HTBLuVA St. Pölten
Higher Institute for Electronics and Technical Informatics
Specializations in Embedded and Wireless Systems

DIPLOMA THESIS

Modular Smart Home System

Add subtitle

Performed in 2025/26 by:

Fabian Schätzschock
Richard Krammer

Supervisors:

Supervisor 1
Supervisor 2

St. Pölten, on 04.04.2025

Submission Statement:
Date:

Supervisors:

Affidavit

The undersigned candidates have chosen to prepare a diploma thesis with the following task description in accordance with the Schulunterrichtsgesetz (School Education Act) § 34 Abs. 3 Z 1 and § 37 Abs. 2 Z 2 [1], in conjunction with the provisions of the Prüfungsordnung BMHS (Examination Regulations for Vocational Middle and Higher Schools), Federal Law Gazette II No. 177/2012 [4], as amended

Modular Smart Home System
Add subtitle

Individual tasks within the overall project:

- **Fabian Schätzschock**: Task 1
- **Richard Krammer**: Task 2

The candidates acknowledge that the diploma thesis must be worked on and completed independently and outside of class time, although class results may be incorporated if appropriately cited as such.

The complete diploma thesis must be submitted digitally and in two printed copies to the supervising teacher no later than **04.04.2025**.

The candidates further acknowledge that cancellation of the diploma thesis is not possible.

Fabian Schätzschock _____

Richard Krammer _____

Contents

1 Introduction

1.1 Our Ambitions

Most free smart home solutions are either too limited in functionality or depend on a strong WLAN-coverage inside the home. The goal of this Project is to provide an open source smart home solution that enables the user to easily modify and extend the system with their own design. Another goal of this project is to provide a smart home solution that does not require every node to be reachable via WLAN.

1.2 Terminology

- **Node:** A whole smart home device that consists of a base module and optional extension modules.
- **Module:** A part of a Node that provides a specific functionality. An extension module can be attached to a Node to extend its functionality.

2 Executive Summary

2.1 Project Overview

The Modular Smart Home System project aims to develop an open-source, all-in-one smart home solution with a modular design. This system allows users to easily modify and extend its functionalities via extension slots to suit their specific needs.

2.2 Objectives

The primary objectives of this project are:

- To create a user-friendly smart home system that integrates self-made home automation devices.
- To ensure the system is modular, allowing for easy customization and expansion.
- To provide comprehensive documentation and support for users and developers.

2.3 Key Features

The key features of the Modular Smart Home System include:

- A robust backend system for managing device communication and data processing.
- A user-friendly frontend interface for monitoring and controlling the smart home system.
- Easily changeable modules to adjust what a Node can do.

2.4 Conclusion

The Modular Smart Home System project aims to deliver a versatile and customizable smart home solution that meets the needs of a wide range of users. By focusing on modularity and ease of use, the project seeks to empower users to create their ideal smart home environment.

3 Requirements

3.1 Hardware

3.2 Software

3.2.1 Microcontroller

IDE

For developing purposes an IDE fulfilling the following criteria is needed:

- **Integrates the Arduino framework:** Needs to integrate Arduino as most of the code will be written using it.
- **Support for different microcontrollers:** Supporting different microcontrollers, to allow for flexibility in developing new Nodes.
- **Manages Library integration:** Streamlines installing and managing libraries used in the project.
- **Free and Open Source:** To allow anyone to easily modify or create their own smart home using this project as a base.
- **Compiling and uploading code:** Simplifying the process of uploading code to the microcontroller.

3.2.2 Communication between Nodes

Note-to-Note

Communication between nodes should be possible between any two nodes both ways.

Note-to-Server

Communication between the nodes and the web server, as well as the database, should be possible both ways.

3.2.3 Backend

To record and read previous data, a database is needed, it should ideally be working with time series data and be able to handle a large amount of data, as it will mainly be used for storing sensor data.

3.2.4 Frontend

The frontend should be able to visualize the data stored in the database, control the actuators and sensors, and display the current state of the system. To implement this, a high-level web framework is preferred to simplify the development process.

4 Networking

4.1 Communication Protocols

4.1.1 MQTT

MQTT is an OASIS standard messaging protocol for IoT-Applications that builds on top of TCP/IP and works on a publish/subscribe model, which makes it easy to set up communication between devices without knowing their IP-Addresses. It requires a dedicated broker to function, that can be set up as a service on the Raspberry Pi [3]. One mayor upside of using MQTT is that it is compatible with every, in this smart home involved, device.

4.1.2 ESP NOW

ESP-NOW is a wireless communication protocol capable of working with Wi-Fi and Bluetooth. It supports a multitude of Espressif microcontrollers and is widely used in IoT-Applications. The main advantage of using ESP-NOW is that is requires no router or existing network to function meaning it's not dependent on a good Wi-Fi coverage. While indoor use drastically reduces the range of 200m per unit depending on various factors [2] the integrated multihop feature that allows the devices to relay messages to each other can be used to extend the range to hard-to-reach locations.

Due to the listed advantages, ESP-NOW is the **preferred communication protocol** for this project.

4.2 Networking Structure

The Nuxt.js Webserver will function as a main broker between the Database and a dedicated ESP32 device that acts as a bridge between every ESP32 and the Nuxt.js Webserver. This is done because a single ESP32 device can only handle either ESP-NOW or a standard Wi-Fi connection, but not both simultaneously. This dedicated bridge is a special ESP32-Board with both a Wi-Fi antenna and an Ethernet port, enabling it to do both at the same time.

4.3 Node-to-Server Communication

Due to the fact that ESP-NOW is exclusive to Espressif microcontrollers, a bridge between the nodes and the server is needed. This bridge can be routed through a preexisting Network as the Web server requires one anyway.

4.3.1 MQTT

MQTT can be used to bridge the gap between the nodes and the server. The server can subscribe to the same topics as the nodes, and the nodes can publish to the same topics as the server.

Disadvantages

Unfortunately, MQTT integration for the Web server as well as the database is not as straightforward as the alternative and would require a lot more overhead.

4.3.2 HTTP

HTTP can be used to bridge the gap between the nodes and the server. This way a dedicated Node can be set up to handle the communication with both the Web server and the database and distribute it to the rest of the nodes.

For this project, HTTP is the preferred protocol for node-to-server communication.

5 ESP32

5.1 Project Structure

5.2 FaRiLib

5.3 ESP NOW

6 Raspberry Pi

6.1 Raspberry to ESP Communication

6.2 Database

6.3 Web Server

6.3.1 Frameworks

To create a web server, a suitable framework needs to be chosen. The following frameworks were considered:

Flask

Flask is a lightweight WSGI web application framework. It's designed to make getting started quick and easy. It's one of the most popular Python web frameworks and can virtually use any Python library including some to control GPIO pins on the Raspberry Pi.
Compared to modern JavaScript frameworks such as Nuxt.js, Flask is considered to be less powerful and less feature-rich.

Nuxt.js

Nuxt.js is a higher-level framework built on top of Vue.js. It's designed to make web development simple and powerful. It abstracts away a lot of complexity when it comes to routing and UI rendering.

6.3.2 Installing Nuxt.js

7 Results

LaTeX is a widely used document preparation system [1].
LaTeX is a widely used document preparation system [5].

[1] A footnote citation [5]

8 Time Tracking

8.1 Name 1

2024		
Week	**Task Description**	**Hours**
36	Preparation	12

2025		
Week	**Task Description**	**Hours**
35	Polishing	12

Total	**24**

Bibliography

[1] *Bundesgesetz über die Ordnung von Unterricht und Erziehung in den im Schulorganisationsgesetz geregelten Schulen*. 1986. URL: https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10009600.

[2] *Data transmission reliability over ESP-NOW protocol in indoor environment*. Espressif Developer Portal. Section: blog. Dec. 19, 2024. URL: https://developer.espressif.com/blog/reliability-esp-now/ (visited on 12/29/2024).

[3] *MQTT - The Standard for IoT Messaging*. URL: https://mqtt.org/ (visited on 01/15/2025).

[4] *Prüfungsordnung BMHS, Bildungsanstalten*. 2012. URL: https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=20007845.

[5] Wikipedia contributors. *LaTeX — Wikipedia, The Free Encyclopedia*. 2024. URL: https://en.wikipedia.org/wiki/LaTeX (visited on 12/06/2024).