

# EPICS와 SNMP 통합

박미정\*

Rare Isotope Science Project  
Institute for Basic Science, Daejeon, South Korea

April 9, 2015

## Abstract

본 문서는 중이온가속기 제어의 기본 Framework이 되는 EPICS와 IP 네트워크 상의 장비 관리 및 모니터링 프로토콜인 SNMP통합 및 모니터링 시스템의 구현에 대해 논한다.

## 1 중이온가속기 제어 시스템

가속기 제어 시스템은 사용자가 원하는 빔을 사용자가 원하는 장소로 효율적으로 보낼 수 있도록 가속기를 구성하는 모든 요소를 감시하며 원격으로 제어하는 장치 조직망이다. EPICS는 실시간 분산 제어 시스템이자 중이온가속기 제어의 기본 Framework로 중이온가속기 제어 시스템 개발에 사용된다.

### 1.1 중이온가속기

한국의 기초과학연구원(Institute for Basic Science) 산하 중이온가속기구축사업단(Rare Isotope Science Project)은 빔에너지 200MeV/u, 빔 출력 400kW급 희귀동위 원소 가속기 시설을 구축하고 있다. 중이온가속기구축사업단은 수소, 헬륨보다 무거운 지구 상의 모든 원소의 이온을 빛의 속도에 가깝게 가속하는 가속기를 구축하는 연구시설로, 중이온가속기는 전기장을 이용해 중이온(탄소, 우라늄 등)을 빠른 속도로 가속하며, 이렇게 가속된 입자들을 표적 물질에 충돌시켜 자연 상태에 존재하지 않는 다양한 희귀동위원소 생성에 이용하거나, 원자핵 등의 관찰, 물질의 성질 연구에 이용한다[1][2].

---

\*mijoy0909@ibs.re.kr

## 1.2 SNMP

SNMP (Simple Network Management Protocol)<sup>1</sup>는 IP 네트워크 상의 장치 및 장비들을 관리하고 모니터링하기 위한 인터넷 표준 프로토콜이다[3]. SNMP는 인증과 암호화 등의 차이로 v1/2c/3의 세 가지 버전으로 나뉘며, 그림 1과 같이 Manager/Agent 구조이다[4]. Manager는 Agent에게 원하는 장비의 정보를 요청하며, 장비의 설정을 변경한다. Agent는 Manager가 요청한 장비의 정보를 제공하고, 시스템 충돌이나 재부팅 등의 장비에 영향을 미치거나 발생한 Event를 비동기적으로 알리기 위해 Trap 메시지를 보낸다. OID (Object Identifiers) 객체로 구성된 MIB (Management Information Base)는 계층구조를 이루며, 장비의 정보를 내포하고 있다. 일반적인 TCP/IP 관리정보는 MIB-2(RFC 1213)에 포함돼있고, 특정 장비들의 정보는 장비제조업체에서 제공한다.

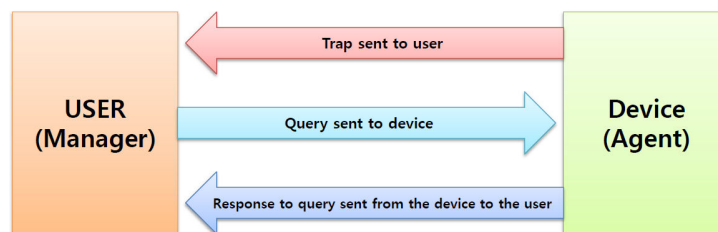


Figure 1 Manager와 Agent의 관계

## 1.3 EPICS

EPICS (Experimental Physics and Industrial Control System)는 Los Alamos 국립 연구소와 Argonne 국립 연구소에서 공동개발 되어 오픈 라이선스로 제공되는 실시간 분산 제어 시스템으로 네트워크로 연결된 다양한 장비들의 모니터링 및 제어를 위해 사용된다. EPICS는 현재 전 세계 과학시설의 개발자들에 의해 개발이 진행되고 있다. EPICS는 그림 2의 EPICS 로고와 같이 네트워크 기반의 클라이언트/서버 구조이며, TCP/UDP 프로토콜을 사용하는 CA (Channel Access) 통신 프로토콜을 사용한다. 클라이언트는 CSS, MEDM, StripTool 등의 다양한 프로그램을 사용해 모니터링 및 제어를 요청하고, 서버는 PV Database, StreamDevice, Modbus 등의 IOC (Input Output Controllers) 소프트웨어와 PV (Process Variable)를 통해 요청에 대해 응답한다. 특히, CA는 높은 대역폭, soft real-time 네트워크 응용프로그램 제어에 맞춰 설계되었고, 이는 엄청난 수의 컴퓨터와 장비들을 포함한 제어시스템 구축에 EPICS가 사용될 수 있는 이유이다. 또한, EPICS는 신뢰성 레벨을 제공하며, 이미 구축된 시스템의 유지보수가 쉽다는 장점이 있다[5].

<sup>1</sup>\* SNMP에 관한 자세한 설명은 SNMP 이해 및 응답시간 테스트 문서를 참조 바란다

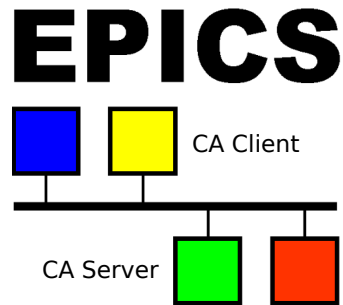


Figure 2 Channel Access 구조

## 1.4 EPICS와 SNMP의 통합

EPICS와 통합된 네트워크 기반의 제어 시스템 구축에 SNMP가 용이한 이유는 가속기 제어 시스템에 사용될 장비와 장치들은 Ethernet 환경을 통해 연결되며, 대부분 SNMP를 지원하기 때문이다. 따라서 EPICS와 SNMP의 통합은 중이온가속기 중앙 제어 시스템의 일관성, 유지관리의 용이성, 그리고 최적화 기술의 습득 및 축적의 관점에서 중요하다. 그림 3은 EPICS와 SNMP 통합 시스템의 구조로, 통합 시스템에서 모니터링 및 제어될 장비는 SNMP MIB를 통해 EPICS와 연결되고, 이는 IOC로 개발되어 CA를 통해 PV와 장비를 통신할 수 있게 한다. PV는 MIB가 가진 장비의 정보로 모니터링 및 제어에 사용된다[6].

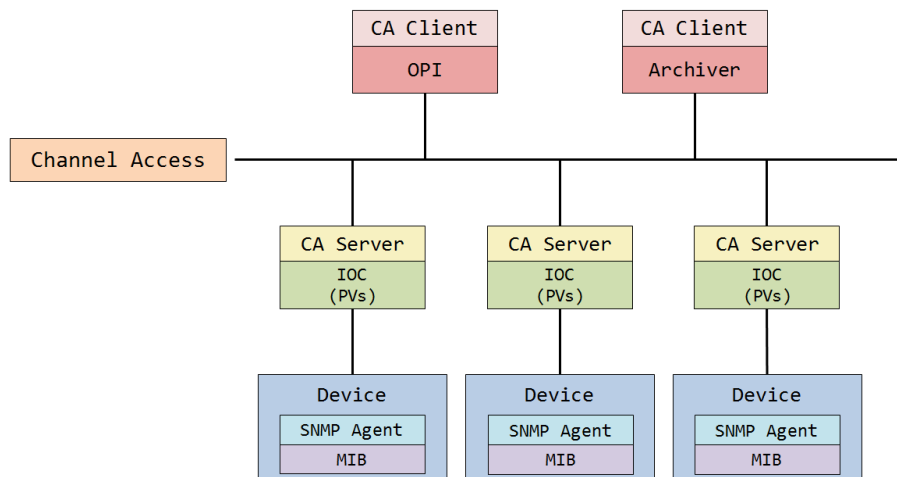


Figure 3 EPICS와 SNMP 통합 시스템 아키텍처

## 2 EPICS IOC Support Modules

EPICS IOC Support Module은 다른 종류의 인터페이스를 가진 하드웨어에 따라 새로운 타입의 레코드나 소프트웨어를 지원하는 Module인 Soft Support와 IOC 내에서 사용 가능한 하드웨어를 지원하는 Module인 Hardware Support로 분류된다. SNMP Soft Support Module은 이미 다수의 개발자에 의해 devSNMP와 이를 보완한 snmp-nslcl 등이 존재한다. 본 문서에서는 NSCL/FRIB의 SNMP Device Support Module(snmp-nslcl)을 사용하여 초기 버전의 모니터링 시스템을 구축한다[7].

### 2.1 NSCL/FRIB SNMP Device Support Module

NSCL/FRIB SNMP Device Support Module (이하 snmp-nslcl)은 EPICS에서 SNMP를 통한 하드웨어 장비와의 통신을 위해 2004년 LANL<sup>2</sup>의 Richard Dubney에 의해 최초 개발된 후 DESY<sup>3</sup>의 Albert Kagarmanov에 의해 개발된 devSNMP를 기반으로 NSCL/FRIB<sup>4</sup>의 John Priller에 의해 개발되었다. DESY의 devSNMP는 snmpv1/2c와 Read만을 지원했다. 이에 devSNMP 코드를 기반으로 snmpv1/2c/3과 Read/ Write를 지원하는 snmp-nslcl이 개발되었고, 이는 현재 Wiener/ISEG/MPOD power supply crates에 초점을 두고 RC9 버전까지 개발되었으며, SNMPv2에 최적화 되어있다[8].

#### 사용 방법

snmp-nslcl을 사용하여 SNMP를 통해 하드웨어와 EPICS가 통신하기 위해서는 Net-SNMP library와 SNMP를 지원하는 장비, 그리고 장비의 MIB파일이 필요하다. MIB는 보통 리눅스 시스템에서 아래의 경로에 있다.

```
/usr/local/share/snmp/mibs/  
/usr/share/mibs
```

하지만 일반적인 경로에 없을 경우, 아래와 같이 IOC st.cmd에 epics환경변수로 MIB의 경로를 추가할 수 있다.

```
epicsEnvSet("MIBDIRS", "+$(TOP)/mibs:/some/other/directory")
```

기본적인 설정을 끝낸 후, 장비를 모니터링 및 제어하기 위해서는 원하는 정보 값 즉, MIB의 OID 객체를 Support Module의 DB 파일에 record로 정의하여야 한다. snmp-nslcl은 SNMP Read/Write에 따라 Input(ai, longin, stringin, waveform(DBF\_STRING, DBF\_CHAR, DBF\_UCHAR))과 Output(ao, longout, stringout)의 record 타입을 지원한다. record는 설명, 디바이스 타입, 스캔 시간, 초기화 등의 다양한 필드<sup>6</sup>를 가지며, 필드는 각 Record의 목적에 맞게 사용하면 된다. record의 포맷은 다음과 같다.

---

<sup>2</sup>\* Los Alamos National Laboratory

<sup>3</sup>\* Deutsches Elektronen-Synchrotron

<sup>4</sup>\* National Superconducting Cyclotron Laboratory

<sup>5</sup>\* Facility for Rare Isotope Beams

<sup>6</sup>\* 각 레코드 및 필드에 대한 정의는 EPICS Record Reference Manual[9]을 참조 바란다

```
record(record type, "PV Name") {
    field(DESC, "Description")
    field(DTYP, "Device Type")
    field(SCAN, "Scan rate")
    field(PREC, "Display Precision")
    field(INP/OUT, "@host community OIDname mask dataLength [set_type[special_flags]]")
    .
    .
}
```

snmp-nslcl의 정의된 record에서 INP/OUT 필드는 장비의 hostname, community string, 객체의 이름 등의 형태가 snmpget 명령어와 유사하다는 것을 알 수 있다.

```
snmpget -v 2c -c public localhost RFC1213-MIB::sysName.0
```

또한, 데이터 타입에 따라 정의되는 mask와 snmpset 명령어의 데이터 타입과 유사하게 set\_type을 정의하여야 한다<sup>7</sup>. 예를 들어, 장비의 Voltage 모니터링 및 제어 시 record는 다음과 같다.

```
record(ai, "VoltageRead") {
    field(DESC, "SNMP channel")
    field(DTYP, "Snmp")
    field(SCAN, ".2 second")
    field(PREC, "3")
    field(INP, "@host community WIENER-CRATE-MIB::outputMeasurementSenseVoltage.u0 Float: 100")
}

record(ao, "VoltageSet") {
    field(DESC, "SNMP channel")
    field(DTYP, "Snmp")
    field(SCAN, "Passive")
    field(PREC, "3")
    field(OUT, "@host community WIENER-CRATE-MIB::outputVoltage.u0 Float: 100 F")
}
```

각 record는 객체의 Access 권한과 데이터 타입에 따라 VoltageRead, VoltageSet의 이름으로 DB 파일에 정의되고, 이는 PV가 되어 CA를 통한 EPICS와 하드웨어의 통신이 가능해지므로, Voltage 값 변경 및 모니터링을 할 수 있다.

## 장단점

snmp-nslcl은 오랜 개발로 IOC의 안정성이 검증되어있으며, 여러 종류의 IOC Shell Commands가 제공되어 모듈 파라미터 구성과 문제를 진단하는 데 도움이 된다. 또한, SNMP의 다양한 데이터 타입(Integer, String, Gauge, IPAddress 등)을 지원한다. 하지만 v3를 지원한다고는 하나 사용할 수 없으며, 데이터 타입에 따라 mask와 data length 값을 설정해야 하며, 가끔 반환되는 데이터 값이 실제 값과 차이 나는 문제점이 있다.

---

<sup>7</sup>\* 추가적인 설명은 devSNMP Documentation[8]을 참조 바란다

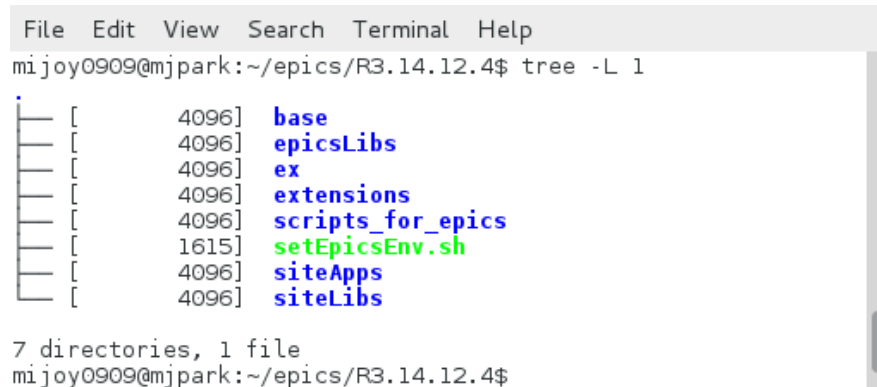
## 2.2 Device Support Module을 이용한 통합 시스템 구축

본 문서에서는 SNMPv1/2c를 지원하는 snmp-nslc RC8 버전을 사용하여 종이온가속기 제어 시스템 개발 환경에 맞춘 통합 모니터링 시스템 초기 버전을 구축하였다. 구축된 통합 시스템은 가속기 제어 시스템에 사용되는 다양한 장비 적용에 앞서 사무실 내의 프린터기에 적용되었다. 프린터 모니터링 시스템 구축에 사용된 소프트웨어와 하드웨어는 다음과 같다.

- Debian Linux 7 Wheezy
- NET-SNMP v5.4.3
- EPICS v3.14.12.4
- snmp-nslc (NSCL/FRIB SNMP Device Support Module) vRC8
- EPICS CSS (Control System Studio)
- Printers(XEROX ApeosPort-IV C3375, KYOCERA FS-9530DN)

### Customize 이유 및 수정 사항

종이온가속기 제어 시스템의 개발 환경은 각 시스템의 원활한 개발 및 개발자 간의 의사소통, 그리고 개발되는 코드들의 단계별 형상관리를 위해 그림 4와 같이 표준화된 구조로 되어있다[10]. 따라서 snmp-nslc를 제어 개발 환경과 프린터 모니터링 환경에 최적화하기 위해 아래 몇 가지 사항을 수정하였다.



```
File Edit View Search Terminal Help
mijoy0909@mjpark:~/epics/R3.14.12.4$ tree -L 1
.
├── [ 4096] base
├── [ 4096] epicsLibs
├── [ 4096] ex
├── [ 4096] extensions
├── [ 4096] scripts_for_epics
├── [ 1615] setEpicsEnv.sh
├── [ 4096] siteApps
└── [ 4096] siteLibs

7 directories, 1 file
mijoy0909@mjpark:~/epics/R3.14.12.4$
```

Figure 4 표준화된 종이온가속기 제어 환경의 구조

1. 표준화된 종이온가속기 제어 환경에 맞춰 Library (siteLibs)와 Application (siteApps)으로 분리
  - 종이온가속기 제어 환경은 소프트웨어의 재사용을 위해 Library와 IOC를 개발하는 Application을 분리했다. 따라서 그림 5와 같이 Library와 Application이 통합된 구조였던 snmp-nsl을 그림 6(a), 6(b)의 Makefile 수정과 파일 이동의 과정을 거쳐 표준화된 환경에 맞게 siteLibs(그림 6(c))과 siteApps(그림 6(d))로 분리했다.

```

File Edit View Search Terminal Help
mijoy0909@mjpark:~/fribsnmp/snmp$ tree -L 2
.
├── [ 4096] configure
│   ├── [ 507] CONFIG
│   ├── [ 579] CONFIG_APP
│   ├── [ 315] Makefile
│   ├── [ 895] RELEASE
│   ├── [ 133] RULES
│   ├── [ 0] RULES.Db
│   ├── [ 55] RULES_DIRS
│   ├── [ 53] RULES.ioc
│   └── [ 54] RULES_TOP
├── [ 4096] documentation
│   └── [ 24135] devSnm.html
├── [ 4096] iocBoot
│   ├── [ 4096] iocsnmp
│   └── [ 135] Makefile
├── [ 391] Makefile
├── [ 4096] mibs
│   ├── [ 56987] BRIDGE-MIB-rfc4188.txt
│   ├── [ 61238] ENTITY-MIB-rfc4133.txt
│   ├── [ 686647] FOUNDRY-MIB.txt
│   ├── [ 53157] IEEE8021-PAE-MIB-bridge-03.txt
│   ├── [ 45959] IEEE8023-LAG-MIB.txt
│   ├── [ 87014] MAU-MIB-rfc2668.txt
│   ├── [ 41916] P-BRIDGE-MIB-rfc4363.txt
│   ├── [ 86098] Q-BRIDGE-MIB-rfc4363.txt
│   ├── [ 12876] RADIUS-AUTH-CLIENT-MIB-rfc2618.txt
│   ├── [ 225585] RMON2-MIB-rfc4502.txt
│   ├── [ 95167] TOKEN-RING-RMON-MIB-rmonmib-01.txt
│   └── [ 100507] WIENER-CRATE-MIB.txt
├── [ 4096] snmpApp
│   ├── [ 4096] Db
│   ├── [ 304] Makefile
│   └── [ 4096] src
├── [ 4096] WienerCrate
│   ├── [ 4096] configure
│   ├── [ 4096] iocBoot
│   ├── [ 523] Makefile
│   ├── [ 4096] mibs
│   ├── [ 4096] opi
│   ├── [ 437] README.txt
│   ├── [ 4096] wienerCrateExampleApp
│   └── [ 4096] wienerCrateSup
└── [ 15 directories, 27 files
mijoy0909@mjpark:~/fribsnmp/snmp$

```

Figure 5 snmp-nsl의 기본 구조

```
File Edit View Search Terminal Tabs Help
mjoy0909@mjpark: ~/epics/R3.14.12.4/sit... x mjoy0909@mjpark: ~/epics/R3.14.12.4/sit... x
TOP = ../../
include $(TOP)/configure/CONFIG

LIBRARY += devSnmp
DEB += snmpRecord.dbd
DEBINC += snmpRecord
DEB += menuCalc.dbd
DEBINC += menuCalc
devSnmp_SRCS += snmpRegister.cpp snmpSessShow.c devSnmpSoft2.cpp snmpRecord.c

USR_CFLAGS += 'net-snmpp-config --cflags'
USR_LDFLAGS += 'net-snmpp-config --libs'
PROD_LDLIBS += 'net-snmpp-config --libs'

USR_CFLAGS += $(shell $(PERL) ../getNetSNMPVersion.pl)
USR_CPPFLAGS += $(shell $(PERL) ../getNetSNMPVersion.pl)

DEB += snmpDevSoft.dbd

include $(TOP)/configure/RULES

...

*Makefile* 22L, 598C 1,1 All
```

```
File Edit View Search Terminal Tabs Help
mjoy0909@mpjark: ~/epics/R3.14.12.4/sit... x mjoy0909@mpjark: ~/epics/R3.14.12.4/sit... x
|TOP.../..
include $(TOP)/configure/CONFIG

USR_INCLUDES += -I$(RAON_SITELIBS)/include
USR_OBD_FLAGS += -I$(RAON_SITELIBS)/dbd

devSnpM_DIR += $(RAON_SITELIBS)/lib/$(T_A)

#-----
# ADD MACRO DEFINITIONS AFTER THIS LINE
#-----

# Build the IOC application

PROD_IOC = snmp
# snmp.dbd will be created and installed
DBD += snmp.dbd

# snmp.dbd will be made up from these files:
snmp_DBD += base.dbd
snmp_DBD += snmpRecord.dbd

# Include dbd files from all support applications:
#snmp_DBD += xxx.dbd
snmp_DBD += snmpDevSoft.dbd

# Add all the support libraries needed by this IOC
#snmp_LIBS += xxx
snmp_LIBS += devSnpM

# snmp_registerRecordDeviceDriver.cpp derives from snmp.dbd
snmp_SRCS += snmp_registerRecordDeviceDriver.cpp

# Build the main IOC entry point on workstation OSs.
snmp_SRCS_DEFAULT += snmpMain.cpp
snmp_SRCS_vxWorks += -nil

# Add support from base/src/vxWorks if needed
snmp_OBJ_S_vxWorks += $(EPICS_BASE_BIN)/vxWorksLibrary

# Finally link to the EPICS Base Libraries
snmp_LIBS += $(EPICS_BASE_LIBS)

#-----
include $(TOP)/configure/RULES
#-----
# ADD RULES AFTER THIS LINE
Makefile 55L, 1248C
1,1
All
```

```
File Edit View Search Terminal Help
mjoy0909@mjpark:~/epics/R3.14.12.4/siteLibs/snmpMSULib$ tree -L 2
.
├── [ 305] Makefile
├── [ 4096] src
│   ├── [ 18485] devSnmp.h
│   ├── [ 160323] devSnmpSoft2.cpp
│   ├── [ 157118] devSnmpSoft.cpp
│   ├── [ 153250] devSnmpSoft_tmp.cpp
│   ├── [ 635] getNetsSNMPVersion.pl
│   ├── [ 730] Makefile
│   ├── [ 137] menuCalc.dbd
│   ├── [ 672] requireNetsSNMPVersion.h
│   ├── [ 104] snmpDevSoft.dbd
│   ├── [ 14156] snmp.h
│   ├── [ 8575] snmpRecord.c
│   ├── [ 26696] snmpRecord.dbd
│   ├── [ 6462] snmpRegister.cpp
│   └── [ 2887] snmpSessShow.c
└── 1 directory, 15 files

mjoy0909@mjpark:~/epics/R3.14.12.4/siteLibs/snmpMSULib
```

[illegible]

**Figure 6** 중이온가속기 제어 개발 환경에 맞춰 수정된 snmp-nsl

## 2. snmpRecord, snmpstrRecord 생성

- 앞서 언급했듯 snmp-nsl은 SNMP Read/Write에 따라 사용할 수 있는 record를 제공한다. 하지만 본 통합 시스템에서는 기존의 record가 아닌 string타입의 데이터에서 사용될 snmpstrRecord와 그 외 타입의 데이터에서 사용될 snmpRecord를 생성하여 사용한다.

```
1 menu(menuCalc) {
2     choice(menu_Plus,"Plus")
3     choice(menu_Minus,"Minus")
4     choice(menu_Mul,"Multiplication")
5     choice(menu_Div,"Division")
6 }
```



```

7
8 recordtype(snmpr) {
9     #include "dbCommon.dbd"
10
11     field(NAME,DBF_STRING) {
12         prompt("Record Name")
13         special(SPC_NOMOD)
14         size(61)
15     }
16     field(DESC,DBF_STRING) {
17         prompt("Descriptor")
18         promptgroup(GUI_COMMON)
19         size(41)
20     }
21     field(MJP,DBF_MENU) {
22         prompt("Desired Output")
23         promptgroup(GUI_OUTPUT)
24         menu(menuCalc)
25         interest(1)
26     }
27 .
28 .
29 .
30 }
31
32 recordtype(snmprstr) {
33     field(NAME,DBF_STRING) {
34         prompt("Record Name")
35         special(SPC_NOMOD)
36         size(61)
37     }
38     field(DESC,DBF_STRING) {
39         prompt("Descriptor")
40         promptgroup(GUI_COMMON)
41         size(41)
42     }
43     field(SCAN,DBF_MENU) {
44         prompt("Scan Mechanism")
45         promptgroup(GUI_SCAN)
46         special(SPC_SCAN)
47         menu(menuScan)
48         interest(1)
49     }
50 .
51 .
52 .
53 }

```

Record 생성 후 dbd 파일에 아래와 같이 추가한다.

```

device(snmpr,INST_I0,devSnmprSoft,"SoftChannel")
device(snmprstr,INST_I0,devSnmprstrSoft,"SoftChannel")

```

### 3. 소스코드 수정 및 새로운 필드 생성

- 프린터의 종이, 잉크의 잔량은 SNMP MIB의 객체에서 전체의 양과 남은 양의 정보를 계산하여 얻을 수 있다. 따라서 snmprRecord 내에 사칙연산 메뉴의 MJP 필드를 생성하였으며, siteLibs내 소스코드에 사칙연산을 위한 코드를 추가하였다. 이를 활용하면, 레코드 생성 시 MJP, SVAL, OVAL 필드 사용하여 잔량의 정보를 퍼센트로 나타낼 수 있다.

```

1 static long snmpSoftReadback(devSnmp_pv *pPV)
2 {
3     struct snmpRecord *psnmp = (struct snmpRecord *) pPV->record();
4     epicsStatus status = epicsError;
5     double new_val;
6     bool process_record = false;
7     .
8     .
9     .
10     psnmp->rval = new_val;
11     char oval[40];
12     char sval[40]; //percent
13     switch(psnmp->mjp)
14     {
15         case menu_Plus:
16             psnmp->val = ceil(psnmp->rval+psnmp->oval*psnmp->sval);
17             break;
18         case menu_Minus:
19             psnmp->val = ceil(psnmp->rval-psnmp->oval*psnmp->sval);
20             break;
21         case menu_Mul:
22             psnmp->val = ceil(psnmp->rval*psnmp->oval*psnmp->sval);
23             break;
24         case menu_Div:
25             psnmp->val = ceil(psnmp->rval/psnmp->oval*psnmp->sval);
26             break;
27     }
28
29     process_record = true;
30 }
31 }
32
33 .
34 .
35 .
36 // process record if needed
37 if (process_record) pPV->processRecord();
38
39 return(status);
40 }

```

## 프린터 모니터링 시스템 구축

그림 6과 같이 수정된 snmp-nscl을 사용하여 프린터 모니터링 시스템을 구축하기 위해서 몇 가지 절차만 수행하면 된다.

1. siteLibs에서 snmpMSULib을 컴파일한다.

```
mijoy0909@mjpark:~/epics/R3.14.12.4/siteLibs/snmpMSULib$ make
```

컴파일 후 그림 7과 같이 siteLibs에 dbd, lib, include 폴더가 생기게 되고, 각 폴더에는 아래 리스트의 파일들이 생성된다. 이 파일들은 siteApps에서 IOC 실행에 사용된다.

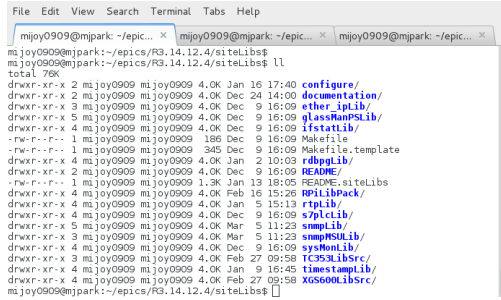
```

# include
mijoy0909@mjpark:~/epics/R3.14.12.4/siteLibs/include$ ls
menuCalc.h  snmpRecord.h

```

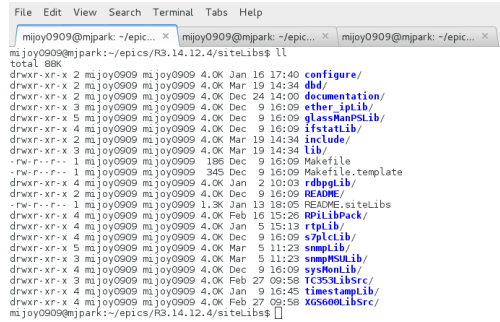
```
# lib
mjoy0909@mpark:~/epics/R3.14.12.4/siteLibs/lib/linux-x86_64$ ls
libdevSnmp.a libdevSnmp.so*

# dbd
mjoy0909@mpark:~/epics/R3.14.12.4/siteLibs/dbd$ ls
menuCalc.dbd snmpDevSoft.dbd snmpRecord.dbd
```



```
mjoy0909@mpark:~/epics/R3.14.12.4/siteLibs$ ll
total 79K
drwxr-xr-x 2 mjoy0909 mjoy0909 4.0K Jan 16 17:40 configure/
drwxr-xr-x 2 mjoy0909 mjoy0909 4.0K Dec 24 14:00 dbd/
drwxr-xr-x 3 mjoy0909 mjoy0909 4.0K Dec 9 16:09 documentation/
drwxr-xr-x 3 mjoy0909 mjoy0909 4.0K Dec 9 16:09 ether.igLib/
drwxr-xr-x 5 mjoy0909 mjoy0909 4.0K Dec 9 16:09 glassRanPSLib/
drwxr-xr-x 4 mjoy0909 mjoy0909 4.0K Dec 9 16:09 ifstatLib/
-rw-r--r-- 1 mjoy0909 mjoy0909 106 Dec 9 16:09 Makefile
-rw-r--r-- 1 mjoy0909 mjoy0909 345 Dec 9 16:09 Makefile.template
drwxr-xr-x 4 mjoy0909 mjoy0909 4.0K Jan 2 10:03 rdbpgLib/
drwxr-xr-x 2 mjoy0909 mjoy0909 4.0K Dec 9 16:09 README/
-rw-r--r-- 1 mjoy0909 mjoy0909 1.3K Jan 13 18:05 README.siteLibs
drwxr-xr-x 4 mjoy0909 mjoy0909 4.0K Feb 16 15:26 RPILibPack/
drwxr-xr-x 4 mjoy0909 mjoy0909 4.0K Jan 5 15:13 rtLib/
drwxr-xr-x 4 mjoy0909 mjoy0909 4.0K Dec 9 16:09 s7clib/
drwxr-xr-x 5 mjoy0909 mjoy0909 4.0K Mar 5 11:23 snmpLib/
drwxr-xr-x 3 mjoy0909 mjoy0909 4.0K Mar 5 11:23 snmpPSLib/
drwxr-xr-x 4 mjoy0909 mjoy0909 4.0K Dec 9 16:09 sysModLib/
drwxr-xr-x 3 mjoy0909 mjoy0909 4.0K Feb 27 09:58 TC353LibSrc/
drwxr-xr-x 4 mjoy0909 mjoy0909 4.0K Jan 9 16:45 timesampLib/
drwxr-xr-x 4 mjoy0909 mjoy0909 4.0K Feb 27 09:58 XOS600LibSrc/
mjoy0909@mpark:~/epics/R3.14.12.4/siteLibs$
```

(a) Before library Compile



```
mjoy0909@mpark:~/epics/R3.14.12.4/siteLibs$ ll
total 88K
drwxr-xr-x 2 mjoy0909 mjoy0909 4.0K Jan 16 17:40 configure/
drwxr-xr-x 2 mjoy0909 mjoy0909 4.0K Mar 19 14:34 dbd/
drwxr-xr-x 2 mjoy0909 mjoy0909 4.0K Dec 24 14:00 documentation/
drwxr-xr-x 3 mjoy0909 mjoy0909 4.0K Dec 9 16:09 ether.igLib/
drwxr-xr-x 5 mjoy0909 mjoy0909 4.0K Dec 9 16:09 glassRanPSLib/
drwxr-xr-x 4 mjoy0909 mjoy0909 4.0K Dec 9 16:09 ifstatLib/
drwxr-xr-x 2 mjoy0909 mjoy0909 4.0K Mar 19 14:34 include/
drwxr-xr-x 3 mjoy0909 mjoy0909 4.0K Mar 19 14:34 lib/
-rw-r--r-- 1 mjoy0909 mjoy0909 186 Dec 9 16:09 Makefile
-rw-r--r-- 1 mjoy0909 mjoy0909 345 Dec 9 16:09 Makefile.template
drwxr-xr-x 4 mjoy0909 mjoy0909 4.0K Jan 2 10:03 rdbpgLib/
drwxr-xr-x 2 mjoy0909 mjoy0909 4.0K Dec 9 16:09 README/
-rw-r--r-- 1 mjoy0909 mjoy0909 1.3K Jan 13 18:05 README.siteLibs
drwxr-xr-x 4 mjoy0909 mjoy0909 4.0K Feb 16 15:26 RPILibPack/
drwxr-xr-x 4 mjoy0909 mjoy0909 4.0K Jan 5 15:13 rtLib/
drwxr-xr-x 4 mjoy0909 mjoy0909 4.0K Dec 9 16:09 s7clib/
drwxr-xr-x 5 mjoy0909 mjoy0909 4.0K Mar 5 11:23 snmpLib/
drwxr-xr-x 3 mjoy0909 mjoy0909 4.0K Mar 5 11:23 snmpPSLib/
drwxr-xr-x 4 mjoy0909 mjoy0909 4.0K Dec 9 16:09 sysModLib/
drwxr-xr-x 3 mjoy0909 mjoy0909 4.0K Feb 27 09:58 TC353LibSrc/
drwxr-xr-x 4 mjoy0909 mjoy0909 4.0K Jan 9 16:45 timesampLib/
drwxr-xr-x 4 mjoy0909 mjoy0909 4.0K Feb 27 09:58 XOS600LibSrc/
mjoy0909@mpark:~/epics/R3.14.12.4/siteLibs$
```

(b) After library Compile

Figure 7 siteLibs 파일 리스트

2. siteApps의 snmpApp/Db에 db 파일을 추가한다.  
그림 8과 같이 모니터링을 원하는 MIB 객체정보를 record에 정의한 db 파일을 만들고, Makefile에 추가한다.



```
File Edit View Search Terminal Help
# -----
# XEROX Printer
# -----
# toner
record(snm, "${USER}:xerox_toner_B") {
    field(DESC, "SNMP channel")
    field(SCAN, "Passive")
    field(DTYP, "Softchannel")
    field(OUT, "%(XEROX) %(CM2) %(PR)prtMarkerSuppliesLevel.1.1 INTEGER: 100 ")
    field(IMP, "Division")
    field(OVAL, "26000")
    field(SVAL, "100")
}

record(snm, "${USER}:xerox_toner_Y") {
    field(DESC, "SNMP channel")
    field(SCAN, "Passive")
    field(DTYP, "Softchannel")
    field(OUT, "%(XEROX) %(CM2) %(PR)prtMarkerSuppliesLevel.1.2 INTEGER: 100 ")
    field(IMP, "Division")
    field(OVAL, "15000")
    field(SVAL, "100")
}

25,1 1%
```

Figure 8 EPICS DB파일

```
1 TOP=../..
2 include $(TOP)/configure/CONFIG
3 #-----
4 # ADD MACRO DEFINITIONS AFTER THIS LINE
5
6 #-----
7 # Optimization of db files using dbst (DEFAULT: NO)
8 #DB_OPT = YES
9
```

```

10 #-----
11 # Create and install (or just install) into <top>/db
12 # databases, templates, substitutions like this
13 #DB += xxx.db
14 DB += printer.vdb
15
16 #-----
17 # If <anyname>.db template is not named <anyname>*.template add
18 # <anyname>_template = <templatename>
19
20 include $(TOP)/configure/RULES
21 #-----
22 # ADD RULES AFTER THIS LINE

```

### 3. siteApps에서 snmp를 컴파일 한다.

```
mijoy0909@mpark:~/epics/R3.14.12.4/siteApps/snmp$ make
```

컴파일 후 그림 9와 같이 bin, db, dbd 파일이 생성되고, 각각의 폴더에는 siteLibs와 마찬가지로 IOC 실행에 사용될 파일이 생성된다.

```

mijoy0909@mpark:~/epics/R3.14.12.4/siteApps/snmp$ ll
total 24K
drwxr-xr-x 2 mijoy0909 mijoy0909 4.0K Mar 19 14:53 configure/
drwxr-xr-x 3 mijoy0909 mijoy0909 4.0K Mar 19 15:53 iocBoot/
-rw-r--r-- 1 mijoy0909 mijoy0909 467 Mar 19 15:53 Makefile
drwxr-xr-x 2 mijoy0909 mijoy0909 4.0K Mar 18 18:21 mibs/
drwxr-xr-x 3 mijoy0909 mijoy0909 4.0K Mar 19 15:53 api/
drwxr-xr-x 4 mijoy0909 mijoy0909 4.0K Mar 18 18:15 snmpApp/
mijoy0909@mpark:~/epics/R3.14.12.4/siteApps/snmp$

```

(a) Before Application Compile

```

mijoy0909@mpark:~/epics/R3.14.12.4/siteApps/snmp$ ll
total 36K
drwxr-xr-x 3 mijoy0909 mijoy0909 4.0K Mar 19 14:54 bin/
drwxr-xr-x 4 mijoy0909 mijoy0909 4.0K Mar 19 14:54 configure/
drwxr-xr-x 2 mijoy0909 mijoy0909 4.0K Mar 19 14:54 db/
drwxr-xr-x 2 mijoy0909 mijoy0909 4.0K Mar 19 14:54 dbd/
drwxr-xr-x 3 mijoy0909 mijoy0909 4.0K Mar 19 15:53 iocBoot/
-rw-r--r-- 1 mijoy0909 mijoy0909 467 Mar 19 15:53 Makefile
drwxr-xr-x 2 mijoy0909 mijoy0909 4.0K Mar 18 18:21 mibs/
drwxr-xr-x 3 mijoy0909 mijoy0909 4.0K Mar 19 15:53 api/
drwxr-xr-x 4 mijoy0909 mijoy0909 4.0K Mar 18 18:15 snmpApp/
mijoy0909@mpark:~/epics/R3.14.12.4/siteApps/snmp$

```

(b) After Application Compile

Figure 9 siteApps/snmp 파일 리스트

### 4. IOC 실행 전 st.cmd파일을 수정 및 실행권한을 부여한다.

/iocBoot/iocsnmp로 이동 후 st.cmd 파일에 앞서 생성한 db파일을 추가해준다. 이때, MIB파일의 경로, 사용하고자 하는 SNMP의 버전, 사용한 매크로 값에 대한 정보도 추가해준다.

```

1 #!../bin/linux-x86_64/snmp
2
3 ## You may have to change snmp to something else
4 ## everywhere it appears in this file
5
6 < envPaths
7
8 cd ${TOP}
9
10 epicsEnvSet("MIBDIRS", "+$(TOP)/mibs")
11 devSnpSetSnpVersion("10.1.4.184", "SNMP_VERSION_1")
12 devSnpSetSnpVersion("10.1.4.182", "SNMP_VERSION_2c")
13
14 epicsEnvSet("PR", "Printer-MIB::")
15 epicsEnvSet("JM", "Job-Monitoring-MIB::")
16 epicsEnvSet("SMI", "SNMPv2-SMI::")

```

```

17 epicsEnvSet("XM", "XEROX-SERVICE-MONITORING-MIB::")
18 epicsEnvSet("CM1", "admin")
19 epicsEnvSet("CM2", "public")
20 epicsEnvSet("XEROX", "10.1.4.182")
21 epicsEnvSet("KYOCERA", "10.1.4.184")
22
23 ## Register all support components
24 dbLoadDatabase "dbd/snmp.dbd"
25 snmp_registerRecordDeviceDriver pdbbase
26
27 ## Load record instances
28 dbLoadRecords("db/printer.vdb", "USER=mijoy0909Host")
29
30 cd ${TOP}/iocBoot/${IOC}
31 iocInit
32
33 ## Start any sequence programs
34 #seq sncxxx, "user=mijoy0909Host"

```

5. IOC 실행 및 통신 확인 과정을 거친다.  
 IOC가 실행 후 PV 리스트를 확인한 결과는 그림 10과 같다.

```

File Edit View Search Terminal Tabs Help
mijoy0909@mpark: ~/epic... x mijoy0909@mpark: ~/epic... x mijoy0909@mpark: ~/epic... x
dbLoadDatabase "dbd/snmp.dbd"
snmp_registerRecordDeviceDriver pdbbase
## Load record instances
dbLoadRecords("db/printer.vdb", "USER=mijoy0909Host")
cd /home/mijoy0909/epics/R3.14.12.4/siteApps/snmp/iocBoot/iocsnmp
iocInit
Starting iocInit
#####
## EPICS R3.14.12.4 $Date: Mon 2013-12-16 15:51:45 -0600$
## EPICS Base built Aug 29 2014
#####
iocRun: All initialization complete
## Start any sequence programs
#seq sncxxx, "user=mijoy0909Host"
epics>
epics> dbL
mijoy0909Host:kyocera_TR1
mijoy0909Host:kyocera_TR2
mijoy0909Host:kyocera_TR3
mijoy0909Host:kyocera_TR4
mijoy0909Host:kyocera_TR5
mijoy0909Host:kyocera_active
mijoy0909Host:kyocera_length1
mijoy0909Host:kyocera_length2
mijoy0909Host:kyocera_length3
mijoy0909Host:kyocera_length4
mijoy0909Host:kyocera_length5
mijoy0909Host:kyocera_power
mijoy0909Host:kyocera_toner
mijoy0909Host:kyocera_width1
mijoy0909Host:kyocera_width2
mijoy0909Host:kyocera_width3
mijoy0909Host:kyocera_width4
mijoy0909Host:kyocera_width5
mijoy0909Host:xerox_TR1
mijoy0909Host:xerox_TR2
mijoy0909Host:xerox_TR3
mijoy0909Host:xerox_TR4
mijoy0909Host:xerox_TR5
mijoy0909Host:xerox_length1
mijoy0909Host:xerox_length2
mijoy0909Host:xerox_length3
mijoy0909Host:xerox_length4
mijoy0909Host:xerox_length5
mijoy0909Host:xerox_power
mijoy0909Host:xerox_toner_B
mijoy0909Host:xerox_toner_C
mijoy0909Host:xerox_toner_M
mijoy0909Host:xerox_toner_Y
mijoy0909Host:xerox_toner_drum_B
mijoy0909Host:xerox_toner_drum_C
mijoy0909Host:xerox_toner_drum_M
mijoy0909Host:xerox_toner_drum_Y
mijoy0909Host:xerox_waste
mijoy0909Host:xerox_width1
mijoy0909Host:xerox_width2
mijoy0909Host:xerox_width3
mijoy0909Host:xerox_width4
mijoy0909Host:xerox_width5
mijoy0909Host:kyocera_state
mijoy0909Host:xerox_print
mijoy0909Host:xerox_state
mijoy0909Host:xerox_state_sub
epics>

```

Figure 10 EPICS IOC 실행 및 PV 리스트

PV가 장비와 통신 되고 있는지 확인하기 위해 EPICS의 Command, CA monitor, snmpget을 값을 확인한 결과, 같은 값을 보이므로 EPICS로 프린터기가 모니터링 됨을 확인할 수 있다.

```
# EPICS Command
epics> dbpr mijoy0909Host:xerox_state
ASG:          DESC: SNMP channel  DISA: 0          DISP: 0
DISV: 1       NAME: mijoy0909Host:xerox_state      SEVR: NO_ALARM
STAT: NO_ALARM TPRO: 0          VAL: Printing...
```

```
# CA monitor
mijoy0909@mjpark:~$ camonitor mijoy0909Host:xerox_state
mijoy0909Host:xerox_state      2015-03-19 15:23:32.007583 Printing...
```

```
# snmpget
mijoy0909@mjpark:~$ snmpget -v 2c -c public 10.1.4.182 prtConsoleDisplayBufferText.1.1
Printer-MIB::prtConsoleDisplayBufferText.1.1 = STRING: "Printing..."
```

## OPI 구현

IOC가 실행되어 각각의 PV들이 CA 통신으로 프린터기의 정보를 모니터링하면, EPICS의 UI (User Interface) 프로그램인 CSS (Control System Studio)를 사용하여 프린터 모니터링 시스템의 OPI (Operator Interface)를 구현한다. 그림 11은 프린터의 상태, 소모품의 잔량 등의 정보 확인 및 종이 걸림, 소모품 부족 등의 알림을 알 수 있도록 구현되었다.



Figure 11 프린터 모니터링 시스템 OPI

### 3 EPICS Integration

이미 개발되어있는 EPICS Support Module을 사용하여 EPICS와 SNMP를 통합할 수 있지만, 자체적으로 중이온가속기 제어 시스템에 최적화된 통합 시스템 구축을 위해 유동적이고, API 확장이 가능한 Net-SNMP와 EPICS를 통합한다. 통합 시스템은 모니터링 시에는 SNMPv2c, 제어 시에는 SNMPv3로 최적화한다. 통합에 사용한 소프트웨어 하드웨어는 그림 12와 같다.

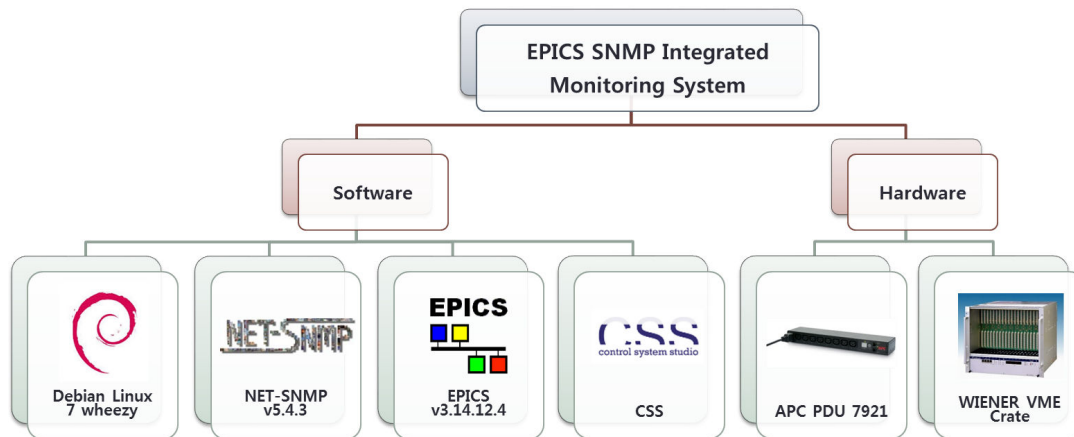


Figure 12 모니터링 시스템의 구성

#### 3.1 Legacy SNMP Code (Net-snmp tutorial)

Net-SNMP는 1992년 Carnegie Mellon University에서 시작된 프로젝트에서 개발된 공개 코드 SNMP(v2.1.2.1)를 기반으로 개발된 패키지이다. 이는 CMU-SNMP, UCD-SNMP를 거쳐 2000년 Net-SNMP가 되어, IPv4, IPv6를 사용하여 SNMPv1/2c/3을 구현하고, Agent와 Manager가 정보를 주고받을 수 있도록 snmpget, set, walk 등의 명령어를 제공하는 패키지로, Tutorials를 제공해 패키지 사용에 도움을 준다[11][12]. 따라서 Net-SNMP에서 제공하는 Client/Manager Coding Tutorials를 활용하여 SNMP와 Net-SNMP 명령어와 패키지 등에 대해 이해하고 EPICS와의 통합에 사용한다.

#### Simple Async Application

Client/Manager Coding Tutorials에는 장비에서 원하는 정보 값을 얻는 방법으로 동기식의 Simple application과 비동기식의 Simple Async Application이 있다. Simple Async Application 예제의 Host, Community, OID 값을 원하는 장비에 맞춰 변경해 컴파일한 후 실행시키면 아래와 같이 Simple application의 결과와 Simple Async Application 결과를 모두 보여준다.

```

mijoy0909@mjpark:~/rawcode$ make asyncapp
gcc -I. 'net-snmp-config --cflags' -c -o asyncapp.o asyncapp.c
gcc -o asyncapp asyncapp.o 'net-snmp-config --libs'
  
```

```

1 mijoy0909@mjpark:~/rawcode$ ./asynccapp
2 ----- synchronous -----
3 11:08:18.712241 10.1.4.182: HOST-RESOURCES-MIB::hrDeviceDescr.1 = STRING: FUJI XEROX ApeosPort-IV
   C3375 v 84. 18. 0 Multifunction System
4 11:08:18.714129 10.1.4.182: Printer-MIB::prtGeneralPrinterName.1 = STRING: "ApeosPort-IV C3375 "
5 11:08:18.724114 10.1.4.182: Printer-MIB::prtConsoleDisplayBufferText.1.1 = STRING: "Print-User
   Action"
6 11:08:18.760751 10.1.4.184: HOST-RESOURCES-MIB::hrDeviceDescr.1 = STRING: FS-9530DN
7 11:08:18.780953 10.1.4.184: Printer-MIB::prtGeneralPrinterName.1 = STRING: "FS-9530DN"
8 11:08:19.144312 10.1.4.184: Printer-MIB::prtConsoleDisplayBufferText.1.1 = STRING: "Please wait
   "
9 ----- asynchronous -----
10 11:08:19.185517 10.1.4.182: HOST-RESOURCES-MIB::hrDeviceDescr.1 = STRING: FUJI XEROX ApeosPort-IV
   C3375 v 84. 18. 0 Multifunction System
11 11:08:19.185877 10.1.4.184: HOST-RESOURCES-MIB::hrDeviceDescr.1 = STRING: FS-9530DN
12 11:08:19.186001 10.1.4.182: Printer-MIB::prtGeneralPrinterName.1 = STRING: "ApeosPort-IV C3375 "
13 11:08:19.186261 10.1.4.184: Printer-MIB::prtGeneralPrinterName.1 = STRING: "FS-9530DN"
14 11:08:19.194116 10.1.4.182: Printer-MIB::prtConsoleDisplayBufferText.1.1 = STRING: "Print-User
   Action"
15 11:08:19.244038 10.1.4.184: Printer-MIB::prtConsoleDisplayBufferText.1.1 = STRING: "Please wait
   "

```

Simple application은 Agent에 요청을 보내고 응답이 올 때까지 블럭킹되어 대기상태로 있으므로 중이온가속기 제어 시스템에 사용되는 다양한 장비를 모니터링하기엔 적합하지 않다. 반면에 Simple Async Application은 Agent로부터 요청에 대한 응답이 있을 때 callback 형식으로 호출되어 결과를 출력한다. 따라서 EPICS와 SNMP의 통합에는 비동기식으로 데이터를 전송하는 Simple Async Application 예제를 사용한다.



## 3.2 Library 구성

Simple Async Application(이하 AsyncApp)과 EPICS의 통합에서 Library는 소스코드, Record파일, dbd파일로 구성된다.

### Record

AsyncApp과 EPICS IOC의 통신에 사용할 Record를 생성한다. EPICS의 Example IOC Application에서 xxxRecord.dbd를 사용하여 SNMP의 다양한 데이터 타입 중 analog와 string 타입의 snmpRecord와 snmpstrRecord의 dbd파일을 생성한다. 이때, 기본적인 필드 외에 그림 13과 같이 통신에 필요한 장비의 Host, Community string, OID와 snmp 버전 정보 및 인증과 암호화 비밀번호에 대한 필드를 추가한다.

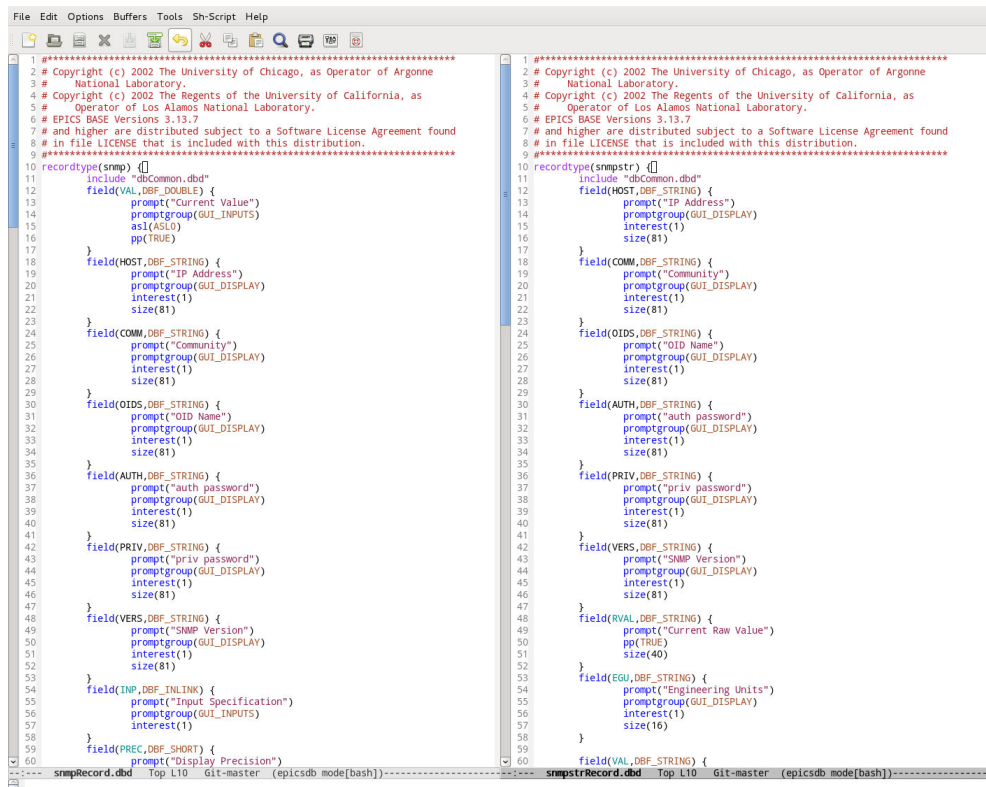


Figure 13 record.dbd 파일 (left: snmpRecord.dbd right: snmpstrRecord.dbd)

Record 생성 후 xxxRecord.c를 수정하여 그림 14와 같이 snmpRecord.c와 snmpstrRecord.c를 생성한다. 이때 string 타입의 레코드는 analog타입의 레코드와 달리 알람을 설정 할 필요가 없으므로 snmpstrRecord.c는 process함수의 checkAlarms부분을 주석처리 한다.

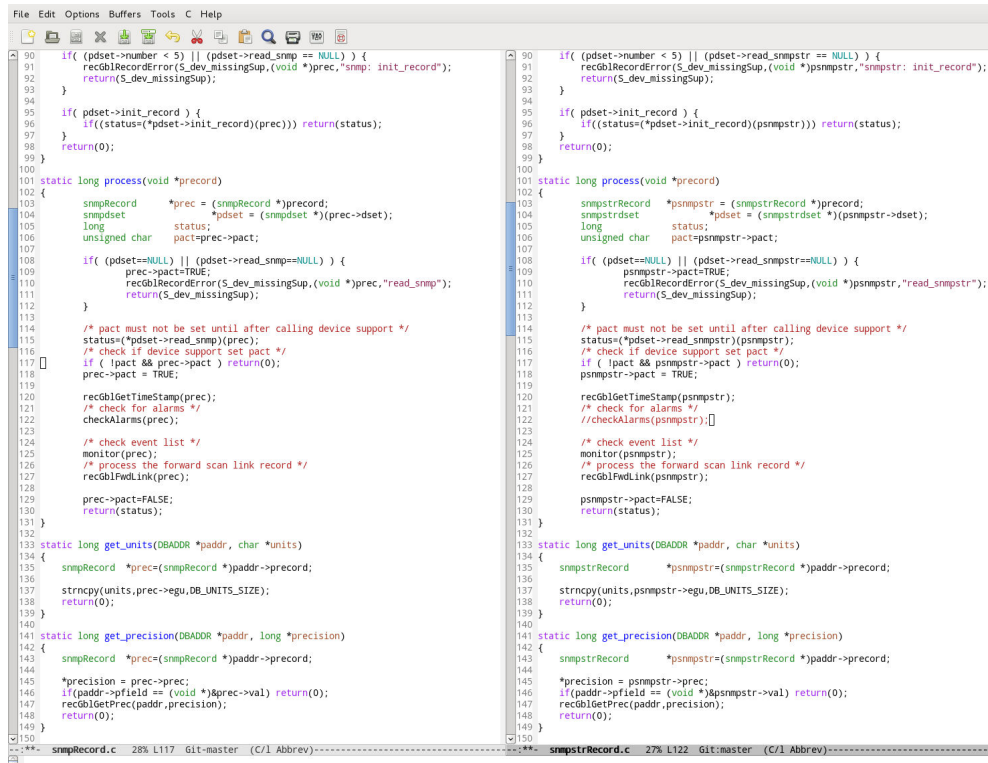


Figure 14 record.c 파일 (left: snmpReocrd.c right: snmpstrRecord.c)

## 소스파일

AsyncApp의 소스코드는 EPICS 환경에 맞춰 수정이 필요하다. 몇 가지 주요 수정 사항은 아래와 같고, 나머지 코드는 APPENDIX를 참조 바란다.

1. AsyncApp의 코드와 달리 host, community, oids는 필드로 사용되므로 OID정보와 snmp session, pdu, record 등에 대한 값은 snmp\_info의 구조로 정의한다.

```
/*
 * a list of variables to query for
 */
struct oid_info {
    char Name[81];
    oid Oid[MAX_OID_LEN];
    unsigned int OidLen;
};

typedef struct oid_info OID;
typedef struct snmp_info
{
    OID oid_info;
    struct snmp_session *sess;
    struct snmp_session ss;
    struct snmp_pdu *getreq;
    struct snmp_pdu *setreq;
    /* startup all hosts */
    /* startup all hosts */
}
```

```

char sval[12];
char type;
epicsMutexId mutexId;
void *psnmpRecord;
void *psnmpstrRecord;
} SNMP_INFO;

```

2. devSNMPSoft, devSNMPSoftstr을 위한 dset(Address of Device Support Entry Table)을 생성한다.

```

typedef struct devSNMPSoft{
    long number;
    DEVSUPFUN report;
    DEVSUPFUN init;
    DEVSUPFUN init_record;
    DEVSUPFUN get_ioint_info;
    DEVSUPFUN read_write_record;
    DEVSUPFUN special_linconv;
}devSNMPSoft;

typedef struct devSNMPSoftstr{
    long number;
    DEVSUPFUN report;
    DEVSUPFUN init;
    DEVSUPFUN init_record;
    DEVSUPFUN get_ioint_info;
    DEVSUPFUN read_write_record;
    DEVSUPFUN special_linconv;
}devSNMPSoftstr;

devSNMPSoft devSNMPRead = {6, NULL, NULL, snmp_RInit, NULL, snmp_Read, NULL};
devSNMPSoft devSNMPWrite = {6, NULL, NULL, snmp_WInit, NULL, snmp_Write, NULL};
epicsExportAddress(dset, devSNMPRead);
epicsExportAddress(dset, devSNMPWrite);
devSNMPSoftstr devSNMPRead_str = {6, NULL, NULL, snmp_strInit, NULL, snmp_strRead, NULL};
epicsExportAddress(dset, devSNMPRead_str);

```

3. initialize에 SNMP의 기본 정보와 SNMPv3의 인증과 암호화 부분을 추가한다.

```

void initializeR(void *precord)
{
    snmpRecord *psnmp = (snmpRecord *)precord;
    SNMP_INFO *snmpinfo = (SNMP_INFO*)psnmp->dpvt;

    static int snmpInitd = FALSE;
    snmpinfo->oid_info.OidLen = MAX_OID_LEN;
    if (!snmpInitd)
    {
        init_snmp("snmpDevAsync.c");          /* initialize library */
        snmpInitd = TRUE;
    };

    if (strlen(psnmp->vers) == 14) {           /* VERSION 3 */
        snmp_sess_init(&(snmpinfo->ss));      /* initialize session */
        snmpinfo->ss.version = SNMP_VERSION_3;
        snmpinfo->ss.peername = strdup(psnmp->host);
        snmpinfo->ss.securityName = strdup(psnmp->comm);
        snmpinfo->ss.securityNameLen = strlen((char*)snmpinfo->ss.securityName);
        snmpinfo->ss.securityLevel = SNMP_SEC_LEVEL_AUTHPRIV;
    }
}

```

```

/* Authentication Protocol */
/* MD5 */
snmpinfo->ss.securityAuthProto = usmHMACMD5AuthProtocol;
snmpinfo->ss.securityAuthProtoLen = sizeof(usmHMACMD5AuthProtocol)/sizeof(oid);
snmpinfo->ss.securityAuthKeyLen = USM_AUTH_KU_LEN;

/* Authentication keys generation */
if (generate_Ku(snmpinfo->ss.securityAuthProto,
    snmpinfo->ss.securityAuthProtoLen,
    (u_char *) strdup(psnmp->auth), strlen(psnmp->auth),
    snmpinfo->ss.securityAuthKey,
    &snmpinfo->ss.securityAuthKeyLen) != SNMPERR_SUCCESS) {
    snmp_perror("Error!!!!!!!!!!!!!!");
    snmp_log(LOG_ERR,
        "Error generating Ku from authentication pass phrase. \n");
    exit(1);
}

/* Privacy protocol */
/* DES */
snmpinfo->ss.securityPrivProto = usmDESPrivProtocol;
snmpinfo->ss.securityPrivProtoLen = USM_PRIV_PROTO_DES_LEN;
snmpinfo->ss.securityPrivKeyLen = USM_PRIV_KU_LEN;

/* Privacy keys generation */
if (generate_Ku(snmpinfo->ss.securityAuthProto,
    snmpinfo->ss.securityAuthProtoLen,
    (u_char *) strdup(psnmp->priv), strlen(psnmp->priv),
    snmpinfo->ss.securityPrivKey,
    &snmpinfo->ss.securityPrivKeyLen) != SNMPERR_SUCCESS) {
    printf ("Error generating Ku from privacy pass phrase. \n");
    exit(1);
}

snmpinfo->ss.callback = asynch_response; /* default callback */
snmpinfo->ss.callback_magic = snmpinfo;
} else { /* VERSION 2c */
    snmp_sess_init(&(snmpinfo->ss)); /* initialize session */
    snmpinfo->ss.version = SNMP_VERSION_2c;
    snmpinfo->ss.peername = strdup(psnmp->host);
    snmpinfo->ss.community = (unsigned char*)strdup(psnmp->comm);
    snmpinfo->ss.community_len = strlen((char*)snmpinfo->ss.community);
    snmpinfo->ss.callback = asynch_response; /* default callback */
    snmpinfo->ss.callback_magic = snmpinfo;
}

/* parse the oids */
if (!read_objid(psnmp->oids, snmpinfo->oid_info.Oid, &snmpinfo->oid_info.OidLen))
{
    printf("parse the oids %s\n",psnmp->oids);
    snmp_perror("read_objid");
    exit(1);
}
}

```

4. Switch문을 사용하여 SNMP 데이터 타입에 따라 callback 되는 데이터값을 처리한다.

```

int return_data(int status, struct snmp_info *sinfo, struct snmp_pdu *pdu)
{
    char buf[1024];

```

```

struct variable_list *vp;
int ix;
char *getdata = NULL;
int nVal;
char tVal[40];
char *sVal;
char *pValStr;
struct snmp_session *sp = sinfo->sess;
snmpRecord *psnmp = (snmpRecord *)sinfo->psnmpRecord;
snmpstrRecord *psnmpstr = (snmpstrRecord *)sinfo->psnmpstrRecord;

switch (status) {
case STAT_SUCCESS:
    vp = pdu->variables;
    if (pdu->errstat == SNMP_ERR_NOERROR) {
        while (vp) {
            snprint_variable(buf, sizeof(buf), vp->name, vp->name_length, vp);
            /*mjpark-----*/
            pValStr = strrchr(buf, ':');
            sprintf(tVal, "%s", pValStr+2);
            getdata = (char *)malloc(1 + vp->val_len);
            memcpy(getdata, vp->val.string, vp->val_len);
            getdata[vp->val_len] = '\0';
            sVal = trimwhitespace(replace_str2(tVal, "\",\"", ""));

            switch(vp->type)
            {
                case ASN_INTEGER:
                {
                    memcpy((void *)&nVal, getdata, sizeof(int));
                    psnmp->val = nVal;
                    break;
                }
                case ASN_OCTET_STR:
                case ASN_BIT_STR:
                {
                    sprintf(psnmpstr->val, "%s", sVal);
                    break;
                }
                case ASN_GAUGE:
                case ASN_OPAQUE:
                case ASN_COUNTER:
                case ASN_TIMETICKS:
                case ASN_IPADDRESS:
                default :
                    sscanf(tVal, "%lg", &psnmp->val);
            }
            /*-----*/
            vp = vp->next_variable;
        }
        free(getdata);
    }
    else {
        for (ix = 1; vp && ix != pdu->errindex; vp = vp->next_variable, ix++);
        if (vp)
            snprint_objid(buf, sizeof(buf), vp->name, vp->name_length);
        else
            strcpy(buf, "(none)");
        fprintf(stdout, "%s: %s: %s\n", sp->peername, buf, snmp_errstring(pdu->errstat));
    }
    return 1;
}

```

```

case STAT_TIMEOUT:
    fprintf(stdout, "%s: Timeout\n", sp->peername);
    return 0;

case STAT_ERROR:
    snmp_perror(sp->peername);
    return 0;
}
return 0;
}

```

5. 제어를 위해 AsyncApp에 없는 snmpset을 추가한다.

```

int set_snmp(void *precord, const char *sval)
{
    snmpRecord *psnmp = (snmpRecord *)precord;
    SNMP_INFO *snmpinfo = (SNMP_INFO*)psnmp->dpvt;
    char type = 'i';

    if (!(snmpinfo->sess = snmp_open(&snmpinfo->ss))) {
        snmp_perror("snmp_open");
        return -1;
    };

    static int count = 0;
    printf("SValue[%d]: %s\n", count++, sval);

    snmpinfo->setreq = snmp_pdu_create(SNMP_MSG_SET); /* send the first GET */
    if (! snmpinfo->setreq) {
        snmp_free_pdu(snmpinfo->setreq);
        snmp_close(snmpinfo->sess); /* cleanup */
        return -1;
    }

    /* snmp_add_var(netsnmp_pdu *, const oid *, size_t, char, const char *) */
    snmp_add_var(snmpinfo->setreq, snmpinfo->oid_info.Oid, snmpinfo->oid_info.OidLen, type,
        sval);

    if (snmp_send(snmpinfo->sess, snmpinfo->setreq))
        hosts++;
    else {
        snmp_perror("snmp_setsend");
        snmp_free_pdu(snmpinfo->setreq);
        snmp_close(snmpinfo->sess); /* cleanup */
        return -1;
    }

    active_hosts();
    snmp_close(snmpinfo->sess); /* cleanup */
    return 0;
}

```

## dbd

코드가 작성되면 dset에 맞춰 dbd파일을 작성한다.

```

include "snmpRecord.dbd"
include "snmpstrRecord.dbd"

device(snmp, INST_IO, devSNMPRead, "SNMP Read")

```

```
device(snmp,INST_IO,devSNMPWrite,"SNMP Write")
device(snmpstr,INST_IO,devSNMPRead_str,"SNMP Read")
```

## Makefile

위의 과정을 거쳐 수정 및 생성한 파일들을 라이브러리로 만들어 EPICS IOC에서 실행하기 위해 Makefile을 수정한다. 라이브러리로 사용할 이름을 설정하고, 빌드할 파일들을 설정한다.

```
1 TOP = ../../
2 include $(TOP)/configure/CONFIG
3
4 LIBRARY += snmpMon
5 DBDINC      += snmpRecord snmpstrRecord
6 DBD         += snmpRecord.dbd snmpstrRecord.dbd
7 DBD         += snmpDevSoft.dbd
8 snmpMon_SRCS += snmpDevAsync.c snmpRecord.c snmpstrRecord.c
9
10 netsnmp_DIR = /usr/lib
11 # for jessie, need to check for Wheezy
12 # jhlee
13 netsnmp_DIR += /usr/lib/x86_64-linux-gnu
14 snmpMon_LIBS +=netsnmp
15
16 # need to check where it is in Wheezy
17
18 USR_CFLAGS += 'net-snmp-config --cflags'
19 USR_LDFLAGS += 'net-snmp-config --libs'
20 PROD_LDLIBS += 'net-snmp-config --libs'
21
22 include $(TOP)/configure/RULES
```

Makefile 수정 후 make를 실행하면 아래와 같이 lib, include, dbd폴더와 파일들이 생성된다.

```
mijoy0909@mjpark:~/epics/R3.14.12.4/siteLibs$ tree -L 3
```

```
├── [ 4096] dbd
│   ├── [ 445] snmpDevSoft.dbd
│   ├── [ 5926] snmpRecord.dbd
│   └── [ 3024] snmpstrRecord.dbd
├── [ 4096] include
│   ├── [ 19578] snmpRecord.h
│   └── [ 14389] snmpstrRecord.h
├── [ 4096] lib
│   ├── [ 4096] linux-x86_64
│   │   ├── [ 165542] libsnmpMon.a
│   │   └── [ 103574] libsnmpMon.so
├── [ 186] Makefile
├── [ 345] Makefile.template
├── [ 4096] README
├── [ 4096] snmpLib
│   ├── [ 305] Makefile
│   ├── [ 4096] mibs
│   ├── [ 4096] rawcode
│   └── [ 4096] src
│       ├── [ 628] Makefile
│       ├── [ 4096] 0.Common
│       └── [ 4096] 0.linux-x86_64
```

```

|      |----- [      27789]  snmpDevAsync.c
|      |----- [      445]  snmpDevSoft.dbd
|      |----- [      7086]  snmpRecord.c
|      |----- [      5926]  snmpRecord.dbd
|      |----- [      7803]  snmpstrRecord.c
|      |----- [      3024]  snmpstrRecord.dbd
|      |----- [      4096]  test

```

### 3.3 Application 구성

만들어진 Library를 사용하여 EPICS IOC를 실행하기 위한 Application을 구성한다.

#### Application 환경 폴더생성

Application 구성을 위해 EPICS에서 제공되는 makeBaseApp.pl을 사용하여 Application 환경폴더를 생성한다.

```

mijoy0909@mjpark:~/epics/R3.14.12.4/siteApps/snmp2$ makeBaseApp.pl -t ioc snmp2
mijoy0909@mjpark:~/epics/R3.14.12.4/siteApps/snmp2$ makeBaseApp.pl -i -t ioc snmp2
Using target architecture linux-x86_64 (only one available)
The following applications are available: snmp2
What application should the IOC(s) boot?
The default uses the IOC's name, even if not listed above.
Application name? snmp2
mijoy0909@mjpark:~/epics/R3.14.12.4/siteApps/snmp2$ ls
configure/ iocBoot/ Makefile snmp2App/

```

Application 환경을 만든 후 make를 하면 아래와 같이 bin, dbd 폴더가 생성된다.

```

mijoy0909@mjpark:~/epics/R3.14.12.4/siteApps/snmp2$ make
mijoy0909@mjpark:~/epics/R3.14.12.4/siteApps/snmp2$ ls
bin/ configure/ dbd/ iocBoot/ Makefile snmp2App/

```

#### src/Makefile 수정

환경을 구성한 후 앞서 만든 library를 사용하기 위해 siteApps/snmp2/snmp2App/src/Makefile을 수정한다.

```

1 TOP=../..
2
3 include $(TOP)/configure/CONFIG
4
5
6 USR_INCLUDES += -I${RAON_SITELIBS}/include/
7 USR_DBDFLAGS += -I${RAON_SITELIBS}/dbd/
8 USR_INCLUDES += -I$(EPICS_EXTENSIONS)/include
9
10 snmpMon_DIR += ${RAON_SITELIBS}/lib/$(T_A)
11
12 #-----
13 # ADD MACRO DEFINITIONS AFTER THIS LINE
14 #=====
15
16 #=====
17 # Build the IOC application
18

```



```

19 PROD_IOC = snmp2
20 # snmp2.dbd will be created and installed
21 DBD += snmp2.dbd
22
23 # snmp2.dbd will be made up from these files:
24 snmp2_DBD += base.dbd
25 snmp2_DBD += snmpRecord.dbd
26
27 # Include dbd files from all support applications:
28 #snmp2_DBD += xxx.dbd
29 snmp2_DBD += snmpDevSoft.dbd
30
31 # Add all the support libraries needed by this IOC
32 #snmp2_LIBS += xxx
33 snmp2_LIBS += snmpMon
34
35
36 # snmp2_registerRecordDeviceDriver.cpp derives from snmp2.dbd
37 snmp2_SRCS += snmp2_registerRecordDeviceDriver.cpp
38
39 # Build the main IOC entry point on workstation OSs.
40 snmp2_SRCS_DEFAULT += snmp2Main.cpp
41 snmp2_SRCS_vxWorks += -nil-
42
43 # Add support from base/src/vxWorks if needed
44 #snmp2_OBJS_vxWorks += $(EPICS_BASE_BIN)/vxComLibrary
45
46 # Finally link to the EPICS Base libraries
47 snmp2_LIBS += $(EPICS_BASE_IOC_LIBS)
48
49 #=====
50
51 include $(TOP)/configure/RULES
52 #-----
53 # ADD RULES AFTER THIS LINE

```

## db파일 생성

EPIC IOC를 통해 모니터링 및 제어를 원하는 장비의 MIB정보를 record로 만들어 db 파일을 생성한다. 아래는 WIENER Crate db파일의 예이다. 레코드의 Read/Write여부를 결정하고, 그에 맞춰 버전과 인증비밀번호, Host, Communiy, OID의 정보를 필드로 추가한다.

```

record(snmpp, "${W}:${C}_MainPower_W") {
    field(DESC, "WIENER Main Power Switch")
    field(DTYP, "SNMP Write")
    field(SCAN, "5 second")
    field(VERS, "${V3}")
    field(AUTH, "${AUTH_P}")
    field(PRIV, "${PRIV_P}")
    field(HOST, "${HOST}")
    field(COMM, "${USER}")
    field(OIDS, "${WI}sysMainSwitch.0")
    field(MDEL, "-1")
    field(RVAL, "1")
}

record(snmpp, "${W}:${C}_MainPower_R") {
    field(DESC, "WIENER Main Power Switch")

```

```

    field(DTYP, "SNMP Read")
    field(SCAN, "5 second")
    field(VERS, "${V2C}")
    field(HOST, "${HOST}")
    field(COMM, "${COM}")
    field(OIDS, "${WI}sysMainSwitch.0")
    field(MDEL, "-1")
}

```

## Db/Makefile 수정

생성된 db파일을 사용하기 위해 siteApps/snmp2/snmp2App/Db의 Makefile을 수정한다.

```

1 TOP=../..
2 include $(TOP)/configure/CONFIG
3 #-----
4 #  ADD MACRO DEFINITIONS AFTER THIS LINE
5
6 #-----
7 #  Optimization of db files using dbst (DEFAULT: NO)
8 #DB_OPT = YES
9
10 #-----
11 #  Create and install (or just install) into <top>/db
12 #  databases, templates, substitutions like this
13 #DB += xxx.db
14 DB += wiener.vdb
15
16 #-----
17 #  If <anyname>.db template is not named <anyname>*.template add
18 #  <anyname>_template = <templatename>
19
20 include $(TOP)/configure/RULES
21 #-----
22 #  ADD RULES AFTER THIS LINE

```

## IOC 실행

EPICS IOC 실행에 앞서 siteApps/snmp2에서 make를 실행한다. 실행 후 문제가 없다면 iocBoot/iocsnmp2로 이동하여 st.cmd에 db파일을 추가한다.

```

1 #!../bin/linux-x86_64/snmp2
2
3 ## You may have to change snmp2 to something else
4 ## everywhere it appears in this file
5
6 < envPaths
7
8 cd ${TOP}
9
10 epicsEnvSet("MIBDIRS", "${EPICS_PATH}/siteLibs/snmpLib/mibs")
11
12 ## Register all support components
13 dbLoadDatabase "dbd/snmp2.dbd"
14 snmp2_registerRecordDeviceDriver pdbname
15
16 ## Load record instances
17 #dbLoadRecords("db/xxx.db", "user=mijoy0909Host")

```

```

18 dbLoadRecords("db/wiener.vdb", "W=WIENER, C=CRATE3, HOST=10.1.5.123, COM=public, USER=admin, V2C=
    SNMP_VERSION_2c, V3=SNMP_VERSION_3, WI=WIENER-CRATE-MIB::, AUTH_P=MySecret, PRIV_P=MySecret")
19
20 cd ${TOP}/iocBoot/${IOC}
21 iocInit
22
23 ## Start any sequence programs
24 #seq sncxxx,"user=mijoy0909Host"

```

위의 과정이 모두 완료되면 st.cmd에 실행권한을 부여하고 실행시킨다.

```
mijoy0909@mjpark:~/epics/R3.14.12.4/siteApps/snmp2/iocBoot/iocsnmp2$ chmod 755 st.cmd
```

EPICS IOC 실행 결과는 다음과 같다.

```

mijoy0909@mjpark:~/epics/R3.14.12.4/siteApps/snmp2/iocBoot/iocsnmp2$ ./st.cmd
#!../bin/linux-x86_64/snmp2
## You may have to change snmp2 to something else
## everywhere it appears in this file
< envPaths
epicsEnvSet("ARCH","linux-x86_64")
epicsEnvSet("IOC","iocsnmp2")
epicsEnvSet("TOP","/home/mijoy0909/epics/R3.14.12.4/siteApps/snmp2")
cd /home/mijoy0909/epics/R3.14.12.4/siteApps/snmp2
#epicsEnvSet("MIBDIRS", "${EPICS_PATH}/siteLibs/snmpLib/mibs")
## Register all support components
dbLoadDatabase "dbd/snmp2.dbd"
snmp2_registerRecordDeviceDriver pdbname
#drvSNMPConnInit("10.1.5.142", "qwertyuiop12345", "qwertyuiop12345");
#drvSNMPConnInit("SNMP_VERSION_2c", "10.1.5.142", "public", NULL, NULL);
## Load record instances
#dbLoadRecords("db/xxx.db", "user=mijoy0909Host")
dbLoadRecords("db/pdu.vdb", "A=APC, P=PDU2, HOST=10.1.5.142, COM=public, USER=mijoy, V2C=
    SNMP_VERSION_2c, V3=SNMP_VERSION_3, P0=PowerNet-MIB::, AUTH_P=qwertyuiop12345, PRIV_P=
    qwertyuiop12345")
dbLoadRecords("db/wiener.vdb", "W=WIENER, C=CRATE3, HOST=10.1.5.123, COM=public, USER=admin, V2C=
    SNMP_VERSION_2c, V3=SNMP_VERSION_3, WI=WIENER-CRATE-MIB::, AUTH_P=MySecret, PRIV_P=MySecret")
dbLoadRecords("db/timestamp.vdb", "S=SNMPIOC")
cd /home/mijoy0909/epics/R3.14.12.4/siteApps/snmp2/iocBoot/iocsnmp2
iocInit
Starting iocInit
#####
## EPICS R3.14.12.4 $Date: Mon 2013-12-16 15:51:45 -0600$
## EPICS Base built Aug 29 2014
#####
iocRun: All initialization complete
## Start any sequence programs
#seq sncxxx,"user=mijoy0909Host"
epics>
epics> db1
WIENER:CRATE3_FanNominalSpeed_R
WIENER:CRATE3_FanNominalSpeed_W

```

EPICS IOC의 PV와 장비의 통신을 확인한다. IOC 내에서 dbpr 명령어로 pv의 값을 확인한 후 dbpf 명령어로 값을 변경했을 때 값이 변경되는지 확인한다. 또한 camonitor 및 snmpget 통해 값의 변경을 확인한다.

```

epics> dbpr WIENER:CRATE3_FanNominalSpeed_R
ASG:          DESC: WIENER Fan rotation speed          DISA: 0
DISP: 0        DISV: 1          NAME: WIENER:CRATE3_FanNominalSpeed_R
OVAL: 0        RVAL: 0          SEVR: NO_ALARM         STAT: NO_ALARM
SVAL: 0        TPRO: 0          VAL: 2000

```

```

epics> dbpf WIENER:CRATE3_FanNominalSpeed_W 2500
DBR_DOUBLE:      2500
epics> dbpr WIENER:CRATE3_FanNominalSpeed_W
ASG:              DESC: WIENER Fan rotation speed          DISA: 0
DISP: 0           DISV: 1          NAME: WIENER:CRATE3_FanNominalSpeed_W
OVAL: 2000        RVAL: 0          SEVR: NO_ALARM          STAT: NO_ALARM
SVAL: 0           TPRO: 0          VAL: 2500
epics> dbpr WIENER:CRATE3_FanNominalSpeed_R
ASG:              DESC: WIENER Fan rotation speed          DISA: 0
DISP: 0           DISV: 1          NAME: WIENER:CRATE3_FanNominalSpeed_R
OVAL: 0           RVAL: 0          SEVR: NO_ALARM          STAT: NO_ALARM
SVAL: 0           TPRO: 0          VAL: 2500

mijoy0909@mjpark:~$ camonitor WIENER:CRATE3_FanNominalSpeed_R WIENER:CRATE3_FanNominalSpeed_W
WIENER:CRATE3_FanNominalSpeed_R 2015-03-24 09:57:20.552400 2000
WIENER:CRATE3_FanNominalSpeed_W 2015-03-24 09:57:45.593398 2500
WIENER:CRATE3_FanNominalSpeed_R 2015-03-24 09:57:50.575369 2500

mijoy0909@mjpark:~$ snmpget -v 2c -c public 10.1.5.123 fanNominalSpeed.0
WIENER-CRATE-MIB::fanNominalSpeed.0 = INTEGER: 2000 RPM
WIENER-CRATE-MIB::fanNominalSpeed.0 = INTEGER: 2500 RPM

```

그림 15의 WIENER사에서 제공하는 웹 페이지를 통해 값 변경을 확인한다.

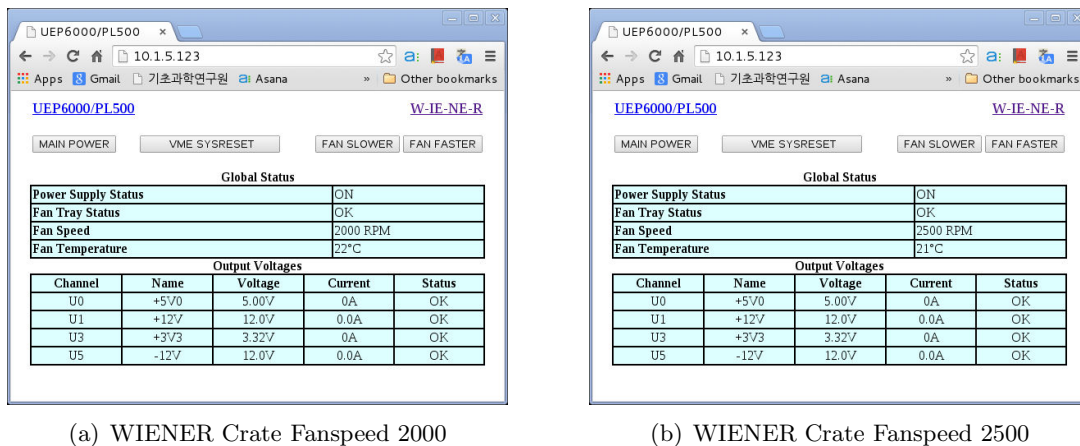


Figure 15 WIENER Crate webpage

모든 값이 같게 바뀌므로 EPICS IOC와 장비가 연결되어 통신함을 알 수 있다. 추가적인 장비의 정보를 레코드로 생성하고 다른 장비에 대한 Db파일을 생성하는 과정을 거치면 하나의 IOC에서 여러 장비의 모니터링 및 제어를 할 수 있다. 위의 integration 과정을 거치면 EPICS와 SNMP 통합 모니터링 시스템이 완성된다.

### 3.4 EPICS Support Module과 차이점

중이온가속기 제어 환경에 맞춰 개발된 통합 모니터링 시스템은 기존의 EPICS IOC Support Module과 달리 library와 application이 분리돼 library의 재사용이 가능하며, 기존의 여러 종류의 Record 대신 snmpRecord, snmpstrRecord 만으로 다양한 SNMP

의 데이터 타입을 지원한다. 따라서, INP/OUT 필드에 데이터 타입에 대한 마스크와 datalength를 정의하지 않아도 된다. 또한, Support Module에서 지원하지 않던 SNMPv3가 지원되고, Support Module에서 호스트 주소에 하나의 버전만 지정되던 것과 달리 레코드에 사용자가 원하는 버전에 맞춰 필드를 사용하여 장비에 대해 여러 버전의 snmp를 사용해 Record를 생성 할 수 있다.

### 3.5 안정성 테스트

완성된 통합 모니터링 시스템의 안정성을 테스트하기 위해 PV의 값을 주기적으로 변경하고, 변경된 PV데이터는 EPICS Extension 툴인 Archiver Appliance를 사용해 저장했다.

#### 테스트 Script

WIENER Crate의 FanSpeed와 APC PDU의 Outlet 8번의 상태에 대한 PV값을 주기적으로 변경하는 Bash Script를 아래와 같이 작성한다.

```
#!/bin/bash

while [ : ]; do
#++

for i in {1200..3000..100};do
stat='caget APC:PDU2_Outlet8_R'
echo "*****"
caget SNMPIOC:TIMESTAMP

echo "-----"
echo "snmpget -----"
snmpget -v 1 -c public 10.1.5.123 WIENER-CRATE-MIB::sysMainSwitch.0 WIENER-CRATE-MIB::sysStatus.0
WIENER-CRATE-MIB::fanAirTemperature.0 WIENER-CRATE-MIB::fanNominalSpeed.0
snmpget -v 1 -c public 10.1.5.142 PowerNet-MIB::sPDUOutletCtl.8
echo "-----"
echo "caget -----"
caget WIENER:CRATE3_MainPower_R WIENER:CRATE3_PS_Status WIENER:CRATE3_FanairTemp WIENER:
CRATE3_FanNominalSpeed_R WIENER:CRATE3_FanNominalSpeed_W APC:PDU2_Outlet8_R APC:
PDU2_Outlet8_W
echo "caput -----"

case "$stat" in
"APC:PDU2_Outlet8_R 1")
caput APC:PDU2_Outlet8_W 2;;
"APC:PDU2_Outlet8_R 2")
caput APC:PDU2_Outlet8_W 1;;
*)
echo "error";;
esac

caput WIENER:CRATE3_FanNominalSpeed_W $i
sleep 60
done

#--
for i in {2900..1300..100};do
stat='caget APC:PDU2_Outlet8_R'
```

```

echo "*****"
caget SNMPIOC:TIMESTAMP
echo "-----"
echo "snmpget -----"
snmpget -v 1 -c public 10.1.5.123 WIENER-CRATE-MIB::sysMainSwitch.0 WIENER-CRATE-MIB::sysStatus.0
WIENER-CRATE-MIB::fanAirTemperature.0 WIENER-CRATE-MIB::fanNominalSpeed.0
snmpget -v 1 -c public 10.1.5.142 PowerNet-MIB::sPDUOutletCtl.8
echo "-----"
echo "caget -----"
caget WIENER:CRATE3_MainPower_R WIENER:CRATE3_PS_Status WIENER:CRATE3_FanairTemp WIENER:
CRATE3_FanNominalSpeed_R WIENER:CRATE3_FanNominalSpeed_W APC:PDU2_Outlet8_R APC:
PDU2_Outlet8_W
echo "caput -----"

case "$stat" in
"APC:PDU2_Outlet8_R" 1)
caput APC:PDU2_Outlet8_W 2;;
"APC:PDU2_Outlet8_R" 2)
caput APC:PDU2_Outlet8_W 1;;
*)
echo "error";;
esac

caput WIENER:CRATE3_FanNominalSpeed_W $i
sleep 60
done
done
done

```

Script는 1분 주기로 Crate의 FanSpeed를 100단위로 1200에서 3000까지 올렸다, 내리고 PDU Outlet 8번의 전원을 켜다 끈다. 이때 snmp명령어와 camonitor로 통신 되는 값을 모두 확인하고 Script의 결과는 아래와 같은 형식으로 log파일에 저장된다.

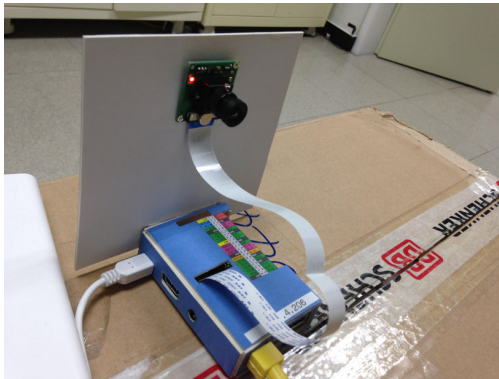
```

*****
SNMPIOC:TIMESTAMP 2015/03/24 19:07:23
-----
snmpget -----
WIENER-CRATE-MIB::sysMainSwitch.0 = INTEGER: ON(1)
WIENER-CRATE-MIB::sysStatus.0 = BITS: 80 mainOn(0)
WIENER-CRATE-MIB::fanAirTemperature.0 = INTEGER: 22 \ufffdC
WIENER-CRATE-MIB::fanNominalSpeed.0 = INTEGER: 2400 RPM
PowerNet-MIB::sPDUOutletCtl.8 = INTEGER: outletOn(1)
-----
caget -----
WIENER:CRATE3_MainPower_R 1
WIENER:CRATE3_PS_Status 80 mainOn(0)
WIENER:CRATE3_FanairTemp 22
WIENER:CRATE3_FanNominalSpeed_R 2400
WIENER:CRATE3_FanNominalSpeed_W 2400
APC:PDU2_Outlet8_R 1
APC:PDU2_Outlet8_W 1
caput -----
Old : APC:PDU2_Outlet8_W 1
New : APC:PDU2_Outlet8_W 2
Old : WIENER:CRATE3_FanNominalSpeed_W 2400
New : WIENER:CRATE3_FanNominalSpeed_W 2500

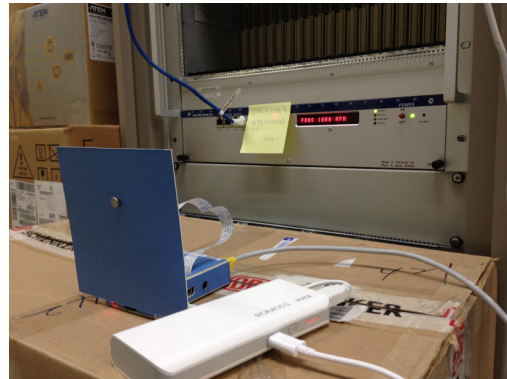
```

## Raspberry Pi를 활용한 모니터링

스크립트가 실행될 때 변경되는 값이 실제 Crate의 FanSpeed에 영향을 미치는지 확인하기 위해 Raspberry Pi의 카메라 모듈을 활용하여 그림 16과 같이 Crate의 패널을 영상으로 모니터링한다.



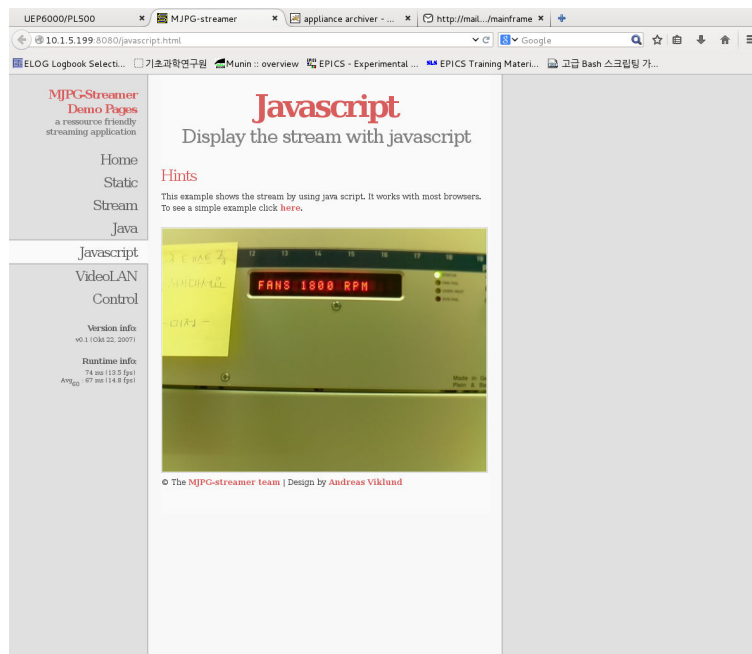
(a) Raspberry Pi Camera Module



(b) Raspberry Pi Monitoring Environment

**Figure 16** Monitoring using Raspberry Pi Camera

카메라 모듈로 촬영되는 영상은 Web Streaming을 통해 그림 17과 같이 웹페이지로 볼 수 있다.



**Figure 17** R.pi 카메라 모듈 모니터링 페이지



## Archiver Appliance

EPICS의 extension틀인 Archiver Appliance 로 주기적으로 변하는 PV 데이터를 저장하고 모니터링한다. Archiver Appliance의 Archiver Viewer로 Crate의 상태, 온도 및 FanSpeed와 PDU Outlet 8번의 상태를 그림 18과 같이 확인한다.

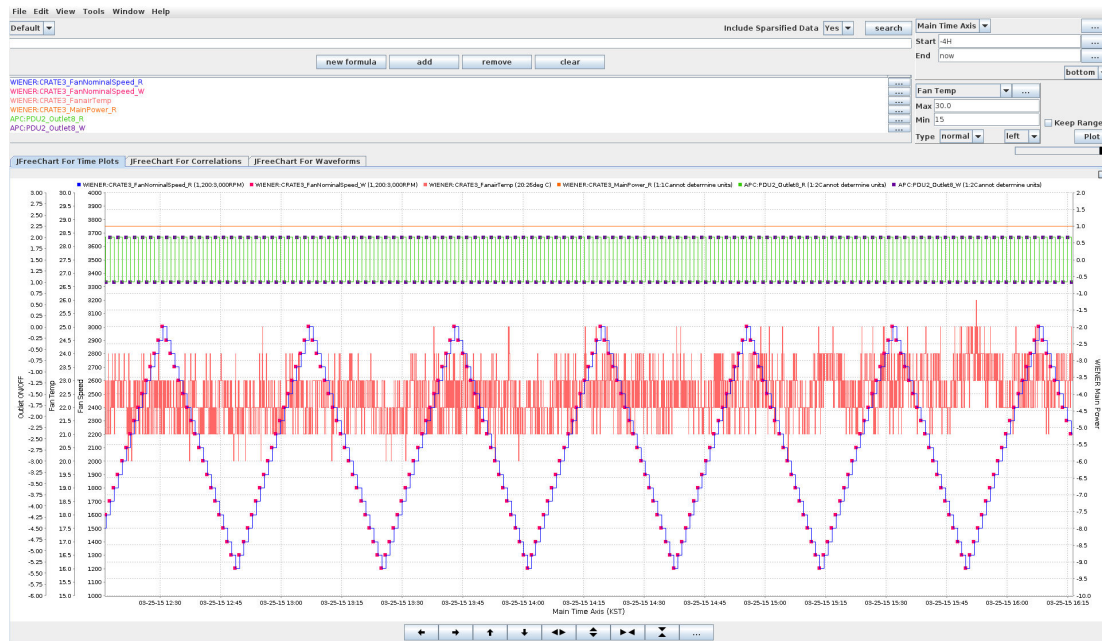


Figure 18 Archiver Appliance Viewer

Archiver Appliance의 모니터링 구성은 그림19와 같다.

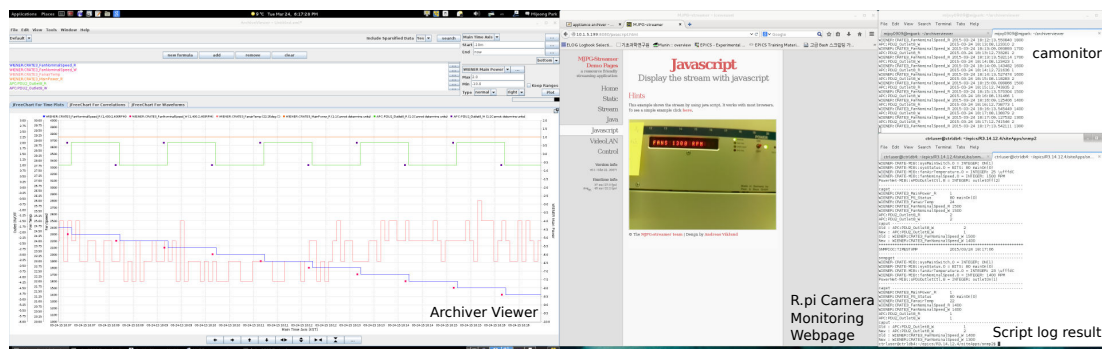


Figure 19 Archiver Appliance Environment



## 안정성 테스트 결과

테스트 후 쓸 것  
테스트 후 쓸 것  
테스트 후 쓸 것  
테스트 후 쓸 것  
테스트 후 쓸 것  
테스트 후 쓸 것

## 3.6 OPI 구현

OPI 구현에는 EPICS의 UI (User Interface) 프로그램인 CSS (Control System Studio)를 사용한다. CSS는 가속기 사이트와 같이 큰 규모의 제어 시스템의 동작 및 모니터링에 사용하는 Eclipse 기반의 툴이다[13]. EPICS IOC의 PV 데이터를 사용자가 원하는 환경에 맞춰 UI를 구현한다. 그림 20은 APC사의 PDU와 WIENER사의 Crate의 PV로 만든 UI다. Crate와 PDU의 PV리스트는 아래 표 1, 2며, PV를 통해 장비 상태 모니터링 및 제어를 할 수 있다[14].

PV Name	Type	Access	Comments
\${W}:\${C}_MainPower_R/W	Integer	R/W	Power ON/OFF
\${W}:\${C}_NomVoltage0,1,3,5_R/W	float	R/W	Channel voltage
\${W}:\${C}_LimCurrent0,1,3,5_R/W	float	R/W	Channel current limit
\${W}:\${C}_Voltage0,1,3,5	float	R	Measured Channel Voltage
\${W}:\${C}_Current0,1,3,5	float	R	Measured Channel current
\${W}:\${C}_SensorTemp1...8	Integer	R	Measured temperature of sensors
\${W}:\${C}_FanairTemp	Integer	R	Air inlet temperature
\${W}:\${C}_sensorWarning Threshold_temp1...8_R/W	Integer	R/W	Warning temperature limit
\${W}:\${C}_sensorFailure Threshold_temp1...8_R/W	Integer	R/W	Over temperature limit
\${W}:\${C}_FanNominalSpeed_R/W	Integer	R/W	Fan speed
\${W}:\${C}_Fanspeed1...3	Integer	R	Measured speed of fan

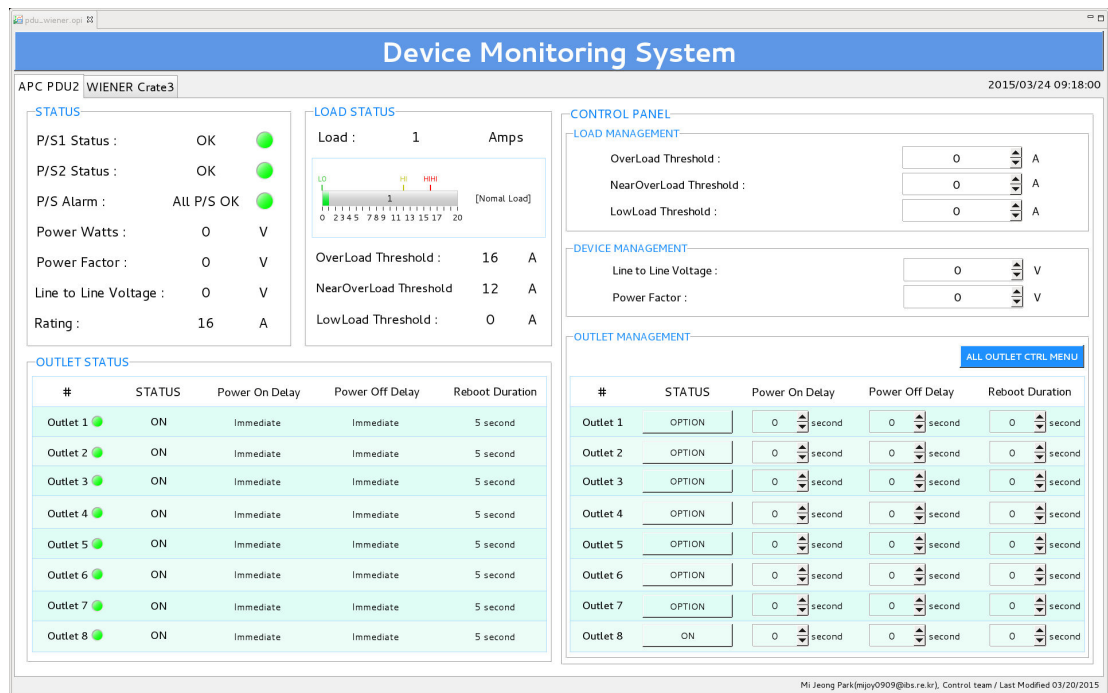
**Table 1** WIENER Crate PV List (R:Read/W:Write)

\* \${W}:\${C}\_-는 WIENER:CRATE3

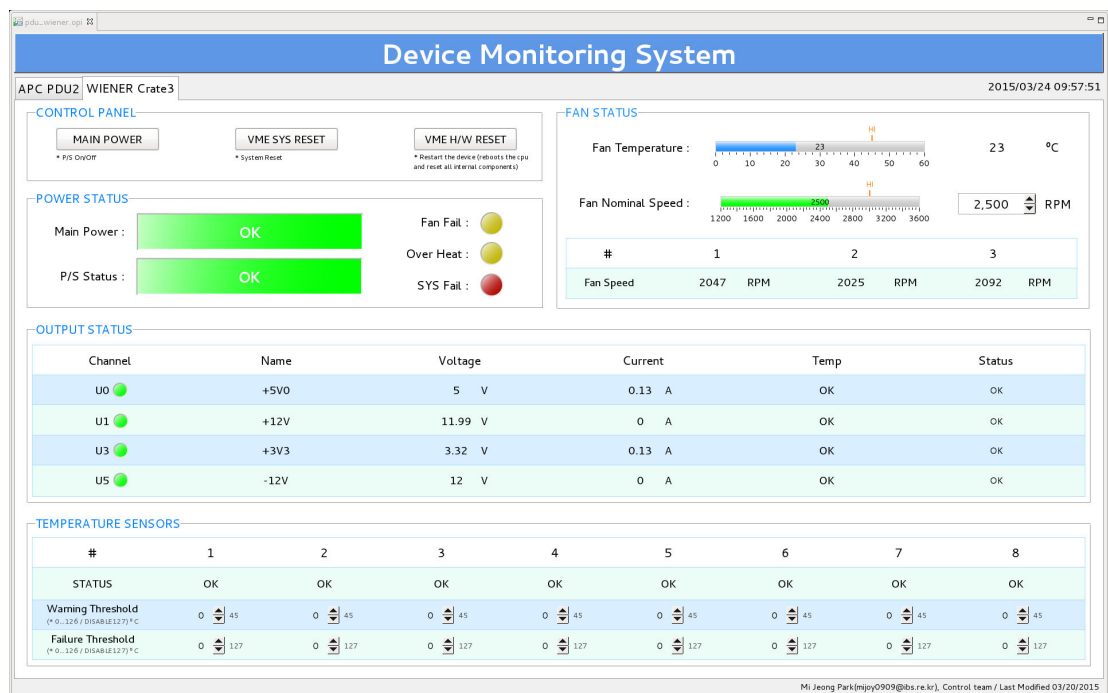
PV Name	Type	Access	Comments
\${A}:\${P}_PowerSupply1Status	Integer	R	PowerSupply1 status
\${A}:\${P}_PowerSupply2Status	Integer	R	PowerSupply2 status
\${A}:\${P}_PowerSupplyAlarm	Integer	R	PowerSupply Alarm
\${A}:\${P}_OutletDevCommand_R/W	Integer	R/W	Setting all outlets
\${A}:\${P}_Outlet1...8_R/W	Integer	R/W	Outlet state
\${A}:\${P}_OutletOnTime1...8_R/W	Integer	R/W	Powering on time
\${A}:\${P}_OutletOffTime1...8_R/W	Integer	R/W	Powering off time
\${A}:\${P}_RebootTime_Outlet1...8_R/W	Integer	R/W	Reboot Duration
\${A}:\${P}_Rating	integer	R	Rating of the device
\${A}:\${P}_LineVoltage_R/W	Integer	R/W	Line to Line Voltage
\${A}:\${P}_PowerWatts	Integer	R	Power in Watts
\${A}:\${P}_PowerFactor_R/W	Integer	R/W	Power Factor
\${A}:\${P}_LowLoadThreshold_R/W	Integer	R/W	LowLoadThreshold
\${A}:\${P}_NearOverloadThreshold_R/W	Integer	R/W	NearOverloadThreshold
\${A}:\${P}_OverloadThreshold_R/W	Integer	R/W	OverloadThreshold
\${A}:\${P}_PowerLoad	Gauge	R	phase/bank load
\${A}:\${P}_LoadStatus	Integer	R	phase/bank load state
\${A}:\${P}_currentPhaseAlarm	Integer	R	Current Phase Alarm

**Table 2** APC PDU PV List (R:Read/W:Write)

\* \${A}:\${P}\_-는 APC:PDU2



(a) APC PDU UI



(b) WIENER Crate UI

Figure 20 CSS 사용한 UI

## Bibliography

- [1] Rare Isotope Science Project, 2013. Y.K.Kwon et al, “Status of Rare Isotope Science Project in Korea”, Few Body Syst. 54, 961.
- [2] Rare Isotope Science Project, 2006.  
<http://www.risp.re.kr/index.jsp>.
- [3] Official Internet Standards Protocol, 2002.  
<http://www.rfc-editor.org/search/standards.php>.
- [4] Relationship between an Manager and an Agent, 2005. Douglas R. Mauro, Kevin J. Schmidt, “Essential SNMP. O’Reilly”, 2nd ed.
- [5] Experimental Physics and Industrial Control System, 2011.  
<http://www.aps.anl.gov/epics/about.php>.
- [6] Integrate SNMP into EPICS, 2006.  
[http://www-mks2.desy.de/content/e4/e40/e41/e12212/index\\_ger.html](http://www-mks2.desy.de/content/e4/e40/e41/e12212/index_ger.html).
- [7] EPICS IOC Support Modules, 2014.  
<http://aps.anl.gov/epics/modules/index.php>.
- [8] EPICS SNMP Device Support Module (NSCL/FRIB), 2014.  
<https://groups.nsl.mscl.msu.edu/controls/files/devSnmp.html>.
- [9] EPICS Record Reference Manual. Philip Stanley, Janet Anderson, Marty Krammer, “EPICS Record Reference Manual”.
- [10] EPICS Environment Tree, 2015. 4th report....
- [11] Net-SNMP, 2000.  
<http://www.net-snmp.org/>.
- [12] Net-SNMP Tutorials.  
<http://www.net-snmp.org/wiki/index.php/Tutorials>.
- [13] Control System Studio.  
[www.cs-studio.org](http://www.cs-studio.org).
- [14] WIENER PV List, 2015. WIENER, “User’s Manual”.