

Hochschule Bremerhaven
Fakultät II – Management und Informationssysteme
Informatik
Modul Eingebettete Systeme

Design einer I^2C -Schnittstelle

von

Christopher Stieglitz Matrikel-Nr. 38785

25. April 2024

Inhaltsverzeichnis

I. System-Projektmanagementplan (SPMP)	5
1. Einleitung	7
1.1. Motivation für das Dokument	7
1.2. Projektübersicht	7
1.3. Projekterzeugnisse	7
2. Projektorganisation	8
2.1. System-Prozessmodell	8
2.2. Rollen und Verantwortlichkeiten	8
2.3. Werkzeuge und Techniken	8
2.4. Vorkehrungen zur Arbeitssicherheit	8
3. Projektmanagementplan	9
3.1. Aufgaben	9
3.1.1. Planung	9
3.1.2. Dokumentation	9
3.1.3. Entwicklung	10
3.1.4. Testen	11
3.2. Zeitplan	11
4. Zusatzmaterial	12
II. System-Entwicklungsbeschreibung (SDD)	13
5. Einleitung	15
5.1. Designübersicht	15
5.2. Anforderungsverfolgungsmatrix	15
6. Systemarchitektur	16
6.1. Gewählte Systemarchitektur	16
6.2. Diskussion alternativer Designs	16
6.2.1. Experiment-N	16
6.3. Systemschnittstellenbeschreibung	16
6.3.1. Schnittstelle-N	16
7. Benutzerschnittstellendesign	17
7.1. Beschreibung der Benutzerschnittstelle	17
7.1.1. Bildschirmfotos / Fotos der Benutzeroberfläche	17
7.1.2. Objekte und Aktionen	17
8. Detaillierte Beschreibung der Komponenten	18
8.1. Baustein/Komponente-N	18
9. Zusatzmaterial	19
III. System-Testdefinition (STD)	21
10. Einleitung	24
10.1. Systemübersicht	24
10.2. Testverfahren	24

11. Testplan	25
11.1. Zu testende Produktmerkmale	25
11.2. Nicht zu testende Produktmerkmale	25
11.3. Testwerkzeuge und Testumgebung	25
12. Testfälle	26
12.1. Testfall-N	26
12.1.1. Berührte Anforderungen	26
12.1.2. Aufgabe	26
12.1.3. Eingaben und Einstellungen	26
12.1.4. Erwartete Ausgaben und Erfüllungskriterien	26
12.1.5. Testdurchführung	26
13. Zusatzmaterial	27
A. Testprotokolle	28
A.1. Testergebnisse	28
A.2. Fehlerberichte	28
IV. Literatur	29

Eidesstattliche Erklärung

Mit der Einreichung dieses Dokuments erklären und versichern wir, dass dieses Dokument selbstständig verfasst wurde und keine anderen Quellen oder Hilfsmittel als die genannten benutzt wurden und die Stellen des Dokuments, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen wurden, in jedem Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht worden sind.

Teil I.

**System-Projektmanagementplan
(SPMP)**

System Project-Management Plan

Design einer I^2C -Schnittstelle

Informatik – Eingebettete Systeme – Hochschule Bremerhaven

Version	Autor*inn*en	geändert	Beschreibung	fertig am
0.1	Stieglitz	Abschnitte 1 bis 4	Erstellung des Dokumentes	18.04.2024
0.1.1	Stieglitz	Abschnitt 2.4	Abschnitt Arbeitssicherheit erstellt und ausgefüllt	18.04.2024
0.1.2	Stieglitz	Abschnitt 3.2	Abschnitt 3.2 aktualisiert	18.04.2024
0.1.3	Stieglitz	Abschnitt 3.1	Abschnitt fertiggestellt	25.04.2024

Tabelle 1.: Versionshistorie

Relevante Standards und Normen

- IEEE 16326-2009 – ISO/IEC/IEEE International Standard – Systems and Software Engineering – Life Cycle Processes – Project Management
[5]
- IEEE 16085-2006 – ISO/IEC 16085:2006, Standard for Software Engineering – Software Life Cycle Processes – Risk Management
[4]

1. Einleitung

1.1. Motivation für das Dokument

In diesem Dokument liegt das Konzept, wie Sie die Projektaufgabe angehen, Ihre Entscheidungen nachvollziehbar gestalten und Transparenz herstellen wollen. Dieser Hinweistext muss für die letzte Abgabe im Semester gestrichen sein.

1.2. Projektübersicht

Diese Übersicht zeigt das Projekt ganz allgemein. Das Projekt befasst sich mit der I2C-Kommunikation der Evaluationsboards mit einem ATmega644PA.

Es gibt mehrere dieser Boards, die untereinander kommunizieren sollen. In diesem speziellen Fall soll eine Kommunikation unter Hardwarekomponenten simuliert werden. Aufgabe ist es die Schnittstelle der I2C Kommunikation zu programmieren, zu testen und zu spezifizieren, damit sie im zuvor beschriebenen System eingesetzt werden kann.

1.3. Projekterzeugnisse

- Spezifikation (SPMP, SRS, SDD, STD)
- Programmcode - auf jeden ATmega644PA lauffähig
- Testprotokoll (STD)
- Testaufbau

2. Projektorganisation

2.1. System-Prozessmodell

Hiermit sind der Software- und der Hardware-Entwicklungsprozess gemeint. Sie schreiben hier, nach welchen Standards und Vorgehensmodellen Sie arbeiten. ENTFÄLLT

2.2. Rollen und Verantwortlichkeiten

ENTFÄLLT

2.3. Werkzeuge und Techniken

TODO

Hier stehen als Liste mit Erläuterungen Werkzeuge und wofür diese benutzt werden und insbesondere auch, wenn Werkzeugketten eingesetzt werden, werden diese hier beschrieben.

Werkzeuge und Techniken schließt auch alles ein, was Sie unternehmen, um die Qualität Ihres Projektes und Produktes sicherzustellen.

2.4. Vorkehrungen zur Arbeitssicherheit

Gefahr	Maßnahme
Vergiftung (durch z.B. Blei im Lötzinn)	Essen im Labor verboten
Verletzungen	Verbandskasten an der Labortür. Pflegen des Verbandsbuch auch wenn kein Material entnommen wurde
Stromschlag & elek. Gefahren	Notaus neben der Tür, bzw. nicht akut: Stecker vom Gerät entfernen und melden. Nicht in der Nähe von elek. Gerät trinken.
Brand und überhitzte Geräte	Feuerlöscher neben der Tür. Bei nicht akuter Gefahr Handschuhe bei Feuerlöscher um Gerät vorsichtig aus dem Fenster fallen zu lassen.
Feueralarm	Feuer/Rauchmelder im Labor verteilt, Alarmschalter in der Hochschule verteilt. Bei Alarm nächsten, nicht blockierten Notausgang nehmen. Im Labor ist das die Fenstertür, Sammelplatz ist die Wiese an der Geeste.
Personenschaden bei Feueralarm	Bei Feueralarm ist die Ruhe zu wahren, Fehlende Personen, die sich im Gebäude befinden sind zu melden, Personen die auf dem Weg verunfallen helfen.
Gefahrenvermeidung	Sicherheitsunterweisung für jeden Laborteilnehmer
Personenschaden	Sicht/Rufschutz durch eine weitere Person im Labor. Anders gesagt: nicht alleine im Labor arbeiten, bzw. zumindest einer Person bescheid sage.
Sonstiges	Keine Tiere im Labor
Vergiftung/Schaden am Gerät	Der Arbeitsplatz ist stets sauber zu halten, Geräte müssen vor und nach jeder Benutzung auf Schäden gesichtet werden.
Persönliche Schutzausrüstung	Wenn vorhanden ist die PSA situativ anzulegen. Die PSA darf keine Schäden aufweisen
Brand, Gas, sonstige	Versuchsaufbauten, die unabhängig laufen müssen vor defekten abgesichert sein

