 secs
else
if left < 12 and right < 12 then
Set Direction turnright Speed 80
wait 0.5 secs
else
Set Direction goforward Speed 80

Figure 3-3-3-1

# 3.4 Infrared Barrier Avoidance Robot

## 3.4.1 Principle of Infrared Obstacle Avoidance Module

he principle of the infrared obstacle avoidance module is similar to that of the ultrasonic module. The difference is that the ultrasonic module measures the distance by transmitting ultrasonic wave. The infrared obstacle avoidance module measures the distance by transmitting infrared ray. The infrared obstacle avoidance module has a pair of infrared emission and receiving tubes, which emits infrared ray. When the emitted infrared ray meets the obstacle (reflecting surface), the infrared ray is reflected back and received. Receiving and receiving, the green indicator lights up. After the infrared signal is processed by the control motherboard of the robot, the detection distance is calculated. The infrared obstacle avoidance module is shown in Figure 3-4-1-1.

Figure 3-4-1-1

## 3.4.2 Usage of Infrared Obstacle Avoidance Module

In the control module of the robot, we find the control building blocks of the infrared obstacle avoidance module. As shown in Figure 3-4-2-1, the building blocks can detect the distance between the robot and the left and right obstacles. Note: In order to use the infrared obstacle avoidance module, we first need to choose the mode to control the building blocks and set the mode to the infrared obstacle avoidance mode. When the obstacles completely block the ultrasonic module or face the ultrasonic module to the distance, the infrared radiation emitted by the infrared obstacle avoidance module can not be received because it is too far or too close to detect the distance between the robot and the obstacle, so it is used. When infrared obstacle avoidance module ranging, it is appropriate to keep the distance between the obstacle and the infrared obstacle avoidance module between 5 and 15.

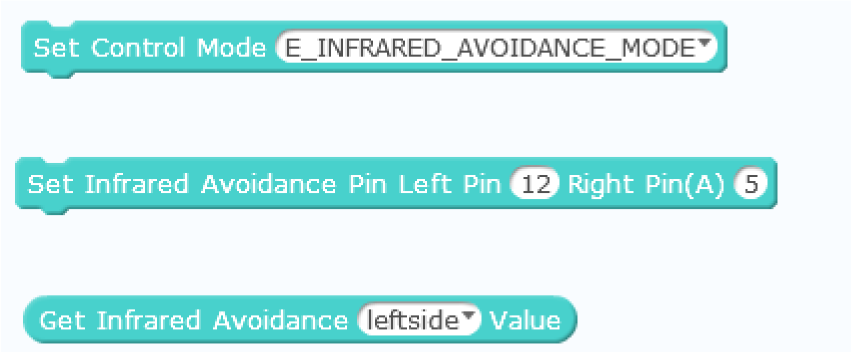



Figure 3-4-2-1

## 3.4.3 Test of Infrared Obstacle Avoidance Module

We can first write a program to test the infrared obstacle avoidance module, using serial port printing. When we hand close to the infrared obstacle avoidance module and far away from the infrared obstacle avoidance module, we observe the contents of serial port printing. This infrared obstacle avoidance module only reads two analog values of 0 or 1. 1 represents the undetected obstacle, 0 represents the detected obstacle, and the distance limit of whether the obstacle is detected or not. It needs to be adjusted with a screwdriver. Let's write a test program first.

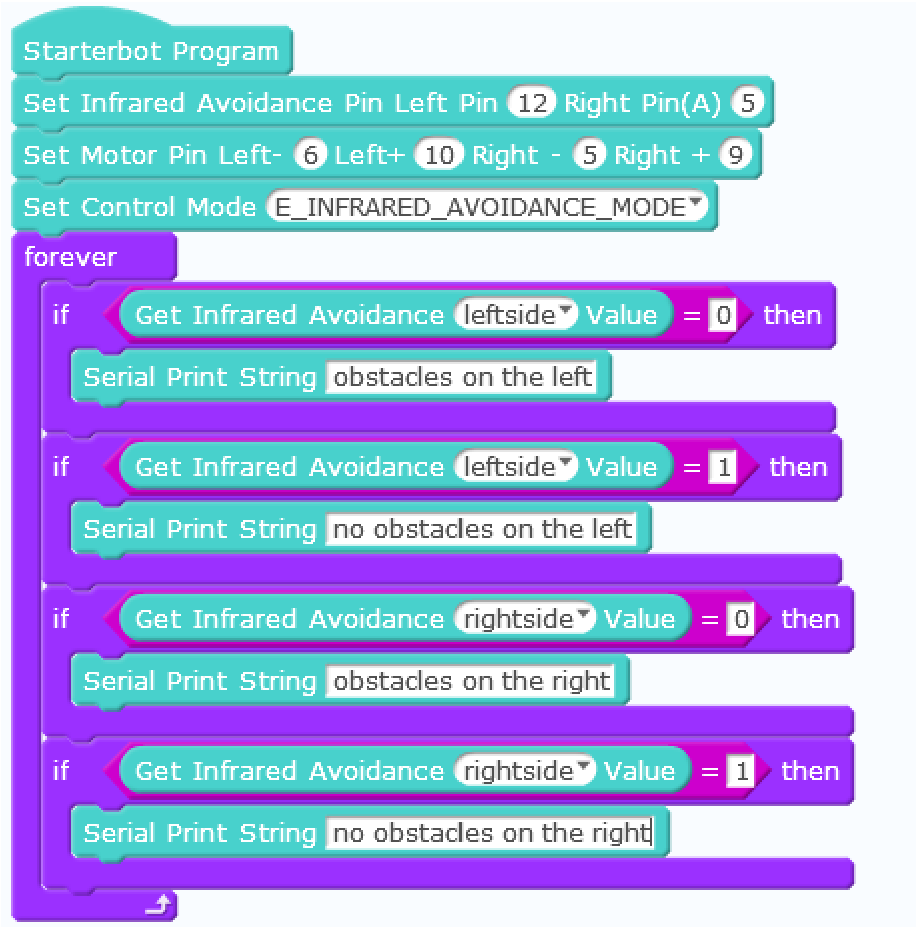



Figure 3-4-3-1

We write the program as shown in Figure 3-18, then upload the program successfully, open the serial port, and then close or far from the infrared obstacle avoidance module, the serial port will print the corresponding distance.

## 3.4.4 Robot Writing Runway

If we want to let the robot run smoothly on the track to avoid encountering the track, we need to keep a certain distance between the robot and both sides of the track at all times. At this time, we can use the infrared obstacle avoidance module to achieve this effect. The basic idea is to set the safe distance of the infrared obstacle avoidance module first, the robot moves forward, and the infrared obstacle avoidance module measures it with the right track. The distance between the edge and the left side of the track is measured by the infrared obstacle avoidance module. Then the robot compares the distance measured twice. If the left is larger than the right, the robot turns left, then drives, and vice versa. If the distance measured on both sides is less than the safe distance, the robot will stop walking. Follow this train of thought to learn how to write a racetrack robot.


Figure 3-4-4-1

Running the program shown in Figure 3-19 above, the robot moves forward; when the left side of the robot approaches the edge of the runway, the robot turns right. When the right side of the robot is close to the edge of the runway, the robot turns left, and when the left and right sides of the robot are close to obstacles, the robot retreats.

## 3.5 Ultrasound+Infrared Barrier Avoidance Robot

We have already known the use and programming of the ultrasonic module and the infrared obstacle avoidance module separately. Next, we will combine the ultrasonic module and the infrared obstacle avoidance module to use together. The programming idea is to set the mode as the ultrasonic infrared obstacle avoidance mode first, then set the ultrasonic module, the safe distance of the infrared obstacle avoidance module, and then detect the obstacle in front of the robot by ultrasonic. Obstacles, infrared obstacle avoidance module detects whether there are obstacles on the left and right sides of the robot. Now let's write the program of the ultrasonic infrared obstacle avoidance robot.

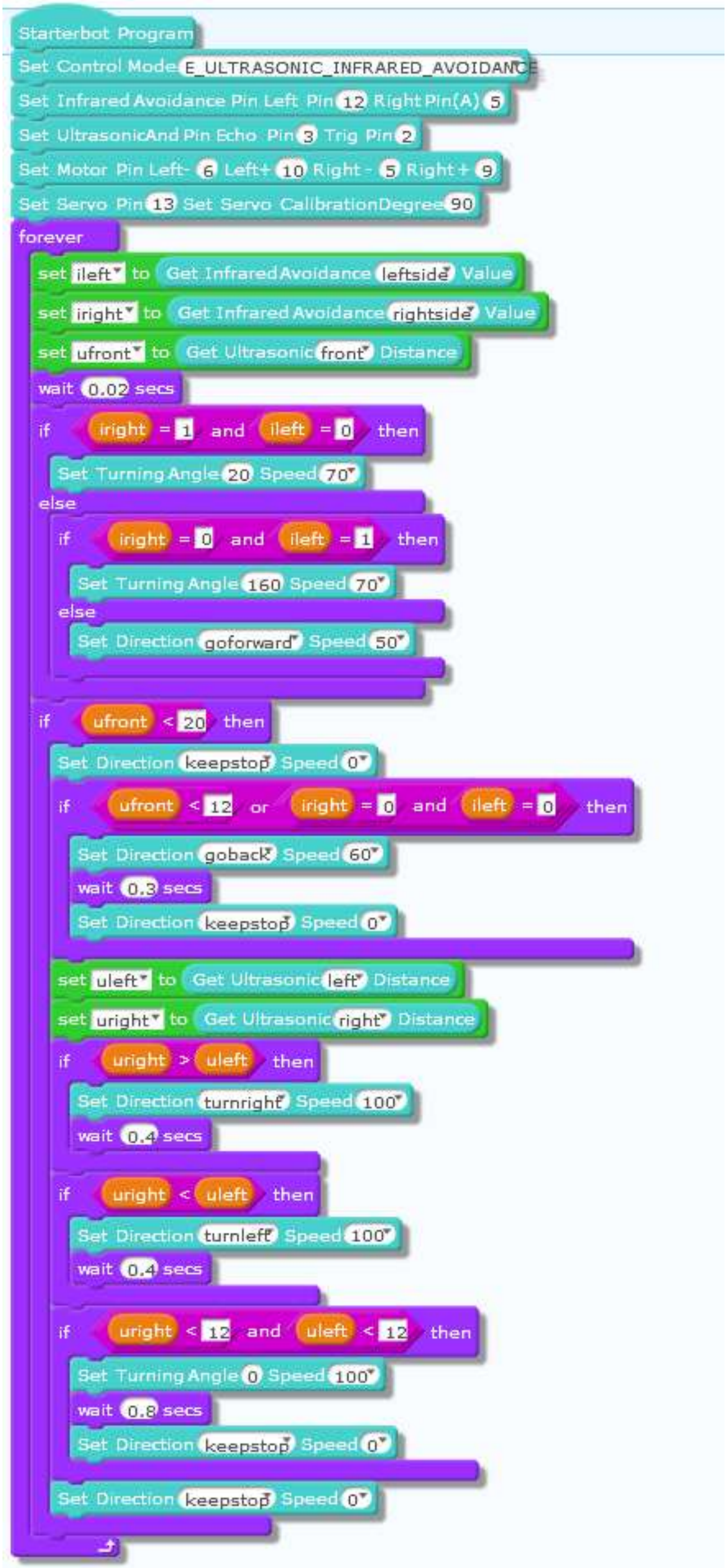

Starterbot Program
Set Control Mode E_ULTRASONIC_INFRARED_AVOIDANCE
Set Infrared Avoidance Pin Left Pin 12 Right Pin(A) 5
Set UltrasonicAnd Pin Echo Pin 3 Trig Pin 2
Set Motor Pin Left- 6 Left+ 10 Right- 5 Right+ 9
Set Servo Pin 13 Set Servo CalibrationDegree 90
forever
set ileft to Get Infrared Avoidance leftside Value
set iright to Get Infrared Avoidance rightside Value
set ufront to Get Ultrasonic front Distance
wait 0.02 secs
if iright = 1 and ileft = 0 then
Set Turning Angle 20 Speed 70
else
if iright = 0 and ileft = 1 then
Set Turning Angle 160 Speed 70
else
Set Direction goforward Speed 50
if ufront < 20 then
Set Direction keepstop Speed 0
if ufront < 12 or iright = 0 and ileft = 0 then
Set Direction goback Speed 60
wait 0.3 secs
Set Direction keepstop Speed 0
set uleft to Get Ultrasonic left Distance
set uright to Get Ultrasonic right Distance
if uright > uleft then
Set Direction turnright Speed 100
wait 0.4 secs
if uright < uleft then
Set Direction turnleft Speed 100
wait 0.4 secs
if uright < 12 and uleft < 12 then
Set Turning Angle 0 Speed 100
wait 0.8 secs
Set Direction keepstop Speed 0
Set Direction keepstop Speed 0

Figure 3-5-1

Figure 3-5-1 is the program of the ultrasonic infrared obstacle avoidance robot. In the above program, the delay time after steering needs to be adjusted according to the speed of the robot, which needs attention in programming.

# Chapter 4 Tracking Robot

## 4.1 Principle of Tracking Module

Tracking module works by using the characteristics of different reflective properties of infrared rays on the surface of objects with different colors. During the movement of the robot, infrared light is continuously emitted to the ground. When the emitted infrared rays are not reflected back or reflected back, but the intensity is not high enough, the robot thinks that the area is the route it is going to take; when the infrared light is reflected back to the machine, the robot thinks that the area is the route it is going to take. Man recognizes that the robot thinks the area is not the route it takes. The robot is based on the reflected infrared light to determine the position of the black line and its walking path. The Starterbot robot is equipped with three tracking modules, one on the left, one on the right, and the principle of tracking is shown in Figure 4-1-1.

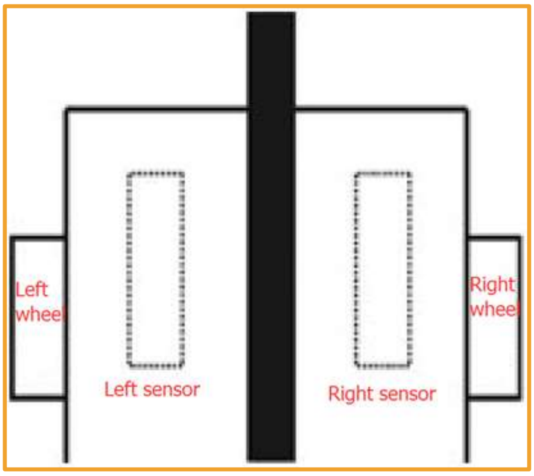



Figure 4-1-1

## 4.2 Use of Trace Module

In the control module of the robot, the control building blocks of the infrared tracking module are found. As shown in Figure 4-2-1, using the building blocks, the robot can automatically run along its own track. Note: In order to use the infrared tracking module, we first need to select the mode to control the building blocks, and set the mode to the infrared tracking mode. Besides the mode setting, we need to define a variable to save the values read by each infrared tracking, so that when the robot runs off, it can correct the direction according to the values read last time and find the route again. Read the trace value building blocks, the drop-down menu has all black, all white, left 1, left 2, center, right 1, right 2 options, their interpretation is as follows:

- All black: black lines were detected in all three tracking modules.
- Whiteness: None of the three tracking modules detected black lines.
- Left 1: The black line was detected by the tracking module on the left.
- Left 2: Black lines were detected in the left and middle tracing module.
- Center: The black line was detected by the tracing module in the middle.
- Right 1: The black line was detected by the tracking module on the right.
- Right 2: The black line was detected by the tracing module on the right and in the middle.

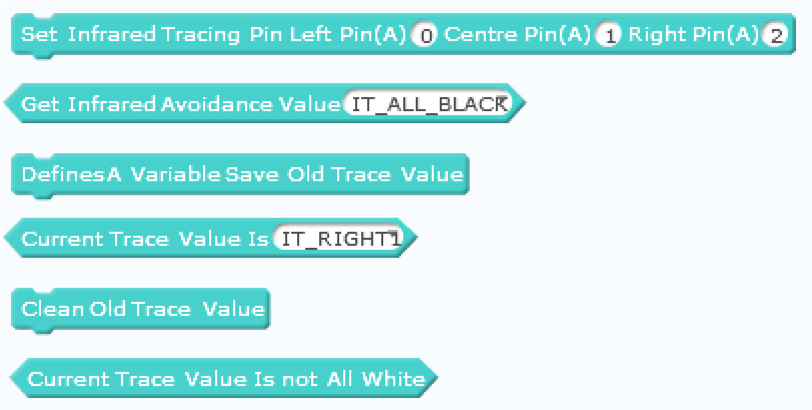



Figure 4-2-1

## 4.3 Testing of Infrared Tracking Module

We can first write a program to test the infrared tracing module, using serial port printing, using black tape near the infrared tracing module and far away from the infrared tracing module, we can observe the contents of serial port printing, we can more intuitively see the tracing effect of each module when the infrared tracing module tracks. Now we first write a test program.

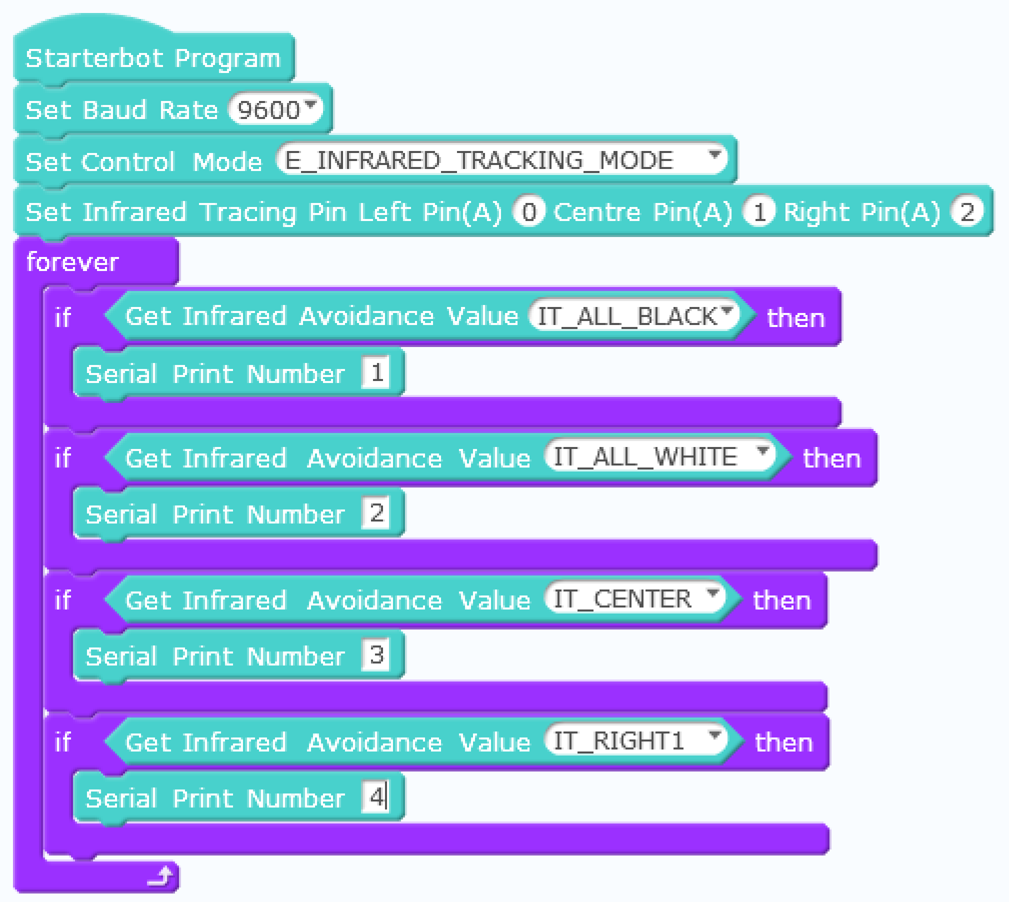



Figure 4-3-1

As shown in Figure 4-3-1 above, when we open the serial port, we close the black strip to any one of the tracking modules, and the serial port will print the corresponding number.

## 4.4 Writing Tracking Robot Program

The programming idea of the tracking robot is to set the infrared tracking mode first, and then set the connecting pins of the infrared tracking module according to its actual wiring mode. Then the robot executes the corresponding left-to-left, forward and right-to-right movement according to the detection results of the left-to-center and right-to-right tracking modules. Let's write the program of the tracking robot together.

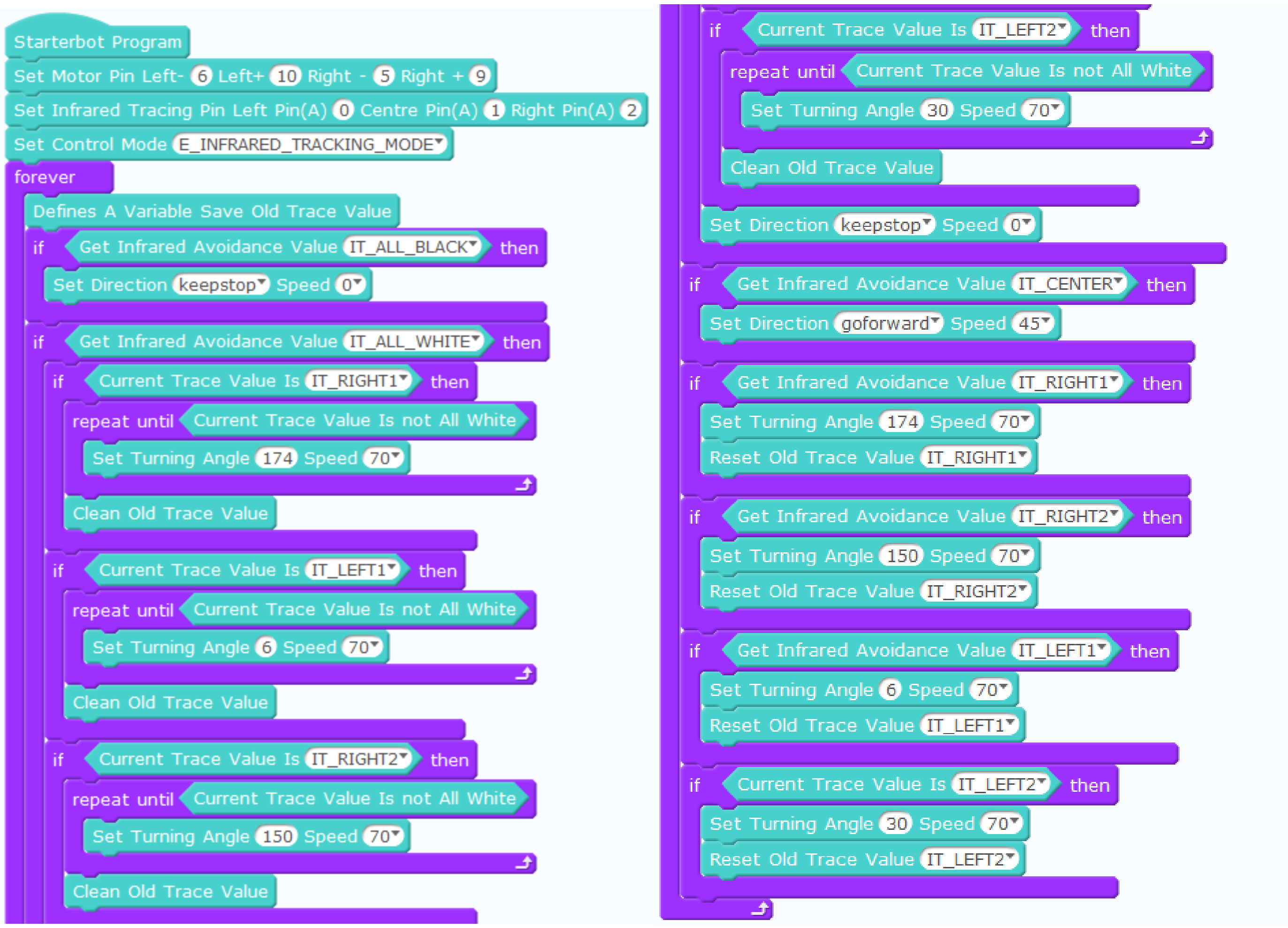



Figure 4-4-1

# Chapter 5 Infrared Telecontrol Robot

## 5.1 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. After receiving the instruction from the remote controller, the receiver 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, and then control its limbs (four wheels) to do the corresponding. Actions, remote controls and receivers are shown in Fig. 5-1-1.


Figure 5-1-1

## 5.2 The Use of Remote Controller

To control the robot through the remote controller, 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 controller, so that when we press the key of the remote controller, the robot will do the action defined in our program.

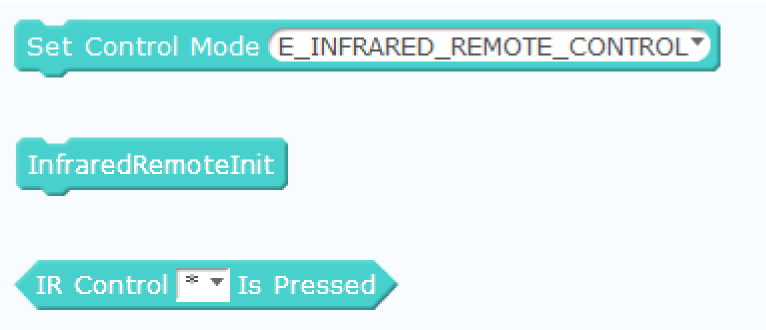



Figure 5-2-1

## 5.3 Infrared remote control test

We can first write a program to test infrared remote control, using serial port printing, when the remote control button is pressed, the serial port prints the corresponding key values, we observe the contents of serial port printing, we can see which key of the infrared remote control is pressed. Now we first write a test program.

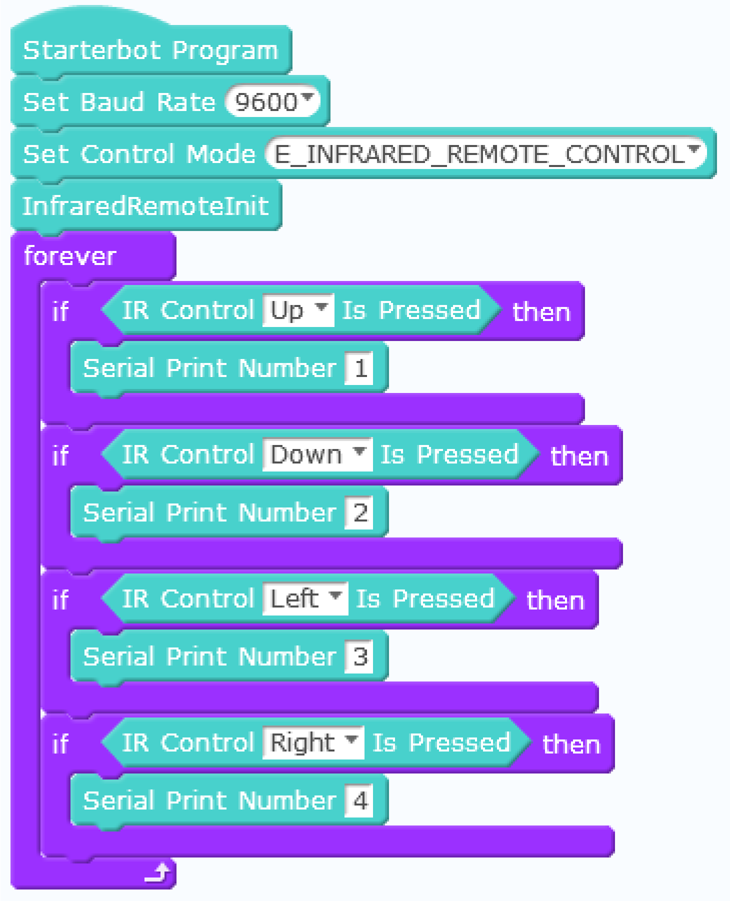



Figure 5-3-1

We write the program as shown in Fig. 5-3, then upload the program successfully, open the serial port, then press 1, 2, 3 and 4 keys on the robot with the remote control, and the serial port will print the corresponding key numbers.

# 5.4 Programming of Remote Control Robot


Figure 5-4-1

We write a remote control button function as defined in Fig. 5-4-1, and set the speed adjustment 1 (*) for acceleration and the speed adjustment 2 (#) for deceleration. We have to do a 0.12S delay after the acceleration and deceleration procedures, which is to make the robot accelerate and decelerate a little bit more smoothly. Without the delay, the speed will directly change to the maximum or minimum.


Starterbot Program
Set Control Mode E_INFRARED_REMOTE_CONTROL
Set Motor Pin Left- 6 Left+ 10 Right - 5 Right + 9
InfraredRemoteInit
forever
if IR Key Is Pressed then
if IR Control + Is Pressed then
Speed Up
set currentSpeed to Get Current Speed
if IR Control - Is Pressed then
Speed Down
set currentSpeed to Get Current Speed
if IR Control Up Is Pressed then
Set Direction goforward Speed currentSpeed
if IR Control Down Is Pressed then
Set Direction goback Speed currentSpeed
if IR Control Left Is Pressed then
Set Direction turnleft Speed currentSpeed
if IR Control Right Is Pressed then
Set Direction turnright Speed currentSpeed
if IR Control ok Is Pressed then
Set Direction keepstop Speed currentSpeed
wait 0.11 secs
else
Set Direction keepstop Speed currentSpeed

Figure 5-4-2

Note: When setting the steering speed, the speed value can not be set too small, otherwise the robot will not be strong enough when turning, and will be stuck in place. Therefore, the best setting speed value is above 40.

# Chapter 6 Bluetooth Remote Control Robot

## 6.1 Bluetooth Control Principle

Bluetooth is a kind of long-distance communication mode. We send the instructions to the Bluetooth module on the robot from the APP end of the mobile phone through the Bluetooth of the mobile phone. Then the Bluetooth module on the robot sends the instructions to the robot brain (main control board). The robot controls its limbs (four wheels) to do the corresponding actions. At the same time, the robot will also install Bluetooth through itself. The module sends the action being done to the mobile app, so that the mobile app can be remotely controlled by Bluetooth.

## 6.2 Bluetooth Module and Starterbot Robot APP KeywishBot Connection Method

The Starterbot robot is equipped with a Bluetooth module (as shown in Figure 6-2-1). The blue light on the Bluetooth module flickers when the power is turned on. After turning on the APP, select Starterbot, Bluetooth module and the Starterbot robot APP KeywishBot on the mobile phone (as shown in Figure 6-2-3) and connect via Bluetooth. When the connection is successful, the blue light will be on for a long time, and then select "remote". The control interface (Fig. 6-2-4) enters the gravity induction control interface (Fig. 6-2-5). Handle control mode can also be switched as shown in Figure 6-2-6.

Figure 6-2-1

Figure 6-2-3


Figure 6-2-5


Figure 6-2-6

## 6.3 Bluetooth module testing

We can first write a program to test Bluetooth module, using serial port printing. When the Bluetooth module of the robot receives data and the serial port prints the data received by Bluetooth, we can see the effect of Bluetooth data transmission by observing the content sent by Bluetooth and the content printed by serial port. Here we write a test program.


We have written the program as shown in Figure 6-3-1. After uploading the program successfully, we open the serial port, then connect it with the robot Bluetooth using mobile APP, press the left and right buttons, and the serial port will print the corresponding key numbers.

## 6.4 Programming Thought of Bluetooth Remote Control Robot

We have written the program as shown in Figure 6-3-1. After uploading the program successfully, we open the serial port, then connect it with the robot Bluetooth using mobile APP, press the left and right buttons, and the serial port will print the corresponding key numbers.

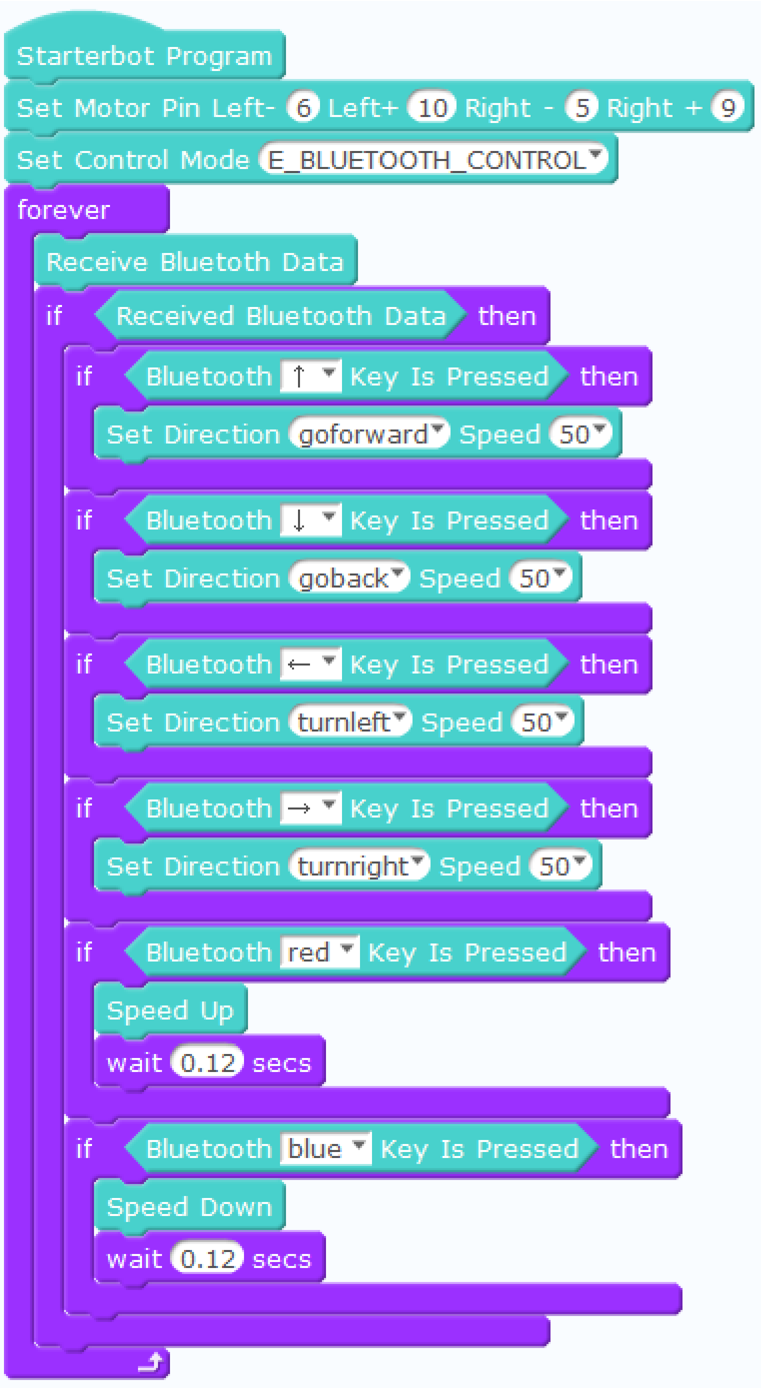



Figure 6-4-1

In the above programming, notice that we have made a 0.12S delay after the acceleration and deceleration program. This is to make the acceleration and deceleration of the robot a little more smooth. If there is no delay, the speed will directly change to the maximum or minimum.

# Chapter 7 PS2 Telecontrol Robot

## 7.1 Principle of PS2 remote control handle

The PS2 handle is composed of a handle and a receiver (as shown in Figure 7-1-1). The handle needs two sections of 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, before searching for the receiver, the handle will enter standby mode, and the light on the handle will go out. At this time, press START. ” Key, wake up 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. When the pairing is successful, the green light on the receiver is always on, the light on the handle is always on, and the button "MODE" (with different batches of the handle, the above logo may be "AnalyOG", but it will not affect the use), so the "red light mode" can be selected. "Green light mode".


Figure 7-1-1

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 action according to the command received.

