Optical Alignment System

Analysis tutorial

Author: Hiroki Kanoh

Jan. 26, 2006

Index

.

1． Schematic View of the System

The Optical Alignment System (OASYS) consists of the optics (light source, lens, and CCD camera attached on the Muon Tracking Chamber) and the DAQ system. It aims at explaining the composition and data flow in the DAQ system by this document. Another reference is yielded about the details of the optical system attached in Muon Tracking Chamber.

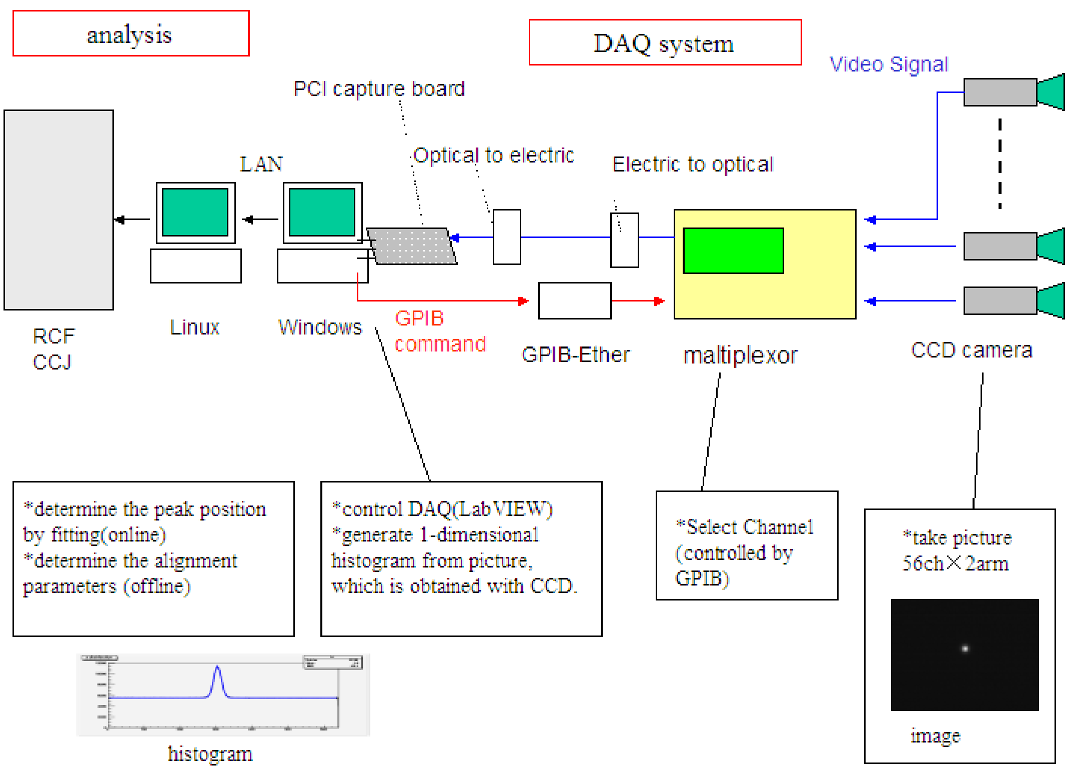


Fig. 1：schematic of the OASYS DAQ

The DAQ system of OASYS has composition as shown in the Fig. 1. A video signal is sent to multiplexor with a coaxial cable from the CCD camera (56 cameras per 1Arm), which is an input device. From windows PC, the multipexor is controlled through a GPIB signal, and one video signal from the CCD camera is chosen. Once the selected video signal was changed into the optical signal by the fiber transfer, and it is sent by light fiber from the experiment hall to the rack room. And then, it is again changed into an electric signal to be captured by the windows PC using PCI capture board. All data measurement is controlled by automatically using the software called LabVIEW installed in windows PC. The information about hardware is shown below.

•CCD Camera (Hitachi KPM1): 56/arm *2
•Multiplexer : Keithley 7001
(slot card) : Keithley 7011 GPIB Address 0::7 (north)
Keithley 7011 GPIB Address 0::7 (south)
•GPIB-Ether Interface : National Instrument GPIB-Ether 100
North IP : 130:199.98.211
Ethernet Address : 08:00:11:09:25:A5
South IP : 130.199.99.135
Ethernet Address : 08:10:11:17:DD:B5
•PCI Capture Board : National Instrument IMAQ PCI-1407
•Windows Machine
oasys1(north) : oasys1.phenix.bnl.gov (130.199.98.243)
oasys2 (south): oasys2.phenix.bnl.gov (130.199.98.244)
•Linux Machine
oasys3 : oasys3.phenix.bnl.gov (130.199.98.245)

Below applications are used in windows PC (oasys1, oasys2) and Linux PC (oasys3).

