Hardware Software Platforms Project Presentation

Servomotor Control with DE1-SoC

Elora Amorison & Clelia Galvanin
elora.amorison@student.umons.ac.be
Clelia.galvanin@student.umons.ac.be

Contents

- Context and Objectives
- Software and Hardware installation
  - Software to install
  - Board installation
- Tests with the board and LabView
  - Creating a new project
  - Servo-motor
  - Board Control
  - Simulations
  - Results
- Software operation

Contents

- Context and Objectives
- Software and Hardware installation
  - Software to install
  - Board installation
- Tests with the board and LabView
  - Creating a new project
  - Servo-motor
  - Board Control
  - Simulations
  - Results
- Software operation

Context and Objectives

- This project is part of the course Hardware/Software Platforms for students in first master in Electrical Engineering at the Faculty of Engineering of Mons

- Objectives:
  - To handle an entire electronical project
  - To familiar us with processors De-SoC
  - To control a Servomotor
  - To understand the VHDL coding language
  - To create a tutorial for other users

Contents

- Context and Objectives
- Software and Hardware installation
  - Software to install
  - Board installation
- Tests with the board and LabView
  - Creating a new project
  - Servo-motor
  - Board Control
  - Simulations
  - Results
- Software operation

Software to install

- Quartus II
  - http://fpgasoftware.intel.com/?edition=lite
  - To install the software, you'll need both the Quartus software tarfile, and the CycloneV qdz file.
  - Save these both to the same directory.

- ModelSim
  - Select also the "ModelSim-Altera Starter Edition".
  - ModelSim is a HDL Simulator and will be used to simulate signals before using the board.

Hardware to install

- DE1-SoC
  - https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=836

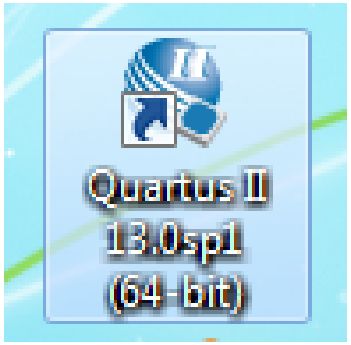
- In the package, you can find the cable to supply the FPGA and an other one to connect to the computer

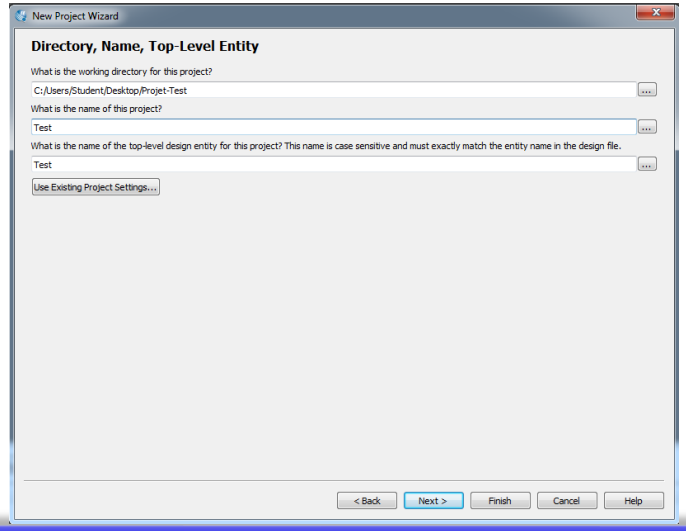
Contents

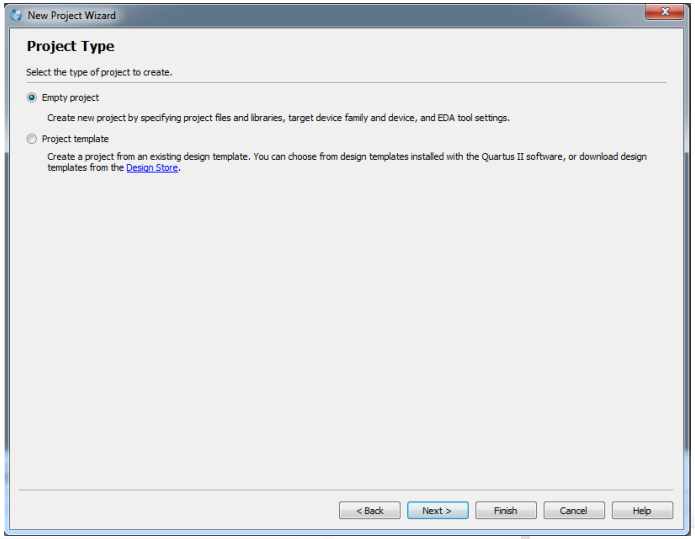
- Context and Objectives
- Software and Hardware installation
  - Software to install
  - Board installation
- Tests with the board and LabView
  - Creating a new project
  - Servo-motor
  - Board Control
  - Simulations
  - Results
- Software operation

Create a new project on Quartus

- Open Quartus II
- Click on “Create a new project”
- A new window will open
  - Select the folder and the title of your project
  - Next
  - Click on “empty project”

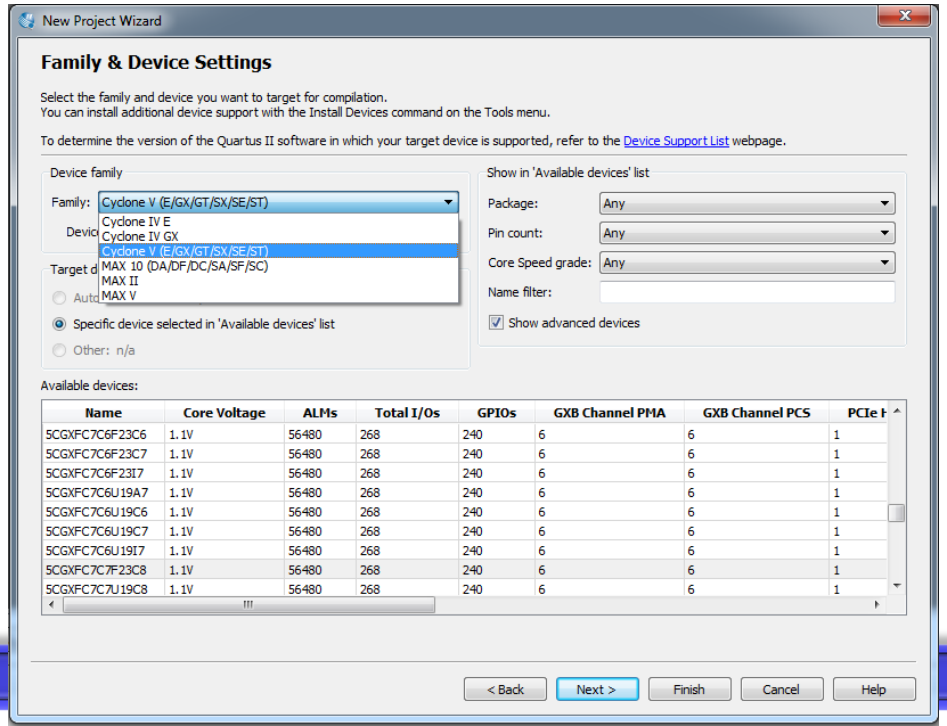






Create a new project on Quartus

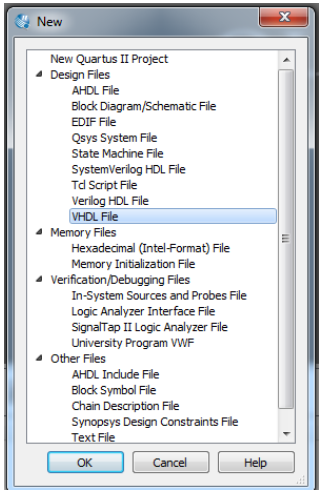
- A window “Family & Device Settings” will open
  - Select the right family of your component and then “next”



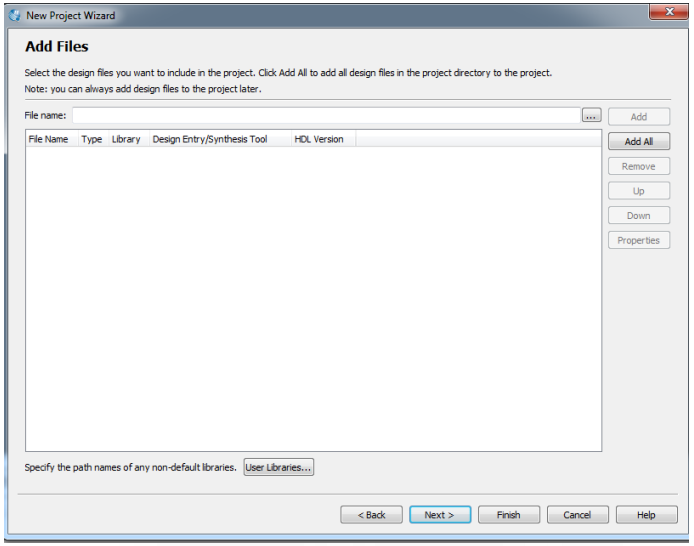
Here, it is a Cyclone V 5CSEMA5F31C6N

Create a new project on Quartus

- Create new file
  - Create a new file at your project and select the VHDL Type File



  - You can also add an existing file at your project



Contents

- Context and Objectives
- Software and Hardware installation
  - Software to install
  - Board installation
- Tests with the board and LabView
  - Creating a new project
  - Servo-motor
  - Board Control
  - Simulations
  - Results
- Software operation

Servomotor

- **What is a servomotor ?**

Electrical device, that rotate a part of its body with high efficiency and great precision to a particular angle between 0 and 180°)

- **How to connect a servomotor?**

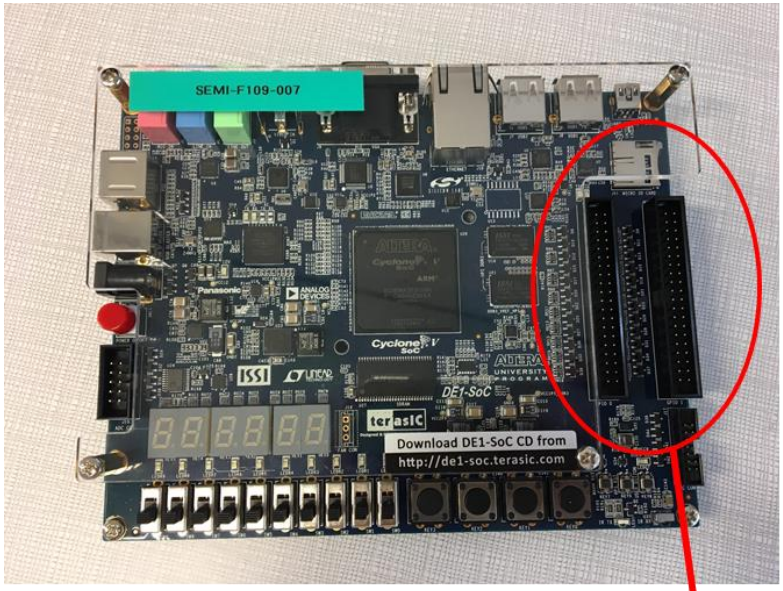
•Three cables

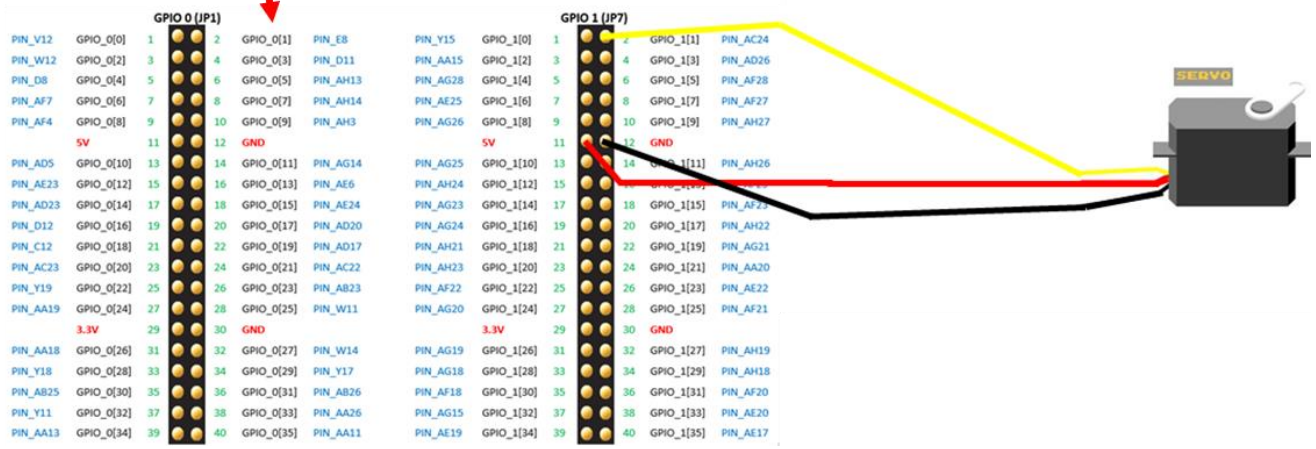
•The yellow wire connected to the pin of the board: GPIO1(1)

•The red wire connected to 5V on the board GPIO1(10)

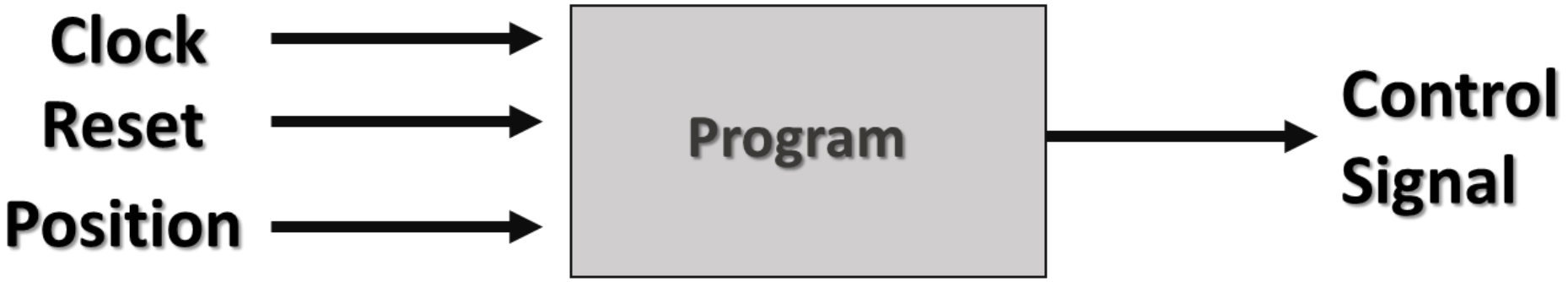
•The brown (or black) wire connected to the ground on the board GPIO1(11)

Servomotor


SEMI-F109-007
DE1-SoC
terasIC
Download DE1-SoC CD from
http://de1-soc.terasic.com

