Pelco PTZ Protocols

# D Protocol

## Version 5.2.2

**February 20, 2012**

**Eric Hamilton**


**Proprietary Notice**



[1] $Header: PdfE.tex: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:13 PM$

### Notice of Disclaimer

Pelco makes no claims, expressed or implied, regarding the usefulness of this protocol, it's implementation, or it's correctness. Any use of this protocol is the sole responsibility of the agency implementing the protocol. The contents of this document and the function of the protocol are subject to change without notice.

### Questions

Questions regarding this protocol, it's implementation, use, and distribution should be addressed to:


| | |
|---|---|
| Pelco | (559) 292-1981 Voice |
| 3500 Pelco Way | (800) 289-9100 Voice |
| Clovis, California, USA | (559) 292-1018 FAX |
| 93612-5699 | `www.pelco.com` |


### Change Log

This is a total rewrite of the previously available protocol documentation for D protocol. Thus it is a new document.

The base line document for D Protocol was the former "Pelco Engineering 'D' Protocol Manual" TF-0001, Version 4, Revision 1, dated 4/7/2004, many e-mails and code inspection.

This is an abbreviated change log, the full change log is in Appendix Q.

# Contents



[2] tocdepth = 2

## List of Figures

# 1 What this Manual Covers

This manual describes the minimum requirements for implementing a common Pelco protocol. This protocol is used to communicate between a controlling device (e.g. a matrix switching system or keyboard) and a receiver/driver (e.g. a dome drive). **Pelco's primary protocol for controlling PTZ units is D Protocol**.

Not all devices will be able to accommodate all of the features available in these protocols. These protocols are designed to cover the feature sets of a wide variety of equipment.

D Protocol is a "serial" protocol. This means that it is normally transmitted over a 4 wire, RS-422 circuit. There is nothingthat actually requires usage of one or another format, however all Pelco PTZ units receive and transmit with RS-422[4] levels. If another communications media is desired, then it is the user's responsibility to turn the signals at the PTZ unit into RS-422 levels.

## 1.1 Protocol overview

Pelco's PTZ Serial Line protocols are a Master-Slave protocol. These protocols takes place at levels 1, 2 and 7 of the OSI model.

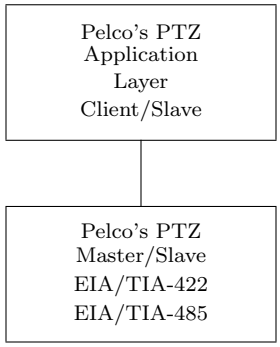
A master-slave type system has one node (the master node) that issues explicit commands to one of the "slave" nodes and processes responses. Slave nodes do not transmit data without a request from the master node, and do not communicate with other slaves.

At the physical level, Pelco's PTZ over Serial Line systems may use different physical interfaces (RS422, RS485, RS232). TIA/EIA-422 (RS422) Four-Wire, interface is the most common. As an option, TIA/EIA-485 (RS485) Four-Wire and TIA/EIA-485 (RS485) Two-Wire interface may also be implemented.

The following table gives a general representation of Pelco's PTZ serial communication stack compared to the 7 layers of the OSI model.

| Layer | ISO/OSI Model | |
|---|---|---|
| 7 | Application | Pelco's PTZ Application Protocols |
| | | D bytes 3 → 6 |
| 6 | Presentation | Empty |
| 5 | Session | Empty |
| 4 | Transport | Empty |
| 3 | Network | Empty |
| 2 | Data Link | Pelco's PTZ Serial Line Protocol |
| | | D Protocol, bytes 1, 2 and 7 |
| 1 | Physical | EIA/TIA-422 or EIA/TIA-485 |

[3] `$Header: Intro.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:13 PM$`
[4] The Spectra IV has introduced RS-485 logic levels.




$RCSfile: Intro.inc,v $

Pelco's PTZ application layer messaging protocol, positioned at level 7 of the OSI model, provides client/-server communication between devices connected on buses or networks. On Pelco's PTZ Serial Line the client role is provided by the Master of the serial bus and the Slave nodes act as servers.

## 1.2 Pelco's PTZ Master/Slave protocol principle

Pelco's PTZ Serial Line protocol is a Master-Slave protocol. Only one master is connected to the bus, and one or several (255 maximum number) slave nodes are also connected to the same serial bus. Pelco's PTZ communication is always initiated by the master. The slave nodes will never transmit data without receiving a request from the master node. The slave nodes will never communicate with each other. The master node initiates only one Pelco PTZ transaction at the same time. (I.e. there is no broadcast capability which addresses all units at the same time.)

The master node issues a Pelco PTZ request to the slave nodes in only a unicast mode, where master addresses an individual slave. After receiving and processing the request, the slave returns a message (a 'reply') to the master. In this mode, a Pelco PTZ transaction consists of 2 messages: a request from the master, and a reply from the slave. Each slave must have an unique address (from 1 to 255) so that it can be addressed independently from other nodes.

Pelco's PTZ Master node has no specific address, only the slave nodes have an address. The slave address must be unique on a Pelco PTZ serial bus.

There is no broadcast mode, where the the master can send a request to all slaves.

## 1.3 Fine Print Notes

In the descriptions of each command there is usually a set of notes that are marked "**FPN**". These are known as "**F**ine **P**rint **N**otes". While these are not part of the protocol they are indented to help with the understanding of the protocol. There is additional information about the protocol in the various appendices.

## 1.4 Byte Format

Transmitters will format a single character and receivers will be able to decipher a single character with: 1 start bit, 8 data bits, 1 stop bit, and no parity.

All units that support D Protocol have an ability to operate at 2400 baud. Other baud rates are supported on a device by device basis. There are Pelco units that operate with baud rates as high as 115,200. 2,400 is the lowest baud rate supported.

- Coding System: 8-bit binary

- Bits per Byte: 1 start bit
- 8 data bits, least significant bit sent first
- 1 stop bit
- No parity

FPN

## 1.5 Various Protocol Baud Rate Dependent Times

All units that support D Protocol have the ability to operate at 2400 baud. Most units also support 4800 and 9600 baud. Newer equipment supports baud rates of up to 115200 for special purposes (downloading of revised software). 2400 baud is the lowest speed supported by any Pelco PTZ units. Some of the TXBs supported by Pelco products need to work at slower speeds. In the following table the slower known baud rates used by our competitors is included for information.

1. In the below table, items that have "Yes" in the "Pelco Standard" column have been implemented in all current products. There is a special case and that is for the 115,200 baud rate that is only in the Spectra series units with an RJ-45 jack as an RS-232 level signal. All units that may be downloaded, support 115,200 during the download process.
2. Some of the items in the table are not usually supported directly by Pelco products. However our competitors do use these baud rates and they are used with TXBs and they are available as SMR options.
3. **Baud:**, the baud rate that this information applies to.
4. **Bit:**, time in seconds to send one bit at this baud.
5. **Byte:**, time in seconds to send one byte of 1 start, 8 data and 1 stop bit. (10 bits total)
6. **General:**, transmission time, in seconds of a general response. This does not include the time that the PTZ takes to generate the reply.
7. **Extended:**, transmission time, in seconds of a command or of an extended response. This does not include the time that the PTZ takes to generate the reply.
8. **Query:**, transmission time, in seconds to receive the query response. This does not include the time that the PTZ takes to generate the reply.

| Baud Rate | Bit Duration | Byte 10-bits | General 4-bytes | Extended 7-bytes | Query 18-bytes | Pelco Standard — |
|---|---|---|---|---|---|---|
| 300 | 0.003 333 | 0.033 333 | 0.133 333 | 0.233 333 | 0.600 000 | No |
| 600 | 0.001 667 | 0.016 667 | 0.066 667 | 0.116 667 | 0.300 000 | No |
| 1,200 | 0.000 833 | 0.008 333 | 0.033 333 | 0.058 333 | 0.150 000 | No |
| 2,400 | 0.000 417 | 0.004 167 | 0.016 667 | 0.029 167 | 0.075 000 | Yes |
| 4,800 | 0.000 208 | 0.002 083 | 0.008 333 | 0.014 583 | 0.037 500 | Yes |
| 9,600 | 0.000 104 | 0.001 042 | 0.004 167 | 0.007 292 | 0.018 750 | Yes |
| 14,400 | 0.000 069 | 0.000 694 | 0.002 778 | 0.004 861 | 0.012 500 | No |
| 19,200 | 0.000 052 | 0.000 521 | 0.002 083 | 0.003 646 | 0.009 375 | No |
| 28,800 | 0.000 035 | 0.000 347 | 0.001 389 | 0.002 431 | 0.006 250 | No |
| 38,400 | 0.000 026 | 0.000 260 | 0.001 042 | 0.001 823 | 0.004 688 | No |
| 115,200 | 0.000 009 | 0.000 087 | 0.000 347 | 0.000 608 | 0.001 563 | Yes |

# 2 General Command Information

**Command format** All commands are seven (D Protocol) bytes long. Note in the following formats that several bytes have been marked with double vertical lines. These are the portion of the message that comprise ISO level 7 data. The other bytes are ISO level 2 data.

It is convenient to consider the two CMND bytes as either two 8-bit `unsigned char` bytes or one 16-bit `unsigned int` when the two CMND bytes are concatenated together. The same hold true for the two DATA bytes. They may be considered to be two 8-bit `unsigned char` bytes or one 16-bit `unsigned int` value when concatenated together. The DATA bytes are always `unsigned char` or `unsigned int` values. Thus when a floating point value is needed the "real" value is multiplied by 10 or 100, depending on the actual command. Likewise since they are `unsigned` they are never negative. There is a problem in only one place in the protocol where this is a problem and that is in representing tilt readout angles. Pelco uses horizontal as 0° and displays it that way on the PTZ video, however when the camera in pointing down, i.e. below the horizon, the value on the screen is displayed as negative values (−12° is a value pointing down) and the protocol returns a value of $1200_{10}$ (`0x04B0`), see subsection 5.39 and subsection 5.46 for more details.

| D Protocol Command Format/Names | | | | | | | |
|---|---|---|---|---|---|---|---|
| Byte | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| | `0xFF` | — | `0x00` | `0x00` | `0x00` | `0x00` | — |

1. **D Protocol:**

   1.1 SYNC: Always `0xFF` to indicate the start of a command.

   1.2 ADDR: Camera address. Range is 0 → 255 (`0x00` → `0xFF`). `0x01` is for camera #1, etc. I.e address #0 is valid.

   1.3 CMND1: Extension of the basic command. The basic command is in the CMND2 position.

   1.4 CMND2: The basic command. For extended commands, all of these are odd numbers.

   1.5 DATA1: Usually this is `0x00` or the Pan speed index. However on a command by command basis it may have additional information. The most typical use is to make up a 16-bit value when concatenated with DATA2. For details, see the actual command description in section 5.

   1.6 DATA2: Usually is the argument for "this" command or the Tilt speed index. When a 16-bit value is required, it is the lower half of the value. (See section 5)

   1.7 CKSM: This is the arithmetic sum of all bytes except for the SYNC byte and itself.

   1.8 All D Protocol commands get replies. These may be General Response (4 bytes long subsection 4.2), Extended Response (7 bytes long subsection 4.3) or Query Response (18 bytes long subsection 4.4)[7]

## 2.1 General Notes

**Note**

1. Values in this document prefixed with "`0x`" are hexadecimal numbers.

[5] `$Header: Commands.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$`

[6] `$Header: Dbytes.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:12 PM$`

[7] There is also a "Super Extended Response" of 11 bytes length which is only used on one Pelco PTZ product. See subsection 5.57 for information about this response.

2. The symbol "—" is used to indicate variable data and is normally used in the ADDR and CKSM fields only.
3. The synchronization byte (SYNC) is always `0xFF` in D Protocol.
4. The Address (ADDR) is the logical address of the receiver/driver device being controlled. I.e. `0x05` in D Protocol will address camera #5.
5. The Checksum (CKSM) is calculated by performing an 8 bit (modulo 256) sum of the payload bytes (bytes $2 \rightarrow 6$) of the message in D Protocol. .

# 3 Command Sets

There are two sets/types of commands:

1. "Motion" commands, i.e. pan, tilt, iris, zoom and focus; are "bit encoded" commands that always have bit 0 in CMND2 set to `0`. Any number of non-exclusive bits may be set in the CMND1 and CMND2 bytes for this format of command[8].

   It is not legal to set bits for pan right **and** pan left in the same command, however it is legal to have control of up to five different motions in the same command. For historical reasons these commands are usually called "Standard Commands". (subsection 3.1)

   All Motion commands generate a 4-byte General Response subsection 4.2 if the command is good. A NAK reply is never generated.

2. "Non-motion" commands, i.e. call preset, request pan angular position; are "numerically encoded". These commands always have bit 0 in CMND2 set to a `1`. Thus all of these commands have "odd" numerical values and each command may only do one thing at a time. For historical reasons these commands are usually called "Extended Commands". (subsection 3.2)

[8] With the current camera manufacturer it has been found that setting Tele or Wide and either of Near or Far results in the camera module doing the Tele/Wide action for about $\frac{1}{4}$ second and then nothing. It is not recommended that combining Tele/Wide with Near/Far be used.

## 3.1 Standard Commands

### 3.1.1 Command bytes 1 and 2

**cmnd1** and **cmnd2** are represented as follows in D:

| **D Protocol**: Byte 3, CMND1 | | | | | | | |
|---|---|---|---|---|---|---|---|
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| Sense | 0 | 0 | Auto/-Manual Scan | Camera On/Off | Iris Close | Iris Open | Focus Near |

| **D Protocol**: Byte 4, CMND2 | | | | | | | |
|---|---|---|---|---|---|---|---|
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| Focus Far | Zoom Wide | Zoom Tele | Down | Up | Left | Right | Always 0 |

**All devices that support D Protocol, support these bit encoded commands.**

A value of '1' entered in the bit location for the function desired will enable that function. A value of '0' entered in the same bit location will disable or 'stop' the function.

Most Pelco equipment incorporates a "runaway protect" feature on all motion commands. What this means is that all motion causing commands will time out in about 15 seconds. (Some older equipment has a different timeout, but 15 seconds is used in the Spectra and Esprit systems.) To obtain continuous motion, a motion causing command should be sent about every 5 seconds.

The sense bit has meaning in D Protocol only, it (CMND1 bit 7) indicates the meaning of bits 4 and 3. If the sense bit is on (value of '1'), and bits 4 and 3 are on, the command will enable auto-scan and turn the camera on. If the sense bit is off (value of '0'), and bits 4 and 3 are on the command will enable manual scan and turn the camera off. Of course, if either bit 4 or bit 3 are off then no action will be taken for those features.

CMND1 bits $7 \rightarrow 3$ are not used with newer equipment in D Protocol.

| Sense bit<br>Bit 7 | Auto/-Manual Scan<br>Bit 4 | Camera On/Off<br>Bit 3 | Results |
|---|---|---|---|
| 0 | 0 | 0 | Nothing Happens |
| 0 | 0 | 1 | Camera Off |
| 0 | 1 | 0 | Manual Scan On |
| 0 | 1 | 1 | Manual Scan On, Camera Off |
| 1 | 0 | 0 | Nothing Happens |
| 1 | 0 | 1 | Camera On |
| 1 | 1 | 0 | Auto Scan On |
| 1 | 1 | 1 | Auto Scan On, Camera On |

It should be pointed out that newer Pelco units such as the Spectra and Esprit systems, always have their cameras turned on and that this command does not turn them off.

All reserved bits ($6 + 5$ or $7 \rightarrow 4$) must be set to 0.

[9] `$Header: DBits.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:12 PM$`

### 3.1.2 Data bytes 1 and 2

**Pan/Data1** Byte 5 (**data1**) contains the pan speed. Pan speed is in the range of '0x00' to '0x3F' (high speed) and '0x40' for "turbo" speed. Turbo speed is the maximum speed the device can obtain and is considered separately because it is not generally a smooth step from high speed to turbo. That is, going from one speed to the next usually looks smooth and will provide for smooth motion with the exception of going into and out of turbo speed. A pan speed value of '0x00' results in very slow motion, not cessation of motion. To stop pan motion both the Left and Right direction bits must be turned off — set to '0' — regardless of the value set in the pan speed byte. This is because a speed of 0x00 is a valid, but slow, speed. Typical pan speeds vary from 0.1° /sec to 80° /sec.

**Tilt/Data 2** Byte 6 (**data2**) contains the tilt speed. Tilt speed is in the range of '0x00' to '0x3F' (maximum speed). Turbo speed is not allowed for the tilt axis. A tilt speed value of '0x00' results in very slow motion, not cessation of motion. To stop tilt motion both the Down and Up direction bits must be turned off — set to '0' — regardless of the value set in the tilt speed byte. This implies that when the Up and Down bits are set to 0, this value must be ignored by decoding devices. Having Up and Down set at the same time is an error. Typical tilt speeds vary from 0.1° /sec to 40° /sec.

### 3.1.3 Checksum byte

In D Protocol, byte 7 is the checksum (**cksm**). The checksum is the 8 bit (modulo 256) sum of the payload bytes (bytes 2 → 6) in the message.

## 3.2 Extended Commands

In addition to the "PTZ" commands shown in subsection 3.1, there are control commands that allow access to the more advanced features of some equipment. Byte 4 (CMND2) can be thought of as the command's opcode. Byte 3 (CMND1 can be thought of as the command's sub-opcode. Byte 4's value is given in hexadecimal and decimal formats as in many listings this value is used either way in the `#define`s. See the description of each command type for details.

| Command | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Response Type |
|---|---|---|---|---|---|
| Standard Extended Response subsection 5.1 | ACK/NAK | 0x01/$1_{10}$ | Future Use | Future Use | Not Applicable This is not a command it is a reply. |
| Set Preset subsection 5.2 | 0x00 | 0x03/$3_{10}$ | 0x00 | PRESET ID | General |
| Clear Preset subsection 5.3 | 0x00 | 0x05/$5_{10}$ | 0x00 | PRESET ID | General |
| Go To Preset subsection 5.4 | 0x00 | 0x07/$7_{10}$ | 0x00 | PRESET ID | General |
| Flip (180° about) subsection 5.4 | 0x00 | 0x07/$7_{10}$ | 0x00 | 0x21/33 | General |

*Continued on the next page.*

*Continued from the previous page.*

| Command | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Response Type |
|---|---|---|---|---|---|
| Go To Zero Pan subsection 5.4 | 0x00 | 0x07/$7_{10}$ | 0x00 | 0x22/34 | General |
| Set Auxiliary subsection 5.5 | SUB OP-CODE | 0x09/$9_{10}$ | 0x00 | AUX ID | General |
| Clear Auxiliary subsection 5.6 | SUB OP-CODE | 0x0B/-$11_{10}$ | 0x00 | AUX ID | General |
| Dummy subsection 5.7 | SUB OP-CODE | 0x0D/-$13_{10}$ | 0x00 | 0x00 | General |
| Remote Reset subsection 5.8 | 0x00 | 0x0F/-$15_{10}$ | 0x00 | 0x00 | General |
| Set Zone Start subsection 5.9 | 0x00 | 0x11/-$17_{10}$ | 0x00 | ZONE ID | General |
| Set Zone End subsection 5.10 | 0x00 | 0x13/-$19_{10}$ | 0x00 | ZONE ID | General |
| Write Character to Screen subsection 5.11 | 0x00 | 0x15/-$21_{10}$ | Column 0 $\rightarrow$ $39_{10}$ | ASCII Value | General |
| Clear Screen subsection 5.12 | 0x00 | 0x17/-$23_{10}$ | 0x00 | 0x00 | General |
| Alarm Acknowledge subsection 5.13 | SUB OP-CODE | 0x19/-$25_{10}$ | 0x00 | ALARM ID | General |
| Zone Scan On subsection 5.14 | 0x00 | 0x1B/-$27_{10}$ | 0x00 | 0x00 | General |
| Zone Scan Off subsection 5.15 | 0x00 | 0x1D/-$29_{10}$ | 0x00 | 0x00 | General |
| Record Pattern Start subsection 5.16 | 0x00 | 0x1F/-$31_{10}$ | 0x00 | PATTERN ID | General |
| Record Pattern Stop subsection 5.17 | 0x00 | 0x21/-$33_{10}$ | 0x00 | 0x00 | General |
| Run Pattern subsection 5.18 | 0x00 | 0x23/-$35_{10}$ | 0x00 | PATTERN ID | General |
| Set Zoom Speed subsection 5.19 | 0x00 | 0x25/-$37_{10}$ | 0x00 | ZOOM SPEED | General |
| Set Focus Speed subsection 5.20 | 0x00 | 0x27/-$39_{10}$ | 0x00 | FOCUS SPEED | General |
| Reset Camera defaults subsection 5.21 | 0x00 | 0x29/-$41_{10}$ | 0x00 | 0x00 | General |
| Auto-focus auto/on/-off subsection 5.22 | 0x00 | 0x2B/-$43_{10}$ | 0x00 | AUTO FOCUS CTRL | General |

