



## URM07-UART Ultrasonic Sensor SKU: SEN0153



### Introduction

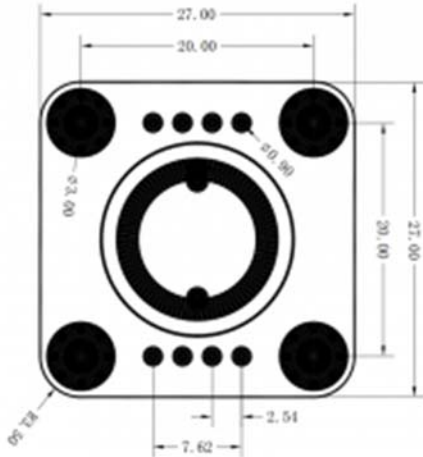
An ultrasonic sensor measures distance by calculating the time difference between sending a signal and receiving an echo from an object. It uses echolocation to measure distances, just like a bat.

The URM07 is an ultra low-power sensor. The effect range is up to 7.5m, and it has a wide range power supply of 3.0 - 5.5V. It is directly compatible with 3.3V or 5V devices such as Arduino, Raspberry Pi .etc. The average operating current is only 5mA and standby current is only 14uA.

According to the principle of ultrasound sensors, the accuracy of the distance value will be affected by the air temperature and wind direction, so using a DFRobot URM07 built-in temperature sensor, you can read the ambient temperature, and automatically calculate temperature compensation. The sensor uses an integrated ultrasonic probe with a detection angle of approximately 60° in a compact and lightweight unit. The unit uses a 2.54mm pitch 4Pin interface, using UART communication, compatible with most common microcontrollers.

URM07 UART ultrasonic distance sensors have undertaken rigorous road testing and optimization. We guarantee that this sensor has first-class response speed, ultra high stability, the highest sensitivity and ultra low power consumption. If you have harsh environmental requirements for your sensor performance in your design, the URM07 is definitely the best choice.

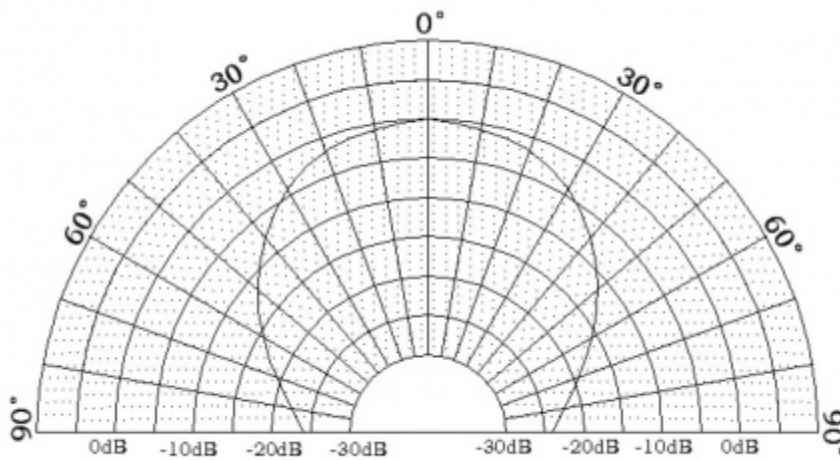
## Features



## Dimension

- Streamlined design
- High Stability
- High Sensitivity
- Built-in temperature compensation
- Low Power Consumption
- Reverse polarity protection (short-term protection)
- Overvoltage Protection
- Convenient connection and Usage

## Specification



## URM07 Detecting Angle

- Operating Voltage: 3.0 ~ 5.5V DC
- I/O Operating Mode: Open Drain (OD), integrated pull-up resistor
- Effective Range: 20 ~ 750 cm
- Direction Angle: 60 °
- Standby Current: <14 uA (mainly by the power chip consumption, the core circuit consumption <1uA)
- Peak Current: <9 mA
- Average Current: <5 mA (measured under continuous measurement, for reference only)
- Distance Resolution: 1cm
- Distance Error: 1%
- Operating Temperature Range: -10 ~ 70 °C
- Temperature Error: ± 1 °C
- Operating Humidity Range: RH <75%
- Acoustic Frequency: 38 ~ 42KHz
- Boot Start Time: <20ms
- Measurement Period: <60ms
- Dimension: 27 \* 27 mm/ 1.06 \* 1.06 inches
- Weight: 4.2g.

