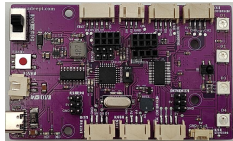




Lesson 4 How to Control a DC Motor

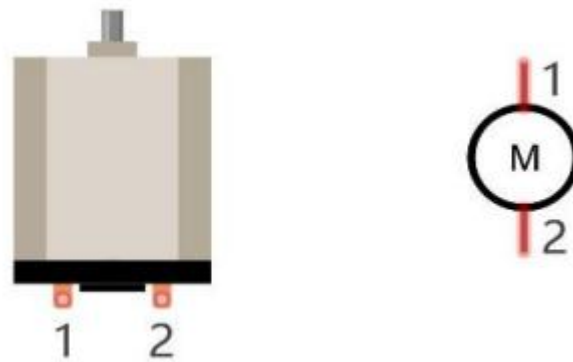
In this lesson, we will learn how to control the Motor.

4.1 Components used in this course

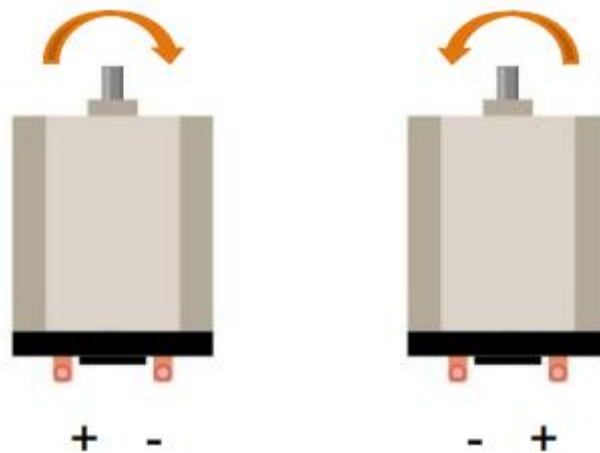
Components	Quantity	Picture
Adeept Robot Control Board	1	
Type-C USB Cable	1	
DC Motor	1	

4.2 The introduction of the Motor

Our products use DC motor as a power device. A motor is a device that converts electrical energy into mechanical energy. Motor consists of two parts: stator and rotor. When motor works, the stationary part is stator, and the rotating part is rotor. Stator is usually the outer case of motor, and it has terminals to connect to the power. Rotor is usually the shaft of motor, and can drive other mechanical devices to run. The schematic below is a small DC motor with two pins.



When a motor gets connected to the power supply, it will rotate in one direction. Reverse the polarity of power supply, then the motor rotates in opposite direction.

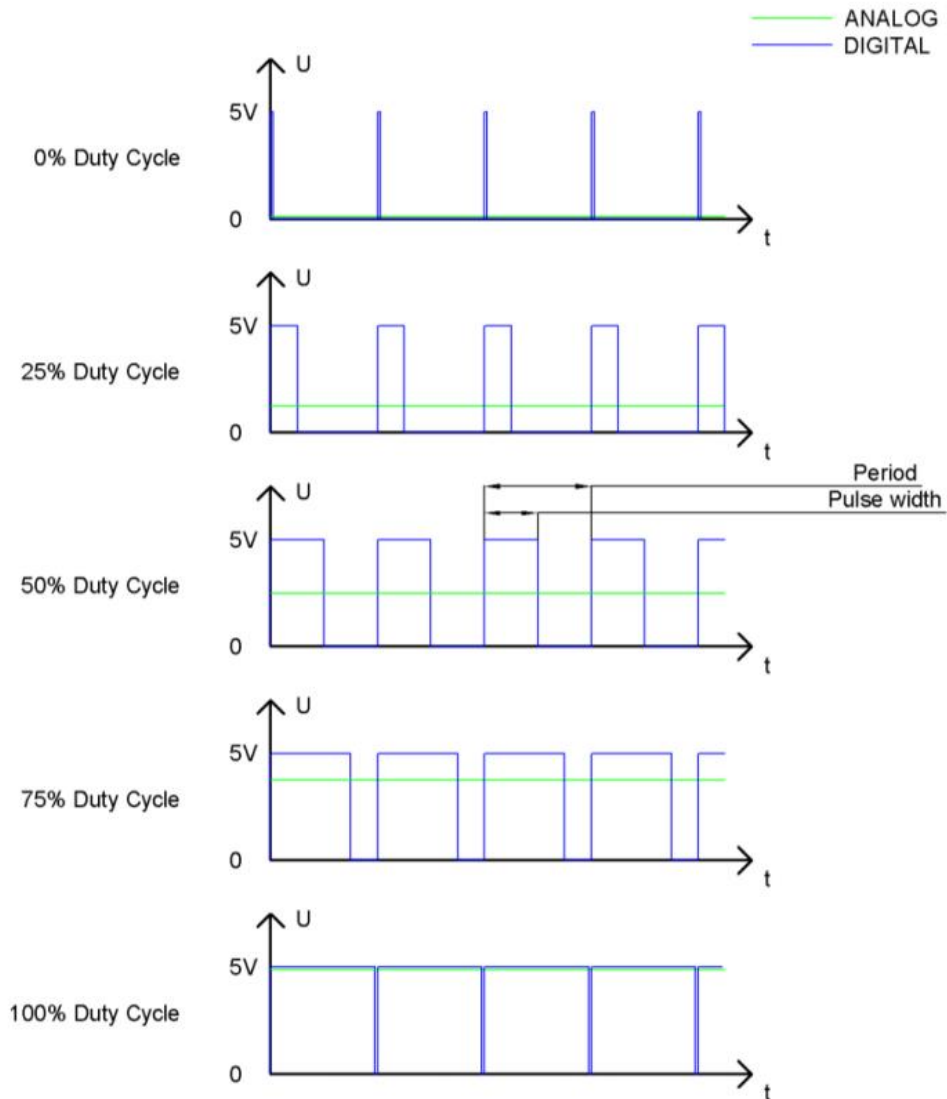


PWM

PWM, Pulse Width Modulation, uses digital pins to send certain frequencies of square waves, that is, the output of high levels and low levels, which alternately last for a while. The total time for each set of high levels and low levels is generally fixed, which is called the period (the reciprocal of the period is frequency). The time of high level outputs are generally called “pulse width”, and the duty cycle is the percentage of the ratio of pulse duration, or pulse width (PW) to the total period (T) of the waveform.

The longer the output of high levels last, the larger the duty cycle and the higher the corresponding voltage in analog signal will be. The following figures show how the analogs

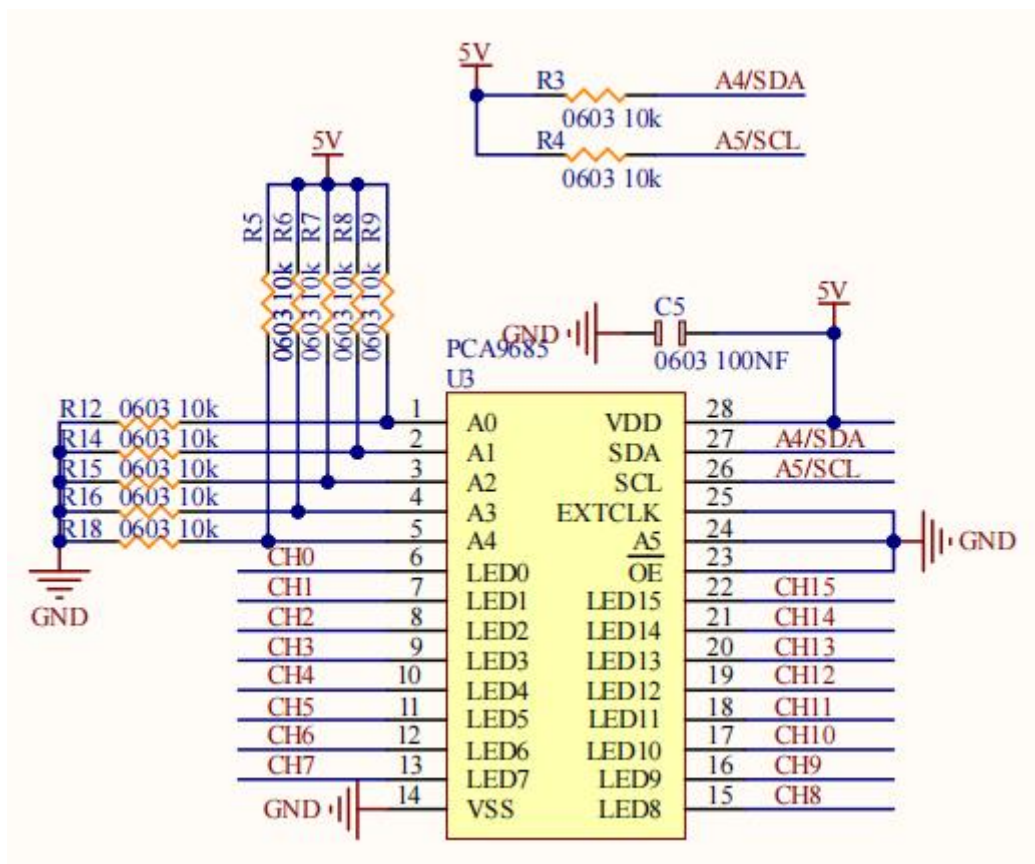
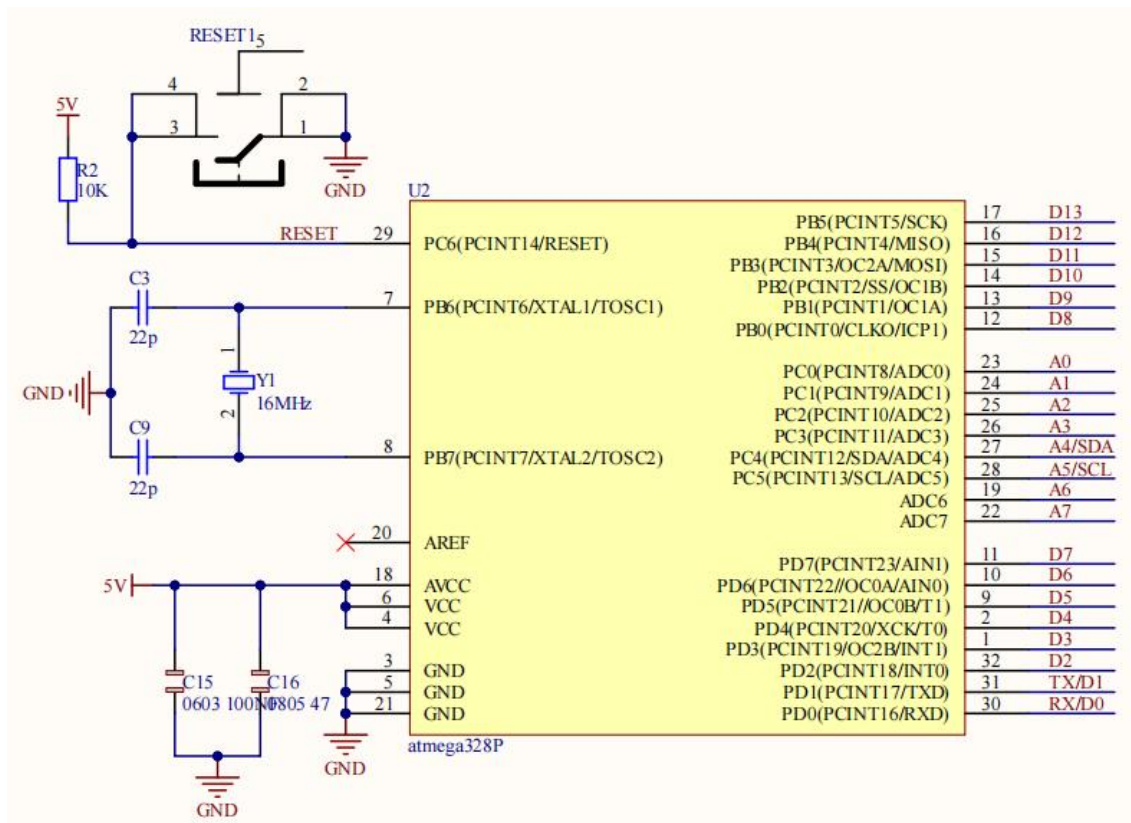
signal voltage vary between 0V-5V(high level is 5V) corresponding to the pulse width 0%-100%:

