

*Start up Guide
for
 $n^3\text{He}$ Analysis*

Latiful Kabir

Contents

1	Resources	3
2	Quick start on basestar	4
3	Event length and file size calculation	5
4	The data file structure	6
5	The ADC channel to wire map	8
6	Setting up local version of n3He analysis library on basestar	8
7	Setting up local version of data browser on basestar	9
8	Current tree structure in n3He analysis library	10
9	Sample analysis	10

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 = 240×10^6 Bytes

Dirty DAQ file size (before process): 25000 T0 x 1625 samples x 4 Bytes per sample x 8 channels = 1300×10^6 Bytes

Dirty DAQ file seize (after process) : 25000 T0 x 1625 samples x 4 Bytes per sample x 2 channels = 325×10^6 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

```

modievent1sample1Ch0 modievent1sample1ch1 modievent1sample1ch3 ..... modievent1sample1ch8
modievent1sample1Ch9 modievent1sample1ch10 modievent1sample1ch11 ..... modievent1sample1ch16
modievent1sample1Ch17 modievent1sample1ch18 modievent1sample1ch19 ..... modievent1sample1ch23

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

modievent2sample1Ch0 modievent2sample1ch1 modievent2sample1ch3 ..... modievent2sample1ch8
modievent2sample1Ch9 modievent2sample1ch10 modievent2sample1ch11 ..... modievent2sample1ch16
modievent2sample1Ch17 modievent2sample1ch18 modievent2sample1ch19 ..... modievent2sample1ch23

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 the 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 library 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](#).
lib3He is the analysis library for n3He experiment.
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 library works both for ROOT 5 and ROOT 6.

7 Setting up local version of data browser on basestar

1. Download the source code from here.
2. In bin directory: contains just binary files(obtained after doing make) named **n3HeData**.
3. In libn3He directory: Contains all the library required for running the Data check GUI.
4. Modify and compile the library:(Unless you have already set up the library) You need to change the Data file directory from **Constants.h** in **libn3He** to appropriate directory. Before you do **make** do: **make clean** in the same directory. and then make a fresh shared binary files after you make any changes.
5. Place .so file under **LD_LIBRARY_PATH**: Now make sure the shared library (**libn3He.so**) file is in a directory under your **LD_LIBRARY_PATH**.
OR, a more professional way is as follows: Open **/libn3He/bin/thisn3He.sh** and define **n3HeROOT** path variable to the location where **libn3He** is located. for example: **n3HeROOT=/home/siplu/libn3He/**

Now include command in your **.bashrc** file to run **thisn3He.sh** file each time you open the terminal. i.e. include the following lines:

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

6. Produce binary for GUI: To produce a new binary file named **n3HeData**, go to n3HeData directory, open makefile and change **LIB_INCLUDE** and **GLIBS** to appropriate location for you and then do **make**. It will produce **n3HeData** binary file in the same directory.
7. Run the GUI: Now run the binary file **n3HeData** by doing **./n3HeData** from your recently compiled version in **/n3He_DAQ_GUI/n3HeData/**.
8. Modify the **.desktop** file (included one only for Ubuntu distribution) and **.sh** file in bin directory accordingly and place it in your desktop if you want to run the GUI just by double clicking from your desktop.

8 Current tree structure in n3He analysis library

9 Sample analysis