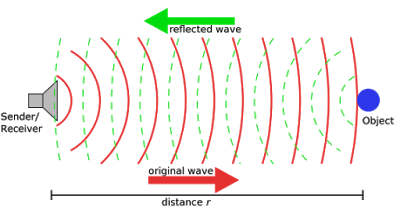
# **[Demo 37: Display distance measured by ultrasonic sensor using module 7-segment-LED-N-Digits](http://www.iotsharing.com/2017/12/display-information-using-module-7-segment-LED-N-Digits.html)**

**1. Introduction**  
Today I will show you how to use module 7-segment-LED-N-Digits to display distance which is measured by ultrasonic sensor.   
**2. Hardware and Software**  
**2.1 Ultrasonic sensor**  
I used HC - SR04.

[](https://1.bp.blogspot.com/-VF5or8dyKR8/WiO66aNjIsI/AAAAAAAAF0A/9Oiz5JmsC7A19BjRm5bnYNRI-inCG1ImgCLcBGAs/s1600/esp-ultrasonic-sensor.png)

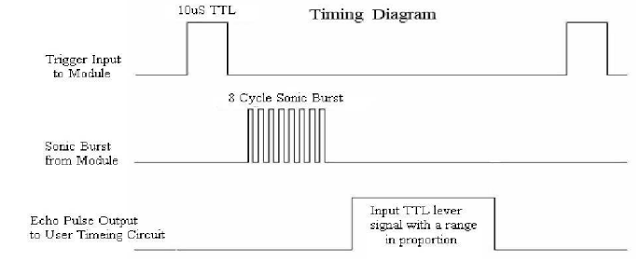
**Figure: Ultrasonic sensor HC - SR04 (Source: [Internet](http://www.micropik.com/PDF/HCSR04.pdf))**

The sensor has 2 heads: one is emit the ultrasonic and one receives it when the ultrasonic is reflect by the obstacle. The range of this sensor is 2cm -  400cm non-contact.

[](https://1.bp.blogspot.com/-qVZcOEdiH5M/WiO88pXw1JI/AAAAAAAAF0M/GeallDJ54l4cVd1GVtv68jvXwtJvgBymgCLcBGAs/s1600/esp-ultrasonic-mechanism.png)

**Figure: the operation mechanism of ultrasonic sensor**

The picture below is timing diagram of HC - SR04.



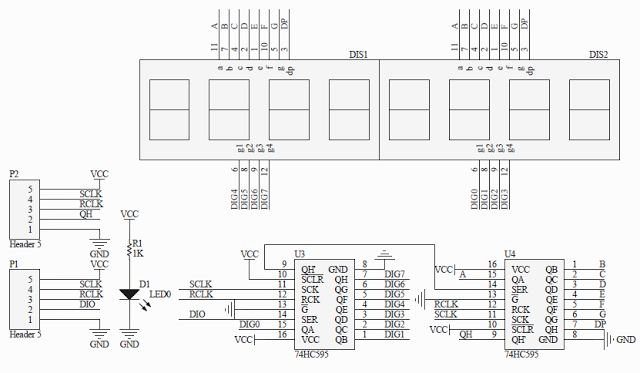
**Figure: timing diagram of HC - SR04**

The basic principle of work:   
(1) Using IO trigger for at least 10us high level signal.   
(2) The Module automatically sends eight 40 kHz and detect whether there is a ultrasonic signal back.   
(3) IF the ultrasonic signal back, through high level, time of high output IO duration is the time from sending ultrasonic to returning.   
Test distance = (High level time x velocity of sound (299 cm/us) / 2  
**2.2 7-segment-LED-8-digit module**  
**I used**

[](https://2.bp.blogspot.com/-WrvGgWV8NiY/WiPCV2kL2KI/AAAAAAAAF0o/F1H_d_weLGg52p0DAKewT--Kes78HIyKwCEwYBhgL/s1600/esp-led7seg8digit.jpg)

**Figure: 7-segment-LED-8-digit module**

This module used IC 74HC595. It is a 8 bits shift register. With this IC we can save more IO digital instead of using a lot of IO pins to trigger each segment of LED.

