INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE x maillon mobility

Final Oral: Intermediate Lightweight Tricycle Vehicle for Daily Usage

Achille Caute
Brian Biendou
Marie Brunetto
Timothé Bigot
Tutor: Pr Thierry Monteil

23/01/2025

5A ISS-A2
Promotion 57-58
Year 2024/2025

INSA Toulouse

Introduction

What is Maillon Capitole and its aim?

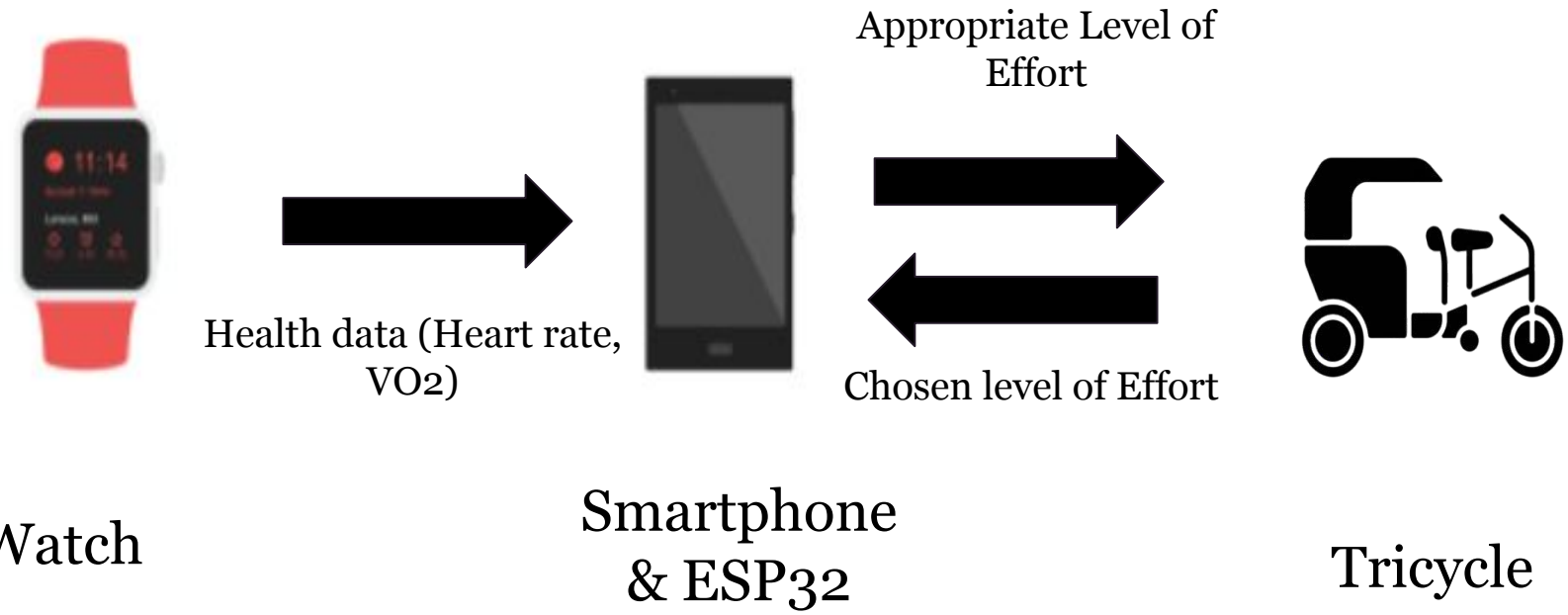
Created for peri-urban rides: Work, school, shopping

Speed max: 25km/h

Can carry:
- various objects through its modules
- up to 2 adults and 2 children or 1 adult and 4 children

Maillon Capitole with some of its modules
https://www.maillonmobility.com/maillon capitole

Heart rate regulation project

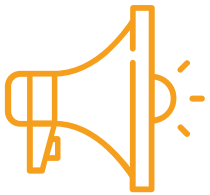


Speaker: Timothé

Current state of our system

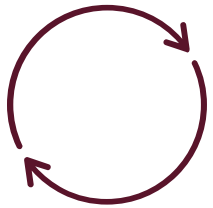
Communication
Watch-Phone

Phone Application



Communication
Phone-ESP

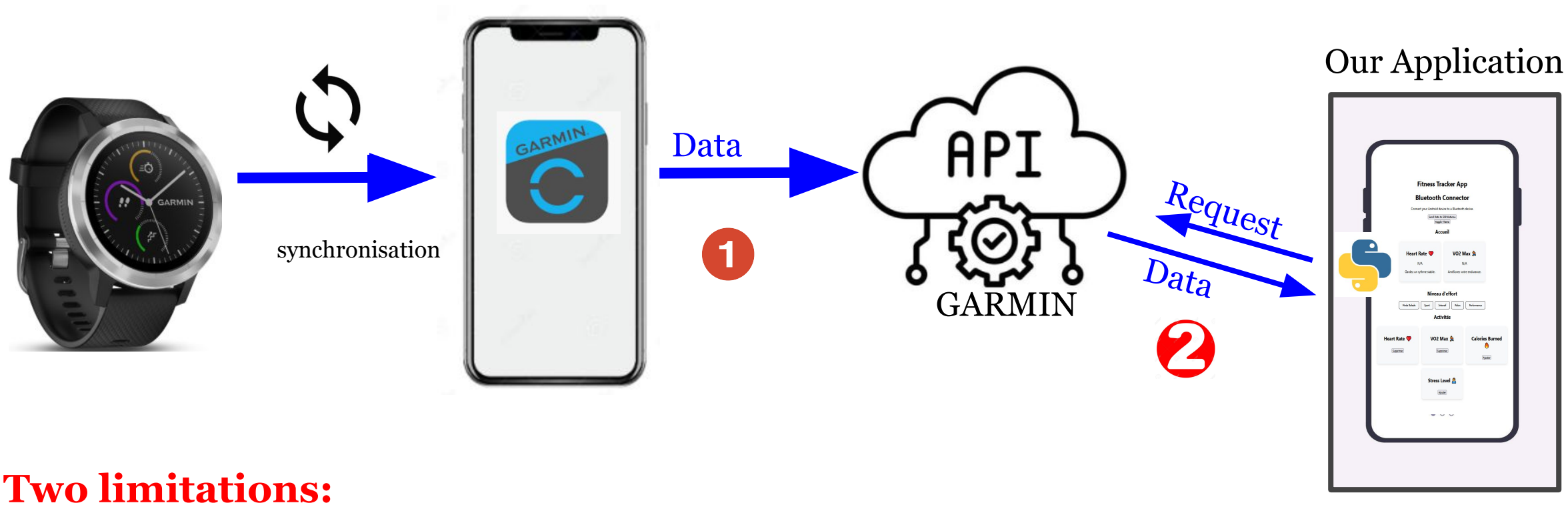
Communication
ESP32-Testbench



Control Law

Communication Watch-Phone

First approach



Two limitations:

- Data frequency (Only when i open application GarminConnect)
- Limitation on the number of requests (10 requests every 90 min)

Speaker: Brian

Communication Watch-Phone

First solution

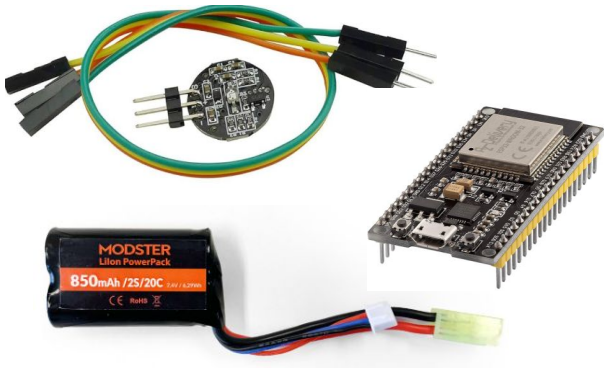
- Request access to the Garmin Health API.
- This API is reserved for companies.
- Charges may apply depending on the number of requests made.

Second solution

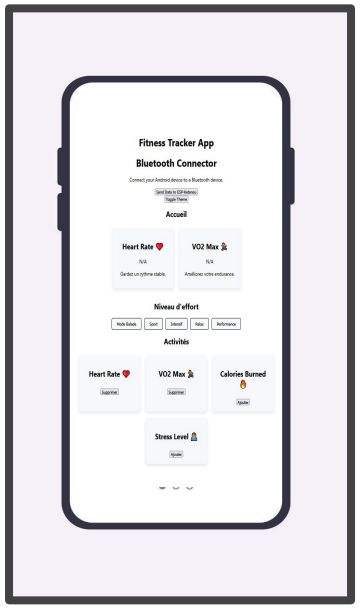
ANT+

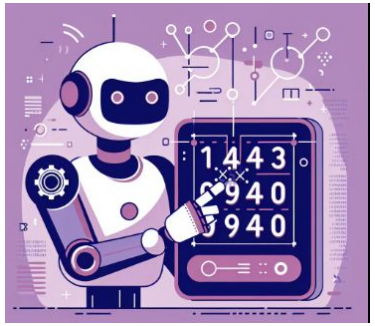
Speaker: Brian

Second & Last approach



currently being tested



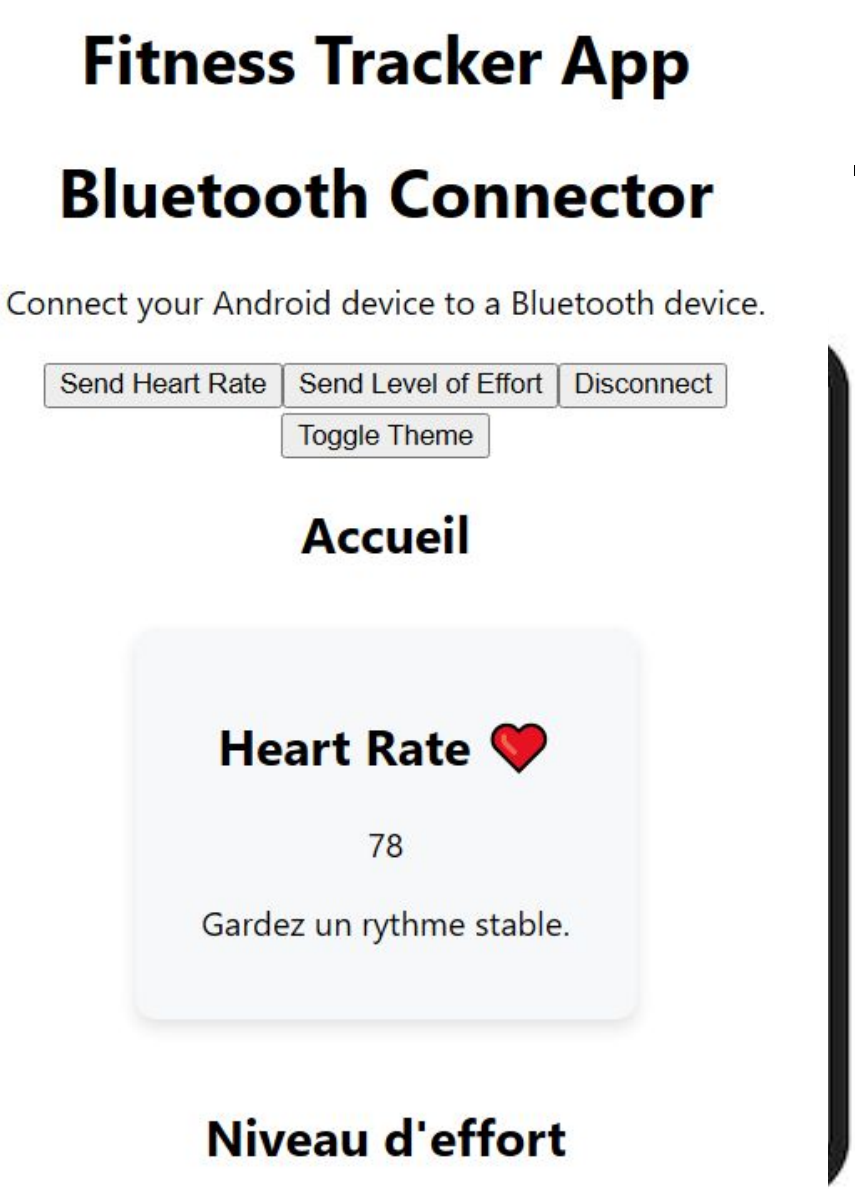


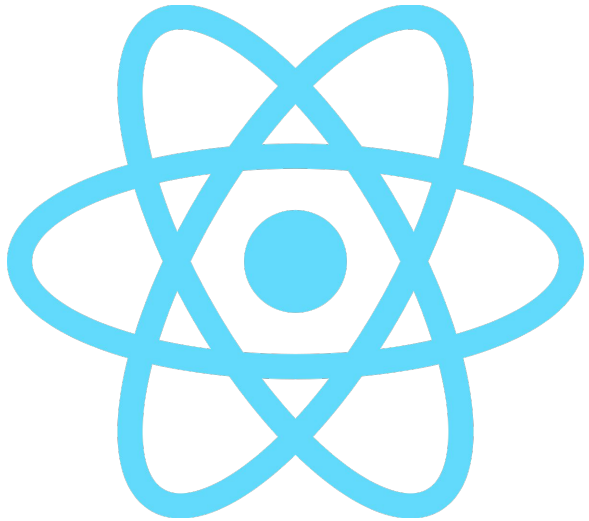
generate heart rates

Speaker: Brian

Phone Application

- Front End





Interface of our application made using the framework React

Phone application - Back End

Website or Application?

We choose to develop on a website:

Pros:

- Does not require installation
- Web requests could be needed anyway for future applications
- Easier to deploy as a test
- Can still be later converted as an application

Cons:

- Requires connexion
- Consumes more through wireless communication

Phone application - Back End

Backend of our code made with JavaScript, compatible with React

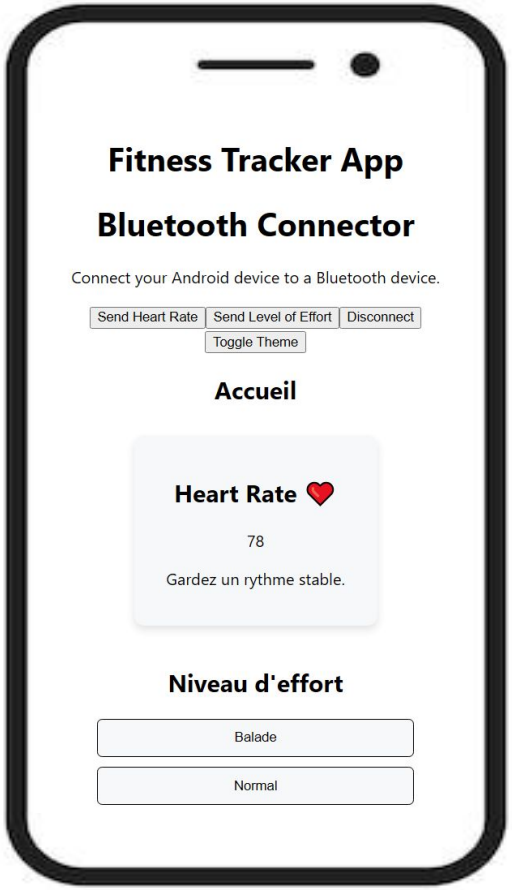
Hosting our website on Github

Communication Phone-ESP

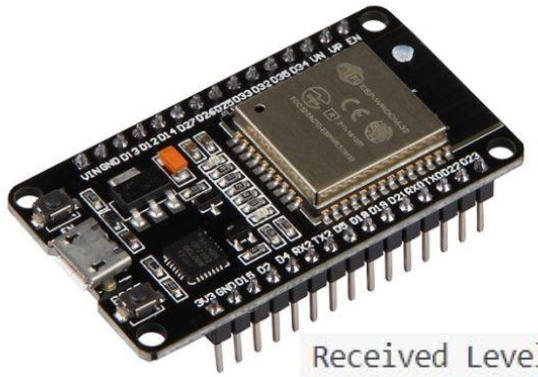
Communication Phone-ESP

- Available on Both Phone and ESP
- Convenient Range
- Low energy consumption
- Existing library for JavaScript

Communication Phone-ESP



Sending
Heart Rate
Level of Effort



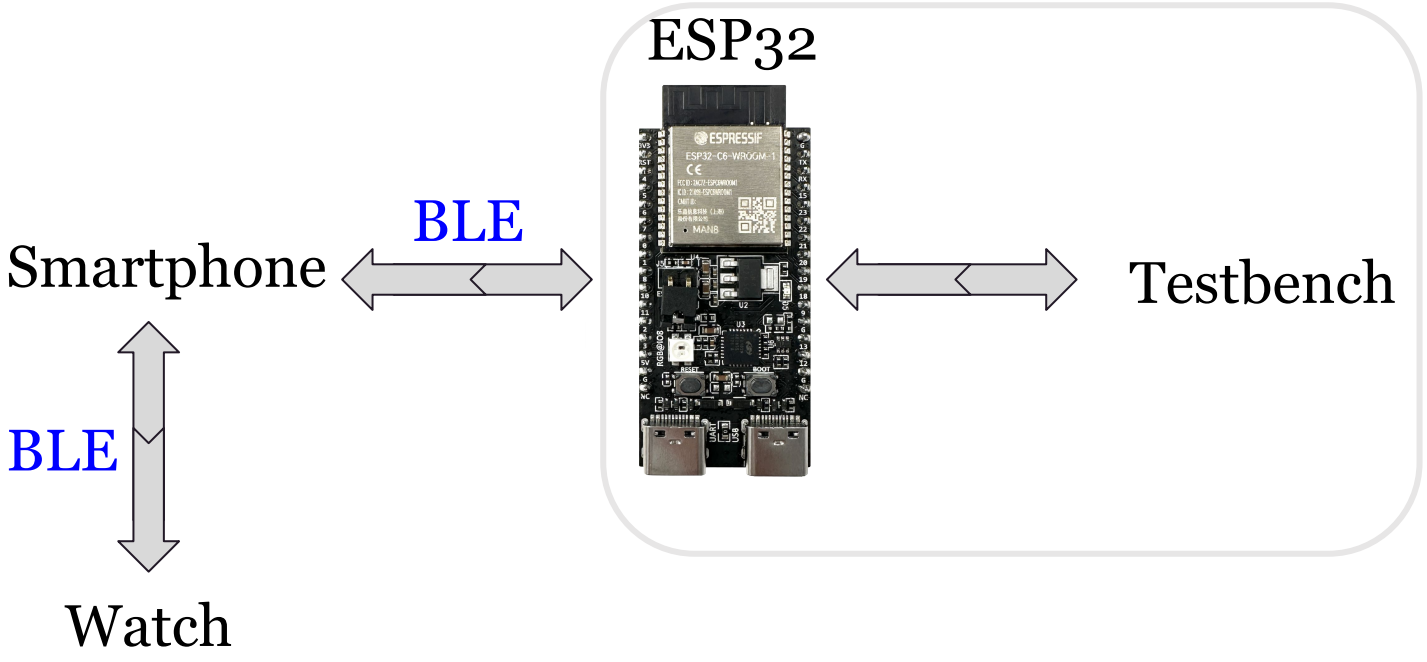
```
Received Level of Effort: 0
Received Level of Effort: 1
Received Level of Effort: 2
Received Level of Effort: 0
Received Heart Rate: 76
Received Heart Rate: 82
Received Heart Rate: 80
Received Heart Rate: 77
Received Heart Rate: 81
```

Communication ESP-Testbench

Communication and connectivity with the Testbench

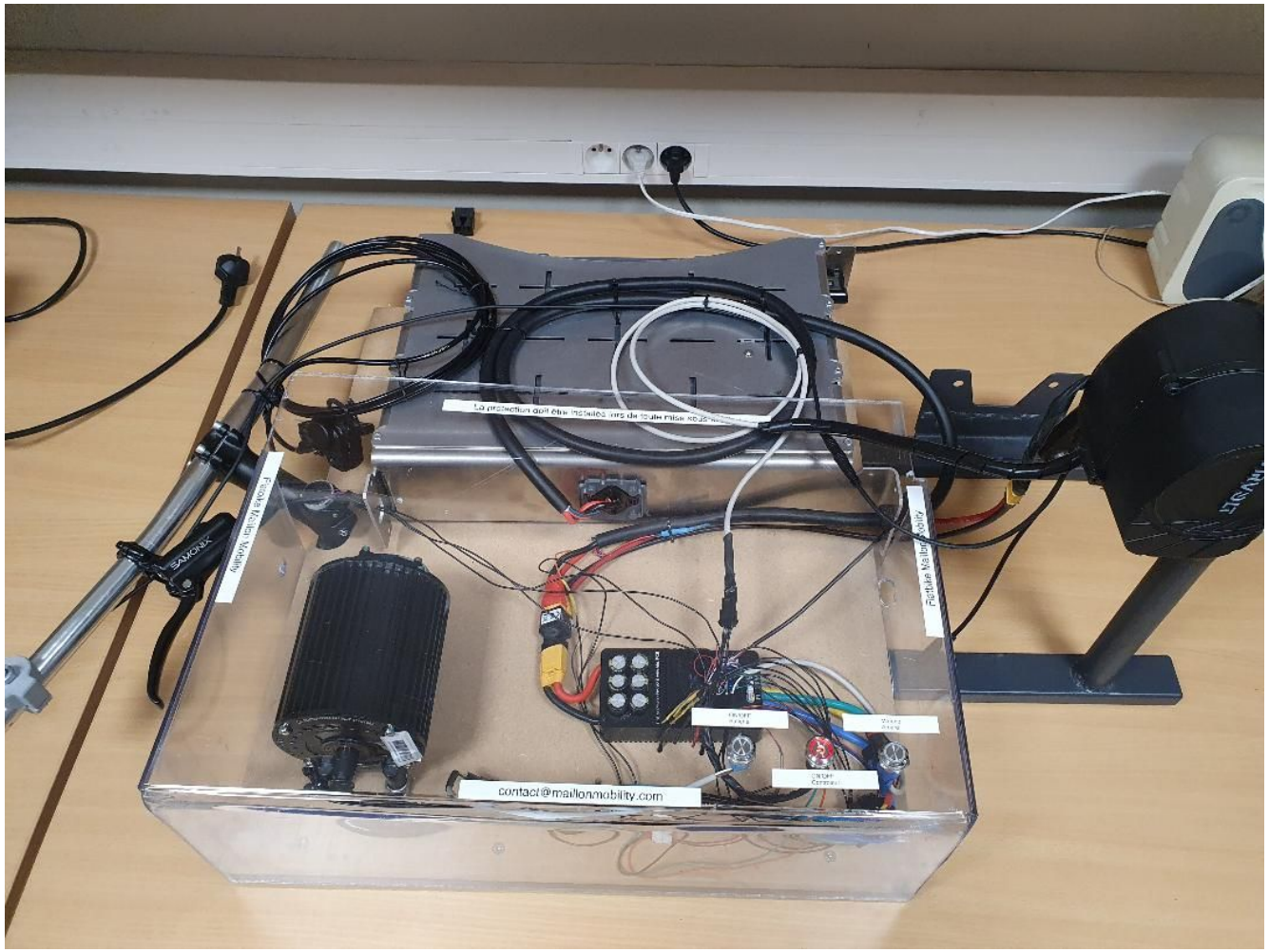
Problematic:

How to get the data from Flipsky speed controller and the GenePi Generator in order to understand the system and then applicate the level of effort and control law regulation?

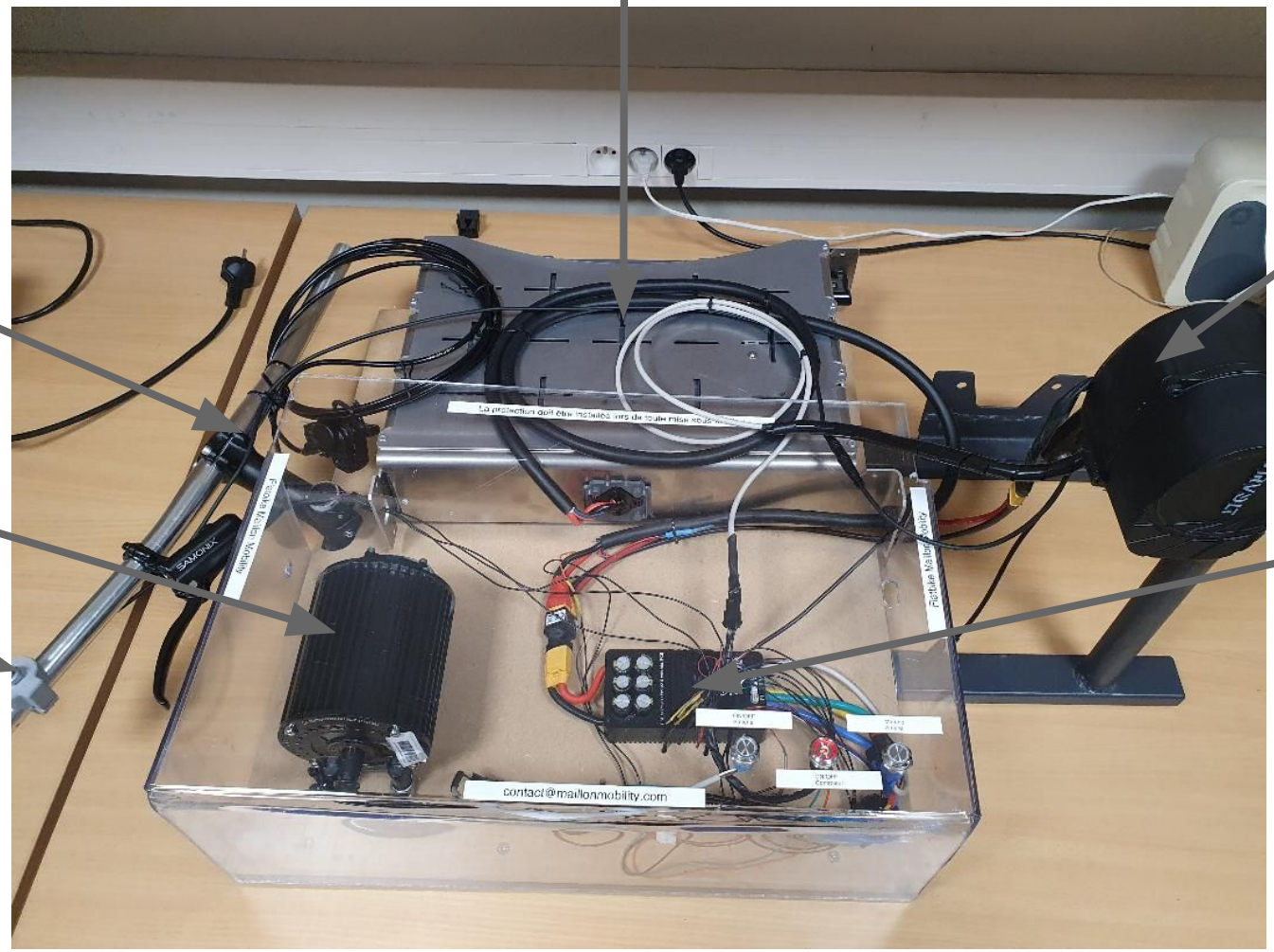


Understand how it work
Reverse engineering
Lack of documentation
Time consuming
Exchanges with Maillon Mobility

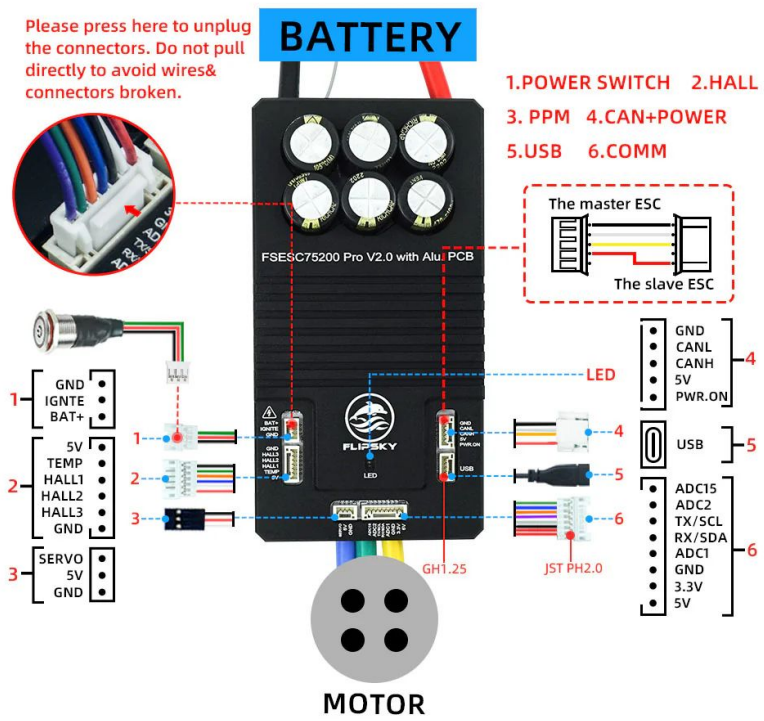
Testbench overview:



Testbench overview:



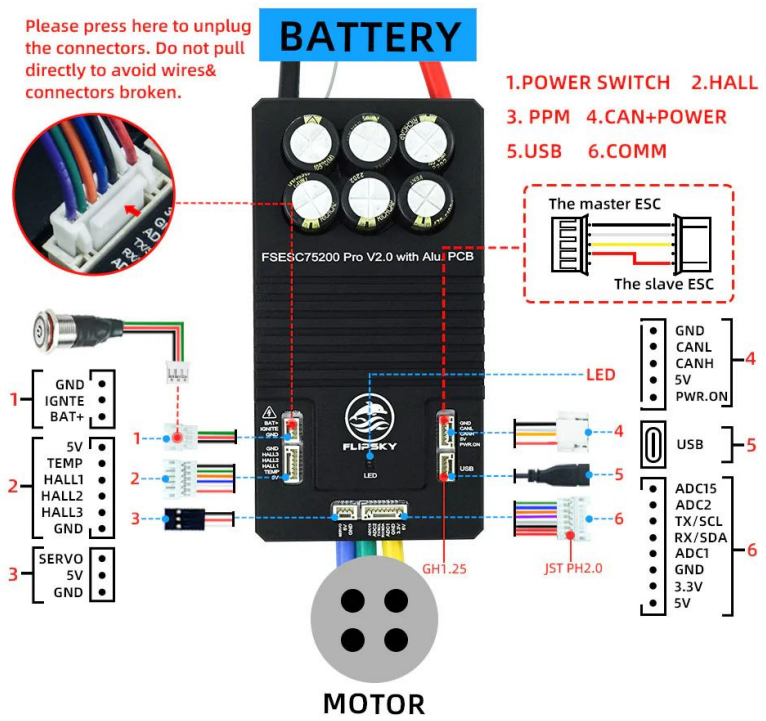
1. Data acquisition from Flipsky with VESC Tool



Caracterise and centralize all the data needed to set the right torque value

Extract data from using VESC

1. Data acquisition from Flipsky with VESC Tool

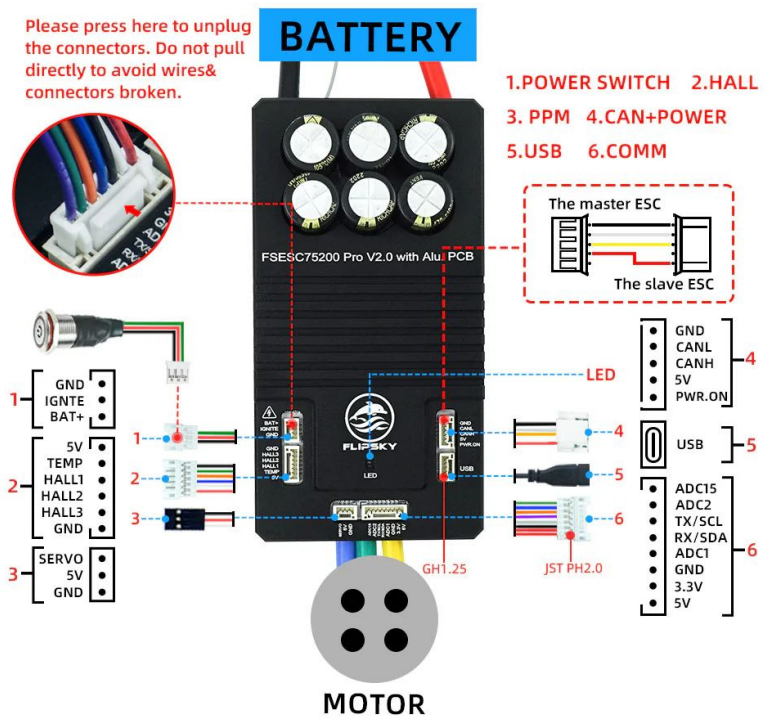


Caracterise and centralize all the data needed to set the right torque value

Extract data from using VESC

1. BLE/UART with smartphone:
   not able to manage and treat data

1. Data acquisition from Flipsky with VESC Tool



VESC® Free of charge edition TOOL

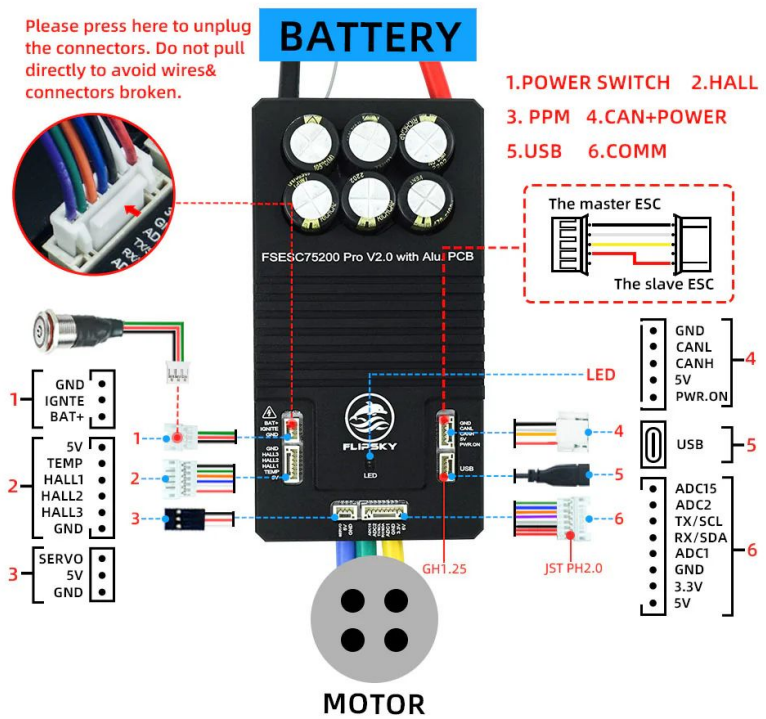
Caracterise and centralize all the data needed to set the right torque value

