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 3b - Implementierung der Motor-Control-Unit

Autoren: Lt Marc Kossmann
Lt Michael Riedel
Gruppe: SoC04

Abgabedatum: 02.12.2014

Inhaltsverzeichnis

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 3b 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.

Die Gesamtzeiten berechnen sich aus der Aufsummierung der einzelnen Zeiten der Studenten Marc Kossmann und Michael Riedel sowie der gemeinsam zusätzlich aufgewendeten Zeit.

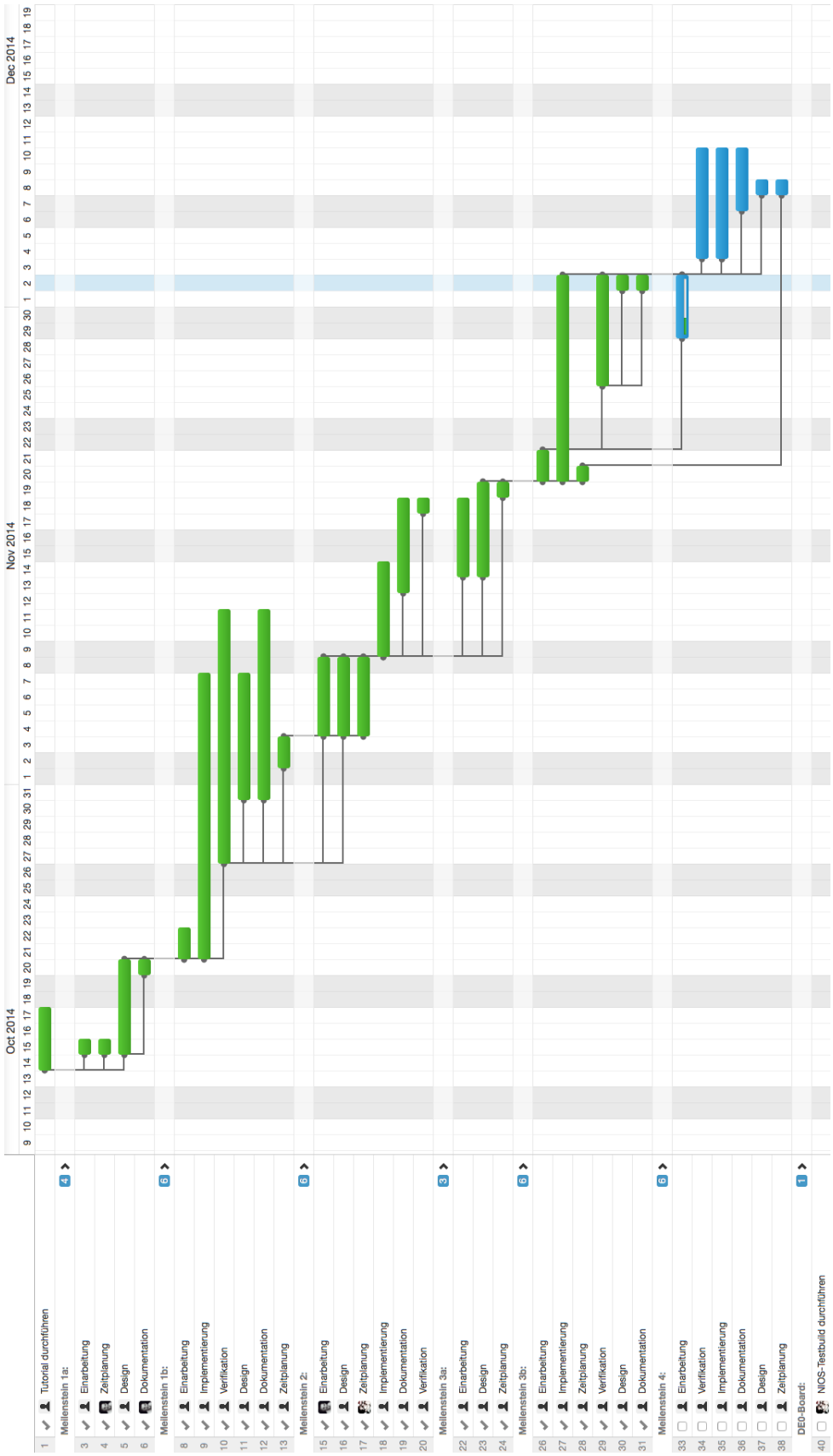


