

Lesson 8 Read Bus Servo Status

1. Working Principle

Bus servo has voltage, temperature, angle feedback and other functions so that we can read its status in real time.

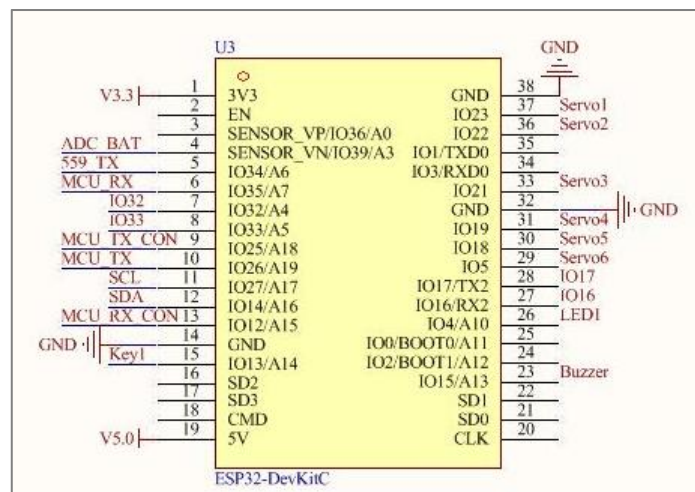
The path to the source code of the program is 5.MaxArm Hardware Basic Learning/Arduino Development/Game Programs/Read Bus Servo Status/BusServo_status/BusServo_status.ino

```

12 void setup() {
13     // put your setup code here, to run once:
14     Serial.begin(115200);      // Set the baud rate
15     Serial.println("start..."); // The serial port prints "start..."
16     BusServo.OnInit();
17     HardwareSerial.begin(115200, SERIAL_8N1, SERVO_SERIAL_RX, SERVO_SERIAL_TX);
18     delay(500);                // The delay of 500ms
19 }
20
21 bool start_en = true;
    
```

Firstly, define the serial pin and set the baud rate. Then get the servo position and voltage by BusServo.LobotSerialServoReadPosition() and BusServo.LobotSerialServoReadVin() function. Finally, the value of position and voltage is printed in serial monitor by Serial.print() function.

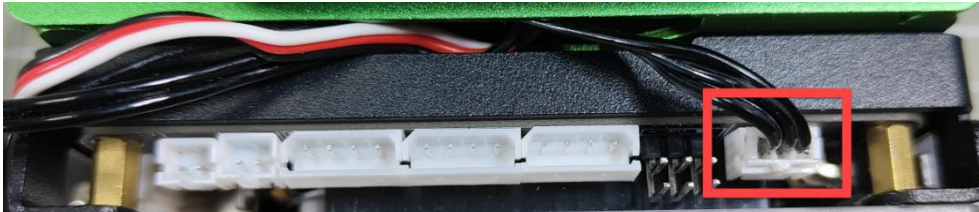
According to the following circuit diagram, 12 and 32 on the expansion board have serial ports function.



2. Preparation

2.1 Hardware

Please make sure the bus servo is individually connected to bus servo port on MaxArm controller (The bus servo had been assembled before you received MaxArm). The wiring method is as follow:



Note: The servo cable uses anti-reverse plug. Please do not insert it violently.

2.2 Software

Please connect MaxArm to the Arduino editor according to the tutorial in folder “4. MaxArm Underlying Program/Python Development/Lesson 1 Set Development Environment”.

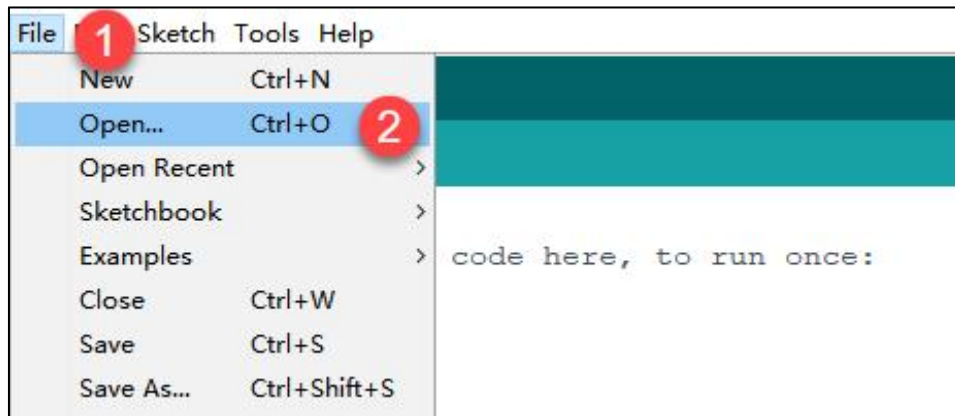
3. Program Download

- 1) Double click to open Arduino IDE.

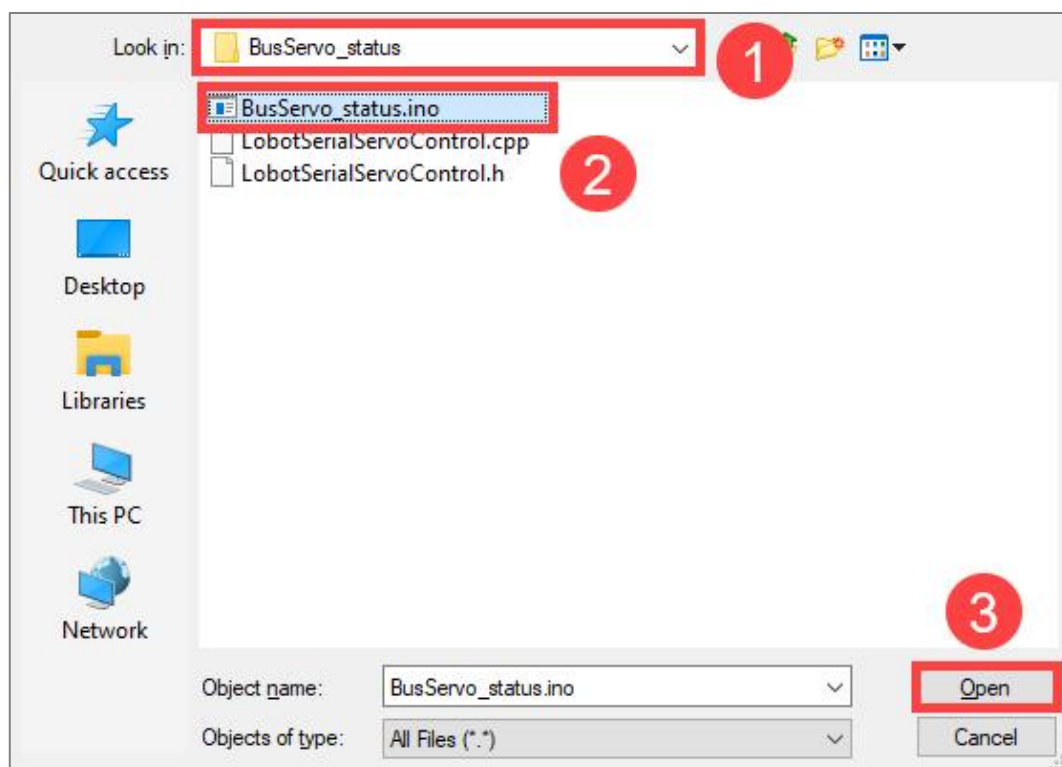
```
sketch_nov17b | Arduino 1.8.5
File Edit Sketch Tools Help

sketch_nov17b
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

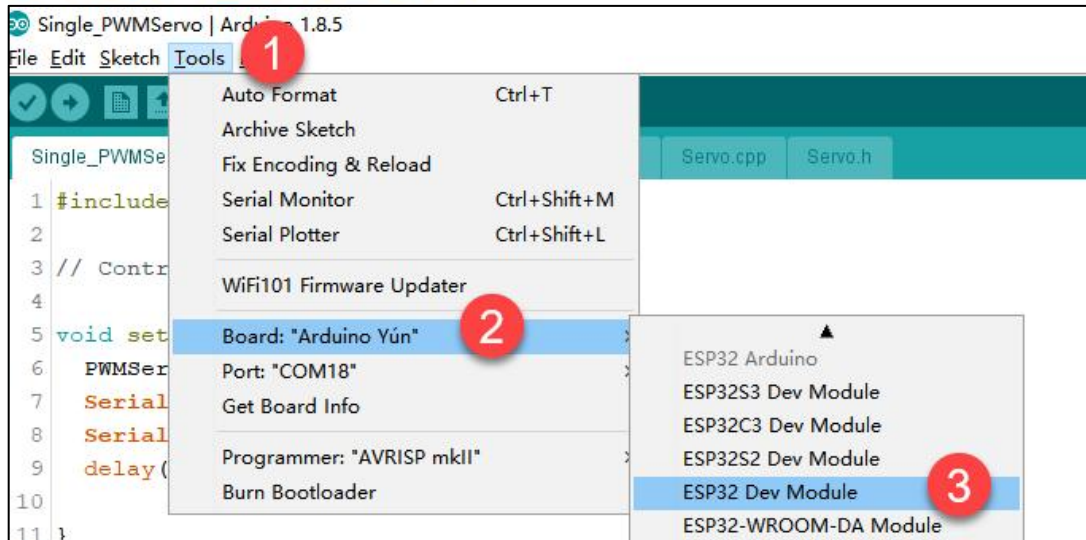
2) Click “File->Open” in turn.



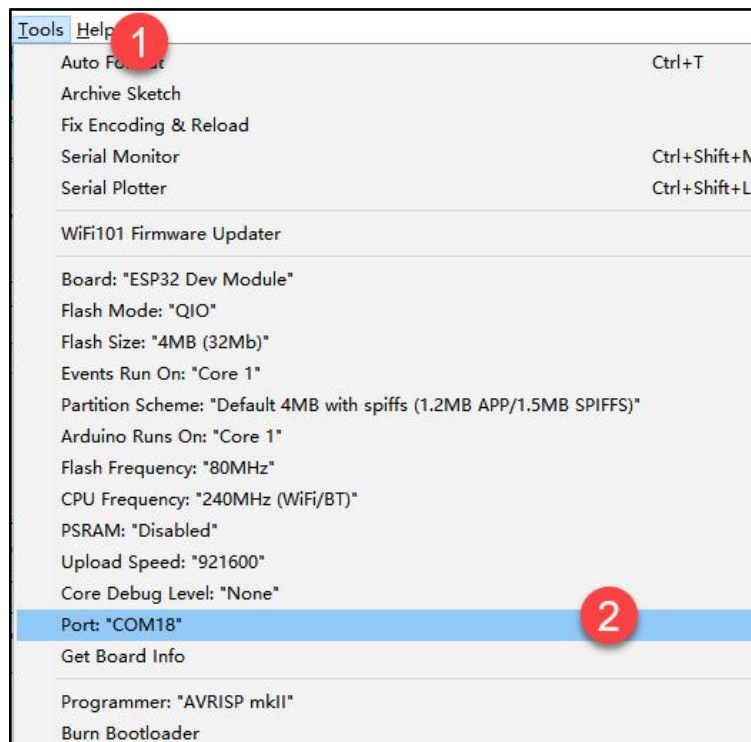
3) Select the program “BusServo_status.ino” in the folder “5.MaxArm Hardware Basic Learning/Arduino Development/Game Programs/Read Bus Servo Status/ BusServo_status”.



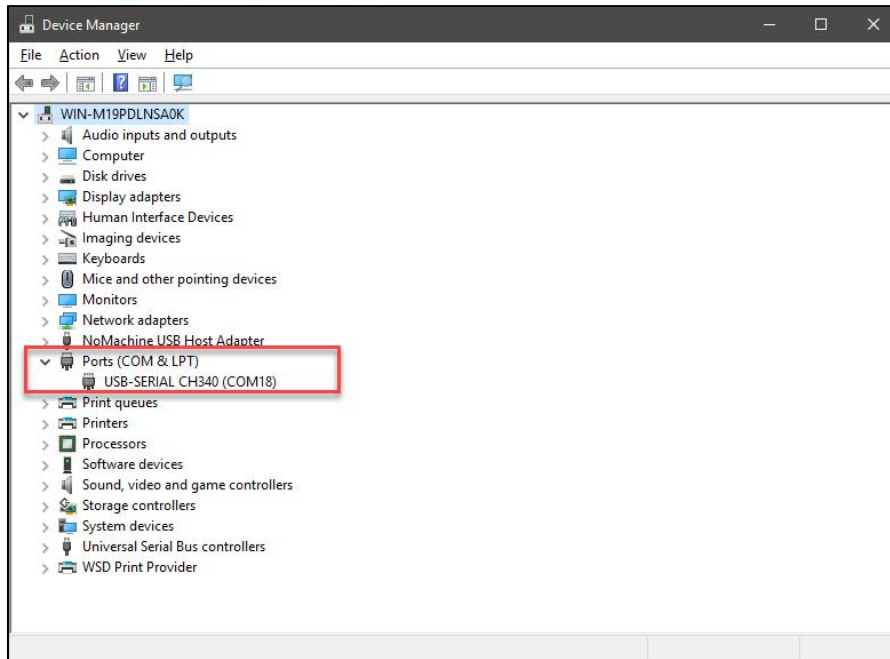
4) Select the board model. Click “Tools->Board” and select “ESP 32 Dev Module”. (If the model of development board has been configured when setting the development environment, you can skip this step.)



- 5) Select the corresponding port of ESP32 controller in “Tools->Port”. (Here take the port “COM5” as example. Please select the port based on your computer. If COM1 appears, please do not select because it is the system communication port but not the actual port of the development port.)




