

INF208 IoT Laborprojekt

Ein durch Annäherung aktiviertes Lichtsystem

Burak Polat (180503033)

Bariş Köse (190503059)

Vorlesung:

INF208: Eingebettete Systeme

Dozent:

- Prof. Dr. Faruk Bağcı

Laborassistenten:

- Onur Akgün
- Ebru Subutay
- Ferit Tiryaki

Was dieses Projekt zu lösen beabsichtigt

Der Dauerbetrieb der Leuchte über den ganzen Tag oder in regelmäßigen Abständen verkürzt die Lebensdauer der Leuchte. Außerdem wird dadurch Energie verschwendet. Dieses Projekt zielt darauf ab, die Effizienz von Beleuchtungssystemen zu erhöhen. Auf diese Weise wird Strom gespart und die Lebensdauer der verwendeten Beleuchtungsanlagen verlängert.

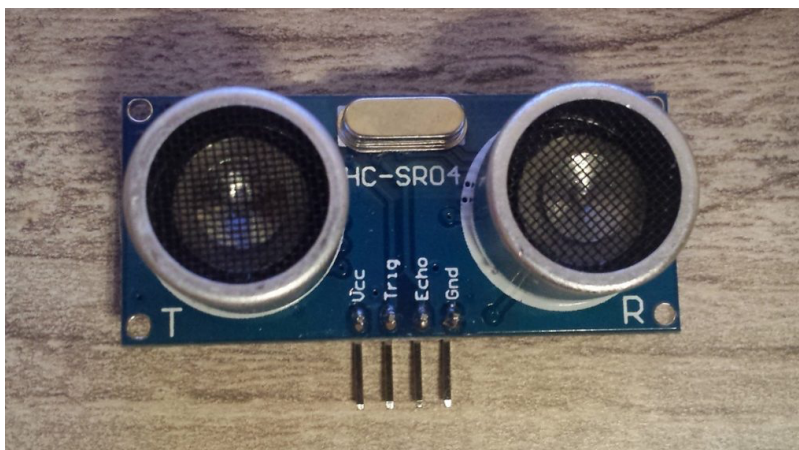
Die Komponenten dieses Projekts

- Raspberry Pi 4
- Ein Ultraschallsensor (HC-SR04)
- Die Widerstände: 330Ω und 470Ω
- Jumper Kabeln
- Eine 5mm rote LED-Ampel
- Steckbrett

Die Lösungsmethode

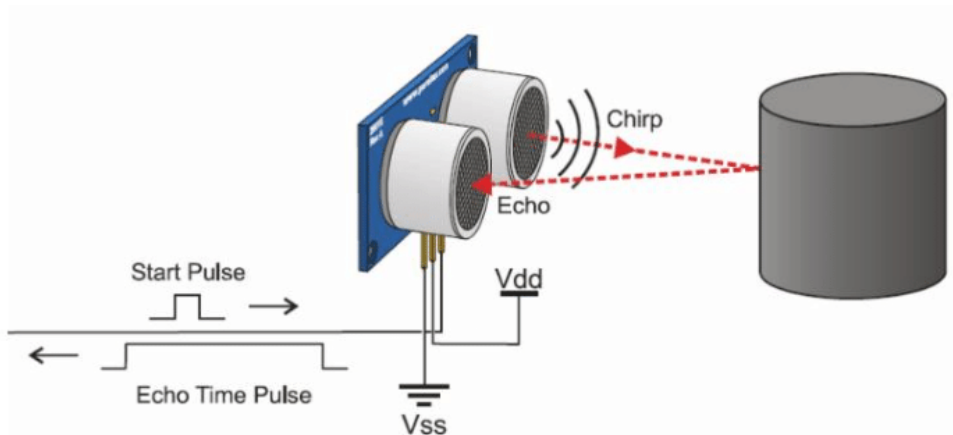
Das IoT-System soll sicherstellen, dass das Licht nur dann funktioniert, wenn die vom Sensor gemessene Entfernung unter einem bestimmten Wert liegt. Dies kann durch die Verwendung eines Entfernungssensors erreicht werden, der die Entfernung zwischen dem Objekt und sich selbst auf der Grundlage der zuvor bekannten Geschwindigkeit der Ultraschallwelle berechnet.

