

 <p>Planetary Fourier Spectrometer PFS</p>		<p>PFS for Mars Express</p>	<p>PFS-FUM 3 Page 1</p>	<p>P.I. Vittorio Formisano CNR IFSI</p>
--	---	------------------------------------	------------------------------------	---

MARS EXPRESS

PLANETARY FOURIER SPECTROMETER

SOFTWARE DESCRIPTION

MEX-CNR-FUM 3

 Planetary Fourier Spectrometer PFS	 Mars Express	PFS for Mars Express	PFS-FUM 3 Page 2	P.I. Vittorio Formisano CNR IFSI
---	--	-----------------------------	----------------------------	--

FUM 3 - INSTRUMENT SOFTWARE AND MEMORY MAP

- 1.- Instrument Software Description
 - 1.1-Module O Software
 - 1.2-The FFT Software
 - 1.3-The Overall Software of the Experiment: Module E
- 2.-Dam Software
- 3.-Dam Software Architecture
- 4.-OBDM Software
- 5.-OBDM Commands
- 6.-ICM Software
- 7.-Memory Map

		<p>PFS for Mars Express</p>	<p>PFS-FUM 3 Page 3</p>	<p>P.I. Vittorio Formisano CNR IFSI</p>
--	--	-----------------------------	-----------------------------	---

1.- INSTRUMENT SOFTWARE DESCRIPTION

The experiment PFS makes use of three microprocessors , therefore there are 3 software packages running , eventually simultaneously. We use 2 μ p 8086 and one AD2100.

See also document MEX-CNR-IPDR- 10.01

1.1- MODULE O SOFTWARE.

The board OBDM controls the entire module O and all its delicate functions. The actual tasks performed by OBDM software are :

- Selftest.
- Receive setting parameters.
- Receive and execute commands from module E.
- Set hardware parameters.
- Read housekeeping.
- Perform autonomous test.
- Unblock/block double pendulum.
- Read interferograms.
- Reduce LW interferogram if needed.
- Transmit data to module E.
- Execute failure mode commands.

1.1.1– Selftest.

At switch on the first thing it is done by OBDM is a self test, in order not to have registers with wrong data.

1.1.2 - Receive setting parameters.

At switch on of the experiment a data block is passed from module E to O , containing the information for setting a number of parameters of critical importance : among these are the temperatures of the laser diodes, the status in which should be many different registers when no activity is going on , etc.

3.1.1 - Receive and execute commands from module E.

There may be different commands to execute :

- start data acquisition.
- start autonomous test.
- unblock/block double pendulum.
- take housekeeping data.
- give data to module E.
- give status to module E.
- go to initial position.
- monitor of reference channel.

		<p>PFS for Mars Express</p>	<p>PFS-FUM 3 Page 4</p>	<p>P.I. Vittorio Formisano CNR IFSI</p>
---	---	-----------------------------	-----------------------------	---

3.1.2 - Set hardware parameters.

There are a number of parameters to be set (or changed if needed) in the hardware, in order to start and work correctly : these are temperature of the laser diodes , temperature of the optical bench , but also in many other places in the different digital controller of the hardware.

3.1.3 - Read housekeeping.

The housekeeping is made of tbd parameters mostly temperatures and voltages, but there are also other important quantities like the status of module O, the status of the gas tight valve, the status of the block/unblock pin, the direction of motion of the pendulum, etc.

3.1.4 - Perform autonomous test.

The autonomous test is ment to be a complete check-out of module O , electrical, optical and mechanical conditions : it is performed by means of a number of motion of the double pendulum : first we measure the sine wave of the laser diode interferogram; then we measure the time for each zero crossing of the sine interferogram, so that we have a full study of the speed control loop hardware; we repeat then a few measurements for different gain settings; etc...

3.1.5 - Unblock/block double pendulum.

It is a simple routine that gives power to the Starsys device and waits the time needed for the device to block or unblock the double pendulum. It reports back the new status.

3.1.6 - Read interferograms.

Upon command the double pendulum moves and at each zero crossing of the sine interferogram the A/D converter provides the readings of the SW and of the LW channel radiation intensity of the interferogram. In this way two vectors of 16384 points are acquired.

3.1.7 - Reduce LW interferogram if needed.

From what we know there is no need to transmit 16384 points for the LW channel, so we can reduce them to 4096 points (one out of four) in order to reduce the amount of data to be transferred from O to E modules.

3.1.8 - Transmit data to module E.

Housekeeping data, the status data and the two interferograms are transferred from O to E according to a fixed protocol and under the control of module E. This data transfer must be minimised in order to reduce the repetition cycle. Presently it is estimated 2.5 sec for transferring LW 4096 + SW 16384 words; note that this time adds

		<p>PFS for Mars Express</p>	<p>PFS-FUM 3 Page 5</p>	<p>P.I. Vittorio Formisano CNR IFSI</p>
---	---	-----------------------------	-----------------------------	---

to the acquisition time of 5 sec, that is why we consider the repetition cycle of 10 sec.

3.1.9 - Execute failure mode commands.

There are several failure modes which are implemented to overcome possible problems : broken laser diode ; broken A/D converter ; no zopd signal ;etc...

Broken/misalignment of laser : we switch to the other one (normally the motion is controlled by the SW laser), we can use also the LW laser for motion control. Furthermore the interferogram signal is sampled by using each laser for each channel, but PFS can also make use of one single laser for sampling both channels.

Broken A/D converter : we switch to the other one (normally each channel use its own ADC, but in failure mode each ADC can be used for both).

No Zopd pulse : in this case the entire motion control would not work, and the experiment would fail : we can have a software routine that gives an order for the motion to start, and eventually correct the zopd position after one or more motions. A flag should indicate this way of working.

1.2- THE FFT SOFTWARE.

In order to reduce the amount of information to transmit to ground, the ICM hardware is able to perform the Fast Fourier Transform of the interferogram acquired by module O. There are 4 different ways to work for ICM, depending on the Coded Control Words sent from DAM before the data. These words specify ICM mode of operation and the memory bank to be used when the LW interferogram is being processed.

- 4K – 4 kilowords are given to ICM, no memory bank specified.
- 4Ks – 4 kilowords are given , but a memory bank is specified.
- 16Kl – 16 kilowords are given , but being a LW interferogram, only 2K are returned.
- 16K – 16 kilowords are given , but being a SW interferogram, 8K are returned.

In the Control Words can also be specified if the interferogram must be apodized (only a very weak apodization is implemented).

The above 4 modes of operation have another 4 counterpart as there is the possibility to return back the FFT module (spectral intensity) or the Im + Re parts (sine/ cos transform) . In the last case the data block returned has the same size as the data block given.

1.3- THE OVERALL SOFTWARE OF THE EXPERIMENT : MODULE E.

The software of module E has many important tasks as DAM is the center of the entire experiment in terms of decision taking, control, communication. The tasks of DAM are :

- Initialization.
- Switch on of subsystems.
- Control OBDM.
- Control ICM.
- Control Scanner.

		<p>PFS for Mars Express</p>	<p>PFS-FUM 3 Page 6</p>	<p>P.I. Vittorio Formisano CNR IFSI</p>
---	---	-----------------------------	-----------------------------	---

- Go to sleeping mode.
- Wake-up from sleeping mode
- Check time and calculation of orbital parameters.
- Give commands to module O.
- Give commands to module P.
- Give commands to module S.
- Rotate scanner.
- Get black body calibration data.
- Get Space calibration data.
- Get Autonomous test data.
- Get mars data.
- Control Mass Memory.
- Give data to ICM.
- Get data from ICM.
- Cut data according to data transmission mode.
- Get data from Mass memory.
- Prepare data in telemetry formats.
- Give data to spacecraft.
- Get commands from spacecraft.

Execute commands.

2. DAM Software

There's a principal unit called *control-unit* that manage through the other peripheral unit all the operation needed to execute the command sending using a telecommand from spacecraft. The control unit is based on a *session*, this one execute all the step following the mode and parameter that could be changed using the telecommands.

The session work in 9 mode: in mode 0 the system is in Stop mode ,in mode 2 the system is in Standby Mode, in mode 3 the system perform Test 1, in mode 5 the software perform Test 3, in mode 6 the software perform Test 4, in mode 7 the software perform Test 5, in mode 8 the system make a measurement, in 9 mode the system perform a single autotest.

2.1 Operations during the measurement session

Each measurement session provides with a sequence of Data Packs which are stored in the mass memory to be sent to the Earth.

One Data Pack represents the housekeeping and scientific information obtained during the one acquisition process (the movement of the double pendulum at some position of the pointing device – Scanner). The scientific information contains one interferogram or spectrum, probably cutted by some method, which takes into account a character of data (selection of spectra, averaging, etc.). The housekeeping information contains angles of the Scanner position, state of the module O during the acquisition, cutting method and other parameters to be used for the correct interpretation of the scientific information.

 Planetary Fourier Spectrometer PFS		PFS for Mars Express	PFS-FUM 3 Page 7	P.I. Vittorio Formisano CNR IFSI
---	---	----------------------	---------------------	-------------------------------------

At the beginning of the observation session DAM switches on the O module and receives their selftest information. If the selftest information of some module is not satisfactory (or the module does not respond), DAM switches the module off and on again until the maximal number of attempts is reached. Each attempt with the corresponding diagnostics is registered in the DAM Status Area. The DAM Status Area which contains also the own DAM status information is stored in the mass memory as the Data Pack number zero.

If O module are capable of working, DAM starts the regular acquisition processes (rotation of the Scanner, reception of interferograms, Fourier transformation, etc.), beginning with the Data Pack number three.

After that DAM sends the OBDM command "Halt" and, on receipt of the OBDM message "Halted", registers this message in the DAM Status Area and stores the status information in the mass memory as the Data Pack number Mh3.

Upon completion of the measurement session, DAM switches all modules off.

2.2 Operations during the autotest session

If both the O module is capable of working, DAM sends the OBDM command "Start Autotest". OBDM, in response to this command, moves the double pendulum forward, providing with a speed control information. This information is stored in the mass memory as the Data Pack number one. After that DAM sends this command again, OBDM moves the double pendulum backward and the speed control information is stored as the Data Pack number two.

3. DAM Software architecture

The scheme of the software is the following:

