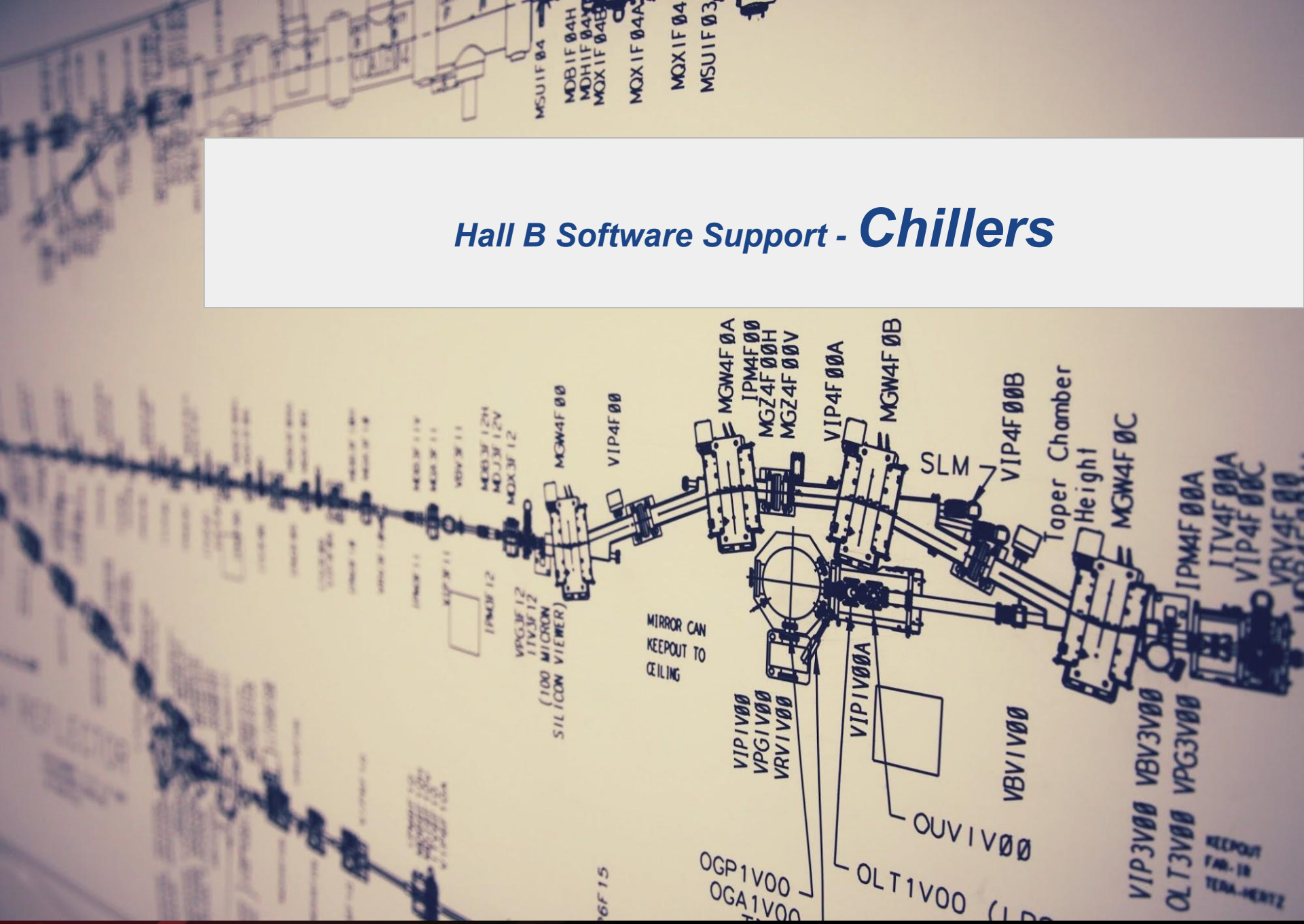
Hall B Software Support - Chillers



Overview

- Hardware
  - Anova A-25 Circulator
  - Configuration
- Software
  - Database
  - streamDevice protocol
- Operator Interfaces

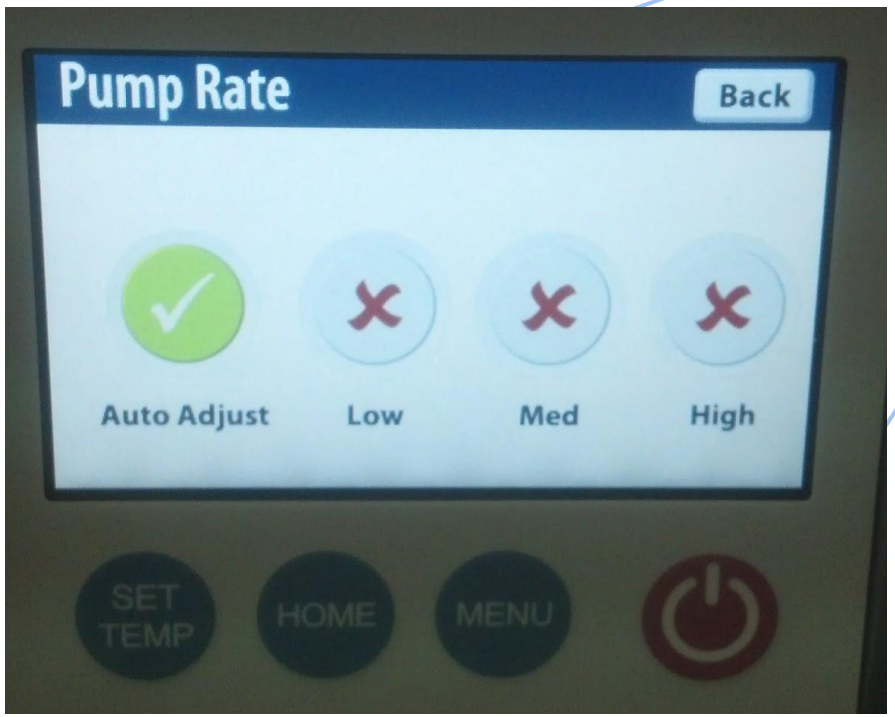
Hardware

Anova A25 Circulator

- -25*C to 200*C, +/-0.01*C
- Tank Volume, 7 liters

Communication

- RS232 interface
- Digi EtherLite (TCP/IP -> RS232)



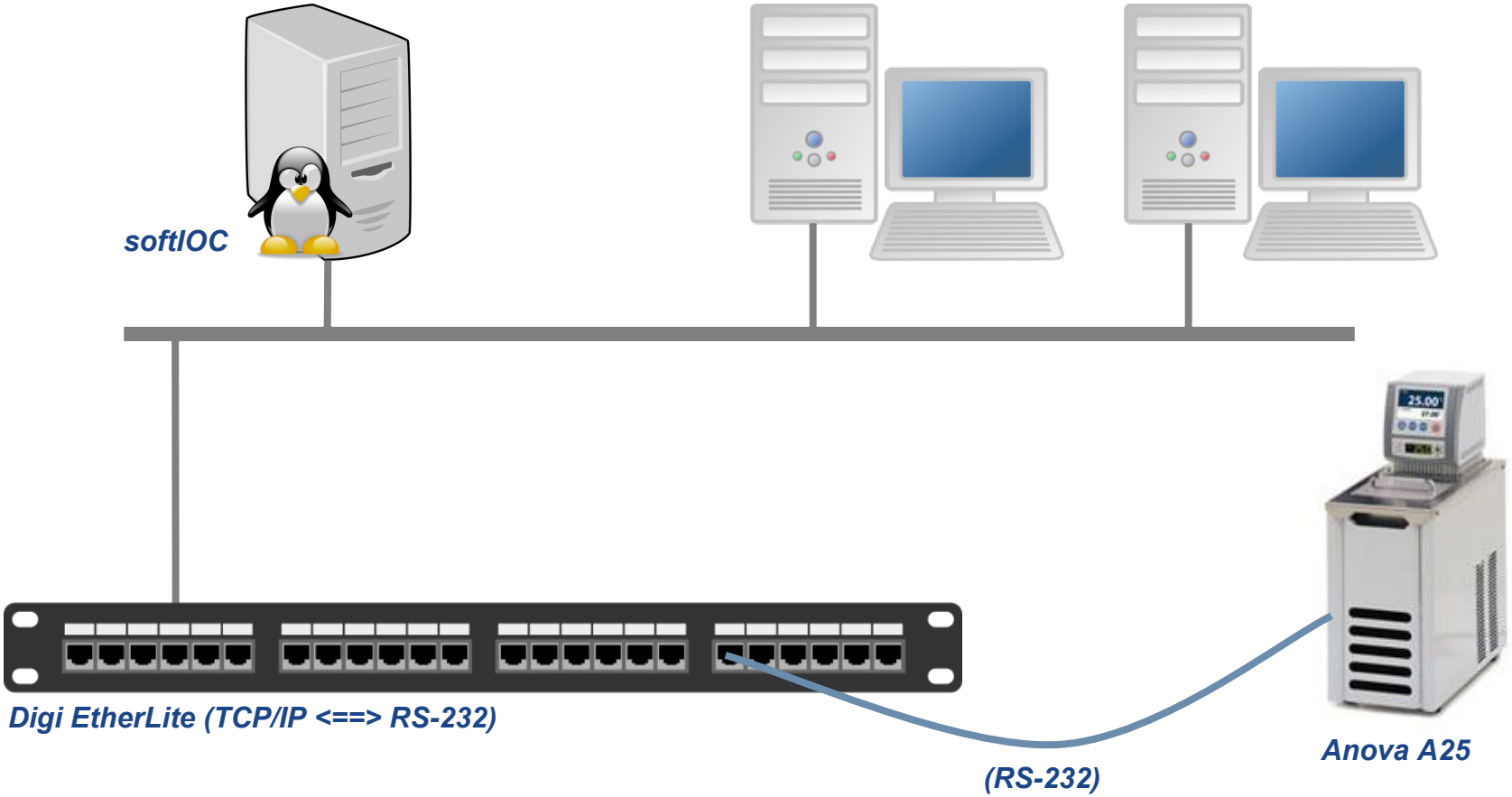
Touchscreen for local control

Located in HPS Test Area

Hardware - Configuration

Communication

- RS-232 interface
- Digi EtherLite (TCP/IP <==> RS232)



Software

- **EPICS 3.14.12.3**
- **asyn** - low level communication driver
- **streamDevice** - “byte stream” interface
- **autosave** - save/restore setpoints, bumpless reboots
- **devIocStats** - ioc health and status, basic control features

Software - EPICS db

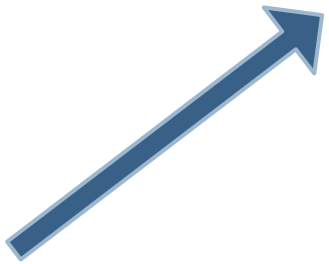
```
#
# Anova Refrigerated and Heating Circulators - A Series
#
#
# Notes:
#   [Macros]
#   P        - standard record prefix
#   R        - standard record prefix
#   PROTO - streams protocol file name
#   PORT  - asyn port name
#
# Author: Wesley Moore
# Date:   June 2014
#

record(asyn, "$(P)$(R)ASYN") {
    field(PORT, "$(PORT)")
    field(OEOS, "\r")
    field(IEOS, "\r")
}

########################################
# Status Commands
########################################

record(stringin, "$(P)$(R)VERSION") {
    field(DTYP, "stream")
    field(INP, "@anova.proto getVersion $(PORT)")
    field(VAL, "")
    field(PINI, "1")
}

record(stringin, "$(P)$(R)STATUS") {
    field(DTYP, "stream")
    field(INP, "@anova.proto getStatus $(PORT)")
    field(VAL, "")
    field(SCAN, "1 second")
}

record(bo, "$(P)$(R)DEFAULT") {
    field(DTYP, "stream")
    field(OUT, "@anova.proto setDefault $(PORT)")
}
```

```
record(ai, "$(P)$(R)TEMP") {
    field(DTYP, "stream")
    field(INP,  "@anova.proto getBathTemp $(PORT)")
    field(PREC, "2")
    field(SCAN, "1 second")
}

########################################
# Control Commands
########################################
record(bo, "$(P)$(R)START") {
    field(DTYP, "stream")
    field(OUT, "@anova.proto start $(PORT)")
}

record(bo, "$(P)$(R)STOP") {
    field(DTYP, "stream")
    field(OUT, "@anova.proto stop $(PORT)")
}

record(bo, "$(P)$(R)CLEAR") {
    field(DTYP, "stream")
    field(OUT, "@anova.proto clear $(PORT)")
}

########################################
# Settings Commands
########################################
record(ai, "$(P)$(R)TEMP_REQD") {
    field(DTYP, "stream")
    field(INP, "@anova.proto getTemp $(PORT)")
    field(PREC, "2")
    field(SCAN, "1 second")
}

record(ao, "$(P)$(R)TEMP_SETPT") {
    field(DTYP, "stream")
    field(OUT, "@anova.proto setTemp $(PORT)")
    field(PREC, "2")
}

...
(additional records not shown)
...
```

Software - streamDevice

Db/anova.db

```
#
# Anova Refrigerated and Heating Circulators - A Series
#
# Notes:
#   [Macros]
#   P          - standard record prefix
#   R          - standard record prefix
#   PROTO - streams protocol file name
#   PORT    - asyn port name
#
# Author: Wesley Moore
# Date:   June 2014
#

record(asyn, "$(P)$(R)ASYN") {
   field(PORT, "$(PORT)")
   field(OEOS, "\r")
   field(IEOS, "\r")
}

########################################
# Status Commands
########################################

record(stringin, "$(P)$(R)VERSION") {
    field(DTYP, "stream")
    field(INP, "@anova.proto getVersion $(PORT)")
    field(VAL, "")
    field(PINI, "1")
}

record(stringin, "$(P)$(R)STATUS") {
   field(DTYP, "stream")
   field(INP, "@anova.proto getStatus $(PORT)")
   field(VAL, "")
   field(SCAN, "1 second")
}

...
(additional records not shown)
...
```



Db/anova.proto

```
...

InTerminator  = "";   # differs for several readbacks...
OutTerminator = CR;
ReadTimeout = 1000;
ReplyTimeout = 1000;

########################################
# Status Commands
########################################
getVersion {
   out "version";
   in  "version\rFirmware Version: %s\r\n\r";
}

getStatus {
   out "status";
   in  "status\r%s\r\n\r";
}

setDefault {
   out "default";
   in  "%*s";
}

getBathTemp {
  out "temp";
  in  "temp\r %f\r";
}

########################################
# Control Commands
########################################
start {
   out "start";
   in  "%*50c";
}

...
(additional protocols not shown)
...
```

