

Projektdokumentation

Thema: Fahrstuhl

Dokumentinformationen

Dateiname: Projektdokumentation
Autoren: Timofey, Loris, Virginia
Speicherdatum: 25.11.2022

Inhaltsverzeichnis

1 Informieren	2
1.1 Ausgangslage.....	2
1.2 Vorgaben	2
1.3 Ziele	2
1.4 Anforderungen	2
2 Planen	3
3 Entscheiden	3
4 Realisierung	3
5 Testprotokoll	4
5.1 Lösung der Probleme.....	5
6 Auswertung	6
Einleitung bauen des Projects	6
Quellen	8

1 Informieren

1.1 Ausgangslage

Als Projektarbeit für den üK 216 – Internet of Everything, haben wir den Auftrag erhalten uns mit IoT-Geräte auseinanderzusetzen. Unsere Gruppe beschäftigt sich mit dem Thema Fahrstuhl. Die Noser Young befindet sich im 6. Stock des Gebäudes und es werden die beiden Fahrstule von den Mitarbeiter gebrauch. Die Fahrstule sind oft besetzt und aus diesem Grund ist auch die Wartezeit lange, so geht wertvolle Arbeitszeit verloren.

Aus diesem Grund wollen wir ein hightech Gerät entwickeln, der uns Informationen geben kann, wo sich aktuell die Fahrstule befinden. Zusätzlich soll das Gerät uns ermöglichen vom Arbeitsplatz aus zu rufen. Mit dieser Technologie können wir entweder den Lift rufen bevor man raus läuft und damit Zeit sparen.

1.2 Vorgaben

Für die Realisierung von dieser Projektarbeit haben wir hilfreiche Elemente zur Verfügung bekommen:

- ❖ ESP32-Board
- ❖ Breadboards
- ❖ Diverse Sensoren
- ❖ Dupont Lines (Patchkabel, Jumper-Kabel)
- ❖ Beispielcode für ESP32-Board
- ❖ Hilfsdokumentation(siehe Dokument «Getting Started» in Microsoft Teams)
- ❖ Infrastruktur
 - WLAN
- ❖ IoT-Plattform
 - MQTT
 - Node-RED

1.3 Ziele

- ❖ Wir können den Lift via Fernbedienung rufen
- ❖ Wir können nachschauen, ob sich gerade der Lift auf der 6. Etage befindet
- ❖ Gute Zusammenarbeit und Kommunikation innerhalb der Gruppe

1.4 Anforderungen

- ❖ Umsetzung gem. Bewertungsraster (siehe Dokument «Bewertung» in Microsoft Teams)
- ❖ Projektarbeit vor dem Abgabetermin fertig haben und abgeben
- ❖ Fahrstuhl-Dasein soll mittels des Lämpchens aussen beim Knopf ermittelt werden
- ❖ Holen des Fahrstuhls mittels eines Servo-Motors, der auf den Fahrstuhlknopf drückt
- ❖ Status bei Änderung an MQTT Topic «zuerich/lift/[xyz]» übermitteln
- ❖ Eingabe und Ausgabe von Daten über Node-RED
 - Eingabe: Lift holen
 - Ausgabe: Ist ein Lift zurzeit im 6. Stockwerk?

2 Planen

Die Planung ist ein wesentlicher Teil von jeder Arbeit. Man soll genug Zeit dafür investieren, damit man später möglichst wenige Missgeschicke geschehen. Aus diesem Grund haben wir uns als erstes zu dritt gessesen und haben angeschaut was alles zu tun ist. Wir haben uns überlegt wie wir vorgehen wollen. Danach haben wir drüber nachgedacht, welche Komponenten wir möglicherweise brauchen würden und haben diskutiert welche Lösungswege möglich sind. Folgend haben wir ein Tagesziel gesetzt.

Wir hatte den Grund Prinzip, wer oder was jeder macht, aber wir alle hatten mit jeder Aufgabe etwas zu tun, es war eben nicht konkret unterteilt.

