UiO : Department of Informatics
University of Oslo

Connecting Humans to Nature Through Technology

Using Internet of Things to sample data and provide it as experiences for humans

Lars Erik Realfsen

Master's Thesis Autumn 2017

Connecting Humans to Nature Through Technology

Lars Erik Realfsen

9th November 2017

Abstract

In nature, everything is connected. In the recent years, researchers have found that plants and trees is connected by a network of fungi, the Wood Wide Web. Nature benefits from this network of contact, and humans can as well, if we can connect to it.

This thesis discusses and explores how we can connect humans to nature through technology, by measuring variables in trees and plants in a cheap, easy and controlled manner. A framework for connecting sensors and sensor hubs to different parts of nature has been provided. We present our choice of sensors and hubs, and how to connect them.

We also examine the possibilities the measurement data provides, by researching opportunities to present the data to a wider audience. By creating a robot that can draw the data to a canvas, we have seen that people get exited by the measurements.

Our research proves that contact between humans and nature is possible through technology. It also suggests that connecting humans to nature by making art with technology is exiting and makes people interested in the subject.

Acknowledgements

First, I would like to thank my supervisor Mats Høvin. He offered me this thesis and provided valuable feedback along the way. He have also opened my eyes for exiting subjects which will follow me long after this thesis is left in the bookshelf. I would never finish without a supervisor with this kind of respect and care for my way of working.

I also have to thank Cybernetisk Selskab for providing not only the best coffee on campus, but also the best buddies one can have in a student society. I will never give up as long as I have good, supportive friends (and a bar).

Contents

List of Figures

List of Tables

Part I

Introduction

1
Introduction

Nature is all around us. We may live in large cities, with months between hikes in the woods, but we still have lots of connections to our natural surroundings. Our food and water come from the nature, and the oxygen we breath is likely a product of photosynthesis. Some of us like to explore the wilderness in days of exciting adventures, while others prefer a walk in the neighbour park. We all have in common that decisions on whether we should build a wall or be better off with a hedge, relies on the knowledge of our surroundings and how it reacts to change.

We have viewed trees and other plants as individual plants with almost no communication between them. Recent research in biology have proven this wrong with the finding of a network of mycorrhizal fungi[5] which carries nutrients, minerals and other types of interaction which remains to be seen. This network has been named "The Wood Wide Web". One of the outcomes of this, is a potential for exiting research in plant communication and, as we have done in this thesis, plant to human communication.

Humans are also a part of nature, but we tend to gather i large cities with seldom contact with the parts of the world who cover our bare necessities. We can observe the complex interactions of the world wood web, but we have yet to transform this data into something we all can experience.

In this thesis, we have deigned a framework for connecting humans to nature, through technology. We start out with data collection by measuring different aspects of a tree, continuing with data conditioning, and end up prototyping a robot to make the data into an experience for humans. The robot is set up to draw data from the tree on a paper, with people standing around watching and giving feedback.

1.1 Motivation

The time scale in nature spans over a vast amount of time, and changes to development of some plants have to be observed over years of studies. A common pine tree in Norway have a life expectancy of 100 to 200 years. The tree in the neighbourhood park may have grown from a small sampling to a fifteen meter tall tree in twenty years. We all understand this, and we can easily observe trees in all of the stages to becoming a fully grown tree, just by walking out in to the woods. Many parts of the nature is connected through the wood wide web, but we have not made much use of this. One of the first steps is to collect measurements, and start a one-way connection to nature through data.

1.2 Goal

The goal of this thesis is to determine how we can make use of readily available sensors to measure different aspects of the life of a plant, and further, use this data to connect humans to what the plant experiences. The aspect we will discuss the most is the presentation of the data. Because of ethical reasons, we have decided that encroachment of nature is unacceptable throughout the whole process. We will also

explore the possibility of presenting the captured sensor data as two different types of interesting experiences. We define "interesting" as an experience which connects humans to nature. This experience should be available to the public in an easy manner, and contribute to the connection to nature.

1.3 Approach

By combining our discussion with the understanding of recent sensory equipment and data collection methods, we can gather relevant data. Sensors used in this thesis is described in detail with explanations on why we selected them. Data captured from the sensors needs conditioning and processing, this is solved by normalising the data and storing it for future processing. The user, or audience, will be provided with an experience based on our data. This is carried out by a drawing-robot which draws the data to a paper. A short user review shows what and how people react to the experience.

1.4 Work Done

We have researched relevant topics in IoT, sensor technology and data presentation. The research resulted in a hardware implementation with the desired outcome of being able to record sensory data from a tree. Furthermore, the data was conditioned and sent to a small variety of presentation methods, one of which was a robot. We got user feedback on the trials with the robot, which helped us with decisions on how further work should be conducted.

All the program source code is available open source on the authors Github page[17].

1.5 Evaluation

The wide array of goals presented here rises some questions to how we are going to evaluate both the work done, and the results. As the process is split up in three parts, the logics of evaluation will follow the logical division. The evaluation was based on the requirements detailed in section 5.1. User review from the drawing-robot experience is also used to evaluate the project.

1.6 Results

The collection of data from nature by sensing with IoT connected sensors, provide a flexible and easy way to attain information. The testing of the system in a closed, in-door situation, enables a fast development pace. However, there are certain limitations to this approach, such as non-natural conditions. A better understanding of the conditions where we measure would contribute to a better measurement.

Viewing the data as drawing preformed by a robot provides a valuable method of presentation. Discussion on how to accomplish this is important to further extend our understanding of the measured data, and how it connects us to the nature.

1.7 Contributions

Our contribution is a framework for measuring various elements in nature, and present them as experiences in which connect humans to nature. We have discussed what to measure, how to measure, and why we measure. The implementation explores two types of sensor hubs, and four different sensors. A presentation of the data was done using physical model and a web experience. The physical model is i fully featured drawing-robot which has been tested at a conference.

1.8 Conclusion

This thesis has shown that collecting data through IoT technology can be easy, and a good way of connecting humans to nature through technology. It has also shown that some of the sensing techniques can be adapted to new types of sensors, and at a larger scale.

1.9 Outline

This thesis is divided into five separate parts: Introduction, Background, Sensor System Implementation, Data Presentation and Conclusion.

Part I: Introduction
Gives a brief introduction to the thesis and the work preformed.

Part II: Background
Provides the background for common concepts used in this thesis, how we intend to use them, earlier work and limitations.

Part III: Sensor System Implementation
Describes how the work was done and how it can be reproduced.

Part IV: Data Presentation
Details how we can use the measured data to make a one-way connection between nature and humans through an interactive experience.

Part V: Conclusion
Presents how the thesis answers the questions set in the goal, and how future work can add to this.

Part II

Background

2
Background

This chapter introduces the approach and our selection of criteria for a solution. It also elaborates on limitations set by the goal and project span. Earlier work is explored to get a better grip on how we can expect to preform, and to find inspiration for our outcome. Work based on this is also easier to evaluate, because we can draw parallels to what others have concluded.

2.1 Limitations

The scope of this thesis is pretty large. From an abstract view of experience and subjective analysis to a real implementation of hardware, we have had to make compromises and limit ourself to a more constrained project scope. As we always try to provide good a description of the development process, some of the limitations will be divided between what we are able to conduct, and what one might expect to find if one choose to reproduce the results.

Equipment availability

While most of the sensors and controllers we use in this thesis can be bought relatively cheap and easy, the most – in our opinion – interesting ones are harder to obtain. Sap flow, nitrogen uptake and precise angular rate are good examples.

Weather conditions

Weather conditions plays a role in the measurements. Rough weather can be fatal to both the sensor hub and the sensors. This is of course dependent on location and test runs. We can do testing on sensors and hubs separately, as done in section 5.5, but the indoor testing set-up will not provide a realistic test-load.

Project time span

The research conducted in this thesis had a time-span of one year, where two months were dedicated to experimentation on plants. As the field of IoT is under rapid development, we had to perform contiguous research on the subject through the whole project. One of the goals for our final implementation is a solution with the capacity to operate autonomous for one month. The final solution has capacity for dealing with rough weather, but we have not performed tests in the more extreme weather conditions. This might break a period of operation.

Power measurement

Hubs and active sensors will always draw some power to keep timers and interrupts in operation. We have tried to measure the power draw of the different set-ups provided in this thesis, but accurate power measurement is hard, and we did not have time to follow through on a large-scale power model. There is a god chance of approximating the static power draw, as this is available in the data sheets.

Evolving requirements

Requirements will change over time, as the research leads to further questions we have to answer. An *agile process* will help us accomplish this.

