

Pool-Überwachung

DIPLOMARBEIT

verfasst im Rahmen der

Reife- und Diplomprüfung

an der

Höheren Abteilung für Informatik

Eingereicht von:

Florian Wilflingseder

Betreuer:

Michael Wagner, Gerald Köck

Leonding, April 2023

Inhaltsverzeichnis

1	Einleitung	1
1.1	Sinn des Projektes:	1
1.2	Marktanalyse:	1
1.3	Am Markt befindliche Produkte:	2
1.4	Grundkonzept:	3
1.5	Umfrage:	3
1.6	Key-Features:	4
2	Technologien Hardware	5
2.1	TTGO-Lora-Funk-ESP32	5
2.2	ESP32-Chip:	7
2.3	LoRa-Funk Modul:	8
2.4	Verwendete Sensoren:	9
3	Technologien Software	13
3.1	Zu Beginn verwendete Software:	13
3.2	Unterschied Flutter und React:	14
3.3	Vor- und Nachteile von C gegenüber Arduino IDE:	16
3.4	Flutter:	18
3.5	Verwendete Libraries in Flutter:	20
3.6	C:	25
3.7	Verwendete Libraries in C:	25
4	Technologien Entwicklungsumgebung	31
4.1	Visual Studio Code:	31
5	Umsetzung	32
5.1	Umsetzung:	32
6	Erkenntnisse	III

Literaturverzeichnis	IV
Abbildungsverzeichnis	V
Quellcodeverzeichnis	VI

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Die Arbeit wurde bisher in gleicher oder ähnlicher Weise keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Die vorliegende Diplomarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

Leonding, April 2023

Florian Wilflingseder

1 Einleitung

1.1 Sinn des Projektes:

Das Projekt Pool-Überwachung soll IoT-Liebhabern ermöglichen jegliche Art von stillen Gewässern zu überwachen und dies alles auf ihrem Android, IOS, Windows, Linux Geräten oder in einer Webanwendung einfach einzusehen. Das Projekt ermöglicht es den Benutzern einen Schwimmer ins Wasser zu legen wie bei einem einfachen Pool-Thermometer. Jedoch kann dieser nicht nur die Temperatur anzeigen, sondern verfügt über die Funktion den PH-Wert und den NTU-Trübungswert zu erkennen. Ein zusätzliches Feature welches, vielen mit noch zu beaufsichtigenden Kindern ein Gefühl von Absicherung geben kann, ist eine Wellenerkennung die einen Alarm an das Handy sendet, falls ein verbauter Beschleunigungssensor verdächtige Aktivitäten wahrnimmt. Man kann diese Daten alle gemütlich in seiner App sehen und sich auch den Verlauf der Werte über bestimmte Zeiträume anzeigen lassen, dies kann wichtig sein falls nicht genug Chlor im Wasser ist oder nicht genügend reinigende Pflanzen in einem Teich gepflanzt sind. Da das Verbauen solcher Instrumente in das Pool sehr teuer ist oder im Nachhinein schwer möglich ist kommt Pool-Überwachung ins Spiel.

Angefangen wurde mit einer Marktanalyse, die die Mitbewerber in diesem Markt veranschaulicht, die Vor- und Nachteile heraushebt und den genauen Markt definiert.

1.2 Marktanalyse:

Das Produkt richtet sich an Privatpersonen die einen Garten mit einer beliebigen Schwimmanlage besitzen. Die Altersgruppe wird sich an alle Geschlechter zwischen 25-60 Jahren richten. Das Produkt wird besonders attraktiv für Erwachsene mit Kindern sein die noch Beaufsichtigung benötigen wenn sie sich in der Nähe von Gewässern aufhalten, wegen der Wellenerkennungsfunktion.

1.3 Am Markt befindliche Produkte:

Smart Poolthermometer Starter Set:

Das Set besteht aus 3 Teilen, einem schwimmenden Poolthermometer, einem Aussenfühler und einem Netzwerk Gateway. Kommt mit einer eigenen Smart App dazu. Es kann die Wassertemperatur die Luftaußentemperatur und die Luftfeuchtigkeit gemessen werden. Netzwerkfähig (funkts das Gateway (bei freier Sichtline bis zu 100 Meter) an, der mit dem W-Lan Verbunden ist). Unterstützt Amazon Alexa und Conrad Connect. Wie der Pool so auch der Außenthermometer werden mit 2x AA Batterien betrieben. Der Außenthermometer ist dazu nicht Wetterfest und es muss eine extra Schutzhülle mitbestellt werden. Details über die einzelnen Bauteile werden nicht bekannt gegeben.

WLAN Swimming Pool Thermometer Bundle|Inkbird:

Hat ein Aufstellbaren-Display, Outdoor Hygrometer und Thermometer, Datenlogger, Export-Funktion, Cloud, App. Es kann die Wassertemperatur die Luftaußentemperatur und die Luftfeuchtigkeit gemessen werden. Es wird nur ein Display zur Anzeige mitgeliefert und man kann seine Daten 12 Monate kostenlos in einer Cloud speichern. Details über die einzelnen Bauteile werden nicht bekannt gegeben.

Elektrobock ELBO-073 Wireless Pool Alarm:

Dieses Gerät dient nur als Alarmanlage für das Wasser. Es erkennt den Wellengang und löst seine eingebaute Sirene aus. Das Gerät besitzt ein Gegenstück welchen man in seiner Nähe an einer Steckdose anstecken kann und ein Signal von dem Sender im Wasser bekommt. Dadurch läutet die Sirene an 2 Standorten. Ansonsten bietet dieses keine zusätzlichen Funktionen. Details über die einzelnen Bauteile werden nicht bekannt gegeben.

1.4 Grundkonzept:

Man lässt die Sendereinheit im Wasser schwimmen und stellt die Empfängereinheit an einem trockenen Ort mit Strom- und W-Lan-Zugang auf. Nach dem beide Geräte aktiviert worden sind fangen sie mit dem Monitoring an und überwachen das Gewässer. Die Empfängereinheit erhält via Funk vom Sender die Daten und speichert diese ab. Auf diese Daten kann man dann mit seinem Smartphone oder Computer mit der dazugehörigen App zugreifen und sich die Daten anzeigen lassen. Der verbaute Alarm lässt sich in der App aktivieren und deaktivieren falls dieser nicht benötigt wird um unnötige Benachrichtigungen zu vermeiden. Der Sender sendet alle 3 Stunden die Daten an die Empfängereinheit um ein regelmäßiges und vergleichbares Bild der aufgenommenen Daten zu ermöglichen. Wenn der Sender inaktiv ist befindet er sich im Stromsparmodus um Strom zu sparen.

1.5 Umfrage:

Im Zuge der Diplomarbeit wurden Freunde Verwandte und Bekannte zum Design des Frontends befragt. Die Fragen bezogen sich auf das Design des Frontends, welche Daten angezeigt werden sollen, den Zeitraum den die Grafiken abdecken sollen und welche Funktionen in welcher Ausführung gefragt sind.

Zum Backend wurde besprochen welche wie lange die Daten gespeichert werden sollen. Ob es nötig ist diese über mehrere Jahre zu speichern und ob Sie immer Abrufbar sein sollen oder möglicherweise als CSV-Datei bereitgestellt werden sollen. Aus den Antworten hat sich dann der das jetzige Design und der Aufbau des Projektes ergeben.

1.6 Key-Features:

Die Key-Features dieses Produktes sind:

- Temperatur (Messung der Temperatur)
- Ph-Wert (Messung des Ph-Wertes)
- Trübungseinheit-NTU (Messung des NTU-Wertes)
- Alarm bei hohem Wellengang an das Smartphone
- Datenübertragung über große Entfernung
- Weltweiter Zugriff auf meine Daten
- Anschaulich gestaltete App

Die Sender- und Empfängereinheit können bis zu 10 Kilometer voneinander entfernt stehen.

2 Technologien|Hardware

2.1 TTGO-Lora-Funk-ESP32

Der TTGO-Lora-Funk-ESP32 ist ein leistungsstarker Mikrocontroller, der speziell für die Verwendung in der drahtlosen Datenübertragung entwickelt wurde. Er basiert auf dem ESP32-Chip von Espressif-Systems und verfügt über ein integriertes LoRa-Funkmodul, das eine drahtlose Kommunikation über große Entfernungen ermöglicht. Die Verwendung des ESP32-Chips bietet dem TTGO-Lora-Funk-ESP32 eine hohe Rechenleistung und Speicherkapazität. Der Chip verfügt über einen Dual-Core-Prozessor mit einer Taktrate von bis zu 240 MHz, der eine schnelle Datenverarbeitung ermöglicht. Darüber hinaus bietet der Chip bis zu 4 MB Flash-Speicher und bis zu 520 KB SRAM, was ausreichend Platz für die Speicherung von Programmcode und Daten bietet. Das integrierte LoRa-Funkmodul ermöglicht die drahtlose Übertragung von Daten über große Entfernungen. Es unterstützt eine Übertragungsgeschwindigkeit von bis zu 300 kbps und eine Reichweite von bis zu 10 km in ländlichen Gebieten und bis zu 2 km in städtischen Gebieten da dort die Signale von den Gebäuden geschwächt werden. Dies macht den TTGO-Lora-Funk-ESP32 ideal für Anwendungen, die eine drahtlose Kommunikation über große Entfernungen erfordern, wie beispielsweise die Überwachung von Umgebungsbedingungen in der Landwirtschaft oder die Übertragung von Daten in Smart-City-Anwendungen. Im Zuge dieser Diplomarbeit kam diese Funktion zur Verwendung um die von den Sensoren gelesenen Daten an einen zweiten ESP-32 mit identer Ausstattung zu senden und von dort für das Flutter Frontend bereit zu stellen. Der TTGO-Lora-Funk-ESP32 ist auch mit einer Vielzahl von Schnittstellen ausgestattet, die eine einfache Integration in verschiedene Anwendungen ermöglichen. Er verfügt über einen Micro-USB-Anschluss für die Stromversorgung und Programmierung, sowie über einen JST-XH-Anschluss um eine externe Batterie anzuschließen. Darüber hinaus verfügt der Mikrocontroller über GPIO-Pins, I2C-, SPI- und UART-Schnittstellen, die eine einfache Integration mit Sensoren, Displays und anderen Geräten ermöglichen. Darüber hinaus bietet der TTGO-Lora-Funk-ESP32 eine kosteneffektive Lösung für die drahtlose Datenübertragung. Im Vergleich zu anderen drahtlosen Übertragungstechnologien wie Mobilfunk oder Wi-Fi ist LoRa-Funk eine kosteneffektive Alternative. Eine weitere

Stärke des TTGO-Lora-Funk-ESP32 ist seine Energieeffizienz. Der ESP32-Chip unterstützt verschiedene Energiesparmodi und bietet einen integrierten Stromsparmodus, der die Energieaufnahme des Mikrocontrollers reduziert, wenn er nicht aktiv ist. Dies ist besonders nützlich in batteriebetriebenen Anwendungen, da es die Lebensdauer der Batterie verlängert. Ein weiterer Vorteil des TTGO-Lora-Funk-ESP32 ist seine Unterstützung durch eine aktive Community von Entwicklern. Es gibt eine Vielzahl von Foren und Online-Communities, die sich auf die Entwicklung von Anwendungen mit dem TTGO-Lora-Funk-ESP32 spezialisiert haben. Dies macht es einfacher, Unterstützung und Hilfe zu finden, wenn man bei der Entwicklung von Anwendungen auf Probleme stößt. Da der TTGO-Lora-Funk-ESP32 ein leistungsstarker und vielseitiger Mikrocontroller ist, der für die drahtlose Datenübertragung entwickelt wurde eine hohe Rechenleistung besitzt, dazu aber noch kosten- und energieeffizient ist, fiel die Entscheidung bei dieser Arbeit auf ihn. [1]

