

# Introduction to the Timing System

April 14, 2015

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Quick start</b>	<b>2</b>
<b>3</b>	<b>Generate a pulse upon receiving an event</b>	<b>3</b>
3.1	Instructions: . . . . .	4
3.2	Substitution snippet explanation: . . . . .	5
<b>4</b>	<b>Process an EPICS record upon receiving an event</b>	<b>5</b>
4.1	Instructions: . . . . .	5
4.2	Substitution snippet explanation: . . . . .	6
<b>5</b>	<b>Generate a clock signal</b>	<b>6</b>
5.1	Instructions: . . . . .	6
5.2	Substitution snippet explanation: . . . . .	7
<b>6</b>	<b>Output a Distributed Bus bit</b>	<b>7</b>
6.1	Instructions: . . . . .	8
6.2	Substitution snippet explanation: . . . . .	8

# 1 Introduction

A timing system consists of an Event Generator (EVG), a series of Event Receivers (EVR), software controlling them and a timing network. EVG generates a series of events, which are delivered to EVRs through a timing network. An EVR is then configured to respond to specific events in various ways, including processing EPICS records, generating pulses, synchronized clock or custom signals on its outputs. This document contains step-by-step instructions to configuring some of the basic functionalities of the Event Receiver.

## 2 Quick start

To set up a timing system we need a VME crate, a Single Board Computer (SBC) and an EVR. A VME crate has a number of slots where SBC, EVR and other components can be inserted. Slot numbering should be checked with the VME crate documentation. To use a VME crate for the tutorials, check that:

- a VME64x IFC 1210 Single Board Computer is inserted into VME crate slot 1 (how to set up IFC 1210 [3]),
- an EVR is inserted in slot 2,
- the EVR is connected to the timing network through an optical cable.

To set up an IOC application for Event Receiver we need a startup script and a substitution file. The following steps demonstrate how to prepare such a SWIT compatible IOC application:

1. Create a **project folder**, eg. MTEST-VME-EVRTEST and a **sub-folder named** *cfg* in your project folder: MTEST-VME-EVRTEST/cfg/

```
mkdir MTEST-VME-EVRTEST
cd MTEST-VME-EVRTEST
mkdir cfg
```

2. Copy the **startup script** to your project. Use the following command in your project folder:

```
cp /fin/dev1/iocBoot/templates/slejko/EVR_VME.startup
MTEST-VME-EVRTEST_startup.script
```

The startup script should look similar to

```
##System configuration
epicsEnvSet SYS MTEST-VME-EVRTEST

### EVR configuration
#epicsEnvSet EVR EVR0 ##EVR name (default EVR0)
#epicsEnvSet EVR_SLOT 2 ##EVR SLOT (default slot 2)
```

```
< $(TEMPLATE\DIR)/EVR_VME.startup
```

```
## END OF EVR configuration
```

System name (SYS) variable must be defined. When omitted, the EVR name and EVR\_SLOT will fall-back to default values. Using the above startup script, the system name is set to *MTEST-VME-EVRTEST*, and the Event Receiver named *EVR0* is placed in the physical slot 2 of the VME crate. Line `< $(TEMPLATE_DIR)/EVR_VME.startup` includes a generic EVR startup script, so it should not be changed by the user.

3. Copy the **substitution file** [2] to *MTEST-VME-EVRTEST/cfg/EVR.subs*. This substitution file can always be used as a starting point for new applications. Use the following command in your project folder:

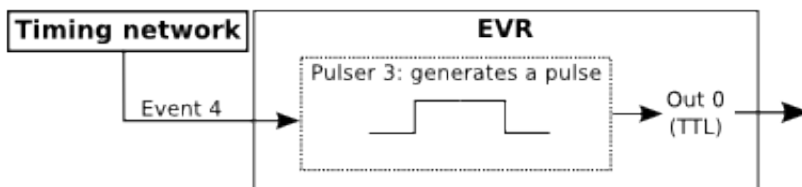
```
cp /fin/dev1/iocBoot/templates/slejko/EVR.subs cfg/
```

The macro definitions in the substitution file are used to configure the EVR. All the available macros are already present in the substitution file [2] and set to their default values, so the user can simply change the desired values, remove unused macros (same effect as leaving them with their default value). Detailed description of the substitution file is available in the EVR manual [1].

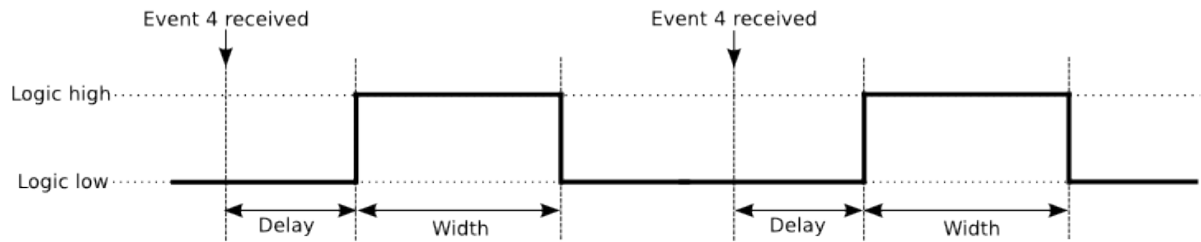
**Command summary** for creating a new IOC application:

```
mkdir MTEST-VME-EVRTEST
cd MTEST-VME-EVRTEST
mkdir cfg
cp /fin/dev1/iocBoot/templates/slejko/EVR_VME.startup
MTEST-VME-EVRTEST_startup.script
cp /fin/dev1/iocBoot/templates/slejko/EVR.subs cfg/
```

### 3 Generate a pulse uppon receiving an event



This tutorial demonstrates how to configure an EVR to generate a 80ns wide pulse, 40ns after each occurrence of event 4 and output it through the front panel TTL output 0(FrontOut0). In order to achieve this, pulser 3 delay and width are configured by setting macro value of *Pul3-Delay-SP* to 40 and of *Pul3-Width-SP* to 80. The value of the output source macro *FrontOut0-Src-SP* is set to 3, which configures the front panel output 0(FrontOut0) to use pulser 3 as its source. Finally pulser 3 is set to trigger on event 4 by setting macro value *EVT* to 4 in *evr-pulserMap.template*.



### 3.1 Instructions:

1. Create a **project folder**, eg. `MTEST-VME-EVRTEST`.
2. Copy **startup script** and an example **substitution file** to your project. Use the following commands in your project folder:

```
cp /fin/devl/iocBoot/templates/slejko/EVR_VME.startup
    MTEST-VME-EVRTEST_startup.script
cp /fin/devl/iocBoot/templates/slejko/EVR.subs cfg/
```

3. Set the macro values in the substitution file according to this snippet:

```
file "$(TEMPLATE_DIR)/evr-vmerf230.template"
{
    {
        ...
        Pul3-Delay-SP=40,
        Pul3-Width-SP=80,
        ...
        FrontOut0-Src-SP=3,
        ...
    }
}

file "$(TEMPLATE_DIR)/evr-pulserMap.template"{
pattern { PID    F,          EVT, ID}
        ...
        { 3,     Trig,      4,    0 }
        ...
}
```

4. Optionally, you can remove all the macros whose values you did not change.
5. **Install the prepared IOC** by running command `swit -V` from your project folder `MTEST-VME-EVRTEST`.

## 3.2 Substitution snippet explanation:

First we set up the pulse generator:

- **Pul3-Delay-SP** [ns]: Set the delay from when the event occurs to the start of the pulse (pulse rising edge) for pulser 3.
- **Pul3-Width-SP** [ns]: Set the pulse width (time between the pulse rising and falling edge) for pulser 3.

Next, we use the macro **FrontOut0-Src-SP** to set the source of the front panel output 0 to pulser 3 through a mapping. Mappings 0-15 correspond to pulsers 0-15. A complete list of mappings is available in the EVR manual [1].

Finally set the Pulser 3 to trigger on reception of the event 4:

- **PID**: Select Pulser 3
- **F**: Select the *Trigger* function of the pulser
- **EVT**: Map Pulser 3 Trig function to event 4
- **ID**: Unique ID for each PID-F combination.

In order to use different pulser simply change the pulser number, eg. using **Pul5-Delay-SP** instead of **Pul3-Delay-SP** sets the delay of pulser 5 instead of pulser 3. Similar is for outputs, eg. using **FrontOut1-Src-SP** instead of **FrontOut0-Src-SP** sets the output source signal of front panel output 1 instead of front panel output 0.

## 4 Process an EPICS record upon receiving an event

This tutorial demonstrates how to use macro substitutions to trigger an EPICS event number 1 and increase a counter upon each reception of event 1 from the timing network.

The counter is a calc record, that counts how many times the event 1 was received from the timing system. Its name is in form of  $\$(SYS)-\$(EVR):Event-\$(EVT)-Cnt-I$ , where  $\$(SYS)$  represents the system name (set in the startup script),  $\$(EVR)$  represents the Event Receiver name (set in the startup script, default "EVR0") and  $\$(EVT)$  is the event from the timing network (in this case "1"), thus the full **name of the counter record** is **MTEST-VME-EVRTEST-EVR0:Event-1-Cnt-I**.

### 4.1 Instructions:

1. Create a **project folder**, eg. **MTEST-VME-EVRTEST**.
2. Copy **startup script** and an example **substitution file** to your project. Use the following commands in your project folder:

```
cp /fin/dev1/iocBoot/templates/slejko/EVR_VME.startup
MTEST-VME-EVRTEST_startup.script
cp /fin/dev1/iocBoot/templates/slejko/EVR.subs cfg/
```

3. Set the macro values in the substitution file according to this snippet:

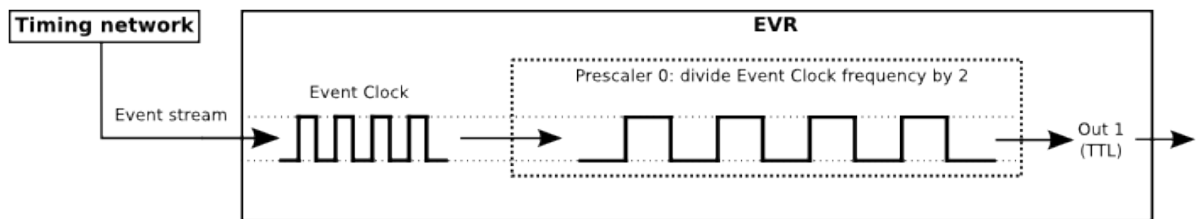
```
file "${TEMPLATE_DIR}/evr-softEvent.template"{
pattern { EVT,      CODE }
        { "1",      "1"}
        ...
}
```

4. Optionally, you can remove all the macros whose values you did not change.
5. **Install the prepared IOC** by running command `swit -V` from your project folder *MTEST-VME-EVRTEST*.

## 4.2 Substitution snippet explanation:

- **EVT**: Set to event received from timing network (hardware).
- **CODE**: Set to EPICS event number (software). It is suggested that macros *EVT* and *CODE* are set to the same value for simplicity, although this is not mandatory.

## 5 Generate a clock signal



Event Receivers have synchronized event clock across the timing system (the same phase and frequency). The event clock can be prescaled and mapped to the EVR output. In order to achieve this, the macro `PS0-Div-SP` is set to 2, which configures the prescaler 0(PS0) to divide the event clock frequency by 2. Then the value of the output source macro `FrontOut1-Src-SP` is set to 40, which configures the front panel output 1(FrontOut1) to use prescaler 0 as its source.

### 5.1 Instructions:

1. Create a **project folder**, eg. *MTEST-VME-EVRTEST*.
2. Copy **startup script** and an example **substitution file** to your project. Use the following commands in your project folder:

```
cp /fin/devl/iocBoot/templates/slejko/EVR_VME.startup
MTEST-VME-EVRTEST_startup.script
cp /fin/devl/iocBoot/templates/slejko/EVR.subs cfg/
```

3. Set the macro values in the substitution file according to this snippet:

```

file "$(TEMPLATE_DIR)/evr-vmerf230.template"
{
  {
    ...
    PS0-Div-SP=2,
    ...
    FrontOut1-Src-SP=40,
    ...
  }
}

```

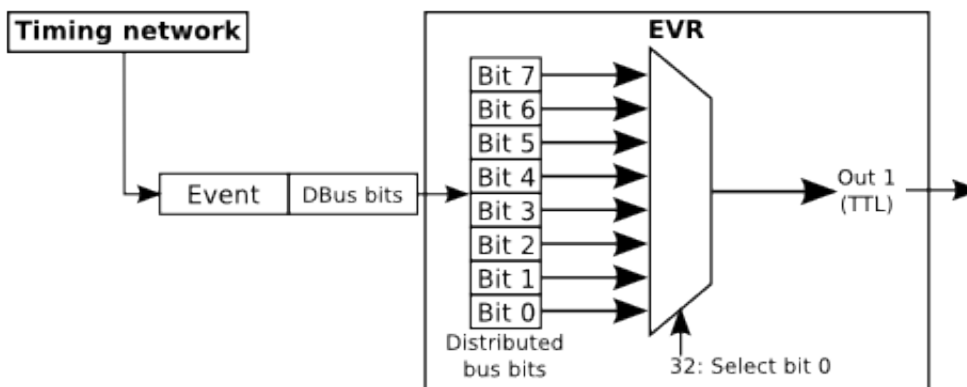
4. Optionally, you can remove all the macros whose values you did not change.
5. **Install the prepared IOC** by running command `swit -V` from your project folder *MTEST-VME-EVRTEST*.

## 5.2 Substitution snippet explanation:

- **PS0-Div-SP**: Set the Prescaler 0 to divide event clock frequency by 2.
- **FrontOut1-Src-SP**: Set the source of the Front Panel Output 1 to Prescaler 0 through a mapping. Mappings 40-42 correspond to prescalers 0-2. A complete list of mappings is available in the EVR manual [1].

In order to use different prescaler, simply change the prescaler number, eg. using `PS2-Div-SP` instead of `PS0-Div-SP` sets the divider of prescaler 2 instead of prescaler 0. Similar is for outputs, eg. using `FrontOut0-Src-SP` instead of `FrontOut1-Src-SP` sets the output source signal of front panel output 0 instead of front panel output 1.

## 6 Output a Distributed Bus bit



This tutorial demonstrates how to set up the DBus bit as a source of an EVR front panel output. To achieve this, the front panel output 0 should be mapped to the distributed bus bit (signal) 0. This is configured by setting the value of the output source macro `FrontOut1-Src-Sp` to 32. Mapping 32 corresponds to the DBus bit 0.

## 6.1 Instructions:

1. Create a **project folder**, eg. `MTEST-VME-EVRTEST`.
2. Copy **startup script** and an example **substitution file** to your project. Use the following commands in your project folder:

```
cp /fin/dev1/iocBoot/templates/slejko/EVR_VME.startup
MTEST-VME-EVRTEST_startup.script
cp /fin/dev1/iocBoot/templates/slejko/EVR.subs cfg/
```

3. Set the macro values in the substitution file according to this snippet:

```
file "$(TEMPLATE_DIR)/evr-vmerf230.template"
{
    {
        ...
        FrontOut1-Src-SP=32,
        ...
    }
}
```

4. Optionally, you can remove all the macros whose values you did not change.
5. **Install the prepared IOC** by running command `swit -V` from your project folder *MTEST-VME-EVRTEST*.

## 6.2 Substitution snippet explanation:

- **FrontOut1-Src-SP**: Set the source of the front panel output 1 to DBus bit 0 through a mapping. Mappings 32-39 correspond to DBus bits 0-7. A complete list of mappings is available in the EVR manual [1].

In order to use different front panel output, simply change the front panel output number, eg. using `FrontOut0-Src-SP` instead of `FrontOut1-Src-SP` sets the output source signal of front panel output 0 instead of front panel output 1.

## References

- [1] Evr manual. [https://github.psi.ch/projects/ED/repos/mrfioc2/browse/documentation/PSI/evr\\_manual.pdf](https://github.psi.ch/projects/ED/repos/mrfioc2/browse/documentation/PSI/evr_manual.pdf).
- [2] Substitution file. [https://github.psi.ch/projects/ED/repos/mrfioc2/browse/PSI/evr\\_ex.subs?raw](https://github.psi.ch/projects/ED/repos/mrfioc2/browse/PSI/evr_ex.subs?raw).
- [3] PSI. How to set up IFC 1210. <https://controls.web.psi.ch/cgi-bin/twiki/view/Main/HowToSetupIFC1210ioc>.