LYRA on PROBA-2

LYRA
Data Manager

M. Dominique, J-F. Hochedez, A BenMoussa.

Royal Observatory of Belgium
Solar Physics Department
Avenue Circulaire 3, B-1180 Bruxelles
Tel : ++32-(0)2-373 02 76 ; fax : ++32-(0)2-373 02 24
hochedez@oma.be
mariedo@oma.be

Acronyms

ADPMS	Advanced Data and Power Management System
CS	Checksum
CSL	Centre Spatial de Liège, Belgium
FIFO	First In First Out memory
FOV	Field-of-view
HK	Housekeeping
H/W	Hardware
IIU	Instrument Interface Unit
I/F	Interface
LED	Light Emitting Diode
LDM	LYRA Data Manager
LIM	LYRA Instrument Manager
LOB	LYRA Optical Box
LYRA	The Lyman-alpha Radiometer onboard PROBA, this proposal
MCPM	Mass memory Compression and Packetisation Module
MSB	Most Significant Bit
MUX	Multiplexer
PMOD	Physikalisch-Meteorologisches Observatorium Davos
PROBA	PRoject for On-Board Autonomy
QE	Quantum Efficiency
ROB	Royal Observatory of Belgium
S/C	SpaceCraft
SNR	Signal to Noise Ratio
S/W	Software
SWAP	Sun Watcher using APS detectors and image Processing
TBC	To Be Confirmed
TBD	To Be Defined
TC	Telecommand
TM	Telemetry
UTC	Coordinated Universal Time
UV	Ultraviolet
VFC	Voltage to Frequency Converter
VIS	Visible

1 Reference Documents

RD-1 RP-ROB-LYR-4001 Lyra Instrument Manager 2004-02-09
RD-2: RP-ROB-LYR-4002 Lyra Data Manager 2004-02-09

2 Applicable Documents

[1] Electrical Interface Control Document (EICD)
2004/09/21
S. Koller (PMOD)
[2] LYRA instrument manager (LIM)
Marie Dominique, Silvio Koller, J-F. Hochedez
LYRA-LIM-V1R13-200411126-ROB.doc
[3] Telemetry Packet (blue book), CCSDS [CCSDS 102.5-B-5], November 2000;

3 Document Change Record

Issue	Date	Modifications
V2R10	13 October 2004	CDR document
V2R11	26 November 2004	- Hilighted in blue and yellow
V2R12	29 November 2004	- This document change record - note2 §5.1.1 pg 9

4 Introduction

4.1 LYRA software

The onboard LYRA software (S/W) running on the main spacecraft computer (ADPMS) is split into two parts:

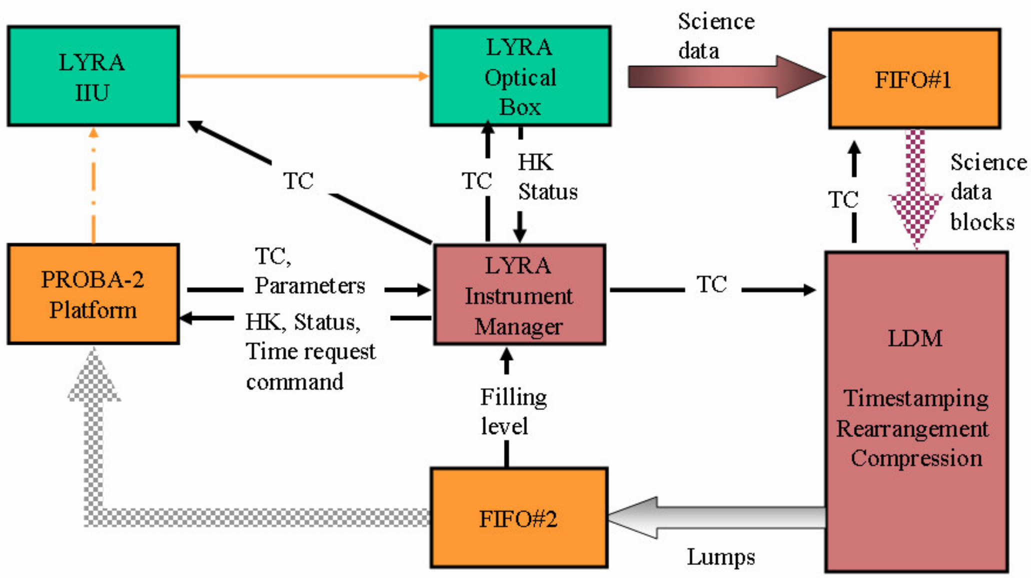
1. **The *instrument manager*** (LIM) manages the overall LYRA operations. It interacts with the PROBA-2 platform (sending and reading commands, parameters or ancillaries), with the data manager and with the LYRA hardware (through the two RS422 channels and a digital line). The Instrument Manager can perform activities at a maximum rate of 20 Hz[1] (in the sense that it can access the various H/W at a maximum rate of 20 Hz). Timestamping, which is a faster operation, is handled by the LDM which uses a dedicated system described in the next chapter.
2. **The *data manager*** (LDM) reads the data (Req. 1.1), checks their validity (Req. 1.2), adds the timestamps (Req. 1.3), rearranges and compresses the data (Req. 1.4) before they are stored in the ADPMS memory which will be responsible for their sending to the ground by telemetry (Req. 1.5). The data manager interacts with the LIM, the S/C, the LOB (interrupts reception and FIFO#1) and the OBET. Its parameters can be changed by TC from the ground (Req. 1.6.1).
   To summarise, the parameters of the LDM are:
   - En-/disable the compression (Req. 1.6.2)
   - En-/disable the rearrangement (Req. 1.6.3)
   - Keep one science block out of X (Req. 1.6.4)

   Notes:

   1. The LIM shall be able to interrupt the reading of FIFO#1 by the LDM, which shall imply to close the lumps (Req. 1.1.2). Before to resume the reading of the data in that FIFO (which is commanded by the LIM), it shall be cleaned in order to avoid a desynchronization with the timestamps (Req. 1.1.3)
   2. It shall be possible to reload a new version of the LDM from the ground (Req. 1.7).
   3. The data bank of the spacecraft shall be accessible to the LDM in order, for example, to access to the version number defined in chapter 6 (Req. 1.8).
   4. The LDM will also analyze the filling level of the FIFO#1 in order to detect the occurrence of a latch up (Req 2; cfr. Req. 4.2

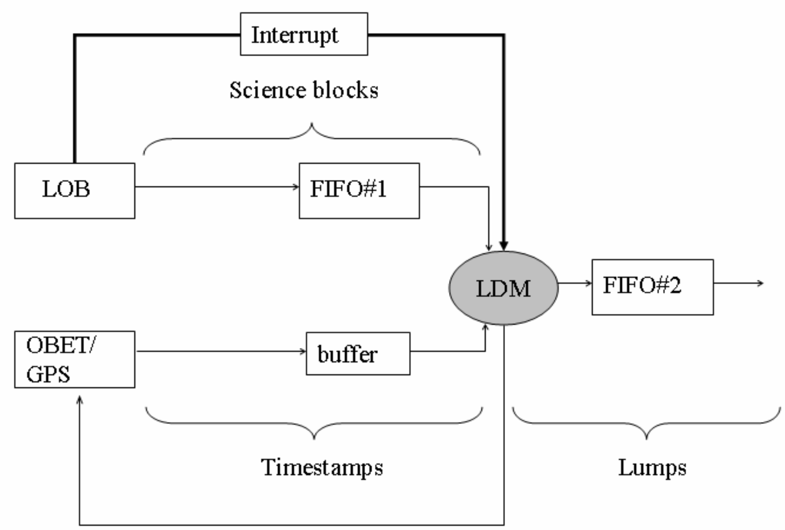
[1] The 20Hz value isn't a requirement but is information from SBI.

in [2]). In that case, no data are sent from the LOB and the FIFO#1 filling level will remain constant. If after 11 (adaptable value) readings of this FIFO no data have been registered and if the lumps have not been closed by the LIM (acquisition shouldn't have stopped), the LDM will conclude that a latch up occurred (Req. 2.1) and an event will be logged (Req. 2.2), followed by a closing of the current lumps by the LDM (Req. 2.3) and the turn on procedure of the ASIC by the LIM



5 Data Manager functions

The ~~general~~ main goal of the LDM is to create data lumps with the scientific data, each data lump corresponding to the contents of one telemetry packet.