Hier waren einige der Aufgaben:

- Projekt Dokumentation schreiben
- Das Code für ESP32 schreiben
- Node-Red Flow erstellen
- Testing Protokoll durchführen

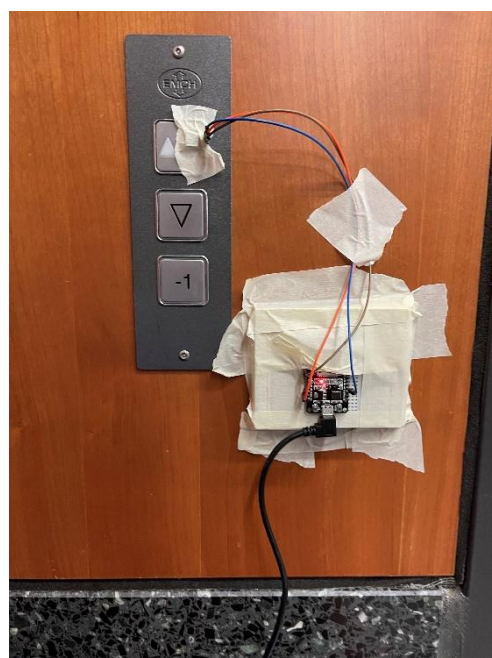
3 Entscheiden

Nachdem wir die verschiedene Vorgehen besprochen haben, haben wir uns als Gruppe schliesslich für eine entschieden. Wir machten uns zuerst einen Plan, was wir für Komponente wir brauchen. Wir haben dann entschieden, welchen Lift es zu montieren. Wir entschieden den Rechten. Ich haben auch entschieden, wie wir das Flow auf Node-Red alles verbinden. Wir haben auch noch entschieden, was und wie wir alles Testen werden und können.

Später haben wir entschieden, den anderen Lift auch zu montieren, aber nur mit einem Sensor, denn wir hatten noch Zeit gehabt.

4 Realisierung

Wir haben zwei Module für die beiden Liften Gebaut. Das Eine für mit einem Servo und Licht Sensor. Das Andere Modul wird eben nur mit einem Licht Sensor verbunden. Wir haben dann alles mit Klebeband auf der Wand verbunden.



Wir haben beide ESP-32 mit Powerbanks verbunden für Strom. Wenn es zu Schwierigkeiten mit den Boards haben, konnten wir sie mit dem Computer verbinden, um Zugriff zu Serial-Monitor haben. Mit Node-Red konnten wir das Flow und Logik für das Projekt erstellen. Wir haben am Ende alles getestet, was wichtig ist, das Projekt zu funktionieren.

5 Testprotokoll

Wir haben ein Testprotokoll auf Excel gemacht und haben alle Testergebnisse aufgelistet. Die Tabelle dient der Übersicht der Tests. Anschliessend gibt es noch bei denen der Status «In Bearbeitung» ist, eine zusätzliche Tabelle wie wir vorgegangen sind, das Problem zu lösen. Da die Tabelle zulange ist, musste man die Tabelle spalten.

Nr.	Test	Eingabe
1	Bekommt das ESP32 Strom?	ESP32 mit Powerbank verbunden
2	Funktioniert der (1.) Sensor?	Sensor am rechten Lift montiert
3	Werden Infos (Sensor) erhalten?	Rechter Lift gerufen
4	Funktioniert der Aktor richtig?	Auf der Webseite "Lift rufen" gedruckt
5	Dreht sich die Flügel vom Motor?	Auf der Webseite "Lift rufen" gedruckt
6	Funktioniert der (2.) Sensor?	Sensor am linken Lift montiert
7	Werden Infos (2. Sensor) erhalten?	Linker Lift gerufen
8	Werden Infos (2. Sensor) erhalten?	Linker Lift gerufen

Es geht eben weiter, aber konnte hier nicht reinpassen. Das Excel Dokument findet man im Repo.