•LabVIEW (Windows)
•ROOT (Linux)
•SAMBA server (Linux)
•Apache (Linux)

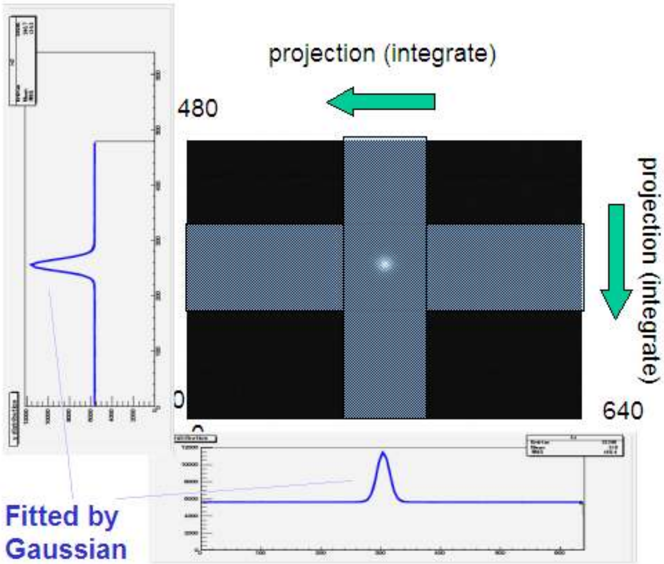
2. Algorithm of DAQ

In this section, an algorithm of DAQ system is simply explained. It is essential that how about process the picture that obtained from the CCD camera. Another reference was referred to about the analysis of movement of muon tracking chamber.

In order to use the image data obtained from OASYS and to analyze movement of tracking chamber, we should express the position of a light spot in a picture numerically before analysis. In the OASYS DAQ system, after performing "projection operation" to two directions (horizontal and vertical) to 2-dimensional image data, and generating two 1-dimensional histogram data, we determine the position of a light spot by fitting. The algorithm is as follows.

1. Convert the obtained picture (jpeg format) into 640x480 sized matrix. The element of this matrix corresponds to the luminosity in each pixel of a picture.
2. Obtain the picture of 128 sheets from 16 sheets continuously about one CCD camera. And the sum of all matrixes that is converted from all pictures as mentioned above each time, is taken. This matrix is determined as "image matrix."
3. Search for an element with the largest value in the image matrix. The neighborhood of this position corresponds to light spot of a picture.
4. Perform "projection operation" over the range of 8 sigma and generate histogram. This operation should be performed toward two directions (horizontal and vertical). This projection operation is defined as integration like the following formula.

$$
\begin{aligned}
f_x(i) &= \sum_{j=c_x-5\sigma_x}^{c_x+5\sigma_x} b(i,j) \\
f_y(j) &= \sum_{i=c_y-5\sigma_y}^{c_y+5\sigma_y} b(i,j) \\
&\quad (0 \leq i < 640,\ 0 \leq j < 480)
\end{aligned}
$$



where $f_x(i)$, $f_y(j)$ means image matrix and (c_x, c_y) means the position of light spot, which is searched for in process 3.

5. Fit the histogram with the below function which added the constant term to gaussian function, and determine the position of light spot p[1].

$$f(x) = p[0] + p[3] \exp\left(-\frac{(x-p[1])^2}{2p[2]^2}\right)$$

The program based on the LabVIEW on each windows PC performs the process of 1 to 4, and a histogram is outputted to the Linux server (oasys3) through samba. The process of 5 is performed using a SHELL script and ROOT on Linux. Details of each program are given in the following paragraph.

3. Detail of Code

The program group currently used for the analysis of OASYS is explained in this section.

3.1. DAQ

The main portions of DAQ system of OASYS are controlled using framework called LabVIEW of National Instruments. Therefore, the skill of G language (language used by LabVIEW) is needed for understanding the contents of the code. The program used for DAQ control is in the following places. The structure of a folder is common to oasys1 and oasys2. This program performs the process of 1 to 4 that is mentioned in the foregoing section.

E:¥LabOAsys	¥OAsysRun.exe	program
	¥data¥confDAT¥	configure file
	¥data¥error¥	error log
	¥data¥hist¥	histogram
	¥data¥image¥	image (jpeg)
	¥data¥raw¥	raw data (image matrix)
	¥source¥	source
¥¥OAsys3¥OAsys3-smb¥		SAMBA

3.1.1 main body of code

An executable file is "E:¥LabOAsys¥OAsysRun.exe." If it starts in a double click, it will measure automatically. One reboot per one day is recommended. The easy directions below for the following clause are carried (extracted from the DAQ operation manual).

2. How to use

Bring up

◆Bring up Hardware (3 steps)

1.Check Muon Tracking LV

Note!

The MuTr LV system should only be turned on by muTr experts. So,when the LV power has been turned off, ask to the MUTR expert

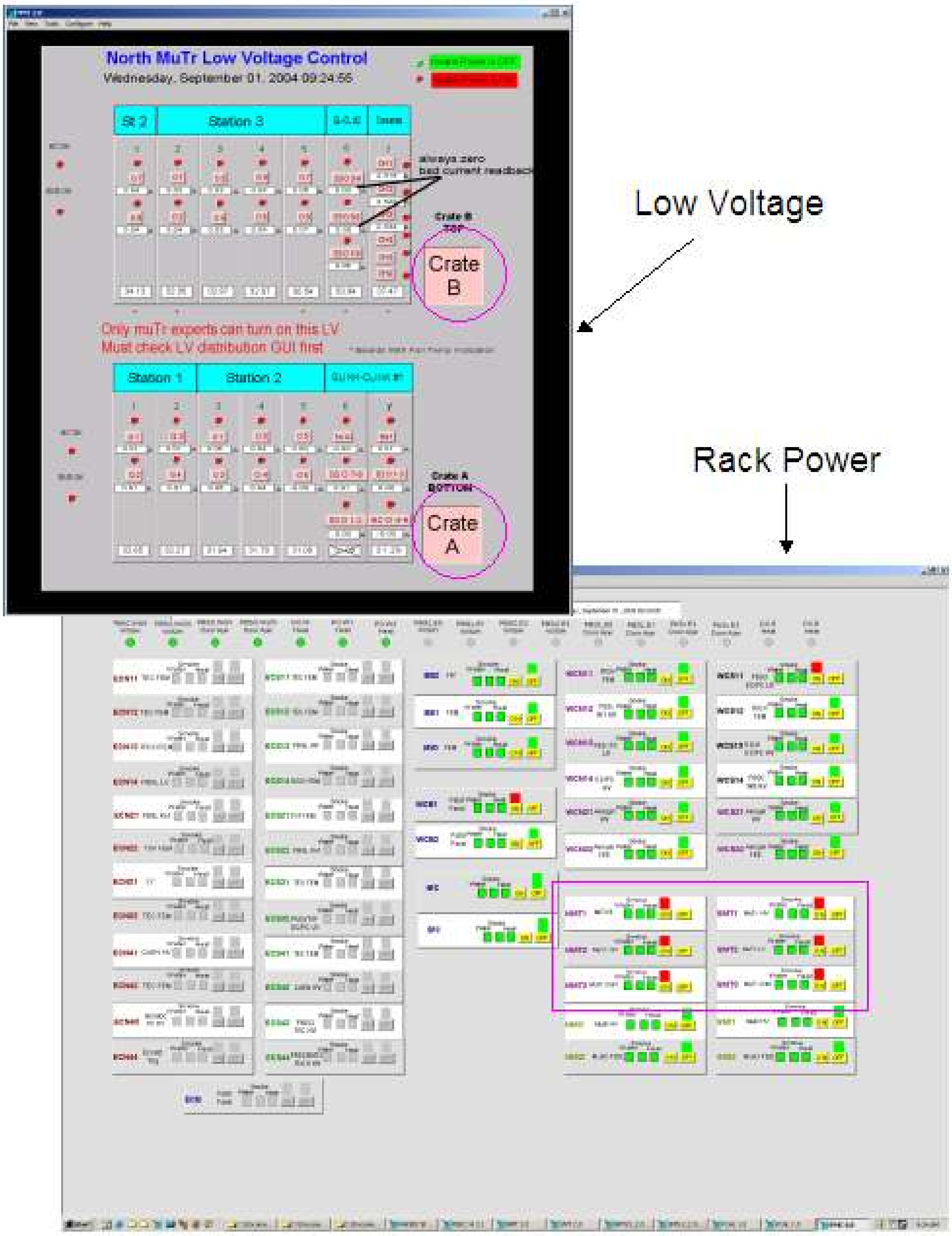
ⅰ. Rack Power

The status of low voltage rack power for the MuTr can be seen on the screen of phoncs12, which is located at the north west corner of the counting room. It is commonly used to control LV for entire PHENIX.In the "Rack Monitoring and Control" window, you can find 6 segments of MuTr low voltage control (NMT1, NMT2, NMT3, SMT1, SMT2, and SMT3).

Click "**ON**" button of voltage control and check indicator above the "**ON**" button turns from Green to Red.

ⅱ. **Low Voltage**

On the same screen as "Rack Monitoring and Control" there are graphical user interface windows called "SMT" for the south muon arm and "NMT" for the north. Click "Crate A" and "Crate B" button, and check "Crate A" and "Crate B" button turns from Green to Red.


Low Voltage
Rack Power
North MuTr Low Voltage Control
Wednesday, September 01, 2004 09:24:56
St 2
Station 3
always zero
bad current readback
Crate B
TOP
Crate
B
Only muTr experts can turn on this LV
Must check LV distribution GUI first
Station 1
Station 2
Crate A
BOTTOM
Crate
A

2.Turn cameras on

To turn cameras on, use graphical user interface called “North (South) Muon Tracking Low Voltage Power Control” (NMTLV_2.0,SMTLV_2.0).
On this window, you can see “Turn Cameras On” button and 56 segment.

of camera indicators.

