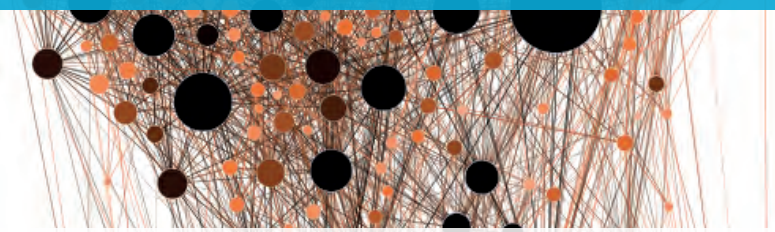
DynamIoT



DID-C1 Technical Report

Using a dynamic sensor network to obtain spatiotemporal data in an urban environment

L. Angelova, P. Flikweert, P. Karydakis, D. Kersbergen, R. Teeuwen, K. Valečkaitė
Supervisors: ir. E. Verbree, dr. ir. M. Meijers

TU Delft

DynamIoT

[This page has been left blank intentionally]

DynamIoT

Using a dynamic sensor network to obtain spatiotemporal data in an urban environment

DID-C1 Technical Report

By

Angelova, L., Flikweert, P., Karydakis, P., Kersbergen, D., Teeuwen, R., Valečkaitė, K.

In partial fulfilment of the requirements for the degree of

Master of Science
in Geomatics

at the Delft University of Technology,
to be presented publicly on June 23, 2017

Supervisors: ir. E. Verbree, dr. ir. M. Meijers

[This page has been left blank intentionally]

TU Delft

Preface

This research project was performed as part of the Synthesis Project within the Master's Programme on Geomatics for the Built Environment at the Technical University of Delft. This year, five different Synthesis Project research subjects were offered to the students. The establishment of our project team was therefore done based on our subject preference: our interest into researching the possibilities of a Dynamic Internet of Things network. Within this subject, we were free to choose our own focus. Therefore we can say that we were as a group intrinsically motivated to make the absolute best of this project and to develop our skills and knowledge during it. It is with full satisfaction that we now hand in this report after ten weeks of research: we are confident that this report is a suitable conclusion to our first year as students at the TU Delft Geomatics Programme.

In the process of making this project successful, acknowledgements are obliged to:

Edward Verbree, Martijn Meijers and Stefan van der Spek, for sharing their knowledge and experience, for asking the right questions and for their interest in our findings.

Bastiaan van Loenen and Lorenzo Dalla Corte, for answering our questions regarding privacy issues.

Rob Braggaar, Teun Verkerk, Aidan Wyber, and Marijn Tiggelman, for their practical support as far as sensors and code are concerned.

Bas van Goor from Sweco and the IoT Academy, for their knowledge about trilateration and LoRa.

The Delft City Shuttle and Rondvaart Delft, for their enthusiasm and willingness to make our sensors into dynamic ones.

The Delft Municipality, for funding the research as part of the project '*technology in the city*'.

To conclude, of course, each other, for the weeks of fruitful cooperation and for experiencing many ups and downs as a team, working ourselves to a result to be proud of.

Liliya Angelova
Puck Flikweert
Panagiotis Karydakis
Daniël Kersbergen
Roos Teeuwen
Kotryna Valečkaitė

Delft University of Technology
Delft, June 2017

Contents

[This page has been left blank intentionally]

List of Figures

List of tables

List of acronyms and abbreviations

AES	Advanced Encryption Standard
API	Application Programming Interface
AU	Arbitrary Unit
BLE	Bluetooth Low Energy
°C	Celsius
COP	Computer Operating Properly
CSI	Channel State Information
CRS	Coordinate Reference System
CSS	Chirp Spread Spectrum
DBMS	Database Management System
DPA	Data Protection Agency
GDPR	General Data Protection Regulation (95/46/EC and EU 2016/679)
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IEEE	Institute of Electrical and Electronics Engineers
INS	Inertial Navigation System
IoT	Internet of Things
IP	Internet Protocol
LoRa ®	Long Range Radio
LoRaWAN	Long Range Radio Wide Area Network
LoS	Line of Sight
LPWA	Low Power Wide Area
LSB	Least Significant Bit
MAC	Media Access Control
MCU	Microcontroller Unit
MoSCoW	Must – Should – Could – Would prioritization structure
MSB	Most Significant Bit
NPM	Node Package Manager
OGC	Open Geospatial Consortium
RSS	Radio Signal Strength
RSSI	Received Signal Strength Indicator
SF	Spreading Factor
SSID	Service Set Identifier
SWE	Sensor Web Enablement
ToA	Time of Arrival
TDoA	Time Difference of Arrival
TTFF	Time To First Fix
VM	Virtual Machine
WAP	Wireless Access Point
WBP	Wet Bescherming Persoonsgegevens
WLAN	Wireless Local Area Network
σ	Standard Deviation

[This page has been left blank intentionally]

Abstract

Along with the rise of the smart city movement, Internet of Things is an upcoming phenomenon. Objects and devices are becoming more and more wirelessly interconnected, communicating information between themselves and to human beings. As an extension on static sensor networks that gather real-time environmental data, the feasibility of implementing a dynamic sensor network based on LoRa communication is researched. To achieve such a dynamic system, a self-developed sensor platform was constructed, based on the microcontroller LoPy. Sensors attached to it include a hygrometer, thermometer and microphone.

The emphasis of the research was on localisation of the sensors, to put the gathered sensor data into geographical context. A WiFi fingerprinting radiomap was constructed based on available MAC-addresses, their signal strengths, and GPS coordinates. The GPS module was only used for composing the radiomap. When the radiomap is completed, the module can be switched off, only to be switched on for periodical updates of the radiomap. The quality of the radiomap methodology was evaluated by constructing it of measurements gathered in four days, and testing it for the remaining three days. This test gave a correctness of 50% while another 38% of measurements were localised in a neighbouring cell. The correctness can be improved by having a longer training period.

The quality of the collected sensor data turned out to be dependent on the weather conditions and the placement location on the carrier vehicle. Vehicle requirements were specified as driving through the city centre and having a schedule and route producing as little noise, heat and air pollution as possible. Another topic of research was LoRa communication, which was deemed as very limited for dynamic implementations, as the sending of location-related data takes up a large part of the already limited message size. To decrypt the sent message and store it in a meaningful database, Node-RED was used. Despite visualisation of measurements showed promising results, there is margin for improvement as far as data capturing is concerned.

Executive Summary

In this executive summary, the most important technical details of the Dynamic IoT research project are brought together. The summary is provided in the form of a research paper following a similar outline of contents as the report. It can be found from the next page onwards.

USING A DYNAMIC SENSOR NETWORK TO OBTAIN SPATIOTEMPORAL DATA IN AN URBAN ENVIRONMENT

Angelova, L., Flikweert, P., Karydakis, P., Kersbergen, D., Teeuwen, R., Valečkaitė, K.

Supervisors: Verbree, E., Meijers, M., van der Spek, S.

KEYWORDS: Dynamic Sensor Network, Internet of Things, WiFi Fingerprinting, Localisation, Sensor Data

ABSTRACT: Along with the rise of the smart city movement, Internet of Things is an upcoming phenomenon. Objects and devices are becoming more and more wirelessly interconnected, communicating information between themselves and to human beings. As an extension on static sensor networks that gather real-time environmental data, the feasibility of implementing a dynamic sensor network based on LoRa communication is researched. To achieve such a dynamic system, a self-developed sensor platform was constructed, based on the microcontroller LoPy. Sensors attached to it include a hygrometer, thermometer and microphone. The emphasis of the research was on localisation of the sensors, to put the gathered sensor data into geographical context. A WiFi fingerprinting radiomap was constructed based on available MAC-addresses, their signal strengths, and GPS coordinates. The GPS module was only used for composing the radiomap. When the radiomap is completed, the module can be switched off, only to be switched on for periodical updates of the radiomap. The quality of the radiomap methodology was evaluated by constructing it of measurements gathered in four days, and testing it for the remaining three days. This test gave a correctness of 50% while another 38% of measurements were localised in a neighbouring cell. The correctness can be improved by having a longer training period. The quality of the collected sensor data turned out to be dependent on the weather conditions and the placement location on the carrier vehicle. Vehicle requirements were specified as driving through the city centre and having a schedule and route producing as little noise, heat and air pollution as possible. Another topic of research was LoRa communication, which was deemed as very limited for dynamic implementations, as the sending of location-related data takes up a large part of the already limited message size. To decrypt the sent message and store it in a meaningful database, Node-RED was used. Despite visualisation of measurements showed promising results, there is margin for improvement as far as data capturing is concerned.

INTRODUCTION

This research is performed as a part of the TU Delft Master's programme in Geomatics. The subject of the research is a dynamic Internet of Things (IoT) network. In parallel, complementary research is performed on a static IoT network by fellow students.

With the rise of the world's population and the intensive urban growth, the need for Smart Cities rises. According to Caragliu et al. (2011), a city is smart "when investments in human and social capital and traditional (transport) and modern (ICT) communication infrastructure fuel sustainable economic growth and a high quality of life, with a wise management of natural resources, through participatory governance". However, these implementations require investments. On the other hand, the costs of implementing electronic systems have recently decreased. On the other hand, questions are to be raised on the data quality, skills needed and minimum requirements for the application of these systems.

The concept of sensor networks has already been researched multiple times over the last decades (Khedo, Perseedoss & Mungur, 2010; Szewczyk et al., 2004; Yick, Mukherjee & Ghosal, 2008). If a sensor network is made dynamic by mounting sensors on moving vehicles, it has beneficial effects: a larger area can be covered by a moving sensor, as opposed to a static one, and therefore fewer sensors are needed (Larson et al., 2017).

The goal of this research is stated to be: "Localisation of the transmitters, providing insight in the location and conditions of the measurements and provide an interactive data platform for empowering citizens: households, visitors and shopkeepers." (van der Spek, 2017, p6).

To this goal, the following research question is defined:

> *To what extent can near real-time spatiotemporal data be obtained using a dynamic sensor network based on LoRa communication in an outdoor urban environment?*

By 'obtain' data collection, storage and access is meant.

The temporal scope of this research is ten weeks in spring 2017. The geographical scope is the extent of the city centre of Delft, the Netherlands.

The expectation of the project team is that it is possible to obtain near real-time spatiotemporal data using a dynamic sensor network based on LoRa communication in an urban environment, although the level of location granularity and localisation correctness is expected to require further improvement in order of relevant spatiotemporal data to be obtained. However, it is expected that it does provide a usable proof of concept for further research.

This article starts with theoretical background on relevant localisation techniques. Afterwards, a practical background will be provided by describing technical implementation details and benchmark tests. Then, the methodology followed is described and results are presented. In the end, the main question is answered in the conclusion and a research discussion and future recommendations are provided.

1. LOCALISATION TECHNIQUES

For this research, differentiation is made between localisation and positioning: localisation is concerned with context or meaning, in contrast to positioning, which only refers to specific coordinates (Mautz, 2012).

1.1 Localisation requirements

The method regarding the localisation of the moving sensor platforms has numerous requirements. Firstly, the technique should be operable within the temporal scope of ten weeks. Also, the method should be applicable for an urban environment and over an extent of at least one square kilometre. Moreover, the range of the communication used in the localisation technique should be sufficient for the geographical extent as well. The localisation granularity and correctness should be such that usable data can be obtained. For this, reliable localisation on street level is used as a guideline. In addition, it should be stated that the application does not require the sensor platforms to be aware of their own location, but only the server should.

1.2 Localisation alternatives

To find the localisation technique of best use, seven alternatives were evaluated. In table 1, their advantages and disadvantages are stated.

From this table it can be concluded that no alternative is perfect according to the application requirements. However, some are more apt than others. The energy consumption involved with the use of Global Navigation Satellite System (GNSS) limits the capabilities of the sensor platform or other implementations for the Internet of Things. WiFi and Bluetooth Low Energy (BLE) fingerprinting are a proven success, but they do come with the downside of continuous maintenance or the use of non-transparent Application Programming Interfaces (API) provided by third parties. Trilateration based on Received Signal Strength Indicator (RSSI) measurements from BLE and WiFi is heavily influenced by the existence of multipath and would require an implementation with an extensive new network. An Inertial Navigation System (INS) is not sufficient for localisation without the use of extra localisation methods. Long Range Radio (LoRa) is a new communication technique which has possible capabilities to provide periodic position information. This research was continued to explore the possibilities of WiFi fingerprinting, WiFi trilateration and LoRa positioning, since these techniques provide the best opportunities while using the least amount of resources. To do so, range tests and calibration tests are performed in part 2 and 3.

Technique	Advantages	Disadvantages
GNSS	Proven accuracy, no new infrastructure needed, local localisation possible	Expensive chip, energy consumption, affected by environment, extra hardware needed, dependence on owners, extra communication needed, multipath
WiFi Fingerprinting	Low cost implementation, available APIs and methodologies	Dependency, depreciation of signal map, multipath interference, offline training period, extra communication needed
BLE Fingerprinting	Low energy consumption, low cost implementation	New infrastructure needed, depreciation of signal map, offline training period, CSI recommended, extra communication needed
WiFi Trilateration	Low cost implementation, no offline training period	New infrastructure needed, fixed node location needs to be known, affected by environment
BLE Trilateration	Low cost implementation, no offline training period	New infrastructure needed, fixed node location needs to be known, affected by environment
INS	Relatively accurate nearby, cheap components	Drift, extra hardware needed, fusion of data necessary
LoRa	Low cost implementation, extreme range	Physical accuracy limitations, limited communication, little research available.

Table 1: advantages and disadvantages of localisation technique alternatives (own work)

2. TECHNICAL IMPLEMENTATION

During the research, several technical implementation details were considered.

2.1 Vehicles for dynamic network

Firstly, deploying a dynamic sensor network means that sensors should be attached to moving objects. Several vehicle alternatives were evaluated which are already continuously operating in the city of Delft. One advantage is that they have a minimal amount of environmental emissions, as measurements will be done on temperature, air humidity and noise levels. The vehicles should therefore influence these measurements as little as possible. It is regarded advantageous when the vehicle is operating on a fixed route: this will enlarge the measurement density along these routes. The higher density and more regular the measuring is, the better conclusions can be drawn. Two vehicle types amongst the alternatives were found to fulfil

these requirements: the electrical tourist Tuk Tuks and boats, operating daily on fixed routes through the city centre of Delft. Consent to attach sensor platforms during the measurement period was obtained from both companies.

2.2 Sensor platforms

In addition, the sensor platforms themselves bring implementation constraints and requirements. In broad lines, the platform can be separated into two parts; electronics and casing. The electronic elements selected for the final sensor platform include a microphone (MAX9814), temperature and humidity sensor (AM2302), GPS module (u-blox NEO-6M), two antennas (LoRa and WiFi 8dBi), and a battery of 5000mAh, which is charged with a USB-based circuit of 500mA (ADA-1305). These modules, apart from the charging circuit, were assembled on a perfboard. The system with LoPy deep sleep energy saving code can measure for two days, and with an additional GPS on/off mode, for three days. If the GPS module was not used in the setup, the battery life could be extended to over ten days. One of the most important restraints regarding electronics part of the platform is associated with the electronics and outer-air contact. The humidity, temperature and noise sensors have to be in contact with outside air, whilst still remaining protected from water. On the other hand, the rest of the components do not require the contact with outside air, while protection from the elements is just as important. Therefore, it was chosen to separate the sensor platform into two separate casing compartments: a closed compartment and a ventilated compartment.

Regarding the casing, a solution was chosen involving a pair of connected water-tight food boxes, complemented with 3D printed elements. The latter involves ventilation covers for the openings of the ventilated compartments for airflow. 3D printed pins were added to balance the electronics components and reduce the stress on the solder connections.

2.3 LoRa restrictions

Since in this research the capabilities of LoRa communication are analysed, the LoRa protocol implementation details are of interest. In the Netherlands, LoRa communication is possible via the network of telecom provider KPN. However, a maximum payload size of 54 bytes per message is set. Messages are sent over the unlicensed 868 MHz band, to which the government regulation is enforced that a device can only send messages for 1% of the time. The time on air of a message is defined by its payload combined with its spreading factor: a message is either long and less powerful (high spreading factor) or short and powerful (low spreading factor) dependent on its reception of available LoRa masts (Jol, n.d.). A LoRa message should be in hexadecimal format. The message is encrypted, sent to the KPN antenna with the best connection and from there it can be accessed by the owner of the sending device via the internet.

2.4 Storage in database

The last type of implementation details concerns the database in which the measurements are stored. By the TU Delft, a server was provided. A virtual machine was installed with PostgreSQL and PostGIS spatial data extension software. In addition, the programming tool Node-RED was installed, to allow wiring together different data flows. Within this virtual environment, a table was created with sufficient number of columns and tested with dummy data and later on with real data. Messages are sent over the LoRa network and can be imported by the database over internet as a JavaScript object. Then, the data can be retrieved from the database via a query. The database architecture allows for near real-time dashboards as well, although only limited use is made of those in this research.

As mentioned earlier, a LoRa message should be of hexadecimal format and is encrypted when sent. The encryption method is by default AES-128 bits encryption (Lora Decryption on Application Server, 2017). The choice was made not to let KPN decrypt the payload in the KPN ThingPark platform: the decryption was performed on the server itself, since device address and key are known. Decryption is performed per block of 16 bytes by using Node-RED.

3. BENCHMARK TESTS

For the system set-up to be designed optimally, tests regarding the performance of both electronic and casing solutions were carried out.

3.1 Localisation implementation

Three techniques for localisation are considered: WiFi fingerprinting, WiFi trilateration and LoRa Time Difference of Arrival (TDoA).

Within the LoPy microcontroller, Bluetooth, LoRa and WiFi communication technologies are integrated. It can be extended with external WiFi and LoRa antennas and a GPS module. Tests were performed both with and without external antenna.

The setup of the first test was to measure the WiFi RSSI between two LoPy devices every 10 metres for a range of 130 metres with direct Line of Sight (LoS). The RSSI values deviate from -5dB to -90dB. The minimum signal strength for reliable communication is considered to be -70db. At -80dB there is still a connection between the devices but the data may be not delivered. With the external antenna, considerable stronger WiFi connections are achieved: on average, a factor 10 times stronger with peaks to factor 100. A reliable connection with external antenna can be achieved at 120 metres distance, while only up to 70 metres without external antenna. The difference is significant when localisation in an urban environment and therefore under imperfect conditions has to be performed.

For the assessment of WiFi trilateration possibilities, linear least squares regression on the measurements done with external antenna was performed. A variance of over 20% was found, which cannot be explained by a linear approximation, even though direct LoS was present. Also, trilateration would require extra network components to provide coverage of the whole research area. Therefore, WiFi trilateration is not further considered for practical implementation in this research.

For localisation using LoRa, TDoA principles have been considered. This principle would limit the need for time synchronisation to the LoRa gateways only: avoiding precarious procedures on the sensor platforms. In a static experiment, the synchronization was assessed. It was found that the clocks of available LoRa gateways were not synchronized and the time accuracy provided was only milliseconds, which would correspond to kilometres inaccuracy when applying TDoA localisation. Also, very limited documentation and research is available on LoRa localisation. Therefore, LoRa localisation was considered not feasible within this research.

Since, as will be described in the methodology, Global Positioning System (GPS) data will be used in the localisation method, a static test in a nearly unobstructed environment on the LoPy GPS module accuracy was performed as well. GPS coordinates were collected from a known position and then converted to the metrical Dutch Amersfoort/RD New Coordinates Reference System (CRS) for postprocessing purposes. A standard deviation of GPS measurements was assumed. For each measurement, the Euclidean distance in x and y direction to the known location was determined. Based on these distances, the standard deviation σ and 2σ were defined, as shown in Table 2.

σx	**σy**	**σ**	**2σ**
2.0 m	4.9 m	5.3 m	10.6 m

Table 2: Standard deviation of GPS position (own work)

It can be assumed that 95.4% of the measurements fall in the range of 2σ i.e. 10.6 m, which is the accuracy that is taken into account for the rest of the research. This accuracy is considered sufficient to use GPS as described in the methodology and for quality control of localisation results, as the expected accuracy of the localisation technique to be described in the methodology is less.

3.2 LoRa communication

The performance of LoRa communication between two LoPy devices was researched using them as private Nano-gateways in a dynamic range test. When direct LoS was not available, the communication between the devices was proven to be unstable. The instability of communication was only the case using two LoPy devices. The communication between a LoPy device and the KPN gateways was found to be more stable, especially outdoors.

3.3 Sensor platform casing

Since concerns were raised about the way in which the casing of the sensor platform would affect the temperature and humidity sensor measurements, benchmark tests were performed. A sensor platform was modified to have sensors not only inside the casing but outside as well. Then, temperature and humidity measurements were taken every 20 seconds both dynamically and statically, and both in the sun and in the shade.

A greenhouse effect of the casing was not found in neither sunny nor shadow conditions for the dynamic measurements. However, the box was found to work like a low-pass filter, smoothing out humidity and temperature fluctuations. This filter is beneficial for the implementation of sending message only every three minutes, as elaborated on in the methodology, as it decreases the chance of outliers.

However, it is found that in a static sunny situation, temperature rises significantly both inside and outside the casing. This rise can be attributed to the absence of considerable airflow and the solar gain on the deployment location. To be able to determine the extent of the influence of the observed effect, further research is required.

4. METHODOLOGY

In this part, the methodology followed to reach the main goal and to answer the research question as defined in the introduction is elaborated on. The methodology is based on the theoretical and practical research described above.

4.1 Collection of data

As far as collection of data is concerned, eight sensor platforms have been developed. The platform, with the elements explained in the section 2.2, measured temperature, humidity and sound. Also, it collected Media Access Control (MAC) addresses of Wireless Access Points, their RSSI values and GPS positions.

The sensor platforms were attached to six Tuk Tuks and two boats. Data was gathered on seven consecutive days from Thursday June 8th to Wednesday June 14th, 2017. Data was only gathered during operating hours of the vehicles: starting 10:00h for the Tuk Tuks and 11:00h for the boats and ending at 18:00h.

To ensure efficient operation time, the data was gathered periodically: each three minutes a 30-byte hexadecimal message was sent containing MAC addresses, RSSI values to them, temperature, humidity and sound levels and GPS coordinates. To compress the size of the GPS coordinates on the LoPy before sending, a reference point was added to the map: a GPS coordinate is then expressed only in relation to this point. This reference point is only applicable for Delft area, and therefore only coordinates within Delft city centre are considered.

The collected data covers all seven days of the experiment, but was measured not with the full collection of the sensor platforms. It was attributed to the mechanical connector failures caused by carrier vehicle vibrations.

4.2 Processing of data

As stated above, a 30-byte message is sent by the sensor station each three minutes. Via LoRa, it is received in the database and is then decrypted. Based on the MAC address and RSSI values, the Google geolocation API is used to calculate coordinates to each measurement (Google, 2017). However, Google geolocation is not the chosen localisation technique: it is only included for quality control reasons later on.

Further processing is done for localisation purposes. To achieve localisation, a 40 by 40 cells grid of 50 by 50 metres per cell is fit over Delft city centre. This grid covers the whole area of interest, based on the vehicle routes. The cell size was chosen such that each cell is expected to be identifiable from another, the accuracy of the GPS fix is lower than the cell size but the granularity of data is still high enough to provide for useful data analysis.

4.3 Localisation of sensor data

Based on the theoretical and practical findings in part 1, 2 and 3, the choice was made to localise measurements using WiFi fingerprinting. To avoid the drawback of calibration of measurements (Mautz, 2012), the radiomap was chosen to be composed automatically by making use of GPS coordinates collected in the measurement period.

These GPS coordinates were converted to the Dutch CRS and the measurements belonging to them were assigned to a grid cell based on this coordinate. Then, a list of MAC addresses and corresponding RSSI values was composed per grid cell based on the measurements assigned to it. In case a MAC address is entered in the list multiple times, logarithmic averaging of the RSSI values is applied. The radiomap consists of such an MAC-RSSI list per grid cell.

After having composed the radiomap, the localisation can be performed: new measurements can be matched to a location in the radiomap. The concept of Euclidean Distance in signal space is used to do so. Again, care is needed when handling the logarithmic RSSI values. Equation 2 describes the calculation of Euclidean Distance in signal space.

$$d_{j,k} = \sqrt{\left(c_j^{AP_1} - s_k^{AP_1}\right)^2 + \left(c_j^{AP_2} - s_k^{AP_2}\right)^2 + \ldots + \left(c_j^{AP_i} - s_k^{AP_i}\right)^2}$$

Where:

c	RSSI value radiomap data [dB]
s	RSSI value measurement data [dB]
d	euclidean distance in signal space [-]
i	number of access points [-]
j	index in radiomap [-]
k	index in measurements [-]
AP	access Point MAC address [-]

Eq. 2: Distance in signal space (Teuber and Eissfeller, 2006)

For each measurement, the radiomap is checked for the presence of all three MAC addresses to a specific grid cell. These are then considered candidate locations. If this is not the case, grid cell candidates where two MAC addresses match are used. For each candidate grid cell, the Euclidean Distance in signal space between the measurement MAC-RSSI combinations and the radiomap MAC-RSSI combinations is calculated. The measurement is assigned to the location for which the Euclidean Distance is minimal.

5. RESULTS

In this part, the results following from the methodology described above are elaborated on.

5.1 Sensor platform results

In this section, results are described comprising temperature and humidity measurements at different locations obtained during seven continuous days of measuring. The results for the noise measurements are not discussed as they are unreliable, due to the automatic gain control of the microphone. During these days, there were different devices not working for different periods of time, due to connector failures or connection deficiency due to lose solder, which were faced during the measurement week. Therefore, the study area, which is covered by a 50x50 m grid, shows diversity of measurements considering location and time.

Overall, the different days of measuring are characterised by diversity of temperature intervals, following a pattern. For each day, both temperature and humidity line graphs present stable, upward or downward trends without having extreme peaks and inconsistencies. The observed average values deviate between 15 and 43 °C. Moreover, for the whole week of measurements there is an expected correlation between temperature and the relative humidity during the day; whenever the temperature rises, the humidity levels decrease. In addition to that, a considerable difference in the performance of the sensors mounted on the different vehicles is observed. The temperature levels derived from the sensors, placed on the boats, are higher than the ones on the Tuk Tuks. Consequently, the humidity values obtained from boats are lower than the ones from Tuk Tuks. Another important remark on the performance of the systems on the boats is the high fluctuation of the values during the day, compared to those derived from the Tuk Tuk platforms. The highest measurements do not correlate with location of the platform since for every day they are located in different grid cells.

The first two days of measurements have the lowest observed temperatures: from 15 to 32 °C. For the next days, there is clear upward trend of temperature, having as a peak 59 dB on the 14th of July. These values deviate a lot from the actual temperatures this day, which were up to 25 °C (KNMI, 2017).

On the 14th of July, the most extreme measurements were obtained. The extreme values are represented by just a few measurements around 15 o'clock. Overall there are a lot of measurements in the temperature interval 39-46 °C, which are obtained in different hours of the day. These extreme values are assessed in the following section.

5.2 Quality of sensor results

As far as the quality of these results is concerned, difference is made between sensor data and locations. Assessment of the sensor data results is done based on benchmark tests proving the temperature values. It was found that the sensor platforms on the boats measured unexpected high values in case they were not moving. The difference between the data from boats and from Tuk Tuks in the same time intervals is around 5 °C. Further research on sensor data quality is recommended.

5.3 Quality of localisation results

As far as localisation results are concerned, quality is assessed in three ways: by comparing the locations found from the methodology to the GPS coordinates, to the Google geolocation API and to expectations based on their prior location.

For the GPS and Google API comparison, coordinates found from these reference techniques were converted to grid cells. In case the same grid cell is found for a measurement as from the localisation methodology, it is considered correct. Additional analysis on prediction of location was done, based on assuming a constant vehicle speed. This analysis was performed on only a small set of sample measurements. Results of the correctness checks are shown in table 3.

Correctness	GPS	Google API	Previous
Correct	79.6 %	2.5 %	33.3 %
Within 1 cell	20.0 %	31.4 %	44.4 %
Within 2 cells	0.3 %	11.3 %	11.1 %
Incorrect	0.1 %	54.8 %	11.1 %

Table 3: correctness of localisation method (own work)

Striking is the difference between GPS and Google API coordinates, which goes up to 2.5 km. The GPS measurements are assumed to be reliable since they are positioned at expected places on the route. The self-composed radiomap is more reliable than the Google API radiomap. This is the case since the self-made radiomap is constructed in an exactly similar way as the measurements are taken which have to be localised afterwards. This similarity makes the radiomap very reliable for the specific purpose.

Further research on the accuracy of the Google API coordinates should be performed. Another remark has to be made on the fact that the assumption of speed for the previous location check is very rough: including an accelerometer when using this test in future research is recommended.

6. CONCLUSIONS

This chapter summarises the current research by outlining all conclusions and the answering the main research question.

6.1 Localisation

Regarding the localisation methodology, the following conclusions were drawn: GPS is reliable, but is least efficient with regard to energy usage. The GPS sensor has still been used in our research, as a reference localisation method and way to create the radiomap.

The trilateration method based on both BLE and WiFi RSSI measurements is not sufficiently reliable, due to the fact that these measurements are affected by multipath and also require an extension of the network infrastructure with static components covering the study area. Therefore, this technique was deemed as not feasible to be implemented.

LoRa localisation was rejected as option as well due to it depending on the presence of LoS. In addition, internal clocks of the gateways were found not to be synchronised and the time precision was far from sufficient to provide for reliable localisation.

Therefore, use was made of WiFi fingerprinting for localisation. A major drawback of this technique was avoided by automating the process of composing the radiomap, using GPS coordinates. Euclidean Distance in signal space was then used to match measurements to a location in the radiomap.

6.2 LoRa communication

The LoRa communication technology introduces considerable limitations for dynamic sensor systems. When a 30-byte message is sent at SF12 (which is the size and spreading factor most used in this research), it means that approximately once every 10 minutes a message can be sent. For obtaining near real-time sensor data in a city, more sensors are needed to get full coverage of the area. The message size also limits the amount of MAC addresses that can be sent for the radiomap, which makes the location determination less trustworthy. However, in case the localisation can be performed on the sensor platform itself by increasing its storage capacity, the LoRa message length could be decreased significantly therefore allowing a higher message frequency. As a communication method LoRa could be very suitable for sending sensor data via static devices of which the location is already known. This possibility should be further researched.

6.3 Vehicles

Any electrical vehicles with regular operating times following a constant route would be suitable for the proposed application. In this specific case, Tuk Tuks and tourist boats were selected. The tests have revealed that the specific location for fastening the sensor platform has major influence on the sensor readings. Also, significant differences in measured values by Tuk Tuks and boats were found. These require further investigation to determine how to handle the quality of these measurements.

6.4 Main research answers

To conclude, the main research question "*To what extent can **near real-time spatiotemporal data** be obtained using a **dynamic** sensor network based on **LoRa communication** in an urban environment?"* can be answered as follows:

- *Near real-time*: The developed measurement system provided readings every three minutes, which were distributed throughout the case study area in an urban environment. The data was represented to the public in an online dashboard. The frequency of the data exchange for a location is highly dependent on the number of vehicles that is included in the research.

- *Dynamic*: The sensor platform prototype was sufficient for deployment on moving vehicles, even though the functionality was not completely stable and continuous. Furthermore, in order for complete coverage of the city centre to be obtained, a network with higher density is required.

- *Spatiotemporal*: After inspecting and testing the board's WiFi, Bluetooth and LoRa capabilities, it was concluded that the most effective way to determine the location is by using WiFi fingerprinting, based on a radiomap composed with additional GPS measurements during the radiomap composing period.

- *Data*: the quality of the collected data returned contrary results. While the localisation proved to be sufficient for the current use case, the sensor data was often not representative of the ambient air temperature. Further research about the most suitable sensor setup, acquiring unbiased measurements, is required.

- *LoRa communication*: this communication method is not fully suitable for the deployment of a dynamic sensor network, due to the small maximum payload size and sending frequency restrictions. Nevertheless, the power usage is very low, the range of long and it can be used to a certain extent.

7. DISCUSSION AND RECOMMENDATIONS

Due to the limited time and scope, not all localisation techniques have been researched to their full extent. While wireless communication fingerprinting methodologies have been documented extensively, trilateration for these techniques are considered too unreliable. However, the theoretical description of tests provided in this research shows potential in the relationship between distance and RSSI. Further research is required to prove the feasibility of such localisation methods for practical implementation with the use of self-learning algorithms and network approaches. Such implementations could be envisioned in an IoT environment in which more devices are connected and together creating a network.

Sensor fusion could also be used for the localisation of different sensor platforms: this would mean combining the information from different sensors to deduct a location. The prediction of movement used in this research is an example of this methodology to predict or validate a location. The addition of extra, inertial, sensors would in that case be beneficial.

The legal implications of collecting MAC addresses from WAPs should be avoided. In a broader roll out of IoT applications, resulting in more publicly available WAPs, only a specific set of MACs can be considered for fingerprinting, thus avoiding the possibility to be dealing with personal data.

The current localisation methodology can be improved by adding a Euclidean distance threshold value, including the likelihood of measurements per cell and involving knowledge about a previously known location. In addition, a procedure is to be developed to keep the radiomap up to date when measuring over longer periods of time.

For the use of dynamic sensor networks in a broader implementation, attention should be paid to sensor network standards. Examples of appropriate standards are the Sensor Web Enablement and SensorThings by the Open Geospatial Consortium (OGC).

ACKNOWLEDGEMENTS

In the process of making this research project a success, acknowledgements are obliged to the municipality of Delft for their funding, E. Verbree, M. Meijers and S. van der Spek for their guidance, the Science Centre TU Delft, with T. Verkerk, R. Braggaar and A. Wyber in specific, for sharing their experience, B. van Loenen and L. Dalla Corte for their advice, M. Tiggelman for his support, B. van Goor, and IoT academy for sharing their knowledge and Delft City Shuttle and Rondvaart Delft for offering their vehicles.

REFERENCES

Caragliu, A., Del Bo, C., & Nijkamp, P. (2011). Smart Cities in Europe. Journal of Urban Technology, 18(2), 65–82.

Google. (2017). The Google Maps Geolocation API. Retrieved May 10, 2017, from https://developers.google.com/maps/documentation/geolocation/intro

Jol, M. (n.d.) KPN LoRa: Advanced Workshop. Presentation, IoT Academy.

Khedo, K. K., Perseedoss, R., & Mungur, A. (2010). A wireless sensor network air pollution monitoring system. arXiv preprint arXiv:1005.1737.

KNMI. (2017). Uurgegevens van het weer in Nederland. Retrieved on June 19, 2017, from: *http://projects.knmi.nl/klimatologie/uurgegevens*

Larson, T., Gould, T., Riley, E. A., Austin, E., Fintzi, J., Sheppard, L., ... & Simpson, C. (2017). Ambient air quality measurements from a continuously moving mobile platform: Estimation of area-wide, fuel-based, mobile source emission factors using absolute principal component scores. Atmospheric Environment, 152, 201-211.

Lora Decryption on Application Server (2017), Zakelijk KPN Forum, Retrieved June 13, 2017, from https://zakelijkforum.kpn.com/lora-forum-16/lora-decryption-on-application-server-8416

Mautz, R. (2012). Indoor Positioning Technologies. (D. R. Mautz, J. Müller-Gantenbein, & A. Geiger, Eds.), Sgc.Ethz.Ch. Zürich: Print-Atelier ADAG.

NovAtel. (2003). GPS Position Accuracy Measures. NovAtel. Retrieved from https://www.novatel.com/assets/Documents/Bulletins/apn029.pdf

Szewczyk, R., Osterweil, E., Polastre, J., Hamilton, M., Mainwaring, A., & Estrin, D. (2004). Habitat monitoring with sensor networks. Communications of the ACM, 47(6), 34-40.

Teuber, A., & Eissfeller, B. (2006). WLAN Indoor Positioning Based on Euclidean Distances and Fuzzy Logic. Institute of Geodesy and Navigation, University FAF, Munich, Germany.

Van der Spek, S. (2017). Smart(er) Environments (1st ed.). Delft.

Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor network survey. Computer networks, 52(12), 2292-2330.

[This page has been left blank intentionally]

[This page has been left blank intentionally]

Introduction

This introduction provides information about the context, requirements and goals of this research project. It is divided into seven parts. Firstly, the setup of the project within the Master of Geomatics Programme is elaborated on. The relevance of the project is explained in the problem introduction. Then the context of the project and its goal are defined. Based on this, a main research question and several sub-questions are defined which the research project is to answer. A hypothesis is formed about these questions in the next part. Afterwards, the scope of the project is defined and also the technical requirements of the research are stated. The introduction closes with a brief reading guide to the rest of the document: how is it set up in order to be able to answer the research question in the end.

1. Project setup

The project described in this document is a part of the educational Master's programme in Geomatics, at the Faculty of Architecture, TU Delft. The project is called Synthesis Project and aims at applying all knowledge acquired in prior mandatory courses of the programme into practice. In total, five different Synthesis Projects are executed in parallel by different student groups of each five or six students. With respect to their contents, the five projects can be divided into two categories: Points Clouds and Internet of Things.

The research project described in this document is, together with one other Synthesis Project, focused on the Internet of Things (IoT), together forming a couple of partially overlapping but most of all complementary research. The difference between the two projects is their practical setup character: one is focused on static sensors, whilst the other is focused on dynamic sensor platforms. This document describes the Dynamic Internet of Things Synthesis Project research.

The project team consists of six students who are guided by two mentors and a supervisor. The research is executed within a ten weeks period in spring 2017.

2. Problem introduction

Over half of the world's population is currently living in cities and by the year 2050, the number in some regions is expected to grow up to 90% (United Nations, 2014). With this intensive growth of urban areas, the issues of providing housing, public transportation or even sanitation for masses without damaging the environment will have to be handled. With the recent surge in popularity (European Commission, n.d) the Smart Cities concept is seen by many as a possible solution to this issue. According to Caragliu et al. (2011), a city is smart "when investments in human and social capital and traditional (transport) and modern (ICT) communication infrastructure fuel sustainable economic growth and a high quality of life, with a wise management of natural resources, through participatory governance". In practice, this often involves not only policy, planning and financing, but also collecting and sharing data about the cities with its' inhabitants (EIP-SCC, n.d.). On the one hand, this requires investments and with the economic disparity at its current status, the areas with the most projected growth (e.g. central Asia, Sub-Saharan Africa) also have predominantly low GDP per Capita (Figure I.1). On the other, the cost of implementation of any electronic system has incredibly decreased due to the mass production in China. However, this raises many questions, such as: what is the quality of the data? What skills are needed? Is the technology sufficient for the requirements of the application?

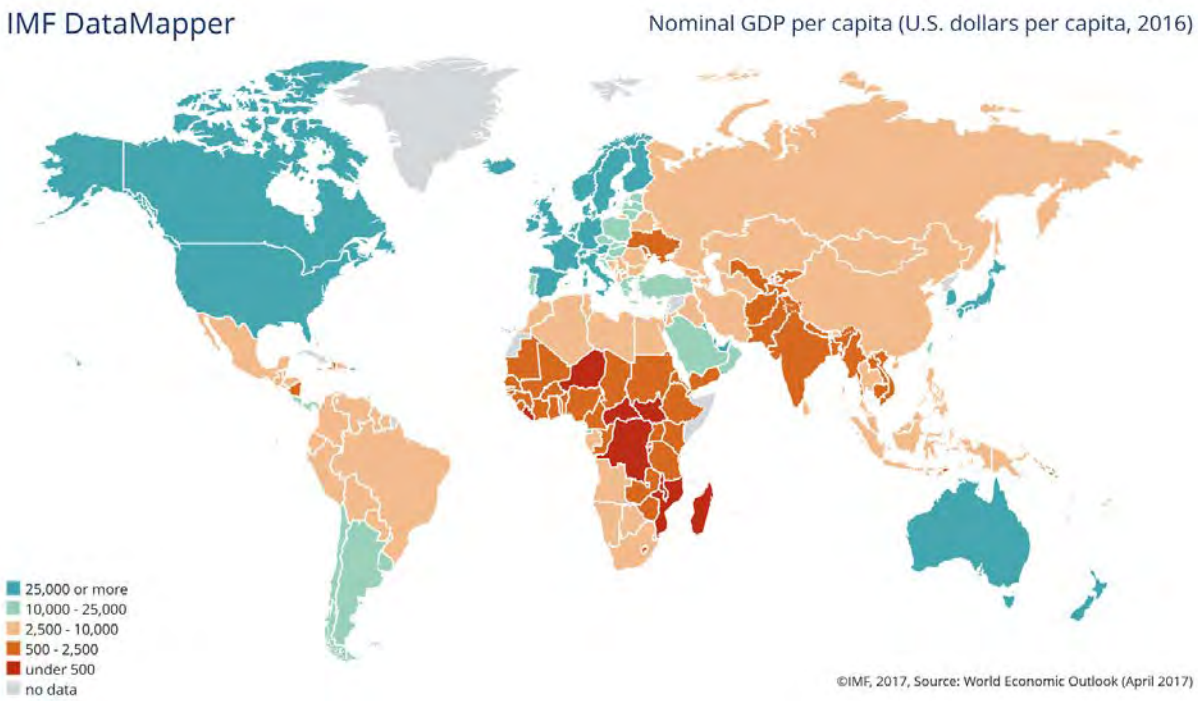


Figure I.1: Nominal GDP per Capita per country (International Monetary Fund, 2017)

3. Research context and goal

Internet of Things (IoT) is the phenomenon of objects or devices that are wirelessly interconnected and can communicate information to human beings (Xia, Yang & Vinel, 2012). A requirement for IoT applications is that its nodes use both low computation power and low energy (Atzori, Iera & Morabito, 2010). The concept of IoT can also be applied to sensor networks, which can send updates on sensor values, such as air quality and temperature, of the nodes in this network. This has already been done multiple times for static networks over the last ten to twenty years (Khedo, Perseedoss & Mungur, 2010; Szewczyk et al., 2004; Yick, Mukherjee & Ghosal, 2008). If this static network is made dynamic, by mounting sensors on moving vehicles, it has beneficial effects. Fewer sensors are needed, because a larger area can be covered by a moving sensor, as opposed to a static one (Larson et al., 2017).

When building a dynamic sensor network, localisation is one of the most important aspects; in order to make a good spatial analysis. It is important to know where a sensor is at any moment in time. There are multiple existing methods for localisation in urban environments, of which Global Navigation Satellite System (GNSS), or an adjusted form of GNSS, is the most common method (Huang & Tan, 2006; Davidson et al., 2009; Tian et al., 2016; El-Mowafy & Kubo, 2017). As is mentioned by Bakker (2016) GNSS modules are not suitable for dynamic sensor nodes, because they use too much energy. For longer use of the sensor network, without daily recharging of the batteries, other methods for localisation should be introduced (Schmidt, 2016).

In this report the possibilities of localisation of vehicles in a dynamic sensor network are researched and the experiments carried out on their suitability. The main goal of the research is stated as:

> *"Localisation of the transmitters, providing insight in the location and conditions of the measurements and provide an interactive data platform for empowering citizens: households, visitors and shopkeepers." (Spek, 2017, p6)*

4. Research question

Within the main goal described above, the following research question is defined:

> *To what extent can near real-time spatiotemporal data be obtained using a dynamic sensor network based on LoRa communication in an outdoor urban environment?*

In addition to this research question, eight corresponding sub-questions are defined below:

1. *Which localisation techniques are fit to localise sensors in a dynamic sensor network in an urban environment?*
2. *What level of location granularity and localisation correctness can be achieved as a result of the dynamic sensor network?*
3. *To what extent is LoRa communication fit to provide near real-time spatiotemporal data from moving sensor platforms in an urban environment?*
4. *What electronic and physical components are essential for the sensor platform?*
5. *What database architecture is sufficient to store a significant amount of sensor data?*
6. *Which vehicles are suitable for measurements and how does this deployment affect the sensor data?*
7. *What are the fitting methods to visualise the obtained data?*

In chapter 6, Conclusions, these main question and sub-questions will be answered. However, a hypothesis on the main question was defined at the start of the project.

5. Hypothesis

As far as the research question is concerned, some expectations can be defined. These together form the hypothesis of this research. Whether the hypothesis is correct will be found in the end of this document, in chapter 6 on conclusions.

The expectation of the project team is that near real-time *spatiotemporal data* can be obtained using a dynamic sensor network in an urban environment, although the level of location granularity and localisation correctness is expected to require further improvement, in order for relevant spatiotemporal data to be obtained. However, it is also expected that this research provides a first basis for future research with potential in the field of dynamic sensor networks.

6. Scope

The aim of the project team is to develop a network of approximately ten sensor platforms moving throughout the city, so that the coverage could be improved without the need of deploying more stations. The network will be primarily based on LoRa communication, but secondary systems can be used to achieve higher precision (Spek, 2017).
The sensor platforms, that measure environmental parameters, will be mounted on moving vehicles (ibid.). To keep the data manageable and comparable to the parallel running project on a static sensor network, the group chose to limit the study area to the city centre of Delft. The measurements will ideally be carried out during a two week period but the system design should be aimed at permanent collection on the long run.

7. Requirements

At the start of the research project, project requirements were specified in a Must-Should-Could-Would (MoSCoW) framework (Table I.1). The highest level, *Must*, defines the demands without which the project would either be not legal/safe, or would not provide a viable solution (Agile Business Consortium, n.d.). In other words, the delivery of these conditions are guaranteed. The next level, *Should*, implies that the requirement is needed, but not vital for the project. The only difference between that and the *Could* category lies in the impact on the project, the latter being of less importance. The last class, *Would*, could also be called “Won’t Have”: the project team must agree on specific needs that fall beyond the time scope, time of the project. This category is there to manage expectations and ensure that the project team does not informally re-introduce the requirement. All of this helps keep the focus and help decide on planning throughout the project.

Must	Should
1. localise the sensors *2. implement usable sensor platforms on moving objects* *3. store the data* *4. continuous and safe functionality*	*5. compare results with static network* *6. check the results with Move3 software* *7. provide data quality*
Could	**Would**
8. provide (near) real-time data *9. visualise the analysis* *10. provide a dashboard* *11. count by-passers*	*12. create a web application* *13. track vehicles with irregular movement patterns* *14. integrate the static and dynamic networks to support and calibrate each other* *15. Push large data-sets online for more detailed environmental research*

Table I.2: MoSCoW method evaluation for the project requirements (own work)

Two of these requirements are identified as killer requirements at the start of the research project: they have a high impact on the project, but are also likely to fail. The killer requirements of this project are:

1. localise the sensor platform: this is regarded likely to cost a lot of time, as localisation with the proposed technique LoRa is expected to be of insufficient accuracy.

2. implement sensor platforms on moving objects: the chance is likely that not enough vehicles can be used. In addition, it is regarded a challenge to design a system which provides results usable for any environmental analysis.

Nevertheless, with high level of technical support from the mentors and external parties, confidence is present that the project is feasible.

8. Reading guide

This document describes all the steps taken to answer the research questions defined above. To do so, it is divided into seven parts. Firstly, theoretical research is performed in chapter 1 *Localisation Techniques* on different possibilities for localisation of dynamic sensor stations in an urban environment. Then, practical research is performed in chapter 2 *Technical Implementation* on how to deploy a dynamic sensor stations network. In chapter 3 *Benchmark Tests* findings from chapter 1 and 2 are brought into practice to test the principles and find the capabilities of different system components. Chapter 4 *Methodology* describes the methodology principles and details to be followed in order to answer the research questions. After having followed this methodology, results are elaborated on in chapter 5 *Results*. To do so, visualisations, tables and textual explanations are used. In chapter 6 *Conclusions* the research question is to be answered. Finally, chapter 7 *Discussion and Recommendations* provides a retrospective view into the way the research project was executed technically as well as a peek into the future on which field the research is recommended to be continued. The document finishes with *References* to sources used and further relevant research information *Appendices*.

[This page has been left blank intentionally]

1. Localisation Techniques

This chapter provides the theoretical background on possible localisation techniques to be applied in this research. The chapter consists of four parts. Firstly, the term localisation is defined. Afterwards, further insight is provided into the requirements of the localisation technique for this research. Then, a short description of numerous localisation techniques is given, as well as their advantages and disadvantages for this research. Lastly the conclusions are presented. The chosen localisation technique applied is elaborated on in chapter 4.3 in the methodology.

1.1. Localisation and Positioning

For this research, a differentiation is made between location and localisation on one side, and position and positioning on the other. According to Mautz (2012), localisation can be defined as a rough estimation of location where the absolute coordinates are of less importance. A location can for instance be a specific intersection or a room. While the absolute coordinates, geographic or Cartesian, are defining a position. Mautz states that a major difference between a position and a location is that a location is concerned with a context; a location has a meaning to a person.

This research will focus on localisation; determining the location of a specific sensor set at a specific moment in time. This choice was made in line with the overall goal and scope of the project: to obtain spatiotemporal data in an urban environment. This goal does not require data linked to a position but data linked to a location. An example of this would be the temperature and humidity at a specific intersection or city block in the centre of Delft.

1.2. Requirements of the Technique

The aim and context of this research sets the requirements for the localisation technique to be applied. The temporal scope, geographical scope, desired range and desired accuracy of the technique are described below as key requirements.

1.2.1. Scope

The temporal scope of this research is two months in the spring of 2017. During this period, the research plan, the measurements and the analysis of results were to be developed and executed. The localisation technique chosen should be expected to lead to a reasonable result within this period.

Pertaining to the geographical scope, the localisation technique should be fit to work within the city centre of Delft. Ideally, the sensor platforms can be localised in the whole city. The route of the Tuk Tuks spans a large part of the city centre, the university campus and the east side of the city. However, the decision was made to focus on a geographical scope covering the city centre. The reason for this is that a parallel research on a static sensor network is running in this area. If necessary this allows the static sensor network to be incorporated in the dynamic research.

Although the scope, both temporal and geographical, of this research is rather limited, longer temporal scopes and larger geographical scopes for future applications are kept in mind. This research project is regarded as a proof of concept for future research.

1.2.2. Range

The distance between sensor platforms is of importance, since many communication techniques require contact between the devices. When two devices are out of range, no contact can be made. An example of such contact is the connection of two devices over WiFi. The received signal strength indicator (RSSI) between the devices will give an indication of their distance from each other in radio space and subsequent (im)possibility to transmit data.

Range indicators like RSSI can furthermore be used to calculate distance between nodes in a network as described by the theory discussed by Parameswaran, Husain, & Upadhyaya, (2009). Thus with this

information in combination with position information of some of the nodes an estimation could be made of the location of the remaining nodes.

However, since the range between components of the sensor network is highly dependent on the localisation and communication technique, the required range is elaborated on per localisation technique in section 1.3 on localisation techniques.

1.2.3. Correctness

The level of location correctness provides a measure of quality of the localisation process. It can be seen as the equivalent of accuracy in the process of positioning. A measurement can be assigned either the correct location or an incorrect location. The percentage of measurements assigned the correct location can be used as a means to define the quality of the localisation technique: the correctness (Senaratne, 2013).

The correctness of the localisation process depends on both the technique itself and the granularity of partition of the area into locations. The desired level of correctness for the localisation process is determined by the way in which the sensor spatiotemporal data obtained in this research is to be used later on. An important stakeholder in this is the municipality of Delft.

In the process of choosing the localisation technique, the level of correctness which can be achieved when combined with a certain granularity of partition is taken into account. The granularity can be seen as a main factor in the process of implementing the technique in an apt way. The level of correctness should match the usage of spatiotemporal data obtained in the future to a sufficient level.

1.2.4. Location awareness

As the IoT enabled device will act as a dynamic sensor platform and will not be performing any of the context related tasks described by Bolchini, Curino, Quintarelli, Schreiber and Tanca (2007), there is no necessity to internally process the location. Due to the limited computation power and storage options it would even benefit the device if no localisation processing would have to be performed. The necessity for location awareness is therefore limited to the server. These so called centralised algorithms have been described by Yunhau and Zheng (2011) to be able to avoid the limitations of the devices. It is however noted that this comes at an increased cost for transmitting location information. The perfect solution would therefore use the message containing solely the measurements to localise the devices.

It can be concluded that the scope of the technique is the main characteristic of a localisation technique. Some techniques require continuous connection, which can be a very strict requirement for localisation. However this also dependent on the communication method. The correctness is mainly dependent on the usage of the results. In this research this has been accepted as a derivative of the selected localisation method.

1.3. Localisation Techniques

According to Yunhau and Zheng (2011) current localisation concepts comprise two steps. Firstly physical geographic information should be measured; followed by the computation of the required location using this data. This section will consider multiple possibilities for the first step of this process. The localisation methodologies taken into consideration are limited by the possibilities according to the specification of the hardware (*LoPy v1.0*, 2017; *U-blox*, 2011). These methodologies are: 1) GNSS, 2) Fingerprinting, 3) Trilateration, 4) Inertial navigation system (INS), and 5) LoRa. Each of these will be analysed and their advantages and disadvantages presented.

1.3.1. GNSS

Global navigation satellite systems have been developed since the 1960s and the systems have proven their worth in their extensive use (Bonnor, 2012). The positioning method is based on the time of arrival (ToA) and is documented comprehensively (Hofmann-Wellenhof, Lichtenegger, and Wasle, 2008). The main disadvantage for the use of GNSS in IoT applications is the inefficient energy consumption associated with the hardware (Bulusu, Heidemann, and Estrin, 2000; Bakker, 2016; Schmidt, 2016).

The accuracy of GNSS implementations have proven sufficient for navigation applications and in most day-to-day cases (Sommer, n.d.). Secondly all infrastructures needed for positioning already exist, which would limit the cost for an operator of an implementation. Finally the location is not known by the network or method operator, but solely by a user with a GNSS chip. Therefore no privacy issues arise (Hofmann-Wellenhof et al. 2008).

The disadvantages for GNSS in IoT or sensor networks are plentiful. It must however be noted that the systems have never been designed for such implementations (Hofmann-Wellenhof et al. 2008). First and foremost in IoT applications resources are scarce. Energy is usually provided by batteries and GNSS modules are not known to be power efficient[1] (Bulusu et al. 2000; Bakker, 2016). Besides this, the costs associated with an extra hardware module and antennas would require IoT chips to be more expensive and could therefore hinder the deployment of different applications (Schmidt, 2016). Furthermore multipath affects both the accuracy and precision of the GNSS implementation (Tsakiri, Stewart, Forward, Sandison, and Walker, 1998; Groves 2011). According to Le Marchand, Bonnifait, Ibañez-Guzmán, Betaille, and Peyret (2009) this error can reach up to 200 meter, putting the usability of the methodology for localisation in an urban environment up for discussion. The location awareness of the chip and not the system provides an extra challenge in the applications envisioned by the authors. As the network needs to know the nodes location, and communication over GNSS is not possible, another method needs to be found the send this information. Finally the operation of GNSS systems is dependent on many external factors, e.g. GPS and GLONASS are operated by two major militaries. This limits the information available about these systems and could lead to coverage issues if either of them decides to switch off their system (Hofmann-Wellenhof et al. 2008).

All in all GNSS does not effectively provide correct localisability for the planned implementation of the IoT sensor platforms. The increased burden on resources is not counteracted by the ease of use and available infrastructure.