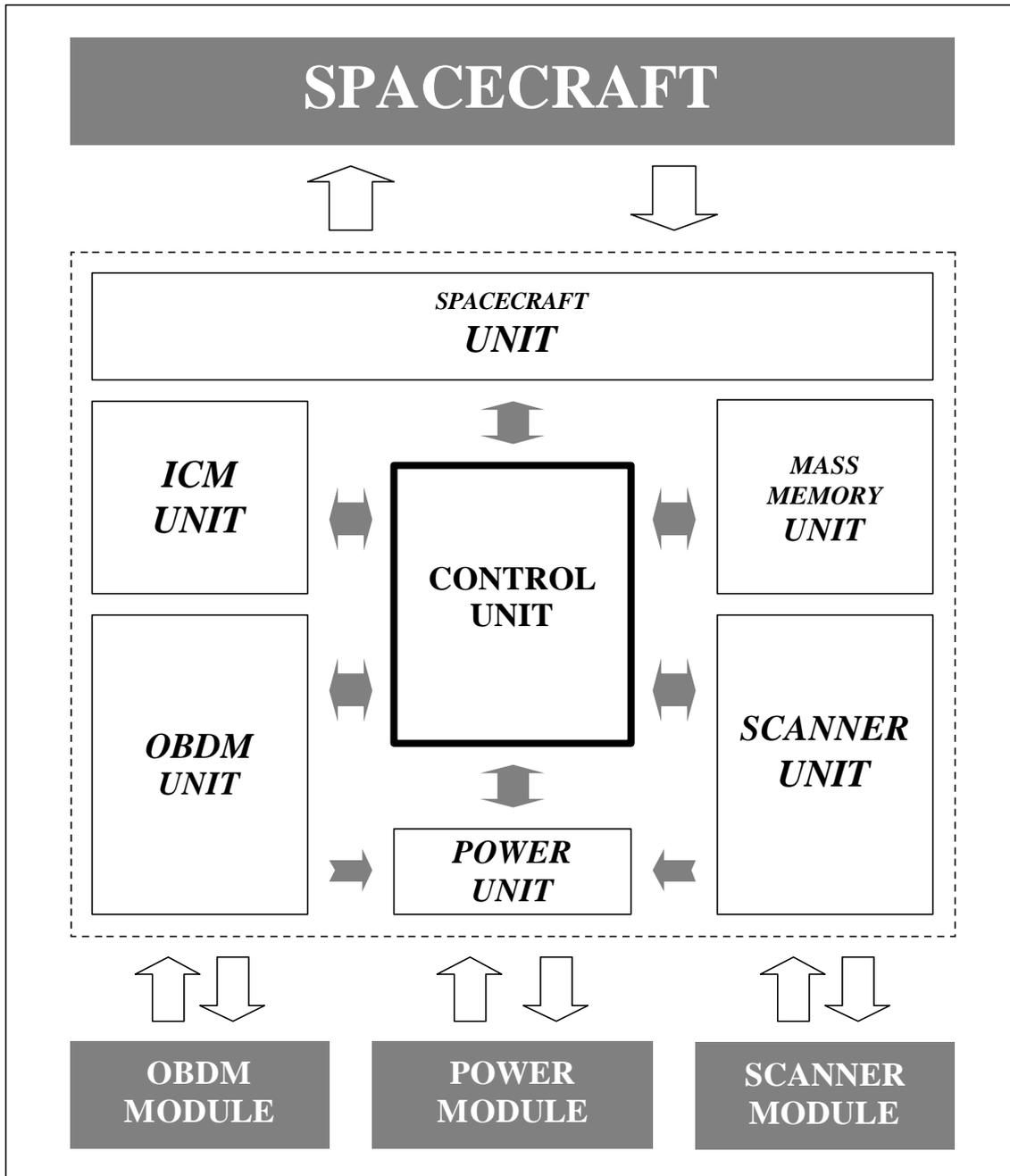


Fig. 1.1

The peripheral unit need to connect the principal unit (Control Unit) to the external peripherals, the spacecraft and the module O.

There is a 6 peripheral unit :

1. Spacecraft Unit
2. OBDM Unit

 <p>Planetary Fourier Spectrometer PFS</p>		<p>PFS for Mars Express</p>	<p>PFS-FUM 3 Page 9</p>	<p>P.I. Vittorio Formisano CNR IFSI</p>
---	---	-----------------------------	-----------------------------	---

3. Scanner Unit
4. Power Unit
5. ICM Unit
6. Mass Memory Unit

Spacecraft Unit is organized to receive the telecommand in the spacecraft format check the header, decode it and set the parameter used for the session in the control unit to do the operation.

For more information see the Database document.

The following is the type of data flow that the central unit exchanges through the peripheral unit to the external peripherals

Scanner command

An information which DAM sends to the Scanner in order to force it to perform requested action

Scanner message

An information which the Scanner sends to DAM upon completion of the Scanner command

OBDM command

An information which DAM sends to OBDM in order to force it to perform requested action

OBDM message

An information which OBDM sends to DAM upon completion of the OBDM command

Telecommand

A command sent to PFS from the Earth in order to force DAM to perform some single action. A Telecommand is represented by a sequence of CCW (one or more).

Data Pack

One Data Pack represents the housekeeping and scientific information obtained during the one Acquisition Process (movement of the double pendulum at some position of the Pointing Device). Scientific information contains one interferogram or spectrum. Housekeeping information contains angles of the Scanner position, state of the module O during the acquisition and other parameters to be used for the correct interpretation of the scientific information.

PFS Status Block

It contains essential information about the PFS state.

Event Logbook

A register of events detected by the DAM software. Each event is represented by the Event Code and, in some cases, Event Message. Each event has a timestamp.

		<p>PFS for Mars Express</p>	<p>PFS-FUM 3 Page 10</p>	<p>P.I. Vittorio Formisano CNR IFSI</p>
---	---	-----------------------------	------------------------------	---

Data Block

Both the Data Pack and the Event Logbook are divided into blocks. All Data Blocks have the same signs as the Data Pack (or Event Logbook) they belong to, and, in addition, the sign: - a block number which identifies it. In combination with other signs it allows to select all Data Blocks to be assembled into one Data Pack or Status Information.

3.1 OBDM Unit

OBDM Unit provide manage the communication with OBDM do the following operation:

1. Start and Stop OBDM
2. Reset OBDM
3. Check OBDM
4. Send command to OBDM
5. Receive and check data from OBDM

- Start and Stop OBDM

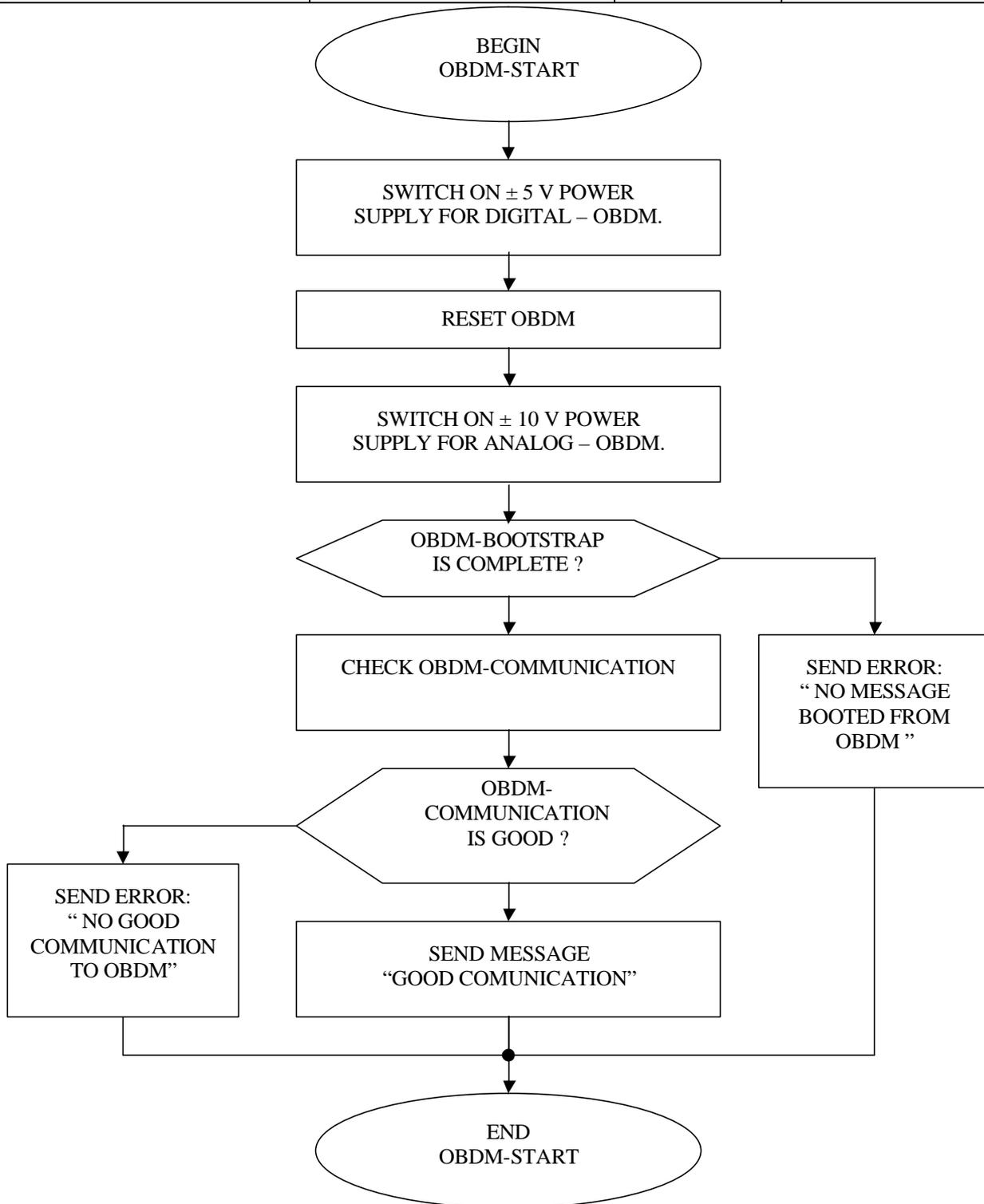
This operation is made in according to Power unit, OBDM unit provide to set up the power unit for switch on and off the power to module O digital (± 5 V) and analogue electronics (± 12 V).

- Start and Stop OBDM

The reset of OBDM is made forcing the reset pin to *low level* for 200 ms. It permit to synchronize the module E and module O and clean eventually spurious data into the I/O port.

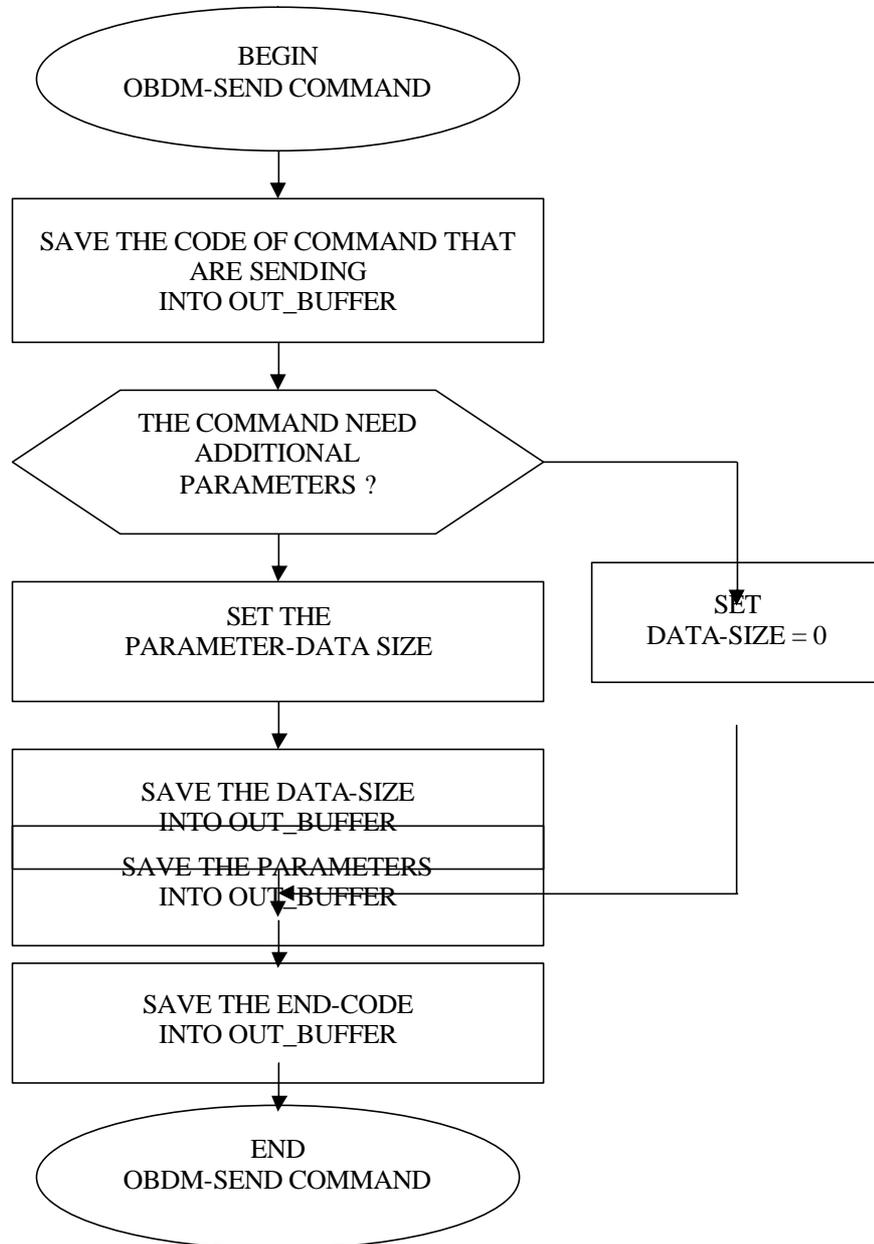
- Check OBDM

The unit check the communication to OBDM send a SW request without load into OBDM memory synthetic or real interferogram and the OBDM send a know-sequence of number, DAM receive this data and check it. If the sequence are correct DAM send a code message "GOOD COMMUNICATION WITH OBDM".



- Send command to OBDM

This unit provide to send the command needed to execute the telecommand. It send the command using a internal protocol describe in the OBDM-Software documentation.



- Receive and check data from OBDM

This unit provide to check if there is an error during the decode or execution the command. It check if the data transmitted is correct using a check sum. The checksum is a sum without carry. OBDM sends the message "Error in command" if one of the following conditions takes place:

 <p>Planetary Fourier Spectrometer PFS</p>		<p>PFS for Mars Express</p>	<p>PFS-FUM 3 Page 13</p>	<p>P.I. Vittorio Formisano CNR IFSI</p>
---	---	-----------------------------	------------------------------	---

- the command code is unknown
- the data size is invalid (if, for example, the data size field of the command "Send SW block" is not equal to 2)
- the check sum (if data are present) is wrong
- the datum value is invalid

OBDM, having the message "Error in command" sent, makes a transition to the state "Waiting for the command".

DAM, upon receipt of this message, resends the command again. DAM repeats resending of the command several times. After several unsuccessful attempts DAM resets OBDM or switches the OBDM power supply off.

3.2 SCANNER Unit

The Scanner (the pointing device) is used for pointing the field of view of PFS to a given observation point of the Martian surface. DAM controls over the Scanner operations, sending commands through a serial interface and analyzing received messages.

3.3 POWER Unit

The power control module P, this unit provide to send some sequences of byte to the power module to switch on or off all subsystem, and read the power status.

3.4 ICM Unit

This unit provide to transfer the interferogram from DAM memory to ICM memory, wait the completion of FFT and retransfer the spectra from ICM memory to DAM memory.

For detail see the ICM Software chapter.

3.5 Mass Memory Unit

The mass memory is a large size memory that contain the data after sending to spacecraft. This memory is organized in 3 bank. The CPU access to this memory using I/O port and memory address decodified using hamming code.

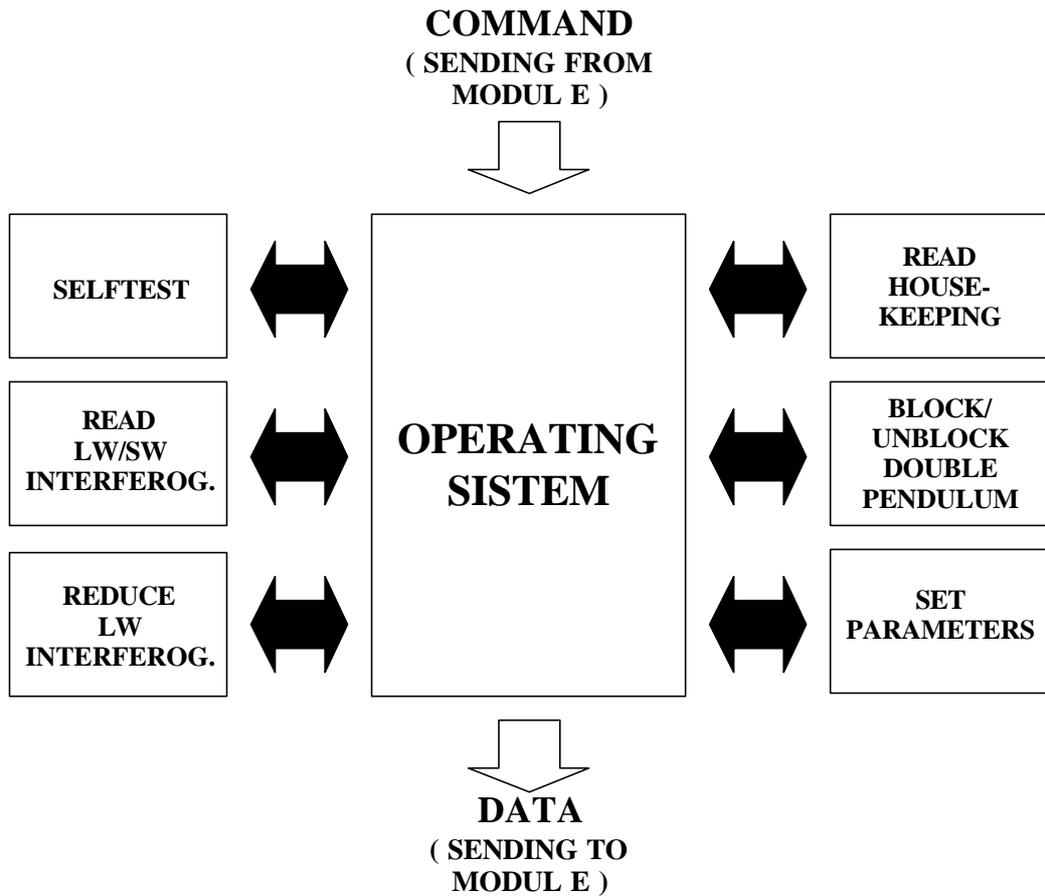
4. OBDM Software

The OBDM Software is a sort of "DAM-slave". It must executes a command sending from DAM and response data or code error if occur a error only during the decodifying of the command.

In the OBDM Software there's a principal program-unit called "operating system" that provide to read, decodify command code, sending from DAM, and dialog with the subroutine program needing to execute the command acquired from DAM.

There is a subroutine that execute 6 principal process:

1. Selftest
2. Read short and long wave interferogram
3. Reduce long wave interferogram
4. Read housekeeping
5. Block and Unblock double pendulum
6. Set parameters



 Planetary Fourier Spectrometer PFS		PFS for Mars Express	PFS-FUM 3 Page 15	P.I. Vittorio Formisano CNR IFSI
---	---	-----------------------------	-----------------------------	---

4.1 Operating system.

The operating system perform 3 operation: send and receive commands and data, decodify the commands sending from modul E and execute failure mode commands.

The operating system provide the execution of cycle-activity. The operation sequences is the following:

OBDM receive the whole command (sequence of bytes) before it will begin to execute it.

Upon completion of the command execution OBDM must send a corresponding message which informs DAM that the command was either executed or ignored by OBDM. OBDM, having the message sent, must be ready to receive and execute a new command.

DAM must receive the whole message before it will begin to analyze it.

DAM must not send a command if a previous one was not completed (the corresponding message is not received) unless the timeout condition comes.

4.2 Send and receive commands and data

This function implemented by operating system provide to send and receive the commands and data using a particular protocol.

Both OBDM commands which DAM sends to OBDM and OBDM messages which DAM receives from OBDM are represented by sequences of bytes, which have the same structure but different format.

The protocol consist in 2 operating mode depending of the flow-way: when the data or command flow is from DAM to OBDM the byte is encrypted, when the data or command flow is from OBDM to DAM the byte isn't encrypted.

The structure of the OBDM commands or messages sending from DAM is the following:

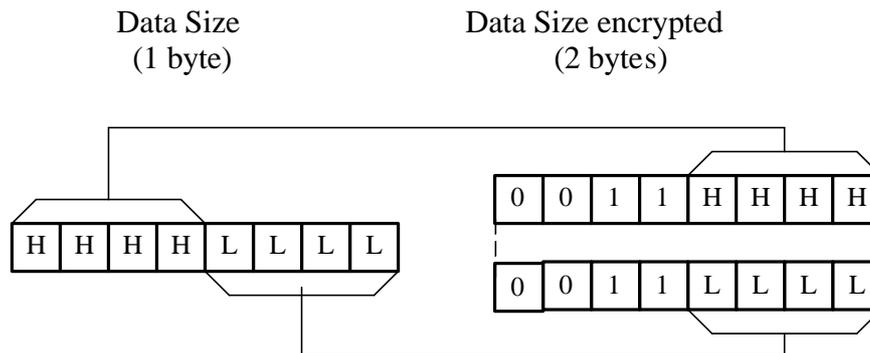
<i>CODE</i>	<i>DATA SIZE</i>	<i>DATA</i>	<i>CHECKSUM</i>
1 byte	2 byte	[DATA SIZE] byte	1 byte

CODE is the command numeric-code. This one identify the action that OBDM must execute.

This field is not encrypted. The Code must be within the range 10H..2FH.

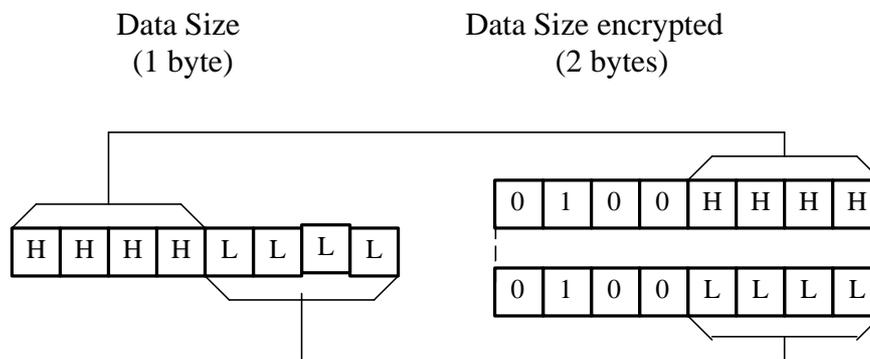
DATA SIZE is number of data bytes sending with the command-code.

This field is represented by two bytes: the first byte contains the code that identify the type of byte (0011) and the high nibble (four most significant bits) of the Data Size value, the second byte contains the code that identify the type of byte (0011) and the low nibble of the Data Size value:



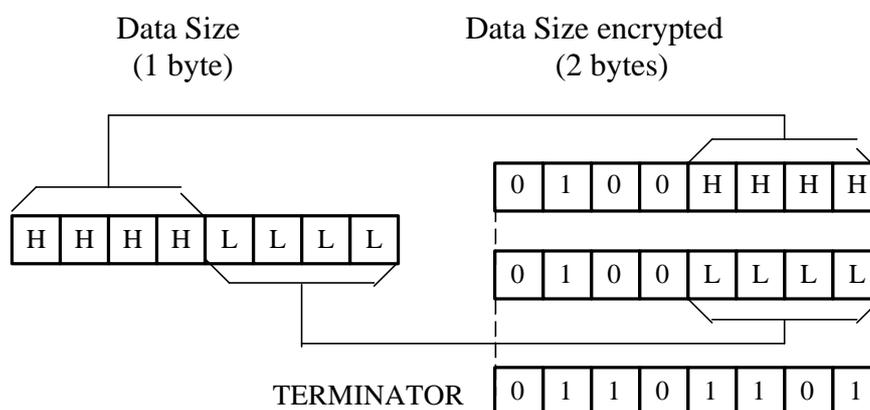
DATA is the parameters that OBDM need to execute the command.

Each Data byte is represented by two bytes: the first byte contains the code that identify the type of byte (0100) and the high nibble (four most significant bits) of the Data byte, the second byte contains the code that identify the type of byte (0100) and the low nibble of the Data byte:



CHECK SUM is the sum without carry of data byte.

This field is represented by three bytes: the first byte contains the code that identify the type of byte (0101) and the high nibble (four most significant bits) of the Data Check Sum value, the second byte contains the code that identify the type of byte (0101) and the low nibble of the Data Check Sum value, and the third byte contains the "terminator" (01101101) :



		<p>PFS for Mars Express</p>	<p>PFS-FUM 3 Page 18</p>	<p>P.I. Vittorio Formisano CNR IFSI</p>
---	---	-----------------------------	------------------------------	---

DATA and *CHECK SUM* field is absent if *DATA SIZE* is 0.

Usage of such encryption allows to avoid confusion in the case when some bytes are lost (or spurious bytes are inserted) due to hardware errors. Since there is only one OBDM command which transfers many data, and this command is used only once during the session, such an encryption will not slow down the data transfer.

If OBDM begins to analyze a byte sequence and does not detect the command Code (the Code field is lost or a spurious byte is inserted), OBDM must ignore all bytes until the command Code byte is encountered.

4.3 Decodify commands sending from DAM

The operating system recognize the commands and call the corresponding subroutine that execute all the action to accomplish the command.

4.4 Failure mode commands

There are several failure modes which are implemented to overcome possible problems : broken laser diode, broken A/D converter, no zopd signal.

Broken/misalignment of laser

We switch to the other one (normally the motion is controlled by the SW laser), we can use also the LW laser for motion control. Furthermore the interferogram signal is sampled by using each laser for each channel, but PFS can also make use of one single laser for sampling both channels.

Broken A/D converter

We switch to the otherone (normally each channel use its own ADC, but in failure mode each ADC can be used for both).

No Zopd pulse

In this case the entire motion control would not work, and the experiment would fail : we can have a software routine that gives an order for the motion to start, and eventually correct the zopd position after one or more motions. A flag should indicate this way of working.

5. OBDM Commands

Following table describes commands which DAM sends to OBDM:

Code (HEX)	Data Size (byte)	Comando	Data
11H	0	Unblock pendulum	No
12H	0	Block pendulum	No
13H	0	Send selftest data	No
14H	32	Load Control Table	Control Table
15H	0	Pendulum in Initial position	No
17H	1	Start Autotest	Autotest parameters
16H	0	Send status	No
18H	0	Start Acquisition	No
19H	0	Send HouseKeeping block	No
1AH	2	Send ShortWave block	SW block number
1BH	2	Send LongWave block	LW block number
1CH	0	End of Acquisition	No
1EH	0	Get synt.interf. & HK	No

Messages which OBDM sends to DAM in response to the command are following:

Code (HEX)	Data Size (byte)	Comando	Data
11H	0	Pendulum unblocked	No
12H	0	Pendulum unblocked	No
13H	48	Selftest result	Selftest data
14H	0	Control Table loaded	No
15H	0	In initial position	No
16H	0	Autotest completed	No
17H	32	OBDM status data	OBDM Status
18H	0	Acquisition completed	No
19H	128	Housekeeping block	HK block
1AH	128	Shortwave block	SW block
1BH	128	Longwave block	LW block
1CH	0	Acquisition terminated	No
1EH	0	Synt.interf. & HK ready	No
2EH	0	Error in command	No
2FH	0	Mechanical problems	No
99H	0	Bootstrap completed	No

Appearance of a complete OBDM message (except "Error in command") is treated by DAM as a completion (normal or abnormal) of the command execution.

6. ICM Software

The ICM is an FFT unit inside DAM. ICM performs the Fourier transform of the interferograms which DAM obtains from the OBDM.

There are 3 different modes of the ICM operations:

4K - for the 4Kw LW interferograms

4KS - single bank, - input/output is the same as for 4K, but all operations are performed in the specified bank of the ICM memory

16K - for the 16Kw SW interferograms

ICM provides with either module (power) or sine/cosine spectra. If required, ICM also performs an apodisation and an average suppression.

5.1 Protocol of the data exchange between DAM and ICM.

DAM controls over ICM operations sending the ICM Control Word. The high byte sets the mode of ICM operations:

B ₁	B ₀	S ₁	S ₀	A _p	A _v	O ₁	O ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

where

B₁ B₀ specify ICM mode and memory bank:

0 0 4KS, bank 0

0 1 4KS, bank 1

1 0 4KS, bank 2

1 1 4K

S₁ S₀ specify the interferogram size:

0 0 not used

0 1 4K (LW)

1 0 not used

1 1 16K (SW)

A_p switches apodisation on(1)/off(0)

A_v switches average suppression on(1)/off(0)

O₁ O₀ specifies the ICM output (type of the spectrum):

0 0 module (power)

0 1 not used

1 0 nor used

1 1 sine/cosine

The low byte of the ICM Control Word is meaningless except LSB, which must be set. Having the Control Word sent, DAM programs the DMA channel 2 for the uploading of the interferogram.

ICM, upon reception of the Command Word, sends the ICM Status Word. The high byte of the ICM Status Word is the same as in the ICM Control Word. The low byte contains an information about the ICM status:

Pc	0c	1c	2c	T			0
----	----	----	----	---	--	--	---

where

Pc=1 means PROM checksum error

0c=1 means memory bank 0 error

1c=1 means memory bank 1 error

2c=1 means memory bank 2 error

T=1 means timeout during the data reception

The reading of the ICM Status Word by DAM is a signal for ICM to start the interferogram transfer. DAM clears the mask of the DMA channel 2. The data transmission is performed by the DMA controller, and the ICM starts the computation.

After finishing the Fourier transform, ICM sends the ICM Status Word which informs DAM that the obtained spectrum can be downloaded from the ICM. DAM programs the DMA channel 2 for the reception of the spectrum. The reading of the ICM Status Word by DAM is a signal for ICM to start the spectrum transfer. DAM clears the mask of the DMA channel 2. The data reception is performed by the DMA controller.

5.2 Structure of the spectra obtained from ICM

Depending on the mode, ICM provides with either module or Re/Im parts of the spectrum. ICM transfers spectra in the following order:

interf size	mode	spectr type	ICM output
4K	4K	module	2K module
4K	4K	sin/cos	2K Im + 2K Re
4KS	4K	module	2K module
4KS	4K	sin/cos	2K Re + 2K Im
16K	16K	module	8K module
16K	16K	sin/cos	8K Im + 8K Re

Each module or Re/Im part contains the housekeeping information in the first 6 words:

word 1	blk_exponent
word 2	sum_exponent
word 3	rcv_size
word 4	blk_exponent
word 5	sum_exponent
word 6	rcv_size

Words 1 and 4, 2 and 5, 3 and 6 are equal and can be used for the check of the data correctness. The value of the rcv_size must be either 4096 or 16384 (depending on the interferogram size). Two other values allow to determine the scaling factor:

$$\text{Scale} = 2^{\text{blk_exponent}} / 2^{\text{sum_exponent}}$$

Other words are spectral data in the signed integer (2's complement) format.

7- Memory Map

Here we present only the DAM memory map.
The memory is organized in 4 bank :

00000H – 0FFFFH	RAM0-BANK0	DATA-[32k]TM buffer-[8k]LW interf.-[8k]LW spectra	65536 B
10000H – 1FFFFH	RAM1-BANK1	[32K]SW interf.- [32K]SW spectra	65536 B
20000H – 2FFFFH	RAM2-BANK2	[32K]SW interf.prev.- [32K]LW interf.prev.	65536 B
30000H – 3FFFFH	RAM3-BANK3	Code	65536 B
F0000H – FFFFFH	ROM	Permanent Code	65536 B

RAM0 - BANK0

CPU area		
00000H - 003FFH	Interrupt Vectors	1024 Byte
00400H - 007FFH	Stack Area	1024 Byte
Data area		
00800H - 008DFH	DAM Status	224 Byte
008E0H - 008FFH	OBDM Status info	32 Byte
00900H - 0097FH	OBDM HK info	128 Byte
00980H - 0099FH	OBDM Control Table (to be loaded)	32 Byte
009A0H - 009DFH	Received Telecommands: T/S Seq	64 Byte
009E0H - 009EFH	Merely to align tables at 16b boundary	16 Byte
009F0H - 00A0FH	Default OBDM Control Table (not to be loaded)	32 Byte
00A10H - 00BCFH	Dummy for patch	448 Byte
00BD0H - 00C4FH	MH1	128 Byte
00C50H - 00CCFH	MH2	128 Byte
00CD0H - 00DCFH	MH3	256 Byte
00DD0H - 00E4FH	Previous MH1	128 Byte
00E50H - 00ECFH	Previous MH2	128 Byte
00ED0H - 00FCFH	Previous MH3	256 Byte
00FD0H - 0104FH	Last MH1	128 Byte
01050H - 010CFH	Last MH2	128 Byte
010D0H - 011CFH	Last MH3	256 Byte
011D0H - 011D3H	Head and Tail of OBDM Output Buffer	4 Byte
011D4H - 013D3H	OBDM Output Buffer	512 Byte
013D4H - 013D9H	Head, Tail and Pointer of OBDM Input Buffer	6 Byte
013DAH – 023C9H	OBDM Input Buffer	4096 Byte
023DAH – 023DBH	Number of bytes to read in polling mode	2 Byte
023DCH – 023DDH	OBDM Selftest info length	2 Byte
023DEH – 0240DH	OBDM Selftest info	48 Byte
0240EH – 0240FH	OBDM Status info length	2 Byte
02410H – 02411H	OBDM HK info length	2 Byte
02412H – 02419H	OBDM ZOPD	8 Byte
0241AH – 0241DH	Initialization test variable, Monitor cycle counter	4 Byte
0241EH – 02428H	Mass Memory status	11 Byte
02429H	Specified DTM	1 Byte

 Planetary Fourier Spectrometer PFS		PFS for Mars Express	PFS-FUM 3 Page 24	P.I. Vittorio Formisano CNR IFSI
--	---	----------------------	----------------------	-------------------------------------

<i>0242AH – 0242FH</i>	Dummy for alignment	6 Byte
<i>02430H</i>	DTM of previous measurement	1 Byte
<i>02431H</i>	DTM of pre-previous measurement	1 Byte
<i>02432H – 02435H</i>	Number of last stored measurement and shift	4 Byte
<i>02436H – 0243DH</i>	ICM status	8 Byte
<i>0243EH – 02845H</i>	Memory Load Buffers and Pointers	1032 Byte
<i>02846H – 034F1H</i>	TC, Event and Dump Buffers, Pointers and Variables	3243 Byte
<i>034F2H – 034F5H</i>	Table of calibration positions and pointer	4 Byte
<i>034F6H – 034F7H</i>	Count of Calibration (CalMode=6..8)	2 Byte
<i>034F8H – 034F9H</i>	Number of Gain(Test5) and Number of Temp(Test6)	2 Byte
<i>034FAH – 034FBH</i>	Current SW and LW Laser Temperat (for Test 6)	2 Byte
<i>034FCH – 034FDH</i>	TC to move the Scanner and Flag "Scan. was moved"	2 Byte
<i>034FEH – 034FFH</i>	Flag "Mod. O and Scanner moved to Initial Position"	2 Byte
<i>03500H – 03501H</i>	Number of Science Report (0..40)	2 Byte
<i>03502H</i>	New CPU Code Segment	1 Byte
<i>03503H</i>	Current CPU Code Segment	1 Byte
<i>03504H</i>	OBDM message code	1 Byte
<i>03505H</i>	OBDM message status	1 Byte
<i>03506H</i>	Flag: "New OBDM commands"	1 Byte
<i>03507H</i>	Flag: "Use polling talking with OBDM"	1 Byte
<i>03508H – 03509H</i>	Pointer in the Test Buffer	2 Byte
<i>0350AH – 03D09H</i>	Test Buffer	2048 Byte
<i>03D0AH – 03D1DH</i>	Offset, length and checksum of MH,SW and LW area	20 Byte
<i>03D1EH – 03D1FH</i>	TM block num (used by CheckFIFO)	2 Byte
<i>03D20H – 03D23H</i>	State of TMblock1	2 Byte
<i>03D24H – 03D25H</i>	Merely for the mem. distrib. control	2 Byte
<i>03D26H – 03FFFH</i>	Free Necessary to align TMblocks at 16K of DATAseg	730 Byte
TMblock area		
<i>04000H – 07FFFH</i>	TMblock1	16384 Byte
<i>08000H – 0BFFFH</i>	TMblock2	16384 Byte
LW interferogram and spectra area		
<i>0C000H – 0DFFFH</i>	LW interferogram	8192 Byte
<i>0E000H – 0FFFFH</i>	LW spectrum	8192 Byte

RAM1 – BANK1

SW interferogram area		
<i>10000H – 17FFFH</i>	SW interferogram	32768 Byte
SW spectra area		
<i>18000H – 1FFFFH</i>	SW spectrum	32768 Byte

 Planetary Fourier Spectrometer PFS		PFS for Mars Express	PFS-FUM 3 Page 25	P.I. Vittorio Formisano CNR IFSI
---	---	-----------------------------	-----------------------------	---

RAM2 – BANK2

SW interferogram area		
<i>20000H – 27FFFH</i>	SW interferogram previous	32768 Byte
LW interferogram area		
<i>28000H – 2FFFFH</i>	LW interferogram previous	32768 Byte

RAM3 – BANK3

CODE area		
<i>30000H – 34A93H</i>	CODE copy from ROM	19092 Byte
<i>34A94H – 3FFFFH</i>	Free for patch	46444 Byte

ROM

CODE area		
<i>F0000H – F4A93H</i>	Permanent CODE	19092 Byte
<i>F4A94H – FFFFFH</i>	Free	46444 Byte