UNIVERSITÄT KOBLENZ · LANDAU

IWVI

Vinio – an Open Service Infrastructure approach to IoT monitoring of vineyard wellness

A Customer operated information system offering high geographic and time resolution data in real-time and its variation - fulfilling winemaker's target to identify timely vineyard care needs

Bachelor Thesis
Zur Erlangung des Grades eines
Bachelor
im Studiengang Wirtschaftsinformatik

vorgelegt von

Philipp Thom

Supervisor: Prof. Dr. Susan P. Williams
Enterprise Information Management (FG EIM)

Supervisors 2: Patrick Nitschke
Enterprise Information Management (FG EIM)

Koblenz, in October 2018

Declaration / Erklärung

I declare that,

This thesis presents work carried out by myself and does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university. To the best of my knowledge, it does not constitute any previous work published or written by another person except where due reference is made in the text.

Ich versichere,
dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden. Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.

Philipp Thom
Koblenz, Datum

Abstract (English)

This thesis connects the endeavors of the winemaker's intention in perfect and profitable wine making with an innovative technological application to use Internet of Things. Thereby the winemaker's work may be supported and enriched – and enables until recent years still unthinkable optimization of managing and planning of his business, including close state control of different areas of his vineyard, and more than that, not ending up with the single grapevine. It is exemplarily shown in this thesis how to measure, transmit, store and make data available, exemplarily demonstrated with "live" temperature, air and soil humidity values from the vineyard. A modular architecture was designed for the system presented, which allows the use of current sensors, and similar low-voltage sensors, which will be developed in the future.

By using IoT devices in the vineyard, the winemaker advances to a new quality of precision of forecasted data, starting from live data of his vineyard. Of more and more importance, the winemaker can start immediate action, when unforeseen heavy weather conditions occur. Immediate use of current data enabled by a Cloud Infrastructure. For this system, an open service infrastructure is employed. In contrast to other published commercial approaches, the described solution is based on open source.

As an alone-standing part of this work, a physical prototype for measuring relevant parameters in the vineyard was de-novo designed and developed until fulfilling the set of specifications. The outlined features and requirements for a functioning data collection and autonomously transmitting device was developed, described, and the fulfilment by the prototype device were demonstrated. Through literature research and supportive orientationally live interviews of winemakers, the theory and the practical application were synchronized and qualified.

For the development of the prototype the general principles of development of an electronic device were followed, in particular the Design Science Research development rules, and principles of Quality Function Deployment. As a characteristic of the prototype, some principles like re-use of approved construction and material price of the building blocks of the device were taken into consideration as well (e.g. housing; Arduino; PCB). Parts reduction principles, decomplexation and simplified assembly, testing and field service were integrated to the development process by the modular design of the functional vineyard device components, e.g. with partial reference to innovative electrical cabinet construction system Modular-3.

The software architectural concept is based on a three-layer architecture inclusive the TTN infrastructure. The front end is realized as a rich web client, using a WordPress plugin. WordPress was chosen due to the wide adoption through the whole internet, enabling fast and easy user familiarization. Relevant quality issues have been tested and discussed in the view of exemplary functionality, extensibility, requirements fulfilment, as usability and durability of the device and the software.

The prototype was characterized and tested with success in the laboratory and in field exposition under different conditions, in order to allow a measurement and analysis of the fulfilment of all requirements by the selected and realized electronic construction and layout.

The solution presented may serve as a basis for future development and application in this special showcase and within similar technologies. A prognosis of future work and applications concludes this work.

Abstract (German)

Diese Arbeit verbindet die Geschäftstätigkeit von Winzern im Weinbau mit einer innovativen technologischen Anwendung des Internet of Things. Die Arbeit des Winzers kann dadurch unterstützt und bereichert werden – bis hin zu einer bisher nicht möglichen Bewirtschaftungsoptimierung, insbesondere bei einer Überwachung einzelner Lagen bis hin zum einzelnen Rebstock. Exemplarisch werden Temperatur-, Luftfeuchtigkeit- und Bodenfeuchtigkeit-Daten gemessen, übertragen, gespeichert und bereitgestellt. Durch ein modulares Design des Systems können heute verfügbare Sensoren und gleichartige Niedervolt-Sensoren, die künftig entwickelt werden, sofort eingesetzt werden.

Durch IoT-Geräte im Weinberg erhält der Winzer eine neue Qualität der Genauigkeit der Vorhersage auf Basis aktueller Zustandsdaten seines Weinbergs. Zusätzlich kann er bei unvorhergesehenen Wetterbedingungen sofort eingreifen. Die sofortige Nutzbarkeit der Daten wird durch eine Cloud Infrastruktur möglich gemacht. Dabei wird eine offene Service-Infrastruktur genutzt. Im Gegensatz zu anderen bisher veröffentlichten kommerziellen Ansätzen ist dabei die beschriebene Lösung quelloffen.

Als eigenständiger Bestandteil der Arbeit wurde ein physikalischer Prototyp zur Messung relevanter Parameter im Weinberg neu entworfen und bis zur Erfüllung der gesetzten Spezifikationen entwickelt. Die skizzierten Merkmale und Anforderungen an eine funktionierende Datensammlung und ein autonom übertragendes IoT-Gerät wurden entwickelt, beschrieben und die Erfüllung durch das Prototypgerät demonstriert. Durch Literaturrecherche und unterstützende, orientierende Interviews von Winzern wurden die Theorie und die praktische Anwendung synchronisiert und qualifiziert.

Für die Entwicklung des Prototyps wurden die allgemeinen Prinzipien der Entwicklung eines elektronischen Geräts befolgt, insbesondere die Entwicklungsregeln von Design Science Research und die Prinzipien des Quality Function Deployment. Als ein Merkmal des Prototyps wurden einige Prinzipien wie die Wiederverwendung von bewährten Konstruktionen und die Materialpreise der Bausteine des Prototypen wurden ebenfalls in Betracht gezogen (z. B. Gehäuse; Arduino; PCB). Teilezahl-Reduktionsprinzipien, Dekomplexierung und vereinfachte Montage, Prüfung und Vor-Ort-Service wurden in den Entwicklungsprozess durch den modularen Aufbau der funktionellen Weinberg-Gerätekomponenten integriert, wie es der Ansatz des innovativen Schaltschrankbau-System Modular-3 beschreibt.

Das Software-Architekturkonzept basiert auf einer dreischichtigen Architektur inklusive der TTN-Infrastruktur. Das Frontend ist als Rich-Web-Client realisiert, als ein WordPress-Plugin. WordPress wurde aufgrund der weiten Verbreitung über das gesamte Internet und der Einfachheit in der Bedienung ausgewählt, was eine schnelle und einfache Benutzereinweisung ermöglicht. Relevante Qualitätsprobleme wurden im Hinblick auf exemplarische Funktionalität, Erweiterbarkeit, Erfüllung von Anforderungen, Verwendbarkeit und Haltbarkeit des Gerätes und der Software getestet und diskutiert.

Der Prototyp wurde mit Erfolg im Labor und im Einsatzgebiet unter verschiedenen Bedingungen charakterisiert und getestet, um eine Messung und Analyse der Erfüllung aller Anforderungen durch die geplante und realisierte elektronische Konstruktion und Anordnung des Prototypen, zu ermöglichen.

Die entwickelte Lösung kann als Grundlage für eine zukünftige Anwendung und Entwicklung in diesem speziellen Anwendungsfall und ähnlichen Technologien dienen. Ein Ausblick möglicher zukünftiger Arbeiten und Anwendungen schließt diese Arbeit ab.

Table of Contents

List of Abbreviations

A
B
C
D

DSR – Design-Science Research

E

EU – European Union

F
G
H
I

IoT – Internet of Things

J

JSON – JavaScript Object Notation

K
L
M

MQTT – Message Queuing Telemetry Transport

N
O
P

PCB – Printed Circuit Board

Q
R

REST – Representational State Transfer

S
T

TTN – The Things Network

U
V
W
X
Y
Z

List of Figures and Tables

1 Introduction

Industry 4.0 is massively changing the industrial, economic and private world – however, it is in part still a mystic for many of us, and masses of people are not sure about contents and changes, which will come to our society with realization of concepts, technological innovation and changed conditions of working and living. Human work is said to be replaced in the next decades by machine and computer-based work on a scale, which will have consequence on employees, and the content of work of their jobs (Benedikt Frey et al., 2013). Assumed this socio-technological dislocation will be mastered by mankind, there are advantages of Industry 4.0 technological changes and advancement realization, touching all areas of commercial and private life in the near future. Even formerly unthinkable but useful applications will be realized, giving large contribution of economic efficiency, welfare and safe-guarding mankind's future.

A traditional cultural habit is since more than 2000 years the cultivation of vine (Medela et al., 2013). It is a challenge for all of us, to combine tradition with future, and as a practical advancement of the modern winemaking would be the winemaker's advancement of knowledge about the wellness of his matter of care, the optimum growth and quality of the grapes in his vineyard.

This thesis connects the endeavors of the winemaker's intention with Internet of Things application in an open service infrastructure approach and as a base-foundation for future application and evolution of this specific and similar technologies.

1.1 Problem Definition and Motivation

Studies on world's climate change point out the near-future challenge that successful winemaking needs adaptation to changes in the climatic conditions over the year, and recently, needs both, adaptation to the single events of local weather disturbances (storm, massive rain downfalls), and, as well the geographical shifting of winemaking zones (Stock, 2006). Thus, as a profound change, today's preference areas for white wine cultivation, will move towards the northern hemisphere, and turn over the decades to red wine cultivation zones, just due to the climate change. On the micro-scale, winemakers will need to develop many more tools for climate and other parameters' short-term prediction, and for the acute condition of their vineyards, where the fruit of their work endeavors come from. As an outlook, the importance of development of new technology can be seen on a larger scale as a contributing factor for the of earth population's world nutrition by agriculture and farming as a whole.

More and more economic sectors benefit from robotic and automated systems […] and becoming increasingly common in all economic sectors (Seelye *et al.*, 2011).

Extending the robotic and automated systems known, Haller et. al. defined this extension Internet of Things (IoT) in 2008 as “A world where physical objects are seamlessly integrated into the information network, and where the physical objects can become active participants in business processes. Services are available to interact with these “smart objects” over the Internet, query their state and any information associated with them, taking into account security and privacy issues.”

According to this definition there will be a fundamental role to services delivering functionality to operate in the information network and enable businesses to benefit from the service-based integration of physical objects. As a point of technological interest in upcoming Industry 4.0 there is a foreseen link between enhancing the basic production of human life – nutrition and food – with the systems of IoT technologies. As one core application can be seen to provide plant growth data and monitoring core parameters of their successful growth and maturity by support with a large range of IoT sensors, even many at low-cost (Shrouf, Ordieres and Miragliotta, 2014).

In order to combine both service-based information network from possibilities brought by the before defined IoT and plant growth data collection this thesis will show a flexible and easily deployable open service approach to make use of the information network to monitor vineyard wellness as a practical use case. Although it is capable of an unforeseen number of other use cases that could be realized by the intended implementation. As a foundation to realize the prototype on modern IoT technology a basic already existing infrastructure for the Internet of Things is used. It’s an open service project called The Things Network (TTN). They describe themselves as „The Things Network is building a network for the Internet of Things by creating abundant data connectivity […].“ (The Things Network, 2017).

Covering those innovative options and new application of IoT and TTN there could be a way of never-seen-before benefit, using this technological innovation as opportunity, firstly to assure quality and quantity, and secondly to enhance productivity in agricultural production profoundly.

In general the agricultural production can be optimized by plant monitoring of “[…] critical temperature, humidity and soil signals [that] are collected real-time in agriculture production process […]” (Zhao *et al.*, 2010, p. 463). Following this idea multiple use cases can be developed, which combine possibilities of IoT realized with TTN, in order to bring business value to the agricultural economy.

As a generic segment of agriculture, the winemakers’ efforts are to be rendered under complex growth and varying weather conditions and following a high number of restricting factors. These are, among others, preliminarily the quantities (yield), rules of the territory, region and government, the appropriate use of growth accelerators (chemicals, biologicals), use of protection measures (among them pesticide, fungicide use), indirect target factors like grape liquid density for glucose content, general and individual requirements of the grapes of concern, the different soils given in their vineyards, and differences in microclimate conditions throughout the individual position of the different vineyards of a terroir and even the individual grapevine location within the vineyard. The intended result is to yield high quality fruit and final product, being the wine of their produce, under an economic effort, to lead finally to profit and joy. As a conclusion, winemakers can benefit from real-time data collection in high local and time resolution to monitor vineyard wellness.

To prove business value in this approach there can be multiple cases derived from it which can be e.g. partial root-zone drip irrigation (Jones and Davis, 2000, p. 253). The main goal though is to give winegrowers a decision basis to care at the right time in the right way. But surely the data could inevitably bring gain in quality of vines and earnings control. Being able to decide upon in-place collected data could be even more important for vine grower's businesses regarding the aspects of global warming and its relating climate change.

On the ever-improvable Customer advantage, there is the vision to develop, based on GPS geo location (BIG DATA), algorithms in order to lead to a simplified recommendation to winemakers, what could be advisable actions for caring best for their vineyards. This approach may be seen as a second level, after firstly, generate the basis for direct remote monitoring of the vineyard wellness. "While [this will bring] easing the management of vineyard and winery, as well as improving traceability […]" (Medela *et al.*, 2013). Even if such algorithmic attempts are contemporarily reported from Australian vineyard observations, it was seen out-of-scope of the thesis presented here.
As a regional assay for demonstrating usefulness of the presented approach of remote vineyard monitoring and allow the vine maker pro-active management especially in changing climate conditions as a validity for this IoT based application to support vine maker pro-active management, an agro-climatological characteristics research of central Moselle valley was performed and proves of this relation of climate change and more difficult planting conditions. „The increased temperature and sunshine duration in combination with drier conditions in May and June lead to a considerable forward displacement of the flowering dates and to a fortunate displacement of the fruit ripening stages into the hot, dry and sunny midsummer" ('Agro-climatological and phenological characteristics of the Middle Moselle valley. Effects of climate change to grapevine, Moselle River Region.', 2001).

To bring the relevance of a smart vineyard into perspective there is the evaluation of the total market size for IoT devices, according to publication by Statista was analyzed for the year of 2013 to span on the global scale a

market size of <u>485.6 billion U.S. dollars (global) for IoT devices</u>

(Statista, 2018).

IoT devices for vineyards – market size

The size of the IoT market explains clearly the heavy interest and large initiative of large global corporations, to support and to pilot small developer initiatives in the wine-making field. For the presented vineyard-IoT application and the relevant market segment of total IoT, there are initiatives and market entry strategies known e.g. by Ericsson and Intel. Though a smaller scale of market may apply to global number of winemakers and vineyards, it is of high public attention.

In 2017 and 2018, German Chancellor Angela Merkel, the Minister of Agriculture, Julia Klöckner, and Dorothee Bär, Minister for Digitalization, were introduced to the winegrower's situation and one commercial solution "TracoVino" by the relative provider (www.myomegasys.com). Besides market expectation, the technological solution is of high public interest in the context of current pressure of German government, to demonstrate active interest and success in Industry 4.0 applications, leading to recent appointment of the Minister of Digitalization.

Global number of winemakers in the framework of this thesis are estimated as follows: wine making is known from 70 countries worldwide, and the largest are put to the table below. World production is estimated to 36 billion bottles of 0,75 l content (36 Billion Bottles, 2015).

Table 1. 1 World Vineyard Acreage By Country (Trade Data And Analysis, 2014)

Country[1] (top producers 2011)	Winemakers[1] total	Wine-makers[1] with commercial distribution	Liters[2] Produced 2014 in Hectoliter (% of global)	% [2] of global wine production	Rank [2]	Acreage [2] in 1.000 acres
France	115.000	27.000	4.670.100	16.54	1	1.876
Italy	1.000.000	200.000	4.473.900	15.85	2	1.705
Spain	280.000	112.000	3.820.400	13.53	3	2.340
USA	3.500		3.021.400	10.70	4	1.035
Argentina			1.519.700	5.38	5	552
Australia	3.000		1.200.000	4,25	6	341
South Africa			1.131.600	4.01	7	316
China			1.117.800	3.96	8	1.974
Chile			1.050.000	3.72	9	521
Germany	69.000	28.000	849.300	3.01	10	253

TOP TEN (above)			22.854.200	80.95		10.913
Global			28.230.400	100.00		17.960
Russia			720.000		11	
Portugal	58.000	26.000	623.800		12	
Romania			511.300		13	
Greece	185.000	37.000	334.300		14	
New Zealand			320.400		15	
Hungary			294.400			

For the exemplary German pilot area Mosel[1], there are 3.200 winemakers registered with 89.000.000 litres of wine (2014 with top annual quantity; 91% white wines, 9% red wines), with 8.770 hectar (87.700.000 m²) wineyard area in total. There are 6 areas of winemaking, which are Moseltor, Obermosel, Saar, Ruwertal, Bernkastel, and Burg Cochem. There are 19 registered large category areas (Großlagen), and 520 small areas (Einzellagen). 60 mio grapevines (Weinstöcke) are distributed among all these areas. The size of a cultivation area of one winemaker averages to 27.000 m². Since many winemakers with areas below 50.000 m² give up their business, the larger winemakers are in possession of vineyards with a total size of more than 50.000 m². Before analyzing the situation in more detail and precision, one could say, that only 60 winemakers will produce 80% of the wine of that region. Assuming this particular area has a low degree of automatization, and high degree of hand-made cultivation under difficult geological condition (more than 30% extremely steep vineyards, “Steillagen”), the potential client in that area will approximate to a total number of more than 100 winemakers, with an approximate requirement to control more than 50.000 m². This might lead to a consumption of 30-50 IoT devices per client, thus approx. a minimum of 5.000 devices for a first installation.

When relating to competitors and to requirements, the market is open, and currently pilot field-study is ongoing with the system TracoVino from MyOmega, performed by Winemaker Haart in Piesport/Mosel (Frank Feil, 2016). It was reported, that the winemaker can obtain with this system a forecast for the condition of his vineyards, potential pest, and grape yield and quality, and may start the relative actions after remote information.

This development phase of the cited monitoring system may add much for increased awareness of potential useful application in the field. A comparative study may be orchestrated between the device presented here, and the TracoVino system, in order to identify specific advantages of the presented system layout over the cited system, and other competitors.

[1] www.weinland-mosel.de

The following section will give reasons and base for the targeted winemakers market entry, and to expand from viniculture into different markets.

As result of this thesis there shall be a statement regarding the relation of IoT deployment and vine maker's businesses, proving it by a prototype implementation. The prototype shall be technically working and shall be to a certain level reliable in real world use.

1.2 Research Objectives and Research Questions

This thesis follows the principles and objectives of the design science research abbreviated as DSR (Hevner and Chatterjee, 2010). As fundamentally required to apply any kind of DSR there must be kept in mind, a "[…] justified theory that is not useful for the environment contributes as little to the IS literature as an artifact that solves a nonexistent problem" (Hevner *et al.*, 2004). Covering this fundamental requirement there was shown a challenging relation between agricultural planting cares and climate warming in sections before. Justifying the research method of DSR, research will be worked in an iterative cycle that reaches to bring "technology-based solutions to important and relevant business problems" (Hevner and Chatterjee, 2010). To validate the relevance of the business problem there is posted a hypothesis in section 1.1 to overcome those challenges by using IoT technologies.

Already supported by literature references, the objective derived from the hypothesis is to develop and describe an open service infrastructure approach to monitor vineyard wellness IoT-based with high local and time resolution and identification of needs for timely care of vineyards. One main reason to develop this approach is to realize earnings control on terroir level. Though focus is given to demonstrate a general pilot approach as a blueprint for any other sensing supported type of care.

Measured data is selected in the scope of a pragmatic pilot approach: air humidity, soil moisture and air temperature. Nevertheless, any kind of data could be collected by extending the developed open service infrastructure. More details on extensibility limitations, found by own experimentation, are expressed in Section 3.1. Further limitations of this thesis and its described prototype are, not to extend the applied research to the delivery and display data to support e.g. computer-controlled irrigation systems, which work in automated or man-controlled fashion driven by collected soil moisture or air humidity and temperature data. Though those applications are only examples of use cases and are not explicitly described in implementation with real world hardware. These real world hardware is out of scope of this thesis, following the definition (Hevner *et al.*, 2004, p. 83) uses to express that design-science research is mostly not a full-grown information system that can be used in practice as is. Additionally, derived care decisions must be taken by winegrowers themselves as this cannot be handled in the scope of this thesis as it is looking from a technology-based point of view and focuses on the technical aspects of the realization of the open service infrastructure to support the described business case. But surely, linking business need with evidences from winegrower's point of view, is a further requirement in a holistic realization. As an impact this causes to left decision on various specific cares individually derived to be made based on winegrowers experience themselves. Those derived cares made from delivered data are in effect not explicitly described.

The first research question asks if there is a technological solution to support winegrowers in their efforts to prevent threats arising from climate change by monitoring climate data directly in place in their vineyards?

1. As a resulting objective there shall first be a technically working and, to a certain level, reliable prototype as an operable outcome and practical, physical realization.
2. The deployed and characterized infrastructure shall secondly be able to collect, transmit and store soil moisture, air temperature and relative air humidity data from multiple statically geo-positioned sensors and display them in charts and graphs, visualized in a responsively designed web portal.

The second objective is to show feasibility to support Winegrowers in monitoring their vineyards by an open IoT service infrastructure, to enhance earnings control even on terroir level.

To explain and validate the chosen IoT approach the following is about to justify this requirement.
Climate Change with effects of weather extrema threatens agriculture in general. Therefore, it threatens the winegrower’s economy too.
The former traditional craft of winegrowing was about to stay natural and free from artificial growth support by irrigation systems in early 2000. Nevertheless, changes in federal states law, which are based on EU changes, made irrigation systems in viniculture legal. In the federal parliament there was a so-called small inquiry directional initiated. In question No. 2 of this small inquiry the parliament confirmed the permission to water steep slope vineyards (Ministerium für Wirtschaft, Verkehr, 2002). This enables winegrowers to use artificial irrigation systems even in viniculture.

Extending the first objective it is central to research and approve or disapprove the applicability of technology-based IoT solutions to important and relevant business problems of winegrower’s economy with the following two questions to be answered.

1. Is this infrastructure able to support smart irrigation systems?
2. Is there a potential in preventing failure of grapevine earnings by using IoT based vine production and using the benefits from failure prevention to strengthen winegrower’s business?

1.3 Research Design

The previously described relations, possibilities and derived decisions will be developed in detail throughout the following chapters and must be validated and extended to reach a sustainable level of completeness. The underlying research design follows explanatory research, which is described by DSR. As a "[…] design artifact is complete and effective when it satisfies the requirements and constraints of the problem it was meant to solve" (Hevner *et al.*, 2004, p. 85). Also, this level is reached once someone can continue the developed work on extending the prototype developed in this thesis.

In Figure 1. 1 there are two DSR cycles visualized in a single graph. The first Cycle is meant to represent the main goal of evaluate assembly techniques for prototype creation. It is intended to elaborate and research how to use all fundamental skills required for this thesis' purpose to create an open service infrastructure for IoT devices. Initially there was simply the try to create a single device prototype to only fulfill a winegrower's need to monitor his vineyard remotely. Nevertheless, in just the first research activities an extension to this approach is pretty clear to find. The use case of a vineyard enabled with smart IoT technology is just one single use case that is in its kind similar to multiple other use cases. This causes to extend the first DSR cycle to a second cycle. In evaluation phase of the first DSR cycle the before described findings are so clear, that the second DSR cycle focuses on generalization of the smart vineyard use case. The first DSR cycle is directly closed by the evaluation result of the need for an open service infrastructure. The used hardware, software and techniques found in the first DSR cycle are then to be reused and extended to a generic and fully open service infrastructure approach as a foundation for any other suitable application of IoT.

Following there are basic methods described. As an evaluation method to measure wireless LoRa transmission ranges, a comprehensible and provable way is to use a cell phone app that is connected to TTN and uploads its measured data publicly to ttnmapper.org. The cell phone app uses MQTT to transmit the current GPS location measured by the cell phone to link this location with a data package, that is send over LoRa to the TTN network and received at the TTN gateway, at the same time.

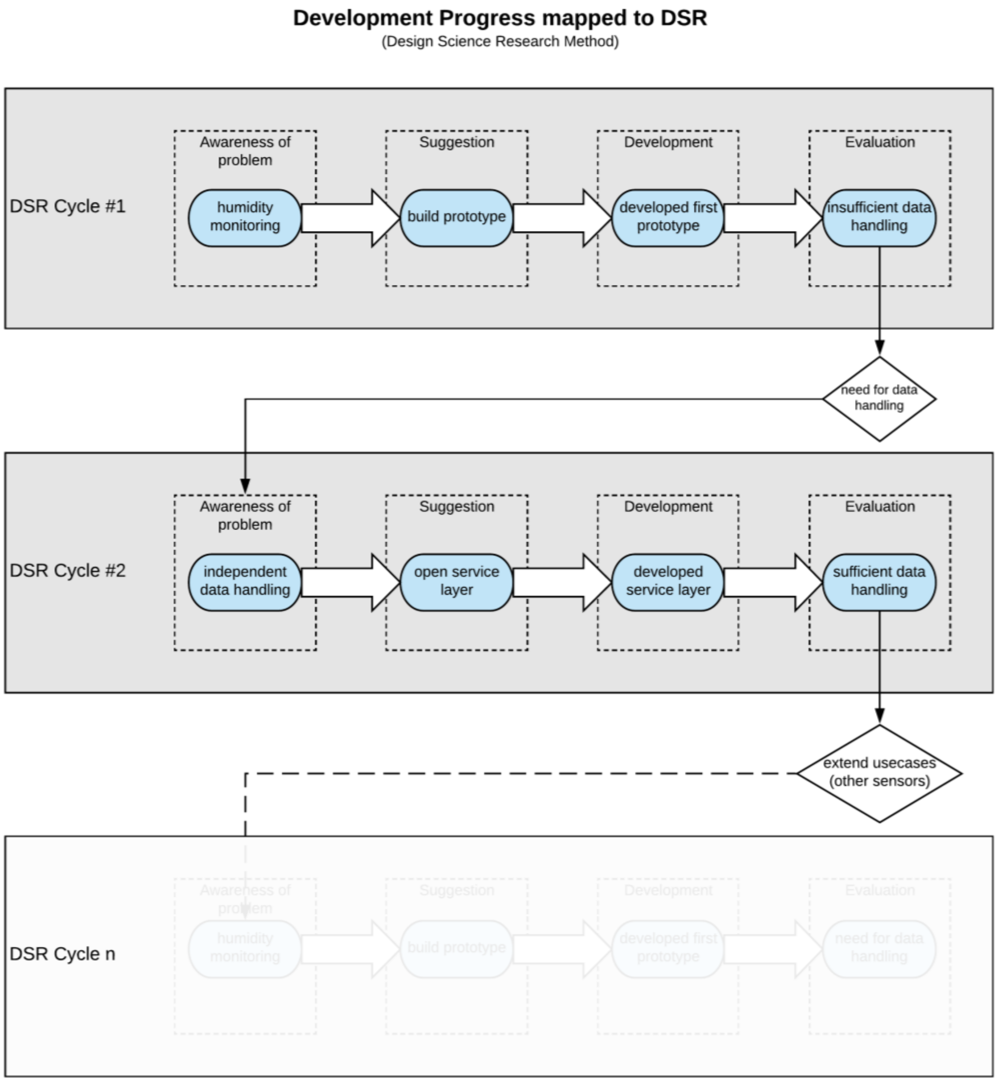


Figure 1. 1 Development Progress mapped to DSR (own illustration)

DSR guideline concerning Data collection, use, processing and aggregation

As data source, there will mostly be iterative gain of knowledge, reliance "[…] on creativity and trial-and-error search" and improvement of the prototype as these "are characteristic of such research efforts" (Hevner *et al.*, 2004, p. 81). Concomitant to the documentation of the implementation progress there will be internet research in academic writings according to the topic of IoT and specific needs and circumstances according the winery economy. Data will also be collected in support of The Things Network.

As introduced before number three of the seven DSR guidelines is about evaluation. To be more precise the evaluation will focus on observational, experimental and descriptive design evaluation methods (Hevner *et al.*, 2004, p. 86). To justify the first evaluation method the artifact must be studied "[…] in depth in business environment" (Hevner *et al.*, 2004, p. 86). In reflection to the business environment of this artifact it must be studied in a vineyard. Going on evaluating the artifact it must be simulated "[…] with artificial data" (Hevner *et al.*, 2004, p. 86). Simulation will be done automatically by testing it in the business environment. Data will be sent and processes during the whole observational testing.

DSR guideline concerning Data collection, use, processing and aggregation
As data source, there will mostly be iterative gain of knowledge, reliance "[…] on creativity and trial-and-error search" and improvement of the prototype as these "are characteristic of such research efforts" (Hevner *et al.*, 2004, p. 81). Concomitant to the documentation of the implementation progress there will be internet research in academic writings according to the topic of IoT and specific needs and circumstances according the winery economy. Data will also be collected in support of The Things Network.

DSR guideline concerning Evaluation:
As introduced before, number three of the seven DSR guidelines is about evaluation. To be more precise, the evaluation will focus on observational, experimental and descriptive design evaluation methods (Hevner *et al.*, 2004, p. 86).
To justify the first evaluation method the artifact must be studied "[…] in depth in business environment" (Hevner *et al.*, 2004, p. 86). In reflection to the business environment of this artifact it must be studied in a vineyard. Going on evaluating the artifact it must be simulated "[…] with artificial data" (Hevner *et al.*, 2004, p. 86). Simulation will be done automatically by testing it in the business environment. Data will be sent and processed during the whole observational testing.
"The business environment establishes the requirements upon which the evaluation of the artifact is based. […] Thus evaluation includes the integration of the artifact within the technical infrastructure of the business environment" (Hevner *et al.*, 2004, p. 85).
"IT artifacts can be evaluated in terms of functionality, completeness, consistency, accuracy, performance, reliability, usability, fit with the organization, and other relevant quality attributes" (Hevner *et al.*, 2004, p. 85).

DSR guideline concerning Display Data:
As method to display the results, a visual web dashboard framework will be used to visualize the collected data from IoT devices. It will be deployed as a web application. As DSR is "limited to the activities of building the IS infrastructure within the business organization." This thesis will also be limited to not include "issues of strategy, alignment, and organizational infrastructure design" which are outside the scope of this thesis (Hevner *et al.*, 2004).

DSR guidelines concerning IS
"Design-science research in IS addresses what are considered to be wicked problems (Brooks 1987, 1996; Rittel and Webber 1984)" (Hevner *et al.*, 2004, p. 81).

“The critical nature of design-science research in IS lies in the identification of as yet undeveloped capabilities needed to expand IS into new realms “not previously believed amendable to IT support (Markus et al. 2002, p. 180)” (Hevner *et al.*, 2004, p. 84).
“To be relevant […] research must address the problems faced and the opportunities afforded by […] information technology” (Hevner *et al.*, 2004, p. 85).

DSR guidelines on Design Process Figure 1. 2 Build and Final Design Artifact: Having focused on topics of this thesis’ subjects there are also more references then DSR needed to be considered when observing the underlying research method DSR, and this is described in detail by Hevner et al., 2014. “The design process is a sequence of expert activities that produces an innovative product (i.e., the design artifact). The evaluation of the artifact then provides feedback information and a better understanding of the problem in order to improve both the quality of the product and the design process. This build-and-evaluate loop is typically iterated a number of times before the final design artifact is generated” (Markus et al. 2002). „A justified theory that is not useful for the environment contributes as little to the IS literature as an artifact that solves a nonexistent problem“ (Hevner *et al.,* 2004).

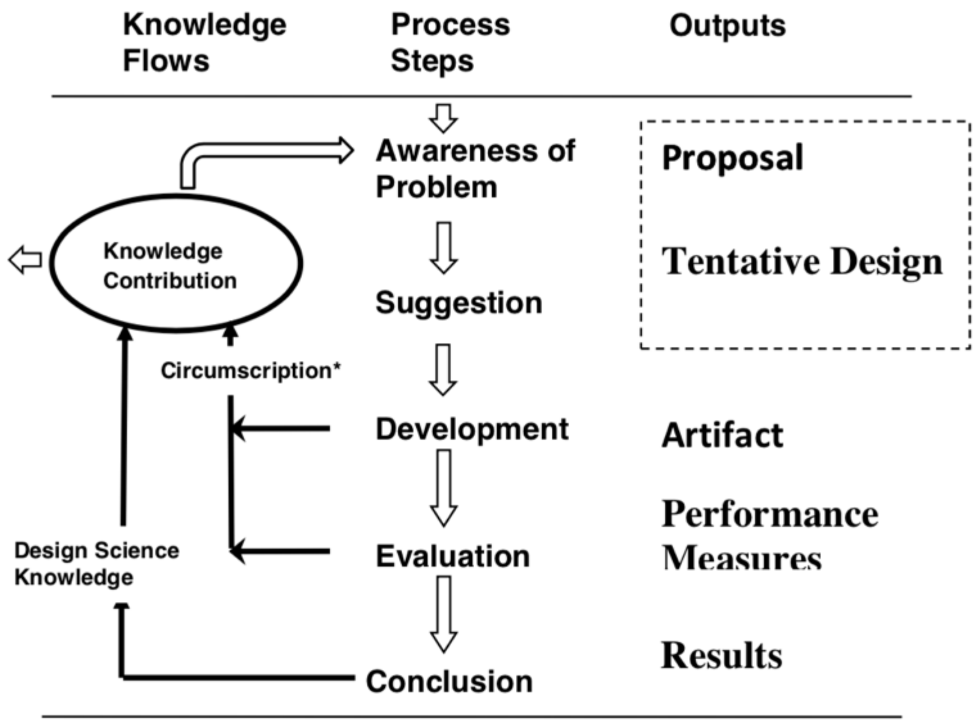


Figure 1. 2 General Design Science Research Model (Hevner and Chatterjee, 2010)

Hardware development follows DSR during development phase. To bring more guidance into the prototype development process it is partly extended by Quality Function Deployment (QFD) method, which describes a “[…] purpose […] to assure that true customer needs are properly deployed throughout the design, build and delivery of a new product, whether it be assembled, processed, serviced, or even software, and to improve the product development process itself“ (Akao and

Mazur, 2003). Improvements to be realized in the future are referenced in chapter 5. The main areas defined for product development from QFD are quality, technology, cost and reliability.

1.4 Structure of Thesis

Following the DSR cycle there was a fundamental finding. During the initially intended prototype development of one single device proving the concept as described in section 1.1 a need for change of the initial approach occurred. This change brought with it the decision to run two DSR cycles to advance the foundation of the infrastructure approach rather than just a prove of concept for a single use case. More detail will follow at the end of chapter 3 during the evaluation of DSR cycle number one. In order to analyze the quality of the research result they will be evaluated in a manner of checking the acceptance criteria reached, defined as DSR Requirements. There are questions to answer like: “Does the design artifact improve the environment and how can this improvement be measured” (Hevner and Chatterjee, 2010)?

But before there can be any DSR related research there must be basics and theory about the underlying research techniques and frameworks, which will be handled in chapter 2. This chapter focuses on two major topics. One will be some facts about basic electronics to clarify this theory due to the perspective of this thesis from an informatics and economical point of view. The other will be about informatics paradigms that come in place for the infrastructure approach.

Having described those chapters the fourth chapter will be about the second DSR cycle which will describe, document and evaluate the Infrastructure development and implementation. As third component of the whole infrastructure there should obviously be a front-end to display and give user interaction it’s place. The development needed for this will be elaborated in chapter 4.

Concluding the thesis there is chapter 5 about analysis of all findings in whole perspective and concluding those findings by presenting an outlook and a future work perspective to show the reached current use and required future needs.

There will be a proposition concluding this thesis in the domain of opening the view to an outlook about the used technology and its opportunities derived from the described approaches on open service infrastructure monitoring in an agricultural economical sector.

2 Theoretical Foundations

The theoretical foundations ground the knowledge this thesis is based on. In this chapter there are topics of Internet of Things in 2.1, LoRaWAN in 2.2 and Internet of Things in Agriculture in 2.3.

2.1 Internet of Things

Trying to solve issues of modern agriculture by the use of IoT, there are several aspects to cover. IoT has increasingly become important to global market aspects as “IoT could grow into a market worth $7.1 trillion by 2020 […]” (Wortmann and Flüchter, 2015). This is a remarkable suggestion and follows intuition while thinking about the potential of IoT as it can be seen as a “basic idea [to connect] virtually every physical thing in this world […] to the Internet” (Zhao *et al.*, 2010). Additionally “[t]he phrase “internet of things” has arisen to reflect the growing number of smart, connected products and highlight the new opportunities they can represent“ (Porter and Heppelmann, 2014). According to (Atzori et. al 2010) definitions of IoT, that can be found in appropriate literature, can be grouped into three types. The first type is focusing on the things that are connected to the internet. The second type is grouped by connectivity through IP-based networks and based on data transportation protocols. Finally, the third type is grouped by aspects of data handling or management of large volumes of data. Additionally the history of IoT is one of a long term in the realm of computer industry, as “[t]he origins of the term date back [even] more than 1[8] years […]” (Wortmann and Flüchter, 2015). The start of research on IoT like technology is linked to research of “Massachusetts Institute of Technology (MIT) on networked radio-frequency identification (RFID) infrastructures […]” (Wortmann and Flüchter, 2015). There are two types of RFIDs when viewing at a meta-level. One is an active RFID, the other one is a passive RFID. By using the most popular type, which is according to Want, 2006, the passive one, there are core principles that apply to many IoT use cases. A somehow electronic sensor measures some data that can be transmitted, stored and processed somewhere. RFID stores an ID which e.g. can be linked to an object that can be identified by a relation or linking stored in a database. This makes possible to abbreviate an observation about IoT in general. Logic is mostly excluded from the IoT devices. Logic comes into perspective once multiple single data measurements are connected, cumulated or in other aspects related or linked to each other. This explored aspect is also a fundamentally origin of this thesis’ prototyping process and deciding how to implement logic. More detail to this aspect will follow in section 3.3 and chapter 4.

As efforts grow on research in IoT, simultaneously hardware development in terms of cheapening, increasing power efficiency and multiple other aspects of improved hardware specifications are driven by global growth of the computer industry. This evolution was “made possible by vast improvements in processing power and device miniaturization” (Porter and Heppelmann, 2014, p. 4). This global market growth of computer industry brought a hype of cloud computing in focus of last years. Topics like “Big Data”, “Artificial Intelligence”, “Virtual Reality” and many more arise from increased computing power and better cost efficiency, making cloud computing more important as this global infrastructure is the necessary foundation that make all those services and business models possible. A fundamental question appears, which asks:” Why now? [There is an answer.] An array of innovations across the technology landscape have converged to make smart, connected products technically and eco-nomically feasible. These include breakthroughs in the performance, miniaturization, and energy efficiency of sensors and batteries; highly compact, low-cost computer processing power and data storage, which make it feasible to put computers inside products […]” (Porter and Heppelmann, 2014).

2.2 The Things Network and LoRaWAN

Mentioning cloud infrastructures there is a important one in terms of IoT. It’s called The Things Network and was founded in Amsterdam, Nederland in 2015 by Wienke Giezeman (Giezeman, 2018).
TTN is about enabling low power devices to use long range Gateways to connect to an open-source, decentralized Network to exchange data with Applications and Platforms (The Things Network, 2017). Speaking of low power long range network devices this brings some restrictions and limitations but also some chances and benefits from technology related circumstances. Some of them can be listed as extreme low bandwidth, need for long battery life, need for cost effectiveness and low cost. But also, the chances to use a worldwide open source, community driven project and infrastructure that is headed towards low cost and usability even for individuals. More detail on TTN technology aspects will follow later on in this chapter. TTN is also an element in one of the three layers of the technology stacks of IoT, the cloud layer. Following the perspective described by (Wortmann and Flüchter, 2015). The “technology stack[s are] usually composed of three core layers, i.e., the thing or device layer, the connectivity layer and the IoT cloud layer” (Wortmann and Flüchter, 2015).
Regarding the device layer there are specialized hardware components to be used. A wide range of sensors, actuators or processors are available to fit to any suitable use case. This thesis focuses on only basic examples of sensors to proof the concept. Nevertheless, the prototype is designed with extensibility options to use any kind of sensors or actuators to be connected, that fit the hardware specifications explained in detail in section 3.3. The above lying connectivity layer describes the communication protocols used between devices and the cloud. As an example, MQTT (Message Queuing Telemetry Transport), which is developed by IBM, can be named, which enables devices to send and receive information through network connections persistently. Additionally, “It consumes very little power, and has a very high accuracy. So it basically meets the needs of application.”, which are low power and low bandwidth (Tang *et al.*, 2013). Covering MQTT there is also an other protocol to be named, it is called REST, which is the abbreviation for “Representational State Transfer (REST)“ (Battle and Benson, 2008). REST is a

protocol to be used on a higher abstraction level above HTTP protocol and it "deals only with data structures and the transfer of their state. REST's simplicity, along with its natural fit over HTTP, has contributed to its status as a method of choice for Web 2.0 applications […]" (Battle and Benson, 2008). "Representational State Transfer is a pattern of resource operations that has emerged as a *de facto* standard for service design in Web 2.0 applications" (Battle and Benson, 2008). More design aspects regarding transmission protocols and realization made in the developed prototype will follow in section 4.1. Such a protocol comes into account when interaction between humans and devices is important, and hardware (technology) vanishes in the background (Weiser, 1991). Justifying this, Collina et al. defined that "[i]n general the requirements of machines and people are distinctly different […]", which causes to introduce specially designed protocols for this purpose. They added that MQTT can be combined with REST, which "envisions a world where digitally enhanced objects […] are accessible [and can be] […] referenced […]" individually. Though this Collina et al.'s idea is well suitable it is decided as out of scope of this thesis, but obviously lefts room for future development, can be seen as an outlook for suitable extensions of this thesis' approach.
Following the ideas of Collina et al. "[t]his approach enables application developers to leverage their existings [sic] skills (HTML, JavaScript, PHP, Ruby, Python) to build new ways to interact with objects […]". Extending this idea there is in short hand a possibility to see a big chance for former non-hardware programmers to join this area and develop user friendly applications, based on their application engineering skills also. Which will lead to smart applications based on real world hardware as IoT's key principle is to interconnect real world physical objects, which was already introduced before. But additionally, the real-world objects get smart, which enables them to interact with or sense their environment. This fundamental knowledge gain brings reason to the globally observed growth of IoT applications and their underlying business use cases. As one of them, agricultural viniculture is therefore obviously one of the business use cases that justify audience to research in solutions to apply IoT to it.
Obtaining back to the left-over layer, which is the cloud layer, there are aspects of provisioning, execution, communication and management of the IoT application(s) (Wortmann and Flüchter, 2015). Which itself leads to another abstraction. The abstraction is meant in terms of the before abstracted hardware, which can be used by former web-programmers. So, the abstraction of "serverless" computing is another step forward in cloud computing, which "[allocates] no resources […] or [will not charge] […] until a function is called. It's like the difference between a rental car and a taxi: you will be charged for the rental car even if you park it for a week, unlike a taxi" (Eivy, 2017). As a key benefit for which brings the approach of cloud computing is the possibility for any winegrower to use "Software as a Service (SaaS) […] The broadest definition would encompass any on demand software, including those that run software locally but control use via remote software licensing" (Armbrust *et al.*, 2010). Winegrowers could rent a cloud-based software that processes, enriches, stores, and displays all the data collected in their vineyard. Cloud based software is able to push notifications of any interest, a

winegrower could specify abbreviated from his vineyard data. This situation comes with opportunities that could be named as alert notifications to the winegrower's mobile phone for urgent threats, happening in his vineyard. Or scheduled reports that interpret all the data collected to even make suggestions for fertilization, suggested amount of applied irrigation or time to bring out plant protection products.

2.3 Internet of Things in Agriculture

The picture drawn before about need, realization of soft- and hardware development, and concurrent uprising cloud infrastructures and IoT use case extension should be proof of the relevance and potential of IoT in general. Covering those aspects there should be also enough reason for trying to use IoT to support agricultural production, precisely as in this thesis to support current practices of viniculture. Linking technology and traditional production methods to display weaknesses and reduce risks by providing forecasts, in-time and in-place monitoring and historical data aggregation there could be a also a noticeable monetary benefit for winegrowers. Benefits can be derived by aspects of value creation brought by IoT. Providing an example Wortmann & Flüchter explained the value creation as "[…] when a bin is enriched with IoT technology it may moreover measure and monitor its own weight, thus detect levels of low stock and offer an automatic replenishment service." Obtaining Wortmann & Flüchters idea on viniculture the according example could be a grapevine enriched by an IoT device, that measures plant health data and micro climate data in the plants close surrounding. Taking the chosen simple example of soil moisture measurement, the value creation could be seen in opportunities to link collected soil moisture data through a cloud API to be used by a computer-controlled or even also IoT-controlled irrigation system. Importance of vineyard watering regarding aspects of quality and market value is described by (N-düngebedarfs *et al.*, 2017) and (Koundouras *et al.*, 2006). This value creation must be questioned by "which [use of] IoT technologies […] as a core element of value creation and as a source of competitive advantage" can fit the business case of viniculture (Wortmann and Flüchter, 2015). To answer this question there are all aspects described here beforehand, additionally there are other aspects to be named as winegrower's acceptance of technology, customers' acceptance or interest in technology-based viniculture.

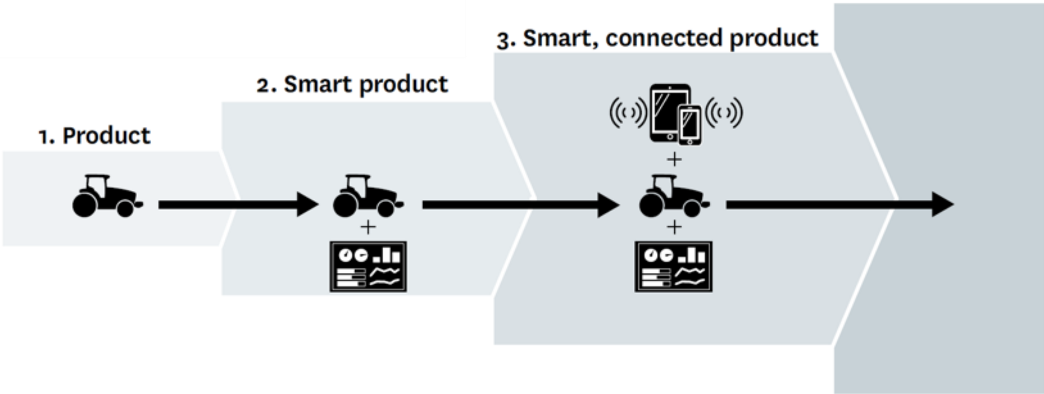


Figure 2. 1 Redefining Industry Boundaries (Part 1) (Porter and Heppelmann, 2014)

There could even be something like a data graph widget that displays information about a specific vine, which could be placed inside an online shop where vine products are sold. It could be even interactive to show graphs or to switch to a map view with geo position data to present the location the vine was grown at.
Linking multiple systems that use IoT or even just computer-based system controlling, an unforeseeable number of possibilities can arise to support all kinds of businesses.

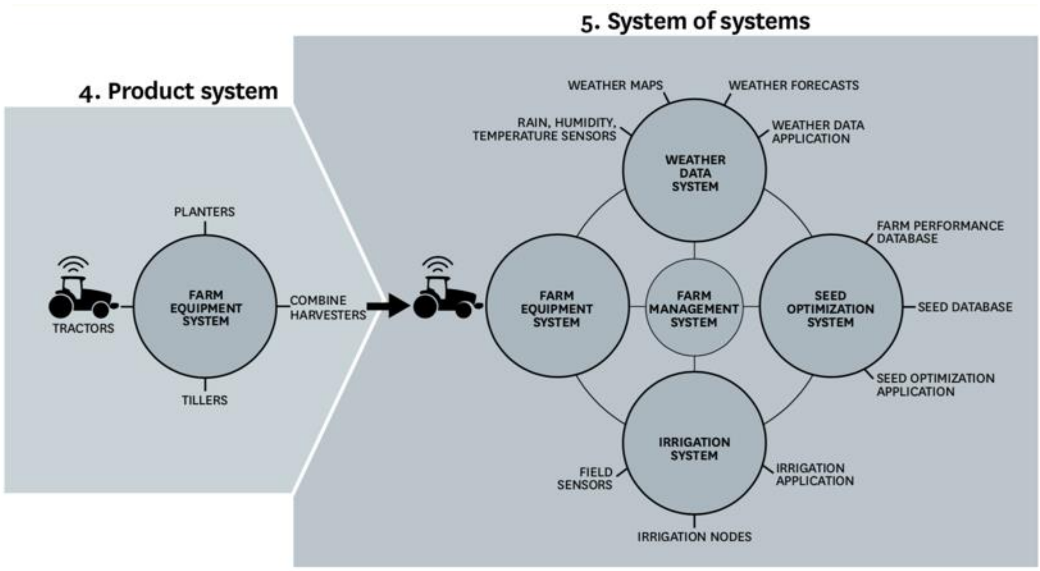


Figure 2. 2 Redefining Industry Boundaries (Part 2) (Porter and Heppelmann, 2014)

Having found this link viniculture could surely be able to also benefit from this kind of value creation brought by the use of IoT and building "systems of systems" as (Porter and Heppelmann, 2014, p. 13) described. Systems of systems is a construction of multiple individual systems that together create a new product. Starting with a single product to extend it to a smart product is the first step in product evolution towards future IoT systems. Smart products lead to smart, connected products that enable to use smart devices to monitor or even control those products. Once starting to build a system, another or multiple other smart products join the first smart product to form a system that is linked inside, which can be seen in step 4 in. Each system itself combines and needs all the containing systems to work together. Which will raise winegrowers into data suppliers to online shops that ask for vine-product growth data to promote the products. This could be handled by an API that is already planned as a core part of this thesis's concept. Nevertheless, this kind of API endpoint must be added to the implemented API in the future. More detail about the implemented API will follow up in chapter 4. To describe the concept of systems of systems Figure 2. 1 and Figure 2. 2 can be consulted, which display such an ecosystem. They contain an example of a tractor manufacturing company that now not only produces tractors but even have to produce all kinds of corresponding products or deliver a possibility to link them with third party products. Or they see competitive threads of suppliers which will take part in their product margin. “Multiple products connect to many other types of products and often also to external data sources. An array of types of farm equipment are connected to one another, and to geolocation data, to coordinate and optimize the farm system. For example, automated tillers inject nitrogen fertilizer at precise depths and intervals, and seeders follow, placing corn seeds directly in the fertilized soil“ (Porter and Heppelmann, 2014). Seeing all those linked systems, building ecosystems, there is the important aspect of weather data. As Figure 2. 2 illustrates there are several abbreviations that can be taken from a “weather data system”. This data can be analyzed and processed in combination with all the other data collected from other connected systems. The tractor can collect its geo position while driving through the vineyard. Other farm equipment like the automated irrigation system has data about how much water was supplied in a certain time. Taking all the data from those data sources together there can be algorithms developed that give advices or suggestions to help winegrowers to act preventive or in case of an urgent threat, to act in the right way.

Combining the linking of systems of systems and opportunities from digital production data integration, there are also competitive challenges, which can be overcome by product differentiation. As other winegrowers could also adopt the full stack of IoT production technology to benefit from enhancing the effectiveness and reducing threads to the plant growing and using individual grapevine data to promote their product more effectively. Winegrowers could even spread the word in social media or promote their products by embedding widgets, as described earlier, into online shops or even share to gain customer relationship and customer to product identification. Additionally, winegrowers could place “QR” codes onto their product labels to link to a special product page designed to promote and display all described data about the specific vine a customer is interested in. To be precise “Quick Response Code, widely known as “QR” code, is a two-dimensional digital image that can be easily scanned by any mobile device's camera. [“QR” codes have] “data embedded in the code“ that can contain URLs to a website which could contain the above described product data of a specific vine“ (Cata, Patel and Sakaguchi, 2013). Gaining more customer to product interaction will make winegrowers more competitive. This “approach with an Interaction Intense communication, can lead to best results of using QR code for marketing purposes“ (Cata, Patel and Sakaguchi, 2013). Customers could choose a vine they’d like to purchase, not only based on a special location, but also based on special growing conditions a vine is based on. This opportunity for customers to choose vines from completely new criteria will on the other hand force winegrowers to compete better and could force them to also produce IoT based and supported.

Having covered topics from a higher point of view and of the realm and domain this thesis is about, there is additional need for some insights in technical details of hardware elements. Those details cover among others, aspects of wireless data transmission, power efficiency and which hardware to be used best, that follow in chapters 3 and 4.
Starting with wireless data transmission there is one most known technology to consumers, which is Wireless Fidelity or "WiFi". “WiFi is the term used to describe a class of certified wireless networking products conforming to an industry standard designated by the Institute of Electrical and Electronics Engineers (IEEE) as “802.11b“[…]“ (Al-Alawi, 2006).
Now clarified the term, restrictions follow close, as “[b]ecause it operates in unlicensed frequency bands, anyone can set up a WiFi network and cover an area of typically 100-500 feet [approximately 30-150 meters] with high speed wireless access to a LAN and hence to the Internet.“ (Al-Alawi, 2006) Both the fact that WiFi is working in an unlicensed frequency and that everyone can setup and run it there are limitations important to the use of IoT. WiFi’s power consumption is too high in terms of IoT for low power devices to operate for a long time, e.g. years without recharge. Also, the transmission range covered is not suitable for many use cases of IoT. Thinking of WiFi range there could even be personal experiences supporting thoughts here. Trying to cover even a private estate with stable, high speed and full range, WiFi is often tricky. Transferring this experience to an industrial thinking of IoT application there is only way to use WiFi in small ranges e.g. in small storages. Thinking of large storage there are big connectivity issues. As personal experience made with a client from international logistics business, based in Worms Germany, they got problems to setup requested mobile applications depending on local WiFi network connection, that were to be used in large storage halls. To give a dimension of large in this relation, the storage halls cover areas of soccer fields. Goods stored there produce interferences that cause standard WiFi connection to fail. “When using a WiFi network the frequency specification used is 802.11b, which is the same frequency that many cellular phones, blue tooth, microwave ovens operate on. So it is in the hands of the WiFi users to choose an access point that is not close to the one of the devices mentioned, because an interruption will occur and this may cause a loss of signal or it may deteriorate the connection“ (Al-Alawi, 2006).
Comparing those circumstances with conditions given in a vineyard, there is at least the similarity of range that must be covered. Additionally, there are other environmental interferences that are not to be estimated clearly. As open space could always underlie changed conditions brought by natural environment, weather conditions or artificial changes. There is another wireless data transmission

technology upcoming popularity, which “has received a lot of attention […] from network operators and solution providers“ (Adelantado *et al.*, 2017). This type of network is one of “Low-Power Wide Area Networking (LPWAN) [and its] technology offers long-range communication, which enables new types of services. […] LoRaWAN is arguable the most adopted. It promises ubiquitous connectivity in outdoor IoT applications, while keeping network structures, and management, simple“ (Adelantado *et al.*, 2017). Simplicity and already receiving well acceptance and support, this is the technology to go for this thesis. Additionally, “LoRaWAN is a network stack rooted in the LoRa physical layer. LoRaWAN features a raw maximum data rate of 27 kbps (50 kbps when using FSK instead of LoRa), and claims that a single gateway can collect data from thousands of nodes deployed kilometers away. These capabilities have really resonated with some solution providers and network operators, who have created a large momentum behind LoRaWAN to the point that it is sometimes touted as the connectivity enabler for any IoT use case“ (Adelantado *et al.*, 2017).

3 Hardware Prototype

Hardware prototyping and use of hardware is a differentiated area that can be covered in multiple aspects and ways. Therefore this chapter focuses on some aspects derived from “Quality Function Deployment (QFD)“ (Akao and Mazur, 2003) as introduced in section 1.3.
The first operation to define the hardware is the thorough description of the requirements to the product. At first the requirements to the product have to be thoroughly defined. Those requirements are described in section 3.1. Implementation and prototype development follow in section 3.3, which is concluded in consequent section describing programming and finally prototype assembly. These sequential steps may help to elaborate through the hardware prototype section.

3.1 Scenario

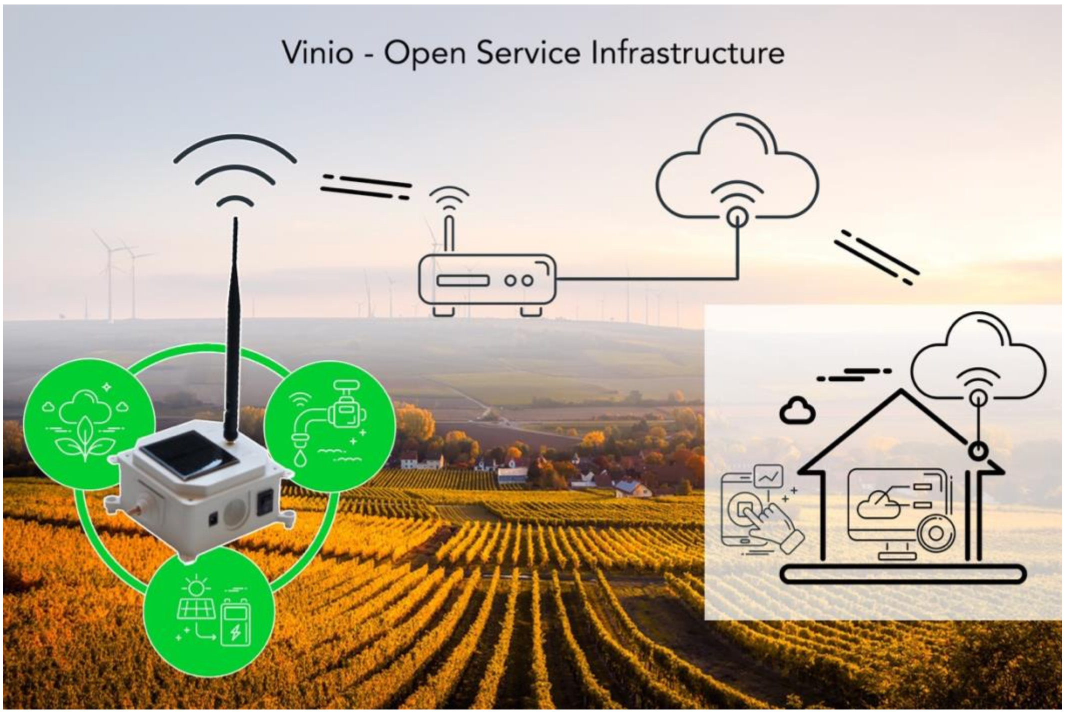


Figure 3. 1 Vinio – Open Service Infrastructure Schematic (own illustration)

The main scenario as a starting point and the main guideline to elaborate to is visualized in Figure 3. 1. On the left side there is a picture of the prototype placed inside a green circle of icons to symbolize the use of solar power, be able to connect to a water irrigation system and an icon for plant growth. Plant growth is the main sensing target as this thesis’ target is to enable winegrowers monitor their vineyard smart with IoT technology. The prototype then transmits data to a gateway, which must be connected to the internet. It must be connected by WAN either or connected by cellular. The data is then transmitted to TTN and from there to the Vinio Service Layer. The winegrower is then able to monitor and visualize his collected data in his vinery or mobile on his smartphone or cellular tablet. A winegrower is then able to decide which terroir is most urgent to be cared for this day he is viewing his live data. This enables a winegrower to improve his efficiency by only caring for the terroir that needs most care at a specific day. The reason terroir is mentioned in this scenario is due to the possibility to place enough prototype devices in a vineyard to monitor with accuracy even on terroir level. There could be even more devices placed. Even each grapevine could be monitored and made smart. Nevertheless, there is currently no need found to monitor on such a high accuracy or low level. Describing the scenario there are some limitations brought by nature, climate and technology. The natural environment of slopes brings difficulties for radio transmission. Topology and afforestation limit the radio transmission range due to radio shadow caused by rocks and trees. As each vineyard is different every installation of a smart vineyard must be done manually in the first place. Once the radio coverage is setup sufficiently additional data measuring devices can be installed easily in the vineyard. Natural limitations come into account as the used hardware can only withstand a defined range of temperature and air humidity to operate reliable. Additionally, the hardware and sensors could be damaged due to temperatures out of operation range.

3.2 Requirements Definition

Requirements are derived from the sources of use case domain, customer perspective and customer needs. The applicable areas of QFD - regarding the scope of this thesis which are product development aspects - are described subsequently, as mentioned in section 1.3. QFD core areas to consider during product development are the fields of quality, technology, cost and reliability.

The devices are placed in vineyards which bring with itself requirements based on physical aspects: the need for device installation, device usability, function, features and cost. Since this thesis does not deal with the industrialization of the device, as e.g. deployment of assembly layout and production factors, the cost section is just named to be complete, and is considered as to be out of scope. Nevertheless, aspects from QFD, like quality, technology and reliability are contained elements.

Requirements can be aggregated into multiple categories that group individual requirements in an intuitive way, derived from the use case and application field of the devices. There are various aspects to be covered when evolving requirements. These can be split into two major parts. Regulatory requirements decreed by law beforehand and followed by non-regulatory requirements. Regulatory requirements are described in section 3.2.1. Subsequently non-regulatory aspects follow in section 3.2.2.

3.2.1 Regulatory Requirements

Regulatory aspects must be considered, as the devices are electrical devices and those underlie regulations and requirements brought by law. Electrical devices sold in the European Union (EU) must satisfy a number of qualities, reliability and safety tests, which are then be marked with the according test labels, like the CE marking. Those are "safety, health, and environmental protection requirements" (*Manufacturers - European Commission*, 2018). There are four aspects to be covered when taking the conformity assessment to gain the CE marking. Those are first that "[t]he conformity of a product is assessed before it is placed on the market. [Second i]t needs to be demonstrated that all legislative requirements are met. [Third i]t includes testing, inspection and certification. [Fourth t]he procedure for each product is specified in the applicable product legislation." (*Conformity assessment - European Commission*, 2018). Additionally, there is another certificate to be satisfied, which is the "Restriction of the use of certain hazardous substances (RoHS)", which includes „Directive 2011/65/EU of the European Parliament and of the Council of 8 June 2011 on the restriction of the use of certain hazardous substances in electrical and electronic equipment" (*Restriction of the use of certain hazardous substances (RoHS) - European Commission*, 2018).

The realization of regulatory requirements in the prototype are also out of scope of this thesis. There may also be other certificates that are required to be satisfied. Nevertheless, the certificates described are chosen to satisfy as many aspects of the product development approach possible. They shall define future work, that is required to use the final prototype as a real-world sellable product. In general, there may be requirements, that follow regulations based on military or industrial normative.

3.2.2 Non-Regulatory Requirements

Though regulatory requirements are necessary for real world electronic end-products, there are basic technical and physical requirements on the constructive and layout aspects to be realized in prototypes. Those requirements include the device to be weatherproof and in particular to have a waterproof housing. This requirement is derived from the pure fact, that devices must withstand open space weather conditions that come with prevailing weather conditions in vineyards. Also, vineyards are not connected to electrical power grid, which formulate the need of a power supply not depending on a power grid. As referring to the intended use by winemakers and respecting that single device price must be kept as low as possible, there is need for low or even no maintenance through lifetime of the device. A need for maintenance would increase the attributable cost per device as seen from winegrower's perspective. In terms of eco-compatibility use of energy, there are requirements of energy efficiency and low power consumption.

Primarily customer expectation and customer view will decide upon the subsequent success of the development. From the perspective of the intended use, for an installation of the device there must be an easy way to register it and to add new and additional devices. A device setup process could be autonomous. Details for the installation requirements will be described in more detail in the following section. Nevertheless, those requirements come from customer and use perspective. Customer perspective is driver of usability efforts. Naming these, the main points are "plug and play" functionality, simple handling and no maintenance for parts. Functions and features include potentially universal sensor connectivity, suited types of sensors and range of operation. Range of operation means the distance between gateway and device that must be reliable in the whole vineyard.

3.2.3 Outlook for industrial realization

On aspects about industrial realization there are things to cover that go beyond this thesis' scope but are inevitable to be discussed as an outlook perspective. First there is the aspect of market review. In market review there must be considered things like market size, market segmentation and market strategy. This leads to an assessment of opportunities, challenges and weaknesses to position the final product in an appropriate way. Therefore, the competitors are to be observed and assessed too. Some competitors e.g. TracoVino started up from a similar only vineyard IoT product but realized to expand with their sponsor Intel into logistics businesses where market review obviously brought some key benefits. Their product is fully compatible to be used in logistics businesses use cases as out-of-the-box implementation. Public indication for a relative market size has not been discovered in research done.

As a precondition, there is intellectual property to be secured upon copying. This is possible by registering a trademark and patenting by e.g. fining of use and construction. One product extension is to offer options to product configuration or even product customization and offering accessories to make the product more useful, more dedicated to a specific use case or differ price ranges from equipment configuration. Decision must be taken to have a single or reuse product and a certain and suitable level of the ability to recycle parts or even the whole product. Those aspects are derived from product life cycle. The product life cycle concept demands to define the determination of the product life cycle. Nevertheless, before the end of life cycle there are decisions to be made upon proprietary manufacturing versus contracted manufacturing. Both bring benefits and obstructions. In the next section there are aspects of prototype development covered and experienced while elaborating all methodological work packages.

3.3 Prototype Development

The prototype development follows the previously described method from section 1.3. This includes, to take customers perspective into account, derived from QFD.

In this case customers are winegrowers. Winegrowers are interested in a solution that include IoT devices with multiple features. Those features must be combined with the requirements defined before, in section 3.1. Features to be named are long term operation, eco-design, preventive care (which includes fertilization suggestions) and automated irrigation based on IoT collected data. To really benefit from IoT application a winegrower could want to receive simple and correct advice for actions needed at his vineyard level. Seeing the needs in combination a winegrower benefits from work reduction, receiving good value for money and maintenance-free devices. The customer perspective named here is based on mind mapping and personal survey of winegrowers (e.g. Weingut WICK, Zellertal/Pfalz).

There could also be additional and future work done to extend the requirements formulated in this pathfinding thesis for innovative IoT use. However, the listed requirements are the foundation to realize the prototype development described in this thesis. As a consequence, there is a must to create a simple and (e)valuable first version, which can fit the described use case.

Extended features or requirements can be added later to an advanced prototype, which is also a key aspect of a prototype development. The prototype shall be a universal foundation to continue from its status in the future. The collaboration has been approved within a first voluntary network of stakeholders. From this, a more powerful interest group or development base could be derived in near future.

As a fundamental concept, the prototype is developed in a modular design, concerning firstly the components. Creating modular electronic products makes future variation easier. This allows to change sensors to be used as requirements of winegrowers could vary, based on region, size of vineyard or other circumstances. Also, industrial production based on modular products is more efficient as individual modules can be produced separately and just assembled to the final product before shipping. This also enables customization and flexibility of the devices and even opens them up for possible other use cases not thought of here. “Modularization refers to the scheme by which interfaces shared among components in a given product architecture are specified and standardized to allow for greater reusability and commonality sharing of components among product families. It is also a new product development (NPD) strategy for increasing product variety and customization. When interfaces of components or modules within a system becomes standardized, outsourcing decisions can be made accordingly with respect to a firm’s long-term strategic planning of its NPD, manufacturing and supply chain management activities. “ (Mikkola, 2000).

In order to realize the product design element of modularization, there is one option among others to use standardized components. This can be done by using commodity components. Using commodity components brings multiple benefits. At first there is a reduction of complexity as commodity products are useable - as they are delivered - to be assembled. Secondly, products based on commodity components are already partly tested and certified products. This could reduce certification process efforts.

Following the modularization approach, the main core and embedded software unit of the prototype is realized by an Arduino®. “There are multiple hardware choices available, however the Arduino is currently the most flexible and easy-to-use hardware and embedded software [platform], with low cost, easy communication […]“ (Alves *et al.*, 2013).
Following this assessment, the choice for the prototypes main processing unit is therefore Arduino. In particular an Arduino Pro Mini 3V/8MHz is used, equipped with an ATmega328 microcontroller. Flash Memory is in size of “32KB of which 2 KB used by bootloader” (Arduino, 2016). Whereas the latest version of the prototypes embedded software is filling up programmable flash memory in size of 23728 Bytes, which is 77% of available space, according to a measurement of the Arduino Integrated Development Environment (IDE) software.

In order to decide which sensors can be used based on the decision to use an Arduino Pro Mini, the power supply for sensors has to be considered. The Arduino’s “DC Current per I/O Pin“ is specified as 40 mA according to the official technical specifications on the Arduino website (Arduino, 2016). There are some weaknesses of Arduino, that must be described in short, though. Its clock error comes into account on precise live data measurement. But vineyard data is not in the need for such time accuracy. Therefore “[t]his behaviour [sic] was already expected due to the Arduino’s clock accuracy error of 0.2%” (Alves *et al.*, 2013).

In terms of energy, there are requirements of energy efficiency and low power consumption. The devices must work for multiple years out in the fields without a single artificial recharge cycle. To reach this requirement it is decided to use two components that ensure power supply over the whole device lifetime. One component is a battery. The battery is designed lightweight and small. To even reduce power consumption and receive gain in energy efficiency there is a library that enables sleep a mode that reduces Arduino’s power consumption. There are two types of batteries which have been tested. At first a Lithium Polymer (LiPo) battery and second a Nickel Metal Hydrate (NiMH) battery. Both bring pretty different requirements, benefits and weaknesses with them. A LiPo battery has a higher power capacity on smaller volume. But it needs a Printed Circuit Board (PCB) with balancer capability to charge the battery cells evenly. This brings extra need for a specific PCB component, which is extra gain in complexity. Nevertheless, this PCB component is one of the before described commodity components. It’s the TP4056. “The TP4056 is a complete constant-current/constant-voltage linear charger for single cell lithium-ion batteries“ (Corp, 2015). Its standard use case is to charge cell phone batteries from USB 5V power supply. The “Operating Ambient Temperature Range: [is from] -40 °C [up to] ~85°C” (Corp, 2015). Additionally, it has a second power input and the electrical load is separated from battery charge load. This makes power supply for the Arduino even. Both solar power and battery power are leveled and supplied evenly. The PCB balancer component can withstand voltages up to 12V and amperage of up to 1A input. This technical property is specified by the manufacturer and therefore it can be used to either charge a LiPo or a NiMH battery, although NiMH batteries do not need balanced charging.

As the Arduino is the main processing unit inside the IoT prototype device it is responsible to send collected and measured data up to the cloud layer. Data transmission is done by wireless transmission with LoRa technology, as described in section 2.2. The according hardware component for data transmission is the Hope RFM95W version 1.2 transceiver. It is a "Low Power Long Range Transceiver Module […] LoRa™ Modem" as described by the manufacturer. (Hope Microelectronics Co., 2014) It is connected to Arduino's pins as follows. It requires „SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support [Serial Peripheral Interface (SPI)] communication, which, although provided by the underlying hardware, is not currently included in the Arduino language" (Arduino, 2016). "Serial Peripheral Interface (SPI) is a synchronous serial data protocol used by microcontrollers for communicating with one or more peripheral devices quickly over short distances" (Arduino, 2018). Nevertheless, there is a library that implements the needed hardware communication between Arduino and the transceiver module. This library is the Arduino-LMIC library which is a modified version of the "IBM LMIC (LoraMAC-in-C) library" published on GitHub (Kooijman, 2018). Derived needs for the embedded software follow in section 3.4.

Evaluation as a key principle is described in section 1.3, and there is a further essential aspect to be covered according to wireless data transmission: this is namely the wireless data transmission range. Ranges of LoRa based data transmissions span distances of more than 200 km, but only in specific test setups with direct sight between the gateway and the IoT node or device (*TTN Mapper*, 2018). The prototype developed in this thesis reaches ranges of multiple kilometers on direct sight and could reach the 200 km of the above described test, as the used hardware is the same.
The chosen test set-up scenario, as Figure 4 illustrates, was a repeated, successful test of 2.3 km distance on direct sight in hilly terrain. This terrain is chosen to succeed in an even more sophisticated environment, compared to the use in a vineyard. The screenshot used shows a map with two map markers in between the data transmission was successfully recorded, which was successful even behind trees but on direct sight without a hill blocking the signals. Radio is shielded by hills and there is no connectivity nor data transmission possible. Measurement technique is described in section 1.3. LoRa signal results on longest range were reached on frequency of 868.5 MHz, SF 7, RSSI of -107.0 and SNR of -5.0. Signals were captured at 1 m above ground level. Lower positioning of the IoT device resulted in signal loss. Results can be viewed publicly on the website https://ttnmapper.org.

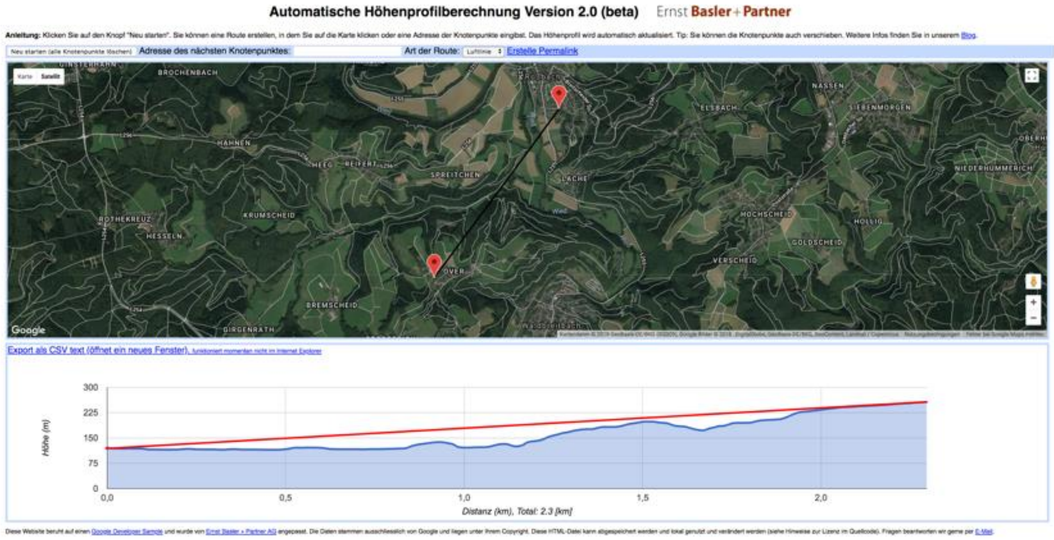


Figure 3. 2 Automatic Terrain Relief Calculation (Ernst Basler + Partner AG, 2018)

In terms of usability for winegrowers there is need for an easy installation process to register devices or add new and additional devices. A device setup process could be autonomous, which requires a pre-configured and pre-registered device state from factory. This restriction is reasoned by the dependency to TTN and the necessity to register each device at TTN and hardcode the according IDs into the embedded software on each Arduino prototype device. This could be seen as a final manufacturing process right before the end of production line testing. The registration process to the front-end is described in chapter 4.

Development tasks to cover:

- Layout and dimensions of housing.
- Status light indicator.
- Interfaces layout antenna, power supply plug, on/off switch

A number of those development tasks are elaborated in DSR cycle number one. This leads to the documentation of DSR cycle one as follows.

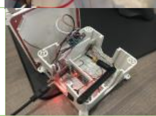
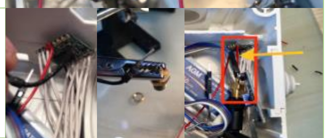
Phase	Desired Target	Reached Target	Problems	Changes / Extensions / Optimizations derived	Time needed	Photo
Phase 1	Assemble Case	Case assembled	Case was uneven inside, so needed grinding with hand milling machine.	Need case mount buttons to reach reset function and add on/off switch on board inside. Add Power input to charge the node.		
Phase 2	Add Power input	Power Input mounted.	Transceiver connections are not reliable, though they are soldered.	Researched internet for pcb plan to support better connectivity for transceiver module. And ordered some.		
Phase 3	Solder transceiver wiring to special build io pcb shield for RFM95 transceiver	Transceiver sufficiently soldered to breakout PCB.	PCB connections are still not reliable enough, though they are soldered to the io shield.	Solution didn't add extra benefit. So use of PCB was discontinued.	1h	
Phase 4	Solder transceiver wires to arduino board from backside through vertical mounting wall.	All wires were successfully, carefully and precisely soldered.	No major problem occured.	White cables could be shorter for future assembly.	2h	
Phase 5	Sufficient soldered wires.	All wires were successfully, carefully and precisely soldered.	Insulation of used solid wire melt on wire tip due to much heat from soldering iron, during soldering.		2h	
Phase 6	Device is programmable and successfully sends data to TTN Gateway.	Device is programmable and successfully sends data to TTN Gateway.	Transceiver didn't stick to the wall applyed with hot glue.	Connect sensor module temporarily to arduino, due to send test data.		
Phase 7			Battery plug made a short. So LiPo juged.	Insulation with heat shrinking tubing on plus wire.	15m	
Phase 8	Potential customers shall be able to easily change the battery without need to have to worry about to break things.		Antenna cable often breaks due to opening and closing the case to change things or work on assembly.	Soldered a solid wire instead of the before used stock wire. **Left picture:** broken wire **Center picture:** removed weak/old wire **Right picture:** new strong and solid wire	15m	
Phase 9	Device is fully assembled.	Device is fully assembled, is fully working, transmits data to ttn.				
Phase 10	Reach appropiate distance range.	Longest distance reached without direct sight (behind trees) was 1.15 km	Range test suddenly was interupted by unknown reason.		2h	
Phase 11	Mount Buttons	Buttons mounted				

Table 3. 1 Work Log DSR Cycle 1 – Assembly Techniques Research (own table)

3.4 Embedded Software

Embedded software is “[…] in everything from telephones and pagers to systems for medical diagnostics, climate control, and manufacturing. Its main task is to engage the physical world, interacting directly with sensors and actuators“ (Lee, 2000). The embedded software is written in C as Arduino is programmed in C.

3.4.1 Libraries

There are 3 libraries used that are required for the thesis’ prototype to operate in the desired way. First DHT library to use humidity and temperature sensor. The DHT library is “an Arduino library for the DHT series of low cost temperature/humidity sensors” (adafruit, 2018). This sensor simply measures air humidity and air temperature. Secondly a so called Low-Power library. The low power library is a “[l]ightweight low power library for Arduino“ (Rocketscream, 2018). Low power means, that the Arduino goes into a kind of sleep mode where it does not power any runtime functions or sensors. The only runtime function to still be executed is the loop function. Between every measurement and every data transmission Arduino saves energy by sleeping and waiting for the next scheduled activity. Third one is CayenneLPP to encode data to be transmitted through LoRa. “This is an Arduino Library for Arduino Compatible with Cayenne Low Power Payload“ (Stokking, 2018). Sending data over TTN is limited due to the nature of LoRaWANs low bandwidth. The maximum number of bytes to be send over LoRaWAN is 51 bytes. To bring that into perspective, all kinds of text must be encoded into numbers. Unicode comes with 128000 characters that must be encoded to numbers before such text can be send over LoRaWAN. Sticking to Unicode each character needs 3 bytes. In an early implemented example node there is a need to define the collected data send by the moisture sensor. To define the moisture sensor in a human readable way one could think of an identification such as a single character for defining a sensor which could be just “s”. The “s” then could be followed by an underscore to separate the generic identification as sensor data to the kind of sensor used. This cloud be something like the sensors name, e.g. “moisture”. One will then end up with an identification of the moisture sensor as “s_moisture”. This will itself have 10 characters which results in a need for 30 bytes to be send only to identify 3 bytes of measured data. The used moisture sensor has a range of values between 0 and 255 which is a 3-digit number that results in a need of 1 byte per digit. To send measured data from the sensor there must be 3 bytes.

This comes into account when trying to distinguish between numeric values that are sent by one of the IoT generic devices. There must be a way to map numeric values, sent by this node, to the according sensor. Reason to need a mapping is to be able to interpret the measured values of a sensor when processing them inside the services layer. The same concept is used in CayenneLPP which is the result of an evolution step in the prototype development. CayenneLPP defines various types of data to be encoded for LoRa transmission. Those types have data identifiers like “analog_in_2”. Using those data encodings there is no need to think of custom data encoding like the first part of this paragraph describes. Compatible data that justifies the specifications of CayenneLPP data types can be used directly from sensor measurements.

3.4.2 Components

The embedded software is based on an Over The Air Activation (OTAA) as defined by TTN for Arduino by Thomas Telkamp and Matthijs Kooijman, 2015. It requires some TTN data transmission protocol specific configurations. These are “APPEUI” which is a little endian format octal value to identify the TTN app and to identify the transmitted data that belongs to it. A “DEVEUI” which is also a little endian format octal value to identify the individual device that is sending data to and receiving data from TTN. The last TTN related configuration is the “APPKEY” which will provide access right to the previously configured application. Additionally, there are multiple functions that realize the features of the exemplary used sensors.

3.4.3 Software Logic

Logic comes into perspective once multiple single data measurements are connected, cumulated or in other aspects related or linked to each other. This is an explored aspect and is also a fundamentally origin of this thesis’ prototyping process. The embedded software is reduced to a minimum of high-level logic. The embedded software cares for protocol implementation as described in section 3.4.1 based on used libraries. Additionally, the embedded software collects data from used sensors. Thinking of the soil moisture sensor there is a fundamental aspect of corrosion caused by electrolysis. As the soil moisture sensor uses electrical power to measure the resistance of electrical current trough the soil it is threatened by before named corrosion. “In an electrolysis cell, also called electrolyzer, an external voltage (or current) is applied to the electrodes in order to carry out an electrolysis” (Landolt, 2007). As described by Landolt, 2007 the electro chemical environment of the soil moisture sensor is similar to the electrolysis cell. A basic water electrolysis system is shown in Figure 1. 3. This is the typical environment the prototype devices are placed in.

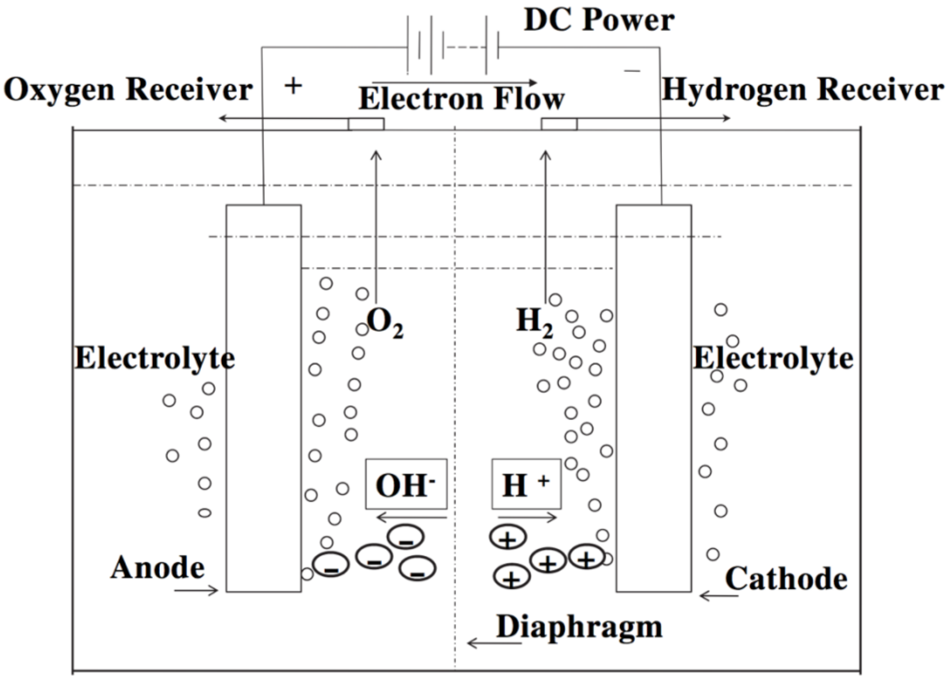


Figure 1. 3 A schematic illustration of a basic water electrolysis system. (Landolt, 2007)

Therefore, the software logic needs to overcome the corrosion problem. This problem is addressed and solved by not connecting the voltage pin of the soil moisture sensor to the 3.3V current supply of the Arduino. Instead to connect this voltage pin to a digital pin of the Arduino to only apply current during measurement. The digital pin is set to HIGH (current flow) for only 10 milliseconds. Other logic is minimal and brought by libraries, as the DHT library has built in functions to read air moisture and air temperature. Both functions are just outsourced into their own abstracted, higher level functions to only return the final value each.

The other major logic that is built into the embedded software is the purely software-based battery voltage measurement. It is realized by the "AVR chips ability to measure the internal 1.1 volt reference" (SCOTT, 2012). To measure the battery voltage the voltage reference is set to VCC first. Then the value of the internal reference is measured. The ADC value is in a range of 0 to 1023 which is the reason to divide by 1023 to set the measured value into correlation of the maximum ADC value of 1023. Finally, the value of VCC is calculated as follows.

$$Vcc * (ADC - measurement) / 1023$$

Which is related to 1.1 volts. Therefore, solving for VCC the calculation ends up as follows.

$$Vcc = 1.1 * 1023 / ADC - measurement$$

Having covered the implemented logic of the embedded software there is need to give reason to the reduced logic of the embedded software. The general architectural design concept is to gain logic complexity in each higher layer. As the embedded software is located in the physical layer, the lowest layer it contains the lowest logic complexity and highest genericity.

3.5 Prototype Assembly

The prototype assembly is about to describe and document the steps elaborated during the prototype development process. Assembling techniques and evaluation of different methods of component manufacturing or combination are described and documented.

3.5.1 Preconditions and constraints

In industrial production layout there are fundamental approaches to improve efficiency in device manufacturing. Modular design is one of them to be named. Use of commodity parts is another one and use of same parts and standard components represent further strategies. Production line efficiency benefits from spliced elements combined into modules. Multiple modules can be assembled in different locations and asynchronously, hence with higher efficiency in separate serial manufacturing at different times independent from each other. “Modular products consist of detachable modules, which can be manufactured, assembled, and serviced separately” (Gu and Sosale, 1999). In many cases, total ownership cost of expensive equipment profits from decreased service and repair cost when the design layout is modular (e.g. Rittal-Siemens-Trumpf electric cabinet modular design “modular3”) (‘Mehrpolige Steckverbindungseinheit für Dreiphasen-Wechselstromsysteme’, 2012). These principles are applicable also for sensor measurement modules as of the core logical unit. The module construction is defined based on several criteria. First the location of the modules is an important criterion. Components that are located next to each other could be combined into a module. Secondly components which have certain relations to each other can be combined into modules. There are components which are defined as commodities in section 3.3. Some of them are e.g. the LoRa antenna or the solar panel. These types of commodities are better described as peripherals. Nevertheless, modules can be categorized from logical aspects also. Logic comes into account once there are components like the Arduino or the transceiver. Components like the transceiver can be seen as sub modules to be combined into modules as defined before. Additionally, there is the LiPo balancer PCB which itself can be seen as a module, as it is described by the manufacturer in the manual, too. In the following the modules are described and defined as 5 modules.

- Module 1: Core Unit
- Module 2: Housing or Case
- Module 3: Sensor(s)
- Module 4: Peripherals
- Module 5: Power Supply

In Figure 3. 3 Prototype Modules there are all major parts aligned according to their modules.

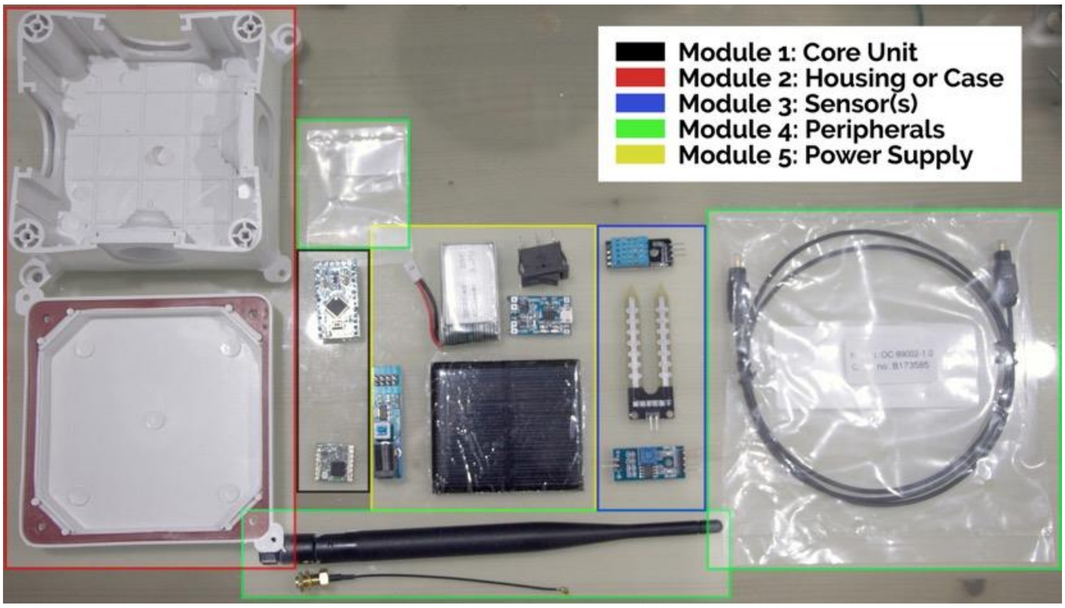


Figure 3. 3 Prototype Modules (own illustration)

Module 1 combines the Arduino as it executes the embedded software as described in section 3.4 and the transceiver module. In order to even reduce the manufacturing complexity further, the LiPo Balancer PCB and the transceiver module can be hard wired and fully soldered to the Arduino board in a single assembly step. Extending this idea all these components could be soldered to a kind of mainboard that holds these sub modules to follow the same idea as PC manufacturers decided to build a single module like the motherboard to have multiple components and sub modules combined into one module.
Module 2 is the housing component which is also kind of a module. It exists of an IP44 case “Schutzart IP44 (spritzwassergeschützt)” that is originally intended to be mounted on walls with splash water resistance needed for electrical wiring (Anlagen and Montagestellen, 2013). It is then assembled with mounts for Module 1 inside the housing. As limitations of prototyping the IP44 certification gets lost by manipulating the case physical attributes during the prototype assembly.
Module 3 is more of a category of modules than of a single module. Sensors can be assembled separately and independent from other modules. Other modules have standard connectors to connect any sensor to them. Restrictions and requirements for sensor connectivity is described in section 3.3 and is reasoned by chosen hardware specifications. Sensors come in different sizes and shapes. They can be

mounted to accessories or just wired and soldered. Each sensor can be assembled as a module or a group of sensors. This varies fundamentally from one use case to another. This thesis’ chosen use case requires a humidity, temperature and soil moisture sensor. There is a sensor called DHT11 which is a combination of a humidity and a temperature sensor and a soil moisture sensor. Both sensors can be assembled as separate modules of Module 3 type.
Examples for Module 4 are LoRa antenna and assembled sensors, solar panel or power supply. Components of Module 4 type can be manufactured separately. There is even no need to assemble them in production line except for testing purposes. Even Module 4 components are in fact large commodity components, as they can be used as final products themselves. Finally, customers will use them to install the device in the field.
As a key requirement of modularization, the separated and reduced amount of assembly steps in production is therefore fulfilled. Though final case design may vary from prototype design due to complex requirements e.g. ventilation or convection, which come into consideration during full commercialization.

3.5.2 Case Design and Assembly

In terms of case design, the dimensions are assumed to be approximately 10 cm in length and width and approximately 6 cm in height. The needed size is developed by a paper cube that is dimensioned to hold all components described as modules 1, 3 and 5 in section 3.5.1. There is an already fitting industrial standardized case. This case is based on an IP44 “Schutzart IP44 (spritzwassergeschützt)” case that is originally intended to be mounted on walls with splash water resistance needed for electrical wiring. (Anlagen and Montagestellen, 2013) In Figure 3. 4 there is the IP44 electrical case side by side to the paper cube tinkered to estimate the needed case size.

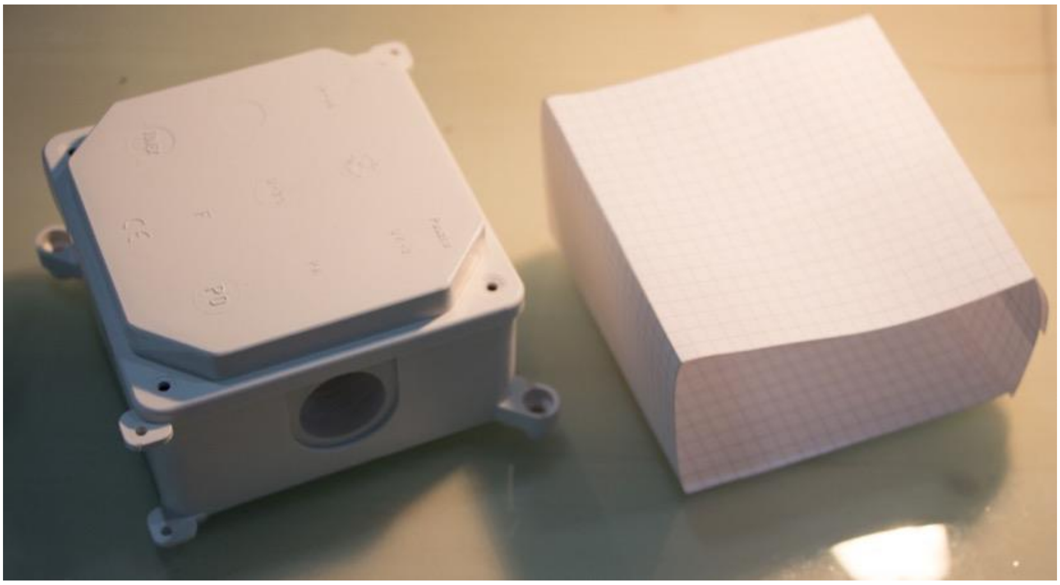


Figure 3. 4 Prototype Housing Size Evaluation (own illustration)

The case gets screw mounts glued into place to supply above ground mounting for the hardware components. The mounts lift the hardware components up to ensure enough space for wires that connect all electronics from modules 1, 4 and 5 inside the housing. In Figure 3. 5 and Figure 3. 6 PCB Mounts Top View the mounts that lift the PCB hardware plate can be seen. In Figure 3. 6 there is also a zoom that shows more detail of the PCB mount glued into place to hold the PCB above case ground. The PCB mounts are taken from an other unused spare part of a flush LAN wire wall mount. Those mounts are originally ment to hold the front plate of the LAN wire wall mount. They are cut out from the original cases.

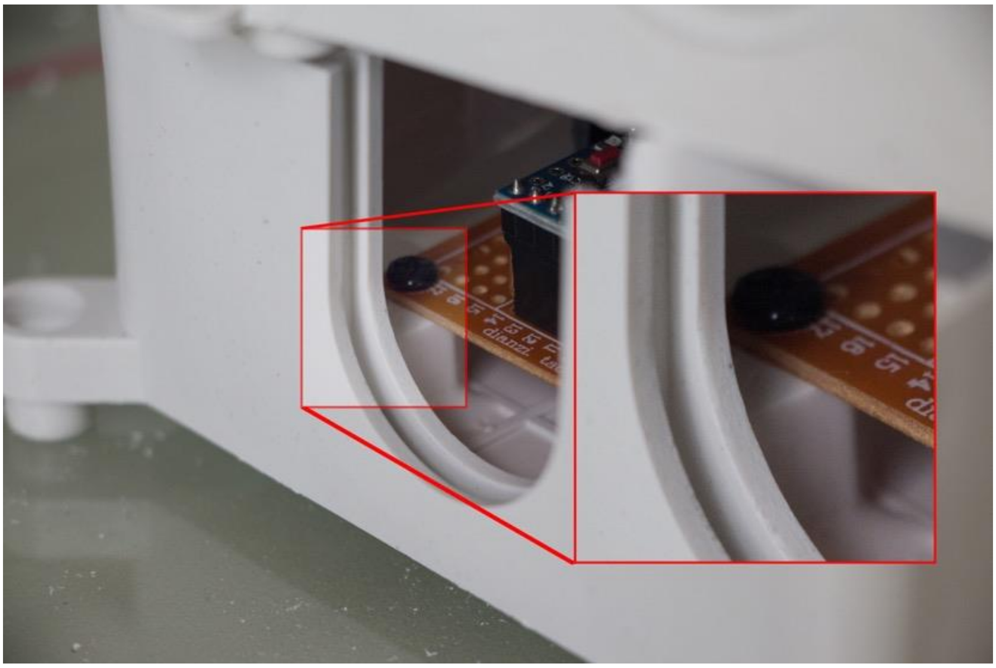


Figure 3. 5 PCB Mounts Detail View (own illustration)

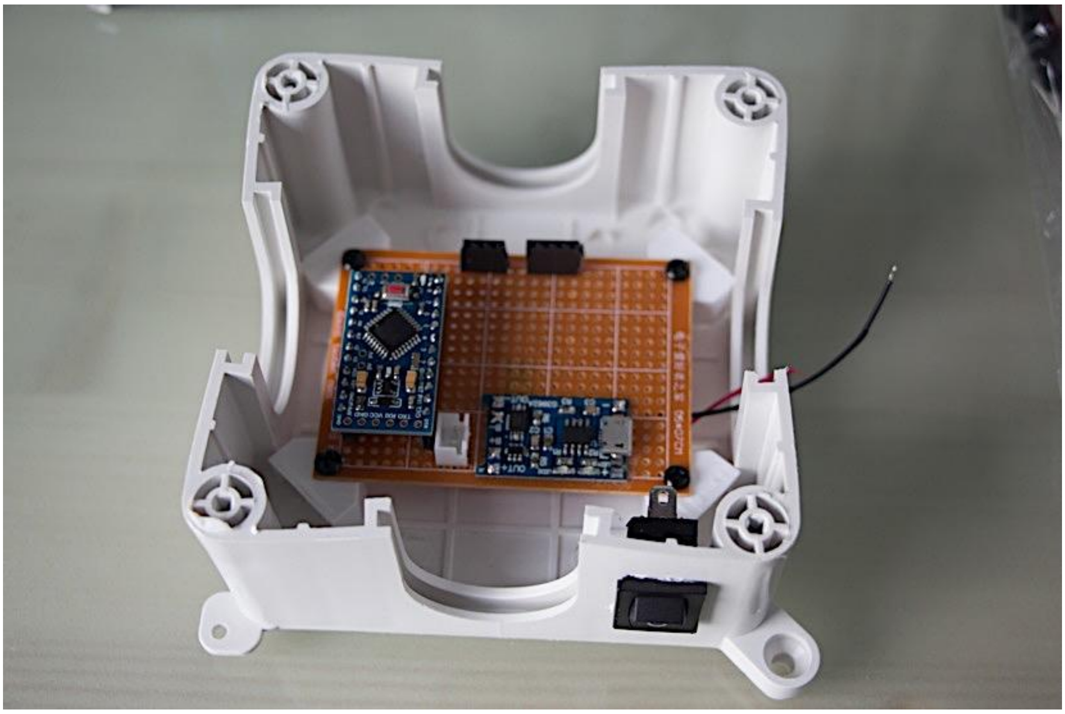


Figure 3. 6 PCB Mounts Top View (own illustration)

In Figure 3. 7 there is the glued solar panel applied flat on the case top. In Figure 3. 8 the power wire holes are glued to ensure the case top to be waterproof again.

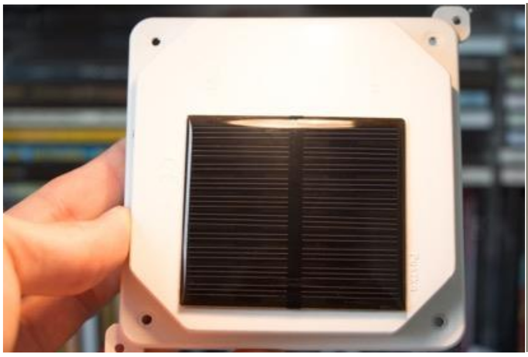


Figure 3. 7 Solar Panel Glued in Place (own illustration)

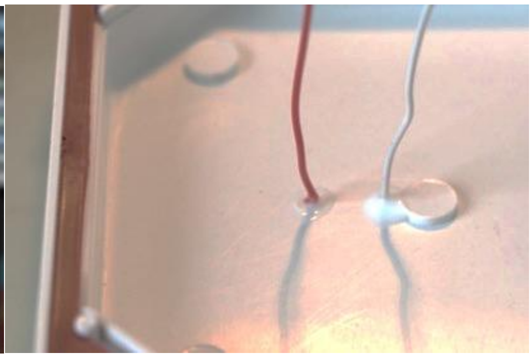


Figure 3. 8 Solar Panel Bottom View (own illustration)

There is a power switch mounted into the side of the case. This can be seen in Figure 3. 9 from outside and in Figure 3. 10 from inside without wiring. The use case requires to use the prototype in an open space and in every weather condition that is in a vineyard which causes a fairly large power switch to be chosen to bring better usability. Though this power switch is able to withstand voltages up to 230V, which is originally intended to be used in home electronics, it is suitable for the described use case to supply an easy accessibility.

Figure 3. 9 Power Switch Outside (own illustration)

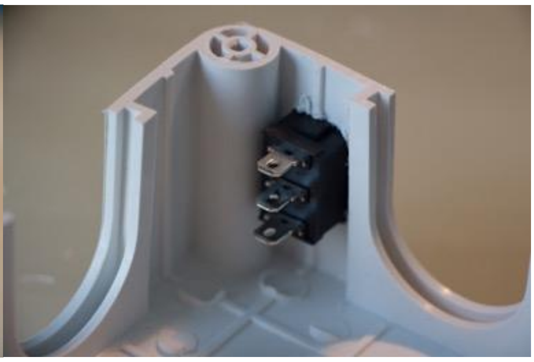


Figure 3. 10 Power Switch Inside (own illustration)

Once fully assembled the final case looks like Figure 3. 11. There are sensor wires coming out of the case on one side. The solar panel is glued and wired waterproof on the case top and the antenna screwed into the antenna mount.

Figure 3. 11 Fully Assembled Prototype (own illustration)

3.5.3 PCB Design and Assembly

The PCB is used to place all hardware components on top of one single plate. The used PCB is a prototyping board that can hold multiple components soldered with surface mount technique. All wirings are connected below to make every hardware connector and plugs reachable for fast exchangeability.
The battery is connected with a standardized 2 pin connector to ensure exchangeability again. This opens up possibilities to connect any preferred battery that satisfies the specifications named by TP4056 solar power balancer module in section 3.3. Figure 3. 12 shows a simple LiPo with 3.7V, 20 C and 650 mAh capacity connected to the Arduino which is executing the compiled and embedded software at this moment. Figure 3. 13 shows the bottom side which has the two pins wired and soldered to the according Arduino pins. A note should be placed at this point. The shown pictures display a test state during prototype assembly. The final wiring is different as the battery balancer module TP4056, which is described in section 3.3, is placed in circuit between the battery and the Arduino.

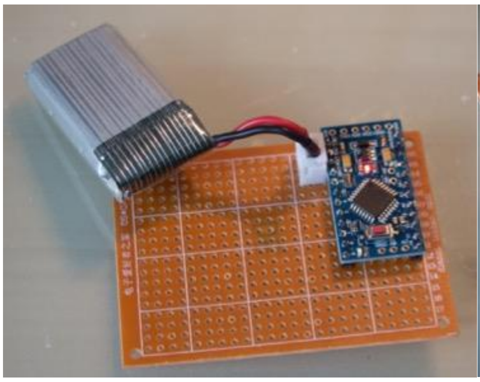


Figure 3. 12 PCB Battery Test Top (own illustration)

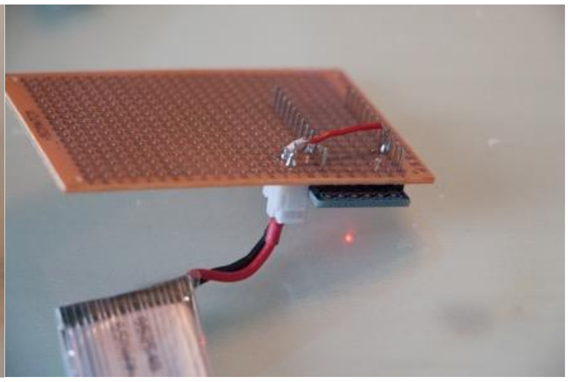


Figure 3. 13 PCB Battery Test Bottom (own illustration)

Figure 3. 14 shows the TP4056 battery power balancer module which is mounted by solder and soldered directly on the PCB prototyping board. In Figure 3. 15 the bottom side is shown with the soldered wiring that connects the balancer module with the Arduino. The hole distance of the standardized prototyping board is not fully equivalent to those of the TP4056 but when using thin solid wires, the soldering can be assembled with no additional adjustments needed.

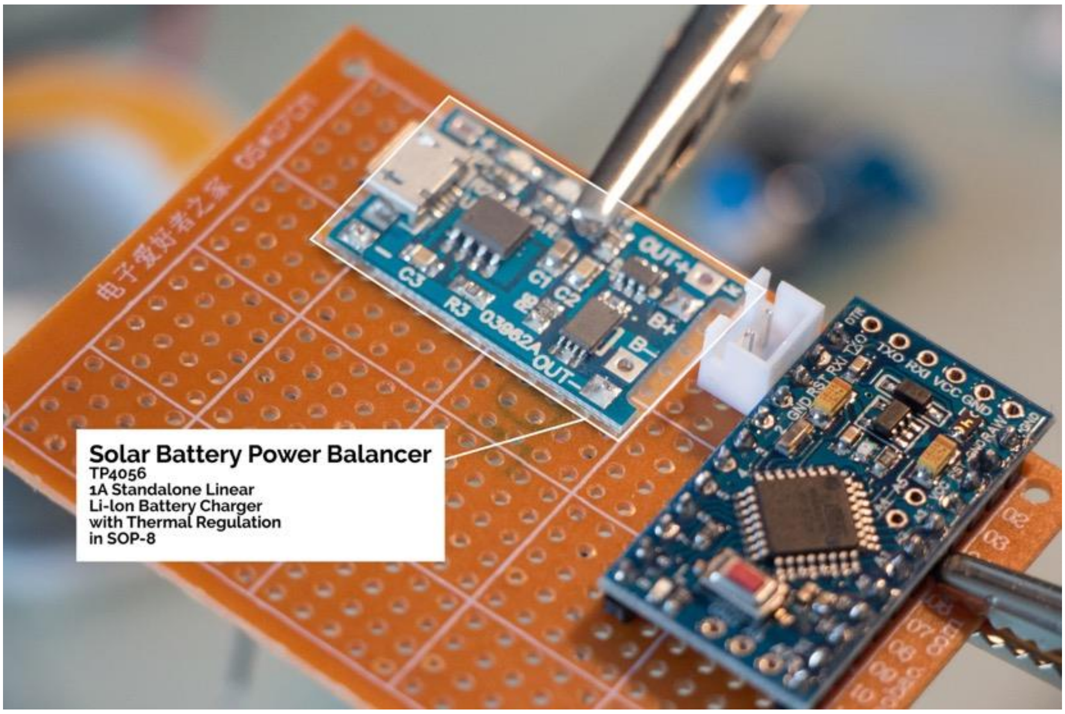


Figure 3. 14 PCB Solar Battery Power Balancer Mounted Top View (own illustration)

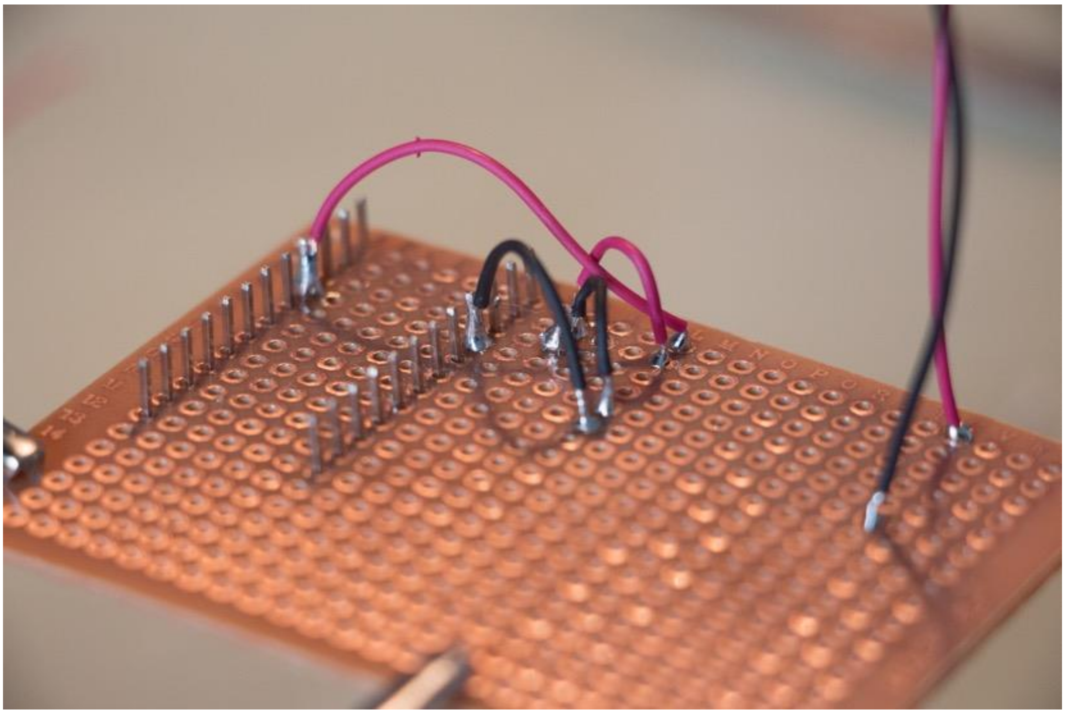


Figure 3. 15 PCB Solar Power Balancer Mounted Bottom View (own illustration)

For testing purposes, the battery is plugged in, to validate that the Arduino is operating with 3.3V supplied through the TP4056 balancer board. This is shown in Figure 3. 16.

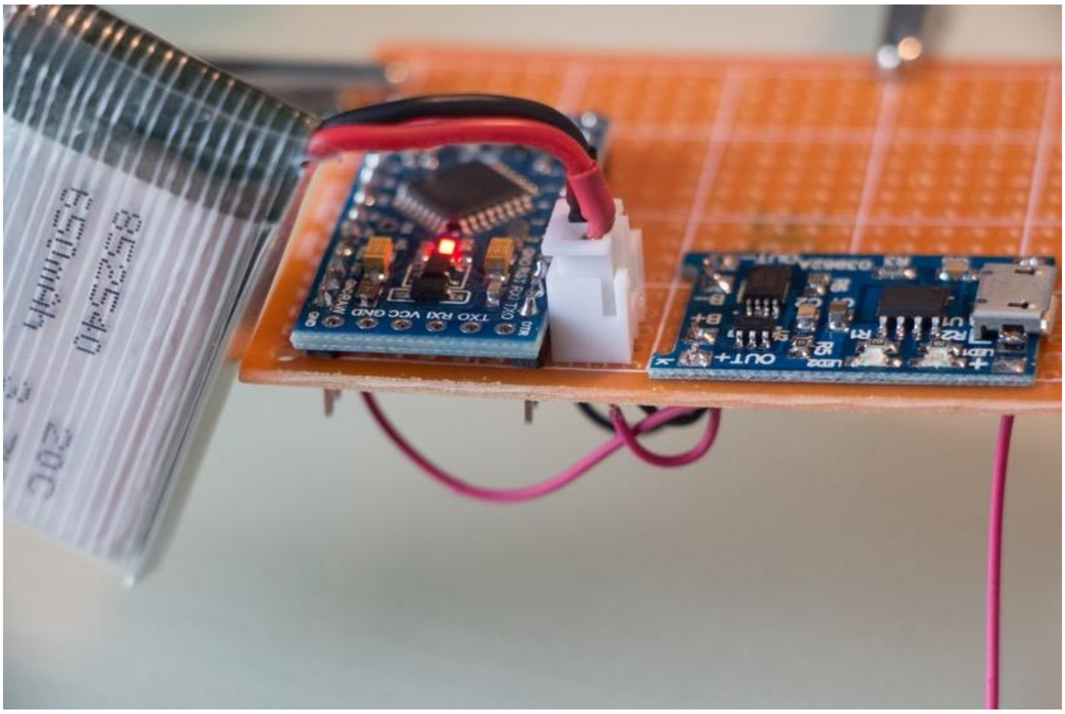


Figure 3. 16 PCB Solar Power Balancer Mounted Side View (own illustration)

Following the battery power test there is a solar power supply test. The solar panel is connected temporarily with multimeter jams to use solar module input power. The test construction is shown and annotated in Figure 3. 18.

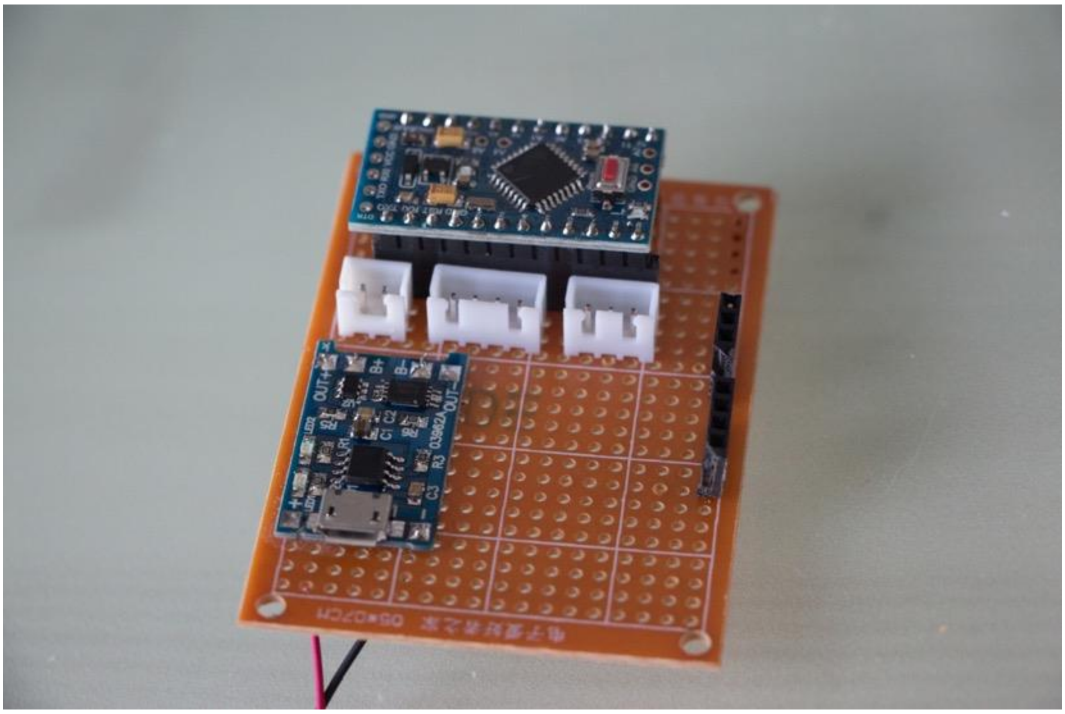


Figure 3. 17 PCB Additional Connector Jacks Soldered in Place (own illustration)

There are two types of sensor connectors. A three-pin connector which offers a 3.3V power supply pin, a ground pin and a digital pin. Additionally, there is a four-pin connector which offers a 3.3V power supply pin, a ground pin, a digital and an analog pin. The unplugged female connectors are shown in Figure 3. 19.

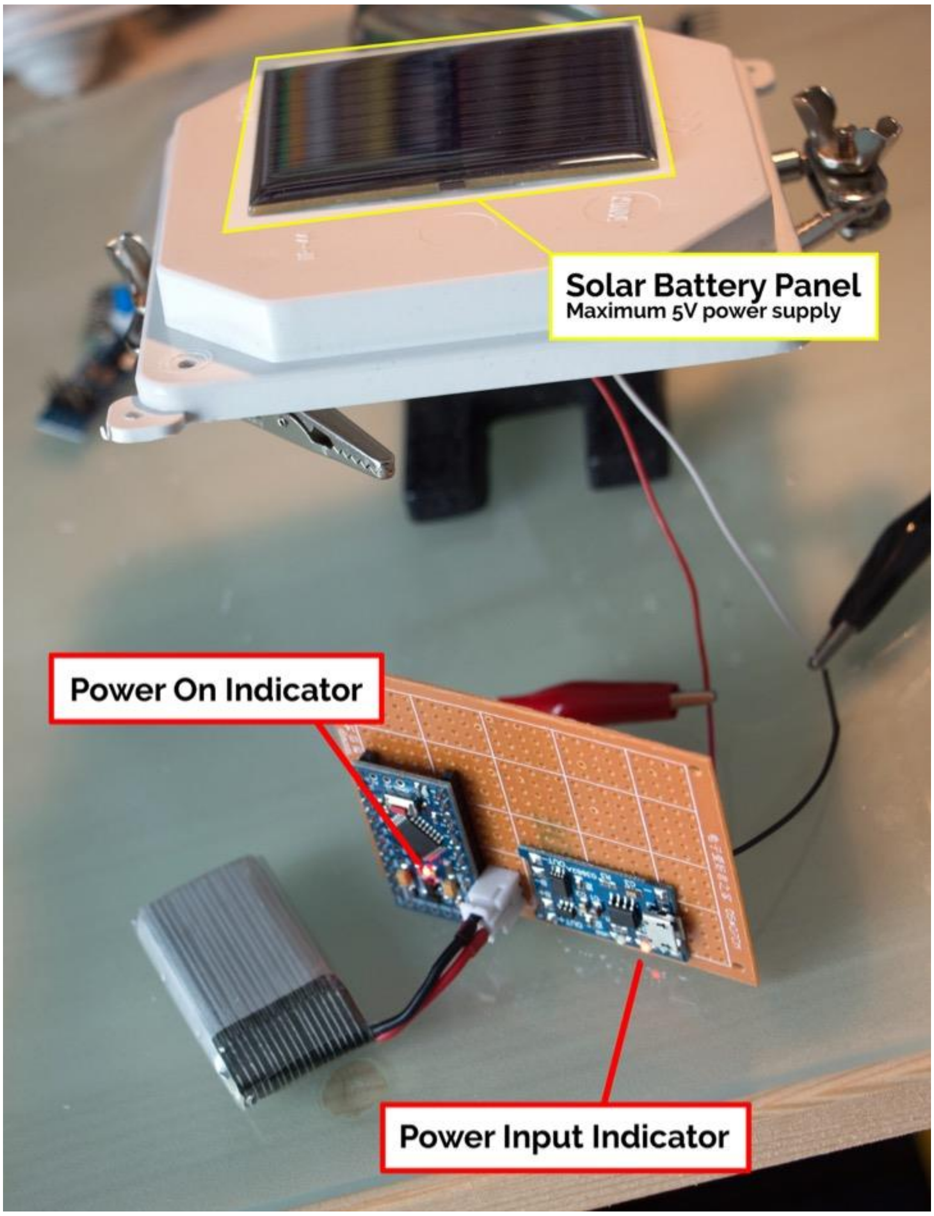


Figure 3. 18 PCB Solar and Battery Power Test (own illustration)

3.5.4 Sensor and Peripherals Assembly

Sensors and peripherals are meant to be exchangeable too. In the developed prototype there are two types of sensors possible. The first one is a three-pin sensor with a 3.3V power supply pin, a ground pin and a digital pin. Additionally, there is

a four-pin sensor with a 3.3V power supply pin, a ground pin, a digital and an analog pin. Both sensors are fully exchangeable. Any low power sensor frugal enough to operate with the supplied 3.3V and a low power can be connected to both of the sensor connectors.
The four-pin sensor is a soil moisture sensor and the three-pin sensor is a digital DHT-11 air humidity and air temperature sensor which can be seen in
Figure 3. 20.

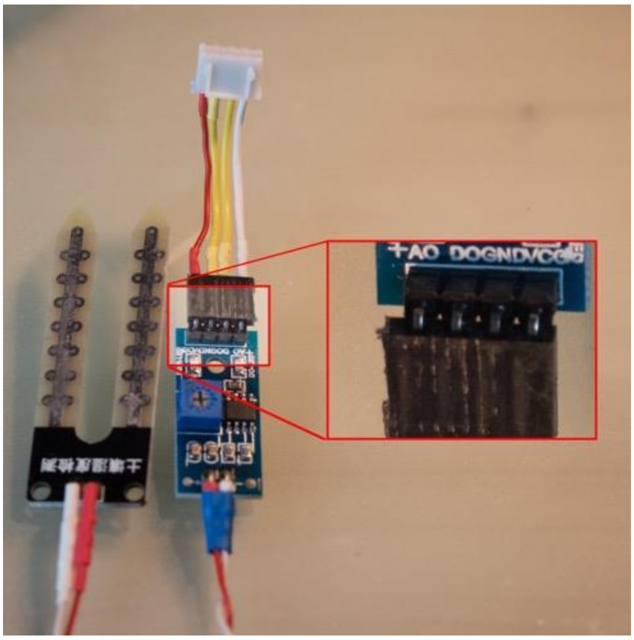


Figure 3. 19 Soil Moisture Sensor Detail View (own illustration)

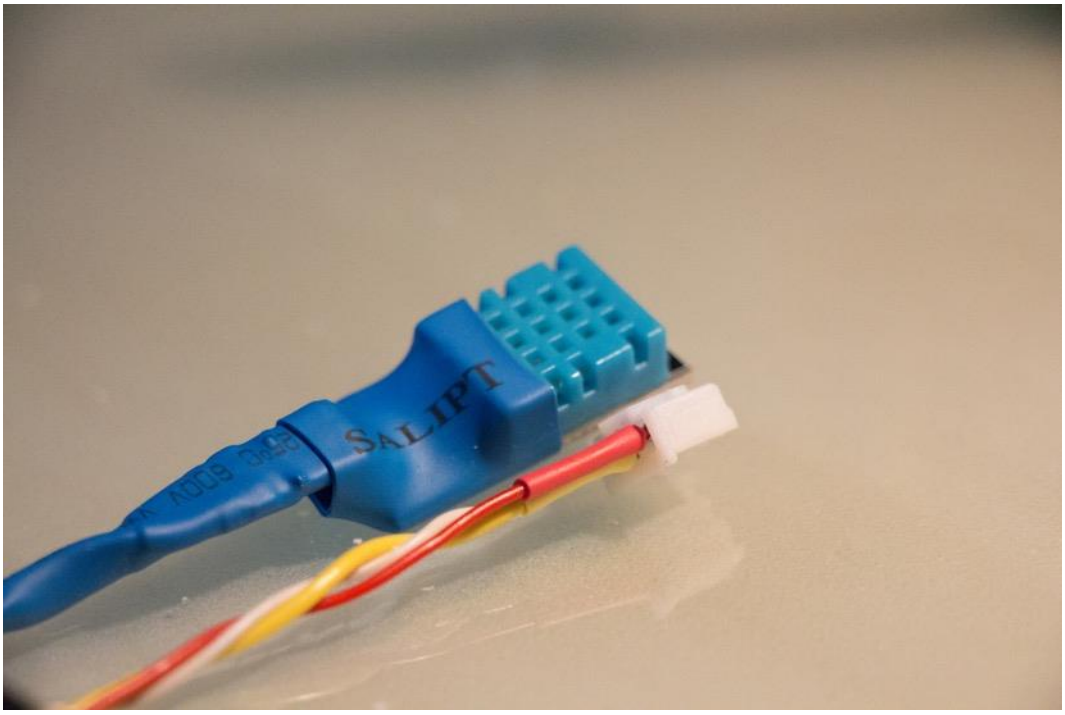


Figure 3. 20 DHT 11 Sensor Detail View (own illustration)

In Figure 3. 21 there is an example of a ground connector bridge concept. This concept is a suitable concept to split an connect multiple electrical sub-modules, peripherals, connector jacks or any other part of the devices electrical circuit to

Arduinos ground pin. This bridge offers flexibility to extend one single ground pin to any number of multiple other ground connections.

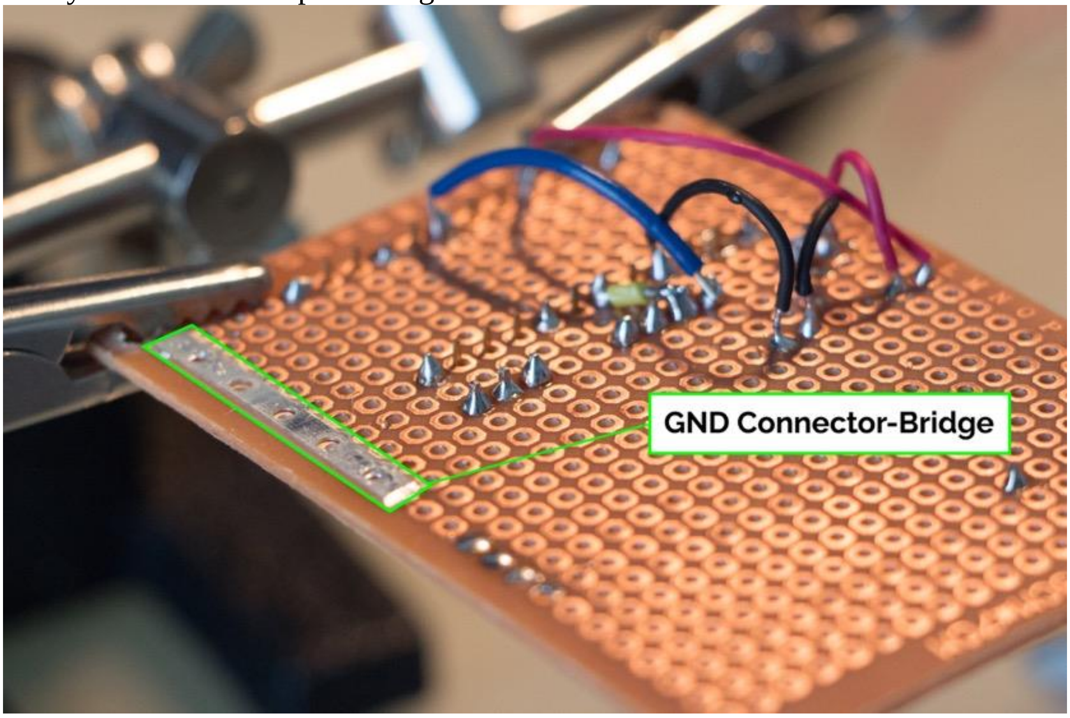


Figure 3. 21 Ground Connector-Bridge (own illustration)

In Figure 3. 22 there is the soil moisture sensor control module mounted on a PC motherboard lift stand. Those lift allow PC assemblies to exchange motherboards from any standardized ATX form factor PC cases. As this standard is spread around the globe it is chosen to be used to mount the soil moisture control module. This enables later exchangeability of the soil moisture sensor. Any other sensor control module can be mounted that fits the motherboard lift stands. In Figure 3. 23 the lift stands can be seen in more detail.

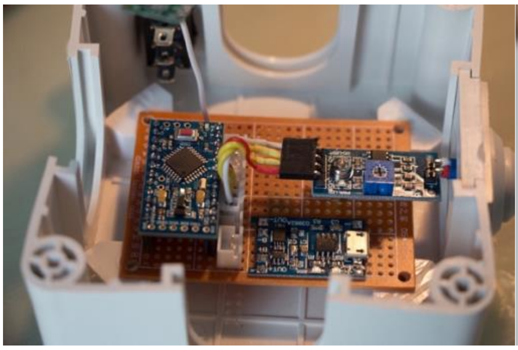


Figure 3. 22 Soil Moisture Sensor Mounted (own illustration)

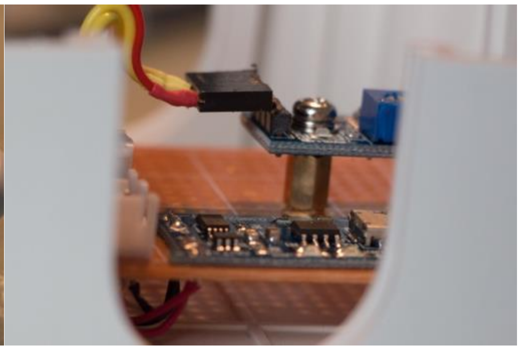


Figure 3. 23 Soil Moisture Sensor Detail View (own illustration)

As next step there is the power switch soldered and connected to the module 1 core unit. The power switch as printed in Figure 3. 24 controls the whole power circuit. It switches the solar panel, battery panel and the power cord power supply.

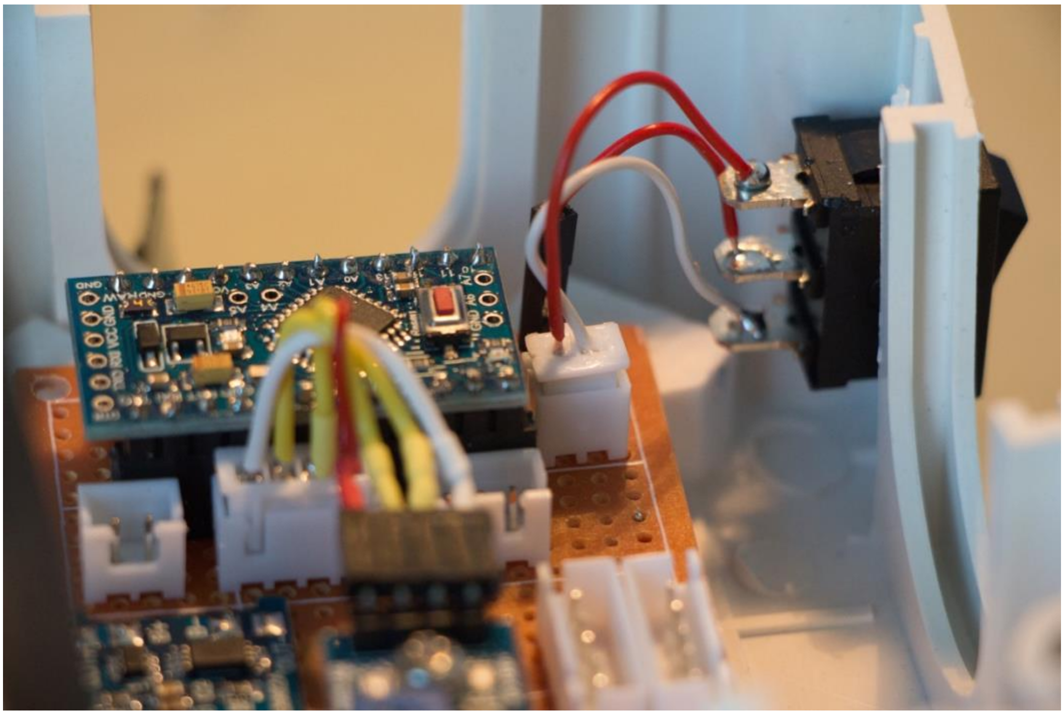


Figure 3. 24 Power Switch Mounted and Wired (own illustration)

The solar module is soldered with a 2-pin connector to ensure easy detaching when changing modules on the inside. Such as changing a sensor configuration or replacing the battery. The twisted and fully assembled solar panel and the case top can be seen in Figure 3. 25.

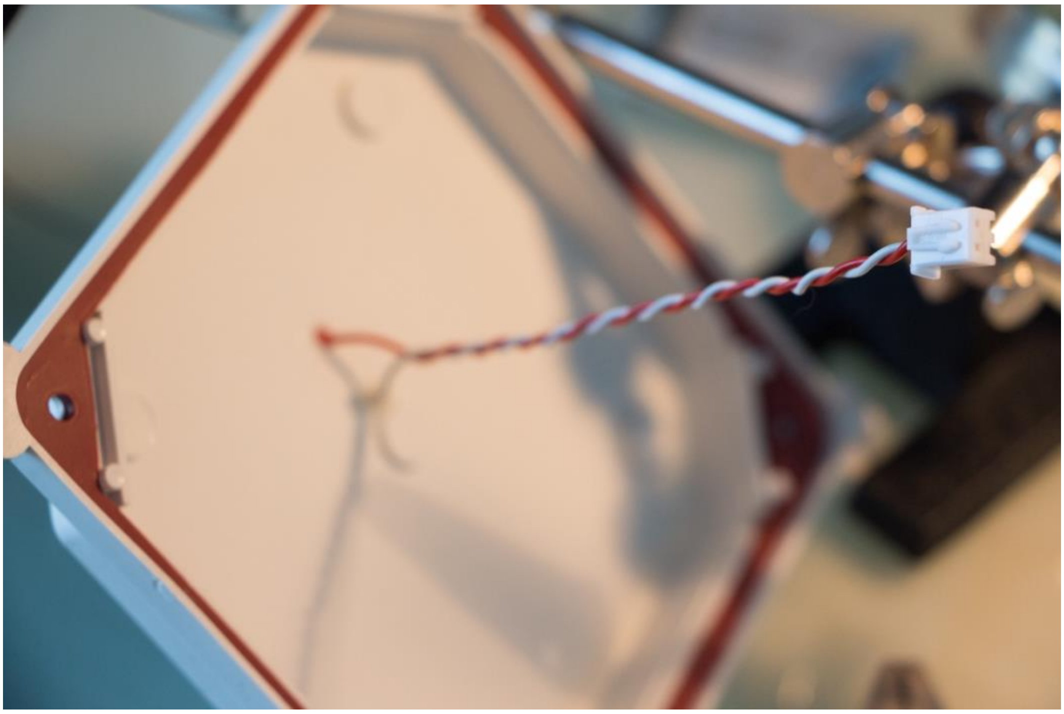


Figure 3. 25 Solar Panel Wiring with Connector Jack (own illustration)

Similar as the solar module power jack the power cord jack is assembled as seen in Figure 3. 26. Figure 3. 27 shows the assembled power cord jack.

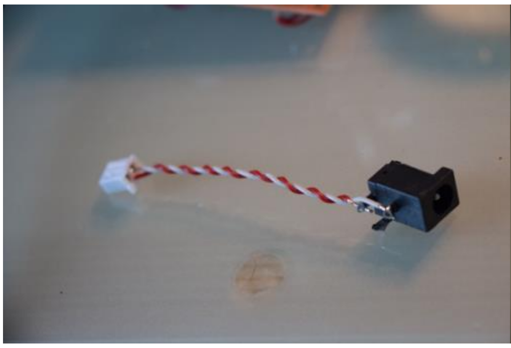


Figure 3. 26 Power Supply Jack (own illustration)

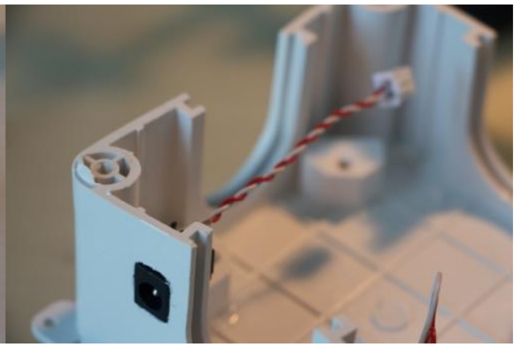


Figure 3. 27 Power Supply Jack Mounted (own illustration)

Once the solar panel is connected, the battery and the power cord connector are working simultaneously to supply power, which can be seen in Figure 3. 28. The TP4056 controls all the power supplies to ensure stable and balanced 3.3V power supply.

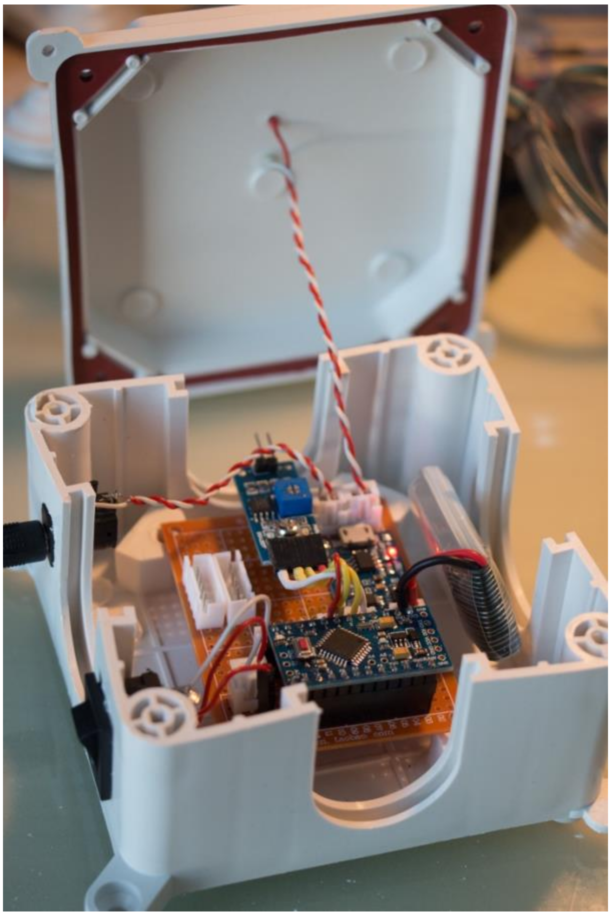


Figure 3. 28 All Power Supplies Connected (own illustration)

The last module to be assembled is the transceiver module. It is also connected with jacks to make exchangeability possible as there are different frequencies depending on the country a device is used. So different transceiver modules are needed to be connected. To connect a transceiver module there are two 5-pin jacks needed. The wiring and soldered transceiver wires are shown in Figure 3. 29. The only left and single wire is the antenna cable.

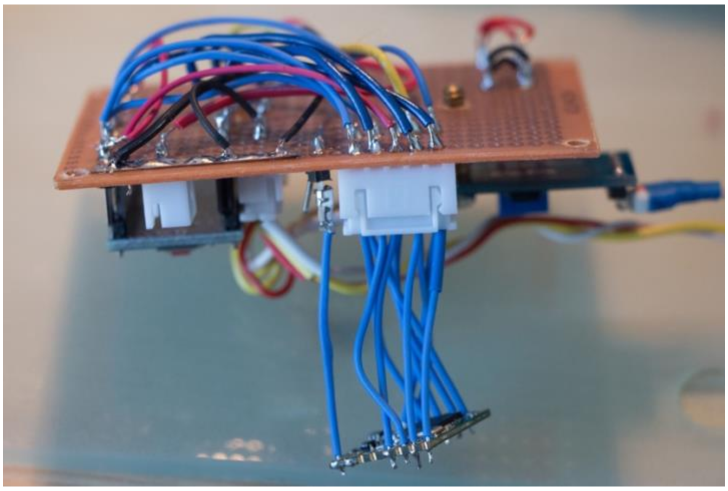


Figure 3. 29 Transceiver Connected (own illustration)

Once all components and modules are assembled the prototype looks like Figure 3. 30 when case top is left open.

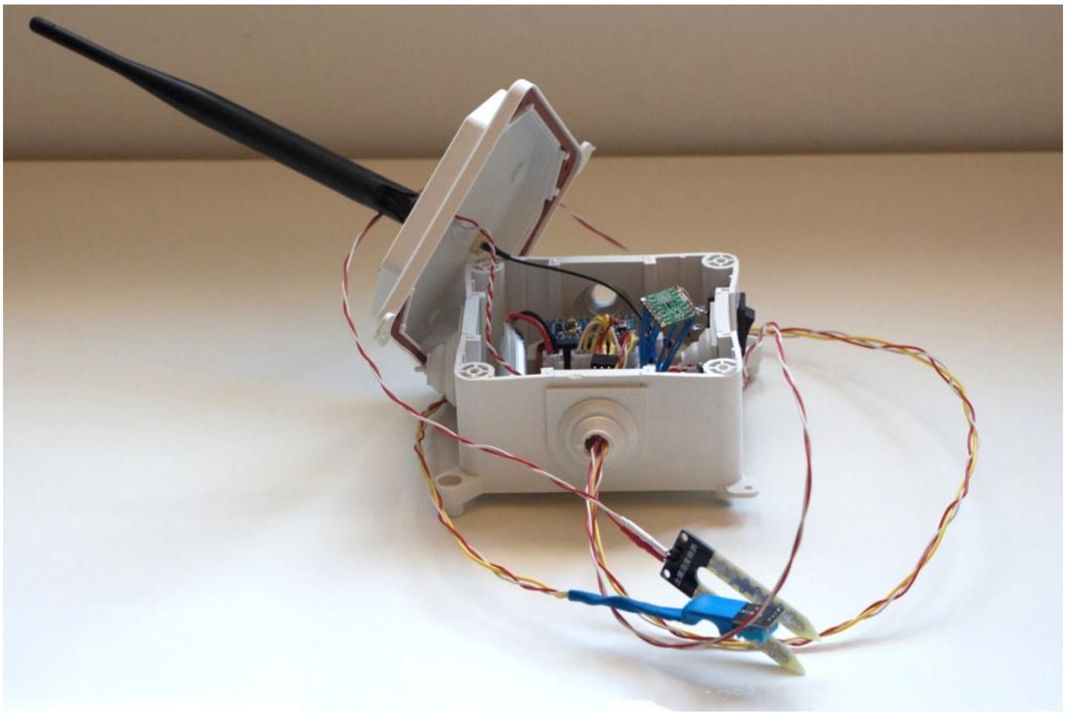


Figure 3. 30 Fully Assembled Prototype with Peripherals Wiring (own illustration)

4 Software Prototype

Describing the software prototype there are multiple topics to cover. This ranges from software concept including the architectural concept, framework concept and components concept, over to topics of requirements, programming and deployment related aspects.

4.1 Software Concept

In terms of the software concept there are several aspects to cover. First there are basic concepts. Second there are architectural concepts. Finally, there are multiple software components that realize the described concepts and architecture. The requirements are derived from QFD aspects of client demands. Winegrowers need simple to use and in best case a software free of manual maintenance. The even need a software that does not need complicated installation and configuration. Following those constrains there is a way to solve this. It can be solved by cloud software or remotely maintained software. Both can be easily realized by WordPress combined with the right tools, which is formerly described in section 4.1.2 in detail.

4.1.1 Architectural Concept

The architectural concept is based on a three-layer architecture described in section 2.2. The three layers are “the thing or device layer, the connectivity layer and the IoT cloud layer” (Wortmann and Flüchter, 2015). The device layer is covered in chapter 3 before. This section is about the connectivity and cloud layer. The connectivity layer is realized by the global infrastructure of TTN, which is already described in detail in section 2.2. TTN offers some connectivity integrations such as a HTTP API, which is used and described in section 4.3.1 for configuration and in section 4.3.3 for sensor data push services. The cloud layer is realized by two applications. The first application when viewing bottom up is the services layer application. The services layer is kind of a sub layer inside the cloud layer which acts as a middleware. The services layer application itself is a partly RESTful API service which collects, stores and offers sensor data from TTN towards any kind of client, which can consume this standard HTTP RESTful JSON API. The first and so far, advanced client example is part of this thesis’ prototype implementation, which is the web frontend. The web frontend as a client is responsible for displaying the data loaded from the services layer but will not change the data inside the services layer. This is an important architectural aspect to be explained. As the data shall be useful for any kind of application and any kind of use case derived from the sensor data collected. Some examples for those use cases are described in multiple chapters before. A straight forward example is an automated irrigation system that consumes sensor data from the services layer to improve the plant irrigation in specific locations in the vineyard. Such a system would consume and use the data solely instead of displaying them in a human understandable way. The following Figure 4. 1 describes the data flow through all three layers. All data starts in the IoT device. From there the data is transmitted over LoRa to a TTN gateway. This gateway routes all received data towards TTN cloud services. The data is there cached inside TTN for seven days. Simultaneously the data is pushed over HTTP to the Service Layer application. The Service Layer application is intentionally planned to implement an OAuth 2.0 authentication service. This can be extended later if needed. There is an interface already predefined for that purpose. For simplicity the prototype version uses HTTP "Basic Auth" mechanism for authorization. Following the next steps in Figure 4. 1 there is a CRUD implementation. This implementation is considered in conjunction with a pretty generic database connector. This database connector can handle a wide range of modern standard database systems. From MySQL to MSSQL to MongoDB. Finally, there is the before described REST API that is consumed by the web front end.

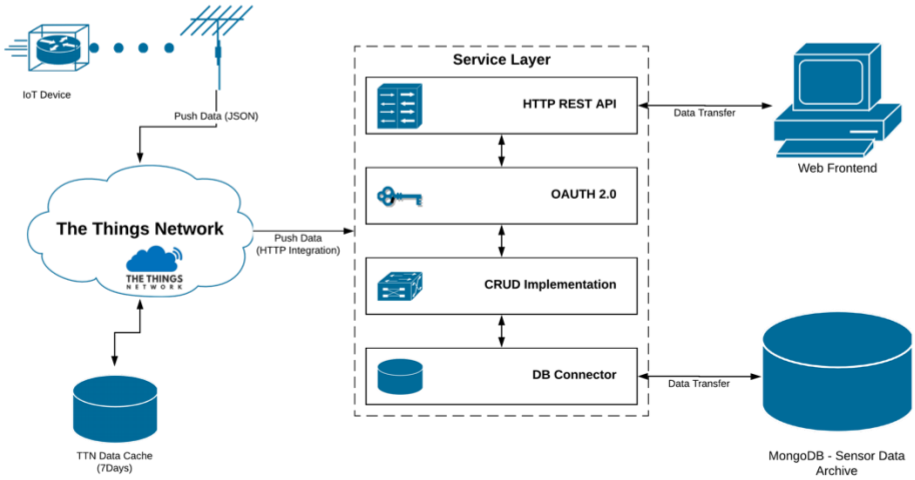


Figure 4. 1 Vinio Infrastructure Architecture (own illustration)

4.1.2 Framework Concept

As basic concept for the software prototype is to use multiple frameworks. Though the services layer is not based on a framework it is even based on multiple libraries. Nevertheless, the front end is realized as a rich web client. It is realized as a WordPress plugin and uses WordPress plugin framework features. WordPress is chosen due to its wide adoption through the whole internet. A common number is often referred as, around 30% of the whole world's websites are powered by WordPress. This is often related to the simplicity of WordPress' use. Additionally, WordPress brings standard features as a user management system with login and rights management. Due to WordPress' global adoption there are plenty of resources to refer to and get help from. The plugin can be easily installed to any current WordPress Website from a packed ZIP-file. It then must be configured with the right credentials to connect to the services layer. Configuration is made through an administration panel in the WordPress back-end. Figure 4. 2 shows a screenshot of the administration panel. The administration panel is a framework designed for WordPress themes and plugins. It is called “Codestar Framework”. This framework is one of the most intuitive open source options panel frameworks for WordPress that also follows the WordPress back end look and feel. Besides this major reason the other major reason to choose this options panel framework is the simple way to create the administration panel. It is divided into multiple sections aligned vertically as a menu structure on the left side of the panel.

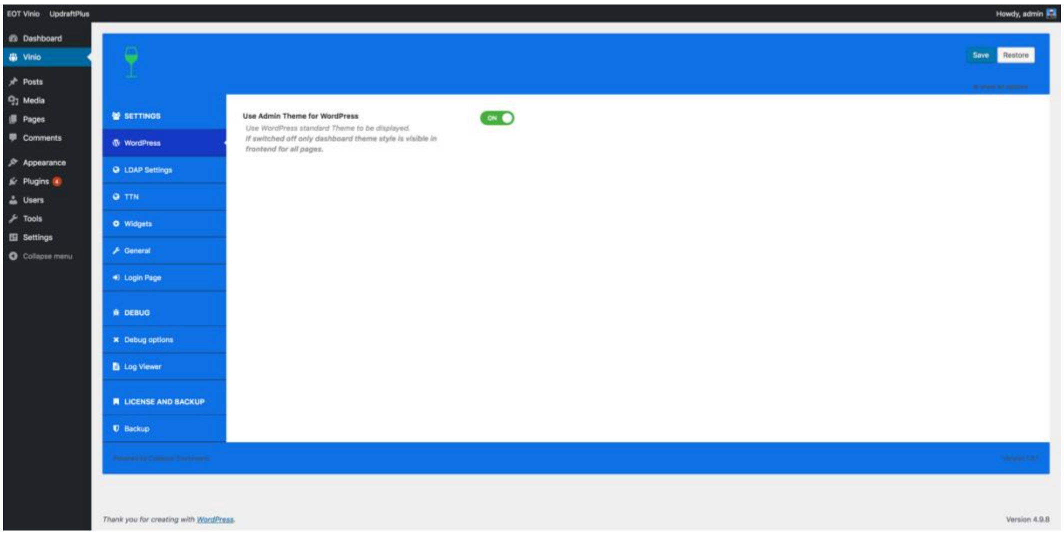


Figure 4. 2 Administration Panel Overview (own illustration)

4.1.3 Components Concept

The most important section is the TTN section. This section handles all TTN related configuration. It is shown in Figure 4. 3 and it realizes the key concept of the client application. The key concept of the client application is the generic adoption of all IoT devices registered to the connected TTN application. The client synchronizes all registered TTN devices automatically. Once there is a test data set send over through TTN console it can be synced by the front-end client and is available in the settings panel.

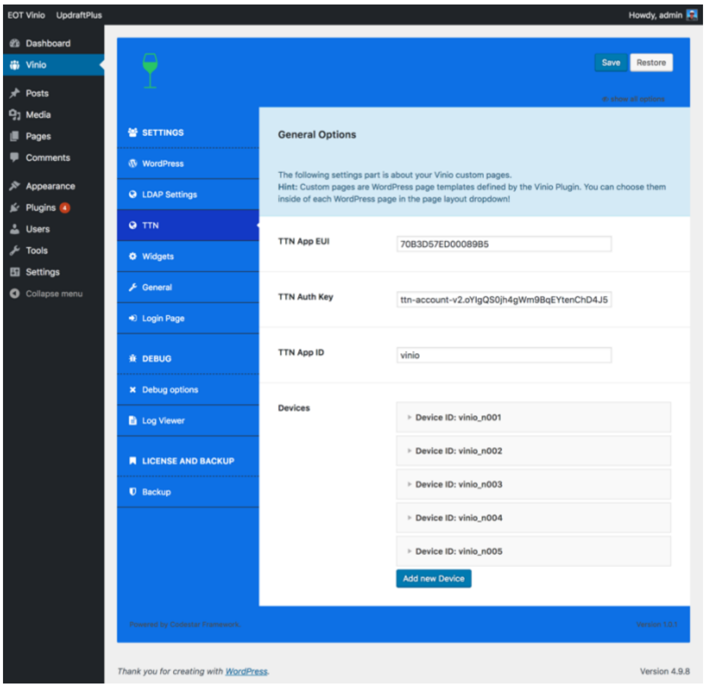


Figure 4. 3 Adminsitration Panel TTN Section (own illustration)

Once the TTN connection is fully set up there are all TTN devices available for the corresponding TTN application. In Figure 4. 4 there is a device selected, which is called “vinio_n002”. To be able to use Arduino nodes generically the back end is able to update and receive newly registered devices from TTN automatically. In order to make this possible the user must map all available sensor data with an alias due to the naming of underscore as a word separator in Cayenne LPP encoding. As an example, “relative_humidity_#” is used to identify a float value of relative humidity, while “#” is a placeholder for an integer number to transfer multiple humidity values. It can get an alias which is displayed in all data presentation graphs in the front end of the web client application. Additionally, the geolocation can be configured. This is possible due to the static location of an IoT device in the vineyard use case. In other use cases there could be a necessity to add GPS data to one of the transmission data sets. Describing the transmission data sets there is also the ability to assign a data mapping to the device data. The device data section is described as this section allows a user to map incoming source data fields from an IoT device to corresponding data set and assign it an alias for better understanding it's meaning while viewing them in the front end. Each data field can be selected, and an alias can be assigned. Additionally, a data calculation formula can be configured. As Figure 4. 4 shows the battery voltage data must be converted to a relative percentage. This conversion is calculated by dividing the data value by 325 and multiplying it with 100. The resulting value is in dimension of percent from 0 % to 100 %. The meaning of this example data is fully discharged on 0% and fully charged on 100 %. The value of 325 corresponds to 3.25 Volts current that is supplied by the battery built into the IoT device prototype.

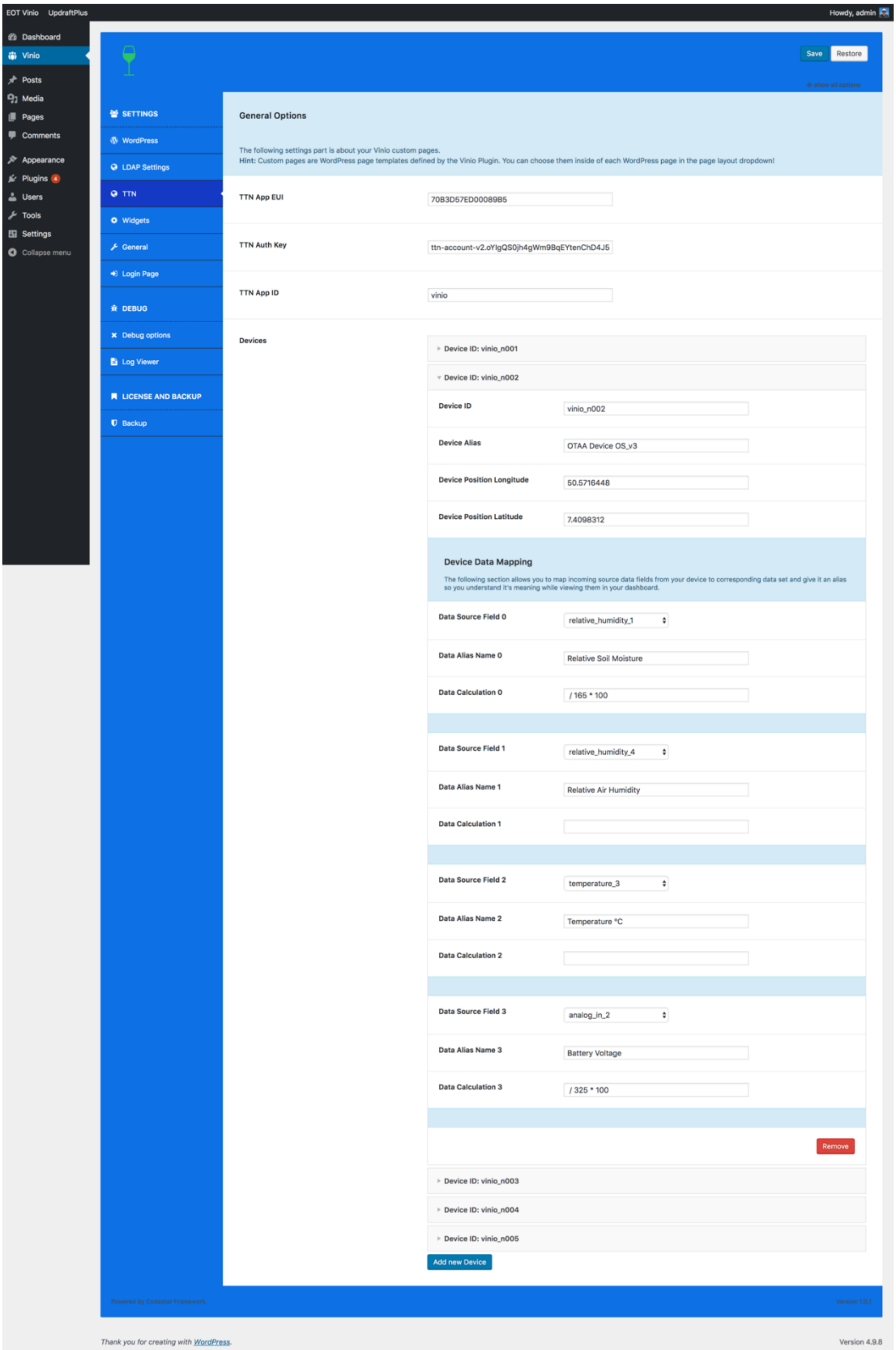


Figure 4. 4 Adminsitration Panel TTN Section Devices Unfolded (own illustration)

4.2 Graphical User Interface

The graphical user interface (GUI) is based on the GUI framework "Gentelella!". Which causes the used map tiles from a free map by mapbox.com, that is customized in color palette, to seamlessly integrate and fit the GUIs color scheme. This framework is made to simplify admin dashboard development and is only built in HTML, CSS and JavaScript. The use of additional libraries is optional and depending on the use case's needs. Gentelella! is based on Bootstrap 3 with jQuery plugins and tools.

Figure 4. 5 (Gentelella!, 2018)

Another basic aspect of the GUI is the front-end application environment. Thus, it is a browser application there are common requirements and restrictions brought by web-browser applications. There are multiple different client browsers that can run the front-end. Additionally, the design must be responsive to support any modern end user devices. From desktop PCs down to smart cell phones. Even tablets are an important device formfactor. All those devices must display the smart vineyard infrastructure front-end. Thus, the use of a GUI framework is a solid foundation to rely on when looking for solutions to meet the requirements brought by web application requirements.

4.3 Programming and Deployment

The programming follows the WordPress Codex standards described in the online resource of the WordPress Codex documentation. The front-end web client is programmed as a WordPress Plugin which indicates to use PHP and JavaScript for programming and CSS and HTML for style and markup. The libraries are managed by a PHP specific and most common package management system called "composer". The "composer" enables to store a JSON file including versions of

wanted or needed public open source libraries. Mostly they are hosted on GitHub where most sources can be pulled from. Additionally, the Services Layer is also based on PHP libraries that implement fundamental features such as the OAuth or the mostly generic database connection.
Regarding the deployment both applications must run in a standard web server with a PHP 5.6 or later environment. As web server either Apache or NGINX can be used. But any environment suitable for WordPress is also suitable for the web front end client plugin as the plugin follows the WordPress codex. Deployment can be simply done by the standard WordPress plugin update process. Once there is a new version of the web client front end WordPress displays an update notification. Updates can then be installed by just a few clicks or remotely by a system provider. The described abilities refer to the fundamental idea of an open service infrastructure to monitor vineyard wellness.

4.3.1 Data collection

In general data is collected through the before described The Things Network. Figure 4. 6 shows an overview of the live data received from devices belonging to the Vinio TTN application. The exact meaning of the fields section is explained in section 4.3.3.
The TTN HTTP API offers multiple REST endpoints to connect to and operate with. There is an endpoint to receive a list of all devices belonging to an application. This endpoint is used to synchronize all devices to be displayed in the device configuration as explained in section 4.1.3 before. There is always a gateway connected to the internet that routs the received data to the according data target. Either the TTN 7-day data cache or any other data integration configured for an application through TTN.

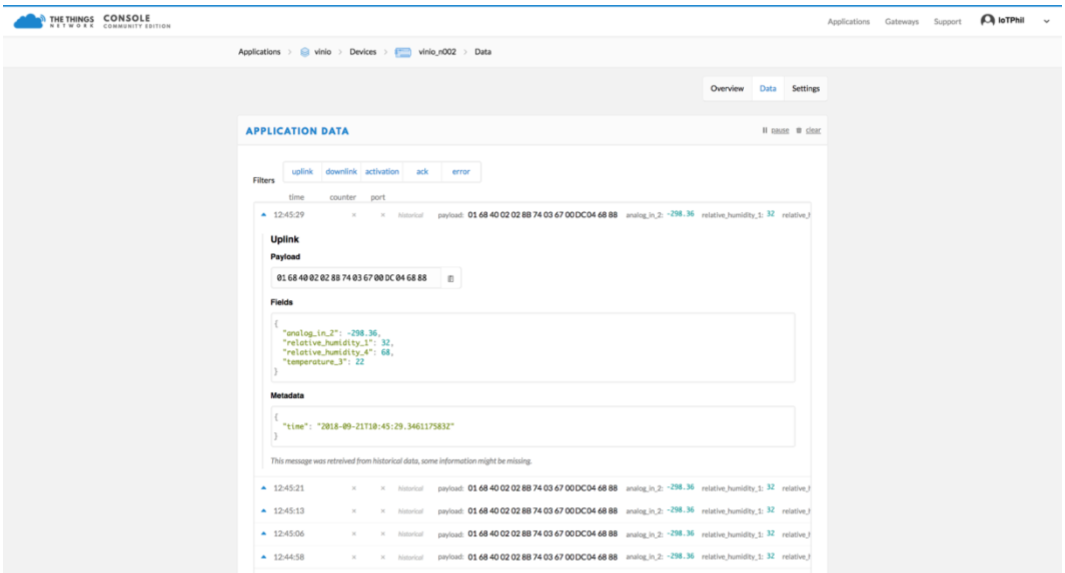


Figure 4. 6 TTN Live Data Observation (own screenshot)

4.3.2 Data presentation

Data is presented by different graphs. One central graph is a line chart as shown in Figure 4. 7. This line chart is based on "Chart.js". The used JavaScript library is open source. It is able to display any number of data points per line and any custom defined range of time. But all data is aggregated down to the maximum of 40 data points. The maximum number of data points is chosen to ensure best performance while displaying the line chart and data retrieving performance through the services layer. Also, Web-Browsers are restricted due to the end user device they run on. The line graph shows two dimensions. The abscissa shows the time dimension and ordinate shows the values defined by the current sensor data. The abscissa range can be adjusted by a minimum and a maximum value as shown in the widget settings in Figure 4. 8. Settings are saved as cookies in the client's web-browser. This enables users to set individual settings per user and per client device.

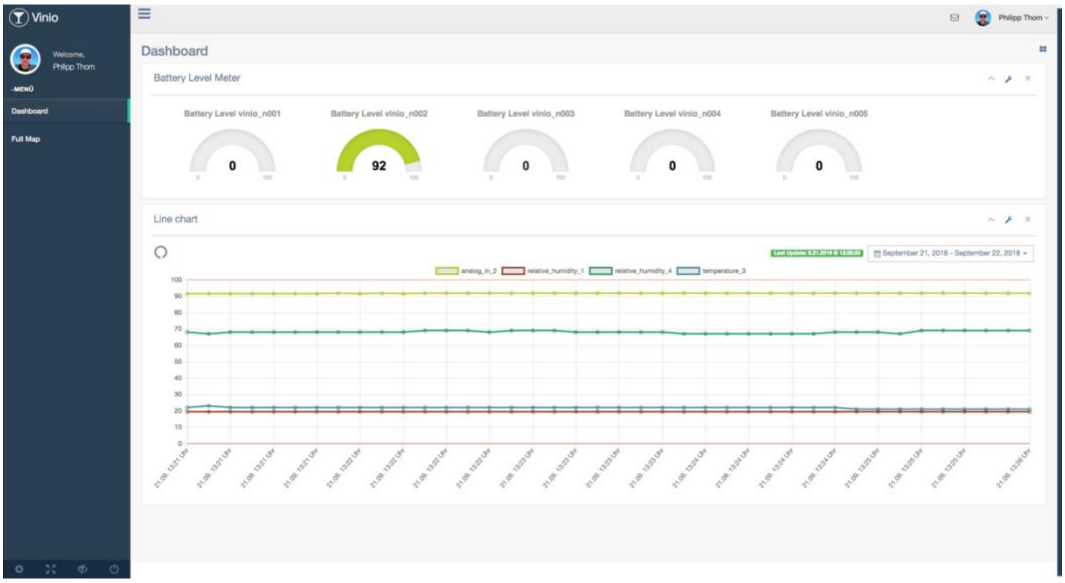


Figure 4. 7 Front-end Line Chart and Gage Graphs (own illustration)

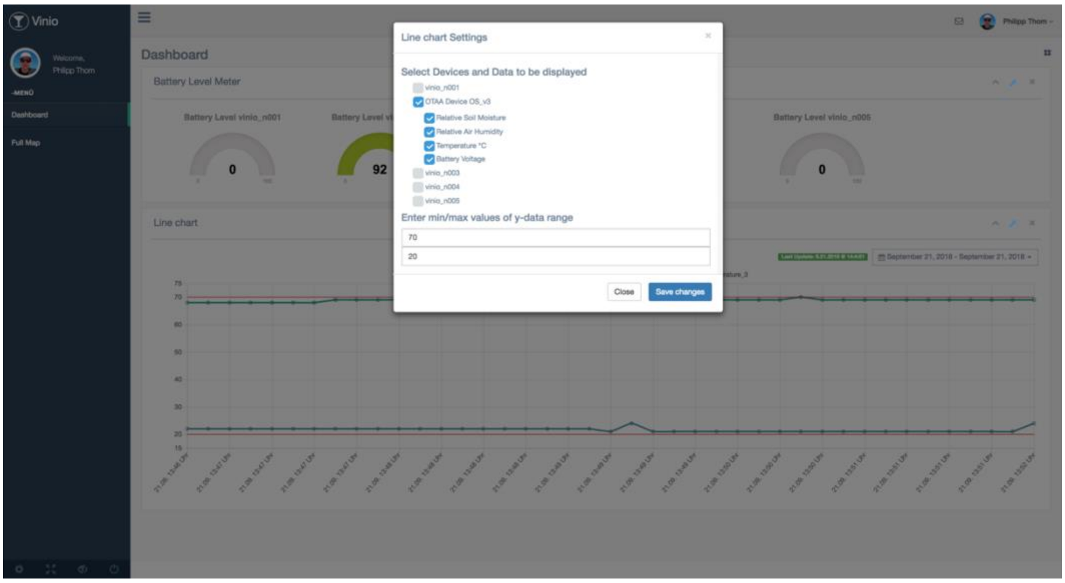


Figure 4. 8 Dashboard Line Chart Settings (own illustration)

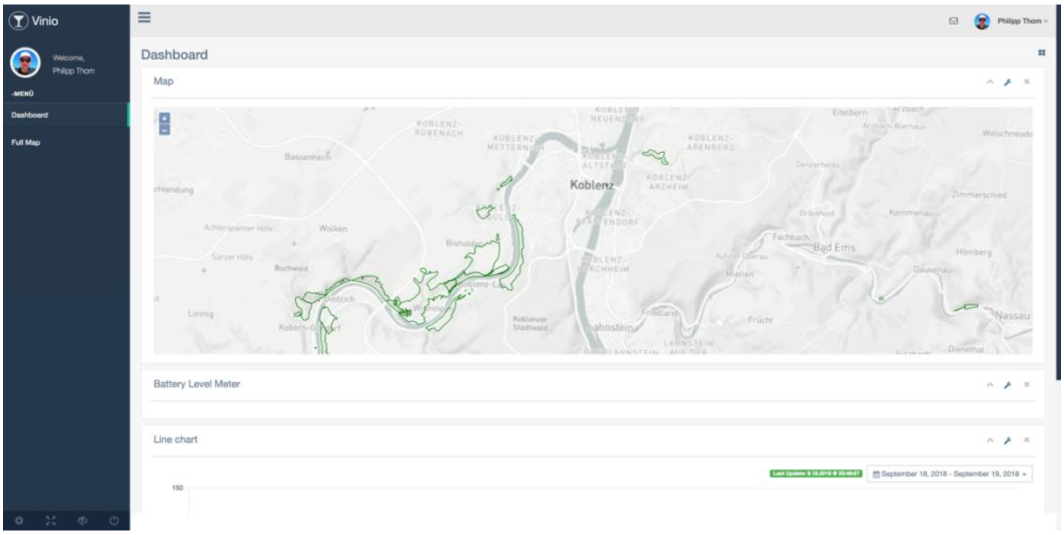


Figure 4. 9 Front-end Vineyard Map (own illustration)

In Figure 4. 7 there is a map currently showing Koblenz and its surrounding area. There are green shapes that mark the vineyard areas. The vineyard areas are projections of shape files from ".wms" vector files. They are provided by Institute ©Landwirtschaftskammer RLP (2018), dl-de/by-2-0, http://www.weinlagen.lwk-rlp.de The projection is rendered by the open source map library open layers. The projection has the specification of "EPSG:25832". Thus, open layers cannot render projections by default an additional library is needed. It is the "Proj4js" which is a JavaScript library that transforms coordinates from one coordinate system to another. The shape files must be prerendered to vector files by another service. This service is called "GeoServer". It is a java application free of use and is deployed as a docker container.

