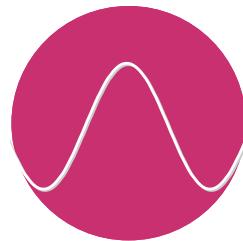


# Inverter - Conversion d'énergie

Projet Conception numérique

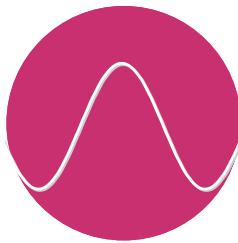


Orientation : Énergie et techniques environnementales (ETE)  
Cours : Conception numérique (Cnum)  
Auteur : Christophe Bianchi, François Corthay, Silvan Zahno, Axel Amand  
Date : 12 avril 2023  
Version : v2.0



## Table des matières

|                   |  |           |
|-------------------|--|-----------|
| <b>1</b>          | <b>Introduction</b>                    | <b>2</b>  |
| <b>2</b>          | <b>Spécification</b>                   | <b>3</b>  |
| 2.1               | Fonctions . . . . .                    | 3         |
| 2.2               | Circuit . . . . .                      | 3         |
| 2.3               | Projet HDL-Designer . . . . .          | 5         |
| <b>3</b>          | <b>Composants</b>                      | <b>6</b>  |
| 3.1               | Inverter PCB . . . . .                 | 6         |
| 3.1.1             | Pont-H . . . . .                       | 7         |
| 3.1.1.1           | Temps mort . . . . .                   | 7         |
| 3.1.2             | Filtre passe-bas . . . . .             | 8         |
| 3.1.3             | Transformateur . . . . .               | 8         |
| 3.1.4             | Filtre passe-bas logic-level . . . . . | 9         |
| 3.1.5             | Points de mesures . . . . .            | 9         |
| 3.2               | Carte FPGA . . . . .                   | 10        |
| 3.3               | Boutons et LEDs . . . . .              | 11        |
| <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 générer une alimentation AC 50Hz à partir d'un signal DC. Le système est visible sur la figure 1

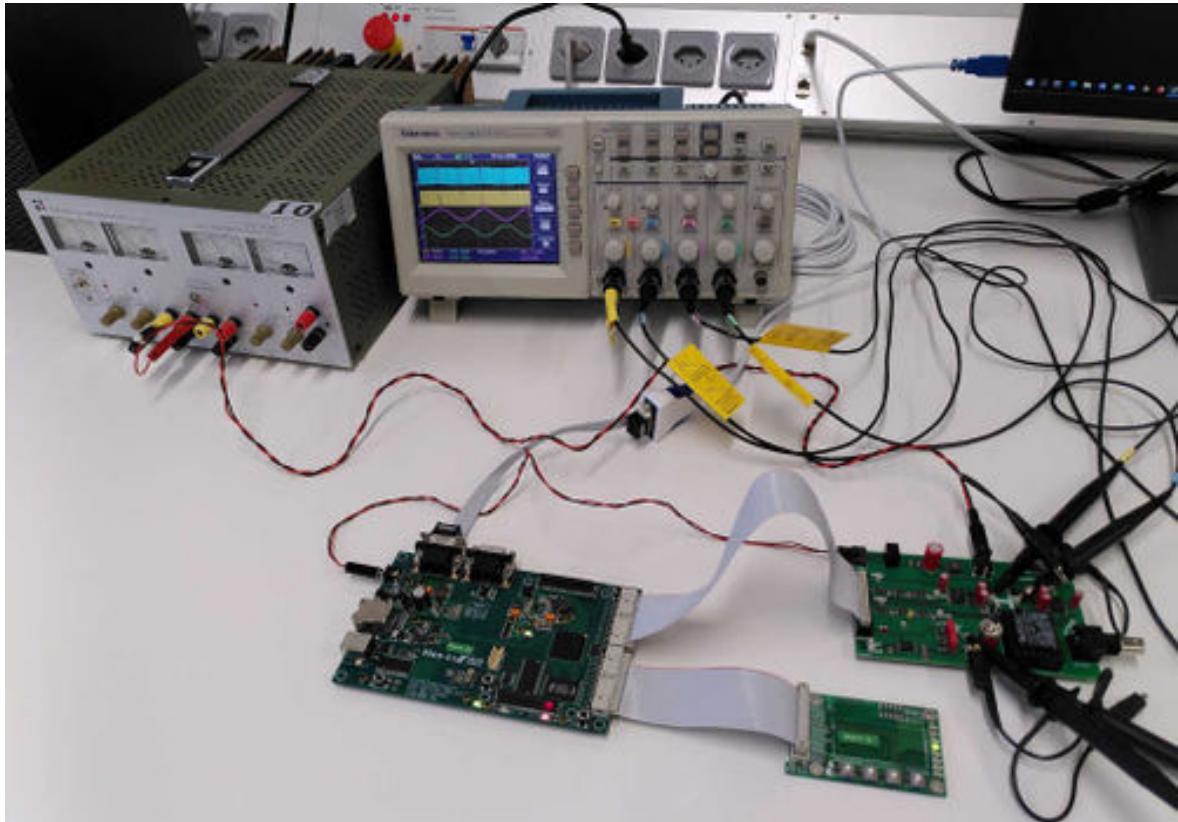
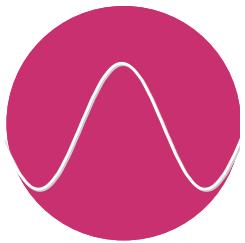


FIGURE 1 – Équipement de l'inverter

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

Le système génère un signal sinusoïdal **AC** 50Hz à partir d'un signal **DC**. Celui-ci est haché par un **pont-H** et lissé par un **filtre passe-bas** afin de fournir un sinus propre. Le **pont-H** est contrôlé par 4 signaux **PWM**.

### 2.1 Fonctions

Les fonctions de base sont définies comme suit :

- Une génération d'un signal **AC** 50Hz sinusoïdal en pilotant le **pont-H** à l'aide de signaux **PWM**.
- Un mode **PWM** à deux niveaux. Celui-ci utilise des contrôles inversés sur chaque branche du **pont-H**. Le principe est démontré à la figure 2.
- Un mode **PWM** à trois niveaux. Chaque branche du **pont-H** est contrôlée indépendamment. Le principe est démontré à la figure 3.

L'erreur de fréquence admettable sur le sinus est de 1%.

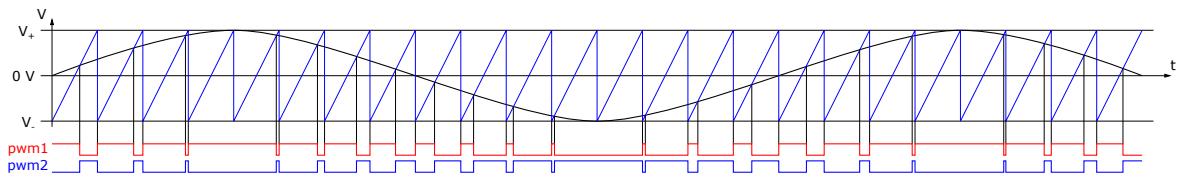


FIGURE 2 – Diagramme du contrôle à 2 niveaux

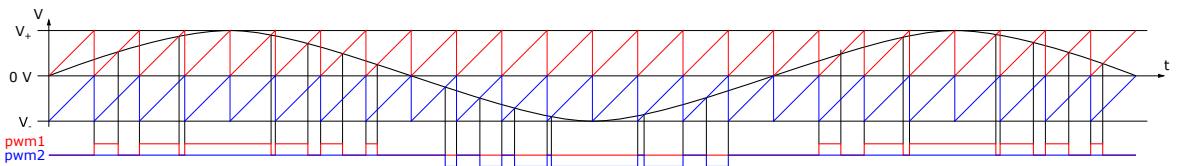


FIGURE 3 – Diagramme du contrôle à 3 niveaux

### 2.2 Circuit

Pour accomplir la tâche, le circuit montré à la figure 4 est utilisé.

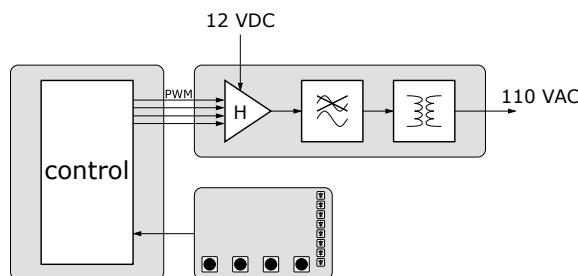
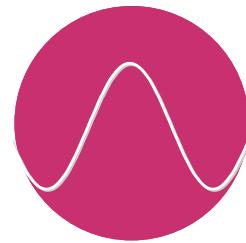


FIGURE 4 – Circuit de l'inverter



Le circuit fonctionne comme suit :

- Le compteur de phase génère un signal triangulaire qui sera utilisé par le bloc **CORDIC** pour générer le sinus.
  - Le modulateur **PWM** utilise le sinus pour générer deux signaux **PWM** permettant de contrôler les 2 branches du **pont-H**.
  - Les blocs de non-overlap permettent de contrôler la charge sans court-circuit sur l'alimentation.
  - Les boutons permettent de passer d'un mode **PWM** à l'autre en agissant sur les signaux *threeLevel*, *doubleFrequency* et *switchEvenOdd*.
  - Les broches *testOut* peuvent être utilisées pour sortir des informations supplémentaires du système, par exemple pour le débogage ou pour contrôler les **LEDs**.
  - Les signaux *trigger*, *adcData* (qui est traitée en interne) et *mainsTriggered* fournissent des signaux utiles pour des options.

Le toplevel du design (inverter-toplevel.pdf) montre tous les signaux connectés à la platine **Field Programmable Gates Array (FPGA) 5** ainsi que la structure de base fournie. La figure 6 montre le bloc de niveau supérieur avec ses signaux d'entrée et de sortie.

Certains blocs ne sont pas vides et réutilisent des éléments vus aux labos précédents, c'est le cas du [CORDIC](#).

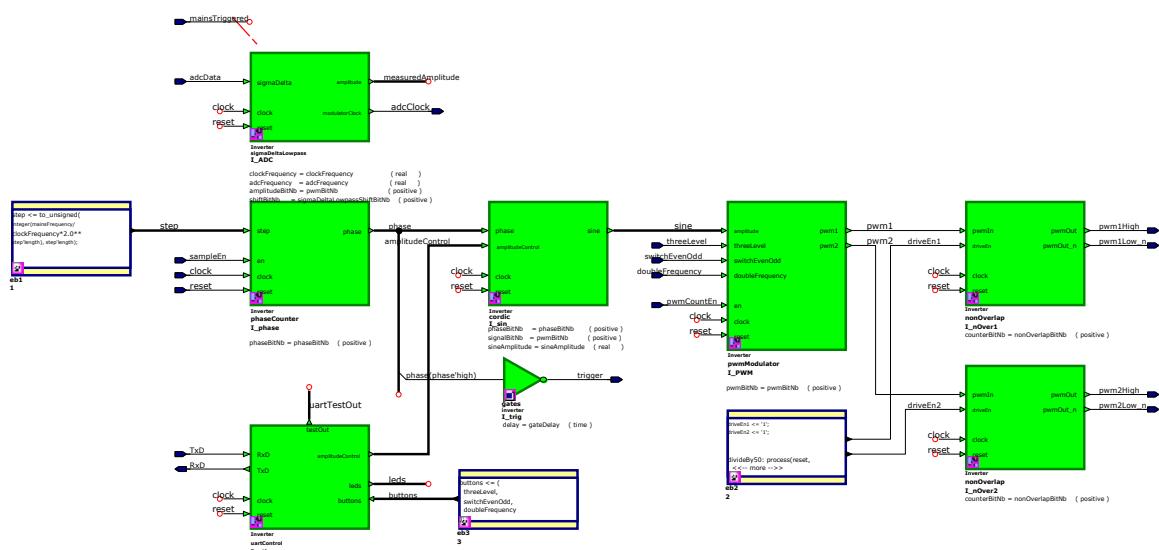


FIGURE 5 – Circuit Toplevel

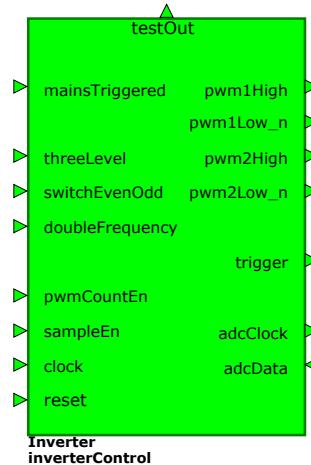
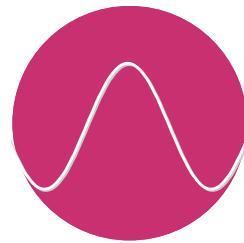


FIGURE 6 – Top bloc

### 2.3 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 :

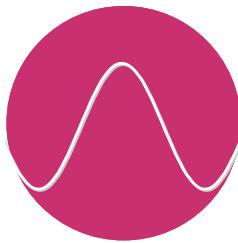
```
did_inverter
+--Board/           # Project and files for programming the FPGA
|   +--concat/      # Complete VHDL file including PIN-UCF file
|   +--ise/          # Xilinx ISE project
+--Inverter/        # Library for the components of the student solution
+--Inverter_test/   # Library for the simulation 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 convertisseur avec une carte électronique "Printed Circuit Board (PCB)" qui commande les éléments de puissance, voir figure 7
- Une carte de développement **FPGA**, voir figure 16
- Une carte de contrôle à 4 boutons et 8 LEDs, voir figure 17

#### 3.1 Inverter PCB

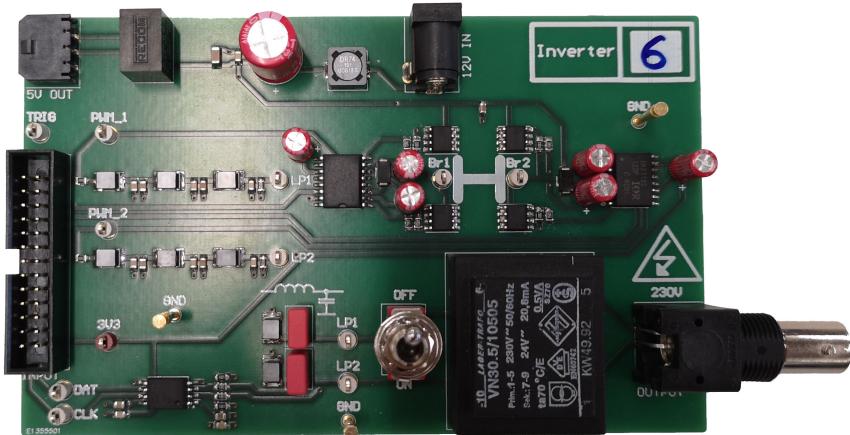


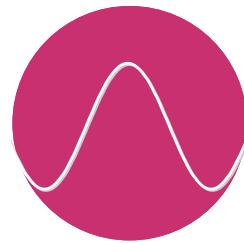
FIGURE 7 – PCB de l'inverter

Le **PCB** de l'inverter comporte l'intégralité des éléments permettant le passage de signaux numériques (sous forme de **PWM**) à un signal analogique [12].

- Le **pont-H**, décrit à la section 3.1.1.
- Un **filtre passe-bas LC**, décrit à la section 3.1.2.
- Le transformateur permettant de fournir un signal jusqu'à 230V, décrit à la section 3.1.3
- Une **filtre passe-bas** filtrant les signaux **PWM**, décrit à la section 3.1.4.

Le **PCB** comporte également un régulateur 5V alimentant la platine **FPGA**. Les alimentations et signaux sont transportés via les câbles plats connectant les différents éléments du système.

Un **convertisseur sigma-delta** permet de mesurer les signaux générés en sortie du **filtre passe-bas**.



### 3.1.1 Pont-H

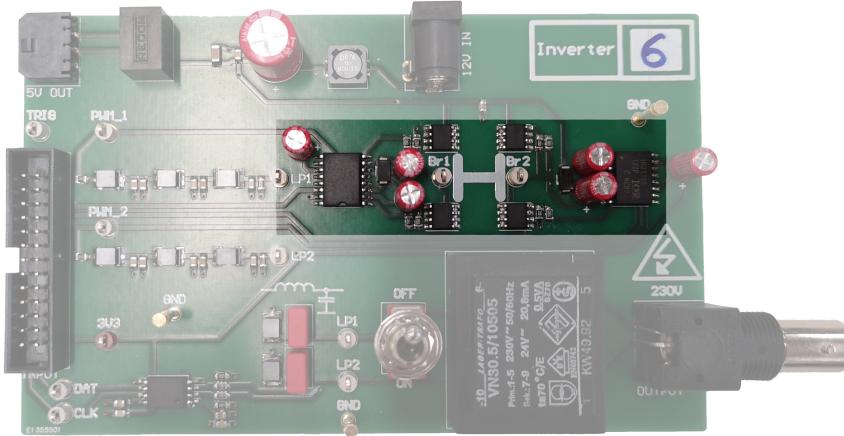


FIGURE 8 – PCB de l'inverter - partie pont-H

Le **pont-H** (mis en évidence sur la figure 8) est alimenté en 12V et constitué de 4 **MOSFET** IRF7809 [7] commandés par deux drivers IR2113S [6] (étage de puissance et étage de commande).

Les drivers sont contrôlés directement par les signaux **PWM** filtrés. Ceux-ci doivent être à une fréquence inférieure à 100kHz.

**3.1.1.1 Temps mort** La figure 9 montre le principe d'un **pont-H** et en particulier le risque de court-circuit qui apparaît si les deux **MOSFET** sont fermés en même temps lors d'un changement d'état. Pour palier à ce problème, un temps mort doit être prévu. La figure 10 illustre l'effet de ce temps-mort sur les signaux **PWM**.

Ce temps doit être au minimum de 200ns pour permettre aux **MOSFET** de changer d'état complètement. Afin de ne pas impacter les signaux **PWM**, la période de ceux-ci doit être supérieure à 100 fois l'ordre de grandeur.

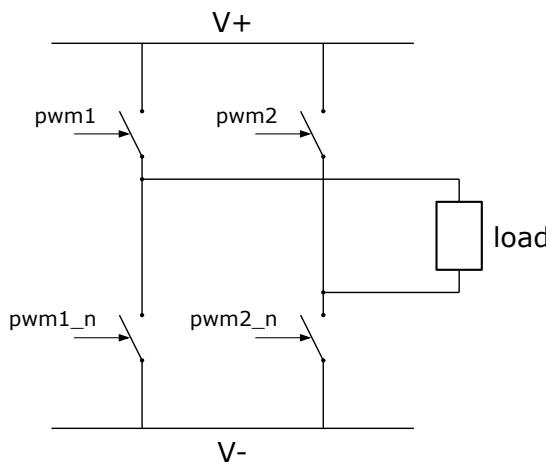


FIGURE 9 – Schéma d'un pont-H

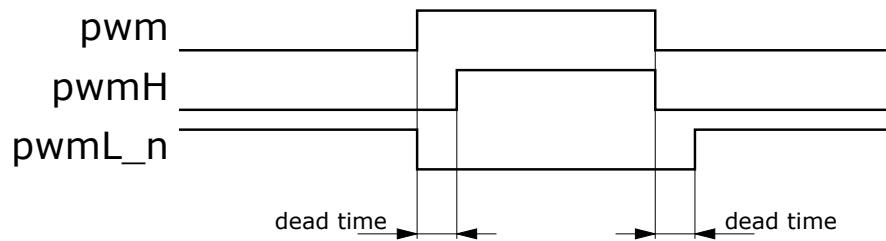
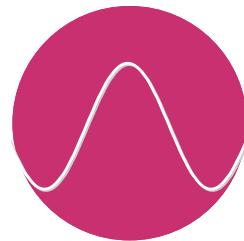


FIGURE 10 – Chronogramme du temps mort dans le pont-H

### 3.1.2 Filtre passe-bas

Le **filtre passe-bas** (mis en évidence sur la figure 11) de type LC du 2nd ordre permet de lisser le signal **PWM** de sortie du **pont-H** vers un signal sinusoïdal. Sa fréquence de coupure est de 8.76kHz.



Veillez à ne pas choisir une fréquence de **PWM** trop proche de cette fréquence de coupure ou des artefacts pourront se présenter.

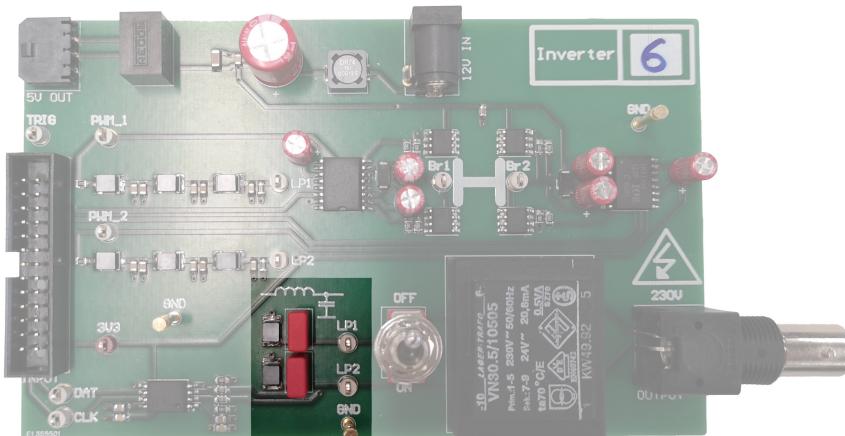


FIGURE 11 – PCB de l'inverter - partie filtre passe-bas

### 3.1.3 Transformateur

Le transformateur (mis en évidence sur la figure 12) permet d'élever la tension de sortie de 12V vers une tension pouvant atteindre 230V. Il permet également de référencer le signal différentiel de sortie à n'importe quelle tension commune.



La sortie du transformateur peut être désactivée à l'aide de l'interrupteur placé à cet effet.



Une grande prudence est de mise lors de l'activation du transformateur à cause des tensions élevées en sortie !

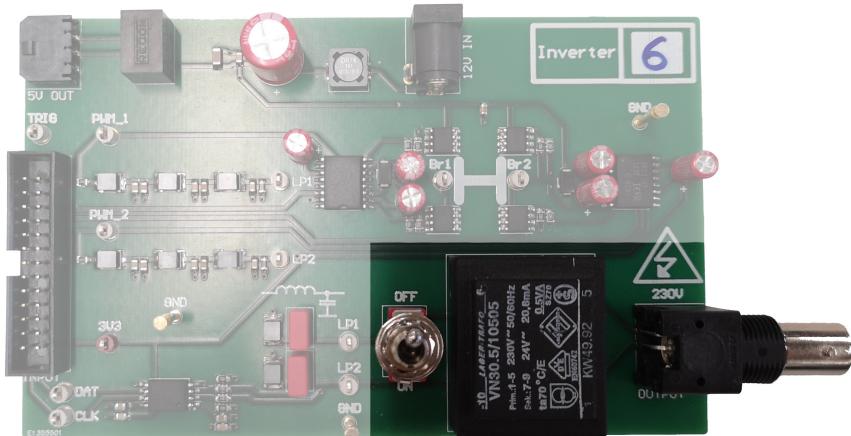
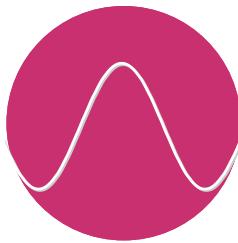


FIGURE 12 – PCB de l'inverter - partie transformateur

### 3.1.4 Filtre passe-bas logic-level

Le **filtre passe-bas** (mis en évidence sur la figure 13) de type LC du 6ème ordre permet de lisser les signaux **PWM** de contrôle **pont-H** vers une paire différentielle sinusoïdale. Sa fréquence de coupure est proche de 20kHz.

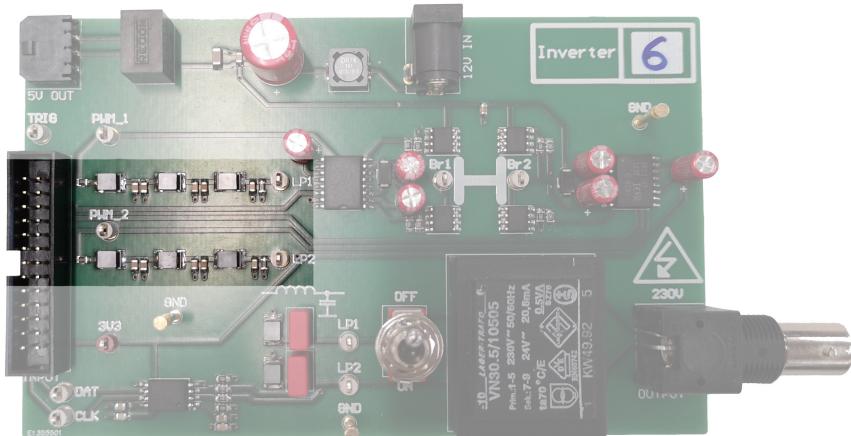
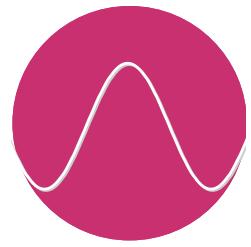


FIGURE 13 – PCB de l'inverter - partie filtre passe-bas logic-level

### 3.1.5 Points de mesures

Le **PCB** de l'inverter comporte plusieurs points de mesures (mis en évidence sur la figure 14). Ceux-ci permettent de mesurer :

- les signaux **PWM** générés par votre système (*PWM\_1* et *PWM\_2*)
- ces mêmes signaux en sortie du **filtre passe-bas logic-level** (*LP1* et *LP2* situés près du centre de la carte)
- les deux signaux différentiels de sortie du **pont-H**, *Br1* et *Br2*
- ainsi que les signaux de sortie finale, après le **filtre passe-bas** suivant le **pont-H** (*LP1* et *LP2* situés au centre en bas de la carte)



D'autres points sont disponibles, tout d'abord des bornes *GND* et *3V3* ainsi que les deux signaux de sortie du [convertisseur sigma-delta](#), *DAT* et *CLK* ou encore un signal de trigger *TRIG*.

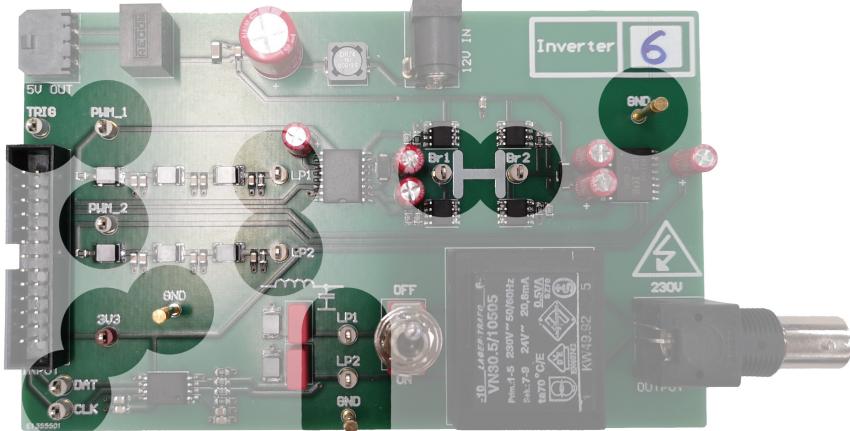


FIGURE 14 – [PCB](#) de l'inverter - parties points de mesures

### 3.2 Carte FPGA

La carte principale est la carte de développement de laboratoire FPGA-EBS 2 de l'école [8]. Elle héberge une puce [Xilinx Spartan xc3s500e FPGA](#) [[Spartan3FPGAFamily](#)] [13] 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  $f_{clk} = 66MHz$  [3].

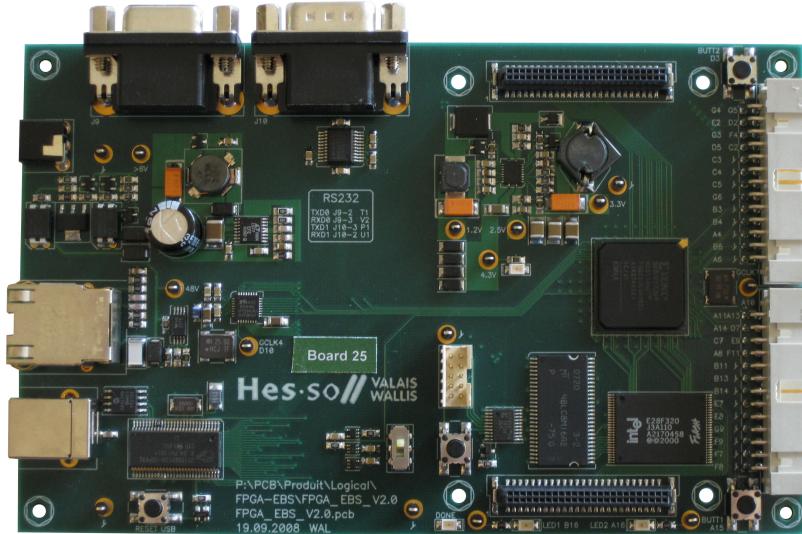
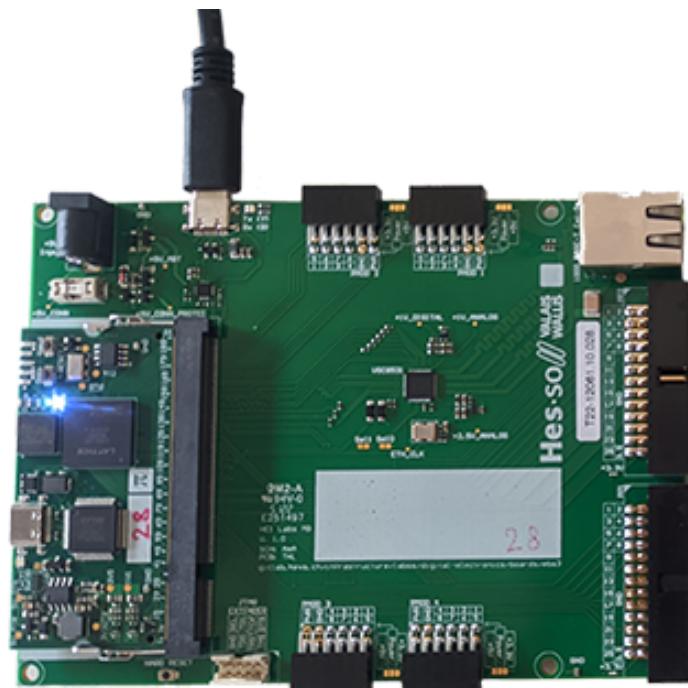
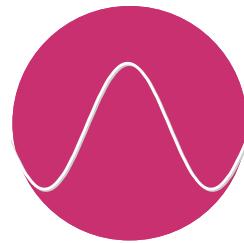


FIGURE 15 – [Carte électronique FPGA](#) [8]

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

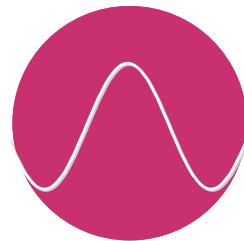
FIGURE 16 – Carte électronique **FPGA EBS3** [1]

Les simulateurs sont réglés par défaut pour les boards EBS3. 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.

### 3.3 Boutons et **LEDs**

La platine avec les boutons et les **LEDs** [9] 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** [10] [4].

FIGURE 17 – Carte électronique boutons-**LED-LCD** [9]



## 4 Evaluation

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

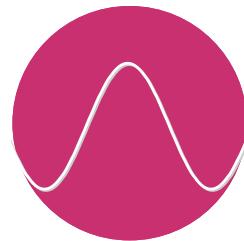
La note finale contient le rapport, le code ainsi qu'une présentation de votre système.

| 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" [2]



## 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.
- Implémenter et simuler les différents blocs.
- Testez la solution sur le circuit imprimé et trouvez les éventuelles erreurs .

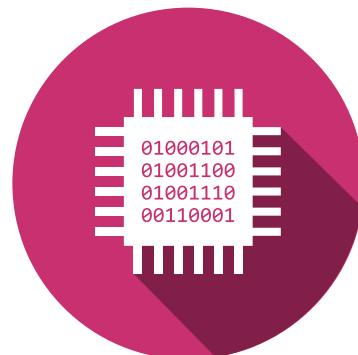
### 5.1 Tips

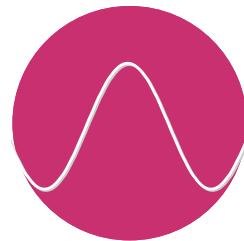
Ci-joint quelques conseils supplémentaires pour éviter les problèmes et les pertes de temps :

- Divisez le problème en différents blocs, utilisez pour cela le document Toplevel (inverter-toplevel.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. [5].
- Il est recommandé de réaliser le système en plusieurs étapes.
