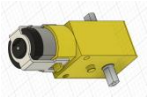
Lesson 9 Control the DC Motor to Work for Ordinary Wheels

9.1 Overview

This course focuses on DC motor control and aims to guide learners in mastering the methods of using Adeept Robot Control Board and related components to achieve DC motor operation control. It covers hardware connection, principle analysis, code writing and debugging, and helps learners to deeply understand and practice the application of DC motors in practical projects.

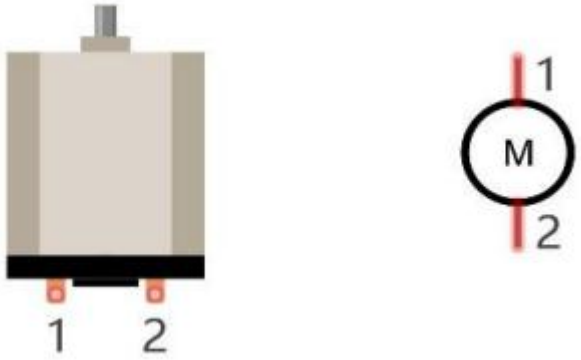
9.2 Required Components

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

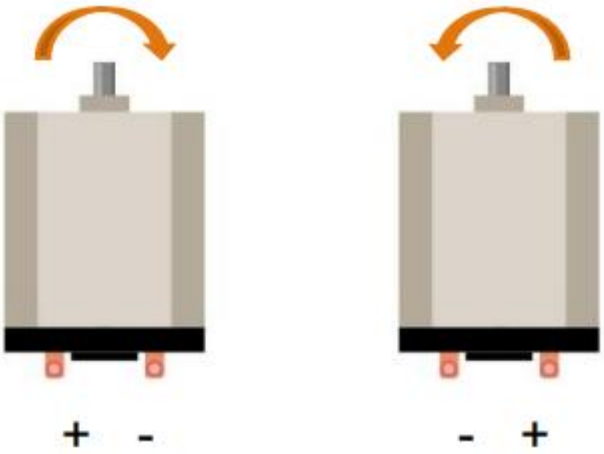
9.3 Principle Introduction

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.



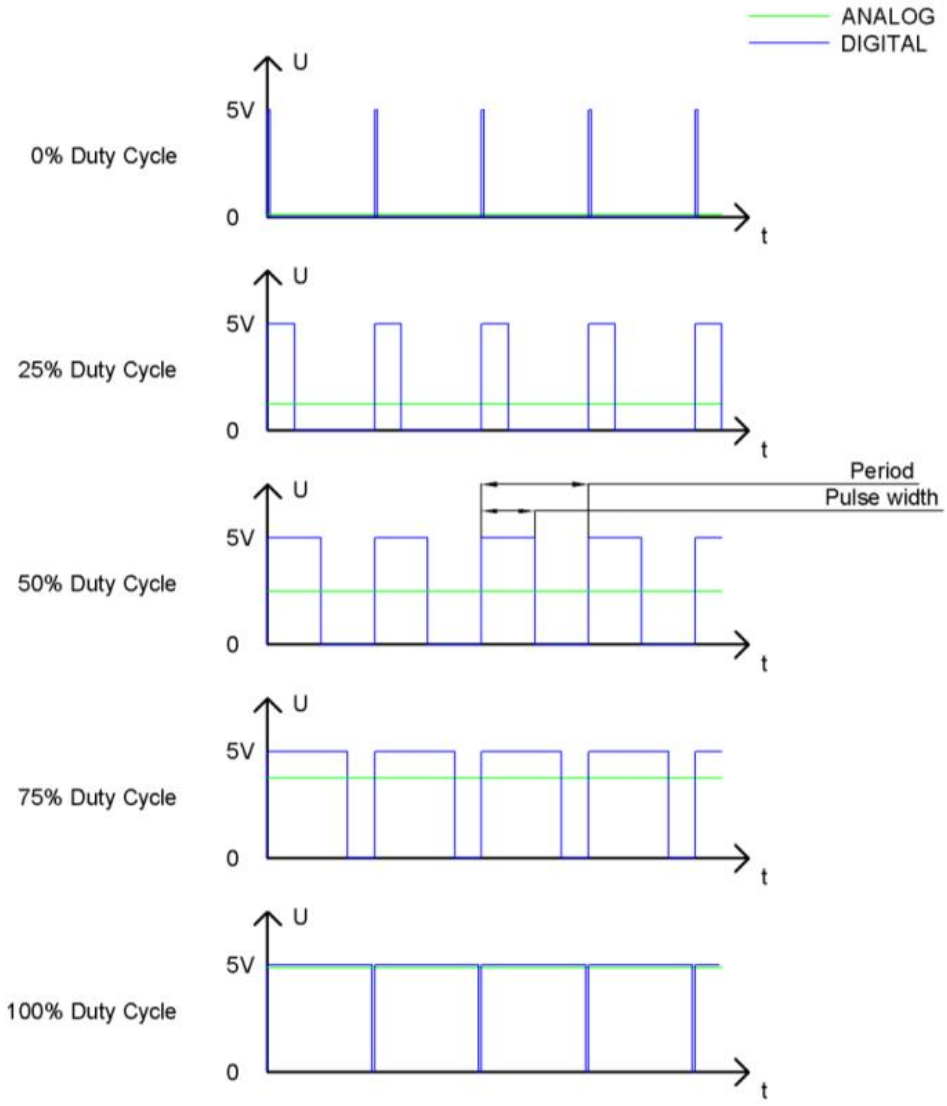
When a motor gets connected to the power supply, it will rotate in one direction. Reverse the polarity of power supply, then the motor rotates in opposite direction.



