20-21 International Project

REPORT: SPRINT 3

Jyri Jacobson, Kalle Kaukola, Timi Partala, Arvo Cant & Casper De Keyser

Group 2

Table of contents

1 Overview

In this sprint we focused on making some progress on the separate parts of the project and then establishing a connection between the different components. It's important that we start working towards each other so we deliver a complete project consisting of the parts that we each developed.

2 Sprint review

2.1 Hardware

These are the goals we had in mind for this sprint:

- Get device prototype ready and working.
- Get data sending code ready and working.
- Measuring heartbeat with piezo.

These are the goals we were able to achieve:

- Measuring heartbeat with piezo has been achieved successfully and is working like a charm, we can filter the results with our microcontroller and can calculate heartbeat per minute from the data.

These are some problems that caused us to not reach our goals:

- We have example code for data sending in InfluxDB but different kind of issues with server has put us few steps back as we have not been able to configure influx settings on our server.
- PCB for the whole device has been ordered but will not be assembled before it is delivered sometime this week.
- ECG measurements still are not consistent and disturbance in readings is too much for making any kind of conclusions from the data.

These are some other issues we're having concering the hardware side of the project:

- We do not have enough know-how for assembling ECG-monitor with less disturbance.
- Microcontroller we have decided to use in this project does not have more than 1 analog input that makes it impossible to measure more than one analog sensor. Because of that we will be only measuring data to the database and it is from piezo as it has been more consistent.
- ECG device will be measured with Arduino Mega; it does not have internet connection, but it is able to measure and plot the data.

2.2 Software

2.2.1 Frontend

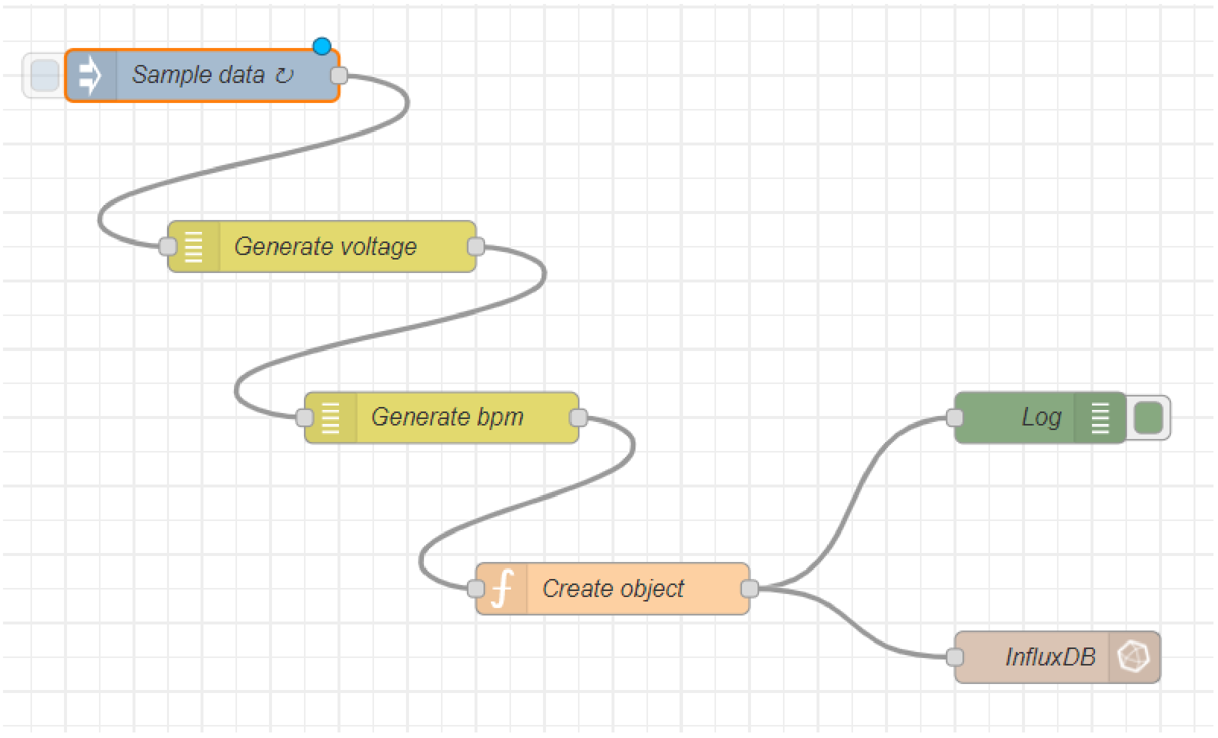
The code is making some progress, the Http side should work with dummy data however some solutions cause other problems. Our code is written in a stateful manner, a lot of parts of the Http code work stateless however. This is quite a big problem seeing as the two can't easily connect. The only real solution is to rewrite the early part of the frontend to be stateless but this will take some time and more understanding of the language, seeing as we don't have a lot of it left we need to take some actions quickly. The JSON data should in principle work however I don't have enough knowledge of the Dart language and thus need to learn more in the first place.

