# NV HIL Vivado Project

## Introduction

Vivado can create such a big and disgusting set of files for a project whose weigh can made it clearly inoperative for tasks like version control or for sending it.

Included in this folder are all required files necessary to regenerate the whole Vivado project.

The procedure that must be followed relies in different scripts that will regenerate the whole project. The order in the execution of these scripts is critical because some of them are based in the previous ones.

The following software are involved: Vivado 2017.4 and Vivado HLS 2017.4; other versions of the software may require minor adjustments.

## Path assignments

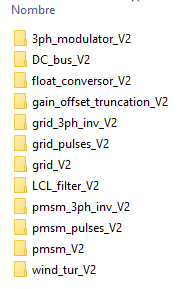
All paths are referred to the main scripts ones. No absolute paths are used during the project.

However, in order to avoid any issue, it is recommended to work in this path: C:\NV\_HIL

## Vivado HLS

The first step of the whole process is to regenerate all the Vivado HLS IPs. Those IPs are the ones in which the HIL emulator equations are included.

Those IPs are:



To generate those IPs, the following process must be done with all of them:

1. Open Vivado HLS command prompt
2. Type: *cd C:\NV\_HIL\Vivado\HW\_repo\HLS\3ph\_modulator\_V2*

Where “C:\NV\_HIL\Vivado\HW\_repo\HLS\3ph\_modulator\_V2” is the path of run.tcl files

1. Type: *Vivado\_hls –f run.tcl*
2. Wait for it until it finishes
3. In case it is wanted Vivado GUI to open, type: *Vivado\_hls –p 3ph\_modulator\_V2.prj*

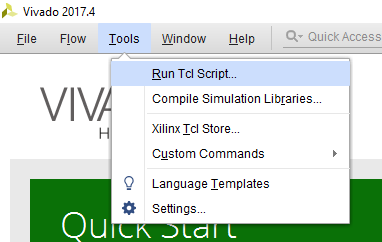
Where 3ph\_modulator\_V2.prj is the folder where the HLS project has been created.

1. Repeat from 1 to 5 with all the HLS folders

## Vivado

To fulfil this step all the Vivado HLS IPs must have been generated.

Open Vivado GUI and click in Tools/Run Tcl Script



Select the file run\_prj.tcl and click run.

run\_prj.tcl file internally only adjusts the initial path of Vivado and then called the file NV\_HIL\_prj.tcl which is the one that contains all hardware information of the project.

Wait until the script finishes - It can take up to one hour.

When finished, Vivado will export the bitstream file and will launch SDK.

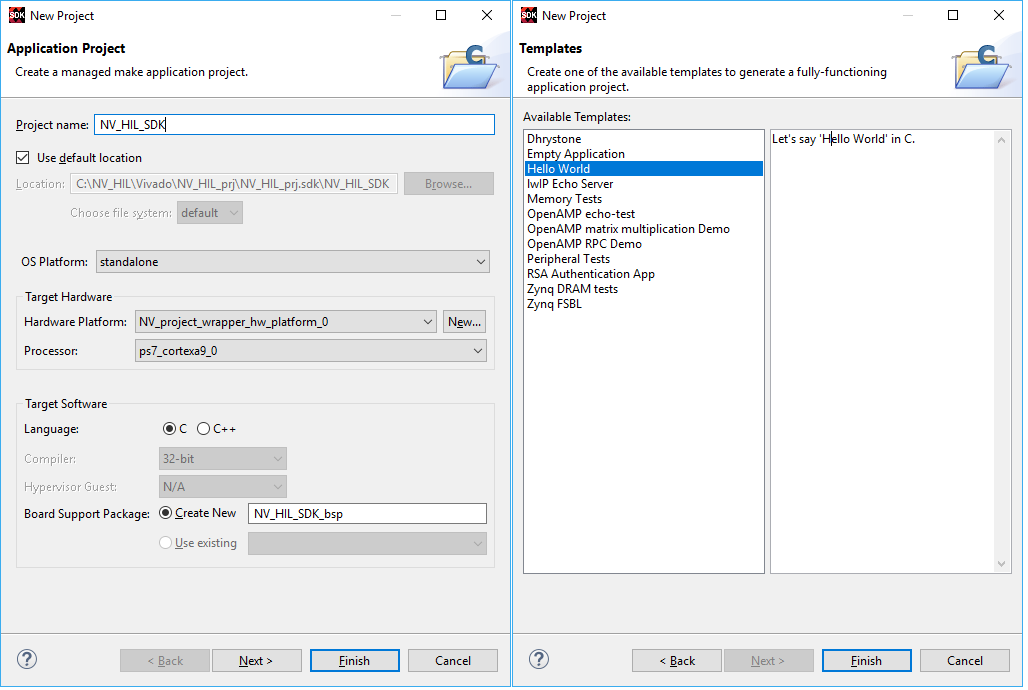
## SDK

SDK will configure the ARM microprocessors.

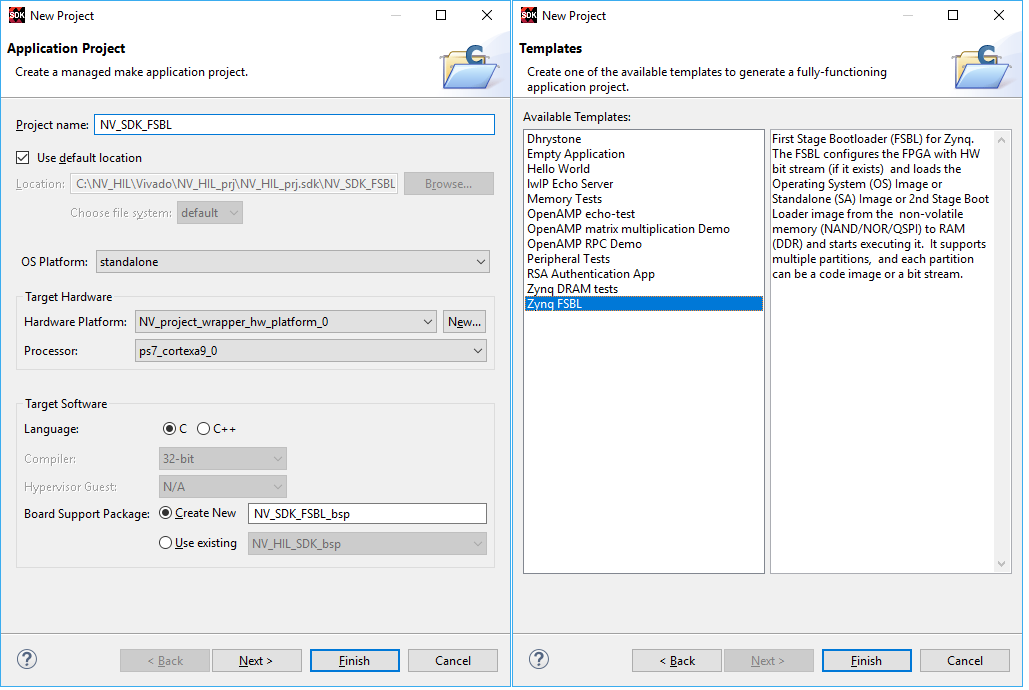
Only one is used and it is in charge of managing the communication with the user desktop console by stablishing a serial communication with 1 second of refresh rate.

To configure SDK, the following steps must be accomplished:

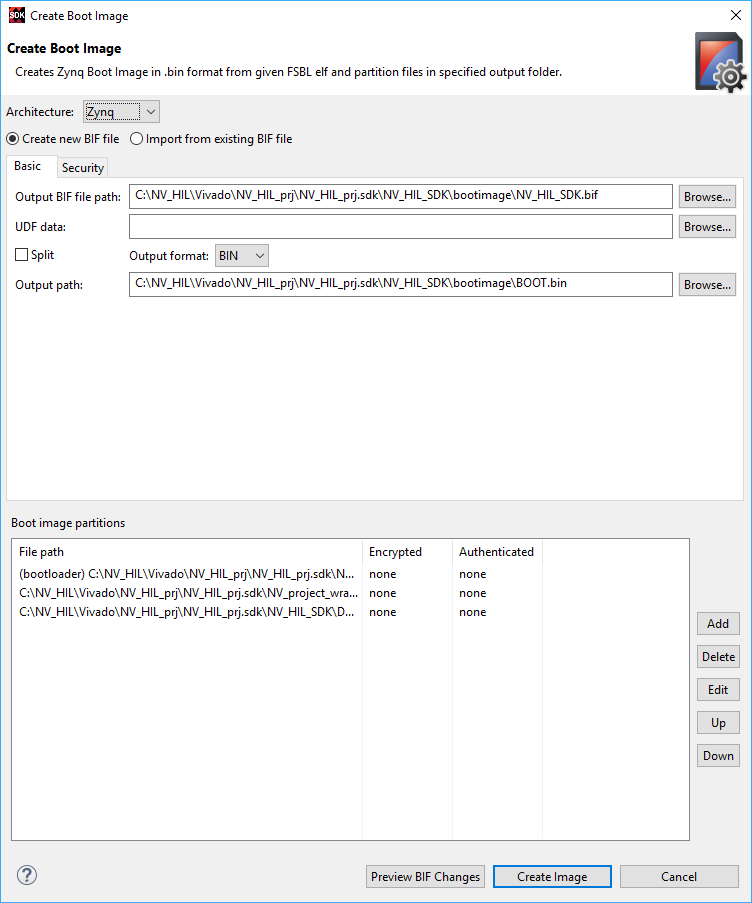
1. Create a new empty project by clicking: File/New/Application Project with the name NV\_HIL\_SDK



1. Copy the contents of provided helloword.cpp and paste it in SDK new project. Be sure to delete all previous code. This file can be found in SW\_repo folder.
2. Create a new empty project called NV\_SDK\_FSBL in order to generate the BOOT.ini file



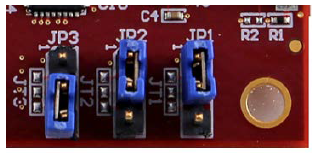
1. Create the BOOT.ini file by selecting in SDK the folder NV\_HIL\_SDK and clicking in Xilinx/Create Boot Image menu.



1. Go to this path in order to foud the BOOT.ini file: C:\NV\_HIL\Vivado\NV\_HIL\_prj\NV\_HIL\_prj.sdk\NV\_HIL\_SDK\bootimage

NOTE: the path may vary.

1. Insert an 8GB microSD card in the computer and format it as FAT32.
2. Paste BOOT.ini file in the microSD card without modifying its name.
3. Remember to configure correctly the microZED board jumpers to boot from microSD card.



## Notes

To avoid the user to do the whole regeneration of the project, the following files are provided as they will probe to be useful:

1. BOOT.bin
2. NV\_project\_wrapper.bit

Both of them are in programming\_files folder