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.

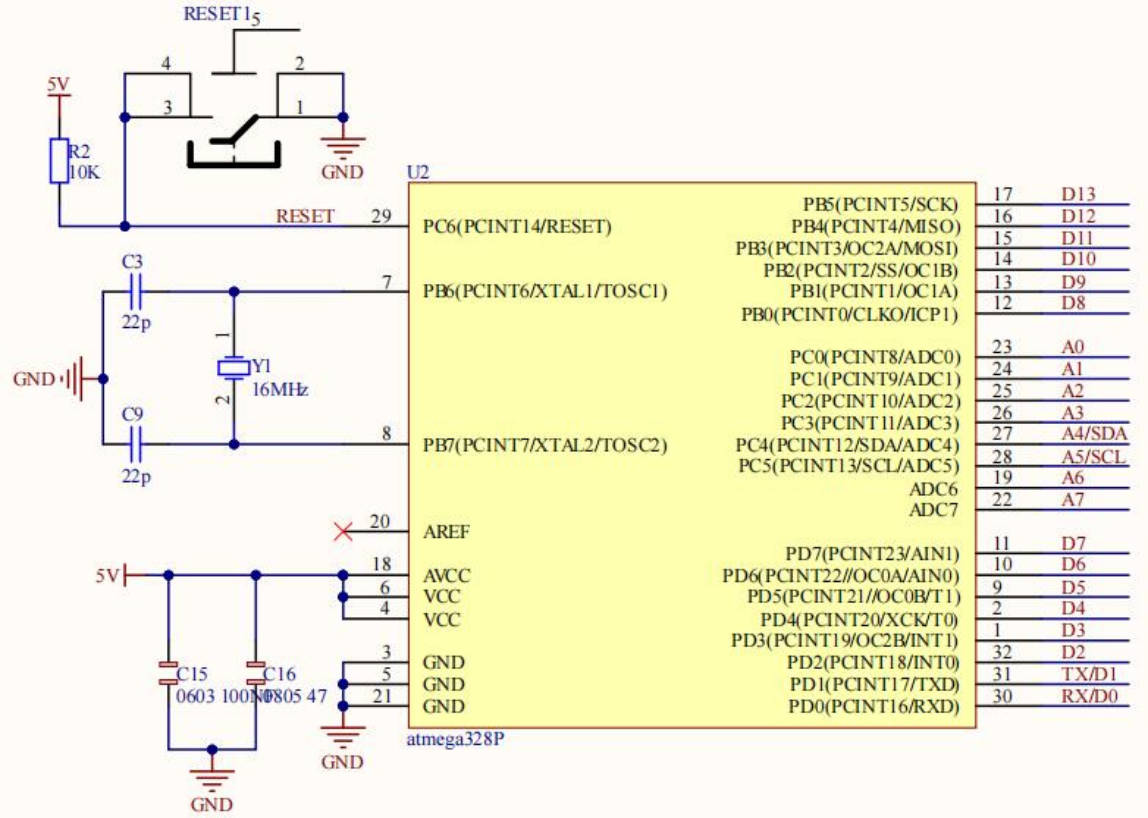
The longer the output of high levels last, the larger the duty cycle and the higher the corresponding voltage in analog signal will be. The following figures show how the analogs signal voltage vary between 0V-5V(high level is 5V) corresponding to the pulse width 0%-100%:

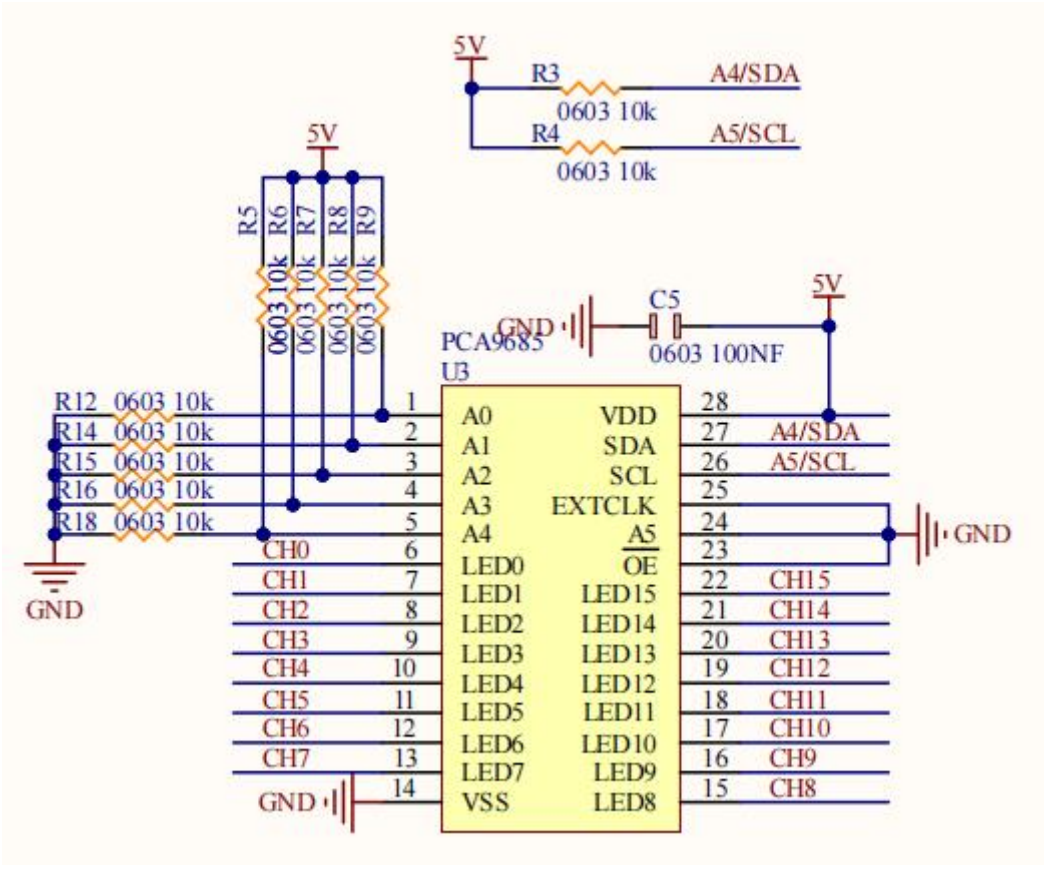


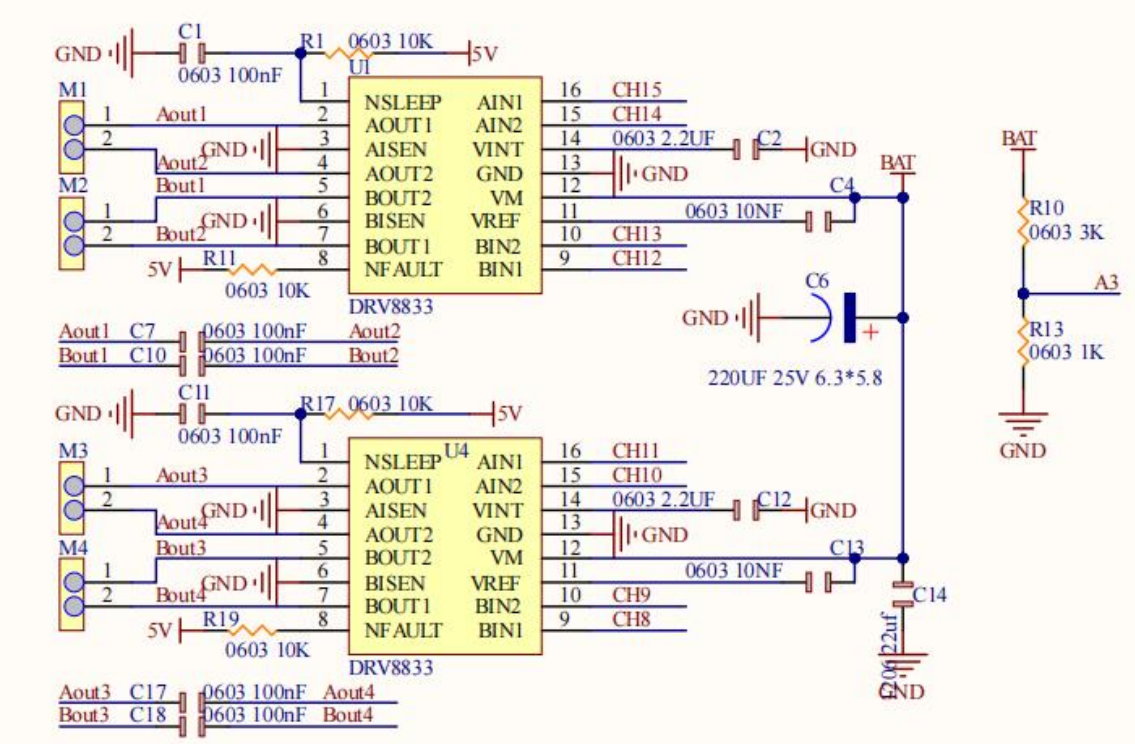
The longer the PWM duty cycle is, the higher the output power will be. Now that we understand this relationship, we can use PWM to control the brightness of an LED or the speed of DC motor and so on.

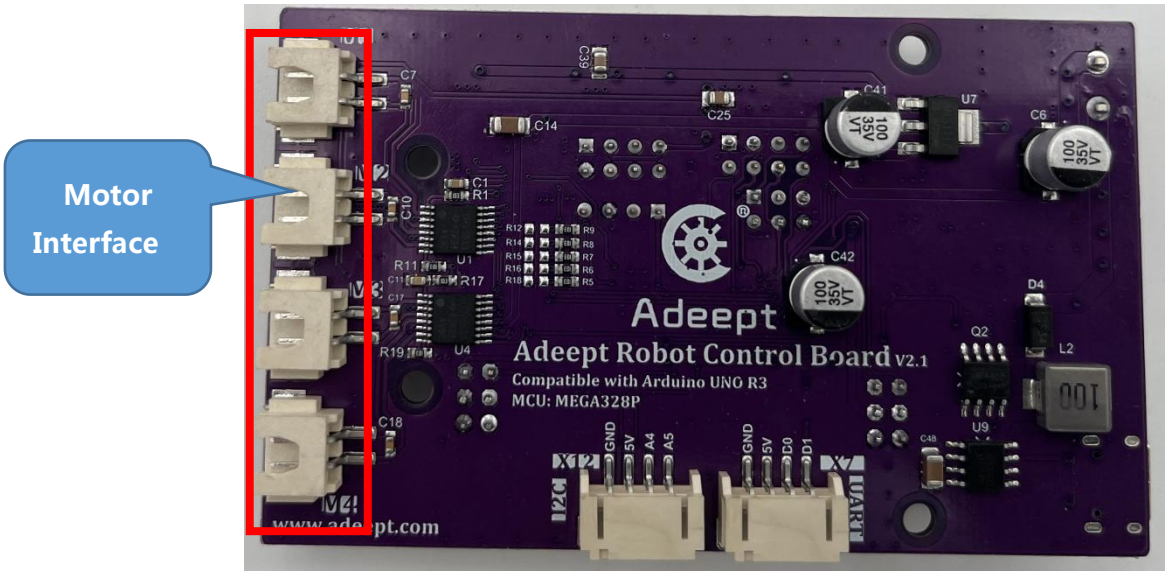
9.4 Wiring Diagram

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


RESET
5V
R2
10K
GND
U2
PC6(PCINT14/RESET)
PB6(PCINT6/XTAL1/TOSC1)
PB7(PCINT7/XTAL2/TOSC2)
AREF
AVCC
VCC
VCC
GND
GND
GND
C3
22p
C9
22p
Y1
16MHz
C15
0603 100NF
C16
0805 47
atmega328P
PB5(PCINT5/SCK)
PB4(PCINT4/MISO)
PB3(PCINT3/OC2A/MOSI)
PB2(PCINT2/SS/OC1B)
PB1(PCINT1/OC1A)
PB0(PCINT0/CLKO/ICP1)
PC0(PCINT8/ADC0)
PC1(PCINT9/ADC1)
PC2(PCINT10/ADC2)
PC3(PCINT11/ADC3)
PC4(PCINT12/SDA/ADC4)
PC5(PCINT13/SCL/ADC5)
ADC6
ADC7
PD7(PCINT23/AIN1)
PD6(PCINT22//OC0A/AIN0)
PD5(PCINT21//OC0B/T1)
PD4(PCINT20/XCK/T0)
PD3(PCINT19/OC2B/INT1)
PD2(PCINT18/INT0)
PD1(PCINT17/TXD)
PD0(PCINT16/RXD)
D13
D12
D11
D10
D9
D8
A0
A1
A2
A3
A4/SDA
A5/SCL
A6
A7
D7
D6
D5
D4
D3
D2
TX/D1
RX/D0


5V
R3
0603 10k
A4/SDA
R4
0603 10k
A5/SCL
R5
R6
R7
R8
R9
0603 10k
C5
0603 100NF
GND
PCA9685
U3
R12 0603 10k
R14 0603 10k
R15 0603 10k
R16 0603 10k
R18 0603 10k
A0
A1
A2
A3
A4
LED0
LED1
LED2
LED3
LED4
LED5
LED6
LED7
VSS
VDD
SDA
SCL
EXTCLK
A5
OE
LED15
LED14
LED13
LED12
LED11
LED10
LED9
LED8
CH0
CH1
CH2
CH3
CH4
CH5
CH6
CH7
CH15
CH14
CH13
CH12
CH11
CH10
CH9
CH8

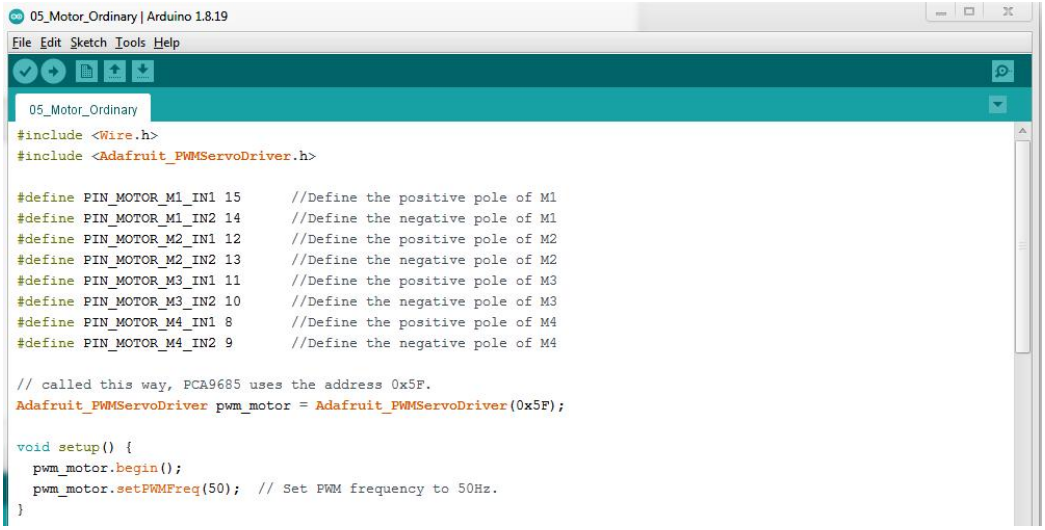




9.5 Demonstration

1. Connect your computer and Adeept Robot Control Board (Arduino Board) with a USB cable.

2. Open "**05_Motor_Ordinary**" folder in "/Code", double-click "05_Motor_Ordinary.ino".

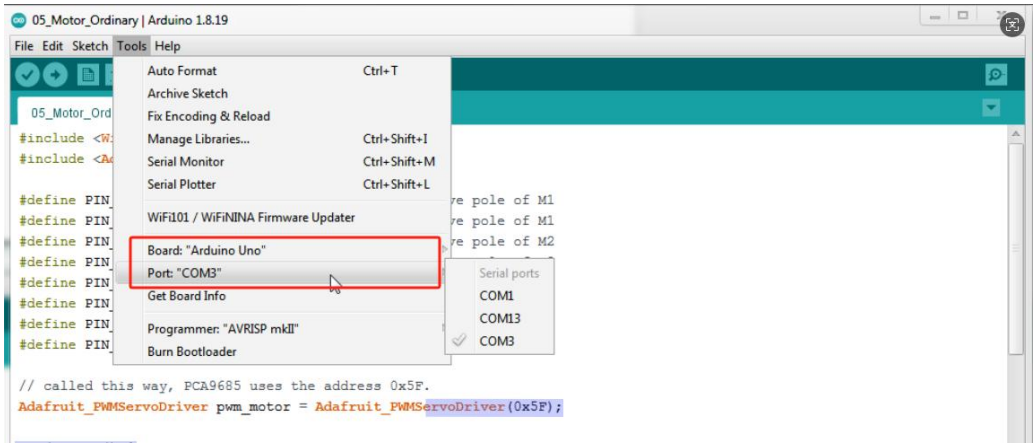


3. Select development board and serial port.

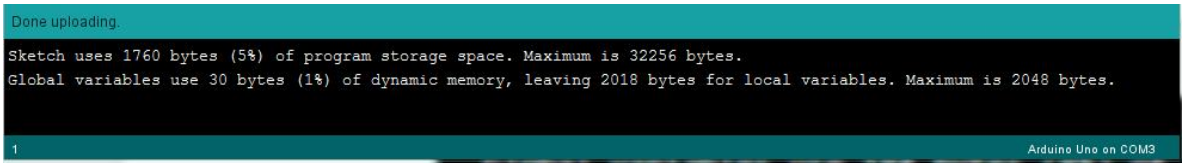
Board: Tools--->Board--->Arduino AVR Boards--->Arduino Uno

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. The DC motor operates according to the program settings, which sequentially execute the actions of forward full speed rotation for 1 second, stop for 1 second, reverse full speed rotation for 1 second, and stop for 1 second.

9.6 Code

Complete code refer to 05_Motor_Ordinary.ino

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

#define PIN_MOTOR_M1_IN1 15      //Define the positive pole of M1
#define PIN_MOTOR_M1_IN2 14      //Define the negative pole of M1
#define PIN_MOTOR_M2_IN1 12      //Define the positive pole of M2
#define PIN_MOTOR_M2_IN2 13      //Define the negative pole of M2
#define PIN_MOTOR_M3_IN1 11      //Define the positive pole of M3
#define PIN_MOTOR_M3_IN2 10      //Define the negative pole of M3
#define PIN_MOTOR_M4_IN1 8       //Define the positive pole of M4
#define PIN_MOTOR_M4_IN2 9       //Define the negative pole of M4

// called this way, PCA9685 uses the address 0x5F.
Adafruit_PWMServoDriver pwm_motor = Adafruit_PWMServoDriver(0x5F);

void setup() {
  pwm_motor.begin();
  pwm_motor.setPWMFreq(50);  // Set PWM frequency to 50Hz.
}

void loop() {
  //Motor(motor_ID, direction, speed)
  // motor_ID: Motor number, 1-4(M1~M4)
  // direction: Motor rotation direction. 1 or -1.
  // speed: Motor speed. 0-100.
  Motor(1, 1, 100); // M1, forward rotation, fast rotation.
  Motor(2, 1, 100);  // M2, forward rotation,low rotation.
  Motor(3, 1, 100); // M3, forward rotation, fast rotation.
  Motor(4, 1, 100); // M4, forward rotation, fast rotation.
  delay(2000);      // delay 2s.

  Motor(1, 1, 0);   // stop 1s.
  Motor(2, 1, 0);
  Motor(3, 1, 0);
  Motor(4, 1, 0);
  delay(1000);

  Motor(1,-1, 100);  // reverse rotation 2s.
  Motor(2,-1, 100);
  Motor(3,-1, 100);
  Motor(4,-1, 100);
  delay(2000);

  Motor(1, 1, 0);    // stop 1s.
  Motor(2, 1, 0);
```

```
046   Motor(3, 1, 0);
047   Motor(4, 1, 0);
048   delay(1000);
049
050 }
051
052 // Convert motor speed to PWM value.
053 void motorPWM(int channel, int motor_speed){
054   motor_speed = constrain(motor_speed, 0, 100);
055   int motor_pwm = map(motor_speed, 0, 100, 0, 4095);
056   if (motor_pwm == 4095){
057     pwm_motor.setPWM(channel, 4096, 0);
058   }
059   else if (motor_pwm == 0){
060     pwm_motor.setPWM(channel, 0, 4096);
061   }
062   else{
063     pwm_motor.setPWM(channel, 0, motor_pwm);
064     // pwm_motor.setPWM(channel, 0, 4095 - motor_pwm);
065   }
066 }
067
068 // Control motor rotation.
069 void Motor(int Motor_ID, int dir, int Motor_speed){
070   if(dir > 0){dir = 1;}
071   else {dir = -1;}
072
073   if (Motor_ID == 1){
074     if (dir == 1){
075       motorPWM(PIN_MOTOR_M1_IN1, 0);
076       motorPWM(PIN_MOTOR_M1_IN2, Motor_speed);
077     }
078     else{
079       motorPWM(PIN_MOTOR_M1_IN1, Motor_speed);
080       motorPWM(PIN_MOTOR_M1_IN2, 0);
081       }
082   }
083   else if (Motor_ID == 2){
084     if (dir == 1){
085       motorPWM(PIN_MOTOR_M2_IN1, 0);
086       motorPWM(PIN_MOTOR_M2_IN2, Motor_speed);
087     }
088     else{
089       motorPWM(PIN_MOTOR_M2_IN1, Motor_speed);
090       motorPWM(PIN_MOTOR_M2_IN2, 0);
091       }
092   }
093   else if (Motor_ID == 3){
094     if (dir == 1){
095       motorPWM(PIN_MOTOR_M3_IN1, 0);
096       motorPWM(PIN_MOTOR_M3_IN2, Motor_speed);
097     }
098     else{
099       motorPWM(PIN_MOTOR_M3_IN1, Motor_speed);
100       motorPWM(PIN_MOTOR_M3_IN2, 0);
101       }
```

```
  }
  else if (Motor_ID == 4){
    if (dir == 1){
      motorPWM(PIN_MOTOR_M4_IN1, 0);
      motorPWM(PIN_MOTOR_M4_IN2, Motor_speed);
    }
    else{
      motorPWM(PIN_MOTOR_M4_IN1, Motor_speed);
      motorPWM(PIN_MOTOR_M4_IN2, 0);
      }
  }
}
```

Code explanation

Initialization Stage:

Connect the PCA9685 module (address 0x5F) via the I2C protocol. Set the PWM frequency to 50Hz and initialize 4 DC motors: M1 - M4 (bind them to GPIO pins respectively and configure them in slow decay mode).

Loop Control Process:

Stage 1:Motor M1->M4 rotates forward at 100% speed → Run for 1 seconds.

Stage 2:Motor M1-M4 stops running ->Run for 1 second.

Stage 3:Motor M1->M4 rotates backward at 100% speed → Run for 1 seconds.

Stage 4:Motor M1-M4 stops running ->Run for 1 second.