Reproducibility

Our final set-up should be well documented through the inclusion of diagrams and examples of expected results. *Discussion* and *exemplification* will be easier by introducing a testing phase as soon as possible.

Result clearness

We should always look for results, failure is also a result. This spans from process, through hardware and software implementation, and ends up as measurements provided to humans for a subjective review. *Tables* and *figures* of the process will help us present the results.

2.2 Earlier Work

One of the main objectives of our implementation, is the collection of data. We are considering the use of different types of microcontrollers ans SoCs, and the Arduino is an appealing choice. In the article "An Agricultural Telemetry System Implemented Using an Arduino-Android Interface", Wilton Lim, Hans Kaell Torres, Carlos M. Oppus [13] explores the use of Arduino to wireless telemetry system to be used in the field of agriculture. They successfully implemented a wireless sensor system using Arduino and Bluetooth.

On the sensory end of our project, lies an important connection to nature. We have to get as close as possible to the plant, both in terms of accuracy and diversity of our measurements. When doing measurements on organic life, nutrients plays an important role. In the article

"Sensors and instrumentation to measure Sap flow in small stem plants" by David Morton, H.Ghayvat and S. C. Mukhopadhyay, they successfully use an Arduino to measure sap flow in plants very accurately.

The use of wireless sensor technology in agriculture is growing fast, but as stated in "Wireless sensors in agriculture and food industry—Recent development and future perspective" by Ning Wang, Naiqian Zhang and Maohua Wang[23], we have to be careful as reliability is not at the level of communication by wire.

2.3 Definitions

2.3.1 Internet of Things

In the past ten years, work has been done to provide internet connectivity to all kinds of services. We have been using internet to not only read the news, but now also to do banking, taxes, ordering food and trading other commodities. In the area of Web 2.0, web pages can load data dynamically to provide live-score in sports, and web streams of events on the other side of the world can reach your smart phone in seconds. This opens a great opportunity for real-time sensory information and accumulation of different types of data.

The Internet of Things consists of many different types of devices, all connected to the internet with their own unique identifier. There has been some concerns around the security of these devices, but we have not done any further discussion on this in this thesis. A good review of the security of IoT can be found in the article "A Critical Analysis on the Security Concerns of Internet of Things (IoT)"[22].

2.3.2 Sensor

Professional sensors used in measurement equipment tend to be very expensive. This can be worth it because they provide good documentation and support, as well as a calibration sheet with an accurate description

of the sensor behaviour in various external conditions. The quality of cheap sensors have had a tremendous jump since the beginning of the smartphone revolution. This has made it easier to do rapid prototyping, and the availability of the electronic parts has sky-rocketed.

2.3.3 Sensor Hub

Sensors alone cannot do any good work without a way to sample, transfer and store what they measure. This is where the sensor hub comes in to play. A sensor hub provide a way to interface connected sensors. It can do caching of intermediate data and short-term storage. For many years, this hub was only attainable at a very steep pricing point. Embedded hardware has traditionally been very pricey. After the introduction of Arduino and Raspberry Pi, this has changed. We can now by sensor hubs with good connectivity for a reasonable price at the local electronics store.

2.3.4 Wood Wide Web

Studies have shown that plants communicate with each other through mycorrhizal networks, referred to as wood wide web. This is underground networks of mycorrhizal fungi, connecting plants together so that they can share resources. Plants connected through this network can share water, carbon, nitrogen and other nutrients and minerals. By enabling communication between plants, the wood wide web can help plants fight illness through sharing of resistance [24]. A study have shown that tomato plants connected through fungi can help the colony resistance to infectious diseases.

While this is something researchers all over the globe is interested in, most of the common public do not know about this, even though it is right under our feet. The connection of plants in the wood wide web is interesting to us, because on a higher level, humans are also a part of the nature. Connecting us humans with this information highway would be really interesting and can open us for a wide array of possibilities.

3

Summary

This part have made the necessary steps to limit our project both in terms of constraints set by us, and external constraints provided by the environment. We have also provided some earlier work on the subjects discussed in this thesis, and how they relate to our project. When working with a broad array of topics, as we do in this thesis, we also needed a brief introduction of the topics.

Part III

Sensor System Implementation

4

Criteria

In this chapter, we discuss the different measures we have to consider while implementing a solution to how we collect the data. Some of the discussion addresses the limitation provided in section 5.5, and some are made to tailor the hardware decisions to our goals. Our hardware selection is directly affected by these criteria.

4.1 What We Want to Measure

One of the goals of this project is to support measurement from a diverse set of sensors. We picked sensors from a wide range of different properties, including placement, data type, structure and sample rate. We decided to start out with the following measurements.

- Swaying
- Air temperature

- Humidity
- Soil moisture

This is a selection of common measurements when studying nature, and the addition of swaying will give us a measure with a high sample rate. This will demand local caching at the sensor hub, or a sustained network load with subsequent power demand.

4.2 Weather Conditions

The sensor platform should be able to handle tough weather conditions. Weather predictions for a whole month forward in time is pretty uncertain and we cannot expect it to be stable. As we are doing the tests in Norway, the local weather is what we expect to do the measurements in. We do have good historical data on the expected max/min temperatures, and the amount of rain. We are not designing a commercial product, so bear that in mind.

Data from our local weather forecaster Yr.no, suggest that wind does not fluctuate much through the moth of July, when we are expecting to do our testing. Ater the measurements took place, we have seen that the wind is in fact pretty stable with an average of 2,8 m/s, and 8,3 m/s at the strongest [25].

4.3 Modularity

The sensor hub and platform must support a wide range of sensors. Our pick of sensors is, as mentioned in section 5.1, suited for a variety of tasks and data types. We expect the sensor system to provide us with data of not only different types, but also hold claim of contrastive methods of data transfer. The transfer types reviewed for this thesis is shown in Table 4.1.

Bus	Pins	Features we want
SPI	4+1 (chip select)	Full-duplex, fast
I2C	2	Easily addressable
1-wire	1	Simple

Table 4.1: Sensor connection tabel

4.4 Scalability

Measuring one tree alone is not the final goal of this project. While there is a lot of interesting data from each individual tree we measure, the collection of data from more than one tree is worth a lot. We can start seeing relationships and connections between different parts of the nature. Comparisons between the development of individual trees in the same specie are a different set of data compared to the development of trees of different species. When we add more sensor hubs and make them cooperate, we can call them a mesh network. We try to lay the ground for future work on a mesh of sensor hubs.

To support such a mesh, we need to take into account that all of the sensor hubs must be able to communicate simultaneously, and that the data is available at all parts of the system. We can guarantee this by having all the hubs connected to the internet, with their own unique IP-address. If mobile networking is unavailable at the location, then maybe SMS is an option. The SMS alternative requires some service that are capable of receiving the message. The receiver can be one of the hubs connected to the mobile network, since the selected transport protocol (MQTT) is capable of pushing the messages on to a different topic.

4.5 Ease of Use

As mentioned in section 1.2, the platform should be easy to use for the public. We can split the operation up in the following steps.

1. Deployment on tree / in plant bed

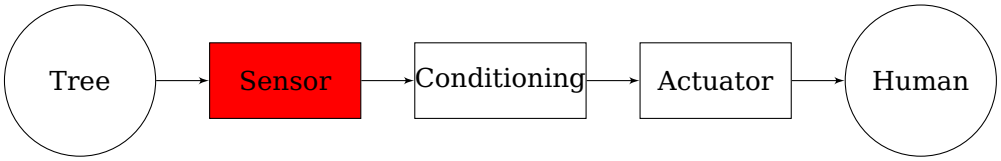
2. Connecting the sensors
3. Adding data collection methods
4. Connecting to server
5. Run test suite
6. Check test results

4.6 Safety

As we are dealing with electricity and energy storage, there are always some concerns to what we must do to take the necessary precautions. We do not want any damage to humans or nature, and it is therefore critical to not use components with poisonous gasses, liquid chemicals or explosive capabilities. All the components used in this thesis is CE rated and are not harmful when handled correctly. Our selected battery is a special case, and might become esxplosive if broken or charged wrong. More information on LiPo safety can be found in the artice "A general discussion of Li ion battery safety" [6]

4.7 Data Involvement

One of the more tricky parts of this thesis is the exploration of how we can enable an experience for humans based on our data. We introduced the concept of exploring a time-line of the life of a plant. A time-line should be able to rewind in time, and play both in real-time and in a given scale. This time-line approach has a lot of potential, but we went for a more physical experience.



5

Hardware Implementation

As explained in section 1.2, the goal of the set-up is to provide a better data foundation to methods of data presentation. We will try to gather data from all parts of the nature, but as this thesis is limited in scope, we will start with focus on a single tree. We will also elaborate on how we can use this for a more comprehensive study, involving more than one tree.

