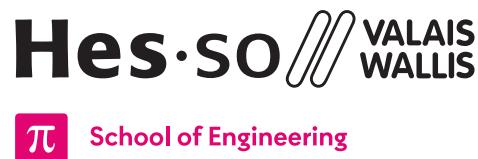


## Cursor (Cur)

### Lecture Digital Design (DiD)



**Orientation:** Information and Communication Technology (ISC)  
**Specialisation:** Data Engineering (DE)  
**Course:** Digital Design (DiD)  
**Authors:** Silvan Zahno, Axel Amand, François Corthay, Christophe Bianchi  
**Date:** 13.03.2025  
**Version:** v3.0



# Contents

1	Introduction .....	3
2	Specification .....	4
2.1	Functions .....	4
2.2	Circuit .....	4
2.3	Scenario (example) .....	6
2.4	HDL-Designer Project .....	7
3	Components .....	8
3.1	Carriage .....	8
3.2	Motor control circuit .....	8
3.2.1	Direct current motor .....	9
3.3	Encoder .....	10
3.4	Reed relay .....	10
3.5	FPGA board .....	10
3.6	Buttons and LEDs .....	12
4	Evaluation .....	13
5	First steps .....	14
5.1	Tips .....	14
	Glossary .....	15



# 1 | Introduction

The goal of this project is to apply the acquired knowledge in a practical example at the end of the semester. It involves controlling a DC motor to precisely move a carriage along a lead screw to predefined positions. This positioning system can be seen in [Figure 1](#).

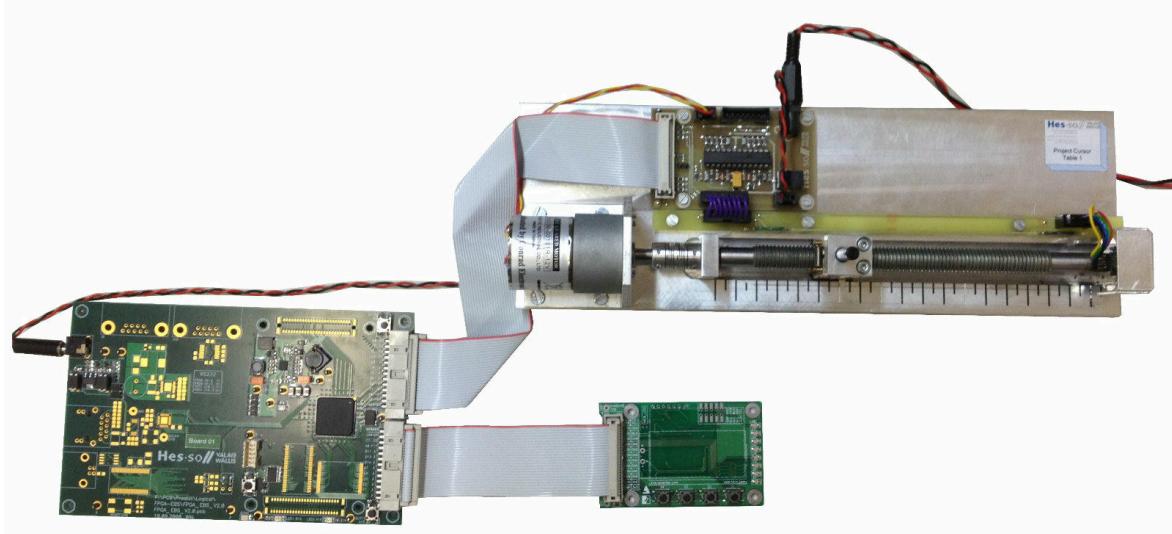


Figure 1 - Hardware setup Cursor (EBS2)

The minimum specifications (see [Section 2](#)) must be met, but students are encouraged to implement additional features. For example, an [Liquid Crystal Display \(LCD\)](#) screen can be used to display various information.



Implementing extra features will provide additional points in the final evaluation.



## 2 | Specification

### 2.1 Functions

The basic functions are defined as follows:

- When the **restart** button is pressed, the cursor moves to the start position indicated by a reed relay (Section 3.4) located near the DC motor (Section 3.2.1).
- When the Position<sub>1</sub> button is pressed, the cursor must first accelerate smoothly to position 1 ( $p_1$ ), then move at full speed and finally decelerate smoothly to stop at position 1 ( $p_1$ ). This can be done from the start position or from position 2, see Figure 2.
- When the Position<sub>2</sub> button is pressed, the cursor must first accelerate smoothly to position 2 ( $p_2$ ), then move at full speed and finally decelerate smoothly to stop at position 2 ( $p_2$ ), see Figure 2.

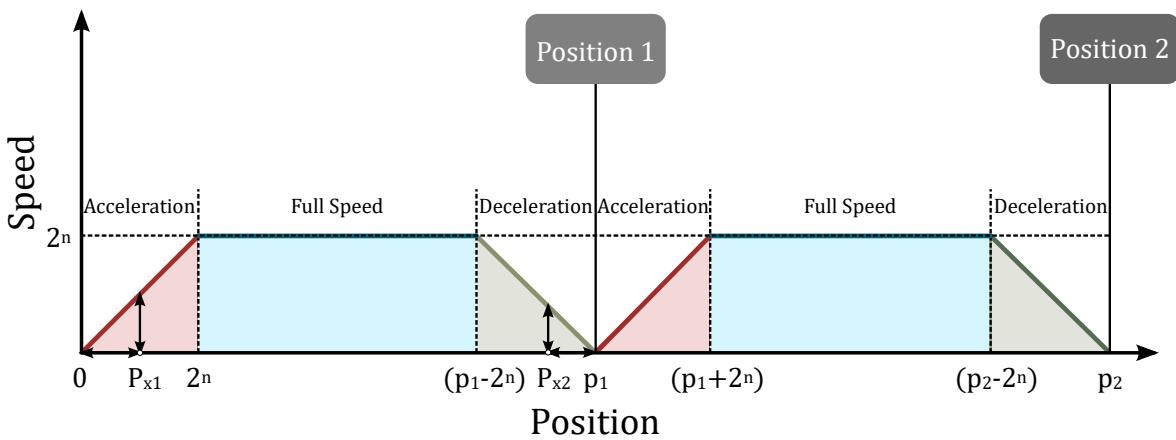


Figure 2 - Cursor speed diagram

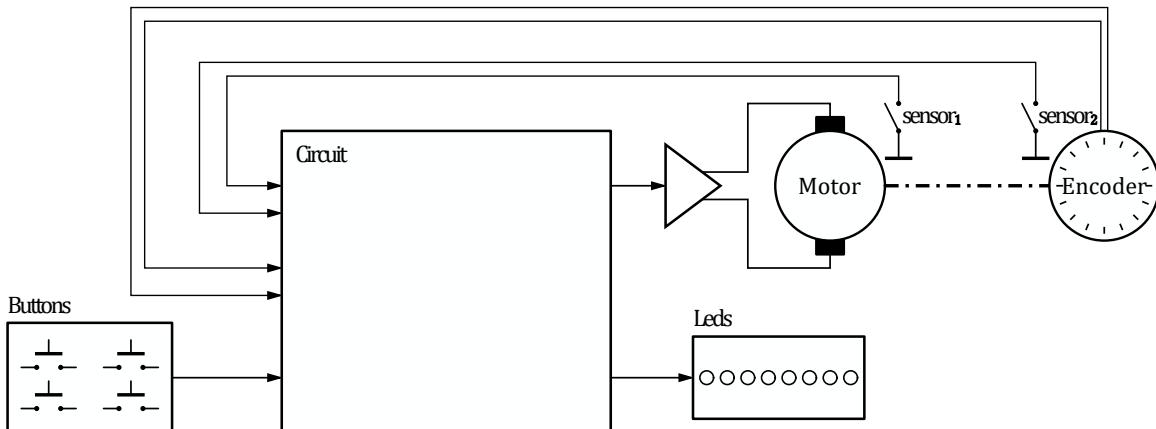


Figure 3 - Cursor circuit

### 2.2 Circuit

The circuit works as follows:

- The DC motor (Section 3.2.1) is controlled by the three signals  $\text{motor}_{\text{On}}$ ,  $\text{side}_1$ ,  $\text{side}_2$ . Its speed is controlled by a **Pulse Width Modulation (PWM)** modulation applied to the signals  $\text{side}_1$  or  $\text{side}_2$ .



- Two reed relays ([Section 3.4](#)) are placed at the ends of the rail [1]. They detect the presence of the cursor carriage (sensor<sub>1</sub> as well as sensor<sub>2</sub>).
- The encoder ([Section 3.3](#)) is used to track, respectively count, the position of the cursor. Its three outputs, encoder<sub>A</sub>, encoder<sub>B</sub> and encoder<sub>I</sub>, allow to track the movements of the lead screw.
- Three buttons are used to control the system: restart, go<sub>1</sub> and go<sub>2</sub>. An additional button, button<sub>4</sub>, can be used for optional functions.
- The testOut pins can be used to output additional information from the system, for example for debugging or to control the [Light Emitting Diodes \(LEDs\)](#).

The empty toplevel design ([cursor-toplevel-empty.pdf](#)) shows all signals connected to the [Field Programmable Gate Array \(FPGA\)](#) board [Figure 4](#).



Figure 4 - Empty Toplevel Circuit



## 2.3 Scenario (example)

In [Figure 5](#), three different scenarios are presented. First, the restart button is pressed and the carriage moves at full speed to the start position ( $\text{sensor}_1$ ). The other two scenarios  $\text{go}_2$  and  $\text{go}_1$  move the carriage to the position $_2$  and position $_2$  respectively, a variable **PWM** signal is applied to the signals  $\text{side}_2$  and  $\text{side}_1$  to accelerate and decelerate the carriage.

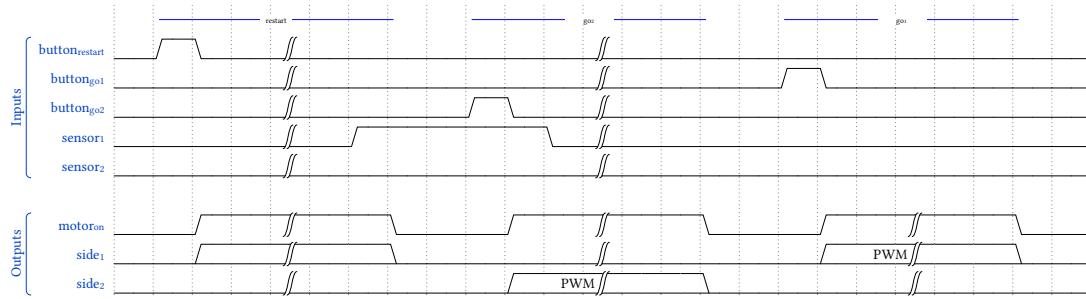


Figure 5 - Scenario cursor



The above scenarios are examples, it is up to the students to complete them.



## 2.4 HDL-Designer Project

A predefined HDL-Designer project can be downloaded or cloned from [Cyberlearn](#) or [Github](#). The file structure of the project looks as follows:

```
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
```



The path of the project folder must not contain spaces.



In the project folder **doc/** many important information can be found. Datasheets, project evaluation as well as help documents for HDL-Designer to name just a few.



# 3 | Components

The system consists of 3 different hardware boards, visible in the [Figure 1](#).

- A carriage assembly with a [Printed Circuit Board \(PCB\)](#) board that controls the motor and reads the sensors ([Figure 6](#))
- A [FPGA](#) development board ([Figure 13](#) or [Figure 14](#))
- A control board with 4 buttons and 8 [LEDs](#) ([Figure 15](#))

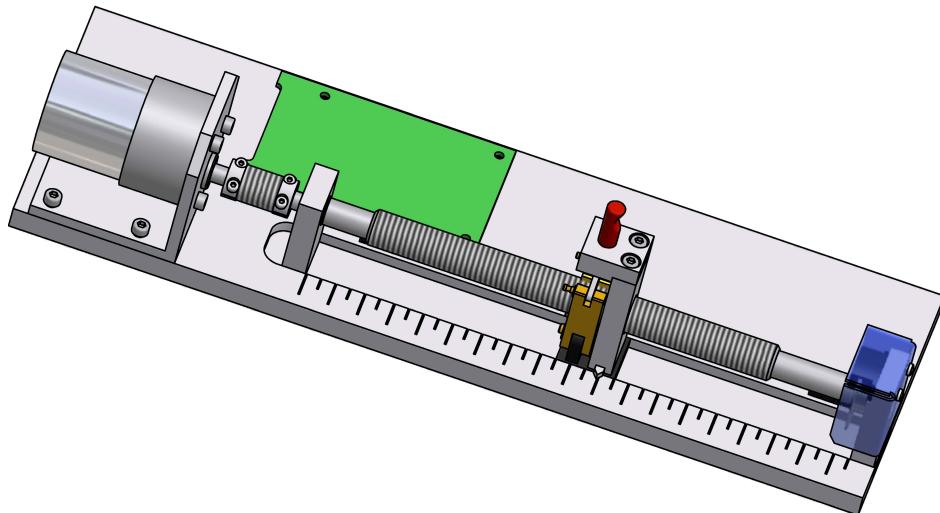


Figure 6 - Cursor carriage assembly

## 3.1 Carriage

The carriage structure includes the DC motor, the two reed relays [Section 3.4](#) as well as the carriage and the lead screw. The thread of the lead screw has a size of M12x1.75, meaning that a distance of 1.75mm is covered per turn ([Figure 7](#)).

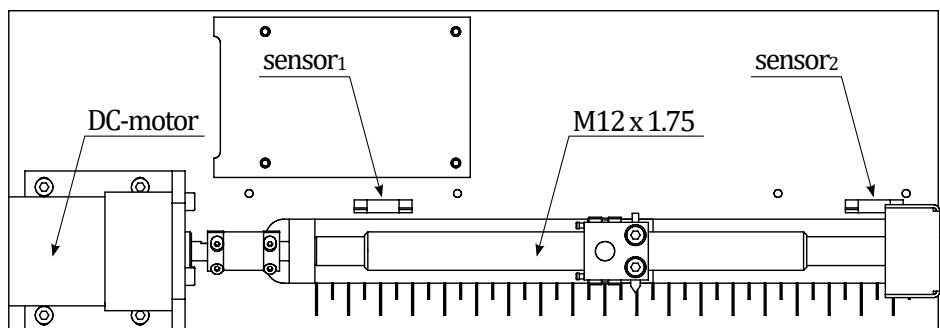


Figure 7 - Detailed assembly of the cursor carriage

## 3.2 Motor control circuit

The DC motor of the carriage is powered by 12V. The power board has an H-bridge that is controlled by digital signals. On the power board, a 5V regulator generates the voltage powering the [FPGA](#) board [\[2\]](#).



### 3.2.1 Direct current motor

The DC motor is controlled by an H-bridge driver L6207 [3], see figure [Figure 8](#). The maximum switching frequency of the H-bridge is 100kHz. This should be taken into account when creating the [PWM](#) signal.

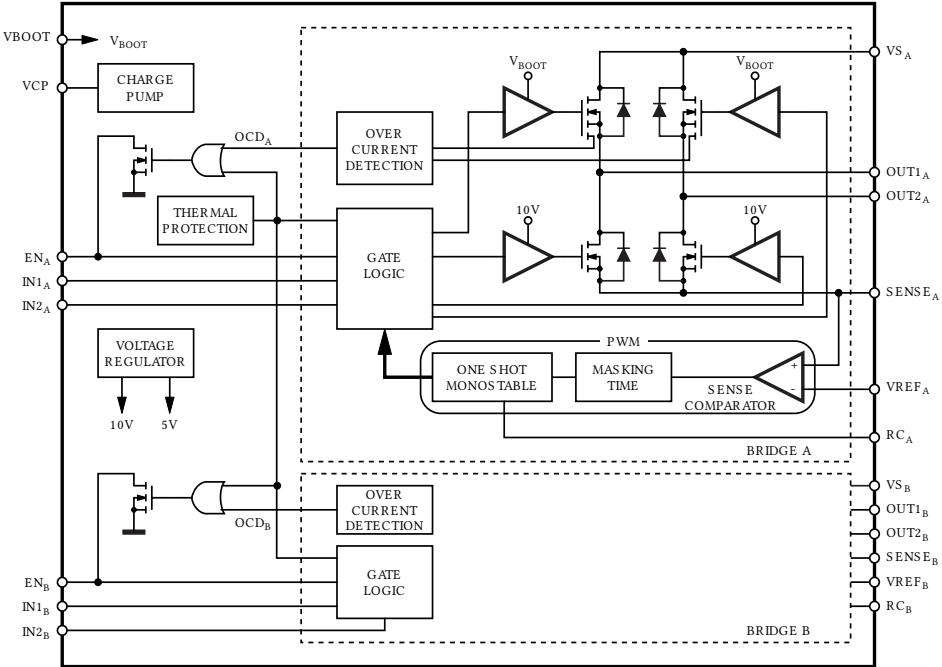


Figure 8 - Block diagram of the L6207N H-bridge circuit [3]

To adjust the speed of the DC motor, a [PWM](#) signal must be applied to the side<sub>1</sub> or side<sub>2</sub> signals, the maximum frequency being 100kHz. The longer the voltage is applied to the motor, the faster it turns. In the [Figure 9](#), the motor turns slower with the [green signal](#) than with the [blue signal](#) and than with the [red signal](#).

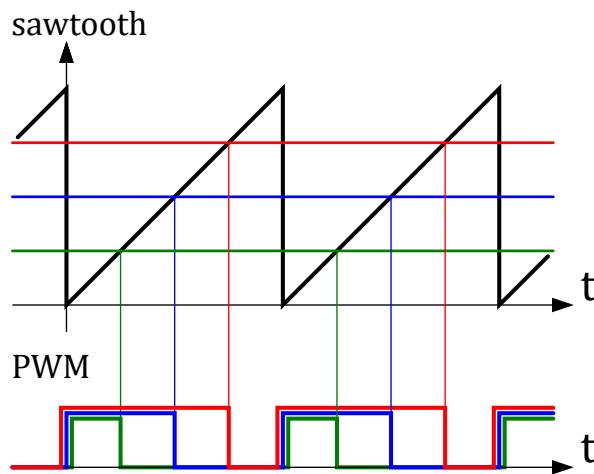


Figure 9 - [PWM](#) signals



### 3.3 Encoder

The angle of the lead screw can be measured using an [incremental encoder](#). The model used in the assembly is an AEDB-9140-A12 [4] ([Figure 11](#)) with 500 [Counts per Revolution \(CPR\)](#) (counts per revolution) per channel, represented in the [Figure 10](#).

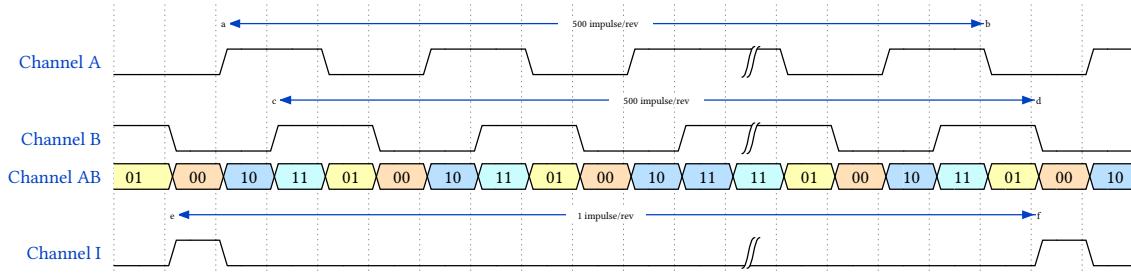


Figure 10 - Signals of incremental encoders



Figure 11 - Encoder AEDB-9140-A12 [4]

### 3.4 Reed relay

The reed relay is a switch that can be switched using electromagnets [1]. When a magnet is near the sensor, the contact closes ([Figure 12](#)). On the module, 2 reed relays are used ( $\text{sensor}_1$  and  $\text{sensor}_2$ ) to identify the left and right limits of the carriage.

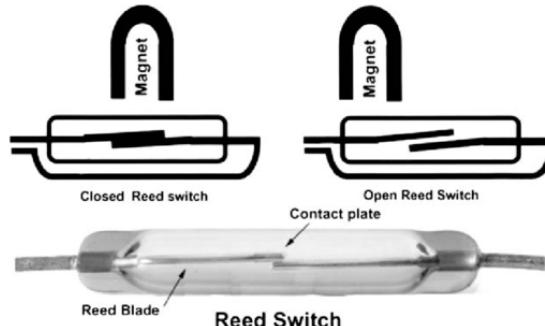


Figure 12 - Reed relay [5]

### 3.5 FPGA board

The main board is the school's FPGA-EBS 2 laboratory development board [6]. It hosts a [Xilinx Spartan xc3s500e FPGA](#) [7], [8] and has many different interfaces ([Universal Asynchronous Receiver Transmitter \(UART\)](#), [Universal Serial Bus \(USB\)](#), Ethernet, etc.). The oscillator used produces a clock signal (**clock**) with a frequency of  $f_{\text{clk}} = 66\text{MHz}$  [9].

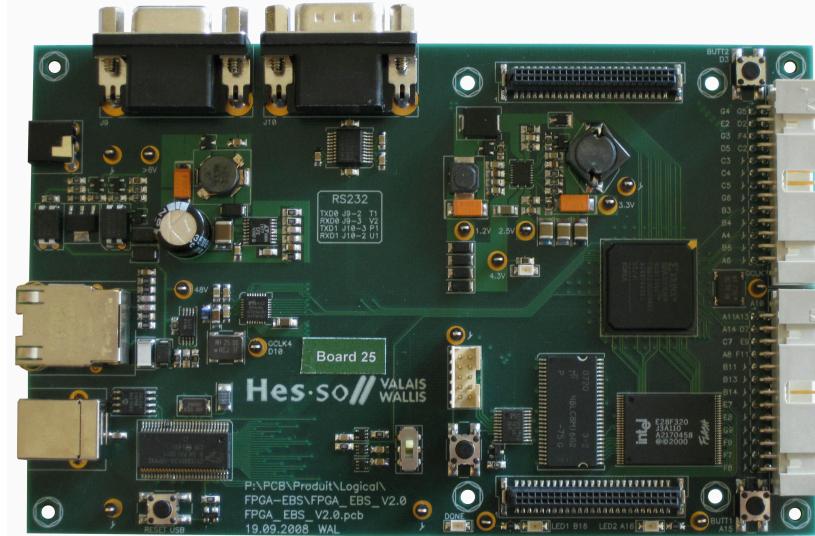


Figure 13 - FPGA board [6]

On the EBS3 board, the oscillator used produces a clock signal (**clock**) with a frequency of  $f_{clk} = 100\text{MHz}$ , reduced by PLL to  $f_{clk} = 60\text{MHz}$ .

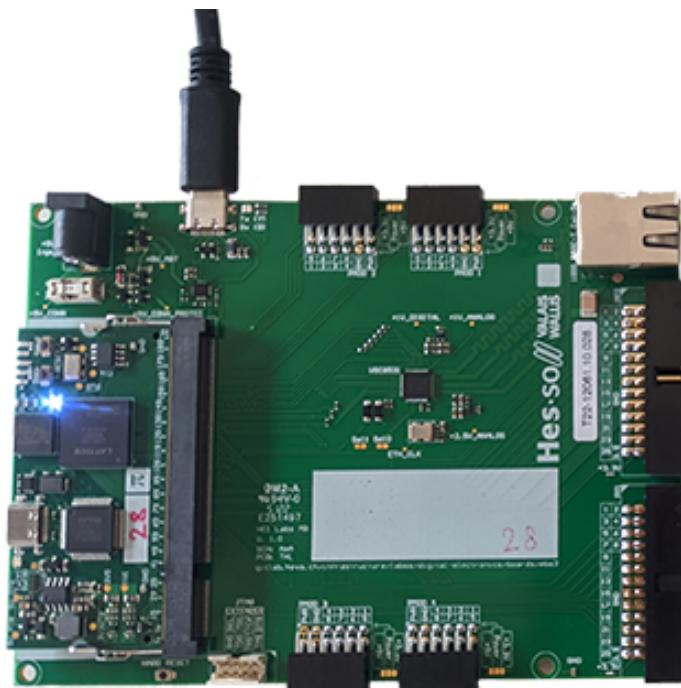


Figure 14 - FPGA board [10]



The simulators are set by default for the EBS3 boards. To change them, open a testbench block **xxx\_tb** and double-click on the **Pre-User** declarations (top left of the page) to modify the **clockFrequency** variable according to the desired clock value.



### 3.6 Buttons and LEDs

The board with the buttons and [LEDs](#) [11] is connected to the [FPGA](#) board. It has 4 buttons and 8 [LEDs](#) that can be used in the design. If desired, this board can be equipped with an [LCD](#) display [12], [13].



Figure 15 - Button-LEDs-LCD board [11]



## 4 | Evaluation

In the **doc/** folder, the file **evaluation-bewertung-cursor.pdf** shows the detailed evaluation scheme, **Table 1**.

The final grade includes the report, the code as well as a presentation of your system.

<b>Evaluated aspects</b>	<b>Points</b>
<b>Report</b>	<b>55</b>
Introduction	3
Specification	5
Project	20
Verification and validation	10
Integration	9
Conclusion	3
Formal aspects of the report	5
<b>Functionality of the circuit</b>	<b>30</b>
<b>Quality of the solution</b>	<b>10</b>
<b>Presentation</b>	<b>10</b>
<b>Total</b>	<b>105</b>

Table 1 - Evaluation grid



The evaluation grid already gives indications about the structure of the report. For a good report, consult the document “How to write a project report” [14].



## 5 | First steps

To start the project, you can proceed as follows:

- Read the specifications and information above carefully.
- Check the hardware and test the pre-programmed program.
- Browse through the documents in the **doc/** folder of your project.
- Develop a detailed functional diagram. You should be able to explain the signals and their functions.
- Implement and simulate the different blocks.
- Test the solution on the printed circuit board and find any errors .

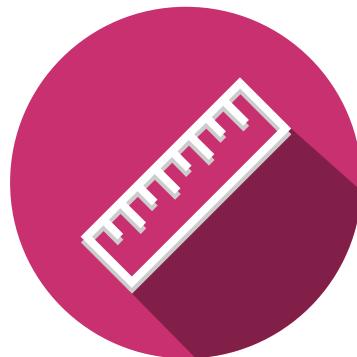
### 5.1 Tips

Here are some additional tips to avoid problems and time loss:

- Divide the problem into different blocks, use the empty Toplevel document (**cursor-toplevel-empty.pdf**) for this. It is recommended to have a balanced mix between the number of components and the size/complexity of the components.
- Analyze the different input and output signals, for this it is recommended to use the data sheets.
- Follow the DiD chapter “Methodology for the development of digital circuits (MET)” when creating the system [15].
- It is recommended to realize the system incrementally, for example:



Don't forget to have fun.





# Glossary

**CPR** – Counts per Revolution [10](#)

**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 [4, 6, 9](#)

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

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



# Bibliography

- [1] “Reed Relay.” Dec. 05, 2020. Accessed: Nov. 24, 2021. [Online]. Available: [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] Olivier Walpen, “Schematic: Cursor Chariot Power Circuit.” 2009.
- [3] STMicroelectronics, “Datasheet: DMOS Dual Full Bridge Driver with PWM Current Controller.” 2003.
- [4] Agilent Technologies, “Datasheet Agilent AEDB-9140 Series Three Channel Optical Incremental Encoder Modules with Codewheel, 100 CPR to 500 CPR.” 2005.
- [5] “Magnetic-Reed-Switch-Above-Closed-and-open-reed-switch-in-response-to-magnet-placement.Png (850×345).” Accessed: Nov. 24, 2021. [Online]. Available: <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>
- [6] Silvan Zahno, “Schematic: FPGA-EBS v2.2.” 2014.
- [7] Xilinx, “Spartan-3 FPGA Family.” Accessed: Nov. 20, 2021. [Online]. Available: <https://www.xilinx.com/products/silicon-devices/fpga/spartan-3.html>
- [8] Xilinx, “Datasheet Spartan-3E FPGA Family.” 2008.
- [9] CTS, “Datasheet CTS Model CB3 & CB3LV HCMOS/TTL Clock Oscillator.” 2006.
- [10] A. Amand and S. Zahno, “FPGA-EBS3 Electornic Technical Documentation.” 2022.
- [11] Silvan Zahno, “Schematic: Parallelport HEB LCD V2.” 2014.
- [12] Sitronix, “Datasheet Sitronix ST7565R 65x1232 Dot Matrix LCD Controller/Driver.” 2006.
- [13] Electronic Assembly, “Datasheet: DOGM Graphics Series 132x32 Dots.” 2005.
- [14] Christophe Bianchi, François Corthay, and Silvan Zahno, “Comment Rédiger Un Rapport de Projet?” 2021.
- [15] François Corthay, Silvan Zahno, and Christophe Bianchi, “Méthodologie de Conception de Circuits Numériques.” 2021.