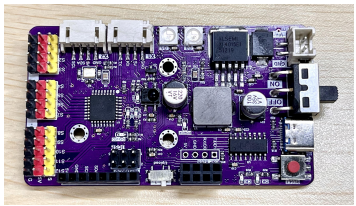


Lesson 4 Control the Servo to Rotate

4.1 Overview

In the field of robot manufacturing, servos are core components. This lesson will help you master the principle of servo control and how to control a servo to rotate to a specified angle.

4.2 Required Components

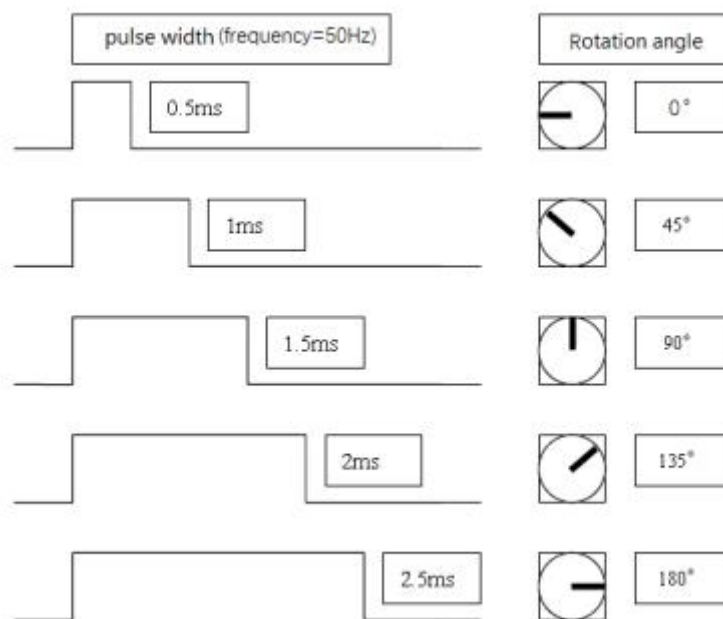
Components	Quantity	Picture
Adeept Pixie Board	1	
Type-C USB Cable	1	
Servo	1	

4.3 Principle Introduction

Working principle

The servo operates based on a pulse positioning mechanism. Essentially, when the servo receives a pulse, it rotates by a corresponding angle to achieve displacement. The servo has the ability to send pulses, and for each rotation angle, a specific number of pulses are emitted. This forms a closed - loop system for pulse reception and transmission, enabling the system to precisely control the motor's rotation and achieve accurate positioning.

In practical scenarios, the Adeept Pixie Board sends PWM (Pulse - Width Modulation) signals to the servo. These signals are processed by the integrated circuit (IC) on the circuit board, which calculates the rotation direction of the driving motor. The rotation is then transmitted to the swing arm through a reduction gear. Simultaneously, the position detector returns a position signal to determine whether the set position has been reached. At a frequency of 50Hz, different pulse widths correspond to different rotation angles: a 0.5ms pulse corresponds to 0°, a 1ms pulse to 45°, a 1.5ms pulse to 90°, a 2ms pulse to 135°, and a 2.5ms pulse to 180°.

