

Description

The JSN-SR04T Waterproof Ultrasonic Range Finder uses ultrasonic waves to detect the presence of and measure the distance to objects in front of it. The V3.0 is the latest version of the module with additional measurement modes available over earlier versions.

PACKAGE INCLUDES:

- JSN-SR04T V3.0 Waterproof Ultrasonic Range Finder
 - JSN-SR04T V3.0 Electronic Control Module
 - Waterproof Transducer with 2.5 Meter Cable

KEY FEATURES OF JSN-SR04T V3.0 WATERPROOF ULTRASONIC RANGE FINDER:

- 23 to 600 cm (9" to 20 feet) detection range
- 40 kHz operation
- ± 37.5 degree field of view
- 2.5 meter (8') transducer cable
- 3.3 & 5V operation and logic compatible (see notes below about 3.3V operation)

The JSN-SR04T is very similar to the standard ultrasonic modules like the HC-SR04 with the exception that it uses a single waterproof transducer instead of two and it also provides more methods for interfacing to a microcontroller.

Because the transducer is waterproof, it can be used for indoor or outdoor sensing such as providing a car parking sensor, security alarm, obstacle avoidance or monitoring the water level in water storage tanks,

Like any ultrasonic sensor that depends on sound waves bouncing off an object to detect it, it is able to detect hard objects quite well including the surface of calm water, but detection of soft fabrics or other sound absorbing material is less consistent and empirical testing may be required to determine suitability for a particular application.

Because these modules use sound waves for detection, they are not sensitive to light sources or optically reflective surfaces like IR can be. In addition, because sound travels relatively slowly it is possible to measure the time that it takes for the sound to travel to the object and be echo'd back and therefore the distance to the object can be calculated with a fair amount of accuracy which can be quite handy.

The detection and measurement range is from 23 cm up to 600 cm which is about 9" to 20 feet with a stated accuracy of 1cm. The ultrasonic sound is pulsed at 40 kHz and is not audible to the human ear.

If an echo is not returned (no object detected), the module will timeout after a fixed delay. This is required to prevent the module from hanging if there is no return echo.

The module has two basic modes of operation.

Ping mode works the same basic way as the HC-SR04, HY-SRF05, US-100 and other ultrasonic modules. The main difference is that the JSN-SR04T requires a longer 20uS TRIG pulse for stable readings. It also can run Ping mode in an automatic mode of operation where it is automatically triggered every 200mS.

The way Ping mode works is as follows:

1. The module is normally idle.
2. A 20 uSec or wider pulse is sent to the TRIG pin on the module, usually by a microcontroller. This will cause the module to start a detection cycle.
3. The module emits eight bursts of 40KHz sound and sets it's ECHO pin output HIGH.
4. When the signal is reflected back from an object and is detected by the module, the ECHO pin is set back to LOW.
5. By measuring the amount of time that the ECHO pin is held HIGH, the distance to the object can be calculated using the basic formula (**Time that ECHO is held HIGH * Speed of sound / 2**) The divide by 2 is because the sound has to travel in both directions (both out and back) and we want to know just the out distance.

Serial mode operates at 9600 baud and works similar to serial mode on the US-100 module. The main difference is that it also returns a start byte and checksum byte. It also can be operated in an automatic mode of operation where the module is automatically triggered every 100mS.

The way Serial mode works is as follows:

1. The module is normally idle.
2. A command byte of 0x55 on the RX pin initiates a detection cycle.
3. The module responds with 4 bytes of information
 1. Start byte of 0xFF
 2. Most Significant Byte (MSB)
 3. Least Significant Byte (LSB)
 4. Checksum Byte

