Drive L298N motor modules to run the 4WD car

Table

1. Preface

Our final form of this product is a small car chassis with 4 motors and 4 wheels, without motor drive modules, control boards, batteries, and other things. Its highlight lies in its development and scalability. You can choose the motor drive and control board you want to use, install it on the chassis of this car, and make it run, becoming a four-wheel drive car. This will be a challenging and fulfilling task. Wishing you success!
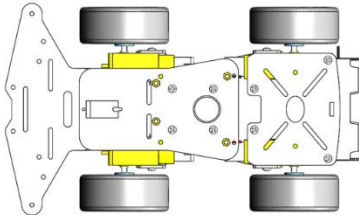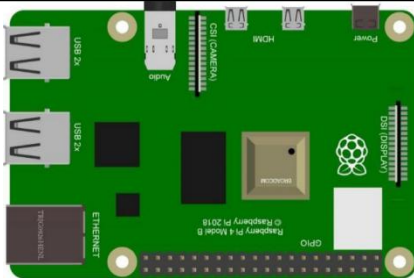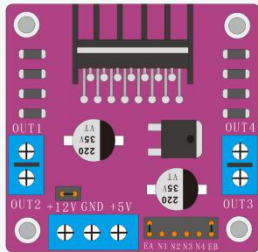
You can also refer to our Demo1, where we assembled 2pcs L298N motor modules, Raspberry Pi 4B, and 2pcs 18650 batteries onto the chassis of the car, creating a 4WD car based on Raspberry Pi. The following course will provide a detailed introduction to Demo1, including component list, component related knowledge, circuit connection, code, and more. If you are interested in demo1, you can refer to the demo1 checklist to prepare relevant materials for experimentation.

If you have any technical issues or suggestions, please provide feedback to us via email:
cokoino@outlook.com

2. Component List

For this Demo experiment, what do you need to prepare like below list

Component/Module	QTY	Picture	Remark
4WD Car Chassis	1		Provided by the 4WD Car Chassis Kit
Raspberry Pi 4B	1		You need to prepare these by yourself.These just as an example,you can DIY what is you want and prepared
L298N Motor driver module	2		
18650 battery	2		

Dupont line	Some		

3. Component related knowledge

3.1 Knowledge of the TT Motor

As shown in the figure below, the TT motor consists of a DC motor and related gears, with a yellow outer shell fastened.
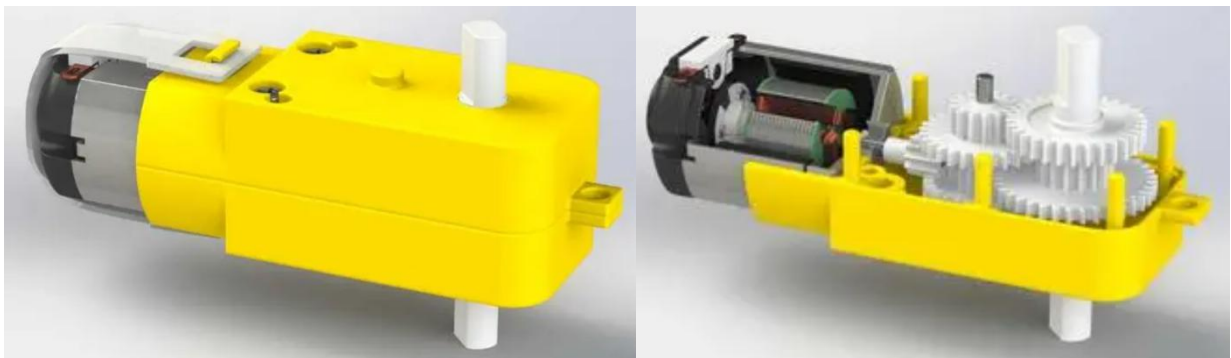
DC Motor

When motor is connected to the power supply, it will rotate in one direction. Reverse the polarity of power supply, the motor will rotate in the opposite direction.

And the speed of motor depends on the voltage between two ends. The larger the voltage, the lager the speed.

Support : cokoino@outlook.com
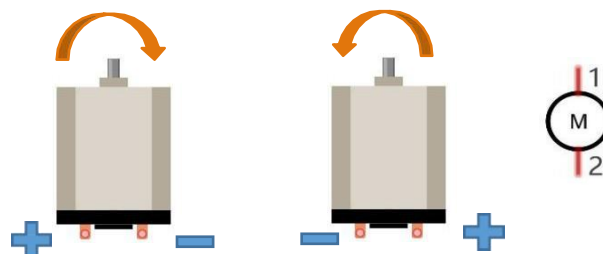
PWM

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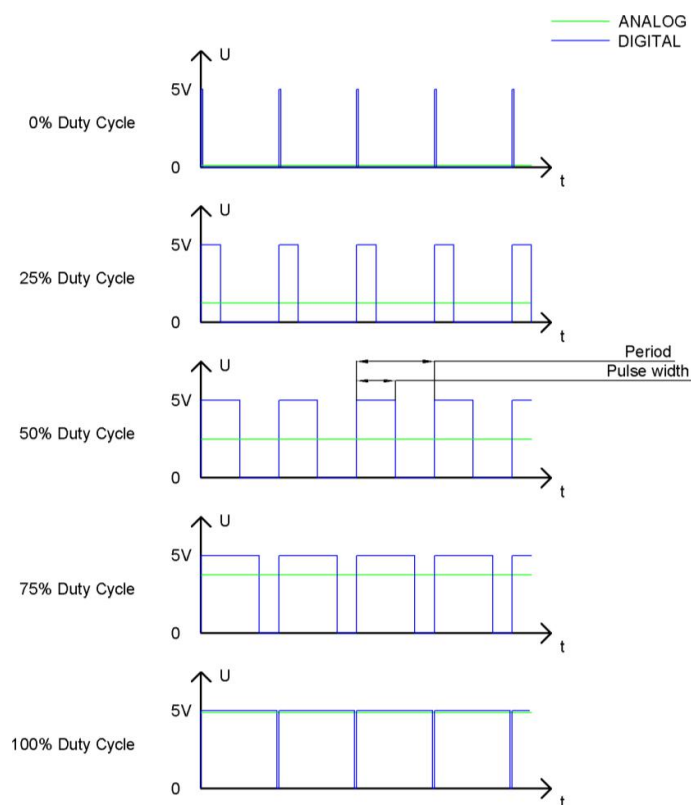
The longer the PWM duty cycle is, the higher the output power will be. Now that we understand this relationship, we can use PWM to control the brightness of an LED or the speed of DC motor and so on.

GND   GND

4
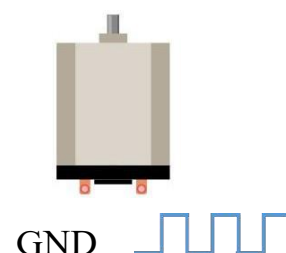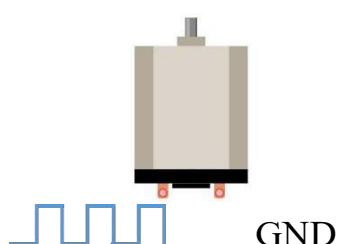
3.2 Knowledge of L298N Motor driver module

L298N is a specialized driver integrated circuit, belonging to H-bridge integrated circuits. Its output current is 2A, with a maximum current of 4A and a maximum working voltage of 50V. It can drive inductive loads such as high-power DC motors, stepper motors, solenoid valves, etc. Especially, its input end can be directly connected to the microcontroller, making it easy to be controlled by the microcontroller. When driving a DC motor, the stepper motor can be directly controlled and the forward and reverse rotation of the motor can be achieved. To achieve this function, only the logic level of the input terminal needs to be changed. In order to avoid interference from the motor on the microcontroller, an optocoupler is added to this module for photoelectric isolation, so that the system can work stably and reliably.
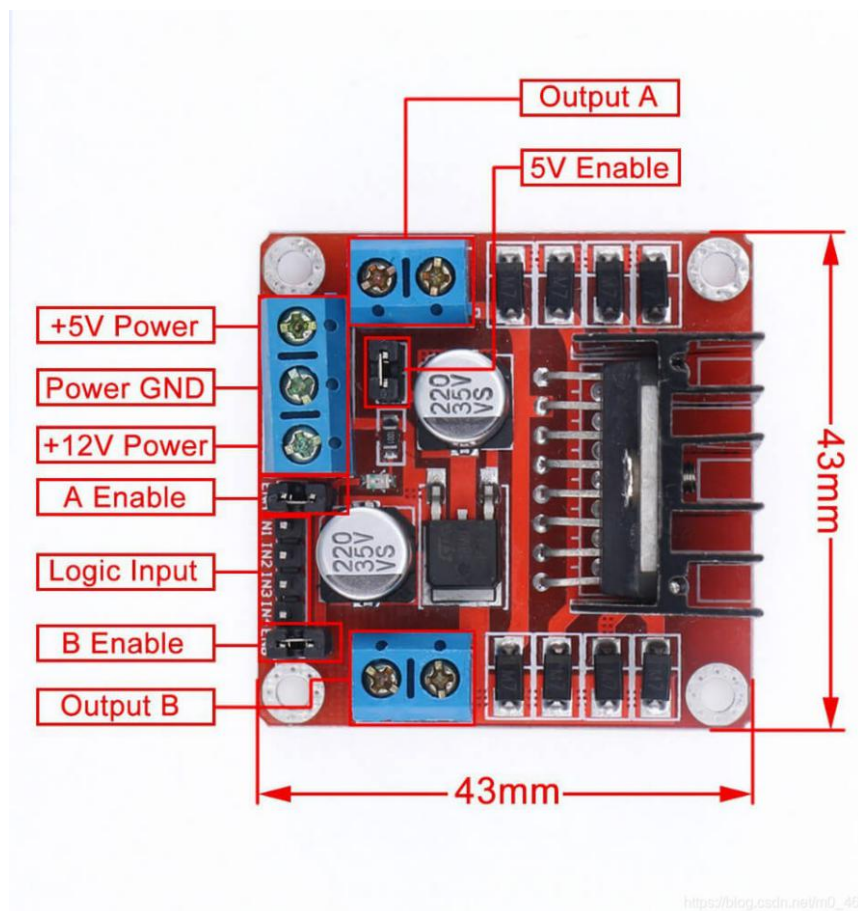
Output A: Connect A+and A - of DC motor 1 or stepper motor;

Output B: Connect B+and B - of DC motor 2 or stepper motor;

5V Enable: If using a power supply with an input voltage greater than 12V, please remove the jumper cap. When the input power is less than 12V, short circuiting can provide 5V power output;

+5V Power: When the input power is less than 12V and 5V Enable is in a short circuited state, it can provide+5V power output; (Please refer to the markings on the driver board for actual location)

GND: power ground;

+12V Power: Connect the motor power supply, maximum 35V. When the input voltage is greater than 12V, to ensure safety, please remove the jumper cap on the 5V Enable pin; (Please refer to the markings on the driver board for actual location)

A/B Enable: can be used to input PWM pulse width modulation signals for speed control of motors. If there is no need to adjust the speed, the two pins can be connected to 5V to make the motor work at the highest speed, which is achieved by short circuiting the short-circuit cap. It is easier to achieve forward and reverse rotation of the motor. The input signal terminal IN1 is connected to the high-level input terminal IN2 is connected to the low-level input terminal, and the motor M1 rotates forward. (If the signal terminal IN1 is connected to a low level, IN2 is connected to a high level, and motor M1 is reversed.) Control another motor in the same way, with the input signal terminal IN3 connected to a high level, input terminal IN4 connected to a low level, and motor M2 rotating forward. (On the contrary, it will be reversed), PWM signal terminal A controls M1 speed regulation, and PWM signal terminal B controls M2 speed regulation. You can refer to the following chart:

ENA	IN1	IN2	Description
0	x	x	Motor A is off.
1	0	0	Motor breaks and stops
1	0	1	Motor turns forward
1	1	0	Motor turns backward
1	1	1	Motor breaks and stops

Parameters:

1. Driver chip: L298N dual H-bridge DC motor driver chip

2. The power supply range of the driving part terminals is Vs:+5V -+35V;

If it is necessary to take power from the board, the power supply range is Vs:+7V -+35V

3. Peak current of the driving part Io: 2A

4. Logic section terminal power supply range Vss:+5V~+7V (can be powered on board+5V)

5. Operating current range of logic part: 0-36mA

6. Control signal input voltage range:

Low level: -0.3V ≤ Vin ≤ 1.5V

High level: 2.3V ≤ Vin ≤ Vss

7. Enable signal input voltage range:

Low level: -0.3 ≤ Vin ≤ 1.5V (invalid control signal)

High level: 2.3V ≤ Vin ≤ Vss (effective control signal)

8. Maximum power consumption: 20W

3.3 Knowledge of the Raspberry Pi

GPIO

GPIO: General Purpose Input/Output. Here we will introduce the specific function of the pins on the Raspberry Pi and how you can utilize them in all sorts of ways in your projects. Most RPi Module pins can be used as either an input or output, depending on your program and its functions.

When programming GPIO pins there are 3 different ways to reference them: GPIO Numbering, Physical Numbering and WiringPi GPIO Numbering.

BCM GPIO Numbering

The Raspberry Pi CPU uses Broadcom (BCM) processing chips BCM2835, BCM2836 or BCM2837. GPIO pin numbers are assigned by the processing chip manufacturer and are how the computer recognizes each pin. The pin numbers themselves do not make sense or have meaning as they are only a form of identification. Since their numeric values and physical locations have no specific order, there is no way to remember them so you will need to have a printed reference or a reference board that fits over the pins.

Each pin's functional assignment is defined in the image below:
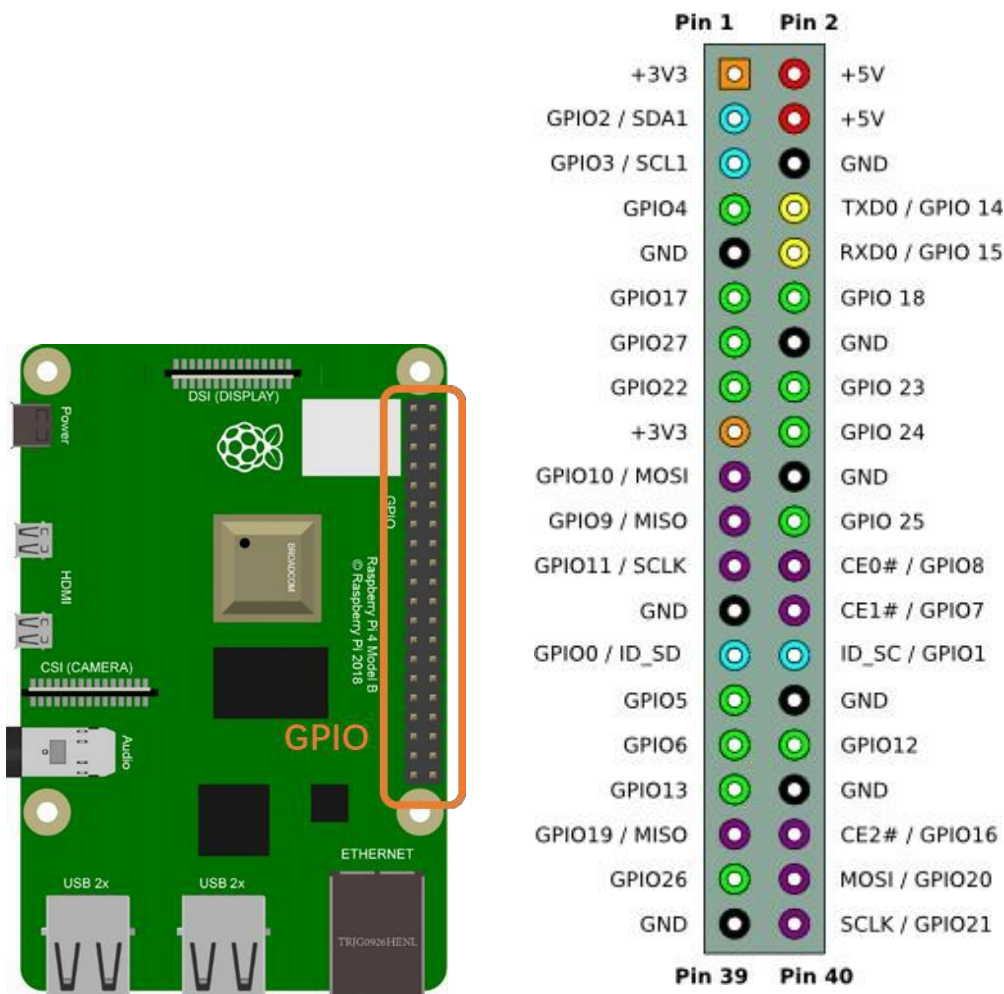
For more details about pin definition of GPIO, please refer to http://pinout.xyz/

PHYSICAL Numbering

Another way to refer to the pins is by simply counting across and down from pin 1 at the top left (nearest to the SD card). This is 'Physical Numbering', as shown below:

Raspberry Pi A+ / B+ and Raspberry Pi 2 physical pin numbers

- GPIO
- Ground
- 3.3v
- 5v
- ID EEPROM Advanced use only!