Click “Turn Cameras ON” button and check indicator turn on Red

You must check the racks and crates are on

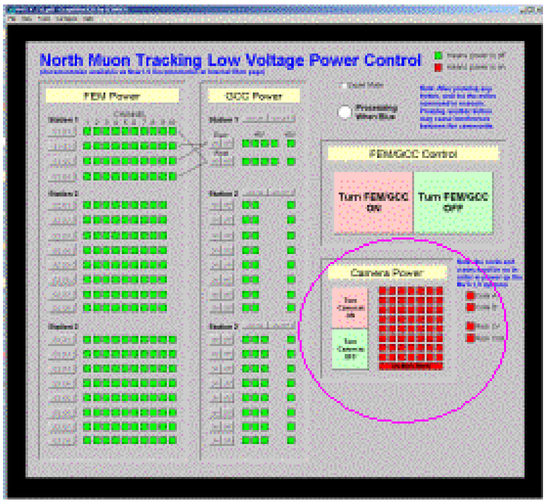


Fig.North Muon Tracking Low Voltage

3.Tunr light source on

At last, turn light source using graphical user interface called “North (South) Muon Tracker monitoring” (MuTr.N, MuTr.S) .
Click “light source” control and check it turn on Red.

Fig. North Muon Tracker monitoring
Note : GPIB Reset is obsolete.
Because, it is never needed in the new DAQ system.

◆Bring up DAQ Software

*You must done procedure bellow both oasys1and oasys2, which is allocated in the rack room.

Double click "OAsysRun " on the Desktop or directly bring up executable file

(Program Path E:¥LabOAsys¥OAsysRun.exe)

Error check

When you would bring DAQ software up, automatically data taking will be started and system take data every 10 min. After you bring DAQ software up, you should check if system works properly or not. Checking procedure is as bellow.

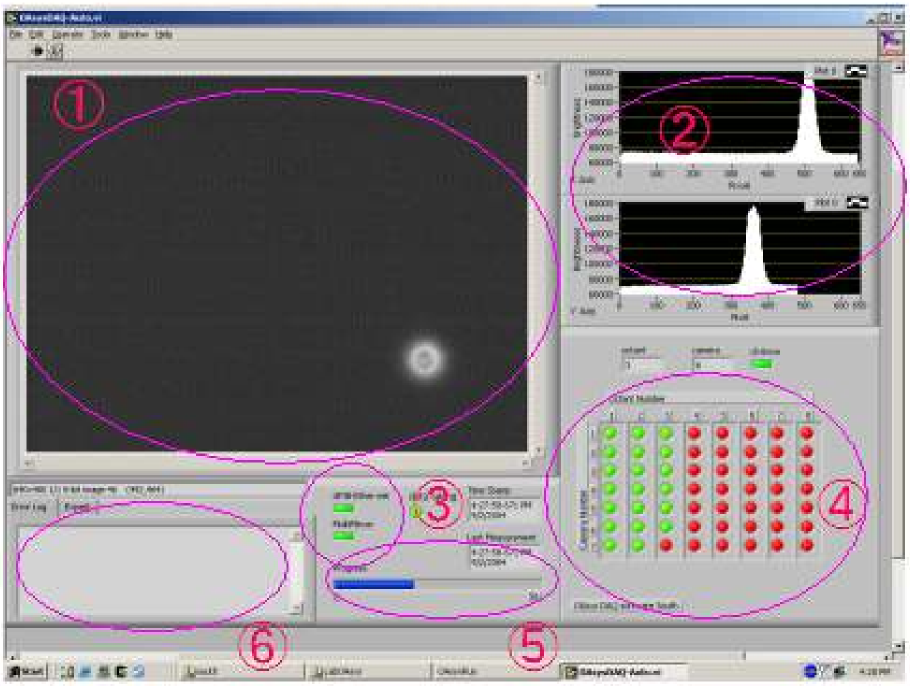


Fig. DAQ software
(on Windows)

Check it

- ✓Is there light image on display? ⇒ Look ①
- ✓Is there peak on the histogram? ⇒ Look ②
- ✓CPIB-Ether interface & Multiplexer status is tuned on Green ? ⇒ Look ③
- ✓Each CCD status should be Green ? ⇒ Look ④

①image display
if there are no signal from CCD, gray image will be displayed.

Note : Good CCD has peak but some one has no light spot at all. List is shown later.

②Histogram display
③GPIB-Ether interface status (upper) and, Multiplexer status (lower)

indicator means
Good: Green
Bad : Red

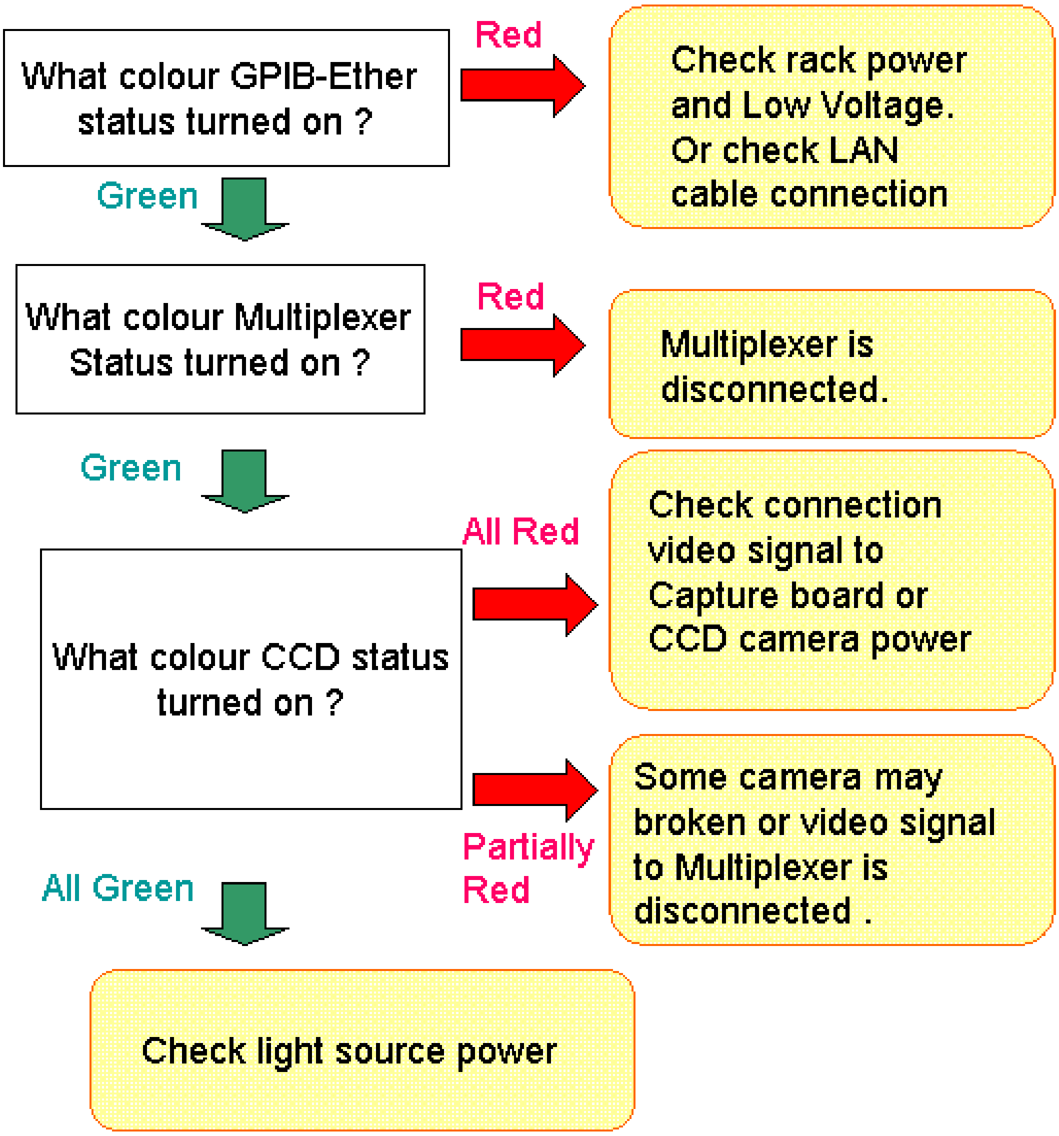
④CCD camera status 7 camera/octant * 8 octant

indicator means
Good : Green
Not measured yet or Bad: Red
Note : if you wait more than 10 min, should not be Red

⑤Progress bar
⑥Error log & expert tab

Error handling

In case of at least 1 CCD camera turned on Red or no light image in the image display for all CCD camera, you should manage error according to procedure bellow.

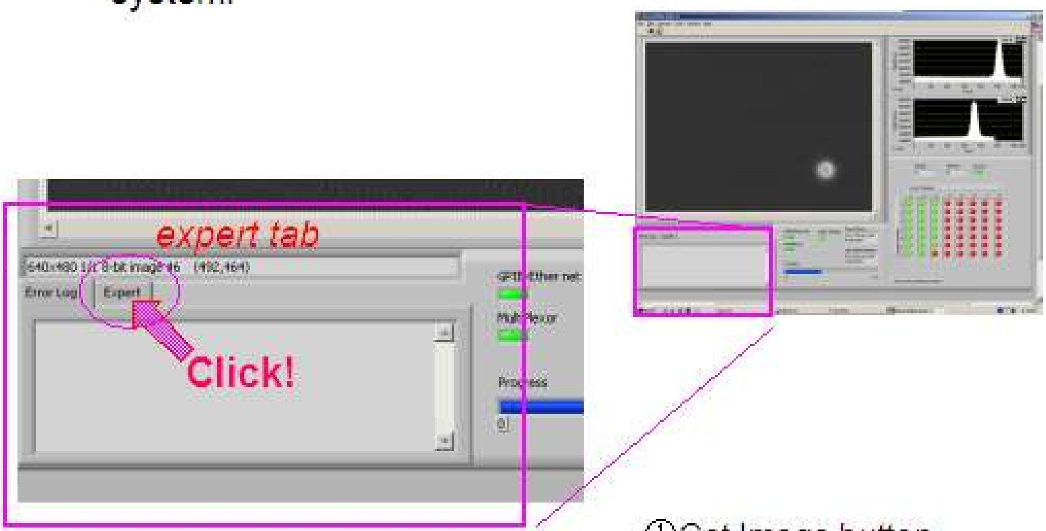


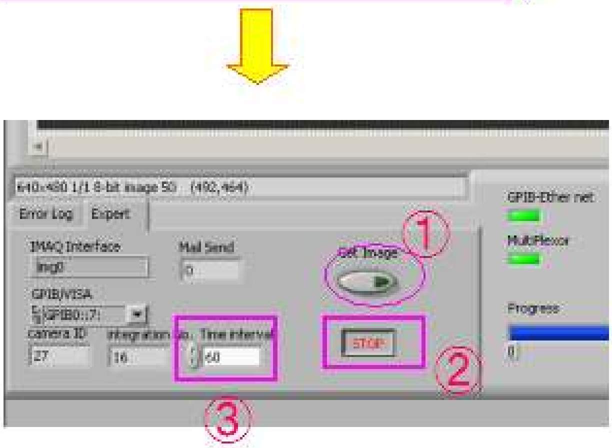
If system does not work correctly yet after your error management, or in case of any other error, Call expert.

System control

If you are DAQ expert, you can control some measurement parameter from DAQ software graphical user interface. To control DAQ, you should do as bellow.

1. Click “expert” tab in the left bottom of the window.
2. Then tab window change as bellow picture. From this window, you can control some part of DAQ system.





①Get Image button
Start measurement immediately

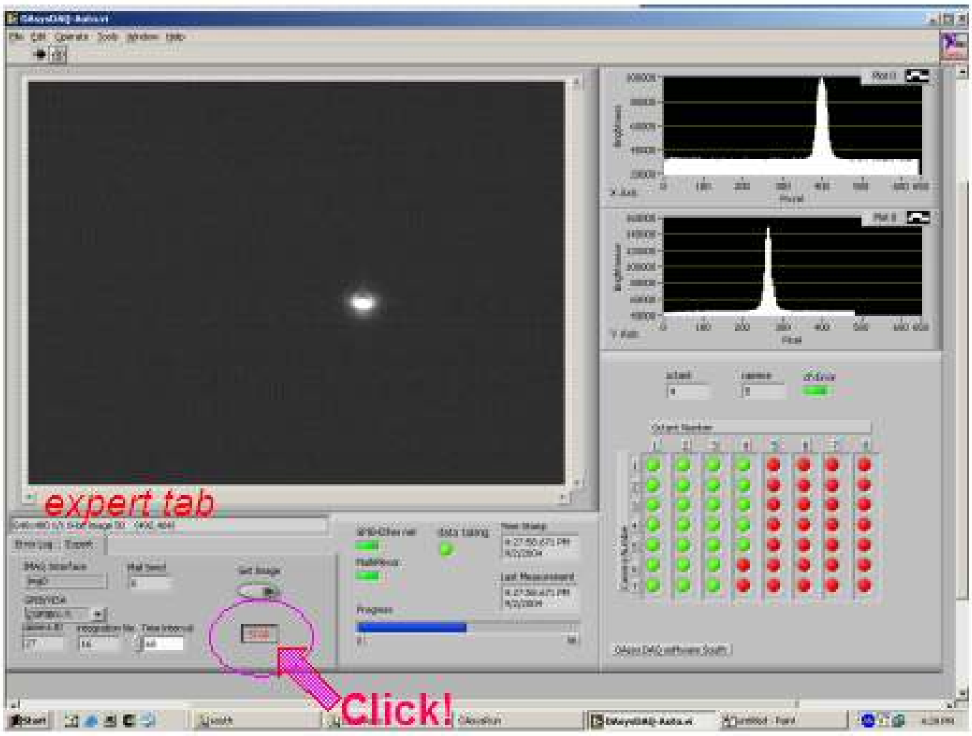
②Stop button
Stop measurement

③Time interval control
Set time interval among measurement in seconds.
(default value : 600 sec)

Exit or Restart

If you want to kill or restart program, do as bellow.

1.Go to "expert" window.
2.Click "STOP" button.
If system is during data taking, program will be stopped after it take 56 images for all CCD cameras
3.Close Window
4.If you want to restart program, double click program icon again.



3.1.2. Configure file

The configuration file of measurement is settled in ¥data¥confDAT¥. The items and role are as follows.

・Nintg.conf : definition of the iteration of integration of picture. (North Arm)

・Sintg.conf : Same as the above. (South Arm)

Both of the files are written by text format, and it has become the length of 56 lines, and width 2 sequences. A left-hand side numbers expresses a channel and a right-hand side number expresses a setting value (iteration of integration of a picture).

・Nsig.conf : The definition of the integration range of a picture. (North Arm)

・Ssig.conf : Same as the above. (South Arm)

Both of the files are written by text format, and it has become the length of 56 lines, and width 3 sequences. A left-hand side number expresses a channel and right-hand side 2 numbers express a setting value (width of the peak of a horizontal and vertical directions). Projection operation is performed over the 8 times as much range as this value focusing on a peak.

3.1.3. Preservation of Data

The three types of data were saved at windows PC (under E:¥LabOAsys¥).

