SLS2 Sensor Communications

COMMANDS SUMMARY

Code	Description
0	- Turn on Part-In-View PIV LED (not used anymore)
1	- Read sensor Temperature (8 bit)
2	- Request sensor to resend last message
3	- Request sensor to transmit its linearization data
4	- Request sensor to transmit its status byte
5	- Request sensor to perform a Laser/Video control operation
6	- Send (burn) linearization data to the sensor
7	- Perform a PCB rapid assembly test (not used anymore)
8	- Read sensor software Version (NEW)
9	- Read sensor Temperature (16 bit)
10	- Set Baudrate
11	- Soft Reboot
12	– Request ISP Mode (for loading a new program into sensor)
13	- Erase Flash Data

Revision History

Date	Who	Description of Change
14-May-98	Marc Noel	Added 'Camera gain' field to the 'ACTION_BYTE' for a '5' [LASER/VIDEO CONTROL] command.
28-May-98	Don Clarke	Added command 8 - 'Read sensor software Version'
1-Feb-00	A. Gray	Added command 9 - return a 16 bit temperature number
15-Feb-00	D. Clarke	Modified command 9 slightly - return a 16 bit temperature number
26-Jun-00	A. Gray	- Re-titled this doc to sat "SLS2" - added commands 11 reset and 13 Erase Flash - Corrected typing errors in 'Burn Linearization Data', CMD 6
8-Sep-00	A.Gray	Corrected comments about conversion of temperature counts to degrees C
14-Sep-00	D. Clarke	Added 'Aux light' bit 4 and 'NoAutoOff' (bit 5) to the 'ACTION_BYTE' for a '5' [LASER/VIDEO CONTROL] command
22-Sep-00	D. Clarke	Redefined use of camera gain control bit (bit 3) in LASER/VIDEO CONTROL command (CMD 5) in Version 3 Sensors. Updated 'NOTE FOR SMARTLITE' in 'Burn Linearization Data', CMD. Documented commands 10 – Set Baudrate and 12 – Request ISP mode.
16-Mar-01	D. Clarke	Fixed Documentation of command 10 – Set Baudrate
04-Dec-01	D. Clarke	Updated 'NOTE FOR SMARTLITE' in 'Burn Linearization Data', CMD.
01-Jan-03	D. Clarke	Fixed documentation only. Hi / lo bytes reversed in both SMARTLIGHT ASSOCIATED SENSOR S/N and SMARTLITE's INTEGRATION VALUE LO/HI
0s-Sept-03	A. Gray	Corrected STX and ETX ascii codes

COMMUNICATION PROTOCOL

This section defines the protocol for messages that are sent between the K16 and the SLS sensors on its sensor chain.. The protocol requires that there be only one "TALKER" (the K16) and any number of "LISTENERS" (sensors). "LISTENERS" may only talk when directed to do so by the "TALKER".

K16 USART configuration:

Asynchronous
Full Duplex
9600 Baud
1 Stop Bit
8 Data Bits
Odd Parity

A message from the K16 to a sensor consists of the following sequence:

DLE	0x10
STX	0x02
Sensor ID (low byte)	
Sensor ID (high byte)	
:	
DATA	
:	
CHECKSUM	
DLE	0x10
ETX	0x03

A response message from a sensor can take two forms:

A) If data is to be returned to the K16, the sensor responds with the following sequence:

DLE
STX
Sensor ID (low byte)
Sensor ID (high byte)
:
DATA
:
CHECKSUM
DLE
ETX

B) If no data is to be returned to the K16, the sensor responds with the following sequence:

DLE
ETX

Example 1:
A K16 sends a "Turn on PIV command" to sensor S/N 1405 (0x057D) by sending the following:

DLE		0x10	
STX		0x02	
S_IDlo	0x7D		| used to compute checksum
S_IDhi	0x05		|
PIV command	0x00		|
Checksum		0x7D	CHK = ~(7D+05+00) == 7D
DLE		0x10	
ETX		0x03	

Sensor S/N 1405 (0x057D) would respond as follows and turn on its PIV LED:

DLE	0x10
ETX	0x03

Example 2:
A K16 sends a "Laser/Video Control" command to sensor S/N 1997 (0x07CD). We want to turn the laser line on for an integration time of 65%, and want (obviously) to have the sensors' video connected to the video bus lines. We accomplish this by sending:

