AGILE	IASF RM

DOCUMENT TYPE:	**TECHNICAL NOTE**
TITLE:	**SAIE TE pseudo ESA TM packets**
DOCUMENT No.:	AGILE-IAS-TN-023 **No. OF PAGES:** 35
ISSUE No.:	1 **DATE:** 23/5/2005
PREPARED BY:	L. PACCIANI, G. PORROVECCHIO
CHECKED BY:	M. TRIGOGLIO, F. GIANOTTI, A. BULGARELLI
APPROVED BY:	M. FEROCI **DATE:** 23/5/2005

CHANGE RECORD

Issue	Date	Page	Description of Changes	Release
1	23/05/2005		First issue related to SAIE TE PSEUDO ESA TM PACKETS	1

Table of Contents

TABLE OF CONTENTS......3

RULES......5

1 SAIE TE PSEUDO ESA TM PACKETS......7

1.1 PACKET HEADER......8
1.2 PACKET DATA FIELD......9
1.2.1 Data Field for <SCI>and <CAL>......*11*
1.2.1.1 Data Field Header......11
1.2.1.2 <SCI> and <CAL> Source Data Field......11
1.2.1.2.1 Data Block Elements......12
1.2.1.2.1.1 SAIE TE Raw Data description......13
1.2.1.2.1.1.1 ABT event......13
1.2.1.2.1.1.2 Dummy event......13
1.2.1.2.1.1.3 Good event......13
1.2.1.2.1.1.4 Calibration event......15
1.2.1.2.1.2 Data Block Element format description......15
1.2.2 Data Field for <DHK>......*17*
1.2.2.1 Data Field Header for <DHK>......17
1.2.2.2 <DHK> Source Data Field......17
1.2.2.2.1 Data Block Elements......18
1.2.3 Data Field for <AHK>......*20*
1.2.3.1 Data Field Header for <AHK>......20
1.2.3.2 <AHK> Source Data Field......20
1.2.3.2.1 Data Block Elements for format version 0......21
1.2.3.2.2 Data Block Elements for format version 1......23
1.2.4 Data Field for <CONF>......*26*
1.2.4.1 SAIE TE Configuration File......26
1.2.4.2 Data Field Header for <CONF>......27
1.2.4.3 <CONF> Source Data Field......28
1.2.4.3.1 <CONF> Data Block Header......28
1.2.4.3.2 <CONF> Data Block elements......29
1.2.5 Data Field for <LOG>......*30*
1.2.5.1 Data Field Header for <LOG>......30

1.2.5.2 <LOG> Source Data Field....30
1.2.5.2.1 <LOG> Data Block Header....31
1.2.5.2.2 <LOG> Data Block Elements....31
1.2.6 Data Field for <TUT>....33
1.2.6.1 Data Field Header for <TUT>....33
1.2.6.2 <TUT> Source Data Field....33
BIBLIOGRAPHY....35

Rules

***RULE 0*..4**

Rule 1..7

Introduction

SuperAgile SW team chose to use the existing DISCoS [1] tool to perform data storage and quick look analysis for Test Equipments.

This Software was developed to be detector indipendent. It only requests data in a pseudo ESA TM format, a <START> and a <STOP> telecommand [2] corrisponding to *acquisition start* and *stop* respectively.

> ***RULE 0***
>
> *Data directly produced from SAIE TE are called "**Raw Data**",*
>
> *Raw Data translated in TM format are called "**TM Packets**" or "**TM Data**".*

Raw Data produced from Test Equipment doesn't match ESA TM.

In this document we establish the procedure to translate "**Raw Data**" in **TM Packets**" and the TM format.

Some constraint comes from PacketLib [3] (a library developed to handle packets) that we chose to use for TM data manipulation.

1 SAIE TE pseudo ESA TM packets

The quantum is a 16 bit Big Endian word,

Packets are almost in agreement with ESA standard for TM packets.

Every packet consists of

1. a fixed lenght Packet Header Field,
2. a Data Field. Its format depends on TM flavour, actually we foresee these TM packets:
   - Digital Housekeeping (<DHK>),
   - Analogue Housekeeping (<AHK>),
   - Scientific Data (<SCI>),
   - Calibration Data: (<CAL>),
   - Configuration (<CONF>),
   - Operator log (<RUNLOG>),
   - Universal Time (<TUT>).

1.1 Packet Header

MSB LSB

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CCOE															
Version Number			Type	DFHF	APID										
SF		Source Sequence Counter													
Packet Lenght															

CCOE: specifies the lenght of the following packet (bytes) This word is not in ESA standard TM format but is strongly suggested by Massimo Trifoglio and Fulvio Gianotti for compatibility with DISCoS tool. According to existing DISCoS version, the maximum packet lenght is 1024Bytes;

Version Number: 001 binary;

Type: 0 for TM (1 for TC);

DFHF: data field header flag, 1,

APID: Application Process Identifier,

APID for SAIE TE : 1297;
SF: Sequence Flag, 11 binary;
Source Sequence Counter: packet counter (one counter for every APID);

Packet Lenght: Data Field Lenght (bytes) –1.

1.2 Packet Data Field

This field format must match:

1. the incoming “Raw Data” format (based on 16 bit Little Endian Words),
2. the ESA Standard (based on 16 bit Big Endian Words),
3. the existing PacketLib (developed by A. Bulgarelli),
4. the existing DISCoS tool.

> ***Rule 1***
>
> *16 bit Little Endian Words from “incoming RAW DATA” are translated into 16 bit Big Endian Words (necessary for pseudo ESA TM) in the following way:*
>
> *Incoming 16 bits RAW DATA word is divided in 2 bites (MSB and LSB). The MSB and LSB are byte-swapped (for Little -> Big Endian Conversion).*

In the following, when we refer to 16 bit words in packets, we assume that these words have been translated using *rule 1*.

TM Data Fields are divided in

1. **Data Field Header**,
2. **Source Data Field**.

1. In the **Data Field Header** we put all the informations needed to handle the Source Data Field. To distinguish between different TM packets, in the Data Field Header, we reserve two fields to identify the **Packet Type** and the **Packet SubType**.

   We chose:

   <SCI> : 15/1

   <CAL> : 15/2

   <DHK> : 1/1

   <CONF> : 1/2

   <LOG> : 1/3

   <TUT> : 1/4

   <AHK> : 1/5

   for SAIE TE.

2. Usually the **Source Data Field** has an internal structure. Assuming to use the existing PacketLib, we can use:
   - the "PacketLib packet format scheme 1" [3] for <SCI>, <CAL>, <DHK> (i.e. *N* Data Blocks with a fixed number of elements each one),
   - the "PacketLib packet format scheme 2" [3] for <LOG>, <CONF> (i.e. *N* Data Blocks, each Data Block has an header and a variable number of elements),
   - the "PacketLib packet format scheme 3" [3] for <TUT> (i.e. One Data Block with a fixed number of elements each one).

1.2.1 Data Field for <SCI>and <CAL>

According to the “PacketLib packet format “scheme 1” we chose:

1.2.1.1 Data Field Header

MSB LSB

0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
Spare						Checksum flag		Packet Type				Packet SubType			
Time Tag second (32 bit integer signed MSB)															
Time Tag second (32 bit integer signed LSB)															
Time Tag millisecond (16 bit unsigned)															
Packet Format version															
Number of Words per block															
Number of events (blocks) in the packet															

CheckSum Flag: 00 binary, **TBD**;

Packet TYPE/SUBTYPE: they specify the packet type/subtype.

We chose:

<SCI> : 15/1

<CAL> : 15/2

for SAIE TE.

During a scientific Run, the SAIE generates ABT, Dummy and Good events.

During a Calibration Run, the SAIE generates Calibration events.

Time Tag: CPU System Time expressed as second and millisecond since 00:00 UTC january 1, 1970 (i.e. as gven by the C function ftime) corresponding to the creation of the packet (see [4]).

Packet Format Version: this field is filled with the current version for the format of the packet identified with APID/Type/SubType.

Number of words for block: It corresponds to the number of elements for each block. Each element is 2 Bytes long. The number of elements in the block is fixed to 5

Number of events (blocks) in the packet: It corresponds to the number of Data Blocks (one block for event) in the packet.

1.2.1.2 <SCI> and <CAL> Source Data Field

According to “PacketLib Packet Format scheme 1” [3], the Source Data Field is divided in Data

Blocks. Each Block has a fixed number of elements of fixed dimension.

There is a variable number of Data Blocks. In our case, there is a Data Block for every event.

1.2.1.2.1 Data Block Elements

Each block contains 5 elements. Each element is 2 Bytes long.

The SAIE generates 4 types of event:

1. ABT (absolute time) event,
2. Dummy event,
3. Good event,
4. Calibration event.

All of them are originally composed by 60 bits (i.e. 5 word of 12 bit) but the TE will use 5 word of 16 bit. Therefore the four most significant bits are always zero.

1.2.1.2.1.1 SAIE TE Raw Data description

The “Raw Data” coming from the SAIE

1.2.1.2.1.1.1 ABT event

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	**0**	**0**						AMP1	AMP0	DC2	DC1	DC0
0	0	0	0	DT11	DT10	DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
0	0	0	0	PPS15	PPS14	PPS13	PPS12	PPS11	PPS10	PPS9	PPS8	PPS7	PPS6	PPS5	PPS4
0	0	0	0	PPS3	PPS2	PPS1	PPS0	US19	US18	US17	US16	US15	US14	US13	US12
0	0	0	0	US11	US10	US9	US8	US7	US6	US5	US4	US3	US2	US1	US0

Where:

- **0,0** is the event type identifier,
- AMP1,AMP0 is the amplitude’s value of the calibration pulse,
- DC2,DC1,DC0 is the triggered daisy chain identifier,
- DT1,...,DT0 is the differential time,
- PPS15,...,PPS0 is the seconds’ counter,
- US19,...,US0 is the micro seconds’ counter.

1.2.1.2.1.1.2 Dummy event

Very similar to the ABT event. The only difference are the 2 bit in the first word concerning the event type identifier.

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	**0**	**1**						AMP1	AMP0	DC2	DC1	DC0
0	0	0	0	DT11	DT10	DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
0	0	0	0	PPS15	PPS14	PPS13	PPS12	PPS11	PPS10	PPS9	PPS8	PPS7	PPS6	PPS5	PPS4
0	0	0	0	PPS3	PPS2	PPS1	PPS0	US19	US18	US17	US16	US15	US14	US13	US12
0	0	0	0	US11	US10	US9	US8	US7	US6	US5	US4	US3	US2	US1	US0

Where:

- **0,1** is the event type identifier,
- AMP1,AMP0 is the amplitude’s value of the calibration pulse,
- DC2,DC1,DC0 is the triggered daisy chain identifier,
- DT1,...,DT0 is the differential time,
- PPS15,...,PPS0 is the seconds’ counter,
- US19,...,US0 is the micro seconds’ counter.

1.2.1.2.1.1.3 Good event

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	**1**	**0**						AMP1	AMP0	DC2	DC1	DC0

MSB LSB

0	0	0	0	DT11	DT10	DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
0	0	0	0	MAD11	MAD10	MAD9	MAD8	MAD7	MAD6	MAD5	MAD4	MAD3	MAD2	MAD1	MAD0
0	0	0	0	LAD11	LAD10	LAD9	LAD8	LAD7	LAD6	LAD5	LAD4	LAD3	LAD2	LAD1	LAD0
0	0	0	0	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Where:

- **1,0** is the event type identifier,
- AMP1,AMP0 is the amplitude's value of the calibration pulse,
- DC2,DC1,DC0 is the triggered daisy chain identifier,
- DT1,...,DT0 is the differential time,
- MAD11,...,MAD0 are the 12 most significant bit of the address of the triggered channel,
- LAD11,...,LAD0 are the 12 least significant bit of the address of the triggered channel,
- E11,...,E0 is the energy of the photon.

1.2.1.2.1.1.4 Calibration event

MSB LSB

15	*14*	*13*	*12*	*11*	*10*	*9*	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
0	0	0	0	**1**	**0**						AMP1	AMP0	DC2	DC1	DC0
0	0	0	0	RAD11	RAD10	RAD9	RAD8	RAD7	RAD6	RAD5	RAD4	RAD3	RAD2	RAD1	RAD0
0	0	0	0	MAD11	MAD10	MAD9	MAD8	MAD7	MAD6	MAD5	MAD4	MAD3	MAD2	MAD1	MAD0
0	0	0	0	LAD11	LAD10	LAD9	LAD8	LAD7	LAD6	LAD5	LAD4	LAD3	LAD2	LAD1	LAD0
0	0	0	0	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Where

- **1,0** is the event type identifier,
- AMP1,AMP0 is the amplitude's value of the calibration pulse,
- DC2,DC1,DC0 is the triggered daisy chain identifier,
- RAD11,...,RAD0 is the real strip address (i.e. the only channel enabled to receive the cal pulse),
- MAD11,...,MAD0 are the 12 most significant bit of the address of the triggered channel,
- LAD11,...,LAD0 are the 12 least significant bit of the address of the triggered channel,
- E11,...,E0 is the energy of the photon.

1.2.1.2.1.2 Data Block Element format description

To match DISCoS and PacketLIB issues, we decided to use this fields subdivision of the Raw data:

MSB LSB

15	*14*	*13*	*12*	*11*	*10*	*9*	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
Dummy				Event ID		Dummy					Pulse Ampl. ID		D.C. ID		
Dummy				Field 1											
Dummy				Field 2											
Dummy				Field 3.1				Field 3.2							
Dummy				Field 4											

Where:

Event ID identifies the event type, i.e.

00 binary for ABT events,

01 binary for Dummy events,

10 binary for Good events,

11 binary for Calibration events;

Pulse Ampl. ID specifies the Pulse Amplitude in Calibration events, while in the other kinds of events has no meaning;

D.C. ID identifies the daisy chain that gave the processed signal.

The meaning of the other fields depend on the Event Type:

- For “Good” events

  Field 1 is the field for Differential time,

  Field 2 is the field for the 12 most significant bits of the triggered channel,

  Field 3.1+3.2 are the fields for the 12 most significant bits of the triggered channel,

  Field 4 is the field for the response of the strip.

- For “Calibration” events

  Field 1 is the field for the real strip address,

  Field 2 is the field for the 12 most significant bits of the triggered channel,

  Field 3.1+3.2 are the fields for the 12 most significant bits of the triggered channel,

  Field 4 is the field for the response of the strip.

- For “ABT” and “Dummy” events

  Field 1 is the field for the Differential time,

  Field 2+3.1 are the field for the seconds’ counter,

  Field 3.2+4 are the fields for the micro seconds’ counter.

We remember that the words described here are in “Little Endian” format. They must be translated using **rule 1**, to obtain Block Data Elements of pseudo TM Packets (that are in “Big Endian” format).

1.2.2 Data Field for <DHK>

According to the "PacketLib packet format scheme 1" we chose:

1.2.2.1 Data Field Header for <DHK>

MSB LSB

0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
Spare						Checksum flag		Packet Type				Packet SubType			
Time Tag second (32 bit integer signed MSB)															
Time Tag second (32 bit integer signed LSB)															
Time Tag millisecond (16 bit unsigned)															
Packet Format Version															
Number of elements for each block															
Number of blocks in the packet															

CheckSum Flag: 00 binary TBD;

Packet TYPE/SUBTYPE: they specify the packet type/subtype.

We chose:

<DHK> : 1/1

for SAIE TE.

Time Tag: CPU System Time expressed as second and millisecond since 00:00 UTC january 1,1970 (i.e. as given by the C function ftime) corresponding to the creation of the packet (see [3]).

Number of Blocks in the packet: this is the number of DHK blocks in the packet (chosen to be 1).

Packet Format Version: this field is filled with the current version for the format of the packet identified with APID/Type/SubType.

Number of elements for each block: this is the number of DHK elements for each block (chosen to be 31) + 3 words with the time of dhk generation.

1.2.2.2 <DHK> Source Data Field

According to "PacketLib Packet Format scheme 1" [3], the Source Data Field is divided in Data Blocks. Each Block has a fixed number of elements (1) of fixed dimension.

There is a variable number of Data Blocks (in our case we fixed to 1).

In our case, there is a Data Block for every Digital HouseKeeping.

We chose to have 1 block of Housekeeping in a <DHK> (N=1).

1.2.2.2.1 Data Block Elements

Each block is composed of 37 element, each element is 2 Bytes long.

MSB LSB

0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
Time Tag second (32 bit integer signed MSB)															
Time Tag second (32 bit integer signed LSB)															
Time Tag millisecond (16 bit unsigned)															
Single hit DC 0															
Single hit DC 1															
Single hit DC 2															
Single hit DC 3															
Single hit DC 4															
Single hit DC 5															
Single hit DC 6															
Single hit DC 7															
Multi hit DC 0															
Multi hit DC 1															
Multi hit DC 2															
Multi hit DC 3															
Multi hit DC 4															
Multi hit DC 5															
Multi hit DC 6															
Multi hit DC 7															
Good Event DC 0															
Good Event DC 1															
Good Event DC 2															
Good Event DC 3															

MSB LSB

Good Event DC 4
Good Event DC 5
Good Event DC 6
Good Event DC 7
AC TOP rejected detector 0
AC TOP rejected detector 1
MCAL rejected detector 0
MCAL rejected detector 1
ST rejected detector 0
ST rejected detector 1
Energy rejected
DC supply status
Config FPGA status
Control FPGA status

1.2.3 Data Field for <AHK>

According to the "PacketLib packet format scheme 1" we chose:

1.2.3.1 Data Field Header for <AHK>

MSB LSB

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Spare						Checksum flag		Packet Type				Packet SubType			
Time Tag second (32 bit integer signed MSB)															
Time Tag second (32 bit integer signed LSB)															
Time Tag millisecond (16 bit unsigned)															
Packet Format Version															
Number of elements for each block															
Number of blocks in the packet															

CheckSum Flag: 00 binary TBD;

Packet TYPE/SUBTYPE: they specify the packet type/subtype.

We chose:

<AHK> : 1/5

for SAIE TE.

Time Tag: CPU System Time expressed as second and millisecond since 00:00 UTC january 1,1970 (i.e. as given by the C function ftime) corresponding to the creation of the packet (see [3]).

Number of Blocks in the packet: this is the number of HK blocks in the packet (chosen to be 1).

Packet Format Version: this field is filled with the current version for the format of the packet identified with APID/Type/SubType.

Number of elements for each block: this is the number of AHK elements for each block (chosen to be 64 words for version 0 and 64+3 words with the time of ahk generation for version 1).

1.2.3.2 <AHK> Source Data Field

According to "PacketLib Packet Format scheme 1" [3], the Source Data Field is divided in Data Blocks. Each Block has a fixed number of elements (1) of fixed dimension.

There is a variable number of Data Blocks (in our case we fixed to 1).

In our case, there is a Data Block for every Analogue HouseKeeping.

We chose to have 1 block of Analogue Housekeeping in a <AHK> (N=1).

1.2.3.2.1 *Data Block Elements for format version 0*

Each block is composed of 64 element, each element is 2 Bytes long. This version was used only at the very beginning, new TM data use version 1.

MSB LSB

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
AVCC															
DVCC_AHK															
AVCC_AHK															
AVEE_AHK															
XVDD_AHK															
XVSS_AHK															
SAIE_TEMP_1															
SAIE_TEMP_2															
S1_DET_V															
S1_DET_I															
S2_DET_V															
S2_DET_I															
S1_VCC_AHK															
S1_VEE_AHK															
S2_VCC_AHK															
S2_VEE_AHK															
S1_2VP_DC3															

MSB LSB

S1_2VP_DC2
S1_2VP_DC1
S1_2VP_DC0
S2_2VP_DC3
S2_2VP_DC2
S2_2VP_DC1
S2_2VP_DC0
S1_2VN_DC3
S1_2VN_DC2
S1_2VN_DC1
S1_2VN_DC0
S2_2VN_DC3
S2_2VN_DC2
S2_2VN_DC1
S2_2VN_DC0
S1_TEMP_AHK_7
S1_TEMP_AHK_6
S1_TEMP_AHK_5
S1_TEMP_AHK_4
S1_TEMP_AHK_3
S1_TEMP_AHK_2
S1_TEMP_AHK_1
S1_TEMP_AHK_0
S2_TEMP_AHK_7
S2_TEMP_AHK_6
S2_TEMP_AHK_5
S2_TEMP_AHK_4
S2_TEMP_AHK_3
S2_TEMP_AHK_2
S2_TEMP_AHK_1
S2_TEMP_AHK_0
AGND
AGND
AGND
AGND
AGND

MSB LSB

AGND
AGND
AGND
AGND
AGND
AGND
AGND
AGND
AGND
AGND
AGND

1.2.3.2.2 Data Block Elements for format version 1

Each block is composed of 64+3 element, each element is 2 Bytes long.

MSB LSB

0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
Time Tag second (32 bit integer signed MSB)															
Time Tag second (32 bit integer signed LSB)															
Time Tag millisecond (16 bit unsigned)															
AVCC															
DVCC_AHK															
AVCC_AHK															
AVEE_AHK															
XVDD_AHK															
XVSS_AHK															
SAIE_TEMP_1															
SAIE_TEMP_2															
S1_DET_V															
S1_DET_I															
S2_DET_V															
S2_DET_I															
S1_VCC_AHK															

MSB LSB

S1_VEE_AHK
S2_VCC_AHK
S2_VEE_AHK
S1_2VP_DC3
S1_2VP_DC2
S1_2VP_DC1
S1_2VP_DC0
S2_2VP_DC3
S2_2VP_DC2
S2_2VP_DC1
S2_2VP_DC0
S1_2VN_DC3
S1_2VN_DC2
S1_2VN_DC1
S1_2VN_DC0
S2_2VN_DC3
S2_2VN_DC2
S2_2VN_DC1
S2_2VN_DC0
S1_TEMP_AHK_7
S1_TEMP_AHK_6
S1_TEMP_AHK_5
S1_TEMP_AHK_4
S1_TEMP_AHK_3
S1_TEMP_AHK_2
S1_TEMP_AHK_1
S1_TEMP_AHK_0
S2_TEMP_AHK_7
S2_TEMP_AHK_6
S2_TEMP_AHK_5
S2_TEMP_AHK_4
S2_TEMP_AHK_3
S2_TEMP_AHK_2
S2_TEMP_AHK_1
S2_TEMP_AHK_0
AGND

MSB LSB

AGND
AGND
AGND
AGND
AGND
AGND
AGND
AGND
AGND
AGND
AGND
AGND
AGND
AGND
AGND

1.2.4 Data Field for <CONF>

1.2.4.1 SAIE TE Configuration File

The SAIE TE configuration File is composed by (1 word means a 16 bit long data)

- 12 Digital configurations of the chips in one SAFEE. Each digital configuration is 40 words long
- The digital configuration of the three SAFEE's DACs. Each DAC can control up to 8 outputs. The meaning of the DACs' outputs are described in the following table.
- The digital configuration of the three SAIE's DACs. Each DAC can control up to 8 outputs. The meaning of the DACs' outputs are described in the following table.
- A set of words concerning other SAIE TE parameters such as veto configuration, dc supplied etc.. (see the last rows in the following table)

Word n	Name	Description	
0	Word 0 Register in XAA1 0	Digital Configuration chip 0	
.....	....		
39	Word 39 Register in XAA1 0		
40	Word 0 Register in XAA1 1	Digital Configuration chip 1	
....	....		
79	Word 39 Register in XAA1 1		
.....	....	....	
440	Word 0 Register in XAA1 11	Digital Configuration chip 11	
......	......		
479	Word 39 Register in XAA1 11		
480	Analog Thr DC 0	DAC 1 SAFEE	OUT A
481	Analog Thr DC 1		OUT E
482	Analog Thr DC 2	DAC 2 SAFEE	OUT A
483	Analog Thr DC 3		OUT E
484	VFS DC 0	DAC 1 SAFEE	OUT B
485	VFS DC 1		OUT F
486	VFS DC 2	DAC 2 SAFEE	OUT B
487	VFS DC 3		OUT F
488	VFP DC 0	DAC 1 SAFEE	OUT C
489	VFP DC 1		OUT G
490	VFP DC 2	DAC 2 SAFEE	OUT C
491	VFP DC 3		OUT G
492	VPB DC 0	DAC 1 SAFEE	OUT D
493	VFP DC 1		OUT H
494	VFP DC 2	DAC 2 SAFEE	OUT D
495	VFP DC 3		OUT H
496	TRIGWBIAS	DAC 3 SAFEE	OUT A

497	RESWBIAS	DAC 3 SAFEE	OUT B
498	MGO UPPER THR DC 0	DAC 1 SAIE	OUT E
499	MGO UPPER THR DC 1		OUT F
500	MGO UPPER THR DC 2		OUT G
501	MGO UPPER THR DC 3		OUT H
502	MGO LOWER THR DC 0	DAC 1 SAIE	OUT A
503	MGO LOWER THR DC 1		OUT B
504	MGO LOWER THR DC 2		OUT C
505	MGO LOWER THR DC 3		OUT D
506	MGO UPPER THR DC 0 (SAFEE 2)	DAC 2 SAIE	OUT E
507	MGO UPPER THR DC 1 (SAFEE 2)		OUT F
508	MGO UPPER THR DC 2 (SAFEE 2)		OUT G
509	MGO UPPER THR DC 3 (SAFEE 2)		OUT H
510	MGO LOWER THR DC 0 (SAFEE 2)	DAC 2 SAIE	OUT A
511	MGO LOWER THR DC 1 (SAFEE 2)		OUT B
512	MGO LOWER THR DC 2 (SAFEE 2)		OUT C
513	MGO LOWER THR DC 3 (SAFEE 2)		OUT D
514	HVBIAS	DAC 3 SAIE	OUT A
515	CAL PULSE ENERGY 0		OUT E
516	CAL PULSE ENERGY 1		OUT F
517	CAL PULSE ENERGY 2		OUT G
518	CAL PULSE ENERGY 3		OUT H
519	ENERGY UPPER THRESHOLD		OUT B
520	NUMBER OF CAL PULSES	dummy	8 number of cal pulses
521	DC UNDER CALIBRATION	8 Stretcher bit	8 DC to calibrate
522	DC TO CONFIGURE	8 Stretcher bit	8 DC to config.
523	DC STATUS	dummy	8 DC supplied
524	DC ENABLE	dummy	8 DC listened
525	DC MASTER RESET	dummy	8 DC to reset
526	VETO ENABLE	dummy / 2 AUX	6 VETO enable
527	VETO CONFIG	dummy / 3 width	2 delay

1.2.4.2 Data Field Header for <CONF>

The lenght of configuration file exceeds the maximum packet lenght. So we divided the configuration file into two packets. According to “PacketLib Packet Format scheme 2” [3], the Source Data Field has a variable number of Data Blocks. In <CONF> we have only one block. This Data Block has a variable number (n) of elements of fixed dimension. The number of elements is specified in the Data Block Header. Each element is 2 Bytes long.

According to the “PacketLib packet format scheme 2” we chose:

MSB LSB

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Spare						Checksum flag		Packet Type				Packet SubType			
Time Tag second (32 bit integer signed MSB)															
Time Tag second (32 bit integer signed LSB)															
Time Tag millisecond (16 bit unsigned)															
Packet Format Version															
Number of configuration blocks in the packet															

CheckSum Flag: 00 binary TBD;

Packet TYPE/SUBTYPE: they specify the packet type/subtype.

We chose:

<CONF> : 1/3

for SAIE TE .

Time Tag: CPU System Time expressed as second and millisecond since 00:00 UTC january 1,1970 (i.e. as gven by the C function ftime) corresponding to the creation of the packet (see [3]).

Packet Format Version: this field is filled with the current version for the format of the packet identified with APID/Type/SubType.

Number of configuration blocks in the packet: Number of configuration blocks contained in this data packet (fixed to 1).

1.2.4.3 <CONF> Source Data Field

1.2.4.3.1 <CONF> Data Block Header

MSB LSB

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Number of configuration words (elements)															
Index of configuration packet															EOC

Number of configuration words (elements): Number of configuration words contained in this data packet.

Index of configuration packet: The SAIE TE configuration doesn't fit the packet limit, so we divide the configuration in more than one packet. This field indicates which configuration "block" is described by this <CONF>.

EOC: This bit is set to 1 for the last configuration block of the last configuration packet, and is set to 0 for all the other cases.

