Description	Engineering Manual
Document number	ESS-XXXXXXXX
Date	January 22, 2018
Revision	0.1
State	Early Draft
Classification	ESS Use Only
Page	1 (12)

Timing System Introduction

for Raster Magnet System

	Name (Role/Title)
Author	Javier Cereijo Garcia Jeong Han Lee
Reviewer	TBD
Owner	ICS
Approver	ICS

Contents

Description	Engineering Manual
Document number	ESS-XXXXXXXX
Date	January 22, 2018
Revision	0.1
State	Early Draft
Classification	ESS Use Only

1 Overview

At European Spallation Source (ESS), Integrated Control System (ICS) uses the Micro Research Finland (MRF) Timing System[1] foir the synchronization the facility. This document describes the basic set up to interface the Timing System to the Raster Magnet System.

1.1 Scope

- This document describes the hardware sent to Aarhus University from ICS for the Timing Synchronization of the Raster Magnet Power Supplies.
- This document describes the configuration of the software to set the Timing System up and running.
- This document explains how to set up an EPICS (Experimental Physics and Industrial Control System) IOC (Input/Out Controller) to synchronize the Raster Magnet Power Supplies.

1.2 Target Audience

This document is targeted to the Raster Magnet System technical stakeholders of the ESS timing system. It is assumed that the target audience has a technical background in the MRF Timing System, the EPICS development, and a Linux environment.

2 System Description

MRF Technical Reference [see 1, p45] explained Event Receivers and wrote :

> *Event Receivers (EVRs) decode timing events and signals from an optical event stream transmitted by an Event Generator (EVG). Events and signals are received at predefined rate the event clock that is usually divided down from an accelerators main RF reference. The event receivers lock to the phase event clock of the Event Generator and are thus phase locked to the RF reference. Event Receivers convert event codes transmitted by an Event Generator to hardware outputs. They can also generate software interrupts and store the event codes with globally distributed timestamps into FIFO memory to be read by a CPU.*

The latest series of MRF EVRs can also work in standalone, without the need of an external EVG to generate events. The current configuration of the hardware shipped works in standalone mode.

[1] http://www.mrf.fi/

2.1 Hardware description

Table 1 shows the hardware list sent to Aarhus University. Here, `TAG` is used as the prefix of the ICS internal inventory system in order to track it down.

Hardware	Info	Serial Number
MRF PCIe-EVR-300DC*	`ICS TAG-402`	M263047
MRF IFB-300 and micro-SCSI cable	`ICS TAG-490`	K472001
TTL, HFBR-1414, HFBR-1258 modules*		
LC-LC, ST-ST cables		
Adlink-MVP-6021 and rack shelf	`ICS TAG-838`	93-41036-G10E

Table 1 Hardware List (*PCIe-EVR-300DC is already installed in PC, universal modules are already installed in IFB-300).

Figure 1 shows how the hardware should be put together. The IFB-300 with the universal modules is not shown in the figure. The cable from the PCIe-EVR-300DC to the IFB-300 should be connected with the system powered down. The LC-LC cable should be connected to the SFP module of the PCIe-EVR-300DC in a loop.

The IFB-300 has three universal modules installed, each with two outputs:

- UNIV0 and UNIV1 outputs are HFBR-1414 optical transmitters, intended for the pre-trigger signal.
- UNIV4 and UNIV5 outputs are HFBR-1528 optical transmitters, intended for the polarity signal.
- UNIV8 and UNIV9 outputs are TTL level with LEMO connectors, intended for the data acquisition of the Bdot signals.

2.2 Software description

Table 2 shows the software list, its environment and log-in information.

After turning on the PC and logging in, one should run the `E3` command to load the EPICS environment. After this one could start an IOC:

```
iocuser@icslab-ipc01: ~$ E3
iocuser@icslab-ipc01: ~$ iocsh.bash
#
# Start at "2018-W04-Jan22-1559-48-CET"
#
# Version information:
# European Spallation Source ERIC : iocsh.bash (v0.2-9d66ded.PID-586)
## can you see this?
```

Figure 1 Hardware put together ready for rack installation. The cable from the PCIe-EVR-300DC to the IFB-300 should be connected with the system powered down.

```
#
# HOSTDISPLAY=""
# WINDOWID=""
# PWD="/home/iocuser"
# USER="iocuser"
# LOGNAME="iocuser"
# EPICS_HOST_ARCH="linux-x86_64"
# EPICS_BASE="/epics/bases/base-3.15.5"
# EPICS_LOCATION="/epics/bases"
# EPICS="/epics/bases"
# EPICS_MODULES="/epics/modules"
# REQUIRE="require"
# REQUIRE_VERSION="2.5.4"
# REQUIRE_BIN="/epics/modules/require/2.5.4/bin"
# REQUIRE_LIB="/epics/modules/require/2.5.4/R3.15.5/lib"
# REQUIRE_DBD="/epics/modules/require/2.5.4/R3.15.5/dbd"
# EPICS_CA_AUTO_ADDR_LIST="yes"
# EPICS_CA_ADDR_LIST=""
# PATH="/epics/modules/require/2.5.4/bin:/epics/bases/base-3.15.5/bin/linux-x86_64:/usr/local/bin
    :/usr/bin:/bin:/usr/local/games:/usr/games:/sbin:/home/iocuser/bin:/opt/etherlab/bin:/opt/
    etherlab/sbin"
# LD_LIBRARY_PATH="/epics/bases/base-3.15.5/lib/linux-x86_64:/epics/modules/require/2.5.4/R3.15.5/
```

Item	Version Info.
CentOS Linux	`7.4.1708`
Kernel	`3.10.0-693.11.6.el7.x86_64`
mrf kernel module	version : `1` / srcversion `A998B22F1425D7388F5F7A7`
E3	`0.2`
EPICS Base	`3.15.5`
E3 require	`2.5.4`
mrfioc2	`2.2.0`
devLib2	`2.9.0`
User	`timinguser`
Password	`Rastering`

Table 2 Software and its version information.

```
    lib/linux-x86_64:/usr/local/lib:/home/iocuser/lib:/opt/etherlab/lib"
#
# Please Use Version and other environment variables
# in order to report or debug this shell
#
# Loading the mandatory require module ...
#
dlload /epics/modules/require/2.5.4/R3.15.5/lib/linux-x86_64/librequire.so
dbLoadDatabase /epics/modules/require/2.5.4/R3.15.5/dbd/require.dbd
require_registerRecordDeviceDriver
Loading module info records for require
#
#
epicsEnvSet IOCSH_PS1 "9d66ded.icslab-ipc01.599 > "
epicsEnvShow T_A
T_A is not an environment variable.
epicsEnvShow EPICS_HOST_ARCH
EPICS_HOST_ARCH=linux-x86_64
iocInit
Starting iocInit
############################################################################
## EPICS R3.15.5-EEE-3.15.5-patch
## EPICS Base built Jan  2 2018
############################################################################
iocRun: All initialization complete
9d66ded.icslab-ipc01.599 >
```

3 Timing System configuration

There are 2 files important for setting the timing system for the raster magnet power supplies up and running. They can live wherever, for our purposes, let's assume they live in the Desktop. The first one is the IOC startup file `raster01.cmd`:

```
# This the full setup for the Timing System with E3.
#
#require require,2.5.4
require devlib2,2.9.0
require mrfioc2,2.2.0
require iocStats,1856ef5


epicsEnvSet("ENGINEER","Han")
epicsEnvSet("LOCATION","Table at ICS Tuna Lab")

epicsEnvSet("EPICS_CA_MAX_ARRAY_BYTES","10000000")

epicsEnvSet("IOC", "RM01-PCIE")
epicsEnvSet("DEV1", "EVR1")

epicsEnvSet("MainEvtCODE" "14")
epicsEnvSet("HeartBeatEvtCODE"   "122")
epicsEnvSet("ESSEvtClockRate"  "88.0525")

mrmEvrSetupPCI("$(DEV1)",  "01:00.0")
dbLoadRecords("evr-pcie-300dc-ess.db","EVR=$(DEV1), SYS=$(IOC), D=$(DEV1), FEVT=$(ESSEvtClockRate)
    ")

# needed with software timestamp source w/o RT thread scheduling
var evrMrmTimeNSOverflowThreshold 100000


# iocStats
dbLoadRecords("iocAdminSoft.db", "IOC=$(IOC)-IocStats")



iocInit()

dbl > "${IOC}_PVs.list"

dbpf $(IOC)-$(DEV1):Ena-Sel 1

# Get current time from system clock
dbpf $(IOC)-$(DEV1):TimeSrc-Sel 2

# Set up the prescaler that will trigger the sequencer at 14 Hz
dbpf $(IOC)-$(DEV1):PS0-Div-SP 6289464 # divide event clock (88.0525 MHz) by 6289464 to get 14 Hz

# Set up the sequencer
dbpf $(IOC)-$(DEV1):SoftSeq0-RunMode-Sel 0 # normal mode
dbpf $(IOC)-$(DEV1):SoftSeq0-TrigSrc-2-Sel 2 # prescaler 0
dbpf $(IOC)-$(DEV1):SoftSeq0-TsResolution-Sel 2 # us
dbpf $(IOC)-$(DEV1):SoftSeq0-Load-Cmd 1
dbpf $(IOC)-$(DEV1):SoftSeq0-Enable-Cmd 1
# Remember to run the script that populates the sequencer!

# Set up output 8 at 14 Hz
```

```
dbpf $(IOC)-$(DEV1):DlyGen0-Evt-Trig0-SP $(MainEvtCODE) # set delay generator 0 to trigger on
     event MainEvtCode (14 Hz)
dbpf $(IOC)-$(DEV1):DlyGen0-Width-SP 1000 # Time up in us, as selected with $(IOC)-$(DEV1):
     SoftSeq0-TsResolution-Sel
# dbpf $(IOC)-$(DEV1):OutRB08-Src-SP 0 # trigger output 8 on delay generator 0

# Set up output 9 at 10 KHz
dbpf $(IOC)-$(DEV1):PS1-Div-SP 8805 # divide event clock (88.0825 MHz) by 8805 to get 10 KHz
# dbpf $(IOC)-$(DEV1):OutRB09-Src-SP 41 # trigger output 9 on prescaler 1

pcidiagset 2 0 0
#pciread 32 0x50 100
pciwrite 32 0x50 42000200
```

And the second file `populate_sequencer_raster01.cmd` will populate the sequencer:

```
caput -a RM01-PCIE-EVR1:SoftSeq0-EvtCode-SP 2 14 127
caput -a RM01-PCIE-EVR1:SoftSeq0-Timestamp-SP 2 0 1
caput RM01-PCIE-EVR1:SoftSeq0-Commit-Cmd 1
```

Open a terminal and run `E3` for loading the environment and then `iocsh.bash raster01.cmd`. The output should be something like this:

```
iocuser@icslab-ipc01: Desktop$ E3
iocuser@icslab-ipc01: Desktop$ iocsh.bash raster01.cmd
#
# Start at "2018-W04-Jan22-1707-40-CET"
#
# Version information:
# European Spallation Source ERIC : iocsh.bash (v0.2-9d66ded.PID-13580)
## can you see this?
#
# HOSTDISPLAY=""
# WINDOWID=""
# PWD="/home/iocuser/Desktop"
# USER="iocuser"
# LOGNAME="iocuser"
# EPICS_HOST_ARCH="linux-x86_64"
# EPICS_BASE="/epics/bases/base-3.15.5"
# EPICS_LOCATION="/epics/bases"
# EPICS="/epics/bases"
# EPICS_MODULES="/epics/modules"
# REQUIRE="require"
# REQUIRE_VERSION="2.5.4"
# REQUIRE_BIN="/epics/modules/require/2.5.4/bin"
# REQUIRE_LIB="/epics/modules/require/2.5.4/R3.15.5/lib"
# REQUIRE_DBD="/epics/modules/require/2.5.4/R3.15.5/dbd"
# EPICS_CA_AUTO_ADDR_LIST="yes"
# EPICS_CA_ADDR_LIST=""
# PATH="/epics/modules/require/2.5.4/bin:/epics/bases/base-3.15.5/bin/linux-x86_64:/usr/local/bin
     :/usr/bin:/bin:/usr/local/games:/usr/games:/sbin:/home/iocuser/bin:/opt/etherlab/bin:/opt/
     etherlab/sbin"
# LD_LIBRARY_PATH="/epics/bases/base-3.15.5/lib/linux-x86_64:/epics/modules/require/2.5.4/R3.15.5/
     lib/linux-x86_64:/usr/local/lib:/home/iocuser/lib:/opt/etherlab/lib"
#
# Please Use Version and other environment variables
# in order to report or debug this shell
#
# Loading the mandatory require module ...
#
dlload /epics/modules/require/2.5.4/R3.15.5/lib/linux-x86_64/librequire.so
dbLoadDatabase /epics/modules/require/2.5.4/R3.15.5/dbd/require.dbd
```

```
require_registerRecordDeviceDriver
Loading module info records for require
#
#
< "raster01.cmd"
# This the full setup for the Timing System with E3.
#
#require require,2.5.4
require devlib2,2.9.0
Module devlib2 version 2.9.0 found in /epics/modules/devlib2/2.9.0/
Loading library /epics/modules/devlib2/2.9.0/R3.15.5/lib/linux-x86_64/libdevlib2.so
Loaded devlib2 version 2.9.0
Loading dbd file /epics/modules/devlib2/2.9.0/R3.15.5/dbd/devlib2.dbd
Calling function devlib2_registerRecordDeviceDriver
Loading module info records for devlib2
require mrfioc2,2.2.0
Module mrfioc2 version 2.2.0 found in /epics/modules/mrfioc2/2.2.0/
Module mrfioc2 depends on devlib2 2.9+
Module devlib2 version 2.9.0 already loaded
Loading library /epics/modules/mrfioc2/2.2.0/R3.15.5/lib/linux-x86_64/libmrfioc2.so
Loaded mrfioc2 version 2.2.0
Loading dbd file /epics/modules/mrfioc2/2.2.0/R3.15.5/dbd/mrfioc2.dbd
Calling function mrfioc2_registerRecordDeviceDriver
Loading module info records for mrfioc2
require iocStats,1856ef5
Module iocStats version 1856ef5 found in /epics/modules/iocStats/1856ef5/
Loading library /epics/modules/iocStats/1856ef5/R3.15.5/lib/linux-x86_64/libiocStats.so
Loaded iocStats version 1856ef5
Loading dbd file /epics/modules/iocStats/1856ef5/R3.15.5/dbd/iocStats.dbd
Calling function iocStats_registerRecordDeviceDriver
Loading module info records for iocStats
epicsEnvSet("ENGINEER","Han")
epicsEnvSet("LOCATION","Table at ICS Tuna Lab")
epicsEnvSet("EPICS_CA_MAX_ARRAY_BYTES","10000000")
epicsEnvSet("IOC", "RM01-PCIE")
epicsEnvSet("DEV1", "EVR1")
epicsEnvSet("MainEvtCODE" "14")
epicsEnvSet("HeartBeatEvtCODE"   "122")
epicsEnvSet("ESSEvtClockRate"  "88.0525")
mrmEvrSetupPCI("EVR1",  "01:00.0")
Notice: devPCIFindSpec() expect B:D.F in hex
Device EVR1  1:0.0 slot=(null)
Using IRQ 125
FWVersion 0x17080207
Found version 519
Found SFP EEPROM
Sequencer capability detected
PCIe: Out FP:0 FPUNIV:16 RB:0 IFP:2 GPIO:0
Enabling interrupts
EVR FIFO task start
dbLoadRecords("evr-pcie-300dc-ess.db","EVR=EVR1, SYS=RM01-PCIE, D=EVR1, FEVT=88.0525")
macLib: macro PINITSEQ is undefined (expanding string # $(PINITSEQ,undefined)Cont-FOut_
)
Warning: 'evr-pcie-300dc-ess.db' line 1441 has undefined macros
macLib: macro PINITSEQ is undefined (expanding string # $(PINITSEQ,undefined)Cont-FOut_
)
Warning: 'evr-pcie-300dc-ess.db' line 1613 has undefined macros
macLib: macro PINITSEQ is undefined (expanding string # $(PINITSEQ,undefined)Cont-FOut_
)
Warning: 'evr-pcie-300dc-ess.db' line 1785 has undefined macros
# needed with software timestamp source w/o RT thread scheduling
var evrMrmTimeNSOverflowThreshold 100000
```

```
# iocStats
dbLoadRecords("iocAdminSoft.db", "IOC=RM01-PCIE-IocStats")
iocInit()
Starting iocInit
############################################################################
## EPICS R3.15.5-EEE-3.15.5-patch
## EPICS Base built Jan  2 2018
############################################################################
require: record RM01-PCIE:MODULES not found
require: record RM01-PCIE:VERSIONS not found
require: record RM01-PCIE:MOD_VER not found
Set EVR clock 88052500.000000
iocRun: All initialization complete
dbl > "RM01-PCIE_PVs.list"
dbpf RM01-PCIE-EVR1:Ena-Sel 1
DBR_STRING:          "Enabled"
# Get current time from system clock
dbpf RM01-PCIE-EVR1:TimeSrc-Sel 2
DBR_STRING:          "Sys. Clock"
# Set up the prescaler that will trigger the sequencer at 14 Hz
dbpf RM01-PCIE-EVR1:PS0-Div-SP 6289464 # divide event clock (88.0525 MHz) by 6289464 to get 14 Hz
DBR_LONG:           6289464   0x5ff838
# Set up the sequencer
dbpf RM01-PCIE-EVR1:SoftSeq0-RunMode-Sel 0 # normal mode
DBR_STRING:          "Normal"
dbpf RM01-PCIE-EVR1:SoftSeq0-TrigSrc-2-Sel 2 # prescaler 0
DBR_STRING:          "Prescaler 0"
dbpf RM01-PCIE-EVR1:SoftSeq0-TsResolution-Sel 2 # us
DBR_STRING:          "uSec"
dbpf RM01-PCIE-EVR1:SoftSeq0-Load-Cmd 1
DBR_STRING:          "1"
dbpf RM01-PCIE-EVR1:SoftSeq0-Enable-Cmd 1
DBR_STRING:          "1"
# Remember to run the script that populates the sequencer!
# Set up output 8 at 14 Hz
dbpf RM01-PCIE-EVR1:DlyGen0-Evt-Trig0-SP 14 # set delay generator 0 to trigger on event
     MainEvtCode (14 Hz)
DBR_LONG:           14        0xe
dbpf RM01-PCIE-EVR1:DlyGen0-Width-SP 1000 # Time up in us, as selected with RM01-PCIE-EVR1:
     SoftSeq0-TsResolution-Sel
DBR_DOUBLE:         1000
# dbpf $(IOC)-$(DEV1):OutRB08-Src-SP 0 # trigger output 8 on delay generator 0
# Set up output 9 at 10 KHz
dbpf RM01-PCIE-EVR1:PS1-Div-SP 8805 # divide event clock (88.0825 MHz) by 8805 to get 10 KHz
DBR_LONG:           8805      0x2265
# dbpf $(IOC)-$(DEV1):OutRB09-Src-SP 41 # trigger output 9 on prescaler 1
pcidiagset 2 0 0
Looking for 2:0.0
Mapping 2:0.0
BAR 0 from 0x7fed4a4af000 for 262144 bytes
#pciread 32 0x50 100
pciwrite 32 0x50 42000200
epicsEnvSet IOCSH_PS1 "9d66ded.icslab-ipc01.13593 > "
epicsEnvShow T_A
T_A=linux-x86_64
epicsEnvShow EPICS_HOST_ARCH
EPICS_HOST_ARCH=linux-x86_64
9d66ded.icslab-ipc01.13593 >
```

Don't close this terminal, since it is were the IOC is running. Open a second terminal and run `E3` and

```
sh populate_sequencer_raster01.cmd
```

The output will show the old and new waveform records (not shown for clarity).

3.1 User configuration

The EVR is now internally generating a signal at 14 Hz with a width of 1 ms, and a square signal at 10 KHz. To get these signal on the outputs, one should set the `RM01-PCIE-EVR1:OutRBXX-Src-SP` records to 0 to get the 14 Hz signal and 41 to get the 10 KHz. For example, if one wants output 08 at 14 Hz, and output 09 at 10 KHz, one would run in a terminal where `E3` has been loaded (for example the same where `sh populate_sequencer_raster01.cmd` has been run):

```
caput RM01-PCIE-EVR1:OutRB08-Src-SP 0
caput RM01-PCIE-EVR1:OutRB09-Src-SP 41
```

If one wants to run these commands in the IOC shell, replace `caput` with `dbpf`. One can include these `dbpf` commands at the end of the `raster01.cmd` file so that they happen automatically when the IOC starts.

To change the width of the 14 Hz signal, one should run:

```
caput RM01-PCIE-EVR1:DlyGen0-Width-SP *time*
```

where `*time*` is expressed in μs with a granulairy of 11.37 ns.

To change the frequencies, one must modify the value of the prescalers. The prescalers work by dividing the event clock frequency of 88.0525 MHz (actually 88.0519 MHz in our standalone system) by an integer (minimum value 2). The current value of prescaler 0 is 6289464, as one can check with

```
caget RM01-PCIE-EVR1:PS0-Div-SP
```

to get a frequency of 14 Hz ($\frac{88.0525MHz}{6289464} = 14Hz$). The current value of prescaler 1 is 8805 ($\frac{88.0525MHz}{8805} = 10KHz$). Just run:

```
caput RM01-PCIE-EVR1:PSX-Div-SP *integer*
```

to modify the value of any of the prescalers, and thus its frequency.

Bibliography

[1] MRF Technical Reference. *Event System with Delay Compensation Technical Reference Firmware 0205*, April 26, 2016.