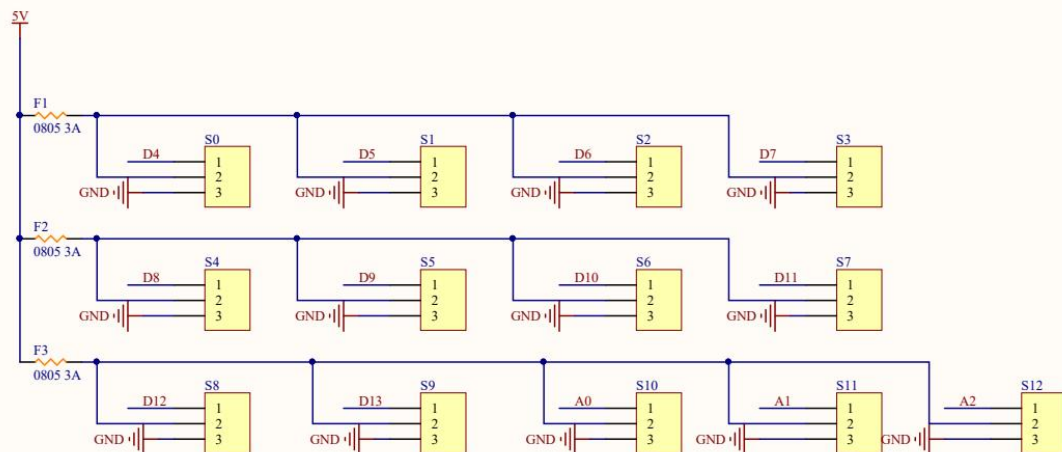
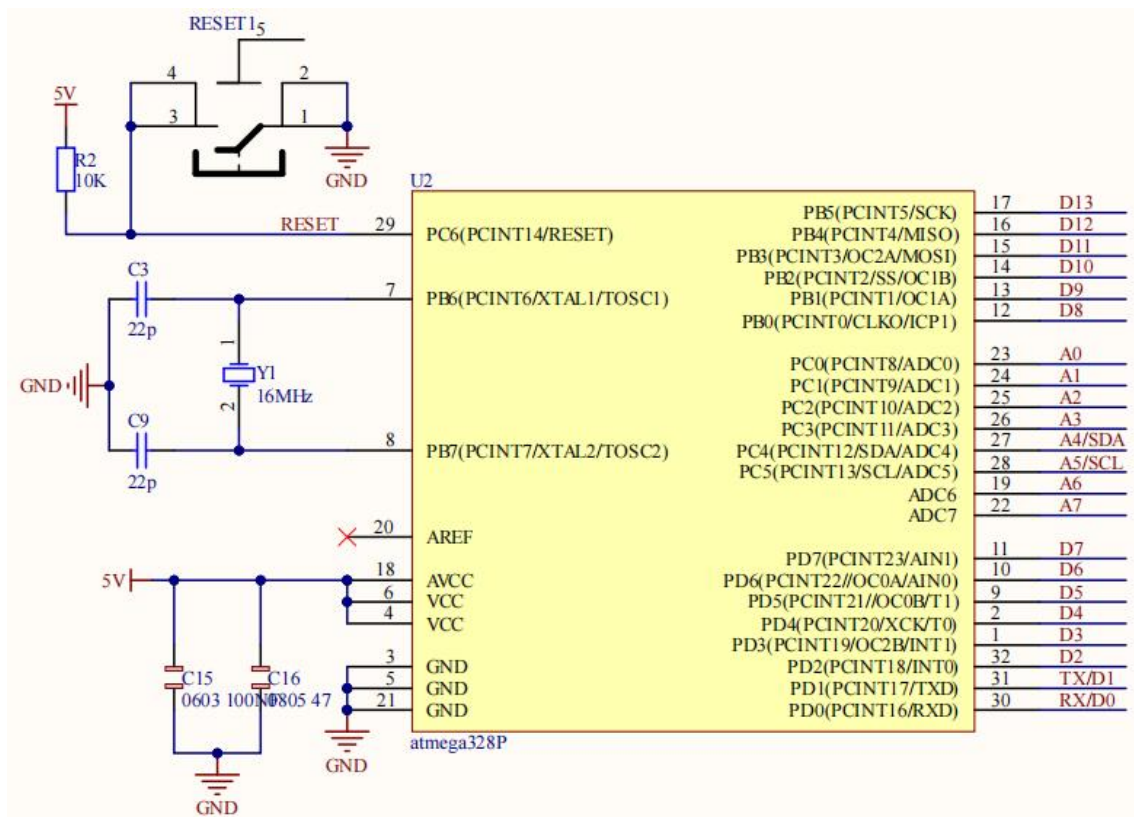


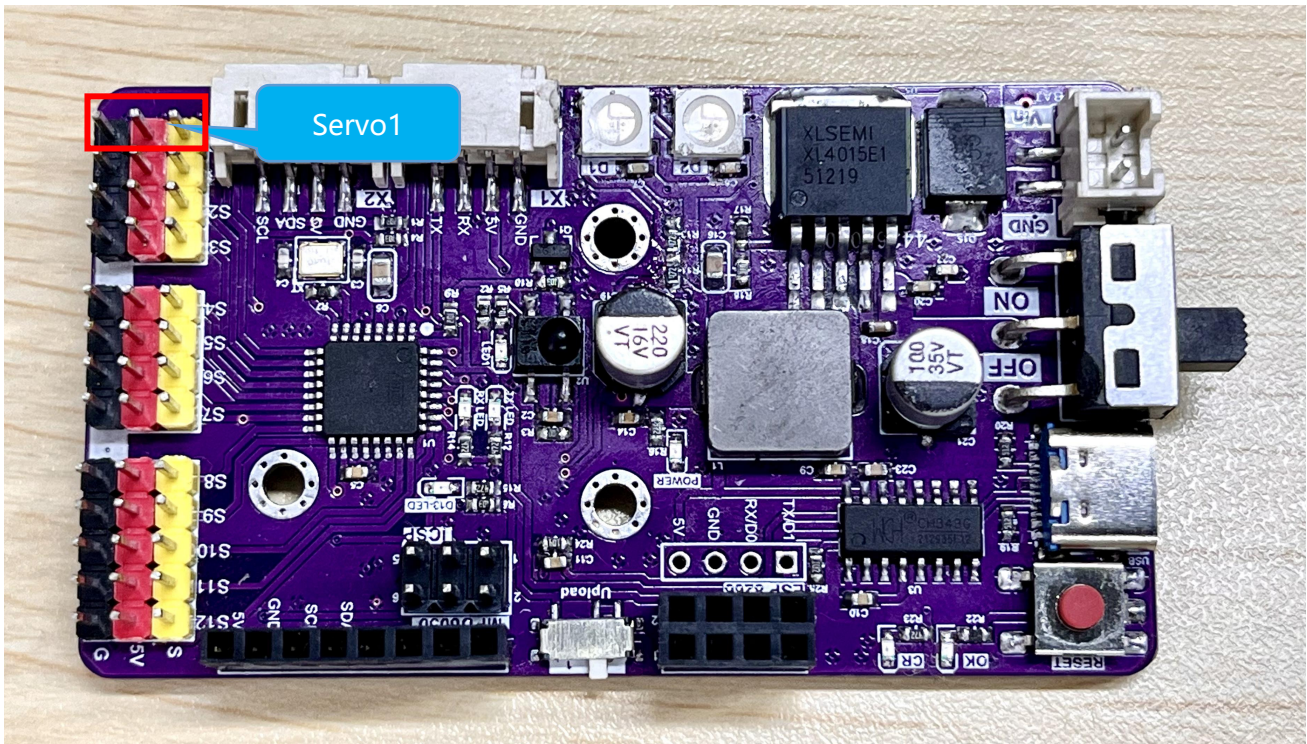
Principle of the Write() function

In programming, the write() function is utilized to control the rotation of the servo. For standard servos, this function rotates the servo shaft to the corresponding angular position. For continuous - rotation servos, the write() function can set their rotation speed. A value of 0 represents full - speed rotation in one direction, 180 represents full - speed rotation in the opposite direction, and 90 represents a stop. In this tutorial, we focus on standard servo motors.

4.4 Wiring Diagram

Connect the servo to ports D4, Pay attention to the pin correspondence.





PINS of Adeept Pixie Board	Servo
S1	Servo1

4.5 Demonstration

1. Connect your computer and Adeept Pixie Board with a USB cable.
2. Open "04_Servo" folder in "/Code" , double-click "04_Servo.ino" .

```

1  /*****
2  File name: Servo.ino
3  Website: www.adeept.com
4  E-mail: support@adeept.com
5  *****/
6  #include <Servo.h>
7  int servoPin1 = 4; //Define servo interface digital
8
9  Servo servo1;
10
11 void setup() {
12   // put your setup code here, to run once:
13   pinMode(servoPin1,OUTPUT); //Set the servo interface as the output interface
14
15   servo1.attach(servoPin1);
16 }
17
18 void loop()
19 {
20   servo1.write(0); // Set servo 1 to turn to 0 degrees.
21   delay(1000); // delay 1s.
22   servo1.write(90); // 90 degrees,servo have angular deviation
23   delay(1000);
24   servo1.write(180); // 180 degrees.
25   delay(1000);
26 }
27

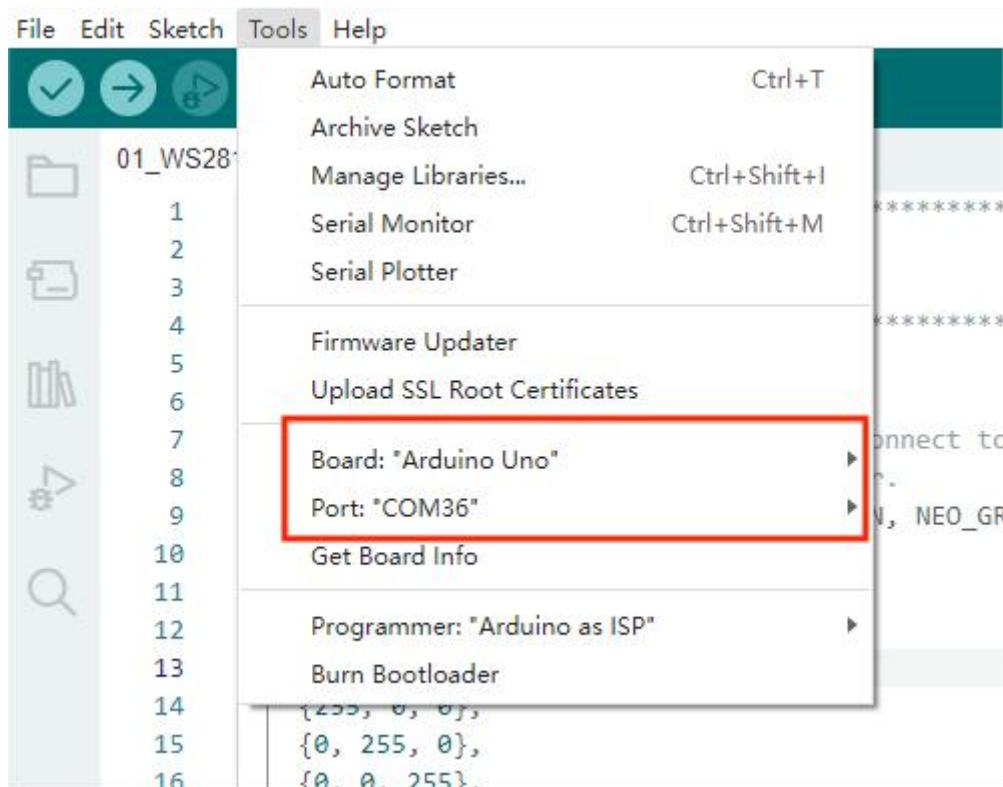
```

3. Select development board and serial port.

Board: **Tools--->Board--->Arduino AVR Boards--->Arduino Uno**

Port: **Tools --->Port--->COMx**

Note: The port number will be different in different computers.





4. After opening, click  to upload the code program to the Arduino. If there is no error warning in the console below, it means that the Upload is successful.

```

Output
Sketch uses 2952 bytes (9%) of program storage space. Maximum is 32256 bytes.
Global variables use 82 bytes (4%) of dynamic memory, leaving 1966 bytes for local variables. Maximum is 2048 bytes.

```

5. After the program runs successfully, the onboard servo will rotate in sequence to 0 degrees, 90 degrees, and 180 degrees, maintaining each angle for 1 second, and so on.

4.6 Code

```

01  /*****
02  File name: Servo.ino
03  Website: www.adeept.com
04  E-mail: support@adeept.com
05  *****/
06  #include <Servo.h>
07  int servoPin1 = 4;    //Define servo interface digital
08
09  Servo servo1;
10
11  void setup() {
12      // put your setup code here, to run once:
13      pinMode(servoPin1,OUTPUT);//Set the servo interface as the output interface
14
15      servo1.attach(servoPin1);
16  }
17
18  void loop()
19  {
20      servo1.write(0); // Set servo 1 to turn to 0 degrees.
21      delay(1000);     // delay 1s.
22      servo1.write(90); // 90 degrees,servo have angular deviation
23      delay(1000);
24      servo1.write(180); // 180 degrees.
25      delay(1000);
26  }

```

Code explanation

Initialization phase:

The `setup()` function is called once at the start of the program. It configures the specified pin (D4 in this case) as an output pin for the servo control signal and attaches the servo object to this pin.

Loop control process:

Stage 1: The servo system first rotates to 0 degrees → Delay for 1 second

Stage 2: Then the servo rotates to 90 degrees→ Delay for 1 second

Stage 3: Finally, the servo rotates to 180 degrees→ Delay for 1 second

Forming a cyclic effect, the servo system cycles at these angles.