Communication between CPU, HK-O and HK-N

Housekeeping system is able to turn on/off any subsystem, by request of CPU, and also to log and monitor different parameters of all subsystems.

HK-O and HK-N have a capability to send/receive data to/from CPU. CPU can send commands, read values and configure the subsystems operation thru HK-O and HK-N.

The commands and parameters for HK-O were implemented and updated on February 2014 and the communication between CPU and the HK-N are a new protocol proposal.

The protocols between CPU and HK-O are shown in table 1.

Table 1

Description	Header	End	Origin to destiny	Number of the bytes un the frame
Send a command	0x31	0x17	CPU to HK-O	5
Read a value	0x32	0x17	CPU to HK-O	6
Response for a value reading	0x32	0x17	HK-O to CPU	8
Configure parameter	0x33	0x17	CPU to HK-O	9
Low Priority Alarms	0x34	0x17	HK-O to CPU	9
High Priority Alarms	0x35	0x17	HK-O to CPU	5
NEW Data reading	0x36	0x17	HK-O to CPU	5
NEW Contact Closure info	0x37	0x17	HK-O to CPU	5

The proposed protocols between CPU and HK-N are based on the last table and shows in table 2

Table 2

Description	Header	End	Origin to destiny	Number of the bytes un the frame
Send a command	0x31	0x17	CPU to HK-N	5
Read a value	0x32	0x17	CPU to HK	6
Response for a value reading	0x32	0x17	HK-N to CPU	8
Low Priority Alarms	0x34	0x17	HK-N to CPU	9
High Priority Alarms	0x35	0x17	HK-N to CPU	5
Contact Closure info	0x37	0x17	HK-N to CPU	5

The CRC polynomial continues being the same one on previous documentation:
CRC (16 bits):

0xAC9A $= x_{16} + x_{14} + x_{12} + x_{11} + x_8 + x_5 + x_4 + x_2 + 1$

10101100100110101

0101 10001 0011 0101 = 0x5935

Send a command

When HK receives the header value 0x31h (Send a command), it knows it is a function command. The bytes structure is shown in table 3.

Header	Command	End	CRC16H	CRC16L
0x31	0x65	0x17	0xFE	0x68

Table 3.- Example: Send a command to HK-O, in order to turn on CLKB

The table 4 has the commands that HK-O can recognize of CPU, and the table 5 contains the commands of between HK-N and CPU.

Table 4

Hexadecimal	Command	System
0x65	Turn ON	CLKB
0x71	Turn ON	CCB
0x74	Turn ON	GPS
0x7A	Turn ON	PDM
0x63	Turn ON	EC
0x64	Turn ON	DST
*0x61	Turn ON	CPU
0x72	Turn OFF	CLKB
0x77	Turn OFF	CCB
0x79	Turn OFF	GPS
0x78	Turn OFF	PDM
*0x76	Turn OFF	EC
0x66	Turn OFF	DST
0x73	Turn OFF	CPU
0x48	Program_B	CCB
0x4A	Program_B	CLKB
0x49	Program_B	PDM

Hexadecimal	Command	System
0x4F	Reset Line	CCB
0x42	Reset Line	CLKB
0x4E	Reset Line	PDM
0x30	ASK for CC Status	Relays
0x31	Packet Data Request	Temp HK/CPU
0x32	Packet Data Request	LVPS1-DP
0x33	Packet Data Request	LVPS2-DP
0x34	Packet Data Request	LVPS-HK
0x35	Packet Data Request	LVPS-PDM
0x36	Packet Data Request	CCB
0x37	Packet Data Request	CLKB
0x38	Packet Data Request	GPS
0x39	Packet Data Request	PDM
0x3A	Packet Data Request	HVPS
0x3B	Packet Data Request	LENSES

* The commands to turn on/off EC are obsolete, because in the new design of LVPS-PDM, only require activate/desactivate one DC-DC converter to provide the power of the PDM and EC system.