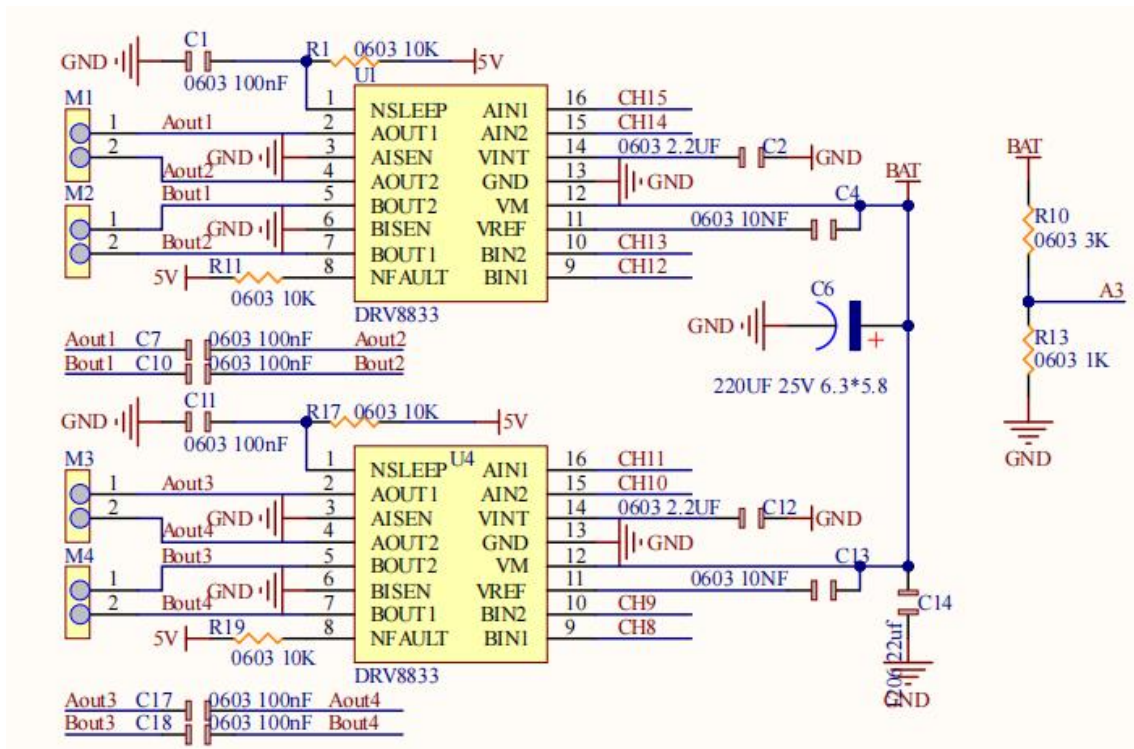


The longer the PWM duty cycle is, the higher the output power will be. Now that we understand this relationship, we can use PWM to control the brightness of an LED or the speed of DC motor and so on.

Extend the Arduino pin interface using PCA9685. Use DRV8833 as the motor driver chip.

Circuit schematic:





4.3 Wiring diagram

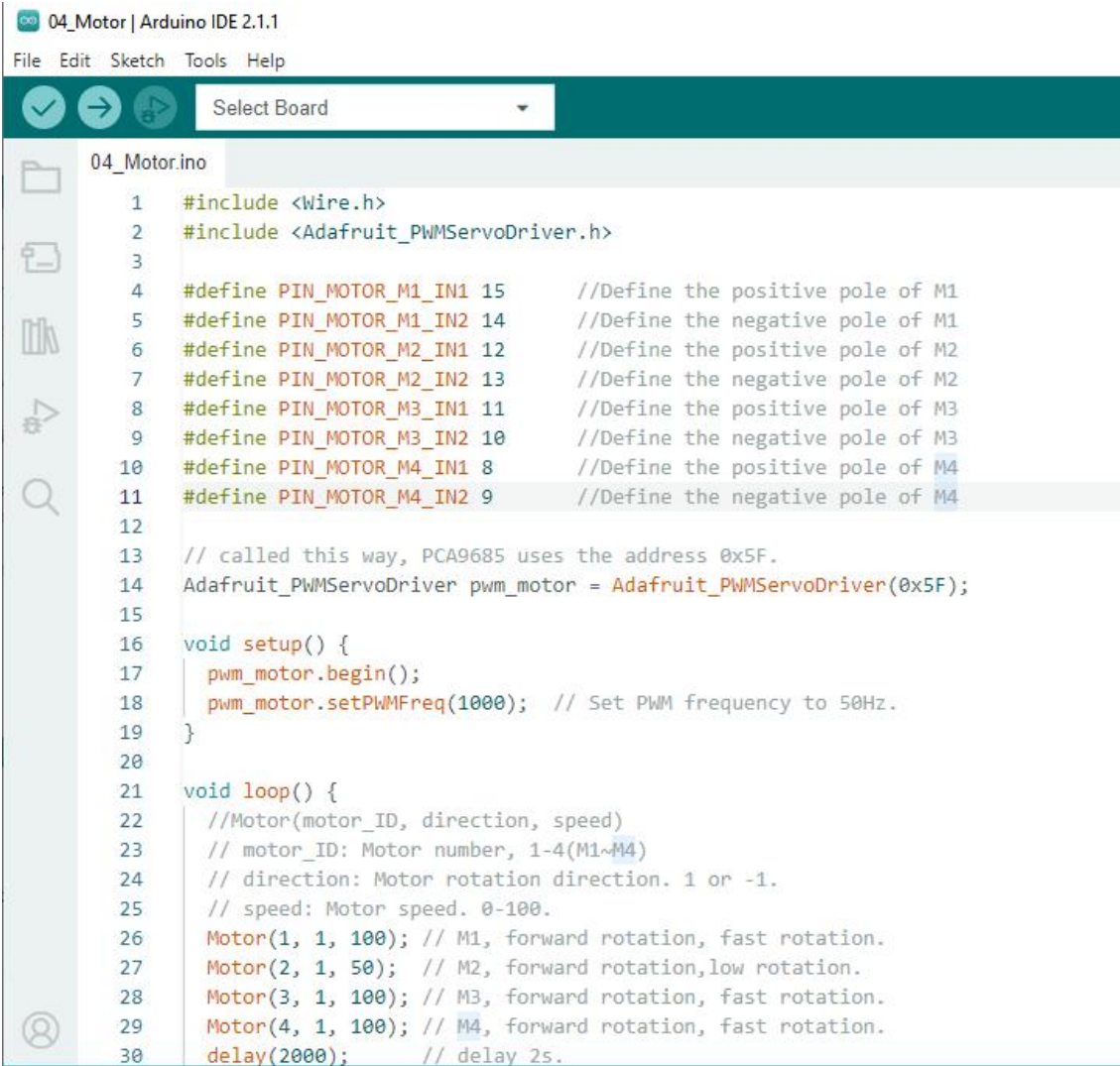
Connect the motor to the M1/M2/M3/M4 motor interface.

Figure as below:



4.4 How to control Motor

1. Connect your computer and Adeept Robot Control Board (Arduino Board) with a USB cable.
2. Open "04_Motor" folder in `/Code`, double-click `"04_Motor.ino"`.



```
04_Motor | Arduino IDE 2.1.1
File Edit Sketch Tools Help

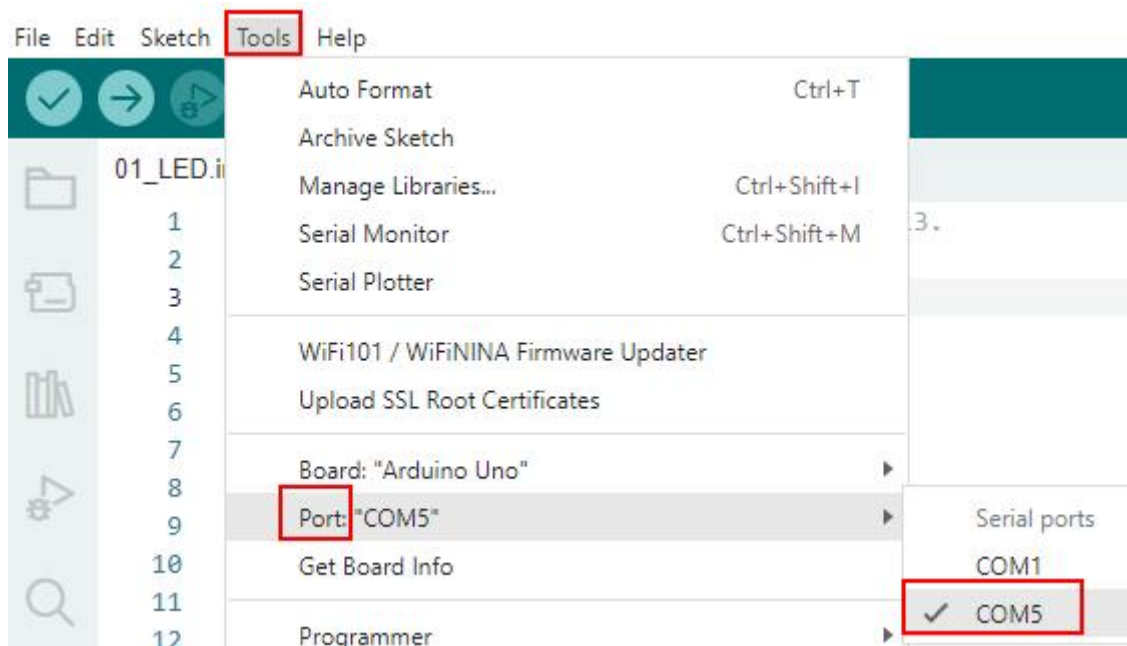
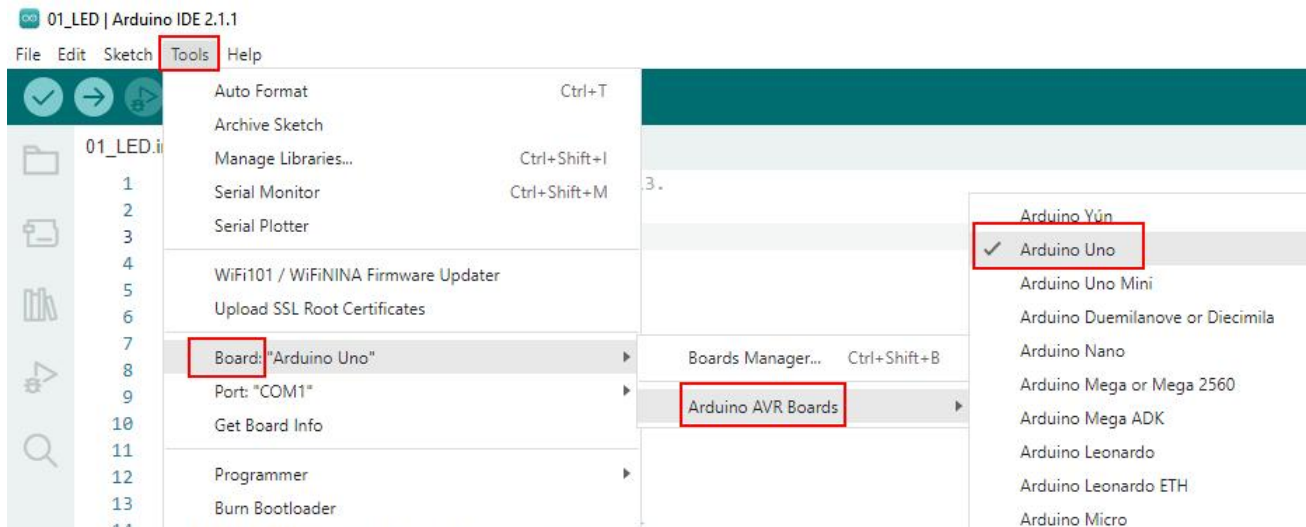
04_Motor.ino
1  #include <Wire.h>
2  #include <Adafruit_PWMServoDriver.h>
3
4  #define PIN_MOTOR_M1_IN1 15    //Define the positive pole of M1
5  #define PIN_MOTOR_M1_IN2 14    //Define the negative pole of M1
6  #define PIN_MOTOR_M2_IN1 12    //Define the positive pole of M2
7  #define PIN_MOTOR_M2_IN2 13    //Define the negative pole of M2
8  #define PIN_MOTOR_M3_IN1 11    //Define the positive pole of M3
9  #define PIN_MOTOR_M3_IN2 10    //Define the negative pole of M3
10 #define PIN_MOTOR_M4_IN1 8     //Define the positive pole of M4
11 #define PIN_MOTOR_M4_IN2 9     //Define the negative pole of M4
12
13 // called this way, PCA9685 uses the address 0x5F.
14 Adafruit_PWMServoDriver pwm_motor = Adafruit_PWMServoDriver(0x5F);
15
16 void setup() {
17   pwm_motor.begin();
18   pwm_motor.setPWMFreq(1000); // Set PWM frequency to 50Hz.
19 }
20
21 void loop() {
22   //Motor(motor_ID, direction, speed)
23   // motor_ID: Motor number, 1-4(M1~M4)
24   // direction: Motor rotation direction. 1 or -1.
25   // speed: Motor speed. 0-100.
26   Motor(1, 1, 100); // M1, forward rotation, fast rotation.
27   Motor(2, 1, 50);  // M2, forward rotation, low rotation.
28   Motor(3, 1, 100); // M3, forward rotation, fast rotation.
29   Motor(4, 1, 100); // M4, forward rotation, fast rotation.
30   delay(2000);      // delay 2s.
```


3. Select development board and serial port.

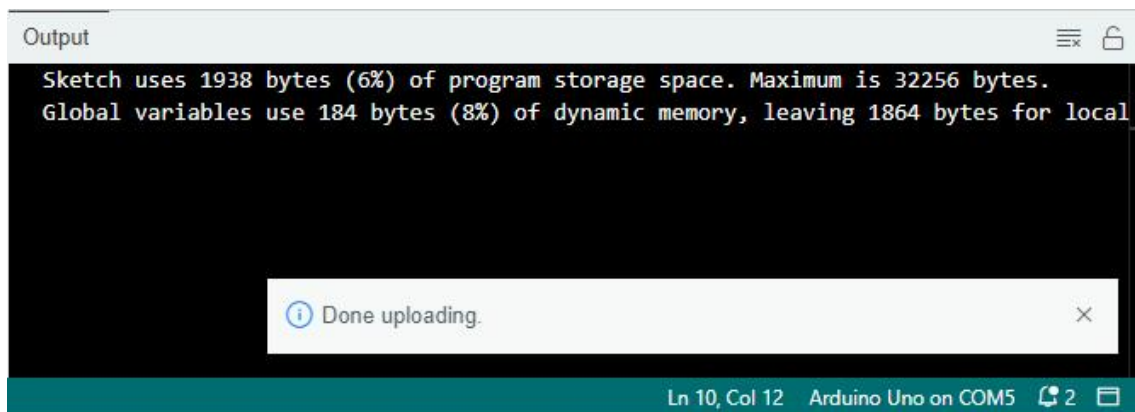
Board: Tools--->Board--->Arduino AVR Boards--->Arduino Uno

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 motor on the M1-M4 interface will cycle rotate : forward for 2 seconds, stop for 1 second, reverse for 2 seconds, stop for 1 second.

4.5 Code

```

1. #include <Wire.h>
2. #include <Adafruit_PWMServoDriver.h>
3.
4. #define PIN_MOTOR_M1_IN1 15      //Define the positive pole of M1
5. #define PIN_MOTOR_M1_IN2 14      //Define the negative pole of M1
6. #define PIN_MOTOR_M2_IN1 12      //Define the positive pole of M2
7. #define PIN_MOTOR_M2_IN2 13      //Define the negative pole of M2
8. #define PIN_MOTOR_M3_IN1 11      //Define the positive pole of M3
9. #define PIN_MOTOR_M3_IN2 10      //Define the negative pole of M3
10. #define PIN_MOTOR_M4_IN1 8       //Define the positive pole of M4
11. #define PIN_MOTOR_M4_IN2 9       //Define the negative pole of M4
12.
13. // called this way, PCA9685 uses the address 0x5F.
14. Adafruit_PWMServoDriver pwm_motor = Adafruit_PWMServoDriver(0x5F);
15.
16. void setup() {
17.   pwm_motor.begin();
18.   pwm_motor.setPWMFreq(1000); // Set PWM frequency to 50Hz.
19. }
20.
21. void loop() {
22.   //Motor(motor_ID, direction, speed)

