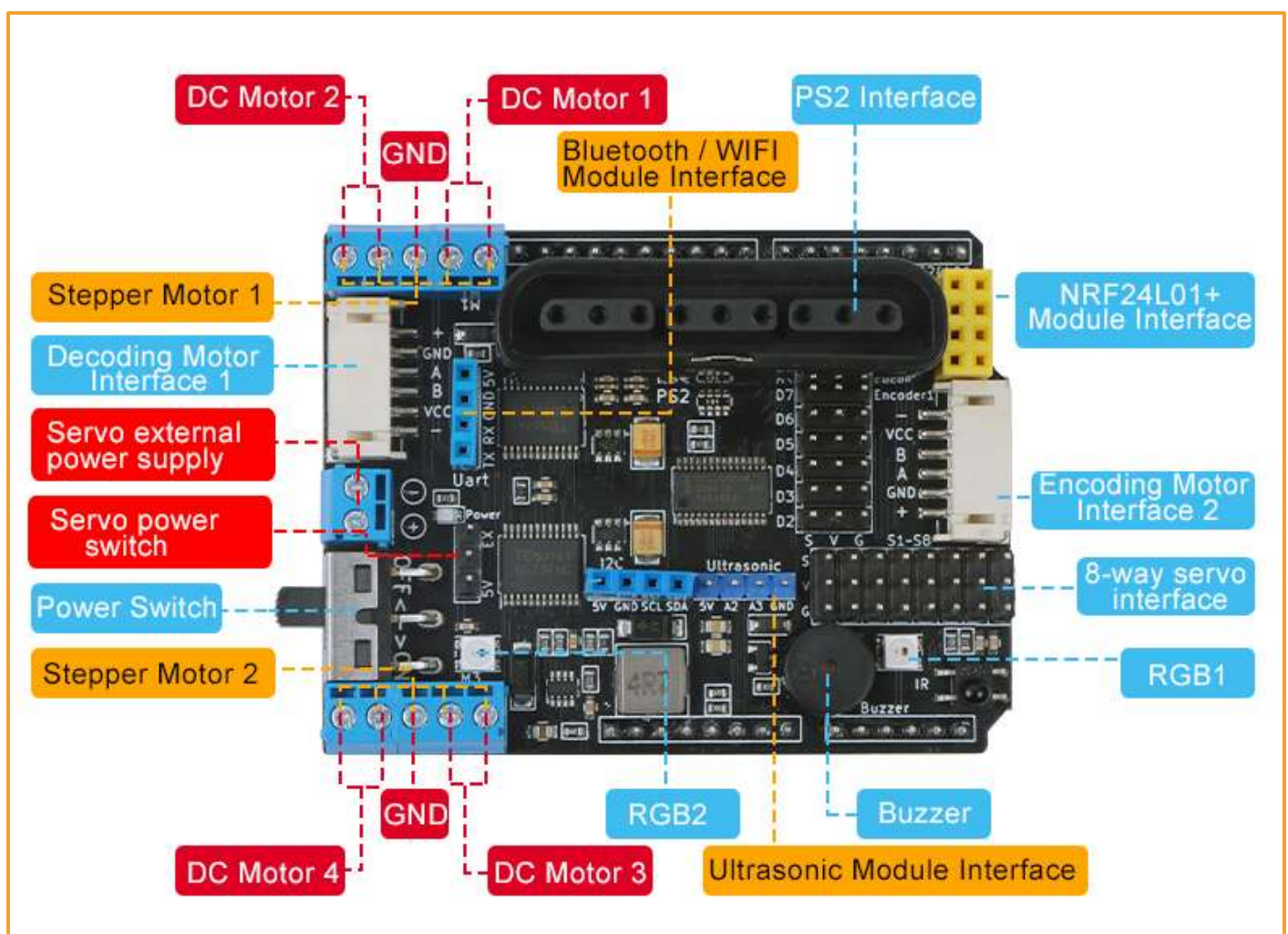# MotorDriverBoard Graphic Programming Tutorial

## MotorDriverBoard



The MotorDriverBoard can drive 4 DC motors, 2 encoder motors, 2 stepper motors, 8 servos (external power supply), and drive current up to 2A. Designed specifically for the Arduino uno R3 and Arduino mega 2560 motherboards, the driver board can be directly plugged into the Arduino Uno and Arduino mega 2560. The motherboard integrates a passive buzzer, 2 RGB LED lights, and 1 IR receiver. head. Also reserved for PS2 socket, Uart interface, I2C interface, ultrasonic obstacle avoidance module socket and other sensor interfaces, it is very convenient to externally connect various sensor modules.

# MagicBlock tutorials

## MagicBlock introduction

Magicblock is a graphic programming teaching software for steam education based on scratch3.0. It adds many common sensors and graphic programming blocks of Arduino motherboard on the basis of scratch3.0, and supports stage mode and upload mode. It can write programs for electronic hardware and show creativity through the building of graphic blocks. It is a kind of software that can let young and young people live Learn programming, quick start programming software.

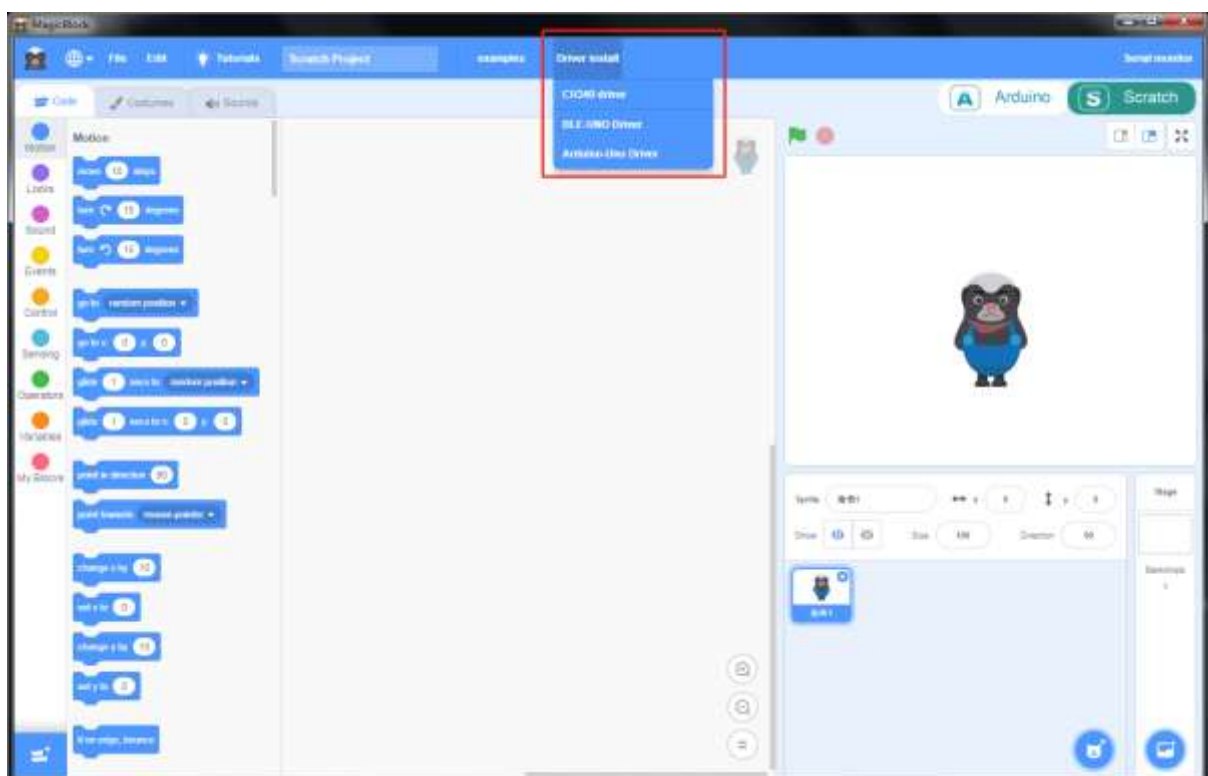## MagicBlock Download and install

Enter www.emakefun.com/download into the software download page on the official website, select magicblock for windows or Mac, and click to download



Double click the icon to install after downloading
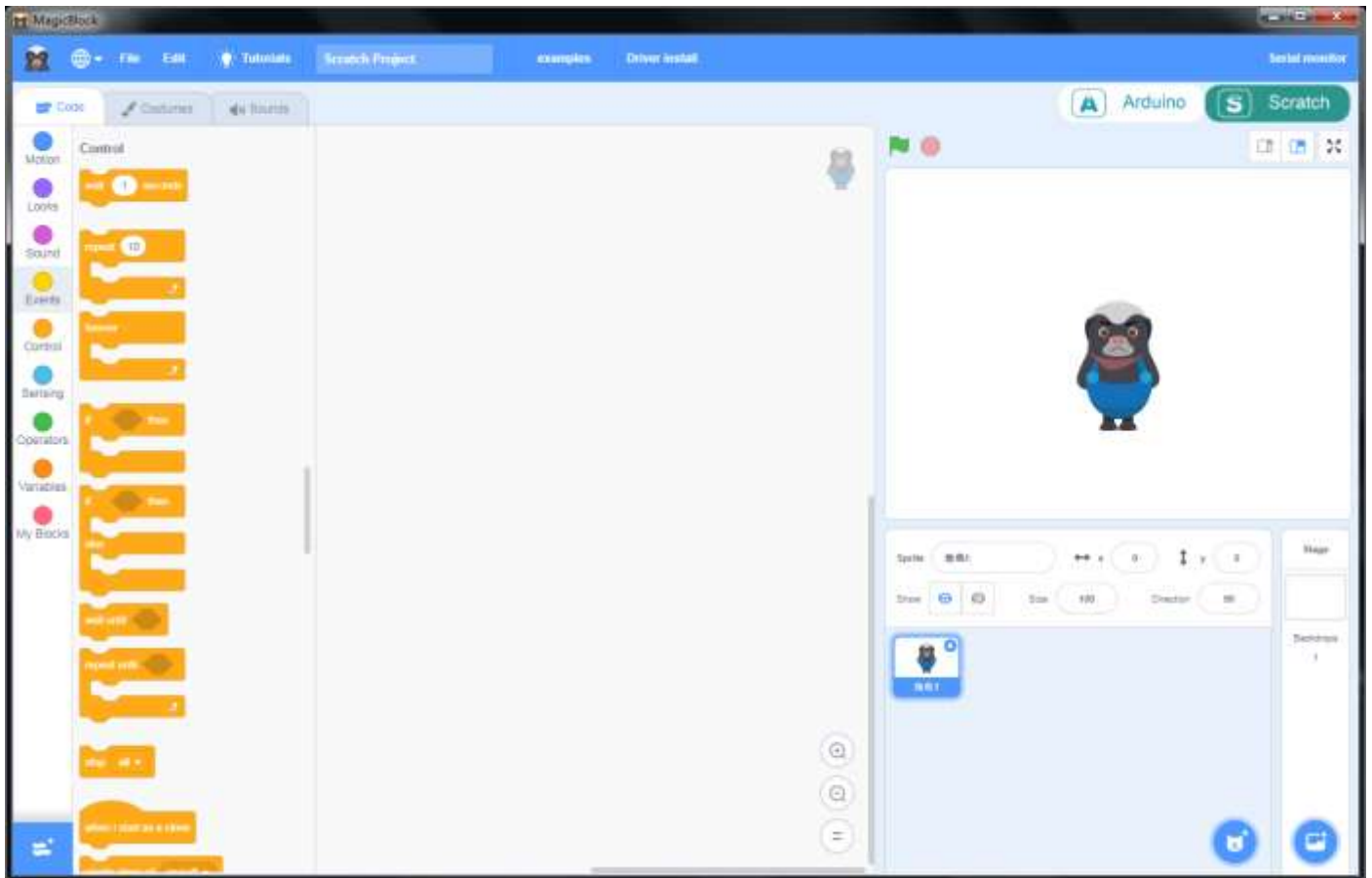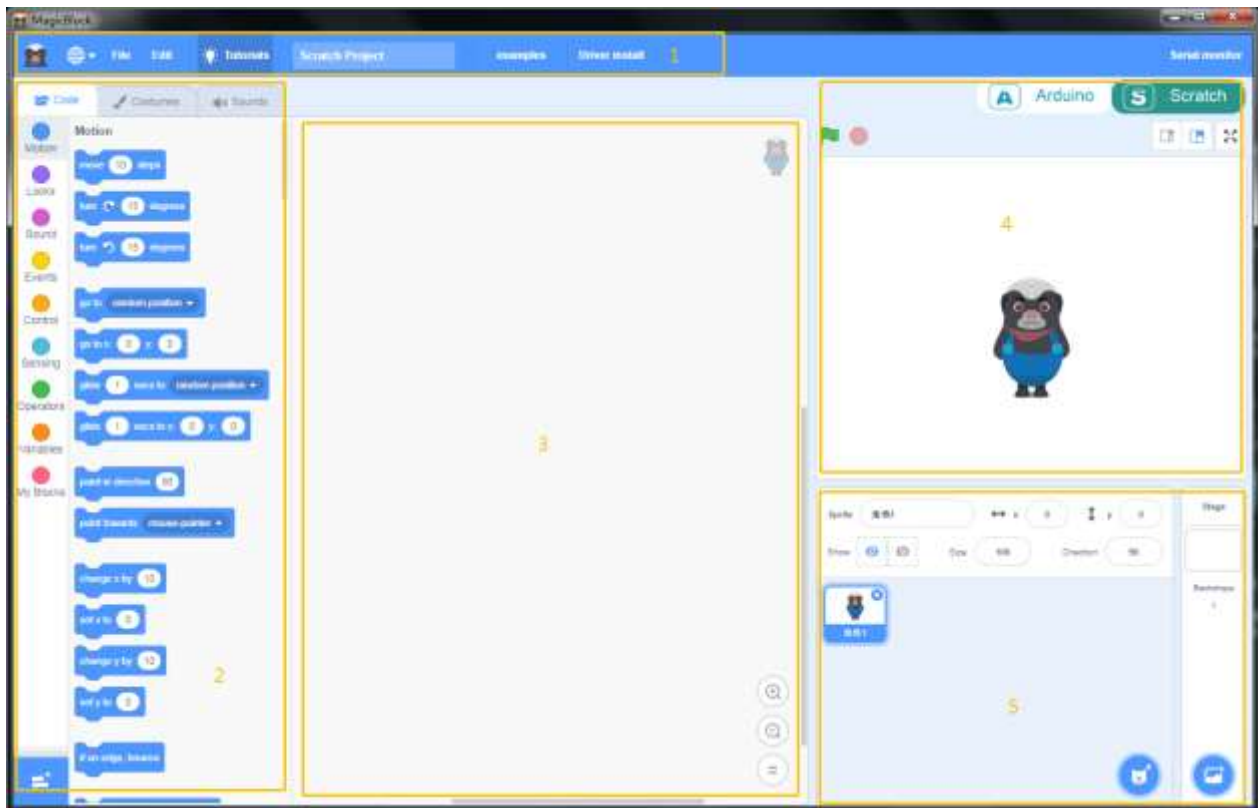
Click the setting button in the upper right corner of magicblock software, select the

driver which you want , and click Install

# MagicBlock tutorial



## Tab 1: menu bar

Language choice

Import file, save file

Tutorial learning

Naming works

View examples

Install driver

## Label 2: building block area

All blocks

Use to switch workspace columns

Add extension

## Tab 3: programming area

Drag the graphic block of the building block area here

The graph of the programming area can generate the corresponding Arduino C + + code to the code area

## Tab 4: stage area (in scratch mode) / code area (in Arduino mode)

Controlling cartoon characters by graphic block in programming area in scratch mode

You can add cartoon characters and background pictures through Annie

In the Arduino mode, the graphics of the programming area generate the corresponding Arduino C + + code into it

## Label 5: control area

Control some actions of cartoon characters

# Upload the first program for the development board in Arduino mode: light up the LED

## Choice mode

First of all, we click the Arduino button in the upper right corner to switch the mode to Arduino mode. In Arduino mode, we can connect the computer and Arduino development board through the USB cable, and then write a program to light the LED light with the graphic block of magicblock and upload it to the board
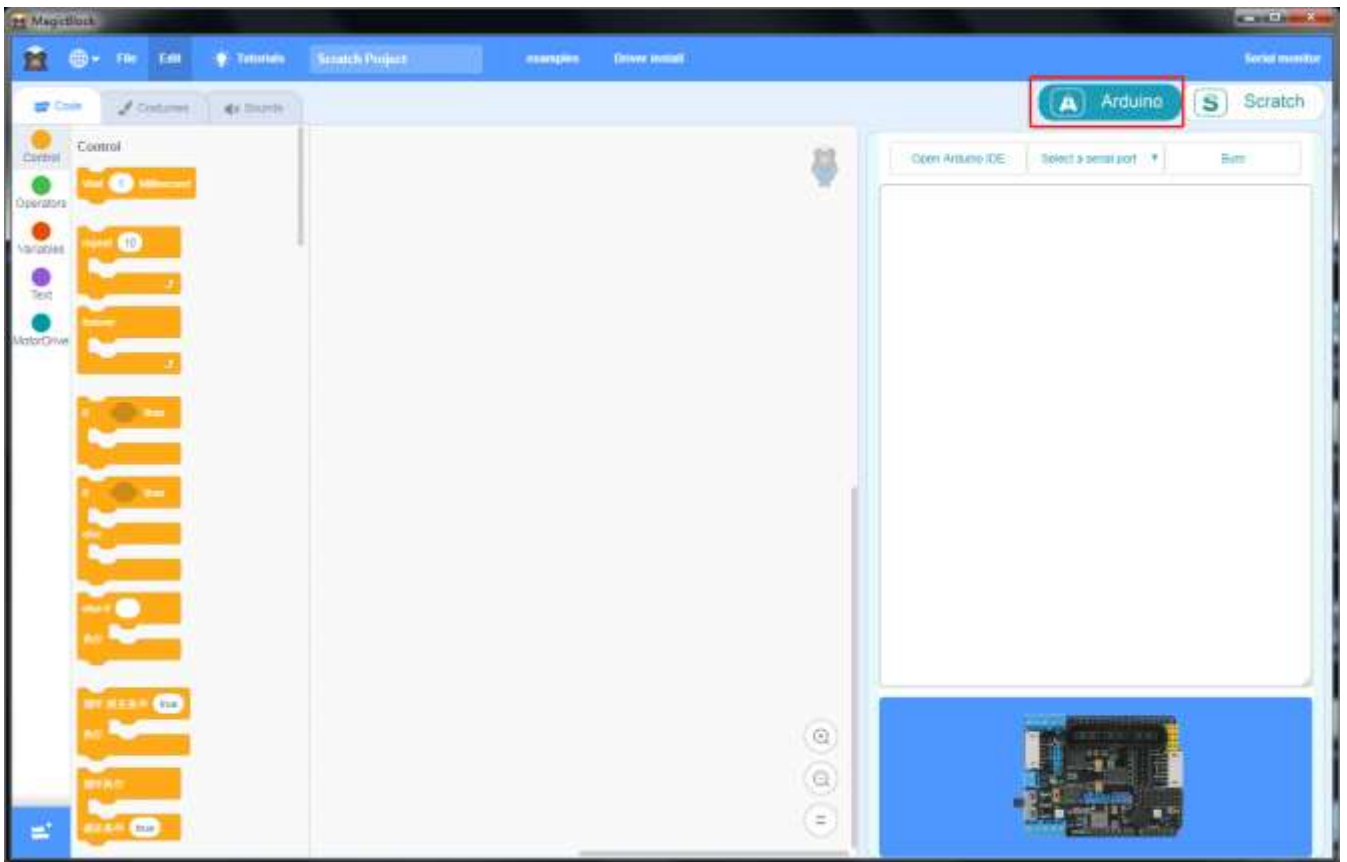
## Block description

In the control class, there are some graphic blocks used to control the execution order of the program, such as the "waiting" graphic block, which is used to delay the execution of the program. For example, if we want the LED light to be on for one second, we need this graphic block, "repeat execution" graphic block is used to make the program execute in a circular way, and we can specify the number of times to let the program execute in a circular way, "if, otherwise" graphic block is used to judge one Whether the expressions are correct or not to let the program execute
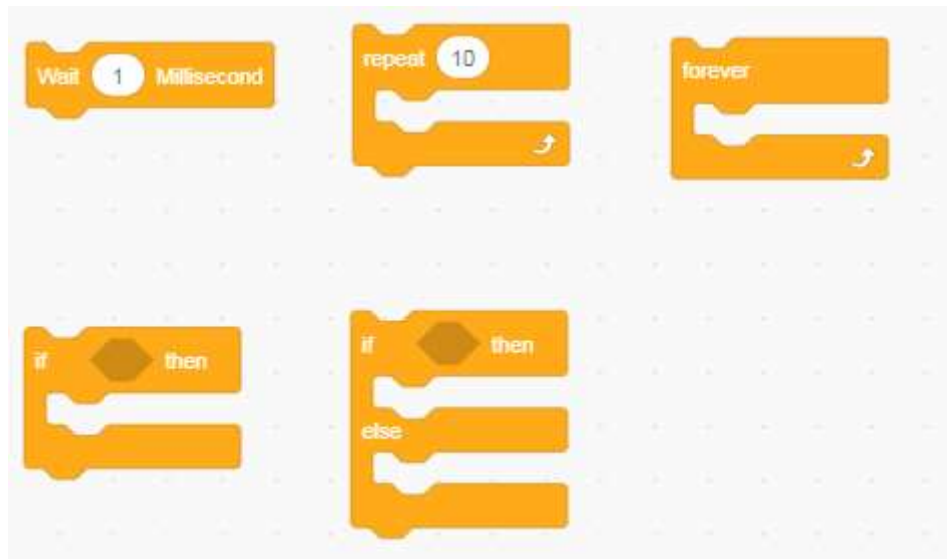
In the operation class, there are some graphic blocks for mathematical addition, subtraction, multiplication, division, size judgment and logical judgment with, or, true and false
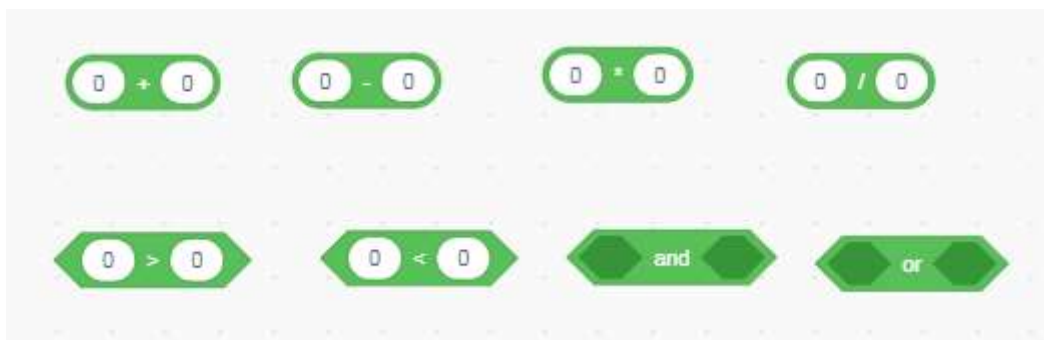


In the variable class, there are some graphic blocks used to create global or local variables and set variable values. The maximum, minimum, remainder, random number and rounding of variable values and numbers are taken
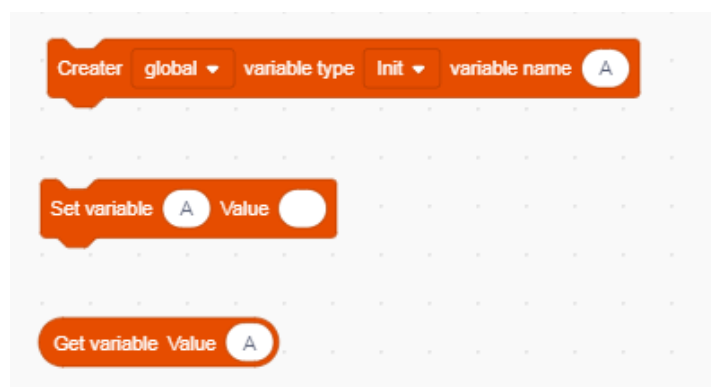
Arduino class, serial port class, sensor class, actuator class, display class graphics block: these five types of graphics block belong to Arduino graphics block, which is the main graphics block we use for Arduino board writing program

Ultrasonic HC-SR04 Initialization TrigPin  0 ▾

Ultrasonic HC-SR04 Ranging

DHT11 Temperature And Humidity Sensor Initial

DHT11 Temperature And Humidity Sensor  Get

Infrared Transmission Pin  0 ▾  Coded value

Infrared Receiving Initialization Pin  0 ▾

Infrared Received data

Value Infrared Received

Infrared Receiving Next Value

DS1302 Initialization RST Pin  0 ▾  DAT Pin

DS1302 Get  Year ▾

DS1302 Set Time  Hour  8  Minute  0  S

DS1302 Set Date  Year  2019  Month  8

LM35 Temperature Sensor Initialization Pin  0

LM35 Temperature Sensor Get Temperature Pi

DS18B20 Temperature Sensor Initialization Pin

LCD1602 Initialization Address  0x27

LCD1602 Position  Row  1  Column  1

LCD1602 Display  hello

LCD1602 Clear

4-digit Digital Tube Initialization CLK Pin  0 ▾

4-digit Digital Tube Lighting Levels  20

4-digit Digital Tube Display  1234

4-digit Digital Tube Clear

8x8MAX7219 Lattice Screen Initialization DIN

8x8MAX7219 Lattice Screen Display  0 ▾

8x8MAX7219 Lattice Screen  X Axis  0  y A

RGB Initialization Pin  0 ▾  Number Of Pixe

RGB pin  0 ▾  Pixel  1  R-Value  255  G

Initialization of Colorful LED Pin  0 ▾

## Write and upload a program

Now that we know the usage of Arduino graphics block, we can write a program to make the LED light on. We can make the LED light of pin 7 light up every 1 second for 2 seconds. The program can write this way

In the code area, we can see the Arduino code corresponding to the block program
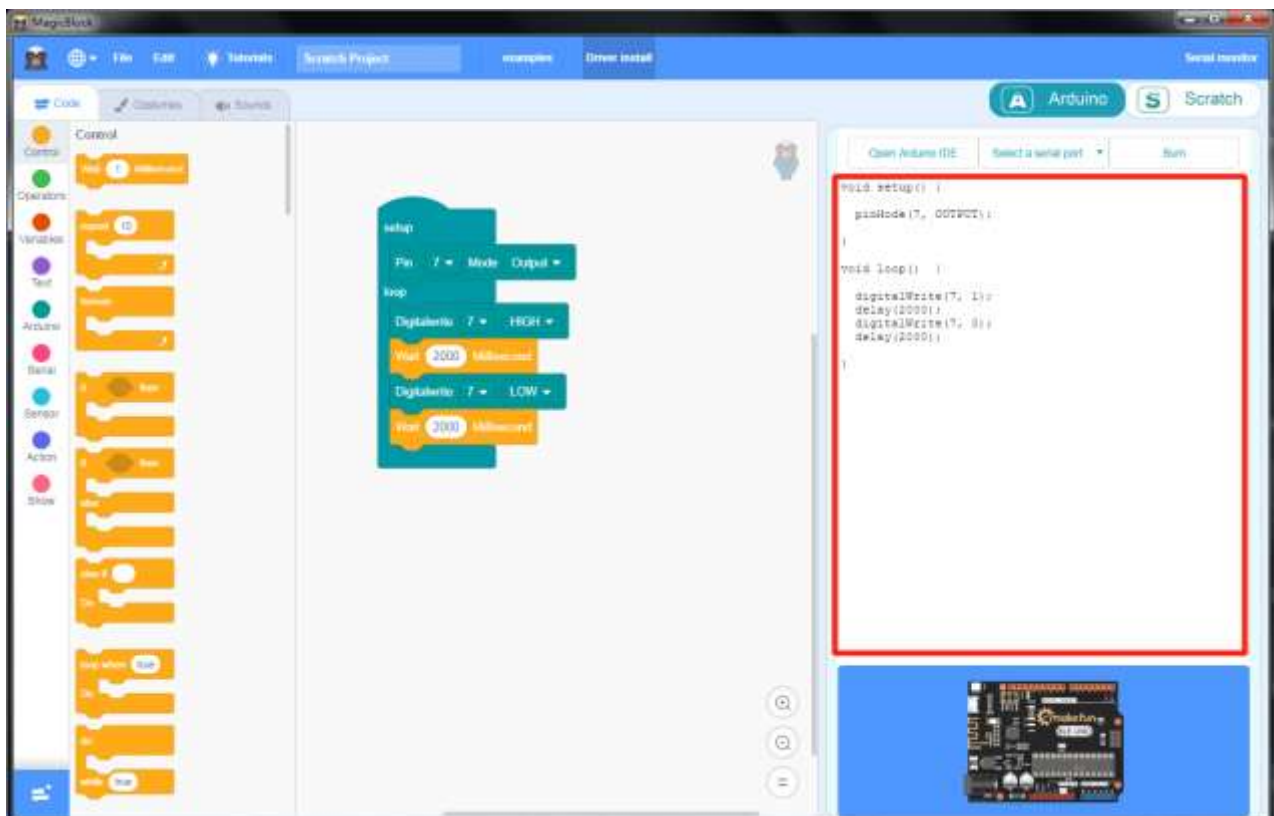


After writing the program, we need to connect the computer and Arduino board with

USB cable, and then click "please select serial port" in the code area, a com serial port

number will appear here, click "connect", then click the burn button on the right, and the

program will start burning into the board, which will take about ten seconds
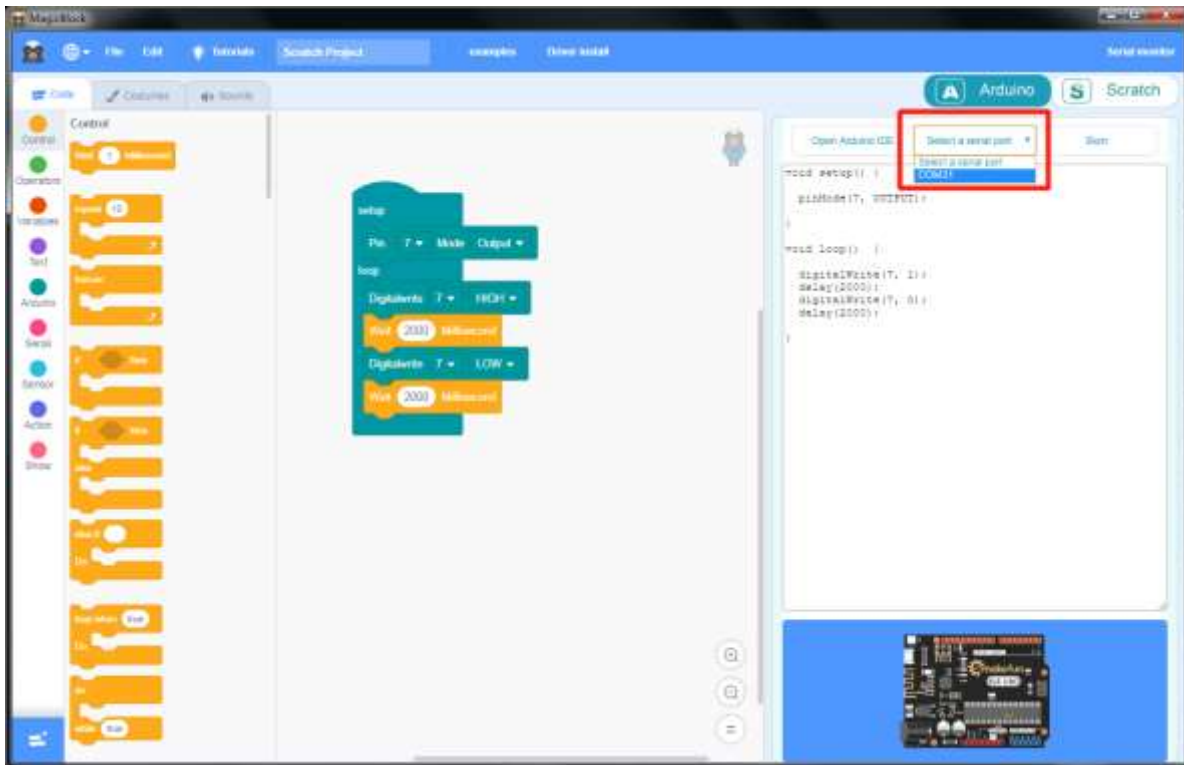


A prompt box "finish" will pop up in the upload success interface. Click "OK"
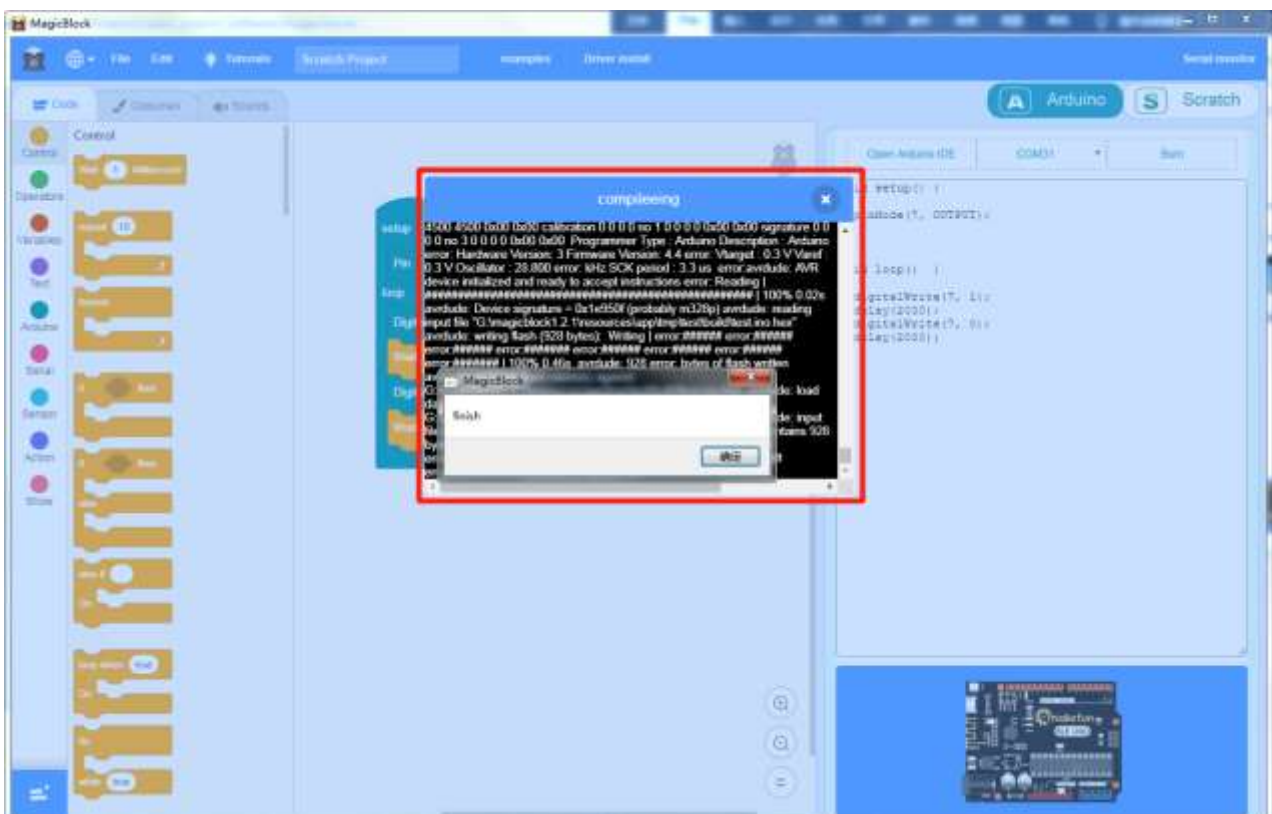
After we upload the program, connect the LED wire, power on the Arduino board, and then we can see the experimental effect of the LED light. We can see that the LED light will light up for two seconds every second, as we used magicblock to write the upload program

# Writing program for motordriverboard with MagicBlock

## Open the motordriverboard extension on magicblock

First of all, we need to switch the mode to Arduino mode, because the default mode is scratch mode when opening magicblock. We can click the Arduino button in the upper right corner, and then click the blue button in the lower left corner of the interface to add the extension
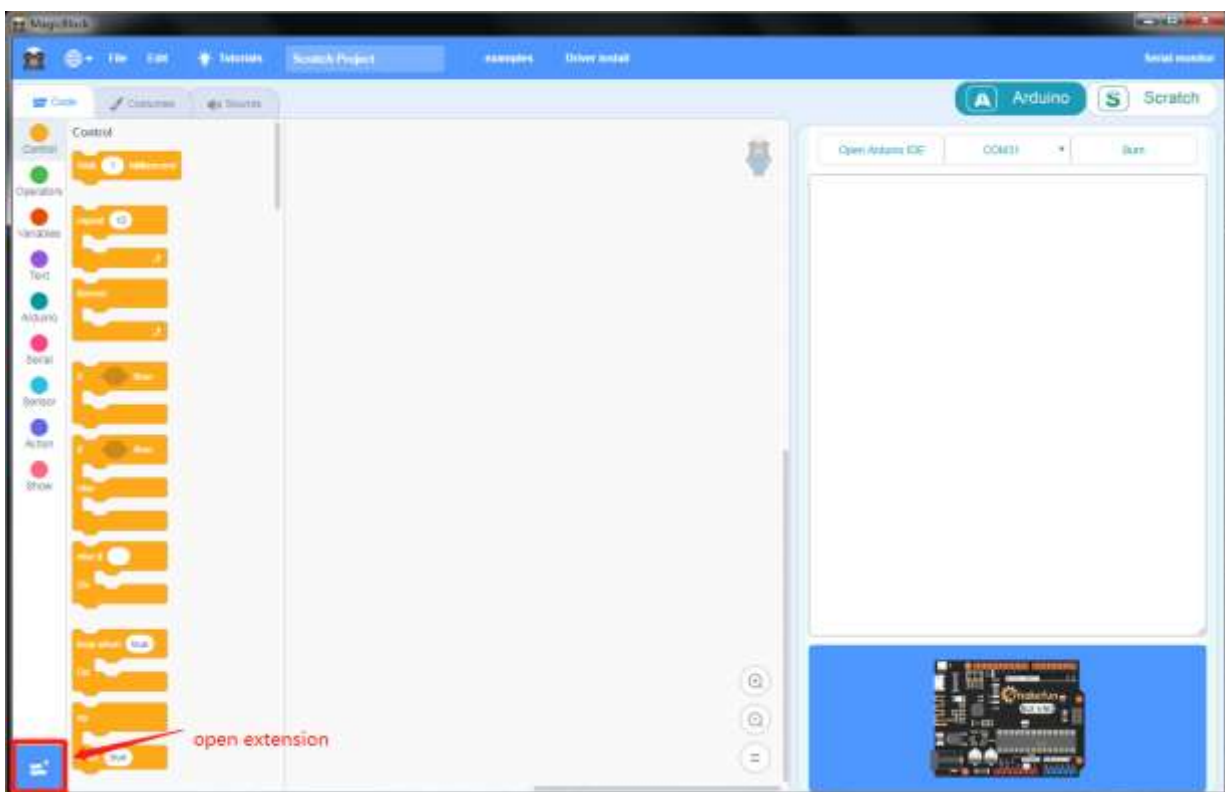


Select extension interface appears, left click to select motordriverboard extension
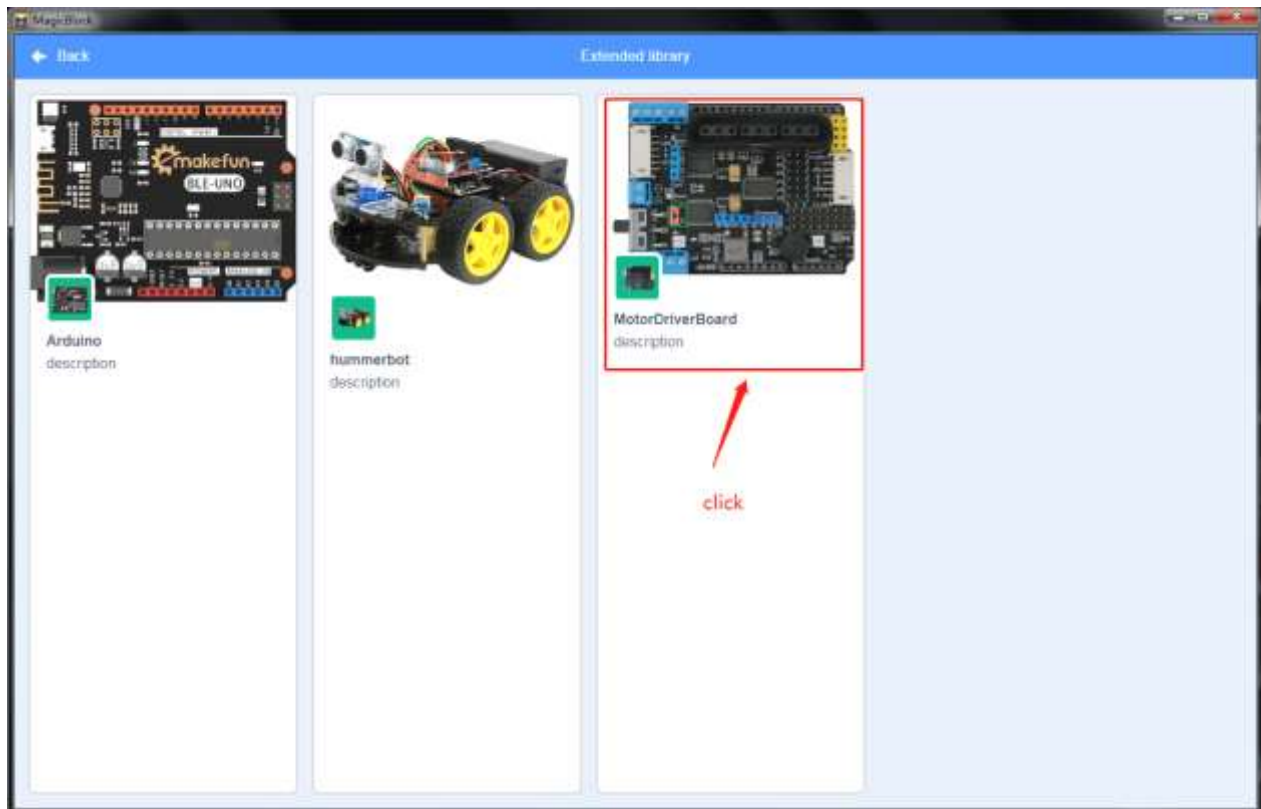
After selecting the MotorDriverBoard extension, the hummerbot extension will appear

in our graphics block area. It contains all the graphics blocks used for programming by

MotorDriverBoard
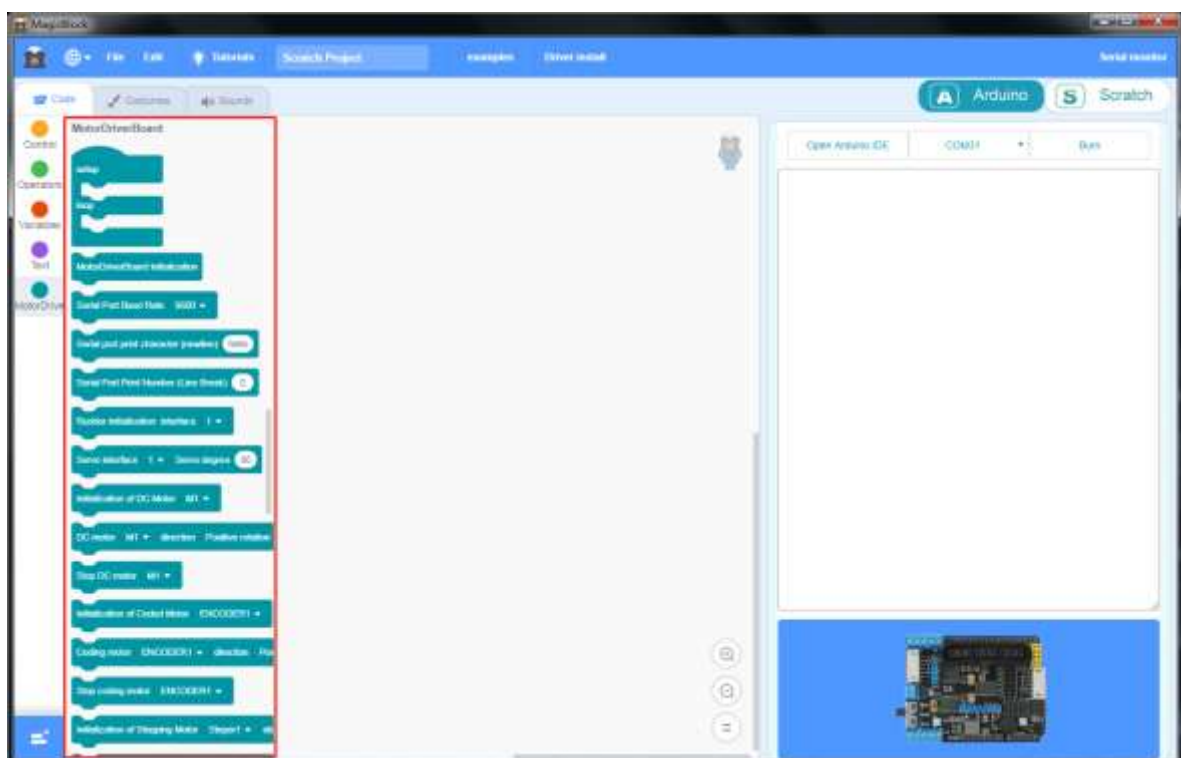
# How to use the MotorDriverBoard blocks

The first graphics block of the motordriverboard is the setup / loop graphics block, which is the main graphics block. Every program of the motordriverboard needs to use it first, and other graphics blocks of the motordriverboard need to be put into the setup or loop of this graphics block. The setup is the initialization graphics blocks, such as DC motor initialization, RGB initialization, etc., which only execute one The loop needs to put some graphics blocks that may be executed many times, such as DC motor forward and reverse, setting RGB light color, setting the sound played by buzzer, etc



# Introduction to the module that can be driven by motordriverboard and sample program of magicblock

## DC motor

### Driving DC motor

The motordriverboard has four DC motor interfaces, namely, DC motor interface M1, DC motor interface m2, DC motor interface m3 and DC motor interface M4. The DC motor can be directly connected to the driver through the terminal.
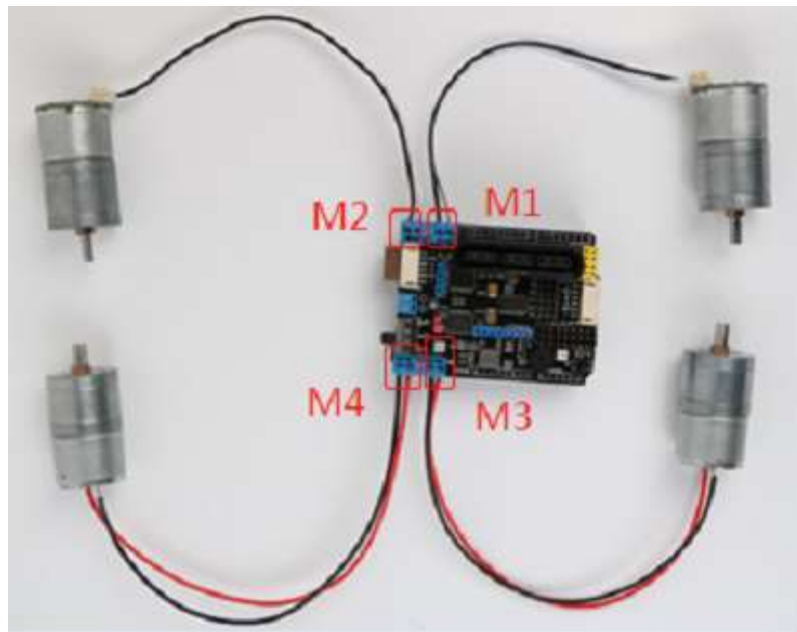
# Example program

## Stepper motor

## Driving Stepper motor

The motordriverboard has two ports for the stepping motor. The working voltage of

the stepping motor is between 5-12v. If the working voltage of the stepping motor is too

low or too high, the motor will burn down or the driving board will collapse. In addition, when driving the motor, the battery power should be sufficient
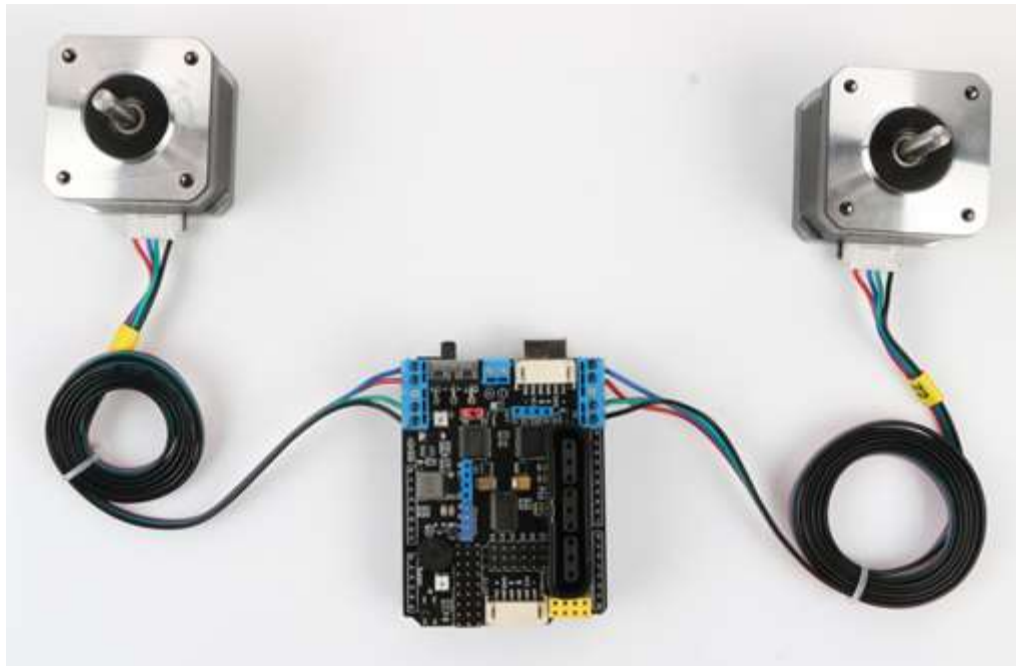


Example program

## Coding motor

## Driving Coding motor

The motordriverboard has two interfaces for encoding motor, and the working voltage of encoding motor is between 5-12V

# Example program
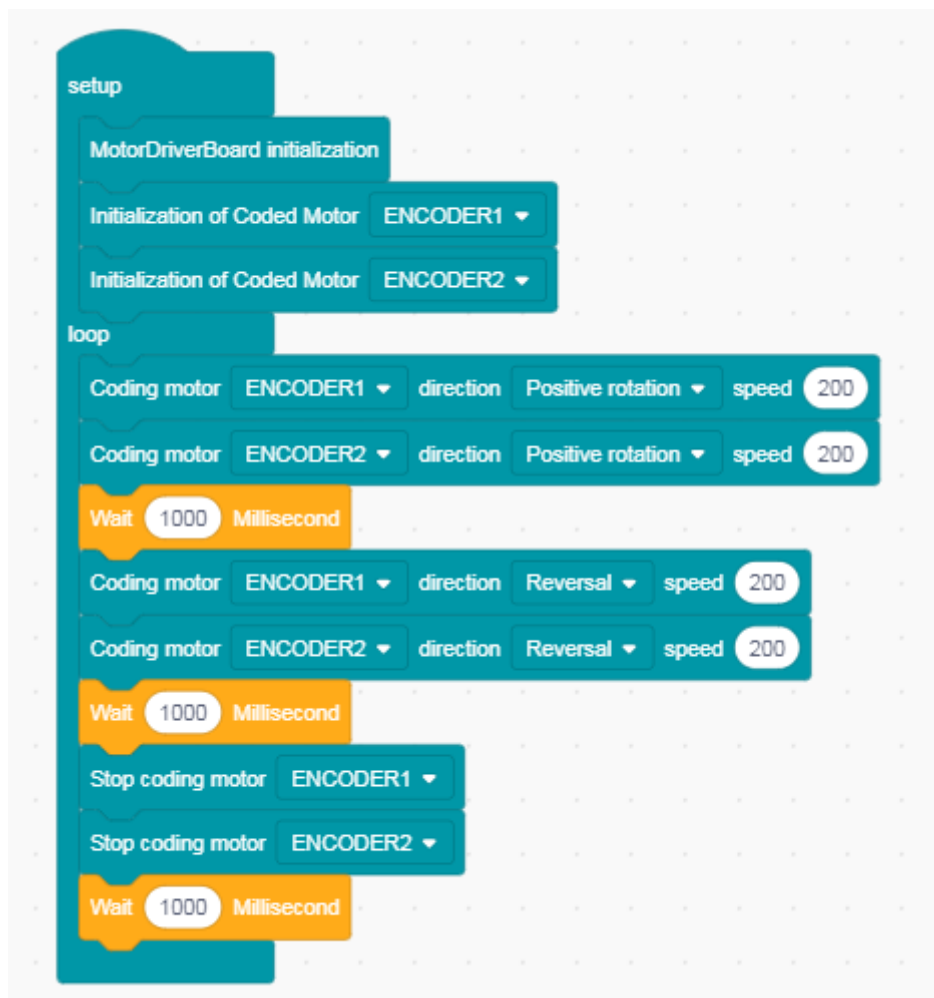
## RGB

### Introduction to RGB ws2812

Ws2812 RGB LED lamp is a three channel driving control system, which contains the driving circuit of data lock signal shaping and amplification of intelligent digital interface, high-precision internal oscillator and 15V high-voltage programmable constant current output driver. At the same time, in order to reduce the power ripple, three channels have a certain delay conduction function, so that when the frame is refreshed, it is easier to reduce the circuit ripple installation. The ws2812 RGB LED lamp is different from the traditional lamp. The driver control chip is integrated inside the lamp. Only one signal line is needed to control one lamp bead or multiple modules

# Example program



## Buzzer

## Driving Buzzer

There is a passive buzzer on the driver board of the motordriverboard, which can be

programmed to play prompt tone or music

# Example program



## Servo

## Driving Servo

The driving board of the motordriverboard can drive 8-way steering gear. The position of the steering gear pin on the driving board (row pin above the infrared receiving head), set the ground pin (g), power pin (V) and signal pin (s). The three pins are respectively corresponding to the corresponding pin of the steering gear. The steering gear can also be controlled through I2C communication
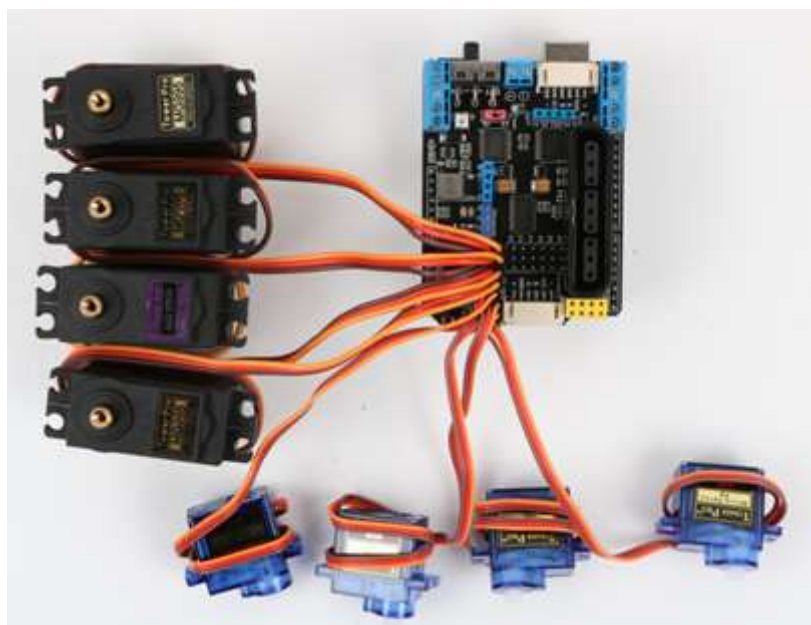


Example program
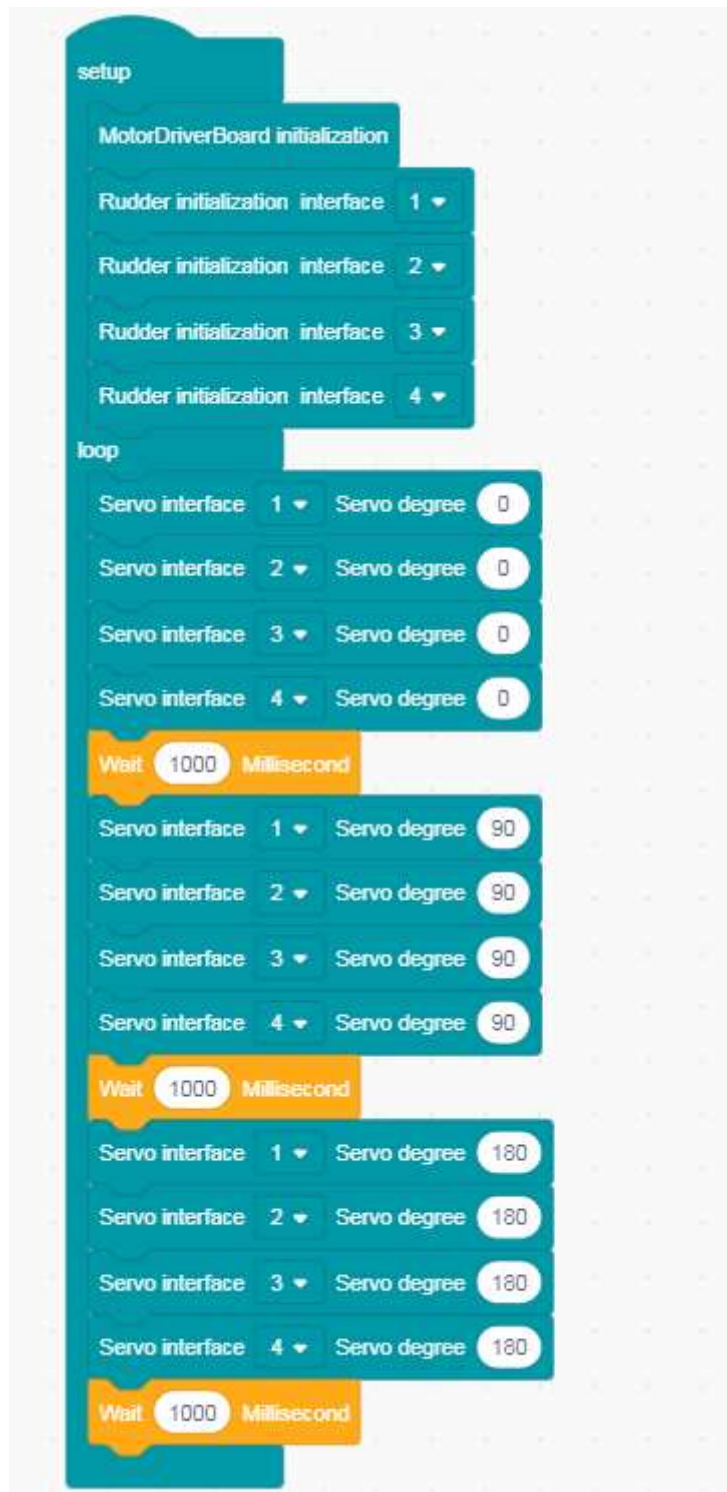
## Ultrasonic module

### Driving Ultrasonic module

There are four ultrasonic module interfaces on the driving board of the

motordriverboard. The four pins are respectively the power supply pin (VCC), the

ultrasonic signal transmitting pin (A2), the ultrasonic signal receiving pin (A3), the ground wire (GND). The four pins are respectively connected with the corresponding pins of the ultrasonic module
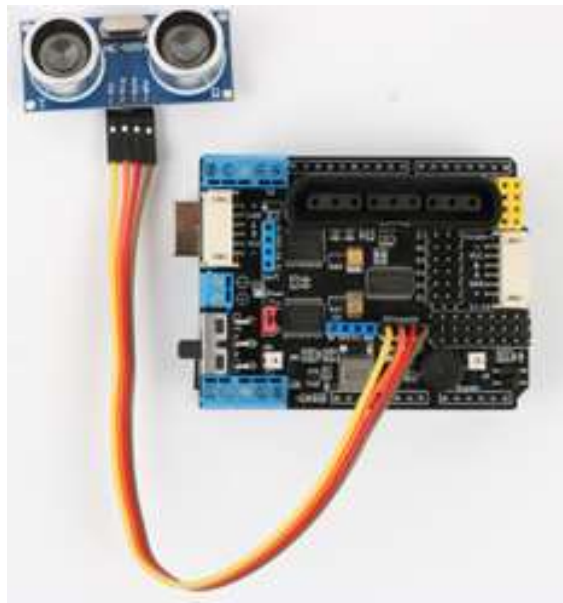


# Example program
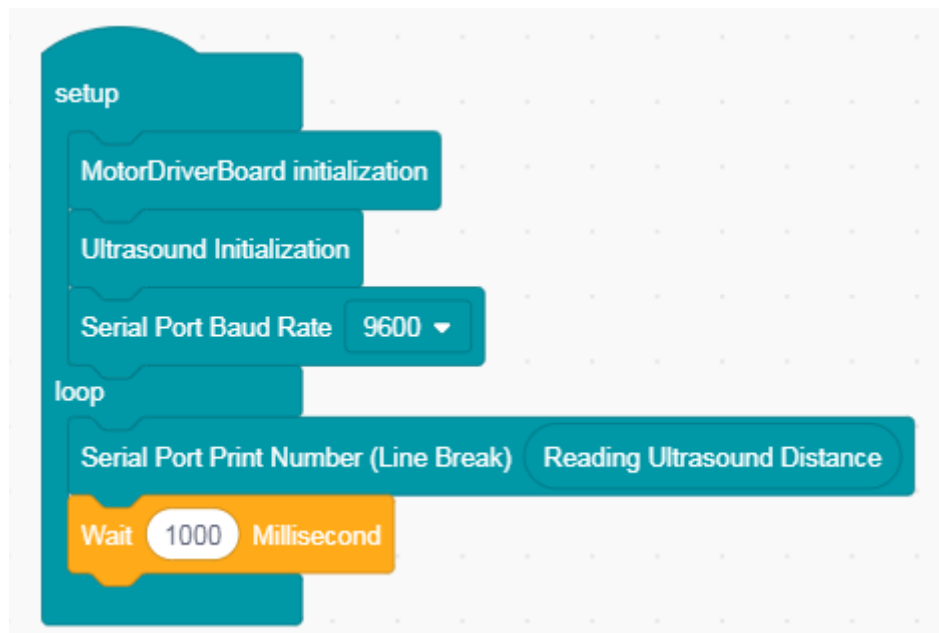


# Infrared remote

# Driving Infrared remote

There is an infrared remote control receiving probe on the driving board of the motordriverboard. When it is used, it only needs to insert the expansion board into Arduino. When the infrared coding signal is transmitted, it is processed by the infrared connector, and the square wave signal after detection and shaping is output, which is directly provided to the single chip microcomputer, and the corresponding operation is performed to achieve the purpose of controlling the motor.





Example program
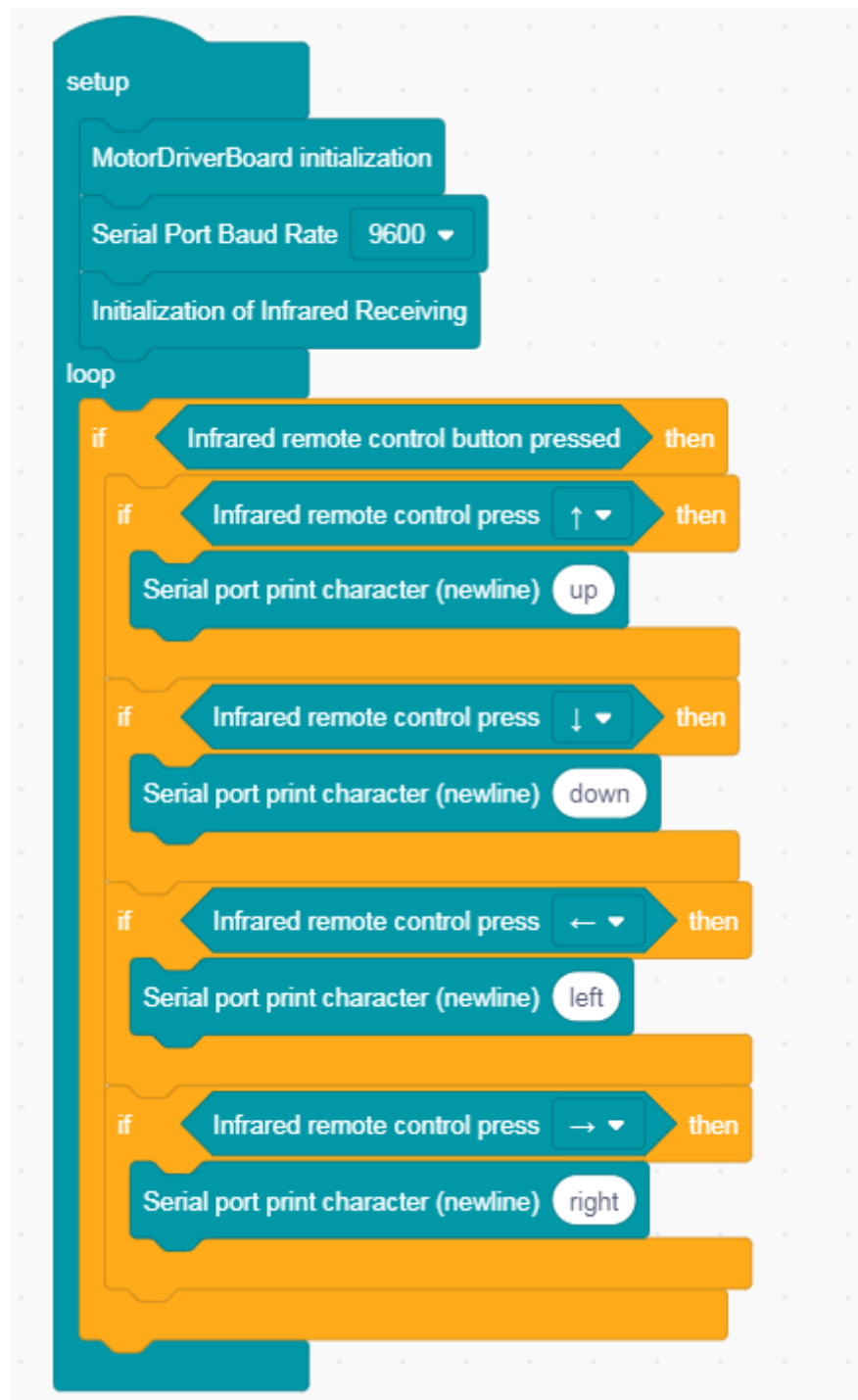
## PS2

## Driving PS2

There is a PS2 port on the driving board of the motordriverboard, which can directly plug the PS2 infrared receiving head into the PS2 port. The PS2 handle consists of two parts: the handle and the receiver. The handle needs two sections of 7 1.5V power supply.

The receiver's power supply and Arduino use the same power supply. The power supply

range is 3-5v, which can not be reversed or over-voltage. Overvoltage and reverse

connection will burn the receiver. There is a power switch on the handle, on / off. Turn the

handle switch to on. When the receiver is not searched, the light on the handle will flash

continuously. If the receiver is not searched in a certain period of time, the handle will

enter the standby mode and the light on the handle will be extinguished. Press the "start"

key to wake up the handle



Example program

nRF24L01+

Driving nRF24L01+

NRF24L01 + module is a 2.4G wireless communication module developed by Nordic based on nRF24L01 chip. With FSK modulation, Nordic's enhanced short burst protocol is internally integrated. It can realize point-to-point or 1-to-6 wireless communication. The wireless communication speed can reach up to 2m (BPS). NRF24L01 has four working modes, i.e. transceiver mode, configuration mode, idle mode and shutdown mode. There is an nRF24L01 interface on the motordriverboard drive board
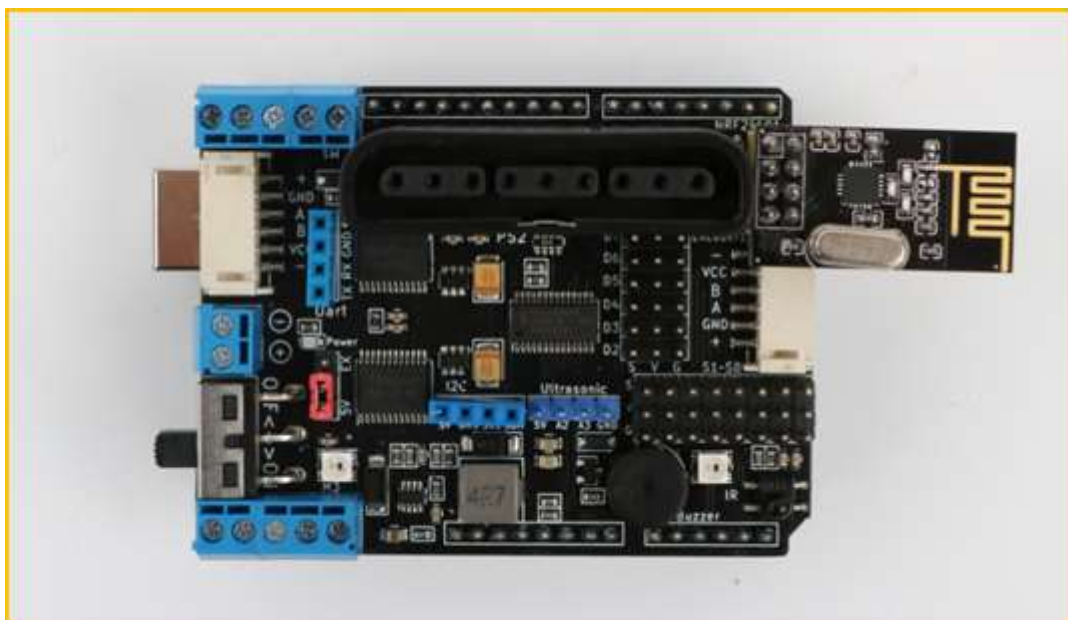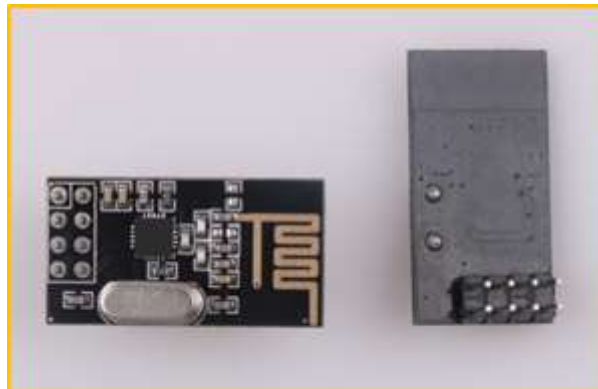




Example program
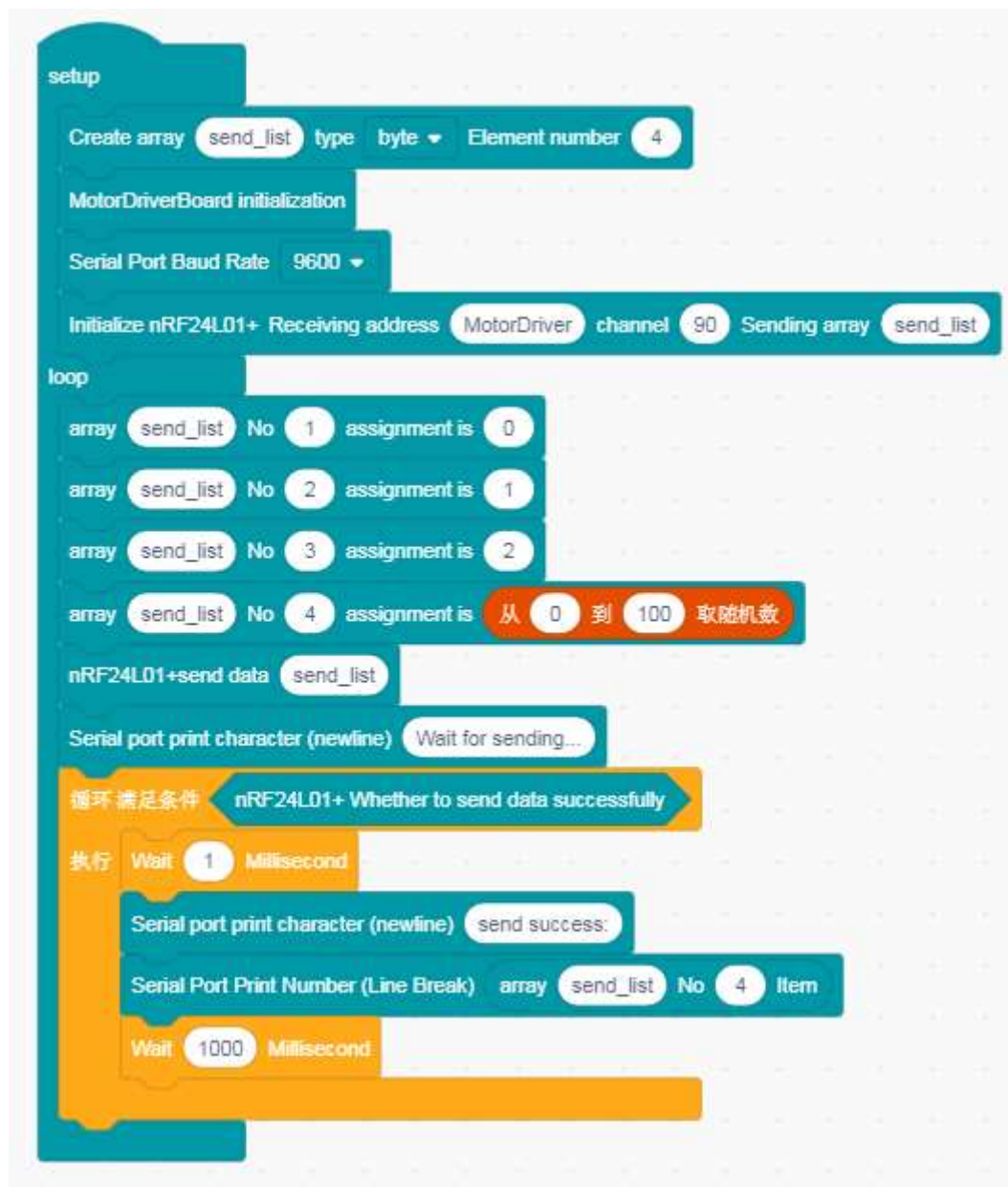
Send
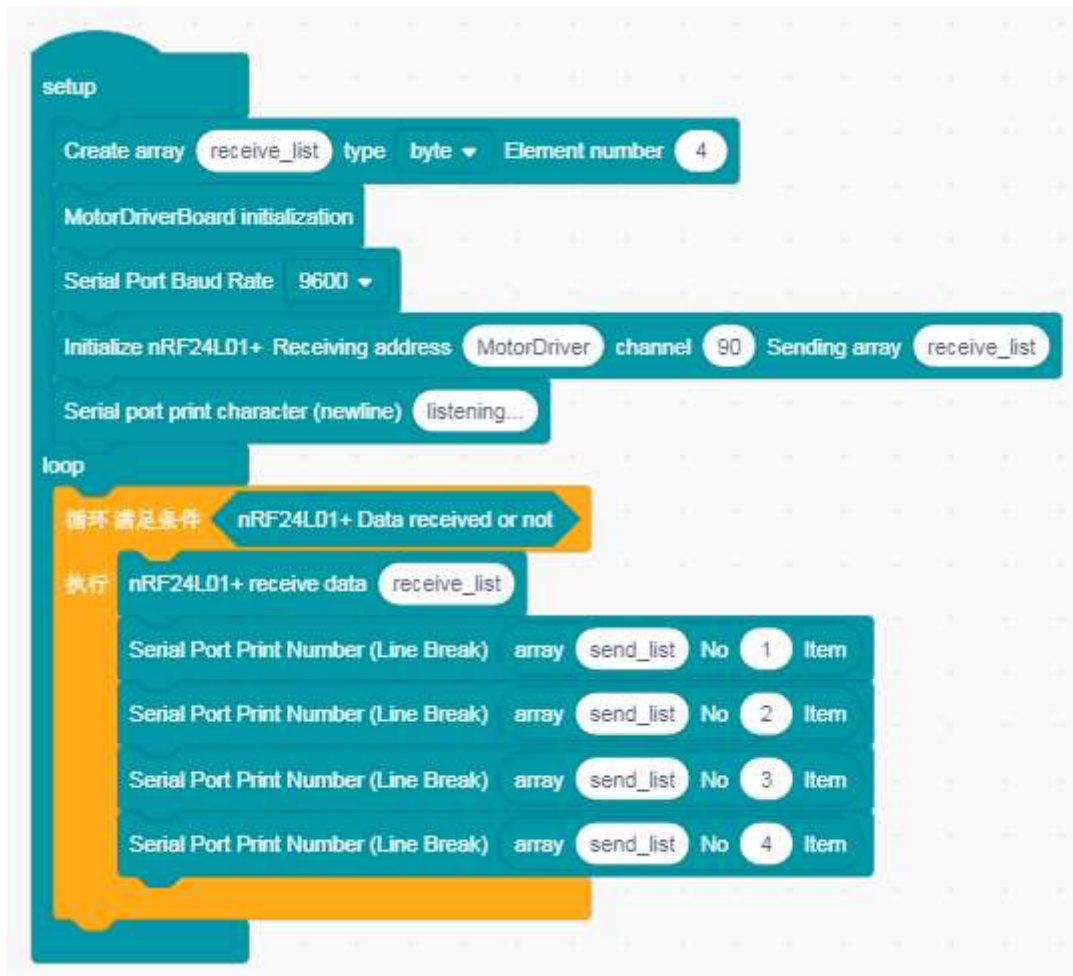
Receive

# Classic applications

In order to make it convenient for you to program your own 4WD car with magicblock, we have added the graphic block of initializing the 4WD car and the sample programs of PS2 controlling the 4WD car, nRF24L01 + controlling the 4WD car, infrared remote controlling the 4WD car and Bluetooth controlling the 4WD car in magicblock. For those who have hobbies and needs in this regard, please refer to the following classic applications

## PS2 controls 4WD car

The PS2 graphical programming of motordriverboard has been introduced in the way of printing PS2 keys through serial port. Now let's expand the function of PS2 to operate a four-wheel drive car driven by motordriverboard. Here's an example program for reference. This program controls the direction and speed of the four-wheel drive car through the eight keys of PS2

```
setup

MotorDriverBoard initialization

4WD bot initialization Motor Interface  Left front  M2 ▼   Right front  M1 ▼   Left rear  M4 ▼   Right rear  M3 ▼

Creater  global ▼  variable type  Init ▼  variable name  speed

Set variable  speed  Value  100

RGB initialization

PS2 initialization

loop

Read PS2 status

if  PS2 key pressed  then

    if  PS2 press  up ▼  then

        RGB  all ▼  white ▼  brightness (0~255)  255

        DC motor  M2 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed

        DC motor  M1 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed

        DC motor  M4 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed

        DC motor  M3 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed

    if  PS2 press  down ▼  then

        RGB  all ▼  red ▼  brightness (0~255)  255

        DC motor  M2 ▼  direction  Reversal ▼  speed  Get variable Value  speed

        DC motor  M1 ▼  direction  Reversal ▼  speed  Get variable Value  speed

        DC motor  M4 ▼  direction  Reversal ▼  speed  Get variable Value  speed

        DC motor  M3 ▼  direction  Reversal ▼  speed  Get variable Value  speed

    if  PS2 press  left ▼  then

        RGB  all ▼  red ▼  brightness (0~255)  255

        DC motor  M2 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed / 2

        DC motor  M1 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed
```

```
DC motor  M4 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed  /  2
DC motor  M3 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed

if  PS2 press  right ▼  then
    RGB  all ▼  red ▼  brightness (0~255)  255
    DC motor  M2 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed
    DC motor  M1 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed  /  2
    DC motor  M4 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed
    DC motor  M3 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed  /  2

if  PS2 press  △ ▼  then
    Set variable  A  Value  Get variable Value  speed  +  5

if  PS2 press  × ▼  then
    Set variable  A  Value  Get variable Value  speed  -  5

if  PS2 press  □ ▼  then
    RGB  all ▼  yellow ▼  brightness (0~255)  255
    DC motor  M2 ▼  direction  Reversal ▼  speed  Get variable Value  speed
    DC motor  M1 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed
    DC motor  M4 ▼  direction  Reversal ▼  speed  Get variable Value  speed
    DC motor  M3 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed

if  PS2 press  □ ▼  then
    RGB  all ▼  purple ▼  brightness (0~255)  255
    DC motor  M2 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed
    DC motor  M2 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed
    DC motor  M1 ▼  direction  Reversal ▼  speed  Get variable Value  speed
    DC motor  M4 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed
    DC motor  M3 ▼  direction  Reversal ▼  speed  Get variable Value  speed

else
    Stop DC motor  M2 ▼
    Stop DC motor  M1 ▼
    Stop DC motor  M4 ▼
    Stop DC motor  M3 ▼
    RGB  all ▼  off ▼  brightness (0~255)  255

Wait  50  Millisecond
```

## PS2 control mechanical arm 4WD car

The motordriverboard supports 8-way steering gear. We use 4 steering gear for the robot arm matched with the motordriverboard. The program principle of PS2 controlling the robot arm 4WD car is to add the program of using the rocker to control the steering gear on the program of PS2 controlling the 4WD car, realizing 8 keys to control the driving direction and speed of the 4WD car. The left rocker controls the left and right movement of the robot arm, and the right rocker controls the machinery When the arm moves up and down, the first side key on the left is to close the pliers of the mechanical arm, and the first side key on the right is to open the pliers

## Infrared remote control 4WD car

We have written the test program of infrared remote control. Now we can make a program of infrared remote control to control the 4WD car. We can control the direction of the 4WD car up, down, left and right, a left spin, C right spin, + acceleration, - deceleration

```
setup
  MotorDriverBoard initialization
  Initialization of Infrared Receiving
  4WD bot initialization Motor Interface  Left front  M2 ▼  Right front  M1 ▼  Left rear  M4 ▼  Right rear  M3 ▼
  Creater  global ▼  variable type  Init ▼  variable name  speed
  Set variable  speed  Value  100
loop
  if  Infrared remote control button pressed  then
    if  Infrared remote control press  ↑ ▼  then
      DC motor  M2 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed
      DC motor  M1 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed
      DC motor  M4 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed
      DC motor  M3 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed

    if  Infrared remote control press  ↓ ▼  then
      DC motor  M2 ▼  direction  Reversal ▼  speed  Get variable Value  speed
      DC motor  M1 ▼  direction  Reversal ▼  speed  Get variable Value  speed
      DC motor  M4 ▼  direction  Reversal ▼  speed  Get variable Value  speed
      DC motor  M3 ▼  direction  Reversal ▼  speed  Get variable Value  speed

    if  Infrared remote control press  ← ▼  then
      DC motor  M2 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed  / 2
      DC motor  M1 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed
      DC motor  M4 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed  / 2
      DC motor  M3 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed

    if  Infrared remote control press  → ▼  then
      DC motor  M2 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed
```

## Bluetooth control 4WD car

The motordriverboard motor driver board is equipped with a Bluetooth socket. When the Bluetooth module is inserted into the socket, you can use the Bluetooth connection of the mobile phone. First, we download the emakefun mobile app, open the app, and select a four-wheel drive car such as hummerbot for Bluetooth connection. After we need to

write a Bluetooth control program for the driver board, you can enter the app Bluetooth

control interface to control four Drove

```
setup
    MotorDriverBoard initialization
    Serial Port Baud Rate   9600 ▾
    Bluetooth initialization
    Creater   global ▾   variable type   Init ▾   variable name   speed
    Set variable   speed   Value   100
    4WD bot initialization Motor Interface  Left front   M2 ▾   Right front   M1 ▾   Left rear   M4 ▾   Right rear   M3 ▾
loop
    Receive Bluetooth data
    if    Bluetooth data received    then
        if    Bluetooth data type is   Key ▾    then
            if    Bluetooth press button   ↑ ▾    then
                DC motor   M2 ▾   direction   Positive rotation ▾   speed   Get variable Value   speed
                DC motor   M1 ▾   direction   Positive rotation ▾   speed   Get variable Value   speed
                DC motor   M4 ▾   direction   Positive rotation ▾   speed   Get variable Value   speed
                DC motor   M3 ▾   direction   Positive rotation ▾   speed   Get variable Value   speed

            if    Bluetooth press button   ↓ ▾    then
                DC motor   M2 ▾   direction   Reversal ▾   speed   Get variable Value   speed
                DC motor   M1 ▾   direction   Reversal ▾   speed   Get variable Value   speed
                DC motor   M4 ▾   direction   Reversal ▾   speed   Get variable Value   speed
                DC motor   M3 ▾   direction   Reversal ▾   speed   Get variable Value   speed

            if    Bluetooth press button   ← ▾    then
                DC motor   M2 ▾   direction   Positive rotation ▾   speed   Get variable Value   speed / 2
                DC motor   M1 ▾   direction   Positive rotation ▾   speed   Get variable Value   speed
                DC motor   M4 ▾   direction   Positive rotation ▾   speed   Get variable Value   speed / 2
                DC motor   M3 ▾   direction   Positive rotation ▾   speed   Get variable Value   speed
```
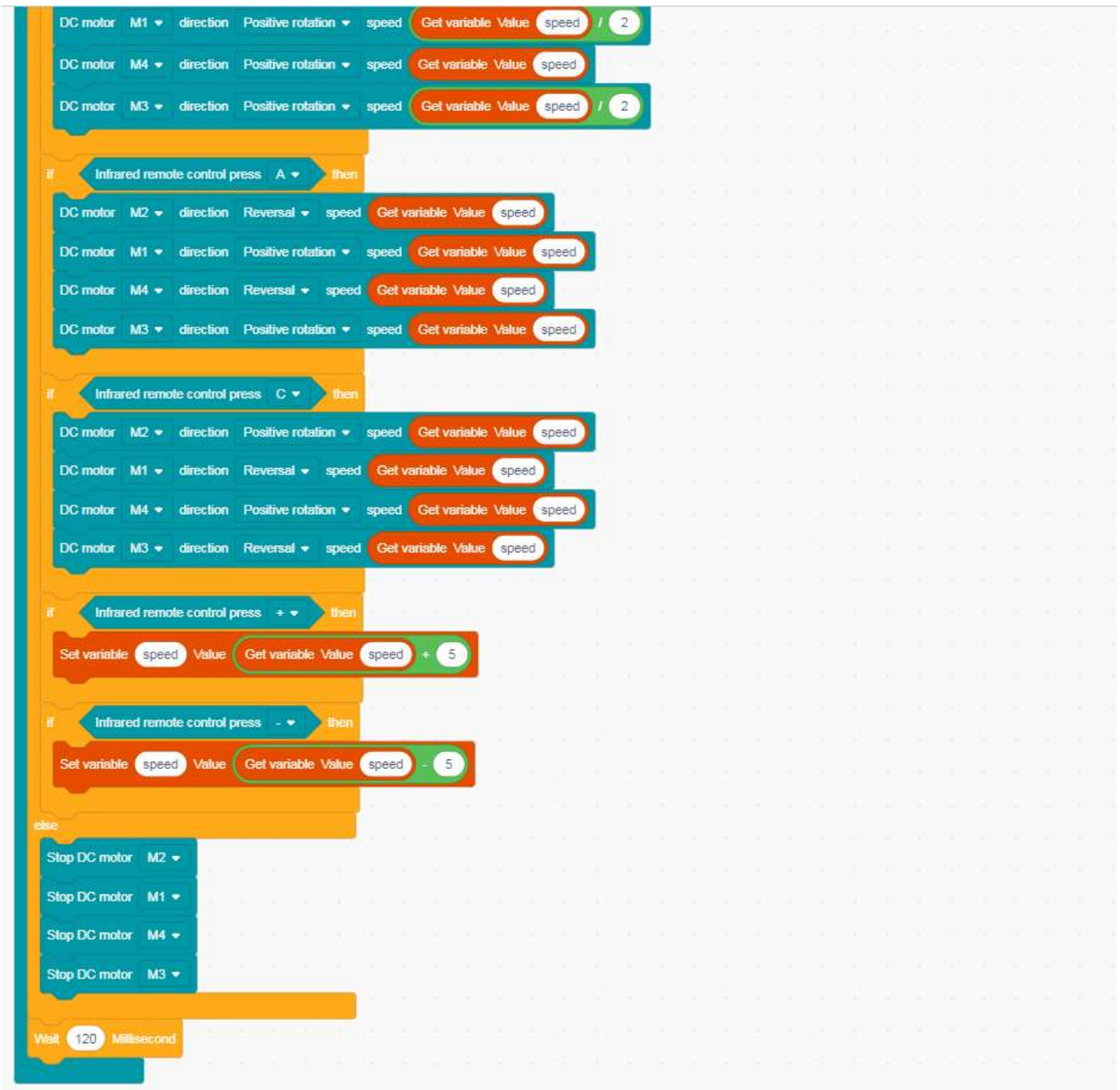
```
if    Bluetooth press button  → ▼    then
    DC motor  M2 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed
    DC motor  M1 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed  / 2
    DC motor  M4 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed
    DC motor  M3 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed  / 2

if    Bluetooth press button  pink ▼    then
    DC motor  M2 ▼  direction  Reversal ▼  speed  Get variable Value  speed
    DC motor  M1 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed
    DC motor  M4 ▼  direction  Reversal ▼  speed  Get variable Value  speed
    DC motor  M3 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed

if    Bluetooth press button  red ▼    then
    DC motor  M2 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed
    DC motor  M1 ▼  direction  Reversal ▼  speed  Get variable Value  speed
    DC motor  M4 ▼  direction  Positive rotation ▼  speed  Get variable Value  speed
    DC motor  M3 ▼  direction  Reversal ▼  speed  Get variable Value  speed

if    Bluetooth press button  green ▼    then
    Set variable  speed  Value  Get variable Value  speed  + 5

if    Bluetooth press button  blue ▼    then
    Set variable  speed  Value  Get variable Value  speed  - 5

if    Bluetooth acquisition angle  =  0xFFFF    then
    Stop DC motor  M2 ▼
    Stop DC motor  M1 ▼
```