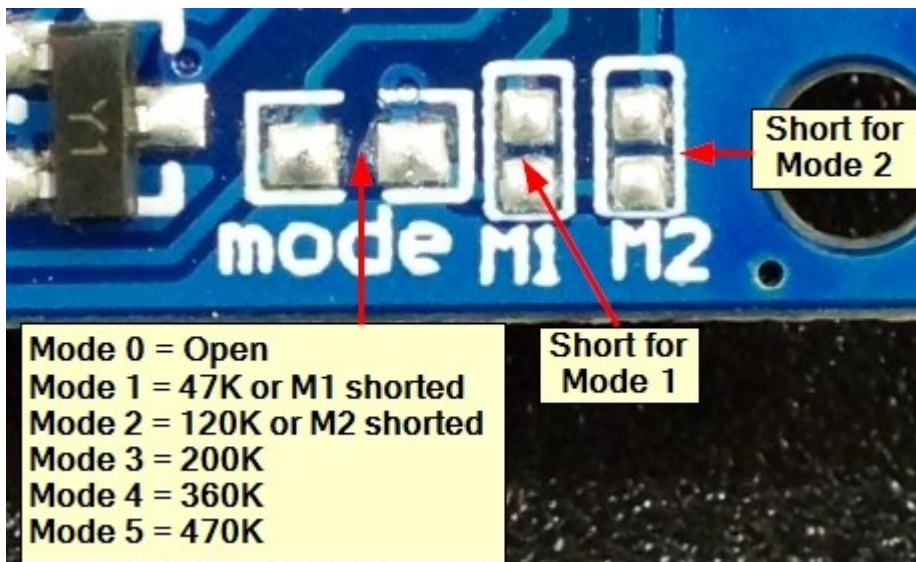
The returned value can be calculated using the formula (**MSB * 256**) + **LSB**.

The checksum byte is calculated using **0xFF + MSB + LSB**. It returns the least significant byte of the calculation only.

Operating Modes

The module can operate in 6 different modes of operation using either the Ping or Serial mode of operation. These are set by shorting the M1 or M2 pads or placing a resistor of a certain value across the MODE pads. The resistor value sets an analog voltage that is read by an ADC, so the exact value isn't overly critical.

Mode 0 and Mode 2 are the most commonly used modes.



MODE 0: Ping Mode

Mode 0 is the default mode when there is no resistor on the MODE pads and neither M1 or M2 pads are shorted and is how the module is shipped.

In this mode, the module operates very similarly to the HC-SR04 and similar ultrasonic range finder modules and sample code for those modules can be used with this module with one exception. With the HC-SR04, the trigger pin is held high for 10uS. With the JSN-SR04T, the trigger pin must be held high longer – usually about 20uS for reliable operation. The first test program below uses this mode.

MODE 1: Automatic Serial Mode

To enter Mode 1, the M1 pads are shorted OR a resistor value of 47K is placed across the MODE pads.

In this mode the module continuously takes measurements approximately every 100mS and outputs the distance on the TX pin at 9600 baud where it can be read using a serial RX pin on an MCU.

MODE 2: MCU Controlled Serial Mode

To enter Mode 2, the M2 pads are shorted OR a resistor value of 120K is placed across the MODE pads.

This mode is the same as Mode 1 except that the module takes a measurement only when a trigger command of 0x55 is sent to the RX pin on the module. The module then takes a single measurement and outputs the distance on its TX pin.

The minimum trigger cycle time is 60mS

MODE 3: Automatic Ping mode

To enter Mode 3, a resistor value of 200K is placed across the MODE pads.

Mode 3 is similar to Mode 0 (ping mode) except that it automatically generates the ping every 200mS and outputs a pulse width high level corresponding to the distance.

MODE 4: Low power Ping mode.

To enter Mode 4, a resistor value of 360K is placed across the MODE pads.

This mode is the same as Mode 0 except that the module is set to low power in between Pings with a stated current consumption of $< 70\mu\text{A}$.

MODE 5: Threshold Switch Output

To enter Mode 5, a resistor value of 470K is placed across the MODE pads.

In this mode, the module automatically takes a reading every 200mS and checks the measurement against a factory default threshold setting of 1.5 meters. If the detected distance is less than 1.5 meters, the Echo pin is set to HIGH. If the detected distance is greater than 1.5m, the Echo pin is set to LOW. This can be handy for using the output to directly control something like a relay or LED.

Module Connections:

There is a 4-pin header on the assembly.

1 x 4 Header

- **5V** – Connect to 5V
- **RX / TRIG** – Connect to any digital output pin on MCU. A 20uSec or wider pulse starts a measurement cycle in Ping mode of operation. Alternatively it serves as the serial RX (receive) pin.
- **TX / ECHO** – Connect to any digital input pin on MCU. This pin is held high for the duration of the the measurement cycle. Alternatively it serves as the serial TX (transmit) pin.
- **GND** – Connect to system ground. This ground needs to be in common with the MCU.

Transducer Mounting

The transducer can be mounted in a hole with a diameter of approximately 23mm. The plastic housing has rubber inserts to help grip the sides of the mounting hole.

