KEK ADC
How to use

For Firmware Version 1104XXXXX

Rev. 0.3

Koji Ishidoshiro, Nagai Makoto
and KEK QUIET group

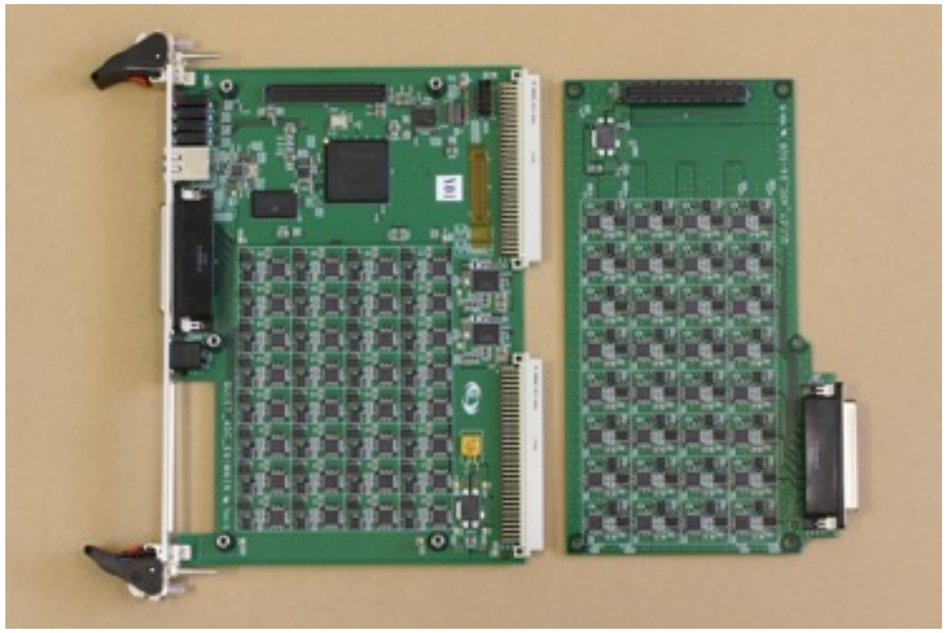
Contact information
- E-Mail: koji_at_post.kek.jp ('_at_' -> @)
- Skype: koji-ishi

Overview

KEK ADC

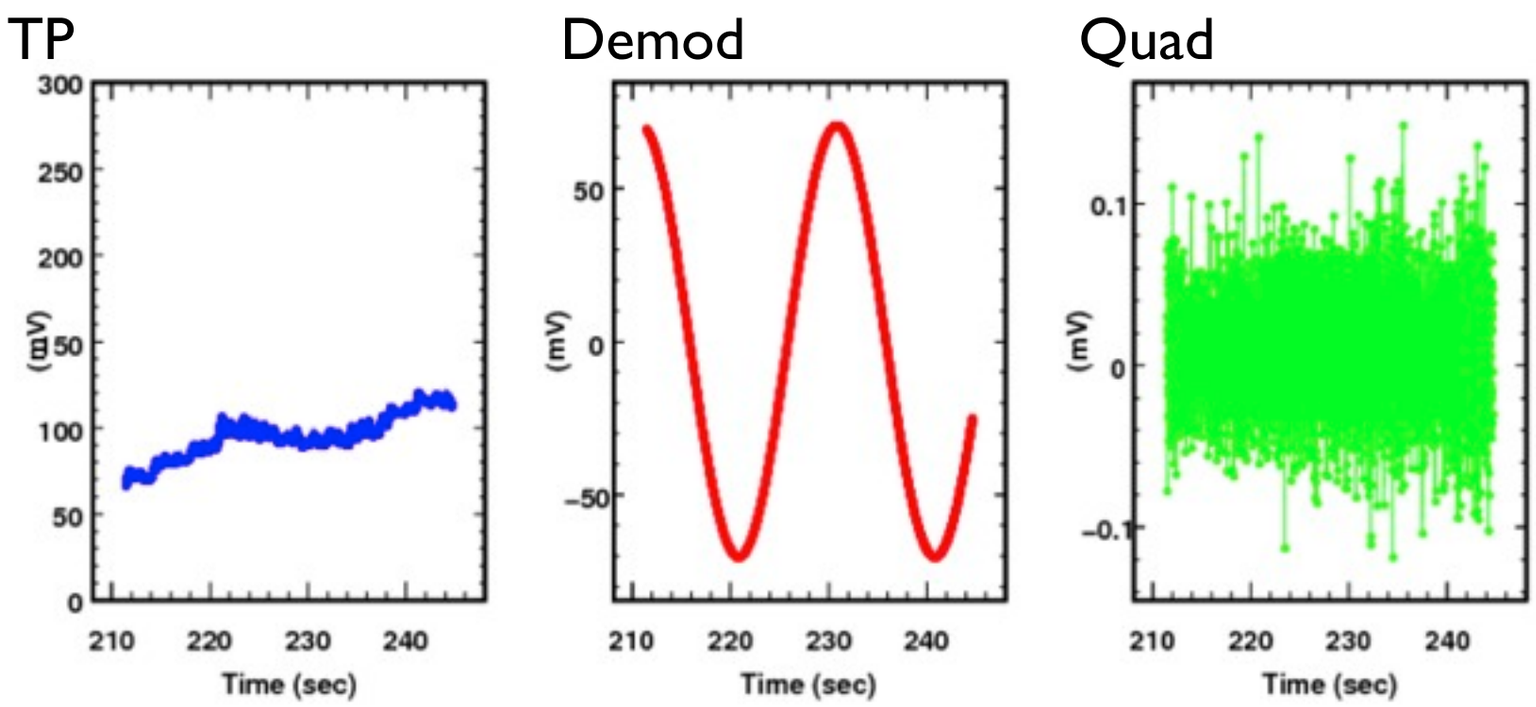
Main Features

- # of ADC: 64
  - Sampling: 800 kHz
  - Resolution: 18 bit
- Size: VME6U
- Easy data transfer using TCP/UDP
- On board double demodulation
- Recorded data: TP, Demod, Quad at 100 (50) Hz sampling
- Output port: 1 LVDS, 24 LVCMOS (for phase switch)
- No type-B glitch



Demonstration

- Artifactual modulated signals ⇒ TP: 1/f noise, Demod: clear sine signal, Quad: noise only



Easy data transfer using TCP/UDP

Two protocols

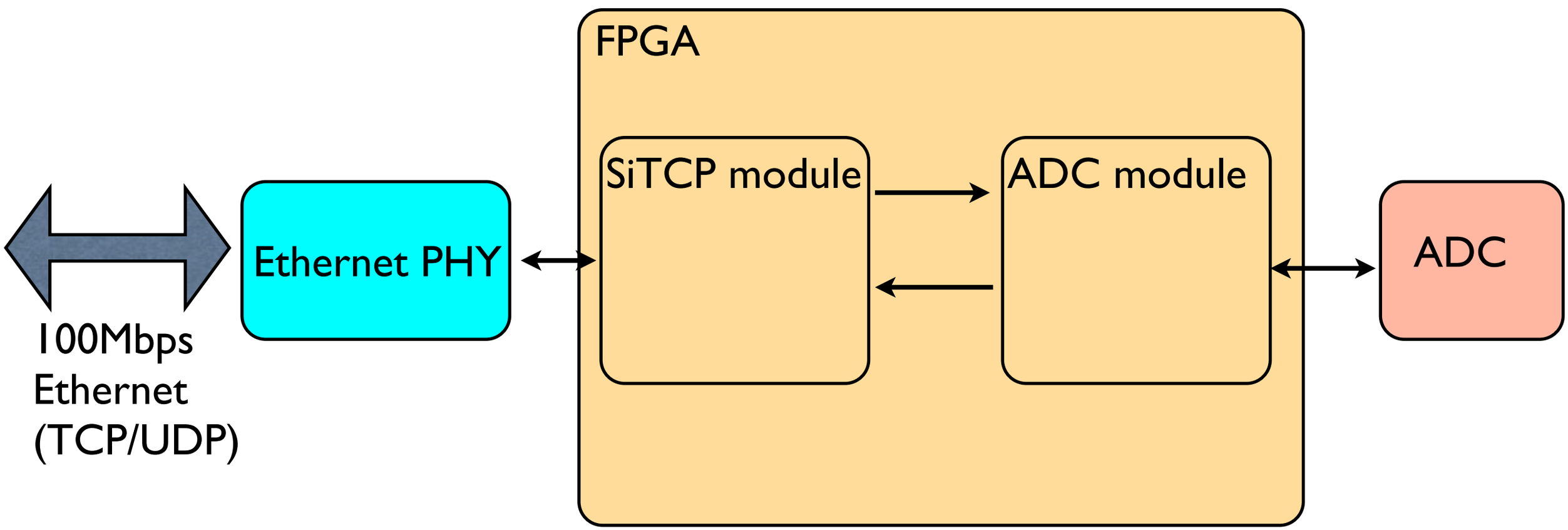
TCP: fast reliable connection.

Used for data transfer (TP, Demod, Quad)

UDP: simple, but less reliable connection.

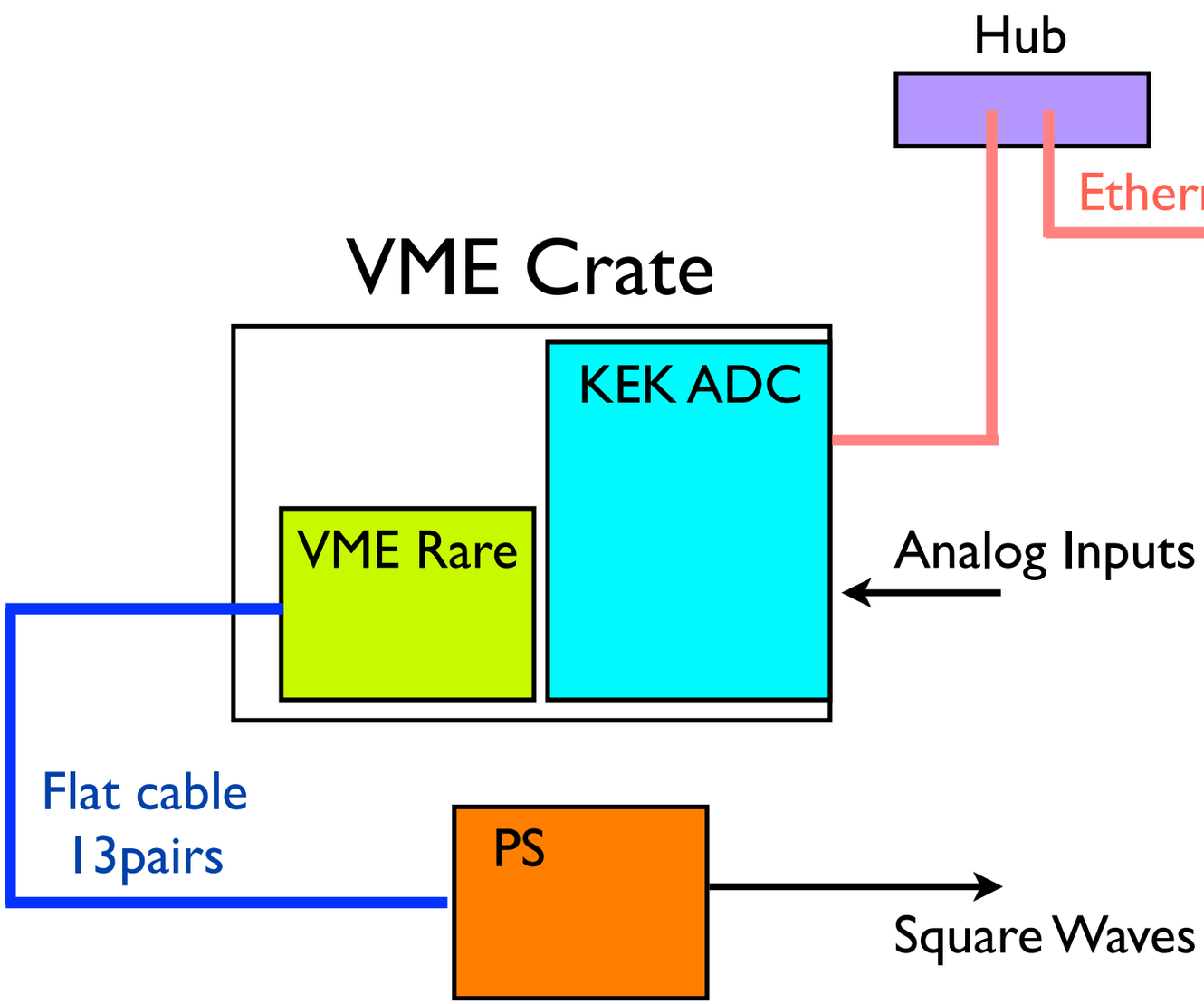
Used for the remote access to FPGA registers (slow control)

More details are described in Ref (T. Uchida, IEEE Trans. NSS 55, 1631, 2008 and T. Uchida, IEEE NSS Conference Record 1411, 2006)



Setup

Hardware Setup



Analog Inputs

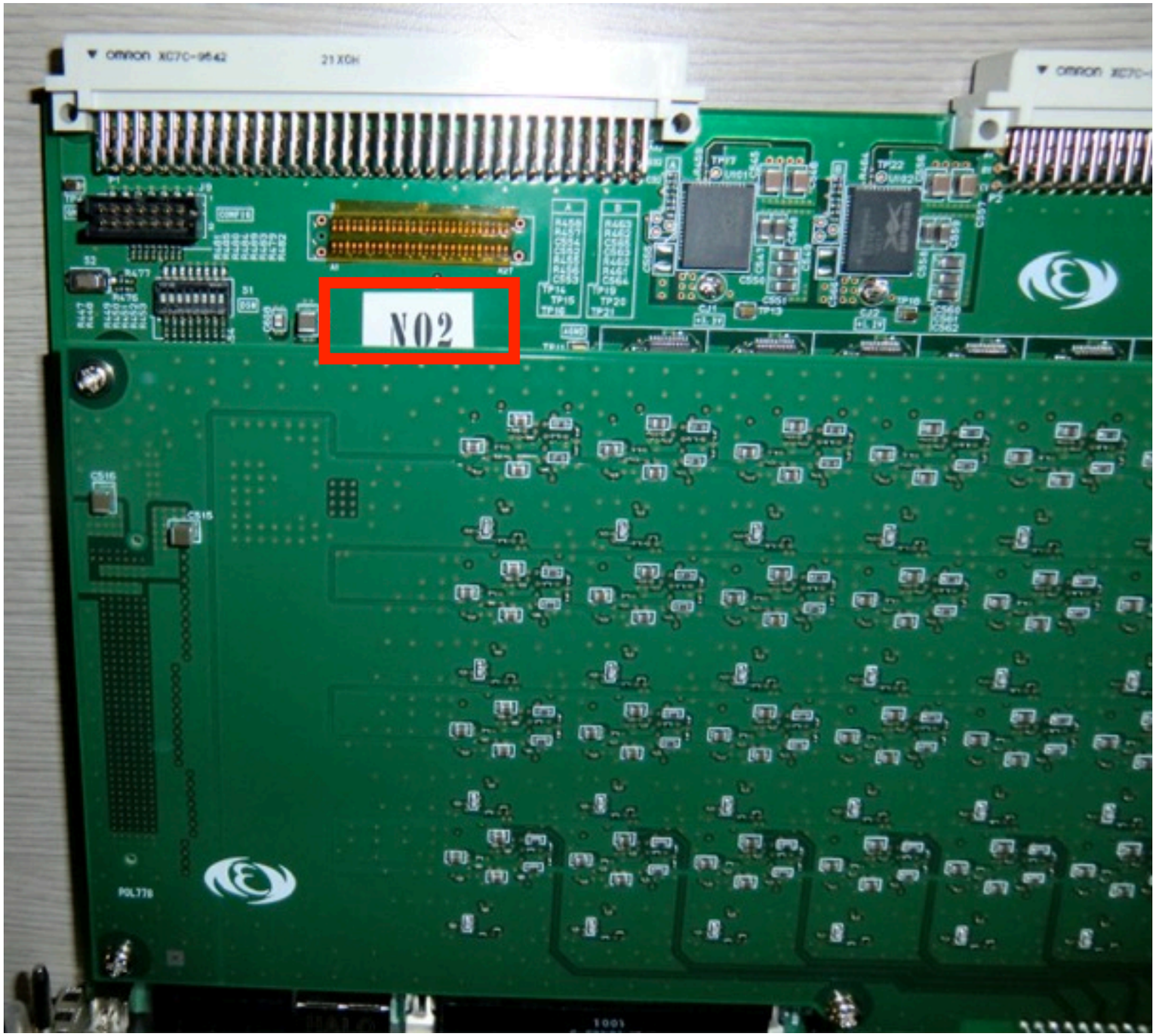
- Differential signals
- Range: ±2V
- Connector: Dsub78

Square Waves

- 4kHz, -4kHz, 50Hz, -50Hz
- Connector:Lemo
- Level: +1V/ 0V

KEK ADC works in the local network.
You must not connect the KEK ADC to the global network.

Check serial number

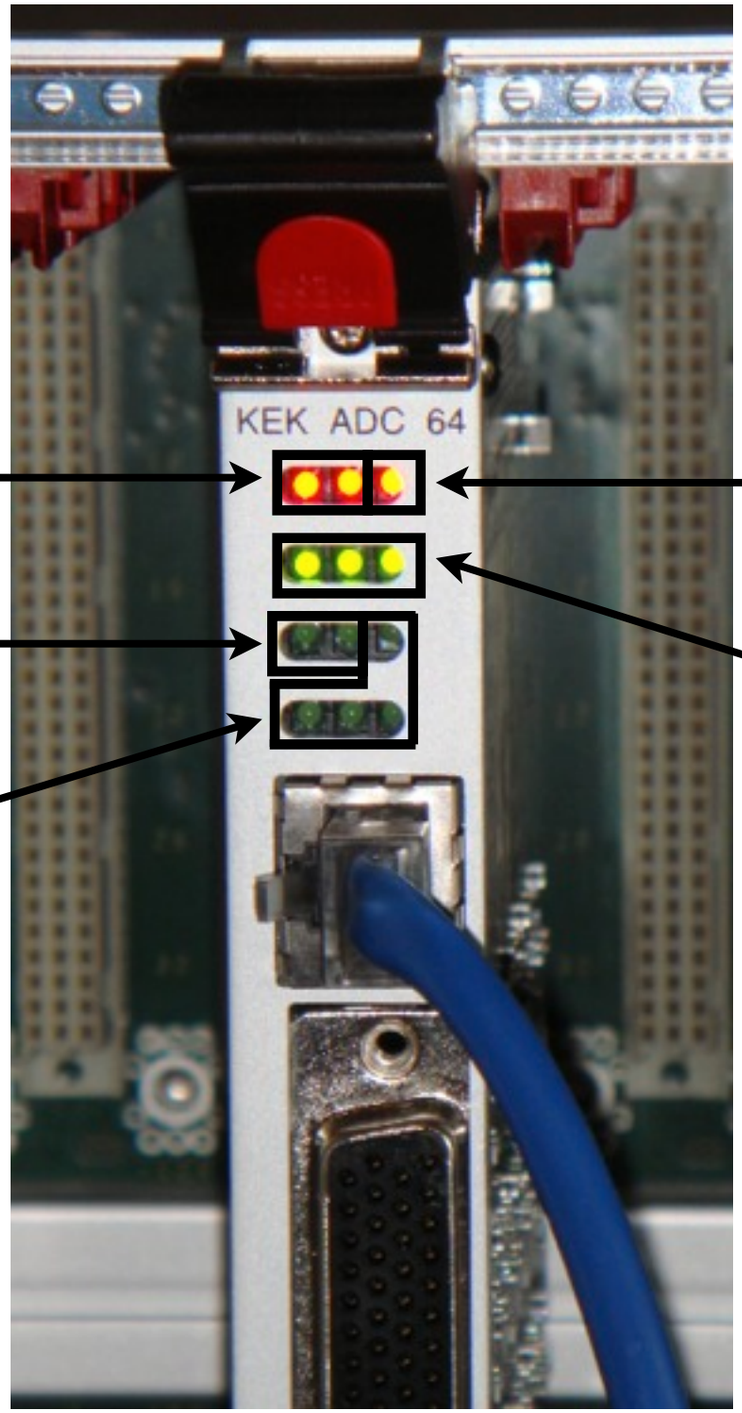


Serial number: No. 01
IP address:192.168.10.17
@FNAL

Serial number: No. 02
IP address:192.168.10.16
@KEK

Check front LEDs

12 LEDs are implemented



Power

Off

Flashing at 1Hz

Normal operation: Top three LEDs are on and bottom four LEDs are flashing at 1Hz.

