

sPHENIX INTT 日本語ミーティング、2020/10/09

外部トリガー DAQ の進捗状況

RBRC：糠塚 元気 (RBRC)

奈良女：蜂谷 崇，柴田 実香，森田 美羽，

高濱 瑠菜，並本 ゆみか，西森 早紀子

やること

INTT イベントの
時刻を決める

- INTT テストベンチ用 PC へ移植
- INTT DAQ との連動
- オフライン解析で INTT のイベントと外部トリガーのイベントが同期できるか確認
- INTT とトリガーシンチレータで宇宙線の測定, 解析を試みる
- INTT DAQ の GUI 改良

INTT の DAQ

navis_gui.py で GUI を起動

FFR	Enable RO	Latch FPGA	Core Reset	Start DAQ	Check GLINK
Init	Disable RO	Calib	JTAG Sync	Stop DAQ	Check FEM
FO Sync	Set L1 Delay	Delay 59	BCO Start	Global Start	Self Trig
FPGA RST	Er. EEPROM	Write Page	Read Page	Write All	Cosmic Start

FPHX TestStand DAQ

File

Module 15 | Module 0 | Module 1

Reg	Desc	To Chip	From Chip	Chip Command
*	Wild	0		Read Write Set Reset Default
1	Mask	0		Read Write Set Reset Default
2	Dig Ctrl	5		Read Write Set Reset Default
3	Vref	1		Read Write Set Reset Default
4	DAC0	10		Read Write Set Reset Default
5	DAC1	23		Read Write Set Reset Default
6	DAC2	48		Read Write Set Reset Default
7	DAC3	98		Read Write Set Reset Default
8	DAC4	148		Read Write Set Reset Default
9	DAC5	172		Read Write Set Reset Default
10	DAC6	223		Read Write Set Reset Default
11	DAC7	248		Read Write Set Reset Default
12	N1Sel <3:0>	6		Read Write Set Reset Default
	N2Sel <7:4>	4		
13	FB1Sel <3:0>	4		Read Write Set Reset Default
	.eakSel <7:4>	0		
14	P3Sel <1:0>	0		Read Write Set Reset Default
	P2Sel <7:4>	4		
15	GSel <2:0>	2		Read Write Set Reset Default
	BWSel <7:3>	8		
16	P1Sel <2:0>	5		Read Write Set Reset Default
	InjSel <5:3>	0		
17	LVDS Current	3		Read Write Set Reset Default
18	Resets	n/a		Read Write Set Reset Default

Chip Control

Display/Modify Configuration for Chip ID 21 Side 15

Channel Mask [Red = Off, Green = On]

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127

Mask All Unmask All Toggle All Send

Chip Side Enable

0	15	0	1	8	15	0	1	16	15	0	1	24	15	0	1
1	15	0	1	9	15	0	1	17	15	0	1	25	15	0	1
2	15	0	1	10	15	0	1	18	15	0	1	26	15	0	1
3	15	0	1	11	15	0	1	19	15	0	1	27	15	0	1
4	15	0	1	12	15	0	1	20	15	0	1	28	15	0	1
5	15	0	1	13	15	0	1	21	15	0	1	29	15	0	1
6	15	0	1	14	15	0	1	22	15	0	1	30	15	0	1
7	15	0	1	15	15	0	1	23	15	0	1	31	15	0	1

TestStand

Spartan3 ROC ROC+FEM FEM Addr 15 DB Access On Off

Global Chip/DAQ Operations

FFR	Enable RO	Latch FPGA	Core Reset	Start DAQ	Check GLINK
Init	Disable RO	Calib	JTAG Sync	Stop DAQ	Check FEM
FO Sync	Set L1 Delay	Delay 59	BCO Start	Global Start	Self Trig
FPGA RST	Er. EEPROM	Write Page	Read Page	Write All	Cosmic Start