OUR EVALUATION RESULTS:

If you do a Google search, you will find a lot of hand-wringing over using this module. This is partly due to differences in how this module behaves compared to the more common ultrasonic modules as well as the fact that it is often put into situations that the other modules are unsuitable for that creates additional challenges. Having at least 3 versions on the market also leads to some confusion. Here are a few key things to keep in mind when determining suitability for your application.

1. The Rev 3.0 as we have here and older Rev 2.0 of this module are stated to be 3.3V compatible as well as 5V. We have found 3.3V operation to be spotty at best in our testing, so if using with a 3.3V MCU, it is recommended to operate the module at 5V and use level shifting for the logic lines going to the MCU.
2. The minimum detection distance is about 23cm (9") which is greater than most ultrasonic modules. The greater minimum distance is because it uses a single ultrasonic element to both transmit and then receive the reflected ultrasonic waveform instead of using two sensors like in most of the modules like the HC-SR04.
3. The ultrasonic cone spread is specified as $75^\circ (\pm 37.5^\circ)$. This is wide compared to most other ultrasonic sensors, so it will tend to pickup objects on the sides of this cone when used in

- tight spaces. It works best in open spaces where there is a clear view to the object being detected with nothing in its peripheral vision.
4. When using in the default Ping mode of operation, the JSN-SR04T is completely software compatible with the HC-SR04 EXCEPT that the trigger must be held high for 20uS rather than the standard 10uS or else there will be numerous erroneous readings.
 5. Though the transducer is waterproof, it will not function correctly underwater, so don't start drilling into the hull of your boat quite yet for your home-brew depth finder.
 6. Similarly, water droplets on the transducer sensor can affect the sensor reading.
 7. The Rev 3.0 of the module has additional operating modes over those offered with the earlier versions.

These modules work well if the limitations noted above are taken into consideration. The most common issue is that the wide viewing cone picking up unintended targets.

The modules have a large adjustable transformer which is the metal can with a screw adjustment on top. That adjustment should not be messed with or the unit may stop working correctly.

Note that the optional HC-SR04 Ultrasonic Distance Measurement Control module (MOD-113) will not work with the JSN-SR04T because it uses a shorter trigger of about 14uS.

The first program below implements the default Ping (Mode 0) mode of operation. The second program implements the MCU controlled serial output (Mode 2) mode of operation.

This program uses the default Ping mode of operation (Mode 0) where the measured range is displayed in centimeters, inches and feet. This is the mode the module is in when it first comes out of the bag. Since the speed of sound varies with the temperature, for maximum accuracy the temperature can be measured and used in the calculation, so this is broken out in the code below but defaults to 20C.

JSN-SR04T Ultrasonic Range Finder Ping Mode of Operation Program

```

/*
 * JSN-SR04T Ultrasonic Range Finder Ping Mode Test
 *
 * Exercises the ultrasonic range finder module and prints out measured distance
 * 5V - connect to 5V
 * GND - connect to ground
 * RX/TRIG - connect to digital pin 12. Can be any digital pin
 * TX/ECHO - connect to digital pin 13. Can be any digital pin
 */
const int TRIG_PIN = 12;
const int ECHO_PIN = 13;
float temp_In_C = 20.0; // Can enter actual air temp here for maximum accuracy
float speed_Of_Sound;    // Calculated speed of sound based on air temp
float distance_Per_uSec;  // Distance sound travels in one microsecond

//
=====
// Initialization
//
=====
void setup() {
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  // Formula to calculate speed of sound in meters/sec based on temp
  speed_Of_Sound = 331.1 +(0.606 * temp_In_C);