Support : cokoino@outlook.com
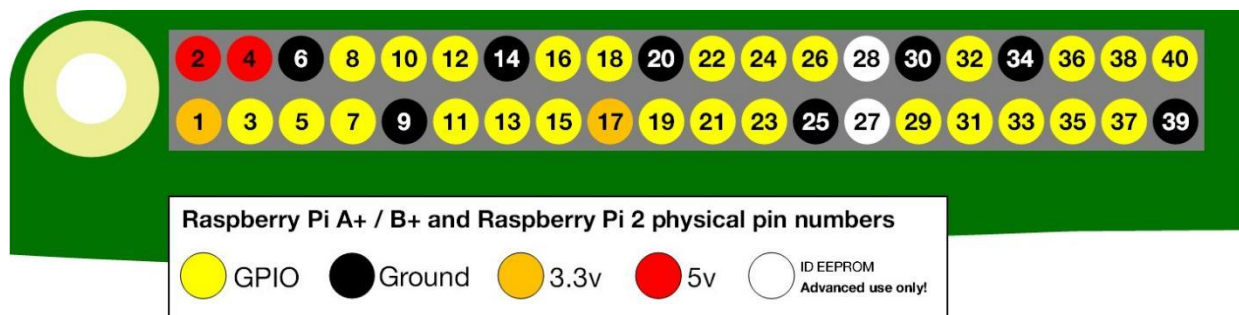
WiringPi GPIO Numbering

Different from the previous two types of GPIO serial numbers, RPi GPIO serial number of the WiringPi are numbered according to the BCM chip use in RPi.

wiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	wiringPi Pin		For Pi B	For A+, B+, 2B, 3B, 3B+, 4B, Zero
—	—	3.3v	1 | 2	5v	—	—			
8	R1:0/R2:2	SDA	3 | 4	5v	—	—			
9	R1:1/R2:3	SCL	5 | 6	0v	—	—			
7	4	GPIO7	7 | 8	TxD	14	15			
—	—	0v	9 | 10	RxD	15	16			
0	17	GPIO0	11 | 12	GPIO1	18	1			
2	R1:21/R2:27	GPIO2	13 | 14	0v	—	—			
3	22	GPIO3	15 | 16	GPIO4	23	4			
—	—	3.3v	17 | 18	GPIO5	24	5			
12	10	MOSI	19 | 20	0v	—	—			
13	9	MISO	21 | 22	GPIO6	25	6			
14	11	SCLK	23 | 24	CE0	8	10			
—	—	0v	25 | 26	CE1	7	11			
30	0	SDA.0	27 | 28	SCL.0	1	31			
21	5	GPIO.21	29 | 30	0V					
22	6	GPIO.22	31 | 32	GPIO.26	12	26			
23	13	GPIO.23	33 | 34	0V					
24	19	GPIO.24	35 | 36	GPIO.27	16	27			
25	26	GPIO.25	37 | 38	GPIO.28	20	28			
		0V	39 | 40	GPIO.29	21	29			
wiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	wiringPi Pin			

(For more details, please refer to https://projects.drogon.net/raspberry-pi/wiringpi/pins/)

Support : cokoino@outlook.com

You can also use the following command to view their correlation.

`gpio readall`

```
+-----+-----+---------+------+---+---Pi 4B--+---+------+---------+-----+-----+
| BCM | wPi |   Name  | Mode | V | Physical | V | Mode |  Name   | wPi | BCM |
+-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
|     |     |    3.3v |      |   |  1 || 2  |   |      | 5v      |     |     |
|   2 |   8 |   SDA.1 | ALT0 | 1 |  3 || 4  |   |      | 5v      |     |     |
|   3 |   9 |   SCL.1 | ALT0 | 1 |  5 || 6  |   |      | 0v      |     |     |
|   4 |   7 |  GPIO. 7 |  IN | 1 |  7 || 8  | 0 | IN   | TxD     | 15  | 14  |
|     |     |      0v |      |   |  9 || 10 | 1 | IN   | RxD     | 16  | 15  |
|  17 |   0 |  GPIO. 0 |  IN | 0 | 11 || 12 | 0 | IN   | GPIO. 1 |  1  | 18  |
|  27 |   2 |  GPIO. 2 |  IN | 0 | 13 || 14 |   |      | 0v      |     |     |
|  22 |   3 |  GPIO. 3 |  IN | 0 | 15 || 16 | 0 | IN   | GPIO. 4 |  4  | 23  |
|     |     |    3.3v |      |   | 17 || 18 | 0 | IN   | GPIO. 5 |  5  | 24  |
|  10 |  12 |    MOSI |  IN  | 0 | 19 || 20 |   |      | 0v      |     |     |
|   9 |  13 |    MISO |  IN  | 0 | 21 || 22 | 0 | IN   | GPIO. 6 |  6  | 25  |
|  11 |  14 |    SCLK |  IN  | 0 | 23 || 24 | 1 | IN   | CE0     | 10  |  8  |
|     |     |      0v |      |   | 25 || 26 | 1 | IN   | CE1     | 11  |  7  |
|   0 |  30 |   SDA.0 |  IN  | 1 | 27 || 28 | 1 | IN   | SCL.0   | 31  |  1  |
|   5 |  21 |  GPIO.21 |  IN | 1 | 29 || 30 |   |      | 0v      |     |     |
|   6 |  22 |  GPIO.22 |  IN | 1 | 31 || 32 | 0 | IN   | GPIO.26 | 26  | 12  |
|  13 |  23 |  GPIO.23 |  IN | 0 | 33 || 34 |   |      | 0v      |     |     |
|  19 |  24 |  GPIO.24 |  IN | 0 | 35 || 36 | 0 | IN   | GPIO.27 | 27  | 16  |
|  26 |  25 |  GPIO.25 |  IN | 0 | 37 || 38 | 0 | IN   | GPIO.28 | 28  | 20  |
|     |     |      0v |      |   | 39 || 40 | 0 | IN   | GPIO.29 | 29  | 21  |
+-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
| BCM | wPi |   Name  | Mode | V | Physical | V | Mode |  Name   | wPi | BCM |
+-----+-----+---------+------+---+---Pi 4B--+---+------+---------+-----+-----+
```

4. Circuit Connection

First, turn off power of your RPi board. Then build the circuit according to the circuit connection diagrams. After the circuit is built and verified correct, turn on the power of RPi board.

CAUTION: Avoid any possible short circuits (especially connecting 5V or GND, 3.3V and GND)!

WARNING: A short circuit can cause high current in your circuit, create excessive component heat and cause permanent damage to your RPi!

