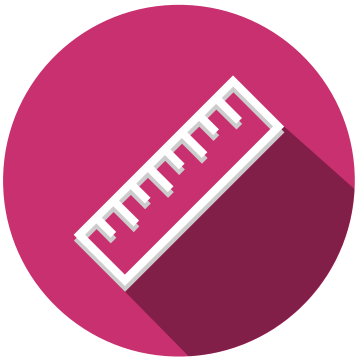


Curseur - commande de moteur

Projet Conception numérique

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

Orientation : Systèmes industriels (SYND)
Cours : Conception numérique (Cnum)
Auteur : Christophe Bianchi, François Corthay, Silvan Zahno, Axel Amand
Date : 14 mars 2023
Version : v2.0

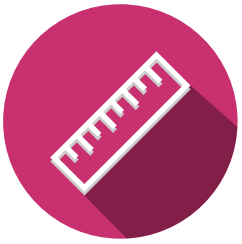
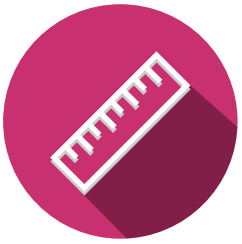


Table des matières



1 Introduction

Le but du projet est d'appliquer directement les connaissances acquises à un exemple pratique en fin de semestre. Il s'agit de piloter un moteur à courant continu, pour déplacer précisément un chariot sur une vis, vers des points prédéfinis. Ce système de curseur peut être observé dans le figure 1.

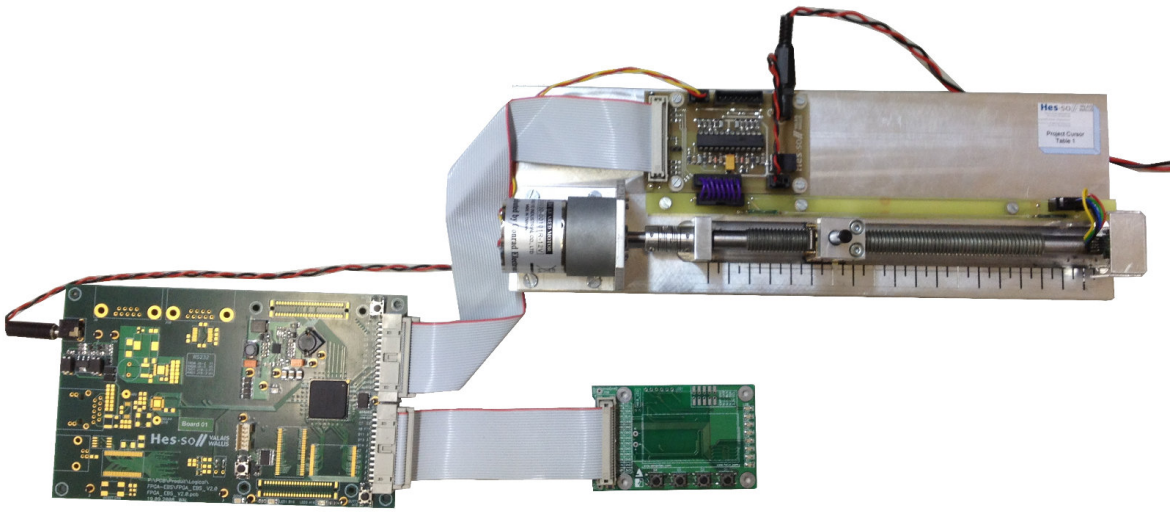
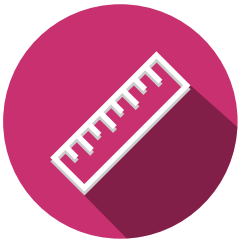


FIGURE 1 – Équipement du curseur

Le but est de réaliser la Spécification minimale définie au (chapitre 2). Les étudiants peuvent, en option, ajouter des fonctions supplémentaires. Il n'y a pas de limites aux idées, par exemple l'écran LCD peut être utilisé pour afficher certaines informations.

Les fonctions supplémentaires permettent d'obtenir quelques points supplémentaires

.



2 Spécification

2.1 Fonctions

Les fonctions de base sont définies comme suit :

- Lorsque la touche *restart* est appuyée, le curseur se déplace vers la position de départ indiquée par un Reed-Relais situé à proximité du Moteur à courant continu.
- Lorsque l'on appuie sur la touche $Position_1$, le curseur doit d'abord accélérer régulièrement vers la position 1 (p_1), puis avancer à pleine vitesse et enfin ralentir régulièrement pour s'arrêter à la position 1 (p_1). Cela peut se faire à partir de la position de départ ou de la position 2, voir figure 2.
- Lorsque l'on appuie sur la touche $Position_2$, le curseur doit d'abord accélérer régulièrement vers la position 2 (p_2), puis avancer à pleine vitesse et enfin ralentir régulièrement pour s'arrêter à la position 2 (p_2), voir figure 2.

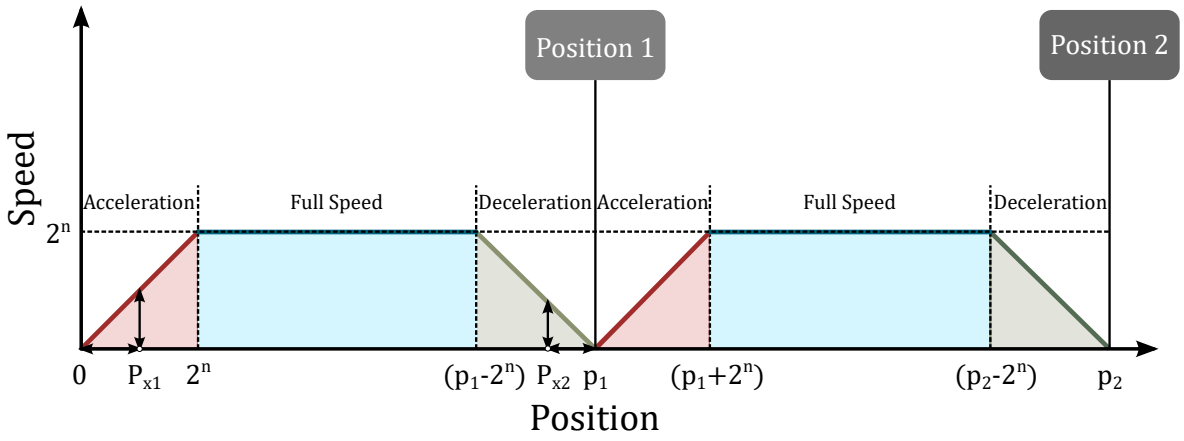


FIGURE 2 – Diagramme de la vitesse du chariot

Les rampes d'accélération et de décélération sont fonctions de la position et non du temps. Il serait en effet très difficile de savoir quand il faut commencer à ralentir pour atteindre l'une des positions, si la décélération dépendait du temps et non de la position. La pente de la rampe doit être choisie de manière à ce que les distances d'accélération et de décélération soient de l'ordre de $1cm$.

Les positions à atteindre sont :

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

2.2 Circuit

Pour accomplir la tâche, le circuit suivant est donné pour déplacer le chariot.

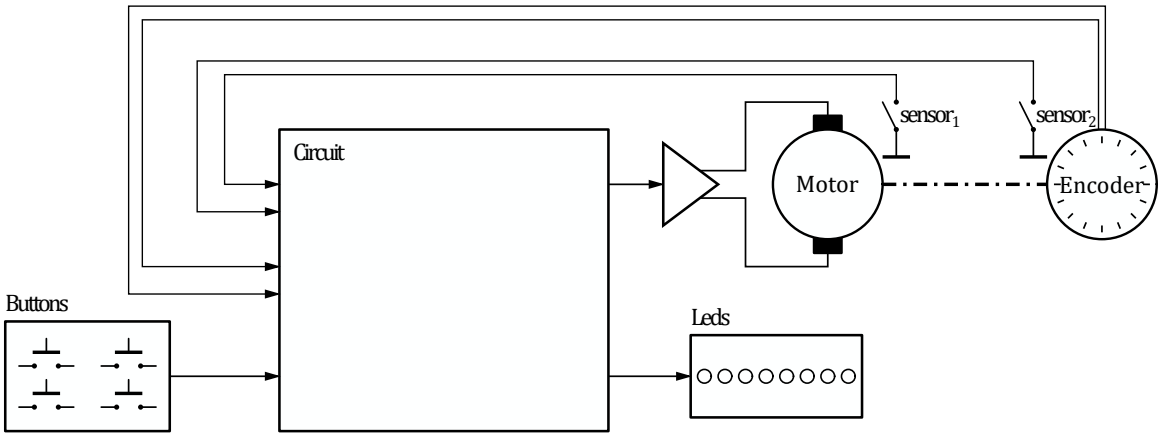


FIGURE 3 – Circuit du curseur

Le circuit fonctionne comme suit :

- Le Moteur à courant continu est commandé par les trois signaux $motor_{On}$, $side_1$, $side_2$. Sa vitesse est contrôlée par une modulation PWM appliquée aux signaux $side_1$ ou $side_2$.
- Deux Reed-Relais sont placés aux extrémités du rail [9]. Ils détectent la présence du chariot du curseur ($sensor_1$ ainsi que $sensor_2$).
- Le Codeur (encodeur) est utilisé pour suivre, respectivement compter, la position du curseur. Ses trois sorties, $encoder_A$, $encoder_B$ et $encoder_I$, permettent de suivre les mouvements de la vis.
- Trois touches sont utilisées pour contrôler le système : $restart$, go_1 et go_2. Une touche supplémentaire, $button_4$, peut être utilisée pour des fonctions optionnelles.
- Les broches $testOut$ peuvent être utilisées pour sortir des informations supplémentaires du système, par exemple pour le débogage ou pour contrôler les LEDs.

Le toplevel vide du design (cursor-toplevel-empty.pdf) montre tous les signaux connectés à la platine **F**ield **P**rogrammable **G**ates **A**rray (FPGA) 4.

FIGURE 4 – Circuit Toplevel vide

2.3 Scénario (exemple)

Dans l'illustration 5, trois scénarios différents sont présentés. Tout d'abord, on appuie sur la touche *restart* et le chariot se déplace à pleine vitesse vers la position initiale ($sensor_1$). Les deux autres scénarios go_2 et go_1 déplacent le chariot vers la *position*2 et la *position*2 respectivement, un signal PWM variable est appliqué aux signaux $side_2$ et $side_1$ pour accélérer et ralentir le chariot.

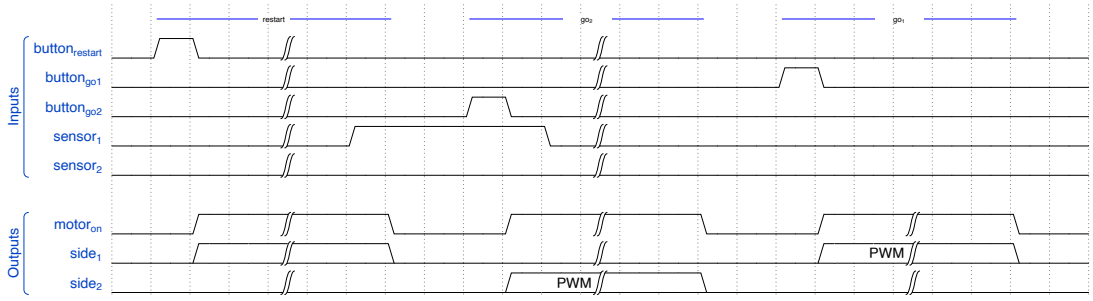
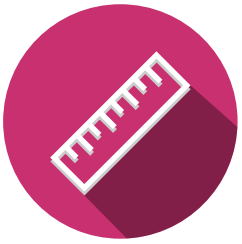


FIGURE 5 – Scenario curseur

Les scénarios ci-dessus sont des exemples, c'est aux étudiants de les compléter

.



2.4 Projet HDL-Designer

Un projet HDL-Designer prédéfini peut être téléchargé ou cloné dans Cyberlearn. La structure de fichier du projet se présente comme suit :

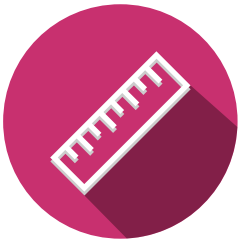
```
did_cursor
+--Board/           # Project and files for programming the FPGA
|   +--concat/      # Complete VHDL file including PIN-UCF file
|   +--ise/         # Xilinx ISE project
+--Cursor/          # Library for the components of the student solution
+--Cursor_test/     # Library for the simulation 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
```

Le chemin d'accès au dossier du projet ne doit pas contenir d'espaces

.

Dans le dossier de projet *doc/*, on peut trouver de nombreuses informations importantes : fiches techniques, évaluation de projet et documents d'aide pour HDL-Designer, pour n'en citer que quelques-unes

.



3 Composants

Le système se compose de 3 platines matérielles différentes, visibles dans la figure 1.

- Un assemblage de chariot avec une carte électronique "**P**rinted **C**ircuit **B**oard (PCB)" qui commande le moteur et lit les capteurs, voir figure 6
- Une carte de développement FPGA, voir figure 14
- Une carte de contrôle à 4 boutons et 8 LEDs, voir figure 15

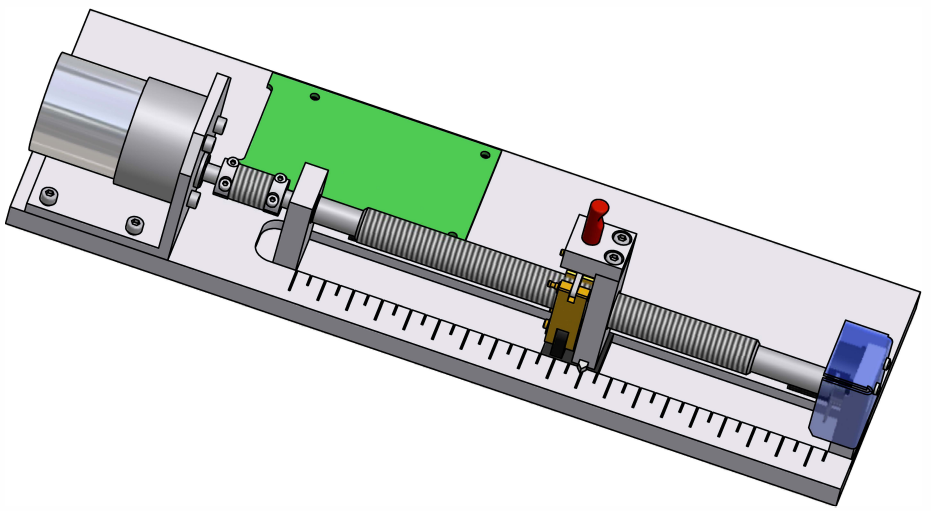


FIGURE 6 – Assemblage de chariot du curseur

3.1 Chariot

La structure du chariot comprend le moteur à courant continu, les deux Reed-Relais ainsi que le chariot et la vis. Le filetage de la vis a une taille de M12x1.75, ce qui signifie qu'une distance de $1.75mm$ est parcourue par tour, voir figure 7.

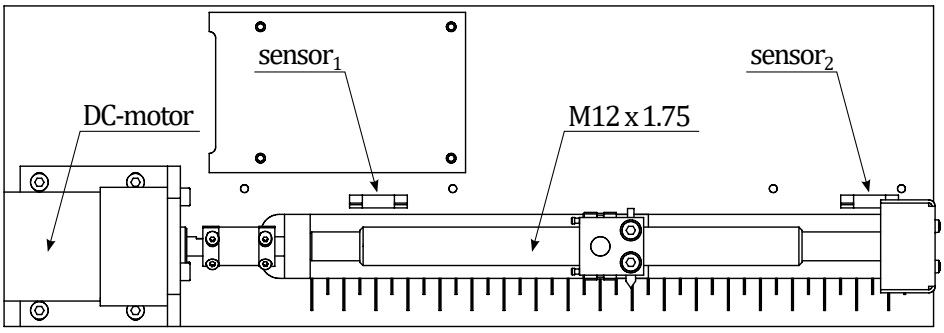


FIGURE 7 – Assemblage détaillé du chariot du curseur

3.2 Circuit de commande de moteur

Le moteur à courant continu du chariot est alimenté en 12V. La carte d'alimentation possède un pont en H qui est commandé par des signaux numériques. Sur la platine d'alimentation, un régulateur 5V génère la tension alimentant la platine FPGA [8].

3.2.1 Moteur à courant continu

Le moteur à courant continu est commandé par un driver de pont en H L6207 [14], voir figure 8. La fréquence de commutation maximale du pont en H est de $100kHz$. Ceci doit être pris en compte lors de la création du signal **P**ulse **W**idth **M**odulation (PWM).

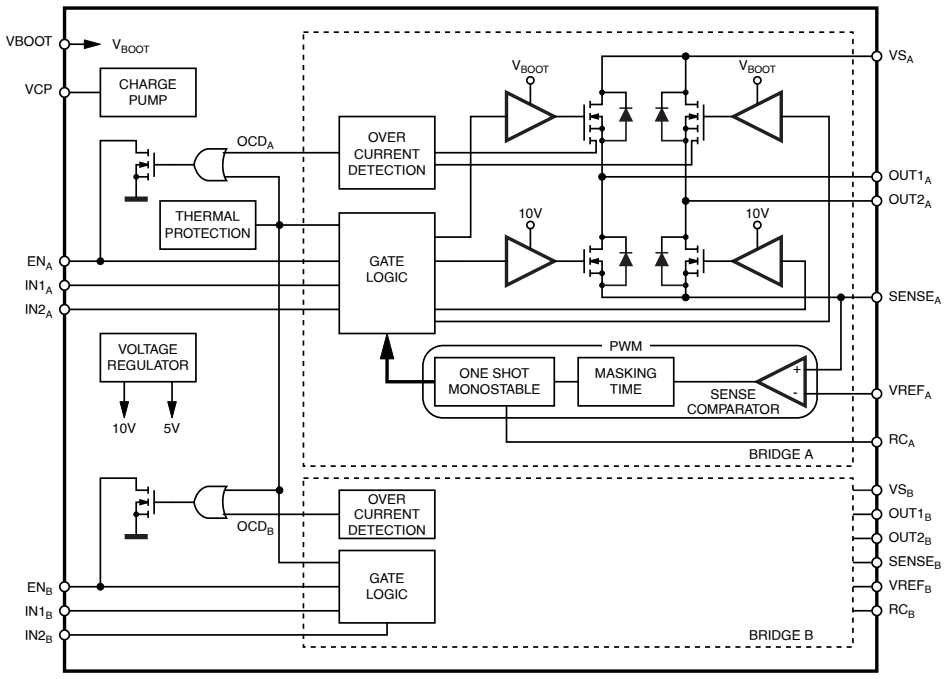


FIGURE 8 – Schéma bloc du circuit du pont-H L6207N [14]

Pour ajuster la vitesse du moteur à courant continu, un signal PWM doit être appliqué aux signaux $side_1$ ou $side_2$, la fréquence maximale étant de $100kHz$. Plus la tension est appliquée longtemps au moteur, plus il tourne vite. Dans la figure 9, le moteur tourne plus lentement avec le signal vert qu'avec le signal bleu et qu'avec le signal rouge.

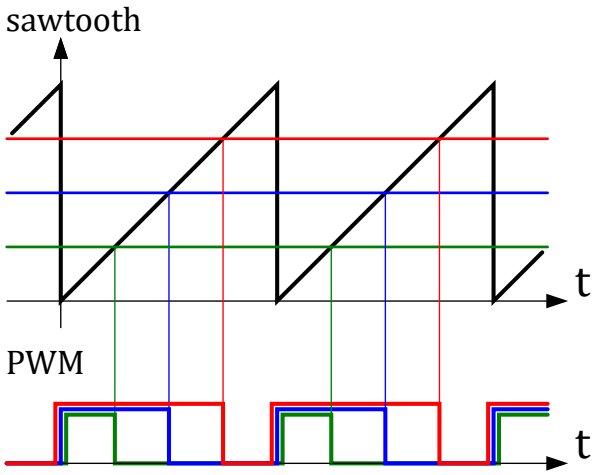


FIGURE 9 – Signaux PWM

3.3 Codeur (encodeur)

L'angle de la vis peut être mesuré à l'aide d'un codeur incrémental. Le modèle utilisé dans l'assemblage est un AEDB-9140-A12 [1] (figure 11) avec 500 **C**ounts **p**er **R**evolution (CPR) (impulsions par tour) par canal, représenté dans la figure 10.

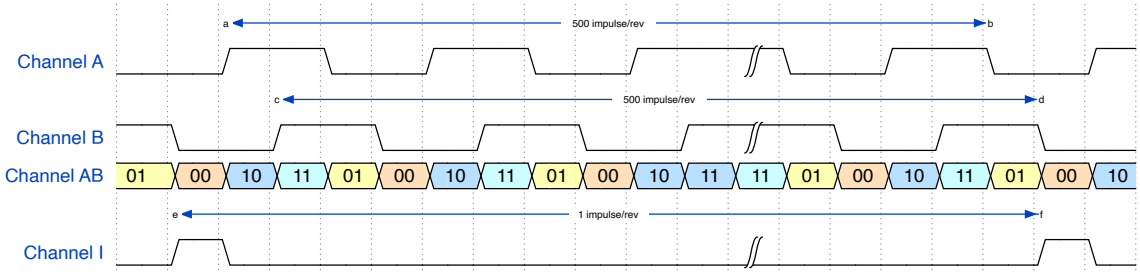


FIGURE 10 – Signaux de codeurs incrémentaux

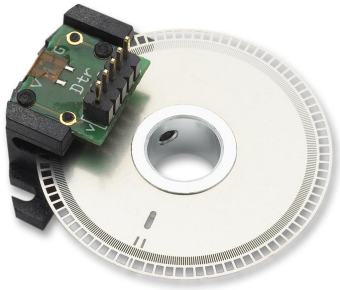


FIGURE 11 – Encodeur AEDB-9140-A12 [1]

3.4 Reed-Relais

Le reed-relais est un interrupteur qui peut être commuté à l'aide d'électro-aimants [9]. Lorsqu'un aimant se trouve à proximité du capteur, le contact se ferme, voir figure 12. Sur le module, 2 reed-relais sont utilisés ($sensor_1$ et $sensor_2$) pour identifier les limites gauche et droite du chariot.

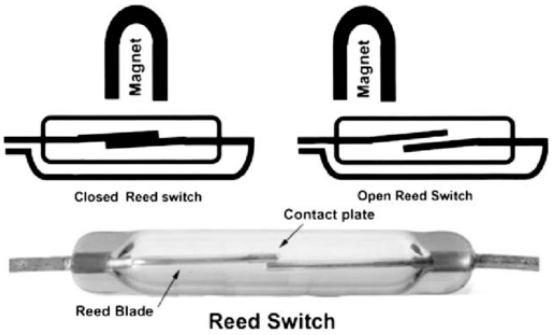


FIGURE 12 – Reed relais [7]

3.5 Carte FPGA

La carte principale est la carte de développement de laboratoire FPGA-EBS 2 de l'école [11]. Elle héberge une puce Xilinx Spartan xc3s500e FPGA [**Spartan3FPGAFamily**] [15] et dispose de nombreuses interfaces différentes (**U**niversal **A**synchronous **R**eceiver **T**ransmitter (UART), **U**niversal **S**erial **B**us (USB), Ethernet, etc.). L'oscillateur utilisé produit un signal d'horloge ($clock$) avec une fréquence de $f_{clk} = 66MHz$ [4].

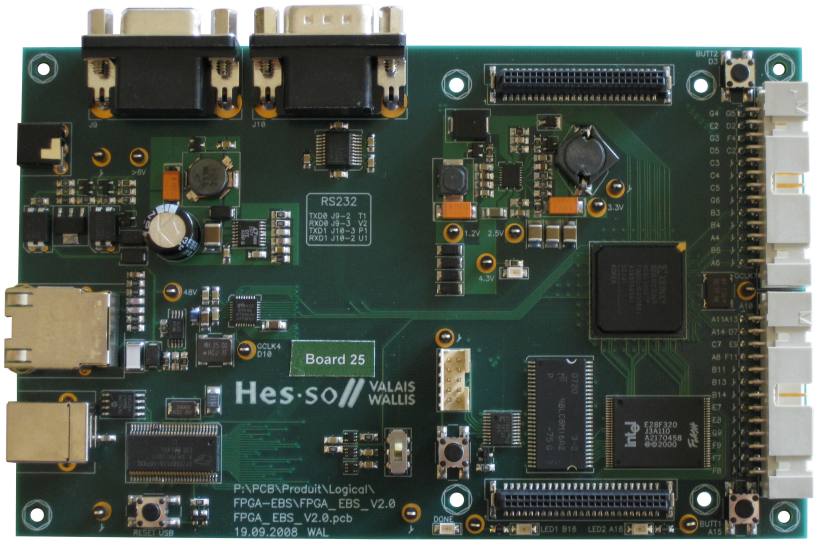


FIGURE 13 – Carte électronique FPGA [11]

Sur la carte EBS3, l'oscillateur utilisé produit un signal d'horloge ($clock$) avec une fréquence de $f_{clk} = 100MHz$, réduit par PLL à $f_{clk} = 60MHz$.

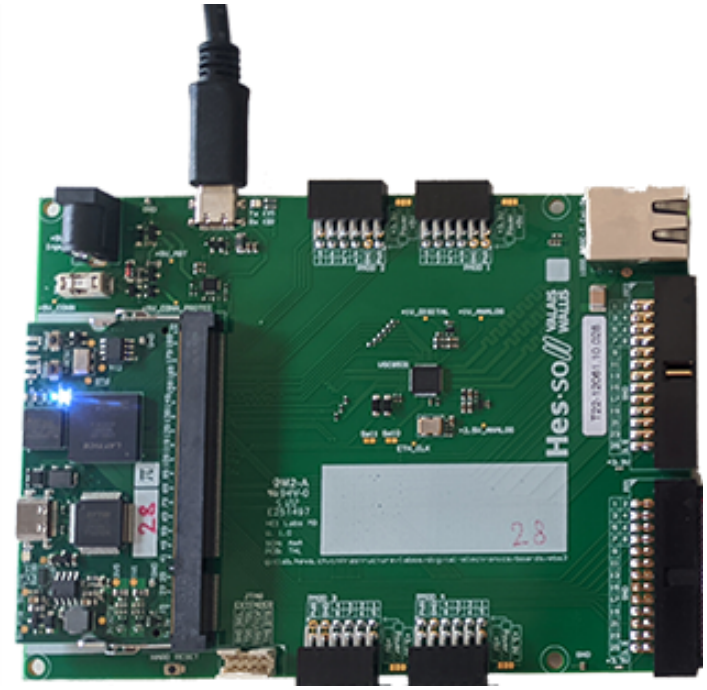
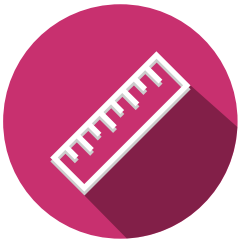


FIGURE 14 – Carte électronique FPGA EBS3 [2]



3.6 Boutons et LEDs

La platine avec les boutons et les LEDs [12] est connectée à la platine FPGA. Elle a 4 boutons et 8 LEDs qui peuvent être utilisés dans le design. Si on le souhaite, cette platine peut être équipée d'un affichage LCD [13] [5]. .

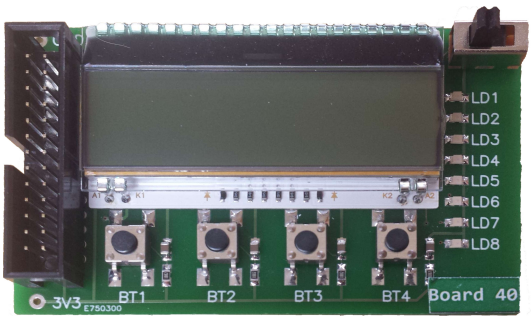
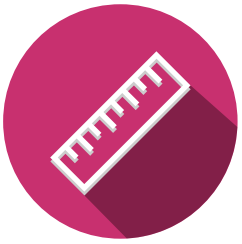


FIGURE 15 – Carte électronique boutons-LED-LCD [12]



4 Evaluation

Dans le dossier *doc/*, le fichier evaluation-bewertung-cursor.pdf montre le schéma d'évaluation détaillé, tableau 1.

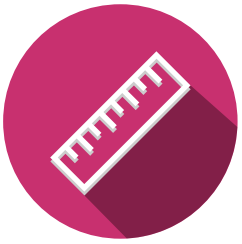
La note finale contient le rapport, le code ainsi qu'une présentation de votre système.

Aspects évalués	points
Rapport	**55**
Introduction	3
Spécification	5
Projet	20
Vérification et validation	10
Intégration	9
Conclusion	3
Aspects formels du rapport	5
Fonctionnalité du circuit	**30**
Qualité de la solution	**10**
Présentation	**10**
Total	**105**

TABLE 1 – Grille d'évaluation

La grille d'évaluation donne des indications sur la structure du rapport. Pour un bon rapport, consultez le document "Comment rédiger un rapport de projet" [3]

.



5 Premières étapes

Pour commencer le projet, on peut procéder de la manière suivante :

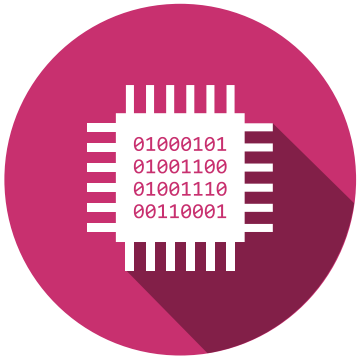
- Lisez attentivement les spécifications et les informations ci-dessus.
- Examinez le matériel et testez le programme préprogrammé.
- Parcourez les documents dans le dossier *doc/* de votre projet.
- Développez un schéma fonctionnel détaillé. Vous devriez pouvoir expliquer les signaux et leurs fonctions.
- Implémentez et simuler les différents blocs.
- Testez la solution sur le circuit imprimé et trouvez les éventuelles erreurs .

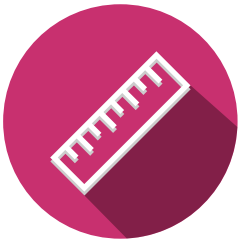
5.1 Tips

Ci-joint quelques conseils supplémentaires pour éviter les problèmes et les pertes de temps :

- Divisez le problème en différents blocs, utilisez pour cela le document Toplevel vide (cursor-toplevel-empty.pdf). Il est recommandé d'avoir un mélange équilibré entre le nombre de composants et la taille/complexité des composants.
- Analyser les différents signaux d'entrée et de sortie, pour cela il est conseillé d'utiliser en partie les fiches techniques.
- Respectez le chapitre DiD "Méthodologie de conception de cicuits numériques (MET)" lors de la création du système. [6].
- Il est recommandé de réaliser le système en deux étapes.
  - D'abord un système qui réagit aux boutons et qui amène le chariot à la position correspondante (toujours à la vitesse maximale).
  - Intégrer les phases d'accélération dans le système existant.

N'oubliez pas de vous amuser .





Références

[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 et Silvan Zahno. *Comment Rédiger Un Rapport de Projet ?* 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 et Christophe Bianchi. *Méthodologie de Conception de Cicuits Numériques*. 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 (visité le 24/11/2021).

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

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

[10] *Rotary Encoder*. In : *Wikipedia*. 23 août 2021. URL : https://en.wikipedia.org/w/index.php?title=Rotary_encoder&oldid=1040238329 (visité le 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.

Acronymes

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