DAQ Configuration

DAQ Program C:/Users/sphenix/Documents/INTT_testba Browse

NI DAQ Sample Rate (MHz) 5

Num of events (0=inf) 0

Duration HH:MM:SS (0:00:00=inf) HH:MM:SS

Print Output Print Off

FPHX version (for Print) 2

Run Number

Filename

Beam Species None

Beam Energy 0

Pulsar Configuration

Pulse amplitude (10 bits max) 255 Config Amp Pulse

Num of Pulses 1 Pulse Train

BCOs between pulses 1023 Wedge 0 Module 0 Set Module

Module Enable

Module 15 On Off Both Side 0 Side 1

Module 0 On Off Both Side 0 Side 1

Module 1 On Off Both Side 0 Side 1

Manual Packet Send

Packet file to send Browse Send

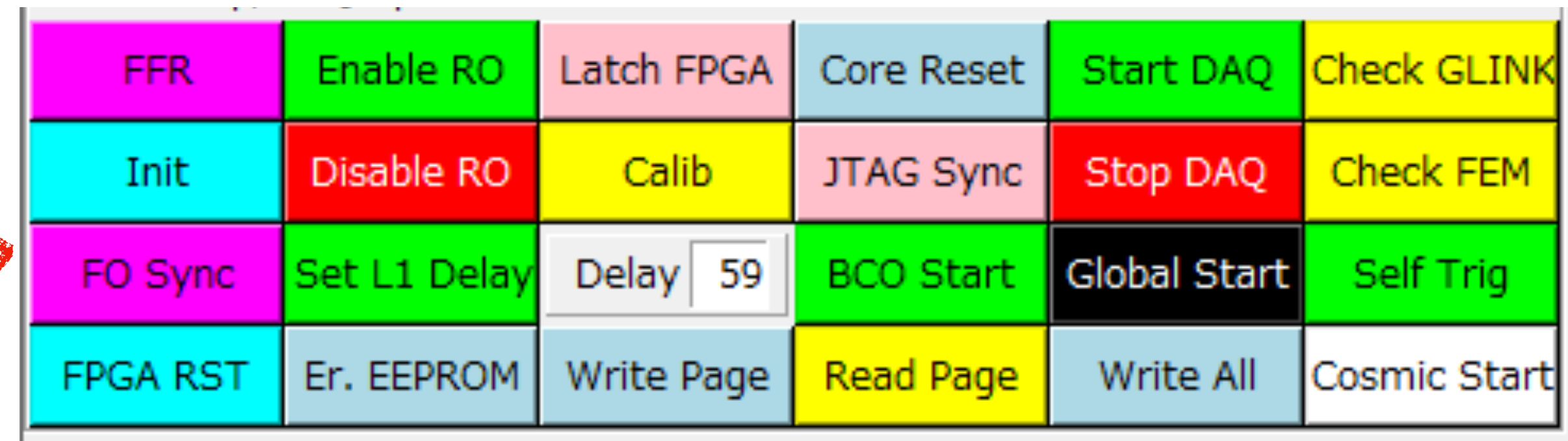
Communications

USB None Ethernet IP Addr 192.168.60.: Port 9900

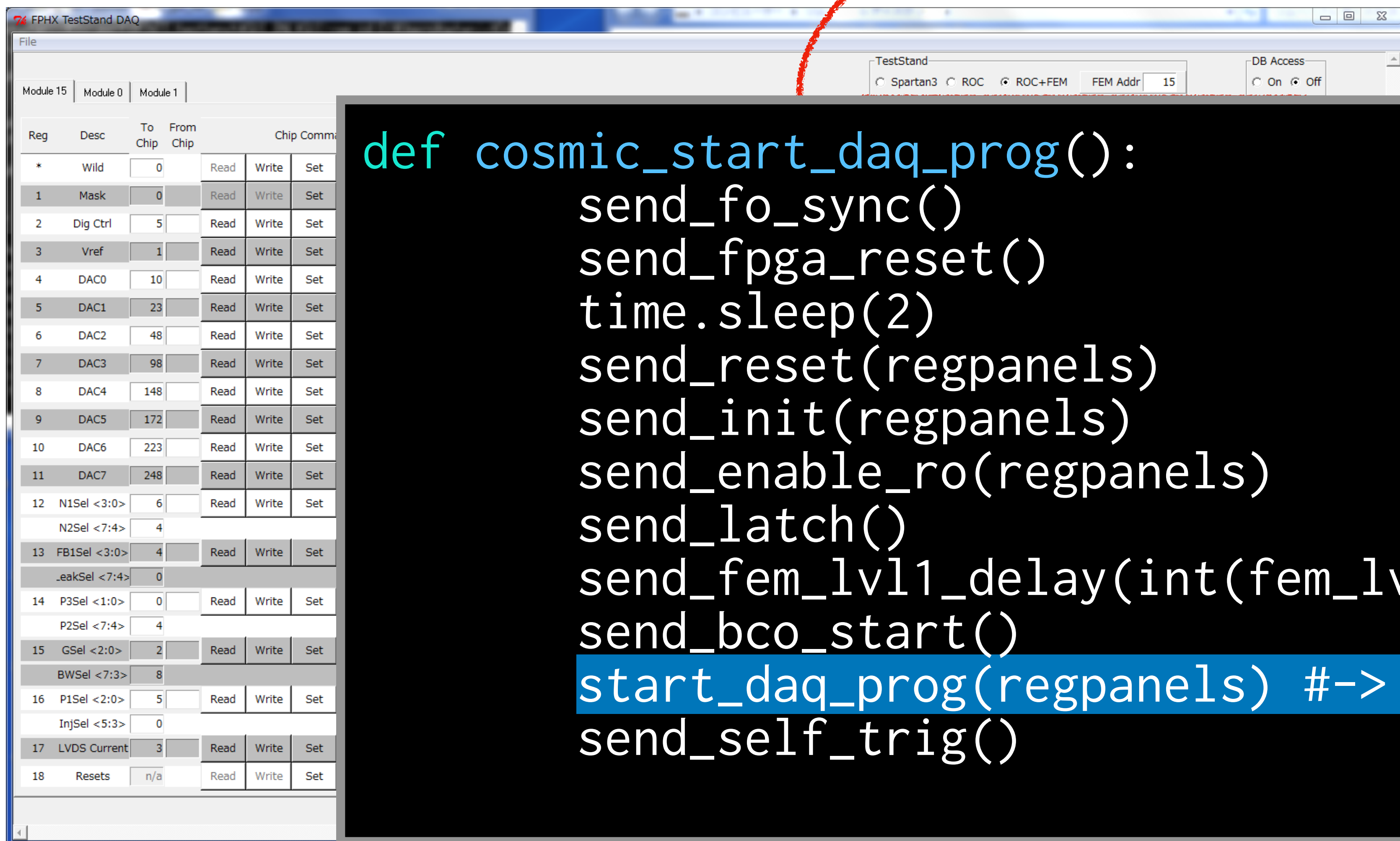
Baud Rate 115200

INTT の DAQ

navis_gui.py で GUI を起動



FFR	Enable RO	Latch FPGA	Core Reset	Start DAQ	Check GLINK
Init	Disable RO	Calib	JTAG Sync	Stop DAQ	Check FEM
FO Sync	Set L1 Delay	Delay 59	BCO Start	Global Start	Self Trig
FPGA RST	Er. EEPROM	Write Page	Read Page	Write All	Cosmic Start



Reg	Desc	To Chip	From Chip	Chip Comm
*	Wild	0		Read Write Set
1	Mask	0		Read Write Set
2	Dig Ctrl	5		Read Write Set
3	Vref	1		Read Write Set
4	DAC0	10		Read Write Set
5	DAC1	23		Read Write Set
6	DAC2	48		Read Write Set
7	DAC3	98		Read Write Set
8	DAC4	148		Read Write Set
9	DAC5	172		Read Write Set
10	DAC6	223		Read Write Set
11	DAC7	248		Read Write Set
12	N1Sel <3:0>	6		Read Write Set
	N2Sel <7:4>	4		
13	FB1Sel <3:0>	4		Read Write Set
	.eakSel <7:4>	0		
14	P3Sel <1:0>	0		Read Write Set
	P2Sel <7:4>	4		
15	GSel <2:0>	2		Read Write Set
	BWSel <7:3>	8		
16	P1Sel <2:0>	5		Read Write Set
	InjSel <5:3>	0		
17	LVDS Current	3		Read Write Set
18	Resets	n/a		Read Write Set

```
def cosmic_start_daq_prog():  
    send_fo_sync()  
    send_fpga_reset()  
    time.sleep(2)  
    send_reset(regpanels)  
    send_init(regpanels)  
    send_enable_ro(regpanels)  
    send_latch()  
    send_fem_lv11_delay(int(fem_lv11_delay_var.get()))  
    send_bco_start()  
    start_daq_prog(regpanels) #-> read_DAQ.exe を実行  
    send_self_trig()
```


read_DAQ.c, AcquireData 関数

終了ボタンを押すまでループ

AcquireData 関数

read_DDAQ.c, AcquireData 関数

終了ボタンを押すまでループ

NI のボードからデータ読み出し

AcquireData 関数

read_DDAQ.c, AcquireData 関数

終了ボタンを押すまでループ

NI のボードからデータ読み出し

バッファにデータがある程度たまったらファイルにデータ書き出し

AcquireData 関数

read_DDAQ.c, AcquireData 関数

終了ボタンを押すまでループ

NI のボードからデータ読み出し

バッファにデータがある程度たまったらファイルにデータ書き出し

ループ終了時もファイルにデータ書き出し

AcquireData 関数

書き出されるデータ

```
$ head data/INTT_test_bench/to_be_processed/nwu_fphx_raw_20201002-2032_0.dat
e

d
{O{i{o{W#{g{_{o{^{{v{g{{{g{{{~{${{{6{^{{f{{{u@}}},}{{G{{G{
                                     }}} }}G} }}G}}O}!!}!}W}
}W}~_#}$}g}g}x}c}'}}o}/}}o3}/#}8}w}7}Z}}?!~\}+}?I}}/~G})}M}G'}S}0%}+}1}03}}I}W}}{W'}'}}_}%}9}}_u}"}}g}$}oG}g#}}oI}'  }}
w!}'C},}w)}~/+}6}9}/U}<}I}7}b}7}7}1}[?W}}e}?e}/~o}G)}=}M}GC}?S}05}G}}0A}A}}W+}K}}W5}A}}_}K}oU}_~,}oY}g)}0}wC}g-}}wU}'S}
S}'U}<}Y}/e}L}}/{j}}7}}u}7}}}?u}}c}?{G}m}G}_c}}Ga}U}}OW}}]0c}S}WA}[o]}WC}0}og}_}Q}w[~_}@}wc}g1}F}}]g5}g}'k}}'u}'}}/~
v}}/}}6}}6}}}?}}>}e}}Gm}m}Gq}g}Oi}k}oi}Om}{}oo}WE}}wi}WI}[w}_}m}}_}H}}g=}V}}g0}}'{}' }|}}/}}}/}}5}}6}}}>}}Gs}}oq}
G}q}o}Os}}w}Ow}}w}Wc}}Wi}}}_}}_}X}gQ}^}}gW}}'}}' }}~/}}}/}}6}}4}}}_o}}o}G}}~u@~~,~
                                     ~ ~ G ~ ~ G ~ ~ ~ 0 ~ ! ~ ! ~ W ~
~W~_#~$~g~g~x~c~'~~o~/~~o3~/#~8~w~7~Z~~?!~\~+~?I~~/~G~)~M~G'~S~0%~+~1~03~~I~W~~{~W'~'~~_}%~9~~_u~~"~~g~$~oG~g#~~oI~'  ~w!
~'C~,~w)~/+~6~9~/U~<~I~7~b~7~7~1~[~?W~~e~?e~/~o~G)~=~M~GC~?~S~05~G~~0A~A~~W+~K~~W5~A~~_~K~oU~_~,~oY~g)~0~wC~g-
~~wU}'S~~S}'U~<~Y~/e~L~~/{~j~~7~~u~7~~~?u~~c~?
{~G~m~G~_~c~~Ga~U~~OW~]~~0c~S~~WA~[~o]~WC~0~og~_~Q~w[~_~@~wc~g1~F~]~g5~~g~'k~~~'u~`~~/~v~~/~~6~~6~~~?~~~>~e~~Gm~m~
~Gq~g~
```

