Projektauftrag - Sitzungszimmer

üK 216

Dokumentinformationen

Dateiname:	Dokumentation_Sitzungszimmer_üK216
Speicherdatum:	13.12.2024

Autoreninformationen

Autor:	Leon Probst, Diego De Corso, Nico Linder
E-Mail:	leon.probst@noseryoung.com diego.decorso@noseryoung.com nico.linder@noseryoung.com
GitHub:	GitHub

Inhaltsverzeichnis

1 Einleitung

Das Ziel unseres Projekts ist es ein System zu entwickeln, das erkennt ob das Sitzungszimmer offen oder geschlossen ist und die Möglichkeit, das Zimmer zu reservieren. Dabei haben wir den ESP32 und verschiedene Sensoren eingesetzt. Diese Dokumentation beschreibt den gesamten Prozess nach der IPERKA-Methode.

2 Ablauf

2.1 IPERKA

2.1.1 Informieren

Am Anfang des Projekts haben wir uns mit den Anforderungen und Zielen auseinandergesetzt. Unsere Schritte in dieser Phase waren:

- **Projektauftrag analysieren:** Der Auftrag wurde genau durchgelesen und die Anforderungen festgehalten.
- **Ziele definieren:** Wir haben festgelegt, dass wir zuerst mit dem erkennen des offenen oder geschlossenen Zimmer beginnen und danach mit der Reservierungsmöglichkeiten arbeiten.
- **Ideensammlung:** Gemeinsam haben wir überlegt, wie der ESP32 erkennen kann, dass die Tür erkannt wird und wie wir vorangehen können.
- **Tools:** Zur Visualisierung unserer Ideen nutzten wir ein Miroboard.

Abbildung:
Das verwendete Miroboard zeigte eine Übersicht über alle Ideen und half dabei, die weiteren Schritte klar zu strukturieren.

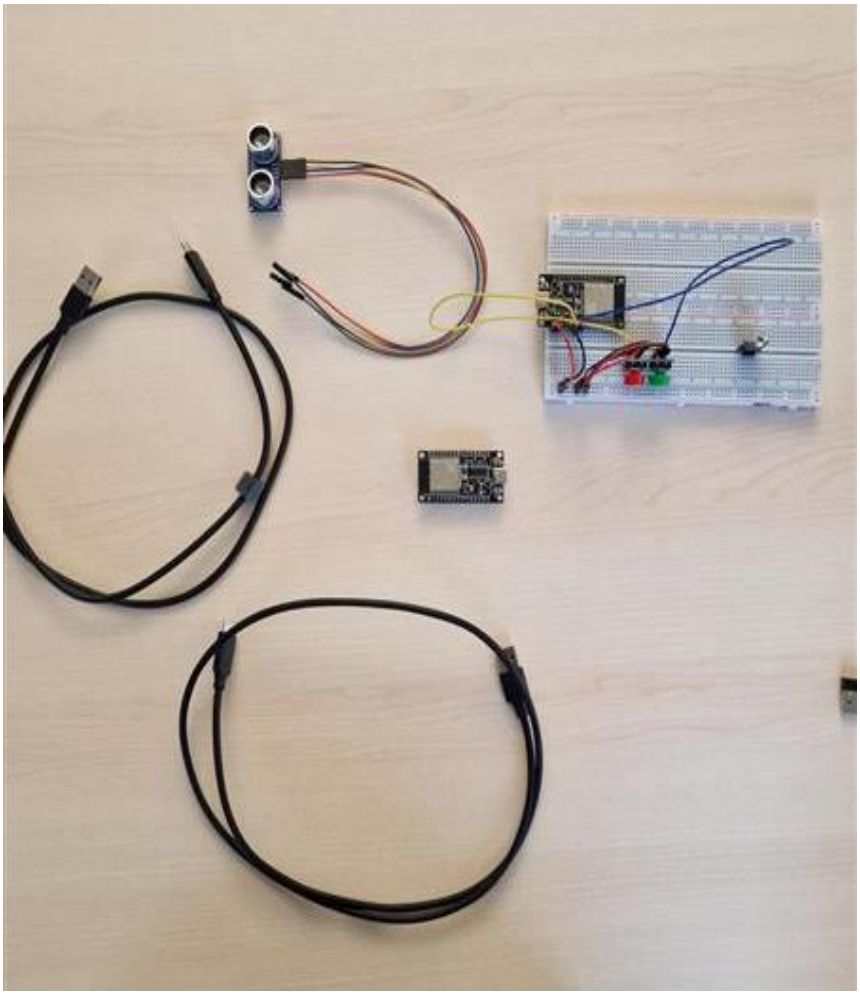
Miroboard

2.1.2 Planen

Wir haben uns überlegt, welche Teile wir verwenden werden und wo wir sie anschliessen müssen. Wir haben auch darüber nachgedacht, wo der ESP32 im Raum am besten platziert werden sollte, damit er das Sitzungszimmer gut erkennen kann. Unsere Schritte umfassten:

- **Platzierung:** Wir haben entschieden, den Entfernungssensor in der Höhe des Türgriffs zu montieren, da dies die beste Position zur Erkennung der Tür war.
- **Komponentenwahl:**
  - ESP32
  - Steckplatine
  - Male-to-Male Jumper Cable *6x*
  - Entfernungssensor
  - USB-C zu USB-A Kabel
  - Female-to-Female *4x*

- **Programmnutzung:**
  - Arduino IDE
  - Node-RED
  - MQTT Explorer
  - Word
  - Teams



2.1.3 Entscheiden

Nach dem Brainstorming haben wir uns für folgende Lösungen entschieden:

- **Sensor:** Ein Entfernungssensor wurde ausgewählt, da er einfach und genau misst, ob die Tür geschlossen oder offen ist. Dieser Sensor arbeitet von Ultraschall, was eine hohe Präzision bei der Erkennung gewährleistet.

- **Platzierung:** Der Sensor wurde auf der Türgriffhöhe positioniert, da diese Position genau optimal ist, um die Tür gut zu erfassen. Ausserdem wurde darauf geachtet, dass der Sensor nicht durch etwas blockiert wird.

Durch diesen Entscheid konnten wir eine robuste Lösung finden, die den Anforderungen unseres Projekts entspricht.

2.1.4 Realisieren

Die Umsetzung begann mit der Verkabelung und dem Aufbau der Hardware. Unsere Schritte waren:

1. **Hardwareaufbau:**
    - Der ESP32 wurde mit der Steckplatine verbunden.
    - Der Entfernungssensor wurde entsprechend platziert und verkabelt.
    - Alle Verbindungen wurden mit Jumper-Kabeln hergestellt.

2. **Softwareentwicklung:**
    - Mit der Arduino-IDE wurde der Code geschrieben und auf den ESP32 hochgeladen.
    - Node-RED wurde konfiguriert, um die Sensordaten zu empfangen.

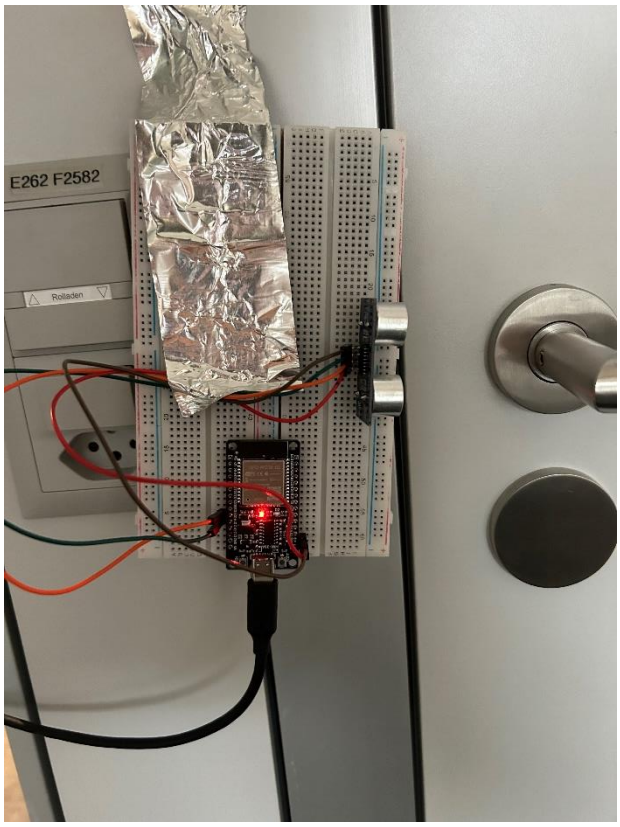
3. **Testlauf:**
    - Die Funktionalität wurde durch Tests der Hardware und Software sichergestellt.

2.1.5 Kontrollieren

Nach der Realisierung des Systems wurde es getestet, um sicherzustellen dass:

- Der Sensor den Status der Tür korrekt erkennt.
- Der Reservierungsstatus aktualisiert wird.
- Die Daten über MQTT-Explorer erfolgreich übertragen werden

- Das Node-RED Dashboard die Informationen korrekt darstellt.



Unsere Tests waren nicht immer erfolgreich, aber das hat uns nicht davon abgehalten, weiterzumachen und die Fehler direkt zu korrigieren.

2.1.6 Auswerten

In der letzten Phase haben wir unser Resultat nochmals überprüft:

- **Erfolge:**
  - Die Türstatus-Erkennung funktionierte wie geplant.
  - Die Reservierungsfunktion wurde erfolgreich integriert.
- **Verbesserungspotential:**
  - Zusatzfunktionen erst am Ende einfügen und sich zunächst auf das Wichtigste konzentrieren.

3 Tests

Die Tests wurden wie folgt ausgeführt und es wurde sichergestellt dass:

- **Testdokumentation:** Alle durchgeführten Tests detailliert dokumentiert wurden.

Testfall-ID	Testfallbeschreibung	Testart	Anzahl der Tests	Status
TEST 01	Überprüfung der Hardwareverbindungen	Whitebox	1	✓
TEST 02	Überprüfung, ob der Entfernungssensor die Entfernung korrekt misst.	Blackbox	3	✓
TEST 03	Überprüfung, ob der Entfernungssensor korrekt erkennt, ob die Tür offen oder geschlossen ist.	Blackbox	3	✓
TEST 04	Überprüfung der Sensorplatzierung (Höhe des Türgriffs)	Whitebox	2	✓
TEST 05	Funktionalität der Datenübertragung über MQTT	Blackbox	4	X
TEST 06	Funktionalität der Datenübertragung über MQTT 2.1	Blackbox	2	✓
TEST 07	Test der Benutzeroberfläche im Node-RED Dashboard	Blackbox	3	X
TEST 08	Test der Benutzeroberfläche im Node-RED Dashboard 2.0	Blackbox	1	✓

4 Ergebnisse und Fazit

Unser Projekt hat uns gezeigt, wie effizient ein kleiner Mikrocontroller zur einer einfachen aber doch noch nützlichen Lösung eingesetzt werden kann. Wir haben ein funktionierendes System entwickelt, das:

- Den Status des Sitzungszimmers erkennt.
- Uns eine Reservierungsmöglichkeit bietet.

Trotz kleinen Herausforderungen war das Projekt schlussendlich ein Erfolg. Wir konnten unsere Kenntnisse im Bereich IoT, IoE, Software wie Node-RED, Arduino IDE und den MQTT Explorer als auch Hardware wie, den ESP32 und dazugehörige Sensoren wie auch Aktuatoren, vertiefen.

5 Wissensbeschaffung

00_modul-216-ku-getting-started_zh-usb-c_v1-6.pdf
01_modul-216-ku-grundlagen-ioe-iot_v1-3.pdf
02_modul-216-ku-esp32-node-red-mqtt_zh-usb-c_v1-7.pdf

04_modul-216-ku-begleiter_zh-usb-c_v1-1.pdf

Entfernungsmessung