ガス制動放射測定実験における Data taking system の使い方の簡単な説明

平成 10 年 12 月 22 日

ガス制動放射線測定実験におけるテイキングシステムの使い方の簡単な説明を書きます。詳しい説明書としては、KEK Internal 94-17 (Tutorial for Running UNIDAQ DECstation with VME-CAMAC Interface) と UNIDAQ Users Manual を読んで下さい。

1 準備

ここでは、CAMAC でデータをとる場合について書きます。はじめに、UNIX に Login します。次に NIM、CAMAC の順に電源をいれます。NIM や CAMAC は安定するまでに時間がかかるので、必要以上に電源をきらないで下さい。最後にクレートコントローラのスイッチを online にします。ここで注意することとして、CAMAC や PC の電源を ON、OFF する時は、必ずクレートコントローラのスイッチを offline にして行なって下さい。

UNIX の term が開きましたら、カレントディレクトリーを $HOME/v2.3/user/bin に移します。ここに画面表示で使うマクロと言うものが置いてあります。

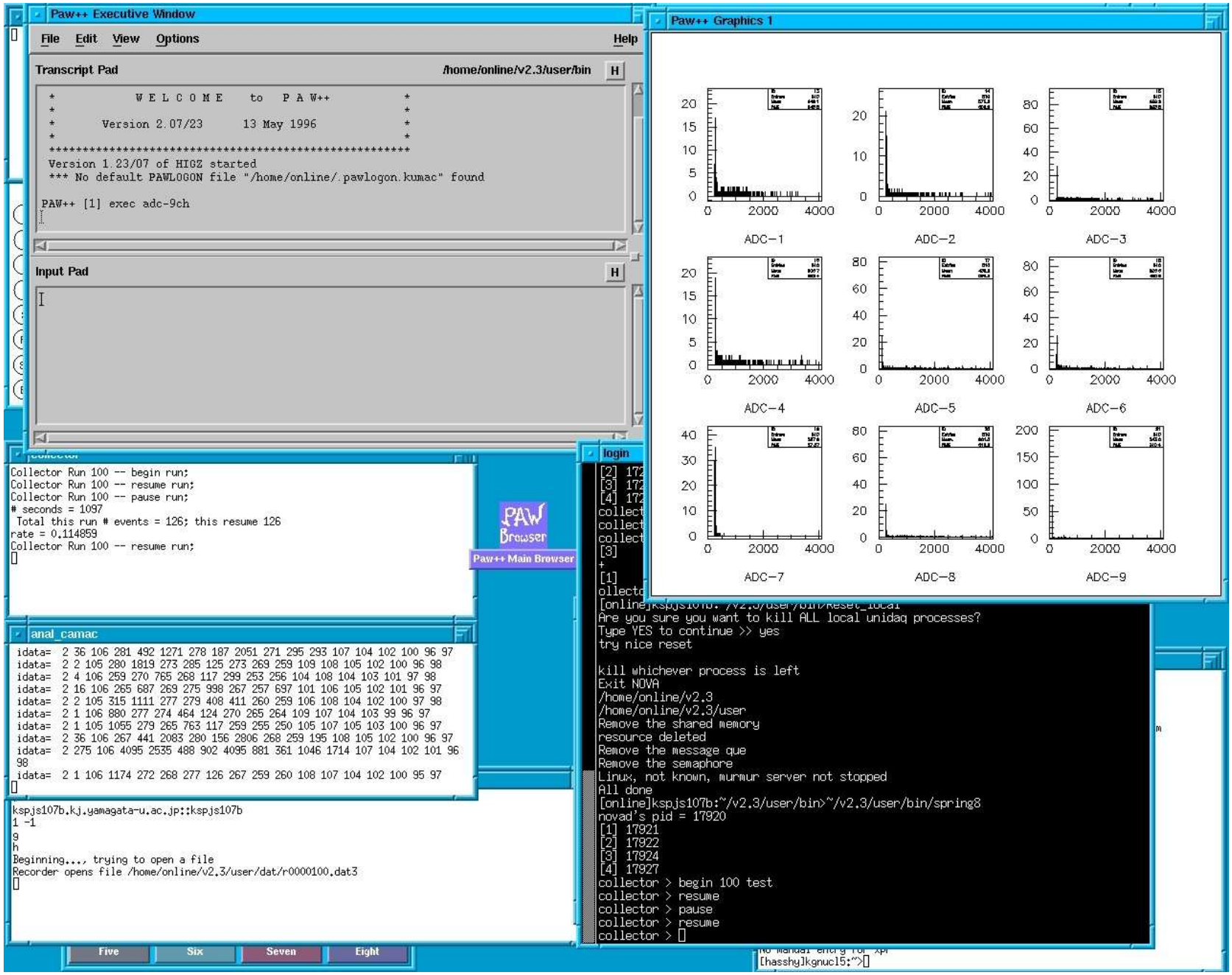
```
% cd ~/v2.3/user/bin
```

次に、spring8 という shell script を実行します。このファイルには、DAQ を起動させるためのコマンドが入っています。

```
% ~/v2.3/user/bin/spring8
```

これでもうデータを取り込める状態ができました。画面には、paw++や term などが開かれ、プロンプトは以下のように表示され、次のページのような画面が現れます。

```
collector>
```


Paw++ Executive Window
File Edit View Options Help
Transcript Pad
/home/online/v2.3/user/bin
* W E L C O M E to P A W++ *
* Version 2.07/23 13 May 1996 *
Version 1.23/07 of HIGZ started
*** No default PAWLOGON file "/home/online/.pawlogon.kumac" found
PAW++ [1] exec adc-9ch
Input Pad
Paw++ Graphics 1
ADC-1
ADC-2
ADC-3
ADC-4
ADC-5
ADC-6
ADC-7
ADC-8
ADC-9
Collector Run 100 -- begin run;
Collector Run 100 -- resume run;
Collector Run 100 -- pause run;
seconds = 1097
Total this run # events = 126; this resume 126
rate = 0.114859
Collector Run 100 -- resume run;
PAW Browser
Paw++ Main Browser
anal_camac
idata= 2 36 106 281 492 1271 278 187 2051 271 295 293 107 104 102 100 96 97
idata= 2 2 105 280 1819 273 285 125 273 269 259 109 108 105 102 100 96 98
idata= 2 4 106 259 270 765 268 117 299 253 256 104 108 104 103 101 97 98
idata= 2 16 106 265 687 269 275 998 267 257 697 101 106 105 102 101 96 97
idata= 2 2 105 315 1111 277 279 408 411 260 259 106 108 104 102 100 97 98
idata= 2 1 106 880 277 274 464 124 270 265 264 109 107 104 103 99 96 97
idata= 2 1 105 1055 279 265 763 117 259 255 250 105 107 105 103 100 96 97
idata= 2 36 106 267 441 2083 280 156 2806 268 259 195 108 105 102 100 96 97
idata= 2 275 106 4095 2535 488 902 4095 881 361 1046 1714 107 104 102 101 96
98
idata= 2 1 106 1174 272 268 277 126 267 259 260 108 107 104 102 100 95 97
kspjs107b.kj.yamagata-u.ac.jp::kspjs107b
1 -1
g
h
Beginning..., trying to open a file
Recorder opens file /home/online/v2.3/user/dat/r0000100.dat3
login
Are you sure you want to kill ALL local unidaq processes?
Type YES to continue >> yes
try nice reset
kill whichever process is left
Exit NOVA
/home/online/v2.3
/home/online/v2.3/user
Remove the shared memory
resource deleted
Remove the message que
Remove the semaphore
Linux, not known, murmur server not stopped
All done
[online]kspjs107b:~/v2.3/user/bin>~/v2.3/user/bin/spring8
novad's pid = 17920
[1] 17921
[2] 17922
[3] 17924
[4] 17927
collector > begin 100 test
collector > resume
collector > pause
collector > resume
collector >
Five
Six
Seven
Eight

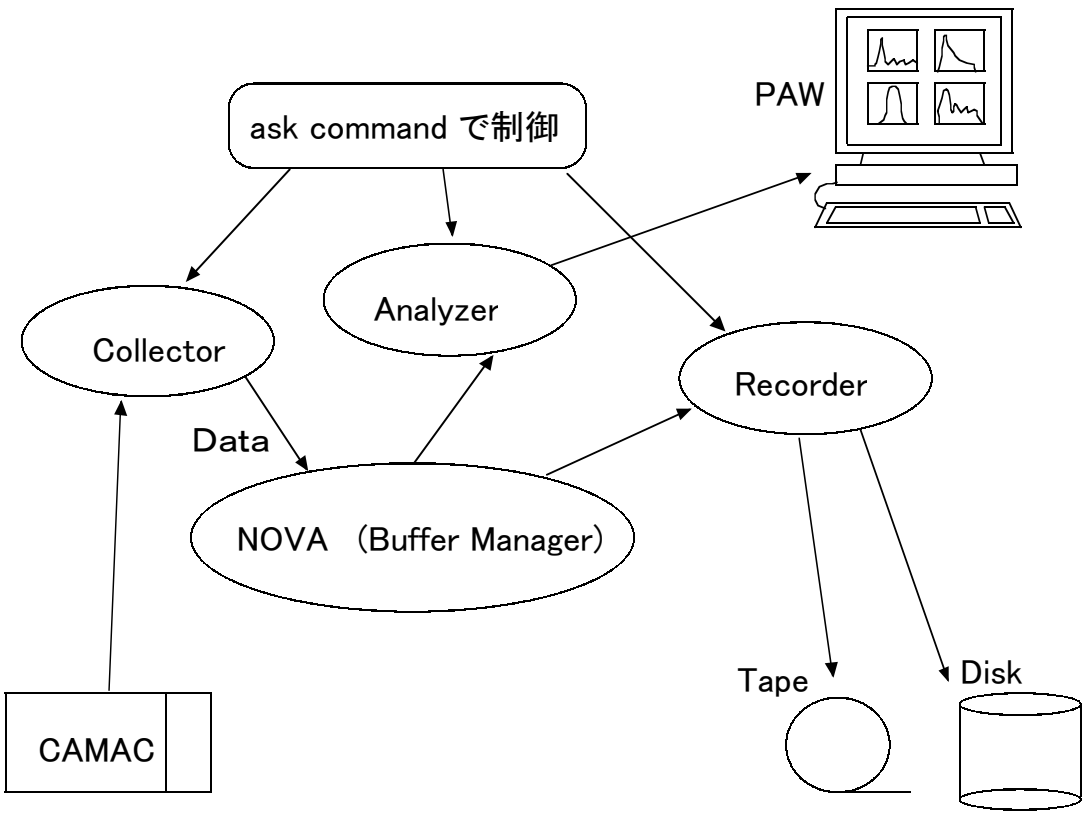


図 1: Data の流れ

となっています。ここでシステム中の簡単なデータの流れを図 1 に表して置きます。

次に、paw++のコマンドラインに移って、マクロを実行させておきます。マクロとは、paw++専用の shell script です。今回の実験のために次のマクロを用意しました。ファイルは、$HOME/user/bin/に置いてあります。

adc-all.kumac	ADC の全チャンネルを表示する
adc-9ch.kumac	PWO の全 9 チャンネルを表示する
adc-0.kumac	ADC の 0 チャンネルを表示する
adc-1.kumac	ADC の 1 チャンネルを表示する
adc-2.kumac	ADC の 2 チャンネルを表示する
adc-3.kumac	ADC の 3 チャンネルを表示する
adc-4.kumac	ADC の 4 チャンネルを表示する
adc-5.kumac	ADC の 5 チャンネルを表示する
adc-6.kumac	ADC の 6 チャンネルを表示する
adc-7.kumac	ADC の 7 チャンネルを表示する
adc-8.kumac	ADC の 8 チャンネルを表示する
adc-9.kumac	ADC の 9 チャンネルを表示する
adc-10.kumac	ADC の 10 チャンネルを表示する
adc-11.kumac	ADC の 11 チャンネルを表示する
adc-12.kumac	ADC の 12 チャンネルを表示する
adc-13.kumac	ADC の 13 チャンネルを表示する
adc-14.kumac	ADC の 14 チャンネルを表示する
adc-15.kumac	ADC の 15 チャンネルを表示する
position.kumac	入射位置を表示する
pwo-allsum.kumac	PWO9 本の和を表示する
hbook.kumac	すべてのヒストグラムをコピーして hbookfile に落とす

ここでは、adc-9ch.kumac を実行させておきます。

```
paw++> exec adc-all
```

とします。これで、data 収集の準備が整いました。

2 Data収集

簡単に collector コマンドについて説明します。begin は Run の設定、resume は Data 収集の開始、pause は Data 収集の中断、end は Run の終了です。以下に例をあげて説明します。

はじめに collector のコマンドラインに戻って begin と打ちますが、ここで run number や取ったデータに書き込んでおきたい comment を打ち込みます。

```
collector> begin <run number> <comment>
```

例えば、

```
collector> begin 100  energy = 1GeV
```

とすれば、

```
~/v2.3/user/dat/
```

に r0000100.dat という run number が入った data file を作成し、ここにデータが入ります。ここで注意しておきますが、同じ run number で RUN を行なってもファイルは上書きされず、r0000100.dat2 というファイルが作られます。

次に resume します。

```
collector> resume
```

これによって、実際にデータが読み込まれていき、paw++で表示されていきます。違うマクロを使いたいときは、paw++のコマンドラインで CTL + c と打ちます。そうすると、いままで実行されていたマクロが終了して、paw++のコマンドラインに戻ります。

paw はヒストグラムを表示するために、analyzer とメモリを共有しています。ですので paw からヒストグラムを直接操作することは出来ません。paw++でヒストグラムに対して操作を行なう時は、ヒストグラムを別の ID にコピーしてから行なって下さい。例えば、12 番のヒストグラムを GAUSS でフィットしたい時はまず 1000 番にコピーしてからフィットします。

```
paw++> hi/copy 12 1000
```

その後で、

```
paw++> hi/fit 1000 g
```

paw++で表示しているヒストグラムのデータのクリアーする時は、まず paw++ で走らせているマクロを終了させ、その後 collector を pause させます。

```
collector>pause
```

pause させたら、analy_camac のプロンプトで Reset させます。この時クリアーさせたいヒストグラム ID を入れてやります。何も入れないと、すべてのヒストグラムをクリアーします。

```
anal_camac> Reset 17
```

ここで、data 収集の流れを例示します。

```
collector>begin 100 comment 1
collector>resume
        ・
        ・<RUN 100 Data 収集中>
        ・
collector>pause
```

```
collector>end
collector>begin 200 comment 2
collector>resume
         ・
         ・
         ・<RUN 200 Data 収集中>
         ・
collector>pause
collector>end
```

この段階で注意することは、一度 resume をさせたら、何かデータを取らないと終了出来ません。無理に終了させると、システムが使えなくなります。もしデータが取れない場合は、Master trigger にパルサーのシグナル等をいれればいいと思います。

3 終了の仕方

データを取るのを終了するときは、pause 、end させます。

```
collector> pause
collector> end
```

次に collector と paw++のコマンドラインで exit と打ちます。すると、collector と paw++は終了します。

```
collector>exit
paw++>exit
```

Do you want to exit ? と聞かれますので、ここで yes をクリックします。

最後に nova をとめます。これによって DAQ のすべてのプロセスが終了します。

```
 % Reset_local
```

ここでももう一度聞かれますので、終りたい時は yes と打って下さい。

4 回路

ロジック図を図 2 に示します。今回の実験における trigger は、回路に示した通り PWO クリスタル自身から作ります。PWO からのシグナルは、DIVIDER で 2 つに分けられ 1

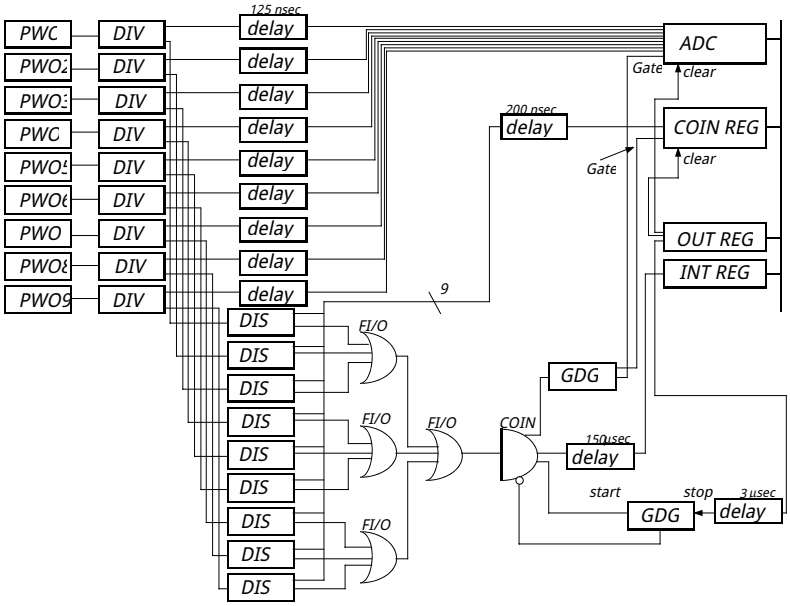


図 2: ガス制動放射実験における LOGIC

つは ADC で取り込み、もう一方はそれぞれ DISCREMINATOR で threshold 以下のイベントを切り捨て、全体の OR をとり Trigger が作られます。

Trigger が入ると、Data を取り込んでる間に次の event がこないように、Trigger で latch をかけて veto します。ADC のゲートが閉じると、ADC はデータの変換を開始します。ADC の変換が終了した後に INTERRUPT REGISTER に Trigger が入り、割り込みが発生しデータの取り込みをはじめます。Data をとりこんだあとで、OUTPUT REGISTER から各モジュールに Clear シグナルが送られ、次の取り込み準備をします。準備が整うと OUTPUT REGISTER によって latch が外され、次の event を待ちます。これらの事を Data 収集の間繰り返されます。

5 Data Format

今回のシステムでは、データとして interrupt register、coincidence register 、ADC 16ch をとっています。このシステムのデータ フォーマットは、図 3 に示す通り、Event データ以外にいつも 6word のヘッダが入ります。

Buffer length はこのイベントの word 数で、Record type は以下の通りです。

```
<Record type>
Record type = 0 : Data
```

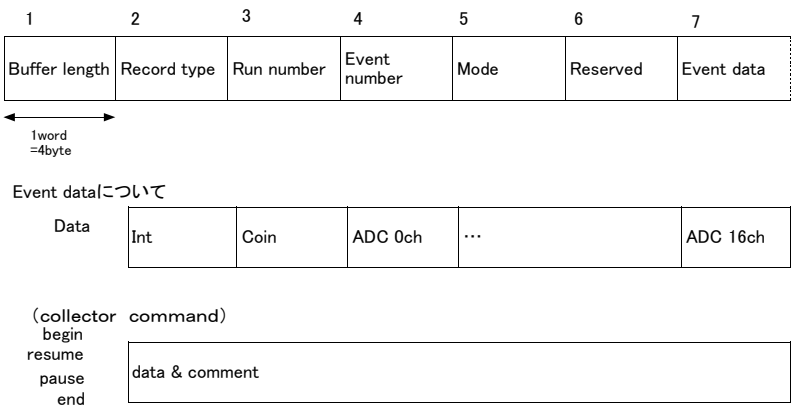


図 3: Event Data Record

```
Record type = 1 : Begin
Record type = 2 : Pause
Record type = 3 : Resume
Record type = 4 : End
```

以上で説明を終ります。