

Table of Contents

WIB Basic Operation.....	1
WIB Overview.....	1
Power Off FEMBs.....	1
Configure & Check Status.....	1
WIB Hardware Status.....	2
WIB Hardware in Vertical Slice Test.....	2
WIB Hardware on ProtoDUNE.....	2
Firmware.....	2
Instructions for installing Firmware.....	2
Remote Firmware Update.....	2
Using USB Blaster to program VST WIBs.....	4
Instructions for using WIB with FELIX Firmware.....	4
Instructions for using WIB with RCE Firmware.....	4
General Firmware Instructions.....	4

WIB Basic Operation

WIB Overview

The WIB communicates with the FEMBs and the RCE/FELIX. The only thing the WIB does from the standpoint of artDAQ is configure itself and the FEMBs.

WIB expert understanding of steps that must happen to get into normal data-taking state:

1. Power on WIB (if necessary)
2. Configure WIB
3. Reset timing system (or equivalent that makes WIB endpoint go from W_RDY 0x6 to RUN 0x8) This should also be able to happen before the WIB is configured.
4. RCEs and FELIX start recording data

Nothing needs to happen to the WIB at the end of a run, you just start the steps over when you want to start a new run.

Power Off FEMBs

Login to np04-srv-012 as np04daq. From a control room computer the srv012 command does this.

Setup the BUTool:

```
cd /nfs/sw/wib/WIBSoftwareTrunk
source env.sh
```

```
cd scripts
BUTool.exe -X power_off_ALL_FEMBs.script
# If FELIX readout is in progress leave APA6 on by turning off APA1-5 FEMBs individually
BUTool.exe -X power_off_FEMB_APA1.script
BUTool.exe -X power_off_FEMB_APA2.script
BUTool.exe -X power_off_FEMB_APA3.script
BUTool.exe -X power_off_FEMB_APA4.script
BUTool.exe -X power_off_FEMB_APA5.script
```

Press ctrl-d to exit the BUTool command line interface.

Verify the FEMBs are off with the femb-status.sh command, written by Karol.

Configure & Check Status

Login to np04-srv-012 as np04daq.

Setup the BUTool:

```
cd /nfs/sw/wib/WIBSoftwareTrunk
source env.sh
```

cd into the scripts directory and run one of the below commands to setup the WIB:

```
BUTool.exe -X config_WIB_VST_RCE_default.script # configure VST RCE WIB and FEMBs with default re
BUTool.exe -X config_WIB_VST_FELIX_fake_WIB_data.script # configure VST FELIX WIB to produce fake
```

```
BUTool.exe -X config_WIB_ColdBox_RCE_default.script # configure coldbox WIBs and FEMBs with default
BUTool.exe -X config_WIB_ColdBox_RCE_fake_WIB_data.script # configure coldbox WIBs to produce fake
# and similar for the other configs, see README in scripts directory for more info about configs
```

Press ctrl-d to exit the BUTool command line interface.

To dump the status of the WIBs, run one of the dump*.sh scripts in the directory piping the output somewhere (there is a lot of output). The first lines are the WIB config.

WIB Hardware Status

#HardwareStatus

WIB MAC and IP Address scheme [↗](#)

WIB Hardware in Vertical Slice Test

- WIB v2 in slot 2 of the PTC, 192.168.200.2, hooked up to RCE, flash firmware changing often while we test remote programming
 - ◆ Port 1 hooked up to ProtoDUNE FEMB, firmware 0x323
 - ◆ Port 2 empty
 - ◆ Port 3 empty
 - ◆ Port 4 empty
- WIB v2 in slot 5 of the PTC, 192.168.200.5, hooked up to FELIX, flash firmware: BU version 0x18072501 RCE
 - ◆ Port 1 empty
 - ◆ Port 2 empty
 - ◆ Port 3 hooked up to ProtoDUNE FEMB box 20, firmware 0x323
 - ◆ Port 4 hooked up to ProtoDUNE FEMB box 9, firmware 0x323 - Toy TPC attached
- PTC card, right now PTC cable going into here and fanned out to WIBs on the backplane

WIB Hardware on ProtoDUNE

- APA 1-6 have FEMBs with firmware version 0x323
- Cold box WIB crate has been taken off the cold box

Firmware

You can check the firmware version and if it is for the RCE or FELIX by logging in to np04-srv-012 and running:

```
cd /nfs/sw/wib/WIBSoftwareTrunk
source env.sh
BUTool.exe -w 192.168.200.5 # or 192.168.200.2 for FELIX WIB or np04-wib-101, etc.
status 1 fw
```

Instructions for installing Firmware

Remote Firmware Update

First, login to np04-srv-012.

WIBBasicOp < CENF < TWiki

To program the flash with the BU firmware on the FPGA, download the firmware .hexout file to that computer (you may have to source `~np04daq/bin/web_proxy.sh`).

Next,

```
cd /nfs/sw/wib/WIBSoftwareTrunk
source env.sh
```

And for the WIB you want to update (np04-wib-Y0X where Y is crate number and X is WIB number in the crate):

```
BUTool -w np04-wib-Y0X
```

then check the current firmware on with the `status 1 fw` command. The BU version should be equal to or greater than 0x18071902 to support remote firmware update. You then run `flash_wr <path to .hexout firmware file>` to update the firmware, and type Yes. It should take about 30 minutes, and you can update multiple WIBs in parallel. Example commands:

```
flash_wr /nfs/sw/fpga/firmwares/protoDUNEWIB_RCE_18071902.hexout
flash_wr /nfs/sw/fpga/firmwares/protoDUNEWIB_FELIX_18071902.hexout
```

This reprograms the flash memory but not the FPGA. Then you can use the flash reconfig command in the BU firmware to load the firmware from the flash to the FPGA. Before you do this, check that the `RECONFIG_BUSY` is 0x0:

```
>read FLASH.RECONFIG_BUSY
FLASH.RECONFIG_BUSY: 0x00000000
>write FLASH.RECONFIG 1
FLASH.RECONFIG: 0x00000001
```

Then you've loaded the new programming to the FPGA. To program all WIBs on an APA at once go to `/nfs/sw/wib/WIBSoftware/scripts` and use:

```
./wib_program_remote_apa [apa number] [programming file]
```

Wait for all 5 BUTool subprocesses to be done (usually takes around 30 minutes), then run:

```
./wib_reload_fw_from_flash_apa [apa number]
```

That should take just a few minutes and then all FPGAs on the specified WIB should have the new firmware loaded.

