EINDHOVEN UNIVERSITY OF TECHNOLOGY

Group T05 – Reality Stone

Lisa Booms

Danny Yang

Bas van der Meer

Glenn Mossel

Reka Magyari

March 4th 2019

# Technologies for Connectivity Challenge 1

DBSU10

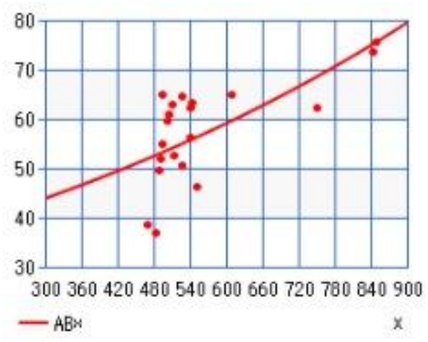
## Measuring the data around the stone

**Which data is collected and how**

The aim of this module is to detect and measure busy times around the Reality stone, using a sound sensor disguised in an everyday object. The everyday object we picked to use a cardboard box of which there are many placed on the steel shelves within short distance from our stone. The sound sensor we used can measure up to 3 meters consistently, which suits the distance between the stone and the placement of the shelf in the squad space in Atlas. In addition, it can also detect any distinctive sounds at further distance such as the entrance door of the space i.e.: slamming doors or the coffee machine.

**The developed module**

The module is designed to process sound input and send a dataset to the OOCSI server as output. The core of the module consists of an ESP32, which processes the incoming signals and communicates with the OOCSI server. Three sound sensors are attached to the ESP32 for multi-directional sound samples. Inside the ESP32 happens the sound processing. This happens in the form of filtering the sounds and sorting the sound in different sound ranges. The captured values will be converted into decibels. After this, the data will be converted into one message and send to OOCSI for data storage. As an additional future feature, the module will become smarter over time due to the self-learning mechanism. The module will try to assign specific events to certain sound patterns at repeating times of the day or the week. This pattern recognition feature needs to be implemented before the data processing to compare the different amplitudes of sound.




| function | value |
|---|---|
| mean of x | 561.1 |
| mean of y | 56.8435 |
| correlation coefficient r | 0.6072 |
| A | 32.6 |
| B | 1.00099 |

Fig 1. C. (2019, February 27). Ab-Exponential regression Calculator [Analyzes the data table by ab-exponential regression and draws the chart.]. Retrieved February 27, 2019, from https://keisan.casio.com/exec/system/14059930973581

**The corresponding API**

The process of capturing to representing the data consist of different functions. These functions are dictated by a set of procedures.

1. Starting with the sampling of the sounds. A capture of two seconds will be made with its values between 0 and 1023.
2. The sensor input of the sound sample could be seen as a set of amplitudes. From these the maximum and minimum valid amplitudes with its sound wave period could be retrieved and summarized as the event's ambient noise or event detection.
3. To communicate through OOCSI, the module first needs to connect to the server authentication of module is crucial to properly locate the database. Storing data on the server requires the module encode data to be send as a message.
4. To retrieve the data, a procedure is written to decode the stored information. Lastly, the captured data needs to be represented as a context activity map. Using the data canvas, the data will be represented on the floor plan with its sensor value over time.

## Prospect to Challenge 2

**User scenario**

In the second challenge collaboration with the other teams will become more important. Our module is very helpful for the others, but we can benefit from the others' work as well. As they are measuring vibrations and movement, that information tell us about the presence of people and we can provide those groups with the precise times when they are active or entering/leaving the room.

**Maintenance and support**

All code is documented on Github. Here other teams can find out how to use the database or open an issue for further support.

https://github.com/Glivmo/OOCSI-Database

https://github.com/Glivmo/OOCSI-mic