

5.2 Car advance

1. Learning goal:

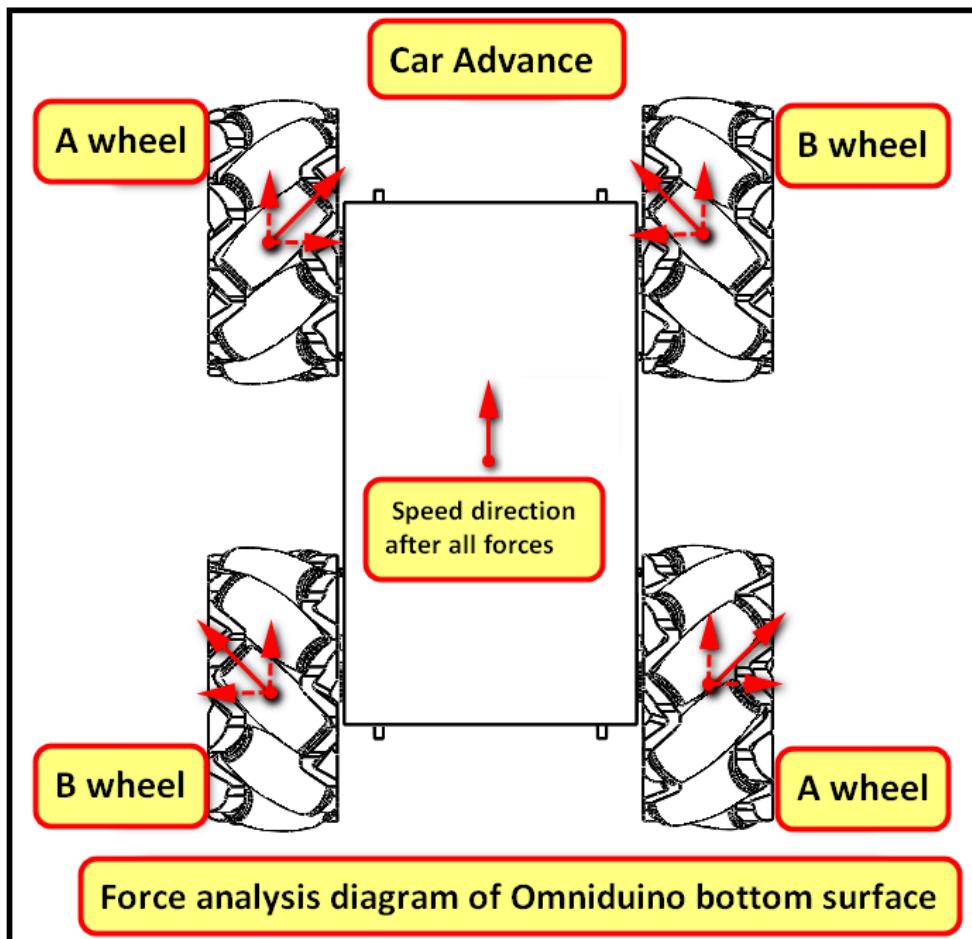
Control car advance and back by button.

2. Experimental phenomena:

After we open the power of switch, car is stop, when we press button K1(side of the car). When we press button again, car will stop.

3. Force Analysis:

According to the characteristics of the Mecanum wheel, if the omniduino car advances, the four wheels must rotate forward. The force analysis diagram is shown below:



3.1 According to the physics, the forces of equal magnitude and opposite directions can cancel each other out. If the A and B wheels forward at the same speed, the A wheel decomposes the right force and the B wheel decomposes the left force will cancel each other out.

According to "F=ma", we can know that the acceleration direction is forward, and the final speed direction is also forward.

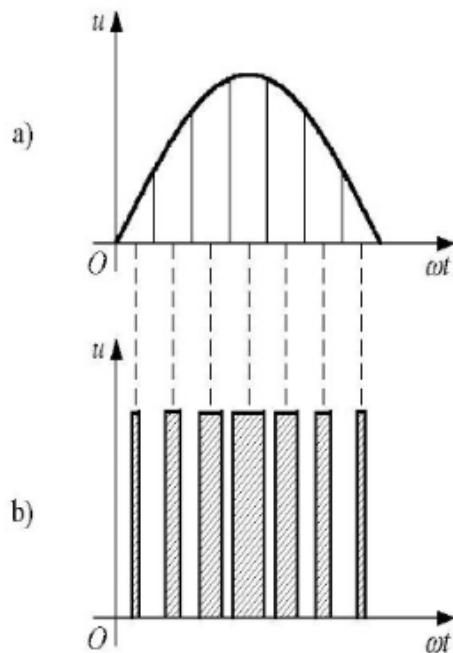
3.2 According to the hardware manual, we can know that PCA9685 controls one wheel by two pins, one pin controls forward rotation and another pin controls reverse rotation.

