

ICE Cubes TM/TC Protocol

ICE Cubes

Title : ICE Cubes TM/TC Protocol

Abstract : This document describes the ICE Cubes TM/TC protocol format used to exchange data packets between the Experiment Cubes, the ICF and the ground Mission Control System (MCS).

Grant Agreement N° : 666815



**International Commercial
Experiments Service**

ICE Cubes TM/TC Protocol

Table of Contents

1	Introduction	1
1.1	Purpose and Scope	1
1.2	Applicable Documents	1
1.3	Reference Documents	1
1.4	Terms, definitions and abbreviated terms	1
2	ICE Cubes TM/TC Protocol.....	3
2.1	Overview	3
2.2	Packet header	3
2.3	User data	4
2.4	Definition of TM/TC Packets in a Mission Database	5
2.5	TM/TC Packets Examples	5
2.5.1	Simple Packets	5
2.5.2	TM Packet with Inheritance	6

List of Figures

No table of figures entries found.

List of Tables

No table of figures entries found.

ICE Cubes TM/TC Protocol

1 Introduction

1.1 Purpose and Scope

This document describes the TM/TC protocol used by the Experiment Cubes and the ICE Cubes Facility (ICF) to exchange telemetry (TM) and telecommand (TC) with the ground Mission Control System (MCS).

This document is intended for the use of the Experiment Cubes developers who desire to use the ICF TM/TC service and to the ICF administrators.

The ICE Cubes TM/TC protocol is used to transport:

- telemetry data from the ICE Cubes Facility (ICF) to the ground MCS
- telemetry data from the Experiment Cubes to the ground MCS
- telecommands from the ground MCS to the ICF
- telecommands from the ground MCS to the Experiment Cubes.

Telemetry data can include data such as housekeeping (HK) parameters, H&S data or on-event reports.

The purpose of the ICE Cubes TM/TC protocol is to provide a TM/TC protocol simpler to implement than the CCSDS Space Packet Protocol, and to generate TM/TC data files easier to read for a final user.

1.2 Applicable Documents

- AD1 Space Applications Services – ICE Cubes System Requirements Specification, ICU-SA-RQ-001, Version 1.4.0, 30-Jan-2017
- AD2 IETF – Common Format and MIME Type for Comma-Separated Values (CSV) Files, RFC 4180, Oct-2005

1.3 Reference Documents

- RD1 ISO – Data elements and interchange formats -- Information interchange -- Representation of dates and times, ISO 8601:2004, 01-Dec-2004
- RD2 CCSDS – XML Telemetric and Command Exchange (XTCE), CCSDS 660.0-B-1, Issue 1, Oct-2007
- RD3 Space Applications Services – Yamcs Server Manual, YAMCS-SA-MA-001, 08-Dec-2015

1.4 Terms, definitions and abbreviated terms

AD	Applicable Document
CCSDS	Consultative Committee for Space Data Systems
CR/LF	Carriage Return Line Feed
CSV	Comma-Separated Values
HK	Housekeeping
H&S	Health and Status
ICD	Interface Control Document
ICE Cubes	International Commercial Experiment Cubes

ICE Cubes TM/TC Protocol

ICF	ICE Cubes Facility
ICMCC	ICE Cubes Mission Control Centre
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
MCS	Mission Control System
MDB	Mission Database
MS-DOS	Microsoft Disk Operating System
OMG	Object Management Group
PDR	Preliminary Design Review
RD	Reference Document
RID	Review Item Discrepancy
TC	Telecommand
TM	Telemetry
UHB	User Home Base
US-ASCII	American Standard Code for Information Interchange
UTC	Coordinated Universal Time
XML	Extensible Markup Language
XTCE	XML Telemetric and Command Exchange

ICE Cubes TM/TC Protocol

2 ICE Cubes TM/TC Protocol

2.1 Overview

The ICE Cubes TM/TC packet format shall follow the CSV format compliant with the RFC 4180 (AD2). This includes the following requirements:

Field delimiter	<ul style="list-style-type: none"> - Any field may be quoted (with double quotes). - Fields containing a LF, a double-quote, and/or commas should be quoted. - A double quote character in a field must be represented by two double quote characters.
Fields separator	Comma
End of line	<ul style="list-style-type: none"> - MS-DOS-style lines that end with (CR/LF) characters (optional for the last line). - The CR/LF sequence is not allowed inside a field.

One line of the CSV file corresponds to one ICE Cubes TM/TC packet. No header line is supported nor required.

Additionally, each line of the CSV shall follow the format:

```
<Packet header>,<User data>
```

TM/TC packets are usually sent:

- Periodically, with a 1Hz or 0.1Hz frequency for housekeeping parameters
- On request, in response to telecommands or special events.

2.2 Packet header

The packet header shall be composed of the following mandatory fields:

```
<Timestamp>,<Packet counter>,<TM|TC>,<Origin ID>,<Destination ID>,<Packet type>
```

The table below describes each of the header fields:

Component	Description	Example
<Timestamp>	Generation time of the packet on the format ISO 8601 (RD1) up to the millisecond, in UTC. The system building the TM/TC packet is responsible for creating this timestamp.	2015-10-27T10:06:25.269

ICE Cubes TM/TC Protocol

Component	Description	Example
<Packet Counter>	Unsigned integer in the range [0, 2147483647] The packet counter shall be incremented by the sender in each packet occurrence. This allows to check for data loss. There shall be one packet counter per packet type.	0
<TM TC>	String that identifies if the line is a telemetry or a telecommand item. Possible values are: -TM -TC	TM
<Origin ID>	Integer value that identifies the subsystem sending the data packet. For TM packets, possible values are: 1: ICF 101 – 121: Cubes 1 to Cube 21 For TC packets, possible values are: 2: ICMCC 201 – 220: UHB 1 to UHB 20	2
<Destination ID>	Integer value that identifies the destination subsystem of a command. For TM packets, possible values are: 2: ICMCC 201 – 220: UHB 1 to UHB 20 For TC packets, possible values are: 1: ICF 101 – 121: Cubes 1 to Cube 21	1
<Packet type>	Unsigned integer in the range [0, 2147483647] that identifies the packet type. Possible packet types will be defined in the ICF ICD and in each Experiment Cube ICD.	3

2.3 User data

The user data section of the TM/TC packets shall be composed of a series of CSV fields:

<value 1>, <value 2>, ..., <value n>

ICE Cubes TM/TC Protocol

This section lists the datatypes accepted in for the fields <value>. Each field shall follow the RFC 4180 (AD2) as described in section 2.1.

The type of each field shall match the expected types of the TM/TC packets of the destination system for the current packet type. The TM/TC packet types and associated data formats of each system shall be defined in the respective systems' ICDs (ICF ICD, Experiment Cubes ICDs).

Type	Format	Example
Integer	Decimal in range [-]	1234560
Integer	Hexadecimal, prefix with "x"	x123456A
Float	Decimal separator is a dot.	145.125
String	Any characters of US-ASCII set	"some ""example"" data"
Enumeration	Enumeration shall be encoded as unsigned integers. The calibration from integer to full string is performed by the destination system.	0

2.4 Definition of TM/TC Packets in a Mission Database

The semantics of the packets' user data for each packet type shall be described in the relevant systems' ICD.

The definition of the packets shall then be implemented on board the systems and in the Mission Database (MDB) of the ground MCS.

In addition to the interpretation of the TC and TM values, the MDB of the MCS can be completed with definitions such as:

- limits and alarms on TM parameters values
- telemetry expected refresh rate
- computation of derived parameters from TM parameters by custom algorithms
- significance and constraint check on TC before the command release
- etc.

The complete set of the MDB features is described in the XTCE standard (RD2) and in the Yamcs manual (RD3).

2.5 TM/TC Packets Examples

2.5.1 Simple Packets

This example shows a TM packet and a TC packet, highlighting the packet headers and the user data parts. The user data part meaning is not defined.

With the following legend:

Packet header, user data

TM packet:

ICE Cubes TM/TC Protocol

```
2015-10-27T10:06:25.269,21,TM,1,0,3,123456,"some ""interesting""
data",y,bpoZ2Y=,0.1
```

TC packet:

```
"2015-10-27T10:06:25.269",45,TC,2,1,4,"an interesting command",0
```

2.5.2 TM Packet with Inheritance

In this example, the TM packet's user data fields have the following semantic:

Field	Type	Description
Value 1	String	Firmware version
Value 2	Integer	Subsystem ID: -1 = electric board: the following user data fields will report the electric board status. -2 = camera: the following user data fields will report the camera status.

For subsystem electric board (value 2 = 1):

Value 3	Float	Current voltage
Value 4	Float	Current power
Value 5	Float	Current temperature

For subsystem camera (value 2 = 2):

Value 3	String	Camera firmware version
Value 4	Integer	Current ISO setting
Value 5	Float	Current aperture
Value 6	Integer	Shutter speed (milliseconds)

TM packets examples:

```
2015-10-27T10:06:25.000,0,TM,101,2,1,"FW rev1.1",1,5.1,0.2,55
2015-10-27T10:06:26.000,1,TM,101,2,1,"FW rev1.1",2,"cam2.0",800,2.5,100
```


ICE Cubes TM/TC Protocol

```
2015-10-27T10:06:27.000,2,TM,101,2,1,"FW rev1.1",1,5.1,0.2,55
2015-10-27T10:06:28.000,3,TM,101,2,1,"FW rev1.1",2,"cam2.0",800,2.5,100
2015-10-27T10:06:29.000,4,TM,101,2,1,"FW rev1.1",1,5.1,0.2,55
2015-10-27T10:06:30.000,5,TM,101,2,1,"FW rev1.1",2,"cam2.0",800,2.5,100
```

- ⇒ On the 27th Oct 2015, telemetry is produced by Experiment Cube #1 and received on the ground MCS. The Experiment Cube produces a packet of type “1”.
- ⇒ The Cube #1 electric board is configured to 5.1 Volts, 0.2 Amp and has a temperature of 55 degrees.
- ⇒ The Cube #1 camera is configured to 800 ISO, an aperture of 2.5, and a shutter speed of 100 milliseconds.

Note: in this example, the Cube #1 produces only one type of packet (packet type = 1), to generate telemetry for the two subsystem. Other solutions may have been to:

- have a different packet type for each of the subsystems;
- embed all subsystem’s values in a same packet instance, using 9 user values.

TC Example

TC packet type = 11: adjust camera ISO setting:

Field	Type	Description
Value 1	Integer	New camera ISO setting [100-25000]

```
2015-10-27T10:06:29.000,1,TC,201,101,11,1000
2015-10-27T10:06:39.000,2,TC,201,101,11,800
```

Those telecommands are sent from UHB #1 to Experiment Cube #1, adjusting ISO setting to 1000 and then 800