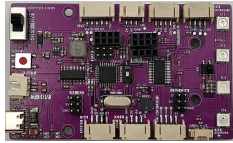




Lesson 3 How to Control the Servo through PCA9685

3.1 Components used in this course

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

3.2 Introduction of the PCA9685

Due to the limited pins of Arduino, only 2 pins are reserved as servo interface. In order to increase the number of servo interfaces, use the PCA8685 chip to add 2 servo interfaces.

PCA9685

The PCA9685 is a 16-channel, 12-bit PWM controller. It is a chip specially used to control devices such as servos and LEDs. PCA9685 communicates with the main control device (such as Arduino, Raspberry Pi, etc.) through the I2C bus, allowing the main control device to control multiple servos, motors or LEDs at the same time.

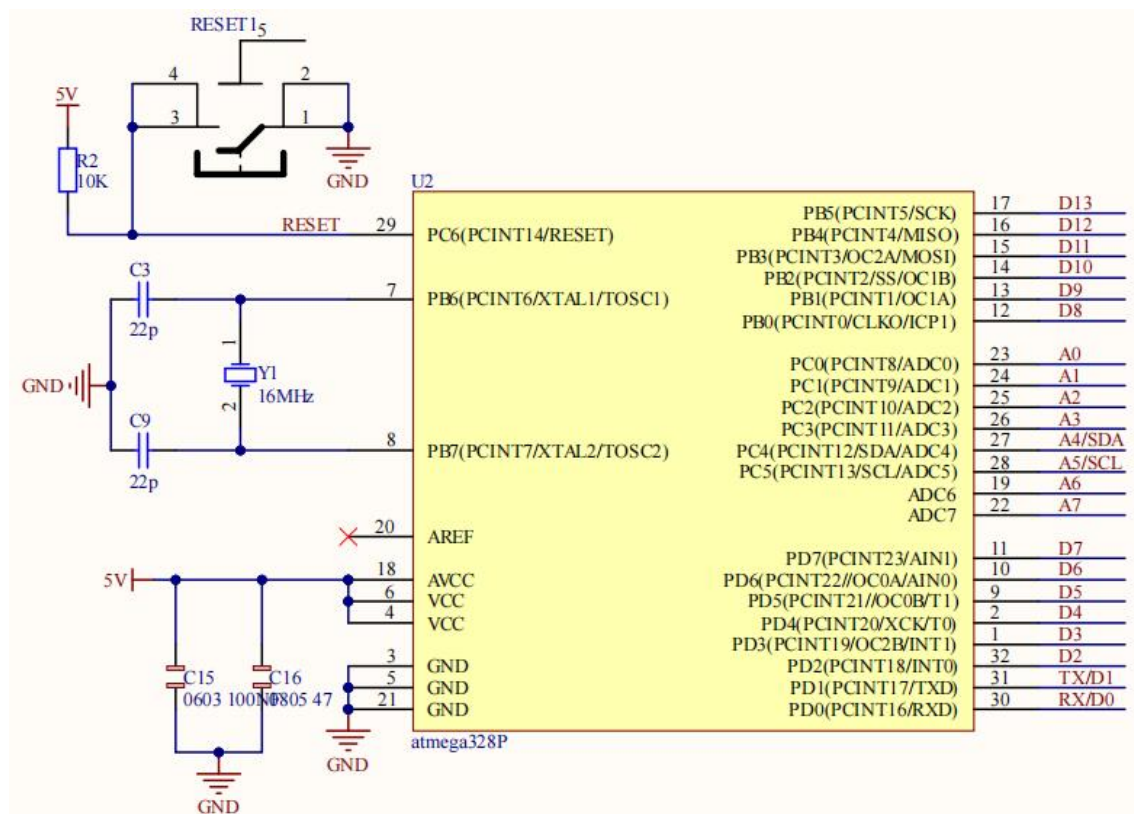
PCA9685 integrates 16 independent PWM output channels, and each channel can set the pulse width to control the brightness or angle of external devices (such as servos or LEDs). Its PWM output frequency can be set according to application requirements. By changing the pulse width, the brightness or position of external devices can be adjusted.

