Self-driving-tank
- version 1.1

Michał Nizioł, Michał Zelek

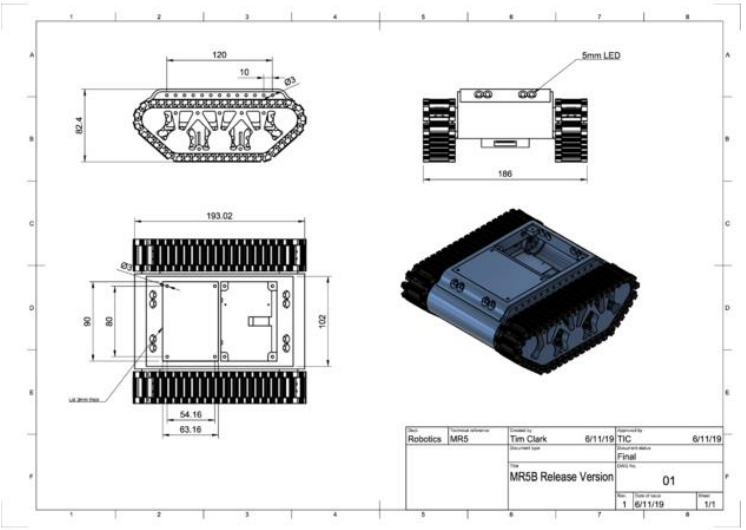
AGH

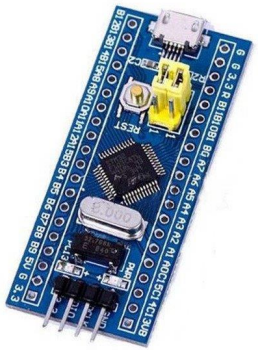
Project objectives

- Goal of a project is to create robot which moves on tracked chassis & is remotely controlled by website app (rest-api)
- Gathers acceleration data from accelerometer and calculates current net acc, velocity and position, which is displayed in real time on website
- Checks distances data from ultrasound and laser sensor which impacts the tank movement and behaviour
- Authorises user by RFID card and allows to use tank only with proper access

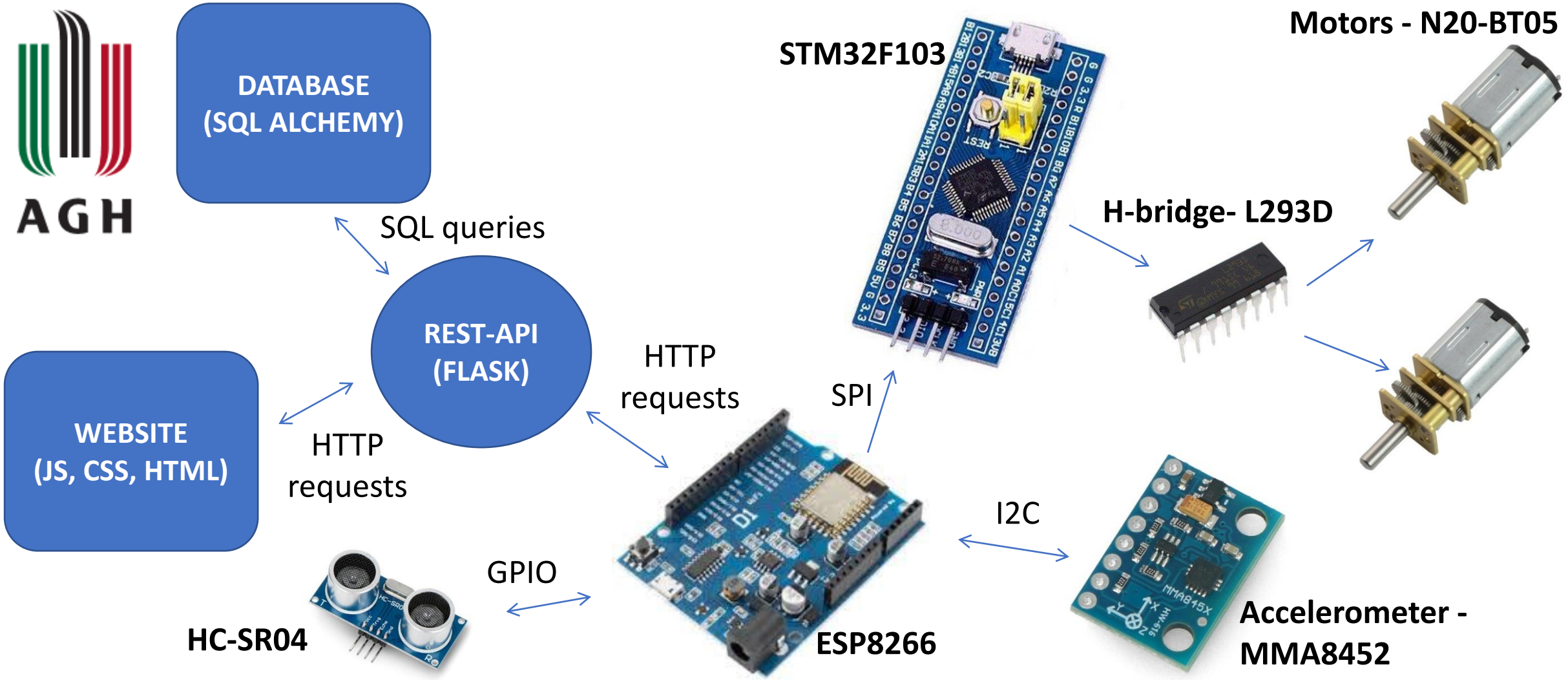
List of used elements

- STM32F103C8T6 ARM Cortex M3
- Accelerometer MMA8452
- ESP8266 with WiFi module
- Voltage stabilizer LM317
- H-bridge L293D
- 2x motors DC 12V N20-BT05 micro 50:1 625RPM
- Tracked chassis created in 3D printing
- Ultrasound sensor HC-SR04
- Laser sensor – TOF VL53L0X
- RFID MF RC522 module
- ESP32



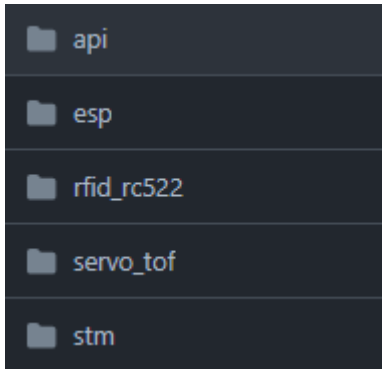


Block diagram



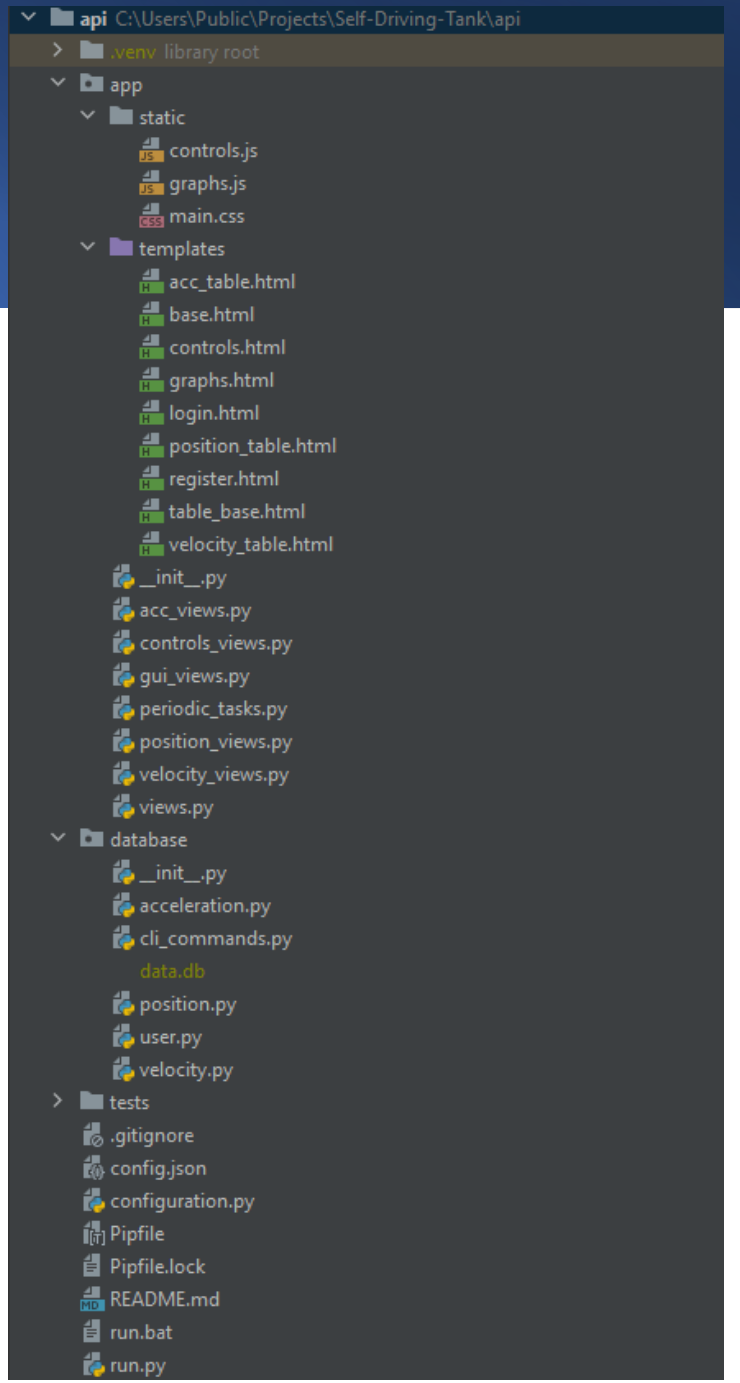
Software – project structure

- **api** -> contains Flask application (Python), which handles requests, collects data in database, allows user to control tank via website
- **esp** -> contains code for ESP-8266 (C++), which reads data from accelerometer and ultrasound sensor, makes POST & GET requests to api, and communicates with STM32
- **stm** -> contains code for STM32F103 (C) responsible for motors controls via H-bridge
- **rfid_rc522** & **servo_tof** -> contains code for ESP32 (C++) which controls servo, gaters data from laser sensor and handles RFID MF RC522 module

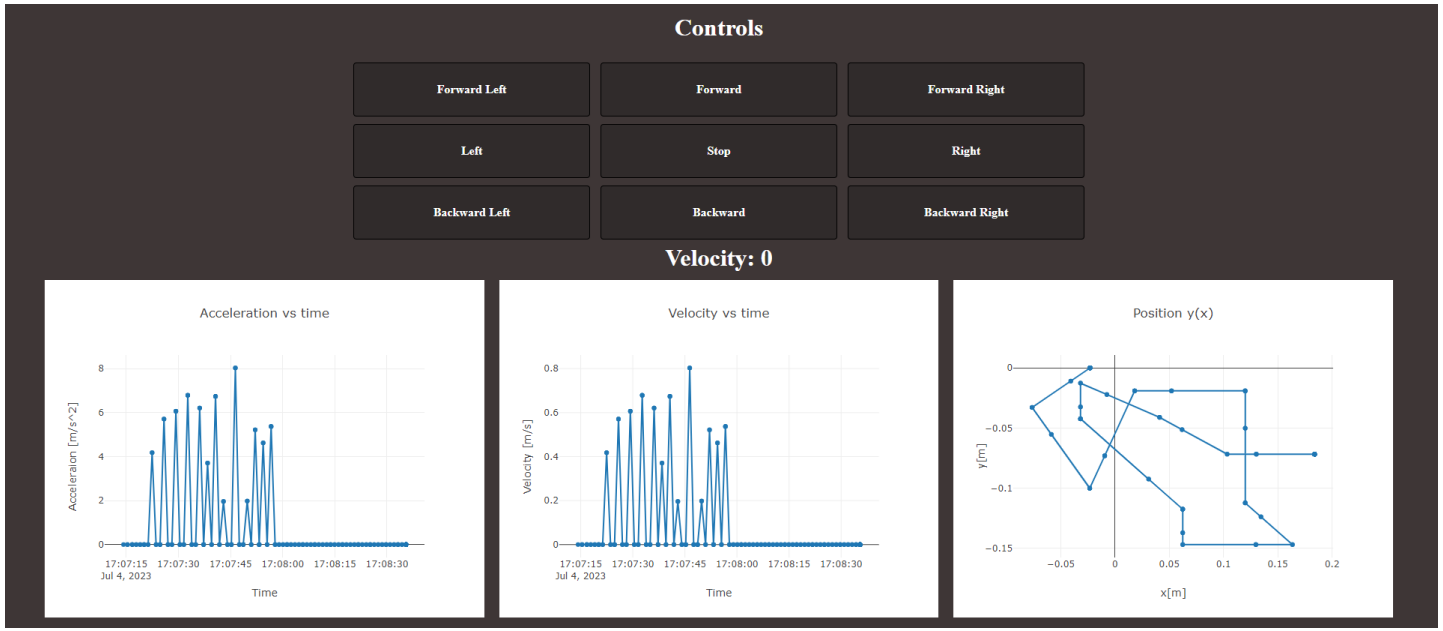


Software – api structure

- To setup api run venvSetup.bat which will install all nedded dependencies from Pipfile
- Service can be deployed by run.bat or server.py
- Dir database contains SQL alchemy object, models & schemas for data tables and database file (.db)
- Dir app contains Flask application with defined endpoints (acc, velocity, positions, controls and GUI views), templates files & APScheduler



View - /controls/

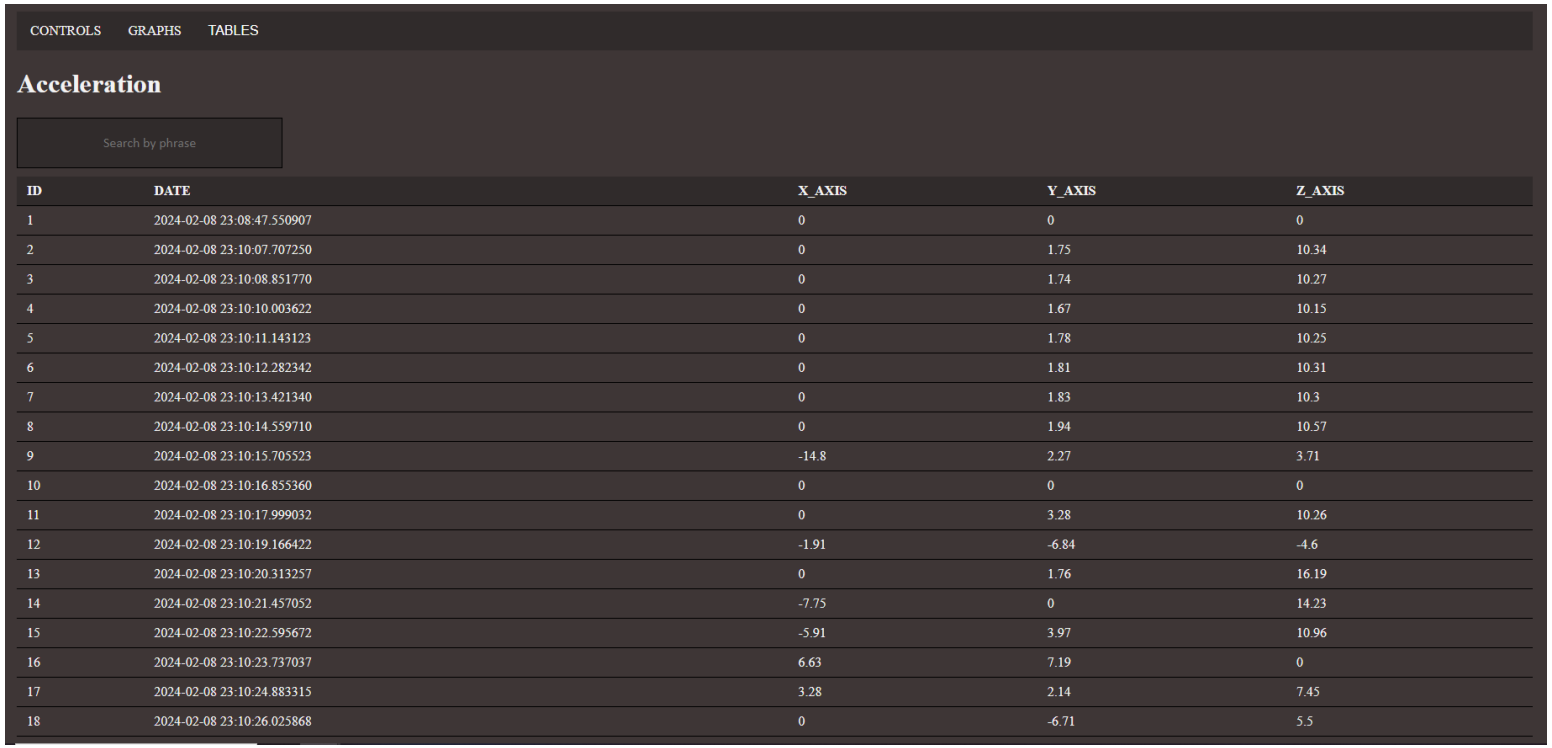


Tank can be steer by controls (buttons). In real time are updated graphs: net acc of time, net velocity of time, position y of x.

View - /graphs/

In view graph are displayed graphs of all gathered data.
(acc of time for all axis, postion of time for x i y and x of y.

View - /<data_name>_table/



CONTROLS GRAPHS TABLES

Acceleration

Search by phrase

ID	DATE	X_AXIS	Y_AXIS	Z_AXIS
1	2024-02-08 23:08:47.550907	0	0	0
2	2024-02-08 23:10:07.707250	0	1.75	10.34
3	2024-02-08 23:10:08.851770	0	1.74	10.27
4	2024-02-08 23:10:10.003622	0	1.67	10.15
5	2024-02-08 23:10:11.143123	0	1.78	10.25
6	2024-02-08 23:10:12.282342	0	1.81	10.31
7	2024-02-08 23:10:13.421340	0	1.83	10.3
8	2024-02-08 23:10:14.559710	0	1.94	10.57
9	2024-02-08 23:10:15.705523	-14.8	2.27	3.71
10	2024-02-08 23:10:16.855360	0	0	0
11	2024-02-08 23:10:17.999032	0	3.28	10.26
12	2024-02-08 23:10:19.166422	-1.91	-6.84	-4.6
13	2024-02-08 23:10:20.313257	0	1.76	16.19
14	2024-02-08 23:10:21.457052	-7.75	0	14.23
15	2024-02-08 23:10:22.595672	-5.91	3.97	10.96
16	2024-02-08 23:10:23.737037	6.63	7.19	0
17	2024-02-08 23:10:24.883315	3.28	2.14	7.45
18	2024-02-08 23:10:26.025868	0	-6.71	5.5

In this view can be seen all gathered data in database in useful table format (with sorting option, search filter etc.)