4.1 Circuit connection list

Pin of L298N module1	Corresponding connection pins
OUT1	Right back wheel motor's red wire
OUT2	Right back wheel motor's black wire
OUT3	Left back wheel motor's black wire
OUT4	Left back wheel motor's red wire
IN1	GPIO23 of Raspberry Pi
IN2	GPIO24 of Raspberry Pi
IN3	GPIO25 of Raspberry Pi
IN4	GPIO18 of Raspberry Pi

+12V	Battery box's red wire
GND	Battery box's black wire, GND of Raspberry Pi
+5V	N/A
ENA	N/A
ENB	N/A

Pin of L298N module2	Corresponding connection pins
OUT1	Right front wheel motor's black wire
OUT2	Right front wheel motor's red wire
OUT3	Left front wheel motor's red wire
OUT4	Left front wheel motor's black wire
IN1	GPIO4 of Raspberry Pi
IN2	GPIO17 of Raspberry Pi
IN3	GPIO27 of Raspberry Pi
IN4	GPIO22 of Raspberry Pi
+12V	Battery box's red wire
GND	Battery box's black wire, GND of Raspberry Pi
+5V	N/A
ENA	N/A
ENB	N/A

Support : cokoino@outlook.com

4.2 Circuit connection diagram:

CAUTION: Avoid any possible short circuits (especially connecting 5V or GND, 3.3V and GND)!

WARNING: A short circuit can cause high current in your circuit, create excessive component heat and cause permanent damage to your RPi!

Support : cokoino@outlook.com
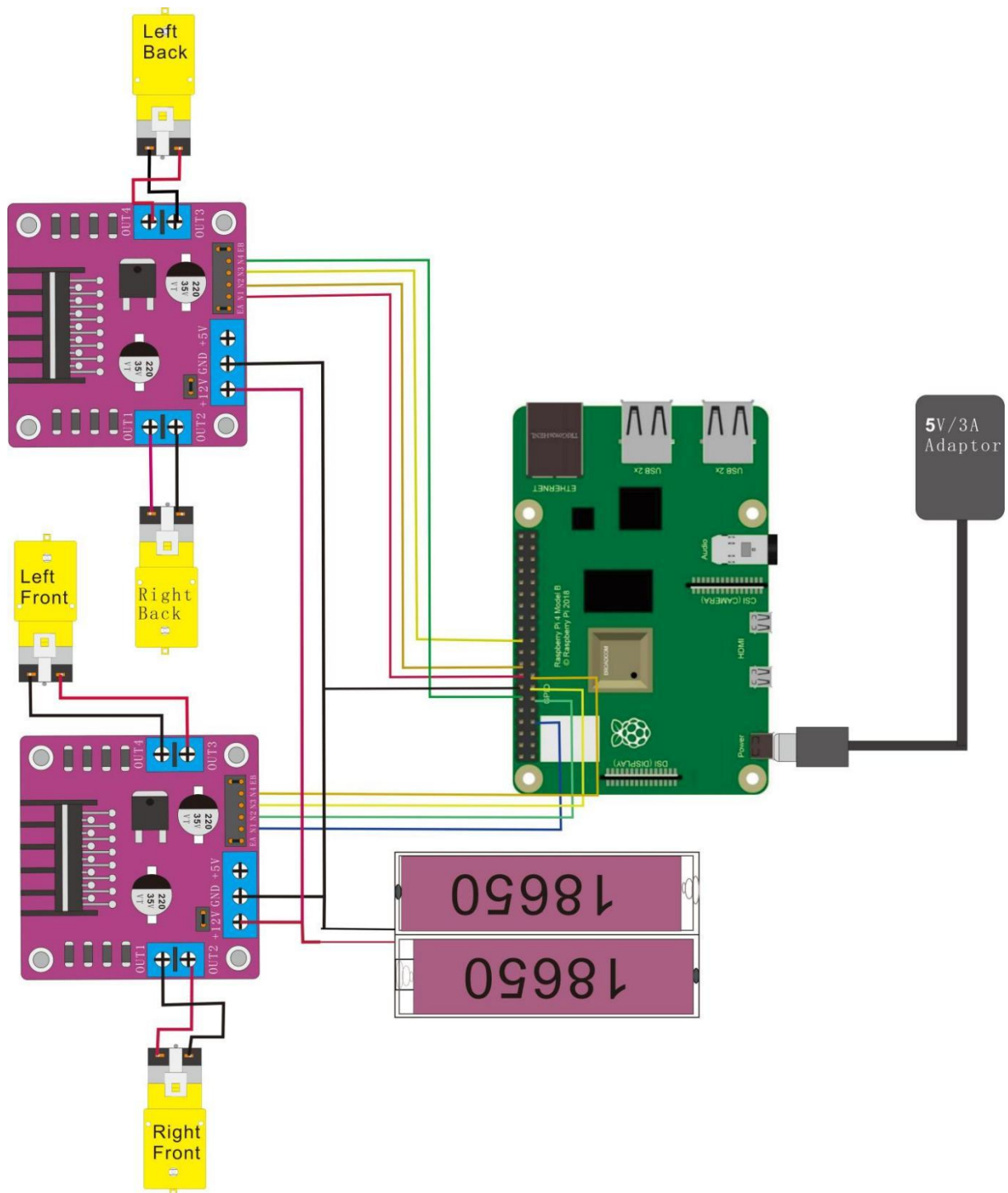
5. Download Code and Run

We recommend use Python code to achieve the function.

Please visit our GitHub resources at (**https://github.com/cokoino**) to download the latest available project code. We provide **Python** language code for this project .