DLE		0x10	
STX		0x02	
S_IDlo	0xCD		| used to compute checksum
S_IDhi	0x07		|
LASER/VIDEO cmd		0x05	|
action		0x07	| 00000111b == 0x07
integ_hi		0x19	| 65% integ: 6500 dec, 0x1964
integ_lo		0x64	|
Checksum		0xA2	CHK = ~(CD+07+05+07+19+64) == A2
DLE		0x10	
ETX		0x03	

Sensor S/N 1997 (0x07CD) would respond as follows and perform the LASER/VIDEO operation:

DLE	0x10
ETX	0x03

Example 3:
A K16 sends a "Request Temperature" command to sensor S/N 3031 (0x0BD7). We accomplish this by sending:

DLE		0x10	
STX		0x02	
S_IDlo	0xD7		| used to compute checksum
S_IDhi	0x0B		|
READTEMP cmd		0x01	|
Checksum		0x1C	CHK = ~(D7+0B+01) == 1C
DLE		0x10	
ETX		0x03	

Sensor S/N 3031 (0x0BD7) would respond as follows:

DLE		0x10	
STX		0x02	
S_IDlo	0xD7		| used to compute checksum
S_IDhi	0x0B		|
Temp value		0x84	| assume the temp count is 132, or 0x84
Checksum		0x99	CHK = ~(D7+0B+84) == 99
DLE		0x10	
ETX		0x03	

DEL Sequences

In messages sent to the sensors, the character DLE is used by the K16 to inform the sensor that the next character to be received is a control character rather than a part of the message.

All messages to the sensor start with DLE,STX (0x10,0x02). Whenever this sequence is received the sensor prepares to receive a new message. Any characters received prior to receiving DLE,STX are ignored by the sensor.

Messages to the sensor are terminated with DLE,ETX (0x10,0x03). All the characters received between DLE,STX and DLE,ETX are considered to be part of a single message.

If DLE occurs anywhere else in the message, the K16 will send the sequence DLE,DLE in place of the single DLE. The sensor will discard one of them and treat the other as a data character.

CHECKSUM Computation

Checksums are performed at the byte level, by summing the byte values of the message and taking the two's complement of that sum. Refer to the examples above for clarification.

"0": TURN ON PIV LED (used for Zsensors, not SLS)

This command is used to turn on the "Part In View" LED of a sensor for 1.0 sec.

K16 sends:

DLE	0x10	1 byte	
STX	0x02	1 byte	
Sensor IDlo	0x??	1 byte	for CHKSUM
Sensor Idhi	0x??	1 byte	|
TURN ON PIV command	0x00	1 byte	|
Checksum	0x??	1 byte	
DLE	0x10	1 byte	
ETX	0x03	1 byte	

Sensor returns:

DLE	0x10	1 byte	
ETX	0x03	1 byte	

"1": REQUEST SENSOR TEMPERATURE (8 BIT)

For versions 1 and 2 of the sensor, this command is used to get a sensor's internal temperature as a count from 0-255. A count of 128 = 25 degC approximately and each count represents 0.390625°C (there are 2.56 counts/°C).

For Version 3 sensors there is a 16 bit version of this call. See **REQUEST 16 BIT SENSOR TEMPERATURE.** When this 8 bit call is used on a version 3 sensor, each count is still 0.390625 degC but a count of 0 is equal to 0 degC.

K16 sends:

DLE	0x10	1 byte	
STX	0x02	1 byte	
Sensor IDlo	0x??	1 byte	for CHKSUM
Sensor Idhi	0x??	1 byte	|
READ TEMP command	0x01	1 byte	|
Checksum	0x??	1 byte	
DLE	0x10	1 byte	
ETX	0x03	1 byte	

Sensor returns:

DLE	0x10	1 byte	
STX	0x02	1 byte	
Sensor IDlo	0x??	1 byte	for CHKSUM
Sensor Idhi	0x??	1 byte	|
Temp value	0x??	1 byte	|
Checksum	0x??	1 byte	
DLE	0x10	1 byte	
ETX	0x03	1 byte	

"2": REQUEST SENSOR RESEND LAST MESSAGE

This command is used to get a sensor to retransmit its last message. The last message sent to the host will be resent providing the previous K16 command did not request "SEND LINEARIZATION DATA". The linearization data is the only message that is not buffered by the sensor.

K16 sends:

DLE	0x10	1 byte	
STX	0x02	1 byte	
Sensor IDlo	0x??	1 byte	for CHKSUM
Sensor Idhi	0x??	1 byte	|
RESEND command	0x03	1 byte	|
Checksum	0x??	1 byte	
DLE	0x10	1 byte	
ETX	0x03	1 byte	

Sensor returns:

previous response data

"3": REQUEST SENSOR SEND ITS LINEARIZATION DATA

This command is used to get a sensor to send portions of its linearization data.
OFFSET refers to the number of bytes into the sensor's EEPROM from which data will be returned. LENGTH refers to the number of data bytes to be returned.

OFFSET can be 0x0000 - 0x1FFF
LENGTH can be 0x0001 - 0x01F4 (1-500 decimal)
NOTE: you **must** abide by (OFFSET+LENGTH) <= 0x1FFF

K16 sends:

DLE	0x10	1 byte	
STX	0x02	1 byte	
Sensor IDlo	0x??	1 byte	for CHKSUM
Sensor Idhi	0x??	1 byte	
SEND LIN DATA command	0x03	1 byte	
OFFSET HI	0x??	1 byte	
OFFSET LO	0x??	1 byte	
LENGTH HI	0x??	1 byte	
LENGTH LO	0x??	1 byte	
Checksum	0x??	1 byte	
DLE	0x10	1 byte	
ETX	0x03	1 byte	

Sensor returns:

DLE	0x10	1 byte	
STX	0x02	1 byte	
Sensor IDlo	0x??	1 byte	CHKSUM
Sensor Idhi	0x??	1 byte	
:	:	:	
lin data	0x??	LENGTHbytes	
:	:	:	
Checksum	0x??	1 byte	
DLE	0x10	1 byte	
ETX	0x03	1 byte	

"4": REQUEST SENSOR STATUS BYTE

This command is used to get a sensor's status. The sensor will return 2 status bytes.
The first byte is the HEALTH byte, the second byte is the number of times a checksum error was detected while receiving messages from the K16.

HEALTH BYTE:

Bit Position	AND with	Meaning if set to '1'
7	0x80	n/a, always '0'
6	0x40	n/a, always '0'
5	0x20	n/a, always '0'
4	0x10	LASER is burnt out (not indicated any more, so ignore)
3	0x08	TEMP sensing chip failure
2	0x04	RAM failure
1	0x03	CHKSUM error in the LINEARIZATION EEPROM
0	0x01	CHKSUM error detected in the sensor's CODE PROM

After each status request, the HEALTH byte is ANDed with 0x07 so that power-up error conditions are preserved. Note the the CHKSUM and RAM failures are only detected at power up and when reset using command 11.

NUMBER OF CHKSUM ERRORS BYTE:

This indicates the number of times a checksum error was detected when receiving a message from the host. This byte is cleared after each status request to the sensor.

K16 sends:

DLE	0x10	1 byte	
STX	0x02	1 byte	
Sensor IDlo	0x??	1 byte	for CHKSUM
Sensor Idhi	0x??	1 byte	
GET STATUS command	0x04	1 byte	
Checksum	0x??	1 byte	
DLE	0x10	1 byte	
ETX	0x03	1 byte	

Sensor returns:

DLE	0x10	1 byte	
STX	0x02	1 byte	
Sensor IDlo	0x??	1 byte	for CHKSUM
Sensor Idhi	0x??	1 byte	
HEALTH BYTE	0x??	1 byte	
# CHKSUM ERRORS	0x??	1 byte	
Checksum	0x??	1 byte	
DLE	0x10	1 byte	
ETX	0x03	1 byte	

"5": LASER/VIDEO CONTROL

This command is used to control the sensor's laser, area illuminators and video bus relay.