6 Lump creation

Science blocks are continuously stored in the ADPMS memory at up to 100 Hz by a dedicated system (FIFO#1) (Req. 1.1.1). They are then handled by the LYRA Data Manager (LDM), which rearranges and compresses them before resending them to the ADPMS memory. The LDM offers the possibility to shortcut some operations. For example, it shall be possible to avoid the compression (cfr. Req. 1.6.2) or to conserve only one science block out of X (where X is a parameter to choose) (cfr. Req. 1.6.4). But it shall be also possible to transmit data from the LOB without any rearrangement (except grouping them in order to define the packet contents) (cfr. Req. 1.6.3). In this case, LDM should only join timestamps to science blocks (if applicable) and group a given number (default = 130 but should be a parameter) of science blocks to constitute the packet content (Req. 1.6.3.1).

6.1 Data series rearrangement

The rearrangement of a time series aims at minimizing the loss of information in case a packet is lost during telemetry sending. It shall also increase the compression performance. It consists in removing all unused bits from the data blocks (Req. 1.4.1.1.2.4) and in organizing these blocks in such a way that two consecutive blocks will not be put in the same telemetry packet (Req. 1.4.1). The number of unused bits is described in annexe.

6.1.1 Composition of a science block

A science block is composed of a start byte, 8 signals, each corresponding to one acquisition channel (4 belong to the nominal unit and 4 belong to an optional back up unit), a counter, a flag, an integration time, a multiplexer address and a checksum byte.

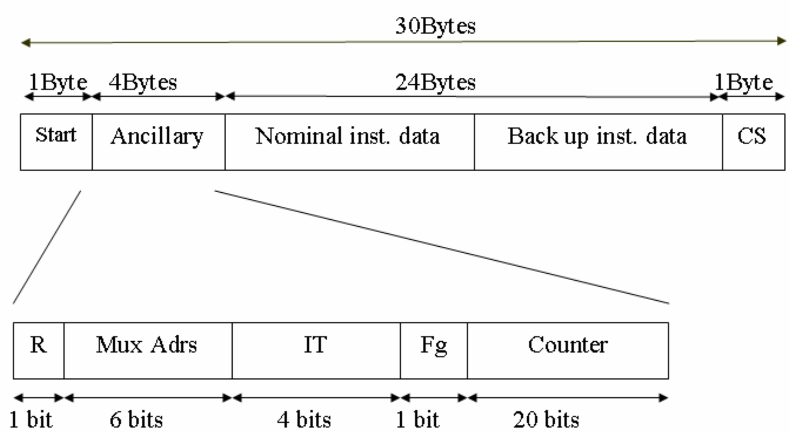


Figure 1: Composition of a science block

The counter is reset each orbit (more precisely at X1, see [2], the moment when ASIC is reloaded). At the same time the gate sends an interrupt to the LDM, which shall store a 5 Bytes timestamp (Req. 1.3.1). **Such interrupts shall frequently happen thereafter (baseline is every 10 seconds (even during night period)** (Req. 1.3.1.1)**, and when the counter is reset** (Req. 1.3.1.2)**, and when the integration time is modified** (Req. 1.3.1.3)) and a new timestamp shall be recorded at the same memory address than the old one (Req. 1.3.2). That means that only one timestamp will be accessible at a time. When the LDM reads the FIFO#1, it continuously analyses the science blocks and checks their validity by re-calculating the CS and comparing this with the CS stored in the science block (cf. Req. 1.2). Each science block corresponding to a timestamp is identified by the flag. Each time the LDM receives a science blocks with an ON flag, it copies the value of the timestamp and the counter of the associated data (see below) (Req. 1.3.3). All this assumes that the FIFO#1 is read fast enough to avoid the overwriting of a timestamp that wouldn't have been associated to the appropriate science block. This implies a FIFO#1 reading speed equivalent to the maximum timestamp emission speed (presently equal to once a second) (Req. 1.1.4). Thus, the FIFO#1 should have a size of 30 Bytes x 110 acquisitions[2] = 3300 Bytes (Req. 1.1.5).

Note 1: In case of latch up, the ASIC will autonomously turn off and no more data will be acquired. But, since no command has been sent to the LOB to stop the acquisition, no command will be sent to the LDM to close the lumps. The LDM will detect such a situation by observing the filling of FIFO#1. Indeed, if it remains constant during 11 (following the readings cadence) readouts[3] while LYRA is supposed to be acquiring, the LDM will deduce that a latch up occurred and an event will be logged. It will be followed by a turn on procedure of the ASIC and by a cloture of the current lumps (cf. Req. 2).

Note 2:

(1) The proposed stored time can be the GPS Time, but the onboard time is also accepted.

(2) The accuracy requirement on the time between gate signal reception and storing of the time must be better than 5ms.

6.1.2 Rearrangement

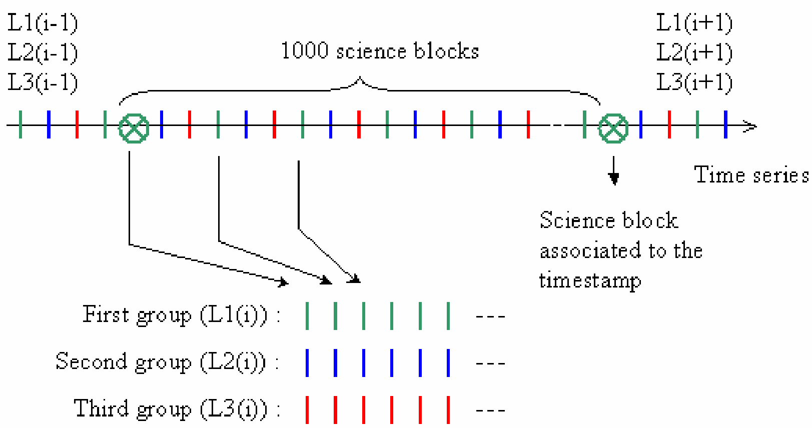
The LDM shall sort 1000 consecutive science blocks having the same integration time to form three data lumps (Req. 1.4.1.1). **Note that, in some situations, the lumps shall be prematurely closed** (they will contain a smaller number of science blocks) (Req. 1.4.1.2). These situations are **a change of the integration time** (Req. 1.4.1.2.1)**, a reload of LDM or a change in its parameters** (Req. 1.4.1.2.2)**, a reload of the LIM modes** (Req. 1.4.1.2.3)**, interruptions of acquisitions** (Req. 1.4.1.2.4) **and the occurrence of a latch up** (Req. 1.4.1.2.5)**.** Also **at the beginning of each orbit** (simultaneously to the counter reset, i.e. when triggering the stand by mode : X1) the currently open lumps are closed to avoid that two science blocks with the same counters are comprised into the same lump (Req. 1.4.1.2.6). All the blocks composing a same lump have thus the same integration time. Of course, if no new data has been acquired since the last closing of the lumps, a new "close the lumps" command will be neglected (no void lump will be created) (Req. 1.4.1.3). Data lump formation is achieved in two steps.

First step: Science blocks will be distributed between three groups as illustrated below (Req. 1.4.1.1.1).

---

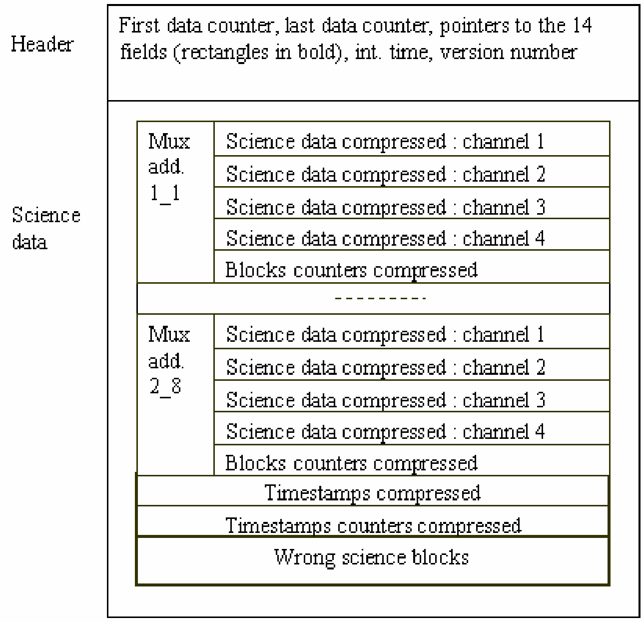
[2] Comprising a margin.

