Lesson 7 Control Bus Servo Speed

1. Working Principle

According to the communication protocol, servo is controlled to rotate by sending the commands including servo ID, rotation angle and time.

The path to the source code of the program is 5. MaxArm Hardware Basic Learning/Arduino Development/Game Programs/Control BUS Servo Speed/ BusServo_turn/BusServo_speed.ino

```
26  if(start_en){
27    int t[2] = {500, 2000};
28    for(int i=0; i < 2; i++){ // Run one round at different times, the longer the time, the slower the speed.
29      BusServo.LobotSerialServoMove(1,600,t[i]); // The ID1 servo is set to run to 600 pulse width.
30      delay(t[i]);
31
32      BusServo.LobotSerialServoMove(1,500,t[i]); // The ID1 servo is set to run to 500 pulse width.
33      delay(t[i]);
34
35      BusServo.LobotSerialServoMove(1,400,t[i]); // The ID1 servo is set to run to 400 pulse width.
36      delay(t[i]);
37
38      BusServo.LobotSerialServoMove(1,500,t[i]); // The ID1 servo is set to run to 500 pulse width.
39      delay(t[i]);
40    }
```

Control bus servo by calling BusServo.LobotSerialServoMove() in LobotSerialServoControl.h library.

Take the code "BusServo.LobotSerialServoMove(1,500,t[i])" as example.

The first parameter "1" is the servo ID. Here is ID1 servo.

The second parameter "500" represents the rotation position. The parameter is the data converted by angle.

The third parameter "1000" represents the rotation time (unit is ms). Here the time is 1000ms.

The rotation range of bus servo is between 0 and 1000 pulse width which corresponds to 0°-240°, i.e, 1° is roughly equal to 4.2 pulse width. The conversion formula for angle and pulse width is: pulse width=4.2×angle (just for reference).

2. Preparation

2.1 Hardware

Please make sure the bus servo is individually connected to bus servo port on MaxArm controller. The wiring method is as follow:
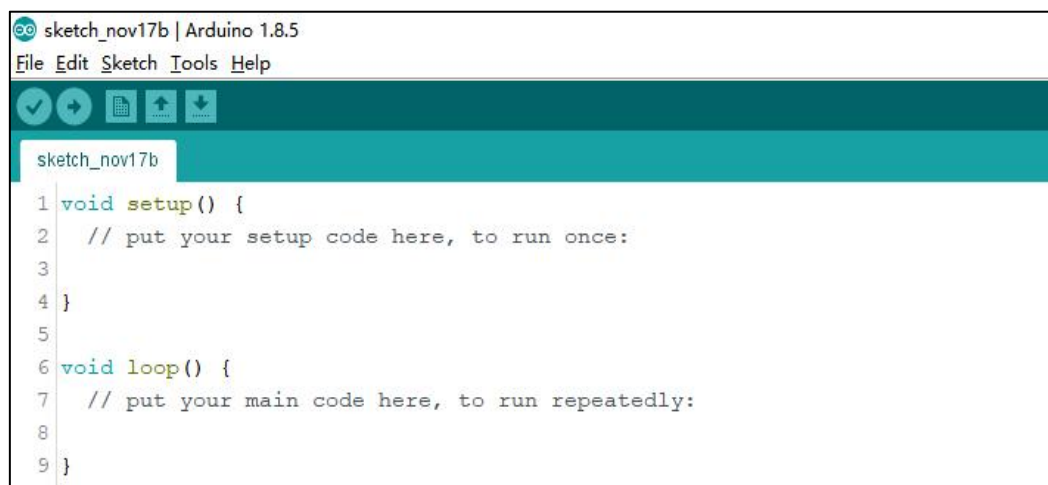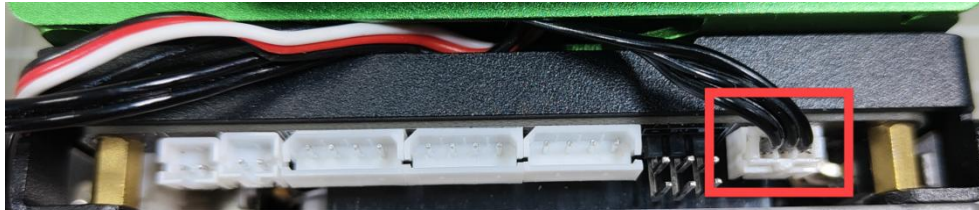

Note: The servo cable uses anti-reverse plug. Please do not insert it violently.

2.2 Software

Please connect MaxArm to the Arduino editor according to the tutorial in folder "4. MaxArm Underlying Program Learning/Python Development/Lesson 1 Set Development Environment".
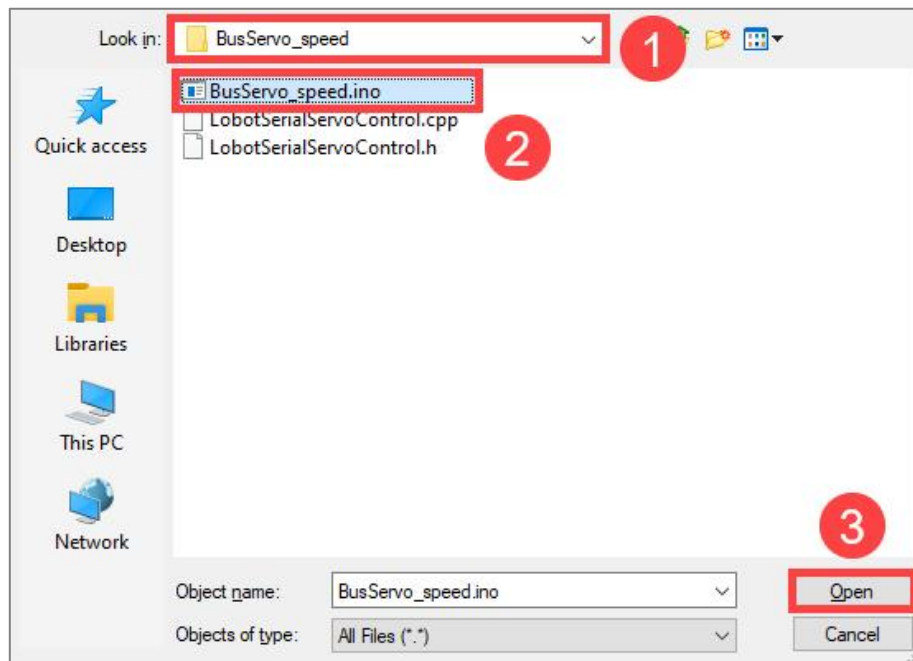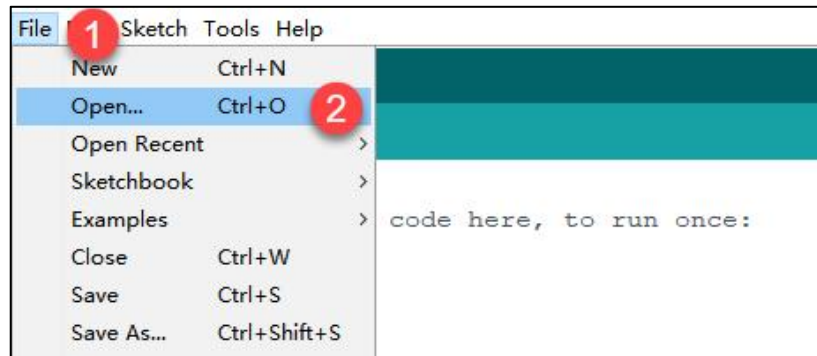
3. Program Download

1) Double click on icon to open Arduino IDE.

2) Click "File->Open" in turn, and then select the program

"BusServo_speed.ino" in the folder "5.MaxArm Hardware Basic

Learning/Arduino Development/Game Programs/Control Bus Servo

Speed/BusServo_speed.ino".

