## VME PC config, trouble, installations…

1. Get the USB and GPIB mode r+w all the time after reboot :

Add file inside /etc/udev/rules.d/ 60-laser-usb.rules and 60-laser-gpib.rules Ask Savahna to have this into Profile (keep config after any re installation of SLC6)

1. #ldd laser give the list of library used by laser

Libforms.so.1 - libg2c.so.o not found To install libforms , thus “Xforms” ->

cd ~/software/xforms-1.0-release make install ; with sudo and add one file in "/etc/ld.conf.d/gpib.conf", then run ldconfig.

1. for laser need also cern.tar.gz

Need to compile the source code again. In home directory need .tcshrc.

**In order to compile** (and be able to do ‘make’ for dp2) :

# tcsh

#### With Laser account !

#cd dim

#source .setup

# cd dp2 make

**FAST MONITORING:**

1. Define DIM\_DNS\_NODE (if there is no .dat file !)

Need tschrc shell… but not ok for CMS machine

*Hi,*

*The point is we don't change the default shell for root because other users might want to log on the machine and they may be not familiar with tcsh (we, sysadmins, are not familiar with TCSH).*

*We may change YOUR user default shell in LDAP but that will apply on ALL the machines (not only the laser machines).*

*It will be easier for you to launch a TCSH shell from your ~/.bashrc.*

*And what do you mean by 'have defined DIM\_DNS\_NODE'? Environment variables can also be defined in bash.*

##### So add ‘tcsh’ in file ‘.bashrc”

**DNS, DIM SERVER**

DNS can work on the same machine as slow monitor BUT it was on separate machine since the beginning and can’t remember why !

Modify file ON PC where slow monitor is running (PC02 or PC05):

**.tcshrc** setenv DIM\_DNS\_NODE ecal-laser-room-01 If DNS is running on PC01.

IF DNS running on same PC, change it !

6 Oct 2014: Start DNS on ecal-laser-room-02 as well to get histo& .dat !

# laser/dim/linux# .dns

##### To see the DIM server/client : Did

To enable Did, juste type : source .setup before Did

Ecal-laser-room-05 /home/laser/dim/linux $ ./did nLasers, nDso…

**Acqiris driver installation:**

Do not install the one from the CD: MD1.1-10.5 because gcc compiler not the same… can’t work ( command : # gcc –version)

##### Use Acqiris-4.2a-gcc-4.3….tar.gz.

Acqiris/…/kernel26module : make clean all, make install

To enable or disable the kernel module, you can run the commands:

/etc/init.d/acqiris start

/etc/init.d/acqiris stop

First update the kernel modules dependencies:

/sbin/depmod -a

Then load the kernel module:

/sbin/modprobe -v acqiris

[http://www.keysight.com/upload/cmc\_upload/All/README?&cc=CH&lc=gerOnly](http://www.keysight.com/upload/cmc_upload/All/README?&amp;cc=CH&amp;lc=gerOnly)

1. View PCI board:

#modeprobe –list -> list of all modules files (tnt4882 should be there)

Or same with :

#/sbin/lsmod | grep tnt

List of devices: #/sbin/lspci | grep PCI

##### For Acqiris we have: Device: 43…

**Class: Unassigned Class ff00**

**Vendor: Agilent technologies**

**Device: device 0001**

!!!! ~/monitor/src should not be copy under root privilege ! else not working… !!!!!

Compile the GPIB driver:

Yes, you need to do “configure-make-make install”, and then edit the “/etc/gpib.conf”.

If not working : I think the kernel source you installed doesn’t match the running kernel. You have to install the same version.

Ask Savanah..

# Compile dp2 slow monitoring:

To compile, please do following commands first

### With Laser account

[ECAL-LASER-ROOM-02 dim]$ tcsh

cd dim source .setup

You may run “cd ~/dim; source .setup” first and then go to dp2 directory to run make.

 Do all this under the same terminal ! (change folder on the same terminal .Another terminal to run make failed ! )

**New:**

**Sept 2020 with new program (root based) :** # gmake

Not "make", because the compilation is starting on remote computer (ecalod-daq?) in this case, but libraries are absent in that computer.

**To start DNS automatically at the PC booting:**

[laser@ecal-laser-room-05 dp2n]$ crontab –l

Give @reboot /home/laser/dim/linux/dns

To modify the text as it is : #crontab –e

<https://www.simplified.guide/linux/automatically-run-program-on-startup>

1. **VNC:**

Install X11vnc and not vncserver:

# sudo yum install X11vnc

The just run X11 vnc as ecal-laser-room-01:

# x11vnc –display :0 –passwd xxxx – forever – gone ‘xlock&’

To see the list of process running :

# ps -ef | grep x11vnc Should be only one else Kill xxxxx

#  IPMI :

### <https://twiki.cern.ch/twiki/bin/viewauth/CMS/HowToRebootPcWithIPMI>

#ipmitool -H $HOSTIPMI -U root -f ~ecaldev/utils/ipmi chassis power off COMMAND="sudo ipmitool -I lan -U root -f /etc/ipmi.conf -H $HOSTIPMI $IPMI\_OPTIONS"

Normal command:

#sudo ipmitool -I lan -H ecal-laser-room-05.ipmi.cms -U root chassis power status Better with “lanplus “ than lan for newest machine.

(Or my account

#ipmitool -I lan -H ecal-laser-room-05.ipmi.cms -U Bailleux chassis power status )

check the scripts that are in ~ecaldev/utils/systools They have the correct options to operate the ipmi.

Edit text on RACK PC:

Gedit not working with sudo used

# **sudo nano** xxxx.txt ,ect…

##### COPY FOLDER FROM RACK PC to ANOTHER (PC02 to PC05):

1. Copy #monitor from PC05 to PC02:

Login on laser@PC02: laser@ecal-laser room02~$

scp –r bailleux@ecal-laser-room-05:/home/laser/monitor /home/laser/monitor

The folder monitor will be created automatically

1. Copy file for PC02 to PC05:

laser@ecal-laser-room-05~$ scp bailleux@ecal-laser-room-02:/home/laser/dp2n/dsoxxx.ps .

1. Copy file from bailleux@cmsusr to PC05 (or from Windows to PC05)
* On cmsusr:

[bailleux@srv-C2C03-22 ~]$ scp ser2net-2.10.0.tar.gz bailleux@ecal-laser-room-05

* On PC05:

[laser@ECAL-LASER-ROOM-05 ~/green]$ scp bailleux@ecal-laser-room-05:ser2net-2.10.0.tar.gz .

Oblige comme ca , car /home/laser est restrain..

1. **Copy laser program PC02 to bailleux@cmsusr:**

For 1 file:

[bailleux@kvm-s3562-1-ip150-16 dp2]$ scp laser@ecal-laser-room-02:/home/laser/dp2n/dso21490-2.ps bailleux@cmsusr:/nfshome0/bailleux/dp2

For ALL: scp –r ….

Ask for laser login, then login to laser@ecal-laser-room-02 : il suffit de faire #exit et les files sont copiés

# Matacq DATA/soft:

### Start/check the Matacq-Light CHECKER:

ssh ecalod-laser-xdaq cd ~ecallaser

sudo –u ecallaser –H bash –l

#ecallaser@..-09 (or 04..)$ cd ~ecallaser/bin

$ **./lmfc** status matacq\_feedback

./lmfc start matacq\_feedback

Twiki: <https://twiki.cern.ch/twiki/bin/view/CMS/EcalLaserMonitoring>

# VALUE OF THE EMTC :

Connect to the laser supervisor :  **ecalod-laser-xdaq**

Ecallaser user: sudo –u ecallaser –H bash –l

cd ~ecallaser

cd emtc/emtccons\_fw14/caen/

**./emtccons 17**

>H = help

>Status, ..

For the green laser delay:

>Read 28 4 : read the last 4 bytes address 28.

Value by default is 1ED9. To be changed to 1ECC to **adjust new Green laser** …

>Write 28 4 1ECC

>trigdel

 View de trigger delay WTE.

The registerd “28” for green timing is resetting after each new run to 1ED( (7897)

Last good value : write 28 4 0x**1EDC** (7885)

Compare to original one, 1ED9 7897, it is 120ns difference ? or..

**The main delay of the calibration is done by:**

**delay = WTE\_2\_GREEN\_LASER \* 25ns + GREEN\_LASER\_PHASE \* 3.125ns + delay Reg.28 \* 12.5ns**

The value WTE,ect are stored on the DBGui.