Erwartetes Ergebnis	Erhaltenes Ergebnis	Status
---------------------	---------------------	--------

Rotes Licht am Board wird eingeschaltet	Daten Übermittlung zu langsam	In Bearbeitung
Infos vom Sensor wird am Programm übertragen	Infos wurden übertragen	OK
Auf der Webseite steht "Lift ist hier"	Braucht zu lang zum anzeigen	In Bearbeitung
Lift wird gerufen und kommt zu den 6. Stock	Lift wird gerufen	OK
Es dreht bis der Knopf gedruckt wird und retour	Dreht sich zu viel	In Bearbeitung
Infos vom Sensor wird am Programm übertragen	Infos wurden nicht übertragen	In Bearbeitung
Auf der Webseite steht "Lift ist hier"	"Lift ist hier" und "Lift ist nicht hier", haben sich abgewechselt	In Bearbeitung
Lift wird gerufen und kommt zu den 6. Stock	Beim linker Lift steht "Lift ist hier"	OK

5.1 Lösung der Probleme

Alle Probleme die beim testing aufgetaucht sind, haben wird angeschaut und haben zusammen nach einer Lösung gesucht. Die Lösung haben wir in einer zweiten Tabelle festgehalten.

Nr.	Lösungsweg	Erhaltenes Ergebnis	Status
1	Kabeln verkürzt & Breadboard an die Wand	Daten schneller übertragen worden	OK
3	Sobald der Sensor das Licht erkannt hat OK	"Lift ist hier" wurde angezeigt	OK
5	Drehwinkel und Position von Flügel angepasst	Hat sich korrekte Winkel gedreht und retour	OK
6	Pins umgestellt	Infos wurden übertragen	OK
7	Auf Node Red ein zweites Text Feld für den Linken Lift eingebaut	Infos der beiden Lift wurden erhalten und wurden angezeigt	OK

6 Auswertung

Das üK 216 hat uns viele Schwierigkeiten und Hürden bereitet, aber wir konnten es mit der Gruppe lösen. Die Teamfähigkeit und das Durchhaltevermögen waren in diesem üK immer im Mittelpunkt. Das Fachgespräch hat uns viele Nerven gekostet, aber durch die vielen Übungen und das lernen, wurde die Angst entzogen.

Die Projektarbeit war eine schwierige, aber doch eine lustige Zusammenarbeit. Nach diesem üK kann man viel Mitnehmen ins berufliche Leben, da die Arbeit umfang reich war. Es beinhaltete programmieren mit dem Arduino Programm, es brauchte Fingerspitzengefühl, um die verschiedene Komponente mit dem ESP-32 zu verbinden. Es braucht auch wissen, wie man mit den verschiedenen Tools verbinden kann und Ausdauer, das Projekt in reale zu bringen.

Einleitung bauen des Projects

Hier sind alle Dingen, die du ums Bauen des Projekt brauchst / BOM (Bill of Materials):

- Zugriff zu Node-Red
- MQTT Server haben
- Ein Computer mit Arduino IDE
- 2x ESP-32 Dev Module
- 2x TEMT-6000 Lichtsensoren
- Breadboards (Steckplattinen)
- 1x SG-90 360° servo Motor
- Kabeln für Verbindung (genug viel)
- USB B micro Kabel

Bilder von den Komponenten:



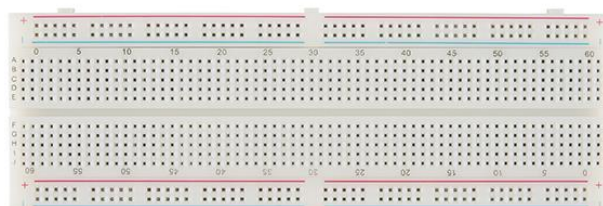
TEMT-6000



ESP-32



SG-90 servo



Breadboard

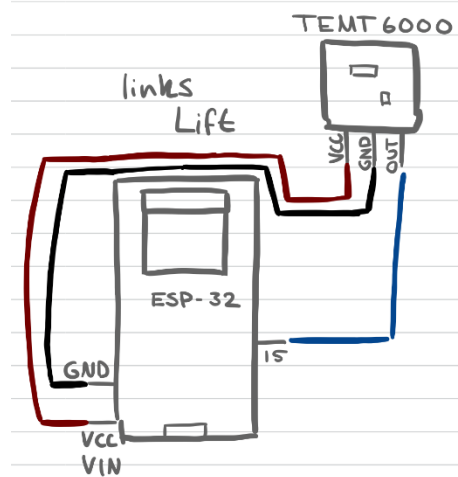
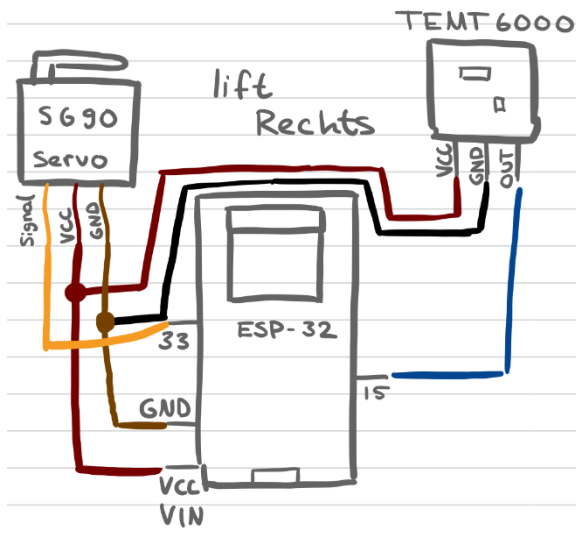
Hier ist der Link zu Arduino IDE: <https://www.arduino.cc/en/software>

Man muss Special Libraries für das Arduino IDE installieren mit mindesten Version:

- PubSubClient von Nick O'Leary ver. 2.8.0
- ServoESP32 von Jaroslav Paral ver. 1.0.2

Die Erwartung ist, dass du ein MQTT server hast und ein Node-Red Server hast.

Zuerst muss man die ESP-32 mit Sensoren, Aktoren und Strom verbinden, das Strom kommt vom USB Stecker, aber die Verbindung kann man mit diesen Diagrammen nachschauen.



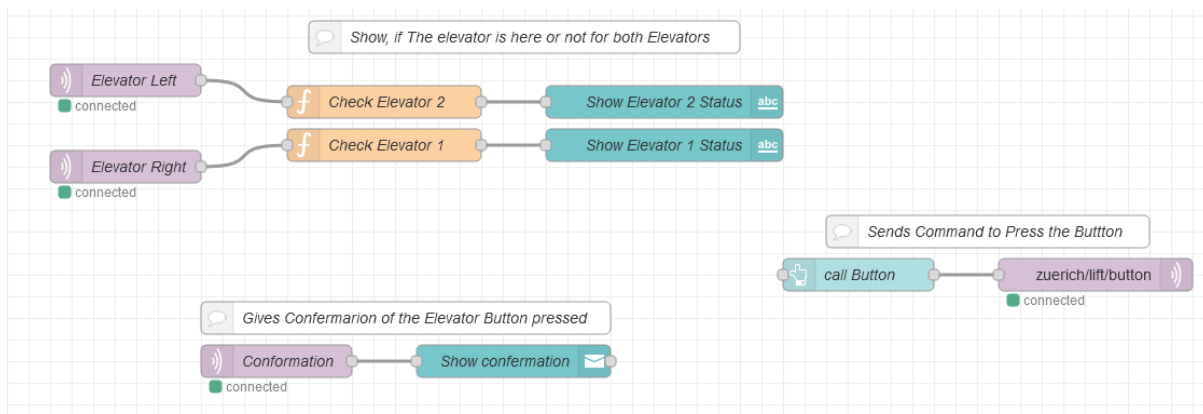
Das Code für die beiden ESP-32 findet man im Repo mit den Namen:

https://github.com/Timofey-Makhankov/BLJ2022_uk216_lift

- Für Rechts -> lift_right_node/lift_right_node.ino
- Für Links -> lift_left_node/lift_left_node.ino

Mit dem Computer verbinden und beide Mikrocontroller programmieren.

Für das Flow für Node-Red kann man unter Node-Red Flows Folder im Repo zu finden. Es sollte so aussehen:



Es sollte auch so montieren wie beim Realisierungs bilder.

Quellen

- ESP-32 -> <https://www.amazon.co.uk/AZDelivery-NodeMCU-CP2102-Development-Parent/dp/B07ZZFXRTY>
- TEMT-6000 -> <https://cdn.sparkfun.com/r/500-500/assets/parts/1/8/5/9/08688-01.jpg>
- SG-90 Servo -> <https://www.robotistan.com/tower-pro-sg90-rc-mini-servo-motor-eng>
- Breadboard -> <https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard/all>