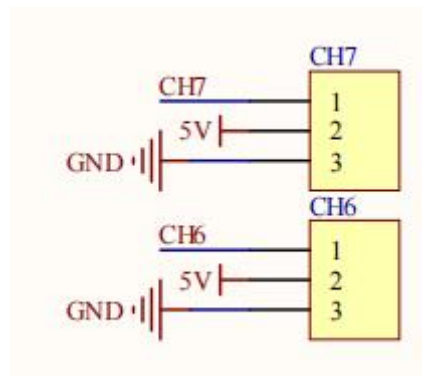
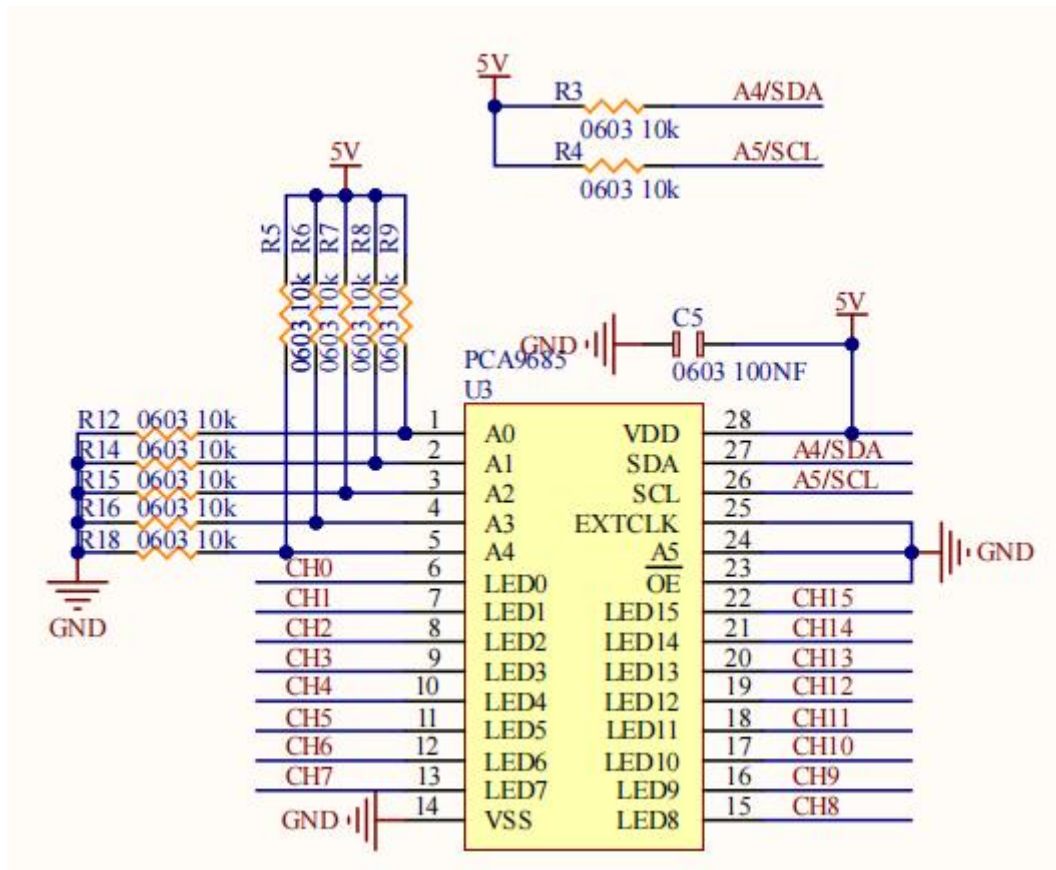
16-channel PWM output: PCA9685 can control up to 16 devices at the same time, such as servos or LEDs.

12-Bit Precision: Each channel's PWM output supports 12-bit resolution, meaning there are 4096 different brightness or angle levels to choose from.

Multi-frequency selection: PWM output frequency can be set to meet the requirements of different equipment.

