

MHS3D

19. Dezember 2024

Inhaltsverzeichnis

1	Requirements	3
2	Projektplan	4
2.1	Projektaufbau	4
2.2	grober Zeitplan	5
3	Schaltplan	6
3.1	Stromverbrauch der Uhr:	6
4	Datenübertragung	7

1 Requirements

1. Die Smartwatch soll den MPU6050 Beschleunigungssensor + Gyroskop ansteuern und deren Messdaten abfragen können.
2. Die von der Uhr gesammelten Messdaten sollen im laufenden Betrieb an einen PC übertragen werden können.
3. Bewegungsrichtungen sollen live in 3D (als Linie im "Raum") angezeigt werden.
4. Das Bewegungstracking der letzten 2 Stunden soll während des Betriebs gespeichert und dargestellt werden können.

Erweiterungsideen:

- Der Puls des Trägers wird als Liniendiagramm, gefärbt nach Bereichen, altersabhängig (grün bis rot) angezeigt werden.
- Pulsänderungen sollen als farbliche Veränderung des Bewegungsstriches im Raum visualisiert werden.
- Aufnahme der Daten in Sessions mit Button an der Uhr als Start/Stop
- Interaktive Drehung der Ansicht möglich
- Mapping der Bewegungen in die Raumpläne der HAW

2 Projektplan

2.1 Projektaufbau

Ein Projektplan der bei Konkretisierung und Arbeitsteilung nach dem Aufstellen der Anforderungen entstanden ist.

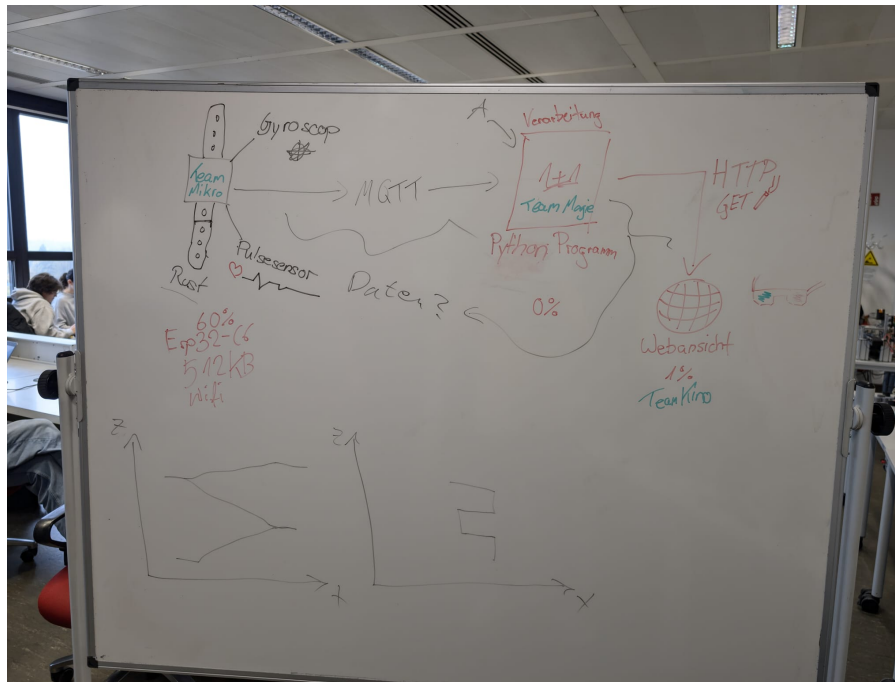


Abbildung 1: Projektaufbau

Die Idee war es mit der Uhr über die Sensorik Daten zu sammeln, diese per MQTT einem PC/Laptop zur Verfügung zu stellen und dort die gesammelten Daten aufzubereiten um diese dann in einer Webansicht darstellen zu können.

Nach der Gruppeneinteilung haben wir uns entschieden den Code für die Uhr in Rust zu schreiben, die Datenaufbereitung in Python zu machen und diese Daten per HTTP einem Javaprogramm zur Darstellung zugänglich zu machen.

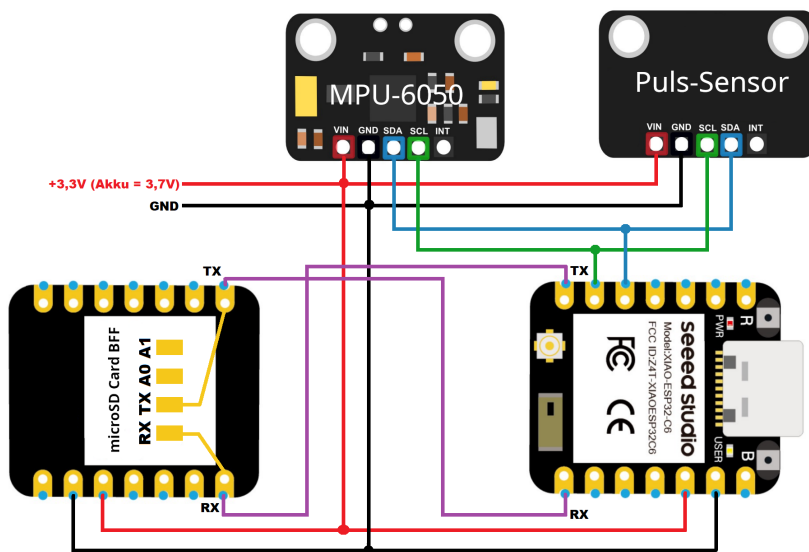
(vermutlich um angepassten digitalen Projektaufbau ergänzen/ersetzen und Änderungsgründe erläutern)

2.2 grober Zeitplan

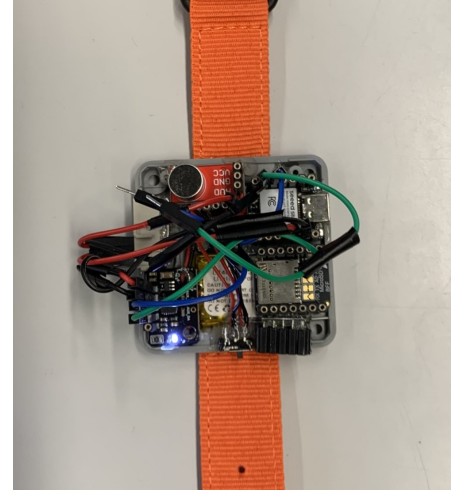
Ein grober Zeitplan der Recherche und Umsetzung des Projektplans den wir zu Anfang der Planung geschrieben haben.

KW	Micro	Magic	Kino
44	Projekt aufsetzen		
45	Daten per HTTP senden, Format der Daten festlegen	Recherche Algorithmen	Recherche: Darstellung Was und Wie?
46			Mockups erstellen
47	Herzschrittzähler zum Laufen kriegen		Aufsetzen des Frontends
48	Done		
49			Testen der Darstellung mit Dummy-Daten
50			
51			Daten von Magic empfangen und darstellen
54			
55			Testen
56			
57			

3 Schaltplan



(a) Schaltplan Smartwatch



(b) verkabelte Smartwatch

An den Mikrocontroller haben wir einen Puls-Sensor, den MPU-6050, ein 3-Achsen Gyroskop und Beschleunigungssensor, sowie einen microSD Kartenslot angeschlossen.

Zur Stromversorgung haben wir die Sensoren, den microSD Kartenslot und den Mikrocontroller mit 3,3V + Ground (im Schaltplan Rot + Schwarz) verbunden, zur Kommunikation zwischen dem Mikrocontroller und dem Puls-Sensor und dem MPU-6050 benutzen wir I2C (im Schaltplan Grün und Blau).

Die microSD Karte ist für lesende und schreibende Kommunikation mit 2 Rx Tx Verbindungen an den Mikrocontroller angeschlossen (im Schaltplan Lila).

Zur vereinfachten Verkabelung sind zwei Steckleisten auf der Uhr befestigt über die einmal die Stromversorgung und einmal die I2C Kabel gesteckt wurden.

3.1 Stromverbrauch der Uhr:

der Akku umfasst 700 mAh, der MPU6050 benötigt 4 mA und der ESP32C6 im Worst Case 240mA da wir wlan für http benötigen. Daraus ergibt sich bei unserer Verwendung für die Uhr eine Laufzeit von 2,8h.

4 Datenübertragung

Die Datenübertragung findet als HTTP GET Anfrage an den Webserver der Smartwatch statt. Die Smartwatch liefert dann einen JSON String von Datensätzen.

Ein Datensatz enthält einen Timestamp, Acceleration x,y und z Achse, Gyroskop x,y und z Achse sowie ein Pulssensor Messwert in Folgendem Format: