



11-Jun-19	1/4	

## 1. Slow Control Test: Voltage and Current check

The *test.cgi* calls the executable called *get\_i\_v* which has as output a JSON file in which all the current and voltage values are reported (the data that come out with the command are reported slowc -a). In this example the executable does not take any arguments as input.

The JSON response consists of:

Item of JSON file	MEANING
<b>V_EXT1_24V</b>	Value of the voltage of EXT1
<b>I_V_INPUTS</b>	Value of the current of EXT1
<b>SP_VOLT</b>	Value of the SP voltage
<b>SP_CURR</b>	Value of the SP current
<b>V_1V0</b>	Value of the 1V voltage
<b>I_1V0</b>	Value of the 1V current
<b>V_1V2</b>	Value of the 1V2 voltage
<b>I_1V2</b>	Value of the 1V2 current
<b>V_1V8</b>	Value of the 1V8 voltage
<b>I_1V8</b>	Value of the 1V8 current

<b>V_3V3</b>	Value of the 3V3 voltage
<b>I_3V3_SC</b>	Value of the 3V3 current
<b>V_GPS_5V</b>	Value of the GPS voltage
<b>I_GPS_5V</b>	Value of the GPS current
<b>V_RADIO_12V</b>	Value of the Radio 12V voltage
<b>I_RADIO_12V</b>	Value of the Radio 12V current
<b>V_PMTS_12V</b>	Value of the PMTs 12V voltage
<b>I_PMTS_12V</b>	Value of the PMTs 12V current
<b>LOADCURR</b>	Value of the Load current
<b>PMT1_HVM</b>	Value of the PMT1 voltage
<b>PMT1_CM</b>	Value of the PMT1 current
<b>PMT2_HVM</b>	Value of the PMT2 voltage
<b>PMT2_CM</b>	Value of the PMT2 current
<b>PMT3_HVM</b>	Value of the PMT3 voltage
<b>PMT3_CM</b>	Value of the PMT3 current
<b>PMT4_HVM</b>	Value of the PMT4 voltage
<b>PMT4_CM</b>	Value of the PMT4 current

<b>PMT5_HVM</b>	Value of the PMT5 voltage
<b>PMT5_CM</b>	Value of the PMT5 current
<b>PMT6_HVM</b>	Value of the PMT6 voltage
<b>PMT6_CM</b>	Value of the PMT6 current

In the case where it is necessary to divide the values related to the PMT by the fixed values of the low-voltage, two different executables can be created.

## 2. Slow Control Test: HV\_Test

The *test.cgi* calls the executable *HV\_Test*, this routine sets all DACs to zero and requests the read-back voltages for each PMTs output. Then it sets the DACs to maximum value and requests the voltages again. The difference of two values is returned in JSON format for each PMTs:

<b>Item of JSON file</b>	<b>MEANING</b>
<b>PMT1_HVM_diff</b>	Difference related to PMT1
<b>PMT2_HVM_diff</b>	Difference related to PMT2
<b>PMT3_HVM_diff</b>	Difference related to PMT3
<b>PMT4_HVM_diff</b>	Difference related to PMT4
<b>PMT5_HVM_diff</b>	Difference related to PMT5
<b>PMT6_HVM_diff</b>	Difference related to PMT6

### 3. Slow Control Test: fpga version

The *test.cgi* calls the executable *get\_fpga\_version* which reads the FPGA version from the register ID\_REG\_ADDR and outputs a JSON file with the following item:

Item of JSON file	MEANING
<b>VERSION</b>	Date of last compilation as reported in the FPGA records.

### 4. FPGA interface Test: test of GPS

The *test.cgi* calls the executable GPS\_serial that checks the GPS serial port. It has as output a JSON file which has only one item that changes depending on the result of the check:

Item of JSON file	MEANING
<b>message</b>	Ok Ok
<b>message</b>	failed message

## 5. Usb interface Test: `umount_usb`

The *test.cgi* calls the executable *umount\_usb*: which has no input parameters and replies with a JSON file that can have three kinds of message:

Item of JSON file	MEANING
message	/usb already unmounted (if the usb is already unmounted)
message	/usb unmounted (if the usb is correctly unmounted)

## 6. Usb interface Test: `mount_usb`

The *test.cgi* calls the executable *mount\_usb*: which has no input parameters and replies with a JSON file that can have three kinds of message:

Item of JSON file	MEANING
message	/usb already mounted (if the usb is already mounted)
message	OK (if the usb is correctly mounted)
message	ERROR (if an error occurred in mounting usb)

## 7. Front-end Test: signal test.

The *test.cgi* calls the executable record that which has as input the values of the thresholds on the PMTs and responds with a JSON file that shows the following values for each ADC:

Item of JSON file	MEANING
<b>baseline</b>	Baseline average value for each adc.
<b>RMS</b>	Baseline rms value for each adc.
<b>Pulsepos</b>	Value of pulse position for each adc.
<b>valuepeak</b>	Value of peak for each adc.