The method presented here will be based on research on new development in IoT technology. We will start out with a rather simple set-up by sending dummy sensor information from different sensor hubs. Sensor hubs are the brain of the sensor system, and we will focus on using small and uncomplicated hubs, because a small hub can provide the necessary connection without using too much power or being too costly. We have explored two types of hubs in detail, the Arduino and the ESP8266. Both of them provide compatibility for an expansive range of sensors, and they are quite similar in GPIO capabilities. We intend to

connect the system as shown in Figure 5.1.

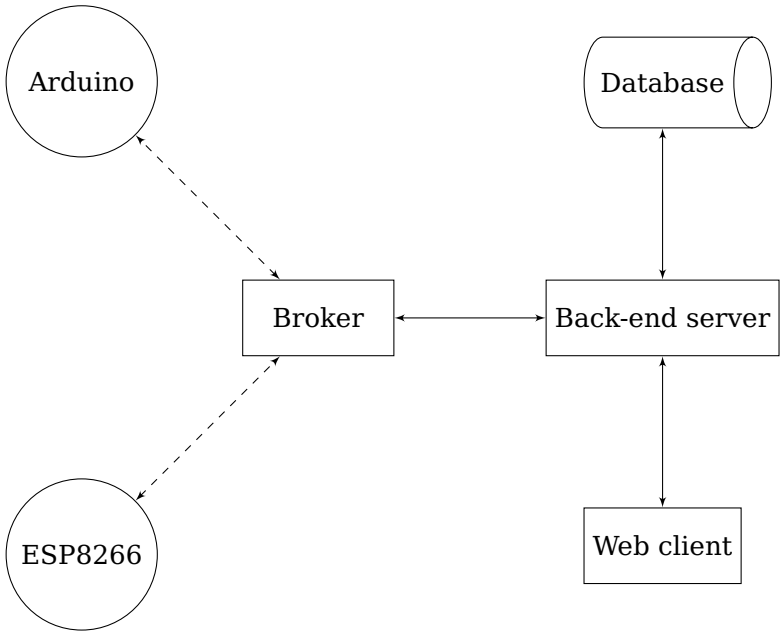


Figure 5.1: System connection diagram

5.1 Sensors

After we have defined what we want to measure, we have to pick our sensors. We have some criteria4 we need to uphold, both hardware and software have to pass our criteria to get picked. Note the limitations in section 2.1.

One of the goals of this project is to provide a framework for a way of understanding the nature from a human perspective. As stated in the goal, we cannot accept encroachment of the trees or in any way damage the nature we seek to protect. Another project goal is to make the system easy testable. Some test requirements have been defied in section 5.5. Testing ensures the data integrity from individual sensors.

Power efficiency have to be taken into account because we are running on battery without any power supply during the operation. We are

not going in-depth on the power subject, but we need to account for it. There has been a lot of good research on this. S. Lindsey and C. S. Raghavendra did a great review of sensor power usage in their article "Power-efficient gathering in sensor information systems"[14]. There is also a good article "Simulating the Power Consumption of Large-scale Sensor Network Applications"[18].

Even though we have decided what transport protocol we want to use, different types of transfer methods must be accounted for. While WiFi may provide great bandwidth and latency, communicating over mobile networks can be both slow and expensive. Thus, our sensor pick must take bandwidth and response time into account.

It is important that our sensors does not interfere with the measurement. We have taken precautions when placing sensors on the tree, so that the sensors don't influence each other. While not in the goal, open source and free hardware and software is always appreciated.

5.1.1 Swaying

To detect swaying we opted for an Adafruit FLORA LSM9DS0 module [20]. Seen in Figure 5.2, we have made some additions to the PCB. To shelter the module from the harsh environment discussed in section 4.2, we have placed the module in a shrink tube and stuffed the ends with glue. We hope this will provide accommodation for the sensitive electronics.

The module is an all-in-one 9-DOF sensor, with the possibility to calibrate for our usage. Since we prefer accuracy over range, as discussed in section 4.2, the sensor has been calibrated to a low range with high accuracy. Our calibrated configuration can be found in Table 5.2.

5.1.2 Soil Moisture

We can measure soil moisture by measuring conductance in the soil between two measuring electrodes. This system will be dependent on

Summary	
Feature	Range of operation
Acceleration	±2/±4/±6/±8/±16 g
Angular rate	±2/±4/±8/±12 gauss
Magnetic field	±245/±500/±2000 dps
Data output resolution	16-bit
Output data-bus	SPI or I2C
Supply voltage	2.4 V to 3.6 V
Temperature range -40°C to +85°C	

Table 5.1: Summary of relevant FLORA LSM9DS0 capabilities

Figure 5.2: Adafruit FLORA LSM9DS0

Feature	Range of operation	Accuracy
Acceleration	2 g	0.061 mg/LSB
Angular rate	245 dps	8.75 mdsp/digit
Magnetic field	2 gauss	0.08 mgauss/LSB

Table 5.2: Our configuration of the LSM9DS0

Figure 5.3: FLORA swaying sensor mounted on tree

soil granularity and composition, thus making calibration important. We bought a cheap sensor from eBay. The sensor works by comparing voltage differences between two electrodes with a known distance. The comparison is done by a Texas Instruments LM393 dual differential comparator[11].

5.1.3 Temperature and Humidity

Temperature is a relatively easy measurement for us. Temperature sensors are readily available and are easy to install. Our chosen sensors is the DHT11 and the DHT22[2]. They are very similar, but DHT22 is more accurate, have a larger range of operation, although more expensive. Note that winter in the north may require the DHT22 (or better) because the DHT11 only measures down to zero degrees celsius.

Our sensor is very basic, but good enough for our usage. If we wanted to make a better reading of humidity on the tree itself, a better solution would be to put a sensor in the bark for direct measurement. An exiting sensor has been developed at the University of Michigan, wich

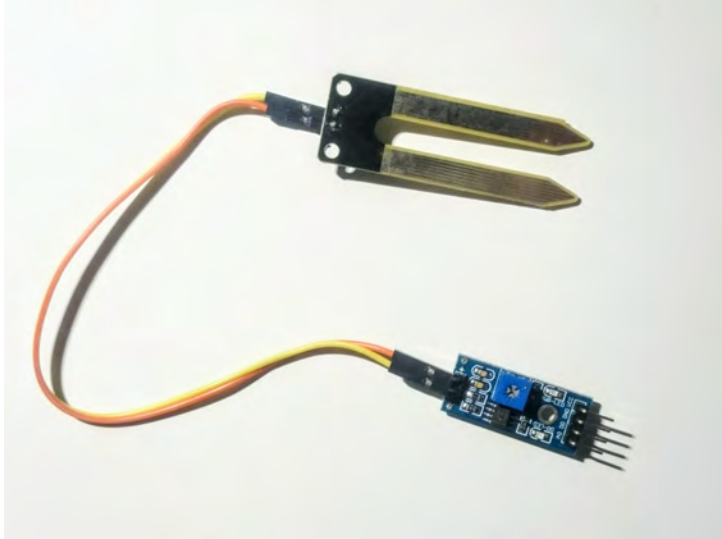


Figure 5.4: LM393 based soil moisture sensor

	DHT11	DHT22
Temperature	0-50°C, ±2°C	-40-125°C, ±0.5°C
Humidity	20- 0%, 5% accuracy	0-100% 2-5% accuracy
Sample rate	1 Hz	0.5 Hz
Max current	2.5mA	
I/O voltage	3-5V	

Table 5.3: Summary of DHT11 sensor

might povide this capability [10].

5.1.4 Sap Flow

Plants, and especially, trees, have to gather water and nutrients from the soil. This is very interesting for our study, and we have tried to measure it. Unfortunately, good pre-made sensors are expensive, and we did not have time to make our own system for measurement. However, a paper by David Morton, H.Ghayvat and S. C. Mukhopadhyay [16] have shown that this can be done easy and with good accuracy.

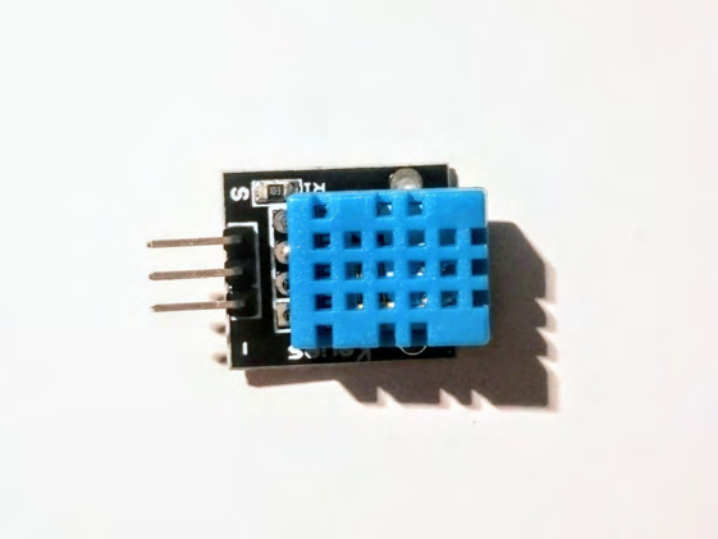


Figure 5.5: DHT11 based temperature and humidity sensor

5.2 Sensor Hub

The understanding of how trees and plants communicate with each other have come a long way in the recent years. As scientists always strive to reach new levels of data quality and quantity, the methods become very expensive. With the progression in smart phones with required sensory equipment, the sensors have had a revolution in pricing. Makers and maker-spaces all over the globe have contributed to the rising availability of what was considered unattainable equipment only a few years back. Sensors available in smart phones can now be bought on the cheap from consumer electronic stores, and this enables us to build a sensor rig of good quality on the cheap.

We have chosen two very similar platforms, the Arduino and the ESP8266. We also did experiments on the Raspberry Pi[9], but it is more expensive, uses more power and is more sensitive to external conditions such as humidity. We did not end up doing any further research on the Raspberry Pi.

Summary	
Microcontroller	ATmega328P[4]
Digital I/O pins	14 (of which 6 provide PWM output)
Analog input pins	6
Clock speed	16 MHz
SRAM	2 KB
Input voltage	7-12 V
Operating temperature	-65°C to +150°C

Table 5.4: Summary of relevant Arduino UNO properties

5.2.1 Arduino

The Arduino platform has proven to be one of the easiest to work with. We have experimented extensively with data gathering using this platform and different ways to collect the data.

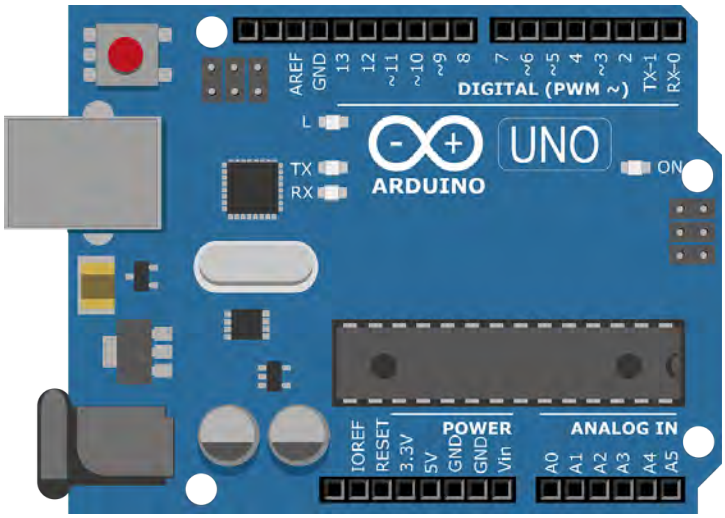


Figure 5.6: The Arduino UNO board / CC BY

We started out with an Arduino Uno and the DHT11 temperature/humidity sensor. The sensor was placed on the household test plant, and the Arduino was running on stable power from a wall-wart. To store the data, we used an SD card mounted on the Arduino using a SD card shield. The data was logged periodically once every five minutes for 24

hours. After 24 hours, we examined the stability of the measurements and concluded that the Arduino platform is ideal for static logging, but that the collection has to be automated.

5.2.2 ESP8266

The ESP8266 is essentially a Wi-Fi controller running on a 80 Mhz RISC CPU. The device is capable of transmitting data over a built-in Wi-Fi antenna, and can be programmed to do I/O on the 16 GPIO (General Purpose Input Output) pins. The ESP8266 package is mounted on a breakout PCB with voltage regulators, pull-up resistors and input-protection. Our test board is the NodeMCU which is FCC approved. The programming is similar to the Arduino and the chip can be flashed using the Arduino IDE (Integrated Development Environment).

Figure 5.7: The ESP8266 board / CC BY

We are using the same DHT11 sensor as previously used on the Arduino, but this time collecting the data over Wi-Fi. To assure a good connection considering the small antenna on the ESP8266, some measurements of the Wi-Fi strength and quality had to be done manually before permanent placement of the equipment. The signal strength/quality measurements was taken using a Sony Xperia Z2 Android smartphone and the Wifi Analyzer[7] app. We checked the signal strength, band-

Summary	
Microcontroller	Tensilica L106 32-bit [21]
Standby power consumption	< 1.0 mW (DTIM3)
Connectivity	WiFi 2.4 GHz 802.11 b/g/n
Input voltage	3-3.6 V
Operating temperature	-40°C to +125°C

Table 5.5: Summary of relevant ESP8266 properties

width, and that no other channels was overlapping our transmission channel. After the setup was complete, we checked the signal again from the ESP8266, and enabled auto-connection in case the connection was unstable. We have not accounted for external EMI (electromagnetic interference) or weather conditions.

5.3 Battery

Our sensors and hubs need power to function. As stated in section 2.1, we have not done any in-depth power optimization of our circuits or code. To accomplish a sufficient longlivety, a rather large energy source have to provide us power. Fortunately, battery technology has come far, and lithium-polymer batteries are both safe and compact. We can use a 3.7V battery without much concern for voltage because some of our parts have regulators and step-up converters already built in. A criteria for the components we pick is safety4.6. LiPo batteries can be dangerous without some precautions, like charging them with a good charger and handling them with care. We bought a 5V USB LiPo charger 5.9 which we configured to 500 mAh charging. This result in safe, but also slow charging. More information on LiPo safety can be found in the artice "A general discussion of Li ion battery safety" [6]

Figure 5.8: 2200mAh, 3.7V LiPo battery

5.4 Encapsulation

To withstand the harsh weather conditions, the hardware needs to be encapsulated in some water resistant container. After an accident where a sensor-hub drowned and shorted, we have chosen to play it safe with a simple water resistant storage box. Our container is a lunchbox with water proofing, and a hot-snoted hole for cables in and out to the sensors.

5.5 Hardware Testing

Selecting equipment for the system usually starts with internet research, but since we have some physical requirements, we also need to take a good look at the product and do testing in a friendly environment before strapping it to a tree in cold weather. Building a test environment can be done very fast, but we also need to consider what we are missing out on. Our testing methodology will start with recognition of the properties we need. This will be based on the sensor requirements found in section 4.1.

Figure 5.9: LiPo charger

5.5.1 Integration Test

As discussed in section 5.1, we use a variety of sensors for obtaining a diverse selection of data types. They all have to be tested under some of the various conditions in section 4.2.

To simplify the testing, we have done integration tests inside on a test plant. The test plant was a basil plant which consumes lots of water. This makes it easy to get basic measurements fast. The sensor integration tests was preformed on the Arduino platform, with simple code to minimize sources of error. The code running the soil moisture tests can be seen in Figure 5.11. When testing the network connection from the ESP8266, the code in appendix section .1 was used, connected as shown in Figure 5.13. A picture of the test plant in use can be seen in Figure 5.12.

Figure 5.10: Luchbox used for enclosing the electronics

```
const int analogInPin = A0;
int sensorValue = 0;

void setup() {
 Serial.begin(9600);
}

void loop() {

 sensorValue = analogRead(analogInPin);
 Serial.println(sensorValue);

 delay(500);
}
```

Figure 5.11: Test plant code for soil moisture reading

Figure 5.12: Watering the test plant while capturing data

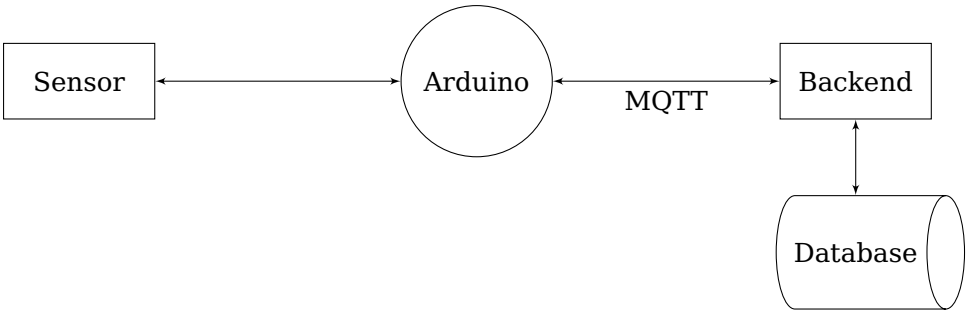


Figure 5.13: Test plant block diagram

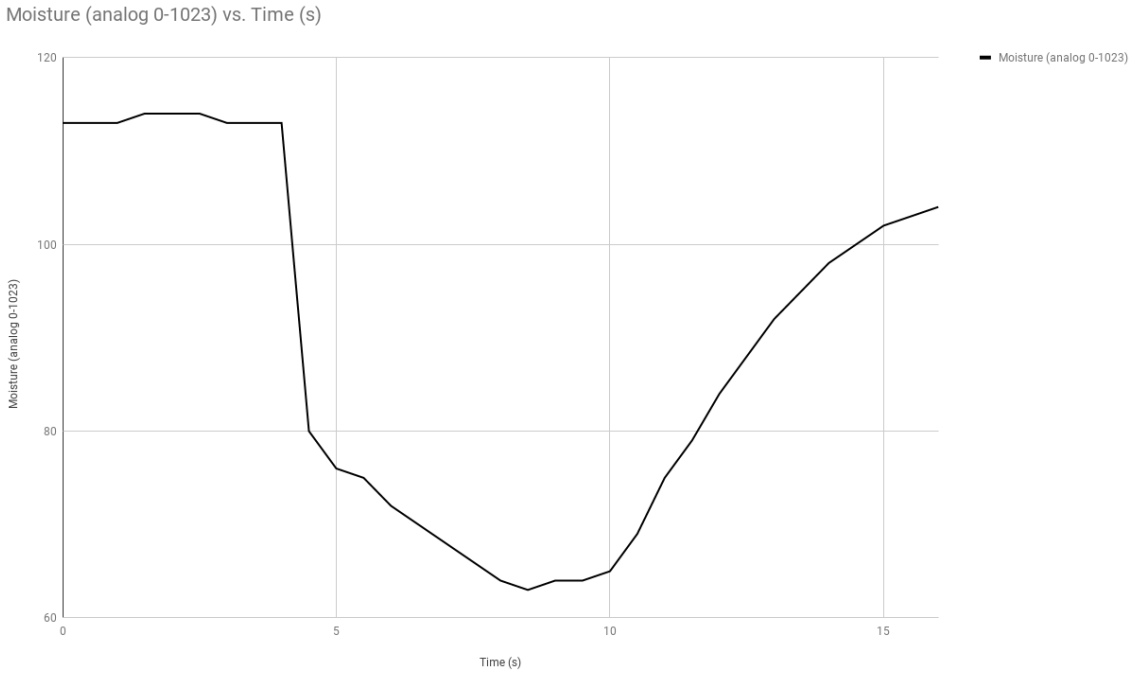


Figure 5.14: Data from basil plat before and after watering (soil dryness vs. time)

6

Software Implementation

Hardware is a large part of this project, but it is worthless without software. Our hardware will work in an infrastructure made up of different configurations and implementations. It is very important to guarantee compatibility between devices, and to use stable, standardised platforms. We have chosen not to go in-depth on the raw data from our picked sensors, as this may vary a lot from other brands and types. However, we detail how the data is processed and stored.

6.1 Data transport protocol

There are several transport protocols to consider. We have boiled it down to the most used protocols, which are all stable with good backing from both industry and community.

- AMQP (Advanced Message Queuing Protocol)

- MQTT (Message Queue Telemetry Transport)
- STOMP (Simple/Streaming Text Oriented Messaging Protocol)

We are using the MQTT (Message Queue Telemetry Transport) protocol for communication between the sensor hub and the server. The choice of MQTT comes from its small footprint and simple implementation. MQTT only needs five methods to provide all the features we need. The protocol is based on the Publish–subscribe pattern, where the sensor hubs publish messages on selected channels, and the server subscribes to relevant channels. It is also possible for the sensor hubs to subscribe to channels. This might come in handy if we later on want to trigger an event on a sensor hub based on data from another hub. We have not explored this, as we don't have a good use case, and power management might become tricky.

The publish/subscribe pattern and its use in MQTT has been discussed in detail by A. J. Stanford-Clark and G. R. Wightwick [19]. They found the pattern to be very useful in the integration of real-time control systems.

The channels is managed by a message broker. The broker is a machine responsible for the inter-communication, and a central gateway between the devices in the network.

6.2 Back-end

6.2.1 Web server

After a short review of different web server back-ends, we ended up with Node[8] as our platform. This choice was based on the fact that our chosen MQTT platform is good supported, and that the server works on nearly all platforms. It is also practical that the server is open source, so that we can modify it if required in the future. We have set up Node server on a light Ubuntu server instance. This is where we run the Node application that receives the data from the sensor system and

```
var redis = require('redis');
var client = redis.createClient();

app.get('/sensor/:name', (request, response) => {
  var sensorHandle = req.params.sensor.name;

  client.get(sensorHandle, function(error, result) {
    res.send({ "sensorValue": result, "source": "redis cache" });
  });

});
```

Figure 6.1: Node server code for reading a sensor value from Redis

saves it in a data base. To make it easy for beginners, the software is very hardware-agnostic and light-weight, and can be run on any platform that supports Node (Linux, Windows, Mac OS). Though, Linux is the only platform we did the full testing and development on.

7

Data Conditioning

On the back end of the data pipeline, the data alone is in a raw condition and we can plot it on graphs to show how the plants behaves over time. The interesting parts lies in the understanding of relations and correlations between different types of data. We have not put this idea to the test, but described it in chapter 15.

7.1 Normalisation

To attain usable data from the sensors, we must condition it first. The data is saved in a raw format in the data base, but we need to normalize it for better usage. Our first goal is to bring the data to a 3D application. In 3D graphics it is common to bring all values into the range

$$X' = [0, 1]$$

This can be done with feature scaling. And we implemented the scaling for X' as

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

3D graphics in not the only presentation method we explore. Other usages such as physical ones discussed in chapter chapter 12 might be candidates for other data ranges. This is archived with

$$X' = a + \frac{(X - X_{min}) - (b - a)}{X_{max} - X_{min}}$$

where the resulting data point X' will be in range of the arbitrary points a and b.

7.2 Storage

After some basic normalisation, the data is ready for storage. Testing is done with Redis[12], which is an open source in-memory data structure store. Redis is chosen because it is the de-facto in-memory database at the time of writing, and it is good supported on all platforms. Redis can be used as a cache, a message broker or, in our case, a fast in-memory database. Our Redis database works by storing values by keys. A key in our database has the following format

$$"sensorhub : sensor : datetime : sample"$$

Where $sensorhub$ refers to the physical MAC address of the sensor-hub and $datetime$ is the Unix time of the $sensorhub$ at the time of measurement. If the measurement requires a collection of samples, the sampler counter will be used. Collections of samples might come from a swaying measurement. The $sensor$ parameter is i bit special. It is used for a description of what type of sensor the data comes from, and has to be pre-defined at the sensor-hub level.

As Redis is an in-memory database, the data will be stored in an volatile location, which means it will be lost on power outage. For a more persistent storage, a non-volatile database has to be used. For testing persistent storage, an instance of MongoDB[15] was used. MongoDB

was chosen as it supports our file formats out of the box and it supports migration from Redis in an easy manner. We connected MongoDB to our back end, and stored it in the same way as Redis.

Note that we don't optimize our database for performance. This might be required if the system scale large enough, or a lot of web clients requests data simultaneously.

7.3 Output Formats

The data are ready to be served to the world when it is conditioned to suit different types of output. We can read it directly from the database by using a web API. This is flexible and the API can be tailored to provide real-time data from a web-socket. We have not implemented web-sockets in this thesis, but it should not be very difficult since the rest of our data flow is optimized for low latency.

8

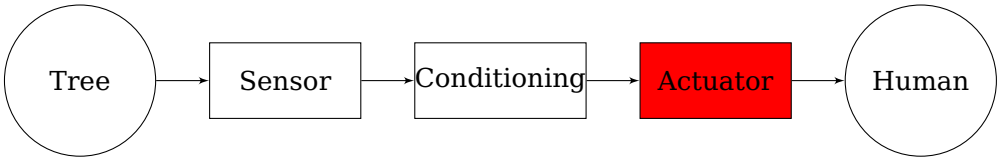
Summary

We have discussed what we want to measure, and how to do it. The part also sets some criteria to the implementation of both hardware and software. We have detailed the sensors we use, and one we did not end up using. The implementation of hardware and software is described, and also what we do with the data.

All of our chosen sensors is capable of doing an accurate reading of what we want to measure and they are also fulfilling our goal of making it easy and non-encroaching. The sensors hubs have both proven to be easy to work with, mostly because of the great communities on the web. Finding examples and prototyping software solutions is very easy with both these platforms. They are also extensible and scalable, especially the Arduino with all of its GPIO capabilities.

Part IV

Data Presentation



9

Creating an experience

So far, we have collected large amount of data. Additionally, preparations for further usage has been performed. Our preparations include dynamic adjustments to the data, of which adjustments can be fed back to the conditioning process. This chapter discusses how we can adapt the data to serve as an experience for humans. We have explored two types of data representation, one on the web, and one physical. The difference between these methods are manifested in the way we can interact with the data. Physical formats can be difficult to adapt to the wide variety of data types, while a virtual representation can be adapted to whatever the data we want. Another difference lies in the availability of the service, where the web experience has a far better reach. Note that we had to limit this to a small review without any real scientific value. However, we requested feedback from the people interacting with the physical model. This kind of study can be valuable because it helps us shape the experience.

10 Criteria

The broad goal of this project is to make a connection between humans ans nature through technology. To limit to what we have time to do in this thesis, the human-nature connection has been defined as a one-way connection with the direction of nature to human. This criterion alone is not enough to constrain us down to a small enough workload. This chapter sets further criteria to our solution.

10.1 Engaging

While we have limited our project to a one-way communication, the *data* can be used in an interactive demo. We have not conducted a large-scale experiment, and the limited resources binds us to make experiences which will engage to play with the data. A method of creating engagement is to make the experience *interactive*.

10.2 Ease of use

Our sensor system has the criteria of ease of use, this is important because we want to make the system available in an easy manner. The same goes for the experiences. With limited resources, the robot did not end up as a product ready for the market, bur rather as a system for exploring the possibilities.

10.3 Available for All Ages

The age of our audience have never been discussed so far in this thesis. We don't think that it's necessary to put age constrains to our experiences. One of the first thing we learn is to draw, and we have exploited that in our drawing robot.

11

Web Experience

Web technology has come far in the recent years, and we can now make use of technologies such as WebGL to view data in 3D inside a web-browser. Since we already have all the data collected and stored within a database, we have explored the possibility of using modern web platforms to offer the data to a broad audience. This is supported in all modern browsers both on desktop and mobile.

The tree shown in Figure 11.1 is drawn with the ThreeJS library in JavaScript. The tree colour of the tree is dependent on temperature, where it goes from blue to red on low to high temperatures. It will also change shape, when the wind is blowing, and the sensors pick up the swaying.

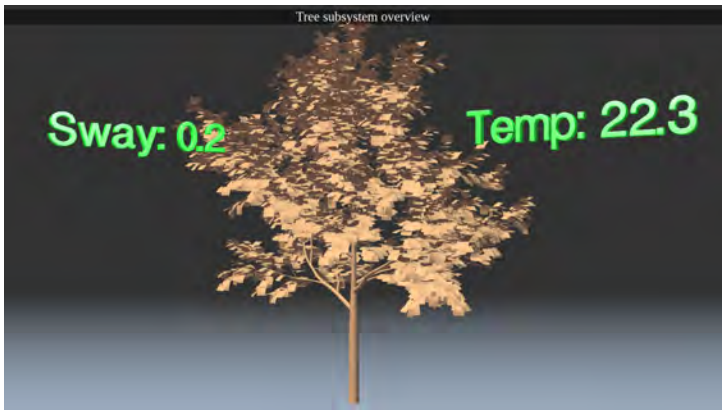


Figure 11.1: Tree drawn in a web browser, showing temperature with color and swaying with displacement.

12

Physical Experience

To further explore how the data can be brought to life, we set out to investigate how we can use physical representation on an analogue surface. The analogue surface paves the way for even more exploration, but we have limited ours to the drawing of geometrical shapes. This analogue approach to the data will be a key step in further investigation because it creates a closeness to the data as never seen before in this project.

12.1 Implementation

In May of 2017, Henie Onstad Kunstsenter approached us, asking for a collaboration on a project to create an experience for children at the summer school of art. The background for the project is an art exhibition by Carsten Höller called "Henie Onstad Sanatorium", where one of the main attractions is two robot beds where you can sleep at night and at the same time be transported around in the exhibition. In the base-

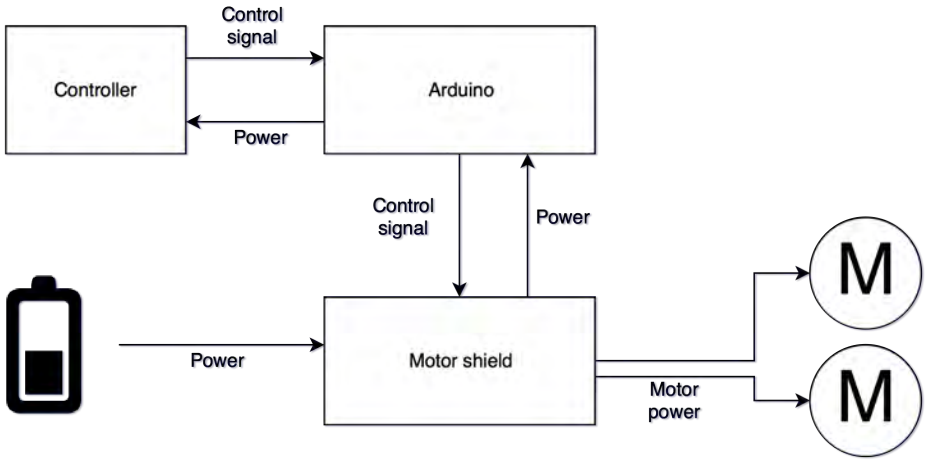


Figure 12.1: Drawing robot block diagram

ment, a laboratory for children were set up so that the children could play with the ideas of the exhibition. Our role came in as the children where to build a small model of the robot bed. We collaborated on making this bed move and draw with selected colors at a paper. The robot where used successfully throughout the summer school, and we got positive feedback on the execution and the learning outcome. The children learned about algorithms and geometry in a way that is applicable to other types of data as well.

12.2 Construction

The robot frame was built using plywood and glue. As mentioned, the robot was built as a part of a project at Henie Onstad Kunstsenter. At fist, we prototyped the robot with DC engines and a control circuit with transistors and a varistor for limiting the current. This proved to be a difficult build with few options for control. After some thought and a prototyping process, we ended up with the configuration in Figure 12.1.

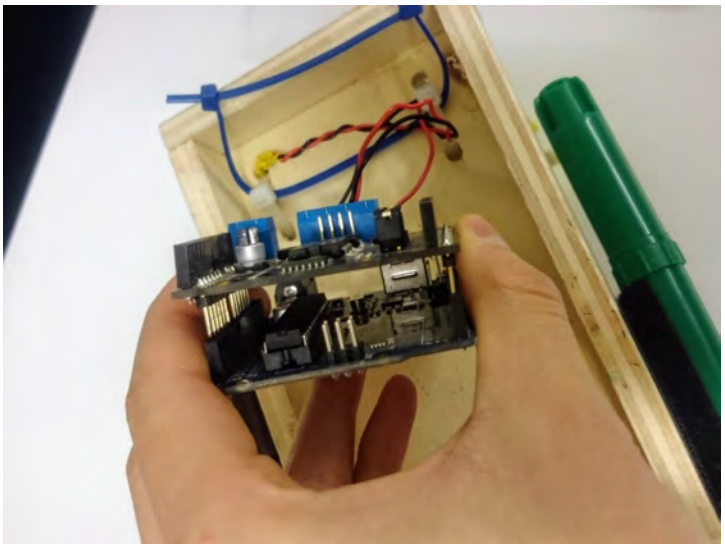


Figure 12.2: Arduino microcontroller with the power delivery shield

12.2.1 Microcontroller and Power Delivery

The final solution uses the aforementioned5.2.1 micro-controller which facilitate circuits for power delivery and signal I/O. We connected the Arduino to a motor control "shield"[3]. This permits fine control over two DC motors which drives the two wheels through a gearing mechanism. We have not connected any wireless connectivity, but that is possible with a WiFi add-on to the Arduino. The Arduino with the motor control shield can be seen in Figure 12.2.

12.2.2 Manual Control Board

The robot have to be simple to turn on and off, and We went for a design that is easy to maintain with few components. Two potentiometers is enough for this, and the Arduino can measure the input-resistance from each potentiometer with ease. One of the potentiometers control start/stop/speed, and the other can be programmed to make changes to the drawing. We have used it to scale the drawing so we can draw on small or large canvas. The controller can be seen in Figure 12.3.

Figure 12.3: Robot controller with two potentiometers

12.2.3 Crayon Attachment

The robot have drawing capabilities by attaching crayons to the outside front of the robot. The crayons are attached by Velcro straps. This can be seen at the front rim of the robot in Figure 12.5. As the crayons is attachable under operation, the uses can decide color and placement of the crayon(s). This provides some engagement and interactivity with the data.

12.3 Presentation

Experience from this collaboration were later used with the data from our sensors placed in a tree, reacting to swaying. At a maker-fair atsnd at the Netlight EDGE Conference in Helsinki 15th of October, the robot was fed with this swaying-data, and people could view the swaying of a tree drawn on paper. The feedback from the users was good and got summed up in keywords like "fun, intriguing, unexpected". The robot in operation at the conference can be seen in Figure 12.6.

A pattern created by the robot is shown in Figure 12.7. The pattern

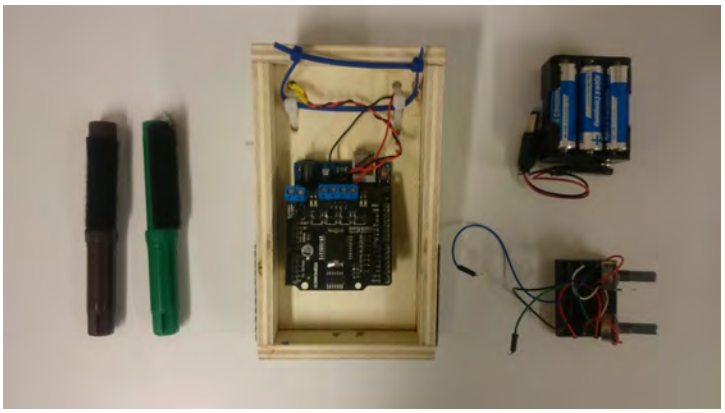


Figure 12.4: The drawing robot disassembled

can be adjusted to the size of the canvas, by narrowing the loop of the robot. This is controlled by one of the potentiometers, which controls the speed of the inner wheel. If both of the wheels run at the same speed, the pattern can be drawn as a straight graph.

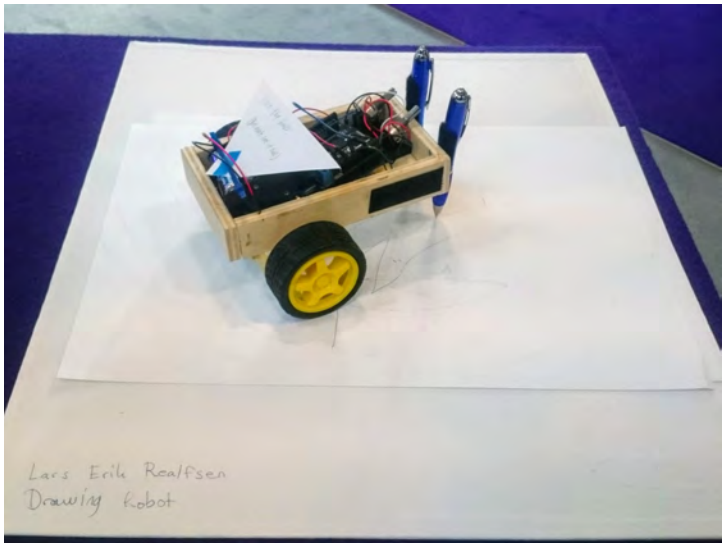


Figure 12.5: Robot used to draw the tree swaying

Figure 12.6: Robot showing off its drawing capabilities at the Netlight EDGE Conference in Helsinki

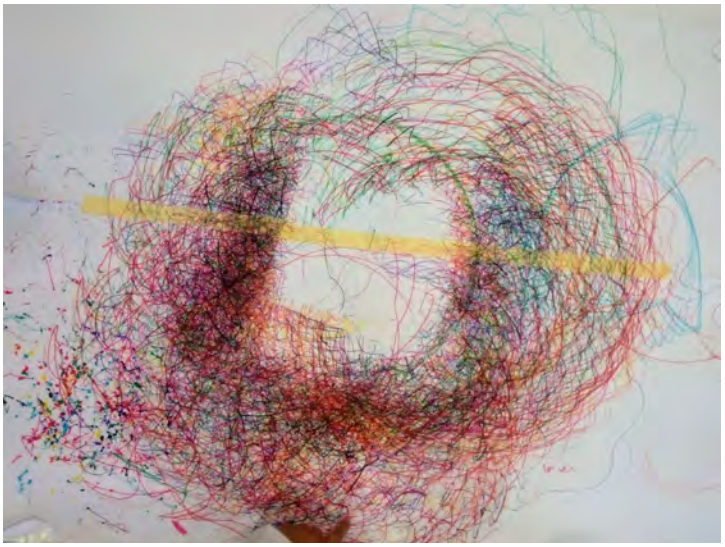


Figure 12.7: Outline of the swaying pattern drawn by the robot

13

Summary

After a discussion on how we want to create an experience, we have come up with some criteria to how we want to execute the project. The experience should be *engaging*, *easy to use* and *available for all ages*.

After the criteria was set, we ended up with two experiences to show data from the sensor system in a way that connects humans to nature. A robot and a web experience can do this. The web experience was built to show data from a tree on a 3D tree in a web browser. The robot was built to draw the data on to a canvas of paper. The web experience was not tested in public, but the robot was. The robot fulfilled all of our criteria and proved to be a useful tool for showing our data and connect to the movements of the tree.

Part V

Conclusion

14

Conclusion

14.1 Summary

In this thesis we have discussed how humans can connect to nature through technology. We have introduces technologies which can enable us to connect sensors to trees and gather data from measurements done on them. Data from the measurements where normalized and sent to a storage. The normalized data was further used in a robot we built to draw the data on paper. A small study on how humans react when watching the robot was conducted.

14.2 Contributions

Our contribution is a framework for a better connection between humans and nature. This framework is composed of a study of sensors, how to connect them and how to store the data. The sensors was well-

tested in various conditions, and we provided their limitations. A brief discussion on our technology-stack was given, with explanations of our final selection.

We have described how we conditioned the data, and how we manage to use it in an experience. We also did a briefing on how we can use the collected data to make a human-nature connection. A robot was created and connected with the data from our sensor-framework. The robot drew the sensor data on to a paper and a human trial provided useful feedback.

15

Future Work

This thesis has many loose ends with opportunities in expansion and elaboration of the work done. In addition, possibilities to use the research for whole other types of projects is in reach. This chapter discusses how we can use the knowledge and outcome from this thesis to enable future work with the methods provided here, and also how the data from our system can enable new types of experiences.

15.1 End-user Experience

Data and discussion in this thesis can be used for more exploration in the field of informatics, but it can also extend into other fields. In chapter 12, we made a end-user experience based on the data from our system. This is only one of forest of possibilities, and what an artist would make out of this would be exiting to see. For this to happen, either the data or the system must be easy available to the public, and this can be done with our system, as it uses open software and readily available hardware.

15.1.1 Time

As time goes by, the perspective of time might become skewed by our experience. Maybe we should stop for a moment and reflect on our own perspective, as a change of perspective can result in a better understanding of the subject. This sudden change of perspective is often referred to in pop-culture. One of the most iconic songs of the 20th century is the song "Time" by Pink Floyd, where the passage of time is the central theme.

The passing of time and growth of a tree can be explained by showing pictures or video, but the experience always lacks some degree of interactivity. Images and video usually capture the state of the plant in a small period of time, and that is understandable as this kind of information uses a lot of resources in terms of storage and bandwidth. If instead capture the state of the tree with sensor data, we can save large amounts of data usage, and get the added bonus of more information. This information can be used as a basis to do data science and statistical reviews of, for example, health and human influence.

15.2 Expanding our System

Our system is limited in scope and we have not had time to do a full one-month study from an autonomous system. We have, however, aimed at modularity4.3 and scalability4.4. Whether the Arduino or the ESP8266 sensor hub is used, we can guarantee that a wide range of sensors can be connected. We have also found that the good community support of all the hardware and software used in this thesis has helped a lot.

15.2.1 Connecting Through new IoT Technologies

The ability to send and receive information relies on a connection to a server. The growth of mobile networks give us the opportunity to send and receive information from places unreachable by cable, and without expensive satellite connections. LTE, or the beginning of the

fourth generation of mobile networks, gave us a fast connection with a good distance of operation. This is good for internet of things, but as devices grow in number, the base stations must follow with capacity for more simultaneous connections.

A light bulb which can be automated and controlled from a smart phone can be bought from most of the consumer electronic stores, and smart devices tend to be the flagship devices of their product category. This is an argument for the mobile carriers to expand their mobile networks. New types of mobile networks are under rapid development, and we will start seeing support for them in flagship smart phones. The Samsung Galaxy S8 already have support for the GB speed class of LTE.

