FELIX Software Suite Distribution for DUNE

Minimal FELIX Distribution

Minimal FELIX Distribution contains the bare minimum to be able to read out the card, open/close the card and initiate DMA transfer (start/stop).

NOTE: The minimal distribution is only capable to readout the card and is not sufficient for controlling the card. The "full" distribution is sufficient in controlling the card.

Required packages:

Package	Git branch	Commit hash
flxcard	rm-5.0	53a2d137
packetformat	rm-5.0	15c0fc19
regmap	rm-5.0	ff9cad8f
drivers_rcc: 1. cmem_rcc 2. rcc_error	master	3f278dc0

Required headers/libraries:

- libcmem_rcc.so
- libcmem_rcc.so.4.5.0
- libDFDebug.so
- libDFDebug.so.4.5.0
- libdrivers_rcc.so
- libdrivers_rcc.so.debug
- libFlxCard.so
- libFlxCard.so.debug
- libgetinput.so
- libgetinput.so.4.5.0
- libio_rcc.so
- libio_rcc.so.4.5.0
- libpacketformat.so
- libpacketformat.so.debug
- librcc_error.so
- librcc_error.so.4.5.0
- libregmap.so
- libregmap.so.debug
- io_rcc_common.h
- io_rcc.h
- io_rcc_driver.h
- flx_common.h
- FlxCard.h

- FlxException.h
- cmem_rcc_drv.h
- cmem_rcc_common.h
- cmem_rcc.h
- flx_common.h
- DFDebug.h
- GlobalDebugSettings.h
- regmap-struct.h
- regmap-symbol.h
- regmap-common.h
- regmap.h
- rcc_error.h
- block_parser.hpp
- block_reader.hpp
- block_format.hpp
- detail/block_parser.hpp

"Full feature" FELIX Distribution

The full feature distribution will contain all the packages we need to include in the software for the following features:

- handle card readout
- initialize card, read-write registers
- configure elinks
- start/stop dataflow

The following packages and their dependencies required for these features are shown below. Note the package dependencies are in square brackets. Plain text denotes a FELIX software package, *italic* text denote external software packages and **bold** text indicates external packages provided by LCG.

- regmap - [no dependencies]
- drivers_rcc (cmem_rcc, rcc_errror etc.) - [no dependencies]
- packetformat - [*catch*]
- flxcard - [regmap drivers_rcc]
- ftools - [regmap, drivers_rcc, flxcard, *json*]
- elinkconfig - [regmap, drivers_rcc, flxcard, **Qt5**, *json*]
- *catch*
- *json*
- **Qt5**

Every package branch should be on rm-5.0 if available otherwise, they should be on the master branch.

Current Issues and resolutions

- There are available tools to help build the FELIX software into a distribution package, but some packages do not get installed as part of the distribution
    - Should initiate contact with ATLAS-TDAQ-Felix software distribution developers
- Currently unsure about how we should be organising the FELIX software distribution similar to the other DUNE products

- Discuss with DUNE-DAQ software librarians on how to proceed. One question to ask is if
    1. we clone-mirror the ATLAS FELIX Software Suite and maintain our own version
    2. we treat it as a dependency (we convince ATLAS FELIX devs to modify the distribution mechanism, so one is able to build a distro with EVERY package included.) We build the distro on our own, then it's handed over to DUNE DAQ Software librarians as a product.

Step By Step Instructions On Building the FELIX Software

Pre-Requisites:

For SLC6 and Centos7 you need the following packages:

```
gcc
mesa-libGL
xkeyboard-config
librdmacm
```

in addition, for Centos7 (based on CC7 Base x86_64 2016-02-09) you need:

```
make
libpng
libSM
libXrender
fontconfig
```

To rebuild parts of the driver (driver_rcc) you need the following packages:

```
kernel
kernel-devel
```

You need access to LCG on cvmfs either by accessing cvmfs from CERN directory OR using a local copy of LCG for Felix distribution. To check you can access LCG:

```
ls /cvmfs/sft.cern.ch/lcg
```

If not, then cvmfs may need to be installed and configured to access LCG. Otherwise, you can get a local copy of LCG from the site:

```
https://atlas-project-felix.web.cern.ch/atlas-project-
felix/user/dist/software/cvmfs/
```

and if you use a local copy, you will need to set some environment variables:

```
export LCG_BASE=<path to local copy of LCG>/cvmfs/sft.cern.ch/lcg
export TDAQ_BASE=<path to local copy of LCG>/cvmfs/atlas.cern.ch/repo/sw/tdaq
```

Finally, the command `lsb_release` should work, if not, install `redhat-lsb`.

Initial Setup:

First clone the top-level software directory:

```
git clone ssh://git@gitlab.cern.ch:7999/atlas-tdaq-felix/software
cd software
```

Regardless of the packages you want in the FELIX software, the following repositories must be cloned:

```
git clone ssh://git@gitlab.cern.ch:7999/atlas-tdaq-felix/client-template.git
git clone ssh://git@gitlab.cern.ch:7999/atlas-tdaq-felix/cmake_tdaq.git
git clone ssh://git@gitlab.cern.ch:7999/atlas-tdaq-felix/distribution.git
```

Now you need to clone the packages you want to build. For example, if you want to build the packages for the minimal distribution, clone the following packages:

```
git clone ssh://git@gitlab.cern.ch:7999/atlas-tdaq-felix/regmap.git
git clone ssh://git@gitlab.cern.ch:7999/atlas-tdaq-felix/drivers_rcc.git
git clone ssh://git@gitlab.cern.ch:7999/atlas-tdaq-felix/packetformat.git
git clone ssh://git@gitlab.cern.ch:7999/atlas-tdaq-felix/flxcard.git
mkdir external
cd external
ssh://git@gitlab.cern.ch:7999/atlas-tdaq-felix/external-catch.git
cd ..
```

Or, you can just edit the script `clone-all.sh` and remove all the packages and externals you don't need from the list. Then you can run the command (Probably best to do it this way):

```
./clone-all.sh
```

So this will clone `client-template`, `cmake_tdaq` , `distribution` and your desired software packages.

Now, make sure the relevant packages are from the rm-5.0 branch

```
./checkout_all.sh rm-5.0
```

This will switch all package branches to rm-5.0 if it is available, otherwise they remain on the master branch.

Creating the build directory and building the software:

To setup the correct compiler, the correct version of cmake and some other parameters, use the following while in the software directory:

```
source cmake_tdaq/bin/setup.sh x86_64-centos7-gcc8-opt
```

then create the build directory:

```
cmake_config x86_64-centos7-gcc8-opt
```

then compile the code at once:

```
cd x86_64-centos7-gcc8-opt
make -j
```

Creating the software Distribution (TBD):

???