KY-050 Ultrasonic-distance-sensor

Contents

1 Picture 1
2 Technical data / Short description 1
3 Pinout 1
4 Function principle 1
5 Code example Arduino 3
6 Code example Raspberry Pi 4

Picture

Technical data / Short description

If the trigger input gets a signal, the distance measurement will start and the result will be sended as a PWM-TTL signal via echo-output.

measureable distance: 2cm—300cm **measurement resolution**: 3mm

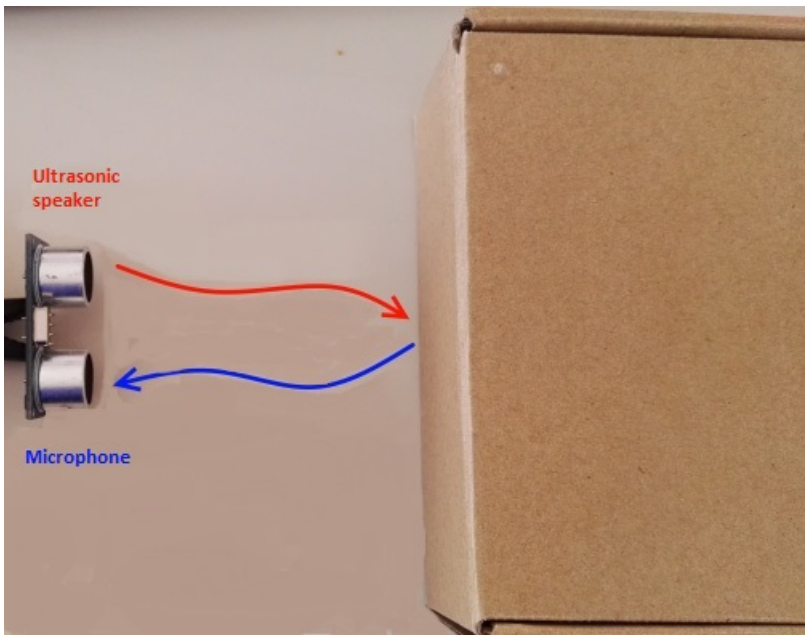
min. time between the measurements 50µs

Pinout

Function principle

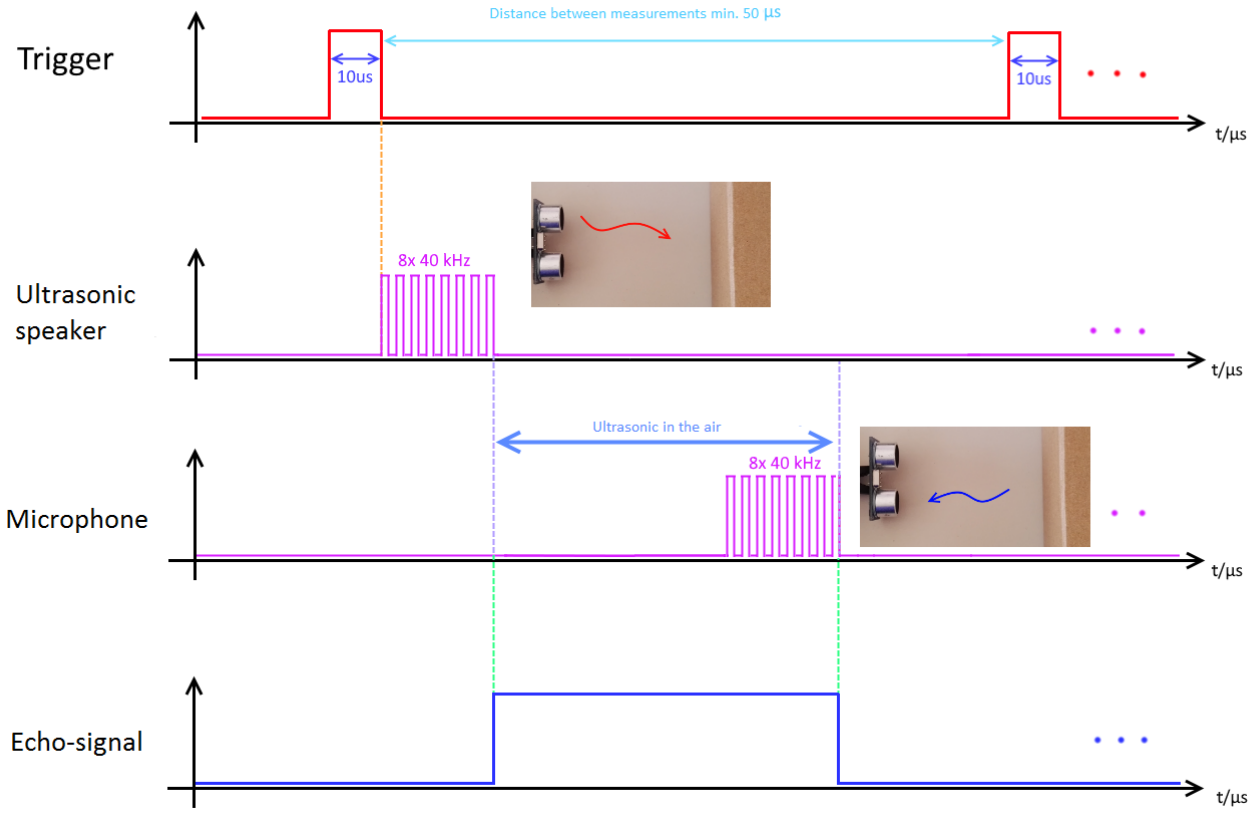
This modules shows how to measure a distance to an object with an ultrasonicspeaker and a microphone. The principle is based on the theory that the speed of the sonic, at a stable temperature, is constant - at 20° C it's 343,2m/s.

With this fact you can make the distance measurement to a time measurement, which can be easiely taken from microcontrollers.



The Ultrasonicspeaker sends 8x 40KHz signals, which will be reflected by objects and recorded with the microphone, in this sensor module. Ultrasonic is being used because you are not able to hear it with the human sense of hearing (Human sense of hearing is nearly 20HZ - 22.000Hz).

The ultrasonic signal output will start after the "trigger input pin" gets a start signal with the lenght of 10µs (active high). Activating the signal (active high) at the "echo output signal pin" after outputting. If you now record the reflected signal with the microphone, the echo signal will deactivated after the detection of the echo-signal. You can measure the time between activating and deactivating of the echo signal and calculate the distance with it.



Code example Arduino

The example program activates the distance measurement and measures the time of the ultrasonic signal. This time will be taken as a base to calculate the distance - after that the result will be printed to the serial output. If the signal is not in the measurement range, an error message will be printed..

```
#define Echo_EingangsPin 7 // Echo input-pin
#define Trigger_AusgangsPin 8 // Trigger output-pin

// Needed variables will be defined and initialized
int maximumRange = 300;
int minimumRange = 2;
long Abstand;
long Dauer;

void setup() {
 pinMode(Trigger_AusgangsPin, OUTPUT);
 pinMode(Echo_EingangsPin, INPUT);
 Serial.begin (9600);
}

void loop() {

 // Distance measurement will be started with a 10us long trigger signal
 digitalWrite(Trigger_AusgangsPin, HIGH);
 delayMicroseconds(10);
 digitalWrite(Trigger_AusgangsPin, LOW);

 // Now it will be waited at the echo input till the signal was activated
 // and after that the time will be measured how long it is active
```

```
  Dauer = pulseIn(Echo_EingangsPin, HIGH);

  // Now the distance will be calculated with the recorded time
  Abstand = Dauer/58.2;

  // Check if the measured value is in the permitted range
  if (Abstand >= maximumRange || Abstand <= minimumRange)
  {
     // An error message will be shown if it's not
       Serial.println("Distance is not in the permitted range");
       Serial.println("------------------------------------");
  }

  else
  {
     // The calculated distance will be shown at the serial output
       Serial.print("The distance is: ");
       Serial.print(Abstand);
       Serial.println("cm");
       Serial.println("------------------------------------");
  }
   // Break between single measurements
  delay(500);
 }
```

Connections Arduino:

VCC	=	[Pin 5V]
Echo	=	[Pin 7]
Trigger	=	[Pin 8]
Sensor GND	=	[Pin GND]

Example program download

KY-050_ultrasonic-distance

Code example Raspberry Pi

!! Attention !! 5V Voltagelevel !! Attention !!

The Raspberry Pi works with its ARM-CPU-core with a voltage level of 3,3V instead of 5V, like the Arduino. This sensor needs a higher voltage level to work. If you use this sensor without restriction you may damage the input pins of your Raspberry Pi permanently. But this sensor-kit also comes with a voltage-translater (KY-051) which reduces the voltage level and saves your Raspberry Pi from permantent damage. It needs to be connected between the sensor and the Raspberry Pi.

You can find more informations to that here KY-051 Voltage Translator / Level Shifter

!! Attention !! 5V Voltagelevel !! Attention !!

The example program activates the distance measurement and measures the time of the ultrasonic signal. This time will be taken as a base to calculate the distance. The result will be given via the serial output. If the signal is not in the measurement range, an error message will be printed to the serial output.

```
# coding=utf-8
# Needed modules will be imported and configured
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)

# You can pick the input and output pins here
Trigger_AusgangsPin = 17
Echo_EingangsPin    = 27

# You can set the delay (in seconds) between the single measurements here
sleeptime = 0.8

# Here, the input and output pins will be configured
GPIO.setup(Trigger_AusgangsPin, GPIO.OUT)
GPIO.setup(Echo_EingangsPin, GPIO.IN)
GPIO.output(Trigger_AusgangsPin, False)

# Main program loop
try:
    while True:
        # Distance measurement will be started with a 10us long trigger signal
        GPIO.output(Trigger_AusgangsPin, True)
        time.sleep(0.00001)
        GPIO.output(Trigger_AusgangsPin, False)

        # The stop watch will start here
        EinschaltZeit = time.time()
        while GPIO.input(Echo_EingangsPin) == 0:
            EinschaltZeit = time.time() # The time will be saved till the signal is active

        while GPIO.input(Echo_EingangsPin) == 1:
            AusschaltZeit = time.time() # The last time will be recorded in which the sign

        # The difference between the times gives the searched duration
        Dauer = AusschaltZeit - EinschaltZeit
        # With it you can calculate the distance
        Abstand = (Dauer * 34300) / 2

        # Here you check if the measured value is in the permitted range
        if Abstand < 2 or (round(Abstand) > 300):
            # If not an error message will be shown
            print("Distance is not in the permitted range")
            print("------------------------------")
        else:
            # The value of the distance will be reduced to 2 numbers behind the comma
            Abstand = format((Dauer * 34300) / 2, '.2f')
            # The calculated distance will be shown at the terminal
            print("The distance is:"), Abstand,("cm")
            print("------------------------------")

        # Break between the single measurements
        time.sleep(sleeptime)

# Scavenging work after the end of the program
except KeyboardInterrupt:
    GPIO.cleanup()
```

Connections Raspberry Pi:

Sensor KY-050:

VCC = 5V [Pin 2 (RPi)]

Trigger	=	Pin B1	[KY-051-Voltage Translator]
Echo	=	Pin B2	[KY-051-Voltage Translator]
GND	=	GND	[Pin 6 (RPi)]

KY-051- Voltage Translator:

VCCb	=	5V	[Pin 04(RPi)]
Pin B1	=	Trigger	[KY-050-UltrasonicSensor]
Pin B2	=	Echo	[KY-050-UltrasonicSensor]
VCCa	=	3,3V	[Pin 01(RPi)]
Pin A1	=	GPIO17	[Pin 11(RPi)]
Pin A2	=	GPIO27	[Pin 13(RPi)]
GND	=	GND	[Pin 06(RPi)]

- You must not connect the other pins of the KY-051-Voltage-Translator-Modul (OE,B3,B4,A3,A4).

Example program downlaod

KY-50_ultrasonic-distance_RPi

To start, enter the command:

```
sudo python KY-50_ultrasonic-distance_RPi.py
```