Tabelle 2.1.: Arbeitssicherheit

3. Projektmanagementplan

3.1. Aufgaben

3.1.1. Planung

3.1.1.1. Beschreibung

Die Planung ist der erste Schritt eines Projektes, wird aber mindestens mit anfang jedes Meilensteins fortgeführt. Es ist eine sehr grobe Aufgabe und umfasst die Planung des Projektes selbst, in Aufgabe, Ziel und Vorgehen, sowie allen Komponenten im Projekt, aber auch außerhalb, wenn es um Zeitplanung usw. geht. Die Planung enthält spezifisch diese Teilaspekte:

- Planung des Projekts (Aufgabe/Zieldefinition)
- Planung des Zeitmanagement, Meilensteine, Terminabsprachen
- Planung der Teilaufgaben in Vorgehen und Ziel
- Vorbereitung um Teilaufgaben durchführen zu können
- Ausfüllen der ersten Projektspezifikationen

3.1.1.2. Erzeugnisse und Meilensteine

Erzeugnis der Planung ist ein Backlog von Aufgaben, ein Zeitstrahl zur Abarbeitung der Aufgaben und Meilenstein als Miniziele, nachdem sich der Backlog und die Zeiteinteilung nochmal angeschaut werden muss, ob noch alles passt. Die Planung muss in jedem Meilenstein durchgeführt werden und bringt pro Meilenstein mindestens eine Iteration des Systemspezifikation-Dokument.

3.1.1.3. Benötigte Ressourcen

- Latex - Festhalten der Planungsergebnisse in der Spezifikation
- Tool für Zeit- und Aufgabenmanagement (z.B. Jira, Gitlab) - hier Notion
- Kommunikationstool zur Absprache mit Projektmitarbeiter, Professor und ggf. weiteren

3.1.1.4. Abhängigkeiten und Bedingungen

Die Planung hängt von den Informationen ab, die zur Verfügung stehen, wie z.B. Aufgabenbeschreibung usw.

3.1.1.5. Risiken

Risiken bei der Planung ist ein überplanen, sodass die Arbeit an dem eigentlichen Projekt zu kurz kommt. Andererseits sollte soviel geplant werden, dass es die Arbeit am Projekt erleichtert, indem es eine Richtung und Ziel vorgibt. Gegenmaßnahmen sind Aufgaben die noch weit vorausliegen nur grob zu definieren.

3.1.2. Dokumentation

3.1.2.1. Beschreibung

Ähnlich wie die Planung ist die Dokumentation ebenfalls in jedem Meilenstein vorhanden. Sie ist sogar ein Teilaspekt der Planung, da ohne Dokumentation die Planung nutzlos wäre. Die Aufgabe erfüllt die Aufgabe, dass jeder Projektmitarbeiter den genauen Stand des Projektes wissen und leicht anknüpfen können, aber auch nach dem Projekt jeder nachvollziehen kann, welche Schritte wie und warum gemacht wurden und das fertige Produkt richtig anwenden können. Es ist wichtig, dass WIE und WARUM, bzw. Fehlschläge ebenfalls zu Dokumentieren, damit auftretende Fehler oder ggf. Schritte zu Erweiterungen, besser identifizierbar sind.

3.1.2.2. Erzeugnisse und Meilensteine

Erzeugnis der Dokumentation ist die vollständige Systemspezifikation mit allen Versuchsaufbauten und Beschreibungen usw., sowie ein sorgfältig Kommentierter, oder sogar dokumentierter Code.

3.1.2.3. Benötigte Ressourcen

- Latex - Festhalten der Planungsergebnisse in der Spezifikation
- Editor der Wahl - um Code-Dokumentation durchzuführen

3.1.2.4. Abhängigkeiten und Bedingungen

Die Aufgabe ist unabhängig von anderen Aufgaben und sollte bedingungslos durchgeführt werden.

3.1.2.5. Risiken

- Unsaubere Dokumentation
 - Dokumentation gegenlesen lassen, bzw. Projektaufbau von jemanden anderes nur anhand der Doku aufbauen und durchführen lassen
 - Nach jedem Schritt dokumentieren
- Fehler in der Dokumentation
 - Regel: Keine Aktion ohne vorherige und sofortig anschließende dokumentation

3.1.3. Entwicklung

3.1.3.1. Beschreibung

Die Entwicklungsarbeiten beinhalten auch etwas Testen. Grundlegend geht es darum sich der Problemlösung zu nähern, bis es erfüllt ist. Entwickelt wird immer Schrittweise, je nach Planung und immer begleitend mit der Dokumentation. Dennoch ist es die Hauptaufgabe, die am meisten Zeit in Anspruch nehmen sollte.

3.1.3.2. Erzeugnisse und Meilensteine

Bei fertigstellung eines geplanten Abschnittes, sollten auch die geplanten Erzeugnisse entstanden sein. Alles in allem ist alles was bei dem Projekt entstanden ist, das Erzeugniss der Entwicklungsarbeit. Im klaren sind das:

- Ein Testaufbau zur Demonstration mit mindestens 3 Boards
- Die Software-Schnittstelle (Code), dokumentiert und im Git
- Ein Demo-Programm
- Testergebnisse

3.1.3.3. Benötigte Ressourcen

- Git-Repo
- 3xEvaluationsboard mit ATmega644PA
- Laborrechner
- Editor der Wahl

3.1.3.4. Abhängigkeiten und Bedingungen

Die Entwicklung hängt indirekt von der Planung ab, da die Planung den Lauf der Entwicklung steuern sollte. Bedingungen sind vor allem technische, dass die Hardware bereitsteht, usw.

3.1.3.5. Risiken

- Elektronische Gefahren:
 - Kurzschluss durch falsch gesetzte Kabel
 - Kurzschluss durch Wasser und anderen Verunreinigungen
 - Stromschlag durch offene Kabel & unsachgemäßen Umgang
 - -> Aufmerksamkeit schaffen und Abschnitt 2.4 Arbeitssicherheitsunterweisung Labor
- Ausfall eines Boards, oder Funktionseinschränkungen
 - Board austauschen

3.1.4. Testen

3.1.4.1. Beschreibung

Gute Tests spiegeln die Qualität des Produktes wieder, indem sie die allgemeine Funktionalität, sowie das Verhalten bei Randbedingungen sicherstellen. Da Tests des gesamten Systems schwer sind, werden die einzelnen Softwarekomponenten mit Realistischen und Kritischen Eingaben getestet.

3.1.4.2. Erzeugnisse und Meilensteine

Die Tests werden als Code geschrieben und durchgeführt und dann im Abschnitt 15 dokumentiert.

3.1.4.3. Benötigte Ressourcen

Die Ressourcen sind gleich wie in der Entwicklung, da das Entwickelte getestet wird.

- 3xEvaluationsboard mit ATmega644PA
- Laborrechner
- Editor der Wahl

3.1.4.4. Abhängigkeiten und Bedingungen

Die Tests sind stark von der Entwicklung abhängig und können nur bei fertigen Abschnitten angewandt werden.

3.1.4.5. Risiken

Risiken beim Testen gibt es so keine. Die Qualität der Tests ist schwierig zu messen. Erfolg oder Misserfolg ist beim Testen das Produkt und kein Risiko.

3.2. Zeitplan

Beginn: Aufgabebesprechung - 13.03.2024

Meilenstein 1 - Vorbereitung: geplant 24.03.24; abgeschlossen 21.04.2024

Meilenstein 2 - Boardkommunikation: 28.04.2024

Meilenstein 3 - I2C Datenversand: 03.05.2024

Meilenstein 4 - Feinschliff: 12.05.2024

Endabgabe 4: 16.05.2024

4. Zusatzmaterial

Teil II.

**System-Entwicklungsbeschreibung
(SDD)**

System Design-Description

Design einer I^2C -Schnittstelle

Informatik – Eingebettete Systeme – Hochschule Bremerhaven

Version	Autor*inn*en	geändert	Beschreibung	fertig am
0.1	Stieglitz	Abschnitt	WAS	25.04.2024

Tabelle 4.1.: Versionshistorie

Relevante Standards und Normen

- IEEE 1016-2009 – IEEE Standard for Information Technology – Systems Design – Software Design Descriptions
[3]

5. Einleitung

5.1. Designübersicht

5.2. Anforderungsverfolgungsmatrix

Anforderung	Baustein-1	Baustein-2	Baustein-3	Baustein-4	...
REQ-EXT-USER-1					
REQ-EXT-USER-2					
REQ-EXT-HW-1					
REQ-EXT-HW-2					
REQ-EXT-HW-3					
REQ-EXT-PROD-1					
...					

6. Systemarchitektur

6.1. Gewählte Systemarchitektur

Hier steht die Architektur, die Sie ausgewählt haben, wenn es andere Möglichkeiten gibt, die Sie betrachtet haben, dann stehen diese im nächsten Punkt, der Diskussion alternativer Designs

6.2. Diskussion alternativer Designs

Dieser Abschnitt enthält alles, was Sie zur Lösungsfindung probiert und diskutiert haben. Insbesondere schreiben Sie hier auch Experimente, deren Aufbau, Durchführung und deren Ergebnisse auf.

6.2.1. Experiment-N

6.2.1.1. Zweck

6.2.1.2. Aufbau

6.2.1.3. Durchführung

6.2.1.4. Ergebnisse

6.3. Systemschnittstellenbeschreibung

6.3.1. Schnittstelle-N

Gliedern Sie die Schnittstellenbeschreibung passend auf.

6.3.1.1. Kommunikation: Initiierung, Nachrichten und Ende

6.3.1.2. Zeitverhalten

Hier steht, wie sich die Beteiligten der Schnittstelle zeitlich verhalten sollen. Z.B.

- *bei sehr vielen Nachrichten („Meldungsschauer“ (engl. „burst“))*
- *Timeouts*
- *Timing-Diagramme!*

6.3.1.3. Fehlerbehandlung

7. Benutzerschnittstellendesign

Benutzerschnittstellen sind durchaus auch Systemschnittstellen. Sie sind besonders, da hier nicht mit technischen Systemen interagiert wird, sondern mit Menschen. Man könnte diesen Punkt auch Mensch/Maschine-Kommunikation bezeichnen. Taster und LEDs gehören zur Benutzerschnittstelle.

7.1. Beschreibung der Benutzerschnittstelle

7.1.1. Bildschirmfotos / Fotos der Benutzeroberfläche

7.1.2. Objekte und Aktionen

8. Detaillierte Beschreibung der Komponenten

8.1. Baustein/Komponente-N

Hier informieren Sie über

- *die interne Struktur des Bausteins/der Komponente, in Hardware und in Software*
- *die physikalischen und logischen Eigenschaften und deren Aufteilung in Hardware und Software*
- *interne Zustände (sofern es das gibt)*
- *interne Abläufe (Prozesse)*
- *Datenstrukturen*
- *Initialisierung*
- *...*

9. Zusatzmaterial

Teil III.

**System-Testdefinition
(STD)**

System Test-Definition

Design einer I^2C -Schnittstelle

Informatik – Eingebettete Systeme – Hochschule Bremerhaven

Version	Autor*inn*en	geändert	Beschreibung	fertig am
0.1	Stieglitz	Abschnitt	WAS	18.10.2023

Tabelle 9.1.: Versionshistorie

Relevante Standards und Normen

- IEEE 829-2008 – IEEE Standard for Software and System Test Documentation
[9]
- IEEE 1008-1987 – IEEE Standard for Software Unit Testing
[1]
- IEEE 1012-2016 – IEEE Standard for System, Software, and Hardware Verification and Validation
[2]
- IEEE 29119-2-2013 – ISO/IEC/IEEE International Standard – Software and systems engineering – Software testing – Part 2: Test processes
[6]
- IEEE 29119-3-2013 – ISO/IEC/IEEE International Standard – Software and systems engineering – Software testing – Part 3: Test documentation
[7]
- IEEE 29119-4-2015 – ISO/IEC/IEEE International Standard – Software and systems engineering – Software testing – Part 4: Test techniques
[8]

Im Dokument STD werden Testfälle definiert. Diese Testfälle dienen der Prüfung des fertigen Produkts und prüfen die Anforderungen aus dem SRS-Dokument. Die gesamte Spezifikation beschreibt, welche Anforderungen das Produkt hat, wie man es herstellen und wie man es prüfen kann. Ein Testprotokoll, welches tatsächlich durchgeführte Prüfungen beschreibt, gehört in den Abschnitt Testprotokoll.

10. Einleitung

10.1. Systemübersicht

10.2. Testverfahren

11. Testplan

11.1. Zu testende Produktmerkmale

11.2. Nicht zu testende Produktmerkmale

11.3. Testwerkzeuge und Testumgebung

12. Testfälle

12.1. Testfall-N

12.1.1. Berührte Anforderungen

Ein Beispiel: Dieser Testfall prüft die folgenden Anforderungen ab:

- *REQ-SYS-EXT-USER-1 Titel*
- *REQ-PROD-2*

12.1.2. Aufgabe

12.1.3. Eingaben und Einstellungen

12.1.4. Erwartete Ausgaben und Erfüllungskriterien

12.1.5. Testdurchführung

13. Zusatzmaterial

A. Testprotokolle

A.1. Testergebnisse

A.2. Fehlerberichte

Teil IV.

Literatur

Literatur

- [1] IEEE. “1008-1987 – IEEE Standard for Software Unit Testing”. In: *IEEE Standards* (1987). URL: <https://standards.ieee.org/standard/1008-1987.html>.
- [2] IEEE. “1012-2016 – IEEE Standard for System, Software, and Hardware Verification and Validation”. In: *IEEE Standards* (2016). URL: <https://standards.ieee.org/standard/1012-2016.html>.
- [3] IEEE. “1016-2009 – Systems Design – Software Design Descriptions”. In: *IEEE Standard for Information Technology* (2009). URL: <https://standards.ieee.org/standard/1016-2009.html>.
- [4] IEEE. “16085-2006 – Software Life Cycle Processes – Risk Management”. In: *ISO/IEC/IEEE International Standard – Systems and Software Engineering* (2006). URL: <https://standards.ieee.org/standard/16085-2009.html>.
- [5] IEEE. “16326-2009 – Life Cycle Processes – Project Management”. In: *ISO/IEC/IEEE International Standard – Systems and Software Engineering* (2009). URL: <https://standards.ieee.org/standard/16326-2009.html>.
- [6] IEEE. “29119-2-2013 – Software testing – Part 2:Test processes”. In: *ISO/IEC/IEEE International Standard – Systems and Software Engineering* (2013). URL: <https://standards.ieee.org/standard/29119-2-2013.html>.
- [7] IEEE. “29119-3-2013 – Software testing – Part 3:Test documentation”. In: *ISO/IEC/IEEE International Standard – Systems and Software Engineering* (2013). URL: <https://standards.ieee.org/standard/29119-3-2013.html>.
- [8] IEEE. “29119-4-2013 – Software testing – Part 4:Test techniques”. In: *ISO/IEC/IEEE International Standard – Systems and Software Engineering* (2013). URL: <https://standards.ieee.org/standard/29119-4-2013.html>.
- [9] IEEE. “829-2008 – IEEE Standard for Software and System Test Documentation”. In: *IEEE Standards* (2008). URL: <https://standards.ieee.org/standard/829-2008.html>.