

# Dokumentation: Klimaanlage

Chris Müller, Kilian Röper

December 18, 2024

## Contents

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Ideenfindung und Konzeptphase</b>	<b>3</b>
2.1	Skizze der Zielanwendung aus Kundensicht . . . . .	3
2.2	Anforderungen aus Kundensicht . . . . .	4
<b>3</b>	<b>Anforderungsanalyse</b>	<b>5</b>
3.1	Software-Anforderungen . . . . .	5
<b>4</b>	<b>Software Architektur</b>	<b>6</b>
4.1	Softwarearchitektur als Simulink-Diagramm . . . . .	6
4.2	Schnittstellen . . . . .	6
<b>5</b>	<b>Software Design</b>	<b>7</b>
5.1	Softwaredesign als Simulink Modell . . . . .	7
5.2	Funktionalität der einzelnen Units . . . . .	10
5.3	Liste der Schnittstellen . . . . .	11
<b>6</b>	<b>Implementation</b>	<b>12</b>
6.1	Model Guidelines . . . . .	12
6.1.1	Blockfarben . . . . .	12
6.1.2	Millersche Zahl . . . . .	12
6.1.3	Einordnung in Hierarchien und Wiederverwendbarkeit . . . . .	12
6.1.4	Signalfluss . . . . .	12
6.2	Beispiel zur Implementierung . . . . .	13
<b>7</b>	<b>Implementation Review</b>	<b>14</b>
7.1	Review . . . . .	14
7.2	Vorher-Nachher-Vergleich der Implementierung . . . . .	15
7.3	Review-Ergebnisse . . . . .	15
<b>8</b>	<b>Software Unit Tests</b>	<b>16</b>
8.1	Testdokumentation . . . . .	16
8.2	Testergebnisse B2B-Test . . . . .	16
8.3	Testergebnisse Funktionstest . . . . .	17

<b>9</b>	<b>Software Integration und Integrationstests</b>	<b>18</b>
9.1	Simulink-Modell der integrierten Software . . . . .	18
9.2	Testdokumentation . . . . .	18
9.3	Testergebnisse . . . . .	19
<b>10</b>	<b>Softwaretests</b>	<b>20</b>
10.1	Testdokumentation . . . . .	20
10.2	Testergebnisse Qualification Test 1 . . . . .	20
10.3	Testergebnisse Qualification Test 2 . . . . .	21

# 1 Einleitung

Dieses Dokument beschreibt eine in Simulink entwickelte Klimaanlage. Ziel ist es, alle während des Projekts entstandenen Work Products eines Worksteps innerhalb eines eigenen Kapitels zu dokumentieren.

## 2 Ideenfindung und Konzeptphase

### 2.1 Skizze der Zielanwendung aus Kundensicht

Die Klimaanlage dient der Regelung der Temperatur und Luftfeuchtigkeit in einem geschlossenen Raum. Die Hauptkomponenten sind:

- Ein Sensor zur Messung der aktuellen Innen- und Außentemperatur.
- Ein Steuergerät zur Verarbeitung der Sensordaten und Berechnung der Steuerbefehle.
- Aktoren wie ein Heiz-/Kühlelement.
- Ein Benutzerinterface zur Eingabe von Sollwerten (gewünschte Temperatur).

Abbildung 1 zeigt eine schematische Darstellung der Klimaanlage.

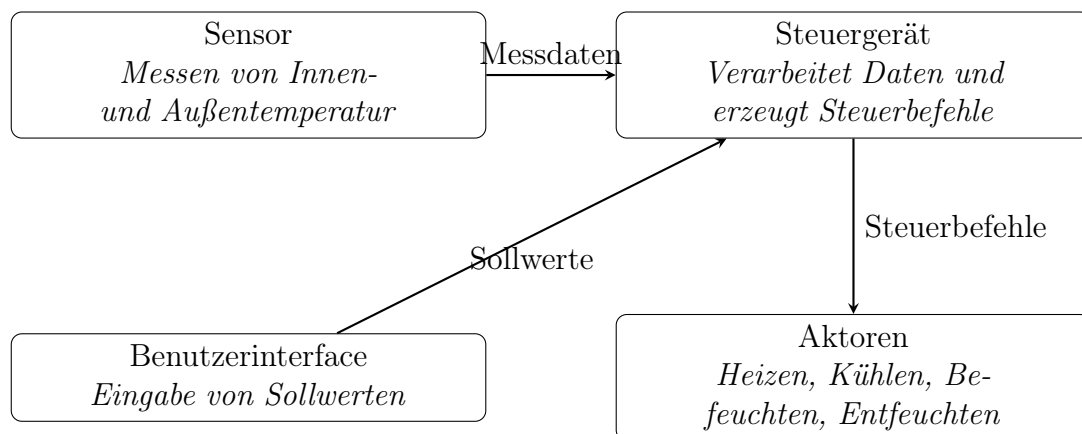


Figure 1: Schematische Darstellung der Klimaanlage

## 2.2 Anforderungen aus Kundensicht

Nr.	Anforderung
1	Die Klimaanlage muss in der Lage sein, die Innenraumtemperatur in einem Bereich von 16°C bis 28°C zu regeln.
2	Die Klimaanlage muss bei allen üblichen Außentemperaturen zwischen -20°C und +50°C zuverlässig funktionieren.
3	Die Klimaanlage soll das Fahrzeug innerhalb von 5 Minuten auf eine angenehme Temperatur herunterkühlen oder aufheizen.
4	Die Temperatur im Fahrzeuginnenraum soll konstant gehalten werden und Schwankungen von $\pm 1^\circ\text{C}$ nicht überschreiten.
5	Die Klimaanlage soll einen Automatikmodus besitzen, der selbstständig die ideale Temperatur auf Innen- und Außentemperatur regelt.
6	Bei einem Defekt oder einem ungewöhnlich hohen Energieverbrauch soll die Klimaanlage in einen sicheren Modus wechseln, der keine weiteren Schäden verursacht und einen Hinweis im Fahrzeugdisplay gibt.

Table 1: Zusammenfassung der Kundenanforderungen für die Klimaanlage

## 3 Anforderungsanalyse

### 3.1 Software-Anforderungen

Nr.	Anforderung
1	Die Klimaanlagesteuerung muss die Temperatur auf eine voreingestellte Zieltemperatur zwischen 16°C und 28°C regeln können.
2	Die Software muss die Klimaanlage so steuern, dass sie die Innenraumtemperatur in maximal 5 Minuten um mindestens 10°C herunterkühlen oder aufheizen kann.
3	Die Software muss die Innenraumtemperatur auf $\pm 1^\circ\text{C}$ der eingestellten Zieltemperatur halten.
4	Die Klimaanlage muss basierend auf der Innenraum- und Außentemperatur selbständig die Innenraumtemperatur auf den vom Nutzer eingestellten Sollwert regeln.
5	Die Software muss eine Fehlererkennungsfunktion implementieren, die bei Außentemperaturen ab $-20^\circ\text{C}$ und niedriger, sowie $+50^\circ\text{C}$ und höher, einen Sicherheitsmodus aktiviert.
6	Die Software muss eine Fehlererkennungsfunktion implementieren, die bei einem Energieverbrauch größer als 5kW einen Sicherheitsmodus aktiviert.
7	Im Sicherheitsmodus muss die Klimaanlage deaktiviert werden, bis der Energieverbrauch wieder unter 5kW und die Temperatur wieder über $-20^\circ\text{C}$ oder unter $50^\circ\text{C}$ beträgt.
8	Eine Meldung „Klimaanlage im Sicherheitsmodus“ muss im Fahrzeugdisplay erscheinen, wenn der Sicherheitsmodus aktiviert ist.
9	Die Temperatur muss in $1^\circ\text{C}$ Schritten zwischen $16^\circ\text{C}$ und $28^\circ\text{C}$ vom Nutzer am Bedienelement der Klimaanlagesteuerung eingestellt werden können.

Table 2: Zusammenfassung der Softwareanforderungen für die Klimaanlagesteuerung

## 4 Software Architektur

### 4.1 Softwarearchitektur als Simulink-Diagramm

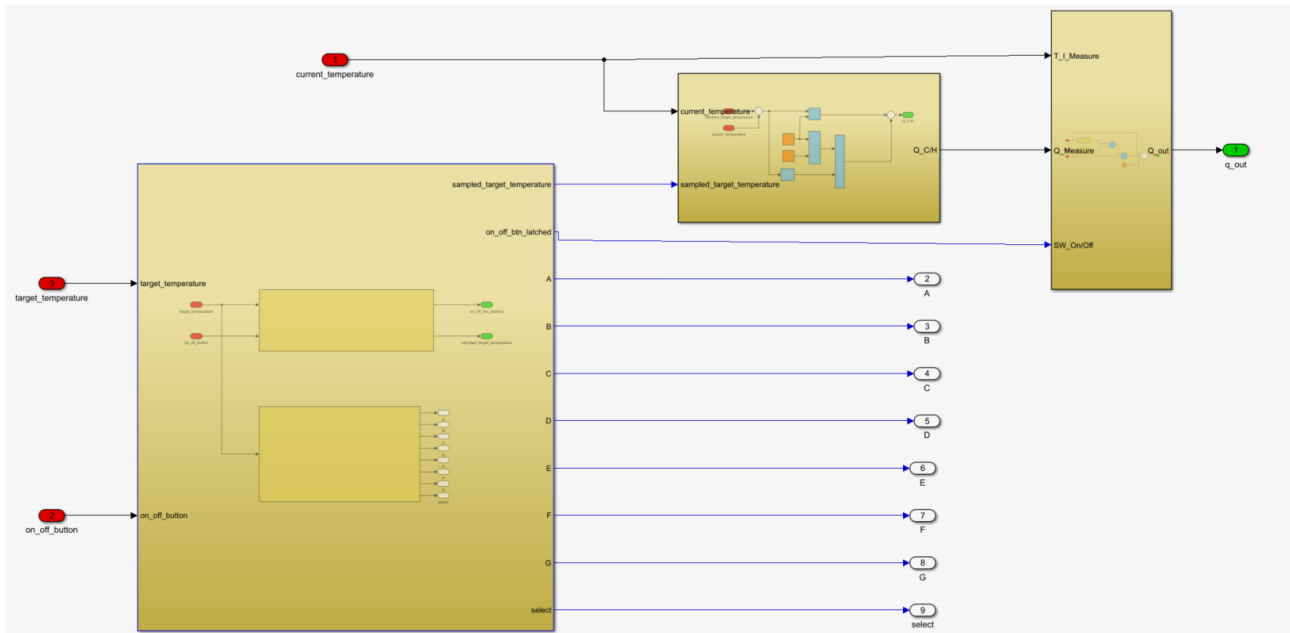


Figure 2: Architektur der Software

### 4.2 Schnittstellen

Schnittstelle	Zweck	Datentyp	Wertebereich	Weitere Informationen
q_out	Heiz- und Kühlleistungswert zur Regelung der Leistungszufuhr zu den Heiz - bzw. Kühlelementen	Integer	-5000–5000	eine Bereichsüberschreitung führt zu einem Fehler
A bis G	Pin Ausgänge zum 7 Segment Display auf dem Nucleo-Board Shield	bool	0-1	nur boolsche Werte erlaubt
select	Signal zur Wahl zwischen den beiden Displays	bool	0–1	wird von einer Software-internen Clock angetrieben, zur Wahl zwischen den beiden Displays

