

25.11.2022

Modul 216 Projektarbeit Briefkasten



Filip, Oscar, Nikola
NOSER YOUNG AG

Inhaltsverzeichnis

1	Informieren.....	2
1.1	Projektauftrag	2
1.2	Vorgaben	2
1.3	Anforderung	2
1.4	Formale Aspekte.....	2
2	Planen	3
2.1	Aufteilung.....	3
2.2	Systemübersicht	3
2.4	Zeitmanagement.....	4
3	Entscheidung	4
3.1	Sensoren	4
3.1.1	KY-050 Ultraschallabstandssensor.....	4
3.1.2	TEMT6000 Lichtsensor	5
3.2	Ressourcen.....	5
4	Realisierung.....	6
4.1	Vorbereitung	6
4.2	Verkabelung.....	7
4.4	Code	8
4.5	Node-RED	8
5	Kontrolle	10
5.1	Fehler und Probleme auf dem Weg	10
5.2	Black-Box-Testing.....	11
5.3	White-Box-Testing	11
5.4	Testkonzept	11
5.4.1	Testziele.....	11
5.4.2	Testobjekte	11
5.4.3	Testarten.....	11
5.4.4	Testinfrastruktur	12
5.4.5	Testorganisation.....	12
5.4.6	Testplanung	12
5.5	Testdrehbuch.....	13
6	Auswertung.....	17
7	Quellenverzeichnis	18

1 Informieren

1.1 Projektauftrag

Die Noser Young AG besitzt einen Briefkasten, welcher täglich geprüft und geleert werden muss. Es kommt hin und wieder vor, dass der Briefkasten geprüft wird, bevor Post angekommen ist. Manchmal wird der Briefkasten auch gar nicht geleert. In unserer Projektarbeit soll nun ein Gerät entwickelt werden, welcher den Zustand des Briefkastens beobachtet.

1.2 Vorgaben

Folgende Elemente sind zum Realisieren dieser Aufgabe bereits vorhanden und müssen nicht erstellt bzw. beschafft werden:

- ESP32-Board
- Breadboards
- Diverse Sensoren
- Dupont Lines (Patchkabel, Jumper-Kabel)
- Beispielcode für ESP32-Board
- Hilfsdokumentation (Dokument "Getting started" in Microsoft Teams)
- Infrastruktur (WLAN)
- IoE-Plattform (MQTT, Node-RED)

1.3 Anforderung

- Umsetzung gemäss Bewertungsraster (Dokument „Bewertung“ in Microsoft Teams)
- Zeitlicher Verlauf
- Zustand des Briefkastens soll daran erkannt werden, ob die Türe offen oder zu ist
- Möglichkeit zur Benachrichtigung über einen Kommunikationskanal (Email, SMS,..)
- Status bei Änderung an MQTT-Topic „zuerich/briefkasten/[xyz]“ übermitteln
- Ausgabe von Daten über Node-RED (Datum und Uhrzeit, wenn Briefkasten benutzt wird / ob es Post hat & Differenzierung zwischen Post eingelegt und Post geholt)

1.4 Formale Aspekte

- Projektdokumentation
- Testing inkl. Doku
- Lerndokumentation
- Arbeitsaufteilung

- Beurteilungskriterien erfüllen gemäss Liste
- Sensorwahl und -Speisung (Akku, USB, ...)
- Sämtliche Projektunterlagen sind in einem Git-Repository „BLJ2022_uek216_briefkasten“ unterzubringen, dass regelmässig aktualisiert wird. Ausbilder werden in das Repository eingeladen.
- Abgabe der persönlichen Lerndokumentation am Ende des Kurses an die Kursleitung, in elektronischer Form

2 Planen

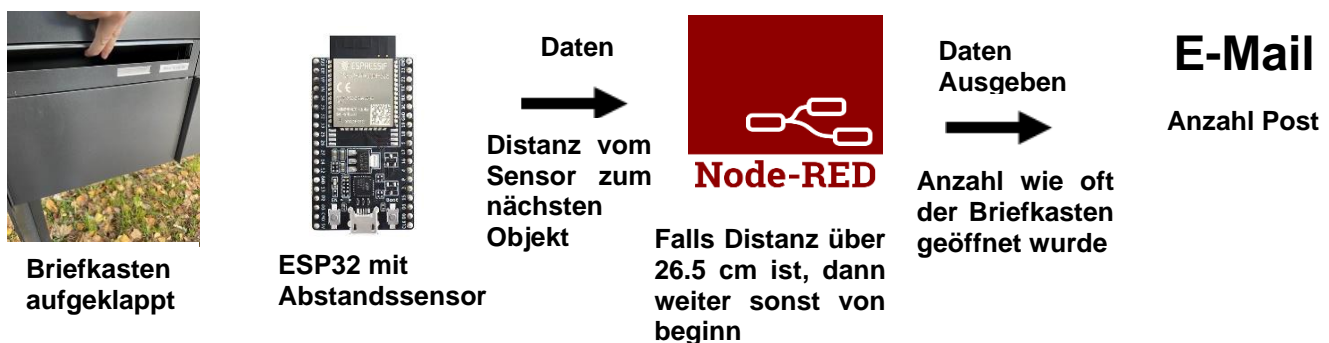
2.1 Aufteilung

Dies sind die folgenden Aufteilungen im Team

Aufgabe	Verantwortlicher
Schwerpunkt Dokumentation schreiben	Filip Franek
Schwerpunkt Verkabeln der Komponenten	Nikola Bogosavljevic
Schwerpunkt Code schreiben	Oscar Doell
Testen und Kontrollieren	Oscar/Nikola/Filip

2.2 Systemübersicht

Hier werden die Komponente und ihre Zusammenhänge gezeigt.



[Quelle](#)

2.3 Zeitmanagement

In dieser Tabelle ist zu sehen, wie wir alles geplant haben. Man kann dazu auch die erreichten Ziele sehen. In den Kommentaren wird geschrieben, warum wir ein Ziel nicht erreicht haben.

Tag	Ziele	Erreichte Ziele
Mittwoch	<ul style="list-style-type: none"> • Richtigen Sensor auswählen • Abstandssensor und ESP-32 richtig verkabeln • Code schreiben, um den Abstand anzeigen zu lassen • Abstand mit Node-RED anzeigen lassen 	<ul style="list-style-type: none"> • Richtigen Sensor auswählen • Abstandssensor und ESP-32 richtig verkabeln • Code schreiben, um den Abstand anzeigen zu lassen • Abstand mit Node-RED anzeigen lassen
Donnerstag	<ul style="list-style-type: none"> • Reset Knopf in Node-RED einbauen • Testing durchführen • Projektdokumentation Kapitel 3 und 4 schreiben 	<ul style="list-style-type: none"> • Reset Knopf in Node-RED einbauen • Testing durchführen
Freitag	<ul style="list-style-type: none"> • Dokumentation fertig schreiben • Alles auf GitHub hochladen 	<ul style="list-style-type: none"> • Doku wurde fertiggeschrieben • Es wurde alles hochgeladen

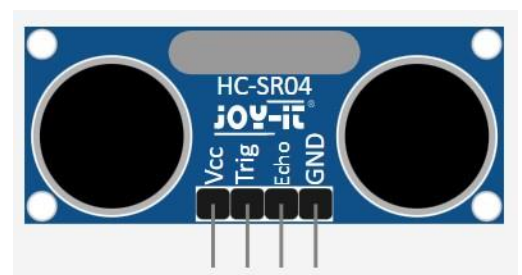
3 Entscheidung

3.1 Sensoren

Hier werden die benutzten Sensoren aufgezählt und beschrieben. Diese messen oder erfassen physikalische Eigenschaften und wandeln sie meistens in eine elektrische Grösse um.

3.1.1 KY-050 Ultraschallabstandssensor

Der Sensor dient für die Messung der Distanz in einem Bereich zwischen zwei Zentimeter und drei Meter. Am Trigger-Eingang wird ein Signal angelegt, eine Abstandsmessung wird durchgeführt und am Echo-Ausgang als PDM-TTL Signal ausgegeben. Die Entfernung zu einem Objekt kann berührungslos mit einem Ultraschalllautsprecher und einem Mikrofon gemessen werden. Der Ultraschall- Abstandssensor



[Quelle](#)

Modul 216 Projektarbeit Briefkasten

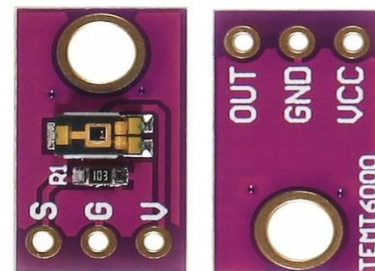
eignet sich vor allem zur Hinderniserkennung, Entfernungsmessung als Füllstandanzeiger und für Industrieanwendungen.

PDM (PWM): Die Pulsdauermodulation ist ein digitales Signal, welche analoge Ausgabe «simuliert» durch unterschiedliche Low- und High-Zeiten.

TTL: Die Transistor-Transistor Logik beschreibt nur die Spannungen des digitalen Aus- oder Einganges.

3.1.2 TEMT6000 Lichtsensor

Der TEMT6000 Sensor hat eine maximale Empfindlichkeit von 570 nm (Wellenlänge) und einen Halbwinkel von etwa 60 Grad. Da es das menschliche Auge nachahmen soll, reagiert es nicht gut auf IR- oder UV- Licht.



Quelle

3.2 Ressourcen

Kategorie	Tool	Beschreibung
Software	Arduino IDE	DIE zur Programmierung des Codes für den Briefkasten
Software	MQTT	Server zur Übermittlung der Daten von ESP32 zu Node-RED
Software	Node-RED	Informationen bekommen, bearbeiten und senden
Hardware	ESP32	Informationen von den Sensoren erhalten und verarbeiten, Ausführung des Codes
Hardware	Breadboard	Verbindung von allen Komponenten
Hardware	Ultraschallabstandssensor (KY-050)	Abstand in Daten umwandeln, weiterleiten
Hardware	Dupont Lines (Jumper-Kabel, Patchkabel) x8	Verbindung zwischen ESP, Breadboard und Breadboard und Sensoren
Software	Word	Erstellung der Projektdokumentation

Software	GitHub	Abgeben von fertigen Produkten und Resultaten
Software	Teams	Updates Empfangen und Kommunikation
Infrastruktur	WLAN	Allgemeine Nutzung

4 Realisierung

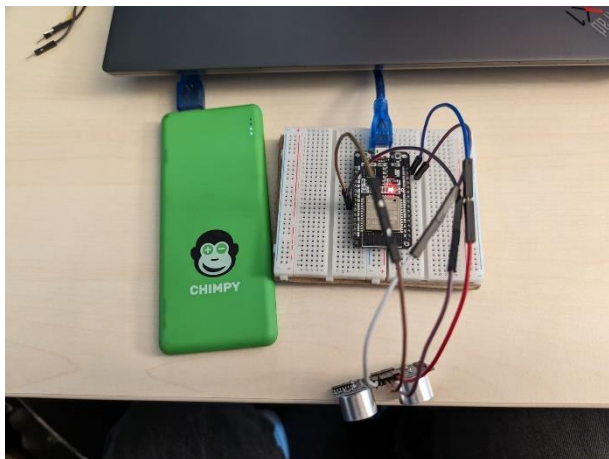
4.1 Vorbereitung

Wir haben die gleiche Gruppe, wie in der ersten Woche gemacht. Nach der Aufteilung haben wir zuerst diskutiert und geplant, wie wir vorgehen. Zwischen dem Abstandssensor und Lichtsensor haben wir uns für den ersten entschieden. Unsere Idee war es an der hinteren Seite des Briefkastens den Sensor zu montieren, damit wir die Daten bekommen, wenn jemand den Briefkasten öffnet. Wenn der Briefkasten geöffnet wird, sehen wir die Distanz. Dafür haben wir an einem Breadboard.

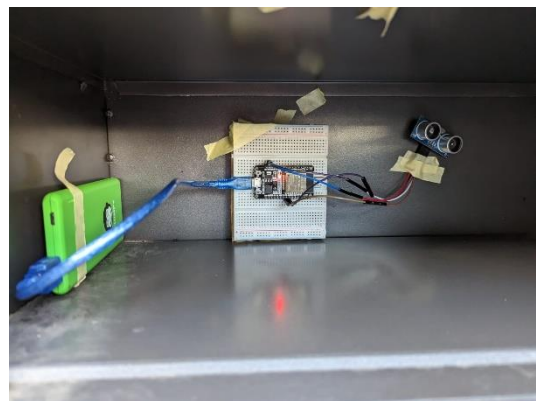
4.2 Verkabelung

Die Verkabelung beinhaltet das Verbinden des ESP mit dem Breadboard und anschliessend das Verbinden des Breadboard mit Sensoren oder Aktoren. Die meisten Sensoren besitzen drei GPIO (Generell Purpose Input Output). Einer davon ist für die Stromversorgung zuständig. Somit ergibt sich, dass eine Erdung benötigt wird. Pins Trig und Echo kann man mit irgendwelchen GPIO-Pins, die einen ADC besitzen, verbinden. Man muss die Zahlen der Pins im Code aktualisieren.

ESP-32	Ultraschallabstandssensor (KY-050)
GND	GND
VIN	VCC
D5	ECHO
D18	TRIG



Im Briefkasten sah es nachher so aus →



Quelle

4.3 Code

Für unseren Code haben wir zuerst online einen Demo-Code für unseren Sensor gesucht, damit wir wissen konnten, wie unser Ultraschall Sensor programmiert wird. Wir haben danach diesen Code mit dem Code für die MQTT-Verbindung vom PDF 02 zusammengefügt und haben dann den Output in dem Serial Monitor angeschaut. Hier ist der Link zum Code:

https://github.com/f171p/BLJ2022_uek216_briefkasten/tree/main/Code/esp32CodeWithWiFi

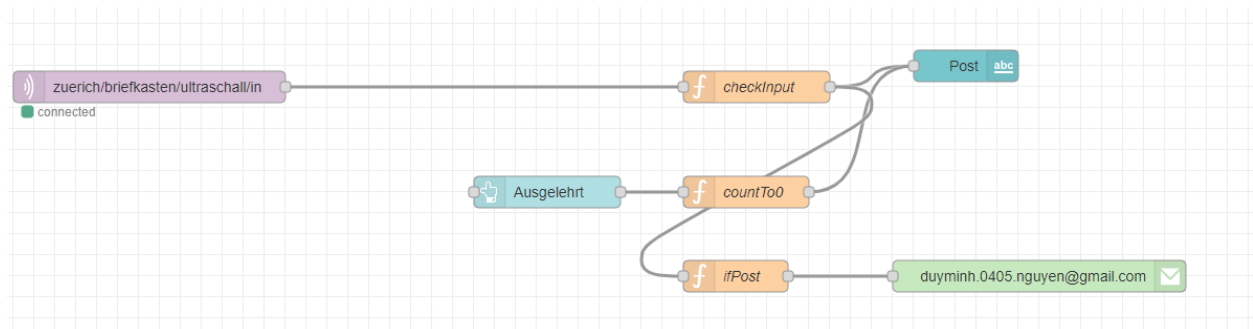
Damit wir unser Gerät auch draussen verwenden konnten, musste es auch möglich, dass der ESP-32 mit einem Hotspot verbinden kann. Dafür haben wir einen zweiten, fast identischen, Code geschrieben. Der Einzige unterschied der Beiden, ist, dass das Programm mit dem Hotspot noch eine konstante Variable, der den Hotspot-Passwort enthält. Diese wird in der setup_wifi Funktion beim Aufruf des WiFi.begin Methode benutzt wird. Der Code mit dem Noser Young WLAN braucht nur die ssid Variable. Hier ist der Link zum Code:

https://github.com/f171p/BLJ2022_uek216_briefkasten/tree/main/Code/esp32CodeWithHotspot

Das Programm verbindet sich zuerst mit dem Internet und MQTT und versendet danach jede Sekunde die Distanz vom Sensor zu dem nächsten Gegenstand per MQTT an alle Geräte, die zum topic «zuerich/briefkasten/ultraschall/in» abonniert sind.

4.4 Node-RED

Am Anfang des Node-RED flows steht das MQTT-in Node welche alle Nachrichten vom topic „zuerich/briefkasten/ultraschall/in“ bekommt. Es schickt diese Daten an dem „checkInput“ Node, welche die Daten anschaut und bestimmt, ob der Abstand vom Sensor genug gros ist. Wenn schon, dann inkrementiert es die globale Variable „count“ um 1. Diese Variable wird schlussendlich mit dem „Post“ Node in der Node-RED UI angezeigt. Der Button „Ausgelehrt“ setzt mit Hilfe der „countTo0“ Funktion die globale Variable „count“ auf 0. Die „ifPost“ Funktion schickt mit dem E-Mail Node eine E-Mail, wenn „count“ über 0 ist. Die E-Mail beinhaltet einen kurzen Satz: „Sie haben Post!“. Damit aber diese E-Mail nicht immer geschickt wird, bis count wieder auf 0 ist, haben wir eine zweite Variable namens „count2“ erstellt, die immer um 1 erhöht wird, wenn eine E-Mail geschickt wird und damit nochmals eine E-Mail geschickt wird, muss count grösser als count2 sein.



[Quelle](#)

4.5 Fertiges Produkt

Wir haben ein kleines Video gedreht, das zeigt, wie das fertige Produkt aussieht und funktioniert.
https://github.com/f171p/BLJ2022_uek216_briefkasten/blob/main/Photos/WhatsApp%20Video%202022-11-24%20at%202.29.21%20PM.mp4

5 Kontrolle

5.1 Fehler und Probleme auf dem Weg

Aufgetretenes Problem	Beschreibung	Lösung
Keine Abstandssensor gefunden	Wir konnten bei den Komponenten keinen Abstandssensor finden	Wir haben jemanden gefragt, der es nach Hause genommen hat
VIN-Pin nicht gefunden	Wir wollten den ESP-32 und den Sensor verkabeln. Im Internet ist gestanden, dass man den VIN-Pin benutzen soll, doch wir hatten keinen auf unseren ESP-32.	Wir haben eine ältere Version vom ESP-32 genommen, auf dem VIN vorhanden war.
Verbindung zwischen Laptop und ESP nicht möglich	Als wir mit dem Coden anfangen wollten, haben wir den ESP-32 mit Oscars Laptop verbunden. Der ESP-32 konnte in Arduino nicht gefunden werden	Wir haben die Arduino IDE auf Oscars Laptop neu gestartet.
Variable Count immer auf 0 zurückgesetzt	Unsere Variable, die für den Count zuständig war, wurde nach jedem Durchgang wieder auf 0 gesetzt.	Die Variable Count haben wir am Anfang nicht mit 0 definiert, sondern es soll schauen, ob es eine Variable Count schon gibt. So kann Count immer eins grösser werden.
ESP-32 vom Briefkasten mit Laptop verbinden nicht möglich	Wir gingen zum ersten Mal zum Briefkasten unten, um es zu testen. Der Laptop, den wir nicht mitgenommen haben, hat keine Reaktion gezeigt.	Wir mussten alles mit den Handy Hotspots verbinden.
Sensor nicht richtig funktioniert	Der Sensor hat aufgehört Werte aufzunehmen, und dann ging es wieder für wenige Sekunden.	Wir haben einen neuen Abstandssensor genommen und diesen neu mit allem verkabelt.
Nicht die richtige Zahl angezeigt bekommen	Unser Programm soll zeigen, wie oft der Briefkasten geöffnet wurde. Wenn man den Briefkasten zu lange offengehalten hat, hat es mehr als 1 gezeigt.	Nach jeder Öffnung haben wir einen Delay gecoded, so wird immer 1 angezeigt, wenn man es einmal aufmacht.
Falsche Daten beim E-Mail Node	Es konnten keine E-Mails geschickt werde, weil der E-Mail Node ein spezielles Passwort braucht, den man von seinem eigenen Google Account bekommen muss.	Auf Oscars Google Account ein App password generiert und diesen im Node eingegeben

5.2 Black-Box-Testing

Wir haben uns entschieden den Black-Box-Test direkt beim Briefkasten durchzuführen. Wir haben den Button getestet, der den Count wieder auf null setzt. Um es zu testen haben wir versucht irgendwelche Fehler zu finden, deswegen haben wir den Briefkasten einmal sehr schnell aufgemacht, um zu testen, ob es das immer noch erkennen kann. Wir haben den Briefkasten auch lange aufgehalten, um zu sehen, wie der Count sich verändern würde. Als wir es schnell aufgemacht haben, wurde nur 1 gezählt, doch als wir den Briefkasten länger aufgehalten haben, hat es einfach weitergezählt.

5.3 White-Box-Testing

Beim White-Box-Test haben wir mehr mit dem Code getestet. Wir benutzten unser Wissen vom Code, um alles besser zu testen. Damit es immer funktioniert, mussten wir den perfekten Abstand finden, bei dem der Sensor reagiert. Wir haben im Code immer wieder den Abstand verändert, um die richtige Zahl zu finden. Wir haben auch versucht den Briefkasten so lange aufzuhalten, um zu sehen was passiert, wenn der Delay fertig ist. Wir konnten so herausfinden, wie lange der Delay sein sollte.

5.4 Testkonzept

5.4.1 Testziele

Mit dem Testing wollen wir,

- dass unseres Projektziel erreicht wird
- dass unser Code gut aufgebaut ist und funktioniert
- dass alles am richtigen Ort eingesteckt ist
- dass wir auffällige Fehler finden und diese korrigieren

5.4.2 Testobjekte

Diese Komponente und wurden benutzt:

- ESP32
- Ultraschallabstandssensor
- Jumper Kabel
- Breadboard
- Briefkasten
- Powerbank (Chimpy style)

5.4.3 Testarten

Mit welchen Testarten wurde angewendet? Welche Komponenten/Funktionen werden mit welcher Art von Test überprüft?

White-Box:

- Code
- Delay

Black-Box:

- Button
- Zeitlänge beim Aufmachen des Kastens

5.4.4 Testinfrastruktur

Folgende Tools wurden bei der Arbeit benutzt:

- MQTT
- Node-RED
- Arduino IDE
- Google
- Word

5.4.5 Testorganisation

Wie ist das Testteam organisiert? Wer führt welche Tests durch? Wer ist wofür verantwortlich?

White-Box Testing: Oscar, Nikola

Black-Box Testing: Oscar, Nikola, Filip

5.4.6 Testplanung

Wir haben unsere Tests für den Donnerstag geplant. Diese haben wir auch schlussendlich bis am Mittag ausgeführt. Wir haben für das Testing etwa eine Stunde geplant, es ging aber fast zwei Stunden.

5.5 Testdrehbuch

Testfall-Nr.	1				
Testfall-Bezeichnung	Ultraschall Sensor				
Anforderungs-Nr.	1				
Testumgebung	Noser Young ESP-32, Breadboard, HC-SR04, 8 Jumper Cables Laptop mit <ul style="list-style-type: none">Windows 10 Pro 64 Bit, Version 22H2 – Build 19045.2251Google Chrome, Version 107.0.5304.10714 Zoll Bildschirm, Auflösung 3840 x 2160				
Zu testende Funktionalität	Testet den Ultraschal Sensor. Getestet wird die Distanz, die gemessen wird.				
Datum der Testdurchführung	23.11.2022				
Tester	Oscar Doell				
Testschritte:					
Nr.	Aktion	Erwartetes Ergebnis	Effektives Ergebnis	Erfüllt	Kommentar
1	ESP-32 Strom geben	Im Serial Monitor werden die Distanz Daten angezeigt	Wie erwartet	Ja	
2	Die Hand etwa 10cm vor dem Sensor halten	Es wird ungefähr 10cm als Output im Serial Monitor angezeigt	Manchmal ging der Output bis auf 15 hoch, obwohl wir nichts geändert.	Nein	Die Daten waren aber meistens ~10cm.
3	Die Hand nahe am Sensor gehalten	Es wird 0 oder 1 cm im Serial Monitor angezeigt	Es wurden Werte über 1100 angezeigt	Nein	Die Schallwellen kommen nicht an, also wird der maximaler Wert angezeigt

