

*Praktikum*

*Forschungsprojekt*

*Anthropomatik praktisch*

*erfahren*

**28. Juli 2016**

Projekt 2: Andrea Bittner, Maren Kucza  
Analyse von Schüttgutverhalten unter  
Verwendung von integrierter Sensorik



# **Projekt 2: Analyse von Schüttgutverhalten unter Verwendung von integrierter Sensorik**

**– Praktikum: Forschungsprojekt Anthropomatik praktisch erfahren –**

**Andrea Bittner, Maren Kucza**

**28. Juli 2016**

## **Zusammenfassung**

In diesem Projekt wurde untersucht, ob mit Hilfe von instrumentiertem Schüttgut mehr Informationen über das Bewegungsverhalten von Schüttgütern gewonnen werden können, um den Sortierprozess einer Bandsortieranlage besser verstehen und optimieren zu können.

Es wurde ein instrumentiertes Schüttgut entwickelt, das in eine Überraschungsei-Kapsel passt und mit dem Beschleunigungs- und Orientierungsdaten gemessen werden können. Der Prototyp wurde zuerst in Machbarkeitsstudien evaluiert und anschließend entwickelt. Über eine Bluetooth Low Energy-Verbindung wurden gemessene Daten durch den Mikrocontroller an eine eigens implementierte Android-App übermittelt und gespeichert. Zusätzlich konnte über die App das Verhalten des Mikrocontrollers und des Speicherns auf dem Smartphone kontrolliert werden. Für die Analyse der Daten wurde ein in Python entwickelter Client verwendet, der die Rohdaten zunächst aufbereitete und anschließend visualisierte.

Im weiteren Verlauf wurden die gesammelten Daten ausgewertet, dabei wurde auch das Savitzky-Golay-Filter eingesetzt. Es konnten Bewegungsmuster in den Daten erkannt werden, die sich durch bestimmte Charakteristiken auch auf die Anlagenmodule abbilden ließen. Neben dem Fazit mit den gewonnenen Erkenntnissen aus Messreihen und Datenanalysen wurden Vorschläge für Erweiterungen gegeben.

Das Projekt basierte auf der FlexSort-Anlage, eine Bandsortieranlage des Fraunhofer IOSB in Karlsruhe.

## Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>5</b>
<b>2 Projektplanung</b>	<b>7</b>
2.1 Aufgabenstellung . . . . .	7
2.2 Zeitliche Planung . . . . .	8
<b>3 Arbeitsschritte</b>	<b>10</b>
3.1 Design des instrumentierten Schüttguts . . . . .	10
3.1.1 Kriterien für das instrumentierte Schüttgut . . . . .	10
3.1.2 Recherche . . . . .	11
3.1.3 Machbarkeitsstudie . . . . .	13
3.1.4 Prototyping . . . . .	15
3.1.5 Löten . . . . .	15
3.2 Programmierung des Mikrocontrollers . . . . .	17
3.2.1 Benötigte Software . . . . .	17
3.2.2 Programmierung des Mikrocontrollers . . . . .	17
3.3 Empfangen der Messwerte . . . . .	19
3.3.1 Python-Client für BLE-Verbindung . . . . .	19
3.3.2 Alternativer Client zum Empfang der Daten per BLE: Android-App . . .	19
3.4 Client zur Datenanalyse . . . . .	22
3.4.1 Graphisches User Interface . . . . .	22
3.4.2 Datenformat . . . . .	22
3.4.3 Parser . . . . .	23
3.4.4 Darstellen von Daten . . . . .	23
3.4.5 Datenaufbereitung . . . . .	24
<b>4 Datenmessung</b>	<b>25</b>
4.1 Erkenntnisse aus dem ersten Testlauf . . . . .	25
4.2 Erster Messlauf . . . . .	25
4.3 Zweiter Messlauf . . . . .	27

<b>5 Auswertung</b>	<b>28</b>
5.1 Datenanalyse . . . . .	28
5.2 Evaluation der Methode . . . . .	31
5.3 Fazit . . . . .	31
5.4 Ausblick . . . . .	32



## 1 Einleitung

Das Projekt basiert auf der FlexSort-Anlage, eine optische Bandsortieranlage für Schüttgut am Fraunhofer IOSB. Die Sortieranlage kann Schüttgut mit Hilfe einer Flächenkamera optisch klassifizieren und mittels Druckluftdüsen voneinander trennen. Damit eine konstante und gleich verteilte Menge von Schüttgut auf dem Band liegt, wird mittels Rüttler das Material freigegeben und rutscht über eine Rutsche auf das Band. Die nachfolgenden Abbildungen 1 bis 3 zeigen den FlexSort.



**Abbildung 1:** FlexSort: Schüttgut fällt vom oberen Querrüttler auf den kurzen Rüttler und rutscht dann auf das Sortierband



**Abbildung 2:** FlexSort: Sortierband mit optischer Objekterkennung und Druckluftdüsen zum Ausblasen



**Abbildung 3:** FlexSort: Außenansicht mit großem Rüttler und Förderbändern, um Schüttgut für geschlossenen Messlauf wieder nach oben zu befördern

Da die Klassifizierung und das Ausblasen von Teilchen etwas verzögert stattfinden, ist eine zeitlich gut geplante Aktivierung der Druckluftdüsen notwendig, um die Anlage möglichst kostensparend und effizient zu betreiben. Hierfür muss die genaue Position der auszusortierenden Teilchen zum Zeitpunkt der Düsenüberquerung ermittelt werden. Dabei ist zu beachten, dass sich das Schüttgut vom Klassifizierungszeitpunkt bis zum Zeitpunkt des Ausblasens auf dem Band bewegt. Die Bewegung der Teilchen kann unter Umständen von einzelnen Parametern der Anlage beeinflusst werden und wirkt sich dadurch auf das Sortierergebnis aus. Beispielsweise kann die Geschwindigkeit des Bandes dazu führen, dass die Teilchen darauf springen oder der Rüttler durch ungünstige Vibrationsbewegungen keine konstante Menge von Teilchen über die Rutsche auf das Band freigibt.

In diesem Projekt soll eine Möglichkeit gefunden werden, den Prozess des Sortierens von Schüttgut genauer zu verstehen und dadurch weitere Optimierungen an der Anlage und im Prozess vornehmen zu können. So könnte ein stabileres Sortierergebnis erreicht werden. Neben den rein optisch gewonnenen Daten können weitere Messwerte durch andere Verfahren unterstützend sein. In diesem Projekt soll dies mit Hilfe eines instrumentierten Schüttguts passieren, über das Bewegungsdaten gesammelt und ausgewertet werden können. Es soll untersucht werden, ob sich in den gewonnenen Daten bestimmte Bewegungsmuster erkennen lassen, die auf einzelne Anlagenmodule zurückgeführt werden können. Gegebenenfalls lassen sich zwischen den Anlagenmodulen verschiedene Korrelationen erkennen, die zukünftig zur Optimierung der Anlage genutzt werden können. Darüber hinaus könnte ein Qualitätsmaß erstellt werden, anhand dessen die gewählten Parameter der Anlage bewertet werden.

## 2 Projektplanung

### 2.1 Aufgabenstellung

Im Rahmen des Forschungspraktikums soll ein Schüttgut konstruiert werden, das über Sensorik verfügt, mit der Beschleunigungs- und Orientierungsdaten gemessen werden können. Das Schüttgut soll annähernd die Größe des zu sortierenden Schüttguts haben. Die maximale Größe ist jedoch durch die Kapsel eines Überraschungs-Eis limitiert, in der die Sensorik untergebracht werden soll. Zunächst muss solch ein instrumentiertes Schüttgut entworfen werden.

Nach der Recherche nach geeigneten Bauteilen soll ein Prototyp des Sensorik-Schüttguts erstellt werden, mit dem Daten auf einer Bandsortieranlage gewonnen werden können.

Zur Datengewinnung muss das Schüttgut programmiert werden, sodass die gelieferten Daten der Sensoren an eine Analysesoftware auf einem PC übertragen werden können. In einem weiteren Schritt sollen diese Daten ausgewertet werden. Hierfür muss eine Anwendung entwickelt werden, mit der sich die Daten verständlich darstellen und analysieren lassen. Dafür müssen die gemessenen Werte der Sensoren aufgearbeitet werden. Anschließend sollen sie in Korrelation zu den einzelnen Anlagenmodulen gebracht werden. Hierfür wäre es notwendig, charakteristische Bewegungsmuster für die einzelnen Module in den Daten zu finden. Abschließend soll validiert werden, ob mit Hilfe von instrumentierten Schüttguts ein verbessertes Verständnis für den Sortierprozess der Bandsortieranlage erzielt werden kann.

Folgende Punkte beschreiben Herausforderungen, die großen Einfluss auf den Projektverlauf haben könnten:

**Begrenzte Größe:** Die Größe der Sensorik bestimmt zum einen, auf welcher Anlage anschließende Messungen durchgeführt werden können, da das instrumentierte Schüttgut zur Größenordnung des zu sortierenden Schüttgutes passen sollte. Zum Anderen wirkt sich dies stark auf die Wahl der verwendeten Bauteile aus, die im verbundenen Zustand Platz in der Kapsel eines Ü-Eis finden müssen.

**Implementierung:** Von Sensoren gesammelte Daten müssen an den PC weitergeleitet werden. Hierfür gibt es verschiedene Möglichkeiten (Daten loggen oder per Funk direkt übertragen).

**Bewertung der Daten:** Die von den Sensoren gelieferten Daten müssen aufbereitet werden, bevor sie interpretiert werden können.

**Korrelation zwischen gemessenen Daten und Anlagenmodulen:** Finden einer Korrelation und Erkennen von charakteristischen Bewegungsmustern sind für die Evaluation dieser Methode notwendig, um festzustellen, ob durch instrumentiertes Schüttgut weitere Daten für die Optimierung der Anlage zu gewinnen sind.

## 2.2 Zeitliche Planung

Die beschriebenen Herausforderungen spiegeln sich auch in den Meilensteinen des Projektes wieder:

**Meilenstein 1** beinhaltet das Design eines Schüttguts mit Sensorik, welches die maximale Größe eines Ü-Eis hat. Zusätzlich wurde eine Machbarkeitsstudie durchgeführt, mit der die Wahl der Bauteile begründet werden kann.

**Meilenstein 2** beinhaltet die Fertigung und Programmierung des Schüttguts mit Sensorik.

Nach Abschluss liegt ein fertiges und funktionsfähiges instrumentiertes Schüttgut vor, das gemessene Daten per Bluetooth überträgt.

**Meilenstein 3** umfasst das Sammeln und Darstellen von Daten aus der Anlage mit Hilfe des instrumentierten Schüttguts. Nach ausreichender Anzahl von Messdaten, Aufbereitung sowie Darstellung in einem verständlichen Format ist der Meilenstein erreicht.

**Meilenstein 4** beinhaltet die Analyse der Daten und die Erkennung von charakteristischen Bewegungsmustern, die sich zu den Anlagenmodulen zuordnen lassen.

**Meilenstein 5** schließt das Projekt mit der Evaluation der Ergebnisse ab. Es liegt nach Projektende eine Bewertung für das Verfahren vor, in dem mit weiteren gewonnenen Daten (außer den optischen) der Sortierprozess positiv beeinflusst werden kann.

Um das neue Verfahren zur Datengewinnung zu evaluieren, sind folgende Arbeitsschritte wie in Abbildung 4 im zeitlichen Verlauf geplant und durchgeführt worden.

Die ursprüngliche Planung hatte folgende Meilensteinverteilung vorgesehen: Meilenstein 1 ist für die ersten drei Projektwochen angesetzt. Kalenderwoche 19 bei Aufgabe 3 wird für die Bestelldauer der Bauteile geblockt. Solange kann nicht mit Aufgabe 4 begonnen werden. Allerdings können einzelne Teilaufgaben aus Aufgabe 5 vorgezogen werden. Fortführend werden die Aufgaben 4 und 5 parallel bearbeitet. Mit Meilenstein 2 nach Kalenderwoche 23 existiert ein fertiges instrumentiertes Schüttgut. Meilenstein 3 wird zwei Wochen später nach Beenden von Aufgabenteil 5 und dem Sammeln von Daten an der Anlage erreicht. Ab Kalenderwoche 26 beginnt die Analyse der Daten für den Abschluss von Meilenstein 4. Die letzten beiden Wochen sind für die Evaluation des Projektes angesetzt, womit auch Meilenstein 5 erreicht wird. Zusätzlich sollte das Projekt fortlaufend dokumentiert sowie drei Präsentationen ausgearbeitet werden.

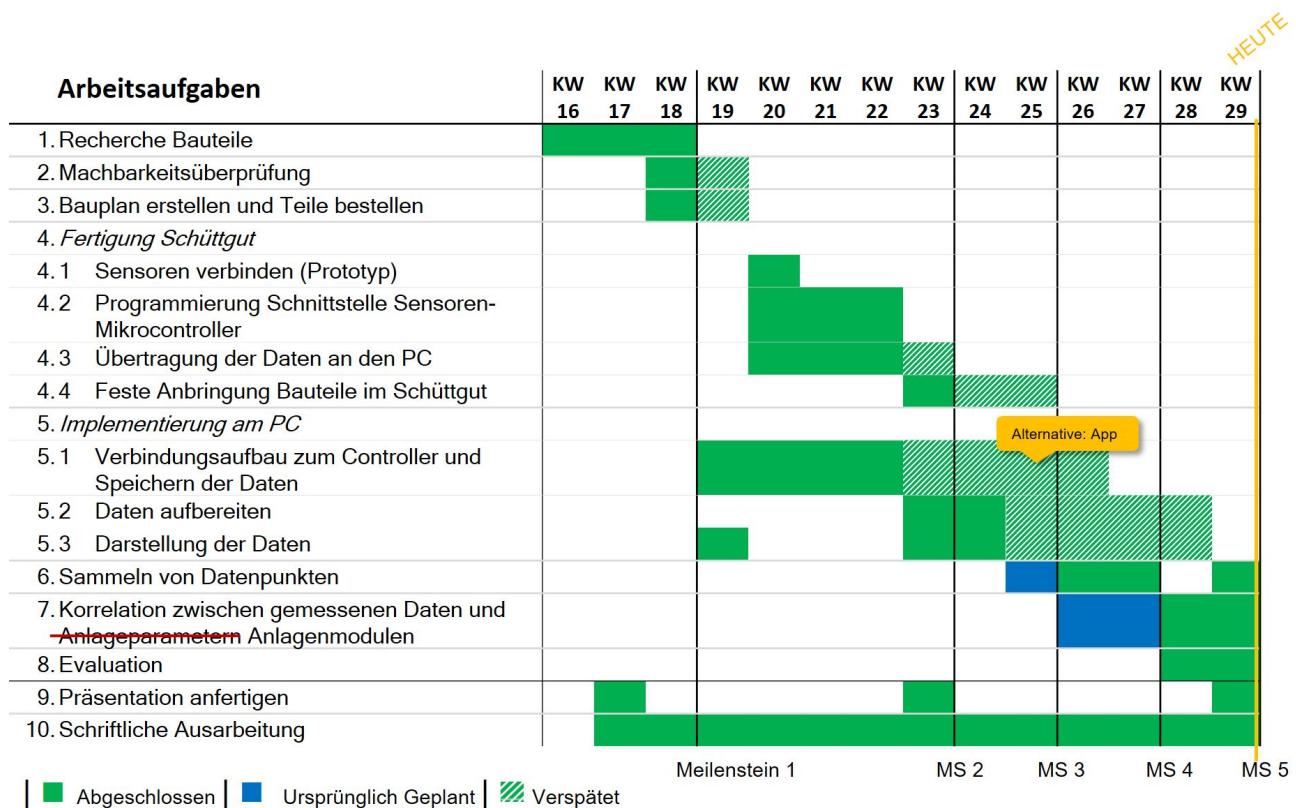
In der Abbildung markieren grüne Wochen die sowohl geplante als auch zu dem Zeitpunkt durchgeführte Aufgabe. Grün schraffierte Wochen kennzeichnen, dass die Aufgabe länger als geplant bearbeitet wurde. Blau zeigt dagegen die ursprüngliche Zeitplanung der Aufgabe, die allerdings nicht eingehalten werden konnte. Wie in der Abbildung zu sehen ist, kam es zu zwei Zeitverlusten. Das Ergebnis der Machbarkeitsstudie hätte sich negativ auf die Qualität des Projektergebnisses auswirken können, weshalb der Zeitverzug begründet werden konnte. Zusätzlich

konnte wieder durch eine schnelle Lieferung ausgeglichen werden, für die ein Puffer von einer Woche angesetzt war.

Der größere Zeitverzug entstand bei Aufgabe 5.1, bei der es zu Problemen bei der Implementierung des Clients kam, mit dem eine Verbindung zum Controller für die Datenübertragung aufgebaut werden musste. Nach Überschreiten der Frist für den ersten Ansatz wurde an einer Alternative gearbeitet. Durch den mehrwöchigen Verzug wurden Aufgabe 6 und 7 etwas verschoben durchgeführt.

Wie bei Aufgabe 7 sichtbar, sollte ursprünglich eine Korrelation zwischen Daten und Anlagenparametern gefunden werden, womit ein Qualitätsmaß für die Sortieranlage bestimmt werden kann, den es zu optimieren galt. Aufgrund des Anlagenaufbaus, bei dem sich das Schüttgut für Forschungszwecke in einem Kreislauf bewegt, konnten keine Messungen mit verschiedenen Anlagenparametern durchgeführt werden, da die Module bereits sehr genau aufeinander abgestimmt sind. Daher wurde die Aufgabe nachträglich angepasst: Es sollte eine Korrelation zwischen gemessenen Daten und den Anlagenmodulen gefunden werden.

Das Projekt konnte rechtzeitig fertig gestellt werden.



**Abbildung 4:** Projektplanung runtergebrochen auf einzelne Arbeitsschritte; Update nach Projektende am 22.07.2016

### 3 Arbeitsschritte

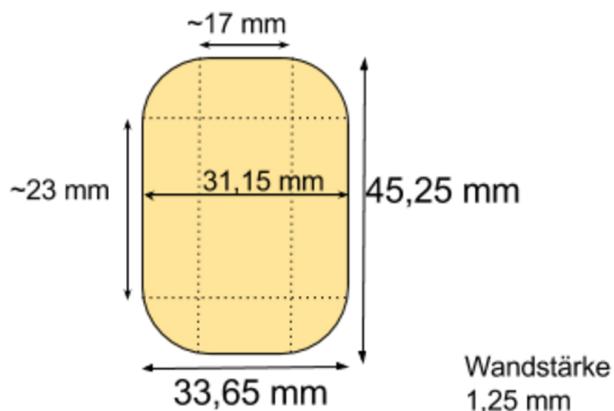
#### 3.1 Design des instrumentierten Schüttguts

##### 3.1.1 Kriterien für das instrumentierte Schüttgut

Allgemeine Anforderungen an das instrumentierte Schüttgut:

- Möglichst klein, maximale Größe der Module <sup>1)</sup>
- Aufnehmen/Speichern von Bewegungsdaten
- Übertragung von Daten an einen PC
- Betreiben durch interne Batterie
- Optional: Cachen von Daten, bis diese ausgelesen werden

<sup>1)</sup> Das instrumentierte Schüttgut soll in eine Kapsel eines Kinder-Überraschungseis passen. Dadurch wird die maximale Größe des Schüttguts festgelegt (siehe Abbildung 5). Die Wahl der Ü-Ei-Kapsel als Hülle für das Schüttgut eignet sich dahingehend gut, dass es einerseits den Mikrocontroller schützt, andererseits sehr einfach zu beschaffen ist und kein Behältnis aufwendig produziert werden muss (zum Beispiel mittels 3D-Druck).



**Abbildung 5:** Maße der Kapsel aus einem Kinder Überraschungsei

Die Beschaffung der Bauteile sollte zudem möglichst einfach sein. Von Vorteil ist, wenn alle Bauteile von einem deutschen Händler bezogen werden können, um einerseits nur einen einzelnen Liefervorgang zu haben und andererseits eine geringere Lieferzeit (anstatt international). Auch eine ausreichende Dokumentation der einzelnen Module für eine schnelle Implementierung des Mikrocontroller-Codes ist sehr nützlich. Die einzelnen Module müssen kompatibel zueinander sein.

### 3.1.2 Recherche

Begonnen wurde mit einer Recherche nach Bauteilen, mit denen sich solch ein instrumentiertes Schüttgut bauen lässt. Als Hauptplatine wird ein Mikrocontroller benötigt, an dem weitere Bauteile angeschlossen werden können. Es gibt einzelne Sensormodule, die z.B. Orientierung und Beschleunigung in je 3 Achsen gleichzeitig messen können. Ebenso gibt es verschiedene Speicher- und Übertragungsmodule, die Daten auf eine SD-Karte loggen oder per Bluetooth / WLAN / ZigBee weiterleiten können. Ausschlaggebend bei der Wahl der einzelnen Module war zunächst die Größe der einzelnen Bauteile, die die Größe eines Ü-Eis nicht übersteigen dürfen. Nach der Wahl eines passenden Mikrocontrollers müssen auch die Anschlussmöglichkeiten für weitere Module beachtet werden.

**Mikrocontroller**, Gewähltes Bauteil: Adafruit Pro Trinket 3V 12MHz

Das Pro Trinket 3V 12MHz von Adafruit besitzt einen ATmega328-Chip, der auch auf einigen Arduino-Mikrocontrollern verwendet wird. Er besitzt einen Speicher von 28KB und einen RAM von 2KB. Ein Vorteil ist außerdem, dass er mit 18 GPIOs (General Purpose Input Output) und 2 analogen Anschlüssen weitaus mehr Anschlussmöglichkeiten bietet, als beispielsweise der Arduino Pro Mini. Die Maße des Boards sind 38mm x 18mm x 4mm, sodass die Platine ausreichend Platz in der Kapsel haben sollte. Auf dem Trinket ist ein USB Bootloader vorinstalliert, sodass der Mikrocontroller per USB programmiert werden kann. (Quelle: <https://www.adafruit.com/products/2010>, 12.05.2016)

**Aufnahme von Bewegungsdaten**, Gewähltes Bauteil: Adafruit 9-DOF IMU Breakout - L3GD20H + LSM303

Das Modul L3GD20H kann drei verschiedene Daten in jeweils drei Achsen messen: Beschleunigungsdaten, Rotationsdaten und Magnetische Daten ähnlich zu einem Kompass. Da Beschleunigungs- und Gyroskopdaten gemessen werden sollen, ist ein Bauteil, das beide Daten zusammen liefern kann, sehr platzeffizient. Für die Kommunikation zwischen dem Mikrocontroller und dem Datensor wird das I2C-Protokoll benutzt. Das Bauteil hat die Maße 38mm x 23mm und ist damit um 5 Millimeter breiter als der Mikrocontroller Pro Trinket. (Quelle: <https://www.adafruit.com/products/1714>, 12.05.2016)

### Übertragung der Daten

Zur möglichst schnellen Übertragung von Daten an den PC soll eine drahtlose Verbindung verwendet werden. Dafür ergeben sich drei verschiedene Möglichkeiten:

- ZigBee: Ist zu aufwändig, da ein separates Modul für den PC benötigt wird. Hingegen sind fast alle Laptops heute standardmäßig mit WLAN und Bluetooth ausgestattet.
- WLAN: Die zur Auswahl stehenden Platinen sind entweder deutlich zu groß oder in ihrer Verwendung zu heikel, da sie extrem empfindlich auf kleinste Spannungsschwankungen reagieren. Zudem ist der Stromverbrauch größer als bei Bluetooth. Da das Schüttgut nur durch eine sehr kleine Batterie betrieben werden kann, ist dies hinderlich.

- Bluetooth: Bluetooth ist weit verbreitet und die Module sind klein genug, um sie mit anderen Bauteilen im Ü-Ei unterzubringen.

Gewähltes Bauteil: Adafruit Bluefruit LE SPI Friend - Bluetooth Low Energy (BLE)

Gewählt wurde das Adafruit Bluefruit LE SPI Friend, das durch Bluetooth Low Energy einen optimierten Energieverbrauch hat. Zur Kommunikation mit dem Mikrocontroller wird das Protokoll SPI benutzt. Mit den Maßen 23mm x 26mm x 5mm ist es dicker als die anderen Bauteile, aufgrund der geringeren Breite sollte das Bauteil dennoch in der Rundung des Ü-Eis Platz finden. Das Modul hat außerdem einen Speicherplatz von 256KB. (Quelle: <https://www.adafruit.com/products/2633>, 12.05.2016)

**Betrieb durch interne Batterie**, Gewählte Bauteile: Adafruit Pro Trinket LiIon/LiPoly Backpack Add-On, Lithium Ion Polymer Battery - 3.7V 150mAh

Um einen universellen Einsatz des instrumentierten Schüttguts zu garantieren, muss es mit einer eigenen Batterie oder einem eigenen Akkumulator betrieben werden. Mit dem Adafruit Trinket Lilon Backpack Add-On lässt sich ein Akkumulator an das Pro Trinket anschließen. Das Add-On dient als Ladestation, wenn das Pro Trinket per USB mit einem Rechner verbunden ist. Sobald das USB-Kabel entfernt wird, schaltet das Modul automatisch in den batteriebetriebenen Modus. Das Modul hat eine Größe von 15mm x 17mm x 7mm. Die relativ große Höhe im Vergleich zu den anderen Modulen kommt durch den Anschluss für den Akkumulator zustande. (Quelle: <https://www.adafruit.com/products/2124>, 12.05.2016)

Hinzu kommt eine Lithium Ion Polymer Batterie mit 3,7V und 150mAh. Dieser Akku wurde aufgrund der geringen Größe von 19,75mm x 26mm x 3,8mm gewählt. Es gibt auch Modelle mit mehr Kapazität, die aber andere Abmessungen besitzen. Zum Teil geht die Länge über 30mm hinaus. Es werden allerdings schon Bauteile über diese Länge verwendet, die in der vertikalen Achse der Kapsel platziert werden müssen. Der Akku sollte daher näher an der Außenwand positioniert werden, wo die Höhe durch die Abrundung der Kapsel jedoch eingeschränkt wird. (Quelle: <https://www.adafruit.com/products/1317>, 12.05.2016)

### Alternative Überlegungen zum Cachen der Daten

Da die meisten Mikrocontroller nur sehr wenig eigenen Speicher haben und eine drahtlose Datenverbindung nicht zuverlässig genug ist, müssen die vom Sensor gelesenen Daten auf einem extra hinzugefügten Speicher zwischengespeichert werden, bis sie vom PC ausgelesen werden können.

Mögliche Bauteile: Adafruit I2C Non-Volatile FRAM Breakout - 256Kbit / 32KByte (Quelle: <https://www.adafruit.com/products/1895>, 12.05.2016), Adafruit SPI Non-Volatile FRAM Breakout - 64Kbit / 8KByte (Quelle: <https://www.adafruit.com/products/1897>, 12.05.2016)

Vorerst werden diese Bauteile nicht weiter betrachtet, da das Bluetooth-Modul über 256 KB Speicher verfügt, die als Puffer genutzt werden. Mit den genannten Bauteilen lässt sich der Speicher des Mikrocontrollers von den integrierten 28 KB (schon abzüglich der 4 KB für die Dateien des Bootloaders) auf um 8 bzw. 32 KB erhöhen.

Die gewählten Bauteile für das instrumentierte Schüttgut (siehe Tabelle 1) sind alle von der Marke Adafruit. Der Vorteil darin liegt in der Kompatibilität der Bauteile aufgrund der Abstimmung zueinander, sowie in der sehr ausführlichen Dokumentation dieser. Tabelle 2 beinhaltet weitere zur Verfügung stehende Hardware zur Durchführung des Projektes.

## Verwendete Hardware

Bauteil	Menge
Adafruit Pro Trinket 3V 12MHz	3
Adafruit 9-DOF IMU Breakout - L3GD20H + LSM303	3
Adafruit Bluefruit LE SPI Friend - Bluetooth Low Energy (BLE)	3
Adafruit Pro Trinket LiIon/LiPoly Backpack Add-On	3
Lithium Ion Polymer Battery - 3.7v 100mAh <sup>*)</sup>	9

**Tabelle 1:** Bestellte Hardware

<sup>\*)</sup> Aufgrund der Nichtverfügbarkeit des gewünschten Akkus bei einem Lieferanten musste ein anderer Akku gewählt werden. Dadurch konnte die Bestellung aber bei nur einem Lieferanten getätigkt und eine kurze Lieferzeit erreicht werden.

Materialien	Menge
Breadboard	2
Set mit Steckkabeln	2
Arduino Uno	1
Arduino Duemilanove	1
Ardunio Feather	1

**Tabelle 2:** Weitere Hilfsmittel stehen zur Verfügung

### 3.1.3 Machbarkeitsstudie

Vor der Bestellung wurden noch einige Machbarkeitsstudien durchgeführt. Neben dem Vergleich gegenüber anderen verfügbaren Modulen wurde ein Größentest durchgeführt und eine Wiring-Skizze angefertigt.

Bei dem Größentest wurden die einzelnen Module maßstabsgerecht mit Moosgummi nachgebaut. Dadurch konnte getestet werden, wie sich die Module am besten in der Ü-Ei-Kapsel anordnen lassen. Zusätzlich sollte in der Kapsel aber noch genügend Freiraum sein, da auch die einzelnen Kabelverbindungen zwischen den Modulen Platz benötigen.

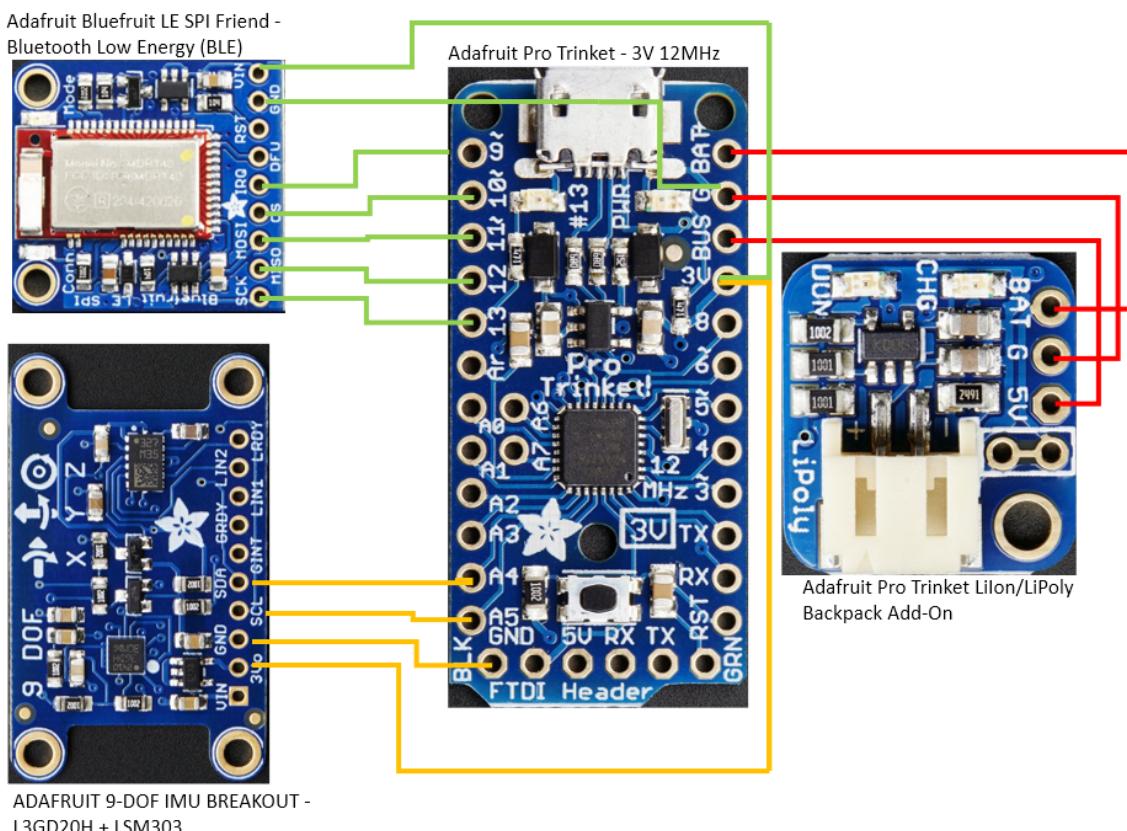
Wie in Abbildung 6 ersichtlich wird, passen alle Module in die Kapsel und es besteht noch ausreichend Platz für Verbindungen.

Die Wiring-Skizze in Abbildung 7 zeigt die Verbindungen der einzelnen Module untereinander. Diese ist nicht maßstabsgerecht. Ziel war es, alle Module korrekt miteinander verbinden zu



**Abbildung 6:** Machbarkeitsstudie: Nachbau der Module mit Hilfe von Moosgummi

können. Damit konnten auch schon erste Überlegungen angestellt werden, wie die Module später in der Kapsel angeordnet werden könnten. Letztendlich wird die Skizze auch hilfreich beim Löten sein.

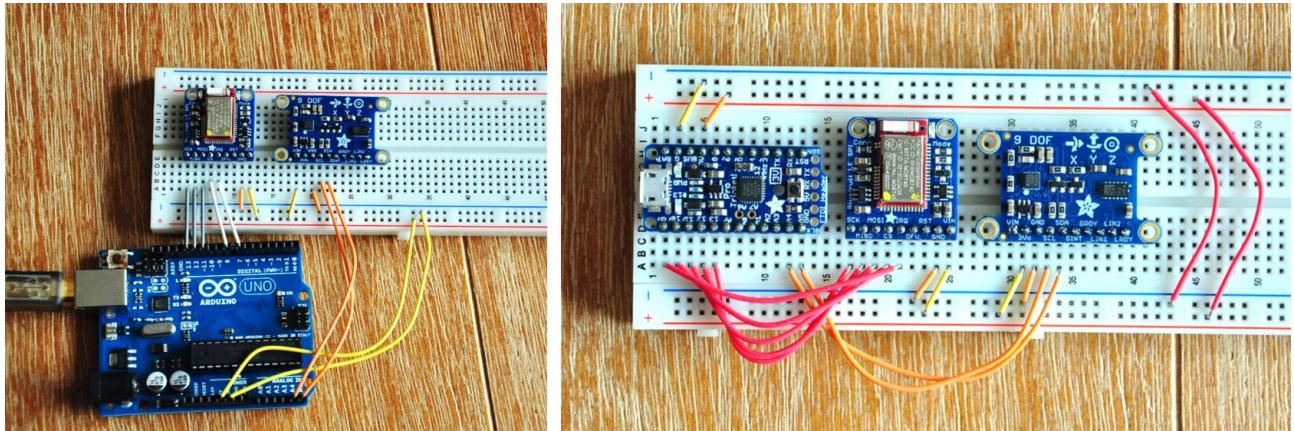


**Abbildung 7:** Wiring-Skizze, wie die Bauteile miteinander verbunden werden

Nach positiven Ergebnissen aus den Machbarkeitsstudien wurden die Bauteile wie sie in Tabelle 1 aufgeführt sind, bestellt. Da die Bestellung bei einem Händler getätigt wurde, waren die Bauteile in der folgenden Woche schon verfügbar.

### 3.1.4 Prototyping

Für einen ersten Prototypen wurden die Bauteile mit Hilfe eines Breadboards und Steckkabeln, wie in Abbildung 8 zu sehen, miteinander verbunden. Zuerst wurden die Pro Trinkets getestet. Hier ist leider ein Defekt eines Pro Trinkets aufgefallen. Anschließend wurde das Bluetooth-Modul und zum Schluss das Sensor-Modul hinzugefügt.

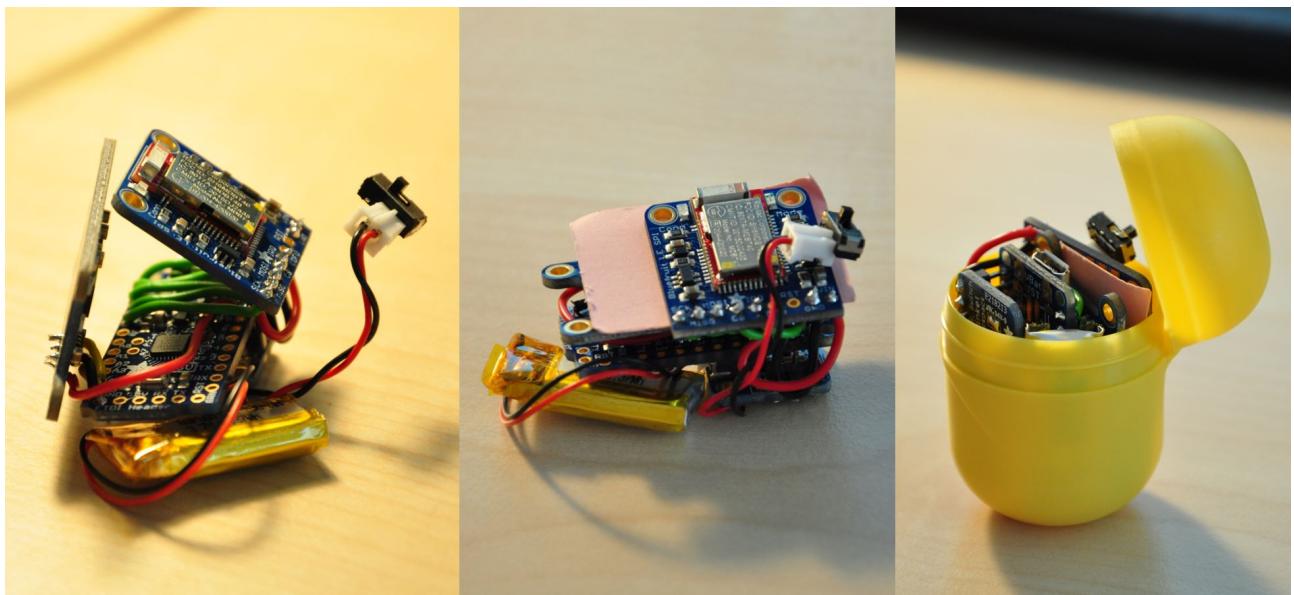


**Abbildung 8:** Prototyping mit dem Arduino Uno (links) und dem Adafruit Pro Trinket (rechts)

An dieser Stelle wurde außer dem Adafruit Pro Trinket noch ein Arduino Uno zur Hilfe genommen, um während der Programmierung des Mikrocontrollers (siehe Kapitel 3.2) auch mit einer seriellen Ausgabe testen zu können. Nach erfolgreichem Testen des Programmcodes wurde dieser auf den Adafruit Pro Trinket übertragen.

### 3.1.5 Löten

Da drei Bausätze vorhanden waren und nur einer für das Prototyping genutzt wurde, konnte mit dem Löten des ersten Bausatzes begonnen werden. Abbildung 9 zeigt das erste fertig gelötete Bauteil, das in eine Ü-Ei-Kapsel passen soll.



**Abbildung 9:** Links: das erste gelötete Bauteil; Mitte: zwischen dem Bluetooth- und dem Sensor-Modul wird Papier zur Isolierung verwendet; Rechts: das Bauteil passt in eine Ü-Ei-Kapsel

## 3.2 Programmierung des Mikrocontrollers

### 3.2.1 Benötigte Software

Folgende Software und Treiber mussten zur Programmierung des Mikrocontrollers installiert werden.

**Entwicklungsumgebung:** Arduino IDE (1.6.8), für die Programmierung des Arduino Chip ATmega328, der auf dem Adafruit Pro Trinket verbaut ist

**Bootloader:** Innerhalb der Arduino IDE muss der Bootloader Adafruit AVR Boards (1.4.7) installiert werden

**Treiber für die Adafruit Module:** Adafruit\_driver, Adafruit\_9DOF, Adafruit\_BluefruitLE\_nRF51, Adafruit\_Sensor, Adafruit\_LSM303LHC, Adafruit\_L3GD20\_U, Adafruit\_BMP085\_Unified

Adafruit stellt außerdem über Google Play die Android-App „Adafruit Bluefruit LE Connect“ zur Verfügung, über die mit Hilfe der beigefügten Testbeispiele in den Treiberpacketen der Mikrocontroller mit dem Smartphone verbunden und ausprobiert werden kann.

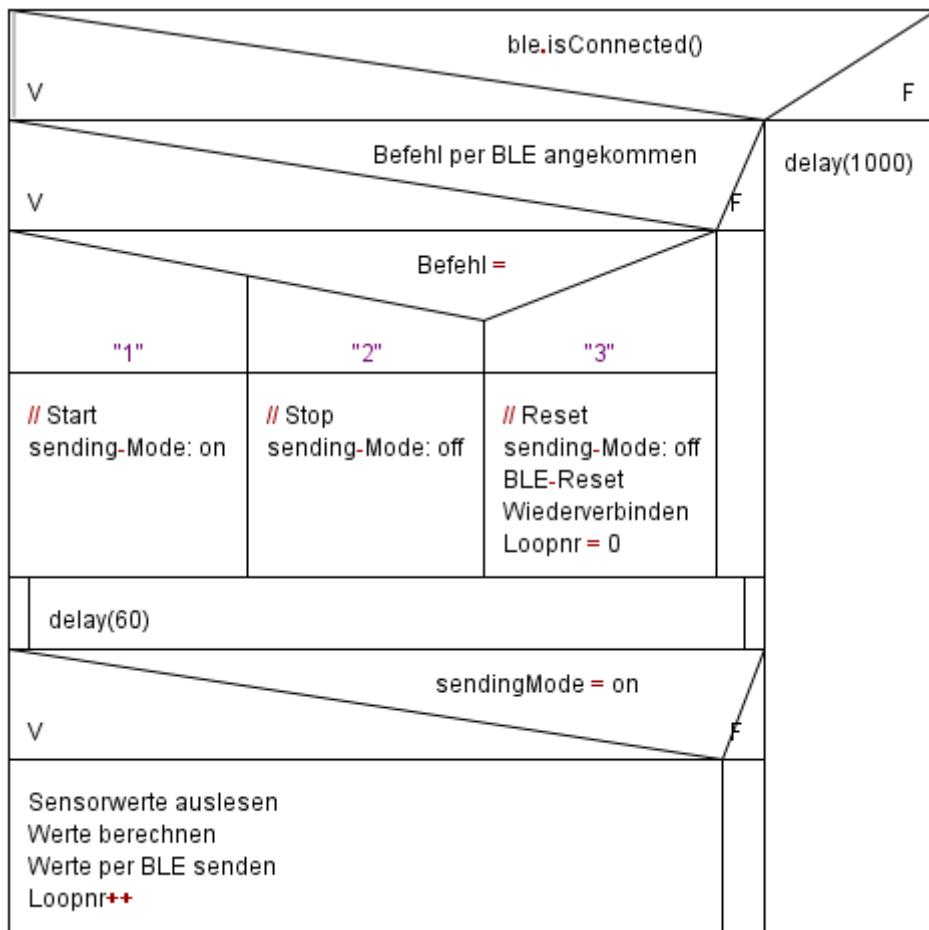
Für das Projekt wurde eine eigene Android-App entwickelt, die in Kapitel 3.3.2 beschrieben wird.

### 3.2.2 Programmierung des Mikrocontrollers

Da der Adafruit Pro Trinket keinen seriellen Ausgabeport hat, wurde die Programmierung zuerst auf einen Arduino Uno geladen, der wie auf der linken Seite der Abbildung 8 mittels eines Breadboards mit den Adafruit-Modulen verbunden war. Während der Programmierung konnte mit `Serial.print()`-Befehlen der Programmablauf auf dem Mikrocontroller kontrolliert werden, bis er anschließend mit dem gewünschten Verhalten auf den Adafruit Pro Trinket übertragen werden konnte.

Nach dem ersten Testlauf wurde der Programmcode des Mikrocontrollers nochmals überarbeitet und auch die Android-App zum Datenempfang (siehe Kapitel 3.3.2) dazu angepasst. Die Datei `BluefruitConfig.h` beinhaltet die Belegungen der verwendeten Pinouts des Mikrocontrollers zu den funktionalen Anschlüssen der anderen Modulen. Der eigentliche Programmcode befinden sich in der Datei `Program.ino`, die im nachfolgenden grob erklärt wird.

Zunächst werden Bibliotheken für die Adafruit Module eingebunden, sowie eine BLE-Instanz erstellt, die eine Bluetooth Low Energy-Verbindung aufbaut. Außerdem gibt es eine `Setup()`-Funktion, die nur einmal beim Start des Mikrocontrollers aufgerufen wird. Hier wird eine Bluetooth-Verbindung mit einem Smartphone hergestellt und die Sensoren initialisiert. Danach folgt die `Loop()`-Funktion, die wiederholt aufgerufen wird. Da diese auch auf die gesendeten



**Abbildung 10:** Struktogramm des Codeinhaltes der `Loop()`-Methode

Befehle der Android-App reagieren muss, wird der Ablauf der Funktion mittels Struktogramm in Abbildung 10 veranschaulicht.

Besteht keine BLE-Verbindung, wartet der Mikrocontroller 1000ms bis er wieder mit dem erneuteten Aufruf der Funktion `Loop()` beginnt. Bei bestehender Verbindung werden zunächst eventuell empfangene Befehle der Android-App ausgewertet. Durch `delay(60)` wartet der Mikrocontroller 60 Millisekunden. Diese Pause ist notwendig, da es ansonsten zu Problemen beim Auswerten des eingehenden Befehls kommt, speziell wenn die BLE-Verbindung zurückgesetzt wird. Sollen Daten gesendet werden (`sending-Mode = on`), wird mit dem Auslesen der Sensor-Werte und dem Senden dieser per BLE fortgefahren.

### 3.3 Empfangen der Messwerte

#### 3.3.1 Python-Client für BLE-Verbindung

Der Client wurde in Python geschrieben, da die Programmiersprache zum einen leicht zu verwenden ist und zum anderen wird von Adafruit die Bibliothek „Adafruit\_Python\_BluefruitLE“ zum Verbindungsauflauf mit BLE-Modulen bereitgestellt. Durch die beigefügten Beispiele hätte der Verbindungsauflauf mit dem Laptop einfach zu realisieren und somit eine Echtzeitdarstellung der ankommenden Daten möglich sein sollen. Vorher nicht bekannt war jedoch der Entwicklungsstand sowie der geringe Funktionsumfang dieser Bibliothek. Da in der kurzen Zeit keine Möglichkeit gefunden werden konnte, basierend auf diesen Klassen eine BLE-Verbindung aufzubauen, wurde mit der Entwicklung einer Alternative begonnen.

#### 3.3.2 Alternativer Client zum Empfang der Daten per BLE: Android-App

Da es bereits die Android-App „Adafruit Bluefruit LE Connect“ gibt, mit der Daten per Bluetooth empfangen und gesendet werden können, konnte davon ausgegangen werden, dass in der Konstellation Smartphone-Bluefruit-Modul BLE-Verbindungen funktionieren und unterstützt werden. Daraus ergibt sich sogar der Vorteil, dass ein Smartphone als Empfänger wesentlich mobiler ist, um an der Anlage am Sender mitlaufen zu können.

Von Adafruit gibt es über Github<sup>1</sup> die Klasse „Adafruit Android BLE UART“, mit der eine BLE-Verbindung zwischen einem Bluefruit-Modul und einem Android-Smartphone hergestellt werden kann. Dabei handelt es sich um eine abgespeckte Variante der oben erwähnten App. Basierend auf dem Quellcode für BLE-UART-Verbindungen wurde eine eigene App entwickelt. Als Entwicklungsumgebung kam Android Studio zum Einsatz. Neben dem Verbindungsauflauf muss die App Befehle an den Mikrocontroller senden können. Sobald Sensordaten vom Mikrocontroller ankommen, sollen diese Werte in lokalen Dateien gespeichert werden.

Zuerst musste eine Verbindung mit dem Bluefruit-Modul hergestellt werden. Bei eingeschaltetem Bluetooth am Smartphone sucht die App automatisch nach BLE-fähigen Geräten in der Nähe.

Die App verfügt über vier Buttons, um das Verhalten des Mikrocontrollers sowie das Speichern der Daten kontrollieren zu können:

**Start:** Sendet einen Start-Befehl an den Mikrocontroller, damit dieser mit dem Senden von gemessenen Sensordaten beginnt.

**Stop:** Sendet einen Stop-Befehl an den Mikrocontroller, damit dieser mit dem Senden von gemessenen Sensordaten stoppt. Die gemessenen Daten bleiben dabei in einem StringBuffer vorhanden.

<sup>1</sup>Github-Link: [https://github.com/adafruit/Adafruit\\_Android\\_BLE\\_UART](https://github.com/adafruit/Adafruit_Android_BLE_UART), 25.06.2016

**Save:** Speichert die gemessenen Daten in einer neuen Datei ab. Dabei werden genutzte Zähler nicht zurückgesetzt. Der Button kann benutzt werden, um bei einem Messlauf markante Punkte auf der Sortieranlage zu markieren, indem eine neue Datei erstellt wird. Zusätzlich wird mit dem Zwischenspeichern die Performance verbessert.

**Reset & Save:** Speichert die gemessenen Werte in eine Datei und setzt anschließend genutzte Zähler zurück. Zusätzlich wird ein Reset-Befehl an den Mikrocontroller gesendet, um die BLE-Verbindung neu zu starten und ebenfalls Zähler zurückzusetzen. Bei erfolgreicher Wiederverbindung, beginnt direkt wieder der Datenempfang, sofern der Mikrocontroller vorher auch im Senden-Modus war. Der Button kann benutzt werden, um bei einem Messlauf eine neue Runde auf der Anlage zu markieren, sowie eine gute Performanz zu behalten, indem gespeicherte Felder nicht zu groß werden.



**Abbildung 11:** Android-App zum Steuern des Mikrocontrollers und Empfangen der Sensordaten

Die Daten werden in Dateien lokal auf dem Smartphone gespeichert und müssen anschließend auf einen Rechner übertragen werden. Derzeit ist der „Download“-Ordner des Smartphones fest implementiert. Hier gibt es sicher noch Potential zur Erweiterung, indem die Daten direkt über

einen Data Service an einen PC oder Laptop weitergeleitet werden oder zumindest der Speicherort selbst wählbar ist.

Abgelegt werden die Dateien in einem bestimmten Namensformat. Sie beginnen mit dem aktuellen Timestamp, um identische Dateinamen zu verhindern. Zusätzlich wird noch der Bereich mittels eines internen App-Zählers angegeben, um zusammenhängende Dateien für einen Messlauf identifizieren zu können.

### 3.4 Client zur Datenanalyse

Der Client zur Datenanalyse wurde in Python entwickelt, da Python leicht erlernbar ist und gute Bibliotheken für eine Vielfalt von Anwendungsfällen bietet.

#### 3.4.1 Graphisches User Interface

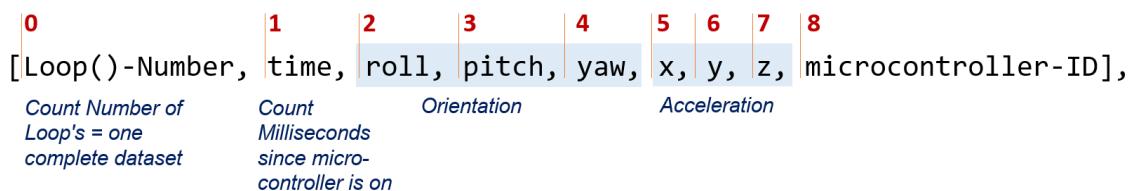
Für die Darstellung graphischer Bedienelemente im Client wurde die Library **PyQt4** verwendet, die eine leichte Integration der Sensordaten als Graphen ermöglicht. Diese werden mit Hilfe einer anderen Bibliothek erzeugt, können aber als ein sogenanntes Widget leicht in die GUI integriert werden. Da keine Echtzeitübertragung der Daten realisiert wurde, werden nur wenige Bedienelemente, welche in einer Menüleiste untergebracht sind, benötigt:

- Öffnen einer Datei mit Messwerten
- Anzeigen von Statistiken über den Messlauf (Dauer des Messlaufs, gemessene minimal und maximal Werte)
- Zusammenfügen von Dateien, wenn ein Messlauf in verschiedenen Dateien gespeichert ist
- Parametrisieren und Anwenden des Filters

Werden vom Nutzer ungültige Eingaben gemacht, wird ihm dieser Fehler durch Dialogfenster mitgeteilt.

#### 3.4.2 Datenformat

Um alle weiteren Verarbeitungsschritte zu erleichtern, wird vorab ein festes Format der Datentupel (Abbildung 12) definiert. Die ausgelesenen Sensordaten werden in diesem Format zusammen mit einem Zeitstempel, einer fortlaufenden Nummer und einer ID, welches den verwendeten Mikrocontroller identifiziert, gesendet.



**Abbildung 12:** Format der Datentupel

### 3.4.3 Parser

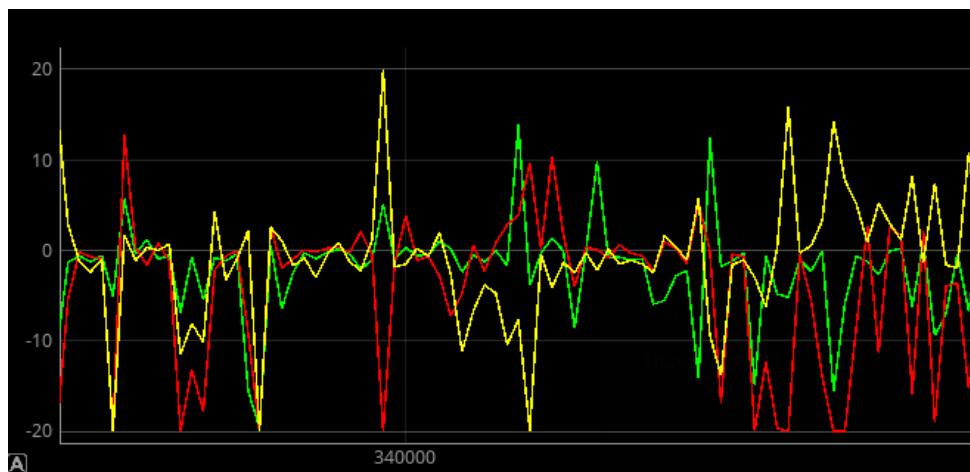
Der Parser operiert auf dem oben beschriebenen Datenformat und hat zwei Hauptaufgaben:

- Bereinigen der Daten, falls unvollständige oder inkorrekte Tupel vorkommen
- Aufteilen der Daten in Arrays, welche dann als Graph dargestellt werden können

Durch Eingabe einer Datendatei werden sechs Listen erzeugt, welche zur Darstellung an die Klasse **Plotter** übergeben werden. Die ersten drei Listen enthalten die Daten der drei Gyroskop-Achsen, die letzten drei Listen die Daten der Achsen des Beschleunigungssensors. Da für die Werte der x-Achse die Zeitstempel verwendet werden, enthalten diese Listen jeweils Tupel der Form (**Zeitstempel, gemessener Wert**).

### 3.4.4 Darstellen von Daten

Das Erzeugen der Graphen erfolgt in der Klasse **Plotter**, welche die Bibliotheken **numpy** und **pyqtgraph** verwendet. Für den Filter wird zudem noch die Bibliothek **scipy** benötigt. Durch **pyqtgraph** wird ein **PlotWidget** erzeugt, welches genau einen Graphen beinhaltet, der entweder die Werte der drei Achsen des Gyroskops oder des Beschleunigungssensors darstellt. Die y-Achse ist in der Einheit  $\circ(Grad)/s$  bei ersterem Graph und  $m/s^2$  bei letzterem. Die x-Achse ist in beiden Graphen der Zeitstempel des Datentupels. In Abbildung 13 ist ein solcher Graph beispielhaft dargestellt.



**Abbildung 13:** Beispielgraph

Ein **PlotWidget** enthält ein **PlotItem**, welches eine Menge von **PlotDataItems** enthält, d.h. eine Menge von gemessenen Datenpunkten. Für jeden der beiden Graphen wird in der Klasse **Presenter** eine Instanz der Klasse **Plotter** erzeugt und zu einer **QVBoxLayout** hinzugefügt. Diese dient dazu, die einzelnen enthaltenen Widgets relativ zueinander anzutragen. Damit die Graphen unabhängig von der Anzahl der enthaltenen Datenpunkte gleich groß dargestellt

werden, wird die Größe des Fensters für jeden Graphen neu berechnet. Übersteigt die Größe des Graphen die Größe des Fensters, wird automatisch die horizontale Scrollbar angepasst.

### 3.4.5 Datenaufbereitung

Durch verschiedene äußere Bedingungen (z.B. Empfindlichkeit des Sensors oder Vibration der Bahn) werden die gemessenen Rohdaten verfälscht. Durch Anwendung von Filtern können die Daten nachträglich überarbeitet werden. Hierbei wurden zwei Ansätze betrachtet, aber nur einer implementiert:

1. Savitzky-Golay-Filter
2. Kalman-Filter

**Savitzky-Golay-Filter** Das Savitzky-Golay-Filter ist ein einfaches Glättungsfilter, bei dem auf die Daten der Kurve innerhalb eines Fensters eine polynomiale Regression angewandt wird. Das Filter hat gute Eigenschaften, wenn es um die relative Verteilung von Maxima und Minima geht. Es ist wahrscheinlich, dass durch die Empfindlichkeit der Sensoren bereits kleinste Vibrationen der Bahn aufgezeichnet werden. Diese geben jedoch kaum Aufschluss, in welchem Abschnitt der Sortieranlage sich das Schüttgut derzeit befindet. Größere Bewegungen, die auch in größeren Ausschlägen in den Sensoren resultieren, sind dafür besser geeignet. Insofern ist der Erhalt der Maxima und Minima von Nutzen. Die Parameter, die bei diesem Filter verändert werden können, sind die Fensterbreite und der Grad des Polynoms. Je größer das Fenster und desto niedriger der Grad, desto mehr wird die Kurve effektiv gedämpft.

**Kalman-Filter** Das Kalman-Filter zieht im Gegensatz zum Savitzky-Golay-Filter zwei verschiedene Arten von Rauschen in Betracht:

- Prozessrauschen
- Sensorrauschen

Die Wahl und Beschreibung des zugrunde liegenden Systems ist jedoch entscheidend für die Parameterwahl. Ist das System nur unzureichend bekannt und wird deshalb schlecht beschrieben, ist der Filter nicht mehr als ein Tiefpass und wird keine guten Ergebnisse liefern. Aufgrund fehlender Erfahrungswerte in der Anwendung dieses Filters, wurde dieser Ansatz verworfen.

Die Wahl der richtigen Parameter ist beim Einsatz von Filtern entscheidend. Diese haben großen Einfluss auf die Form des resultierenden Graphen. Das macht sie aber auch besonders problematisch, da sie meist nur experimentell ermittelt werden können und es ist nicht immer ersichtlich, wie weit die Graphen tatsächlich geglättet werden müssen, um ein verwertbares Ergebnis zu erhalten.

## 4 Datenmessung

### 4.1 Erkenntnisse aus dem ersten Testlauf

Nach dem ersten Testlauf mit der Android-App und dem instrumentierten Schüttgut konnten wichtige Erkenntnisse gewonnen werden. Zwischen der App und dem Mikrocontroller kam es zu häufigen Verbindungsabbrüchen. In der App gibt es noch einige Unstimmigkeiten, die behoben werden sollten, damit verwertbare Daten aufgezeichnet werden können:

- Es muss einen fortlaufenden Zähler vom Mikrocontroller geben, damit nach einem Verbindungsabbruch und -wiederaufbau der Zähler mitten im Messlauf weiterläuft und nicht durch eine Zurücksetzung fälschlicherweise eine neue Testrunde markiert
- Beheben eines Bugs beim Datenaufzeichnen nach dem ersten Verbindungsabbruch
- Der Zugriff auf die UI-Elemente ist zu hoch. Die Ausgabe auf die UI ist langsamer, als die eingehenden Daten ankommen, weshalb die App bei zu vielen Ausgaben nicht mehr reagiert. Dabei konnten bisher empfangene Daten auch nicht mehr gespeichert werden
- Das Speichern sollte performanter sein
- Wie lässt sich ein Rundendurchlauf bzw einzelne markante Punkte in der Anlage während der Datenaufzeichnung markieren, um nachträglich die analysierten Daten evaluieren zu können?
- Bluetooth funktioniert über eine Reichweite zwischen 5-10 Metern. Bei zu vielen Verbindungsabbrüchen muss mit dem Empfangsgeräte an der Anlage mitgelaufen werden, da die Wände der Anlage zu abschirmend sind. Eventuell gibt es eine Möglichkeit mehrere BLE-Empfänger an der Anlage zu positionieren
- Zum Schutz des Bauteils wurde die Kapsel in Polsterfolie eingepackt. Dadurch wurde das Bauteil langsamer durch die Anlage befördert als die Steine
- Ein Akku mit 100 mAh hält länger als anfangs angenommen

### 4.2 Erster Messlauf

Nach Optimierung der Android-App konnten wieder Daten gemessen werden. Der Bluetooth-Verbindungsauflauf wurde überarbeitet, sodass die Verbindung nun viel stabiler läuft. Pro Rundendurchgang wurde ein interner Zähler der App zurückgesetzt, um Runden im Datensatz identifizieren zu können. Zusätzlich wurde an immer gleichen Punkten der Anlage die empfangen Daten zwischengespeichert: Übergänge von Rüttler auf Förderband und andersrum, Ausnahmen: Langes Förderband auf Querrüttler oben und ganz kurzer Rüttler bis Sortierband. Da damit eine neue Datei angelegt wurde, können auch einzelne Anlagenabschnitte innerhalb eines Messlaufs zugeordnet werden.

Die Kapsel wurde bei den ersten Messrunden in Polsterfolie eingepackt (Abbildung 14). Da dadurch die Kapsel auf der Anlage langsamer als die Steine befördert wurde, wurden an die Verpackung noch drei Flügel, wie in Abbildung 15 zu sehen, angebracht. Dadurch sollen Steine die Kapsel besser mitnehmen können. Damit war die Kapsel ebenso schnell wie die Steine und lag auch auf den einzelnen Beförderungsmodulen sehr stabil, d.h. wenige Orientierungsänderungen. Zum Abschluss wurde die Kapsel ohne Verpackungsmaterial durch die Anlage geschickt (Abbildung 16). Die Kapsel war hier ebenso schnell wie die Steine. Allerdings war hier zu beobachten, dass sich die Kapsel stark um die Längsachse drehte.

In den letzten Dateien kam es zu Testfehlern, in der keine aktuellen Daten vom Sensormodul mehr gesendet wurden.



**Abbildung 14:** Variante 1: Kapsel in Polsterfolie eingepackt



**Abbildung 15:** Variante 2: Die Polsterverpackung der Folie wurde mit drei Flügeln aus Duct Tape erweitert



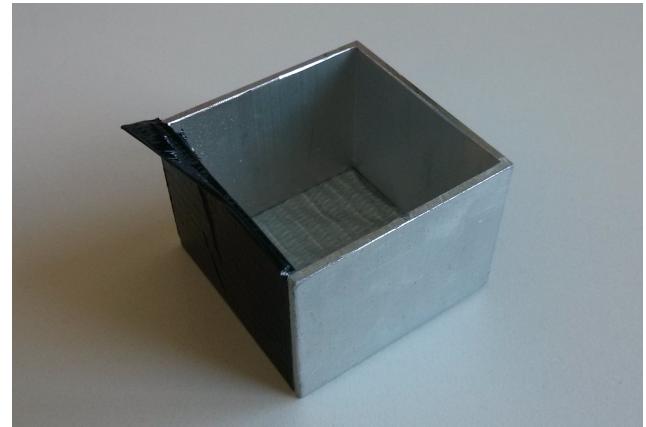
**Abbildung 16:** Variante 3: Die unverpackte Kapsel in der Anlage

### 4.3 Zweiter Messlauf

In einem weiteren Messlauf wurde noch einmal die Auswirkungen der Form des instrumentierten Schüttguts getestet. Dafür wurde wie in Abbildung 17 und 18 die Kapsel in einen quadratischen Alu-Rahmen mit Duct Tape fixiert. Damit kann das Verhalten eines Schüttguts mit rechteckiger Form getestet werden.



**Abbildung 17:** Variante 4: Die Kapsel ist in einen Alurahmen geklebt, um eine Kastenform zu erhalten



**Abbildung 18:** Variante 4: Die Kapsel wird mittels Duct Tape im Alurahmen fixiert

Nach fünf Testrunden und der Kontrolle der gespeicherten Datendateien musste leider festgestellt werden, dass vom Mikrocontroller fehlerhafte Sensorwerte geliefert wurden. Beim Ausweichen auf ein anderes instrumentiertes Schüttgut musste zunächst ein Kabel verlötet werden, das abgebrochen war. Dies zeigt, dass die Mikrocontroller nicht sehr robust sind, allerdings auf der Anlage auch starken Vibrationen und Erschütterungen ausgesetzt sind.

Nach einem Verbindungsauflauf mit der Android-App war jedoch ein fehlerfreies Empfangen der Daten mit dem weiteren Mikrocontroller nicht möglich. Es kamen geringer getaktete Datentupels an, begleitet von vielen Bluetooth-Verbindungsabbrüchen. Eventuell beeinträchtigt der Alurahmen die Bluetooth-Verbindung, weswegen zukünftig auch noch weitere Materialien testet werden sollten. Widersprüchlich ist nur, dass der erste Mikrocontroller trotz des Rahmens senden konnte. Daraus ergibt sich auch die Notwendigkeit, vor Messbeginn Referenzmessungen mit jedem verwendeten instrumentierten Schüttgut durchzuführen, um festzustellen, ob es unterschiedliche Verhalten bei den einzelnen Adafruit-Bauteilen gibt.

Obwohl keine gemessenen Sensordaten zur Verfügung stehen, konnte beobachtet werden, dass der Alurahmen auf den Förderbändern sehr ruhig liegend befördert wurde. Auf den Rüttlermodulen jedoch drehte sich die Box auch und sprang teilweise sogar einige Zentimeter zurück. Auch das Oberflächenmaterial könnte eine wichtige Rolle spielen. Sowohl Variante 3 mit der unverpackten Kapsel und Variante 4 mit Duct Tape und Alurahmen haben eine sehr glatte Oberfläche, bei denen auch ein unruhiges Laufverhalten beobachtet werden konnte.

## 5 Auswertung

### 5.1 Datenanalyse

Die gemessenen Daten des Beschleunigungssensors werden in  $m/s^2$  dargestellt, die Daten des Gyroskopes in  $\circ/s$  ( $\circ$ : Grad).

