11.Ultrasonic_Sensor

Introduction

In this lesson,we use ultrasonic ranging module to measure distance, and print out the data in the terminal.

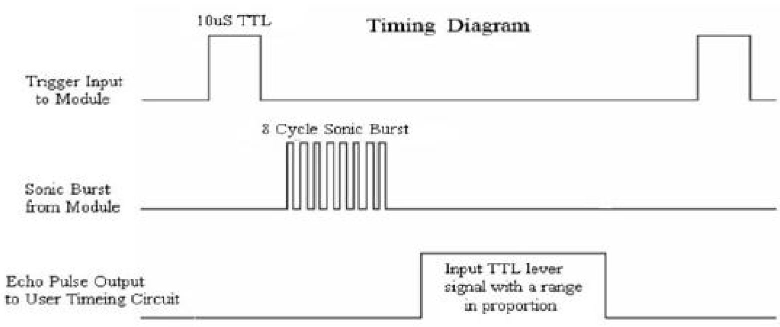
Hardware Required

- ✓ 1 * Raspberry Pi
- ✓ 1 * T-Extension Board
- ✓ 1 * 40-pin Cable
- ✓ Several Jumper Wires
- ✓ 1 * Breadboard
- ✓ 1 * HC-SR04 Ultrasonic Sensor Module

Principle

HC-SR04 Ultrasonic Sensor Module

Ultrasonic Sensor emits an ultrasound at 40 000 Hz which travels through the air and if there is an object or obstacle on its path It will bounce back to the module. Considering the travel time and the speed of the sound you can calculate the distance. In order to generate the ultrasound you need to set the Trig on a High State for 10us. That will send out an 8 cycle sonic burst which will travel at the speed sound and it will be received in the Echo pin. The Echo pin will output the time in microseconds the sound wave traveled.



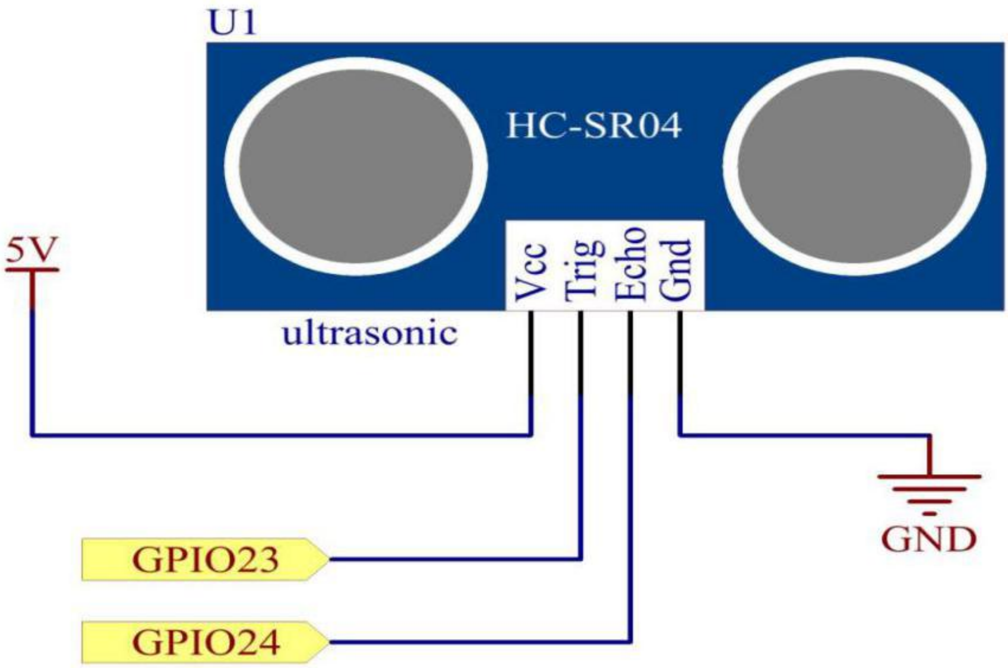
Test distance = (high level time × velocity of sound (340m/s) /2

For example, if the object is 10 cm away from the sensor, and the speed of the sound is 340 m/s or 0.034 cm/µs the sound wave will need to travel about 294 u seconds.

But what you will get from the Echo pin will be double that number because the sound wave needs to travel forward and bounce backward. So in order to get the distance in cm we need to multiply the received travel time value from the echo pin by 0.034 and divide it by 2.

T-Board Name	physical	wiringPi	BCM
GPIO23	Pin 16	4	23
GPIO24	Pin 18	5	24

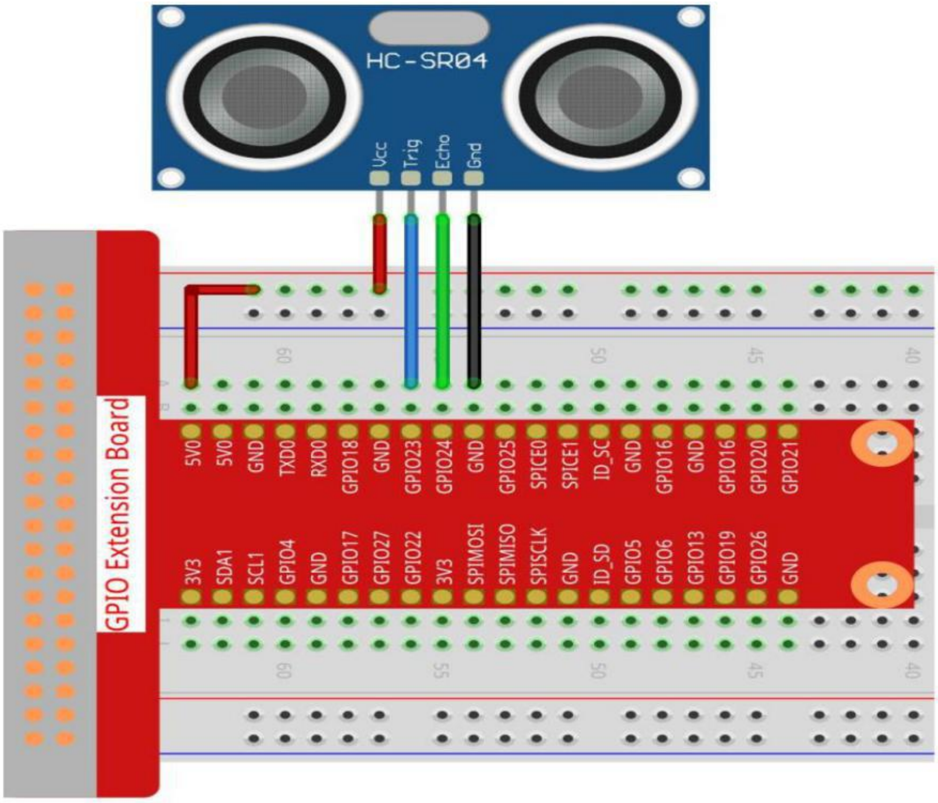
Schematic Diagram



Experimental Procedures

Step 1: Build the circuit.

11.Ultrasonic_Sensor



For C Language Users

Step 2: Open the code file.

```
cd /home/pi/REXQualis_Raspberry_Pi_Complete_Starter_Kit/C/11.Ultrasonic_Sensor
```

Step 3: Compile the code.

```
gcc 11.Ultrasonic_Sensor.c -o Ultrasonic_Sensor.out -lwiringPi
```

Step 4: Run the executable file above.

```
sudo ./Ultrasonic_Sensor.out
```

With the code run, the ultrasonic sensor module detects the distance between the obstacle ahead and the module itself, then the distance value will be printed on the screen.

Code

```
#include <wiringPi.h>

#include <stdio.h>

#include <sys/time.h>    //provide timer
```

```
#define Trig      4
#define Echo      5
void ultraInit(void)
{
    pinMode(Echo, INPUT);     //define the 4 ---input
    pinMode(Trig, OUTPUT);     //define the 5 ----output
}

float disMeasure(void)
{
    struct timeval tv1;
    struct timeval tv2;
    long time1, time2;
   float dis;

    digitalWrite(Trig, LOW);
    delayMicroseconds(2);     //2ms to pwm

    digitalWrite(Trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(Trig, LOW);// _'_ to triger to get out

    while(!(digitalRead(Echo) == 1));
    gettimeofday(&tv1, NULL);    //transform to time

    while(!(digitalRead(Echo) == 0));
    gettimeofday(&tv2, NULL);      //transform to time
```

```
    time1 = tv1.tv_sec * 1000000 + tv1.tv_usec;     //get the microsecond
    time2 = tv2.tv_sec * 1000000 + tv2.tv_usec;     //get the microsecond

    dis = (float)(time2 - time1) / 1000000 * 34000 / 2;     //the voice speed * timer

    return dis;
}

int main(void)
{
    float dis;
    if(wiringPiSetup() == -1){ //when initialize wiring failed,print message to screen
        printf("setup wiringPi failed !");
        return 1;
    }

    ultraInit();

    while(1){
        dis = disMeasure();
        printf("%0.2f cm\n\n",dis);
        delay(300);
    }

    return 0;
}
```

Code Explanation

```
void ultraInit(void)
{
```

```
    pinMode(Echo, INPUT);      //define the 4 ---input
    pinMode(Trig, OUTPUT);      //define the 5 ----output
}
```

Initialize the ultrasonic pin; meanwhile, set Echo to input, Trig to output.

```
float disMeasure(void){}
```

This function is used to realize the function of ultrasonic sensor by calculating the return detection distance.

```
    struct timeval tv1;
    struct timeval tv2;
```

Struct timeval is a structure used to store the current time. The complete structure is as follows:

```
struct timeval
{
__time_t tv_sec; /* Seconds. */
__suseconds_t tv_usec; /* Microseconds. */
};
```

Here, tv_sec represents the seconds that Epoch spent when creating struct timeval. Tv_usec stands for microseconds or a fraction of seconds.

```
    digitalWrite(Trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(Trig, LOW);
```

A 10us ultrasonic pulse is being sent out.

```
    while(!(digitalRead(Echo) == 1));
    gettimeofday(&tv1, NULL);
```

This empty loop is used to ensure that when the trigger signal is sent, there is no interfering echo signal and then get the current time.

```
    while(!(digitalRead(Echo) == 0));
    gettimeofday(&tv2, NULL);
```

This empty loop is used to ensure that the next step is not performed until the echo signal is received and then get the current time.

```
time1 = tv1.tv_sec * 1000000 + tv1.tv_usec;
time2 = tv2.tv_sec * 1000000 + tv2.tv_usec;
```

Convert the time stored by struct timeval into a full microsecond time.

```
dis = (float)(time2 - time1) / 1000000 * 34000 / 2;
```

The distance is calculated by the time interval and the speed of sound propagation. The speed of sound in the air: 34000cm/s.

For Python Language Users

Step 2: Open the code file.

```
cd /home/pi/REXQualis_Raspberry_Pi_Complete_Starter_Kit/Python
```

Step 3: Run.

```
sudo python3 11.Ultrasonic_Sensor.py
```

After the program is executed, make the detector of ultrasonic ranging module aim at the plane of an object, then the distance between the ultrasonic module and the object will be displayed in the terminal. As is shown below:

Code

The code here is for Python3, if you need for Python2, please open the code with the suffix py2 in the attachment.

```
#!/usr/bin/env python3

import RPi.GPIO as GPIO
import time

TRIG = 16    #define the 16 ---input
ECHO = 18    #define the 18 ---input

#notion BROAD
```

11.Ultrasonic_Sensor

```
def setup():
    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(TRIG, GPIO.OUT)
    GPIO.setup(ECHO, GPIO.IN)

def distance():
    GPIO.output(TRIG, 0)
    time.sleep(0.000002)    #0.000002 s low

    GPIO.output(TRIG, 1)
    time.sleep(0.00001)    #0.000001 s high
    GPIO.output(TRIG, 0)

    while GPIO.input(ECHO) == 0:        #get the triger_begin
        a = 0
    time1 = time.time()
    while GPIO.input(ECHO) == 1:        #get the triger_end
        a = 1
    time2 = time.time()     #voice begin-end time

    during = time2 - time1
    return during * 340 / 2 * 100    #the speed of voice 314m/s

def loop():
    while True:
        dis = distance()
        print ('Distance: %.2f' % dis)
        time.sleep(0.3)
```

```
def destroy():
    GPIO.cleanup()

if __name__ == "__main__":
    setup()
    try:
        loop()
    except KeyboardInterrupt:
        destroy()
```

Code Explanation

```
def setup():
```

This function is used to realize the function of ultrasonic sensor by calculating the return detection distance.

```
    GPIO.output(TRIG, 1)
    time.sleep(0.00001)
    GPIO.output(TRIG, 0)
```

This is sending out a 10us ultrasonic pulse.

```
    while GPIO.input(ECHO) == 0:
        a = 0
    time1 = time.time()
```

This empty loop is used to ensure that when the trigger signal is sent, there is no interfering echo signal and then get the current time.

```
while GPIO.input(ECHO) == 1:
        a = 1
    time2 = time.time()
```

This empty loop is used to ensure that the next step is not performed until the echo signal is received and then get the current time.

```
    during = time2 - time1
```

Execute the interval calculation.

```
return during * 340 / 2 * 100
```

The distance is calculated in the light of time interval and the speed of sound propagation. The speed of sound in the air: 340m/s.

Phenomenon Picture