2.2.2 Backend

During this sprint I was able to finish the setup of the backend of our project. The backend now works completely on my Raspberry PI. Our goal is to set everything up on Jyri's system, so our project works on his setup locally. The backend now consists of the following components:

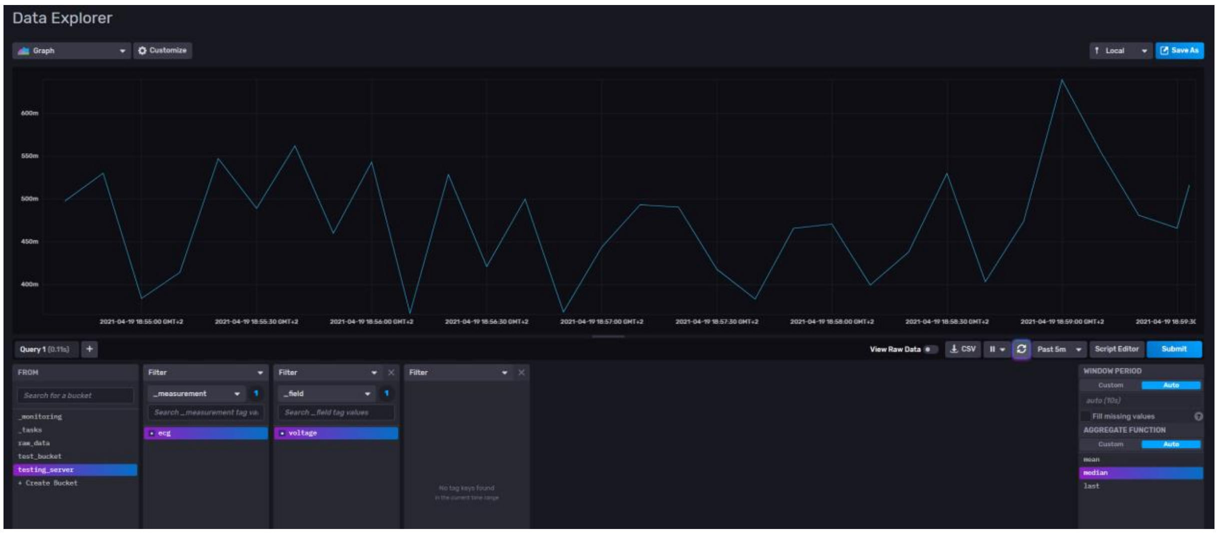
- **NodeRed**

This application is used to generate test data. This will not be used in the final version of the server, but it comes in handy for testing purposes. The "flow" creates an object with a timestamp, an ECG voltage (float) and a beats per minute number (integer). This object is then sent to the InfluxDB instance.

