



ATLAS Note

ATL-COM-DAQ-2019-160

24th May 2021



Draft version 0.1

1

RD53A readout with FELIX and YARR System Setup

2 B. Abbott^a, J. Lambert^a, F. Metcalfe^b, A. Paramonov^b, M. Trovato^b

3

^a*University of Oklahoma*, ^b*Argonne National Laboratory*

4

The procedure for setting up a system to test RD53A modules is described in this document.
The system relies on FELIX for readout and control of the modules. A dedicated server
running YARR software is used to perform testing/tuning operations of the modules.

5

6

7

8

11 Contents

12	1 Introduction	3
13	1.1 How to use this Document	3
14	2 Server Setup	4
15	2.1 VC709 Setup and Installation (<i>direct readout</i>)	4
16	2.2 FLX712 Setup and Installation (<i>direct and LpGBT readouts</i>)	7
17	3 Software Setup	10
18	3.1 Vivado and Cable Driver Installation	10
19	3.2 FELIX Software Installation	10
20	3.3 YARR Installation	12
21	3.4 Program FELIX	13
22	3.5 FELIX Drivers	13
23	3.6 Notes	14
24	4 Hardware Setup	16
25	4.1 Direct readout system	16
26	4.2 LpGBT readout system	22
27	4.3 RD53A SCC (single chip card) for Direct and LpGBT Readout	26
28	5 Using the System	28
29	5.1 Configure VLDB (<i>direct readout</i>)	28
30	5.2 Configure VLDB+ (<i>LpGBT readout</i>)	29
31	5.3 Configure Optoboard (V0)	31
32	5.4 Configure VC709 (<i>direct readout</i>)	31
33	5.5 Configure FLX712 (<i>direct readout</i>)	36
34	5.6 Configure FLX712 (<i>LpGBT readout</i>)	37
35	5.7 Configuring Yarr (<i>direct readout</i>)	41
36	5.8 Running YARR	44
37	5.9 Notes	44
38	6 Conclusions	45
39	Appendices	48
40	A 4-channel Interface Card (superseded)	48
41	B Hardware Needed	50
42	B.1 Computer Part List	50
43	B.2 Additional Hardware	51
44	B.3 VLDB Configuration	52

45 1 Introduction

46 To handle the conditions expected at the HL LHC (High Luminosity LHC) interaction points, the ATLAS
47 detector will be upgraded. The two relevant upgrades for this document are the ones involving the inner
48 detector and the readout system. The inner detector will consist of pixel modules based on the RD53A
49 [1] prototype pixel modules currently available [2]. These modules will be read out using FELIX [3,
50 4]. To test the readout of the RD53A pixel modules with FELIX, the NetIO [5] networking library has
51 been Incorporated into YARR (Yet Another Rapid Readout) [6], providing YARR with the ability to
52 communicate with the RD53A modules through the FELIX system. This document describes the procedure
53 for setting up a Felix server to readout the Rd53A chip with YARR and NetIO.

54 It is assumed that all required hardware is available. Appendix B.1 contains the list of components used to
55 build the server described in this document. Appendix B.2 lists the additional hardware required to setup a
56 FELIX server.

57 1.1 How to use this Document

58 This document describe the procedure to set up the system for the *direct readout* and *LpGBT readout* flavors
59 of the RD53A modules via FELIX. The document should be followed exactly. All material should be
60 read and all steps taken. Each step is described in **bold face** text, followed by a more detailed explanation.
61 Steps are separated based on the choice of readout flavor, which is always indicated in *italic face* text.
62 Where possible, intermediate results are reported as additional guidance. Links to additional resources
63 or information is provided; it is recommended that the user follow these links and read the associated
64 material.

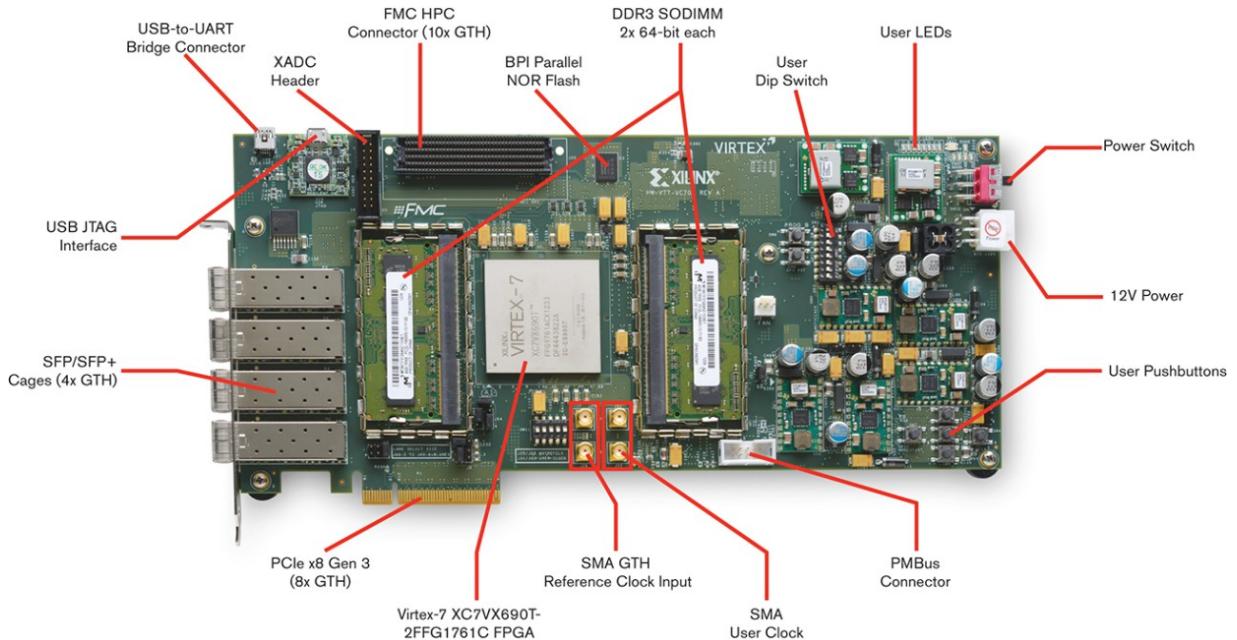


Figure 1: VC709. The VC709 hosts the FELIX firmware in the FELIX server. Note: the FMC port that the TTCfx attaches to (top-left), the SMA connectors used to supply the clock signal (bottom-middle), the power switch (top-right), the SFP cages (left), and the USB interface to the on-board JTAG chip (top-left). More details on each of these are covered later in the document. This figure was taken from the FELIX user manual[9]

65 2 Server Setup

66 This section describes the installation of the VC709 [7] (*direct readout*) and the FLX712 [8] (*direct and*
 67 *LpGBT readouts*) into the FELIX server. For more information on setting up FELIX, see the [FELIX online](#)
 68 [manual](#).

69 2.1 VC709 Setup and Installation (*direct readout*)

70 The VC709 Xilinx evaluation board, also named mini-FELIX, can be seen in Figure 1. The procedure
 71 below needs to be followed to properly install and setup the card.

72 **Attach TTCfx mezzanine card to FMC port on the VC709.** Before installing the VC709 into the server,
 73 we need to attach the TTCfx mezzanine card to the FMC port on the VC709. The TTCfx mezzanine card is
 74 shown in Figure 2. This card acts as a jitter cleaner and is used to pass an external TTC signal to the FPGA
 75 or to pass the internally generated clock to the FPGA in the case where a TTC signal is not available.

76 **Connect the SMA cables to TTCfx and VC709.** To pass the clock from the TTCfx to the FPGA, you
 77 need to connect the TTCfx and the VC709 board using two SMA cables. This can be done by connecting
 78 the SMA connector labeled Positive (Negative) in Figure 2 and the SMA connector labeled Positive
 79 (Negative) in Figure 3.

80 **Install the VC709 board in the FELIX server PCIe slot and power the board.** At this point the SFPs
 81 should **not** be inserted in the SFP cages on the VC709. If they are inserted into the SFP cages before the

Not reviewed, for internal circulation only

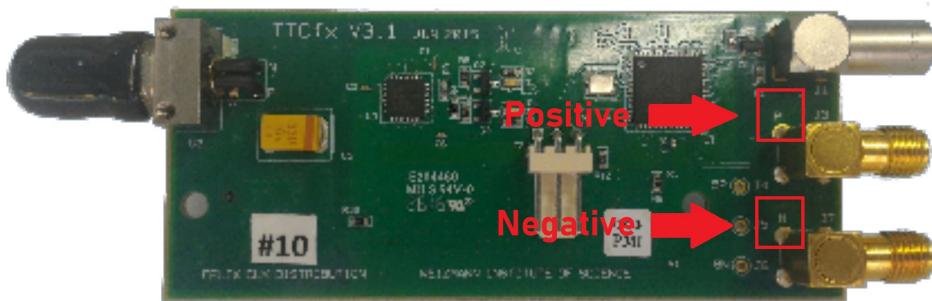


Figure 2: TTCfx mezzanine card. The TTCfx is used to supply the clock to the VC709. The clock can either be supplied from an external TTC system or generated locally in the VC709.

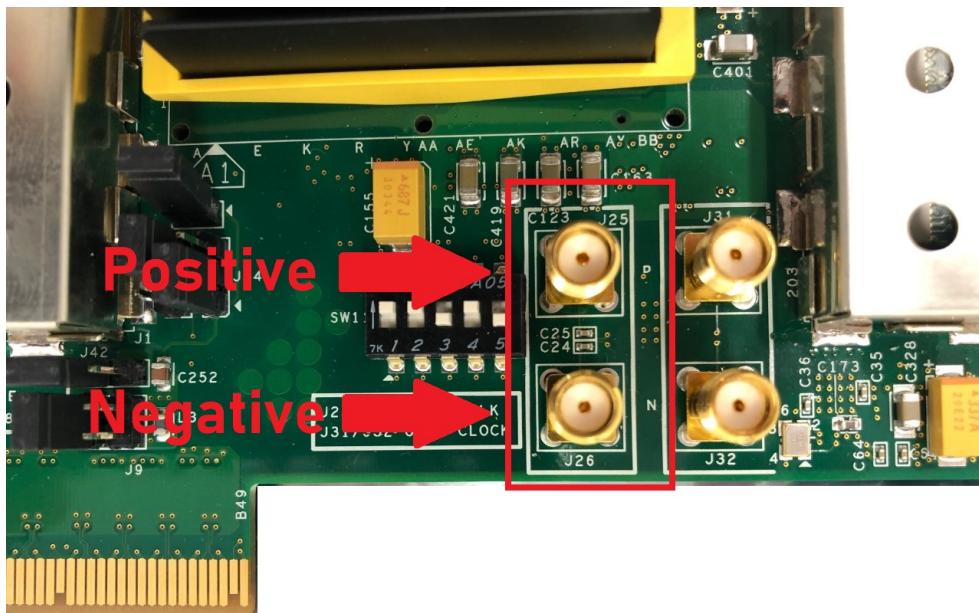


Figure 3: The VC709 has four SMA connectors. Two of these are used to supply the firmware with the appropriate clock signal. Since the clock signal is LVDS, there is a positive and a negative connector. Note the 'P' (positive) and 'N' (negative) in the figure.



Figure 4: ATX adapter used to power the VC709 when installed in the FELIX server. **Note:** do not power the VC709 using the PCIe power cable available on your servers power supply. This may damage the VC709.

Not reviewed, for internal circulation only

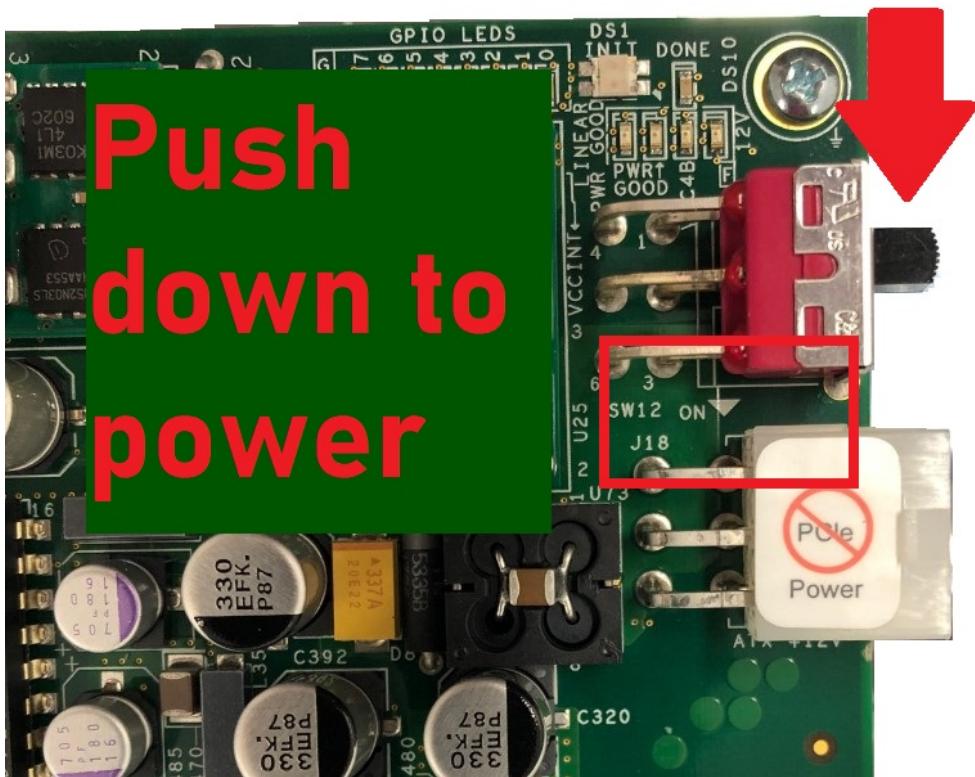


Figure 5: To power the VC709, you will need to press the power switch down. If the server is powered on and the VC709 is receiving power, the fan on the VC709 will start and you will see several LEDs light up on the VC709, as well.

82 VC709 is installed, it may be difficult to get the VC709 to fit properly into the servers PCIe slot. Once the
 83 board is inserted into the PCIe slot, use the 4-pin ATX adapter included with VC709 to provide power to
 84 the VC709. The ATX adapter is shown in Figure 4. To power the VC709, make sure the power switch is in
 85 the correct position, as indicated in Figure 5. When the server is powered on, you should see the lights on
 86 the VC709 come on and the fan start. If they do not, make sure that the power switch is set to the correct
 87 position and that the VC709 is correctly being powered by the servers power supply.

88 **Install the SFPs into the SFP cages on the VC709.** You can now install the SFPs in the SFP cages. Be
 89 sure that they are pushed in all of the way or you will experience issues later. When removing the SFPs,
 90 **do not pull them out directly.** You will need to pull the small lever back and then pull the SFP out. If
 91 you pull the SFP out without using the lever, you could damage the VC709, the SFP, or both. The SFPs



Figure 6: The SFPs installed in the VC709 after VC709 installation. The release lever used to release the SFPs from the SFP cages is indicated. Also indicated is the arrows on the SFP that indicate if the fiber optic port is Rx (sends data out) and Tx (receives data).

⁹² installed in the SFP cages can be seen in Figure 6.

⁹³ 2.1.1 Notes

- ⁹⁴ • Depending on what computer you have, you may have to remove some of the feet from the VC709 so
⁹⁵ that the board will fit.
- ⁹⁶ • Make sure the board is properly seated in the PCIe slot before continuing. If the board is not seated
⁹⁷ properly, the server will not be able to detect the VC709.
- ⁹⁸ • **Do not power the VC709 through PCIe (the 6-pin Molex connector coming from the power
⁹⁹ supply).** Instead, you must use the adapter that should have come with the VC709 to convert a 4-pin
¹⁰⁰ ATX connector to 6-pin Molex.
- ¹⁰¹ • When you power on the computer, the VC709 fan should start as well as some lights on the board. If
¹⁰² not, make sure that the power switch on the VC709 has the correct position.
- ¹⁰³ • Do not pull out the SFP with out pulling on the lever.

¹⁰⁴ 2.2 FLX712 Setup and Installation (*direct and LpGBT readouts*)

¹⁰⁵ The FLX712 board, FELIX, can be seen in Figure 7. The procedure below needs to be followed to properly
¹⁰⁶ install and setup the card. For more information on setting up FELIX, see the [FELIX online manual](#).

¹⁰⁷ **Install the FLX712 board in the FELIX server PCIe slot and power the board.** Once the board is
¹⁰⁸ inserted into the PCIe slot, the board must be connected to power from the system's internal ATX power
¹⁰⁹ supply via the available PCIe 6-pin, 8-pin, or 6+2-pin power cables. (Figure 8).

¹¹⁰ **Connect MTP24 or MTP48 fibers to FELIX couplers.** Depending on whether you have a FLX712
¹¹¹ board equipped with 8 minipods (Figure 7) or 4 minipods you need to use MTP48-to-4xMTP12 breakout
¹¹² cables or MTP24-to-2xMTP12 breakout cables (Appendix B.2). In both cases, OM4 multimode, Type B
¹¹³ Rollover polarity and female connectors for both sides of the cable are advised.



Figure 7: The BNL712 v2.0 board with the Phase1-TTC mezzanine

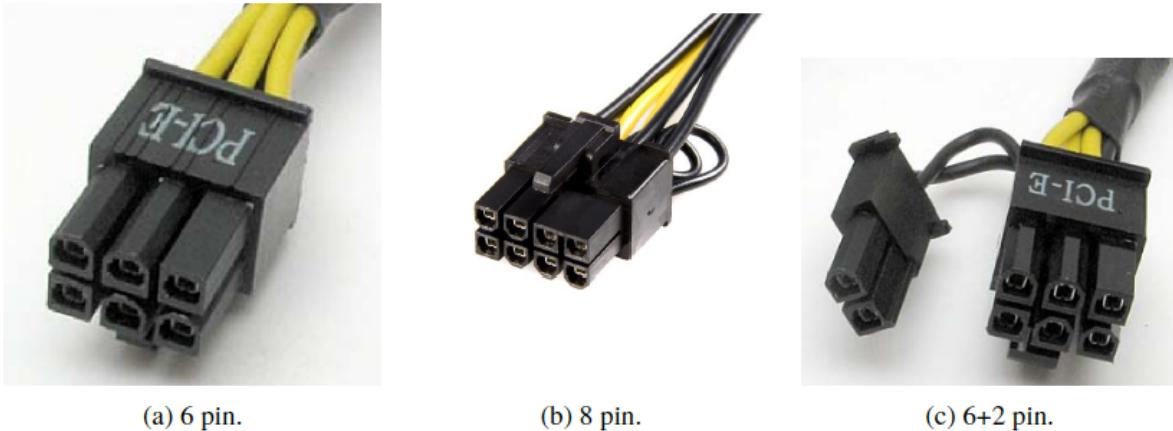


Figure 8: PCIe 6-pin (a), 8 pin (b), 6+2 pin (c) connectors

¹¹⁴ In addition, MTP12-to-12xLC cables need to be purchased in order to connect FELIX to the VLDB or
¹¹⁵ VLDB+ (Appendix B.2). Notice that the MTP connector side should be male and not female. See Section
¹¹⁶ 4 for more details.