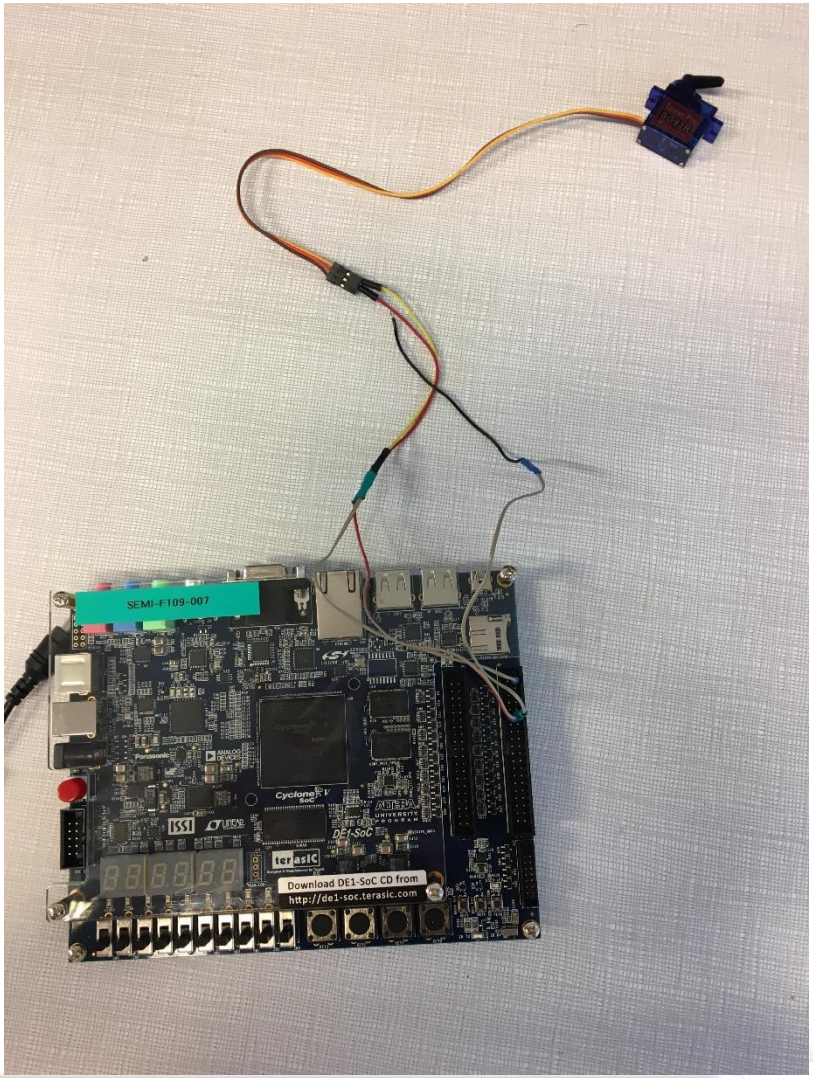

GPIO 0 (JP1)
GPIO 1 (JP7)
SERVO

Bloc Diagram to Control a Servomotor



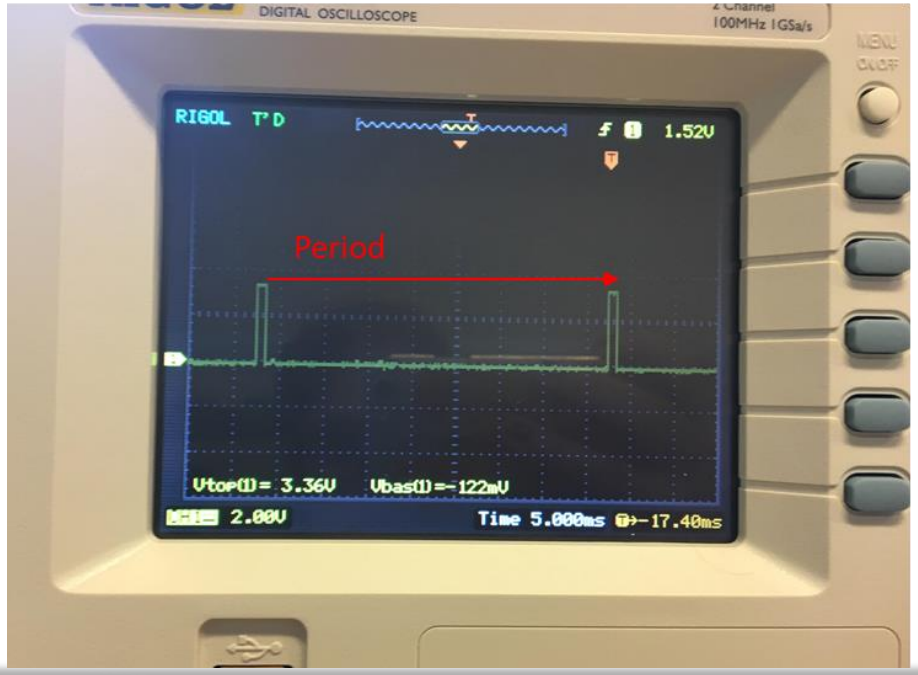
- The values of the position came from the Switch of the board
- The control signal goes to the GPIO1(1) pin and then directly to the servomotor

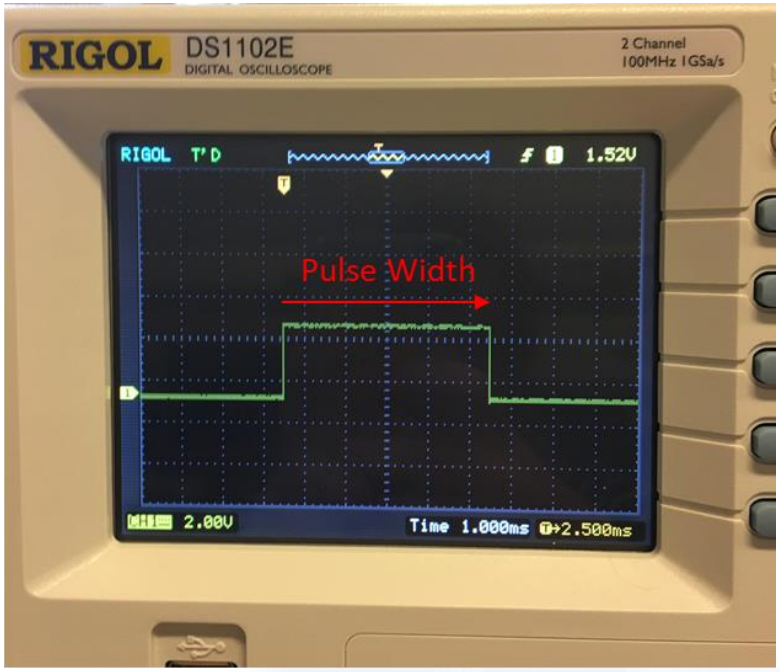
Connection between the board and the motor



Servomotor

- How it works?
- Coding a square wave of different pulse width
- Period of 20 ms
- Minimal value of 0,5ms and maximal value 2,5ms



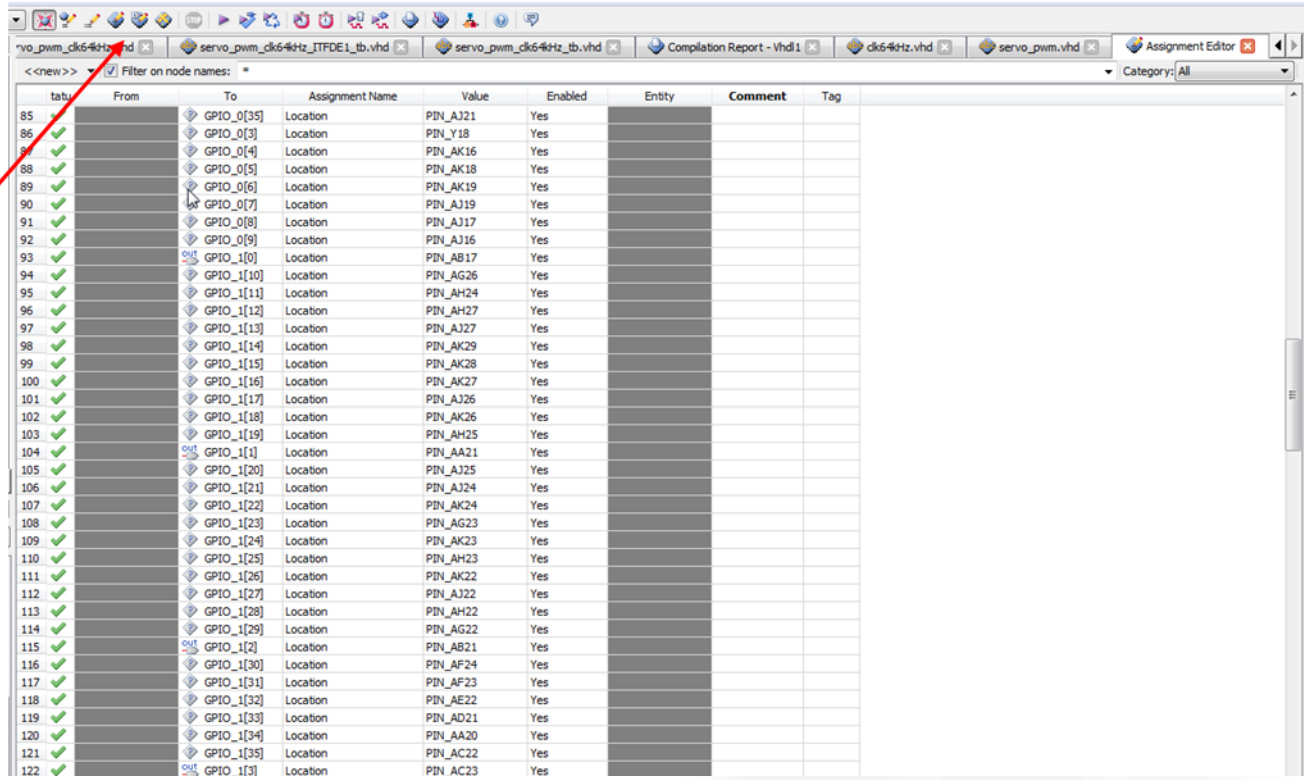


Contents

- Context and Objectives
- Software and Hardware installation
  - Software to install
  - Board installation
- Tests with the board and LabView
  - Creating a new project
  - Servo-motor
  - Board Control
  - Simulations
  - Results
- Software operation

