

# Inverter - Energieumwandlung

Projekt Digitales Design



$\pi$  Haute Ecole d'Ingénierie  
Hochschule für Ingenieurwissenschaften

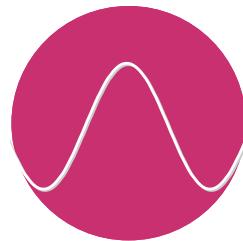
Orientierung: [Energie und Umwelttechnik \(ETE\)](#)

Kurs: [Digitales Design \(DiD\)](#)

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

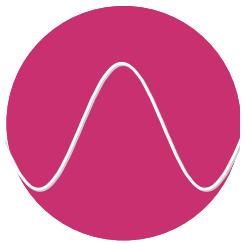
Datum: 12. April 2023

Version: v2.0



## Inhaltsverzeichnis

<b>1 Einführung</b>	<b>2</b>
<b>2 Spezifikation</b>	<b>3</b>
2.1 Funktionen . . . . .	3
2.2 Schaltung . . . . .	3
2.3 HDL-Designer Projekt . . . . .	5
<b>3 Komponenten</b>	<b>6</b>
3.1 Inverter PCB . . . . .	6
3.1.1 H-Brücke . . . . .	7
3.1.1.1 Totzeit . . . . .	7
3.1.2 Tiefpassfilter . . . . .	8
3.1.3 Transformatoren . . . . .	8
3.1.4 Tiefpassfilter logic-level . . . . .	9
3.1.5 Messpunkte . . . . .	9
3.2 FPGA-Platine . . . . .	10
3.3 Knöpfe und LEDs . . . . .	11
<b>4 Bewertung</b>	<b>12</b>
<b>5 Erste Schritte</b>	<b>13</b>
5.1 Tips . . . . .	13
<b>Literatur</b>	<b>14</b>
<b>Akronyme</b>	<b>14</b>



## 1 Einführung

Das Projekt hat die Aufgabe das erlernte Wissen am Ende des Semesters an einem praktischen Beispiel direkt anzuwenden. Es geht darum, aus einem Gleichstromsignal eine 50-Hz-Wechselstromversorgung zu erzeugen. Das System ist in Abbildung 1 zu sehen. Das System kann in der Abbildung 1 betrachtet werden.

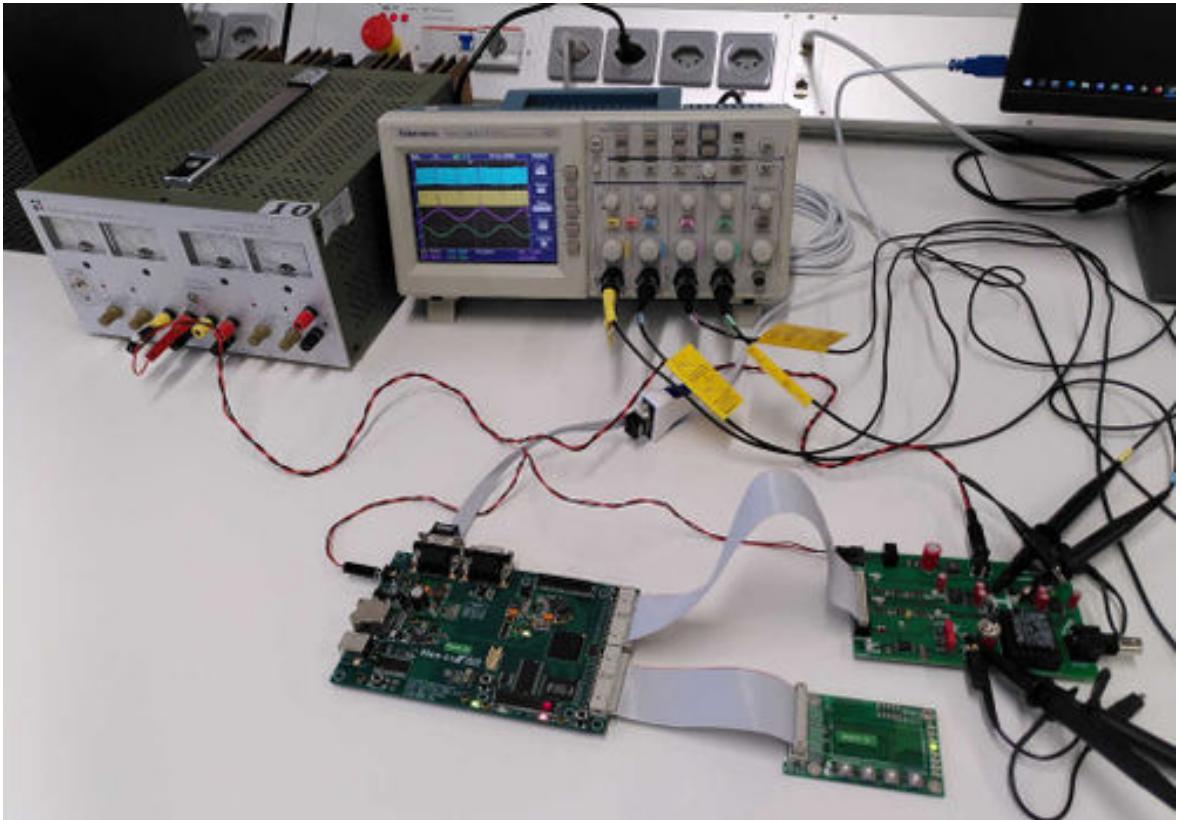
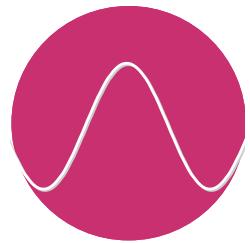


Abbildung 1: Hardwareaufbau Inverter

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

Das System erzeugt ein 50Hz Sinussignal **AC** aus einem **DC**-Signal. Dieses wird durch eine **H-Brücke** gehackt und durch einen **Tiefpassfilter** geglättet, um einen sauberen Sinus zu liefern. Der **H-Brücke** wird durch 4 **PWM**-Signale gesteuert.

### 2.1 Funktionen

Die Basisfunktionen sind wie folgt definiert:

- Erzeugung eines sinusförmigen 50 Hz **AC**-Signals durch Ansteuerung des **H-Brücke** mit **PWM**-Signalen.
- Ein zweistufiger **PWM**-Modus. Dieser verwendet die invertierten Signale auf jeder Seite der **H-Brücke**. Das Prinzip wird in Abbildung 2 demonstriert.
- Ein dreistufiger **PWM**-Modus. Jede Seite der **H-Brücke** wird unabhängig gesteuert. Das Prinzip wird in der Abbildung 3 demonstriert.

Der zulässige Frequenzfehler beim Sinus beträgt 1%.

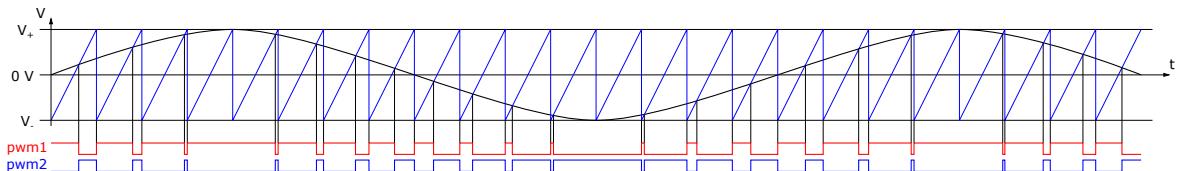


Abbildung 2: Diagramm der 2-stufigen Kontrolle

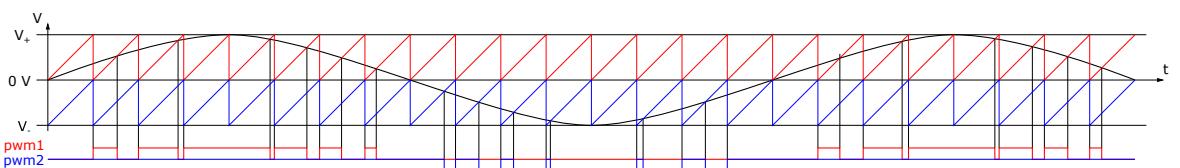


Abbildung 3: Diagramm der 3-stufigen Kontrolle

### 2.2 Schaltung

Um die Aufgabe zu erfüllen, wird die in Abbildung 4 gezeigte Schaltung verwendet.

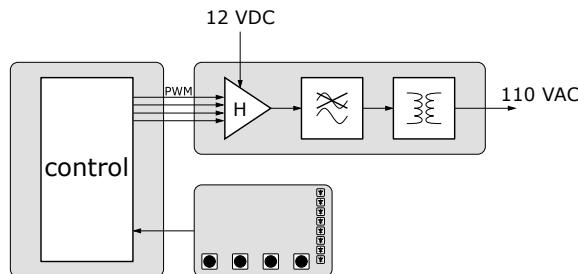
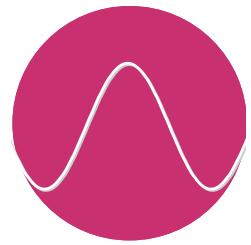


Abbildung 4: Inverter Schaltung



Die Schaltung funktioniert wie folgt:

- Der Phasenzähler erzeugt ein Dreieckssignal, das vom Block **CORDIC** verwendet wird, um den Sinus zu erzeugen.
- Der **PWM** Modulator verwendet den Sinus, um zwei **PWM**-Signale zu erzeugen, mit denen die beiden Zweige der H-Brücke gesteuert werden können.
- Non-Overlap-Blöcke ermöglichen die Steuerung der Last ohne Kurzschluss der Stromversorgung.
- Mit den Tasten können Sie durch Beeinflussung der Signale *threeLevel*, *doubleFrequency* und *switchEvenOdd* von einem **PWM**-Modus in den anderen umschalten.
- Die *testOut*-Pins können verwendet werden, um zusätzliche Informationen aus dem System auszugeben, z. B. für Debuggingzwecke oder zur Steuerung von **LEDs**.
- Die Signale *trigger*, *adcData* (welche intern benutzt werden) und *mainsTriggered* liefern nützliche Signale für Optionen.

Der Design-Toplevel (inverter-toplevel.pdf) zeigt alle Signale, die mit der Platine verbunden sind **Field Programmable Gates Array (FPGA)**. 5 sowie die mitgelieferte Grundstruktur. Die Abbildung 6 zeigt dem Toplevel Block mit seinen dazugehörigen Ein- und Ausgangssignalen.



Einige Blöcke sind nicht leer und verwenden Elemente, die wir in den vorherigen Labors gesehen haben, die ist beim **CORDIC** der Fall.

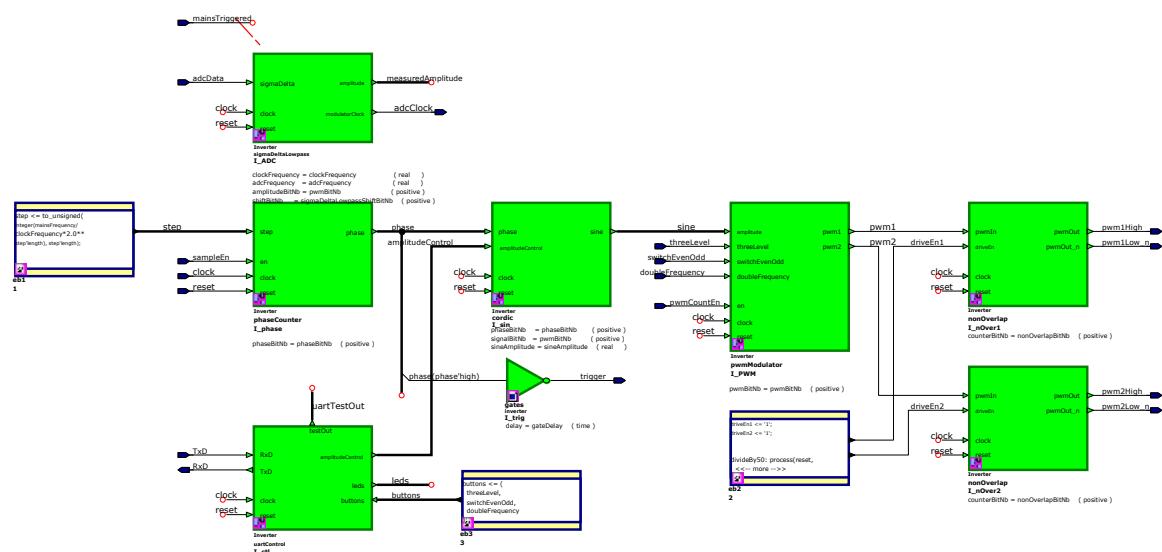


Abbildung 5: Toplevel Schaltung

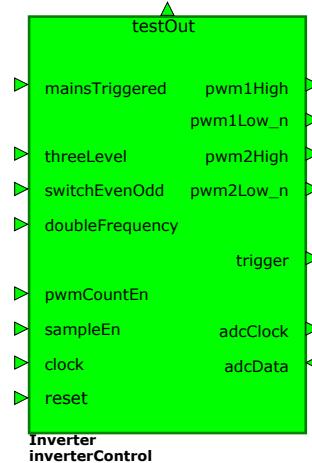
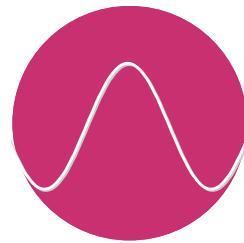


Abbildung 6: Topbloc Schaltung

### 2.3 HDL-Designer Projekt

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

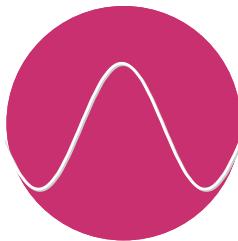
```
did_inverter
+--Board/           # Project and files for programming the FPGA
|   |--concat/      # Complete VHDL file including PIN-UCF file
|   |--ise/          # Xilinx ISE project
+--Inverter/         # Library for the components of the student solution
+--Inverter_test/    # Library for the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



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 Hardwareplatten, welche in der Abbildung 1 ersichtlich sind:

- eine Umrichter mit einer elektronischen Platine "Printed Circuit Board (PCB)", der die Leistungselemente steuert, siehe Abbildung 7
- einer **FPGA** Entwicklungsplatine, siehe Abbildung 16
- einer Steuerplatine mit 4 Tasten und 8 **LEDs**, siehe Abbildung 17

#### 3.1 Inverter PCB

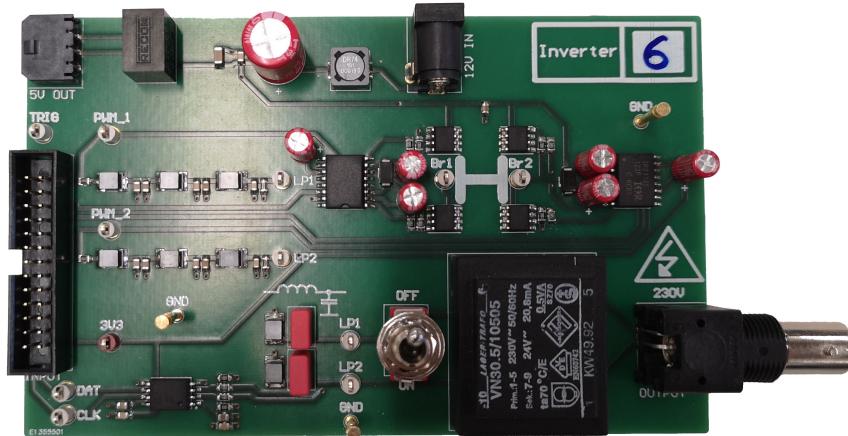


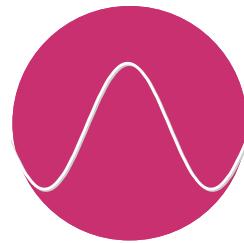
Abbildung 7: Inverter PCB

Der **PCB** des Inverters enthält alle Elemente, die es ermöglichen, digitale Signale (in Form von **PWM**) in ein analoges [12]-Signal umzuwandeln.

- Die **H-Brücke**, die im Abschnitt 3.1.1 beschrieben wird.
- Einen **LC-Tiefpassfilter**, beschrieben im Abschnitt 3.1.2.
- Der Transformator, der ein Signal bis zu 230 V liefert, wird im Abschnitt 3.1.3 beschrieben.
- Einen weiteren **LC-Tiefpassfilter**, der die **PWM** Signale filtert, wird im Abschnitt 3.1.4 beschrieben.

Das **PCB** enthält auch einen 5-V-Regler, der die **FPGA**-Platine mit Strom versorgt. Die Stromversorgung und die Signale werden über die Flachbandkabel übertragen, die die verschiedenen Komponenten des Systems miteinander verbindet.

