



# Chronometer (Chrono)

## Course Conception Numérique (CNum)



**Orientation:** Systèmes industriels (Synd)  
**Spécialisation:** Infotronics (IT)  
**Cours:** Conception Numérique (CNum)  
**Auteurs:** Silvan Zahno, Axel Amand, François Corthay, Christophe Bianchi  
**Date:** 13.03.2025  
**Version:** v3.0



# Contenu

1	Introduction .....	3
2	Spécification .....	4
2.1	Fonctions .....	4
2.2	Circuit .....	4
2.3	Scénario (exemple) .....	5
2.4	Projet HDL-Designer .....	7
3	Composants .....	8
3.1	Cadran d'horloge .....	8
3.2	Circuit de commande de moteur .....	8
3.2.1	Moteur pas-à-pas .....	8
3.3	Relais Reed .....	10
3.4	Carte FPGA .....	10
3.5	Boutons et LED .....	12
4	Evaluation .....	13
5	Premières étapes .....	14
5.1	Tips .....	14
	Glossaire .....	15



# 1 | Introduction

Le but de ce projet est d'appliquer les connaissances théoriques dans un cadre pratique en fin de semestre. L'objectif est de contrôler un moteur pas à pas pour déplacer une aiguille d'horloge de manière précise sur un cadran, simulant un simple chronomètre. Le système est représenté dans la Figure 1.