そのままだと読めないなので、人が読めるように書き出すと・・・

書き出されるデータ

```
$ less -S nwu_fphx_raw_20201002-2032_0_decoded.dat
  4      101      11      1000
 12      100       0      8181      0      0      0      17104896 1647318794 4175408276
8177     102     65537 2147483650 123456 2046878604 2046872990 2046879646 2046879047 2046883750
8176     102 238354433 2110324738 123456 4816896 4836176 238485505 2147287042 123456
8174     102 429785089 3651993602 123456 2981888 3011928 429916161 3673948162 123456
8177     102 621150209 553910274 123456 294912 283976 621215745 560791554 123456
8178     102 816381953 3108175874 123456 4423680 4446000 816513025 3115974658 123456
8178     102 1012006913 782893058 123456 2785280 2795864 1012137985 794755074 123456
```

書き出されるデータ

```
$ less -S nwu_fphx_raw_20201002-2032_0_decoded.dat
```

4	101	11	1000								
12	100	0	8181	0	0	0	0	17104896	1647318794	4175408276	
8177	102	65537	2147483650	123456	2046878604	2046872990	2046879646	2046879047	2046883750		
8176	102	238354433	2110324738	123456	4816896	4836176	238485505	2147287042		123456	
8174	102	429785089	3651993602	123456	2981888	3011928	429916161	3673948162		123456	
8177	102	621150209	553910274	123456	294912	283976	621215745	560791554		123456	
8178	102	816381953	3108175874	123456	4423680	4446000	816513025	3115974658		123456	
8178	102	1012006913	782893058	123456	2785280	2795864	1012137985	794755074		123456	

バッファーにたまったイベント
↓

書き出されるデータ

```
$ less -S nwu_fphx_raw_20201002-2032_0_decoded.dat
```

4	101	11	1000							
12	100	0	8181	0	0	0	17104896	1647318794	4175408276	
8177	102	65537	2147483650	123456	2046878604	2046872990	2046879646	2046879047	2046883750	
8176	102	238354433	2110324738	123456	4816896	4836176	238485505	2147287042	123456	
8174	102	429785089	3651993602	123456	2981888	3011928	429916161	3673948162	123456	
8177	102	621150209	553910274	123456	294912	283976	621215745	560791554	123456	
8178	102	816381953	3108175874	123456	4423680	4446000	816513025	3115974658	123456	
8178	102	1012006913	782893058	123456	2785280	2795864	1012137985	794755074	123456	

バッファーにたまったイベント



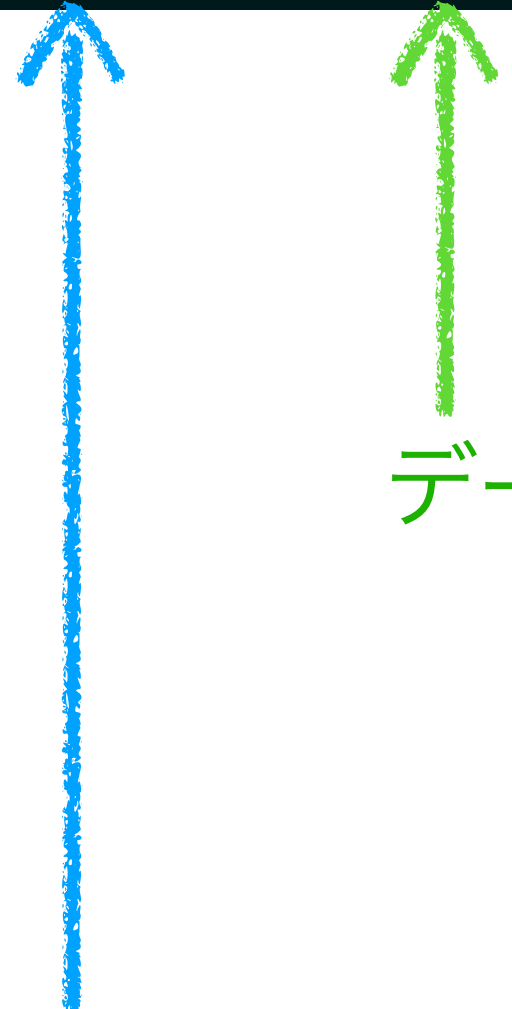
データグループにあるデータの数 (N)

書き出されるデータ

```
$ less -S nwu_fphx_raw_20201002-2032_0_decoded.dat
```

4	101	11	1000								
12	100	0	8181	0	0	0	0	17104896	1647318794	4175408276	
8177	102	65537	2147483650	123456	2046878604	2046872990	2046879646	2046879047	2046883750		
8176	102	238354433	2110324738	123456	4816896	4836176	238485505	2147287042		123456	
8174	102	429785089	3651993602	123456	2981888	3011928	429916161	3673948162		123456	
8177	102	621150209	553910274	123456	294912	283976	621215745	560791554		123456	
8178	102	816381953	3108175874	123456	4423680	4446000	816513025	3115974658		123456	
8178	102	1012006913	782893058	123456	2785280	2795864	1012137985	794755074		123456	

バッファーにたまったイベント



データのタイプ

100 設定に関する出力

101 タイムスタンプ

102 INTT のデータ

データグループにあるデータの数 (N)

書き出されるデータ

```
$ less -S nwu_fphx_raw_20201002-2032_0_decoded.dat
```

4	101	11	1000							
12	100	0	8181	0	0	0	0	17104896	1647318794	4175408276
8177	102	65537	2147483650	123456	2046878604	2046872990	2046879646	2046879047	2046883750	
8176	102	238354433	2110324738	123456	4816896	4836176	238485505	2147287042	123456	
8174	102	429785089	3651993602	123456	2981888	3011928	429916161	3673948162	123456	
8177	102	621150209	553910274	123456	294912	283976	621215745	560791554	123456	
8178	102	816381953	3108175874	123456	4423680	4446000	816513025	3115974658	123456	
8178	102	012006913	782893058	123456	2785280	2795864	1012137985	794755074	123456	

バッファーにたまったイベント



INTT のイベント

1 ~ N-1 番目 : INTT のイベント (32bit)

N 番目 : チェックサム

データのタイプ

100 設定に関する出力

101 タイムスタンプ

102 INTT のデータ

データグループにあるデータの数 (N)

書き出されるデータ

```
$ less -S nwu_fphx_raw_20201002-2032_0_decoded.dat
```

4	101	11	1000							
12	100	0	8181	0	0	0	0	17104896	1647318794	4175408276
8177	102	65537	2147483650	123456	2046878604	2046872990	2046879646	2046879047	2046883750	
8176	102	238354433	2110324738	123456	4816896	4836176	238485505	2147287042	123456	
8174	102	429785089	3651993602	123456	2981888	3011928	429916161	3673948162	123456	
8177	102	621150209	553910274	123456	294912	283976	621215745	560791554	123456	
8178	102	816381953	3108175874	123456	4423680	4446000	816513025	3115974658	123456	
8178	102	012006913	782893058	123456	2785280	2795864	1012137985	794755074	123456	

バッファーにたまったイベント



INTT のイベント

1 ~ N-1 番目 : INTT のイベント (32bit)

N 番目 : チェックサム

データのタイプ

100 設定に関する出力

101 タイムスタンプ

102 INTT のデータ

データグループにあるデータの数 (N)



```
TTree *tree = new TTree("tree", "chip info");
tree->Branch("adc", &adc, "adc/I");
tree->Branch("ampl", &ampl, "ampl/I");
tree->Branch("chip_id", &chip_id, "chip_id/I");
tree->Branch("fpga_id", &fpga_id, "fpga_id/I");
tree->Branch("module", &module, "module/I");
tree->Branch("chan_id", &chan_id, "chan_id/I");
tree->Branch("fem_id", &fem_id, "fem_id/I");
tree->Branch("bco", &bco, "bco/I");
tree->Branch("bco_full", &bco_full, "bco_full/I");
tree->Branch("event", &ievent, "event/I");
```

INTT イベントに時刻を割り当てる

終了ボタンを押すまでループ

NI のボードからデータ読み出し

```
$ less -S nwu_fphx_raw_20201002-2032_0_decoded.dat
```

4	101	11	1000		
12	100	0	8181	0	0
8177	102	65537	2147483650	123456	2046878604
8176	102	238354433	2110324738	123456	4816896
8174	102	429785089	3651993602	123456	2981888
8177	102	621150209	553910274	123456	294912
8178	102	816381953	3108175874	123456	4423680
8178	102	1012006913	782893058	123456	2785280

イベントを取得する
ループの中で時刻を得て、
データに挟めばいいのでは？

INTT イベントに時刻を割り当てる

終了ボタンを押すまでループ

NI のボードからデータ読み出し

```
$ less -S nwu_fphx_raw_20201002-2032_0_decoded.dat
```

4	101	11	1000		
12	100	0	8181	0	0
8177	102	65537	2147483650	123456	2046878604
8176	102	238354433	2110324738	123456	4816896
8174	102	429785089	3651993602	123456	2981888
8177	102	621150209	553910274	123456	294912
8178	102	816381953	3108175874	123456	4423680
8178	102	1012006913	782893058	123456	2785280

~~イベントを取得する
ループの中で時刻を得て、
データに挟めばいいのでは？~~

INTT イベントに時刻を割り当てる

終了ボタンを押すまでループ

NI のボードからデータ読み出し

```
$ less -S nwu_fphx_raw_20201002-2032_0_decoded.dat
```

4	101	11	1000		
12	100	0	8181	0	0
8177	102	65537	2147483650	123456	2046878604
8176	102	238354433	2110324738	123456	4816896
8174	102	429785089	3651993602	123456	2981888
8177	102	621150209	553910274	123456	294912
8178	102	816381953	3108175874	123456	4423680
8178	102	1012006913	782893058	123456	2785280

~~イベントを取得する
ループの中で時刻を得て、
データに挟めばいいのでは？~~

この関数で大量のイベントを
(ソースコード的に) 同時に
取得しており、イベント発生時に
何かをすることができない

```
DAQmxErrCheck(DAQmxReadDigitalU32(taskHandle, -1, 10.0, DAQmx_Val_GroupByChannel,  
data, 1000000, &numRead, NULL));
```



INTT イベントに時刻を割り当てる：苦肉の策

```
$ less -S nwu_fphx_raw_20201002-2032_0_decoded.dat
  4      101      11      1000
 12      100       0      8181      1647318794
8177     102    65537  2147483650  2046879047
8176     102  238354433  2110324738  2147287042      + チェックサム
8174     102  429785089  3651993602  . . . . . 3673948162
8177     102  621150209   553910274  560791554
8178     102  816381953   3108175874  3115974658
8178     102 1012006913   782893058   794755074
```

データグループ最初の
イベント E_0 の時刻 t_0 は
取得できる

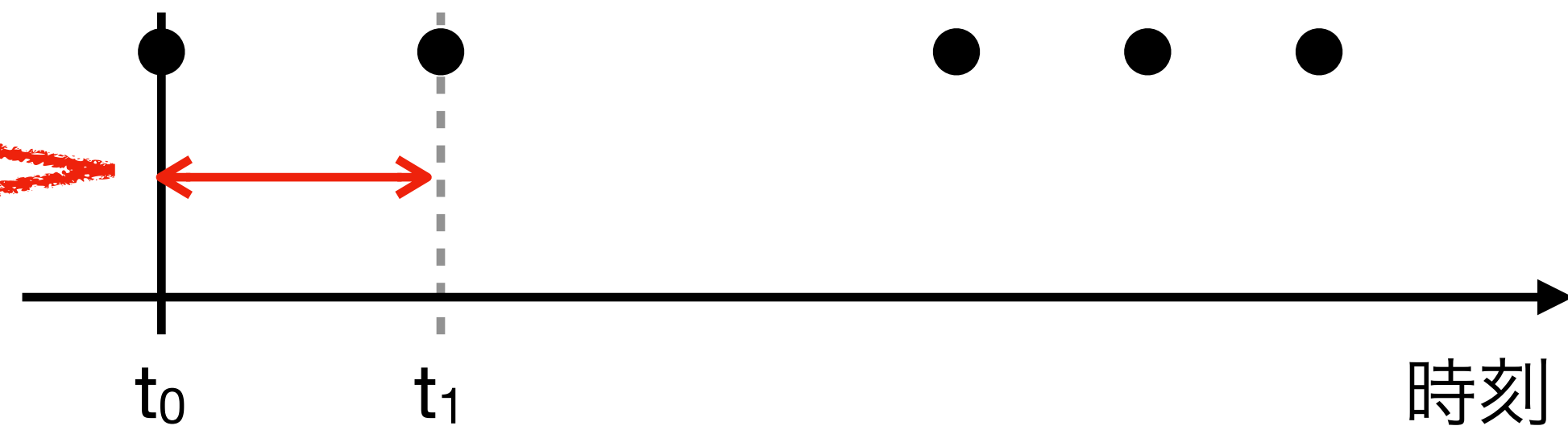
INTT イベントに時刻を割り当てる：苦肉の策

```
$ less -S nwu_fphx_raw_20201002-2032_0_decoded.dat
  4      101      11      1000
 12      100       0      8181      1647318794
8177     102    65537  2147483650  2046879047
8176     102  238354433  2110324738  2147287042
8174     102  429785089  3651993602  3673948162
8177     102  621150209   553910274  560791554
8178     102  816381953  3108175874  3115974658
8178     102 1012006913   782893058  794755074
```

+ チェックサム

データグループ最初の
イベント E_0 の時刻 t_0 は
取得できる

bco カウントか
ら計算できる？



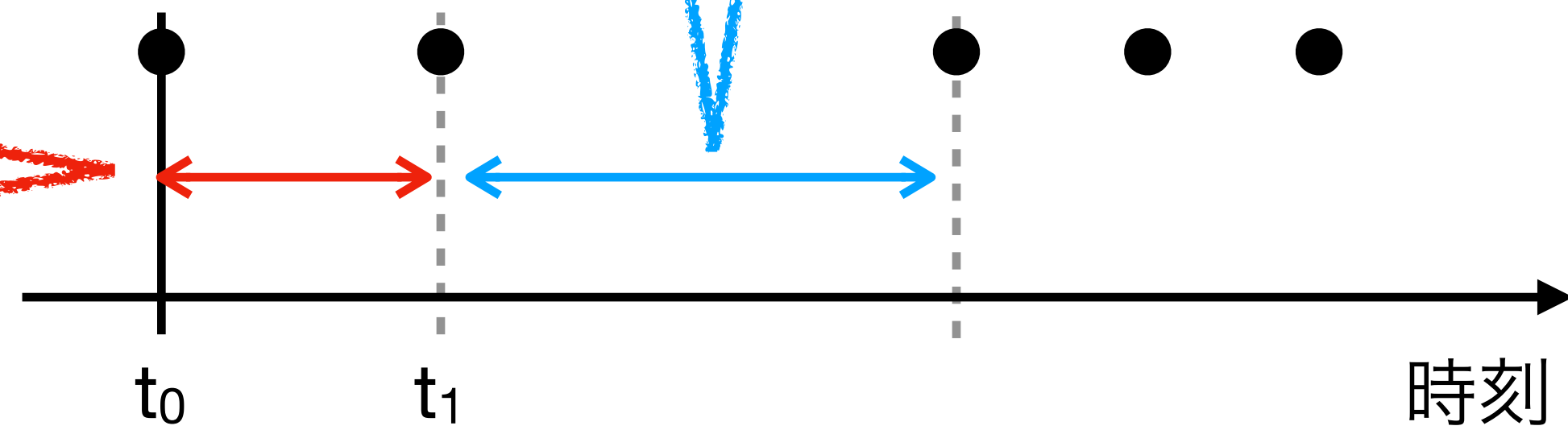
INTT イベントに時刻を割り当てる：苦肉の策

```
$ less -S nwu_fphx_raw_20201002-2032_0_decoded.dat
  4      101      11      1000
 12      100       0      8181      1647318794
8177     102     65537  2147483650  2046879047
8176     102  238354433  2110324738  2147287042      + チェックサム
8174     102  429785089  3651993602  3673948162
8177     102  621150209   553910274  560791554
8178     102  816381953  3108175874  3115974658
8178     102 1012006913   782893058  794755074
```

データグループ最初のイベント E_0 の時刻 t_0 は取得できる

イベント間隔が長すぎると bco_full が何周したかわからない

bco カウントから計算できる？



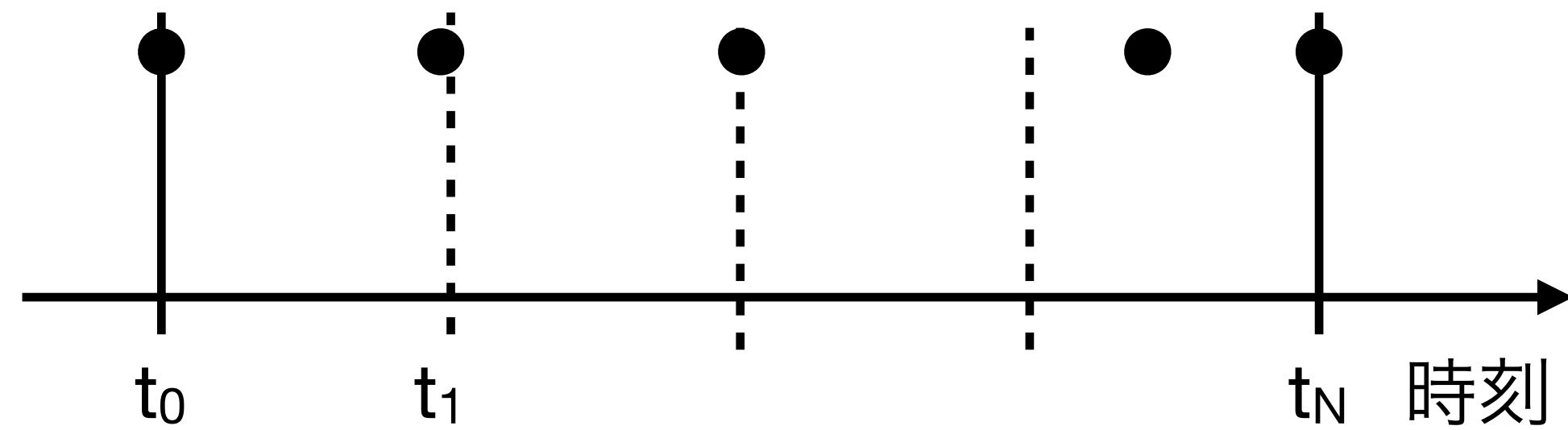
bco カウント : 100 ns
 bco : 128 カウントで 1 週 \rightarrow 13 μ s
 bco_full : 2^{16} カウントで 1 週 \rightarrow 6.6 ms
 \therefore 全イベント間隔が 12 ms 程度以下ならイベントの時刻を計算できる

INTT イベントに時刻を割り当てる：苦肉の策

```
$ less -S nwu_fphx_raw_20201002-2032_0_decoded.dat
  4      101      11      1000
 12      100       0      8181      1647318794
8177     102    65537  2147483650      2046879047
8176     102  238354433  2110324738      2147287042      + チェックサム
8174     102  429785089  3651993602      .....      3673948162
8177     102  621150209   553910274      560791554
8178     102  816381953   3108175874      3115974658
8178     102 1012006913    782893058      794755074
```

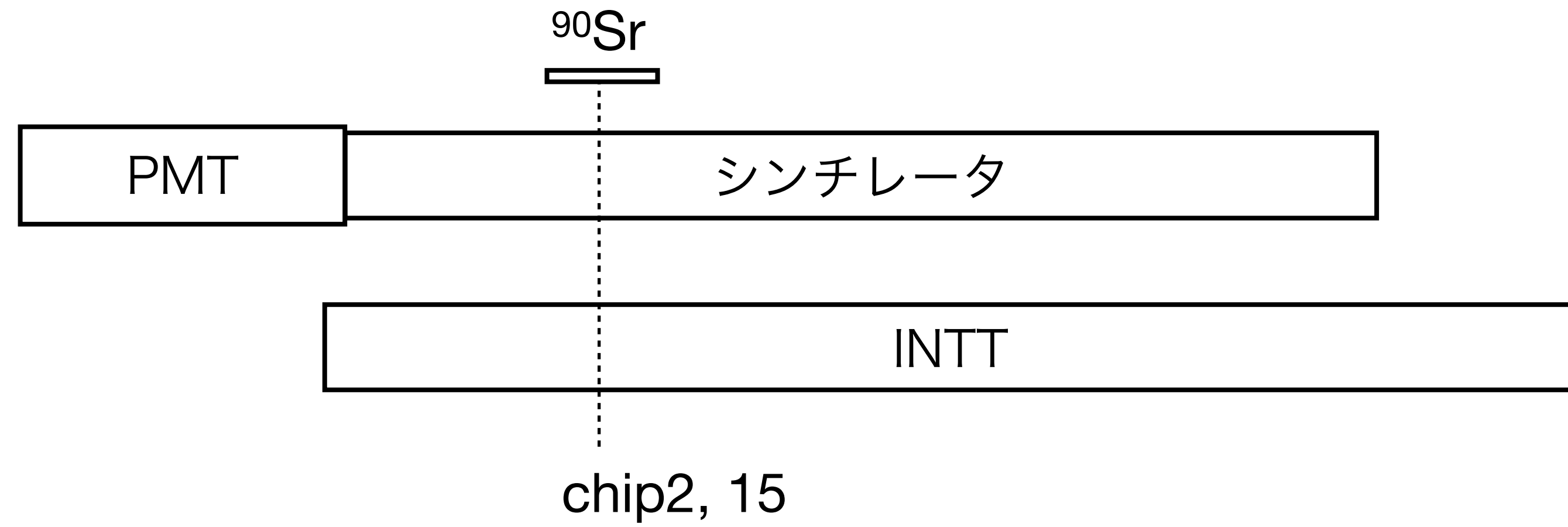
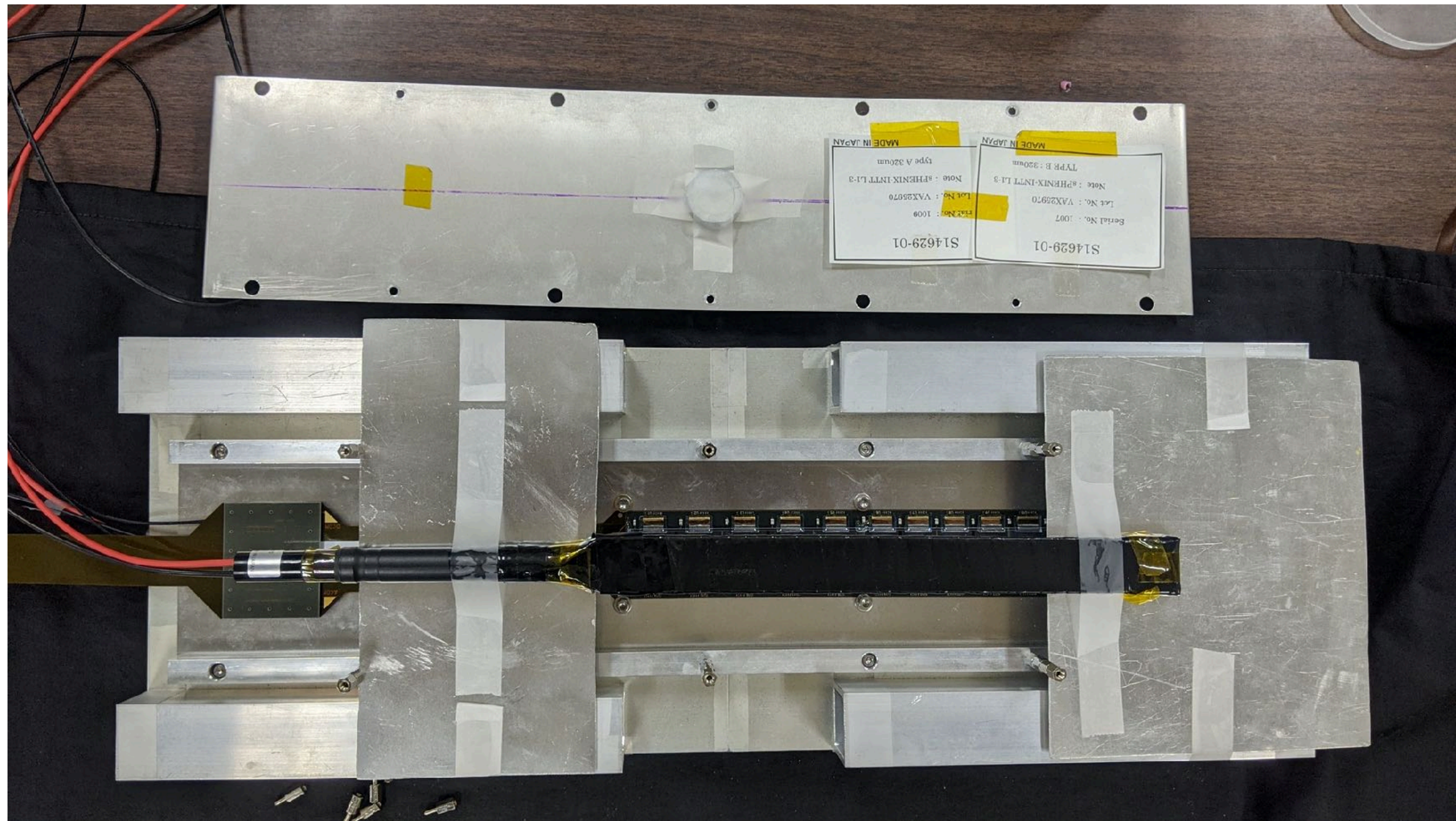
データグループ最初の
イベント E_0 の時刻 t_0 は
取得できる

データグループ最後の
イベント E_N の時刻 t_N は
取得できる (見込み)

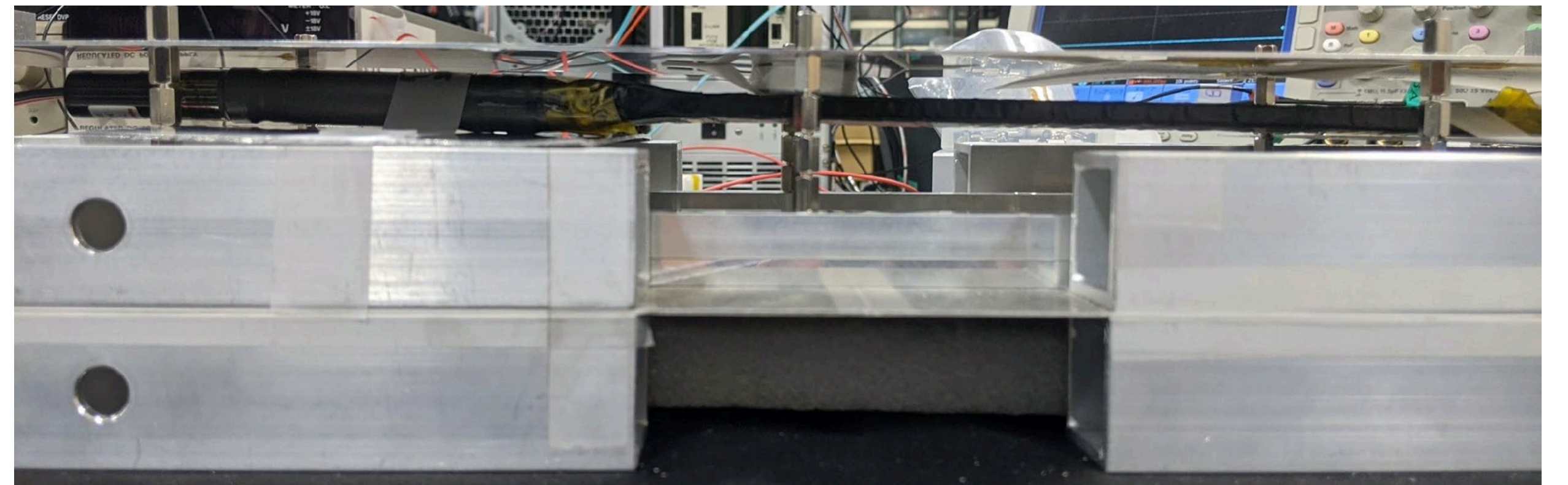


t_0 から t_N の間でイベントが等間隔に
分布しているなら、計算できる・・・
宇宙線測定なら適用できる???

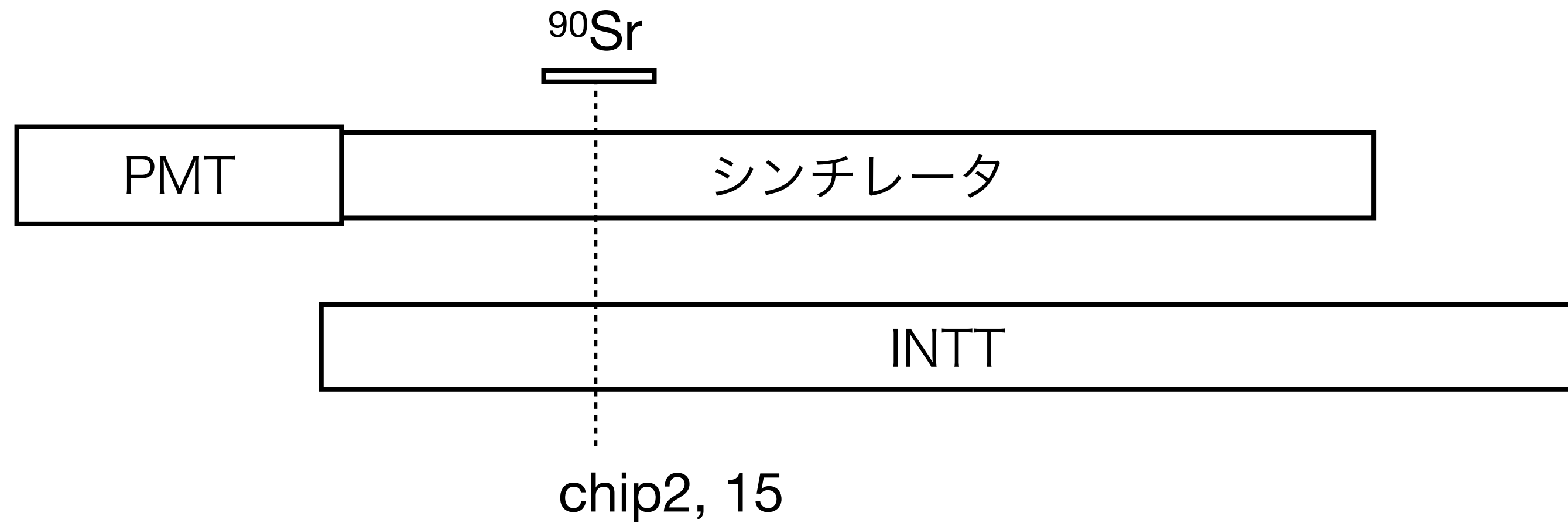
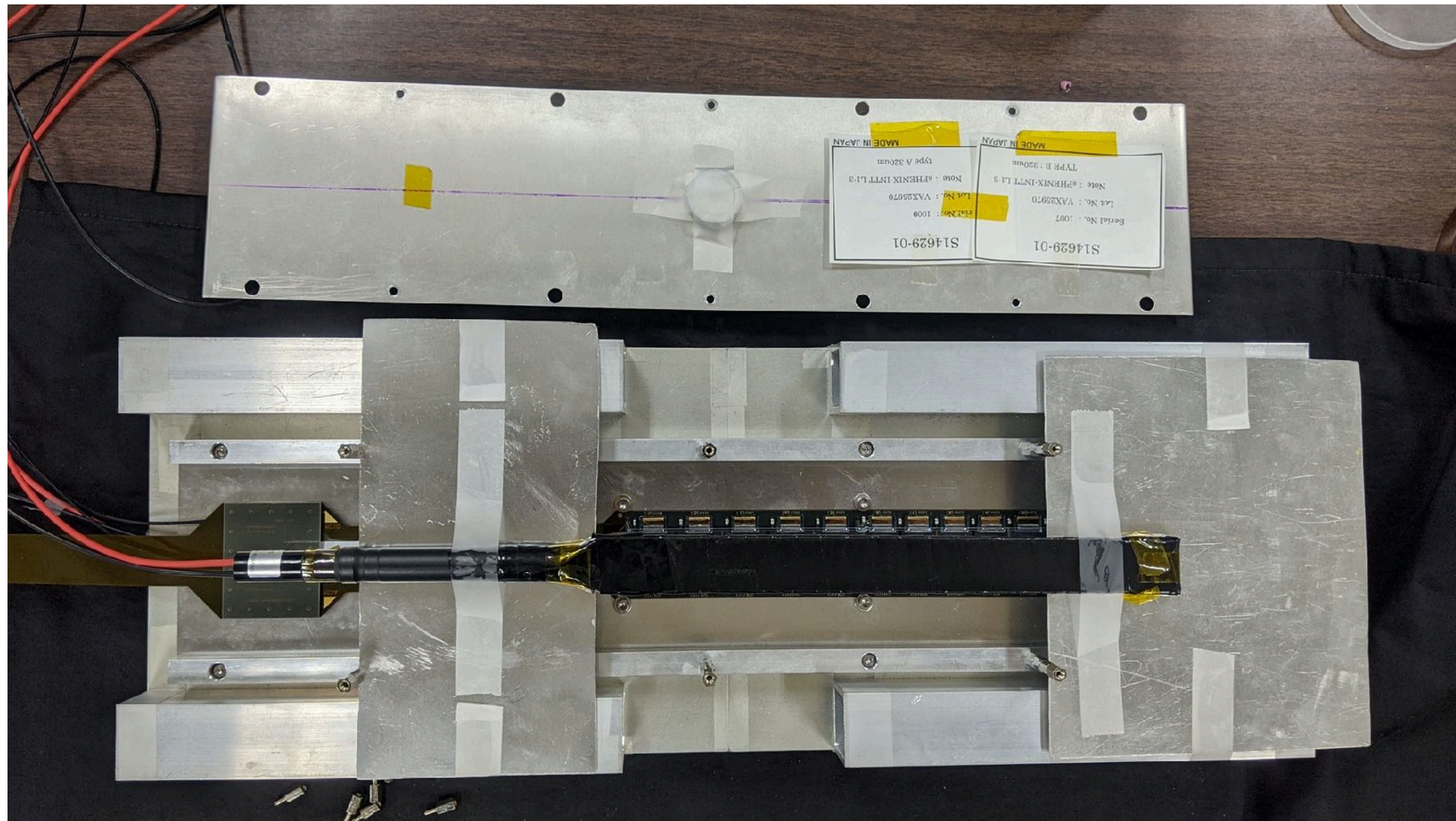
INTT データとトリガーデータを取ってみる