```

```

    // Calculate the distance that sound travels in one microsecond in centimeters
    distance_Per_uSec = speed_Of_Sound / 10000.0;
    Serial.begin(9600);
}
//
=====
//  Main
//
=====
void loop() {
    float duration, distanceCm, distanceIn, distanceFt;

    digitalWrite(TRIG_PIN, HIGH);      // Set trigger pin HIGH
    delayMicroseconds(20);              // Hold pin HIGH for 20 uSec
    digitalWrite(TRIG_PIN, LOW);       // Return trigger pin back to LOW again.
    duration = pulseIn(ECHO_PIN,HIGH);  // Measure time in uSec for echo to come
    back.

    // convert the time data into a distance in centimeters, inches and feet
    duration = duration / 2.0; // Divide echo time by 2 to get time in one
    direction
    distanceCm = duration * distance_Per_uSec;
    distanceIn = distanceCm / 2.54;
    distanceFt = distanceIn / 12.0;

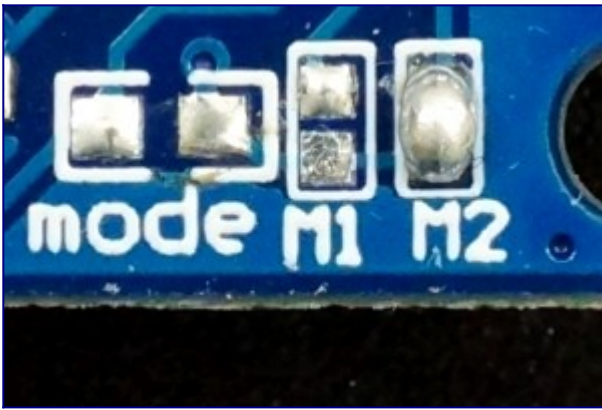
    if (distanceCm <= 0){
        Serial.println("Out of range");
    }
    else {
        Serial.print(duration, 0);
        Serial.print("uSec, ");
        Serial.print(distanceCm, 0);
        Serial.print("cm, ");
        Serial.print(distanceIn,0);
        Serial.print("in, ");
        Serial.print(distanceFt,1);
        Serial.print("ft, ");
        Serial.println();
    }
    delay(1000); // Delay between readings
}

```

The program below implements the Mode 2 – MCU controlled serial output.

A softserial port is used for maximum flexibility so that any two pins can be used, but a hardware serial port can also be used.

For this mode to work, the M2 pads should be shorted with solder or a 120K resistor can be soldered across the MODE pads.



JSN-SR04T Ultrasonic Range Finder MCU Controlled Serial Mode of Operation Program

```

/*
 * JSN_SR04T Ver 3.0 Ultrasonic Test - Serial Interface Mode
 *
 * Exercises the ultrasonic range finder module in serial mode
 * and print out the current measured distance
 * This uses 'softserial' to communicate with the module to avoid any conflict
 * with the HW serial port if using an UNO.
 * VCC - connect to 5V
 * GND - connect to ground.
 * Serial pins are cross connected RX to TX and TX to RX
 * TRIG/TX pin - connect to digital pin 11 (RX)
 * ECHO/RX - connect to digital pin 10 (TX)
 *
 * For Serial mode of operation, ensure the M2 location is shorted
 */
#include <SoftwareSerial.h>;
const int TX_PIN = 10;      // Pin for Soft Serial TX
const int RX_PIN = 11;      // Pin for Soft Serial RX
// Create instance of soft serial port for comm with module
SoftwareSerial JSN_SR04T(RX_PIN, TX_PIN);

byte StartByte = 0;         // 0xFF signifies a new packet of info coming in
byte MSByte = 0;            // Most Significant Byte of distance measurement
byte LSByte = 0;            // Least Significant Byte of distance measurement
byte CheckSum = 0;          // CheckSum = 0xFF + MSByte + LSByte
unsigned int mmDist = 0;    // Returned distance in millimeters
//
=====
// Initialization
//
=====
void setup() {
    Serial.begin(9600);
    JSN_SR04T.begin(9600);    // Initialize the software serial port
}
//
=====
// Main
//
=====
void loop() {
    JSN_SR04T.flush();        // Clear the serial port buffer
    JSN_SR04T.write(0x55);    // Request distance measurement
    delay(100);               // Give sensor time to make the measurement
    (>60mSec)

```

```

    if(JSN_SR04T.available() >= 4){ // Looking for 4 bytes to be returned
        StartByte = JSN_SR04T.read(); // Read first byte
        if (StartByte == 0xFF){ // 1st byte is 0xFF for new measurement
            MSByte = JSN_SR04T.read(); // Read in the MSB (Most Significant
Byte)
            LSByte = JSN_SR04T.read(); // Read in the LSB (Least Significant
Byte)
            CheckSum = JSN_SR04T.read(); // Read the checksum byte
            mmDist = MSByte * 256 + LSByte; // Calculate the distance from the two
bytes
            Serial.print("Distance: ");
            Serial.print(mmDist); // Print in millimeters
            Serial.print("mm / ");
            Serial.print(mmDist/25.4, 2); // Print in inches
            Serial.println("in");
        }
        else{
            JSN_SR04T.flush(); // Flush the buffer until valid start
byte
        }
    }
}

```

The board has a four M3 holes that can be used for mounting.

FURTHER READING:

David Pilling has a nice writeup with scope captures of the board waveforms for anyone wanting to learn more about the inner workings of the device.

<https://www.davidpilling.com/wiki/index.php/JSN>