Die Funktionsweise des Sensors



Der Sender im Ultraschallsensor erzeugt einen 40 KHz-Ultraschall. Der TRIG-Pin des Ultraschallsensors muss HIGH sein, um ein Signal für eine Mindestdauer von $10\mu\text{s}$ zu senden, da $10\mu\text{s}$ die Länge der Ultraschallwelle ist. Sobald das Signal den Sender verlässt, breitet es sich in

der Luft aus, und wenn sich ein Hindernis in seinem Weg befindet, trifft das Signal auf das Hindernis und prallt zurück. Dieses zurückgeworfene Signal wird dann vom Ultraschallempfänger empfangen. Anhand der Zeit, die das Signal für die Übertragung, das Auftreffen auf das Hindernis und den Rückempfang am Empfänger benötigt, können Sie die Entfernung des Objekts berechnen, da die Schallgeschwindigkeit bereits bekannt ist.¹



Wir werden im nächsten Schritt sehen, wie man die Entfernung eines Objekts mit dem HC-SR04 Ultraschallsensor messen kann. Wenn zum Beispiel der Sender zum Zeitpunkt T1 ein Signal aussendet und das Signal aus dem Sender aus, trifft auf das Objekt und wird zum Zeitpunkt T2 wieder empfangen.

Zeit, die das Signal für die Übertragung und den Rückempfang benötigt:

$$(T) = T2 - T1$$

Zeitaufwand in einer Richtung = $T/2$

Wenn wir davon ausgehen, dass die Geschwindigkeit der Ultraschallwellen in der Luft S/sec beträgt, können wir die Entfernung berechnen, da wir die Zeit kennen, die das Signal benötigt, um auf das Objekt zu treffen, und die Schallgeschwindigkeit.

Entfernung = $T/2 \times S$

¹ [Raspberry Pi Project - Ultrasonic Distance Sensor In Civil Engineering \(Prakash Kanade, 2020\)](#)

Softwaretechnische Seite des Projekts

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <wiringPi.h>
4
5 #define TRUE 1
6
7 #define TRIG 4
8 #define ECHO 5
9 #define LedPin 0
10
11 void setup() {
12     wiringPiSetup();
13     pinMode(TRIG, OUTPUT);
14     pinMode(ECHO, INPUT);
15     pinMode(LedPin, OUTPUT);
16
17     //TRIG pin must start LOW
18     digitalWrite(TRIG, LOW);
19     delay(30);
20 }
21
22 int getCM() {
23     //Send trig pulse
24     digitalWrite(TRIG, HIGH);
25     delayMicroseconds(20);
26     digitalWrite(TRIG, LOW);
27
28     //Wait for echo start
29     while(digitalRead(ECHO) == LOW);
30
31     //Wait for echo end
32     long startTime = micros();
33     while(digitalRead(ECHO) == HIGH);
34     long travelTime = micros() - startTime;
35
36     //Get distance in cm
37     int distance = travelTime / 58;
38
39     return distance;
40 }
41
42 int main(void) {
43     setup();
44
45     while(TRUE){
46         delay(300);
47         printf("Distance: %dcm\n", getCM());
48         if (getCM() > 20)
49             digitalWrite(LedPin, LOW);
50         else{
51             digitalWrite(LedPin, HIGH);
52             delay(10000);
53         }
54     }
55
56     return 0;
57 }
58 }
```

Die Funktion *setup()* trägt dazu bei, die Pins des Sensors und der LED zu definieren. Sie setzt auch den Pegel des Triggers zu Beginn, vor den Berechnungen, auf niedrige Spannung (*LOW*). Die Funktion *getCM()* setzt den Trigger auf HIGH und beginnt damit die Berechnung der Entfernung. Nachdem der Impuls gesendet wurde, wird ECHO aktiviert (startZeit = *LOW*, endZeit = *HIGH*), um das von TRIG gesendete Signal zu empfangen. Nachdem das Signal erfasst wurde, wird die Laufzeit berechnet. Und es gibt die Entfernung, wenn sie durch 58 (Geschwindigkeit der Ultraschallwelle) geteilt wird. Und schließlich ruft die Hauptfunktion die zuvor deklarierten Funktionen auf und fügt zwischen den Berechnungen eine kleine delay hinzu, um die Möglichkeit eines Absturzes zu minimieren.