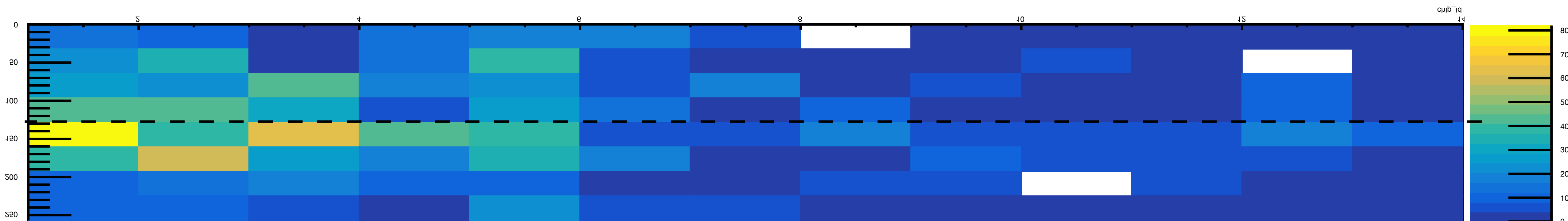
セットアップを横から見た図



INTT データとトリガーデータを取ってみる

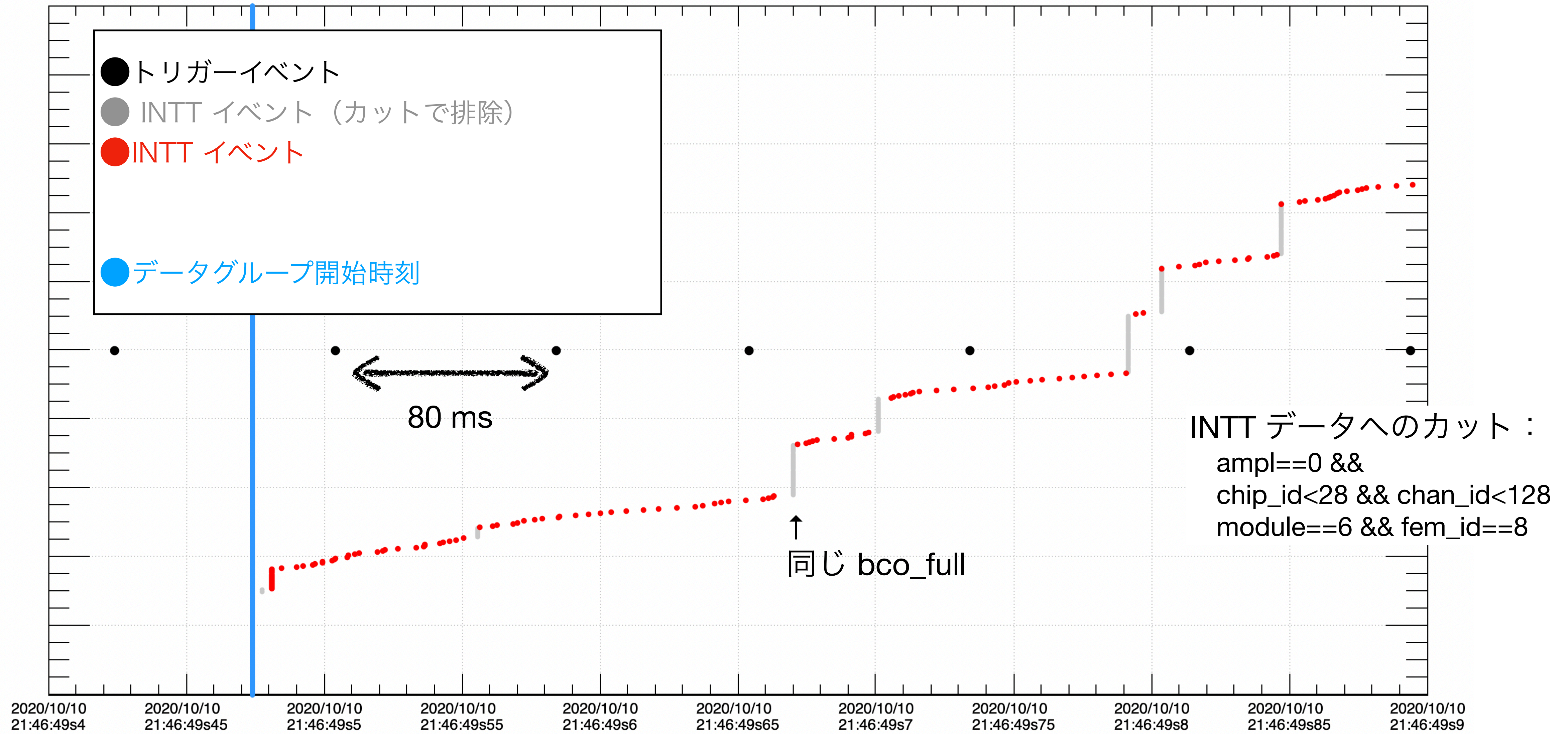


カット : `ampl==0 &&`
`chip_id<28 && chan_id<128`
`module==6 && fem_id==8`



INTT のヒットマップ

INTT データとトリガーデータを取ってみる

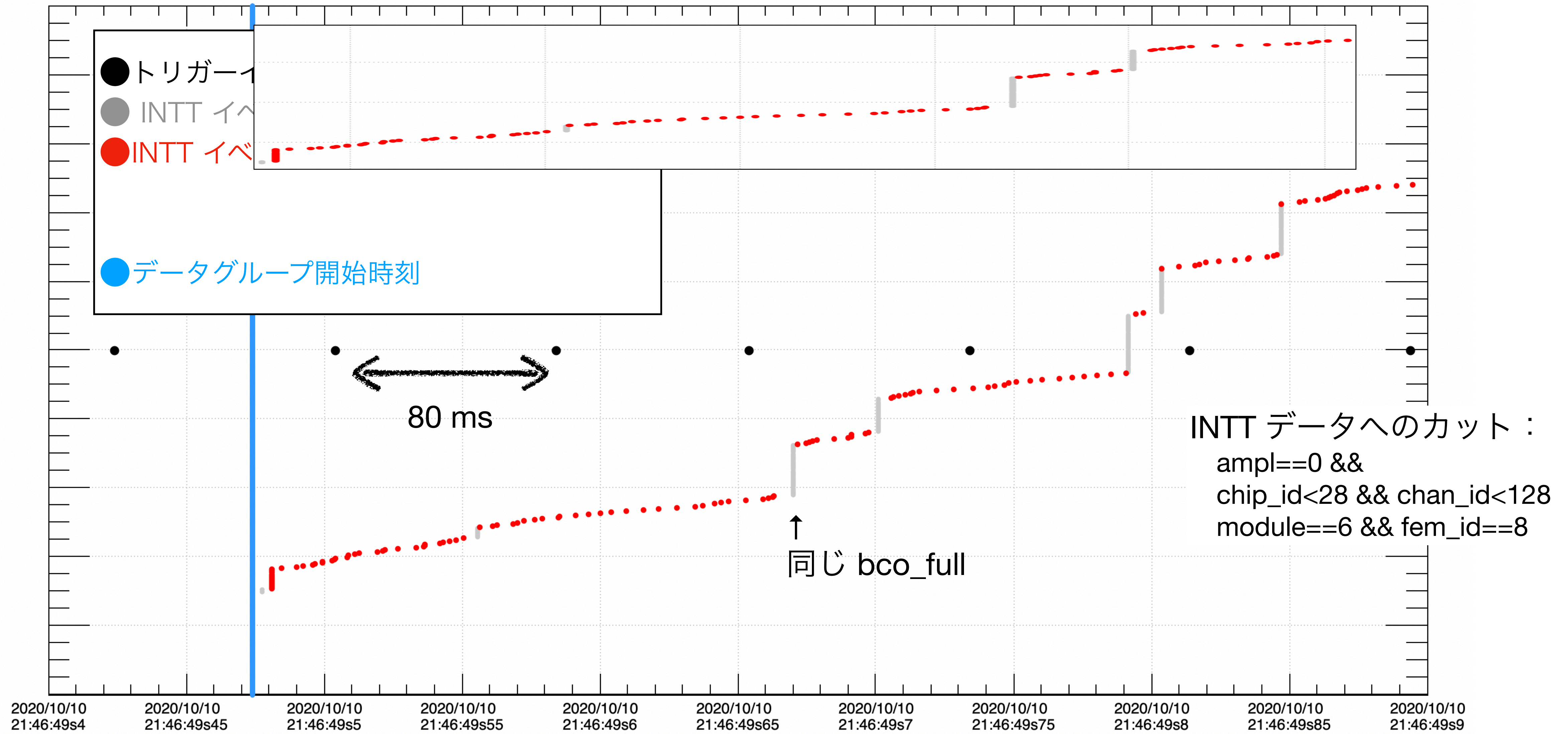


トリガーイベントの時間間隔 80 ms を bco_full のカウントで埋めるのは難しい・・・

INTT イベントの時間軸に対してトリガーイベントがシフトしているか未検証

トリガー 1 イベントだけ収集して様子を見てみたい

INTT データとトリガーデータを取ってみる

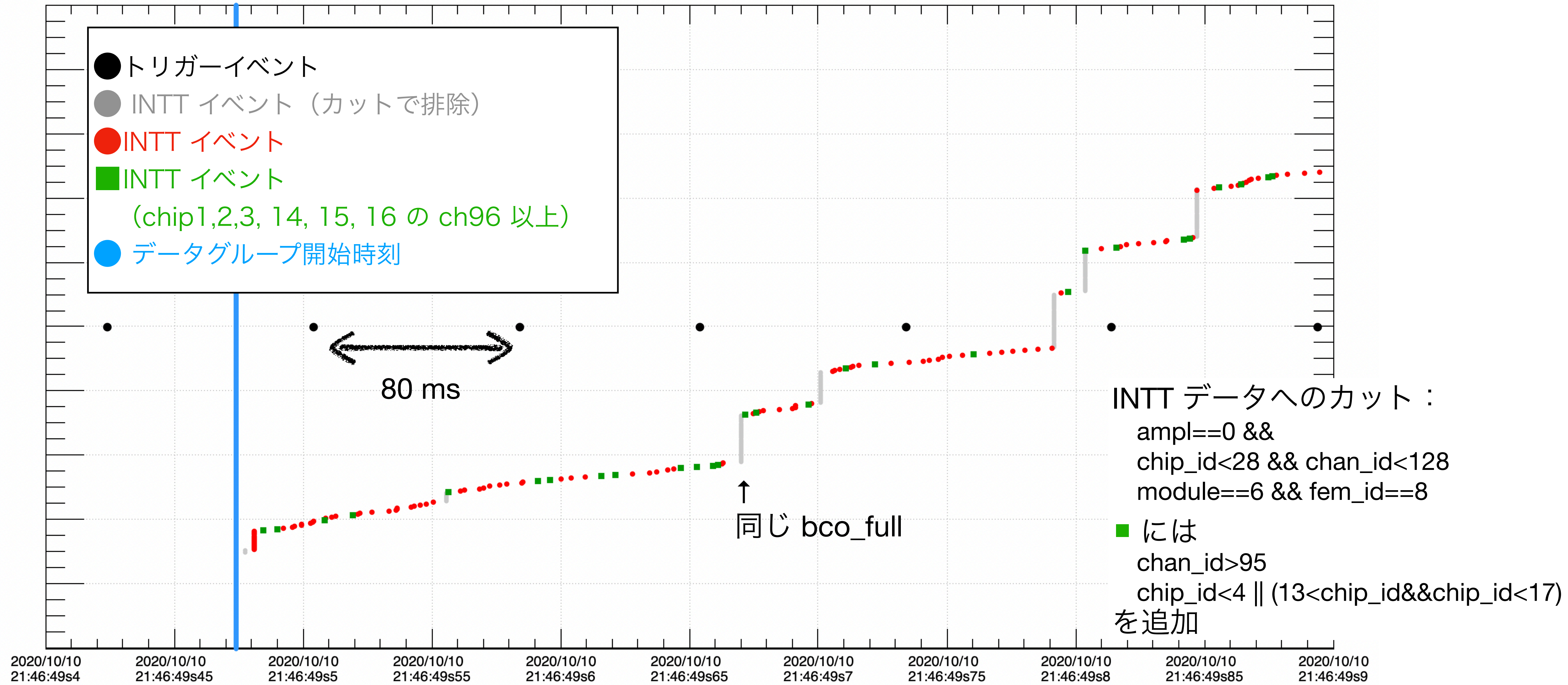


トリガーイベントの時間間隔 80 ms を bco_full のカウントで埋めるのは難しい・・・

INTT イベントの時間軸に対してトリガーイベントがシフトしているか未検証

トリガー 1 イベントだけ収集して様子を見てみたい

INTT データとトリガーデータを取ってみる



トリガーの ADC データはディレイモジュールのせいで信号が不安定だった・・・
⁹⁰Sr にコリメータをつけて、照射範囲を絞ってみたい