[](https://1.bp.blogspot.com/-ZJxfjhl0WHA/WiPDIqwu63I/AAAAAAAAF0s/2E-CvXU93aUGYSf2dc0yJXgWLADYFzxzQCLcBGAs/s1600/esp-7-seg-led-n-digit-module.png)

**Figure: schematic application of IC 74HC595**

In order to bring data to this LED module, first we shift data that will be displayed on LED then we shift the value that indicate which LED in order that will display the data. i created the library [here](https://github.com/nhatuan84/arduino-led-7-seg-N-digit.git). The library is easy to use. You create an instance of library 

**EspLed7SegNDigit ledm(SCLK, RCLK, DIO, 8);**

where SCLK, RCLK, DIO is pins that connect between ESP and LED module. 8 is number of LEDs in module.

We can use**ledm.setCharAt(8, 'd');**to set the value (here is character 'd') that will be displayed at specific LED (here is LED number 8)**.**

In order to human can see the LED clearly, the LED scan time will be default 400ms (25 frames/second). You can set it using setRefreshTime(ms). The callback function **updateDisplayCb()**will be invoked after every refreshing time to update new value for LED displaying. You should call "**ledm.clearDisplay()**" to clear LED module before updating new display. You can use **ledm.displayNum(num, 3);**to display the float number with 3 digits behind the dot character. The function "**ledm.loop();**" will run continuously to update the display on LED.  
**Pins connection:**  
**LED module**  
**+SCLK with ESP32 GPIO14+RCLK with ESP32 GPIO27**  
**+DIO with ESP32 GPIO12**  
**Ultrasonic sensor**  
**+TRIG with ESP32 GPIO25**  
**+ECHO with ESP32 GPIO33**

**2.3 Full software**

Our application has 2 [FreeRTOS tasks](http://www.iotsharing.com/2017/06/how-to-apply-freertos-in-arduino-esp32.html): 1 for ultrasonic measurement and 1 for LED updating.

|  |
| --- |
| #include "EspLed7SegNDigit.h"  //LED pins  int SCLK = 14; //pulse  int RCLK = 27; //latch  int DIO = 12; //data  //utrasonic pins  int TRIG\_PIN = 25;  int ECHO\_PIN = 33;  unsigned long startMeasure = 0;  unsigned long endMeasure = 0;  unsigned long measureTime = 0;  unsigned long distance = 0;  EspLed7SegNDigit ledm(SCLK, RCLK, DIO, 8);  void updateDisplayCb(void){  ledm.clearDisplay();  ledm.setCharAt(8, 'd');  ledm.setCharAt(7, 's');  ledm.setCharAt(6, 't');  ledm.displayNum(distance, 3);  }  void initUltra(){  pinMode(TRIG\_PIN, OUTPUT);  pinMode(ECHO\_PIN, INPUT);  }  void ultraTask(){  //trigger sensor with pulse LOW-HIGH-LOW  digitalWrite(TRIG\_PIN, LOW);  //wait 2 us  delayMicroseconds(2);  digitalWrite(TRIG\_PIN, HIGH);  //wait 10 us  delayMicroseconds(10);  digitalWrite(TRIG\_PIN, LOW);    //at the beginning ECHO pin will be pull LOW until finishing transmitting ultrasonic signal  while (digitalRead(ECHO\_PIN) == 0){  startMeasure = micros();  }  //ECHO pin will be pulled HIGH until get response  while (digitalRead(ECHO\_PIN) == 1){  endMeasure = micros();  }    //response time will be calculated by  measureTime = endMeasure - startMeasure;  //convert to cm  distance = (measureTime)/29/2;  Serial.printf("distance = %d\n", distance);  }  void setup() {  // put your setup code here, to run once:  Serial.begin(115200);  initUltra();  //this callback will be invoked for updating new ultrasonic value  ledm.setUpdateCb(updateDisplayCb);  ledm.setCharAt(8, 'd');  ledm.setCharAt(7, 's');  ledm.setCharAt(6, 't');  //in loop it take time to do ultraTask so we decrease refresh time  ledm.setRefreshTime(100);  }  long tick = 0;  void loop() {  // put your main code here, to run repeatedly:  ledm.loop();  long now = millis();  if (now - tick > 1000) {  tick = now;  ultraTask();  }  } |