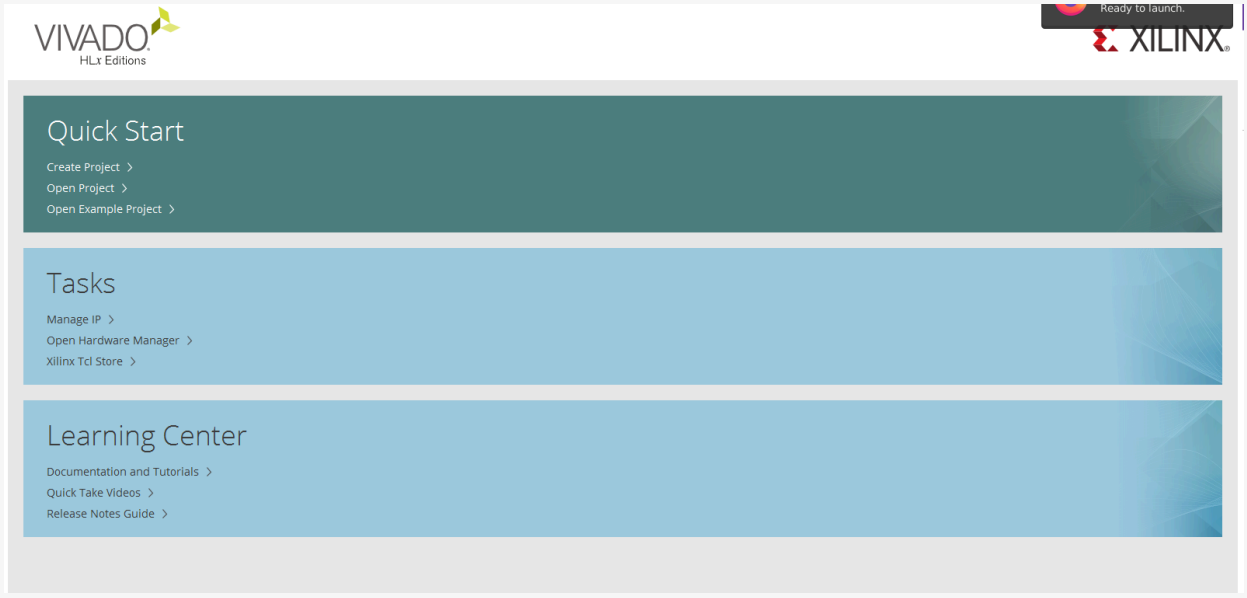
How to Remotely Update the FLX-712 Firmware on Nevis flx-srv-atlas
2024-09-06

The following instructions describe how to remotely reprogram the flx-srv-atlas FLX-712 FPGA, reinitialize the system and test firmware functionality.

1) Start Vivado on flx-srv-atlas:
-log in to the flx-srv-atlas using the atlas-group account:
ssh -XY atlas-group@flx-srv-atlas
(ask for the password)
-initialize and run the vivado installation:
cd ~/software/Xilinx/Vivado/2019.2/
source settings64.sh
./bin/vivado

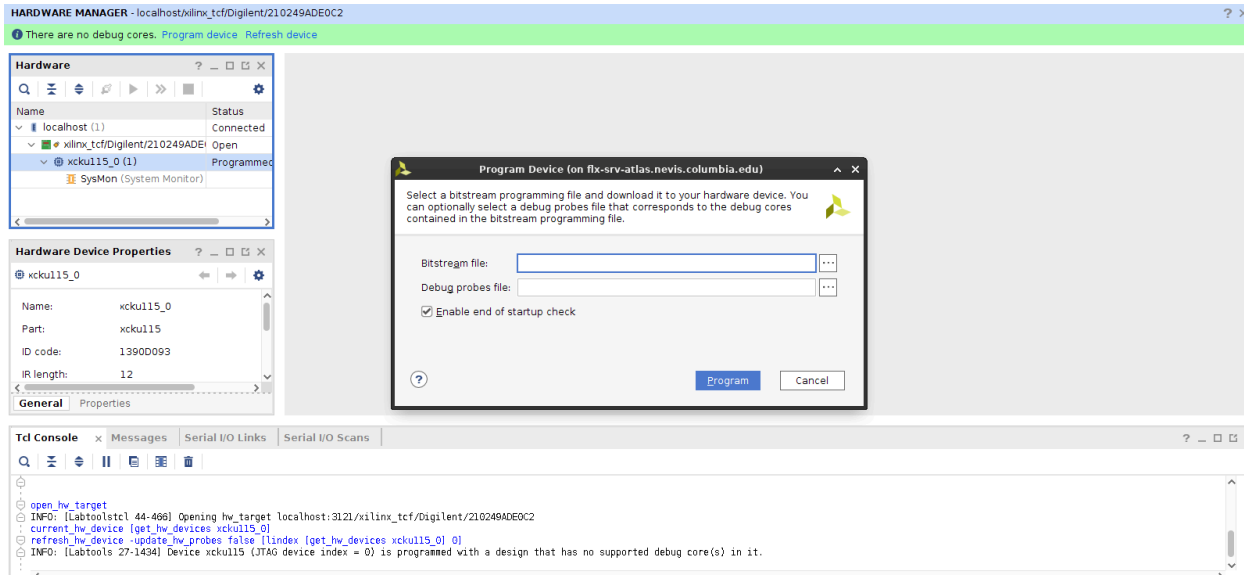
2) Program the FPGA:
-when the Vivado GUI appears, select “Open Hardware Manager”



-click “Open target”, then select “Auto connect” from the drop-down menu:

-right click on the FPGA part xcku115_0 and select “Program Device”

-use the interface to select which bitstream file to program, then press “Program”



-for reference the current “standard” bit stream is:
FLX712_LpGBT_RM0406_24CH_LOCALCLK_GIT_master_a5c8df7_1_240314_16_58.bit

-verify the programming process succeeds, then exit Vivado and log off the “atlas-group” account:

3) Reboot flx-srv-atlas:
-log in to the flx-srv-atlas using the dawillia account:
ssh -XY dawillia@flx-srv-atlas
(ask for the password)

-restart flx-srv-atlas:
sudo reboot 0
(ask for the password)

-at this point the ssh session should terminate
-wait ~2 minutes while the compute reboots

4) Reinitialize the FELIX readout:
-log in to the flx-srv-atlas using the dawillia account:
ssh -XY dawillia@flx-srv-atlas
(ask for the password)

-restart the FELIX interface drivers:
cd /home/dawillia/FLX/software/drivers_rcc/script
sudo ./drivers_flx_local start
(ask for the password)

-reinitialize the FELIX:
flx-init

-write FELIX register 0x66D0
fpepo 0x66d0 0x102

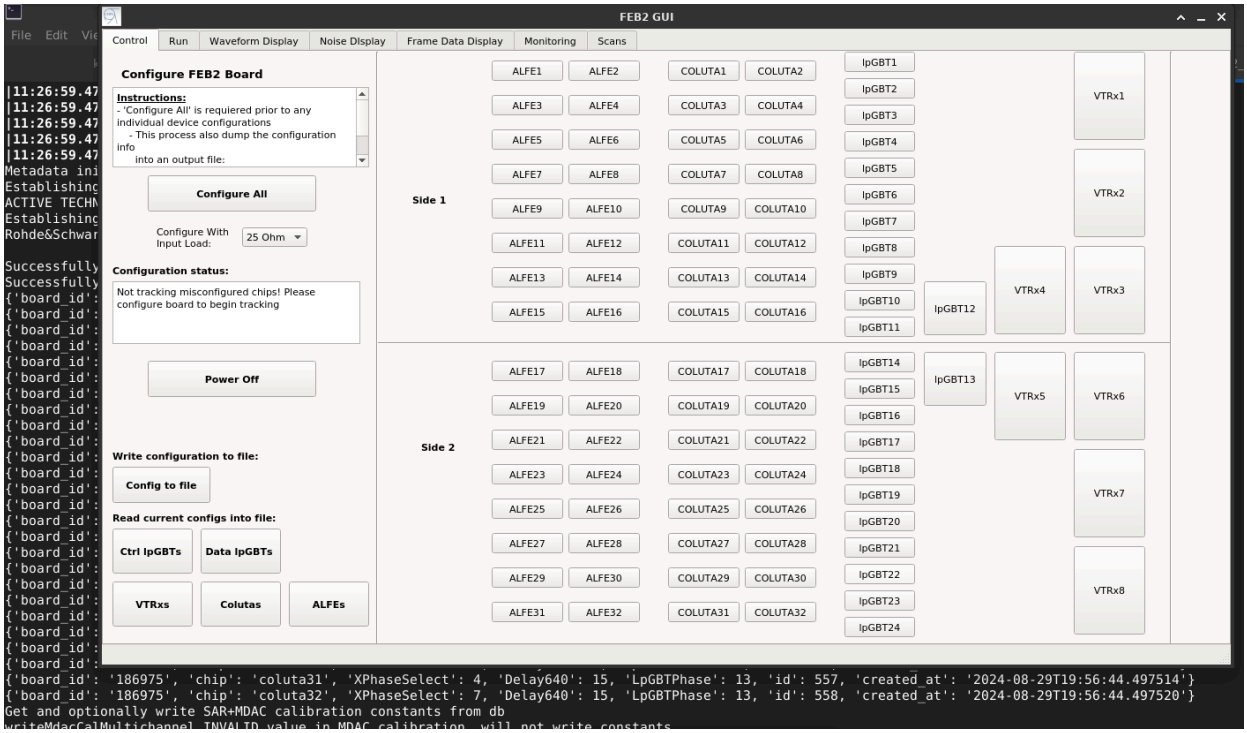
5) Try to run the GUI or board interface:

-go to the main working directory, or your own working directory:
cd ~/FLX/feb2/gui-new/feb2_gui/

-activate the anaconda environment
conda activate coluta

-run the GUI:
python main.py

-at this point the GUI window should appear, press “Configure All”:



-if the system is working the board should configure normally (note half the board is kept off in this screenshot). Note that if just testing the IC interface to the control lpGBTs it is sufficient to see that the control lpGBT configuration completed in the terminal text:

```
Metadata initialized from ../metadata/metadata.json
Successfully loaded local .env file!
Successfully loaded local .env file!
Loading restuls for board 186975 from master json file
Get and optionally write SAR+MDAC calibration constants from json
Get SAR+MDAC calibration DONE
Begin control lpGBT configuration
Configuring lpgbt12...
Success: True
Configuring lpgbt13...
Success: True
```

-when done, prepare the board to get turned off by pressing “Power Off”