Table 3: Liste der Schnittstellen

## 5 Software Design

### 5.1 Softwaredesign als Simulink Modell

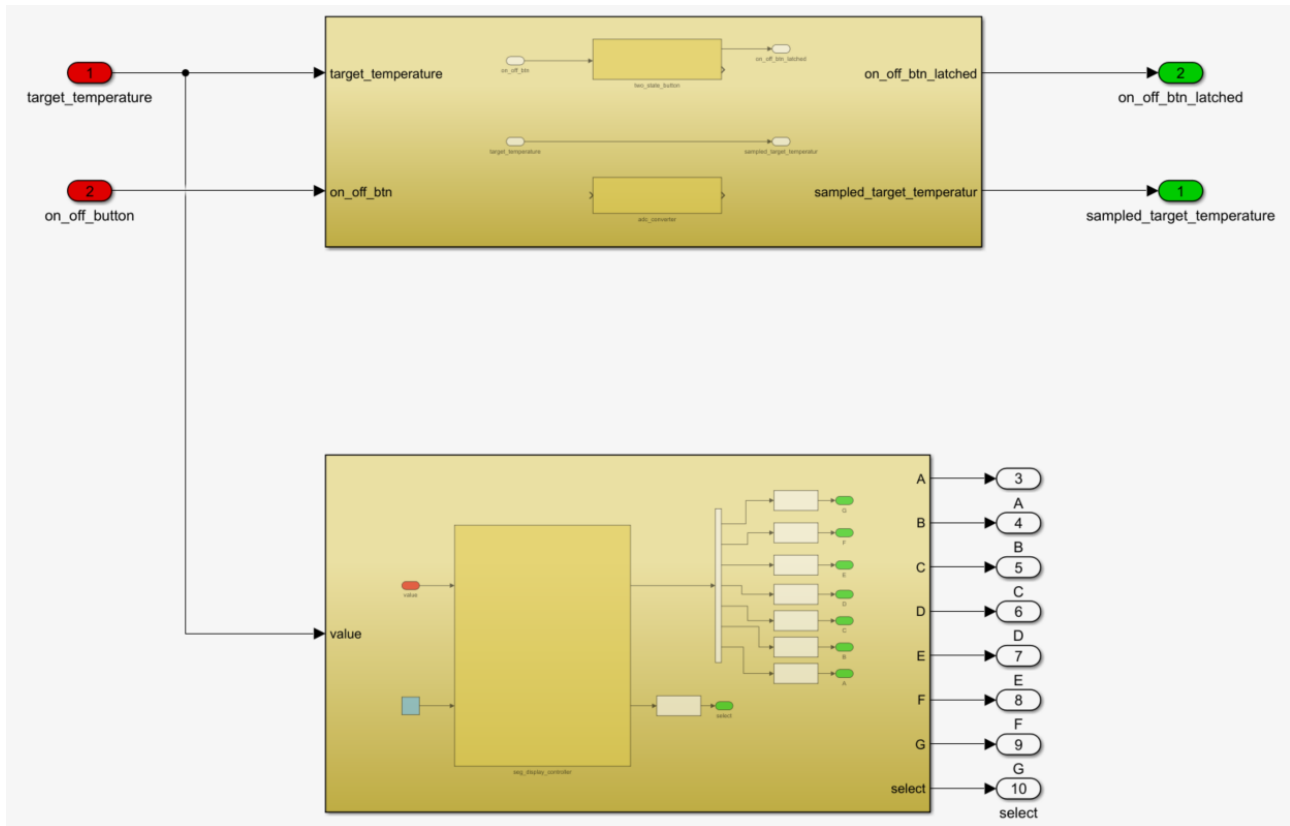


Figure 3: User Interface der Software

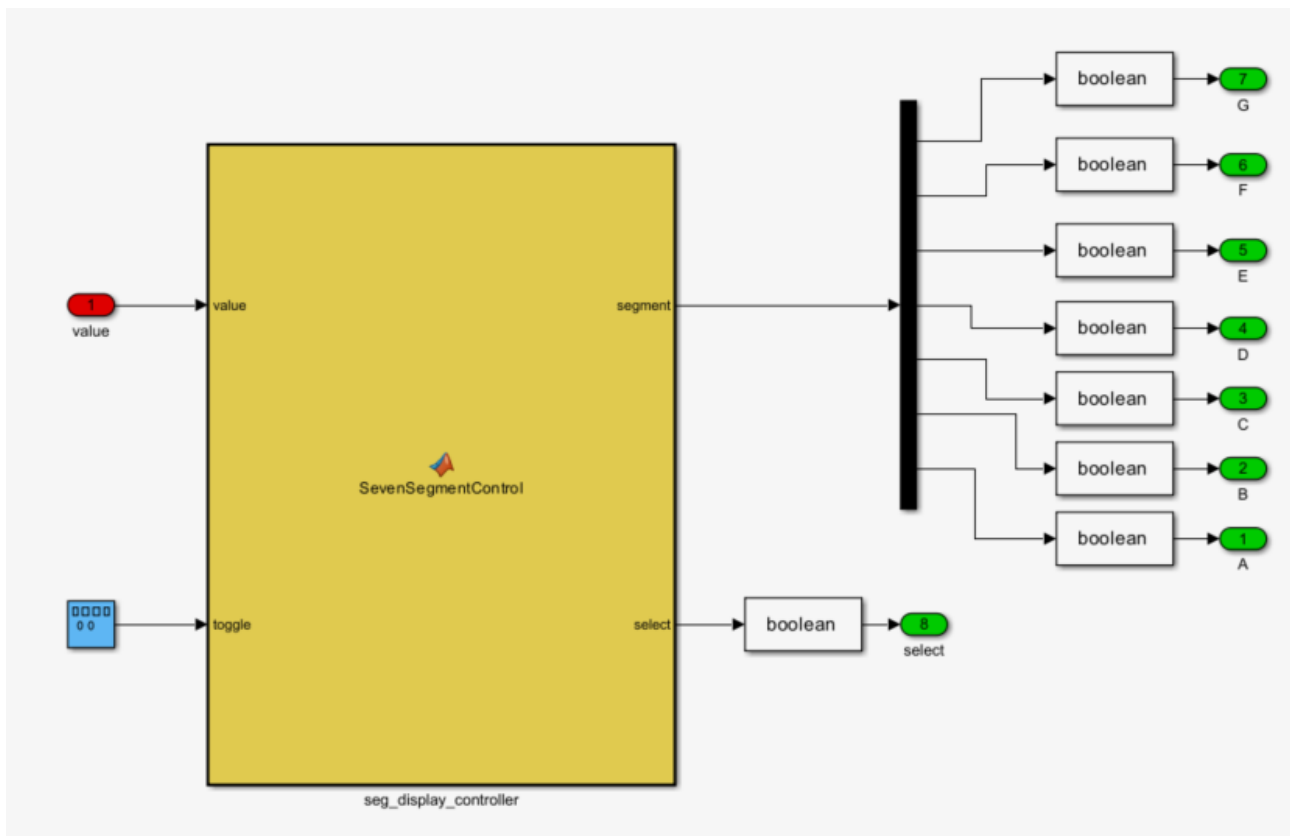


Figure 4: 7-Segment-Display Ansteuerung zum Shield der Software

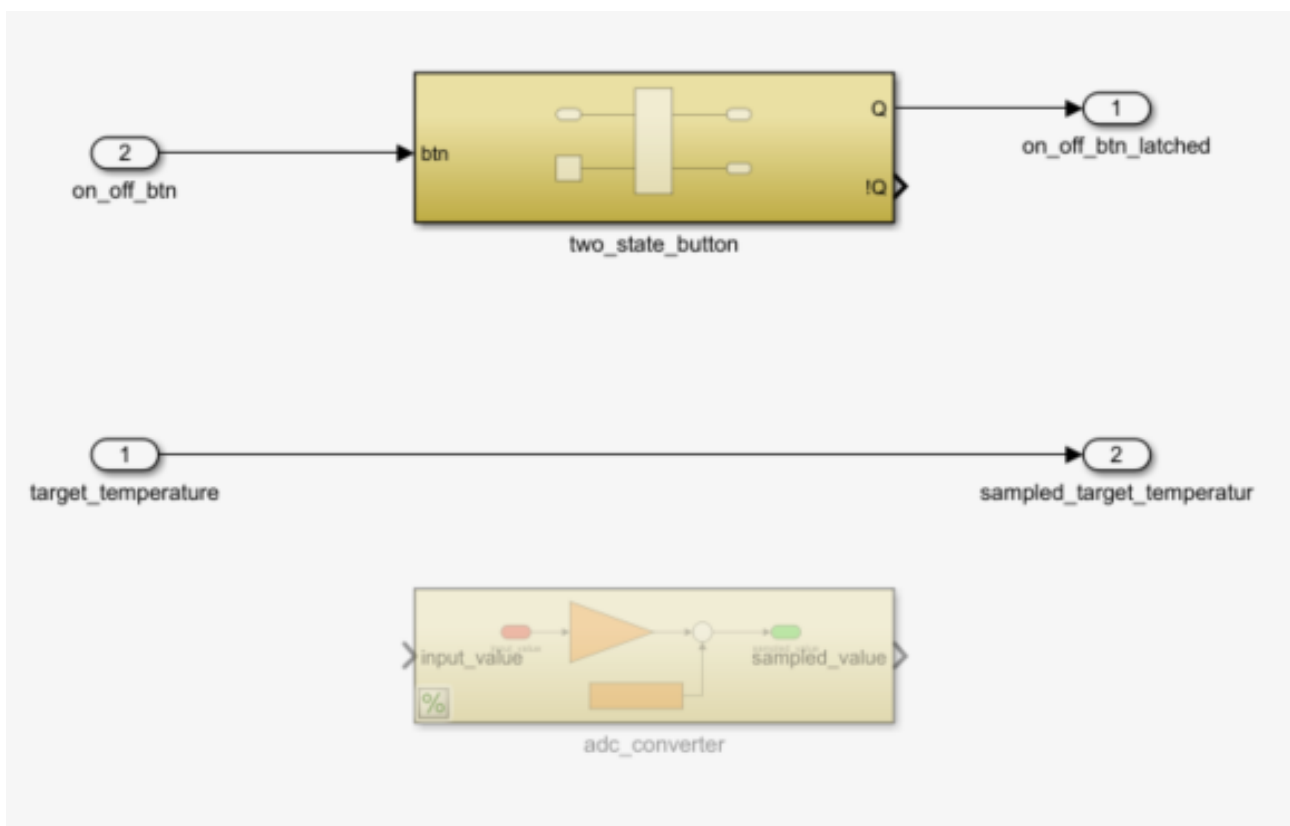


Figure 5: Eingabe System der Software



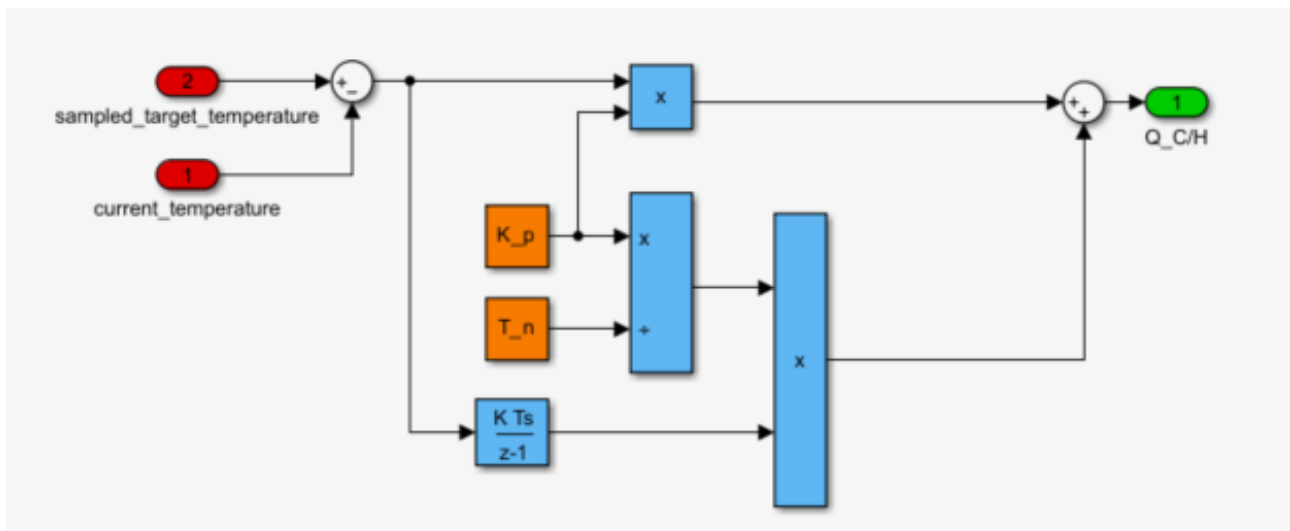


Figure 6: Leistungsregler der Software zur Temperaturregelung

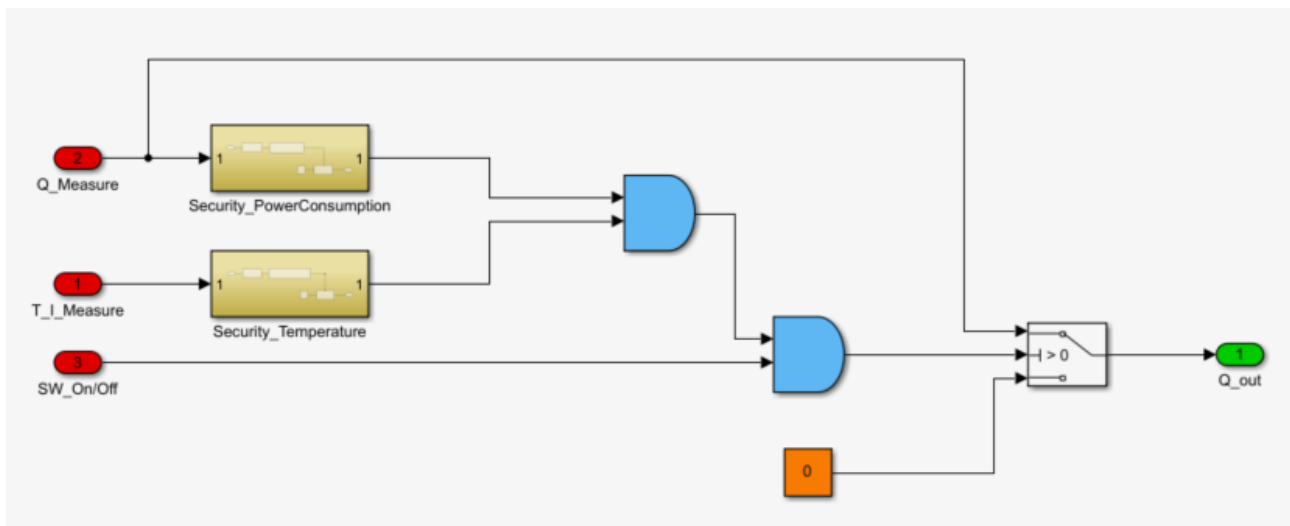


Figure 7: Sicherheitsblock der Software zur Sicherstellung entsprechender Leistung und Temperatur

## 5.2 Funktionalität der einzelnen Units

Softwareunit	Zweck	Umsetzung
User Interface (Abb.3)	Eingabe aller vom Nutzer bestimmten Eingaben, wie der on-off button oder die Zieltemperatur	Implementierung zweier Module. Eines soll den button input verarbeiten und der das andere soll die Temperatureingabe auf das 7 Segment Display mappen
7-Segment-Display (Abb. 4)	mappen der Temperatureingabe auf die 7 Steuerepins für das Nucleo Board	Implementierung einer Matlabb-internen Funktion die die Aufgabe des Mappings übernimmt
Eingabe System (Abb. 5)	Verarbeitung des Buttons und der Nutzertemperatur	Der Button State muss gespeichert werden. Dafür muss ein Latch verwendet werden. Wird eine Eingabe vom Board für die Temperatur genutzt (ADC), dann wird über den ADC Converter der Wert in einen entsprechenden Temperaturwert zwischen 16-28°C umgerechnet
Leistungsregler (Abb. 6)	Regelung der Leistung zur Temperatureinstellung	Implementierung mit einem PI Regler der Innenraumtemperatur entsprechend der vom Nutzer gewählten Temperatur einstellt
Sicherheitsblock (Abb. 7)	Abschalten der Regelung wenn Temperatur oder Leistungswerte über- oder unterschritten werden	Implementierung mit einer if-else Abfrage. Die derzeitige Temperatur und Leistung sind die Eingaben.

Table 4: Liste der Schnittstellen

### 5.3 Liste der Schnittstellen

Schnittstelle	Zweck	Datentyp	Wertebereich	Weitere Informationen
current _temperatur	Rückführung der Temperatur zur kontinuierlichen Regelung	double	-20 – 50	dieser Bereich darf nicht überschritten werden
on-off-button	Nutzereingabe zum Ein- oder Ausschalten der Regelungsanlage	bool	0-1	nur boolsche Werte erlaubt
q_out	Heiz- und Kühlleistungswert zur Regelung der Leistungszufuhr zu den Heiz - bzw. Kühlelementen	Integer	-5000-5000	eine Bereichsüberschreitung führt zu einem Fehler
A bis G	Pin Ausgänge zum 7 Segment Display auf dem Nucleo-Board Shield	bool	0-1	nur boolsche Werte erlaubt
select	Signal zur Wahl zwischen den beiden Displays	bool	0–1	wird von einer Software-internen Clock angetrieben, zur Wahl zwischen den beiden Displays

Table 5: Liste der Schnittstellen

## 6 Implementation

### 6.1 Model Guidelines

Um die einzelnen Einheiten lesbarer und übersichtlicher zu gestalten, wurden die folgenden Modellrichtlinien berücksichtigt:

#### 6.1.1 Blockfarben

Für unser Modell wurden die folgenden Blockfarben verwendet:

Farbe	Funktion
Rot	Inports
Grün	Outports
Orange	Konstanten und Gains
Hellblau	MATLAB-eigene Funktionsblöcke

Table 6: Farbliche Zuordnung der Blockfunktionen

#### 6.1.2 Millersche Zahl

Die Einheiten enthalten jeweils die  $5 \pm 2$  Komponenten.

#### 6.1.3 Einordnung in Hierarchien und Wiederverwendbarkeit

Wie bereits in den Abschnitten Softwaredesign und Architektur beschrieben, gliedern sich unsere Komponenten in einzelne Units, die die jeweilige Implementierung enthalten. Ziel ist es, die Funktionalität der einzelnen Codeabschnitte so zu kapseln, dass sie ohne großen Aufwand wiederverwendet werden können. Dabei können einzelne Units auch wieder Units enthalten.

#### 6.1.4 Signalfluss

Der Signalfluss erfolgt typischerweise von links nach rechts, nur die Signalerückführung erfolgt in umgekehrter Richtung. In diesem Modell gibt es nur einen Fall, nämlich die Rückführung der Regelstreckenausgangsgröße (thermal vehical body).

Die Abbildung ?? zeigt eine frühe Implementierung unseres Modells, die alles versucht, um ein Subsystem testweise zu implementieren.



Komponente	Änderung
7 Segmentanzeige	Nun in einem eigenen Subsystem mit den anderen Komponenten des User-Interface
Der Regler	In einem eigenen Subsystem
Die Regelstrecke	Außerhalb des Blocks zur Code-Generation

Die folgenden Abbildungen 9 und 10 zeigen noch einmal, wie die Model Guideliens eingesetzt wurden.



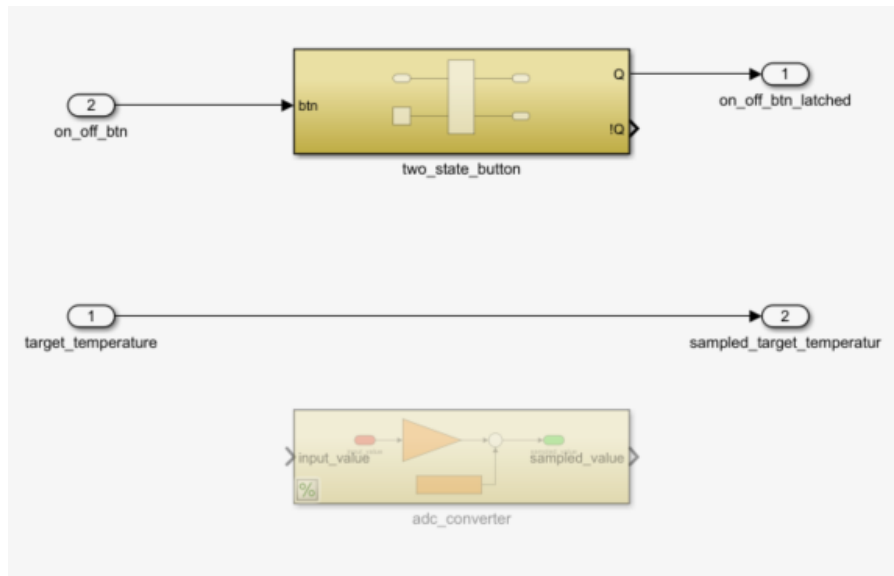


Figure 10: Eingabe System der Software

## 7 Implementation Review

### 7.1 Review

#### 1. Auswahl der zu reviewenden Unit:

Die Komponente **User Interface** wurde für den Review ausgewählt. Sie ist entscheidend für die Eingabe der Zieltemperatur und das Ein- und Ausschalten der Klimaanlage.

#### 2. Review-Prozess:

- **Reviewer:** Kilian Röper (externes Teammitglied).
- **Schritte:**
  - Überprüfung der **Einhaltung der Modeling Guidelines**: Blockfarben, Hierarchien, Signalfluss.
  - **Dokumentationsprüfung**: Sind alle Blöcke ausreichend kommentiert, um die Funktion der Unit verständlich zu machen?
  - **Laufzeitanalyse**: Gibt es Optimierungsmöglichkeiten für die Verarbeitung der Eingaben?
  - **Funktionale Analyse**: Existieren potenzielle Fehlerquellen, z. B. ungenaue Umsetzung der Solltemperatur?

#### 3. Findings und Anpassungen:

- **Finding 1**: Der Signalfluss war nicht vollständig nachvollziehbar, da die Temperaturdaten direkt vom ADC blockiert wurden. **Lösung**: der ADC Converter wurde entfernt und wird nur hinzugefügt wenn das Nukleo Board zum Einsatz kommt.
- **Finding 2**: Die Frequenz zum verarbeiten und speichern des Button States war zu hoch und es gab daher Fehler. **Lösung**: Reduktion der Frequenz von 1000Hz auf 60Hz.
- **Finding 3**: Die Strukturierung der Komponenten two-state button, ADC Converter und Segment Display war nicht hinreichend, da keine Libraries verwendet wurden. **Lösung**: Implementierung von Libraries für jede einzelne Komponente.

## 7.2 Vorher-Nachher-Vergleich der Implementierung

Eine vergleichende Darstellung zeigt exemplarisch die Änderungen an der Unit **User Interface**:

### 1. Vorher:

- Signalfluss direkt vom ADC zur Verarbeitung.
- Frequenz auf 1000Hz und Fehler durch zu viel Simulation.
- Keine Unterteilung der einzelnen Komponenten in einzelne Libraries.

### 2. Nachher:

- Signalfluss wird auswählbar über einen ADC gesteuert.
- Frequenz auf 60 Hz.
- Einteilung einzelner Komponenten innerhalb des User Interfaces in weitere Libraries.

Änderung	Vorher	Nachher
Signalfluss	Ohne Demux	Mit Signal-Demux
Frequenz	1000Hz	60Hz
Libraries	keine vorhanden	vorhanden

## 7.3 Review-Ergebnisse

- Die Reviewergebnisse wurden besprochen und in der Implementierung umgesetzt.
- Die geänderte Implementierung wurde validiert und zeigt nun verbesserte Reaktionszeiten und eine übersichtlichere Signalverarbeitung.

## 8 Software Unit Tests

### 8.1 Testdokumentation

Zum Ausführen der Unit-Tests muss zuerst das init-Skript ausgeführt werden, um alle Variablen aus "params/ac\_params.m" im Workspace zu speichern. Danach kann in "Klimaanlage/tests/unittest" das "run\_unit\_tests.m" Skript ausgeführt werden, mit dem die funktionalen und b2b tests gestartet und geplottet werden.

Andernfalls kann das "run\_all\_tests.m" Skript ausgeführt werden wodurch alle Tests gestartet werden aber auch die Unit-Tests.

### 8.2 Testergebnisse B2B-Test

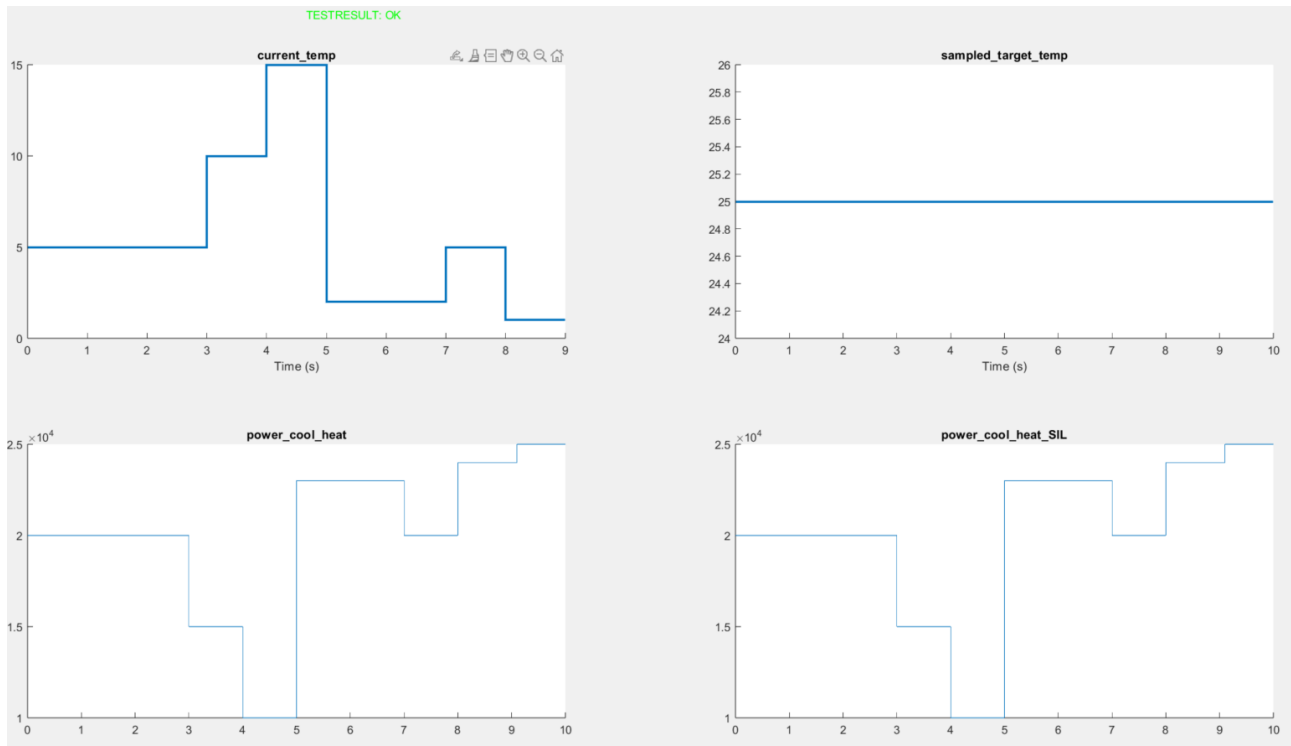


Figure 11: Ergebnis des b2b-Tests

Der B2B-Test zeigt, dass die Software Komponenten (generiert aus dem Regler) sich genauso verhält wie das Simulink Model. "Power\_cool\_heat" und "Power\_cool\_heat\_SIL" verhalten sich im Plot genau gleich. Das Ergebnis ist außerdem mit einem grünen OK nach erfolgreicher Verifizierung am oberen Rand angezeigt.



### 8.3 Testergebnisse Funktionstest

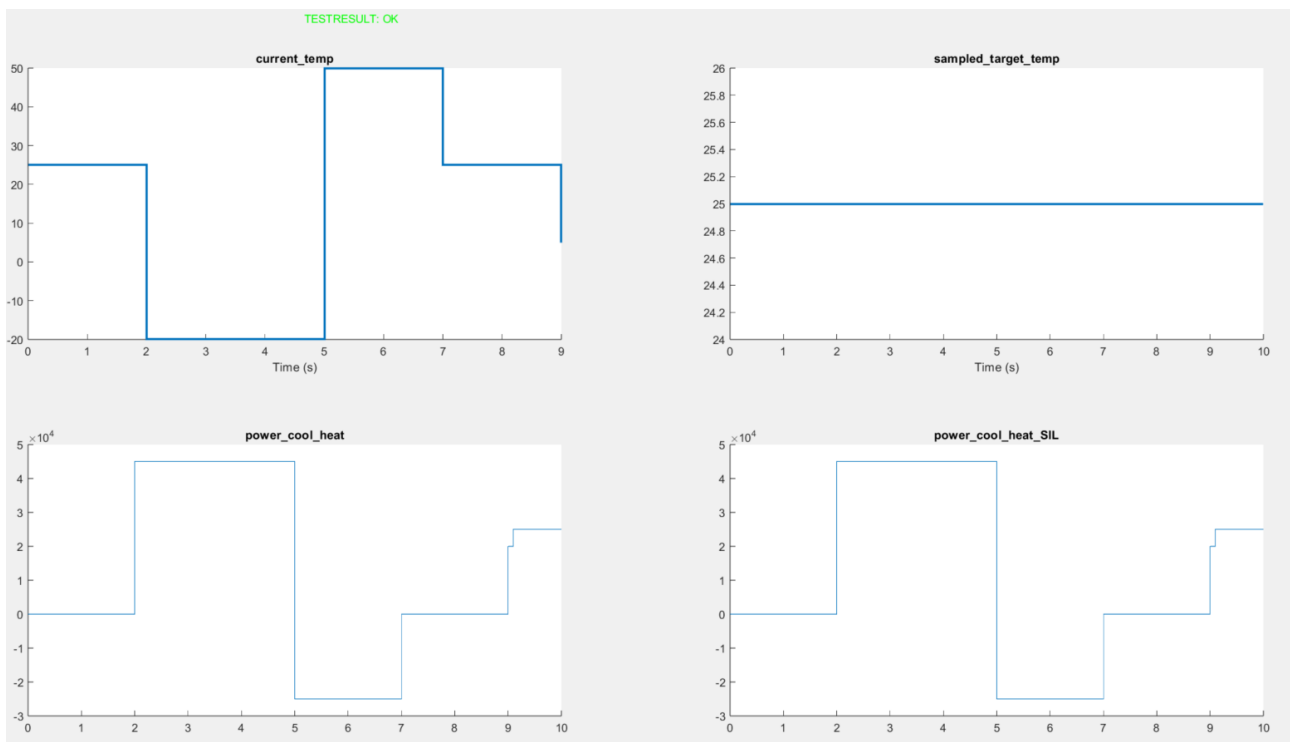


Figure 12: Ergebnis des funktionalen Tests

Der funktionale Test zeigt, dass sich die Software Komponente (generiert aus dem Regler) funktional so verhält wie sie sollte. Es wurde geprüft ob eine entsprechende Heiz oder Kühlleistung geliefert wird zum Erreichen des Temperaturziels. 25°C soll gehalten werden. Bei einer Temperatur von 50°C wird ein negativer Wert (kühlen) und bei einer Temperatur von -20°C ein positiver Wert (Heizen) abgegeben. Bei 25°C Außentemperatur fällt keine Leistung ab, da das Ziel bereits erreicht ist. Das Ergebnis ist außerdem mit einem grünen OK nach erfolgreicher Verifizierung am oberen Rand angezeigt.



### 9.3 Testergebnisse

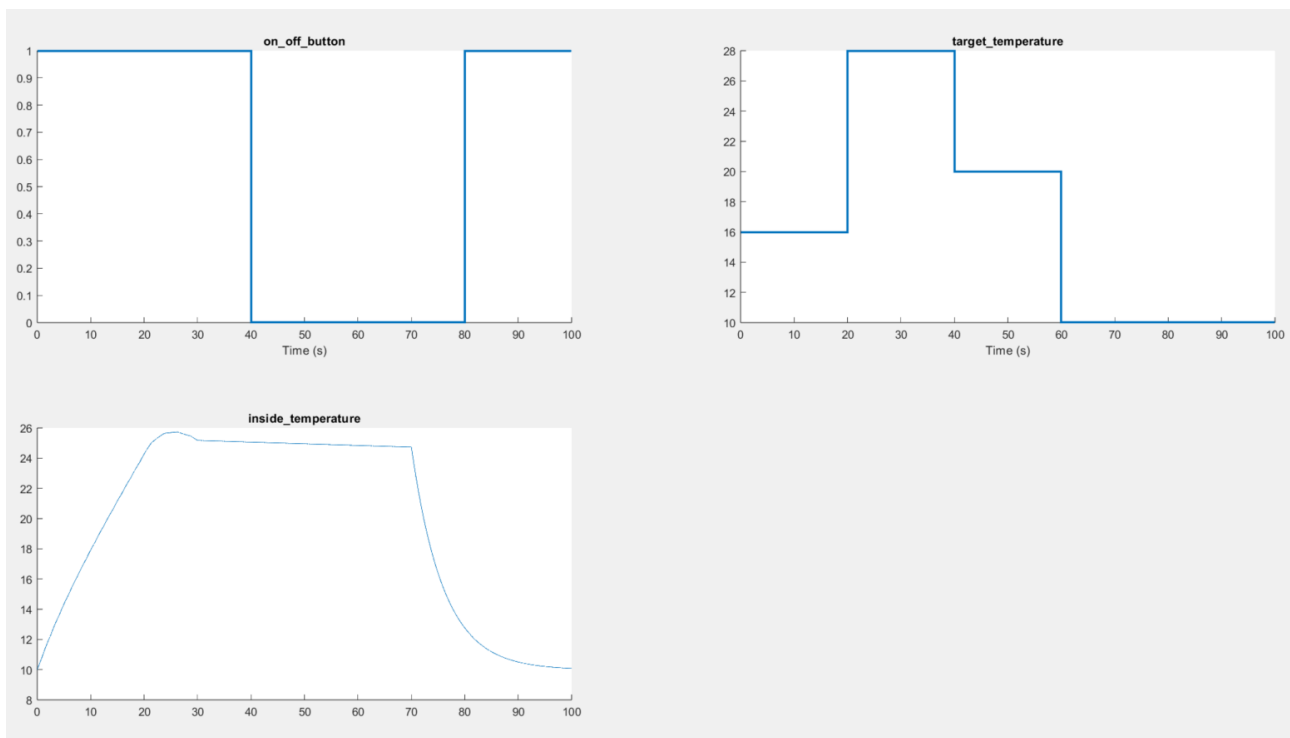


Figure 14: Ergebnis des Integrationstests

Der Integrationstest zeigt, dass das gesamte Software Model erfolgreich ausgeführt werden kann. Hier ist bereits zu sehen, dass sich die Temperatur gehalten wird, wenn der button ausgeschalten ist. Es wird keine Leistung mehr abgegeben und die Temperatur fällt nur leicht ab, weil sie von der Außentmeperatur beeinflusst wird.

## 10 Softwaretests

### 10.1 Testdokumentation

Zum Ausführen der Software-Tests muss zuerst das init-Skript ausgeführt werden, um alle Variablen aus "params/ac\_params.m" im Workspace zu speichern. Danach kann in "Klimaanlage/tests/qualification\_tests/" das "run\_qualification\_tests.m" Skript ausgeführt werden, mit dem die Software Tests gestartet und geplottet werden.

Andernfalls kann das "run\_all\_tests.m" Skript ausgeführt werden wodurch alle Tests gestartet werden aber auch die Software-Tests.

### 10.2 Testergebnisse Qualification Test 1

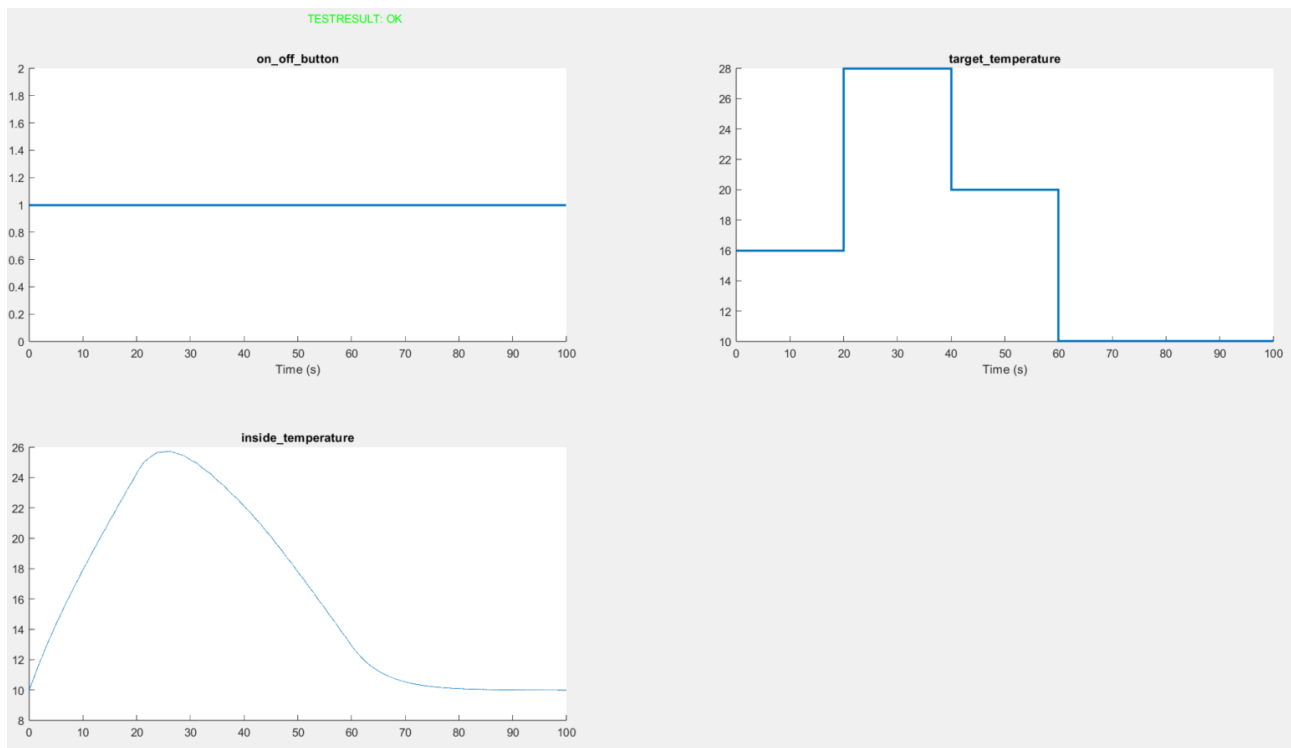


Figure 15: Ergebnis des ersten Qualification Tests

Mit den Qualification Tests wurde gezeigt, dass die generierte Software funktional so funktioniert, wie sie in den Requirements ausgelegt wurde. Der erste Test überprüft grundsätzlich, ob die vom Nutzer eingestellte Zieltemperatur erreicht wird. Dies war erfolgreich und wird mit einem grünen OK am oberen Rand signalisiert. Explizit wurden die Werte 28°C und 10°C verifiziert.

### 10.3 Testergebnisse Qualification Test 2

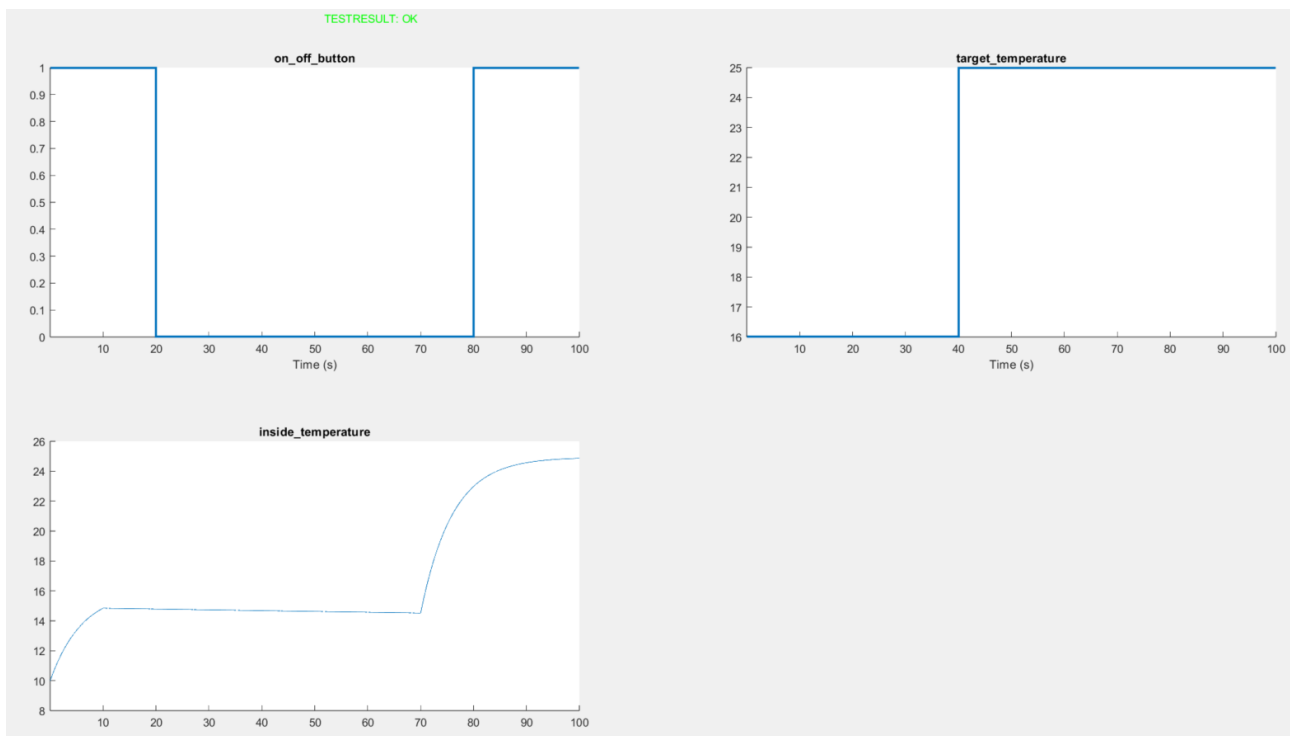


Figure 16: Ergebnis des zweiten Qualification Tests

Der zweite Qualification Test überprüft die funktionalität des on-off Buttons. Wenn dieser off ist, dann soll die Temperatur nicht geregelt werden, auch wenn eine Solltemperatur vom Nutzer anliegt. Dies war erfolgreich und wird mit einem grünen OK am oberen Rand signalisiert. Die Temperatur wird bei 16°C noch eine Weile gehalten, bevor die vom Nutzer eingestellten 25 angesteuert werden sobald der Button auf on gesetzt ist.