

HOCHSCHULE BREMERHAVEN

EMBEDDED SYSTEMS

Software Project Management Plan

Description of the software approach and associated milestones.

Autoren: Jan Löwenstrom (34937) und Johann Hoffer (34461)
begleitet von
Prof. Dr. Lipskoch

1. Februar 2019

Inhaltsverzeichnis

1	Introduction	2
1.1	Project Overview	2
1.2	Project Deliverables	2
2	Project Organization	3
2.1	Software Process Model	3
2.2	Roles and Responsibilities	3
2.3	Tools and Techniques	3
2.3.1	Git	3
2.3.2	simavr	4
2.3.3	GTK-Wave	4
3	Project Management Plan	4
3.1	Tasks	4
3.2	Task01 - Qualitätsszenarios	5
3.3	Task02 - Anforderungen festlegen	6
3.4	Task03 - Testfälle	6
3.5	Task04 - Timer	7
3.6	Task05 - Datenstrukturen	8
3.7	Task06 - DCF abtasten	8
3.8	Task07 - Validierung	9
3.9	Task08 - Hochzählen der internen Zeit	10
3.10	Task09 - Darstellung auf Display	11
3.11	Task10 - Lötboard	12
3.12	Task11 - Inbetriebnahme	13
3.13	Timetable	14

1 Introduction

Dieses Projekt findet im Modul Eingebettete System im 5. Semester im Studiengang Informatik statt und wird begleitet von Herrn Professor Lipskoch. In diesem Projekt geht es um die komplette Entwicklung einer Uhr, deren Zeit aus dem DCF-77 Signal stammt. Dazu gehören das Zusammenstellen mehrerer Hardwarekomponenten, die über verschiedene Schnittstellen miteinander verbunden sind, und das Entwickeln einer Software, durch die die Hardware die Eingangssignale auswertet und eine gewünschte Ausgabe liefert. Dieses System wird für einen Kunden hergestellt, der die wesentlichen Anforderungen und Rahmenbedingungen vorgibt. In diesem Fall handelt es sich dabei um Professor Dr. Lipskoch.

1.1 Project Overview

Das Ziel ist es, das DCF-Signal korrekt auszuwerten und die darin enthaltenen Informationen auf einem Display darzustellen. Um das Projekt erfolgreich abschließen zu können, sind die im Folgenden genannten Schritte notwendig. Dazu müssen zunächst die Anforderungen an das System erhoben werden, welche aus der Software Requirement Specification entnommen werden können. Dazu zählen Hardware- und Softwareschnittstellen, die Softwarefeatures sowie die nicht-funktionalen Anforderungen. Außerdem gibt es einige Randbedingungen, die die zu benutzende Hardware und die Zeit für das Projekt eingrenzen. Dazu zählen auch zu verwendende Protokolle und die zu wählende Programmiersprache. Nach der Anforderungserhebung geht es um die Einrichtung der richtigen Timer, die für das Abtasten des DCF-Signals und das Weiterzählen der Zeit benötigt werden. Ein weiterer Milestone ist das Wählen einer passenden Speicherstruktur für das DCF-Signal. Danach wird das Abtasten des Signals programmiert, genauso wie passende Validierungsvorgänge nach dem erfolgreichen Auslesen, die teilweise auch für das Weiterzählen benötigt werden. Dafür wird eine Dekodierungsanleitung für das Signal benötigt. Zum Schluss muss ein passendes Display in Betrieb genommen und programmiert werden, um die gewünschten Informationen anzuzeigen, wofür die genauen Anweisungen des spezifischen Displays benötigt werden. Außerdem werden die Schnittstellen zwischen dem DCF-Empfänger und dem ATmega und zwischen dem ATmega und dem Display benötigt. Dafür werden vor allem Informationen über die elektrische Übertragung der Signale benötigt.

1.2 Project Deliverables

Der zu liefernde Gegenstand ist das komplette System, das erfolgreich eine DCF-Zeit anzeigt. Davon wird genau eines hergestellt. Als Lieferdatum ist der 25.01.2019 angegeben, an dem das System vorgestellt werden soll, was an der Hochschule Bremerhaven

geschieht.

2 Project Organization

2.1 Software Process Model

Die Entwicklung der Software fand testgetrieben statt. Wird diese Methode strikt umgesetzt, werden zunächst die Tests geschrieben und danach die funktionalen Methoden implementiert. In der Theorie arbeiten diese Methoden richtig, sobald alle Tests erfolgreich durchlaufen wurden. Die Tests können natürlich angepasst werden, falls Änderungen an den Anforderungen auftreten oder wenn trotz erfolgreicher Tests ein fehlerhaftes Verhalten festgestellt wird. Die Software wurde Komponentenweise entwickelt. Zu Anfang müssen immer sämtliche Informationen über die zu verwendene Hardware und die benötigten Schnittstellen beschaffen werden. Nur so kann die Software passend entwickelt werden und schließlich einwandfrei laufen. Als Kriterium für einen erfolgreichen Abschluss der Entwicklung ist der Echtttest ausschlaggebend.

2.2 Roles and Responsibilities

Die verschiedenen Rollen und Verantwortlichkeiten hatten jeweils beide Teammitglieder inne, da wir eine strenge Trennung in einem Team mit nur zwei Mitgliedern nicht für sehr sinnvoll erachteten. Darunter fallen die Anforderungserhebung, die Diskussion über verschiedene Implementierungsansätze, die Implementierung der Tests sowie die Implementierung der Funktionen. Aufgaben wie das Löten oder das Zusammenbauen von Hardware, die vor allem motorische Fähigkeiten erfordern, hat Herr Löwenstrom mit geistiger Unterstützung von Herrn Hoffer ausgeführt.

2.3 Tools and Techniques

2.3.1 Git

Als Hilfsmittel für die Softwareentwicklung kam die Versionskontrolle Git zum Einsatz. Das Git-Repository für unser Projekt befindet sich in der Hochschulinfrastruktur auf dem Server „hopper.hs-bremerhaven.de“. Über Git können mehrere Personen gleichzeitig an einem Projekt arbeiten und Veränderungen vornehmen. Außerdem können sie das Repository auf beliebig vielen Geräten „clonen“, und dort am Projekt weiterarbeiten. Beim Clonen“ wird eine lokale Kopie des Repositorys auf dem jeweiligen Gerät angelegt. So kann anschließend auch ohne Internetverbindung an dem Projekt gearbeitet werden. Die getätigten Veränderungen werden dann über sogenannte „Commits“ und anschließenden „Pushes“ an das Repository übertragen. Dabei muss nicht für

jeden „Commit“ „gepusht“ werden. Die „Commits“ stellen Zustände des Projekts zur Verfügung, zu denen jederzeit zurückgesetzt werden kann. Für den „Push“ wird wieder eine Internetverbindung benötigt. Haben mehrere Personen zu der selben Zeit an der gleichen Datei gearbeitet, so kann ein „Merge“ ausgeführt werden. Dabei werden manuell die zu Verbleibenden passagen ausgewählt, wobei zusätzliche Veränderungen möglich sind.

2.3.2 simavr

Bei simavr handelt es sich um eine Simulationssoftware für den ATmega32. Sie bietet den Vorteil, dass einige Funktionen getestet werden können, ohne dass die Software auf einen realen ATmega aufgespielt werden muss. Bei diesen Funktionen handelt es sich um jene, die etwas auf den Ports des Evaluationsboard ausgeben oder einlesen. Diese Funktionen können auf diese Weise ausgeführt werden, auch wenn kein realer ATmega zur Verfügung steht, zum Beispiel außerhalb des Labors. Ausführen bedeutet in diesem Fall das Erstellen einer VCD-Datei.

2.3.3 GTK-Wave

Aus der VCD Datei, die von simavr erzeugt wurde, kann mit GTK-Wave eine Visualisierung der Portzustände über einen bestimmten Zeitraum durchgeführt werden. Diese Zeiträume können dabei im Nanosekundenbereich liegen, was einen weiteren großen Vorteil bedeutet, da dieser vom menschlichen Auge nicht erfasst werden kann.

3 Project Management Plan

3.1 Tasks

In unserem Projekt gibt es verschiedene Arten von Tasks. Zunächst geht es um anforderungserhebende Aufgaben. Die erste davon ist das Aufstellen der Qualitätsszenarios. Diese basieren auf den vom Kunden gewünschten Funktionen, die das System erfüllen soll. Die nächste Aufgabe besteht darin, aus diesen Szenarien spezifische Anforderungen zu entwickeln. Zu diesen spezifischen Anforderungen können dann schon Testfälle aufgestellt werden, um später feststellen zu können, ob die Anforderungen erfüllt wurden.

Die nächste Art von Aufgaben stellen diejenigen aus der Softwareentwicklung dar. Dazu zählen die korrekte Programmierung der Timer, die Erstellung einer passenden Datenstruktur für die DCF-Folge und die daraus resultierende interne Zeit des ATMegas, das Abtasten des DCF-Signals, sämtliche Validierungsschritte, das Hochzählen der Zeit und zu guter Letzt die Darstellung auf dem Display. Für die Verwertung des

DCF-Signals muss recherchiert werden, wie es dekodiert wird und in welcher Form diese weitergeleitet werden. Beim Display muss die Ansteuerung hinreichend bekannt sein, sodass die Inhalte wie gewünscht angezeigt werden können.

Da wir mit verschiedenen Hardwarekomponenten arbeiten, müssen diese über die passenden Schnittstellen miteinander verbunden werden. Dafür haben wir ein eigenes Lötboard entwickelt, über das der ATmega mit dem DCF-Empfänger und dem Display verbunden ist. Für alle Komponenten muss bekannt sein, wie diese über elektrische Signale mit anderen Komponenten kommunizieren können. Beispielsweise muss mit den korrekten Spannungsleveln gearbeitet werden, damit es nicht zu Fehlern oder sogar zu Schäden an den elektrischen Bauteilen kommen kann.

Zum Schluss steht dann die entgeltliche Inbetriebnahme des Displays und des DCF-Empfängers in Verbund mit dem ATmega auf dem Evaluationsboard an. Es handelt sich also um die entgeltliche Inbetriebnahme des ganzen Systems. Sobald alles genauso wie vorher festgelegt funktioniert, ist diese Inbetriebnahme erfolgreich verlaufen.

3.2 Task01 - Qualitätsszenarios

Description

Dieser Task behandelt das Aufstellen der Qualitätsszenarios. Diese dienen als Einstieg für die Anforderungserhebung, da aus ihnen direkt spezifische Anforderungen abgeleitet werden können. Außerdem kann mit ihnen geplant werden, wie das System aussehen soll und teilweise auch Prioritäten festgelegt werden.

Deliverables and Milestones

Diese Aufgabe soll die aufgestellten Qualitätsszenarios liefern.

Resources Needed

Für diese Aufgabe werden die grundlegenden Anforderungen des Kunden benötigt.

Dependencies and Constraints

Diese Aufgabe ist zum Großteil von den Wünschen des Kunden abhängig. Beschränkungen werden durch die Randbedingungen verursacht. Dies können bestimmte Hardwarekomponenten und Schnittstellen sein, die verwendet werden müssen.

Risks and Contingencies

Es kann passieren, dass die aufgestellten Qualitätsszenarios teilweise nicht den Kundenwünschen entsprechen. Wenn dieser Fall eintritt, müssen diese angepasst werden.

Daraus können neue konkrete Anforderungen entstehen, verändert werden oder auch entfallen.

3.3 Task02 - Anforderungen festlegen

Description

Task 2 behandelt das Umsetzen der Qualitätsszenarios in spezifische Anforderungen. Diese dienen als Grundlage für die Entwicklung des ganzen Systems.

Deliverables and Milestones

Diese Aufgabe soll alle Anforderungen, die das System erfüllen soll, festlegen. Sie können aus den Qualitätsszenarien hervorgehen oder auch unabhängig von den Szenarien aufgestellt werden.

Resources Needed

Es werden die Qualitätsszenarien benötigt. Außerdem müssen die Beschränkungen auf bestimmte Hardwarekomponenten und Schnittstellen festgelegt worden sein.

Dependencies and Constraints

Wie bei den Qualitätsszenarios sind die Kundenwünsche maßgeblich für die Anforderungen. Bei Änderungen müssen die Anforderungen angepasst werden.

Risks and Contingencies

Auch bei dieser Aufgabe können sich die Anforderungen zu einem späteren Zeitpunkt ändern. Dann müssten diese ebenfalls angepasst werden.

3.4 Task03 - Testfälle

Description

Mit den Tests kann die Erfüllung der Anforderungen überprüft werden. Sie gehen aus den konkreten Systemanforderungen hervor.

Deliverables and Milestones

Es sollen verschiedene Testfälle geliefert werden, um in der Endphase des Projekts zu testen, ob das System die Anforderungen erfüllen kann.

Resources Needed

Es werden die Systemanforderungen benötigt.

Dependencies and Constraints

Die Testfälle sind komplett von den Anforderungen abhängig. Falls die Anforderungen geändert werden, müssen Testfälle erstellt oder modifiziert werden, oder sie können entfallen.

Risks and Contingencies

Falls mindestens ein Test fehlschlägt, muss die Ursache dafür ermittelt werden. Sie kann in der Software, Hardware oder in anderen äußeren Einflüssen, die beispielsweise das DCF-Signal stören, liegen. Sobald sie festgestellt wurde, muss ermittelt werden, ob die Behebung des Problems für uns möglich ist.

3.5 Task04 - Timer**Description**

Für unser System sind die Timer die wichtigste grundlegende Funktion. Auf ihnen basiert das Abtasten des DCF-Signals, da dabei jede Millisekunde geprüft werden muss, ob sich das Signal auf einem Low- oder High-State befindet. Je nachdem, wie oft in der Sekunde ein High-State festgestellt wurde, wird eine 0 oder eine 1 in der DCF-Folge gespeichert. Es wird dabei also auch der Sekundentimer benötigt, der jede Sekunde den Index auf die nächste Stelle der DCF-Folge setzt. Der Sekundentimer wird ebenfalls für das Hochzählen der internen Zeit des ATMegas und das Anzeigen dieser auf dem Display benötigt, da beides jeweils einmal in der Sekunde stattfinden muss.

Deliverables and Milestones

Nach Beendigung dieser Aufgabe sollen der sekundliche Timer, der für das Hochzählen der Zeit benötigt wird, und der Millisekundentimer, der für das Abtasten des DCF-Signals notwendig ist, zur Verfügung stehen.

Resources Needed

Es werden die Dokumentation des ATMegas, ein C-Compiler und die Simulationssoftware GTK-Wave benötigt.

Dependencies and Constraints

Damit die Timer funktionieren, muss der ATMEGA natürlich in einem vollkommen intakten Zustand sein.

Risks and Contingencies**3.6 Task05 - Datenstrukturen****Description**

Diese Aufgabe beinhaltet die Programmierung der Datenstrukturen für die DCF-Folge und die interne Zeit des ATMEGAs. Erst wenn diese bereitstehen, können die anderen Funktionen implementiert werden.

Deliverables and Milestones

Nach Beendigung dieser Aufgabe sollen die benötigten Datenstrukturen für die DCF-Folge und die interne Zeit des ATMEGAs zur Verfügung stehen.

Resources Needed

Es muss die Länge der DCF-Folge bekannt sein.

Dependencies and Constraints

Für die Datenstruktur der DCF-Folge muss deren korrekte Länge bekannt sein.

Risks and Contingencies

Die Länge des DCF-Arrays muss korrekt sein, ansonsten kann es zu unzulässigen Speicherzugriffen kommen.

3.7 Task06 - DCF abtasten**Description**

Bei dieser Aufgabe soll das Abtasten einer DCF-Folge programmiert werden, die vom DCF-Empfänger empfangen wird. Je nachdem, wie groß die Amplitude des Signals ist, liegt auf dem Datenübertragungskabel eine Spannung an oder nicht.

Deliverables and Milestones

Nachdem die Entwicklung der Methode fertiggestellt ist, sollen DCF-Folgen erfolgreich eingelesen werden können.

Resources Needed

Damit das Einlesen der DCF-Folge richtig implementiert werden kann, muss bekannt sein, wie der DCF-Empfänger funktioniert. Daunter fällt, in welchem Zustand des Signals eine 0 oder eine 1 übertragen wird.

Dependencies and Constraints

Für ein korrektes Empfangen der DCF-Folge wird ein einwandfrei funktionierender DCF-Empfänger zum Einsatz kommen müssen. Außerdem wird ein ungestörtes Signal ohne starkes Rauschen benötigt. Falls dem Empfänger kein gutes Signal zur Verfügung steht, ist es nicht möglich dieses zufriedenstellend auszuwerten. Dieses Problem kann auch nicht durch einen perfekten Abtast-Algorithmus egalisiert werden.

Risks and Contingencies

Das Risiko hierbei sind mehrere verschiedene Fehlerquellen. Der Empfänger könnte fehlerhaft arbeiten, das Signal könnte gestört sein, das Einlesen des Pins am ATmega könnte nicht richtig funktionieren oder der programmierte Algorithmus nicht geeignet sein. Das Risiko, eine falsche DCF-Folge eingelesen zu haben, muss durch die Validierungsvorgänge gemindert werden.

3.8 Task07 - Validierung**Description**

Die Validierung ist der wichtigste Vorgang bei der Auswertung der DCF-Folge. So kann zwischen fehlerfreien und fehlerbehafteten Folgen differenziert werden und es wird vermieden, dass nicht valide Zeiten als interne Zeit des ATmegas übernommen werden. Unsere Software bietet 2 Validierungsprozesse, zum Einen die Überprüfung der Paritäten in der DCF-Folge und zum Anderen die anschließende Validierung der dekodierten Dateninformationen.

Deliverables and Milestones

Nach Abschluss dieser Aufgabe sollen die Methoden zur Prüfung der Parität und zur Validierung einer Uhrzeit und eines Datums zur Verfügung stehen.

Resources Needed

Für die erfolgreiche Validierung der Paritäten muss bekannt sein, an welchen Positionen in der DCF-Folge sich die Paritätsbits befinden und ob mit einer geraden oder ungeraden Parität gearbeitet wird.

Dependencies and Constraints

Damit die DCF-Folge und die dekodierten Daten im Echtbetrieb des Systems erfolgreich validiert werden können, muss das Empfangen und Auswerten der Folge korrekt implementiert worden sein. Außerdem müssen die Timer richtig funktionieren, da der korrekte Rhythmus beim Auslesen eingehalten werden muss, um richtig auf die Werte der einzelnen Bits schließen zu können.

Risks and Contingencies

Es kann passieren, dass die Methoden nicht korrekt funktionieren und somit nicht valide DCF-Folgen, Daten oder Zeiten als richtig eingestuft werden. Genauso könnten diese als falsch eingestuft werden, auch wenn sie valide sind, sodass nie eine korrekte Zeit übernommen werden könnte. Diese Methoden müssen also ausreichend getestet worden sein, damit sie im realen System zum Einsatz kommen dürfen.

3.9 Task08 - Hochzählen der internen Zeit**Description**

Diese Aufgabe behandelt das Hochzählen der internen Zeit des ATMegas. Dies ist wichtig, damit auf dem Display die Zeit korrekt angezeigt wird, auch wenn kein DCF-Signal vorhanden ist, dieses gestört wird oder der DCF-Empfänger nicht angeschlossen/funktionsunfähig ist.

Deliverables and Milestones

Bei einer erfolgreichen Beendigung der Aufgabe kann die Zeit jede Sekunde genau einmal hochgezählt werden, und ist danach stets valide.

Resources Needed

Es werden keine weiteren Ressourcen benötigt.

Dependencies and Constraints

Für diese Methode müssen mehrere Bedingungen erfüllt sein. Erstens muss die Validierung der internen Zeit vorhanden sein, damit nach jedem Hochzählen weiterhin eine valide Zeit angezeigt wird. Zweitens muss der Sekundentimer im richtigen Intervall einen Interrupt auslösen, damit die Abweichung der internen Zeit zu der DCF-Zeit stets möglichst gering ausfällt. Drittens, sofern die angezeigte Zeit korrekt sein soll, muss zuvor eine DCF-Folge richtig eingelesen, ausgewertet und validiert werden.

Risks and Contingencies

Falls die Validierung nicht korrekt ist, kann es vorkommen, dass die Zeit nach dem Hochzählen nicht mehr valide ist. Ansonsten bietet der Sekundentimer eine Fehlerquelle, falls dieser nicht jede Sekunde auslöst.

3.10 Task09 - Darstellung auf Display

Description

Diese Aufgabe behandelt die Darstellung der Zeit und weiteren Informationen auf dem Display. Die angezeigte Zeit soll dabei die interne Zeit des ATMegas sein, die jede Sekunde hochgezählt wird. Darum muss auch das Display jede Sekunde aktualisiert werden.

Deliverables and Milestones

Nach Beendigung der Aufgabe sollen die Methoden der Displaydarstellung komplett zur Verfügung stehen, sodass dieses sofort angeschlossen werden kann und die angegebenen Informationen korrekt dargestellt werden.

Resources Needed

Für die Displaysteuerung wird die spezifische Dokumentation des jeweiligen Displays benötigt. Ohne die Informationen über die Steuerung des Displays zu kennen, können keine Funktionen dazu implementiert werden.

Dependencies and Constraints

Die Steuerung des Displays und das, was angezeigt werden soll, ist von allen anderen Funktionen abhängig. Wenn die richtige Zeit angezeigt werden soll, muss zuvor eine DCF-Folge erfolgreich eingelesen, verarbeitet und validiert werden, Ansonsten kann die angezeigte Zeit nicht korrekt sein. Auch vom Sekundentimer ist das Display abhängig,

der jede Sekunde den vollständigen Visualisierungsprozess anstößt. Falls dabei Fehler auftreten, liefert die Anzeige ebenfalls ein falsches Ergebnis.

Risks and Contingencies

Diese Aufgabe birgt viele verschiedene Risiken. Zunächst können die Instruction-Methoden für die Steuerung des Displays Fehler enthalten. Dies kann dazu führen, dass es gar nicht erst eingeschaltet werden kann, oder sonst nicht das gewünschte Verhalten zeigt. Des Weiteren können die Arrays mit den Zeichen, die angezeigt werden sollen, inkorrekt sein. Dies sollte jedoch leicht zu beheben sein.

3.11 Task10 - Lötboard

Description

Aufgabe 10 behandelt die Herstellung des Lötboards. Dieses bietet die Schnittstelle zwischen dem ATmega und dem DCF-Empfänger sowie die Schnittstelle zwischen dem ATmega und dem Display. So wird sichergestellt, dass die Bauteile mit den richtigen Spannungen versorgt werden. Außerdem kann für die Bauteile eine zusätzliche Spannung aus einer externen Quelle angelegt werden.

Deliverables and Milestones

Diese Aufgabe liefert das fertige, vollkommen funktionstüchtige Lötboard, über das externe Geräte betrieben und erfolgreich mit dem ATmega verbunden werden können.

Resources Needed

Zunächst müssen alle Bauteile vorhanden sein. Dazu gehören:

- Lötboard
- Kabel
- Widerstände
- Display
- DCF-Empfänger
- Steckplätze

Dazu werden Kenntnisse über die elektronischen Bauteile und Schaltungen benötigt. Außerdem werden die Dokumentationen und Schaltpläne aller Geräte benötigt, damit sämtliche Pins korrekt miteinander verbunden werden können.

Dependencies and Constraints

Der Einsatz des Lötboards mit Erfolg setzt einige erfüllte Bedingungen voraus. Eine dieser Bedingungen geht aus der Entwicklung der Software hervor. So müssen alle Pins des ATMegas korrekt besteuert werden, damit die Signale über das Lötboard an die richtigen Stellen gelangen können. Genauso verhält es sich mit Signalen, die zum ATmega gelangen sollen. Diese müssen genau an den Pins eintreffen, die von der Software eingelesen werden. Die anderen Bedingungen betreffen die auf dem Lötboard eingesetzte, teilweise verlötete Hardware, oder den ATmega auf dem Evaluationsboard selbst. Dazu zählen auch alle Kabelverbindungen. Mechanisch müssen sich die Bauteile allesamt in einem voll funktionsfähigen Zustand befinden. Ist dies nicht der Fall und manche Signale und Spannungen werden durch kaputte Teile nicht ordnungsgemäß verteilt, muss eine mühsame Suche nach dem/den Fehler(n) erfolgen. Dabei müssen alle Spannungen mit einem Multimeter einzeln durchgemessen werden. Des Weiteren müssen die Bauteile an den genau richtigen Stellen eingelötet werden, die Leiterbahnen dort wo es nötig ist unterbrochen werden und nur die jeweils zusammengehörenden Pins über die Kabel miteinander verbunden sein.

Risks and Contingencies

Die Risiken bei dieser Aufgabe bestehen bei den unzähligen Fehlerquellen, die geboten werden. Diese sind sowohl auf Hardware- als auch auf der Softwareebene zu finden. Sobald mindestens einer dieser Fehler auftritt, wird eine aufwändige Fehlersuche von Nöten sein, die viel Zeit in Anspruch nehmen kann.

3.12 Task11 - Inbetriebnahme

Description

Dieser Task ist die entgeltliche Inbetriebnahme des Systems. Dabei kommen alle Hardwarebausteine und Softwarefunktionen aus dem Hardwareteil zum Einsatz.

Deliverables and Milestones

Es handelt sich hierbei um die finale Inbetriebnahme, sodass das fertige System geliefert wird. Damit dieses als funktionsfähig eingestuft werden kann, muss die Inbetriebnahme zwingend Reibungslos ablaufen.

Resources Needed

Es werden alle Hardware- und Softwarekomponenten benötigt.

Dependencies and Constraints

Diese Aufgabe ist davon abhängig, dass alle Elemente des Systems ausgiebig getestet wurden. Dazu zählen die komplette Hard- und Software. Daneben dürfen keine, bisher noch nicht festgestellten Fehler auftreten. Außerdem müssen sich alle elektrischen Bauteile in einem funktionfähigen Zustand befinden. Für das Display muss in unserem Fall zudem eine zusätzliche Spannungsquelle zur Verfügung stehen. Zu guter Letzt muss mindestens einmal erfolgreich ein valides DCF-Signal empfangen und ausgewertet werden. Sollte dies aus irgend einem Grund nicht stattfinden, ist die vollumfängliche Inbetriebnahme nicht erfolgreich verlaufen.

Risks and Contingencies

Bei dieser Aufgabe haben Fehler die negativste Auswirkung, da es sich um die finale Inbetriebnahme handelt. Bei jedem Auftreten eines Fehlers ist diese nicht erfolgreich. Außerdem sind alle möglichen Fehlerquellen vorhanden, da das gesamte System im Einsatz ist.

3.13 Timetable

Beginn	Aufgabe	dazugehörige Tasks
05.10.18	Anforderungserhebung	01, 02
12.10.18	Testfälle erstellen	03
09.11.18	Validierung Datum und Zeit	07
14.11.18	Implementierung Timer	04
14.11.18	Datenstruktur für DCF-Folge	05
30.11.18	Datenstruktur für interne Zeit	05
30.11.18	Hochzählen interne Zeit	08
14.12.18	Einlesen DCF-Folge	06
19.12.18	Herstellung Lötboard	10
09.01.19	Darstellung auf Display	09
23.01.19	Paritätschecker	07
24.01.19	Inbetriebnahme System	11

Anmerkungen:

Alle Aufgaben wurden bis zum 25.01.19 ausgeführt, da stets Änderungen nötig waren und die vorhandenen Funktionen nach der Implementierung von neuen Funktionen häufig angepasst werden mussten.

Bis auf die Herstellung des Lötboards wurden die meisten Aufgaben gemeinsam durchgeführt. Dies war bei der Diskussion über die anzuwendenden Verfahren immer der Fall, die Implementierungen wurden jedoch teilweise in Einzelarbeit durchgeführt.