Status of RAON (with delight) Control Systems

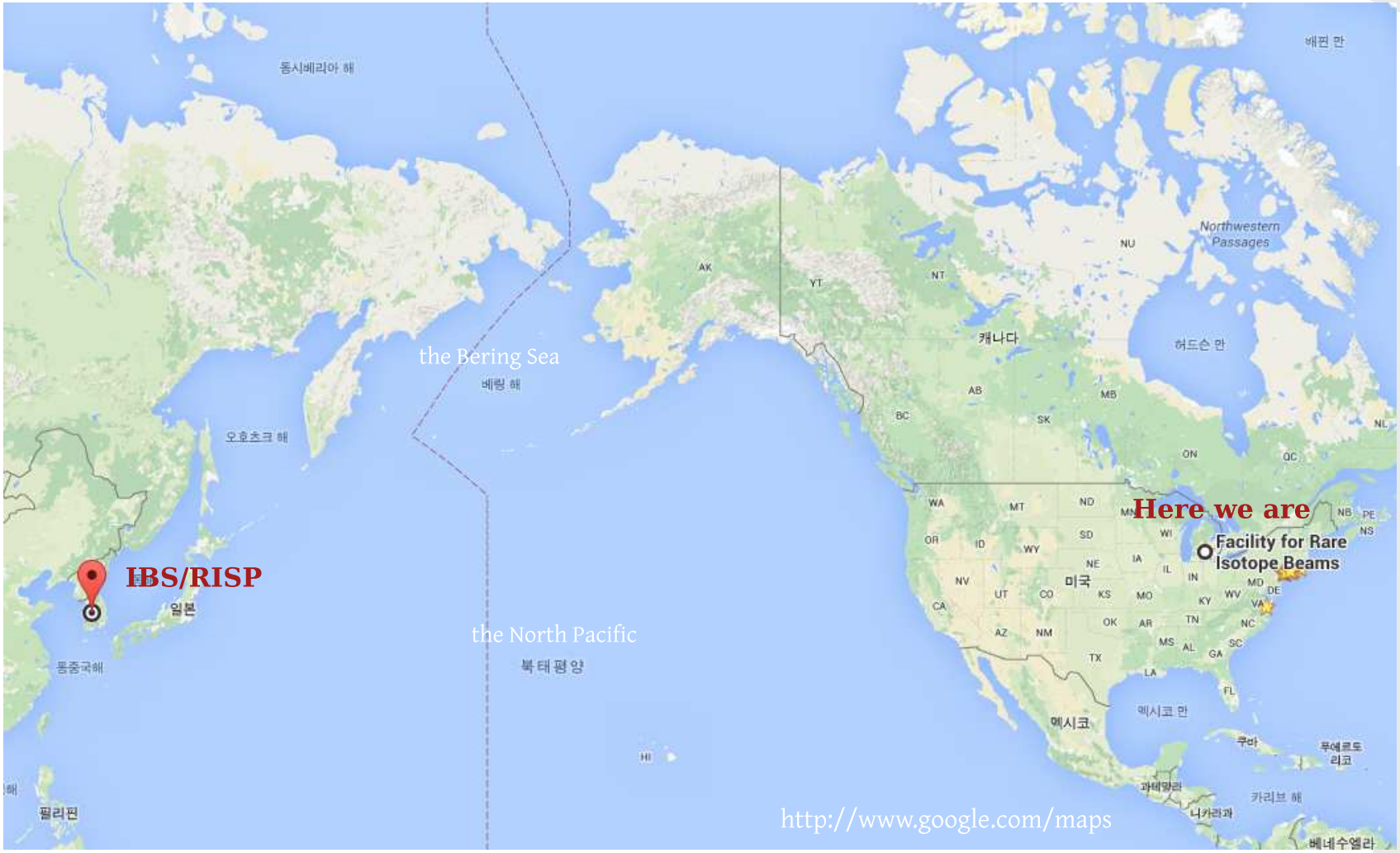
RAON is the name of the RISP accelerator

Jeong **Han** LEE

Rare **I**sotope **S**cience **P**roject
Institute for **B**asic **S**cience
South Korea

May 19, 2015

Rare Isotope Science Project (RISP) - Location

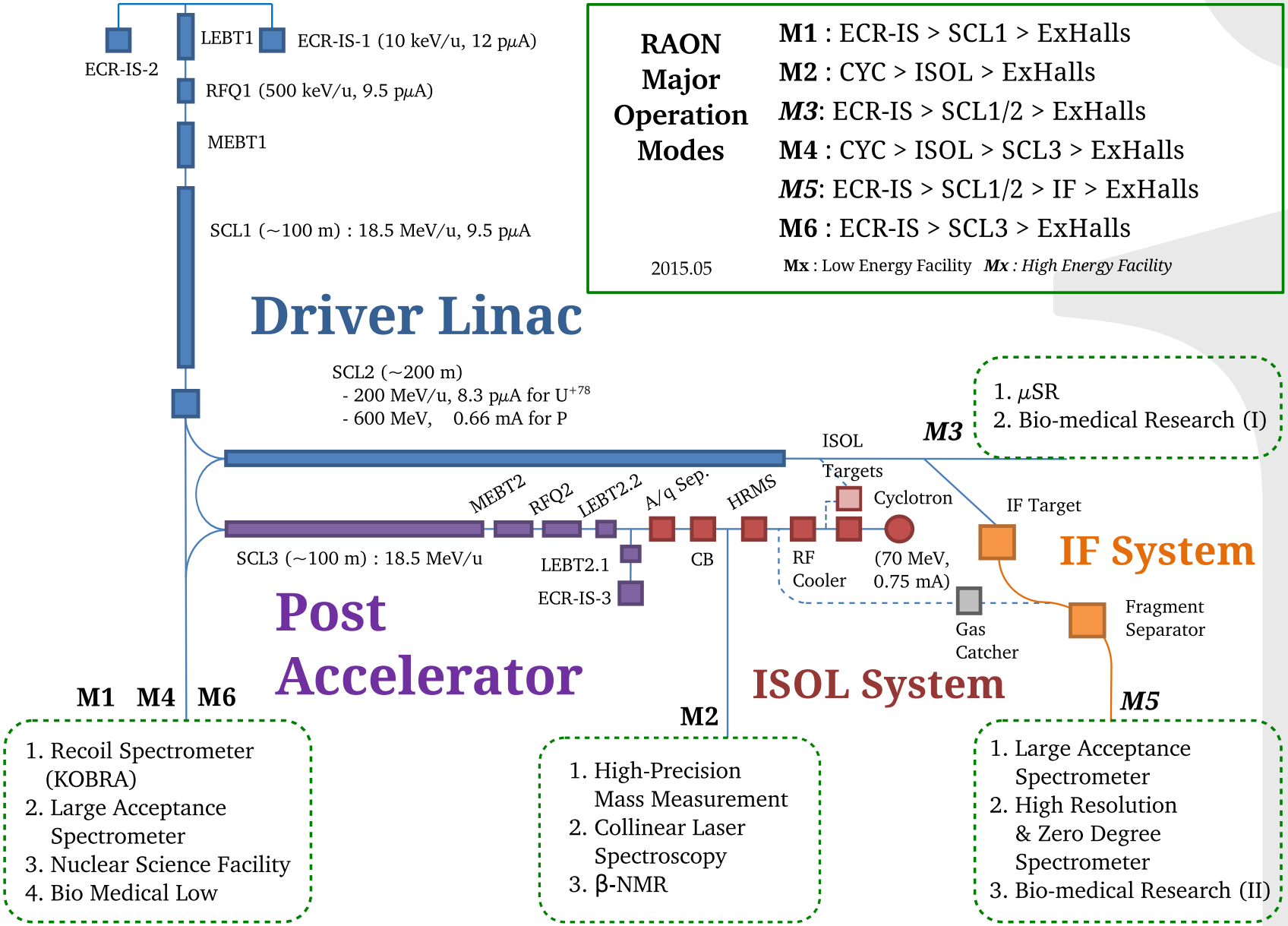


Rare Isotope Science Project (RISP) - Location

RISP - Aerial Conceptual View

RISP - Aerial Conceptual View

RISP accelerator - RAON - Operation Modes



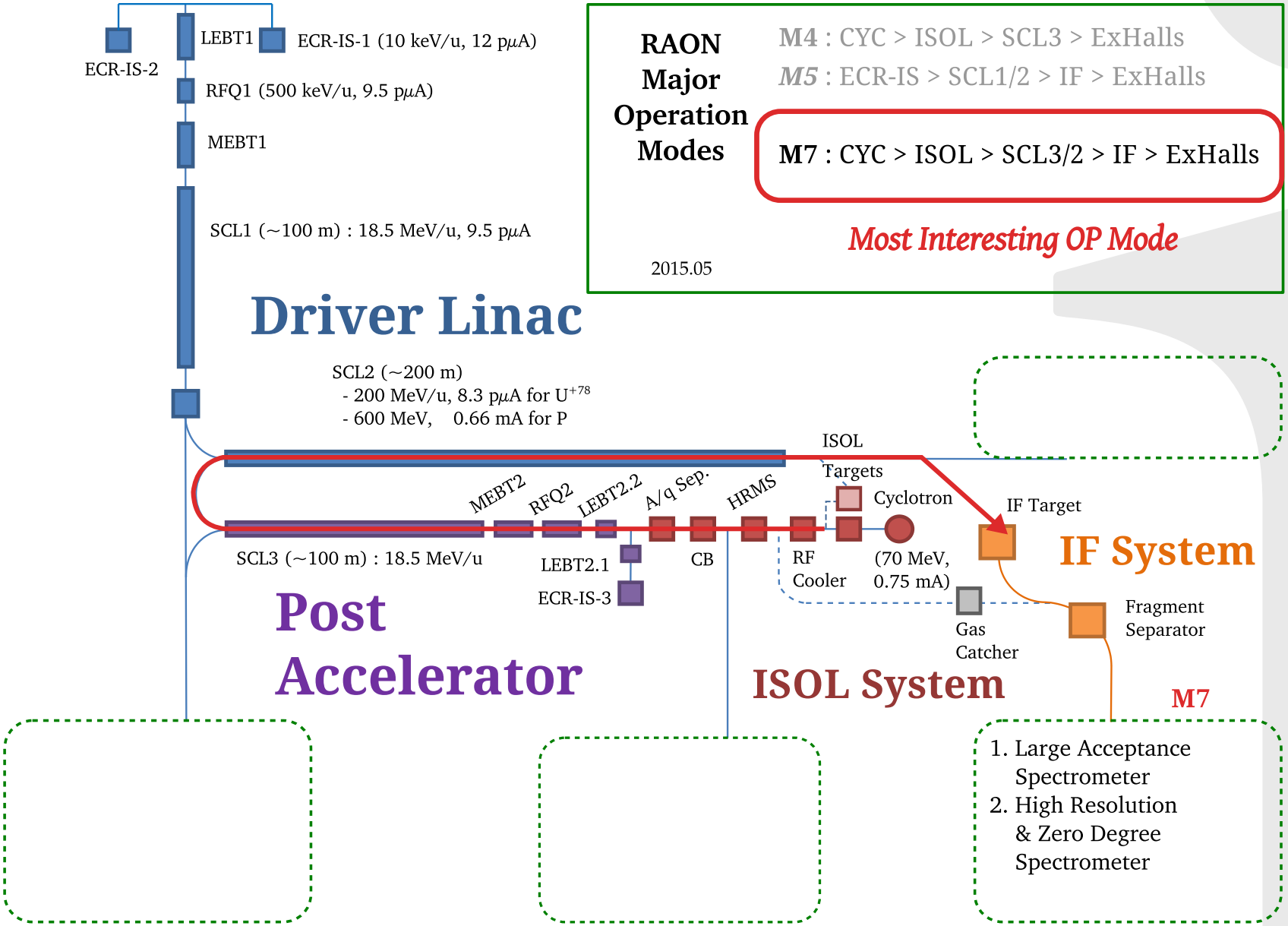
RAON Major Operation Modes

M1 : ECR-IS > SCL1 > ExHalls
M2 : CYC > ISOL > ExHalls
M3: ECR-IS > SCL1/2 > ExHalls
M4 : CYC > ISOL > SCL3 > ExHalls
M5: ECR-IS > SCL1/2 > IF > ExHalls
M6 : ECR-IS > SCL3 > ExHalls

2015.05

Mx : Low Energy Facility ***Mx*** : *High Energy Facility*

RISP accelerator - RAON - Operation Modes



RAON Major Operation Modes

M4 : CYC > ISOL > SCL3 > ExHalls

M5 : ECR-IS > SCL1/2 > IF > ExHalls

M7 : CYC > ISOL > SCL3/2 > IF > ExHalls

Most Interesting OP Mode

2015.05

RAON Control System Mission, Goals, & Members

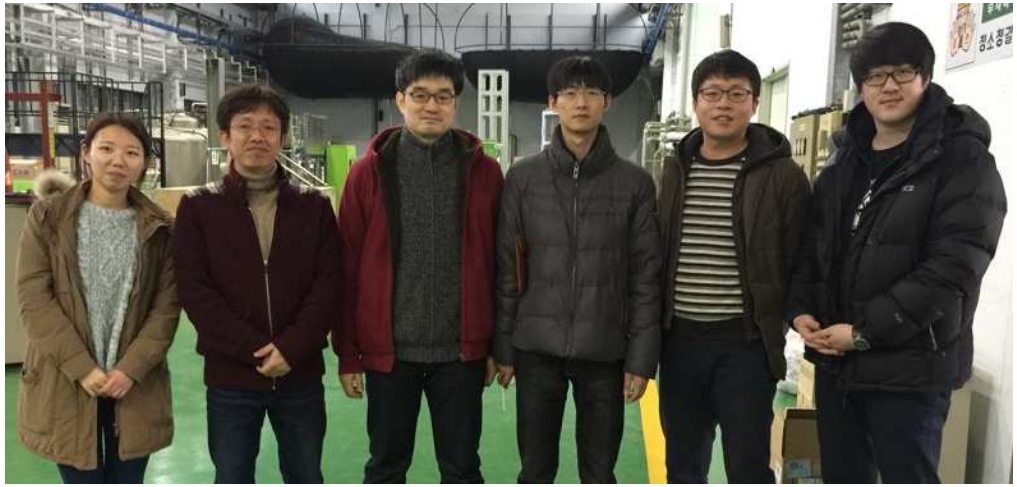
Mission

- Build reliable, usable, scalable, and efficient accelerator control systems for the RAON accelerator facility and its user community

Goals

- An efficient and conservative system in terms of budget, schedule, construction phase, and human resources
- EPICS integration of all possible signals from overall accelerator sub-systems
- Versatility in order to meet design requirements, which are operation modes, beam species, and beam energies, from user community and various science goals

Members



Mijeong Park Graduate Student	**Sang Il Lee** M.Sc. Researcher	**Jeong Han Lee** Dr.rer.nat. Research Fellow	**Chang Wook Son** M.Sc. Researcher	**Hyung Joo Son** M.Sc. Researcher	**Seung Hee Nam** Graduate Student

Control Members have

- Work Experiences at
  JLab (U.S.A.), MAMI, GSI (Germany), PAL, & KSTAR (S.Korea)

- Various Major Backgrounds
  * Experimental Nuclear/Hadron Physics
  * Electrical Engineering
  * Bio-medical Engineering
  * Control Engineering
  * Computer Science and Engineering

RAON Control System Fact

until 2015.05

- **EPICS**, All and Sundry systems ;-) (R3.14.12.5)
- Debian Linux 64bit for OS (Wheezy)
- PostgreSQL, MySQL/MariaDB for SW and configuration
- for sources & documents version control
- MRF EVG/EVR boards, MVME6100, MVME3100, VxWorks, and RTEMS for timing system
- Broadcade ICX 6430 and 6910 for Layer 2 Network Switch
- AB, Siemens, LSIS (S.Korea Domestic vendor) for PLC
- RTP3000
- CSS and KSTAR Widget Toolkit (QT based) for OPI
- Simple Network Management Protocol (SNMP) V2c (Read) and V3 (Write) for overall Ethernet-based devices
- Raspberry Pi (low cost) for monitoring low priority PVs

