MiraTherm Radiator Thermostat Software Specification

Requirement Specification for a Master Project

Fulda University of Applied Sciences
Department of Electrical Engineering

submitted by:	Alexander Menzel
Degree Program:	M.Eng. Embedded Systems (PO 2020)
Matriculation Number:	1316814
Advisor:	Prof. Dr. Uwe Werner

January 21, 2026

Contents

Change Log

Table 1: Document Change Log

№	**Date**	**Version**	**Changed Chapters**	**Change Type**	**Editor**
1	4.11.2025	0.1	All	Initial version	A. Menzel
2	10.11.2025	0.2	3	Requirements correction	A. Menzel
3	17.11.2025	0.3	1, 2, 3	Amendment of the project context, prototype and target hardware separation, requirements correction	A. Menzel
4	19.1.2026	1.0	2, 3, 4	Update of hardware diagram, removal of valve control related requirements, amendment of User Interface (UI) related requirements, time plan update, addition of deliverables section	A. Menzel

1 Introduction

In this chapter, the purpose of this document and the context of the project are described.

1.1 Document Purpose

This document provides a requirements specification for software of a Micro Controller Unit (MCU) based radiator thermostat prototype, which will be developed as part of a master project at Fulda University of Applied Sciences. This software should implement basic consumer functions and could be used as a base for research, development and production of smart heating controllers or thermostats.

1.2 Project Context

The master project will be realized as part of a bigger interdisciplinary development named “MiraTherm Radiator Thermostat”, which includes the following areas:

- **Mechanics**: Development of the thermostat’s power transmission mechanism for proper function with commonly used radiator valves, followed by the design of an enclosure. This work is being realized by Anton Surikov and advised by Prof. Dr. Tobias Müller.
- **Control algorithms**: Engineering of control algorithms to be used by the thermostat. This work will be realized after the completion of the thermostat’s mechanics development.
- **Electronics**: Development of the thermostat’s Printed Circuit Board (PCB) and its integration with mechanical components. It is being realized by Thomas Schneider and advised by Prof. Dr. Daniel Schönherr.
- **Software**: The subject of this work, development of the thermostat’s software and its integration with PCB components. It is being realized by Alexander Menzel and advised by Prof. Dr. Uwe Werner.

2 Concept

In this chapter, the overall concept and approach for the development of software of the radiator thermostat prototype is described. Additionally, the required hardware for development and testing is outlined.

2.1 Solution Approach

In this project, a basic software for a device prototype should be implemented including hardware drivers and general program logic. The description of the eQ-3 eqiva Bluetooth from [1] will be used as a reference for defining the functional scope of the software to be developed.

Due to time constraints, the implementation of control algorithms, wireless connectivity, advanced features, and energy management will be considered out of scope for this project. Instead, the focus will be on developing a solid software foundation that can be extended in the future.

The software should be designed ready for prospective integration of the control algorithms and Matter-over-Thread standard. (For further details about this standard see [2].) The device is supposed to be used for smart home applications, specifically for the integration of the thermostat in Home Assistant, apps like Google Home and/or custom Application Programming Interfaces (APIs).

2.2 Required Hardware

To ensure a degree of independence from the PCB design and mechanics, the software will be developed using a prototype hardware set that resembles the target thermostat hardware (results of the PCB and mechanics development) in terms of components and interfaces. This approach enables early software development and testing before all hardware of the thermostat is available.

A block diagram of the prototype hardware is shown in Figure 2.1. The following components are required:

- **P-NUCLEO-WB55** - MCU development board with Matter-over-Thread standard support

- **eQ-3 eqiva Model N** - Radiator thermostat for disassembly and reuse of its C300 3V motor, gear box and valve connector
- **DRV8833** - Motor driver module
- **Shunt resistor** - 1Ω shunt resistor for current measurement of the motor
- **1.3" 128 × 64 Organic Light-Emitting Diode (OLED) Display incl. SH1106** - Display with an embedded driver
- **KY-040** - Rotary encoder
- **Connecting wires**
- **Breadboard(s)**

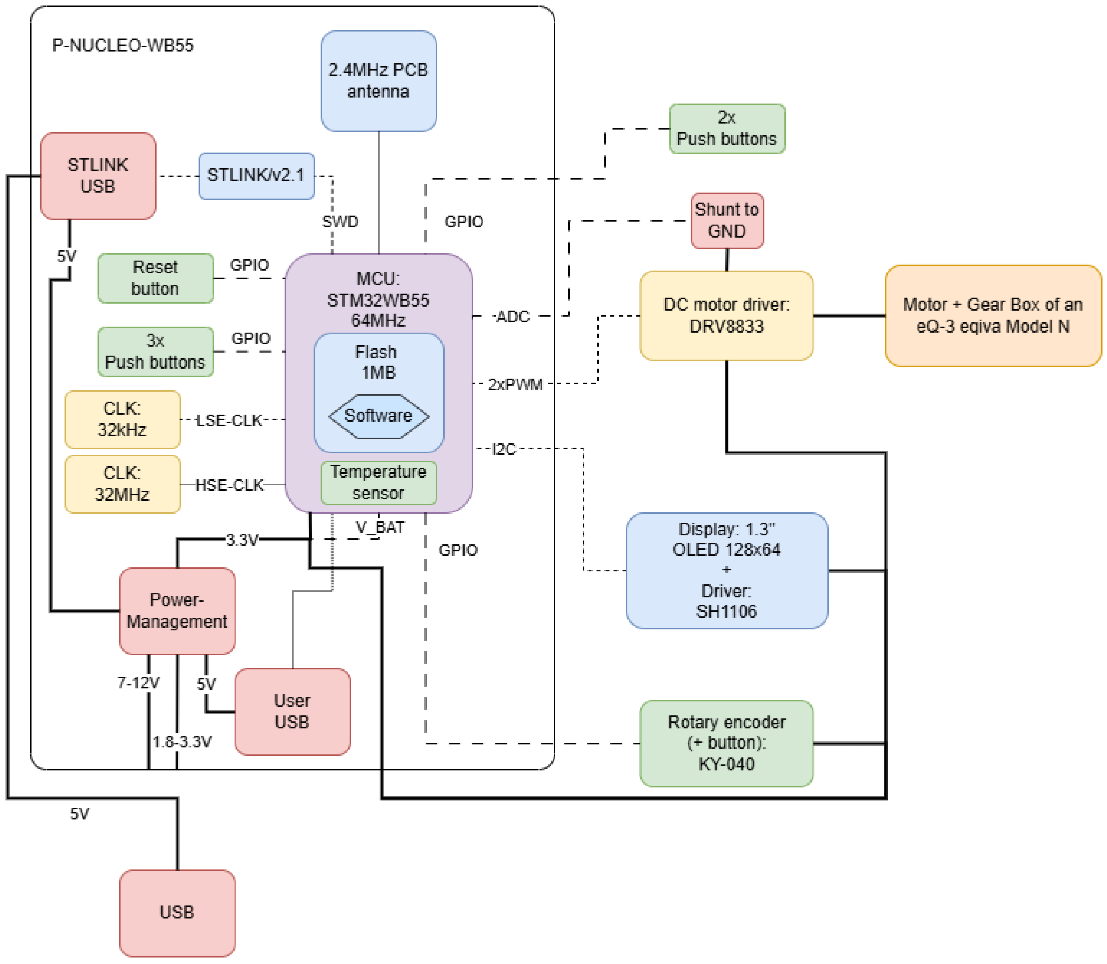

Figure 2.1: Block diagram of the thermostat's prototype hardware

3 Requirements

3.1 Functional Requirements

REQ 1: Display

The software shall integrate a display.

REQ 1.1: Prototype Display

A 1.3” 128×64 OLED display with SH1106 controller over the Inter-Integrated Circuit (I2C) protocol shall be integrated.

REQ 2: Rotary Encoder

The software shall integrate a rotary encoder as a control wheel via Quadrature Decoder (QDEC) interface.

REQ 2.1: Prototype Rotary Encoder

A KY-040 rotary encoder module shall be integrated as the control wheel.

REQ 3: Push Buttons

The software shall integrate the following push buttons via GPIOs:

- **Left** button
- **Middle** button
- **Right** button

REQ 3.2: Rotary Encoder Button

The software shall use the push button integrated into the rotary encoder as the **Middle** button.

REQ 4: Motor Control

The software shall support control of motor driver hardware.

REQ 4.1: Prototype Motor Control

The software shall support control of the eQ-3 eqiva Model N C300 3V motor through a DRV8833 motor driver module. (E.g., using two GPIO or Pulse Width Modulation (PWM) signals.)

REQ 5: Motor Current Measurement

The software shall measure motor current consumption.

REQ 5.1: Prototype Motor Current Measurement

The software shall measure motor current consumption using an Analog-to-Digital Converter (ADC) channel connected to a 0.22Ω shunt resistor in series with the motor circuit.

REQ 6: Power Supply Voltage Measurement

The software shall measure the power supply voltage. The measured voltage shall be used to calculate the battery charge level as a percentage.

REQ 6.1: Prototype Power Supply Voltage Measurement

The software shall measure the power supply voltage using an ADC channel.

REQ 8: Temperature Measurement

The software shall measure the ambient temperature using a digital temperature sensor.

REQ 8.1: Prototype Temperature Sensor

The software shall use a temperature sensor integrated into the STM32WB55 MCU of the P-NUCLEO-WB55 development board.

REQ 9: Configuration Routines

The software shall provide the following configuration routines for initial setup:

- **Configuration on Device (COD)**: Manual configuration via buttons on the device without wireless connectivity.

The setup process shall run at start-up. Valid configuration options saved in persistent storage shall be used as initial values for the setup process.

REQ 10: Date and Time Configuration

The software shall request date and time configuration as the first step in the COD. The user shall be able to set the year, month, day, hour, and minute through the control wheel interface with confirmation capability via the centered button. After setting the date and time, the software shall ask whether the user wants to enable automatic summer/winter time switching.

REQ 20: Target Temperature Range, Resolution and Valve States

The software shall allow setting target temperatures in the range of 5.0°C to 30.0°C in increments of 0.5°C in all related input fields. The software shall also support the setting of following valve states in the same input fields:

- **OFF/CLOSED state**: If the user sets a target temperature below 5.0°C, “OFF” shall be displayed instead of the target temperature number in an input or output field.
- **ON/OPEN state**: If the user sets a target temperature equal or above 30.0°C, “ON” shall be displayed instead of the target temperature number in an input field.

REQ 13: Daily Schedule Configuration

The software shall request daily schedule configuration as the second step in the COD. The user shall be able to set 3, 4 or 5 time slots per day with corresponding target temperatures or valve states using the control wheel interface with confirmation capability via the centered button. The first time slot shall always start at 00:00 and the last time slot shall always end at 23:59. It shall not be possible to set the start of the next time slot before the end of the previous one.

REQ 11: Installation Command

The software shall wait for an installation begin command from the user as the third step in the COD. The text "Begin Installation?" shall be displayed. The user shall initiate the installation by pressing the centered button, which will start the valve adaptation procedure.

REQ 12.1: Mocked Valve Adaptation

The software shall perform a mocked adaptation procedure as the fourth step in the COD. The procedure shall:

1. Display "Adaptation..." indicator page for 10s.

After a successful adaptation the software shall proceed to the home display page.

REQ 14: Home/Main Display Page

On the Home/main display page, the software shall display at least the following information:

- Current time in hours and minutes.
- Current temperature.
- Target temperature or valve state. Adjustable using the control wheel in "Auto" and "Manual" operational modes (see **REQ 17**).
- End time of the current time slot.
- Operational mode indicator.

- Battery charge as a percentage.

REQ 15: Menu

The software shall provide a menu page accessible from the main display page by pressing the left button. The menu shall list the following options:

- Schedule change.
- Temperature offset configuration.
- Factory reset function.

REQ 17: Operational Modes

The software shall support the following operational modes:

- **Manual Mode**: The user can manually set the target temperature or valve state.
- **Auto Mode**: The software follows the heating program and sets the target temperature or valve state according to the current time slot. If the user sets the target temperature or valve state manually in the Auto Mode, the modified temperature shall remain until the next programmed change point.
- **Boost Mode**: Described in REQ 19.

REQ 18: Switching Operational Modes

The software shall allow switching between Manual Mode and Auto Mode using the left button on the home display page.

REQ 19.1: Mocked Boost Mode

The software shall show a boost mode page for 5min after pressing the middle button on the home page. The text "Boost Mode" and the remaining time as a countdown in seconds shall be displayed on this screen until the countdown reaches zero. The function shall be deactivatable at any time by pressing the middle button.

REQ 21: Temperature Offset Configuration

The software shall allow setting a temperature offset between −15.0°C and +15.0°C in increments of 0.5°C. The factory default value shall be 0.0°C. The offset shall be applied to the measured temperature to calculate the effective temperature.

REQ 22: Automatic Summer/Winter Time Switching

The software shall automatically switch between summer and winter time.

REQ 23: Persistent Settings Storage

The software shall persist all user-configured settings in non-volatile memory. These settings shall survive complete power loss (e.g., battery removal) and be restored upon power-up..

REQ 24: Factory Reset

The software shall provide a factory reset function that clears all user settings and returns to factory defaults. A confirmation prompt page ("Factory reset?") shall be displayed to prevent accidental data loss.

3.2 Non-Functional Requirements

REQ 30: Hardware Components Replaceability

The software shall be designed to allow easy replacement of hardware components (e.g., display, driver Integrated Circuit (IC) of the motor, temperature sensor) with minimal changes to the codebase. Hardware abstraction layers shall be implemented to facilitate this replaceability.

REQ 31: Display Button Hints

The software shall display button hints on the screen for each operational page, indicating the function of each button (left, middle, right) relevant to the current context.

4 Planning

4.1 Time plan

The master project will presumably have the duration of 16 Calendar Weeks (CWs) with a pause of 2 CWs and will be divided into:

- CWs 44-45: Software requirements analysis.
- CW 46: Software architecture design.
- CW 47: Design of software interfaces.
- CW 48: Implementation and tests of software drivers.
- CWs 49-51: Implementation of program logic.
- 52-02: Pause
- 03-04: Integration tests and documentation.
- CWs 05-07: Paper writing.
- CWs 08-09: Final review and submission of the paper.

Each calendar week will approximately consist of $\frac{150\text{h}}{14} \approx 10.7$ hours of work.

4.2 Responsibilities

The whole work will be carried out by Alexander Menzel. The advisor for this master project will be Prof. Dr. Uwe Werner.

E-Mails and questions should be answered within 2 working days by both parties. If any problems arise, the advisor has to be informed as soon as possible.

4.3 Deliverables

The following deliverables are expected from this master project:

- **Software Specification Document**: This document.
- **Software Source Code inkl. Comments**: The complete source code of the radiator thermostat software, including in-line comments and documentation. If needed, markdown files explaining the code structure and usage will be provided in the source code repository.
- **Software Architecture, Design and Verification Documentation**: Documentation detailing the software architecture, design decisions, and verification procedures/results. It is allowed to use markdown files for this purpose as well.
- **Final Master Project Paper**: A detailed paper in the form of an Institute of Electrical and Electronics Engineers (IEEE) report summarizing the entire master project. It should include references to all relevant documents and materials produced during the project.

Bibliography

[1] eQ-3 AG, *Operating Manual BLUETOOTH® Smart Radiator Thermostat UK eqiva CC-RT-M-BLE-EQ*, May 2018.

[2] Wikipedia contributors, *Matter (standard) — Wikipedia, the free encyclopedia*, [Online; accessed 28-October-2025], 2025. [Online]. Available: `https://en.wikipedia.org/w/index.php?title=Matter_(standard)&oldid=1318221979`

List of Figures

List of Abbreviations

CW	Calendar Week
MCU	Micro Controller Unit
PCB	Printed Circuit Board
API	Application Programming Interface
UI	User Interface
IEEE	Institute of Electrical and Electronics Engineers
COD	Configuration on Device
OLED	Organic Light-Emitting Diode
QDEC	Quadrature Decoder
GPIO	General Purpose Input/Output
I2C	Inter-Integrated Circuit
PWM	Pulse Width Modulation
ADC	Analog-to-Digital Converter
IC	Integrated Circuit

List of Symbols

Symbol	**Meaning**	**Units**
D	Distance	mm
U	Electric voltage	V
R	Electrical resistance	Ω
d	Diagonal	” (inches)
T	Temperature	°C
t	Time	ms, s, min
v	Valve position	%