

How does the ultrasonic component work

De Borrekens Gauthier

Intro

In this theoretical study I will try to understand how the hc-sr04 ultrasonic ranging component works. Then I will analyse some programming examples. Finally I will make my own program and test this out in the lab as I lack the required equipment to test at home

Materials and Methods

The references listed at the end.

I also borrowed the breadboard with the nodemcu, the hc-sr04 and some cables but as I lack the necessary cables I cannot test at home.

Results

As this is a preparation and the validity of everything has to be tested in the lab, it does not have any results.

Information & conclusion

Analog part

The HC-SR04 has two ultrasonic components. One transmits a wave, the other one receives a wave. The module has four pins:

- Ground: self-explanatory. Has to be connected to one of the grounds of the NodeMCU.
- Power: A 5V power source, This is inconvenient as the NodeMCU Vout is 3.3V, Maybe we can drain power from the same batteries that connect the NodeMCU but this will get tested in the labo.
- Trig: A pulse input. When this receives a pulse of at least 10 microseconds, it will start transmitting 8 40KHz waves.
- Echo: An output, this will send back a pulse when the wave is picked up by the receiving component. The pulse will be the same length as the duration of the arrival of the sent pulse, transmit of wave and picking up of wave.

(SparkFun, 2018)

Calculations

Distance = (time between sending pulse and receiving pulse)/58 * 340m/s / 2.

This is what is written on the official documentation but will need testing to see the legitimacy.

(SparkFun, 2018)

Coding example analysis

The code example uses an Arduino board and some leds instead of printing out the results on the serial connection, but should be comparable to our setup.

First he defines some names to the pins. This is optional and not important to analyse as we can choose this ourselves/

In the setup() part, he begins a serial connection with 9600 baud, we will use 115200 baud. Then he uses the 'pinmode' command to set the previously defined pins to in or output. This is all very simple and easy to follow.

Next comes the loop(). Here he puts a 'high' on the pin connected to the trig with the digitalWrite() command and waits 10 microseconds before reverting it to 'low'. Then checks if the echo pin is high with the PulseIn() command and stores that value

in a long called 'duration'. This is also very easy to follow.

Now comes the only part i did not understand. To get the distance, he takes the duration, divides it by 2 and then divides it by 29.4. I have no idea where this value comes from. In the lab we will test this and the aforementioned values and see which one to use. Further documentation on this will follow.

He then follows by turning on the leds according to the distance, but as we won't be using leds, we will write this part from scratch. (jsvester, 2018)

Some commands to clarify;

`pinMode(pinNumber, Mode)`: Will set the pin defined by `pinNumber` to one of these options: INPUT, OUTPUT, INPUT_PULLUP, INPUT_PULLDOWN

`digitalWrite(pinNumber, Value)`: will set the value of the pin defined by `pinNumber` to one of these values: HIGH, LOW.

`pulseIn(pin, value, timeout)`: Will sense a pin defined by `pinNumber` until a value (HIGH or LOW) gets set. Optional timeout. (Arduino, 2018)

Own code

Based upon that example, I will write a small program which has to be tested in the lab.

Referencelist

Arduino. (2018, Maart 24). *Language reference*. Opgehaald van Arduino: <https://www.arduino.cc/reference/en/>

jsvester. (2018, Maart 24). *SIMPLE ARDUINO AND HC-SR04 EXAMPLE*. Opgehaald van instructables: <http://www.instructables.com/id/Simple-Arduino-and-HC-SR04-Example/>

SparkFun. (2018, Maart 24). *Ultrasonic Ranging Module HC - SR04*. Opgehaald van sparkFun: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>