3.3 Wiring diagram

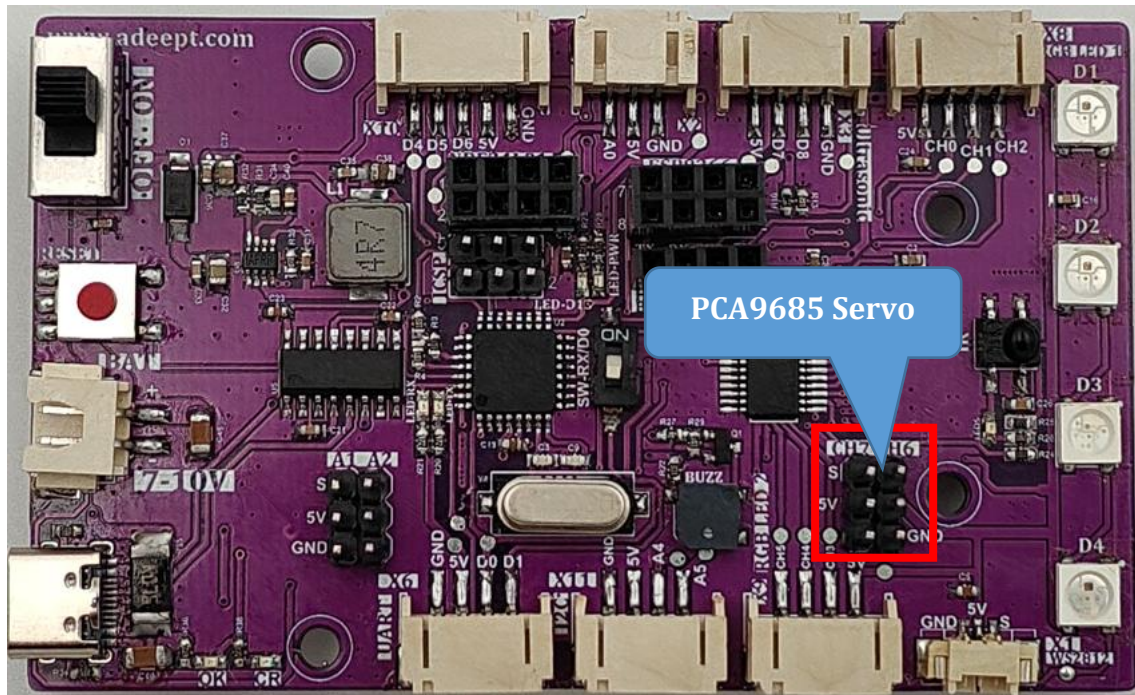




3.4 Wiring diagram

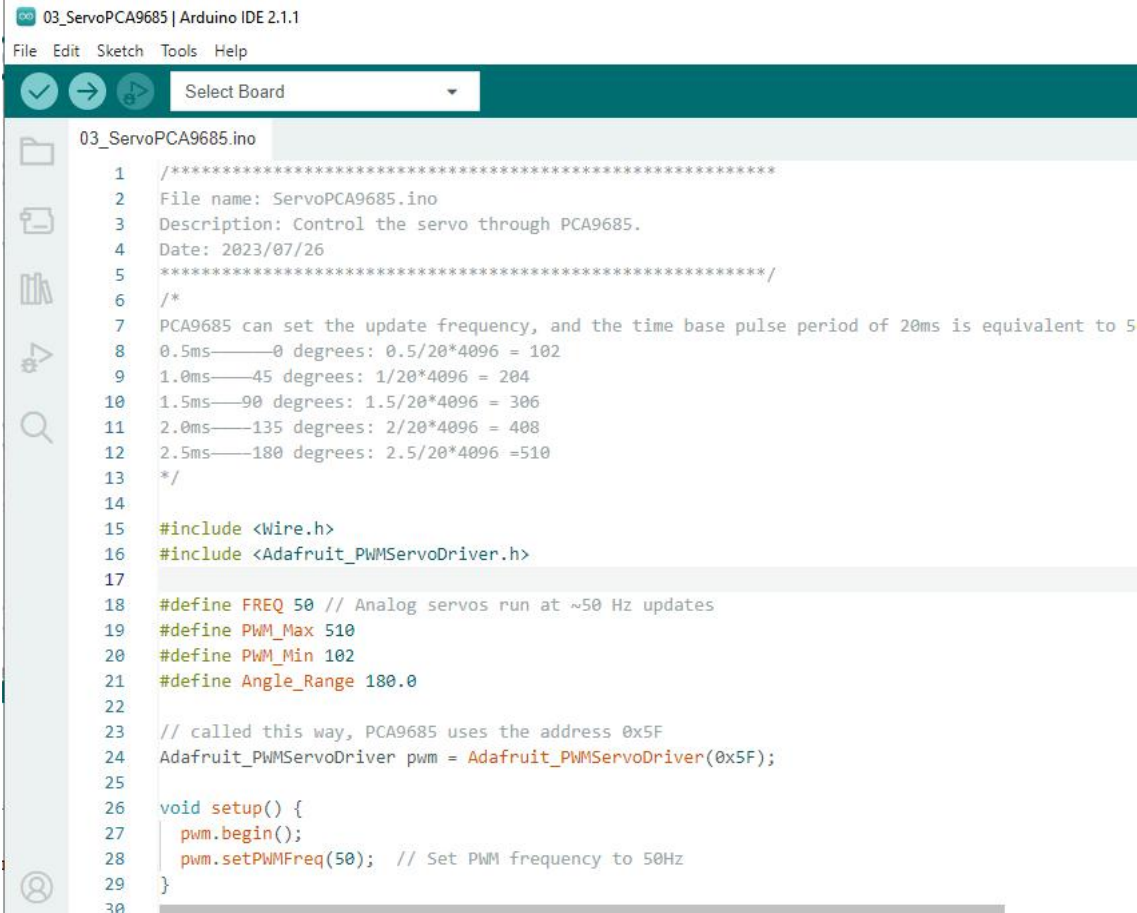
Connect the servo to the CH6/CH7 port. Note: The yellow servo wire is connected to the yellow pin, the red servo wire is connected to the red pin, and the black servo wire is connected to the black servo pin.

Figure as below:



3.5 How to control Servo through PCA9685

1. Connect your computer and Adeept Robot Control Board(Arduino Board) with a USB cable.
2. Open “03_ServoPCA9685” folder in “[Adeept_UnoCar-B/Code](#)”, double-click “03_ServoPCA9685.ino”.



```

03_ServoPCA9685 | Arduino IDE 2.1.1
File Edit Sketch Tools Help
Select Board

03_ServoPCA9685.ino
1  /*****
2  File name: ServoPCA9685.ino
3  Description: Control the servo through PCA9685.
4  Date: 2023/07/26
5  *****/
6  /*
7  PCA9685 can set the update frequency, and the time base pulse period of 20ms is equivalent to 5
8  0.5ms——0 degrees: 0.5/20*4096 = 102
9  1.0ms——45 degrees: 1/20*4096 = 204
10 1.5ms——90 degrees: 1.5/20*4096 = 306
11 2.0ms——135 degrees: 2/20*4096 = 408
12 2.5ms——180 degrees: 2.5/20*4096 =510
13 */
14
15 #include <Wire.h>
16 #include <Adafruit_PWMServoDriver.h>
17
18 #define FREQ 50 // Analog servos run at ~50 Hz updates
19 #define PWM_Max 510
20 #define PWM_Min 102
21 #define Angle_Range 180.0
22
23 // called this way, PCA9685 uses the address 0x5F
24 Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x5F);
25
26 void setup() {
27   pwm.begin();
28   pwm.setPWMFreq(50); // Set PWM frequency to 50Hz
29 }
30

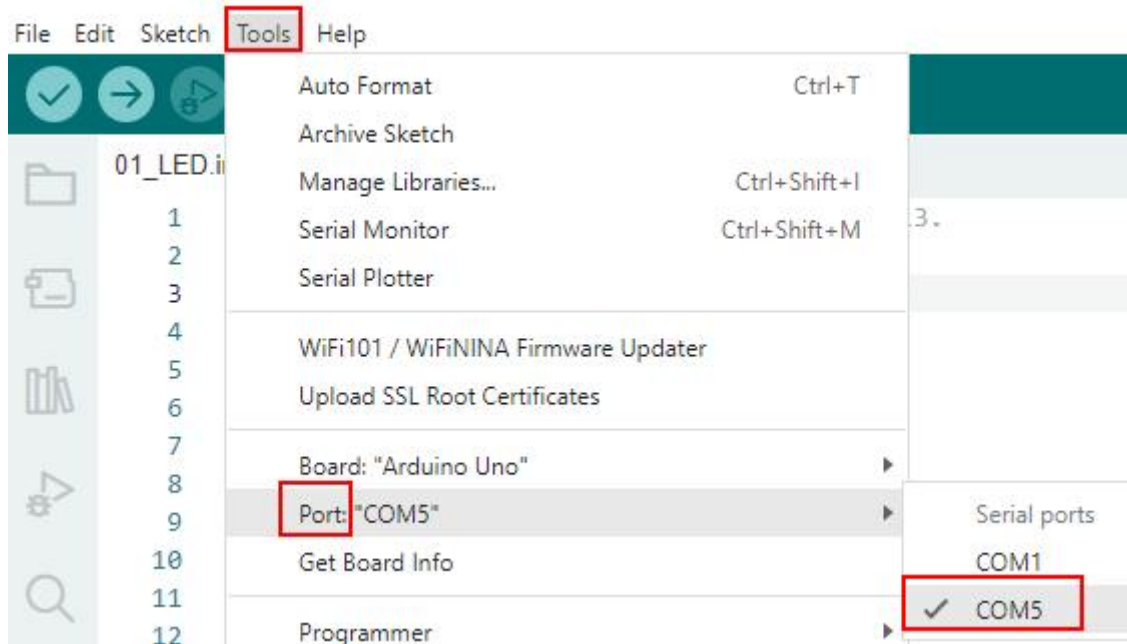
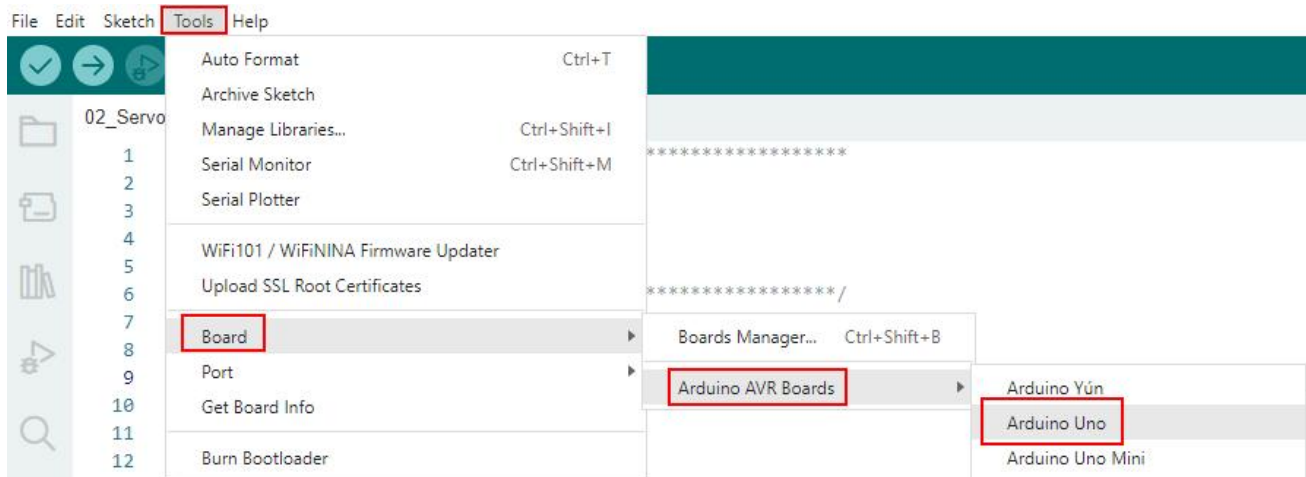
```


