EPICS NTTABLES FOR MACHINE TIMING CONFIGURATION

A. Gorzawski[*] , J. P. S. Martins, N. Milas,
ESS, European Spallation Source, Lund, Sweden

Abstract

The European Spallation Source (ESS), currently under construction and initial commissioning in Lund, Sweden, will be the brightest spallation neutron source in the world, when its driving proton linac achieves the design power of 5 MW at 2 GeV. Such a high power requires production, efficient acceleration, and almost no-loss transport of a high current beam, thus making the design and beam commissioning of this machine challenging. The recent commissioning runs (2021-2023) showed a need for a consistent and robust way of setting up the machine for beam production. One of the big challenges at ESS is joining the machine setup and the timing setup limiting the need for operator actions. In this paper, we show a concept of using EPICS 7 NTTables to enable this machine settings consistency. Related to that, we also highlight a few challenges related to other EPICS tools like *Save And Restore* and *EPICS Archiver*.

INTRODUCTION

ESS is a collaboration of 17 European nations and its objective is to be the world's most powerful spallation neutron source [1]. The neutrons are produced by a pulsed proton beam hitting a solid, rotating tungsten target at a distance of 600 m from the ion source. The ESS linac, the driver of the protons onto the target, requires site-wide synchronization in order to accelerate the desired beam through its components. The production of the proton beam begins with the ion source providing the pulse of protons with an optimized current and of a given length [2]. Later, the two choppers (LEBT and MEBT) shape the pulse to the desired pulse length. The acceleration is given by the radio-frequency cavities pulsing at the correct moment. The overall repetition rate is a consequence of the definition of the arbitrary number of consecutive sets of timing events (cycles), that are pre-loaded before the ion source starts producing the beam. By design the actual maximum repetition rate is 14 Hz.

The ESS distributed control system is based on Experimental Physics and Industrial Control System (EPICS [3]). The full synchronization is provided by a distributed timing system with its own network infrastructure, and it is operated within a *Beam Production* environment. The entire timing system is configured with the pre-created timing tables, that consist of a definition of super-cycles, namely sequenced definitions of what the RF, actuators, and instrumentation do in the given cycle.

This paper summarises the efforts put in the beam commissioning of the Normal Conducting Linac throughout the multiple commissioning periods, i.e. from the ion source to the DTL4 FC with a special look at the machine timing setup along with the development and implementation of the new approach to handling timing tables using the `epics:NTTable` available in EPICS 7.

[*] arek.gorzawski@ess.eu

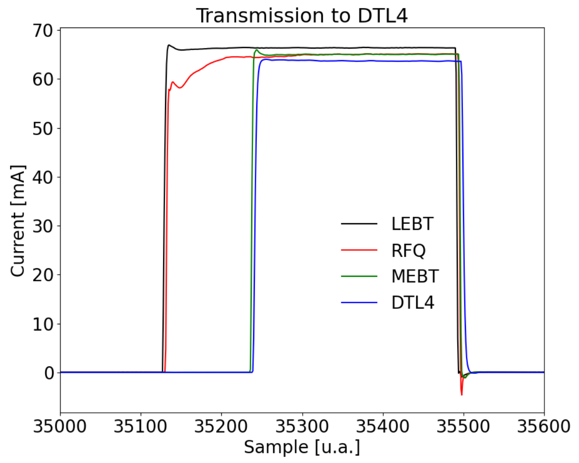


Figure 1: The successful beam pulse signatures read by available beam instrumentation in the fully available NCL part of ESS linac.

BEAM COMMISSIONING

Recently four designated periods of hardware and beam commissioning were successfully completed [4]. In each of the commissioning periods, we tested, verified, and established the ways the timing setup is used with an increasing number of commissioned equipment. The following top-level milestones were achieved:

1. *October 2021 - December 2021*, the MEBT FC commissioning with a small current beam, the first time with the timing system available,
2. *February 2022 - March 2022*, MEBT FC, commissioning with high current beam,
3. *May 2022 - July 2022*, DTL1 FC commissioning, low and high current beam,
4. *April 2023 - July 2023*, DTL4 FC commissioning, low and high current beam,

Figure 1 illustrates the Beam Current Monitor (BCM) signals from different locations at the ESS linac. One can see the shortening of the pulse in the more downstream locations (MEBT and DTL4). This is due to the acting MEBT chopper for the head of the beam. The responsibility for the success shown in that plot lies in a series of functioning timing-related events (see next section).

TIMING SYSTEM FOR ACCELERATOR

While the hardware details of the ESS timing system are described in [5] and [6], the top-level timing parameters are

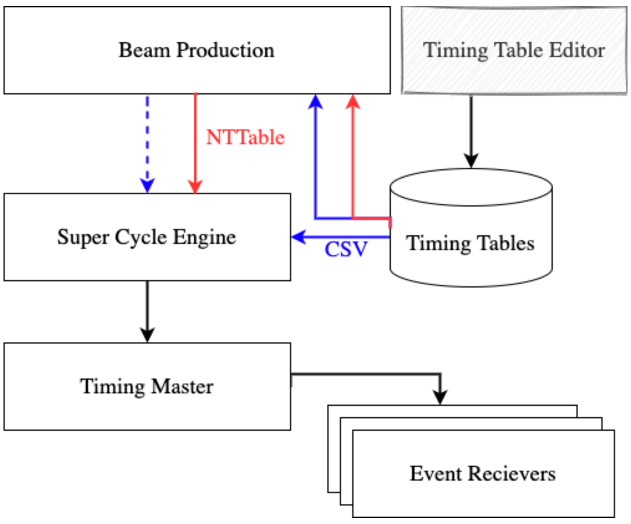


Figure 2: The conceptual breakdown of the Timing system-related layers. From the top (Control room perspective) is the Beam production, later the Super Cycle Engine, and later the Timing Master that distributes timing Events to all connected nodes. Aside, the editor (not available at the time of current commissions) and storage for the validated and available Timing Tables.

defined such that the ESS machine ticks in ≈ 71.43 ms cycles (≈ 14 Hz). The intermediate layer responsible for the setup of the actual frequency (anything reasonable between 0 Hz than 14 Hz) is called the *Super Cycle Engine*. The top-level operator control relates to *Beam Production* and allows to setup machine beyond only the beam pulse shape, but also sets up the beam mode, beam destination, and beam current. The latter one was only introduced and semi-commissioned in the last commissioning period. The conceptual breakdown of these layers is depicted in Fig. 2 and a detailed description of Engine and Beam Production can be found in the following sections. Data flow are shown in picture: the blue path (CSV) is the current implementation, and the red path (NT Table) is implemented in parallel.

Supercycle Engine

The cycles can be added together making a meaningful supercycle, so a collection of cycles. The operation team defines each supercycle and a certain supercycle is played during different site acceptance tests, or site integration tests, or the production operation. The machine ticking is produced by the ESS in-house EPICS module called Supercycle Engine (SCE) which is an extension of mrfioc2 [7] EPICS driver running on the Timing Master. In addition, the engine runs routines supporting asynchronous and synchronous management, safety (inhibiting required ESS actuators), the data bus, and the overall system status.

Timing Events

Withing the arbitrarily defined timing events for the accelerator use, three main groups can be isolated:

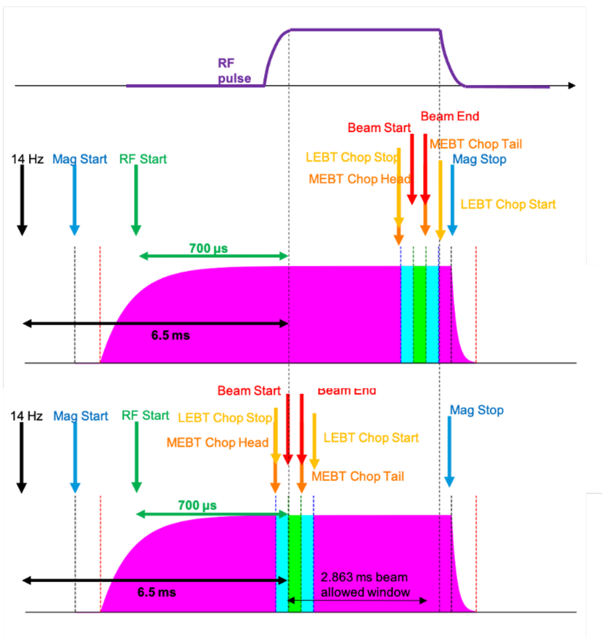


Figure 3: The conceptual breakdown of the Beam pulse creation at ESS. Timing events defining the start and the end of the beam pulse are the product of the choppers applied to the extracted pulse from the source. Beam pulse at the end of the main source extraction (top) or at the closest upfront position (bottom).

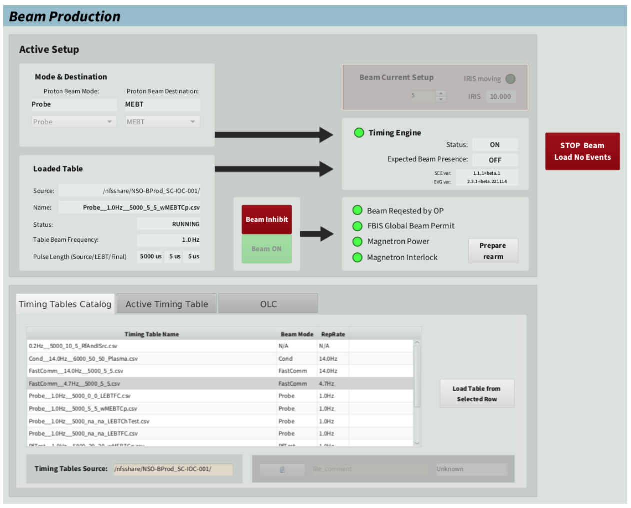
- Longitudinal beam shaping: Ion Source Magnetron; LEBT chopper; MEBT chopper,
- Beam pulse: Beam pulse position; RF start,
- Beam instrumentation acquisition and synchronous data acquisition.

These events are used as a base for the cycle definition. Along the cycle, a dedicated data buffer is sent, it contains the beam mode and beam destination, and also dedicated cycle-dependent beam parameters like intended beam energy and current for the purpose of the beam loading compensation provided by the LLRF. All of the events definitions and data buffer definitions are kept as the configuration of the *Supercycle Engine* in the dedicated git repository [8]. Figure 3 depicts the relative time of the timing event arrival along with the final shaping of the pulse. The magenta (Mag Start to Mag End) is the whole pulse extracted from the ion source, while the green one (Beam Start to Beam End) is the pulse that travels beyond the MEBT chopper. The additional events (responsible for the blue part of the beam - initiated with MEBT Head and Tail) can be additionally enabled/disabled depending on the request.

Beam Production

The ESS Linac beam pulse is set via the two main channels, that are channeled through the beam production interface (see Fig. 4):

- The pulse current is set by the hardware settings. Its control is realized via iris (aperture collimator) and is independent of the timing setup. The change of the



Proton Beam Mode: Probe
Proton Beam Destination: MEBT

Source: /nfsshare/NSO-BProd_SC-IOC-001/
Name: Probe__1.0Hz__5000_5_5_wMEBTCp.csv
Status: RUNNING
Table Beam Frequency: 1.0 Hz
Pulse Length (Source/LEBT/Final) 5000 us 5 us 5 us

IRIS 10.000

Status: ON
Expected Beam Presence: OFF
SCE ver: 1.1.1+beta.1
EVG ver: 2.3.1+beta.221114

Timing Table Name	Beam Mode	RepRate
0.2Hz__5000_10_5_RfAndISrc.csv	N/A	N/A
Cond__14.0Hz__6000_50_50_Plasma.csv	Cond	14.0Hz
FastComm__14.0Hz__5000_5_5.csv	FastComm	14.0Hz
FastComm__4.7Hz__5000_5_5.csv	FastComm	4.7Hz
Probe__1.0Hz__5000_0_0_LEBTFC.csv	Probe	1.0Hz
Probe__1.0Hz__5000_5_5_wMEBTCp.csv	Probe	1.0Hz
Probe__1.0Hz__5000_na_na_LEBTChTest.csv	Probe	1.0Hz
Probe__1.0Hz__5000_na_na_LEBTFC.csv	Probe	1.0Hz

Timing Tables Source: /nfsshare/NSO-BProd_SC-IOC-001/
file_comment Unknown

Figure 4: The main operator view for the Beam Production interface. Selection of the beam mode and destination is available (left-top part), along with the dedicated expert view on the predefined timing tables (bottom). The status of the loaded table is also shown along with the indication of the Software Interlock System (not shown here).

beam current is not a pulse-to-pulse process, and cannot be faster than tens of cycles,

- The beam repetition and pulse length are set by the corresponding timing events. By controlling the time when choppers should chop the beam, the pulse length can be achieved.

In the last commissioning period, we introduced the beam current setup through the beam production interface to streamline the process. However, due to manpower and intense commissioning activities, we did not deploy it entirely and only the proof of concept was available.

Input to Beam Interlock System

The combination of the three beam parameters, i.e. beam repetition rate, beam pulse length, and beam current (that is defined independently of the timing system) defines the envelope of the accelerator working point defined as a beam mode. At any time of the machine operation, the Fast Beam Interlock System (FBIS [9]) with Beam Current Monitors (BCMs) is supervising the compliance to the beam mode, i.e. asserting if the produced beam pulse belongs to the allowed parameter space. Pulse timing information distributed to the BCMs containing the planned beam pulse position in the cycle has therefore also a machine protection purpose.

ESS Beam Production layer ensures compliance with the currently set Beam Mode and Beam Destination. This information is a direct input to the Software Interlock System that without any doubt inhibits the beam. This is to prevent loading or configuring the timing system in the control room. The validity of the check strongly depends on the consistency of the set parameters, therefore the described hereafter method with NT Tables becomes very handy.

Managing the Timing Tables

Throughout the commissioning phases, the creation of the timing tables was done using the automated script (external Python notebook). The generated setup files were then sent to the shared repository and available for the system to pick up and load upon request.

The creation, the storage, the file format, and the bug fixing were heavily practiced and revised during all four periods of the commissioning. Throughout the commissioning and as an outcome we allocated resources to build a dedicated tool for editing, storing, and verifying the available timing tables, the Supercycle Editor. While the prototypes were available during the 3rd commissioning phase, the production tool will come later in the future commissioning phase in early 2024. The timing table validation (semantic and numeric) was implemented at the end of 1st commissioning period in the *Beam Production* IOC and it was extended during the later commissionings.

With the current approach (post-4th commissioning period), the timing setup is prone to errors in the data being set (i.e. read) from the externally kept files. Therefore, another important point is traceability. In the current implementation, the EPICS PVs responsible for the name of the timing table are archived, however, the state of the file is not. This setup is far from ideal and currently undergoing changes and requires further developments.

A separate note should also be taken on the *Save And Restore* [10]. Whilst settings for many PVs are persisted as one snapshot, there is no guarantee that the snapshot will be consistently restored. This is due to the nature of how *Save And Restore* works and goes beyond the scope of this paper. However, as an implication of this fact, we could not rely on that aspect either.

During the recent commissioning (i.e. period 4), we introduced in the Beam Production layer an additional mechanism to expose a dedicated PV that keeps a consistent structure of the actual set timing setup.

Normative Types and Structures

As a solution for the issues described in the previous paragraph Normative Types [11] come in very handy. In this type, the user can define more than one field in the same PV, reaching up to all structures that can be treated as atomic as canonical one-type records, e.g. double or string. Moreover, these structures can have additional fields with metadata. These kinds of records would provide a solution for the atomic sense of *Save And Restore* snapshot and would ensure that instead of the record being processed (i.e. only one record processing is needed) all the configuration data is passed and sent.

