Group B | Fontys

Zet hier een project afbleeding

Sensor reader server

Air quality project

# Changelog

|  |  |  |
| --- | --- | --- |
| **Who** | **What** | **When** |
| Bart van de Loo | First draft | 31-03-2022 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table of contents

[Changelog 1](#_Toc87012286)

[Preface 2](#_Toc87012287)

[{hoofdvraag} 2](#_Toc87012288)

[{deelvraag 1} 2](#_Toc87012289)

[{deelvraag 2} 2](#_Toc87012290)

[{deelvraag 3} 2](#_Toc87012291)

[Conclusion research 2](#_Toc87012292)

[Design of project 2](#_Toc87012293)

[Diagrams 2](#_Toc87012294)

[Implementation 2](#_Toc87012295)

[Retrospective 2](#_Toc87012296)

[Bibliography 3](#_Toc87012297)

# Preface

The goal of this document is to describe the process of researching and developing a solution for the question:

“How does one create a software solution to receive and log data from an IoT sensor array that uses the MyriaNed6 as communication protocol?”

A small and robust system is needed to analyse and report on the air quality in small to medium enclosed spaces. To achieve this a IoT system was chosen for its flexibility and relatively simple installation.

# Approach

To achieve a viable solution a couple of things must be researched. The first thing to look into is what the MyriaNed6 protocol is and how it works. The second thing that must be researched is what kind of data we will receive from the Aemon sensor array, and how this can be translated into useful data. Lastly a suitable database solution must be found to store the data.

After this research is done a prototype will be developed.

## What is MyriaNed6

Myrianed is a wireless sensor network technology developed by a company called DevLab (<http://www.devlab.nl/>). The technology was inspired by the human way of communicating known as ‘gossiping’. A message sent by one of the nodes gets sent to all neighboring nodes. Then the neighboring nodes will relay or ‘gossip’ the messages to other nodes, this process repeats for a while until the message is seen as old or outdated.

By doing this the message spreads to all nodes in the network. This makes for a robust protocol since; First of all the devices don’t have to know how many devices or which devices are in the network to relay messages; And second of all the network is capable of using communication routes that are undefined, so even if one node fails to function the chances are big the message will just take another route to spread trough the network.

The network can use a couple of nodes to relay a message. One of these nodes is imaged below on the right. On the left there’s an image of the Aemon bridge, this houses our sensors

Afbeelding met tekst, binnen

Automatisch gegenereerde beschrijvingAfbeelding met binnen

Automatisch gegenereerde beschrijving

## What does our data look like?

The data consists of a big string of data. Most of the message is in a hexadecimal format. According to the documentation about the sensor node provided to me by Teade (which in turn was provided by Van Mierlo), the data is set up as follows:

The interface is text-based. Each line holds data from all valid sensors of a wireless node. Sample data lines are shown below.

*D15RR 0000 295a 26 3c5e327e 25 40ff740c 24 3f2621d8 1a 443afe9b cb*

*D 15RM 2a08 2a06 26 3f72bd4e 25 3f438c79 24 41f38b80 1a 444cec8d 67*

*D 15RR 2a4c 2a4b 26 3c5f30e9 25 40ff5450 24 3f246158 1a 44394752 45*

*D 15RM 2af9 2af7 26 3f80a33a 25 3f3a5ac5 24 41f1e36a 1a 4453590c e4*

Each line has the following structure

* The first character is always (normally always D, others can be ignored)
* The sensor id (15RM and 15RR in this case).
* Two timestamps. This enables a receiver to determine the exact time of measurement, independent from the flight time of the message in the network. These figures are two 16-bit hex numbers. To calculate the "exact" time of measurement, each message is sent with two time stamps (16-bit hex numbers). The first one is the actual network time. The second one is the network time at the moment of the measurement. The increments are 500ms tics. At the end a rollover will occur from 0xffff to 0x0001. The value 0x0000 is invalid and indicates that the network time is not (yet) defined.
* The sensor data. The sensor data starts with an 8-bit hex number defining the sensor type. The following number is a 32 bit float value with the actual sensor data. The following sensors are available:
  + TEMPERATURE = 0x01, C
  + HUMIDITY = 0x02, 0…100%
  + LIGHT\_INTENSITY = 0x1a, lux
  + NH3\_PPM = 0x24, ppm
  + NO2\_PPM = 0x25, ppm
  + CO\_PPM = 0x26, ppm
  + CO2\_PPM = 0x27, ppm
  + BAROMETRIC\_PRESSURE = 0x28, mbar
* Afbeelding met tekst

  Automatisch gegenereerde beschrijvingA crc8 (Cyclic redundancy check, a process to detect and correct bit errors) calculated over the data. To be able to check the transmission a CRC8 is calculated over the complete data line. The calculation starts with the character 'D' and ends with the whitespace before the two CRC characters. The crc8 calculation uses 0xA5 as an initial value and 0x07 as polynomial. There is no additional data operation on the output.

Tekening van de gehele data packet, opgedeeld in de verschillende data onderdelen

## How will the database be set up?

## Conclusion research

*(conclusie trekken op basis van research om bepaalde design principes te verantwoorden vederop in het bestand)*

# Design of project

The project will has been designed to incorporate the following design principles

## Diagrams

*(alle gebruikte diagrammen invoegen, elk met een eigen kopje en een kleine beschrijving)*

# Implementation

*(code snippets toevoegen van gecompliceerdere onderdelen, video’s/foto’s van werkend project toevoegen)*

## Test report

*(aantal tests schrijven om het gedrag van het project te valideren, en deze tests ook uitvoeren)*

# Retrospective

*(Terug kijk op gemaakt project, wat kan beter? wat ging er goed?)*

# Bibliography