3. Select development board and serial port.

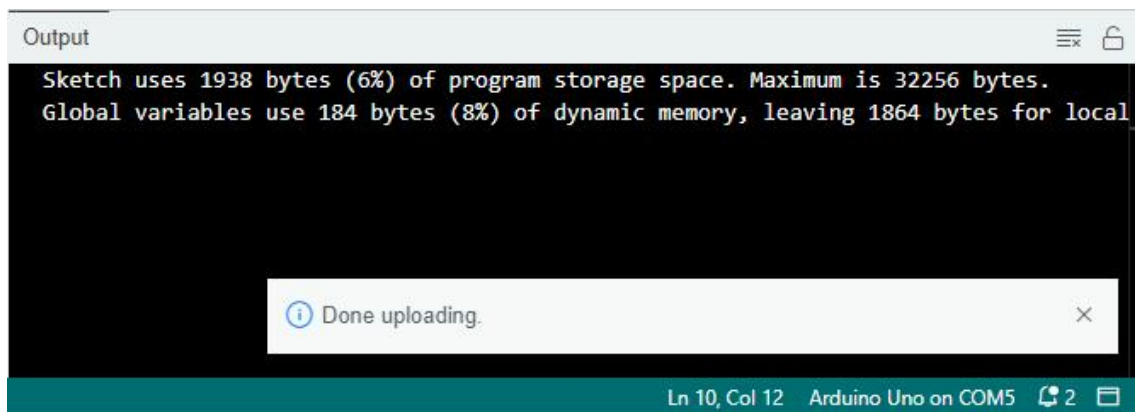
Board: Tools--->Board--->Arduino AVR Boards--->Arduino Uno

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.



5. After successfully running the program, The servo will turn back and forth.

3.6 Code

```
1. #include <Wire.h>
2. #include <Adafruit_PWMServoDriver.h>
3.
4. #define FREQ 50 // Analog servos run at ~50 Hz updates
5. #define PWM_Max 510
6. #define PWM_Min 102
7. #define Angle_Range 180.0
8.
9. // called this way, PCA9685 uses the address 0x5F
10. Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x5F);
11.
12. void setup() {
13.   pwm.begin();
14.   pwm.setPWMFreq(50); // Set PWM frequency to 50Hz
15. }
16.
17. void loop() {
18.   setAngle(6, 0, 90); // CH6 servo turn to 90°
19.   setAngle(7, 0, 90); // CH7 servo turn to 90°
20.   delay(1000); // delay 1s.
21.
22.   setAngle(6, 0, 0); // turn to 0°
23.   setAngle(7, 0, 0);
```

```
24. delay(1000);
25.
26. setAngle(6, 0, 180); // rurn to 180°
27. setAngle(7, 0, 180);
28. delay(1000);
29. }
30.
31. // Convert PWM value to angle value.
32. int setAngle(int num, int start, int angle){
33.     // num: servo channel.
34.     // start: starting point of the pulse signal.
35.     // angle: servo angle.
36.     int pwm_value = int((PWM_Max - PWM_Min)/Angle_Range)*angle);
37.     Serial.println(pwm_value);
38.     pwm.setPWM(num, start, pwm_value);
39. }
```

PCA9685 can set the update frequency, and the time base pulse period of 20ms is equivalent to 50HZ update frequency. PCA9685 uses a 12-bit register to control the PWM ratio. For 0.5ms, it is equivalent to a register value of $0.5/20 \times 4096 = 102$. And so on as follows:

0.5ms———0 degrees: $0.5/20 \times 4096 = 102$

1.0ms———45 degrees: $1/20 \times 4096 = 204$

1.5ms———90 degrees: $1.5/20 \times 4096 = 306$

2.0ms———135 degrees: $2/20 \times 4096 = 408$

2.5ms———180 degrees: $2.5/20 \times 4096 = 510$