	Document # **DESI-5227**	Date 2019-09-07	Status DRAFT 3
	Author(s) Laurent Le Guillou		
	Subsystem/Office Calibration System		

Document Title

DESI Calibration System: PDU Control

Change History Log		
1	2018-05-20	First draft version
2	2019-09-07	Release for Klaus H.
3	2019-09-08	minor changes

Contents

Contents

1 Introduction - Motivation

In this document we describe the way to remotely control the DESI calibration lamps through the network interface of their associated Raritan remote power unit (PDU). We focus on the SNMP protocol, which is the native network protocol for this PDU family and for many similar network devices, but the other protocols available (web interface, ssh, serial) to remotely control the calibration rack PDUs are also briefly described.

Warning: this document is still a draft, and some details are missing.

2 Brief description of the DESI calibration system

The DESI calibration system is made of:

- A lambertian screen with a useful diameter of 5173 mm, coated with Permaflect, sold by Labsphere; the screen coating provides a high reflectance (better than 90% from 315 nm to 1380 nm) which is independent of the incident angle. The screen has been delivered and installed on the Mayall dome in 2017.
- Four (4) calibration boxes (see fig 1) to be installed on the upper ring of the telescope, providing 5 arc lamps (Cd, Ne, Kr, Xe, HgAr) and 3 continuum lamps (2 halogens and a home-maded LED-based lamp (*still to be delivered*)) to illuminate the screen. Each box contains an 8-outlet PDU (Raritan PX3-5145R) remotely controlled through ethernet, and removable drawers, one drawer per lamp. Each drawer contains a lamp and its associated power supply, powered by the PDU. A fifth calibration box, fully populated with the same lamps, is also provided as a spare. Each Raritan PDU is equipped with temperature and humidity sensors to protect the unit and forbid turning the lamps on when the environment conditions are outside the allowed range.

The requirements for the calibration system were described in [DESI-1067]. The tests done on the arc lamp candidates are detailed in [DESI-2674]. The detailled design of the calibration system is described in [DESI-1673]. The qualification tests of the illumination system are described in [DESI-3639].

Figure 1: Top: DESI calibration box DESI-CALIB-02 (front side), containing HgAr, Xe, Kr, Ne and Cd arc lamps, and the two halogen lamps. Bottom: A lamp drawer with the high voltage power supply (the lamp bulb is not yet installed), the reflector, and the safety grid.

3 The Raritan Power Distribution Unit (PDU)

Each illumination box is equipped with a 8-outlet PDU from Raritan, model PX3-5145R. Each outlet (numbered [1-8]) may be controlled individually. The PDU monitors the outlet states, the voltage, current, power, frequency, power consumption which may be read remotely. The PDU also monitors its external sensors, here a DPX-T1H1 triple sensor with three (3) temperature and one (1) humidity sensor, and may trigger action and send alarms when any sensor value (of the inlet, any outlet or any environment sensor) is out of a specified range.

The Raritan PDU controls could be accessed through the PDU web interface (On its HTTPS ethernet port, see section 9), as well as through its native SNMP MIB tree (on the 161 ethernet port), and also a command line interface through SSH or USB/RS232 (see section 10). In this document, we will focus on the SNMP protocol and detail the SNMP objects which are useful for the remote control of the DESI calibration boxes and the individual calibration lamps, but also describe briefly how to do the same through the other available protocols.

See [DESI-4872] and the Raritan online support (`https://www.raritan.com/support/product/px3`) for a more detailled documentation of the Raritan PX3 series.

4 SNMP interface and protocol

The native protocol of the Raritan PDU is SNMP, like many network-controlled power distribution units and other network equipments. The SNMP protocol for this PDU model is well described in the Raritan PDU User Guide ([Raritan PX3 User Guide] and [DESI-4872]), in its chapter 7: *Using SNMP*.

In SNMP jargon, the authentication process involves what is called *community* (login/password equivalent). On the DESI PDU two 'community' have been set: *public* giving read access, and *private* providing read/write access and which should be used when turning on/off the outlets or changing the settings.

The various PDU internal parameters are organized in a hierarchical way, like a tree, and they could be accessed through their address that tree; these addresses are called "OID" in the SNMP jargon, and could be provided in two forms, either numeric, like this:

```
.1.3.6.1.4.1.13742.6.4.1.2.1.2.1.1
```

Or in a human readable form, like this:

```
.iso.org.dod.internet.private.enterprises.raritan.pdu2.\
    control.outletControl.outletSwitchControlTable.\
    outletSwitchControlEntry.switchingOperation.1.1
```

For instance, the two strings above point to the same OID, which is an integer value which commands the outlet #1 state of the PDU (see below in 5). The equivalence between numeric and text OIDs is provided by a product specific file called a "MIB" for "Management Information Base". For this PDU, the MIB file is `PDU2-MIB-3.3.10-43736.txt` (See [DESI-4872] to get it).

There are several tools and libraries to manipulate the SNMP tree of OIDs. In the examples of this document, the SNMP commands are sent to the PDU using the SNMP toolbox called `snmptools`, which is a suite of command line programs to do atomic manipulations of the PDU internal parameters. A short description of these tools is provided in C.

In this document, for the various OID below, we follow the following conventions:

- The specific prefix for this Raritan product series:

```
.iso.org.dod.internet.private.enterprises.raritan.pdu2
.1.3.6.1.4.1.13742.6
```

is often omitted or sometimes replaced by:

```
PDU2-MIB::
```

- When the OIDs are provided in human readable form to the various SNMP tools (`snmpget`, `snmpset`, and so on), you should provide the PDU MIB file to allow the translation into numeric OIDs. For instance,

```
snmpwalk -v2c -c public 140.252.50.192:161 \
  -m PDU2-MIB-3.3.10-43736.txt PDU2-MIB::outletSensorDecimalDigits.1.5
```

  otherwise the translation to numeric OIDs will fail.

- In the following descriptions, the `<pdu>` value is always `1` (different value if chaining several PDU together through USB cables). In the DESI calibration system, the PDU selection is done through its IP address (See section A).

- `<outlet>` is the outlet number `[1-8]`.

5 Turning individual lamps on and off

To turn on or off a given lamp in one DESI calibration box, the corresponding OID is:

```
.iso.org.dod.internet.private.enterprises.raritan.pdu2\
    .control.outletControl.outletSwitchControlTable\
    .outletSwitchControlEntry.switchingOperation.<pdu>.<outlet>

.1.3.6.1.4.1.13742.6.4.1.2.1.2.<pdu>.<outlet>
```

Which may be shortened to:

```
PDU2-MIB::control.outletControl.outletSwitchControlTable\
  .outletSwitchControlEntry.switchingOperation.<pdu>.<outlet>
```

Or even:

```
PDU2-MIB::switchingOperation.<pdu>.<outlet>
```

This object is an integer (`i`): the 0 value means off and value 1 means on.

For instance, to turn on the outlet #5 on the DESI calibration box `DESI-CALIB-02` whose IP address is `140.252.50.192`, we can do (<pdu> is 1 for the DESI boxes):

```
snmpset -v2c -Ovq -c private 140.252.50.192:161 \
    PDU2-MIB::switchingOperation.1.5 i 1
```

or

```
snmpset -v2c -Ovq -c private 140.252.50.192:161 \
    1.3.6.1.4.1.13742.6.4.1.2.1.2.1.5 i 1
```

And, to turn it off,

```
snmpset -v2c -Ovq -c private 140.252.50.192:161 \
    PDU2-MIB::switchingOperation.1.5 i 0
```

or

```
snmpset -v2c -Ovq -c private 140.252.50.192:161 \
    1.3.6.1.4.1.13742.6.4.1.2.1.2.1.5 i 0
```

To know which lamps are turned on in a given DESI calibration box, it suffices to read the state of its corresponding power outlets. The MIB object PDU2-MIB::switchingOperation.<pdu>.<outlet> previously can be read as well:

```
snmpget -v2c -Ovq -c public 140.252.50.192:161 \
    1.3.6.1.4.1.13742.6.4.1.2.1.2.1.5
```

and the command will return (for instance):

```
1
```

which means the outlet 5 is on (1).

A more detailed status of a given outlet is provided by the following OIDs:

```
.iso.org.dod.internet.private.enterprises.raritan.pdu2\
    .control.outletControl.outletSwitchControlTable\
    .outletSwitchControlEntry.switchingState.<pdu>.<outlet>

PDU2-MIB::control.outletControl.outletSwitchControlTable\
    .outletSwitchControlEntry.switchingState.<pdu>.<outlet>

PDU2-MIB::switchingState.<pdu>.<outlet>

.1.3.6.1.4.1.13742.6.4.1.2.1.2.<pdu>.<outlet>
```

which returns an integer value, with the following meaning:

```
unavailable = -1, open = 0, closed = 1, belowLowerCritical = 2,
belowLowerWarning = 3, normal = 4, aboveUpperWarning = 5,
aboveUpperCritical = 6, on = 7, off = 8, detected = 9,
notDetected = 10, alarmed = 11, ok = 12, fail = 14, yes = 15, no = 16,
standby = 17, one = 18, two = 19, inSync = 20, outOfSync = 21,
i1OpenFault = 22, i1ShortFault = 23, i2OpenFault = 24, i2ShortFault = 25,
fault = 26, warning = 27, critical = 28, selfTest = 29, nonRedundant = 30
```

As an example of using the SNMP interface to control the PDU outlets, we provide an demo python script, desi-calib-ctrl, based on the SNMP tools, which can read the state of all the outlets of one or several DESI calibration boxes, or be used to turn on/off any outlet. It also translates the calibration box names (DESI-CALIB-XX) into their IP address, and the lamp names ('HgAr','Xe' and so on) into outlet numbers (See section D).

5.1 Outlet behavior at PDU start-up

We configured the Raritan outlets to always be off when the PDU is turned on. This behavior is safer (the default is to restore the previous state; if for any reason the PDU is set back to factory defaults, it means that all outlets will be turned on at startup, which may present some danger for the operator eyes with the spectral lamps. It is why we **strongly** advise to never reset the Raritan PDUs to their factory defaults).

6 Monitoring voltage, current, power and cumulated energy

6.1 Monitoring voltage, current and power per outlet

For each of its eight (8) outlets (*outlets*), the Raritan PX3-5145R PDU monitors the voltage, the current, the frequency, the power usage, in the following table:

```
PDU2-MIB::measurementsOutletSensorValue.<pdu>.<outlet>
1.3.6.1.4.1.13742.6.5.4.3.<pdu>.<outlet>
```

The table may be read with the `snmpwalk` command, like this (for outlet #5 for instance):

```
snmpwalk -v2c -c public 140.252.50.192:161 \
    PDU2-MIB::measurementsOutletSensorValue.1.5
```

Which will returns the following values:

```
PDU2-MIB::measurementsOutletSensorValue.1.5.rmsCurrent = Gauge32: 158
PDU2-MIB::measurementsOutletSensorValue.1.5.rmsVoltage = Gauge32: 120
PDU2-MIB::measurementsOutletSensorValue.1.5.activePower = Gauge32: 4
PDU2-MIB::measurementsOutletSensorValue.1.5.apparentPower = Gauge32: 19
PDU2-MIB::measurementsOutletSensorValue.1.5.powerFactor = Gauge32: 19
PDU2-MIB::measurementsOutletSensorValue.1.5.activeEnergy = Gauge32: 2
PDU2-MIB::measurementsOutletSensorValue.1.5.onOff = Gauge32: 0
PDU2-MIB::measurementsOutletSensorValue.1.5.frequency = Gauge32: 600
```

As the PDU internal numeric variables could only be integers, and not float values, the number of significant decimal digits is provided in another table, as well as the corresponding unit. For the outlet sensors, the following table provides the corresponding values:

Variable	Meaning	Digits	Unit	Effective Unit
`rmsCurrent`	Outlet current	3	Amps [A]	mA
`rmsVoltage`	Outlet voltage	0	Volts [V]	V
`activePower`	Outlet power	0	Watts [W]	W
`apparentPower`	Outlet apparent power	0	Volt-Amps [VA]	VA
`powerFactor`	Ratio between power and apparent power	2	—	0.01
`activeEnergy`	Accumulated power consumption	0	Watt-hours [Wh]	W
`onOff`	On/Off state	–	–	–
`frequency`	Line frequency	1	Hertz [Hz]	0.1 Hz

This means that for the `rmsCurrent` value, as there are 3 decimal digits, the value we get is in fact expressed in 10^{-3} A, so we should read `rmsCurrent = 158` above as $158 \times 10^{-3}\,\mathrm{A} = 158\,\mathrm{mA}$. In the same way, `frequency = 600` means $600 \times 10^{-1}\,\mathrm{Hz} = 60.0\,\mathrm{Hz}$.

The properties of each outlet sensor and the informations needed to properly interpret the sensor values are stored in the following table:

```
PDU2-MIB::outletSensorConfigurationTable
```

which may be read through this command:

```
snmptable -v2c -c public 140.252.50.192:161 \
    PDU2-MIB::outletSensorConfigurationTable
```

The individual properties may also be directly accessed by the following commands (for outlet #5 in this example):

```
snmpwalk -v2c -c public 140.252.50.192:161 \
    PDU2-MIB::outletSensorDecimalDigits.1.5
```

```
PDU2-MIB::outletSensorDecimalDigits.1.5.rmsCurrent = Gauge32: 3
PDU2-MIB::outletSensorDecimalDigits.1.5.rmsVoltage = Gauge32: 0
PDU2-MIB::outletSensorDecimalDigits.1.5.activePower = Gauge32: 0
PDU2-MIB::outletSensorDecimalDigits.1.5.apparentPower = Gauge32: 0
PDU2-MIB::outletSensorDecimalDigits.1.5.powerFactor = Gauge32: 2
PDU2-MIB::outletSensorDecimalDigits.1.5.activeEnergy = Gauge32: 0
PDU2-MIB::outletSensorDecimalDigits.1.5.onOff = Gauge32: 0
PDU2-MIB::outletSensorDecimalDigits.1.5.frequency = Gauge32: 1
```

The outlet sensor value unit could be obtained the same way:

```
snmpwalk -v2c -m mibs/PDU2-MIB-3.3.10-43736.txt \
    -c public 140.252.50.192:161 PDU2-MIB::outletSensorUnits.1.5
```

```
PDU2-MIB::outletSensorUnits.1.5.rmsCurrent = INTEGER: amp(2)
PDU2-MIB::outletSensorUnits.1.5.rmsVoltage = INTEGER: volt(1)
PDU2-MIB::outletSensorUnits.1.5.activePower = INTEGER: watt(3)
PDU2-MIB::outletSensorUnits.1.5.apparentPower = INTEGER: voltamp(4)
PDU2-MIB::outletSensorUnits.1.5.powerFactor = INTEGER: none(-1)
PDU2-MIB::outletSensorUnits.1.5.activeEnergy = INTEGER: wattHour(5)
PDU2-MIB::outletSensorUnits.1.5.onOff = INTEGER: none(-1)
PDU2-MIB::outletSensorUnits.1.5.frequency = INTEGER: hertz(8)
```

Monitoring the current and power for each outlet is important, at least when the outlet is on. For instance, an anomalous value for the current and/or power would indicate that a given lamp is dead. Change in the electric current and/or power would also give hints of the aging of the lamp (see below).

6.2 Accumulated electric power consumption

Amongst the different sensor data available for each outlet, the Raritan PX3-5145R PDU provides a counter named **`activeEnergy`** (already listed above), which monitors the accumulated power consumption for each of its eight outlets. We propose to regularly record these data, in order to use them to evaluated the aging of each individual calibration lamp. By knowing the accumulated energy for a lamp (`activeEnergy` in Watt-hours) and its electric power (in Watts) when it is turned on, we would be able to deduce the age of that lamp, and anticipate its replacement.

The corresponding OID is:

```
PDU2-MIB::measurementsOutletSensorValue.<pdu>.<outlet>.activeEnergy
.1.3.6.1.4.1.13742.6.5.4.3.1.4.<pdu>.<outlet>.8
```

For instance, we can get the `activeEnergy` for outlet #5,

```
snmpget -v2c -c public 140.252.50.192:161 \
    PDU2-MIB::measurementsOutletSensorValue.1.2.activeEnergy
```

```
PDU2-MIB::measurementsOutletSensorValue.1.2.activeEnergy = Gauge32: 39
```

6.3 Reset *Active Energy* for a outlet

When a lamp is replaced by a new one (either because the bulb dies or for whatever reason), it is important to reset the `activeEnergy` counter for the corresponding outlet. The way to do it in by setting the following SNMP variable to zero (only the zero value is allowed):

```
PDU2-MIB::outletSensorResetValue.<pdu>.<outlet>.8
.1.3.6.1.4.1.13742.6.4.7.1.1.1.<pdu>.<outlet>.8
```

For instance, to reset the active energy counter for PDU DESI-CALIB-02 (IP 140.252.50.192), outlet #3 (because we replaced the corresponding lamp), we could do:

```
snmpset -v2c -Ovq -c private 140.252.50.192:161 \
    PDU2-MIB::outletSensorResetValue.1.3.8 i 0
```

or

```
snmpset -v2c -Ovq -c private 140.252.50.192:161 \
    .1.3.6.1.4.1.13742.6.4.7.1.1.1.1.3.8 i 0
```

7 Temperature and Humidity sensors

We equipped the Raritan PDU of each calibration box with a DPX-T3H1 triple sensor which combines three (3) temperature and one (1) humidity sensor, in order to monitor the environment conditions and be able to automatically shutdown lamps if the temperature or the humidity is outside the allowed range (see 8.1). In the PDU terminology, these sensors are named "external sensors"; each individual external sensor has a unique serial number to identify it, and a unique configuration (limits) which is stored in the PDU permanent memory. It is why DPX-T3H1 sensors should not be exchanged between PDU units.

