MCU /DCU

Command List ICD

Prepared by :	Date :
Michel Jevaud Didier Ferrand	20/09/2001
Checked by :	**Date :**
Dominique Pouliquen	20/09/2001

Document change record

Issue - rev.	Date	Modifications
1 - 0	20-09-00	First issue

Distribution list

Jean Louis Auguères	CEA	X	
Christophe Cara	CEA	X	
Brian Stobie	ROE	X	
Ian Pain	ROE	X	
Didier Ferrand	LAM	X	
Michel Jevaud	LAM	X	
Dominique Pouliquen	LAM	X	
Bruce Swinyard	RAL	X	
Ken King	RAL	X	
Ricardo Cerrulli	IFSI	X	

Acronyms

AD	Applicable Document
AVM	Avionics Model
BOL	Begin Of Life
BSM	Beam Steering Mirror
CQM	Cryogenic Qualification Model
EGSE	Electrical Ground Support Equipment
EOL	End of Life
ESA	European Space Agency
FIRST	Far Infra Red and Sub-millimeter Telescope
FM	Flight Model
FPU	Focal Plane Unit
FTS	Fourier Transform Spectrometer
FTSE	FTS warm Electronics
FTSP	FTS Preamplifier for the position encoder signals
H/K	House Keeping
H/W	Hardware
I/F	Interface
LAM	Laboratoire Astrophysique de Marseille
MAC	Multi Axes Controller
MCU	Mechanism Control Unit
N/A	Not Applicable
RAL	Rutherford Appleton Laboratory
RD	Reference Document
ROE	Royal Observatory of Edinburgh
S/C	Spacecraft
SM	Spare Model
SMEC	Spectrograph MEChanism
S/W	Software
TBC	To Be Confirmed
TBD	To Be Define
TBW	To Be Written
TC	Tele-Command
TM	TeleMetry
WE	Warm Electronics

Table of contents

INTRODUCTION

1.1 PURPOSE AND SCOPE

The purpose of this document is to describe the commands which can be sent by the DPU to MCU in order to control and monitor the 3 motorised axis of SPIRE (BSM Chopper / Jiggle axis and SMEC axis)

The control of the 3 axis is performed by a 21020 DSP on the basis of trajectory generators and digital PID controllers associated with filtering for notching of mechanism modes. The control parameters are put in memory for configuration purpose by mean of a command line (bi-directional 32 bit/330 kHz synchronous serial line) connected with the DPU. The software shall be based on a master scheduler on the principle of time sharing without the use of a specific multitask kernel. The tasks to be performed shall be called on a software interrupt generated by the inner DSP timer. The software interrupt defines the global sampling time (i.e. the computation cycle) of the DSP tasks @ a programmable rate between 100 us min and 300 us max[1] . At each cycle the following tasks are performed :

- the SMEC control loop task
- the chopper control loop task,
- the jiggle control loop task,
- the communication with the command line and other various internal DSP tasks,
- telemetry packet concatenation and transmission to high rate serial link

1.2 APPLICABLE AND REFERENCE DOCUMENTS

1.2.1 Applicable documents

AD1	Operating Modes for the SPIRE Instrument (SPIRE-RAL-DOC-000320)
AD2	FIRST/Planck Packet Structure Interface Control Document (SCI-PT-IF-07527)
AD3	Spire Spectrometer Mirror Mechanism Subsystem Specification (SPIRE-LAM-PRJ-000460)
AD4	FIRST / Planck Instrument Interface Document Part B (SCI-PT-IIDB/SPIRE-02124)
AD5	DRCU Electrical Interface Control Document (SAp-SPIRE-CCa-24-00)
AD6	SPIRE Instrument Requirements Specification (IRD) (SPIRE-RAL-PRJ-000034)
AD7	DPU Interface Control Document SPIRE-IFS-PRJ-000650 2 April 2001 Issue 1.0

1.2.2 Reference documents

RD1	Beam Steering Mirror Control Software Requirements (Spire-ATC-Draft)
RD2	Beam Steering Mirror Warm Electronics (Spire-ATC-Draft)
RD3	Beam Steering Mirror Electronics Electrical Interface (Spire-ATC-Draft)

2 COMMAND TYPES

There are typically three types of commands :

- **Set Command** : the parameter is put in the DSP memory for immediate or delayed execution
- **Get Command** : the dedicated parameter in memory can be read by the DPU
- **Immediate Command:** Reset Command or Start Datation Counter

2.1 COMMAND GENERAL FORMAT

The command format specification shall be compatible with DPU Interface Control Document [AD6].
A command is a 32 bits word with the following allocation:

Bit Alloc	31-30	Bits 29-28	Bits 27	Bits 26-25	Bits 24-16	Bits 15-0
Function	Sync Signals	MCU header	Set or Get	Axis id.	Command type	Parameter
Meaning	10:Reply 11: No reply	01 is the MCU address	Set: write parameter Get: read parameter	0: all axis 1:SMEC axis 2:Chopper axis 3: Jiggle axis	see mnemonic list	16 bits parameter

2.2 HOW TO CONCATENATE COMMANDS ON THE SERIAL LINE ?

The following example shows how to send bit pattern on the command line for the first steps of the initialisation on the SMEC control axis.

#1	Power on the optical encoder				**SetEncoderPwr(on)**			
Sync	SS @	12 Bits command mnemonic			16 Bits parameter			
1 0	0 1	0 0 0 0 0 0 0 0 0 0 0 0			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1			
9		0	0	0	0	0	0	1

#2	Power on the LVDT Oscillator				**SetLVDTPwr(on)**			
Sync	SS @	12 Bits command mnemonic			16 Bits parameter			
1 0	0 1	0 0 0 0 0 0 0 0 0 0 0 1			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1			
9		0	0	1	0	0	0	1

#3	Close the control loop on optical encoder on steady position				**SetLoopMode (on optical encoder)**			
Sync	SS @	12 Bits command mnemonic			16 Bits parameter			
1 0	0 1	0 0 0 0 0 0 0 0 0 0 0 1			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1			
9		0	0	1	0	0	0	1

Etc .....

2.3 BUFFERED COMMANDS AND PARAMETERS

Some parameters set by a Set Command should not take immediate effect on the control. These parameters are called **buffered parameters**. They are taken into account when a specific **Start** command is sent. This is useful for a configuration set definition such as the scan speed and length before a validation of the configuration. The buffered parameters relates to the motion trajectory, control loop and telemetry configuration set up parameters.

2.4 ACTIVITY STATUS REGISTER

The activity status register is a 16 bits word which is updated at each DSP cycle. This register reflects the status of the dedicated axis and can be read by the GetSMECStatus, GetChopStatus or Get Jiggle Status command. This can be used by the DPU to obtain the status of the dedicated axis in order for check the and diagnostic purpose.There is one activity register per axis.

Bit	Name	Description
0	**Motion Complete**	Set when a trajectory profile/scan is complete
1		
2		
3		
4	**Motion error**	Set when the actual position differs from the commanded position by an amount more than the specified maximum position error
5	**Positive limit**	Set to 1 when the positive limit switch is active
6	**Negative limit**	Set to 1 when the negative limit switch is active
7	**Instruction error**	Set when an instruction error occurs
8	**Invalid parameter**	
9	**@ Maximum velocity**	Set to 1 when the trajectory is @ maximum velocity. Set to 1 when the axis is within the tracking window
10		
11		
12	**Axis Initialized**	
13	**Motor Mode on/off**	
14		
15	**Time out of Command link FPGA**	The command line FPGA sends to DPU the error status word with this bit set to one if the DSP does not read the command within a time out.

3 COMMAND LIST

The complete list of parameters associated with the Set or Get command is listed after. The parameter numbering is up to now given as examples and may change during development until the first software issue.

3.1 GENERAL COMMAND MNEMONIC MAPPING

Commands parameters are put at dedicated address in the DSP memory for immediate or postponed execution (on START command). The symbolic address in memory is determined by the command mnemonic. The general mapping is defined hereafter.

3.1.1 SET COMMANDS

000h	07Fh	SMEC Init Set commands
080h	0FFh	SMEC General set commands
100h	17Fh	SMEC Control loop and trajectory tuning set commands
180h	1FFh	free: dedicated to not writable SMEC HKs
200h		Chop Init Set commands
280h		Chop General Set commands
300h		Chop Control loop and trajectory tuning Set commands
380h	3FFh	free: Chop HKs
400h		Jiggle Init Set commands
480h		Jiggle General Set commands
500h		Jiggle Control loop and trajectory tuning Set commands
580h	5FFh	Jiggle HKs
600h	67Fh	Telemetry and Trace configuration set commands
680h	7FFh	HK and Miscellaneous Set Commands

3.1.2 GET COMMANDS

800h	87Fh	SMEC Init Get commands
880h	8FFh	SMEC General Get commands
900h	97Fh	SMEC Control loop and trajectory tuning Get commands
980h	9FFh	SMEC HKs Get commands
A00h	A7Fh	Chop Init Get commands
A80h	AFFh	Chop General Get commands
B00h	B7Fh	Chop Control loop and trajectory tuning Get commands
B80h	BFFh	Chop HKs Get commands
C00h		Jiggle Init Get commands
C80h		Jiggle General Get commands
D00h		Jiggle Control loop and trajectory tuning Get commands
D80h	DFFh	Jiggle HKs Get commands
E00h	E7Fh	Telemetry and Trace configuration Get commands
E80h	FFFh	HK and Miscellaneous Get Commands

3.2 SMEC COMMANDS

3.2.1 Initialisation set commands

Com. Mnemonic	Com. Name	Action
000h	**SetEncoderPwr**	Power on the encoder led. The led has 8 levels encoded on 3 bits from 0 to max.
001h	**SetLVDTPwr**	Power on/off the LVDT oscillator (Boolean)
002h	**SetLoopMode**	Open or close loop mode selection Parameter definition: 00 = Loop opened (0020000h) 01 = Loop closed on Back EMF(0020001h) 10 = Loop closed on LVDT(0020002h) 11 = Loop closed on optical encoder(0020003h)
003h	**SetLaunchLatch1**	Engage/disengage Launch Latch 1 (0 or 1)
004h	**SetLaunchLatch2**	Engage/disengage Launch Latch 2 (0 or 1)
005h	**SetInitMotion**	Starts the motion initialisation procedure (0 or 1)

3.2.2 General configuration commands

080h	**SetScanStart**	specifies the scan starting position
880h	**GetScanStart**	Reads the start position
081h	**SetScanSpeed**	specifies the velocity to reach the position target (scan speed)
881h	**GetScanSpeed**	
082h	**SetScanEnd**	
882h	**GetStartScan**	0: Scan commanded to stop 1: Scan commanded to run
083h	**Update**	**Cancelled: the update shall be taken into account on each SetStartScan Command**
084h	**SetScanMode**	0: motion stopped 1:position step 2: sawtooth motion 3:triangular motion
085h	**SetScanNumber**	Specifies the number of scans to be performed

Starts/Stops the scan. (start of trajectory generation and the closed loop control scan according to Scan parameters)

3.2.3 Control loop and trajectory tuning specific commands

Command Mnemonic	Command Name	Action
100h	**SetKpHigh**	Proportional gain of the digital PID controller
101h	**SetKpLow**	
102h	**SetKdHigh**	Derivative Gain of the digital PID controller
103h	**SetKdLow**	
104h	**SetDerivFilterHigh**	Sets the filtering time constant to calculate the derivative term.
105h	**SetDerivFilterLow**	
106h	**SetKiHigh**	Integral gain of the digital PID controller
107h	**SetKiLow**	
108h	**SetIntegrationLimit**	Loads/Reads the integration saturation for the integral compensation of the servo
109h	**Set PositionErrorLimit**	determine/reads the position error value that causes an error on the servo
10Ah	**SetNotchParamHigh**	Parameters for notch filtering of mechanical modes. Here is the pole compensation of the first parasitic mechanical mode
10Bh	**SetNotchParamLow**	

3.2.4 HK specific get commands

980h	**GetSmecStatus**	reads the Activity Status Register. This Status Register can only be read and no bit can be cleared since the word is refreshed by the chipset.
981h	**GetActualPosition**	returns the last absolute position
982h	**GetActualVelocity**	Get the instantaneous (20 Hz filtered) velocity
983h	**GetMeanSpeed**	Actual scan mean measured speed Allows to verify the velocity scan error
984h	**GetMeanPositionError**	Returns the mean position error over a scan

3.3 CHOP COMMANDS

To be Completed

3.3.1 Initialisation set commands

Com. Mnemonic	Com. Name	Parameter / Action
200h	**SetSensorPwr**	On/Off (0 /1) : Power on magnetoresistive sensor.
201h	Reserve	
202h	**SetChopLoopMode**	Open or close loop mode selection Parameter definition: 00 = Loop opened (2020000h) 01 = Loop closed on Back EMF(2020001h) 10 = Loop closed on magnetoresistive(2020002h)
203h	**SetChopLaunchLatch**	Engage/disengage Launch Latch (0 or 1)

3.3.2 General configuration commands

280h	**SetPosition0**	specifies the y0 position step
A80h	**GetPosition0**	
281h	**SetPosition1**	specifies the y1 position step
A81h	**GetPosition1**	
282h	**SetChopPeriod**	Specifies the chopping cycle
A82h	**GetChopPeriod**	
284h	**SetChopMode (alias MoveChop)**	0: Chopper commanded to stop 1: Chopper commanded to run in step mode 2: Chopper commanded to run in toggle mode
285h	**SetScanNumber**	Specifies the number of scans to be performed

3.3.3 Control loop and trajectory tuning specific commands

Command Mnemonic	Command Name	Action
300h	**SetKpHigh**	Proportional gain of the digital PID controller
301h	**SetKpLow**	
302h	**SetKdHigh**	Derivative Gain of the digital PID controller
303h	**SetKdLow**	
304h	**SetDerivFilterHigh**	Sets the filtering time constant to calculate the derivative term.
305h	**SetDerivFilterLow**	
306h	**SetKiHigh**	Integral gain of the digital PID controller
307h	**SetKiLow**	
308h	**SetIntegrationLimit**	Loads/Reads the integration saturation for the integral compensation of the servo

309h	**Set PositionErrorLimit**	determine/reads the position error value that causes an error on the servo

3.3.4 HK specific get commands

B80h	**GetChopStatus**	reads the Activity Status Register. This Status Register can only be read and no bit can be cleared since the word is refreshed by the chipset.
B81h	**GetActualPosition**	returns the last absolute position
B82h	**GetMeanPositionError**	Returns the mean position error over a scan

TO BE COMPLETED

3.4 JIGGLE COMMANDS

Idem than Chopper commands with command mnemonics + 200h

3.5 TELEMETRY AND TRACE CONFIGURATION COMMANDS

#	Parameter	Action
600h	**SetTelemetry**	Activate or inhibits the possible predefined packets to be sent
601h	**SetTelemetrySampling**	Set the sampling rate of the telemetry packets
602h	**Set TraceSampling**	Sets the time period, expressed in ms, between successive trace points
603h	**SetTraceBuffer**	Sets the length of the trace data buffer.
604h	**SetTraceParam#1**	parameter identifier to be put in the trace frame up to 6 values In the trace frame, the 2 first data are the data name and the data sampling.
605h	**SetTraceParam#2**	
606h	**SetTraceParam#3**	
607h	**SetTraceParam#4**	
608h	**SetTraceParam#5**	
609h	**SetTraceParam#6**	

3.6 HK AND MISCELLANEOUS COMMANDS

680h	**SetDPUPollingTime**	Set a maximum time between 2 commands before a IO Error code is generated, meaning a communication problem
E81h	**ReadMemory**	Returns the value of whatever DSP memory mapping

4 TYPICAL COMMANDING SCENARIOS

4.1 SMEC COMMANDING SCENARIOS

4.1.1 SMEC INITIALISATION PROCEDURE

Previous State: The Main Power supply switch on

	Command to be sent	Action	Remarks - Error switch to degraded mode procedure
1	**SetEncoderPwr(on)**	Power on the optical encoder led	
2	**GetEncoderSignalsStatus()**	Verify if the optical encoder works properly	Goto procedure #2 'Initialisation in degraded mode'
3	**SetLoopMode (on optical encoder)**	Close the loop on optical encoder	It is recommanded to move the SMEC while latched thanks to the latch clearance of few 10 microns
4	**GetSMEC Status()**	Verify if the loop is closed properly	
5	**SetLVDTPwr(on)**	Power on the LVDT Oscillator	
6	**GetLVDTStatus**	Verify the oscillator works properly	
7	**SetLaunchLatch1(off)**	Disengage Launch Latch 1	
8	**GetLaunchLatch1Status**	Verify the first launch latch disengagement	
9	**SetLaunchLatch2(off)**	Engage/disengage Launch Latch 2	
10	**GetLaunchLatch2Status**	Verify the launch latch #2 disengagement	
11	**SetInitMotion(on)**	Starts the motion initialisation procedure (0 or 1)	Consists in applying the mechanism on mechanical stop for absolute position reference capture
12	**GetSMECStatus**	Verify if the motion is properly initialised	

AT THIS STAGE, THE SMECm IS INITIALISED, WAITING IN CLOSED LOOP CONTROL AT HOME POSITION. Next possible State: operating the SMEC in nominal mode

4.1.2 Commanding the SMEC

#	Command to be sent	Parameter	Remark
1	**SetScanMode**	0	SMEC stopped
2	**SetScanSpeed**	Scan Speed	
3	**SetScanNumber**	Number of scan	
4	**SetScanStart**	number of counts from home at which to start scientific part of scan	
5	**SetScanEnd**	number of counts from home position of end of scientific part of scan	
6	**SetTelemetrySampling**	Bit related to SMEC telemetry packet and encoder count sample value	
7	**SetTelemetry**	Bit related to SMEC telemetry packet = 1	generate a SMEC data sample at every i counts
8	**SetScanMode**	on: can be sawtooth,triangular,step	The motion is started with set up parameters

- If the scan is running while configuration is set up, parameters shall not be taken into account and the command reply shall indicate an error to DPU
- This operation should be followed by a flush_fifo command to the DPU
- Scan quality information should be put in memory for reading as housekeeping data (it will need to be associated with a scan number)

4.2 COMMANDING THE CHOPPER

Chopping can be commanded in the MCU in the following ways:
CHOP(y0,y1)

4.2.1 Automatic

#	Command to be sent	Parameter	Remark
1	**SetChopPosition0**	(y0)	
2	**SetChopPosition1**	(y1)	
3	**SetChopPeriod**	Period	
4	**SetTelemetrySample**	Sample interval	
5	**SetChopSamples**	number of samples per chop position	
6	**SetChopCycles**	Number of steps 0 = indefinite chopping	
7	**SetChopMode**	1 : Automatic Step mode	Update parameters and starts chopping

4.2.2 Command Driven

#	Command to be sent	Parameter	Remark
1	**SetChopPosition0**	y0	
2	**SetChopPosition1**	y1	
3	**SetChopPeriod**	Period	
4	**SetTelemetrySample**	Sample interval	
5	**SetChopSamples**	number of samples per chop position	
6	**SetChopMode**	2 :Toggle Mode	Update parameters and starts chopping
7	**SetChopMode**	2 :Toggle Mode	Move the chopper
8	**SetChopMode**	2 : Toggle Mode	Move the chopper
9	**SetChopMode**	Toggle Mode	Move the chopper

	etc		

4.3 COMMANDING THE JIGGLE AND CHOPPER

Jiggle is commanded in the following way:
Input: z, y0,y1 for each jiggle position
JIGGLEANDCHOP(z,y0,y1,……)

#	Command to be sent	Parameter	Remark
1	**SetJigPosition**	z	
2	**SetChopPosition0**	y0	
3	**SetChopPosition1**	y1	
4	**SetJigMode**	1 :Step Mode	This updates parameters and Move jiggle
5	**SetChopMode**	1 :Step Mode	Move the chopper
6	**SetJigPosition**	z’	
7	**SetChopPosition0**	y0’	
8	**SetChopPosition1**	y1’	
9	**SetJigMode**	1 :Step Mode	This updates parameters and Move jiggle
10	**SetChopMode**	1 :Step Mode	Move the chopper
	Etc....		