

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 3a - Design der Motor-Control-Unit



Autoren: Lt Marc KOSSMANN  
Lt Michael RIEDEL  
Gruppe: SoC04  
Abgabedatum: 24.11.2014

# Inhaltsverzeichnis

	Seite
<b>1. Planung des Projekts</b>	<b>1</b>
<b>2. Anpassungen an der Komponente register_interface</b>	<b>5</b>
<b>3. Design der Motor-Control-Unit</b>	<b>6</b>
3.1. Überblick . . . . .	6
3.2. Design der Komponente counter . . . . .	8
3.3. Design der Komponente signal_generator . . . . .	8
3.4. Integration in eine Testumgebung durch die Komponente debug_key_detect	9

# 1. Planung des Projekts

Das Gantt-Diagramm in Abbildung 1.1 wurde auf eingetretene Verzögerungen angepasst, sodass die Deadline Mitte Dezember eingehalten werden kann.

Abbildung 1.2 zeigt den geplanten und benötigten Zeitaufwand für die Erstellung des Meilenstein 3a 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.

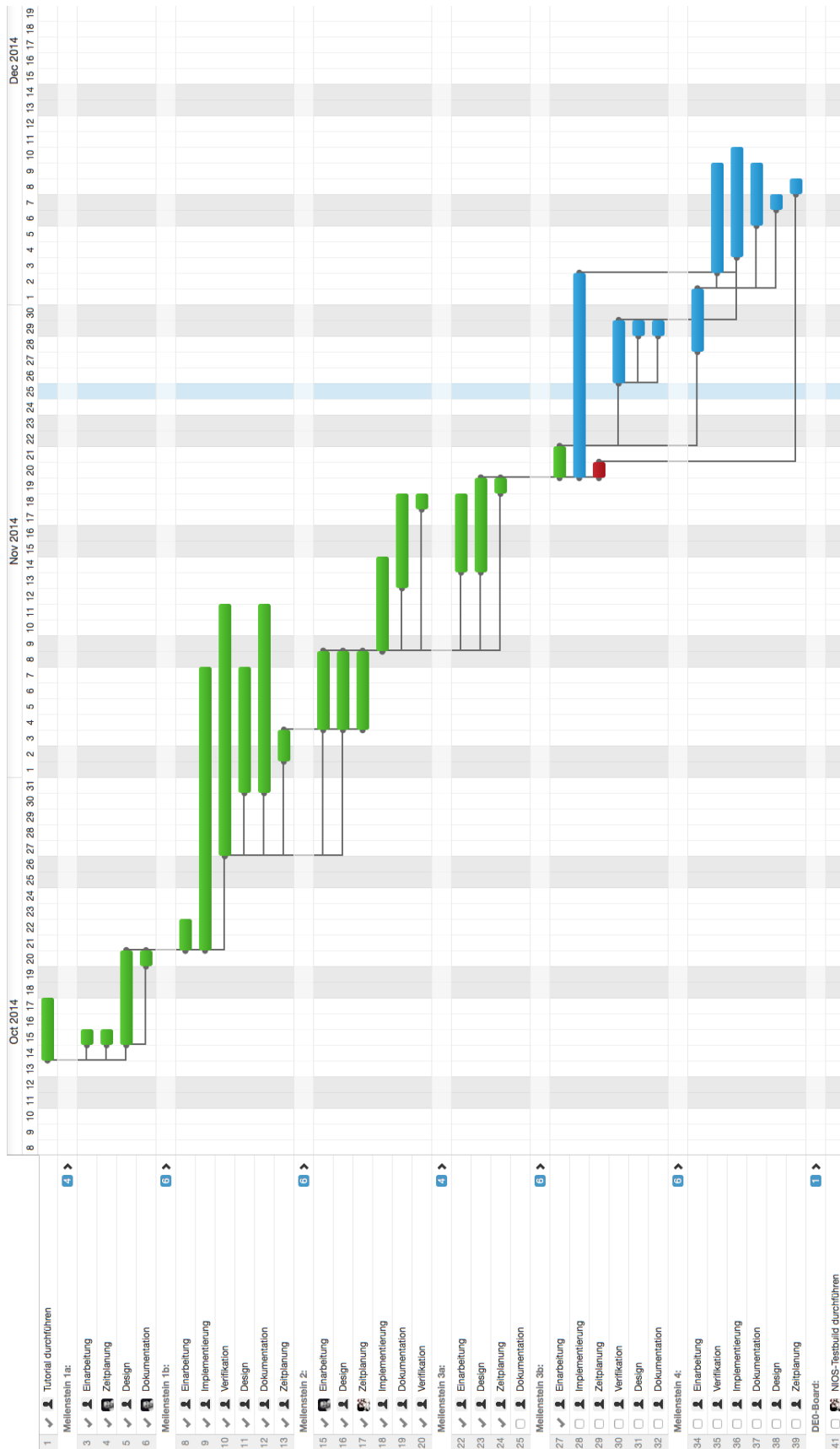
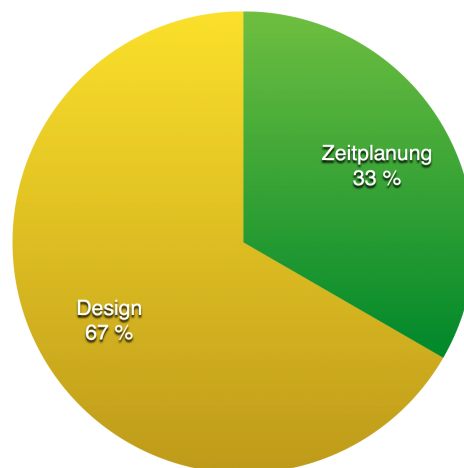


