

Aufsetzen eines NodeMCU8266 zum Auslesen sensorspezifischer Daten

Hello World

für die

Prüfung zum Bachelor of Science

des Studiengangs IT-Automotive

an der Dualen Hochschule Baden-Württemberg Stuttgart

von

Jonas Lieben

September 2024

Bearbeitungszeitraum
Matrikelnummer, Kurs
Ausbildungsfirma
Betreuer
Gutachter

01.10.2024 - 15.12.2024
8761357, TINFXX
Mercedes-Benz AG, Stuttgart
Betreuer

Erklärung

Ich versichere hiermit, dass ich meine Hello World mit dem Thema: *Aufsetzen eines NodeMCU8266 zum Auslesen sensorspezifischer Daten* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Stuttgart, September 2024

Jonas Lieben

Zusammenfassung

Deutscher Abstract

Inhaltsverzeichnis

| | |
|--|-------------|
| Abkürzungsverzeichnis | V |
| Abbildungsverzeichnis | VI |
| Tabellenverzeichnis | VII |
| Listings | VIII |
| 1 Einleitung | 1 |
| 1.1 Thema | 1 |
| 1.2 Fragestellung | 1 |
| 1.3 Zielsetzung | 1 |
| 1.4 Aufbau | 1 |
| 1.5 Probleme (gehört hier nicht her) | 2 |
| 2 Stand der Technik | 3 |
| 2.1 Mikrocontroller | 3 |
| 2.1.1 ESP8266 | 3 |
| 2.2 Abstandssensoren | 3 |
| 2.2.1 Der HC-SR04 Ultraschallsensor | 4 |
| 2.2.2 Ungenauigkeit und Messfehler | 6 |
| 2.2.3 Interpretation von Messdaten | 6 |
| 2.3 Anwendung von Abstandssensoren in der Automobilindustrie (Quelle: https://www.bmw.com/de/innovation/sensoren-im-auto.html) | 7 |
| 2.3.1 Parkdistanzkontrolle (PDC) | 7 |
| 2.3.2 Automatische Einparkhilfe | 7 |
| 2.3.3 Autonomes Fahren | 7 |
| 2.4 Übertragung von Daten mittels eines Webservers | 7 |
| 2.4.1 OSI-Schichtenmodell | 8 |
| 2.4.2 HTTP | 8 |
| 3 Ausarbeitung von Lösungen | 10 |
| 3.1 Aufbau des Projektes | 10 |
| 3.1.1 Physikalische Verbindung zwischen Sensor und Arduino herstellen (Schaltkreis aufbauen) | 10 |
| 3.1.2 Programmierung zum Ermitteln und Bereitstellen der Sensordaten . | 11 |
| 3.1.3 Programmierung zum Anzeigen der bereitgestellten Sensordaten . . | 11 |
| 4 Zusammenfassung und Ausblick | 13 |
| 4.1 Vorgehen | 13 |

| | | |
|----------|----------------------|-----------|
| 4.2 | Ergebnisse | 13 |
| 4.3 | Ausblick | 13 |
| 5 | EMPTY | 14 |
| 6 | EMPTY | 15 |
| 7 | EMPTY | 16 |
| | Literatur | 17 |

Abkürzungsverzeichnis

| | |
|--------------|--|
| ISO | International Standardization Organisation |
| OSI | Open Systems Interconnect |
| HTML | Hypertext Markup Language |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |
| LAN | Local Area Network |
| WLAN | Wireless Local Area Network (LAN) |

Abbildungsverzeichnis

| | | |
|-----|--|----|
| 2.1 | Ultraschallwellen einer Fledermaus | 4 |
| 2.2 | HC-SR04 Sensor Vorderseite | 5 |
| 3.1 | Entwickelter Schaltkreis | 10 |
| 6.1 | 3 Images | 15 |
| 6.2 | Bild mit Quelle | 15 |
| 6.3 | einfaches Bild | 15 |

Tabellenverzeichnis

| | | |
|-----|---|---|
| 2.1 | Open Systems Interconnect (OSI)-Schichten | 8 |
|-----|---|---|

Listings

1 Einleitung

In modernen Fahrzeugen spielen Sensordaten eine zentrale Rolle. Egal ob bei der Steigerung der Effizienz, bei der Verringerung von Risiken, oder bei der Verbesserung des Fahrerlebnisses: Sensordaten sind ein zentraler Teil des modernen Automobils.

Schon 1982 wurde im Toyota Corona die Erste auf Sensordaten basierende, aktive Einparkhilfe verbaut (vgl. [Online.2022], S.22-28). Beispielsweise wurde im Toyota Corona bereits 1982 die Erste auf Sensordaten basierende, aktive Einparkhilfe verbaut. Diese verwendete Ultraschallsensoren, um die Distanz zum nächsten Objekt nach Hinten zu messen und den Fahrer wie heute fast selbstverständlich akustisch auf den Abstand hinzuweisen.

1.1 Thema

TODO

1.2 Fragestellung

TODO

1.3 Zielsetzung

Zielsetzung des Hello-Word Projektes ist die Nachstellung Gewinnung von Sensordaten über einen ESP8266. Zudem soll der ESP8266 sich mit einem lokalen WLAN verbinden und als Webserver fungieren, welchen man zum Erhalt der Daten ansprechen kann.

1.4 Aufbau

Aufgabe des Hello World-Projektes ist es, mithilfe eines ESP8266 ein System zu entwickeln, welches Sensordaten in einem Mercedes-Benz Fahrzeug erfasst und über einen Webserver im Rahmen einer Website als Live-Daten zur Verfügung stellt. Dieses Projekt dient später als Grundlage für die in der Uni abzugebende T1000-Dokumentation.

1.5 Probleme (gehört hier nicht her)

- Wie schafft man es, dass die Daten aktuell auf einer Website angezeigt werden

Übergabe von HTML, JS und CSS, wobei JS-schicht die Daten live abfragt und auf der Seite visualisiert

- Treiber-Installation auf Firmenrechner unmöglich -> Eigener Rechner benötigt
- Wie verwende ich den Sensor?

2 Stand der Technik

2.1 Mikrocontroller

Unter einem Mikrocontroller versteht man einen kleinen, eigenständigen Computer. Sie bestehen aus einem Prozessor, Speicher und Ein-/Ausgabefunktionen und sind speziell für die Steuerung von elektronischen Geräten und Systemen entwickelt (vgl. [2], S.2).

2.1.1 ESP8266

Projektvorgabe war die Arbeit mit einem sogenannten ESP8266. Das ist ein spezieller Mikrocontroller, welcher sich besonders durch eine erhöhte Rechenleistung und einen integrierten Wireless LAN (WLAN)-Chip für einen vergleichsweise geringen Preis auszeichnet.

2.2 Abstandssensoren

Abstandssensoren sind Sensoren, welche zur Ermittlung der Distanz zwischen zwei Punkten verwendet werden. Im Laufe der Zeit wurden verschiedenste Sensoren entwickelt, um das zu ermöglichen. Zu den bekanntesten Abstandssensoren gehören beispielsweise der Ultraschallsensor, sowie auch der Infrarotsensor. Der Grundgedanke hinter der Abstandsermittlung mit einem dieser beiden Sensoren wurde von der Fledermaus adaptiert. Diese sendet Schallwellen im für Menschen nicht hörbaren Bereich aus. Kommen die Schallwellen zurück, so ist in der entsprechenden Richtung ein Gegenstand. So kann die Fledermaus ihre Beute und Hindernisse in der Flugbahn selbst bei absoluter Dunkelheit ausfindig machen.

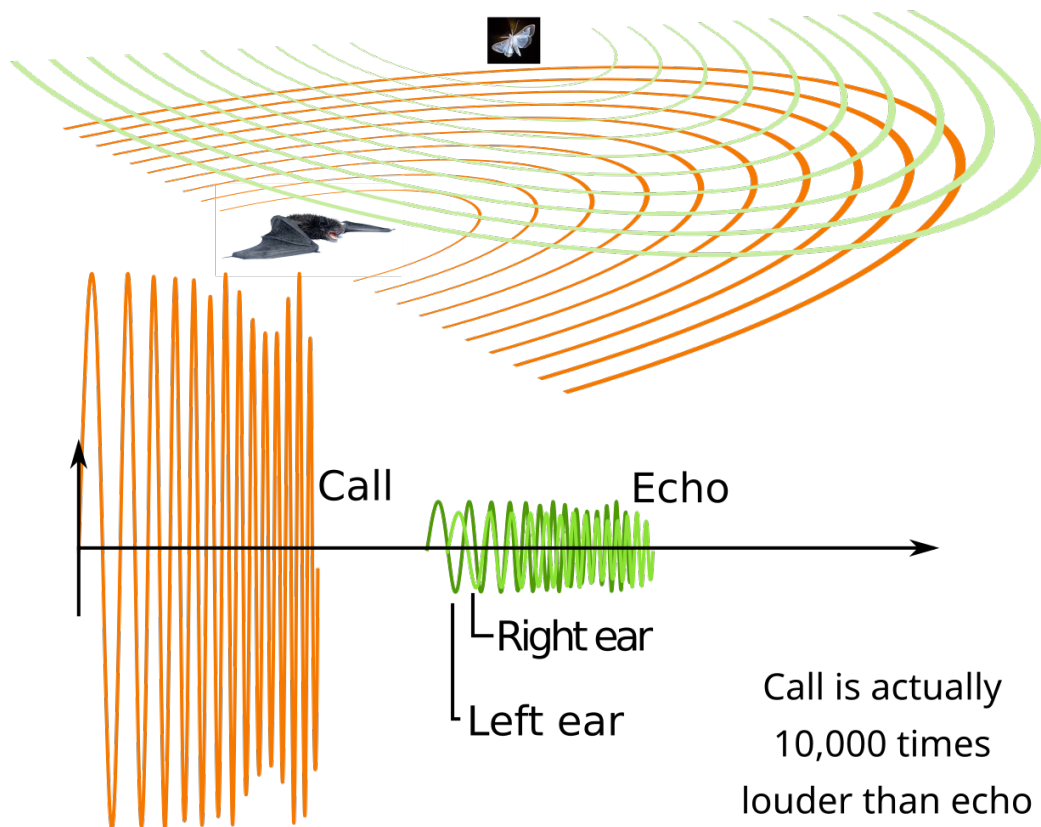


Abbildung 2.1: Animal Echolocation, (vgl. [1])

Wie auch bei der Fledermaus senden die beiden Sensortypen ein Signal aus und messen dessen Rückkehr nach der Kollision mit einem Gegenstand. Wenn die Ausbreitungsgeschwindigkeit des entsprechenden Signals bekannt ist, so kann die zurückgelegte Distanz und somit auch die Entfernung zum Gegenstand ermittelt werden. Unterscheiden tun sich die Sensortypen lediglich in der Art des ausgesendeten Signals. Der Ultraschallsensor sendet, wie der Name schon sagt, Ultraschallwellen aus (vgl. [5], S.126). Der Infrarotsensor entsprechend Infrarotwellen (vgl. [5], S.127).

2.2.1 Der HC-SR04 Ultraschallsensor

Im Rahmen dieses Projektes wurde für die Abstandsmessung ein vergleichsweise günstiger Entfernungssensor verwendet. Dieser hat eine Breite von 4,5 cm und ohne einberechnung der Pins eine Höhe von 2 cm.

Bei dem Sensor handelt es sich um einen auf Ultraschalltechnologie basierenden Distanzsensor. Er enthält einen Ultraschallsender sowie einen Ultraschallempfänger und wird mit einer Spannung von 5V betrieben. Im Rahmen einer einzelnen Messung sendet der Ultraschallsender Ultraschallsignale aus. Der Ultraschallempfänger empfängt diese Signale

