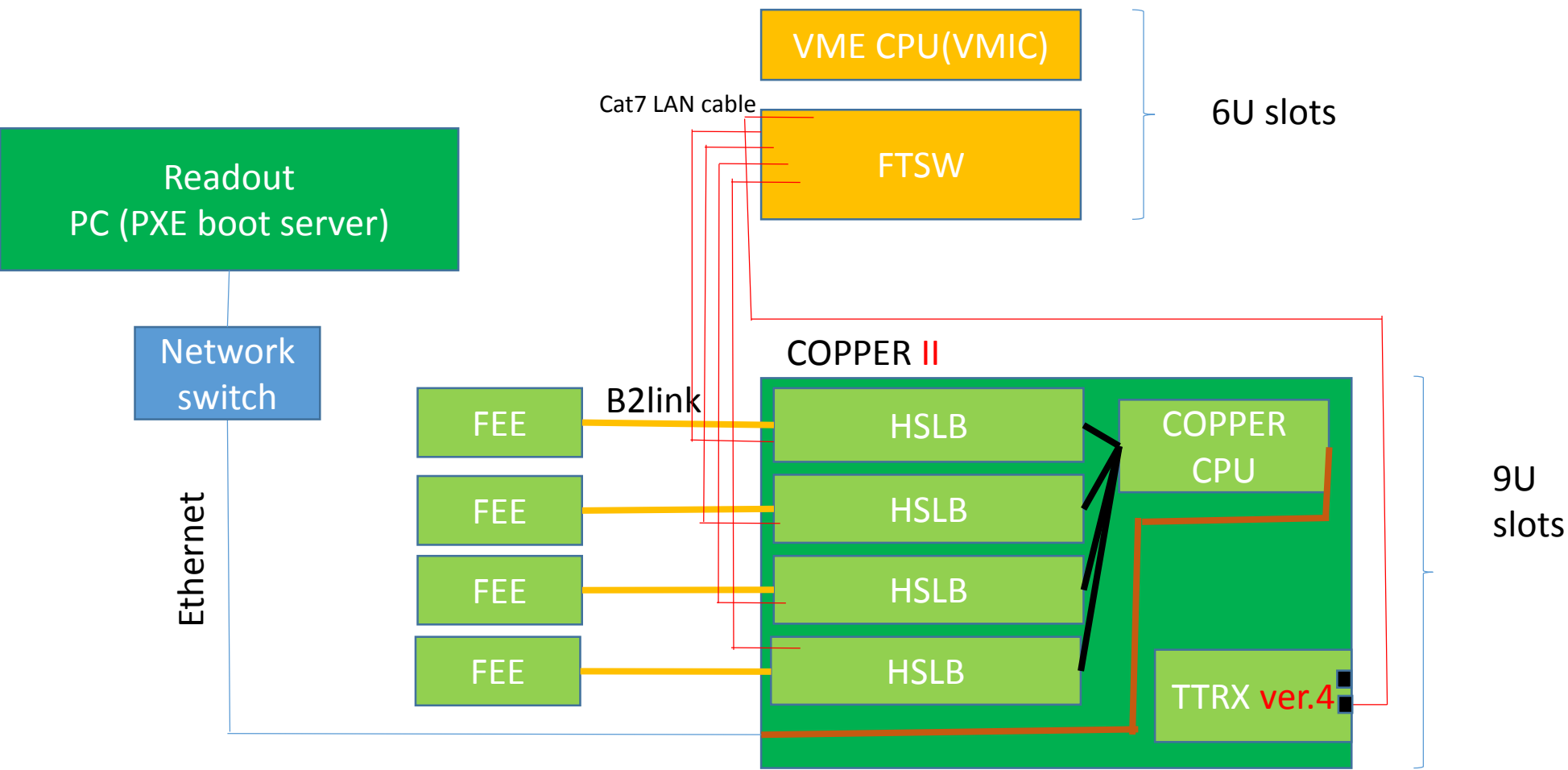# Pocket DAQ manual (rev. 17080)

Apr. 13, 2015

Satoru Yamada
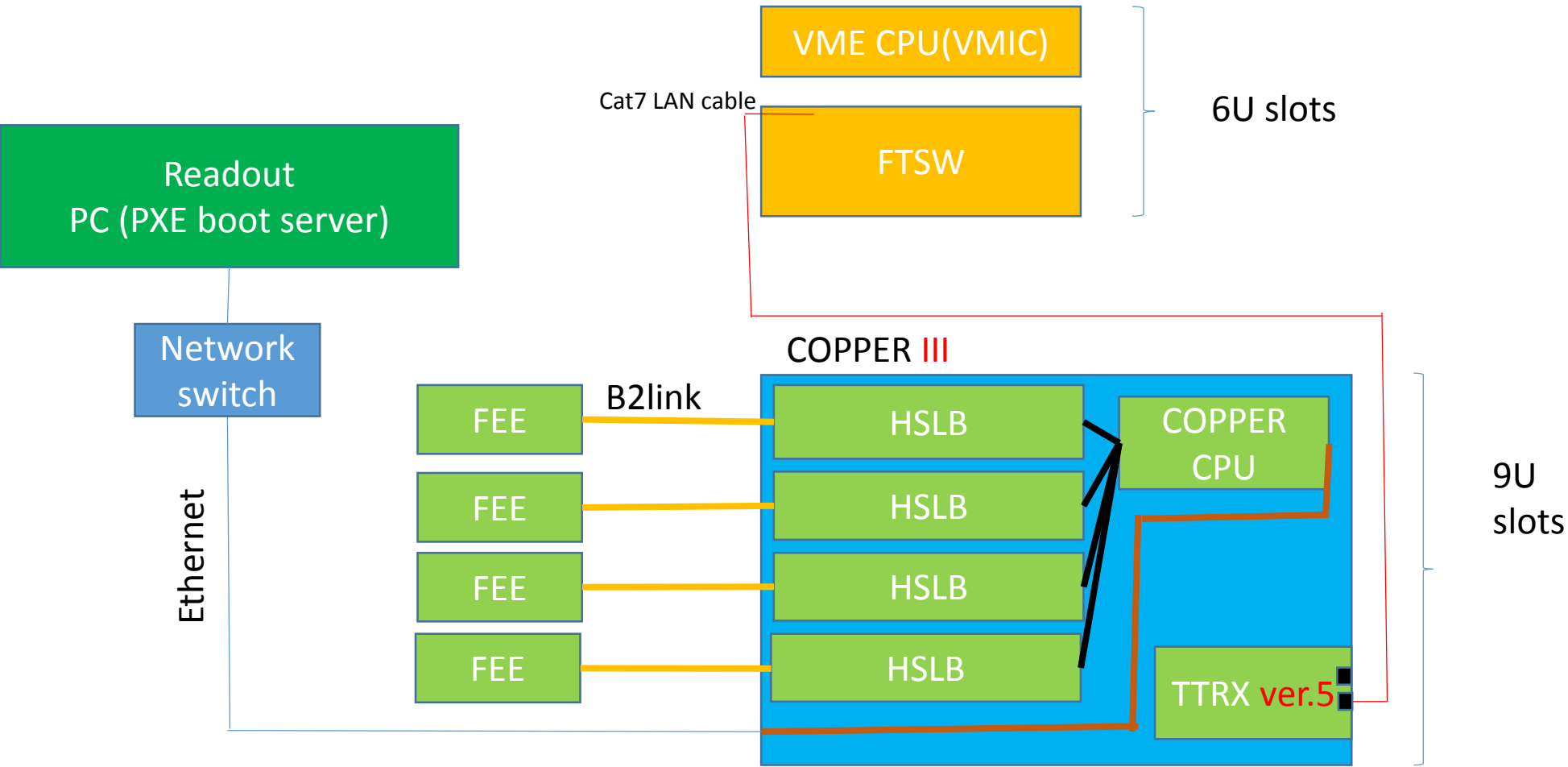
Connection between components (COPPER-II)

About port assignment of FTSW, see
svn+ssh://xxx@bdaq.local.kek.jp/bdaq/svn/firmware/ft2u/trunk/doc/ft2u.pdf

VME CPU(VMIC)

6U slots

Cat7 LAN cable

FTSW

Readout
PC (PXE boot server)

Network
switch

COPPER II

Ethernet

FEE — B2link — HSLB — COPPER CPU

FEE — HSLB

FEE — HSLB

FEE — HSLB

TTRX ver.4

9U slots

Connection between components (COPPER-III)

About port assignment of FTSW, see
svn+ssh://xxx@bdaq.local.kek.jp/bdaq/svn/firmware/ft2u/trunk/doc/ft2u.pdf

VME CPU(VMIC)

6U slots

Cat7 LAN cable

FTSW

Readout
PC (PXE boot server)