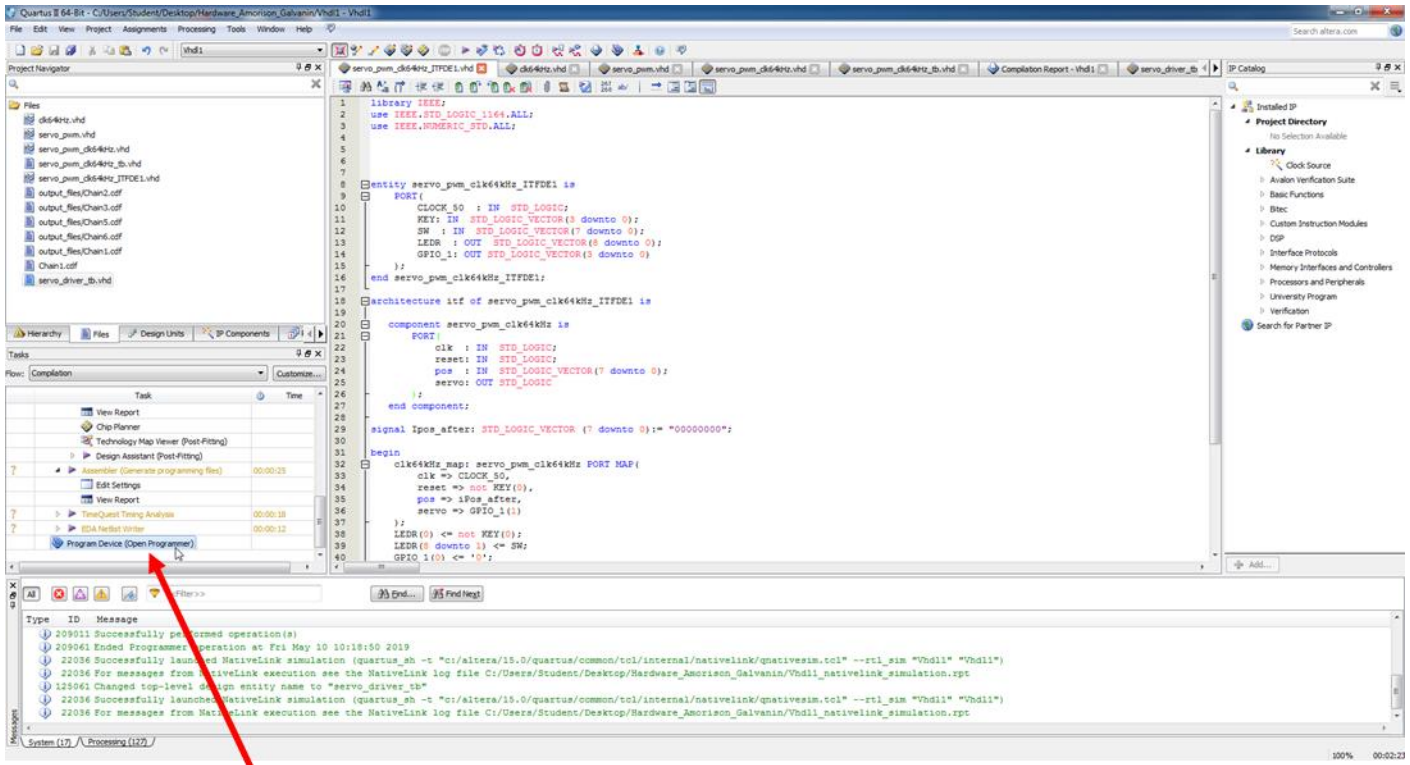
Board Control

- **How to link the hardware and the software?**
- Connect the board to the computer
- Create a file to connect the signals with the pins of the board
- To have a look at all the pins of your board you can click on this icone



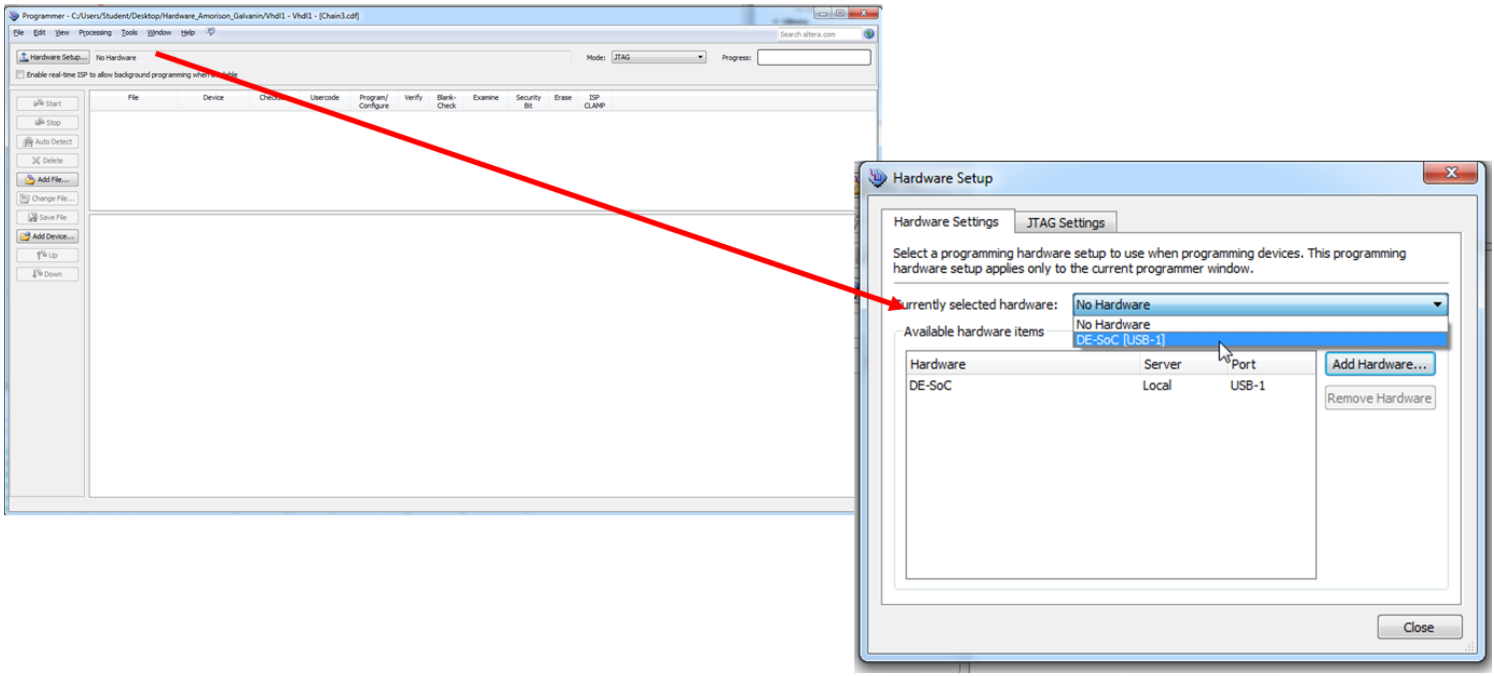
Board Control

- **How to put the code on the processor?**
- Click on the Program Device



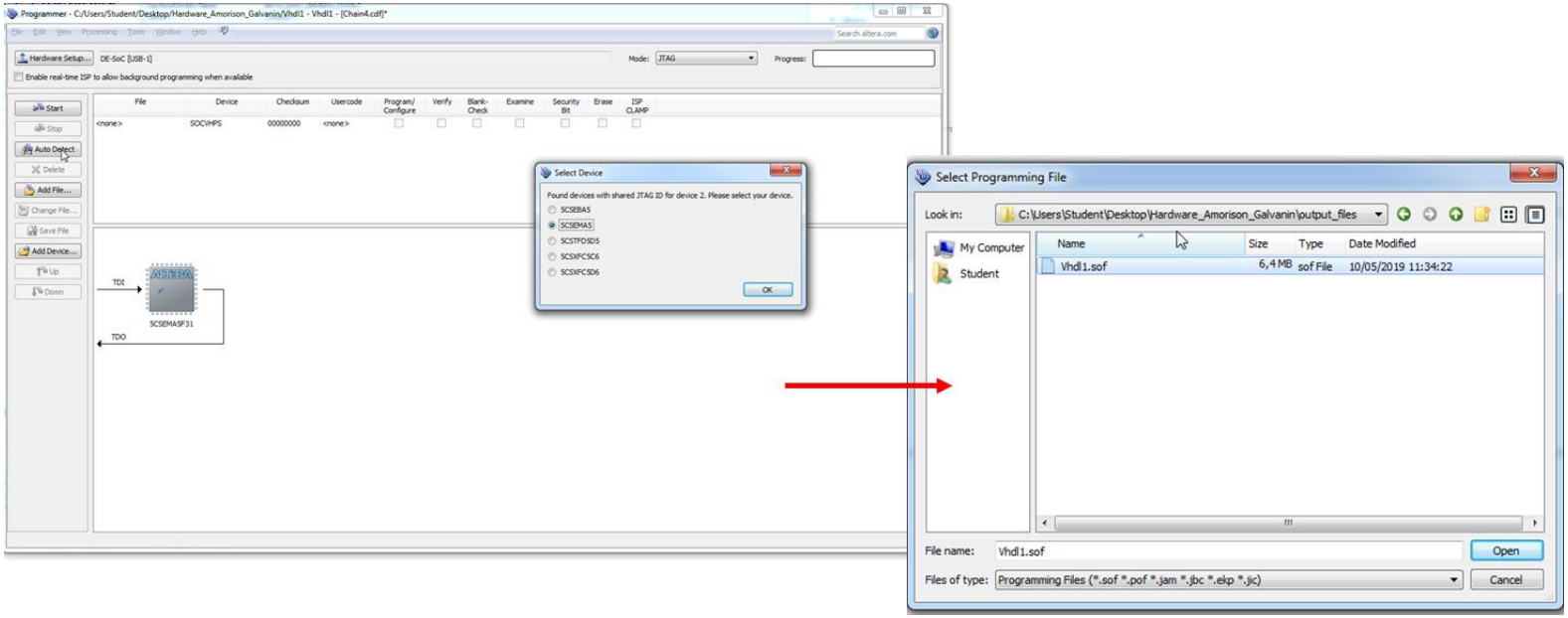
Board Control

- **How to put the code on the processor?**
- A window will open
- Click on “Hardware Setup” and choose your harware



Board Control

- **How to put the code on the processor?**
- Click on “Auto Detect” and choose your correct device
- Then on “Add File” and select your own file



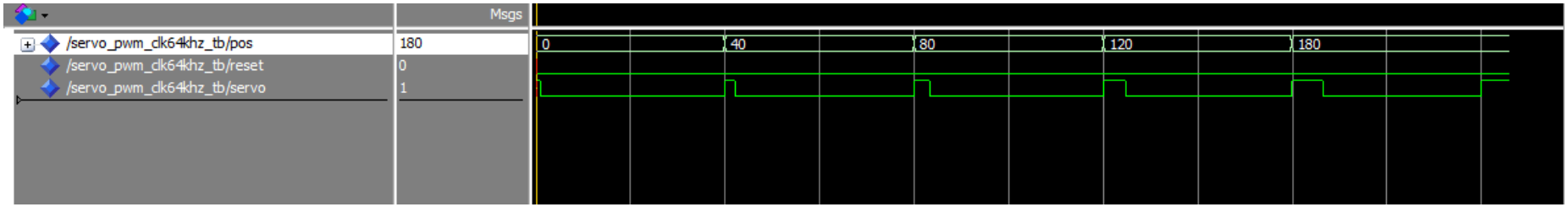
- Finally, click on “Start” and wait for the loading of your code

Contents

- Context and Objectives
- Software and Hardware installation
  - Software to install
  - Board installation
- Tests with the board and LabView
  - Creating a new project
  - Servo-motor
  - Board Control
  - Simulations
  - Results
- Software operation

Simulations

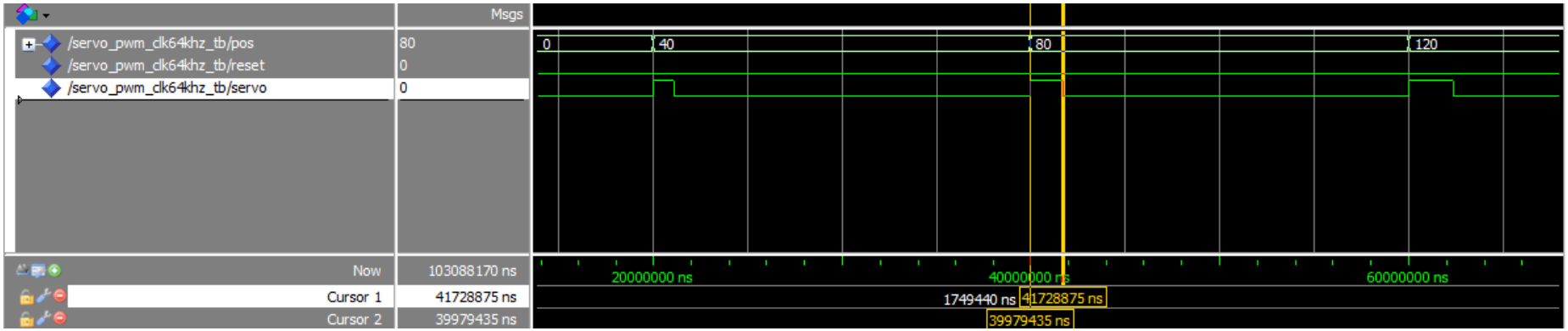
- Driver Simulation



For the simulation, we define some values for the variable « pos ».

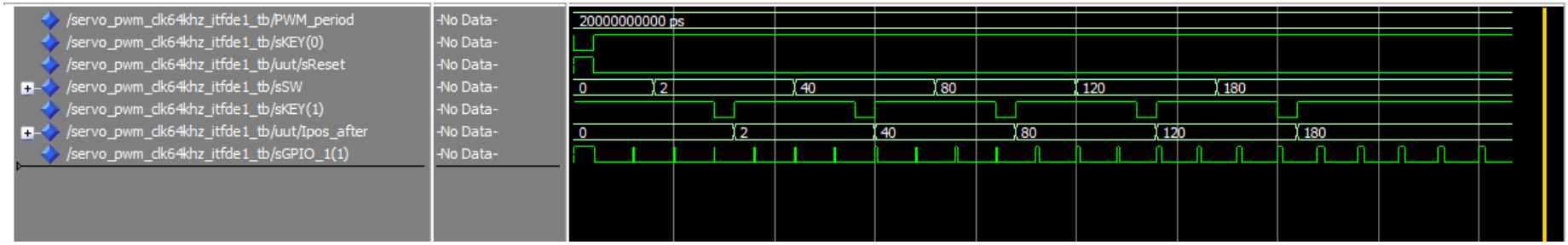
We can see the control signal becoming bigger and bigger.

We can zoom to see the value of PWM and so, the PWM value for an 80° angle is 1749440 ns.



Simulations

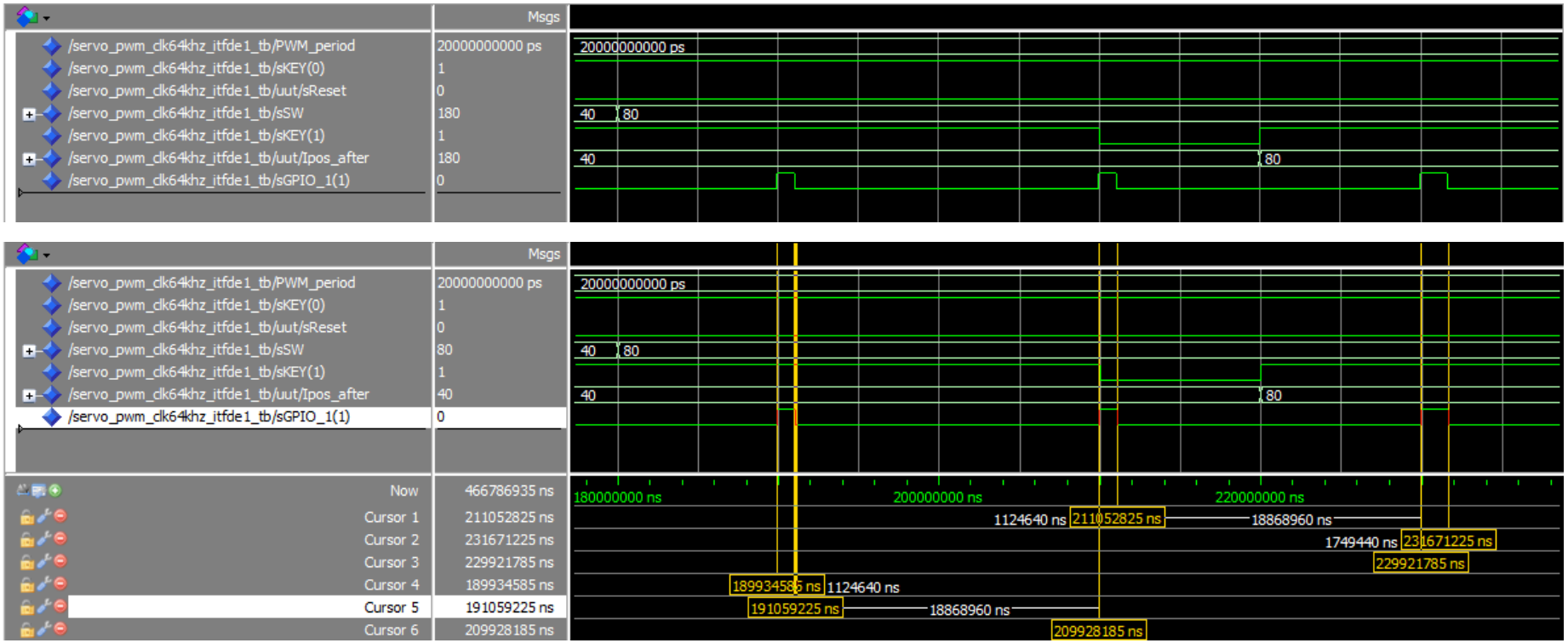
- Driver+Application Simulation



- We change the values of switchs and therefore pos after 3 PWM half period = 3*10ms
- We wait for 3 additional PWM half period before pressing the button.
- We wait for 1 additional PWM half period before releasing the button.
- When the button is released -> switchs values go to Ipos_after.
- PWM value is then modified from the next impulse.
- We notice that the PWM value increases according to the angle.

Simulations

- Driver+Application Simulation



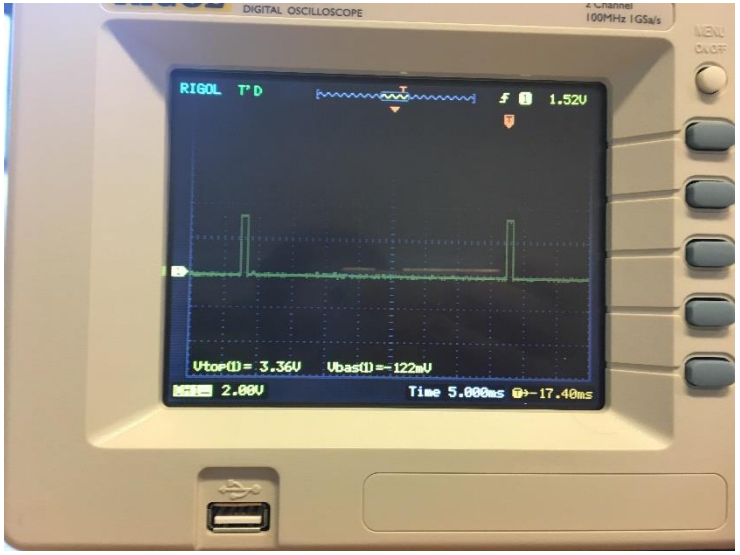
- We make a zoom to measure PWM values and check that the values are growing proportionnaly with the angle.
- Wa can see that the PWM value for an 40° angle is 1 124 640ns and for an 80° angle
- is 1 749 440ns (which is equal to our previous value.)

Contents

- Context and Objectives
- Software and Hardware installation
  - Software to install
  - Board installation
- Tests with the board and LabView
  - Creating a new project
  - Servo-motor
  - Board Control
  - Simulations
  - Results
- Software operation

Results

- **Results on an oscillator**



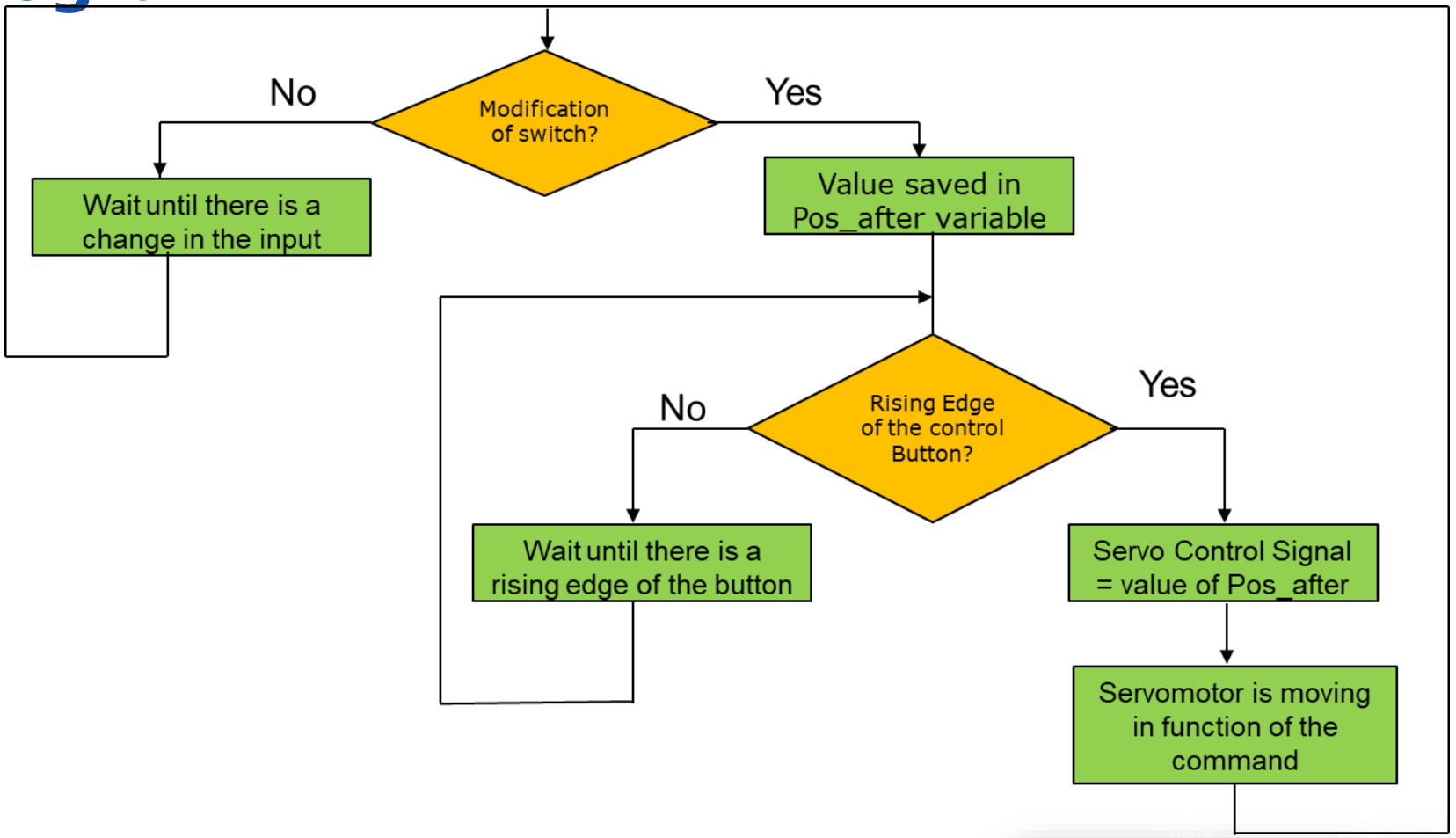
- **Results on the servomotor**
  - ➢See the video

Contents

- Context and Objectives
- Software and Hardware installation
  - Software to install
  - Board installation
- Tests with the board and LabView
  - Creating a new project
  - Servo-motor
  - Board Control
  - Simulations
  - Results
- Software operation

Software Operation

Diagram block of the behaviour of our program



Conclusion

Congrats !

You can now drive a servo-motor with a new processor!

Useful links

- Github:

https://github.com/cleliagal/Control-of-Servo-motor-with-DE1-SoC

- Youtube Video:

https://www.youtube.com/watch?v=HVyE35rkSaQ