IT Technology
Semester 2 Project Report
Team B3

University College

Authors
Aubrey Sebastian Jones - asjo28903@edu.ucl.dk
Henrik Holm Hansen – hhha28796@edu.ucl.dk
Thobias Jon Selmann – tjse28878@edu.ucl.dk
Aleksandra Voronina – alvo28899@edu.ucl.dk
Gladys Waithera - gwma28853@edu.ucl.dk
Lukasz Zwak – luzw28872@edu.ucl.dk
Dainty Olsen – dlbo28887@edu.ucl.dk

June 2nd, 2021

Characters: 19632.

Table of Contents

1 Introduction

This is the Final Report on the group B3 semester 2 project, a culmination of everything that we have learned over the entirety of the second semester.
This project pulled topics from Embedded Systems, Electronics, Networking as well as the various things that the team learned while working in collaboration with the OME's while working on the assignment for Project.
An appendix can be found accompanying this document, and it contains all of the pictures and diagrams related to the project.

2 Description of the project.

It was a collaborative project between UCL and FMS schools.

UCL IT Technology students were responsible for creating a system based on the OME's guidelines. During the project teams were introduced to several project management techniques, such as Scrum and Sprint meetings, brainstorms. Teams had to use Gitlab for tracking the project's progress and resolving the problems. Teams had also built the website introducing the product.

Team B3 had the topic "Cooling" to work with.
Team B3 has spent the semester working towards a possible solution for one of Datacentre Industry. With what the group has learned over the course of the semester, they have developed an affordable and effective solution for monitoring the temperature and humidity of the server racks.

The project's goal was to create a product for monitoring the humidity and temperature in the data center. It uses sensors located on the cooling racks to measure humidity and temperature. That data is then displayed on an OLED for the customer's use.

3 Explain Teamwork Structure (scrum, meetings, workload, collaboration, issue management, milestones).

The group has been doing their best to work with the Scrum model for the entire project, to middling results. A lot of the things that the Scrum model does well are things that the group was already doing last semester (short daily meetings, managing issues within Gitlab and working together to make sure that the work that the team had to accomplish was more or less evenly distributed between the various members of the team).

Scrum meetings were not particularly difficult to work into the team's daily routines as it was already mentioned previously that they were already being performed before being brought to the team's attention. The scrum meeting's strict structure also fostered productivity and helped the team focus on the topic of the meeting and the workload of the day.

What was more difficult for the team was cutting down the time spent in morning meetings.
It took more than a few weeks for the team to get it right, and every member got a chance to lead a Scrum as the Scrum Master and record what went on as the Scrum Secretary before it was found that the process ran much smoother when the Scrum Master and Scrum Secretary were the same members from week to week.

The group also performed Sprint reviews which are planning and retrospective meetings that were utilized to get a better understanding of the Project's progress. At the beginning of the project, they were held regularly but eventually the team decided to stop having them because they were time consuming and did not seem to improve the team's productivity.

The group did not work with Gitlab Milestones at all during this project, unfortunately.
This is because Gitlab Milestones, like some other new concepts introduced during the semester was deemed less important than the current assignment for that week.
The group would have benefitted from looking more into using one of the systems of tracking progression, Milestones. Of particular note is the Burn down chart, which would probably have worked nicely within the group's particular work style.

The most challenging part was the collaboration with the OME students. The group had to find both a time which suited everybody involved and additionally a proper channel of communication. The first several meetings were held in Zoom during the usual project lecture time in the breakout rooms. In an effort to facilitate quicker and easier communication, the team decided to create a Discord Group in cooperation with and consisting of teams B3, A1 and B1 as we all worked within the same "Cooling" topic. Each of us had a separate Channel on the Discord where questions could be written to the OMEs which other groups could see. This helped every group involved to save time because sometimes one group had the exact same question as another because the groups were all working on similar things. The time that the meetings were scheduled was different from week to week.

Another important part of the meetings with OMEs was the documentation of the meeting because we had to make sure we keep our meetings stayed structured and productive. The OMEs did not have the same amount of time to spend on this project as we did so we tried to save their time as much as possible. To avoid repetition the groups were writing a "meeting overview" and sending it to the channel on the Discord after a meeting so everybody is aware of what happened during the meeting even if people were absent. The "Meeting Overview" included several points: The first point was whether any important decisions were made during the meeting. The second point was the "to do" list both for the groups and OMEs with clear deadlines so everyone involved had a clear overview of where everyone was regarding the project.

At the beginning of the project the groups were meeting with the OMEs on a regular basis, but early on the meeting ceased. After the group had hit the major points that were requested of us the group was of the mindset that further communication would not be of use for either party. The groups thinking regarding this decision was the fact that the OMEs schedule was extremely full (so full that the other groups on Discord seemed to have a lot of trouble scheduling a meeting) and that the group's schedule was also very full even with two days in a school week dedicated to the Project. The group has realized that the method that they chose was not the best development method possible, and they have made it a goal to collaborate more successfully with customers and other groups in future endeavors.

4 Use Case.

4.1 Research effect

For the research portion of the Project the group browsed through various online markets and compared products. The group was examining three factors; how the different components worked, the finalized look of the product, and which sensors the group wanted to use to create the product that had been assigned by the customer, in this case the OMEs. Additionally, a decision needed to be made regarding which programming language would be utilized for the project. Consultation with the customer (The OME) determined what goals would need to reached and the research that had been achieved would determine which path the group would take towards delivering a finished product.

4.2 Resources

The project was constructed using a couple of different components and systems. Those components include the NodeMCU, DHT11, and OLED-ssd1306. The components and why they were selected is explained in more detail more in section 5.1 of this document.

4.3 Coding

The project was coded using Python/Micro Python. This programming language was chosen because it worked well with the NodeMCU and the group already had experience working with it.

4.4 Minimal Viable Product

The group was able to successfully complete the milestone to create a Minimal Viable Product. The project showed temperature and humidity on a built in OLED screen, phone app and webpage. It worked together with a NodeMCU sensor system and Raspberry Pi. The final touch was a small wooden case made in the Fab Lab. The case was both stylish and functional, featuring compartments to separate the various components. This made sure that the temperature sensor did not receive any heat pollution from the other sensors.

5 Block Diagram.

The diagram [1]shows the overview of the sensor system that has been built.
Reading from right to left, it can be seen that the team has been working on 3 physical components.

5.1 Components.

The three components that were utilized were the DHT11, NodeMCU-ESP8266, and Raspberry Pi.

5.1.1 DHT11.

This is the sensor that the team picked for measuring the humidity in the datacenter.
It was available from the Freenove kit and could measure both humidity and temperature.
The sensor will send its humidity and temperature data to the ESP8266.

5.1.2 NodeMCU-ESP8266.

The ESP8266 receives the sensor data from the DHT11 sensor.
From here, the ESP8266's task is now to transfer the data to the raspberry Pi via MQTT on a private network. The reason this occurs is to make sure data transfer can still happen even if the internet breaks down. It was also helpful because there were a lot of people on the internet working with it which made troubleshooting easier and smoother.

[1] See fig. 1.1. In the appendix.

5.1.3 Raspberry Pi.

The raspberry pi receives sensor data from all of the ESP8266 within the system in this block diagram. There have been only three sensor systems inserted in this block diagram, although this can of course be changed to the customers' needs. After the raspberry Pi has received all of the data from the sensors it will do a mean calculation on the data and transfer that new value via. MQTT to a virtual machine. So, the raspberry Pi will work as a MQTT broker, a subscriber and a publisher.

5.2 Virtual Machines

On the Microsoft Azure cloud infrastructure, we have initialized 3 virtual machine each of which are running MongoDB, Paho-MQTT, and Node-Red.

5.2.1 MongoDB

The NodeMCU will transfer all the data to this database for long term storage. So, the customer has the opportunity to go in and backtrack the sensor data, and see if there have been any unexpected data measurements. But in the interest of redundancy the raspberry Pi will also store the data in its internal storage

5.2.2 Paho-MQTT broker

The broker will work as the middleman between the raspberry Pi and the Node-Red dashboard where it will send the data from the raspberry pi over to the dashboard where the dashboard will display it.

5.2.3 Node-Red dashboard

The Node-Red virtual machine is where the customer will see the layout of the datacenter as a website. On the website the customer will be able to navigate around the different racks and see some graphs that makes it easier to determine if anything is wrong with the temperature and humidity.

6 Documentation of System Tests.

The inventory that the group decided to work with in the project ended up including the NodeMCU-ESP8266, an OLED-Display as well as the DHT11 humidity and temperature sensor.
The original idea that the team had did not include the ESP8266 but was instead utilizing a Raspberry Pico connected to a Digi Xbee module.
Switching the pieces actually came very late in the process when the group realized that the ESP8266 would better serve the scope of what the group was trying to do, due to the fact that there was an ESP8266 units on hand, and the group had accumulated a bit of experience while working with them.

During our testing we developed some error handling in the code, for when the sensor is unexpectedly disrupted, a message will be shown on the display and in terminal.
Furthermore, when the unit tries to connect to Wi-Fi, should a problem arise, a message will be displayed in terminal and the start sequence of the main code won't initialize.
The unit will instead keep trying to reconnect until a connection has been established.
The same goes for connecting to the MQTT broker, if a connection cannot be established a message will be shown in terminal and on the display.

During the first phase of testing the group concentrated on making the ESP8266 work with the sensor, then testing the display separately.
The final step was to make the display post the sensor readings.

During the second phase of testing the group worked on the ESP8266's ability to connect to Wi-Fi. Once that was working properly it was integrated into the main code along with the sensor and the display.

During the third phase of testing the group added MQTT functionality to the unit and tested the module's ability to subscribe to and publish information to a channel.

During the fourth and final phase of testing the group's focus was on consolidating the different codes into one and making sure that all the features could work together.

7 Link to the Gitlab Webpage.

The GitLab page pertaining to this project can be found here[2].
The webpage has documentation regarding how the project was constructed (So that it can be replicated) and all of the files necessary so that the project can actually be replicated.

8 Suggestion on how to proceed/improve/expand the system.

To improve the system there is a few points worth mentioning.

1. The prototype is using a wooden box, without any ventilation holes. This would not be great final project. To improve it, it should use a plastic box/container with ventilation holes, so it does not contaminate the humidity and temperature data, hence sending faulty data.

2. The OLED screen on the wooden box is a great show of what the final product could be. After we have tested this, we figured out that the 9 Volt battery will not sustain the power to the sensor system for very long. The solution for this would be to place the OLED screen on an external power source which could come from the data center itself. We would probably also place the screen on the end of the aisle with an interface option where you could push a button to see what the temperature and humidity would be within a rack of the aisle.

[2] https://21s-itt2-datacenter-students-group.gitlab.io/team-b3/

3. At the beginning of this project, we were working with Digi Xbee devices in tandem with a Raspberry Pi Pico, it would be more power efficient if we replaced the NodeMCU with those instead.

4. The dashboard is now a standalone system that only tracks the temperature and humidity. This this would of course not be efficient for the customer, because the overall project gave us three parameters that the different teams had to work with.
   That is power cooling and ventilation so therefore there should be a way to interface our humidity and temperature sensors dashboard into the bigger picture.

5. The node red dashboard has some minor problems with the LED lights which need to be fixed. Right now, they are not able to change from green to red if an error is detected.

6. Under Node Red dashboard there should be an alarm system put in place if the temperature or humidity rises above the given maximum values of what it should be.

7. On the web page for this project, we should include a statement from each member of the team and underneath that should be a link to our LinkedIn account so that we can use the website to apply for jobs and internships and to show what we accomplished during our education.

8. There are some minor issues on the web page where some of the columns do not fit.

9. We have disregard MongoDB implementation entirely. This means that our project does not have access or connection to a Database, so for now the customer does not have a way to look back on at the historical differences in temperature and humidity.

10. Within MongoDB the customer should be able to run a script that could make a graph of the data, so that the customer does not receive a bunch of random numbers that they cannot use for anything.

11. Implementation of a redundant system was planned in the form of a Raspberry pi which would function as a backup environment, operating a MQTT broker, MongoDB and Node-red dashboard, and as such functioning as a parity system to continue operating the overall functionality of the infrastructure in the case of failure of connectivity to the cloud-based system as well as being the backup environment in case of power failure should it be necessary to operate locally and on battery.

9 Challenges

There have been some minor issues that the team has encountered along the way, including but not limited to:

Covid

The Covid-19 Pandemic was a large hurdle for our group. due the closure of our school and all other meeting places where we could do our group work. This unfortunately left us with only one option for communication and collaboration with our team. Online communication was not always convenient, and although we have tried to meet as often as possible we seem to always be a few steps behind schedule for our weekly assignments in project teaching.

Fablab

Due to Fablab's laser cutter temporarily being under maintenance the team never made a box that could contain all the components and the 9V battery. The team had or made two different prototypes which unfortunately were never presented in the presentation for the combination of the IT-technology class and Frederica machine engineer school students.

Delays

After a slight misunderstanding regarding our order of the components the team selected to use for the project, made our order go in late and the sensors where out of stock at that point. This made it difficult to get the team members to make a full working circuit. We fortunately received our components in time to complete our prototype.

Miscommunication

When we started the project, the students from Frederica's engineering school did not have much knowledge about what they wanted or needed so therefore, when we made our first brainstorm we thought of all places where we could insert sensors into the cooling system to a data center. This made us waste a full day's work in the wrong direction.

10 Conclusion

In the second semester project task, Team B3 was assigned for building IOT system based on the requirements within their key area in the data center which is cooling.

The team then started planning and organizing, dividing the task into smaller parts so it would not get too overwhelming and used their knowledge of the Gitlab Issue Board and Scrum.
Even though there have been some issues encountered, which made the project harder than expected, the team came together and manage to overcome each and every roadblock.

In the end they manage to create a working prototype[3], that they plan on expanding and working towards completing. Overall, the project may have proved to be a bit challenging for everyone involved, but with the help of the instructors who made sure that they were available when needed, cooperating closely with the OMEs, and everyone's resourcefulness and resilience to unexpected changes. We are really proud of the work we put on making the prototype idea a reality.

[3] See fig. 2.1 2.2 and 2.3 in the appendix

11 Appendix

11.1 Block Diagram

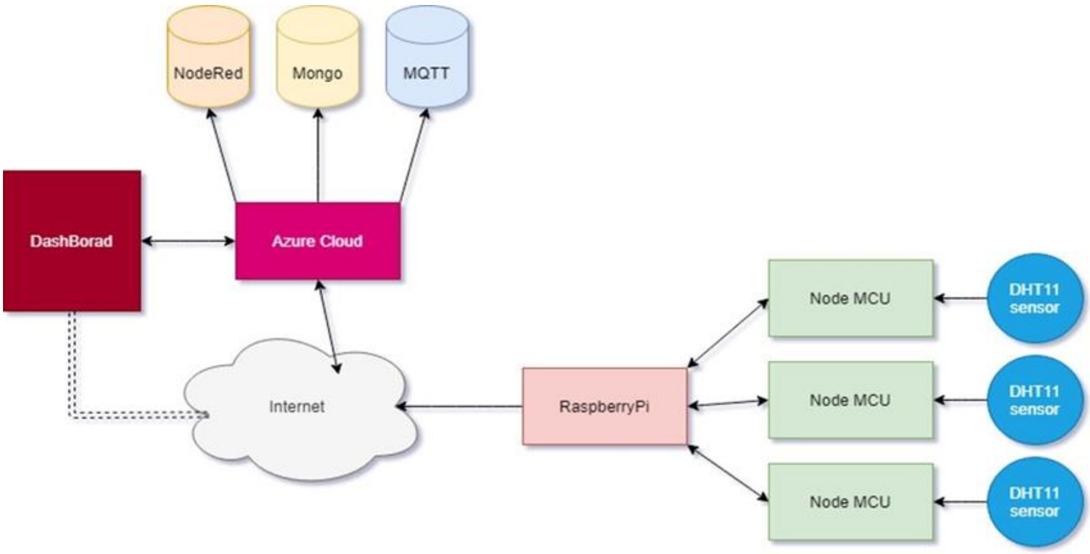

Fig. 1.1

11.2 Conclusion

11.2.1 Circuit on breadboard

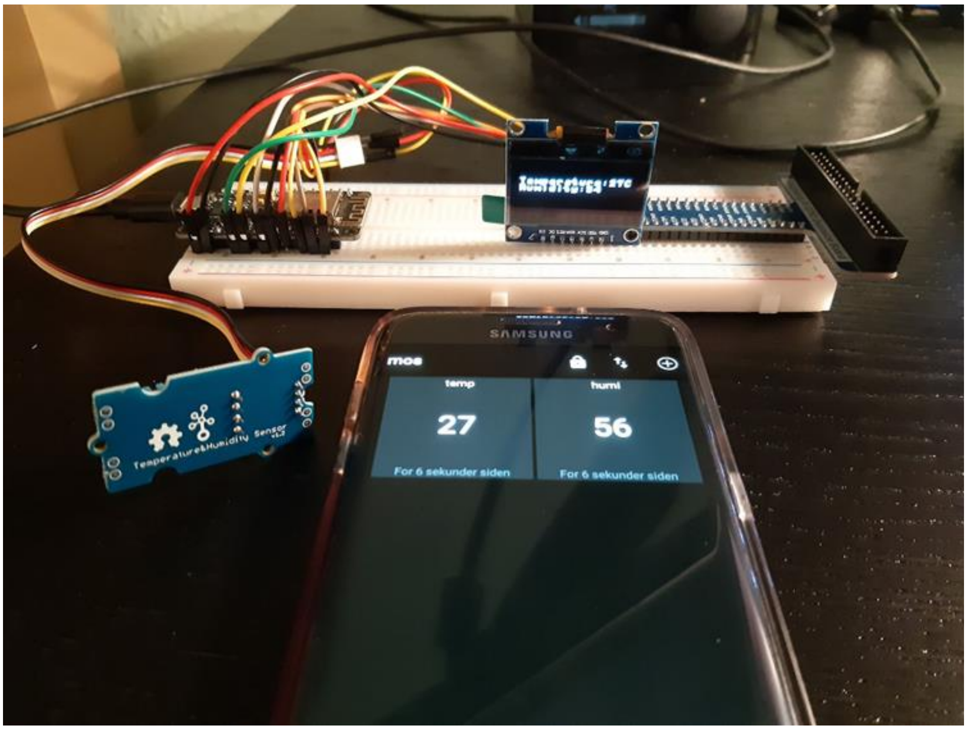

Fig. 2.1

11.2.2 Prototype

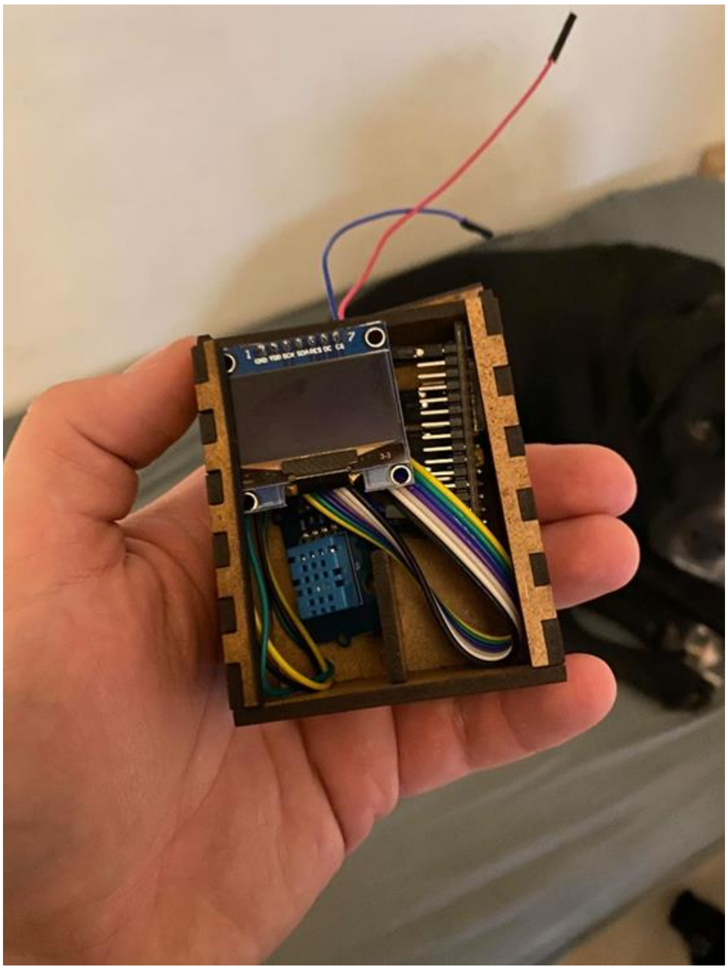

Fig. 2.2

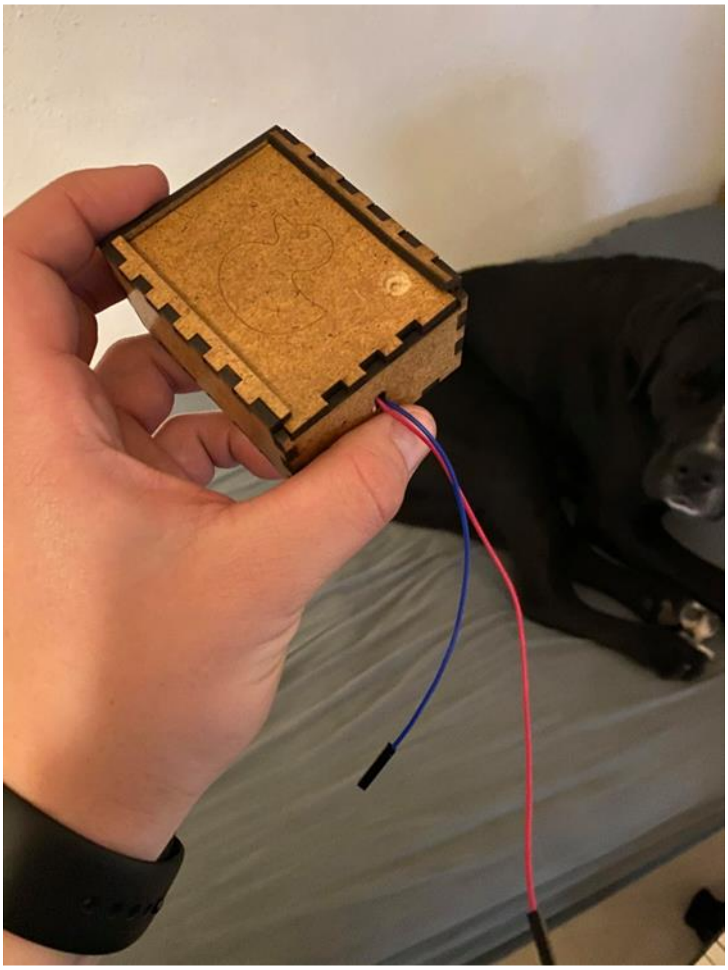

Fig. 2.3