Table of Contents

Timing systems advanced operations **1**
- On-Call Schedule 1
- Repo links 1
- CERN System setup 1
  - Hardware 1
  - Connections 1
- Essential recovery guide 1
  - Cold reset 2
  - Trigger-Timing link recovery 2
  - CRT timing endpoint recovery 3
- Timing system advanced user guide 3
  - Firmware 3
  - Software 4
  - Version Compatibility Warnings 4
  - Upload timing firmware at np04 4
    - Configure the physical layer 5
    - Reset and configure the clock infrastructure 5
  - Installation and build instructions for standalone systems 5
    - Configure and start the timing master 6
    - Periodic and poisson triggers 6
  - Reference info 6
- Fanout0 connections 7
- Connections to TLU from SPS timing signals 7
- White Rabbit 7
- Timing system on call Instructions 8
  - On call duties 8
    - When you encounter a problem 8
  - Instructions for on call duties 9
    - Basic overview of how PDTS works 9
    - Accessing the np04.timing@cernNOSPAMPLEASE.ch Skype for Business account 9
    - How to perform regular status checks 9
    - Meetings you need to attend (times+locations) 9
    - How to get basic debug info from the system 9
    - How to reload firmware 9

Timing systems advanced operations

On-Call Schedule

Link to on-call schedule: https://teamup.com/ks9fj1qdrba84uozmd On-call phone number: +41227665291

Repo links

Link to firmware repo: https://gitlab.cern.ch/protoDUNE-SP-DAQ/timing-board-firmware

Link to the software repo: https://gitlab.cern.ch/protoDUNE-SP-DAQ/timing-board-software

CERN System setup

The system is meant to be operated from the control pc `np04-srv-012` with the exceptions of operations requiring a JTAG connection to the FPGA (i.e. firmware uploads).

Hardware

The JTAG identifiers for the systems at EHN1 are as follows:

Board	IPBus ID	IP address	JTAG pc	JTAG Cable Shortcut	Full JTAG Cable ID
TLU	PROD_MASTER	192.168.200.64	np04-onl-001	53FA	Digilent/25163300153FA
Fanout 0	PROD_FANOUT_0	192.168.200.65	np04-onl-001	4E7A	Digilent/2516330014E7A
Fanout 1	PROD_FANOUT_1	192.168.200.66	np04-onl-001	4D9A	Digilent/2516330014D9A
Fanout 2	PROD_FANOUT_2	192.168.200.67	np04-onl-001	4E4A	Digilent/2516330014E4A
TERTIARY	PDTS_TERTIARY	192.168.200.32	np04-onl-002	F4CB	Digilent/210299A1F4CB
PRIMARY	PDTS_PRIMARY / CRT_ENDPOINT	192.168.200.16	np04-crt-001	c301	XiLinx/00001876c3c301
SECONDARY	PDTS_SECONDARY	192.168.200.17	np04-onl-001	4001	XiLinx/000018af5f4001

Connections

Board	Usage/Connection	Firmware Image	
TLU	Cryostat and COBs 2-8	v5b0-1/overlord_tlu_relval-v5b0-1.tgz	c
Fanout 0	Cryostat and COBs 2-8	v5b0-1/fanout_0_pc059_relval-v5b0-1.tgz	c
Fanout 1	-	v5b0-1/fanout_1_pc059_relval-v5b0-1.tgz	c
Fanout 2	Timing tests	v5b0-1/fanout_2_pc059_relval-v5b0-1.tgz	c
TERTIARY	VST and COB1	v5b0-1/ouroboros_pc059_relval-v5b0-1.tgz	c
CRT	CRT endpoint	/nfs/sw/timing/milestones/v4/firmware/v4b4/crt_fmc_180905_1239.tgz	c

Essential recovery guide

Updating/reloading timing firmware is an expert operation. Please contact a timing expert before attempting a system recovery.

PROM (MCS) files in np04-srv-007:/sw/timing/dev/firmware/promfiles

MCS files have been given the same name as the *.tgz archive containing the *.bit file they were generated from

Cold reset

Reload firmware

```
ssh np04-onl-001
source /nfs/sw/timing/tools/ipbb_env.sh
cd /nfs/sw/timing/dev/firmware/v5b0-1
ipb-prog vivado program 53FA overlord_tlu_relval-v5b0-1.tgz
ipb-prog vivado program 4E7A fanout_0_pc059_relval-v5b0-1.tgz
ipb-prog vivado program 4D9A fanout_1_pc059_relval-v5b0-1.tgz
ipb-prog vivado program 4E4A fanout_2_pc059_relval-v5b0-1.tgz
exit
```

Configure hardware layer

```
ssh np04-srv-012
source /nfs/sw/timing/env_dev.sh

# Hard resets
pdtbutler io PROD_MASTER reset
pdtbutler io PROD_FANOUT_0 reset
pdtbutler io PROD_FANOUT_1 reset
pdtbutler io PROD_FANOUT_2 reset

pdtbutler mst PROD_MASTER synctime
pdtbutler mst PROD_MASTER spillenable

# Trigger interface configuration
pdtbutler mst PROD_MASTER ext-trig ept reset
pdtbutler mst PROD_MASTER ext-trig enable --on

# Trigger interface monitor
pdtbutler mst PROD_MASTER ext-trig status
```

Trigger-Timing link recovery

```
# Trigger interface configuration
pdtbutler mst PROD_MASTER ext-trig ept reset

# Trigger interface monitor
pdtbutler mst PROD_MASTER ext-trig status
```

Command output

Show... Hide...

```
Control registers
+-------------+-----+
| ep_en       | 0x1 |
| ext_trig_en | 0x1 |
+-------------+-----+

Status registers
+---------+-----+
| ep_edge | 0x0 |
| ep_fdel | 0x2 |
| ep_rdy  | 0x1 |
```

```
| ep_stat | 0x8 |
+---------+-----+

...
```

CRT timing endpoint recovery

Set up the working environment

```
ssh np04-crt-001
cd /nfs/sw/crt/timing-board-software
source tests/env.sh

pdtbutler io CRT_EPT reset --force-pll-cfg $PDT_TESTS/etc/clock/devel/Si5344-PDTSCRT1NoZdm-RevD-4
```

Check the endpoint status : `ep_rdy=1`, `ep_stat=0x8`

```
pdtbutler crt CRT_EPT status
```

Command output

Show... Hide...

```
Created crt device
+------------------+------------+
| csr              | 0x0        |
| csr.ctrl         | 0x0        |
| csr.ctrl.tgrp    | 0x0        |
| csr.stat         | 0x81       |
| csr.stat.ep_rdy  | 0x1        |
| csr.stat.ep_stat | 0x8        |
| pulse            | 0x0        |
| pulse.cnt        | 0x11       |
| pulse.ctrl       | 0x0        |
| pulse.ctrl.cmd   | 0x0        |
| pulse.ctrl.en    | 0x0        |
| pulse.ctrl.force | 0x0        |
| pulse.ts_h       | 0x116f765  |
| pulse.ts_l       | 0x90232afa |
+------------------+------------+
```

Configure the endpoint to emit a sync pulse at Run Start

```
pdtbutler crt CRT_EPT configure 0 RunStart
```

Timing system advanced user guide

Firmware

- Reference project/tag: `ouroboros` (master + internal endpoint)/ `v4b1`
- Firmware repository: `https://gitlab.cern.ch/protoDUNE-SP-DAQ/timing-board-firmware`
- In-situ firmware image repository: `/nfs/sw/timing/dev/firmware`
- Current firmware images: Check the Connections section.
- All firmware images built by the CI system: `http://pdts-fw.web.cern.ch/pdts-fw/`

Software

- Software repository: `https://gitlab.cern.ch/protoDUNE-SP-DAQ/timing-board-software`
- Reference installation: `/nfs/sw/timing/pro/software/timing-board-software`
- Local address table repository: `/nfs/sw/timing/pro/addrtab/[tag]`

Version Compatibility Warnings

Please ensure you are running Python 2.7 with the below libraries installed, and that you have boost 1.53 (exactly). Both of them as standard packages available to Centos 7 installation. Additionally, the `uhal` package is required to communicate with the timing firmware. and uhal C++ library version 2.6 (exactly) should be installed. As a list:

Requirements:

- `uhal` C++ library, version 2.6
- python 2.7 with `Click` and `Click_didyoumean` packages
- boost 1.53

All can be checked when you `source env_dev.sh`:

```
[jogreer@np04-srv-012 timing]$ source /nfs/sw/timing/env_dev.sh
uhal is installed
click is installed
click_didyoumean is installed
- LD_LIBRARY_PATH += /opt/cactus/lib
- PATH += /nfs/sw/timing/milestones/v4/software/v4b1-rc1/timing-board-software/core/bin
- PATH += /nfs/sw/timing/milestones/v4/software/v4b1-rc1/timing-board-software/tests/bin
- PATH += /nfs/sw/timing/milestones/v4/software/v4b1-rc1/timing-board-software/tests/scripts
- LD_LIBRARY_PATH += /nfs/sw/timing/milestones/v4/software/v4b1-rc1/timing-board-software/core/li
- PYTHONPATH += /nfs/sw/timing/milestones/v4/software/v4b1-rc1/timing-board-software/python/pkg
- PYTHONPATH += /nfs/sw/timing/milestones/v4/software/v4b1-rc1/timing-board-software/tests/python
```

Upload timing firmware at np04

Updating/reloading timing firmware is an expert operation. Please contact a timing expert before doing it.

Connect to the appropriate JTAG host (see Hardware).

Source ipbb environment

```
source /nfs/sw/timing/tools/ipbb_env.sh
```

Once the ipbb environment is loaded

```
ipb-prog vivado program <jtag cable shortcut>:xc7a35t_0 <bitfile path>
```

The cable names and bitfiles paths are listed the Hardware and Firmware sections.

Example:

```
ipb-prog vivado program c301:xc7a35t_0 /nfs/sw/timing/pro/firmware/ouroboros_v3c/top.bit
```

Configure the physical layer

This step is only required after loading the FPGA with new firmware. Login to `np04-srv-012` to implement pdtbutler commands. The reset command resets all ipbus registers in the master, loads the clock synthesizer chip configuration and starts the sync command generator.

```
pdtbutler io PDTS_TERTIARY reset
```

Reset and configure the clock infrastructure

Resets all master's ipbus registers, loads the SI clock chip configuration and starts the sync command generator. It takes option: `--soft`, which is a speedier reset that skips the clock chip configuration.

```
pdtbutler io PDTS_TERTIARY reset
```

Command output

Show... Hide...

```
Created device PDTS_TERTIARY
Master FW version: 0x40004
Resetting PDTS_TERTIARY
UID I2C Slaves
  FMC_UID_PROM: 0x53
  AX3_Switch: 0x21
  KC705_Switch: 0x74
I2C enable lines: 127
Timing Board UID: 0xd880395e5069
PLL version : 0x5344
Clock configuration to load SI5344/PDTS0000.txt
2018-03-12 10:28:17.239 pdt INFO     | Configuration read from file (445 entries). Starting uploa
Freq: 0 1 249.994377134
Freq: 1 1 249.994377134

--- Global status ---
tx_err: 0x0

Disabled command generator
Time-sync generator enabled
```

Help

Show... Hide...

```
   Usage: pdtbutler io reset [OPTIONS]

  Perform a hard reset on the timing master, including

  - ipbus registers
  - i2c buses
  - pll and pll configuration

Options:
  -s, --soft               Soft reset i.e. skip the clock chip configuration.
  -h, --help               Show this message and exit.
```

Installation and build instructions for standalone systems

Please ensure you are running Python 2.7, and that you have boost 1.53 (exactly) installed. Both of them as standard packages available to Centos 7 installation. Additionally, the `uhal` package is required to

communicate with the timing firmware. The installation instructions of the uhal package are available at the following address

https://ipbus.web.cern.ch/ipbus/doc/user/html/software/installation.html

Download the `v4a3` software tag

```
git clone -b v4a3 --depth 1 https://:@gitlab.cern.ch:8443/protoDUNE-SP-DAQ/timing-board-software.
```

Compile the C++ layer

```
cd timing-board-software
make
```

Source the environment

```
source tests/env.sh
```

Ready to go!

Configure and start the timing master

The typical configuration & start sequence for stand-alone test setups is

```
pdtbutler io PDTS_TERTIARY reset --soft
pdtbutler mst PDTS_TERTIARY faketrig-conf 0 12
pdtbutler mst PDTS_TERTIARY part 0 configure
pdtbutler mst PDTS_TERTIARY part 0 start
pdtbutler mst PDTS_TERTIARY part 0 trig
pdtbutler mst PDTS_TERTIARY part 0 status
```

In this example the fake trigger rate is set to 12 Hz (`faketrig-conf 12`) and only partition 0 is enabled. Note that triggers need to be explicitly enabled for the chosen partition (`part 0 trig`) after start. Please note that the trigger command generator is a global entity. Trigger rate changes will affect all partitions.

If required, the fake trigger generator is disabled by running the command

```
pdtbutler mst PDTS_TERTIARY faketrig-clear 0
```

Periodic and poisson triggers

The fake triggers are typically set to periodic. To set them to a poisson (random) distribution use the --poisson flag

```
pdtbutler mst PDTS_TERTIARY faketrig-conf 0 12
```

Reference info

The status register for the board has the following definition:

State	Description
`0x0`	Standing by
`0x1`	Waiting SFP for signal
`0x2`	Waiting CDR lock
`0x3`	Waiting for good frequency check
`0x4`	Waiting for comma alignment

0x5	Waiting for 8b10 decoder good packet
0x6	Waiting for time stamp initialisation
0x8	Ready
0xc	Error in Rx
0xd	Error in time stamp check
0xe	Error in physical layer after lock

The 'sync' bus has the following commands at the moment:

TimeSync	0x0
Echo	0x1
SpillStart	0x2
SpillStop	0x3
RunStart	0x4
RunStop	0x5
WibCalib	0x6
SSPCalib	0x7
FakeTrig0	0x8
FakeTrig1	0x9
FakeTrig2	0xa
FakeTrig3	0xb
BeamTrig	0xc
NoBeamTrig	0xd
ExtFakeTrig	0xe

Fanout0 connections

Slot	APA
0	CRT
1	US-RaS (3)
2	DS-DaS (4)
3	MS-DaS (6)
4	US-DaS (5)
5	DS-RaS (1)
6	MS-RaS (2)
7	COBs

Connections to TLU from SPS timing signals.

Signal	Label in TLU	Signal
WWE	6 (right hand side)	5
WE	5	4
EE	4	3

All signals NIM. Should be 2us wide. For definitions see, e.g.,
http://sba.web.cern.ch/sba/BeamsAndAreas/H2/H2_faq.html#What_is_the_timing_signal_that_I_receive_from_the_SI

White Rabbit

We have a WR-LEN module (http://sevensols.com/index.php/products/wr-len/) that provides a 10MHz clock (and IRIG-B time-code that we don't use at the moment) . This clock is connected to the TLU via a Balun (SMA at WR-LEN, 2-pole Lemo-00 at TLU).

WR-LEN is monitored over USB Serial link (currently connected to np04-onl-001). Can use e.g. minicom as a terminal (e.g. , sudo minicom -D /dev/ttyUSB0 -b 115200)

Use the "gui" command at the "wrc#" prompt to display status. e.g.

```
WR PTP Core Sync Monitor: PPSI - LEN board
Esc = exit

TAI Time:                  Tue, Sep 4, 2018, 14:06:08

WR-LEN mode : WRC_SLAVE_WR0
-----------------------
Link status:

 wr0 : Link up   (RX: 334697, TX: 102136), mode: WR Slave   Locked  Calibrated
 IPv4: BOOTP running

 wr1 : Link down
-----------------------

Servo state:               TRACK_PHASE
Phase tracking:            ON
Synchronization source:    wr0

Timing parameters:

Round-trip time (mu):    1188624 ps
Master-slave delay:      598623 ps
Master PHY delays:       TX: 224603 ps, RX: 235091 ps
Slave PHY delays:        TX: 205019 ps, RX: 224093 ps
Total link asymmetry:        -8622 ps
Cable rtt delay:         299818 ps
Clock offset:                -6 ps
Phase setpoint:             4926 ps
Skew:                          1 ps
Manual phase adjustment:       0 ps
Update counter:              89112
--
```

... I think the things we are looking for are " WR Slave Locked Calibrated" and "Servo state: TRACK_PHASE"

Timing system on call Instructions

On call duties

What tasks are the on-call person expected to do? These bullets correspond to sections below.

- Make sure you can access the np04.timing@cernNOSPAMPLEASE.ch Skype for Business account
- Read the on-call section of this page at the start of your duties
- Perform regular status checks
- Respond to phone calls to the np04.timing@cernNOSPAMPLEASE.ch Skype for Business account
- Attend meetings
- Report back to PDTS group at the end of the week
- Hand over to the next on-call person

When you encounter a problem

- Get basic debug info
- Check known issues

- Know which expert to call

Problems that on-call person can fix without calling an expert

- ...

Known issues are:

- ...

Who to call for known issues:

- ...

Contact details for experts:

- ...

Instructions for on call duties

Basic overview of how PDTS works

...

Accessing the np04.timing@cernNOSPAMPLEASE.ch Skype for Business account

- Download the Skype for Business app to your phone
- Log in with email address: np04.timing@cernNOSPAMPLEASE.ch
- Get the password from previous on-call person
- The phone number is:+41227665291
- There is an issue which may cause problems when downloading Skype for Business to macbook laptops from CERN MDM mac self-service (http://information-technology.web.cern.ch/services/fe/mac-support/howto/mac-self-service)

How to perform regular status checks

Meetings you need to attend (times+locations)

How to get basic debug info from the system

How to reload firmware

Instructions above in the Essential recovery guide (for full recovery of timing system) or Upload timing firmware at np04 sections (upload individual firmware files).

This topic: CENF > TimingSystemAdvancedOp
Topic revision: r35 - 2020-11-16 - DavidCussans