The following is a list of commands and instructions to manipulate the Wiener High Voltage power supply unit (MPOD) used at LeRIBSS.

The status of MPOD can be viewed in a browser by setting the browser address to MPOD (http://MPOD) in any window opened by the dssd account.

The script commands to manipulate MPOD are located in `~dssd/HV_WIENER/`

To run the script commands you must login to dssd@grpc02.phy.ornl.gov (the machine running the experiment)

The README in the top of the directory contains much of the following information.

**BEFORE EXECUTING ANY COMMANDS ENSURE THAT THE CORRECT VOLTAGES FOR EACH DETECTOR & CHANNEL ARE APPROPRIATELY SET. THIS REQUIRES PHYSICALLY CHECKING THE CONNECTIONS ON THE ISEG MODULES IN THE CRATE FOR CHANNEL ASSIGNMENTS AND POLARITY SETTINGS.**

<table>
<thead>
<tr>
<th>Clover 1 Crystal 1 (RED)</th>
<th>MPOD ch u0</th>
<th>Clover 3 Crystal 1 (RED)</th>
<th>MPOD ch u100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clover 1 Crystal 2 (GREEN)</td>
<td>MPOD ch u1</td>
<td>Clover 3 Crystal 2 (GREEN)</td>
<td>MPOD ch u101</td>
</tr>
<tr>
<td>Clover 1 Crystal 3 (BLUE)</td>
<td>MPOD ch u2</td>
<td>Clover 3 Crystal 3 (BLUE)</td>
<td>MPOD ch u102</td>
</tr>
<tr>
<td>Clover 1 Crystal 4 (BLACK)</td>
<td>MPOD ch u3</td>
<td>Clover 3 Crystal 4 (BLACK)</td>
<td>MPOD ch u103</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Clover 2 Crystal 1 (RED)</th>
<th>MPOD ch u4</th>
<th>Clover 4 Crystal 1 (RED)</th>
<th>MPOD ch u104</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clover 2 Crystal 2 (GREEN)</td>
<td>MPOD ch u5</td>
<td>Clover 4 Crystal 2 (GREEN)</td>
<td>MPOD ch u105</td>
</tr>
<tr>
<td>Clover 2 Crystal 3 (BLUE)</td>
<td>MPOD ch u6</td>
<td>Clover 4 Crystal 3 (BLUE)</td>
<td>MPOD ch u106</td>
</tr>
<tr>
<td>Clover 2 Crystal 4 (BLACK)</td>
<td>MPOD ch u7</td>
<td>Clover 4 Crystal 4 (BLACK)</td>
<td>MPOD ch u107</td>
</tr>
</tbody>
</table>

| Beta 1 (negative) -1800V | MPOD ch u200 | Beta 2 (negative) -1800V | MPOD ch u201 |

**Executing Script Commands**

The entire clover array can be powered up / down by executing the command “./cards_up(down)”.

- “_up” Clocers 1- 4 up with appropriate voltage and current limits: RAMP speed 10V/s
- “_down” Clocers 1-4 down and switched OFF

Similarly the beta detectors can be powered up/down by “./beta_up(down)”.

- “_up” Scintillators 1& 2 up with appropriate voltage and current limits: RAMP speed 50V/s
- “_down” Scintillators 1& 2 down and switched OFF

After powering up any detectors required the monitoring current limits can be set to safe levels by executing the command “./iLimit”

In an emergency the crate can be immediately shut down by “./scram_hv”.

- All voltages set to **ZERO**, all RAMP speed **20V/s**, all channels **OFF**
Adjusting Variables

The master scripts are strings of scripts for individual detectors which are located in the /SCRIPTS directory. In order to set a particular voltage, current limit or ramp speed edit the appropriate detector script, this will then be used when the master command is executed. An example bash is shown on the next page with appropriate fields highlighted.

The monitoring current limits for the clover detectors can be set in clover_monitoring.bash The iLimit set in the clover_up/down script is ONLY valid during ramping up or down and not suitable during normal operations.

Procedure

1. Ensure the master switch on the MPOD crate is on (GREEN).
2. On MPOD web page check all channels are set to 0 and switched OFF.
3. Check the bias voltage on each clover matches the voltage set limit for each detector.
4. Connect at least one channel of each detector to an oscilloscope.
5. Execute the script commands – while ramping up occurs monitor the oscilloscope and the MPOD web output.
6. Set the current limit monitor to safe operating values.
Example bash script

#!/bin/bash clover1_up.bash
# CLOVER 1 CARDS
# Script for read and set channels in an MPOD crate
# modified from Wiener Manual Example
# IGD 18.Apr.08
# option -Oqv suppresses "WIENER-CRATE-MIB::outputIndex.u0 = INTEGER:" eg for outputIndex
# options -q "quick print for easier parsing" -v "print values only" -U "dont print units"

ip=192.168.13.31
path=/usr/share/snmp/mibs

setVoltage=3000
isetCurrent=0.001
setRamp=10

#Status 0 off 1 on DO NOT TURN ON IF ALREADY ON
setStatus=1

channelCount=$(snmpget -Oqv -v 2c -M $path -m +WIENER-CRATE-MIB -c guru $ip outputNumber.0)
indices=$(snmpwalk -Oqv -v 2c -M $path -m +WIENER-CRATE-MIB -c guru $ip outputIndex)
x=(`echo $indices | tr ''```
COUNTER=0

while [ $COUNTER -lt $channelCount ]; do
    index=$(echo ${x[$COUNTER]})
    if [ $index == u0 -o $index == u1 -o $index == u2 -o $index == u3 ]; then
        #set parameters
        voltage=$(snmpset -Oqv -U -v 2c -M $path -m +WIENER-CRATE-MIB -c guru $ip outputVoltage.$index F $setVoltage)
iLimit=$(snmpset -Oqv -U -v 2c -M $path -m +WIENER-CRATE-MIB -c guru $ip outputCurrent.$index)
rampspeed=$(snmpset -Oqv -U -v 2c -M $path -m +WIENER-CRATE-MIB -c guru $ip outputVoltageRiseRate.$index)
    fi

    #ON OFF PROTECTION TEST
    status=$(snmpget -Oqv -U -v 2c -M $path -m +WIENER-CRATE-MIB -c public $ip outputSwitch.$index)
    if [ $setStatus -eq 1 -a $status == off ]; then
        status=$(snmpset -Oqv -U -v 2c -M $path -m +WIENER-CRATE-MIB -c guru $ip outputSwitch.$index i $setStatus)
    elif [ $setStatus -eq 0 -a $status == on ]; then
        status=$(snmpset -Oqv -U -v 2c -M $path -m +WIENER-CRATE-MIB -c guru $ip outputSwitch.$index i $setStatus)
    fi

    #review settings
    voltage=$(snmpget -Oqv -U -v 2c -M $path -m +WIENER-CRATE-MIB -c public $ip outputVoltage.$index)
iLimit=$(snmpget -Oqv -U -v 2c -M $path -m +WIENER-CRATE-MIB -c public $ip outputCurrent.$index)
sense=$(snmpget -Oqv -U -v 2c -M $path -m +WIENER-CRATE-MIB -c public $ip outputMeasurementSenseVoltage.$index)
current=$(snmpget -Oqv -U -v 2c -M $path -m +WIENER-CRATE-MIB -c public $ip outputMeasurementCurrent.$index)
rampspeed=$(snmpget -Oqv -U -v 2c -M $path -m +WIENER-CRATE-MIB -c public $ip outputVoltageRiseRate.$index)
    status=$(snmpget -Oqv -U -v 2c -M $path -m +WIENER-CRATE-MIB -c public $ip outputSwitch.$index)

    echo "$index $voltage $iLimit $sense $current $rampspeed $status"

    let COUNTER=COUNTER+1
done