

# Programmentwurf

## Entwicklung eines rudimentären Zeitscheibensystems in C

Studiengang: Embedded Systems

Studienrichtung: Automotive Engineering

Duale Hochschule Baden-Württemberg Ravensburg, Campus Friedrichshafen

von

Fabian Klotz

Abgabedatum: 09.04.2023

Kurs: TSA22

Studiengangsleiter: Prof. Dr. Florian Leitner-Fischer

# Erklärung

gemäß Ziffer 1.1.13 der Anlage 1 zu §§ 3, 4 und 5 der Studien- und Prüfungsordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden-Württemberg vom 29.09.2017 in der Fassung vom 25.07.2018.

Ich versichere hiermit, dass ich meinen Programmentwurf mit dem Thema:

*Programmentwurf - Entwicklung eines rudimentären Zeitscheibensystems in C*

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Weingarten, den 18. März 2023

---

Fabian Klotz

---

# Kurzfassung

Im Rahmen des Programmentwurfs im Studiengang TSA22/TSL22 sollte ein Entwicklung eines rudimentären Zeitscheibensystems in C entworfen werden. Hierbei waren einige Anforderungen zu erfüllen, wie zum Beispiel eine sehr große Variabilität des Programms, sodass mit möglichst wenig Aufwand neue Tasks angelegt, oder zu anderen Zykluszeiten umgesetzt werden können.

Mein Programm basiert hierbei auf dem Konzept des Threadings um die zeitliche Abfolge zu organisieren. In den einzelnen Tasks können mathematische Funktionen mit einer bestimmten Priorität (also Vorgabe der auszuführenden Reihenfolge) konfiguriert werden. Sollte eine Funktion länger zur Ausführung brauchen wie die vorgegebene Zykluszeit, wird der Task erst neu gestartet, wenn er regulär wieder an der Reihe wäre, damit keine Asymmetrie in der zeitlichen Abfolge entsteht.

Abschließend wurde der Worst-Case Jitter der einzelnen Tasks berechnet.

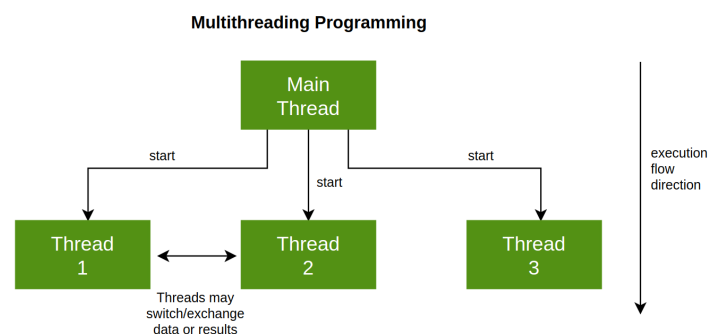
# Inhaltsverzeichnis

<b>1</b>	<b>Konzeptentwurf</b>	<b>1</b>
<b>2</b>	<b>Umsetzung</b>	<b>2</b>
	<b>Sachwortverzeichnis</b>	<b>3</b>
	<b>Abbildungsverzeichnis</b>	<b>3</b>
	<b>Anhang A</b>	<b>5</b>
A.1	Details zu bestimmten theoretischen Grundlagen . . . . .	5
A.2	Weitere Details, welche im Hauptteil den Lesefluss behindern . . . . .	5
	<b>Anhang B</b>	<b>6</b>
B.1	Versuchsanordnung . . . . .	6
B.2	Liste der verwendeten Messgeräte . . . . .	6
B.3	Übersicht der Messergebnisse . . . . .	6
B.4	Schaltplan und Bild der Prototypenplatine . . . . .	6
	<b>Anhang C</b>	<b>8</b>
C.1	Struktogramm des Programmentwurfs . . . . .	8
C.2	Wichtige Teile des Quellcodes . . . . .	8
	<b>Anhang D</b>	<b>9</b>
D.1	Einbinden von PDF-Seiten aus anderen Dokumenten . . . . .	9
	<b>Abbildungen</b>	<b>13</b>
	<b>Abbildungsverzeichnis</b>	<b>14</b>

# 1 Konzeptentwurf

Um ein passendes Konzept für die Umsetzung des Programms zu finden, wurden im Voraus einige Überlegungen, vor allem zur Steuerung der zeitlichen Abfolge angestellt. Mit dem Konzept des Threadings kann sowohl die zeitliche Abfolge, als auch der Aufruf der einzelnen Funktionen in den Tasks effizient gesteuert werden.

Die Abbildung soll Threading bildhaft darstellen.



**Abbildung 1.1:** Threading

<https://www.baeldung.com/wp-content/uploads/sites/4/2020/07/multithreading.png>

Der „Main Thread“ startet mehrere untergeordnete Threads. Alle Threads greifen auf die gleichen Ressourcen und globalen Variablen zu. Da alle Threads gleichzeitig ausgeführt werden, werden auch die Aufgaben in den Threads gleichzeitig ausgeführt. In unserem Programmentwurf sollten mehrere Tasks zu bestimmten Zykluszeiten ausgeführt werden. In meiner Umsetzung sollte es in jedem Thread eine Schleife, die den Thread jeweils so lange pausiert, bis der nächste Zyklus ansteht.

## 2 Umsetzung

... Text Umsetzung: Beschreibung der Umsetzung und eigener Untersuchungen ...

# Abbildungsverzeichnis

1.1	Threading . . . . .	1
-----	---------------------	---



Abbildungsverzeichnis

# Anhang A

A.1 Details zu bestimmten theoretischen Grundlagen

A.2 Weitere Details, welche im Hauptteil den  
Lesefluss behindern

# Anhang B

B.1 Versuchsanordnung

B.2 Liste der verwendeten Messgeräte

B.3 Übersicht der Messergebnisse

B.4 Schaltplan und Bild der Prototypenplatine

Diese Seite wurde eingefügt, um zu zeigen, wie sich der Inhalt der Kopfzeile automatisch füllt.

# Anhang C

C.1 Struktogramm des Programmentwurfs

C.2 Wichtige Teile des Quellcodes

# Anhang D

## D.1 Einbinden von PDF-Seiten aus anderen Dokumenten

Auf den folgenden Seiten wird eine Möglichkeit gezeigt, wie aus einem anderen PDF-Dokument komplette Seiten übernommen werden können. Der Nachteil dieser Methode besteht darin, dass sämtliche Formateinstellungen (Kopfzeilen, Seitenzahlen, Ränder, etc.) auf diesen Seiten nicht angezeigt werden. Die Methode wird deshalb eher selten gewählt. Immerhin sorgt das Package „*pdfpages*“ für eine korrekte Seitenzahleinstellung auf den im Anschluss folgenden „nativen“ L<sup>A</sup>T<sub>E</sub>X-Seiten.

Eine bessere Alternative ist, einzelne Seiten mit „*\includegraphics*“ einzubinden. Z.B. wenn Inhalte von Datenblättern wiedergegeben werden sollen.









# Abbildungen

# Abbildungsverzeichnis