K16 sends:

DLE		0x10	1 byte	
STX		0x02	1 byte	
Sensor IDlo		0x??	1 byte	for CHKSUM
Sensor Idhi		0x??	1 byte	|
LASER/VIDEO CONTROL command		0x05	1 byte	|
ACTION_BYTE		0x??	1 byte	|
INTEG_HI		0x??	1 byte	|
INTEG_LO	0x??		1 byte	|
Checksum		0x??	1 byte	
DLE		0x10	1 byte	
ETX		0x03	1 byte	

Sensor returns:

DLE	0x10	1 byte
ETX	0x03	1 byte

ACTION_BYTE:

Controls	**Bit Pos'n**	**Meaning if '0'**	**Meaning if '1'**
n/a	7		
n/a	6		
NoAutoOff	5	see note1	
Aux light	4	off	on
Camera gain	3		
Sensor V1&2		High gain	Low gain
Sensor V3		No meaning	See note 2
Put video relay on	2	no	yes
Internal Light Source	1	no	yes
Light Source	0	use AREA LEDs	use LASER line

NOTES:

1: When bit 4 is low all lights are automatically turned off on any communications to any sensor. If high the lights remain until this command is received again by this sensor.

2: Bit 3 meaning for Version 3 sensors
LO - standard response to other control bits in cmd 5
HI - request of area light is ignored
- For V3 sensors only, so the operator could still request that the internal light not turn on even though a V3 smartlite partnered with this sensor was responding to the area light request, without making any changes to the K16 code

Ex. 1: Turn video relay on, and use the laser line, high gain:action = 0x07
Ex. 2: Turn video relay and all light sources off, high gain: action = 0x00
Ex. 3: Turn video relay on, no light source on, high gain: action = 0x04 or 0x05

Ex. 4: Turn video relay off, and area LED's on, high gain: action = 0x03
Ex. 5: Turn video relay on, and use the laser line, low gain: action = 0x0F

INTEG_HI & INTEG_LO:

Version 3 Sensors:

The laser line or the area lights are NOT controlled by an 'integration time'. They are only turned on or off as requested. The 'integration time' is now sent to the sensors camera and really does control it's exposure time. Values between 0 and 512 are divided by 2, then sent to the camera. It's response is non-linear and approximately equal to 10 raised to the power of (0.12876 j – 5.009965).

Version 1&2 Sensors:

The laser line and the area LED's can be controlled by an 'integration time'. This specifies the length of time to keep a light source ON.

Two methods of light integration are employed, depending on the 2-byte WORD value for 'integration_time'.

1) For integration times from 0-9 counts, the sensor will use an interrupt procedure to integrate light for 0-9 horizontal sync lines per field.
2) For integration times from 10-499 counts, a polling procedure is used which enables the light source for a fraction of each horizontal sync line. This method offers 51 different integration times. For integration time greater than 499, the light source is on continually.

An integration value from the K16 level is usually expressed as a percentage, like 60.0%.

The internal representation of this value is actually a 2-byte WORD which is the (percentage value * 100). This value, as it is sent to the sensor is divided by 20.

The following table should help clarify how these bytes are determined:

K16 Integration Value (%)	**K16 Integration Value (Dec)**	**Integ Value Sent To Sensor (Dec)**	**Integ Value Sent To Sensor (Hex)**	**INTEG_HI**	**INTEG_LO**
0.00	0	0	0x00000x00	0x00	
0.20	20	1	0x00010x00	0x01	
0.40	40	2	0x00030x00	0x03	
0.60	60	3	0x00030x00	0x03	
0.80	80	4	0x00040x00	0x04	
1.00	100	5	0x00050x00	0x05	
1.20	120	6	0x00060x00	0x06	
1.40	140	7	0x00070x00	0x07	
1.60	160	8	0x00080x00	0x08	
1.80	180	9	0x00090x00	0x09	
2.00	200	10	0x000A	0x00	0x0A
2.20	220	11	0x000B	0x00	0x0B
2.40	240	12	0x000C	0x00	0x0C
:	:	:	:	:	:
99.60	9960	498	0x01F2	0x01	0xF2
99.80	9980	499	0x01F3	0x01	0xF3

100.00	10000	500	0x01F4	0x01	0xF4

*

(Frame grabber output LUT is scaled by 2)					
100.20	10030	250	0x00FA	0x00	0xFA
100.40	10040	251	0x00FB	0x00	0xFB
:	:	:	:	:	:
199.80	19980	499	0x01F3	0x01	0xF3
200.00	20000	500	0x01F4	0x01	0xF4
(Frame grabber output LUT is scaled by 3)					
200.20	20030	333	0x014D	0x01	0x4D
:	:	:	:	:	:
299.80	29980	499	0x01F3	0x01	0xF3
300.00	30000	500	0x01F4	0x01	0xF4
(Frame grabber output LUT is scaled by 4)					
300.20	30030	375	0x01770x01	0x77	
:	:	:	:	:	:
399.80	39980	499	0x01F3	0x01	0xF3
400.00	40000	500	0x01F4	0x01	0xF4
(Frame grabber output LUT is scaled by 5)					
400.20	40030	400	0x01900x01	0x90	
:	:	:	:	:	:
499.80	49980	499	0x01F3	0x01	0xF3
500.00	50000	500	0x01F4	0x01	0xF4
(Frame grabber output LUT is scaled by 6)					
500.20	50030	416	0x01A0	0x01	0xA0
:	:	:	:	:	:
599.80	59980	499	0x01F3	0x01	0xF3
600.00	60000	500	0x01F4	0x01	0xF4
(Frame grabber output LUT is scaled by 7)					
600.20	60030	428	0x01AC	0x01	0xAC
:	:	:	:	:	:
655.20	65520	468	0x01D4	0x01	0xD4

*To see how the integration value mechanism works for greater than 100.00%, refer to acquire_an_image() in ACQ.C.

"6": BURN LINEARIZATION DATA

This command is used to program a sensor's linearization table.
A maximum of ten bytes can be burned at a time. The address limitations are the same as for the READ LINEARIZATION DATA command.
OFFSET refers to the start address on the EEPROM where burn-in will start.
LENGTH refers to the number of consecutive data bytes to be burned.
SUCCESS_CODE is 0xFF if the burn succeeded, and is 0x00 if the burn failed.

OFFSET can be 0x0000 - 0x1FFF
LENGTH can be 0x01 - 0x0A (1-10 decimal)
NOTE: you **must** abide by (OFFSET+LENGTH) <= 0x1FFF

K16 sends:

DLE	0x10	1 byte	
STX	0x02	1 byte	
Sensor IDlo	0x??	1 byte	for CHKSUM
Sensor IDhi	0x??	1 byte	|
BURN LIN DATA command	0x06	1 byte	|
OFFSET HI	0x??	1 byte	|
OFFSET LO	0x??	1 byte	|
LENGTH	0x??	1 byte	|
:	:	:	|
data	0x??	1 byte	|
:	:	:	|
Checksum	0x??	1 byte	
DLE	0x10	1 byte	
ETX	0x03	1 byte	

Sensor returns:

DLE	0x10	1 byte	
STX	0x02	1 byte	
Sensor IDlo	0x??	1 byte	for CHKSUM
Sensor Idhi	0x??	1 byte	|
SUCCESS_CODE	0xFF/ 0x00	1 byte	|
Checksum	0x??	1 byte	
DLE	0x10	1 byte	
ETX	0x03	1 byte	

SPECIAL NOTE FOR SMARTLITE CONFIGURATION: (SLS only)

Version 3 Sensors:

Similar operation as described for Version 1&2 sensors below, except:
To change smartlite partners requires that the Smartlites flash memory be erased. This is accomplished by using the K16 to burn partner#1 = 0 (ignore any errors). When the smartlite is rebooted, it's internal flash will be erased and it will then become ready for a burn of new partner(s) as described below. Note that Smartlite V3 no longer uses the integration values supplied, but always turns on full. In Version 1&2 sensors, it was in fact the sensor internal area light intensity was controlled by the sensor's 'integration time'. Now the camera really does change its exposure time, and the sensor's internal light may be kept off by using the modified 'camera gain ' bit as shown in CMD 5 - LASER/VIDEO CONTROL.

Version 1&2 Sensors:

The SMARTLITE is a light source which responds to the AREA LED instructions given to an "ASSOCIATED SENSOR". In essence, the SMARTLITE sits on the serial communication lines and listens for instructions that have the "ASSOCIATED SENSOR" as their destination. Then, if the instruction is to control the AREA LED's, the SMARTLITE responds in addition to the associated sensor.

The SMARTLITE requires the use of the "BURN LINEARIZATION DATA" (6) command in order to be configured, because it stores the "ASSOCIATED SENSOR ID" and the SMARTLITE's "INTEGRATION VALUE" as two unsigned int's (4 bytes) starting from offset byte 16 (0x10) in the SMARTLITE's linearization EEPROM.

So, in the SMARLITE's EEPROM:

EEPROM Offset

HEX	DEC	Meaning
0x01	1	SMARTLITE's own S/N LO
0x03	2	SMARTLITE's own S/N HI
:	:	
0x10	16	ASSOCIATED SENSOR S/N LO
0x11	17	ASSOCIATED SENSOR S/N HI
0x12	18	SMARTLITE's INTEGRATION VALUE LO
0x14	19	SMARTLITE's INTEGRATION VALUE HI

And an example of the K16 message to configure a SMARTLITE (S/N 4111 or 0x100F) to associate with sensor S/N 3192 (or 0x0C78) and have an integration value of 80% .
An integration value of 80% is 8000 as a UINT. Which means we should send 8000/20 = 400 to the Smartlite. Finally, 400 decimal is 0x0190.

K16 sends:

DLE		0x10	1 byte	
STX		0x02	1 byte	
Sensor IDlo (Smartlite)		0x0F	1 byte	for CHKSUM
Sensor IDhi (Smartlite)		0x10	1 byte	
BURN LIN DATA command	0x03	1 byte		
OFFSET HI		0x00	1 byte	
OFFSET LO		0x10	1 byte	
LENGTH		0x04	1 byte	
ASS. SENSOR S/N LO		0x78	1 byte	

ASS. SENSOR S/N HI		0x0C	1 byte	|
INTEG_LO	0x90		1 byte	|
INTEG_HI		0x01	1 byte	|
Checksum ***		0xB4	1 byte	
DLE		0x10	1 byte	
ETX		0x03	1 byte	

***Checksum = ~(0F+10+03+00+10+04+78+0C+90+01) = 0xB4

Smartlite returns:

DLE		0x10	1 byte	
STX		0x02	1 byte	
Sensor IDlo		0x0F	1 byte	| for CHKSUM
Sensor IDhi		0x10	1 byte	|
SUCCESS_CODE	0xFF/ 0x00		1 byte	|
Checksum		0x??	1 byte	
DLE		0x10	1 byte	
ETX		0x03	1 byte	

"7": PCB RAPID ASSEMBLY TEST COMMAND (used for Zsensors, not SLS)

This command is used to test the linearization PROM connected to the 8255. The first word returned is the checksum of the PROM. The second word returned is the number of non-MASK bytes in the PROM. The search is started at address 0x0000 and ends at address 0x1FFF.

K16 sends:

DLE	0x10	1 byte	
STX	0x02	1 byte	
Sensor IDlo	0x??	1 byte	for CHKSUM
Sensor Idhi	0x??	1 byte	
PCB TEST command	0x07	1 byte	
Checksum	0x??	1 byte	
DLE	0x10	1 byte	
ETX	0x03	1 byte	

Sensor returns:

DLE	0x10	1 byte	
STX	0x02	1 byte	
Sensor IDlo	0x??	1 byte	for CHKSUM
Sensor Idhi	0x??	1 byte	
PROMCHKSUM_HI	0x??	1 byte	
PROMCHKSUM_LO	0x??	1 byte	
NONMASK_HI	0x??	1 byte	
NONMASK_LO	0x??	1 byte	
Checksum	0x??	1 byte	
DLE	0x10	1 byte	
ETX	0x03	1 byte	

"8": READ SENSOR SOFTWARE VERSION

This command is used to read a sensor's software version. This is number between 2 and 255. This is not available currently for the Smartlite.

K16 sends:

DLE	0x10	1 byte	
STX	0x02	1 byte	
Sensor IDlo	0x??	1 byte	for CHKSUM
Sensor Idhi	0x??	1 byte	
READ VERSION command	0x08	1 byte	
Checksum	0x??	1 byte	
DLE	0x10	1 byte	
ETX	0x03	1 byte	

Sensor returns:

DLE	0x10	1 byte	
STX	0x02	1 byte	
Sensor IDlo	0x??	1 byte	for CHKSUM
Sensor Idhi	0x??	1 byte	
Sensor Software Version	0x??	1 byte	
Checksum	0x??	1 byte	
DLE	0x10	1 byte	
ETX	0x03	1 byte	

"9": REQUEST SENSOR TEMPERATURE (16 BIT)

This command is not available in version 1 and 2 of the SLS sensor. The number returned is a signed 16 bit number. Bits 0-2 are always cleared. Bit 6 has a value of .390625 degC/count. The value is offset such that 0 counts = 0 degC, (So to get the same count as is returned by the 8 bit version of this command, divide this 16 bit value by 64).

K16 sends:

DLE	0x10	1 byte	
STX	0x02	1 byte	
Sensor IDlo	0x??	1 byte	for CHKSUM
Sensor Idhi	0x??	1 byte	
READ TEMP command	0x09	1 byte	
Checksum	0x??	1 byte	
DLE	0x10	1 byte	
ETX	0x03	1 byte	

Sensor returns:

DLE	0x10	1 byte	
STX	0x02	1 byte	
Sensor IDlo	0x??	1 byte	for CHKSUM
Sensor Idhi	0x??	1 byte	
Temp value high byte	0x??	1 byte	
Temp value low byte	0x??	1 byte	
Checksum	0x??	1 byte	
DLE	0x10	1 byte	
ETX	0x03	1 byte	

"10": SET BAUDRATE

This command is used to select the communication Baudrate. The Baudrate is changed only after a sensor soft reboot. The low speed Baudrate of 9600 is always selected after a hard reboot.

Set BAUDRATE baud = 57600 by writing 0FFH , any other value sets baud back to default of 9600

K16 sends:			
DLE	0x10	1 byte	
STX	0x02	1 byte	
Sensor IDlo	0x??	1 byte	for CHKSUM
Sensor Idhi	0x??	1 byte	|
SET BAUDRATE command	0x0A	1 byte	|
BAUDRATE	0x??	1 byte	|
Checksum	0x??	1 byte	
DLE	0x10	1 byte	
ETX	0x03	1 byte	
Sensor returns:			
DLE	0x10	1 byte	
ETX	0x03	1 byte	

The sensor must then be rebooted before the selected Baudrate will take effect. Of course it then takes 1-3 seconds after the reboot to be ready for more commands.

"11": SOFT REBOOT

This command is used to restart the sensor to its power up state. The health status is rechecked.

K16 sends:

DLE	0x10	1 byte	
STX	0x02	1 byte	
Sensor IDlo	0x??	1 byte	for CHKSUM
Sensor Idhi	0x??	1 byte	|
RESET command	0x0B	1 byte	|
Checksum	0x??	1 byte	
DLE	0x10	1 byte	
ETX	0x03	1 byte	

Sensor returns:

DLE	0x10	1 byte	
ETX	0x03	1 byte	

The sensor then takes 1-3 seconds to reboot and be ready for more commands.

"12": REQUEST ISP MODE

This command causes the sensor to go into the 'In System Programming' mode when hard rebooted.

K16 sends:

DLE	0x10	1 byte	
STX	0x02	1 byte	
Sensor IDlo	0x??	1 byte	for CHKSUM
Sensor Idhi	0x??	1 byte	|
RESET command	0x0C	1 byte	|
Checksum	0x??	1 byte	
DLE	0x10	1 byte	
ETX	0x03	1 byte	

There is a considerable delay (1-2 sec), then the

Sensor returns:

DLE	0x10	1 byte
ETX	0x03	1 byte

"13": ERASE FLASH DATA

This command is used to reset the stored linearization data including checksums to 0xff.

K16 sends:

DLE	0x10	1 byte	
STX	0x02	1 byte	
Sensor IDlo	0x??	1 byte	for CHKSUM
Sensor Idhi	0x??	1 byte	|
ERASE command	0x0D	1 byte	|
Checksum	0x??	1 byte	
DLE	0x10	1 byte	
ETX	0x03	1 byte	

There is a considerable delay (2-3 sec), then the

Sensor returns:

DLE	0x10	1 byte
ETX	0x03	1 byte