Operator Interfaces

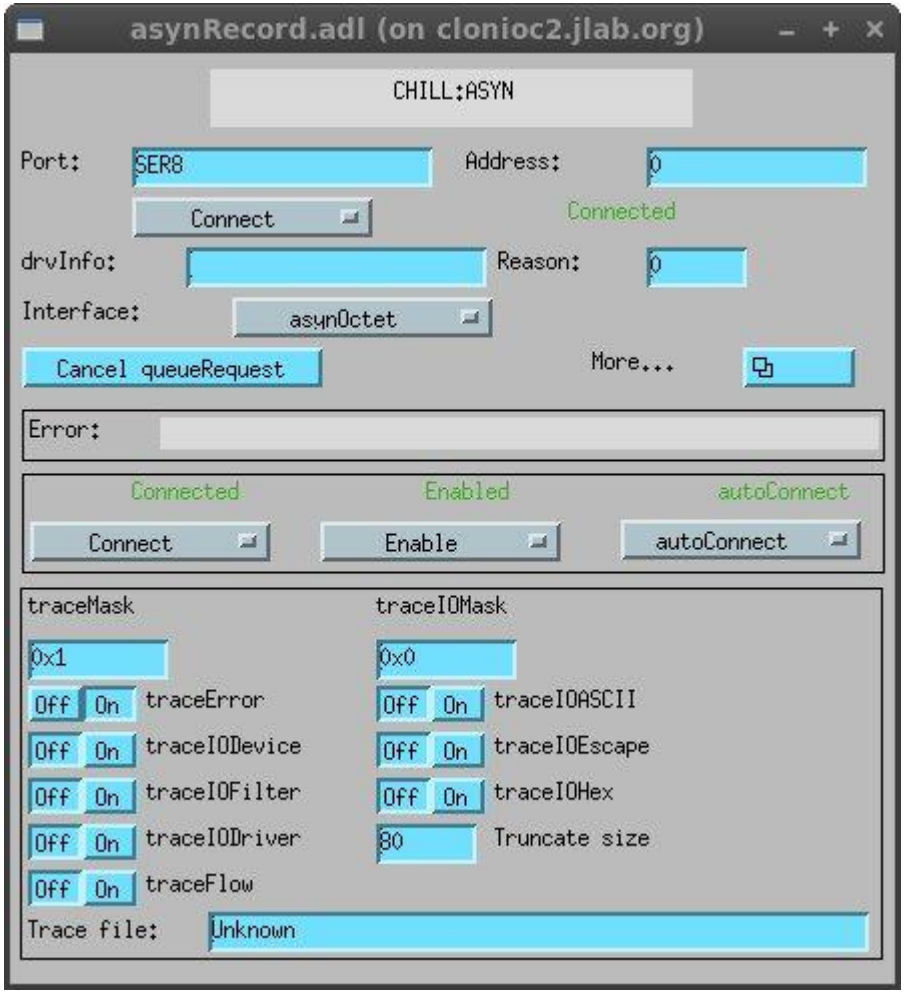
st.cmd

```
#!../../bin/linux-x86/chiller

< envPaths

cd ${TOP}

## Register all support components
dbLoadDatabase("dbd/chiller.dbd")
chiller_registerRecordDeviceDriver(pdbbase)

epicsEnvSet("STREAM_PROTOCOL_PATH","${TOP}/proto")

drvAsynSerialPortConfigure("SER8","/dev/tty_dgrp_D_7",0,0,0)

# debugging...
#asynSetTraceMask("SER8",-1,0x09)
#asynSetTraceIOMask("SER8",-1,0x2)

## Load record instances
dbLoadRecords("db/save_restoreStatus.db", "P=iocchillerTest:")
dbLoadRecords("db/anova.db", "P=CHILL:,R=,PROTO=anova.
proto,PORT=SER8")

cd ${TOP}/iocBoot/${IOC}

# autosave setup
< save_restore.cmd

dbl > pv.list
iocInit

# autosave startup
create_monitor_set("anova_settings.req", 30, "P=CHILL:,R=")

# Handle autosave 'commands' contained in loaded databases.
makeAutosaveFiles()
create_monitor_set("info_positions.req", 5, "P=xxx:")
create_monitor_set("info_settings.req", 30, "P=xxx:")
```

*Need to fix fonts

*autosaved

Operator Interfaces *(cont'd)*



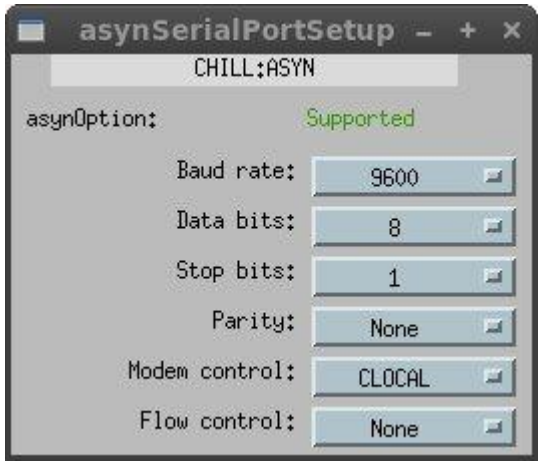
Asyn Screens

Stats

- Port Name
- Port config (baud rate, etc)
- Connection status
- Present error, if any

Controls

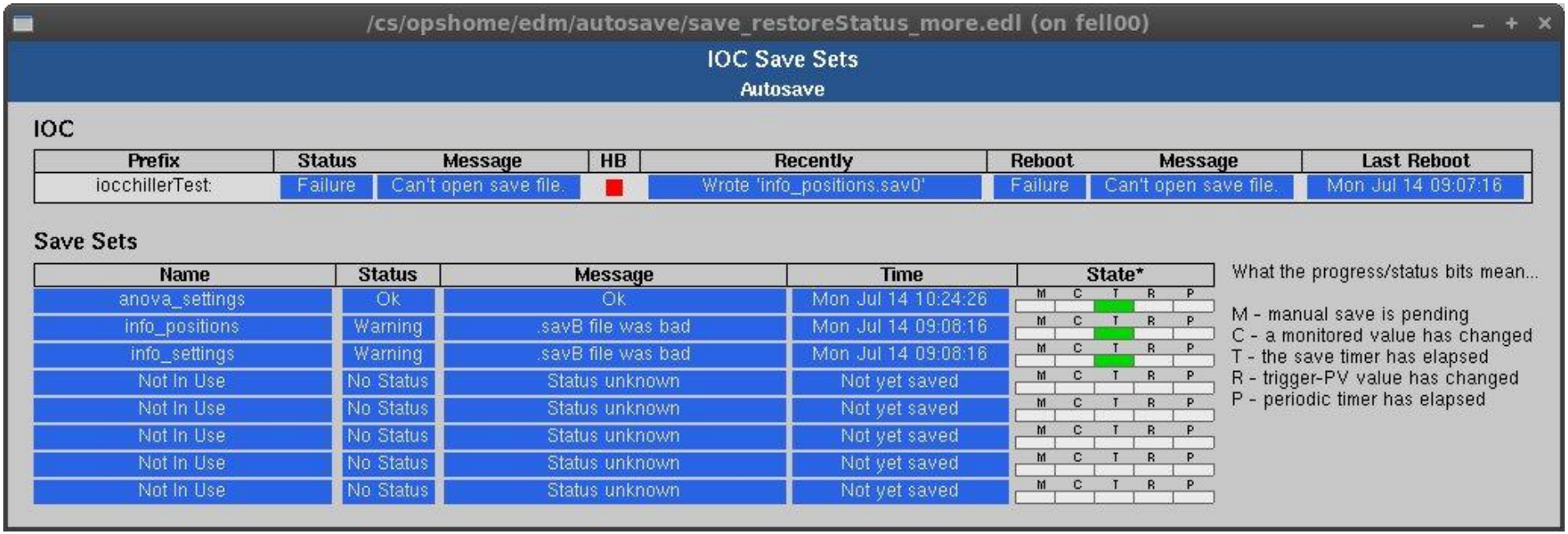
- Port debug levels
- General testing



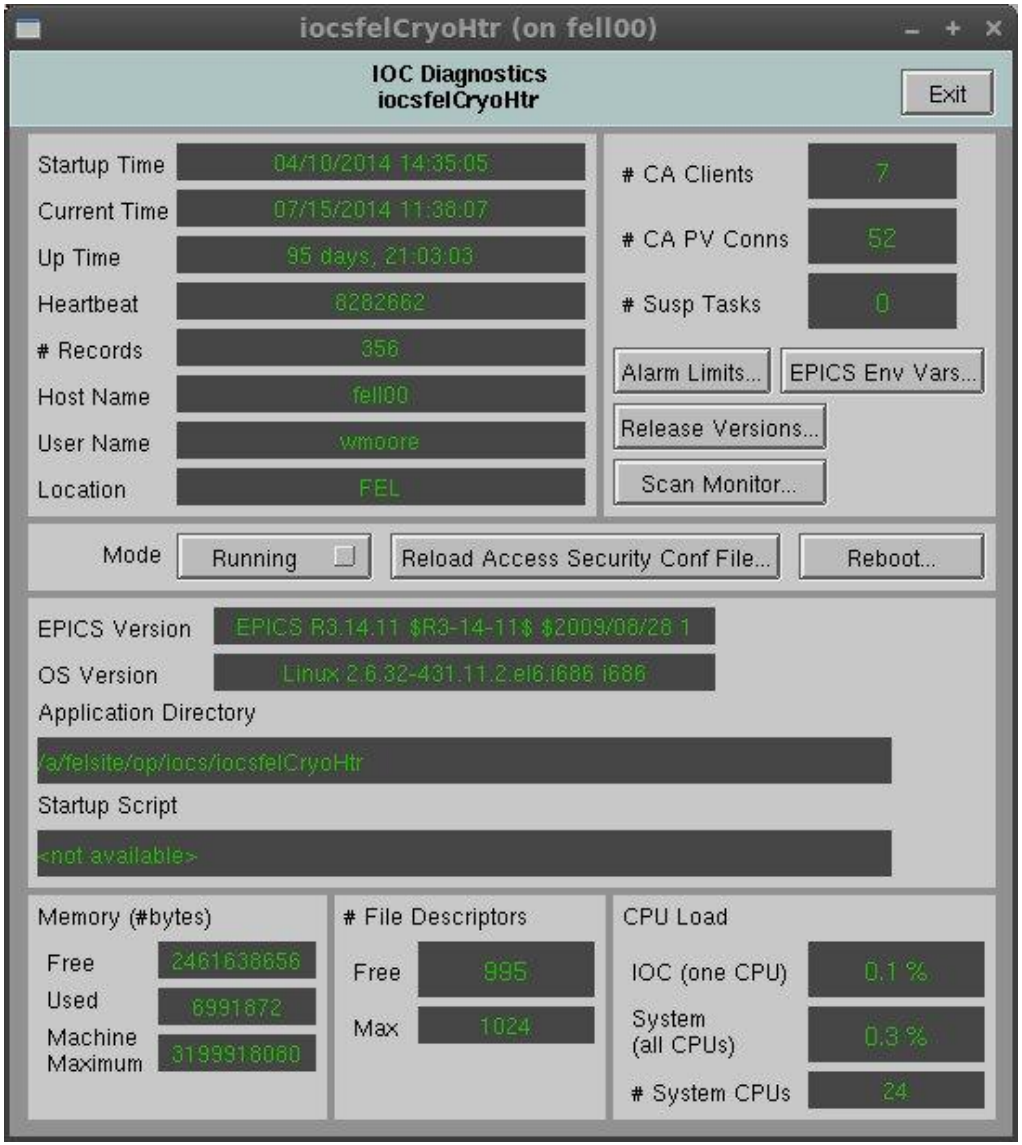
Operator Interfaces *(cont'd)*

Autosave Screens

- Status of save and restore for IOC and each save set. Including:
  - Timestamp of last save
  - Saves pending
  - Restore failure due to IOC being started in production area



Operator Interfaces *(cont'd)*



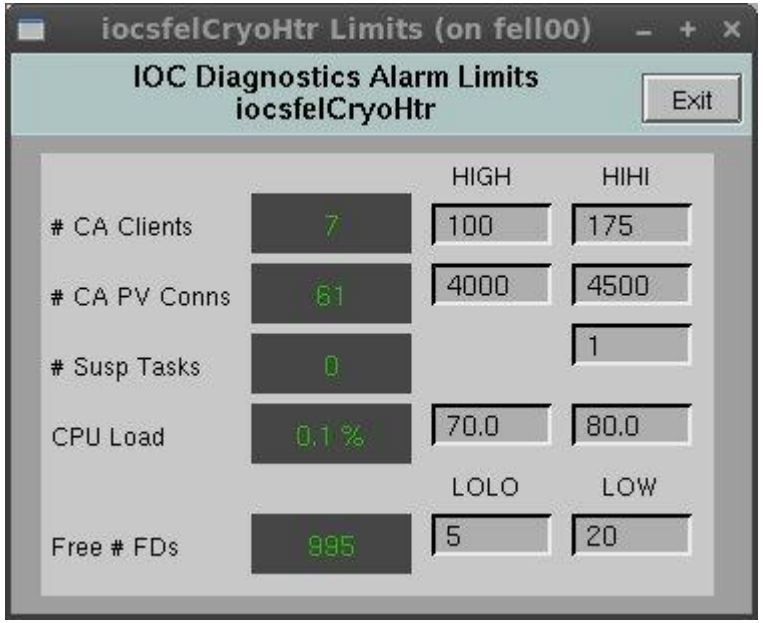
devlocStats Screens *(future addon)*

Stats

- CPU load
- Num. clients
- Num. PV connections

Controls

- Reboot
- Reload access security config



Questions?

(This slide intentionally left blank.)