1. Histogram: 1-dimensional picture information generated by projection operation. It saves at ¥hist.
2. Image data: Image data of Jpeg format. It saves at ¥image.
   !! Since this picture is not integrated, it is not suitable for analysis.
3. Binary data (raw data): Data of only the peak circumference of a image matrix. Binary format. It saves at ¥raw.

These data are saved through samba also at Linux (oasys3). The preservation place in windows is ¥¥Oasys3¥OAsys3-smb and the directory in Linux is /north and /south directory below /data1/u/OAsys-samba

3.1.4. Remote control

The DAQ program is available from a remote host through the VNC interface. The port number for the VNC is “5900”.

3.2 online analysis

Online analysis of a histogram is performed in the Linux PC (oasys3). This is the process of 5 of algorithm. The code, input and output files, which are used for this analysis, exist in the directory under /data1/u/OAsys-samba in oasys3.

3.2.1 analysis code

The program group for online analysis is put on /data1/u/OAsys-samba/online. There are the following three executable files to use it. .

- submit.pl --- Generalizes a series of processing. Described in Perl.
- oasysfit --- Execute the fitting of a histogram. Described in C++.
- genhtml.bin --- Output the results of CCD monitoring. Described in C++.

The outline of each code is explained below.

- submit.pl

This script calls "oasysfit" and "genhtml.bin" inside of itself, and performs a series of processing's. It performs at intervals of 10 minutes using cron daemon. The input-and-output file in an inside is as follows. .

Input

- /data1/u/OAsys-samba/north/hist, /data1/u/OAsys-samba/south/hist:
  (histogram in text format)

Output

- /data1/u/OAsys-samba/fit/north(south) : fitting result
- /data1/u/OAsys-samba/north/html, /data1/u/OAsys-samba/south/html;
  monitoring result.

If you want to run this script, just command from terminal

```
[user]$ ./submit.pl
```

After your execution, histogram files are moved to other directory from the input directory. (moved to "/data1/u/OAsys-samba/north(south)/done")

・ oasysfit

This script executes a fitting of histograms. ROOT on C++ is used for this. The usage of this program is

```
[username]$ ./oasysfit   Input   Output   Arm   UNIX-time
```

Here,

Input; histogram file from DAQ output(*.hist)。

Output; fitting result in the text format(*.fit)。

Arm; North (0), South (1)

The source code exits at "source¥OASYSFit".

・ genhtml.bin

The program acts as the monitor of the state of the present camera using the result of fitting. A result is outputted to HTML and it can peruse now out of a network using Apache. Directions are as follows.

```
[username]$ ./genhtml.bin Output Fitfile(north) Fitfile(south)
```

Output; monitoring result(HTML)

Fitfile; fitting result from "oasysfit". /online/NoasysDB.fit or /online/SoasysDB.fit

It is the /data1/u/OAsys-samba/WWW directory which is exhibited outside by Apache.

<u>3.2.2 data backup</u>

The data acquired by OASYS in 2003 and afterwards is saved at data archive area in oasys3 (/data2/u/Dat_archive/.)

3.3 offline analysis

In offline analysis, it asks for the time-dependent movement of the muon tracking chamber based on the position information on the picture obtained from DAQ. Details of algorithm are given by another reference. The code of offline analysis is operated in the Linux environment where ROOT is installed. It is desirable to carry out on RCF or CCJ. Since it becomes the hindrance of the online analysis, don't carry out on oasys3. The code group is prepared for RCF.

/direct/phenix+workarea/hkanou/OASYS

The directory composition below /OASYS is as follows.

/alignment; codes for analysis

/online; code for re-analysis of a histogram (same as that of online analysis)

/north; data of OASYS in North Arm

/south; data of OASYS in South Arm

Step.1

The file as a result of two or more fitting (*. fit) is summarized to single Ntuple. This work is done using the Macro of /north(south)/fit/makeNtuple.C. First, the list of *.fit files is created. Then, you should executes macro “makeNtuple.C”

```
[username]$ ls *.fit > hlist;
[username]$ root makeNtuple.C;
```

Ntuple is outputted as oasysDB.root.

Step.2

The data, which can be used for analysis, is sorted out. What has the bad accuracy of fitting, and what has the greatly shifted peak position from the averaged one, are excepted. This work is done using /alignment/Select/soasys.

After bringing oasysDB.root to the directory in which soasys is exits, you should type

```
[username]$ ./job
```

oasysSEL.root is generated by this. The information of data selection is included in this ntuple.

Setp.3

The movement of muon tracking chamber is calculated This work is done using /alignment/MOASYS/oasys. oasysDB.root and oasysSEL.root should be placed on the same directory. Then, just type,

```
[user]$ ./job
```

The oasysALI.root is outputted by this command. There are some convenient tools in /tools. For example, a total of three files of oasysDB.root and oasysSEL.root and oasysALI.root can be collectively seen using /tools/oasysFriend.C.

Details of data

Here, the contents of Branch of each Ntuple are explained.

OasysDB.root

```
TTree octant1("octant1","octant1");
octant1.Branch("time",&time,"time/I");
octant1.Branch("stat",connect,"stat[7]/I");
octant1.Branch("cid",cid,"cid[7]/I");
octant1.Branch("octant",&octant,"octant/I");
octant1.Branch("arm",&arm,"arm/I");
octant1.Branch("x_img",xpeak,"x_img[7]/F");
octant1.Branch("y_img",ypeak,"y_img[7]/F");
octant1.Branch("x_img_chi2",chi2x,"x_img_chi2[7]/F");
octant1.Branch("y_img_chi2",chi2y,"y_img_chi2[7]/F");
octant1.Branch("x_img_error",xerror,"x_img_error[7]/F");
octant1.Branch("y_img_error",yerror,"y_img_error[7]/F");
octant1.Branch("x_img_sigma",xsigma,"x_img_sigma[7]/F");
octant1.Branch("y_img_sigma",ysigma,"y_img_sigma[7]/F");
octant1.Branch("x_img_signal",signlx,"x_img_signal[7]/F");
octant1.Branch("y_img_signal",signly,"y_img_signal[7]/F");
octant1.Branch("x_img_ndf",ndfx,"x_img_ndf[7]/F");
octant1.Branch("y_img_ndf",ndfy,"y_img_ndf[7]/F");
octant1.Branch("x_img_offset",offsetx,"x_img_offset[7]/F");
octant1.Branch("y_img_offset",offsety,"y_img_offset[7]/F");
```

time; Unix time

stat; CCD camera connet (0=on, 1=off)

cid; identifical number of CCD camera (1 to 56)

octant; identifical number of octant (1 to 8)

arm; North Arm =0, South Arm=1

x_img; Peak position of light spot (horizontal)

y_img; Peak position of light spot (vertical)

x_img_chi2; fitting chi2 (horizontal)

y_img_chi2; fitting chi2 (vertical)

x_img_error; fitting error for the peak position (horizontal)

y_img_error; fitting error for the peak position (vertical)

x_img_sigma; width of the Gaussian (horizontal)

y_img_sigma; width of the Gaussian (vertical)

x_img_signal; height of the peak (horizontal)

y_img_signal; height of the peak (vertical)

x_img_ndf; number of degree of freedom (horizontal)

y_img_ndf; number of degree of freedom (vertical)

x_img_offset; constant term of fitting function (horizontal)

y_img_offset; constant term of fitting function (vertical)

oasysSEL.root

```
TTree *sel1 = new TTree(branch_name_out,branch_name_out);

sel1->Branch("select",sflug,"select[7]/I");
sel1->Branch("N_cm",&N_cm,"N_cm/I");
```

select; result of data selection. (0=selected, 1=rejected)

N_cm; the number of selected CCD (0 to 7)

OasysALI$1.root____

```
TTree *ali1 = new TTree(branch_name_out,branch_name_out);

ali1->Branch("dx_img",dx_img,"dx_img[7]/F");
ali1->Branch("dy_img",dy_img,"dy_img[7]/F");
ali1->Branch("dx1_m",&dx1_m,"dx1_m/F");
ali1->Branch("dx2_m",&dx2_m,"dx2_m/F");
ali1->Branch("dx3_m",&dx3_m,"dx3_m/F");
ali1->Branch("dphi1_m",&dphi1_m,"dphi1_m/F");
ali1->Branch("dphi2_m",&dphi2_m,"dphi2_m/F");
ali1->Branch("dphi3_m",&dphi3_m,"dphi3_m/F");
ali1->Branch("dx1_rms",&dx1_rms,"dx1_rms/F");
ali1->Branch("dx2_rms",&dx2_rms,"dx2_rms/F");
ali1->Branch("dx3_rms",&dx3_rms,"dx3_rms/F");
ali1->Branch("dphi1_rms",&dphi1_rms,"dphi1_rms/F");
ali1->Branch("dphi2_rms",&dphi2_rms,"dphi2_rms/F");
ali1->Branch("dphi3_rms",&dphi3_rms,"dphi3_rms/F");
ali1->Branch("data_num",&data_num,"data_num/F");
ali1->Branch("chi2",&chi2,"chi2/F");
ali1->Branch("chi2_x",&chi2_x,"chi2_x/F");
ali1->Branch("chi2_y",&chi2_y,"chi2_y/F");
ali1->Branch("residual_x",residual_x,"residual_x[7]");
ali1->Branch("residual_y",residual_y,"residual_y[7]");
ali1->Branch("residual",residual,"residual[7]");
ali1->Branch("time_0",&t_0,"time_0/I");
ali1->Branch("stat_0",stat_0,"stat_0[7]/I");
```

____

dx_img; displacement of image (horizontal)

dy_img; displacement of image (vertical)

dx1_m; dx

dx2_m; dy

dx3_m; dz

dphi1_m; pitch

dphi2_m; yaw

dphi3_m; roll

dx1_rms to dphi3_rm; not used.
data_num; number of available CCD
chi2_x; chi2 of LS (horizontal)
chi2_y; chi2 of LS (vertical)
residual_x; residual of horizontal direction
residula_y; residual of vertical direction
residual; length of residual
time_0; unix time of initial position
stat_0; CCD connection at initial time