Logo

Description automatically generated

Air Quality Innovation

PROJECT PROPOSAL

Marco C, gABrIEL R, vALENTIN S, Niklas r

2021

Contents

[IoT Proposal 2](#_Toc90466941)

[Problem Statement 2](#_Toc90466942)

[Air Quality Values per Zone 2](#_Toc90466943)

[Expansion Possibilities 2](#_Toc90466944)

[Network Device Location Map Zone TQ5 2](#_Toc90466945)

[Network Device Location Map Zone R10 3](#_Toc90466946)

[STAR Network Topology Zone TQ5 4](#_Toc90466947)

[STAR Network Topology Zone R10 4](#_Toc90466948)

[Implementation 5](#_Toc90466949)

[Location 5](#_Toc90466950)

[Example Data: 5](#_Toc90466951)

[Data Proposal 6](#_Toc90466952)

[Problem Statement 6](#_Toc90466953)

[Local Database Architecture 6](#_Toc90466954)

[Chart Wire frames 6](#_Toc90466955)

[Heatmap 6](#_Toc90466956)

[Timeseries 7](#_Toc90466957)

[Dashboard Wireframe 7](#_Toc90466958)

Air Quality Innovation Proposal

# IoT Proposal

## Problem Statement

This past year has been filled with restrictions and dangers due to the Pandemic. It is not safe to have many students grouped up in small spaces. With this project we aim to measure the quality of the air, PPM / VOC values, and display the data live using a Bluetooth Low Energy network.

## Air Quality Values per Zone

The idea of this project is to calculate the carbon dioxide levels in the floor of the school building. We will be using the ESP32 as a client and the Nordic Thingy:52, as well as the Raspberry Pi, as a Server. The idea was to collect as many values as possible and log them. Our proposal will be divided in steps:

1. Obtain the PPM values using an ESP from the Thingy:52
2. Obtain the PPM values using an ESP from multiple Thingy:52’s
3. Write the obtained values to the Raspberry Pi
4. Log the values from the Raspberry Pi to a database’
5. Visualize the values from the Raspberry Pi using D3js or Grafana

We want to use BLE in this project since we got acquainted with the GATT Server architecture. The Thingy:52 has an environmental service with multiple characteristics; each characteristic provides a sensor value that can describe the environment (Example: Temp, Hum, CO2…).

## Expansion Possibilities

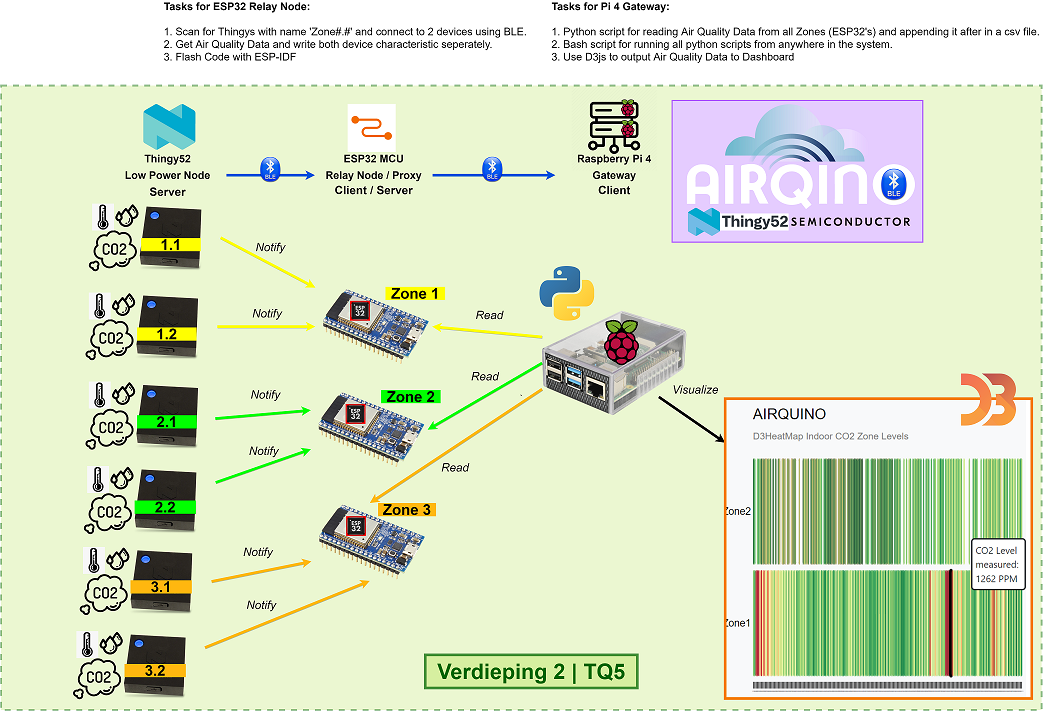
This could be an interesting project for future Smart industry students to pick up. This semester was filled with unknowns and time was a constraint. Possible expansion possibilities are to create a meshed network with the ESP32’s and avoid centralizing the Raspberry Pi to be in range of all ESP’s. The code could be improved to be “Plug and Play” and it could be expanded by adding a camera and counting how many people are in the area with tensor flow (or other libraries). This could improve the accuracy of the “danger meter” as it could correlate the air quality with number of people in the area.

## Network Device Location Map Zone R10



## Network Topology

We have a raspberry Pi logging the average of the carbon dioxide levels collected (PPM) every 15 seconds. We use the ‘watch -n 16 script’ command to logs in a .csv file, this command allows us to run a certain script every x amount of second. We have created a script to log an average of all the values connected. We have chosen 15 seconds as an interval since it was an approximate walking distance between each sensor area.



## Implementation

### Location

After setting up the network in school, we will now collect data for multiple days to be able to visualize the data. The ESPs manage to advertise the data collected from the Thingy:52 correctly, and the Raspberry PI, acting as a gateway, is logging the data correctly. The next step would be to setup the graph to use live data but that will be completed in the Data / HMI Module mini project.A picture containing diagram

Description automatically generated

## Text Description automatically generatedExample Data:

This data format will help us differentiate the zone where the device is, and we can display it with a corresponding color. We will have an interactive dashboard; it will be explained in the [Data Proposal](#_Data_Proposal).

# Data Proposal

## Problem Statement

Having the data can be useful but if you cannot visualize it, it might not be easy to understand to the public. Considering the strawberry field project, we have thought to propose a visual representation of the CO2 using a heat map.

For these data visualizations we will use the Python framework, the library dash will be used for the interactive dashboard. We will explore both options and **if time is not an issue**, we will develop the project a little further and implement augmented reality. What we want to do with augmented reality is to define the levels of carbon dioxide by brightness (lighter = safe, darker = less safe).

## Chart Wire frames

### Heatmap

Chart, bar chart

Description automatically generatedOne of the charts we want to use to display the data is going to be a heat map. This heat map will be used to represent the levels of the value selected on the top, VOC(Volatile Organic Compounds) or PPM(CO2). The right side will serve as a legend to indicate whether the levels are safe or not. The redder it gets, the higher the levels being read. This chart will help us compare the levels from different zones and compare them.

On the other hand, if we would like to see the live data according to the heatmaps color code, we will add a map of the floor and color the zone according to the corresponding sensor values. This type of data is useful because it could give us insight on the amount of people gathered per area.

A picture containing diagram

Description automatically generated

T1

T2

ZONE 1

ZONE 2

T2

T4

T3

T1

PPM

VOC

### Chart, line chart Description automatically generatedTimeseries

This chart will be used to compare the values real time. We can have a more in depth look of what is happening around the building. The difference between this chart and the previous, is that the previous one is used for statistics vs this one is used to visualize the data. The timeseries chart will be less accurate as it could spike the levels if someone passes by or someone tampers with the sensor

## A picture containing timeline Description automatically generatedDashboard Wireframe

Diagram

Description automatically generated