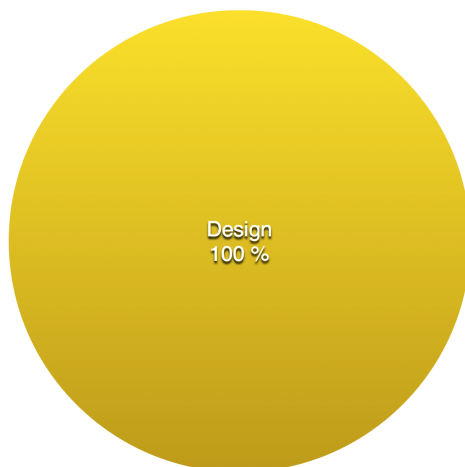
Abbildung 1.1: Gantt-Diagramm zur kompletten Zeitplanung

Meilenstein 3a								
	Marc		Michael		gemeinsam		gesamt	
	geplant	benötigt	geplant	benötigt	geplant	benötigt	geplant	benötigt
Einarbeitung	1	0	1	0	0	0	2	0
Zeitplanung	0	0	0	0,32	1	0,5	1	0,82
Design	4	3	5	7,57	1	1	10	11,57
Implementierung	0	0	0	0	0	0	0	0
Verifikation	0	0	0	0	0	0	0	0
Dokumentaion	0	0	4	2,68	0	0	4	2,68
Summe	5	3	10	10,57	2	1,5	17	15,07

gemeinsam (Verteilung der benötigten Zeit)



Marc (Verteilung der benötigten Zeit)



Michael (Verteilung der benötigten Zeit)

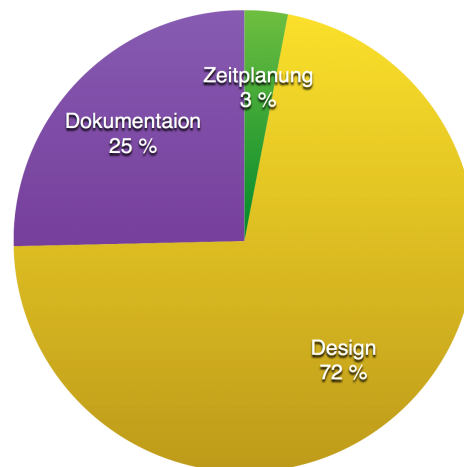
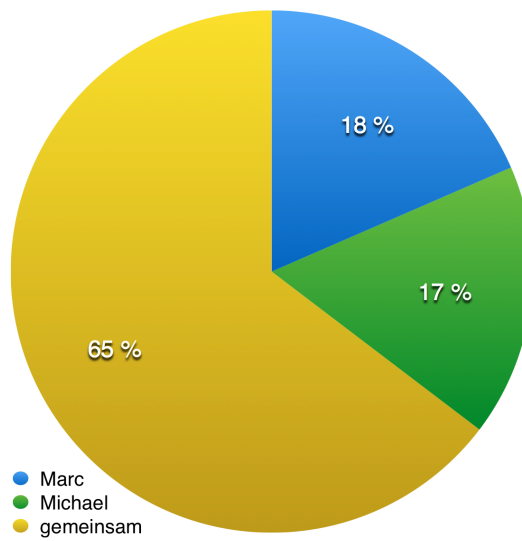


Abbildung 1.2: Projektplanung für Meilenstein 3a

Zeitbedarfsübersicht

	Marc	Michael	gemeinsam	gesamt
geplant	46	42	161	249
benötigt	73,25	70,31	39,02	182,58

Aufwandsverteilung (geplant)



Aufwandsverteilung (benötigt)

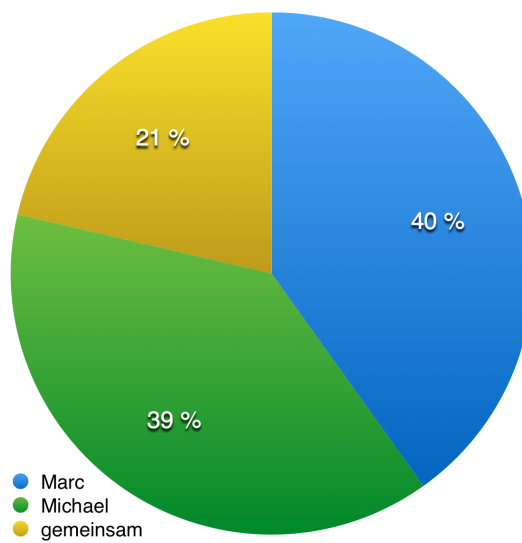


Abbildung 1.3: Zeitbedarfsübersicht für das gesamte Projekt

## 2. Anpassungen an der Komponente register\_interface

Die Entity dieser Komponente wurde gemäß Abbildung 2.1 für die Verwendung mit der Motor-Control-Unit angepasst und bietet nun die zusätzlichen Dateneingänge der **steps** (Anzahl der verfahrenen Schritte) sowie einem **ir**-Eingang, der das stoppen des Motors markiert. Ebenfalls wurden die Datenausgänge **mode** (der vom Benutzer eingestellte Betriebsmodus), die **direction** (die vom Benutzer gewählte Drehrichtung), **speed** (die vom Benutzer gewählte Verfahrensgeschwindigkeit) sowie **run** (die Eingabe durch den Benutzer, den Motor zu starten oder zu stoppen). Die Architektur wurde gemäß Kundenwunsch angepasst und korrigiert.

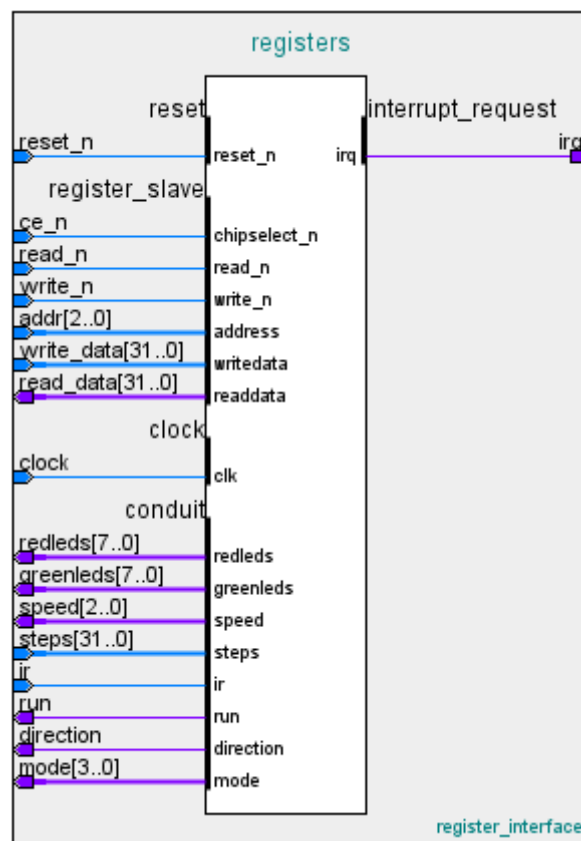


Abbildung 2.1: Block-Diagramm der register\_interface-Komponente

### 3. Design der Motor-Control-Unit

Die Motor-Control-Unit ist die Schnittstelle zwischen dem NIOS-II Prozessor und der Schrittmotor-Einheit. Sie erstellt entsprechend der vom Benutzer eingestellten Betriebsmodi die Pulsweiten-modulierten (PWM)-Signale für den Schrittmotor, aktiviert, stoppt und überwacht das gewünschte Verfahren.

Kombination	Modus
xx00	Stop
xx01	Continuous Run
0010	Chain of Steps - 1/4 rotation
0110	Chain of Steps - 1/2 rotation
1010	Chain of Steps - 1 rotation
1110	Chain of Steps - 2 rotations
other	reserved

Tabelle 3.1: Betriebsmodi der Motor-Control-Unit

Gemäß Kundenwunsch werden die Betriebsmodi gemäß Tabelle 3.1 durch die Komponente zur Verfügung gestellt.

#### 3.1. Überblick

Gemäß Abbildung 3.1 besteht die Komponente `motor_control_unit` aus den zwei Teilkomponenten `counter` und `signal_generator`. Der Signalgenerator wird dabei durch das `clk_out`-Signal des Taktteilers gespeist.

Die `motor_control_unit` wird gemäß Abbildung 3.2 in Form einer State-Machine realisiert. In ihr sind die Übergänge zwischen den verschiedenen Betriebsmodi, sowie der Anzahl der zu zählenden Schritte für die entsprechenden Rotationen dargestellt. Der Übergang wird nur dann gewechselt, sofern ein `clk`-Signal, ein gültiger Mode sowie der Benutzer den Motor starten möchte.



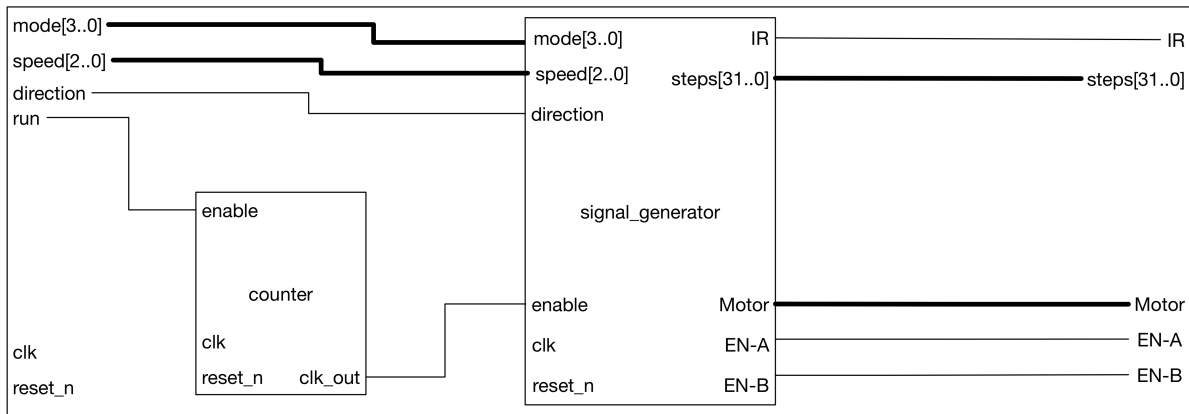


Abbildung 3.1: Block-Diagramm der Motor-Control-Unit-Komponente

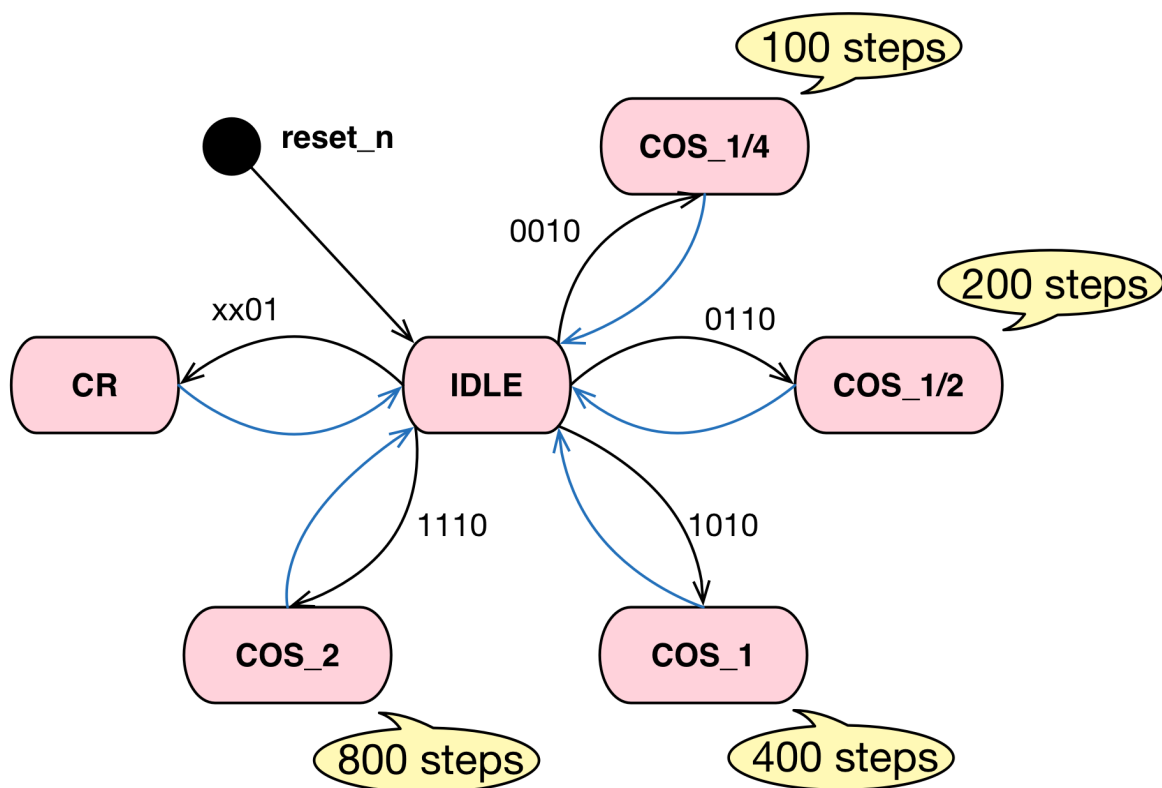


Abbildung 3.2: State-Machine zu den Mode-Übergängen der Motor-Control-Unit-Komponente

### 3.2. Design der Komponente counter

Der Counter wird allgemeingültig realisiert. Dadurch kann der Teilerwert individuell durch ein `generic` konfiguriert und mit unterschiedlichen Frequenzen betrieben werden. Dies ermöglicht eine einfache Anpassung von Meilenstein 3b zu Meilenstein 4.

Der einzustellende Taktteiler berechnet sich wie folgt:

$$\text{Taktteiler} = \frac{\text{clk}_{\text{in}}}{\text{clk}_{\text{out}}} \quad (3.1)$$

$\text{clk}_{\text{in}} = \text{Eingangsfrequenz}$   
 $\text{clk}_{\text{out}} = \text{Ausgabefrequenz}$

### 3.3. Design der Komponente signal\_generator

Der Signalgenerator wird gemäß Abbildung 3.3 in Form einer State-Machine realisiert, um die Windungen des Motors in korrekter Reihenfolge anzusteuern.

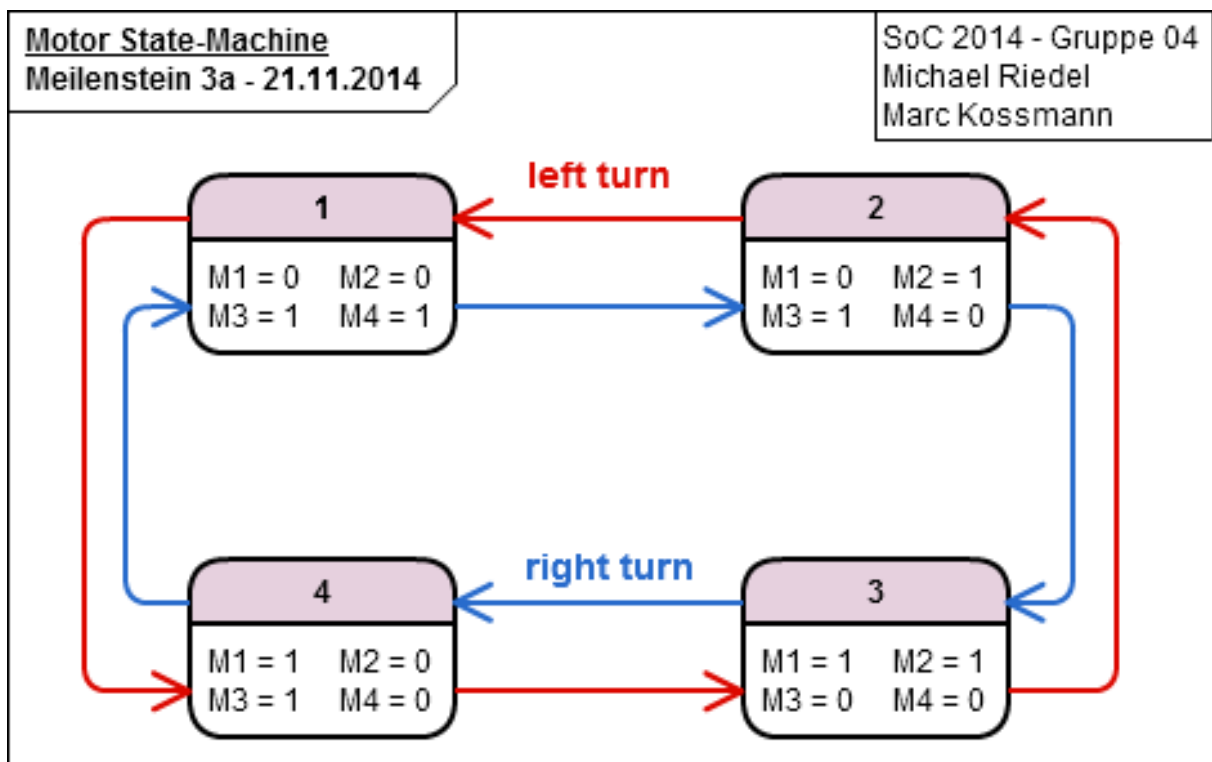


Abbildung 3.3: State-Machine zu Beschaltung der Wicklungen des Schrittmotors

### 3.4. Integration in eine Testumgebung durch die Komponente *debug\_key\_detect*

Damit die *motor\_control\_unit* unabhängig vom NIOS-II Prozessor getestet werden kann, wird im Meilenstein 3b eine eigene Komponente erstellt, die gemäß Abbildung 3.4 das *register\_interface* und die *motor\_control\_unit* mit einer *debug\_key\_detect*-Komponente verbindet.

Diese Komponente ermöglicht das gezielte Simulieren der Prozessoranfragen an die Motor-Control-Unit. Sie beschreibt das *register\_interface* mit den gewünschten Daten und initiiert das Verfahren des Schrittmotors.

Gemäß Abbildung 3.5 können die einzelnen Aufgaben, die die jeweiligen Komponenten übernehmen.

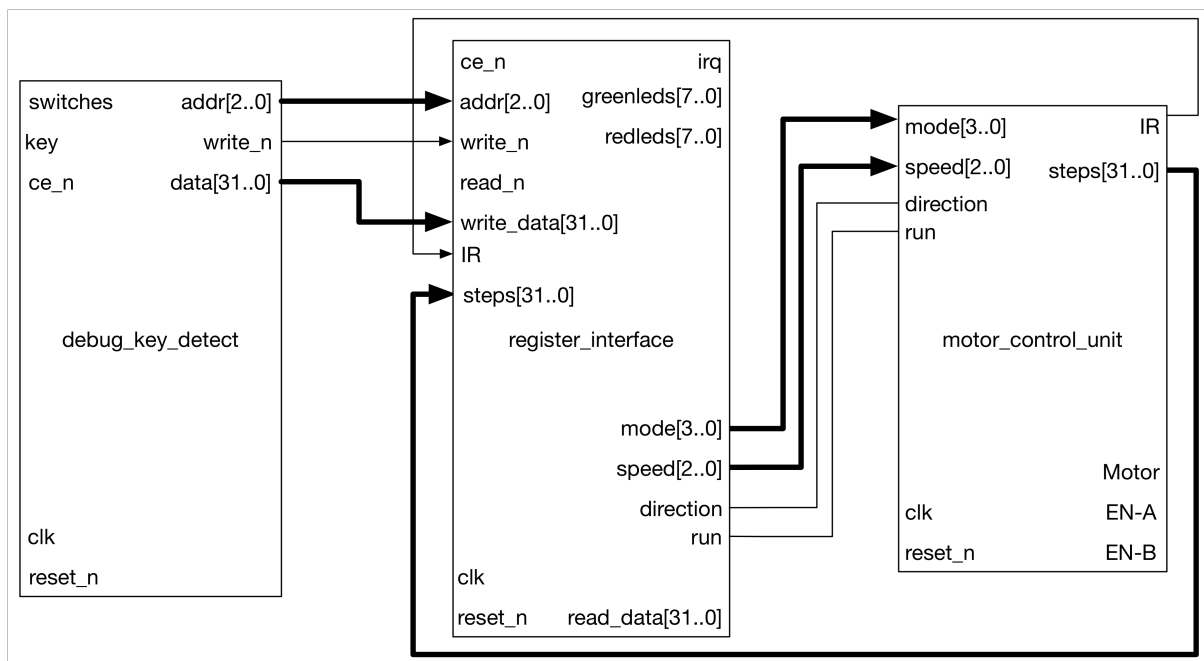


Abbildung 3.4: Block-Diagramm der Testumgebung

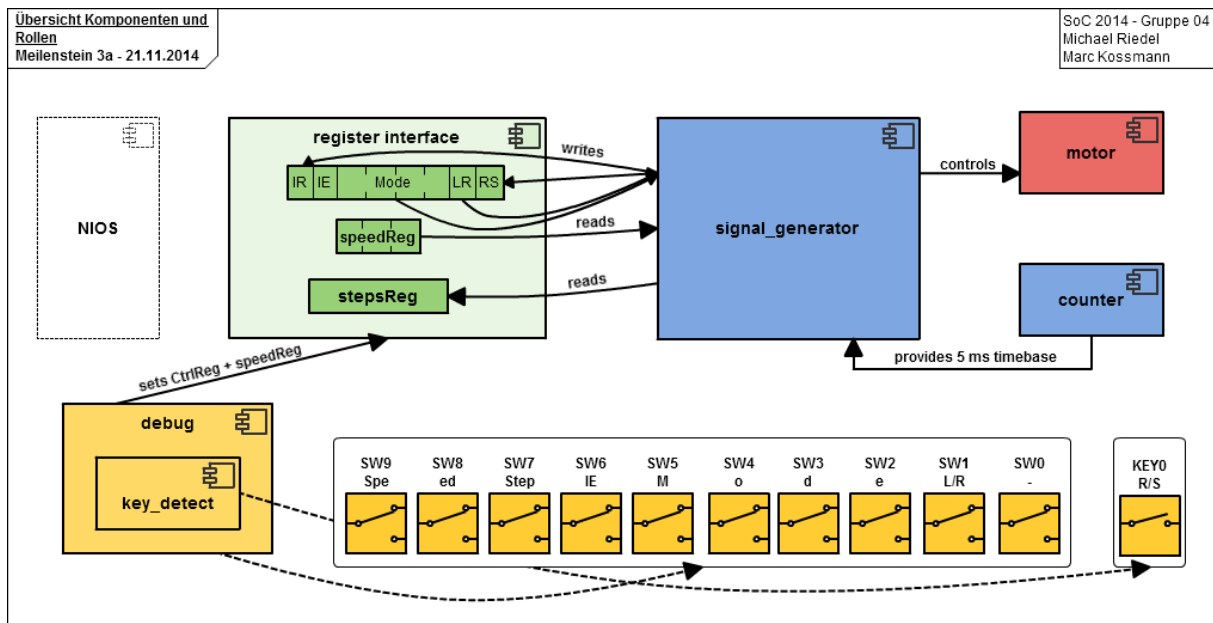


Abbildung 3.5: Überblick der Komponenten und Rollen