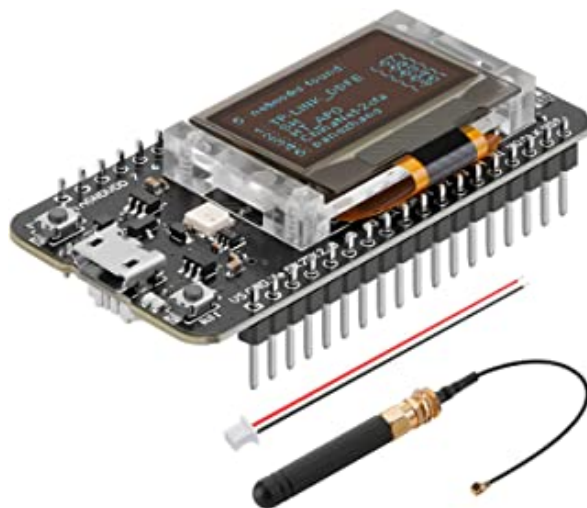


Abbildung 1: ESP32-TTGO-LoRa-Oled

2.2 ESP32-Chip:

- Der ESP32 unterstützt bis zu 4×16 MB externen Speicher.
- Man kann den internen Takt (8 MHz) oder einen externen Quarztakt mit üblicherweise 160 MHz nutzen. Wenn man den Prozessor zurück setzt übernimmt er auch das System-Timing.
- Hall-Sensor: Kann für die Messung von Magnetfeldschwankungen genutzt werden.
- Ein interner Temperatursensor mit einem Messbereich von -40 bis 125 Grad ist ebenfalls vorhanden. Die analogen Messwerte werden wie bei dem Hall-Sensor von einem Analog-Digital-Wandler digitalisiert. In neueren Modellen ist kein Temperatur Sensor mehr verbaut.
- Es sind 34 GPIOs (universelle Ein- und Ausgänge) vorhanden. Verwendet werden diese für Ein-und Ausgabe analoger und digitaler Signale. Die Pins sind mehrfach belegt. Mit internen Pull-down und Pull-up-Widerständen können definierte Zustände herbeigeführt werden.
- Der ESP32 kann Signale von bis zu zehn unterschiedlichen TouchSensoren verarbeiten.
- Der ESP32 Unterstützt Bluetooth als auch W-Lan im 2.4-GHz Bereich und kann diese Empfangen und Senden.
- Es wird Bluetooth 4.2 und Bluetooth low-energy unterstützt.
- Der ESP32 hat vier 64-Bit-UniversalTIMER. Diese kann man via Software steuern.
- Es gibt Watchdog-Timer. Man unterscheidet zwischen Main-Watchdog-Timern und RTC-Watchdog-Timern. Auslösen kann man hiermit einen CPU-Reset , ein Core-Reset oder ein Interrupt.
- 12-Bit-A/D-Wandler (Analog-Digital-Wandler) mit 18 Kanälen
- 8-Bit-DAC (Digital-Analog-Wandler)
- SPI-Schnittstellen (SPI1, HSPI and VSPI) mit Master- oder Slave-Modus
- Es werden zwei I2C-Bus-Schnittstellen vorgehalten, die im Master- oder SlaveModus betrieben werden können.
- Pulsweitenmodulation (PWM) um Geräte wie Motoren, elektrische Heizungen oder Ähnliches zu steuern.

[2]

2.3 LoRa-Funk Modul:

LoRa (Long Range) ist eine drahtlose Übertragungstechnologie, die speziell für den Einsatz in IoT (Internet of Things) Anwendungen entwickelt wurde. Das LoRa-Funk-Modul ist ein elektronisches Bauteil, das die LoRa-Technologie integriert und es Geräten ermöglicht, Daten über große Entfernungen drahtlos zu übertragen. Das LoRa-Funk-Modul arbeitet auf der Basis von Funkfrequenzen im ISM-Band (Industrial, Scientific and Medical Band) von 868 MHz oder 915 MHz. Diese Frequenzen ermöglichen eine Übertragung von Daten über Entfernungen von bis zu 10 Kilometern, was die LoRa-Technologie ideal für den Einsatz in Anwendungen wie Smart-Cities, Landwirtschaft oder Industrie macht. Ein weiterer Vorteil des LoRa-Funk-Moduls ist seine geringe Stromaufnahme. Das LoRa-Funk-Modul ist auch sehr zuverlässig. Das LoRa-Funk-Modul ist auch sehr einfach zu integrieren. Es ist kompatibel mit einer Vielzahl von Mikrocontrollern und kann über verschiedene Schnittstellen wie UART oder SPI angeschlossen werden. Darüber hinaus gibt es eine Vielzahl von Bibliotheken und Entwicklertools, die von der Community entwickelt wurden, um die Integration mit verschiedenen Plattformen und Anwendungen zu erleichtern.

Zusammenfassend ist das LoRa-Funk-Modul eine leistungsfähige und zuverlässige drahtlose Übertragungstechnologie, die speziell für den Einsatz in IoT-Anwendungen entwickelt wurde und deshalb perfekt für dieses Projekt geeignet ist.

2.4 Verwendete Sensoren:

TemperaturSensor:

Der DS18B20 ist ein digitaler Temperatursensor, der von der Firma Maxim Integrated entwickelt wurde. Es handelt sich um einen 1-Wire-Sensor, der eine hohe Genauigkeit und Zuverlässigkeit bietet und in einer Vielzahl von Anwendungen eingesetzt wird. Der DS18B20 besteht aus einem temperaturabhängigen Sensor, einem Analog-Digital-Converter (ADC) und einer seriellen Schnittstelle. Der Sensor ist in einem wasserdichten Edelstahlgehäuse untergebracht. Der ADC wandelt das analoge Ausgangssignal des Sensors in ein digitales Signal um, das über die serielle Schnittstelle übertragen wird. Der DS18B20 bietet eine hohe Genauigkeit mit einer Auflösung von bis zu 12 Bit und einer Genauigkeit von $\pm 0,5^{\circ}\text{C}$ im Temperaturbereich von -10°C bis $+85^{\circ}\text{C}$. Ein weiterer Vorteil des DS18B20 ist seine einfache Verbindung mit einem Mikrocontroller oder einem Computer. Der DS18B20 bietet auch eine programmierbare Auflösung, die es ermöglicht, die Genauigkeit des Sensors an die spezifischen Anforderungen einer Anwendung anzupassen. Entwickler können die Auflösung des Sensors auf 9, 10, 11 oder 12 Bit programmieren, um eine höhere Genauigkeit oder eine schnellere Reaktionszeit zu erreichen. Da der DS18B20 eine hohe Genauigkeit, eine schnelle Reaktionszeit, eine einfache Integration in Systeme und eine programmierbare Auflösung bietet wurde er für dieses Projekt ausgewählt.

Datasheet: https://cdn.shopify.com/s/files/1/1509/1638/files/DS18B20_3mCable_datasheet.pdf?v=1644320674

[3]



Abbildung 2: TemperaturSensor

pH-Sensor:

Der E-201-C ist ein pH-Sensor, der zur Messung des pH-Werts von Flüssigkeiten und Lösungen verwendet wird. Es ist ein elektrochemischer Sensor, der durch die Messung der Spannung zwischen einer pH-sensitiven Elektrode und einer Referenzelektrode arbeitet. Der Sensor wird über eine BNC-Buchse an ein Messgerät angeschlossen. Der E-201-C bietet eine hohe Genauigkeit und Empfindlichkeit mit einer Auflösung von 0,01 pH-Einheiten und einer Genauigkeit von $\pm 0,1$ pH-Einheiten im Temperaturbereich von 0 bis 80°C. Der Sensor bietet auch eine schnelle Reaktionszeit, wodurch Messungen in Echtzeit durchgeführt werden können. Ein weiterer Vorteil des E-201-C pH-Sensors ist seine hohe Stabilität und Haltbarkeit. Die pH-sensible Membran ist chemisch inert und widerstandsfähig gegenüber Korrosion und Abnutzung. Dies gewährleistet eine lange Lebensdauer des Sensors und eine genaue Messung über einen längeren Zeitraum. Der E-201-C pH-Sensor wird in einer Vielzahl von Anwendungen eingesetzt, darunter in der chemischen Industrie, der Lebensmittelindustrie, der Umweltüberwachung und der pharmazeutischen Industrie. Es ist ein wichtiger Sensor zur Messung des pH-Werts in Lösungen und Flüssigkeiten, um sicherzustellen, dass sie den erforderlichen Spezifikationen entsprechen.

Datasheet: <https://www.e-gizmo.net/oc/kits%20documents/PH%20Sensor%20E-201-C/PH%20Sensor%20E-201-C.pdf>

[4]



Abbildung 3: Ph-Sensor

NTU-Sensor|Trübungs-Sensor:

Das TS-300B Trübungssensor-Modul ist ein Qualitäts-Trübungssensor, der zur Messung der Trübung von Flüssigkeiten verwendet wird. Es ist ein optischer Sensor, der durch Messung des Streulichts arbeitet, das von Verunreinigungen in der Flüssigkeit reflektiert wird. Dieser Trübungssensor bietet eine hohe Empfindlichkeit und Genauigkeit mit einer Messauflösung von bis zu 0,1 NTU (Nephelometric Turbidity Unit) und einer Messgenauigkeit von ± 5 Prozent im Bereich von 0 bis 1000 NTU. Der Sensor bietet auch eine schnelle Reaktionszeit und kann Trübungen in Echtzeit messen. Der Sensor kann über einen einfachen Analogausgang an ein Messgerät oder einen Mikrocontroller angeschlossen werden. Es ist auch einfach zu installieren und zu warten, da es keine beweglichen Teile hat und daher eine geringe Wartung erfordert. Der TS-300B Trübungssensor wird in einer Vielzahl von Anwendungen eingesetzt, darunter in der Wasseraufbereitung, der Abwasserbehandlung, der Lebensmittel- und Getränkeindustrie und der Umweltüberwachung. Es ist ein wichtiger Sensor zur Messung der Wasserqualität und zur Überwachung von Prozessen, die von der Trübung der Flüssigkeit abhängen.

Datasheet: <https://de.aliexpress.com/item/4000321791231.html>

[5]



Abbildung 4: NTU-Sensor

Gyroskop/Beschleunigungssensor:

Der SEN-MPU6050 ist ein Kombinationssensor, der sowohl einen Gyroskop- als auch einen Beschleunigungssensor enthält. Diese beiden Sensoren arbeiten zusammen, um die Bewegungen und Orientierungen eines Objekts in der Raumebene zu messen. Der Sensor misst die Winkelgeschwindigkeit eines Objekts um seine drei Achsen (x, y, z). Der Sensor erfasst dabei die Änderungen der Rotationsgeschwindigkeit, die durch die Drehung des Objekts verursacht werden. Der Beschleunigungssensor hingegen misst die Beschleunigung des Objekts in jeder der drei Achsen (x, y, z). Der Sensor erfasst dabei die Änderungen der Geschwindigkeit, die durch eine Beschleunigung oder Verzögerung des Objekts verursacht werden. Der SEN-MPU6050 bietet eine hohe Genauigkeit und Empfindlichkeit mit einer Auflösung von bis zu 16 Bit für beide Sensoren. Der SEN-MPU6050 wird in einer Vielzahl von Anwendungen eingesetzt, darunter in der Robotik, der Luft- und Raumfahrt, der Drohnensteuerung, der Navigation, der virtuellen Realität und der Spieleentwicklung. Es ist ein wichtiger Sensor zur Messung der Bewegung und Orientierung von Objekten in der Raumebene.