ONE Development Environment for control members

RAON EPICS Development Environment

can be ready for use by 'shell scripts' semi- automatically (x86, x64, arm)

```
jhlee@kaffee:~/epics$
jhlee@kaffee:~/epics$
jhlee@kaffee:~/epics$
jhlee@kaffee:~/epics$
jhlee@kaffee:~/epics$ tree -L 2
.
├── [jhlee    4.0K]  downloads
│   ├── [jhlee    488K]  alh1_2_34.tar.gz
│   ├── [jhlee    7.1M]  asyn4-24.tar.gz
│   ├── [jhlee    1.4M]  baseR3.14.12.5-rc1.tar.gz
│   ├── [jhlee    1.4M]  baseR3.14.12.5.tar.gz
│   ├── [jhlee     15K]  busy_R1-6-1.tar.gz
│   ├── [jhlee     37K]  cau_20130110.tar.gz
│   ├── [jhlee     15K]  dbVerbose_20130124.tar.gz
│   ├── [jhlee    7.4M]  EPICS-CPP-4.4.0.tar.gz
│   ├── [jhlee     12K]  extensionsTop_20120904.tar.gz
│   ├── [jhlee    1.8M]  gateway2_0_4_0.tar.gz
│   ├── [jhlee     94K]  gnuregex0_13.tar.gz
│   ├── [jhlee    131K]  ipac-2.13.tar.gz
│   ├── [jhlee    1.4M]  medm3_1_9.tar.gz
│   ├── [jhlee    2.7M]  motorR6-9.tar.gz
│   ├── [jhlee     12K]  msi1-6.tar.gz
│   ├── [jhlee     42K]  nameserver2_0_0_12.tar.gz
│   ├── [jhlee     26K]  probe1_1_8_0.tar.gz
│   ├── [jhlee    299K]  seq-2.1.16.tar.gz
│   ├── [jhlee    264K]  StripTool2_5_16_0.tar.gz
│   ├── [jhlee    172M]  synApps_5_7.tar.gz
│   ├── [jhlee    139M]  synApps_5_8.tar.gz
│   └── [jhlee    3.1M]  VisualDCT-dist-2.6.1274.zip
└── [jhlee    4.0K]  R3.14.12.5
    ├── [jhlee    4.0K]  base
    ├── [jhlee    4.0K]  epicsLibs
    ├── [jhlee    4.0K]  extensions
    ├── [jhlee    1.5K]  setEpicsEnv.sh
    ├── [jhlee    1.5K]  setEpicsEnv.sh~
    ├── [jhlee    4.0K]  siteApps
    └── [jhlee    4.0K]  siteLibs

7 directories, 24 files
jhlee@kaffee:~/epics$
jhlee@kaffee:~/epics$
jhlee@kaffee:~/epics$
jhlee@kaffee:~/epics$
```

Lv0. download

Lv0. epics version

- Lv1. base : EPICS base
- Lv1. epicsLibs : synApps, and others EPICS Libs
- Lv1. extensions : EPICS extensions
- Lv1. siteApps : RAON specific EPICS apps
- Lv1. siteLibs : RAON EPICS Libs
- Lv1. setEpicsEnv.sh : Dynamic Env Setup Script

This is the quick-and-dirty approach, but the cost-effective way for us.
Note that we want to move the 100% Debian packaging SoOoOoOoON~~~

RAON EPICS Development Environment

can be ready for use by 'shell scripts' semi- automatically (x86, x64, arm)

```
├── [jhlee    4.0K]  sorensenPS
├── [jhlee    4.0K]  srs725
├── [jhlee    4.0K]  Tr1
├── [jhlee    4.0K]  Tr2
├── [jhlee    4.0K]  Tr3
├── [jhlee    4.0K]  Tr4
├── [jhlee    4.0K]  Tr5
└── [jhlee    4.0K]  Tr6

24 directories, 1 file
jhlee@kaffee:~/epics/R3.14.12.5/siteApps$
jhlee@kaffee:~/epics/R3.14.12.5/siteApps$
jhlee@kaffee:~/epics/R3.14.12.5/siteApps$
jhlee@kaffee:~/epics/R3.14.12.5/siteApps$ goLibs
jhlee@kaffee:~/epics/R3.14.12.5/siteLibs$ tree -L 1
.
├── [jhlee    4.0K]  bin
├── [jhlee    4.0K]  configure
├── [jhlee    4.0K]  db
├── [jhlee    4.0K]  dbd
├── [jhlee    4.0K]  documentation
├── [jhlee    4.0K]  ether_ipLib
├── [jhlee    4.0K]  glassManPSLib
├── [jhlee    4.0K]  ifstatLib
├── [jhlee    4.0K]  include
├── [jhlee    4.0K]  lib
├── [jhlee     186]  Makefile
├── [jhlee     345]  Makefile.template
├── [jhlee    4.0K]  rdbpgLib
├── [jhlee    4.0K]  README
├── [jhlee    1.3K]  README.siteLibs
├── [jhlee    4.0K]  RPiLibPack
├── [jhlee    4.0K]  rtpLib
├── [jhlee    4.0K]  s7plcLib
├── [jhlee    4.0K]  snmpLib
├── [jhlee    4.0K]  snmpMSULib
├── [jhlee    4.0K]  sysMonLib
├── [jhlee    4.0K]  TC353LibSrc
├── [jhlee    4.0K]  timestampLib
└── [jhlee    4.0K]  XGS600LibSrc

21 directories, 3 files
jhlee@kaffee:~/epics/R3.14.12.5/siteLibs$
```

RAON_SITELIBS : epics/versions/siteLibs

Lv0. db,dbd,lib,include
use only RAON_SITELIBS to make any EPICS Apps

Lv0. Various Developing RAON Specific EPICS Libs

Lv0. One Makefile
.....

maintained through *github.com*

This is the quick-and-dirty approach, but the cost-effective way for us.
Note that we want to move the 100% Debian packaging SoOoOoOoON~~~

RAON EPICS Development Environment

can be ready for use by 'shell scripts' semi- automatically (x86, x64, arm)

```
jhlee@kaffee:~$
jhlee@kaffee:~$
jhlee@kaffee:~$
jhlee@kaffee:~$
jhlee@kaffee:~$
jhlee@kaffee:~$
jhlee@kaffee:~$
jhlee@kaffee:~$
jhlee@kaffee:~$ goApps
jhlee@kaffee:~/epics/R3.14.12.5/siteApps$
jhlee@kaffee:~/epics/R3.14.12.5/siteApps$
jhlee@kaffee:~/epics/R3.14.12.5/siteApps$
jhlee@kaffee:~/epics/R3.14.12.5/siteApps$
jhlee@kaffee:~/epics/R3.14.12.5/siteApps$ tree -L 1
.
├── [jhlee    4.0K]  bin
├── [jhlee    4.0K]  exAsynRecord
├── [jhlee    4.0K]  exSysMon
├── [jhlee    4.0K]  exTimeStamp
├── [jhlee    4.0K]  exTr5
├── [jhlee    4.0K]  glassManPS
├── [jhlee    4.0K]  grackApps
├── [jhlee    4.0K]  keithley2kdmm
├── [jhlee    4.0K]  keithley6514
├── [jhlee    4.0K]  modbusR2-7
├── [jhlee    4.0K]  raspberry
├── [jhlee    4.0K]  rdbPG
├── [jhlee    3.3K]  README.md
├── [jhlee    4.0K]  rtp
├── [jhlee    4.0K]  snmp2
├── [jhlee    4.0K]  snmpMSU
├── [jhlee    4.0K]  snmpTest
├── [jhlee    4.0K]  sorensenPS
├── [jhlee    4.0K]  srs725
├── [jhlee    4.0K]  Tr1
├── [jhlee    4.0K]  Tr2
├── [jhlee    4.0K]  Tr3
├── [jhlee    4.0K]  Tr4
├── [jhlee    4.0K]  Tr5
└── [jhlee    4.0K]  Tr6

24 directories, 1 file
jhlee@kaffee:~/epics/R3.14.12.5/siteApps$
```

RAON_SITEAPPS : epics/versions/siteApps

Lv0. bin : raon_ioc_bash

> start the selected IOC within the SCREEN session,

- if the IOC is already running, attach to it
- if no IOC is running, create and attach to it

Lv0. EPICS Apps Examples

Lv0. Developing EPICS Applications

.....

Lv0. Internal Traning EPICS IOCs

.....

maintained through *github.com*

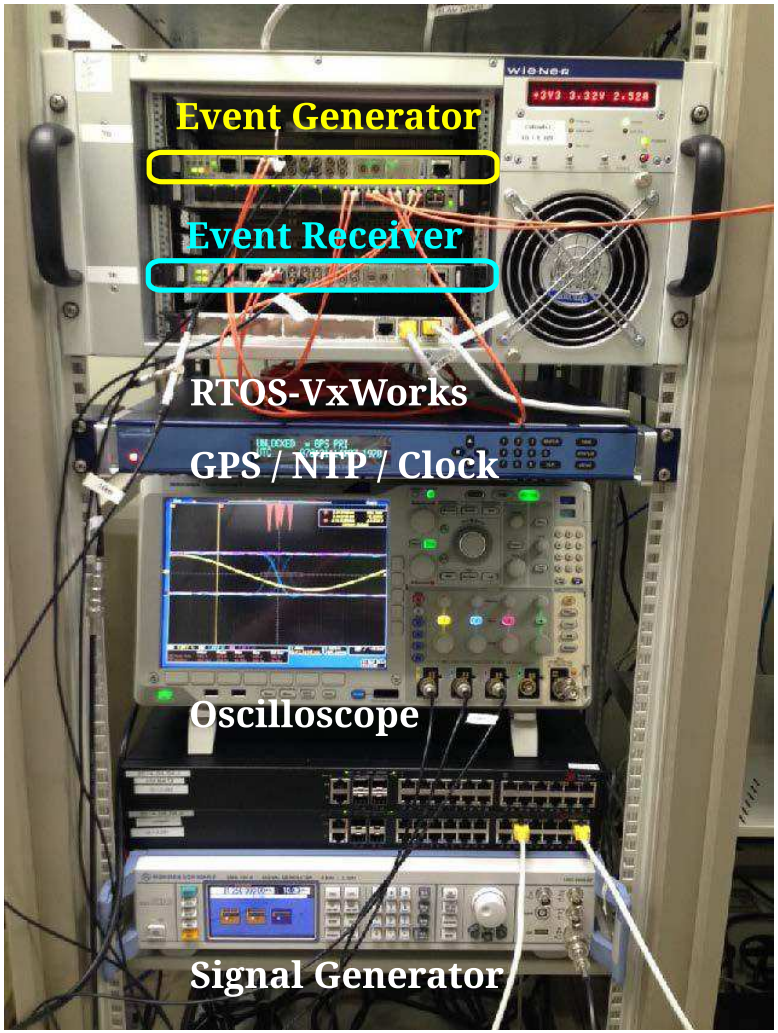
This is the quick-and-dirty approach, but the cost-effective way for us.
Note that we want to move the 100% Debian packaging SoOoOoOoON~~~

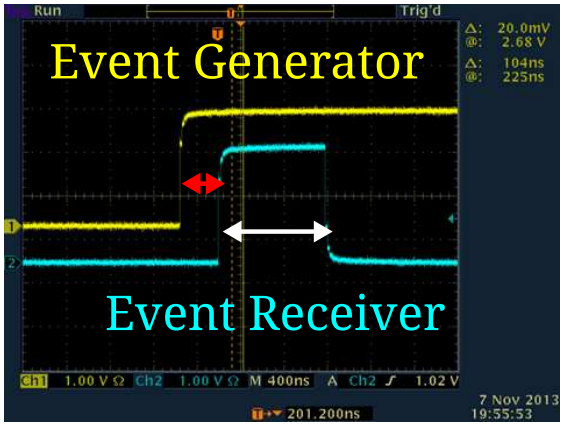
Scene - where we burn ourselves out

Control Test Room

Test Racks for Ctrl & Timing

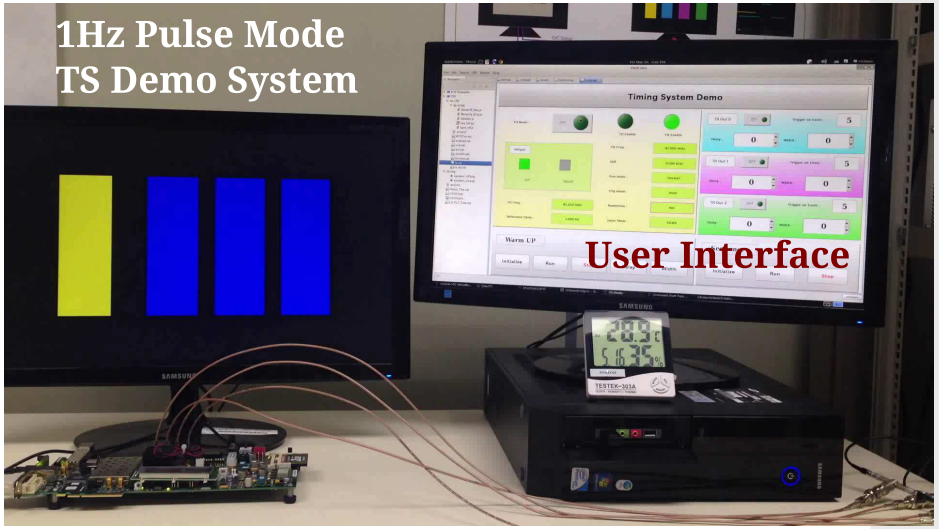
Timing System



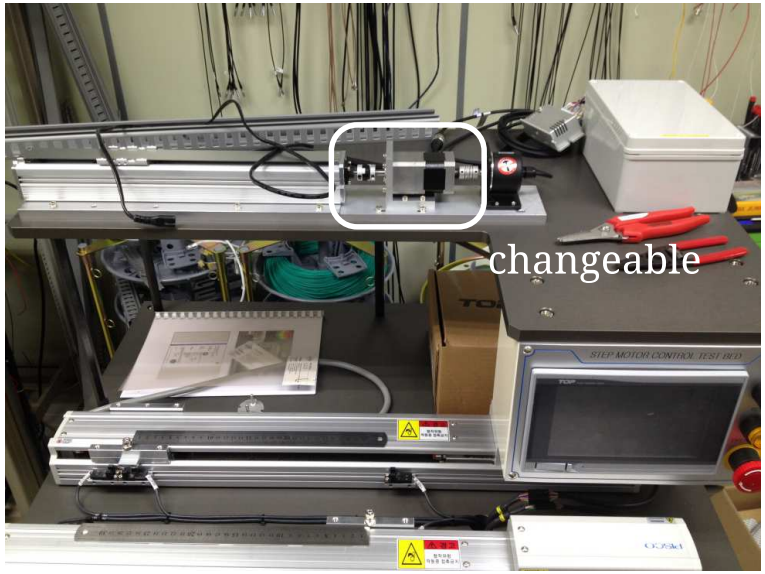


Programmable Time Delay

Programmable Time Width



Development on RAON control system



Stepper Motor Testbeds

- can test 3 types of Stepper Motor (encoder, spindle, motor)
- based on LSIS PLC + touch pad
- EPICS integration in progress

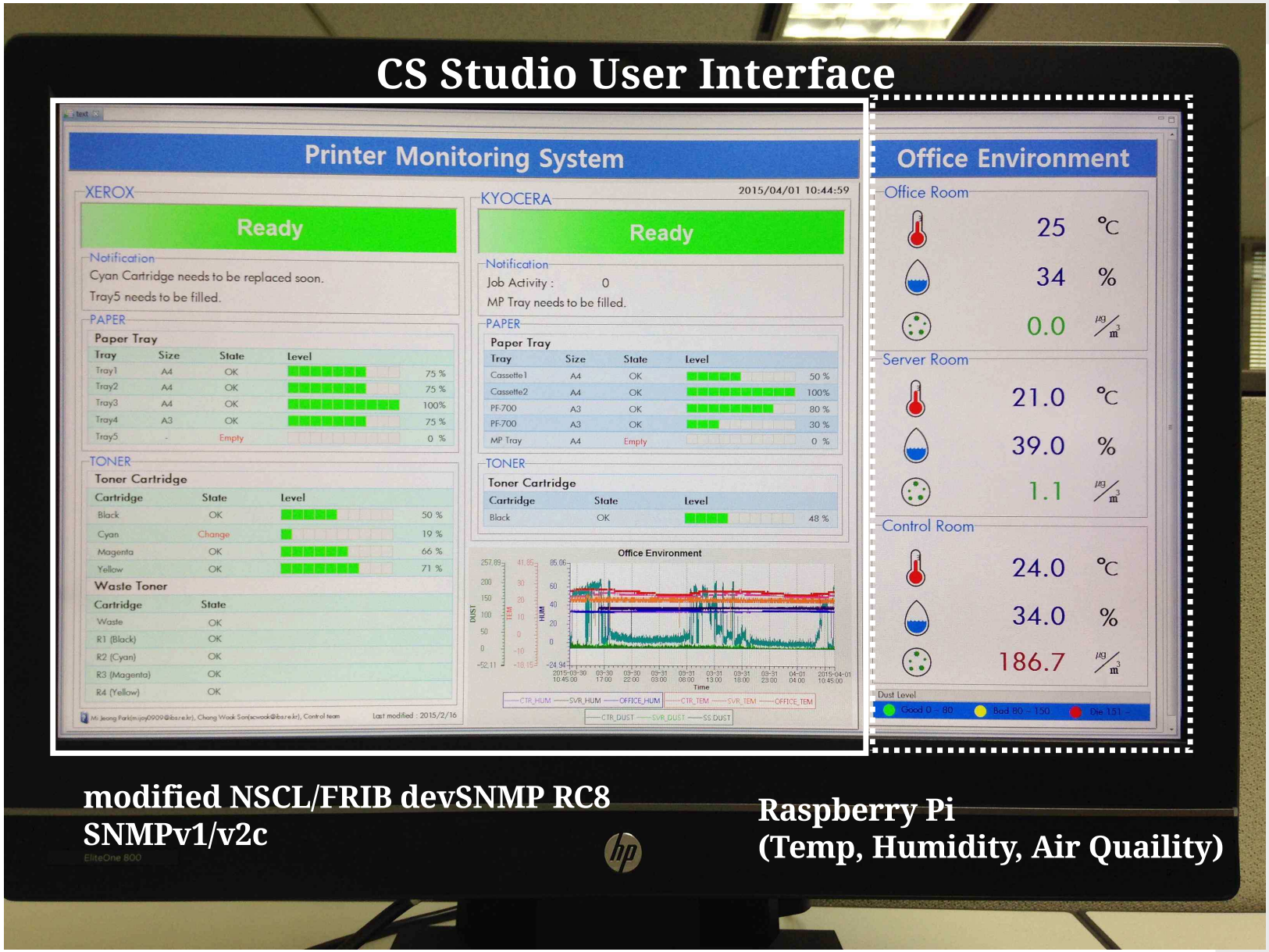
A Stepper Motor Control Box

for leak valve control

- Raspberry PI B+(R.Pi) as EPICS IOC (Ethernet to R.Pi) : Rev 0
- So far, no issue yet, except Ethernet connection loss
- R.Pi2 as EPICS IOC
  > Ethernet to R.Pi
  > Direct Serial2Optical to R.PI in progress
  : can control "a motor" when we lose the Ethernet connection

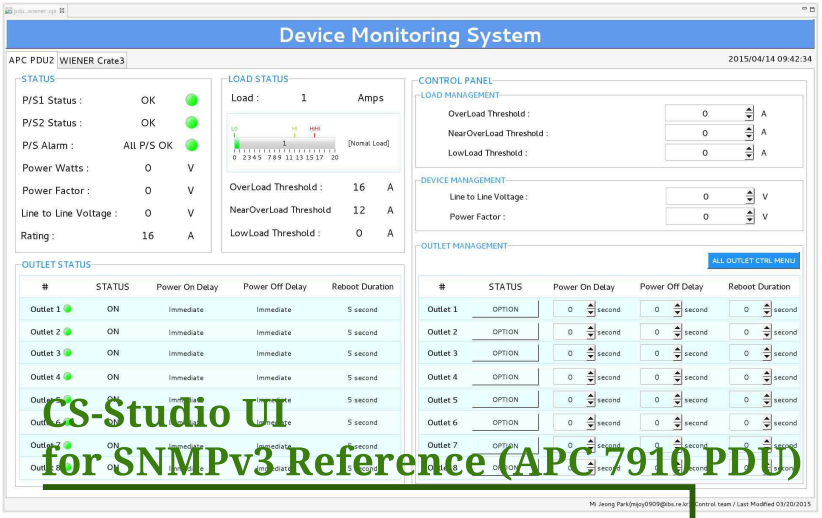
Development on RAON control system

EPICS monitoring system for Office Environment

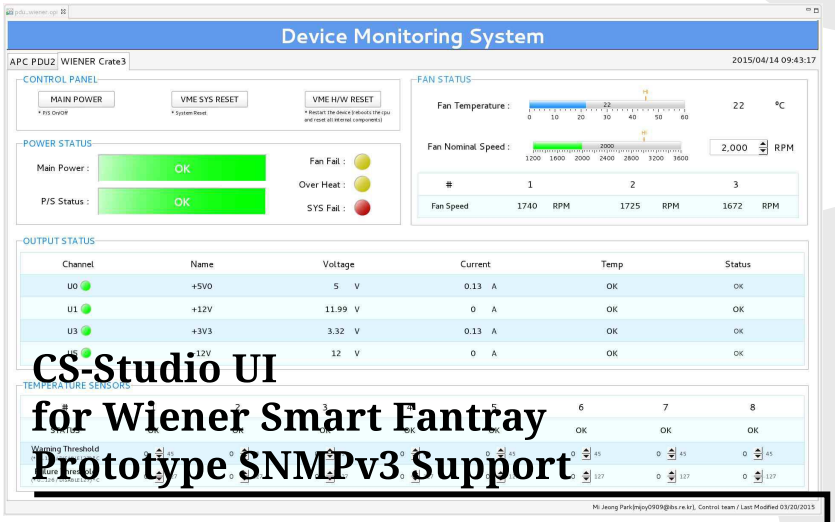


Development on RAON control system

RAON EPICS integration of SNMPv3 - independent upon devSNMP

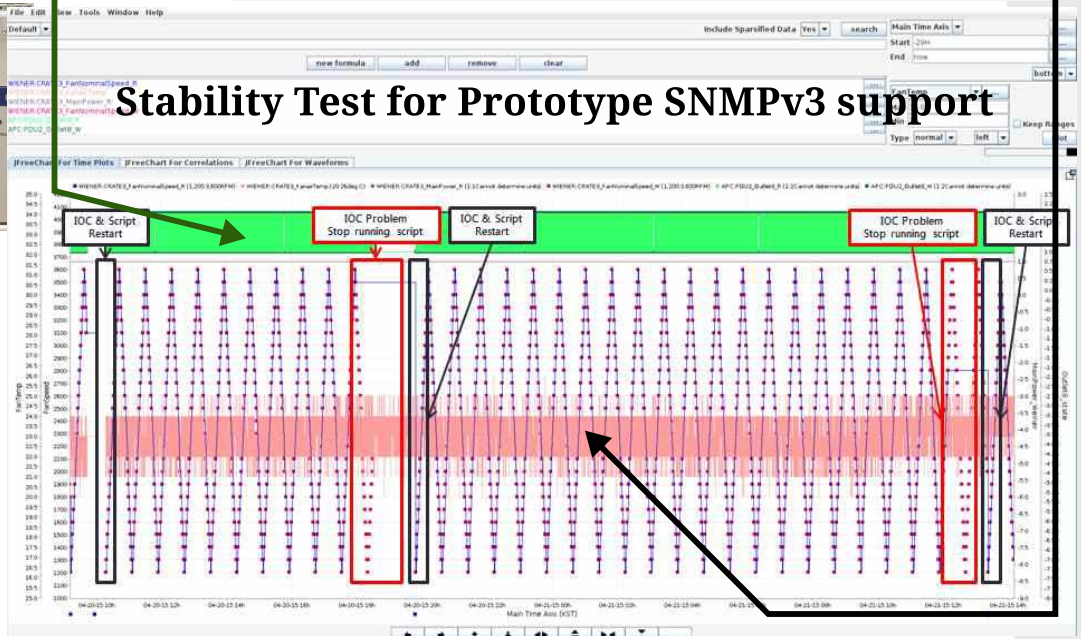


CS-Studio UI
for SNMPv3 Reference (APC 7910 PDU)



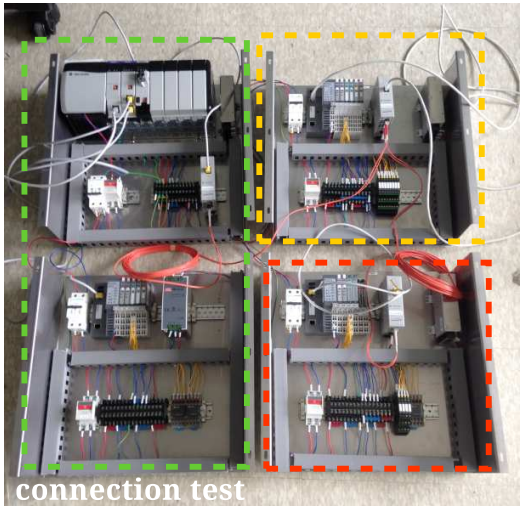
CS-Studio UI
for Wiener Smart Fantray
Prototype SNMPv3 Support

- work well with APC PDU
- asked the Wiener-d to support SNMPv3 a year ago
- delivered the prototype fantray
- testing it under EPICS+SNMPv3
- **issue : no response from Fanspeed (R/W). Still unknown where it comes from**



Stability Test for Prototype SNMPv3 support

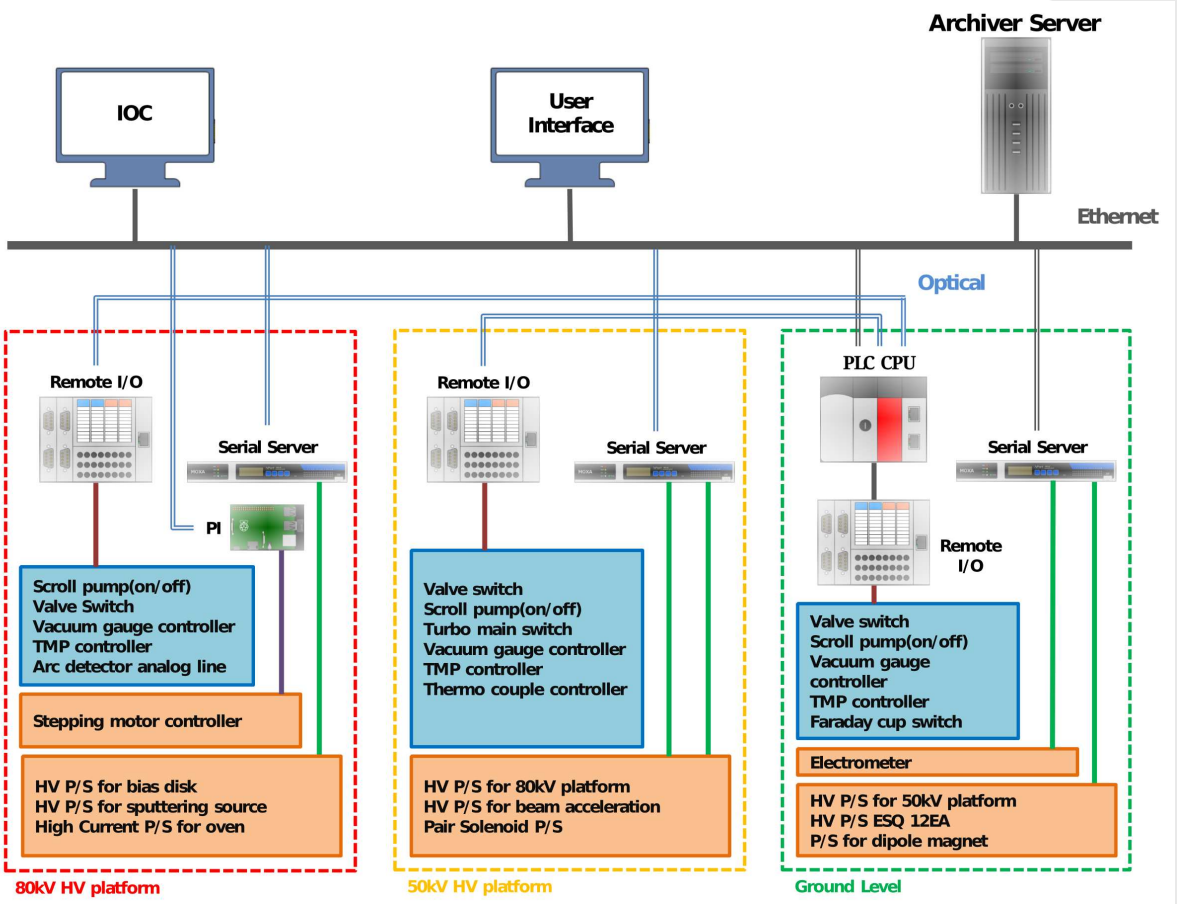
Development on RAON control system



ECR-IS Control System Works in progress

EPICS IOCs for Others

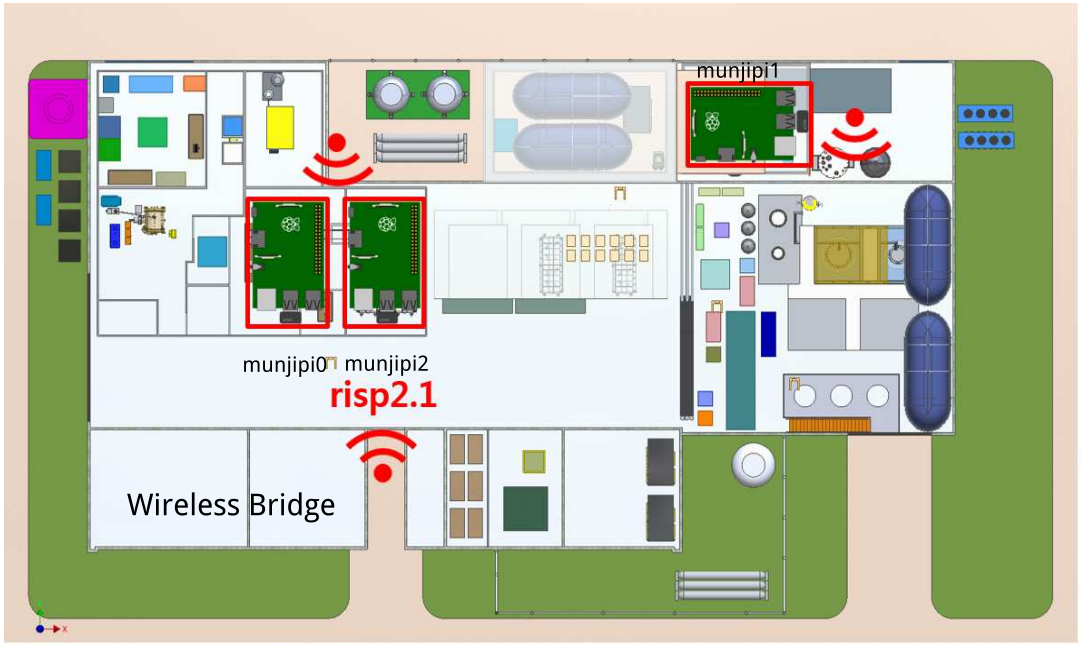
AB PLC for Vacuum

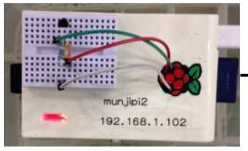


Development on RAON control system

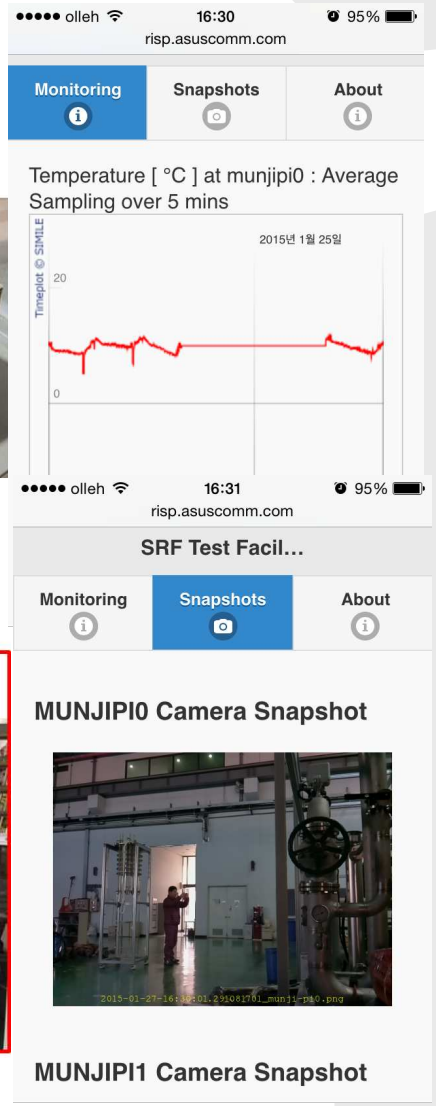
Environment Monitoring System at SRF test facility

Raspberry Pi, EPICS, Archiver Appliance, jQuery, web monitoring site

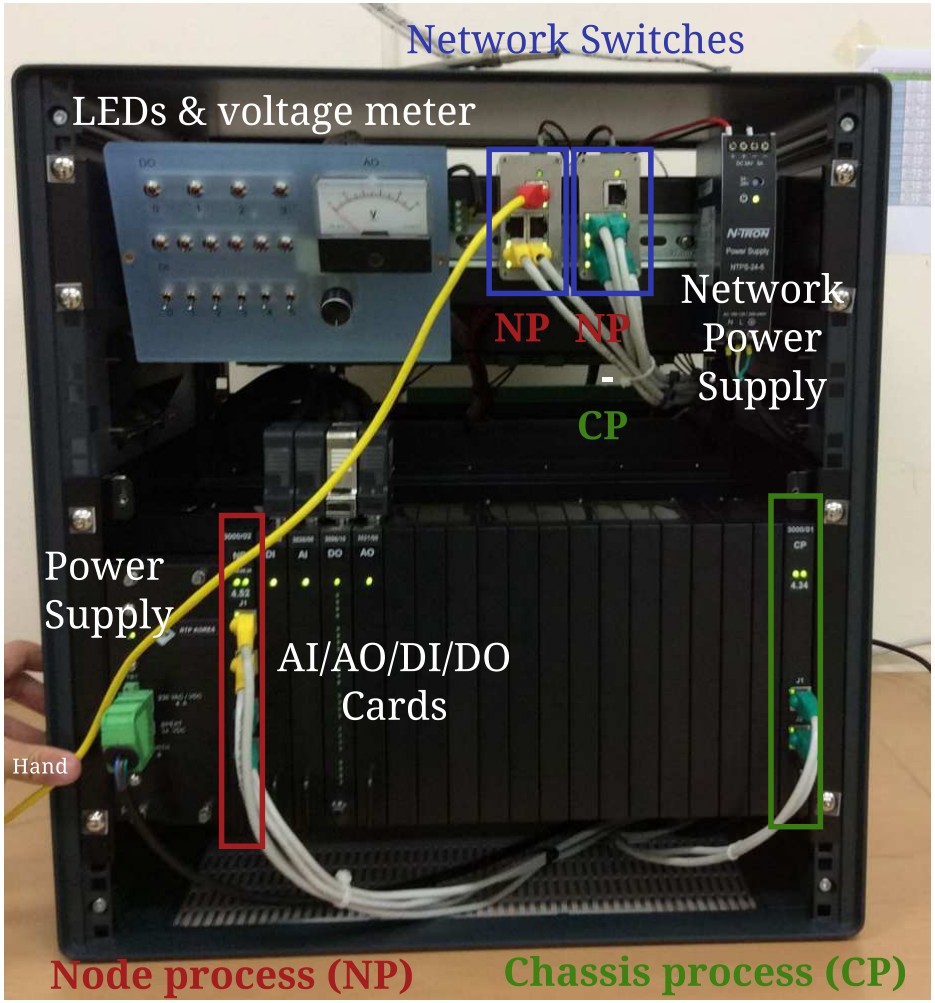




Archiver Appliance Server



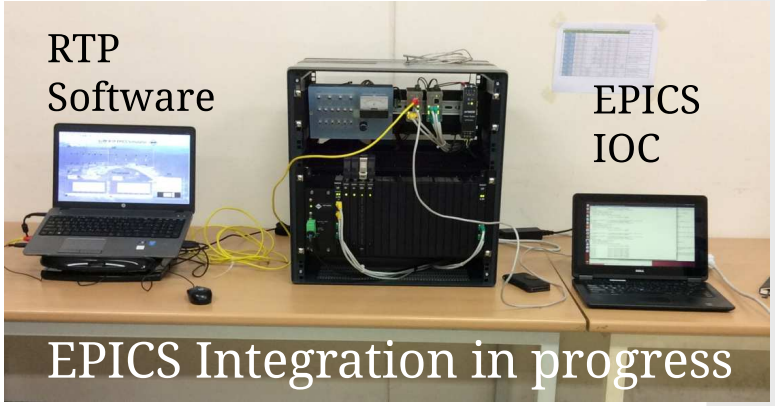
RTP3000 Testbed - Dual Redundant



Nucleus RTOS

Components

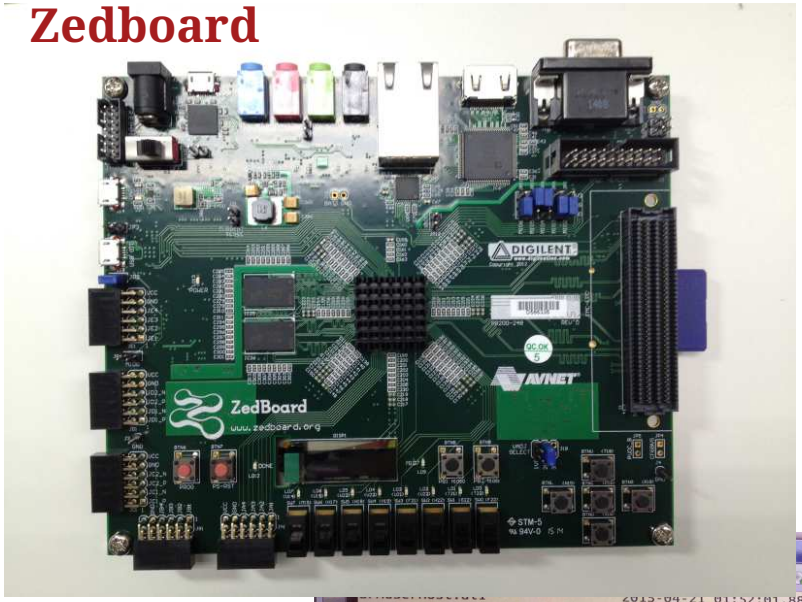
- Node & Chassis Processors
- Power Supplies
- AI, DI, DO, AO cards
- AI convertor (backplane)
- Thermo couple sensor (backplane)
- DI LED Lamps
- DO switch & LED lamps
- AI voltage meter



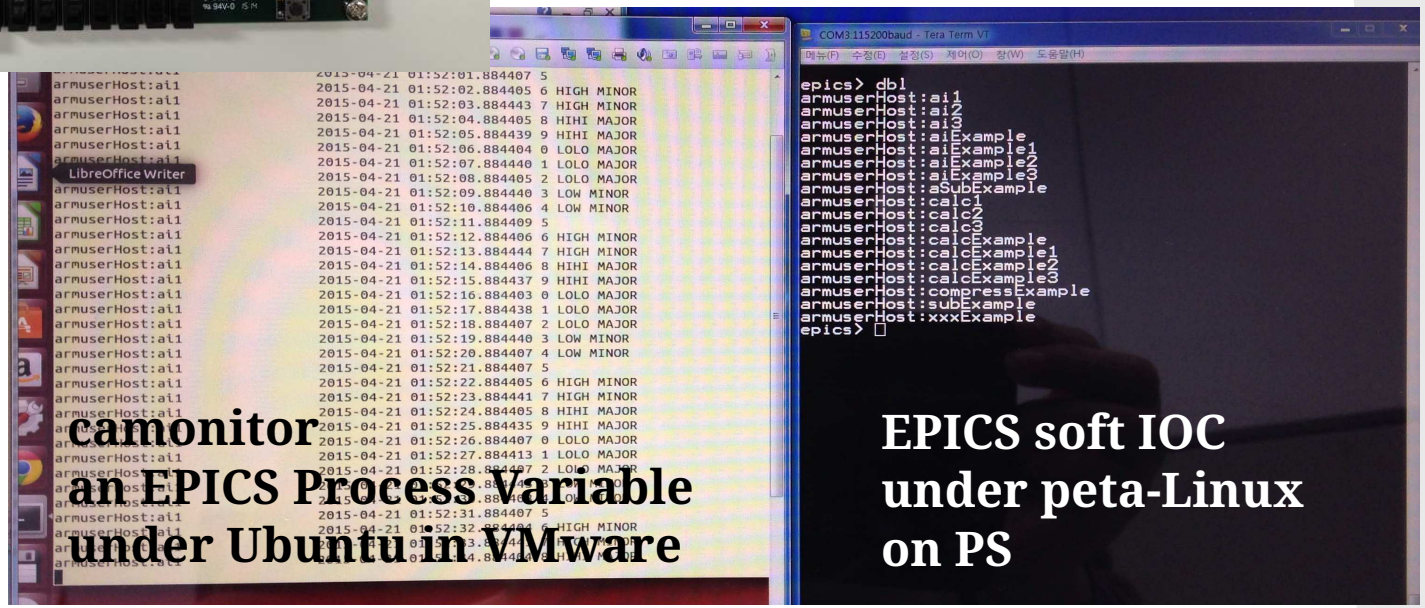
Warm-up studies on RAON control system

ZynQ : Processing System (PS) with Programmable Logic (Xilinx FPGA)

Zedboard



- attend ZynQ trainings
- run EPICS soft IOC on PS (Dual core ARM)
- plan to connect with Stepper motor testbed and the timing system (ZC706 Evaluation kit has SFP/SFP+ module connector)
- plan to extend this study to BPM electronics Digital Frontend and FPGA communication logic development

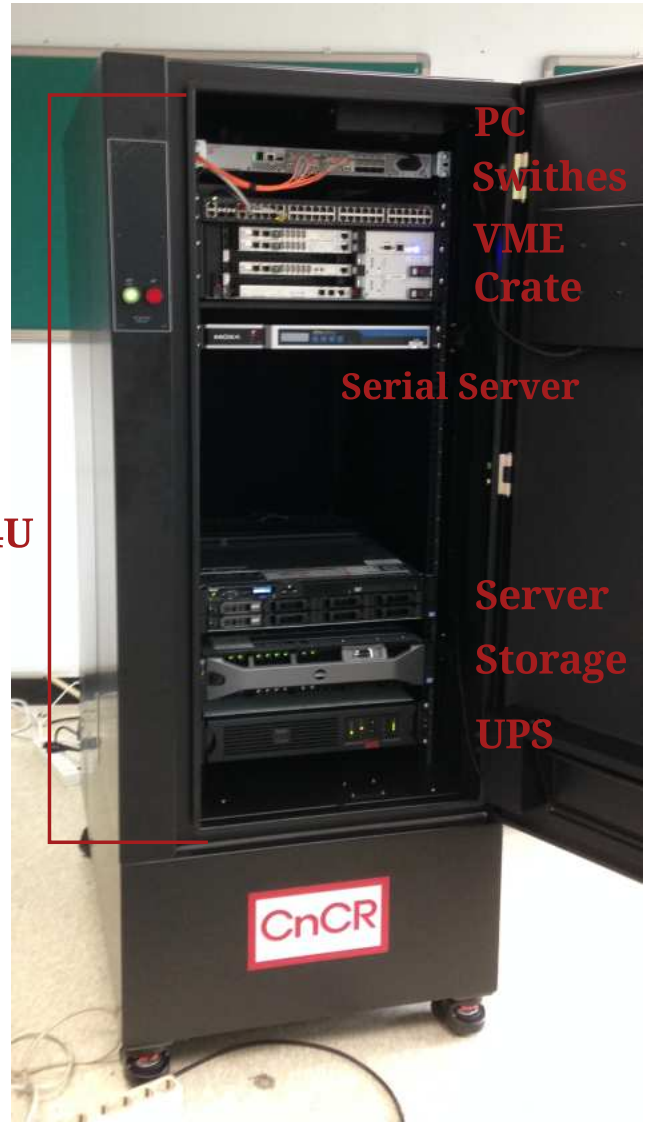


Warm-up studies on RAON control system

Temperature Control Rack : CnCR

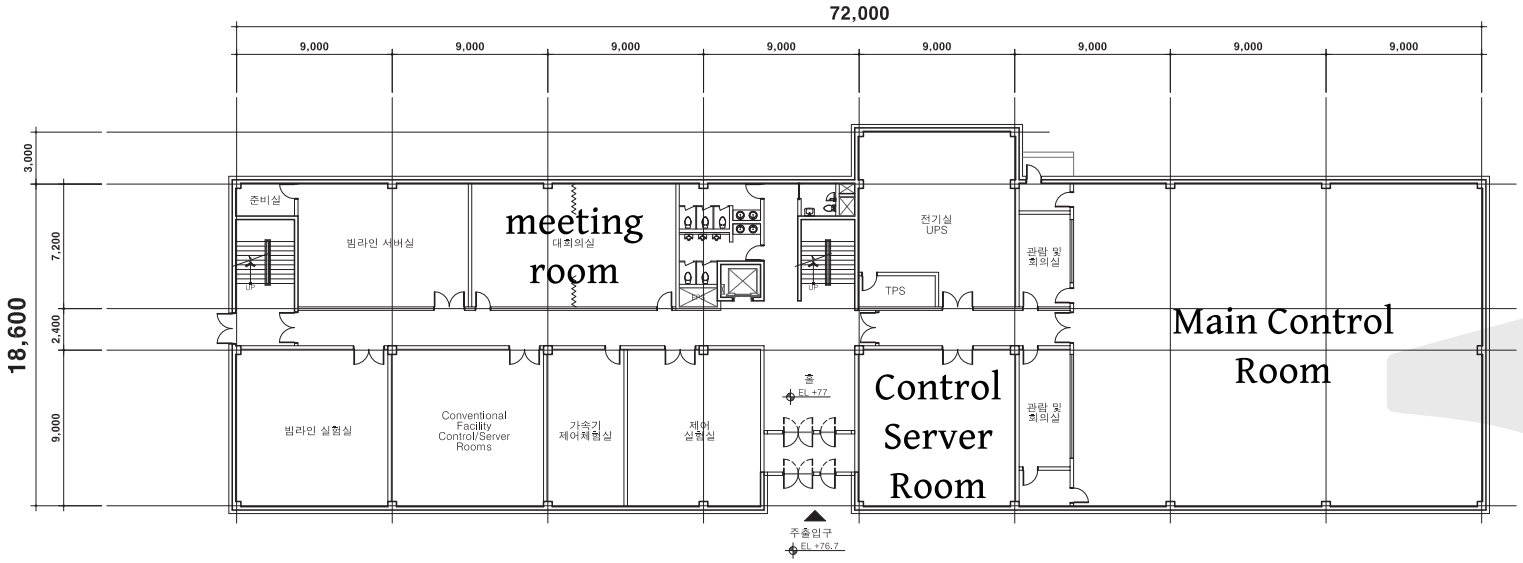
Admin Web Site

24U

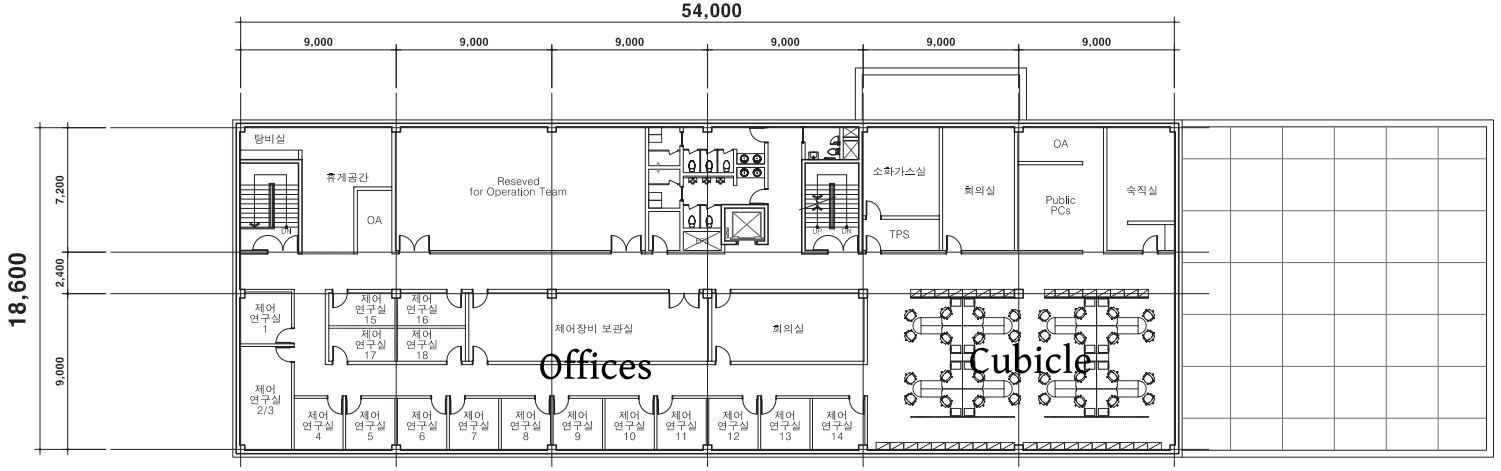


- Temperature control within $\pm 0.1°C$ can be archived by the vendor. However, we want to test it.
- Planning for Temp. control test
- The main controller is a Linux PC (CentOS 6 with JAVA)
- EPICS integration within a Debian Linux PC in progress
- proposed the vendor CnCR to an Ethernet Port with SNMP support instead of a Linux PC
- 42 U rack estimated cost ~ 12,000 k KRW

Main Control Center : MCC



Main Control Center Floor Plan 2015.4.



SCALE : 1/300

Summary and Outlook

- We are laying the slow and steady groundwork for EPICS integrated RAON control system.
- Still, we know, it is the big challenge to build the system so as to fulfill various operation modes.
- And definitely, there are many subjects that we can collaborate together within any forms.
- We would welcome your advice, critic, comment, suggestion, possible collaborated work with open arms.
- My email is `jhlee@ibs.re.kr` and `jeonghan.lee@gmail.com`

감사합니다!

Thank you!

Dankeschön!

謝謝!

¡Gracias!

Merci!

ありがとう!

☺