# Introduction

## Purpose

The Software Requirements Specification defines and describes the requirements of the EPD ICU Software of the SRG Mission. The requirements described in this document are derived from [AD1](#AD1).

## Scope

This document applies to the EPD ICU Software of the SRG Mission. It does not provide information for the Ground Support Software (GSS), which is maintained separately as part of the Electronic Ground Support Equipment (EGSE). This document supplies information applicable to the Software Requirements Specification.

# Applicable Documents

AD1 SRG-EPD Software System Specification. SRG-EPD-SSS-0001

AD2 SRG-EPD TM-TC Interface Control Document. SRG-EPD-ICD-0001

AD3 Software. ECSS-E-ST-40C. 06/03/2009

AD4 Software Product Assurance. ECSS-Q-ST-80C. 06/03/2009

AD5 SRG-EPD Software Product Assurance Plan. SRG-EPD-SPA-0001

AD5 SRG-EPD Software Development Plan. SRG-EPD-SDP-0001

# Reference Documents

RD1 SRG-EPD Software System Specification. SRG-EPD-SSS-0001

RD2 Ground systems and operations - Telemetry and telecommand packet utilization. ECSS-E-ST-70-41C. 15/04/2016

RD3 SpaceWire - Links, nodes, routers and networks. ECSS-E-ST-50-12A. 24/01/2003

RD4 SpaceWire - CCSDS packet transfer protocol. ECSS-E-ST-50-53C

RD5 Space engineering. Electric and electronic. ECSS-E-20B issue 8 rev 0. 03/2007

RD6 Spacecraft discrete interfaces. ECSS-E-ST-50-14C

# Terms, Definitions and Abbreviations

AD Applicable Document

ASW Application Software

BSW Boot Software

C&DH Control and Data Handling

CDPU Common Data Processing Unit

EGSE Electrical Ground Support Equipment

EPD Energetic Particle Detector

GSS Ground Support System

ICU Instrument Control Unit

IRQ Interrupt Request

FPGA Field Programmable Gate Array

LVPS Low Voltage Power Supply

OBDH On-Board Data Handling

RAM Random Access Memory

RD Reference Document

S/C Spacecraft

SDRAM Synchronous Dynamic RAM

SVVP Field Programmable Gate Array

SW Software Verification and Validation Plan

TBC To Be Confirmed

TC Telecommand

TM Telemetry

# Software Overview

## Function Purpose

EPD ICU Software purpose is to provide support to EPD sensors. Each of the EPD sensors has some data processing capabilities either by software running in the sensor DPUs/microcontrollers or by logic functions. The main functionality, however, is running in the ICU. The principal functions of the ICU software are:

* To collect scientific data from the instruments periodically 1 Hz.
* To accumulate high-cadence science data from the sensors to specified lower time resolution packages.
* To collect housekeeping data from the four units.
* To time tag the data if necessary.
* To perform lossless compression of the data if necessary.
* To format the data for transmission by the S/C.
* To accept and syntactically check TC from the S/C.
* To deliver these commands to the individual sensors for execution.
* To accept from the ground and distribute to the sensors tables and software.
* To handle the burst mode.

A Real-Time Operating System is proposed to provide the basic facilities such as multitasking, timing and semaphore/mutex synchronization. A high-level software development tool (UML2, ROOM or similar) could be employed in the software design process.

All software will comply with the following specification, namely:

1. Different areas of memory will be employed to store code, fixed constants and variable parameters of the in-flight software.
2. The PROM memory will include minimum boot software able to manage EEPROM updates and hardware test functions.
3. Independent software module memory updates will be supported.
4. Instrument safety hazard will be avoided in the software design process.
5. On-board software will acquire, store and telemeter the employed resources, including memory usage, central processor unit (CPU) usage and I/O usage.
6. If a processor reset is detected, an event report will be generated.
7. If a processor overload condition is detected, an event report will be generated.
8. If an unexpected arithmetic overflow condition is detected, an event report will be generated.
9. If an illegal program instruction is encountered during execution of a program code, an event report will be generated.
10. If a data bus error is detected by software, an event report will be generated.
11. If a memory corruption is detected by an EDAC, an event report will be generated.
12. If a checksum error is detected, an event report will be generated.
13. A modularized structure will be employed in the software design process.

### ICU Software Modes

In [ICU\_SW\_modes](#ICU_SW_modes) the modes of operation of SRG EPD ICUSW are represented.

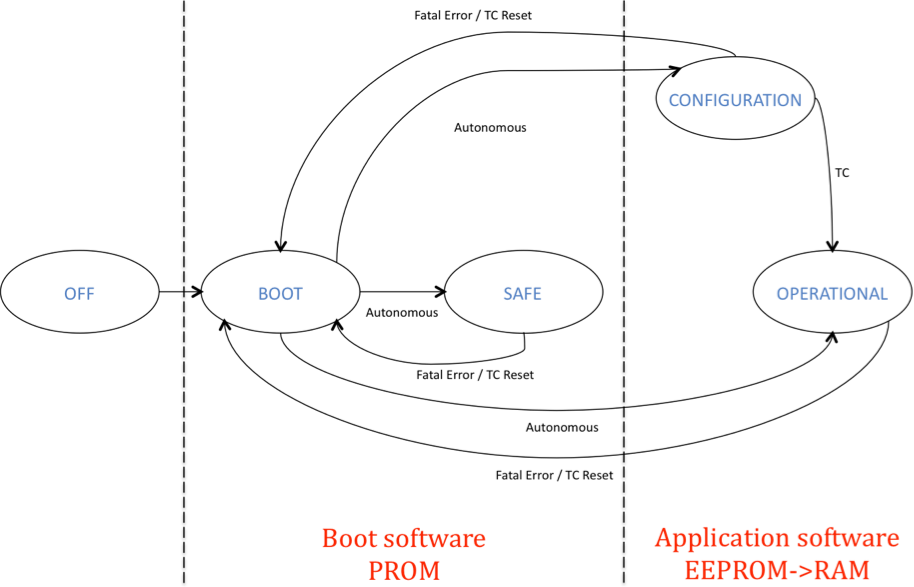


Figure 1: ICU SW operation modes

The ICUSW shall be able to support the following modes:

* OFF: The ICU is not powered.
* BOOT: The ICU is just switched on, a soft reset from other mode has been performed, or a mode transition TC has been received. This mode is in charge of the initial ICU configuration and checking. It also tries to load the application software and pass control to it. The boot software of the ICUSW shall implement the BOOT mode.
* SAFE: is a mode in which EPD is powered but in a safe configuration, which can be maintained indefinitely (e.g. main control unit powered but sensors are off). In this mode EPD will generate only non-science telemetry. The configuration of the instrument in SAFE mode will be unambiguous (i.e. only one defined configuration). The boot software of the ICUSW shall implement the SAFE mode, and it will trigger a transition from BOOT mode to SAFE mode when application software cannot be executed. In SAFE mode the damaged application software images stored in EEPROM can be restored by means of patch telecommands.
* CONFIGURATION: this mode shall be used for required operations to activate or configure the instrument. In this mode non-science data may be generated, depending on the instrument configuration (e.g. sensors activated or not).
* OPERATIONAL: this mode is the mode in which EPD is fully operative and generates non-science as well as science data.

Autonomous transitions are the following:

* BOOT to CONFIGURATION. This transition is triggered autonomously after a reset when application software is correctly loaded, the target mode is CONFIGURATION, and the control is passed to it.
* BOOT to OPERATIONAL. This transition is triggered autonomously after a reset when application software is correctly loaded, the target mode is OPERATIONAL, and the control is passed to it.
* BOOT to SAFE. This transition is triggered when application software cannot be loaded and the boot software remains in control of the ICU.
* SAFE to BOOT. This transition is triggered by a soft reset (fatal error detected such as bus error, memory access error, etc.) or a TC.
* CONFIGURATION to BOOT. This transition is triggered by a soft reset (fatal error detected such as bus error, memory access error, etc.), by a recovery action (minor error detected such as driver or controller error etc.) or a TC.
* OPERATIONAL to BOOT. This transition is triggered by a soft reset (fatal error detected such as bus error, memory access error, etc.), by a recovery action (minor error detected such as driver or controller error etc.) or a TC.

## Environmental Considerations

This section provides an overview of the key hardware components. This section is intended for information only, and does not describe all of the details of the various items.

### General Description

[ICU\_IF](#ICU_IF) illustrates the top-level interfaces that affect the EPD Instrument Control Unit Software.

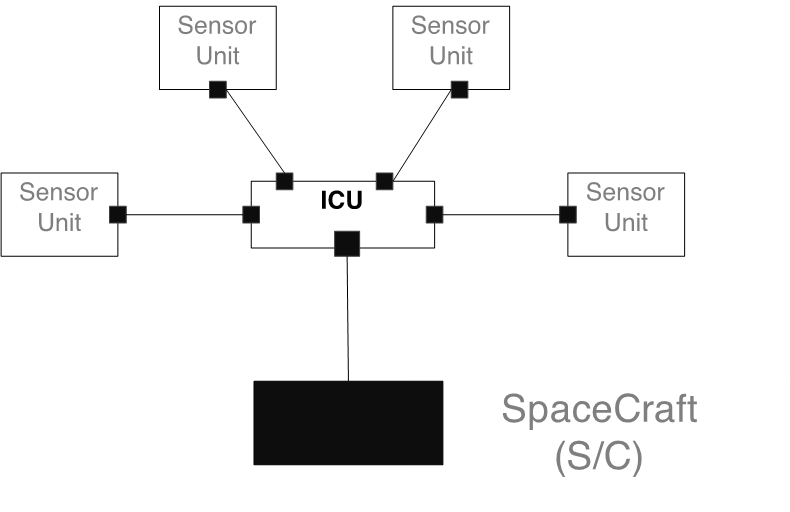


Figure 2: ICU interfaces

EPD ICU provides the interface between the SRG Spacecraft C&DH system and the EPD sensors. All information transfer between the EPD sensors and the Spacecraft/Ground flows through the ICU, including telemetry, commands and status. The EPD sensors communicate with the EPD ICU over a dedicated serial interface.

### Hardware Overview

The EPD ICU is composed by the CDPU and the LVPS. Both units are redundant, working in a cold configuration. Thus we have the CDPU/LVPS nominal unit and the CDPU/LVPS redundant unit. According to the cold redundancy configuration pattern, if one unit is switched ON, the other must be OFF. [ICU\_HW\_schema](#ICU_HW_schema) shows the basic HW diagram of the EPD's ICU.

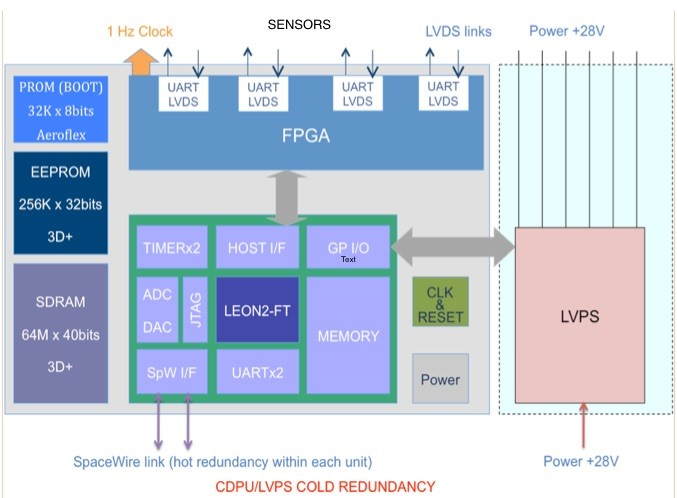


Figure 3: ICU hardware schema

As it can be observed, the design is based on a FPGA that includes a LEON2 soft processor, the UART-LVDS links with the sensors, the SpaceWire cores and the interfaces with the LVPS and the memory.

## Relation Other Systems

### User Interfaces

The EPD Instrument has no direct in-flight human user interfaces. The Mission Operation Center (MOC) manages all human interaction with EPD.

### Hardware Interfaces

The detailed EPD's ICU hardware to software interfaces are shown in [ICU\_HW\_IFs](#ICU_HW_IFs).

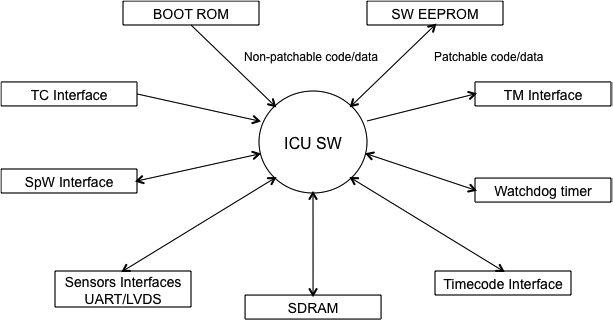


Figure 4: ICU hardware interfaces

### External Software Interfaces

SRG EPD's ICU uses the RTEMS operating system to perform task control functions and provide some resource management services.

ICU is planning on using the following capabilities of RTEMS:

A detailed list of each used function is provided in the SRG EPD ICU Software Detailed Design Document.

* Prioritized, preemptive task scheduler.
* Inter-process communication.
* Resource locks (semaphores).
* Timers.

### Communication Interfaces

#### Sensor Interface

All EPD sensors commands and telemetry are communicated via UART protocol at 115200 bauds using Low-Voltage Differential Signalling (LVDS).

#### S/C Interface

Each CDPU is a remote terminal on the spacecraft SpaceWire link. The implementation will be compliant up to the RD2 and RD3. Two separate SpaceWire links in a hot redundant configuration will be employed. SpaceWire is the command and data interface with the S/C.

EPD command messages are contained in CCSDS telecommand packets. EPD CDPU strips off the CCSDS packet headers and analyzes the destination identifier. If the identifier is associated to the CDPU, the telecommand is processed directly by the CDPU. If the identifier is associated to any sensor, the CDPU forward the command to the specific sensor. Command responses from the sensor are sent to the CDPU that transmits them to the S/C in CCSDS telemetry packets format. Command responses generated by the CDPU itself are also CCSDS formatted and transmitted to the S/C.

The list of telecommands and telecommands responses are defined in AD2.

Transfer of data and command response from CDPU to the S/C shall take place according to the CCSDS recommendations defined in RD1 and RD2.

#### LVPS Interface

The interface between the CDPU and the LVPS is an internal interface. It is used to switch ON/OFF the sensors and to monitor voltages and currents.

### Time

The Orbiter DMS maintains a time code pattern as Spacecraft Elapsed Time (SCET). The SCET is considered as the Central Time Reference (CTR) and it is used as on-board reference for the correlation of time on Ground. The Central Time Reference (CTR) is maintained at spacecraft level and distributed to the Instruments in order to synchronize instruments and allow instruments to time-stamp their telemetry packets. The CDPU receives the time from the S/C.

## Constraints

N/A

# Requirements

In the following paragraphs a lists of software requirements will be included. Every single requirement has the following format:

| Name | SRS-ICUSW XX-YYYY |
| Description | Requirement Description |
| Validation | Validation Approach |
| Parent | Requirement Traceability |

**SRS-ICUSW XX-YYYY:** Identifies the requirement. XX is the type of requirement and YYYY is the number used to identify the requirement. Types of requirements are:

* **GE.** General
* **FU.** Functional
* **PE.** Performance
* **IF.** Interface
* **OP.** Operational
* **RS.** Resources
* **DI.** Design and Implementation Constraints
* **SE.** Security
* **PO.** Portability
* **QU.** Quality
* **RE.** Reliability
* **MA.** Maintainability
* **SA.** Safety
* **CD.** Configuration and delivery
* **DD.** Data definition and database
* **AI.** Adaptation and installation

Examples:

* SRS-ICUSW FU-0025
* SRS-ICUSW PE R-0037

**Description:** is the text that describes the requirement.

**Validation approach:** establishes how the requirement is verified:

* **R.** Review
* **A.** Analysis
* **I.** Inspection
* **T.** Test
* **S.** Similarity

**Parent:** contains the list of "software specification" that are cover by the requirement.

For each requirement type, subsections are provided when requirements apply only to boot software or to application software or they are common.

Other subsections or extended descriptions can be also added in order to better organize and clarify the requirements.

Extended descriptions will be added using **D:**. Example:

D: The ICU ASW shall handle the telecommand and telemetry according to the following requirements:

## General Requirements

### Common requirements

|  |  |
| --- | --- |
| **Name** | **SRS-ICUSW-GE-010** |
| **Description** | The ICU Boot Software (BSW) Development Process shall be compliant to the software standard [ECSS-E-ST-40C](#AD3). |
| **Validation Method** | Review |
| **Parent** | SSS-ICUSW-GE-010 |

|  |  |
| --- | --- |
| **Name** | **SRS-ICUSW-GE-020** |
| **Description** | The ICU Application Software (ASW) Development Process shall be compliant to the software standard [ECSS-E-ST-40C](#AD3). |
| **Validation Method** | Review |
| **Parent** | SSS-ICUSW-GE-010 |

## Functional Requirements

### ASW requirements

D: The ICU ASW shall handle the telecommand and telemetry according to the following requirements:

|  |  |
| --- | --- |
| **Name** | **SRS-ICUSW-FU-010** |
| **Description** | The ICU-ASW shall store the received telecommands in a FIFO queue |
| **Validation Method** | Review |
| **Parent** | SSS-ICUSW-CA-010 |

## Performance Requirements

### ASW requirements

D: The ICU ASW shall handle the received telecommands according to the following requirements:

|  |  |
| --- | --- |
| **Name** | **SRS-ICUSW-PE-010** |
| **Description** | The ICU-ASW shall retrieve the stored telecommands from the FIFO queue and accept them within an interval of 1 second after its reception. |
| **Validation Method** | Analysis |
| **Parent** | SSS-ICUSW-CA-010 |

## Interface Requirements

### ASW requirements

D: The ICUSW will handle the Telecommand Verification Service 1 according to the following requirements:

|  |  |
| --- | --- |
| **Name** | **SRS-ICUSW-IF-010** |
| **Description** | The ICU-ASW shall generate a PUS TM (1.1) as result of the telecommand acceptance. |
| **Validation Method** | Testing |
| **Parent** | SSS-ICUSW-IF-010 |

|  |  |
| --- | --- |
| **Name** | **SRS-ICUSW-IF-020** |
| **Description** | The ICU-ASW shall generate a PUS TM (1.2) as result of the telecommand rejection. |
| **Validation Method** | Testing |
| **Parent** | SSS-ICUSW-IF-010 |

## Operational Requirements

## Resources Requirements

## Design Requirements

## Security Privacy Requirements

## Portability Requirements

## SW Quality Requirements

## SW Reliability Requirements

## SW Maintainability Requirements

## SW Safety Requirements

## SW Configuration Delivery Requirements

## Data Definition DB Requirements

## Human Factors Requirements

## Adaptation Installation Requirements

# Logical Models