

# ME007YS Waterproof Ultrasonic Sensor

SKU:SEN0312

---



## Introduction

---

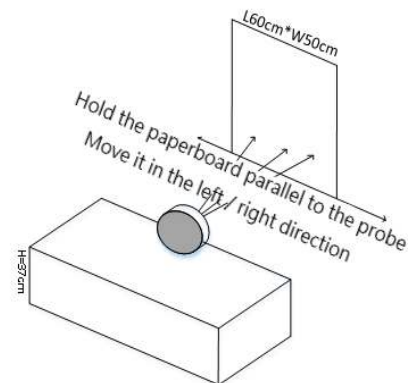
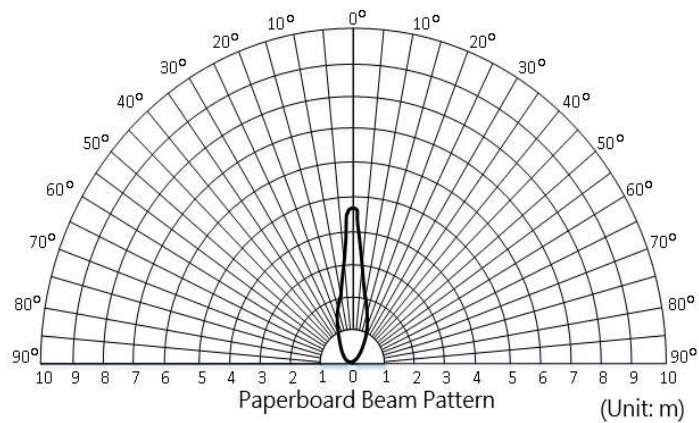
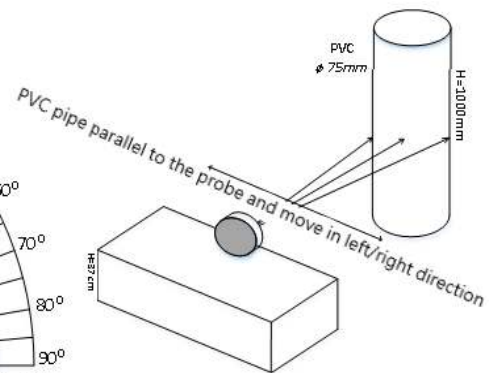
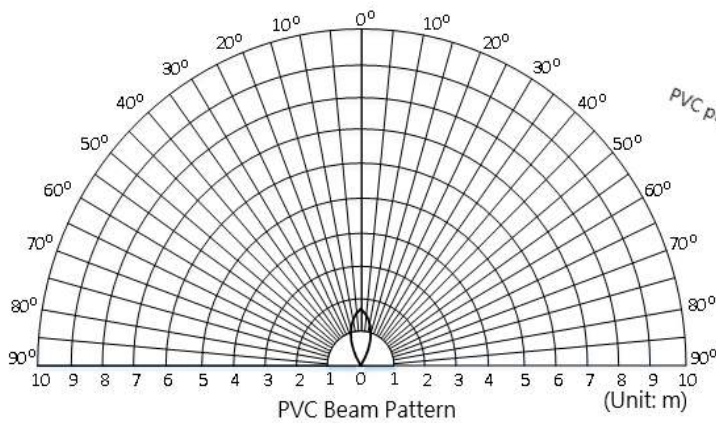
Ultrasonic distance sensor determines the distance to a target by measuring time lapses between the sending and receiving of the ultrasonic pulse. ME007YS is an waterproof ultrasonic sensor module with 4.5m effective ranging distance. It is compatible with 5V device like Arduino, Raspberry Pi, etc. The average current of ME007YS is only 8mA so it can be powered by most controllers' IO port.

