

# System for Automatic Heating Control with Multiple Heating Sources

Lukáš Ježek\*

## Abstract

This system aims to design and implement a system like a smart home thermostat to control the house heating and water heating of buildings divided into multiple zones with multiple heating sources. Of the existing solutions, none allows efficient use of multiple sources with zoning, or the solution is expensive and inefficient. In addition, the system offers extensibility to other IoT industries. This application specifically controls the fireplace installation with a heat exchanger, gas boiler, and combined water tank. The system can be controlled either remotely or locally. RaspberryPi 4B was chosen as the IoT platform with DS18B20 sensors for temperature measures and relay modules as actuators. The Control app is developed with web technologies for cross-platform usage. Using Laravel with REST API for the backend part. And Vue.js for the user interface. In addition, it is extendable for other IoT usage.

\*[xjezek19@stud.fit.vutbr.cz](mailto:xjezek19@stud.fit.vutbr.cz), Faculty of Information Technology, Brno University of Technology

## 1. Introduction

This work aims to create a system that automatically controls heating in buildings divided into multiple zones. It will allow the control of a fireplace insert with a heat exchanger, a gas boiler, and a combination water tank. It will also unify the control of all elements into one application. It will also extend it to other types of sensors and controlled elements.

Modern smart thermostats are increasingly being used to control heating. These control the heat source and algorithms for heating individual rooms or zones. However, if the heating system uses multiple heat sources, such as a fireplace with a heat exchanger [1], collisions would occur when heating with both heat sources. Furthermore, the system could overheat. Therefore, it is usually necessary to manually turn off the second heat source or use multiple applications for control. At the same time, the actual solution must include the possibility of control for individual zones.

In addition to heating the building, the same problem arises when providing hot water, whether in the bathroom, kitchen, or other uses. If a building contains multiple sources, they must not overlap, and only one can be used at a time. A control needs to be designed to handle the switching of the individual

sources based on the settings.

Existing solutions do not offer these options, or the systems are expensive, inefficient, and require more complex installation. One of these solutions is the smart thermostat from Netatmo, but like most of the others, it allows control of only one heat source. A custom alternative may be the HomeAssistant system, which allows custom scenarios to be created, but this solution depends on system knowledge and requires custom configuration [2].

The goal is to unify and simplify the control of all elements. Create a hardware implementation to allow control of the heating valves and other elements of the system. And a software implementation for control to create an automated algorithm to control the heating and hot water sources. The entire solution must include the possibility of manual control in case of failure.

The output is an electrical box containing the main controller, relay modules, connection of individual temperature sensors, and connection of valves for heating with the possibility of switching to manual mode independently of the automation. And software providing automatic control via PWA.

## 2. Implementation

The architecture of the whole system can be seen in [Figure 4](#). The system consists of hardware and software, further divided into a backend and a frontend. As hardware, a RaspberryPi 4B microcontroller was chosen, providing sufficient performance for collecting and storing sensor data and running a web server for the backend part of the application [\[3\]](#).

As software, a backend server running on PHP 8.1 was developed using the Laravel<sup>1</sup> framework. MySQL database was chosen for data storage because of the frequency of interconnection between the different parts of the system. This part of the application has the task of data collection and automatic control according to the algorithms described in [figure 1](#), describing the algorithm for heating each zone. In [figure 2](#), the decision algorithm for selecting the heat source, according to the currently set heating mode, is described. The server part provides a REST API to communicate with the user part.

The implementation provides users with a wide range of customization options, and the backend is ready for future sensor and actuator support expansion. For now, DS18B20 sensors and relays connected to GPIO outputs are supported. An extension to the MQTT protocol is planned, which will be used for collecting data and controlling individual relays [\[4\]](#).

The system provides several types of zones: zones for heating, water heating, hot water source, and general zones. The system provides four preset heating modes- fireplace and gas boiler- combined and switched off. Combination mode is fully automated, and the user does not have to deal with manual controls. In case a relay always needs to be opened when heating with the fireplace – for example, in a situation where the fireplace is located after the valve. There is an option in the system to enable specific relays to always open when the fireplace is heated. In addition, the user can preset a preference for electric hot water heating.

The system also provides protection against overheating of the heating circuit when the limit set by the administrator is exceeded. It will open all heating zones or shut down the heat source until the temperature drops back down.

The user part of the application can be seen in [figure 3](#). It was developed using the JavaScript framework Vue.js<sup>2</sup> and Vuetify.js<sup>3</sup>, providing ready-to-

use MaterialDesignLite-based components to simplify the whole application development. To avoid redundant API calls and to work efficiently with data. It uses a local storage system and a repository managed by the Pinia.js library.

The different available pages can be seen in the left navigation bar in [figure 3](#). The data in the same figure comes from the *Reports* page. In the upper part, the current heating mode and the boiler heating by electricity can be read and changed. And data below is a summary from *Zones*, *Temperatures*, and *Equipment* pages.

In addition, the system provides the option of heating by program, where it is possible to set a heating program for each zone. Overview of measured data in a graph by zone or sensor. There is also the possibility of manual control when the heating is switched off or when heating with a fireplace. Alternatively, if the zone is not in automatic mode, it can be controlled manually in any heating mode. The fault information is recorded in the system log, which can be viewed on the *System logs* page.

## 3. Conclusions

All parts of the system were successfully implemented and tested. After testing, a boiler pump run was added to take advantage of the residual heat in the system. Further extensions are planned to support the DHT22 sensors, which also measure humidity. Support for the MQTT protocol for both relays and sensors, allowing the functionality to be extended to other devices using this protocol.

In the future, in case of larger facilities or system expansion, it would be useful to use a time-series database to store measured temperatures and perform analyses on data. This could bring new functionality to the system, such as heating prediction. Another element for development is the implementation of a cache server that, if the API is located in areas with poorer connectivity, will provide data to the user and continuously synchronize it with the server.

Also in the process of being finalized is the English mutation of the entire system and the conversion of the server part into a Docker container.

## Acknowledgements

I would like to thank my supervisor Ing. Jiří Hynek Ph.D. for his advices and professional guidance.

<sup>1</sup><https://laravel.com/>

<sup>2</sup><https://vuejs.org/>

<sup>3</sup><https://vuetifyjs.com/en/>

## References

- [1] krbylevne.cz. Jak fungují teplovodní krbové vložky či kamna? [online], 2023. <https://www.krbylevne.cz/advisor/jak-funguje-teplovodni-vytapeni>.
- [2] HomeAssistant. Generic thermostat. [online], 2023. [https://www.home-assistant.io/integrations/generic\\_thermostat/](https://www.home-assistant.io/integrations/generic_thermostat/).
- [3] Raspberry Pi. Raspberry pi os. [online], 2023. <https://www.raspberrypi.com/documentation/computers/os.html>.
- [4] mqtt.org. Mqtt: The standard for iot messaging. [online], 2024. <https://mqtt.org/>.