EPICS Control command for the Emittance Measurement Unit (EMU)

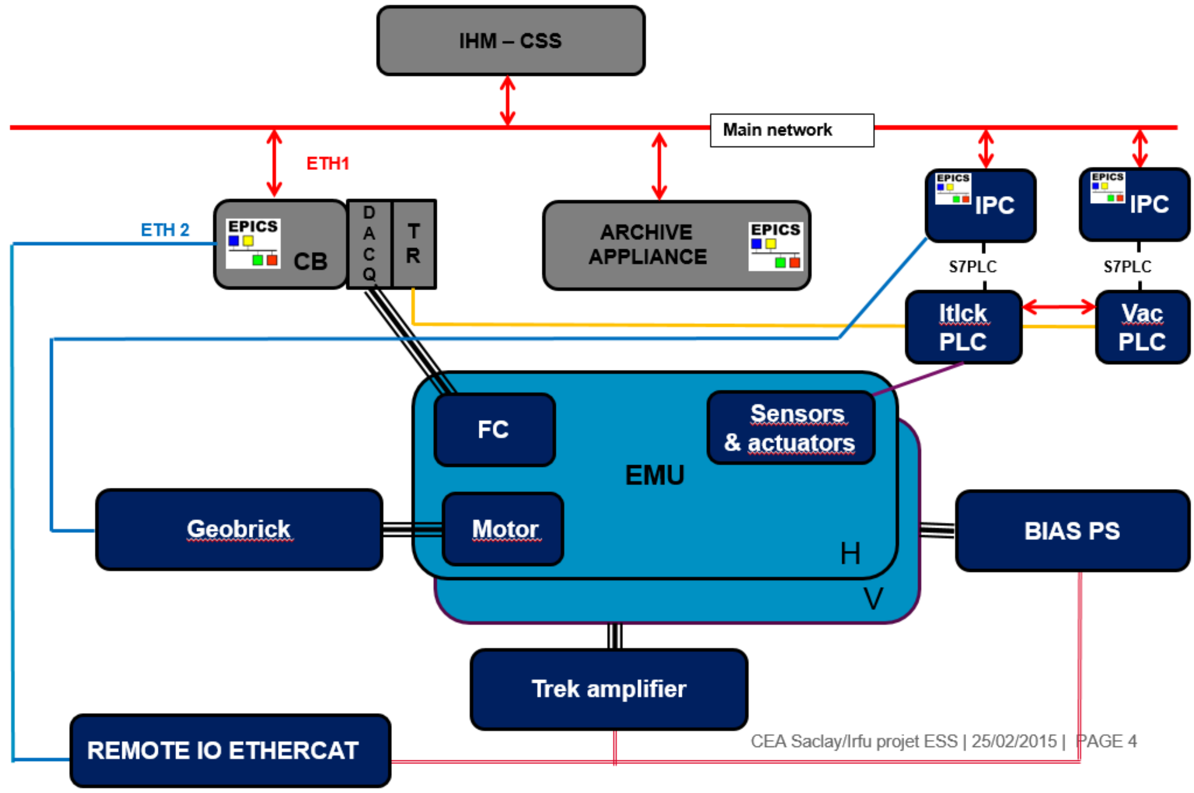
version 1.1

Reference	*Date*
	August 31, 2017

	Editor	*Reviewer*	*Approving*
CEA	Jean-François DENIS	Françoise Gougnaud	

Table of contents

Synoptique



1 - Presentation of each device

Master Server

This main goal of the master server is to centralize all the files needed by each PC/VME/IPC…

Trek amplifier

The Trek model 609E-6 is designed to provide a bi-polar output voltages in the range of 0 to +/- 4kV DC to power plates. It can be configured as a noninverting, inverting amplifier with a fixed gain of 1000V/V. The slow rate is greater than 150V/µs. The control can be done by an analog signal 0-10V.

An EMU measurement can only be done for one plan, so that means only once one EMU can run. So to reduce the cost of Treck power supply, it was decided to use only two Trek amplifiers for two EMU and an electrical switch which allows to switch the power to the plate of the vertical EMU or the Horizontal EMU. The safety of this device is controlled by the PLC.

BIAS Power supply

The ISEG model THQ 1CH 2HE 60W is designed to provide an output voltage in the range of 0 to +30kV. The control is provided by an analog signal 0-5V. Each EMU needs one bias power supply.

Remote EtherCAT IO

ETherCAT is an open real-time Ethernet network developed by Beckhoff in order to control different kinds of remote IOs like analog I/O, binary I/O… This standard protocol and remote IOs were chosen by ***ESS***.

The EtherCAT EPICS driver is available and provided by the EEE.

Geobrick

Geobrick is a motor controller and it can be controlled by TCP.

EMU motors were chosen according to the characteristics of this controller and validated by ***ESS***.

This controller includes General Purpose Input Output (GPIO) in order to control external device like a brake.

Faraday Cup (FC)

The goal of the FC is to collect the current of the beam.

Fast- Acquisition (DACQ)

A Fast- Acquisition is needed to measure the current of the Faraday cup and the voltage on both plates of each EMU.

Timing generator and receiver

The timing generator is an independent VME card. It allows to generate time events.

The timing receiver receives time event from the Timing generator, and can generate a trigger signal on its output which is connected directly to the terminal block of the Fast- acquisition.

PLC interlock

The PLC ensures the safety of both EMU. It controls the movement of both EMUs, and avoids any mechanical issues. It will also give authorizations to move an EMU according to the context of the beam.

Control Box (CB)

The Control Box is a VME crate which integrates the mother board (***IFC1210***), Acquisition board (***D-tAcq***) and timing system. On the ***IFC1210*** runs a real-time (RT) Linux. It controls also every ETherCAT remote IO on a dedicated network card.

Industrial PC (IPC)

An IPC is used to control the data exchanged between the PLC and EPICS, and the control of the **Geobrick**.

Archiving

The Archiving allows storing all data needed to extract data for post treatment.

General User Interface – CSS

The tool used to design and run the General User Interface is Control System Studio (CSS). This is the standardized tool for ESS User Interface.

2 – Description of the EPICS control

Trek amplifier

EPICS control

Each amplifier is controlled by only one voltage. The input range is 0-10V.

For setting voltage, the Beckoff module **ES4134** is used. It can control four analog outputs (-10V/+10V) with a 16-bit resolution.

A reading voltage is also needed. Each amplifier has a monitoring voltage (0-10V) provided on a BNC connector. The output of each plate is connected directly to the Fast- acquisition terminal block.

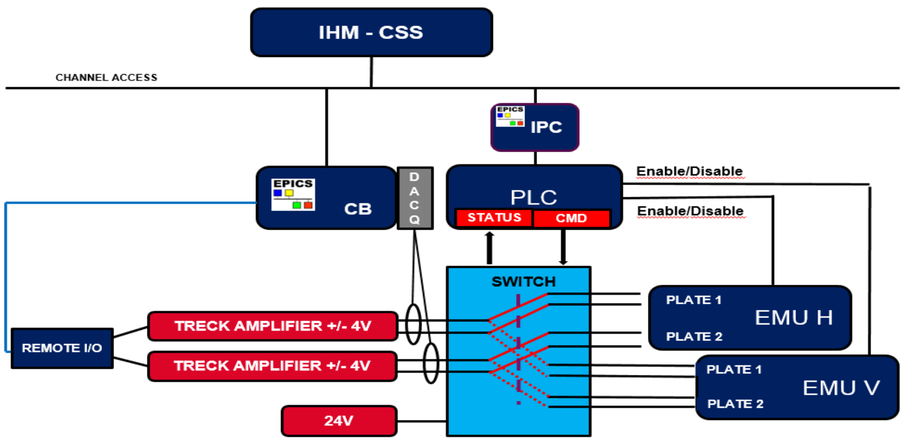
EPICS module required: ecat2db, version : 0.4.3

Lines in substitution files:

```
file ecat2el41xx.template
{
         pattern {PREFIX,CH_ID, SLAVE_IDX,PDO_IDX, EOFF, ESLO, DRVL, DRVH, PREC,EGU}
         ##### Trek Amplifier #######
         {"LNS-ISRC-010:PBI-EM","HVhor-SP",  3, 0, 0, 0.0003051850947599719, -10, 10,5,"kV"}
         {"LNS-ISRC-010:PBI-EM","HVver-SP",  3, 2, 0, 0.0003051850947599719, -10, 10,5,"kV"}
         …
}
file ecat2slave.template
{
         pattern {PREFIX,DTYP,MOD_ID,SLAVE_IDX,DEVICENAME}
         ##### Bias power supply & Trek power supply ####
         {"LNS-ISRC-010:PBI-EM","ES4134","ES4134",3,""}
         …
}
```

PLC control

Each amplifier has an input to enable/disable the High Voltage (ON/OFF). This input is controlled by the PLC in order to ensure the control of one EMU at each time. Like previously mentioned, an electrical switch is used for Plates power supplies. The PLC sends an electrical signal to control the switching, and gets the status.



Bias power supply

EPICS control

Each bias power supply is controlled by an analog signal. The input and output (voltage and current) control range is 0-5V.

For the setting of the voltage and current, the Beckoff module **ES4104** is used. It can control four analog outputs (0V/+10V) with a 16-bit resolution.

For the reading of the voltage and current, the Beckoff module **ES3164** is used. It can control four analog inputs (0V/+10V) with a 16-bit resolution.

EPICS module required: ecat2db, version: 0.4.3

Lines in substitution files:

```
file ecat2el41xx.template
{
          …
          ##### Bias power supply ####
          {"LNS-ISRC-010:PBI-EM","BPS1-CurS", 2 ,0, 0,0.000244148, 0, 8,1,"mA"}
          {"LNS-ISRC-010:PBI-EM","BPS1-VolS", 2 ,1, 0 , 0.09155553, 0, 1500,1,"V"}
          {"LNS-ISRC-010:PBI-EM","BPS2-CurS", 2 ,2, 0, 0.000244148, 0, 8,1,"mA"}
          {"LNS-ISRC-010:PBI-EM","BPS2-VolS", 2 ,3, 0 , 0.09155553, 0, 1500,1,"V"}
}

file ecat2el316x.template
{
          pattern {PREFIX,CH_ID, SLAVE_IDX, PDO_IDX}
          ##### Bias power supply ####
          {"LNS-ISRC-010:PBI-EM","BPS1-CurR",1 ,0, 0, 0.000244148}
          {"LNS-ISRC-010:PBI-EM","BPS1-VolR",1 ,1, 0, 0.09155553}
          {"LNS-ISRC-010:PBI-EM","BPS2-CurR",1 ,2, 0, 0.000244148}
          {"LNS-ISRC-010:PBI-EM","BPS2-VolR",1 ,3, 0, 0.09155553}
}

file ecat2slave.template
{
          pattern {PREFIX,DTYP,MOD_ID,SLAVE_IDX,DEVICENAME}
          ##### Bias power supply & Trek power supply ####
          {"LNS-ISRC-010:PBI-EM","ES4134","ES4134",3,""}
          {"LNS-ISRC-010:PBI-EM","EL4104","EL4104",2,""}
          {"LNS-ISRC-010:PBI-EM","ES3164","ES3164",1,""}
}
```

PLC control

Each BIAS power supply has an input to enable/disable the High Voltage (ON/OFF). This input is controlled by the PLC in order to assure the control of one EMU at each time.

Geobrick

EPICS control

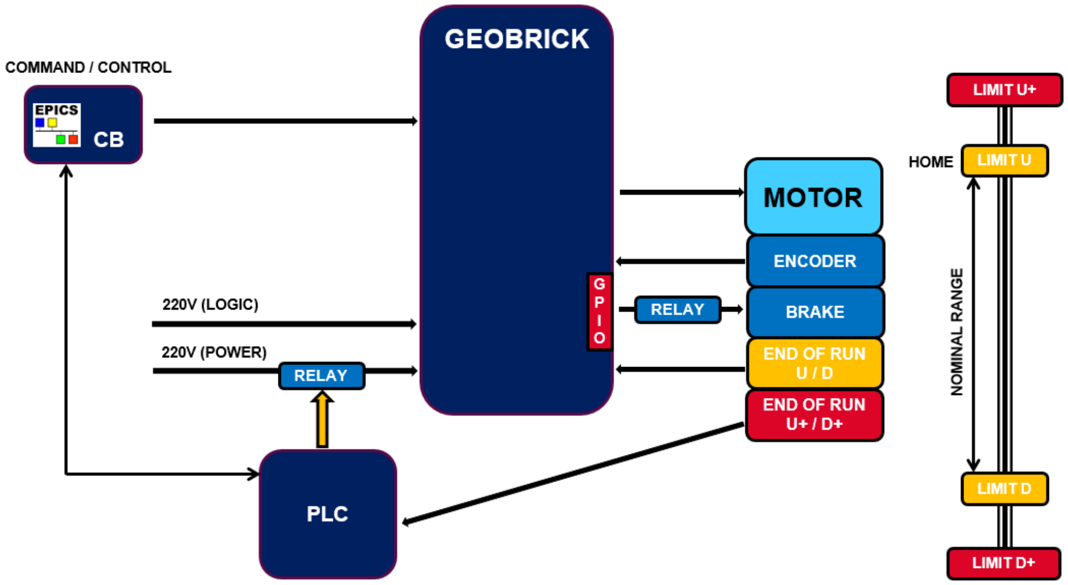
The Geobrick needs firstly to be correctly configured. For each motor, the configuration is different. The first step is to collect all the information about the motor (Power, current, etc…) and the encoder if needed. To test this configuration file, a mock motor with an encoder is used.

After the configuration file loading, the Geobrick can be controlled by ***EPICS*** over TCP.

To control the brake, Geobrick provides some GPIO. A short program needs to be written in order to put ON/OFF the brake when a brake command is sent to the Geobrick.

The EPICS module "SCANNING" is used to synchronize acquisitions with the timing system, the setting of PS voltage on plate, the moving of the motor and the saving data on the archiver.

A set of limit switches is used for the nominal run and is connected to the Geobrick.



EPICS modules required: singlemotion, version :1.4.6

Emu-motor,0.1.0

Some templates files are also useful to complete the control of the **Geobrick**

Template Name	Description
emu_console.template	Template which create a terminal ASK/ANSWER for Geobrick
emu_get_value_pmac.template	Template binary output record for geobrick
emu_set_value_pmac.template	Template binary input record for geobrick

PLC control

Another set of limit switches is used for the safety hardware and is directly sent to the PLC. If a hardware limit (U+/D+) is active, the PLC stops the power of the Geobrick.

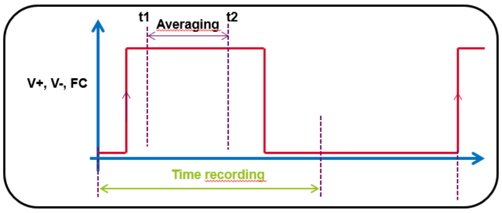
Fast Acquisition

EPICS control

The Fast- acquisition is done by a D-tAcq (ACQ420FMC) board. This board was chosen by ***ESS***. The sampling frequency goes up to 2 Mhz with a 16-bit resolution. This board includes 4 Channels. A terminal bock is provided with the board.

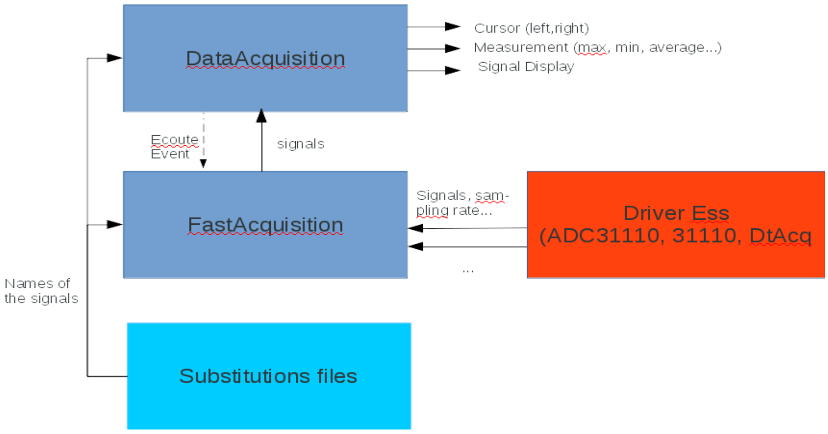
An input trigger with BNC connector is located on the terminal block. It will be used for the timing receiver.

The signal average of each signal is calculated between two cursors like presented just below:



Three modules are re:

- EPICS ***Driver ESS*** in order to access cards
- ***FastAcquisition*** to configure the acquisition through EPICS Driver ESS
- ***DataAcquisition*** for displaying or calculs



<u>EPICS modules required</u>: ifcdaq, version: 0.2.0
Fastacquisition, version: 1.0.4-catania
dataacquisition, version:1.1.0
pev, version: 0.1.1

```
file DataAcquisitionChannel.template
{
Pattern {SECTION, SUBSECTION, DISC, DEVICE, SIGNAL, DESCRIPTION,  EVENT_IN, NELM, SIZE,
SAMPLING_RATE_PV}
         {"LNS","ISRC-010","PBI-EMV","FC","CurrR","Current FC",10,16000,520,LNS-ISRC-
010:PBI-EM:CARD0:SAMPLINGRATE-RB}
         {"LNS","ISRC-010","PBI-EM","HV1","VoltR","Voltage HV1 FC",11,16000,520,LNS-ISRC-
010:PBI-EM:CARD0:SAMPLINGRATE-RB}
         {"LNS","ISRC-010","PBI-EM","HV2","VoltR","Voltage HV2 FC",12,16000,520,LNS-ISRC-
010:PBI-EM:CARD0:SAMPLINGRATE-RB}
}

file FastAcquisition.template
{
Pattern { SECTION,SUBSECTION, DISC, DEVICE, SIGNAL,NICKNAME,EVENT_OUT,CARD,CHANNEL,NELM }

        {"LNS","ISRC-010","PBI-EMV","FC","CurrR","CurrR",10,0,3,16000}
        {"LNS","ISRC-010","PBI-EM","HV1","VoltR","HV1VoltR",11,0,0,16000}
        {"LNS","ISRC-010","PBI-EM","HV2","VoltR","HV2VoltR",12,0,1,16000}
}

file FastAcquisitionInformation.template
{
Pattern { CARD,FMC,NELM,PREFIX}
        {0,2,16000,"LNS-ISRC-010:PBI-EM"}
}
```

Timing generator and receiver

The link between these two cards is an optical fiber. From the timing receiver, a copper line is used for the connection to the acquisition board (D-tAcq) through the terminal block.

EPICS modules required: mrfioc2, version: 2.7.13

Lines in startup file:

```
### Initialize EVR ###
mrmEvrSetupPCI($(EVR), $(EVR_PCIDOMAIN), $(EVR_PCIBUS), $(EVR_PCIDEVICE), $(EVR_PCIFUNCTION))

### LOAD RECORDS ###

#Load EVG records
dbLoadRecords("evg-vme-230.db", "DEVICE=$(EVG), SYS=$(SYS)")

#Load EVR records
dbLoadRecords("evr-pmc-230.db", "DEVICE=$(EVR), SYS=$(SYS)")
dbLoadRecords("evr-softEvent.template", "DEVICE=$(EVR), SYS=$(SYS), EVT=$(EVR_PUL0_EVENT),
CODE=$(EVR_PUL0_EVENT)")
dbLoadRecords("evr-pulserMap.template", "DEVICE=$(EVR), SYS=$(SYS), EVT=$(EVR_PUL0_EVENT),
PID=0, F=Trig, ID=0")
```

EMU

The main control is based on the EPICS scanning application.

Some templates files are also useful to complete the control.

Template Name	Description
EMU_BUFF_CHAN.template	Template which save data in a buffer for each channel (V+, V-, FC Current) and each column (=Position)
EMU_BUFF_POS.template	Template which save motor position in a buffer
EMU_CALCUL.template	Template which do all calculs
EMU_DACQ_SYNC.template	Template which configure cursors for acquisition
EMU_PROCESS.template	Template for configuration process

Macro :

EMU_BUFF_CHAN.template:

DEVICE	Name of the device
DET	Name of the detector
NELM_COMP	Number of elements for the circular buffer (Compress)
SIGNAL_IN	Name of the input signal to be buffered
SIGNAL2	Name of the output signal buffered
NELM	Number of element for the input waveform
SCAN_SLIT	Name of the SCAN MOTOR
SCAN_EF	Name of the SCAN Power Supply

EMU_BUFF_POS.template:

DEVICE	Name of the device
NELM_COMP	Number of elements for the circular buffer (Compress)
MTRREC	Name of the motor

EMU_CALCUL.template:

DEVICE	Name of the device
NELM_COMP	Number of elements for the circular buffer (Compress)
DAQPREFIX	Name of the DAQ prefix
DET2	Name of the detector for PS1
DET2WF	Name of the waveform PS1 compressed
DET3	Name of the detector for PS2
DET3WF	Name of the waveform PS2 compressed
SCAN_SLIT	Name of the SCAN MOTOR
SCAN_EF	Name of the SCAN Power Supply

EMU_CALCUL.template:

PREFIX	Prefix
DEVICE	Name of the device
SIGNAL1	Name of the signal ICurr
SIGNAL2	Name of the signal VPS1
SIGNAL3	Name of the signal VPS2

Archiving

The tool used to archive is the ***Archive Appliance***. This archiving is recommended by ***ESS***. At Catania the archiving installed for the source is used to store all data for EMU. This kind of archiving is available to store waveform of 10000 points each 14Hz. It gives also the possibility to extract data by a Python script.

The archiving is started for each emittance measurement.

About the space to save data: the maximum width of the beam pulse is around 6ms every 14Hz, so that means with a sample frequency of 1 Mhz, the maximum number of points to save is about 6000 points per waveform.

Three waveforms need to be saved at each pulse, so the maximum storage space for an emittance measurement is about: 80 positions x 80 angles x 3 waveforms x 6000 points = 822 Megabytes.

The storage capacity is designed accordingly to our need.

3 - Startup script:

IPC startup script:

```
require emu-plc, 0.1.0
require emu-motor, 0.1.0
require singlemotion,1.4.6+

# Set environmental variables
epicsEnvSet("ASYN_PORT",        "GEOBRICK_ASYN")
epicsEnvSet("PMAC_IP",              "10.10.3.42")
epicsEnvSet("PMAC_PORT",        "1025")


# fonctions from TPMAC
# Connection to GEOBRICK, create a asyn port
pmacAsynIPConfigure($(ASYN_PORT), $(PMAC_IP):$(PMAC_PORT))

# s7plcConfigure (PLCname, IPaddr, port, inSize, outSize, bigEndian, recvTimeout, sendIntervall)
s7plcConfigure ("plc", "10.10.2.3", 2000, 40, 6, 1, 1000, 500)

# i/o PLC
dbLoadRecords("emu_input.db")
dbLoadRecords("emu_output.db")

# load PMAC (geobrick) database
dbLoadRecords("emu_get_value_pmac.db")
dbLoadRecords("emu_set_value_pmac.db")
dbLoadRecords("emu_console.db")

#communication PLC PMAC
dbLoadRecords("emu_set_bo_PLC.db")
dbLoadRecords("emu_set_bo_pmac.db")
```

VME startup script:

```
# Required modules ###
require ifcdaq,0.2.0+
require singlemotion,1.4.6+
require mrfioc2,2.7.13
require pev,0.1.1
require ecat2db,0.4.3
require FastAcquisition,1.0.4-catania
require DataAcquisition,1.1.0
#require emu-motor-test,1.0.0
require tpmac,3.11.2+
require PVArchiving,1.0.2
require emu,1.0.6


### MACROS ###
# PMAC common macros3       )
epicsEnvSet("PMACPREFIX", "LNS-ISRC-010:PBI-EM-PMAC")
epicsEnvSet("PMACPORT",   "GEOBRICK_ASYN")
epicsEnvSet("MOTOR_PORT", "pmac1" )
epicsEnvSet("NB_AXES", "2" )

# ARCHIVE macros #
epicsEnvSet("ARCHIVE-MACRO-EMUV","LNS-ISRC-010:PBI-EMV")
epicsEnvSet("ARCHIVE-MACRO-EMUH","LNS-ISRC-010:PBI-EMH")

# Motor Macros
epicsEnvSet("EGU",       "mm"   )
epicsEnvSet("DIR",       "Pos"  )
epicsEnvSet("MRES",      "0.000249515")
epicsEnvSet("PREC",      "4"    )
epicsEnvSet("DHLM",      "4"   )
epicsEnvSet("DLLM",      "-200"  )
epicsEnvSet("VELO",      "1.5"    )
epicsEnvSet("HVEL",      "1.5"    )
epicsEnvSet("VBAS",      "0.1"  )
epicsEnvSet("VMAX",      "7"    )
epicsEnvSet("ACCL",      "5"  )
epicsEnvSet("BDST",      "0"    )
epicsEnvSet("INIT",      ""     )
epicsEnvSet("OFF",       "0"    )

epicsEnvSet("MOTOR_NAME1",  "MTR1")
epicsEnvSet("AXIS_NO1",     "1")
epicsEnvSet("MOTOR_NAME2",  "MTR2")
epicsEnvSet("AXIS_NO2",     "2")

# Motor Status macros
epicsEnvSet("SCAN",         "1 second")
epicsEnvSet("OVERHEAT1",    "MAJOR"   )
epicsEnvSet("SWITCH_OFF1",  "MINOR"   )
epicsEnvSet("OVERHEAT2",    "0"       )
epicsEnvSet("SWITCH_OFF2",  "0"       )
epicsEnvSet("MOTOR_ERROR1", "0"       )
epicsEnvSet("MOTOR_ERROR2", "0"       )
epicsEnvSet("MOTOR_ERROR3", "0"       )
epicsEnvSet("MOTOR_ERROR4", "0"       )

# EVG Macros
epicsEnvSet("SYS"             "SYS0")
epicsEnvSet("EVG_VMESLOT"     "8")
```

```
# EVR Macros
epicsEnvSet("EVR_PCIDOMAIN",    "0000")
epicsEnvSet("EVR_PCIBUS",       "05")
epicsEnvSet("EVR_PCIDEVICE",    "00")
epicsEnvSet("EVR_PCIFUNCTION",  "0")
epicsEnvSet("EVR",              "EVR0")
epicsEnvSet("EVR_PUL0_EVENT",   "14")
epicsEnvSet("EVR_PUL0_DELAY",   "$(EVR_EV14_OUT0_DELAY=0)")
epicsEnvSet("EVR_PUL0_WIDTH",   "$(EVR_EV14_OUT0_WIDTH=1000)")

#DAQ macros
epicsEnvSet("DAQPREFIX",  "LNS-ISRC-010"  )
epicsEnvSet("DAQBUFSIZE", "1024"         )
epicsEnvSet("BUFFERSIZE", "40000"         )

#Scanning macros
epicsEnvSet("SCANPREFIX_EMU_V", "LNS-ISRC-010:PBI-EMV")
epicsEnvSet("SCANPREFIX_EMU_H", "LNS-ISRC-010:PBI-EMH")
epicsEnvSet("SCAN_SLIT", "MTR"          )
epicsEnvSet("SCAN_EF", "PS"              )
epicsEnvSet("SCAN_POINTS_EF", "400"     )
epicsEnvSet("SCAN_POINTS_SLIT", "100"    )

epicsEnvSet(EPICS_CA_MAX_ARRAY_BYTES, 8000000)

# Connection to PMAC and setup for $(NUM_MOTORS) motors
pmacAsynIPConfigure($(PMACPORT), "10.10.3.42:1025")
pmacCreateController($(MOTOR_PORT), $(PMACPORT), 0, $(NB_AXES), 50, 300)
### Motor Vertical
pmacCreateAxis($(MOTOR_PORT), $(AXIS_NO1))
### Motor Horizontal
pmacCreateAxis($(MOTOR_PORT), $(AXIS_NO2))

# Initialize EVR
mrmEvrSetupPCI($(EVR), $(EVR_PCIDOMAIN), $(EVR_PCIBUS), $(EVR_PCIDEVICE), $(EVR_PCIFUNCTION))

# Initialize daq
ndsCreateDevice(ifcdaq, CARD0, card=0, fmc=2)

### LOAD RECORDS ###

######### EVR #########
dbLoadRecords("evr-pmc-230.db", "DEVICE=$(EVR), SYS=$(SYS)")
dbLoadRecords("evr-softEvent.template", "DEVICE=$(EVR), SYS=$(SYS), EVT=$(EVR_PUL0_EVENT),
CODE=$(EVR_PUL0_EVENT)")
dbLoadRecords("evr-pulserMap.template", "DEVICE=$(EVR), SYS=$(SYS), EVT=$(EVR_PUL0_EVENT),
PID=0, F=Trig, ID=0")

######### MOTORS #########
# EMU vertical
dbLoadRecords("motor.template",
"PREFIX=$(PMACPREFIX),MOTOR_NAME=$(MOTOR_NAME1),MOTOR_PORT=$(MOTOR_PORT),AXIS_NO=$(AXIS_NO1),
EGU=$(EGU),DIR=$(DIR),MRES=$(MRES),PREC=$(PREC),DHLM=$(DHLM),DLLM=$(DLLM),HVEL=$(HVEL),VELO=$
(VELO),VBAS=$(VBAS),VMAX=$(VMAX),ACCL=$(ACCL),BDST=$(BDST),INIT=$(INIT),OFF=$(OFF)")

# EMU horizontal
dbLoadRecords("motor.template",
"PREFIX=$(PMACPREFIX),MOTOR_NAME=$(MOTOR_NAME2),MOTOR_PORT=$(MOTOR_PORT),AXIS_NO=$(AXIS_NO2),
EGU=$(EGU),DIR=$(DIR),MRES=$(MRES),PREC=$(PREC),DHLM=$(DHLM),DLLM=$(DLLM),HVEL=$(HVEL),VELO=$
(VELO),VBAS=$(VBAS),VMAX=$(VMAX),ACCL=$(ACCL),BDST=$(BDST),INIT=$(INIT),OFF=$(OFF)")
```

```
######### MOTORS STATUS  #########
# EMU vertical
dbLoadRecords("motorStatus.template",
"PREFIX=$(PMACPREFIX),MOTOR_NAME=$(MOTOR_NAME1),ASYN_PORT=$(PMACPORT),AXIS_NO=$(AXIS_NO1),SCA
N=$(SCAN),EGU=$(EGU),PREC=$(PREC),OVERHEAT1=$(OVERHEAT1),SWITCH_OFF1=$(SWITCH_OFF1),OVERHEAT2
=$(OVERHEAT2),SWITCH_OFF2=$(SWITCH_OFF2),MOTOR_ERROR1=$(MOTOR_ERROR1),MOTOR_ERROR2=$(MOTOR_ER
ROR2),MOTOR_ERROR3=$(MOTOR_ERROR3),MOTOR_ERROR4=$(MOTOR_ERROR4)")

# EMU horizontal
dbLoadRecords("motorStatus.template",
"PREFIX=$(PMACPREFIX),MOTOR_NAME=$(MOTOR_NAME2),ASYN_PORT=$(PMACPORT),AXIS_NO=$(AXIS_NO2),SCA
N=$(SCAN),EGU=$(EGU),PREC=$(PREC),OVERHEAT1=$(OVERHEAT1),SWITCH_OFF1=$(SWITCH_OFF1),OVERHEAT2
=$(OVERHEAT2),SWITCH_OFF2=$(SWITCH_OFF2),MOTOR_ERROR1=$(MOTOR_ERROR1),MOTOR_ERROR2=$(MOTOR_ER
ROR2),MOTOR_ERROR3=$(MOTOR_ERROR3),MOTOR_ERROR4=$(MOTOR_ERROR4)")

######### HOMING PROCEDURE #########
# EMU vertical
dbLoadRecords("motorHoming.template",
"PREFIX=$(PMACPREFIX),MOTOR_NAME=$(MOTOR_NAME1),ASYN_PORT=$(PMACPORT),AXIS_NO=$(AXIS_NO1),PRE
C=$(PREC),EGU=$(EGU)")

# EMU horizontal
dbLoadRecords("motorHoming.template",
"PREFIX=$(PMACPREFIX),MOTOR_NAME=$(MOTOR_NAME2),ASYN_PORT=$(PMACPORT),AXIS_NO=$(AXIS_NO2),PRE
C=$(PREC),EGU=$(EGU)")

######### EMU #########
# common database
dbLoadRecords("EMU_VME_common.db","DEVICE=LNS-ISRC-010:PBI-EM")

# EMU vertical
dbLoadRecords("EMU_VME.db", "DEVICE=$(SCANPREFIX_EMU_V), SCAN_SLIT=$(SCAN_SLIT),
SCAN_EF=$(SCAN_EF),MTRREC=$(PMACPREFIX):$(MOTOR_NAME1), MAX_POINTS_EF=$(SCAN_POINTS_EF),
MAX_POINTS_SLIT=$(SCAN_POINTS_SLIT), DAQNDSPREFIX=$(DAQPREFIX), DAQBUFSIZE=$(DAQBUFSIZE),
DET1WF=PBI-EMV-FC:CurrR_Y, DET2WF=PBI-EM-HV1:VoltR_Y, DET3WF=PBI-EM-HV2:VoltR_Y,
TR_TL=$(SYS)-$(EVR):Pul0-Ena-Sel, TR_TLTSEL=$(SYS)-$(EVR):Event-$(EVR_PUL0_EVENT)-SP.TIME,
PSU_SP=LNS-ISRC-010:PBI-EM:HVhor-SP, PSU_RBV=LNS-ISRC-010:PBI-EM:HVhor-SP, PSU2_SP=LNS-ISRC-
010:PBI-EM:HVver-SP, PSU2_RBV=LNS-ISRC-010:PBI-EM:HVver-SP,BUFFERSIZE=$(BUFFERSIZE)")

# EMU horizontal
dbLoadRecords("EMU_VME.db", "DEVICE=$(SCANPREFIX_EMU_H), SCAN_SLIT=$(SCAN_SLIT),
SCAN_EF=$(SCAN_EF),MTRREC=$(PMACPREFIX):$(MOTOR_NAME2), MAX_POINTS_EF=$(SCAN_POINTS_EF),