1.2.4.3.2 <CONF> Data Block elements

MSB LSB

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CONF Word 0, MSW															
CONF Word 0, LSW															
. . .															
CONF Word n-1, MSW															
CONF Word n, LSW															

1.2.5 Data Field for <LOG>

“Log” packets contain a “General Log”. Each packet contains a row of “General Log”. “General Log” contains:

- the Commands log fields coming from the “SAIE TE”

According to “PacketLib Packet Format scheme 2” [3], we chose:

1.2.5.1 Data Field Header for <LOG>

MSB LSB

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Spare						Checksum flag		Packet Type				Packet SubType			
Time Tag second (32 bit integer signed MSB)															
Time Tag second (32 bit integer signed LSB)															
Time Tag millisecond (16 bit unsigned)															
Packet Format Version															
Number of Blocks in the packet															

CheckSum Flag: 00 binary TBD;

Packet TYPE/SUBTYPE: they specify the packet type/subtype.

We chose:

<LOG> : 1/3

for SAIE TE.

Time Tag: CPU System Time expressed as second and millisecond since 00:00 UTC january 1,1970 (i.e. as gven by the C function ftime) corresponding to the creation of the packet (see [3]).

Packet Format Version: this field is filled with the current version for the format of the packet identified with APID/Type/SubType.

Number of Blocks in the packet: this is the number of data blocks contained in the packet (fixed to 1).

1.2.5.2 <LOG> Source Data Field

According to “PacketLib Packet Format scheme 2” [3], the Source Data Field contains a single block.

This Block has a variable number of elements of fixed dimension. One element is an unsigned short word.

1.2.5.2.1 <LOG> Data Block Header

MSB LSB

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Number of configuration words (elements)															
Index of configuration packet															

Number of characters of the row: this is the number of characters contained in the "Log Row" including the termination character.

Index of Log row: is the index of the row in the log file to be build (and the index of <LOG>).

1.2.5.2.2 <LOG> Data Block Elements

The "Log Row" could be even or odd bytes long. The number of elements is calculated on the basis of

the "**Number of characters of the row**". If this number is odd, the last element contains a spare byte.

If m is the "**Number of characters of the row"** and m is odd, we have:

MSB LSB

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Character 1								Character 2							
Character 3								Character 4							
. . .															
Character m								Spare							

If m is the "**Number of characters of the row"** and m is even, we have:

MSB LSB

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Character 1								Character 2							
Character 3								Character 4							
. . .															
Character m-1								Character m							

1.2.6 Data Field for <TUT>

This packet contains the "Run start" time. This information is given by the PC serving the acquisition. The accuracy of the time of the run start, estimated in this way, is around some tenth of millisecond. Moreover the PC time is not synchronized with other instruments or clocks (for example http://www.boulder.nist.gov/timefreq/cesium/parcs.htm).

1.2.6.1 Data Field Header for <TUT>

MSB LSB

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Spare						Checksum flag		Packet Type				Packet SubType			
Time Tag second (32 bit integer signed MSB)															
Time Tag second (32 bit integer signed LSB)															
Time Tag millisecond (16 bit unsigned)															
Packet Format Version															
Block Lenght (number of unsigned short words)															

CheckSum Flag: 00 binary TBD;

Packet TYPE/SUBTYPE: they specify the packet type/subtype.

We chose:

<TUT> : 1/4

for SAIE TE.

Time Tag: CPU System Time expressed as second and millisecond since 00:00 UTC january 1,1970 (i.e. as gven by the C function ftime) corresponding to the creation of the packet (see [3]).

Packet Format Version: this field is filled with the current version for the format of the packet identified with APID/Type/SubType.

Block Lenght: this is the number of unsigned short words contained in the block, this number is fixed to 4.

1.2.6.2 <TUT> Source Data Field

According to "PacketLib Packet Format scheme 3" [3], the Source Data Field contains a single block.

This Block is 3 unsigned short words long.

MSB LSB

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Time Tag second (32 bit integer signed MSB)															
Time Tag second (32 bit integer signed LSB)															
Time Tag millisecond (16 bit unsigned)															

Time Tag: CPU System Time expressed as second and millisecond since 00:00 UTC january 1,1970 (i.e. as gven by the C function ftime) corresponding to the "Run Start" in the DAQ-PC reference system (see [3]).

Bibliography

1: M. Trifoglio et al, astro-ph/0101122, 9 January 2001
2: M. Trifoglio et al, AGILE-ITE-SR-001, January/2002
3: A. Bulgarelli, Data Field Handled by PacketLib, 8/5/2
4: M. Trifoglio et al, AGILE-ITE-SR-003, April/2002