To program the flash with the BNL firmware on the FPGA, first download the rpd file you need.

Next,

```
cd /nfs/sw/wib/FW_PRGM/src
```

Then check the current FPGA firmware on with the `fw_prgm` tool, using the WIB IP address you want. The BNL version should be equal to or greater than V117 to support remote firmware update.

```
./fw_prgm -w any-text-here 10.73.137.29
```

Then you can use the same tool to program the flash to the BU firmware (you will need the rpd file for the BU version you want):

```
./fw_prgm -ep /nfs/sw/wib/firmware/protoDUNEWIB_RCE_18080101.rpd 10.73.137.29
```

Don't worry about the "write again" messages and it should take around 5 minutes. Then you must power cycle the WIB to load the new firmware from the flash to the FPGA. Then you can use the BUTool to recheck the BU firmware is correctly loaded to the FPGA. To program with BNL firmware to all WIBs on an APA go to /nfs/sw/wib/WIBSoftware/scripts, and use:

```
./wib_bnl_fw_program [apa number] [programming file]
```

Then you will need to power cycle that APA's WIB crate to load the firmware to all the WIB FPGAs.

Using USB Blaster to program VST WIBs

SSH to np04-onl-002

```
cd /nfs/sw/wib/WIBSoftwareTrunk/scripts
```

For the RCE VST WIB 2:

```
./wib_program -c "USB-Blaster [3-2]" <path to .sof or .jic programming file>
```

e.g.

```
./wib_program -c "USB-Blaster [3-2]" /nfs/sw/fpga/firmwares/protoDUNEWIB_RCE_18041701.sof
```

For the FELIX WIB 5:

```
./wib_program -c"USB-Blaster [3-1]" <path to .sof or .jic programming file>
```

e.g.

```
./wib_program -c "USB-Blaster [3-1]" /nfs/sw/fpga/firmwares/protoDUNEWIB_FELIX_18060501.sof
```

Instructions for using WIB with FELIX Firmware

SSH to np04-srv-012

```
cd /nfs/sw/wib/WIBSoftwareTrunk/  
source env.sh  
cd scripts  
BUTool.exe -X config_WIB_VST_FELIX_fake_WIB_data.script
```

The board should be setup properly now for the FELIX DAQ and timing system.

Instructions for using WIB with RCE Firmware

SSH to np04-srv-012

```
cd /nfs/sw/wib/WIBSoftwareTrunk/  
source env.sh  
cd scripts  
BUTool.exe -X config_WIB_VST_RCE_default.script
```

The board should be setup properly now for the RCE DAQ and timing system

General Firmware Instructions

Dan will post new files at: http://gauss.bu.edu/redmine/projects/dune-wib/wiki/ProtoDUNE_WIB_Firmware

The .hexout files are used to update the WIB flash over ethernet using the BUTool. A powercycle is required to actually use the firmware loaded this way. The below firmwares are for uploading using a USB blaster connected to a WIB.

sof files are for programming just the FPGA with firmware using the P09 header on the board or front panel--it will disappear after a power cycle. jic files are for programming the flash memory from header P09 or the front panel--this needs a power cycle to be loaded into the FPGA, but will persist after power cycles. pof files are for programming the flash from the P10 header on the board--this needs a power cycle to be loaded into the FPGA, but will persist after power cycles. When plugging the USB-Blaster cable into the header, take care that the end of the connector connected to the red wire is connected to pin 1 on the board.

When loading the WIB firmware using the front panel dongle, sof files can be used to program the FPGA and jic files can be used to program the flash.

To see if the programmer can talk to the board, on np04-onl-002, run:

```
/nfs/sw/fpga/intelFPGA/16.1/qprogrammer/bin/jtagconfig
```

The output should be something like:

```
[juhugon@np04-onl-002 ~]$ /nfs/sw/fpga/intelFPGA/16.1/qprogrammer/bin/jtagconfig
1) USB-Blaster [1-1.5]
   02A030DD 5AGT (FD3H3|MD3G3)/5AGXBB3D4/..

2) USB-Blaster [2-1.2.2]
   02A030DD 5AGT (FD3H3|MD3G3)/5AGXBB3D4/..
```

If you get error code 89 the udev rules are not in place - add the file /etc/udev/rules.d/40-usbblaster.rules

```
SUBSYSTEM=="usb", ATTRS{idVendor}=="09fb", ATTRS{idProduct}=="6001", GROUP="plugdev",
MODE="0666", SYMLINK+="usbblaster"
```

If you get "jtag chain broken", then make sure the WIB is powered on and that the USB-Blaster cable is plugged in the correct direction.

If you continually get permission denied or jtag chain broken errors, then you should try to restart the jtagd process:

```
sudo killall jtagd
sudo /nfs/sw/fpga/intelFPGA/16.1/qprogrammer/bin/jtagd
```

Matthew Worcester - 2018-08-09

This topic: CENF > WIBBasicOp

Topic revision: r63 - 2018-09-21 - Matthew Worcester



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback