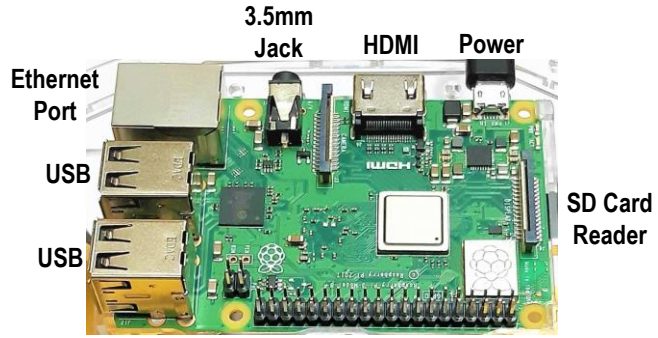
GPIO Reference Sheet

YoungWonks

Raspberry Pi



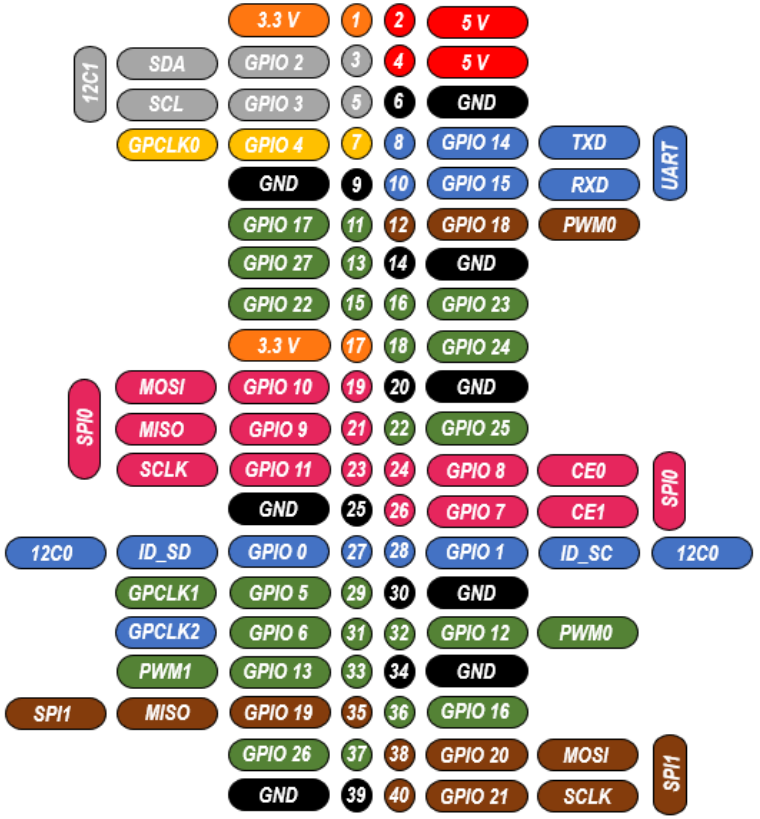
Minimum Code

```
# import GPIO module
import RPi.GPIO as GPIO
# use Broadcom mode
GPIO.setmode(GPIO.BCM)
# setup output pin
GPIO.setup(pin_number, GPIO.OUT)
# setup input pin
GPIO.setup(pin_number, GPIO.IN)
# set value of pin HIGH
GPIO.output(pin_number, GPIO.HIGH)
# set value of pin LOW
GPIO.output(pin_number, GPIO.LOW)
# clear all value after program
GPIO.cleanup()
```

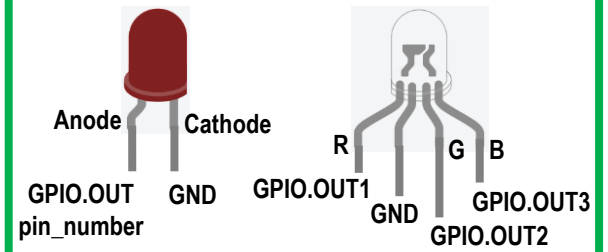
$$frequency\ (Hz) = \frac{1}{time\ (s)}$$

$$duty\ cycle\ (\%) = \frac{ON\ time}{ON\ time\ \times\ OFF\ time} \times 100$$

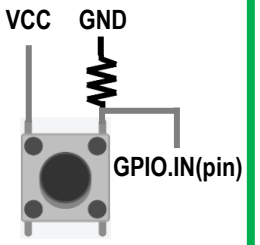
PIN Configuration



LED (light emitting diode)



Button



Resistor

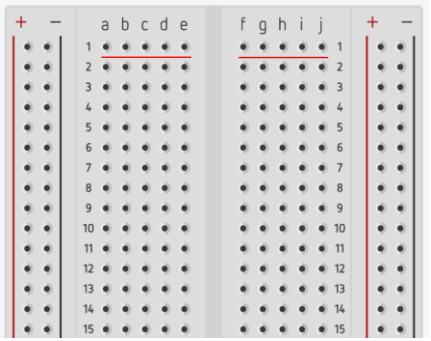
10x100 = 1KΩ ± 5%

Color	1st Band	2nd Band	Multiplier	Tolerance
Black	0	0	1	
Brown	1	1	10	
Red	2	2	100	
Orange	3	3	1K	
Yellow	4	4	10K	
Green	5	5	100K	
Blue	6	6	1M	
Violet	7	7	10M	
Gray	8	8	100M	
White	9	9	1000M	
Gold	-	-	0.1	+/- 5%
Silver	-	-	0.01	+/- 10%

PWM (Pulse Width Modulation)

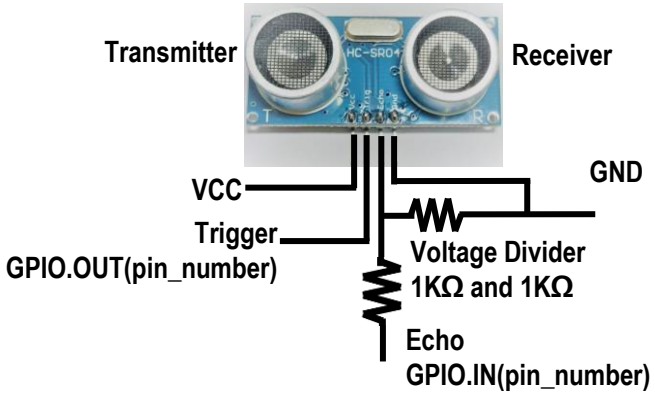
```
# create PWM function
p = GPIO.PWM(pin_number, frequency)
# start PWM with 50% duty cycle
p.start(50)
# change duty cycle
p.ChangeDutyCycle(new_dutycycle)
# change frequency
p.ChangeFrequency(new_frequency)
# stop PWM function
p.stop()
```

Bread Board



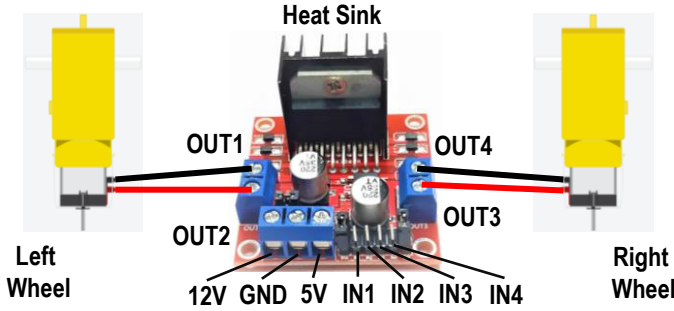
(+) rail is usually used for VCC and (–) rail for GND. Rails in the middle are connected according to rows.
For example:
First row '1' connects 'a' to 'j'' with a break between 'e' and 'f'.

Ultrasonic Sensor



When the output pin is triggered, the sensor transmits a sound pulse. The receiver senses the reflection of the sound pulse and calculates the distance of the obstacle.
Speed of sound = 343 m/s (~ 340m/s)
Distance = $\frac{speed \times time}{2}$
Voltage Divider helps reduce 5V to 3.3V to prevent damage to Raspberry Pi.

L298N Motor Driver

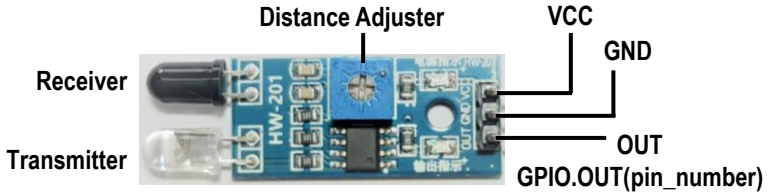


IN1,IN2,IN3,IN4 are GPIO.OUT(pin_number)

Pin Name	Description
IN1 & IN2	Motor A input pins. Control the spinning direction
IN3 & IN4	Motor B input pins. Control the spinning direction
OUT1 & OUT2	Output pins of Motor A
OUT3 & OUT4	Output pins of Motor B
12V	12V input from DC power Source
5V	Supplies power for the switching logic circuitry inside L298N IC
GND	Ground pin

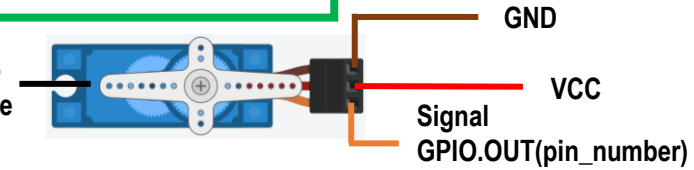
Spin Direction	Pin Numbers				Spin Direction
	IN1 OUT1	IN2 OUT2	IN3 OUT3	IN4 OUT4	
STOP	LOW	LOW	LOW	LOW	STOP
FORWARD	LOW	HIGH	HIGH	LOW	FORWARD
BACKWARD	HIGH	LOW	LOW	HIGH	BACKWARD
STOP	HIGH	HIGH	HIGH	HIGH	STOP

IR Sensor



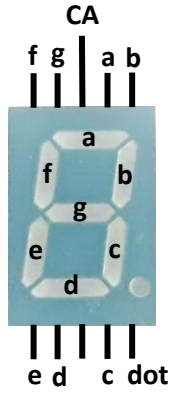
The transmitter transmits light pulse and receiver detects it. There is an obstacle in front if no light is detected.

Servo Motor



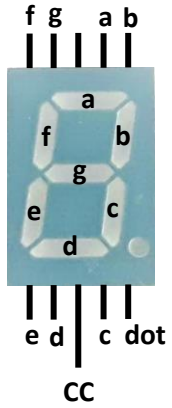
For instance if the pulse width is 20ms then, duty cycle of 0.5ms i.e., 2.5% will set the motor to 0°, 1.5ms i.e., 7.5% will set the motor to 90° and 2.5ms i.e., 12.5% will set the motor to 180°.

7-segment Display

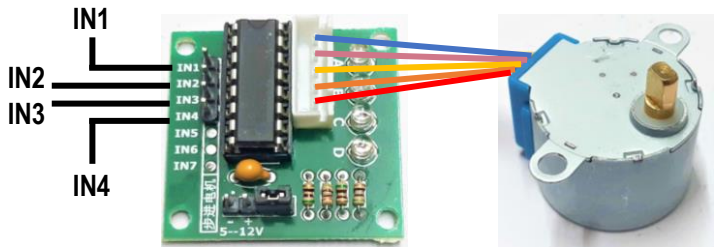


Common Cathode : All the negative terminals are connected together and lights are controlled by turning the positive terminals high or low. '1' turns the segment on.

Common Anode : All the negative terminals are connected together and lights are controlled by turning the positive terminals high or low.'0' turns the segment on.



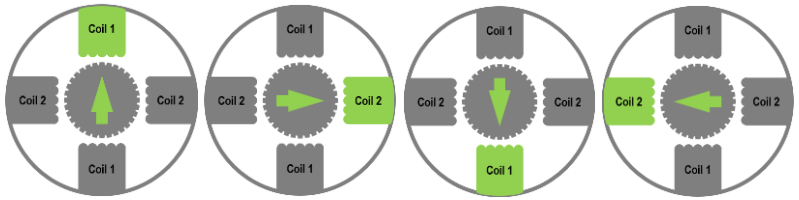
Stepper Motor



ULN2003 Stepper Motor Driver

Stepper Motor

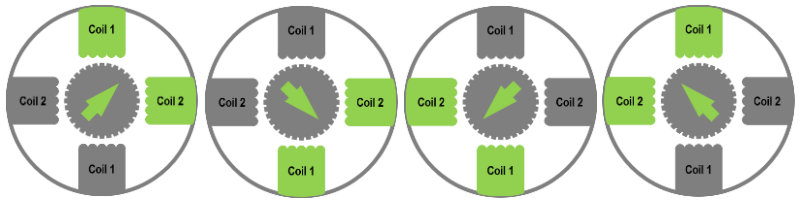
IN1, IN2, IN3 and IN4 are connected to GPIO pins and these pins are set as input pins. These pins represent Step1, Step2, Step3 and Step4 respectively. Setting these pins HIGH and LOW as per the table given on right makes the motor turn.



Step 1 Step 2 Step 3 Step 4

Single Phase Stepping				
	Step1	Step2	Step3	Step4
Coil1	1	0	0	0
Coil2	0	1	0	0
Coil1	0	0	1	0
Coil2	0	0	0	1

It takes four steps to complete one cycle in Single Phase Stepping.



Step 1 Step 2 Step 3 Step 4

Dual Phase Stepping				
	Step1	Step2	Step3	Step4
Coil1	1	0	0	1
Coil2	1	1	0	0
Coil1	0	1	1	0
Coil2	0	0	1	1

It takes four steps to complete one cycle in Dual Phase Stepping and eight steps in Half Stepping.

Half Stepping				
	Coil1	Coil2	Coil1	Coil2
Step1	1	0	0	0
Step2	1	1	0	0
Step3	0	1	0	0
Step4	0	1	1	0
Step5	0	0	1	0
Step6	0	0	1	1
Step7	0	0	0	1
Step8	1	0	0	1

Key input for Robot

```
'''the getch() method can determine which key has been pressed by the user on the keyboard by accessing the system files, it will then return the pressed key as a variable'''

def getch():
    import sys, tty, termios
    fd = sys.stdin.fileno()
    old_settings = termios.tcgetattr(fd)
        try:
            tty.setraw(sys.stdin.fileno())
            ch = sys.stdin.read(1)
        finally:
            termios.tcsetattr(fd,
                              termios.TCSADRAIN,
                              old_settings)
    return ch
```

Connect Robot using ssh

```
# syntax
ssh username@ip address

# example
ssh pi@10.1.10.21

# choose 'yes' for the next question
# now you can launch your code by running
sudo python3 mycode.py
```