Modul 216 Projektarbeit Briefkasten

Testfall-Nr.	2				
Testfall-Bezeichnung	Node-Red UI				
Anforderungs-Nr.	1				
Testumgebung	Noser Young ESP-32, Breadboard, HC-SR04, 8 Jumper Cables Laptop mit <ul style="list-style-type: none">Windows 10 Pro 64 Bit, Version 22H2 – Build 19045.2251Google Chrome, Version 107.0.5304.10714 Zoll Bildschirm, Auflösung 3840 x 2160 Node-RED v 2.1.3				
Zu testende Funktionalität	Testet der Node-RED UI. Getestet wird, ob die gemessenen Daten vom HC-SR04 in der UI angezeigt werden.				
Datum der Testdurchführung	23.11.2022				
Tester	Oscar Doell				
Testschritte:					
Nr.	Aktion	Erwartetes Ergebnis	Effektives Ergebnis	Erfüllt	Kommentar
1	ESP-32 Strom geben	Im Serial Monitor und in der Node-RED UI werden die Distanz Daten angezeigt	Wie erwartet	Ja	
2	Die Hand etwa 10cm vor dem Sensor halten	Es wird ungefähr 10cm als Output im Serial Monitor und in der UI angezeigt. Es wird auch angezeigt, dass es Post gibt in der UI.	Sobald man die Hand wegnimmt, steht es, dass es keine Post gibt. Die Distanz Werte stimmen.	Nein	In der Node-RED Funktion habe ich es geändert, dass der Text nach dem Auslösen nur auf „Keine Post“ zurückgeht, wenn ich es Manuell mache. Dies wird schlussendlich mit einem Button gemacht.

Modul 216 Projektarbeit Briefkasten

Testfall-Nr.	3				
Testfall-Bezeichnung	Briefkasten				
Anforderungs-Nr.	1				
Testumgebung	Noser Young Briefkasten ESP-32, Breadboard, HC-SR04, 8 Jumper Cables, Chimp Powerbank Laptop mit <ul style="list-style-type: none">• Windows 10 Pro 64 Bit, Version 22H2 – Build 19045.2251• Google Chrome, Version 107.0.5304.107• 14 Zoll Bildschirm, Auflösung 3840 x 2160 Node-RED v 2.1.3				
Zu testende Funktionalität	Testet, ob das IoE Gerät im Briefkasten funktioniert. Getestet wird, ob es gleich funktioniert, wie im Office und ob der Button „ausgelehrt“ den count Wert auf 0 zurücksetzt.				
Datum der Testdurchführung	24.11.2022				
Tester	Oscar Doell, Nikola Bogosavjlevic				
Testschritte:					
Nr.	Aktion	Erwartetes Ergebnis	Effektives Ergebnis	Erfüllt	Kommentar
1	ESP-32 mit der Powerbank Strom geben	Im Serial Monitor und in der Node-RED UI werden die Distanz Daten angezeigt	MQTT-Daten wurden nicht übertragen.	Nein	Der ESP-32 konnte nicht mit dem WLAN verbinden, da wir Draußen waren.

Modul 216 Projektarbeit Briefkasten

Testfall-Nr.	4				
Testfall-Bezeichnung	Briefkasten 2				
Anforderungs-Nr.	1				
Testumgebung	Noser Young Briefkasten ESP-32, Breadboard, HC-SR04, 8 Jumper Cables, Chimp Powerbank Laptop mit <ul style="list-style-type: none">Windows 10 Pro 64 Bit, Version 22H2 – Build 19045.2251Google Chrome, Version 107.0.5304.10714 Zoll Bildschirm, Auflösung 3840 x 2160 Node-RED v 2.1.3				
Zu testende Funktionalität	Testet, ob das IoT Gerät mit einem Hotspot im Briefkasten funktioniert. Getestet wird, ob es gleich funktioniert, wie im Office und ob der Button „ausgelehrt“ den count Wert auf 0 zurücksetzt.				
Datum der Testdurchführung	24.11.2022				
Tester	Oscar Doell, Nikola Bogosavjevic, Filip Franek				
Testschritte:					
Nr.	Aktion	Erwartetes Ergebnis	Effektives Ergebnis	Erfüllt	Kommentar
1	ESP-32 mit der Powerbank Strom geben	Im Serial Monitor und in der Node-RED UI werden die Distanz Daten angezeigt	Wie erwartet	Ja	Mit dem Hotspot konnte es die MQTT-Daten übertragen.
2	Briefkasten aufmachen	Den Wert count wird um 1 erhöht und in der UI angezeigt.	Die MQTT-Daten wurden nur manchmal übertragen.	Nein	Wir haben ein neues HC-SR04 und neue Kabel geholt und haben diese Tests wiederholt.
3	Briefkasten aufmachen 2	Der Wert count wird um 1 erhöht und in der UI angezeigt.	Der Wert count wurde manchmal um mehr als 1 erhöht.	Nein	Wir haben den Delay von den Messages auf 1 sec erhöht.
4	Den Button „ausgelehrt“ anklicken	Der Wert count wird auf 0 zurückgesetzt	Wie erwartet	Ja	

Modul 216 Projektarbeit Briefkasten

Testfall-Nr.	5				
Testfall-Bezeichnung	E-Mail Notification				
Anforderungs-Nr.	1				
Testumgebung	Noser Young ESP-32, Breadboard, HC-SR04, 8 Jumper Cables, Chimpy Powerbank Laptop mit <ul style="list-style-type: none">Windows 10 Pro 64 Bit, Version 22H2 – Build 19045.2251Google Chrome, Version 107.0.5304.10714 Zoll Bildschirm, Auflösung 3840 x 2160 Node-RED v 2.1.3				
Zu testende Funktionalität	Testet, ob eine E-Mail an einer Person geschickt wird, wenn der Wert count sich um 1 erhöht.				
Datum der Testdurchführung	24.11.2022				
Tester	Oscar Doell				
Testschritte:					
Nr.	Aktion	Erwartetes Ergebnis	Effektives Ergebnis	Erfüllt	Kommentar
1	Der Wert count um 1 erhöhen	Eine E-Mail wird an einer Person geschickt mit der Nachricht „Sie haben Post!“	Wie erwartet	Ja	
2	Den Wert auf 1 lassen	Es werden keine E-Mails weitergesendet, bis der Wert count auf 2 gesetzt wird.	Wie erwartet	Ja	

6 Auswertung

Wir konnten als Gruppe gut kommunizieren und arbeiten. In diesem üK konnten wir viel lernen und ausprobieren. Es hat keine Kommunikationsprobleme, denn wir haben sehr schnell die Rollen aufgeteilt. Dieser üK wird uns in der Zukunft auf jeden Fall weiterhelfen, falls wir wieder mit Hardware. Es wird uns aber auch in zukünftigen Projekten viel weiterhelfen, weil wir auch vieles zu verbessern haben. Wir sind sehr zufrieden mit dem endgültigen Produkt. Wir haben unterschätzt wie viel Zeit die Lern- und Projektdokumentation einnehmen würden. Für zukünftige Projekte würden wir die Projektdokumentation häufiger aktualisieren.

7 Quellenverzeichnis

Systemübersicht-Quellen: Teams Ordner

Quelle Ultraschallabstandssensor: <https://sensorkit.joy-it.net/de/sensors/ky-050>

Quelle Lichtsensor: <https://de.aliexpress.com/item/32583469115.html>

Quelle Verkabelung: Selbstgemachte Bilder

Quelle Node-RED: ioe.noseryoung.ch:1884