Table 5

Hexadecimal	Command	System
0x51	Turn ON	Cooler
0x52	Turn ON	IR-C
0x53	Turn ON	Photodiode
0x54	Turn ON	Health LED 1
0x55	Turn ON	Health LED 2
0x56	Turn ON	SiPM
0x57	Turn ON	Compass
0x58	Turn ON	HAM Xmitter
0x59	Turn ON	HVPS
0x2C	Reset Line	CCB
0xCC	ASK for CC Status	Relays
0x2B	Packet Data Request	Battery

Hexadecimal	Command	System
0x41	Turn OFF	Cooler
0x4C	Turn OFF	IR-C
0x43	Turn OFF	Photodiode
0x43	Turn OFF	Health LED 1
0x44	Turn OFF	Health LED 2
0x45	Turn OFF	SiPM
0x46	Turn OFF	Compass
0x47	Turn OFF	HAM Xmitter
0x4B	Turn OFF	HVPS
0x2F	Packet Data Request	LVPS-AN
0x2E	Packet Data Request	Temp HK

Read a value from any parameter measured by HK

CPU can ask HK-O and HK-N the value of some specific parameter. The bytes structure of this command is shown in table 6.

Header	Command	Parameter	End	CRC16H	CRC16L
0x32	0x11	0x81	0x17	0x37	0x78

Table 6.- Example: CPU ask HK-O for the Thermistor 1 value.

The table 7 has the parameters that HK-O can recognize of CPU, and the table 8 contains the parameters of between HK-N and CPU.

Table 6

		Parameter	HK-O	LVPS1 DP	LVPS2 DP	LVPS HK-O	LVPS PDM	CCB	GPS	CLKB	PDM	HVPS	LENSES
		Hex	**0x11**	**0x22**	**0x33**	**0x44**	**0x55**	**0x66**	**0x77**	**0x88**	**0x99**	**0xAA**	**0xBB**
Voltage	**0x61**	V1	-	V 5 CCB	V 12 CPU	V 12 HK-O	V 5 PDM	V 3.3	-	Vc Aux	?	DAC out 1	-
	0x62	V2	-	V 5 CLKB	V 5 DST	V 3.3 HK-O	V 5 EC	V 2.5	-	Vc Int	?	DAC out 2	-
	0x63	V3	-	V 5 GPS	-	-	-	V 1.2	-	-	?	DAC out 3	-
	0x64	V4	-	-	-	-	-	V 5	-	-	?	DAC out 4	-
	0x65	V5	-	-	-	-	-	V 1.8	-	-	?	DAC out 5	-
	0x66	V6	-	-	-	-	-	-	-	-	?	DAC out 6	-
	0x67	V7	-	-	-	-	-	-	-	-	?	DAC out 7	-
	0x68	V8	-	-	-	-	-	-	-	-	?	DAC out 8	-
	0x69	V9	-	-	-	-	-	-	-	-	?	DAC out 9	-
Current	**0x71**	I1	-	I 5 CCB	I 12 CPU	I 12 HK-O	I 5 PDM	I 3.3	-	-	?	-	-
	0x72	I2	-	I 5 CLKB	I 5 DST	I 3.3 HK-O	I 5 EC	I 2.5	-	-	?	-	-
	0x73	I3	-	I 5 GPS	-	-	-	I 1.2	-	-	?	-	-
	0x74	I4	-	-	-	-	-	-	-	-	?	-	-
	0x75	I5	-	-	-	-	-	-	-	-	?	-	-
Temperature	**0x81**	T1	Thermistor 1	-	-	-	-	T max	Temp 1	Temp 1	?	-	Thermistor 1
	0x82	T2	Thermistor 2	-	-	-	-	T fpga	ADC	-	?	-	Thermistor 2
	0x83	T3	Thermistor 3	-	-	-	-	-	-	-	?	-	Thermistor 3
	0x84	T4	Thermistor 4	-	-	-	-	-	-	-	?	-	Thermistor 4
	0x85	T5	Thermistor 5	-	-	-	-	-	-	-	?	-	Thermistor 5
	0x86	T6	IC SPI	-	-	-	-	-	-	-	?	-	Thermistor 6
	0x87	T7	-	-	-	-	-	-	-	-	?	-	Thermistor 7
	0x88	T8	-	-	-	-	-	-	-	-	?	-	Thermistor 8
	0x89	T9	-	-	-	-	-	-	-	-	?	-	Thermistor 9
Pressure	**0x91**	P1	Pressure sensor	-	-	-	-	-	-	-	?	-	-
Accelerometer	**0x99**	A1	Accelerometer	-	-	-	-	-	-	-	?	-	-
		59	8	6	4	4	4	10	2	3		9	9

Table 8

	Parameter / Hex		HK-N	LVPS AN	LVPS HK-N	BATTERY	
			0xCC	**0xDD**	**0xEE**	**0xFF**	**0x1A**
Voltage	**0x61**	V1	-	V 5 Slot_1	V 5 HK-N	V Battery	-
	0x62	V2	-	V 5 Slot_2	V 3.3 HK-N	-	-
	0x63	V3	-	V 5 Slot_3	V 1.0 HK-N	-	-
	0x64	V4	-	V 5 Slot_4	-	-	-
	0x65	V5	-	V 5 Slot_5	-	-	-
	0x66	V6	-	V 5 Slot_6	-	-	-
	0x67	V7	-	-	-	-	-
	0x68	V8	-	-	-	-	-
	0x69	V9	-	-	-	-	-
Current	**0x71**	I1	-	V 5 Slot_1	I 5 HK-N	I Battery 1	-
	0x72	I2	-	V 5 Slot_2	I 3.3 HK-N	I Battry 2	-
	0x73	I3	-	V 5 Slot_3	I 1.0 HK-N	-	-
	0x74	I4	-	V 5 Slot_4	-	-	-
	0x75	I5	-	V 5 Slot_5	-	-	-
	0x76	I6	-	V 5 Slot_6	-	-	-
Temperature	**0x81**	T1	Thermistor 1	-	-	Temp 1	Thermistor 1
	0x82	T2	Thermistor 2	-	-	Temp 2	Thermistor 2
	0x83	T3	Thermistor 3	-	-	-	Thermistor 3
	0x84	T4	-	-	-	-	Thermistor 4
	0x85	T5	-	-	-	-	Thermistor 5
	0x86	T6	-	-	-	-	Thermistor 6
	0x87	T7	-	-	-	-	Thermistor 7
	0x88	T8	-	-	-	-	Thermistor 8
	0x89	T9	-	-	-	-	Thermistor 9
		35	3	12	6	5	9

At this point HK-O and HK-N will send back to CPU the information requested, as shown in the next table.

Header	**Command**	Num	**DATA H**	**DATA L**	**End**	**CRC16H**	**CRC16L**
0x32	0x11	0x81	0x01	0x00	0x17	0x56	0xAA

Table 9 Example: HK-O will respond with the next frame containing the Data requested, in this example 0x0100

The header will be the same indicating it is a response for header of lecture

Configure high and low value range operation for a parameter measured by HK-O

When the header value is 0x33 hexadecimal, it means a configuration function for a limit operation range. The bytes structure of this function is shown below. Higher Value Byte is always 0x48 hexadecimal and Lower Byte is 0x4c Hexadecimal, for the 4th byte on the frame.

Header	Command	Parameter	Configure High Value	DATA H	DATA L	End	CRC16H	CRC16L
0x33	0x11	0x81	0x48	0x03	0xFF	0x17	0x60	0xDF

Table 10.- Example: Set the higher value for the threshold on Thermistor 1 alarm on the HK board: 0x03FF

Header	Command	Parameter	Configure High Value	DATA H	DATA L	End	CRC16H	CRC16L
0x33	0x11	0x81	0x4C	0x00	0x80	0x17	0x28	0xCB

Table 11.- Example: Set the lower value for the threshold on Thermistor 1 alarm on the HK board: 0x0080

Table 12

	Low Value		High Value	HK-O	LVPS1 DP	LVPS2 DP	LVPS HK-O	LVPS PDM	CCB	GPS	CLKB	PDM	HVPS	LENSES
				0x11	0x22	0x33	0x44	0x55	0x66	0x77	0x88	0x99	0xAA	0xBB
Voltage	0x61		0x43	-	V 5 CCB	V 12 CPU	V 12 HK-O	V 5 PDM	V 3.3	-	Vc Aux	?	DAC out 1	-
				-	V 5 CLKB	V 5 DST	V 3.3 HK-O	V 5 EC	V 2.5	-	Vc Int	?	DAC out 2	-
				-	V 5 GPS	-	-	-	V 1.2	-	-	?	DAC out 3	-
				-	-	-	-	-	V 5	-	-	?	DAC out 4	-
				-	-	-	-	-	V 1.8	-	-	?	DAC out 5	-
				-	-	-	-	-	-	-	-	?	DAC out 6	-
				-	-	-	-	-	-	-	-	?	DAC out 7	-
				-	-	-	-	-	-	-	-	?	DAC out 8	-
				-	-	-	-	-	-	-	-	?	DAC out 9	-
Current				-	I 5 CCB	I 12 CPU	I 12 HK-O	I 5 PDM	I 3.3	-	-	?	-	-
				-	I 5 CLKB	I 5 DST	I 3.3 HK-O	I 5 EC	I 2.5	-	-	?	-	-
				-	I 5 GPS	-	-	-	I 1.2	-	-	?	-	-
				-	-	-	-	-	-	-	-	?	-	-
Temperature				Thermistor 1	-	-	-	-	T max	Temp 1	Temp 1	?	-	Thermistor 1
				Thermistor 2	-	-	-	-	T fpga	ADC	-	?	-	Thermistor 2
				Thermistor 3	-	-	-	-	-	-	-	?	-	Thermistor 3
				Thermistor 4	-	-	-	-	-	-	-	?	-	Thermistor 4
				Thermistor 5	-	-	-	-	-	-	-	?	-	Thermistor 5
				IC SPI	-	-	-	-	-	-	-	?	-	Thermistor 6
				-	-	-	-	-	-	-	-	?	-	Thermistor 7
				-	-	-	-	-	-	-	-	?	-	Thermistor 8
				-	-	-	-	-	-	-	-	?	-	Thermistor 9
Pressure				Pressure sensor	-	-	-	-	-	-	-	?	-	-
Accelerometer				Accelerometer	-	-	-	-	-	-	-	?	-	-
				8	6	4	4	4	10	2	3		9	9

Sending low priority alarms vector

When the header value is 0x34 hexadecimal, CPU must identify this vector as a frame containing low priority alarms data. This indicates that the alarm generated, can be cleared itself. As for example a temperature value can change and return to the nominal operation value then alarm will be cleared.

Depending on the alarm type, CPU could request an action to be executed by HK. This has to be discussed further.

Header	Command	High Generated Alarm 2	High Generated Alarm 1	Low Generated Alarm 2	Low Generated Alarm 1	End	CRC16H	CRC16L
0x34	0x11	000	0x00	0x00	0x03	0x17	0x60	0xDF

Table 13.- Example: When the acquired data value on thermistor 1, by the HK board, is lower than the value previously configured on the lower threshold then an alarm is registered and send to CPU.

Header	Command	High Generated Alarm 2	High Generated Alarm 1	Low Generated Alarm 2	Low Generated Alarm 1	End	CRC16H	CRC16L
0x34	0x11	000	0x01	0x00	0x00	0x17	0x60	0xDF

Table 14.- Example: When the acquired data value on thermistor 1, by the HK board, is higher than the value previously configured on the higher threshold then an alarm is registered and send to CPU.

Bit assignation for each subsystem, at the moment considered

HK-O 0x11

Higher Threshold															
Byte 2								Byte 1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
										Tdigital	T5	T4	T3	T2	T1
Lower Threshold															
Byte 2								Byte 1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
										Tdigital	T5	T4	T3	T2	T1

Table 15

LVPS Voltages 0x22

Higher Threshold															
Byte 2								Byte 1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
SiPM	Health Led 1	Health Led 2	IRc	HK-N	Compass			PDM	HK_33	HK_12	DST	CPU	GPS	CLKB	CCB
Lower Threshold															
Byte 2								Byte 1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
SiPM	Health Led 1	Health Led 2	IRc	HK-N	Compass			PDM	HK_33	HK_12	DST	CPU	GPS	CLKB	CCB

Table 16

LVPS Currents 0x33

Higher Threshold															
Byte 2								Byte 1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
SiPM	Health Led 1	Health Led 2	IRc	HK-N	Compass			PDM	HK_33	HK_12	DST	CPU	GPS	CLKB	CCB
Lower Threshold															
Byte 2								Byte 1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
SiPM	Health Led 1	Health Led 2	IRc	HK-N	Compass			PDM	HK_33	HK_12	DST	CPU	GPS	CLKB	CCB

Table 17

CCB 0x66

Higher Threshold															
Byte 2								Byte 1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
						Tfpga	Tmax	I1.2	I2.5	I3.3	V1.8	V5	V1.1	V2.5	V3.3
Lower Threshold															
Byte 2								Byte 1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
						Tfpga	Tmax	I1.2	I2.5	I3.3	V1.8	V5	V1.1	V2.5	V3.3

Table 18

GPS 0x77

Higher Threshold															
Byte 2								Byte 1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
														ADC	Temp1
Lower Threshold															
Byte 2								Byte 1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
														ADC	Temp1

Table 19

CLKB 0x88

Higher Threshold															
Byte 2								Byte 1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
													Temp1	VcInt	VcAux
Lower Threshold															
Byte 2								Byte 1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
													Temp1	VcInt	VcAux

Table 20

CLKB 0x88

Higher Threshold															
Byte 2								Byte 1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
													Temp1	VcInt	VcAux
Lower Threshold															
Byte 2								Byte 1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
													Temp1	VcInt	VcAux

Table 20

HVPS 0xAA

Higher Threshold															
Byte 2								Byte 1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
							STA9	STA8	STA7	STA6	STA5	STA4	STA3	STA2	STA1
Lower Threshold															
Byte 2								Byte 1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
							STA9	STA8	STA7	STA6	STA5	STA4	STA3	STA2	STA1

Table 21

LENSES 0xBB

Higher Threshold															
Byte 2								Byte 1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
							T9	T8	T7	T6	T5	T4	T3	T2	T1
Lower Threshold															
Byte 2								Byte 1							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
							T9	T8	T7	T6	T5	T4	T3	T2	T1

Table 21

Only the one in red are now implemented, but the other will be sent when the interaction with the system is performed, should be only considered now.

HK will send high priority alarms data

When the header value is 0x35 hexadecimal, CPU must identify this vector as a frame containing a high priority alarm. Those are related to problems that may need CPU to take decision about what to do next. The structure of the frame containing all the bytes is the following:

Header	Command	End	CRC16H	CRC16L
0x36	0x4A	0x17	0xB9	0x9C

Table 22.- Example: HK Send to CPU a High Priority alarm about DONE signal not received from CCB.

These are the first High Priority alarms implemented as soon as they are more cases detected, updates will be sent.

Hexadecimal	Problem	System
0x41	No SPI communication	CCB
0x42	No SPI communication	CLKB
0x44	Alarm CLK-Stopped 200MHz	CCB
0x45	Alarm CLK-Stopped 100MHz	CCB
0x46	Alarm CLK-Stopped 200MHz	CLKB
0x47	Alarm CLK-Stopped 100MHz	CLKB
0x48	Alarm GPS Off	CLKB
0x49	Alarms PPS off	CLKB
0x4A	Alarm DONE not Recived on Time	CCB
0x4B	Alarm DONE not Recived on Time	CLKB

Hexadecimal	Problem	System
0x51	Wrong Contact Closure detected	CPU
0x52	Wrong Contact Closure detected	DST
0x53	Wrong Contact Closure detected	GPS
0x54	Wrong Contact Closure detected	CCB
0x55	Wrong Contact Closure detected	CLKB
-	Wrong Contact Closure detected	EC
0x57	Wrong Contact Closure detected	PDM
0x58	Wrong Contact Closure detected	SiPM
0x59	Wrong Contact Closure detected	Health LED 1
0x5A	Wrong Contact Closure detected	Health LED 1
0x5B	Wrong Contact Closure detected	Health LED 2
0x5C	Wrong Contact Closure detected	IR-C
0x5D	Wrong Contact Closure detected	Compass
0x5E	Wrong Contact Closure detected	HAM Xmitter
0x5F	Wrong Contact Closure detected	Cooler
0x50	Wrong Contact Closure detected	Photodiode
0x56	Wrong Contact Closure detected	HVPS

Optimization of the Commands

When the header value is 0x36 hexadecimal, CPU is asking for the reading parameters to HK, and a new frame containing all parameters per system was created, this will be more efficient for CPU when asking for the values, instead of just to ask one value, all the parameters are going to be sent.

According to the maximum parameters like CCB, that are 10 values from 2 bytes each one, this frame is created with the maximum parameters possible, so the total packets will be 25 bytes placed in the next order and this apply to all the systems.

Header	Adress	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	End	CRC16H	CRC16L

Each system will have different values of parameters staring from P1, when a system doesn't have many parameters as CCB, then the rest of the values are filled with **0x3030**, so on the following table you can find the number of parameters per system.

System	Adress	Header for the response	Bytes with Data	Total Packet byte size
Temp HK / Cpu	0x11	0x36	12	25
LVPS1-DP	0x22	0x36	12	25
LVPS2-DP	0x33	0x36	8	25
LVPS-HK	0x44	0x36	8	25
LVPS-PDM	0x55	0x36	8	25
CCB	0x66	0x36	20	25
CLKB	0x77	0x36	4	25
GPS	0x88	0x36	6	25
PDM	0x99	0x36	20	25
LENSES	0xBB	0x36	18	25

Example: **Total packet for the reading of GPS system, on table 23 we can see that the address for GPS is 0x88 and also that it only contains 6 bytes of data, in this case filled with 0x1111 each parameter, the other not used bytes are filled with 0x3030 for parameter, finally the CRC for the whole frame is 0x80c8.**

0x36	0x88	0x1111	0x1111	0x11	0x3030	0x3030	0x3030	0x3030	0x3030	0x3030	0x3030	0x17	0x80	0xC8

Contact Closure command

When the header value is 0x37 hexadecimal, HK-O is sending the status for the Contact Closure, the status will be masked into one byte containing the next bit structure:

Data from HK-O

MSB	Contact Clousure Status Byte						LSB
0	CPU	DST	CLKB	CCB	PDM	EC	GPS

When the header value is 0x37 hexadecimal, HK-N is sending the status for the Contact Closure, the status will be masked into two bytes containing the next bit structure:

Data from HK-N

MSB	Contact Clousure Status Byte														LSB
						HVPS	Cooler	HAM-X	AN6	IR	Photo Diode	Compass	HL2	HL1	SiPM

In this case a low value of 0 means not closed (OFF) and high value of 1 is closed (ON).

Example: **HK Sending the byte for CC status, the systems that are ON are: CPU, DST, CLKB, CCB and GPS.**

Header	CC Byte	End	CRC16H	CRC16L
0x37	0x79	0x17	0xBA	0x16

HVPS Commands not implemented Yet