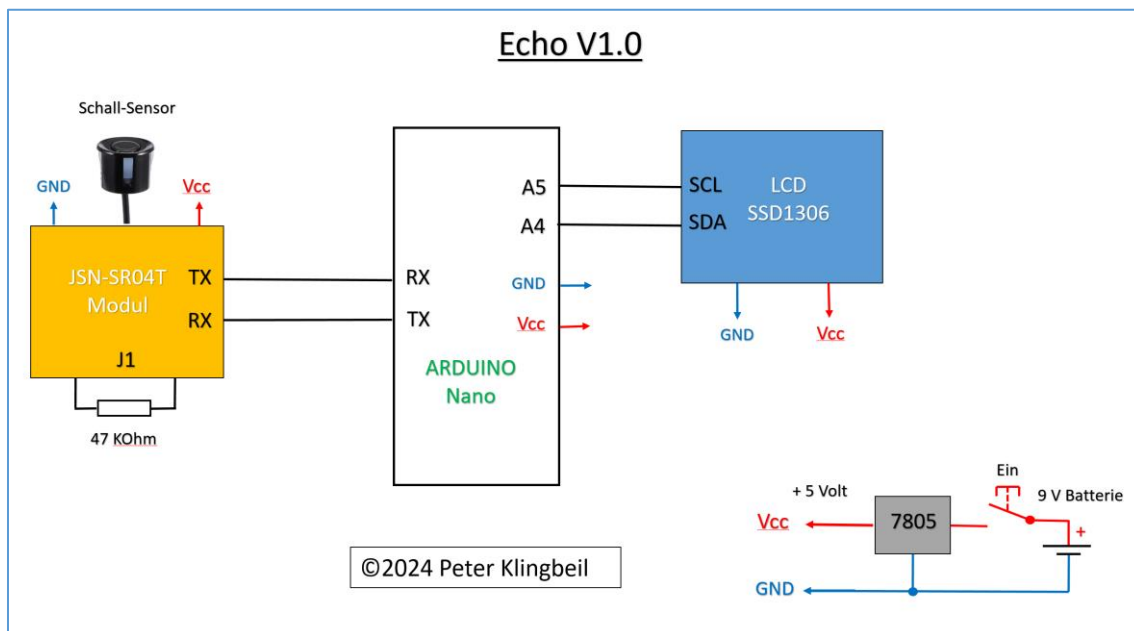
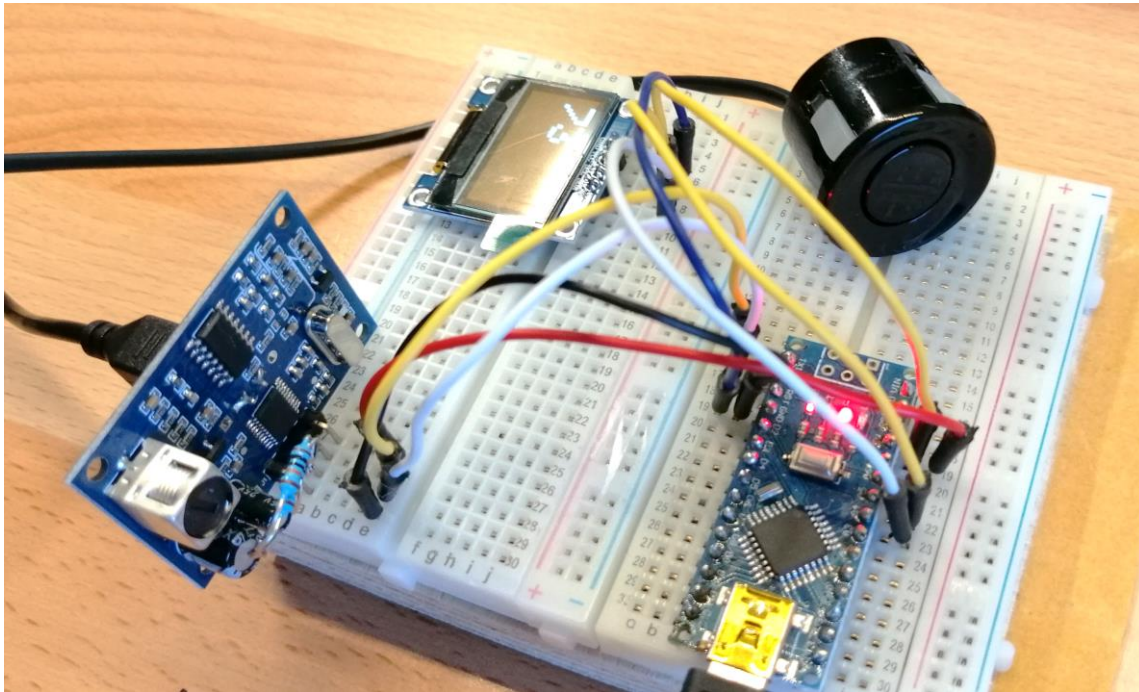


Tiny Echo Abstandsmessung mit Ultraschall

Das System besteht aus einem Arduino Nano Microcomputer, einen Ultraschall Echo Sensor und einem Displaymodul, nebst Software.

