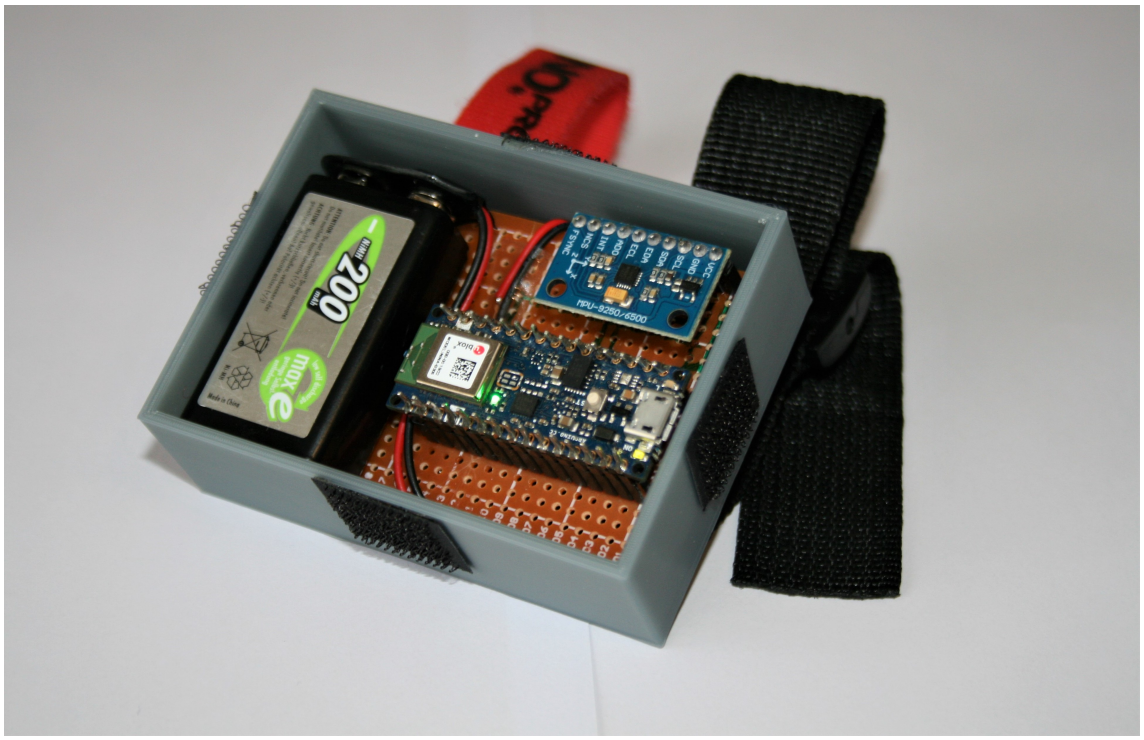

Archer-Tracking

Bewegungstracking für Sportler

Antonio Rehwinkel

29. Dezember 2021

Unterstützende Lehrer: Herr Czernohous
Herr Dierle
`Emails?`

und Professoren: Hochschule1
Hochschule2
`Emails?`

Inhaltsverzeichnis

1 Vorwort

Mithilfe eines Beschleunigungssensoren kann man viele Bewegungen erforschen und vermessen. Mit meinem System sollen mehrere Beschleunigungssensoren dazu eingesetzt werden können, Bewegungsabläufe aufzunehmen, miteinander zu vergleichen und zu erkennen.

Die Daten werden über Bluetooth-Low-Energy an ein Handy geschickt, wo Sie sowohl gespeichert als auch verwertet werden können. Als Beispiel gilt hier für mich das Bogenschießen, bei dem selbst kleine Bewegungen immer wieder auf gleiche Weise ausgeführt werden müssen. Mit meinen Sensor sollen hier teure Kamerasysteme abgeschafft werden und es so jeden ermöglichen, selbst ohne Bogen oder Trainer bei sich Zuhause zu den Bewegungsablauf zu trainieren.

2 IMU vs Kameratracking

Mein Projekt überschneidet sich in seinen Zielen häufig mit Tracking das bei VR-Brillen eingesetzt wird. Hier wird zur Feststellung der Position des Spielers häufig eine Kombination aus Kameratracking und Infrarot-LED.
Hierbei muss der Spieler die Fernbedienungen festhalten die die Infrarot-LEDs beinhalten, während die Kameras im Raum so verteilt werden müssen das der Spieler immer erkannt wird.
Die Neigung des Kopfes und der Hände werden auch hier häufig Mithilfe eines IMU bestimmt.

Da diese Systeme viel Platz benötigen, viel Geld kosten und für die Bildverarbeitung häufig eine große Rechenkraft benötigen ist dieses System nicht für viele Privatnutzer sinnvoll oder bieten einen Bewegungsfreiraum der Sport zu lässt.

Die IMU-Sensoren bestehen mindestens aus einem Gyroskop und einem Beschleunigungssensor, manche bieten sogar ein Magnetometer an. Somit sollte es möglich sein, über die Beschleunigung die Distantz die ein Körper mit diesem Sensor zurück legt zu messen. Die Neigung und Orientation sind über das Gyroskop und Magnetometer sehr genau messbar.

Die Vorteile der IMU liegen auf der Hand, sie sind günstig, klein und leicht. Aus diesen Gründen trägt fast jeder heutzutage so einen Sensor bei sich, die meisten Handys haben ihn schon eingebaut.

Für mein Projekt benutze ich dennoch einen eigenen IMU, um die Qualität der Messdaten sicher zu stellen.

3 Bluetooth-Low-Energy

Mit Bluetooth 5.0 wurde eine neue übertragunsweise zu Bluetooth hinzugefügt. Diese nennt sich Bluetooth-Low-Energy und zeichnet sich durch einen geringen Stromverbrauch und damit einem höherem Datendurchsatz aus.

Bluetooth sendet Daten in Paketen. Hierbei ist bei Bluetooth-Low-Energy (zukünftig BLE) der Sender als Server ausgewiesen und der Empfänger als Client.

Hierdurch kann der Client bis zu ??? Server abfragen ohne sich mit diesen verbinden zu müssen. Jeder Server kann unbegrenzt Characteristiken anbieten, diese Stellen verschieden Datensätze dar, die vom Client abgefragt werden können.

Laut Dokumentation beträgt der Maximale Datensatz 244 Bytes pro Paket bei aktiviertem DLE. Diese Funktion ließ ich ausgeschaltet, wodurch ich Maximal 27 Bytes pro Paket versenden kann.

3.0.1 Datendurchsatz per BLE

2Mbps, steht in Prozessor Doku, lieber nochmal in BLELib nachlesen!
Das Sendeprotokoll von Bluetooth schreibt vor, das ein Datenpaket von leeren Datenpakten eingepackt wird, somit beträgt die Sendezeit pro Datenpaket
“Zeit = Leer + IFS + Data + IFS”
“Leer = leerespacket(größe) / datarate”
Für mich heißt das:
leerGröße = 2 + 4 + 2 + 3 = 11 Bytes == 88 bits
und die Sendezeit für das leere Paket beträgt damit:
leerZeit = 88/2Mpbs = 44 Mikro Sekunden.
Für ein volles DAtenpaket brauche ich:
Voll = 44 + 2*150+ 2+4+2+4+20+3 = 44+300+35 = 379Bytes *8 ==3032 Bits
Vollzeit = 3032 / 2 = 1,516 Mikrosekunden
Für ein gesaamtes Datenpaket brauche ich damit mindestens:
Zeit = 44 + 2*150 + 1,516 = 345,516 Mikrosekunden
88/2 + 2*150 + (2 + 4 + 2 + 4 + 20 + 3)*8/2

3.0.2 Datengröse

Die Daten werden als String versendet, diese werden von Arduino mit einer Null Terminiert. Die Größer der Sensordaten beträgt:
Vorkommastellen (3) + Komma (1) + Dezimalstellen (2) + Terminierung (1) = 7 Char
1 Char entspricht 1 Byte, somit gilt:
9 Sensoren * 7 Byte = 63 Byte
63 Byte / 27 Byte = 2,3 Datenpakete pro alle Sensoren Somit brauche ich für das Senden

aller Sensoren mindestens 3 Characteristics.

3.0.3 Tatsächliche Übetragunsgeschwindigkeit

4 Hardware

4.1 Arduino Nano 33 BLE

Der Arduino Nano 33 BLE wurde mit einem M4-ARM-Processor und einem 5.0 Bluetooth Modul ausgestattet. Damit ist er BLE-Fähig und liefert einen starken Prozessor für Kommarechnungen.

Der Arduino eignet sich durch seine BLE-Fähigkeit und I2C beziehungsweise SPI-Anschlüsse zur Verwendung mit dem Verwendetem MPU9250. Die geringe Größe und Stromverbrauch sind weitere Pluspunkte.

4.2 MPU9250

Der benutze IMU in diesem Projekt ist ein Multi-Chip mit einem 3-Achsen Gyroskop, 3-Achsen Beschleunigungssensor und einem 3-Achsen Magnetometer. Alle Sensoren wurden noch in der Firma kalibriert. Der Chip bietet einen eingebauten Low-Pass-Filter der eine erste Bearbeitung der Daten vornimmt und so den Hauptprozessor entlastet.
In der folgenden Tablle ist die Empfindlichkeit und Genauigkeit der einzelnen Sensoren notiert, sowie die Übetragunsart und Geschwindigkeit.

Für eine einfache Handhabung benutze ich die I2C Verbindung zwischen Arduino und MPU9250.

4.3 Stromverbrauch

Der Stromverbrauch wurde mit einem Multimeter am Batterieanschluss in verschiedenen Modi gemessen. Die Ergebnisse stehen in der Tabelle:

Der Stromverbrauch lässt so berechnen und erleichter die Korrekte Batterie-Wahl.
Die Verwendeten Formeln: P = U / I
Ah / V = Wh
Wh / W = t –> (U*I*t) / U*I = t

So verbraucht der Arduino
0,005 A * 7,4 V = 0,037 W

Die Verschiedenen Batterien-Typen stehen in der folgenden Tabelle:

Um eine Stromversorgung des Arduinos sicher zu stellen benötigt man 2 Knopfzellen des Typs CR2025.Dennoch hat die Knopfzelle CR2025 nicht nur eine bessere Laufzeit sondern ebenfalls weniger Gewicht, weniger Platz und eine bessere Differenz zwischen Cut-Off-Spannung zu angenotener Spannung.

Gegen das umrüsten auf die Knopfbatterie spricht einzig der Umweltschutz. Denn im Gegensatz zu 9-Volt-Batterien gibt es keine Akkus für Knopfzellen.

5 Software

5.1 Android-App

5.2 Arduino Firmware

5.3 Gleichmäsige und ungleichmäßige Beschleunigung

5.3.1 Gleichmäßige Formel

5.3.2 Integral

6 Machine Learning

6.1 Datenauswahl und Dimensionen

6.2 Modell und Funktion

7 Fazit