Extract data from using VESC

1. BLE/UART with smartphone:
   not able to manage and treat data

2. CAN with computer:
   not all the material
   the GenePi not in CAN yet
   next version of flatbike: all in CAN

1. Data acquisition from Flipsky with VESC Tool



VESC® Free of charge edition TOOL

Caracterise and centralize all the data needed to set the right torque value

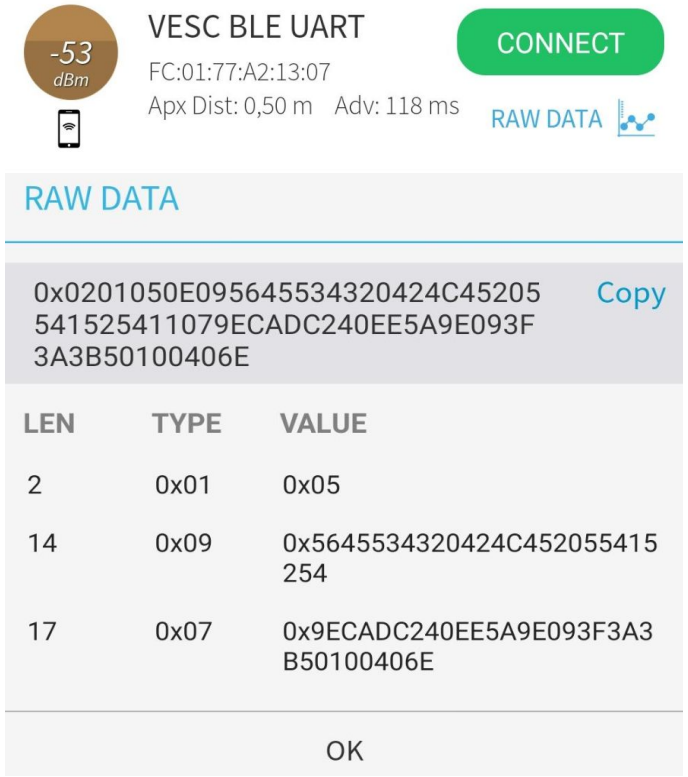
Extract data from using VESC

1. BLE/UART with smartphone:
   not able to manage and treat data

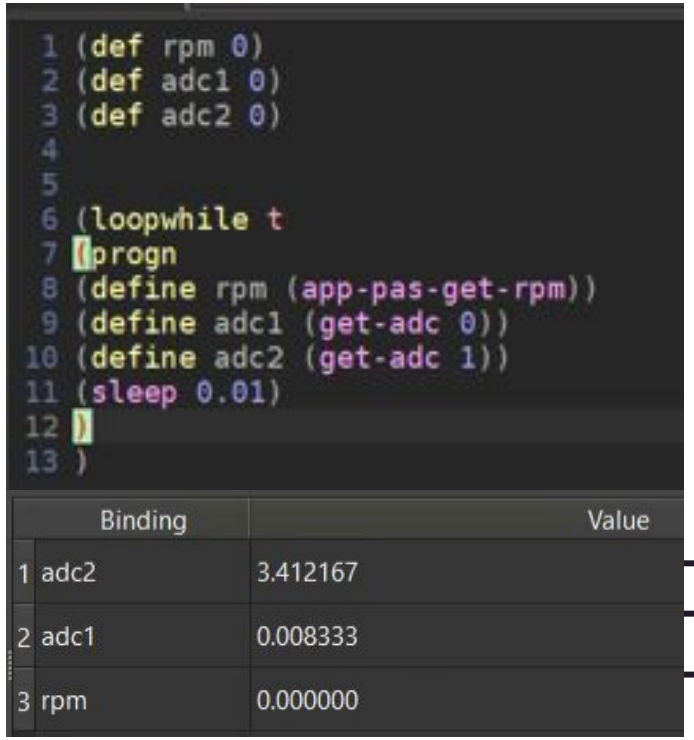
2. CAN with computer:
   not all the material
   the GenePi not in CAN yet
   next version of flatbike: all in CAN

3. **UART with computer & LISP interface:**
   issue with and motor parameters
   but display of data

1. Data acquisition from Flipsky with VESC Tool



BLE frames on phone



Data from VESC

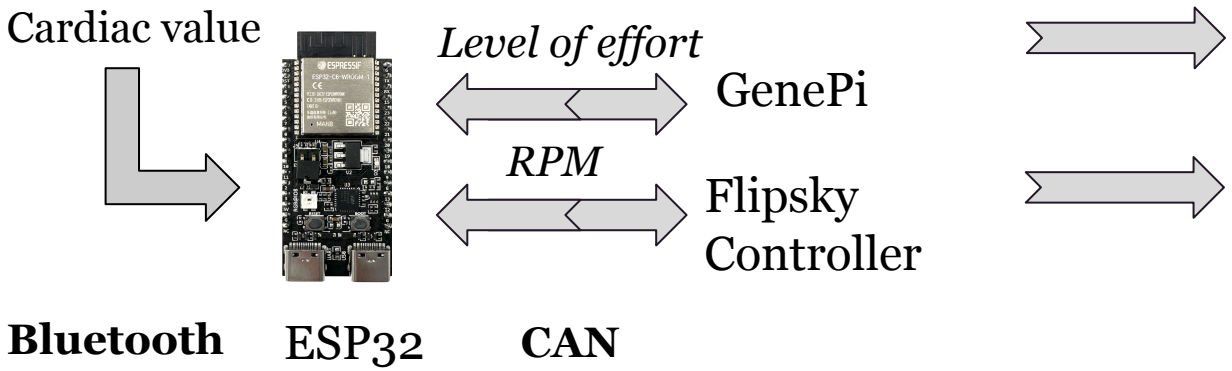
VESC Parameters Updates:
- Version 6.02 instead of 6.05
- Set to “ADC and PAS”
- App & Motor Configuration Files

→ Brake value: 0.95-3.41
→ Forward motor: not used
→ Rpm: 0-125 rpm/mn

→ Multiple issue with the testbench
Time consuming

Speaker: Timothé

2. Data transmission to ESP32



Define and implement an architecture for data exchanges and computing

Select a protocol that is possible to use with both ESP32 & Flipsky controller and having the data

Architecture choice:
Compute of data on the ESP32
Bidirectional communication
Bluetooth incoming data to consider

Speaker: Timothé

2. Data transmission to ESP32

Cardiac value

Level of effort

GenePi

RPM

Flipsky Controller

Bluetooth ESP32 **CAN**

Define and implement an architecture for data exchanges and computing

Select a protocol that is possible to use with both ESP32 & Flipsky controller and having the data

Protocol Choice:
UART: Easier solution, but already used by the motor
CAN: Complex, external module, best solution, in progress
SPI/I2C

Architecture choice:
Compute of data on the ESP32
Bidirectional communication
Bluetooth incoming data to consider

Speaker: Timothé

2. Data transmission to ESP32

Cardiac value

Bluetooth

ESP32

Level of effort (CAN/Digital)

GenePi

RPM

Flipsky Controller

CAN

Define and implement an architecture for data exchanges and computing

Select a protocol that is possible to use with both ESP32 & Flipsky controller and having the data

Protocol Choice:
UART: Easier solution, but already used by the motor
CAN: Complex, external module, best solution, in progress
SPI/I2C

Architecture choice:
Compute of data on the ESP32
Bidirectional communication
Bluetooth incoming data to consider

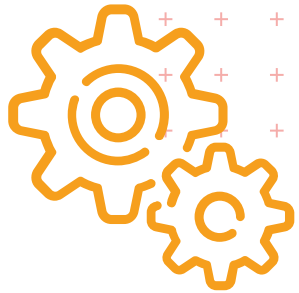
Difficulties:
Two differents programming interface (VESC LISP & ESP32) to make it work together
Adding the data incoming from smartphone

→ Still working on it

Speaker: Timothé

Control Law

Motivation for a non-linear approach:



- Linear approach : linearization around an operating point.

⟶ The modelization error is bigger as the range increase.

⟶Hence we use a pseudo linear approach, the Takagi Sugeno model.

HR dynamic model:

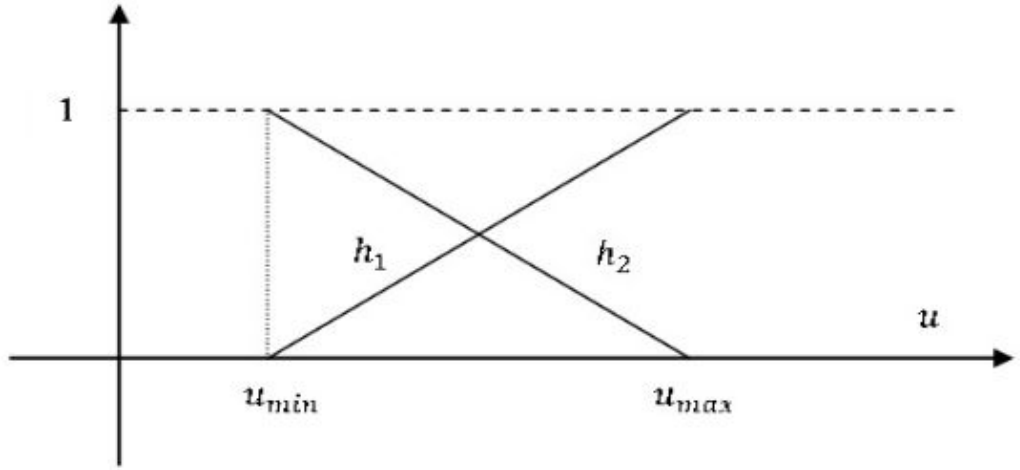
- Takagi-Sugeno non-linear fuzzy model
- power as input, HR as output
- Small dynamic (between 20-150W)
- Fairly simple (approximation with a fuzzy set of linear models)

$$\begin{cases} x(t+1) = Ax(t) + \sum_{i=1}^{r} h_i(z(t))B_i u(t) \\ y(t) = Cx(t) \end{cases}$$

[*] Sami MOHAMMAD, Thierry Marie GUERRA, Jean Marie GROBOIS, and Bernard HECQUET. Heart rate control during cycling exercise using takagi-sugeno models. IFAC Proceedings Volumes, 44(1):12783–12788, 2011. 18th IFAC World Congress.

HR dynamic model:

- Membership function:



Schedule the contribution of the member subsystems.

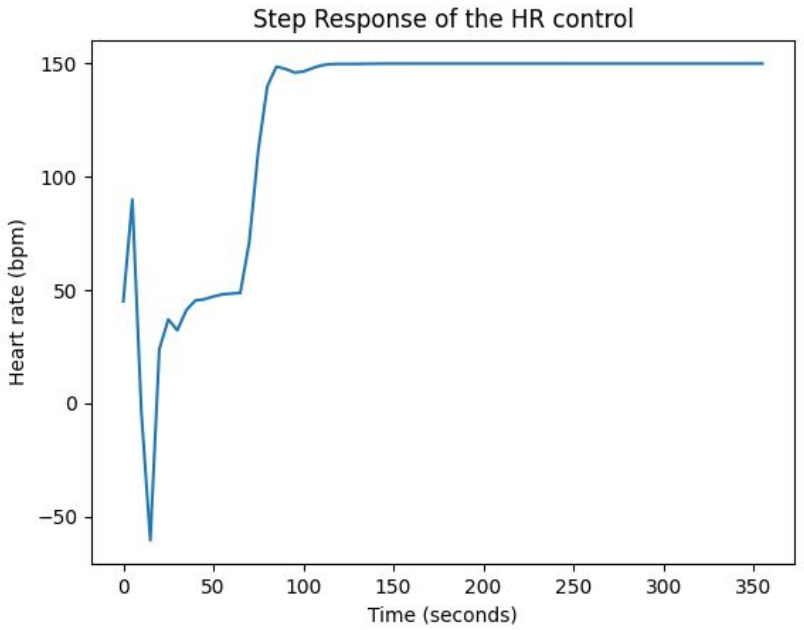
HR control law:

- Guaranteed stability over the input range
- Scheduled state feedback
- Low computational needs

Scheduled state feedback:

$$u(t) = -F_z P_z^{-1} x(t)$$

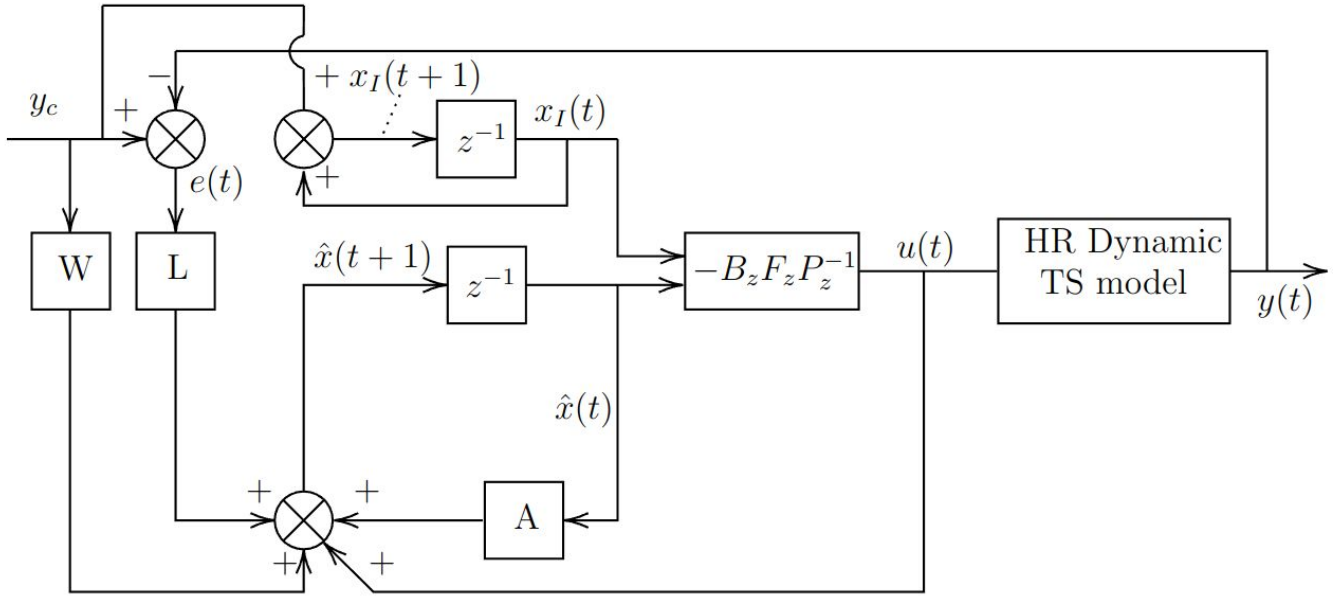
With Pz the Lyapunov matrices of each membership subsystems.



[*] Thierry Marie Guerra and Laurent Vermeiren. Lmi-based relaxed nonquadratic stabilization conditions for nonlinear systems in the takagi–sugeno's form. Automatica, 40(5):823–829, 2004.

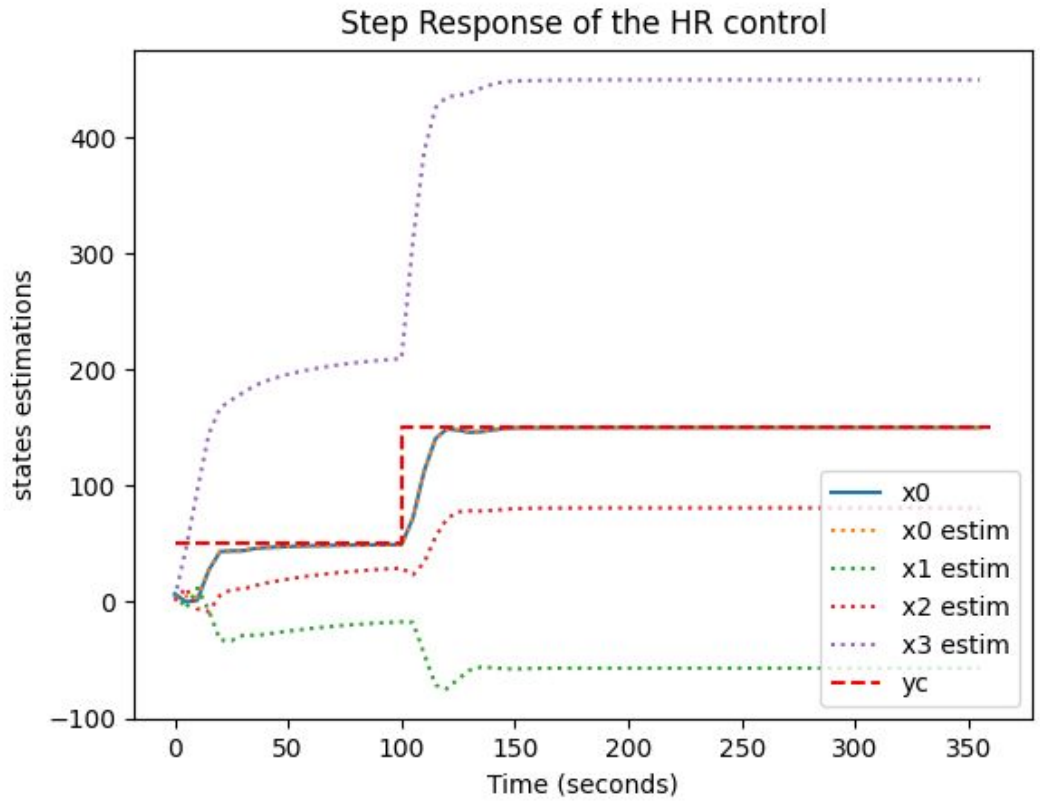
HR control law with state observer, Implementation:

- Use of a state observer to get an estimation of the states
- The observer has a faster dynamic then the model.



HR control law with state observer, Implementation:

- Use of a state observer to get an estimation of the states
- The observer has a faster dynamic then the model.
- Some troubles at first as one pole of the augmented state space is not observable.
- But it works like a charm!



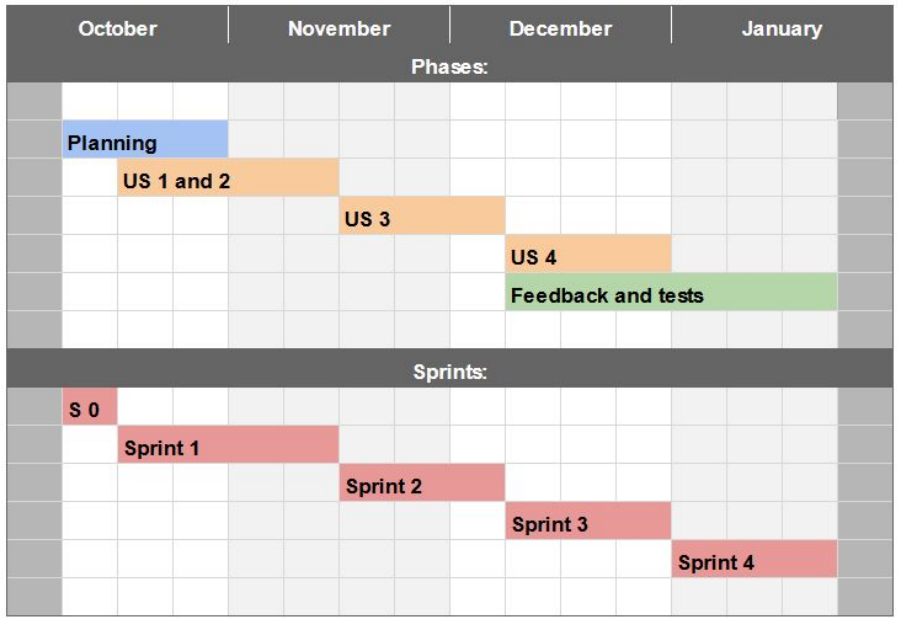
HR control law with state observer, Implementation:

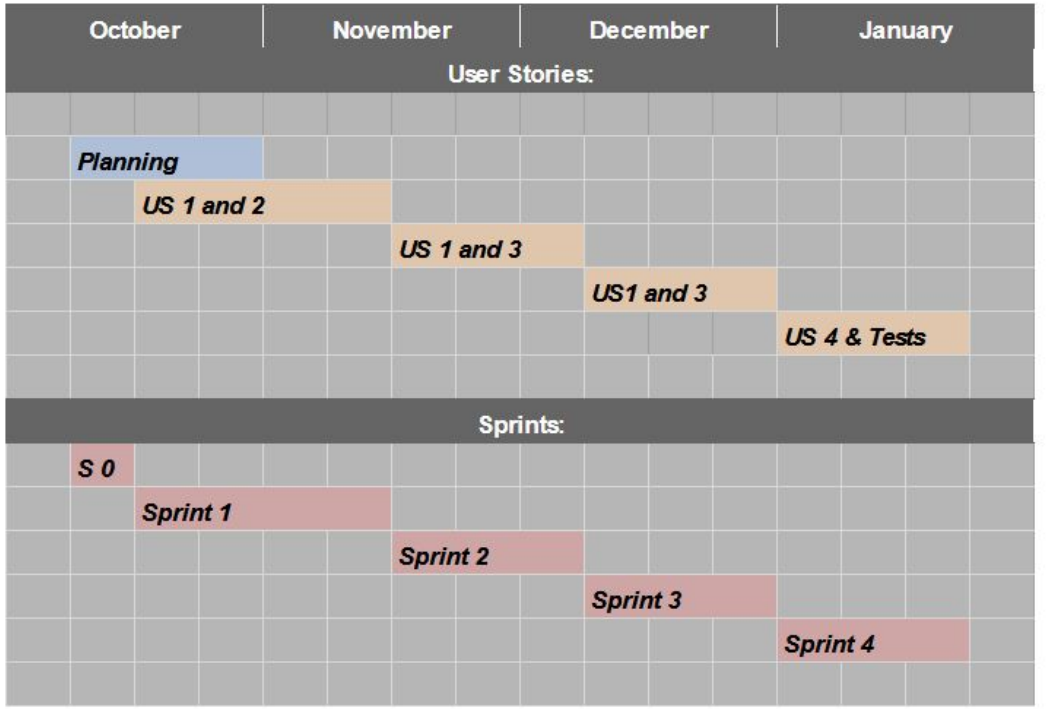
- Torque control for a given power wanted power.
- The relation is linear, and given by a gain and an offset from a voltage reference

- Might have some non-linearities because of the torque range of the crankset.

Summary of our project

Schedule





Evolution of our planning from our forecast to the final realization

* US = User Story

Speaker: Marie

Current State of our project

- **What has been done**
  - Functional website
  - Successful communication between the phone and the ESP
  - Proper identification of test bench
  - Non-lineare control law using Takagi Sugeno models

- **What remains to be done**
  - Find the solution to get the cardiac rhythm using a sensor
  - Eventually host the website on Maillon Mobility's domain
  - Connectivity to be established between ESP32 and the Flipsky/GenePi
  - Finalize the state observer and implement the control law in C++

Delay due to difficulties and shipping of components

Thank you

Do you have any questions?

Speaker: Marie