

---

# EMCOR EVR

---

## Test Plan

Revision:	0.2
Status:	Released
Repository:	Share Point
Project:	MR023 ICD – EMCOR EVR
Folder:	<a href="#">SlacSpace</a> -> <a href="#">Instrumentation and Controls Division</a> -> <a href="#">Special Projects</a> -> AIP: MCOR Controller
Document ID:	
File:	EMCOR EVR Test Plan.docx
Owner:	
Last modification:	February 26, 2016
Created:	August 4, 2014

Prepared by	Reviewed by	Approved by
Tomo Cesnik	Briant Lam	

## Document History

---

Revision	Date	Changed/reviewed	Section(s)	Modification
0.1	2015-03-17	tcesnik		Initial version.
0.2	2015-07-01	tcesnik	3	Changed device file names.

---

## Confidentiality

---

This document is classified as a public document. As such, it or parts thereof are openly accessible to anyone listed in the Audience section, either in electronic or in any other form.

## Scope

---

This document defines the Test Plan for the EMCOR Linux kernel driver and EPICS device support that uses this kernel driver. It is concerned mainly with the implementation of the interrupts and memory mapping of the kernel driver.

## Audience

---

The audience of this document is the test engineers who will execute the Test Plan and produce the Test Report.



## Table of Contents

<b>1. Introduction</b>	<b>5</b>
1.1. Overview.....	5
<b>2. Test Environment</b>	<b>6</b>
2.1. Hardware .....	6
2.2. Software .....	6
<b>3. Kernel Module Tests</b>	<b>7</b>
3.1. Test FPGA reading [TP-EKM-0001].....	7
3.2. Test software IRQ [TP-EKM-0002].....	7
3.3. Test EVR IRQ [TP-EKM-0003].....	8
<b>4. EPICS Tests</b>	<b>10</b>
4.1. Test software IRQ [TP-EE-0001].....	10
4.2. Test EVR IRQ [TP-EE-0002].....	10

## Figures

No table of figures entries found.

## Tables

Table 1: List of hardware in the test environment .....	6
Table 2: List of software in the test environment.....	6

## Glossary of Terms

BAR	Base Address Register
EPICS	Experimental Physics and Industrial Control System
EVR	Event Receiver
FPGA	Field-Programmable Gate Array
IOC	Input / Output Controller
IRQ	Interrupt Request
NFS	Network File System
RT	Real Time



---

## References

---



# 1. Introduction

---

## 1.1. Overview

---

The EMCOR controller board contains an FPGA, which contains EVR logic for the timing system as well as the magnet controlling functionality. For the user to properly receive the interrupts from the subpart that he/she is interested in, special Linux kernel module has been developed. The module takes care of making the interrupts independent from one another through selectively masking the interrupt control register and providing the interrupts to the user through two separate file descriptors. The test plan provides tests for checking if the kernel driver works as specified and if the interrupts can be successfully received by the user. There are two types of tests:

- Tests that check the driver interface with the usage of simple C programs.
- Tests that use EPICS and also check the EPICS modules, which have been additionally modified to support the new kernel driver.

## 2. Test Environment

---

All tests in this document are performed on the SLAC premises with the test setup described in the following chapters.

### 2.1. Hardware

---

Table 1: List of hardware in the test environment

ID	Hardware item	Location
eioc-b34-mg08	EMCOR board	B34 MG08

---

The EMCOR board must be connected to the working timing system, which produces fiducial events.

### 2.2. Software

---

Table 2: List of software in the test environment

Software component	Version	Where installed
Linux RT	3.14.12	EMCOR board
EMCOR kernel module	1.1.0	EMCOR board
EPICS	3.14.12	Available on NFS
EPICS Event module	4.1.7	Available on NFS
evrLab	1.0	Available on NFS
EPICS drvPciMcor module	1.0.2	Available on NFS

---

## 3. Kernel Module Tests

All kernel module tests use simple programs that are part of the kernel module's software suite. They are available in sub-directory tools of the kernel module software unit.

To perform the tests, login on the EMCOR controller board and move to the directory that contains the aforementioned tools.

### 3.1. Test FPGA reading [TP-EKM-0001]

#### Description

Checks that the data can be read from the firmware.

#### Pre-requisites

#### Test procedure

1. Read the device name from BAR0:

```
./mmap /dev/mcor $((0x05C8))  
READ (0x000005C8): 0x524F434D
```

2. Read the EVR firmware version from BAR2:

```
./mmap /dev/era3 $((0x002C))  
READ (0x0000002C): 0x1F000001
```

**Expected result:** The output should be the same as specified.

**Test result:**  Not tested  Passed  Failed

**Comment:**

### 3.2. Test software IRQ [TP-EKM-0002]

#### Description

Tests that MCOR interrupts are triggered and can be received by the user.

#### Pre-requisites

#### Test procedure

1. Read interrupt file descriptor (blocks the terminal):

```
./read /dev/mcor
```

2. Open new terminal and generate software IRQ:

```
./mmap /dev/mcor $((0x0690)) $((0xFFFFFFFF))  
WRITE (0x00000690): 0xFFFFFFFF
```

```
READ (0x00000690): 0x00000000
```

3. Check that the read in the first terminal still blocks and no output has been produced.
4. Enable interrupts:

```
./write /dev/mcor $((0xFFFFFFFF))
```

5. Generate software IRQ:

```
./mmap /dev/mcor $((0x0690)) $((0xFFFFFFFF))
WRITE (0x00000690): 0xFFFFFFFF
READ (0x00000690): 0x00000000
```

6. Check that the read in the first terminal returned and produced the following output:

```
Read: 0x00000100
```

**Expected result:** When interrupts are disabled, there should be no interrupt received. When interrupts are enabled, the interrupt should be received by the user.

### Test variation

Use poll instead of blocking read in the first step:

```
./poll /dev/mcor
```

Test result:  Not tested  Passed  Failed

Comment:

## 3.3. Test EVR IRQ [TP-EKM-0003]

### Description

Tests if interrupts from EVR can be received by the user.

### Pre-requisites

### Test procedure

1. Read interrupt file descriptor (blocks the terminal):

```
./read /dev/era3
```

2. Open new terminal and enable EVR:

```
./mmap /dev/era3 $((0x0004)) $((0x80000000))
WRITE (0x00000004): 0x80000000
READ (0x00000004): 0x80000000
```

3. Set EVR to receive interrupts:



```
./mmap /dev/era3 $((0x000C)) $((0xFFFFFFFF))  
WRITE (0x0000000C): 0xFFFFFFFF  
READ (0x0000000C): 0xFFFFFFFF
```

4. Acknowledge possible old EVR interrupts:

```
./mmap /dev/era3 $((0x0008)) $((0xFFFFFFFF))  
WRITE (0x00000008): 0xFFFFFFFF  
READ (0x00000008): 0x80000000
```

5. Enable EVR interrupts:

```
./write /dev/era3 $((0xFFFFFFFF))
```

6. Enable data buffer receiving:

```
./mmap /dev/era3 $((0x0020)) $((0x0000D000))  
WRITE (0x00000020): 0x0000D000  
READ (0x00000020): 0x00001035
```

7. Check that the read in the first terminal returned and produced the following output:

```
Read: 0x00000001
```

8. Read EVR interrupt status register:

```
./mmap /dev/era3 $((0x0008))  
READ (0x00000008): 0x80000020
```

**Expected result:** The interrupt should be received by the user and the EVR interrupt status register should be set appropriately.

### Test variation

Use poll instead of blocking read in the first step:

```
./poll /dev/era3
```

Test result:  Not tested  Passed  Failed

Comment:

## 4. EPICS Tests

EPICS tests need certain EPICS modules compiled and available on the NFS.

To perform the tests, login on the EMCOR controller board and move to the directory that contains scripts for running the EPICS IOCs:

```
cd eioc-b34-mg08
```

### 4.1. Test software IRQ [TP-EE-0001]

#### Description

Tests that MCOR software interrupts are triggered and can be received by EPICS application.

#### Pre-requisites

#### Test procedure

1. Start EPICS Event IOC:

```
./startupConsole-ioc.cmd
```

2. Open new console and generate software IRQ:

```
./mmap /dev/mcor $((0x0690)) $((0xFFFFFFFF))  
WRITE (0x00000690): 0xFFFFFFFF  
READ (0x00000690): 0x00000000
```

3. Check that in EPICS console, there is a printout:

```
Status: 0x00000100
```

**Expected result:** There should be printout in the EPICS console that corresponds to the software interrupt status.

**Test result:**  Not tested  Passed  Failed

**Comment:**

### 4.2. Test EVR IRQ [TP-EE-0002]

#### Description

Tests if interrupts from EVR can be received by the user.

#### Pre-requisites

#### Test procedure

1. Start EPICS Event IOC:

```
./startupConsole-ev99.cmd
```

- Open new console and check the status of interrupts:

```
watch -n1 cat /proc/interrupts
16:      257391          0  IO-APIC-fasteoi  uhci_hcd:usb5, emcor
```

**Expected result:** The interrupt count of the interrupts that are affected by the EMCOR kernel driver should increase by at least 360 (fiducial) in each second as the time goes by.

- Additionally check the evrLab screens for Rx Link status and IRQ rates.

**Expected result:** The Rx Link status should be ON if there is optical connection and the IRQ rates should show interrupt numbers.

Test result:  Not tested  Passed  Failed

Comment: