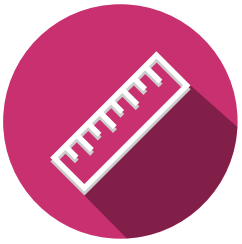
Cursor - Motorsteuerung

Projekt Digitales Design

Hes·so VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

Orientierung: Systemtechnik (SYND)
Kurs: Digitales Design (DiD)
Verfasser: Christophe Bianchi, François Corthay, Silvan Zahno, Axel Amand
Datum: 11. April 2023
Version: v2.0



Inhaltsverzeichnis

1 Einführung

Das Projekt hat die Aufgabe das erlernte Wissen am Ende des Semesters an einem praktischen Beispiel direkt anzuwenden. Es soll ein Gleichstrommotor so angesteuert werden, damit ein Schlitten auf einer Schraube, vordefinierte Punkte präzise anfahren kann. Das System kann in der Abbildung 1 betrachtet werden.

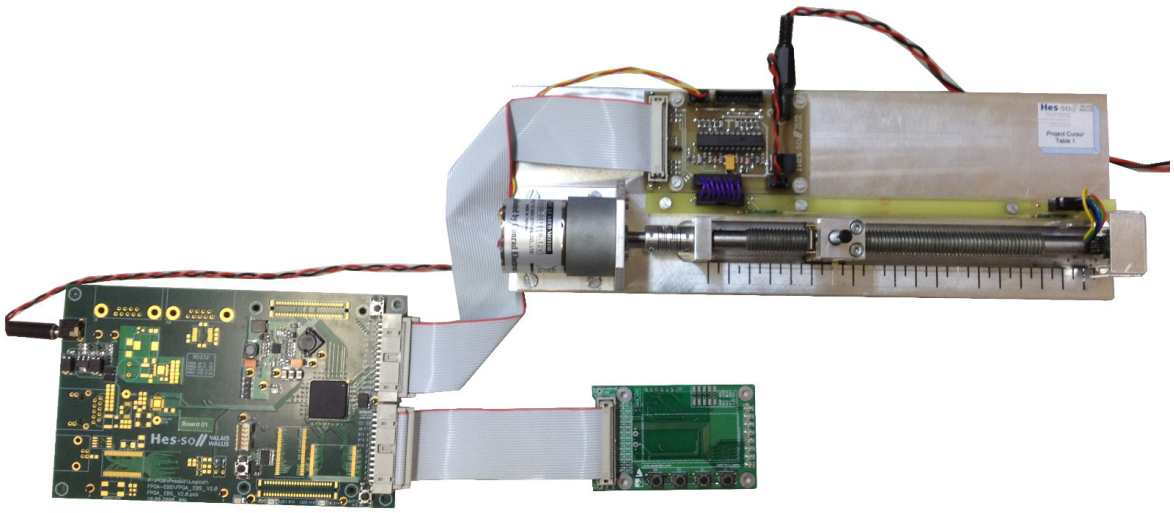
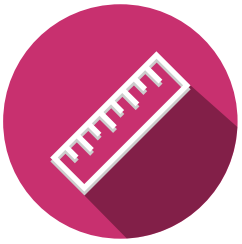


Abbildung 1: Hardwareaufbau Cursor

Die Aufgabe besteht aus einer klar definierten minimalen Spezifikation (Kapitel 2), welche von der Entwicklungsgruppe mit zusätzlichen Funktionen optional erweitert werden kann. Den Ideen sind hier keine Grenzen gesetzt, als Beispiel kann das LCD Display benutzt werden um bestimmte Informationen anzuzeigen.

Mithilfe von Zusatzfunktionen können einige Extrapunkte erarbeitet werden.



2 Spezifikation

2.1 Funktionen

Die Basisfunktionen sind wie folgt definiert:

- Wenn die *restart*-Taste gedrückt wird, bewegt sich der Cursor auf die Startposition, die von einem Reed-Relais in der Nähe des Gleichstrommotor angegeben wird.
- Wenn die Taste $Position_1$ gedrückt wird, muss der Cursor in Richtung Position 1 (p_1) zuerst stetig beschleunigen, danach mit voller Geschwindigkeit vorrücken und schlussendlich stetig abbremsen um an der Position 1 (p_1) anzuhalten. Dies kann von der Startposition oder von der Position 2 aus geschehen, siehe Abbildung 2.
- Wenn die Taste $Position_2$ gedrückt wird, muss der Cursor in Richtung Position 2 (p_2) zuerst stetig beschleunigen, danach mit voller Geschwindigkeit vorrücken und schlussendlich stetig abbremsen um an der Position 2 (p_2) anzuhalten, siehe Abbildung 2.

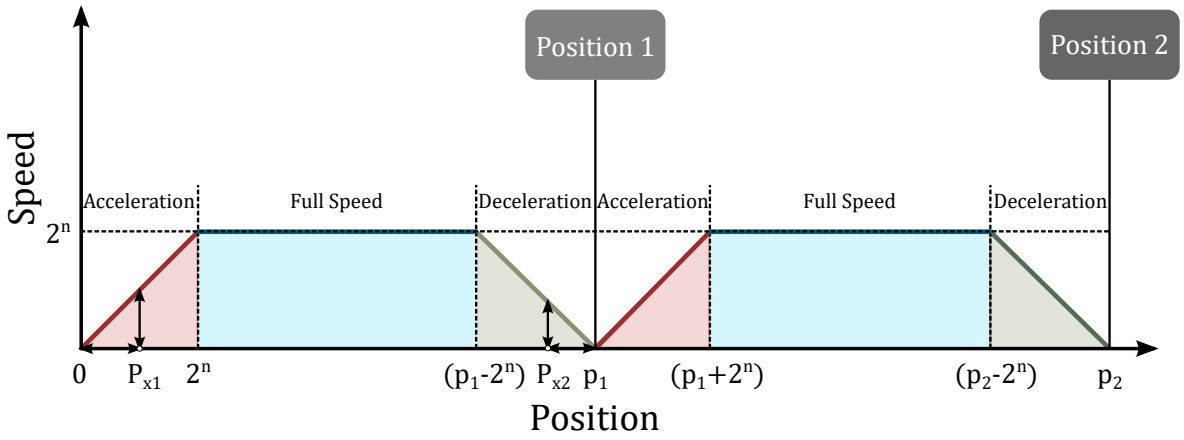


Abbildung 2: Diagramm der Geschwindigkeit des Schlittens

Die Beschleunigungs- und Abbremsrampen sind eine Funktion der Position und nicht der Zeit. Es wäre in der Tat sehr schwierig zu wissen, wann mit der Abbremsung begonnen werden muss, um eine der Positionen zu erreichen, wenn die Abbremsung von der Zeit und nicht von der Position abhinge. Die Steigung der Rampe muss so gewählt werden, dass die Beschleunigungs- und Abbremswege in der Grössenordnung von $1cm$ liegen.

Die zu erreichenden Positionen sind:

- Position 1 (p_1) = 8 cm
- Position 2 (p_2) = 12 cm

2.2 Schaltung

Um die Aufgabe zu lösen wird folgende Schaltung vorgegeben um den Schlitten zu bewegen.

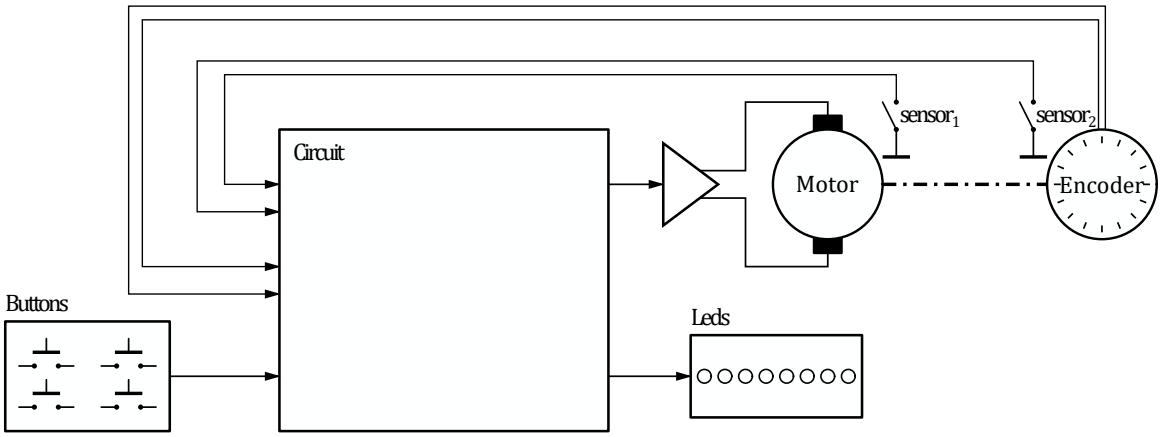


Abbildung 3: Cursor Schaltung

Die Schaltung funktioniert wie folgt:

- Der Gleichstrommotor wird durch die drei Signale $motor_{On}$, $side_1$, $side_2$ gesteuert. Seine Geschwindigkeit wird durch eine PWM-Modulation auf den Signalen $side_1$ oder $side_2$ gesteuert.
- An den Enden der Schiene befinden sich zwei Reed-Relais [9], die das Vorhandensein des Cursor-Wagens erkennen ($sensor_1$ sowie $sensor_2$).
- Der Drehgeber (Encoder) wird verwendet, um die Position des Cursors zu verfolgen, respektive zu zählen. Seine drei Encoderausgänge, $encoder_A$, $encoder_B$ und $encoder_I$, ermöglichen es, die Bewegungen der Schraube zu verfolgen.
- Drei Tasten werden zur Steuerung des Systems verwendet: $restart$, go_1 und go_2. Eine zusätzliche Taste, $button_4$, kann für optionale Funktionen verwendet werden.
- Die $testOut$-Pins können verwendet werden, um zusätzliche Informationen aus dem System auszugeben, z.B. für Debugging oder um die LED zu steuern.

Das Empty Toplevel Design (cursor-toplevel-empty.pdf) zeigt alle Signale, die mit der **F**ield **P**rogrammable **G**ates **A**rray (FPGA)-Platine verbunden sind 4.

Abbildung 4: Leere Toplevel Schaltung

2.3 Szenario (Beispiel)

In der Abbildung 5 sind drei unterschiedliche Szenarien ersichtlich. Zuerst wird die Taste *restart* gedrückt und der Schlitten wird mit voller Geschwindigkeit zur Initialposition ($sensor_1$) bewegt. Die zwei weiteren Szenarien go_2 sowie go_1 bewegen den Schlitten zur vorgegebener $Position2$ respektive $Position2$, hierbei wird auf den Signalen $side_2$ respektive $side_1$ ein sich veränderndes PWM Signal angelegt um den Schlitten zu beschleunigen und zu verlangsamen.

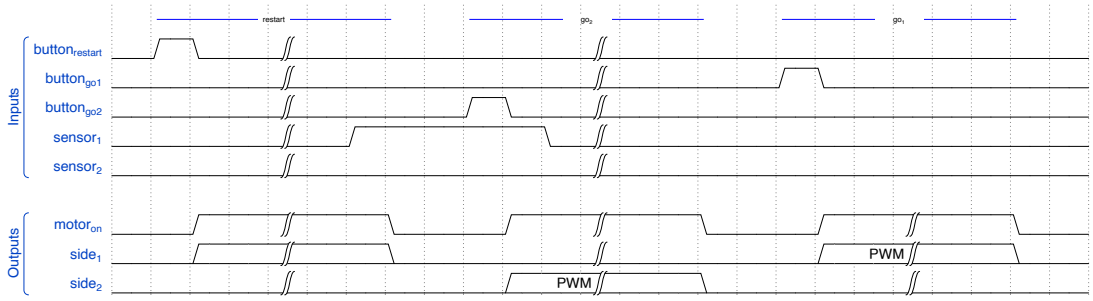
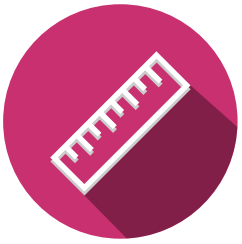


Abbildung 5: Cursor Szenario

Die obigen Szenarien sind Beispiele, es liegt an den Studenten diese zu komplettieren.



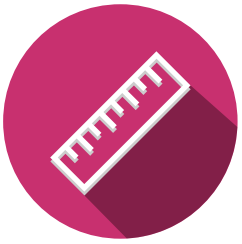
2.4 HDL-Designer Projekt

Ein vordefiniertes HDL-Designer Projekt kann im Cyberlearn heruntergeladen oder geklont werden. Die Dateistruktur des Projektes sieht folgendermassen aus:

```
did_cursor
+--Board/           # Project and files for programming the FPGA
|   +--concat/      # Complete VHDL file including PIN-UCF file
|   +--ise/         # Xilinx ISE project
+--Cursor/          # Library for the components of the student solution
+--Cursor_test/     # Library for the simulation testbenches
+--doc/             # Folder with additional documents relevant to the project
|   +--Board/       # All schematics of the hardware boards
|   +--Components/  # All data sheets of hardware components
+--img/             # Pictures
+--Libs/            # External libraries which can be used e.g. gates, io, sequential
+--Prefs/           # HDL-Designer settings
+--Scripts/         # HDL-Designer scripts
+--Simulation/      # Modelsim simulation files
```

Der Pfad des Projektordners darf keine Leerzeichen enthalten.

Im Projektordner *doc/* können viele wichtige Informationen gefunden werden. Datenblätter, Projektbewertung sowie Hilfsdokumente für HDL-Designer um nur einige zu nennen.



3 Komponenten

Das System besteht aus 3 verschiedenen Hardwareplatinen, welche in der Abbildung 1 ersichtlich sind:

- einer Wagenbaugruppe mit einer elektronischen Platine "**P**rinted **C**ircuit **B**oard (PCB)", die den Motor antreibt und die Sensoren ausliest, siehe Abbildung 6
- einer FPGA Entwicklungsplatine, siehe Abbildung 14
- einer Steuerplatine mit 4 Tasten und 8 LEDs, siehe Abbildung 15

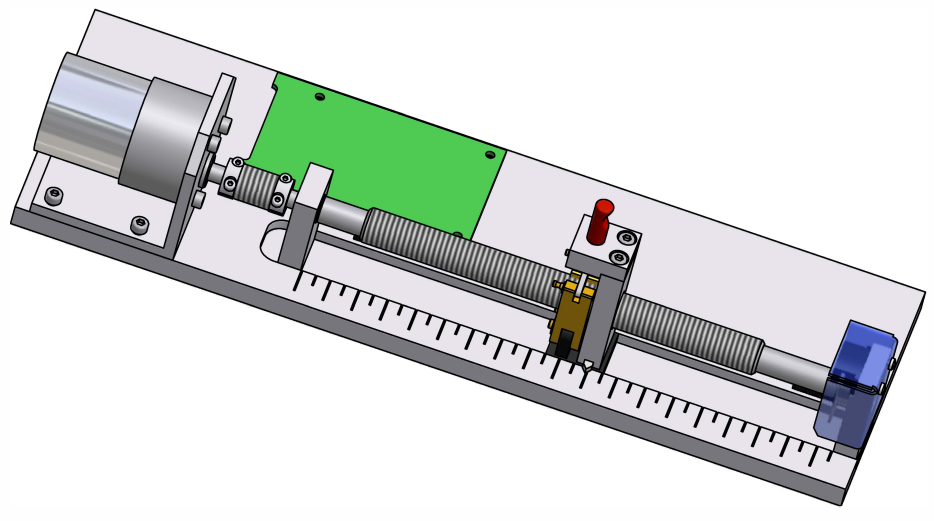


Abbildung 6: Cursor Schlittenaufbau

3.1 Schlitten

Der Schlittenaufbau beinhaltet den Gleichstrommotor, die zwei Reed-Relais sowie den Schlitten mitsamt Schraube. Das Gewinde der Schraube hat eine Grösse von M12x1.75 dies bedeutet das pro Umdrehung eine Distanz von $1.75mm$ zurückgelegt wird, siehe Abbildung 7.

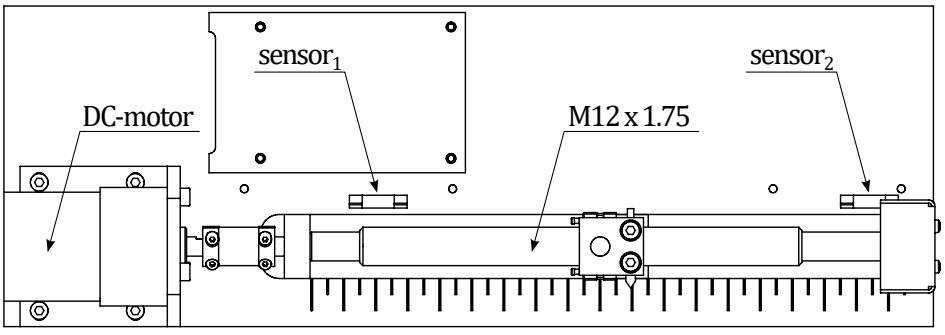


Abbildung 7: Cursor Schlittenaufbau Detail

3.2 Motorsteuerungsschaltung

Der Gleichstrommotor des Wagens wird mit 12V versorgt. Die Stromversorgungsplatine besitzt eine H-Brücke, die durch digitale Signale gesteuert wird. Auf der Stromversorgungsplatine erzeugt ein 5V-Regler die richtige Spannung für die Versorgung der FPGA-Platine [8].

3.2.1 Gleichstrommotor

Der Gleichstrommotor wird von einem L6207 H-Brücken-Treiber [14] gesteuert, siehe Abbildung 8. Die Maximale Schaltfrequenz der H-Brücke beträgt $100kHz$. Dies sollte bei der Erstellung des **P**ulse **W**idth **M**odulation (PWM) Signales in Betracht gezogen werden.

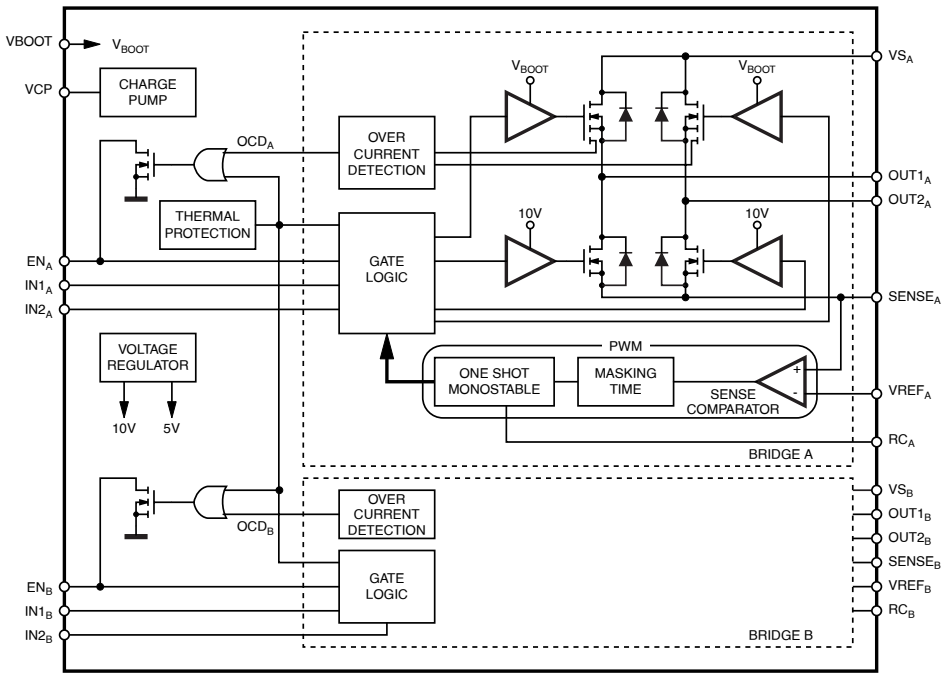


Abbildung 8: H-Brücke L6207N Schaltung [14]

Um die Geschwindigkeit des Gleichstrommotors zu regeln sollte an den Signalen $side_1$ oder $side_2$ ein PWM signal angelegt werden, wobei die maximale Frequenz $100kHz$ betragen darf. Je länger das Signal auf '1' gelegt wird, desto schneller dreht der Motor. In der Abbildung 9 dreht der Motor beim grünen Signal am langsamer und mit dem roten Signal schneller.

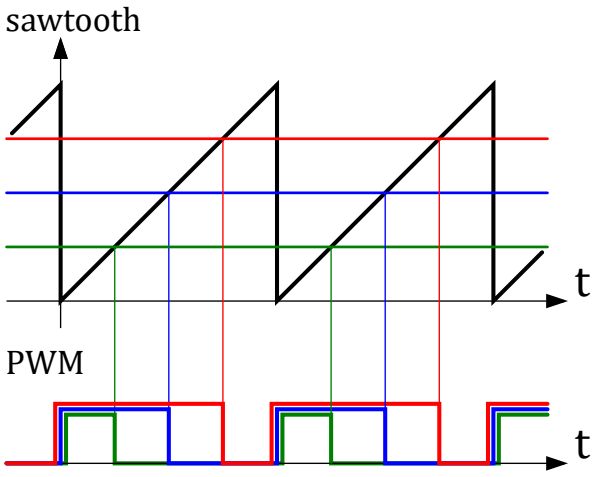
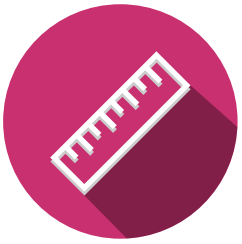


Abbildung 9: PWM signale



3.3 Drehgeber (Encoder)

Der Winkel der Schraube kann mit Hilfe eines inkrementalen Drehgebers gemessen werden. Das in der Baugruppe verwendete Modell ist ein AEDB-9140-A12 [1] (Abbildung 11) mit 500 **C**ounts **p**er **R**evolution (CPR) (Zählungen pro Umdrehung) pro Kanal, dargestellt in Abbildung 10.

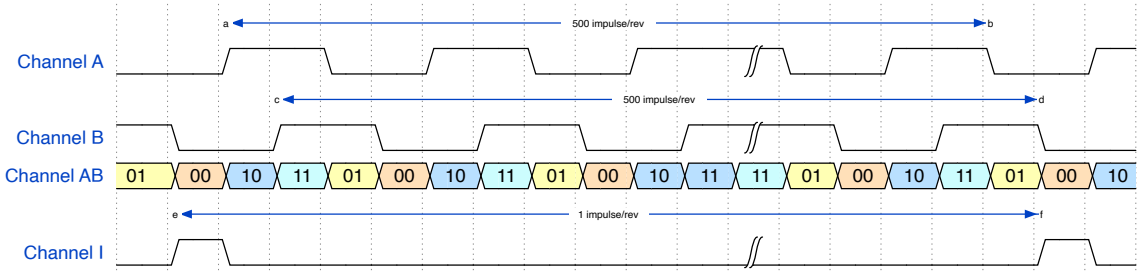


Abbildung 10: Signale inkrementaler Drehgeber

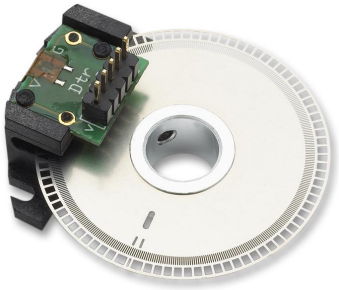


Abbildung 11: Encoder AEDB-9140-A12 [1]

3.4 Reed-Relais

Das Reed relais ist ein Schalter, der mithilfe von elektromagneten geschaltet werden kann [9]. Wenn sich ein Magnet in der Nähe des Sensors befindet, schliesst sich der Kontakt, siehe Abbildung 12. Auf der Baugruppe werden 2 Reed relais verwendet ($sensor_1$ und $sensor_2$, um die linke und rechte Grenze des Wagens zu identifizieren.

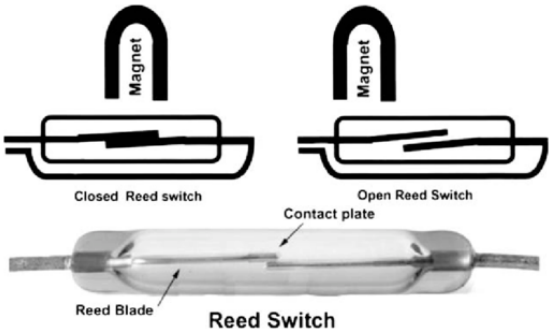
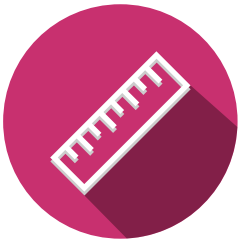


Abbildung 12: Reed Schalter [7]



3.5 FPGA-Platine

Die Hauptplatine ist die FPGA-EBS 2 Laborentwicklungsplatine der Schule [11]. Diese beherbergt eine Xilinx Spartan xc3s500e FPGA [**Spartan3FPGAFamily**] [15] und verfügt über viele verschiedene Schnittstellen (**U**niversal **A**synchronous **R**eceiver **T**ransmitter (UART), **U**niversal **S**erial **B**us (USB), Ethernet, etc.). Der benutzte Oszillator erstellt ein Taktsignal ($clock$) mit einer Frequenz von $f_{clk} = 66MHz$ [4].

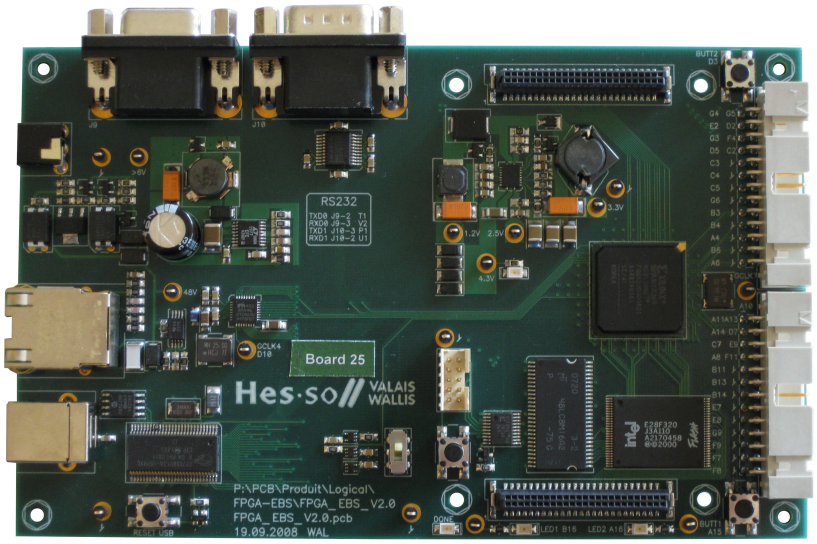


Abbildung 13: FPGA Platine [11]

Auf der EBS3-Karte erzeugt der verwendete Oszillator ein Taktsignal ($clock$) mit einer Frequenz von $f_{clk} = 100MHz$, die durch PLL auf $f_{clk} = 60MHz$ reduziert wird.

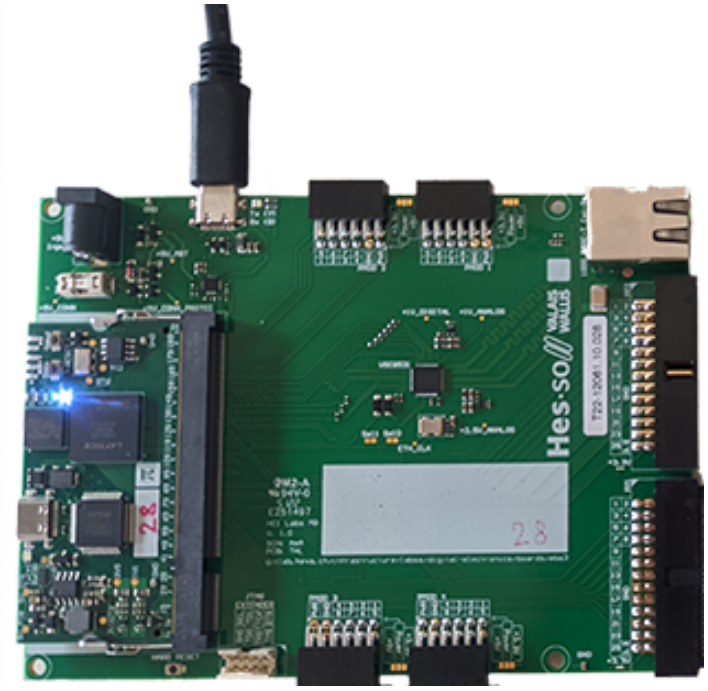
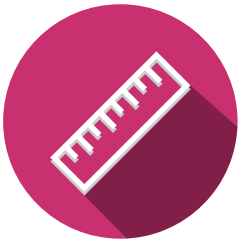


Abbildung 14: EBS3 FPGA Platine [2]



Die Simulators sind standardmäßig für die EBS3 boards eingestellt. Um sie zu ändern, öffnen Sie einen Block von testbench **xxx_tb** und doppelklicken Sie auf die **Pre-User**-Deklarationen (oben links auf der Seite), um die Variable **clockFrequency** auf den gewünschten clock-Wert zu ändern.

3.6 Knöpfe und LEDs

Die Platine mit den Knöpfen und LEDs [12] wird an die FPGA Platine angeschlossen. Sie hat 4 Tasten und 8 LEDs, die im Design verwendet werden können. Falls gewünscht kann diese Platine mit einer LCD Anzeige ausgestattet werden [13] [5].

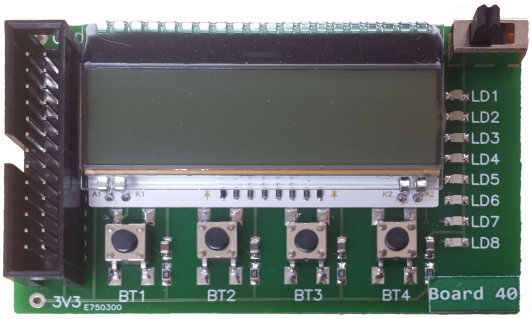
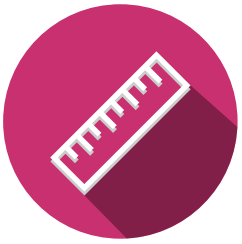


Abbildung 15: Knöpfe-LED-LCD Platine [12]



4 Bewertung

Im Ordner *doc/* zeigt die Datei evaluation-bewertung-cursor.pdf das detaillierte Bewertungsschema, Tabelle 1.

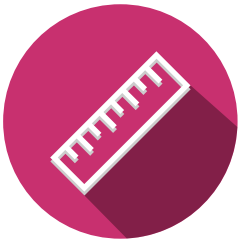
Die Schlussnote beinhaltet den Bericht, den Code sowie eine Präsentation eurerseits des Systems.

Evaluierte Aspekte	Punkte
Bericht	**55**
Einleitung	3
Spezifikation	5
Entwurf	20
Verifizierung und Validation	10
Integration	9
Schlussfolgerung	3
Formale Aspekte des Berichtes	5
Funktionalität der Schaltung	**30**
Qualität der Lösung	**10**
Präsentation	**10**
Total	**105**

Tabelle 1: Bewertungsraster

Das Bewertungsraster gibt bereits Hinweise über die Struktur des Berichtes. Für einen guten Bericht konsultieren Sie das Dokument “Wie verfasst man einen Projektbericht?” [3]

.



5 Erste Schritte

Um mit dem Projekt zu beginnen, kann folgendermassen vorgehen werden:

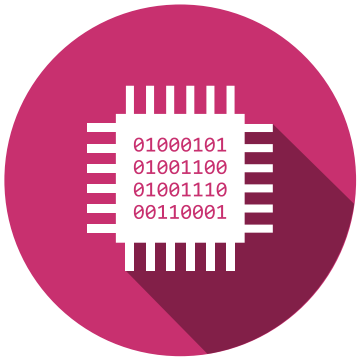
- Lest die obigen Spezifikationen und Informationen genau durch.
- Schaut euch die Hardware und testet das vorprogrammierte Programm.
- Stöbert durch die Dokumente im Ordner *doc/* eures Projektes.
- Entwickelt ein detailliertes Blockdiagramm. Die Signale und deren Funktionen solltet Ihr erklären können.
- Implementierung und Simulation der verschieden Blöcken.
- Testen der Lösung auf der Platine und finden etwaiger Fehler .

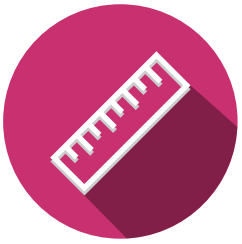
5.1 Tips

Anbei noch einige zusätzlichen Tips um Probleme und Zeitverlust zu vermeiden:

- Teilt das Problem in verschiedene Blöcke auf, benutzt hierzu das leere Toplevel Dokument (cursor-toplevel-empty.pdf). Es ist ein ausgeglichener Mix zwischen Anzahl Komponenten und Komponentengrösse empfohlen.
- Analysiert die verschiedenen Ein- sowie Ausgangssignale, hierzu sollten teilweise die Datenblätter zu Hilfe genommen werden.
- Beachtet bei der Erstellung des Systems das DiD Kapitel “Methodologie für die Entwicklung von digitalen Schaltungen (MET)” [6]
- Es wird empfohlen das System in zwei Schritten zu realisieren.
  - Zuerst ein System das auf die Knöpfe reagiert und den Wagen an die entsprechende Position bringt (immer mit maximaler Geschwindigkeit).
  - Integration der Beschleunigungsphasen ins bestehende System

Vergesst nicht Spass zu haben .





Literatur

[1] Agilent Technologies. *Datasheet Agilent AEDB-9140 Series Three Channel Optical Incremental Encoder Modules with Codewheel, 100 CPR to 500 CPR*. 2005.

[2] Amand Axel. *Schematic: FPGA-EBS3 v1.0*. 2023.

[3] Christophe Bianchi, François Corthay und Silvan Zahno. *Wie Verfasst Man Einen Projektbericht?* 2021.

[4] CTS. *Datasheet CTS Model CB3 & CB3LV HCMOS/TTL Clock Oscillator*. 2006.

[5] Electronic Assembly. *Datasheet: DOGM Graphics Series 132x32 Dots*. 2005.

[6] François Corthay, Silvan Zahno und Christophe Bianchi. *Methodologie Für Die Entwicklung von Digitalen Schaltungen*. 2021.

[7] *Magnetic-Reed-Switch-Above-Closed-and-open-reed-switch-in-response-to-magnet-placement.Png (850×345)*. URL: https://www.researchgate.net/profile/Sidakpal-Panaich-2/publication/51169357/figure/fig1/AS:394204346896388@1470997048549/Magnetic-reed-switch-Above-Closed-and-open-reed-switch-in-response-to-magnet-placement.png (besucht am 24. 11. 2021).

[8] Olivier Walpen. *Schematic: Cursor Chariot Power Circuit*. 2009.

[9] *Reed Relay*. In: *Wikipedia*. 5. Dez. 2020. URL: https://en.wikipedia.org/w/index.php?title=Reed_relay&oldid=992433034 (besucht am 24. 11. 2021).

[10] *Rotary Encoder*. In: *Wikipedia*. 23. Aug. 2021. URL: https://en.wikipedia.org/w/index.php?title=Rotary_encoder&oldid=1040238329 (besucht am 20. 11. 2021).

[11] Silvan Zahno. *Schematic: FPGA-EBS v2.2*. 2014.

[12] Silvan Zahno. *Schematic: Parallelport HEB LCD V2*. 2014.

[13] Sitronix. *Datasheet Sitronix ST7565R 65x1232 Dot Matrix LCD Controller/Driver*. 2006.

[14] STMicroelectronics. *Datasheet: DMOS Dual Full Bridge Driver with PWM Current Controller*. 2003.

[15] Xilinx. *Datasheet Spartan-3E FPGA Family*. 2008.

Akronyme

CPR **C**ounts **p**er **R**evolution. 9

FPGA **F**ield **P**rogrammable **G**ates **A**rray. 4, 7, 10, 11

LCD **L**iquid **C**rystal **D**isplay. 2, 11

LED **L**ight **E**mitting **D**iodes. 4, 7, 11

PCB **P**rinted **C**ircuit **B**oard. 7

PWM **P**ulse **W**idth **M**odulation. 4, 5, 8

UART **U**niversal **A**synchronous **R**eceiver **T**ransmitter. 10

USB **U**niversal **S**erial **B**us. 10