Network
switch

COPPER III

B2link

FEE — HSLB

COPPER
CPU

9U
slots

FEE — HSLB

FEE — HSLB

FEE — HSLB

TTRX ver.5

Ethernet

Old Software components

These are stored under https://belle2.cc.kek.jp/svn/trunk/software/daq

COPPER CPU

Receive data from HSLB and send to ROPC : RecvSendCOPPER.py

COPPER CPU

Receive data from HSLB and send to ROPC : RecvSendCOPPER.py

Readout PC

Merger : eb0.cc

recorder: RecvStream1.py

RootOutput.root

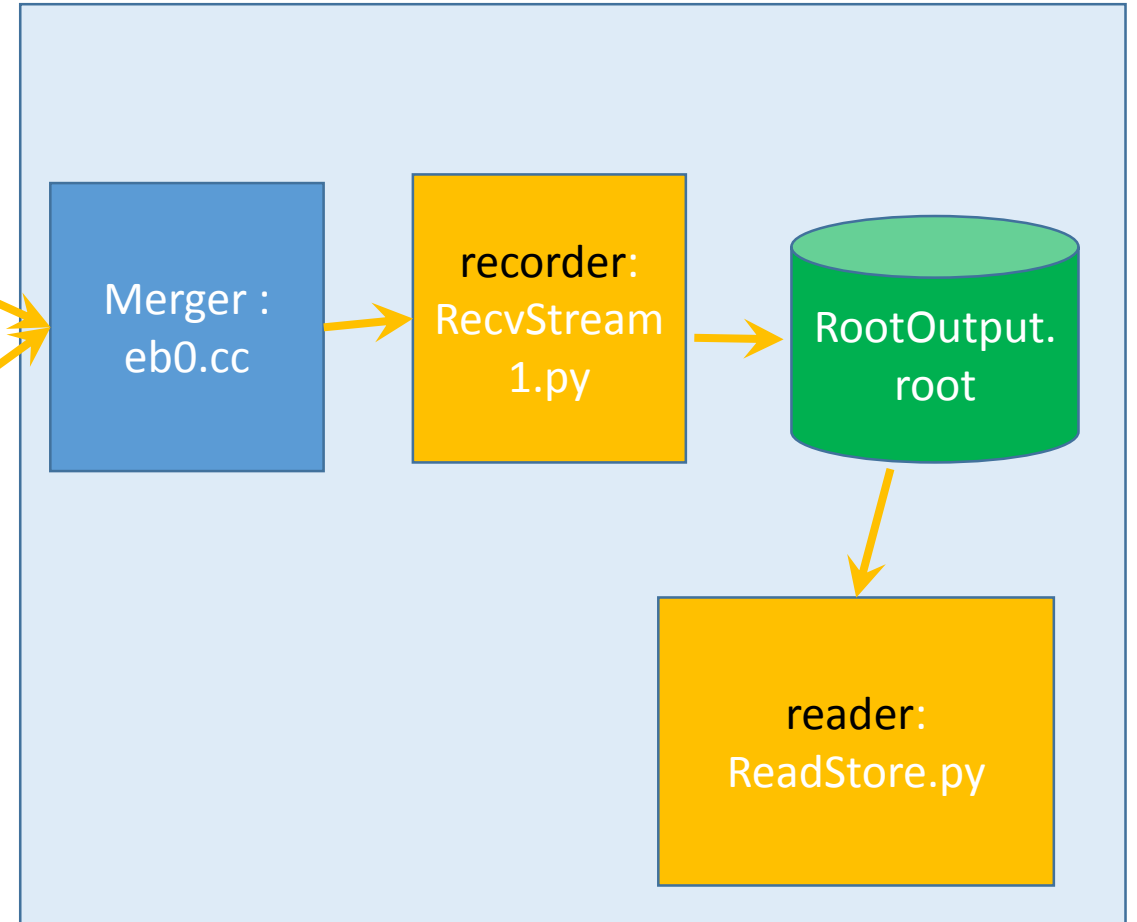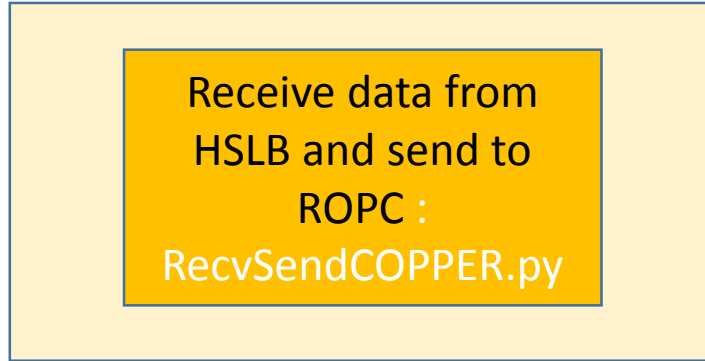reader: ReadStore.py
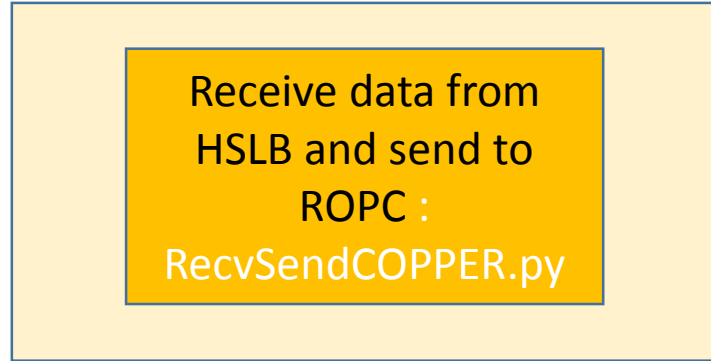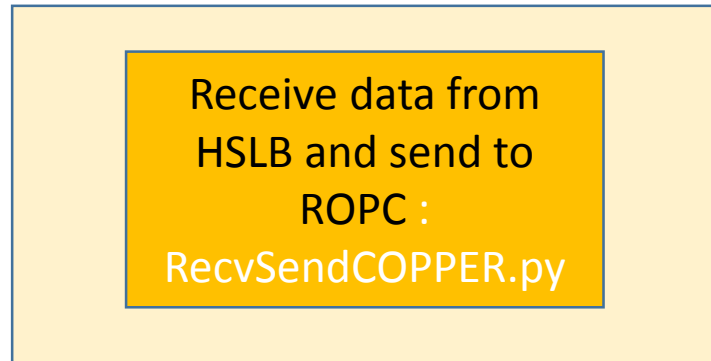
Orange color components are basf2 components.

Old Software components (from rev. 13542 to rev. 15005)

These are stored under https://belle2.cc.kek.jp/svn/trunk/software/daq

COPPER CPU

Receive data from HSLB and send to ROPC : RecvSendCOPPER.py

COPPER CPU

Receive data from HSLB and send to ROPC : RecvSendCOPPER.py

Readout PC

recorder: RecvStream0.py

recorder: RecvStream0.py

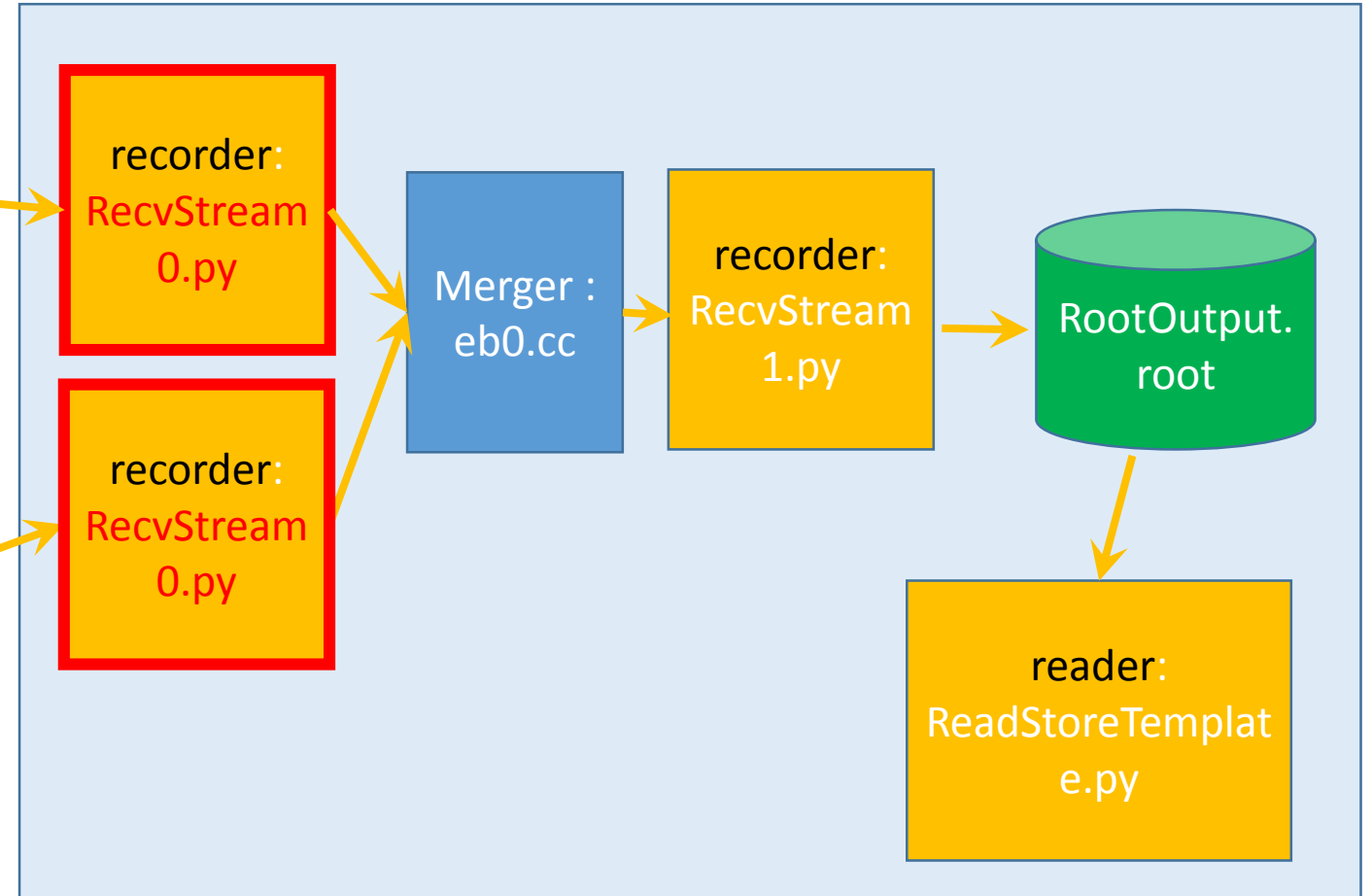Merger : eb0.cc

recorder: RecvStream1.py

RootOutput.root

reader: ReadStoreTemplate.py
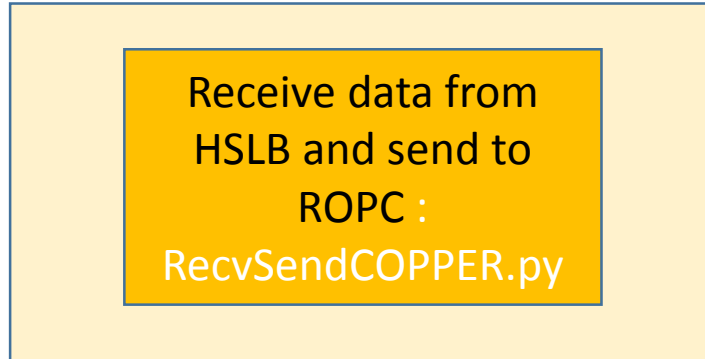
Orange color components are basf2 components.

New Software components (from rev. 15006)

These are stored under https://belle2.cc.kek.jp/svn/trunk/software/daq

COPPER CPU

Receive data from HSLB and send to ROPC : RecvSendCOPPER.py

COPPER CPU

Receive data from HSLB and send to ROPC : RecvSendCOPPER.py

Readout PC

Receive/send: RecvStream2.py

Receive/send: RecvStream2.py

Merger : eb0.cc

recorder: RecvStream1.py

RootOutput.root

reader: ReadStoreTemplate.py
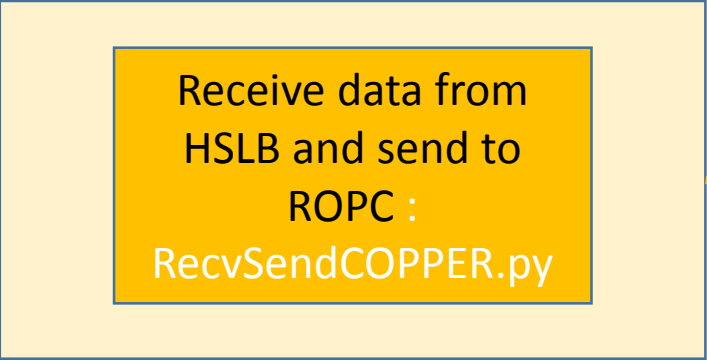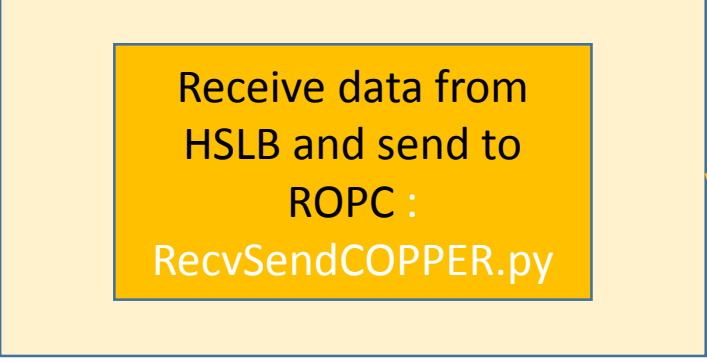
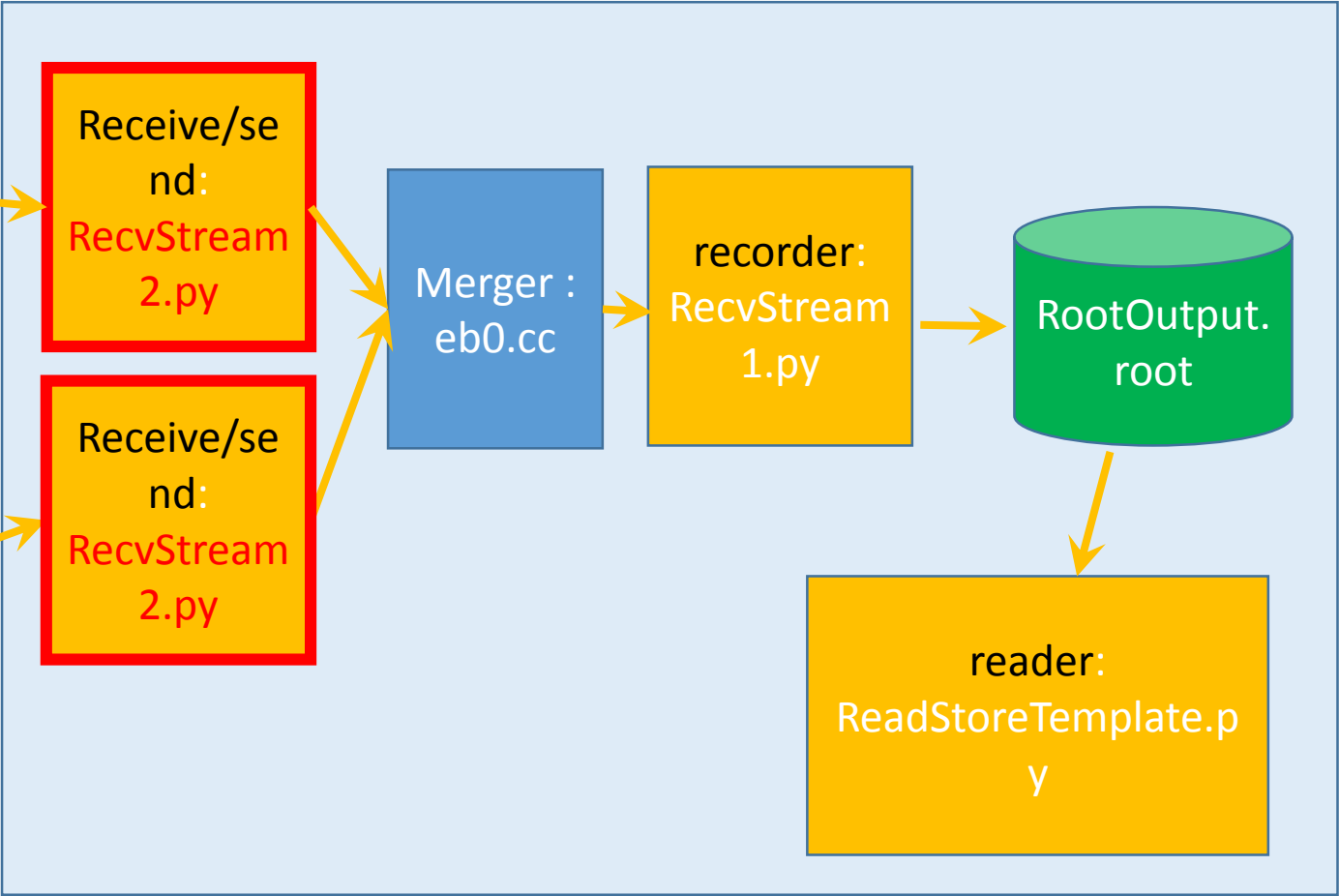**Orange color components are basf2 components.**

Pocket DAQ w/o slow controller and GUI

0. Before using pocket DAQ

- Setup a PXE boot server for COPPER CPU and install driver for COPPER etc
  - See https://belle2.cc.kek.jp/~twiki/bin/view/Detector/DAQ/PocketDAQ
- Install basf2 on both COPPER CPU and Readout PC
  - See https://belle2.cc.kek.jp/~twiki/bin/view/Software/SoftwareInstallation .
    "addall" does no install the daq package. You need to do "addpkg daq".
  - Check daq/Sconscripts and check if env['CONTINUE'] = False is commented out.
  - Compile with scons.
- Compile eventbuilder
  - cd ${BELLE2_LOCAL_DIR}/daq/eventbuilder/evb0/ ; gmake eb0

Notice from Konno-san about compiling daq package

IMPORTANT!

- Please run the following command to update nsm2 programs in your external directory

$ source ${BELLE2_LOCAL_DIR}/ daq/slc/extra/nsm2/export.sh

Otherwise you may get error messages about compilation of slow control programs ;

daq/slc/nsm/src/NSMData.cc: In member function 'void Belle2::NSMData::parse(const char*, bool)':*

daq/slc/nsm/src/NSMData.cc:299:68: error: too many arguments to function 'NSMparse nsmlib_parsefile(const char*, int, const char*, char*)'*

In file included from include/daq/slc/nsm/NSMData.h:20:0,

from daq/slc/nsm/src/NSMData.cc:1:

Install GLEW for running basf2 program

After external ver.0.5.0?, basf2 requires GLEW(The OpenGL Extension Wrangler Library).
The following procedure worked fine.
1, install the yum epel repository.
ropc01$ wget http://ftp-srv2.kddilabs.jp/Linux/distributions/fedora/epel/5/i386/epel-release-5-4.noarch.rpm
ropc01$ ssh cpr** -lroot
cpr**# rpm -ivh epel-release-5-4.noarch.rpm
cpr**# ls -lrt /etc/yum.repos.d/
cpr**# exit
2, install glew on COPPER
ropc01$ sudo yum install glew --installroot=/tftpboot/copper/root

0.5, Set active HSLB slots in TTRX register

From Nakao-san's mail : [b2link_ml:0159] Re: ft2u-0.65, b2tt-0.31, hslb-0.36

To partially mask HSLB, one can modify register 130 of tt4r:
  - setting each of bit 0..3 to 0 corresponds to masking HSLB A..D
  - default value of register 130 is 0010000f (all HSLB are enabled)
  - bit 20..23 are read only and should be 1 as above
  - for example, one can mask HSLB-A by
    copper_CPU $ regrx 130 0000000e

Example :

When you use;
Slot A : **copper_CPU $ regrx 130 00000001**
Slot B : **copper_CPU $ regrx 130 00000002**
Slot A and B : **copper_CPU $ regrx 130 00000003**

1, Set parameters in run_start.sh (1)

[ROPC] % cd ${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts

[ROPC] % emacs run_start.sh

Set arguments of start_copper.sh

Usage : start_copper.sh **<HOSTNAME> <COPPER node> <FINNESSE bit flag: A=1, B=2, C=4, D=8> <Use network shared memory : Yes=1, No=0> <node name(used for Network Shared Memory)>**
e.g.)
/usr/bin/xterm -e ${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts/start_copper.sh cpr006 1 1 0 0&
/usr/bin/xterm -e ${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts/start_copper.sh cpr007 2 3 0 0&

<COPPER node ID> will be attached to RawCOPPER header.

1, Set parameters in run_start.sh (2)

NOTICE : Environment variables for basf2

In ${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts/copper.sh, a line, "source ~/.bash_profile", is for setting up basf2 environment. You need to add basf2 setting commands in your .bash_profile (or other script file).
Please see "Setup of Software Tools" at https://belle2.cc.kek.jp/~twiki/bin/view/Software/SoftwareInstallation **for details.**

[${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts/copper.sh]
#
setup basf2 environment (See "Setup of Software Tools" at
https://belle2.cc.kek.jp/~twiki/bin/view/Computing/SoftwareInstallation
#
source ~/.bash_profile

1, Set parameters in run_start.sh (3)
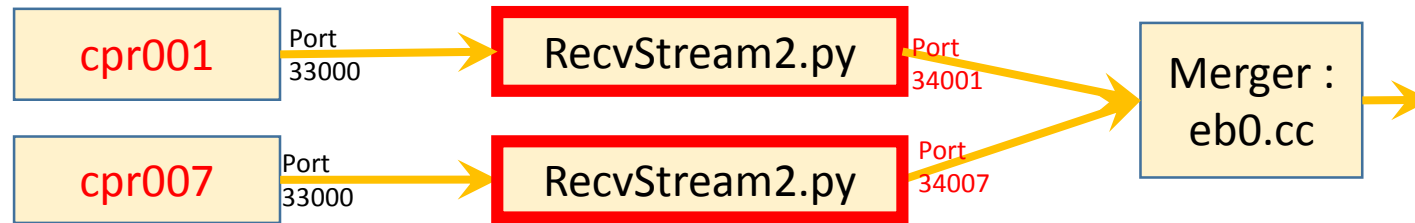
Set arguments of RecvStream2.sh

Usage : RecvStream2.py <COPPER hostname> <Use NSM(Network Shared Memory)? yes=1/no=0> <port # to accept connection from eb0 > <NSM nodename>

Example :
/usr/bin/xterm -e "${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts/RecvStream2.sh cpr001 0 34001 hogehoge1000;" &
/usr/bin/xterm -e "${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts/RecvStream2.sh cpr007 0 34007 hogehoge1000;" &

Readout PC

1, Set parameters in run_start.sh (4)

Specify whether communication with GUI via shared memory will be used or not :

1, Edit ${BELLE2_LOCAL_DIR}/daq/rawdata/examples/RecvStream1.py

If you use GUI, set 'UseShmFlag' = 1
receiver.param('UseShmFlag', 1)
If you don't use GUI set 'UseShmFlag' = 0
receiver.param('UseShmFlag', 0)

2, Edit ${BELLE2_LOCAL_DIR}/daq/rawdata/examples/RecvStream2.py

If you use GUI, set 'UseShmFlag' = 1
receiver.param('UseShmFlag', 1)
If you don't use GUI set 'UseShmFlag' = 0
receiver.param('UseShmFlag', 0)

2, Setting of merger (eb0) (1)

1, compile
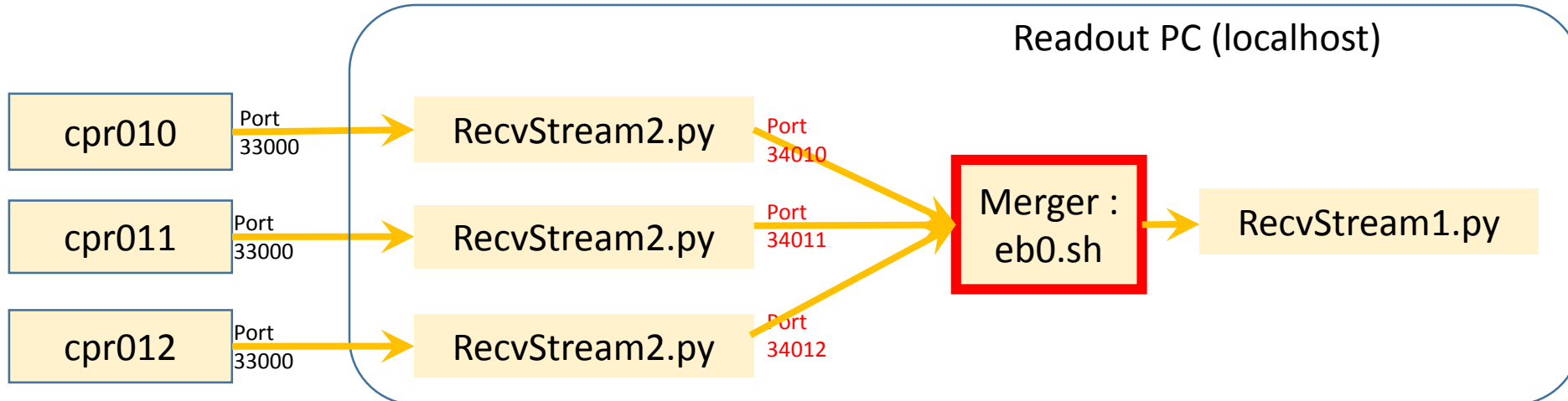$ cd ${BELLE2_LOCAL_DIR}/daq/eventbuilder/evb0
$ make

2, edit eb0.sh
cd /home/usr/b2daq/eb # change to cd "your local BelleII directroy"/daq/eventbuilder/evb0/eb-xinetd
(**e.g.** cd /home/usr/yamada/basf2/release/daq/eventbuilder/evb0/eb-xinetd)

If you want to read data from COPPERs with hostnames of cpr010, cpr011 and cpr012 and corresponding port # of RecvStream2.py are 34010, 34011 and 34012, respectively, use the following ;
~~./eb0 cpr010:33000 cpr011:33000 cpr012:33000~~
./eb0 localhost:34010 localhost:34011 localhost:34012

2, Setting of merger (eb0) (2)

3, edit eb0's part in ${BELLE2_LOCAL_DIR}/daq/eventbuilder/evb0/eb-xinted

Modify user and server.

```
service eb0
{        bind = 127.0.0.1
         socket_type = stream
         type = UNLISTED
         port = 5101
         wait = no
         user = g0cdc  # Change to your user name (e.g. yamagata)
         server = /home/usr/b2daq/eb/eb0.sh
         #  change to "your Belle2 local directory"/daq/eventbuilder/evb0/eb0.sh
         #                              (e.g. /home/usr/yamagata/basf2/release/daq/eventbuilder/evb0/eb0.sh)
}
```

You need to change the other services in eb-xinetd (eb1tx, eb1rx, eb2tx, eb2rx ….) in the same way.

2, Setting of merger (eb0) (3)

4, Set link to eb-xinetd
cd /etc/xinetd.d/
ln –s ${BELLE2_LOCAL_DIR}/daq/eventbuilder/evb0/eb-xinetd eb-xinetd
/sbin/service xinetd restart

The eb0 daemon will be automatically invoked when a basf2 program on a readout PC connects to a port specified in eb-xinted.

3, How to start DAQ

[ROPC] % cd ${BELLE2_LOCAL_DIR}/release/daq/copper/daq_scripts

[ROPC] % ./run_start.sh

It is convenient to login COPPER w/o typing password.
For example;
ropc $ cd ~/.ssh
ropc S ssh-keygen rsa
ropc $ cat ./id_dsa.pub >> authorized_keys
ropc $ chmod 600 authorized_keys

4, How to stop DAQ

- No stop button for now (If you use GUI, you have.)
- You need to specify max # of events or time to stop the run on a basf2 python file.

ROPC % cd ${BELLE2_LOCAL_DIR}/daq/rawdata/examples
Edit RecvStream1.py

You can set following paramters to stop a run.
receiver.param('MaxTime', 300.)
receiver.param('MaxEventNum', 30.)

-1 means inifinite.

5. Read an output file and extract FEE buffer

- Output file name
  - ${BELLE2_LOCAL_DIR}/daq/copper/daq_scripts/root_output.root
  - You can change the filename by editing ${BELLE2_LOCAL_DIR}/daq/rawdata/examples/RecvStream1.py
- Read the root file
  - Please see dataformat and unpacker manuals.
  - Data format
    - https://belle2.cc.kek.jp/svn/trunk/software/daq/copper/doc/SetupPocketDAQ_4p1_RawCOPPERDataHandling.pdf
  - Unpacker
    - https://belle2.cc.kek.jp/svn/trunk/software/daq/copper/doc/SetupPocketDAQ_4p5_RawDataUnpackerPacker.pdf

Revision history(1)

- 2013/10/16
  - Change explanation about option of run_start.sh script

- 2014/7/15 (rev. 11619)
  - To specify FINESSE bit field, you need to use decimal value (not hexadecimal).
  - Add an argument of RecvSenDCOPPER.py
  - Update eb0 related issues.
    - About eb0.sh
    - About eb0-xinted

- 2014/7/16(rev. 11654)
  - eb0.sh and eb-xinetd need more modification

- 2014/7/19
  - Add a comment about "addpkg daq"

- 2014/8/8
  - Modify usage of start_run.sh. (nodename is only used for NSM)

- 2014/10/29
  - Change a DAQ scheme so that RecvStream0.py is used on a readout PC

- 2014/11/10
  - Add figures to indicate connection between COPPER and FTSW
  - Add procedure to ssh-login w/o typing password

- 2014/11/14
  - Add "Notice from Konno-san about compiling daq" page

- 2015/1/22
  - Use RecvStream2.py(=DesSerPrePC.cc) instead of RecvStream0.py(=DeSerializerPrePC.cc+Serializer.cc)

Revision history(2)

- 2015 4/13
  - Add "0.5, Set active HSLB slots in TTRX register"

Appendix

A-1, yum setting for COPPER

Sometime yum fails when you try to install software to COPPER directory;

readoutPC # yum install **** --installroot=/tftpboot/copper/root

In that case, you need to change yum setting for COPPER host:

1, Copy yum repository files.
readoutPC # cp */etc/yum.repo.d/* /tftpboot/copper/root/etc/yum.repo.d/**

2, Set a proxy server in /tftpboot/copper/root/etc/yum.conf **same as** readoutPC:/etc/yum.conf

A-2, Socket parameters on COPPER, readout PC and etc..

1, Enable "Jumbo Frame" in network switch's setting, if necessary (e.g. HP1810-24G)
If the switch does not support Jumbo frame, setting MTU a large value may cause a problem.

2, Change socket parameters

/sbin/ifconfig eth? mtu 9000

/sbin/sysctl -w net.ipv4.tcp_rmem="8388608 8388608 8388608"
/sbin/sysctl -w net.ipv4.tcp_wmem="8388608 8388608 8388608"
/sbin/sysctl -w net.ipv4.tcp_mem="9000000 9000000 9000000"
/sbin/sysctl -w net.core.rmem_default=8388608
/sbin/sysctl -w net.core.wmem_default=8388608
/sbin/sysctl -w net.core.rmem_max=8388608
/sbin/sysctl -w net.core.wmem_max=8388608

3, edit /etc/rc.local(or /tftpboot/copper/root/etc/rc.local) if you want the parameters automatically set at boot time

A-3, Use ntpd to adjust time on COPPER

1, Edit /tftpboot/copper/root/etc/rc.d/init.d/ntpd

Source networking configuration.
. /etc/sysconfig/network

ntpdate 192.168.10.1

if [-f /etc/sysconfig/ntpd];then
      . /etc/sysconfig/ntpd
fi

2, check ntpd is started at boot time
cpr**** $ chkconfig –list | grep ntp

A-4, syslogd

Install syslogd :

 readoutPC # yum install syslogd --installroot=/tftpboot/copper/root

Add entry to /tftpboot/copper/root/etc/syslogd.conf if needed:

Log anything (except mail) of level info or higher.

Don't log private authentication messages!

*.info;mail.none;news.none;authpriv.none;cron.none   /var/log/messages

B-1, Test bench at Tsukuba B3