3) 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.)
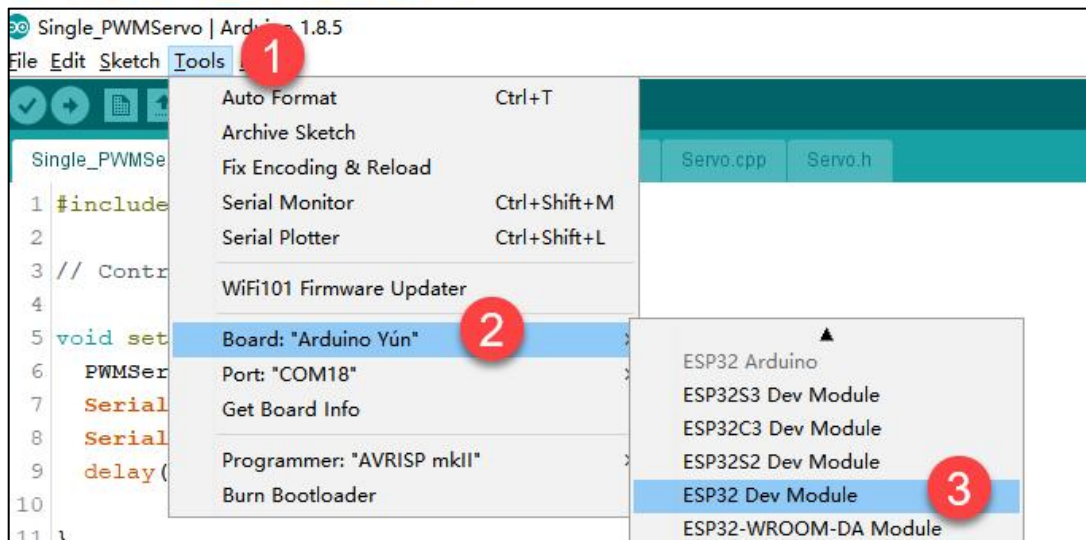
4) Select the corresponding port of ESP32 controller in "Tools->Port". (Here take the port "COM5" as example. Please select the port based on your computer. If COM1 appears, please do not select because it is the system communication port but not the actual port of the development port.)
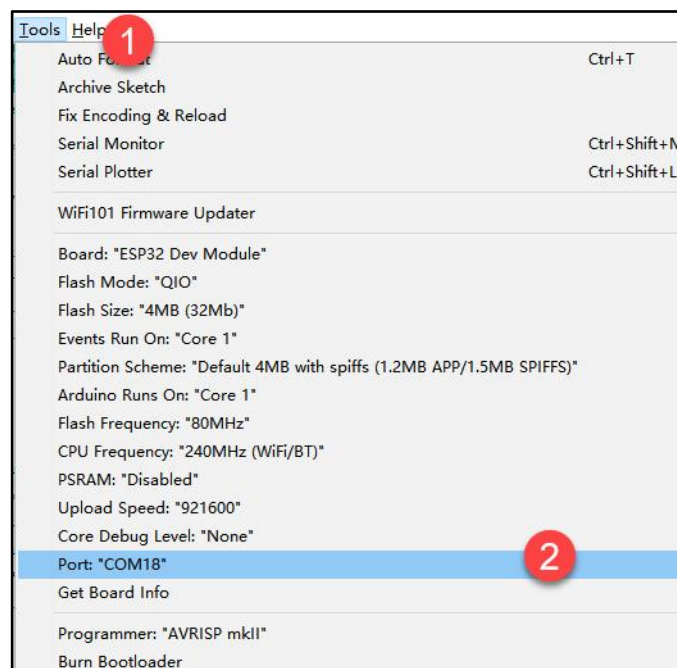
5) If you are not sure about the port number, please open the "This PC" and click "Properties->Device Manger" in turns to check the corresponding port number (the device is with CH340).
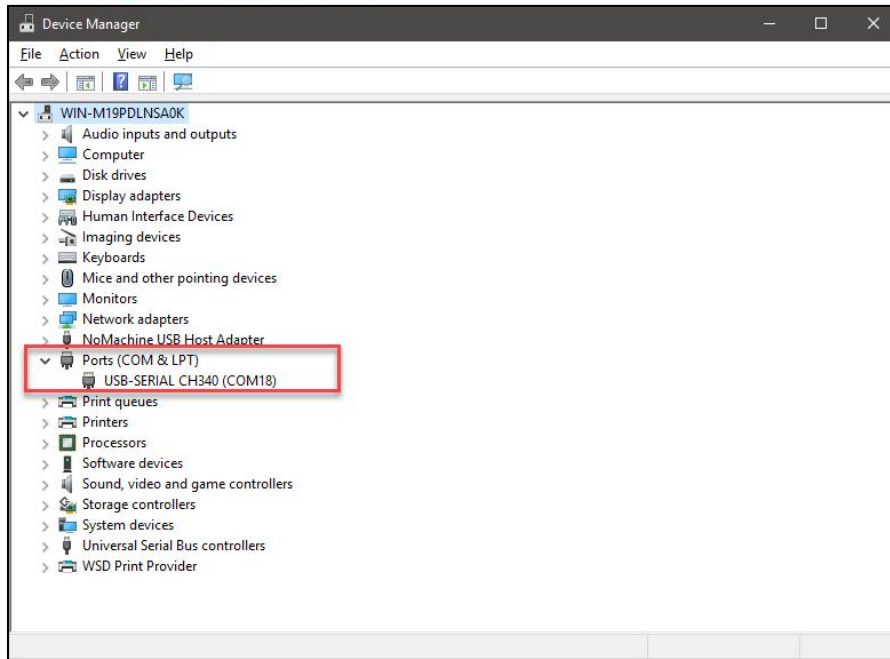
6) After selecting, confirm the board "ESP32 Dev Module" in the lower right corner and the port number "COM5" (it is an example here, please refer to the actual situation).

