



## LABOR BERICHT

# Biosignalverarbeitung MGST-B-3-BB-BS-ILV

Gruppe A6: Labor 1

WINTERSEMESTER 2025/26

Studiengang

BACHELOR MEDIZIN-, GESUNDHEITS- UND SPORTTECHNOLOGIE

Verfasser:

*Moritz Mattes*

*Elias Maier*

*Hauke Döllefeld*

LV-Leiter:

*Dr. Aitor Morillo*

*Dr. Gerda Strutzenberger*

letzte Aktualisierung: 19. November 2025

**Inhaltsverzeichnis**

<b>1 Einleitung und Zielsetzung</b>	<b>1</b>
<b>2 Vorbereitende Arbeiten</b>	<b>2</b>
2.1 Verwendete Hardware . . . . .	2
<b>3 Versuchsaufbau und Durchführung</b>	<b>3</b>
3.1 Versuchsaufbau . . . . .	3
3.2 Durchführung . . . . .	3
3.2.1 Aufgabe 1 und 2 - Teil 1-4 . . . . .	3
3.2.2 Aufgabe 2 - Teil 5-6 . . . . .	3
3.2.3 Aufgabe 3 . . . . .	3
3.2.4 Aufgabe 4 . . . . .	4
3.3 Abgabe 1 . . . . .	4
<b>4 Ergebnisse und Interpretation</b>	<b>5</b>
4.1 Abgabe 1 . . . . .	5
4.1.1 Aufgabe 5 (a) . . . . .	5
4.1.2 Aufgabe 8 . . . . .	6
4.1.3 Aufgabe 9 und 12 . . . . .	7
4.1.4 Aufgabe 10 . . . . .	8
4.1.5 Aufgabe 13 . . . . .	8
<b>Literaturverzeichnis</b>	<b>III</b>
<b>Abbildungsverzeichnis</b>	<b>IV</b>
<b>Tabellenverzeichnis</b>	<b>V</b>

## 1 Einleitung und Zielsetzung

Die Laboreinheiten haben das Ziel den Studierenden die Grundlagen der Biosignalverarbeitung näherzubringen. Darin soll ein Crashkurs in den Umgang mit Arduino IDE, C und Arduino Uno enthalten sein, sowie in die Datenaquise mit dem Mikrocontroller-Setup. Damit gehen der Aufbau der Hard- und Software zur Detektion von Bewegungen und die Programmierung zur Analyse von Beschleunigungsdaten einher.

Im Labor 1 lag der Fokus auf dem Erkennen der richtigen Parameter für die Aufnahme der Daten eines IMU's, Inertial Measurement Unit, und dem Verwenden eines externen Dataloggers. Dabei wurden hauptsächlich ein Arduino Mikrocontroller vom Typ CH340 und ein MMA8452Q IMU-Sensor verwendet. In einer kleineren Teilaufgabe sollte auch ein Qwiic Open Log Datalogger angeschlossen werden, um die aufgezeichneten Daten zu speichern. Allerdings sollte die Teilaufgabe mit Datalogger auf Grund von Zeitmangel nur kurz behandelt werden, sodass diese im Laborbericht 1 nicht weiter beschrieben wird. Zur Aufnahme der Daten wurde die Bibliothek Sparkfun benutzt, welche eine einfache Schnittstelle zur Kommunikation mit dem Sensor bietet.

## 2 Vorbereitende Arbeiten

Die Aufgaben 1 und 2 sind als vorbereitende Maßnahmen bearbeitet worden. Im Rahmen der Aufgabe 1 wurden der Treiber für den Arduino Mikrokontroller installiert und der Blinktest durchgeführt. Die zweite Aufgabe beinhaltete das Einbinden der Sparkfun spezifischen Bibliotheken für Board und Sensor in die Arduino IDE. Zudem wurde mit einem Beispielskript erstmals Daten ausgelesen, über den seriellen Plotter ausgegeben und mit dem seriellen Monitor gespeichert.

### 2.1 Verwendete Hardware

Die Laboreinheit wurde mit den folgenden Hardware-Komponenten durchgeführt:

- Fixe Komponenten für alle Aufgaben
  - Mikrocontroller (Sparkfun)
  - Micro-USB-Kabel
  - Beschleunigungssensor (MMA8452Q)
  - Verbindungskabel (Qwiic)
- benötigte Komponenten für mobiles Setup
  - 9V Blockbatterie
  - 9V Batterieanschluss
  - Datalogger (Qwiic Open Log)
  - Micro-SD Karte
  - Verbindungskabel (Qwiic)
- Verwendete Software
  - Arduino IDE
  - Python 3.13.9 für unser Python Visualisierungsskript
  - .ino Skript Lab1Code1.ino für die Datenerfassung

Für die verschiedenen Aufgaben wurden unterschiedliche Kombinationen der oben genannten Hardware-Komponenten verwendet. Bei entsprechenden Aufgaben werden die Hardware Kombinationen und der genaue Versuchsaufbau näher beschrieben.

### 3 Versuchsaufbau und Durchführung

Im Folgenden Abschnitt wird der Versuchsaufbau beschrieben und die Durchführung des Experiments erläutert.

#### 3.1 Versuchsaufbau

Die zum Versuchsaufbau verwendeten Geräte und Materialien sind im Abschnitt Vorbereitende Arbeiten aufgelistet.

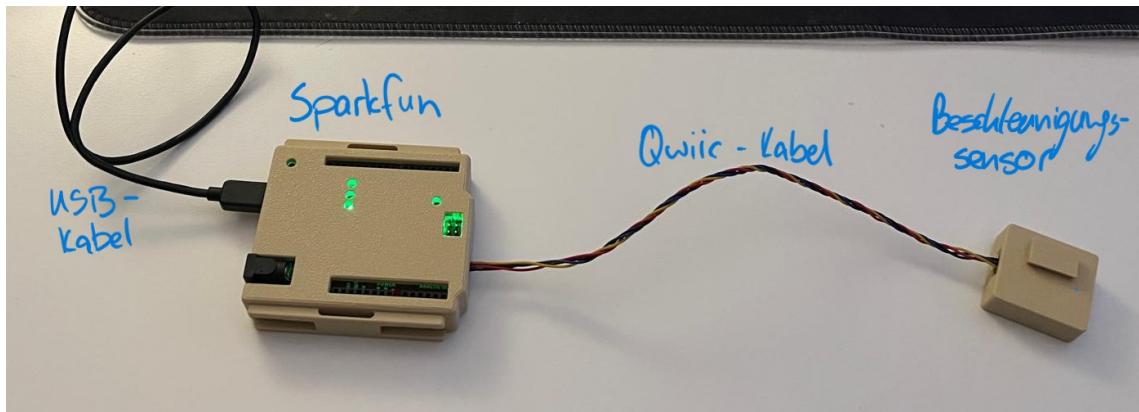


Abbildung 1: Versuchsaufbau

Wie in Abbildung 1 zu sehen ist, wurde der Mikrocontroller über das USB-Kabel mit dem Laptop und über ein Qwiic-Kabel mit dem Beschleunigungssensor verbunden.

#### 3.2 Durchführung

##### 3.2.1 Aufgabe 1 und 2 - Teil 1-4

Aufgabe 1 und 2 (Teil 1-4) wurden bereits im Abschnitt Vorbereitende Arbeiten beschrieben.

##### 3.2.2 Aufgabe 2 - Teil 5-6

Mit Hilfe der Analyse aus Punkt 4 konnte Punkt 5 (Koordinatensystem) beantwortet werden. Anschließend wurden die Daten die im Serial Monitor angezeigt und per copy-paste in ein txt-Datei gespeichert (Punkt 6).

##### 3.2.3 Aufgabe 3

Zuerst wurde, wie in Punkt 1 beschrieben, die Funktion des Dataloggers überprüft, indem das Testprogramm ausgeführt wurde und anschließend die Daten im PC-Interface betrachtet wurden. Hier gab es zuerst Schwierigkeiten, die aber nach einigen Versuchen überwunden werden konnten. Daraufhin wurde die Verbindung des Computers mit dem Sparkfun Board getrennt und die Stromversorgung über die 9V Batterie hergestellt (Punkt 2). Aufgrund von massiven Problemen die trotz troubleshooting nicht behoben werden konnten wurde auf Anweisung des Lektors die Aufgabe abgebrochen und zur nächsten weitergegangen.

**3.2.4 Aufgabe 4**

In der Aufgabe 4 wurden die Daten der mikro-SD-Karte des Dataloggers ausgelesen und im bereitgestellten Jupyter-Notebook ausgewertet.

**3.3 Abgabe 1**

In der Abgabe wurden folgende Punkte erledigt:

- 5 (a)
- 6
- 8 (a)
- 9
- 10
- 12

Die Aufgaben wurden entsprechend der Anweisungen erledigt und im Jupyter-Notebook dokumentiert.

Aufgabe 13 wurde nicht erledigt, da die Datenaufzeichnung mit dem Datalogger per 9V Batterie nicht funktionierte.

## 4 Ergebnisse und Interpretation

In diesem Abschnitt werden die Ergebnisse der im Abschnitt Versuchsaufbau und Durchführung beschriebenen Abgabe präsentiert und interpretiert.

### 4.1 Abgabe 1

#### 4.1.1 Aufgabe 5 (a)

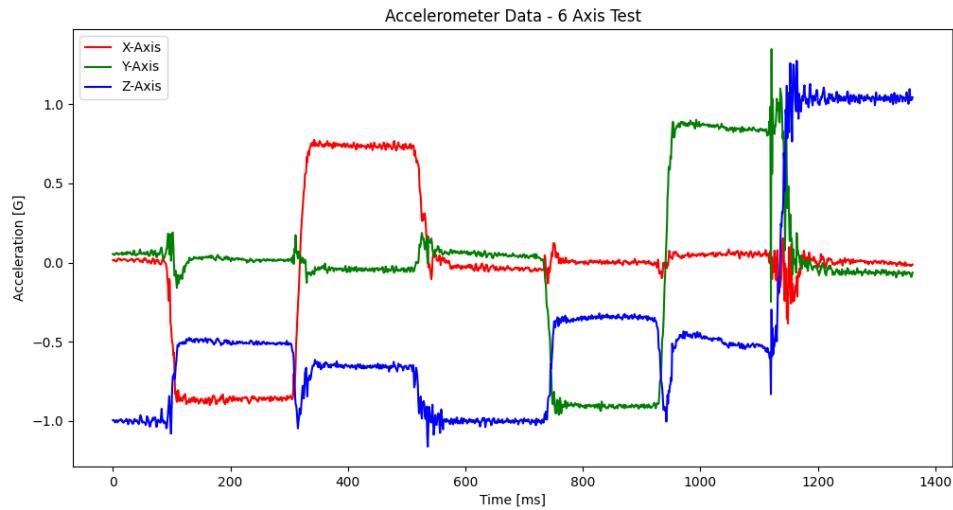


Abbildung 2: Abgabe 1 - Aufgabe 5 (a)

In Abbildung 2 ist die gemessene Beschleunigung des 6-Achsen-Tests dargestellt. Die X-, Y- und Z-Achse sind farblich unterschiedlich gekennzeichnet. Es ist zu erkennen, dass die Beschleunigung in den drei Achsen variiert, wenn der Sensor in verschiedene Positionen gebracht wird. Dies bestätigt die Funktionsfähigkeit des Beschleunigungssensors.  
Die Farben der entsprechenden Achsen sind wie folgt:

- X-Achse: Rot
- Y-Achse: Grün
- Z-Achse: Blau

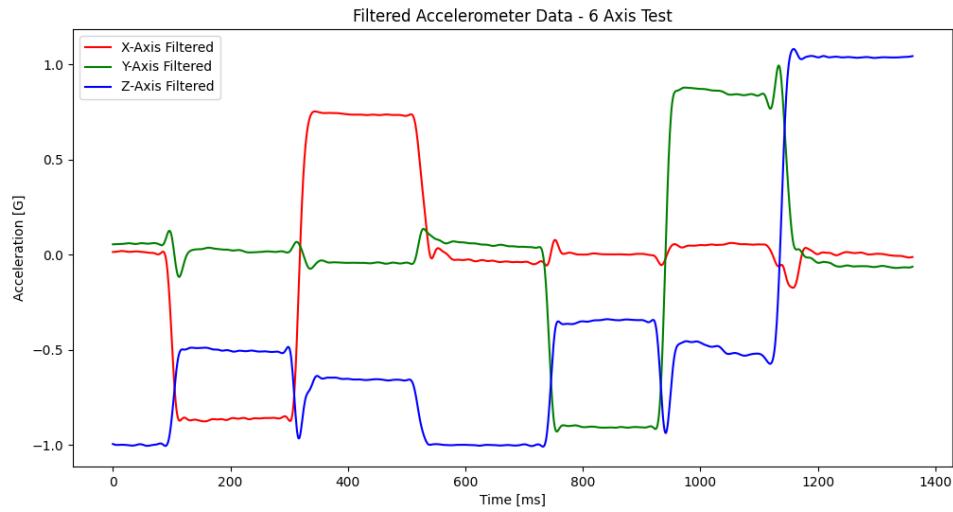


Abbildung 3: Abgabe 1 - Aufgabe 6

In der obigen Abbildung sind die gleichen Achsen wie in Abbildung 2 verwendet worden. Der eingesetzte Butterworth-Filter 4. Ordnung hat erreicht, dass die hochfrequenten Anteile des Signals deutlich reduziert wurden. Dadurch ist das Signal deutlich geglättet und Rauschen wurde entfernt. Da dieser Filter eine hohe Flankensteilheit besitzt, werden die niederfrequenten Anteile des Signals kaum beeinflusst, was zu einer guten Signalqualität führt.

#### 4.1.2 Aufgabe 8

An dieser Stelle muss angemerkt werden, dass die in Abbildung 4 dargestellten Daten nach dem Anpassen der falsch eingestellten Messparameter aufgezeichnet wurden. Da wir keine falschen Messwerte aus Aufgabe 8 haben, werden wir uns auf die Beantwortung der Fragen aus den Aufgaben 9 und 12 konzentrieren.

### 4.1.3 Aufgabe 9 und 12

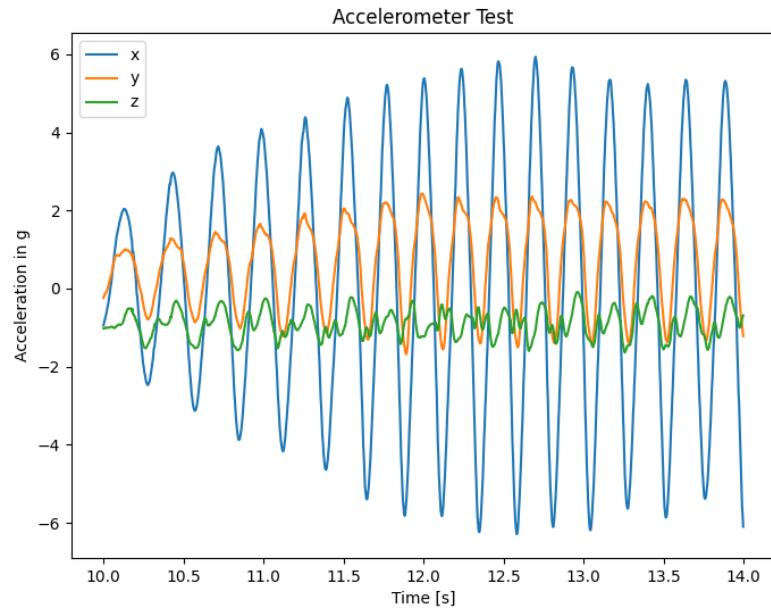


Abbildung 4: Abgabe 1 - Aufgabe 9

In Abbildung 4 ist die gemessene Beschleunigung des 10 Sekunden Tests auf einer Achse dargestellt. Die X-, Y- und Z-Achse sind farblich wie folgt gekennzeichnet:

- X-Achse: Blau
- Y-Achse: Orange
- Z-Achse: Grün

Es ist zu erkennen, dass die meiste Beschleunigung in der X-Achse auftritt, was darauf hinweist, dass die Bewegung hauptsächlich entlang dieser Achse stattgefunden hat. Die Y- und Z-Achsen zeigen nur geringe Schwankungen, was darauf hindeutet, dass die Bewegung in diesen Richtungen minimal war.

Folgende Anpassungen mussten vorgenommen werden, um die Messdaten korrekt aufzuzeichnen:

- Die Baudrate wurde auf 57600 Baud (Bits per second) geändert, damit die Schnittstelle und der Arduino/Sparkfun die gleiche Baudrate haben.
- setscale wurde von 2G auf 8G geändert, um die höheren Beschleunigungswerte korrekt zu erfassen und kein Clipping entsteht (Plateaus).
- Die Sample-Rate wurde von 1,56 Hz auf 800 Hz (`ODR_1` zu `ODR_800`) erhöht, um dem Nyquist-Shannon-Abtasttheorem gerecht zu werden und Aliasing-Effekte zu vermeiden. Bei einer Abtastrate von 800 Hz beträgt die Nyquist-Frequenz 400 Hz, was für die Erfassung der relevanten Signalkomponenten bei unserer Beschleunigungsmessung ausreichend ist.

Die Informationen zu den Anpassungen wurden dem Datenblatt des Sensors auf Seite 1 entnommen [1]. Mögliche Parameter für die Auswahl der setscale sind 2G, 4G und 8G. Um Plateaus zu vermeiden, sollte der höchste Wert (8G) gewählt werden, um Clipping zu vermeiden.

#### 4.1.4 Aufgabe 10

Aus den aufgezeichneten Daten der Abbildung 4 wurde die Berechnung der Messfrequenz durchgeführt. Zu unserer Überraschung ergab die Berechnung eine Abtastfrequenz von ca. 270 Hz, obwohl die Sample-Rate auf 800 Hz eingestellt war. Dies dürfte vor allem auf die eingestellte Baudrate von 57600 Baud zurückzuführen sein, da die Übertragung der Daten über die serielle Schnittstelle begrenzt ist.

#### 4.1.5 Aufgabe 13

Diese Aufgabe konnte leider nicht durchgeführt werden, da das Aufzeichnen von Messdaten mit Energieversorgung per 9V Blockbatterie nicht funktionierte. Auch die Verwendung des Reset-Buttons führte nicht zum Erfolg.

Welcher Unterschied besteht zwischen der Aufzeichnung mit und ohne Batterieversorgung? Die maximale Messfrequenz ist durch die Schreibgeschwindigkeit der SD-Karte begrenzt. Beim Auslesen der Daten über die serielle Schnittstelle ist die Übertragungsrate (Baudrate) der limitierende Faktor, dieser ist jedoch deutlich höher als die Schreibgeschwindigkeit der SD-Karte.

## Literaturverzeichnis

- [1] NXP Semiconductors, “Mma8452q: 3-axis, 12-bit/8-bit digital accelerometer,” <https://www.nxp.com/docs/en/data-sheet/MMA8452Q.pdf>, 2023, datasheet, accessed: 2025-11-19.

Bei der Überarbeitung von Textstellen und beim Erstellen von Code für das Einlesen und Plotten der Dataframes wurden Chat-GPT und Claude Sonnet verwendet.

## **Abbildungsverzeichnis**

1	Versuchsaufbau . . . . .	3
2	Abgabe 1 - Aufgabe 5 (a) . . . . .	5
3	Abgabe 1 - Aufgabe 6 . . . . .	6
4	Abgabe 1 - Aufgabe 9 . . . . .	7

## Tabellenverzeichnis