¹¹⁷ FLX712 is already equipped with minipods (<https://www.digikey.com/en/products/detail/foxconn-oe-technologies-singapore-pte-ltd/AFBR-814FN1Z/5043130>) connected to MTP couplers (<https://store.cablesplususa.com/mt-adpt.html>) via PRIZM custom-made patches. Therefore, no internal cabling work is needed.

¹²⁰ PUT FIGURE of MTP48, 24 to 12LC here

¹²¹ **2.2.1 Notes**

- ¹²² • MTP48 and MTP24 cables are compatible with the 48-channel (8 minipods) and 24-channel (4
¹²³ minipods) respectively.

[Not reviewed, for internal circulation only]

¹²⁴ 3 Software Setup

¹²⁵ This section describes how to install Vivado, FELIX and cable drivers, YARR and FELIX software
¹²⁶ packages. For more information, see the [Vivado installation guide](#), the [FELIX users manual](#), the [FELIX](#)
¹²⁷ [software README](#), and the [YARR repository](#).

¹²⁸ 3.1 Vivado and Cable Driver Installation

¹²⁹ To start, we will install Vivado [10]. You can download a free version of Vivado from the [Xilinx website](#).
¹³⁰ The versions used are 2015.4 (*direct readout*, VC709), 2018.1 (*direct readout*, FLX712), 2020.1 (*LpGBT*
¹³¹ *readout*, FLX712). The lab edition, which does not require a license, is enough for programming the FPGA
¹³² and monitor signals.

¹³³ Assuming you have downloaded the 2015.4 Lab Edition version of Vivado, you can untar the tarball and
¹³⁴ start the installation process by running (this may require root privileges):

```
135 $ tar -xzf Xilinx_Vivado_Lab_Lin_2015.4_1118_2.tar.gz
136 $ ./Xilinx_Vivado_Lab_Lin_2015.4_1118_2/xsetup
```

¹³⁷ Follow the prompts given by the GUI to finish installing Vivado. If you have any trouble, see the Vivado
¹³⁸ installation guide.

¹³⁹ After installation, you may need to install the cable drivers so that Vivado can communicate with the
¹⁴⁰ VC709 using JTAG. To do this see the 'Installing Cable Drivers' section in the Vivado installation guide.
¹⁴¹ You will be instructed to execute the following

```
142 sudo ./<Vivado Install
143 Dir>/data/xicom/cable_drivers/lin64/install_script/install_drivers/
144 install_drivers.sh
```

¹⁴⁵ **Note:** You may need to execute the other scripts in the above folder; I recommend running all of the driver
¹⁴⁶ scripts just to make sure you do not have any cable driver related issues. Once Vivado and the associated
¹⁴⁷ cable drivers are installed, it is a good idea to reboot the server.

¹⁴⁸ This completes the Vivado setup. Next we look at how to install the necessary FELIX software.

¹⁴⁹ 3.2 FELIX Software Installation

¹⁵⁰ Before installing the FELIX software, you will need access to CVMFS and to install the FELIX software
¹⁵¹ dependencies. For more information on the installation of the FELIX software, refer to the Felix software
¹⁵² GitLab README. I recommend looking over this before proceeding.

¹⁵³ **Obtain access to CVMFS.** You can either mount CVMFS directly to access the file system over the
¹⁵⁴ internet (recommended), download CVMFS, or download a subset of CVMFS available from the FELIX
¹⁵⁵ developers that contains only what is needed to use a specific version of the FELIX software. There is more
¹⁵⁶ information about this on the [FELIX software GitLab README](#). The downloadable version of CVMFS
¹⁵⁷ can be obtained from the [FELIX software archive](#), or by running the following command:

```
158 $ wget https://atlas-project-felix.web.cern.ch/atlas-project-felix/
159   user/dist/software/cvmfs/cvmfs-felix-93-4.tar.gz
```

160 Once the download has completed, run

```
161 $ tar -xzf cvmfs-felix-93-4.tar.gz
162 $ sudo mv cvmfs /
```

163 This will place a CVMFS directory in the root directory. **Note:** if you have trouble compiling the FELIX software, you may need to try a different version of the downloadable CVMFS directory.

165 **Install FELIX software dependencies.** The full set of dependencies can be found on the Felix software GitLab README. They can also be installed using the following command:

```
167 $ sudo yum install gcc mesa-libGL xkeyboard-config librdmacm make
168   libpng libSM libXrender fontconfig kernel kernel-devel
```

169 After installing the necessary dependencies, you should restart your computer.

170 Install FELIX Software. A different software version is required depending on the chosen readout chain.

171 **Install Modified Version of FELIX Software (*direct readout, VC709*)** Here, we will be installing a modified version of the FELIX software. The modified version has a minor change that allows the clock on the VC709 to be set to the 160 MHz. The custom version is packaged together in a tarball to insure that everything will work properly. The modified FELIX software can be cloned from GitLab [J. Lambert gitlab repo](#) or by running the following command

```
176 git clone ssh://git@gitlab.cern.ch:7999/jolamber/rd53a-felix-software
177   .git
```

178 This command expects that you have the proper ssh keys installed on the server.

179 To untar and compile the FELIX software run the following commands

```
180 $ cd rd53a-felix-software/
181 $ tar -xzf software.tar.gz
182 $ cd software
183 $ source cmake_tdaq/bin/setup.sh x86_64-centos7-gcc62-opt
184 $ cmake_config x86_64-centos7-gcc62-opt
185 $ cd x86_64-centos7-gcc62-opt
186 $ make -j
```

187 **Note:** The Felix software may not compile if devtoolset-7 is enabled (see [3.3](#)). After compiling, the executables will not be added to the your PATH environment variable so you can either run them from the corresponding directory in the x86_64-centos7-gcc62-opt directory or add each of these directories to the PATH environment variable. Every time you open a new terminal, you will need to source the setup script in the cmake_tdaq folder as given in the above list of commands, if you intend to run the Felix commands. I recommend making a setup script that sets up the FELIX environment and adds the necessary executables to your PATH environment variable.

194 **Install Version of FELIX Software (*direct readout, FLX712*)**. The 4.01.00-rc7 version of the pre-compiled software was used. That version is available in the [Felix software archive](#). Follow the README to learn how to use it.

197 **Install Version of FELIX Software (*LpGBT readout, FLX712*)** The software being used is the rm-5.0
 198 branch of the [FELIX software](#). Follow the README to understand how to recursively switch the rm-5.0
 199 branch for all the packages and how to compile. **Note** that the software is still work in progress and only
 200 low level tools can be used. Felixcore/star are not compatible yet with the firmware.

201 3.3 YARR Installation

202 For RD53A readout, we are using YARR. The software can be found on [YARR gitlab repo](#). For additional
 203 information on YARR, I recommend that you look through the detailed documentation located at the page
 204 [YARR readthedocs](#) before proceeding.

205 **Clone YARR repository.** For our purposes, we will clone the `devel_rd53a_felixNetio_multichip_rebase`
 206 branch of YARR. This branch contains some slight modifications that allow it to work properly for the
 207 RD53A readout with FELIX. To clone this branch, run the following command:

```
208 $ git clone -b devel_rd53a_felixNetio_multichip_rebase ssh://  
209     git@gitlab.cern.ch:7999/YARR/YARR.git
```

210 Here, I have assumed that you have the appropriate ssh keys setup on the server.

211 **Install necessary packages for YARR.** Before we can compile YARR, we will need to make sure that all
 212 of the dependencies are installed on the server. The steps to install these dependencies can be found at the
 213 [YARR readthedocs page](#) and I have listed the necessary steps below, as well.

214 Assuming that you are using CentOS7, you can run the following commands:

```
215 $ sudo yum install centos-release-scl  
216 $ sudo yum-config-manager --enable rhel-server-rhscl-7-rpms  
217 $ sudo yum install devtoolset-7
```

218 This installs devtoolset-7 on your machine, giving you access to GCC version 7. In order to enable
 219 devtoolset-7, you need to run the following

```
220 $ source /opt/rh/devtoolset-7/enable
```

221 You will need to run the above command each time you start a new terminal and want to run/compile
 222 YARR. **Note:** devtoolset-7 may affect the FELIX software so you should **not** run this command in a
 223 terminal you plan to run FELIX software in.

224 Other dependencies can be installed by running

```
225 $ sudo yum install gnuplot texlive-epstopdf cmake3 zeromq zeromq-  
226     devel
```

227 **Compile YARR.** To compile YARR

```
228 $ cd YARR && mkdir build && cd build  
229 $ cmake3 -DYARR_CONTROLLERS_TO_BUILD=Spec\;Emu\;NetioHW ..
```

230 This will compile YARR with NetIO. If you have issues, try restarting your computer.

231 We will now describe how to program the VC709 and install the FELIX drivers. To program the FPGA,
 232 you will need to have Vivado installed on a computer (not necessarily the FELIX server) and a USB to
 233 micro-USB cable connecting the computer with Vivado installed and the JTAG USB interface indicated in
 234 Figure 1.

235 3.4 Program FELIX

236 Firmware is available in two flavors: *direct readout* and *LpGBT readout*. *Direct readout* is available for
 237 both VC709 and FLX712. *LpGBT readout* is only available for FLX712. For more details see the [M. Trovato's ITk Pixel Readout slides](#). Bit files for the VC709 and FLX712 are stored in the [Pixel group webpage](#) or [M. Trovato's cern box](#)

240 **Program the VC709.** Once you have Vivado (2015.4 or 2018.1) open:

- 241 • Click the 'Hardware Manager' button
- 242 • In the green bar at the top, click 'Open target', then 'auto connect'. This can be seen in Figure 9.
- 243 • Right click the FPGA and select 'program device'
- 244 • Navigate to and select the FELIX firmware (the .bit file) and click 'OK'

245 After this, the FPGA should be programmed and you will need to restart the computer. **Note:** if you restart
 246 the FELIX server, the FPGA may no longer be programmed.

247 **Program the FLX712.** Similar procedure to the one described above for the VC709 should be used.
 248 Vivado 2018.1 (*direct readout*) or 2020.1 (*LpGBT readout*) should be used.

249 3.5 FELIX Drivers

250 **Install the FELIX drivers.** The FELIX drivers tell the FELIX server how to communicate with the
 251 VC709 or FLX712. The Felix drivers can be installed in two ways:

- 252 • Using an RPM from [FELIX driver archive](#)
- 253 • Installing them manually from the Felix software available on Git.

254 For instructions on install the drivers using the RPM, see the Felix users manual. 4.0.4 and 4.5.2 versions
 255 are used respectively for the VC709 and FLX712 (both *direct* and *LpGBT readouts*)

256 **Note:** I have had issues using the RPM drivers in the past. If you have issues with the RPM drivers,
 257 compile and install the drivers from the FELIX software downloaded in an earlier step. To install the
 258 drivers from GitLab, **do not** setup the FELIX environment, setup devtoolset-7 (available after following
 259 the instructions to install YARR, see [3.3](#)), and run the following commands:

```
260 $ cd <Path to felix software>
261 $ cd drivers_rcc/src
262 $ make
263 $ cd ../../script
264 $ sudo ./drivers_flx_local start
```

Not reviewed, for internal circulation only

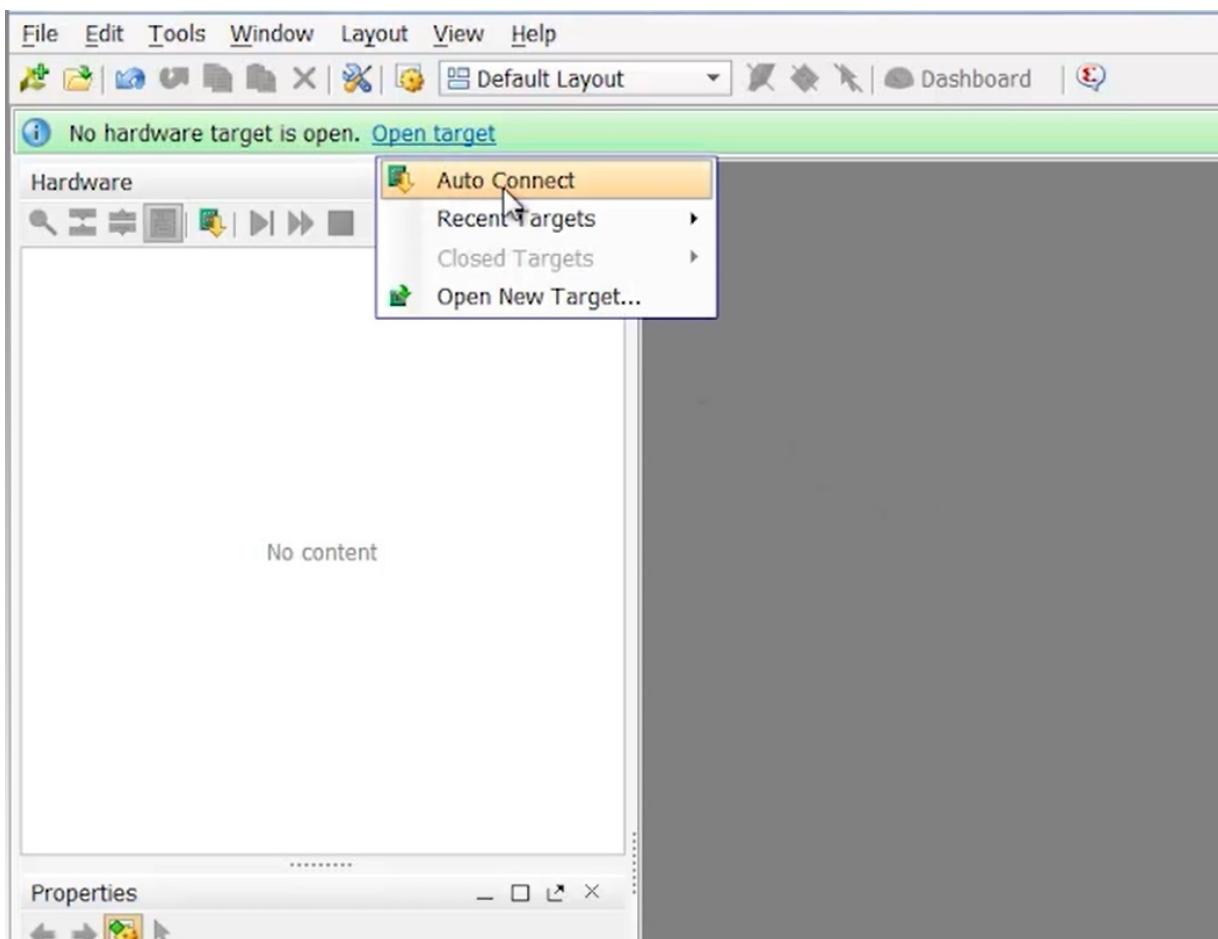


Figure 9: To connect to the FPGA, click 'Open target' in the green bar, and then select 'auto connect'.

265 **Note:** The drivers may not compile properly if you have run the setup script in the cmake_tdaq directory
 266 to setup the environment for FELIX commands. Also, you will need to setup devtoolset_7 and run the
 267 drivers_flx_local start script each time you restart the computer. If the drivers are setup successfully and
 268 the VC709 has been programmed, you will be able to run the following:

269 \$ `cat /proc/flx`

270 and see something similar to what is seen in Figure 10.

271 If everything in this section has been completed successfully, you are ready to test the RD53A with YARR
 272 and FELIX.

273 3.6 Notes

- 274 • When using the FELIX software from GitLab, you will need to add the executables to your PATH
 275 (recommended) or execute them directly from the x86_64-centos7-gcc62-opt/ directory that is
 276 created when compiling.

```

FLX driver for RM4.0 F/W for release tdaq710_for_felix_4.0.4. Distributed with driver RPM 4.0.4

Debug          = 0
Number of cards detected = 1

Locked resources
card | global_locks
=====|=====
0   | 0xffffffff

Locked resources
card | resource bit |      PID | tag
====|=====|=====|=====
0   |       0 | 17009 | 43
0   |       1 | 17009 | 43
0   |       2 | 17009 | 43
0   |       3 | 17009 | 43
0   |       4 | 17009 | 43
0   |       5 | 17009 | 43
0   |       6 | 17009 | 43
0   |       7 | 17009 | 43
0   |       8 | 17009 | 43
0   |       9 | 17009 | 43
0   |      10 | 17009 | 43
0   |      11 | 17009 | 43

```

Figure 10: The expected output of cat /proc/flx after successfully installing the Felix drivers and after programming the FPGA. Your output may differ but the important thing to note is that if this command will tell you whether FELIX is available or not.

- 277 • If you have trouble using the RPM FELIX drivers, try compiling and loading the FELIX drivers
278 from the FELIX GitLab repository.
- 279 • YARR and the FELIX drivers need to be compiled with devtoolset-7 and not the environment setup
280 using the setup script in the cmake_tdaq directory.
- 281 • The FELIX software needs to be compiled with the environment setup by the setup script in the
282 cmake_tdaq directory and not devtoolset-7.
- 283 • If you are using the drivers from the FELIX GitLab repository, you will need to reload them each
284 time the server is rebooted.
- 285 • If you reboot the server, the FPGA may not be programmed any more.
- 286 • You need to reboot the computer after programming the VC709.