Ein Sigma-Delta-Konverter misst die Signale am Ausgang des **Tiefpassfilters**.



### 3.1.1 H-Brücke

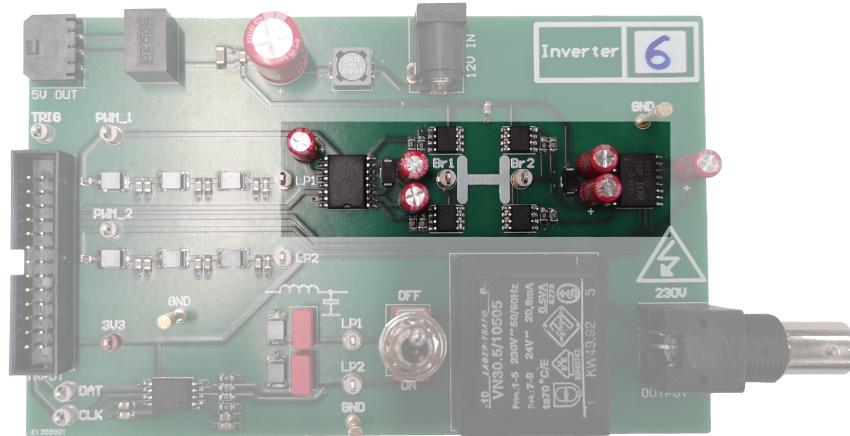


Abbildung 8: Inverter PCB - Bereich der H-Brücke

Die **H-Brücke** (in der Abbildung 8) wird mit 12 V betrieben und besteht aus vier **MOSFET** Treibern IRF7809 [7], die von zwei Treibern IR2113S [6] (Leistungsstufe und Steuerstufe) angesteuert werden.

Die Treiber werden direkt von den gefilterten **PWM**-Signalen gesteuert. Diese müssen eine Frequenz von weniger als 100 kHz haben.

**3.1.1.1 Totzeit** Die Abbildung 9 zeigt das Prinzip einer **H-Brücke** und insbesondere die Gefahr eines Kurzschlusses, die entsteht, wenn beide **MOSFET** Treiber bei einem Zustandswechsel gleichzeitig geschlossen werden. Um dieses Problem zu lösen, muss eine Auszeit (Totzeit) vorgesehen werden. Die Abbildung 10 zeigt, wie sich diese Auszeit auf die **PWM**-Signale auswirkt.

Diese Zeit muss mindestens 200ns betragen, damit die **MOSFET** Treiber ihren Zustand vollständig ändern können. Um die **PWM**-Signale nicht zu beeinflussen, muss die Periode dieser Signale mehr als das 100-fache der Größenordnung betragen.

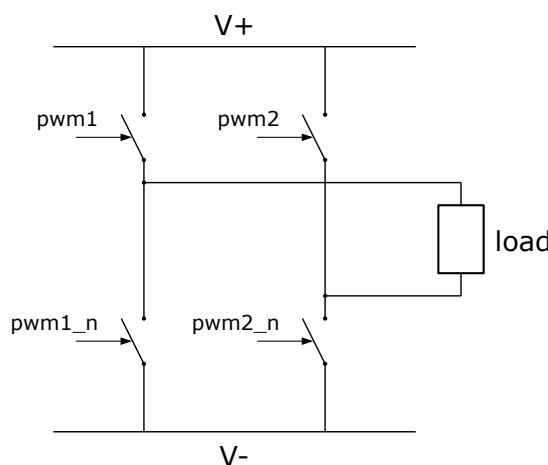


Abbildung 9: Schema eines H-Brücke

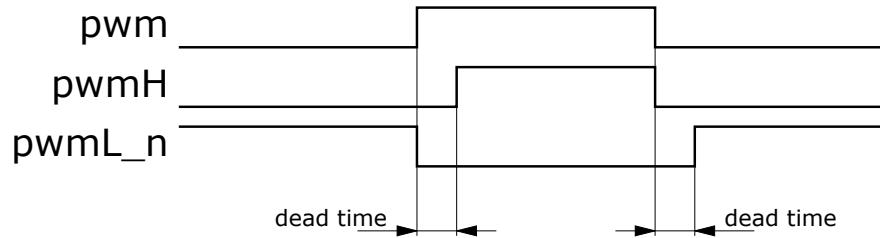
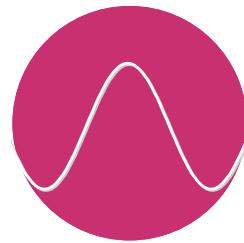


Abbildung 10: Chronogramm der Totzeit in der H-Brücke

### 3.1.2 Tiefpassfilter

Der Tiefpassfilter (in der Abbildung 11 hervorgehoben) vom Typ LC 2ter Ordnung glättet das PWM-Ausgangssignal des H-Brücke zu einem Sinussignal. Seine Schnittfrequenz liegt bei 8,76 kHz.



Achten Sie darauf, dass die Frequenz des PWM nicht zu nahe an dieser Schnittfrequenz liegt, da sonst Artefakte auftreten können.

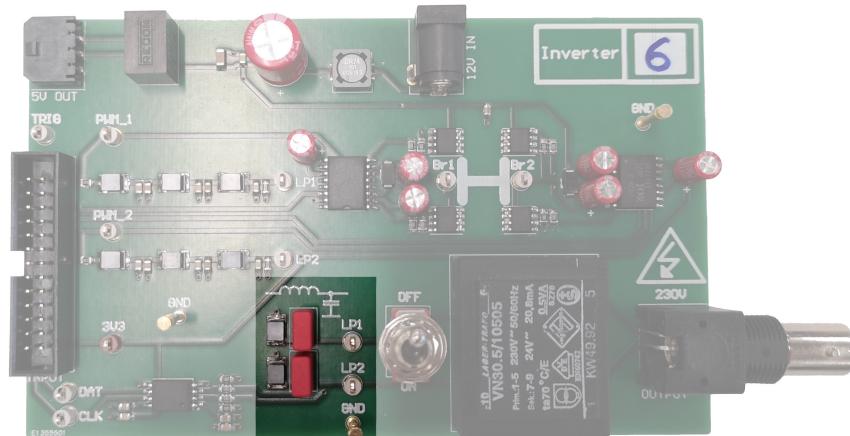


Abbildung 11: Inverter PCB - Bereich des Tiefpassfilters

### 3.1.3 Transformator

Der Transformator (in der Abbildung 12 hervorgehoben) ermöglicht es, die Ausgangsspannung von 12 V auf eine Spannung von bis zu 230 V zu erhöhen. Außerdem ermöglicht er es, das differentielle Ausgangssignal auf eine beliebige gemeinsame Spannung zu beziehen.



Der Ausgang des Transformators kann mit dem Schalter deaktiviert werden.



Wegen der hohen Ausgangsspannungen ist bei der Aktivierung des Transformators grosse Vorsicht geboten!

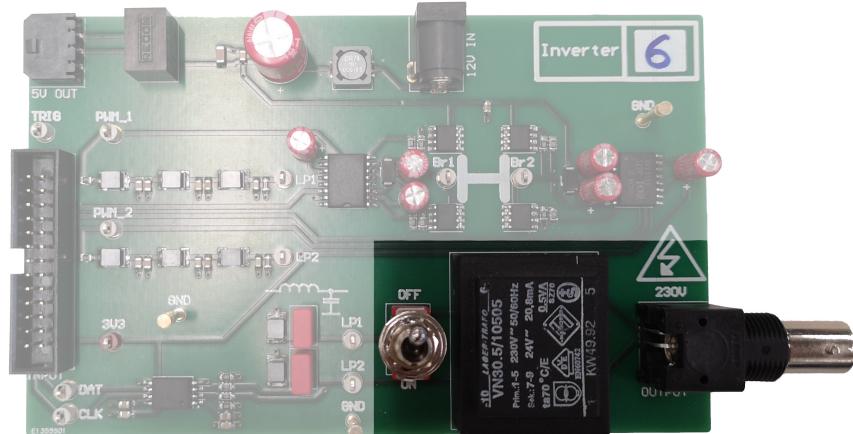
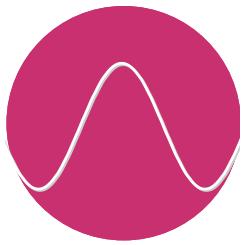


Abbildung 12: Inverter [PCB](#) - Bereich des Transformators

### 3.1.4 Tiefpassfilter logic-level

Der Tiefpassfilter (siehe Abbildung 13) vom Typ LC 6ter Ordnung glättet die [PWM](#) Steuersignale der [H-Brücke](#) zu einem differentiellen Sinuspaar. Seine Schnittfrequenz liegt nahe bei 20kHz.

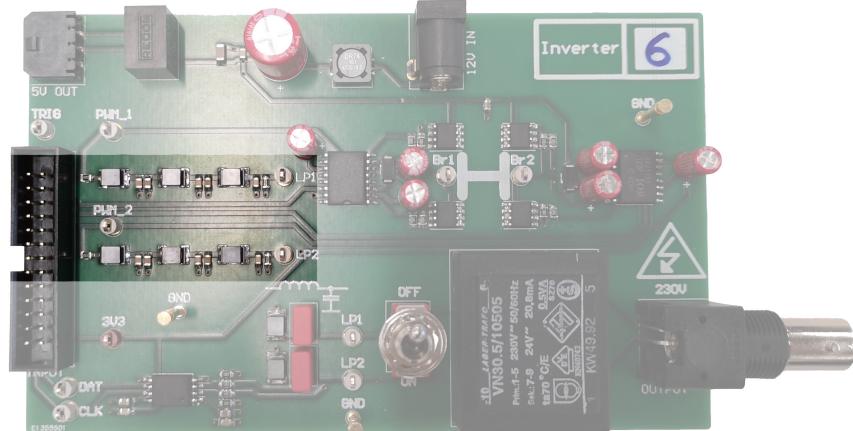
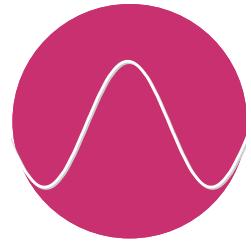


Abbildung 13: Inverter [PCB](#) - Bereich des [Tiefpassfilters logic-level](#)

### 3.1.5 Messpunkte

Der Inverter verfügt über mehrere Messpunkte (in der Abbildung 14 hervorgehoben). Diese ermöglichen die folgenden Messung:

- die von Ihrem System erzeugten Signale ( $PWM_1$  und  $PWM_2$ ).
- die gleichen Signale am Ausgang des logic-level Tiefpassfilters ( $LP1$  und  $LP2$  befinden sich in der Nähe der Kartenmitte).
- die beiden Differenzsignale am Ausgang des [H-Brücke](#),  $Br1$  und  $Br2$ .
- sowie die endgültigen Ausgangssignale nach dem [Tiefpassfilter](#), welcher der [H-Brücke](#) folgt. ( $LP1$  und  $LP2$  befinden sich mittig im unteren Bereich der Karte).



Weitere Punkte sind verfügbar, die Speisungen *GND* und *3V3* sowie die beiden Ausgangssignale des **Sigma-Delta-Konverters** *DAT* und *CLK*, oder auch das Triggersignal *TRIG*.

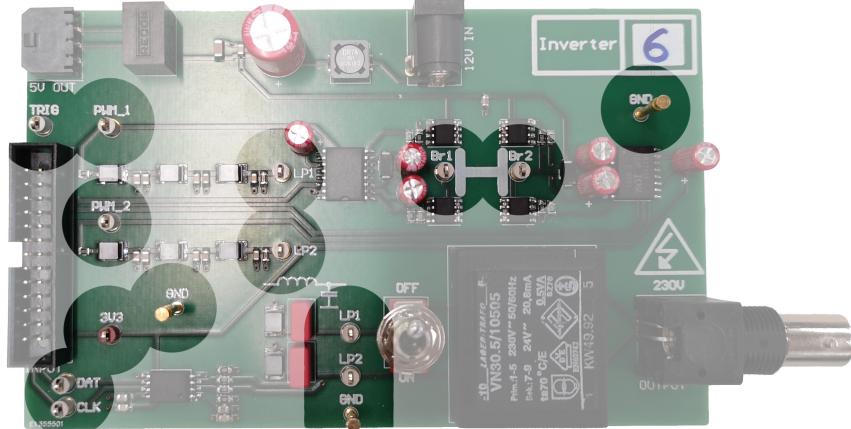


Abbildung 14: Inverter PCB - Teile Messpunkte

### 3.2 FPGA-Platine

Die Hauptplatine ist die FPGA-EBS 2 Laborentwicklungsplatine der Schule [8]. Diese beherbergt eine **Xilinx Spartan xc3s500e FPGA** [**Spartan3FPGAFamily**] [13] und verfügt über viele verschiedene Schnittstellen (**Universal Asynchronous Receiver Transmitter (UART)**, **Universal Serial Bus (USB)**, Ethernet, etc.). Der benutzte Oszillator erstellt ein Taktsignal (*clock*) mit einer Frequenz von  $f_{clk} = 66MHz$  [3].

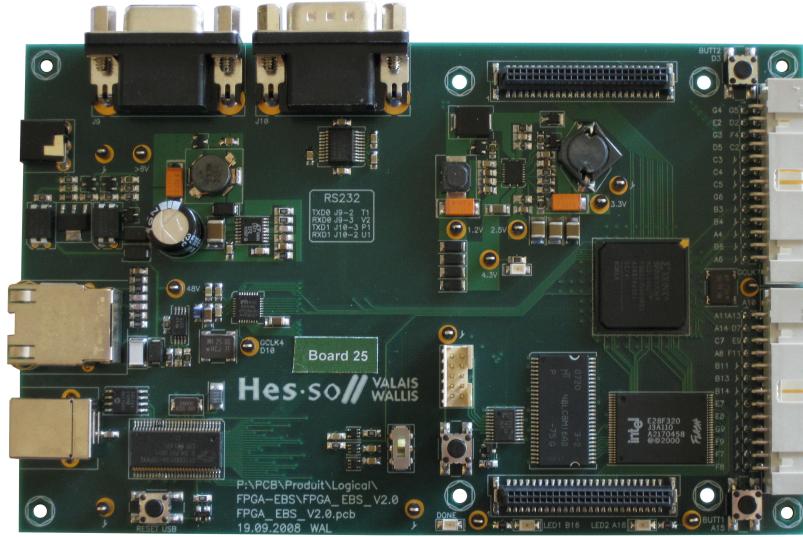
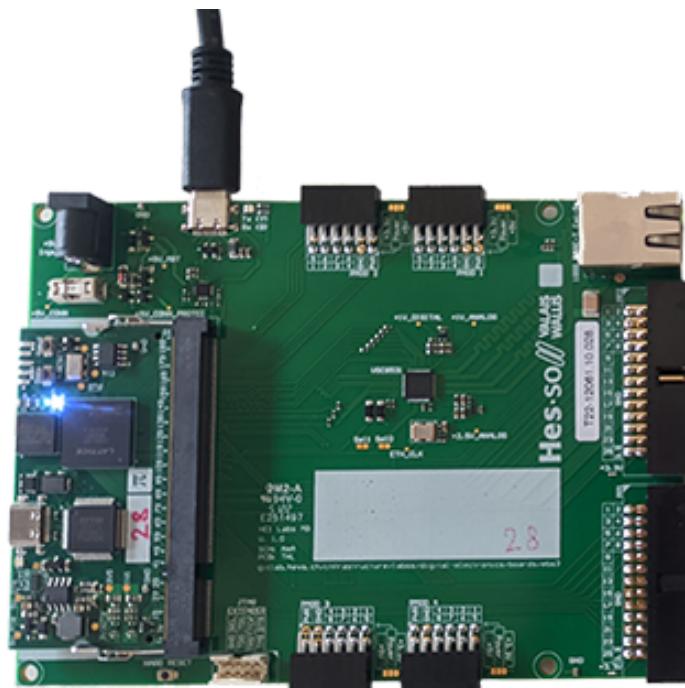
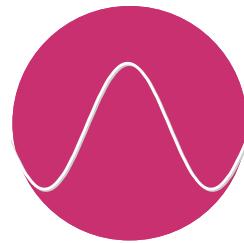


Abbildung 15: FPGA Platine [8]

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.

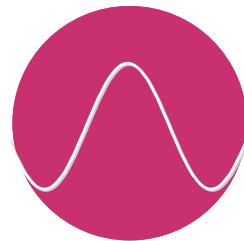
Abbildung 16: EBS3 **FPGA** Platine [1]

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.

### 3.3 Knöpfe und LEDs

Die Platine mit den Knöpfen und **LEDs** [9] 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 [10] [4].

Abbildung 17: Knöpfe-**LED-LCD** Platine [9]



## 4 Bewertung

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

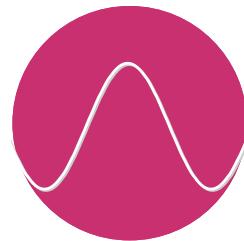
Die Schlussnote beinhaltet den Bericht, den Code sowie eine Präsentation eurerseits des Systems.

Evaluierter Aspekt	Punkte
<b>Bericht</b>	<b>55</b>
Einleitung	3
Spezifikation	5
Entwurf	20
Verifizierung und Validation	10
Integration	9
Schlussfolgerung	3
Formale Aspekte des Berichtes	5
<b>Funktionalität der Schaltung</b>	<b>30</b>
<b>Qualität der Lösung</b>	<b>10</b>
<b>Präsentation</b>	<b>10</b>
<b>Total</b>	<b>105</b>

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?" [2]



## 5 Erste Schritte

Um mit dem Projekt zu beginnen, kann folgendermassen vorgegangen 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 verschiedenen Blöcken.
- Testen der Lösung auf der Platine und finden etwaiger Fehler .

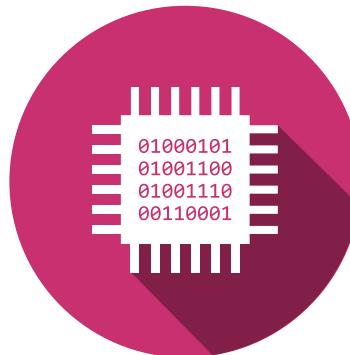
### 5.1 Tips

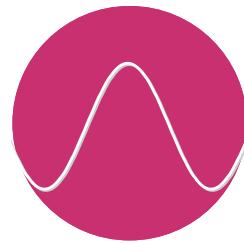
Anbei noch einige zusätzlichen Tips um Probleme und Zeitverlust zu vermeiden:

- Teilt das Problem in verschiedene Blöcke auf, benutzt hierzu das leere Toplevel Dokument (*inverter-toplevel.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)" [5]
- Es wird empfohlen das System in mehreren Schritten zu realisieren.
  - Berechnen Sie die verschiedenen Frequenzen und Größen von Zählern.
  - Beginnen Sie mit den vorgelagerten Blöcken, die die grundlegenden Funktionen ausführen.
  - Fügen Sie dann die erweiterten Funktionen hinzu, wie z. B. die Verwaltung von Totzeiten.



Vergesst nicht Spass zu haben 😊.





## Literatur

- [1] Amand Axel. *Schematic: FPGA-EBS3 v1.0*. 2023.
- [2] Christophe Bianchi, François Corthay und Silvan Zahno. *Wie Verfasst Man Einen Projektbericht?* 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 und Christophe Bianchi. *Methodologie Für Die Entwicklung von Digitalen Schaltungen*. 2021.
- [6] Infineon. *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.

## Akronyme

**AC** Alternative Current. [3](#)

**DC** Direct Current. [3](#)

**FPGA** Field Programmable Gates Array. [4](#), [6](#), [10](#), [11](#)

**LCD** Liquid Crystal Display. [2](#), [11](#)

**LED** Light Emitting Diodes. [4](#), [6](#), [11](#)

**MOSFET** metal–oxide–semiconductor field-effect transistor. [7](#)

**PCB** Printed Circuit Board. [6–10](#)

**PWM** Pulse Width Modulation. [3](#), [4](#), [6–9](#)

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

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