While speed clearly is important for wide-band streaming of multimedia, simple sensor data from a temperature sensor is not very bandwidth consuming. Some sensors do crave bandwidth for streaming of high frequency data - for example sonic measurements - but most of the sensors is in place for convenient measurements whilst using low bandwidth. What we need are simultaneous connections for the majority of the low-power and also low-bandwidth devices.

The METIS-II[1] flagship EU project for the future of mobile networking, namely 5G, have taken IoT into account when designing the requirements for the 5G network. They specify ten times longer battery life for low-powered M2M communication, and ten to a hundred times higher number of connected devices.

These new methods of connection will play a big role in the coming of IoT. This can be leveraged for our network of connections in nature by using more and smaller devices, with smaller energy requirements. The maker revolution takes place, kids learn to code, and the common population become more technology trained.

Appendices

.1 Sensor Code

```
/*
 * This scetch shows how the ESP8266 can be used for reading data
 * form analog input, and send it to a web server over WiFi.
*/

#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>

// Server port
ESP8266WebServer server(80);

// WiFi credentials
const char* ssid = "XXXXX";
const char* password = "XXXXXX";

void setup() {

   Serial.begin(115200);
   WiFi.begin(ssid, password);

   while (WiFi.status() != WL_CONNECTED) {

      delay(500);
      Serial.println("Waiting to connect...");

   }

   Serial.print("IP address: ");
   Serial.println(WiFi.localIP());

   // If called for data, return a reading from the sensor
   server.on("/data", data);

   server.begin();
   Serial.println("Server listening");

}

void loop() {
   server.handleClient();
```

```
}

// Reading the analog input on pin A0
// Note that pin A0 only accepts 0-1V to ADC
void data() {
    String message = "data: ";
         message += analogRead(A0);
         message += "\n";

    server.send(200, "text/plain", message);
    Serial.println(message);
}
```

.2 Arduino robot code

```
/* This Arduni code will make the robot follow
 * manually entered data points.
 */

int E1 = 5;
int M1 = 4;
int E2 = 6;
int M2 = 7;

// Manual data input. All datapoints must be
// normalized to 0-1024 range.
const int data[] = {103, 106, 110, 140, 252, 101, 93, 68};
const int data_length = 8;

void setup()
{
  pinMode(M1, OUTPUT);
  pinMode(M2, OUTPUT);
}

void loop()
{
  int M1_in = analogRead(A1);
  int M2_in = analogRead(A2);
```

```
  if(M1_in < 70) {
    delay(100);
    return;
  }

  for(int i = 0; i < data_length; i++){
    int M1_speed = map(data[i], 0, 1024, 0, 255);
    int M2_speed = M2_in;

    digitalWrite(M1, HIGH);
    analogWrite(E1, M1_speed);

    digitalWrite(M2, HIGH);
    analogWrite(E2, M2_speed);

    delay(1000);
  }

}
```

Bibliography

[1] European Union 5G-PPP. *METIS-II*. 2015. URL: https://www.metis2020.com/ (visited on 04/10/2017).

[2] Ltd Aosong(Guangzhou) Electronics Co. *Digital-output relative humidity and temperature sensor/module AM2303*. URL: https://cdn-shop.adafruit.com/datasheets/DHT22.pdf (visited on 24/10/2017).

[3] Arduino. *Arduino website*. 2017. URL: https://store.arduino.cc/arduino-motor-shield-rev3 (visited on 01/11/2017).

[4] Atmel. *Atmel® picoPower® ATmega328/P DATASHEET COMPLETE*. 2016. URL: http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf (visited on 08/10/2017).

[5] N. S. Bolan. "A critical review on the role of mycorrhizal fungi in the uptake of phosphorus by plants". In: *Plant and Soil* 134.2 (July 1991), pp. 189–207. ISSN: 1573-5036. DOI: 10.1007/BF00012037. URL: https://doi.org/10.1007/BF00012037.

[6] Daniel H Doughty and E Peter Roth. "A general discussion of Li ion battery safety". In: *The Electrochemical Society Interface* 21.2 (2012), pp. 37–44.

[7] farproc. *Wifi Analyzer*. 2017. URL: https://play.google.com/store/apps/details?id=com.farproc.wifi.analyzer (visited on 04/10/2017).

[8] Linux Foundation. *Node.js website*. 2017. URL: https://nodejs.org/en/ (visited on 31/10/2017).

[9] RASPBERRY PI FOUNDATION. *Raspberry Pi website*. 2017. URL: https://www.raspberrypi.org/ (visited on 01/11/2017).

[10] Timothy J Harpster, Brian Stark and Khalil Najafi. “A passive wireless integrated humidity sensor”. In: *Sensors and Actuators A: Physical* 95.2 (2002). Papers from the Proceedings of the 14th IEEE Internat. Conf. on MicroElectroMechanical Systems, pp. 100–107. ISSN: 0924-4247. DOI: https://doi.org/10.1016/S0924-4247(01)00720-8. URL: http://www.sciencedirect.com/science/article/pii/S0924424701007208.

[11] Texas Instruments. *LMx93-N, LM2903-N Low-Power, Low-Offset Voltage, Dual Comparators*. 2014. URL: http://www.ti.com/lit/ds/symlink/lm393-n.pdf (visited on 06/10/2017).

[12] Redis Labs. *Redis website*. 2017. URL: https://redis.io/ (visited on 31/10/2017).

[13] W. Lim, H. K. Torres and C. M. Oppus. “An agricultural telemetry system implemented using an Arduino-Android interface”. In: *2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*. Nov. 2014, pp. 1–6. DOI: 10.1109/HNICEM.2014.7016199.

[14] S. Lindsey and C. S. Raghavendra. “PEGASIS: Power-efficient gathering in sensor information systems”. In: *Proceedings, IEEE Aerospace Conference*. Vol. 3. 2002, 3-1125-3-1130 vol.3. DOI: 10.1109/AERO.2002.1035242.

[15] Inc MongoDB. *MongoDB website*. 2017. URL: https://www.mongodb.com/ (visited on 31/10/2017).

[16] D. Morton et al. “Sensors and instrumentation to measure Sap flow in small stem plants”. In: *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*. May 2016, pp. 1–6. DOI: 10.1109/I2MTC.2016.7520519.

[17] Lars Erik Realfsen. *Sensor framework code*. 2017. URL: https://github.com/Larsera/Sensors (visited on 31/10/2017).

[18] Victor Shnayder et al. “Simulating the Power Consumption of Large-scale Sensor Network Applications”. In: *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*. SenSys ’04. Baltimore, MD, USA: ACM, 2004, pp. 188–200. ISBN: 1-58113-879-2. DOI: 10.1145/1031495.1031518. URL: http://doi.acm.org/10.1145/1031495.1031518.

[19] A. J. Stanford-Clark and G. R. Wightwick. "The application of publish/subscribe messaging to environmental, monitoring, and control systems". In: *IBM Journal of Research and Development* 54.4 (July 2010), pp. 1–7. ISSN: 0018-8646. DOI: 10.1147/JRD.2010.2050982.

[20] STMicroelectronics. *LSM9DS0 - iNEMO inertial module: 3D accelerometer, 3D gyroscope, 3D magnetometer*. 2013. URL: https://cdn-shop.adafruit.com/datasheets/LSM9DS0.pdf (visited on 04/10/2017).

[21] Espressif Systems. *ESP8266EX Datasheet*. 2015. URL: https://cdn-shop.adafruit.com/product-files/2471/0A-ESP8266__Datasheet__EN_v4.3.pdf (visited on 08/10/2017).

[22] M UFarooq et al. "A Critical Analysis on the Security Concerns of Internet of Things (IoT)". In: *International Journal of Computer Applications* 111.7 (2015), pp. 1–6. ISSN: 09758887.

[23] Ning Wang, Naiqian Zhang and Maohua Wang. "Wireless sensors in agriculture and food industry—Recent development and future perspective". In: *Computers and Electronics in Agriculture* 50.1 (2006), pp. 1–14. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2005.09.003. URL: http://www.sciencedirect.com/science/article/pii/S0168169905001572.

[24] L. J. Xie et al. "[Disease resistance signal transfer between roots of different tomato plants through common arbuscular mycorrhiza networks]". In: *Ying Yong Sheng Tai Xue Bao* 23.5 (May 2012), pp. 1145–1152.

[25] Yr. *Yr weatherstatistics*. 2017. URL: https://www.yr.no/place/Norway/Oslo/Oslo/Oslo/statistics.html?spr=eng (visited on 31/10/2017).