Not reviewed, for internal circulation only

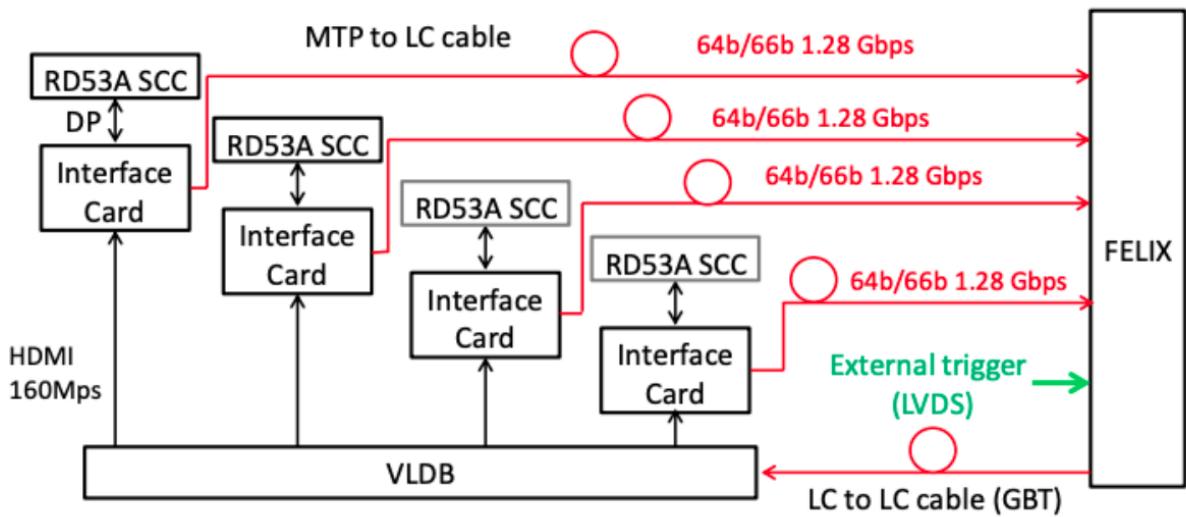


Figure 11: Block diagram of the *direct readout* system. FELIX stands for the FELIX server hosting either a VC709 or a FLX712. DisplayPort (DP), HDMI, or optical cables used to connect the different components are also indicated. Commands from FELIX are passed via the LC cable to the VLDB. The VLDB routes the command to the right mini-HDMI port according to the elink specified in the packet. The command is then routed to the interface card via HDMI cables and to the RD53A via DP cables. Data returned from the RD53A is routed to the interface card via DP cables and to FELIX via optical cables.

287 4 Hardware Setup

288 In this section, we will discuss how to setup the hardware required to configure and readout the RD53A
 289 modules using the *direct readout* (Section 4.1) or *LpGBT readout* (Section 4.2) system. Configuration of
 290 the RD53A Single chip card (SCC), which is the same for both readout systems, is described in Section
 291 4.3

292 **4.1 Direct readout system**

293 The following hardware components need to be connected: FELIX (VC709/FLX712), interface card,
 294 VLDB, and the RD53A SCC. To give you an idea of what the setup will look like when completed, a block
 295 diagram can be found in Figure 11. It may be helpful to refer back to this figure as you work on the setup.

296 **4.1.1 VLDB Setup**

297 The versatile link demonstrator board (VLDB) [11], seen in Figure 12 is used to distribute the data from the
 298 VC709 to different modules by routing the data to the correct mini-HDMI port on the board. This section
 299 will describe the dip switch settings, the voltage settings, how to connect the VLDB to the VC709/FLX712,
 300 and how to program the VLDB.

Not reviewed, for internal circulation only

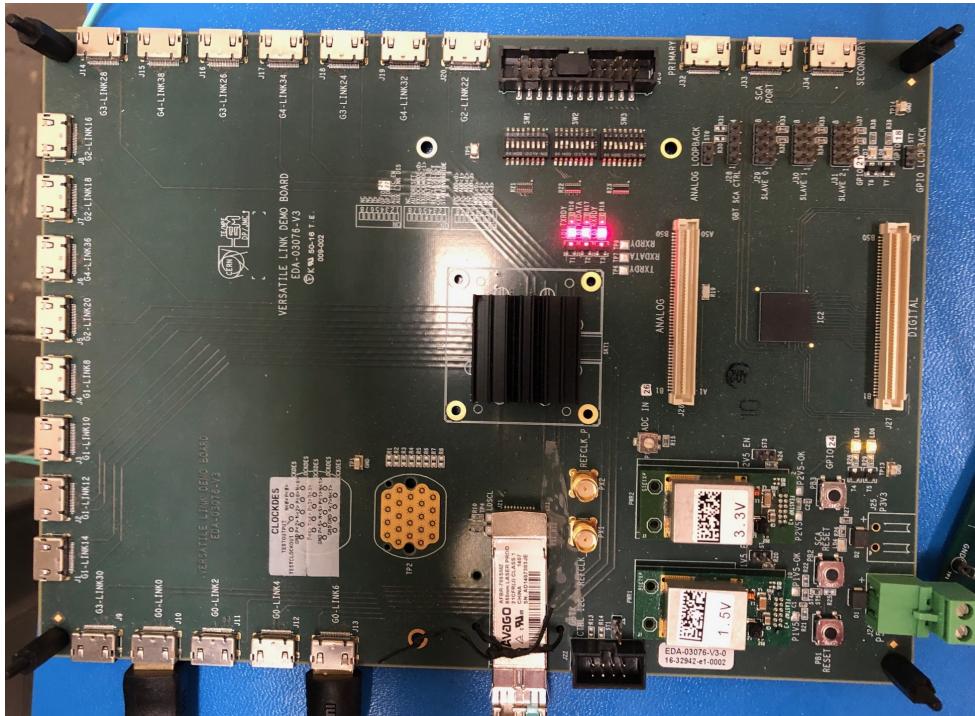


Figure 12: VLDB used to readout the RD53a. Note the power connector bottom-right; mini-HDMI ports that connect the VLDB to the interface card; the SFP that connects the VLDB to the VC709; the red indicator lights; the dip switches above the indicator lights; and the step-down converters in-between the SFP and the green power connector.

301 **Configure VLDB dip switch as shown in Figure 13.** There are several dip switches located on the VLDB
 302 that need to be set. For the readout of the RD53A, the dip switches should be set as seen in Figure 13. For
 303 more information on the VLDB and what these dip switches do, you can visit the [VLDB website](#).

304 **Set the VLDB voltages using the step-down converters as seen in Figure 14.** The voltages for different
 305 components on the VLDB can be set using the small step-down converters that should have come with the
 306 VLDB. These converters step-down the input 5V to the desired voltage. Here, we use the 1.5V and 3.3V
 307 converters.

308 **Place an SFP on VLDB as seen on the bottom of Figure 12.** In order for FELIX to pass data to the
 309 VLDB, we need to connect an SFP to the board. If you have a VC709, it should have come with four SFP
 310 modules; you can use one of these on the VLDB. Unfortunately, there is no cage to hold the SFP in place,
 311 so you should tie it down in some way. I used a twist-tie as can be seen in Figure 12.

312 **Power VLDB with 5V / 1A.** The VLDB will need to be powered using an external power supply capable
 313 of providing 5V and handling up to 1A¹. To do this you will need to find a connector since the VLDB does
 314 not come with one. The one I used can be seen in Figure 15. More information on the connector can be
 315 found in Appendix B.2. The correct polarity for each wire is indicated in the figure: the red wire should be
 316 connected to the positive terminal of your power supply and the black wire should be connected to the
 317 negative wire of your power supply.

318 **Connect VLDB and FLX712/VC709 with a fiber optic cable:**

¹ make sure the over-current protection limit of your power supply is above 1A or above to avoid problems. It may be hard to debug if the limit is below that. See [VLDB quick start guide](#) or [VLDB specs](#) for more details

Not reviewed, for internal circulation only

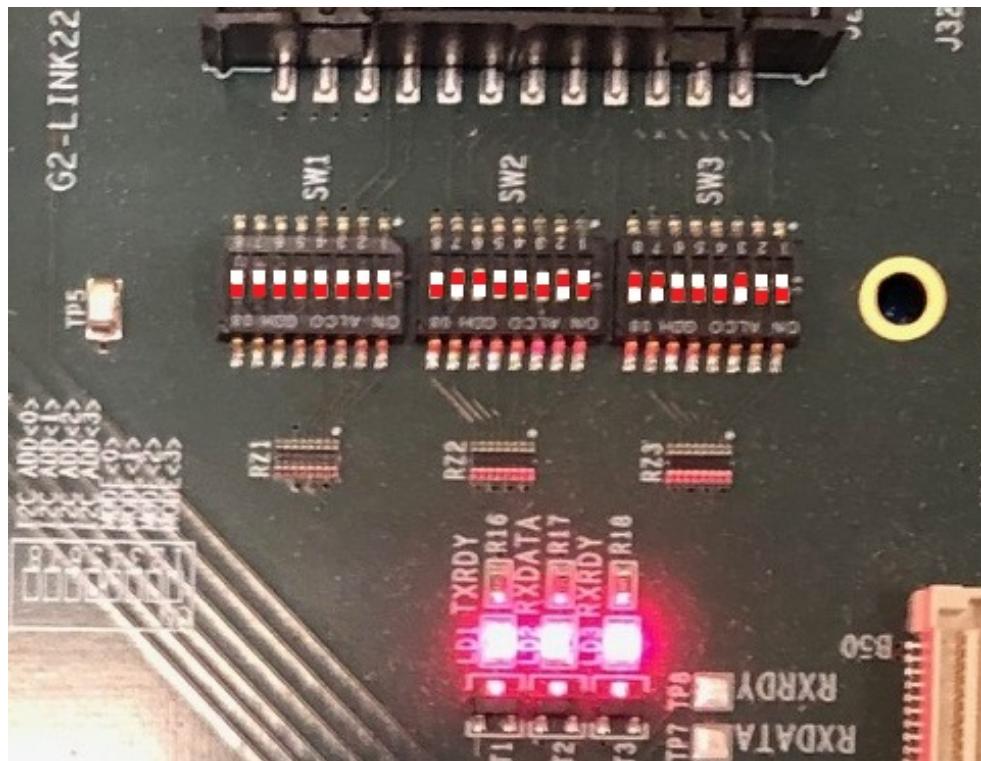


Figure 13: There is a row of dip switches on the VLDB that need to be set as shown here. The red squares in the figure represent the correct location of the dip switch.

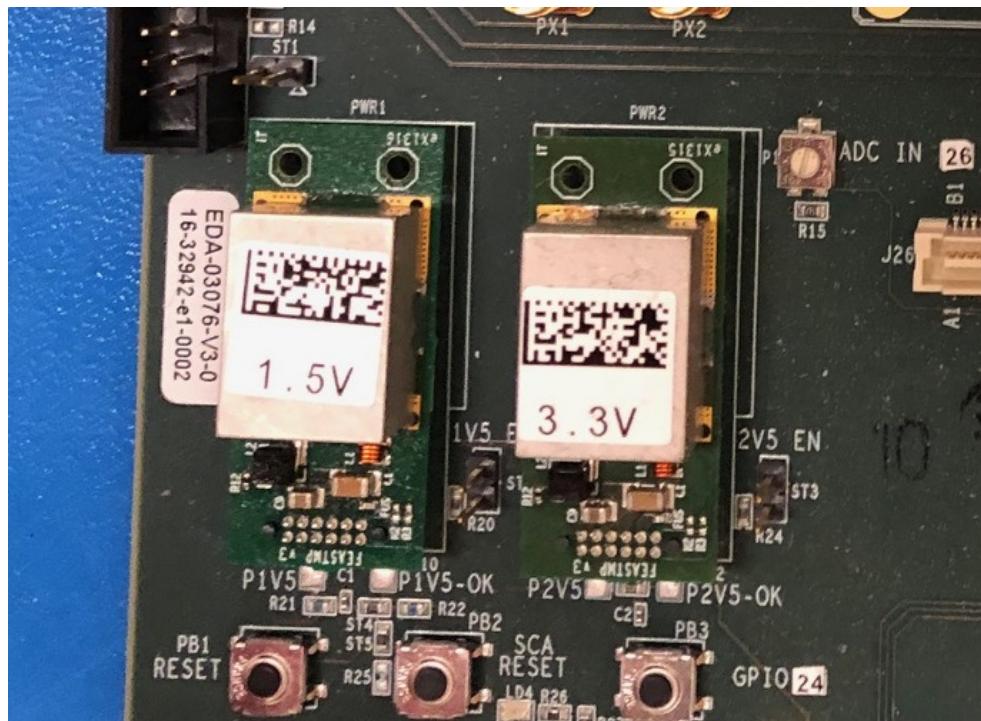


Figure 14: The VLDB allows the voltage for different components on the board to be set using the small step-down converters. For the RD53A readout, the converters need to be set as seen in the figure.



Figure 15: Power connector for the VLDB. In the figure, the positive terminal is indicated by the red wire and the negative terminal is indicated by the black wire.

- FLX712: the VLDB needs to be connected to one of the LC fiber carrying commands from FELIX. Any LC fiber outputting light may be used. However, in this example it is assumed that the fiber labeled with the number "12", which corresponds to channel "0" in FELIX, is used. MTP48 or MTP24 breakout cables and MTP to LC breakout cables should be used (see Appendix B.2 and Figures 16).
- VC709: the VLDB needs to be connected to the TX port of the SFP (arrow pointing out of the SFP on the release lever) **on the VC709 SFP marked 0** in Figure 45 to the Rx port of the SFP (arrow pointing into the SFP release lever) on the VLDB SFP using a single fiber optic cable. Figure 6 indicates the arrows that determine the Rx and Tx ports on the SFP. **NOTE:** you must connect the fiber from the VLDB to the SFP marked 0 in Figure 45 or your setup will not work. The Rx half of the SFP on the VC709 can be used to get data from an interface card so you can separate the fiber-pair and only plug in the Tx half on the VC709. In the next section, we will describe how to program the VLDB so that it is ready to use.

4.1.2 Interface Card

This section describes the setup of the 12-channel interface card developed at Argonne. The description of the 4-channel interface card, which is now superseded, is in Appendix A. The interface card can be seen in Figure 17. The interface card passes data received from the VLDB to the RD53A and data from the RD53A to the FLX712/VC709. Schematics for the card are available in the [RD53A Testing Twiki](#)

Connect the interface card to one of the VLDB ports seen in Figure 44 The VLDB should be connected

Not reviewed, for internal circulation only

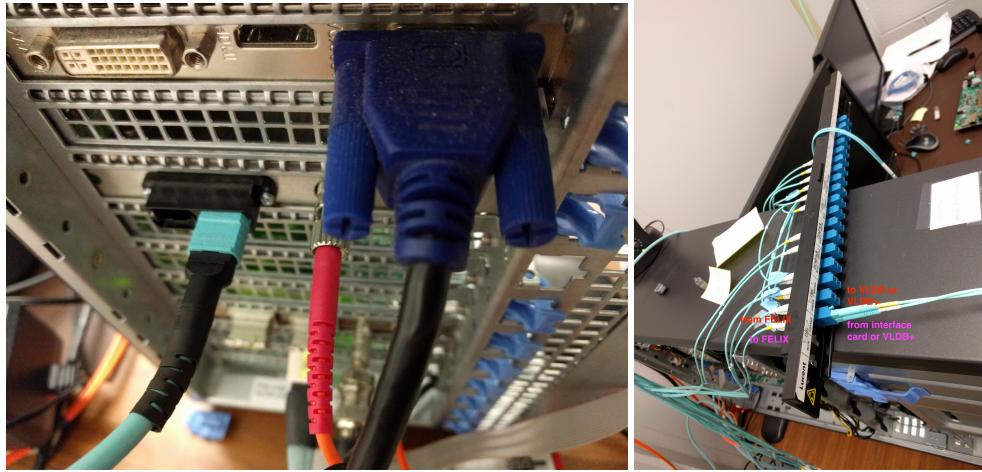


Figure 16: MTP48 to 4xMTP12 connected to a FELIX MTP coupler (left). One downlink MTP12 and one uplink MTP12 are connected two MTP to 12xLC fibers. From each bundle of twelve LC fibers, the fibers labeled with "12" are connected to left-hand side of the patch panel (right). A fiber going to VLDB (*direct readout*) or VLDB+ (*LpGBT readout*) is connected to the right-hand side of patch panel, as well a fiber from the interface card (*direct readout*) or VLDB+ (*LpGBT readout*)

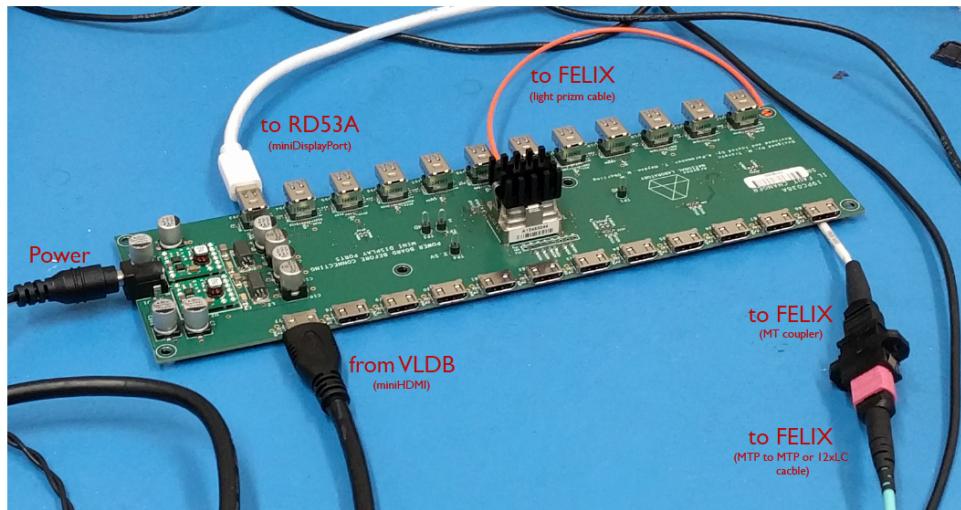


Figure 17: 12 channel interface card used to convert electrical signals from the RD53a to optical and pass data from the VLDB to the RD53A.

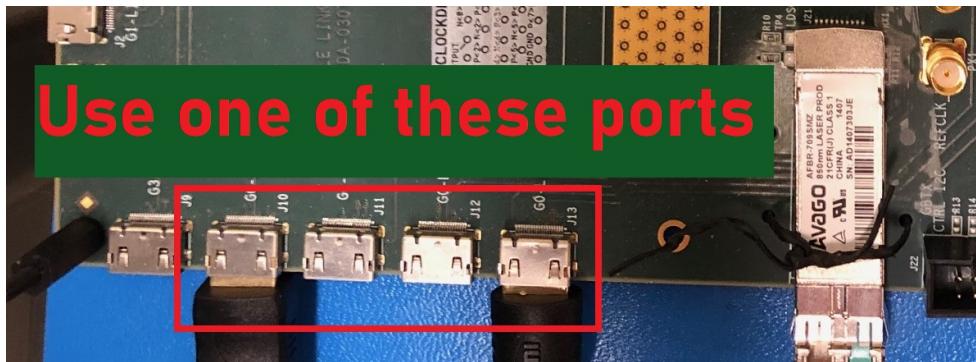


Figure 18: Mini-HDMI ports that can be used to connect to interface cards.

338 to the interface card using mini-HDMI-to-mini-HDMI cables (Appendix B.2). You should note the port
 339 that you use on the VLDB. Data passed to the VLDB from FELIX will be passed to one of these depending
 340 upon elink configuration (Section 5.4).