MAX_POINTS_SLIT=$(SCAN_POINTS_SLIT), DAQNDSPREFIX=$(DAQPREFIX), DAQBUFSIZE=$(DAQBUFSIZE),
DET1WF=PBI-EMH-FC:CurrR_Y, DET2WF=PBI-EM-HV1:VoltR_Y, DET3WF=PBI-EM-HV2:VoltR_Y,
TR_TL=$(SYS)-$(EVR):Pul0-Ena-Sel, TR_TLTSEL=$(SYS)-$(EVR):Event-$(EVR_PUL0_EVENT)-SP.TIME,
PSU_SP=LNS-ISRC-010:PBI-EM:HVhor-SP, PSU_RBV=LNS-ISRC-010:PBI-EM:HVhor-SP, PSU2_SP=LNS-ISRC-
010:PBI-EM:HVver-SP, PSU2_RBV=LNS-ISRC-010:PBI-EM:HVver-SP,BUFFERSIZE=$(BUFFERSIZE)")

#
# Init ETHERCAT module
#

ecat2configure(0,500,1,1)

#
# Archive module
#

ArchiveConfigure("$(REQUIRE_PVArchiving_PATH)","10.10.0.11:17665")
dbLoadRecords("PVArchiving.template",PREFIX=$(ARCHIVE-MACRO-EMUV))
dbLoadRecords("PVArchiving.template",PREFIX=$(ARCHIVE-MACRO-EMUH))
```

```
#############################################################################
######################### iocInit ############################################
#############################################################################
iocInit

#### Motor homing
seq motorHoming "PREFIX=$(PMACPREFIX),MOTOR_NAME=$(MOTOR_NAME1)"
seq motorHoming "PREFIX=$(PMACPREFIX),MOTOR_NAME=$(MOTOR_NAME2)"
seq homeMtrAuto
"PMACPREFIX=$(PMACPREFIX),MOTOR_NAME=$(MOTOR_NAME1),DEVICE=$(SCANPREFIX_EMU_V),PREFIX=LNS-
ISRC-010"
seq homeMtrAuto
"PMACPREFIX=$(PMACPREFIX),MOTOR_NAME=$(MOTOR_NAME2),DEVICE=$(SCANPREFIX_EMU_H),PREFIX=LNS-
ISRC-010"
seq checkMTRBrake
"PMACPREFIX=$(PMACPREFIX),MOTOR_NAME=$(MOTOR_NAME1),DEVICE=$(SCANPREFIX_EMU_V),PREFIX=LNS-
ISRC-010"
seq checkMTRBrake
"PMACPREFIX=$(PMACPREFIX),MOTOR_NAME=$(MOTOR_NAME2),DEVICE=$(SCANPREFIX_EMU_H),PREFIX=LNS-
ISRC-010"

# SET INTERFACE FROM IFCDAQ TO Allison scanner
# Forward link wf record to Detector guard

# EMU vertical by default
dbpf $(DAQPREFIX):PBI-EMV-FC:CurrR_Y.FLNK $(SCANPREFIX_EMU_V):DET1-GUARD
dbpf $(DAQPREFIX):PBI-EM-HV1:VoltR_Y.FLNK $(SCANPREFIX_EMU_V):DET2-GUARD
dbpf $(DAQPREFIX):PBI-EM-HV2:VoltR_Y.FLNK $(SCANPREFIX_EMU_V):DET3-GUARD

# EMU horizontal: after the switch between the EMU vertical and horizontal, these PVs have to
be processed
dbpf $(DAQPREFIX):PBI-EMH-FC:CurrR_Y.FLNK $(SCANPREFIX_EMU_H):DET1-GUARD
#dbpf $(DAQPREFIX):PBI-EM-HV1:VoltR_Y.FLNK $(SCANPREFIX_EMU_H):DET2-GUARD
#dbpf $(DAQPREFIX):PBI-EM-HV2:VoltR_Y.FLNK $(SCANPREFIX_EMU_H):DET3-GUARD

# Forward link from state record to Detector trigger
dbpf $(DAQPREFIX):PBI-EM:CARD0-STAT-RB.FLNK $(SCANPREFIX_EMU_V):DAQTRG-DAQSTGUARD
#dbpf $(DAQPREFIX):PBI-EM:CARD0-STAT-RB.FLNK $(SCANPREFIX_EMU_H):DAQTRG-DAQSTGUARD

# Forward link number of samples rbv to compress N record, so that detetor know number of
elements to average over.
dbpf $(DAQPREFIX):PBI-EM:CARD0:NSAMPLES-RB.FLNK "$(SCANPREFIX_EMU_V):DET1-COMPRESSN"
#dbpf $(DAQPREFIX):PBI-EM:CARD0:NSAMPLES-RB.FLNK "$(SCANPREFIX_EMU_H):DET1-COMPRESSN"


# Set tsel on WF record to event record in EVR
dbpf $(DAQPREFIX):PBI-EMV-FC:CurrR_Y.TSEL $(SYS)-$(EVR):Event-$(EVR_PUL0_EVENT)-SP.TIME
dbpf $(DAQPREFIX):PBI-EMH-FC:CurrR_Y.TSEL $(SYS)-$(EVR):Event-$(EVR_PUL0_EVENT)-SP.TIME
dbpf $(DAQPREFIX):PBI-EM-HV1:VoltR_Y.TSEL $(SYS)-$(EVR):Event-$(EVR_PUL0_EVENT)-SP.TIME
dbpf $(DAQPREFIX):PBI-EM-HV2:VoltR_Y.TSEL $(SYS)-$(EVR):Event-$(EVR_PUL0_EVENT)-SP.TIME
dbpf $(SYS)-$(EVR):Time-I.TSE -2

############################################################
################### Configuration Timing ###################
############################################################

### Setup EVR
##set the pulser 0 to trigger on output 0
########## TIMING RECEIVER: setup OUT0 ##############
dbpf $(SYS)-$(EVR):Link-Clk-SP           88.0519
dbpf $(SYS)-$(EVR):FrontOut0-Src-SP      0
dbpf $(SYS)-$(EVR):Pul0-Evt-Trig0-SP  $(EVR_PUL0_EVENT)
dbpf $(SYS)-$(EVR):Pul0-Width-SP         20000
dbpf $(SYS)-$(EVR):Pul0-Delay-SP         0
dbpf $(SYS)-$(EVR):FrontOut0-Ena-SP      "Enabled"
```

```
### Setup IFCDAQ PVs ###
# Set trigger repeat to 1
dbpf $(DAQPREFIX):PBI-EM:CARD0:TriggerRepeat 1
# Set trigger source to external GPIO
dbpf $(DAQPREFIX):PBI-EM:CARD0:TRIGGERSOURCE "EXT-GPIO"
sleep(1)
dbpf $(DAQPREFIX):PBI-EM:CARD0-STAT ON
sleep(3)
dbpf $(DAQPREFIX):PBI-EM:CARD0-STAT RUNNING
sleep(1)
dbpf $(DAQPREFIX):PBI-EM:CARD0-STAT RUNNING
# Set the Frequency to 1MHz
dbpf $(DAQPREFIX):PBI-EM:CARD0:SAMPLINGRATE 1000000

### Setup reasonable values for sscan record ###
# Initialize Motor positioners vertical
dbpf $(SCANPREFIX_EMU_V):$(SCAN_SLIT).P4SP -195
dbpf $(SCANPREFIX_EMU_V):$(SCAN_SLIT).P4EP -105
dbpf $(SCANPREFIX_EMU_V):$(SCAN_SLIT).P2SP 1.5
dbpf $(SCANPREFIX_EMU_V):$(SCAN_SLIT).P2EP 1.5
dbpf $(SCANPREFIX_EMU_V):$(SCAN_SLIT).P1SP 5
dbpf $(SCANPREFIX_EMU_V):$(SCAN_SLIT).P1EP 5
dbpf $(SCANPREFIX_EMU_V):$(SCAN_SLIT).NPTS 11
dbpf $(SCANPREFIX_EMU_V):$(SCAN_SLIT).PDLY 0.1

# Initialize Motor positioners horizontal
dbpf $(SCANPREFIX_EMU_H):$(SCAN_SLIT).P4SP -195
dbpf $(SCANPREFIX_EMU_H):$(SCAN_SLIT).P4EP -105
dbpf $(SCANPREFIX_EMU_H):$(SCAN_SLIT).P2SP 1.5
dbpf $(SCANPREFIX_EMU_H):$(SCAN_SLIT).P2EP 1.5
dbpf $(SCANPREFIX_EMU_H):$(SCAN_SLIT).P1SP 5
dbpf $(SCANPREFIX_EMU_H):$(SCAN_SLIT).P1EP 5
dbpf $(SCANPREFIX_EMU_H):$(SCAN_SLIT).NPTS 11
dbpf $(SCANPREFIX_EMU_H):$(SCAN_SLIT).PDLY 0.1

# Initialize Power supply positioners vertical
dbpf $(SCANPREFIX_EMU_V):PROC-AngleMax-SP 79
dbpf $(SCANPREFIX_EMU_V):PROC-AngleMin-SP -79
dbpf $(SCANPREFIX_EMU_V):$(SCAN_EF).NPTS 200
dbpf $(SCANPREFIX_EMU_V):$(SCAN_EF).PDLY 0.04
dbpf $(SCANPREFIX_EMU_V):$(SCAN_EF).AAWAIT 1

# Initialize Power supply positioners horizontal
dbpf $(SCANPREFIX_EMU_H):PROC-AngleMax-SP 79
dbpf $(SCANPREFIX_EMU_H):PROC-AngleMin-SP -79
dbpf $(SCANPREFIX_EMU_H):$(SCAN_EF).NPTS 200
dbpf $(SCANPREFIX_EMU_H):$(SCAN_EF).PDLY 0.04
dbpf $(SCANPREFIX_EMU_H):$(SCAN_EF).AAWAIT 1

dbpf LNS-ISRC-010:PBI-EM:HVver-SP.PREC 4
dbpf LNS-ISRC-010:PBI-EM:HVhor-SP.PREC 4

dbpf $(DAQPREFIX):PBI-EM:CARD0:TriggerEdge 1
dbpf $(DAQPREFIX):PBI-EM:CARD0:NSAMPLES_ms 7
dbpf $(DAQPREFIX):PBI-EMV:LCURSOR_ms 2
dbpf $(DAQPREFIX):PBI-EMV:RCURSOR_ms 6
dbpf $(DAQPREFIX):PBI-EMH:LCURSOR_ms 2
dbpf $(DAQPREFIX):PBI-EMH:RCURSOR_ms 6

dbpf LNS-ISRC-010:PBI-EMV-FC:CurrR:AVG.MDEL -1
dbpf LNS-ISRC-010:PBI-EMV-FC:CurrR:AVG.ADEL -1
dbpf LNS-ISRC-010:PBI-EMH-FC:CurrR:AVG.MDEL -1
dbpf LNS-ISRC-010:PBI-EMH-FC:CurrR:AVG.ADEL -1
dbpf LNS-ISRC-010:PBI-EM-HV1:VoltR:AVG.MDEL -1
dbpf LNS-ISRC-010:PBI-EM-HV1:VoltR:AVG.ADEL -1
```

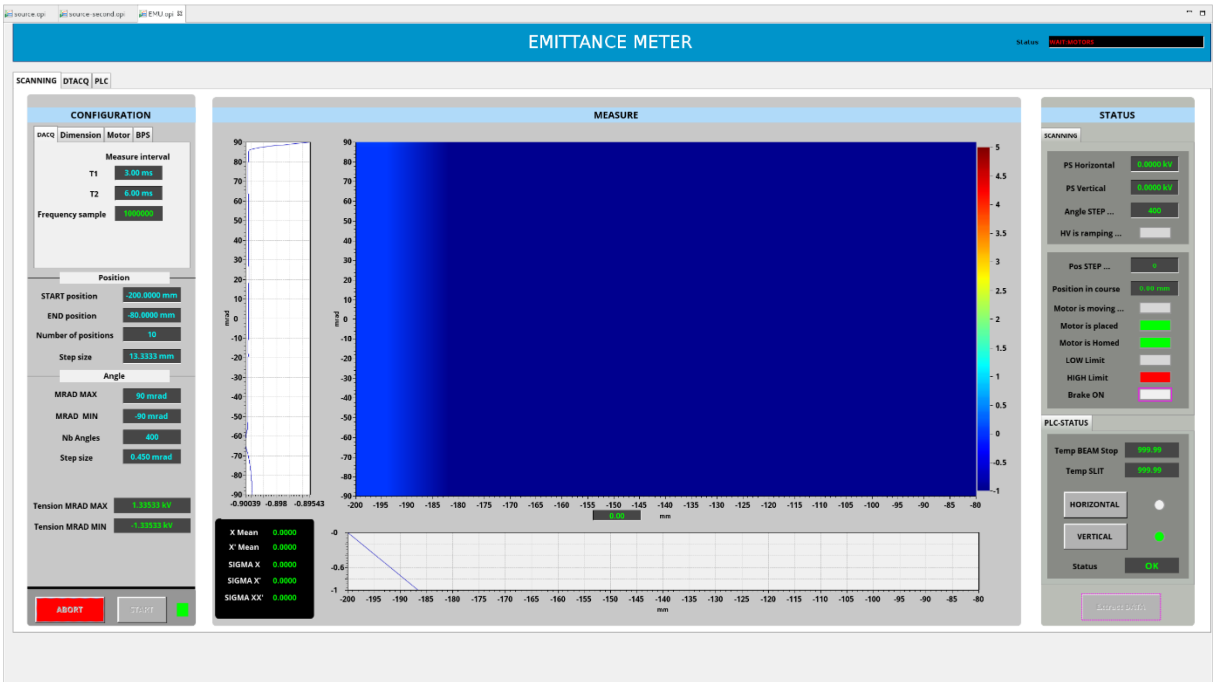
```
############################################################
################### START ARCHIVE ##########################
############################################################

dbpf $(ARCHIVE-MACRO-EMUV):PVS "LNS-ISRC-010:PBI-EM-HV1:VoltR,LNS-ISRC-010:PBI-EM-
HV2:VoltR,LNS-ISRC-010:PBI-EMV-FC:CurrR,LNS-ISRC-010:PBI-BCM:CurR,LNS-ISRC-010:PBI-EMV:BUFF-
MTR-POS")
dbpf $(ARCHIVE-MACRO-EMUV):Archive 0
dbpf $(ARCHIVE-MACRO-EMUH):PVS "LNS-ISRC-010:PBI-EM-HV1:VoltR,LNS-ISRC-010:PBI-EM-
HV2:VoltR,LNS-ISRC-010:PBI-EMH-FC:CurrR,LNS-ISRC-010:PBI-BCM:CurR,LNS-ISRC-010:PBI-EMH:BUFF-
MTR-POS")
dbpf $(ARCHIVE-MACRO-EMUH):Archive 0
```

