ITESRE	AGILE

DOCUMENT TYPE: INTERFACE CONTROL DOCUMENT

TITLE: INTERFACE CONTROL DOCUMENT OF THE AGILE MINICALORIMETER SCIENCE CONSOLE FOR THE TEST BEAM CAMPAIGN (MAY 2000) AT CERN

DOCUMENT No. AGILE-ITE-ICD-001 **PAGE:** I of iv,13

PROJECT Ref.: AGILE GSE

ISSUE No.: Rev. 1.0/DRAFT 2.0 **DATE:** APRIL 2000
Te.S.R.E. Report 276/00

PREPARED BY: MASSIMO TRIFOGLIO
FULVIO GIANOTTI
JOHN B. STEPHEN

CHECKED BY: MASSIMO TRIFOGLIO

DISTRIBUTION LIST

NAME	DEPT.	e-mail address	
Labanti Claudio	ITE	labanti@tesre.bo.cnr.it;	
Prest Michela	ITS	Michela.Prest@trieste.infn.it;	
Gianotti Fulvio	ITE	gianotti@tesre.bo.cnr.it;	
John B. Stephen	ITE	stephen@tesre.bo.cnr.it;	
Trifoglio Massimo	ITE	trifoglio@tesre.bo.cnr.it;	

CHANGE RECORD

Issue 1.0/Draft 1.0	First draft circulated within TESRE and INFN/Trieste, on 10 April 2000.
Issue 1.0/Draft 2.0	Draft circulated within TESRE and INFN/Trieste on 12 April 2000.

Note: Text modifications respect to the previous version are identified by a vertical bar on the right side

TABLE OF CONTENTS

1. Introduction

1.1. Purpose and Scope

The main purpose of the present note is to define the protocols and the data formats to be used in order to interface the Minicalorimeter Science Console (MCAL SC), provided by ITESRE/CNR, with the Test Equipment, provided by INFN – Trieste, during the test campaign to be carried out at CERN in May 2000 in the framework of the AGILE instrument prototyping.

The document is organised as follows:

- section 2 gives an overview of the Test Scenario, which configuration is sketched in Annex A;
- section 3 defines the structure and the contents of the Data Packets to be produced by the Main Console and to be sent to the MCAL SC; details on the format of each event data is given in Annex B;
- section 4 present the operational environment concerning the data link and the data exchange among the Main Console and the MCAL SC;
- annex C deals with the format of the data produced by the MCAL SC itself

The proposal takes into accounts further enhancements which will be required in the future in order to support the AGILE instrument development.

As well, the proposal allows the re-usage of some of the test software already developed by ITESRE for other instruments (EPIC-XMM, IBIS-INTEGRAL, PICsIT-IBIS).

1.2. Reference Documents

[1] "Packet Telemetry Standard", ESA PSS-04-106 Issue 1, January 1988.

2 Test Scenario

The test set up for the Test Campaign at CERN is depicted in Annex 1, where:

- the Main Console will interface to the instruments in order to conduct the test and to acquire the data;

- for each detected event, the Main Console acquires a set of data from all the instruments, namely:
  - Cherenkov: 1 bit flag;
  - Monitor 1: (X,Y);
  - Telescope 1: (X,Y);
  - Telescope 2: (X,Y);
  - Tracker TBD;
  - Telescope 3: (X,Y);
  - Monitor 2: (X,Y);
  - Minicalorimeter bars: 4 PHA
- from each of this data set, the Main Console will extract the data which are relevant to the Minicalorimeter data processing and creates an *MCAL event data* format (as given Annex B);

- with the *MCAL event data*, the Main Console will prepare an *MCAL Data Packet* of fixed length having the structure complying with of the ESA Telemetry Standard [1]; each Data Packet will be capable of containing a fixed number of event data;

- as soon as an MCAL Data Packet is filled with the event data or the *time-out* is expired since the last packet has been sent, the Data Packet will be sent by the Main Console to the MCAL Science Console;

- the MCAL Science Console will acquire and archive the Data Packets in one set of files for each measurement;

- in near real time the MCAL Science Console will provide a graphical representation on a percentage of the data (Quick Look).

The format of the MCAL Data Packets is detailed in section 3.

3 Packet Format

3.1 Packet Structure

With the data acquired from the instrument chain, the Main Console generates packets having the same basic structure of the ESA Packet TM.

Each packet is of fixed length of 486 bytes and consists of a Packet Header (6 bytes) followed by a Packet Data field (480 bytes).

Each packet is sent to the SC in one TCP/IP message containing an additional prefix of two bytes which contain the total number of bytes contained in the packet, specified in binary format (big-endian). I.e. 0x01E6 (486 decimal).

Data words belonging to the same event cannot be split into two different packets.

The part of the Source Data Field which does not contain event data is filled with dummy bytes.

The value of the dummy byte is configurable; by default it will be set to 0xFF.

3.1.1 Packet header

The Packet Header will have the following format:

Packet Header (3x16-bit word)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MSB															LSB
Version Number			Type	DHFH	APID										
SF		Source Sequence Counter													
Packet Length															

Where:

First word:

Version number: must be set to '100' binary

Type: must be set to '0' binary

DFHF: Data Field Header Flag, must be set to '1' binary

APID: Application Process Identifier, must be set to 0x50D (decimal 1293)

Hence: this word must be set to the fixed value: 0x8D0D

Second word:

SF: Sequence Flag, must be set to '11' bin

Source Sequence Counter: counts the packets of the above APID

Third word:

Packet Length: [number of octects in Packet Data Field] – 1;

Hence: this word must be set to the fixed value: 0x01DF~~1E5~~ (~~485~~ 479 decimal).

3.1.2 Packet Data Field

The Packet Data Field will consists of the Data Field Header (4 byte) and the Source Data Field (476 byte).

The former will have the following format:

Data Field Header (2 x 16-bit word)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MSB															LSB
Time Tag (MSB)															
Time Tag (LSB)															

Where:

Time Tag: provides a time tag info in format TBD, representing the time measured by the Main Console CPU clock when the first event was written to the packet.

The Source Data Field will be have the following format:

Source Data Field (14 x 17 16-bit word record)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MSB															LSB
Source Data Field (word 1)															
…															
Source Data Field (word 238~~17~~)															

I.e. the Source Data Field will be capable of containing up to 14 MCAL event Data, each having the format detailed in Annex B.

4 Operational Environment

At start up a TCP/IP link (**Data Link**) will be established on the LAN among the Main Console (Client) and the MCAL Science Console (Server).

The Data Link will be kept along the whole test session.

In case the Data Link drops, automatically the Server restarts waiting for a new connection request from the Client.

The data taking from the instrument will be under control of the Main Console operator.

By using local commands and without interfering with the Main Console, at any time the Science Console operator will be able to close all the files related to the current measurement and to open a new set of files where all the forthcoming packets will be archived as a new measurement.

Annex A Test set up configurations

A.1 Test Equipment Hardware Configuration

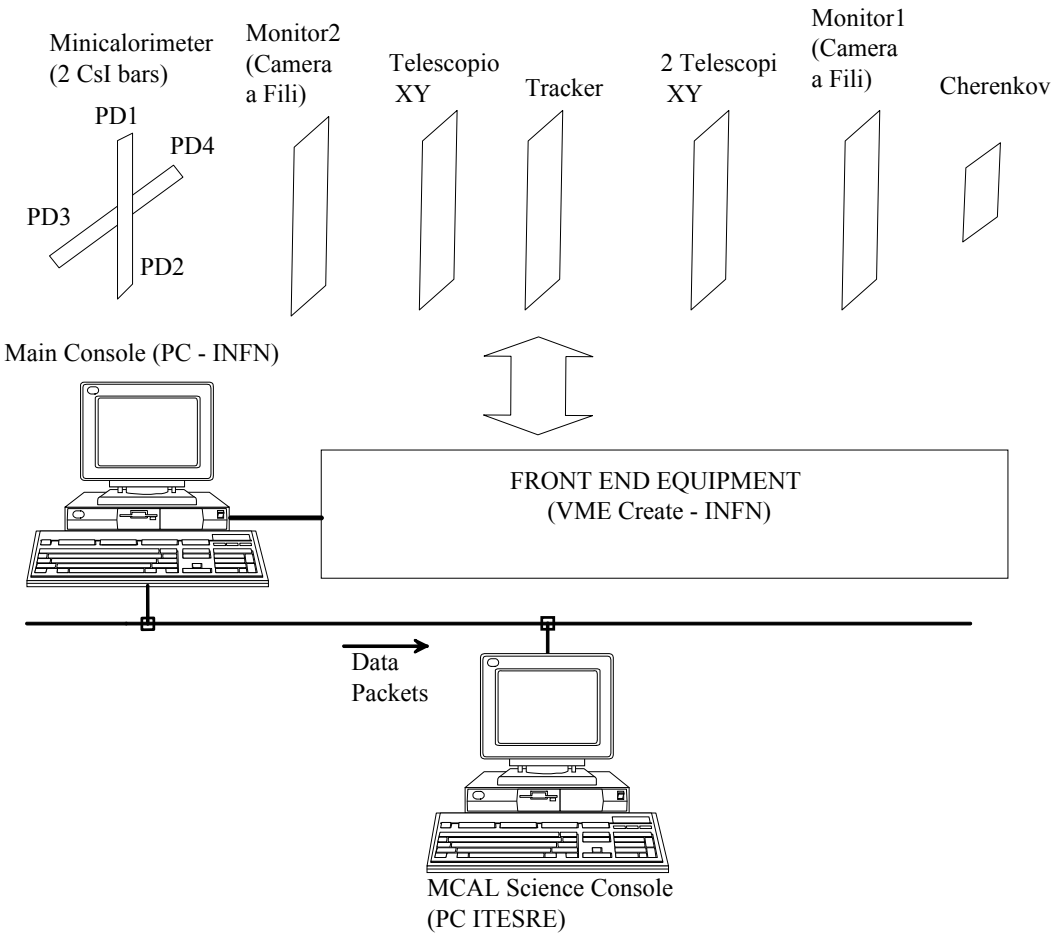

fig. A.1-1 Test set up configuration

The above diagram sketches the configuration of the test set up foreseen for the Test Beam Campaign. As far as the Test Equipment is concerned, it is noted that:

- The Main Console, through a dedicated I/F, is connected to the FEE which allows the PC to acquire the data produced by all the instruments;

- The Main Console and the MCAL Science Console are connected to a 100BasetT LAN through a Mini HUB having at least 4 100BaseT RJ ports.

- The Main Console sends to the MCAL Science Console a subset of the instrument data through a TCP/IP link established on the LAN.

Annex B Event Data format

For each detected pulse event, the Main Console adds 34 bytes in the Packet Data Field. The format of each event data set is shown below.

Event Data Format (17x16 bit word)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MSB															LSB
0	MCAL - PD1 Energy (TBD bits) [horizontal right]														
0	MCAL – PD2 Energy (TBD bits)) [horizontal left]														
0	MCAL – PD3 Energy (TBD bits) [vertical top]														
0	MCAL – PD4 Energy (TBD bits) [vertical bottom]														
0	Monitor 1 - X coordinate (TBD bits) [horizontal]														
0	Monitor 1 - Y coordinate (TBD bits) [vertical]														
0	Monitor 2 - X coordinate (TBD bits) [horizontal]														
0	Monitor 2 - Y coordinate (TBD bits) [vertical]														
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Flag
0	Telescope 1 - X coordinate (TBD bits)														
0	Telescope 1 - Y coordinate (TBD bits) [vertical]														
0	Telescope 2 - X coordinate (TBD bits) [horizontal]														
0	Telescope 2 - Y coordinate (TBD bits) [vertical]														
0	Telescope 3 - X coordinate (TBD bits) [horizontal]														
0	Telescope 3 - Y coordinate (TBD bits) [vertical]														
0	Tracker - X coordinate (TBD bits)														
0	Tracker PHA (TBD bits)														

Where:

Flag Cherenkov flag, which gives the event type: 0=electron, 1=muone;

The remaining are self-explaining.

Annex C Science Console Data Formats

C.1 T103a Packets

C.1.1 FITS files format

No FITS files will be produced.

C.1.2 Quick Look shared memory data format

The parameters defining the dimensions of the Quick Look shared memory (SHMF) and the buffer BUFFER to be written on it, is set as follows:

```
C from parameters.inc
C
C           SIZE_FRAME was set to 4224 to be multiply by:
C                   100 ( buffers per SHMF channel)
C                   21  ( total number of SHMF channels)
C                   ---> total of 8.870.400 byte of RAM
C          N.B.: on 25/01/2000 changed from 100 a 300
C
C   BLHEADER  is the header length in bytes
C
C   BLEVENT  is the event length  in bytes
C   for the time being, for each event it is reserved the maximum
C   space as required by the Compton events.
C
C   MAXEV    is the max number of events per frame which can be
C           written in each SHMF buffer; this is also the number of events
C           which are buffered in the FITS arrays.
C           Note: in the SHMF buffer are written: 1 header+1frame+1header
C
      INTEGER*4 BLEVENT
      PARAMETER (BLEVENT=40)
C
      INTEGER*4 BLBUFFER, BLHEADER, MAXEV, MAXCELL
      PARAMETER (BLBUFFER=4224, BLHEADER=40,
     +          MAXEV=( (BLBUFFER-2*BLHEADER)/(BLEVENT) ) /2 ,
     +          MAXCELL=848)
C
```

The following format is applied to the structure which contains the header and the events. Depending on the number of events, one or more of this structure can fit into the buffer BUFFER located in the SHMF shared memory.

```
C from PUTFRAME
C
      INTEGER*2 HEADER(BLHEADER/2), BUFFER(BLBUFFER/2)
C
```

```
C     HEADER  is the array where the header info are prepared
C
C     word  1: BLHEADER, header length in bytes
C     word  2: PKT_APID, APID
C     word  3: SUBTYPE
C     word  4: SHM channel
C     word  5: NEVENT, number of events contained in the frame
C     word  6: BLEVENT, event length in bytes
C     word  7: SPARE
C     word  8-9: NFRAME, frame #
C     word 10-16: used by Histog.f
C     word 17-18: RUNID
C     word 19-20: spare
C
C     Each event consists of the following info:
C
C     word  1-4:  TIME (8 bytes)
C     word  5:      MCAL PD1 Energy (2bytes)
C     word  6:      MCAL PD2 Energy (2bytes)
C     word  7:      MCAL PD3 Energy (2bytes)
C     word  8:      MCAL PD4 Energy (2bytes)
C     word  9:      Monitor 1 - X (2bytes)
C     word  10:    Monitor 1 – Y (2 bytes)
C     word  11     Monitor 2 - X (2bytes)
C     word  12     Monitor 2 - Y (2bytes)
C     word  13     Cherenkov Flag (2 bytes)
C     word  14-20  Spare (for the time being, the remaining event info are not processed by the QL)



C
C
C     BUFFER is the buffer where HEADER + events are
C           saved to the Shared Memory SHMF when the current
C           frame does not fit into it
C
```

It is noted that only a subset of the event data will be processed by the Quick Look.