## 7.2 PS2 remote control test

We can first write a program to test the PS2 remote control, using serial port printing, when the PS2 remote control keys press, serial port printing the corresponding key values, we observe the contents of serial port printing, you can see which keys of the PS2 remote control are pressed, next we first write a test program.

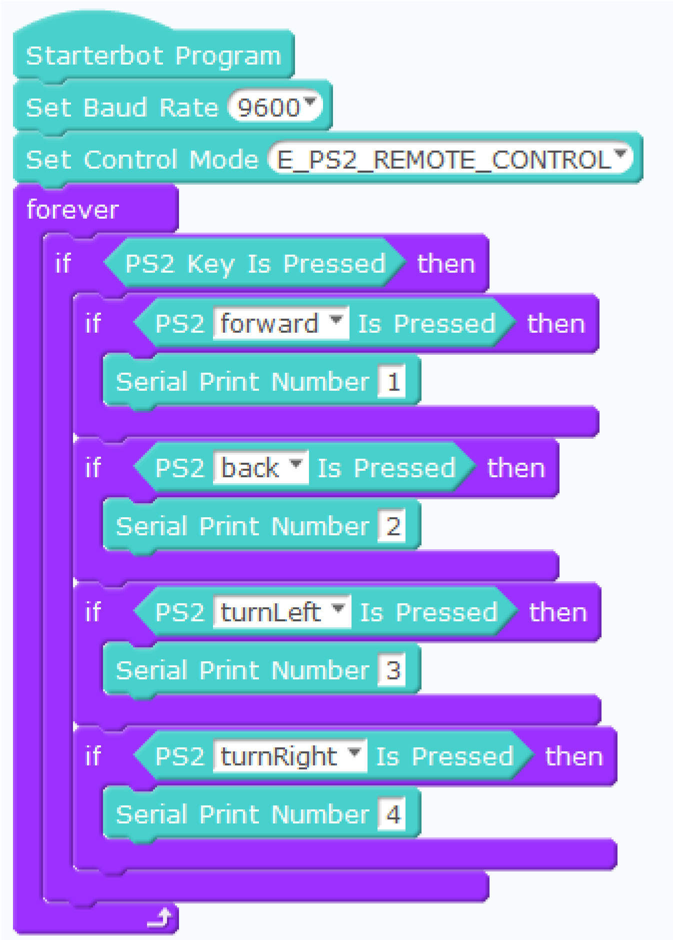



Figure 7-2-1

Fig. 7-3 bit PS2 test program, after uploading the program to the robot, open the serial port monitor, press the buttons around PS2, and the serial port will print the corresponding number.

## 7.3 Programming Thought of PS2 Remote Control Robot

The programming idea of PS2 remote control robot is to set the mode as PS2 control mode first, then define the function of keys on PS2 handle according to its own needs. Our next programming is programmed according to the definition of keys shown in Figure 7-3. Then the robot moves left, forward and right according to the received value of keys on the handle, or accelerates, decelerates, operates the steering gear, etc. PS2 handle is remote. Control the car and define all key functions as shown in Figure 7-3-1.


Figure 7-3-1

- Sign UP: Forward
- Sign DOWN: Back
- Sign LEFT: Turn left
- Sign RIGHT: Turn right
- Sign A: Acceleration
- Sign B: Left turn of steering gear
- Sign C: Deceleration
- Sign D: steering gear right turn

Now let's write the program of PS2 remote control robot.

