

# User manual of the BNL-711 v1.5 for sPHENIX

Kai CHEN

*kchen@bnl.gov*

*Brookhaven National Laboratory*

March 16th, 2017

updated@ June 26, 2017

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Firmware</b>	<b>2</b>
<b>3</b>	<b>Software</b>	<b>2</b>

## 1 Introduction

This manual is a short introduction about how to use the BNL-711 v1.5. The chapters 2 & 3 are the guideline to run it with basic SW & FW from the FELIX project.

## 2 Firmware

FELIX is used to interface front-ends in ATLAS experiment. The firmware will support GBT mode and FULL mode. GBT mode in FELIX is a redesign based on the GBT architecture designed by CERN. FULL mode is a light-weighted mode to support higher throughput. Besides these IP cores, others like central router, TTC decoder are also included. For other use case, the most important module one can referred is the PCIe DMA engine: Wupper. It has been published on the opencore [https://opencores.org/project,virtex7\\_pcie\\_dma](https://opencores.org/project,virtex7_pcie_dma) with both of firmware and software. So users can refer to the design on this website. To get the latest version, users should extract the relative code from the FELIX firmware.

We use SLC6 as the aimed platform. The current Vivado version we used is 2015.4. To build the FELIX firmware, the steps will be:

1. Go to directory firmware/scripts/FELIX\_top, run below script to build the project and then open vivado.

```
1 vivado -mode batch -source vivado_import_felix_bnl711_ltdb_nocr.tcl
2 vivado &
```

2. Change the parameter GBT\_NUM in firmware/scripts/FELIX\_top/do\_implementation\_BNL711.tcl. Then in Tcl console of the Vivado, run

```
1 cd firmware/scripts/FELIX_top
2 source ./do_implement_BNL711.tcl
```

If you just **want** to refer the PCIe DMA engine, then after step 1, the Wupper core are instantiated as pcie0 and pcie1.

## 3 Software

The FELIX software include the driver, low-level control & monitoring tools and also many special tools for FELIX project. One may be interested the driver, the common low-level tools: control and monitoring the hardware and firmware by read/write PCIe registers.

The README.md in the software shows details about how to install the FELIX software. The driver\_rcc.tar.gz install one copy of drivers. After the programming of FPGA with proper bit file, reboot the PC (Users can program the bit file into flash). After the reboot, one can check details of the two Xilinx PCIe endpoints with:

```
1 sudo lspci -vvv
```

To be convenient, one can add below path to the file .bashrc

```
1 source (path to software)/cmake_tdaq/bin/setup.sh x86_64-slc6-gcc49-opt
2 export PATH=(path to x86_64-slc6-gcc49-opt/flxcard):$PATH
3 export PATH=(path to x86_64-slc6-gcc49-opt/fel):$PATH
4 export PATH=(path to x86_64-slc6-gcc49-opt/fdaq):$PATH
5 export PATH=(path to x86_64-slc6-gcc49-opt/pepo):$PATH
6 export PATH=(path to x86_64-slc6-gcc49-opt/elinkconfig):$PATH
7 export PATH=(path to drivers_rcc/script):$PATH
8 export LD_LIBRARY_PATH==(path to x86_64-slc6-gcc49-opt/flxcard):$LD_LIBRARY_PATH
9 export LD_LIBRARY_PATH==(path to x86_64-slc6-gcc49-opt/regmap):$LD_LIBRARY_PATH
10 export LD_LIBRARY_PATH==(path to x86_64-slc6-gcc49-opt/drivers_rcc):$LD_LIBRARY_PATH
11 export LD_LIBRARY_PATH==(path to x86_64-slc6-gcc49-opt/felixbase):$LD_LIBRARY_PATH
12 export LD_LIBRARY_PATH==(path to x86_64-slc6-gcc49-opt/packetformat):$LD_LIBRARY_PATH
13 export LD_LIBRARY_PATH==(path to x86_64-slc6-gcc49-opt/fel):$LD_LIBRARY_PATH
14 export LD_LIBRARY_PATH==(path to x86_64-slc6-gcc49-opt/fdaq):$LD_LIBRARY_PATH
```

To load the driver (user can let it run automatically when booting PC):

```
1 sudo driver_flx_local start
```

Sometimes, it should be stop then started.

After that, the SI5345 should be configure with

```
1 sh SI5345_40p8mhz_BNL-711v1p5.sh
```

Check with:

```
1 flx-info SI5345
```

The LOL should be NO: means the SI5345 is locked.

The next step is to:

```
1 flx-init
```

If the fibers are connected rightly,

```
1 flx-info GBT
```

will shows all GBT links are locked.

Now, we can run:

```
1 flx-monitor
```

To show the temperature, voltage, current on the board. After reboot, some results of the first running are wrong. After that, the results are right.

Some other tools (-h will show manual):

- *flx-config* it can show all of the register values.

```
1 flx-config list
```

- *pepo* can be used to read and write all PCIe registers.
- *flx-throughput*, *flx-dump-blocks*, *flx-dma-test* can be used to test both endpoints (-d 0 and -d 1 can switch endpoint). *fel*, *fdaq* & *fcheck* can be used to save and check the data in format defined by FELIX. Before using these tools, *elinkconfig* must be used to configure the central router and data emulator inside FPGA.

`data_taking.py` is an example to save data from counter inside FW. It can be used when the attached bit file is used for FPGA.

To do the hot-plugging.

1. Remove PCIe endpoints for Xilinx (x2) and then PEX8732 (x3).
2. Program the FPGA with new bit file.
3. Rescan the PCIe.