### Communication Protocol

- Factory parameters:
- Serial parameters: 19200 (bps) 8N1
- Factory address: 0x11

### Communication Command Frame Format:

Header		Address	Data Length	Command	Data	Checksum
0x55	0xAA	1 byte	1 byte	1 byte	Data 1~Data n	1 byte

### Commands List:

Read distance	Read temperature	Set address	Set baud rate
0x02	0x03	0x55	0x08

### Read the Measured Distance

The host sends a frame command to the ultrasonic module through the UART interface to trigger the module to start detecting, and then receives the distance value command returned by the module.



### URM07 Distance Measurement

For example, the address of the ultrasonic module is 0x11, the host sends: **0x55 0xAA 0x11 0x00 0x02 0x12 [ 55 AA 11 00 02 12 ]**

#### Description:

```

Frame header ----- 0x55
Frame header ----- 0xAA
Device address ----- 0x11
Data length ----- 0x00
Command ----- 0x02
Checksum ----- 0x12

```

The ultrasonic module returns data as: **0x55 0xAA 0x11 0x02 0x02 0x00 0xCA 0xDE**

#### Description:

```

Frame header ----- 0x55
Frame header ----- 0xAA
Local address ----- 0x11
Data length ----- 0x02
Command ----- 0x02
Distance High ----- 0x00
Distance low ----- 0xCA (Distance value 0x00CA unit is centimeter, that is, decimal 202 cm)
Checksum ----- 0xDE

```

## Read the Measured Temperature

The host reads the current temperature measured by the ultrasonic module via the UART interface.



### URM07 Temperature Measurement

For example, the ultrasonic module address is 0x11, the host sends: **0x55 0xAA 0x11 0x00 0x03 0x13 [ 55 AA 11 00 03 13 ]**

#### Description:

```
Frame header ----- 0x55
Frame header ----- 0xAA
Device address ----- 0x11
Data length ----- 0x00
Command ----- 0x03
Checksum ----- 0x13
```

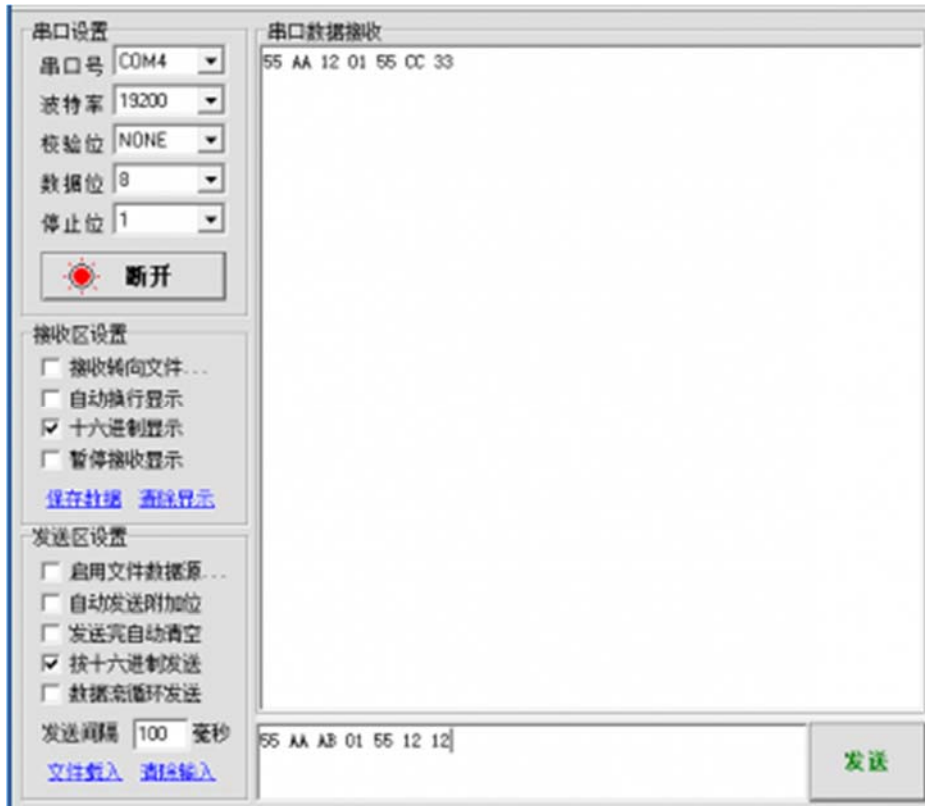
The ultrasonic module returns data as: **0x55 0xAA 0x11 0x02 0x03 0x00 0xFF 0x14**

#### Description:

```
Frame header ----- 0x55
Frame header ----- 0xAA
Local address ----- 0x11
Data length ----- 0x02
Command ----- 0x03
Temperature high ----- 0x01
Temperature low ----- 0x13 (temperature value 0x0113, 27.5 degrees. Note: T =
Returned value/10)
Checksum ----- 0x29
```

## Set the Address of the Ultrasonic Module

The host sets the ultrasonic module address via the UART interface.



### URM07 Modify the Address

For example, set the ultrasonic module address 0x12, the host sends: **0x55 0xAA 0xAB 0x01 0x55 0x12 0x12 [ 55 AA AB 01 55 12 12 ]**

Description:

Frame header ----- 0x55

Frame header ----- 0xAA

Device address ----- 0xAB (0xAB for the broadcast address, that is, all the common address of the module, you can use 0xAB instead of the device itself address if you are not sure the current Address)

Data length ----- 0x01

Command ----- 0x55

Set the address ----- 0x12 (need to set the device address 0x12)

Checksum ----- 0x12

Set the success of the ultrasonic module return command **0x55 0xAA 0x12 0x01 0x55 0xCC 0x33**

Description:

Frame header ----- 0x55

Frame header ----- 0xAA

Device address ----- 0x12 (modified device address)

Data length ----- 0x01

Command ----- 0x55

Operation status ----- 0xCC (0xCC - operation completed 0xEE - operation failed)

Checksum ----- 0x33

Set the UART Communication Baud Rate

The host sets the ultrasonic module communication baud rate via the UART interface.



URM07 Modify the Baudraet

For example, if the address of the ultrasonic module is set to 0x11, the host sends: **0x55 0xAA 0x11 0x01 0x08 0x0B 0x24 //Set baudrate to 256000bps [ 55 AA 11 01 08 0B 24 ]**

Description:

Frame header ----- 0x55  
 Frame header ----- 0xAA  
 Device address ----- 0x11  
 Data length ----- 0x01  
 Command ----- 0x08  
 Baud rate selection --- 0x0B  
 Checksum ----- 0x24

If the setting is successful, the ultrasonic module returns the command as: **0x55 0xAA 0x11 0x01 0x08 0xCC 0xE5**

Description:

Frame header ----- 0x55  
 Frame header ----- 0xAA  
 Device address ----- 0x11  
 Data length ----- 0x01  
 Command ----- 0x08  
 Operation status ----- 0xCC (0xCC - operation completed 0xEE - operation failed)  
 Checksum ----- 0xE5

Baud rate Selection List:

1200bps	2400bps	4800bps	9600bps	14400bps	19200bps	28800bps	38400bps	57600bps	115200bps	128000bps	256000bps
0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B



For example, when the device address is default 0x11, the baud rate setting command is as follows:

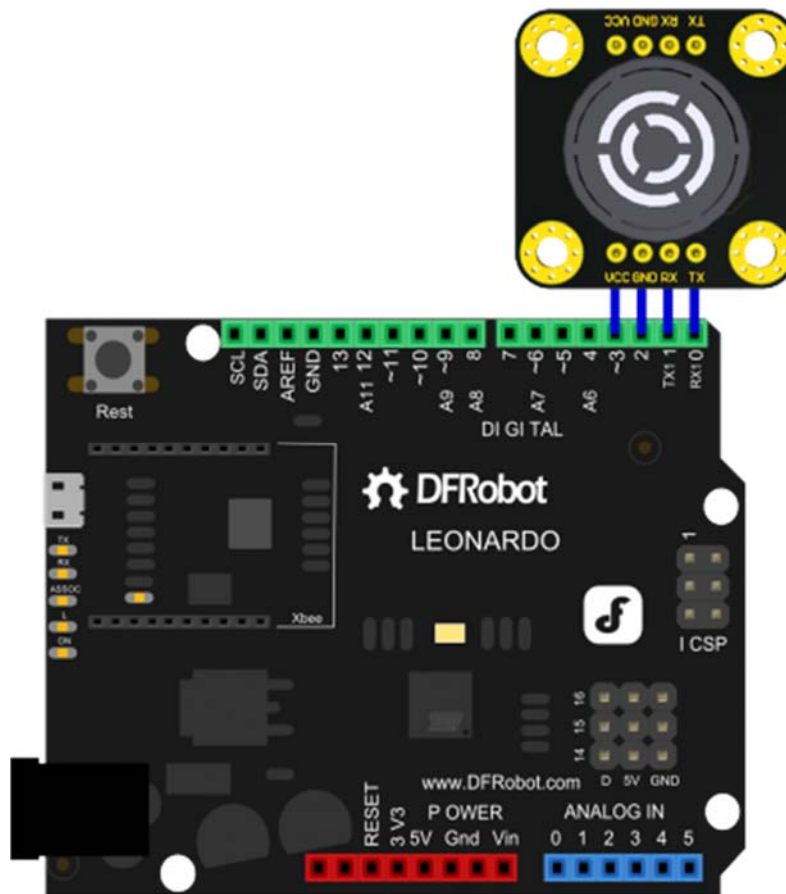
```
55 AA 11 01 08 00 19 //Set the baud rate 1200bps
55 AA 11 01 08 01 1A //Set the baud rate 2400bps
55 AA 11 01 08 02 1B //Set the baud rate 4800bps
55 AA 11 01 08 03 1C //Set the baud rate 9600bps
55 AA 11 01 08 04 1D //Set the baud rate 14400bps
55 AA 11 01 08 05 1E //Set the baud rate 19200bps
55 AA 11 01 08 06 1F //Set the baud rate 28800bps
55 AA 11 01 08 07 20 //Set the baud rate 38400bps
55 AA 11 01 08 08 21 //Set the baud rate 57600bps
55 AA 11 01 08 09 22 //Set the baud rate 115200bps
55 AA 11 01 08 0A 23 //Set the baud rate 128000bps
55 AA 11 01 08 0B 24 //Set the baud rate 256000bps
```

## Arduino URM07 Ultrasonic Tutorial

### Requirements

- **Hardware**  
DFRduino UNO (or similar) x 1  
URM07-UART Ultrasonic Sensor x1  
M-M/F-M/F-F Jumper wires
- **Software**  
Arduino IDE [Click to Download Arduino IDE from Arduino®](https://www.arduino.cc/en/Main/Software%7C)  
<https://www.arduino.cc/en/Main/Software%7C>

## Read Distance Value via Arduino (Single)



### Arduino URM07 Ultrasonic Single Control

#### Description:

Default Address: 0x11; Default Baudrate: 19200 bps  
Leonardo GPIO Power Supply. (40mA)

```
// # Author: Strictus.zhang@dfrobot.com
// # Date: 20.08.2016
// # Product Name: URM07-UART Ultrasonic Sensor
// # SKU: SEN0153
// # version number: 1.0
// # Code Description: 20-750cm distance measurement, the received data is not verified

// # Connection: Arduino LEonardo GPIO Power Supply
// # Pin VCC (URM07 V1.0) -> D3      (Arduino Leonardo)
```

```

// # Pin GND (URM07 V1.0) -> D2      (Arduino Leonardo)
// # Pin  RX (URM07 V1.0) -> TX1/D1 (Arduino Leonardo)
// # Pin  TX (URM07 V1.0) -> RX1/D0 (Arduino Leonardo)

#define header_H      0x55 //Header
#define header_L      0xAA //Header
#define device_Addr  0x11 //Address
#define data_Length  0x00 //Data length
#define get_Dis_CMD  0x02 //Command: Read Distance
#define checksum      (header_H+header_L+device_Addr+data_Length+get_Dis_CMD) /
/Checksum

unsigned char i=0;
unsigned int  Distance=0;
unsigned char Rx_DATA[8];
unsigned char CMD[6]={
    header_H,header_L,device_Addr,data_Length,get_Dis_CMD,checksum}; //Distance
command package

void setup() {
    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    digitalWrite(3, HIGH); //Ultrasonic VCC
    digitalWrite(2, LOW);  //Ultrasonic GND
    Serial1.begin(19200); //Serial1: Ultrasonic Sensor Communication Serial Po
rt, Baudrate: 19200
    Serial.begin(19200);  //Serial: USB Serial Data output, baudrate: 19200
}

void loop() {
    for(i=0;i<6;i++){
        Serial1.write(CMD[i]);
    }
    delay(150); //Wait for the result
    i=0;
}

```

```

while (Serial1.available()){ //Read the return data (Note: this demo is only for the reference, no data verification)
    Rx_DATA[i++]=(Serial1.read());
}
Distance=((Rx_DATA[5]<<8)|Rx_DATA[6]); //Read the distance value
Serial.print(Distance); //print distance value
Serial.println("cm");
}

```

```

1 // # 作者 : Strictus.zhang@dfrobot.com
2 // # 日期 : 20.08.2016
3 // # 产品名 : URM07-DART单探头超低功耗超声波测距传感器
4 // # SKU : SEW0153
5 // # 版本号 : 1.0
6 // # 代码说明: 20-750cm 距离测量, 接收数据未作校验处理
7
8 // # 连接: (单个探头可以使用IO接口供电)
9 // # Pin VCC (URM07 V1.0) -> D2 (Arduino Leonardo)
10 // # Pin GND (URM07 V1.0) -> D3 (Arduino Leonardo)
11 // # Pin RX (URM07 V1.0) -> TX1/D1 (Arduino Leonardo)
12 // # Pin TX (URM07 V1.0) -> RX1/D0 (Arduino Leonardo)
13
14 #define header_H 0x55 // 帧头
15 #define header_L 0xAA // 帧头
16 #define device_Addr 0x11 // 模块地址
17 #define data_Length 0x00 // 数据长度
18 #define get_Dis_CMD 0x02 // 距离测量命令
19 #define checksum (header_H*header_L+device_Addr+data_Length+get_Dis_CMD)
20
21 unsigned char i=0;
22 unsigned int Distance=0;
23 unsigned char Rx_DATA[8];
24 unsigned char CMD[6]={header_H, header_L, device_Addr, data_Length, get_Dis_CMD, checksum};
25
26 void setup() {
27   pinMode(2, OUTPUT);
28   pinMode(3, OUTPUT);
29   digitalWrite(2, HIGH);
30   digitalWrite(3, LOW); //D2和D3分别作为VCC和GND为模块提供电源
}

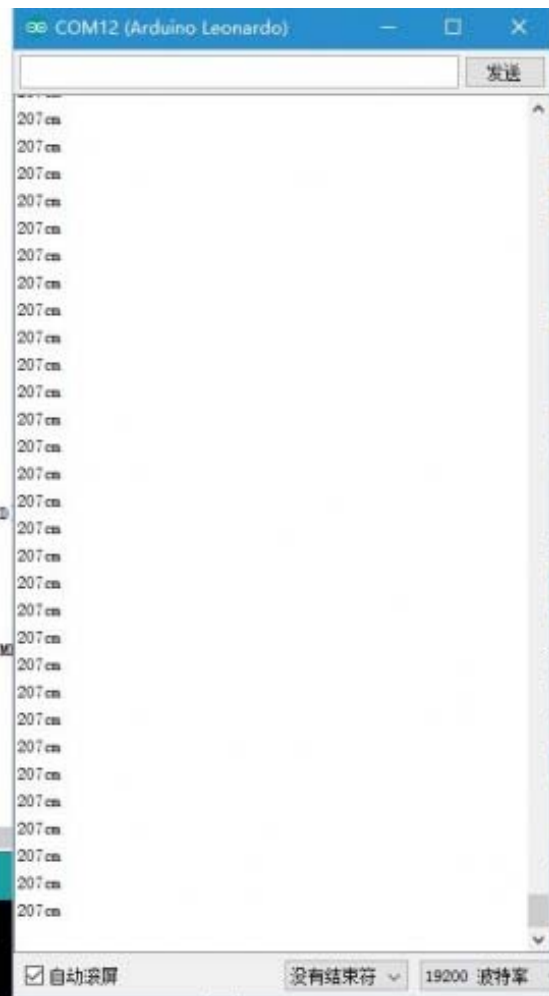
```

上传成功

```

avrduide: verifying ...
avrduide: 6112 bytes of flash verified
avrduide done. Thank you

```



URM07 Distance Display

## Read Temperature Value via Arduino (Single)

```
// # Author: Strictus.zhang@dfrobot.com
// # Date: 20.08.2016
// # Product Name: URM07-UART Ultrasonic Sensor
// # SKU: SEN0153
// # version number: 1.0
// # Code Description: Temperature measurement, the received data is not verified

// # Connection: Arduino LEonardo GPIO Power Supply
// # Pin VCC (URM07 V1.0) -> D3      (Arduino Leonardo)
// # Pin GND (URM07 V1.0) -> D2      (Arduino Leonardo)
// # Pin  RX (URM07 V1.0) -> TX1/D1 (Arduino Leonardo)
// # Pin  TX (URM07 V1.0) -> RX1/D0 (Arduino Leonardo)

#define header_H      0x55 //Header
#define header_L      0xAA //Header
#define device_Addr   0x11 //Address
#define data_Length   0x00 //Data length
#define get_Temp_CMD  0x03 //Command: Read Temperature
#define checksum      (header_H+header_L+device_Addr+data_Length+get_Temp_CMD)
//check sum

unsigned char i=0;
int temperature=0;
unsigned char Rx_DATA[8];

