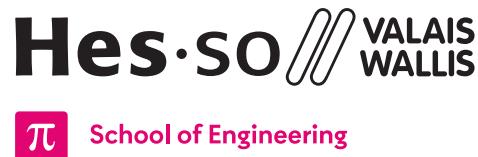




Chronometer (Chrono)

Lecture Digital Design (DiD)



Orientation: Systems Engineering (Synd)

Specialisation: Infotronics (IT)

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)	5
2.4	HDL-Designer Project	7
3	Components	8
3.1	Clock face	8
3.2	Motor control circuit	8
3.2.1	Stepper motor	8
3.3	Reed-Relay	10
3.4	FPGA board	10
3.5	Buttons and LED	11
4	Evaluation	13
5	First steps	14
5.1	Tips	14
	Glossary	15



1 | Introduction

The goal of this project is to apply theoretical knowledge in a practical setting at the end of the semester. The objective is to control a stepper motor to move a clock hand precisely on a dial, simulating a simple chronometer. The system is depicted in [Figure 1](#).

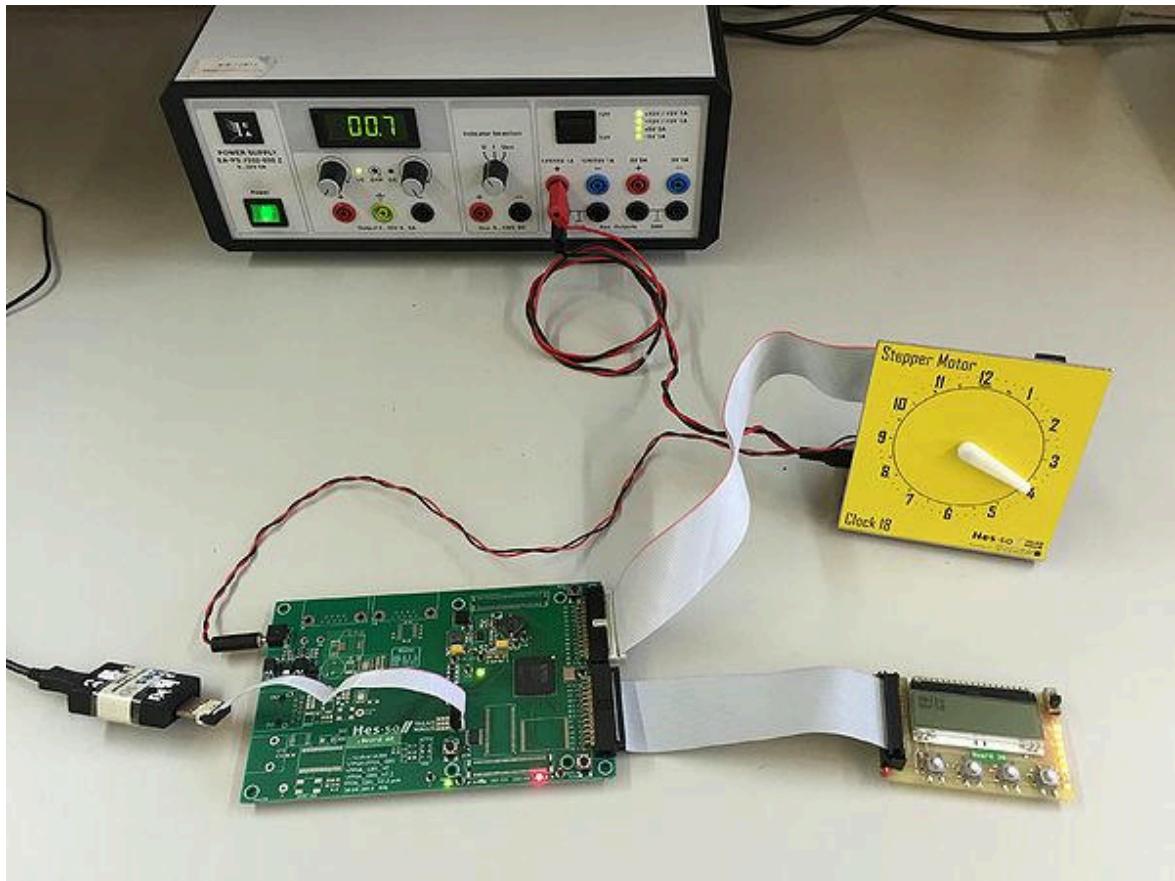


Figure 1 - Hardware setup Chrono (EBS2)

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 fundamental functions of the system are:

- **Restart:** When the restart button is pressed, the hand returns to the 12 o'clock position, detected by a Reed Relay (Section 3.3) near the stepper motor (Section 3.2.1).
- **Start:** Pressing the start button moves the hand by 1/60th of a revolution per second.
- **Stop:** Pressing the stop button halts the hand at its current position.

The basic system does not handle erratic user inputs, such as pressing **restart** while the hand is already at 12 o'clock.

The diagram of the basic functionalities is shown in the [Figure 2](#).

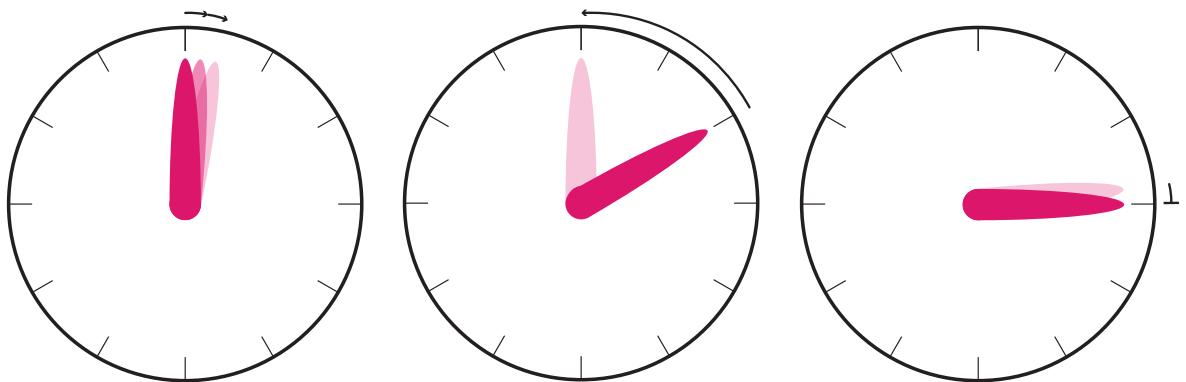


Figure 2 - Diagramme of the basic functionalities. From left to right: **start**, **restart** and **stop**.

2.2 Circuit

The stepper motor controls the hand movement, as shown in [Figure 3](#).

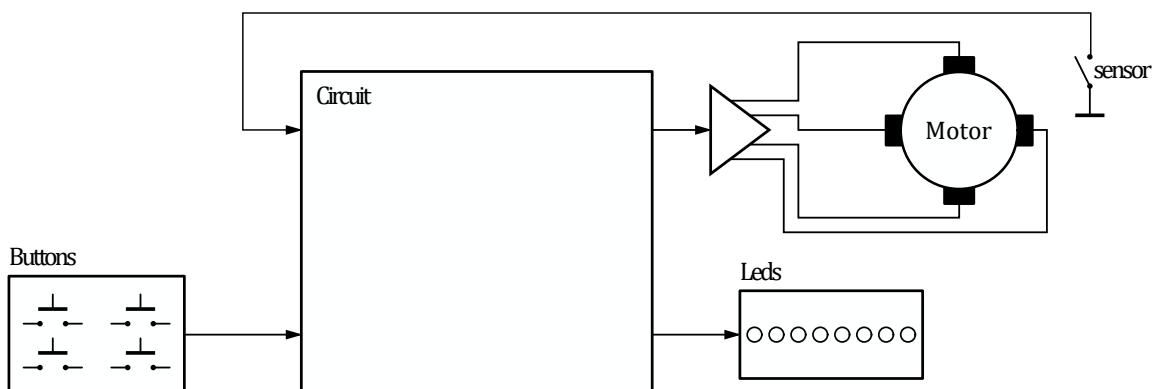


Figure 3 - Circuit of the chrono

The circuit works as follows:

- The stepper motor (Section 3.2.1) is driven by four control signals: **coil1**, **coil2**, **coil3**, and **coil4**.
- A Reed Relay (Section 3.3) at the 12 o'clock position detects when the hand returns to the start.



- Three buttons (**restart**, **start**, **stop**) control the system, with an optional fourth button (**button_4**) for additional features [1].
- **TestOut** pins can be used for debugging or controlling **Light Emitting Diodes (LEDs)**.

Figure 4 shows the toplevel design of the circuit, including all connections to the **Field Programmable Gate Array (FPGA)** board. You can find the file also in the repository (**chrono-toplevel-empty.pdf**).

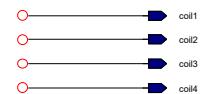
Buttons



12 o'clock Sensor



Stepper Motor Coils



Clock & Reset



Testmode only for simulation



Debug Signal (Leds)

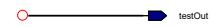


Figure 4 - Empty Toplevel Circuit

2.3 Scenario (example)

In Figure 5 and Figure 6, two different scenarios are presented. First, the **restart** button is pressed and the hand moves at full speed to the initial position (**sensor**). Then by pressing the **start** button, the hand starts turning clockwise. Once in motion, pressing the **stop** button stops the hand at its current position. Pressing **start** or **restart** can then respectively restart the stopwatch or reset it to zero.

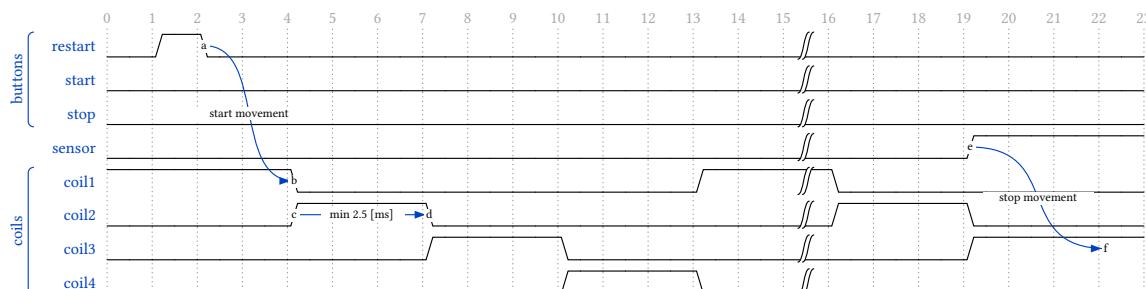


Figure 5 - Chrono Scenario - Restart

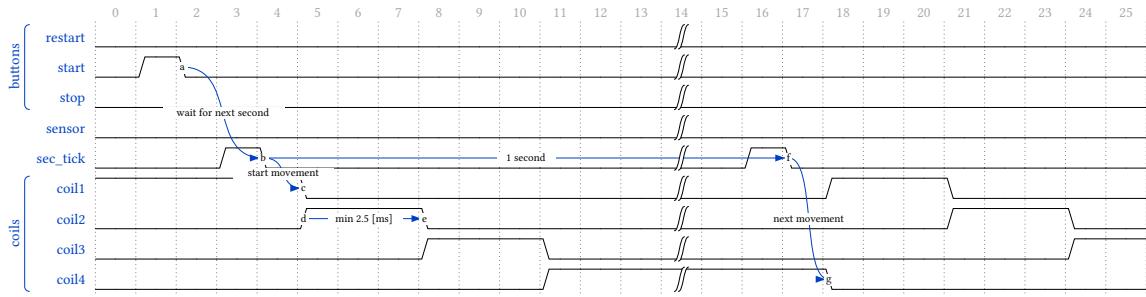


Figure 6 - Chrono Scenario - Start



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 in [Cyberlearn](#) or [Github](#). The file structure of the project is as follows:

```
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
```



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 three different hardware boards, visible in the [Figure 1](#).

- A chrono assembly with a “[Printed Circuit Board \(PCB\)](#)” board that controls the motor and reads the sensor, see [Figure 7](#).
- A [FPGA](#) development board ([Figure 12](#) or [Figure 13](#)).
- A control board with 4 buttons and 8 [LEDs](#) ([Figure 14](#)).

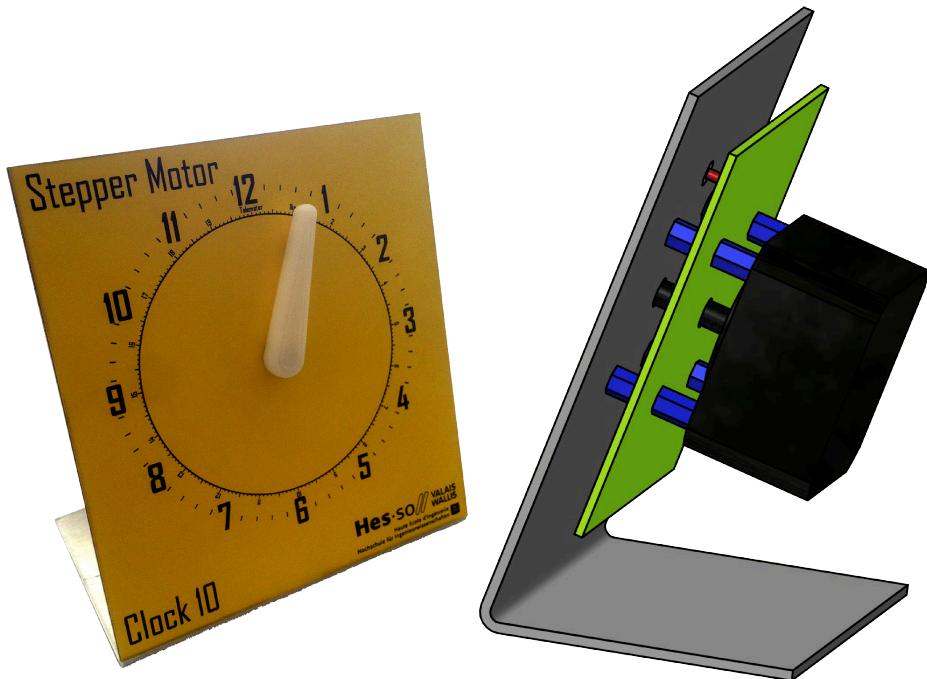


Figure 7 - Assembling the chrono dial

3.1 Clock face

The clock structure consists of the stepper motor ([Section 3.2.1](#)), the Reed Relay ([Section 3.3](#)) and the clock hand.

3.2 Motor control circuit

The chrono stepper motor 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 Stepper motor

The stepper motor has the following characteristics, which can be read in the datasheet [3]:

- 200 steps per revolution
- 8-12V
- 4 phases



The stepper motor is controlled by an H-bridge driver L6207 [4], see [Figure 8](#). The maximum switching frequency of the H-bridge is 100kHz. This must be taken into account when creating the [Pulse Width Modulation \(PWM\)](#) signal.



Experience has shown that the motor can reach a speed of 1-2 revolutions per second.

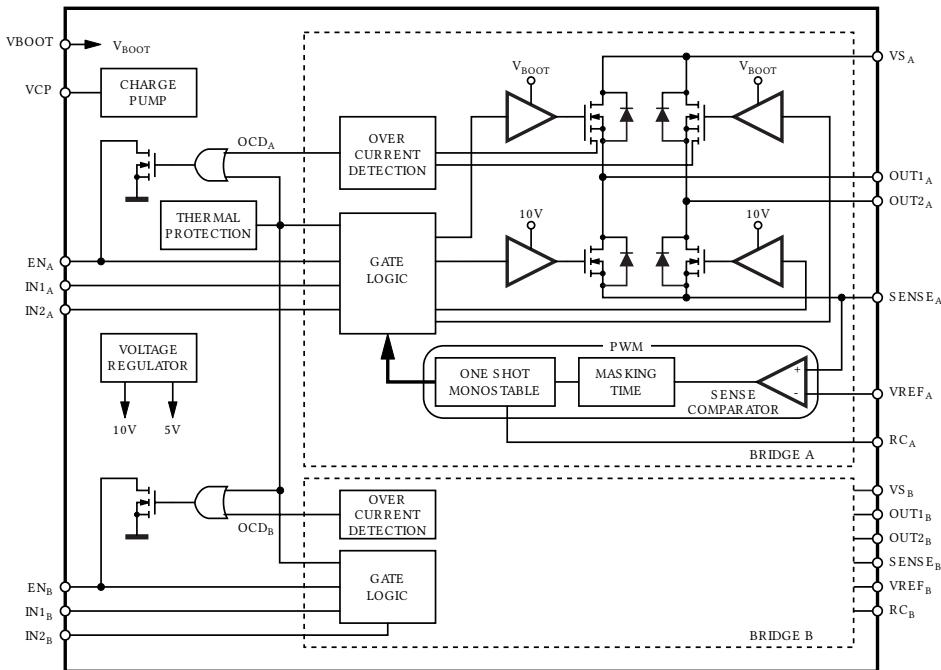


Figure 8 - Block diagram of the L6207N H-bridge circuit [4]

In the example shown in [Figure 9](#), the winding p_1 (controlled by the signal **coil1**) is powered, thus aligning the rotor in this direction (indicated by the black arrow).

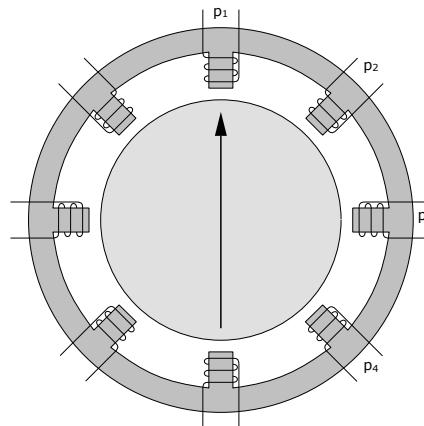


Figure 9 - Block diagram of the stepper motor.



To control the position and speed of the stepper motor, the 4 phases are used to create a magnetic field. The rotor can be modeled as a simple magnet, thus aligning with the position and polarity of the created magnetic field.

By applying successive pulses to the **coil1**, **coil2**, **coil3** and **coil4** signals, a rotating field can thus be created and the rotor will follow this rotation.

The variation of these signals controls the position of the motor. The holding force and power consumption can be controlled using a **PWM** signal (optional task).



Burn / fire hazard! It is important not to stop the motor by keeping 1 (or more) coil continuously powered! The motor may heat up and burn.

3.3 Reed-Relay

The reed relay is a switch that can be switched using magnets [1], [5]. When a magnet is near the sensor, the contact closes, see figure Figure 10. On the clock dial, a relay (**sensor**) is used to indicate the starting position of the hand, at noon. Its position is indicated by the **blue** color in the figure Figure 11 while the magnet is indicated in **magenta**.

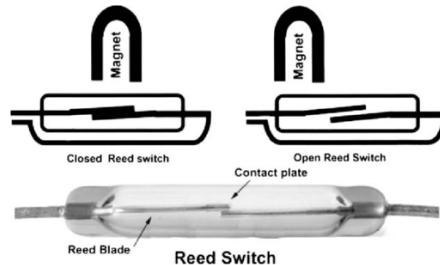


Figure 10 - Reed relais [6]



Figure 11 - Position of the **Reed-Relay** and the **magnet**.

3.4 FPGA board

The main board is the school's FPGA-EBS 2 laboratory development board [7]. It hosts a **Xilinx Spartan xc3s500e FPGA** [8], [9] 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 $f_{\text{clk}} = 66\text{MHz}$ for the EBS2 board [10].

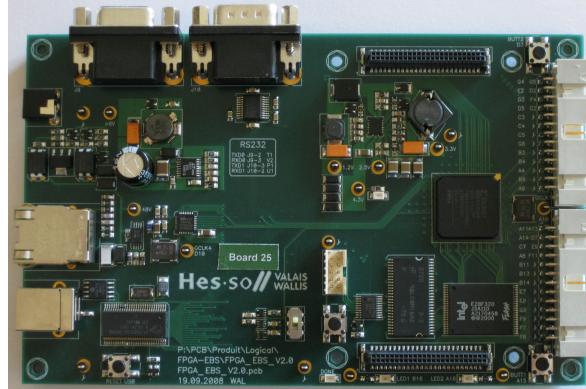


Figure 12 - EBS2 FPGA board [7]

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

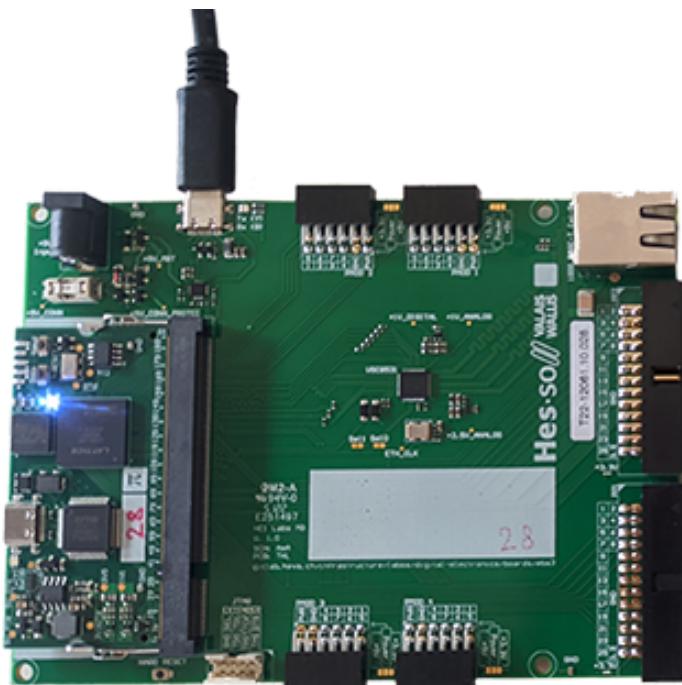


Figure 13 - EBS3 FPGA board [11]



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 change the **clockFrequency** variable to the desired clock value.

3.5 Buttons and LED

The board with the buttons and LEDs [12] 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 [13], [14].



Figure 14 - Button-LEDs-LCD board [12]



4 | Evaluation

In the **doc/** folder, the file **evaluation-bewertung-chrono.pdf** shows the detailed evaluation scheme, [Table 1](#).

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

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

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” [\[15\]](#).



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 (**chrono-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 [16].
- It is recommended to realize the system incrementally, for example:
 - Start by reacting to the buttons and moving the hand
 - Integrate the movement to the second and align it on the dial



Don't forget to have fun.





Glossary

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

LCD – Liquid Crystal Display [3, 11, 12](#)

LED – Light Emitting Diode [5, 8, 11, 12](#)

PCB – Printed Circuit Board [8](#)

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

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
- [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).” 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>
- [7] Silvan Zahno, “Schematic: FPGA-EBS v2.2.” 2014.
- [8] Xilinx, “Spartan-3 FPGA Family.” Accessed: Nov. 20, 2021. [Online]. Available: <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 and S. Zahno, “FPGA-EBS3 Electronic 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, and Silvan Zahno, “Comment Rédiger Un Rapport de Projet?” 2021.
- [16] François Corthay, Silvan Zahno, and Christophe Bianchi, “Méthodologie de Conception de Circuits Numériques.” 2021.