

The ST TELL1 Emulator - OLD

Warning: This page describes an old version of the ST TELL1 Emulator. For the most recent version go here

This page contains information about the ST TELL1 Emulator and how to use it. This information is valid for **ST/STTELL1Algorithms v2r0**.

The packages used by the Emulator are:

- **ST/STTELL1Algorithms**
- **Velo/TELL1Engine**
- **DAQ/Tell1Kernel**

STTELL1Algorithms contains the following 3 algorithms:

- Pedestal Subtraction (Including Pedestal Update and Header Correction)
- Common Mode Subtraction
- Zero Suppression

These algorithms function as a *wrapping* around the C-code written processes (supplied by Guido), which can be found in the TELL1Engine package. STTELL1Algorithms is the only package which is specific to the ST. Both TELL1Engine and Tell1Kernel are also used by the Velo.

Before running the Emulator the raw data must be decoded. This is done using the STFullDecoding algorithm (ST/STOnline). To run the Emulator add the following lines to the options file:

```
ApplicationMgr.TopAlg = {"STFullDecoding", "ProcessPhase/Emulator" };

Emulator.DetectorList += { "ST" };
EmulatorSTSeq.Members += {
    "STTELL1PedestalSubtractor"
    , "STTELL1LCMS"
    , "STTELL1ClusterMaker"
};
```

Another important algorithm is "RawBanktoSTClusterAlg" (ST/STDAQ). With this algorithm clusters are decoded. To compare TELL1 clusters with Emulator clusters, this algorithm needs to be run twice. Once to decode the clusters in the ZS bank and once to decode the clusters created by the ClusterMaker of the Emulator. For the second, the rawEventLocation needs to be set to "Emu/RawEvent" and as output location you can set the clusterLocation to "Emu/IT/Clusters".

Shared Job Options

The Emulator can either be run using the **Conditions Database** or using **Configuration Parameters specified in the options file**. The job options (and their default values) which both methods share are:

STTELL1PedestalSubtractor	STTELL1LCMS	STTELL1ClusterM
InputDataLoc (STTELL1DataLocation::TTFull)	InputDataLoc (STTELL1DataLocation::TTPedSubADCs)	InputDataLoc (STTELL1DataLocation::T
OutputDataLoc (STTELL1DataLocation::TTPedSubADCs)	OutputDataLoc (STTELL1DataLocation::TTLCMSADCs)	
SubPedsLoc (STTELL1DataLocation::TTSubPeds)	ReadoutTool ("TTReadoutTool")	ReadoutTool ("TTReadout
ReadoutTool ("TTReadoutTool")	DetType ("TT")	DetType ("TT")
		CondDB (true)

DetType ("TT")	CondDB (true)	ConvergenceLimit (0)
CondDB (true)	ConvergenceLimit (0)	

The ConvergenceLimit is the number of events used to calculate a good pedestal. For these events only the Pedestal Subtraction algorithm will run. Only when the ConvergenceLimit is reached will the CMS and the ClusterMaker run. (Example: If a strip is off, i.e. it always has ADC 0 and we start with subtracting the default value of 128 (see next section), then it takes the Pedestal Update about 12000 events to get the pedestal to the right value of 0.)

Configuration Parameters

If you set `CondDB = false` the configuration parameters need to be specified in the options file. To all options which are a vector either **1 init value** can be assigned, or the **full size** of the vector. The following parameters (with default values and size of vector) can be set.

STELL1PedestalSubtractor
HeaderCorrectionInit (0), size:192
HeaderThresholdInit (100), size:2
LinkMaskInit (0), size:3072
LinkPedestalInit (128), size:3072
PedestalEnable (1)
UpdateEnable (1)
ZSEnable (1)
HeaderCorrectionEnable (1)

The LinkMask is a vector of size 3072 (its name may be deceiving), which entries can be set to either 0 or 1. 1 meaning that a certain strip is masked for pedestal subtraction and header correction. The LinkPedestal contains the initial pedestals to be subtracted per link. To enable the pedestal subtraction, both PedestalEnable and ZSEnable must be set to 1. Pedestal Following is enabled by setting UpdateEnable to 1. There are two types of header, a "negative" header (lower than the pedestal line) and a "positive" header (higher than the pedestal line). Cross talk between the last header and first strip may occur (no other order corrections are implemented) and therefore we can set 2 corrections (for "negative" and "positive" headers) per link (32 channels). To enable the header correction set HeaderCorrectionEnable to 1.

STELL1LCMS
CMSThresholdInit (12), size:3072
CMSEnable (1)

To enable the CMS process set CMSEnable to 1. If the CMS is disabled, all data is directly copied and supplied to the ClusterMaker algorithm. The CMS Thresholds per strip are set in the CMSThreshold vector.

STELL1ClusterMaker
HitThresholdInit (9), size:3072
ConfirmationThresholdInit (12), size:48
SpilloverThresholdInit (40), size: 48
PPMaxClusters (255)
ZSEnable (1)

To enable the ZS process set ZSEnable to 1. There are three thresholds used in the ZS process: The HitThreshold is per individual strip and thus the charge of each strip in a cluster needs to exceed this threshold. The total charge of the cluster needs to exceed the ConfirmationThreshold. When the total charge of the clusters is higher than the SpilloverThreshold the Spillover bit is set. Both ConfirmationThreshold as SpilloverThreshold can be set per processing channel (64 strips), but clusters are searched for per beetle (128 strips).

The disadvantage of setting these parameters in the job options file is that 1) they cannot be updated while the program is running and 2) the parameters are the same for all TELL1s.

Using the Conditions Database

When using the conditions database (`CondDB = true`), the location of the database needs to be specified in all three algorithms.

STELL1PedestalSubtractor	STELL1LCMS	STELL1ClusterMaker
CondPath ("/dd/Conditions/TELL1Conf/")	CondPath ("/dd/Conditions/TELL1Conf/")	CondPath ("/dd/Conditions/TELL1Conf/")

No other options need to be set. The parameters you want to use need to be specified in the conditions database. To set up your own local conditions database, follow the instructions on the conditions database 'how to' page, or try the following (until the TELL1 configuration parameters have been added to the conditions database):

First get the database:

- `setenvLHCb v23r2`
- `SetupProject LHCb v23r2`
- `dump_db_to_files.py -c 'sqlite_file:$SQLDDDBROOT/db/DDDB.db/DDDB' -T DC06 -d DDDB`
- `dump_db_to_files.py -c 'sqlite_file:$SQLDDDBROOT/db/LHCBCOND.db/LHCBCOND' -T DC06 -d COND`

Open the file `DDDB/lhcb.xml` and change line 13 into `< catalogref href = "$CONDITIONS_PATH#Conditions" />`

In the options file, put something equivalent to (choose the location where you want your database):

```
DetectorDataSvc.DetDbLocation = "$HOME/cmtuser/LHCb_v23r2/DDDB/lhcb.xml";
ApplicationMgr.Environment = [
"CONDITIONS_PATH": "$HOME/cmtuser/LHCb_v23r2/COND/Conditions/MainCatalog.xml"];

```

In file: `COND/Conditions/MainCatalog.xml` add `< catalogref href = "#TELL1Conf" />` in the catalog with name "Conditions".

Then, at the end of the file add the catalog with name "TELL1Conf":

```
< catalog name="TELL1Conf">
  < catalogref href = "IT/TELL1ConfCatalog.xml#IT"/>
  < catalogref href = "TT/TELL1ConfCatalog.xml#TT"/>
< /catalog>

```

Go to the IT directory and create the file: `TELL1ConfCatalog.xml`. (For the TT exactly the same needs to be done in the TT directory, replacing "IT" by "TT") This file needs to contain the following

```
< ?xml version="1.0" encoding="ISO-8859-1"?>
< DOCTYPE DDDB SYSTEM "../../DTD/structure.dtd">
< DDDB>
< catalog name = "IT">
  < conditionref href = "TELL1Conf/TELL1Config.xml#TELL1Board0"/>
  < conditionref href = "TELL1Conf/TELL1Config.xml#TELL1Board1"/>
  ... (add all TELL1 Boards you need)
< /catalog>
< /DDDB>

```

For the IT TELL1s with source IDs: 0-13, 32-45, 64-77 are used. For the TT the valid source numbers are: 0-10, 32-42, 64-76 and 96-108.

In the IT directory, also create a directory with the name: `TELL1Conf`. In the directory `TELL1Conf`, create the file: `TELL1Config.xml`.

In this file all the configuration parameters for all TELL1s are found. A `TELL1Config` file which can be used

has been attached to this page.

Your database should now be ready to be used. See section "Configuration Parameters" for an explanation of the configuration parameters, which are also present in the conditions database.

-- AnneKeune - 20 Jun 2008

This topic: LHCb > LHCbSTEmulatorOld

Topic revision: r2 - 2008-06-30 - AnneKeune



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback