UDC002

GREEN CAMPUS PROJECT SUMMER TERM 2023

Fabian Foltyn
Prof. Dr.-Ing. Volodymyr Brovkov
Augsburg University of Technology,Germany

Abstract. This report describes the sensor network of the Green Campus Team at Augsburg University of Applied Sciences.The following describes the approach to the task and the technologies used. It lists which technologies were considered and how the decision to select the respective technology was made.

Introduction. What does the path look like from a simple sensor to the display of the sensor value on the surface? This is precisely the question that the Green Campus Team at Augsburg University of Applied Sciences asked itself. As Students, our first and most important goal is to learn and discover how the world of IOT works.Therefore,we decided to build a network of sensors and microcontrollers in our project. We want to use sensors that measure everything from temperature to air pressure to motion. To forward the data we use the microcontroller esp8266. With the ESP microcontrollers and a Raspberry Pi we build a mesh network, so that we can use single ESPs also outside the range of a Wi- Fi access point. The frontend will consist of a website to display the data and a program to send emails on certain events.

The purpose of the work. At Augsburg University,all students complete two project assignments. The skills that can be acquired are teamwork, presentation skills and learning about new technologies. The students are free to choose from a range of different topics for the project work. This is also how the members of the teams are formed. All members of the Green Campus team are interested in the Internet of Things.Therefore,the main purpose of our work is to learn and understand IOTtechniques. Furthermore, we want to build a sensor network together.

Main part of the work. How to build a sensor network? How to get the data from the sensors? How to transfer data in wireless networks? What is the best way to store them? And finally, how to display them? These were the questions we asked ourselves at the beginning of our work. We had all worked with sensors or microcontrollers or networks before. But each of us only for himself and each of us had experience in different areas with different technologies.So,we created what is always needed to start the work - a plan (Fig. 1).

Our plan is to connect several sensors that record different values to one esp8266.In several rooms on the campus,one microcontroller with all connected sensors is placed in each room.TheESPs form a painless mesh network among each other. This has the advantage that we only have to make sure that eachESP is within the range of at least one other ESP.Through the mesh network,data can be exchanged over several nodes. In addition, the ESPs can be easily installed in another room because the mesh networkwill reorganize itself.The Central unit of the network will be a Raspberry. The Raspberry gets the information of the painlessMesh network from one ESP,which is connected to the Raspberry via the USB interface and is connected to the other ESPs with the painlessMesh.

The sensor data is stored in an Influx DB.On the Raspberry the software is also installed,which is used to display the sensor data.The Raspberry is connected to the university Wi-Fisothat clients can call up the website on which the sensor data is displayed. In the later course of the project, the Raspberry will also run a program that sends the sensor data via email. Furthermore, we want to equip the Raspberry with a display, from which the sensor data can be read directly.

Our plan foresees several stages with which we will work our way towards the final goal. This is necessary in order to be able to adapt to the remaining time in the semester.

In the first step we have defined our minimum requirements. We want to connect a temperature sensor to a microcontroller. The microcontroller should send the data to a central place via the mesh network.

Each ESP and the Raspberry should be able to receive and send data using the network.The sensor data should be viewable via a website

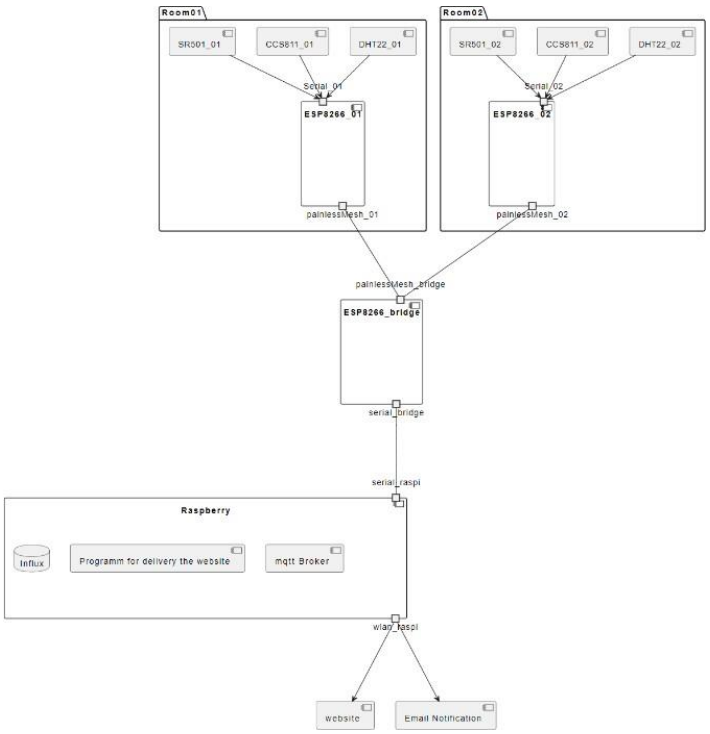

Figure1-Diagram of the sensor network with all components

In the second step we want to connect more sensors to the microcontroller.Also we want the data to be saved in a database. So that we can also display historical sensor data and thus be able to show changes in the data over a longer period of time.In this step we also want to inform the user about certain data via e-mail.

In the last step,we want to venture into more difficult things.For example,we want to determine in this step,with the help of sensors,whether a projector in a room is switched on,or a window is open.In this step,we want to expand the display option of the data with a display that is connected to a Raspberry Pi. We also want to set up an acoustic / visual alarm system in the room and encrypt the sensor data.

With the Sensors we want to record the following values

- Temperature
- Humidity
- CO2
- Movementin theroom

As temperature and humidity sensor we have chosen the DHT22 sensor.To measure the Co2content in the air we chose the CCS811. And to detect movement we choose theSR501 Sensor.TheSensorsvery often used in projects.Therefore,we had a lot of material which we could use as a template.Furthermore, connecting the sensor to the GPIO and reading the data is very easy.

For the microcontrollers we considered the ESP-32and its littlebrotherthe ESP-8266.

We considered both microcontrollers because they have an analogue-to-digital converter and many of the sensors we considered are analogue-based. Furthermore, there are many tutorials and instructions for these Microcontrollers on the internet. So, we had a good literature base to get ideas and help.

In the end we choose the ESP-8266 because it is more power efficient and because we don't need the higher processing power or the Bluetooth function of the ESP-32.

More difficult was the choice of the network. We decided between ZigBee, MQTT or PainlessMesh.In the beginning,we were strongly inclined towardsMQTT,as this protocol was the easiest to implement with the topics and the publisher/subscriber procedure,and the sensor data was easy to distinguish across

different topics and thus easier to display. However MQTT requires a Wi-Fi connection and we were not allowed to connect the ESP8266 microcontrollers to the university Wi-Fi. We considered the possibility of the Raspi setting up its own Wi-Fi network. Or that we equip both the Raspberry Pi and the ESPs with ZigBee modules and send the sensor data to the RaspberryPiviaZigbee.We rejected the approach with Zigbee modules because none of us had any experience with ZigBeeand we were worried that there would be insurmountable timing problems if we transmitted the sensor data withZigBee.We pursued the idea of the Raspberry setting up its own Wi-Fi network and carried out some experiments. However,we also rejected this approach in the end,as the transmission via MQTT requires that there is always a Wi-Fi connection.This requires that theESPs are always within range of the respective access point. Since we wanted to distribute our ESPs across different rooms and buildings on the university campus,we would have had to set up aWi-Fi access point in a room with each ESP and with MQTT we would had not anMeshNetwork.Inaddition,we had already achieved somesuccess with PainlessMesh at this point. Due to the quick successes and the mesh topology of PainlessMesh, we decided to pursue this approach further.
PainlessMesh is a network protocol based on theWi-Fi protocol.Each device connects to another device and is itself an access point forother devices. In this way, a mesh network is created whose range goes far beyond of a Wi-Fi access point. In addition, the network can repair itself and organize itself. painlessMesh can be used on the ESP with the help of a library of the same name. For the Raspberry, there is either a separate implementation that can be installed there. Or it is possible to convert an ESP into an MQTT bridge as painlessMesh and send the data to the Raspberry via MQTT.

ConclusionTo this point,we have set up a mesh network with PainlessMeshbetween4ESPsat the Augsburg University of Applied Sciences. A DHT22 temperature/humidity sensor is connected to each ESP. The data from the sensors is already being sent to the Raspberry via the mesh network. We have already made the first steps with the website to visualize the data with react. However,weare not yet sure about the choice of technology.Furthermore,we will install an Influx DBonthe Raspberry and store our sensor data in it. Furthermore, we will link motion sensors and CO2 sensors with the ESPs.

REFERENCES

1. ESP-MESH with ESP32 and ESP8266 [Online]. Available: https://randomnerdtutorials.com/esp-mesh-esp32-esp8266-painlessmesh/[Accessed on 04.04.2023].
2. painlessMeshgitlab[Online].Available:https://gitlab.com/painlessMesh/painlessMesh[Accessedon07.04.2023].
3. hackster.io ESP32 Wireless Mesh (Made Easy with PainlessMesh) - Part 1 [Online]. Available: https://www.hackster.io/davidefa/esp32-wireless-mesh-made-easy-with-painlessmesh-part-1-f716f5[accessed on 07.04.2023].
4. losant.com Getting started with the ESP8266 and DHT22 Sensor [Online] Available: https://www.losant.com/blog/getting-started-with-the-esp8266-and-dht22-sensor[Accessed on 15.04.2023]
5. bastelgarage DHT22 [Online]. Available: https://www.bastelgarage.ch/dht22-temperatur-und-luftfeuchtigkeitssensor[accessed on 15.04.2023].