Isrc&LEBT Control System Verification Plan

	Name	Affiliation
Authors	William Ledda	ICS – Control System Integrator
Reviewers	Karl Vestin Maria Romedahl Saeed Haghtalab	Group Leader HW & I, ICS Technical Coordinator, ICS Control Systems Engineer, ICS
Approver	Timo Korhonen	Chief Engineer, ICS

TABLE OF CONTENT

1. INTRODUCTION

1.1 Purpose of the document

This verification plan documents the strategy that will be used to verify and ensure that the Control System of the Ion Source (ISrc) and the Low Energy Beam Transport (LEBT) of the ESS linear accelerator complex works according to the specifications.

The verification for the PLCs is left out from this document since require a dedicated integration plan that will be provided as integration.

2. SYSTEM CHARACTERISTICS

2.1 System purpose

The ESS machine, is an accelerator driven neutron spallation source. The linear accelerator, or LINAC, is thus a critical component.

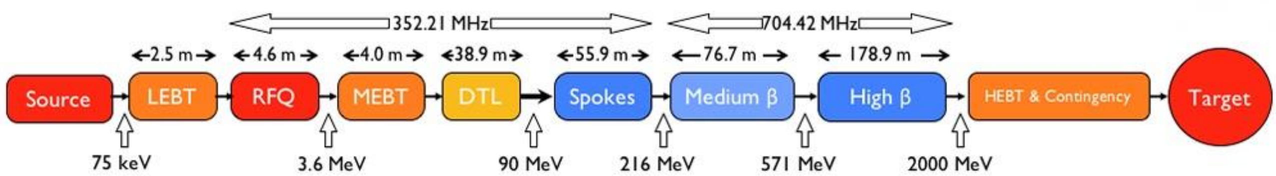


Figure 1 - Schematic picture of the different parts of the ESS machine.

The ESS machine consists of four major parts:

- Source and LEBT ("the Injection unit")
- Accelerator
- Target
- Instruments

On Figure 1, from left to right: The Source (= ion source =ISrc) and the LEBT. The actual accelerator (LINAC) starts with the RFQ and ends just before the Target. The Target. The Instruments are not shown in this picture.

The Isrc is of the type Electron Cyclotron Resonance (ECR), and its task is to produce a plasma, i.e. ions in a vacuum chamber. The ions produced in the ESS machine are H^+, which is the same as protons (p^+). The protons are repelled from the ion source unit, as pulses shorter than 6 ms, into the LEBT unit. In the LEBT the proton pulses get more focused and are chopped to maximum 2.86 ms pulse length. The pulsed proton beam is then repelled from the LEBT and injected into the next unit, the Radio Frequency Quadrupole (RFQ) which is the start of the actual accelerator. In the accelerator the proton beam is steered towards the target station. When the protons hit the target they will emit neutrons that are collected by the neutron instruments to perform different scientific experiments.

2.2 System overview

The Control System for the ISrc & LEBT is intended as the HW and SW infrastructure that allows to operate and monitor and collect data from the machine. It consists of many different type of devices (e.g. Power Supplies, PLCs, field bus devices) communicating each other using standard industrial control protocol (e.g. Profinet, Modbus, EtherCAT).

The Control system architecture of the Isrc is shown in Figure 2, and the one for the LEBT is show in Figure 3.