- 6) If you are not sure about the port number, please open the “This PC” and click “Properties->Device Manager” in turns to check the corresponding port number (the device is with CH340).




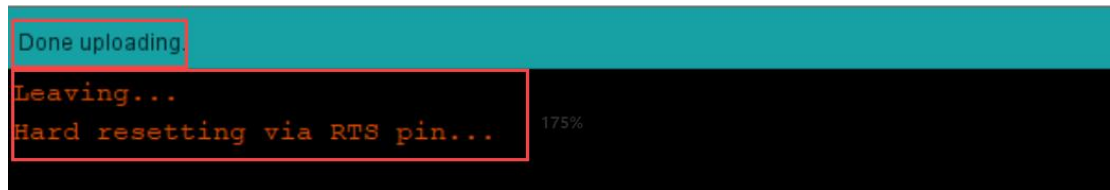
- 7) After selecting, confirm the board “ESP32 Dev Module” in the lower right corner and the port number “COM5” (it is an example here, please refer to the actual situation).


ESP32 Dev Module, Disabled, Default 4MB with spiiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 821800, Core 1, Core 1, None on COM18

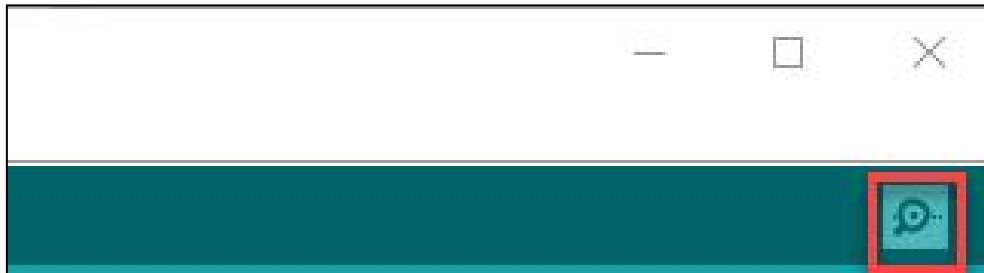
- 8) Then click on  icon to verify the program. If no error, the status area will display “Compiling->Compile complete” in turn. After compiling, the information such as the current used bytes, and occupied program storage space will be displayed.

```
Done compiling.
Sketch uses 247733 bytes (18%) of program storage space. Maximum is 1310720 bytes.
Global variables use 16584 bytes (5%) of dynamic memory, leaving 311096 bytes for local variables. Maximum is 327680 bytes.
```

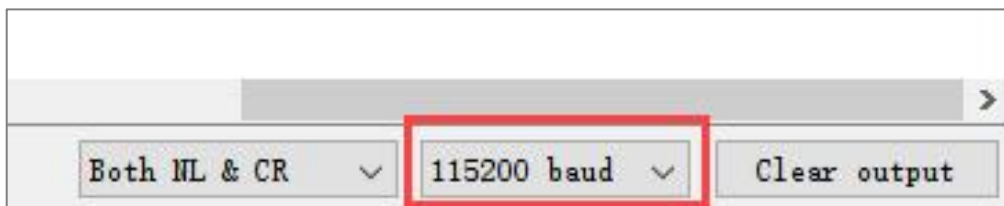
- 9) After compiling, click on  icon to upload the program to the development board. The status area will display “Compiling->Uploading->Complete” in turn. After uploading, the status area will stop printing the uploading information.



10) Then click on the serial monitor icon  in the upper right corner.



11) Select "115200 baud rate" in the pop-up window.



4. Project Outcome

When running program, the terminal will print the current position and voltage information.



5. Function Extension

The program is set to read the information of ID1 servo. If want to read others, you can change to the corresponding code to implement. This lesson will


change to read the information of ID2 servo. The specific operation steps are as follow:

1) Find the following code:

```
25 Serial.println(BusServo.LobotSerialServoReadPosition(1)); // Obtain and print the position of ID1 servo
26 delay(200); // Delay
27 Serial.print("Vin: ");
28 Serial.print(BusServo.LobotSerialServoReadVin(1)/1000.0); // Obtain and print the voltage of ID1 servo
```

2) Change the ID number from 1 to 2, as shown in the figure below:

```
25 Serial.println(BusServo.LobotSerialServoReadPosition(2)); // Obtain and print the position of ID1 servo
26 delay(200); // Delay
27 Serial.print("Vin: ");
28 Serial.print(BusServo.LobotSerialServoReadVin(2)/1000.0); // Obtain and print the voltage of ID1 servo
```

3) After modifying, click on  icon. In the meantime, the terminal will show the following prompt.

```
Done compiling.
Sketch uses 247733 bytes (18%) of program storage space. Maximum is 1310720 bytes.
Global variables use 16584 bytes (5%) of dynamic memory, leaving 311096 bytes for local variables. Maximum is 327680 bytes.
```

4) Click on  icon.

5) Refer to the steps 6-8 in “3. Program Download” to download the program and check the position and voltage information of ID2 servo.