Development of a Smart LED Hula Hoop with Sensibility for Motion

Mariella Mischinger

04.07.2019

Final Presentation: Master Thesis Informatics

Supervisor: Prof. Gudrun Klinker

Advisor: Adnane Jadid

Outline

- Introduction
- Related Work
- Implementation
  - Hardware Design
  - Software Design
- Evaluation: Motion Recognition
- Potential Issues
- Future Work
- Conclusion
- Demo

Introduction

An LED Hula Hoop?!

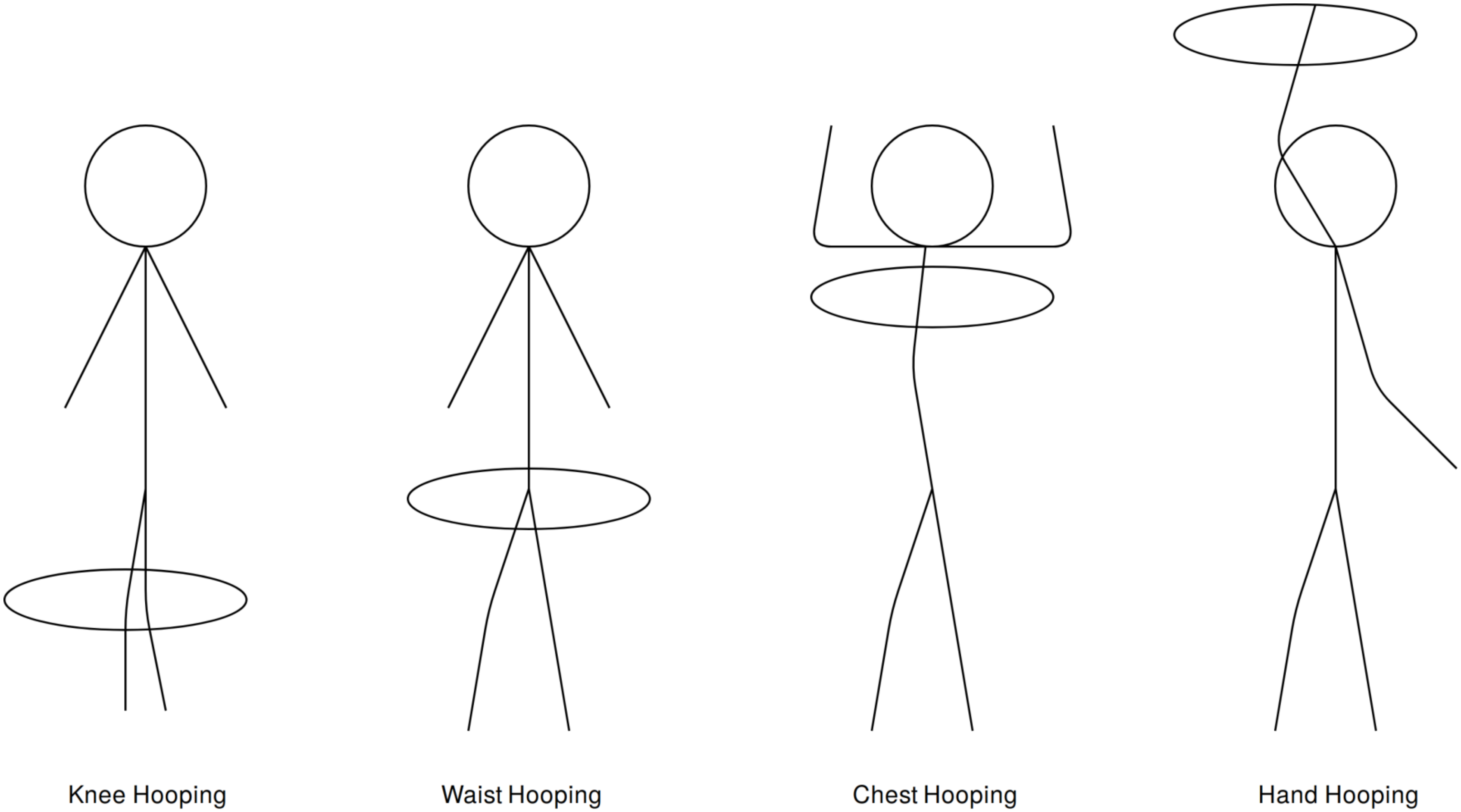
Hula Hoop is

- Creative
- Visual Art
- Dance
- Artistic

Goal of the Thesis - The Master Hoop

- Differentiates between **knee**, **waist**, **chest** and **hand** hooping
- Measures motion with an **Inertial Measurement Unit (IMU)**
- Shows **different colors** for each motion
- Fulfils all **requirements** for a professional led hula hoop

4 Different Motions



Requirements of a Professional LED Hula Hoop

- **Limited Space** of 1.6 diameter inside the tube
- **Light Weight** means less batteries
- **Long Playtime** means more batteries
- **Balanced Weight**

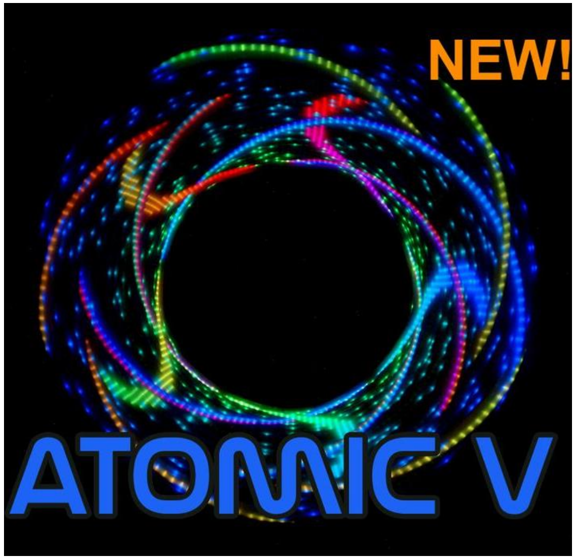
Related Work

Existing Solutions

Future Hoop by moodhoops

MATRIX Hoop by Hula the Hoop

Atomic V by Astral Hoops



Motion Detection in Existing Solutions

Future Hoop by moodhoops

✓ Integrated IMU to track orientation and adjust the lights

MATRIX Hoop by Hula the Hoop

X No integrated IMU

Atomic V by Astral Hoops

✓ Integrated IMU to stabilize images

X **None of them are able to recognize the current motion**

X **Very expensive**

✓ **The Master Hoop detects motion and is a cheap solution**

Approach

Approach

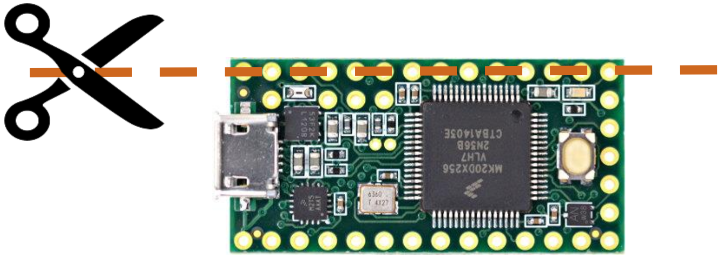
- Research
- Hardware design
- Software design
- Built up of a prototype
- Development of testing software and additional software
- Data collection for each motion
- Evaluation of the data
- Development of Motion recognition

Implementation

Hardware Design

Hardware

Limited space: Hardware has to be small enough



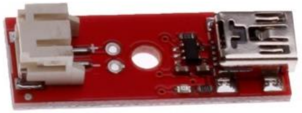
Teensy 3.2

BMI160 Sensor

5 Lithium Ion Batteries

3.7 V to 5 V Boost Converter

WS2812B LED Strip



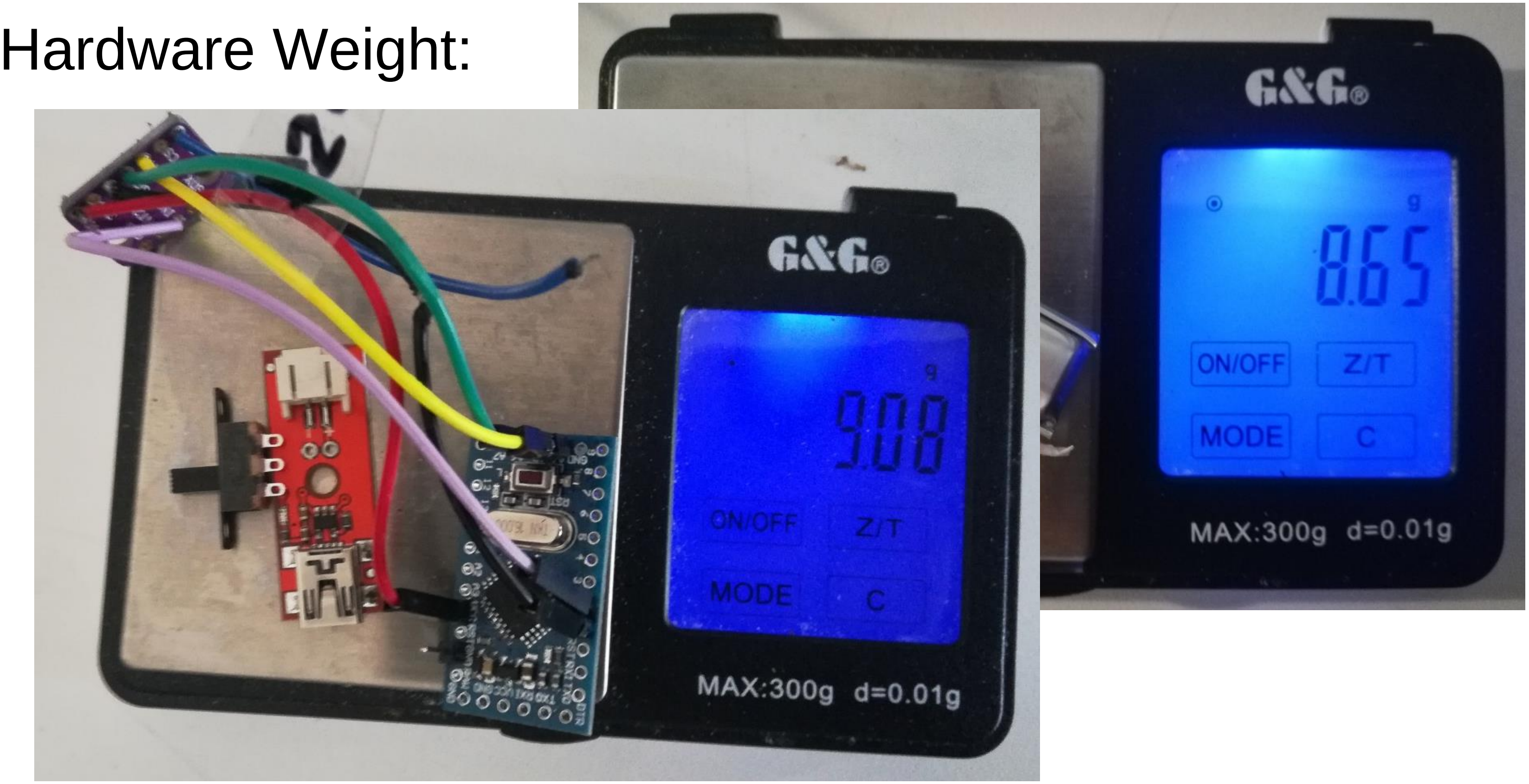
Lipo Charger

On Off Switch

Lipo Discharge Protection

Balanced Weight

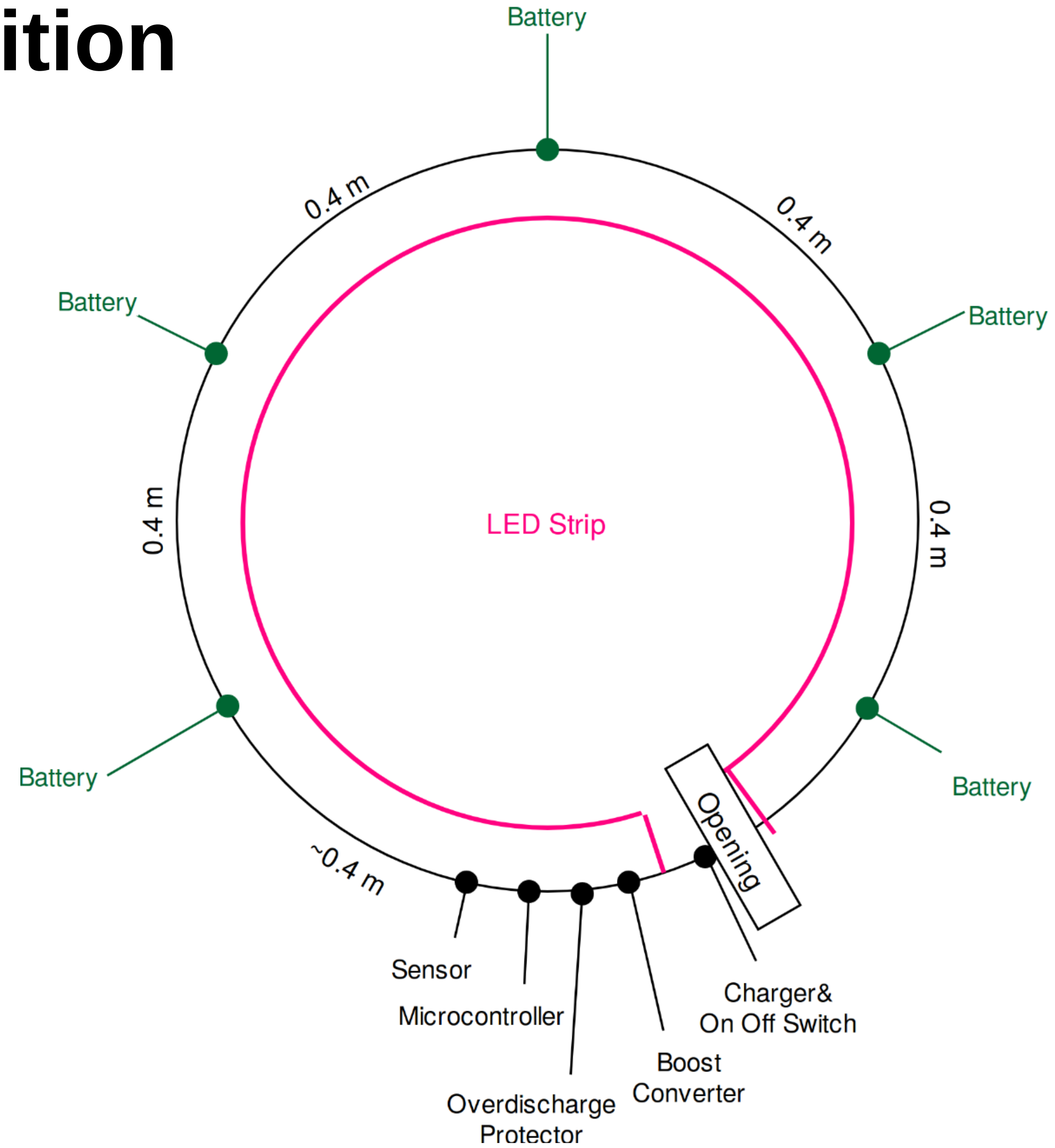
Hardware Weight:



→ 6 Pieces with almost equal weight

Hardware Position

To achieve **balanced weight,** the hardware is distributed inside the hoop



Requirements

Limited Space ✓

Balanced Weight ✓

Light Weight: ✓

5 x ~8,6 g Batteries = **43 g**

Long Playtime: ✓

5 x 500 mAh Batteries = 2500 mAh

Fatigue test: Turning on all LEDs on white color lasts **3,20 h**

Implementation

Software Design

Software Goals

- Software is microcontroller suitable
- Hardware must be compatible (Sensor, Bluetooth, LEDs, Microcontroller)
- Sensor
  - data has to be converted
  - needs to be reliable

Sensor Configuration

Set Sensor **rate** and **range**:

```
//Gyroscope Configuration
BMI160.setGyroRate(800); // 800 Hz
BMI160.setGyroRange(2000); // 2000 degrees per second
// Accelerometer Configuration
BMI160.setAccelerometerRate(800); // 800 Hz
BMI160.setAccelerometerRange(16); // 16 g (g-force)
```

Gyroscope Data Conversion

Convert measured gyroscope data to **deg/second**

```
vec3 convertRawGyro(vec3 gRaw) {
    float scaleFactor = 2000.0 / 32768.0;
    return scaleVec3WithFactor(gRaw, scaleFactor);
}
```

Accelerometer Data Conversion

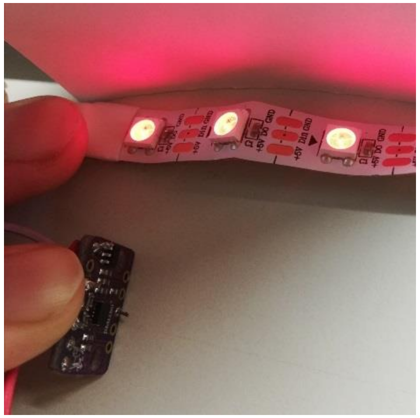
Convert measured accelerometer data to **g-force**

```
vec3 convertRawAcc(vec3 aRaw) {
        float scaleFactor = 16.0 / 32768.0;
        return scaleVec3WithFactor(aRaw, scaleFactor);
}
```

Testing the Accelerometer

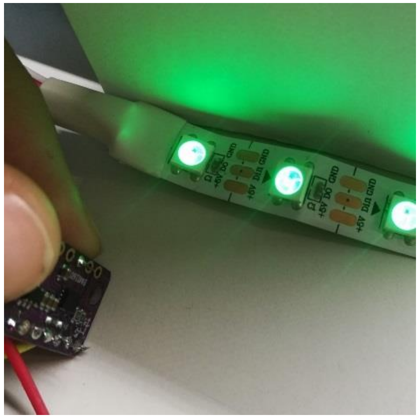
Accelerometer measures **Gravity**

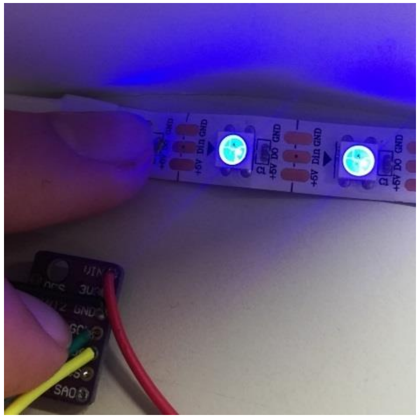
→ Sensors **x, y & z** axes mapped to LED colors **R, G & B**



X-Axis facing down

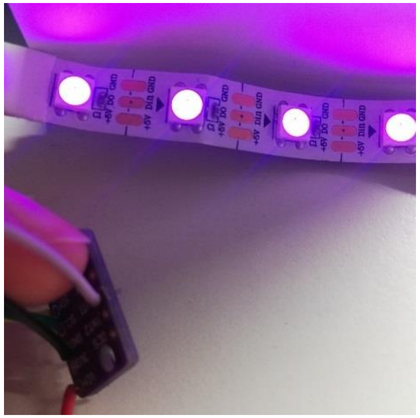
Y-Axis facing down





Z-Axis facing down

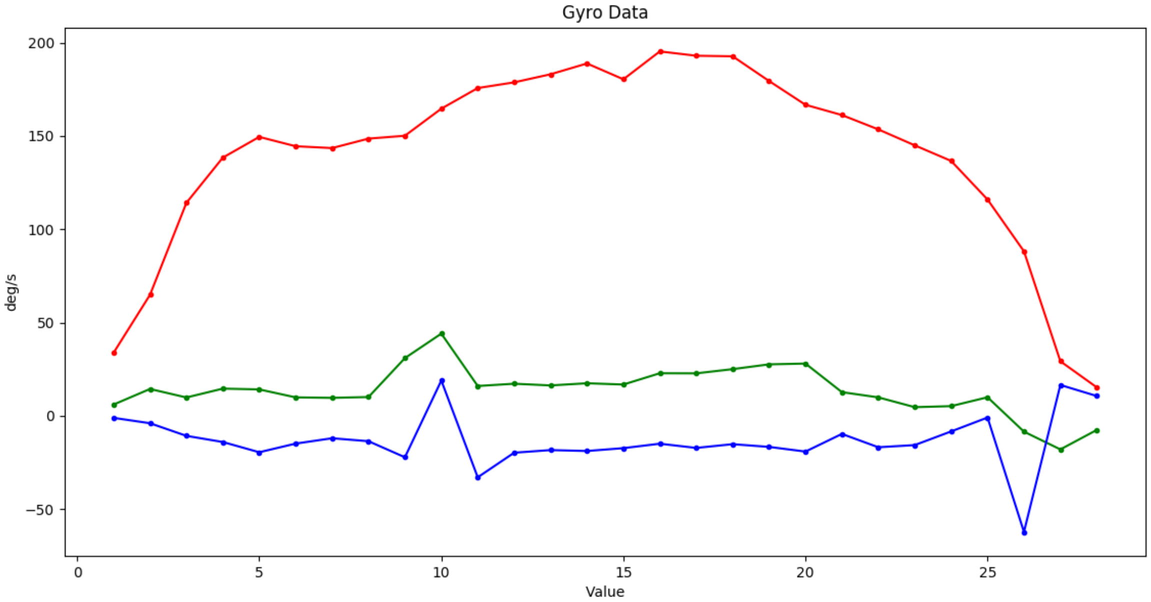
X and Z -Axis facing down



Testing the Gyroscope

- Connected a **Bluetooth module** to the hoop to **transmit the measured IMU data** to the PC
- Rolled the hoop along a wall
  - Start and end position are the same
    → **360° turn**
  - Time Duration was measured
- **Collected** and **plotted** the data on the PC

Testing the Gyroscope



$$averageAngularVelocity = \frac{\sum_{i=1}^{28}(currGyro.x_i)}{28} = \frac{3931.88}{28} = 140.4243\ degree/second$$

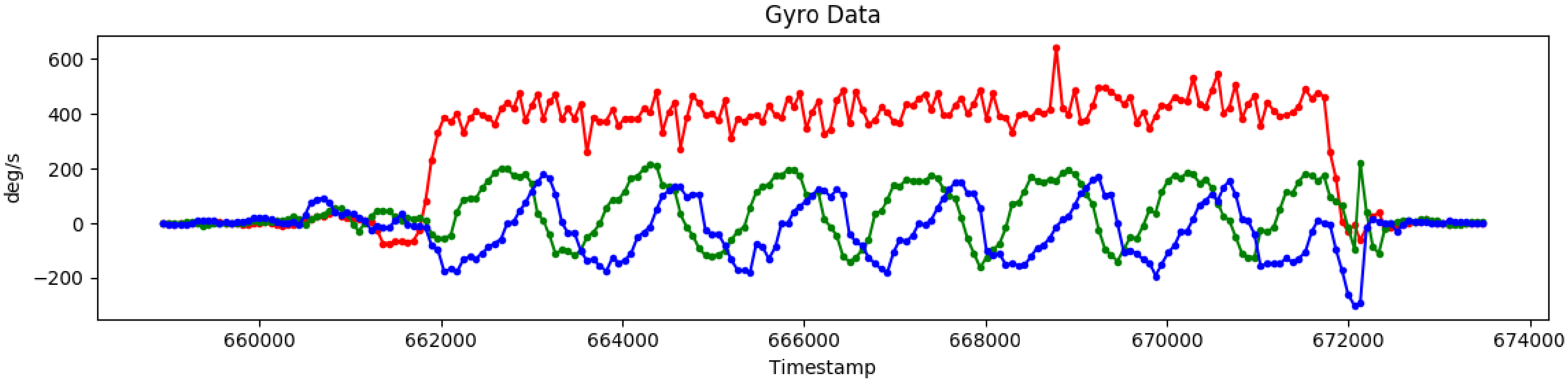
$$measuredTurn = 2.6\ seconds * 140.4243\ degree/seconds = 365.10318\ degree$$

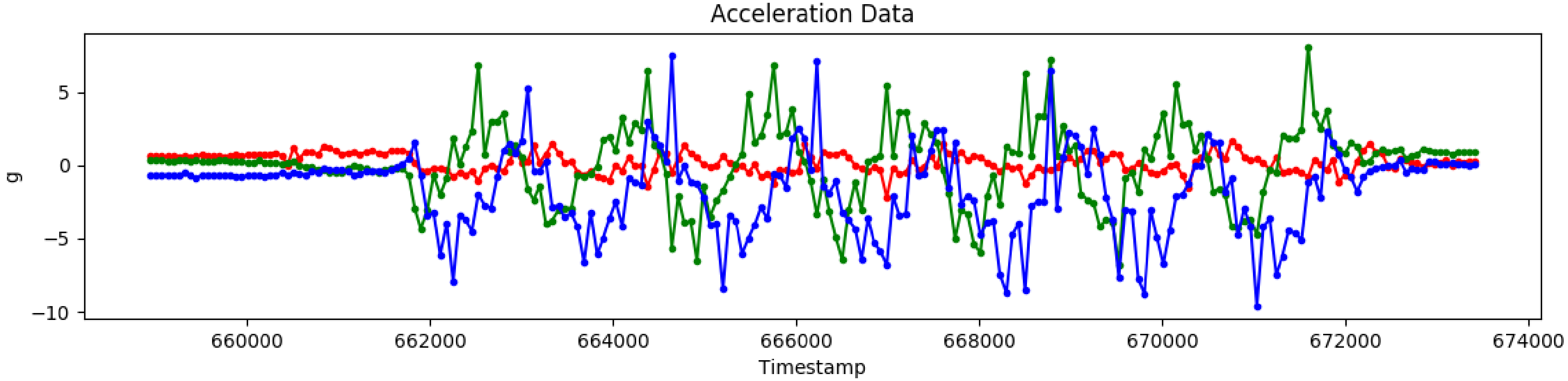
Evaluation: Motion recognition

Preparations and Procedure

- Implementation of Python programs
  - To **receive, save and plot** the measured data
  - To calculate the **Euclidean distance**
  - To calculate the **1st and 2nd derivative**
- Collected the IMU data for each motion and plotted it
- Developed Motion recognition
  - Tried: **Euclidean distance**
  - Tried: **1st and 2nd derivative**
  - Used **wavelength** in the end

Plotted Data for Chest Hooping





Euclidean Distance

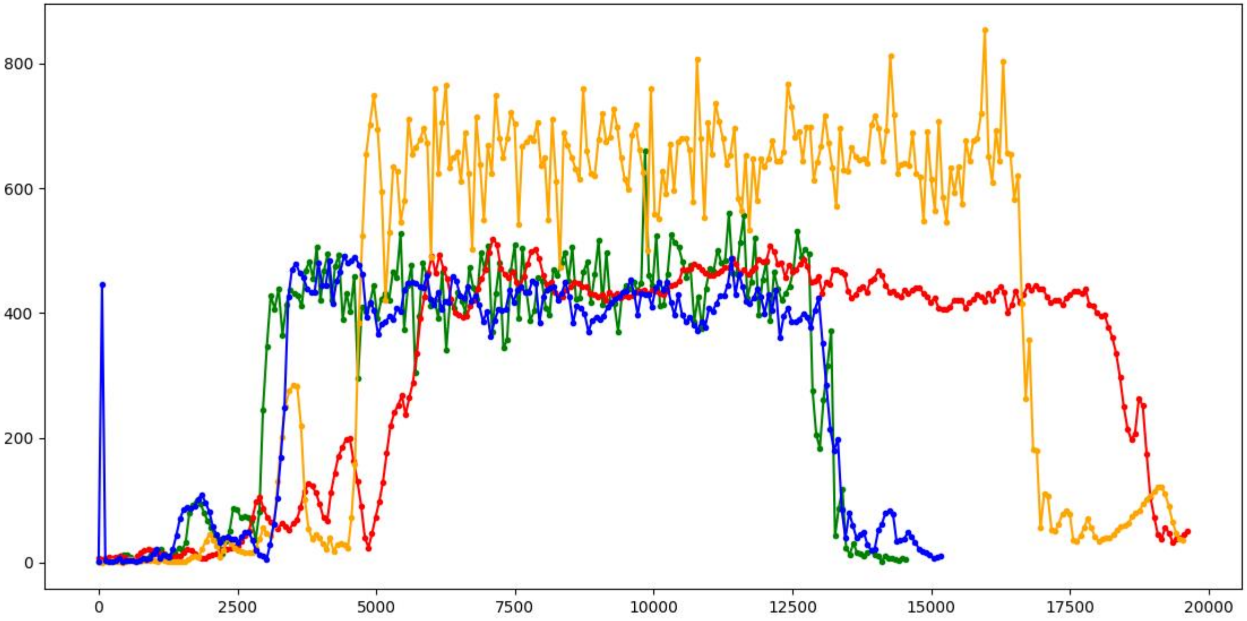
Norm 2 = Sqrt(x^2 + y^2 + z^2)

Red = Hand

Green = Chest

Blue = Waist

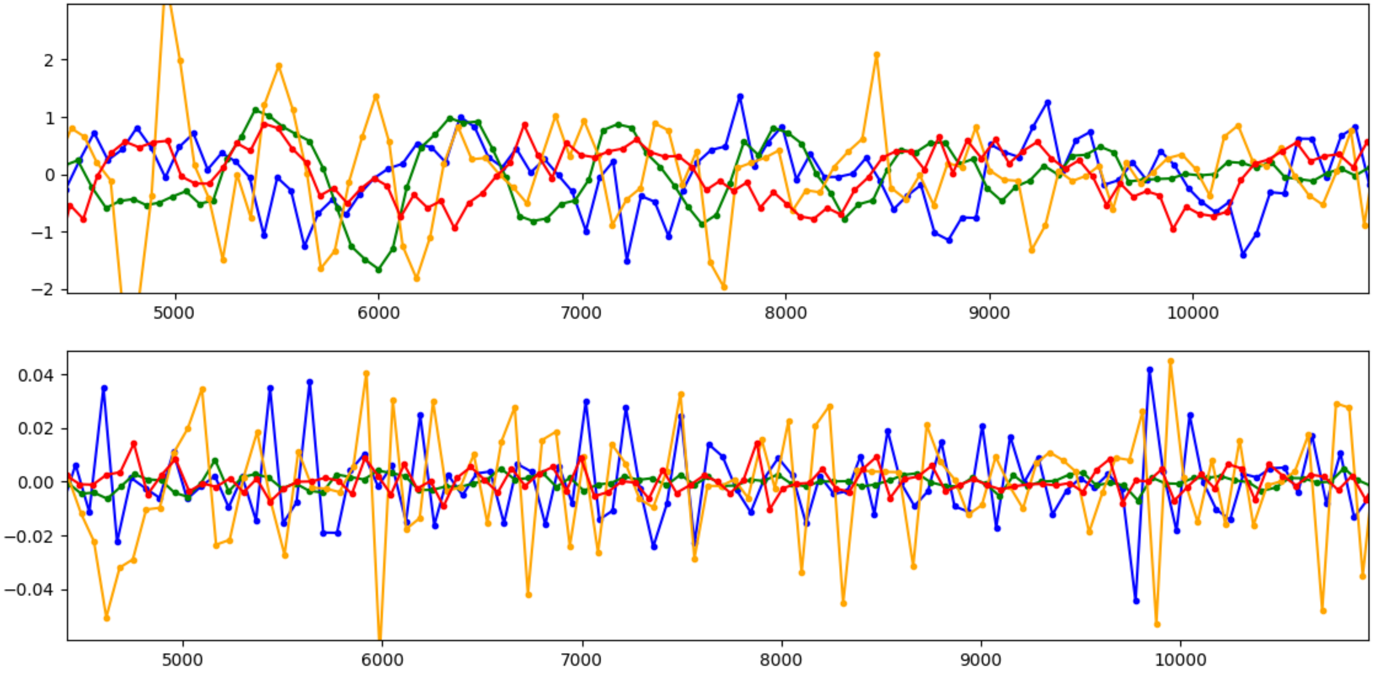
Orange = Knee



→ Hand, Chest and Waist Hooping to similar

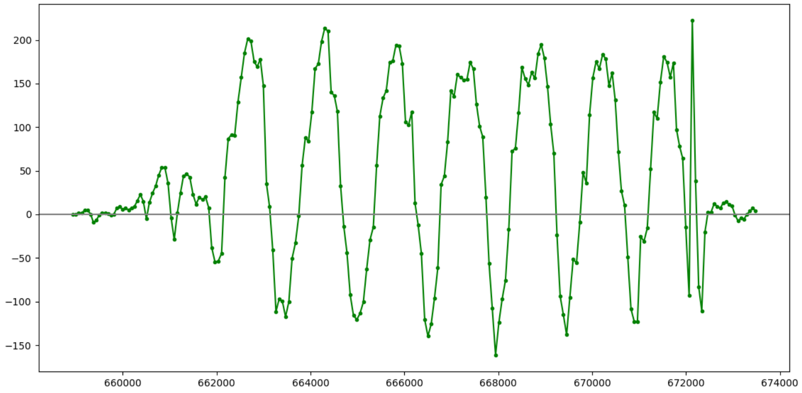
1st and 2nd Derivative of Gyroscope Y-axis

Red = Hand; Green = Chest; Blue = Waist; Orange = Knee

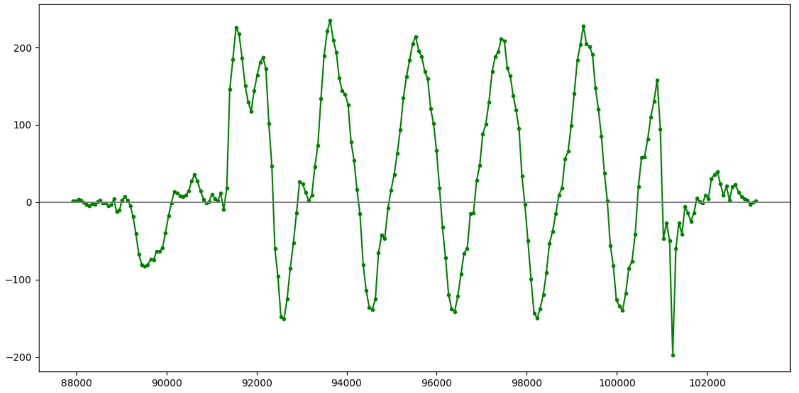


- → Drift-Random Errors make it hard to extract information
- → Intense Computation: the previous values are necessary to calculate the derivative
- → For calculation in real time something simple is needed

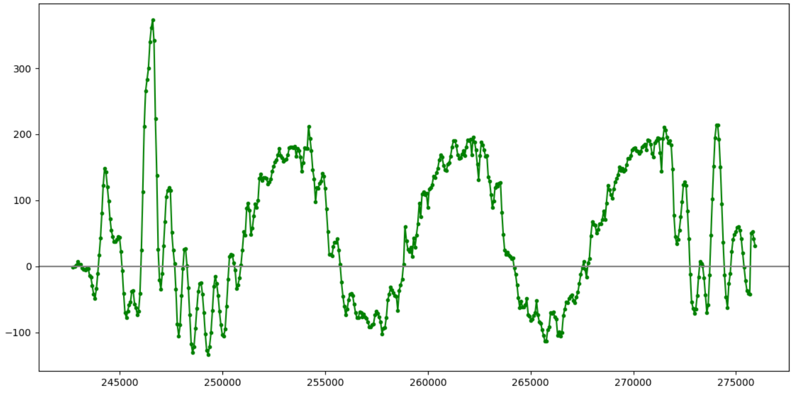
Different Wavelength in Gyroscope Y-Axes



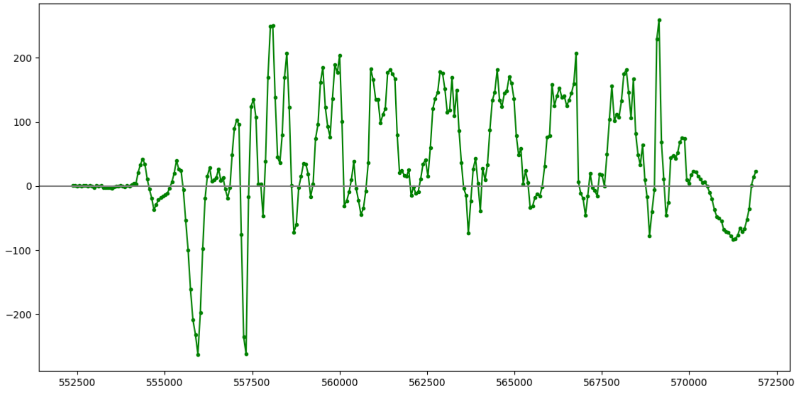
Chest Hooping



Waist Hooping



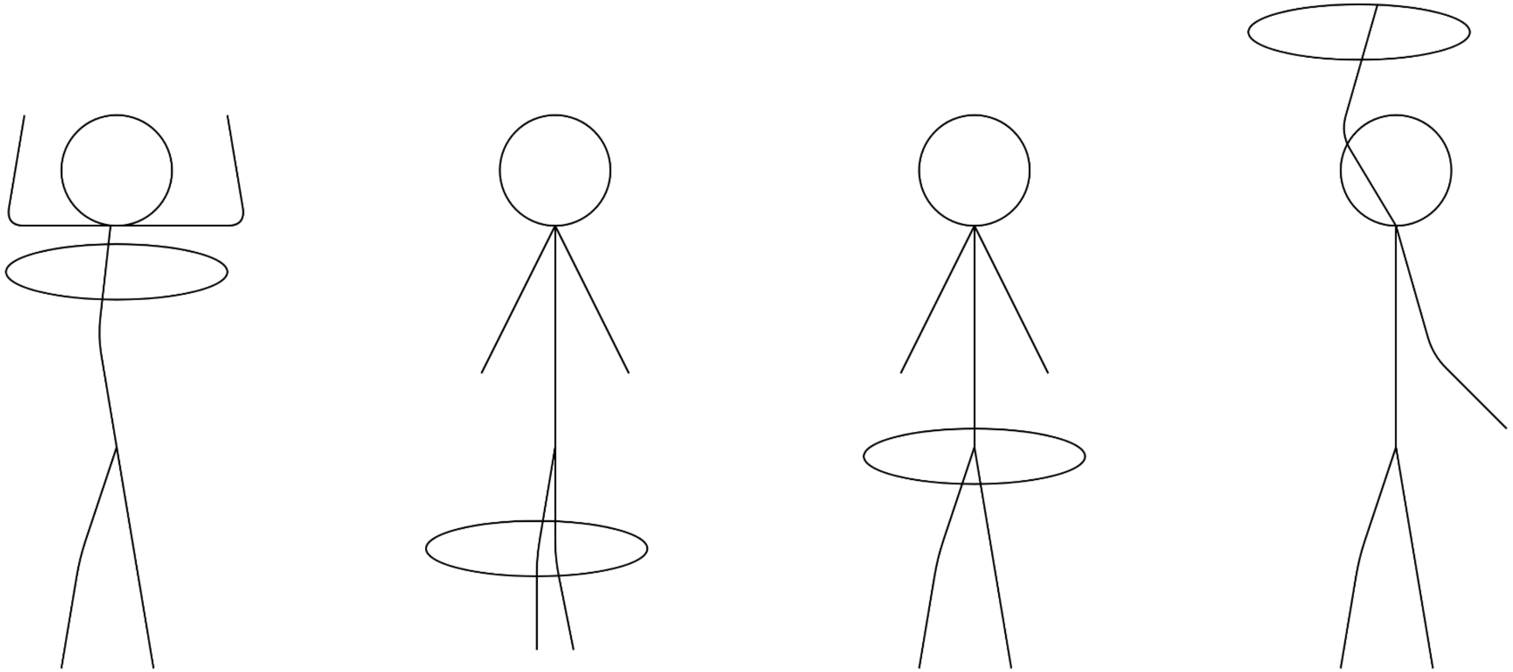
Hand Hooping



Knee Hooping

Average Wavelength

Body parts differ in **motion speed** and **circumference**



Chest Hooping: **1485,2 ms**

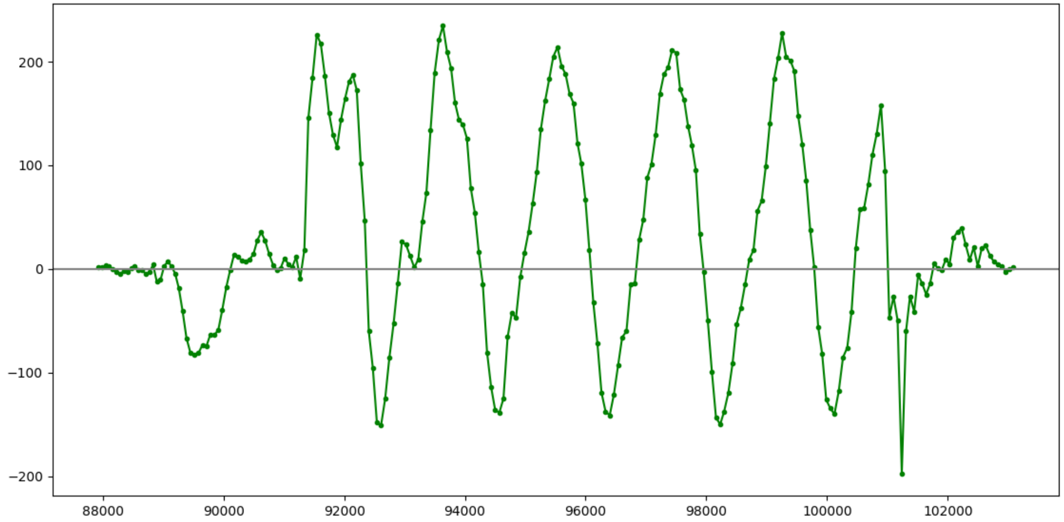
Knee Hooping: **1605,5 ms**

Waist Hooping: **1878 ms**

Hand Hooping: **8234 ms**

Implementation

How can the average wavelength be determined?

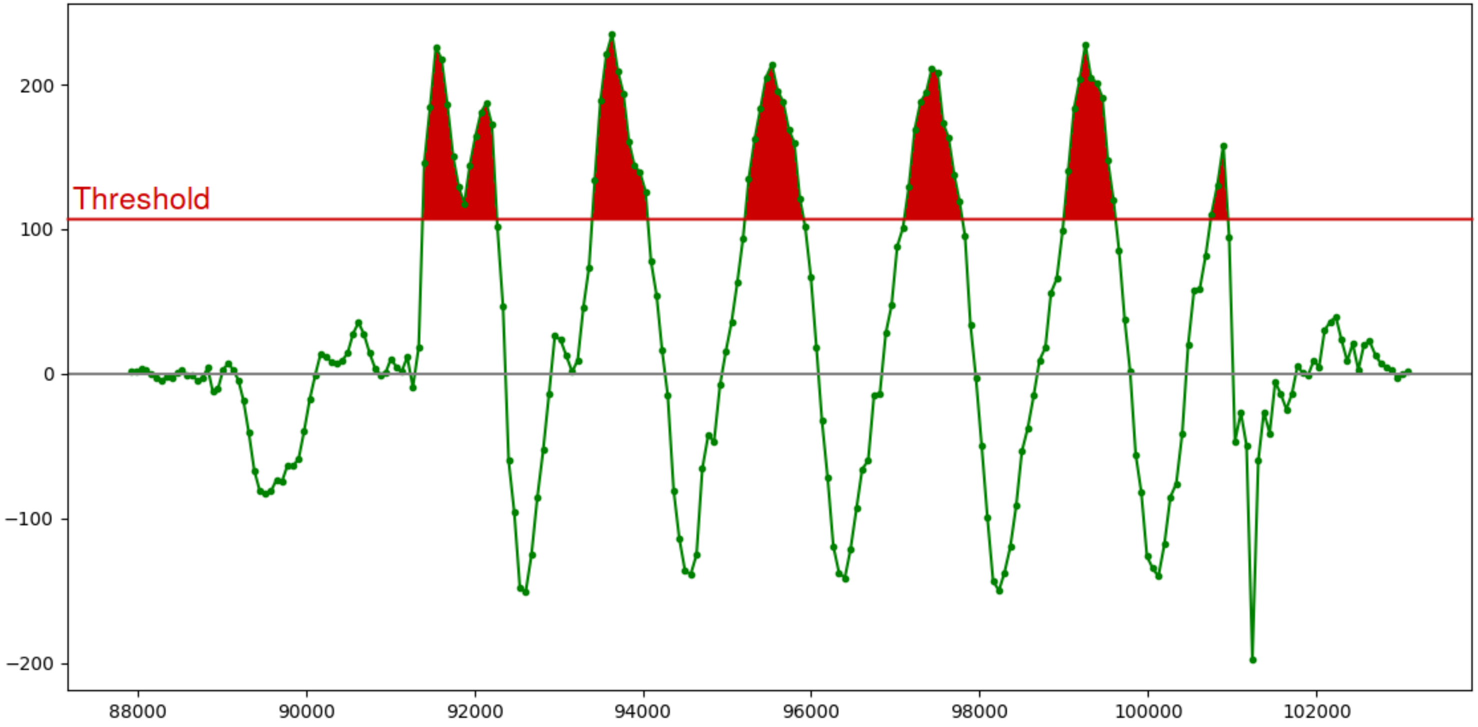


→ By calculating the **time difference** between **maxima**

But how is a maximum detected in code?

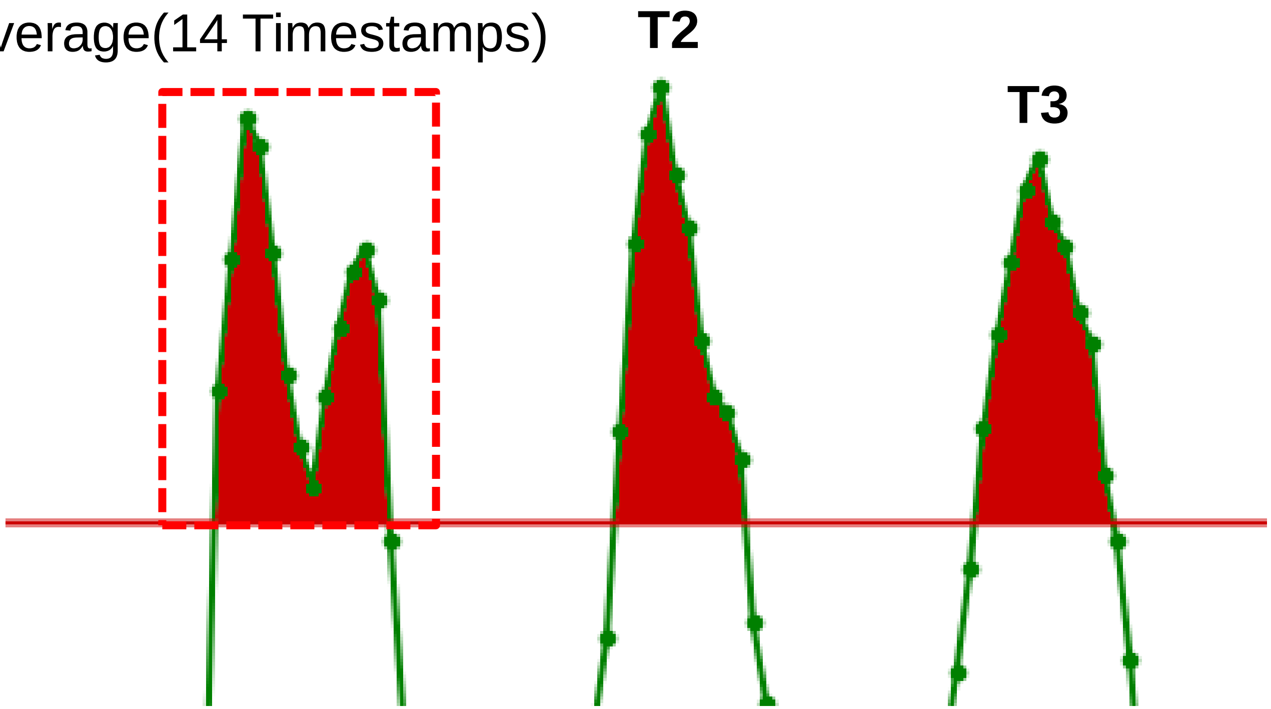
Threshold

Every measurement above threshold is part of maxima



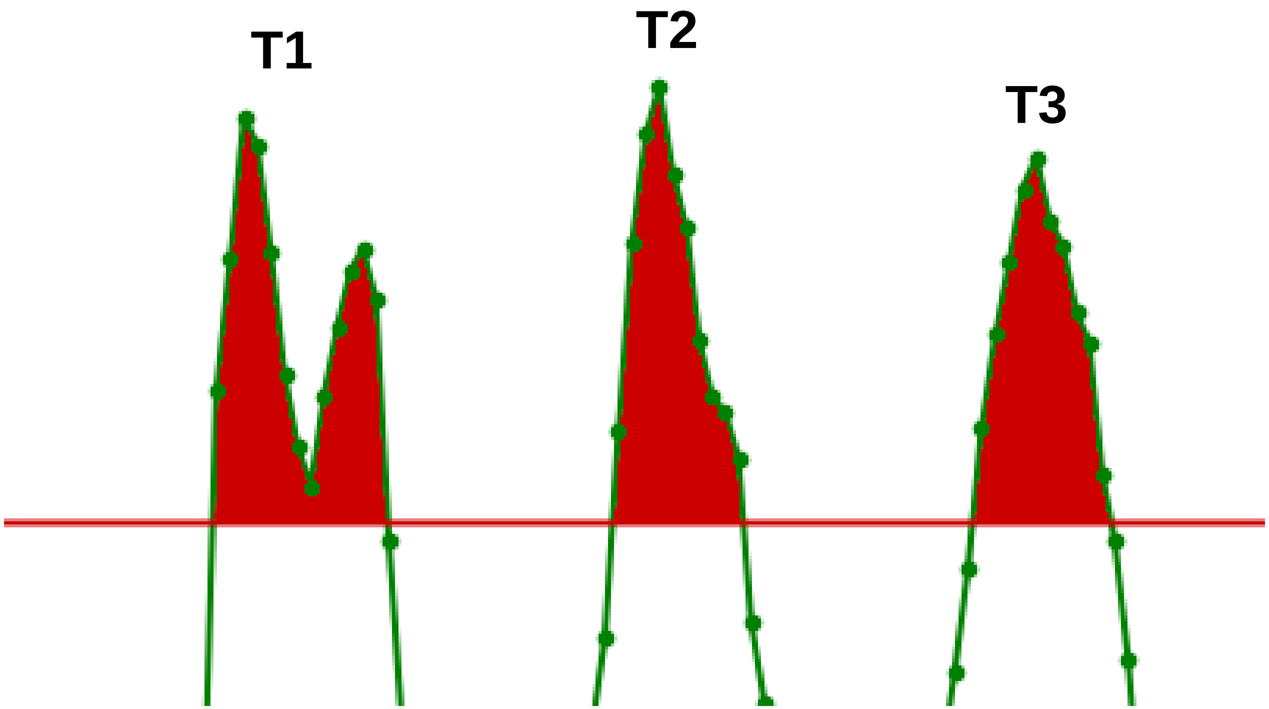
Average Timestamp of one Maximum

T1= Average(14 Timestamps)



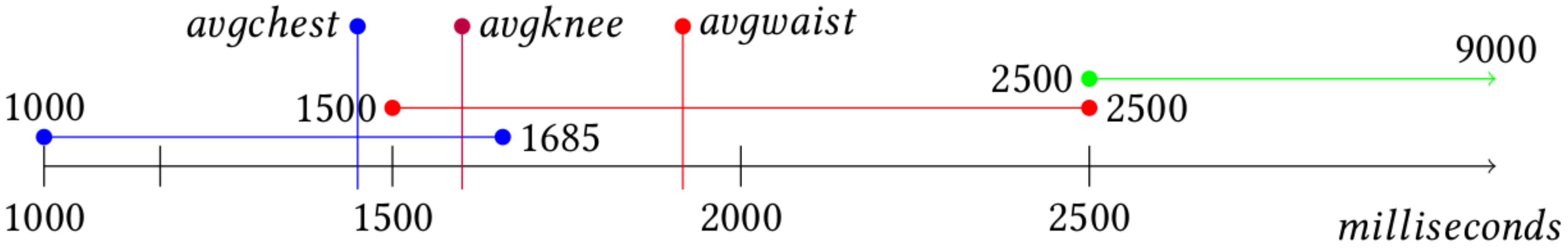
Average Wavelength

Avg. Wavelength = (T2 – T1 + T3 - T2) / 2



Visualization of Wavelength

Defined color areas for R, G & B



Other settings

Spike protection:

New maximum has to be at least 1000 ms after the last maximum

Timeout:

If there is no maximum within 15 seconds later, the hoop is turned off

Potential Issues

Microcontroller

Arduino Pro Mini Atmega 168 used in the beginning

Low memory caused problems:

- FastLED & BMI160 sensor library
  Caused random measurements
- Neopixel & BMI160 library & Bluetooth library
  The more libraries where added, the less LEDs light up
- Bluetooth
  Only short strings were received on the PC

Hand Hooping

- Wavelength of hand hooping ~ 8,2 sec.
- Hooping always causes spikes
- Only Spikes < 1 second are filtered

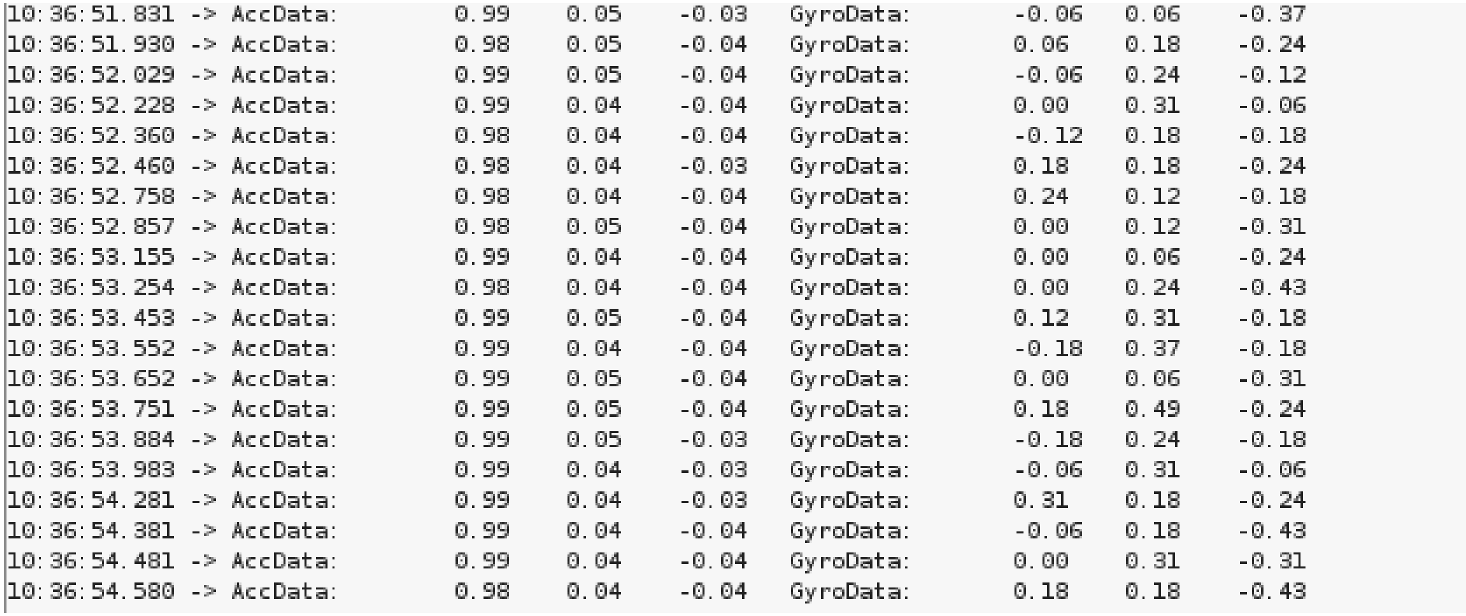
→ Spikes are falsely detected as maxima

→ Wrong color lights up

Future Work

Bias Correction

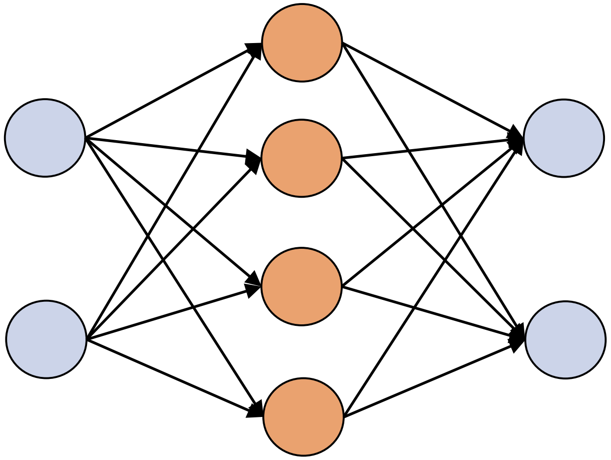
- Use case of the thesis worked fine without calibration
- Sensor noise might be an issue when recognizing more gentle tricks



```
10:36:51.831 -> AccData:        0.99    0.05    -0.03   GyroData:       -0.06   0.06    -0.37
10:36:51.930 -> AccData:        0.98    0.05    -0.04   GyroData:       0.06    0.18    -0.24
10:36:52.029 -> AccData:        0.99    0.05    -0.04   GyroData:       -0.06   0.24    -0.12
10:36:52.228 -> AccData:        0.99    0.04    -0.04   GyroData:       0.00    0.31    -0.06
10:36:52.360 -> AccData:        0.98    0.04    -0.04   GyroData:       -0.12   0.18    -0.18
10:36:52.460 -> AccData:        0.98    0.04    -0.03   GyroData:       0.18    0.18    -0.24
10:36:52.758 -> AccData:        0.98    0.04    -0.04   GyroData:       0.24    0.12    -0.18
10:36:52.857 -> AccData:        0.98    0.05    -0.04   GyroData:       0.00    0.12    -0.31
10:36:53.155 -> AccData:        0.99    0.04    -0.04   GyroData:       0.00    0.06    -0.24
10:36:53.254 -> AccData:        0.98    0.04    -0.04   GyroData:       0.00    0.24    -0.43
10:36:53.453 -> AccData:        0.99    0.05    -0.04   GyroData:       0.12    0.31    -0.18
10:36:53.552 -> AccData:        0.99    0.04    -0.04   GyroData:       -0.18   0.37    -0.18
10:36:53.652 -> AccData:        0.99    0.05    -0.04   GyroData:       0.00    0.06    -0.31
10:36:53.751 -> AccData:        0.99    0.05    -0.04   GyroData:       0.18    0.49    -0.24
10:36:53.884 -> AccData:        0.99    0.05    -0.03   GyroData:       -0.18   0.24    -0.18
10:36:53.983 -> AccData:        0.99    0.04    -0.03   GyroData:       -0.06   0.31    -0.06
10:36:54.281 -> AccData:        0.99    0.04    -0.03   GyroData:       0.31    0.18    -0.24
10:36:54.381 -> AccData:        0.99    0.04    -0.04   GyroData:       -0.06   0.18    -0.43
10:36:54.481 -> AccData:        0.99    0.04    -0.04   GyroData:       0.00    0.31    -0.31
10:36:54.580 -> AccData:        0.98    0.04    -0.04   GyroData:       0.18    0.18    -0.43
```

Motion Recognition with Machine Learning

- Right now: 4 tricks in horizontal position
- So many more tricks
- Reliable motion detection with a neural network



Fancy Light Shows

- Right now: Only 4 different colors
- It is possible to deceive the viewers perception by displaying fancy patterns

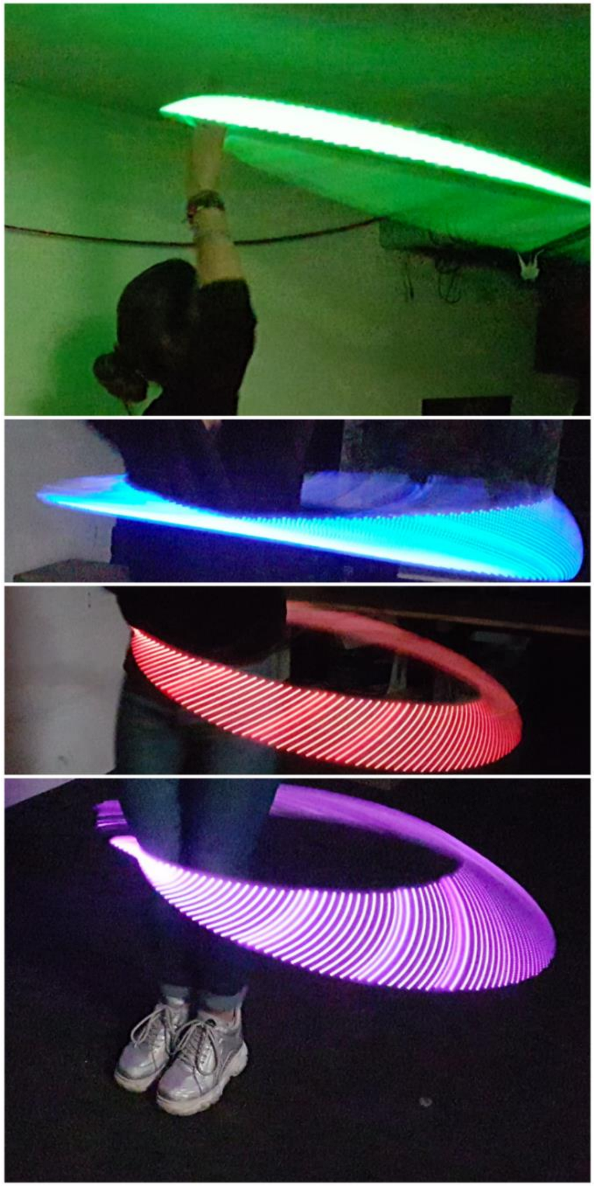
On the go Configuration

- Including a Bluetooth module enables configuration via Phone
- Color, pattern speed, pattern changing, switching between modes, selecting favorite patterns, etc.

Conclusion

Conclusion

- ✓ Fulfilled Hardware requirements
- ✓ Compatible Components
- ✓ Software runs on microcontroller
- ✓ Motion recognition works
- ✓ According colors light up



List of References - Hardware

1. Teensy 3.2, url: https://www.pjrc.com/store/teensy32.html
2. Battery Charger, url: https://de.aliexpress.com/item/32942257898.html?spm=a2g0s.9042311.0.0.27424c4d1sp4Wg
3. BMI160 Sensor, url: https://de.aliexpress.com/item/32718555571.html?spm=a2g0s.9042311.0.0.27424c4d1sp4Wg
4. Discharge Protection Board, url: https://www.amazon.de/gp/product/B01N6ZKDUZ/ref=ppx_yo_dt_b_asin_title_o04_s00?ie=UTF8&psc=1
5. Boost Converter, url: https://www.amazon.de/Homyl-Spannung-Converter-Step-up-Einstellbare-Gr%C3%BCn/dp/B07FQHL9JT
6. Lithium Ion Battery, url: https://de.aliexpress.com/item/32893019555.html?spm=a2g0x.search0104.3.21.71674d989WQkea&ws_ab_test=searchweb0_0%2Csearchweb201602_5_10065_10068_10547_319_317_10548_10696_10084_453_10083_454_10618_10304_10307_10820_10821_537_10302_536_10843_10059_10884_10887_321_322_10103%2Csearchweb201603_53%2CppcSwitch_0&algo_expid=1b9ae9c3-5914-43c8-ab3c-83bc554ffa0d-3&algo_pvid=1b9ae9c3-5914-43c8-ab3c-83bc554ffa0d&transAbTest=ae803_3
7. LED strip, url: https://www.botshop.co.za/product/ws2812b-rgb-led-strip-60-m-5vdc-ip65/
8. Scissors Icon by benjsperry, url: https://icon-icons.com/de/symbol/Schere/50046
9. On Off Button, url: https://www.amazon.de/gp/product/B007QAJLDE/ref=ppx_yo_dt_b_asin_image_o04_s01?ie=UTF8&psc=1

List of References

1. Page 4: hulathehoop.de: LED Smart Hoop - The MATRIX, url: https://www.youtube.com/watch?v=kvTNCoBOkfc
2. Page 9: Future Hoop Pro from Moodhoops, url: https://moodhoops.com/shop/futurehoop-pro/
3. Page 9: MATRIX Hoop from Hula the Hoop, url: https://hulathehoop.de/produkt/matrix-hoop/
4. Page 9: Atomic V from Astral Hoops, url: https://shop.astralhoops.com/collections/hoops/products/in-stock-atomic-v-programmable-led-hoop-designer
5. Page 45: Future Hoop Pro from Moodhoops, url: https://moodhoops.com/shop/futurehoop-pro/

Demo

Thank you