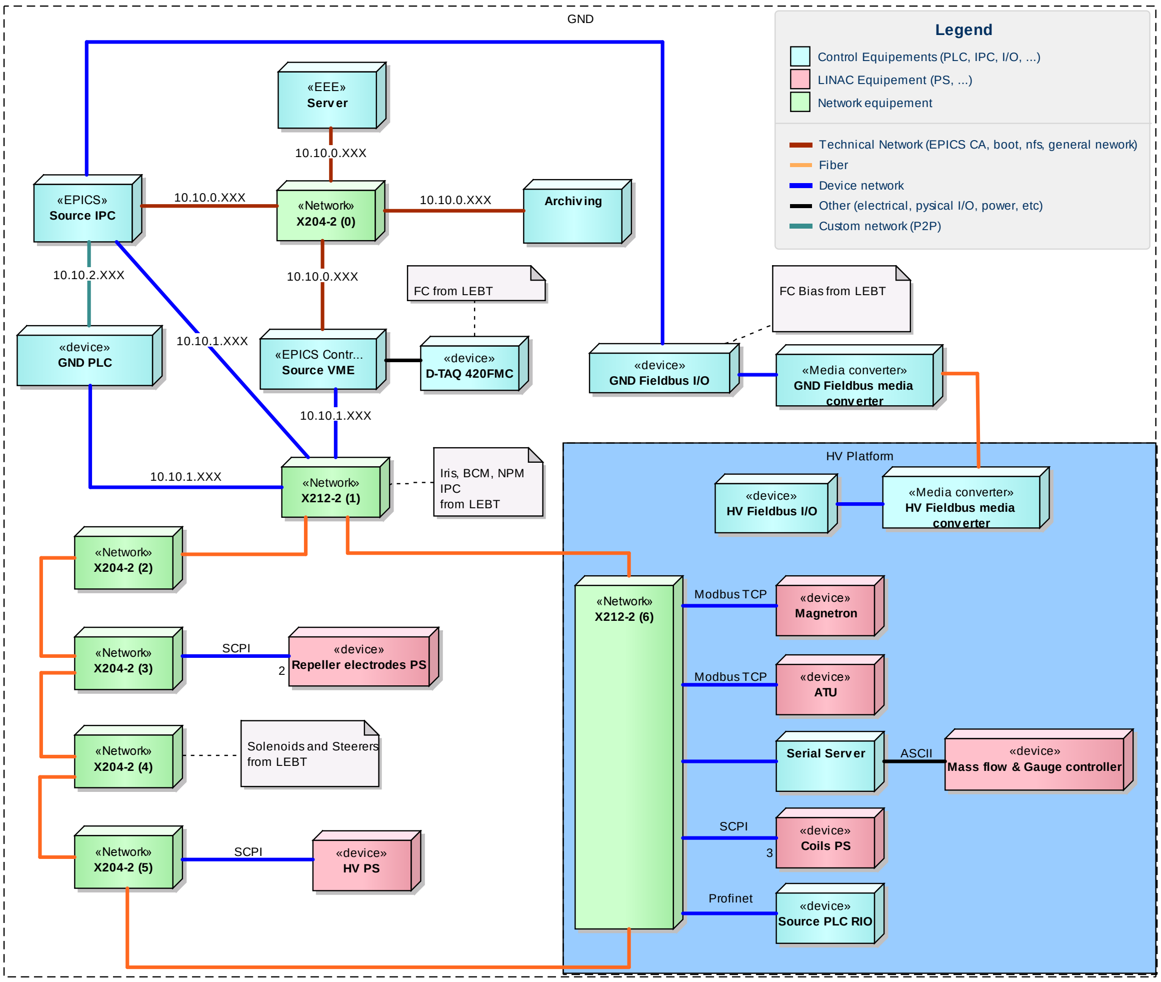


Figure 2 - ISrc Control System Architecture

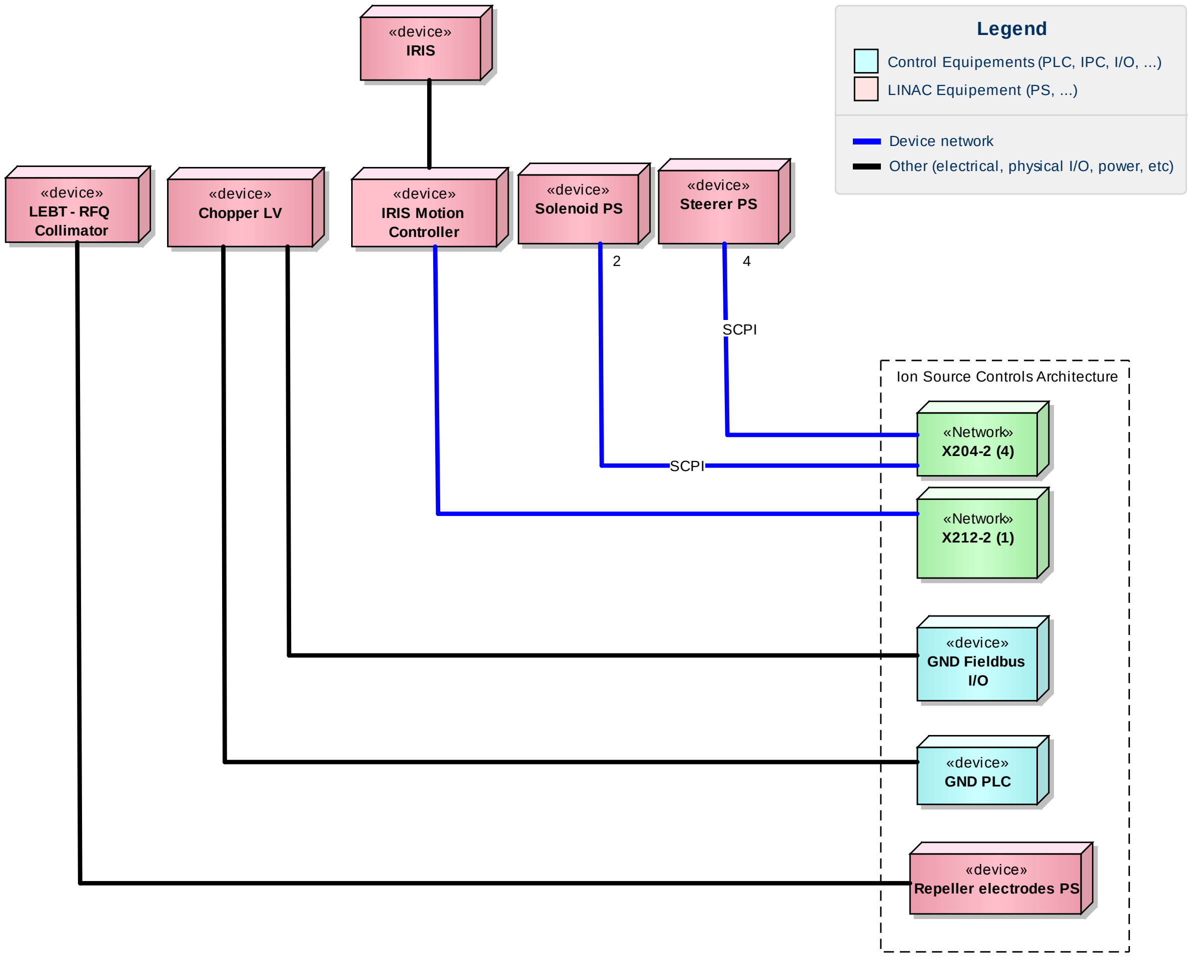


Figure 3 - LEBT Control System Architecture

Aside the LEBT elements shown in Figure 3 (i.e. Solenoids, Steerers, Iris Chopper and LEBT-RFQ Collimator), there other equipment that are omitted for simplicity and that re used for beam diagnostic purposes:

- Faraday cup (FC);
- Emittance Measurement Unit (EMU);
- Beam Current Monitor (BCM);
- Non-Invasive Profile Monitor (NPM);
- Doppler Shift Measurement (DPL or DSM);

There are many different interface and protocols involved and the communication between all devices and control application is granted by the EPICS (Experimental Physics and Industrial Control System) framework.

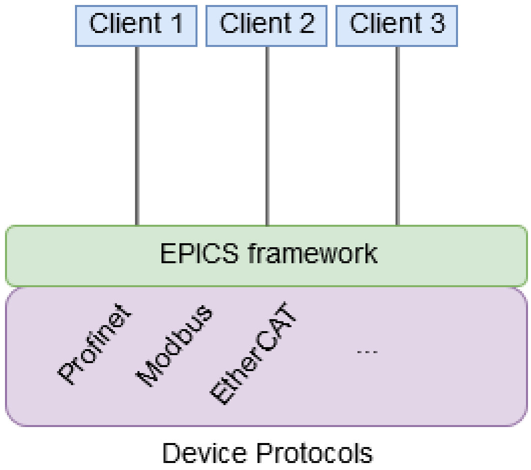


Figure 4 - EPICS Framework

This verification plan covers all devices for the source and the LEBT (i.e. Solenoids, Steerers, Chopper and Iris). In addition it covers the EMU diagnostic device. Faraday Cup, Beam Current Monitor, NPM and DPL are intentionally left out since they are under review and they are in the scope of Beam Diagnostic Division.

3. VERIFICATION IDENTIFICATION

This and the following sections provide details on how the verification is accomplished in terms of HW, SW and human resources.

Many of the tests described in this System Verification Plan (SVP) are executed using the automatic test tool WeTest [2] to run a well-defined set of predefined scenarios ([3] and [4]). Refer to Annex A to install and configure WeTest properly before to run the tests included in this SVP.

A System Verification Report (SVR) shall be provided as result of the execution of the tests included in this SVP. Testing anomalies need to be recorded in the SVR and issues observed should be registered in the ESS issue tracking system ([5]).

The SVP consists in a set of Test Cases (TC). A Test Case is a logical grouping of methods for verification of functions, constraints and performance criteria (if applicable) that has to be verified together. Each test case in the following sections contains the following information:

- A description name and a reference number.
- When applicable, a complete list of the requirements to be verified. For ease of tracing of requirements into the Verification Plan and other documents, the requirements are called in by their Id.
- Any data to be recorded or noted during the test, such as expected results of a test step. Other data, such as a recording of a digital message sent to an external system, may be required to verify the performance of the system.
- A statement of the pass/fail criteria.

- A description of the test configuration. That is a list of the hardware and software items needed for the test and how they should be connected. Often, the same configuration is used for several tests.
- A list of any other important assumptions and constraints necessary for conduct of the test case.

3.1 Test case ISRC-001 – Test suite installation

The purpose of this test is to install the suite of tests required to run the acceptance tests for the source.

3.1.1 Support Environment

A ESS Development Machine connected to the Technical Network (and then to the Control System installed in FEB-090) is required to run the test suite for the Isrc. The Development machine can be both a virtual or a physical one.

Tests can be conducted both in FEB-090 or in a remote location (e.g. in the Local Control Room) connected to the same Technical Network (TN) of the Control System. The WeTest tool must be installed and configured as described in Annex A.

The operator that runs the tests need to have basic knowledge of Linux and Shell bash.

The operator needs to have a valid ESS Git account to access to the test suite repository.

3.1.2 Configuration

This tests requires a proper ESS Development Machine installed.

3.1.3 Setup

Not applicable

3.1.4 Procedure

Steps

1. Open a console in the ESS Development machine.
2. Clone the Isrc test suit from the ESS Git repository:

```
$ git clone
https://<user>@bitbucket.org/europeanspallationsource/source-
acceptance-tests.git

Cloning into 'source-acceptance-tests'...
remote: Counting objects: 132, done.
remote: Compressing objects: 100% (127/127), done.
remote: Total 132 (delta 72), reused 0 (delta 0)
Receiving objects: 100% (132/132), 21.63 KiB | 0 bytes/s, done.
Resolving deltas: 100% (72/72), done.
```

3. Verify that the test suite is cloned into a local workspace

```
$ ls -l source-acceptance-tests/
```

```
total 40
-rw-r--r-- 1 operator operator  976 Apr 16 15:53 iptest.py
-rw-r--r-- 1 operator operator 1989 Apr 16 15:53 lebt-at-plc-status.yml
-rw-r--r-- 1 operator operator 6581 Apr 16 15:53 source-at-0-reference.yml
-rw-r--r-- 1 operator operator 3523 Apr 16 15:53 source-at-1-nominal-
pulsed.yml
-rw-r--r-- 1 operator operator 3217 Apr 16 15:53 source-at-2-nominal-
continuous.yml
-rw-r--r-- 1 operator operator  507 Apr 16 15:53 source-at-continuous.yml
-rw-r--r-- 1 operator operator 3085 Apr 16 15:53 source-at-plc-status.yml
-rw-r--r-- 1 operator operator  487 Apr 16 15:53 source-at-pulsed.yml
-rw-r--r-- 1 operator operator 1570 Apr 16 15:53 source-at-switch-off.yml
```

Expected results
Isrc test suite is cloned from ESS git repository into the local system.

3.2 Test case ISRC-002 – Run Source in pulsed mode

The purpose of this test is to configure and run the Ion Source in “pulsed” mode.

This test scenario runs the following test cases contained in the Isrc test suite (ISRC-001) in sequence:

1. Set up of the source to a known states:
   a. source-at-0-reference.yml
2. Verify the status of the PLC:
   a. source-at-plc-status.yml
   b. lebt-at-plc-status.yml
3. Run the source in pulse mode:
   a. source-at-1-nominal-pulsed.yml
4. Verify the status of the PLC:
   a. source-at-plc-status.yml
   b. lebt-at-plc-status.yml
5. Switch of the source:
   a. source-at-switch-off.yml

3.2.1 Support Environment

The control system for the Isrc is installed in FEB-090 in ISRC-010Row:CnPw-U004 and ISRC-010Row:CnPw-U003 racks. Some of the equipment under tests are installed inside the HV cage platform in the Tunnel connected to the equipment in FEB-090 through a Fiber Optic Link.

The operator that runs the tests need to have basic knowledge of Linux and Shell bash.

3.2.2 Configuration

The following devices of the ISrc will be verified by the test case:

Table 1 - Device Under Test

Device	Location	ESS Name

Magnetron Power Supply	Tunnel, HV Cage	ISrc-010:ISS-Magtr
Coil Power Supply (3x)	Tunnel, HV Cage	ISrc-010:PwrC-CoilPS-01 ISrc-010:PwrC-CoilPS-02 ISrc-010:PwrC-CoilPS-01
High Voltage Power Supply	FEB-090 ISRC-010Row:CnPw-U001	ISrc-010:ISS-HVPS
Repeller Electrode's power supply (Extraction system)	FEB-090 ISRC-010Row:CnPw-U003	ISrc-010:PwrC-RepPS-01

The control controls equipment used to verify the source are listed in Table 2.

Table 2 - Control Equipment

Control Equipment	Location	ESS Name
Source VME	FEB-090, ISRC-010Row:CnPw-U004	ISrc-010:Ctrl-CPU-01
Source-LEBT IPC	FEB-090, ISRC-010Row:CnPw-U004	ISrc-010:Ctrl-IPC-01
Source-LEBT PLC	FEB-090, ISRC-010Row:CnPw-U004	ISrc-010:Ctrl-PLC-01
Source PLC Remote I/O	Tunnel, HV Cage	ISrc-010:Ctrl-RIO- HVRIO1
Source Field I/O (EtherCAT)	Tunnel, HV Cage	ISrc-010:Ctrl-ECATIO-HV

The network infrastructure required to run the tests is listed in Table 3.

Table 3 - Network infrastructure

Network	Description
Technical Network (TN)	EPICS Channel Access network for G01 FEB (ChannelAccess-FEB Vlan)
Device Network (DN)	IScr & LEBT system devices (ISCR-LEBT-Device Vlan)

Even if non part of this SVP, the status of the Interlock PLC is verified as part of the execution of the test case for safety reasons.

The test case shall be executed from the ESS Development Machine where the test suite has been installed (ISRC-001).

3.2.3 Setup

1. All equipment under test (see Table 1) must be properly turned on

2. The Source VME (Table 2) must be turned on and connected to the TN and the DN (Table 3).
3. The Source-LEBT IPC (Table 2) must be turned on and connected to the TN and the DN (Table 3).
4. Test scenario installed as described in ISRC-001.
5. A console running Control System Studio (CSS) with the "CSS-ISrc-LEBT" workspace (Annex B) connected to the TN.
6. Optionally record the screen during the execution of the test (i.e. Ctrl + Alt + Shift + R on CentOS ESS based Development machines)

3.2.4 Procedure

Steps

1. Move to the local directory containing the ISrc test suite (ISRC-001):

```
$ cd source-acceptance-tests
```

2. Run the WeTest environment from the console as follow:

```
$ source activate wetest
```

3. Run the scenario associated to the test case as follow

```
$ wetest -S source-at-pulsed.yml -o isrc_pulsed_$(date +%F_%H%M%S).pdf
```

4. Verify on CSS how the test proceed configuring and activating the different equipment under tests.
5. Verify on the console that the tests are proceeding and that ne error is reported.

```
...........
```

6. Wait until the test is completed (~ 15 minutes).
7. Open the report generated by WeTest and go through the list of tests executed.
8. Verify that all tests are passed.
9. Collect the report generated by WeTest and attach it to the SVR.
10. Deactivate WeTest environment as follow:

```
$ source deactivate wetest
```

Expected results

Tests registered in the report provided by WeTest are passed.

3.3 Test case ISRC-003 – Run Source in pulsed mode

The purpose of this test is to configure and run the Ion Source in "continuous" mode.

This test scenario runs the following test cases contained in the Isrc test suite (ISRC-001) in sequence

1. Set up of the source to a known states:

    a. source-at-0-reference.yml
2. Verify the status of the PLC:
    a. source-at-plc-status.yml
    b. lebt-at-plc-status.yml
3. Run the source in continuous mode
    a. source-at-2-nominal-continuous.yml
4. Verify the status of the PLC:
    a. source-at-plc-status.yml
    b. lebt-at-plc-status.yml
5. Switch of the source
    a. source-at-switch-off.yml

3.3.1 Support Environment

See 3.1.1.

3.3.2 Configuration

See 3.2.2.

3.3.3 Setup

See to 3.2.3

3.3.4 Procedure

Steps

1. Move to the local directory containing the ISrc test suite (ISRC-001):

```
$ cd source-acceptance-tests
```

2. Run the WeTest environment from the console as follow

```
$ source activate wetest
```

3. Run the scenario associated to the test case as follow

```
$ wetest -S source-at-continuous.yml -o isrc_continuous_$(date +%F_%H%M%S).pdf
```

4. Verify on the console that the tests are proceeding and that ne error is reported.

```
...........
```

5. Verify on CSS how the test proceed configuring and activating the different equipment under tests.
6. Wait until the test is completed (~ 15 minutes).
7. Open the report generated by WeTest and go through the list of tests executed.
8. Verify that all tests are passed.
9. Collect the report generated by WeTest and attach it to the SVR.
10. Deactivate WeTest environment as follow:

```
$ source deactivate wetest
```

Expected results
Tests registered in the report provided by WeTest are passed.

3.4 Test case LEBT-001 – Test Installation

The purpose of this test is to install the suite of tests required to run the acceptance tests for the LEBT.

3.4.1 Support Environment

A ESS Development Machine connected to the Technical Network (and then to the Control System installed in FEB-090) is required to run the test suite for the LEBT. The Development machine can be both a virtual or a physical one.

Tests can be conducted both in FEB-090 or in a remote location (e.g. in the Local Control Room) connected to the same Technical Network (TN) of the Control System. The WeTest tool must be installed and configured as described in Annex A.

The operator that runs the tests need to have basic knowledge of Linux and Shell bash.

The operator needs to have a valid ESS Git account to access to the test suite repository.

3.4.2 Configuration

This tests requires a proper ESS Development Machine installed.

3.4.3 Setup

Not applicable

3.4.4 Procedure

Procedure

1. Open a console in the ESS Development machine.
2. Clone the LEBT test suit from the ESS Git repository:

```
$ git clone
https://<user>@bitbucket.org/europeanspallationsource/lebt-acceptance-
tests.git

Cloning into 'lebt-acceptance-tests'...
remote: Counting objects: 49, done.
remote: Compressing objects: 100% (47/47), done.
remote: Total 49 (delta 26), reused 0 (delta 0)
Unpacking objects: 100% (49/49), done.
```

3. Verify that the test suite is cloned into a local workspace

```
$ ls -l lebt-acceptance-tests/
total 64
```

```
-rw-r--r-- 1 operator operator  1013 Apr 16 17:14 diags-at-nominal.yml
-rw-r--r-- 1 operator operator   976 Apr 16 17:14 iptest.py
-rw-r--r-- 1 operator operator 13962 Apr 16 17:14 lebt-at-1-nominal.yml
-rw-r--r-- 1 operator operator  1522 Apr 16 17:14 lebt-at-plc-avoid-
collision.yml
-rw-r--r-- 1 operator operator  2068 Apr 16 17:14 lebt-at-plc-status.yml
-rw-r--r-- 1 operator operator   695 Apr 16 17:14 lebt-with-source-pulsed.yml
-rw-r--r-- 1 operator operator  6581 Apr 16 17:14 source-at-0-reference.yml
-rw-r--r-- 1 operator operator  4311 Apr 16 17:14 source-at-1-nominal-
pulsed.yml
-rw-r--r-- 1 operator operator  3085 Apr 16 17:14 source-at-plc-status.yml
-rw-r--r-- 1 operator operator   487 Apr 16 17:14 source-at-pulsed.yml
-rw-r--r-- 1 operator operator  1570 Apr 16 17:14 source-at-switch-off.yml
```

Expected results
LEBT test suite is cloned from ESS git repository into the local system.

3.5 Test case LEBT-002 – LEBT Automatic tests

The purpose of this test is to configure and run the LEBT Control System. This is achieved running the automated test scenario **lebt-with-source-pulsed.yml** contained in [4]. It consists in the of the following test cases executed in sequence:

1. Set up of the source to a known states:
    a. source-at-0-reference.yml
2. Verify the status of the PLC interlock for both source and LEBT:
    a. source-at-plc-status.yml
    b. lebt-at-plc-status.yml
3. Verify the procedure to avoid collision between the Faraday cup and the EMUs
    a. lebt-at-plc-avoid-collision.yml
4. Run the source in pulse mode:
    a. source-at-1-nominal-pulsed.yml
5. Verify the status of the PLC interlock for both source and LEBT, after running the source:
    a. source-at-plc-status.yml
    b. lebt-at-plc-status.yml
6. Verify LEBT devices: Steerer/ Solenoid / Vacuum / Chopper
    a. lebt-at-1-nominal.yml
7. Verify the status of the PLC interlock for both source and LEBT, after running the test of the LEBT:
    a. source-at-plc-status.yml
    b. lebt-at-plc-status.yml

The test for the LEBT is executed running the source in "pulsed" mode (i.e. **source-at-1-nominal-pulsed.yml** test case) as for ISRC-002 test.

3.5.1 Support Environment

See 3.1.1.

3.5.2 Configuration

The following devices of the LEBT will be verified by the test case:

Table 4 – LEBT Device Under Test

Device	Location	ESS Naming
Solenoid Power Supply (x2)	FEB-090, ISRC-010Row:CnPw-U003	LEBT-010:PwrC-SolPS-01 LEBT-010:PwrC-SolPS-02
Steerer Power Supply (x4)	FEB-090, ISRC-010Row:CnPw-U003	LEBT-010:PwrC-PSCH-01 LEBT-010:PwrC-PSCH-02 LEBT-010:PwrC-PSCV-01 LEBT-010:PwrC-PSCV-02
Chopper (LVM)	TBD	LEBT-010:BMD-Chop

The control controls equipment used to verify the source are listed in Table 2.

Table 5 – LEBT Control Equipment

Control Equipment	Location	ESS Naming
Source VME	FEB-090, ISRC-010Row:CnPw-U004	ISrc-010:Ctrl-CPU-01
Source-LEBT IPC	FEB-090, ISRC-010Row:CnPw-U004	ISrc-010:Ctrl-IPC-01
Source-LEBT PLC	FEB-090, ISRC-010Row:CnPw-U004	ISrc-010:Ctrl-PLC-01
LEBT Field I/O (EtherCAT)	FEB-090, ISRC-010Row:CnPw-U004	LEBT-010:Ctrl-ECATIO-GND

The network infrastructure required to run the tests is listed in Table 3.

Even if non part of this SVP, the status of the Interlock PLC is verified as part of the execution of the test case for safety reasons.

The test case shall be executed from the ESS Development Machine where the test suite has been installed (LEBT-001).

3.5.3 Setup

1. All equipment under test (see Table 4) must be properly turned on
2. The Source VME (Table 5) must be turned on and connected to the TN and the DN (Table 3).
3. The Source-LEBT IPC (Table 5) must be turned on and connected to the TN and the DN (Table 3).
4. Test scenario installed as described in LEBT-001.
5. A console running Control System Studio (CSS) with the "CSS-Diagnostic" workspace (Annex B) connected to the TN (Table 3).
6. Optionally record the screen during the execution of the test (i.e. Ctrl + Alt + Shift + R on CentOS ESS based Development machines)

3.5.4 Procedure

Procedure

1. Move to the local directory containing the LEBT scenarios
`$ cd source-acceptance-tests`
2. Run the WeTest environment from the console as follow:
`$ source activate wetest`
3. Run the scenario associated to the test case as follow:
`$ wetest –S lebt-with-source-pulsed.yml -o lebt_$(date +%F_%H%M%S).pdf`
4. Verify on CSS how the test proceed configuring and activating the different equipment under tests.
5. Verify on the console that the tests are proceeding and that ne error is reported.
`. . . . . . . . . . .`
6. Wait until the test is completed (~ 15 minutes).
7. Open the report generated by WeTest and go through the list of tests executed.
8. Verify that all tests are passed.
9. Collect the report generated by WeTest and attach it to the SVR.
10. Deactivate WeTest environment as follow:
`$ source deactivate wetest`

Expected results
Tests registered in the report provided by WeTest are passed.

3.6 Test case LEBT-003 – LEBT Iris Verification

The test for the LEBT Iris require a manual procedure described in the following sections.

The verification of the Iris involves the following topic:

1. Set the aperture of the Iris
2. Move the center of the Iris

3.6.1 Support Environment

A ESS Development Machine connected to the Technical Network (and then to the Control System installed in FEB-090) is required to run the test suite for the Isrc. The Development machine can be both a virtual or a physical one.

Tests can be conducted both in FEB-090 or in a remote location (e.g. in the Local Control Room) connected to the same Technical Network (TN) of the Control System.

3.6.2 Configuration

The controls equipment under tests for the Iris beam line elements are listed in Table 6

Table 6 – Iris Control Equipment

Control Equipment	Location	ESS naming
Source-LEBT IPC	FEB-090, ISRC-010Row:CnPw-U004	ISrc-010:Ctrl-IPC-01
Iris Motion Control Unit	FEB-090, ISRC-010Row:CnPw-U004	LEBT-010:MC-MCU-001

The network infrastructure required to run the tests is listed in Table 3.

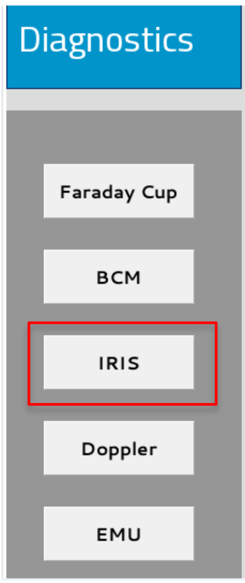
3.6.3 Setup

1. All equipment under test (Table 6) must be properly turned on
2. The Source-LEBT IPC (Table 6) must be turned on and connected to the TN and the DN (Table 3).
3. A console running Control System Studio (CSS) with the "CSS-Diagnostic" workspace (Annex B) connected to the TN (Table 3).
4. Optionally record the screen during the execution of the test (i.e. Ctrl + Alt + Shift + R on CentOS ESS based Development machines)

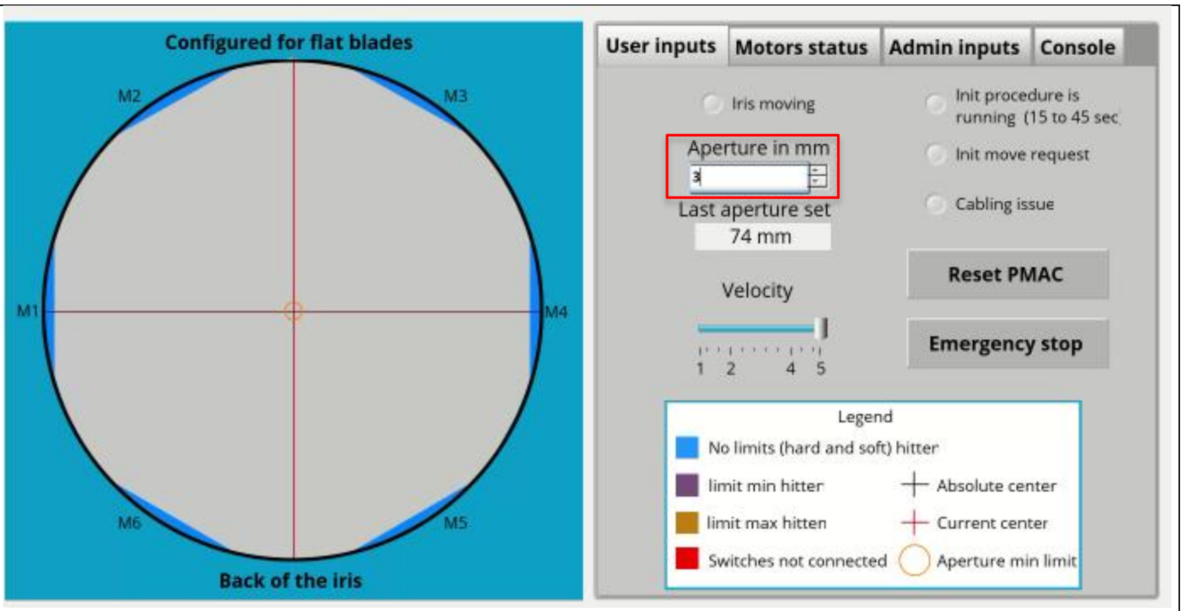
3.6.4 Procedure

Procedure

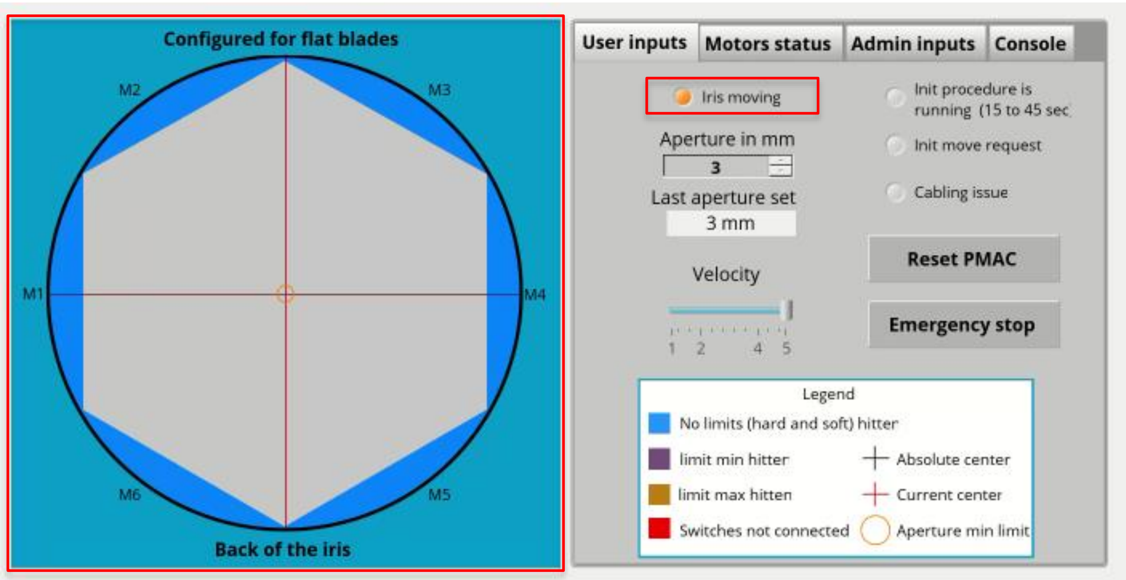
1. On CSS, click the "Iris" button to open the OPI for the Iris.



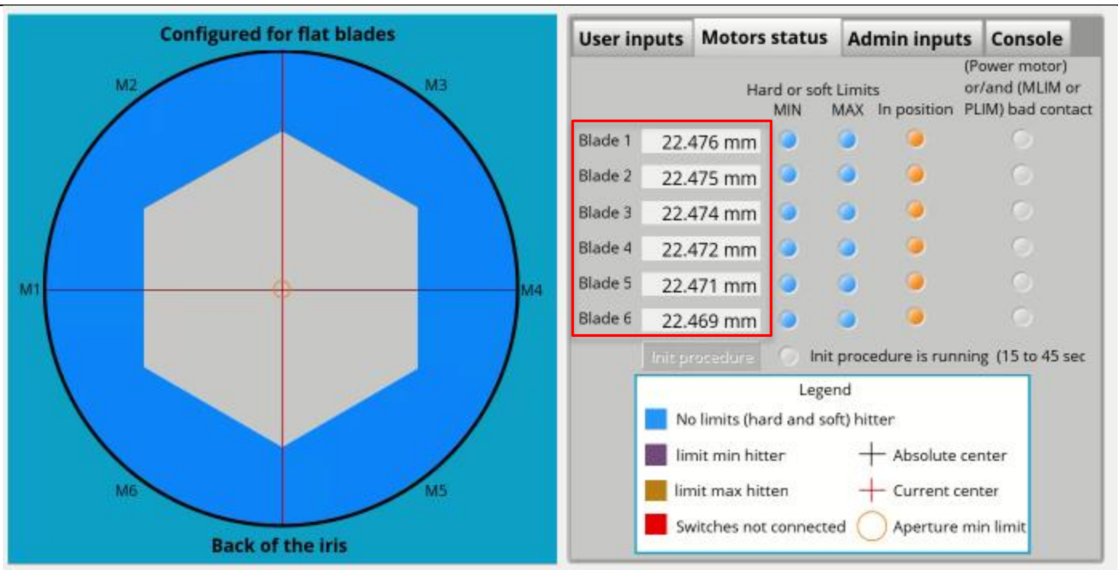
2. From the "User inputs" tab of the OPI, set the "Aperture field" (e.g. 3 mm) and press Enter



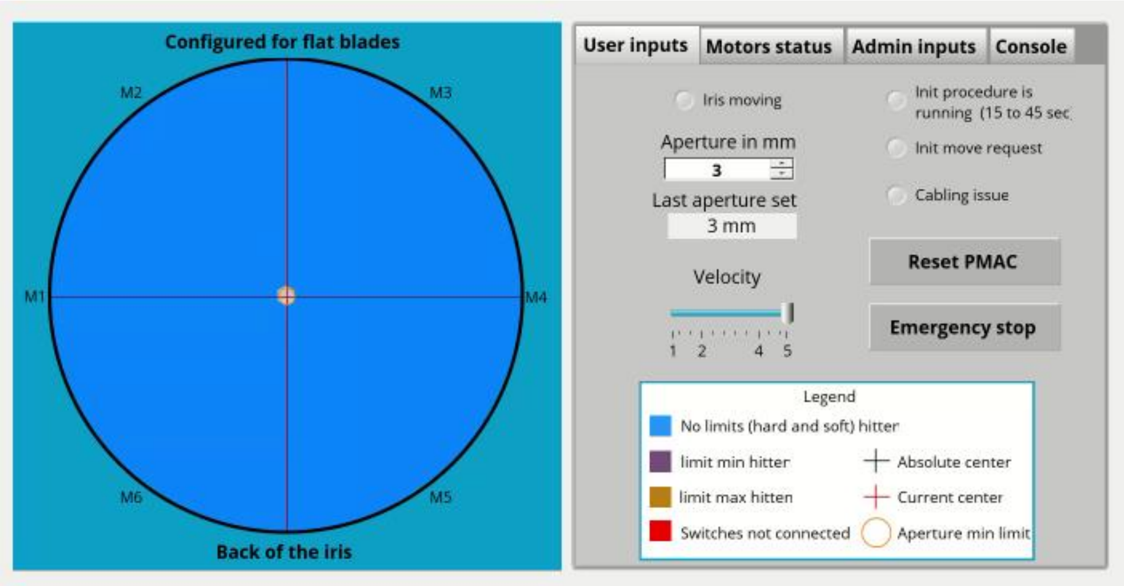
3. Verify that the led "Iris moving" start to blink and that picture on the left is updated with the movement of the blades.



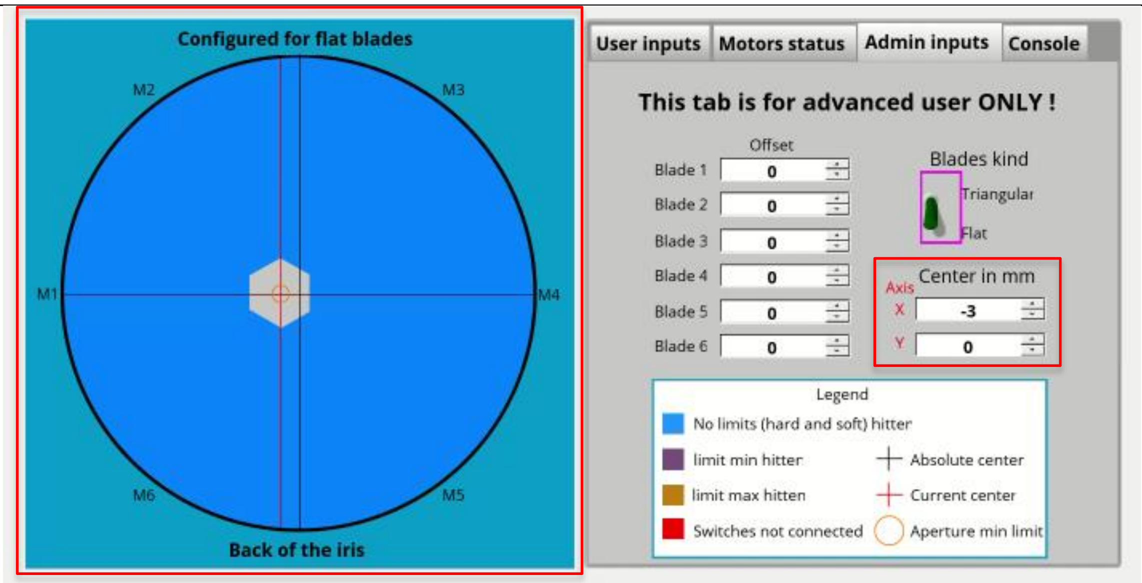
4. Verify from the "Motor Status" Tab that blades are moving



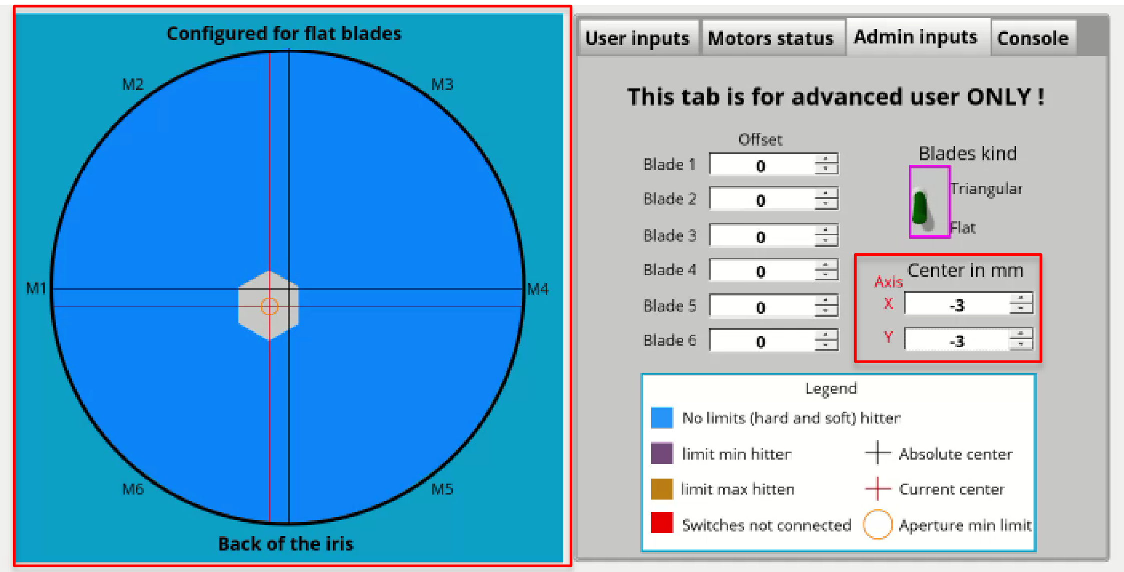
5. Verify that when the "Iris moving" turns off when blades reach the desired position.



6. Select the "Admin Inputs" tab to change the center of the Iris
7. Set the "X" field to -3 mm and verify that the center is moved



8. Set the "Y" to -3 and verify that the center is moved



9. Repeat steps 8 and 9 Setting the center to "X" and "Y" field to 3
10. Verify that the center is moved to the new position.
11. Set the center to 0 in both "X" and "Y" fields and verify that the center is moved to the original position

Expected results

1. It is possible to control the movement of the Iris's blades from the OPI.
2. It is possible to set the aperture of the Iris to the specified value.
3. It is possible to move the center of the Iris to a specified value.

3.7 Test case EMU-001 – EMU Selection

The scope of this test is to verify that it is possible to select one of the EMU head (Horizontal or Vertical).

3.7.1 Support Environment

A ESS Development Machine connected to the Technical Network (and then to the Control System installed in FEB-090) is required to run the test suite for the Isrc. The Development machine can be both a virtual or a physical one.

Tests can be conducted both in FEB-090 or in a remote location (e.g. in the Local Control Room) connected to the same Technical Network (TN) of the Control System.

3.7.2 Configuration

The following devices of the EMU will be verified by the test case:

Table 7 – EMU Device Under Test

Device	Location	ESS Naming
EMU Plates Power Supply (x2)	FEB-090, Rack TBD	FEB-030Row:PBI-HV-004 FEB-030Row:PBI-HV-005
EMU Bias Power Supply	FEB-090, Rack TBD	FEB-030Row:PBI-HV-002
EMU HV switch	FEB-090, Rack TBD	FEB-030Row:PBI-HV-003

The control controls equipment under tests for the EMU are listed in Table 8

Table 8 – EMU Control Equipment

Control Equipment	Location
EMU VME	FEB-090, Rack TBD
EMU IPC	FEB-090, Rack TBD
EMU PLC	FEB-090, Rack TBD
EMU Field I/O (EtherCAT)	FEB-090, Rack TBD
EMU Motion Control Unit	FEB-090, Rack TBD

The network infrastructure required to run the tests is listed in Table 3.

3.7.3 Setup

1. All equipment under test (Table 7) must be properly turned on

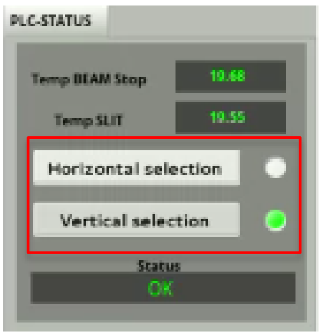
2. Both the EMU VME and IPC (Table 8) must be turned on and connected to the TN and the DN (Table 3).
3. A console running Control System Studio (CSS) with the "CSS-Diagnostic" workspace (Annex B) connected to the TN (Table 3).
4. Optionally record the screen during the execution of the test (i.e. Ctrl + Alt + Shift + R on CentOS ESS based Development machines)

3.7.4 Procedure

Procedure

1. On CSS, click the "EMU" button to open the OPI for the EMU.

2. Click on "Horizontal Selection" or "Vertical Selection" buttons to select the horizontal or the vertical head respectively.



3. Verify that only the led corresponding to the head selected turns on (e.g. "Vertical selection")

Expected results

1. It is possible to control the movement of the Iris's blades from the OPI.
2. Only one between horizontal and vertical head can be selected.

3.8 Test case EMU-002 – Abort a running measure

The scope of this test is to verify that it is possible to abort a running measure of the EMU.

3.8.1 Support Environment

See 3.7.1.

3.8.2 Configuration

See 3.7.2

3.8.3 Setup

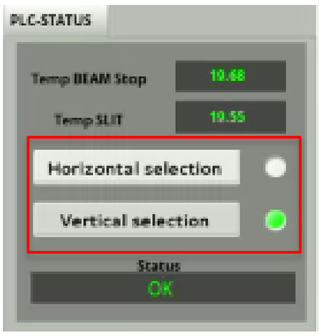
See 3.7.3

3.8.4 Procedure

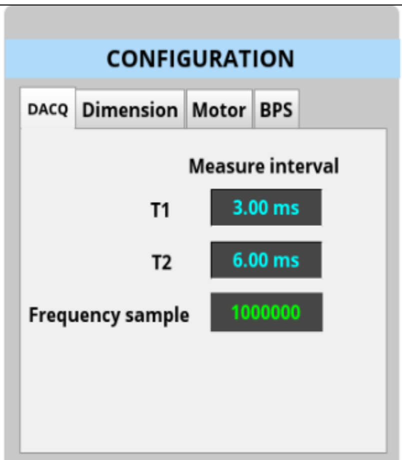
Procedure

1. On CSS, click the "EMU" button to open the OPI for the EMU.

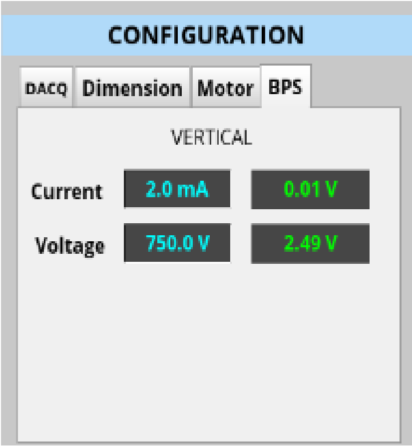
2. Click on "Horizontal Selection" or "Vertical Selection" buttons to select the horizontal or the vertical head respectively.



3. Configure the data acquisition parameters in the "DACQ" tab as follow:



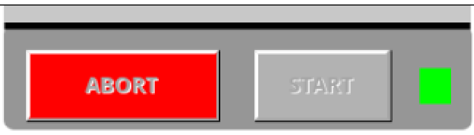
4. Configure the Bias Power Supply parameters in the "BPS" tab as follow:



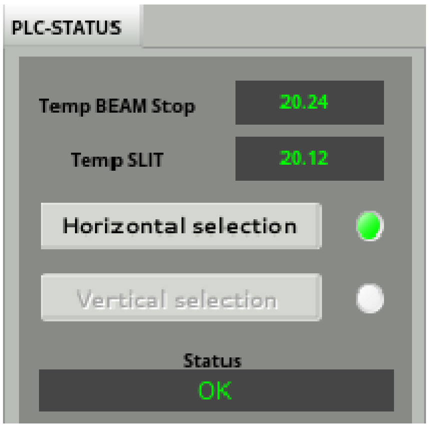
5. Configure the scanning parameters as follow:

6. Press the "Start" button to run the measure

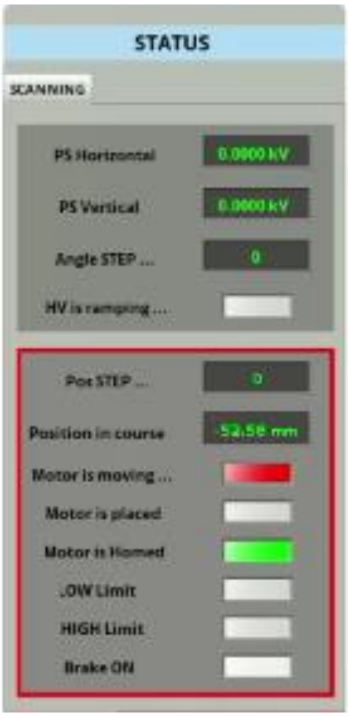
7. Verify that the green led close to the "Start" button turns on



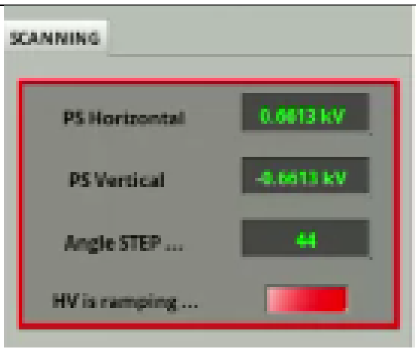
8. Verify that it is not possible to change the head of the EMU during a measure



9. Verify that the EMU moves to the start position set in step 6.



10. Verify that the measure starts when the EMU reach the first position



11. Verify that the EMU moves to the next position after reaching the "Angle STEP" values set on step 6.
12. Wait for three "Position STEP" and press the "Abort" button when enabled to stop the procedure.

Expected results

1. It is possible to configure the data acquisition parameters from the EMU's OPI.
2. It is possible to configure the Bias Power Supply parameter from the EMU's OPI.
3. It is possible to configure the scanning parameter from the EMU's OPI
4. It is possible to Start a new EMU measure
5. It is possible to Abort a running measure

3.9 Test case EMU-003 – Run a measure and extract data

The scope of this test is to verify that it is possible to abort a running measure of the EMU.

3.9.1 Support Environment

See 3.7.1.

3.9.2 Configuration

See 3.7.2

3.9.3 Setup

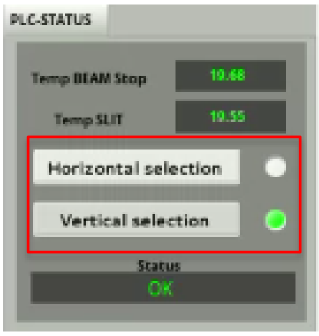
See 3.7.3

3.9.4 Procedure

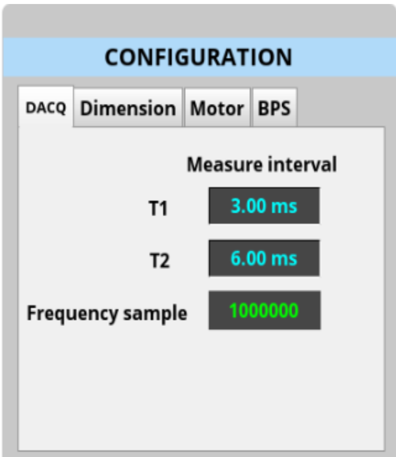
Procedure

1. On CSS, click the "EMU" button to open the OPI for the EMU.
2. From the diagnostic view, click the "EMU" button to open the OPI of the EMU.

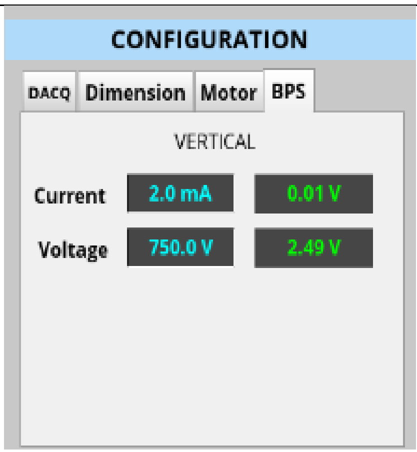
3. Click on "Horizontal Selection" or "Vertical Selection" buttons to select the horizontal or the vertical head respectively.



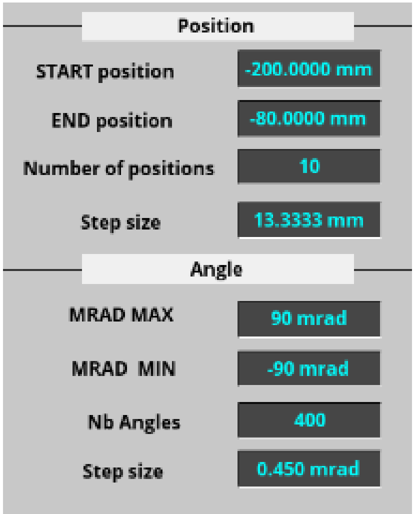
4. Configure the data acquisition parameters in the "DACQ" tab as follow:



5. Configure the Bias Power Supply parameters in the "BPS" tab as follow:

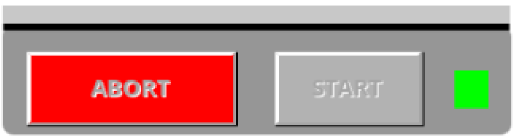


6. Configure the scanning parameters as follow:

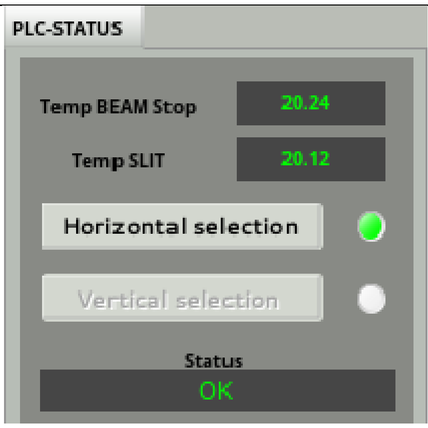


7. Press the "Start" button to run the measure

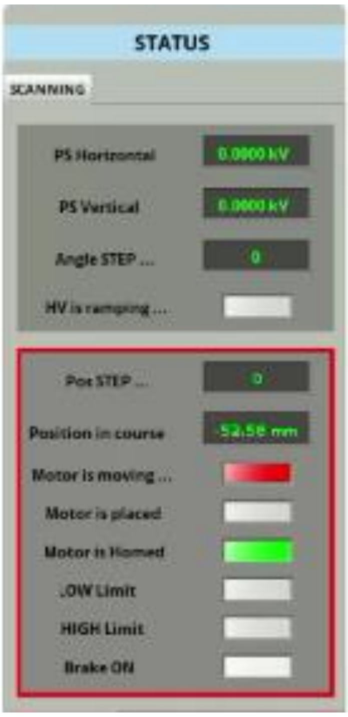
8. Verify that the green led close to the "Start" button turns on and that the "Start" button is disabled



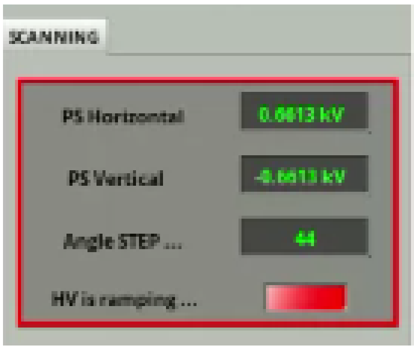
9. Verify that it is not possible to change the head of the EMU during a measure



10. Verify that the EMU moves to the start position set in step 6.



11. Verify that the measure starts when the EMU reach the first position



12. Verify that the EMU moves to the next position after reaching the “Angle STEP” values set on step 6.
13. Wait until the measure is completed (i.e. the “Position STEP” and “Angle Step” equal to the values set in step 6)

14. Press "Extract Data" button and verifies that data are extracted in the following file:

```
$ ls -t ~/data/emit/plotwin/*.dat | head -n 1
~/data/emit/plotwin/emittance<head>_<YYYY-MM-DD>_<HH-mm>.dat
```

Where:

- <head> can be Vertical or Horizontal
- <YYY-MM-DD> is the current date (year-month-day format)
- <HH-mm> is the current time (hour-minute format)

Expected results

6. It is possible to configure the data acquisition parameters from the EMU's OPI.
7. It is possible to configure the Bias Power Supply parameter from the EMU's OPI.
8. It is possible to configure the scanning parameter from the EMU's OPI.
9. It is possible to Start a new EMU measure.
10. At the end of measure data can be extracted.

4. DEFINITIONS, ACRONYMS AND ABBREVIATIONS

Abbreviation	Explanation of abbreviation
ESS	European Spallation Source
HV	High Voltage
HW	Hardware
ISrc	Ion Source
LEBT	Low Energy Beam Transport
LVM	Low Voltage Module
PLC	Programmable Logic Controller
SVP	System Verification Plan
SVR	System Verification Report
SW	Software
TN	Technical Network

5. REFERENCES

[1] Design document (ESS-0177833)
[2] WeTest
[3] ISrc test scenarios

[4] LEBT test scenarios
[5] ESS Jira Issue tracking system
[6] CSS Workspaces

ANNEX A. WETEST INSTALLATION AND CONFIGURATION

WeTest is an automated test tool that allows to define a set of tests scenario and collect them in test suites. Scenario file can are written using a human readable syntax (i.e. YAML) easily understandable and editable.

Another important feature is that it can generate a detailed report of the tests executed with indication of the result and a track of the error encountered in case of a failure.

Many of the ISRC & LEBT tests uses this tool that must be installed and properly configured in a console / workstation from where tests must be executed.

Even if it runs in any Linux system, it is recommended to use one ESS Development machine (physical or virtual) that contains already all libraries and additional tools required to execute the tests.

The following sections explains how to setup, install and run the WeTest environment.

A.1 Installation

Use conda (or miniconda) to setup a virtual environment. It is a good way to avoid poluting your system and to run Python 3 on CentOS 7 (note that currently WeTest is not compatible yet with Python 3).

```
$ wget https://repo.continuum.io/miniconda/Miniconda2-latest-Linux-
x86_64.sh
$ bash Miniconda2-latest-Linux-x86_64.sh
```

Add the conda-forge repository to avoid a bug in readline version provided by conda (This step in only necessary the first time)

```
$ conda config --add channels conda-forge
```

Create a new environment to run the ISRC & LEBT tests and “activate” it as follow:

```
$ conda create -n wetest-isrc-lebt python=2.7
Solving environment: done

## Package Plan ##

  environment location: /home/william/miniconda2/envs/wetest-isrc-lebt

  added / updated specs:
    - python=2.7


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    openssl-1.0.2o             |                0         3.5 MB
conda-forge
    certifi-2018.4.16          |           py27_0         142 KB
conda-forge
```

```
    ca-certificates-2018.4.16  |                0         139 KB
conda-forge
    ------------------------------------------------------------
                                           Total:         3.8 MB

The following NEW packages will be INSTALLED:

    ca-certificates: 2018.4.16-0      conda-forge
    certifi:         2018.4.16-py27_0 conda-forge
    ncurses:         5.9-10           conda-forge
    openssl:         1.0.2o-0         conda-forge
    pip:             9.0.3-py27_0     conda-forge
    python:          2.7.14-5         conda-forge
    readline:        7.0-0            conda-forge
    setuptools:      39.0.1-py27_0    conda-forge
    sqlite:          3.20.1-2         conda-forge
    tk:              8.6.7-0          conda-forge
    wheel:           0.31.0-py27_0    conda-forge
    zlib:            1.2.11-0         conda-forge

Proceed ([y]/n)? y


Downloading and Extracting Packages
openssl 1.0.2o:
######################################################################
######################################################################
######### | 100%
certifi 2018.4.16:
######################################################################
######################################################################
###### | 100%
ca-certificates 2018.4.16:
######################################################################
#################################################################### |
100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use:
# > source activate wetest-isrc-lebt
#
# To deactivate an active environment, use:
# > source deactivate
#

$ source activate wetest-isrc-lebt
```

Clone WeTest repository into a local directory (**you need a ESS Git account clone it**) and install it as follow:

```
(wetest-isrc-lebt) ... $ git clone https://
https://<user>@bitbucket.org/europeanspallationsource/wetest.git
...

(wetest-isrc-lebt) ... $ cd wetest
(wetest-isrc-lebt) ... $ python setup.py install
...
```

```
Finished processing dependencies for wetest==0.4.4
```

Deactivate the environment as described in the following section

A.2 Activate and deactivate WeTest environment

Once installed, it is possible to activate the WeTest environment for Isrc&LEBT as follow:

```
$ source activate wetest-isrc-lebt
# Run your test here

$ source deactivate
# Close wetest-isrc-lebt
```

ANNEX B. ISRC & LEBT CSS WORKSPACES

The operator screens (OPIs) realized for the ISrc & LEBT have been collected in two Control System Studio (CSS) workspaces and they are available on a dedicated bitbucket repository [6] .

It contains two workspaces:

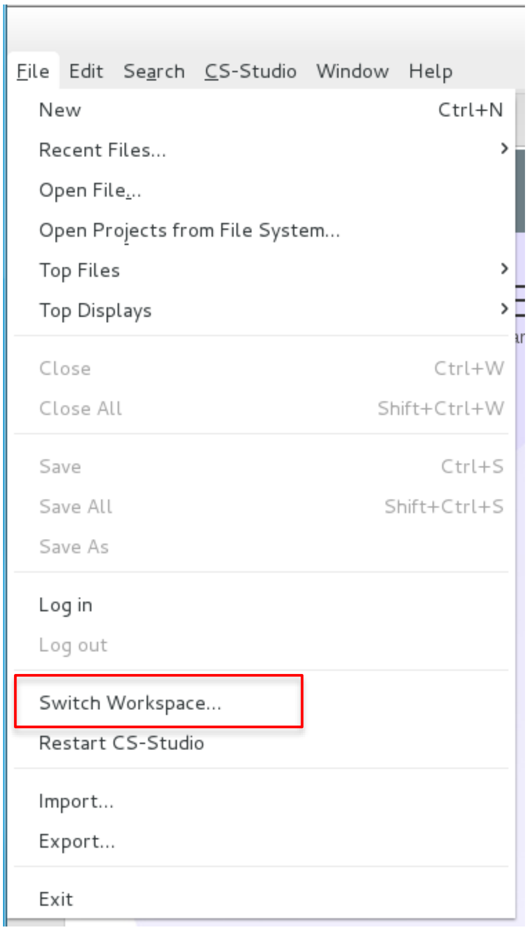
- CSS-ISrc-LEBT, controls for the ISRC & LEBT
- CSS-Diagnostic, for the Iris and the EMU tests

This section explains how to download and open these workspaces in CSS.

1. Clone the repository into a local directory (you need a ESS git account)

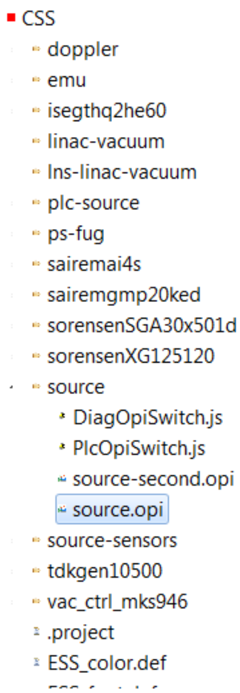
```
git clone https://<user>@bitbucket.org/europeanspallationsource/isrc-
lebt-css-acceptance-ws.git
```

2. Start CSS
3. Go to "`File -> Switch workspace...`"



4. Select one between `<your local directory>/CSS-ISrc-LEBT/` and `<your local directory>/CSS-Diagnostic/` on the "Select Workspace" dialog and click "Ok"

5. Select `CSS -> source -> source.opi` to open the operator screen for the ISrc & LEBT (i.e. CSS-ISrc-LEBT workspace)



6. Select `CSS -> source -> source-second.opi` to open the operator screen for Iris and EMU from the CSS-Diagnostic workspace