341 **Connect the interface card to the RD53A.** The interface card is connected to the RD53A using the
 342 mini-DisplayPort-to-DisplayPort cables (Appendix B.2). Use the miniDisplay connector directly across
 343 from the chosen mini-HDMI connector 17. On the RD53A side, use the DisplayPort connector on the
 344 RD53A closest to the Molex power connectors. The correct DisplayPort on the RD53A to use can be seen
 345 in Figure 25.

346 **Connect cable from the interface card to FLX712/VC709:**

347 • **FLX712** Plug the light prizm cable (Appendix B.2) in the minipod socket. Connect a MTP coupler
 348 (Appendix B.2) to the other side of the cable. At this point, you may choose to use a MTP to MTP
 349 cable (Appendix B.2) or a MTP to 12 LC cable (Appendix B.2) to connect to FELIX. In the former
 350 case plug one end of the cable into the MTP coupler of the interface card and the other end into one
 351 of the MTP coupler of the FLX712 (Figure 16). If the latter option is chosen, it shout be noticed
 352 that each fiber has a number written on a small, white, plastic piece that goes from 1 to 12. One to
 353 four LC fibers carry signals from the RD53A per single mini-DP connector. All 12 fibers can carry
 354 signals if three mini-DP connectors are connected to three RD53A SCCs. Refer to table ?? (TO DO:
 355 SPECIFY FIBER ASSOCIATED TO LANE0 OF RD53A) for further details. You can also closely
 356 look at the fibers outputting lights to select the meaningful ones. Fibers should be then connected to
 357 a patch panel (Figure 16). On the other side of the patch panel another 12-strand fiber finalize the
 358 connection to one of the coupler of the FLX712 (Figure 16)

359 • **VC709** The multi-strand fiber has many different fiber optic pairs. Each of these pairs has a number
 360 written on a small, white, plastic piece that goes from 1 to 8. If you are using a 12-strand fiber, these
 361 numbers will be different. For the 8-strand fiber, each pair corresponds to a different channel that can
 362 be enabled on the RD53A by setting a register in the RD53A. **By default, only the one channel is**
 363 **enabled, corresponding to the fiber numbered 8.** In this manual, we assume that only the default
 364 channel is enabled so this will be the only fiber needed. **Note: one module must be connected**
 365 **to the SFP labeled 2 in Figure 45** There are four different SFPs on the VC709 so we can receive
 366 data from four different modules. The SFP labeled 2 in Figure 45 acts as the master and supplies
 367 the clock to any other modules so you must connect the fiber from one interface card here. Any
 368 additional modules can use any of the other Rx ports. An example illustrating how to connect fibers
 369 into the VC709 SFPs for the case using two chips can be seen in Figure 46.

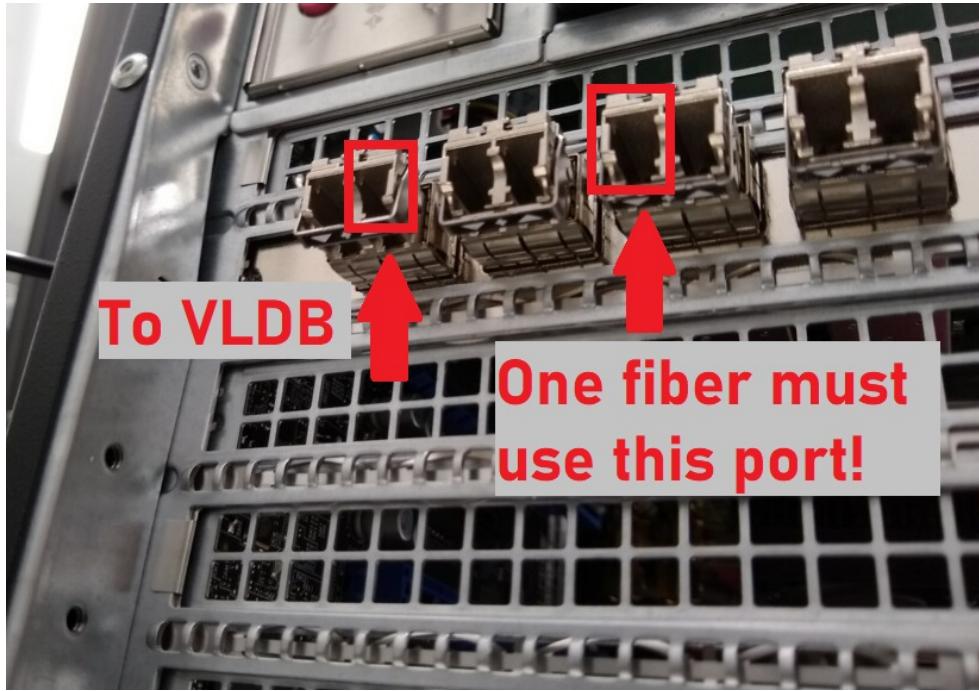


Figure 19: Ports on the VC709 SFPs that are required to be used. This picture was taken looking up towards the ceiling.

370 **Power the interface card using the included power adapter.** The interface card should be powered
371 from a wall outlet with an adaptor like this one (Appendix [B.2](#)). Any other wall outlet adaptor capable of
372 providing 5V to 14V and handle a current of about 3 A would work.

373 4.2 LpGBT readout system

374 The following hardware components needs to be connected: VLDB+ ([VLDB+ manual](#)) plus Ali Skaf's
375 break-out-board ([git repo](#)), or optoboard V0 ([optoboard repo](#)) plus Abishek Sharma's adapter board ([Adapter](#)
376 [Board Schematics](#)), and FELIX (FLX712).

377 4.2.1 VLDB+ Setup

378 This section will describe the dip switch settings, the voltage settings, how to connect the VLDB to the
379 FLX712, and how to program the VLDB.

380 **Configure VLDB dip switch as shown in Figure 22.** There are several dip switches located on the VLDB
381 that need to be set. For the readout of the RD53A, the dip switches should be set as seen in Figure 22. For
382 more information on the VLDB and what these dip switches do, you can visit the [VLDB+ manual](#).

383 **Route data from/to FLX712.** In order to do that a VTRX+ should be mounted on the VLDB+, a MT to
384 MPO adapter plus guide pins from the MT ferrule kit (Sec. [B](#)) should be used. A MTP coupler and a MTP
385 to MTP cable or MTP to 12xLC plus patch panel needs to be used to finalize the connection to FLX712

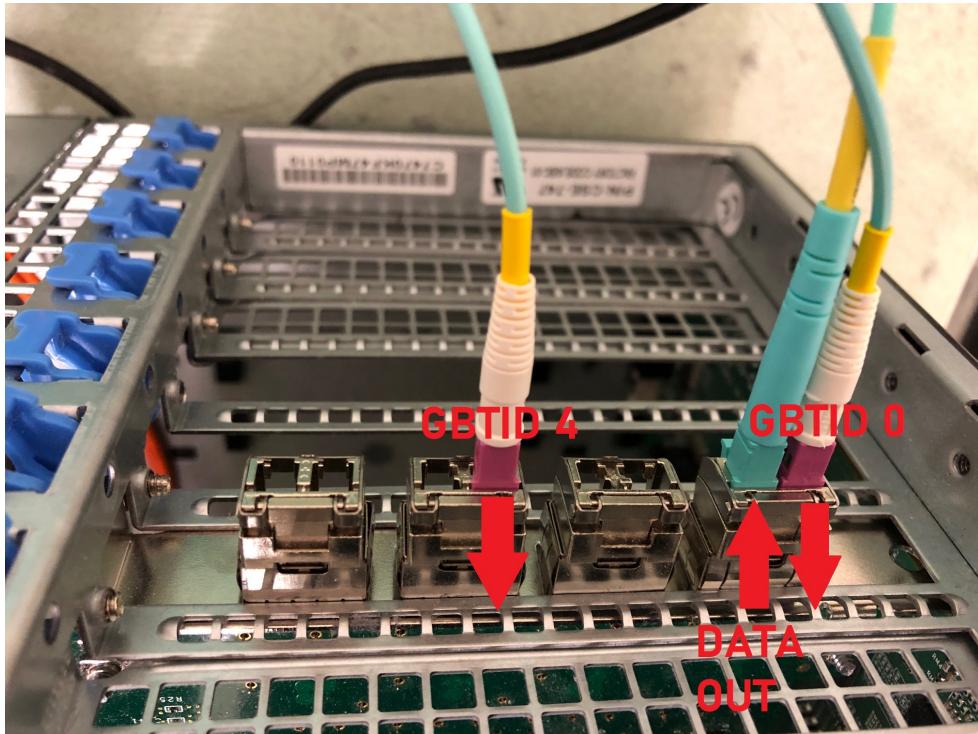


Figure 20: Example illustrating how to connect the fibers from two interface cards to the VC709. In this example, we have connected the fibers for two chips. Note that one fiber is connected to the SFP labeled 2 in Figure 45. This is required. The position to the second fiber is arbitrary. This picture was taken looking over the top of the server case.

³⁸⁶ (see Sec 4.1). Notice that alternatives in the way to adapt from the VTRX+ MT connector to MPO are
³⁸⁷ described in the manual [VLDB+ manual](#).

³⁸⁸ **Power VLDB+ with 10 V.** The VLDB will need to be powered using an external power supply. See the
³⁸⁹ [VLDB+ manual](#) for further details.

³⁹⁰ 4.2.2 Ali Skaf's Break-out-Board

³⁹¹ The break-out-board should be inserted in the VLDB+ FMC connector. The mini-DP cable(s) should be
³⁹² plugged in J12 and/or J13 connectors. In Figure 21 J13 connector has been used.

³⁹³ 4.2.3 Optoboard V0 Setup

³⁹⁴ TO DO

³⁹⁵ 4.2.4 Abishek Sharma's Adapter Board

³⁹⁶ TO DO

Not reviewed, for internal circulation only

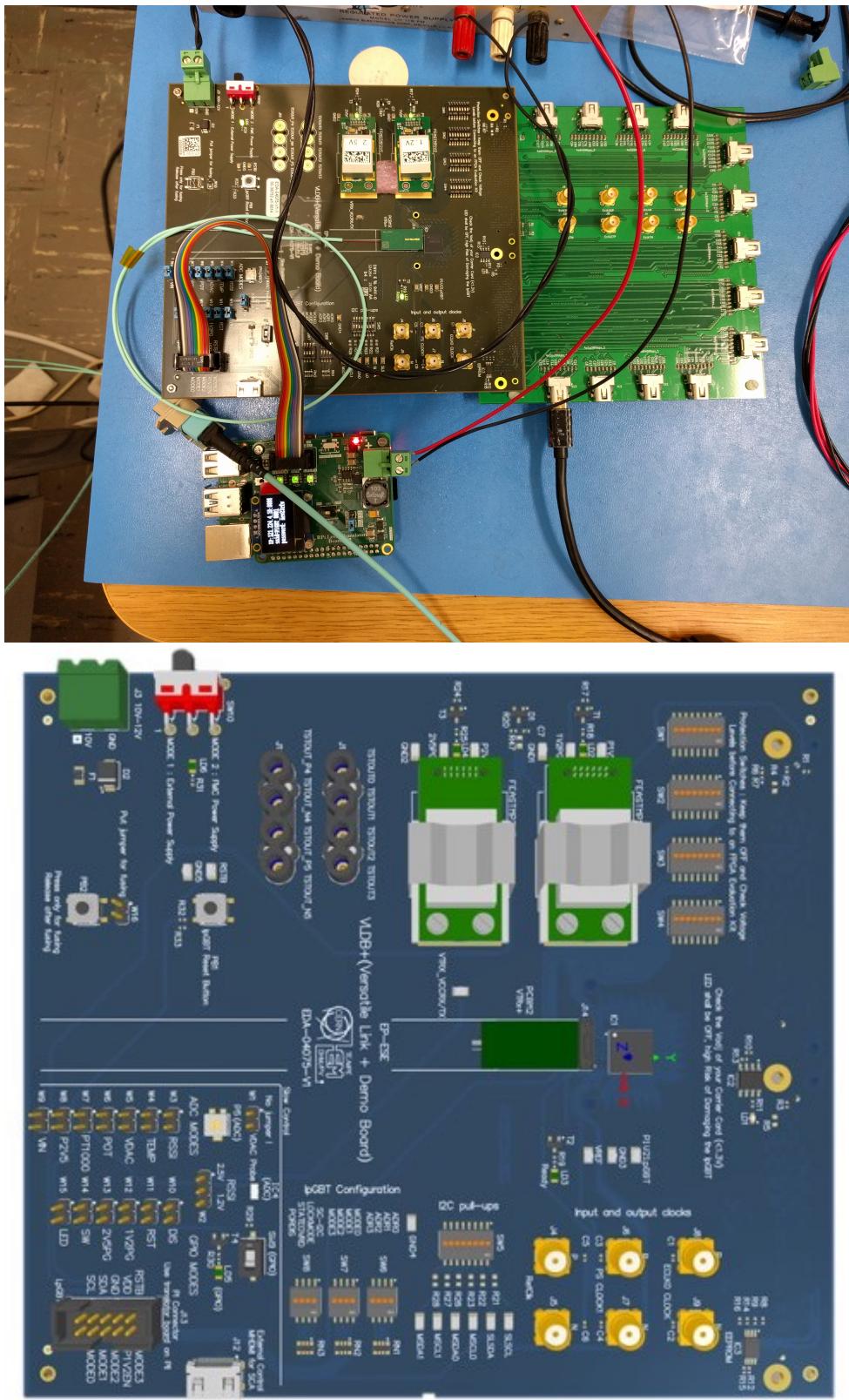


Figure 21: VLDB+ at ANL (top) and from <https://www.digikey.com/en/products/detail/mean-well-usa-inc/GST25A05-P1J/7703645>. Among all elements, VTRX+ connector (J14) is shown in the center of the picture. Three LEDs (LED2,3,4,6) are lit

Not reviewed, for internal circulation only

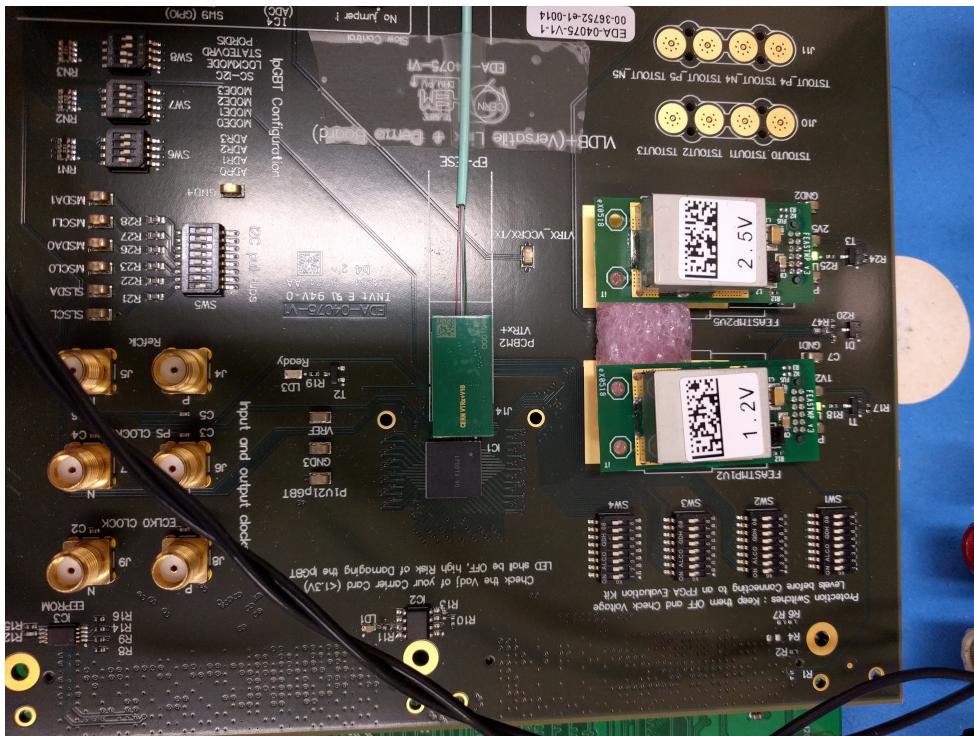


Figure 22: Correct configuration of the dip switches of the VLDB+.

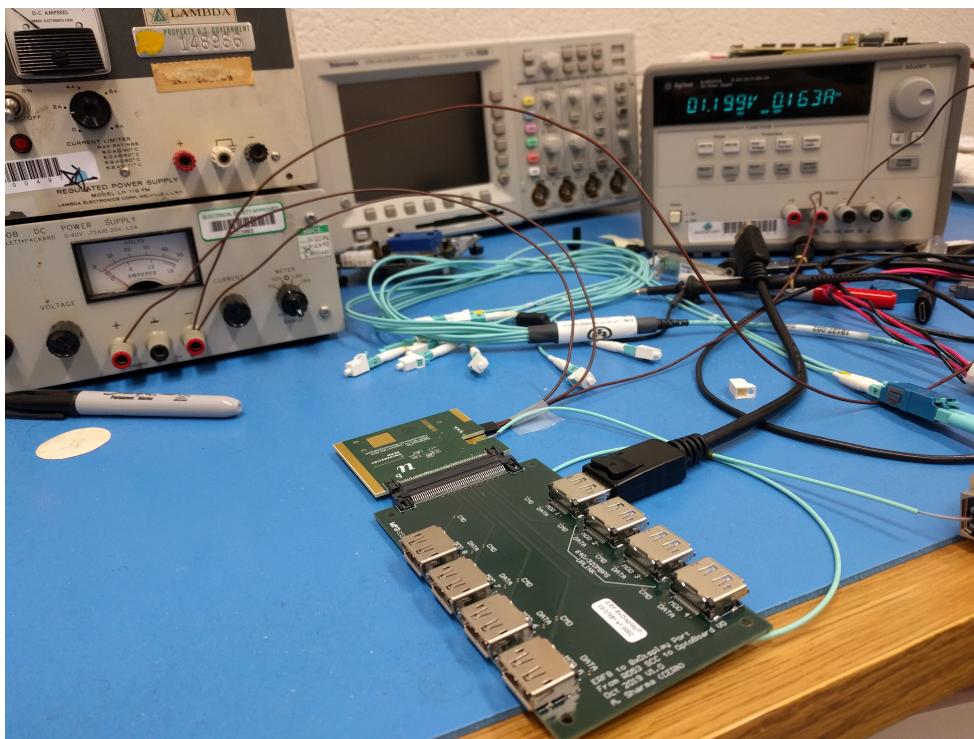


Figure 23: Setup of Optoboard V0 plus adapter board

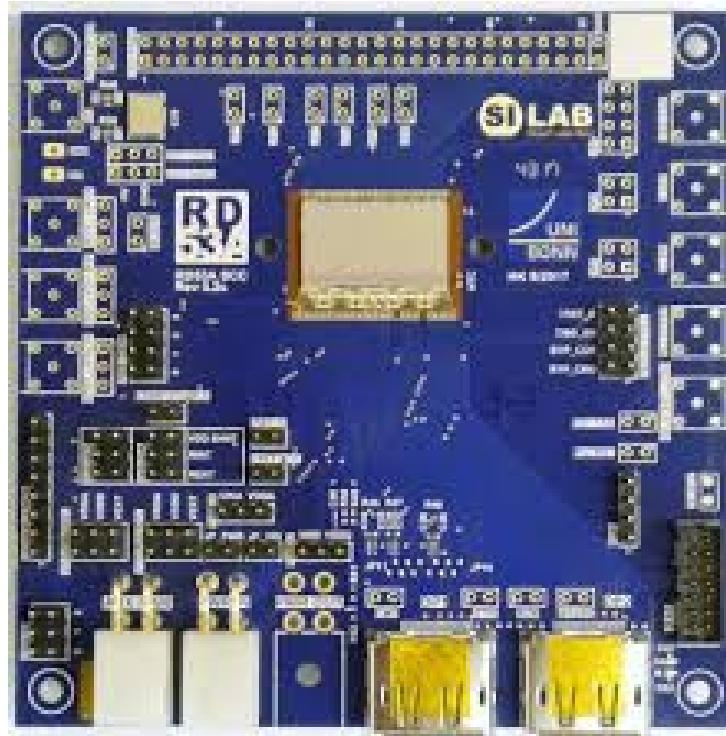


Figure 24: RD53A single chip card.

397 4.3 RD53A SCC (single chip card) for Direct and LpGBT Readout

398 This section assumes that you are using an RD53A SCC seen in Figure 24.

399 **Make sure the jumpers on RD53A SCC are set properly.** The recommended jumper setting are given
 400 in Figure 25. For more information, you can consult the [RD53a twiki](#), and the document (found on the
 401 twiki) describing the jumper settings ([SCC configuration](#)).

402 **Power the RD53A.** There are two different ways to power the RD53A:

- 403 • Using an external power supply.
- 404 • Directly from the interface card.

405 Here, we describe both of these options.

406 **Power RD53A SCC with 1.85V via the Molex connector labeled 'power in' on the SCC.** You will
 407 need to have the correct connector to power the RD53A. The connector is a 4-pin Molex connector with
 408 the two pins near the tab are positive and the two pins away from the tab are negative. The correct Molex
 409 connector to supply power to the RD53A can be seen in Figure 25.

Not reviewed, for internal circulation only

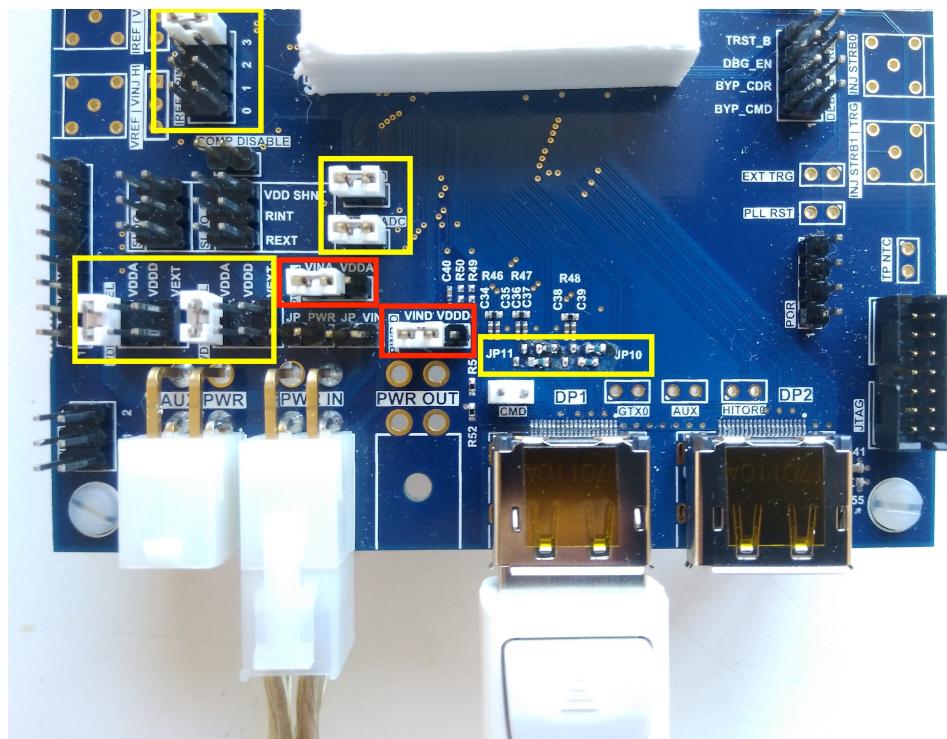


Figure 25: Recommended jumper settings for the RD53A single chip card.



Figure 26: Dongle used to program the VLDB.

410 5 Using the System

411 This section describes the steps necessary to configure the VLDB (*direct readout*), VLDB+ (*LpGBT*
 412 *readout*), YARR, FLX712, and the VC709 (*direct readout*). We will first describe programming the VLDB,
 413 the VLDB+, and the optobox V0, then discuss how to use the FELIX firmware and software. Finally, we
 414 will discuss the configuration files necessary for using YARR with the FELIX setup.

415 5.1 Configure VLDB (*direct readout*)

416 Information in this section should be used only when setting up the *direct readout* chain.

417 To program the VLDB, you will need a few things:

- 418 • The Java application used to program the VLDB
- 419 • the VLDB configuration file
- 420 • a USB dongle to connect the VLDB to the a computer running Java.

421 **Attach dongle to the VLDB.** Your VLDB should have come with a dongle like the one shown in Figure
 422 26. Connect this to the VLDB. Next, connect this to a computer that is running Java. I recommend using a
 423 **USB extension cable to do this.**

424 **Obtain the VLDB configuration file.** The VLDB configuration file can be obtained by copying and
 425 pasting the contents of B.3 into a text file. This is included directly to prevent the link from dying and the
 426 configuration becoming unavailable. Make sure to remove any line numbers or other formatting. Each pair
 427 of characters should be on a new line.

428 **Obtain the Java application.** The Java application can be obtained from **GBTx programmer gitlab**
 429 **repository**. The Java application will be the .jar file located in this directory. Download the Java application.
 430 **For more information on the application, see the README for the GBTx programmer GitLab repository.**
 431 For a Windows computer with Java installed, this application should run by clicking on the downloaded
 432 .jar file. For other OSs, there may be more steps. See the GitLab README if you have trouble.

433 When you open the java programmer application, if you see the error shown in Figure Figure 27, try
 434 closing the Java application, unplugging the dongle from the computer, reconnecting it, and starting the
 435 Java application again. You may have to do this more than once but this is rare. The GUI for the Java
 436 programmer application can be seen in Figure 28. To configure the VLDB, click the import button and
 437 load the file containing the VLDB configuration and then click write. If you receive an error here, the
 438 VLDB is likely not configured. To check to see that the VLDB configuration was successfully written,
 439 click the 'read' button; the configuration will be displayed in the GUI.

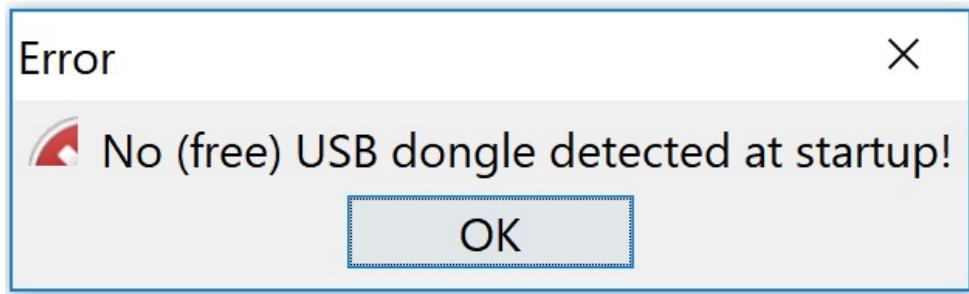


Figure 27: Error from GBTx programmer when unable to connect to VLDB.

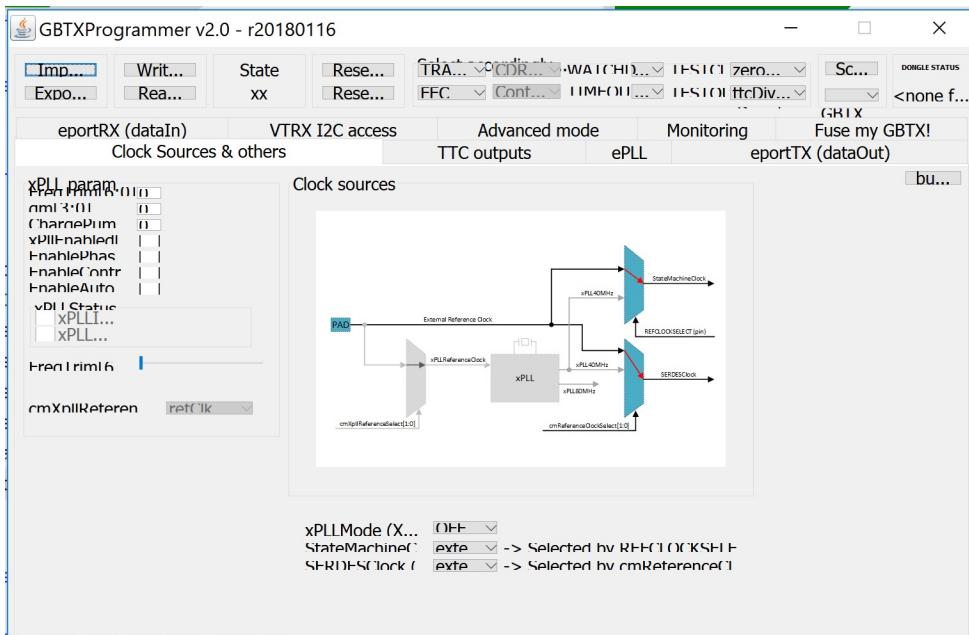


Figure 28: GBTx programmer application GUI.

5.2 Configure VLDB+ (*LpGBT readout*)

Raspberry Pi 29 to configure the VLDB+. VLDB+ can be configured in different ways and are reported in the manual. For this document it chosen to configure the VLDB+ via the provided Raspberry Pi. After powering up the Raspberry Pi with 12V the following settings are used:

- TRX
- TXdata rate = 10 Gbps
- TX encoding = FEC5
- Uplink EPRX:
 - EPRX0-3
 - Data Rate = 1280
 - Track Mode = Continues

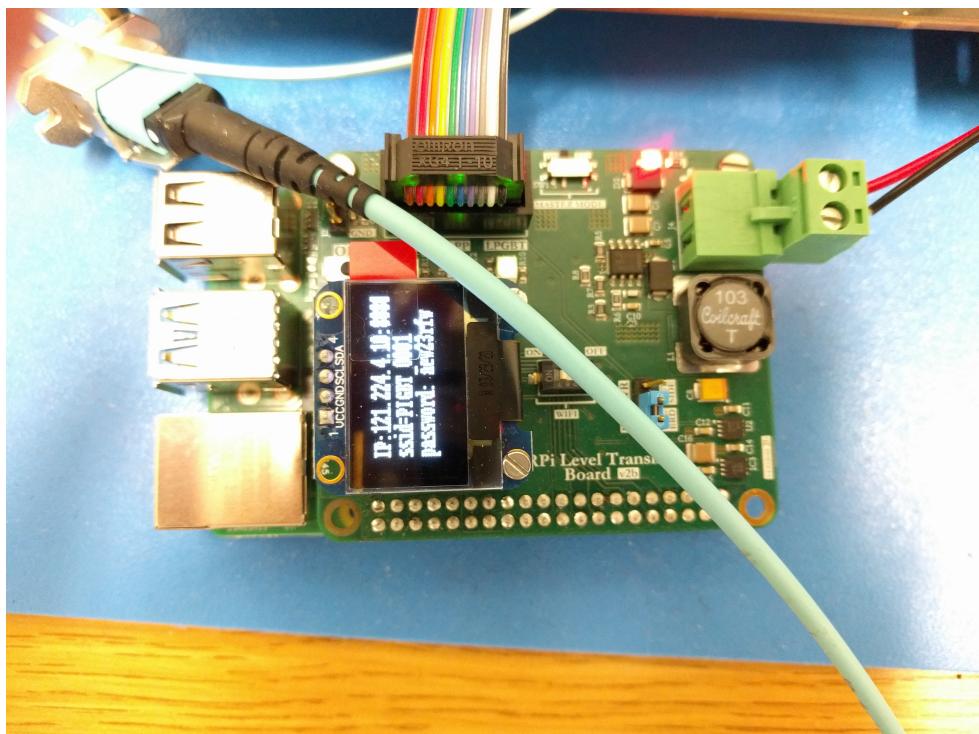


Figure 29: A picture of the Raspberry Pi provided in the VLDB+ kit after powering it up.

- 451 – Control TERM
- 452 – Control INV for EPRX2 due to a polarity swap in the break-out-board schematics
- 453 – Equalization 300 MHz
- 454 – EPRX4-6
- 455 – Data Rate off
- 456 – Track Mode = Continues
- 457 – Control TERM
- 458 – Equalization 300 MHz
- 459 • Uplink EPTX
 - 460 – Data Rate 160 Mbps
 - 461 – DDrive strength 4.0 mA
- 462 • Invert polarity uplinks in the high speed tab
- 463 Additio Further instructions are available at [VLDB+ control toolkit manual](#)

464 5.3 Configure Optoboard (V0)

465 Registers in the LpGBT chip can be configured via e-fuses (a subset of them), I^2C , or serial interface
 466 (IC/EC). In the rest of this section we will assume that a minimal configuration of registers will be loaded at
 467 power-up from fuses and the rest of registers will be configured via serial interface. E-fuses configuration
 468 can be seen here: https://gitlab.cern.ch/bat/optoboard_felix/blob/master/fuse_registers.json. The rest of
 469 registers can be configured via serial interface by running the following command²

470 `./fice_configLpGBT.sh`

471 FELIX needs to be initialized (i.e: flx-init, see [5.6](#)) prior to running fice.

472 5.4 Configure VC709 (*direct readout*)

473 **Configure the clock.** Using the modified Felix software, run flx-init. **Note:** Only run this command one
 474 time. If you run this command more than once, you will need to reboot the computer. To properly configure
 475 the clock for the FELIX firmware you will need to use the version of flx-init from the modified FELIX
 476 software described in section [3.2](#). This command will configure the clock on the transceiver reference
 477 clock to operate at 160 MHz. If you did not add the flx-init executable to your PATH environment variable,
 478 you will have to run something like

479 `$./<PATH TO FELIX SOFTWARE>/software/x86_64-centos7-gcc62-opt/
 480 flxcard/flx-init`

481 **Invert the transmitter polarities.** The command below will set invert the polarities of the first channel
 482 transmitter.

483 `$./<PATH TO FELIX SOFTWARE>/software/x86_64-centos7-gcc62-opt/
 484 flxcard/flx-config set GBT_TX_POLARITY=0x1`

485 The inversion of the polarity is needed because of a SFP rather than VTRX module are used on the VLDB
 486 board ([VLDB FDAQ](#)).

487 After configuring the clock and inverting the polarity, the three red lights on the VLDB should light up, as
 488 seen in [Figure 12](#). This indicates that the VLDB is receiving the correct signal from the VC709.

489 **Configure the elinks.** This section only gives the steps necessary to configure the appropriate elinks
 490 needed for this setup. For more information on what the elinks are and what they do, see the FELIX users
 491 manual.

492 **Run elinkconfig.** The following command should be run:

493 `$./<PATH TO FELIX SOFTWARE>/software/x86_64-centos7-gcc62-opt/
 494 elinkconfig/elinkconfig`

495 You should see something similar to what is shown in [Figure 30](#). If your GUI looks different, the FELIX
 496 drivers may not be installed properly. Run `cat /proc/flx` to make sure the FELIX drivers are installed
 497 properly and the FPGA is detected and programmed. To configure the elinks for RD53A readout:

² `iceconfigLpGBT.sh` script is available at [CERNBOX](#)

Not reviewed, for internal circulation only

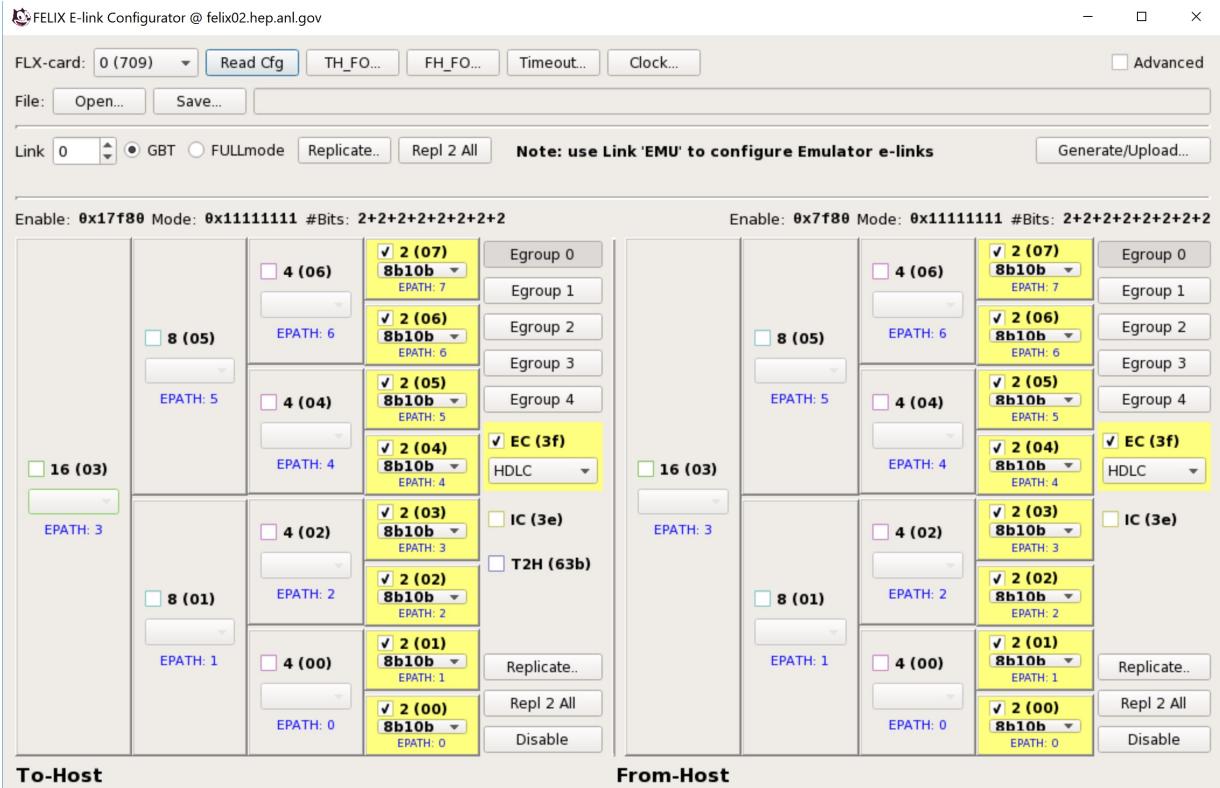


Figure 30: Elinkconfig GUI used to configure elinks for the FELIX system. The figure shows the GUI upon launching elinkconfig for the first time. If the GUI looks different than this, the FPGA may not be detected. This is usually due to a problem with the drivers.

- 498 • Select 'Full mode' radio button in the top-left. The FELIX firmware for RD53A readout does not support GBT mode.
- 499
- 500 • Disable emulators by clicking the TH_FO and FH_FO buttons at the top of the window. Select 'none', then 'OK'. This disables the emulators. You will not be able to receive data if they are not disabled.
- 501
- 502 • Set clock type to be local by clicking the 'Clock' button at the top of the window. Select 'local', and then 'OK'.
- 503
- 504 • Enable appropriate 4-bit wide elinks (more on this below) and be sure to select 8b10. If you select 'direct', the RD53A will not get the correct signal from the VC709 and fail to sync.
- 505
- 506

507 The appropriate 4-bit elinks that you need to enable will depend upon your setup. Each RD53A will use a
 508 single 4-bit wide elink. These elinks each have an elink ID as indicated in Figure 32. These elink IDs
 509 correspond to ports on the VLDB as shown in Figure 31. For example, in Figure 32, we have two elinks
 510 configured: one for EPATH 0 and one for EPATH 6. In this example, we are reading out two chips: one
 511 connected to the port labeled 0 and another connected to the port labeled 6 in 31. Enable the appropriate
 512 elinks for your setup.

513 With the elinks configured, we can now move on to configuring the FPGA.

Not reviewed, for internal circulation only

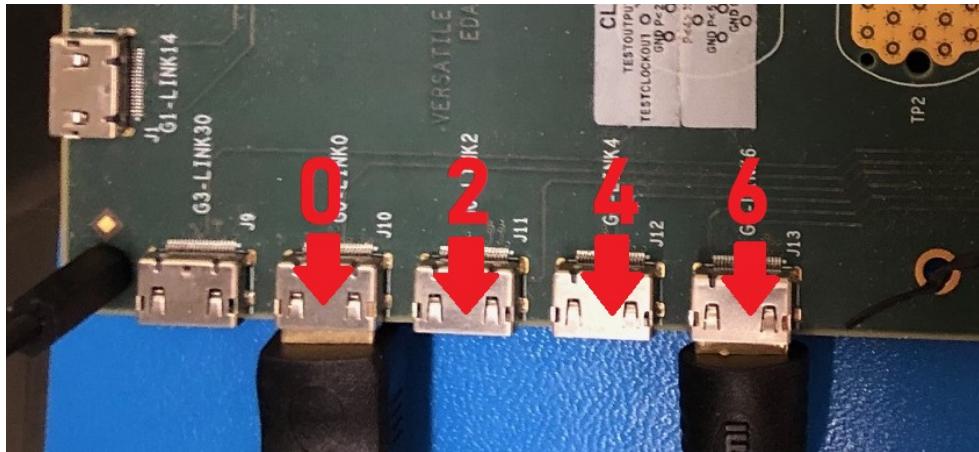


Figure 31: The Tx number for each port is given by 0, 2, 4, 6. This corresponds to the EPATH number in elinkconfig.

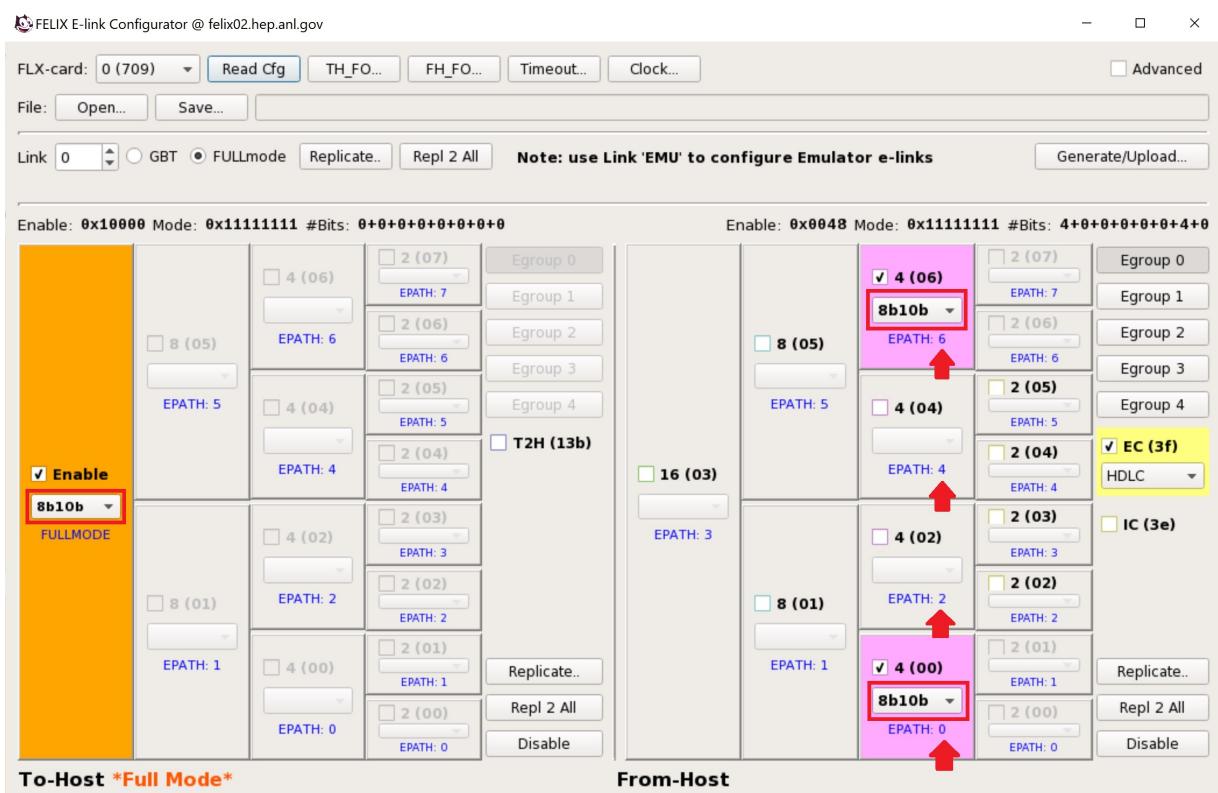


Figure 32: Each 4-bit wide elink has a corresponding elink ID as shown here. These numbers serve to identify a particular RD53A chip connected to the VLDB.

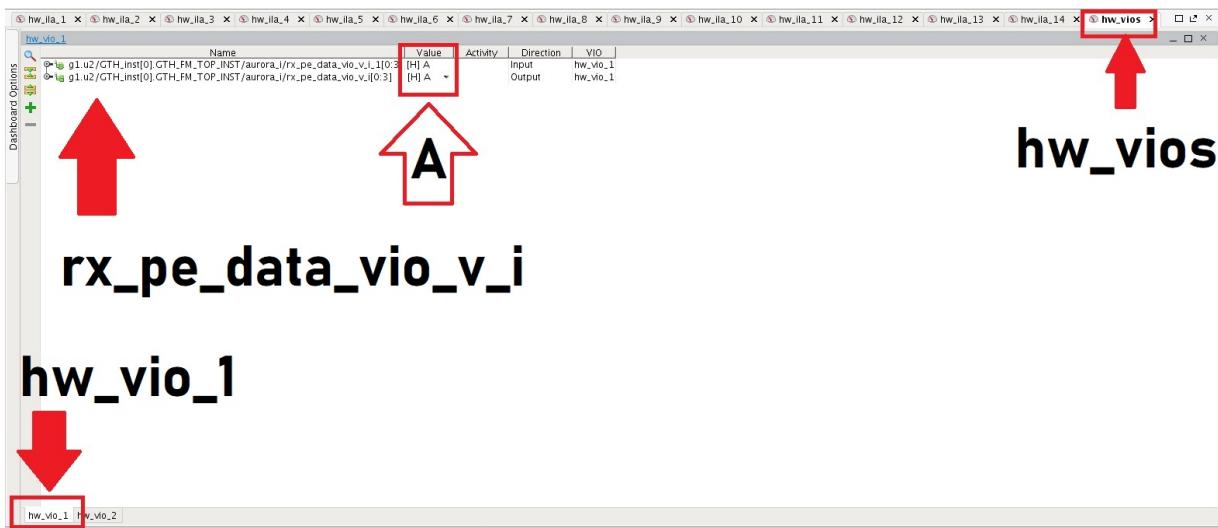


Figure 33: hw_vio_1 (see bottom tab) in hw_vio (top tab on far right). This is used to tell Felix what lanes we expect to receive RD53a data on. This is set to A, here, telling Felix that we expect data on GBT ID 0 and GBT ID 2.

514 VIO in Vivado. This section describes using vio in Vivado to configure registers in the FELIX firmware.
515 To begin, open Vivado and go to the hardware manager. From here, you will need to upload the debug
516 (.ltx) file that came with the .bit file. To do this:

- 517** • Open the Hardware Manager
- 518** • click 'open target' in the green bar near the top and then click 'auto connect'. This can be seen in
519 Figure 9.
- 520** • Click 'Add probe files'
- 521** • Navigate and select the .ltx files that came with the .bit file
- 522** • Click OK and the probe files should load.

523 Navigate to hw_vios. You should see several tabs like hw_il_1, hw_il_2, etc. The hw_vios should be
524 the last tab. Click on the tab. You should now see the panel shown in figure 33. There are several registers
525 that can be changed using vio but we only need to worry about two.

526 Use rx_pe_data_vio_i to tell the FELIX firmware what lanes you are using. In order for the FELIX
527 firmware to work properly, you need to tell it what lanes are in use. This is done by setting
528 rx_pe_data_vio_i in hw_vio_1 to a 4-bit number (in hex) that will have a 1 (when written in binary) for
529 each Rx port you are using on the VC709. Each port has a GBT ID value as given in Figure 34. In our
530 example, we are using two chips. One is using the SFP associated with GBT ID 0 and the other is using the
531 SFP associated with GBT ID 2. Therefore, we set rx_pe_data_vio_i to the A. This is 1010 in hex. So, the
532 most significant bit is used to tell the FELIX firmware that we are using the SFP associated with GBT ID 0
533 and the least significant bit is used to tell the FELIX firmware that we are using the SFP associated with
534 GBT ID 3. If you have only one chip connected, it will be connected to the SFP with associated GBT ID 2
535 and therefore, you will need to set rx_pe_data_vio_i to 2 (0010 in binary).

Not reviewed, for internal circulation only

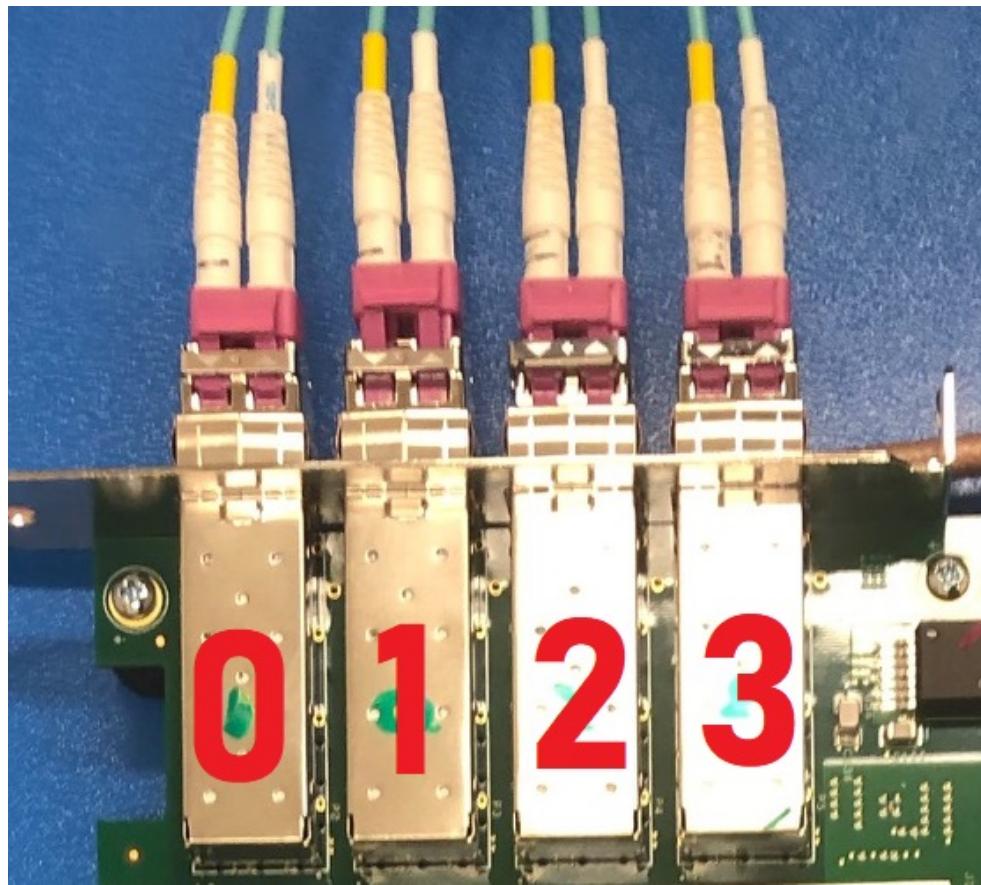


Figure 34: GBT ID numbering for the VC709 SFPs.

536 **Perform a soft reset.** Navigate to hw_vio_2 by clicking on the tab indicated in Figure 35. Set
 537 soft_rxrst_all_vio_i to 1 and then set it back to 0. If you have issues getting the RD53A to sync properly, it
 538 is a good idea to power-cycle the module and perform a soft reset.

539 Verify the Setup

540 If everything is setup properly, the Rd53a chips and the Felix board will be synchronised. To verify that
 541 this is the case, you can use the ILA interface. To do this,

- 542 • Open Vivado
- 543 • Click 'Open Hardware Manager'
- 544 • In the green bar, click 'Open target', then 'auto connect'
- 545 • Click 'Configure probe files and refresh'
- 546 • load the probe files that are associated with the bit file you used to program the FPGA.
- 547 • Go to hw_il_3 as seen in Figure 36.
- 548 • Trigger hw_il_3

Not reviewed, for internal circulation only

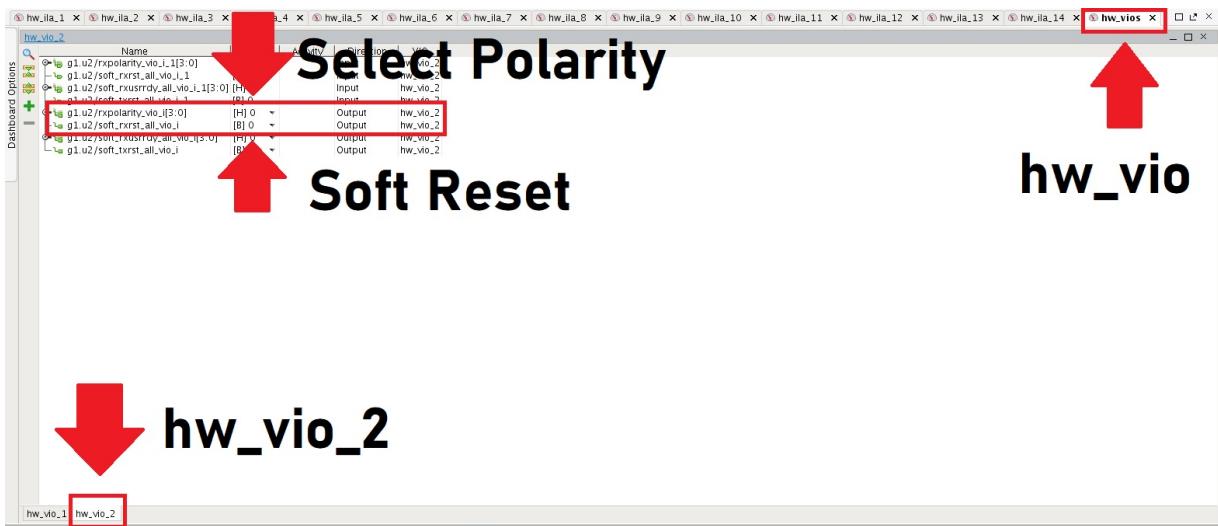


Figure 35: hw_vio_2 allows the user to perform a soft rest. In the case of problems getting the RD53A to sync, it is a good idea to try power-cycling the module and performing a soft reset.

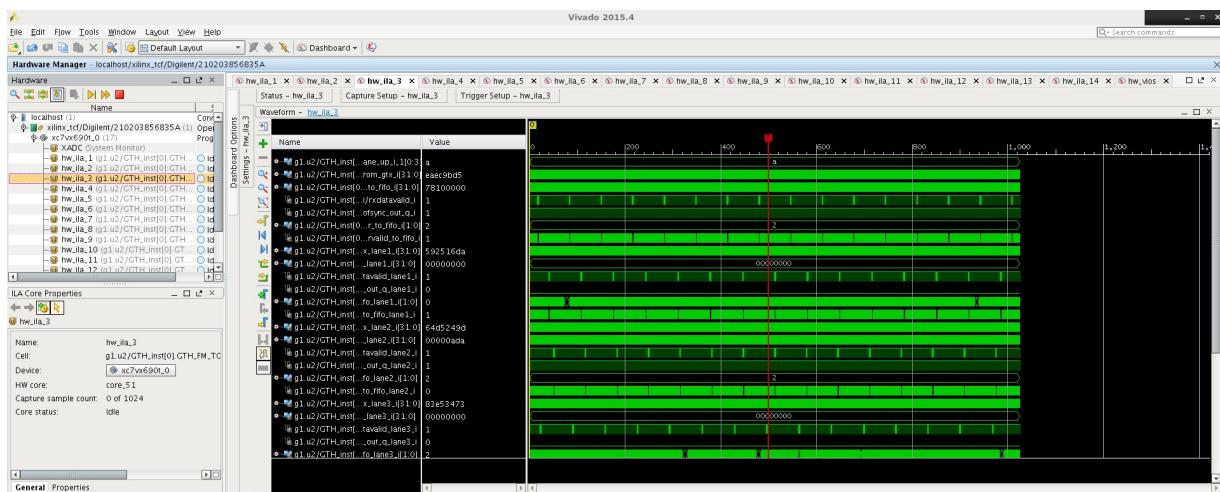


Figure 36: ILA probes in Vivado after triggering on hw_il_3. The first entry for rx_lane_up_i displays A, indicating that lanes 0 and 2 are locked.

- 549 If everything is setup, you will see something like Figure 37. In this figure, we have two chips connected,
 550 one to GBT ID 0 and one to GBT ID 2. We therefore see A which is 1010 in binary. Each 1 lets us know
 551 that that lane is synchronized properly.

5.5 Configure FLX712 (*direct readout*)

Configure the clock.

```
554 $ ./<PATH TO FELIX SOFTWARE>/software/x86_64-centos7-gcc62-opt/bin/
555   flx-init
```

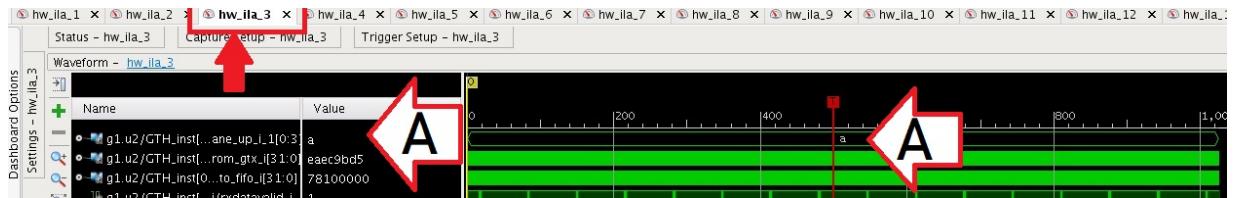


Figure 37: rx_lane_up_i displays A, indicating that lanes 0 and 2 are locked.

556 This step is similar to what described in Sec. 5.4. However, as mentioned in Sec. 3.2, the official version
 557 of the FELIX software can be used. The transceiver reference clock is set to operate at 240 MHz in this
 558 case.

559 Invert the transmitter polarities.

560 \$./<PATH TO FELIX SOFTWARE>/software/x86_64-centos7-gcc62-opt/bin/
 561 flix-config set GBT_TX_POLARITY=0x1

562 This step mirror exactly what is described in Sec. 5.4.

563 After configuring the clock and inverting the polarity, the three red lights on the VLDB should light up
 564 (Figure 12).

565 Run elinkconfig.

566 \$./<PATH TO FELIX SOFTWARE>/software/x86_64-centos7-gcc62-opt/bin/
 567 elinkconfig

568 In the configuration [flix712_4gbt_fullmode.elc file](#) should be loaded by clicking on the "Open" button.
 569 Disable emulators and set the clock to local as indicated in Sec. 5.4.

570 Notice that the step "VIO in Vivado" in Sec. 5.4 is not needed in this case.

571 In order to verify the setup, follow the same steps as in Sec. 5.4

572 5.6 Configure FLX712 (*LpGBT readout*)

573 Bit file, .ltx and script files are available in this [CERN box Configure the clock](#).

574 \$./<PATH TO FELIX SOFTWARE>/software/x86_64-centos7-gcc62-opt/flix-
 575 card/flix-init

576 This step is the same as what described in Sec. 5.5

577 Notice that the step "Invert the transmitter polarities" in Sec. 5.5 is not needed in this case, since VLDB+
 578 is used in the place of the VLDB.

579 After configuring the clock the three lights on the VLDB+ should be lit (Figure 21).

580 Run elinkconfig.

581 \$./<PATH TO FELIX SOFTWARE>/software/x86_64-centos7-gcc62-opt/bin/
 582 elinkconfig

583 The configuration Phase2Lpgbt_fromHost4b.(y)elc file should be loaded by clicking on the "Open" button.
 584 Disable emulators and set the clock to local as indicated in Sec. 5.4. Moreover, run the following commands
 585 to finalize the elinkconfig settings ³ ⁴

```
586 flx-config set DECODING_LINK[N]_EGROUP[G]_CTRL_PATH_ENCODING=0x3
587 flx-config set DECODING_LINK[N]_EGROUP[G]_CTRL_EPATH_ENA=0x1
588 flx-config set DECODING_LINKN_EGROUP[G]_CTRL_EPATH_WIDTH=0x4
589 flx-config set DECODING_REVERSE_10B=0x1
```

590 where N is the link number (e.g: 00) and G is the group number (e.g: 0)

591 **Verify the Setup** If you are using VLDB+ you should follow instructions in 5.2. If you are using an
 592 optobox V0 you should follow instructions in 5.3. Assuming that everything was done correctly you should
 593 verify that the uplink in FELIX is aligned. In order to verify that run the following command:

```
594 flx-info LINK
```

595 The output should look like the example below if the first link is connected in FELIX

```
596 Link alignment status
597 -----
598 (entire FLX-712/711):
599 Channel | 0 1
600 -----
601 Aligned | YES NO
602
603 Channel | 2 3
604 -----
605 Aligned | NO NO
```

606 At this point the RD53A should be synchronized. The drawn current on the chip should be about 0.51 A.
 607 Moreover, the 64b66b module should be locked and idles from the RD53A should be correctly received. If
 608 that is not the case try power cycling the RD53A chip. In order to verify the status of the setup run the
 609 following commands ⁵

```
610 flx-config list | grep DECODING_LINK_ALIGNED_[N]
```

611 Each bit is associated to an Epath. Please notice that, since only Epath=0,4,...,24 are used for hit data, the
 612 lowest significant bit of each hexadecimal is meaningful. See below for example of output when only one
 613 RD53A is enabled.

```
614 0x2180 [R 57:00] DECODING_LINK_ALIGNED_00 0x0000000000000001 \
615 Every bit corresponds to an E-link on one (lp)GBT or FULL-mode
```

616 If you are using an optoboard V0 (Sec. 5.3) you need to invert the uplink polarity and switch to FEC12 by
 617 running the following commands:

³ commands are also available in config_decoding.sh script

⁴ As mentioned in Sec. 3 the software package is not stable yet. This step can be bypassed once the software fully complies with the Phase2 F/W formats

⁵ commands are also available in monitoring.sh script

```

==> JUMBO version
Chunk types: BOTH="<<", FIRST="++", LAST="&&", MIDDLE="==", TIMEOUT="]]", NULL="@@",
OUTOFCBAND="#" and "TEC" for chunk truncation/error/CRCerr.
Blocksize: 1024
==> BLOCK 0 (E=000=0-0-0 seq=1): 1016 databytes
14 11 0 2 f7 ff 0 3c f7 ff 6 3c ff f7 7 3c ff f7 1 3c 7f ff 8 3c f7 ff e 3c 77 ff d 3c
f7 f7 10 3c 77 ff 1a 3c f7 ff 13 3c 77 7f 1d 3c f7 ff 27 3c ff f7 2d 3c f7 ff 2a 3c ff f7 29 3c
7f 7f 39 3c ff 7f 42 3c 7f ff 44 3c ff f7 40 3c f7 ff 41 3c ff 7f 4c 3c ff f7 4e 3c ff f7 4a 3c
f7 ff 52 3c f7 ff 58 3c f7 ff 57 3c 7f 7f 5d 3c 7f ff 66 3c 77 ff 62 3c 7f ff 64 3c ff 7f 60 3c
7f f7 61 3c 77 7f 6c 3c ff f7 6e 3c f7 ff 6a 3c f7 f7 76 3c 7f ff 70 3c ff f7 72 3c 77 ff 77 3c
7f 7f 7e 3c f7 ff 78 3c 77 ff 7c 3c 77 7f 7f 3c 7f f7 79 3c 77 7f 84 3c 7f ff 86 3c 7f ff 82 3c
ff f7 83 3c ff f7 8c 3c 7f ff 8e 3c f7 f7 8a 3c 7f 77 8d 3c e7 fe 89 3c 7f 77 8b 3c 7f ff 97 3c
ff 7f 9b 3c ff 7f ae 3c f7 f7 99 3c 7f ff af 3c 7f ff c0 3c f7 f7 d4 3c 7f ff d6 3c f7 77 d2 3c
7f ff d3 3c 77 7f dc 3c ff f7 d1 3c f7 ff da 3c ff f7 dd 3c 77 f7 e9 3c 7f 7f db 3c f7 ff 0 3d
7f 77 19 3d ff 77 4d 3d ff 7f 39 3d f7 ff 63 3d 0 0 0 ... 0

```

Figure 38: Fdaq/fcheck output after calibration injection plus trigger commands have been sent to the RD53A chip.
See text for more details

```

618 fgpolarity -r set
619 flx-config set GBT_DATA_RXFORMAT2=0x1

```

620 The RD53A should be configured by running the command below:

```
621 ./config_rd53a.sh [D] [NL]
```

622 where *D* is the down e-link number (e.g: 0) and *NL* is the number of active lanes in RD53A. At this point
623 a simple calibration injection plus triggers can be performed via low-level felix software tools by running
624 the following two commands in parallel⁶.

```
625 [terminal 1] ./calplustrigger.sh [D]
626 [terminal 2] fdaq -d 1 -t 10 -D test;
```

627 Data captured via *fdaq* can be visualized by running the following command:

```
628 fcheck -F 1 test*.dat
```

629 The output should look similar to the one shown in Figure 38:

630 If three lanes per RD53A are active the channel bonding functionality should be activated by running the
631 following command⁷:

```
632 flx-config set DECODING_DISEGROUP=0xC
633 flx-config set DECODING_LINK_[N]_CBOPT=0x3
```

634 The status of the channel bonding state machine can be checked by running the following command

```
635 flx-config list | grep DECODING_LINK_[N]_CB_DESKEWED
```

⁶ Notice commands need to be run in parallel (i.e: in two different terminal) as the data from RD53A is not stored anywhere

⁷ first command disables egroup 2 (and 3). This is temporary because of a possible bug in FELIX.

```

==> JUMBO version
Chunk types: BOTH="<<", FIRST="++", LAST="&&", MIDDLE="==", TIMEOUT="]]",
NULL="@@",
OUTOFCBAND="##" and "TEC" for chunk truncation/error/CRCerr.
Blocksize: 1024
==> BLOCK 0 (E=001=0-0-1 seq=0): 1016 databytes
0 8 f2 b4 0 0 88 0 0 8 f2 b4 0 0 88 0 0 8 f2 b4 0 0 88 0
0 8 f2 b4 0 0 88 0 0 8 f2 b4 0 0 88 0 0 8 f2 b4 0 0 88 0
...
0 8 f2 b4 0 0 88 0 0 8 f2 b4 0 0 88 0 0 8 f2 b4 0 0 88 0
0 8 f2 b4 0 0 88 0 0 8 f2 b4 0 0 88 0 0 8 f2 b4 0 0 88 0

```

Figure 39: Fdaq/fcheck output after the register monitoring has been activated. The periodic aurora code $B4$, the status code F , register addresses 130 and 136 and register values of 0 can be extracted from the output. Notice that the output has been published on the second e-link of the group (i.e: e-link number 1). See text for details

636 As previously stated, only the lowest significant bit of each hexadecimal is meaningful. RD53A with two
 637 or four active lanes is not supported by the current version firmware. Please ask if you would like that
 638 feature to be supported.

639 More details are available in [18.12.2020, 07.05.2021 ITk Pixel M. Trovato's Presentations](#).

640 Given that the Felix software is still work in progress scans via YARR cannot be performed for the *LpGBT*
 641 *readout*.

642 **Service data.** If you are interested in start monitoring register values of the RD53A you should run the
 643 following commands:

```

644 fupload -I0 -G0 -w4 -t 0 -D commands/
645   RD53A_write_commands_register_config_reg44.txt
646 fupload -I0 -G0 -w4 -t 0 -D commands/RD53A_write_commands_GlobalPulse
647   .txt

```

648 By default one service data frame is expected by the RD53A every 50 data frames⁸. The service data
 649 contain values for the registers 130 and 136 (lane 0). Please consult the RD53A manual [RD53A user](#)
 650 [manual](#) to change the register addresses to be automatically queried and the interval between service frames.
 651 Once the automatic monitoring is activated you look at the service data frames by running the following
 652 command:

```

653 fdaq -d 1 -t 10 -D test_autoread;
654 fcheck test_autoread*.dat

```

655 The output should be similar to the one in Figure 39 Other service frames (e.g: from read register
 656 commands) can be received in FELIX by running the following command:

```

657 fupload -I0 -G0 -R 1000 -w4 -t 0 -D commands/readreg61.txt

```

658 It may be wise to mask out the block number 3 (ie: $B4$ as per [Aurora 64b66b protocol spec](#)) before issuing
 659 the above command. Masking can be done by running the following command:

⁸ We see the interval to be eight times larger. We also see that two service frames are also being outputted in a row. RD53A experts have been notified

```
==> JUMBO version
Chunk types: BOTH=<<, FIRST="++", LAST="&&", MIDDLE="==", TIMEOUT="]]", NULL="@@", OUTOFBAND="##" and "TEC" for chunk truncation/error/CRCerror.
Blocksize: 1024
==> BLOCK 0 (E=001=0-0-1 seq=9): 1016 databytes.
0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10
0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10
0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10

0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10
0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10
0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10
0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10 0 f4 0 d2 4 0 3d 10
```

Figure 40: Fdaq/fcheck output after the register monitoring has been activated, the aurora block number 3 (ie: $B4$) has been masked, and the read register command has been issued. The periodic aurora code $D2$, the status code 0, both register addresses 61 can be extracted from the output. Notice that the output has been published on the second e-link of the group (i.e: e-link number 1). See text for details

```
660 flx-config set DECODING_MASK64B66BKBLOCK=0x3
```

661 The output should look similar to the one in Figure 40 Please notice that any of the five aurora possible
662 blocks from the RD53A can be ignored.

663 5.7 Configuring Yarr (*direct readout*)

Now, that everything is setup, we need to configure YARR so that it knows how to communicate with each chip. To do this we need to modify the YARR connectivity, controller, and chip configuration files.

666 The YARR directory should look something like this:

```
667 YARR
668     |_build
669     ...
670     |_configs
671     ...
```

672 You should have a configs directory that contains the configuration files, a build directory (if you compiled
673 YARR following the steps in [3](#), and many other directories and files. We will make a soft link to the
674 configuration files in the build directory using

```
675 $ ln -s configs build/  
676 $ cd build
```

677 This isn't strictly necessary but we will assume that this is the case in the steps that follow.

Create a configuration file for each RD53A module. Each RD53A module needs to be configured in order to work properly. The configuration is handled by configuring registers on each chip. The values that each register will hold is contained in a JSON file. The version of YARR that we are using has a default configuration file in configs/defaults/defaultrd53a.json. If no chip configuration file is specified, YARR will use this to set the RD53A registers. In our case, we do not want to use the default configuration so we will need to make another. To do this, move to the root YARR directory and run:

```

    "CdrCpIbias": 50,
    "CdrEnGck": 0,
    "CdrPdDel": 8,
    "CdrPdSel": 0,
    "CdrSelDelClk": 0,
    "CdrSelSerClk": 0,
    "CdrVcoGain": 3,
    "ChSyncLock": 16,
    "ChSyncPhase": 0,
    "ChSyncUnlock": 8,
    "ClkDelay": 0,
```

Figure 41: Felix expects the RD53a output output data at 1.28 GHz. To set this for a pa

```

    "Parameter": {
        "ChipId": 0,
        "InjCap": 8.2,
        "Name": "JohnDoe_0",
        "VcalPar": [-1.0, 0.195, 0.0, 0.0]
    },
```

Figure 42

```

684 $ cp configs/defaults/default_rd53a.json configs/rd53a_0.json
685 $ cp configs/defaults/default_rd53a.json configs/rd53a_6.json
```

686 Here, we made two copies because in our example, we have two RD53A modules. You can make only one
687 copy or even just modify the default configuration directly.

688 **Set value for the CdrSelSerClk field to 0 for each chip configuration file.** The CdrSelSerClk register
689 sets the data rate for the RD53A. Setting this register to 0 tells the RD53A to send data out at 1.28GHz.
690 For more information on the RD53A registers and what they do, consult the [RD53A user manual](#). This
691 register and the correct value can be see in Figure 41.

692 **If you have multiple modules, change the value for the Name field so that it is unique for each module.**
693 The Name field is used by YARR when naming the output for each module. If the name is the same for
694 both chips, the output for one chip will overwright the output for the other. To prevent this and help us
695 keep track of which module is which, we set the Name field to something descriptive. I suggest using a
696 number associated to the port on the VLDB that the module is connected to. The Name filed in the RD53A
697 configuration file can be seen in Figure 42.

698 **Modify the YARR connectivity configuration file to reflect your setup and to tell YARR to use the new**

699 **module configuration files created in the last step.** The connectivity file is used to tell YARR how to communicate with each module and what configuration file to use to set the registers in each module. The default 700 connectivity configuration file for the RD53A is located in configs/connectivity/example_rd53a_setup.json. 701 You should modify this file so that it looks like the file below:

```
703 {
704     "chipType" : "RD53A",
705     "chips" : [
706         {
707             "config" : "configs/defaults/rd53a_0.json",
708             "tx" : 0,
709             "rx" : 0,
710             "enable" : 1
711         },
712         {
713             "config" : "configs/defaults/rd53a_6.json",
714             "tx" : 6,
715             "rx" : 128,
716             "enable" : 1
717         }
718     ]
719 }
```

720 If you only have one module, you will only need one value for the chips field.

721 There are three fields that we will need to make sure are set properly: rx ,tx, and config. The config field 722 should match the path to the module configuration that you made in the first step. The rx and tx fields will 723 depend upon your setup. The tx field tells YARR which port on the VLDB to send data out on and should 724 be set to the number given in Figure 44 which is the same as the EPATH in Figure 32. In our example, we 725 have an interface card attached to VLDB ports 0 and 6. This is reflected in the connectivity configuration 726 above. The rx field tells YARR what SFP on the VC709 the data for that chip will be returned on. The rx 727 field should be set to the number displayed in Figure 45 multiplied by 64. In our example, we are using the 728 ports with GBT ID 0 and 2. Therefore, we set the rx fields to $0 * 64 = 0$ and $2 * 64 = 128$. This is reflected 729 in the above connectivity configuration file. If you have only one module, you would need to use the SFP 730 with GBT ID 2 so your rx field would have a value of 128.