```

```
23. // motor_ID: Motor number, 1-4(M1~M4)
24. // direction: Motor rotation direction. 1 or -1.
25. // speed: Motor speed. 0-100.
26. Motor(1, 1, 100); // M1, forward rotation, fast rotation.
27. Motor(2, 1, 50); // M2, forward rotation, low rotation.
28. Motor(3, 1, 100); // M3, forward rotation, fast rotation.
29. Motor(4, 1, 100); // M4, forward rotation, fast rotation.
30. delay(2000); // delay 2s.
31.
32. Motor(1, 1, 0); // stop 1s.
33. Motor(2, 1, 0);
34. Motor(3, 1, 0);
35. Motor(4, 1, 0);
36. delay(1000);
37.
38. Motor(1, -1, 100); // reverse rotation 2s.
39. Motor(2, -1, 50);
40. Motor(3, -1, 100);
41. Motor(4, -1, 100);
42. delay(2000);
43.
44. Motor(1, 1, 0); // stop 1s.
45. Motor(2, 1, 0);
46. Motor(3, 1, 0);
47. Motor(4, 1, 0);
48. delay(1000);
49. }
50.
51. // Convert motor speed to PWM value.
52. void motorPWM(int channel, int motor_speed){
53.     motor_speed = constrain(motor_speed, 0, 100);
54.     int motor_pwm = map(motor_speed, 0, 100, 0, 4095);
55.     if (motor_pwm == 4095){
56.         pwm_motor.setPWM(channel, 4096, 0);
57.     }
58.     else if (motor_pwm == 0){
59.         pwm_motor.setPWM(channel, 0, 4096);
60.     }
61.     else{
62.         pwm_motor.setPWM(channel, 0, motor_pwm);
63.         // pwm_motor.setPWM(channel, 0, 4095 - motor_pwm);
64.     }
```

```
65. }
66.
67. // Control motor rotation.
68. void Motor(int Motor_ID, int dir, int Motor_speed){
69.     if(dir > 0){dir = 1;}
70.     else {dir = -1;}
71.
72.     if (Motor_ID == 1){
73.         if (dir == 1){
74.             motorPWM(PIN_MOTOR_M1_IN1, 0);
75.             motorPWM(PIN_MOTOR_M1_IN2, Motor_speed);
76.         }
77.         else{
78.             motorPWM(PIN_MOTOR_M1_IN1, Motor_speed);
79.             motorPWM(PIN_MOTOR_M1_IN2, 0);
80.         }
81.     }
82.     else if (Motor_ID == 2){
83.         if (dir == 1){
84.             motorPWM(PIN_MOTOR_M2_IN1, 0);
85.             motorPWM(PIN_MOTOR_M2_IN2, Motor_speed);
86.         }
87.         else{
88.             motorPWM(PIN_MOTOR_M2_IN1, Motor_speed);
89.             motorPWM(PIN_MOTOR_M2_IN2, 0);
90.         }
91.     }
92.     else if (Motor_ID == 3){
93.         if (dir == 1){
94.             motorPWM(PIN_MOTOR_M3_IN1, 0);
95.             motorPWM(PIN_MOTOR_M3_IN2, Motor_speed);
96.         }
97.         else{
98.             motorPWM(PIN_MOTOR_M3_IN1, Motor_speed);
99.             motorPWM(PIN_MOTOR_M3_IN2, 0);
100.        }
101.    }
102.    else if (Motor_ID == 4){
103.        if (dir == 1){
104.            motorPWM(PIN_MOTOR_M4_IN1, 0);
105.            motorPWM(PIN_MOTOR_M4_IN2, Motor_speed);
106.        }
```

```
107.     else{
108.         motorPWM(PIN_MOTOR_M4_IN1, Motor_speed);
109.         motorPWM(PIN_MOTOR_M4_IN2, 0);
110.     }
111. }
112. }
```