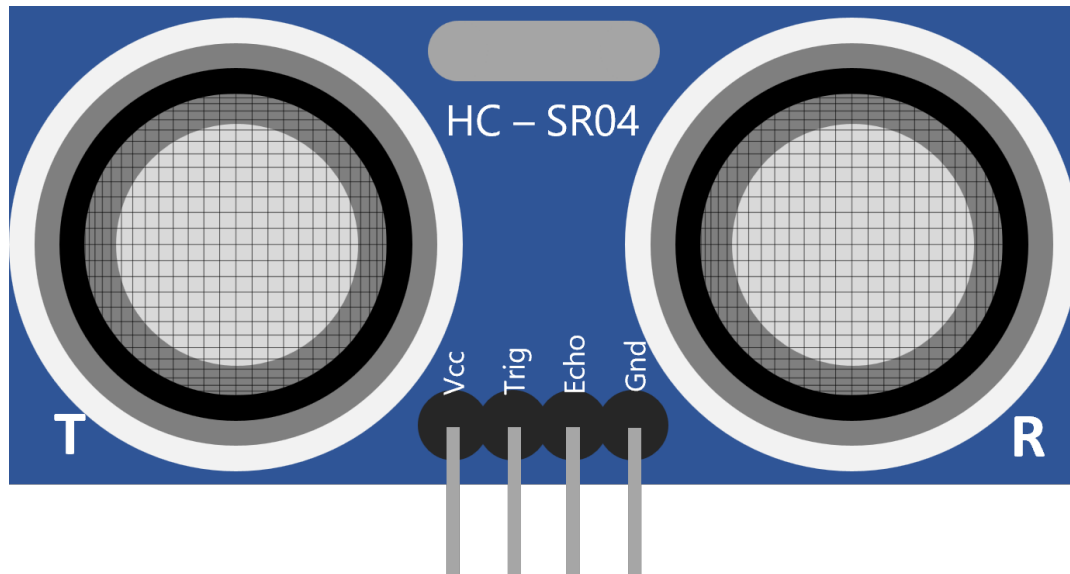


Abbildung 2.2: HC-SR04 Sensor, Vorderseite, eigene Darstellung (vgl. [3])

und wandelt sie in ein elektrisches Signal um (vgl. [6]).

Wenn das im rechten Sensor erzeugte Signal gemessen wird, kann die Zeit ermittelt werden, welche zwischen Absenden des Ultraschallsignals und dessen Wiederkehr liegt. Da die Geschwindigkeit vom Ultraschall durch Luft bekannt ist, kann die zurückgelegte Distanz und dadurch ebenso die Entfernung ermittelt werden. Hierzu wird die folgende Formel verwendet:

$$L = \frac{1}{2} \cdot T \cdot C \quad (2.1)$$

L: Distanz zum nächsten Objekt

T: Gemessene Zeit

C: Schallgeschwindigkeit

Da der Sensor im Rahmen dieser Arbeit ausschließlich unter Normalbedingungen in Luft getestet wurde, kann die Schallgeschwindigkeit als näherungsweise Konstant angenommen werden. Mit diesem Wert lässt sich die Formel ergänzen:

$$L = \frac{343}{2} \cdot \frac{\text{m}}{\text{s}} \cdot T \quad (2.2)$$

L: Distanz zum nächsten Objekt

T: Gemessene Zeit

2.2.2 Ungenauigkeit und Messfehler

Bei der Verwendung von Sensoren ist immer mit einer gewissen Ungenauigkeit der gemessenen Werte zu rechnen. Diese treten unweigerlich durch sich ständig ändernde äußere Faktoren auf, können jedoch ebenso durch eine falsche Verwendung des entsprechenden Sensors hervorgerufen werden.

Äußere Faktoren sind die Erdanziehungskraft, die Temperatur, das Material in dem der Sensor verwendet wird und viele weitere Faktoren. Beispielsweise hängt die Ausbreitungsgeschwindigkeit des Schalls von der Temperatur ab. Ist es extrem kalt, so breitet sich der Schall langsamer aus, als bei hohen Temperaturen (vgl. [4], S.4-5). Viel extremere Ausmaße haben oft Fehler bei der Verwendung des Sensors. Beispielsweise ist der HC-SR04 Sensor nur für das Messen von Distanzen zwischen 2 cm und 4 m ausgelegt. Wenn also Distanzen unter zwei Zentimetern oder über vier Metern gemessen werden, dann könnte ein Fehlerwert das Ergebnis sein. Das liegt daran, dass der Sensor bei so geringen beziehungsweise hohen Distanzen das Signal nicht mehr erhält und nach einer geringen Wartezeit die Prüfung beendet, damit der Computer nicht dauerhaft auf eine Auflösung wartet. Für den HC-SR04 beläuft sich das Ergebnis einer Fehlmessung immer auf den Fehlerwert von acht Metern. In der Realität könnte ein solcher Fehlerwert natürlich weitreichende Auswirkungen haben. Ein einfaches Beispiel für die Auswirkungen einer Fehlmessung ohne Interpretation der Messwerte wäre das Parken eines Autos. Wenn sich sehr nah hinter dem Fahrzeug ein Fußgänger befindet und der Sensor aufgrund der geringen Distanz den Fehlerwert von acht Metern an den Fahrer weitergibt, so könnte dies im schlimmsten Falle tödlich für den Fußgänger enden.

2.2.3 Interpretation von Messdaten

Die Ungenauigkeit von Messergebnissen, ob durch äußere Einflüsse oder falsche Verwendung macht die reinen Messergebnisse unnutzbar. Aus diesem Grund müssen die gemessenen Daten vor der Weitergabe an einen nicht Fachkundigen richtig interpretiert werden. Wie genau die 'richtige' Interpretation der Messdaten aussieht, hängt immer am entsprechenden Anwendungsgebiet und dem verwendeten Sensor.

Für den HC-SR04 können jegliche Werte zwischen zwei Zentimetern und vier Metern als realistische Werte angesehen und weitergegeben werden. Sobald ein Wert außerhalb dieses Bereichs gemessen wird, muss von einer potentiellen Messungenauigkeit ausgegangen werden. Der Wert darf nicht an den Nutzer weitergegeben werden. Besonders interessant sind beim HC-SR04 die Werte von knapp unter bis knapp über acht Metern. Diese bedeuten nämlich in der Regel immer, dass das gesendete Signal nicht wieder aufgenommen

wurde. Wie bereits unter 2.2.2 erwähnt, kann dieser Wert also ein extrem nahes, oder ein extrem weit entferntes Objekt indizieren. Der Umgang mit diesen Werten hängt von der Anwendung des Sensors ab. Befindet sich der Sensor an einem sich Bewegenden Objekt wie beispielsweise einem Fahrzeug und die Werte werden in einer hohen Frequenz mindestens einmalig jede Sekunde abgefragt, so kann man folgende Annahme treffen:

Waren die letzten Werte sehr nah an der unteren Grenze, so besteht große Gefahr einer Kollision. Man sollte von einem Minimalwert ausgehen. Waren die letzten Werte eher in der Nähe der oberen Grenze, so ist die Wahrscheinlichkeit sehr hoch, dass die Distanz sich weiter ausgebaut hat und das nächste Objekt sich nun außerhalb der Reichweite des Sensors befindet.

Wird der Sensor jedoch in einer Art Radar an einem sich drehenden Turm verwendet, so kann aus den letzten Werten kein Schluss auf den aktuellen Wert getroffen werden.

2.3 Anwendung von Abstandssensoren in der Automobilindustrie (Quelle:

<https://www.bmw.com/de/innovation/sensoren-im-auto.html>)

2.3.1 Parkdistanzkontrolle (PDC)

2.3.2 Automatische Einparkhilfe

2.3.3 Autonomes Fahren

2.4 Übertragung von Daten mittels eines Webservers

Nachdem die Daten von einem Sensor erfasst wurden, können diese kurzzeitig ausgewertet und verwendet werden. Oftmals ist in modernen Anwendungen jedoch eine langfristige Auswertung der Daten, beispielsweise aus Versicherungsgründen, notwendig. Im Rahmen des Hello World Projektes soll die Sicherung der Daten außerhalb des ermittelnden Systems ermöglicht werden. Die Vorgabe hierzu ist, dass der genutzte ESP einen Webserver starten soll, welcher einerseits die ermittelten Daten in sinnvollem Format zur Verfügung stellt. Andererseits soll der Webserver auch eine Webseite hosten, welche Echtzeitdaten für einen Benutzer transparent macht.

2.4.1 OSI-Schichtenmodell

Das OSI-Schichtenmodell ein 1984 von der International Standardization Organisation (ISO) entwickeltes Modell zur Aufteilung des Kommunikationsprozesses zwischen Computern in sieben Schichten. Diese Schichtunterteilung erleichtert die Kommunikation zwischen verschiedenen Computern. Die Kommunikation wird so Betriebssystemunabhängig und Implementationsunabhängig gewährleistet, solange beide Geräte sich an die im OSI-Modell beschriebenen Vorgaben halten.

Schichten des OSI-Modells

Tabelle 2.1 zeigt die einzelnen Schichten des OSI-modells.

| Schicht | OSI-Schicht | OSI-Layer (en) |
|---------|-----------------------|--------------------|
| VII | Anwendungsschicht | Application Layer |
| VI | Darstellungsschicht | Presentation Layer |
| V | Kommunikationsschicht | Session Layer |
| IV | Transportschicht | Transport Layer |
| III | Vermittlungsschicht | Network Layer |
| II | Sicherungsschicht | Data Link Layer |
| I | Pysikalische Schicht | Physical Layer |

Tabelle 2.1: Testtabelle

2.4.2 HTTP

Da im Rahmen dieses Projektes mehrere Endpoints gehostet werden sollen, welche die Sensordaten bereitstellen, wird das Hypertext Transfer Protocol (HTTP)-Protokoll verwendet. Für die Kommunikation zwischen zwei PCs über das Internet wird fast ausschließlich dieses 1992 eingeführte Protokoll verwendet. HTTP standardisiert die Kommunikation zwischen den Anwendungsschichten (OSI, Tabelle 2.1) zweier Computer. Hierzu gibt es eine genaue Struktur vor, mit der Daten gesendet werden müssen. Beispielsweise muss der Typ der Anfrage als 'method' und die zu übertragenden Daten als 'body' angegeben werden. Darüber hinaus gibt es noch einige weitere Informationen, welche für die Übertragung per HTTP relevant sind.

HTTP wird beispielsweise immer verwendet, wenn Webseiten in einem Internetbrowser

geladen werden. Wenn hier eine neue Website geladen wird, dann erstellt dieser eine HTTP-Anfrage. Diese Anfrage nutzt den Anfragentyp 'GET', weil die Daten ja von diesem PC angefragt werden. Wenn der anfragende PC dann eine Antwort vom Server erhält, dann handelt es sich in der Regel um einen Hypertext Markup Language (HTML)-Text. Dieser wird vom Browser erkannt und als Website dargestellt.

Ein großer Nachteil von HTTP ist die fehlende Sicherheit der Technologie. Werden Daten nach den Standards dieses Protokolls übertragen, so sind sie nicht verschlüsselt und damit ebenso für jeden im Selben Netzwerk lesbar. Aus diesem Grund wurde das erstmals 1994 verwendete Hypertext Transfer Protocol Secure (HTTPS) zum heutigen Standard für die Übertragung von Daten über das Internet. Dieses Protokoll funktioniert exakt wie dessen Vorbild HTTP. Die einzige Anpassung ist, dass die Daten vor dem Absenden verschlüsselt und beim Empfänger anschließend wieder entschlüsselt werden. So kann jemand im Selben Netzwerk die Daten zwar erhalten und anschauen, jedoch nicht lesen. Hierzu würde er einen einzigartigen Schlüssel benötigen.

3 Ausarbeitung von Lösungen

3.1 Aufbau des Projektes

Um die Sensordaten bereitzustellen, müssen verschiedene Schritte durchlaufen werden. Zuerst muss eine physikalische Verbindung zum Sensor aufgebaut werden. //TODO

3.1.1 Physikalische Verbindung zwischen Sensor und Arduino herstellen (Schaltkreis aufbauen)

Um die Sensordaten zu ermitteln muss zuerst der Sensor mit dem ESP8266 verbunden werden. Zur Inbetriebnahme des HC-SR04 sind vier Verbindungen notwendig.

- VCC: Spannungsversorgung 5V
- GND: Masse
- Trig: Daten (in)
- Echo: Daten (out)

Um also die Sensordaten auslesen zu können, wurde der folgende Schaltkreis entwickelt:

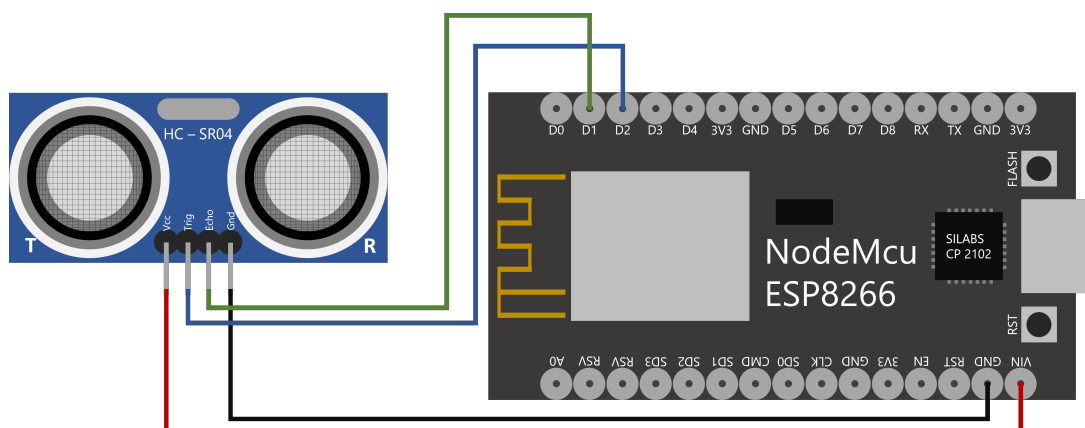


Abbildung 3.1: Entwickelter Schaltkreis, eigene Darstellung (vgl. [3])

3.1.2 Programmierung zum Ermitteln und Bereitstellen der Sensordaten

Nachdem die Physikalische Verbindung zum Sensor hergestellt ist und dieser mit Strom versorgt wird, muss noch Programmcode entwickelt werden, welcher die Daten zum richtigen Zeitpunkt anfragt. Damit der ESP möglichst wenig Strom verbraucht und nicht unnötig belastet wird, werden die Daten nur auf Anfrage von Außen ermittelt. Dieses Vorgehen nennt sich auch Datenkapselung. Leider hat es neben den genannten Vorteilen auch einen Nachteil. Die Daten können nämlich nach diesem Prinzip nur aktuell und nicht in die Vergangenheit abgefragt werden. Sollte man sich also für den Verlauf der Daten interessieren, so kann dieser nicht vom ESP geliefert werden. Dieses Problem ist jedoch einfach zu umgehen, indem die Anwendung sich die erhaltenen Sensordaten einfach in einer eigenen Datenbank speichert. Nach dem Start des ESP lässt sich der Vorgang in verschiedene Schritte unterteilen:

1. Verbindung mit dem konfigurierten WLAN herstellen
2. DNS-Eintrag erstellen
3. NTP-Client erstellen
4. Server-Endpoints definieren

Datenendpoint (JSON)

Websiteendpoint (HTML)

Nach diesem Aufbau wird beim Start des ESPs nur initialisiert. Nachdem dieser Prozess abgeschlossen ist, kann ein User die Website laden. Diese lässt sich dann über den Data-Endpoint die live-Daten geben und zeigt diese an.

3.1.3 Programmierung zum Anzeigen der bereitgestellten Sensordaten

Um die aktuellen Daten zu visualisieren gibt es verschiedene Möglichkeiten. Beispielsweise bietet HTML mit dem 'meta'-Tag eine Möglichkeit zum Aktualisieren der Seite in einem Intervall. Diese Option wäre jedoch sehr unschön, da das Aktualisieren sich statt nur auf die gezeigten Daten auf die gesamte Seite auswirken würde. Eine weitaus schönere Option ist das Aktualisieren der HTML-Elemente, in welchem die Daten angezeigt werden. Hierzu muss die angezeigte Seite ein Script enthalten, welches die Daten in einem festgelegten Intervall vom ESP abfragt und dann nur den Inhalt der spezifischen Elemente aktualisiert.

Ich habe mich für die Umsetzung der zweiten Lösung entschieden, da ich so mit einem geringen Mehraufwand eine beträchtliche Verbesserung für den Nutzer erreichen kann. Wenn die Webseite geladen wird, dann wird eine JavaScript Datei mitgegeben. Dieses Skript aktualisiert die angezeigten Werte auf der Website in einem festen Intervall einmalig jede Sekunde.

4 Zusammenfassung und Ausblick

4.1 Vorgehen

Kurze Zusammenfassung der zentralen Arbeitsschritte

4.2 Ergebnisse

kurze zusammenfassung des erreichten

4.3 Ausblick

Weiterer Umfang im Rahmen der T1000

5 EMPTY

6 EMPTY



(a) img 1



(b) img 2



(c) img 3

Abbildung 6.1: 3 Images



Abbildung 6.2: Bild mit Quelle (vgl. [Online.2022])



Abbildung 6.3: einfaches Bild

Referenz: Abbildung 6.3

7 EMPTY

Folgend werden Stichpunkte aufgelistet, welche während der Bearbeitung / vor der Abgabe kontrolliert werden sollten.

- Grammatik / Rechtschreibung überprüft?
- Bin ich innerhalb meiner Arbeit im Präsens geblieben?
- Ist der wissenschaftliche Aufbau meiner Arbeit erkennbar (Einleitung, theoretische Grundlagen, Hauptteil, Fazit)?
- Sind alle Verzeichnisse vollständig?
- Sind alle Abkürzungen im Abkürzungsverzeichnis eingetragen?
- Haben alle Abbildungen / Tabellen eine verständliche Beschriftung?
- Werden die Abbildungen / Tabellen im Text referenziert?
- Stimmt die Formatierung der Tabellen und Abbildungen?
- Quellen:
 - Autor, Titel, **Verlag** vorhanden?
 - Haben die Onlinequellen ein 'zuletzt aufgerufen'?
 - Sind bei Fachbüchern / E-Books die ISBN vorhanden?
 - Habe ich **jede** Behauptung mit einer Quelle belegt?
- Habe ich, nach den Vorlagen der DHBW, die gendergerechte Sprache genutzt?
- Stimmt das Verhältnis von Theorie / Praxis in meiner Arbeit?
- Sind alle Angaben auf dem Deckblatt aktuell und korrekt?
- Habe ich mir mindestens zwei Sicherheitskopien auf zwei verschiedenen Laufwerken abgelegt?
- Habe ich den korrekten akademischen Grad meines Betreuers angegeben?
- Habe ich die eidesstattliche Erklärung unterschrieben?

Literatur

- [1] Petteri Aimonen. *File:Animal echolocation.svg* - *Wikimedia Commons*. 2009. URL: https://commons.wikimedia.org/wiki/File:Animal_echolocation.svg?uselang=en#Licensing. (zuletzt geprüft: 12.11.2024).
- [2] Herbert Bernstein. *Mikrocontroller*. ger. Springer Vieweg Wiesbaden, 2020. ISBN: 978-3-658-30067-8.
- [3] Kevin Friedemann. *HC-SR04 Ultraschallsensor - Anleitung und Pinout* » *IoTspace.dev*. 2021. URL: <https://iotpace.dev/hc-sr04-ultraschallsensor-anleitung-und-pinout/>. (zuletzt geprüft: 12.11.2024).
- [4] Peter Raabe Armin und Holstein. *Akustik und Raumklima*. ger. Springer Vieweg Wiesbaden, 2021. ISBN: 978-3-658-33324-9.
- [5] Christian Thomas. *Berührungslos arbeitende Abstandssensoren in der Landtechnik*. ger. Carl Hanser Verlag, 1965. URL: <https://440ejournals.uni-hohenheim.de/index.php/Grundlagen/article/view/236/188>.
- [6] *Ultraschall Abstandssensor HC-SR04*. 2023. URL: https://wiki.hshl.de/wiki/index.php/Ultraschall_Abstandssensor_HC-SR04. (zuletzt geprüft: 09.11.2024).