Datasheet: <https://joy-it.net/de/products/SEN-MPU6050>

[6]

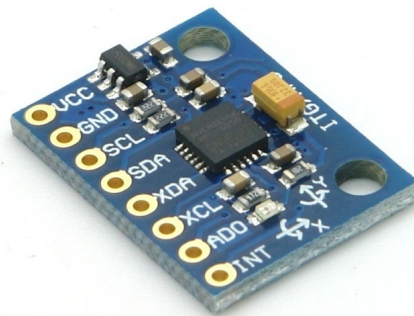


Abbildung 5: Beschleunigungssensor

3 Technologien|Software

3.1 Zu Beginn verwendete Software:

Zu Beginn der Diplomarbeit wurde für die Umsetzung der Software „React“ für das Frontend und die „Arduino IDE“ für das Backend verwendet und sich in die Software eingearbeitet. Nach den ersten Versuchen die Anforderungen der Diplomarbeit in diesen Umgebungen umzusetzen, wurde klar, dass diese die nicht beste Lösung darstellen. React war nicht gut genug für die Plattformübergreifende Programmierung geeignet wie gedacht. „React“ wurde daraufhin durch Flutter ersetzt. Bei der Arduino IDE stellte sich heraus, dass es zwar die nötigen Bibliotheken gab aber die einzelnen Pins des Mikroprozessors durch die Software nicht erkannt wurden und eine performante Datenbank, auf der die Daten gespeichert werden sehr schwer umzusetzen ist. Deswegen wurde das Arduino-Framework durch das ESP32 native ESP-SDK von Espressif-Systems ersetzt da es API's auf Hardware-Ebene beinhaltet. Die optimierten vorkompilierten Bibliotheken und kompilierfertigen Treiberbibliotheken, des SDKs, verkürzen die Markteinführungszeit und gewährleisten gleichzeitig die Freiheit der Anpassung. Die Datenbank wurde mit einem File-System (Spiffs) auf dem ESP32 realisiert. Als Entwicklungs Umgebung wurde „Visual Studio Code“ verwendet.

3.2 Unterschied Flutter und React:

Flutter und React sind zwei der beliebtesten Frameworks für die Entwicklung von mobilen Anwendungen und Webanwendungen. Flutter ist ein mobiles App-Entwicklungs-Framework, das von Google entwickelt wurde. Es basiert auf der Programmiersprache Dart und verwendet eine eigene Rendering-Engine, die auf der GPU basiert. React hingegen ist ein Framework für die Entwicklung von Webanwendungen, das von Facebook entwickelt wurde. Es basiert auf JavaScript und ermöglicht Entwicklern die Erstellung von ansprechenden und interaktiven

Webanwendungen mit einer flexiblen Architektur. React nutzt eine virtuelle DOM-Struktur, die es Entwicklern ermöglicht, schnelle und effiziente Anwendungen zu erstellen. Ein wichtiger Unterschied zwischen Flutter und React ist die Art der Anwendungen, für die sie entwickelt wurden. Flutter wurde speziell für die Entwicklung von mobilen Anwendungen entwickelt, während React in erster Linie für die Entwicklung von Webanwendungen konzipiert wurde.

Dies bedeutet, dass Flutter speziell auf die Bedürfnisse von mobilen Endbenutzern zugeschnitten ist und deswegen den Use-Case dieser Diplomarbeit trifft, während React auf die Bedürfnisse von Webanwendungen ausgerichtet ist. Ein weiterer Unterschied zwischen den beiden Frameworks ist ihre Architektur. Flutter verwendet eine Widget-basierte Architektur, bei der alle UI-Elemente als Widgets dargestellt werden. React verwendet hingegen eine komponentenbasierte Architektur, bei der UI-Elemente als Komponenten dargestellt werden. Sie unterscheiden sich jedoch nicht nur in der Architektur sondern auch in der Art und Weise, wie sie mit der Plattform interagieren. Flutter verwendet eine eigene Rendering-Engine, die auf der GPU basiert, um eine hohe Leistung und eine schnelle Reaktionszeit zu ermöglichen. React hingegen verwendet eine virtuelle DOM-Struktur, die es Entwicklern ermöglicht, Änderungen an der

Benutzeroberfläche schnell zu erfassen und anzuzeigen. Die virtuelle DOM besteht aus JavaScript-Objekten, die die verschiedenen HTML-Elemente, Attribute und Inhalte repräsentieren. Diese virtuelle DOM-Struktur wird verwendet, um Änderungen im UI-Design zu verwalten, bevor sie auf dem tatsächlichen HTML-DOM angewendet werden. Dadurch können Änderungen im UI-Design effizienter und schneller durchgeführt werden, da nur die tatsächlich geänderten Teile aktualisiert werden müssen, ohne das gesamte Dokument neu zu rendern. Ein weiterer wichtiger Unterschied zwischen den beiden Frameworks ist die Verfügbarkeit von Bibliotheken und Tools. Flutter bietet eine umfangreiche Sammlung von Widgets und Tools, die speziell für die Entwicklung von mobilen Anwendungen entwickelt wurden. React hingegen hat eine große und aktive Entwicklergemeinschaft, die eine Vielzahl von Bibliotheken und Tools entwickelt hat, die für die Entwicklung von Webanwendungen genutzt werden können. Ein weiterer wichtiger Unterschied ist die Verfügbarkeit von Plattform-Unterstützung. Flutter unterstützt sowohl Android als auch iOS und bietet eine native Leistung auf beiden Plattformen. React hingegen unterstützt Webanwendungen sowie mobile Anwendungen durch die Verwendung von React Native, das jedoch immer noch auf der Web-Technologie basiert. In Bezug auf die Entwicklungskosten und die Zeit, die für die Entwicklung benötigt wird, haben beide Frameworks ihre Vor- und Nachteile. Flutter bietet eine schnellere Entwicklungszeit und eine geringere Anzahl von Fehlern, da es auf einer Widget-basierten Architektur basiert und über eine umfangreiche Sammlung von Widgets und Tools verfügt. React hingegen hat eine größere Entwicklergemeinschaft und bietet eine größere Flexibilität in der Entwicklung von Webanwendungen. [7]

3.3 Vor- und Nachteile von C gegenüber Arduino IDE:

Die Programmiersprache C ist eine der am häufigsten verwendeten Programmiersprachen für die Mikroprozessorprogrammierung. C ist eine Hochsprache, die auf einer Vielzahl von Plattformen ausgeführt werden kann und gleichzeitig eine systemnahe Programmierung ermöglicht. Dies macht C zu einer beliebten Wahl für die Entwicklung von Firmware und Anwendungen für Mikrocontroller und Mikroprozessoren. Ein Vorteil von C ist seine Effizienz und Leistungsfähigkeit. C ist eine schnelle und effiziente Programmiersprache, die eine direkte Steuerung der Hardware ermöglicht und daher ideal für die Mikroprozessorprogrammierung ist. Es bietet auch eine umfangreiche Unterstützung für arithmetische Operationen und Bitmanipulationen, die bei der Entwicklung von Embedded-Systemen häufig erforderlich sind. C bietet auch eine hohe Portabilität und eine umfangreiche Unterstützung durch eine aktive Entwicklergemeinschaft. C-Code kann auf einer Vielzahl von Mikroprozessoren und Betriebssystemen ausgeführt werden und ist in der Regel plattformunabhängig. Die C-Standardbibliothek bietet eine Vielzahl von Funktionen, die in der Mikroprozessorprogrammierung häufig verwendet werden, wie z.B. mathematische Funktionen, Ein- und Ausgabe, Zeichenkettenmanipulation und Speicherverwaltung.

Ein weiterer Vorteil von C ist die Möglichkeit zur Optimierung des Codes. C bietet Entwicklern die Möglichkeit, den Code für die Mikroprozessorprogrammierung manuell zu optimieren, um eine bessere Leistung und effizientere Nutzung von Ressourcen zu erzielen. Dies ist besonders wichtig bei der Entwicklung von Anwendungen für Embedded-Systeme, bei denen die Ressourcen begrenzt sind und die Leistung von größter Bedeutung ist. Im Vergleich zur Programmierung mit der Arduino IDE gibt es jedoch einige mögliche Vor- und Nachteile von C. Die Arduino IDE ist eine integrierte Entwicklungsumgebung (IDE), die speziell für die Programmierung von Arduino-Mikrocontrollern entwickelt wurde. Die IDE bietet eine vereinfachte Programmierschnittstelle und eine Reihe von Bibliotheken und Beispielcodes, die die Entwicklung von Anwendungen für Arduino-Mikrocontroller erleichtern. Ein Vorteil der Arduino IDE ist ihre Benutzerfreundlichkeit. Die IDE bietet eine intuitive Benutzeroberfläche, die es auch Einsteigern erleichtert, Mikrocontroller zu programmieren. Die IDE enthält auch eine Vielzahl von Beispielen und Bibliotheken, die die Entwicklung von Anwendungen beschleunigen können. Ein Nachteil der Arduino IDE ist jedoch, dass sie auf der Verwendung von speziell entwickelten Funktionen und Bibliotheken basiert, die nicht unbedingt mit anderen Mikroprozessoren und Plattformen kompatibel sind. Dies kann die Wiederverwendbarkeit von Code und die Portabilität von Anwendungen einschränken. Darüber hinaus kann die Verwendung

von vorgefertigten Bibliotheken die Flexibilität und Kontrolle über die Anwendung einschränken. Insgesamt bietet die Programmiersprache C eine leistungsstarke und flexible Option für die Mikroprozessorprogrammierung. Es bietet Entwicklern eine direkte Kontrolle über die Hardware und eine effiziente Nutzung von Ressourcen, was bei der Entwicklung von Embedded-Systemen von entscheidender Bedeutung ist. C bietet auch eine hohe Portabilität und Flexibilität, was die Wiederverwendbarkeit von Code und die Entwicklung von Anwendungen für eine Vielzahl von Plattformen ermöglicht. Jedoch erfordert die Programmierung mit C ein höheres Maß an Erfahrung und technischen Kenntnissen als die Programmierung mit der Arduino IDE. Entwickler müssen in der Lage sein, Code von Grund auf zu schreiben und manuell zu optimieren, was eine höhere Komplexität und ein höheres Maß an Kontrolle erfordert. Dies kann die Entwicklung von Anwendungen in C zeitaufwändiger und komplizierter machen als die Verwendung von vorgefertigten Bibliotheken in der Arduino IDE. Insgesamt hängt die Wahl zwischen C und der Arduino IDE von den spezifischen Anforderungen und Anwendungen ab. Wenn eine höhere Kontrolle über die Hardware und eine optimale Leistung erforderlich sind, ist C möglicherweise die bessere Wahl. Wenn jedoch eine einfachere Programmierschnittstelle und eine schnellere Entwicklung erforderlich sind, kann die Arduino IDE eine bessere Option sein.

3.4 Flutter:

Flutter ist ein Open-Source-Framework, das von Google entwickelt wurde, um die App-Entwicklung zu erleichtern. Es wurde erstmals im Jahr 2017 veröffentlicht und hat seitdem an Beliebtheit gewonnen. Ein großer Vorteil von Flutter ist die Möglichkeit, plattformübergreifende Apps zu erstellen. Mit Flutter können Entwickler Apps für iOS und Android mit einer einzigen Codebase erstellen, was die Entwicklungszeit erheblich verkürzt. Im Vergleich dazu erfordert die native App-Entwicklung separate Codebases für jede Plattform, was zu einer längeren Entwicklungszeit führt. Außerdem können Entwickler mit Flutter auch Apps für das Web, Desktop und Embedded-Geräte erstellen, was die Flexibilität des Frameworks weiter erhöht.

Ein weiterer Vorteil von Flutter ist die schnelle Entwicklung. Flutter bietet ein Hot-Reload-Feature, das Entwicklern die Möglichkeit gibt, Änderungen an ihrem Code vorzunehmen und die Auswirkungen dieser Änderungen sofort zu sehen, ohne dass sie die App neu starten müssen. Dies ermöglicht es Entwicklern, schnell Fehler zu beheben und Funktionen hinzuzufügen, was zu einer schnelleren Entwicklung und einer schnelleren Markteinführung führt. Flutter bietet auch eine hohe Benutzeroberflächen-Performance. Es verwendet eine eigene Rendering-Engine, die es Entwicklern ermöglicht, eine reibungslose und schnelle Benutzeroberfläche zu erstellen. Im Vergleich dazu sind andere Frameworks wie React-Native darauf angewiesen, dass die nativen APIs der Plattform für die Benutzeroberflächen-Performance verwendet werden. Ein weiterer Vorteil von Flutter ist die einfache Anpassbarkeit der Benutzeroberfläche. Flutter bietet eine Vielzahl von Widgets, die Entwickler verwenden können, um eine ansprechende Benutzeroberfläche zu erstellen. Diese Widgets sind anpassbar und können für verschiedene Zwecke verwendet werden.

Darüber hinaus können Entwickler mit Flutter auch benutzerdefinierte Widgets erstellen, um eine einzigartige Benutzeroberfläche zu erstellen. Flutter bietet auch eine umfassende Dokumentation und Unterstützung. Es gibt eine aktive Community von Entwicklern, die Flutter verwenden und Unterstützung bieten. Darüber hinaus bietet Google eine umfassende Dokumentation und Schulungsmaterialien für Entwickler, die Flutter verwenden möchten. Ein weiterer Vorteil von Flutter ist die einfache Integration von Drittanbieter-Bibliotheken. Entwickler können leicht Drittanbieter-Bibliotheken in ihre Flutter-Apps integrieren, um zusätzliche Funktionen hinzuzufügen. Darüber hinaus bietet Flutter eine Vielzahl von Plugins und Paketen, die Entwickler verwenden können, um ihre Apps zu erweitern. Flutter bietet auch eine hohe Stabilität und Sicherheit. Die Entwicklung von Flutter erfolgt in einer isolierten Umgebung, um sicherzustellen, dass

keine externen Faktoren die Stabilität des Frameworks beeinträchtigen. Darüber hinaus bietet Flutter auch eine hohe Sicherheit, da es die Dart-Programmiersprache verwendet, die eine starke Typisierung und Überprüfung von Variablen und Objekten ermöglicht. Dies hilft, potenzielle Fehler und Sicherheitsprobleme frühzeitig zu erkennen und zu vermeiden. Ein weiterer Vorteil von Flutter ist die einfache Wartung. Da Entwickler nur eine einzige Codebase für plattformübergreifende Apps verwenden, wird die Wartung der Apps wesentlich einfacher. Entwickler müssen keine separate Codebase für jede Plattform pflegen, was zu einer deutlichen Reduzierung des Wartungsaufwands führt. Flutter ist auch sehr skalierbar. Es ermöglicht Entwicklern, Apps für eine große Anzahl von Benutzern zu erstellen und zu skalieren, ohne dass es zu Leistungsproblemen kommt. Flutter verwendet eine hocheffiziente Rendering-Engine und bietet auch Funktionen wie Lazy Loading, um sicherzustellen, dass Apps schnell und effizient ausgeführt werden können. Ein Vorteil von Flutter ist die einfache Integration von Backend-Systemen. Flutter bietet eine Vielzahl von Optionen für die Integration von Backend-Systemen, einschließlich RESTful APIs und Datenbanken. Dies ermöglicht Entwicklern, leistungsfähige Apps zu erstellen, die nahtlos mit Backend-Systemen interagieren können. Ein weiterer Vorteil von Flutter ist die Verfügbarkeit von Tools und Frameworks für automatisierte Tests. Mit Flutter können Entwickler automatisierte Tests für ihre Apps erstellen, um sicherzustellen, dass sie korrekt funktionieren und fehlerfrei sind. Flutter bietet auch eine Vielzahl von Tools und Frameworks für die Durchführung von Tests, einschließlich Widget-Tests und Integrationstests. Insgesamt bietet Flutter eine Vielzahl von Vorteilen für Entwickler, die plattformübergreifende Apps erstellen möchten. Es ermöglicht Entwicklern, schnell und effizient Apps zu erstellen, die auf verschiedenen Plattformen und Geräten ausgeführt werden können.

Flutter bietet auch eine hohe Benutzeroberflächen-Performance, Anpassbarkeit, Dokumentation und Unterstützung, Integration von Drittanbieter-Bibliotheken, Stabilität, Sicherheit, Wartbarkeit, Skalierbarkeit und Integration von Backend-Systemen. Daher ist Flutter eine ausgezeichnete Wahl für Entwickler, die eine plattformübergreifende App erstellen möchten, die schnell, effizient und ansprechend ist. [8]

3.5 Verwendete Libraries in Flutter:

syncfusion_flutter_charts:

Die Flutter-Bibliothek `syncfusion_flutter_charts` ist eine leistungsstarke und vielseitige Lösung für Flutter-Entwickler, die qualitativ hochwertige und anpassbare Diagramme in ihren Anwendungen erstellen möchten. In diesem Artikel werden wir die Funktionen und Vorteile der Bibliothek genauer betrachten. Syncfusion-Flutter-Charts ist eine Bibliothek von Diagrammen und Grafiken, die in Flutter-Anwendungen verwendet werden können. Die Bibliothek bietet eine breite Palette von Diagrammtypen, darunter Balken-, Linien-, Kreis-, Flächen-, Säulen-, Scatter- und Bubble-Diagramme. Darüber hinaus bietet es eine Vielzahl von Funktionen und Anpassungsoptionen, um Benutzern eine bessere Kontrolle über das Erscheinungsbild und die Funktionalität ihrer Diagramme zu geben. Die Syncfusion-Flutter-Charts-Bibliothek bietet eine Vielzahl von Diagrammtypen, die für verschiedene Anwendungen und Anforderungen geeignet sind. Zu den verfügbaren Diagrammtypen gehören:

- Balkendiagramm: ein Diagrammtyp, der verwendet wird, um Daten in horizontalen Balken darzustellen.
- Liniendiagramm: ein Diagramm, das verwendet wird, um Daten in einer Linie darzustellen, die durch die Punkte in einem Koordinatensystem verläuft.
- Kreisdiagramm: ein Diagrammtyp, der verwendet wird, um Daten in einem Kreis darzustellen.
- Flächendiagramm: ein Diagramm, das verwendet wird, um Daten in einer Fläche darzustellen, die durch die Punkte in einem Koordinatensystem begrenzt wird.
- Säulendiagramm: ein Diagrammtyp, der verwendet wird, um Daten in vertikalen Säulen darzustellen.
- Scatter-Diagramm: ein Diagrammtyp, der verwendet wird, um Datenpunkte in einem Koordinatensystem darzustellen.
- Bubble-Diagramm: ein Diagrammtyp, der verwendet wird, um Datenpunkte in einem Koordinatensystem darzustellen, wobei die Größe jedes Punktes ein zusätzliches Datenattribut darstellt.

Syncfusion-Flutter-Charts bietet eine breite Palette von Anpassungsoptionen, mit denen Benutzer das Erscheinungsbild und die Funktionalität ihrer Diagramme steuern können. Zu den verfügbaren Anpassungsoptionen gehören:

- **Achsenanpassung:** Die Benutzer können die Achsenbeschriftungen, die Achsentitel und die Achsenposition anpassen.
- **Legendenanpassung:** Die Benutzer können die Position, Größe, Schriftart und Farbe der Legende anpassen.
- **Farbanpassung:** Die Benutzer können die Farbe jedes Diagrammelements anpassen, einschließlich der Datenpunkte, der Achsenlinien und der Hintergrundfarbe des Diagramms.
- **Beschriftungsanpassung:** Die Benutzer können die Größe, Farbe und Schriftart der Beschriftungen anpassen, die jedes Diagrammelement begleiten.

Syncfusion-Flutter-Charts wurde mit einem API-Design entwickelt, das einfach zu verwenden und intuitiv zu verstehen ist. Die API bietet den Benutzern eine klare Dokumentation und viele Beispiele, um den Einstieg zu erleichtern. Die API bietet eine Vielzahl von Methoden und Eigenschaften, die es den Benutzern ermöglichen, ihre Diagramme anzupassen und Daten zu aktualisieren.

Syncfusion-Flutter-Charts ist auch eine leistungsstarke Bibliothek. Die Bibliothek bietet eine Vielzahl von Optimierungen, um sicherzustellen, dass Diagramme schnell gerendert werden und auf verschiedenen Geräten gut funktionieren. Einige der Leistungsmerkmale der Bibliothek sind:

- **Datenbindung:** Syncfusion-Flutter-Charts unterstützt die Bindung von Datenquellen wie Listen und Arrays. Dadurch wird das Aktualisieren von Daten in Echtzeit erleichtert.
- **Animationen:** Syncfusion-Flutter-Charts bietet eine Vielzahl von Animationen, die auf Diagrammen angewendet werden können, um die Benutzererfahrung zu verbessern.
- **Interaktivität:** Die Bibliothek unterstützt eine Vielzahl von Interaktionen, einschließlich Zoomen, Schwenken, Berühren und Klicken.
- **Responsivität:** Syncfusion-Flutter-Charts wurde für verschiedene Gerätegrößen optimiert und kann auf verschiedenen Geräten gut funktionieren.

http-Library:

http-Library ist eine Bibliothek für Flutter, die Entwicklern ermöglicht, HTTP-Anfragen an einen Server zu senden und Daten abzurufen. Die Bibliothek basiert auf der Dart-Programmiersprache und bietet eine einfache und intuitive API für Entwickler, um auf HTTP-Ressourcen zuzugreifen. Mit diesem Library wird auf die http-Requests die von dem ESP32 bereitgestellt werden, zugegriffen. Die http-Library unterstützt eine Vielzahl von HTTP-Anfragen, einschließlich GET, POST, PUT, DELETE, HEAD und PATCH. Entwickler können diese Anfragen verwenden, um Ressourcen von einem Server abzurufen oder zu aktualisieren. Die http-Library wurde mit einem API-Design entwickelt, das einfach zu verwenden und intuitiv zu verstehen ist. Die API bietet den Entwicklern eine klare Dokumentation und viele Beispiele, um den Einstieg zu erleichtern. Die API bietet eine Vielzahl von Methoden und Eigenschaften, die es den Entwicklern ermöglichen, HTTP-Anfragen zu erstellen und auf HTTP-Antworten zuzugreifen. Die http-Library bietet eine Vielzahl von Leistungsmerkmalen, um sicherzustellen, dass HTTP-Anfragen schnell und effizient ausgeführt werden. Einige der Leistungsmerkmale der Bibliothek sind:

- **Parallelität:** http-Library unterstützt parallele HTTP-Anfragen, was die Leistung verbessert und die Antwortzeiten verkürzt.
- **Caching:** Die Bibliothek unterstützt das Zwischenspeichern von HTTP-Anfragen, was die Leistung verbessert und den Netzwerkverkehr verringert.
- **SSL/TLS-Unterstützung:** Die Bibliothek unterstützt SSL/TLS-Verschlüsselung, um sicherzustellen, dass HTTP-Anfragen sicher und geschützt sind.

Intl-Library:

Im Zuge dieser Diplomarbeit wurde das Intl-Library verwendet um den „Epoch-Timestamp“ der von dem Web-Server übergeben wird in ein weiterzuverwendendes Datums-Format zu konvertieren. Jedoch sind dies nicht die einzigen Funktionen die dieses Library bietet.

Intl ist eine Bibliothek für Flutter, die Entwicklern ermöglicht, Anwendungen einfach zu internationalisieren. Die Bibliothek bietet Unterstützung für Textübersetzungen, Datums- und Zeitformatierung sowie Währungsformatierung. Sie bietet Entwicklern eine einfache API, um die Sprache und Region der Benutzeroberfläche von Anwendungen dynamisch zu ändern. Die Bibliothek bietet Unterstützung für Textübersetzungen, die

es Entwicklern ermöglichen, Anwendungen in verschiedene Sprachen zu übersetzen. Entwickler können Textstrings in ihren Anwendungen markieren und dann Übersetzungen in verschiedenen Sprachen bereitstellen. Die Bibliothek unterstützt eine Vielzahl von Sprachen und bietet eine einfache API, um Textstrings in verschiedenen Sprachen abzurufen. Intl bietet auch Unterstützung für Währungsformatierung. Entwickler können Währungsbeträge in verschiedenen Ländern und Regionen formatieren. Die Bibliothek bietet eine Vielzahl von Optionen, um Währungsbeträge in verschiedenen Formaten darzustellen.

