Start up Guide for n^3He ***Analysis***

Latiful Kabir

Contents

1 Resources

All the start up software and manual related to n^3He experiment can be found from the official git repository with detailed instruction.
The easiest way is to go to n3He wiki (n3he.wikispaces.com) and then click on software from the left panel.
Alternatively, here is a direct link.
Any new change goes to this git repository. You might want to clone the entire repository and pull periodically to be updated. Or you can also download the last release of only what your are interested in.

2 Quick start on basestar

On basestar the data is being transferred and saved to the directory /mnt/idata01/data/ and the analysis library is compiled in a shared directory /home/npdg/n3He/libn3He/lib

So a quick start using the compiled library can be as follows from any user account:

1. Add the following lines to the .bashrc file & save it

```
if [ -f /home/npdg/n3He/libn3He/bin/thisn3He.sh ]; then
      . /home/npdg/n3He/libn3He/bin/thisn3He.sh
fi
```

2. Start a new terminal , go to /home/npdg/n3He/libn3He/analysis/ directory & try running sample analysis scripts from ROOT.

3. The data browser GUI (named as n3HeData) can be opened issuing the command/home/npdg/n3He/n3HeData/n3HeData from the terminal. Copy the binary to your home directory if you will be using the GUI frequently.

```
//OnlineAnalysis.C
//Demo Online Analysis using n3He Library.(By Online I mean 'from
   CINT, doing analysis on the fly, less thoughtful but preferred
   by many or in some conditions')
//Author: Latiful Kabir
//Date: 12/23/14

void OnlineAnalysis()
{
   gSystem->Load("libTree"); //You need to load libTree first in
      order to Load libn3He. This is not necessary if you include
      TTree.h
//file like the Offline analysis script.
   gSystem->Load("libn3He.so");
```

```
    TTreeRaw *t=new TTreeRaw(15142);
    t->Draw("d21[][0]:Iteration$","Entry$>3");
}
```

Other user specific customization can be achieved following the instruction in the ReadMe file in the respective directory. The latest developments and releases can be found from n3He wiki ¿ Software.

3 Event length and file size calculation

The event length set in the clean DAQ at 50 KHz sample rate is :830 . Theoretically the maximum possible value is 50KHz x 16.66 ms = 833 samples per T0 .
But 833 event length gives occasional overlap. So we set to 830.
For dirty DAQ at 100 KHz the theoretical number of samples 100KHz x 16.66 ms = 1666
But to avoid overlap we set to 1660.
More over the DAQ has fixed dead time(readout time) of 35 samples(with no averaging) at the end of any event. This amount of time will be missed for every event.

Clean DAQ event length 830 with nacc=16,16 with hi resolution mode=1
Dirty DAQ event length 1660 with nacc=1,1 with hi resolution mode=0
where nacc=n,n indicates how many samples being averaged.

Thus number of sample per event:
Clean DAQ: (830-35)/16=49.68 $\sim$ 50 (1 header + 49 samples)
Dirty DAQ: (1660-35)=1625 (1 header + 1624 samples)

Run Length calculation:
With 25000 T0 per run:
Clean DAQ file size: 25000 T0 x 50 samples x 4 Byte per sample x 48 Channels =240 $\times 10^6$ Bytes
Dirty DAQ file size (before process): 25000 T0 x 1625 samples x 4 Bytes per sample x 8 channels = 1300 $\times 10^6$ Bytes
Dirty DAQ file seize (after process) : 25000 T0 x 1625 samples x 4 Bytes per sample x 2 channels = 325 $\times 10^6$ Bytes

4 The data file structure

48 Clean DAQ channels divided into two modules:
Each sample is 4 bytes(in hexdump one contiguous pair consists one sample or 4 bytes).

With: mod= module ch = channel

```
mod1event1sample1Ch0 mod1event1sample1ch1 mod1event1sample1ch3 ..... ... ... ... ... mod1event1sample1ch8
mod1event1sample1Ch9 mod1event1sample1ch10 mod1event1sample1ch11 ..... ... ... ... ... mod1event1sample1ch16
mod1event1sample1Ch17 mod1event1sample1ch18 mod1event1sample1ch19 ..... ... ... ... ... mod1event1sample1ch23

mod2event1sample1Ch0 mod2event1sample1ch1 mod2event1sample1ch3 ..... ... ... ... ... mod2event1sample1ch8
mod2event1sample1Ch9 mod2event1sample1ch10 mod2event1sample1ch11 ..... ... ... ... ... mod2event1sample1ch16
mod2event1sample1Ch17 mod2event1sample1ch18 mod2event1sample1ch19 ..... ... ... ... ... mod2event1sample1ch23


mod1event2sample1Ch0 mod1event2sample1ch1 mod1event2sample1ch3 ..... ... ... ... ... mod1event2sample1ch8
mod1event2sample1Ch9 mod1event2sample1ch10 mod1event2sample1ch11 ..... ... ... ... ... mod1event2sample1ch16
mod1event2sample1Ch17 mod1event2sample1ch18 mod1event2sample1ch19 ..... ... ... ... ... mod1event2sample1ch23

mod2event2sample1Ch0 mod2event2sample1ch1 mod2event2sample1ch3 ..... ... ... ... ... mod2event2sample1ch8
mod2event2sample1Ch9 mod2event2sample1ch10 mod2event2sample1ch11 ..... ... ... ... ... mod2event2sample1ch16
mod2event2sample1Ch17 mod2event2sample1ch18 mod2event2sample1ch19 ..... ... ... ... ... mod2event2sample1ch23

....     ....             ....  ...          ................            .......................    .....
....     ....             ....  ...          ................            .......................    .....
....     ....             ....  ...          ................            .......................    .....
....     ....             ....  ...          ................            .......................    .....

Up to N number of events.
```

Now the first sample of any event is the event header with following structure:
mod1event1sample1Ch0=mod1event1sample1Ch1=mod1event1sample1Ch3 = Event Signature-1 (0xaa55f154) ,

mod1event1sample1Ch4= Event Number
mod1event1sample1Ch5 = checksum using path-1
mod1event1sample1Ch6 = sample number
mod1event1sample1Ch7 = checksum using path-2
Then this pattern repeats 3 more times (i.e. in quanta of 8 channels) up to channel-23

mod2event1sample1Ch0 =mod2event1sample1Ch1 = mod2event1sample1Ch3= Event Signature-2 (0xaa55f15f)
mod1event1sample1Ch4= 0 (always)
mod2event1sample1Ch5 = checksum using path-1
mod2event1sample1Ch6 = sample number
mod2event1sample1Ch7 = checksum using path-2
Then this pattern repeats 3 more times (i.e. in quanta of 8 channels) up to channel-23

For Dirty DAQ the data is taken in 8 channels (bank mask B) with one module only and then processed to 2 channels. On Batch panel, M1 signal is connected to marked channel-26 and RFSF signal is connected to marked channel 27. This corresponds to ACD channel-5 (with checksum) and channel-6 (with sample number) where for ADC channel number starts with 0.

This for Dirty DAQ after the processing,

event1sample1Ch0 event1sample1ch1
event2sample1Ch0 event2sample1ch1
... ...
....

Up to N events.

with teh first sample of any event being event signature and event number i.e.
event1sample1ch0 = checksum.
event1sample1ch1 = sample number.

5 The ADC channel to wire map

6 Setting up local version of n3He analysis libary on basestar

Eventually you will want to set up your own version of the analysis library and ROOT environment. This way you can modify any part of the library and add more functionality.

1. Download the source code from here
2. Make any necessary changes in Constants.h file that is required.
3. Do make to compile the library.
4. This will produce libn3He.so (shared library will be inside lib directory).
5. Place the .so file in a directory under LD_LIBRARY_PATH .
6. Now start root and load the Library as: gSystem− >Load("libTree") & gSystem− >Load("libn3He.so") . (For Online analysis)
7. For analysis from a script if you include TTree.h file then you need not to do gSystem>Load("libTree"); Just load gSystem− >Load("libn3He.so"). You need to give full path unless the directory is included in LD_LIBRARY_PATH.
8. If you put the rootlogon.C file in macros directory under Root installation directory, then the library will be loaded automatically and step-4 is NOT necessary.
9. Now from your root script create a Tree by calling: TTreeRaw ∗*my_tree* = new TreeRaw(runNumber#) or Just TTreeRaw t(runNumber#)
10. Do *my_tree*− > *Print*() to print the tree and branch structure.
11. Now do what ever analysis you want using my_tree .
12. Try running example analysis scripts in "analysis" directory.
13. To make life easier it's convenient to put the following command into your ∼/.bash_profile or ∼/.bashrc file:

```
if [ -f /path/to/libn3He/bin/thisn3He.sh ]; then
       . /path/to/libn3He/bin/thisn3He.sh
 fi
```

Note: This version of the libary works both for ROOT 5 and ROOT 6.

7 Setting up local version of data browser on basestar

8 Current tree structure in n3He analysis libary

9 Sample analysis