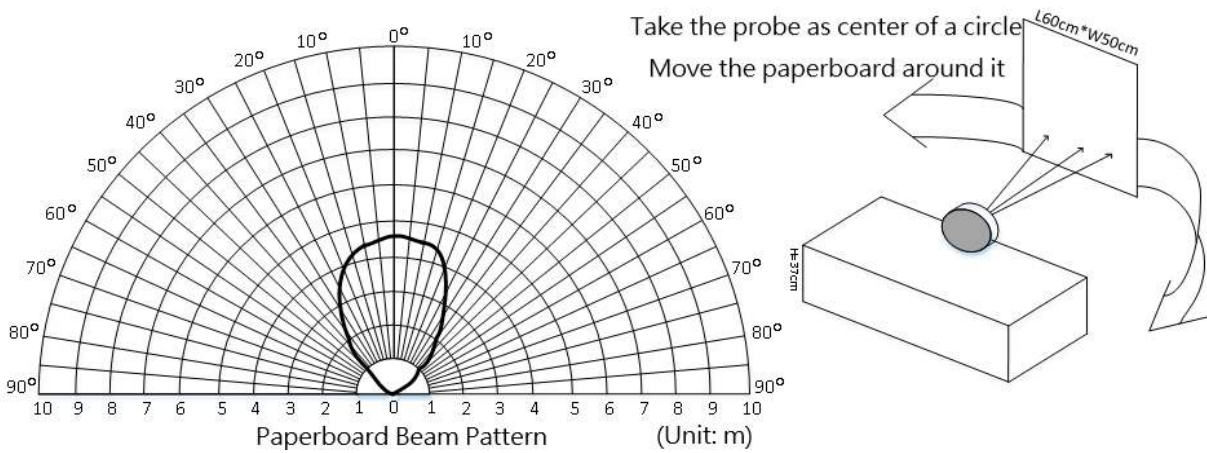
The ultrasonic sensor adopts closed probe of transmitter & receiver, waterproof and dustproof, which could be well suitable for harsh and moist measuring environment. It reserves 2.54-4P interface and adopts UART communication. ME007YS has experienced long-term test and constant optimization so it can offer a pretty fast response time, high stability and sensitivity, and low power consumption.

Use the sensor with Arduino controller to build up your projects, such as backing car annunciator, obstacle avoidance robot, object approaching detection etc.

## Specification

- Operating Voltage: 5V
- Average Current:  $\leq 8\text{mA}$
- Peak Current:  $\leq 50\text{mA}$
- Blind Zone Distance: 0-28cm
- Ranging Distance for Flat Object: 28-450cm
- Output: UART
- Response Time: 100ms
- Probe Center Frequency:  $40\text{K} \pm 1.0\text{K}$
- Operating Temperature:  $-15 \sim 60^\circ\text{C}$
- Storage Temperature:  $-25 \sim 80^\circ\text{C}$
- Operating Humidity:  $\leq 80\%$
- Operating Humidity:  $\leq 90\%$
- Sensing Angle:  $44^\circ$
- Protection Rate: IP67

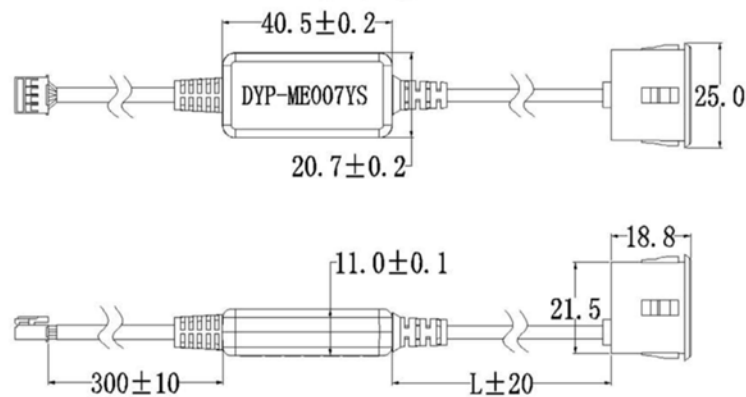




## Features

- High Protection Rate
- Strong Resistance
- Stable Output
- Low Power
- Fast Response
- High Antistatic Performance
- Wide Operating Temperature
- High Accuracy
- Small in Size

## Installation Dimension

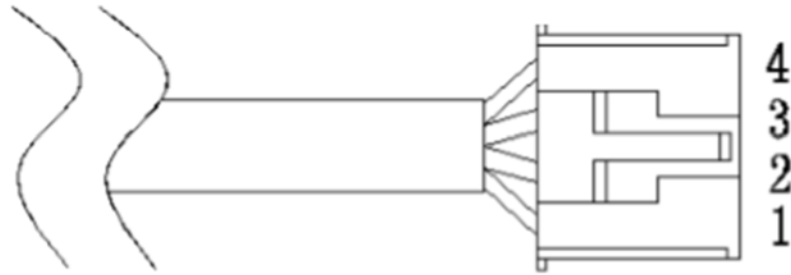


Proportion: 1:1  
Unit: mm

Note: the default length L is 1000mm

## Pinout

---



| Label | Name | Description  |
|-------|------|--------------|
| 1     | VCC  | Power Input  |
| 2     | GND  | Ground       |
| 3     | RX   | Function Pin |
| 4     | TX   | UART Output  |

## UART Output

---

### Output Communication

When "RX" floats or input High level, the module outputs processed value, the data is more steady, response time: 100-300ms; when input Low level, the module outputs real-time value, response time: 100ms.

| UART      | Data bit | Stop bit | Parity | Band rate |
|-----------|----------|----------|--------|-----------|
| TTL level | 8        | 1        | none   | 9600bps   |

### UART Output Form

| Frame Data | Description               | Byte   |
|------------|---------------------------|--------|
| Header     | 0xFF                      | 1 byte |
| DATA_H     | Distance Data High 8-bits | 1 byte |
| DATA_L     | Distance Data Low 8-bits  | 1 byte |
| SUM        | Checksum                  | 1 byte |

### UART Output

| Header | DATA_H | DATA_L | SUM  |
|--------|--------|--------|------|
| 0xFF   | 0x07   | 0xA1   | 0xA7 |

Note: checksum only reserves the low 8-bits of the accumulated value.

```
SUM=(Header+Data_H+Data_L)&0x00FF
=(0xFF + 0x07 + 0xA1)&0x00FF
=0xA7;
```

Copy

Distance= Data\_H\*256+ Data\_L=0X07A1;

Equal to 1953 when converted into decimal;

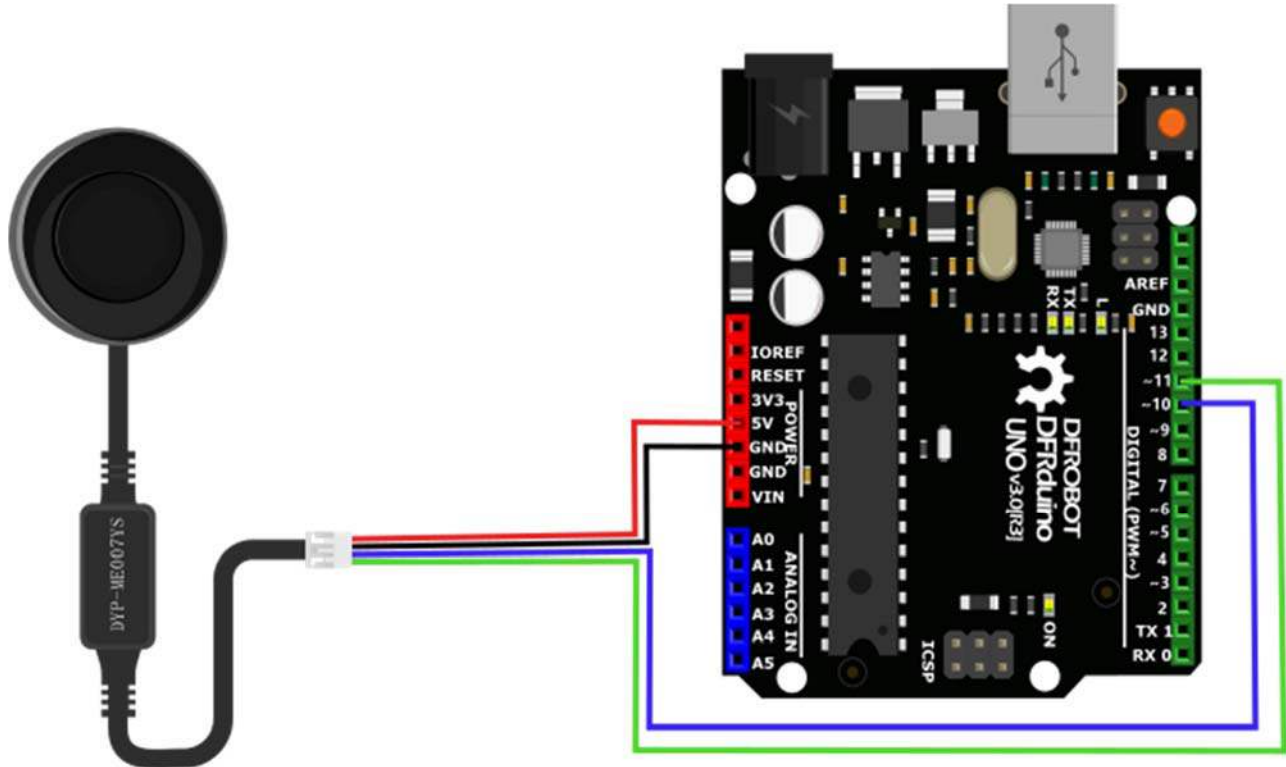
Represent the current measured distance is 1953mm.

## Arduino Platform

### Preparation

- Arduino UNO
- UNO IO Sensor Expansion Board
- ME007YS Ultrasonic Sensor
- 4P Connector

## Connection



## Sample Code

```
/*
 *@File : DFRobot_Distance_ME007YS.ino
 *@Brief : This example use ME007YS ultrasonic sensor to measure distance
 *          With initialization completed, We can get distance value
 *@Copyright [DFRobot](http://www.dfrobot.com),2016
 *          GUN Lesser General Pulic License
 *@version V1.0
 *@data 2019-8-28
 */

#include <SoftwareSerial.h>

SoftwareSerial mySerial(11,10); // RX, TX
unsigned char data[4]={};
float distance;

void setup()
{
  Serial.begin(57600);
  mySerial.begin(9600);
}

void loop()
```

```

{
  do{
    for(int i=0;i<4;i++)
    {
      data[i]=mySerial.read();
    }
  }while(mySerial.read() != 0xff);

  mySerial.flush();

  if(data[0] == 0xff)
  {
    int sum;
    sum=(data[0]+data[1]+data[2])&0x00FF;
    if(sum==data[3])
    {
      distance=(data[1]<<8)+data[2];
      if(distance>280)
      {
        Serial.print("distance=");
        Serial.print(distance/10);
        Serial.println("cm");
      }else
      {
        Serial.println("Below the lower limit");
      }
    }else Serial.println("ERROR");
  }
  delay(100);
}

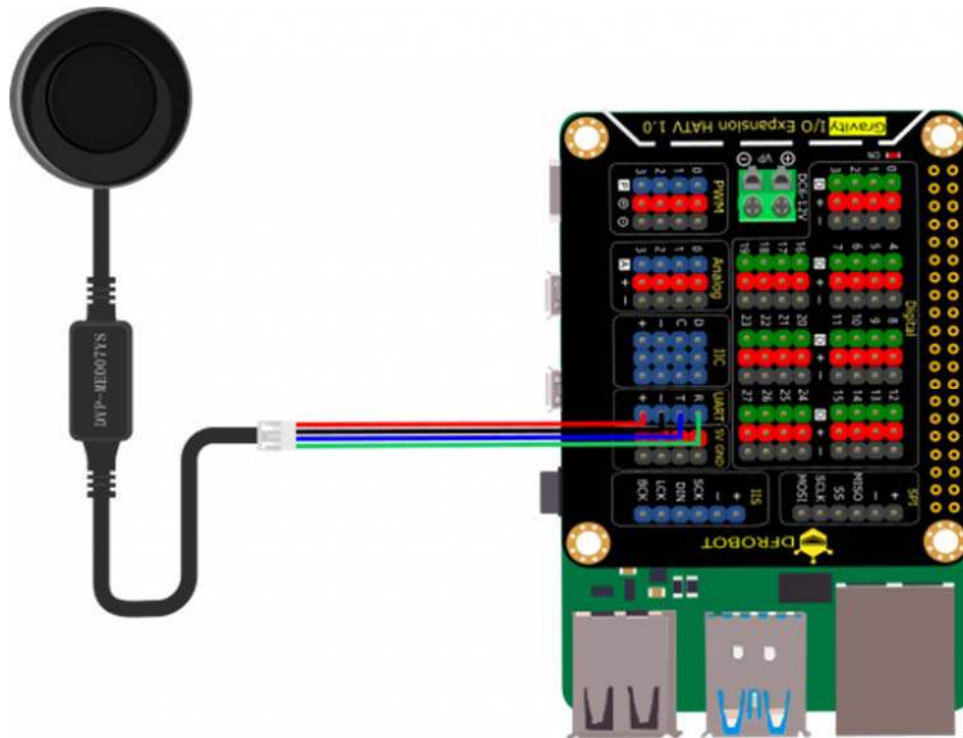
```

## Raspberry Pi Platform

### Preparation

- Raspberry Pi 4B
- Raspberry Pi IO Expansion Board
- ME007YS Ultrasonic Sensor
- 4P Connector

### Raspberry Pi Connection



## Sample Code

Download the [Ultrasonic Sensor Library](#)

```
# -*- coding:utf-8 -*-  
  
...  
# demo_get_distance.py  
#  
# Connect board with raspberryPi.  
# Run this demo.  
#  
# Connect ME007YS to UART  
# get the distance value  
#  
# Copyright [DFRobot](http://www.dfrobot.com), 2016  
# Copyright GNU Lesser General Public License  
#  
# version V1.0  
# date 2019-8-31  
...  
  
import time  
  
from DFRobot_RaspberryPi_A02YYUW import DFRobot_A02_Distance as Board  
  
board = Board()
```



```
def print_distance(dis):
    if board.last_operate_status == board.STA_OK:
        print("Distance %d mm" %dis)
    elif board.last_operate_status == board.STA_ERR_CHECKSUM:
        print("ERROR")
    elif board.last_operate_status == board.STA_ERR_SERIAL:
        print("Serial open failed!")
    elif board.last_operate_status == board.STA_ERR_CHECK_OUT_LIMIT:
        print("Above the upper limit: %d" %dis)
    elif board.last_operate_status == board.STA_ERR_CHECK_LOW_LIMIT:
        print("Below the lower limit: %d" %dis)
    elif board.last_operate_status == board.STA_ERR_DATA:
        print("No data!")

if __name__ == "__main__":
    dis_min = 0 #Minimum ranging threshold: 0mm
    dis_max = 4500 #Highest ranging threshold: 4500mm
    board.set_dis_range(dis_min, dis_max)
    while True:
        distance = board.getDistance()
        print_distance(distance)
        time.sleep(0.3) #Delay time < 0.6s
```

## FAQ

---

For any questions, advice or cool ideas to share, please visit the [DFRobot Forum](#)