4.3.3 Data interconnectivity

The data interconnectivity between the two cloud layers is completely based on HTTP and mostly follows REST standards. The data encoding format used to transfer data through each request is JSON, as it is a used standard for web data transportation.
There are two implemented resources provided by the Services Layer application. First there is a resource that consumes push data from TTN. All measured and received data from TTN is directly pushed to this resource endpoint. The second resource is a data offering endpoint, which offers data based on provided parameters.
The data push call to send and store data in the Services Layer application is based on HTTPS POST. The data push call is defined as follows.

```
https://domain.tld/v1/applications/
```

The above pseudocode follows a REST-like URL notation and shows the API versioning, with "v1" in front. Followed by the applications resource, as defined by REST. REST defines requests to create a resource as HTTP POST method. Data is transmitted in the request body as plain JSON. An example data set looks like follows. The data structure shows a simple JSON key/value pairs. The data encoding to use keys like "analog_in_X" are related to the key naming of CayenneLPP encoding, which is described related to the hardware aspects in chapter 3.

```
{
 "analog_in_2": 292.36,
 "relative_humidity_1": 31,
 "relative_humidity_4": 95,
 "temperature_3": 20
}
```

The data request call to retrieve data is based on HTTPS. It requests the data resource as follows.

```
https://domain.tld/v1/applications/
?app_id=$app_id&dev_id=$dev_id
```

The two parameters "app_id" and "dev_id" are not RESTful but are chosen due to simplicity of the first valuable prototype. The HTTP GET method is used to request data and follows the REST standard.

5 Analysis and Conclusion

Studies on world's climate change point out the near-future challenge that successful winemaking needs adaptation to changes in the climatic conditions over the year, and recently, needs both, adaptation to the single events of local weather disturbances (storm, massive rain downfalls), and, as well the geographical shifting of winemaking zones (Stock, 2006). Thus, as a profound change, today's preference areas for white wine cultivation, will move towards the northern hemisphere, and turn over the decades to red wine cultivation zones, just due to the climate change. On the micro-scale, winemakers will need to develop many more tools for climate and other parameters' short-term prediction, and for the acute condition of their vineyards, where the fruit of their labor endeavors come from.

5.1 Contribution to Theory

The first research question asks if there is a technological solution to support winegrowers in their efforts to prevent threats arising from climate change by monitoring climate data directly in place in their vineyards?

In order answer this question and the first objective, to create a technically working and, to a certain level, reliable prototype, this thesis shows a both service-based information network from possibilities brought by IoT and plant growth data collection in a flexible and easily deployable open service infrastructure approach and an information network to monitor vineyard wellness as a practical use case. Profoundly this thesis proves and realizes a prototype as its main outcome, that is technically working and to the level of multiple field tests reliable. The infrastructure collects, transmits and stores climate data, in a cloud-based infrastructure, from multiple statically geo-positioned sensors and displays them in charts and graphs, visualized in a responsively designed web portal client front-end.

The deployed and characterized infrastructure is secondly able to collect, transmit and store soil moisture, air temperature and relative air humidity data from multiple statically geo-positioned sensors and display them in charts and graphs, visualized in a responsively designed web portal, as described in chapter 4. As shown, the described and built infrastructure developed in this thesis is not only capable of its intended purpose, but also to be capable of an unforeseeable number of other use cases that could be realized by using already available sensors offered by global market of IoT, whilst opening application also outside the winemaking business.

The second objective to show feasibility to support winegrowers in monitoring their vineyards by an open IoT service infrastructure, to enhance earnings control even on terroir level is described and built as the developed infrastructure in this thesis, which is not only capable of its intended purpose to support winemakers in their business and reduce climate change threats by justified monitoring and ability of winegrowers to act on basis of the collected and visualized data. Smart automated irrigation systems can use cloud data to base calculations of irrigation amounts on. Therefore, this infrastructure is able to support smart irrigation systems by supplying data in a cloud realm and its communication standards.

The basic concept of measuring data in a certain location and using them in a cloud infrastructure enables multiple businesses to benefit from and extend the fundamental infrastructure implementation to any kind of use case they found their business or at least a part of their business on. This potential is empowered by the chosen flexible construction principle of the device, reasoned by the continuously realized modularization through all the infrastructure components. Starting from the software, with its components of the embedded software, where any preferred sensor could be added as a library. This concept is continued by the service layer application, which offers data endpoints concluding with the client front-end dashboard with generic widgets, that can be configured to any kind of sensor measuring data input. All those software components are designed to be extendable to another data stream generated by sensors from the IoT prototype device, for which I have created the name *Vinio.* The *Vinio* prototype is also fully modularized as the concepts of modern PC assembly are used to build the electronics design. Standardized connectors give even end-users the ability to change sensors connected to their Vinio IoT device by plugging-in a different sensor as a peripheral like a USB mouse to a PC.

The required potential to prevent failure of grapevine earnings by using IoT supported vine production and using the benefits from failure prevention, enabled by monitoring climate data in vineyards, will finally strengthen winegrower's business. This postulate is profoundly based on the approach to give winegrowers the right tool to act and protect against weather extrema. Nevertheless, derived needs for grapevine care must still be considered by winegrowers themselves.

5.2 Limitations

Although there are still limitations to the prototype implementation as it is the nature of a prototype and as it complies to the DSR method, limitations can be named as regulatory and legal aspects to be allowed to sell the prototype device as a market product. Device reliability to operate over multiple years in the environment is defined by the applicable use case. The environmental conditions of a vineyard are described and fulfilled by the specifications, which the commodity modules fulfill themselves; however, this is not proved by end-of-line tests after assembling the prototype devices for commercial use. These and other device tests must follow in a process of improvements, which are to be realized in future work to follow guidance derived from QFD method. Those requirements must be evaluated and applied whether they are suitable in order to meet commercial production standards. The method of QFD in relation to this thesis' application is solely used to take advantage of the guidance of this industrial standard and to give reference to this method of a structured and regulated product development process.

The full QFD method might be used in future work to create an industrial product for business use.

5.3 Future Work

As an outlook, the importance of development of new technology can be seen on a larger scale as a contributing factor for the of earth population's world nutrition by agriculture and farming as a whole. Small solutions like an agricultural smart weather data monitoring system contribute on the small scale, with validity and importance for the large scale. Even on the given scale this small system can be easily extended to a powerful supporting system by additional implementations of the sensors of choice.

Literature References

adafruit (2018) 'Arduino library for DHT11DHT22, etc Temp & Humidity Sensors'. Available at: https://github.com/adafruit/DHT-sensor-library.

Adelantado, F., Vilajosana, X., Tuset-Peiro, P., Martinez, B., Melia-Segui, J. and Watteyne, T. (2017) 'Understanding the Limits of LoRaWAN', *IEEE Communications Magazine*, 55(9), pp. 34–40. doi: 10.1109/MCOM.2017.1600613.

'Agro-climatological and phenological characteristics of the Middle Moselle valley. Effects of climate change to grapevine, Moselle River Region.' (2001), p. 60.

Akao, Y. and Mazur, G. H. (2003) 'The leading edge in QFD: Past, present and future', *International Journal of Quality & Reliability Management*, 20(1), pp. 20–35. doi: 10.1108/02656710310453791.

Al-Alawi, A. I. (2006) 'WiFi Technology: Future Market Challenges and Opportunities', *Journal of Computer Science*, 2(1), pp. 13–18. doi: 10.3844/jcssp.2006.13.18.

Alves, A., Silva, H., Lourenço, A. and Fred, A. (2013) 'BITalino: A Biosignal Acquisition System based on the Arduino', *Proceeding of the 6th Conference on Bio-Inspired Systems and Signal Processing (BIOSIGNALS)*, (June 2014), pp. 261–264. doi: 10.5220/0004243502610264.

Anlagen, E. and Montagestellen, B.- (2013) 'L4', 67, pp. 7–12.

Arduino (2016) *Arduino Pro Mini*, *Arduino Products*. Available at: https://www.arduino.cc/en/Main/ArduinoBoardProMini.

Arduino (2018) *SPI library*. Available at: https://www.arduino.cc/en/Reference/SPI (Accessed: 26 June 2018).

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D. and Rabkin, A. (2010) 'A View of Cloud Computing Clearing the clouds away from the true potential and obstacles posed by this computing capability', *Communications of the ACM*, 53(4), pp. 50–58. doi: 10.1145/1721654.1721672.

Battle, R. and Benson, E. (2008) 'Bridging the semantic Web and Web 2.0 with Representational State Transfer (REST): Semantic Web and Web 2.0', *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(1), pp. 61–69. doi: 10.1016/j.websem.2007.11.002.

Benedikt Frey, C., Osborne, M. A., Dewey, D., Dorn, D., Flint, A., Goldin, C., Muellbauer, J., Mueller, V., Newman, P., Ó hÉigeartaigh, S., Sandberg, A., Shanahan, M. and Woolcock for their excellent suggestions, K. (2013) 'the Future of Employment: How Susceptible Are Jobs To Computerisation? *', pp. 1–72. doi: 10.1016/j.techfore.2016.08.019.

Cata, T., Patel, P. S. and Sakaguchi, T. (2013) 'QR Code: A New Opportunity for Effective Mobile Marketing', *Journal of Mobile Technologies, Knowledge and Society*, 2013(748267). doi: 10.5171/2013.

Conformity assessment - European Commission (2018). Available at: https://ec.europa.eu/growth/single-market/goods/building-blocks/conformity-assessment_en (Accessed: 20 June 2018).

Corp, N. T. P. A. (2015) 'TP4056 1A Standalone Linear Li-lon Battery Charger with Thermal Regulation in SOP-8', *Data Sheet*, p. 3.

Eivy, A. (2017) 'Be Wary of the Economics of "Serverless" Cloud Computing', *IEEE Cloud Computing*, 4(2), pp. 6–12. doi: 10.1109/MCC.2017.32.

Ernst Basler + Partner AG (2018) *Höhenprofile auf Basis von Google Maps, entwickelt von Ernst Basler + Partner AG*. Available at: https://www.geo.ebp.ch/gelaendeprofil/ (Accessed: 25 June

2018).

Frank Feil (2016) *Das Internet der Dinge macht Deutschlands Weinberge smart*. Available at: https://www.techtag.de/it-und-hightech/internet-der-dinge-weinberge/ (Accessed: 24 September 2018).

Gentelella! (2018) *Gentelella Alela!* |. Available at: https://puikinsh.github.io/gentelella/ (Accessed: 21 September 2018).

Giezeman, W. (2018) *BUILDING A FULLY DISTRIBUTED INTERNET OF THINGS DATA INFRASTRUCTURE*. Available at: https://www.thethingsnetwork.org/#index-team.

Gu, P. and Sosale, S. (1999) 'Product modularization for life cycle engineering', *Robotics and Computer-Integrated Manufacturing*. Pergamon, 15(5), pp. 387–401. doi: 10.1016/S0736-5845(99)00049-6.

Hevner, A. and Chatterjee, S. (2010) 'Design Research in Information Systems', 22, pp. 9–23. doi: 10.1007/978-1-4419-5653-8.

Hevner, A. R., March, S. T., Park, J. and Ram, S. (2004) 'Essay in Information Design Science systems', *Management Information Systems*, 28(1), pp. 75–105. doi: 10.2307/25148625.

Hope Microelectronics Co. (2014) 'Datasheet: RFM95/96/97/98(W) v1.0', 98, p. 121. Available at: http://www.hoperf.com/rf_transceiver/lora/RFM95W.html%5Cnhttp://www.hoperf.com/upload/rf/RFM95_96_97_98W.pdf.

Jones, G. V. and Davis, R. E. (2000) 'Climate influences on grapevine phenology, grape composition, and wine production and quality for Bordeaux, France', *American Journal of Enology and Viticulture*, 51(3), pp. 249–261.

Kooijman, M. (2018) *Arduino-LMIC library*. Available at: https://github.com/matthijskooijman/arduino-lmic.

Koundouras, S., Marinos, V., Gkoulioti, A., Kotseridis, Y. and Van Leeuwen, C. (2006) 'Influence of vineyard location and vine water status on fruit maturation of nonirrigated cv. Agiorgitiko (Vitis vinifera L.). Effects on wine phenolic and aroma components', *Journal of Agricultural and Food Chemistry*, 54(14), pp. 5077–5086. doi: 10.1021/jf0605446.

Landolt, D. (2007) *Corrosion and surface chemistry of metals*, *Materials Today*. doi: 10.1016/S1369-7021(07)70081-0.

Lee, E. A. (2000) 'What's ahead for embedded software?', *Computer*, 33(9), pp. 18–26. doi: 10.1109/2.868693.

Manufacturers - European Commission (2018). Available at: https://ec.europa.eu/growth/single-market/ce-marking/manufacturers_en (Accessed: 20 June 2018).

Medela, A., Cendon, B., Gonzalez, L., Crespo, R. and Nevares, I. (2013) 'IoT multiplatform networking to monitor and control wineries and vineyards', *Future Network & MobileSummit*, pp. 1–10.

'Mehrpolige Steckverbindungseinheit für Dreiphasen-Wechselstromsysteme' (2012). Available at: https://patents.google.com/patent/DE102012203459A1/de (Accessed: 1 October 2018).

Mikkola, J. H. (2000) 'Product Architecture Design: Implications for Modularization and Interface Management', *LINK Workshop "Organizing Processes of Learning", Copenhagen Business School, Denmark*, pp. 1–27.

Ministerium für Wirtschaft, Verkehr, L. und W. (2002) *Bewässerung in Weinbergen*, *Landtag Rheinland-Pfalz*.

N-düngebedarfs, E., Stickstoff, W., Rheinland-pfalz, L., Arbeitsausschuss, W., Juni, S., Der, U., Pham, C., Adelantado, F., Vilajosana, X., Tuset-Peiro, P., Martinez, B., Melia-Segui, J., Watteyne, T., Zhao, J. C., Zhang, J. F., Feng, Y., Guo, J. X., Ministerium für Wirtschaft, Verkehr, L. und W., Prior, B., Bestimmungen, A., Kommunikation, E., Genehmigung, I. Der, Hinweis, A., Nr, C., Cross, R., Plantinga, A. J., Stavins, R. N., Kottek, M., Grieser, J., Beck, C., Rudolf, B., Rubel, F., Koundouras, S., Marinos, V., Gkoulioti, A., Kotseridis, Y., Van Leeuwen, C., Ide, M., Amagai, Y., Aoyama, M., Kikushima, Y., Version, P., Team, Summary, E., Saunders, C., Dalziel, P., Wilson, M., McIntyre, T., Collier, H., Kaye-Blake, W., Mowat, A., Olsen, T., Reid, J., Shrouf, F., Ordieres, J., Miragliotta, G., Li, S., Xu, L. Da, Zhao, S., Reifel, J., Pei, A., Lala, S., Hales, M., Blanter, A., Bhardwaj, N., Kuechler, Vaishnavi, V. K., Petter, Hevner, A. R., Ferrández-Pastor, F., García-Chamizo, J., Nieto-Hidalgo, M., Mora-Pascual, J., Mora-Martínez, J., Vaishnavi, V. K., Kuechler, W., Kuechler, B., Hevner, A. R., Chatterjee, S., Bair, L. a., Seelye, M., Gupta, G. Sen, Bailey, D. and Seelye, J. (2017) 'A Three Cycle View of Design Science Research', *Journal of Agricultural and Food Chemistry*, 1(2), pp. 34–40. doi: 10.1007/978-1-4419-5653-8.

Porter, M. E. and Heppelmann, J. E. (2014) 'How Smart, Connected Product Are Transforming Competition', *Harvard Business Review*, (November), pp. 64–89. doi: 10.1017/CBO9781107415324.004.

Restriction of the use of certain hazardous substances (RoHS) - European Commission (2018). Available at: https://ec.europa.eu/growth/single-market/european-standards/harmonised-standards/restriction-of-hazardous-substances_en (Accessed: 20 June 2018).

Rocketscream (2018) *Low Power Library for Arduino*. Available at: https://github.com/rocketscream/Low-Power.

SCOTT (2012) *Secret Arduino Voltmeter – Measure Battery Voltage*. Available at: https://provideyourown.com/2012/secret-arduino-voltmeter-measure-battery-voltage/ (Accessed: 4 July 2018).

Seelye, M., Gupta, G. Sen, Bailey, D. and Seelye, J. (2011) 'Low cost colour sensors for monitoring plant growth in a laboratory', *Conference Record - IEEE Instrumentation and Measurement Technology Conference*, pp. 972–977. doi: 10.1109/IMTC.2011.5944221.

Shrouf, F., Ordieres, J. and Miragliotta, G. (2014) 'Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm', *IEEE International Conference on Industrial Engineering and Engineering Management*, 2015–Janua, pp. 697–701. doi: 10.1109/IEEM.2014.7058728.

Statista (2018) • *Global Internet of Things market size 2009-2019 | Statistic*. Available at: https://www.statista.com/statistics/485136/global-internet-of-things-market-size/ (Accessed: 24 September 2018).

Stock, M. (2006) 'Entwicklung des Klimas und Auswirkungen auf den Weinbau Entwicklung des Klimas und Auswirkungen auf den Weinbau'.

Stokking, J. (2018) *CayenneLPP*. Available at: https://github.com/sabas1080/CayenneLPP.

Tang, K., Wang, Y., Liu, H., Sheng, Y., Wang, X. and Wei, Z. (2013) 'Design and Implementation of Push Notification System Based on the MQTT Protocol', *Proceedings of the 2013 International Conference on Information Science and Computer Applications (ISCA 2013)*, (January 2016). doi: 10.2991/isca-13.2013.20.

The Things Network (2017) *The Things Network | We are building a global open free crowdsourced long range low power IoT data network.* Available at: https://www.thethingsnetwork.org/docs/ (Accessed: 14 November 2017).

Trade Data And Analysis (2014) *WORLD VINEYARD ACREAGE BY COUNTRY*. Available at: https://www.wineinstitute.org/files/World_Vineyard_Acreage_by_Country_2014_cTradeDataAndAnalysis.pdf (Accessed: 24 September 2018).

TTN Mapper (2018). Available at: https://ttnmapper.org/special.php?node=03ffeebb&allnodes=on&date=2017-02-14&gateways=off (Accessed: 25 June 2018).

Weiser, M. (1991) ‘The computer for the 21 st century’, *Scientific american*, 265(September 1991), pp. 94–104. doi: 10.1145/329124.329126.

Wortmann, F. and Flüchter, K. (2015) ‘Internet of Things: Technology and Value Added’, *Business and Information Systems Engineering*, 57(3), pp. 221–224. doi: 10.1007/s12599-015-0383-3.

Zhao, J. C., Zhang, J. F., Feng, Y. and Guo, J. X. (2010) ‘The study and application of the IOT technology in agriculture’, *Proceedings - 2010 3rd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2010*, 2, pp. 462–465. doi: 10.1109/ICCSIT.2010.5565120.

Appendix

Source Code

1. Embedded Software for Arduino: https://gitlab.uni-koblenz.de/FGEIM/eot_ba_thom_smart_vineyard/vinio_arduino_node
2. Client Front-end WordPress Plugin: https://gitlab.uni-koblenz.de/FGEIM/eot_ba_thom_smart_vineyard/vinio_dashboard_wp_plugin
3. Services Layer: https://gitlab.uni-koblenz.de/FGEIM/eot_ba_thom_smart_vineyard/vinio_services_layer

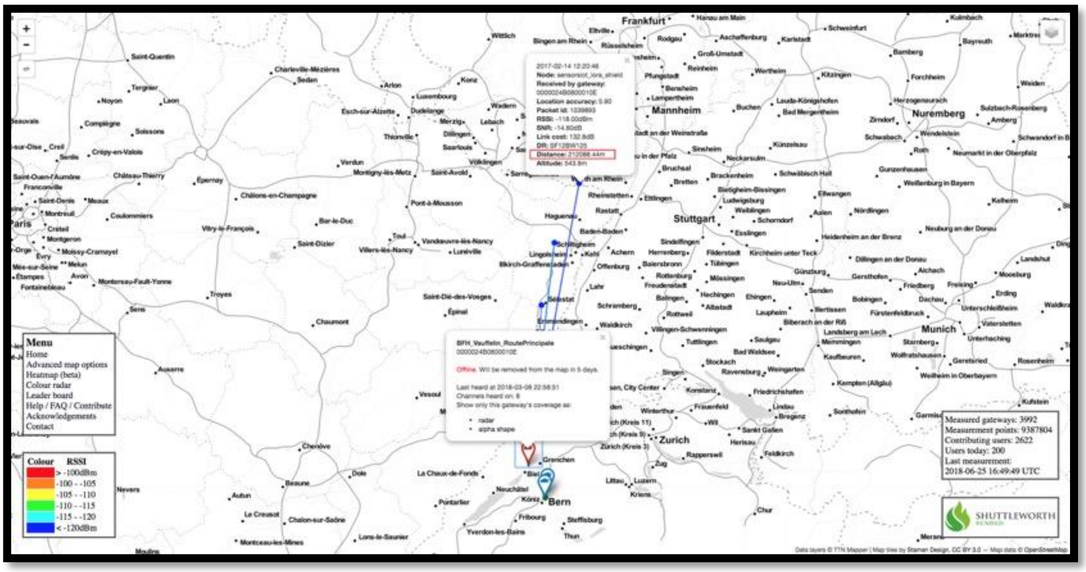
Legal Notice for copyright of Figure 2. 1 and Figure 2. 2.

"As long as the HBR article exhibit is only being used to fulfill the University of Koblenz · Landau class assignment in the pursuit of your degree, permission would be granted at no charge as long as the material is fully cited.

If the thesis is later published in a trade book or distributed as training material, however, then there may be a royalty charge for use of the HBR article material that would be based on how much material is used and the print run/distribution."

Granted 14.06.2018
Permissions Coordinator
HARVARD BUSINESS PUBLISHING
20 Guest St, Suite 700 | Brighton, MA 02135

Prove of 200 km LoRa TTN range test.



Retrieved June 25, 2018, from https://ttnmapper.org/special.php?node=03ffeebb&allnodes=on&date=2017-02-14&gateways=off