  - Calculez les différentes fréquences et tailles de compteurs
  - Commencez par les blocs en amont réalisant les fonctions de bases
  - Puis ajoutez les fonctions plus avancées, telle que la gestion du temps mort



N'oubliez pas de vous amuser 😊.





## Références

- [1] AMAND AXEL. *Schematic : FPGA-EBS3 v1.0.* 2023.
- [2] CHRISTOPHE BIANCHI, FRANÇOIS CORTHAY et SILVAN ZAHNO. *Comment Rédiger Un Rapport de Projet ?* 2021.
- [3] CTS. *Datasheet CTS Model CB3 & CB3LV HCMOS/TTL Clock Oscillator.* 2006.
- [4] ELECTRONIC ASSEMBLY. *Datasheet : DOGM Graphics Series 132x32 Dots.* 2005.
- [5] FRANÇOIS CORTHAY, SILVAN ZAHNO et CHRISTOPHE BIANCHI. *Méthodologie de Conception de Circuits Numériques.* 2021.
- [6] INFEON. *Datasheet IR2110 High Speed MOSFET.* 2012.
- [7] International IOR RECTIFIER. *Datasheet IRF7809 N-Channel MOSFET.* 2012.
- [8] SILVAN ZAHNO. *Schematic : FPGA-EBS v2.2.* 2014.
- [9] SILVAN ZAHNO. *Schematic : Parallelport HEB LCD V2.* 2014.
- [10] SITRONIX. *Datasheet Sitronix ST7565R 65x1232 Dot Matrix LCD Controller/Driver.* 2006.
- [11] STMICROELECTRONICS. *Datasheet : DMOS Dual Full Bridge Driver with PWM Current Controller.* 2003.
- [12] David TAGAN. *Schematic : Parallelport Inverter V1.* 2021.
- [13] XILINX. *Datasheet Spartan-3E FPGA Family.* 2008.

## Acronymes

**AC** Alternative Current. 3

**DC** Direct Current. 2, 3

**FPGA** Field Programmable Gates Array. 1, 4, 6, 10, 11

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

**LED** Light Emitting Diodes. 1, 4, 6, 11

**MOSFET** metal–oxide–semiconductor field-effect transistor. 7

**PCB** Printed Circuit Board. 1, 6–10

**PWM** Pulse Width Modulation. 3, 4, 6–9

**UART** Universal Asynchronous Receiver Transmitter. 10

**USB** Universal Serial Bus. 10