catalogue

1. Software Installation 2
   1. Open 2
   2. Use 2
2. Lesson 1 - Flashing Lights 4
   1. Introduction 4
   2. Circuit Principle 5
   3. Mixly Program 5 5
   4. Expansion 5
3. Lesson 2 - Driving one servo motor 6
   1. Introduction 6
   2. Principle 6
   3. Mixly Program 6
   4. Expansion 8
4. Lesson 3 - Administer one Action 8
   1. Module introduction 8
   2. Mixly Program Example 9
   3. Expansion 9
5. Lesson 4 - Multiple Actions Combinations 9
   1. Introduction 9
   2. Mixly Program Example 10

5.3 Expansion 10

1. Lesson 5 - Wifi remote control robot 10

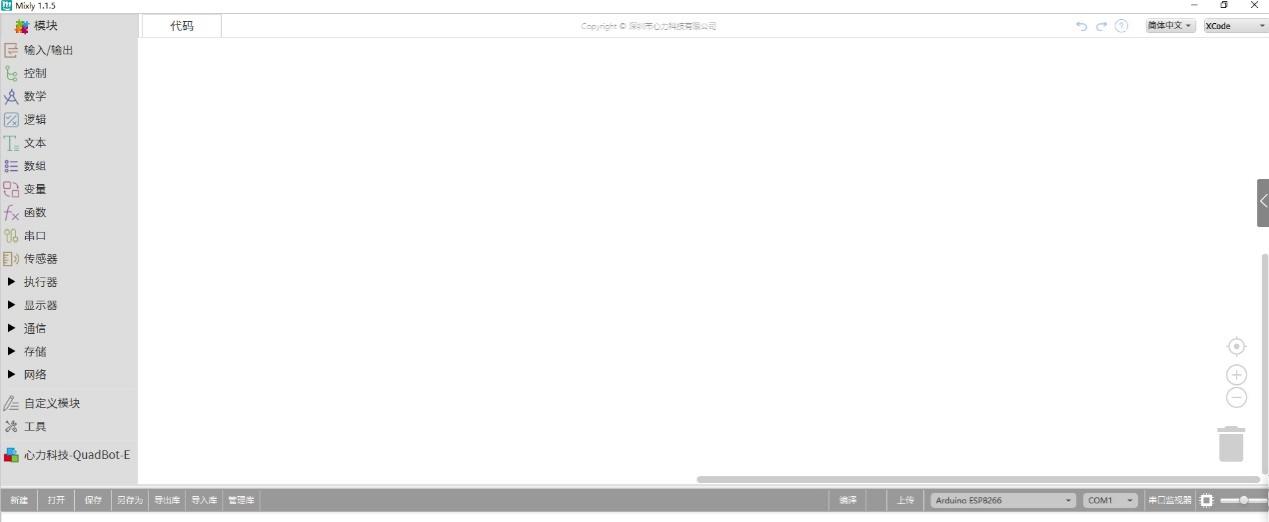
6.1 Introduction 10

6.2 Mixly Program Example 10

## Software installation

# open

Locate the catalogue to the "Software" "Mixly\_WIN1.1.5" double-click "Mixly.exe" to open the graphical programming environment we configured,the interface is shown as below.

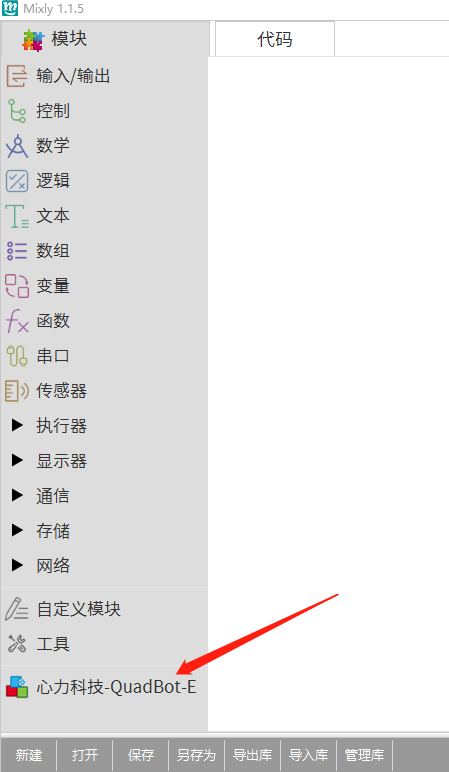


# use

### Module View

The Mixly software is on the left of modules window

The module is in the lower left corner, click the icon "心力科技 QuadBot-E", the interface is shown as below pictur



### Files creation and opening

The file creation and opening buttons are at the lower left corner



### Operation module

Drag the required modules from the modules window to the code window, and the right window display the arduino language program at once, taking the LED module as an example



### Program compilation and download

Select "Arduino ESP8266",select the COM port to your NodeMCU, and then click the "编译" button to compile, and click the "Upload" button to download the program to NodeMCU, as shown in the figure below:

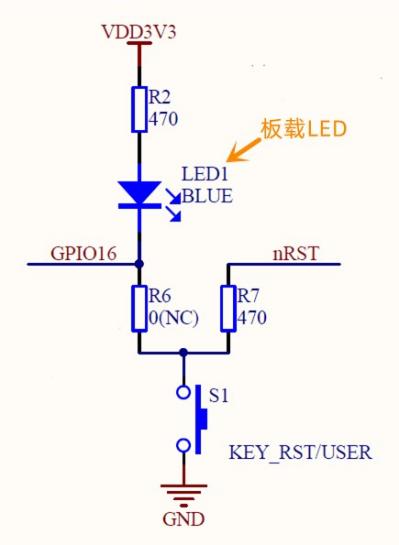


## Lesson 1 - Flashing Lights

# introduce

In this lesson, you will use NodeMcu's on-board LED to do a flashing light experiment, so that you can learn using NodeMcu IO port to output high and low level voltage.

# Circuit principle



It is easy to know from the circuit diagram that when input a low level voltage to GPIO16,the on-board LED is on, Otherwise, input a high level voltage to GPIO16,the on-board LED is off. In this way, Periodically input high and low level voltage,you can see the led light flashing periodically. The flashing cycle time is determined by controlling the delay time between high and low level voltage.The longer the time delay, the slower the flash ,on the contrary ,the faster the flash

# Mixly program

Example 1

Using the flashing light module provided by us, you can modify the switching time between the flashing light on and off by modifying the number on the module, so as to control the flashing speed of the flashing light. the interface is shown as below picture， which can be located to "Blink-1.mix" under the directory of "tutorials"  "Graphical programming"  " lesson 2 Flashing Light" .



Example 2

You can get the same result by locating to the “Blink-2.mix”,which under under the directory of "tutorials"  "Graphical Programming"  "Course 2 Flashlight" ,the interface is shown as below picture.



# expand

Modify the delay time on the module, upload the program to the NodeMCu motherboard, and check the difference.

## Lesson 2 - Driving one servo motor

# introduce

This lesson will introduce how to drive one servo motor, including how to drive the servo motor to a specified angle position and how to turn the servo motor back and forth between two angle positions.

# principle

The signal modulation chip in the servo motor receipt the signal from the control board, and then the servo motor will obtain the DC voltage. There is also a reference circuit inside the servo motor, which can generate standard voltage. These two voltages will be compared with each other and output difference value. Then the motor chip will receive difference value, it will determine the speed, direction and angle. When there is no difference between the two voltages, the servo motor will stop.

To control the rotation of the servo motor, the time pulse should be about 20ms, and the high level voltage pulse width should be about 0.5ms~2.5ms, which is consistent with the angle limit of the servo motor.。

Taking the 180 degree servo motor as an example, the corresponding control relationship is as follows:

|  |  |
| --- | --- |
| 0.5ms | 0° |
| 1.0ms | 45° |
| 1.5ms | 90° |
| 2.0ms | 135° |
| 2.5ms | 180° |

The servo motor we provide is a 180 ° reducer motor， It is controlled by sending electric pulses from the NodeMCu board. These pulses tell the servo motor where it should move. There are three cables for the servo motor. The brown one is the ground cable, connected to the GND port of the servo motor expansion board, the red one is the power cable, connected to the VCC port of the servo motor expansion board, and the orange one is the signal port except A0 and D3 connected to the expansion board.

Note: If your servo motor has been installed on the robot structure, the servo motor angle range is limited; If your servo motor is not installed on the structure, it can rotate from 0 to 180 degrees.

# Mixly program

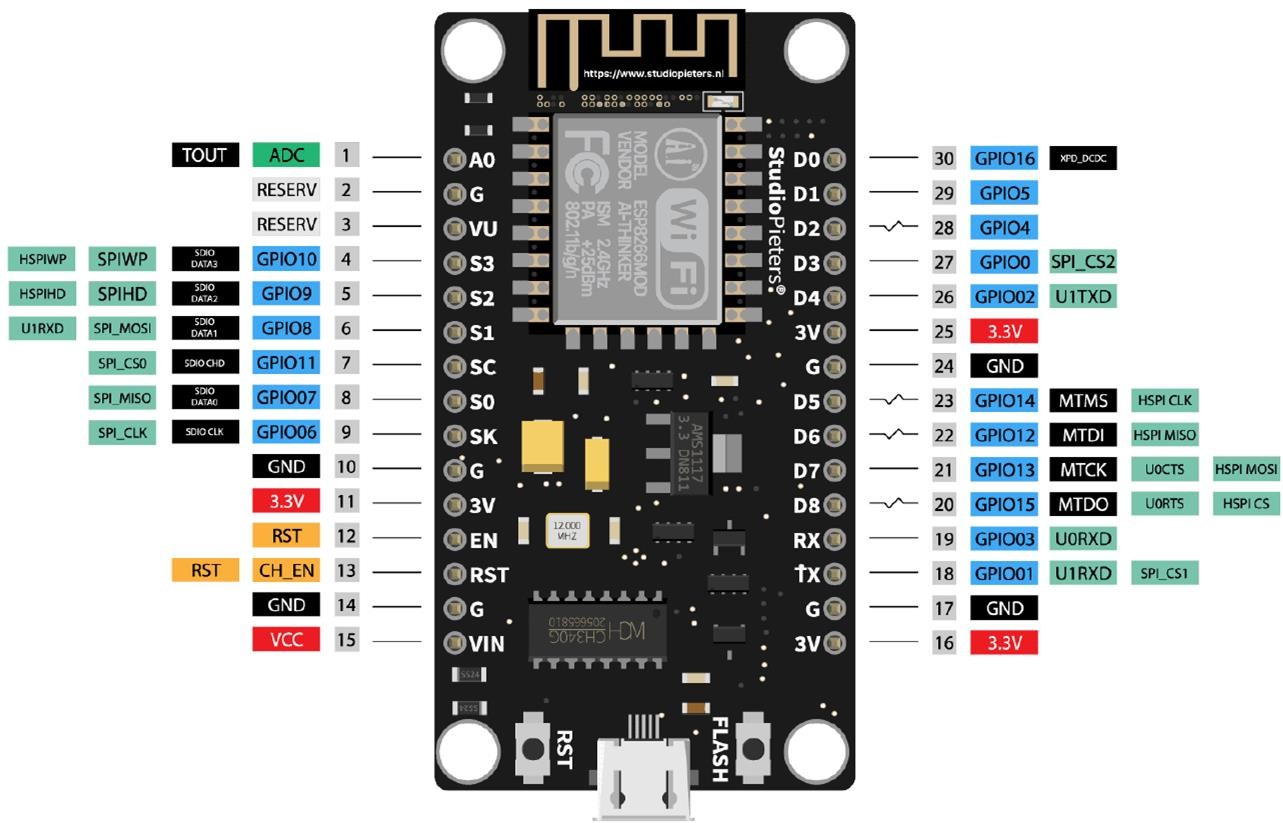
Example 1



You can select from port 1 to port 8 in the port drop-down box. The relationship between ports and expansion boards and NoceMcu main control boards is shown in the following table.

|  |  |  |
| --- | --- | --- |
| Graphical programming module port | Expansion board port | NodeMCu port |
| 1 | D0 | D0(GPIO16 |
| 2 | D1 | D1(GPIO5) |
| 3 | D2 | D2(GPIO4) |

|  |  |  |
| --- | --- | --- |
| 4 | D4 | D4(GPIO2) |
| 5 | D8 | D8(GPIO15) |
| 6 | D7 | D7(GPIO13) |
| 7 | D6 | D6(GPIO12) |
| 8 | D5 | D5(GPIO14) |

You can input the value to get to target angle position of the servo motor。 If your servo motor has been installed on the structure, in order to prevent damage, the range of motion is determined to move 10 degrees between of each servo motors. The middle angle of each servo motor is shown as following picture.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| servo motors | Middle angle |  | servo motors | Middle angle |
| Left front leg | 135 |  | Front right leg | 135 |
| Left front shoulder | 45 |  | Right front shoulder | 45 |
| Left rear shoulder | 135 |  | Rear right shoulder | 135 |
| Left hind leg | 45 |  | Rear right leg | 45 |

Taking the left front leg servo motor as an example,the target angle range in the module is 125 °~145 °.

Example 2

(Note: If your servo motor has been installed on the structure, this is risk of damaging the servo motor!!!) If your servo motor has not been installed on the structure, you can locate to the “Servo-2.mix”, which under "tutorials"  "Graphical Programming"  "Program"  "lesson 3 Servo motor" directory, as shown in the following picture



Upload the program to NodeMCu, then connect the servo motor to the port on the expansion board, connect the battery, and turn on the switch,you can see the servo motor rotate back and forth between 0 ° and 180 °.

# expand

Modify the target angle (within the safe range) in Example 1 to see the result.

Modify the port of the module in Example 2, the delay time and the target angle in the middle input box, and then test it.

## Lesson 3 - Performing one action

# Module introduction

|  |  |
| --- | --- |
|  | Used for assemble. After the program operate, the initial angular positions of 8 sevro motors |
|  | Used to calibrate the servo motor with APP after assembled |
|  | Robot standby |
|  | Robot moving forward |
|  | Robot backward |
|  | The robot moves to the left |
|  | The robot moves to the right |
|  | The robot turn around left |
|  | The robot turn around right |
|  | Robots lie on the ground |
|  | Robot beckons |
|  | The robot assumes a combat posture |
|  | Robots do push ups |
|  | Robots sleep |
|  | first dance |

|  |  |
| --- | --- |
|  | second dance |
|  | third dance |
|  | The robot returns to the center position |
|  | The robot returns to its original position |

# Mixly Program Example

Example 1

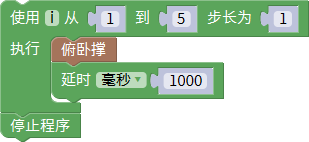
Locate to the "tutorials"  "Graphical Programming"  "Program"  "lesson 3 one Action ", and open “Forward.mix”, as shown in the following picture



After uploading the program to NodeMCU, you can see that the robot moves forward.

Example 2

Locate to the "tutorials "  "Graphical Programming"  "Program"  " lesson 4 one Action ", and open “Push\_ Up.mix”, as shown in the below picture.



After uploading the program to NodeMCU, you can see that the robot stops after 5 push ups.

# expand

Test the remaining actions separately, control the times of one action, and check the result.

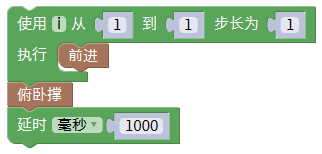
## Lesson 4 - Multiple Actions Combinations

# introduce

# This lesson will learn to combine multiple individual module actions

# Mixly Program Example

Locate to the "tutorials"  "Graphical Programming"  "Program"  "lesson 5 Combined Actions", and open “Forward\_ Push\_ Up.mix”, as shown in below picture

After uploading the program to NodeMCU, you can see that the robot stops to do a push up every two steps forward.

# expand

Test the remain actions。

## Course 5 - Wifi remote control robot

# introduce

This lesson will learn to using wifi to control robots with APP.

The program which operating on NodeMCU reads the APP and receives the command sent from the APP, The robot is controlled to opera the action

# Mixly Program Example

Locate to the " turorials"  "Graphical Programming"  "Program"  "lesson 6 Wifi Control ", and open “Wifi\_ Control.mix”, as shown in the following picture



After uploading the program to NodeMCU, locate to the "user Manual. pdf", find the "Control Experience" section, and use APP to control the robot.