For example, the pin12 of the PCA9685 chip control M1 reserve and the pin13 control M1 forward.

If the 12th output is 0, the 13th normally outputs the PWM pulse, then the M1 motor rotates forward; if the 12th output is output PWM pulse, the output of No. 13 is 0, then the M1 motor is reversed.

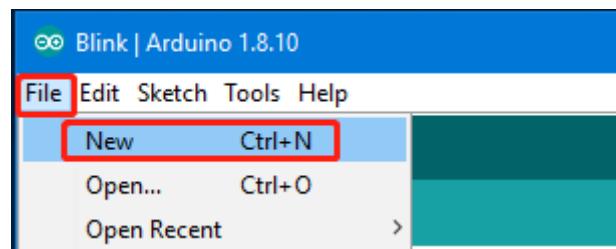
PCA9685PWM controlled component pins		
Pins	Function	Remarks
0~6	no connect	
7	servo angle	0~180
8	M4 motor forward	
9	M4 motor Reserve	Right rear motor
10	M3 motor forward	
11	M3 motor Reserve	Right front motor
12	M1 motor Reserve	
13	M1 motor forward	Left front motor
14	M2 motor Reserve	
15	M2 motor forward	Left rear motor

3.3 PWM refers to pulse width modulation. In the PWM drive control adjustment system, the output is high and low at a fixed frequency, and the length of the "high level 1" and "low level 0" time in one cycle is changed as needed. . In this way, the "duty cycle" is changed to change the average voltage (effective voltage), and then the motor drives the amplified signal to control the motor speed.

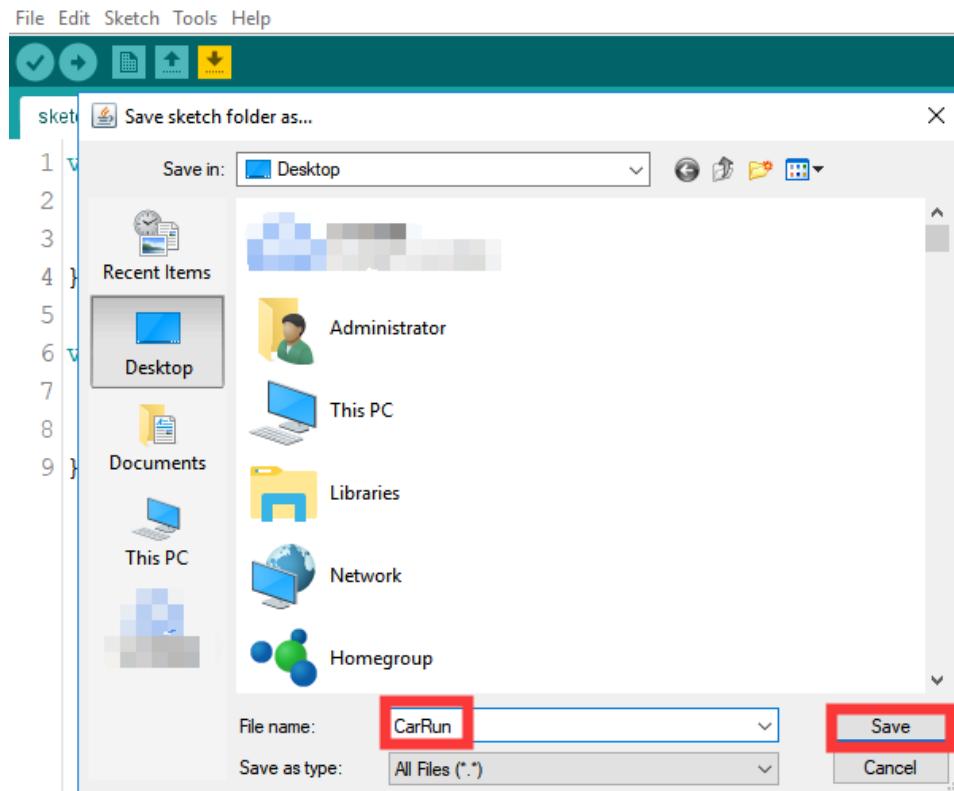


4. Create new project

4.1 Click 【File】-->【New】.



4.2 Press **Ctrl+S** to save and rename Advance. As shown below.



4.3 We can see that there is a Serial folder with **Advance.ino** on the computer desktop.

4.4 We will **Advance.ino** as shown below.

```
void setup() {
    // put your setup code here, to run once:

}

void loop() {
    // put your main code here, to run repeatedly:

}
```

The `setup()` function only runs once when the car is turned on or when the reset button is pressed, and the program for initializing the relevant content can be written;

The `loop()` function is the main loop function of the car and most of the data processing and logic processing are done in this function.

5. Programming

5.1 Car motor controlled by PCA9685, we need to import PCA9685 and I2C library file.

```
//Import library file
#include <Adafruit_PWMServoDriver.h>
#include "Wire.h"
```

5.2 From the hardware manual, you can know that the button connect pin8, and the PCA9685 I2C address is **0x40**.

```
#define KEY_PIN 8      //Define key pin

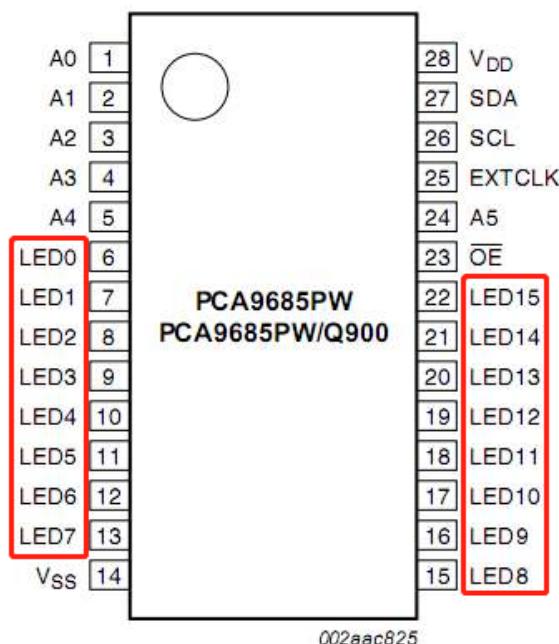
//PCA9685 Initialization
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x40);
```

The external functions of the Adafruit_PWMServoDriver library mainly include the following.

- 1) void begin() function is to open I2C communication and reset PCA9685;
- 2) reset() function is to reset PCA9685;
- 3) setPWMFreq() function is to set the frequency, generally set to 50 or 60;
- 4) the setPWM() function sets the PWM output, which is the most The main function;
- 5) the setPin() function also sets the PWM output, but adds the flip function.

```
public:
    Adafruit_PWMServoDriver(uint8_t addr = 0x40);
    void begin(void);
    void reset(void);
    void setPWMFreq(float freq);
    void setPWM(uint8_t num, uint16_t on, uint16_t off);
    void setPin(uint8_t num, uint16_t val, bool invert=false);
```

5.3 setPWM() function, the first parameter num takes the range 0~15, corresponding to LED0~LED15 in the figure below, the parameter on and off sets the PWM output, the value range is 0~4095.



5.4 Define global variables to save car speed and button status

```
//Car control parameters
int CarSpeedControl = 60;

//Button status
bool button_press = false;
```

5.5 New create run() function to drive car advance, speed range:(40~160).

```
void run(int Speed)
{
    Speed = map(Speed, 0, 160, 0, 2560);
    pwm.setPWM(10, 0, Speed);      //Right rear wheel Forward
    pwm.setPWM(11, 0, 0);
    pwm.setPWM(8, 0, Speed);      //Right front wheel Forward
    pwm.setPWM(9, 0, 0);

    pwm.setPWM(13, 0, Speed);      //Left rear wheel Forward
    pwm.setPWM(12, 0, 0);
    pwm.setPWM(15, 0, Speed);      //Left front wheel Forward
    pwm.setPWM(14, 0, 0);
}
```

Pins	Function
0~6	no connect
7	servo angle
8	M4 motor forward
9	M4 motor Reserve
10	M3 motor forward
11	M3 motor Reserve
12	M1 motor Reserve
13	M1 motor forward
14	M2 motor Reserve
15	M2 motor forward

5.6 New create brake function, car stop

```
void brake()
{
    pwm.setPWM(8, 0, 0);
    pwm.setPWM(9, 0, 0);
    pwm.setPWM(11, 0, 0);
    pwm.setPWM(10, 0, 0);

    pwm.setPWM(12, 0, 0);
    pwm.setPWM(13, 0, 0);
    pwm.setPWM(14, 0, 0);
    pwm.setPWM(15, 0, 0);
}
```

5.7 key scan function

```

void keyscan()
{
    int val;
    val = digitalRead(KEY_PIN); //Read the digital 8-port level
    if (val == LOW)           //When the button is pressed
    {
        delay(10);           //Delayed debounce
        val = digitalRead(KEY_PIN); //Read button status again
        while (val == LOW)
        {
            val = digitalRead(KEY_PIN); //Third read button status
            if (val == HIGH)          //Determine if the button is
            {
                button_press = !button_press;
                return;
            }
        }
    }
}

```

5.8 Initialize the pin mode in the setup() function, turn on the I2C communication function, and initialize the motor to the stop state.

```

void setup() {
    // put your setup code here, to run once:
    pinMode(KEY_PIN, INPUT_PULLUP);
    Wire.begin();
    pwm.begin();
    pwm.setPWMFreq(60); // Analog servos run at ~60 Hz updates
    brake();
}

```

5.9 In the loop() main loop function detection button, the car runs when the button is pressed for the first time, and the car stops when the button is pressed again. Because each time the function button is pressed, button_press will be reversed once.

```

void loop() {
    // put your main code here, to run repeatedly:
    //key scan
    keyscan();

    if (button_press)
    {
        run(CarSpeedControl);
    }
    else
    {
        brake();
    }
}

```

6. Compiling and downloading code

6.1 After the code is written, press Ctrl+S to save, then click the “√” button to compile. If there is no problem, click “→” to upload (the car must be connected to the computer via the USB

cable).

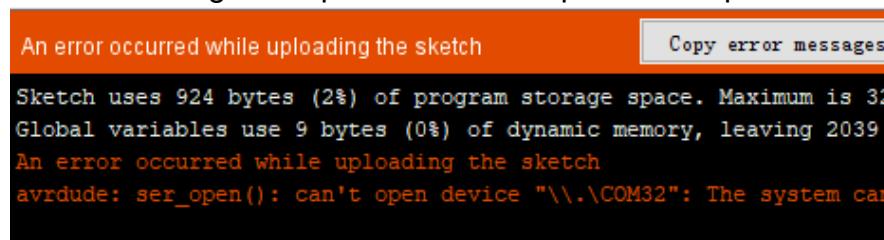


```

File Edit Sketch Tools Help
LED
10 /*
11 //Define LED light(D9)pin
12 #define LED_PIN 5
13
14 void setup() {
15     // put your setup code here, to run once:
16     // set LED pin to output mode
17     pinMode(LED_PIN, OUTPUT);
18 }
19
20 void loop() {
21     // put your main code here, to run repeatedly:
22     digitalWrite(LED_PIN, LOW);           //LED is on
23     delay(500);
24     digitalWrite(LED_PIN, HIGH);          //LED is off
25     delay(500);
26 }

```

6.2 If the compilation passes normally, but the following error occurs during uploading, the reason may be that the wrong serial port or the serial port is occupied.



Solution: Open the device manager to see if there is a serial port with CH340 tag. If not, please restart the Omniduino car, then, re-plug the USB cable or replace a USB cable; If there is a serial port number, we need to close the other serial port or assistant software, avoid serial port occupation, and then re-select the serial port to ArduinoIDE [Tool] --> [Port].

6.3 If there is an error like the following, it means that the library file is missing. Please copy the library file provided by the omniduino omnibus to the library file directory compiled by arduinoIDE.

please refer to **【3.Development Environment Construction】 ---- 【3.4 Add additional library files】**

```
Adafruit_PWMServoDriver.h: No such file or directory

CarRun:2:10: error: Adafruit_PWMServoDriver.h: No such file or directory
#include <Adafruit_PWMServoDriver.h>
^
compilation terminated.

exit status 1
Adafruit_PWMServoDriver.h: No such file or directory
```