4 - EPICS General User Interface

Main user interface

The main user interface is composed of three tabs: scanning on real time, Dtacq, and PLC

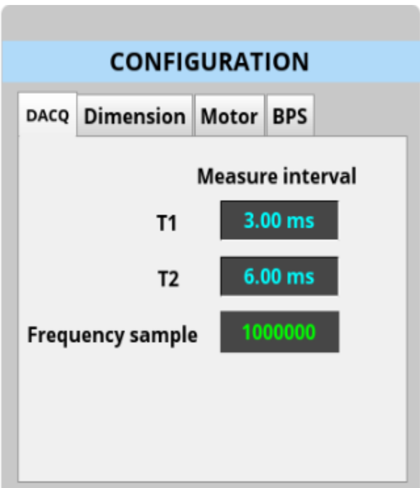


Scanning User interface

Configuration

<u>DACQ</u>:

This tab allows to configure the Acquisition of signals: Cursors Left and Right for average, frequency of sampling:



$(***DTACQ-NAME***)V:LCURSOR_ms

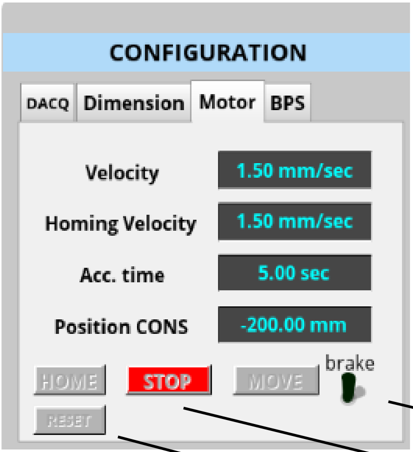
$(***DTACQ-NAME***)V:RCURSOR_ms

$(***DTACQ-NAME***):CARD0:SAMPLINGRATE

Motor:

This tab allows to show the configuration of the motor: Velocity, Homing Velocity, Acc. Time => Read only.

The motor could be also control manually (Postion CONS). Before to move the motor, the brake have to be disabled. The HOME button allows to send the motor to HOME position.



$(***PREFIX_MTR***):$(***PMAC_MTR***).VELO

$(***PREFIX_MTR***):$(***PMAC_MTR***).HVEL

$(***PREFIX_MTR***):$(***PMAC_MTR***).ACCL

$(***PREFIX_MTR***):$(***PMAC_MTR***).VAL

$(**PREFIX_MTR**):$(***PMAC_MTR***).VAL

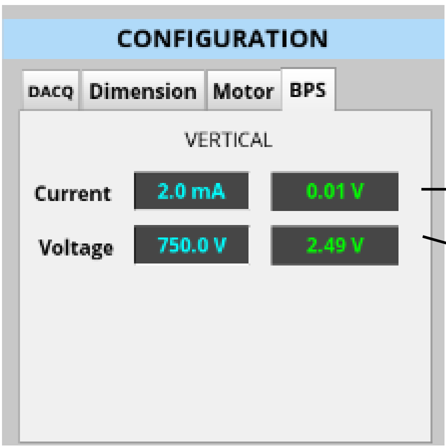
$(**PREFIX**):VER-BRAKE_SET

$(**PREFIX_MTR**):$(***PMAC_MTR***).SPMG

$(**PREFIX_MTR**):$(***PMAC_MTR***)-ACTMTR.PROC

Bias Power Supply (BPS):

This tab allows to show to configure the Bias Power Supply



$(***PREFIX***)$(**TYPE**):BPS1-CurS

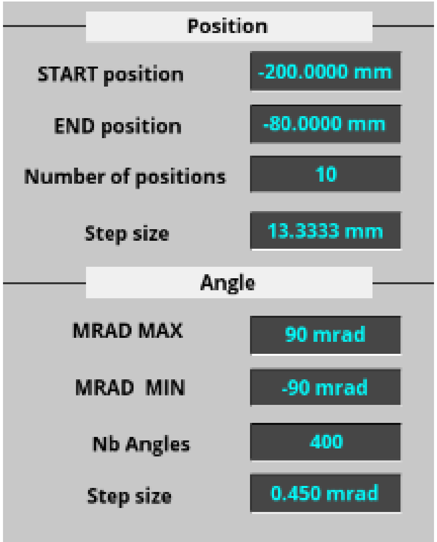
$(***PREFIX***)$(**TYPE**):BPS1-CurR

$(***PREFIX***)$(**TYPE**):BPS1-VolS

$(***PREFIX***)$(**TYPE**):BPS1-VolR

Scanning configuration:

This tab allows to configure scanning operation

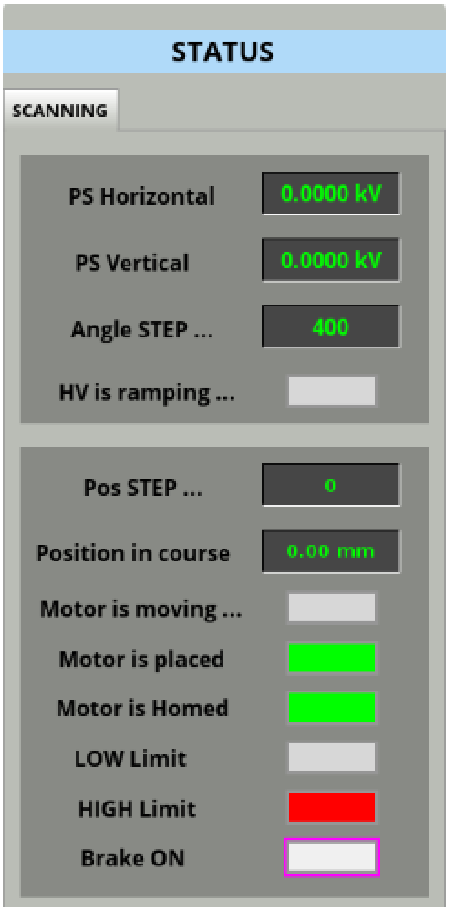


$(***PREFIX_MTR***):$(***NAME_MTR***).P4SP
$(***PREFIX_MTR***):$(***NAME_MTR***). P4EP
$(***PREFIX_MTR***):$(***NAME_MTR***).NPTS
$(***PREFIX_MTR***):$(***NAME_MTR***).P4SI