This is the method for obtaining the code:

```
cd
git clone --depth 1 https://github.com/cokoino/CKK0011
```
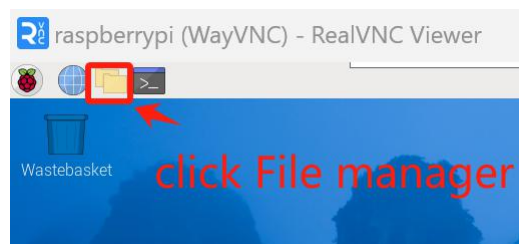
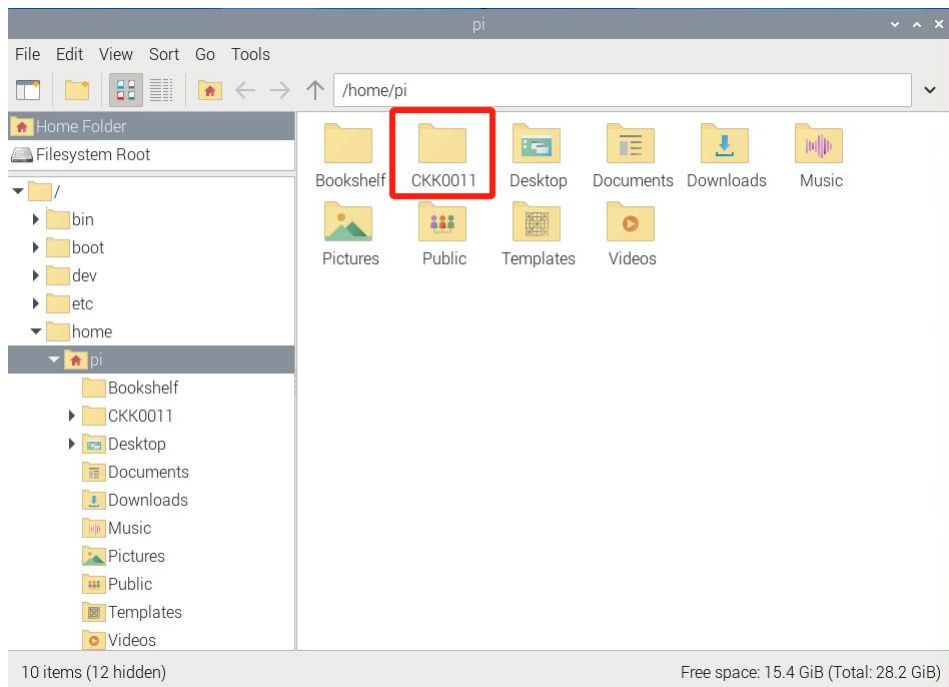In the pi directory of the RPi terminal, enter the following command.

(There is no need for a password. If you get some errors, please check your commands.)
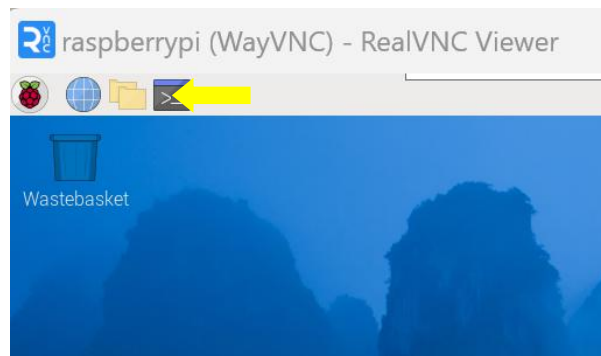
After the download is completed, a new folder "CKK0011" is generated, which contains all of the tutorials and required code.

Click File manager,you will find the folder "CKK0011"

Support : cokoino@outlook.com

If you have no experience with Python, we suggest that you refer to this website for basic information and knowledge.
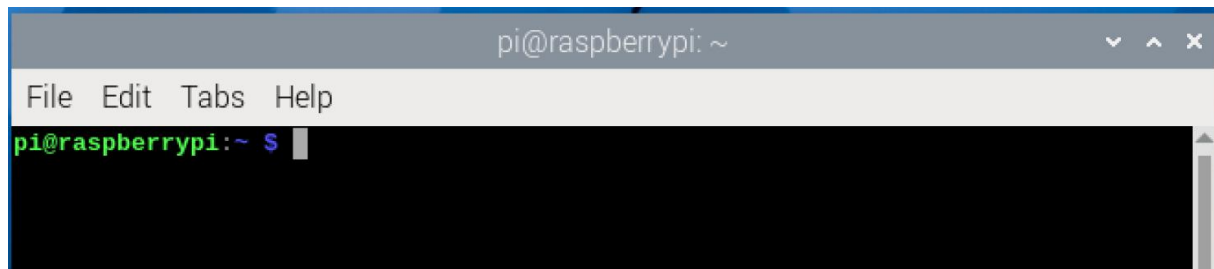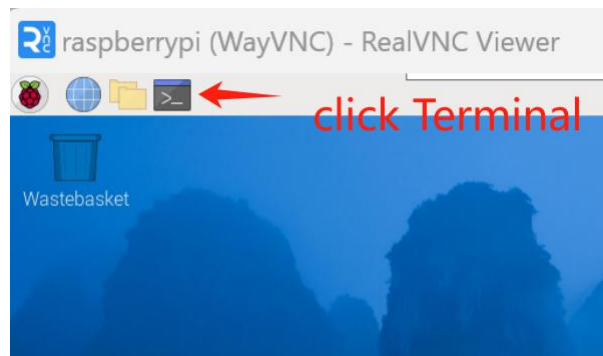
https://python.swaroopch.com/basics.html

Then,log in to Raspberry Pi by using VNC Viewer and operated proper setting.(For those who have no impression or are unclear about the operation, please refer to the PDF document "2 Installing and Configuring Raspberry Pi System").
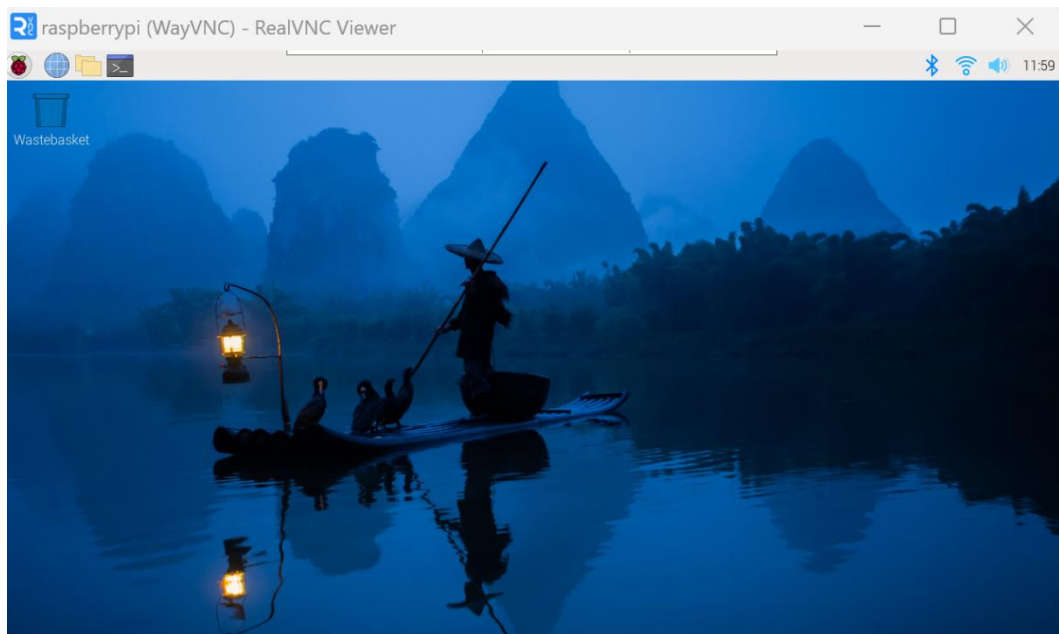
Click "Terminal", following interface appears.
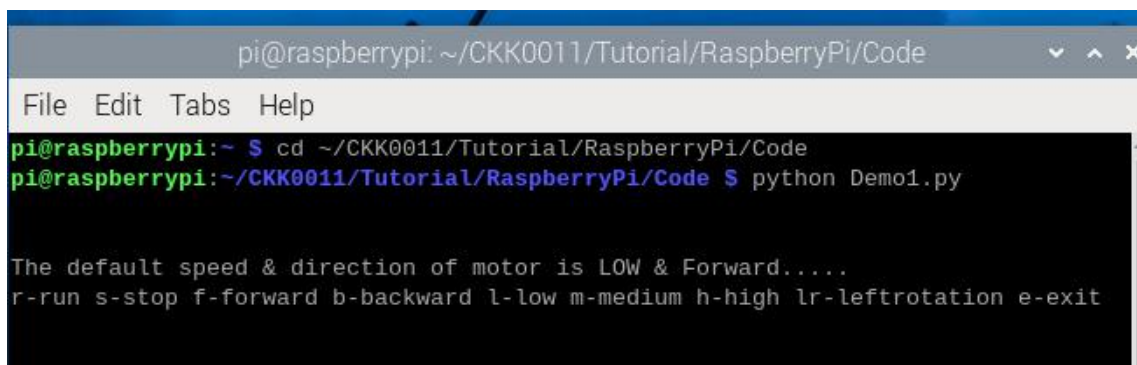
Support : cokoino@outlook.com

Use cd command to enter Demo1 directory of Python code.

```
cd  ~/CKK0011/Tutorial/RaspberryPi/Code
```

Use python command to execute python code Demo1.py.

```
python Demo1.py
```

Then the interface displays "The default direction of motor is Forward......r-run s-stop f-forward b-backward lr-leftrotation rr-rightrotation e-exit"

Finally, Enter the command according to the prompt information and press enter to observe whether the car will perform the corresponding action. For example, after entering "f" and entering the car, the car starts to move forward; After entering "s", the car stops moving.

At the same time, the remote desktop interface has corresponding information display.

Support : cokoino@outlook.com

You can enter the "e" command or press "Ctrl+C" to end the program.

The following is the program code:

```
1.   # Python Script
2.   # https://www.electronicshub.org/raspberry-pi-l298n-interface-tutorial-control-dc-motor-l298n-raspberry-pi/
3.
4.   import RPi.GPIO as GPIO
5.   from time import sleep
6.
7.   in1 = 23
8.   in2 = 24
9.   in3 = 25
10.  in4 = 18
11.  in5 = 4
12.  in6 = 17
13.  in7 = 27
14.  in8 = 22
15.  temp1=1
16.
17.  GPIO.setmode(GPIO.BCM)
18.  GPIO.setup(in1,GPIO.OUT)
19.  GPIO.setup(in2,GPIO.OUT)
20.  GPIO.setup(in3,GPIO.OUT)
21.  GPIO.setup(in4,GPIO.OUT)
22.  GPIO.setup(in5,GPIO.OUT)
23.  GPIO.setup(in6,GPIO.OUT)
24.  GPIO.setup(in7,GPIO.OUT)
25.  GPIO.setup(in8,GPIO.OUT)
26.  GPIO.output(in1,GPIO.LOW)
27.  GPIO.output(in2,GPIO.LOW)
28.  GPIO.output(in3,GPIO.LOW)
```

```python
29.     GPIO.output(in4,GPIO.LOW)
30.     GPIO.output(in5,GPIO.LOW)
31.     GPIO.output(in6,GPIO.LOW)
32.     GPIO.output(in7,GPIO.LOW)
33.     GPIO.output(in8,GPIO.LOW)
34.
35.
36.     print("\n")
37.     print("The default speed & direction of motor is LOW & Forward.....")
38.     print("r-run s-stop f-forward b-backward l-low m-medium h-high lr-leftrotation e-exit")
39.     print("\n")
40.
41.     while(1):
42.
43.        x=input()
44.
45.        if x=='r':
46.            print("run")
47.            if(temp1==1):
48.             GPIO.output(in1,GPIO.LOW)
49.             GPIO.output(in2,GPIO.HIGH)
50.             GPIO.output(in3,GPIO.LOW)
51.             GPIO.output(in4,GPIO.HIGH)
52.             GPIO.output(in5,GPIO.HIGH)
53.             GPIO.output(in6,GPIO.LOW)
54.             GPIO.output(in7,GPIO.HIGH)
55.             GPIO.output(in8,GPIO.LOW)
56.             print("forward")
57.             x='z'
58.            else:
59.             GPIO.output(in1,GPIO.HIGH)
60.             GPIO.output(in2,GPIO.LOW)
61.             GPIO.output(in3,GPIO.HIGH)
62.             GPIO.output(in4,GPIO.LOW)
63.             GPIO.output(in5,GPIO.LOW)
64.             GPIO.output(in6,GPIO.HIGH)
65.             GPIO.output(in7,GPIO.LOW)
66.             GPIO.output(in8,GPIO.HIGH)
67.             print("backward")
68.             x='z'
69.
70.
71.        elif x=='s':
72.            print("stop")
73.            GPIO.output(in1,GPIO.LOW)
74.            GPIO.output(in2,GPIO.LOW)
75.            GPIO.output(in3,GPIO.LOW)
76.            GPIO.output(in4,GPIO.LOW)
77.            GPIO.output(in5,GPIO.LOW)
78.            GPIO.output(in6,GPIO.LOW)
79.            GPIO.output(in7,GPIO.LOW)
80.            GPIO.output(in8,GPIO.LOW)
```

17

```python
81.         x='z'
82.
83.     elif x=='f':
84.         print("forward")
85.         GPIO.output(in1,GPIO.LOW)
86.         GPIO.output(in2,GPIO.HIGH)
87.         GPIO.output(in3,GPIO.LOW)
88.         GPIO.output(in4,GPIO.HIGH)
89.         GPIO.output(in5,GPIO.HIGH)
90.         GPIO.output(in6,GPIO.LOW)
91.         GPIO.output(in7,GPIO.HIGH)
92.         GPIO.output(in8,GPIO.LOW)
93.         temp1=1
94.         x='z'
95.
96.     elif x=='b':
97.         print("backward")
98.         GPIO.output(in1,GPIO.HIGH)
99.         GPIO.output(in2,GPIO.LOW)
100.        GPIO.output(in3,GPIO.HIGH)
101.        GPIO.output(in4,GPIO.LOW)
102.        GPIO.output(in5,GPIO.LOW)
103.        GPIO.output(in6,GPIO.HIGH)
104.        GPIO.output(in7,GPIO.LOW)
105.        GPIO.output(in8,GPIO.HIGH)
106.        temp1=0
107.        x='z'
108.
109.    elif x=='lr':
110.        print("leftrotation")
111.        GPIO.output(in1,GPIO.LOW)
112.        GPIO.output(in2,GPIO.HIGH)
113.        GPIO.output(in3,GPIO.HIGH)
114.        GPIO.output(in4,GPIO.LOW)
115.        GPIO.output(in5,GPIO.HIGH)
116.        GPIO.output(in6,GPIO.LOW)
117.        GPIO.output(in7,GPIO.LOW)
118.        GPIO.output(in8,GPIO.HIGH)
119.        temp1=0
120.        x='z'
121.    elif x=='e':
122.        GPIO.cleanup()
123.        print("GPIO Clean up")
124.        break
125.
126.    else:
127.        print("<<<  wrong data  >>>")
128.        print("please enter the defined data to continue.....")
```

About RPi.GPIO：

RPi.GPIO
This is a Python module to control the GPIO on a Raspberry Pi. It includes basic output function and input function of GPIO, and functions used to generate PWM.
GPIO.setmode(mode)
Sets the mode for pin serial number of GPIO. mode=GPIO.BOARD, which represents the GPIO pin serial number based on physical location of RPi. mode=GPIO.BCM, which represents the pin serial number based on CPU of BCM chip.
GPIO.setup(pin,mode)
Sets pin to input mode or output mode, "pin" for the GPIO pin, "mode" for INPUT or OUTPUT.
GPIO.output(pin,mode)
Sets pin to output mode, "pin" for the GPIO pin, "mode" for HIGH (high level) or LOW (low level).

For more functions related to RPi.GPIO, please refer to:

https://sourceforge.net/p/raspberry-gpio-python/wiki/Examples/

"import time" time is a module of python.

https://docs.python.org/2/library/time.html?highlight=time%20time#module-time

In subfunction setup(), GPIO.setmode (GPIO.BOARD) is used to set the serial number for GPIO based on physical location of the pin.

GPIO Numbering Relationship

WingPi	BCM(Extension)	Physical		BCM(Extension)	WingPi
3.3V	3.3V	1	2	5V	5V
8	GPIO2/SDA1	3	4	5V	5V
9	GPIO3/SCL1	5	6	GND	GND
7	GPIO4	7	8	GPIO14/TXD0	15
GND	GND	9	10	GPIO15/RXD0	16
0	GPIO17	11	12	GPIO18	1
2	GPIO27	13	14	GND	GND
3	GPIO22	15	16	GPIO23	4
3.3V	3.3V	17	18	GPIO24	5
12	GPIO10/MOSI)	19	20	GND	GND
13	GPIO9/MOIS	21	22	GPIO25	6
14	GPIO11/SCLK	23	24	GPIO8/CE0	10
GND	GND	25	26	GPIO7/CE1	11
30	GPIO0/SDA0	27	28	GPIO1/SCL0	31
21	GPIO5	29	30	GND	GND
22	GPIO6	31	32	GPIO12	26
23	GPIO13	33	34	GND	GND
24	GPIO19	35	36	GPIO16	27
25	GPIO26	37	38	GPIO20	28
GND	GND	39	40	GPIO21	29

6. Trouble shooting

Question 1: If the car does not move according to the code instructions, such as the command being forward, but the actual action of the car is backward or other actions.

Answer: Please check if the motors on the four wheels of the car are connected to the corresponding positions of the cokoino Pi Power&4WD HAT. If not, the car will not perform the corresponding actions according to the instructions.

Question2: Using RealVNC Viewer as

a remote desktop connection does not connect to Raspberry Pi, and Raspberry Pi does not display when connected to the monitor.

Answer: Turn off the power, please check if the connection circuit is correct, ensure that there is no short circuit between 5V and GND, and there is no short circuit between 3V and GND; Check if the output of the Raspberry Pi adapter is normal.

7. Any questions and suggestions are welcome

THANK YOU for participating in this Demo1 experience!

If you find any errors, omissions or you have suggestions and/or questions about this lesson, please feel free to contact us:

cokoino@outlook.com

We will make every effort to make changes and correct errors as soon as feasibly possible and publish a revised version.

If you want to learn more about Arduino, Raspberry Pi, Smart Cars, Robotics and other interesting products in science and technology, please continue to visit our Amazon Store by search for "**LK COKOINO**" on Amazon. We will continue to launch fun, cost-effective, innovative and exciting products.

Thank you again for choosing Cokoino products.

LK COKOINO