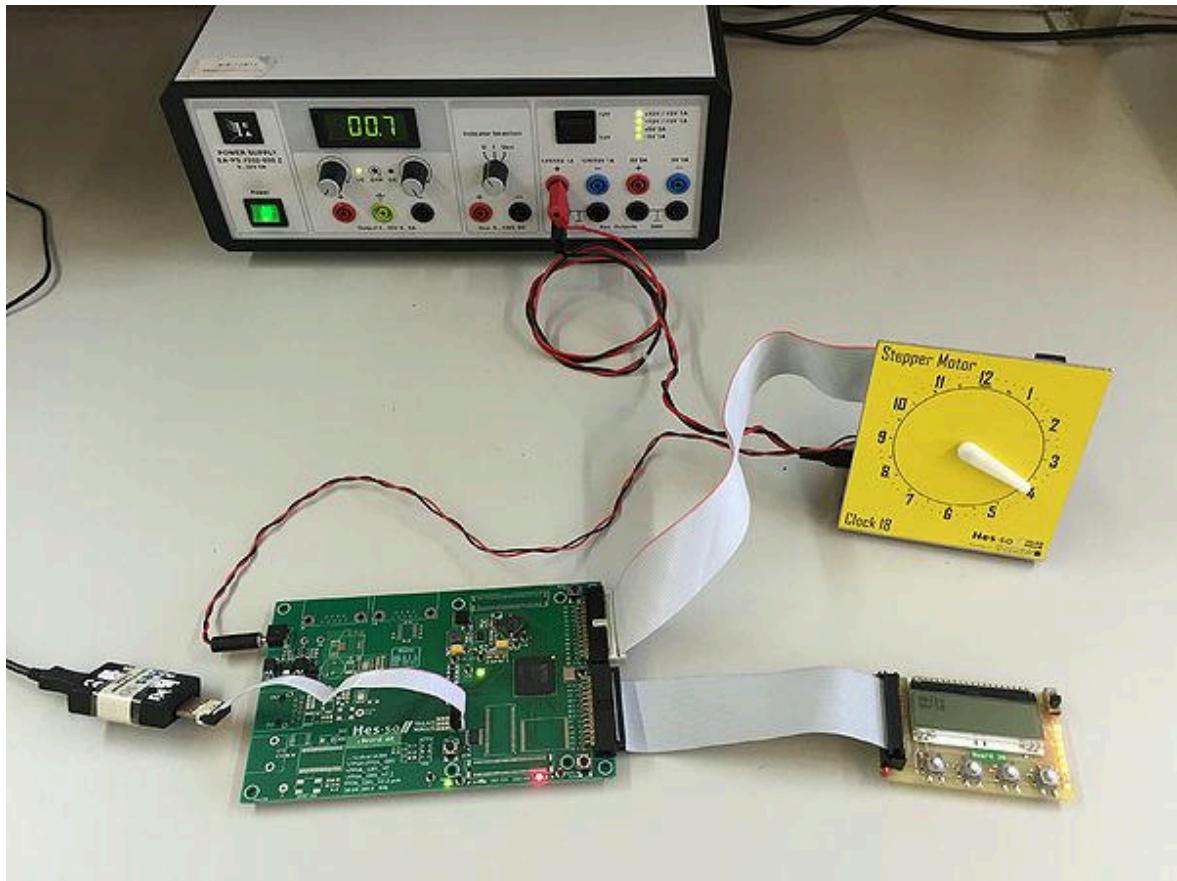


Figure 1 - Équipement du chrono (EBS2)

Les spécifications minimales (voir [Chapitre 2](#)) doivent être respectées, mais les étudiants sont encouragés à implémenter des fonctionnalités supplémentaires. Par exemple, un écran [Liquid Crystal Display \(LCD\)](#) peut être utilisé pour afficher diverses informations.



L'implémentation de fonctionnalités supplémentaires permettra d'obtenir des points supplémentaires dans l'évaluation finale.



## 2 | Spécification

### 2.1 Fonctions

Les fonctions fondamentales du système sont :

- **Restart** : Lorsque le bouton de redémarrage est pressé, l'aiguille revient à la position 12 heures, détectée par un relais reed ([Chapitre 3.3](#)) placé près du moteur pas-à-pas ([Chapitre 3.2.1](#)).
- **Start** : En appuyant sur le bouton de démarrage, l'aiguille se déplace de 1/60e de révolution par seconde.
- **Stop** : En appuyant sur le bouton d'arrêt, l'aiguille s'arrête à sa position actuelle.

Le système de base ne gère pas les entrées utilisateur erratiques, telles que l'appui sur **restart** alors que l'aiguille est déjà à 12 heures.

Le diagramme des fonctionnalités de base est montré dans la [Figure 2](#).

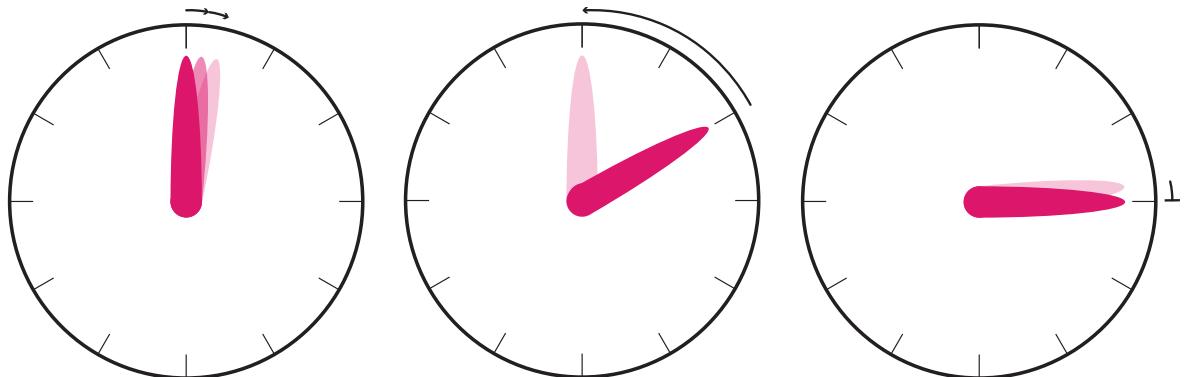


Figure 2 - Diagramme des fonctionnements de base. De gauche à droite: **start**, **restart** et **stop**.

### 2.2 Circuit

Le moteur pas-à-pas contrôle le mouvement de l'aiguille, comme montré dans la [Figure 3](#).

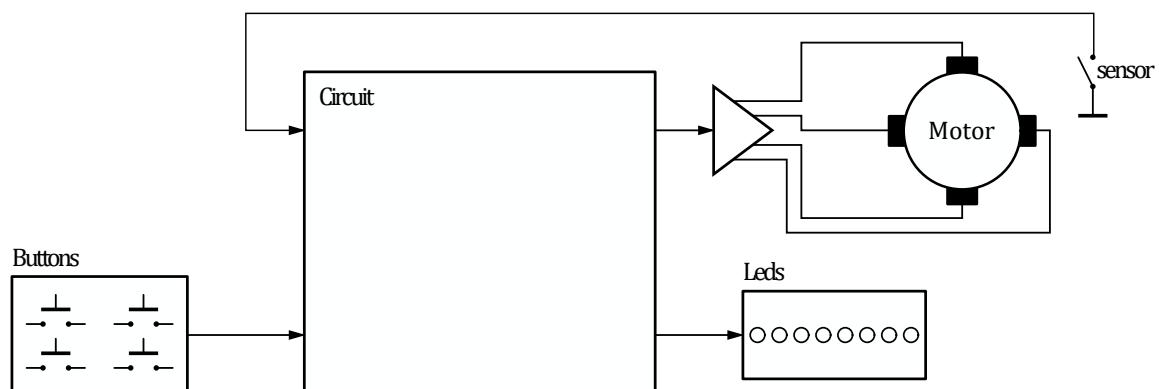


Figure 3 - Circuit du chrono

Le circuit fonctionne de la manière suivante :

- Le moteur pas-à-pas ([Chapitre 3.2.1](#)) est piloté par quatre signaux de contrôle : **coil1**, **coil2**, **coil3** et **coil4**.



- Un relais Reed ([Chapitre 3.3](#)) à la position 12 heures détecte lorsque l'aiguille revient au départ.
- Trois boutons (**restart**, **start**, **stop**) contrôlent le système, avec un quatrième bouton optionnel (**button\_4**) pour des fonctionnalités supplémentaires [1].
- Les broches **TestOut** peuvent être utilisées pour le débogage ou le contrôle de **Light Emitting Diodes (LEDs)**.

La [Figure 4](#) montre la conception de haut niveau du circuit, y compris toutes les connexions à la carte **Field Programmable Gate Array (FPGA)**. Vous pouvez également trouver le fichier dans le dépôt ([chrono-toplevel-empty.pdf](#)).

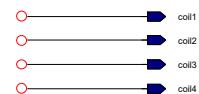
### Buttons



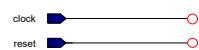
### 12 o'clock Sensor



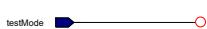
### Stepper Motor Coils



### Clock & Reset



Testmode only for simulation



### Debug Signal (Leds)



Figure 4 - Circuit Toplevel vide

## 2.3 Scénario (exemple)

Dans les [Figure 5](#) et [Figure 6](#), deux scénarios différents sont présentés. Tout d'abord, on appuie sur la touche **restart** et l'aiguille se déplace à pleine vitesse vers la position initiale (**sensor**). Ensuite en appuyant sur la touche **start**, l'aiguille commence à tourner dans le sens horaire. Une fois en mouvement, l'appui sur la touche **stop** arrête l'aiguille sur sa position actuelle. Un appui sur **start** ou **restart** peut alors, respectivement, relancer le chrono ou le remettre à zéro.

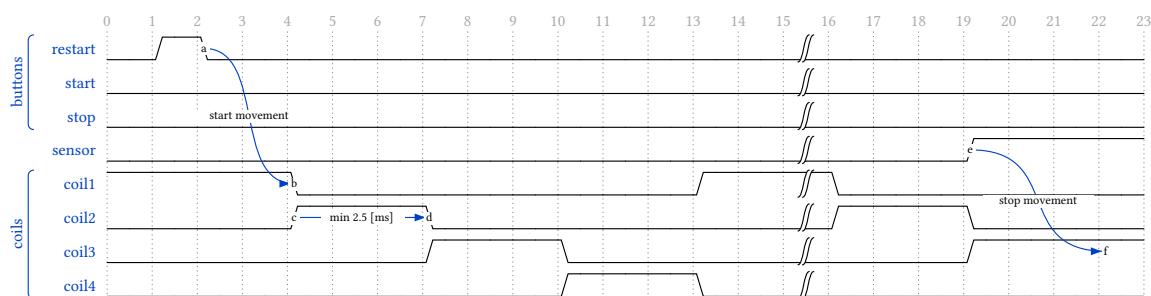


Figure 5 - Scénario chrono - Restart

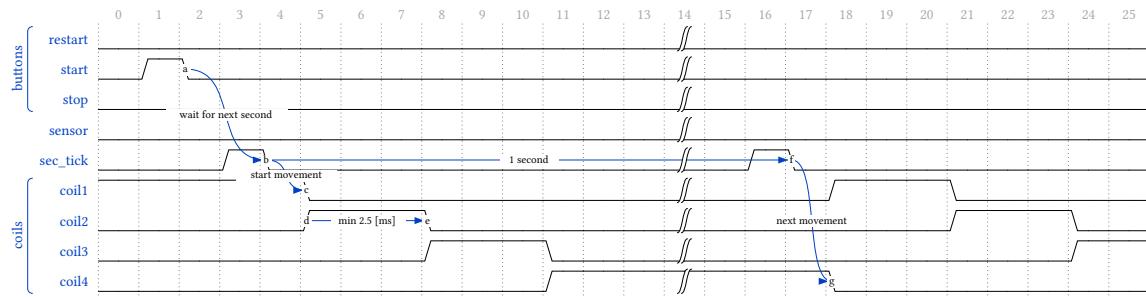


Figure 6 - Scénario chrono - Start



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](#) ou [Github](#). La structure de fichier du projet se présente comme suit:

```
did_chrono
+-Board/          # Project and files for programming the FPGA
|   +-concat/    # Complete VHDL file including PIN-UCF file
|   +-ise/        # Xilinx ISE project
+-Chrono/         # Library for the components of the student solution
+-Chrono_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 chrono avec une carte électronique « [Printed Circuit Board \(PCB\)](#) » qui commande le moteur et lit le capteur, voir [Figure 7](#)
- Une carte de développement [FPGA](#) ([Figure 12](#) ou [Figure 13](#))
- Une carte de contrôle à 4 boutons et 8 LEDs ([Figure 14](#))

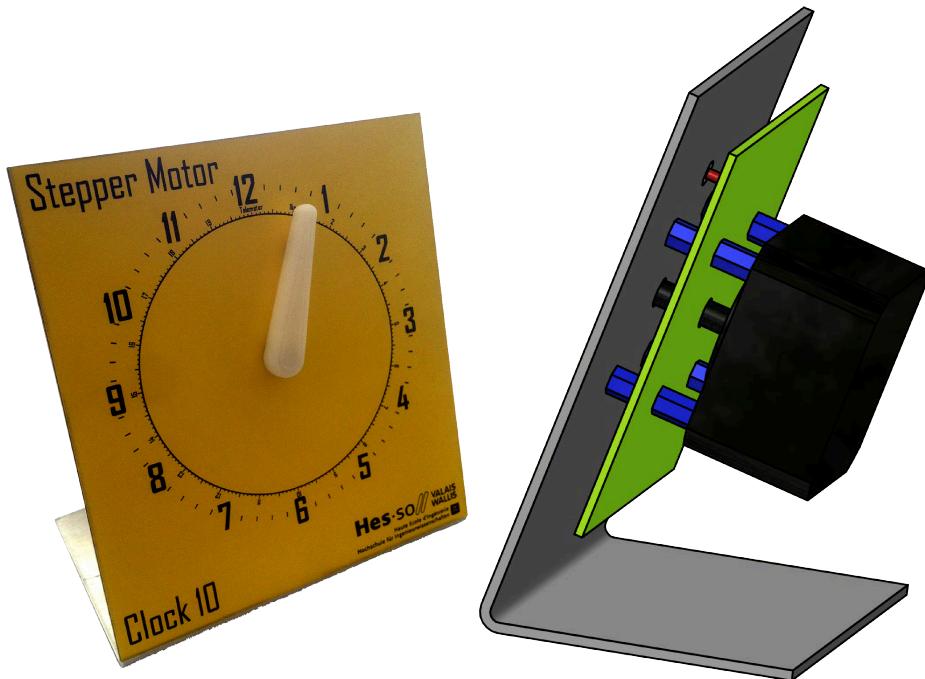


Figure 7 - Assemblage du cadran du chrono

### 3.1 Cadran d'horloge

La structure de l'horloge comprend le moteur pas-à-pas ([Chapitre 3.2.1](#)), le relais reed ([Chapitre 3.3](#)) ainsi que l'aiguille de l'horloge.

### 3.2 Circuit de commande de moteur

Le moteur pas-à-pas du chrono est alimenté en 12V. La carte d'alimentation possède un pont en H commandé par des signaux numériques. Sur la platine d'alimentation, un régulateur 5V génère la tension alimentant la platine [FPGA](#) [2].

#### 3.2.1 Moteur pas-à-pas

Le moteur pas-à-pas possède les caractéristiques suivantes, qui peuvent être lues dans la fiche technique [3]:

- 200 pas par tour
- 9.6V
- 4 phases



Le moteur pas-à-pas est commandé par un driver de pont en H L6207 [4], 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 **Pulse Width Modulation (PWM)**.



L'expérience a montré que le moteur peut atteindre une vitesse de 1 à 2 rotations par seconde.

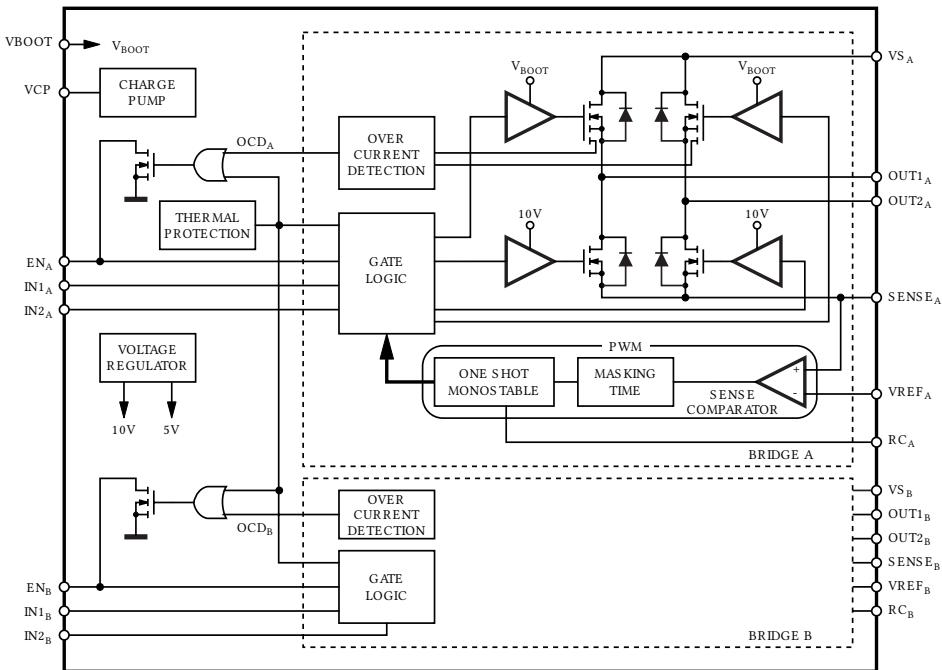


Figure 8 - Schéma bloc du circuit du pont-H L6207N [4]

Dans l'exemple montré à la [Figure 9](#) le bobinage  $p_1$  (contrôlé par le signal **coil1**) est alimenté, orientant ainsi le rotor dans cette direction (indiqué par la flèche noire)

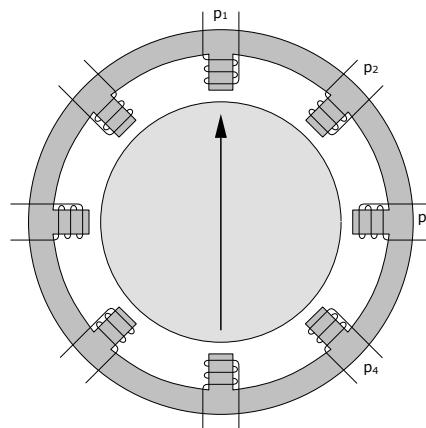


Figure 9 - Schéma de principe du moteur pas-à-pas.



Pour contrôler la position et la vitesse du moteur pas-à-pas, les 4 phases sont utilisées pour créer un champ magnétique. Le rotor peut être modélisé comme un simple aimant, s'alignant ainsi avec la position et la polarité du champ magnétique créé.

En appliquant des impulsions successives sur les signaux **coil1**, **coil2**, **coil3** et **coil4**, un champ tournant peut ainsi être créé et le rotor va suivre cette rotation.

La variation de ces signaux contrôle la position du moteur. La force de maintien et la consommation d'énergie peuvent être contrôlés à l'aide d'un signal **PWM** (tâche optionnelle).



**Danger de brûlure / incendie !** Il est important de ne pas arrêter le moteur en maintenant 1 (ou plus) bobine alimentée en continu ! Le moteur risque de s'échauffer et brûler.

### 3.3 Relais Reed

Le relais Reed est un interrupteur qui peut être commuté à l'aide d'aimants [1], [5]. Lorsqu'un aimant se trouve à proximité du capteur, le contact se ferme, voir [Figure 10](#). Sur le cadran de l'horloge, un relais (**sensor**) est utilisé pour indiquer la position de départ de l'aiguille, à midi. Sa position est indiquée par la couleur bleue sur la figure [Figure 11](#) tandis que l'aimant est indiqué en magenta.

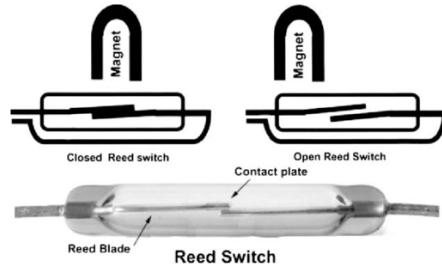


Figure 10 - Reed relais [6]



Figure 11 - Position du [relais reed](#) et de l'[aimant](#).

### 3.4 Carte FPGA

La carte principale est la carte de développement de laboratoire FPGA-EBS 2 de l'école [7]. Elle héberge une puce [Xilinx Spartan xc3s500e FPGA](#) [8], [9] et dispose de nombreuses interfaces différentes ([Universal Asynchronous Receiver Transmitter \(UART\)](#), [Universal Serial Bus \(USB\)](#),



Ethernet, etc.). L'oscillateur utilisé produit un signal d'horloge (**clock**) avec une fréquence de  $f_{\text{clk}} = 66\text{MHz}$  pour la carte EBS2 [10].

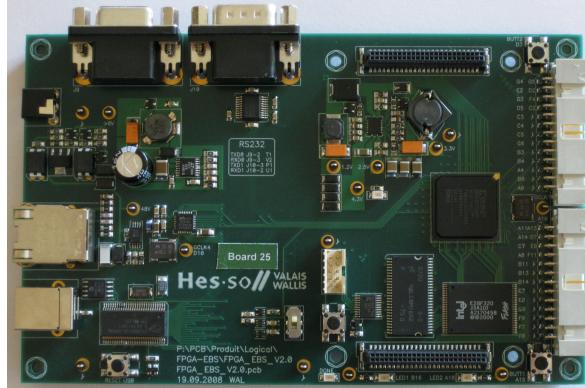


Figure 12 - Carte électronique **FPGA** EBS2 [7]

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

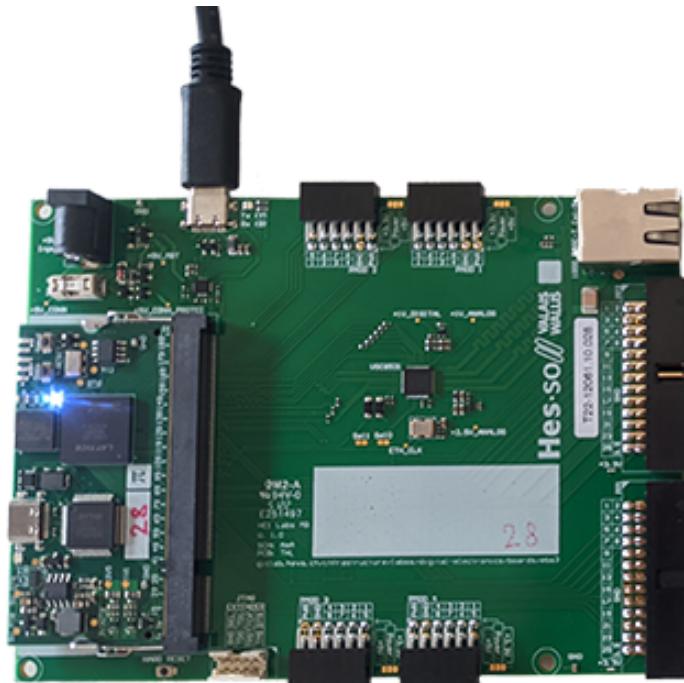


Figure 13 - Carte électronique **FPGA** EBS3 [11]



Les simulateurs sont réglés par défaut pour les boards EBS2. Pour les modifier, ouvrez un bloc de testbench **xxx\_tb** et double-cliquez sur les déclarations **Pre-User** (en haut à gauche de la page) pour modifier la variable **clockFrequency** selon la valeur de clock souhaitée.



### 3.5 Boutons et LED

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], [14].



Figure 14 - Carte électronique boutons-[LEDs](#)-[LCD](#) [12]



## 4 | Evaluation

Dans le dossier **doc/**, le fichier **evaluation-bewertung-chrono.pdf** montre le schéma d'évaluation détaillé, voir [Table 1](#).

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

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

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 » [\[15\]](#).



## 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 simulez 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 (**chrono-toplevel-empty.pdf**). Il est recommandé d'avoir un mélange équilibré entre le nombre de composants et la taille/complexité des composants.
- Analysez 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 circuits numériques (MET) » lors de la création du système [16].
- Il est recommandé de réaliser le système de façon incrémentale, par exemple:
  - Commencer par réagir aux boutons et déplacer l'aiguille
  - Intégrer le déplacement à la seconde et aligné sur le cadran



N'oubliez pas de vous amuser.





# Glossaire

**FPGA** – Field Programmable Gate Array [5](#), [8](#), [10](#), [11](#), [12](#)

**LCD** – Liquid Crystal Display [3](#), [12](#)

**LED** – Light Emitting Diode [5](#), [8](#), [12](#)

**PCB** – Printed Circuit Board [8](#)

**PWM** – Pulse Width Modulation [9](#), [10](#)

**UART** – Universal Asynchronous Receiver Transmitter [10](#)

**USB** – Universal Serial Bus [10](#)



# Bibliographie

- [1] « Reed Relay ». 5 décembre 2020. Consulté le: 24 novembre 2021. [En ligne]. Disponible sur: [https://en.wikipedia.org/w/index.php?title=Reed\\_relay&oldid=992433034](https://en.wikipedia.org/w/index.php?title=Reed_relay&oldid=992433034)
- [2] Silvan Zahno, « Schematic: FPGA-EBS Motor v2.1 ». 2009.
- [3] P. Sartoretti, « Stepper Motor-Module ». 2008.
- [4] STMicroelectronics, « Datasheet: DMOS Dual Full Bridge Driver with PWM Current Controller ». 2003.
- [5] S. Electronics, « Datasheet Reed Sensor ORD213 ». 2001.
- [6] « Magnetic-Reed-Switch-Above-Closed-and-open-reed-switch-in-response-to-magnet-placement.Png (850×345) ». Consulté le: 24 novembre 2021. [En ligne]. Disponible sur: <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>
- [7] Silvan Zahno, « Schematic: FPGA-EBS v2.2 ». 2014.
- [8] Xilinx, « Spartan-3 FPGA Family ». Consulté le: 20 novembre 2021. [En ligne]. Disponible sur: <https://www.xilinx.com/products/silicon-devices/fpga/spartan-3.html>
- [9] Xilinx, « Datasheet Spartan-3E FPGA Family ». 2008.
- [10] CTS, « Datasheet CTS Model CB3 & CB3LV HCMOS/TTL Clock Oscillator ». 2006.
- [11] A. Amand et S. Zahno, « FPGA-EBS3 Electornic Technical Documentation ». 2022.
- [12] Silvan Zahno, « Schematic: Parallelport HEB LCD V2 ». 2014.
- [13] Sitronix, « Datasheet Sitronix ST7565R 65x1232 Dot Matrix LCD Controller/Driver ». 2006.
- [14] Electronic Assembly, « Datasheet: DOGM Graphics Series 132x32 Dots ». 2005.
- [15] Christophe Bianchi, François Corthay, et Silvan Zahno, « Comment Rédiger Un Rapport de Projet? ». 2021.
- [16] François Corthay, Silvan Zahno, et Christophe Bianchi, « Méthodologie de Conception de Circuits Numériques ». 2021.