unsigned char CMD[6]={header_H,header_L,device_Addr,data_Length,get_Temp_CMD,checksum}; //Temperature Command package

void setup() {
    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    digitalWrite(3, HIGH);    // Ultrasonic Sensor VCC
```

```

    digitalWrite(2, LOW);    // Ultrasonic Sensor GND

    Serial1.begin(19200);   // Serial1: Ultrasonic Sensor Communication Serial Port, Baud rate: 19200

    Serial.begin(19200);    // Serial: USB Serial Monitor
}

void loop() {
    for(i=0;i<6;i++){
        Serial1.write(CMD[i]);
    }

    delay(50); //Wait Data Return

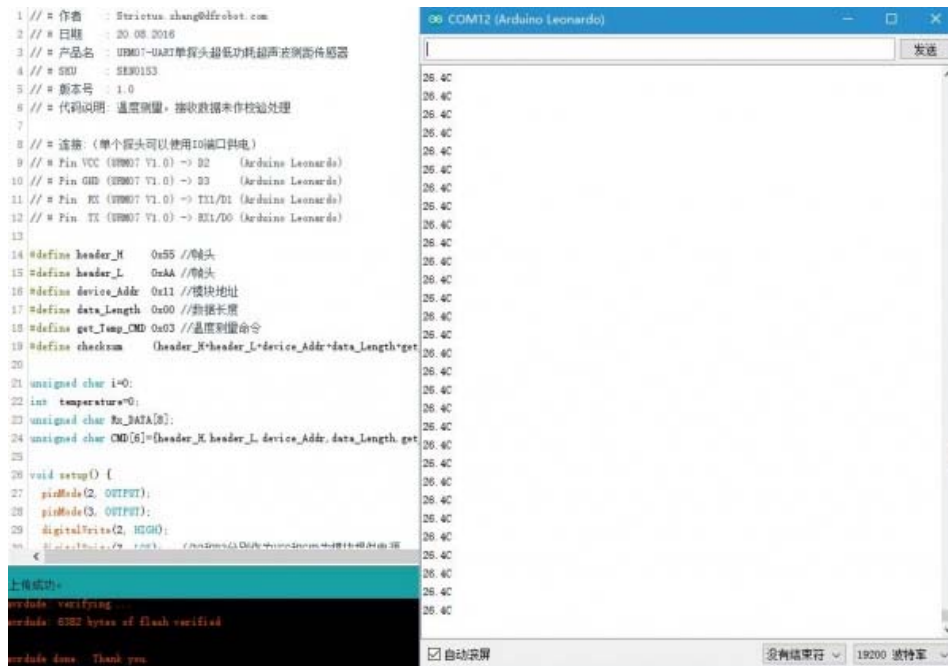
    i=0;

    while (Serial1.available()){ //Read returned Data (Note: Demo is just for Reference , no data verification)
        Rx_DATA[i++]=(Serial1.read());
    }

    temperature=((Rx_DATA[5]<<8)|Rx_DATA[6]); //Read temperature Value (10 times)

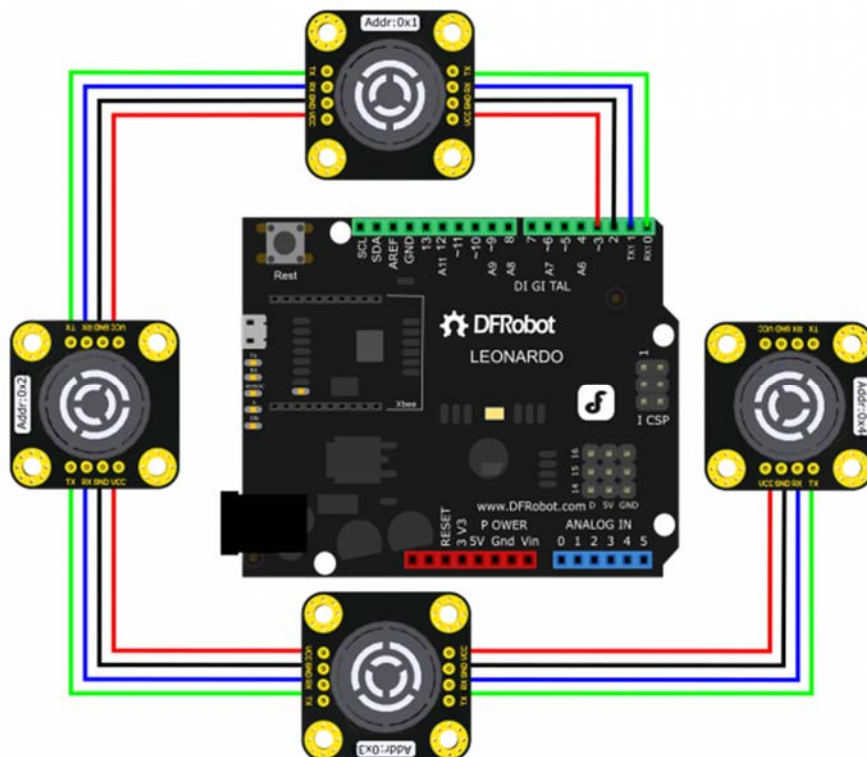
    Serial.print(temperature/10);           //Print Temperature
    Serial.print('.');
    Serial.print(temperature%10);
    Serial.println("C");
}

```



URM07 Temperature Display

### Read Distance Value via Arduino (Cascade)



URM07 Cascade Connection

## Description:

Set different address first: 0x11; 0x22; 0x33; 0x44  
Leonardo GPIO Power Supply. (40mA)  
Default Baudrate: 119200 bps

```
// # Author: Strictus.zhang@dfrobot.com
// # Date: 20.08.2016
// # Product Name: URM07-UART Ultrasonic Sensor
// # SKU: SEN0153
// # version number: 1.0
// # Code Description: Cascade Connection: Temperature & Distance measurement
, the received data is not verified

// # Connection: Arduino LLeonardo GPIO Power Supply
// # Pin VCC (URM07 V1.0) -> D3      (Arduino Leonardo)
// # Pin GND (URM07 V1.0) -> D2      (Arduino Leonardo)
// # Pin  RX (URM07 V1.0) -> TX1/D1 (Arduino Leonardo)
// # Pin  TX (URM07 V1.0) -> RX1/D0 (Arduino Leonardo)

unsigned char i=0,j=0;
unsigned int  Distance=0;
unsigned char Rx_DATA[8];
unsigned char CMD[4][6]={
                                {0x55,0xAA,0x11,0x00,0x02,0x12},
                                {0x55,0xAA,0x22,0x00,0x02,0x23},
                                {0x55,0xAA,0x33,0x00,0x02,0x34},
                                {0x55,0xAA,0x44,0x00,0x02,0x45},
}; //Distance Measurement Package

void setup() {
    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    digitalWrite(3, HIGH); //Ultrasonic Sensor VCC
```



```

    digitalWrite(2, LOW);    //Ultrasonic Sensor GND

    Serial1.begin(19200);   //Serial1: Ultrasonic Sensors Serial Communication
    Port, baudrate: 19200

    Serial.begin(19200);    //Serial: USB Serial Monitor
}

void loop() {
    for(j=0;j<4;j++)
    {
        for(i=0;i<6;i++){
            Serial1.write(CMD[j][i]);
        }
        delay(150);        //Wait returned result
        i=0;
        while (Serial1.available()){ //Read Returned Value (This demo is only for r
        eference, no data verification)
            Rx_DATA[i++]=(Serial1.read());
        }
        Distance=((Rx_DATA[5]<<8)|Rx_DATA[6]); //Read distance value
        Serial.print("URM07-UART module[");    //print distance
        Serial.print(j);
        Serial.print("]get_Dis= ");
        Serial.print(Distance);
        Serial.println("cm");
    }
    Serial.print("\r\n\r\n");
}

```

```
1 // # 作者 : Strictus.zhang@dfrobot.com
2 // # 日期 : 20.08.2016
3 // # 产品名 : URM07-UART单探头超低功耗超声波测距传感器
4 // # SKU : SEN0153
5 // # 版本号 : 1.0
6 // # 代码说明: 4个模块级联, 温度、距离测量, 接收数据未作校验处理
7
8 // # 连接: (四个模块并联, 使用IO端口供电)
9 // # Pin VCC (URM07 V1.0) -> D2 (Arduino Leonardo)
10 // # Pin GND (URM07 V1.0) -> D3 (Arduino Leonardo)
11 // # Pin RX (URM07 V1.0) -> TX1/D1 (Arduino Leonardo)
12 // # Pin TX (URM07 V1.0) -> RX1/D0 (Arduino Leonardo)
13
14 unsigned char i=0, j=0;
15 unsigned int Distance=0;
16 unsigned char Rx_DATA[8];
17 unsigned char CMD[4][6]={
18     {0x55, 0xAA, 0x11, 0x00, 0x02, 0x12},
19     {0x55, 0xAA, 0x22, 0x00, 0x02, 0x23},
20     {0x55, 0xAA, 0x33, 0x00, 0x02, 0x34},
21     {0x55, 0xAA, 0x44, 0x00, 0x02, 0x45},
22 }; //距离测量命令包
23
24 void setup() {
25     pinMode(2, OUTPUT);
26     pinMode(3, OUTPUT);
27     digitalWrite(2, HIGH);
28     digitalWrite(3, LOW); //D2和D3分别作为VCC和GND为模块提供电源
29     Serial1.begin(19200); //使用Serial1与模块通信, 波特率为19200
30     Serial.begin(19200); //使用Serial作为数据输出的串口
31 }
32
```

```
上传成功。
avrdude: verifying ...
avrdude: 6284 bytes of Flash verified
avrdude done. Thank you.
```



## URM07 Cascade

For any questions, advice or cool ideas to share, please visit the **DFRobot Forum**.