

Lucius

2016

Inhaltsverzeichnis

1	Allgemeines	2
1.1	Laboraufbau	2
1.2	Inbetriebnahme	2
2	Hardware	4
3	Software	4
4	<i>Empfohlener Übungsablauf</i>	6
4.1	Aufbau des Programms	6
4.2	Variationen	6
4.3	Benotung	7

Hinweis zur Laborübung: Bitte lesen Sie *vor der Übung* folgende Dokumente sorgfältig durch:

- diese Angabe (Lucius.pdf)
- das Dokument Hinweise_zum_XMC1100.pdf
- im XMC1100-Reference Manual
 - die Abschnitte über externe Interrupts (ERU und CCU4)
 - die Abschnitte über Timer (System Timer und CCU4)
 - den Abschnitt über Interrupts (Interrupt Subsystem)

Nützen Sie die genannten Unterlagen für die Laborvorbereitung! Die Wahrscheinlichkeit, dass Ihnen beim Eingangstest Fragen daraus gestellt werden ist hoch.

1 Allgemeines

Ziel dieser Aufgabe ist es, den Umgang mit externer Peripherie zu erlernen. Der Aufbau, welcher in Abbildung 1 zu sehen ist, besteht aus einem XMC1100 Boot Kit, einem Liftmodell mit Kabine, Tasten und LEDs. Ziel der Aufgabe ist es, eine Liftsteuerung zu implementieren.

1.1 Laboraufbau

Der Übungsaufbau - auch Lucius genannt - setzt sich aus folgenden Komponenten zusammen:

- Mikrocontroller-Board mit XMC1100 Mikrocontroller
- Liftmodell mit Schnittstelle

1.2 Inbetriebnahme

Schalten Sie das Netzgerät des Liftmodells ein um den Lift in Betrieb zu nehmen und stecken Sie anschließend den Mikrocontroller mittels USB-Kabel an den PC. Sollte während des Betriebs (z.B. durch Dauerbetrieb oder Wechsel von Hinauffahren zu Hinunterfahren ohne kurzem Stillstand dazwischen) die

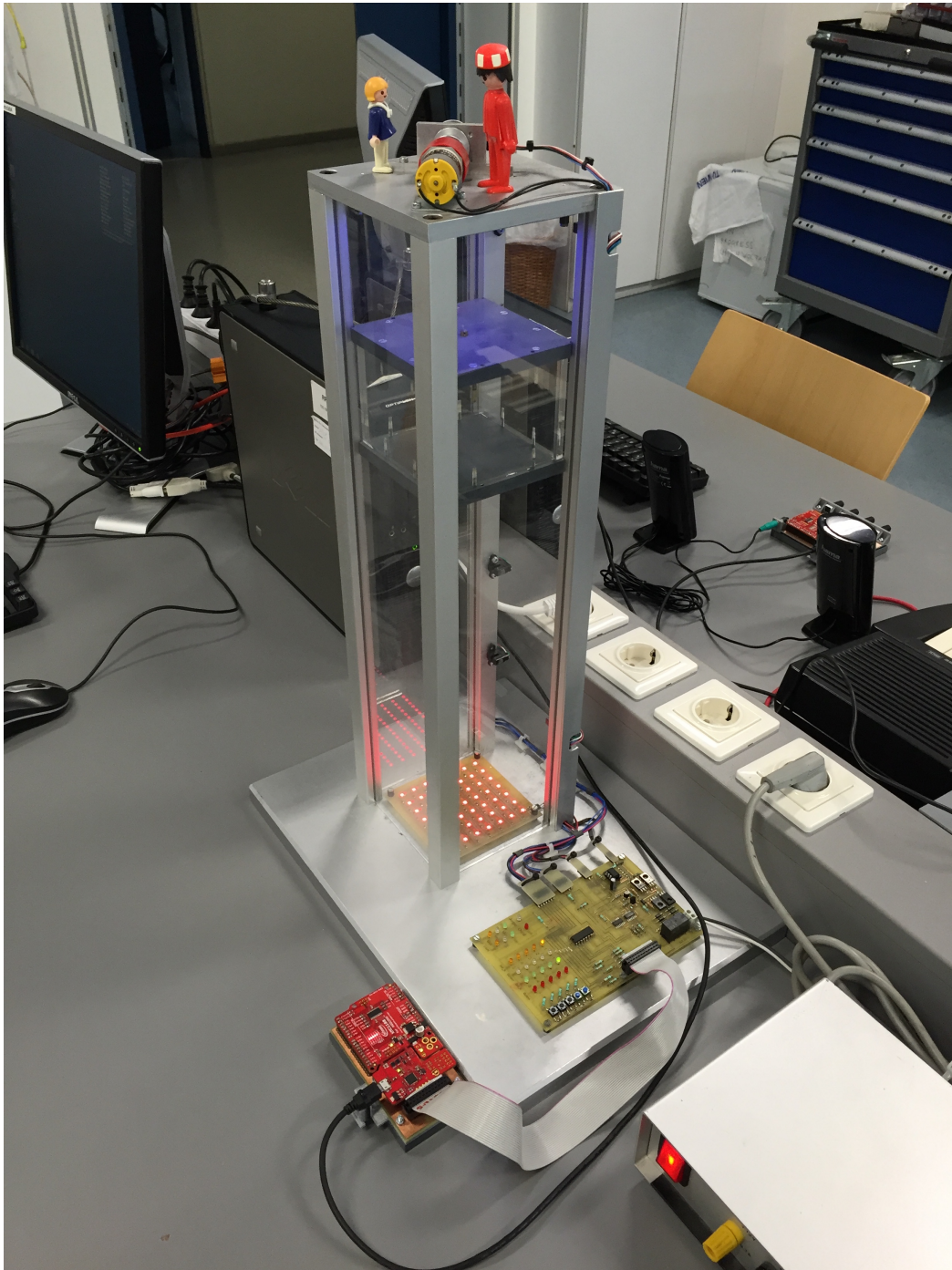


Abbildung 1: Im Labor aufgebautes Liftmodell

rote LED "Overload" aufleuchten, so ist die Hardware überhitzt. Schalten Sie dann das Netzgerät aus, warten Sie einige Minuten und schalten Sie das Netzgerät wieder ein.

Lesen Sie anschließend die Slides "Entwicklung mit dem XMC1100" ([xmc1100_einführung.pdf](#)) und befolgen Sie die Anweisungen darin, um ein Projekt anzulegen und den Aufbau zum ersten Mal in Betrieb zu nehmen.

2 Hardware

Die Zuordnung der entsprechenden Peripherie des XMC1100 zu den Steuereinheiten des Lifes (wie z.B.: Taster, Lichtschranken,...) sind in der Tabelle 1 angegeben. Beachten Sie, dass die Tasten Prellen können und überlegen Sie sich wie Sie die Tasten mit Ihrer Software „Entprellen“ können.

3 Software

Sämtliche Eingangsleitungen sind an Pins angeschlossen, die externe Interrupts auslösen können, NÜTZEN Sie diese Möglichkeit. Die gesamte Steuerung soll über Interrupts erfolgen. Unnötige Warteschleifen führen zu Abzügen.

Über die Tasten soll der Lift in ein gewisses Stockwerk gerufen werden, wobei mehrere Stockwerke gleichzeitig ausgewählt sein können. Kein Tastendruck darf verloren gehen. Der Aufzug soll dabei in der aktuellen Fahrtrichtung zuerst das nächstgelegenen Stockwerk mit einem Haltewunsch anfahren, kurz halten und dann zum Nächsten weiter fahren. Wenn in der aktuellen Fahrtrichtung kein weiteres Stockwerk gewählt ist, soll die Richtung geändert werden und das gleiche Verfahren erneut angewandt werden.

Um die LEDs der Sets A und B zu aktivieren, muss zuerst das entsprechende Bitmuster an den Leitungen Set_bX angelegt werden und dann ein Impuls (bestehen aus 2 Taktflanken) am entsprechenden Clk_setA/B Signal generiert werden. In jedem Set kann immer nur eine LED aktiv sein.

Das LED Set A soll immer das aktuelle Stockwerk anzeigen, in dem sich der Aufzug gerade befindet. Im LED Set B sollen alle Stockwerke für welche derzeit ein Haltewunsch besteht angezeigt werden. Damit beide LED Sets gleichzeitig leuchten, müssen diese mittels Zeitmultiplexens abwechselnd angesteuert werden, wobei ein flimmerfreier Seheindruck erreicht werden soll.

Bezeichnung	XMC Eingang	XMC Ausgang	Beschreibung
Soll_UP		P1.1	Lift fährt nach oben
LB_UP	P0.7		Oberer Anschlag erreicht
Sky		P0.9	Obere LED Matrix
Soll_DOWN		P1.0	Lift fährt abwärts
LB_DOWN	P0.1		Unterer Anschlag erreicht
Hell		P0.8	Untere LED Matrix
Overload	P0.3		Motor Überlast
LB_Rotate	P0.12		Rotationsimpulse
Set_b2(MSB)		P1.2	Ledadresse für set A/B
Set_b1		P1.3	Ledadresse für set A/B
Set_b0(LSB)		P1.4	Ledadresse für set A/B
Clk_SetA		P0.6	Adressierte LED im Set A anzeigen
Clk_SetB		P0.5	Adressierte LED im Set B anzeigen
LB_5	P2.4		Lichtschranke Stock 4
LB_4	P2.8		Lichtschranke Stock 3
LB_3	P0.0		Lichtschranke Stock 2
LB_2	P2.0		Lichtschranke Stock 1
LB_1	P0.2		Lichtschranke EG
Button_1	P2.10		Taste 1
Button_2	P2.6		Taste 2
Button_3	P2.5		Taste 3
Button_4	P2.7		Taste 4
Button_5	P2.1		Taste 5

Tabelle 1: Datenleitungen

Der Motor wird über High-Pegel an Soll_UP und Soll_DOWN gesteuert. Je näher die Kabine dem unteren bzw. oberen Ende des Schachtes kommt, umso heller soll die entsprechende LED Matrix leuchten, in gleichem Maße soll die andere dunkler werden. Bilden sie mittels PWM-Signalgenerierung einen linearen Helligkeitsgradienten. Die Signale von LB_Rotate können ihnen bei der Abstandsmessung zu den Endpunkten helfen.

HINWEIS: Wenn die Kabine die Lichtschranken LB_DOWN oder LB_UP (die Endanschläge) unterbricht, kann die Kabine nur noch in die entgegengesetzte Richtung bewegt werden. Im Betrieb sollte dies nie passieren, es könnte Ihnen allerdings während der Initialisierungsphase helfen.

b2	b1	b0	LED
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	X
1	1	0	X
1	1	1	X

Tabelle 2: LED Adressen Set A/B

4 *Empfohlener* Übungsablauf

4.1 Aufbau des Programms

Bauen Sie Ihr Programm modular auf! Das Beispiel setzt sich aus *fünf Teilaufgaben* zusammen:

- Einlesen und **Entprellen** der Tasten
- Ansteuerung der LED Sets A+B
- Ansteuerung des Motors
- Auslesen der Lichtschrankeninformationen
- Fahrlogik

Implementieren und testen Sie die Teilaufgaben so weit als möglich separat und führen Sie diese erst dann zur Gesamtlösung zusammen.

4.2 Variationen

Seien Sie darauf gefasst, dass während der Laborübung zusätzliche Teilaufgaben und Features zu Ihrem Beispiel von den Tutoren hinzugefügt werden können. Ein Beispiel:

- Steuern Sie den Motor unter Verwendung von PWM Pulsen und den Informationen von der Lochscheibe so, dass der Motor wenn er sich dem gewünschten Stockwerk nähert kontinuierlich langsamer wird bis er schließlich völlig steht.

4.3 Benotung

Die Note für das Labor setzt sich aus dem Eingangsgespräch und dem Abgabegespräch zusammen. Voraussetzung für das Abgabegespräch ist die Vollendung der Übungsaufgabe.

Der Grad Ihrer Note hängt stark von der Qualität des von Ihnen implementierten Codes ab. Sollten Sie die Anforderungen aus dem Kapitel Software erfüllen dürfen Sie mit einem positiven Übungsabschluss rechnen. Je besser die Entprellung der Tasten, die Fahrlogik, die „lineare“ Ansteuerung der LED Matritzen und die Zeitgemultiplexte Ansteuerung der LED Set_A/B funktioniert, desto besser ihre Note.

Beachten Sie aber, dass eine schlechte Leistung beim Abgabegespräch die Note noch verschlechtern kann. Selbiges gilt auch für das Eingangsgespräch.