ESP32 Dev Module, Disabled, Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, Core 1, Core 1, None on COM18

7) Then click on [icon] 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.

8) After compiling, click on [icon] 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.

Done uploading.

Leaving...
Hard resetting via RTS pin...   175%
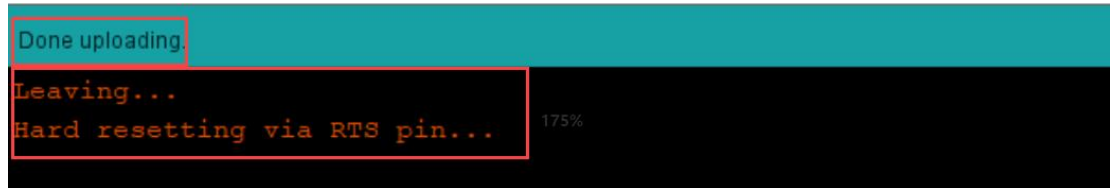
4. Project Outcome

When running program, ID1 servo will rotate 25° to the right, then 45° to the left, finally 25° to the right to return to the initial position. This process will repeat twice and the first rotation will be faster than the first rotation. After the servo stops rotating, exit the program automatically.

The rotation range of bus servo is between 0 and 1000 pulse width which corresponds to 0°-240°, i.e, 1° is roughly equal to 4.2 pulse width. The conversion formula of angle and pulse width is pulse with=4.2×angle (just for reference).

5. Function Extension

The first rotation speed set in program is faster than the second one. If want to change its rotation speed, you can modify the corresponding code to implement. Here the "t" parameter value is changed from (500,2000) to (2000,500). The specific operation steps are as follow:

1) Find the following program:

```
26   if(start_en){
27     int t[2] = {500, 2000};
28     for(int i=0; i < 2; i++){ // Run one round at different times, the longer the time, the slower the speed.
29        BusServo.LobotSerialServoMove(1,600,t[i]); // The ID1 servo is set to run to 600 pulse width.
30        delay(t[i]);
31
32        BusServo.LobotSerialServoMove(1,500,t[i]); // The ID1 servo is set to run to 500 pulse width.
33        delay(t[i]);
34
35        BusServo.LobotSerialServoMove(1,400,t[i]); // The ID1 servo is set to run to 400 pulse width.
36        delay(t[i]);
37
38        BusServo.LobotSerialServoMove(1,500,t[i]); // The ID1 servo is set to run to 500 pulse width.
39        delay(t[i]);
40     }
```

2) Change the first parameter of "t" to 2000 and the second parameter to 500, as shown in the image below:

```
26   if(start_en){
27     int t[2] = {2000, 500};
28     for(int i=0; i < 2; i++){ // Run one round at different times, the longer the time, the slower the speed.
29        BusServo.LobotSerialServoMove(1,600,t[i]); // The ID1 servo is set to run to 600 pulse width.
30        delay(t[i]);
31
32        BusServo.LobotSerialServoMove(1,500,t[i]); // The ID1 servo is set to run to 500 pulse width.
33        delay(t[i]);
34
35        BusServo.LobotSerialServoMove(1,400,t[i]); // The ID1 servo is set to run to 400 pulse width.
36        delay(t[i]);
37
38        BusServo.LobotSerialServoMove(1,500,t[i]); // The ID1 servo is set to run to 500 pulse width.
39        delay(t[i]);
40     }
```

3) After modifying, click on ✓ icon to verify the program.

```
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 outcome.