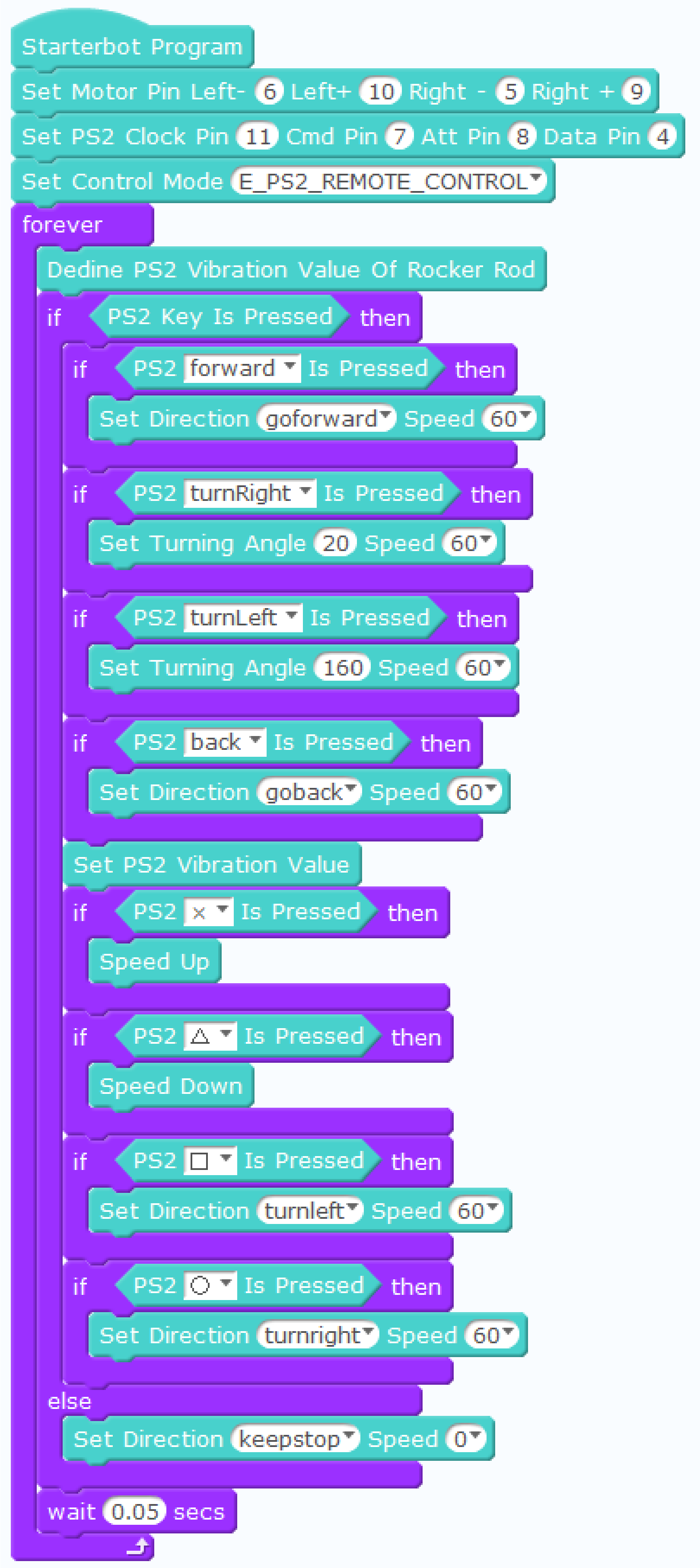

Starterbot Program
Set Motor Pin Left- 6 Left+ 10 Right - 5 Right + 9
Set PS2 Clock Pin 11 Cmd Pin 7 Att Pin 8 Data Pin 4
Set Control Mode E_PS2_REMOTE_CONTROL
forever
Dedine PS2 Vibration Value Of Rocker Rod
if PS2 Key Is Pressed then
if PS2 forward Is Pressed then
Set Direction goforward Speed 60
if PS2 turnRight Is Pressed then
Set Turning Angle 20 Speed 60
if PS2 turnLeft Is Pressed then
Set Turning Angle 160 Speed 60
if PS2 back Is Pressed then
Set Direction goback Speed 60
Set PS2 Vibration Value
if PS2 × Is Pressed then
Speed Up
if PS2 △ Is Pressed then
Speed Down
if PS2 □ Is Pressed then
Set Direction turnleft Speed 60
if PS2 ○ Is Pressed then
Set Direction turnright Speed 60
else
Set Direction keepstop Speed 0
wait 0.05 secs

Figure 7-3-2

Transfer the program of Figure 7-3-2 to the robot. We turn on the power of the robot, connect the PS2 handle to the receiver, press the button on the PS2 handle, and the robot will perform the corresponding action.

# Chapter 8 Light-finding Robots

## 8.1 Light-finding Module and Principle

The light-finding function of the light-finding module is to output analog signals by judging the light intensity of the surrounding environment through the photodiode. When the photoresistor is illuminated by strong light, its resistance value drops rapidly, the passing current increases, the resistance value of the photoresistor increases rapidly in the dark environment, and the passing current decreases. The main control board judges whether there is a light source. The tracking module and program building blocks are shown in Figures 8-1-1 and 8-1-2.

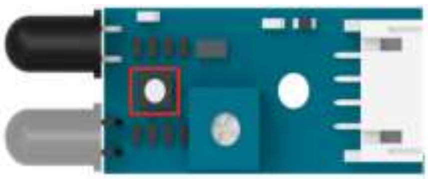


Figure 8-1-1

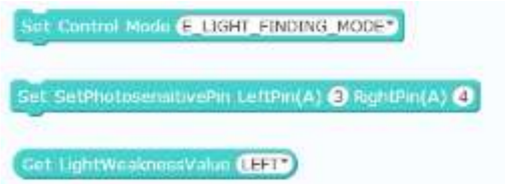



Figure 8-1-2

## 8.2 Light-fingding Program

Understanding the principle of the light-finding module and building blocks, we can build a Starterbot light-finding program, as shown in Figure 8-2-1.

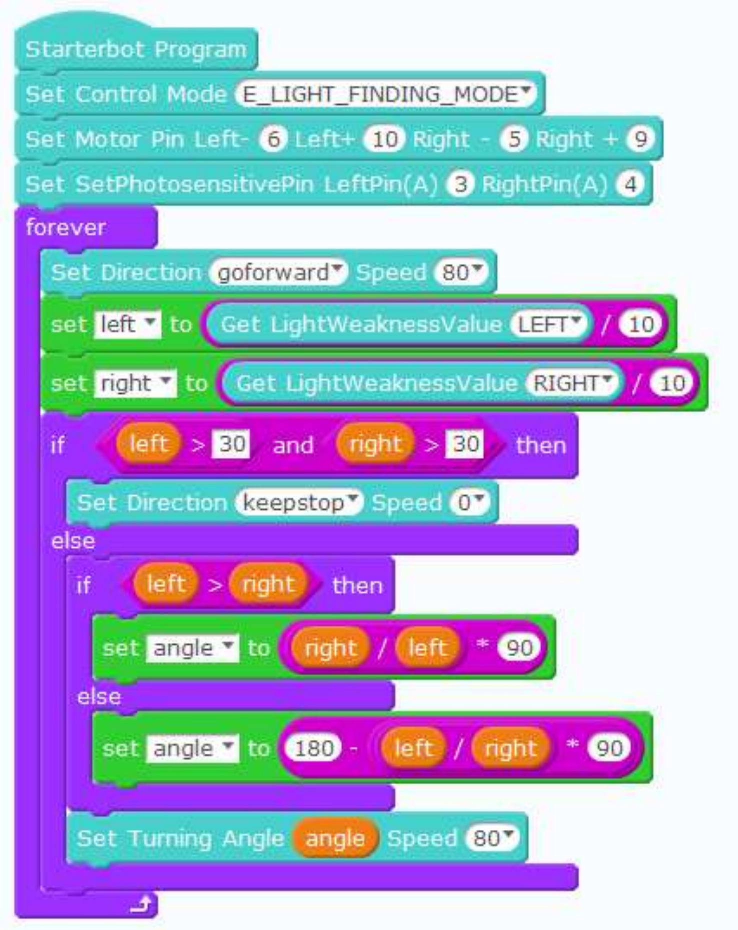

Starterbot Program
Set Control Mode E_LIGHT_FINDING_MODE
Set Motor Pin Left- 6 Left+ 10 Right - 5 Right + 9
Set SetPhotosensitivePin LeftPin(A) 3 RightPin(A) 4
forever
Set Direction goforward Speed 80
set left to Get LightWeaknessValue LEFT / 10
set right to Get LightWeaknessValue RIGHT / 10
if left > 30 and right > 30 then
Set Direction keepstop Speed 0
else
if left > right then
set angle to right / left * 90
else
set angle to 180 - left / right * 90
Set Turning Angle angle Speed 80

Figure 8-2