1.3.2. Fingerprinting

Fingerprinting or neighbourhood measurement is based on proximity measurements to nodes around a location (Yunhau and Zheng, 2011). These measurements are to be compared to a database containing known position with their related proximity measurements to at least two known nodes; based on which the location of the new node can be calculated (Mautz, 2012). This method of localisation can be used in combination with many communications forms, in this paragraph WiFi and Bluetooth Low Energy (BLE) will be considered.

WiFi

WiFi fingerprint has very low extra cost in the implementation. Both in monetary cost (Xue, Qiu, Hua, and Yu 2017) as well as in the cost of resources. The chip used in these experiments is already equipped with WiFi connectivity. Furthermore WiFi access points are abundantly available providing sufficient cover for neighbourhood measurements. According to Mautz (2012) there exist multiple implementations in which high accuracy has been achieved. These have been designed for indoor use; however a similar approach has been developed by Google which theoretically works everywhere (Google, 2017).

According to Mautz (2012) a few drawback exist. All of these drawbacks arise from the fact that wireless local area network (WLAN) communication networks are not built for localisation. This means problems in localisation arise with changes in environmental factors or change in the WiFi chip used. Mautz suggest extensive (pre-) calibration to counteract these complications. The addition of information about the signals besides the received signal strength indicator could further offset these drawbacks (He and Chan, 2016).

BLE

Bluetooth low energy fingerprinting has similar implementation details as WiFi fingerprinting. However according to Faragher and Harle (2015) BLE has some key advantages over WiFi. Their testing shows that the power draw of BLE is lower. This is beneficial for the use of the localisation method in an IoT implementation. They also suggest an easier approach for building the support network for coverage. This is compared to the requirements for building a WiFi network. However this seems to be applicable to a

[1] *The authors are aware of the chip discussed by Bakker (2016), regrettably this chip was not available for this research.*

network built from scratch, while WiFi networks are usually abundant available in the Netherlands. The final implementation details mentioned by Faragher and Harle are subtle difference which can be attributed to the different specifications of WiFi and BLE. They state that these could solve some of the drawbacks found in WiFi fingerprinting, mainly the interoperability between different devices, as the specification requires the signal strength the be reported using a set method. It must however be noted that research by Escudero, Hwang, and Park (2017) recommends the inclusion of channel state information (CSI) to achieve better results.

Fingerprinting could be a reliable method to localise the different sensor platforms. The microprocessors used in the research are equipped with the technology required and power draw, specifically for BLE is low. There are two approaches to the use of a radio map for localisation. Firstly existing pre-built APIs are available from different sources (HERE, n.d.; OpenWiFi, n.d.; Google, 2017), these are continuously updated and maintained by their creators. The Google API (Google, 2017) is an example of an already built service which is trained by its users; however the documentation does not specify all methods used for localisation. This would result in a black box principle, in which the authors are generally aware of the localisation method but cannot check or influence this. Secondly, to use a reliable fingerprint system, a new database would have to be created, and updated regularly, with proprietary algorithms for localisation (Zhang, Liu, Song, Gurrin, and Zhu, 2013; Xue, Qiu, Hua, and Yu, 2017). This latter option would be necessary in case BLE would be chosen over WiFi as well as the building of a network with sufficient coverage as BLE beacons are less prone to be available.

1.3.3. Trilateration using RSSI distance[2]

According to Yunhau and Zheng (2011) radio signal strength (RSS) techniques are based on the notion that radio signals abate over distance. They state that the methodology is a cheap solution as communication methods are supposed to be part of the nodes. However they do also accept that no breakthrough technology exists yet. Rusli, Ali, Jamil, and Din (2016) furthermore note that trilateration has one clear advantage over fingerprinting. This advantage is the absence of an offline training period. Due to the similarities in WiFi and BLE signal propagation models their implementation approaches have the same advantages and disadvantages.

For trilateration using RSSI, similarly to fingerprinting, no expensive extra hardware is needed as the communication techniques are used for the localisation. Additionally the need for an offline (pre-) calibration period is removed. However due to the signal propagation, properties of both WiFi and BLE, shadowing, and multipath are significant issues (Yunhau and Zheng, 2011). These effects would further aggravate the inconsistent measurement performance of the RSSI (Parameswaran et al., 2009). To foil these issues multiple connections would be necessary as well as an overdetermined least squares minimisation algorithm (Mautz, 2012). This processing would have to be done at a server as the microprocessors are not sufficiently powerful for these computations; hence more communication between the devices and the server is needed. Another limiting factor is the range of the communication techniques, these are heavily dependent on the power with which a device transmits, noise, path loss, and the signal attenuation (Haagmans et al. 2017). These limitations would require a new network to be established, using static nodes for localisation.

The signal propagation specifics of the communication methods make this an undesirable method to localise IoT devices. The environmental impact on the RSSI is of such magnitude that other localisation techniques are more appropriate. This, combined with the need for more communication with a server, excludes this method from consideration for implementation, however more research is needed to completely exclude this technique for future research.

1.3.4. Inertial Navigation System (INS)

Inertial navigation systems or the use of dead reckoning is based around the first law of motion from Newton (Mautz, 2012; Noureldin, Karamat, and Georgy, 2013). Two possible set-ups for INSs are described by Noureldin et al. (2013). The first is a strapdown system and the second a gimbaled system. Both make use of accelerometers and gyroscopes. They state that the gimballed system is too expensive for practical

[2] Trilateration using timing is not taken into account as timing is not part of either the WiFi or BLE protocol and accurate measurements are not possible. (Mautz, 2012)

use. However the relative low cost of the electronics involved makes the implementation of strapdown systems, in which the accelerometer and gyroscope are mounted directly to the device, more feasible. Noureldin et al. note that the system is relatively small and light, with a high accuracy. This is however disputed by Mautz (2012). He argues that the use of previous positions without the addition of further sensor data is not sufficient for an accurate localisation tool. Noureldin et al. remark the same over longer distances and suggest the combination of GNSS and INS.

It can be concluded that the use of solely INS will not suffice the needs for localisation in this approach. The required hardware is not part of the microprocessor and would have to be an addition to the sensor platform. Furthermore would it require being operational at all times to provide constant information from the INS sensors, increasing the need for energy and decreasing battery life. This combined with the occurrence of drift over longer periods of time make this method not suitable for IoT applications.

1.3.5. LoRa

Long Range Radio or LoRa® is a relatively new technology designed by Semtech for the ISM Band (Schmidt, 2016). As noted by Henriksson (2016) LoRa is not sole solution using a similar approach, others include Sigfox and NB-IoT. LoRa has been chosen for this research as it is available on the hardware accessible for this research (*LoPy v1.0*, 2017). LoRa has been established to send messages over long distances using minimum resources as catalyst for the IoT movement; by using the ISM band communication is limited to 0.1 % to 10% of the time, restricting the burden on resources (Schmidt, 2016).

The specification of LoRa is under patents from Semtech (Hornbuckle, 2010; Olivier and Sornin, 2016). According to Henriksson (2016) the use of the proprietary Chirp Spread Spectrum (CSS) in combination with a technology that can create the CSS noise resistant, with high precision, and using cheap crystals, is the key to LoRa Wide Area Networks (LoRaWAN). These characteristics lead to a chirp barely affected by multipath fading and slightly affected by the Doppler spread. Although the main goal of LoRa communication is, does Henriksson describe two principles to localise a device using LoRa. These are RSS and Time Difference of Arrival (TDoA). The latter of these is stated to be a multilateration problem which could be solved using hyperbolic functions. However Schmidt (2016) notes that the temporal resolution of LoRa is limited, due to the bandwidth and the resulting multipath; and in the worst case this could lead to a precision of 600 metres within a LoRaWAN. The effects of these limitations has been found by Verbeke, Olti, and Munteanu (2016) as in their research the TDoA measured varies by 56ns under perfect line of sight (LoS), sufficient for a sixteen metre deviation in distance measurements. The multipath error introduction also applies to the RSS measurements in non LoS measurements, similarly as in approaches based on other communication techniques.

KPN and CSEM claim to support localisation over LoRa using trilateration and TDoA (Gray, 2016; Jol, 2016). However their setups are elaborate and require significant time investments to return the location, e.g. a 30 min delay for localisation. Neither company elaborates on their claims or presented any of their accuracy and precision results. The authors reached out to both KPN and CSEM for further explanation. KPN denied to cooperate to this research. CSEM has not responded to multiple request for information.

To summarise, LoRa is a new proprietary communication technique which might have the capabilities to be used as a localisation technique. This would come with the added benefits of low power consumption, and cheap hardware; the key ingredients for IoT applications. However only limited literature is available so far and further research is required before LoRa can be successfully implemented as a localisation method.

1.4. Localisation Conclusions

From table 1.4.1 it can be concluded that none of the discussed localisation methods are perfect for the requirements of the project. However some are more appropriate than others. The extra costs involved with the use of GNSS limits the capabilities of the sensor platform or other implementations for the Internet of Things. WiFi and BLE fingerprinting are a proven success but they do come with the downside of continuous maintenance or the use of non-transparent APIs provided by third parties. Trilateration based on RSSI measurements from BLE and WiFi is heavily influenced by the existence of multipath and would require an implementation with an extensive new network. INS is not sufficient for localisation without the use of extra localisation methods. LoRa is new communication technique which has possible capabilities to

provide periodic location information. This research will continue to explore the possibilities of WiFi fingerprinting and trilateration and LoRa localisation. These techniques provide the best opportunities while using the least amount of resources. Range tests and calibration tests for the used hardware will need to prove whether the capabilities of the devices suffice for the applications developed.

Methodology	**Advantages**	**Disadvantages**
GNSS	Proven accuracy No new infrastructure needed Local localisation possible	Expensive Energy consumption Affected by environment Extra hardware needed Dependence on owners Extra communication needed
WiFi Fingerprinting	Low cost implementation Available APIs and methodologies	Dependency Depreciation of signal map Multipath interference Offline training period Extra communication needed
BLE Fingerprinting	Low energy consumption Low cost implementation	New infrastructure needed Depreciation of signal map Offline training period CSI recommended Extra communication needed
WiFi Trilateration	Low cost implementation No offline training period	New infrastructure needed Fixed node location needs to be known Affected by environment
BLE Trilateration	Low cost implementation No offline training period	New infrastructure needed Fixed node location needs to be known Affected by environment
INS	Relatively accurate nearby Cheap components	Drift Extra hardware needed Fusion of data necessary
LoRa	Low cost implementation Extreme range	Physical accuracy limitations Limited communication Little research available.

Table 1.4.1 Methodology comparison (own work)

2. Implementation Details

The deployment of a dynamic sensor system is limited by numerous implementation details, both within the sensor platform design and the processing of data. These are described in this chapter. Firstly, the limitations of the LoRa communication protocol are elaborated on. Furthermore, the restrictions of the vehicles to be chosen to make the network dynamic are described. Afterwards, implementation detail of the database structure are mentioned. The chapter end with description of the limitations raised by the sensor platform electronics and casing design.

2.1. KPN LoRa

The method used to wirelessly communicate the gathered data from the LoPys to the online database is LoRa. The Dutch telecommunication company KPN has a developer portal set up for LoRa, which private citizens can try out without charge. By setting up a configuration file on the LoPy, which contains the Device Address, Network Session Key and Application Session Key, a decrypted message can be sent by the LoPy and received by one of the KPN LoRa antenna. The network has full coverage in the Netherlands (*LoRa | KPN Grootzakelijk,* n.d.). The maximum payload size of this message is up to 54 bytes (Jol, n.d.).

The power with which an uplink message (device to server) is sent is indicated by the so-called Spreading Factor (SF). The closer the device is to one of the LoRa antenna, the lower the SF can be (Table 2.1.1). One of the reasons why the lowest SF possible is preferred, is to minimise battery usage. The other reason has to do with the Time on Air. The LoRa message is sent over the unlicensed 868 MHz band. To use this spectrum, the user has to obey the rules set by the government (Jol, n.d.). One of the most important rules to be taken into account is that devices can only send messages 1% of the time. This means that when a device sends a message of 10 bytes using SF12 (with Time on Air of 1.6 s), it has to be silent for 158.4 seconds after that.

Spreading Factor	Bitrate	Range in free field	Time on Air[3]
SF7	5470 bps	2 km	50 ms
SF8	3125 bps	4 km	100 ms
SF9	1760 bps	6 km	200 ms
SF10	980 bps	8 km	400 ms
SF11	440 bps	10 km	800 ms
SF12	290 bps	14 km	1600 ms

Table 2.1.1: Relationship between SF, Bitrate, Range and Time on Air of LoRa (Jol, n.d.)[4]

2.2. Vehicle Requirements

The goal of the current project is to establish a dynamic sensor network. In order for this to be achieved the sensors are mounted on moving vehicles. The network is designed in such a way that the location of the sensors is known continuously.

At the start of the project a variety of vehicles were considered for the installation of the sensors, buses, Tuk Tuks, tourist boats, public transport bikes ("*OV-fietsen"*), parking control cars and postal service bikes. As the most preferable option for the vehicles the Tuk Tuks were chosen, because of their advantages – they drive through the city centre and have a clearly defined schedule and route. Moreover because they are electrical vehicles they don't produce additional noise, heat or air pollution, which could affect the measurements from the sensors. Delft City Shuttle, the responsible organization for the Tuk Tuks, agreed to support the project, but the amount of running vehicles per day is only three to five. This is not sufficient for the deployment of the network.

Therefore, some of the additional options were explored further. As least preferable are the buses, because their routes extend further than the city centre of Delft and they produce additional noise, heat and air

[3] *For typical payload around 10 bytes*
[4] *Values are indicative and approximated, because they depend on environmental conditions and hardware conditions.*

pollution, which will affect the gathered environmental data. This is also a negative effect of the parking control cars, as they run on petrol. The other three options – tourist boats, public transport bikes and postal service bikes – present some disadvantages that have to be considered. For example, the post services most probably will not continuously cover the same part of the city the whole day, but only pass every street once per day. The public transport bikes could be used by people who cycle to the university or their job, leave it there the whole day, and return to the train station at the end of the day. This will not contribute to a dynamic network. The data gathered from the boats will be biased from the fact that some conditions within the channels, such as humidity and temperature, will differ from the streets. On the other hand the boats will travel the same route through the city centre during the entire day, so location-wise this would be the best option. As a final requirement the support from the responsible organizations and their employees will affect the suitability of the remaining options. A summary of the requirements can be found in the table 2.2.1.

The responsible party for the tourist boats was contacted, and they also agreed to have some sensors on their electrical boats. This means the network consists of Tuk Tuks, driving a continuous route through Delft, and boats that travel a continuous route through the city centre.

Vehicle type	*Noise/heat/air affection*	*Known route*	*Continuously dynamic in Delft*	*Support organisation*
Tuk Tuks	++	++	++	++
Postal service	++	+	-	?
Public transport bikes	++	--	-	?
Buses	--	++	--	?
Tourist boats	-	++	++	++
Parking control	--	--	+	?

Table 2.2.1: Comparison suitability vehicles (own work)

2.3. Databases

A database is described as an organized tool capable of keeping data or information that you can retrieve in an effective and efficient way when the need arises (Alvaro, 2016). Databases and database systems are an essential component of life in modern society: most of us encounter several activities every day that involve some interaction with a database (Elmasri & Navathe, 2015). It is fair to say that databases play a critical role in almost all areas where computers are used, including business, electronic commerce, social media, engineering, medicine, genetics, law, education, and library science (Elmasri & Navathe, 2015).

2.3.1. Provided infrastructure

A database management system is used in implementation of this project, since part of the project is to store a great amount of implicit information derived from a number of sensors both as raw data and in a meaningful way. That makes necessary the use of a server which will await and fulfils requests from clients. In communication with people who are in charge of running and maintaining the server belonging to the Delft University of Technology, authorized access to the server was obtained. Having access to it, a virtual machine (VM) was installed which is a software layer to a real machine in order to support the desired architecture (Smith & Nair, 2005). VMs circumvent real machine compatibility and hardware resource constraints (Smith & Nair, 2005). VMs enhance software interoperability, system impregnability, and platform versatility (Smith & Nair, 2005). PostgreSQL was installed in the virtual machine, which is a powerful, open source object-relational database system with a proven architecture that has earned a strong reputation for reliability, data integrity, and correctness (PostgreSQL, 2017). PostGIS was installed as well, which is a spatial database extention for PostgreSQL object-relational database, adding support for

geographic objects and allowing location queries in SQL (PostGIS, 2017). In addition to PostgreSQL, installation of Node-RED took place as well. Node-RED is a flow-based programming tool for wiring together hardware devices, APIs and online services (Node-RED, 2017). It provides a browser-based editor that makes it easy to wire together flows, using a wide range of nodes in the palette that can be deployed to its runtime programme in a single-click (Node-RED, 2017). After having installed Node-RED in the virtual machine, it was connected with PostgreSQL, which proved to be a complicated procedure since there were issues with authentication from within the tool. Figure 2.3.1 shows the flow of actions that took place in order to set up the digital environment.

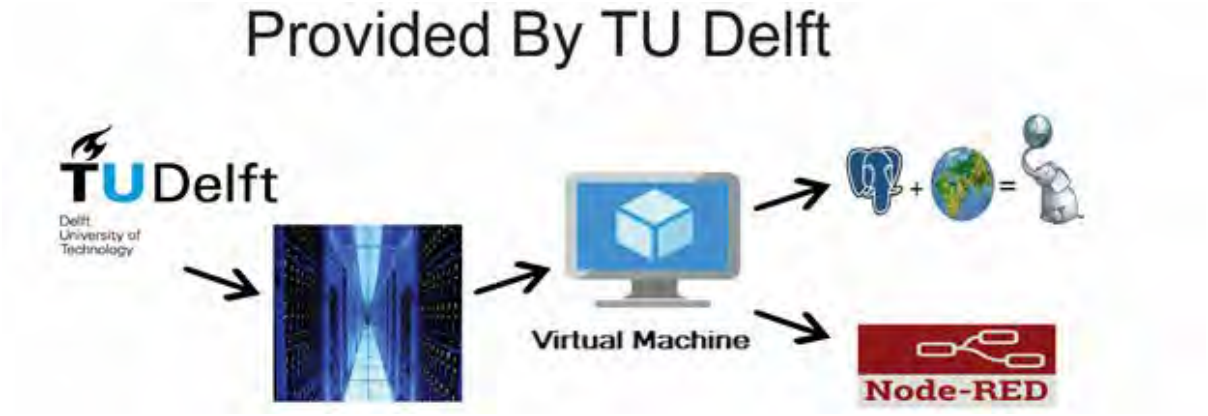


Figure 2.3.1: Infrastructure provided by TU Delft (own work)

2.3.2 Create-Drop-Populate Tables and Retrieve Data

Once the database has been created, it must be populated with database resources (objects and services) to store and/or facilitate reference of data from the system (Foster & Godbole, 2016). These resources include tablespaces, tables, indexes, views, synonyms, procedures, triggers, packages, sequences, users, and roles (Foster & Godbole, 2016).

Tables were created in the database using a template node and a Postgres node. Template nodes set a property based on the provided template and by default they use the so called *'mustache'* format, but this can be switched off if required (Node-RED, 2017). Postgres nodes execute the query specified in msg.payload (Node-RED, 2017). When receiving data from the query, the msg.payload on the output will be a JSON array of the returned records (Node-RED, 2017). A script compliant with SQL language was written to create these tables with columns of certain types to populate them with data. For the Postgres node specifications like host, port, database, username and password have be specified.

To gain insight into the tool, code was written to drop the created tables. The same sequence was followed to populate the table with dummy data: a template node and a Postgres node were used. While the attempt to populate the table with dummy data was successful, populating the columns with parts of the msg.payload triggered errors therefore, a function node was used instead. Functions nodes are function blocks where someone can write code to do more useful things (Node-RED, 2017). The message is passed on as a JavaScript object called msg (Node-RED, 2017). By convention it will have a msg.payload property containing the body of the message (Node-RED, 2017). In the function node attention should be paid, since while writing SQL compliant script, incorrect use of quotes will lead to misconception, especially with values which contain numbers and characters, i.e. hexadecimals. After having populated the database, the next step is to retrieve this data. To accomplish that a *'select'* query in a template node had to be written and to be connected to a Postgres node. This node should be modified, by checking the dialog box 'Receive query output?', in order to be able to return data (see appendix A).

2.3.3 Schema

Knowing how to create, drop, populate and retrieve data from the tables, raw and meaningful data should be stored in an efficient way.

Raw data schema

In order of a raw data table to be created, all attributes in the message received in the LoRa developer portal should be stored. In this message, there is an array which contains attributes of the KPN antennas that detected the message sent by a LoPy. Up to 9 antennas can detect a message that sent from Delft's city centre. So, it is possible to have variables Lrr0_LrrId, Lrr0_chain, Lrr0_LrrRSSI, Lrr0_LrrSNR, Lrr0_LrrESP from 1 up to 9. Considering that it is not possible to dynamically change the number of columns of a SQL table, control of the number of KPN's antennas that detected each message took place. According to this number a different SQL style query was sent to the database populating only the columns for which there were values and leaving the rest of them undefined. Table 2.3.1 shows the variables stored as part of the raw data from the received message (see appendix B).

Time	Mic_hex	Lrrid	Lrr0_LrrESP
DevEUI	Lrcid	Late	CustomerID
FPort	LrrRSSI	LrrLat	CustomerData_pro
FCntUp	LrrSNR	LrrLON	CustomerData_ver
ADRbit	SpFact	Lrr0_LrrId	ModelCfg
MType	SubBand	Lrr0_chain	InstantPER
FCntDn	Channel	Lrr0_LrrRSSI	MeanPER
Payload_hex	DevLrrCnt	Lrr0_LrrSNR	DevAddr

Table 2.3.1: Raw data schema (own work)

Meaningful data schema

Decryption

Regarding the meaningful data, a proper schema should be created. The research by Pottinger and Bernstein (2008) has been consulted for reference. The majority of variables that would be stored in the meaningful table are contained in payload_hex attribute. In order to extract useful information from it, the payload_hex should be decrypted. Encrypting and decrypting data have been widely investigated and developed because there is a demand for a stronger encryption and decryption which is very hard to crack (Al-Hazaimeh, 2013). Advanced Encryption Standard (AES) algorithm is a block cipher text. The block size can be 128, 192 or 256 bits. The key lengths can be 128(AES -128), 192(AES -192), and 256 (AES -256) bits (Al-Hazaimeh, 2013). The LoRa payload is by default encrypted with AES-128 bits encryption, based on the generic algorithm described in IEEE 802.15.4/2006 Annex B [IEEE802154] as described in the LoRaWAN specifications (section 4.3.3.) (Lora Decryption on Application Server, 2017). A KPN customer can choose to decrypt their payload in the KPN Thingpark platform, before they get the data forwarded to their own server (*Lora Decryption on Application Server*, 2017). The Thingpark platform enables scalable LPWA networks and interoperable IoT applications and services (ThingPark Products, 2017). It was chosen to forward the data encrypted, since the Key *AppSKey* and Device Address *DevAddr* are known.

Decryption Scheme
The plaintext is encrypted in the payload and AppSKey and DevAddr are known (Lora Decryption on Application Server, 2017). The scheme works by dividing the payload in blocks of 16 bytes. Figure 2.3.2 shows the sequence of actions that have to take place in order to decrypt the message.

- Per block of 16 bytes payload, an AES vector is calculated and XORed with the payload block.
- The first round is done with the first 16 bytes of the payload.
- In each round, an ABlock is determined by using the round of block# (so first round block#=1), resulting in the following array: [0x01, 0x00, 0x00, 0x00, 0x00, Direction (up=0, down=1), DevAddr bit 0 (LSB), DevAddr bit 1, DevAddr bit 2, DevAddr bit 3 (MSB), FCntUp bit 0 (LSB), FCntUp bit 1, FCntUp bit 2, FCntUp bit 3 (MSB), 0x00, block#}]
- The ABlock is encrypted with the AES-128 ECB scheme using the AppSKey, resulting in an SBlock for that round.
- The 16-bytes Payload block is XORed with the 16 bytes SBlock to get part of plaintext. When the payload block is less than 16-bytes, the plaintext is retrieved by using first part of the SBlock that corresponds to the number of bytes in the payload block. For instance, when the payload block has 9 bytes, it must be XORed with the first 9 bytes of the SBlock of that round.
- By repeating rounds until all the 16-bytes blocks of the payload are decrypted, the complete plaintext is retrieved.

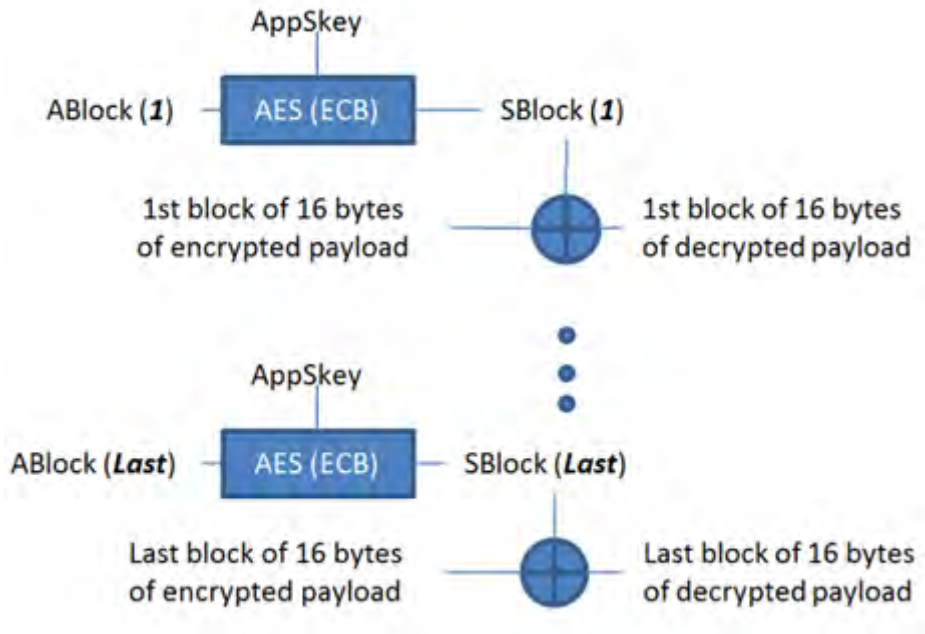


Figure 2.3.2: Decryption schema (Jol n.d.)

Google Maps Geolocation API
After having decrypted the attribute payload_hex, before populating final meaningful table, a function node was used to create a JSON style message containing all mac addresses and the strength for each mac address that was detected by the LoPy in order to send them to the Google Maps Geolocation API (Google, 2017). This API was chosen for its reliability and the simplicity of the message that should be written in order to request data as well as the simplicity of the response received by it. Google Maps Geolocation API was used as a control mechanism regarding localisation, as will be described in the chapter 5. It returns a location and accuracy radius based on information about cell towers and WiFi nodes that the mobile client can detect (Google, 2017). The message was sent using an http request node which is a node to make http requests and its method changed to POST instead of GET, which is the default, in order to transfer the JSON style message, Figure 2.3.3 (see appendix C).

Figure 2.3.3: Message sent to Google Maps Geolocation API (own work)

As Figure 2.3.4 shows, a response received from Google's geolocation API contains the longitude and latitude and accuracy in metres of the mobile client.

Figure 2.3.4: Response from Google's geolocation API (own work)

A function node was used to manipulate the message received from the Google API in order to extract the coordinates and the accuracy as float number, and store them in new variables to populate the meaningful table (see appendix D).

Data was sent by the LoPys and by Google's API should be combined in one function node in order to form the final SQL query which will be send to an SQL node to store all data belonging to one measurement. Considering that every function node sends a message which has a form of msg.payload, it is impossible for the *'receiver'* node to make the distinction between sources of messages. To solve this problem, a msg.topic variable was initiated in the function nodes, which sent the variables from LoPys and from Google's API. So, a control of the topic of each message took place in the function node which constructs the SQL query to populate the meaningful table (see appendix E). Figure 2.3.5 shows the final form of the meaningful data table. The final Node-RED flow is appended to this report in appendix F.

devaddr characte...	date date	time time wit...	mac1 characte...	strength1 integer	mac2 characte...	strength2 integer	mac3 characte...	strength3 integer	tempera... integer	humidity integer	sound integer	gps_latitude double prec...	gps_longitude double precis...	api_latitude double preci...	api_longitude double prec...	api_accuracy double precis...	gps_location geometry	api_location geometry
142042F9	2017-06-...	10:43:28	36:23:87...	62	54:fa:3e...	63	34:23:87...	63	17	71	0	51.966605	4.367016	52.0019421	4.3662195	73	0101000020E...	010100002...
14204AEE	2017-06-...	10:44:08	6a:b5:7e...	48	74:b5:7e...	49	84:16:f9...	54	16	79	47	52.015014	4.369793	52.0149081	4.3692983	52	0101000020E...	010100002...
1420447E	2017-06-...	10:44:28	00:0c:f6...	85	94:a7:b7...	87	d8:9d:67...	87	16	57	0	51.9627	4.3161	52.02524	4.3578835	69	0101000020E...	010100002...
14203742	2017-06-...	10:44:38	58:23:8c...	47	5a:23:8c...	48	6c:aa:b3...	52	15	82	0	52.01033	4.357734	52.0102833	4.3566207	73	0101000020E...	010100002...
1420410E	2017-06-...	10:44:59	ac:22:0b...	60	a4:17:31...	68	64:d1:a3...	70	23	49	0	52.00663	4.363731	52.0067086	4.3629621	67	0101000020E...	010100002...
142042F9	2017-06-...	10:46:40	00:3a:7d...	73	f0:b4:29...	74	00:3a:7d...	74	17	71	0	52.003101	4.370997	52.0043852	4.3699123	63	0101000020E...	010100002...
14204AEE	2017-06-...	10:47:20	dc:9f:db...	83	e6:8d:8c...	86	dc:9f:db...	88	16	80	29	52.017543	4.378973	52.0181461	4.3787618	54	0101000020E...	010100002...
1420447E	2017-06-...	10:47:40	94:a7:b7...	86	d8:9d:67...	87	72:a7:b7...	89	16	57	0	52.024566	4.359501	52.0252141	4.357879	63	0101000020E...	010100002...
14203742	2017-06-...	10:47:51	1c:b9:c4...	61	e0:3f:49...	65	e0:3f:49...	67	16	81	0	52.008163	4.356683	52.0082011	4.3556966	73	0101000020E...	010100002...
1420410E	2017-06-...	10:48:12	82:c7:a6...	54	84:4b:f5...	56	0a:18:d6...	56	23	50	0	52.00901	4.35942	52.0094072	4.3582396	54	0101000020E...	010100002...
142042F9	2017-06-...	10:49:57	64:d1:a3...	55	38:d8:2f...	57	f8:8e:85...	61	18	70	31	52.011375	4.369377	52.0110819	4.3687988	53	0101000020E...	010100002...
1420410E	2017-06-...	10:50:21	5c:dc:96...	53	38:10:d5...	56	b0:c2:87...	57	22	55	0	52.010418	4.358288	52.0104814	4.3573537	53	0101000020E...	010100002...
14204AEE	2017-06-...	10:50:34	b2:c2:87...	65	b0:c2:87...	67	06:18:d6...	68	16	80	0	52.013996	4.365248	52.0138514	4.3646711	53	0101000020E...	010100002...
1420447E	2017-06-...	10:50:54	00:0c:f6...	82	94:a7:b7...	84	d8:9d:67...	87	16	57	0	52.024597	4.359494	52.02524	4.3578835	69	0101000020E...	010100002...
14203742	2017-06-...	10:51:04	e4:f4:c6...	53	88:03:55...	54	6c:aa:b3...	57	16	81	0	52.011135	4.356336	52.0112816	4.3557186	52	0101000020E...	010100002...
142042F9	2017-06-...	10:53:10	dc:9f:db...	70	dc:9f:db...	76	e6:8d:8c...	78	18	69	0	52.01781	4.379577	52.0181461	4.3787618	54	0101000020E...	010100002...
1420410E	2017-06-...	10:53:34	5c:dc:96...	43	70:5a:9e...	53	72:5a:9e...	59	21	56	0	52.010395	4.358307	52.0104839	4.3574421	53	0101000020E...	010100002...
14204AEE	2017-06-...	10:53:47	d0:b2:c4...	58	d2:b2:c4...	58	72:5a:9e...	60	17	79	0	52.010326	4.320235	52.0102339	4.3563142	73	0101000020E...	010100002...
1420447E	2017-06-...	10:54:07	94:a7:b7...	85	d8:9d:67...	87	72:a7:b7...	88	16	57	0	52.024547	4.359493	52.0252141	4.357879	63	0101000020E...	010100002...
14203742	2017-06-...	10:54:19	14:cc:20...	51	4c:5e:0c...	51	90:5c:44...	53	16	78	0	51.9627	4.3161	52.0084715	4.3613008	53	0101000020E...	010100002...
142042F9	2017-06-...	10:56:23	c4:14:3c...	65	c4:14:3c...	67	20:c9:d0...	68	18	69	34	52.014038	4.375771	52.0140123	4.3753828	53	0101000020E...	010100002...
1420410E	2017-06-...	10:56:48	5c:dc:96...	41	4c:09:d4...	59	b0:c2:87...	60	21	56	0	52.010501	4.358299	52.0104874	4.3573695	52	0101000020E...	010100002...
14204AEE	2017-06-...	10:56:59	bc:8c:cd...	61	1c:3a:de...	63	bc:05:43...	63	17	77	0	52.010906	4.356727	52.0104089	4.3561881	52	0101000020E...	010100002...
1420447E	2017-06-...	10:57:19	d8:9d:67...	87	94:a7:b7...	88	00:0c:f6...	89	16	58	0	52.024578	4.359534	52.02524	4.3578835	69	0101000020E...	010100002...
14203742	2017-06-...	10:57:32	f8:04:2e...	62	62:8a:4c...	63	ec:8a:4c...	64	16	78	0	52.009548	4.369564	52.0097622	4.3685986	52	0101000020E...	010100002...
142042F9	2017-06-...	10:59:35	2c:c5:d3...	70	88:5b:dd...	73	2c:c5:d3...	73	18	69	0	52.013263	4.377273	52.0135908	4.3762189	51	0101000020E...	010100002...
1420410E	2017-06-...	11:00:02	08:95:2a...	70	0a:95:2a...	70	b2:c2:87...	71	21	56	0	51.967523	4.358032	52.0104757	4.3572592	53	0101000020E...	010100002...
14204AEE	2017-06-...	11:00:11	e4:6f:13...	53	1c:6a:7a...	67	1c:6a:7a...	67	17	77	0	52.007324	4.359441	52.0068573	4.3589021	53	0101000020E...	010100002...

Figure 2.3.5: Schema of meaningful data (own work)

2.3.4 Dashboards/ Near real-time measurements

Finally, in order to present near real time measurements, three ui_chart nodes were used. These nodes plot the input values on a chart (Node-RED, 2017). This can either be a time-based line chart, a bar chart (vertical or horizontal), or a pie chart (Node-RED, 2017). So, three more function nodes were used to isolate the measurements of temperature, humidity and sound in order to be sent to the ui_chart nodes (see appendix G). The visual output is shown in Figure 2.3.6.

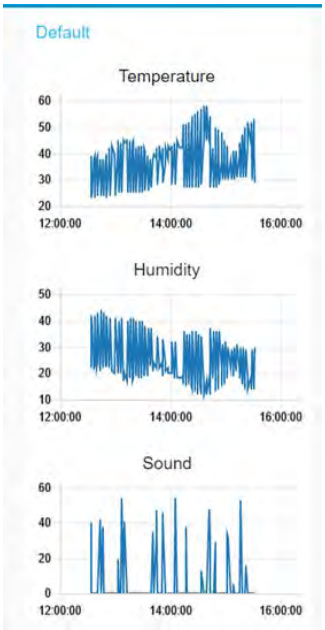


Figure 2.3.6: Real-time dashboards of temperature - humidity – sound (own work)

2.4. Sensor Platform: electronics

One of the main products of this Synthesis project was a self-developed sensor platform. The main requirements for it was to provide different means of localisation and gathering of environmental data. Important part in this was that the scope of components' selection was very limited. Both IoT research groups were to work with the LoPy microcontroller unit (MCU) and most of the other components were already on the way. Having little to no expertise in the field of electronics or environmental measurements within both teams, the most essential part of the project was to determine which of the available elements were to be used and how that had to be done. With this in mind, four environmental factors were chosen to be measured: humidity, temperature, noise levels and air quality. The following modules were selected for that purpose:

- Humidity & temperature sensor- AM2302
- Microphone -MAX9814
- Particle sensors - PMS5003 and alternative GP2Y1010AU0F
- GPS module - Ublox NEO-6M chip based board
- LoRa and WiFi antennas
- Charging circuit- MCP73871 or ADA-1305

In this chapter, the requirements and specifics of each of these modules will be inventoried, including the microcontroller LoPy and the energy supply solution that is to be used in our experiment. Next, the process of designing and production will be discussed. As last, the results will be explained and recommendations for the next year's projects will be provided, including relevant ready-made alternatives.

2.4.1. LoPy

According to the producer specifications (*LoPy v1.0, 2017)* LoPy microcontroller, based on ESP32 chip - a low-cost, low-energy system, is widely used within the maker community. Systems based on this chip include Adafruit HUZZAH32 and SparkFun ESP32 Thing. However, LoPy is also a MicroPython[5] enabled system and, on top of WiFi and Bluetooth connection possibilities, also has built-in LoRa transceiver (SX1272). The system has a dual processor with 512KB RAM and 4MB external flash memory. Moreover, it can support Python multithreading and has hardware floating point acceleration. There are 2 UART and 2 SPI bus connection possibilities, but I2C and I2S busses are also supported (for full pinout see appendix H). There are 8x12 bit Analog Digital Converters (ADCs). Moreover, LoPy has a built-in ceramic Bluetooth and WiFi antenna of -0.5dBi[6]. The LoRa antenna is not and an external antenna is necessary for this functionality. Lopy needs between 3.3V-5.5V input energy. In active Wifi mode it uses 12mA, 5uA in standby, whilst in active LoRa mode it uses 15mA and 1uA in standby. Neither LoPy, nor ESP32 documentation provides insights on the full system power usage.

2.4.2. Power supply and charging circuits

The power supply provided for the project is 3.7 V, rechargeable, 5 Ah capacity battery. Having five working days for measuring, this would allow 41.67 mA for average power consumption. If the batteries were to be changed every day, the average power consumption could rise to 625 mA. Important to note is that the battery temperature should not exceed 60°C, as this might damage it. To reduce the risk of this happening, the casing should at least be covered in reflective material. The battery has a JST connector, which has a pitch not fitting to the standard of 2.54mm (0.1 in), used for most through-hole components. Therefore, to be able to make a connection with, for instance, a perfboard an extra break-out component would have to be soldered. Another thing to consider while using lithium polymer batteries is their discharge characteristics (Figure 2.4.1). The minimum power supply for the 5V pin is 3.4V, which, keeping in mind that the fully charged battery has a voltage of 4.2V, would indicate that the battery would be effective until it is around 90% discharged.

[5] *Python 3.5 implementation optimized to work on microcontrollers*
[6] *Antenna 2450AT43B100, information based on e-mail communication with a Pycom representative.*

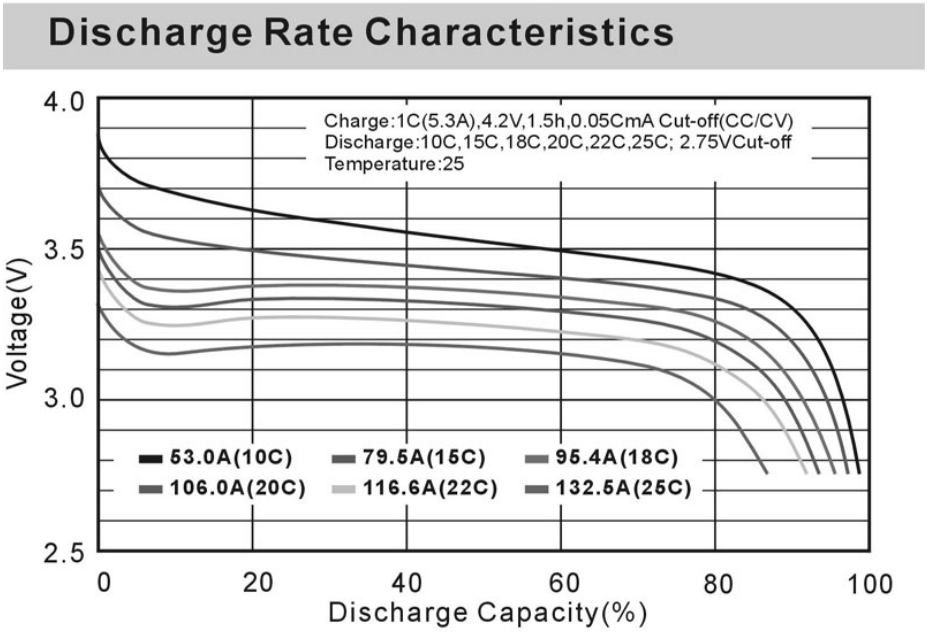


Figure 2.4.1: typical lithium polymer battery discharge characteristics (Fullymax, n.d.)

Furthermore, it is possible to provide extra leeway if the solar charging solution was to be chosen. In this case, the platform would be using MCP73871 charging circuit, which allows for a more efficient power transmission from the solar panels to the battery. What is more, this would also allow charging with mini-USB, meaning that the batteries would not have to be removed every time they are charged. This would warrant for longer durability of the connectors. Nevertheless, that would also require more connections from the casing to the outside, further increasing the risk of humidity near the electronics. Another reason not to use the solar charging solution is that only in the perfect conditions, with the professional panel it would charge 378 mA. Important to note, that this is far from what the charging system will be likely to make, since the project team has access only to low quality solar panels and the conditions will be far from perfect. Moreover, the platforms will likely be in the shade most of the time. Due to this reasoning and the involved cost for the charging circuits, the project team decided to not choose for a solar panel option. Therefore, the final solution was to use a charging circuit ADA-1305, allowing 100mA and, with slight modification, 500mA charging current. This entails that the batteries used for the project can be charged in under five hours.

2.4.2. Humidity & temperature sensor

AM2302 is a sensor with an output temperature compensated and calibrated digital signal (Figure 2.4.2). The sensor requires a start signal sent from a microcontroller unit (MCU) to change from standby to running status. However, this cannot be done more often than every 2 seconds. Once the collection is done, the AM2302 sends a response signal of 40-bit data on relative humidity and temperature values and switches to standby mode. The temperature readings have an accuracy of ±0.5°C, humidity - 2-5%. The energy usage in active mode is 1.5mA, in standby: 1mA. The system works on 1-wire bus, requiring connection to 1 digital pin, 3.3Venergy input and a ground. Between the digital and energy input lines, there needs to be a single 1kΩ resistor (pull-up resistor), as is mentioned in the data sheet. Important to mention, is that application notes mention that the performance of the sensor will debase if it is exposed to strong light and ultraviolet.

Figure 2.4.2: The humidity and temperature sensor (own work)

2.4.3. Microphone

The MAX9814 microphone is a microphone with automatic gain control (AGC), variable gain amplifier (VGA) and low-noise microphone bias (Figure 2.4.3). This implies that the system is not suited for our purposes, as AGC implies that loud noises will be automatically reduced and quiet noises amplified. Unfortunately, this function is impossible to turn off and that was found out only very late into the project. Nevertheless, the sensor is fully implemented on the sensor platform. The input voltage is 2.7-5.5V and the current draw is 3.1-6 mA. The output value is biased at 1.23V, however LoPy's ADC (analog to digital converter) pin input range is 0-1.1V. To be able to extend this range to 3.3V, the attenuation has to be increased to 11dB.

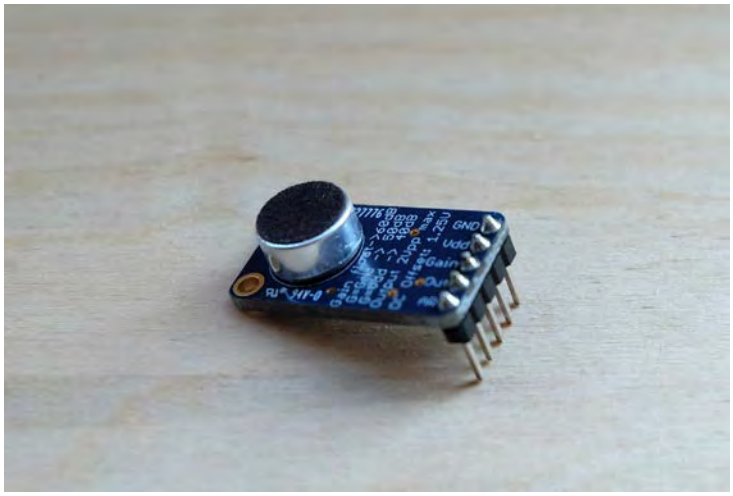


Figure 2.4.3: The microphone module (own work)

2.4.4. Particle sensors

Original sensor

The sensor, which was originally planned to be implemented on the platform is PMS5003 (Figure 2.4.4). The measurements are based on the laser scattering technique. The concentration of particles of different sizes can be measured by evaluating the scattered light profile. The sensor works on a UART bus, which requires RX/TX pin connections with the LoPy. The sensor can detect particles with diameters from 0.3 to 25 µm. The code library to retrieve this information from the sensor clusters the particle count in 6 groups, dependent on the particle size. Moreover, it provides insight into particle[7] concentration in the air and the standard deviation of it. The size of the message in this case is a potential issue, which must be considered if the sensor is to be used with LoRa communication. Another important note is that the device requires 5V supply, which would introduce an extra component to the sensor platform, a boost converter, as the maximum supply corresponds to the voltage of the available battery. Other implementation details include

[7] *Particulate Matter size subtypes: PM1, PM2.5, PM10*

that the inlet/outlet opening must be of at least the size in the sensor and they have to be in close proximity of each other. Moreover, upon having the sensor in sleep mode, the fan should be active for at least 30 seconds to be able to collect reasonable data. This sensor was not implemented on the platform in the end due to its very late arrival during the measurement period and the amount of data.

Figure 2.4.4: The particle sensor PMS5003 (own work)

Alternative

The available particle sensor GP2Y1010AU0F (Figure 2.4.5) is a dust sensor based on dual-LED optical sensing system. It detects the reflected light of dust in air. However, a complete calibration needed and testing with a known particle device. Yet even then, the documentation does not ensure viable results: the sensitivity is variable dependent on many environmental factors. What is more, there are many other factors that may influence the readings:

- When outside light comes through the dust opening on the inlet side;
- When the sensor is under mechanical oscillation (vibration);
- When interior of the sensor is moisturized;
- If the sensor is located close to a noise generator (ex. Electric dust collector, power generator);

Considering all these factors, this sensor appears to be not suitable for the intended application. Therefore, it should not be used in the final design, as it would likely yield not trustworthy results.

Figure 2.4.5: The dust sensor GP2Y1010AU0F (own work)

2.4.5. GPS module

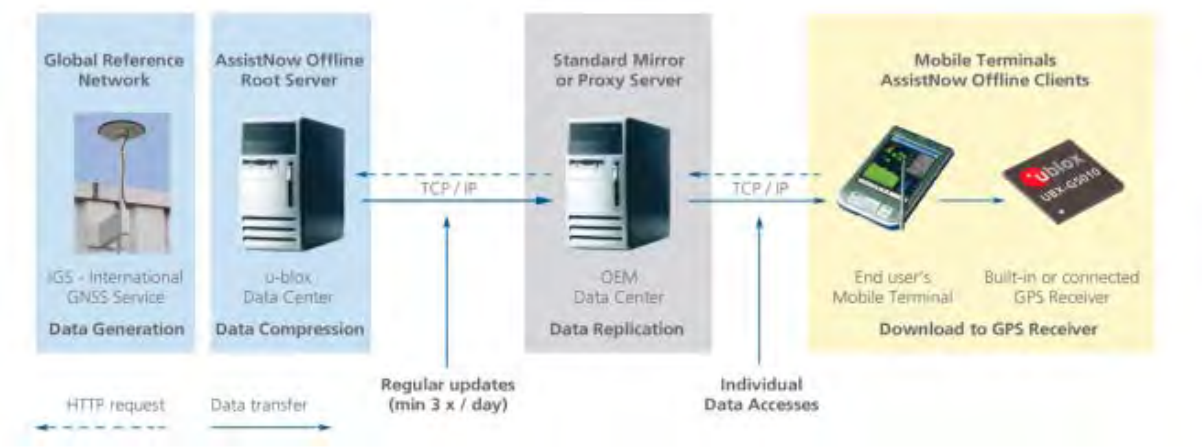


Figure 2.4.6: Principle of Assist Offline (U-blox, 2013)

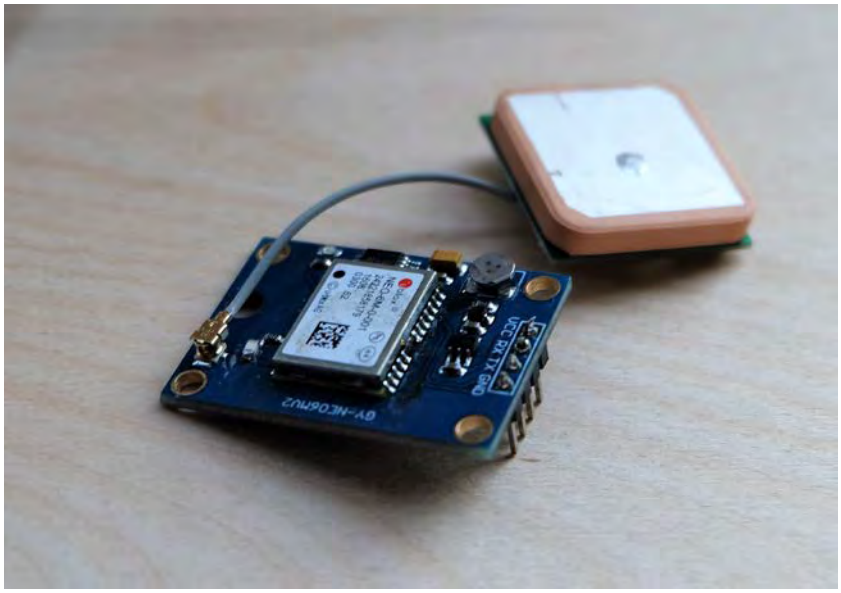


Figure 2.4.7: The GPS module (own work)

The GPS module (Figure 2.4.7) is based on u-blox NEO-6M chip, which in cases of cold and warm start has 27s Time-To-First-Fix (TTFF). If the device is hot, it is under 1s. The chip works based on AssistNow Autonomous technology, which provides functionality similar to Assisted-GPS without the need for a host or external network connection. Based on previously downloaded satellite ephemeris data, retrieved from a data centre and stored by the GPS receiver in an external memory cache, AssistNow Autonomous automatically generates accurate satellite orbital data ("AssistNow Autonomous data") that is usable for future GPS position fixes (Figure 2.4.6). This data is reliable for up to 3 days after initial capture. For such implementation, the platform would have to be extended with a SD card slot, as the LoPy does not have sufficient memory for this data. Nevertheless, the chip boasts 2.5m[8] horizontal position accuracy with only GPS, 2m with SBAS. The module has 3 power modes available:

- During a Cold start, a receiver in **Maximum Performance Mode** continuously deploys the acquisition engine to search for all satellites. Once the receiver has a position fix (or if pre-positioning information is available), the acquisition engine continues to be used to search for all visible satellites that are not being tracked.
- During a Cold start, a receiver in **Eco Mode** works exactly as in Maximum Performance Mode. Once a position can be calculated and a sufficient number of satellites are being tracked, the acquisition engine is powered off resulting in significant power savings. The tracking engine continuously tracks acquired satellites and acquires other available or emerging satellites.

[8] *Measured as circular error probable (CEP)*

- **Power Save Mode (PSM)** allows a reduction in system power consumption by selectively switching parts of the receiver on and off. PSM has two modes of operation: cyclic mode (Figure 2.4.8) and on/off mode. During the cyclic mode the receiver does not shut down completely and uses low power tracking instead, whilst during on/off mode the module switches between normal operation and low or no activity status. Important to note is that the receiver working on PSM mode will not be able to download the ephemeris, thus the modules will have to be started on continuous mode for the satellites to transmit the data.

The power consumption per mode is shown in table 2.4.1.

Parameter	Symbol	Module	Min	Typ	Max	Units	Condition
Max supply current	Iccp	All			67	mA	VCC=3.6V / 1.95V
Average supply current	Icc Acquisition	All		47		mA	VCC=3.0V / 1.8V
	Icc Tracking (Max Performance mode)	Neo-6G/Q/T		40		mA	
		Neo-6M/P/V		39		mA	
	Icc Tracking (Eco mode)	Neo-6G/Q/T		38		mA	
		Neo-6M/P/V		37		mA	
	Icc Tracking (Power Save mode / 1Hz)	Neo-6G/Q		12		mA	
		Neo-6M		11		mA	

Table 2.4.1: Indicative power requirements (product specification sheet)

Power Save Mode is enabled and disabled with the UBX-CFG-RXM message and configured with the UBX-CFG-PM2 message. The board works on the basis of UART bus, which requires TX and RX connections. The input voltage is 3.3V.

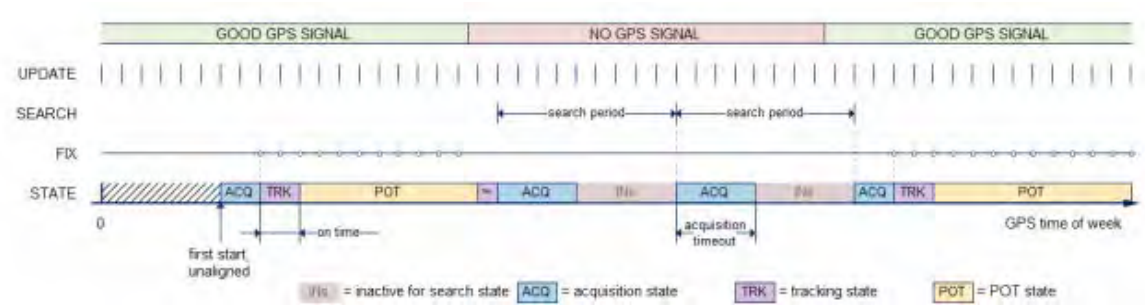


Figure 2.4.8: Diagram of cyclic tracking operation (U-blox, 2013)

2.4.6. Antennas

Firstly, the LoRa antenna and WiFi antenna are actually different, because both communication methods work in different frequencies. Even though in this particular implementation both antennas are omnidirectional and look almost identical, each is optimized for a different frequency. The reason for installing the external WiFi antenna lies in the fact that the 0.5dBi built-in antenna would likely be insufficient for the localisation purposes. The chosen solution has 8dBi antenna gain, which should be sufficient for most tests. For LoRa, as mentioned previously, there is no built-in antenna on the LoPy, thus one needs to be added to be able to send the messages at all. Both antennas can be connected to the MCU with a U.FL type of connector (Figure 2.4.9)

Figure 2.4.9: U.FL connector jack (Farnell, n.d.)

2.4.7. Design and Assembly

The first steps taken to test all of the components connected were done on breadboards (Figure 2.4.7). As not all of the components were at the team's disposal during the first weeks of implementation phase, only some of the functionality and output could be checked. The first two modules to be examined were the GPS and the temperature and humidity sensors. The testing was done in a rapid pace, as libraries for both modules were already available. The one for the humidity and temperature sensor was provided by the Science Centre support member R. Braggaar. Whilst for the GPS sensor, the code written by Peter Affolter was found online[9]. The implementation was not completely functional at first, because it was tailored for a multi-device system with a connection to a server. After filtering out these parts, the module was tested and proved to have the expected ~27s period of TTFF, explained previously. With the assurance that at least some of the modules were functioning, the design process could be pushed to a next stage.