Abbildung 1.1: Gantt-Diagramm zur kompletten Zeitplanung

Meilenstein 3b

	Kossmann		Riedel		gemeinsam		gesamt	
	geplant	benötigt	geplant	benötigt	geplant	benötigt	geplant	benötigt
Einarbeitung	-	-	-	2,43	1	0	1	2,43
Zeitplanung	-	-	-	0,53	1	0,5	1	1,03
Design	-	-	-	-	3	0	3	0
Implementierung	5	6	7	6,32	16	4	28	16,32
Verifikation	-	6	-	11,61	8	0	8	17,61
Dokumentaion	2	3	2	6,31	0	0	4	9,31
Summe	7	15	9	27,2	29	4,5	45	46,7

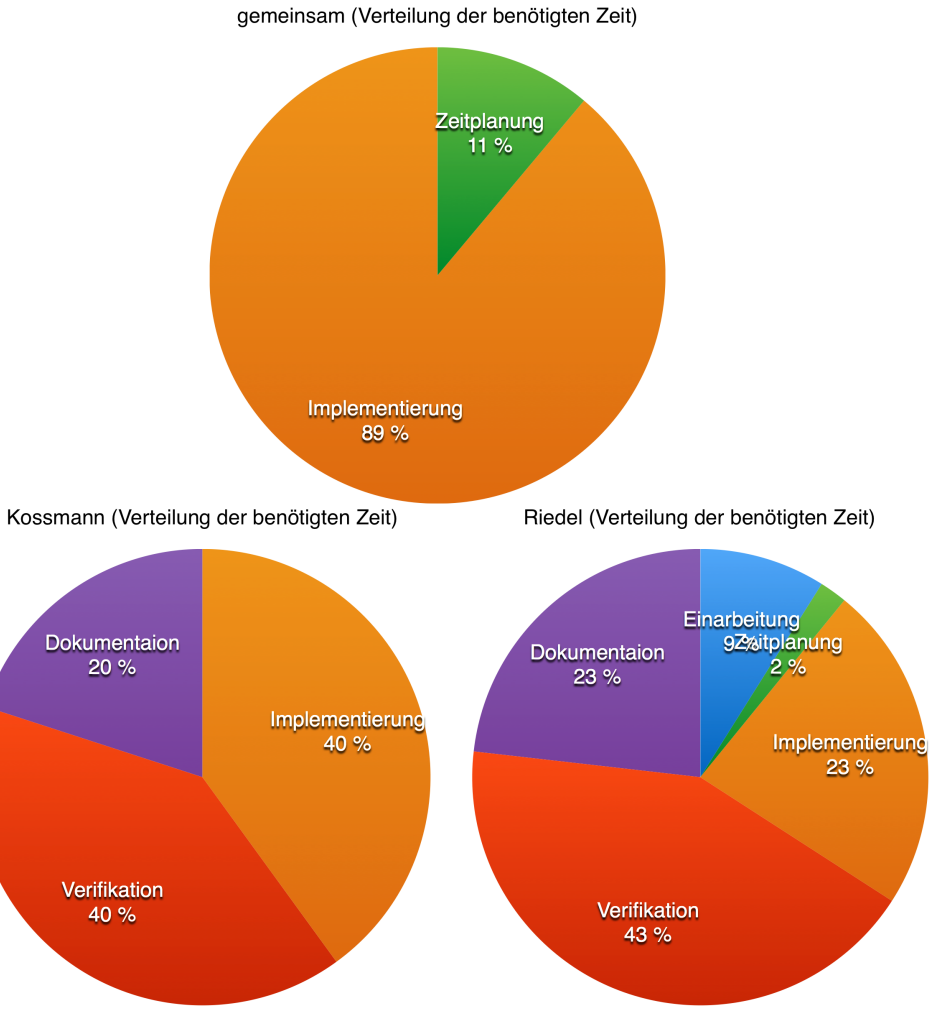
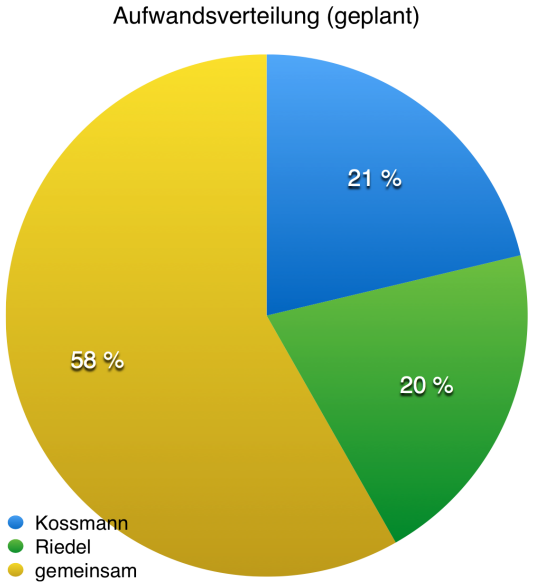


Abbildung 1.2: Projektplanung für Meilenstein 3b

Zeitbedarfsübersicht

	Kossmann	Riedel	gemeinsam	gesamt
geplant	53	51	145	249
benötigt	88,25	97,51	43,52	229,28



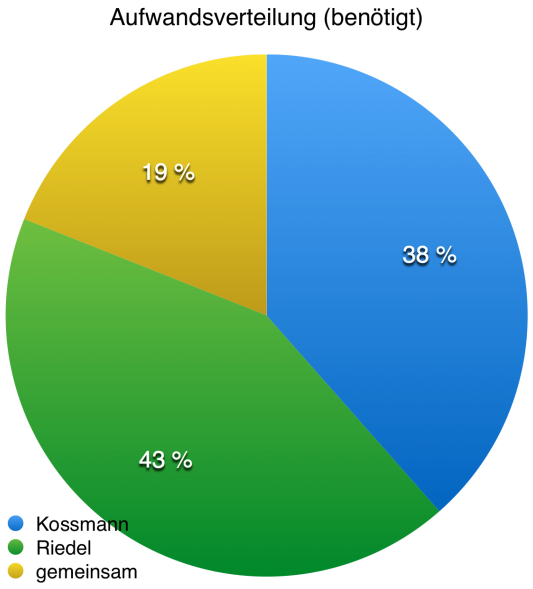


Abbildung 1.3: Zeitbedarfsübersicht für das gesamte Projekt

2. Anpassungen an der Komponente register_interface

Für diesen Meilenstein wurden keine Veränderungen an dieser Komponente vorgenommen. Abbildung 2.1 zeigt das Block-Diagramm des `register_interface`.

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.

Tabelle 3.1: Betriebsmodi der Motor-Control-Unit

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

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 in den `prescaler`-Eingang gespeist.

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:

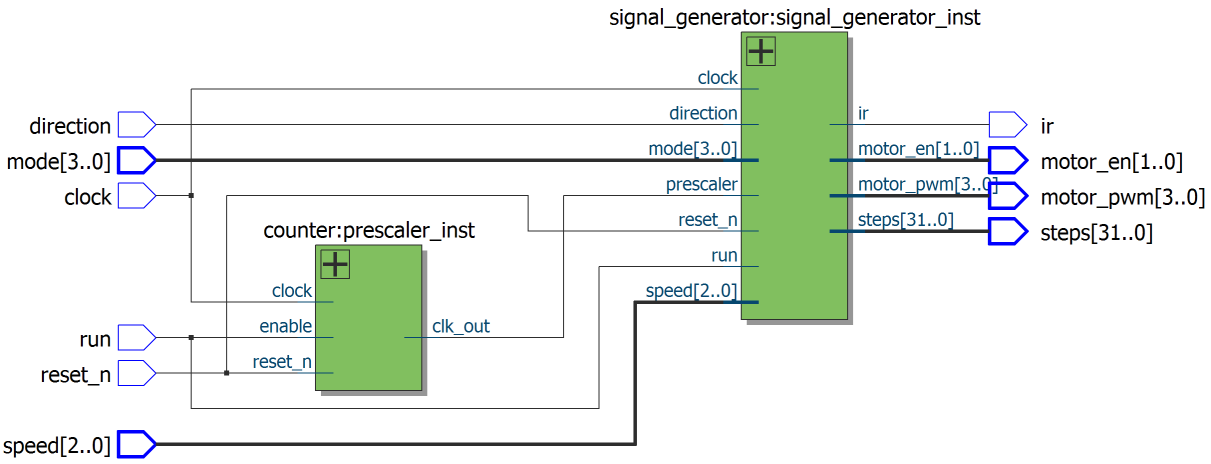


Abbildung 3.1: Block-Diagramm der Motor-Control-Unit-Komponente

$$\text{Taktteiler} = \frac{\text{clk}_{\text{in}}}{\text{clk}_{\text{out}}} \tag{3.1}$$

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

3.3. Design der Komponente signal_generator

Der Signalgenerator wird in Form mehrerer State-Machines realisiert, um die Windungen des Motors in korrekter Reihenfolge und gemäß Eingaben durch den Anwender anzusteuern. Um dies zu ermöglichen, werden verschiedene Sub-State-Machines verwendet, die in den folgenden Abschnitten erläutert werden.

3.3.1. Die Mode-State-Machine

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 `prescaler`-Signal, ein gültiger Mode, ein interrupt-request (`IR`) durch die Steuerung selbst oder der Benutzer einen `reset_n` ausgelöst hat.

3.3.2. Die Speed-State-Machine

Gemäß Abbildung 3.3.2 wird der 3-bit Eingang `speed`auf die zu zählenden Pulsweiten abgebildet. In der vorliegenden Arbeit, wird mit einer Pulsweiten-Auflösung von 5 ms gearbeitet. Die `speed`-Einstellung wird `clock`-synchron übernommen.

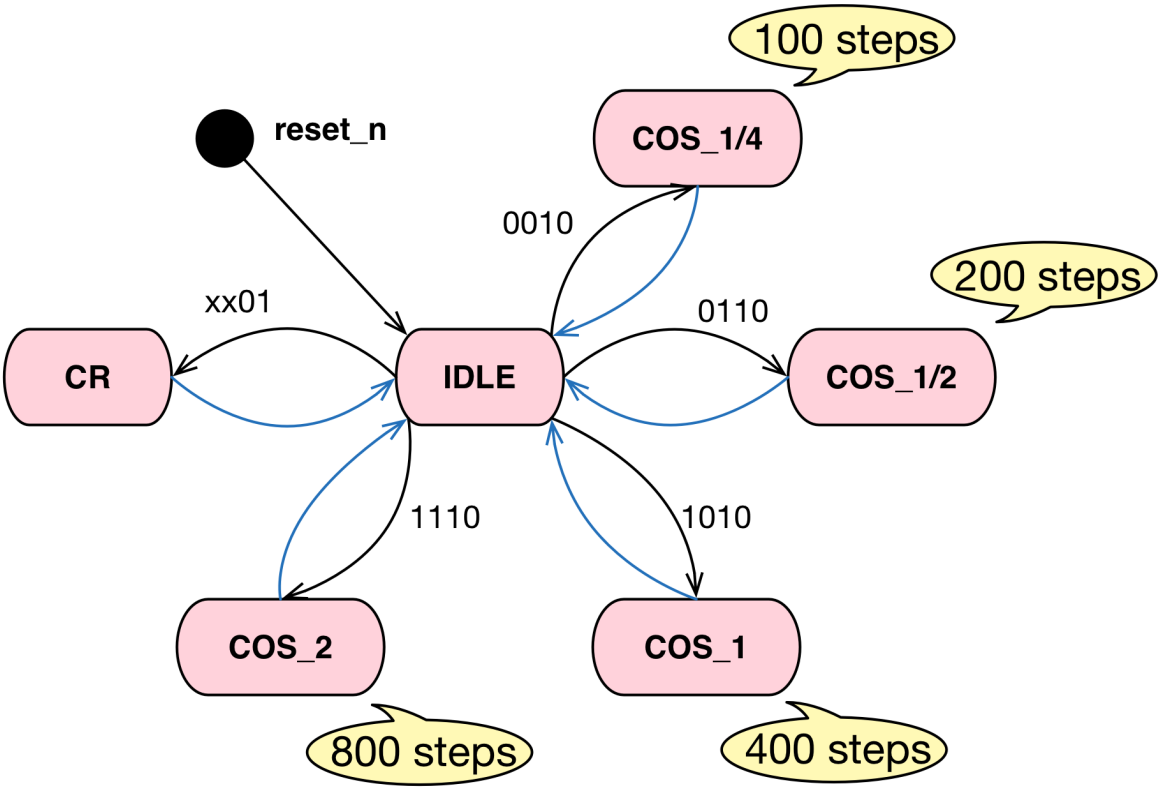


Abbildung 3.2: State-Machine zu den Mode-Übergängen der Motor-Control-Unit-Komponente

[State-Machine zur Verknüpfung der speed-Einstellung mit dem Pulsweiten-Zähler][fig:speed_sm]

3.3.3. Die PWM-State-Machine

Gemäß Abbildung 3.3 ist ersichtlich, wie die Motorwindungen in Abhängigkeit zur `direction` geschaltet werden. Die Umschaltung der Zustände erfolgt nur, wenn eine steigende Flanke des `prescaler`-Eingangs anliegt und der `pwm_5ms_counter = 0` ist. Der `pwm_5ms_counter` wird auf `0` zurückgesetzt, wenn er den eingestellten `speed`-Wert erreicht hat. Im Falle eines `reset_n`-Signals wird er mit `-1` initialisiert. Dieser Wert wird ebenfalls verwendet, wenn sich der `signal_generator` im `IDLE` oder `CR`-Modus befindet, um unendlich viele Schritte zu symbolisieren.

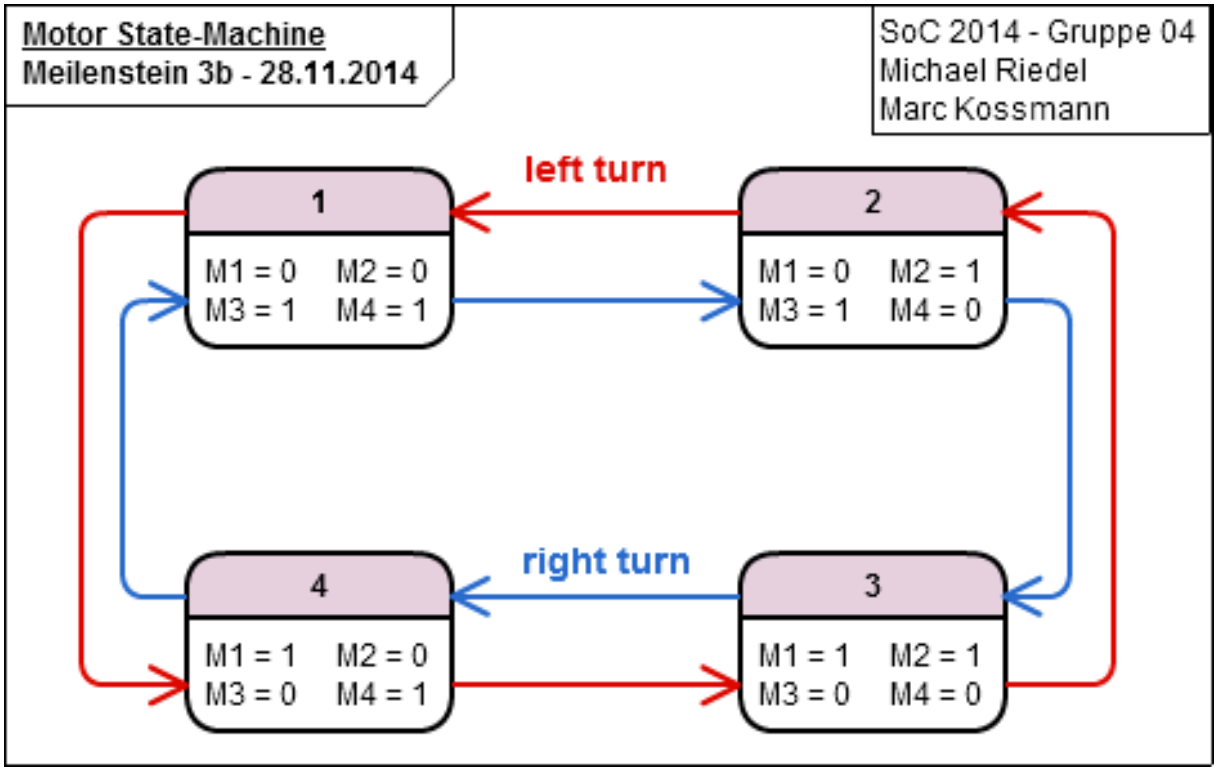


Abbildung 3.3: State-Machine zu Beschaltung der Wicklungen des Schrittmotors

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.

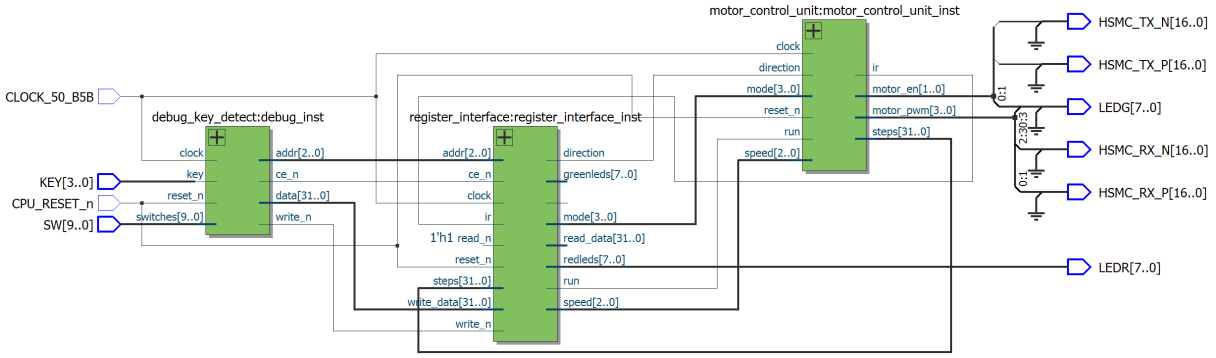


Abbildung 3.4: Block-Diagramm der Testumgebung

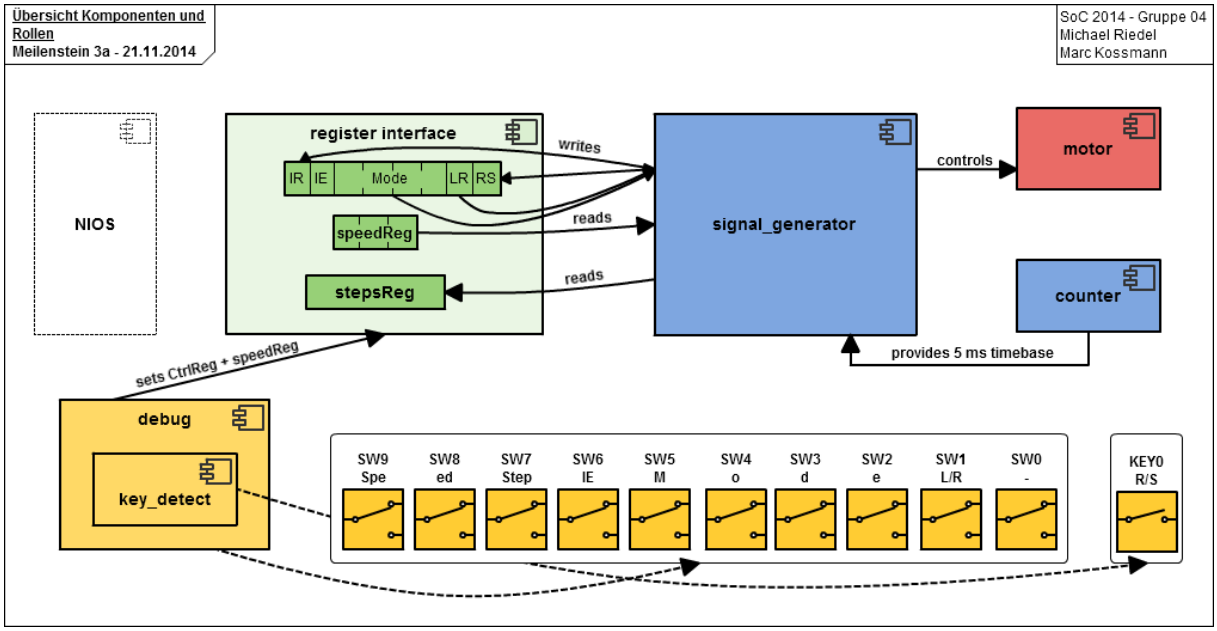


Abbildung 3.5: Überblick der Komponenten und Rollen

4. Test der Motor-Control-Unit

Um die korrekte Funktion der `motor_control_unit` sicherzustellen, wurde eine Testbench erstellt. In ihr werden die verschiedenen Anforderungen an die Komponente verifiziert. Im Allgemeinen werden alle Modi in den verschiedenen Geschwindigkeits- und Richtungseinstellungen getestet. Für weitere Informationen sei auf die Dokumentation des Testbench `motor_control_unit_tb.vhdl` verwiesen.