**Lastenheft Projektarbeit Thermostat**

**Verteilte Systeme (304271)**

**Prof. Dr. -Ing. Ansgar Meroth, M.Sc. Petre Sora**

**Hochschule Heilbronn – Automotive Systems Engineering**

Status: In Arbeit

Freigabevermerk: Nicht freigegeben

**Winter Semester 2022/23**

**Moritz Hoehnel (210258), Mattis Ritter (210265)**

Inhalt

[I. Tabellarische Versionshistorie 3](#_Toc115723943)

[Glossar 3](#_Toc115723944)

[Abkürzungsverzeichnis 3](#_Toc115723945)

[1. Einleitung 3](#_Toc115723946)

[a. Zweck (des Dokuments) 3](#_Toc115723947)

[b. Umfang (des Softwareprodukts) 3](#_Toc115723948)

[c. Erläuterungen zu Begriffen und / oder Abkürzungen 3](#_Toc115723949)

[d. Verweise auf sonstige Ressourcen oder Quellen 3](#_Toc115723950)

[e. Übersicht (Wie ist das Dokument aufgebaut?) 3](#_Toc115723951)

[2. Allgemeine Beschreibung (des Systems) 4](#_Toc115723952)

[a. Produktperspektive 4](#_Toc115723953)

[b. Produktfunktionen (eine Zusammenfassung und Übersicht) 4](#_Toc115723954)

[c. Benutzermerkmale 4](#_Toc115723955)

[d. Einschränkungen, engl. Design Constraints (für die Entwickler) 4](#_Toc115723956)

[e. Annahmen und Abhängigkeiten (Faktoren, die die Entwicklung beeinflussen, aber nicht behindern z. B. Wahl des Betriebssystems) 4](#_Toc115723957)

[f. Aufteilung der Anforderungen (nicht Realisierbares und auf spätere Versionen verschobene Eigenschaften) 4](#_Toc115723958)

[3. Spezifische Anforderungen 5](#_Toc115723959)

[a. funktionale Anforderungen 5](#_Toc115723960)

[i. externe Schnittstellen 5](#_Toc115723961)

[ii. Betriebszustände 6](#_Toc115723962)

[iii. Output 6](#_Toc115723963)

[iv. Umgebungsbedingungen 6](#_Toc115723964)

[b. nicht-funktionale Anforderungen 6](#_Toc115723965)

[i. Anforderungen an Performance 6](#_Toc115723966)

[ii. Usability 6](#_Toc115723967)

[iii. Wartbarkeit 6](#_Toc115723968)

[iv. Änderbarkeit/Skalierbarkeit 6](#_Toc115723969)

[c. Qualitätsanforderungen 6](#_Toc115723970)

[d. System Security 7](#_Toc115723971)

[e. Datenverwaltung- und Sicherheit 7](#_Toc115723972)

[f. Sonstige Anforderungen 7](#_Toc115723973)

[i. Richtlinien und Vorschriften 7](#_Toc115723974)

[ii. Packaging, Handling, Versand und Transport 7](#_Toc115723975)

[4. Verifikation 7](#_Toc115723976)

## I. Tabellarische Versionshistorie

|  |  |
| --- | --- |
| **Änderungen** | **Datum** |
| Version 1.0  Version 2.0  Version 2.1  Version 2.2 | 01.10.2022  03.10.2022  04.10.2022  17.10.2022 |

# Einleitung

## Zweck (des Dokuments)

Dieses Dokument beschreibt die Anforderungen an das Projekt Thermostat für den Kurs Verteilte Systeme.

## Umfang (des Softwareprodukts)

Im Rahmen der Kurses sollen Funktionen eines Smart Homes realisiert werden. Dieses Projekt soll ein Thermostat mit Mehrstufiger-Hysteresereglung umsetzen.

Es sollen dabei Hardware als auch Software entwickelt werden. Es soll eine zusätzliche Platine für das Microcontroller-Board, welches im Kurs Microcontroller (304132) bereitgestellt wurde, entworfen werden. Die Platine soll die fehlenden Bauteile des Microcontroller-Board beherbergen, welche für die Realisierung des Thermostats notwendig sind.

Es soll eine Software entwickelt werden, welche auf dem Microcontroller genutzt werden kann und die Funktionalität des Produktes herstellt.

## Erläuterungen zu Begriffen und / oder Abkürzungen

|  |  |
| --- | --- |
| Servo  µC. | Servomotor  Microcontroller |

## Verweise auf sonstige Ressourcen oder Quellen

Bauteile, Schaltplanbibliothek als auch Code-Elemente werden durch die Dozenten des Kurses Verteilte Systeme bereitgestellt.

## Übersicht (Wie ist das Dokument aufgebaut?)

Nach der Einleitung folgt eine allgemeine Beschreibung des Systems, gefolgt von spezifischen Anforderungen und der Verifikation.

# Allgemeine Beschreibung (des Systems)

## Produktperspektive

Das Produkt soll als Teil eines Smart Homes eine Heizung steuern. Der Nutzer soll die Solltemperatur an dem Microcontroller-Board und an dem Zentralen Display einstellen können. Durch die LED bekommt der Nutzer Feedback, wie stark geheizt wird.

## Produktfunktionen (eine Zusammenfassung und Übersicht)

Wie oben erwähnt soll eine Platine und Software entwickelt werden.

Platine: Die Kommunikation zwischen den Bauteilen der Platine und dem Microcontroller-Board soll über Serielle-Schnittstellen realisiert werden. Die Art der Schnittstellen sind durch die jeweiligen Bauteile vorgegeben. Auch die Bauteile sind vorgeben. Generell gibt es ein Bauteil zum Messen der Ist-Temperatur, ein Stellglied für die Heizung und eine LED zur Visualisierung. Darüber hinaus soll ein CAN-Controller verwendet werden. Die vergebenen Bauteile sind in d.Einschränkungen zu finden. Alle weiteren Bauteile (Buchsen, Quarze, Kondensatoren und Widerstände) sollen passend zu den vorher genannten Bauteilen gewählt werden.

Software: Diese soll die Daten des Temperatur-Sensors auswerten. Es soll eine Regler Logik erstellt werden, welche Ausgangssignale zur Steuerung des Servos und er LED erstellt.

## Benutzermerkmale

Da das Produkt Teil eines Smart-Homes muss jeder Bewohner eines Hauses als Benutzer gewertet werden. Somit sind Nutzer jeden Alters, Bildung, Sprache und Sachkenntnis potenzielle Bediener des Produkts.

## Einschränkungen (für die Entwickler)

Es sollen folgende Bauteile verwendet werden:

- Servo vom Typ FXX-3037-TOP

- Temperatursensor vom Typ TMP75B

- Vollfarb-LED vom Typ WS2812

- CAN-Controller vom Typ MCP2515

## Annahmen und Abhängigkeiten

Der Schaltplan und das Board werden mit dem Programm EAGLE erstellt. Die Platine soll vom Auftraggeber gefertigt werden.

Die Software wird mit dem Programm Microchipstudio entwickelt. Die Programmiersprache ist C.

## Aufteilung der Anforderungen

Eine Vernetzung mit weiteren Funktionen des Smart-Homes ist nicht teil des Projekts, kann aber später realisieret werden.

# Spezifische Anforderungen

## funktionale Anforderungen

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Name | Detaillierte Beschreibung | Werte-  bereich | Einheit | Priorität | Status | Verifikations-  kriterien |
| A.1 | Stellen | Der Bediener soll dabei die Möglichkeit haben in Ganzzahl-Schritten seine Wunschtemperatur zu stellen. | 5-40 | °C | high | offen | Erfolgreiche Vorgabe durch Versuch |
| A.2 | Stellen | Der Bediener soll die Temperatur am µC und am Zentralen Display stellen;  Die Stellung am µC soll nur als Back-up verwendet werden |  |  | low | offen | Erfolgreiche Änderung der Solltemperatur |
| A.3 | Messen | Der Temperatursensor soll die Ist-Temperatur aufzeichnen | min. 5-40 | °C | high | offen | Prüfen wie in 4. beschrieben |
| A.4 | Messen | Temperatursensor Toleranz | +/-0,5 | °C | low | offen | Hersteller-vorgabe |
| A.5 | Servo | Soll die Heizung stellen (Es soll fünf Stufen geben) | 0-4 |  | high | offen | Prüfen wie in 4. beschrieben |
| B.1 | Stufe 0 | Ist-Temperatur >= Solltemperatur;  Die Heizung bleibt aus | 0 | % | high | offen | Durch Manipulation via Laptop soll die Ist-Temperatur auf einen Wert gestellt werden, dass die jeweiligen Grenzen der Temperatur-  bereiche erreicht sind. Der Servo soll die in Wertebereich geforderte Stellung erreicht wird. |
| B.2 | Stufe 1 | Ist-Temperatur < Solltemperatur;  t <= 3°C;  Die Heizung wird zu 25% geöffnet | 25 | % | low | offen |
| B.3 | Stufe 2 | Ist-Temperatur < Solltemperatur;  3°C < t <= 6°C;  Die Heizung wird zu 50% geöffnet | 50 | % | low | offen |
| B.4 | Stufe 3 | Ist-Temperatur < Solltemperatur;  6°C < t <= 9°C;  Die Heizung wird zu 75% geöffnet | 75 | % | low | offen |
| B.5 | Stufe 4 | Ist-Temperatur < Solltemperatur;  9°C < t;  Die Heizung wird zu 100% geöffnet | 100 | % | low | offen |
| C.1 | LED, Stufe 0 | LED ausgeschaltet | aus |  | low | offen | Testdurchführung wie bei Nr. B.X,  jeweilige Farben werden wiedergegeben |
| C.2 | LED, Stufe 1 | LED leuchtet in der Farbe grün | grün |  | low | offen |
| C.3 | LED, Stufe 2 | LED leuchtet in der Farbe gelb | gelb |  | low | offen |
| C.4 | LED, Stufe 3 | LED leuchtet in der Farbe orange | organge |  | low | offen |
| C.5 | LED, Stufe 4 | LED leuchtet in der Farbe rot | rot |  | low | offen |
| A.6 | Regelung | Software soll Signal an Servo geben wie geöffnet wird | 0-4 |  | high | offen | Stellt wie in 4. beschrieben |
| A.7 | Regelung | Beschränkung Regelabweichung | +/-0,5 | °C | high | offen | Keine höheren Temperaturen vom Temperatur-sensor gemessen |
| A.8 | Quellcode | Kommunikation via SPI Schnittstelle mit Servo |  |  | high | offen | Kommunikation möglich |
| A.9 | Quellcode | Kommunikation via I2C mit Temperatursensor |  |  | high | offen | Kommunikation möglich |
| A.10 | Quellcode | Kommunikation via Serieller Schnittstelle mit LED |  |  | low | offen | Kommunikation möglich |
| A.11 | Quellcode | Auswerten Messwerte, mit Regelung Erstellen eines Output Signals für Servo |  |  | high | offen | Servo führt Bewegungen aus |
| A.12 | Quellcode | Melden welche Heizstellung an LED |  |  | low | offen | Farbe passt zur Servostellung |

### externe Schnittstellen

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Name | Detaillierte Beschreibung | Werte-  bereich | Einheit | Priorität | Status | Verifikations-  kriterien |
| D.1 | CAN-Out | 0x400;  Temperatur Vorkomma;  Signed Byte | 8 | bit | low | offen | Nachricht wird an Display empfangen |
| D.2 | CAN-Out | 0x400;  Temperatur Nachkomma;  Unsigned Byte | 8 | bit | low | offen |
| D.3 | CAN-Out | 0x400;  Status der Heizung entsprechend der in B.X definierten Stufen;  Unsigned Byte | 8 | bit | low | offen |
| D.4 | CAN-In | 0x401;  Solltemperatur;  Signed Byte | 8 | bit | high | offen | Nachricht vom Display wird empfangen |

### Betriebszustände

Sobald das Gerät eingeschalter ist, soll es dauerhaft die Wunschtemperatur regeln. Die Abtastung der Temperatur wird maximal im 100ms Abstand gefordert.

### Output

Der Output der Platine ist die Servo-Stellung und die LED.

### Umgebungsbedingungen

Es wird festgelegt, dass das Thermostat Einsatztemperaturen von 5-40°C hat. Diese Voraussetzung wird durch die eingeschränkte Nutzung in einem Haus festgelegt. Hier werden keine größeren Temperaturschwankungen erwartet. Es soll vernachlässigt werden, dass Mess- und Stellelektronik nah beieinandersitzen und eventuelle Ist-Temperaturunterschiede zwischen Raummitte und Thermometer entstehen können.

## nicht-funktionale Anforderungen

### Anforderungen an Performance

Die Platinentempertur ist nicht zu managen.

### Usability

Der Nutzer soll durch Tastendrücke schnell die Temperatur ändern können. Die Stellgeschwindigkeit soll dabei nicht bedeutend langsamer sein als die Heizungsbedienung mit einer herkömmlichen Drehregler Bedienung (Gesamte Dauer zum Ändern der Solltemperatur <= 10s)

### Wartbarkeit

Die Platine soll einfach auf das Microcontroller-Board gesteckte werden können.

### Änderbarkeit/Skalierbarkeit

Eingriffe in die Software können auch noch nach Beendigung des Projekts durchgeführt werden können.

## Qualitätsanforderungen

Alle Lötstellen sollen sauber hergestellt werden. Es ist kein Gehäuse gefordert.

## System Security

## Datenverwaltung- und Sicherheit

## Sonstige Anforderungen

### Richtlinien und Vorschriften

### Packaging, Handling, Versand und Transport

# Verifikation

Es soll die Richtigkeit der Temperaturmessung im Labor geprüft werden. Es soll der Messwert des Thermometers mit dem Wert eines herkömmlichen Thermometers zur Ermittlung der Raumtemperatur verglichen werden.

Darüber hinaus soll eine Sichtprüfung durchgeführt werden, welche die richtige Servo-Stellung verifiziert. Messzenario: Per Manipulation soll ein Temperaturdelta gestellt werden, dass alle 5 Stufen durch den Servo gestellt werden.