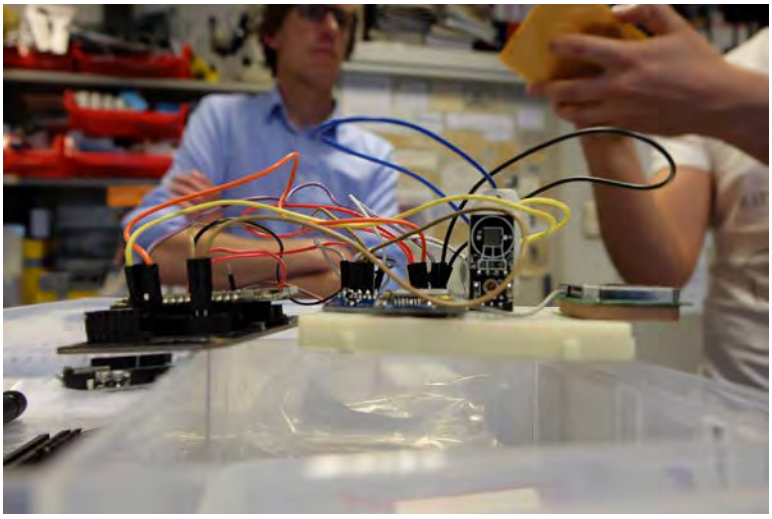


Figure 2.4.10: First test with the LoPy and the extensions that were available at the time (own work)

As the sensor platforms were to be deployed on moving carriers, the final sensor platform could not be implemented on a breadboard. This is due to the fact that they do not provide sufficient grip for the module pins and they are likely to disconnect while passing uneven pavement or speed bumps. Nevertheless, there are two alternative solutions possible, presented in table 2.4.2.

Considering the risks of not receiving all the necessary components before the test in the field and likely changes for different solutions, it was decided to choose for the perfboard. Moreover, for the experiment to succeed it was only necessary to build up to ten platforms, thus the reproduction time and the risk of module failure were also accepted. The amount of modules available provided sufficient buffer in case of this happening.

The benchmark model (Figure 2.4.11; 2.4.12) was designed based on the insights described in the preceding paragraphs and the advices provided by the technical support from the Science Centre and electronics engineers of TU Delft.

[9] *https://github.com/johnmcdnz/GPS1*

Perfboard	PCB
Quick delivery time (within 1 day)	Delivery time 3-7 working days (within reasonable costs)
Costs 2 euro	Costs 10-50 euro per board, dependent on the delivery time
Design expertise within team	Design expertise not within the team: can be outsourced for free
Redesign can be implemented within 1 day	Redesign would take minimum 1 day and the delivery time
Extra components can be added without redesign	Redesign likely needed for extra components
A lot of soldering (wire connections, components and pins)	Limited amount of soldering (components and pins)
Prototype look for the final result	Professional look for the final result
High chance of mistakes in reproduction: places of the components are **not** visible on the base board	Low chance of mistakes in reproduction: places of the components are visible on the base board
Slow reproduction rate	High reproduction rate

Table 2.4.2: Comparison of the base-board solutions (own work)

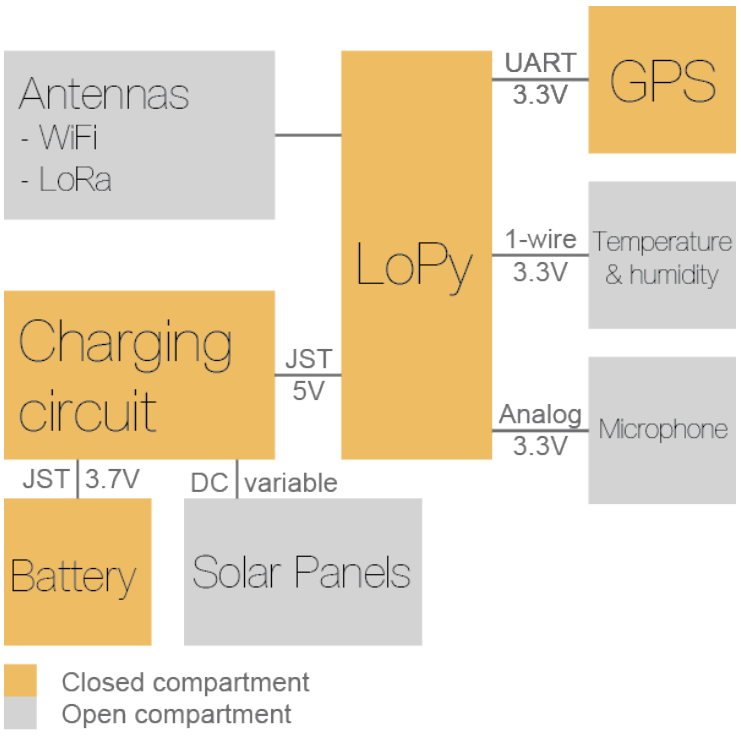


Figure 2.4.11: The schema for the board connections (own work)

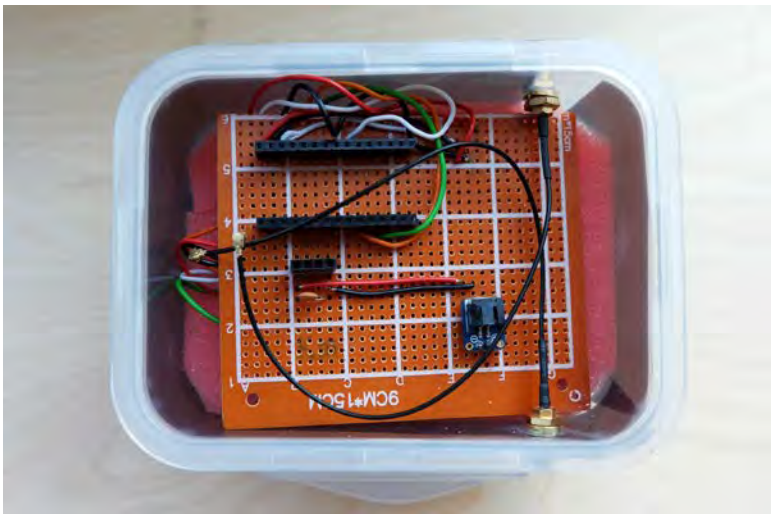


Figure 2.4.12: The perfboard solution, dry compartment without the modules (own work)

2.5. Sensor platform: casing

As important it is to have a platform that functions well in terms of electronics, it is essential to be able to protect it when the conditions are less than perfect. In case of DynamIoT, these would entitle constant vibration, strong rains and wind - all possible during the measurement period on moving vehicles. On top of that, the requirements of specific modules make the situation even more difficult: some of them need to be in direct contact with outside air, whilst with the others one cannot take risk of condensation. This chapter will explain the requirements stemming from the electronics side of the platform and the design process that was taking place to ensure timely delivery of safe and functioning product.

2.5.1 Principles

The modules to be used in this project can be categorized in ones that need no direct contact with outside air (LoPy, GPS and its antenna, battery, charging circuit) and the ones that do (noise, humidity and temperature, solar panels, LoPy/Wifi antennas). This would dictate the need of having two separate compartments for each category and creating a damp- or even watertight connection in-between.

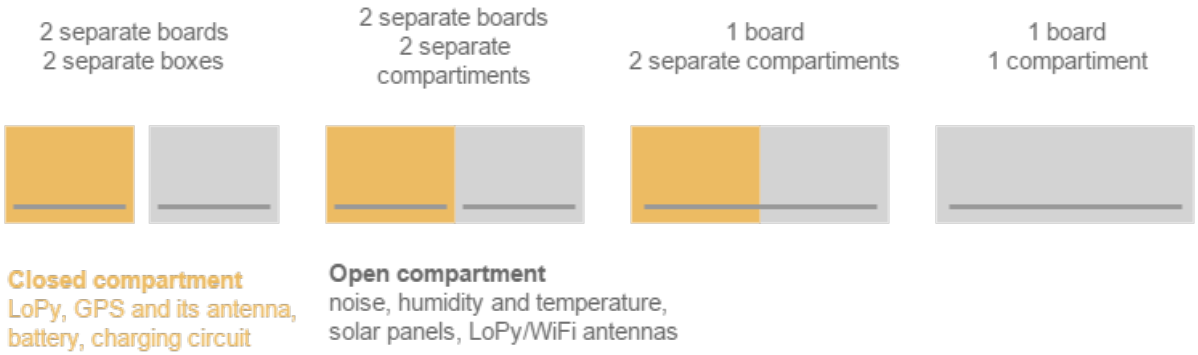


Figure 2.5.1: Compartment principles (own work)

As seen in the Figure above (Figure 2.5.1), four principle solutions were considered. The final choice is the first one, as it would allow the easiest solution for water tightness. It would have been difficult to ensure this in either wall solution (second from the left), if it was not built in the casing. The same applies for the third one, adding that the connection with the board also needed to be watertight. For the last, it would have kept all of the components in the same compartment and that would highly increase the risk of damaging the MCU, as it would be hard to prevent condensation. This component would also need extra attention regarding the greenhouse effect: the temperature inside should not rise above 60°C, as mentioned in the previous chapter.

2.5.2 Prototyping

Next step for casing design was to determine how to prototype it. The possibilities, regarding this were to:

- buy ready-made products
- fully design and creation by the team
- modify/repurpose ready-made products

After an extensive search, it was concluded that there were no suitable (regarding time, money, design) ready-made solutions fitting the requirements of the product. To ensure that this was not a mistake by novices in the field, the team organized a meeting with an expert in casing prototyping Mascha van Oossanen (IDE faculty, Applied Labs). According to her, it was highly advised to choose for the repurposing option and use watertight food containers for the final solution. The reason for such proposition was primarily due to the limited timespan of the project. Moreover, this would allow the team ease and swiftness of redesign, as it would not only be cost-efficient, but also easy to acquire.

The first casing test (Figure 2.5.2) was done on a set of food containers, covered by non-watertight lids. This issue was supposed to be solved by covering the seams with duct tape. As seen in the same Figure, 2 extra components were added: connection profiles and ventilation rasters. It was decided to use the latter to ensure as little water entering the open compartment. Since the team had access to 3D printers, multiple print-tests of ventilation covers were possible (Figure 2.5.3). As first, the team printed a model found online. However, it proved to have too small wall thickness when scaled to the necessary diameter. Consequently, multiple tests were done with own design: the cover provided sufficient coverage while simulating rain conditions, yet failed in all cases when a water spray was in use.

Figure 2.5.2 (Left): First casing solution (own work)
Figure 2.5.3(Right): Ventilation covers (left- own design, right- acquired online)(own work)

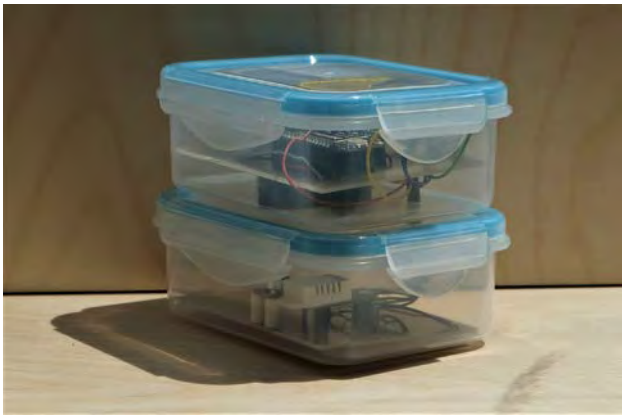


Figure 2.5.4: Second casing solution with the boards inside, but still missing the ventilation openings, sun-reflective foil and the tie-wraps for securing to the vehicle (own work)

Notwithstanding, it was necessary to find a larger container, as it was necessary to provide more space for the electronic parts. The second test (Figure 2.5.4) switched to a more advanced casing, which was both watertight and had a more reliable connection between the lid and the box. Due to the latter, it was decided to abandon the external connector idea and just connect the boxes with bolts with gum rings, to ensure water tightness. The watertight electrical connection between the boards was also ensured in a similar manner. Initially it was meant to be covered with cable tulles, but after multiple failed attempts in assembly, the cables were simply covered in shrink-tube. This ensured a tight-fit between the opening and the content. Moreover, to prevent any mechanical damage to the boards or the modules two measures have been taken. To reduce the rotational stress on the pin connections of the modules, 3D printed pins were added (Figure 2.5.5). Moreover, the boards themselves were fastened to sponge bases, which correspondingly were fixed to the casing.

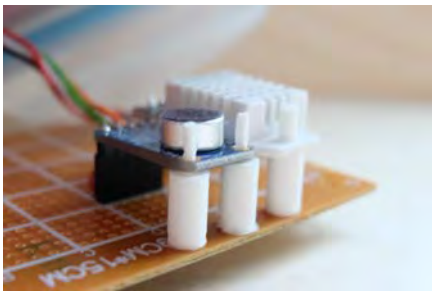


Figure 2.5.5: The modules are supported by 3D printed pins (own work)

To summarize, the requirements were implemented with the following elements:

- Ensuring water tightness (opening covering, tubes, rubber rings, etc.)
- Ensuring mechanical damage prevention (pins, soft pads)
- Ensuring no electrical shorts (bolts in nylon or bolt coverings)
- Fastening to the vehicle (tie-wraps)

The benchmark tests on this and the findings during deployment will be described in the following chapter. The final result during deployment can be seen in Figure 2.5.6.

Figure 2.5.6: The sensor platforms fastened to the carrier vehicles (left- Tuk Tuks, right- boats) (own work)

2.6. Conclusions and recommendations

All in all, this chapter inventoried the possibilities and technical details pertaining to the potential elements of the measurement system are explained. One of the most important findings of this chapter was the limitations of the LoRa communication, provided by KPN: the information can be sent merely 1% of the time and that would limit the sending frequency to once per 2.7 minutes, with the worst spreading factor and the payload of 10 bytes. Moreover, a database architecture was defined too. Database software used was PostgreSQL and the data flow took place using Node-RED. Both raw and meaningful data was to be stored. The SQL query sent to populate the table was further enriched with information from Google Maps Geolocation API. Regarding the physical part of the system, the sensor platforms were to be carried by two types of vehicles: electric Tuk Tuks and electric boats, both running on a route basis. The selected electronic elements for the final sensor platform include a microphone, temperature and humidity sensor, GPS module, 2 antennas, and a battery of 5000mAh, which is charged with a USB-based circuit of 500mA. These modules, apart from the charging circuit, were to be assembled on a perfboard. This low-level electronics prototyping is not advised for future generations of the project. Electronics is not the core of the Geomatics education and thus should either be outsourced or a ready-made solution be selected. The casing prototype is a solution involving a pair of connected water-tight food-boxes, modified with 3D printed elements. The latter involves opening one of the boxes for air-flow, which is necessary to be able to take the environmental measurements, yet keep the sensors protected from direct contact with water. For future reference, more ready-made alternatives should be considered and other casing principles tried and tested.

3. Benchmark tests

As the relevant values in the documentation sheets, especially for LoPy, were not elaborated upon, the only way to know the real performance of the system was to test it. Therefore, prior to the deployment of the sensor solution and selecting specific localisation techniques, multiple experiments were carried out regarding the functionality of both electronic and casing solutions..

3.1 Electronics

3.1.1 WiFi benchmarking

As the localisation of the moving vehicles with sensors on them is one of the most important issues of the project, three techniques for localisation are considered, namely WiFi fingerprinting, WiFi trilateration, and Time Difference of Arrival of LoRa signals. These techniques were tested in small scale in order to obtain initial information about the performance of the system.

Antenna

Within the LoPy microcontroller Bluetooth, LoRa and WiFi communication technologies are integrated. The microcontroller itself is provided with an internal WiFi antenna, but there is possibility for attaching an external one as well (Figure 3.1.1). Therefore, the WiFi range test was performed both with the internal and external antennas.

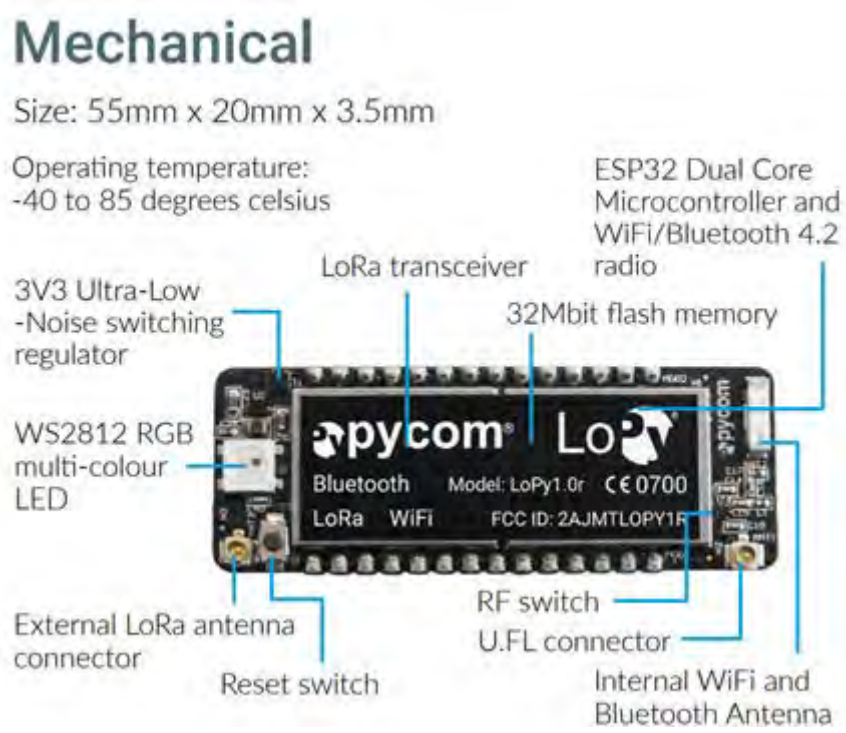


Figure 3.1.1: LoPy hardware information (LoPy v1.0, 2017)

Test setup

The tests were carried out in a systematic manner – the RSS was measured with the distance between the devices increasing with 10 metres per step up to 130 m. At every location, multiple samples were taken. A comparison between the performance of the system with internal and 3dB amplifying external antenna was made.

signal strength values are measured in dB and they deviate from -5dB to -90dB. The minimum signal strength needed for reliable communication and data exchange is up to -70db. At -80dB there is still a connection between the devices but the data may be not delivered.

The experiment started with benchmarking the maximum values that can be received having no distance between the devices and at this point, the average RSS without using antenna was -44 dB, while with the antenna it was -34,7 dB. There is a reliable connection between the devices at up to 70 m distance without using an antenna and up to 120 m with the antenna (Figure 3.1.2).

The relationship between signal strength and its expression in decibels is logarithmic e.g. 3dB decrease equals to 2 times less power, 6dB to 4 times less power, 9dB to 8 times less power, etc.

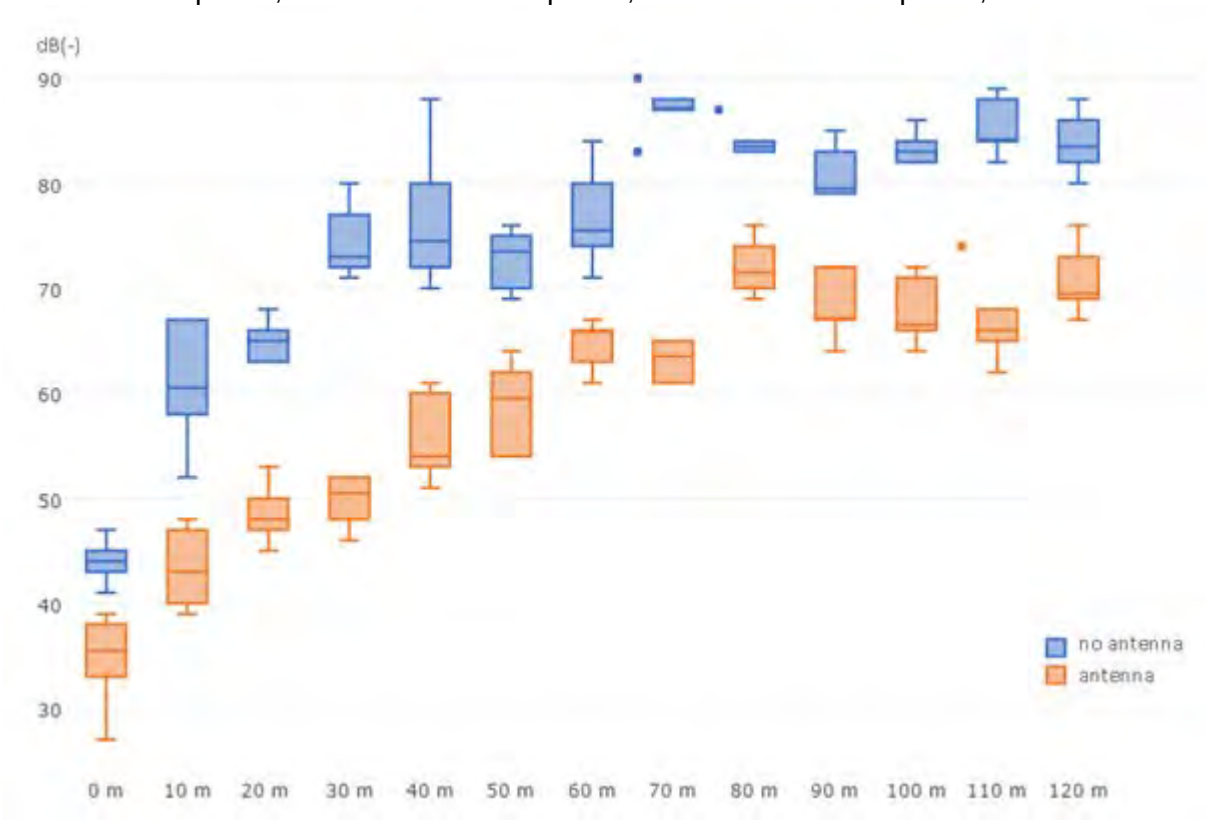


Figure 3.1.2: Plot of WiFi RSS values with or without antenna (own work)

There is considerable variation between the RSS values when the antenna is attached to the microcontroller – on average more than 10 times up to more than 100 times better signal strength.

The change in the performance of the WiFi connection between the LoPy's (with or without antenna) expressed in metrical distance shows that up to 120 m there is still reliable connection when the antenna has been used in contrast to 70 m distance relying only on the internal antenna (Figure 3.1.3) These 50 metres are an important indicator when localisation in an urban environment has to be performed.

This experiment shows that two LoPy devices would be able to communicate at up to 150 m distance. For localisation of moving vehicles within city environment using the WiFi fingerprinting technique, a radio map has to be developed. Estimation of the location of the vehicles can be provided by the system, based on different RSS values, which are obtained after some learning period of the system. This means that the RSS values have to be measured continuously at multiple known locations. Considering the range of the LoPy's WiFi communication the network of known WiFi access points has to be of high density, which can't be achieved easily in an urban environment.

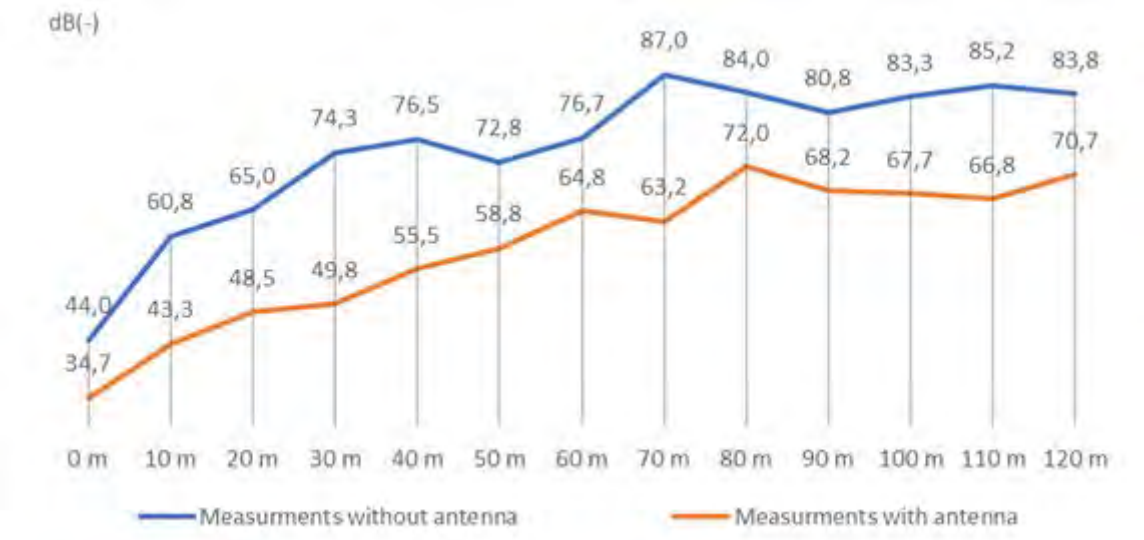


Figure 3.1.3: Comparison of RSS values with and without antenna (own work)

Relation RSSI distance

The results shown in Figure 3.1.2 and 3.1.3 indicate a relation between distance between sensors and their received signal strength. These results have been obtained under LoS conditions but still prove a possibility for trilateration using RSS for WiFi. However due to hardware and time constraint this has not been considered for practical implementation. However, this section will theoretically discuss the research needed to validate the possibility of a practical application of WiFi RSSI trilateration.

Parameswaran et al. (2009) describe the theory behind the relationship between RSSI and distance. However their conclusion is that the lack of reliable stability of the indicator does not allow for an accurate indication of distance. The data collected in the WiFi benchmark indicates differently. This variance in expectancy of the capabilities could be a result of the use of different hardware for the research. Different microprocessors or wireless communication methods are affected differently regarding the signal to noise ratio, signal attenuation, and signal propagation (Tsui, Chuang, and Chu, 2009). The variations that exist between devices and the signals derived therefrom require continuous adjusting. A self-learning unsupervised method as developed by Tsui et al. (2009) would suit these adjustments but an approach with all similar devices could benefit from simpler algorithms for the localisation or positioning of a node in a network.

Based on this knowledge the data, collected in the range benchmark of the LoPy devices, has been evaluated. As all devices are to be assumed equivalent and under LoS conditions, hence a self-learning approach is not considered. In Figure 3.1.4, a linear least squares regression has been used to describe the relationship between the measurements. These equations have been evaluated by their coefficient of determination (R-squared), assuming the relationship between the RSSI and distance to be significant. Equation 3.1.1 is the result of the regression on the data without antenna, where equation 3.1.2 is the inverse of this. This regression resulted in a R-squared of 0.6676. Equation 3.1.3 and 3.1.4 describe the same relation for the measurements with an R-squared of 0.7958.

$$RSSI = 0.2652 * Distance + 59.032$$

(eq. 3.1.1)

$$Distance = \frac{(RSSI - 59.032)}{0.2652}$$

(eq. 3.1.2)

$$RSSI = 0.2762 * \text{Distance} + 42.193$$

(eq. 3.1.3)

$$Distance = \frac{(RSSI - 42.193)}{0.2762}$$

(eq. 3.1.4)

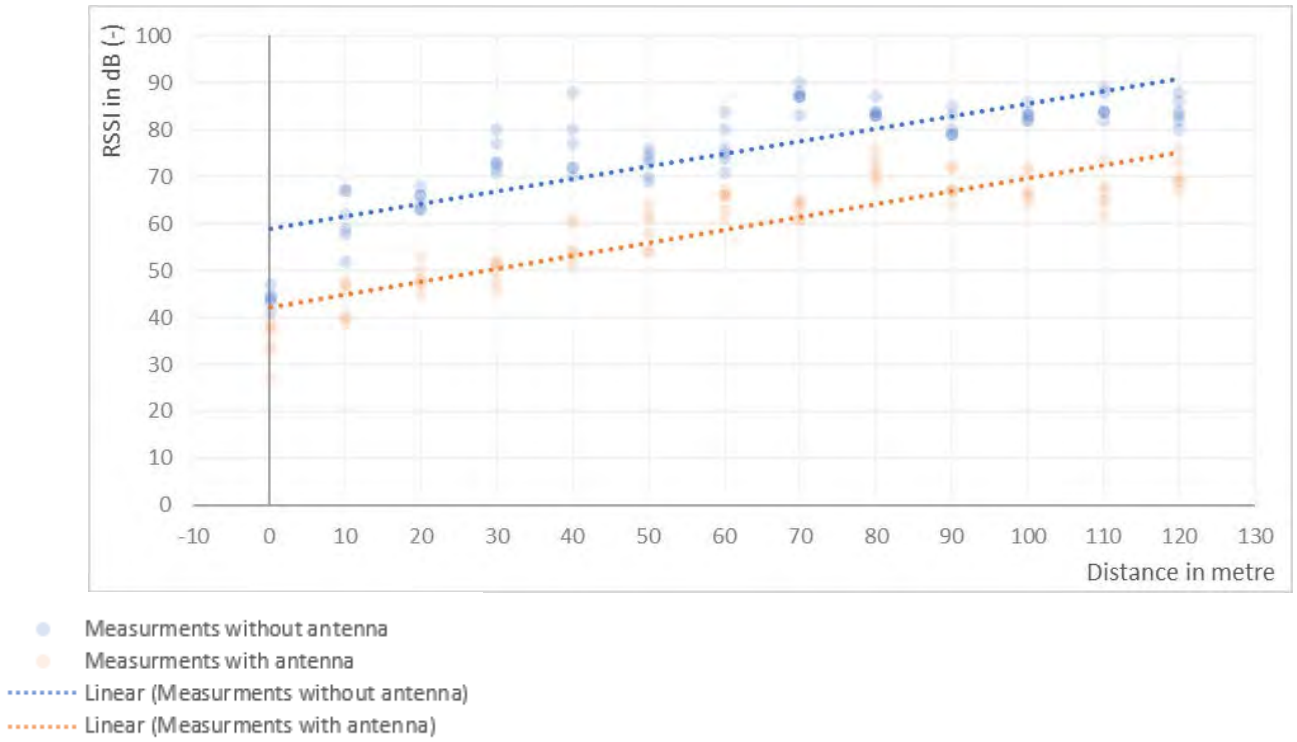


Figure 3.1.4: Linear relationship between RSSI in dB and distance in metres (own work)

To further research the relation between the measurement a 2nd order polynomial has been fitted using least squares regression. A drawback of a second order polynomial is the increase and decline described by such a formula. However this might not be applicable if the part of the polynomial used for describing the measurements is just comprised of the growth, as the range might be a limiting factor. Figure 3.1.5 and equations 3.1.5, for the measurements without antenna, and equation 3.1.6, for the measurements with antenna, show the relationships between the measurements and the regressions. The R-squared have improved over the linear regression with 0.8416 and 0.8984 respectively. However there is ambiguity introduced as both regressions start to decline before the range cut off at 120 metres.

$$RSSI = -0.0041 * Distance^2 + 0.7565 * Distance + 50.001$$

(eq. 3.1.5)

$$RSSI = -0.003 * Distance^2 + 0.636 * Distance + 35.579$$

(eq. 3.1.6)

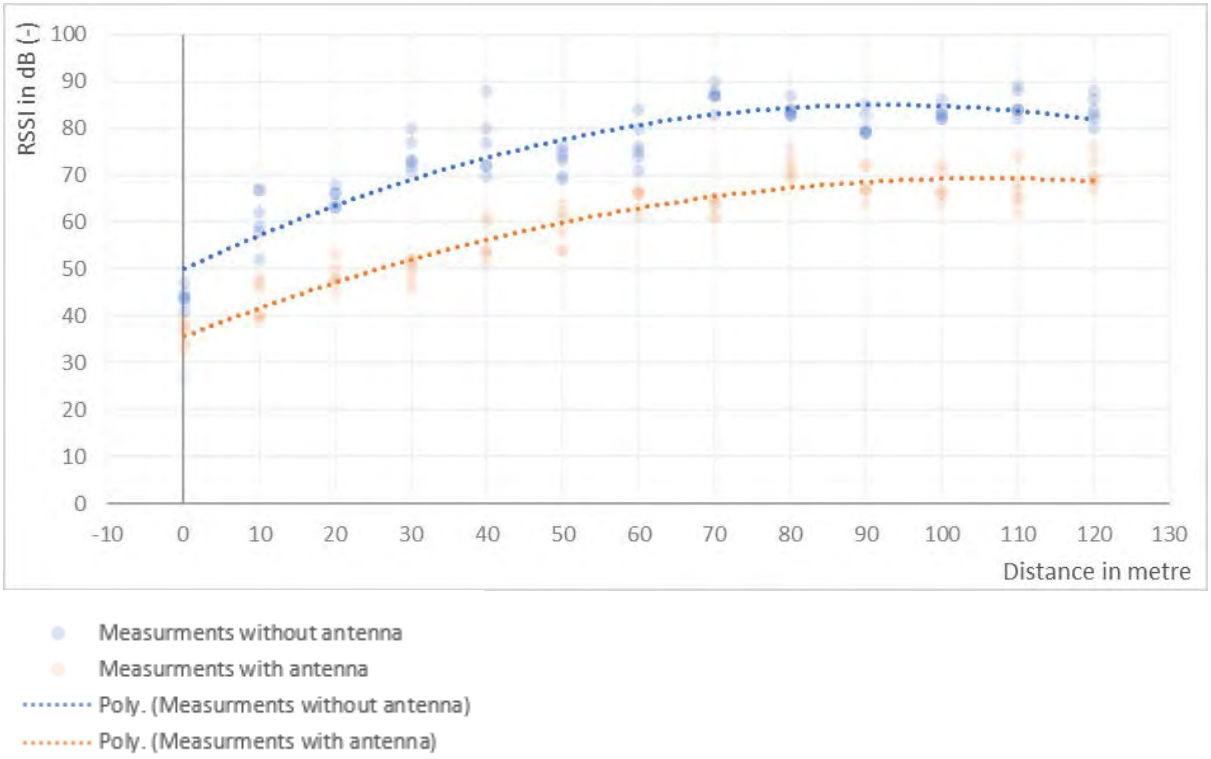


Figure 3.1.5: 2nd Order polynomial relationship between RSSI in dB and distance in metres (own work)

From the least squares regressions two conclusions are drawn. Firstly, it can be concluded that there is a direct relation between the measured RSSI and the distance. Secondly, the addition of antennas to both the receiver and transmitter of the signal have a range increasing and stabilising effect. Both functions with antennas have a smaller offset which indicate the RSSI is higher. Besides this, the R-squared indicator is significantly higher for both tests with antennas, indicating a better model. Even though this relationship is described best by a 2nd order polynomial, this function is not usable in a real world application. The added ambiguity would decrease the confidence in the results. Therefore a linear approximation has to suffice. As the R-squared of the regression, based on the measurements with antennas is 0.7958, 20.42% of the variance is not explained by the linear model. Due to hardware limitations, the extra cost associated with the establishment of a network and localisation technique based on the RSSI measurements, and the 20.42% inaccuracy introduced in near-perfect LoS conditions, the RSSI-distance relationship is not considered for a practical implementation in this research.

An effective implementation of the RSSI-distance relationship can be further researched in which a deeper statistical analysis of the link should be considered. A R-squared predicted could indicate whether the model has been overfitted or is actually describing the situation correctly. Multipath, variance in hardware and non-LoS conditions should be considered as well. In this case, two possibilities should be considered: a self-learning algorithm and a least squares analysis as performed by the Move3 software from Sweco[10]. For this a broader setup would be required in which multiple base stations are developed of which the exact location is known. From here a network, comparable to a geodetic network, could be developed. With connection indicators, this network could be calculated and validated from which the localisation could be performed.

3.1.2 LoRA test

The second localisation technique which was considered is built upon the time difference of arrival (TDOA) notion. In other words, the moving vehicles can be localised using the difference in time of arrival of the LoRa signal between at least 3 gateways. The specification of the LoRa communication implies time restriction for sending data via LoRa, which leads to the conclusion that the frequency with which vehicles could be localised is restrained.

[10] *https://move3software.com*

In order of this possibility to be proved, two Nano-gateways were built and registered in “The Things”[11] network. The sending device was registered as well, which provides the ability to monitor the payload, which is sent. The Things Network provides information like RSS, a time when data is received, the id of the gateways and their location, which is provided by the person who deploys it.

The experiment was performed in two manners. Firstly, a private gateway was used as a receiver, while another device was sending messages at increasing distances (approx. 100m increase). The second test was performed using two private Nano gateways with the same location, for which the difference in the time of receiving was compared.

The first part of the test gave unsatisfying results – only the first message was received by the two gateways – our own and existing one, located on the TU Delft campus on the tallest building (EWI) (Figure 3.1.6). The following message at approximately 250 m distance with no line of sight (LoS) was received only by the EWI gateway.

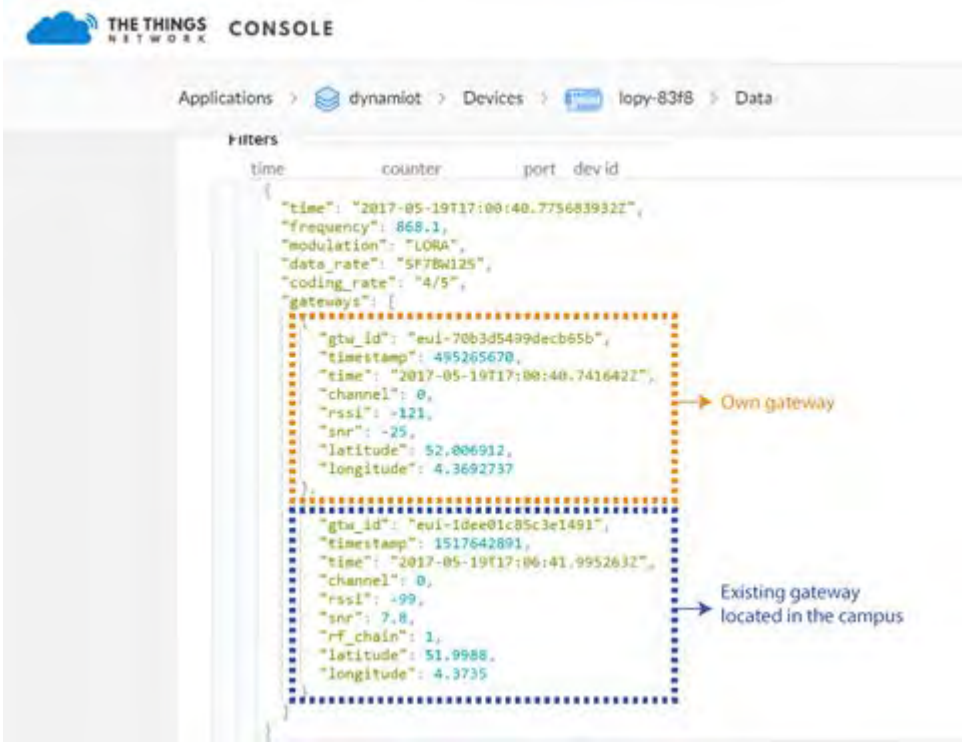


Figure 3.1.6: Information about received LoRa messages (own work)

The second test was performed by placing two Nano gateways at the same location and again a message was sent via LoRa, which gives the time of arrival for both gateways. The time provided by the things network is in accuracy to milliseconds but the clocks of the gateways are not synchronized, which introduces an error up to a few kilometres (Figure 3.1.7, Table 3.1.1). This means that in order to use the TDOA method for localisation, the clocks of the built gateways have to be synchronized. The theoretical time accuracy, which is needed is microseconds, which corresponds to 300 m positional accuracy, or nanoseconds allowing centimetre level accuracy (Table 3.1.1). The Things Network dashboard provides time in accuracy to milliseconds, which is not sufficient for localising moving vehicles in cities.

The limited amount of documentation and research on localisation using LoRa, as described in chapter 1, combined with the inaccurate time reading from the gateways disqualifies LoRa from consideration in a feasible localisation methodology.

[11] *https://www.thethingsnetwork.org/*

Unit of Time	Part of a second	Corresponding accuracy
Millisecond	1/1 000	300 kilometre
Microsecond	1/1 000 000	300 metre
Nanosecond	1/1 000 000 000	300 millimetre

Table 3.1.1: Time difference of arrival for two Nano gateways (own work)

Figure 3.1.7: Time difference of arrival for two Nano gateways (own work)

3.1.3 GPS

In order to determine the quality of the measurements taken with the GPS module that is used in the sensor system, a static test was carried out to get a general overview on these measurements. The outcome of this test gave insight in the reliability of GPS for quality control of other localisation techniques. According to the documentation of the GPS module, the horizontal positional accuracy (measured as circular error probable (CEP)) should be 2.5 m (U-Blox, 2011). The calculation for the CEP is given in equation 3.1.7 (NovAtel, 2003).

$$CEP = 0.62\sigma_y + 0.56\sigma_x \; [m]$$

Eq. 3.1.7: CEP calculation (can be used when σy/σx > 0.3) (NovAtel, 2003)

The test was set up on top of the TU Delft library, a place where the LoS is mostly unobstructed. It can be expected that the research area, the city centre of Delft, the LoS will be obstructed significantly. The result of this test can therefore been seen as the maximum accuracy. The measurements were taken over a time period of half an hour, during midday. For the processing it is assumed that the measurements are distributed via a standard deviation.

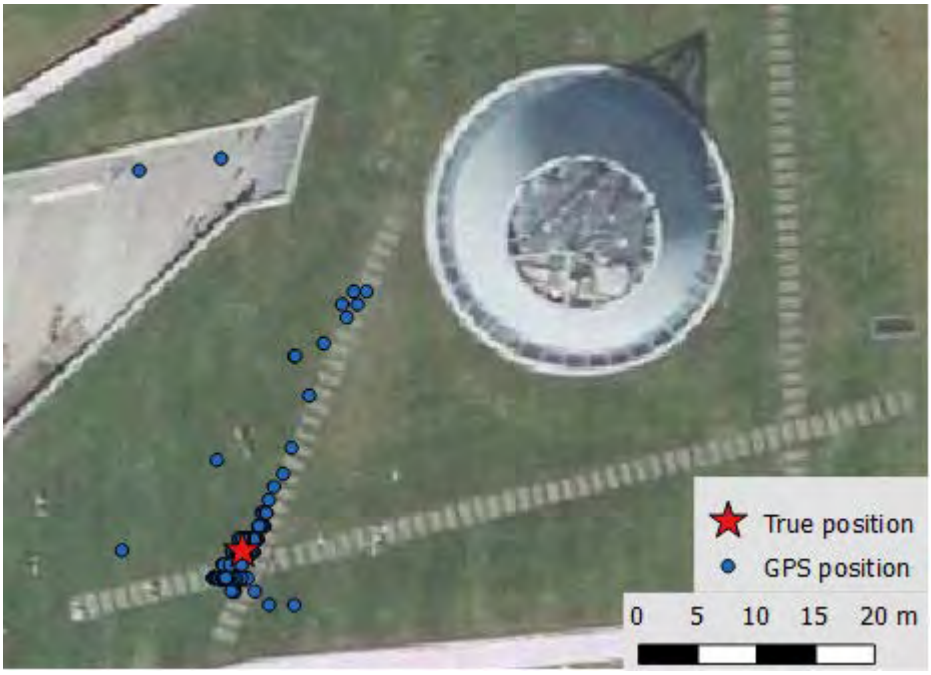


Figure 3.1.8: distribution of GPS measurements (own work, imagery from PDOK (2017))

In Figure 3.1.8 the distribution of the measurements is shown. They were converted from the WGS84 Coordinate Reference System (CRS) to the, local Dutch, Amersfoort/RD New CRS, because this uses coordinates in metres, which makes processing of the results easier.

The measurements were done at exactly the left upper corner of the tiles crossing. This is a point that can easily be found on aerial photographs, and therefore can also be marked as the ground truth position. For every measurement the standard deviation of the distance between its position and that of the ground truth was defined. This was done in both the x and y direction, which was later converted to the overall accuracy using the Euclidian distance. The CEP is calculated using equation 3.1.7. A summary of the results is shown in table 3.1.2.

σx	2.0 m
σy	4.9 m
CEP	4.2 m
σ	5.3 m
2σ	10.6 m

Table 3.1.2: Standard deviation of GPS position (own work)

The CEP calculated for the obtained measurements is 4.2 m, which is 1.68 times as much as the manufacturer claims. 95.4% of the measurements fall in the range of 10.6 m, which is the accuracy that is taken into account for the rest of the research. This range is visualised in Figure 3.1.9. It is expected that with an accuracy of 10.6 m the GPS measurements will be sufficient as quality control for other localisation techniques, as these are expected to give less accurate results.

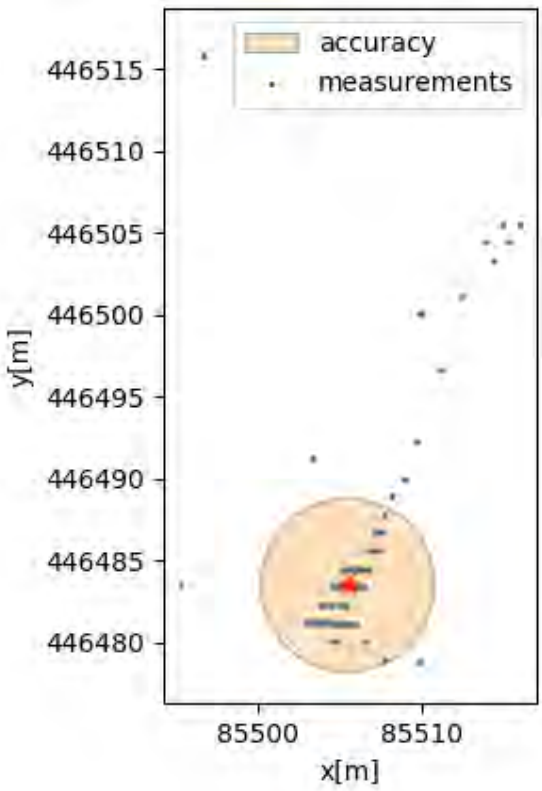


Figure 3.1.9: Standard deviation (2σ) of GPS measurements is 10.6 m. (own work)

3.1.4. Power consumption

To be able to determine how often batteries are to be changed and also to prolong their working time, multiple measurements on the average current draw were carried out. In the following sections the background, setup of the experiments and their results will be explained. In all of the experiments, the batteries in consideration have capacity of 5000mAh and voltage of 3.7V (4.16V when fully charged). The lithium ion battery discharge characteristics, explained in the chapter “Technical implementation details”, are consciously overlooked. This is because looking at typical discharge characteristics (Figure 2.4.1), it is clear that the batteries would be functional up to 90% discharge level and thus further optimisation was not necessary.

Documentation check test

The initial test for the energy consumption was set in place to double check whether the documentation of features indicated it correctly. As this was more an indicative test, there was no specific code written for this purpose and the default code for the MCU was used. This indicates that there is continuous power supply to the modules (3.3V) and the built-in LED is working in pulses. In the table below, the results of the experiment are presented (table 3.1.3). The expected values have been gathered from the module documentation, whilst the measured values were gathered using power supply DP1308A and the perfboard solution, produced by the project team.

From this table, it can be concluded that the values for the sensor modules have been indicated correctly, whilst the documentation for the MCU did not represent real values. The full system, the LoPy, provided overall insufficient information on the power draw. This also indicates, that if the system were to measure for five working days non-stop, the average power consumption would have to be reduced almost four times (to 42mA). For a single day (8 hours) use that would not be necessary, as the average draw could be as high as 437mA. This test also shows that the most optimization has to happen in the code for LoPy, as it draws the most current, and secondly the GPS module.

Part	Expected	Measured
AM2302 (Temperature and humidity sensor)	15 µA (dormant)	<1 mA
U-blox NEO-6M (GPS)	67 mA (active)	67 mA (maximum power)
MAX9814 (microphone)	3.1-6 mA	3 mA
LoPy (MCU ESP32)	37 mA[12]	105-160 mA
All elements	~110 mA	150-300 mA

Table 3.1.3: The results of the documentation check test (own work)

Power consumption with LoPy optimization
The second test for the power consumption was carried out to check the efficiency of LoPy's deep sleep mode. It was to be initiated for 3 minutes after collecting the information (code in Appendix I). It is important to note, that the mode is still under development, as the flash chip and DC-DC switching regulator remain in high performance mode after the deep sleep request (Pycom, 2017). This implies that the current in deep sleep mode continues to flow to the sensor modules. Even with fully functioning deep sleep mode, the current draw would be theoretically reduced by 9mA.

The experiment for the power consumption in this case was performed on breadboard equivalent to the perfboard solution, using a USB Safety Tester J7-t, which has the accuracy of 10mA. The results during the sleep mode were that the platform continued to draw 50-70 mA, 25uA of which should be drawn by the LoPy, according to the documentation. During active mode the current was 130-140 mA. Taking the maximum values, the average current consumption is thus 87.5 mA, allowing a battery life of 40 hours or 2 measurement days.

Power consumption with GPS optimization
As it was necessary to optimize further, the group considered focusing on the GPS module energy use. It was chosen to inventorize the power-save, cyclic and on/off modes, as described in the chapter "Technical implementation". The experiment for the power consumption in this case was performed on the perfboard prototype, using an analog multimeter Unigor 3S. The experiment was based on code, which sends CFG-RXM message, initializing different power modes, as described in u-blox 6 receiver description and protocol specification (code in Appendix I). As mentioned before, the GPS module uses 67mA while in active mode. According to chip specification, with the cyclic mode on, it should use merely 11mA. However, according to own measurements, that only reduced the power consumption to ~50mA. With on/off mode, this was further reduced to 35mA on average and 25mA while most of the chip functionality was off. Furthermore, it minimized the overall system power use to ~40mA while on sleep mode, according to USB Safety Tester J7-t. This results in average 65mA current consumption, allowing 53 hour battery life or 3 measurement days (8 for the first, 24 for the second, 21 for the last).

Alternative platform
An alternative to the proposed system, would be one without a GPS module. That would reduce the duration of the active code to around 5 seconds, as with the on/off implementation it takes at least 26 seconds to get a fix. During this period the system would only need to collect the temperature and humidity readings, collect and average the sound pressure levels, make a snapshot of the 3 strongest MAC addresses and transmit the LoRa message. With the deep sleep function fully in place, the power use during active mode would be maximum 314 mA with all modules and network peripherals active. During sleep mode, it would theoretically drop to 6.5 mA. To sum it up, the average current draw would thus be 22.68 mA, allowing over 10 days of measurement with the 5000mAh battery.

Conclusion and recommendations
All in all, the power consumption tests have clearly indicated which of the component's performance had to be optimized: primary power consumption for active code comes from LoPy, whilst after optimization, GPS

[12] *According to the LoPy and MCU ESP32 documentation: with active WiFi and MCU in writing mode.*

remains the largest consumer during sleep mode. The current level of optimization allows a maximum of three days of measurements, not taking into account the battery discharge characteristics. The next step in this would be implementing the deep sleep shields, which are supposed to overcome the pin charge issue and should be delivered after the measurement phase of DynamIoT project. This would allow to further decrease the current draw in sleep mode by 9mA. However, to fully cut the power to the external modules, including GPS, an external switch or transistor, controllable by the MCU would be necessary. In this case, one should consider the pin state during the deep sleep mode as a potential issue. If the GPS module was not used and the deep sleep would be fully functional, the system could possibly function for over 10 days. This could be extended even further using solar panel solutions.

3.2 Casing

In this paragraph all test are described that were carried out to test the overall build and durability of the platform system. Moreover, the casing will be tested for its influence on the sensor results. At the end of each section, recommendations on improvements will be provided.

3.2.1. Durability tests and deployment

After producing the prototype, it was decided to check the durability of the connections. As potentially weak points, the following were identified:

- JST breakout board solder connection
- U.FL antenna connectors
- Antenna connections with the box
- Female header - LoPy connection[13]

The initial test was carried out while biking through the city centre of Delft with the platform fastened to the bicycle. Having the code with all of the components activated and writing to the flash memory would allow the user to see whether any of the components stopped working and at which point that happened. After running the test it was clear than all of the components continued working throughout the duration of the experiment.

Nevertheless, after deploying the sensor platforms on the Tuk Tuks and boats during the data measurement stage, multiple connector flaws caused the systems to stop working. The first issue with the system was caused by the misalignment of the female headers used for the LoPy unit. The previous test prototype used an unbroken row of headers, whilst the reproductions were using two parts to make up a row. In all of the units this would cause the MCU to pop-out in a short period of time or not give contact to the unit itself. This was temporarily fixed by taping the LoPy to the perfboard. Even so only one platform remained fully functioning. This required all of the headers to be changed to unbroken row solution and that did not cause any further issues.

The second malfunction was due to the difficulty removing the battery from the JST connector on the board. After multiple days in use, the JST breakout boards moving would cause their solder pads to separate from the perfboard. This was due to the click-mechanism of this connector type and thus in future situations this connector type should be avoided or a solution without constant battery removal should be implemented.

The third platform malfunction was WiFi and LoRa antenna connections becoming loose during deployment on the Tuk Tuks. To reduce this, not only the platforms, but also the antennas were fastened to the vehicle. As this might not be possible in alternative vehicle situations, it would be preferable that the antenna is more stably fastened to the box.

The last platform malfunction was caused by the earlier mentioned U.FL antenna connectors detaching. It was observed only once, yet to prevent this a simple tape solution would probably be sufficient in comparable measurement setups. If an alternative to LoPy system was investigated, it would be beneficial to look for a solution with screw-based antenna connector as, for instance, the reverse polarity SMA (Figure 3.2.1).

[13] *Initially it was not a single row, but 6+8 solution*

Figure 3.2.1: Reverse polarity SMA adapter (Online Kabelshop, n.d.)

Conclusion and recommendations
To sum up, even though the original durability test returned no issues, the deployment of the replicated platforms was the opposite. The biggest issue was caused by misaligned headers, which had to be re-soldered. The same had to be done to the JST connectors on the boards. Therefore, another type of connector should be used in future work or the battery not removed during the deployment period. Another recommendation would be to incorporate a mechanism to stabilize the antennas on the casing instead of fixing it to the vehicles.

3.2.2. Casing bias tests

One of the biggest concerns during the project was about the bias that the partially open box would introduce to the measurement equipment. As only the temperature and humidity sensor would produce meaningful data for the final solution, this was chosen to be inspected. The experiment equipment was a modified sensor platform, with one sensor inside of the box and one outside. The measurements would be done 4 different setups:

- Dynamic, sunny day
- Dynamic, cloudy day
- Static, in the sun
- Static in the shade

The measurements in both static and dynamic setups were taken every 20 seconds, writing both temperature and humidity results. For the dynamic tests, the platform was fixed to the back of a bike and the same route, of around 30 minutes, was taken for each of the tests. These were carried out in two different weather conditions, sunny and shade.

As seen in the results (Figure 3.2.2, 3.2.3), the expected greenhouse effect was not evident neither in sunny, nor in shade conditions. However, the box did work like a low-pass filter, which smooths out fast humidity and temperature fluctuations. This is beneficial for the chosen use case (single measurement every 3 minutes), as it decreases the chance of outliers.

In the most problematic situation - static, sunny - both sensors indicated significant temperature rise. This sheds light on the fact, that to be able to measure ambient air temperature, the sensor platform should be out of direct sunlight. Nevertheless, it can be observed, that this is less of an issue while measuring on a dynamic, sunny conditions. This can be attributed to the increased airflow, which constantly cools and exchanges the air inside the box, bringing it possibly close to ambient air temperature. To be able to precisely determine the accuracy of this, a third sensor would have to be introduced, placed at a location in close proximity, optimized for ambient air measurements. Such a test would provide further insights on sensor data quality related to the chosen spot for fastening on the vehicle.

Conclusion and recommendations
All in all, the chosen box solution has sufficient air-exchange and thus provides reliable measurements, under the condition that it is not placed in a direct sunlight with little airflow. Nevertheless, the efficiency of placement on the vehicles is strongly advised to be tested in future work, as this has major influence on the ambient temperature and consequently relative humidity readings.

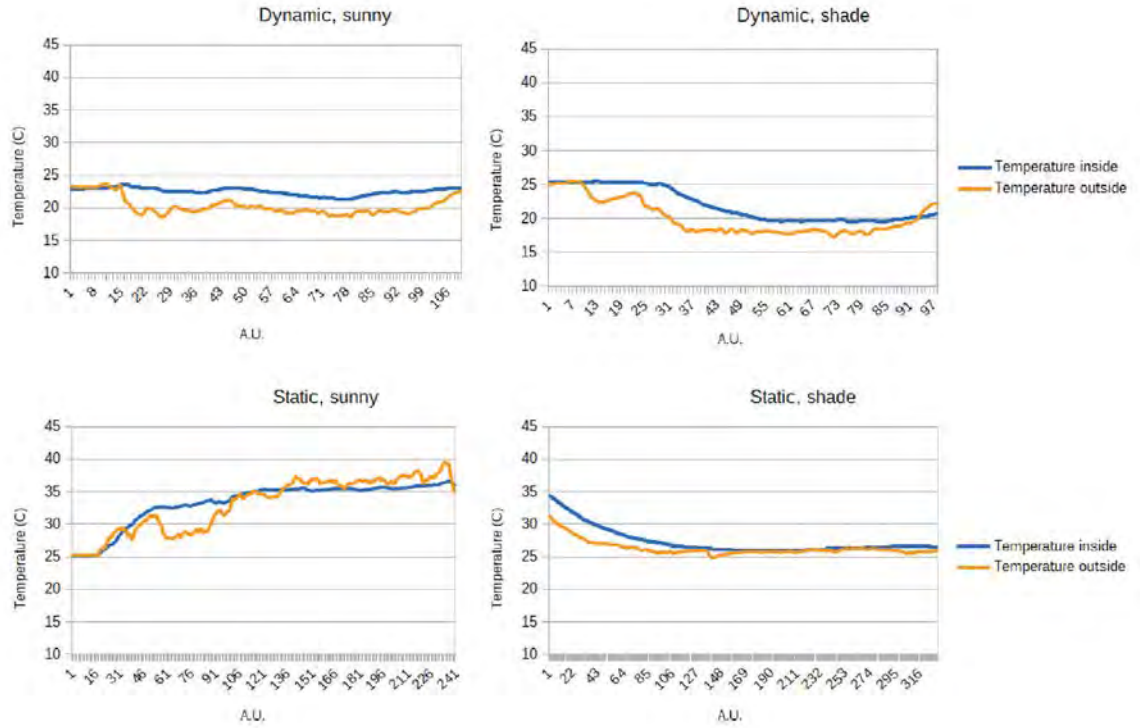


Figure 3.2.2: The temperature results of the box-bias test (own work)

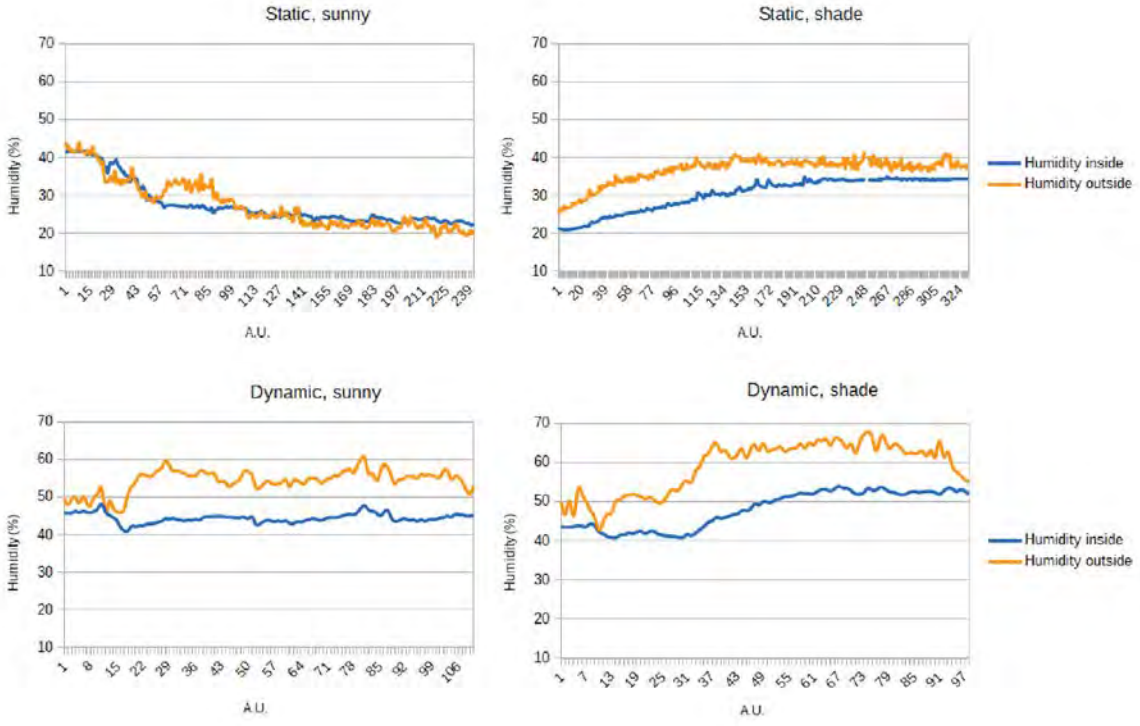


Figure 3.2.3: The humidity results of the box-bias test (own work)

4. Methodology

The following methodology covers the process of data capturing, processing, analysis, presentation and quality control. These steps are recognized as essential in the process of answering the main research question: ***"To what extent can near real-time spatiotemporal data be obtained using a dynamic sensor network in an outdoor urban environment?".*** The important characteristics to be covered are: the continuous gathering of dynamic sensor data in urban environment, complemented by information about the location of the devices at different moments in time.

4.1. Data gathering

The process of collecting data consists of the transmittal of relevant sensor data for the urban environment to a spatial database. In order for this data to be associated with a geographic location and later to be processed as continuous geographic field, localisation will be performed subsequently. Based on the theoretical and practical research in chapters 1, 2 and 3, the setup for data gathering is defined.

For the gathering of meaningful data, eight sensor platforms have been developed. The setup of which is defined in chapter 2.3 and 2.4. It consists of the LoPy platform, to which a microphone, temperature and humidity sensor, and GPS are connected. All of this is powered by a 5000mAh battery. The casing, in which all are housed, consists of two parts, one open to some of the elements, while the other is completely sealed. This to prevent unnecessary corrosion and safeguard the continuous operation of the sensor platform.

To be able to dynamically collect environmental data, the sensor platforms have been attached to vehicles continuously moving throughout the city of Delft. These are 6 electric Tuk Tuks from City Shuttle Delft and 2 electric tour boats operated by Rondvaart Delft. The data has been gathered over 7 consecutive days from June 8th 2017 up until and including June 14th 2017. A full week cycle of data gathering has been systematically covered. As the boats are not moving at night and are not guarded, the sensor platforms have been removed at night to avoid theft. This means the data could be gathered between 11:00 and 18:00. For similar practical reasons, the measurement times for the Tuk Tuks were between 10:00 and 18:00. The vehicles are all supposed to move along predefined routes, however the results show that deviations occur.

All eight sensor platforms are equipped with the same code. (Appendix I, J, K) This code is designed to measure environmental data and data necessary for the localisation of the platforms. To ensure sufficient operation time, the data is gathered periodically. After every measurement period, in which three MAC addresses, the respective RSSIs, temperature, humidity, noise-levels and GPS coordinates are collected, is the LoPy switched for three minutes to deep sleep mode[14]. This periodic approach ensures sufficient running time for the sensor platforms on one battery charge to measure a full day. After a measurement cycle all data is transferred to the database using LoRa. Due to the limitations of this communication technique, described in previous chapters, the message has been constructed carefully. The data gathered is sent using 30 bytes, these consist of:

- 3 x 7 bytes for a MAC address (6 bytes) and a RSSI in dB (-)
- 1 byte for temperature
- 1 byte for humidity
- 1 byte for noise level
- 6 bytes for the GPS location.

The MAC address or Basic Service Set Identifier is comprised of 6 bytes (IEEE 802.11 Working Group. 2010). The absolute value of the RSSI is a value between 20 and 100, therefore can be stored in 1 byte. Temperature and humidity are always values between 0 and 100 in the month of June and are therefore both stored in 1 byte. The noise level is in dB and even though the data is not considered for this research, this adds another byte to the message. The GPS coordinates are too large to add without edited to the message. The coordinates are retrieved in WGS84, latitude and longitude in degrees and decimals. The minimal required decimal accuracy is 4 which would result in an precision measurement of 12 metres, while

[14] *The deep sleep mode turns off the central processing unit and all the peripherals (including network interfaces)*

an extra decimal would reduce this to 1.2 metres (Arlinghaus and Kerski, 2014). For the effective transferral of the GPS data a minimum of two times 6 numerical decimal characters is needed. To achieve sufficient decimals for correct positioning without overfilling the LoRa message a reference point has been added to the map, on location 51.9627 N 4.3161 E (Figure 4.1.1) As the movement extent from the vehicles is known only part of the coordinates are needed, therefore the coordinates of the reference point are subtracted from the measurement locations. Leaving only the information behind the decimal point, by multiplying this by one million, an integer is generated and split to be added to the LoRa message. This compresses the information to two three-byte message parts, one each - longitude and latitude. To avoid the code crashing, only coordinates within the bounding box of Delft are considered. Before the LoPy module returns to the deep sleep mode, it sends the message via the LoRa network of KPN to the database, where processing, as described in 4.2, is executed.

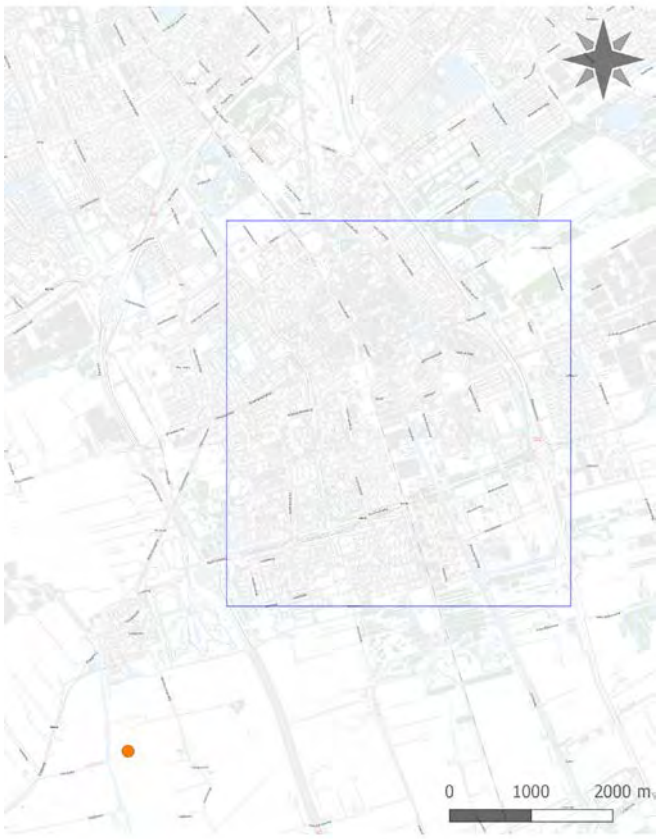


Figure 4.1.1: Reference location and research area (own work, map from PDOK (2017))

Identical code was used during the deployment of the sensor platforms, however continuous operation was not achieved. Some of the devices did not run the full day or not at all on some days. This mostly can be attributed to the issues mentioned in the chapter “Benchmark tests” and also the instability of the code with the GPS power-saving mode. The latter was likely caused by issues associated with the proprietary NMEA message requests, which would sometimes crash the code even with all precautions in place. Moreover, this mode increased the chance of the GPS not getting a fix, as it would not run continuously. Therefore, during the final week of measurements it was decided to use the original, stable code with continuously functioning GPS module. All in all, it can be concluded that the operation of the sensor platforms was sufficient for this pilot research and provided sufficient data for analysis.

4.2. Processing

The post-processing of the collected data is essential for the cleaning and validation of the aggregated information. The refinement of the data consists of removing invalid records resulted from system failures. These could be invalid locations, outliers or inconsistent sensor values.

4.2.1 Data flow and message analysis

After having access to Node-RED instance, tutorials provided by the official page were followed as part of the familiarization procedure to create trial data flows. JavaScript functions can be created within the editor using a rich text editor (Node-RED, 2017). A built-in library allows to save useful functions, templates or flows for re-use (Node-RED, 2017). Considering the self-explanatory part of Node-RED, it could be said that while there is a short explanation for every node, some of them are not clear at all.

Using a LoPy platform, which is a MicroPython development platform supporting LoRa, WiFi and Bluetooth networks, messages were sent through the LoRa developer portal which is run by KPN company. The aim of the KPN LoRa developer portal is to enable users to quickly connect their LoRa device to the KPN LoRa network (KPN LoRa Developer Portal). In the LoRa developer site a destination URL has to be specified which is the ‘’Locator’’ that the messages will be send. ‘’Locator” was instantiated in a http request node having as a URL: https://geo1101.bk.tudelft.nl/iot/dynamic/..... , requesting for data. In order someone not to receive error messages from the developer portal, one should wire the http request node with a http response node. Messages could be presented by wiring the http request node with a debug node. The result is displayed on the debug column on the right side of the interface. Having done this, a great number of variables are contained, but not all of them are useful. Figure 4.2.1 shows the form of a message that sent by the LoRa developer portal.

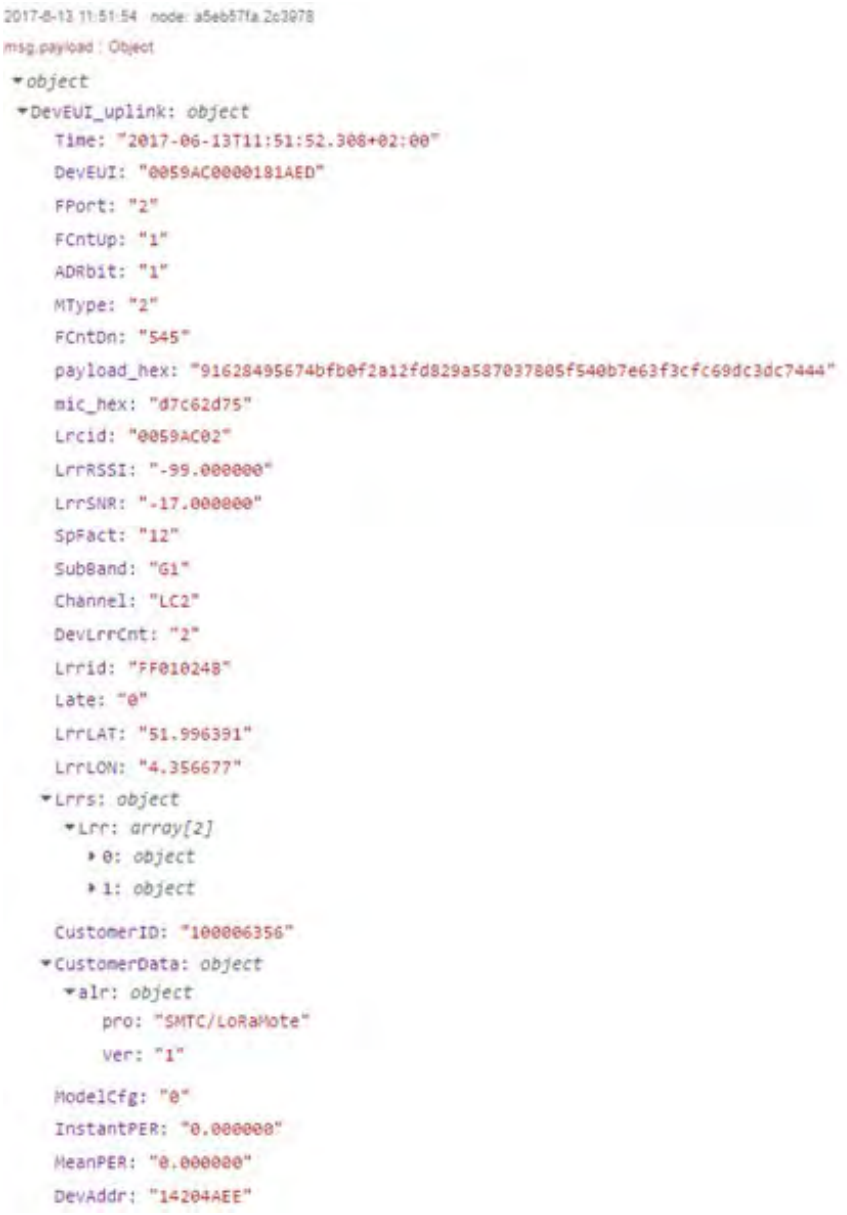


Figure 4.2.1: Structure of the message received by LoRa developer portal (own work)

Strict statements in Javascript should be written in order to isolate, create, store and retrieve variables from the existing schema. Figure 4.2.2 provides a clear indication of the actions followed to achieve data flow.

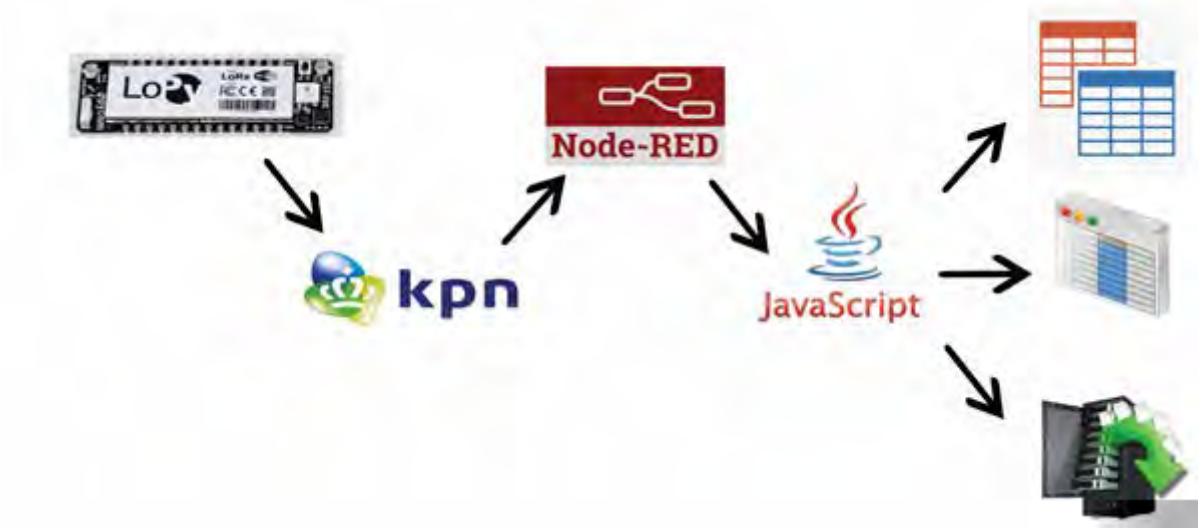


Figure 4.2.2: Sequence of actions followed to achieve data flow (own work)

Having decided the final schema, the database was populated with meaningful information from the decrypted message. In order to decrypt the payload, a lora-decrypt npm package was added to the virtual machine (see appendix L). It is a node container for decrypting LoRaWan payloads (lora-decrypt, 2017). It is used in a function node and has the following form: lora_decrypt(payload_hex, sequence_counter, key, addr). Using a LoPy it is possible to send only 54 bytes as payload. Considering the limit of the bytes, the payload should be modified in a way that each and every character of it to be meaningful (see appendix M). Table 4.2.1 matches the payload' s digits with their meaning.

Digits	Represent
0-5	*mac address 1*
6	*strength of mac address 1*
7-12	*mac address 2*
13	*strength of mac address 2*
14-19	*mac address 3*
20	*strength of mac address 3*
21	*temperature*
22	*humidity*
23	*sound*
(24-26 / 1000000.0) + 51.9627	*GPS latitude*
(27-29 / 1000000.0) + 4.3161	*GPS longitude*

Table 4.2.1: Meaning of payload' s digits (own work)

After decrypting the message, an SQL query was written containing the above information combined with longitude, latitude and accuracy derived from Google API, as described in technical implementation, and sent to an SQL node in order to populate the table with the meaningful information.

4.2.2 Processing for localisation purposes

To be able to process the collected data for localisation purposes, QGIS software was used. Firstly, the routes of the vehicles were plotted in QGIS. The network of routes defines the extent of the area in which measurements are expected. This area was subdivided to a square grid, defined of 40 by 40 cells, each having a size of 50 by 50 metres. Figure 4.2.2 shows the vehicle routes together with the created grid.

The cell size was chosen in such a way that each one could be identified from another, but still to maintain a granularity high enough to provide for a useful analysis of the sensed city later on. The cell size was based

on the range of the WiFi signal in an urban environment and the expected accuracy of the GPS fix, which is lower than the cell size, The choice to build a grid of 40 by 40 cells was made to assure that the area covered by the grid enfolds the full extent of the route, while maintaining a straightforward numbering of grid cells by using a multitude of 10.

Figure 4.2.2: Map showing vehicle routes and grid over city of Delft. (own work)

The data was exported from the database into a csv file which was loaded into QGIS as a set of points based on the GPS measurements. The attributes of these points containing all WiFi and sensor data. The measurements which were outside the extent of the grid were removed from the layer. These outliers could be caused by premature GPS fixes and can be recognized by their position, not following an existing road or route. Nevertheless, there were readings, falling outside of the grid due to the vehicle deviating from its route. These measurements can be recognized by the clear path along a road they follow. Later on in the measurement period, the code of the devices was set such that the location measurements outside of the extent of the grid were excluded before being sent.

In addition, a python script was used to count the number of measurements per grid cell to be able to colour them accordingly and get more insight into the spread of measurements (See Appendix N). In Figure 4.2.3 the map of Delft is shown with a grid over it coloured according to the number of measurements per grid cell used for the radiomap. Figure 4.2.4 shows Delft with a grid, describing each cell's likelihood to have measurements. The cells along the route have the highest probability to contain measurements. directly along the routes are expected to have some measurements as well, due to GPS accuracy issues.

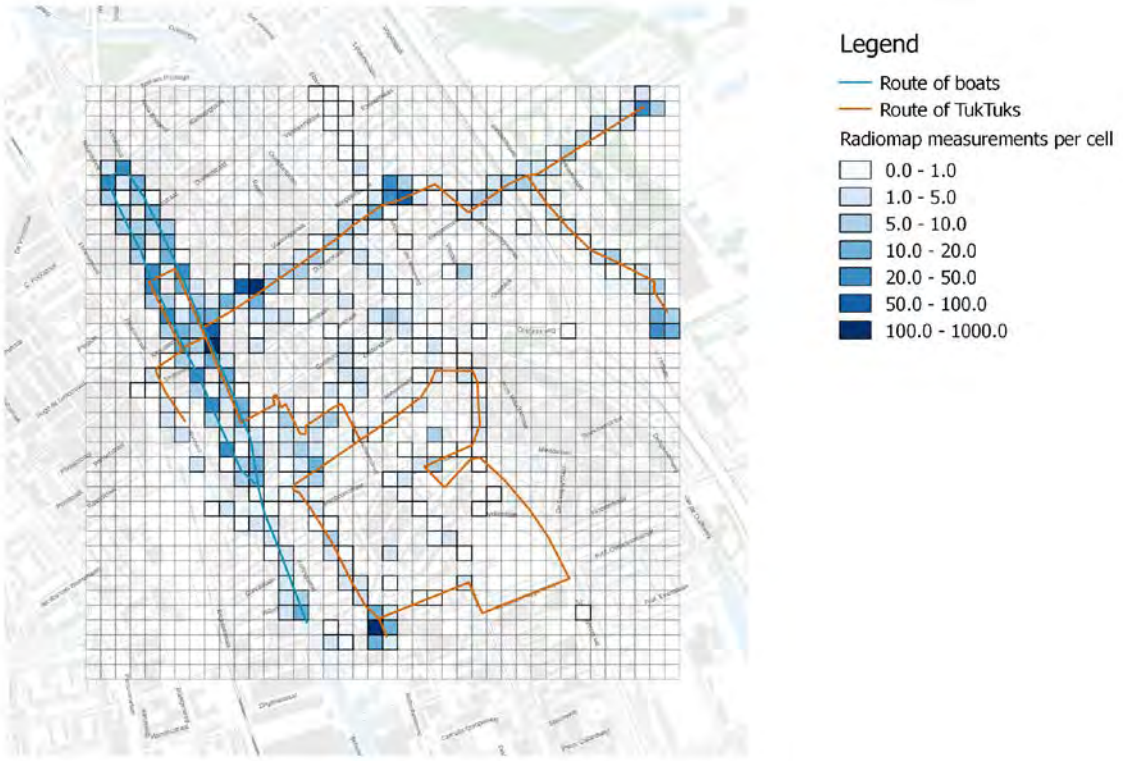


Figure 4.2.3: Map with grid coloured according to number of measurements used for radiomap per grid cell. (own work)

Figure 4.2.4: Map with grid cells coloured according to likelihood of measurements. (own work)

4.3. Analysis

The following section describes the techniques used for the analysis of the data. Firstly, the obtained location of the measurements is processed resulting in radiomap, representing different locations and their corresponding strongest WiFi networks accompanied with the exact RSS values. Furthermore, the obtained sensor values are analysed and represented on a map using different mapping techniques.

4.3.1 Localisation of the measurements

This section covers the method used to localise measurements. Firstly, the principle of the method is described. Then, the two phases of the method are described: composing the radiomap and matching measurements to it. In the end, recommendations are made to improve the technique.

Localisation technique concept

Based on chapter one, a theoretical chapter on localisation techniques, and chapter three, a practical chapter on benchmark tests, the choice was made to assign a location to the measurements based on WiFi fingerprinting. A major drawback of WiFi fingerprinting, however, is that calibration measurements have to be made prior to localisation: WiFi RSSI values per MAC address have to be measured for a set of known locations. These are then stored in a radiomap (Mautz, 2012, p. 52).

To avoid this drawback, the choice was made to partly automate the process of composing the radiomap by using a GPS sensor. The GPS position in the calibration measurement is used to link WiFi fingerprints to a location. This is the first phase of the technique, as will be described in the next section: composing the radiomap. Once the radiomap has been composed, measured fingerprints can be matched in phase two of the method to locations in the radiomap: a location can then be assigned to a measurement.

Of course, as described in the localisation theory in chapter one, a major drawback of GPS is its current consumption. Therefore, this method is to use the GPS only in the calibration phase, to compose the radiomap. After the radiomap has been composed, GPS is no longer required and therefore can be switched off to save energy.

This localisation technique requires the storage of Media Access Control (MAC) addresses together with their RSSI values for a Wireless Access Points (WAP).During the calibration phase, also the GPS latitude and longitude values for each measurement. Regarding storage of MAC addresses, a legal context is provided in chapter 4.4.

Phase 1: composing the radiomap

In phase 1, the radiomap is composed from GPS positions and MAC addresses with their RSSI's. To arrive at this radiomap, the following steps were followed.

Firstly, the GPS positions were converted from Coordinate Reference System (CRS) WGS84, the world-wide CRS, to Amersfoort/RD New, the Dutch CRS. This was done since the Amersfoort/RD New CRS fits best to the Dutch territory. In addition, most Dutch data is available in this CRS and thus allow for easy comparison. Therefore, localising in Amersfoort/RD New allows best for visualisation and analysis purposes later on. For the conversion, Python code, extended with the Pyproj package (Appendix N), was used. This package uses the Proj.4 library. According to its documentation, the Dutch correction grid is classified as a 'Non-Free Grid', as registration is required before use (Proj4, n.d.). No information can be found on whether Pyproj does use the Dutch correction grid. The maximum distortion caused by not using the correction grid of the Netherlands is 25 cm (Crombaghs & Kösters, 2000). This error is larger close to the borders of the Netherlands. In the area of Delft it is limited to a maximum of 10 cm, which is insignificant, compared to the accuracy of the GPS measurements, which is already more than 1 metre, as reported in section 3.1.3.

Next, each measurement was assigned a grid cell, based on the coordinates of the GPS position. For this the grid cells as defined in chapter 4.2 are used. The GPS coordinates are converted to the Amersfoort/RD New CRS and, if it is in the extent of the grid, matched to a grid cell. The grid cells are numbered in ascending order, starting from top left, then row by row and ending bottom right. The implementation of this can be seen in Appendix N.

Although the accuracy and precision of the GPS position is not perfect, the choice was made to still assign each measurement to one cell only: the cell size is such that most errors are expected to fall within the extent of the cell. However, if this method were to be repeated with smaller cell sizes, these errors would have to be taken into account as well. A possibility to do so is to assign a measurement to all cells within a certain error threshold. In addition to this, importance levels can be assigned to measurements: the closer the measurement is to a grid cell centre, the higher its importance level will be.

When each measurement is assigned to a grid cell the radiomap can be made. For each grid cell, the MAC addresses and their corresponding RSSI values are stored in a dictionary. In case one MAC address is seen in one grid cell for multiple measurements, the RSSI value was averaged between all occurrences. Since the RSSI is measured in dB, expressed on a logarithmic scale, the way of averaging is done as in equation 4.3.1.

$$c_{avg} = 10 * \log_{10} \frac{\sum_{1\dots n} 10^{\frac{c_i}{10}}}{n} \quad dB \qquad i = 1,2,\dots,n$$

Where:

c_{avg}	average RSSI value [dB]
c_i	measurement RSSI value [dB]
n	number of measurements [-]

Eq. 4.3.1: Averaging decibel values on logarithmic scale (Tingay, 2013)

This step will result in a dictionary with MAC addresses and their average RSSI values per grid cell, and therefore per location. This dictionary is written to a csv-file, which has a line for every grid cell, and in the columns the different MAC and RSSI combinations.

Phase 2: matching measurements

For each measurement, the three best received MAC addresses are stored, together with their RSSI. To match a measurement to the radiomap, firstly, the library is searched for cells which have the same three MAC addresses as the measurement stored in their table. In case that no three matching MAC addresses are found, the search is extended to cells in which only two MAC address are available, similar to the MAC addresses in the measurement to localise. Cells, for which only one MAC address is similar are ignored, since no match is expected to be found with a certain confidence.

Afterwards, the concept of Euclidean Distance is used, as described by Teuber and Eissfeller (2006). This method compares the RSSI value of the measurement with the RSSI value in the library table and calculates the differences. Again, care is needed when calculating the difference, since the RSSI values are in dB and therefore on a logarithmic scale. For each MAC address both in the measurement and the library this is repeated. In this case at maximum three differences are defined, since only three MAC addresses are stored per measurement. Then, each difference is squared, the values are summed and then the sum is rooted. Result is a value d, describing the norm of the Euclidean distance vector. The smaller the value of d is, the closer the measurement is to the grid cell table in signal space and therefore expected to be in real world space. The equation 4.3.2 for the Euclidean Distance method is shown below.

$$d_{j,k} = \sqrt{\left(c_j^{AP_1} - s_k^{AP_1}\right)^2 + \left(c_j^{AP_2} - s_k^{AP_2}\right)^2 + \dots + \left(c_j^{AP_i} - s_k^{AP_i}\right)^2}$$

Where:

c	RSSI value radiomap data [dB]
s	RSSI value measurement data [dB]
d	euclidean distance in signal space [-]
i	number of access points [-]
j	index in radiomap [-]
k	index in measurements [-]
AP	access Point MAC address [-]

Eq. 4.3.2: Euclidean distance formula for signal space (Teuber and Eissfeller, 2006)

For all measurements, the Euclidean distance in signal space was calculated. The best match is found for the lowest value of Euclidean distance. This best match defines then the grid cell in which the measurement is to be localised, according to its MAC addresses and RSSI values. First all grid cells where all three MAC addresses match are compared. When these do not exist, also grid cells where only two MAC addresses match are taken into account. Only one MAC address match was deemed not enough to find a match. The radiomap matching function can be found in Appendix O.

Future improvements of the method

To improve the method for localising measurements, the use of a threshold match value should be investigated. A threshold value would define a maximum Euclidean Distance value for which a match is actually considered a reliable match. During this research, no clear pattern could be found in the Euclidean Distance values of incorrectly localised measurements compared to correctly localised measurements. Therefore, no apt threshold value could be found and implemented at the moment. However, in the implemented method a match can only be found if a measurement has two or more MAC addresses in common to the radiomap grid cell. All measurements which have none or one MAC address in common to the radiomap are excluded. In this way, a first check of match likelihood is implemented, without involvement of Euclidean Distance values. Still, further research into a Euclidean Distance threshold value would be interesting.

A second possible concept to improve the quality of the localisation method is to take the likeliness of a measurement happening in a specific grid-cell into account. For this reason the likelihood of grid cells, as described in part 4.2.2 on processing, was defined. Eventually, no localisation implementation of these cell likelihood values was made. However, especially for the boats, which are strictly bound to water surfaces, this would be an interesting way to improve quality of localisation.

In addition, to this general likelihood definition of cells, a prediction of a location, based on vehicle's movement pattern could be made. In this case, the assumptions has to be made that the speed of the device is constant and the route of the device is fixed. Also, an assumption on the speed value has to be made. Again, this method seems to suit the boats best. They are strictly bound to the water routes and furthermore have a relatively constant speed on straight parts of their route, not close to the boat stop, as well as a low average speed in relation to the LoRa messaging time interval of about 3.5 minutes, as described in section 4.1. This likelihood control concept is used to perform a small random sample quality check on localised measurements, as will be elaborated on in section 5.3 on quality control of results.

To conclude, when applying this method for long measurement periods, the radiomap should be continuously updated. However as stated in chapter one on Localisation Techniques, permanently running GPS would cause the platforms to use a lot of energy. Especially on the long run, this is highly inconvenient. A solution to keep the radiomap up to date is to keep one platform running with GPS, while deploying the others without. By doing so, the radiomap would be constantly updated. Another solution would be to deploy each platform with GPS, but only switching it on for very limited time frames.

Next to complementing the radiomap, attention should also be paid in removing old fingerprints from the radiomap which are no longer up to date. To do so, an option would be to remove all fingerprints older than a certain period of time or to remove all MAC addresses which have not been seen from a single grid cell for a certain period of time. Next to removing after a certain period of time, the choice could be made as well to define a desired number of fingerprints per grid cell and remove the oldest fingerprint each time a new entry is being made. Both the period of time and the desired number of fingerprints per grid cell require further research to allow for reliable implementation.

Visualisation of results

The resulting values, obtained after seven continuous days of measuring, are representing the temperatures in different locations in different times of the day. Furthermore, the measurements are distinguished between the vehicles, on which the sensors were mounted - boats and Tuk Tuks.

Firstly, the measurements of all days are averaged according to the grid cell that they belong to. For this a python code is used so the grid cell averaging of the measurements for all days is performed

simultaneously. The results are seven different raster images representing the average temperature of the day, which are then further compared. Since the data itself has spatiotemporal characteristics, it is important that the changes in time are also represented. This is done by linear graphs expressing the changes in the temperatures over time for every day, which are then combined with maps.

For one of the days more detailed representation is performed by showing the location of all measurements on the map during the different hours of the day and the corresponding temperatures.

The visualisation products mentioned above are used for the analysis of the performance of the systems during the days of measuring.

4.4. Legal Context

WiFi fingerprinting as used in this research requires the gathering and storage of MAC addresses of surrounding Wireless Access Points (WAP). In 2010 the Dutch Data Protection Agency (DPA)[15], investigated the handling of similar information by Google Inc. (*'Google'*) in the Netherlands (CBP, 2010; Kohnstamm, 2011). This chapter examines the jurisprudence and applicable, Dutch and European, laws and regulations for this research[16]. Conclusions and recommendations are provided for further research using similar data.

The investigation by the Dutch DPA had been initiated after the notice from Google on their activities regarding the collection of additional data using their Street View Cars (CPB, 2010). While collecting imagery for the Street View service, Google also registered MAC addresses, Service Set Identifier (SSID) names and RSSI values of nearby WAPs.[17] From the acquired data the estimated locations of the WAPs were calculated. The DPA investigation focussed on the question whether the combination of this data was considered personal information, and if so; whether Google adhered to the applicable regulations.

From the examination the DPA drew the following conclusions (CPB, 2010). Firstly the collected SSIDs are considered personal data as they are personalisable and could identify the owner of the WAP. Google could not prove their necessity for this data, based on this, regulations were breached on grounds of art. 8f Wet bescherming persoonsgegevens (Wpb) jo. art. 7 Directive 95/46/EC. Furthermore the combination of MAC addresses and location estimation of the WAPs are considered personal information in this context by the DPA. This as the location of the WAP is inseparably connected to the owner. Therefore it was concluded that Google breached regulations in that case as well on grounds of art. 34 Wpb. Considering both the MAC addresses and SSIDs, art. 27 jo. 28 Wpb have not been regarded, with which the obligation of notification to supervisory authority is regulated.

From the DPA investigation it can be concluded that MAC addresses can be considered personal data in specific circumstances. However the data gathered for this research is regarded as non-personal data; based on the following:

The specific context in which MAC addresses are considered personal data according to the DPA is not present in this research. The exact location of the WAPs is not stored or even known.
Data gathered in this research is not further shared and cannot be used to be related to an individual using legal methods; as the MAC addresses of WAPs do not refer to terminal devices[18] of individual users.

Nonetheless the code of conduct for the use of personal data in scientific research from the association of universities (VSNU, 2005) has been regarded in the processing of the data. For this research solely the vital data was gathered, all of which was nonspecific and not directly relatable. Therefore notice is not necessary at the Dutch DPA according to provision 3.8.1 (a) from the code of conduct. The data is only accessible for this pilot research, is stored in non-publicly accessible database and will be discarded after the research.

---

[15] ***In Dutch:*** *"Autoriteit Persoonsgegevens" formerly "College Bescherming Persoonsgegevens" (CBP)*
[16] *These recommendations intend to raise awareness of the different applicable (changes to) regulations and are not intended as binding legal advice.*
[17] *Furthermore, user data from unprotected WAP has been registered (Eustace, 2010), however this is not applicable for this research.*
[18] *As defined in Directive 1999/5/EC art. 2(b)*

Therefore the proper precautions have been taken into consideration while handling the data gathered for the localisation of the dynamic sensor platforms in this pilot project.

In the coming years the two European regulations concerning personal data and ePrivacy will change. These changes are set to take place in May 2018; Directive 95/46/EC, the General Data Protection Regulation (GDPR), will be replaced by EU 2016/679 and Article 29 Data Protection Working Party (2017) proposes to replace Directive 2002/58/EC, Directive on privacy and electronic communications (ePrivacy), at the same moment. The replacement of the GDPR is not focused on the specifics involved in this research. The main changes to the GDPR concern stricter regulations about handling personal data and the transparency towards the data subjects. A notable addition to the regulations is *'an online identifier'* to the definition of personal data in art. 4(1) from EU 2016/679*.* According to the Article 29 Data Protection Working Party (2017), the replacement for 2002/58/EC is focused on a uniform approach for ePrivacy in the European Union and to be an extension to the GDPR. It is described as a concept regulation, based on principles with broad restrictions and tight exemptions on the notion of electronic personal data. These changes are considered not sufficient regarding the tracking of terminal equipment of end users but does not elaborate the data from private interfaces[19].

Future research should take all these changes into consideration. The storage of MAC addresses from WAPs, especially without location are, as of the time of writing, not considered personal data. The changes in regulation are not indicating a modification to this definition, but the regulations regarding ePrivacy have not been finalised yet. Furthermore the interpretation of the new ePrivacy regulation has not been tested and there is jurisprudence in which Internet Protocol (IP) addresses are considered personal data, namely in Scarlet v SABAM (2011) and Breyer v Bundesrepublik Deutschland (2016). Although these addresses are on different levels in the Open Systems Interconnection model they both pertain to the identification of devices in the telecommunications network (Javvin Technologies, 2005, p.342). To ensure compliance to all regulations, and circumvent this untested part of the legislation, it is advisable to treat MAC addresses as personal data. This does not limit the usability of this data for localisation in research as art. 7(f) 95/46/EC provides ground for the use of personal data if the data is necessary for legitimate interests. It would require the implementation of regulations set in the GDPR. Different implementations already exist and could be easily adapted to be realised as part of a new research. Examples of these are the use of blogs to inform people or de facto standards like the addition of *'-nomap'* to SSIDs started by Google (Amy, n.d.). These options would ensure compliance to all regulations while maintaining usability of the MAC addresses or similar data. Another option would be, in case of a widespread existing infrastructure for freely accessible WiFi, the use of a selection of public WAPs. In this case the MAC addresses and RSSI values pertaining the WiFi network established by an organisation are not considered personal data; similarly if WAPs would be placed around the research area specifically for the localisation of the dynamic sensor platforms.

This pilot project is developed to test the implementation details of the establishment of a dynamic sensor network as part of the IoT, this includes the legal context in which such research is permitted. It can be concluded that the registration of MAC addresses from WAPs is not always considered as the handling of personal data. Even though the data for this project, in this context, does not constitute as personal data, all implementation details as described by the code of conduct from the VSNU have been taken into account. However this might not be sufficient for larger scale projects and therefore it is recommended that legal experts are consulted before such research is conducted.

[19] *As defined in Directive 1999/5/EC art. 2(e)(i)*

5. Results

In this chapter, the results of the methods outlined in methodology are described and visualised. The chapter is split into three parts. Firstly, the results of the sensor data are shown. In the visualisation part locations are combined with temperature and humidity levels as and time. Secondly, the quality of the sensor results is assessed. In the third part, the quality of the outcomes of the localisation method is assessed.

5.1. Sensor data results

This chapter is about sensor data results: sensor values - temperature and humidity - at certain locations - as found by the localisation method described in chapter 4 - at a moment in time. The results are presented in both text and visualisations.

The following results are comprising temperature and humidity measurements at different locations obtained after seven continuous days of measuring. The results for the noise measurements are not discussed as they are unreliable, due to the automatic gain control of the microphone as discussed earlier. During these days, there were numerous devices working during different periods of time, due to technical complications which were faced during the measurement week. Therefore the study area, which is covered by 50x50 m grid, shows diversity of measurements considering location and time.

The obtained measurements are visualised using QGIS software. The logical representation follows hierarchical structure - firstly average values of the whole week are represented, then average results for every day of the week are shown and lastly one day is represented in more detail. The main goal of the analysis of the following results is to review the performance of the sensor platform under the different circumstances.

Overall, the different days of measuring are characterized by diversity of temperature intervals, but still following a pattern. For each day, both temperature and humidity line graphs present stability, upward or downward trends without having extreme picks and inconsistencies, which indicate that the measurements sent by the sensors are meaningful and reliable. The observed values deviate between 15 to 43 degrees °C. It can be noticed that, for the whole week of measurements, there is an obvious correlation between temperature and humidity. During the day, when the temperature rises, the humidity levels decrease. In addition to that, a considerable difference is observed in the performance of the sensors mounted on different vehicles. While the temperature levels derived from the sensors placed on the boats are higher than the ones on the Tuk Tuks, the humidity values obtained from boats are lower than the ones from Tuk Tuks. Another important remark of the performance of the boat's sensors is the higher fluctuation of the values during the day, compared to those derived from the Tuk Tuks (Figure 5.1.1). When looking at the visualisations, it can be seen that the highest temperature measurements do not correlate with location of the platform, since for every day they are found in different grid cells.

The first two days of measuring show the lowest observed temperatures - from 15 to 32 degrees °C. For the next days there is clear upward trend of temperature, having as a peak 59 degrees °C on the 14th of June (Figure 5.1.2 and appendix P, Q, R) These values deviate a lot from the actual measured temperatures this day, which were up to 25 degrees °C (KNMI, 2017).

As on the 14th of June the most extreme values were obtained, this day is presented in more detail by mapping the position of the actual measurements along with the time, when they were obtained (Figure 5.1.3, 5.1.4). On the map, it can be seen that the extreme values are represented by just a few measurements around 15 o'clock. Overall there are a lot of measurements in the temperature interval 39-46 degrees °C, which are obtained in different hours of the day. The quality of these extreme values will be assessed further in the following section.

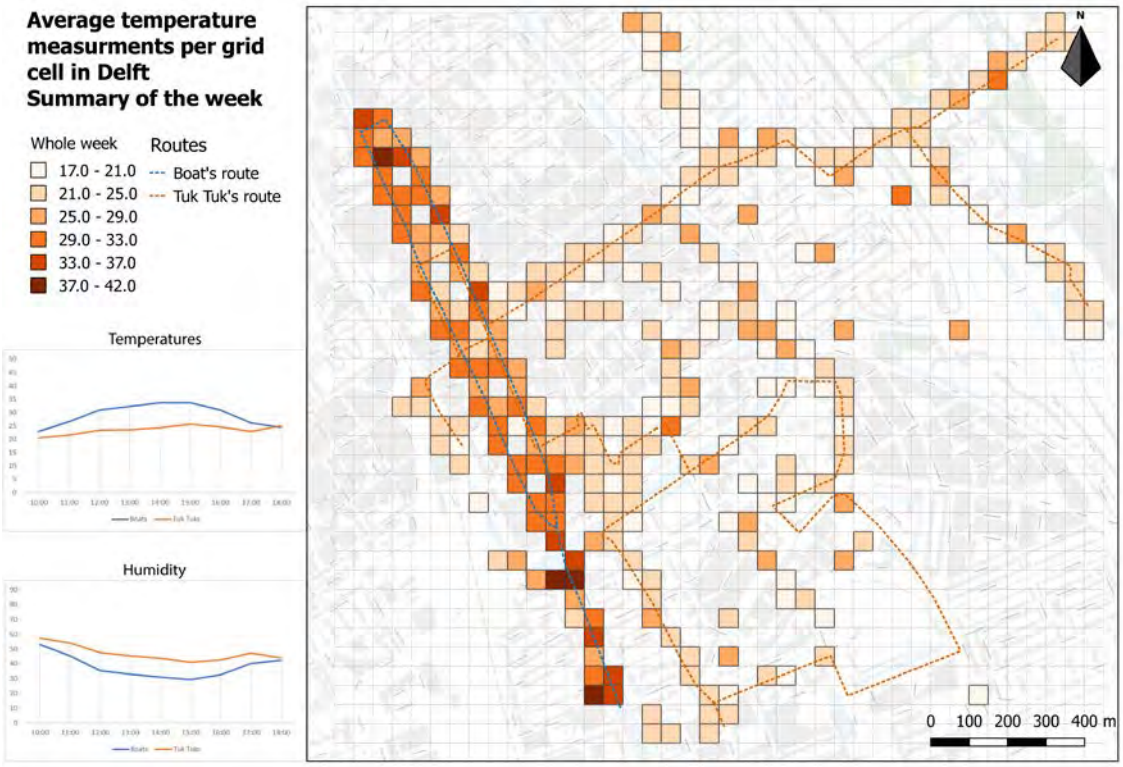


Figure 5.1.1: Average temperature per grid cell in Delft, weekly summary. (own work)(appendix S)

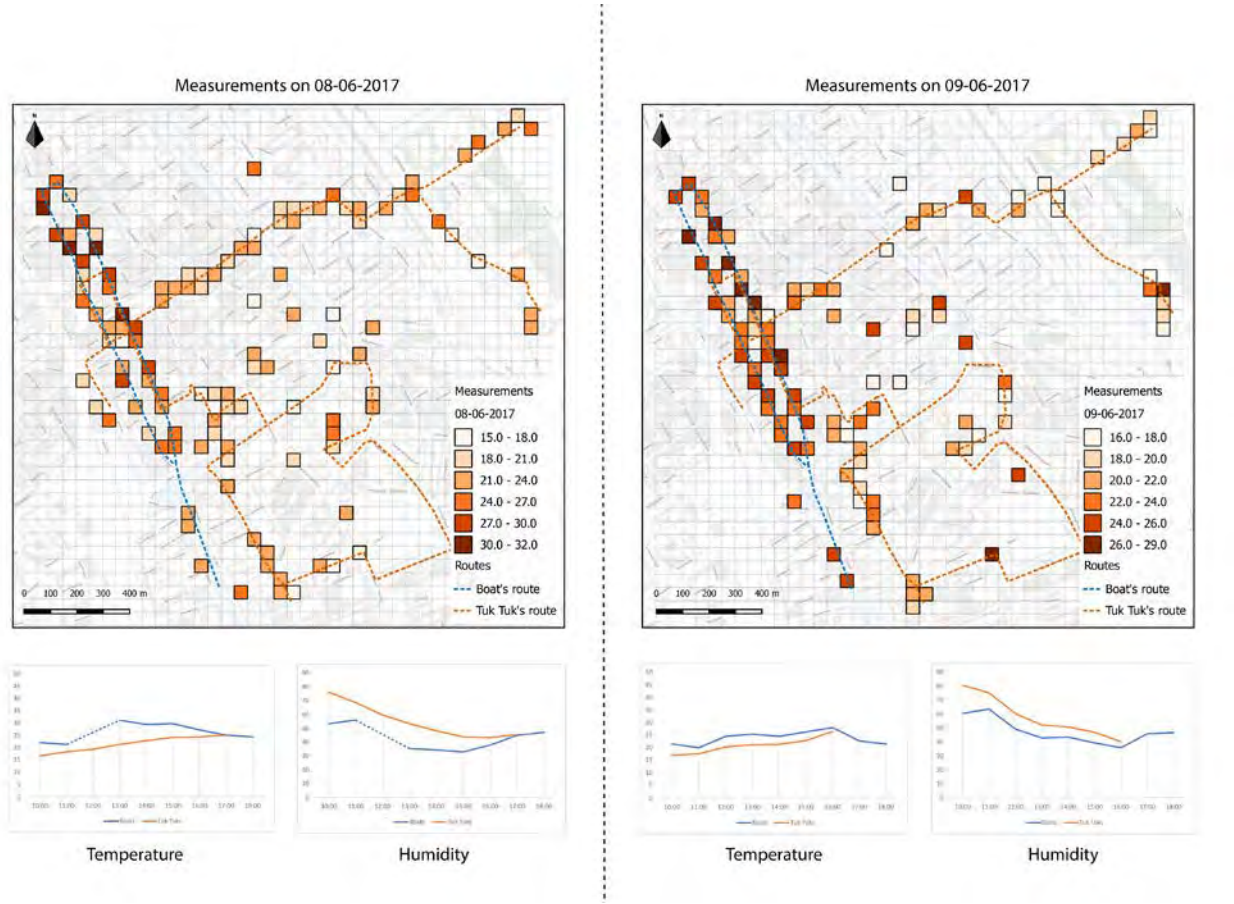


Figure 5.1.2: Average temperature per grid cell in Delft, daily summary 8th and 9th of June. (own work)(appendix P)

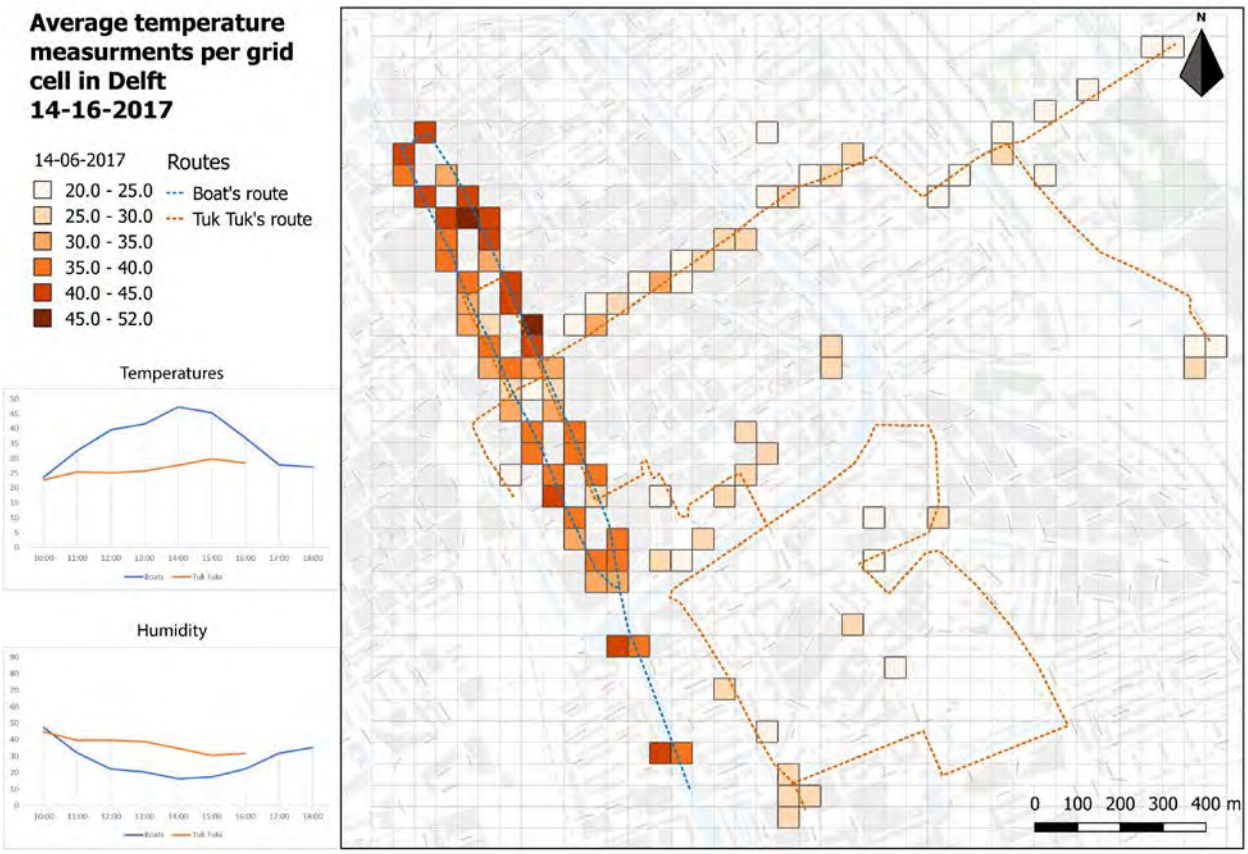


Figure 5.1.3: Average temperature per grid cell in Delft, 14th of June summary. (own work)(appendix T)

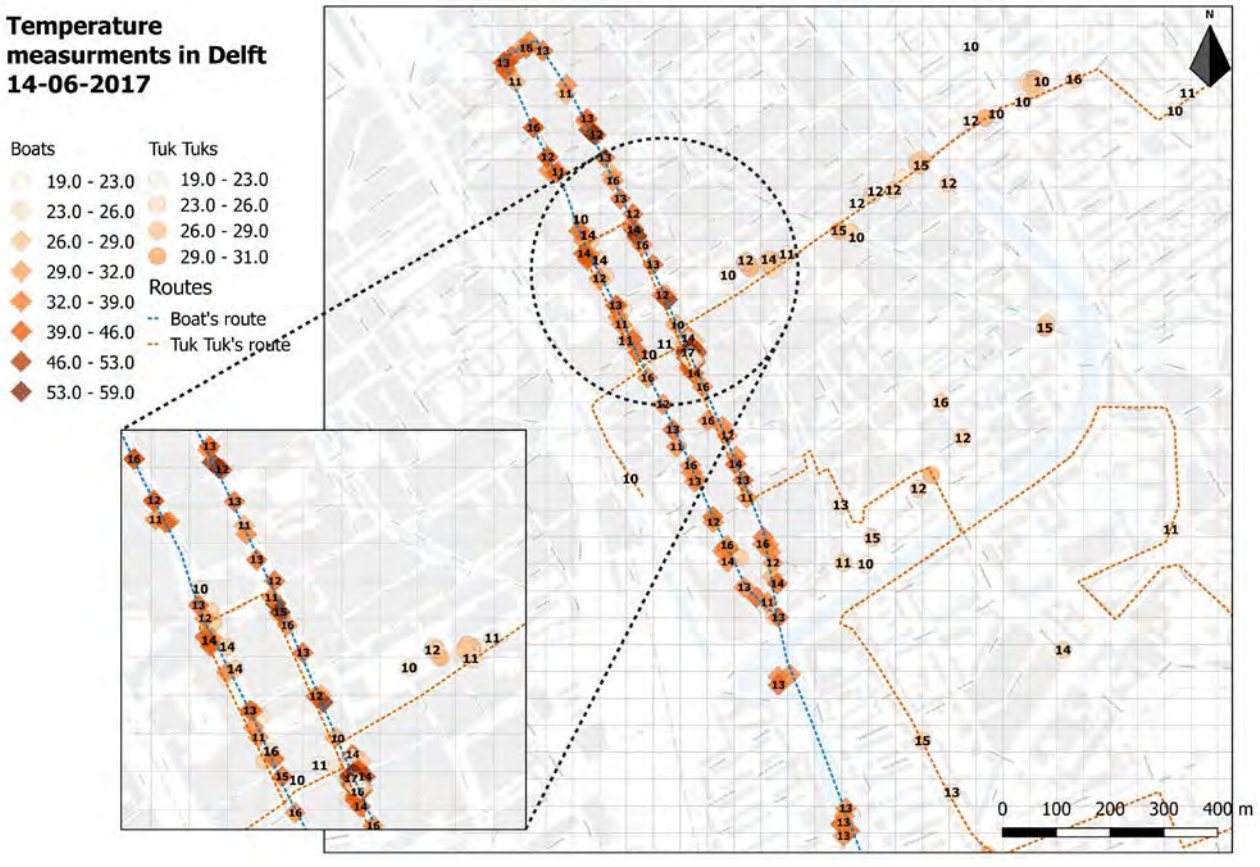


Figure 5.1.4: Measurement overview, 14th of June. (own work)(appendix U)

5.2. Sensor data quality control

In this chapter, a critical review is provided on the results of the sensor data. The quality of these readings is of high importance when interpreting them. Additionally, they are also of high interest for future researchers in order for them to be able to enhance the quality further. The reliability of the sensor results is assessed based on benchmark experiments, covering the placement of sensors on the vehicles.

For this purpose, the day with the most extreme values has been selected to be inspected. The 14th of June was one of the warmest days during the measurement period and was also very sunny (Figure 5.2.1). Nevertheless, the most extreme values occurred on the measurement platforms fixed on the boats. Looking at the measurement results from different sensor platforms (Figure 5.2.2), it can be seen that the overall temperature, just like in Figure 5.2.1 peaks at around 15 o'clock. However, the measured average values are much higher than the ones gathered by the professional Dutch weather station KNMI. This sheds light on the fact that the sensor platforms were by large part in direct sun. The largest peaks of the measurement happen at 14:39 for LoPy6 and 15:46 for LoPy7. In both cases, connecting the measurement to its location reveals the fact that when the boats were at their stops, measured temperature levels rose, while when the boats were moving, temperature levels dropped again. From this it can be concluded that the sensor platforms on the boats are not reliable in sunny conditions, as they are not measuring the ambient air temperature, but the sun-gain on the box. Unfortunately, that same day there was only one Tuk Tuk platform measuring in the same time frame as the one carried by the boats. Looking at the result graph (Figure 5.2.3), the temperatures have less extreme values and the averages differ from the expected values (Figure 5.2.1) by around five degrees °C. This could either be attributed to the context in which the Tuk Tuk was driving or the influence of the sun, like in the case of the boat. To be able to determine which of the two plays a larger role, further testing would be necessary. Therefore, it can be concluded that the sensor readings from the boats are unreliable in sunny conditions. To determine the quality of the readings from the Tuk Tuks further testing would be necessary. However, both should be reliable in cloudy conditions but in both cases they should be reliable in cloudy conditions.

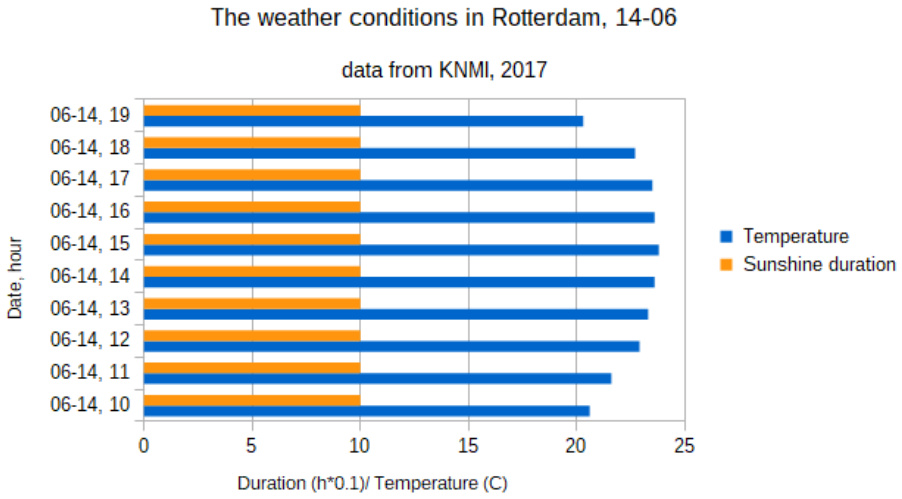


Figure 5.2.1: weather conditions in the closest weather station to Delft (own work, data from KNMI, 2017)

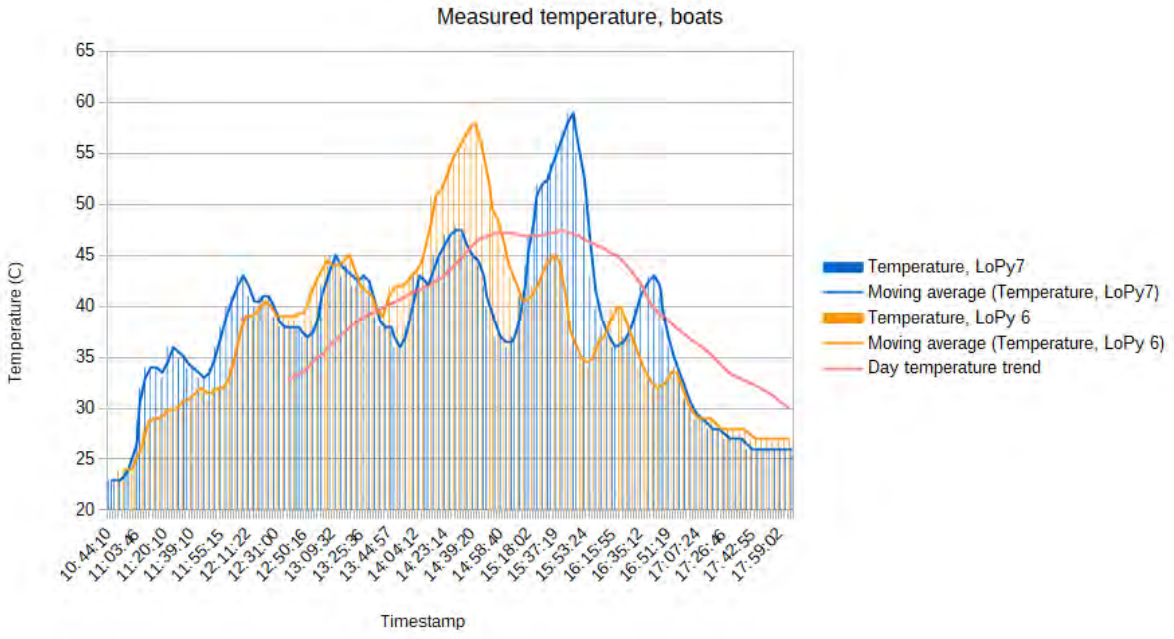


Figure 5.2.2: the temperature measurements of the sensor platforms, carried by the boats on the 14th of June (own work)

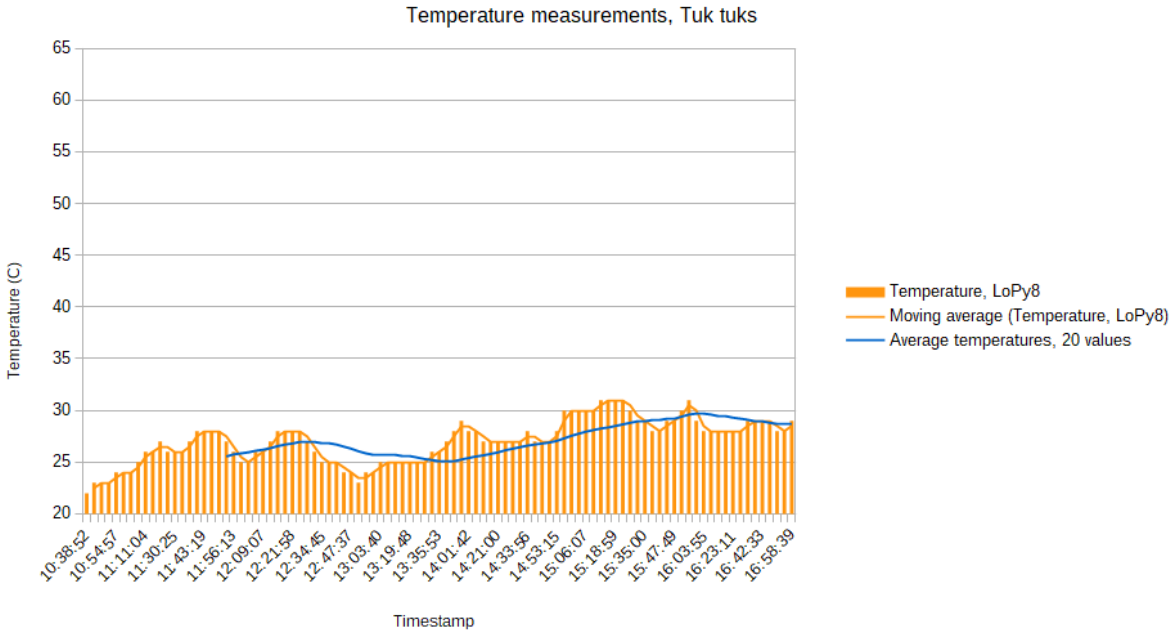


Figure 5.2.3: the temperature measurements of the sensor platform, carried by a Tuk Tuk on the 14th of June (own work)

5.3. Localisation quality control

In this chapter, the quality of the localisation methodology is assessed. Again, this is of high importance for interpretation of the results and future research. The quality of the localisation method is assessed by comparing its results to three reference methods.

Since the localisation method aims at localising a measurement in a grid cell, the quality of the method is expressed in percentage of correct grid cell allocations. This will therefore be referred to as correctness. The correctness will be expressed in percentages of localised measurements.

As far as the localisation method outcomes are concerned, three different ways are used to assess the correctness. The first two methods, GPS coordinate comparison and Google API coordinate comparison, are used to assess the correctness of all localised measurements. The third method is based on a previous location and is only used as a proof of concept for correctness assessment. It is applied on a very limited sample of localisation results.

5.3.1 Correctness compared to GPS coordinates

In this chapter, the GPS coordinate which was measured for each measurement is used to check whether the WiFi fingerprinting based localisation technique provides the same results as expected from the GPS coordinate: is the GPS coordinate in the same grid cell as the WiFi fingerprint method localises the measurement in? The GPS coordinate comes from the GPS module of the sensor platform, according to the specifications in section 3.3.5.

The correctness compared to GPS coordinates is expressed in four different stages. A measurement is localised *correctly* when it is localised in the same grid cell as the GPS coordinate of the measurement is in. It is localised *within radius of 1 cell* when the measurement is localised in a grid cell neighbouring the cell in which the GPS measurement lies. When it is localised two cells away, it is considered *within radius of two cells*. The localisation is stated to be *incorrect* when the measurement is localised in yet another grid cell. In case the measurement could not be localised by the method, due to absence of candidate grid cells as described in section 4.3.1 on methodology, its correctness is referred to as *not localised*. The code for the comparison can be found in Appendix O.

The localisation of measurements is performed with two different radiomaps. In the first radiomap, all measurements are included. Then, the measurements which were used to fill the radiomap were localised by lookup and matching in this same radiomap. In table 5.3.1 and Figure 5.3.1 below, the number of measurements and the corresponding percentage which are localised correct, within radius, not correct or not at all are stated.

Correctness	**# of measurements**	**% of measurements**
Correct	2125	79.6
Within radius of 1 cell	536	20.0
Within radius of 2 cells	8	0.3
Incorrect	2	0.1
Not localised	0	0
Total	**2671**	**100**

Table 5.3.1: correctness of localisation method compared to GPS coordinates, localised measurements also used to compose radiomap (own work)

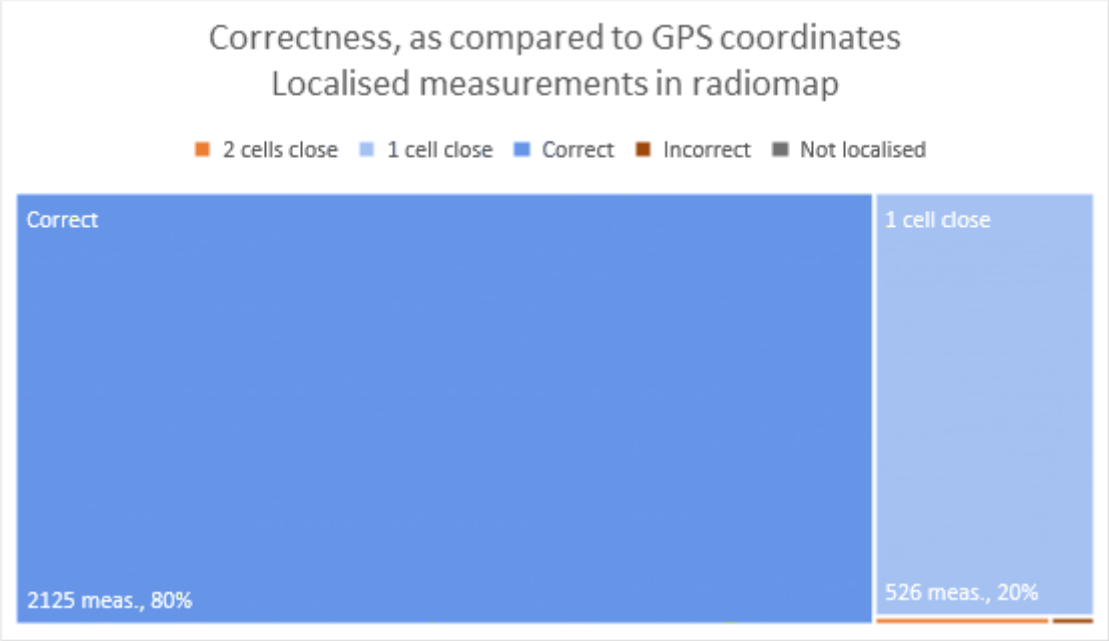


Figure 5.3.1: correctness of localisation method compared to GPS coordinates, localised measurements also used to compose radiomap (own work)

In the second radiomap, only measurements were included from the first half of the measurement period (up until the 12th of June 2017). This radiomap was used to localise measurements from the second half of the measurement period. Then, measurements were localised which were not incorporated in the radiomap (from the 13th of June 2017 onwards). In table 5.3.2 and Figure 5.3.2 below, the correctness of these are included.

Correctness	**# of measurements**	**% of measurements**
Correct	411	49.5
Within radius of 1 cell	314	37.8
Within radius of 2 cells	6	0.7
Incorrect	3	0.4
Not localised	96	11.6
Total	**830**	**100**

Table 5.3.2: correctness of localisation method compared to GPS coordinates, localised measurements not part of radiomap (own work)

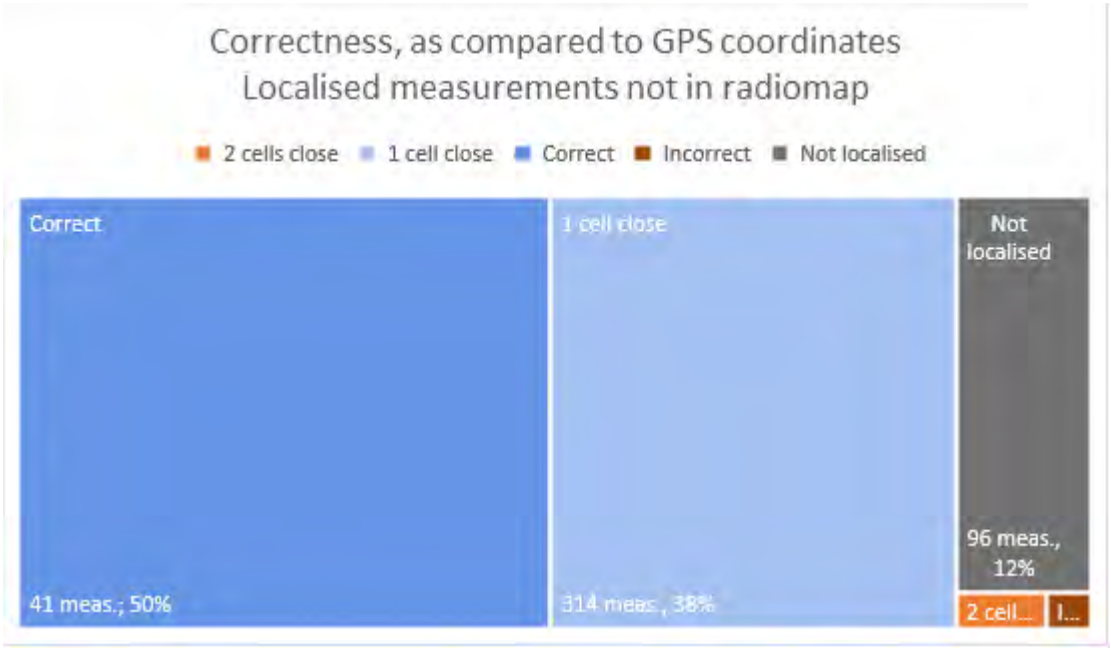


Figure 5.3.2: correctness of localisation method compared to GPS coordinates, localised measurements not part of radiomap (own work)

Seeing that the accuracy of the GPS measurements is 10.6 m, as found in chapter 2 on Benchmark Tests, it seems logical that some of the measurements fall inside a wrong grid cell. This could mean that measurements that are taken at the border of grid cells can easily be classified in the wrong cell. For the localisation of measurements that were not previously included in the radiomap, 12% is not localised, due to missing information. Out of the 1600 grid cells that were defined, 303 had seen measurements at the end of the full week. For the radiomap that was defined for all measurements up until the 12th of June, only 277 cells were filled. This explains the 96 non-localised measurements.

5.3.2 Correctness compared to Google API coordinates

As introduced in chapter 1 on Localisation Techniques, Google has an API aiming at deriving geolocations from WiFi MAC addresses and RSSI values. This API returns coordinates and an accuracy radius to the user (Google, 2017). Although it works as a black box, and therefore no insight can be gained into the techniques behind the API, it can be used as a way to check the localisation performed in this research. Similarly, to GPS, the Google API returns coordinates. These coordinates are matched with a cell in the grid as defined in chapter 4 Methodology. In case the Google coordinate lies in the same cell as a measurement was located to, the localisation is regarded correct. The code used to implement this can be found in Appendix O.

Similarly, to table 5.3.2 in the previous section, the correctness of the localisation as compared to the Google API method is stated below in table 5.3.3 and Figure 5.3.3. For the localisation of the measurements, the full radiomap is used.

Correctness	# of measurements	% of measurements
Correct	69	2.5
Within radius of 1 cell	869	31.4
Within radius of 2 cells	312	11.3
Incorrect	1521	54.8
Not localised	0	0
Total	**2671**	**100**

Table 5.3.3: correctness of localisation method compared to Google API coordinates (own work)

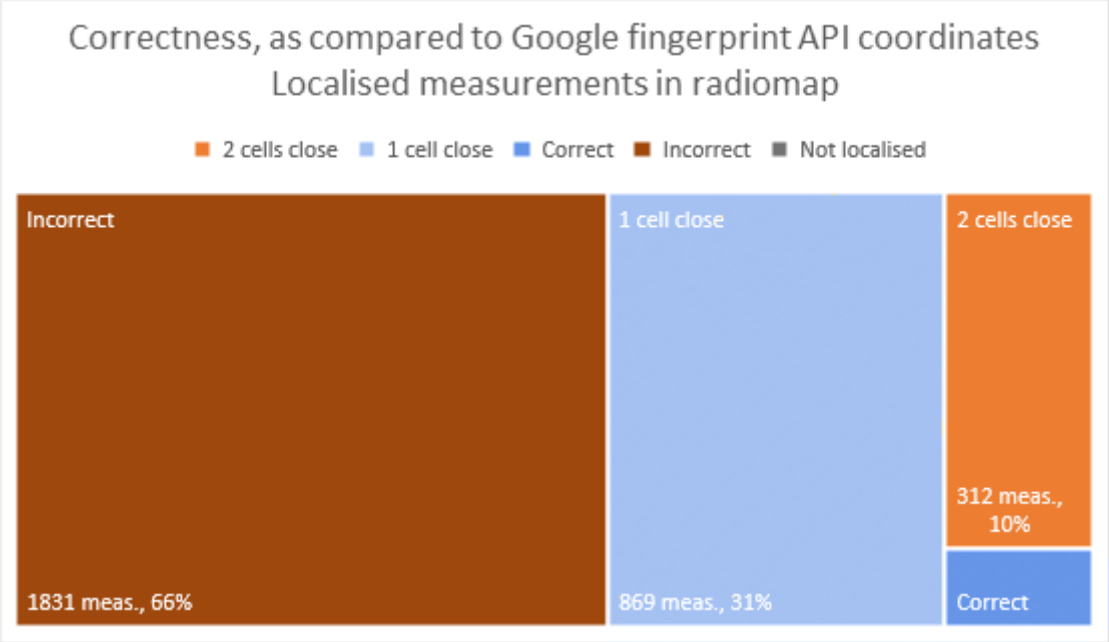


Figure 5.3.3: correctness of localisation method compared to Google API coordinates (own work)

The comparison between the Google API and the created radiomap does not match up as well as the comparison between the GPS measurements and the created radiomap. The accuracy of the GPS module is determined to be 10.6 m, which means that it falls inside the correct grid cell, or inside one of the neighbouring cells. The quality of the Google API coordinate that is sent alongside with the coordinate ranges between 30 and 200 m (Figure 5.3.4). The median of this quality parameter is at 53.0 m. The code for the comparison between the Google API and the GPS coordinate can be found in Appendix V.

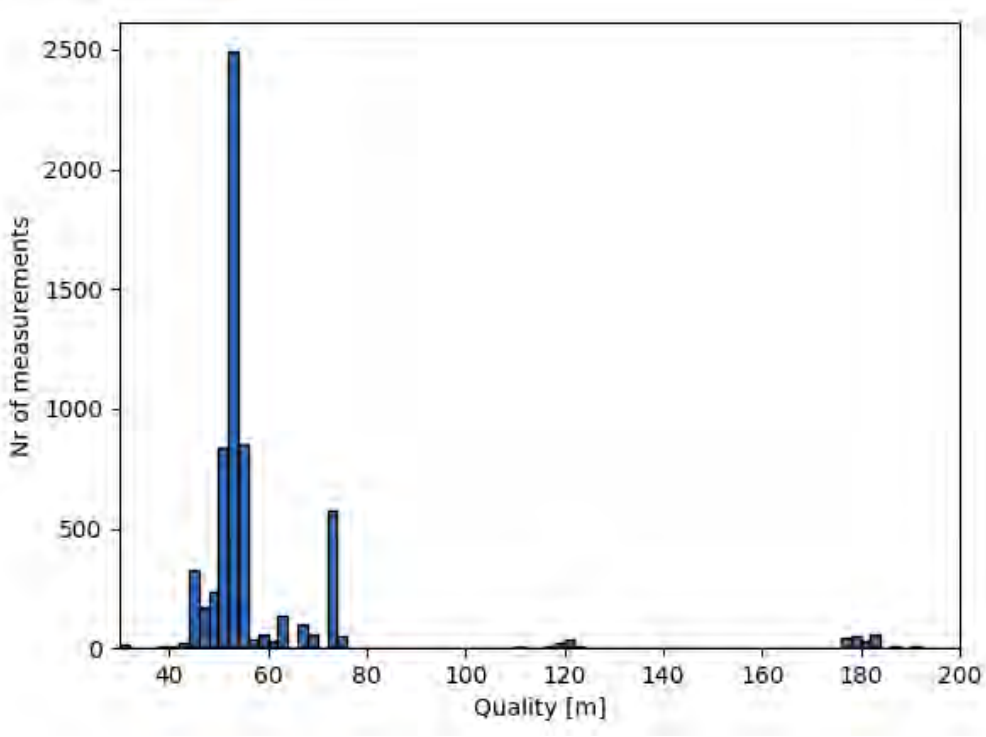


Figure 5.3.4: Distribution of Google API Quality parameter (own work)

To put this into further context an analysis has been made on the difference between the GPS coordinate and the coordinate of the Google API (Figure 5.35). This distance turned out to be up to 2500 m, which is much more than the quality parameter of 200 m and the GPS accuracy of 10.6 m combined. When the first bar in Figure 5.3.6, from 0 to 100 m, is enlarged the peak is found to be between 60 and 70 metres difference, which is similar to the median of the quality parameter added to the quality of GPS (53m + 10.6 m). By looking at the large differences between the GPS coordinate and the Google API, it becomes clear why there was so little overlap between the radiomap and the Google API quality control results.

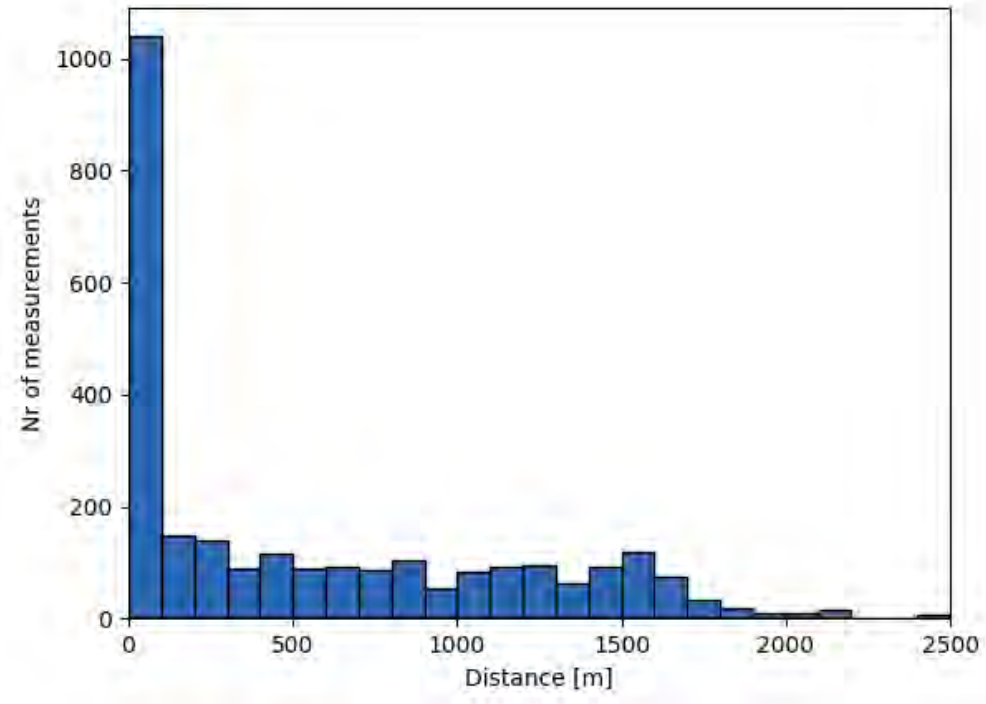


Figure 5.3.5: Difference between measured GPS coordinate and Google API coordinate (own work)

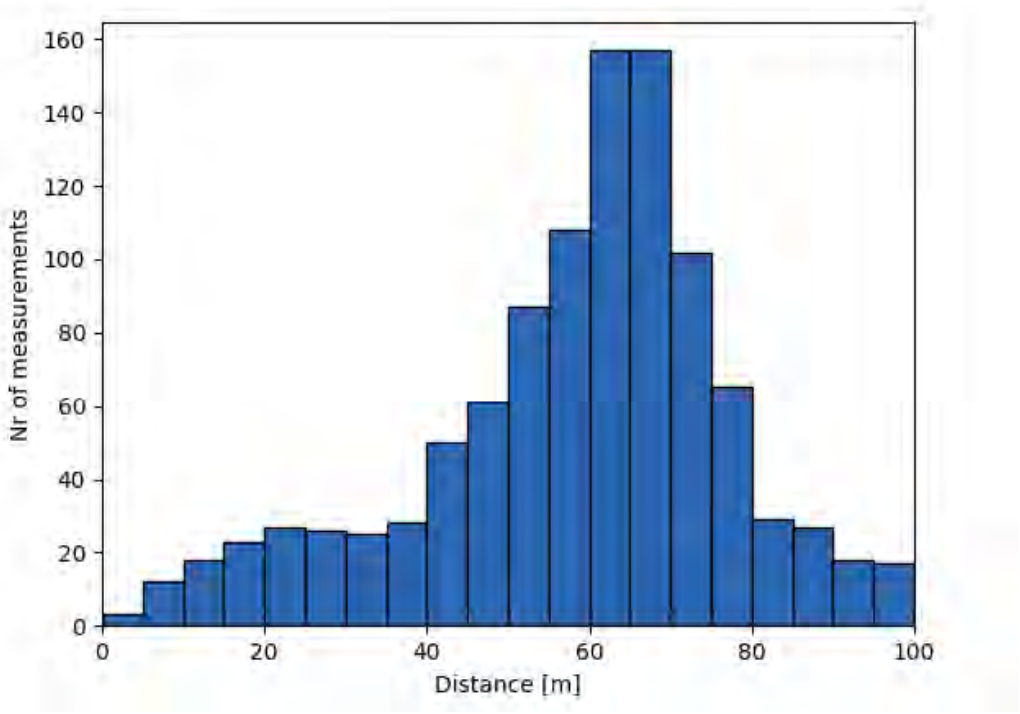


Figure 5.3.6: Difference between measured GPS coordinate and Google API coordinate (zoom on distribution of first 100 m) (own work)

5.3.3 Correctness based on previous known location

In this part, a small test will be done on the concept of verifying location correctness based on a previous known location, time interval and constant speed. This test can be seen as a proof of concept rather than a real check. Reason for this is that no automated implementation could yet be reached. Therefore, only a manual analysis on a very limited amount of sample measurements will be performed. As stated in chapter 4.3.1 on Methodology, a similar concept can be used to improve the localisation method.

For this correctness test, measurements were used of one boat at one day, June 13th 2017. To find usable series of measurements, i.e. forming sets of prior and next locations within a short time frame and with a constant speed to be assumed, the following steps were taken: firstly, the measurements were loaded into QGIS with their GPS latitude longitude values as coordinates. Then, all measurements were removed within short distance of the boat turning points in the North and South of the route. Also, the points on the Eastern half of the route were removed, since the boat stop lies halfway this part of the route: around these turning points and stop, no constant speed assumption can be made. Afterwards, a set of 16 measurements is left which lie, according to their GPS coordinate, on a straight part of the canal without stops. To finalize data preparation, the attribute table was checked for series of consecutive measurements according to their timestamp. From this, three series were identified of three or more measurements. In Figure 5.3.7 below, a visualisation of the processed sample data is shown. In table 5.3.4 below it, the relevant data of these measurements is available.

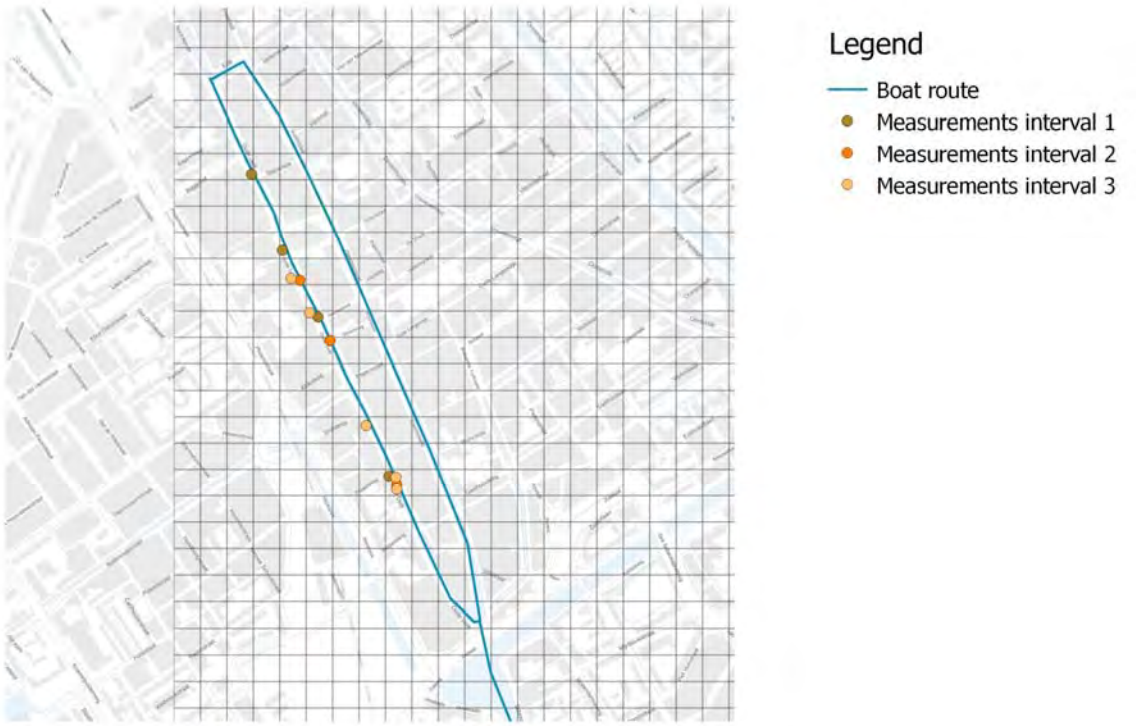


Figure 5.3.7: June 13 boat measurements visualised on GPS coordinates along straight part of boat route without stops, three series of consecutive timestamps (own work)

	Device address	**Date [d-m-y]**	**Time [h:m:s]**	**GPS Latitude**	**GPS Longitude**	**Cell number**
Interval 1 [4 meas.]	1420410E	13-6-2017	13:00:23	52.013671	4.354371	362
	1420410E	13-6-2017	13:03:37	52.012374	4.35525	484
	1420410E	13-6-2017	13:06:52	52.011242	4.356255	605
	1420410E	13-6-2017	13:13:17	52.008548	4.358277	848
Interval 2 [3 meas.]	1420410E	13-6-2017	14:33:49	52.011863	4.35574	564
	1420410E	13-6-2017	14:37:01	52.010841	4.3566	645
	1420410E	13-6-2017	14:43:26	52.008407	4.358494	848
Interval 3 [5 meas.]	1420410E	13-6-2017	18:16:08	52.011894	4.355499	524
	1420410E	13-6-2017	18:19:20	52.01131	4.356026	605
	1420410E	13-6-2017	18:25:47	52.00941	4.357627	767
	1420410E	13-6-2017	18:28:59	52.008533	4.358467	848
	1420410E	13-6-2017	18:32:12	52.008327	4.358495	848

Table 5.3.4: June 13th boat measurements along straight part of boat route without stops, three series of consecutive timestamps with their relevant attributes (own work)

Based on distance and duration of the hourly boat trips, including their turning time, stops and change of passengers, an average speed of 3 km/h was assumed for the straight part regarded in this test. The cell size of the grid is 50 metres. However, the fact that the route is slightly rotated in respect to the grid direction has to be taken into account in case Euclidean distances are calculated.

Below, three tables are included (tables 5.3.5 until 5.3.7), one per time interval. Within each table, sets of two subsequent measurements are identified and the *duration* between the two measurements is calculated. To each set, the *expected distance* in metres was calculated based on the duration and the assumed average speed of 3 km/h. Also to each set, the *actual cell distance* was calculated, based on the heart to heart Euclidean distance of the two cells the two measurements were localised in. The localisation is regarded correct if the difference between expected distance and actual cell distance is under 25 metres, i.e. half the cell size.

Interval 1 [4 meas.]		**Time [h:m:s]**	**Duration [h:m:s]**	**Cell number**	**Expected distance [m]**	**Actual cell distance [m]**	**Difference [m]**
Set 1.1	start	13:00:23	00:03:14	362	161	180	19 correct
	end	13:03:37		484			
Set 1.2	start	13:03:37	00:03:15	484	162	158	4 correct
	end	13:06:52		605			
Set 1.3	start	13:06:52	00:06:52	605	342	335	7 correct
	end	13:13:17		848			

Table 5.3.5: interval 1 measurement sets with expected distance based on speed, actual cell distance of localisation and difference in distances (own work)

Interval 2 [3 meas.]		**Time [h:m:s]**	**Duration [h:m:s]**	**Cell number**	**Expected distance [m]**	**Actual cell distance [m]**	**Difference [m]**
Set 2.1	start	14:33:49	00:03:12	564	160	112	48 incorrect
	end	14:37:01		645			
Set 2.2	start	14:37:01	00:06:25	645	320	292	28 incorrect
	end	14:43:26		848			

Table 5.3.6: interval 2 measurement sets with expected distance based on speed, actual cell distance of localisation and difference in distances (own work)

Interval 3 [5 meas.]		Time [h:m:s]	Duration [h:m:s]	Cell number	Expected distance [m]	Actual cell distance [m]	Difference [m]
Set 3.1	start	18:16:08	00:03:12	524	160	112	48 incorrect
	end	18:19:20		605			
Set 3.2	start	18:19:20	00:06:27	605	322	224	98 incorrect
	end	18:25:47		767			
Set 3.3	start	18:25:47	00:03:12	767	160	112	48 incorrect
	end	18:28:59		848			
Set 3.4	start	18:28:59	00:03:13	848	160	0	160 incorrect
	end	18:32:12		848			

Table 5.3.7: interval 3 measurement sets with expected distance based on speed, actual cell distance of localisation and difference in distances (own work)

From the consecutive measurement sets tables above, it can be concluded that only in interval 1 correctness is reached. In interval 2 and 3 no correct locations are found. However, explanation for this might be that the speed assumption is not reliable. Therefore, it is recommended, when this technique is used in future research, to include an accelerometer in the sensor station to measure speed of the vehicle.

One interesting measurement set however is set 3.4 in table 5.3.7. These two measurements are localised in the same cell, although they differ over three minutes in time and are therefore expected to be around 160 metres apart. However, when the GPS coordinates of these two measurements are regarded, they are found to be only around 25 metres apart. Moreover, these GPS coordinates both lie within cell 848 as localised. This highlights the fact that the correctness check based on speed might not be reliable without accurate speed information: a possible explanation is that the boat has stopped moving here for short time. A summary of the results is represented in table 5.3.8 and Figure 5.3.8.

Correctness	# of measurements	% of measurements
Correct	3	33
Within radius of 1 cell	4	44
Within radius of 2 cells	1	11
Incorrect	1	11
Not localised	0	0
Total	**9**	**100**

Table 5.3.8: correctness of localisation method according to previous known location and speed (own work)

Figure 5.3.8: correctness of localisation method according to previous known location and assumed speed (own work)

From the consecutive measurement sets tables above, it can be concluded that only in interval 1 correctness is reached. In interval 2 and 3 no correct locations are found. However, explanation for this might be that the speed assumption is not reliable. Therefore, it is recommended, when this technique is used in future research, to include an accelerometer in the sensor station to measure speed of the vehicle.

One interesting measurement set however is set 3.4 in table 5.3.8. These two measurements are localised in the same cell, although they differ over three minutes in time and are therefore expected to be around 160 metres apart. However, when the GPS coordinates of these two measurements are regarded, they are found to be only around 25 metres apart. Moreover, these GPS coordinates both lie within cell 848 as localised. This highlights the fact that the correctness check based on speed might not be reliable without accurate speed information: it might be the case that the boat has stopped moving here for short time.

6. Conclusions

This research project aimed to test the feasibility of implementing a dynamic sensor network to gather environmental spatiotemporal data and provide this data near-real-time to the citizens of the city of Delft. To achieve this, a system was developed, involving a moving sensor platform, sending information through LoRa communication, which is consequently decrypted and stored in a database on a virtual machine. Then, all measurements were linked to locations in order to arrive at spatiotemporal data. This chapter will conclude the research by firstly answering the seven sub-questions and finally answering the main research question.

1. Which localisation techniques are fit to localise sensors in a dynamic sensor network in an urban environment?
Various localisation techniques have been researched to determine their suitability for localisation of sensors in an urban environment. GPS is reliable, but is least efficient with regard to energy usage. The GPS sensor has still been used in the research, in order to create the radiomap and to assess the quality of the chosen localisation technique in the end

Another option that has been looked into is trilateration based on WiFi RSSI measurements. However, these measurements are prone to be affected by multipath issues and signal blockages in an urban environment. In addition, WiFi RSSI trilateration would require an infrastructure in which the density of devices, either dynamic or static, is high enough to allow for a triangulation network. Also, within this triangulation network numerous static devices with a known location are required. Also, such a technique would mean that devices are no longer operating autonomously. For this project therefore, this infrastructure was not deemed feasible to implement.

LoRa localisation was found to be not of use either. LoRa was proven to be heavily dependent on the presence of line of sight. In addition, internal clocks of the gateways were found not to be synchronized and the time accuracy far from sufficient to provide reliable localisation.

WiFi fingerprinting has the downside that it requires an up-to-date radiomap to which measurements can be matched. Use could be made as well of third party APIs, that are non-transparent. When the Google API is compared to reference GPS coordinate, questions on the reliability of the Google API results are raised.

In the end it was decided to use WiFi fingerprinting and construct a radiomap based on the reference GPS coordinates and received MAC addresses with their corresponding RSSI values from the first measurement phase. Then, new measurements were matched with the most similar location in signal space by comparing it to radiomap entries.

2. What level of location granularity and localisation correctness can be achieved as a result of the dynamic sensor network?
The grid that is used for localisation of the measurements has cells which are 50 by 50 metres. This size is chosen due to the accuracy of the GPS module, which was determined to be 10.6 m. With this accuracy, a too small grid cell size is not feasible, because this would lead to many errors in classification for the radiomap. The 50 by 50 m grid cell size worked out to be quite sufficient. The correctness of localisation was almost 80% when all measurements that were also included in the radiomap were localised, and 50% when localising measurements that were not included, using a radiomap made up of earlier measurements. Additionally, in this case 38% of measurements were localised within a 1 cell radius of the correct cell.

3. To what extent is LoRa communication fit to provide near real-time spatiotemporal data from moving sensor platforms in an urban environment?
Seeing that the message payload size for LoRa communication is limited to a maximum of 54 bytes and messages can only be sent 1% of the time, this causes huge limitations for a dynamic sensor system. The time it takes to send a 10-byte message at SF12 is 1.6 seconds. This means that the device should be silent for 158.4 seconds (2.64 minutes) afterwards. When a 30 byte message is sent at SF12 (which is the size and spreading factor most used in this research), it means that approximately once every 8 minutes a

message can be sent. For obtaining near real-time sensor data in a city, this would mean that more sensors are needed to get full coverage, because the six sensor platforms that were measuring on average, did not provide real-time coverage in this case. The message size also limits the amount of MAC addresses that can be sent for the radiomap, which makes the location determination less trustworthy. LoRa is as a communication method very suitable for sending sensor data via static devices of which the location is already known, because the location information takes up the largest part of the payload. For dynamic sensor platforms it would improve the usability if localisation via LoRa was possible, for instance via the metadata of the message (i.e. or LoRa RSSI trilateration of the message received), which means that no additional location information has to be sent along in the main payload. However, as stated in the answer on sub-question 1, some significant limitations would then have to be solved in these localisation techniques.

4. What electronic and physical components are essential for the sensor platform?
The selected electronic elements for the final sensor platform include a microphone, temperature and humidity sensor, GPS module, LoRa and WiFi antennas and a battery of 5000mAh, which can be charged with a USB-based circuit of 500mA. These modules, apart from the charging circuit, were to be assembled on a perfboard. The casing prototype is a solution involving a pair of connected water-tight food-boxes, modified with 3D printed ventilation elements. The latter involves opening one of the boxes for air-flow, which is necessary to be able to take the environmental measurements, yet keep the sensors protected from direct contact with water. The platform can function for two measurement days with the final code implementation.

5. Which database architecture is sufficient to store a significant amount of sensor data?
As database software PostgreSQL is installed on a server. The data flow takes place using Node-RED. Both raw and meaningful data are stored. The raw data table was populated with the data received from the KPN developer portal without doing any modification to this, while for populating the meaningful data table, the payload was decrypted and manipulated: the SQL query sent to populate the table was enriched with information from the Google Maps Geolocation API.

6. Which vehicles are suitable for measurements and how does this deployment affect the sensor data?
Any electrical vehicles with regular operating times following a constant route would be suitable for the proposed application. In this specific case, electrical Tuk Tuks and tourist boats were selected. The tests have revealed that the specific location for fastening the sensor platform has major influence on the sensor readings. Also, significant differences in measured values by Tuk Tuks and boats were found. These require further investigation to determine how to handle the quality of these measurements.

7. What are the fitting methods to visualise the obtained data?
The obtained measurements are primarily visualised using QGIS software. The logical representation follows a hierarchical structure: firstly, average values of the whole week are represented, then average results for every day of the week are shown and lastly one day is represented in more detail. Moreover, line plots were used to compare the different platform results in time, while the previously generated localisation grid was used in order the measurements to be represented as raster data. At the online dashboard, only line plots were used to visualise the data.

To conclude, the main research question "*To what extent can **near real-time spatiotemporal** data be obtained using a **dynamic** sensor network based on **LoRa communication** in an urban environment?"* can be answered as follows:

- Near real-time: The developed measurement system provided readings every three minutes, which were distributed throughout the case study area in an urban environment. The frequency of the data exchange is highly dependent on the amount of vehicles that is included in the research. This data can be brought to the public in an online dashboard.

- Dynamic: the sensor platform prototype was sufficient for deployment on moving vehicles, even though the functionality was not completely stable and continuous. Furthermore, in order good for complete coverage of the city centre to be obtained, a network with higher density is required.

- Spatiotemporal: After inspecting and testing the hardware capabilities, it was concluded that the most effective way to determine the location is by using WiFi fingerprinting, based on a radiomap composed with additional GPS measurements during the measurement period.

- Data: the quality of the collected data returned contrary results. While the localisation proved to be sufficient for the current use case, the sensor data was often not representative of the ambient air temperature. Further research about the most suitable sensor setup, acquiring unbiased measurements, is required.

- LoRa communication: this communication method is not very well suitable for the deployment of a dynamic sensor network due to the small payload size and sending frequency restrictions. Nevertheless, the power usage is very low and the range very long, and it can therefore be used to a certain extent.

7. Discussion and future recommendations

As with any project, not all of the defined objectives can be fully reached and throughout the project further understanding is gained on how to improve the efficiency and quality of the results. This chapter discusses and summarizes the recommendations defined in the previous chapters for numerous subjects: localisation techniques, technical improvements, scaling, management and external relations and standardisation.

7.1 Future localisation research

Due to the limited time and scope of the project, not all localisation techniques have been researched to their full extent. While wireless communication fingerprinting methodologies have been documented extensively, trilateration for these techniques are considered too unreliable. However the theoretical test described in this research shows potential in the relationship between distance and RSSI. Further research is required to prove the feasibility of such a localisation methodology for practical implementation with the use of self-learning algorithms and network approaches. Such implementations could be envisioned in an IoT environment in which more devices are connected to the network.

The legal aspect of collecting MAC addresses from WAPs could be avoided in a similar fashion. In a broader roll out of IoT applications, resulting in more publicly available WAPs, only a specific set of MACs can be considered for fingerprinting. This would avoid the possibility of dealing with personal data.

Different companies have claimed the use of LoRa for localisation or positioning. However no research has been found to solidify these bold statements. However TDoA for LoRa would in theory be possible, although the significant limitations were not expected to be solved in this research. More research is required to support the claims of the companies and a practical implementation of LoRa for localisation. An increased amount of IoT devices and time-synchronised LoRa s could support this.

Sensor fusion could also be used for the localisation of different sensor platforms: combining the information from different sensors to deduct a location. The prediction of movement used in this research is an example of this methodology to predict or validate a location. The addition of extra, inertial, sensors would in that case be beneficial. These possibilities are discussed in the following section, 7.2. While this kind of localisation has been proven to work on a smaller scale, further research is required to apply this to a practical implementation in an urban environment, especially where the speed of dynamic sensor platforms might vary.

7.2 Data usability

The data gathered during this project is not likely to be used outside of this report. To prevent such a situation next year, multiple aspects have to be taken into account. To begin with, it would be beneficial to involve stakeholders that need data for a specific application. This would help the team not only identify more environmental factors to be measured (e.g. ambient light, air pressure, magnetic fields), but also create beneficial and meaningful data analysis regarding the environmental issues of the research area. In addition, alternative data transfer methods from LoRa (e.g. GSM, WiFi) would allow larger data packages to be sent. On top of that, it would be likely to improve data coverage in the city, as the results could be sent more frequently or intermediated readings can be added. However, the most crucial part in making the data reusable is verifying and enhancing its reliability. To be able to fully understand whether the dynamic sensor systems are sufficiently reliable, more tests regarding the influence factors would have to be carried out. Keeping this in mind, it would be beneficial to rethink the casing design and the locations of fastening the platforms on the vehicles, as even the current temperature results show inconsistencies with the professional weather station in Rotterdam.

7.3 Technical improvements

To increase the battery life, an external energy source should be introduced: either solar charging or connection to, for instance, the cigarette lighter (12V plug) available in most of the Tuk Tuks. On top of that, the current power consumption can be further limited. Firstly, the code for the GPS energy saving mode should be stabilised, saving over 30mA in sleep mode. Secondly, with the deep sleep shields expected to arrive in mid-July 2017, the energy consumption of the MCU unit could be reduced by 9mA. Furthermore, to

increase the reliability of the sensor platform, a Computer Operating Properly (COP) timer code, which was made available with the most recent firmware update of June 9 2017, should also be added. This would make the system reset in case of unexpected crashes in code, which has caused platforms to stop working multiple times during the measurement phase. Regarding the casing, more prototypes should be made and tested, possibly including ready-made solutions.

Another field of improvement is associated with the battery life of the platform. As found during the benchmark testing, the system without GPS module could measure for over 10 days straight. The battery life could further be prolonged by using the solar panel solution. However, the GPS modules are still essential in building and updating the radiomap for WiFi fingerprinting. This suggests that the platforms should be split into two types: ones with a prolonged battery life (without GPS) and the ones that update the radiomap (with GPS). Regarding the latter, further alterations should also be considered. Among which would be extending the system with an SD memory card slot. The extension would also allow for A-GPS like functionality of the GPS module. On top of that, this would possibly enable storing of the radiomap on the platform, significantly reducing the LoRa payload, if the MAC addresses were given own-defined indicators. Nevertheless, it should be further researched, as alternative communication methods (e.g. GSM, WiFi) or even manual daily data collection could provide better results.

Nevertheless, it would seem logical to reduce the emphasis on electronics engineering in similar Geomatics projects, as it is not taught within the Master of Geomatics and the skills are unlikely to be within the project team beforehand. A solution to this would be either outsourcing this part or using ready-made solutions. In the case of LoPy, there are already two possibilities: PySense and Pytrack shields. The first one has almost all components that might be necessary in comparable applications: ambient light sensor, barometric pressure sensor, humidity sensor, 3 axis 12-bit accelerometer, temperature sensor, USB port with serial access, LiPo battery charger and MicroSD card compatibility. The second one, PyTrack, has GNSS and Glonass GPS, 3 axis 12-bit accelerometer, USB port with serial access, LiPo battery charger and MicroSD card compatibility. Both shields can be expanded with other modules if necessary and also have the functionality of the previously mentioned deep sleep shield.

7.4 Scaling

Once the data is sufficiently reliable, it could be incorporated within existing environmental dashboards, such as the Smart City Dashboard by TNO (Gemeente Delft, 2015). Moreover, if the system assembly and code was made more user friendly, the project could be published within the maker and environmental enthusiast communities. Following the example of uRAD monitor (uRAD, 2017), a large scale environmental observation dashboard could be created, enabling citizens to post and retrieve the volunteered climatological data. On a less ambitious note, the scale of the current implementation could be also increased by placing the sensor platform on public transport, such as buses or trams, or delivery cars with pre-planned routes, such as UPS route planning system ORION (UPS, n.d.). With this in mind, it would be beneficial to display the sensor reading averages on interactive displays inside the carrier vehicles. This would not only make a connection between data and citizen, but also exhibit the technology, being developed by the TU Delft.

7.5 Management and external relations

Besides technical improvements of the project, the project team identified multiple improvements in regard to the subject of management and external relations. To begin with, the delivery and testing of electronic parts should have been done prior the start of the project. This would have allowed the team to get acquainted with the components and their functionality already during the project definition phase. This would have given more time to test and identify the flaws of the used solution and likely produce alternative prototypes. Moreover, having had multiple people advising the project, it would have been beneficial to see the actual role definition, their scope, and relevant protocols (e.g. ordering and payment of parts) at the beginning of the project as an introduction document. This would have reduced the misunderstandings during the first phases. Moreover, the team missed a structured approach of introducing the subject of electronics engineering in the beginning of the project. The workshops given provided insight of how to interact with the Marvin and LoPy microcontroller units, however did not touch upon the theory of communications protocols, basic electronics elements and their functions in the system or essential skills

like soldering or using a multimeter. Even though support for these subjects was evident, an overview lecture or document would have been more efficient.

Regarding the subject of external relations, the team identified multiple stakeholders who would be of great use if a similar project was to be carried out in the future. The first one is KPN, as they could have provided more insight on the inner workings of the LoRa communications on their end. Moreover, having better communication with the company would likely allow one to research the subject of LoRa positioning using their network. During DynamIoT project the endeavours to benchmark the KPN LoRa localisation resulted in the company prohibiting any publications on the subject. This was due to the fact that the KPN representatives were not contacted prior to the project, informing about the intentions and goals of the research. However, they were positive about the possibility of future collaboration in a similar Geomatics project. Another essential stakeholder to be involved is Pycom, the producer and distributor of the LoPy microcontroller units, used in the project. After communications with one of Pycom engineers, it was made clear, that the company was open for organizing workshops and support skype sessions if the future projects required it. The knowledge that could be gained from this collaboration includes more detailed performance explanations and support for realizing the full potential of the MCU. Moreover, they could provide access to code which is under development, thus not only helping the project, but also benefitting their own efforts. The last stakeholder to be included is not a specific company, but a type. As mentioned in the section on data usability, it would be beneficial to involve professionals from the field of environmental research, such as TNO or Arcadis. They could help the students in better understanding of sensor reading influence factors and defining which of them are necessary to be able to identify the environmental issues in the research area.

7.6 Standardisation

For future implementation of a dynamic sensor network, research should be performed into standardisation of the sensor network architecture and its results as well. The outcomes of such research would then allow for better usage of data and a higher level of interoperability with other sensor networks. The Open Geospatial Consortium (OGC) develops open standards for global geospatial applications. Two of these OGC standards seem particularly interesting for future involvement in this research (Open Geospatial Consortium, 2017). The Sensor Web Enablement (SWE) standard aims at enabling developers to make their sensors, transducers and sensor data available via the internet. Within this standard, multiple other standards are involved. The other interesting standard is called SensorThings. The SensorThings standard aims at connecting IoT devices, data and applications over the internet via an API. Again, this standard covers multiple sensor related parts.

References

Agile Business Consortium. (n.d.). MoSCoW Prioritisation Handbook. Retrieved May 2, 2017, from: https://www.agilebusiness.org/content/moscow-prioritisation

Amy. (n.d.). Control access point inclusion in Google's Location services. Retrieved June 16, 2017, from https://support.google.com/nexus/answer/1725632?hl=en

Al-Hazaimeh, O., M., A., (2013). A NEW APPROACH FOR COMPLEX ENCRYPTING AND DECRYPTING DATA

Alvaro, F. (2016). Easy SQL Programming & Database Management

Arlinghaus, S. L., & Kerski, J. J. (2014). Spatial Mathematics: Theory and Practice through Mapping. (I. S. Britton, Ed.) (1st ed.). Boca Raton FL: CRC Press. Retrieved from https://www.crcpress.com/Spatial-Mathematics-Theory-and-Practice-through-Mapping/Arlinghaus-Kerski/p/book/9781466505322

Article 29 Data Protection Working Party. (2017). Opinion 01/2017 on the Proposed Regulation for the ePrivacy Regulation (2002/58/EC). Brussels. Retrieved from ec.europa.eu/newsroom/document.cfm?doc_id=44103

Atzori, L., Iera, A., & Morabito, G. (2010). Internet of Things: A survey. Computer Networks, 54, 2787-2805.

Bakker, W. (2016). Monitoring in buitengebieden nu energiezuinig en goedkoop. Geo-Info, (2016-4), 6-9.

Bolchini, C., Curino, C. A., Quintarelli, E., Schreiber, F. A., & Tanca, L. (2007). A data-oriented survey of context models. ACM SIGMOD Record, 36(4), 8. https://doi.org/10.1145/1361348.1361353

Bonnor, N. (2012). A Brief History of Global Navigation Satellite Systems. Journal of Navigation, 65(1), 1–14. https://doi.org/10.1017/S0373463311000506

Breyer v Bundesrepublik Deutschland, (2016). Retrieved from http://curia.europa.eu/juris/document/document_print.jsf?doclang=EN&text=&pageIndex=0&docid=184668&cid=1035645

Bulusu, N., Heidemann, J., & Estrin, D. (2000). GPS-less low-cost outdoor localization for very small devices. IEEE Personal Communications, 7(5), 28–34. https://doi.org/10.1109/98.878533

Caragliu, A., Del Bo, C., & Nijkamp, P. (2011). Smart Cities in Europe. Journal of Urban Technology, 18(2), 65–82. https://doi.org/10.1080/10630732.2011.601117

CBP. (2010). DEFINITIEVE BEVINDINGEN Onderzoek CBP naar de verzameling van Wifi-. The Hague: Autoriteit Persoonsgegevens. Retrieved from https://autoriteitpersoonsgegevens.nl/nl/nieuws/last-onder-dwangsom-voor-google-wegens-schending-privacywetgeving

Crombaghs, M., Kösters, A. (2000). Coördinaattransformaties en kaartprojecties (MDGAP - 2000.31). Ministerie van Verkeer en Waterstaat

Davidson, P., Hautamäki, J., Collin, J., & Takala, J. (2009, May). Improved vehicle positioning in urban environment through integration of GPS and low-cost inertial sensors. In Proceedings of the the European Navigation Conference (ENC'09).

EIP-SCC. (n.d.). Action clusters [website]. Retrieved on May 20, 2017, from: https://eu-smartcities.eu/action-clusters

Elmasri, R., & Navathe, S. (2015). Fundamentals of database systems. Boston: Pearson.

El-Mowafy, A., & Kubo, N. (2017). Integrity monitoring of vehicle positioning in urban environment using RTK-GNSS, IMU and speedometer. Measurement Science and Technology, 28(5), 055102.

Escudero, G., Hwang, J. G., & Park, J. G. (2017). An Indoor Positioning Method using IEEE 802 . 11 Channel State Information. *Manuscript submitted for publication*, Journal of Electrical Engineering & Technology, 12(1), 1921–1926. https://doi.org/10.5370/JEET.2017.12.1.1921

European Commision. (n.d.). The European Innovation Partnership on Smart Cities and Communities - European Commission. Retrieved May 20, 2017, from http://ec.europa.eu/eip/smartcities/

Faragher, R., & Harle, R. (2015). Location Fingerprinting With Bluetooth Low Energy Beacons. IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, 33(11), 2418–2428. https://doi.org/10.1109/JSAC.2015.2430281

Farnell. (n.d.). 73412-0110 - RF / Coaxial Connector, RF Coaxial, Straight Jack, Surface Mount Vertical, 50 ohm, Brass [image]. Retrieve June 18, 2017, from: http://nl.farnell.com/productimages/standard/en_GB/1340200-40.jpg

Foster, E. C., & Godbole, S. (2016). Database Systems: A Pragmatic Approach. Berkeley, CA: Apress.

Fullymax. (n.d.) Discharge Rate Characteristics [image]. Retrieved on May 20, 2017, from: http://fullymaxusa.com/

Gemeente Delft. (2015). Delft smart city. Retrieved on June 20, 2017, from: http://media.delft.nl/pdf/Eindrapportage Delft Smart City.pdf

Google. (2017). The Google Maps Geolocation API. Retrieved May 10, 2017, from https://developers.google.com/maps/documentation/geolocation/intro

Gray, S. (2016). GPS-free positioning for the Internet of Things. Low-power solution available today. Retrieved from http://www.csem.ch/Doc.aspx?id=42099&name=PR_16_LORA_alliance_EN.pdf

Grit, R. (2015). Projectmanagement: projectmatig werken in de praktijk. Retrieved on May 2, 2017, from: http://web.b.ebscohost.com.tudelft.idm.oclc.org/ehost/ebookviewer/ebook/bmxlYmtfXzEyOTA2MTVfX0FO0?sid=732099b3-c1d1-4932-a7bc-4ce9549dd711@sessionmgr103&vid=0&format=EB&lpid=lp_203&rid=0

Groves, P. D. (2011). Shadow Matching: A New GNSS Positioning Technique for Urban Canyons. Journal of Navigation, 64(3), 417–430. https://doi.org/10.1017/S0373463311000087

Haagmans, G. G., Verhagen, S., Voûte, R. L., & Verbree, E. (2017). A statistical analysis on the system performance of a bluetooth low energy indoor positioning system in a 3D environment. *Manuscript in preparation*

He, S., & Chan, S. H. G. (2016). Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons. IEEE Communications Surveys and Tutorials, 18(1), 466–490. https://doi.org/10.1109/COMST.2015.2464084

Henriksson, R. (2016). Indoor positioning in LoRaWAN networks; Evaluation of RSS positioning in LoRaWAN networks using commercially available hardware. Chalmers University of Technology.

HERE. (n.d.). Rest API - Requsting a Position. Retrieved from https://developer.here.com/rest-apis/documentation/positioning/topics/request-first-locate.html

Hofmann-Wellenhof, B., Lichtenegger, H., & Wasle, E. (2008). GNSS — Global Navigation Satellite Systems (1st ed.). Vienna: Springer. https://doi.org/10.1007/978-3-211-73017-1

Hornbuckle, C. A. (2010). Practional-N Synthesized Chirp Generator (Lora patent). United States of America. Retrieved from http://www.google.com/patents/US7791415

Huang, J., & Tan, H. S. (2006). A low-order DGPS-based vehicle positioning system under urban environment. IEEE/ASME Transactions on mechatronics, 11(5), 567-575.

IEEE 802.11 Working Group. (2010). Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Amendment 6: Wireless Access in Vehicular Environments. IEEE Standard for Information Technology - Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks - Specific Requirements. New York: Institute of Electrical and Electronics Engineers, Inc. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5514475

International Monetary Fund. (2017). Real GDP Growth [image]. Retrieved on June 28, 2017, from: http://www.imf.org/external/datamapper/NGDP_RPCH@WEO/OEMDC/ADVEC/WEOWORLD

Javvin Technologies. (2005). Network Protocols Handbook. Network Protocols Handbook, 342. Retrieved from https://books.google.co.uk/books?id=D_GrQa2ZcLwC&dq=408-872-3881+book&source=gbs_navlinks_s

Jol, M. (2016). LoRa Geolocation. Retrieved May 10, 2017, from https://zakelijkforum.kpn.com/lora-forum-16/lora-geolocation-8555

Jol, M. (n.d.) *KPN LoRa: Advanced Workshop*. Presentation, IoT Academy.

Khedo, K. K., Perseedoss, R., & Mungur, A. (2010). A wireless sensor network air pollution monitoring system. arXiv preprint arXiv:1005.1737.

KNMI. (2017). Uurgegevens van het weer in Nederland. Retrieved on June 19, 2017, from: http://projects.knmi.nl/klimatologie/uurgegevens/selectie.cgi

Kohnstamm, J. (2011). - Last onder dwangsom - Opdracht tot vernietiging payload data. The Hague: Dutch Data Protection Agency. Retrieved from https://autoriteitpersoonsgegevens.nl/nl/nieuws/last-onder-dwangsom-voor-google-wegens-schending-privacywetgeving

KPN LoRa Developer Portal, Zakelijk KPN Forum, June 13, 2017. https://zakelijkforum.kpn.com/lora-forum-16/kpn-lora-developer-portal-8429

Larson, T., Gould, T., Riley, E. A., Austin, E., Fintzi, J., Sheppard, L., ... & Simpson, C. (2017). Ambient air quality measurements from a continuously moving mobile platform: Estimation of area-wide, fuel-based, mobile source emission factors using absolute principal component scores. Atmospheric Environment, 152, 201-211.

Le Marchand, O., Bonnifait, P., Ibañez-Guzmán, J., Betaille, D., & Peyret, F. (2009). Characterization of GPS multipath for passenger vehicles across urban environments. ATTI dell'Istituto Italiano Di Navigazione, (189), 77–88. Retrieved from http://hal.archives-ouvertes.fr/hal-00445114

LoPy v1.0. (2017). Pycom Ltd. Retrieved from https://www.pycom.io/wp-content/uploads/2016/12/lopySpecsheet.pdf

lora-decrypt, (2017) npmjs.com, June 14, 2017 , https://www.npmjs.com/package/lora-decrypt

Lora Decryption on Application Server (2017), Zakelijk KPN Forum, June 13, 2017 https://zakelijkforum.kpn.com/lora-forum-16/lora-decryption-on-application-server-8416

LoRa | KPN Grootzakelijk. (n.d.) *Kpn.com*. Retrieved 18 June 2017, from https://www.kpn.com/zakelijk/grootzakelijk/internet-of-things/lora.htm

Mautz, R. (2012). Indoor Positioning Technologies. (D. R. Mautz, J. Müller-Gantenbein, & A. Geiger, Eds.), Sgc.Ethz.Ch. Zürich: Print-Atelier ADAG. Retrieved from http://www.sgc.ethz.ch/sgc-volumes/sgk-85.pdf

Node-RED (2017). Nodered.org. 2 May 2017. Web. 20 May 2017. https://nodered.org/

Noureldin, A., Karamat, T. B., & Georgy, J. (2013). Fundamentals of Inertial Navigation, Satellite-based Positioning and their integration. Berlin. https://doi.org/10.1007/978-3-642-30466-8

NovAtel. (2003). *GPS Position Accuracy Measures*. NovAtel. Retrieved from https://www.novatel.com/assets/Documents/Bulletins/apn029.pdf

Open Geospatial Consortium (2017). Sensor Web Enablement (SWE). Retrieved June 28, 2017, from: http://www.opengeospatial.org/ogc/markets-technologies/swe

Open Geospatial Consortium (2017). OGC SensorThings API Part 1: Sensing. Retrieved June 28, 2017, from: http://www.opengeospatial.org/standards/sensorthings

Olivier, B. A., & Sornin, N. (2016). Low power long range transmitter. United States of America. Retrieved from https://www.google.com/patents/US9252834

Online Kabelshop. (n.d.) SMA- RP SMA adapter [image]. Retrieved June 18, 2017, from: https://onlinekabelshop-nl.scdn2.secure.raxcdn.com/media/catalog/product/cache/1/image/304x304/9df78eab33525d08d6e5fb8d27136e95/v/l/vlsp02112a.jpg

OpenWiFi. (n.d.). Open WLAN map. Retrieved from https://openwifi.su/

Parameswaran, A. T., Husain, M. I., & Upadhyaya, S. (2009). Is RSSI a reliable parameter in sensor localization algorithms - an experimental study. IEEE International Symposium on Reliable Distributed Systems, 1–5.

PDOK. (2017). Publieke Dienstverlening op de Kaart. Retrieved June 16, 2017, from https://www.pdok.nl

PostGIS (2017) Spatial and Geographic Objects for PostgreSQL." Postgis.net. n.d. Web. 20 May 2017. http://postgis.net/

PostgreSQL 10 Beta 1 Released! (2017) Retrieved May 20, 2017, from https://www.postgresql.org/

Proj4 (n.d.) Proj4 Documentation Grids, Retrieved June 13, 2017, from http://proj4.org/grids.html?highlight=non%20free%20grids#netherlands

Pottinger, R., Bernstein, P. A. (2008). Schema Merging and Mapping Creation for Relational Source. . Retrieved on June 01, 2017, from https://pdfs.semanticscholar.org/1d8e/1090ea2cee43769ea5aaf4b7560da069cc1b.pdf

Pycom. (2017). Root causes of high deep sleep current. Retrieved on June 15, 2017, from: https://forum.pycom.io/topic/1022/root-causes-of-high-deep-sleep-current

Rusli, M. E., Ali, M., Jamil, N., & Din, M. M. (2016). An Improved Indoor Positioning Algorithm Based on RSSI-Trilateration Technique for Internet of Things (IOT). 2016 International Conference on Computer and Communication Engineering (ICCCE), 72–77. https://doi.org/10.1109/ICCCE.2016.28

Scarlet Extended SA v Société belge des auteurs, compositeurs et éditeurs SCRL (SABAM) (2011). Retrieved from eurlex.%0Aeuropa.eu/LexUriServ/LexUriServ.do?uri=CELEX:62010CJ0070:EN:HTML

Schmidt, N. (2016). LoRa: A Radio Approach Towards IoT Geo Localization. Microwave Journal.

Senaratne, H., Bröring, A., Schreck, T. (2013). Using Reverse Viewshed Analysis to Assess the Location Correctness of Visually Generated VGI. University of Konstanz and Institute of Geoinformatics at University of Münster. Retrieved from https://kops.uni-konstanz.de/bitstream/handle/123456789/24372/Senaratne_243720.pdf;sequence=2

Smith, J. E., & Nair R. (2005). The Architecture of Virtual Machines. COVER FEATURE

Sommer, D. M. (n.d.) Principles of GPS. Presentation Slides, date of publication unknown, available at URL http://www.gisresources.com/wp-content/uploads/2013/11/Principles-of-GPS-4-13-04.pdf

Spek (van der), S. (2017). Smart(er) Environments (1st ed.). Delft.

Szewczyk, R., Osterweil, E., Polastre, J., Hamilton, M., Mainwaring, A., & Estrin, D. (2004). Habitat monitoring with sensor networks. Communications of the ACM, 47(6), 34-40.

Teuber, A., & Eissfeller, B. (2006). WLAN Indoor Positioning Based on Euclidean Distances and Fuzzy Logic. Institute of Geodesy and Navigation, University FAF, Munich, Germany. Retrieved from http://wpnc.net/fileadmin/WPNC06/Proceedings/31_WLAN_Indoor_Positioning_Based_on_Euclidean_Distances_and_Fuzzy_Logic.pdf

ThingPark Products | How to deploy your IoT network. (2017). Retrieved June 19, 2017, from https://www.actility.com/products/

Tian, Y., Xu, G., Neitzel, F., He, K., Xu, Y., & Jiang, N. (2016). An approach to improve the GPS positioning performance under urban environment conditions. Measurement, 93, 414-420.

Tingay, J. (2013). *Noise data averaging: How to average noise measurements.* Retrieved on June 20, 2017, from http://www.cirrusresearch.co.uk/blog/2013/01/noise-data-averaging-how-do-i-average-noise-measurements/

Tsakiri, M., Stewart, M., Forward, T., Sandison, D., & Walker, J. (1998). Urban Fleet Monitoring with GPS and GLONASS. The Journal of Navigation, 51(3), 382–393. Retrieved from https://www.cambridge.org/core/journals/journal-of-navigation/article/urban-fleet-monitoring-with-gps-and-glonass/D3B5C49AD64C7DFF2F36D5DA77A663A7

Tsui, A. W., Chuang, Y. H., & Chu, H. H. (2009). Unsupervised learning for solving RSS hardware variance problem in WiFi localization. Mobile Networks and Applications, 14(5), 677–691. https://doi.org/10.1007/s11036-008-0139-0

U-blox. (2013). u-blox 6 Receiver Description Including Protocol Specification. Retrieved on June 4, 2017, from: https://www.u-blox.com/sites/default/files/products/documents/u-blox6_ReceiverDescrProtSpec_%28GPS.G6-SW-10018%29_Public.pdf?utm_source=en%2Fimages%2Fdownloads%2FProduct_Docs%2Fu-blox6_ReceiverDescriptionProtocolSpec_%28GPS.G6-SW-10018%29.pdf

U-blox. (2011). *NEO-6 GPS Modules Data Sheet*. ARM. Retrieved on June 18, 2017, from https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf

uRADMonitor (2017), Global Environmental Monitor Network. Retrieved June 20, 2017, from: https://www.uradmonitor.com/

United Nations. (2014). World's population increasingly urban with more than half living in urban areas. Department of Economic and Social Affairs. Retrieved from http://www.un.org/en/development/desa/news/population/world-urbanization-prospects-2014.html

UPS. (n.d.). Optimized Network [website]. Retrieved on May 20, 2017, from: https://sustainability.ups.com/committed-to-more/optimized-network/

Verbeke, T., Olti, E., & Munteanu, A. (2016). Development and Demonstration of a LoRa TDOA-based Localisation System: Demo. Proceedings of the 10th International Conference on Distributed Smart Camera, 206–207. https://doi.org/10.1145/2967413.2974031

VSNU. (2005). Gedragscode voor het gebruik van persoongegevens in wetenschappelijk onderzoek [Code of conduct for the use of personal data in scientific research]. Retrieved from http://www.vsnu.nl/files/documenten/Domeinen/Accountability/Codes/Gedragscode persoonsgegevens.pdf

Xia, F., Yang, L. T., Wang, L., & Vinel, A. (2012). Internet of things. International Journal of Communication Systems, 25(9), 1101.

Xue, W., Qiu, W., Hua, X., & Yu, K. (2017). Improved Wi-Fi RSSI Measurement for Indoor Localization. IEEE Sensors Journal, 17(7), 2224–2230. https://doi.org/10.1109/JSEN.2017.2660522

Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor network survey. Computer networks, 52(12), 2292-2330.

Yunhau, L., & Zheng, Y. (2011). Location, Localization, and Localizability, Location-awareness Technology for Wireless Networks (1st ed.). New York: Springer. https://doi.org/10.1007/978-1-4419-7371-9

Zakelijkforum.kpn.com. 15 Sept. 2016. Web. 21 May 2017. https://zakelijkforum.kpn.com/lora-forum-16/lora-decryption-on-application-server-8416

Zhang, L., Liu, X., Song, J., Gurrin, C., & Zhu, Z. (2013). A comprehensive study of bluetooth fingerprinting-based algorithms for localization. In Proceedings - 27th International

Appendices

A: Create-Drop-Populate Tables and Retrieve Data
B: Populate raw_data
C: Message to Google API
D: Received Google API
E: Populate meaningful_data_google
F: Node-RED flow
G: Dashboards
H: LoPy Pinout
I: LoPy main code
J: LoPy GPS library
K: LoPy DTH library
L: Decryption
M: Manipulation of decrypted payload_hex
N: Create Radiomap
O: Comparison Radiomap vs GPS and Google API
P: Visualisation 08-09 June
Q: Visualisation 10-11 June
R: Visualisation 12-13 June
S: Visualisation whole week
T: Visualisation grid 14 June
U: Visualisation measurements 14 June
V: Comparison GPS vs Google API

Appendix A- Create-Drop-Populate Tables and Retrieve Data

Create raw_data Table (template node):

```
CREATE TABLE public.raw_data
(
    Decrypted character varying,
    Time character varying,
    DevEUI character varying,
    FPort character varying,
    FCntUp character varying,
    ADRbit character varying,
    MType character varying,
    FCntDn character varying,
    payload_hex character varying,
    mic_hex character varying,
    Lrcid character varying,
    LrrRSSI character varying,
    LrrSNR character varying,
    SpFact character varying,
    SubBand character varying,
    Channel character varying,
    DevLrrCnt character varying,
    Lrrid character varying,
    Late character varying,
    LrrLAT character varying,
    LrrLON character varying,
    Lrrid_Lrr0 character varying,
    Chain_Lrr0 character varying,
    LrrRSSI_Lrr0 character varying,
    LrrSNR_Lrr0 character varying,
    LrrESP_Lrr0 character varying,
    Lrrid_Lrr1 character varying,
    Chain_Lrr1 character varying,
    LrrRSSI_Lrr1 character varying,
    LrrSNR_Lrr1 character varying,
    LrrESP_Lrr1 character varying,
    Lrrid_Lrr2 character varying,
    Chain_Lrr2 character varying,
    LrrRSSI_Lrr2 character varying,
    LrrSNR_Lrr2 character varying,
    LrrESP_Lrr2 character varying,
    Lrrid_Lrr3 character varying,
    Chain_Lrr3 character varying,
    LrrRSSI_Lrr3 character varying,
    LrrSNR_Lrr3 character varying,
    LrrESP_Lrr3 character varying,
    Lrrid_Lrr4 character varying,
    Chain_Lrr4 character varying,
    LrrRSSI_Lrr4 character varying,
    LrrSNR_Lrr4 character varying,
    LrrESP_Lrr4 character varying,
    Lrrid_Lrr5 character varying,
    Chain_Lrr5 character varying,
    LrrRSSI_Lrr5 character varying,
    LrrSNR_Lrr5 character varying,
```

```
    LrrESP_Lrr5 character varying,
    Lrrid_Lrr6 character varying,
    Chain_Lrr6 character varying,
    LrrRSSI_Lrr6 character varying,
    LrrSNR_Lrr6 character varying,
    LrrESP_Lrr6 character varying,
    Lrrid_Lrr7 character varying,
    Chain_Lrr7 character varying,
    LrrRSSI_Lrr7 character varying,
    LrrSNR_Lrr7 character varying,
    LrrESP_Lrr7 character varying,
    Lrrid_Lrr8 character varying,
    Chain_Lrr8 character varying,
    LrrRSSI_Lrr8 character varying,
    LrrSNR_Lrr8 character varying,
    LrrESP_Lrr8 character varying,
    Lrrid_Lrr9 character varying,
    Chain_Lrr9 character varying,
    LrrRSSI_Lrr9 character varying,
    LrrSNR_Lrr9 character varying,
    LrrESP_Lrr9 character varying,
    CustomerID character varying,
    CustomerData_pro character varying,
    CustomerData_ver character varying,
    ModelCfg character varying,
    InstantPER character varying,
    MeanPER character varying,
    DevAddr character varying
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.raw_data
  OWNER TO dynamic;
```

Create meaningful_data_Google Table (template node):

```
CREATE TABLE public.meaningful_data_Google
(
        DevAddr character varying,
    Date DATE,
    Time TIME,
        Mac1 character varying,
        Strength1 int,
        Mac2 character varying,
    Strength2 int,
    Mac3 character varying,
    Strength3 int,
    Temperature int,
    Humidity int,
    Sound int,
    GPS_latitude float,
    GPS_longitude float,
    API_latitude float,
    API_longitude float,
    API_accuracy float,
    GPS_Location geometry(POINT, 4326),
```

```
    API_Location geometry(POINT, 4326)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE public.meaningful_data_Google
  OWNER TO dynamic;
```

Drop raw_data Table (template node):

```
DROP TABLE raw_data;
```

Drop meaningful_data_Google Table (template node):

```
DROP TABLE meaningful_data_Google;
```

Populate Table (template node-experimental):

```
INSERT INTO trial (field1, field2) VALUES ('trial', 4234);
```

Populate Table (function node-experimental):

```
msg.payload = "INSERT INTO trial (field1, field2) VALUES ('trial2', 12345);";
return msg;
```

Retrieve data from raw_data Table (template node):

```
select * from raw_data;

```

Retrieve data from meaningful_data_Google Table (template node):

```
select * from meaningful_data_Google;
```

Appendix B – Query to populate raw_data Table

```
var numberOfRows = msg.payload.DevEUI_uplink.Lrrs.Lrr.length;

if (numberOfRows === 10) {
    val1 = "INSERT INTO raw_data (Decrypted, Time, DevEUI, FPort, FCntUp, ADRbit, MType, FCntDn, payload_hex, mic_hex, Lrcid, LrrRSSI, LrrSNR, SpFact, SubBand, Channel, DevLrrCnt, Lrrid, Late, LrrLAT, LrrLON, Lrrid_Lrr0, Chain_Lrr0, LrrRSSI_Lrr0, LrrSNR_Lrr0, LrrESP_Lrr0, Lrrid_Lrr1, Chain_Lrr1, LrrRSSI_Lrr1, LrrSNR_Lrr1, LrrESP_Lrr1, Lrrid_Lrr2, Chain_Lrr2, LrrRSSI_Lrr2, LrrSNR_Lrr2, LrrESP_Lrr2, Lrrid_Lrr3, Chain_Lrr3, LrrRSSI_Lrr3, LrrSNR_Lrr3, LrrESP_Lrr3, Lrrid_Lrr4, Chain_Lrr4, LrrRSSI_Lrr4, LrrSNR_Lrr4, LrrESP_Lrr4, Lrrid_Lrr5, Chain_Lrr5, LrrRSSI_Lrr5, LrrSNR_Lrr5, LrrESP_Lrr5, Lrrid_Lrr6, Chain_Lrr6, LrrRSSI_Lrr6, LrrSNR_Lrr6, LrrESP_Lrr6, Lrrid_Lrr7, Chain_Lrr7, LrrRSSI_Lrr7, LrrSNR_Lrr7, LrrESP_Lrr7, Lrrid_Lrr8, Chain_Lrr8, LrrRSSI_Lrr8, LrrSNR_Lrr8, LrrESP_Lrr8, Lrrid_Lrr9, Chain_Lrr9, LrrRSSI_Lrr9, LrrSNR_Lrr9, LrrESP_Lrr9, CustomerID, CustomerData_pro, CustomerData_ver, ModelCfg, InstantPER, MeanPER, DevAddr) VALUES ('" + msg.payload.final + "'" + "," + "'" + msg.payload.DevEUI_uplink.Time + "'" + "," + "'" + msg.payload.DevEUI_uplink.DevEUI + "'" + ", '" + msg.payload.DevEUI_uplink.FPort + "'" + "," + "'" + msg.payload.DevEUI_uplink.FCntUp + "'" + ", '" + msg.payload.DevEUI_uplink.ADRbit + "'" + ", '" + msg.payload.DevEUI_uplink.MType + "'" + ", '" + msg.payload.DevEUI_uplink.FCntDn + "'" + ", '" + msg.payload.DevEUI_uplink.payload_hex + "'" + ", '" + msg.payload.DevEUI_uplink.mic_hex + "'" + ", '" + msg.payload.DevEUI_uplink.Lrcid + "'" + ", '" + msg.payload.DevEUI_uplink.LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.SpFact + "'" + ", '" + msg.payload.DevEUI_uplink.SubBand + "'" + ", '" + msg.payload.DevEUI_uplink.Channel + "'" + ", '" + msg.payload.DevEUI_uplink.DevLrrCnt + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Late + "'" + ", '" + msg.payload.DevEUI_uplink.LrrLAT + "'" + ", '" + msg.payload.DevEUI_uplink.LrrLON + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].LrrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].LrrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].LrrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].LrrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].LrrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[5].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[5].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[5].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[5].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[5].LrrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[6].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[6].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[6].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[6].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[6].LrrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[7].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[7].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[7].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[7].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[7].LrrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[8].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[8].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[8].LrrRSSI + "'" + ", '" + msg.payload.DevEU
```

```
I_uplink.Lrrs.Lrr[8].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[8].LrrESP + "'" + ", '"  + msg.payload.DevEUI_uplink.Lrrs.Lrr[9].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[9].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[9].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[9].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[9].LrrESP + "'" + ", '" + msg.payload.DevEUI_uplink.CustomerID + "'" + ", '" + msg.payload.DevEUI_uplink.CustomerData.alr.pro + "'" + ", '" + msg.payload.DevEUI_uplink.CustomerData.alr.ver + "'" + ", '" + msg.payload.DevEUI_uplink.ModelCfg + "'" + ", '" + msg.payload.DevEUI_uplink.InstantPER + "'" + ", '" + msg.payload.DevEUI_uplink.MeanPER + "'" + ", '" + msg.payload.DevEUI_uplink.DevAddr + "'" + ");" ;
} else if (numberOfRows === 9){
    val1 = "INSERT INTO raw_data (Decrypted, Time, DevEUI, FPort, FCntUp, ADRbit, MType, FCntDn, payload_hex, mic_hex, Lrcid, LrrRSSI, LrrSNR, SpFact, SubBand, Channel, DevLrrCnt, Lrrid, Late, LrrLAT, LrrLON, Lrrid_Lrr0, Chain_Lrr0, LrrRSSI_Lrr0, LrrSNR_Lrr0, LrrESP_Lrr0, Lrrid_Lrr1, Chain_Lrr1, LrrRSSI_Lrr1, LrrSNR_Lrr1, LrrESP_Lrr1, Lrrid_Lrr2, Chain_Lrr2, LrrRSSI_Lrr2, LrrSNR_Lrr2, LrrESP_Lrr2, Lrrid_Lrr3, Chain_Lrr3, LrrRSSI_Lrr3, LrrSNR_Lrr3, LrrESP_Lrr3, Lrrid_Lrr4, Chain_Lrr4, LrrRSSI_Lrr4, LrrSNR_Lrr4, LrrESP_Lrr4, Lrrid_Lrr5, Chain_Lrr5, LrrRSSI_Lrr5, LrrSNR_Lrr5, LrrESP_Lrr5, Lrrid_Lrr6, Chain_Lrr6, LrrRSSI_Lrr6, LrrSNR_Lrr6, LrrESP_Lrr6, Lrrid_Lrr7, Chain_Lrr7, LrrRSSI_Lrr7, LrrSNR_Lrr7, LrrESP_Lrr7, Lrrid_Lrr8, Chain_Lrr8, LrrRSSI_Lrr8, LrrSNR_Lrr8, LrrESP_Lrr8, Lrrid_Lrr9, Chain_Lrr9, LrrRSSI_Lrr9, LrrSNR_Lrr9, LrrESP_Lrr9, CustomerID, CustomerData_pro, CustomerData_ver, ModelCfg, InstantPER, MeanPER, DevAddr) VALUES ('" + msg.payload.final + "'" + "," + "'" + msg.payload.DevEUI_uplink.Time + "'" + "," + "'" + msg.payload.DevEUI_uplink.DevEUI + "'" + ", '" + msg.payload.DevEUI_uplink.FPort + "'" + "," + "'" + msg.payload.DevEUI_uplink.FCntUp + "'" + ", '" + msg.payload.DevEUI_uplink.ADRbit + "'" + ", '" + msg.payload.DevEUI_uplink.MType + "'" + ", '" + msg.payload.DevEUI_uplink.FCntDn + "'" + ", '" + msg.payload.DevEUI_uplink.payload_hex + "'" + ", '" + msg.payload.DevEUI_uplink.mic_hex + "'" + ", '" + msg.payload.DevEUI_uplink.Lrcid + "'" + ", '" + msg.payload.DevEUI_uplink.LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.SpFact + "'" + ", '" + msg.payload.DevEUI_uplink.SubBand + "'" + ", '" + msg.payload.DevEUI_uplink.Channel + "'" + ", '" + msg.payload.DevEUI_uplink.DevLrrCnt + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Late + "'" + ", '" + msg.payload.DevEUI_uplink.LrrLAT + "'" + ", '" + msg.payload.DevEUI_uplink.LrrLON + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].LrrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].LrrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].LrrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].LrrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].LrrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[5].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[5].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[5].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[5].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[5].LrrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[6].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[6].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[6].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[6].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[6].LrrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[7].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[7].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[7].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[7].LrrSNR + "'" + ", '" +
```

```
 msg.payload.DevEUI_uplink.Lrrs.Lrr[7].LrrESP + "'" + ", '" + msg.payload.DevEUI_uplink
.Lrrs.Lrr[8].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[8].Chain + "'" +
", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[8].LrrRSSI + "'" + ", '" + msg.payload.DevEU
I_uplink.Lrrs.Lrr[8].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[8].LrrES
P + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' +
 "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + msg.payload.DevE
UI_uplink.CustomerID + "'" + ", '" + msg.payload.DevEUI_uplink.CustomerData.alr.pro + "
'" + ", '" + msg.payload.DevEUI_uplink.CustomerData.alr.ver + "'" + ", '" + msg.payload
.DevEUI_uplink.ModelCfg + "'" + ", '" + msg.payload.DevEUI_uplink.InstantPER + "'" + ",
 '" + msg.payload.DevEUI_uplink.MeanPER + "'" + ", '" + msg.payload.DevEUI_uplink.DevAd
dr + "'" + ");" ;
} else if (numberOfRows === 8){
    val1 = "INSERT INTO raw_data (Decrypted, Time, DevEUI, FPort, FCntUp, ADRbit, MType
, FCntDn, payload_hex, mic_hex, Lrcid, LrrRSSI, LrrSNR, SpFact, SubBand, Channel, DevLr
rCnt, Lrrid, Late, LrrLAT, LrrLON, Lrrid_Lrr0, Chain_Lrr0, LrrRSSI_Lrr0, LrrSNR_Lrr0, L
rrESP_Lrr0, Lrrid_Lrr1, Chain_Lrr1, LrrRSSI_Lrr1, LrrSNR_Lrr1, LrrESP_Lrr1, Lrrid_Lrr2,
 Chain_Lrr2, LrrRSSI_Lrr2, LrrSNR_Lrr2, LrrESP_Lrr2, Lrrid_Lrr3, Chain_Lrr3, LrrRSSI_Lr
r3, LrrSNR_Lrr3, LrrESP_Lrr3, Lrrid_Lrr4, Chain_Lrr4, LrrRSSI_Lrr4, LrrSNR_Lrr4, LrrESP
_Lrr4, Lrrid_Lrr5, Chain_Lrr5, LrrRSSI_Lrr5, LrrSNR_Lrr5, LrrESP_Lrr5, Lrrid_Lrr6, Chai
n_Lrr6, LrrRSSI_Lrr6, LrrSNR_Lrr6, LrrESP_Lrr6, Lrrid_Lrr7, Chain_Lrr7, LrrRSSI_Lrr7, L
rrSNR_Lrr7, LrrESP_Lrr7, Lrrid_Lrr8, Chain_Lrr8, LrrRSSI_Lrr8, LrrSNR_Lrr8, LrrESP_Lrr8
, Lrrid_Lrr9, Chain_Lrr9, LrrRSSI_Lrr9, LrrSNR_Lrr9, LrrESP_Lrr9, CustomerID, CustomerD
ata_pro, CustomerData_ver, ModelCfg, InstantPER, MeanPER, DevAddr) VALUES ('" + msg.pay
load.final + "'" + "," + "'" + msg.payload.DevEUI_uplink.Time + "'" + "," + "'" + msg.p
ayload.DevEUI_uplink.DevEUI + "'" + ", '" + msg.payload.DevEUI_uplink.FPort + "'" + ","
 + "'" + msg.payload.DevEUI_uplink.FCntUp + "'" + ", '" + msg.payload.DevEUI_uplink.ADR
bit + "'" + ", '" + msg.payload.DevEUI_uplink.MType + "'" + ", '" + msg.payload.DevEUI_
uplink.FCntDn + "'" + ", '" + msg.payload.DevEUI_uplink.payload_hex + "'" + ", '" + msg
.payload.DevEUI_uplink.mic_hex + "'" + ", '" + msg.payload.DevEUI_uplink.Lrcid + "'" +
", '" + msg.payload.DevEUI_uplink.LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrr
SNR + "'" + ", '" + msg.payload.DevEUI_uplink.SpFact + "'" + ", '" + msg.payload.DevEUI
_uplink.SubBand + "'" + ", '" + msg.payload.DevEUI_uplink.Channel + "'" + ", '" + msg.p
ayload.DevEUI_uplink.DevLrrCnt + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrid + "'" +
", '" + msg.payload.DevEUI_uplink.Late + "'" + ", '" + msg.payload.DevEUI_uplink.LrrLAT
 + "'" + ", '" + msg.payload.DevEUI_uplink.LrrLON + "'" + ", '" + msg.payload.DevEUI_up
link.Lrrs.Lrr[0].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].Chain + "'
" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].LrrRSSI + "'" + ", '" + msg.payload.D
evEUI_uplink.Lrrs.Lrr[0].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].L
rrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].Lrrid + "'" + ", '" + msg.p
ayload.DevEUI_uplink.Lrrs.Lrr[1].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.L
rr[1].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].LrrSNR + "'" + ", '
" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].LrrESP + "'" + ", '" + msg.payload.DevEUI_upl
ink.Lrrs.Lrr[2].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].Chain + "'"
 + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].LrrRSSI + "'" + ", '" + msg.payload.De
vEUI_uplink.Lrrs.Lrr[2].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].Lr
rESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].Lrrid + "'" + ", '" + msg.pa
yload.DevEUI_uplink.Lrrs.Lrr[3].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lr
r[3].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].LrrSNR + "'" + ", '"
 + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].LrrESP + "'" + ", '" + msg.payload.DevEUI_upli
nk.Lrrs.Lrr[4].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].Chain + "'"
 + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].LrrRSSI + "'" + ", '" + msg.payload.Dev
EUI_uplink.Lrrs.Lrr[4].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].Lrr
ESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[5].Lrrid + "'" + ", '" + msg.pay
load.DevEUI_uplink.Lrrs.Lrr[5].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr
[5].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[5].LrrSNR + "'" + ", '"
 + msg.payload.DevEUI_uplink.Lrrs.Lrr[5].LrrESP + "'" + ", '" + msg.payload.DevEUI_uplin
k.Lrrs.Lrr[6].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[6].Chain + "'" +
 ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[6].LrrRSSI + "'" + ", '" + msg.payload.DevE
UI_uplink.Lrrs.Lrr[6].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[6].LrrE
SP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[7].Lrrid + "'" + ", '" + msg.payl
oad.DevEUI_uplink.Lrrs.Lrr[7].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[
```

```
7].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[7].LrrSNR + "'" + ", '" +
 msg.payload.DevEUI_uplink.Lrrs.Lrr[7].LrrESP + "'" + ", '" + 'undefined' + "'" + ", '"
 + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" +
'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'un
defined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + msg.pa
yload.DevEUI_uplink.CustomerID + "'" + ", '" + msg.payload.DevEUI_uplink.CustomerData.a
lr.pro + "'" + ", '" + msg.payload.DevEUI_uplink.CustomerData.alr.ver + "'" + ", '" + m
sg.payload.DevEUI_uplink.ModelCfg + "'" + ", '" + msg.payload.DevEUI_uplink.InstantPER
+ "'" + ", '" + msg.payload.DevEUI_uplink.MeanPER + "'" + ", '" + msg.payload.DevEUI_up
link.DevAddr + "'" + ");" ;
} else if (numberOfRows === 7){
    val1 = "INSERT INTO raw_data (Decrypted, Time, DevEUI, FPort, FCntUp, ADRbit, MType
, FCntDn, payload_hex, mic_hex, Lrcid, LrrRSSI, LrrSNR, SpFact, SubBand, Channel, DevLr
rCnt, Lrrid, Late, LrrLAT, LrrLON, Lrrid_Lrr0, Chain_Lrr0, LrrRSSI_Lrr0, LrrSNR_Lrr0, L
rrESP_Lrr0, Lrrid_Lrr1, Chain_Lrr1, LrrRSSI_Lrr1, LrrSNR_Lrr1, LrrESP_Lrr1, Lrrid_Lrr2,
 Chain_Lrr2, LrrRSSI_Lrr2, LrrSNR_Lrr2, LrrESP_Lrr2, Lrrid_Lrr3, Chain_Lrr3, LrrRSSI_Lr
r3, LrrSNR_Lrr3, LrrESP_Lrr3, Lrrid_Lrr4, Chain_Lrr4, LrrRSSI_Lrr4, LrrSNR_Lrr4, LrrESP
_Lrr4, Lrrid_Lrr5, Chain_Lrr5, LrrRSSI_Lrr5, LrrSNR_Lrr5, LrrESP_Lrr5, Lrrid_Lrr6, Chai
n_Lrr6, LrrRSSI_Lrr6, LrrSNR_Lrr6, LrrESP_Lrr6, Lrrid_Lrr7, Chain_Lrr7, LrrRSSI_Lrr7, L
rrSNR_Lrr7, LrrESP_Lrr7, Lrrid_Lrr8, Chain_Lrr8, LrrRSSI_Lrr8, LrrSNR_Lrr8, LrrESP_Lrr8
, Lrrid_Lrr9, Chain_Lrr9, LrrRSSI_Lrr9, LrrSNR_Lrr9, LrrESP_Lrr9, CustomerID, CustomerD
ata_pro, CustomerData_ver, ModelCfg, InstantPER, MeanPER, DevAddr) VALUES ('" + msg.pay
load.final + "'" + "," + "'" + msg.payload.DevEUI_uplink.Time + "'" + "," + "'" + msg.p
ayload.DevEUI_uplink.DevEUI + "'" + ", '" + msg.payload.DevEUI_uplink.FPort + "'" + ","
 + "'" + msg.payload.DevEUI_uplink.FCntUp + "'" + ", '" + msg.payload.DevEUI_uplink.ADR
bit + "'" + ", '" + msg.payload.DevEUI_uplink.MType + "'" + ", '" + msg.payload.DevEUI_
uplink.FCntDn + "'" + ", '" + msg.payload.DevEUI_uplink.payload_hex + "'" + ", '" + msg
.payload.DevEUI_uplink.mic_hex + "'" + ", '" + msg.payload.DevEUI_uplink.Lrcid + "'" +
", '" + msg.payload.DevEUI_uplink.LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrr
SNR + "'" + ", '" + msg.payload.DevEUI_uplink.SpFact + "'" + ", '" + msg.payload.DevEUI
_uplink.SubBand + "'" + ", '" + msg.payload.DevEUI_uplink.Channel + "'" + ", '" + msg.p
ayload.DevEUI_uplink.DevLrrCnt + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrid + "'" +
", '" + msg.payload.DevEUI_uplink.Late + "'" + ", '" + msg.payload.DevEUI_uplink.LrrLAT
 + "'" + ", '" + msg.payload.DevEUI_uplink.LrrLON + "'" + ", '" + msg.payload.DevEUI_up
link.Lrrs.Lrr[0].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].Chain + "'
" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].LrrRSSI + "'" + ", '" + msg.payload.D
evEUI_uplink.Lrrs.Lrr[0].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].L
rrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].Lrrid + "'" + ", '" + msg.p
ayload.DevEUI_uplink.Lrrs.Lrr[1].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.L
rr[1].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].LrrSNR + "'" + ", '
" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].LrrESP + "'" + ", '" + msg.payload.DevEUI_upl
ink.Lrrs.Lrr[2].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].Chain + "'"
 + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].LrrRSSI + "'" + ", '" + msg.payload.De
vEUI_uplink.Lrrs.Lrr[2].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].Lr
rESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].Lrrid + "'" + ", '" + msg.pa
yload.DevEUI_uplink.Lrrs.Lrr[3].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lr
r[3].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].LrrSNR + "'" + ", '"
 + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].LrrESP + "'" + ", '" + msg.payload.DevEUI_upli
nk.Lrrs.Lrr[4].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].Chain + "'"
 + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].LrrRSSI + "'" + ", '" + msg.payload.Dev
EUI_uplink.Lrrs.Lrr[4].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].Lrr
ESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[5].Lrrid + "'" + ", '" + msg.pay
load.DevEUI_uplink.Lrrs.Lrr[5].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr
[5].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[5].LrrSNR + "'" + ", '"
 + msg.payload.DevEUI_uplink.Lrrs.Lrr[5].LrrESP + "'" + ", '" + msg.payload.DevEUI_uplin
k.Lrrs.Lrr[6].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[6].Chain + "'" +
 ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[6].LrrRSSI + "'" + ", '" + msg.payload.DevE
UI_uplink.Lrrs.Lrr[6].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[6].LrrE
SP + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined'
+ "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "
'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'"
```

```
    + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + "
    , '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '
    " + msg.payload.DevEUI_uplink.CustomerID + "'" + ", '" + msg.payload.DevEUI_uplink.Cust
    omerData.alr.pro + "'" + ", '" + msg.payload.DevEUI_uplink.CustomerData.alr.ver + "'" +
     ", '" + msg.payload.DevEUI_uplink.ModelCfg + "'" + ", '" + msg.payload.DevEUI_uplink.I
    nstantPER + "'" + ", '" + msg.payload.DevEUI_uplink.MeanPER + "'" + ", '" + msg.payload
    .DevEUI_uplink.DevAddr + "');" ;
11. } else if (numberOfRows === 6){
12.     val1 = "INSERT INTO raw_data (Decrypted, Time, DevEUI, FPort, FCntUp, ADRbit, MType
    , FCntDn, payload_hex, mic_hex, Lrcid, LrrRSSI, LrrSNR, SpFact, SubBand, Channel, DevLr
    rCnt, Lrrid, Late, LrrLAT, LrrLON, Lrrid_Lrr0, Chain_Lrr0, LrrRSSI_Lrr0, LrrSNR_Lrr0, L
    rrESP_Lrr0, Lrrid_Lrr1, Chain_Lrr1, LrrRSSI_Lrr1, LrrSNR_Lrr1, LrrESP_Lrr1, Lrrid_Lrr2,
     Chain_Lrr2, LrrRSSI_Lrr2, LrrSNR_Lrr2, LrrESP_Lrr2, Lrrid_Lrr3, Chain_Lrr3, LrrRSSI_Lr
    r3, LrrSNR_Lrr3, LrrESP_Lrr3, Lrrid_Lrr4, Chain_Lrr4, LrrRSSI_Lrr4, LrrSNR_Lrr4, LrrESP
    _Lrr4, Lrrid_Lrr5, Chain_Lrr5, LrrRSSI_Lrr5, LrrSNR_Lrr5, LrrESP_Lrr5, Lrrid_Lrr6, Chai
    n_Lrr6, LrrRSSI_Lrr6, LrrSNR_Lrr6, LrrESP_Lrr6, Lrrid_Lrr7, Chain_Lrr7, LrrRSSI_Lrr7, L
    rrSNR_Lrr7, LrrESP_Lrr7, Lrrid_Lrr8, Chain_Lrr8, LrrRSSI_Lrr8, LrrSNR_Lrr8, LrrESP_Lrr8
    , Lrrid_Lrr9, Chain_Lrr9, LrrRSSI_Lrr9, LrrSNR_Lrr9, LrrESP_Lrr9, CustomerID, CustomerD
    ata_pro, CustomerData_ver, ModelCfg, InstantPER, MeanPER, DevAddr) VALUES ('" + msg.pay
    load.final + "'" + "," + "'" + msg.payload.DevEUI_uplink.Time + "'" + "," + "'" + msg.p
    ayload.DevEUI_uplink.DevEUI + "'" + ", '" + msg.payload.DevEUI_uplink.FPort + "'" + ","
     + "'" + msg.payload.DevEUI_uplink.FCntUp + "'" + ", '" + msg.payload.DevEUI_uplink.ADR
    bit + "'" + ", '" + msg.payload.DevEUI_uplink.MType + "'" + ", '" + msg.payload.DevEUI_
    uplink.FCntDn + "'" + ", '" + msg.payload.DevEUI_uplink.payload_hex + "'" + ", '" + msg
    .payload.DevEUI_uplink.mic_hex + "'" + ", '" + msg.payload.DevEUI_uplink.Lrcid + "'" +
    ", '" + msg.payload.DevEUI_uplink.LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrr
    SNR + "'" + ", '" + msg.payload.DevEUI_uplink.SpFact + "'" + ", '" + msg.payload.DevEUI
    _uplink.SubBand + "'" + ", '" + msg.payload.DevEUI_uplink.Channel + "'" + ", '" + msg.p
    ayload.DevEUI_uplink.DevLrrCnt + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrid + "'" +
    ", '" + msg.payload.DevEUI_uplink.Late + "'" + ", '" + msg.payload.DevEUI_uplink.LrrLAT
     + "'" + ", '" + msg.payload.DevEUI_uplink.LrrLON + "'" + ", '" + msg.payload.DevEUI_up
    link.Lrrs.Lrr[0].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].Chain + "'
    " + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].LrrRSSI + "'" + ", '" + msg.payload.D
    evEUI_uplink.Lrrs.Lrr[0].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].L
    rrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].Lrrid + "'" + ", '" + msg.p
    ayload.DevEUI_uplink.Lrrs.Lrr[1].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.L
    rr[1].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].LrrSNR + "'" + ", '
    " + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].LrrESP + "'" + ", '" + msg.payload.DevEUI_upl
    ink.Lrrs.Lrr[2].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].Chain + "'"
     + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].LrrRSSI + "'" + ", '" + msg.payload.De
    vEUI_uplink.Lrrs.Lrr[2].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].Lr
    rESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].Lrrid + "'" + ", '" + msg.pa
    yload.DevEUI_uplink.Lrrs.Lrr[3].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lr
    r[3].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].LrrSNR + "'" + ", '"
     + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].LrrESP + "'" + ", '" + msg.payload.DevEUI_upli
    nk.Lrrs.Lrr[4].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].Chain + "'"
    + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].LrrRSSI + "'" + ", '" + msg.payload.Dev
    EUI_uplink.Lrrs.Lrr[4].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].Lrr
    ESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[5].Lrrid + "'" + ", '" + msg.pay
    load.DevEUI_uplink.Lrrs.Lrr[5].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr
    [5].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[5].LrrSNR + "'" + ", '"
    + msg.payload.DevEUI_uplink.Lrrs.Lrr[5].LrrESP + "'" + ", '" + 'undefined' + "'" + ", '
    " + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" +
     'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'u
    ndefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'unde
    fined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefin
    ed' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined'
     + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' +
    "'" + ", '" + msg.payload.DevEUI_uplink.CustomerID + "'" + ", '" + msg.payload.DevEUI_u
    plink.CustomerData.alr.pro + "'" + ", '" + msg.payload.DevEUI_uplink.CustomerData.alr.v
    er + "'" + ", '" + msg.payload.DevEUI_uplink.ModelCfg + "'" + ", '" + msg.payload.DevEU
```

```
I_uplink.InstantPER + "'" + ", '" + msg.payload.DevEUI_uplink.MeanPER + "'" + ", '" + m
sg.payload.DevEUI_uplink.DevAddr + "'" + ");" ;
} else if (numberOfRows === 5){
    val1 = "INSERT INTO raw_data (Decrypted, Time, DevEUI, FPort, FCntUp, ADRbit, MType
, FCntDn, payload_hex, mic_hex, Lrcid, LrrRSSI, LrrSNR, SpFact, SubBand, Channel, DevLr
rCnt, Lrrid, Late, LrrLAT, LrrLON, Lrrid_Lrr0, Chain_Lrr0, LrrRSSI_Lrr0, LrrSNR_Lrr0, L
rrESP_Lrr0, Lrrid_Lrr1, Chain_Lrr1, LrrRSSI_Lrr1, LrrSNR_Lrr1, LrrESP_Lrr1, Lrrid_Lrr2,
 Chain_Lrr2, LrrRSSI_Lrr2, LrrSNR_Lrr2, LrrESP_Lrr2, Lrrid_Lrr3, Chain_Lrr3, LrrRSSI_Lr
r3, LrrSNR_Lrr3, LrrESP_Lrr3, Lrrid_Lrr4, Chain_Lrr4, LrrRSSI_Lrr4, LrrSNR_Lrr4, LrrESP
_Lrr4, Lrrid_Lrr5, Chain_Lrr5, LrrRSSI_Lrr5, LrrSNR_Lrr5, LrrESP_Lrr5, Lrrid_Lrr6, Chai
n_Lrr6, LrrRSSI_Lrr6, LrrSNR_Lrr6, LrrESP_Lrr6, Lrrid_Lrr7, Chain_Lrr7, LrrRSSI_Lrr7, L
rrSNR_Lrr7, LrrESP_Lrr7, Lrrid_Lrr8, Chain_Lrr8, LrrRSSI_Lrr8, LrrSNR_Lrr8, LrrESP_Lrr8
, Lrrid_Lrr9, Chain_Lrr9, LrrRSSI_Lrr9, LrrSNR_Lrr9, LrrESP_Lrr9, CustomerID, CustomerD
ata_pro, CustomerData_ver, ModelCfg, InstantPER, MeanPER, DevAddr) VALUES ('" + msg.pay
load.final + "'" + "," + "'" + msg.payload.DevEUI_uplink.Time + "'" + "," + "'" + msg.p
ayload.DevEUI_uplink.DevEUI + "'" + ", '" + msg.payload.DevEUI_uplink.FPort + "'" + ","
 + "'" + msg.payload.DevEUI_uplink.FCntUp + "'" + ", '" + msg.payload.DevEUI_uplink.ADR
bit + "'" + ", '" + msg.payload.DevEUI_uplink.MType + "'" + ", '" + msg.payload.DevEUI_
uplink.FCntDn + "'" + ", '" + msg.payload.DevEUI_uplink.payload_hex + "'" + ", '" + msg
.payload.DevEUI_uplink.mic_hex + "'" + ", '" + msg.payload.DevEUI_uplink.Lrcid + "'" +
", '" + msg.payload.DevEUI_uplink.LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrr
SNR + "'" + ", '" + msg.payload.DevEUI_uplink.SpFact + "'" + ", '" + msg.payload.DevEUI
_uplink.SubBand + "'" + ", '" + msg.payload.DevEUI_uplink.Channel + "'" + ", '" + msg.p
ayload.DevEUI_uplink.DevLrrCnt + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrid + "'" +
", '" + msg.payload.DevEUI_uplink.Late + "'" + ", '" + msg.payload.DevEUI_uplink.LrrLAT
 + "'" + ", '" + msg.payload.DevEUI_uplink.LrrLON + "'" + ", '" + msg.payload.DevEUI_up
link.Lrrs.Lrr[0].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].Chain + "'
" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].LrrRSSI + "'" + ", '" + msg.payload.D
evEUI_uplink.Lrrs.Lrr[0].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].L
rrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].Lrrid + "'" + ", '" + msg.p
ayload.DevEUI_uplink.Lrrs.Lrr[1].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.L
rr[1].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].LrrSNR + "'" + ", '
" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].LrrESP + "'" + ", '" + msg.payload.DevEUI_upl
ink.Lrrs.Lrr[2].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].Chain + "'"
 + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].LrrRSSI + "'" + ", '" + msg.payload.De
vEUI_uplink.Lrrs.Lrr[2].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].Lr
rESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].Lrrid + "'" + ", '" + msg.pa
yload.DevEUI_uplink.Lrrs.Lrr[3].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lr
r[3].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].LrrSNR + "'" + ", '"
 + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].LrrESP + "'" + ", '" + msg.payload.DevEUI_upli
nk.Lrrs.Lrr[4].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].Chain + "'"
+ ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].LrrRSSI + "'" + ", '" + msg.payload.Dev
EUI_uplink.Lrrs.Lrr[4].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[4].Lrr
ESP + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined'
 + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' +
"'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'"
 + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" +
", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ",
'" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '"
+ 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + '
undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'und
efined' + "'" + ", '" + msg.payload.DevEUI_uplink.CustomerID + "'" + ", '" + msg.payloa
d.DevEUI_uplink.CustomerData.alr.pro + "'" + ", '" + msg.payload.DevEUI_uplink.Customer
Data.alr.ver + "'" + ", '" + msg.payload.DevEUI_uplink.ModelCfg + "'" + ", '" + msg.pay
load.DevEUI_uplink.InstantPER + "'" + ", '" + msg.payload.DevEUI_uplink.MeanPER + "'" +
 ", '" + msg.payload.DevEUI_uplink.DevAddr + "'" + ");" ;
} else if (numberOfRows === 4){
    val1 = "INSERT INTO raw_data (Decrypted, Time, DevEUI, FPort, FCntUp, ADRbit, MType
, FCntDn, payload_hex, mic_hex, Lrcid, LrrRSSI, LrrSNR, SpFact, SubBand, Channel, DevLr
rCnt, Lrrid, Late, LrrLAT, LrrLON, Lrrid_Lrr0, Chain_Lrr0, LrrRSSI_Lrr0, LrrSNR_Lrr0, L
rrESP_Lrr0, Lrrid_Lrr1, Chain_Lrr1, LrrRSSI_Lrr1, LrrSNR_Lrr1, LrrESP_Lrr1, Lrrid_Lrr2,
```

```
 Chain_Lrr2, LrrRSSI_Lrr2, LrrSNR_Lrr2, LrrESP_Lrr2, Lrrid_Lrr3, Chain_Lrr3, LrrRSSI_Lr
r3, LrrSNR_Lrr3, LrrESP_Lrr3, Lrrid_Lrr4, Chain_Lrr4, LrrRSSI_Lrr4, LrrSNR_Lrr4, LrrESP
_Lrr4, Lrrid_Lrr5, Chain_Lrr5, LrrRSSI_Lrr5, LrrSNR_Lrr5, LrrESP_Lrr5, Lrrid_Lrr6, Chai
n_Lrr6, LrrRSSI_Lrr6, LrrSNR_Lrr6, LrrESP_Lrr6, Lrrid_Lrr7, Chain_Lrr7, LrrRSSI_Lrr7, L
rrSNR_Lrr7, LrrESP_Lrr7, Lrrid_Lrr8, Chain_Lrr8, LrrRSSI_Lrr8, LrrSNR_Lrr8, LrrESP_Lrr8
, Lrrid_Lrr9, Chain_Lrr9, LrrRSSI_Lrr9, LrrSNR_Lrr9, LrrESP_Lrr9, CustomerID, CustomerD
ata_pro, CustomerData_ver, ModelCfg, InstantPER, MeanPER, DevAddr) VALUES ('" + msg.pay
load.final + "'" + "," + "'" + msg.payload.DevEUI_uplink.Time + "'" + "," + "'" + msg.p
ayload.DevEUI_uplink.DevEUI + "'" + ", '" + msg.payload.DevEUI_uplink.FPort + "'" + ","
 + "'" + msg.payload.DevEUI_uplink.FCntUp + "'" + ", '" + msg.payload.DevEUI_uplink.ADR
bit + "'" + ", '" + msg.payload.DevEUI_uplink.MType + "'" + ", '" + msg.payload.DevEUI_
uplink.FCntDn + "'" + ", '" + msg.payload.DevEUI_uplink.payload_hex + "'" + ", '" + msg
.payload.DevEUI_uplink.mic_hex + "'" + ", '" + msg.payload.DevEUI_uplink.Lrcid + "'" +
", '" + msg.payload.DevEUI_uplink.LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrr
SNR + "'" + ", '" + msg.payload.DevEUI_uplink.SpFact + "'" + ", '" + msg.payload.DevEUI
_uplink.SubBand + "'" + ", '" + msg.payload.DevEUI_uplink.Channel + "'" + ", '" + msg.p
ayload.DevEUI_uplink.DevLrrCnt + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrid + "'" +
", '" + msg.payload.DevEUI_uplink.Late + "'" + ", '" + msg.payload.DevEUI_uplink.LrrLAT
 + "'" + ", '" + msg.payload.DevEUI_uplink.LrrLON + "'" + ", '" + msg.payload.DevEUI_up
link.Lrrs.Lrr[0].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].Chain + "'
" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].LrrRSSI + "'" + ", '" + msg.payload.D
evEUI_uplink.Lrrs.Lrr[0].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].L
rrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].Lrrid + "'" + ", '" + msg.p
ayload.DevEUI_uplink.Lrrs.Lrr[1].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.L
rr[1].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].LrrSNR + "'" + ", '
" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].LrrESP + "'" + ", '" + msg.payload.DevEUI_upl
ink.Lrrs.Lrr[2].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].Chain + "'"
 + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].LrrRSSI + "'" + ", '" + msg.payload.De
vEUI_uplink.Lrrs.Lrr[2].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].Lr
rESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].Lrrid + "'" + ", '" + msg.pa
yload.DevEUI_uplink.Lrrs.Lrr[3].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lr
r[3].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].LrrSNR + "'" + ", '"
 + msg.payload.DevEUI_uplink.Lrrs.Lrr[3].LrrESP + "'" + ", '" + 'undefined' + "'" + ",
'" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '"
+ 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + '
undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'und
efined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefi
ned' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined
' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' +
 "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'
" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" +
 ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ",
 '" + 'undefined' + "'" + ", '" + msg.payload.DevEUI_uplink.CustomerID + "'" + ", '" +
msg.payload.DevEUI_uplink.CustomerData.alr.pro + "'" + ", '" + msg.payload.DevEUI_uplin
k.CustomerData.alr.ver + "'" + ", '" + msg.payload.DevEUI_uplink.ModelCfg + "'" + ", '"
 + msg.payload.DevEUI_uplink.InstantPER + "'" + ", '" + msg.payload.DevEUI_uplink.MeanP
ER + "'" + ", '" + msg.payload.DevEUI_uplink.DevAddr + "'" + ");" ;
} else if (numberOfRows === 3){
    val1 = "INSERT INTO raw_data (Decrypted, Time, DevEUI, FPort, FCntUp, ADRbit, MType
, FCntDn, payload_hex, mic_hex, Lrcid, LrrRSSI, LrrSNR, SpFact, SubBand, Channel, DevLr
rCnt, Lrrid, Late, LrrLAT, LrrLON, Lrrid_Lrr0, Chain_Lrr0, LrrRSSI_Lrr0, LrrSNR_Lrr0, L
rrESP_Lrr0, Lrrid_Lrr1, Chain_Lrr1, LrrRSSI_Lrr1, LrrSNR_Lrr1, LrrESP_Lrr1, Lrrid_Lrr2,
 Chain_Lrr2, LrrRSSI_Lrr2, LrrSNR_Lrr2, LrrESP_Lrr2, Lrrid_Lrr3, Chain_Lrr3, LrrRSSI_Lr
r3, LrrSNR_Lrr3, LrrESP_Lrr3, Lrrid_Lrr4, Chain_Lrr4, LrrRSSI_Lrr4, LrrSNR_Lrr4, LrrESP
_Lrr4, Lrrid_Lrr5, Chain_Lrr5, LrrRSSI_Lrr5, LrrSNR_Lrr5, LrrESP_Lrr5, Lrrid_Lrr6, Chai
n_Lrr6, LrrRSSI_Lrr6, LrrSNR_Lrr6, LrrESP_Lrr6, Lrrid_Lrr7, Chain_Lrr7, LrrRSSI_Lrr7, L
rrSNR_Lrr7, LrrESP_Lrr7, Lrrid_Lrr8, Chain_Lrr8, LrrRSSI_Lrr8, LrrSNR_Lrr8, LrrESP_Lrr8
, Lrrid_Lrr9, Chain_Lrr9, LrrRSSI_Lrr9, LrrSNR_Lrr9, LrrESP_Lrr9, CustomerID, CustomerD
ata_pro, CustomerData_ver, ModelCfg, InstantPER, MeanPER, DevAddr) VALUES ('" + msg.pay
load.final + "'" + "," + "'" + msg.payload.DevEUI_uplink.Time + "'" + "," + "'" + msg.p
ayload.DevEUI_uplink.DevEUI + "'" + ", '" + msg.payload.DevEUI_uplink.FPort + "'" + ","
```

```
 + "'" + msg.payload.DevEUI_uplink.FCntUp + "'" + ", '" + msg.payload.DevEUI_uplink.ADR
bit + "'" + ", '" + msg.payload.DevEUI_uplink.MType + "'" + ", '" + msg.payload.DevEUI_
uplink.FCntDn + "'" + ", '" + msg.payload.DevEUI_uplink.payload_hex + "'" + ", '" + msg
.payload.DevEUI_uplink.mic_hex + "'" + ", '" + msg.payload.DevEUI_uplink.Lrcid + "'" +
", '" + msg.payload.DevEUI_uplink.LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrr
SNR + "'" + ", '" + msg.payload.DevEUI_uplink.SpFact + "'" + ", '" + msg.payload.DevEUI
_uplink.SubBand + "'" + ", '" + msg.payload.DevEUI_uplink.Channel + "'" + ", '" + msg.p
ayload.DevEUI_uplink.DevLrrCnt + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrid + "'" +
", '" + msg.payload.DevEUI_uplink.Late + "'" + ", '" + msg.payload.DevEUI_uplink.LrrLAT
 + "'" + ", '" + msg.payload.DevEUI_uplink.LrrLON + "'" + ", '" + msg.payload.DevEUI_up
link.Lrrs.Lrr[0].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].Chain + "'
" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].LrrRSSI + "'" + ", '" + msg.payload.D
evEUI_uplink.Lrrs.Lrr[0].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].L
rrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].Lrrid + "'" + ", '" + msg.p
ayload.DevEUI_uplink.Lrrs.Lrr[1].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.L
rr[1].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].LrrSNR + "'" + ", '
" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].LrrESP + "'" + ", '" + msg.payload.DevEUI_upl
ink.Lrrs.Lrr[2].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].Chain + "'"
 + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].LrrRSSI + "'" + ", '" + msg.payload.De
vEUI_uplink.Lrrs.Lrr[2].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[2].Lr
rESP + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined
' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' +
 "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'
" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" +
 ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ",
 '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '"
 + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" +
'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'un
defined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undef
ined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefine
d' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined'
+ "'" + ", '" + 'undefined' + "'" + ", '" + msg.payload.DevEUI_uplink.CustomerID + "'"
+ ", '" + msg.payload.DevEUI_uplink.CustomerData.alr.pro + "'" + ", '" + msg.payload.De
vEUI_uplink.CustomerData.alr.ver + "'" + ", '" + msg.payload.DevEUI_uplink.ModelCfg + "
'" + ", '" + msg.payload.DevEUI_uplink.InstantPER + "'" + ", '" + msg.payload.DevEUI_up
link.MeanPER + "'" + ", '" + msg.payload.DevEUI_uplink.DevAddr + "'" + ");" ;
} else if (numberOfRows === 2){
    val1 = "INSERT INTO raw_data (Decrypted, Time, DevEUI, FPort, FCntUp, ADRbit, MType
, FCntDn, payload_hex, mic_hex, Lrcid, LrrRSSI, LrrSNR, SpFact, SubBand, Channel, DevLr
rCnt, Lrrid, Late, LrrLAT, LrrLON, Lrrid_Lrr0, Chain_Lrr0, LrrRSSI_Lrr0, LrrSNR_Lrr0, L
rrESP_Lrr0, Lrrid_Lrr1, Chain_Lrr1, LrrRSSI_Lrr1, LrrSNR_Lrr1, LrrESP_Lrr1, Lrrid_Lrr2,
 Chain_Lrr2, LrrRSSI_Lrr2, LrrSNR_Lrr2, LrrESP_Lrr2, Lrrid_Lrr3, Chain_Lrr3, LrrRSSI_Lr
r3, LrrSNR_Lrr3, LrrESP_Lrr3, Lrrid_Lrr4, Chain_Lrr4, LrrRSSI_Lrr4, LrrSNR_Lrr4, LrrESP
_Lrr4, Lrrid_Lrr5, Chain_Lrr5, LrrRSSI_Lrr5, LrrSNR_Lrr5, LrrESP_Lrr5, Lrrid_Lrr6, Chai
n_Lrr6, LrrRSSI_Lrr6, LrrSNR_Lrr6, LrrESP_Lrr6, Lrrid_Lrr7, Chain_Lrr7, LrrRSSI_Lrr7, L
rrSNR_Lrr7, LrrESP_Lrr7, Lrrid_Lrr8, Chain_Lrr8, LrrRSSI_Lrr8, LrrSNR_Lrr8, LrrESP_Lrr8
, Lrrid_Lrr9, Chain_Lrr9, LrrRSSI_Lrr9, LrrSNR_Lrr9, LrrESP_Lrr9, CustomerID, CustomerD
ata_pro, CustomerData_ver, ModelCfg, InstantPER, MeanPER, DevAddr) VALUES ('" + msg.pay
load.final + "'" + "," + "'" + msg.payload.DevEUI_uplink.Time + "'" + "," + "'" + msg.p
ayload.DevEUI_uplink.DevEUI + "'" + ", '" + msg.payload.DevEUI_uplink.FPort + "'" + ","
 + "'" + msg.payload.DevEUI_uplink.FCntUp + "'" + ", '" + msg.payload.DevEUI_uplink.ADR
bit + "'" + ", '" + msg.payload.DevEUI_uplink.MType + "'" + ", '" + msg.payload.DevEUI_
uplink.FCntDn + "'" + ", '" + msg.payload.DevEUI_uplink.payload_hex + "'" + ", '" + msg
.payload.DevEUI_uplink.mic_hex + "'" + ", '" + msg.payload.DevEUI_uplink.Lrcid + "'" +
", '" + msg.payload.DevEUI_uplink.LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrr
SNR + "'" + ", '" + msg.payload.DevEUI_uplink.SpFact + "'" + ", '" + msg.payload.DevEUI
_uplink.SubBand + "'" + ", '" + msg.payload.DevEUI_uplink.Channel + "'" + ", '" + msg.p
ayload.DevEUI_uplink.DevLrrCnt + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrid + "'" +
", '" + msg.payload.DevEUI_uplink.Late + "'" + ", '" + msg.payload.DevEUI_uplink.LrrLAT
 + "'" + ", '" + msg.payload.DevEUI_uplink.LrrLON + "'" + ", '" + msg.payload.DevEUI_up
link.Lrrs.Lrr[0].Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].Chain + "'
```

```
" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].LrrRSSI + "'" + ", '" + msg.payload.D
evEUI_uplink.Lrrs.Lrr[0].LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[0].L
rrESP + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].Lrrid + "'" + ", '" + msg.p
ayload.DevEUI_uplink.Lrrs.Lrr[1].Chain + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.L
rr[1].LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].LrrSNR + "'" + ", '
" + msg.payload.DevEUI_uplink.Lrrs.Lrr[1].LrrESP + "'" + ", '" + 'undefined' + "'" + ",
 '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '"
 + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" +
'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '"+ 'und
efined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefi
ned' + "'" + ", '" + 'undefined' + "'" + ", '"+ 'undefined' + "'" + ", '" + 'undefined'
 + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' +
"'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'"
 + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" +
", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ",
'" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '"
+ 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + '
undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'und
efined' + "'" + ", '" + 'undefined' + "'" + ", '" + msg.payload.DevEUI_uplink.CustomerI
D + "'" + ", '" + msg.payload.DevEUI_uplink.CustomerData.alr.pro + "'" + ", '" + msg.pa
yload.DevEUI_uplink.CustomerData.alr.ver + "'" + ", '" + msg.payload.DevEUI_uplink.Mode
lCfg + "'" + ", '" + msg.payload.DevEUI_uplink.InstantPER + "'" + ", '" + msg.payload.D
evEUI_uplink.MeanPER + "'" + ", '" + msg.payload.DevEUI_uplink.DevAddr + "'" + ");" ;
} else {
    val1 = "INSERT INTO raw_data (Decrypted, Time, DevEUI, FPort, FCntUp, ADRbit, MType
, FCntDn, payload_hex, mic_hex, Lrcid, LrrRSSI, LrrSNR, SpFact, SubBand, Channel, DevLr
rCnt, Lrrid, Late, LrrLAT, LrrLON, Lrrid_Lrr0, Chain_Lrr0, LrrRSSI_Lrr0, LrrSNR_Lrr0, L
rrESP_Lrr0, Lrrid_Lrr1, Chain_Lrr1, LrrRSSI_Lrr1, LrrSNR_Lrr1, LrrESP_Lrr1, Lrrid_Lrr2,
 Chain_Lrr2, LrrRSSI_Lrr2, LrrSNR_Lrr2, LrrESP_Lrr2, Lrrid_Lrr3, Chain_Lrr3, LrrRSSI_Lr
r3, LrrSNR_Lrr3, LrrESP_Lrr3, Lrrid_Lrr4, Chain_Lrr4, LrrRSSI_Lrr4, LrrSNR_Lrr4, LrrESP
_Lrr4, Lrrid_Lrr5, Chain_Lrr5, LrrRSSI_Lrr5, LrrSNR_Lrr5, LrrESP_Lrr5, Lrrid_Lrr6, Chai
n_Lrr6, LrrRSSI_Lrr6, LrrSNR_Lrr6, LrrESP_Lrr6, Lrrid_Lrr7, Chain_Lrr7, LrrRSSI_Lrr7, L
rrSNR_Lrr7, LrrESP_Lrr7, Lrrid_Lrr8, Chain_Lrr8, LrrRSSI_Lrr8, LrrSNR_Lrr8, LrrESP_Lrr8
, Lrrid_Lrr9, Chain_Lrr9, LrrRSSI_Lrr9, LrrSNR_Lrr9, LrrESP_Lrr9, CustomerID, CustomerD
ata_pro, CustomerData_ver, ModelCfg, InstantPER, MeanPER, DevAddr) VALUES ('" + msg.pay
load.final + "'" + "," + "'" + msg.payload.DevEUI_uplink.Time + "'" + "," + "'" + msg.p
ayload.DevEUI_uplink.DevEUI + "'" + ", '" + msg.payload.DevEUI_uplink.FPort + "'" + ","
 + "'" + msg.payload.DevEUI_uplink.FCntUp + "'" + ", '" + msg.payload.DevEUI_uplink.ADR
bit + "'" + ", '" + msg.payload.DevEUI_uplink.MType + "'" + ", '" + msg.payload.DevEUI_
uplink.FCntDn + "'" + ", '" + msg.payload.DevEUI_uplink.payload_hex + "'" + ", '" + msg
.payload.DevEUI_uplink.mic_hex + "'" + ", '" + msg.payload.DevEUI_uplink.Lrcid + "'" +
", '" + msg.payload.DevEUI_uplink.LrrRSSI + "'" + ", '" + msg.payload.DevEUI_uplink.Lrr
SNR + "'" + ", '" + msg.payload.DevEUI_uplink.SpFact + "'" + ", '" + msg.payload.DevEUI
_uplink.SubBand + "'" + ", '" + msg.payload.DevEUI_uplink.Channel + "'" + ", '" + msg.p
ayload.DevEUI_uplink.DevLrrCnt + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrid + "'" +
", '" + msg.payload.DevEUI_uplink.Late + "'" + ", '" + msg.payload.DevEUI_uplink.LrrLAT
 + "'" + ", '" + msg.payload.DevEUI_uplink.LrrLON + "'" + ", '" + msg.payload.DevEUI_up
link.Lrrs.Lrr.Lrrid + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr.Chain + "'" + ",
 '" + msg.payload.DevEUI_uplink.Lrrs.Lrr.LrrRSSI + "'" + ", '" + msg.payload.DevEUI_upl
ink.Lrrs.Lrr.LrrSNR + "'" + ", '" + msg.payload.DevEUI_uplink.Lrrs.Lrr.LrrESP + "'" + "
, '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '
" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" +
 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'u
ndefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'unde
fined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefin
ed' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined'
 + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' +
"'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'"
 + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" +
", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ",
'" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '"
```

```
    + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + '
    undefined' + "'" + ", '" + 'undefined' + "'" + ", '"+ 'undefined' + "'" + ", '" + 'unde
    fined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefined' + "'" + ", '" + 'undefin
    ed' + "'" + ", '" + 'undefined' + "'" + ", '" + msg.payload.DevEUI_uplink.CustomerID +
    "'" + ", '" + msg.payload.DevEUI_uplink.CustomerData.alr.pro + "'" + ", '" + msg.payloa
    d.DevEUI_uplink.CustomerData.alr.ver + "'" + ", '" + msg.payload.DevEUI_uplink.ModelCfg
     + "'" + ", '" + msg.payload.DevEUI_uplink.InstantPER + "'" + ", '" + msg.payload.DevEU
    I_uplink.MeanPER + "'" + ", '" + msg.payload.DevEUI_uplink.DevAddr + "'" + ");" ;
}
msg.payload = val1;
return msg;
```

Appendix C – Message to *Google Maps Geolocation API*

```
msg.payload = '{\n  "considerIp": "false",\n  "wifiAccessPoints": [\n    {\n        "macAddress": "' + msg.payload.mac1 + '",\n        "signalStrength":' + msg.payload.strength1 + ',\n        "signalToNoiseRatio": 0\n    },\n    {\n        "macAddress": "' + msg.payload.mac2 + '",\n        "signalStrength":' + msg.payload.strength2 + ',\n        "signalToNoiseRatio": 0\n    },\n    {\n        "macAddress": "' + msg.payload.mac3 + '",\n        "signalStrength":' + msg.payload.strength3 + ',\n        "signalToNoiseRatio": 0\n    }\n  ]\n}';
msg.headers = {
        'Content-Type' : 'application/json'
    };
return msg;
```

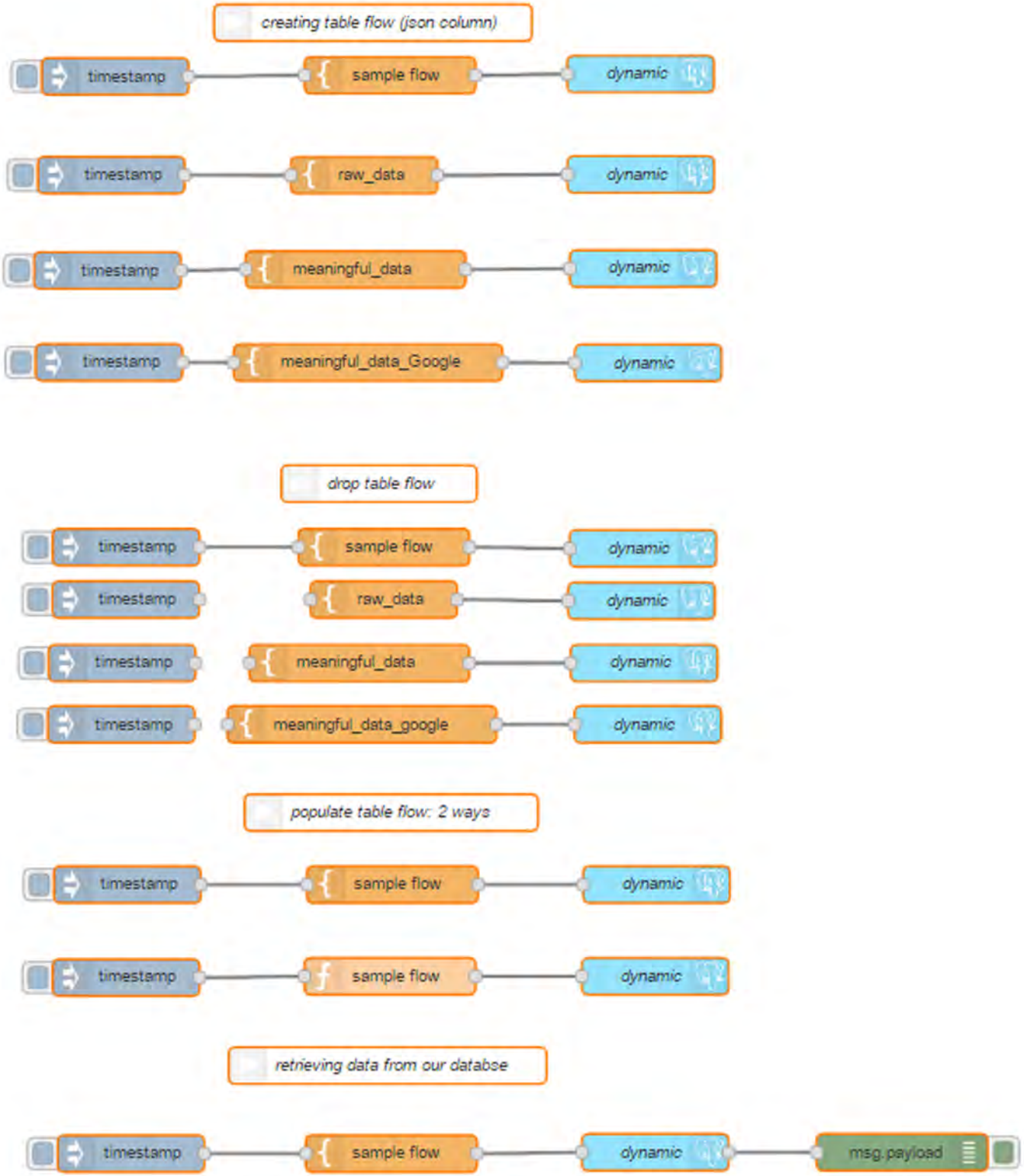
Appendix D – Manipulation of message received by *Google Maps Geolocation API*

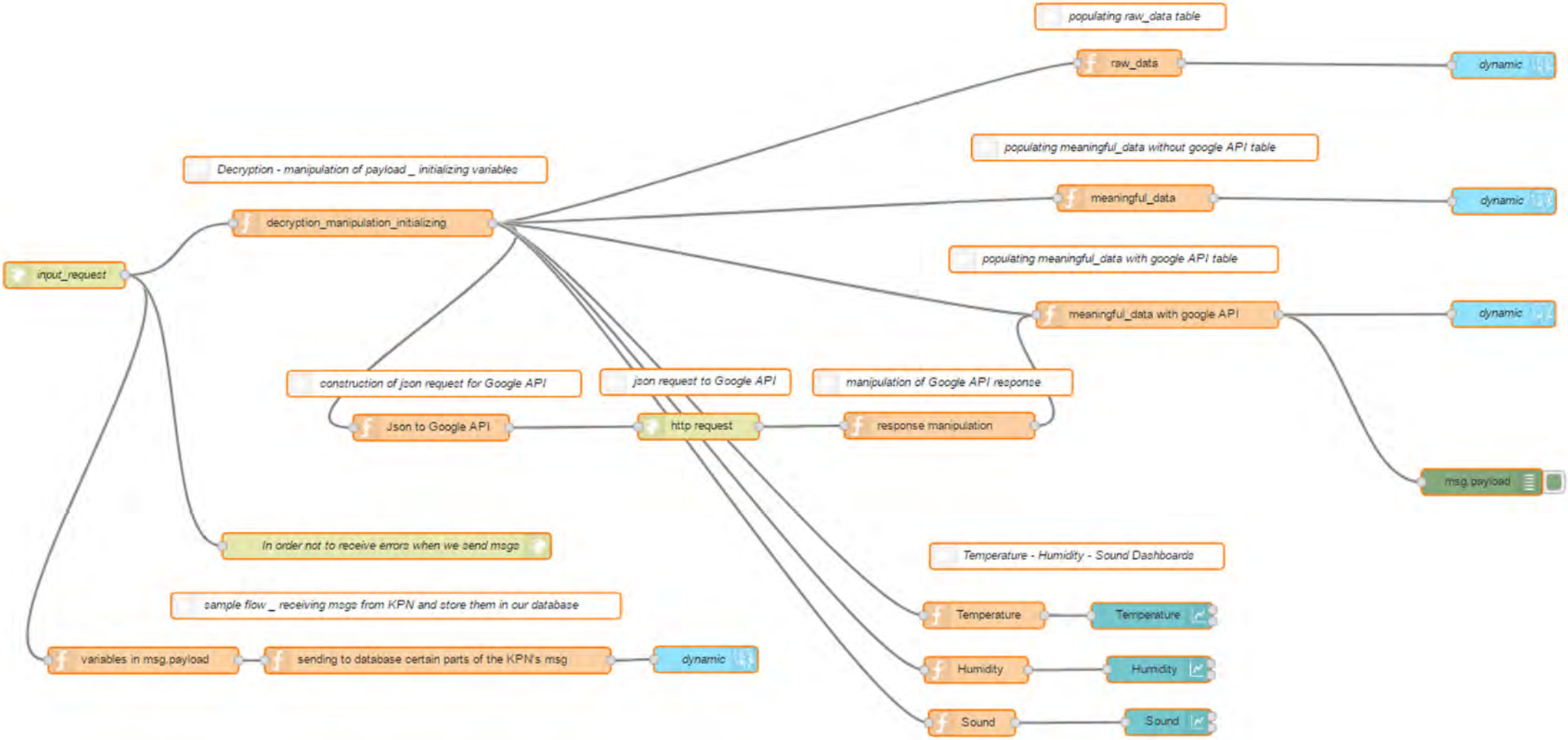
```
val = msg.payload;
err = val.includes("error");

var newPayload = [];


if (err === true) {
    newPayload.push({
        api_lat: 'error',
        api_lon: 'error',
        api_accuracy: 'error'
    });
}else{
    api_lat = val.substring(val.indexOf('52'), (val.indexOf(',\n  "lng":')));
    api_lon = val.substring(val.indexOf('4.'), (val.indexOf('\n },')));
    api_accuracy = val.substring(val.indexOf('"accuracy": ')+12, (val.indexOf('\n}')));

    newPayload.push({
        api_lat: api_lat,
        api_lon: api_lon,
        api_accuracy: api_accuracy
    });
}


msg.payload = newPayload;

msg.topic="google";
return msg;
```

Appendix E – Manipulation of messages received to create the final query in order to populate the meaningful_data_Google Table

```
context.data = context.data || {};

switch (msg.topic) {
    case "decryption":
        context.data.temp = msg.payload;
        msg = null;
        break;
    case "google":
        context.data.humidity = msg.payload;
        msg = null;
        break;

    default:
        msg = null;
        break;

}


if(context.data.temp !== null && context.data.humidity !== null ) {
    msg2 = {};
    msg2.payload = "Temp is:"+context.data.temp.mac2 +" Humidity is:"+ context.data.humidity;
    msg2.payload = "INSERT INTO meaningful_data_Google (DevAddr, Date, Time, Mac1, Strength1, Mac2, Strength2, Mac3, Strength3, Temperature, Humidity, Sound,  GPS_latitude, GPS_longitude, API_latitude, API_longitude, API_accuracy, GPS_Location, API_Location) VALUES ('" + context.data.temp.DevEUI_uplink.DevAddr + "'" + ", " + "'" + context.data.temp.ymd + "'" + "," + " '" +context.data.temp.hms + "'" + ", '" + context.data.temp.mac1 + "'" + ", " + context.data.temp.strength1 + ", " + "'" + context.data.temp.mac2 + "'" + ", " + context.data.temp.strength2 + ", '"+ context.data.temp.mac3 + "'" + ", " + context.data.temp.strength3 + ", " +  context.data.temp.temp + ", " + context.data.temp.hum + ", " + context.data.temp.sound + ", " + context.data.temp.lat  + ", " + context.data.temp.lon + ", " + context.data.humidity[0].api_lat + ", " +  context.data.humidity[0].api_lon + ", "  + context.data.humidity[0].api_accuracy + ", ST_GeomFromText('POINT(" + context.data.temp.lon + " " +context.data.temp.lat + ")',4326)" + ", ST_GeomFromText('POINT(" + context.data.humidity[0].api_lon + " " + context.data.humidity[0].api_lat + ")',4326));";

    context.data=null;
    return msg2;
}
```

Appendix F - Node-RED flow


creating table flow (json column)
timestamp
sample flow
dynamic
timestamp
raw_data
dynamic
timestamp
meaningful_data
dynamic
timestamp
meaningful_data_Google
dynamic
drop table flow
timestamp
sample flow
dynamic
timestamp
raw_data
dynamic
timestamp
meaningful_data
dynamic
timestamp
meaningful_data_google
dynamic
populate table flow: 2 ways
timestamp
sample flow
dynamic
timestamp
sample flow
dynamic
retrieving data from our database
timestamp
sample flow
dynamic
msg.payload


populating raw_data table
raw_data
dynamic
populating meaningful_data without google API table
Decryption - manipulation of payload _ initializing variables
meaningful_data
dynamic
decryption_manipulation_initializing
input_request
populating meaningful_data with google API table
meaningful_data with google API
dynamic
construction of json request for Google API
json request to Google API
manipulation of Google API response
Json to Google API
http request
response manipulation
msg.payload
In order not to receive errors when we send msgs
Temperature - Humidity - Sound Dashboards
sample flow _ receiving msgs from KPN and store them in our database
Temperature
Temperature
variables in msg.payload
sending to database certain parts of the KPN's msg
dynamic
Humidity
Humidity
Sound
Sound

Appendix G – Dashboards

Isolation of temperature variable

```
msg.payload = parseInt(msg.payload.temp);
return msg;
```

Isolation of humidity variable

```
msg.payload =  parseInt(msg.payload.hum);
return msg;
```

Isolation of humidity variable

```
msg.payload =  parseInt(msg.payload.sound);
return msg;
```

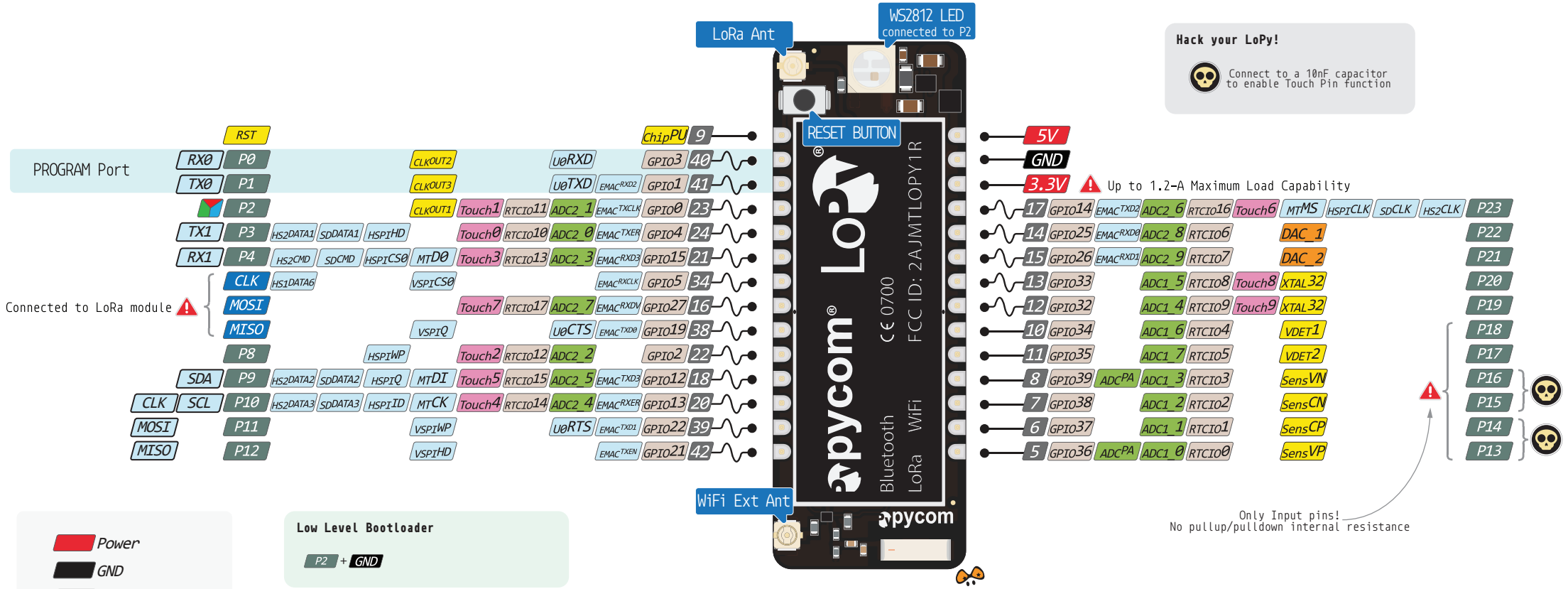
Appendix H - LoPy Pinout

LoPy® PINOUT

Internal Functions

35	GPIO18	VSPICLK	HS1DATA7	Lora Reset
25	GPIO16	EMACCLKOUT	U2RXD	HS1DATA4	External Antenna Switch
36	GPIO23	VSPIID	HS1STROBE	Lora Interrupt
27	GPIO17	EMACCO180	U2TXD	HS1DATA5	Lora Select

Absolute MAX per pin 12mA
recommended 6mA



Hack your LoPy!

Connect to a 10nF capacitor to enable Touch Pin function

- Power
- GND
- Serial Pin
- Analog Pin
- Control
- Physical Pin
- Port Pin
- Touch Pin
- DAC Pin
- PWM Pin

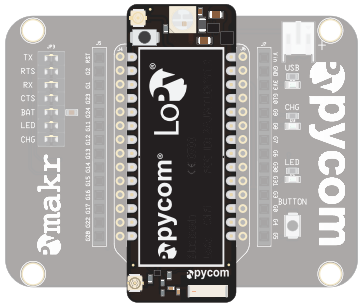
Low Level Bootloader

P2 + GND

Boot modes and safe boot

P12 + 3.3V

1-3sec	Safe boot, latest firmware is selected
4-6sec	Safe boot, previous user update selected
7-9sec	Safe boot, the factory firmware is selected



05 Nov 2016
ver 1 rev 5

Appendix I – LoPy Code main.py

```
import pycom
import time
from machine import Pin
from machine import Timer
from dth import DTH

import gps
import machine
from gps import GPS_UART_start
from gps import NmeaParser
from machine import RTC

from network import LoRa
from network import WLAN
import socket
import binascii
import ubinascii
import struct
import config
from math import log

pycom.heartbeat(False)
th = DTH(Pin('P10', mode=Pin.OPEN_DRAIN), 1)
adc = machine.ADC(bits=10)
apin = adc.channel(pin='P18')
SAMPLE_WINDOW = 50  # Sample window width in ms (50 ms = 20Hz)
sample = 0
gps = GPS_UART_start()
# LoRa details keys obtained from KPN
dev_addr = struct.unpack(">l", binascii.unhexlify(config.DEV_ADDR))[0]
# manually converted hex to decimal for better readability
#dev_addr = 337656918

nwks_key = binascii.unhexlify(config.NWKS_KEY)
apps_key = binascii.unhexlify(config.APPS_KEY)

# Setup LoRa
lora = LoRa(mode=LoRa.LORAWAN, adr=True)

# join a network using ABP
lora.join(activation=LoRa.ABP, auth=(dev_addr, nwks_key, apps_key), timeout=0)

# create a LoRa socket
lora_sock = socket.socket(socket.AF_LORA, socket.SOCK_RAW)

# set the LoRaWAN data rate
lora_sock.setsockopt(socket.SOL_LORA, socket.SO_DR, 5)


def lora_tx(payload):
    lora_sock.send(payload)

def send_blob():
    byms = []

    #WiFi RSSI
    wlanon = WLAN(mode=WLAN.STA, antenna=WLAN.EXT_ANT)
    scan = wlanon.scan()
    i = 0
    messages = []
    if len(scan) >=3:
        while i<3:
```

```
            messages.append(ubinascii.hexlify(scan[i].bssid, "/").decode())
            messages.append(scan[i].rssi)
            i += 1
    elif len(scan) == 2:
        while i<2:
            messages.append(ubinascii.hexlify(scan[i].bssid, "/").decode())
            messages.append(scan[i].rssi)
            i+= 1
        for num in range(7):
            message.append(00)
    else:
        for num in range(21):
            message.append(00)

    for j, part in enumerate(messages):
        if j%2 == 0:
            for byt in part.split('/'):
                byms.append(int(byt, 16))
        elif j%2 == 1:
            byms.append(abs(part))

    #temperature and humidity
    result = th.read()
    byms.append(int(result.temperature))
    byms.append(int(result.humidity))

    #sound
    chrono = Timer.Chrono()
    chrono.start()
    start_time = chrono.read_ms()

    peak_to_peak = 0  # peak-to-peak level
    signal_max = 0  # max signal
    signal_min = 1024  # min signal

    while (chrono.read_ms() - start_time) < SAMPLE_WINDOW:  # 50 ms
        sample = apin()  # read an analog value
               if sample > signal_max:
                   signal_max = sample
               elif sample < signal_min:
                   signal_min = sample

          peak_to_peak = signal_max - signal_min
          db = 0
          if peak_to_peak > 154:
              db = 20 * log((peak_to_peak)-(153.1) , 10)
          byms.append(int(db))

          #gps
          constant = 0
          while constant < 20:
              time.sleep(0.2)
              constant +=1
              if (gps.any()):
                  data = gps.readline()
                  if (data[0:6] == b'$GPGGA'):
                      place = NmeaParser()
                      place.update(data)
                      if (52.0000 < place.latitude < 52.02960797072103) or (4.35254
                          857123 < place.longitude < 4.38126560223):
                          print("fix")
                          constant = 20
                          const_lat = 51.9627
                          const_lon = 4.3161
                          lat = str(int(1000000*(place.latitude - const_lat)))
                          lon = str(int(1000000*(place.longitude - const_lon)))
```

```
                            byms.append(int(lat[0:2]))
                            byms.append(int(lat[2:4]))
                            byms.append(int(lat[4:]))
                            byms.append(int(lon[0:2]))
                            byms.append(int(lon[2:4]))
                            byms.append(int(lon[4:]))
            lora_tx(bytes(byms))

        send_blob()
        machine.deepsleep(180000)
```

Appendix J – Lopy Code gps.py library

```
1.  import os
2.  import time
3.  import struct
4.  import machine
5.  from machine import UART
6.
7.  #Minimalistic NMEA-0183 message parser, based on micropyGPS
8.  #Version 0.1 - January 2017
9.  #Autor: Peter Affolter
10.
11. import utime
12.
13. def GPS_UART_start():
14.     #print ('Start GPS UART1')
15.     com = UART(1,  pins=("P3",  "P4"),  baudrate=9600)
16.     # pins=("G23",  "G24")
17.     time.sleep(1)
18.     return(com)
19.
20.
21. def GPS_go(com):
22.     while (True):
23.         if (com.any()):
24.             data =com.readline()
25.             #print (data)
26.             if (data[0:6] == b'$GPGGA'):
27.                 place = NmeaParser()
28.                 place.update(data)
29.                 info1 = struct.pack('hii',  machine.rng()&0xffff,  int(place.longitu
    de*100000),  int(place.latitude*100000))
30.
31.
32.
33.
34. class NmeaParser(object):
35.     """NMEA Sentence Parser. Creates object that stores all relevant GPS data and st
    atistics.
36.     Parses sentences using update(). """
37.
38.     def __init__(self):
39.         """Setup GPS Object Status Flags, Internal Data Registers, etc"""
40.
41.         ######################
42.         # Data From Sentences
43.         # Time
44.         self.utc = (0)
45.
46.         # Object Status Flags
47.         self.fix_time = 0
48.         self.valid_sentence = False
49.
50.         # Position/Motion
51.         self.latitude = 0.0
52.         self.longitude = 0.0
53.         self.altitude = 0.0
54.
55.         # GPS Info
56.         self.satellites_in_use = 0
57.         self.hdop = 0.0
58.         self.fix_stat = 0
59.
60.         #raw data segments
```

```
        self.nmea_segments = []

    def update(self,  sentence):
        self.valid_sentence = False
        self.nmea_segments = str(sentence).split(',')

        #Parse GPGGA
        if (self.nmea_segments[0] == "b'$GPGGA") and len(self.nmea_segments) >= 12:

            self.valid_sentence = True
            try:
                # UTC Timestamp
                 utc_string = self.nmea_segments[1]

                # Skip timestamp if receiver doesn't have on yet
                 if utc_string:
                    hours = int(utc_string[0:2])
                    minutes = int(utc_string[2:4])
                    seconds = float(utc_string[4:])
                 else:
                    hours = 0
                    minutes = 0
                    seconds = 0.0

                 # Number of Satellites in Use
                 satellites_in_use = int(self.nmea_segments[7])

                 # Horizontal Dilution of Precision
                 hdop = float(self.nmea_segments[8])

                 # Get Fix Status
                 fix_stat = int(self.nmea_segments[6])
            except ValueError:
                return

         # Process Location and Speed Data if Fix is GOOD
            if fix_stat:
                # Longitude / Latitude
                try:
                    # Latitude
                            l_string = self.nmea_segments[2]
                            lat_degs = float(l_string[0:2])
                            lat_mins = float(l_string[2:])
                            lat_hemi = self.nmea_segments[3]
                            # Longitude
                            l_string = self.nmea_segments[4]
                            lon_degs = float(l_string[0:3])
                            lon_mins = float(l_string[3:])
                            lon_hemi = self.nmea_segments[5]
                        except ValueError:
                            return False

                        # Altitude / Height Above Geoid
                        try:
                            altitude = float(self.nmea_segments[9])
                            geoid_height = float(self.nmea_segments[11])
                        except ValueError:
                            return

                        # Update Object Data
                        self.latitude = lat_degs + (lat_mins/60)
                        if lat_hemi == 'S':
                            self.latitude = -self.latitude
                        self.longitude = lon_degs + (lon_mins/60)
                        if lon_hemi == 'W':
                            self.longitude = -self.longitude
```

```
                        self.altitude = altitude
                        self.geoid_height = geoid_height

                    # Update Object Data
                    self.timestamp = (hours, minutes, seconds)
                    self.satellites_in_use = satellites_in_use
                    self.hdop = hdop
                    self.fix_stat = fix_stat

                    # If Fix is GOOD, update fix timestamp
                    if fix_stat:
                        self.fix_time = utime.time()
                    return True


        com=GPS_UART_start()
```

Appendix K – Lopy Code dth.py library

```
import time
from machine import enable_irq, disable_irq,  Pin


class DTHResult:
    'DHT sensor result returned by DHT.read() method'

    ERR_NO_ERROR = 0
    ERR_MISSING_DATA = 1
    ERR_CRC = 2

    error_code = ERR_NO_ERROR
    temperature = -1
    humidity = -1

    def __init__(self, error_code, temperature, humidity):
        self.error_code = error_code
        self.temperature = temperature
        self.humidity = humidity

    def is_valid(self):
        return self.error_code == DTHResult.ERR_NO_ERROR


class DTH:
    'DHT sensor (dht11, dht21,dht22) reader class for Pycom'

    #__pin = Pin('P3', mode=Pin.OPEN_DRAIN)
    __dhttype = 0

    def __init__(self, pin, sensor=0):
        self.__pin = pin
        self.__dhttype = sensor
        self.__pin(1)
        time.sleep(1.0)

    def read(self):
        # time.sleep(1)

        # send initial high
        #self.__send_and_sleep(1, 0.025)

        # pull down to low
        self.__send_and_sleep(0, 0.019)

        # collect data into an array
        data = self.__collect_input()
        # print(data)
        # parse lengths of all data pull up periods
        pull_up_lengths = self.__parse_data_pull_up_lengths(data)
        # if bit count mismatch, return error (4 byte data + 1 byte checksum)
        # print(pull_up_lengths)
        # print(len(pull_up_lengths))
        if len(pull_up_lengths) != 40:
            return DTHResult(DTHResult.ERR_MISSING_DATA, 0, 0)

        # calculate bits from lengths of the pull up periods
        bits = self.__calculate_bits(pull_up_lengths)

        # we have the bits, calculate bytes
        the_bytes = self.__bits_to_bytes(bits)
        # print(the_bytes)
```

```
63.         # calculate checksum and check
64.         checksum = self.__calculate_checksum(the_bytes)
65.         if the_bytes[4] != checksum:
66.             return DTHResult(DTHResult.ERR_CRC, 0, 0)
67.
68.         # ok, we have valid data, return it
69.         [int_rh, dec_rh, int_t, dec_t, csum] = the_bytes
70.         if self.__dhttype == 0:  # dht11
71.             rh = int_rh  # dht11 20% ~ 90%
72.             t = int_t  # dht11 0..50°C
73.         else:  # dht21,dht22
74.             rh = ((int_rh * 256) + dec_rh) / 10
75.             t = (((int_t & 0x7F) * 256) + dec_t) / 10
76.             if (int_t & 0x80) > 0:
77.                 t *= -1
78.         return DTHResult(DTHResult.ERR_NO_ERROR, t, rh)
79.
80.     def __send_and_sleep(self, output, mysleep):
81.         self.__pin(output)
82.         time.sleep(mysleep)
83.
84.     def __collect_input(self):
85.         # collect the data while unchanged found
86.         unchanged_count = 0
87.         # this is used to determine where is the end of the data
88.         max_unchanged_count = 100
89.         last = -1
90.         data = []
91.         m = bytearray(800)        # needs long sample size to grab all the bits from
    the DHT
92.         irqf = disable_irq()
93.         self.__pin(1)
94.         for i in range(len(m)):
95.             m[i] = self.__pin()  # sample input and store value
96.         enable_irq(irqf)
97.         for i in range(len(m)):
98.             current = m[i]
99.             data.append(current)
100.                    if last != current:
101.                        unchanged_count = 0
102.                        last = current
103.                    else:
104.                        unchanged_count += 1
105.                        if unchanged_count > max_unchanged_count:
106.                            break
107.                # print(data)
108.                return data
109.
110.            def __parse_data_pull_up_lengths(self, data):
111.                STATE_INIT_PULL_DOWN = 1
112.                STATE_INIT_PULL_UP = 2
113.                STATE_DATA_FIRST_PULL_DOWN = 3
114.                STATE_DATA_PULL_UP = 4
115.                STATE_DATA_PULL_DOWN = 5
116.
117.                state = STATE_INIT_PULL_UP
118.
119.                lengths = []  # will contain the lengths of data pull up periods
120.                current_length = 0  # will contain the length of the previous period

121.
122.                for i in range(len(data)):
123.
124.                    current = data[i]
125.                    current_length += 1
126.
```

```
127.                     if state == STATE_INIT_PULL_DOWN:
128.                         if current == 0:
129.                             # ok, we got the initial pull down
130.                             state = STATE_INIT_PULL_UP
131.                             continue
132.                         else:
133.                             continue
134.                     if state == STATE_INIT_PULL_UP:
135.                         if current == 1:
136.                             # ok, we got the initial pull up
137.                             state = STATE_DATA_FIRST_PULL_DOWN
138.                             continue
139.                         else:
140.                             continue
141.                     if state == STATE_DATA_FIRST_PULL_DOWN:
142.                         if current == 0:
143.                             # we have the initial pull down, the next will be the dat
    a pull up
144.                             state = STATE_DATA_PULL_UP
145.                             continue
146.                         else:
147.                             continue
148.                     if state == STATE_DATA_PULL_UP:
149.                         if current == 1:
150.                             # data pulled up, the length of this pull up will determi
    ne whether it is 0 or 1
151.                             current_length = 0
152.                             state = STATE_DATA_PULL_DOWN
153.                             continue
154.                         else:
155.                             continue
156.                     if state == STATE_DATA_PULL_DOWN:
157.                         if current == 0:
158.                             # pulled down, we store the length of the previous pull u
    p period
159.                             lengths.append(current_length)
160.                             state = STATE_DATA_PULL_UP
161.                             continue
162.                         else:
163.                             continue
164.
165.                 return lengths
166.
167.             def __calculate_bits(self, pull_up_lengths):
168.                 # find shortest and longest period
169.                 shortest_pull_up = 1000
170.                 longest_pull_up = 0
171.
172.                 for i in range(0, len(pull_up_lengths)):
173.                     length = pull_up_lengths[i]
174.                     if length < shortest_pull_up:
175.                         shortest_pull_up = length
176.                     if length > longest_pull_up:
177.                         longest_pull_up = length
178.
179.                 # use the halfway to determine whether the period it is long or short

180.                 halfway = shortest_pull_up + (longest_pull_up -
     shortest_pull_up) / 2
181.                 bits = []
182.
183.                 for i in range(0, len(pull_up_lengths)):
184.                     bit = False
185.                     if pull_up_lengths[i] > halfway:
186.                         bit = True
187.                     bits.append(bit)
```

```
188.
189.                 return bits
190.
191.             def __bits_to_bytes(self, bits):
192.                 the_bytes = []
193.                 byte = 0
194.
195.                 for i in range(0, len(bits)):
196.                     byte = byte << 1
197.                     if (bits[i]):
198.                         byte = byte | 1
199.                     else:
200.                         byte = byte | 0
201.                     if ((i + 1) % 8 == 0):
202.                         the_bytes.append(byte)
203.                         byte = 0
204.                 # print(the_bytes)
205.                 return the_bytes
206.
207.             def __calculate_checksum(self, the_bytes):
208.                 return the_bytes[0] + the_bytes[1] + the_bytes[2] + the_bytes[3] &255
```

Appendix L- Decryption

```
var lora_decrypt = global.get('loraModule').lora_decrypt;
var lat = 0;
var lon = 0;
payload_hex = msg.payload.DevEUI_uplink.payload_hex;
sequence_counter = msg.payload.DevEUI_uplink.FCntUp;
addr = msg.payload.DevEUI_uplink.DevAddr;

if ( msg.payload.DevEUI_uplink.DevAddr == "14203E56") {
    key = "cd1fffe4f8a8a970feb40736f3d4a8fa";
//na fugei otan tha mpoun ta alla keys
} else if (msg.payload.DevEUI_uplink.DevAddr == "1420410E") {
    key = 'bd5e5d5c39ff4b5765a03da33efb156d';
} else if (msg.payload.DevEUI_uplink.DevAddr == "14204AEE") {
    key = 'd30ce42293854a7bbfbf0375039847b8';
} else if (msg.payload.DevEUI_uplink.DevAddr == "14203742") {
    key = '14bb29305f3a36c92458e26349bebd7d';
} else if (msg.payload.DevEUI_uplink.DevAddr == "142042F9") {
    key = '317cb07d02afdbe87d4251694f108fbf';
} else if (msg.payload.DevEUI_uplink.DevAddr == "1420447E") {
    key = '1c9079fa4528314ae302367f54544c40';
} else if (msg.payload.DevEUI_uplink.DevAddr == "14204AEE") {
    key = 'd30ce42293854a7bbfbf0375039847b8';
} else if (msg.payload.DevEUI_uplink.DevAddr == "142042DA") {
    key = 'd2e4a6ae867b3ff96a3eead12142211d';
} else if (msg.payload.DevEUI_uplink.DevAddr == "142031B0") {
    key = '05380e7d90983e795fa51f3fb6651317';
}
//} else {
//    key = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx';
//}

msg.payload.new = lora_decrypt(payload_hex, sequence_counter, key, addr);
```

Appendix M- Manipulation of decrypted payload_hex and initialisation of variables

```
msg.payload.final = "";
for (var x in msg.payload.new){
    if (x == 6){
         msg.payload.final += msg.payload.new[x].toString();
    }else if (x == 13){
         msg.payload.final += msg.payload.new[x].toString();
    }else if (x > 19){
         msg.payload.final += msg.payload.new[x].toString();
    }else{
         msg.payload.final += msg.payload.new[x].toString(16);
    }
}

msg.payload.ymd = msg.payload.DevEUI_uplink.Time.substr(0, 10);
msg.payload.hms = msg.payload.DevEUI_uplink.Time.substr(11, 8);

msg.payload.mac1 = "";
msg.payload.mac2 = "";
msg.payload.mac3 = "";
msg.payload.strength1 = "";
msg.payload.strength2 = "";
msg.payload.strength3 = "";
msg.payload.temp = "";
msg.payload.hum = "";
msg.payload.sound = "";
msg.payload.lat = "51.9627";
msg.payload.lon = "4.3161";

for (var x in msg.payload.new){
    if (x < 5){
        if (msg.payload.new[x].toString(16).length < 2){
            msg.payload.mac1 += '0' + msg.payload.new[x].toString(16) + ':';
        }else
            msg.payload.mac1 += msg.payload.new[x].toString(16) + ':';
    }else if (x == 5){
        if (msg.payload.new[x].toString(16).length < 2){
            msg.payload.mac1 += '0' + msg.payload.new[x].toString(16);
        }else
            msg.payload.mac1 += msg.payload.new[x].toString(16);
    }else if (x == 6){
         msg.payload.strength1 += msg.payload.new[x].toString();
    }else if (x > 6 && x < 12){
        if (msg.payload.new[x].toString(16).length < 2){
            msg.payload.mac2 += '0' + msg.payload.new[x].toString(16) + ':';
        }else
            msg.payload.mac2 += msg.payload.new[x].toString(16) + ':';
    }else if (x == 12){
        if (msg.payload.new[x].toString(16).length < 2){
            msg.payload.mac2 += '0' + msg.payload.new[x].toString(16);
        }else
            msg.payload.mac2 += msg.payload.new[x].toString(16);
    }else if (x == 13){
         msg.payload.strength2 += msg.payload.new[x].toString();
    }else if (x > 13 && x < 19){
```

```
        if (msg.payload.new[x].toString(16).length < 2){
            msg.payload.mac3 += '0' + msg.payload.new[x].toString(16) + ':';
        }else
            msg.payload.mac3 += msg.payload.new[x].toString(16) + ':';
    }else if (x == 19){
        if (msg.payload.new[x].toString(16).length < 2){
            msg.payload.mac3 += '0' + msg.payload.new[x].toString(16);
        }else
            msg.payload.mac3 += msg.payload.new[x].toString(16);
    }else if (x == 20){
         msg.payload.strength3 += msg.payload.new[x].toString();
    }else if (x == 21){
         msg.payload.temp += msg.payload.new[x].toString();
    }else if (x == 22){
         msg.payload.hum += msg.payload.new[x].toString();
    }else if (x == 23){
         msg.payload.sound += msg.payload.new[x].toString();

    }else if (x > 23 && x < 27){
         lat += msg.payload.new[x].toString();
    }else if (x > 26){
         lon += msg.payload.new[x].toString();
    }
}

msg.payload.lat = (parseInt(lat)/1000000.0) + 51.9627;
msg.payload.lon = (parseInt(lon)/1000000.0) + 4.3161;
msg.payload.gps = msg.payload.lat.toString() + " " + msg.payload.lon.toString();


msg.topic="decryption";
return msg;
```

Appendix N: Create Radiomap

```
1.  import math
2.  import csv
3.  from pyproj import *
4.  import psycopg2
5.  import datetime
6.
7.  #p1 is the CRS of the original coordinates, p2 the intended CRS
8.  p1 = Proj(init='epsg:4326')
9.  p2 = Proj(init='epsg:28992')
10.
11. mac1 = []
12. mac2 = []
13. mac3 = []
14. rss1 = []
15. rss2 = []
16. rss3 = []
17. lat = []
18. lon = []
19.
20. #import directly from the online database
21. conn = psycopg2.connect("dbname=dynamic user = dynamic password = ******* host= 131.
    180.126.34 port = 5432")
22. cur = conn.cursor()
23. cur.execute("SELECT * FROM meaningful_data_google;")
24. for record in cur:
25.     #uncomment the following line and indent the rest to create a radiomap until a c
    ertain date/time
26.     #if record[1] <= datetime.date(2017, 6, 12):
27.     mac1.append(record[3])
28.     rss1.append(-int(record[4]))
29.     mac2.append(record[5])
30.     rss2.append(-int(record[6]))
31.     mac3.append(record[7])
32.     rss3.append(-int(record[8]))
33.     lat.append(float(record[12]))
34.     lon.append(float(record[13]))
35.
36. #variables of the grid
37. minx = 83950.
38. maxx = 85950.
39. miny = 446250.
40. maxy = 448250.
41. size = 50
42.
43. #make step size of grid
44. xstep = int((maxx-minx)/size)
45. ystep = int((maxy-miny)/size)
46.
47. radiomap = {}
48. nrmeasurements = {} #for the heatmap
49.
50. def logavg(values):
51.     total = 0
52.     for val in values:
53.         total += (10**(val/10.0))
54.     return 10*math.log10(total/len(values))
55.
56. for i in range(len(mac1)):
57.     x2, y2 = transform(p1, p2, lon[i], lat[i])
58.     if (miny < y2 < maxy) and (minx < x2 < maxx):            #this statement checks
     in which grid cell the gps coordinate falls
59.         gridx = int((x2 - minx)/size)
```

```
        gridy = ystep - int(math.ceil((y2 - miny)/size))
        gridkey = gridy*ystep+gridx
        if gridkey in radiomap:
            nrmeasurements[gridkey] += 1
            macs = radiomap[gridkey][0]
            rssi = radiomap[gridkey][1]
            if mac1[i] in macs:
                entry = macs.index(mac1[i])
                rssi[entry].append(rss1[i])
            else:
                radiomap[gridkey][0].append(mac1[i])
                radiomap[gridkey][1].append([rss1[i]])
            if mac2[i] in macs:
                entry = macs.index(mac2[i])
                rssi[entry].append(rss2[i])
            else:
                radiomap[gridkey][0].append(mac2[i])
                radiomap[gridkey][1].append([rss2[i]])
            if mac3[i] in macs:
                entry = macs.index(mac3[i])
                rssi[entry].append(rss3[i])
            else:
                radiomap[gridkey][0].append(mac3[i])
                radiomap[gridkey][1].append([rss3[i]])
        else:
            nrmeasurements[gridkey] = 1
            radiomap[gridkey] = [[mac1[i]],[[rss1[i]]]]
            radiomap[gridkey][0].extend((mac2[i], mac3[i]))
            radiomap[gridkey][1].extend(([rss2[i]], [rss3[i]]))

#average the decibels per mac address in each gridcell
for key,value in radiomap.iteritems():
    macs = value[0]
    rssi = value[1]
    for i in range(len(macs)):
        rssi[i] = logavg(rssi[i])

#write radiomap to csv file
with open('radiomap.csv', 'w') as fh:
    fh.write('grid cell; mac address; rssi' + '\n')
            for key, value in radiomap.iteritems():
                fh.write(str(key) + ';')
                for i in range(len(value[0])):
                    fh.write(str(value[0][i]) + ';' + str(value[1][i]) + ';')
                fh.write('\n')

       #write heatmap to csv file
       with open('heatmap.csv', 'w') as gh:
           gh.write('cell; nr of measurements' + '\n')
           for key, value in nrmeasurements.iteritems():
               gh.write(str(key) + '; ' + str(value))
               gh.write('\n')

       #close online database connection
       cur.close()
       conn.close()
```

Appendix O: Localisation via Radiomap and comparison GPS/Google API

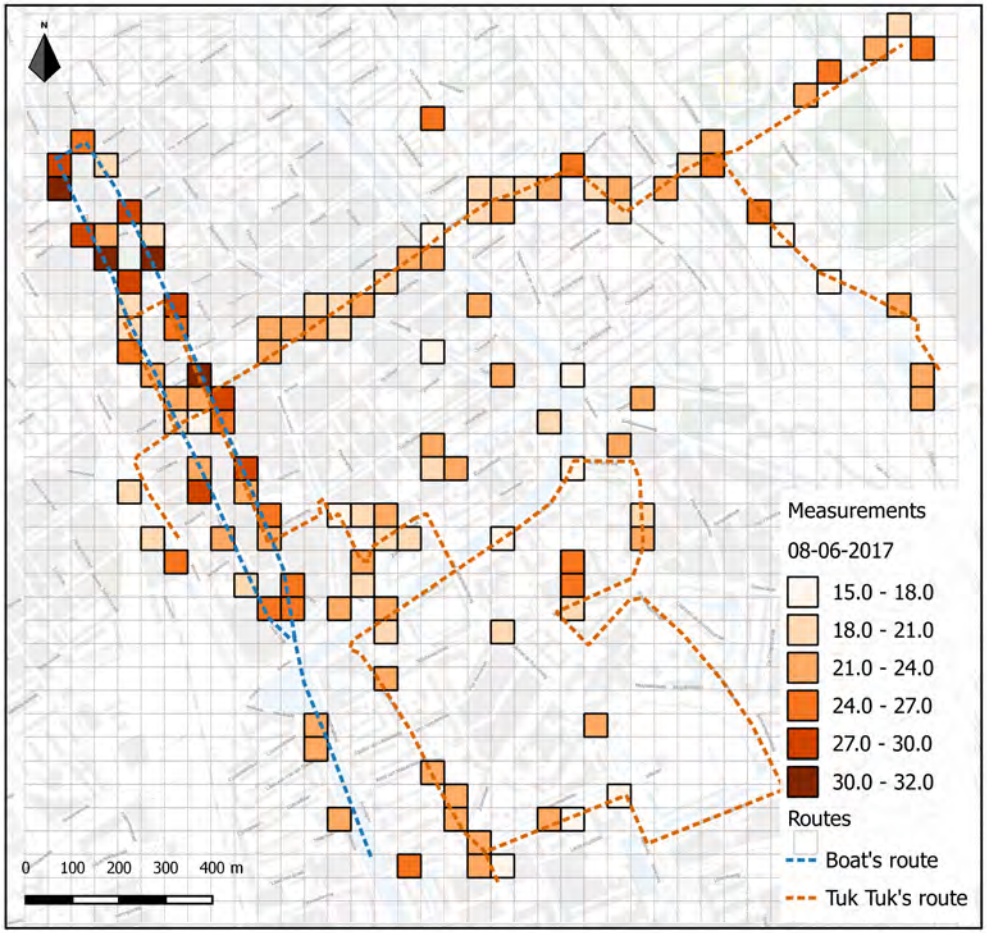
```
1.  import math
2.  import csv
3.  from pyproj import *
4.  import psycopg2
5.  import datetime
6.
7.  p1 = Proj(init='epsg:4326')
8.  p2 = Proj(init='epsg:28992')
9.  #variables of the grid
10. minx = 83950.
11. maxx = 85950.
12. miny = 446250.
13. maxy = 448250.
14. size = 50
15.
16. #make step size of grid
17. xstep = int((maxx-minx)/size)
18. ystep = int((maxy-miny)/size)
19.
20. mac1 = []
21. mac2 = []
22. mac3 = []
23. rss1 = []
24. rss2 = []
25. rss3 = []
26. lat = []
27. lon = []
28. latgo = []
29. longo = []
30. correct = 0
31. wrong = 0
32. undefined = 0
33. almost = 0
34. nomac = 0
35.
36. #connect to database
37. conn = psycopg2.connect("dbname=dynamic user = dynamic password = ******* host= 131.
    180.126.34 port = 5432")
38. cur = conn.cursor()
39. cur.execute("SELECT * FROM meaningful_data_google;")
40. for record in cur:
41.     if record[1] > datetime.date(2017, 6, 12):
42.         mac1.append(record[3])
43.         rss1.append(-int(record[4]))
44.         mac2.append(record[5])
45.         rss2.append(-int(record[6]))
46.         mac3.append(record[7])
47.         rss3.append(-int(record[8]))
48.         lat.append(float(record[12]))
49.         lon.append(float(record[13]))
50.         latgo.append(float(record[14])) #lat google API
51.         longo.append(float(record[15])) #lon google API
52. entrydict = {}
53. wronglist = []
54.
55. for i in range(len(mac1)):
56.     entrydict[i] = [mac1[i],rss1[i],mac2[i],rss2[i], mac3[i],rss3[i], lat[i],lon[i],
    latgo[i],longo[i]]
57.
58. filename2 = 'radiomap.csv'
```

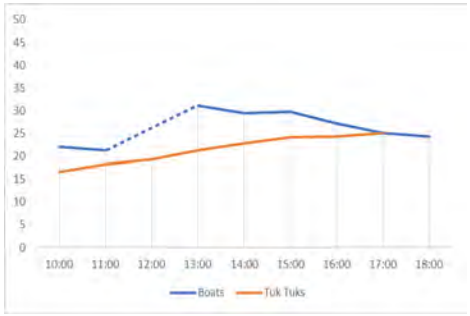
```
radiomap = {}
#make dictionary out of radiomap.csv
with open(filename2,'r') as g:
    for line in csv.reader(g, delimiter = ';'):
        if line[0] != 'grid cell':
            cellid = line[0]
            radiomap[cellid] = [[],[]]
            #print line[1:]
            for i in range(len(line[1:-
1])):                                   #len[1:-
1] because last list index is an empty value, change this if not
                if i % 2 == 0:
                    radiomap[cellid][0].append(line[1+i])
                else:
                    radiomap[cellid][1].append(line[1+i])

def euclidian(rsscell,rssitem):
    if rsscell == rssitem:
        return 0
    else:
        return (abs(10**(rsscell/10.0) - 10**(rssitem/10.0)))**2

def lookup(macs,rssi):
    cellist3 = []       #define a different list of canditate cells where 3 MACs mat
ch and 2 MACs match
    cellist2 = []
    quality3 = []
    quality2 = []
    for cell, item in radiomap.iteritems():
        d = []
        count = 0
        for i in range(len(macs)):
            if macs[i] in item[0]:
                entry = item[0].index(macs[i])
                radiorss = item[1][entry]
                d.append(euclidian(float(radiorss),float(rssi[i])))
                count += 1
        if count == 3:
            cellist3.append(cell)
            totd = math.sqrt(sum(d))
            quality3.append(totd)
        elif count == 2:
            cellist2.append(cell)
            totd = math.sqrt(sum(d))
                   quality2.append(totd)
           if len(cellist3) > 0:
               zipped = zip(quality3,cellist3)
               zipped.sort()
               return zipped[0][1]     #returns the cell that matches the best (lowe
st quality parameter = the best)
           elif len(cellist2) > 0:
               zipped = zip(quality2,cellist2)
               zipped.sort()
               return zipped[0][1]
           else:
               return -1               #no cell is matched


      def control(gridcell,latitude,longitude):
           x2, y2 = transform(p1, p2, longitude, latitude)
           if (miny < y2 < maxy) and (minx < x2 < maxx):             #this statement
 checks in which grid cell the gps coordinate falls
               gridx = int((x2 - minx)/size)
               gridy = ystep - int(math.ceil((y2 - miny)/size))
               gridkey = gridy*ystep+gridx
           else:
```

```
120.                 global undefined        #GPS measurement not in range of grid
121.                 undefined += 1
122.                 return 0
123.             if str(gridkey) == gridcell: #correctly classified
124.                 global correct
125.                 correct +=1
126.                 return 1
127.             elif gridcell == -1: #no grid cell has matched in the lookup function
128.                 global nomac
129.                 nomac += 1
130.             else:
131.                 global wrong
132.                 global almost
133.                 wronglist.append([gridkey,gridcell])
134.                 item = [gridkey,gridcell]
135.                 diff = abs(int(item[0])-int(item[1]))
136.                 if diff in [1,39,40,41]:#Change this when the amount of steps changes
137.                     almost += 1
138.                 else:
139.                     wrong += 1
140.                 return 1
141.
142.         for key,value in entrydict.iteritems():
143.             macs = [value[0],value[2],value[4]]
144.             rssi = [value[1],value[3],value[5]]
145.             cell = lookup(macs,rssi)
146.             #comment first controlfunction and uncomment second one for obtaining Goo
    gle API stats
147.             control(cell,value[6],value[7])
148.             #control(cell,value[8],value[0])
149.
150.         cur.close()
151.         conn.close()
152.         print 'correct', correct
153.         print 'wrong', wrong
154.         print 'In range of 1 cell', almost
155.         print 'out of range/no information', undefined
156.         print 'no mac was matched', nomac
```

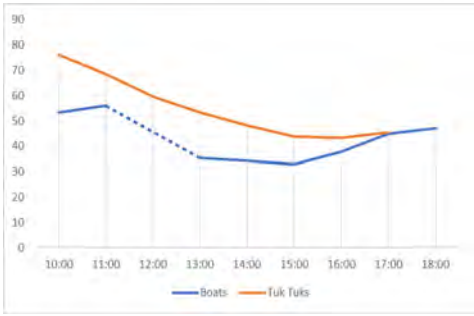
Appendix P - Visualisation 08-09 June

Measurements on 08-06-2017



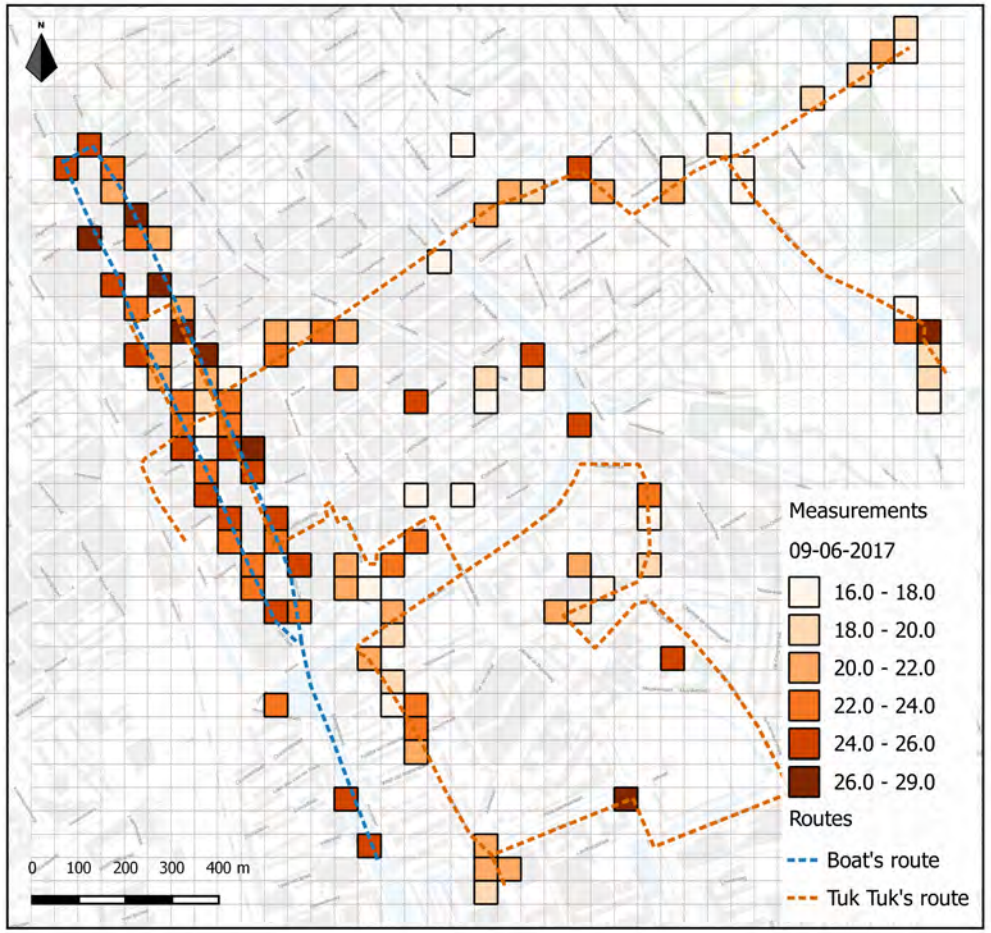


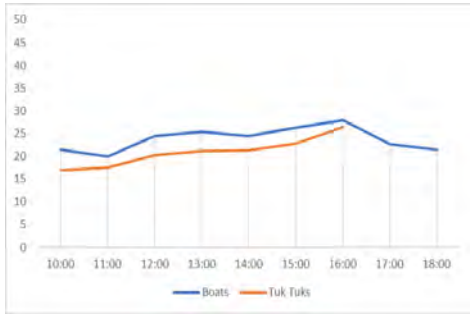
Temperature



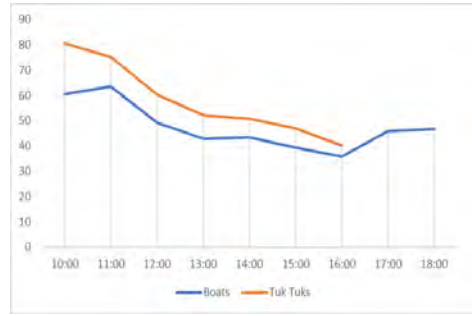
Humidity

Measurements on 09-06-2017





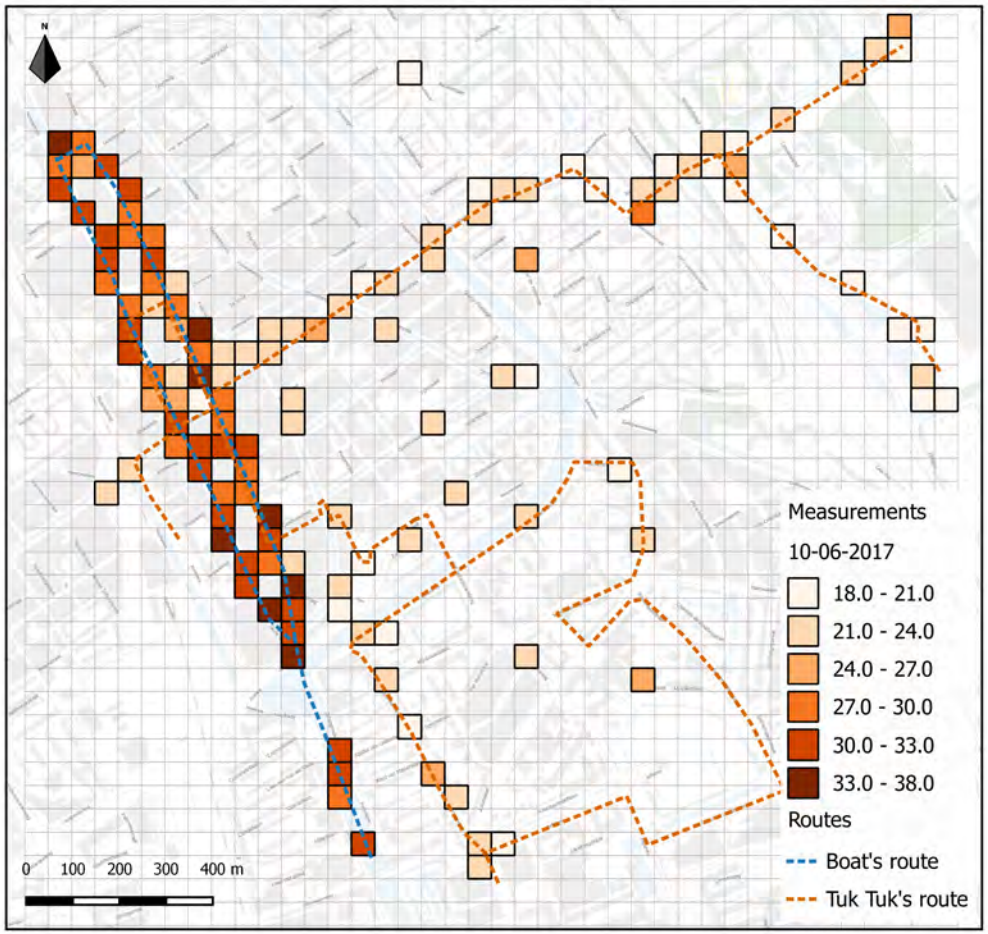
Temperature

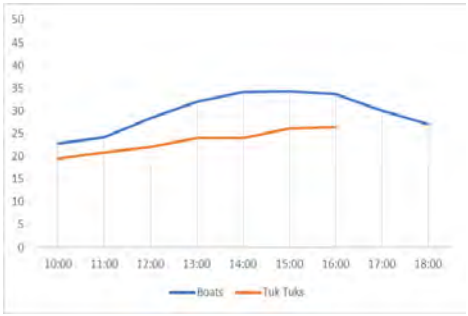


Humidity

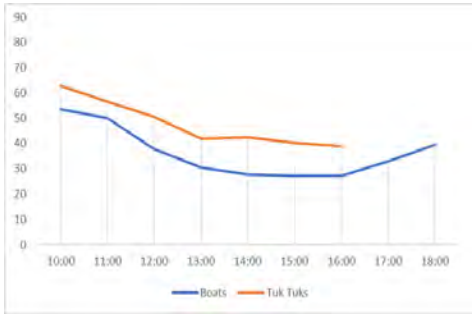
Appendix Q - Visualisation 10-11 June

Measurements on 10-06-2017



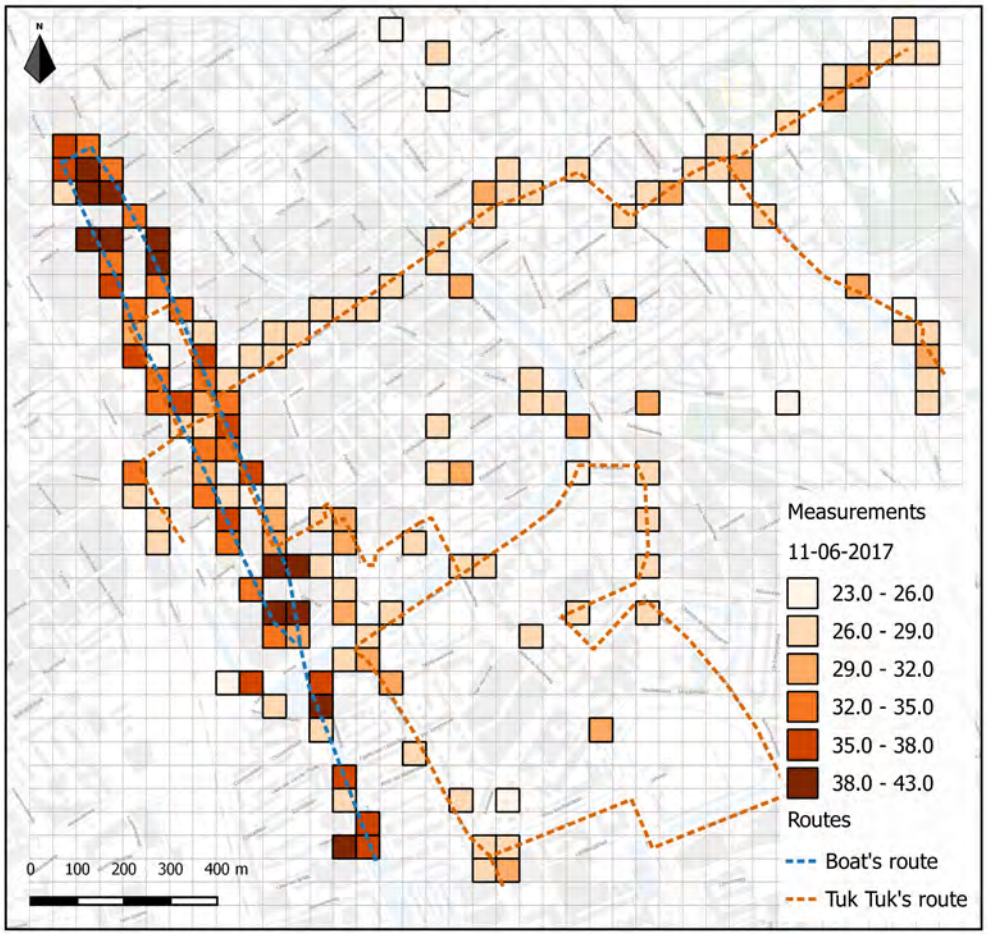


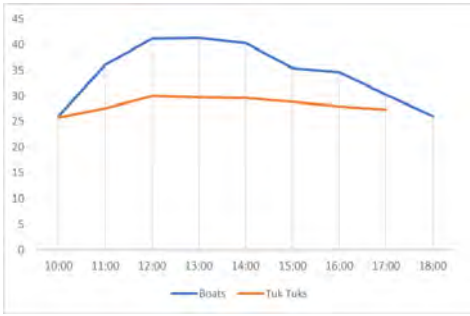
Temperature



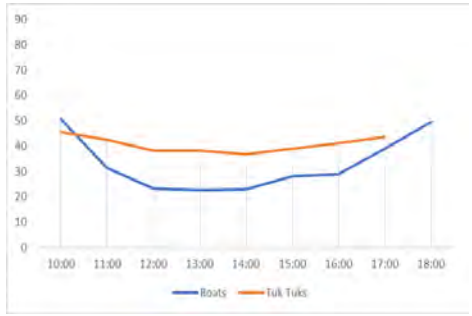
Humidity

Measurements on 11-06-2017





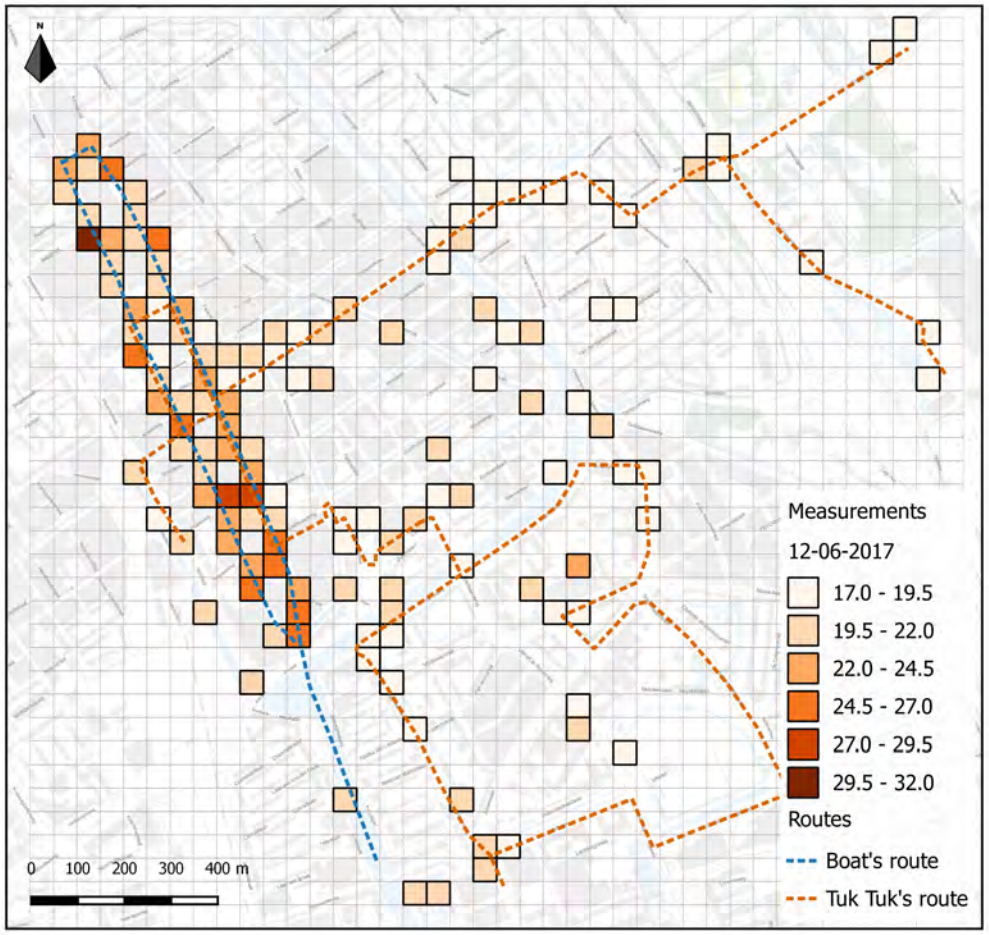
Temperature

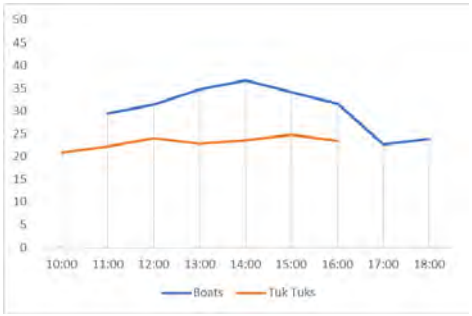


Humidity

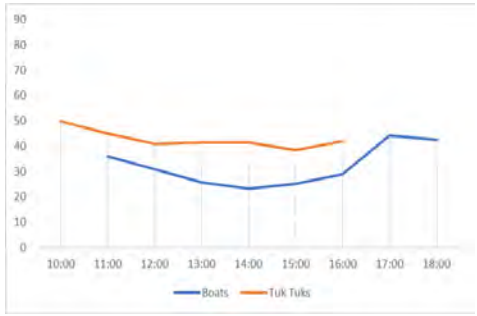
Appendix R - Visualisation 12-13 June

Measurements on 12-06-2017



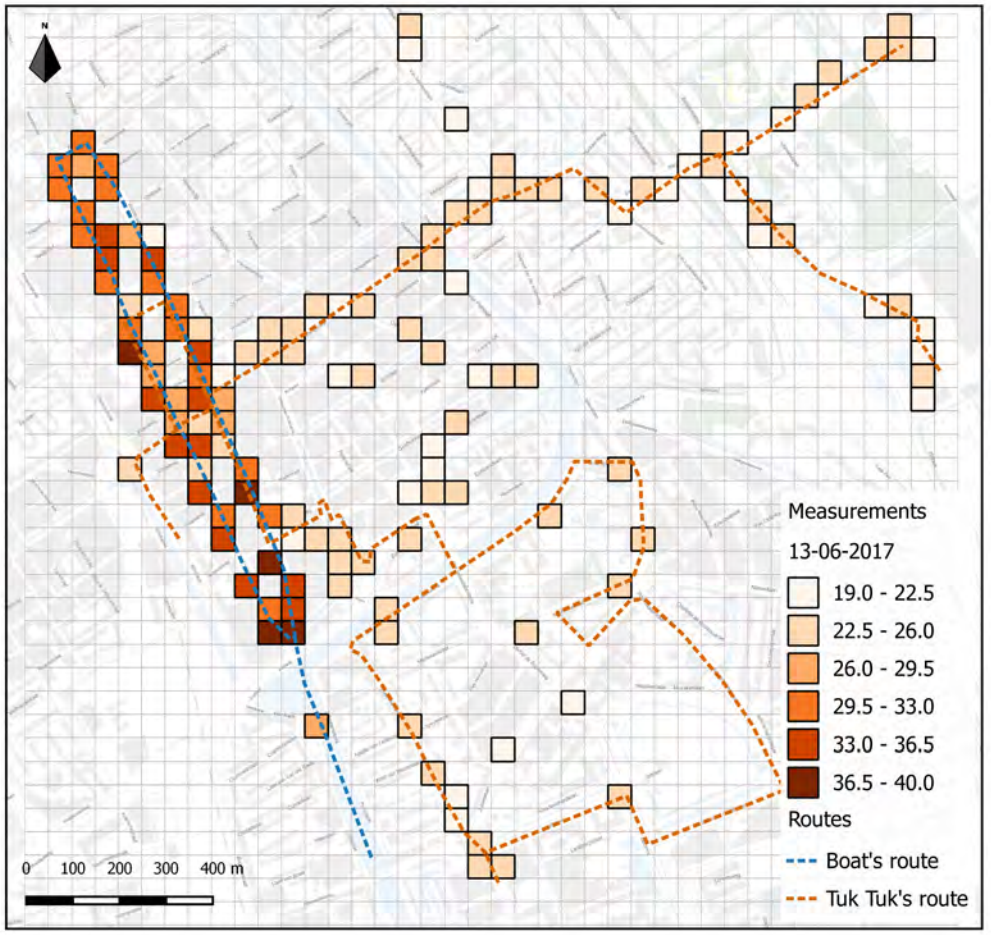


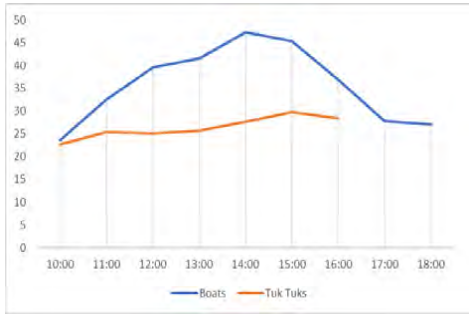
Temperature



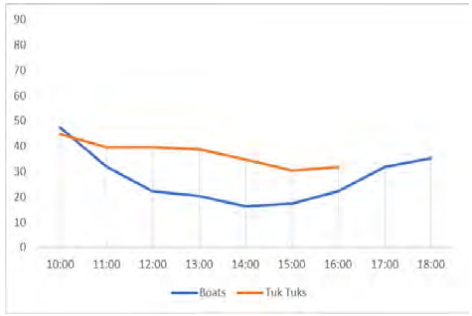
Humidity

Measurements on 13-06-2017





Temperature



Humidity

Appendix S - Visualisation whole week

Average temperature measurments per grid cell in Delft
Summary of the week

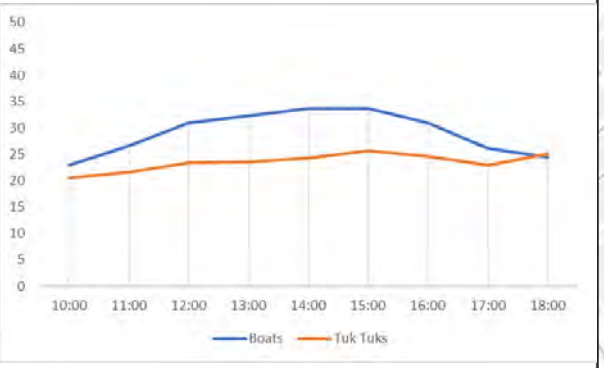
Whole week

- 17.0 - 21.0
- 21.0 - 25.0
- 25.0 - 29.0
- 29.0 - 33.0
- 33.0 - 37.0
- 37.0 - 42.0

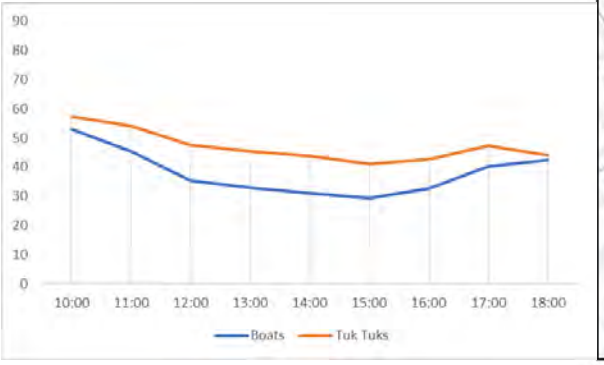
Routes

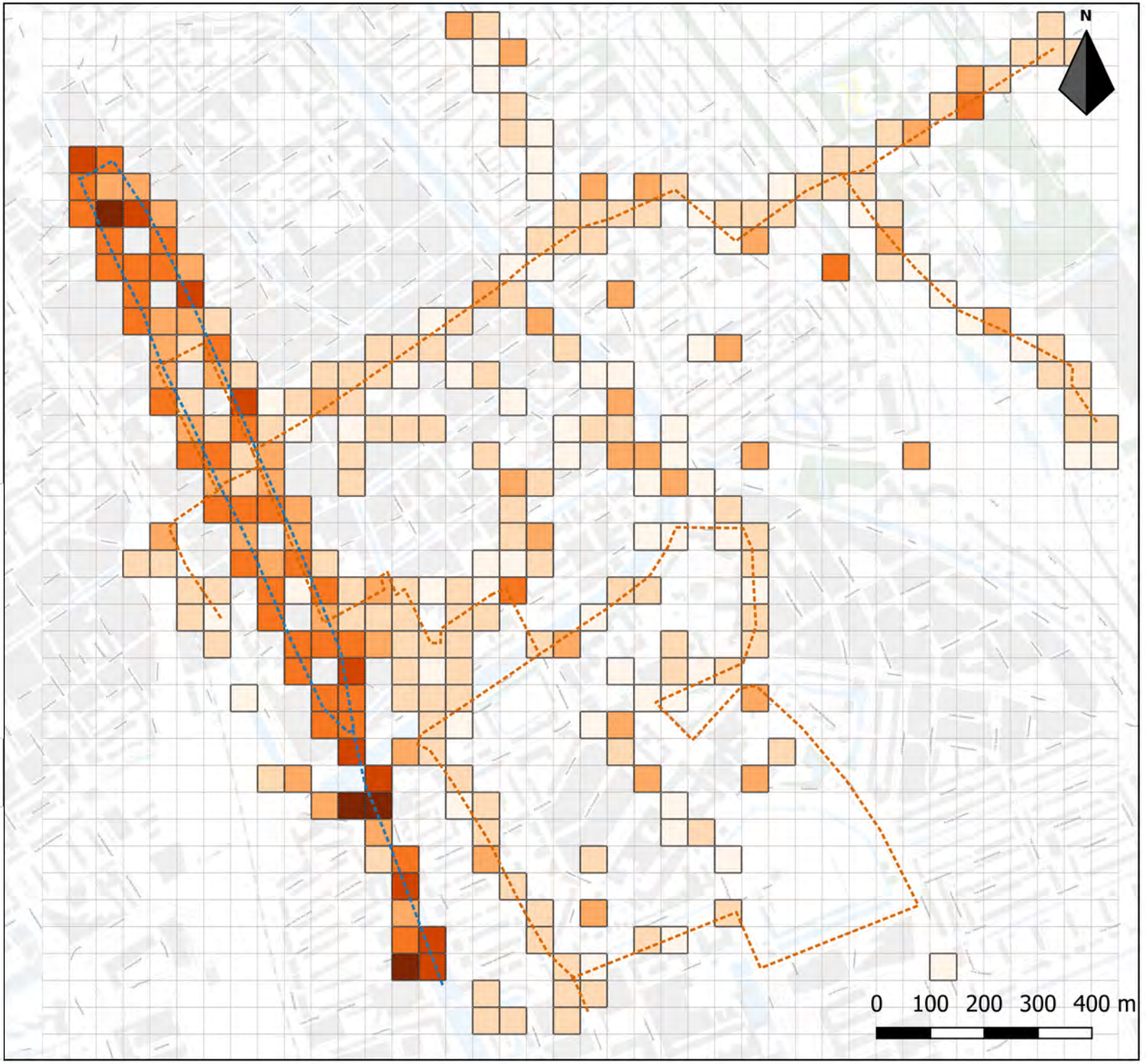
- Boat's route
- Tuk Tuk's route

Temperatures



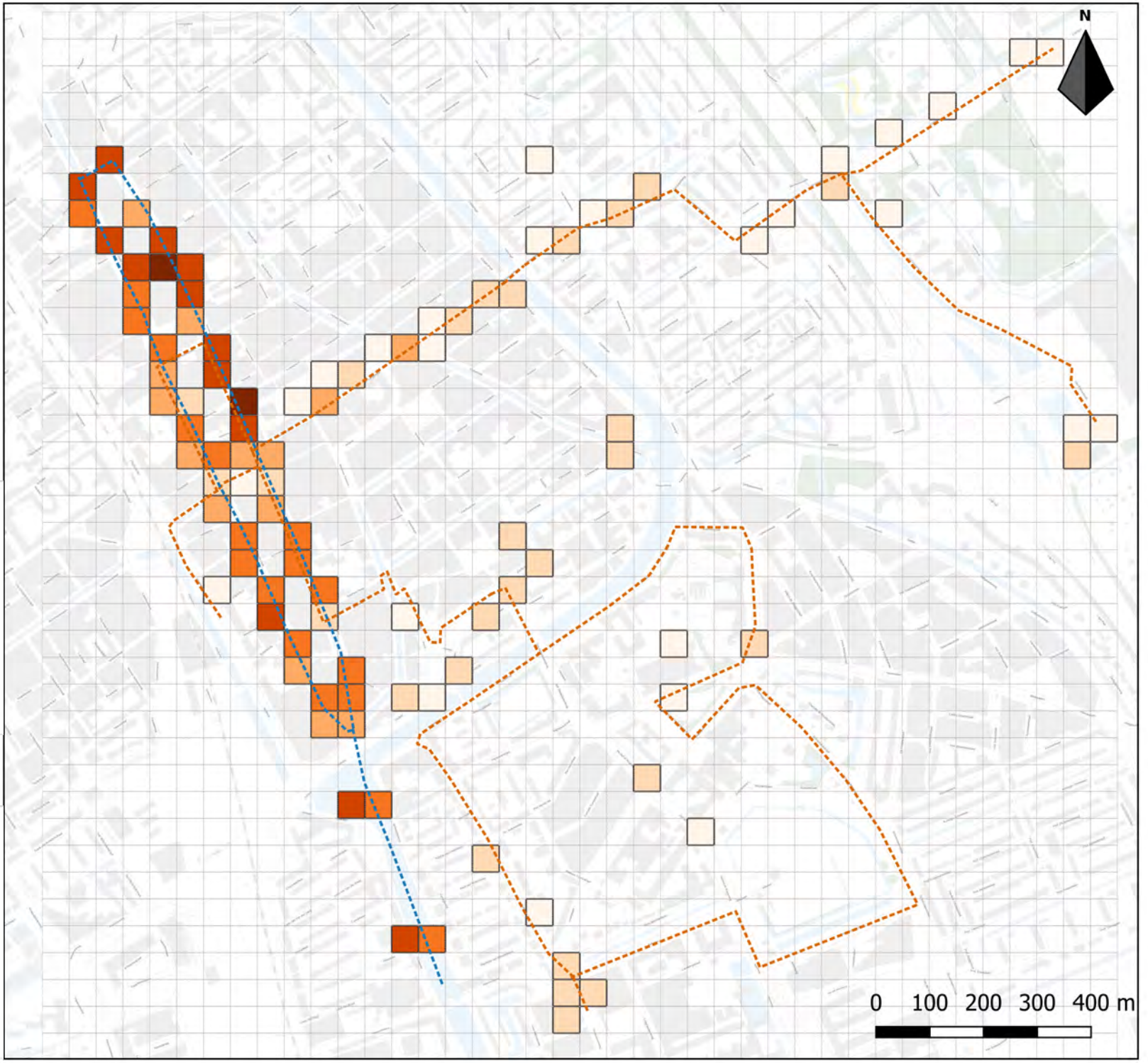
Humidity



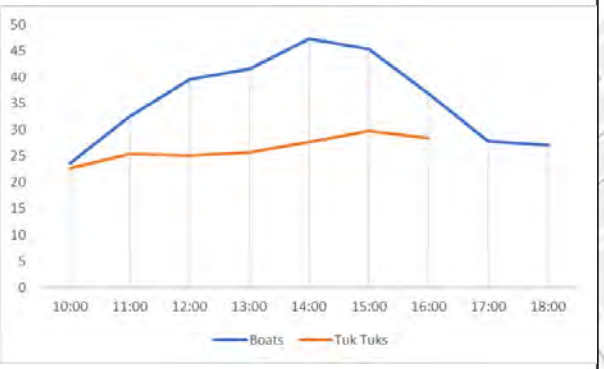


Appendix T - Visualisation grid 14 June

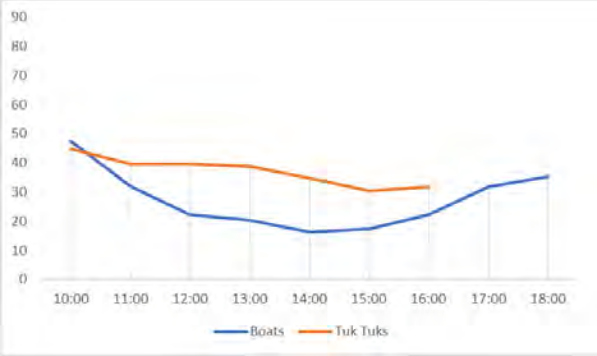
Average temperature measurments per grid cell in Delft 14-06-2017



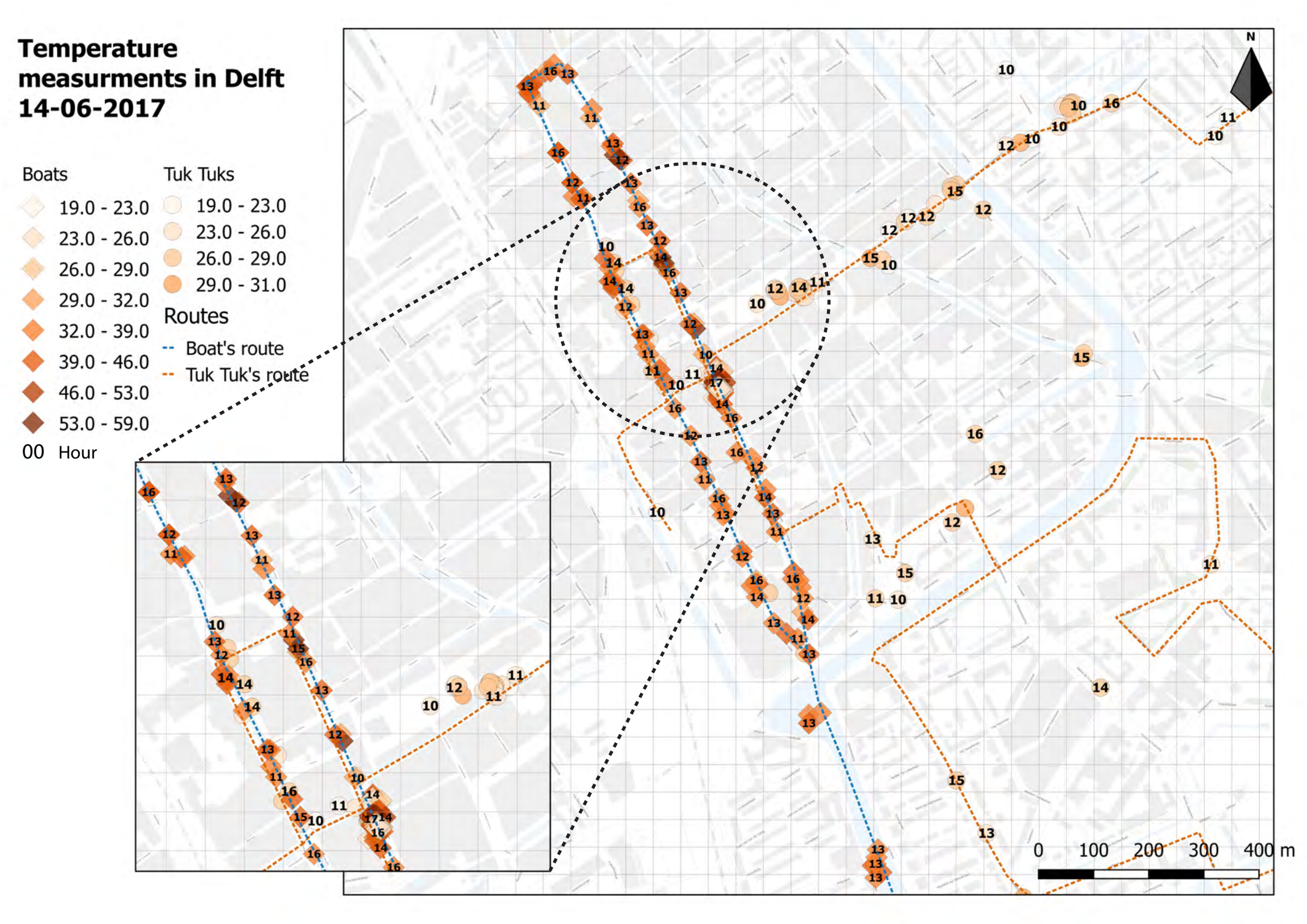
Temperatures



Humidity



Appendix U - Visualisation measurements 14 June


Temperature measurments in Delft 14-06-2017
Boats
19.0 - 23.0
23.0 - 26.0
26.0 - 29.0
29.0 - 32.0
32.0 - 39.0
39.0 - 46.0
46.0 - 53.0
53.0 - 59.0
00 Hour
Tuk Tuks
19.0 - 23.0
23.0 - 26.0
26.0 - 29.0
29.0 - 31.0
Routes
Boat's route
Tuk Tuk's route
N
0 100 200 300 400 m

Appendix V: Comparison Google API vs GPS

```
1.  import math
2.  from pyproj import *
3.  import psycopg2
4.  import datetime
5.
6.  p1 = Proj(init='epsg:4326')
7.  p2 = Proj(init='epsg:28992')
8.
9.  lat = []
10. lon = []
11. latgo = []
12. longo = []
13. qual = []
14.
15. minx = 83950.
16. maxx = 85950.
17. miny = 446250.
18. maxy = 448250.
19.
20. conn = psycopg2.connect("dbname=dynamic user = dynamic password = ******* host= 131.
    180.126.34 port = 5432")
21. cur = conn.cursor()
22. cur.execute("SELECT * FROM meaningful_data_google;")
23. print cur.fetchone()
24.
25. def distance(x1,x2,y1,y2):
26.     return math.sqrt((x1-x2)**2+(y1-y2)**2)
27. for record in cur:
28.     lat.append(float(record[12]))
29.     lon.append(float(record[13]))
30.     latgo.append(float(record[14]))
31.     longo.append(float(record[15]))
32.     qual.append(float(record[16]))
33.
34. dist = []
35. for i in range(len(lat)):
36. #convert longitude/latitude to RD NEW CRS
37.     x, y = transform(p1, p2, lon[i], lat[i])
38.     xgo,ygo = transform(p1,p2,longo[i],latgo[i])
39.     if (miny < y < maxy) and (minx < x < maxx):
40.     #calculate distance between coordinates
41.         interval = distance(x,xgo,y,ygo)
42.         dist.append(interval)
43.
44. cur.close()
45. conn.close()
```