$(***PREFIX_SCAN***):$(***PROC***)-AngleMax-SP
$(***PREFIX_SCAN***):$(***PROC***)-AngleMin-SP
$(***PREFIX_SCAN***):$(***NAME_PS***).NPTS
$(***PREFIX_SCAN***):$(***PROC***)-StepAngle

Status

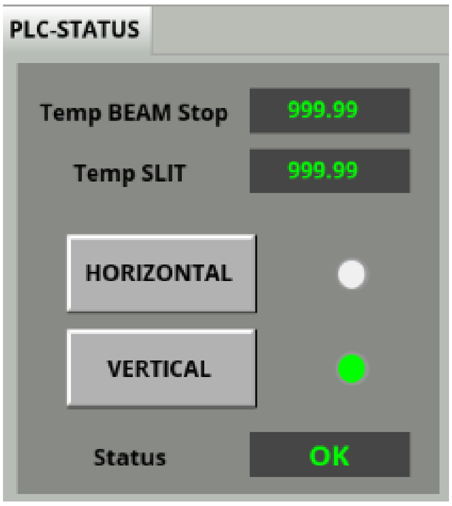
This part allows to watch information about the scanning in running mode



$(***PREFIX_PS***)hor-SP
$(***PREFIX_PS***)ver-SP
$(***PREFIX_SCAN***):$(***NAME_PS***)-NORD_POS
$(***PREFIX_SCAN***):$(***NAME_PS***).BUSY

$(***PREFIX_SCAN):***$(***NAME_MTR***)-NORD
$(***PREFIX_MTR***):$(***PMAC_MTR***).RBV
$(***PREFIX_MTR***):$(***PMAC_MTR***).MOVN
$(***PREFIX_MTR***):$(***PMAC_MTR***).MSTA (Bit 1)
$(***PREFIX_MTR***):$(***PMAC_MTR***).MSTA (Bit 14)
$(***PREFIX_MTR***):$(***PMAC_MTR***).MSTA (Bit 13)
$(***PREFIX_MTR***):$(***PMAC_MTR***).MSTA (Bit 2)
$(***PREFIX)***:VER-BRAKE_GET

This tab allows to watch the status of the PLC dedicated to both EMUs:



$(**PLC-INTK**)-VBS:TEMP

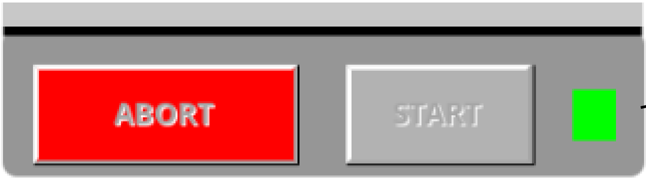
$(**PLC-INTK**)-VS:TEMP

$(**PLC-INTK**)-HMI:VH_SELECT

$(**PLC-INTK**)-HMI:FM1

Start/Stop

These buttons allows to start or stop the scanning. When the scanning is stopped, the EMU goes home.



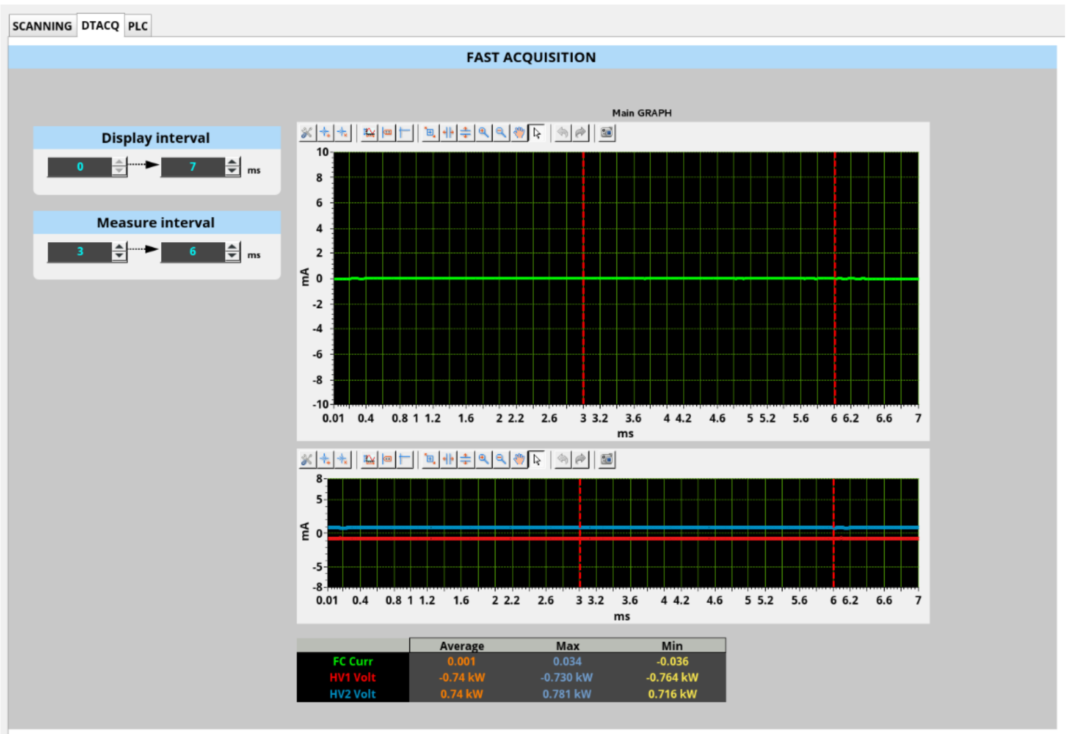
$(***PREFIX***):$(***TYPE***):MTR-ABORT.PROC

$(***PREFIX***):$(***TYPE***):MTR.EXSC

$(***PREFIX***):$(***TYPE***):MTR.BUSY

Scanning Fast acquisition (DTACQ)

This tab allows to configure and watch acquisition of all acquisitions.



<u>Display interval:</u> configure the length of the acquisition

<u>Measure interval</u>: configure the length for the average

The first graph allows to watch Current on the faraday Cup. And The next allows to watch Voltage on both plates.

Name of PV	Average	Max	Min
FC Current	*$(**DTAQ-NAME**)V-$(**DEVICE_FC**):CurrR:AVG*	*$(**DTAQ-NAME**)V-$(**DEVICE_FC**):CurrR:MAX*	*$(**DTAQ-NAME**)V-$(**DEVICE_FC**):CurrR:MIN*
HV1 Voltage	*$(**DTAQ-NAME**)V-$(**DEVICE_HV1**):CurrR:AVG*	*$(**DTAQ-NAME**)V-$(**DEVICE_HV1**):CurrR:Max*	*$(**DTAQ-NAME**)V-$(**DEVICE_HV1**):CurrR:Min*
HV2 Voltage	*$(**DTAQ-NAME**)V-$(**DEVICE_HV2**):CurrR:AVG*	*$(**DTAQ-NAME**)V-$(**DEVICE_HV2**):CurrR:Max*	*$(**DTAQ-NAME**)V-$(**DEVICE_HV2**):CurrR:Min*

4 – Description of the operating mode

The operator has to provide:

- All the information about the beam condition
- The sampling frequency
- The number of positions
- The number of angles for each motor position
- The limit voltage for both Trek amplifiers

For each motor position, there is a number of angles. Each angle represents a combination of a setting voltage for both Trek amplifiers. At each angle, three waveforms are saved and one analog input.

- The intensity measurement of the Faraday cup (waveform)
- The output voltage measurement of the trek amplifier 1 (waveform)
- The output voltage measurement of the trek amplifier 2 (waveform)
- The real motor position (analog input)

To reduce the space storage, the output voltage measurement of both Trek amplifiers is archived after some calculations integrating the mechanical structure of the EMU.

During the measurement, cursors for averaging are fixed.

After each position, data sampled and calculated are displayed.

At the end of the measurement, an emittance is calculated.