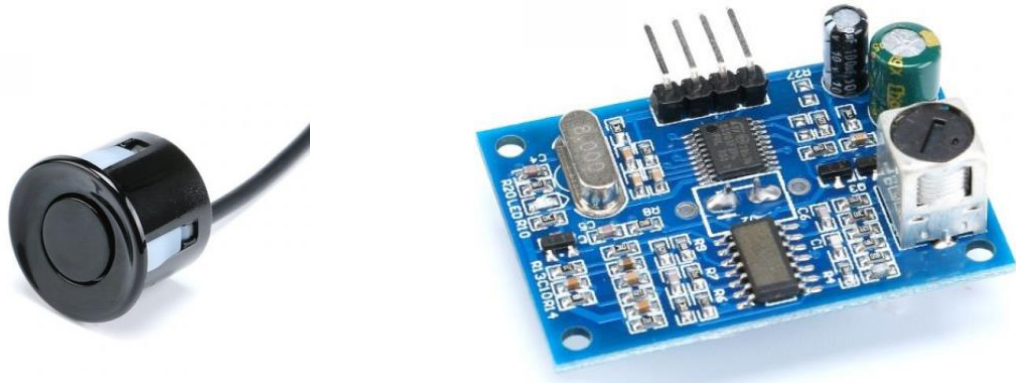


Schaltbild

Funktionsweise

Abstandsmessung

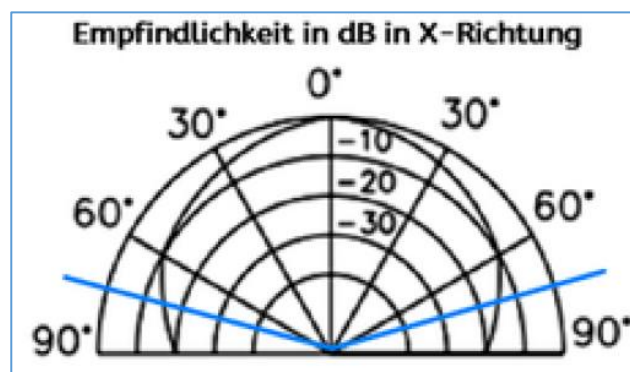
Es ist ein spezieller Ultraschallsensor nebst Elektronik verbaut.



Ultraschall Echo Modul mit Sensor

Mit zugehöriger Elektronik, ist das Echo-Erfassungsmodul JSN-SR04T/M (alternativ SKU: SEN0208) für die Messung geeignet. Es kann im Handel (Stand 2024) preiswert erworben werden.

Das Echo wird mit einem gepulsten 40 KHz Signal erzeugt. Die Sensorreichweite beträgt ca. 6 Meter. Der Erfassungswinkel beträgt 75 Grad. Der Sensor muss im Betrieb mit der Markierung „Up“, nach Oben, ausgerichtet sein.



Sensor Empfindlichkeit Radius

Im freien ist der Sensor nur bedingt einsetzbar, da Ultraschall eine mechanische Welle ist und von der umgebenen Luft abgelenkt wird. Das führt im Außenbereich zu erheblichen Messfehlern bei Wind und größeren Entfernungen.

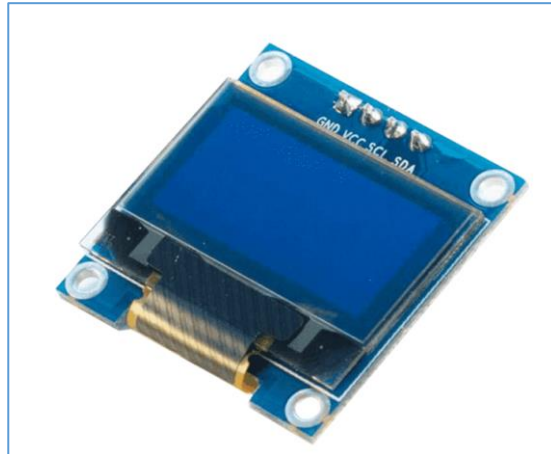
Darüber hinaus hat der Schallgeber vermutlich nur eine geringe Leistung (Schalldruck), was im Außenbereich bei größeren Entfernungen ebenfalls zu Problemen führt.

Im Innenbereich gibt es oft Reflektionen (enge Räume). Um eine gezielte Messung durchzuführen, benutze ich einen Trichter aus Pappe. Der ist um den Sensor gewickelt (Video Tutorial).

Display

Das Display ist ein kleines 0,96" SSD1306 OLED Modul, Einfarbig.

Es wird mit dem I2C-Kommunikationsprotokoll betrieben. Dafür muss die entsprechende Bibliothek in die Arduino Entwicklungsumgebung geladen sein.



Display

Software

Ich benutze die Sensorelektronik mit der Betriebsart 4. Der Sensor Prozessor übernimmt dann autonom Messung und Berechnung und übergibt das Ergebnis der über die serielle Schnittstelle an den Arduino Nano.

Außerdem ist der Sensor Stromverbrauch niedriger als in der Betriebsart 1 im Trigger/Echo Betrieb des Sensormoduls SR04T/M.

Für die Betriebsart 4 muss ein 47 KOhm Widerstand an J1 eingelötet sein! Dann wird Tx vom Sensorbord mit Rx von Nano und Rx vom Sensorbord mit Tx vom Nano verbunden.

Im zugehörigen Video Tutorial können Sie die Funktionsweise sehen. Ich demonstriere hier auch, wie Wind (simuliert mit einem Harrföhn) bei der Abstandsmessung in größerer Entfernung zum Objekt, die Messung beeinflusst.

Fazit

Für Projekte zur Messung mit kurzen Abstand zum Objekt, z.B. 2 Meter an Luft, eignet sich das preiswerte Sensormodul gut. Im Außenbereich aber eher weniger.

Der Sensor ist wasserdicht und kann somit auch unter Wasser eingesetzt werden. Ich habe dazu aber keine Experimente durchgeführt.

Das Programm ist hier im Folgenden gelistet:

Der Scratch

```
// -----  
// Abstandsmessung mit Nano (c)2024 Peter Klingbeil (film-werk56.de) und Display V1.0  
// Quellenangaben:  
// https://wiki.dfrobot.com/Weatherproof_Ultrasonic_Sensor_With_Separate_Probe_SKU_SEN0208  
// https://circuitdigest.com/microcontroller-projects/interfacing-ultrasonic-sensor-module-  
with-arduino  
// https://gitlab.com/devgiants/embedded/arduino/libraries/jsn-sr-04t  
  
#include <Arduino.h>  
#include <Wire.h>  
#include <Adafruit_GFX.h>  
#include <Adafruit_SSD1306.h>  
  
// Echo Abstandsmessung mit AJ-SR04T/M  
//  
// Seriel Mode 4: 47 kOhm Widerstand an R27 (J1)  
// Vor dem Hochladen vom Sketch, den Sensor vom Nano trennen!  
  
#include <SoftwareSerial.h>  
  
#define rxPin 0 // Arduino Nano!  
#define txPin 1 // Arduino Nano!  
  
SoftwareSerial SR04_Serial(rxPin, txPin);  
  
#define SCREEN_WIDTH 128 // OLED display width, in pixels  
#define SCREEN_HEIGHT 64 // OLED display height, in pixels  
#define OLED_RESET 4 // Reset pin # (or -1 if sharing Arduino reset pin)  
#define SCREEN_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for 128x64, 0x3C for  
128x32  
  
unsigned int cmDistanceCurrent; // Wert Messung  
  
Adafruit_SSD1306 display = Adafruit_SSD1306(128, 64, &Wire);  
  
unsigned int getDistance()  
{  
    unsigned int distance;  
    byte startByte, h_data, l_data, sum = 0;  
    byte buf[3];  
  
    startByte = (byte)SR04_Serial.read();  
    if(startByte == 255)  
    {  
        SR04_Serial.readBytes(buf, 3);  
  
        h_data = buf[0];  
        l_data = buf[1];  
        sum = buf[2];  
        distance = (h_data<<8) + l_data;  
  
        if(((startByte + h_data + l_data)&0xFF) != sum)  
        {  
            Serial.println("Invalid result");  
        }  
        else  
        {  
            distance = distance / 10; // cm Umrechnubg  
            //Serial.print("Distance cm : ");  
            //Serial.println(distance);  
        }  
    }  
}
```

```

        return distance;
    }
}
return 0;
}

void setup()
{
    Serial.begin(9600);        // Open serial monitor at 9600 baud to see ping results.
    SR04_Serial.begin(9600); // Echo Sensor
    Wire.begin();             // I2C

    // Send a message to the display
    display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // Address 0x3C for 128x64
    display.clearDisplay();

    // Clear the buffer.
    display.clearDisplay();
    display.display();

    display.setTextSize(2);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0,20);
    display.print(" Echo V 1\n");
    display.display(); // actually display all of the above

    Serial.println("Echo Version 1.0");
    delay(1500); // Version anzeigen
}

void loop()
{
    // Echo Messung
    delay(250); // Pause zwischen den Messungen!!

    if(SR04_Serial.available())
    {
        cmDistanceCurrent = getDistance();

        // Anzeige Abstand
        display.clearDisplay();
        display.setTextSize(4);
        display.setCursor(8,0);

        Serial.print("Abstand: ");
        if (cmDistanceCurrent > 600)
        {
            Serial.println("??");
            display.print("- - -");
        }
        else
        {
            Serial.println(cmDistanceCurrent);
            display.print(cmDistanceCurrent);
        }
    }

    // Display

    display.display(); // Refresh Display
}

```