Material-Library:

Das Flutter-Package "flutter/material" ist ein wichtiger Bestandteil des Flutter-Frameworks, das eine Vielzahl von Material-Design-Widgets für die Entwicklung von ansprechenden und interaktiven Benutzeroberflächen bereitstellt. Das flutter/material Package ist ein Teil des Flutter-Frameworks, das eine Vielzahl von Material-Design-Widgets zur Verfügung stellt. Es ist Teil des Material-Design-Konzepts von Google und bietet Entwicklern eine umfangreiche Sammlung von Widgets und Design-Elementen, um ansprechende und benutzerfreundliche Anwendungen zu erstellen. Material Design ist ein Designkonzept von Google, das darauf abzielt, Benutzererfahrungen zu verbessern, indem es klare visuelle Elemente und Interaktionen bietet. Material Design ist eine einfache, intuitive und konsistente Methode zur Gestaltung von Benutzeroberflächen, die Entwicklern dabei hilft, Anwendungen schnell und effektiv zu erstellen.

Das flutter/material Package stellt eine Vielzahl von Widgets zur Verfügung, die auf das Material-Design-Konzept abgestimmt sind. Diese Widgets sind in der Regel einfach zu verwenden und bieten eine breite Palette von Anpassungsmöglichkeiten, um Benutzeroberflächen anzupassen und individuell zu gestalten. Einige der wichtigsten Material-Design-Widgets, die das flutter/material Package zur Verfügung stellt, sind:

- Buttons
- Textfelder
- Schalter
- Dropdown-Buttons
- Snackbar
- Dialoge

- Navigationselemente

Material-Design-Widgets sind in der Regel einfach zu verwenden und bieten eine breite Palette von Anpassungsmöglichkeiten, um Benutzeroberflächen anzupassen und individuell zu gestalten.

Das flutter/material Package bietet auch eine Vielzahl von Layout-Widgets, die es Entwicklern ermöglichen, ansprechende und gut strukturierte Benutzeroberflächen zu erstellen. Einige der wichtigsten Layout-Widgets, die das Package zur Verfügung stellt, sind:

- Row: Eine Row ist ein Layout-Widget, das Widgets horizontal anordnet. Widgets innerhalb einer Row werden normalerweise gleichmäßig auf die verfügbare Breite verteilt.
- Column: Eine Column ist ein Layout-Widget, das Widgets vertikal anordnet. Widgets innerhalb einer Column werden normalerweise gleichmäßig auf die verfügbare Höhe verteilt.
- Card: Eine Card ist ein Material-Design-Widget, das einen visuellen Container darstellt, der Informationen enthält. Karten können verwendet werden, um Informationen klar und einfach darzustellen
- AppBar: Eine AppBar ist ein Material-Design-Widget, das eine Navigationsleiste am oberen Rand der Anwendung darstellt. Eine AppBar kann verwendet werden, um Navigationsfunktionen bereitzustellen und den Benutzern eine einfache Möglichkeit zu geben, durch Anwendungen zu navigieren.
- Scaffold: Ein Scaffold ist ein Material-Design-Widget, das die Grundlage für die meisten Flutter-Anwendungen bildet. Ein Scaffold stellt eine Standardstruktur für eine Anwendung bereit, einschließlich einer AppBar, einem Navigationsschubladenmenü und einem Bereich zum Anzeigen von Inhalten. Entwickler können das Scaffold-Widget verwenden, um schnell und einfach eine Grundstruktur für ihre Anwendung zu erstellen und diese anzupassen.

Natürlich gibt es noch unzählige weitere Widgets aber diese alle anzuführen würde den Rahmen dieser Diplomarbeit sprengen.

3.6 C:

Die Programmiersprache C hat sich schnell als eine ideale Wahl für die Betriebssystem-Entwicklung etabliert. Ihre Flexibilität und Effizienz erlaubten es den Entwicklern, auf niedriger Ebene zu programmieren und gleichzeitig leicht verständlichen und wartbaren Code zu schreiben. C hat auch in eingebetteten Systemen, wie Mikrocontrollern und hardwarenahen Anwendungen, eine wichtige Rolle gespielt. Aufgrund ihrer Fähigkeit, auf niedriger Ebene zu arbeiten und die Hardware effizient zu nutzen, ist die Sprache C besonders gut geeignet für die Programmierung von eingebetteten Systemen, in denen Ressourcen wie Speicher und Rechenleistung knapp sind. Der erste offizielle Standard wurde 1989 als ANSI C verabschiedet und später im Jahr 1990 als ISO/IEC 9899:1990 international anerkannt. Seit der ersten Standardisierung wurde die Programmiersprache C weiterentwickelt und hat mehrere Revisionen und Erweiterungen erfahren, wie etwa C99, C11 und C18. Diese Revisionen führten zu Verbesserungen in der Sprache, wie etwa die Einführung von neuen Datentypen, verbesserte Typsicherheit und die Integration von Funktionen zur parallelen Programmierung. Darüber hinaus hat C eine Reihe von verwandten Sprachen hervorgebracht, wie zum Beispiel C++, Objective-C und C#. Diese Sprachen haben ihre eigene Identität und Verwendungszwecke, sind aber in vielerlei Hinsicht mit der ursprünglichen Sprache C verwandt und bauen auf ihrem Erbe auf. Die Programmiersprache C hat einen tiefgreifenden Einfluss auf die Informatik gehabt. Sie hat viele moderne Programmiersprachen beeinflusst und war ein entscheidender Faktor für den Erfolg des UNIX-Betriebssystems. Darüber hinaus hat sie die Art und Weise, wie Betriebssysteme und Software entwickelt werden, nachhaltig geprägt. Obwohl in den letzten Jahrzehnten viele neue Programmiersprachen entstanden sind, bleibt C weiterhin relevant und wird in zahlreichen Anwendungsbereichen eingesetzt. [9] [10]

3.7 Verwendete Libraries in C:

3.7.1 ESP-SDK:

Das ESP-SDK ist ein Software Development Kit, das die Entwicklung von Anwendungen für die ESP-Serie von Mikrocontrollern in der Programmiersprache C ermöglicht. Diese Mikrocontroller sind aufgrund ihrer drahtlosen Kommunikationsfähigkeiten und ihres niedrigen Energieverbrauchs ideal für das Internet der Dinge (IoT) geeignet. Das ESP-SDK bietet eine umfassende Sammlung von Softwarebibliotheken und -tools, die die

Entwicklung von IoT-Anwendungen in C erleichtern. Zu den wichtigsten Funktionen gehören:

- Unterstützung für drahtlose Kommunikationsprotokolle, wie Wi-Fi und Bluetooth
- Integration von Netzwerkprotokollen wie HTTP, MQTT und CoAP
- Bibliotheken zur Steuerung von Hardwarekomponenten wie Sensoren und Aktoren
- Unterstützung für Echtzeitanwendungen und Multitasking

<esp_http_server.h>:

Die <esp_http_server> - Bibliothek unterstützt das Hypertext Transfer Protocol (HTTP), ein Anwendungsprotokoll, das die Grundlage für Datenkommunikation im World Wide Web bildet. Die Bibliothek ermöglicht es Entwicklern, HTTP-Server und -Clients zu erstellen, die GET-, POST-, PUT- und DELETE-Anfragen verarbeiten können. Die <esp_http_server>-Bibliothek bietet eine einfache, aber leistungsfähige URL-Routing. Funktionalität, mit der Entwickler benutzerdefinierte Handler-Funktionen für verschiedene Routen und HTTP-Anfragemethoden definieren können. Diese Handler-Funktionen ermöglichen es, spezifische Aktionen auszuführen, wenn eine bestimmte URL angefordert wird. Beispielsweise kann ein Entwickler eine Funktion definieren, die die Temperatur eines Sensors ausliest und an den Client zurücksendet, wenn eine bestimmte URL aufgerufen wird. Die <esp_http_server>-Bibliothek unterstützt asynchrone Verarbeitung, sodass mehrere HTTP-Anfragen gleichzeitig und unabhängig voneinander bearbeitet werden können. Dies ist besonders nützlich, um die Leistung und Reaktionsfähigkeit von IoT-Anwendungen zu verbessern, bei denen die Netzwerklatenz eine Rolle spielen kann. Die <esp_http_server>-Bibliothek ermöglicht die einfache Übertragung von Daten zwischen IoT-Geräten und entfernten Servern oder Clients. Dies ist besonders nützlich für die Fernüberwachung von Sensoren und Aktoren in verschiedenen Anwendungsbereichen. Die Verwendung der <esp_http_server>-Bibliothek ermöglicht es Entwicklern, ihre IoT-Geräte nahtlos in Cloud-Dienste wie AWS, Google Cloud Platform und Microsoft Azure zu integrieren. Durch die Implementierung von HTTP-Servern und -Clients auf den ESP-Mikrocontrollern können Entwickler Daten senden und empfangen, umfangreiche Analysen durchführen und leistungsfähige IoT-Anwendungen entwickeln. Eine der fortgeschrittenen Funktionen dieser Bibliothek ist die Unterstützung für WebSockets, die bidirektionale Echtzeitkommunikation zwischen

Servern und Clients ermöglicht. Die `<esp_http_server>`-Bibliothek bietet eine Reihe von Funktionen, um die Kommunikation über WebSockets zu ermöglichen, einschließlich:

- **WebSocket-Handshake:** Die Bibliothek stellt Funktionen bereit, um den WebSocket-Handshake-Prozess durchzuführen, bei dem der Server und der Client die Kommunikation über WebSockets initiieren und aushandeln.
- **Senden und Empfangen von WebSocket-Nachrichten:** Die `<esp_http_server>`-Bibliothek ermöglicht das Senden und Empfangen von WebSocket-Nachrichten in Text- oder Binärformat, sowohl für den Server als auch für den Client.
- **Ereignis-Handling:** Die Bibliothek bietet eine ereignisgesteuerte Architektur, um auf WebSocket-bezogene Ereignisse wie Verbindungsänderungen, Nachrichtenempfang und Fehler zu reagieren.
- **Verwaltung mehrerer WebSocket-Verbindungen:** Die `<esp_http_server>`-Bibliothek unterstützt die Verwaltung mehrerer gleichzeitiger WebSocket-Verbindungen, was für Anwendungen mit mehreren Clients wichtig ist.

Natürlich werden auch REST-APIs unterstützt.

REST-API-Konzepte:

Die REST-API (Representational State Transfer Application Programming Interface) ist ein Architekturstil, der auf Prinzipien wie Stateless, Cacheable, Client-Server, Layered System und Code on Demand basiert. Sie verwendet Standard-HTTP-Methoden (GET, POST, PUT, DELETE usw.) für den Datenaustausch und ermöglicht die einfache Integration und Interoperabilität zwischen verschiedenen Systemen und Anwendungen. Schlüsselkomponenten der REST-API in der `<esp_http_server>`-Bibliothek:

- **HTTP-Server:** Der HTTP-Server ist das grundlegende Element der Bibliothek und ermöglicht die Verarbeitung von HTTP-Anfragen und -Antworten. Er unterstützt sowohl IPv4 als auch IPv6 und kann mehrere gleichzeitige Verbindungen verwalten. Die Bibliothek ermöglicht es Entwicklern, benutzerdefinierte Handler für verschiedene Routen und HTTP-Methoden zu erstellen, um die gewünschte Funktionalität bereitzustellen.
- **URI-Handler:** Die URI-Handler sind Funktionen, die auf bestimmte URIs und HTTP-Methoden reagieren und die Anfragen entsprechend verarbeiten. Die `<esp_http_server>`-Bibliothek ermöglicht die einfache Registrierung und Entfernung von URI-Handletern, um die Verarbeitung von HTTP-Anfragen an die entsprechenden Ressourcen zuzuweisen.

- **Middleware:** Middleware-Komponenten ermöglichen die Erweiterung der Funktionalität des HTTP-Servers durch die Implementierung von Zusatzfunktionen wie Authentifizierung, CORS (Cross-Origin Resource Sharing), Content-Encoding oder Logging. Diese Komponenten können in den Verarbeitungsprozess der HTTP-Anfrage oder -Antwort eingefügt werden, um zusätzliche Kontrolle und Sicherheit zu gewährleisten.
- **JSON-Unterstützung:** Die Bibliothek bietet native JSON-Unterstützung, die es Entwicklern ermöglicht, die Kommunikation zwischen Clients und Servern in einem standardisierten Format zu erleichtern. Dies erleichtert die Verarbeitung und den Austausch von Daten zwischen verschiedenen Systemen und Anwendungen und fördert die Plattformunabhängigkeit.

<esp_wifi.h>:

Die <esp_wifi.h>-Bibliothek ist eine wichtige Komponente des ESP-SDK und ermöglicht die Programmierung von Wi-Fi-Funktionen für ESP-Mikrocontroller in der Programmiersprache C. Die <esp_wifi.h>-Bibliothek bietet eine umfangreiche Sammlung von Funktionen zur Implementierung von Wi-Fi-Kommunikation auf ESP-Mikrocontrollern. Zu den wichtigsten Funktionen gehören:

- Station (STA) und Access Point (AP) Modi: Die Bibliothek ermöglicht die Konfiguration von ESP-Mikrocontrollern als Wi-Fi-Stationen oder Access Points, um Verbindungen mit anderen Wi-Fi-Geräten herzustellen oder bereitzustellen.
- Wi-Fi-Scan: Die <esp_wifi.h>-Bibliothek stellt Funktionen zur Verfügung, um nach verfügbaren Wi-Fi-Netzwerken in der Umgebung zu suchen und Informationen über sie zu erhalten.
- Wi-Fi-Verbindung und -Trennung: Die Bibliothek bietet Funktionen zum Herstellen und Trennen von Wi-Fi-Verbindungen, einschließlich der Verwaltung von SSIDs, Passwörtern und Sicherheitseinstellungen.

Die <esp_wifi.h>-Bibliothek ermöglicht die Entwicklung einer Vielzahl von Wi-Fi-basierten Anwendungen für IoT-Geräte, einschließlich:

- Smart-Home-Anwendungen: Die Bibliothek ermöglicht die Implementierung von Wi-Fi-Kommunikation für Smart-Home-Geräte wie Smart-Lampen, Thermostate und Überwachungskameras.
- Fernüberwachung und -steuerung: Durch die Verwendung der Wi-Fi-Funktionalitäten können Entwickler IoT-Geräte erstellen, die über das Internet überwacht und gesteuert werden können.
- Integration in Cloud-Dienste.

SPIFFS:

Selbst wenn der ESP32 eine beträchtliche Menge an Flash-Speicher hat, ist dieser immer noch begrenzt und nicht so groß wie der Speicher auf einem typischen Computer. Deshalb müssen Entwickler oft spezielle Dateisysteme wie SPIFFS verwenden, um auf dem ESP32 effektiv Dateien zu speichern und zu verwalten. SPIFFS wurde speziell für den Einsatz auf dem ESP32 entwickelt und bietet eine effektive Möglichkeit, Daten auf dem Flash-Speicher des ESP32 zu organisieren und zu speichern. Das System verwendet einen Puffermechanismus, um die Daten effizient zwischen dem Flash-Speicher und dem Prozessor zu übertragen, was schnelle Zugriffszeiten ermöglicht. Darüber hinaus teilt das Dateisystem jede Datei in mehrere Blöcke auf und speichert sie an verschiedenen Stellen auf dem Flash-Speicher. Dies sorgt dafür, dass das Dateisystem eine hohe Robustheit aufweist und Daten, die aufgrund von Systemabstürzen oder unerwarteten Stromausfällen beschädigt wurden, wiederherstellen kann. Das SPIFFS-Dateisystem bietet auch eine einfache API, die es Entwicklern ermöglicht, schnell auf Dateien zuzugreifen und sie zu verwalten. Das System unterstützt grundlegende Dateisystemfunktionen wie das Öffnen, Schreiben, Lesen und Schließen von Dateien. Darüber hinaus können Entwickler auch benutzerdefinierte Dateiformate definieren, um Daten auf effiziente Weise zu speichern und abzurufen. Ein weiterer Vorteil von SPIFFS ist, dass es sehr platzsparend ist. Das Dateisystem ist so optimiert, dass es sehr wenig Speicherplatz benötigt und somit mehr Platz für Anwendungsdaten zur Verfügung steht. Darüber hinaus ist es einfach zu implementieren und erfordert keine spezielle Hardware oder zusätzliche Komponenten. Allerdings hat SPIFFS auch einige Nachteile. Es ist beispielsweise nicht so skalierbar wie andere Dateisysteme wie FAT oder NTFS. Das bedeutet, dass es nicht so gut für Anwendungen geeignet ist, die viele Dateien oder sehr große Dateien speichern müssen. Außerdem ist das Dateisystem nicht so flexibel wie andere Dateisysteme und es kann schwierig sein, benutzerdefinierte Funktionen zu implementieren. SPIFFS ist eine ausgezeichnete Wahl für Anwendungen auf dem ESP32-Modul, die begrenzten Speicherplatz haben und schnellen Zugriff auf Dateien benötigen. Es bietet eine effiziente Möglichkeit, Daten zu speichern und abzurufen, ist robust und einfach zu verwenden. [11]

4 Technologien|Entwicklungsumgebung

4.1 Visual Studio Code:

Visual Studio Code (VSCode) ist eine plattformübergreifende Entwicklungsumgebung (IDE), die von Microsoft entwickelt wurde. Mit einer breiten Palette von Funktionen und einer hohen Anpassungsfähigkeit ermöglicht VSCode die Programmierung in verschiedenen Programmiersprachen und unterstützt Entwickler durch zahlreiche Erweiterungen wie in diesem Projekt verwendeten Flutter- und Espressif IDF-Erweiterung. Visual Studio Code bietet Entwicklern eine einheitliche und benutzerfreundliche Schnittstelle, um Code in einer Vielzahl von Programmiersprachen zu schreiben und zu bearbeiten. Die IDE unterstützt Syntaxhervorhebung, Code-Navigation, Debugging und Versionskontrolle. Eine der Stärken von Visual Studio Code ist das Erweiterungssystem, das es ermöglicht, den Funktionsumfang der IDE durch den Einsatz von Erweiterungen anzupassen und zu erweitern. Diese Erweiterungen werden von Microsoft, der Community oder Drittanbietern entwickelt und bieten zusätzliche Funktionen, die auf bestimmte Programmiersprachen, Frameworks oder Plattformen zugeschnitten sind. Eine solche Erweiterung ist die Flutter-Extension, die die Entwicklung von plattformübergreifenden mobilen, Web- und Desktop-Anwendungen mit dem Flutter-Framework von Google ermöglicht. Die Flutter-Extension bietet Funktionen wie Autovervollständigung, Syntaxhervorhebung, Fehlererkennung und Integration von Flutter- und Dart-SDKs. Darüber hinaus unterstützt die Erweiterung das Hot-Reload-Feature von Flutter, das es ermöglicht, Änderungen am Code in Echtzeit anzuzeigen, ohne die Anwendung neu starten zu müssen. Eine weitere nützliche Erweiterung ist die Espressif IDF-Extension, die die Entwicklung von Anwendungen für ESP32- und ESP-IDF-basierte Geräte erleichtert. Diese Erweiterung bietet Integration mit dem Espressif IoT Development Framework (IDF) und unterstützt Funktionen wie Codevervollständigung, Projektgenerierung, Debugging und Flashing von Firmware auf ESP32-Geräte. Dazu kommt die automatische Erkennung von angeschlossenen ESP-Geräten an den Computer die separat ausgewählt werden können.

5 Umsetzung

5.1 Umsetzung:

Frontend:



Der Home-Screen basiert auf einem Scroll-View, welches den Vorteil bietet ein einheitliches Design auf jeder Art von Bildschirmgröße zu gewährleisten, ohne die Lesbarkeit zu beeinträchtigen. Ein ScrollView in Flutter ist ein Widget, das verwendet wird, um eine Liste von Elementen anzuzeigen, die größer sind als der verfügbare Bildschirm. Es ermöglicht dem Benutzer, durch die Liste zu scrollen, um alle Elemente anzuzeigen. Es gibt verschiedene Arten von ScrollView in Flutter, darunter ScrollView, ListView, GridView und CustomScrollView. Das ScrollView enthält generisch selbst erstellte Objekte namens DataCard.

Abbildung 6: StartpageBild

DataCards:

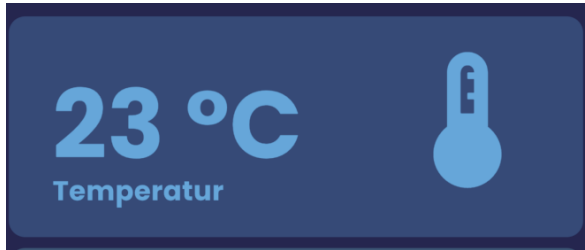


Abbildung 7: DataCard

Das DataCard Widget in Flutter ist ein vorgefertigtes Material Design-Widget, das zur Darstellung von Informationen in einem Kartenformat verwendet wird. Es besteht aus einer rechteckigen Box mit abgerundeten Ecken, die in der Regel eine Hintergrundfarbe, einen Titel, eine Beschreibung und optional eine Aktion oder einen Button enthält. Das DataCard Widget enthält verschiedene Eigenschaften, die angepasst werden können, um das Erscheinungsbild und das Verhalten der Karte zu steuern. Dazu gehören Eigenschaften wie Hintergrundfarbe, Ränder, Schatten, Größe, Padding und Ausrichtung.

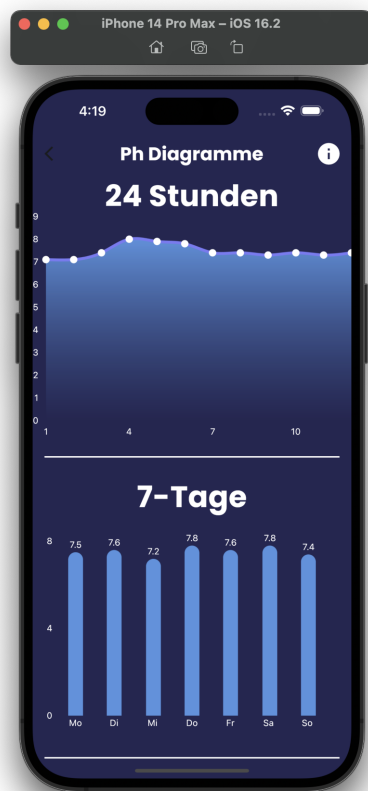
DataCards verfügen über die Funktion „onPressed“ in der man angeben kann was geschieht, wenn man eine der 5 DataCards drückt. Wenn man eine DataCard die den Namen „Temperatur“, „Ph-Wert“ oder „Trübung“ drückt wird man an eine weitere Page der App weiter geleitet in der man Statistiken zu dem jeweiligen Wert erhält. Wenn man die DataCard mit dem Namen „Alarm“ drückt wird der Alarm den man erhält wenn verdächtige Bewegungen des Gyroskopssensors wahrgenommen werden aktiviert oder deaktiviert. Die DataCards auf dem HomeScreen enthalten ansonsten den zuletzt gemessenen Wert, der durch einen WebSocket vom Backend bereitgestellt wird und stellen diesen dar. Die DataCards sind generisch erstellbar mit Übergabewerten, was eine einfache Vergrößerung des Funktionsumfangs des HomeScreens ermöglicht. Die Icons die auf den DataCards zu sehen sind, werden von Google-Material-Icons bereitgestellt und werden mittels API in die App Ressourceneffizient implementiert.



Abbildung 8: AlarmOFF

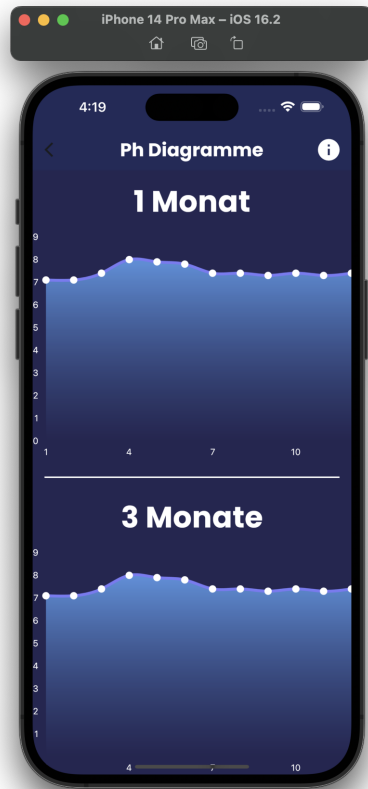
So sieht die DataCard aus wenn man den Alarm durch drücken des Feldes deaktiviert.

Grafische Darstellung der Werte über Zeit:



Hier sind die Diagramme dargestellt auf die man durch das klicken der DataCards weiter geleitet wird. Sie werden durch die Libraries "charts_flutter und Byncfusion_flutter_charts" bereitgestellt und mittels HTTP-Request befüllt und bei aufruf aktualisiert.

Abbildung 9: Diagramme1Bild



Nach dem Freunde Verwandte und Bekannte mittels mündlicher Umfrage befragt wurden, ist beschlossen worden, dass Diagramme in den Zeiträumen "24 Stunden", "7 Tage", "1 Monat" und "3 Monate" angezeigt werden sollen. Diese Zeiträume waren am beliebtesten und sind wenn man die durchschnittliche Dauer einer Badesaison betrachtet am relevantesten für den Benutzer dieser Applikation.

Abbildung 10: Diagramme2Bild

Designkonzept:

Bei dem Design wurde ein sehr schlichtes und leicht zu verstehendes Material-Design gewählt. Es basiert auf der Schriftart "Poppins-Bold" und macht die App sehr anschaulich. Die Graphen werden bei Aufruf dynamisch mit den Daten aus der Datenbank befüllt und aktualisiert und dieser Ablauf durch eine Animation in der sich der Graph langsam aufbaut überbrückt. Die verwendeten Icons werden von Google-Material-Icons bereitgestellt und sind nach Belieben veränderbar. Das Design ist dynamisch und passt sich jeder Bildschirmgröße, Auflösung und Bildschirmverhältnis an.

Da der die Sinus-Graphen jedem Tag nur einen durchschnittlichen Wert zuordnen um somit die Darstellung übersichtlicher zu gestalten wird davor für jeden Tag ein Durchschnittswert berechnet. Da das Backend um Bits zu sparen nur einen Epoch-Timestamp übergibt muss dies auch vorher aufbereitet werden.

```
Center(
  child: SfCartesianChart(
    margin: const EdgeInsets.all(0),
    borderWidth: 0,
    borderColor: Colors.transparent,
    plotAreaBorderWidth: 0,
    primaryXAxis: CategoryAxis(
      labelStyle: const TextStyle(color: Colors.white),
      majorGridLines: const MajorGridLines(
        color: Colors.transparent,
      ), // MajorGridLines
      majorTickLines: const MajorTickLines(
        color: Colors.transparent,
      ), // MajorTickLines
      axisLine: const AxisLine(
        color: Colors.transparent,
      ), // AxisLine
      isVisible: true,
      borderColor: Colors.transparent,
      interval: 1,
    ), // CategoryAxis
    primaryYAxis: NumericAxis(
      labelStyle: const TextStyle(color: Colors.white),
      majorGridLines: const MajorGridLines(
        color: Colors.transparent,
      ), // MajorGridLines
      majorTickLines: const MajorTickLines(
        color: Colors.transparent,
      ), // MajorTickLines
      axisLine: const AxisLine(
        color: Colors.transparent,
      ), // AxisLine
      minimum: 0,
      maximum: 35,
      isVisible: true,
      borderColor: Colors.transparent,
    ), // NumericAxis
    series: <ChartSeries<TemperatureData, String>>[
      SplineAreaSeries(
        dataSource: data,
        xValueMapper: (TemperatureData data, _) => data.day,
        yValueMapper: (TemperatureData data, _) => data.temperature,
        splineType: SplineType.natural,
      ),
    ],
  ),
)
```

Dieses Code-Snippet zeigt einen kleinen Einblick wie man einen Graphen designed und mit seinen dazugehörigen Daten befüllt. Ein Graph braucht eine bestimmte Art von Liste durch die er befüllt wird. Die Daten die sich in dieser Liste befinden werde vorher von der Logik durch die HTTP-Request abgefragt und für jeden Graphen einzeln aufbereitet um die Daten anschaulich darzustellen.

Abbildung 11: CodeSnippetChartDesign

Umsetzung Backend:

Generelle Funktionsweise:

Die generelle Funktionsweise des Backends basiert auf zwei TTGO-ESP32-LoRa Mikroprozessoren. Einer dieser Mikrochips(Sender) hat 4 verbaute Sensoren darunter:

- Temperatursensor
- Ph-Sensor
- NTU-Trübungssensor
- Beschleunigungssensor

Die Sensoren wurden über die vom ESP32 zur Verfügung gestellten Pins angelötet. Der Trübungssensor wurde bei einem ADC-Pin (Analog-Digital-Converter) angelötet, ebenso wie der Ph-Sensor, der Temperatursensor ist bei einem GPIO (General Purpose Input/Output) angebracht und der Beschleunigungssensor bei einem SDA(serial data) und dem SCL(serial clock) Anschluss. Der ESP32(Sender) wird alle 3 Stunden aus seinem Sleep-Zustand aufgeweckt und nimmt eine Messung der Daten vor. Er befindet

sich während er inaktiv ist im Sleep damit er noch Stromsparender agiert da er nicht an einer Steckdose angebracht ist sondern mit einer externen Batterie betrieben wird. Diese Daten werden direkt nach der Messung an den anderen ESP32(Reciever) gesendet. Die Daten werden nicht auf dem im Wasser schwimmenden Sender gespeichert da dieser äußerst stromsparend vorgehen soll. Die Messwerte werden mittels LoRa-Funk-System welches lokal an dem ESP32 verbaut ist an den Reciever gesendet und dort bei Erhalt mit einem Epoch-Timestamp versehen und in das SPIFF-File-System gespeichert. Das äußerst kleine Speichersystem ermöglicht eine Speicherung der Daten in einem Intervall von allen 3 Stunden über 1 Jahr durchgehend. Die Daten werden mittels Web-Sockets(für die Live-Daten-Aktualisierung) bereitgestellt und mit einem http-Request der alle Daten an das Frontend übergibt falls das Fenster mit den Zeit-Diagrammen aufgerufen wird um sie in die jeweiligen Diagramme einzufügen. Die dazu benötigten Libraries um dies alles in C umzusetzen sind unter „Technologien| Software“ zu finden.

Die Daten die für eine Routine-Übertragung alle 3 Stunden nötig sind, lauten:

- Datum
- Temperatur
- Ph-Wert
- NTU-Wert

Die Daten des Beschleunigungssensors werden nicht übertragen, um Speicherplatz zu sparen, die Aktualisierungsgeschwindigkeit zu erhöhen und eine Echt-Zeit-Benachrichtigung zu ermöglichen, sondern der Sender sendet eine extra Benachrichtigung falls der Beschleunigungssensor Werte erreicht hat die auf verdächtige Aktivitäten im Wasser hinweisen. Der (Reciever) ESP32 ist mit dem W-Lan verbunden und permanent eingeschaltet um die Daten 24 Stunden 7 Tage in der Woche zu erhalten zu verarbeiten und für das Frontend für eine http-Request bereitzustellen.

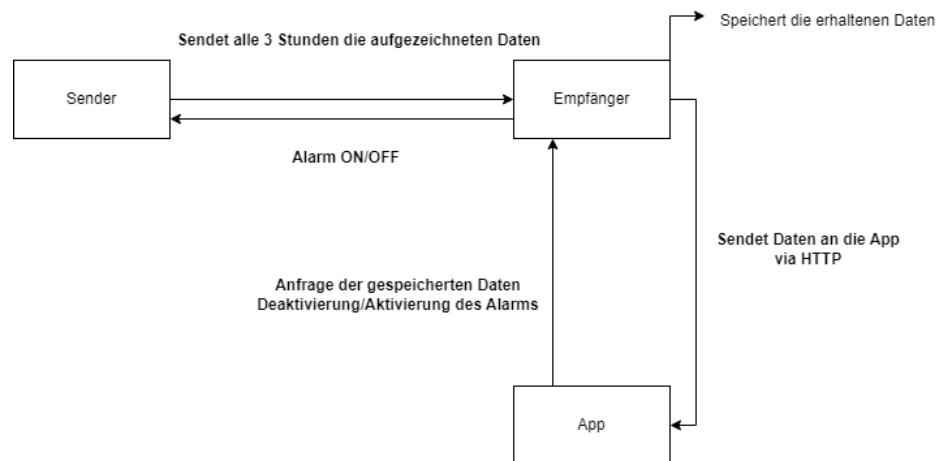


Abbildung 12: Funktionsweise der Datenübertragung

Funktion von LoRa in diesem Projekt:

Die Kommunikationstechnologie, die auf der Basis von Spread-Spectrum-Techniken entwickelt wurde ermöglicht eine drahtlose Kommunikation über große Entfernungen bei niedriger Leistungsaufnahme. In deinem Fall werden die Daten von drei Sensoren (Temperatursensor, pH-Sensor und NTU-Trübungssensor) mithilfe eines ESP32-Mikrocontrollers und eines LoRa-Transceivers an einen zweiten ESP32, der als Empfänger fungiert, übertragen.

Hier ist das Grundkonzept, wie LoRa funktioniert und die Daten überträgt:

- **Sensoren erfassen Daten:** Die drei Sensoren (Temperatur, pH und NTU-Trübung) erfassen Umgebungsdaten und übermitteln diese an den ESP32-Mikrocontroller.
- **Datenverarbeitung:** Der ESP32-Mikrocontroller verarbeitet die empfangenen Daten, konvertiert sie in ein geeignetes Format und erstellt ein Datenpaket. Dieses Paket enthält die Sensordaten.
- **LoRa-Modulation:** Der LoRa-Transceiver, der am ESP32 angeschlossen ist, wandelt das Datenpaket in ein Funksignal um, das für die Übertragung über das LoRa-Protokoll geeignet ist. LoRa verwendet eine Chirp Spread Spectrum (CSS)-Technologie, bei der die Frequenz des Funksignals kontinuierlich über eine bestimmte Bandbreite variiert. Diese Technik erhöht die Störfestigkeit und ermöglicht eine effiziente Nutzung des Funkspektrums.
- **Datenübertragung:** Das modulierte Funksignal wird über die LoRa-Antenne ausgesendet und kann über große Entfernungen übertragen werden. LoRa ist besonders für Anwendungen mit geringem Energieverbrauch und geringer Datenrate geeignet.
- **Empfang des Funksignals:** Der zweite ESP32, der als Empfänger fungiert, ist ebenfalls mit einem LoRa-Transceiver und einer Antenne ausgestattet. Dieser empfängt das Funksignal, das von der Senderantenne ausgesendet wurde.
- **Demodulation und Datenextraktion:** Der LoRa-Transceiver am Empfänger demoduliert das empfangene Funksignal und extrahiert das ursprüngliche Datenpaket. Dabei wird die Spread-Spectrum-Technik rückgängig gemacht, um die übertragenen Informationen zurückzugewinnen.

- Datenverarbeitung: Der empfangende ESP32-Mikrocontroller verarbeitet das extrahierte Datenpaket, um die Daten der einzelnen Sensoren (Temperatur, pH und NTU-Trübung) zu erhalten. Anschließend werden die Daten dann für die Zugriffe des Frontends vorbereitet.

Reciever:

Listing 1: Reciever sendet von LoRa erhaltene Daten an Clients

```

1  for (;;) {
2      lora_receive();
3
4      // CURRENT_EPOCH - TIMESTAMP
5      // 0, 1, 2, 3, ....
6      while (lora_received()) {
7          x = lora_receive_packet(buf, sizeof(buf));
8          buf[x] = 0;
9          char *newB = (char*)buf;
10         char str[256];
11         sprintf(str, "%lld;", asd + time(0));
12
13         prepend(newB, str);
14
15         printf("rec: %s\n", newB);
16
17         // timestamp;ph;ntu;temp
18
19         fprintf(fp, "%s\n", newB);
20         fflush(fp);
21
22         size_t clients = max_clients;
23         int client_fds[max_clients];
24
25         if (httpd_get_client_list(server, &clients, client_fds) == ESP_OK) {
26             for (size_t i = 0; i < clients; ++i) {
27                 int sock = client_fds[i];
28
29                 if (httpd_ws_get_fd_info(server, sock) ==
30                     HTTPD_WS_CLIENT_WEBSOCKET) {
31                     ESP_LOGI(TAG, "Active client (fd=%d) -> sending async
32                         message", sock);
33
34                     struct async_resp_arg *resp_arg = malloc(sizeof(struct
35                         async_resp_arg));
36                     resp_arg->hd = server;
37                     resp_arg->fd = sock;
38                     resp_arg->buf = (char*)buf;
39
40                     if (httpd_queue_work(resp_arg->hd, send_data, resp_arg) !=
41                         ESP_OK) {
42                         ESP_LOGE(TAG, "httpd_queue_work failed!");
43                         break;
44                     }
45                 }
46             }
47         } else {
48             ESP_LOGE(TAG, "httpd_get_client_list failed!");
49         }
50         lora_receive();
51     }
52     vTaskDelay(pdMS_TO_TICKS(1000));
53 }

```

In diesem Ausschnitt des Reciever-Codes wird `lora_recieve()` gestartet. Dadurch wartet der Empfänger auf ein LoRa-Datenpaket. Wenn ein Datenpaket Empfangen wird, wird die „while-Schleife“ ausgeführt und die Pakete in den Buffer gelesen und ein Epoch-

Timestamp vorne daran gehängt. Danach werden die Daten in den File geschrieben. Dann werden alle Web-Server-Clients abgefragt und die neuen Daten an sie gesendet. Dies wird in der „for-Schleife“ für jeden einzelnen Client wiederholt. Wenn der Client einen Web-Socket ist wird ein anderer Code ausgeführt aber in beiden Fällen werden die Daten an den Client gesendet.

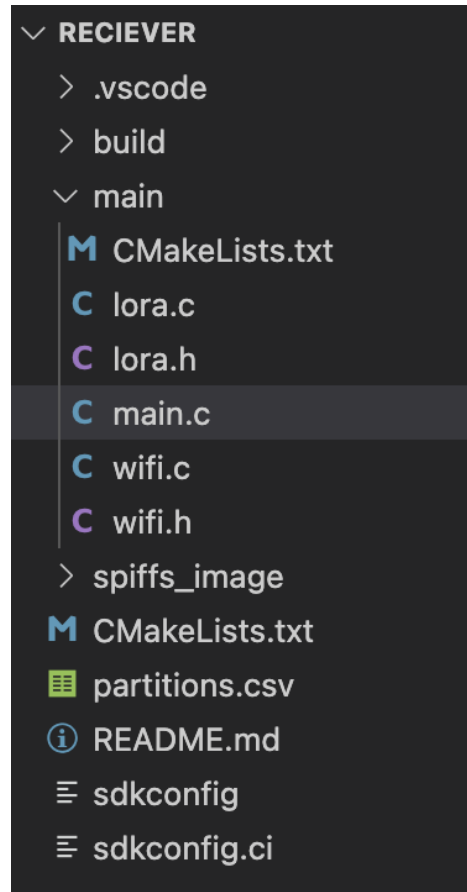


Abbildung 13: Reciever Struktur

In Wifi.c wird das Wifi-Library eingebunden. In Wifi.h werden die Daten des Routers eingetragen(SSID und Passwort), lora.c ist das Lora Library welche beide in main.c benutzt werden.

Sender:

Listing 2: Sender lest die Daten der Sensoren aus und überträgt sie via LoRa

```

1  void
2  sensors_task(void *p)
3  {
4      int raw;
5      float pH = 0, ntu = 0, temp = 0;
6      char str[64];
7
8      for(;;) {
9          ESP_ERROR_CHECK(adc_oneshot_config_channel(adc2_handle,
10              PH_SENSOR_ADC2_CHANNEL, &chan_cfg));
11          ESP_ERROR_CHECK(adc_oneshot_read(adc2_handle, PH_SENSOR_ADC2_CHANNEL,
12              &raw));
13          pH = raw * (5.0 / 1024);
14          pH *= 2;
15
16          esp_err_t err = ds18b20_set_resolution(handle, NULL,
17              DS18B20_RESOLUTION_12B);
18          if (err != ESP_OK) {
19              ESP_LOGI(TAG, "failed to set res");
20          }
21          err = ds18b20_trigger_temperature_conversion(handle, NULL);
22          if (err != ESP_OK) {
23              ESP_LOGI(TAG, "failed to trigger temp conversion");
24          }
25
26          vTaskDelay(pdMS_TO_TICKS(800));
27
28          // 5YSq5pyI44G+44KK44Gq base64
29
30          for (int i = 0; i < device_num; ++i) {
31              float newTemp;
32              err = ds18b20_get_temperature(handle, device_rom_id[i], &newTemp);
33              if (err != ESP_OK) {
34                  continue;
35              }
36
37              if (newTemp != 0) {
38                  temp = newTemp;
39              }
40
41              ESP_ERROR_CHECK(adc_oneshot_config_channel(adc2_handle,
42                  NTU_SENSOR_ADC2_CHANNEL, &chan_cfg));
43              ESP_ERROR_CHECK(adc_oneshot_read(adc2_handle, NTU_SENSOR_ADC2_CHANNEL,
44                  &raw));
45              ntu = raw * (5.0 / 1023);
46              ntu /= 20;
47
48              snprintf(str, 64, "%.2f;%.2f;%.2f", pH, ntu, temp);
49
50              printf("SENT %s\n", str);
51              lora_send_packet((uint8_t*)str, 64);
52
53              vTaskDelay(pdMS_TO_TICKS(DELAY));
54          }
55
56          ESP_ERROR_CHECK(oneshot_del_bus(handle));
57          ESP_LOGI(TAG, "1-wire bus deleted");
58
59          ESP_ERROR_CHECK(adc_oneshot_del_unit(adc2_handle));
60      }

```

In diesem Code-Ausschnitt werden die Sensoren an ihren jeweiligen Pins ausgelesen und der zurückgegebene Referenzwerte in die jeweilige Einheit umgerechnet. Daraufhin werden Sie in einen char-Array(String) geladen und in ein LoRa-Paket gepackt und versendet.

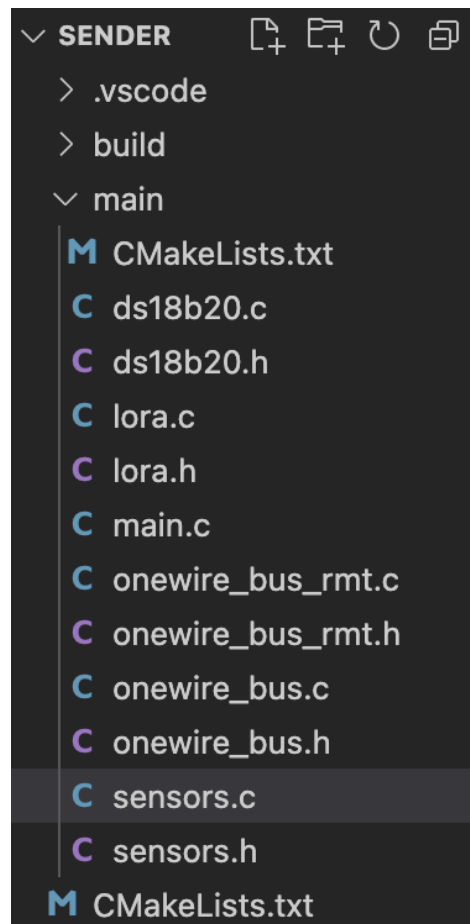


Abbildung 14: Sender Struktur

ds18b20.c und onewire_bus.c sind die Libraries für den Temperatursensor, lora.c das Library für LoRa, in sensors.c befindet sich der Code für alle Sensoren sowie der oben gezeigt Code-Ausschnitt. In main.c wird LoRa und die Sensoren initialisiert.

6 Erkenntnisse

Im Laufe dieser Diplomarbeit wurden sehr viele Erkenntnisse über die Programmiersprachen Dart und C gemacht. Es wurde C im Rahmen der Mikoprozessor-Programmierung kennengelernt und Einblicke in eine Hardware nahe Entwicklung gewonnen, die in der Fachrichtung-Informatik normalerweise nicht üblich sind, der Aufbau neuer Programmiersprachen kennengelernt und ein tiefergehendes Verständnis zur Entwicklung einer Full-Stack-Application wurde entwickelt. Dazu wurde der Umgang mit der Software Latex erlernt. Ebenso wurde bewusst wie wichtig ein Arbeitsplan und Zeit-Management sind um nicht die Zeit aus den Augen zu verlieren und mögliche Probleme die während der Entwicklungszeit auftreten können einplanen zu können.

Hürden

Im Laufe der der Entwicklung eines Projektes gibt es immer wieder Hürden die überwunden werden müssen. Dieses Projekt musste auch einige überwinden wie zum Beispiel defekte Sensoren mit schlechter Qualität aus China, hohe Zoll-Preise und Versandkosten, lange Lieferzeiten bis zu 3 Monaten. Dazu kam, dass eine Funktion (eine KI die verschiedene Wellengänge voneinander unterscheiden kann um keine falschen Alarmer auszulösen) aus den vorher genannten Gründen nicht umgesetzt werden konnte da das Risiko die Sensoren oder die Chips durch Wasserschäden zu beschädigen. Es wurden viele Libraries benötigt um einen Sensor auszulesen und diese mussten zuerst verstanden werden. Zusammenfassend können Probleme große Steine am Weg zum Ziel darstellen.

Literaturverzeichnis

- [1] E. Systems, „ESP32.” Online verfügbar: <https://www.espressif.com/en/products/socs/esp32>
- [2] ———, „ESP32 Technical Reference Manual.” Online verfügbar: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf
- [3] Az-Delivery, „Temperatur-Sensor.” Online verfügbar: https://cdn.shopify.com/s/files/1/1509/1638/files/DS18B20_3mCable_datasheet.pdf?v=1644320674
- [4] e Gizmo, „ph-Sensor.” Online verfügbar: <https://www.e-gizmo.net/oc/kits%20documents/PH%20Sensor%20E-201-C/PH%20Sensor%20E-201-C.pdf>
- [5] AliExpress, „ntu-Sensor.” Online verfügbar: <https://de.aliexpress.com/item/4000321791231.html>
- [6] Joy-It, „Gyroskop/Beschleunigungssensor.” Online verfügbar: <https://joy-it.net/de/products/SEN-MPU6050>
- [7] React, „React.” Online verfügbar: <https://reactjs.org/docs/getting-started.html>
- [8] Flutter, „Flutter.” Online verfügbar: <https://pub.dev/>
- [9] Grund-Wissen.de, „C.” Online verfügbar: <https://www.grund-wissen.de/informatik/c/index.html>
- [10] Wikipedia, „Wikipedia.” Online verfügbar: <https://de.wikipedia.org/>
- [11] Espressif, „SPIFFS.” Online verfügbar: <https://docs.espressif.com/projects/espidf/en/latest/esp32/api-reference/storage/spiffs.html>

Abbildungsverzeichnis

1	ESP32-TTGO-LoRa-Oled	6
2	TemperaturSensor	9
3	Ph-Sensor	10
4	NTU-Sensor	11
5	Beschleunigungssensor	12
6	StartpageBild	32
7	DataCard	33
8	AlarmOFF	33
9	Diagramme1Bild	34
10	Diagramme2Bild	35
11	CodeSnippetChartDesign	36
12	Funktionsweise der Datenübertragung	38
13	Reciever Struktur	41
14	Sender Struktur	43

Quellcodeverzeichnis

1	Reciever sendet von LoRa erhaltene Daten an Clients	40
2	Sender lest die Daten der Sensoren aus und überträgt sie via LoRa . . .	42