FPAG_DONE

(FPGA_DONE turns off when FPGA reads firmware.)

Controlled by the slow control for the connection test.

Software Setup

See Makoto's document

Check connection

$ping 192.168.10.16 (for KEK ADC No.002 at KEK)

Firmware Version 11040502

$ping 192.168.10.17 (for KEK ADC No.001 at FNAL)

Firmware Version 11040501

Control from ADC server

SlowControl

Go to the working directory

Read command

$./SlowControl.sh [IP address] rd [beginning address] [length of read data]

Write command

$./SlowControl.sh [IP address] wr [beginning address] [write-data]

1 address : 8 bit
Hex number is used for r/w command.

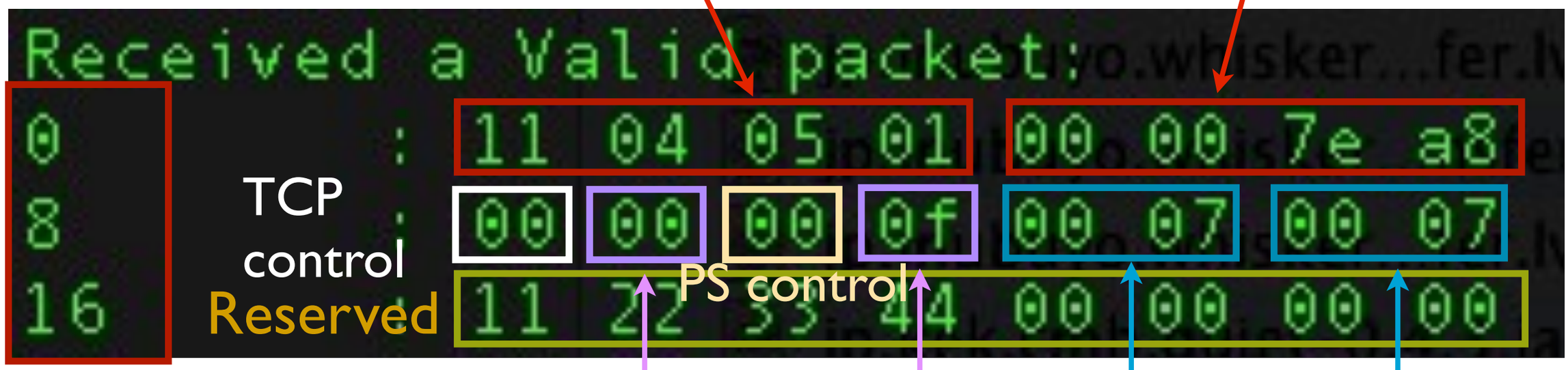
Example

```
moonstone:test koji$ ./SlowControl.sh 192.168.10.16 rd 0x00 0x20
Received a Valid packet:
0        : 11 04 05 01 00 12 a5 f7
8        : 00 00 00 04 00 07 00 07
16       : 11 22 33 44 00 00 00 00
24       : 00 00 00 00 00 00 00 00
moonstone:test koji$ ./SlowControl.sh 192.168.10.16 wr 0x0B 0x00
Received a Valid packet:
0        : 00
moonstone:test koji$ ./SlowControl.sh 192.168.10.16 rd 0x00 0x20
Received a Valid packet:
0        : 11 04 05 01 00 12 ac 78
8        : 00 00 00 00 00 07 00 07
16       : 11 22 33 44 00 00 00 00
24       : 00 00 00 00 00 00 00 00
moonstone:test koji$
```

Register Address

Firmware Version: YY MM DD NN
(This example: 1st version generated in 5th April, 2011)

Time Counter
(100 Hz, 32 bits)



TCP control

PS control

Reserved

Read counter

Masked count after edges

Masked count before edges

X09(7:2): No meaning
X09(1): Data FIFO Full
X09(0): No meaning

X0B(7:3) : No meaning
X0B(2:0) : LED control (active high)

Register Address

Address	Function	Read/Write	Initial value
0x00-03	Firmware Version (YY MM DD NN)	R	0x11040501 or 0x11040502
0x04-07	TimeCounter	R	0x00000000
0x08	TCP Control	R/W	0x00
0x09	Fifo status	R	0x00
0x0A	PS control	R/W	0x00
0x0B	LED	R/W	0x0F
0x0C-0D	Masking count1	R/W	0x0007
0x0E-0F	Masking count2	R/W	0x0007
0x10-13	Reserved for snapshot	R	0x11223344

TCP control

TCP control register

X08[3]: Sampling Speed
- high => single demod mode at 100Hz,
- low => double demod mode at 50Hz

X08[0]: DAQ stop (active low)

Other registers must be '0'.

Recommendation for the TCP control register

Data transformation (single demod mode)

```
$./SlowControl.sh 192.168.10.1? wr x08 0x09
```

Data transformation (double demod mode)

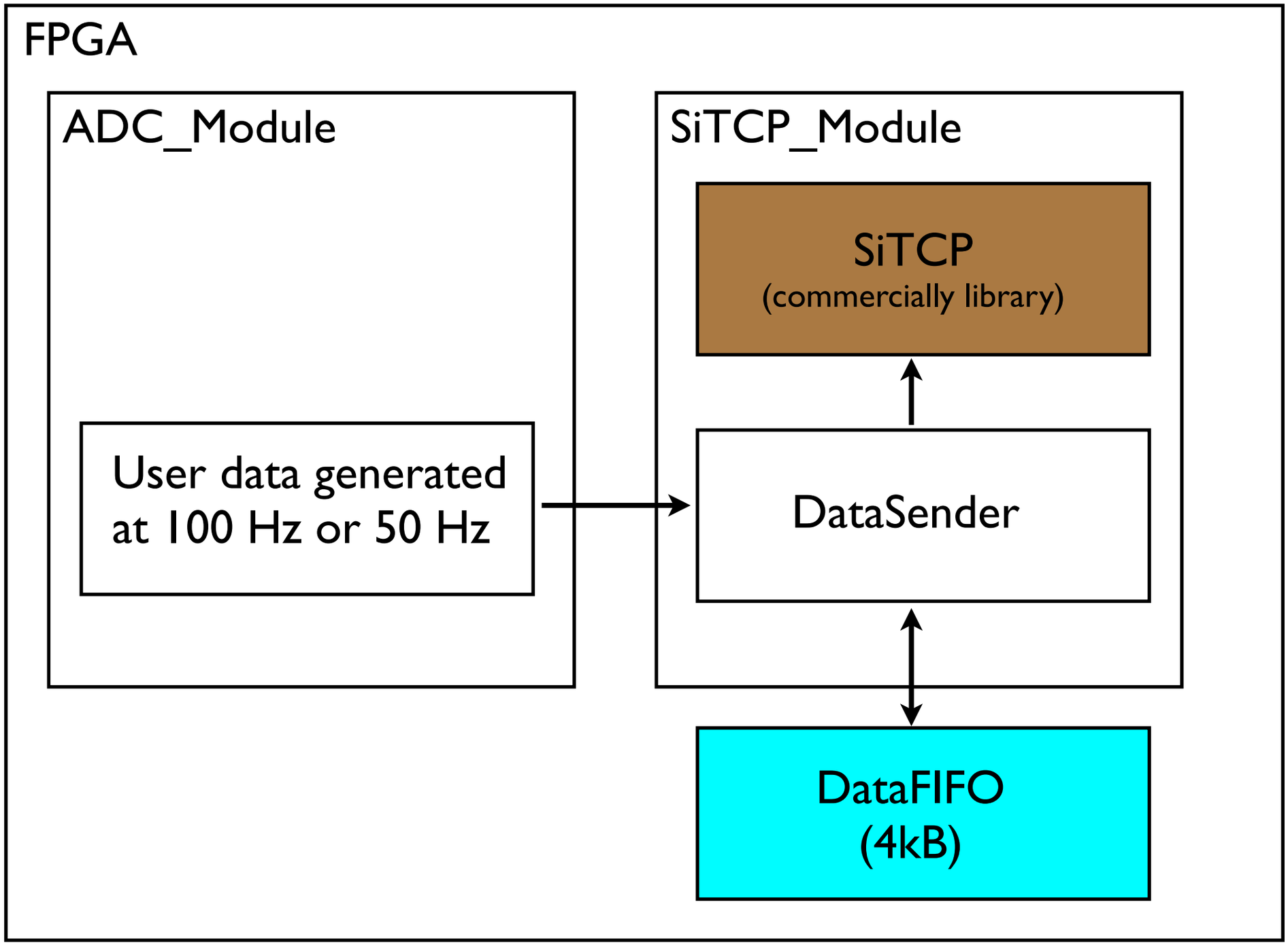
```
$./SlowControl.sh 192.168.10.1? wr x08 0x01
```

Request to stop DAQ

```
$./SlowControl.sh 192.168.10.1? wr x08 0x00
```

Fifo status

You do not have to take care Fifo status
in current single board operation.



PS control

Only: Firmware Version 11042401

PS control register

X0A[7]: PSA1 clock mode (4kHz) enable (active high)
X0A[6]: PSA2 clock mode (not 4kHz) enable (active high)
X0A[5]: PSB1 clock mode (50Hz) enable (active high)
X0A[4]: PSB2 clock mode (not 50Hz) enable (active high)

X0A[3]: decides off state value in PSA1
X0A[2]: decides off state value in PSA2
X0A[1]: decides off state value in PSB1
X0A[0]: decides off state value in PSB2

Eample

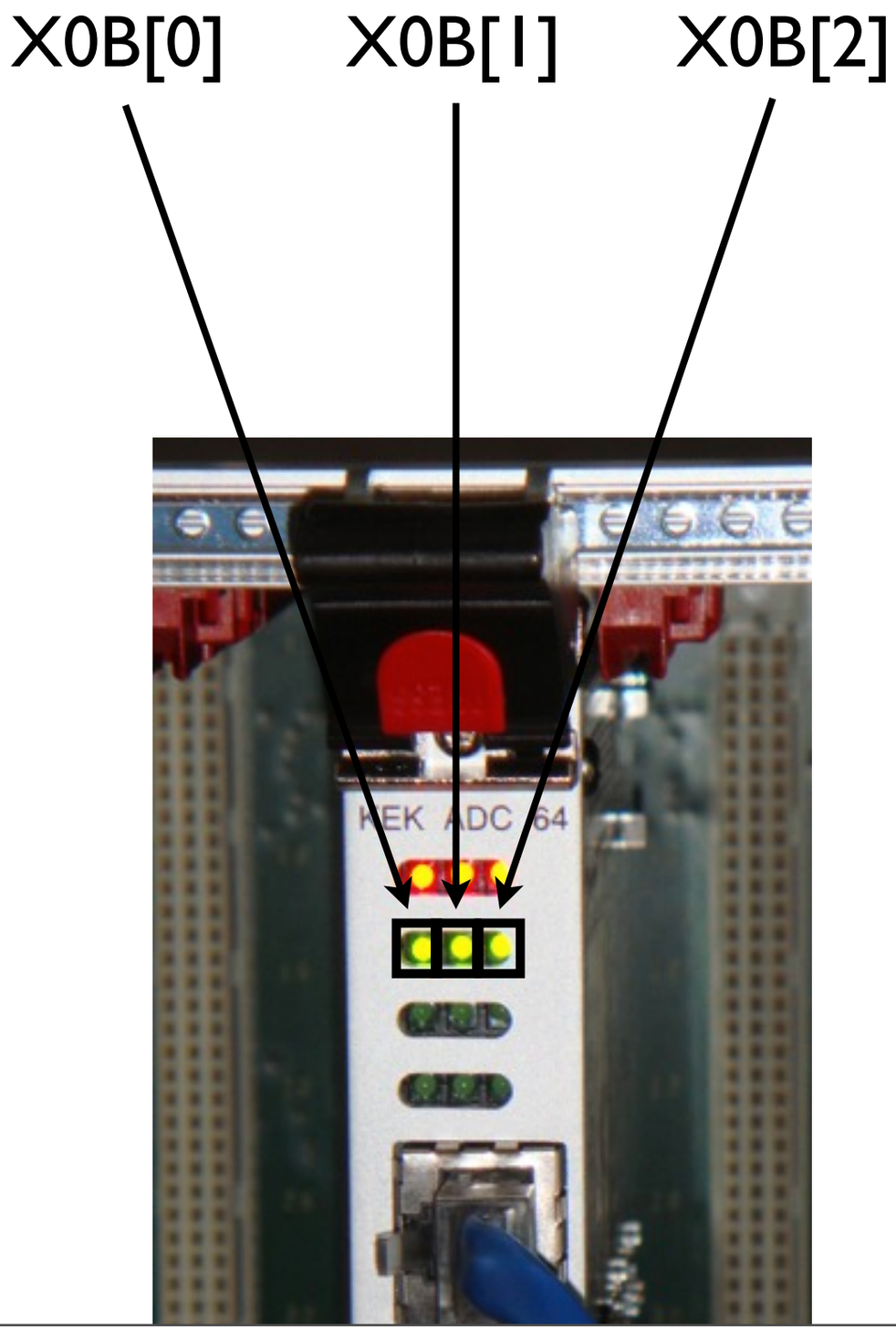
X0A=x"81" = "10000001"

PA1: Clock mode (4KHz), PA2: '0', PB1: '0', PB2: '1'

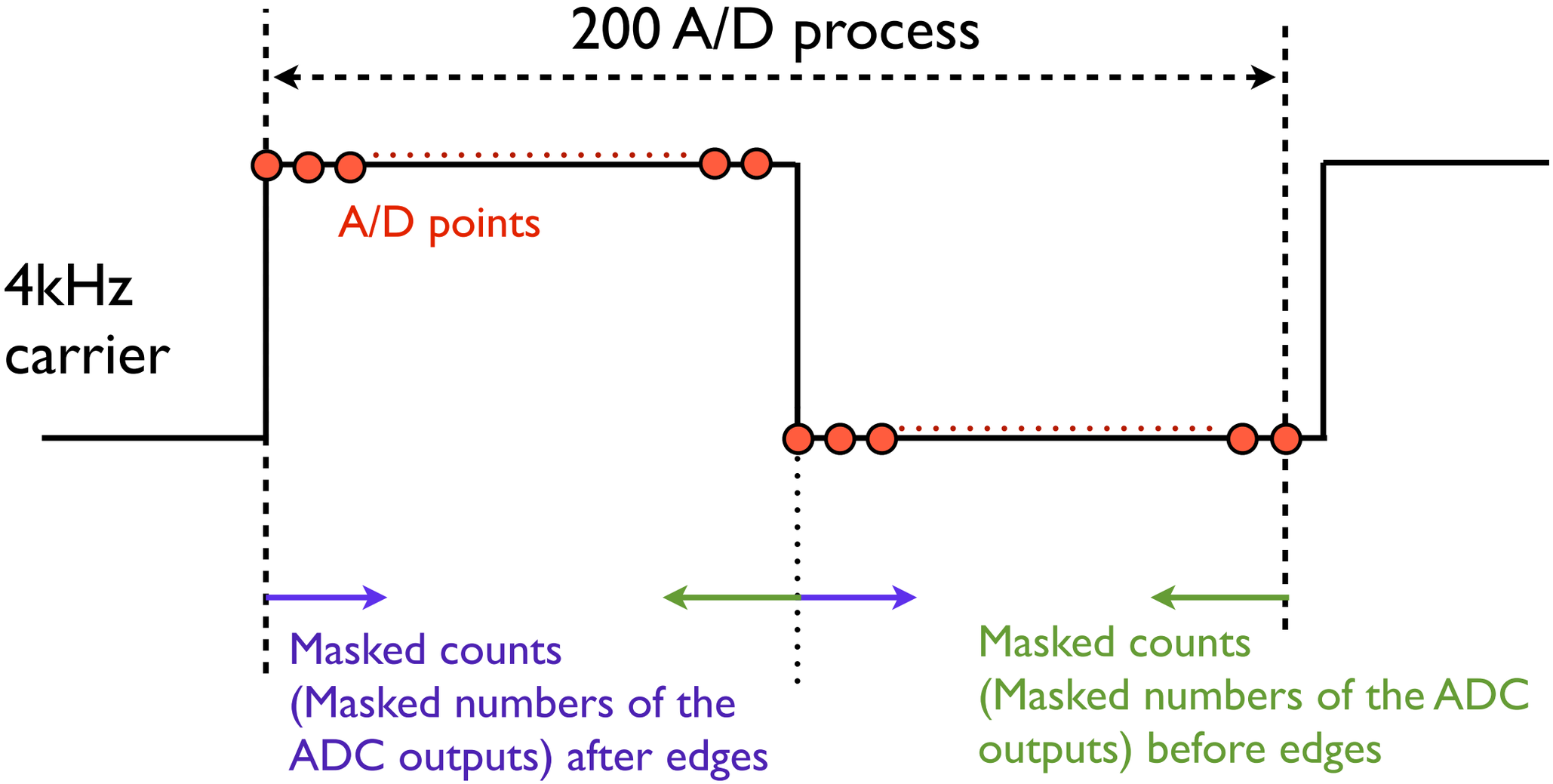
X0A=x"35" = "00110101"

PA1: '0', PA2: '1', PB1: Clock mode, PB2: Clock mode

LED



Mask parameters



Initial values

Masked counts after edges : X0C-0D = x07 = 7

Masked counts before edges : X0E-0F = x07 = 7

Start data transformation

1. Prepare two terminals
2. Socket open in PC (input $./SiTCP.sh)
3. Check connection establishment
4. Turn off DAQ stop
5. Check file name

Example

```
moonstone:test koji$ ./SlowControl.sh 192.168.10.16 wr
 0x08 0x09
Received a Valid packet:
0        : 09
moonstone:test koji$
```

Turn off DAQ stop

Socket open command

Check this message

```
^Chogans-computer:test quiet$ ./SiTCP.sh
2011/04/25 18:59:21.923 10 INFO jp.inubuyo.bone.tcp.CliantPort [connect] connection to 192.168.10.17 is established.
2011/04/25 18:59:21.952 10 INFO jp.kek.cmb.quiet.ADCFrameCliant [work] writer SiTCP1_20110425185921.adc opened.
2011/04/25 18:59:22.035 10 INFO jp.kek.cmb.quiet.ADCFrameCliant [work] SiTCP1_20110425185921.adc is linked by data/latest_1.adc
2011/04/25 18:59:54.784 11 WARNING jp.kek.cmb.quiet.ADCFrameCliant$ADCFrameCliantTalker [talk] 192.168.10.17:24: scale jump; -2147483648 -> 6880
```

File name

Taken data are saved in ‘data/[File name]’ using binary format.
(see Makoto’s documents)

End data transformation

1. Request DAQ stop
2. Close socket in PC using "Ctr-C"

```
moonstone:test koji$ ./SlowControl.sh 192.168.10.16 wr
 0x08 0x09
Received a Valid packet:
0        : 09
moonstone:test koji$ ./SlowControl.sh 192.168.10.16 wr
 0x08 0x00
Received a Valid packet:
0        : 00
moonstone:test koji$
```

Request DAQ stop

```
^Chogans-computer:test quiet$ ./SiTCP.sh
2011/04/25 18:59:21.923 10 INFO jp.inubuyo.bone.tcp.CliantPort [connect] connection to 192.168.10.17 is established.
2011/04/25 18:59:21.952 10 INFO jp.kek.cmb.quiet.ADCFrameCliant [work] writer SiTCP1_20110425185921.adc opened.
2011/04/25 18:59:22.035 10 INFO jp.kek.cmb.quiet.ADCFrameCliant [work] SiTCP1_20110425185921.adc is linked by data/latest_1.adc
2011/04/25 18:59:54.784 11 WARNING jp.kek.cmb.quiet.ADCFrameCliant$ADCFrameCliantTalker [talk] 192.168.10.17:24: scale jump; -2147483648 -> 688
```

"Ctr-C" Here

Data view

Sample is available.

"qufs:/home/koji/KEKADC/view/src20110408/"

Example:

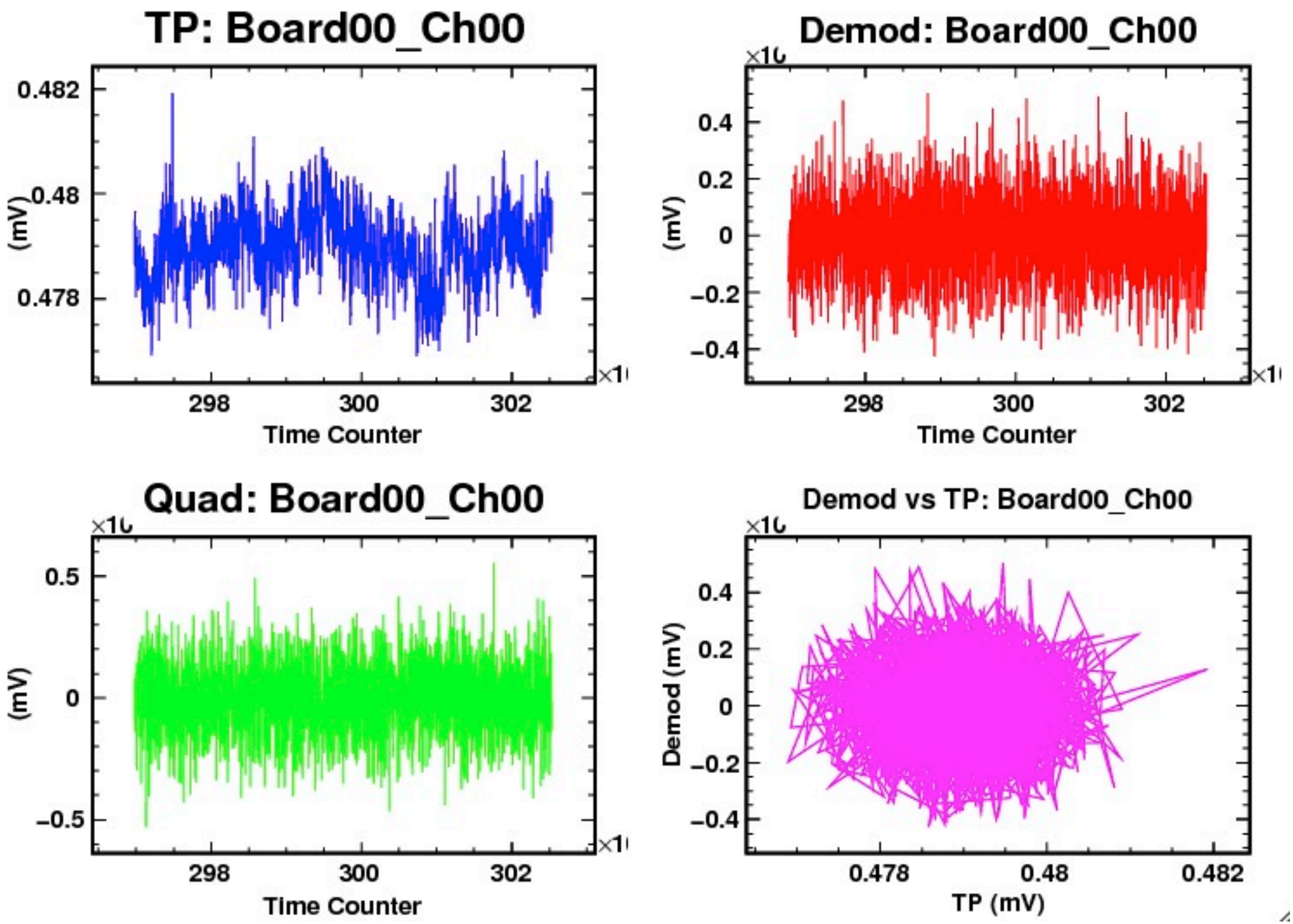
python program

```
[koji@CMBns prog]$ python view_kek_adc.py /dsk/PUB_00/koji/KEK_ADC/201104/FNAL_test/test.dat
-----------------------------------
    /dsk/PUB_00/koji/KEK_ADC/201104/FNAL_test/test.dat
-----------------------------------
ADC Frame Cliant ver. 20101214a
Event Builder Data Format ver. 20101124
Number of ADC Boards: 1
Board 1: 192.168.10.16:24
====
====
====
====
====
The last (10th) line of the header ====
[296978 296980 296982 ..., 302546 302548 302550]
Info in <TCanvas::Print>: eps file board00/c_board.eps has been created
Info in <TCanvas::Print>: eps file board00/z_ch00.eps has been created
Info in <TCanvas::Print>: eps file board00/z_ch01.eps has been created
Info in <TCanvas::Print>: eps file board00/z_ch02.eps has been created
Info in <TCanvas::Print>: eps file board00/z_ch03.eps has been created
```

File name

Note: Python with PyROOT and numpy is required.

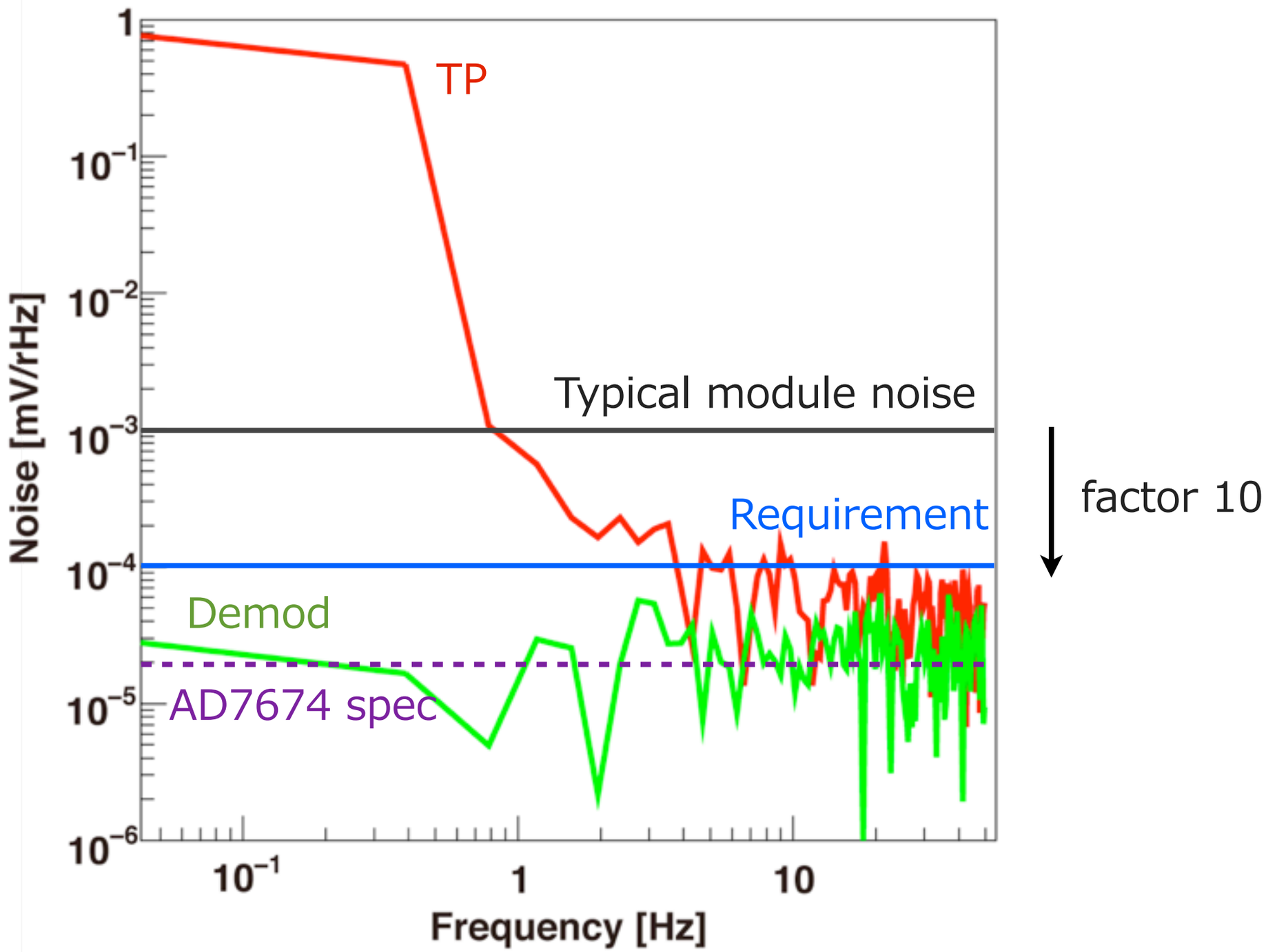
Example of z_chXX.eps:

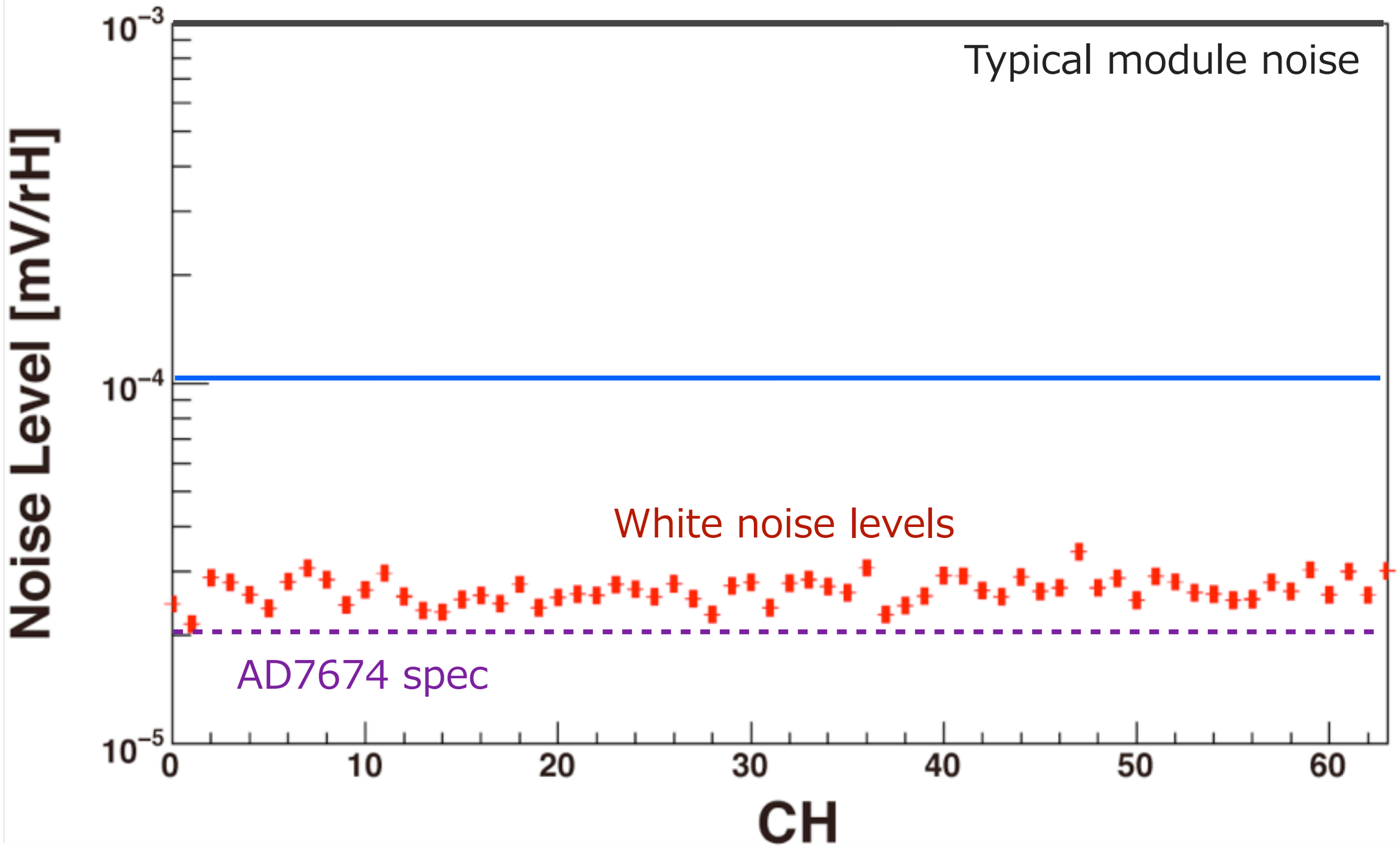


Noise performance

Noise Performance

Typical noise spectrum



Noise Performance

White noise level