

CT Praktikum: Finite State Machine (Waschmaschine)

1 Funktion

In diesem Praktikum implementieren Sie eine Finite State Machine (FSM) als Steuerung für eine Waschmaschine. Abbildung 1 zeigt das im Praktikum verwendete Modell der Waschmaschine. Das Modell kann ans CT-Board angeschlossen werden.

Es sind zwei Programme, Waschen und Schleudern, vorhanden. Das Programm Schleudern soll 5 s laufen. Das Programm Waschen heizt das Wasser auf, bis der Temperaturfühler anspricht, bewegt die Waschtrommel dann 4 s nach links und anschliessend 4 s nach rechts.

2 Lernziele

- Sie können eine Funktionsbeschreibung in ein einfaches 'UML State Diagram' umsetzen.
- Sie können ein 'UML State Diagram' in C implementieren.



Abbildung 1: Modell Waschmaschine

3 Schema der Schaltung

Abbildung 2 zeigt das Schema der Waschmaschineneingänge und -ausgänge.

Die Eingänge sind alle „active-low“. Eine logische „0“ am Eingang 'Door Closed' bedeutet beispielsweise, dass die Türe geschlossen ist. Flanken auf den Signalen zeigen Ereignisse (Events) an. Eine fallende Flanke des Signals 'Door Closed' bedeutet also ein Schliessen, während eine steigende Flanke ein Öffnen der Türe darstellt.

Die Ausgänge sind ebenfalls „active-low“. Beispiel: Soll der Motor langsam laufen, muss der Ausgang Bit 3 (Motor Slow) eine „0“ sein und der Ausgang Bit 2 (Motor Fast) eine „1“.

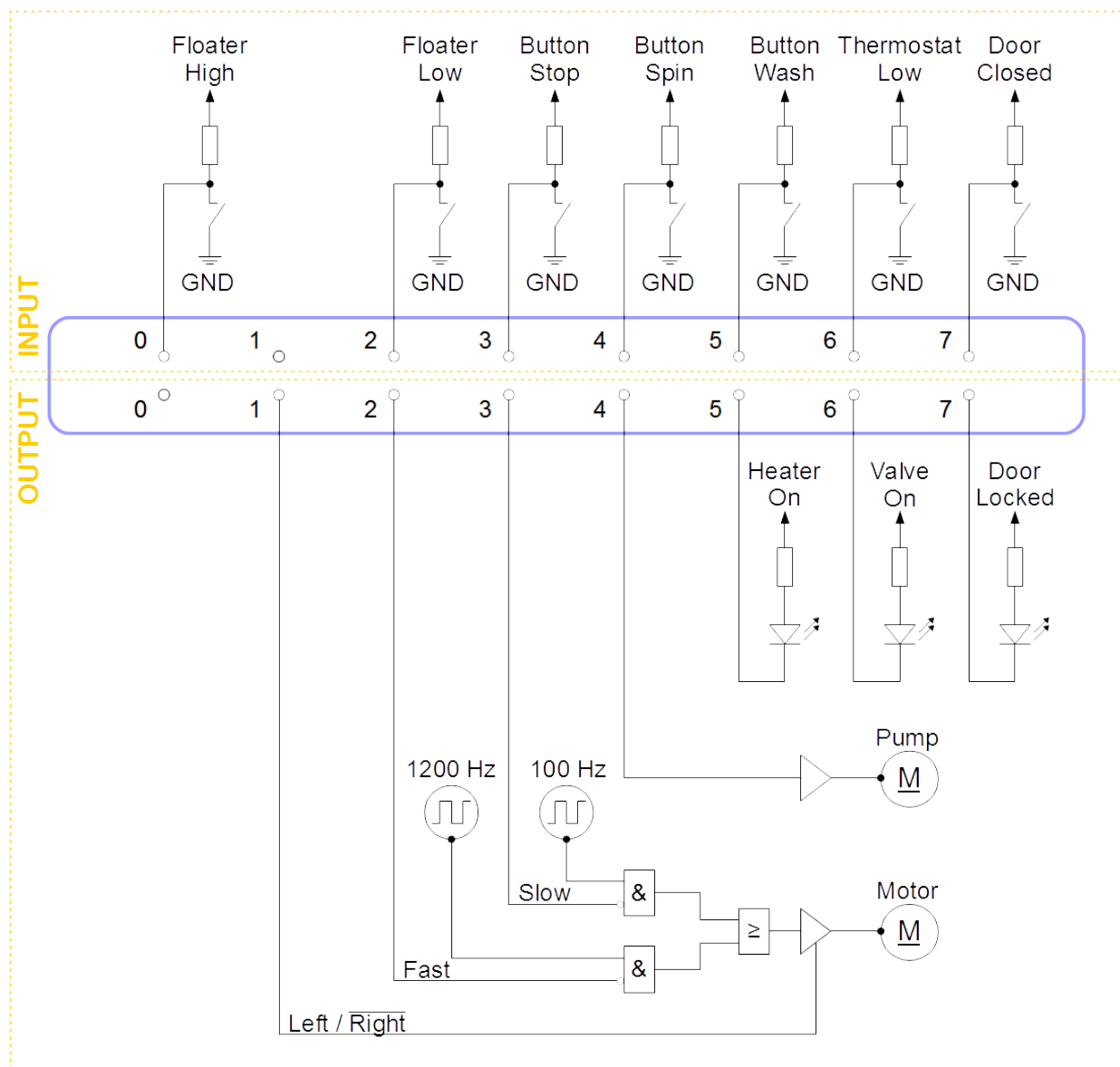


Abbildung 2: Schema der Waschmaschinen Ein- und Ausgänge

Die Waschmaschine wird über die GPIO Ports des CT Boards angeschlossen. Verwenden Sie Port P1 gemäss Abbildung 3.

Hinweis: Für diese Funktion muss das CT Board in *Modus 1* betrieben werden!

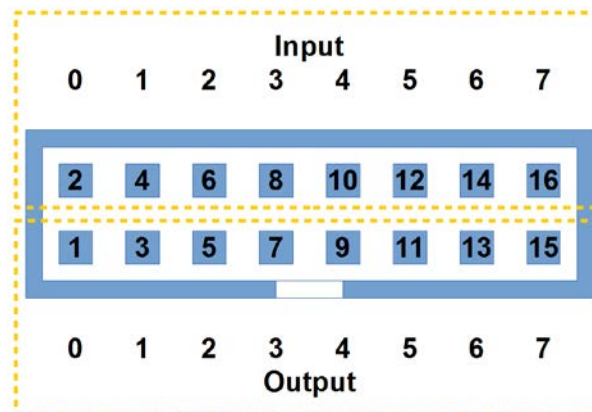
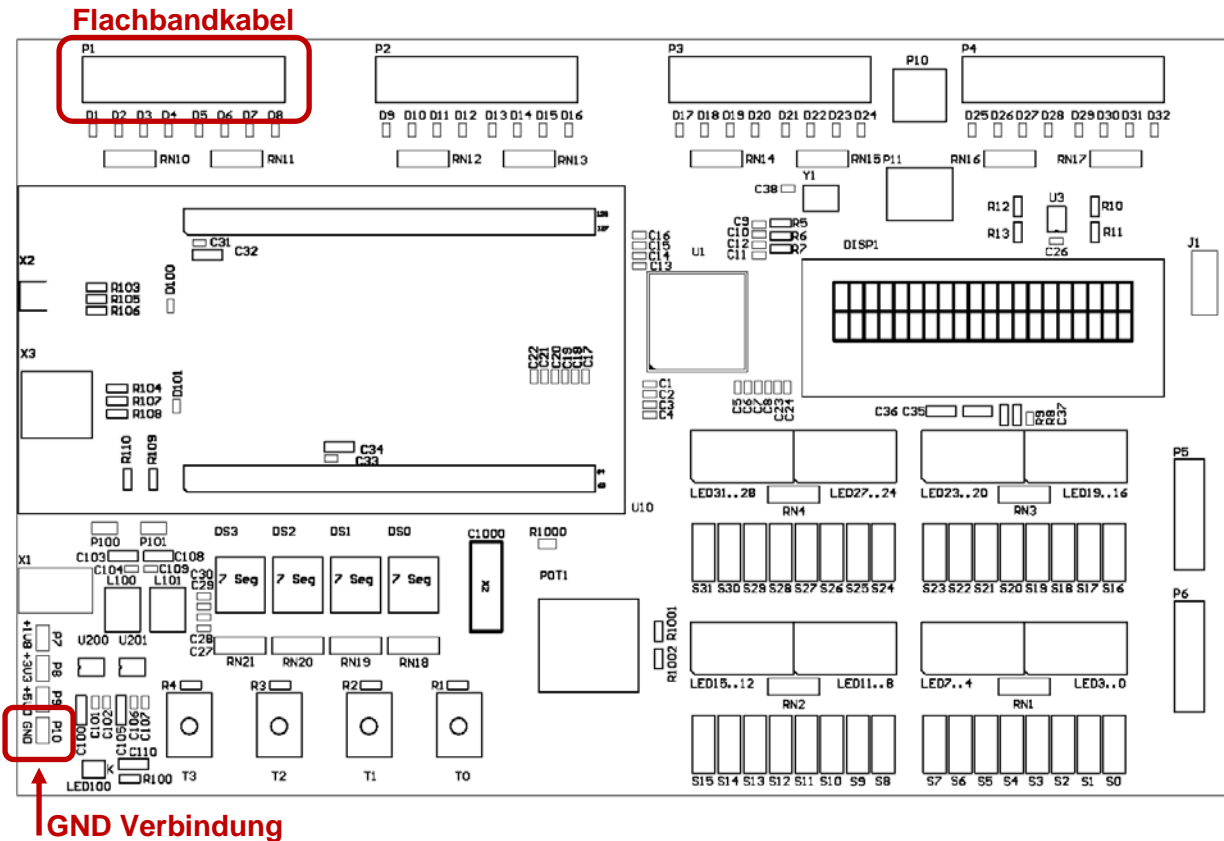


Abbildung 3: Anschluss des Waschmaschinenmodells an CT-Board

Die Waschmaschine wird über ein separates 5V Netzteil gespeist. Daher muss der GND mit dem CT Board verbunden werden:

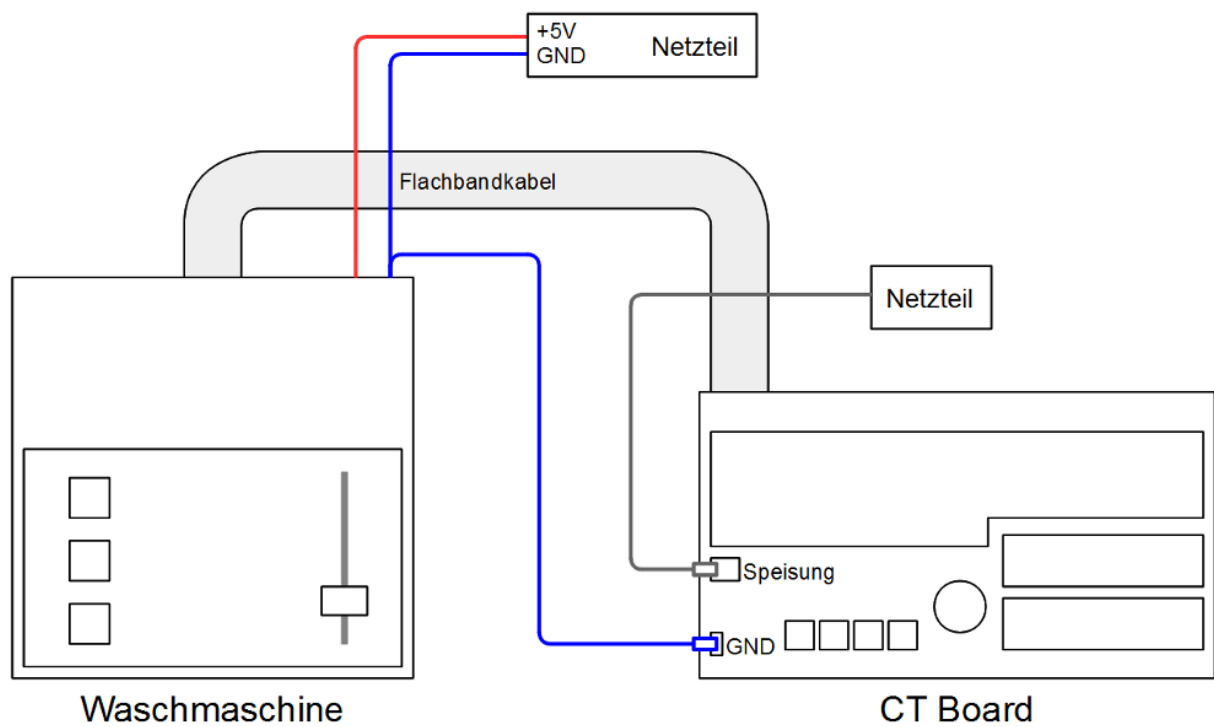


Abbildung 4: Anschluss des Waschmaschinenmodells an CT-Board

4 Aufgaben

Die Funktionen der zwei (Waschmaschinen-) Programme werden schrittweise umgesetzt.

4.1 Türkontrolle

In einem ersten Schritt soll eine einfache FSM gemäss Abbildung 5 für den „Türöffnungsmechanismus“ implementiert werden:

- Erweitern Sie die Funktion `eh_get_event()` im Modul `event_handler` so, dass Sie erkennen können, wann die Türe geschlossen wurde.
 - Detektieren Sie die negative Flanke, die entsteht wenn der entsprechende Eingang von „1“ auf „0“ wechselt. Geben Sie den Event `DOOR_CLOSED` zurück.
 - Detektieren Sie die positive Flanke, die entsteht wenn der entsprechende Eingang von „0“ auf „1“ wechselt. Geben Sie den Event `DOOR_OPEN` zurück.
- Implementieren Sie die Funktion `fsm_handle_event()` im Modul `state_machine` gemäss dem Diagramm in Abbildung 5.

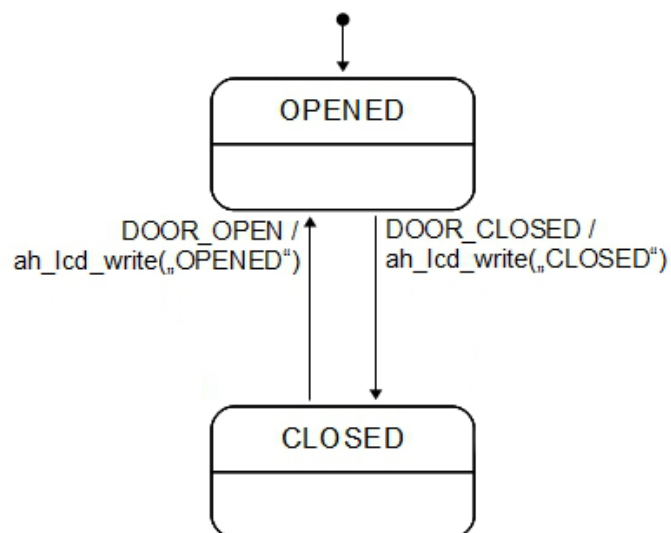


Abbildung 5: 'UML State Diagram' der Türkontrolle

Hinweis: Das Modul `action_handler` enthält eine Funktion `ah_lcd_write(char text[])`. Mit dieser Funktion können Sie Strings für das Debugging auf den LCD ausgeben. Zeigen Sie damit bei einer Transition den Folgezustand an.

Beispiel: `ah_lcd_write("CLOSED");`

4.2 Schleudergang

Im zweiten Schritt wird der Schleudergang gemäss Abbildung 6 implementiert. Benutzen Sie dazu das vorgegebene Modul `timer` um die notwendigen Verzögerungen zu implementieren.

- Erweitern Sie die Funktion `eh_get_event()` im Modul `event_handler` so, dass Sie Timer-Events erkennen können. Wenn der aktuelle Wert des Timers gleich 0 ist und der vorherige Wert ungleich 0 ist, soll der Event `TIME_OUT` erzeugt werden. Der Timer Event soll dabei die höchste Priorität haben.
- Implementieren Sie die Funktionen für die benötigten Aktionen im Modul `action_handler`. Die Funktionsrahmen sind bereits vorgegeben.
- Erweitern Sie die State Machine so, dass die zusätzlichen Transitionen unterstützt werden.
- Den Timer können Sie mittels der Funktion `timer_start(duration)` starten, und mittels `timer_stop()` stoppen, falls der Timer abgebrochen werden soll. Der Funktionsparameter `duration` bezieht sich auf die Anzahl Perioden eines 100 Hz Signals.
- Beachten Sie, dass zwei Events eine Transition von `FAST_SPIN` nach `CLOSED` ermöglichen. Beide müssen unterstützt werden.

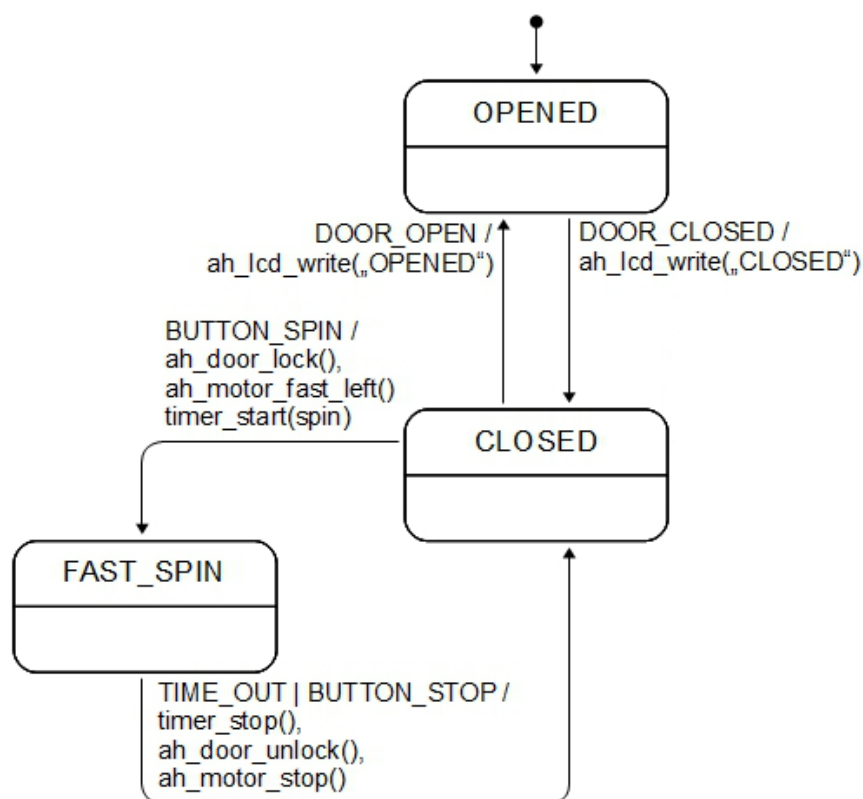


Abbildung 6: 'UML State Diagram' des Schleudergangs

4.3 Komplett

Aufgabe a)

Das vollständige Waschprogramm soll zuerst Wasser in die Waschtrommel einlassen. Dieses Wasser wird dann aufgeheizt um anschliessend den Waschgang durchzuführen (4s nach links, 4s nach rechts). Danach wird das Schmutzwasser abgepumpt. Nun kommt der Schleudergang, den Sie ja bereits umgesetzt haben.

Erweitern Sie ihr Programm um die fehlenden Funktionen:

- Zeichnen Sie dazu ein vollständiges 'UML State Diagram'.
- Setzen Sie dieses in C-Code um und testen Sie Ihren Code am Modell.

(Optional) Aufgabe b)

Erweitern Sie die in Aufgabe a) gestellte Funktion um den Stop Taster. Das laufende Programm soll jederzeit durch die Stop Taste angehalten werden können. Beachten sie dabei, dass die Maschine unter Umständen voll Wasser ist und handeln entsprechend.

4.4 Bewertung

Die lauffähigen Programme müssen präsentiert werden. Die einzelnen Studierenden müssen die Lösungen und den Quellcode verstanden haben und erklären können.

Bewertungskriterien	Gewichtung
Die Türkontrolle aus Aufgabe 4.1 ist funktionsfähig.	1/4
Der Schleudergang aus Aufgabe 4.2 wurde umgesetzt, der Timer funktioniert.	1/4
Die komplette Waschmaschine funktioniert gemäss den Vorgaben aus Aufgabe 4.3 a)	2/4