Using NT Tables for the Timing Setup

The first implementation of the `epics:NTTable` was done to mitigate the setup (and read back) of the timing tables set in the ESS linac. Based on the availability of the supporting software (mainly *Save And Restore*) we did not use set points but only focused on the read-backs and used that as examples on how the whole timing configuration should take place. That approach allowed us to identify the bottlenecks of future usage:

- Possibility to set it through the OPIs would require `epics:NTTable` structure-aware widget, i.e. should those PVs have dedicated widget? Or should the interaction with them not be allowed?
- Possibility to save and restore it from *Save And Restore*,
- Possibility to archive and browse it from the Archiver.

We used the direct implementation using the `pvRecord` and `pvDatabase`. Listing 1 shows the basic definition of the user type, here a structure that supports sub-structures and arrays. Thanks to that definition, a cyclic character of the timing table can be reflected for the need of consistent setup via PVs. Listing 2 shows a console output of `pvget` query for the defined record after being processed and exposed by the Beam Production IOC.

```
StructureConstPtr topStructure(
getFieldCreate()->createFieldBuilder()
->setId("epics:nt/NTTable:1.0")
->addNestedStructure("MetaData")
    ->add("BLenSrc", pvd::pvFloat)
    ->add("BLenLebt", pvd::pvFloat)
    ->add("BLenMebt", pvd::pvFloat)
    ->add("PulseFrequency", pvd::pvFloat)
->endNested()
->addArray("CycleHeader", pvd::pvString)
->addArray("DataBufferHeader", pvd::pvString)
->addNestedStructureArray("SuperCycle")
    ->addArray("Cycle", pvd::pvFloat)
    ->addArray("DataBuffer", pvd::pvFloat)
->endNested()
->createStructure());
```

Listing 1: Proposed definition for the Timing Table PV using `pvDatabase`. Additional metadata for quicker identification of the Super Cycle content.

```
BPROD:Ops:TTLoaded-RB epics:nt/NTTable:1.0
structure MetaData
    float BLenSrc 5000
    float BLenLebt 5
    float BLenMebt 5
    float PulseFrequency 14
string[] CycleHeader [Id, BiAcqSt,
    BiAcqStAhead, IonMagSt, IonMagEnd,
    IonMagStAhead, IonMagEndAhead, RfSt,
    LebtCpOff, LebtCpOn, LebtCpOffAhead,
    LebtCpOnAhead, BPulseSt, BPulseEnd,
    MebtCpHead, MebtCpTail, BPulseStAhead,
    BPulseEndAhead, AcqSync]
    string[] DataBufferHeader [BLen, BEn, BCurr]
structure[] SuperCycle
    structure
        float[] Cycle [0,100,90,1550,6550,
        1450,6450,5800,6500.01,6505.01,
        6400.01,6405.01,6500,6504.99,
        -1,-1,6400,
        6404.99,50000]
        float[] DataBuffer [5,3600,6]
```

Listing 2: Part of the proposed definition for the Timing Table read from the PV for one of the operationaly used Super Cycles. The consistency of all data is encapsulated within one PV.

CONCLUSIONS AND OUTLOOK

Using the first implementation of the timing tables as `epics:NTTable` data we showed the broad range of simplifications and improvements in the process of consistent setting of multi-parameter services such as Beam Production including the timing system at ESS. The successful commissioning runs were supported by reliable implementations of the Beam Production layer. Looking beyond the current runs, we identified a few immediate points to improve in the next commissioning phases.

PVXS

`PVXS` is a software package that implements the client/server interface to EPICS PVAccess [11] protocol. It is being used by two major EPICS modules on the community: the EPICS PVA Gateway and the Python for PVAccess (P4P) projects. Currently, a plan for re-implementation using `PVXS` [12] libraries instead of `pvRecord` and `pvDatabase` is ongoing. `PVXS` offers a more compact syntax to handle PV structures and safer methods to handle PV values via C++ code, in parallel with a very efficient implementation of the PVA network operations. The project has more support from the community and more test coverage of the code than the vast majority of EPICS-related software [13].

Software Integration

After implementing it at the IOC level, another challenge is to use the existing ecosystem to profit from the NT Tables capability. Currently, the efforts are carried on the *Save And Restore* level to be able to save and restore these non-primitive models. Also *EPICS Archiver* and its applience layer requires the appropriate changes for the persistence and extraction of the normative types.

Super Cycle Engine

A common module (for Timing Table records) between Beam Production and Super Cycle Engine needs to be implemented to replace the CSV file import path. The same module can be then used to exchange the timing tables between two layers.

Super Cycle Editor

As shown in Fig. 2 the editor for the Timing Tables was not used, instead we manually (i.e. via dedicated Python scripts) edited and maintained the catalog of valid tables. It is planned to deliver a dedicated application (that encapsulates the creation script but also provides storage service) for the next commissioning phase in 2024, and will simplify the setup of the *exotic* beam pulses but also serve as a good base for the *Simplified Beam Pulse Setup* described below.

Simplified Beam Pulse Setup

In the future version, it is envisaged to use timing tables as a base to dynamically generate new ones, based on the operator request. In this mode, the Beam Production would dynamically recompute and set the PVs related to the timing setup. This way of operation lifts most of the complexity of the timing table PV setup while (internally to the Beam Production) providing a full inhibit mechanism against mistyping in the operator console.

ACKNOWLEDGMENTS

The implementation, deployment, and extensive test of the Beam Production environment involved many ESS colleagues. We would like to thank our collaborators from the Operations, Beam Physics, and Integrated Control Systems.

REFERENCES

[1] R. Garoby *et al.*, "The European Spallation Source Design", *Phys, Scr.*, vol. 93, p. 014001, 2018. doi:10.1088/1402-4896/aa9bff

[2] R. Miyamoto *et al.*, "Highlights from the first beam commissioning stage at ESS for its ion source and low energy beam transport", *J. Instrum.*, vol. 15, p. P07027, 2020. doi:10.1088/1748-0221/15/07/p07027

[3] Experimental Physics and Industrial Control System (EPICS), https://epics-controls.org, https://docs.epics-controls.org/en/latest/

[4] D.C. Plostinar, *et al.*, "Status of the Normal Conducting Linac at the European Spallation Source", in *Proc. IPAC'22*, Bangkok, Thailand, Jun. 2022, pp. 2019–2022. doi:10.18429/JACoW-IPAC2022-WEPOTK001

[5] PICMG MicroTCA Overview, https://www.picmg.org/openstandards/microtca

[6] J. J. Jamroz, J. Cereijo Garcia, T. Korhonen, and J. H. Lee, "Timing System Integration with MTCA at ESS", in *Proc. ICALEPCS'19*, New York, NY, USA, Oct. 2019, pp. 1265. doi:10.18429/JACoW-ICALEPCS2019-WEPHA071

[7] EPICS driver for Micro Research Finland event timing system devices, version 2 (mrfioc2), https://github.com/epics-modules/mrfioc2

[8] ESS Reference Tables "reftabs", https://gitlab.esss.lu.se/icshwi/reftabs

[9] S. Pavinato *et al.*, "The ESS Fast Beam Interlock System - Design, Deployment and Commissioning of the Normal Conducting Linac", presented at ICALEPCS 2023, Cape Town, South Africa, 2020, paper TUPDP081, this conference.

[10] EPICS Save and Restore tool, https://gitlab.esss.lu.se/ics-software/jmasar-service

[11] https://docs.epics-controls.org/en/latest/pv-access/Normative-Types-Specification.html

[12] PVXS client/server for PVA Protocol, https://mdavidsaver.github.io/pvxs/

[13] M. Davidsaver, "PVXS in your IOC", in *EPICS Collaboration Meeting*, Chicago, USA, Apr. 2023. https://indico.fnal.gov/event/58280/contributions/264559/