



# Chrono - commande de moteur

## Projet Conception numérique



Orientation : [Systèmes industriels \(SYND\)](#)

Cours : Conception numérique (Cnum)

Auteur : [Christophe Bianchi](#), [François Corthay](#), [Silvan Zahno](#), [Axel Amand](#)

Date : 13 mars 2023

Version : v2.0



## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Spécifications</b>	<b>3</b>
2.1	Fonctions . . . . .	3
2.2	Circuit . . . . .	3
2.3	Scénario (exemple) . . . . .	4
2.4	Projet HDL-Designer . . . . .	6
<b>3</b>	<b>Composants</b>	<b>7</b>
3.1	Cadran d'horloge . . . . .	7
3.2	Circuit de commande de moteur . . . . .	7
3.2.1	Moteur pas-à-pas . . . . .	7
3.3	Reed-Relais . . . . .	9
3.4	Carte FPGA . . . . .	9
3.5	Boutons et LEDs . . . . .	10
<b>4</b>	<b>Evaluation</b>	<b>12</b>
<b>5</b>	<b>Premières étapes</b>	<b>13</b>
5.1	Tips . . . . .	13
	<b>Références</b>	<b>14</b>
	<b>Acronymes</b>	<b>14</b>



## 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 pas-à-pas, pour déplacer précisément une aiguille sur un cadran d'horloge pour se comporter comme un chronomètre simple. Ce système de chronomètre est montré dans la figure 1

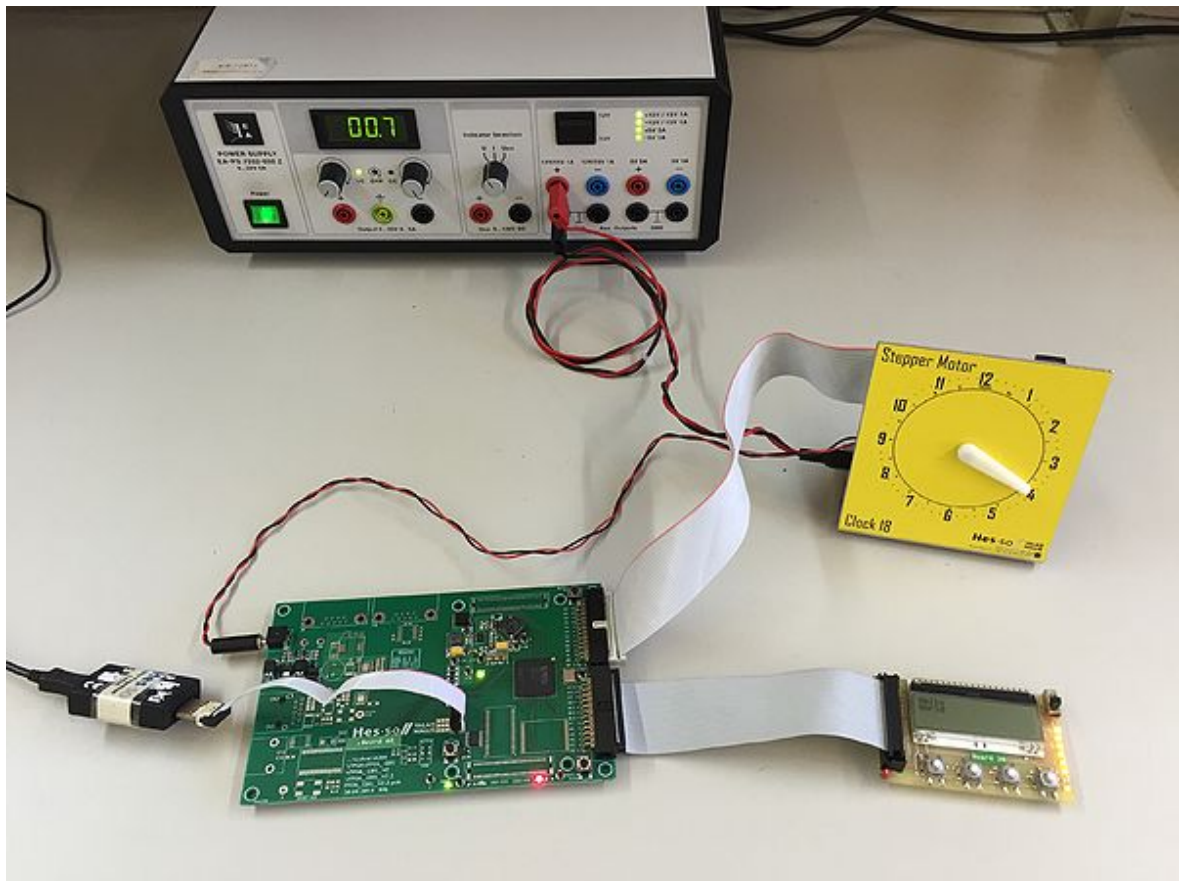


FIGURE 1 – Équipement du chrono

Le but est de réaliser les [Spécifications](#) minimales définies 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 Spécifications

### 2.1 Fonctions

Les fonctions de base sont définies comme suit :

- Lorsque la touche *restart* est appuyée, l'aiguille revient à la position de départ (12h) indiquée par un **Reed-Relais** situé à proximité du **Moteur pas-à-pas**.
- Lorsque la touche *start* est appuyée, l'aiguille se déplace de 1/60ème de tour chaque seconde.
- Lorsque la touche *stop* est appuyée, l'aiguille s'arrête et attend à cette position.

Le système minimal ne gère pas les cas où l'utilisateur agit erratiquement, comme par exemple en appuyant sur *restart* alors que l'aiguille est déjà en position 12h.

La figure 2 schématise ces comportements.

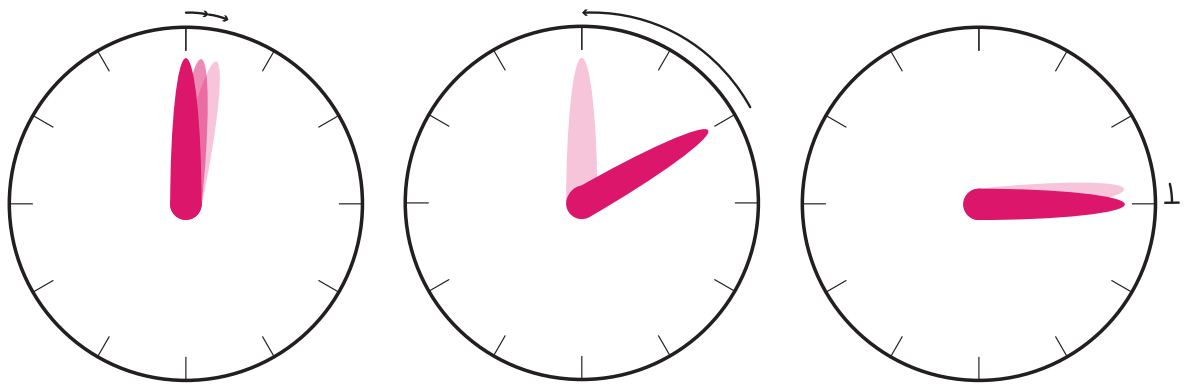


FIGURE 2 – Diagramme des fonctionnements de base. De gauche à droite : *start*, *restart* et *stop*

### 2.2 Circuit

L'aiguille est contrôlée par un moteur pas-à-pas qui prend place dans le circuit montré à la figure 3.

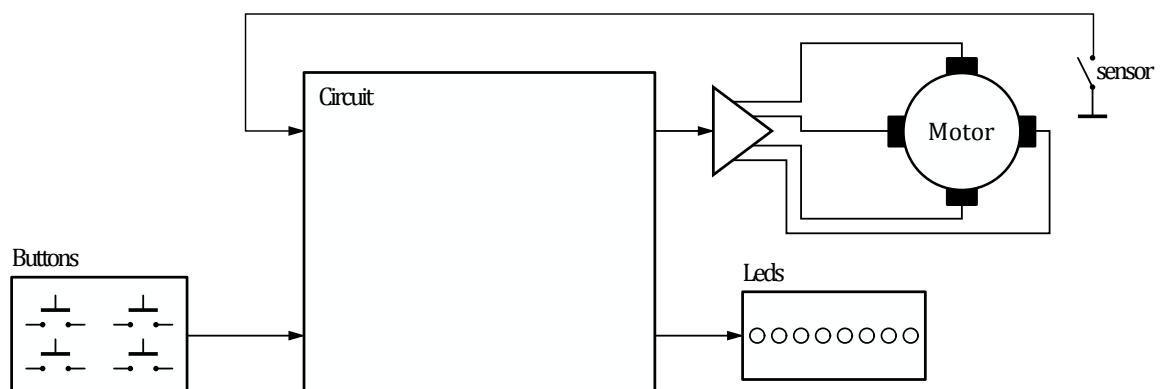




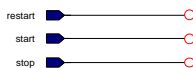
FIGURE 3 – Circuit du chrono

Le circuit fonctionne comme suit :

- Le **Moteur pas-à-pas** est commandé par les quatre signaux *coil1*, *coil2*, *coil3* et *coil4*. Le moteur se contrôle par impulsions successives sur ses 4 bobines.
- Un **Reed-Relais** est placé à la position minuit/midi du cadran de l'horloge [10]. Il détecte la présence de l'aiguille en position de départ (*sensor*).
- Trois touches sont utilisées pour contrôler le système : *restart*, *start* et *stop*. Une touche supplémentaire, *button4*, 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 (chrono-toplevel-empty.pdf) montre tous les signaux connectés à la platine **Field Programmable Gates Array (FPGA)** figure 4.

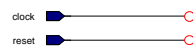
#### Buttons



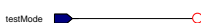
#### 12 o'clock Sensor



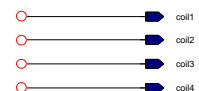
#### Clock & Reset



Testmode only for simulation



#### Stepper Motor Coils



#### Debug Signal (Leds)



FIGURE 4 – Circuit Toplevel vide

## 2.3 Scénario (exemple)

Dans les figures 5 et 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.

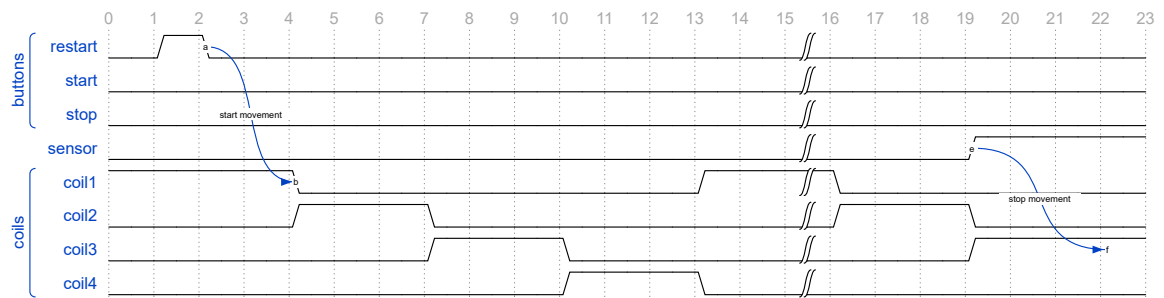


FIGURE 5 – Scénario chrono - Restart

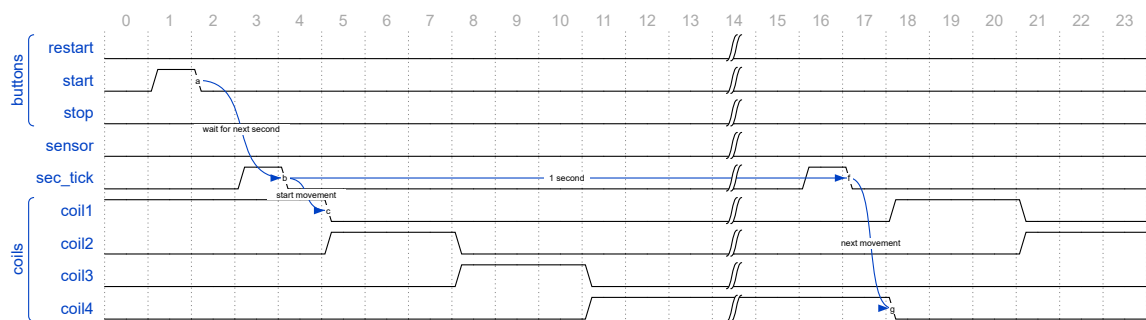


FIGURE 6 – Scénario chrono - Start



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](#). 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**, voir figure 13
- Une carte de contrôle à 4 boutons et 8 **LEDs**, voir figure 14

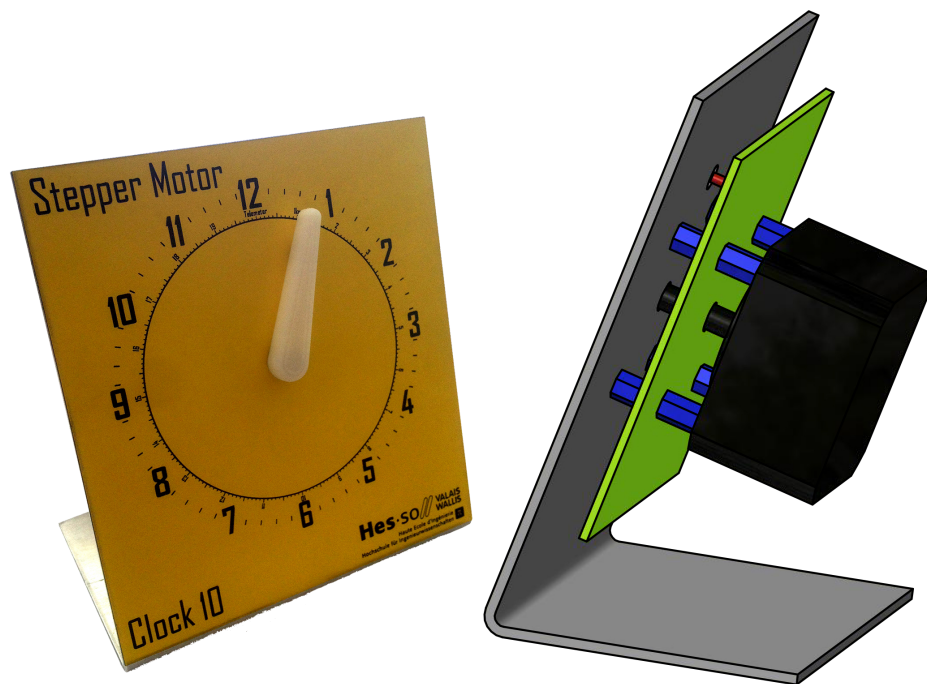


FIGURE 7 – Assemblage du cadran du chrono

### 3.1 Cadran d'horloge

La structure de l'horloge comprend le moteur pas-à-pas, le **Reed-Relais** 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 qui est commandé par des signaux numériques. Sur la platine d'alimentation, un régulateur 5V génère la tension alimentant la platine **FPGA** [13].

#### 3.2.1 Moteur pas-à-pas

Le moteur pas-à-pas possède les caractéristiques suivantes, qui peuvent être lues dans la fiche technique [12] :

- 200 pas par tour
- 8-12V
- 4 phases





Le moteur pas-à-pas est commandé par un driver de pont en H L6207 [17], voir figure 8. La fréquence de commutation maximale du pont en H est de  $100\text{kHz}$ . 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.

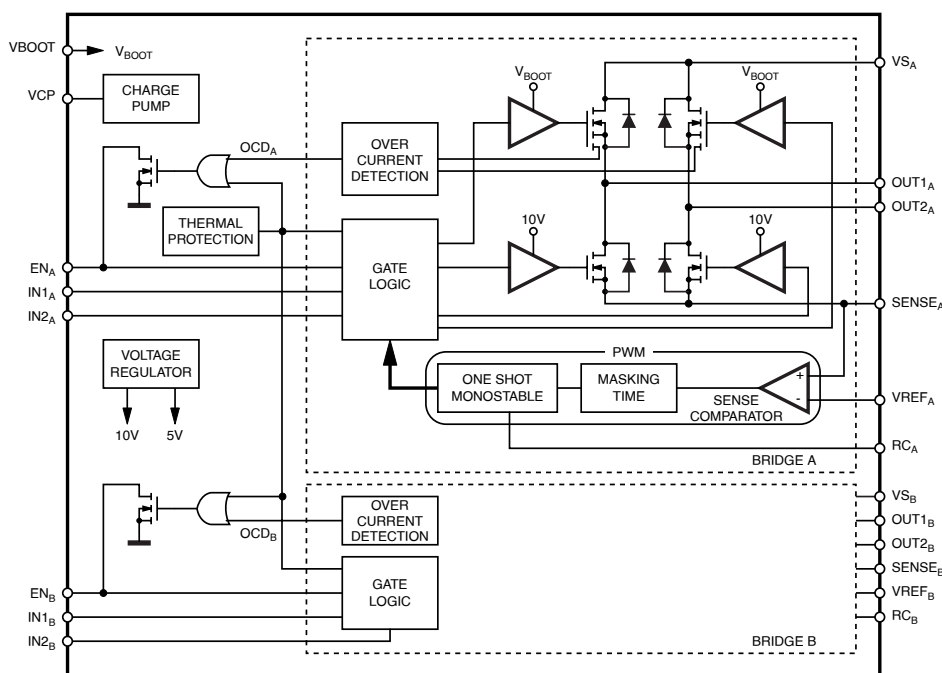


FIGURE 8 – Schéma bloc du circuit du pont-H L6207N [17]

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)

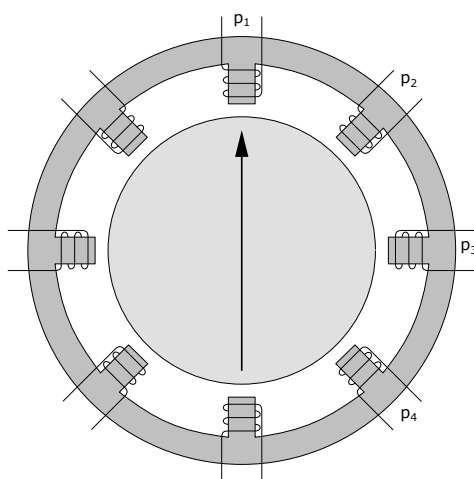


FIGURE 9 – Schéma de principe du moteur pas-à-pas.

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 contrôlées 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 Reed-Relais

Le reed-relais est un interrupteur qui peut être commuté à l'aide d'aimants [10] [6]. 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 11 tandis que l'aimant est indiqué en **magenta**.

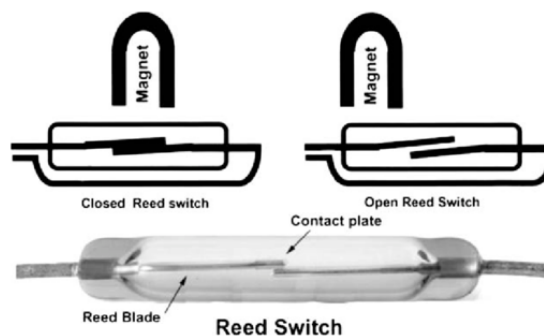


FIGURE 10 – Reed relais [8]

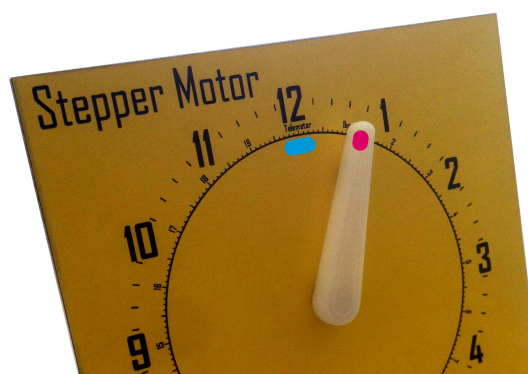


FIGURE 11 – Position du Relais-reed et de l'aimant.

### 3.4 Carte FPGA

La carte principale est la carte de développement de laboratoire FPGA-EBS 2 de l'école [14]. Elle héberge une puce **Xilinx Spartan xc3s500e FPGA [Spartan3FPGAFamily]** [18] 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  $f_{clk} = 66\text{MHz}$  pour la carte EBS2 [4].

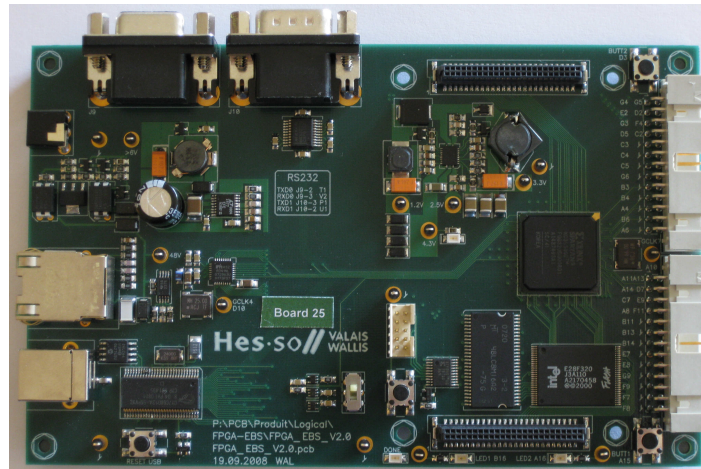


FIGURE 12 – Carte électronique FPGA EBS2 [14]

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

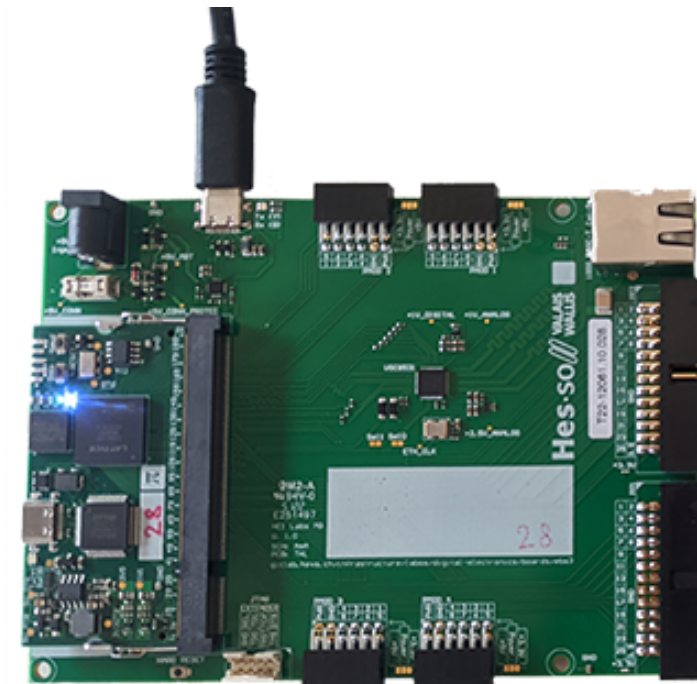


FIGURE 13 – Carte électronique FPGA EBS3 [2]

### 3.5 Boutons et LEDs

La platine avec les boutons et les LEDs [15] 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 [16] [5].

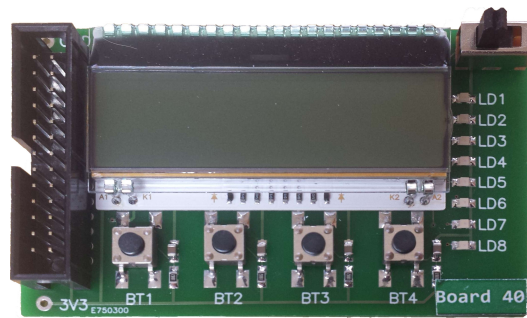


FIGURE 14 – Carte électronique boutons-LED-LCD [15]



## 4 Evaluation

Dans le dossier *doc/*, le fichier *evaluation-bewertung-chrono.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
<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" [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 simulez les différents blocs.
- Testez la solution sur le circuit imprimé et trouvez les éventuelles erreurs 🐛.

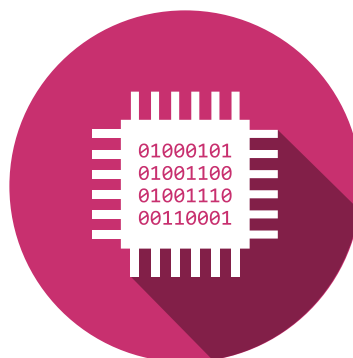
### 5.1 Tips

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 ([chronotoplevel-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. [7].
- 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 😊.





## 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] STANDEx ELECTRONICS. *Datasheet Reed Sensor ORD213*. 2001.
- [7] FRANÇOIS CORTHAY, SILVAN ZAHNO et CHRISTOPHE BIANCHI. *Méthodologie de Conception de Circuits Numériques*. 2021.
- [8] *Magnetic-Reed-Switch-Above-Closed-and-open-reed-switch-in-response-to-magnet-placement.Png (850x345)*. 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).
- [9] OLIVIER WALPEN. *Schematic : Cursor Chariot Power Circuit*. 2009.
- [10] *Reed Relay*. In : *Wikipedia*. 5 déc. 2020. URL : [https://en.wikipedia.org/w/index.php?title=Reed\\_relay&oldid=992433034](https://en.wikipedia.org/w/index.php?title=Reed_relay&oldid=992433034) (visité le 24/11/2021).
- [11] *Rotary Encoder*. In : *Wikipedia*. 23 août 2021. URL : [https://en.wikipedia.org/w/index.php?title=Rotary\\_encoder&oldid=1040238329](https://en.wikipedia.org/w/index.php?title=Rotary_encoder&oldid=1040238329) (visité le 20/11/2021).
- [12] PASCAL SARTORETTI. *Stepper Motor-Module*. 2008.
- [13] SILVAN ZAHNO. *Schematic : FPGA-EBS Motor v2.1*. 2009.
- [14] SILVAN ZAHNO. *Schematic : FPGA-EBS v2.2*. 2014.
- [15] SILVAN ZAHNO. *Schematic : Parallelport HEB LCD V2*. 2014.
- [16] SITRONIX. *Datasheet Sitronix ST7565R 65x1232 Dot Matrix LCD Controller/Driver*. 2006.
- [17] STMICROELECTRONICS. *Datasheet : DMOS Dual Full Bridge Driver with PWM Current Controller*. 2003.
- [18] XILINX. *Datasheet Spartan-3E FPGA Family*. 2008.

## Acronymes

**FPGA** Field Programmable Gates Array. 4, 7, 9, 10

**LCD** Liquid Crystal Display. 2, 10, 11

**LED** Light Emitting Diodes. 4, 7, 10, 11

**PCB** Printed Circuit Board. 7

**PWM** Pulse Width Modulation. 8, 9

**UART** Universal Asynchronous Receiver Transmitter. 9

**USB** Universal Serial Bus. 10