

water level indecator.



May 27, 2021

bue

Contents

[Water Level Indicator 3](#_Toc73006677)

[Pin wiring: 3](#_Toc73006678)

[Ultrasonic Sensor: 3](#_Toc73006679)

[Blink: 4](#_Toc73006680)

[List of components: 5](#_Toc73006681)

[Testing the OLED Display: 6](#_Toc73006682)

[Ultrasonic sensor testing: 7](#_Toc73006683)

[Testing the Arduino Uno Board 7](#_Toc73006684)

[Figure 1:oled display connection. 3](file:///C:\Users\masri\OneDrive\المستندات\Water%20Level%20Indicator.docx#_Toc73028661)

[Figure 2:ultrasonic sensor connection. 4](file:///C:\Users\masri\OneDrive\المستندات\Water%20Level%20Indicator.docx#_Toc73028662)

[Figure 3:led connection. 4](file:///C:\Users\masri\OneDrive\المستندات\Water%20Level%20Indicator.docx#_Toc73028663)

[Figure 4:water level indicator circuit. 5](file:///C:\Users\masri\OneDrive\المستندات\Water%20Level%20Indicator.docx#_Toc73028664)

# Water Level Indicator

The Water Level Indicator employs a simple mechanism to detect and indicate the water level in an overhead tank or any other water container.

By using Arduino uno R3 and connecting oled display, ultrasonic sensor with the breadboard by wires we get our water level indicator.

## Pin wiring:

Because the OLED display uses I2C communication protocol, wiring is very simple. You just need to connect to the Arduino Uno R3 pins as shown in the table below

|  |  |
| --- | --- |
| Vin | 5V |
| GND | GND |
| SCK | A5 |
| SDA | A4 |

Pin Wiring to Arduino Uno

Figure :oled display connection.

Ultrasonic Sensor: HC-SR04 is a sensor that can measure distance. It emits an ultrasound at 40000 Hz (40kHz) 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. The configuration pin of HC-SR04 is VCC (1), TRIG (2), ECHO (3), and GND (4). The supply voltage of VCC is +3V and you can attach TRIG and ECHO pin to any Digital I/O in your Arduino Board.

Figure :ultrasonic sensor connection.

## Blink:

This example shows the simplest thing you can do with an Arduino to see physical output: it blinks the on-board LED.

Hardware Required:

Arduino Board

LED

220ohm resistor

connect one end of the resistor to the digital pin correspondent to the LED\_BUILTIN constant. Connect the long leg of the LED (the positive leg, called the anode) to the other end of the resistor. Connect the short leg of the LED (the negative leg, called the cathode) to the GND. In the diagram below we show an UNO board that has D13 as the LED\_BUILTIN value.

Figure 3:led connection.

Figure 4:water level indicator circuit.

The sensing is done by using an ultrasonic sensor which is placed on a ()cm wooden pole to give the accessibility of putting any type of containers with arrange of height from 10cm to 30cm. the water percentage and animations are shown in an oled display by percentage with an animation showing the container. When the water level reaches 100% the warning flashes from a red led lamb and stops after water level decreases. Most water level indicators are using lcd display. The Water Level Indicator implemented here uses oled display.

# List of components:

|  |  |
| --- | --- |
| Arduino uno R3 | 170L.E |
| Oled display | 65L.E |
| Ultrasonic sensor | 35L.E |
| Sensor mount | 15L.E |
| Led (red) | 0.5L.E |
| Wiers | 10L.E |
| resistor | 0.10L.E |
| breadboard | 25L.E |

# Testing the OLED Display:

After wiring the OLED display to the Arduino and installing all required libraries, you can use one example from the library to see if everything is working properly.

The following sketch displays Hello, world! message in the OLED display.

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#define SCREEN\_WIDTH 128 // OLED display width, in pixels

#define SCREEN\_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, -1);

void setup() {

Serial.begin(115200);

if(!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64

Serial.println(F("SSD1306 allocation failed"));

for(;;);

}

delay(2000);

display.clearDisplay();

display.setTextSize(1);

display.setTextColor(WHITE);

display.setCursor(0, 10);

// Display static text

display.println("Hello, world!");

display.display();

}

void loop() {

}

# Ultrasonic sensor testing:

Ultrasonic sensors work by transmitting a pulse of sound, much like sonar detectors, outside the range of human hearing. This pulse travels away from the range finder in a conical shape at the speed of sound (340 m/s). The sound reflects off an object and back to the range finder. The sensor interprets this as an echo and calculates the time interval between sending the signal and receiving the echo. This interval is then computed by a controller to determine the distance of the object.

// Ultrasonic sensor pins  
#define echoPin A0  
#define pingPin 3

void setup() {  
// put your setup code here, to run once:

Serial.begin(9600);  
// ultrasonic sensors  
pinMode(pingPin, OUTPUT);  
pinMode(echoPin, INPUT);  
}

void loop() {  
// put your main code here, to run repeatedly:  
int distance = msToCm( ping() );  
Serial.println(distance);  
}

// Helper function to manage our ultrasonic sensor.  
long ping() {  
long duration;  
digitalWrite(pingPin, LOW);  
delayMicroseconds(2);  
digitalWrite(pingPin, HIGH);  
delayMicroseconds(10);  
digitalWrite(pingPin, LOW);  
duration = pulseIn(echoPin, HIGH);

return duration;  
long msToCm(long microseconds) {  
return microseconds / 29 / 2;  
}

# Testing the Arduino Uno Board

To verify that you are receiving correct data you can test it by setting each channel to ground and power and read the output in the Serial Monitor. To do this you need a wire to connect between the input connectors and power connectors on the Arduino Uno board.

Connect one end of the wire to A0 port

Connect the other end to GND port

Analog0 in the Serial Monitor should now read 0.0 volts

Remove the wire from GND and connect it to 5V

Analog0 should now read approximately 5.0 volts

Remove the wire from 5V and connect it to 3.3V

Analog0 should now read approximately 3.3 volts

Repeat the same procedure with A1, D2 and D3

Do you get the same value from the digital port in both 3.3V and 5V?

Now, your display prints the “Hello, world!” message in FreeSerif font.

Figure :the result of testing oled.