

Universität der Bundeswehr München
Fakultät für Elektrotechnik und Technische Informatik
Wissenschaftliche Einrichtung 4 - Daten- und Schaltungstechnik

Prof. Dr.-Ing. Ferdinand ENGLBERGER
Prof. Dr.-Ing. Thomas LATZEL

Dokumentation zur Schrittmotorsteuerung

Meilenstein 2 - NIOS II-Basissoftware mit Stepperkomponente



Autoren: Lt Marc KOSSMANN
Lt Michael RIEDEL
Gruppe: SoC04
Abgabedatum: 18.11.2014

Inhaltsverzeichnis

	Seite
1. Planung des Projekts	1
2. Beschreibung der Anwendungsfälle	4
3. Beschreibung und Darstellung der einzelnen Aktivitäten	5
3.1. benötigte Tasks	5
3.1.1. Änderungen an der Heartbeat und Debug-Task	5
3.1.2. Änderungen an der User-Input-Task	5
3.1.3. Änderungen an der User-Output-Task	5
3.2. benötigte Interrupt-Service-Routinen	5
3.2.1. Änderungen an der Key-ISR	5
3.2.2. Änderungen an der Switch-ISR	6
3.2.3. Änderungen an der Motor-ISR	6
3.3. benötigte Funktionen	6
3.3.1. Änderungen an der Main-Funktion	6
3.3.2. Änderungen an der Init-Funktion	6
3.3.3. Änderungen an der Heartbeat-Funktion	6
3.3.4. Änderungen an der Chain of Steps-Funktion	6
3.3.5. Änderungen an der Continuous Run-Funktion	6
4. Darstellung der verschiedenen Aktivitätsdiagramme	7
5. Weitere Darstellungen zur Erläuterung der internen Kommunikation	12

1. Planung des Projekts

In das Gantt-Diagramm in Abbildung 1.1 wurden die gewonnenen Erkenntnisse in Meilenstein 2 eingepflegt.

Abbildung 1.2 zeigt den geplanten und benötigten Zeitaufwand für die Erstellung des Meilenstein 2 unterteilt in folgende Aufgabenbereiche:

- Einarbeitung
- Zeitplanung
- Design
- Implementierung
- Verifikation
- Dokumentation

Die Darstellung wird gesondert für die Studenten Marc Kossmann und Michael Riedel betrachtet. Diese Zeiten sind unabhängig von gemeinsam bearbeiteten Aufgaben. Die Abbildung 1.3 zeigt die komplette geplante und benötigte Zeit, die durch Aufsummierung der einzelnen Meilensteine entsteht.

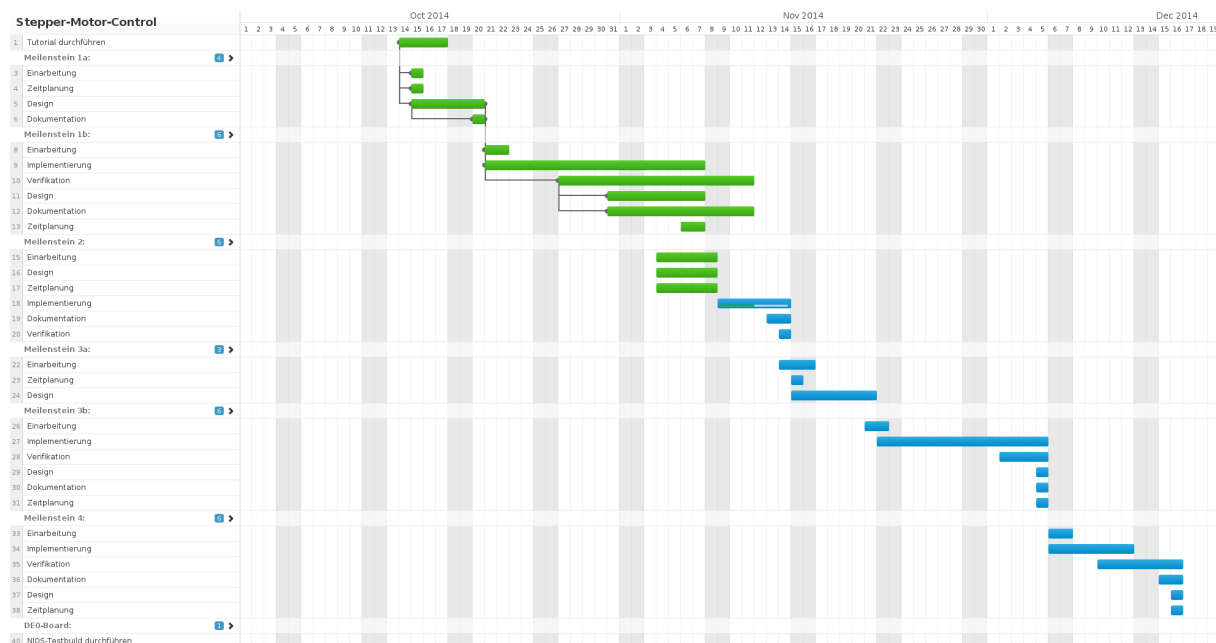
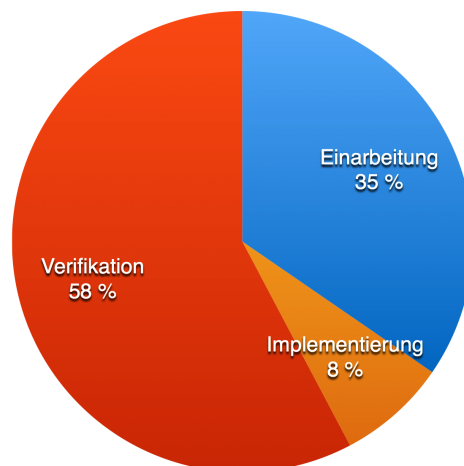


Abbildung 1.1: Gantt-Diagramm zur kompletten Zeitplanung

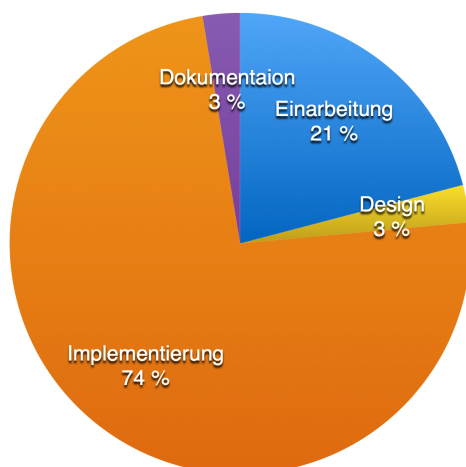
Meilenstein 1b

	Marc		Michael		gemeinsam		gesamt	
	geplant	benötigt	geplant	benötigt	geplant	benötigt	geplant	benötigt
Einarbeitung	1	8	1	8,25	2	5,99	4	22,24
Zeitplanung	0,5	0	0,5	0,38	1	0	2	0,38
Design	1	1	2	2,44	3	0	6	3,44
Implementierung	16	28,25	12	15,45	4	1,33	32	45,03
Verifikation	2	0	1	0	6	10	9	10
Dokumentaion	2	1	2	5,02	4	0	8	6,02
Summe	22,5	38,25	18,5	31,54	20	17,32	61	87,11

gemeinsam (Verteilung der benötigten Zeit)



Marc (Verteilung der benötigten Zeit)



Michael (Verteilung der benötigten Zeit)

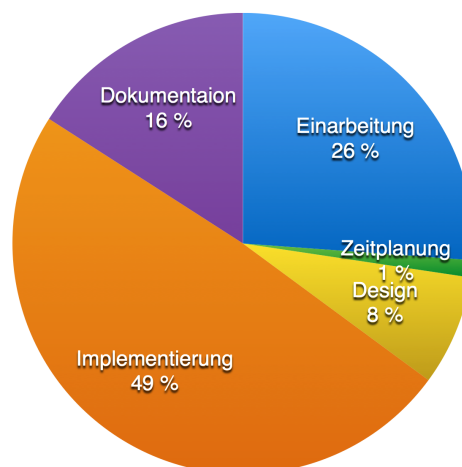
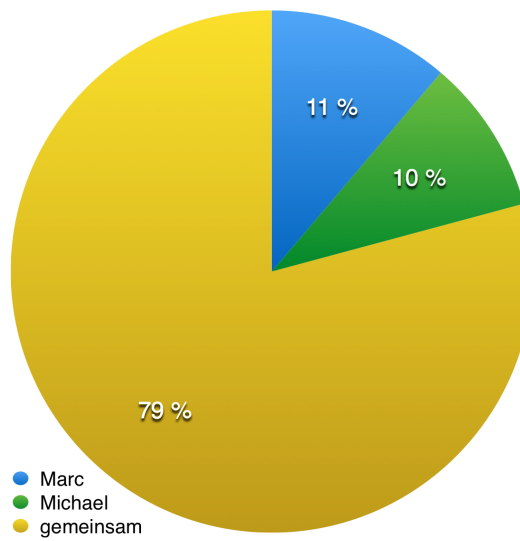


Abbildung 1.2: Projektplanung für Meilenstein 2

Zeitbedarfsübersicht

	Marc	Michael	gemeinsam	gesamt
geplant	27,5	23,5	194	245
benötigt	51,25	50,63	29,27	131,15

Aufwandsverteilung (geplant)



Aufwandsverteilung (benötigt)

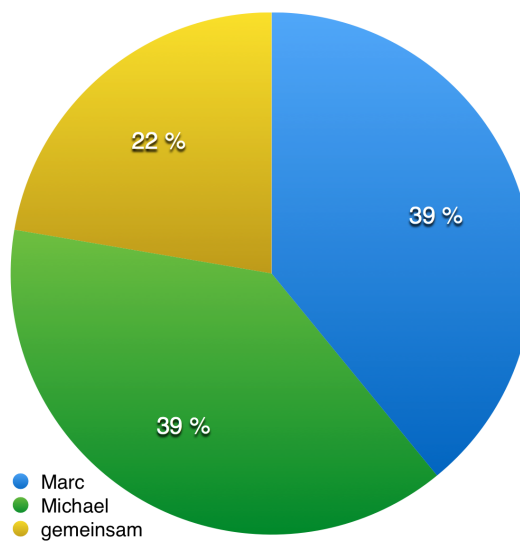


Abbildung 1.3: Zeitbedarfsübersicht für das gesamte Projekt

2. Beschreibung der Anwendungsfälle

An den Anwendungsfällen wurden leichte strukturelle Änderungen aufgrund der Implementierung vorgenommen. Die zugrundeliegende Logik ist gleich geblieben.¹ Zur Nachvollziehbarkeit ist das Anwendungsfalldiagramm in Abbildung 2.1 dargestellt.

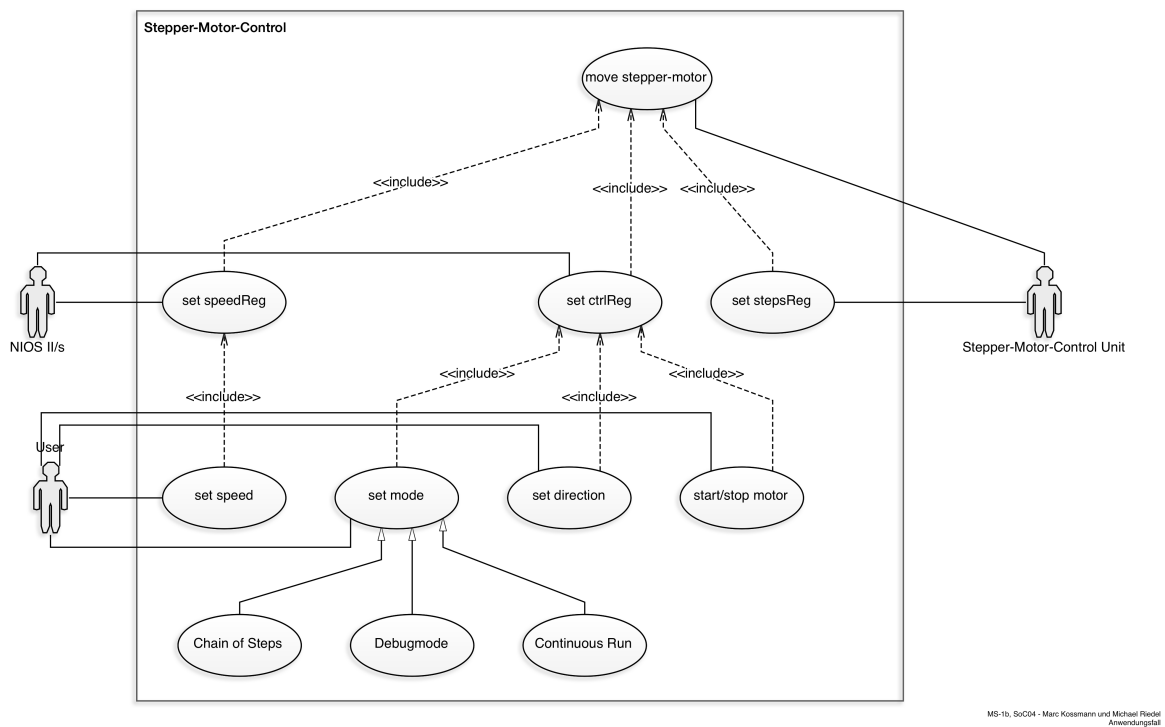


Abbildung 2.1: Anwendungsfälle

¹ Für weitere Details zur allgemeinen Implementierung wird auf die Dokumentation zu Meilenstein 1a verwiesen, da im Folgenden nur auf die Änderungen eingegangen wird.

3. Beschreibung und Darstellung der einzelnen Aktivitäten

Der Entwurf des Steuerprogramms wurde weitestgehend umgesetzt. Natürlich wurden kleinere Anpassungen während der Umsetzung vorgenommen. Genauere Informationen zur Funktionalität sind in der Dokumentation zum Meilenstein 1a zu finden.

3.1. benötigte Tasks

Die benötigten Tasks wurden im Ablauf nicht geändert. In den folgenden Abschnitten werden die Änderungen einzeln benannt und erläutert.

3.1.1. Änderungen an der Heartbeat und Debug-Task

Es wurden gemäß Abbildung 4.1 keine Änderungen im Diagramm vorgenommen.

3.1.2. Änderungen an der User-Input-Task

Die User-Input-Task stellt gemäß Abbildung 4.2 die Haupttask der Steuersoftware dar. Die initialen Ausgaben des LC-Displays und des Terminals wurden aus der Main-Funktion in die User-Input-Task verlegt. Ebenfalls wurden die Event-Bezeichnungen angepasst, um eine Kontinuität im Gesamtprojekt zu ermöglichen.

3.1.3. Änderungen an der User-Output-Task

In der User-Output-Task wurden gemäß Abbildung 4.3 die Anmerkungen zur gewünschten Ausgabe angepasst, um Deckungsgleichheit mit der Aufgabenstellung herzustellen.

3.2. benötigte Interrupt-Service-Routinen

3.2.1. Änderungen an der Key-ISR

In der Key-ISR wurden gemäß Abbildung 4.4 die Event-Bezeichnungen angepasst, um eine Kontinuität im Gesamtprojekt zu ermöglichen.

3.2.2. Änderungen an der Switch-ISR

Die Abfrage der Schalter in der `Switch-ISR` wurde gemäß Abbildung 4.5 vereinfacht. Das `Switch-Update-Flag` wurde eingespart. Stattdessen werden die aktuellen Schalterstellungen ausgelesen und per Mailbox an die User-Output-Task weitergeleitet.

3.2.3. Änderungen an der Motor-ISR

In der Motor-ISR wurde gemäß Abbildung 4.6 die Abfrage, ob das `Interrupt-Enable-Bit` gesetzt ist, hinzugefügt, bevor das entsprechende Event gesendet wird. Des Weiteren wurde die Event-Bezeichnung angepasst.

3.3. benötigte Funktionen

Grundsätzlich ist der Ablauf der Aktivitäten gleich geblieben. Es wurden einige Aktivitäten in Tasks verschoben, um einen homogenen Ablauf zu ermöglichen. Auf eine eigene Init-Funktion wurde verzichtet und die Aktivitäten in die Main-Funktion sowie die User-Input-Task integriert.

3.3.1. Änderungen an der Main-Funktion

Die Main-Funktion führt gemäß Abbildung 4.7 nun die nötigen Initialisierungen des Betriebssystems, der Tasks und ISRs durch. Der letzte Aufruf der Main-Funktion ist das Starten des Betriebssystems.

3.3.2. Änderungen an der Init-Funktion

Auf die Implementierung einer eigenen Init-Funktion wurde verzichtet. Das Diagramm wird aus diesem Grund für die folgende Entwicklung nicht weiter betrachtet.

3.3.3. Änderungen an der Heartbeat-Funktion

Es wurden gemäß Abbildung 4.8 keine Änderungen im Diagramm vorgenommen.

3.3.4. Änderungen an der Chain of Steps-Funktion

Es wurden gemäß Abbildung 4.9 keine Änderungen im Diagramm vorgenommen.

3.3.5. Änderungen an der Continuous Run-Funktion

Es wurden gemäß Abbildung 4.10 keine Änderungen im Diagramm vorgenommen.

4. Darstellung der verschiedenen Aktivitätsdiagramme

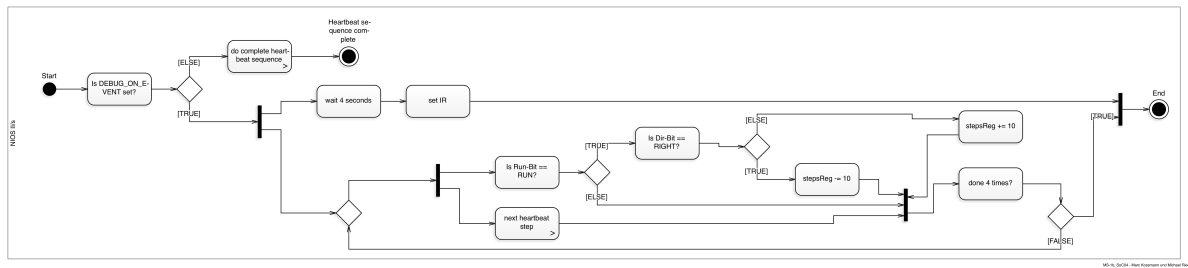


Abbildung 4.1: Heartbeat-Debug Task

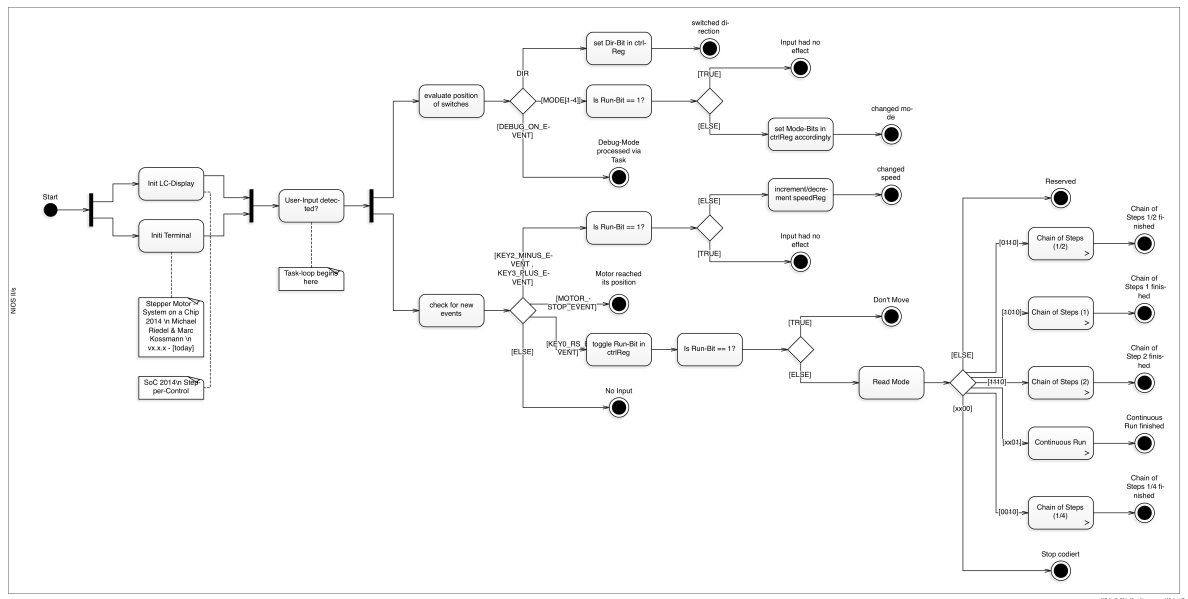
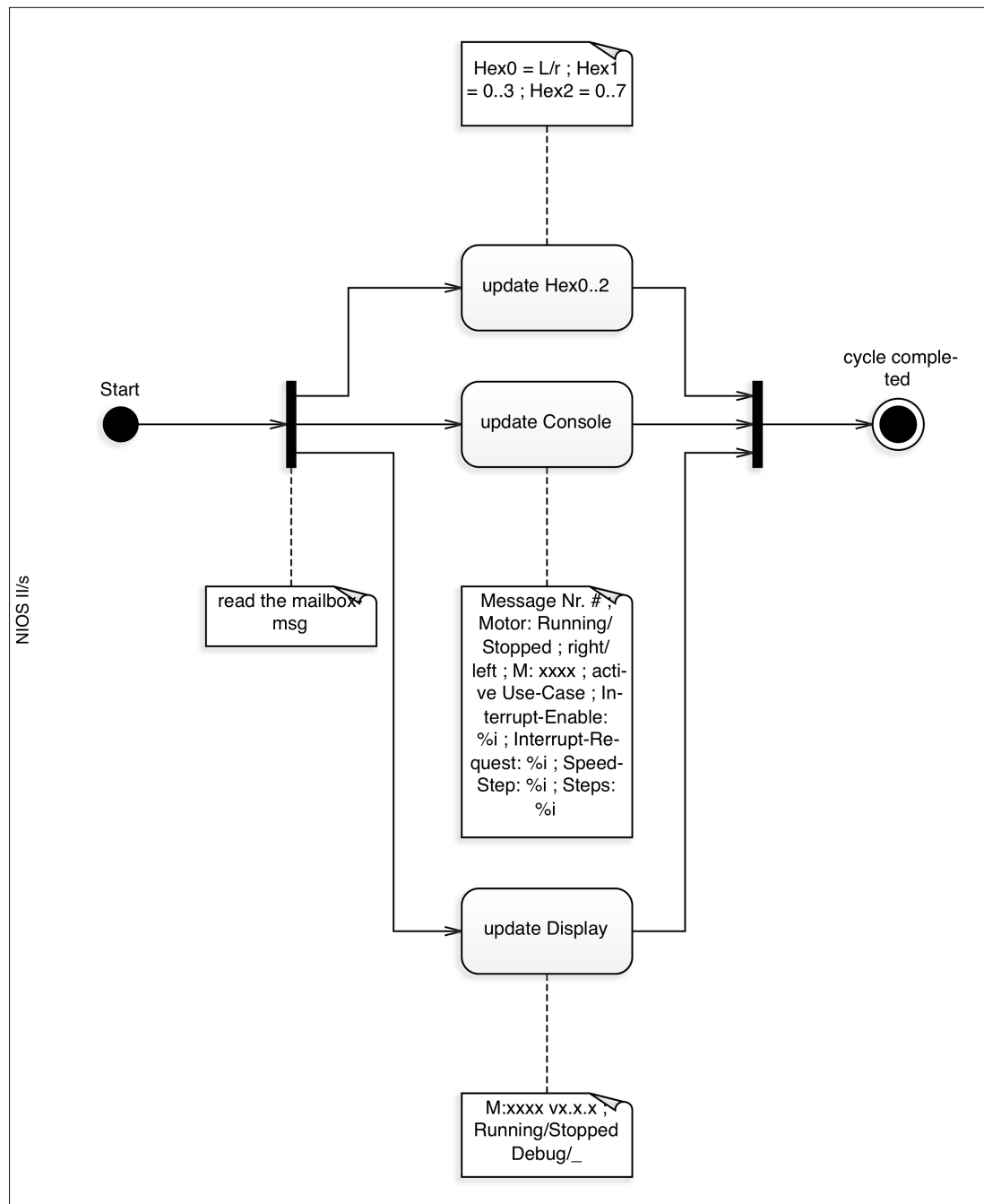
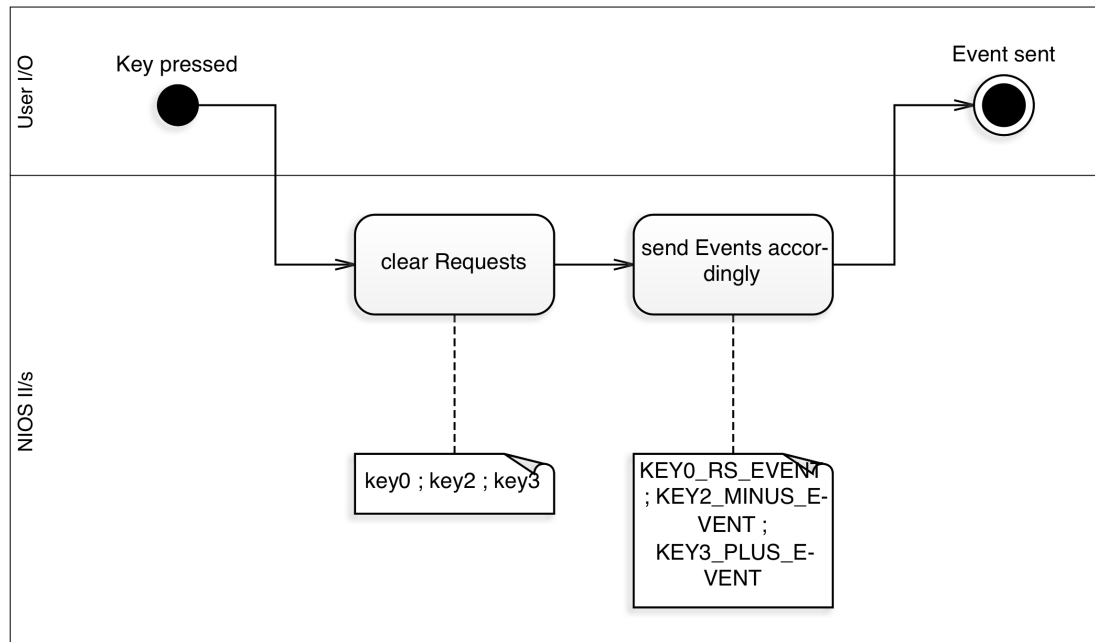


Abbildung 4.2: User-Input Task



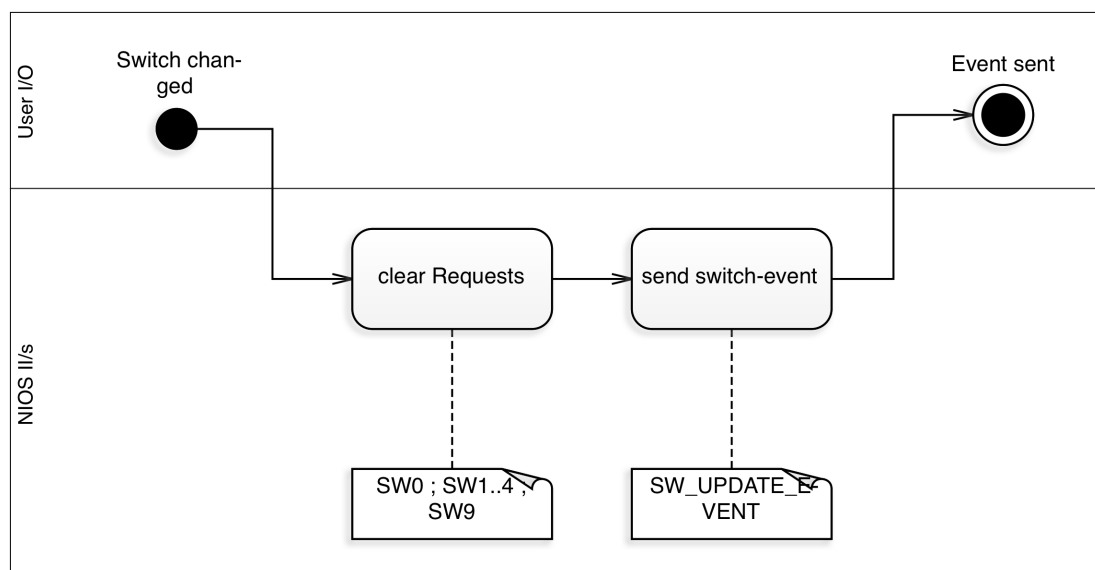
MS-1b, SoC04 - Marc Kossmann und Michael Riedel
User-Output Task

Abbildung 4.3: User-Output Task



MS-1b, SoC04 - Marc Kossmann und Michael Riedel
key ISR

Abbildung 4.4: Key ISR



MS-1b, SoC04 - Marc Kossmann und Michael Riedel
switch ISR

Abbildung 4.5: Switch ISR

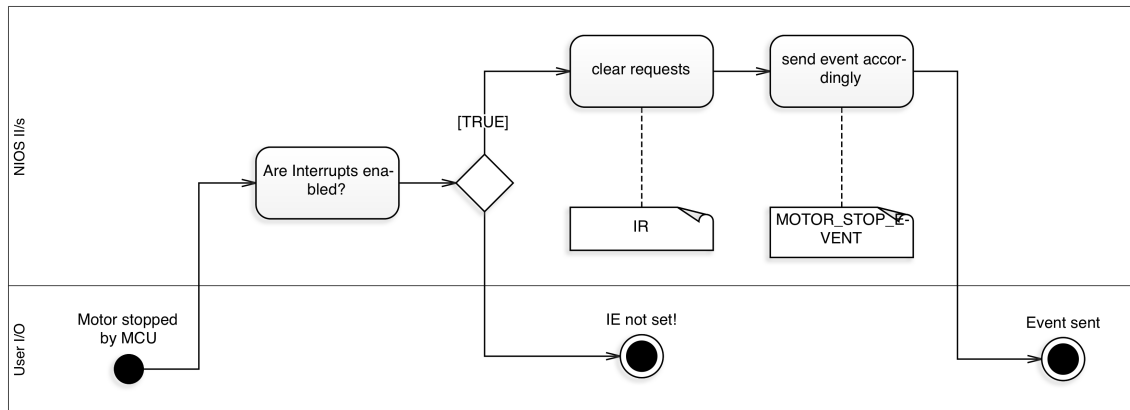
MS-1b, SoC04 - Marc Kossmann und Michael Riedel
motor ISR

Abbildung 4.6: Motor ISR

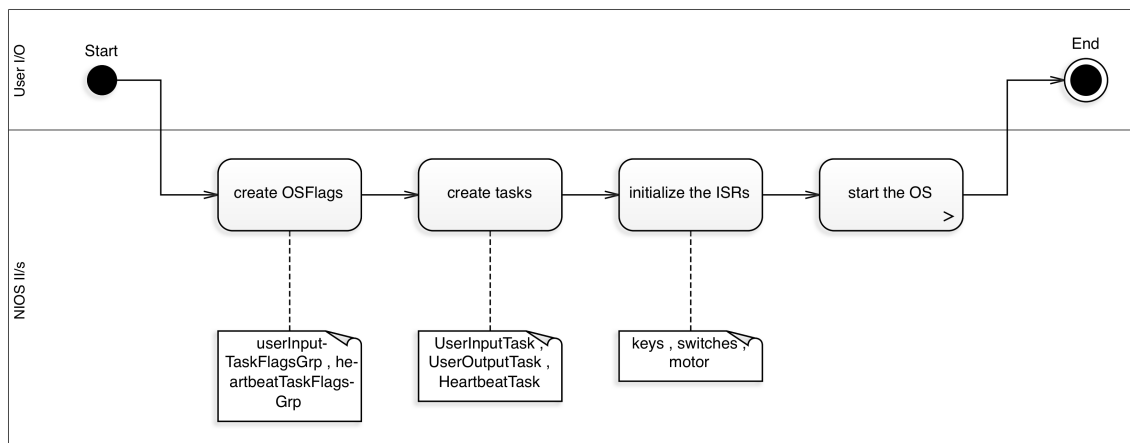
MS-1b, SoC04 - Marc Kossmann und Michael Riedel
Main

Abbildung 4.7: Main Funktion

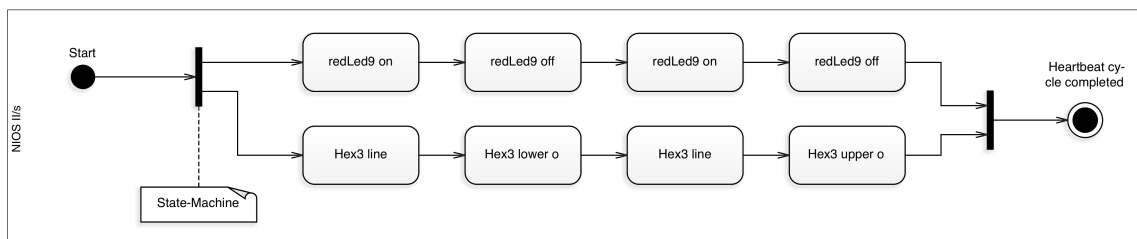
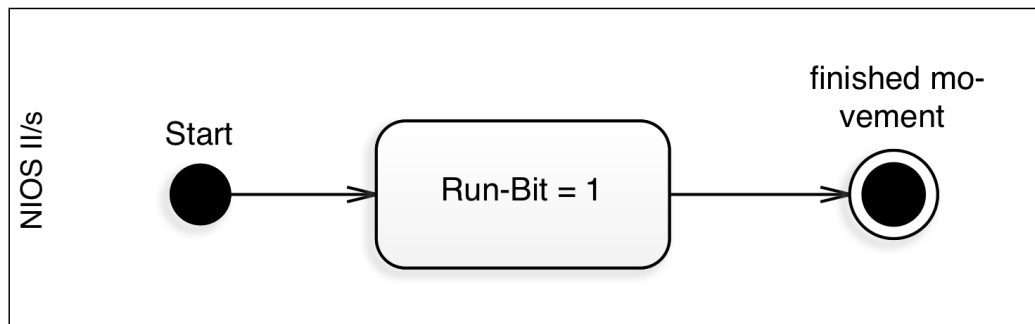
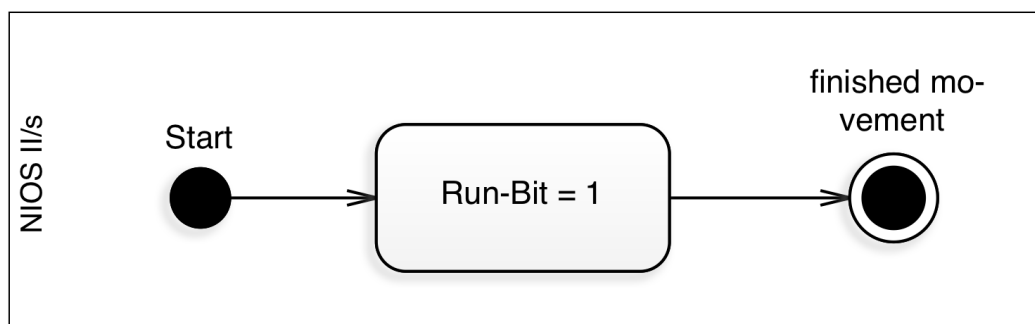
MS-1b, SoC04 - Marc Kossmann und Michael Riedel
HeartBeat

Abbildung 4.8: Heartbeat Funktion



MS-1b, SoC04 - Marc Kossmann und Michael Riedel
Chain of Steps

Abbildung 4.9: Chain of Steps Funktion



MS-1b, SoC04 - Marc Kossmann und Michael Riedel
Continuous Run

Abbildung 4.10: Continuous Run Funktion

5. Weitere Darstellungen zur Erläuterung der internen Kommunikation

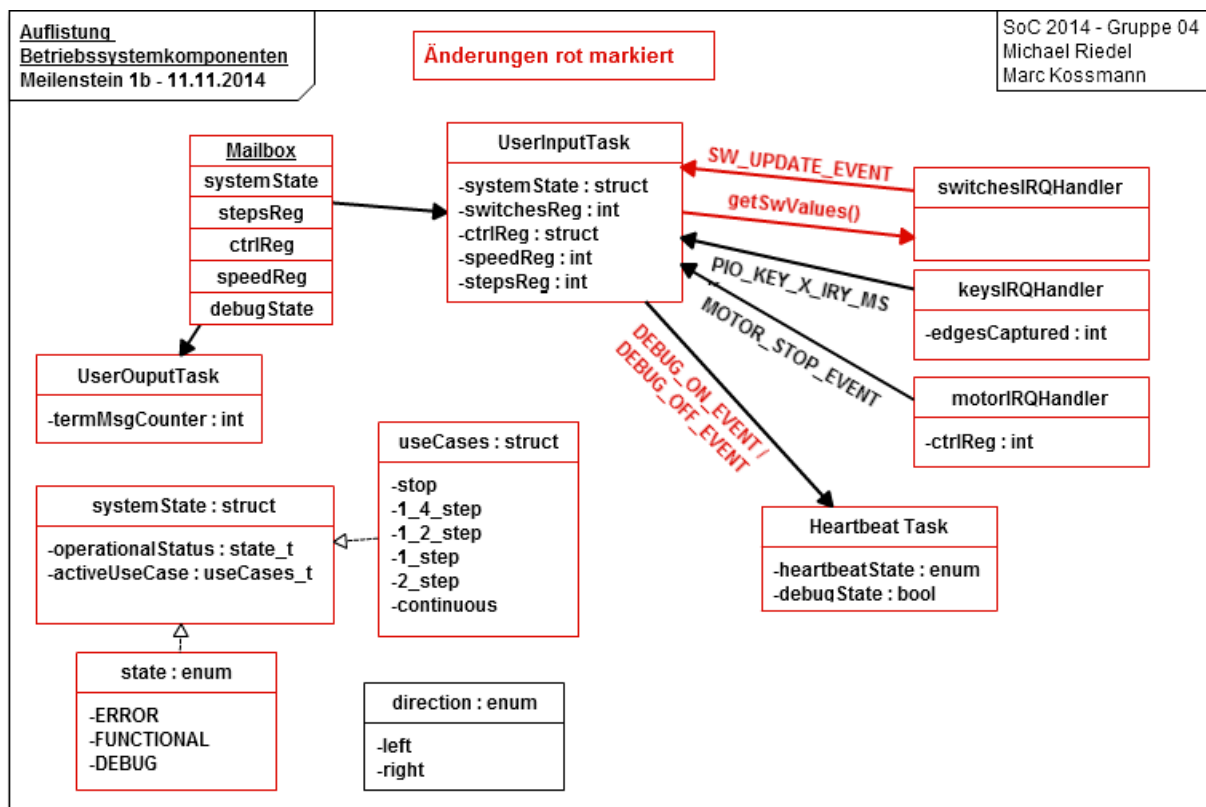


Abbildung 5.1: Auflistung Betriebssystemkomponenten

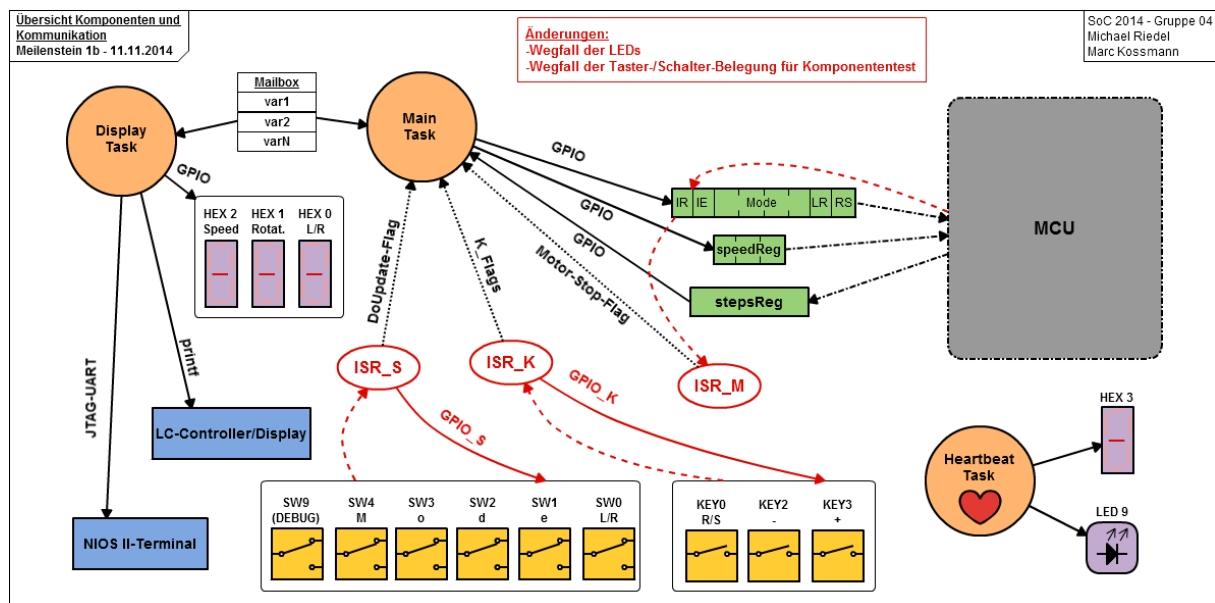


Abbildung 5.2: Übersicht der Komponenten und Kommunikation