Two SNMP tables are useful to manipulate and properly interpret the data from the external sensors. The first one contains the last measurement data:

```
PDU2-MIB::externalSensorMeasurementsTable
```

They could be read like this:

```
snmptable -v2c -Cf "|" -c private 140.252.50.192 \
    PDU2-MIB::externalSensorMeasurementsTable
```

```
 IsAvailable | State              | Value | TimeStamp
 true        | normal             | 247   | 1567548266
 true        | normal             | 251   | 1567548265
 true        | aboveUpperCritical | 100   | 1567548250
 true        | normal             | 248   | 1567548266
```

Where the successive columns tell if the sensor is connected and available (`measurementsExternalSensorIs`
its state (`measurementsExternalSensorState`), its value (`measurementsExternalSensorValue`, an integer), and the timestamp of the measurement (`measurementsExternalSensorTimeStamp`). Here the temperatures are given in unit of 0.1 Celsius, and the relative humidity in percents.

As for the outlet sensors, to properly interpret the values, it is necessary to get the configuration informations for each sensor. These are stored in this huge table:

```
PDU2-MIB::externalSensorConfigurationTable
```

If we select only the relevant columns, we get the sensor type, name, serial number, and also the decimal digits and the unit of the provided value:

```
snmptable -Cf "|" -v2c -c private 140.252.50.192 \
    PDU2-MIB::externalSensorConfigurationTable | cut -d'|' -f 2,3,11,12
```

```
SerialNumber | Name                | Units   | DecimalDigits
AEH7401748   | Temperature 1       | degreeC | 1
AEI7400538   | Temperature 2       | degreeC | 1
AEI7400538   | Relative Humidity 1 | percent | 0
AEH7401747   | Temperature 3       | degreeC | 1
```

which, as previously said, indicates the corresponding decimal digits and units to properly interpret each sensor value.

It would be very useful to store these values regularly, as they provide extra environmental data from the top ring of the telescope.

8 Alert PDU behavior and Alarms

8.1 Permitted environmental conditions

During the validation tests of the calibration boxes and their PDU, we performed thermal and humidity tests using the 540 liters temperature test chamber (model 540T30 from Climats[1]) available at LPNHE (See [DESI-3639]).

We conclude from these tests that the calibration boxes could not only survive, but are fully functional between -20°C and +40°C, and between 0% and 90% of relative humidity.

We set the internal limits for the temperature and humidity sensors of the Raritan PDUs accordingly:

Limit	**Value**
Temperature upper limit (stop)	+40°C
Temperature upper limit (warning)	+35°C
Temperature lower limit (warning)	-15°C
Temperature lower limit (stop)	-20°C
Humidity upper limit (stop)	90%
Humidity upper limit (warning)	85%
Humidity lower limit (warning)	None
Humidity lower limit (stop)	None

These parameters are stored in the following table, already mentioned above:

```
PDU2-MIB::externalSensorConfigurationTable
```

[1]See www.climats-tec.com/produits/1042-nos-gammes/1043-enceinte-thermique-pour-essai-chaud-froid/1-gamme-excal.html

8.2 PDU behavior when outside of allowed ranges

This section is not complete. Some extra tests are required at Mayall before putting the PDU in operation with their sensors.

When the temperature or humidity measurements are outside the permitted ranges, this trigger an *Event* in the corresponding PDU, and the corresponding actions are executed:

1. All the outlets of this PDU are immediatly turned off;
2. The PDU is switched to a specific mode (*Load Shedding Mode* normally used to manage power overloads) where all the non-critical outlets (which mean all of them here) are disabled, so no subsequent tentative to turn them on will succeed (veto);
3. An message (alarm) is sent to the DESI ICS to warn the system that the lamps could no more be turned on. *This has to be properly defined with Klaus H.*

Once the temperature or humidity is back to normal (*deasserted alarm*), another *Event* is triggered, which:

1. Ensure all the outlets of this PDU remain turned off;
2. Take the PDU out of the (*Load Shedding Mode*) so that all outlets could now be turned on if it is desired.
3. A message (warning) is sent to warn the DESI ICS that the lamps can now be safely turned on. *This has to be properly defined with Klaus H.*

To know if the Raritan PDU has been sent to *Load Shedding Mode*, we may read the following OID:

```
PDU2-MIB::loadShedding.<pdu>
.1.3.6.1.4.1.13742.6.3.2.2.1.55.<pdu>
```

It will be `false` in normal operation, and `true` if the PDU has been put into *Load Shedding Mode*, which means that no lamp could then be turned on, for safety. It is important for the ICS (and the observer) to know that due to humidity/temperature conditions, the calibration boxes will refuse to turn on their lamps.

9 Web interface

The PX3-5145R Raritan PDU has a very nice Web interface, accessible at the standard HTTPS (443) ethernet port. For the four (4) calibration boxes (`DESI-CALIB-00` to `DESI-CALIB-03`) and also for the spare one, the login/password are:

```
  login:    admin
  password: DESIpdu
```

The web interface allows to turn on/off the outlets, read the outlets and environment sensors, and do the complete PDU setup. It is very well described in the Raritan PDU User Guide ([DESI-4872], [Raritan PX3 User Guide]), in its chapter 6: *Using the Web Interface*.

To avoid any confusion, we set up the PDU names ("DESI-CALIB-XX") in the web interface, as well as we configured the outlet names to correspond to the lamp names ("HgAr", "Kr", and so on, see B).

10 Command line interface (CLI) through SSH or RS232

The Raritan PDU offers a command line interface, which may be accessed either through SSH, or by using the serial (RS232 over USB) console. The CLI interface is described in great details in the Raritan PDU User Guide ([Raritan PX3 User Guide] and [DESI-4872]), in its chapter 8: *Using the Command Line Interface*. In this section, we will only describe the basic commands for typical operation.

You may either connect via SSH through the ethernet connection to the PDU, or connect a USB cable to the "CONSOLE" socket and use the serial (RS232) emulation with the following parameters:

```
Baudrate (bits per second) = 115200 (115.2Kbps)
Data bits = 8
Stop bits = 1
Parity = None
Flow control = None
```

In both ways, to enter the CLI mode the login/password are:

```
login:    admin
password: DESIpdu
```

10.1 Reset `admin` password

There is a way to reset the password of the `admin` user, which is only available through the console mode, for obvious reasons.

Once the RS232 (over USB) connection is established, you could reset the `admin` password, by simply typing:

```
unblock admin
```

at the login prompt. The PDU will ask for a new password thereafter.

10.2 Turning on/off outlets

The PDU command line interface allows to turn on a given outlet (here outlet #2) by simply typing:

```
power outlets 2 on
```

Or, to turn the same outlet off:

```
power outlets 2 off
```

The state of a given outlet can be displayed with the `show outlets` command. For instance, you may read the status of outlet #2 by typing:

```
show outlets 2
```

```
Outlet 2: 2 ('Continuum-B')
Power state: Off
```

Or simply entering:

```
show outlets
```

will display the power state of all the outlets.

10.3 Display the outlet sensor values

The sensors associated with each individual outlet may be read using the `show sensor outlet` command. For instance, to read the current of outlet 1:

```
show sensor outlet 1 current
```

```
Current sensor of outlet 1:
Reading: 0.000 A
State:   normal
Lower critical threshold: Disabled (0.000 A)
Lower warning threshold:  Disabled (0.000 A)
Upper warning threshold:  7.800 A
Upper critical threshold: 9.600 A
Deassertion hysteresis:   1.000 A
Assertion timeout:        0 samples
```

Or to read the accumulated energy consumption of outlet 1:

```
show sensor outlet 2 activeEnergy
```

```
Active energy sensor of outlet 2:
Reading: 39 Wh
State:   normal
Lower critical threshold: Disabled (0 Wh)
Lower warning threshold:  Disabled (0 Wh)
Upper warning threshold:  Disabled (0 Wh)
Upper critical threshold: Disabled (0 Wh)
Deassertion hysteresis:   0 Wh
Assertion timeout:        0 samples
```

10.4 Display the external sensor values

The command line interface allows to read the external sensors values, with the `show externalsensors` command. For instance, to read the first sensor value, the command is:

```
show externalsensors 1
```

```
External sensor 1 ('Temperature 1')
Sensor type: Temperature
Reading:     24.0 deg C (normal)
```

Or all of them at once:

```
show externalsensors
```

```
External sensor 1 ('Temperature 1')
Sensor type: Temperature
Reading:     24.0 deg C (normal)

External sensor 2 ('Temperature 2')
Sensor type: Temperature
Reading:     24.5 deg C (normal)
```

```
External sensor 3 ('Relative Humidity 1')
Sensor type: Relative Humidity
Reading:      55 % (normal)

External sensor 4 ('Temperature 3')
Sensor type: Temperature
Reading:      23.4 deg C (normal)
```

A Network configuration of the individual PDU

Here are the permanent IP addresses for the calibration box PDUs:

Label (on the box)	Hostname	IP address
`DESI-CALIB-00`	`desi-calib-pdu-0`	140.252.50.190
`DESI-CALIB-01`	`desi-calib-pdu-1`	140.252.50.191
`DESI-CALIB-02`	`desi-calib-pdu-2`	140.252.50.192
`DESI-CALIB-03`	`desi-calib-pdu-3`	140.252.50.193
`DESI-CALIB-SPARE`	`desi-calib-pdu-spare`	140.252.50.194

And here are the device network settings:

```
Network:
    Common Network Settings:
        DNS Resolver Preference: IPv4
        DNS Suffixes: kpno.noao.edu, noao.edu
        First DNS Server:  140.252.62.83
        Second DNS Server: 140.252.51.10
        Third DNS Server:  140.252.1.54
        Default Gateway: 140.252.50.1
    ETH1:
        IP Auto Configuration: Static
        IP Adress/Prefix Length: 140.252.50.XXX/24

Date/Time:
    Time Zone: Arizona
    DST Off
    NTP Servers:
        140.252.51.140
        140.252.1.140
```

Of the two ethernet interfaces, only the `ETH1` netowrk interface should be used.

B Lamps slots and outlets

The following table gives the position (slot) and the outlet corresponding to each lamp in all calibration boxes. When looking at the front face of the calibration box, with the PDU on top, the slots are numbered from 1 (leftmost) to 6 (rightmost).

Lamp Drawer	Rack Slot	PDU Outlet
`HgAr`	1 (leftmost)	8 / "HgAr"
`Kr`	2	7 / "Kr"
`Ne`	3	6 / "Ne"
`Xe`	4	5 / "Xe"
`Halogen No Filter`	5	3 / "C", "NO FILT"
`Halogen Blue Filter`	5	2 / "B", "BLUE"
`Cd`	6	4 / "Cd"

C SNMP Tools

The SNMP command line tools used in the examples in this document are the standard ones provided by NET-SNMP. They are available there: `http://www.net-snmp.org/`, and a detailled documentation is also available online. In this section we describe briefly several of these tools.

C.1 MIB file and path

To work properly, the SNMP tools should have access to the MIB file of the device we want to talk with, except if the OID are provided in numeric only. Otherwise, for human readable OID, either the MIB file should be explicitely specified with the `-m` option, like this:

```
snmpwalk -v2c -c public 140.252.50.192:161 \
    -m PDU2-MIB-3.3.10-43736.txt PDU2-MIB::outletSensorDecimalDigits.1.5
```

Or a search path should be provided with the `-M` option:

```
snmpwalk -v2c -c public 140.252.50.192:161 \
    -M /usr/share/mibs:/home/llg/.snmp/mibs:/usr/share/snmp/mibs:\
       /usr/share/snmp/mibs/iana:/usr/share/snmp/mibs/ietf:\
       /usr/share/mibs/site:/usr/share/snmp/mibs:/usr/share/mibs/iana:\
       /usr/share/mibs/ietf:/usr/share/mibs/netsnmp
    PDU2-MIB::outletSensorDecimalDigits.1.5
```

C.2 `snmpget/snmpset`

The `snmpget` and `snmpset` commands allow to read or write an atomic OID. `snmpget` is used to read one parameter at a time:

```
snmpget -v2c -c public 140.252.50.192:161 \
    PDU2-MIB::measurementsOutletSensorValue.1.2.activeEnergy
```

```
PDU2-MIB::measurementsOutletSensorValue.1.2.activeEnergy = Gauge32: 39
```

And `snmpset` is used to write a parameter value. The type has to be specified ("i" for integer, "s" for character string, and so on). For instance, to write an integer (`i`) value of 1 (which here will turn on the outlet #5):

```
snmpset -v2c -Ovq -c private 140.252.50.192:161 \
    PDU2-MIB::switchingOperation.1.5 i 1
```

C.3 snmpwalk/snmptable

The `snmpwalk` command will recursively walk a subtree of the parameter tree, and display all the parameter values of the OIDs in that subtree. For instance,

```
snmpwalk -v2c -c public 140.252.50.192:161 .1.3.6.1.4.1.13742
```

will display the complete tree containing all the PDU parameters (which is a huge amount of data!).

Some parameters are organised as "tables" with columns and one or several index (for instance, per outlet, per sensor, and so on). The `snmptable` will read all the values of a given table OID, and display them in columns according to the internal table scheme. For instance,

```
snmptable -v2c -c public 140.252.50.192:161 \
    PDU2-MIB::externalSensorMeasurementsTable
```

C.4 snmptranslate

The `snmptranslate` is a resolver: it will translate OID from their human readable form to their numeric form, or vice-versa, depending of the specified command line options.

For instance, to translate a string OID to its numeric form (`-On` option):

```
snmptranslate -On PDU2-MIB::control.outletControl.outletSwitchControlTable\
                     .outletSwitchControlEntry.switchingOperation.1.4
```

```
.1.3.6.1.4.1.13742.6.4.1.2.1.2.1.1
```

This also works in shortened form:

```
snmptranslate -On PDU2-MIB::switchingOperation.1.4
```

```
.1.3.6.1.4.1.13742.6.4.1.2.1.2.1.1
```

Or to do the opposite, from the numeric form to the human readable one:

```
snmptranslate -m mibs/PDU2-MIB-3.3.10-43736.txt \
   .1.3.6.1.4.1.13742.6.4.1.2.1.2.1.1
```

```
PDU2-MIB::switchingOperation.1.1
```

Of course, as for the other SNMP tools, the MIB file or path should be specified (see above) for the translation to work.

D Demo Python scripts

A demo script named `desi-calib-ctrl` written in Python and based on the SNMP tools is available on the DESI github, here:

`https://github.com/desihub/desilamps/`

It allows to read the outlet state, turn on/off any outlet, display the sensor outlet values, and the temperature and humidity sensor states and values. It also translates the calibration box names (`DESI-CALIB-XX`) into their IP address, and the lamp names (`'HgAr'`, `'Xe'` and so on) into outlet numbers. For more details, see the web page above and the documentation of the `desi-calib-ctrl` script.

References

[Jelinsky et al., 2016] Jelinsky, P., Edelstein, J., et al. (2016).
The dark energy spectroscopic instrument (DESI) spectrographs.
In *Proc. SPIE*, volume 9908.

[DESI-1067] Jelinsky, P., Guy, J., Karkar, S. (v3, 2018).
DESI Calibration System Requirements
`https://desi.lbl.gov/DocDB/cgi-bin/private/ShowDocument?docid=1067`

[DESI-1673] Guy, J., Karkar, S., Le Guillou, L., Ghislain, P. (2017).
On site calibration system for DESI spectrographs
Laboratoire de Physique Nucléaire et de Hautes Energies (LPNHE).
`https://desi.lbl.gov/DocDB/cgi-bin/private/ShowDocument?docid=1673`

[DESI-2674] Guy, J., Karkar, S., Le Guillou, L. (2017)
Characterization of spectral lamps candidates for DESI calibration system
Laboratoire de Physique Nucléaire et de Hautes Energies (LPNHE).
`https://desi.lbl.gov/DocDB/cgi-bin/private/ShowDocument?docid=2674`

[DESI-3639] Le Guillou, L., Karkar, S., Guy, J. (2018)
Calibration System: Tests Report
Laboratoire de Physique Nucléaire et de Hautes Energies (LPNHE).
`https://desi.lbl.gov/DocDB/cgi-bin/private/ShowDocument?docid=3639`

[DESI-4872] Raritan PDU Documentation.
Includes: PX3-3000/4000/5000 Series User Guide (Raritan).
`https://desi.lbl.gov/DocDB/cgi-bin/private/ShowDocument?docid=4872`

[Raritan PX3 User Guide] PX3-3000/4000/5000 Series User Guide (Raritan).
`https://www.raritan.com/support/product/px3`