WTE=21 and green laser phase: 4

There is one setting called **“nimdelay”** on **~ecalpro 🡪 emacs .lasercfg** :

Changed by Marc on 10 June to : **nimdelay= 7885.**

**From email :**

the 13.4us is the time between the trigger signal at the output of EMTC and the laser pulse seen on the monitoring photodiode in the laser barrack. So it includes the ~4us delay of the laser.

We received the laser trigger (WTE signal) 16.75us before the L1A. The crystal flashing must occurs 3.2us (should be actually 3.75us but we took 3.2us...) before the L1A because of FE pipeline. We estimated time adjustment due to fiber length to ~175ns to add to the 16.75us. That gave us at the end a ~13.4us. However the EMTC card has a programmable delay with 3.125ns granularity for its laser trigger output. This delay was set to 0, but we can use it for 13.4us delay or part of it.

From Philippe email… :

*We have three different delay setting:*

*- one from the laser controls called "delay A" (in microseconds) for the photonics and nimDelay (in 12.5ns unit) for the green laser.  The first is set on the local laser control application and is managed by David.*

*The second one is configured in the DAQ as the following parameters.*

*- a delay set in the EMTC card, configured in the DAQ for each laser, called* ***WTE\_2****\_XXX\_LASER and expressed in 25ns unit.*

*- a delay set in the EMTC card, configured in the DAQ in 3.125 ns for each laser, called XXX\_****LASER\_PHASE*** *and in 3.125ns unit. It runs from 0 to 21.875ns, hence the name "phase".*

*- a delay offset set individually for each monitoring region in 3.125ns unit. If this offset is different than 0, then the WTE\_2\_XXX\_LASER and XXX\_LASER\_PHASE values are modified before being written in the EMTC card in order to include this offset in the delay.*

*New values are written each time the sequence switch to a different monitoring region.*

#### EMTC log:

The emtc log file is located in

ecal-laser-room-04:/localdata/matacq-data/emtc-log/

Time stamp: 1266996085.673421, status 00000000101111100000000000000010

**Active 2**: 69333426 69378346 234600 24000 606801

TTC: 1: 41 1 2 0 3564 2915 3414 89 193 2 0 0 0 0 3 78300

TTC: 2: 41 1 2 0 3564 2915 3414 89 193 2 0 0 0 0 3 78300

TTC: 3: 41 1 2 0 3564 2915 3414 89 193 2 0 0 0 0 3 78300

TTC: 4: 41 1 2 0 3564 2915 3414 89 193 2 0 0 0 0 3 78300

for each TTC channel (one per partition), you have:

* the last monitoring region requested (1..54)
* the monitoring side (0 or 1)
* the trigger type (0=laser, 1=LED 2=TP, 3=pedestal)
* the color (0=blue, 1=green, 2=red, 3=infra-red)
* orbit length (3564 clocks)
* BC0 to WTE (2915 clocks)
* BC0 to TE1 (3414 clocks)
* TE1 to TE2 (89 clocks)
* TE1 to LV1 (193 clocks)

…

…

Laser farm controls (for experts)

The ~ecallaser/bin/lmfc script provides control for starting, stopping and display status of the laser farm processes. Note that ~ecallaser/bin is in the PATH of user ecallaser so it can usually be omitted while executing lmfc command.

Application lmfc must be run as ecallaser user. In order to switch to the ecallaser user account, execute the command ~ecallaser/bin/laser.

monitoring monitoring analysis process

sorting sorting process

matacq\_feedback matacq feedback process

emtc\_log emtc logger process

matacq\_xfer matacq data copy to LMF process prim\_gen laser primitive generation process

lighhecker web page server of laser light checker [http://ecal-laser-room-](http://ecal-laser-room-03.cms/laser-fastcheck/) [03.cms/laser-fastcheck/](http://ecal-laser-room-03.cms/laser-fastcheck/) (  not yet supported by the lmfc command)

db\_writing process transfering LMF primitives info to OMDS database

cls\_writing process transfering LMF normalized primitives info to OMDS database

corr\_writing process transfering corrections to OMDS database

global\_check process checking last results for LMF

### Plot Matacq

ssh Vmepc-ecal-laser-01 ssh srv-ecal-laser-13

cd /nfshome0/ecallaser/matacqFastFeedback/histo\_dqm or /histo2

Copier plot root:

cp /nfshome0/ecallaser/matacqFastFeedBack/histo\_dqm/analog\_dqm\_20141125\_04\_0\_6.root .

**Linux:**

**PC log file**

 sudo ipmitool –help give list of option:

sensor, or sel for Event log file.

# **sudo ipmitool sel list**

Sensor 🡪 to view if IPMI OK. No hardware issue..

EXAMPLE PROBLEM of REBOOT PC BECAUSE OF POWER SUPPLY:

 f | 11/16/2015 | 05:56:02 | Power Supply #0x74 | Fully Redundant | Asserted

 10 | 01/18/2016 | 20:35:03 | Power Supply #0x74 | Redundancy Lost | Asserted

 11 | 01/18/2016 | 20:35:08 | Power Supply #0x74 | Fully Redundant | Asserted

 12 | 02/21/2016 | 11:46:37 | Processor #0x60 | IERR | Asserted

 13 | 02/21/2016 | 11:46:48 | Power Supply #0x74 | Redundancy Lost | Asserted

 14 | 02/21/2016 | 11:46:53 | Power Supply #0x74 | Fully Redundant | Asserted

 15 | 05/11/2016 | 20:41:28 | Power Supply #0x74 | Redundancy Lost | Asserted

 16 | 05/11/2016 | 20:41:33 | Power Supply #0x74 | Fully Redundant | Asserted

/var/log/messages

 sudo grep pl2303 /var/log/messages | grep -v dell

To view a common log file called /var/log/messages use any one of the following command:

# less /var/log/messages
# more -f /var/log/messages
# cat /var/log/messages
# tail -f /var/log/messages SEE CURRENT MESSAGES ONLINE
# grep -i error /var/log/messages SEE LAST ERROR

# grep –v dell /var/log/messages from Evgueni

# grep –I bug /var/log/messages to see PC crash reboot,….

clear cache PC (ssh log or when changing alias,ect) :

sudo nscd –i host OR sudo sss\_cache –d

***How To* Find USB** Vendor/Product id's and serial numbers in Linux

On vme-ecal-laser-room-01 for example:

# dmesg | grep tty 🡪

*…..*

*console [tty0] enabled*

*console [ttyS1] enabled*

*serial8250: ttyS0 at I/O 0x3f8 (irq = 4) is a 16550A*

*serial8250: ttyS1 at I/O 0x2f8 (irq = 3) is a 16550A*

*00:05: ttyS1 at I/O 0x2f8 (irq = 3) is a 16550A*

*00:06: ttyS0 at I/O 0x3f8 (irq = 4) is a 16550A*

***usb 2-1.2****: pl2303 converter now attached to* ***ttyUSB0***

***usb 2-1.8****: pl2303 converter now attached to* ***ttyUSB1***

YUM LIST ….

YUM INSTALL…

For package

# lsusb

**From Pedro to get the USB all the time recognize after each reboot :**

*just copy the file /etc/udev/rules.d/60-laser-usb.rules from VMEPC-ECAL-LASER-01, and place it in the same place on you other machines*

*Note this fix is only if the dev is installed in /dev/ttyUSB\**

*The content of the file is a single line:*

*KERNEL=="ttyUSB\*", GROUP="uucp", MODE="0666"*

*(Maybe easier than copy the file.)*

**Verify connection to GPIB board**

/home/laser/linux-gpib-3.2.20/examples/ 🡪 **ibtest**

# dmesg | grep gpib

# /sbin/lspci | grep PCI

# lspci –v : to see all port PCI.

On PC0 5 for the GPIB connected to Vinj I have:

*:04.0 Communication controller: National Instruments PCI-GPIB (rev 02)*

 *Flags: bus master, medium devsel, latency 32, IRQ 72*

 *Memory at d40fd000 (32-bit, non-prefetchable) [size=2K]*

 *Memory at d40f4000 (32-bit, non-prefetchable) [size=16K]*

 *Kernel modules: tnt4882*

*C44:06.0 Communication controller: National Instruments PCI-GPIB (rev 02)*

 *Subsystem: National Instruments PCI-GPIB*

 *Flags: bus master, medium devsel, latency 32, IRQ 77*

 *Memory at d40ff000 (32-bit, non-prefetchable) [size=2K]*

 *Memory at d40f8000 (32-bit, non-prefetchable) [size=16K]*

 *Kernel modules: tnt4882*

So we see the driver should be tnt4882 to control the device Vinj.

Be sure to configure appropriate entries in **/etc/gpib.conf** for the card(s)

* Put name of devices

linux gpib driver version: linux-gpib-3.2.08 ( with minor modifications)

---------------------------------------------------------------

I) hard ware installation:

 1) power off computer

 2) install PCI-GPIB card in one of PCI slot

 3) turn on computer

 for plug and play OS one will see the PCI-GPIB card recognised by BIOS automatically.

[qzpeng@localhost gpib\_config]$ /sbin/lspci | grep PCI

04:09.0 Communication controller: National Instruments PCI-GPIB (rev 01)

[qzpeng@localhost gpib\_config]$ ls /proc/bus/pci/04

05.0 09.0

II) driver installation:

 1) unpack linux-gpib-3.2.08 to '/usr/src/linux-gpib-3.2.08/'

 2) follwoing instructions from 'INSTALL'

 i) 'make vmlinux' 'make modules' ?

 ii) ./configure --> make --> make install (as root)

 iii) edit '/etc/gpib.conf to match your interface board and devices.

 iv) load the driver module by /sbin/modprobe tnt4882'

 v) setup the driver by 'gpib\_config --minor 0'

-----------------------------------------------------------------

Comments on installation check:

0) supposed linux-gpib installed in your kernel successfully.

1) At computer boot up, the pci-gpib card will be automatically recognised by OS and driver/module loaded up automatically. (plug and play?)

 To check it:

 a) /sbin/lspci for PCI card recognised

 b) /sbin/lsmod for pci-gpib driver/module installed, 'tnt4882', 'nec7210' and 'gpib\_common'

2) Configure the gpib board and device based on /etc/gpib.conf

 /usr/src/linux-gpib-3.2.08/lib/gpib\_config/gpib\_config --minor 0

3) You can alway check the message for each procedure by looking at

 dmesg | tail -50 or

 tail -50 /var/log/messages

 less /proc/interrupts

 less /proc/ioports

**4) When system got fatal error, you can try unload/reload the module by**

 **/sbin/rmmod tnt4882**

 **/sbin/modprobe tnt4882**

 **/gpib\_config --minor 0**

 **dmesg | tail -50**

5) You may need root priviledge to make your code working... like the /dev/gpib# is R/W accessible.

[root@localhost examples]# ls -l /dev/gpib0

crw-rw-rw- 1 root root 160, 0 Feb 7 02:03 /dev/gpib0

Examples to show:

/sbin/lspci | grep PCI

unload module tnt4882 --> driver for the NI pci-gpib card

[root@localhost examples]# /sbin/lsmod | grep tnt

tnt4882 29572 0

nec7210 23424 1 tnt4882

gpib\_common 37828 2 tnt4882,nec7210

1) load module tnt4882

[root@localhost examples]# /sbin/modprobe tnt4882

[root@localhost examples]# /sbin/lsmod | grep tnt

tnt4882 29572 0

nec7210 23424 1 tnt4882

gpib\_common 37828 2 tnt4882,nec7210

[qzpeng@localhost gpib]$ dmesg | tail -50

2) configure ni\_pci gpib board

[root@localhost gpib\_config]# ./gpib\_config --minor 0

[qzpeng@localhost gpib]$ dmesg | tail -50

||||||||||||||||||||||||||

gpib: (debug) request module returned 256

ACPI: PCI Interrupt 0000:04:09.0[A] -> Link [LNKB] -> GSI 17 (level, low) -> IRQ 22

PCI: Enabling bus mastering for device 0000:04:09.0

mite: 0xdfeff000 mapped to f8b80000

mite: daq: 0xdfef8000 mapped to f8e1e000

tnt4882: irq 22

 **YOKO7555 brancher avec RS232🡪 USB sur ecal-laser-rom-05.**

<https://help.ubuntu.com/community/Minicom>

1. Dmesg | grep tty ou lsusb

ALL OK :

*serial8250: ttyS1 at I/O 0x2f8 (irq = 3) is a 16550A*

*00:05: ttyS0 at I/O 0x3f8 (irq = 4) is a 16550A*

*00:06: ttyS1 at I/O 0x2f8 (irq = 3) is a 16550A*

*usb 1-1.4: pl2303 converter now attached to ttyUSB0*

1. Install “minicom”: sudo yum install minicom.

<http://www.cyberciti.biz/tips/connect-soekris-single-board-computer-using-minicom.html>

1. Run : cd /usr/bin 🡪 sudo ./minicom –s
2. Config Modem replace /dev/modem by /dev/ttyUSB0
3. Save exit
4. type
1 sudo minicom -s
2 select serial port setup
4 press b to select row B – Lockfile Location. Change baud rate to 9600.
5 change /var/lock to **/var/unlock**
5. Run “sudo minicom –c on”
6. Reading all the time : 499E-3 pour 500mV !

**Green Laser:**

I have created on ecallaser@vmepc-ecal-laser-01 : folder green inside cd ~ecallaser

(Where there is /dejardin )

./green is the GUI same as Marc program dtl419 and shutter.

This program could be launched from other PC remotely to communicate with GREEN .

* ./ssh2\_example to be run on ecal-laser-room-05 for example.

**Create and compile GUI program:**

* Create .c file
* Create .fd : open fdesign
* Create a Makefile and modified it (copy from the green laser for an example)
* Il faut faire (source .setup avant de pouvoir compiler ?? que pour dim et program laser) et aussi # **tcsh**

#### With Laser account !

#cd dim

#tcsh

# make ok ..

**Logfile slow monitoring:**

Start laser with ./run to create a logfile.

…

….

Opening shutter B: 601.423000 ms 🡪 **if shutter B = 0ms means it is the Green laser online.**

 Total: 1797218.000000 us

laser.c:1371, 01-Jul-2015--12:52:16, 1, 2, 9, 42, -55

mean :

2= DP2-1

9=% attenuation

42= channel

laser.c:1371, 01-Jul-2015--12:52:16, 1, 1, 5, 42, -55

mean :

1= green laser (the second 1)

5 = % attenuation

### New rack PC August 2020

### All related issue are on Jira: <https://its.cern.ch/jira/browse/CMSONS-12548>

### New PCs are 1U : need to connect again the Startech box to have 2 GPIB on ecal-laser-room-02.

### VME PC: model R340

### PC02 and 05: model R440

### First test failed with GPIB.

### PC have no GUI interface. VNC issue

### Evgueni get the good library NI :

<https://www.ni.com/fr-fr/support/downloads/drivers/download.ni-linux-device-drivers.html>

The command:gpibtswgive:

**Remotely to start display application: email from Giacomo:**

“I have just tried doing:

**ssh -XY ecal-laser-room-05**

startx

using both my account or ecalpro but I have some errors like:

xauth:  file /nfshome0/ecalpro/.serverauth.24667 does not exist

**List of Packages needed** (to ask sysadmin to be included on profile puppet) :

Drivers + software:

1. ni488.2 version 2020  from the page <https://www.ni.com/fr-fr/support/downloads/drivers/download.ni-linux-device-drivers.html>
2. linux-gpib  from the page <https://github.com/vddvss/linux-gpib-packaging>
3.Acqiris Software from the page <https://www.keysight.com/main/software.jspx?ckey=1325010&id=1325010&cc=CH&lc=ger&jmpid=zzfindacqiris-software-cd-rom>

Software:
1. root "yum install root" works.
2. xforms +xforms-devel ; "yum istall xforms" works
3. dim from the page <http://dim.web.cern.ch/>

Hi Diego,
I have tried to extract the minimal list of kernel modules and interface libraries.
It is here, but I am not sure it will be enough. We can try.

Kernel modules:
/lib/modules/3.10.0-957.1.3.el7.x86\_64/extra/gpib\_common.ko.xz
/lib/modules/3.10.0-957.1.3.el7.x86\_64/extra/nec7210.ko.xz
/lib/modules/3.10.0-957.1.3.el7.x86\_64/extra/tnt4882.ko.xz
/lib/modules/3.10.0-957.1.3.el7.x86\_64/extra/nikal.ko.xz
/lib/modules/3.10.0-957.1.3.el7.x86\_64/extra/nipalk.ko.xz
/lib/modules/3.10.0-957.1.3.el7.x86\_64/extra/ni488k.ko.xz
/lib/modules/3.10.0-957.1.3.el7.x86\_64/extra/agmodinst.ko

Libraries:
/lib64/libgpib.so.0
/lib/x86\_64-linux-gnu/libni4882.so.2
/lib/x86\_64-linux-gnu/libni488config.so.15
/lib/x86\_64-linux-gnu/libnipalu.so.1
/lib64/libAgMD1.so
/lib64/libAgMD1Fundamental.so
/lib64/libAqDrv4.so

Cheers,
   Evgueni