731 **Create a YARR controller configuration file.** The controller configuration file tells YARR how to communicate with FELIX using NetIO. The controller configuration file is located at configs/controller/felix.cfg. 732 You should not need to modify this file. I have also reproduced the configuration below.

```
734 {
735     "ctrlCfg": {
736         "type": "Netio",
737         "cfg": {
738             "NetIO": {
739                 "host": "127.0.0.1",
740                 "txport": 12340,
741                 "rxport": 12350,
742                 "manchester": false,
```

```

743     "flip": false,
744     "extend": false,
745     "fetype": "rd53a"
746   }
747 }
748 }
749 }
```

750 The type field tells YARR that it needs to use NetIO; the host is set to a loop back address for the server,
 751 the txport and rxport are determined by the FELIX software; the manchester, flip, and extend fields are
 752 FEI4 specifically and fetype is required to enable features required for the RD53A.

753 5.8 Running YARR

754 We are finally ready to test the system. To do this, you will need to start two separate terminals: one for
 755 YARR and one for the FELIX software.

756 **In one terminal, run felixcore.** You will need to setup the environment for the FELIX software in this
 757 terminal. TO run felixcore, run:

```
758 $ felixcore -t 1 --data-interface lo
```

759 where lo is the default name of the loop back address socket for CentOS7 machines.

760 **In the other terminal, run YARR.** In this terminal, you will need to setup devtoolset-7. To run YARR,
 761 move into the build directory and run:

```
762 $ bin/scanConsole -r configs/controller/felix.json -c configs/
763   connectivity/example_rd53a_setup.json -s configs/scans/rd53a/
764   std_digitalscan.json -p
```

765 This will perform a digital scan. For more information on the types on scans that can be performed, see the
 766 YARR readthedocs page⁹.

767 5.9 Notes

- 768 • If you run flix-init more than once, the FPGA will go into a bad state and you will have to reboot the
 769 computer.
- 770 • If you reboot the computer, the FPGA will may not be programmed any more.
- 771 • If you are using the FELIX drivers from the FELIX GitLab repository, you will have to reload them
 772 each time you reboot the computer.
- 773 • YARR needs to be run **with** devtoolset-7 enabled and **without** the FELIX environment setup.
- 774 • felixcore needs to be run **with** the FELIX environment setup and **without** devtoolset-7 enabled.

⁹ <https://yarr.readthedocs.io/en/devel/rd53a/>

⁷⁷⁵ **6 Conclusions**

⁷⁷⁶ The following items are still missing:

- ⁷⁷⁷ • TO DO

[Not reviewed, for internal circulation only]

778 References

- 779 [1] M. Garcia-Sciveres, *The RD53A Integrated Circuit*, tech. rep. CERN-RD53-PUB-17-001, CERN,
780 2017, URL: <https://cds.cern.ch/record/2287593> (cit. on p. 3).
- 781 [2] A. Collaboration, *Technical Design Report for the ATLAS Inner Tracker Pixel Detector*, tech. rep.
782 CERN-LHCC-2017-021. ATLAS-TDR-030, CERN, 2017, URL: <https://cds.cern.ch/record/2285585> (cit. on p. 3).
- 784 [3] A. Collaboration, *Technical Design Report for the Phase-II Upgrade of the ATLAS TDAQ System*,
785 tech. rep. CERN-LHCC-2017-020. ATLAS-TDR-029, CERN, 2017, URL: <https://cds.cern.ch/record/2285584> (cit. on p. 3).
- 787 [4] J. T. Anderson et al., *FELIX: a PCIe based high-throughput approach for interfacing front-end and*
788 *trigger electronics in the ATLAS Upgrade framework*, tech. rep. ATL-DAQ-PROC-2016-022. 12,
789 CERN, 2016, URL: <https://cds.cern.ch/record/2229597> (cit. on p. 3).
- 790 [5] J. Schumacher, ‘Interfacing Detectors and Collecting Data for Large-Scale Experiments in High
791 Energy Physics Using COTS Technology’, Presented 09 Jun 2017, 2017, URL: <https://cds.cern.ch/record/2268506> (cit. on p. 3).
- 793 [6] T. Heim, *YARR - A PCIe based readout concept for current and future ATLAS Pixel modules*, J. Phys.
794 : Conf. Ser. **898** (2017) 032053. 8 p, URL: <https://cds.cern.ch/record/2297459> (cit. on
795 p. 3).
- 796 [7] Xilinx, *Xilinx VC709 Development Kit*, URL: <http://www.xilinx.com/products/boards-and-kits/%20dk-v7-vc709-g.html> (cit. on p. 4).
- 798 [8] ATLAS FELIX Group, *FLX-712 (BNL-711 V2) Manual*, ATLAS-INT-2018-XXX, 2018, URL:
799 https://atlas-project-felix.web.cern.ch/atlas-project-felix/user/docs/BNL-711_V2P0_manual.pdf (cit. on p. 4).
- 801 [9] A. F. Group, *FELIX User Manual*, URL: <https://atlas-project-felix.web.cern.ch/atlas-project-felix/user/felix-user-manual/versions/4.0.6/felix-user-manual.pdf> (cit. on p. 4).
- 804 [10] Xilinx, *Xilinx Vivado Design Suite, 2015*, URL: <https://www.xilinx.com/products/design-tools/%20vivado.html> (cit. on p. 10).
- 806 [11] R. Martin Lesma et al., *The Versatile Link Demo Board (VLDB)*, *JINST* **12** (2017) C02020. 12 p,
807 URL: <https://cds.cern.ch/record/2275133> (cit. on p. 16).

808 **List of contributions**

B. Abbott

J. Lambert

809 F. Metcalfe

A. Paramonov

M. Trovato

810

Not reviewed, for internal circulation only



Figure 43: Interface card used to convert electrical signals from the RD53a to optical and pass data from the VLDB to the RD53A.

811 Appendices

812 A 4-channel Interface Card (superseded)

813 This section describes the setup of the interface card developed at Argonne. The interface card can be seen
 814 in Figure 43. The interface card passes data received from the VLDB to the RD53A and data from the
 815 RD53A to the VC709.

816 **Install QSFP in the interface card QSFP cage.** Make sure that the QSFP is pushed all of the way in or
 817 you will experience issues later. Once the QSFP is installed, insert the multi-strand fiber into the
 818 QSFP.

819 **Connect the interface card to one of the VLDB ports seen in Figure 44.** The VLDB should be connected
 820 to the interface card using a mini-HDMI to HDMI cable. You should note the port that you use on the
 821 VLDB. Data passed to the VLDB from FELIX will be passed to one of these depending upon which port
 822 the data was addressed to. So the number in Figure 44 will be the address of the RD53a chip connected to
 823 the interface card.

824 **Connect the interface card to the RD53A.** The interface card is connected to the RD53A using the
 825 DisplayPort-to-DisplayPort cable. Use the DisplayPort connector on the RD53A closest to the Molex
 826 power connectors. The correct DisplayPort on the RD53A to use can be seen in Figure 25.

827 **Power the interface card using the included power adapter.** If you have an interface card, you should
 828 have a power adaptor that lets you power the interface card directly from a wall outlet.

829 **Connect multi-strand fiber from the interface card to VC709.** The multi-strand fiber has many different
 830 fiber optic pairs. Each of these pairs has a number written on a small, white, plastic piece that goes from 1
 831 to 8. If you are using a 12-strand fiber, these numbers will be different. For the 8-strand fiber, each pair
 832 corresponds to a different channel that can be enabled on the RD53A by setting a register in the RD53A.
 833 **By default, only the one channel is enabled, corresponding to the fiber numbered 8.** In this manual,

Not reviewed, for internal circulation only

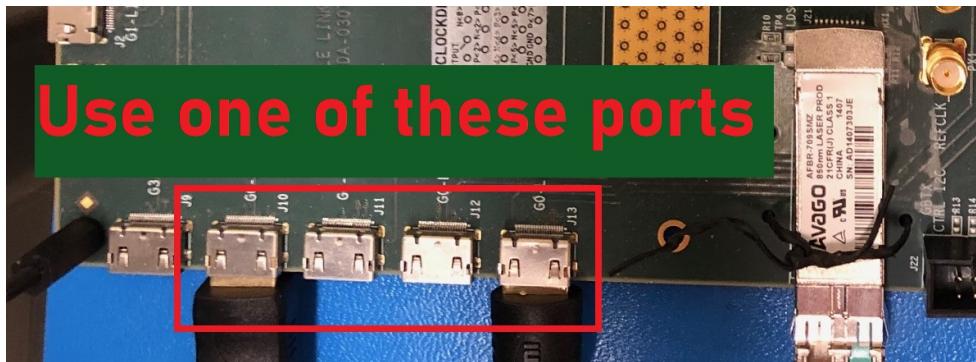


Figure 44: Mini-HDMI ports that can be used to connect to interface cards.



Figure 45: Ports on the VC709 SFPs that are required to be used. This picture was taken looking up towards the ceiling.

834 we assume that only the default channel is enabled so this will be the only fiber needed. **Note: one module**
 835 **must be connected to the SFP labeled 2 in Figure 45** There are four different SFPs on the VC709 so
 836 we can receive data from four different modules. The SFP labeled 2 in Figure 45 acts as the master and
 837 supplies the clock to any other modules so you must connect the fiber from one interface card here. Any
 838 additional modules can use any of the other Rx ports. An example illustrating how to connect fibers into
 839 the VC709 SFPs for the case using two chips can be seen in Figure 46.

840 Additionally, in the configuration described in Sec. ?? you can select the polarity of the data via VIO from
 841 each GBT ID by setting a 4-bit hex in the same way that the active lanes were set. Typically, the polarity
 842 will not need to be changed. In the case that the data appears to be flipped (lots of fffs) on one elink, you
 843 can change the polarity of that elink. This can occur due to a bug in the interface card.

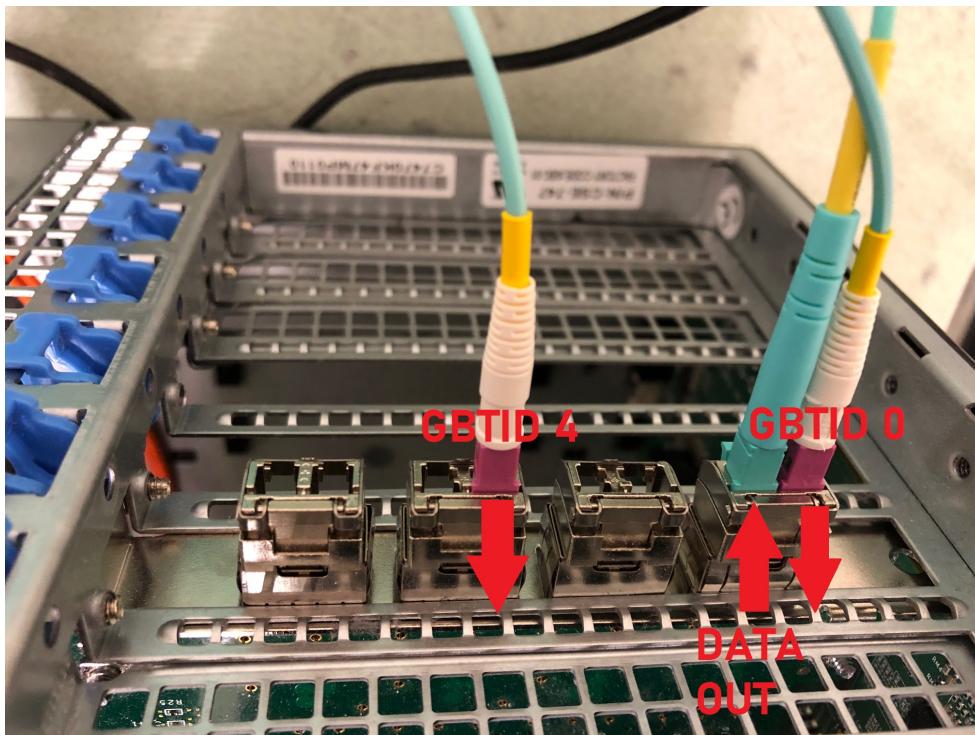


Figure 46: Example illustrating how to connect the fibers from two interface cards to the VC709. In this example, we have connected the fibers for two chips. Note that one fiber is connected to the SFP labeled 2 in Figure 45. This is required. The position to the second fiber is arbitrary. This picture was taken looking over the top of the server case.

844 **Power the RD53A directly from the interface card.** The Rd53A can be connected to the interface card
 845 using a 4-pin molex to 4-pin molex cable to power the Rd53a directly from the interface card. The 4-pin
 846 molex connector should be wired the such that the two wires close to the tab are positive and the two wires
 847 away from the tab are negative. To begin supplying power over the molex cable, press the 'ON' button
 848 located near the QSFP cage. A green LED should light up. If the red 'FAULT' light is on, either press the
 849 'RESET' button or press the 'OFF' button followed by the 'ON' button. The green 'ON' LED should then
 850 be on.

851 **B Hardware Needed**

852 This section briefly describes the components that can be purchased to setup a FELIX server. There are
 853 many solutions to this problem and the reader is encouraged to find a better one.

854 **B.1 Computer Part List**

855 The FELIX servers used at Argonne National Laboratory were built using the following components:
 856 • Intel Xeon E5-1650 v4 Broadwell 3.6 GHz LGA 2011-3 140W BX80660E51650V4 Server Processor
 857 • Supermicro SNK-P0048AP4 CPU Cooling Fan/Heatsink for Socket LGA 2011

- 858 • Rosewill 2.5" SSD to 3.5" SATA Hard Disk Drive HDD Adapter Caddy Tray Cage Hot Swap
859 Plug Converter Bracket Compatible with All 3.5" SATA Drives, Hot-Swap Caddie Tray Cage
860 (RSA-HA001)
- 861 • SUPERMICRO MBD-X10SRA-F-O ATX Server Motherboard LGA 2011-3 Intel C612
- 862 • SUPERMICRO CSE-743TQ-865B Black 4U Pedestal Server Case 865W 2 External 5.25" Drive
863 Bays
- 864 • SAMSUNG 8GB 288-Pin DDR4 SDRAM ECC Registered DDR4 2133 (PC4 17000) Server Memory
865 Model M393A1G40DB0-CPB
- 866 • MZ-7KM1T90 SAMSUNG 1.9TB MLC SATA 2.5" 6Gb/s SSD SM863 SERIES MZ7KM1T9HAJM-
867 00005

868 B.2 Additional Hardware

869 In addition to the FELIX server, there are several other pieces of hardware that were used to configure and
870 readout the RD53a with FELIX:

- 871 • *Direct readout:*
 - 872 – VLDB: CERN is not assembling it any longer, request it to us alexander.paramonov@cern.ch
 - 873 – power connector: <http://www.mouser.com/catalog/645/usd/1739.pdf>.
 - 874 – 12-channel Interface card: request it to us marco.trovato@cern.ch
 - 875 – mini-HDMI to mini-HDMI cables: <https://www.digikey.com/en/products/detail/cnc-tech/742-20010-00100/4340371>
 - 876 – mini-DisplayPort-to-DisplayPort cables: <https://www.digikey.ee/product-detail/en/assmann-wsw-components/AK-340102-010-W/AE11262-ND/7361006>
 - 877 – Fiber optic cable
 - 878 – Light prizm cable: <https://www.digikey.com/en/products/detail/molex/1062673000/3305778>)
 - 879 – MTP coupler: <https://store.cablesplususa.com/mtp-adpt.html>
 - 880 – MTP to 12xLC cable: <https://store.cablesplususa.com/12-fiber-mtp-to-lc-cassette-to-sfp-fiber-optic-cable-singlemode-os2.html>
 - 881 – Power adapter: <https://www.digikey.com/en/products/detail/mean-well-usa-inc/GST25A05-P1J/7703645>
 - 882 – VC709:
 - 883 * QSFP
 - 884 * TTCfx
 - 885 – *FLX712:* request it to the FELIX team (j.panduro.vazquez@cern.ch)

- * MTP48-to-2xMTP12¹⁰ or MTP24-to-2xMTP12 breakout cables <https://store.cablesplususa.com/24-fiber-100g-mtp-to-2x-mtp-cfp-fiber-optic-cable-multimode-om4.html>.
- * MTP to MTP cable: <https://store.cablesplususa.com/mt10muom4fio2.html?Polarity=Type%20B%20-%20Rollover> or
- * MTP to 12xLC cable: <https://store.cablesplususa.com/12-fiber-mtp-to-lc-cassette-to-sfp-fiber-optic-cable-singlemode-os2.html>
- *LpGBT readout:*
 - * FLX712: : request it to the FELIX team (j.panduro.vazquez@cern.ch)
 - * VLDB+: https://espace.cern.ch/GBT-Project/VLDBplus/_layouts/15/start.aspx#/SitePages/Home.aspx
 - * MT to MPO adapter plus guide pins from the MT ferrule KIT: <https://www.mouser.com/ProductDetail/538-106114-1150>, <https://www.mouser.com/ProductDetail/TE-Connectivity-Raychem/2102866-2?qs=%2Fha2pyFadug9mWDn4o%252BsrwvvNDqZ5ZIKwuPr7uC7LjNCYqrTtY090g%3D%3D>
 - * MTP coupler: <https://store.cablesplususa.com/mtp-adpt.html>
 - * MTP12 to MTP12 cable: : <https://store.cablesplususa.com/mt10muom4fio2.html?Polarity=Type%20B%20-%20Rollover>; or
 - * 2x MTP12 to 12xLC: <https://store.cablesplususa.com/12-fiber-mtp-to-lc-cassette-to-sfp-fiber-optic-cable-singlemode-os2.html> plus patch panel
 - * external power 10V supply

908 B.3 VLDB Configuration

```

909 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 7f 28 00 15 15 15 66 00
910 0d 42 00 0f 04 08 00 20 00 00 00 00 15 15 15 00 07 00 3f 00 00 00 00 00 00 00 00 2a 2a dd 0d 00 00 00
911 00 00 00 00 00 00 00 00 00 00 00 55 55 55 00 00 00 2a dd 0d 00 00 00 00 00 00 00 00 00 00 00 00 00 00
912 00 00 55 55 55 00 00 00 2a dd 0d 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 55 55 55 00 00 00 2a dd
913 0d 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
914 00 00 00 00 00 55 55 55 00 00 00 00 00 70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
915 00 00 00 00 70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
916 00 00 00 3f 3f 38 00 00 00 00 00 00 00 07 00 00 0a ff 55 02 ff 55 02 ff 55 02 ff 55 00 00 00 00 00 00
917 00 00 00 00 00 15 00 00 00 00 00 00 00 00 00 00 00 ff ff 40 40 40 2a 2a 2a 00 00 ff ff 40 40 40 2a
918 2a 2a 4e 4e aa 0a 07 00 ff ff ff 00 00 00 00 00 00 00 0a ff 55 02 ff 55 02 ff 55 02 ff 55 0a ff
919 55 02 ff 55 02 ff 55 02 ff 55 00 00 00 aa 00 02 00

```

¹⁰ Contact sales@cablesplususa.com for the cable fabrication