[3] If the 10 sec integration time is selected, the FIFO#1 is still empty during about 9 readings by the LDM (depending on the FIFO#1 reading cadence), independently of the occurrence of a latch up. We have thus to wait more than 10 readings to detect it.



Second step: Each group will be structured in a data lump (Req. 1.4.1.1.2). Common parameters of all science blocks in a same lump shall be written in a header (Req. 1.4.1.1.2.1). The checksum of each science block shall be checked (cf. Req. 1.2). If an error occurs, the science block will be put in a dedicated field of the data lump (see figure below: the last rectangle in bold) without any transformation (Req. 1.4.1.1.2.2). If not, the data contained in science blocks and their respective counters shall be extracted and sorted following their multiplexer addresses[4] (Req. 1.4.1.1.2.3). Data shall be truncated to suppress unused bits (cf. Req. 1.4.1.1.2.4). For each MUX address, counters and data shall be compressed together (Req. 1.4.1.1.2.5). The timestamps and the counters of the corresponding science blocks shall be stored in another field and compressed too (Req. 1.4.1.1.2.6). Pointers to the beginnings of the 14 fields[5] of the lump (12 MUX addresses + timestamps field + wrong sc. blocks field) shall be put in the header (Req. 1.4.1.1.2.1.1). The data lump format is summarized in the following figure(Req. 1.4.1.1.2.1.3).

[4] The multiplexer address allows us to know which VFC group has been used and which input of the multiplexer has been selected (i.e. which channel or which VFC calibration voltage).
[5] Inputs 4 and 5 of each multiplexer being not accessible, only 6 fields per multiplexers group are provided.



Notes:

1. The version number introduced in the header (Req. 1.4.1.1.2.1.2) determines which versions of the LDM and of the LIM have been used when acquiring the data. This number is incremented when a new software version is uploaded (Req. 1.4.1.1.2.1.2.1). The format of this version number should have the form XXXXXXXXYYYYYYYY, where XX..X is the version number of the LIM and YY..Y is the version number of the LDM (Req. 1.4.1.1.2.1.2.2).
2. If an error occurs between the LOB and the LDM, the checksum of the science block will be wrong and this block will receive a special treatment (see above). A particular attention must be given to the case of an error in the science block corresponding to the timestamp. In this case, the timestamp will not be registered because the science block will be recognized as false and stored, without other treatment, in the data lump (see before). Moreover when the next science block associated to a timestamp will be treated, the timestamp corresponding to the false block will have been erased by the next one and no desynchronisation will appear. The time interpolation will then be performed on a 20 s interval instead of a 10 s interval in the case of a 100 Hz acquisition cadence.
3. Most of the time, the back-up channel will not be activated, and the back-up data in the science blocks will be theoretically zero, because the second VFC

group will be switched off. In such a case, these nearly zero data should of course not be written in the data lump (Req. 1.4.1.1.2.3.1). In practise, however, a zero value is not guaranteed because there is a risk for a spike to trigger the counter. Nevertheless, as this situation is not susceptible to occur frequently, this will not be taken into account on the on-board software, and such data will thus be conserved as the relevant ones (Req. 1.4.1.1.2.3.2).

6.2 Compression

To save memory during data storage and sending, data will be compressed. The Welch version of the Lempel-Ziv (LZW) compression algorithm is baseline (Req. 1.4.2). It is a dictionary compression method in which the most frequent strings are coded with the fewest bits. Copyright and the code can be found at the following links.
http://ftp.debian.org/debian/pool/non-free/n/ncompress/ncompress_4.2.4.orig.tar.gz
http://ftp.debian.org/debian/pool/non-free/n/ncompress/ncompress_4.2.4-13.diff.gz

This algorithm offers good results for smooth data (most of the time, during quiet sun periods, it will be the case with LYRA data) when the difference between each value and the previous one is coded instead of the value itself.

Preliminary tests have shown that the compression is fast and the ratio is good.

As a first approximation the amount of data transmissible by PROBA2 telemetry for a 600km orbit (worst case) is summarized in the table below.

Contact possibilities	Amount of data
Only Redu	1 Gbit (0,125 Gbytes) /day
Only Kiruna	2,2 Gbits (0,275 Gbytes) /day
Both	probably 2,7 Gbits (0,3375 Gbytes) / day

One day of LYRA observations shall produce ca. 45 Mbytes[6] (data) + 15 Mbytes (counters) + approximatively 1 Mbytes (headers + timestamps and associated counters + wrong science blocks).
Depending on the percentage of the telemetry dedicated to LYRA, it shall be checked if a compression ratio between 1/3 and 1/5, which has been reached during tests, will be good enough.

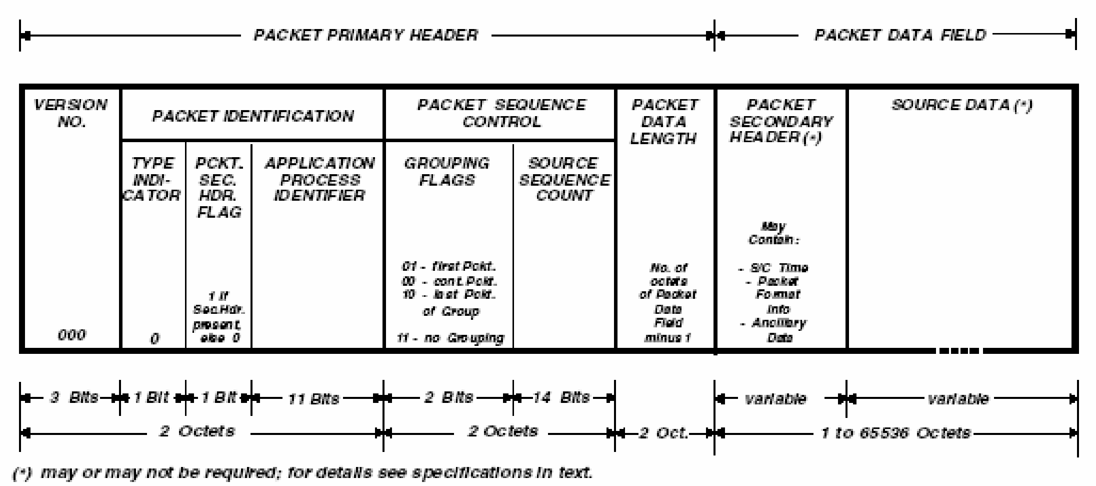
[6] Considering that acquisitions will be performed during 70% of the orbit.

Remark: The compression algorithm shall be implemented in the form of a library function that might be of use for the SWAP data manager as well. The size of the data to be compressed should be a parameter of this function. (Req. 1.4.2.1)

6.3 Packetization (Req. 1.4.3)

The rearrangement process is useful if each packet comprises exactly one data lump (Req. 1.4.3.1). In practise we assume that every lump contains a header, 334 blocks (maximum) and maximum 334 timestamps and their counters. Each block is usually composed of maximum 4 x 14 bits[7] (sometimes 8 x 14 bits) for data (once truncated) and 20 bits for the counters. Each timestamp is coded with 5 Bytes. As said before, data, counters and timestamps shall be compressed (by a factor from 3 to 5). To form a packet, an other header shall be added as specified in the reference document CCSDS 102.0-B [3] (Req. 1.4.3.2).

Note: To be sure that the rearrangement process is useful, the spacecraft must insure that two consecutive packets will not be sent together (Req. 1.4.3.3).



[7] 14 bits is the number (cfr. Annexe 8.2) of bits used to code the data, when integration time is 10 ms. This number will be higher for a longer IT.

7 Annexes: Data format

7.1 Science data block from the LOB

Just for information[8], the onboard detectors shall acquire at a maximum rate of 100Hz ~~(default cadence)~~. Usually, only one unit shall be operated except during the backup mode periods when two units shall be used. However, eight signals shall always be transmitted from LOB to LDM in order to keep the same data pattern as in backup mode. Each science data channel has a maximum resolution of 24 bits per integration cycle. Four Bytes are used to store a counter, a flag, the integration time and the multiplexer addresses. The counter is reset every orbit. Science blocks associated with timestamps are identified by the flag. A checksum method is implemented which requires one more byte to store the CS.

Byte	# of Bytes	Description	Remarks
Byte 29	1	Start byte	ASCII code 231
Byte 28 – 25	4	Header	20 bit Block counter value
			1 bit Synch flag
			4 bit Integration time setting
			6 bit Mux address
			1 bit Reserve
Byte 24 – 22	3	Channel 4	Nominal Instrument
Byte 21 – 19	3	Channel 3	Nominal Instrument
Byte 18 – 16	3	Channel 2	Nominal Instrument
Byte 15 – 13	3	Channel 1	Nominal Instrument
Byte 12 – 10	3	Channel 4	Backup Instrument or 0
Byte 9 – 7	3	Channel 3	Backup Instrument or 0
Byte 6 – 4	3	Channel 2	Backup Instrument or 0
Byte 3 – 1	3	Channel 1	Backup Instrument or 0
Byte 0	1	Checksum	

[8] This section should not be considered as a requirement, it is only summary of the EICD [1]

8 Annexes: Number of significant bits in the data format

The number of significant bits in the data format depends on two issues: the number of bits needed to code the received signal and the importance of the noise relatively to the signal.

8.1 Number of bits needed to code the received signal

Each VFC has a maximum frequency of 1.2288MHz, which corresponds to extreme values of the voltage that can be found at the input of the VFC. A pulse counter, placed after the VFC, counts the pulses emitted during the integration time. That means that for an integration time of 10 sec, a maximum of 12288000 pulses will be registered, while for an integration time of 10 msec the maximum number of pulses will be 12288. The following table shows the maximum number of counted pulses and the number of bits needed to code these numbers of pulses for each possible value of integration time. When reorganizing the data, the LDM will truncate the unused bits.

Integration time	Max number of pulses	Number of bits needed
10 s	12288000	24
5 s	6144000	23
2 s	2457600	22
1 s	1228800	21
500 ms	614400	20
200 ms	245760	18
100 ms	122880	17
50 ms	61440	16
20 ms	24576	15
10 ms	12288	14

8.2 Importance of the noise relatively to the signal

$$SNR = \frac{S}{\sigma_S} = \sqrt{\frac{T_{Integration}(s) \cdot i(Amps)}{e^- \cdot QE}} = 2^{Needed\_resolution}$$

Where :

- i is the current obtained at the output of the detector at saturation
- $T_{integration}$ is the integration time
- e^- is the electron charge
- QE is the quantum efficiency (in electrons /photon) and depends on the wavelength

Thus, the needed resolution will depend on the channel (which determines the saturation current) and on the integration time. This is illustrated on the table below in which we have considered a value of i related to a maximum[9] solar activity and a value of QE which corresponds to the maximum SNR on the wavelength interval considered.

	SNR max				nb of bits needed			
	MSM			PIN	MSM			PIN
Integration Time	1-20 nm	17-70 nm	115-125 nm	200-220 nm	1-20 nm	17-70 nm	115-125 nm	200-220 nm
10 ms	4426	29067	6982	281830	13	15	13	19
20 ms	6259	41107	9874	398568	13	16	14	19
50 ms	9896	64995	15613	630191	14	16	14	20
100 ms	13995	91917	22080	891224	14	17	15	20
200 ms	19792	129991	31225	1260381	15	17	15	21
500 ms	31293	205533	49372	1992838	15	18	16	21
1 s	44256	290668	69822	2818299	16	19	17	22
2 s	62587	411066	98743	3985676	16	19	17	22
5 s	98959	649953	156127	6301907	17	20	18	23
10 s	139948	919173	220796	8912243	18	20	18	24

[9] Relative to an ordinary solar cycle.

In order to simplify the data manager task and taking into account the fact that a suppression of the non-significant bits should not have an important impact on the memory and telemetry needs, only the unused bits in the beginning of each data block will be truncate by the data manager.