Die Analyse wird durch zwei Faktoren erschwert:

1. Die Sensoren sind sehr empfindlich.
2. Baulich bedingt dreht sich das Ei sehr leicht um die Längsachse, wodurch eine kurze Beschleunigung auf eine konstante Geschwindigkeit nicht als ein einzelner Ausschlag, sondern als eine Kombination verschiedener, dargestellt wird und so kaum erkennbar ist.

Da bisher noch kein Klassifikator existiert, wird die Auswertung von Hand durchgeführt. Um dies zu erleichtern, wurde ein Savitzky-Golay-Filter mit den Parameterpaaren (25, 5) und (75, 7) eingesetzt. Die Werte beschreiben in dieser Form (Fenstergröße, Grad des Polynoms) die Eingangsparameter für den Filter. Denn bei direktem visuellen Vergleich zweier Testläufe über den kompletten Zeitverlauf, werden die Graphen zusammengestaucht und Bewegungstendenzen sind dadurch schwerer zu erkennen und zu bewerten. Der Einsatz eines Filters dient an dieser Stelle der Erleichterung, die Einteilung des Graphen in verschiedene Segmente vorzunehmen.

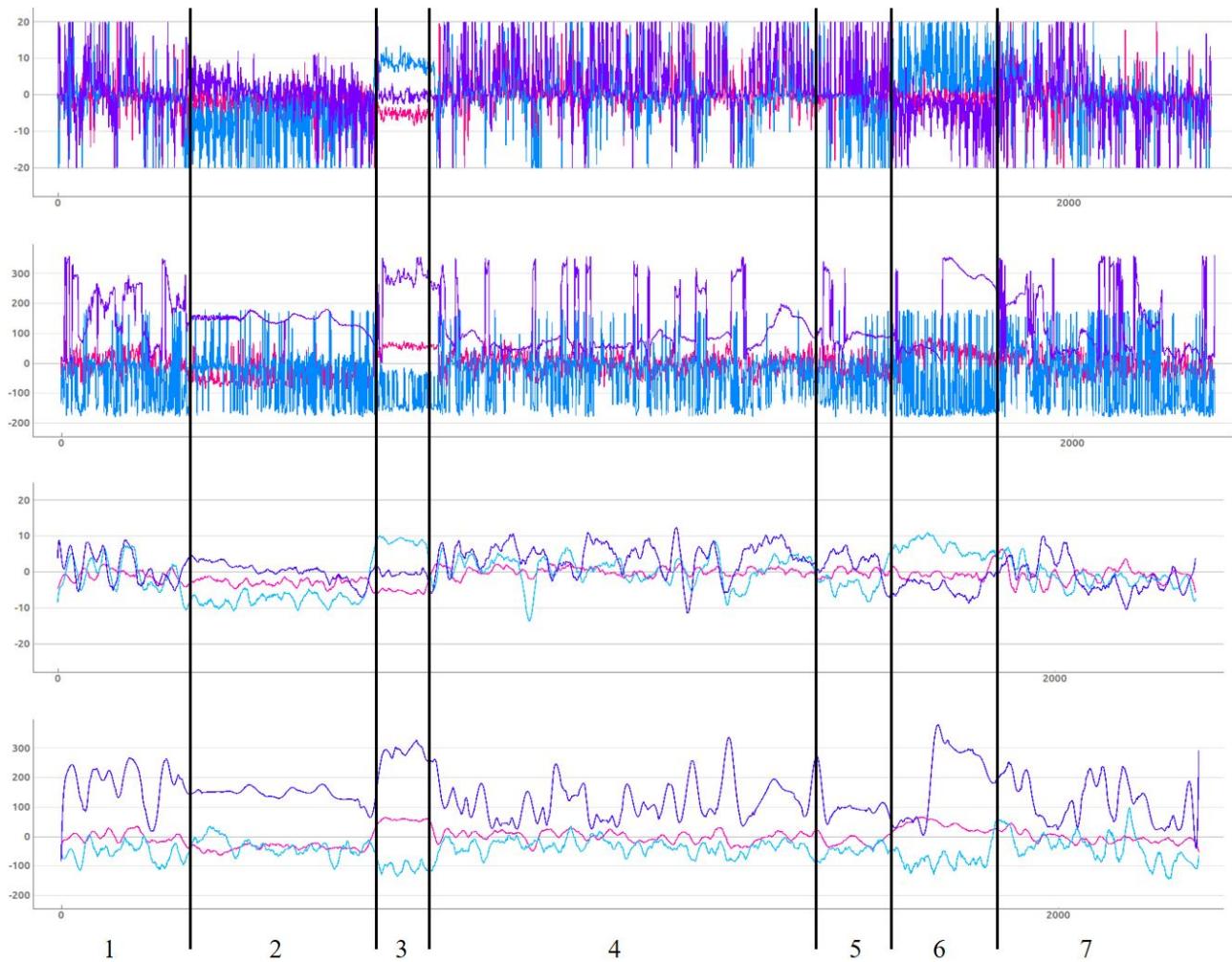
Ob die Verwendung eines Filters auch geeignet beziehungsweise notwendig ist, um Messrauschen aus den Daten zu entfernen, muss noch geprüft werden.

In Abbildung 19 wurde der Datensatz Nr. 1 mit einer Länge von 3,3 Minuten untersucht<sup>2</sup>. Die oberen beiden Graphen zeigen die ungefilterten Beschleunigungswerte und Winkelgeschwindigkeiten der Rohdaten, wogegen die unteren beiden Graphen den gleichen Datensatz mit angewandtem Savitzky-Golay-Filter (75, 7) zeigen. Auf der x-Achse sind die Datentupel aufgetragen, wobei alle 86 Millisekunden ein neuer Datensatz empfangen wurde.

Die Segmentierungen wurden an den Stellen gesetzt, an denen sich stärkere Änderungen bei den Bewegungsmustern erkennen lassen. Die Anzahl der Anlagenmodule ist bekannt. Da ein Messlauf mitten in einem Modul startet und endet, da die Anlage einen geschlossenen Schüttgut-Kreislauf hat, sind sieben Segmente notwendig. Das erste und letzte Segment gehören dabei zum selben Modul. Diese lassen sich wie folgt zu den einzelnen Anlagenmodulen zuordnen und beschreiben:

**1 Großer Rüttler unten:** Der Rüttler hat eine hohe Rüttelstärke, weshalb die Schüttgüter auf diesem stark springen. Dies lässt sich anhand den schwankenden Graphen sowohl bei der Beschleunigung als auch bei der Winkelgeschwindigkeit erkennen. Da ein Rundenlauf der Kapsel in diesem Modul startet und endet, ist das Bewegungsmuster sehr ähnlich zu dem in Segment 7.

<sup>2</sup>Anmerkung: Der Client visualisiert die Daten in einer anderen Farbkombination. Die Abbildungen 19 und 20 wurden nachträglich für eine besser erkennbare Darstellung im Dokument eingefärbt.



**Abbildung 19:** Datenanalyse: Nummerierte Segmentierung des ersten Datensatzes; Dargestellte Graphen: 1) Beschleunigungsdaten im Rohformat, 2) Winkelgeschwindigkeiten im Rohformat, 3) Beschleunigungsdaten mit angewendetem Savitzky-Golay-Filter (7,5,7), 4) Winkelgeschwindigkeiten mit angewendetem Savitzky-Golay-Filter (7,5,7)

**2 Querförderband unten:** Auf dem Förderband liegt das Schüttgut ruhig. Die Graphen haben geringere Ausschläge.

**3 Förderband nach oben:** Das Förderband bewegt das Schüttgut schnell nach oben, weshalb dieses Segment sehr kurz ist. Zudem ist dieses Förderband sehr steil und besitzt daher Mitnehmstützen, sodass das Schüttgut nicht nach unten rollen kann, sondern ruhig liegen bleibt. Hier sind die geringsten Ausschläge zu erkennen.

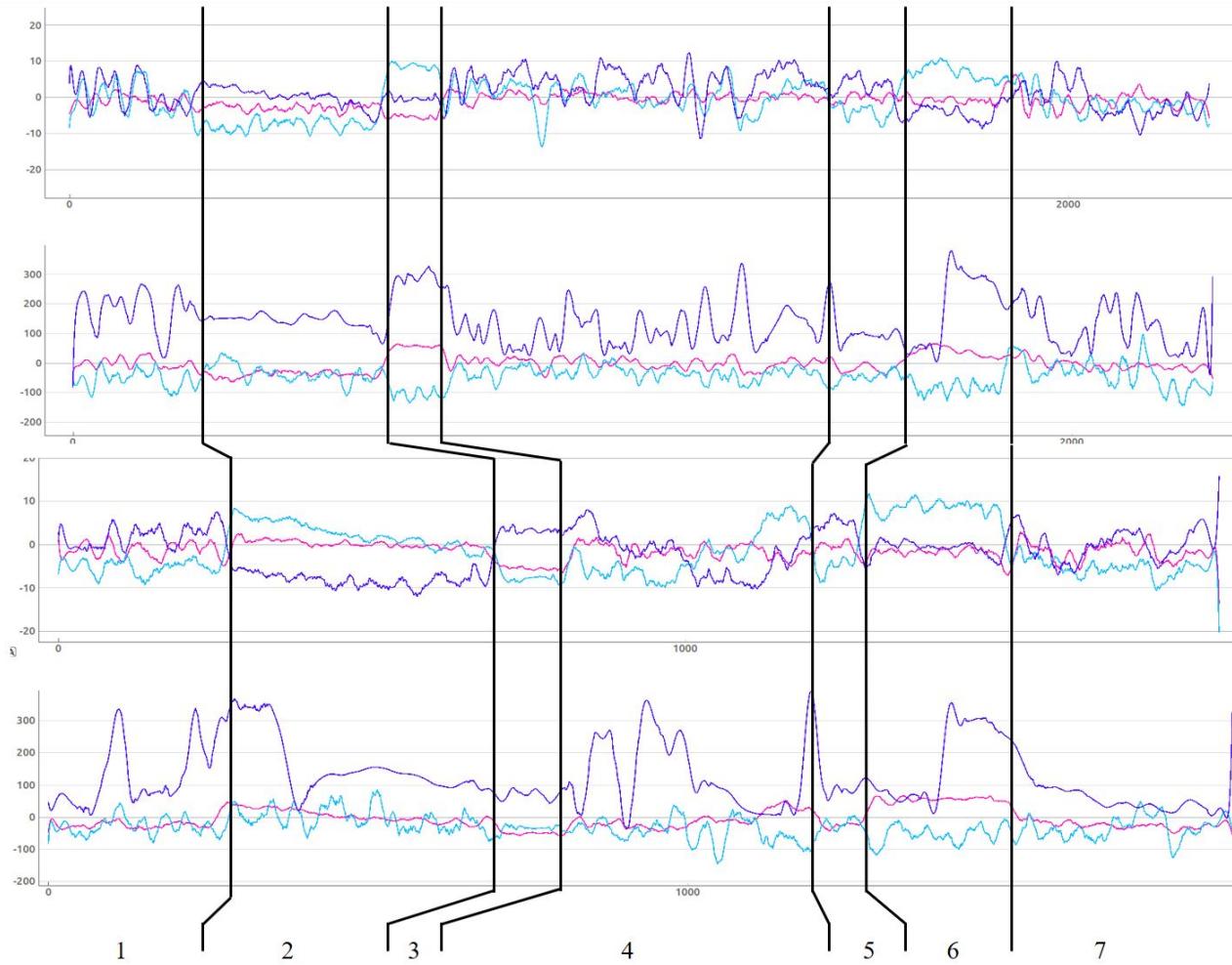
**4 Mittlerer Rüttler, quer oben:** Auf diesem Rüttler sind die Ausschläge für Beschleunigung und Winkelgeschwindigkeit wieder wesentlich stärker

**5 Kurzer Rüttler:** Der kurze Rüttler gibt das Schüttgut für das Sortierband frei und hat eine geringere Rüttelstärke als andere, wodurch sich das Schüttgut ruhig vorwärts schiebt, statt zu springen.

**6 Rutsche und Sortierband** Über die Rutsche fällt das Schüttgut auf das Sortierband, welches das Schüttgut mit  $3 \text{ m/s}$  befördert. Die Beschleunigungsdaten schwanken hier weniger stark, wobei es einen Peak in einer Achse der Winkelgeschwindigkeiten gibt und es zu zwei Vorzeichenänderungen in den Beschleunigungswerten kommt.

**7 Großer Rüttler unten:** Siehe Erklärung für Segment 1.

Nachdem Segmente gefunden wurden, in denen es unterschiedliche Bewegungsmuster gibt, müssen diese mit den weiteren Datensätzen von anderen Messläufen evaluiert werden. Abbildung 20 zeigt einen Vergleich der Segmente zwischen Datensatz 1 (Dauer: 3,3 Minuten) und Datensatz 17 (Dauer: 2,4 Minuten).



**Abbildung 20:** Datenanalyse: Vergleich der gefundenen Segmente von zwei verschiedenen Datensätzen (1 + 17), jeweils mit angewandtem Savitzky-Golay-Filter (75,7); Dargestellte Graphs: 1) Datensatz 1, Beschleunigungsdaten, 2) Datensatz 1, Winkelgeschwindigkeiten, 3) Datensatz 17, Beschleunigungsdaten, 4) Datensatz 17, Winkelgeschwindigkeiten

Anhand der oben beschriebenen Verhaltensmustern lassen sich die Segmente auch in Datensatz 17 in der gleichen Reihenfolge erkennen. Vor allem deutlich sichtbar werden die unterschiedlichen Muster zwischen Rüttler (stark schwankend; 1, 4, 5, 7) und Förderbändern (ruhiger; 2,

3, 6). Besonders markant ist Segment 3, das auch in den Rohdaten durch geringe Schwankungen in den Messwerten deutlich hervor- und auch in ähnlicher Abschnittslänge auftritt. Bei den Rüttlermodulen kann es passieren, dass die Kapsel nicht gut vom Schüttgut mitgenommen wird und mehr Zeit benötigt. Der Übergang zu Segment 5 wird mit einem höheren Ausschlag im Winkelgeschwindigkeits-Graphen begonnen. Bei diesem Anlagenabschnitt fällt die Kapsel etwa einen halben Meter auf den nächsten Rüttler. In Segment 6 lässt sich immer ein stärkerer Peak bei der Winkelgeschwindigkeit finden, der dann langsam abfällt. Der Übergang von Segment 5 auf Segment 6 kann allerdings nicht exakt ausgewertet werden. Es gibt zwar an diesen Stellen Änderungen in der Bewegungstendenz, allerdings ist die Zeitdauer von Segment 6 für Rutsche und Sortierband unerwarteterweise unterschiedlich lang. An dieser Stelle wurden die Daten nie in eine Datei zwischengespeichert.

Die händisch vorgenommene Segmentierung wurde mittels den erstellten Datendateien beim Zwischenspeichern der Android-App an den markanten Punkten kontrolliert.

## 5.2 Evaluation der Methode

Es hat sich gezeigt, dass einzelne Segmente in den Daten erkennbar sind, die sich mit den Modulen der Anlage korrelieren lassen. In Kombination mit dem beobachteten Verhalten des instrumentierten Schüttguts auf der Bahn können diese Erkenntnisse auch auf das zu sortierende Schüttgut übertragen werden.

Mit Hilfe des instrumentierten Schüttguts lassen sich beispielsweise Verweilzeiten auf den einzelnen Anlagenmodulen bestimmen. Durch Anpassung der Anlagenparametern wie Geschwindigkeiten von Förderbändern oder Rüttlerstärke können die Auswirkungen auf die Schüttgüter durch erneut gemessene Daten überprüft werden.

Zusätzlich können mit Hilfe von instrumentierten Schüttgütern Aussagen über das Bewegungsverhalten des Schüttguts selbst gemacht werden, sofern die Abweichungen in Größe, Oberflächenbeschaffenheit und Gewicht nicht zu groß sind. Damit könnten vielleicht Bewegungsmodelle für Schüttgüter gewonnen werden.

## 5.3 Fazit

Es konnte ein instrumentiertes Schüttgut entwickelt werden, mit dem Beschleunigungs- und Orientierungsdaten gemessen werden können. Die gewählten Bauteile von Adafruit passen im gelöteten Zustand in eine Ü-Ei-Kapsel. Die Bluetoothverbindung wird durch die Kapsel selbst nicht beeinträchtigt. Jedoch befindet sich an der Anlage ein großer Anteil Metall, was sich unter Umständen auch negativ auf die Bluetoothverbindung auswirkt. Durch Nutzung einer Android-App kann jedoch auf Höhe des Sensors an der Anlage mitgelaufen werden. Die Hardware des Mikrocontrollers weist noch Schwachstellen auf. Darunter fallen insbesondere die empfindlichen Lötverbindungen an den einzelnen Platinen, die z.B. durch Vibrationen in der Kapsel brechen können. Ebenso sind bei Nutzung mehrerer Mikrocontroller Referenzmessungen notwendig, um

unterschiedliche Verhaltensarten der Mikrocontroller frühzeitig zu identifizieren. Eine besondere Rolle spielt hier vor allem die Taktung des Adafruit-Mikrocontrollers, durch den die Anzahl der gemessenen Datentupel in einem Zeitraum bestimmt werden.

Die Verpackung der Kapsel wirkt sich bei einem Messlauf auf das Bewegungsverhalten aus. Mit der Polsterfolie gewinnt das instrumentierte Schüttgut an Größe. Die Erweiterung mit Flügeln wirkt sich positiv auf die Geschwindigkeit aus, sie dienen als Mitnahmemöglichkeit durch das umliegende Schüttgut. Gegenüber dazu stellen sich die Messreihen mit Kapsel ohne Verpackung und im Alurahmen. Neben der geänderten Form durch die Aluverpackung sind die Oberflächen sehr viel glatter und unähnlicher zu den Steinen. Abgesehen vom Verpackungstyp ist die Kapsel mit ihrer Originalgröße größer bzw. im Verhältnis zu Steinen der gleichen Größe wesentlich leichter. Dadurch erfährt die Kapsel in der Steinmenge einen höheren Auftrieb und spiegelt nicht exakt das Bewegungsmuster des Schüttguts, in diesem Fall der Steine, wieder.

Der Client kann die gewonnenen Daten aufbereiten und visuell darstellen. Implementiert wurde ein Savitzky-Golay-Filter, der bei der Datenanalyse genutzt wurde. Die Filterparameter sind hier experimentell ermittelt, wodurch die Eignung anhand weiterer Daten verifiziert werden muss. Allerdings sollte bei erweiterten Auswertungen evaluiert werden, in wie weit die Nutzung des Filters sinnvoll ist.

Die gewonnenen Daten sind jedoch mit Vorsicht zu betrachten, da die Messläufe unterschiedlich lang sind und Verbindungsabbrüche auftreten können.

Aus der Datenanalyse ging hervor, dass sich bestimmte Bewegungsmuster in immer gleich auftretender Reihenfolge in den Messläufen erkennen lassen. Anhand bestimmten charakteristischen Mustern in Segmenten lässt sich eine Zuordnung zu den Anlagenmodulen vornehmen.

## 5.4 Ausblick

Wie sich gezeigt hat, sind noch einige Erweiterungen und Optimierungen möglich. Im Bereich des Mikrocontrollers könnte statt einer Bluetooth-Verbindung eine Speicherplatte hinzugefügt werden, aus der die Daten am Laptop ausgelesen werden. Dies hätte den Vorteil, dass keine Daten durch Verbindungsabbrüche verloren gehen können. Allerdings würde dadurch auch die Kontrolle über den Mikrocontroller verringert, da keine Befehle, wie z.B. das Zurücksetzen von Zählen zum Markieren einer neuen Messrunde, gesendet werden könnten. In diesem Fall müsste man entweder nach jedem Lauf die Daten aus dem Mikrocontroller auslesen oder eine Segmentierungsmethode finden, um eine große Datei, die mehrere Messläufe enthält, in einzelne Dateien zu separieren, die dann jeweils nur noch einen Messlauf beinhalten.

Unter Umständen ließe sich das Probleme der auftretenden Verbindungsabbrüchen lösen, indem ein anderes Funkmodul wie beispielsweise WLAN verwendet wird. Allerdings ist dabei nicht bekannt, wie sich die Metallwände des FlexSorts auf diese Verbindungsart auswirkt.

Um eine größere Verlässlichkeit der Messergebnisse zu erreichen und das Verhalten der Schüttgüter untereinander besser zu beobachten, könnten parallele Messungen mit mehreren instrumentierten Schüttgütern vorgenommen werden. Dazu müsste entweder die App noch erweitert

werden, sodass Verbindungen zu mehreren Mikrocontrollern möglich sind oder die Mikrocontroller mit internem Speicher ausgestattet werden, sodass keine Verbindung zum Speichern der Daten mehr notwendig ist.

Auch bei der App sind noch Optimierungsmöglichkeiten beziehungsweise mögliche Erweiterungen denkbar. Um die Auswertung am PC komfortabler zu gestalten, könnte eine direkte Datenübertragung an den PC implementiert werden, sodass die Daten nicht zuerst manuell vom Smartphone, eventuell noch über einen Cloudspeicher, auf den Laptop übertragen werden müssen. Ein nützliches Feature wäre ein Live-Graph, der die empfangenen Daten sofort in der App visualisiert.

Eventuell ist es möglich, die empfangenen Daten am Smartphone direkt über einen Data-Service an einen PC zu streamen, an dem weitere Analysen gemacht werden. In Verbindung mit einer Visualisierung gäbe es dann die Möglichkeit, direkt in Echtzeit Aussagen zum Aufenthalt der Kapsel in der Anlage machen zu können.

Beim Client wäre die wichtigste Erweiterung der bereits erwähnte Klassifikator. Dabei bleibt noch zu evaluieren, ob ein supervised oder unsupervised Ansatz gewählt werden soll und welche Art von Klassifikator an sich geeignet ist. Sollte ein supervised Klassifikator gewählt werden, wäre für eine bessere Segmentierung eines Messlaufs mit Zuordnung auf die einzelnen Anlagenmodulen der Einsatz von RFID-Schranken an den entsprechenden Übergängen der Module nützlich. So wäre die Segmentierung deutlich präziser. Unabhängig davon, welcher Ansatz nun letztendlich verfolgt wird, müssen wesentlich mehr Daten gemessen werden. Eine andere mögliche Erweiterung des Clients, ist eine Darstellung der Graphen oder auch eine Simulation in 3D, wie sich der Mikrocontroller im Raum bewegt.