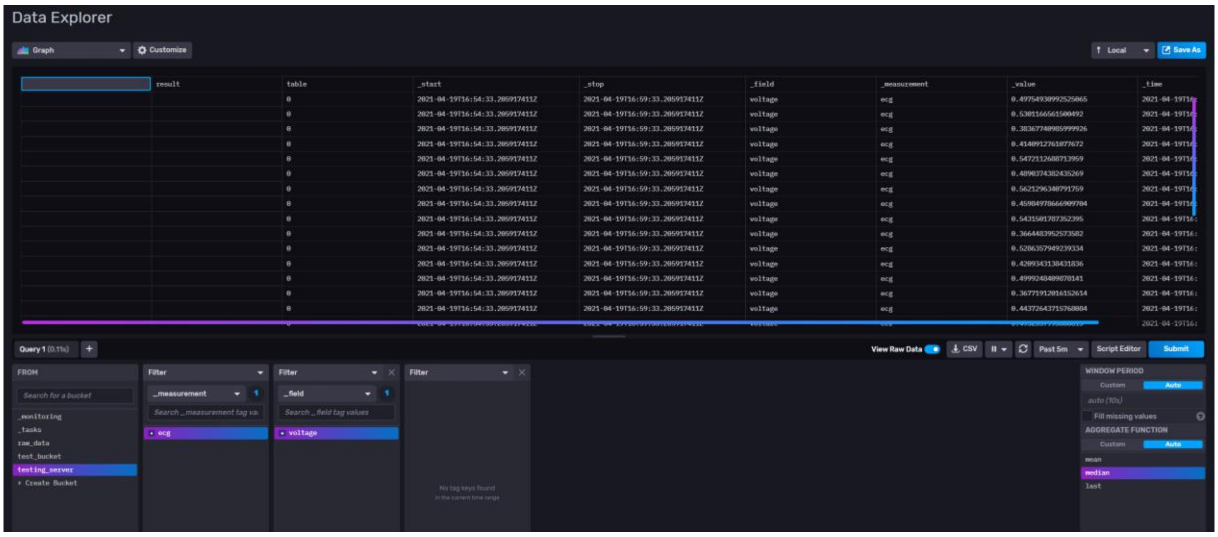


- **InfluxDB**

This database is time-series database and therefore optimized for storing IOT data. It fits perfectly for our project and also has some plugins for Arduino, for sending data. With some test data generated with NodeRed, the dashboard (an overview of the data) looks like this:



With the matching raw data:

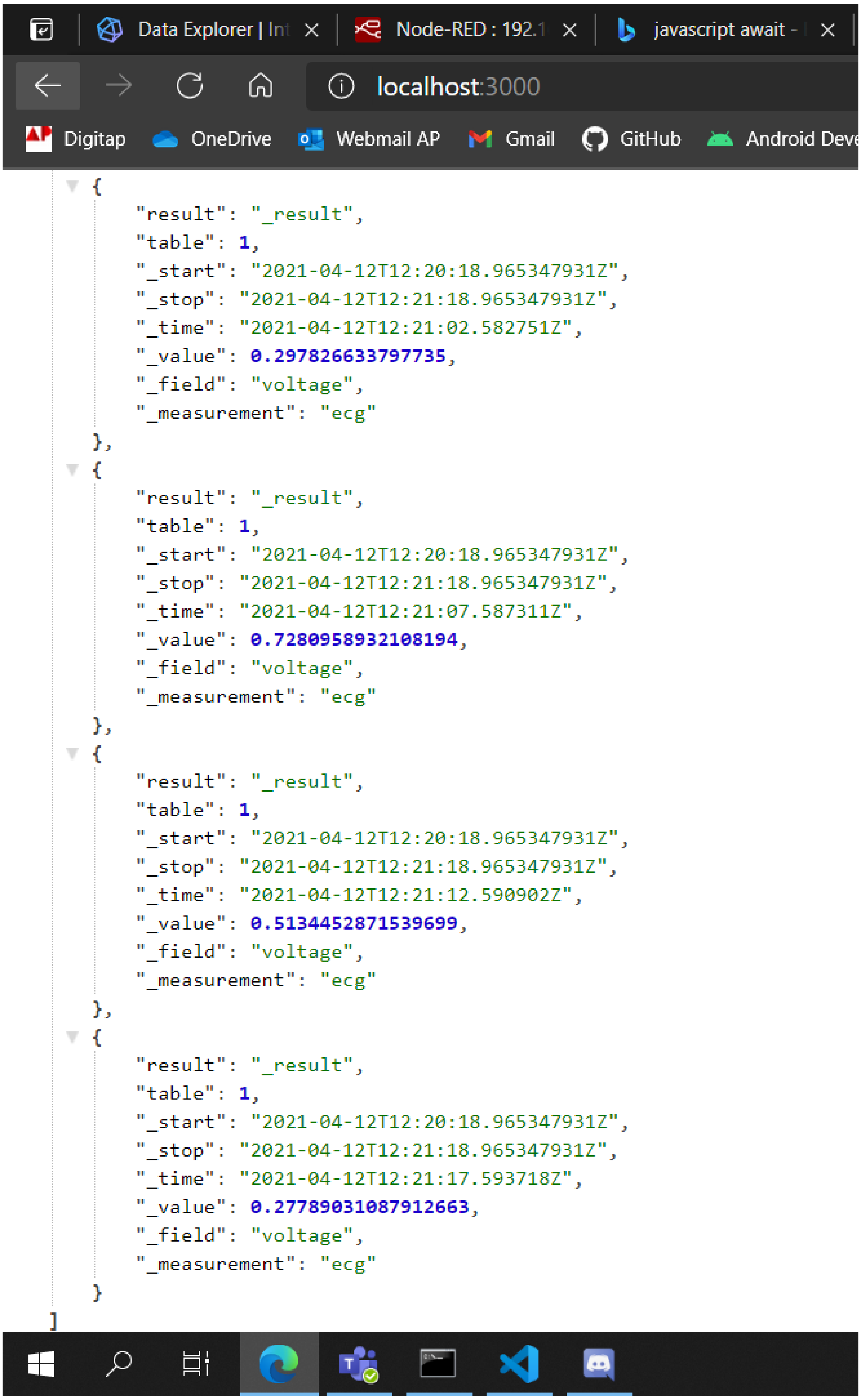


	result	table	_start	_stop	_field	_measurement	_value	_time
		0	2021-04-19T16:54:33.205917411Z	2021-04-19T16:59:33.205917411Z	voltage	ecg	0.49754930992525065	2021-04-19T16
		0	2021-04-19T16:54:33.205917411Z	2021-04-19T16:59:33.205917411Z	voltage	ecg	0.5301166561500492	2021-04-19T16
		0	2021-04-19T16:54:33.205917411Z	2021-04-19T16:59:33.205917411Z	voltage	ecg	0.38367748985999926	2021-04-19T16
		0	2021-04-19T16:54:33.205917411Z	2021-04-19T16:59:33.205917411Z	voltage	ecg	0.4140912761077672	2021-04-19T16
		0	2021-04-19T16:54:33.205917411Z	2021-04-19T16:59:33.205917411Z	voltage	ecg	0.5472112688713959	2021-04-19T16
		0	2021-04-19T16:54:33.205917411Z	2021-04-19T16:59:33.205917411Z	voltage	ecg	0.4090374382435269	2021-04-19T16
		0	2021-04-19T16:54:33.205917411Z	2021-04-19T16:59:33.205917411Z	voltage	ecg	0.5621296340791759	2021-04-19T16
		0	2021-04-19T16:54:33.205917411Z	2021-04-19T16:59:33.205917411Z	voltage	ecg	0.45984978666909704	2021-04-19T16
		0	2021-04-19T16:54:33.205917411Z	2021-04-19T16:59:33.205917411Z	voltage	ecg	0.5431501787352395	2021-04-19T16:
		0	2021-04-19T16:54:33.205917411Z	2021-04-19T16:59:33.205917411Z	voltage	ecg	0.3664483952573582	2021-04-19T16:
		0	2021-04-19T16:54:33.205917411Z	2021-04-19T16:59:33.205917411Z	voltage	ecg	0.5286357949239334	2021-04-19T16:
		0	2021-04-19T16:54:33.205917411Z	2021-04-19T16:59:33.205917411Z	voltage	ecg	0.4209343138431836	2021-04-19T16:
		0	2021-04-19T16:54:33.205917411Z	2021-04-19T16:59:33.205917411Z	voltage	ecg	0.4999248489078141	2021-04-19T16:
		0	2021-04-19T16:54:33.205917411Z	2021-04-19T16:59:33.205917411Z	voltage	ecg	0.36771912016152614	2021-04-19T16:
		0	2021-04-19T16:54:33.205917411Z	2021-04-19T16:59:33.205917411Z	voltage	ecg	0.44372643715768084	2021-04-19T16:

- **NodeJS**:

We use NodeJS to add webserver functionality to the Raspberry Pi. This way, we can do http-request from our app to the NodeJS-server. I added multiple routes so there's a possibility to query for different time periods.

Following screenshots show a query that has been sent to the InfluxDB and show in JSON format on the webserver:

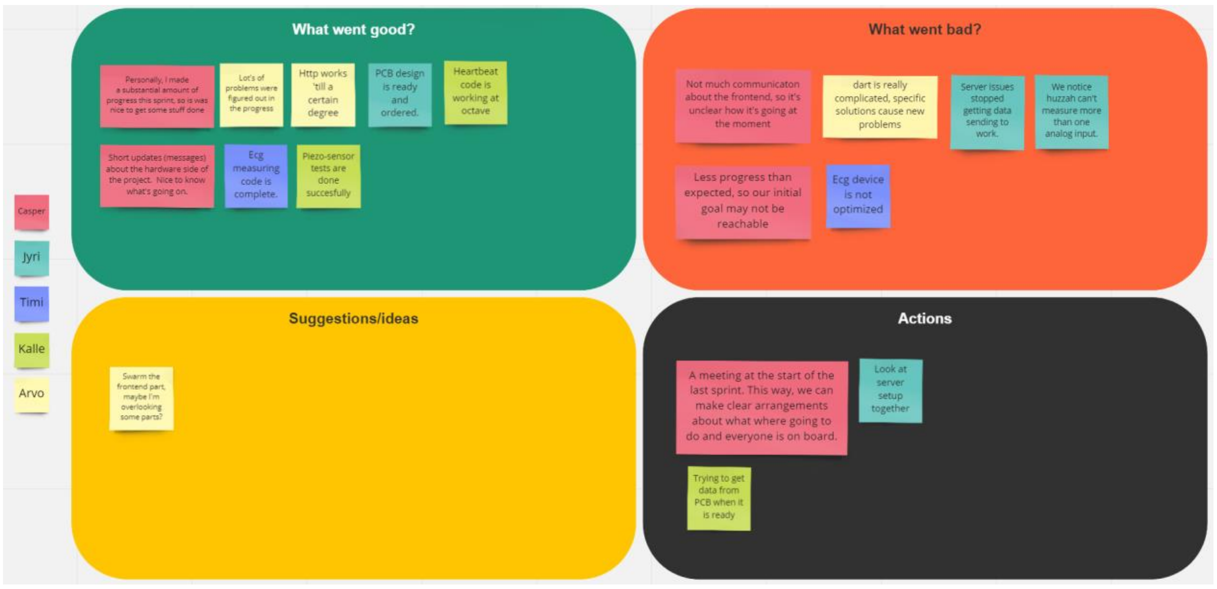


```
  {
      "result": "_result",
      "table": 1,
      "_start": "2021-04-12T12:20:18.965347931Z",
      "_stop": "2021-04-12T12:21:18.965347931Z",
      "_time": "2021-04-12T12:21:02.582751Z",
      "_value": 0.297826633797735,
      "_field": "voltage",
      "_measurement": "ecg"
  },
  {
      "result": "_result",
      "table": 1,
      "_start": "2021-04-12T12:20:18.965347931Z",
      "_stop": "2021-04-12T12:21:18.965347931Z",
      "_time": "2021-04-12T12:21:07.587311Z",
      "_value": 0.7280958932108194,
      "_field": "voltage",
      "_measurement": "ecg"
  },
  {
      "result": "_result",
      "table": 1,
      "_start": "2021-04-12T12:20:18.965347931Z",
      "_stop": "2021-04-12T12:21:18.965347931Z",
      "_time": "2021-04-12T12:21:12.590902Z",
      "_value": 0.5134452871539699,
      "_field": "voltage",
      "_measurement": "ecg"
  },
  {
      "result": "_result",
      "table": 1,
      "_start": "2021-04-12T12:20:18.965347931Z",
      "_stop": "2021-04-12T12:21:18.965347931Z",
      "_time": "2021-04-12T12:21:17.593718Z",
      "_value": 0.27789031087912663,
      "_field": "voltage",
      "_measurement": "ecg"
  }
]
```

3 Sprint retrospective

The second part of this report contains our retrospective sprint. Similar to last retro, we used a tool called Miro where you can setup a board and everyone can fill in what they would like to say.

In the screenshot below you can see our board. If the text is too small, be sure to go this our repo where we've also included an image of this board. (see docs/sprint3)



Casper
Jyri
Timi
Kalle
Arvo

What went good?

Personally, I made a substantial amount of progress this sprint, so is was nice to get some stuff done

Lot's of problems were figured out in the progress

Http works 'till a certain degree

PCB design is ready and ordered.

Heartbeat code is working at octave

Short updates (messages) about the hardware side of the project. Nice to know what's going on.

Ecg measuring code is complete.

Piezo-sensor tests are done succesfully

What went bad?

Not much communicaton about the frontend, so it's unclear how it's going at the moment

dart is really complicated, specific solutions cause new problems

Server issues stopped getting data sending to work.

We notice huzzah can't measure more than one analog input.

Less progress than expected, so our initial goal may not be reachable

Ecg device is not optimized

Suggestions/ideas

Swarm the frontend part, maybe I'm overlooking some parts?

Actions

A meeting at the start of the last sprint. This way, we can make clear arrangements about what where going to do and everyone is on board.

Look at server setup together

Trying to get data from PCB when it is ready