*Continued on the next page.*

*Continued from the previous page.*

| Command | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Response Type |
|---|---|---|---|---|---|
| Auto Iris auto/on/off subsection 5.23 | 0x00 | 0x2D/- $45_{10}$ | 0x00 | AUTO IRIS CTRL | General |
| AGC auto/on/off subsection 5.24 | 0x00 | 0x2F/- $47_{10}$ | 0x00 | AGC CONTROL | General |
| Backlight compensation on/off subsection 5.25 | 0x00 | 0x31/- $49_{10}$ | 0x00 | BLC CONTROL | General |
| Auto white balance on/off subsection 5.26 | 0x00 | 0x33/- $51_{10}$ | 0x00 | AWB CONTROL | General |
| Enable device phase delay mode subsection 5.27 | 0x00 | 0x35/- $53_{10}$ | 0x00 | 0x00 | General |
| Set shutter speed subsection 5.28 | 0x00 | 0x37/- $55_{10}$ | SPEED MSB | SPEED LSB | General |
| Adjust line lock phase delay subsection 5.29 | SUB OPCODE | 0x39/- $57_{10}$ | DELAY MSB | DELAY LSB | General |
| Adjust white balance (R-B) subsection 5.30 | SUB OPCODE | 0x3B/- $59_{10}$ | WB-RB MSB | WB-RB LSB | General |
| Adjust white balance (M-G) subsection 5.31 | SUB OPCODE | 0x3D/- $61_{10}$ | WB-MG MSB | WB-MG LSB | General |
| Adjust gain subsection 5.32 | SUB OPCODE | 0x3F/- $63_{10}$ | GAIN MSB | GAIN LSB | General |
| Adjust auto-iris level subsection 5.33 | SUB OPCODE | 0x41/- $65_{10}$ | 0x00 | AIL VALUE | General |
| Adjust auto-iris peak value subsection 5.34 | SUB OPCODE | 0x43/- $67_{10}$ | 0x00 | AIP VALUE | General |
| Query subsection 5.35 | SUB OPCODE | 0x45/- $69_{10}$ | 0x00 | 0x00 | Query subsection 4.4. |
| Preset Scan subsection 5.36 | 0x00 | 0x47/- $71_{10}$ | 0x00 | DWELL | General |
| Set Zero Position subsection 5.37 | 0x00 | 0x49/- $73_{10}$ | 0x00 | 0x00 | General |
| Set Pan Position subsection 5.38 | 0x00 | 0x4B/- $75_{10}$ | PAN MSB | PAN LSB | General |

*Continued on the next page.*

*Continued from the previous page.*

| Command | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Response Type |
|---|---|---|---|---|---|
| Set Tilt Position subsection 5.39 | 0x00 | 0x4D/- $77_{10}$ | TILT MSB | TILT LSB | General |
| Set Zoom Position subsection 5.40 | 0x00 | 0x4F/- $79_{10}$ | ZOOM MSB | ZOOM LSB | General |
| Query Pan Position subsection 5.41 | 0x00 | 0x51/- $81_{10}$ | 0x00 | 0x00 | Extended subsection 5.45 |
| Query Tilt Position subsection 5.42 | 0x00 | 0x53/- $83_{10}$ | 0x00 | 0x00 | Extended subsection 5.46 |
| Query Zoom Position subsection 5.43 | 0x00 | 0x55/- $85_{10}$ | 0x00 | 0x00 | Extended subsection 5.47 |
| Download subsection 5.44 | 0x00 | 0x57/- $87_{10}$ | 0x00 | 0x00 | General |
| Query Pan Response subsection 5.45 | 0x00 | 0x59/- $89_{10}$ | PAN MSB | PAN LSB | Not Applicable This is not a command it is a reply. |
| Query Tilt Response subsection 5.46 | 0x00 | 0x5B/- $91_{10}$ | TILT MSB | TILT LSB | Not Applicable This is not a command it is a reply. |
| Query Zoom Response subsection 5.47 | 0x00 | 0x5D/- $93_{10}$ | ZOOM MSB | ZOOM LSB | Not Applicable This is not a command it is a reply. |
| Set Magnification subsection 5.48 | SUB OPCODE | 0x5F/- $95_{10}$ | MAG MSB | MAG LSB | General |
| Query Magnification subsection 5.49 | 0x00 | 0x61/- $97_{10}$ | 0x00 | 0x00 | Extended subsection 5.50 |
| Query Magnification Response subsection 5.50 | 0x00 | 0x63/- $99_{10}$ | MAG MSB | MAG LSB | Not Applicable This is not a command it is a reply. |
| Activate Echo Mode subsection 5.51 | 0x00 | 0x65/- $101_{10}$ | 0x00 | 0x00 | General |
| Set Remote Baud Rate subsection 5.52 | SUB OPCODE | 0x67/- $103_{10}$ | 0x00 | 00-05 | General |
| Start Download subsection 5.53 | 0x00 | 0x69/- $105_{10}$ | 0x00 | 0x00 | General |
| Query Device Type subsection 5.54 | 0x00 | 0x6B/- $107_{10}$ | 0x00 | 0x00 | Extended subsection 5.55 |
| Query Device Type Response subsection 5.55 | 0x00 | 0x6D/- $109_{10}$ | SOFTWARE TYPE | HARDWARE TYPE | Not Applicable This is not a command it is a reply. |

*Continued on the next page.*

| *Continued from the previous page.* | | | | | |
|---|---|---|---|---|---|
| Command | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Response Type |
| Query Diagnostic Info subsection 5.56 | 0x00 | 0x6F/-$111_{10}$ | 0x00 | 0x00 | Extended subsection 5.57 |
| Query Diagnostic Info Response subsection 5.57 | 0x00 | 0x71/-$113_{10}$ | Device Dependent | Device Dependent | Not Applicable This is not a command it is a reply. |
| Version Information Macro Opcode subsection 5.58 | SUB OPCODE | 0x73/-$115_{10}$ | Various | Various | Extended subsection 5.58 |
| Everest Macro Opcode subsection 5.59 | SUB OPCODE | 0x75/-$117_{10}$ | Various | Various | Extended subsection 5.59 |
| Time Set Opcode subsection 5.60 | SUB OPCODE | 0x77/-$119_{10}$ | Various | various | Extended subsection 5.60 |
| Screen Move subsection 5.61 | SUB OPCODE | 0x79/-$121_{10}$ | MAG MSB | MAG LSB | General |
| Return Status | 0x00 | 0xFD/-$253_{10}$ | 0x00 | 0x00 | Firmware status |
| Invalid op code | 0x00 | 0xFF/-$255_{10}$ | 0x00 | 0x00 | General |

## 3.3 Creating Labels

Many devices have an ability to display labels on the video. Labels that identify the preset or zone being scanned are common. There is a special technique to establish a label that is associated with either a preset or a zone. First, send the label to the receiver/driver using the "Write Character to Screen" (subsection 5.11) command. After the label is on the screen, then set the preset or zone (subsubsection 5.9.3). That will establish the label and associate it with that preset or zone.

FPN

### 3.3.1 Label Logic in Spectra IV

In response to questions on labels as used in the Spectra IV[11], This is an informal spec on how they currently work in Spectra IV. There is additional information in subsection 5.14.

1. Azimuth Elevation, Direction, and Zoom Magnification:
   - 1.1 PTZ
     - 1.1.1. Labels pop on-screen when there is movement and disappear after movement stops and the display time has expired.
   - 1.2 Patterns
     - 1.2.1. While recording, the labels are shown while there is movement just like normal PTZ.
     - 1.2.2. While playing, the labels are hidden.
   - 1.3 Presets
     - 1.3.1. On a preset GO command, the labels are hidden.
   - 1.4 Scans
     - 1.4.1. While scanning, the labels are hidden.

   Of course if a label's display time is set to `CONSTANT`, it always shows no matter what (except in menus of course).
2. Alarm:
   - 2.1 1 Alarm Active
     - 2.1.1. Alarm label is displayed for display time, then goes off until alarm clears and retriggers.
   - 2.2 Greater than 1 Alarm Active
     - 2.2.1. Alarm labels will sequence on-screen (switching every X seconds) to the next active label in the sequence.
     - 2.2.2. If the sequence time is greater than the display time for the alarm labels, the alarm will disappear until the next sequence.

   In the case of `CONSTANT` display time, alarm labels will remain on-screen until that alarm clears or the next alarm in sequence fires.

   How are alarm priorities handled when two different levels are active? Do they alternate, or only the higher priority displayed?

   Only alarms of the highest priority count... If there are 4 alarms present and only 1 is HIGH priority, it falls into the 1 alarm active category.
3. Preset, Zone:
   - 3.1 On a preset GO command, the preset label pop on-screen when the preset is reached.
     - 3.1.1. In the case of `CONSTANT` display time, this label remains as long as the camera sits at the preset.
   - 3.2 When the camera passes through a zone or is sitting on a zone, the zone label is displayed.
     - 3.2.1. In the case of `CONSTANT` display time, this label remains as long as the camera remains in the zone, otherwise it disapears after the display time has expired.
4. Time/Date:
   - 4.1 Time/Date information is either on the screen at all times or not.

[10] `$Header: Labels.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:13 PM$`
[11] This information about labels came from: Jeremy Watson in November 2006.

## 3.4 Example D Protocol Messages

| D Protocol Sample Commands | | |
|---|---|---|
| Sample # | Message to send | Message |
| 1 | Camera 2, Pan Left | 0xFF, 0x02, 0x00, 0x04, 0x20, 0x00, 0x26 |
| 2 | Camera 2, Stop | 0xFF, 0x02, 0x00, 0x00, 0x20, 0x00, 0x22 |
| 3 | Camera 10, Camera on, Focus far, Tilt Down | 0xFF, 0x0A, 0x88, 0x90, 0x00, 0x20, 0x42 |

**Note**

1. In sample message 2, please note that the pan speed field has not been set to 0x00, but rather has a value in it. When the pan bits in CMND2 are both set to '0', the motion value is ignored.
2. The checksum calculation for message #3 goes like this:

| | | |
|---|---|---|
| 0xFF | 1111 1111 | Sync byte is not used for the checksum |
| 0x0A | 0000 1010 | |
| 0x88 | 1000 1000 | |
| Subtotal | 1001 0010 | 0x92 |
| 0x90 | 1001 0000 | |
| Subtotal | 0010 0010 | 0x22 (modulo 256 allows the high bit to roll off) |
| 0x00 | 0000 0000 | |
| Subtotal | 0010 0010 | 0x22 |
| 0x20 | 0010 0000 | |
| Subtotal | 0100 0010 | 0x42 |
| | 0100 0010 | 0x42 Final checksum value |

| D Protocol Command Format/Names | | | | | | | |
|---|---|---|---|---|---|---|---|
| Byte | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| | 0xFF | — | 0x00 | 0x00 | 0x00 | 0x00 | — |

| **D Protocol**: Byte 3, CMND1 | | | | | | | |
|---|---|---|---|---|---|---|---|
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| Sense | 0 | 0 | Auto/-Manual Scan | Camera On/Off | Iris Close | Iris Open | Focus Near |

| **D Protocol**: Byte 4, CMND2 | | | | | | | |
|---|---|---|---|---|---|---|---|
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| Focus Far | Zoom Wide | Zoom Tele | Down | Up | Left | Right | Always 0 |

# 4 Responses

Devices that receive a "D" protocol command always generate a response[15]. The response formats are described below. There is no "negative acknowledge" response in D protocol. Units that implement D protocol have a receive timer that expects all bytes to be received in less than about 250 milliseconds. If it takes longer than the timeout, then the unit discards the current command and attempts to receive a new command. If a command is "half sent" and then a long delay (over 250 ms.) occurs and then the command finishes up it will be ignored unless a second command is immediately initiated. When this happens, the second command, at some intermediate point, is assumed to be a checksum. If this random byte is an actual checksum, then anomalous behavior will occur.

This is caused by the fact that there is no unique "first byte" in D protocol. The sync byte of 0xFF is also a data byte in many commands. The only way to be sure that a command is accepted in D protocol is to check the response. To ensure that no "confusion" occurs in the Pelco receiving equipment, a delay of at least 300 milliseconds must be inserted between sending commands. Also when a command is sent, do not have long delays in the middle.

1. **D Protocol:** Most D Protocol commands generate a response. When they generate a response there are three/four different lengths for these responses. The lengths are:

   1.1 **0 bytes:** For some commands no response is generated.

   1.2 **4 bytes:** This is called the "**General Response**" and almost all commands generate it (subsection 4.2).

   1.3 **7 bytes:** This is called the "**Extended Response**" and is normally sent as a response when asking for data. It is also sent when some commands are not understood (subsection 4.3).

   1.4 **18 bytes:** This is generated by the "**Query**" command and its variants. It originally was the software part number, then it became the equipment model number. Recently it has also provided the unit's serial number (subsection 4.4).

[12] $Header: Dbytes.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:12 PM$
[13] $Header: DBits.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:12 PM$
[14] $Header: Response.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:13 PM$
[15] Samples of command/response pairs are in Appendix A.

## 4.1 The Standard Extended Response

A seven byte response, see subsection 5.1 for details

## 4.2 The General Response

The General Response is four bytes in length and has the following format:

| General Response Format | | | | |
|---|---|---|---|---|
| Byte | 1 | 2 | 3 | 4 |
| | SYNC | ADDR | ALARMS | CKSM |
| | 0xFF | — | 0x00 | — |

1. SYNC: Always 0xFF to indicate the start of a response.
2. ADDR: Camera address. Range is 1 → 255 (0x01 → 0xFF). 0x01 is for camera #1, etc.
3. ALARMS: This is a bit encoded byte of information. Bit 0 = Alarm 1, etc.
4. CKSM: This is the arithmetic sum of **the checksum of the command that caused this reply** and the ALARMS field of this reply. (Not exactly what might be expected.)

**In the following data capture, several commands with no alarms active are sent:** Note the General Responses.

1. A Zoom In is sent for camera 1.
2. A Motion Stop is sent to stop the zooming.
3. A Call Preset 1 is sent.

```
DTE = D Protocol GlassKeyboard
DCE = Spectra III

     1,       1:  DTE     0.000000   0.000000 ff 01 00 20 00 00 21   Zoom In
     2,       1: DCE      0.011001   0.004762 ff 01 00 21            Response
     2,       8:  DTE     0.219946   0.001041 ff 01 00 00 00 00 01   Motion Stop
     3,       5: DCE      0.413586   0.187405 ff 01 00 01            Response
     3,      15:  DTE    85.024799   0.001041 ff 01 00 07 00 01 09   Call Preset 1
     4,       9: DCE     85.035699   0.004666 ff 01 00 09            Response
```

## 4.3 The Extended Response

The Extended Response is seven bytes in length and has the following format:

| Extended Response Format | | | | | | | |
|---|---|---|---|---|---|---|---|
| Byte | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | SYNC | ADDR | RESP1 | RESP2 | DATA1 | DATA2 | CKSM |
| | 0xFF | — | 0x00 | 0x00 | 0x00 | 0x00 | — |

1. SYNC: Always 0xFF to indicate the start of a reply.
2. ADDR: Camera address. Range is 1 → 255 (0x01 → 0xFF). 0x01 is for camera #1, etc.
3. RESP1: Reply specific data.

4. RESP2: Reply specific data. Usually a secondary op-code.
5. DATA1: Reply specific data.
6. DATA2: Reply specific data.
7. CKSM: This is the arithmetic sum of all bytes except for the SYNC byte and its self.

**In the following data capture, several commands are sent:** Note the General Responses and Extended Responses.

```
DCE = D Protocol GlassKeyboard
DTE = Spectra IV

    14,      92: DCE     44.237911   1.136292 ff 01 00 07 00 22 2a   Goto Preset 34
    14,      87:  DTE    44.276490   0.004139 ff 01 00 2a            General Response
    15,      99: DCE     49.423430   5.134441 ff 01 00 51 00 00 52   Get Pan Position
    15,      91:  DTE    49.453440   0.004168 ff 01 00 59 00 00 5a   Pan 0 degrees Extended Response
```

## 4.4 The Query Response

The Extended Response is eighteen bytes in length and has the following format:

| Query Response Format | | | | | | |
|---|---|---|---|---|---|---|
| Byte | 1 | 2 | 3 | . . . . | 17 | 18 |
| | SYNC | ADDR | DATA1 | . . . . | DATA15 | CKSM |
| | 0xFF | — | 0x00 | . . . . | 0x00 | — |

1. SYNC: Always 0xFF to indicate the start of a reply.
2. ADDR: Camera address. Range is 1 → 255 (0x01 → 0xFF). 0x01 is for camera #1, etc.
3. DATA1 → DATA15: 15 bytes of reply specific data.
4. CKSM: This is the arithmetic sum of all bytes **and the checksum of the command that caused this reply** except for the SYNC byte and its self. (Again not what might be expected.)
5. See subsection 5.35 and Appendix A for more details and samples of operation.

FPN

**Typical QUERY command and its response:** This is a data capture from a QUERY command and its response. The system was running at 2400 baud and the Spectra III SE was running rev 3.31 software.

**Table Notes**

The table consists of two parts:

- In the first part, the various command response bytes have been assembled to make the commands and responses easier to identify.
- In the second part is the semi-raw data returned by the data capture software.

This raw data capture files has been "post-processed" to change the date time stamp so as to have all times be relative to the start of the data capture and to get Δ times between bytes and messages.

Column headings are:

**Msg #** This is the message number. Numbers are sequential within each data source. I.e. there is one set of message numbers for each of DCE and DTE sources.

**Byte #** this is the total byte number within each data source.

**DTE/DCE Byte** This is the source identifier for each data source type. The abbreviations are:

- **DCE** = Data Communications Equipment, in this case the Spectra III and
- **DTE** = Data Terminal Equipment, which is a TXB that is under development.

**Total Byte #** This is the total byte number. I.e. it is the total of both DCE and DTE bytes.

**Total Time** This time has been normalized to start at 0.00000. The data capture software provides a date and time tag for each byte. The post-processing software deletes the date, knows that there 60 seconds in a minute and 60 minutes in an hour and knows how to work with the transition between noon (12:00) and one o'clock (01:00). This has been done to eliminate unexpected time jumps in the time record.

**Δ Time** This is the time between this byte and the previous byte.

**Data** This is the hexadecimal value of the byte recorded.

| Msg # | Byte # | DTE/-DCE Source | Total Byte # | Total Time | Δ Time | Data |
|---|---|---|---|---|---|---|
| 1 | 1 | DTE | 1 | 0.000000 | 0.000000 | ff 01 00 45 00 00 46 |
| 2 | 1 | DCE | 8 | 0.029773 | 0.004778 | ff 01 44 44 35 33 43 42 57 00 00 00 00 00 00 00 00 13 |

| Msg # | Byte # | DTE/-DCE Source | Total Byte # | Total Time | Δ Time | Data |
|---|---|---|---|---|---|---|
| 1 | 1 | DTE | 1 | 0.000000 | 0.000000 | ff |
| 1 | 2 | DTE | 2 | 0.004162 | 0.004162 | 01 |
| 1 | 3 | DTE | 3 | 0.008329 | 0.004167 | 00 |
| 1 | 4 | DTE | 4 | 0.012496 | 0.004167 | 45 |
| 1 | 5 | DTE | 5 | 0.016661 | 0.004165 | 00 |
| 1 | 6 | DTE | 6 | 0.020828 | 0.004167 | 00 |
| 1 | 7 | DTE | 7 | 0.024995 | 0.004167 | 46 |
| 2 | 1 | DCE | 8 | 0.029773 | 0.004778 | ff |
| 2 | 2 | DCE | 9 | 0.033939 | 0.004166 | 01 |
| 2 | 3 | DCE | 10 | 0.038107 | 0.004168 | 44 “D” |
| 2 | 4 | DCE | 11 | 0.042274 | 0.004167 | 44 “D” |
| 2 | 5 | DCE | 12 | 0.046442 | 0.004168 | 35 “5” |
| 2 | 6 | DCE | 13 | 0.050608 | 0.004166 | 33 “3” |
| 2 | 7 | DCE | 14 | 0.054747 | 0.004139 | 43 “C” |
| 2 | 8 | DCE | 15 | 0.058934 | 0.004187 | 42 “B” |

*Continued on the next page.*

*Continued from the previous page.*

| Msg # | Byte # | DTE/-DCE Source | Total Byte # | Total Time | Δ Time | Data |
|---|---|---|---|---|---|---|
| 2 | 9 | DCE | 16 | 0.063081 | 0.004147 | 57 “W” |
| 2 | 10 | DCE | 17 | 0.067248 | 0.004167 | 00 “null” |
| 2 | 11 | DCE | 18 | 0.071414 | 0.004166 | 00 “null” |
| 2 | 12 | DCE | 19 | 0.075613 | 0.004199 | 00 “null” |
| 2 | 13 | DCE | 20 | 0.079747 | 0.004134 | 00 “null” |
| 2 | 14 | DCE | 21 | 0.083914 | 0.004167 | 00 “null” |
| 2 | 15 | DCE | 22 | 0.088086 | 0.004172 | 00 “null” |
| 2 | 16 | DCE | 23 | 0.092294 | 0.004208 | 00 “null” |
| 2 | 17 | DCE | 24 | 0.096414 | 0.004120 | 00 “null” |
| 2 | 18 | DCE | 25 | 0.100581 | 0.004167 | 13 |

Calculations for the checksum for the QUERY response goes like this:

1. Add up all bytes in the QUERY response, except for the SYNC byte. (DCE bytes 2 → 17 in message 2 above.)
2. Add in the checksum from the **originating** QUERY command. (DTE byte 7 in message 1 above.)
3. Use the lower eight bits of the sum as the checksum. (DCE byte 18 in message 2 above.)

```
0x01 + 0x44 + 0x44 + 0x35 + 0x33 + 0x43 +
0x42 + 0x57 + 0x00 + 0x00 + 0x00 + 0x00 +
0x00 + 0x00 + 0x00 + 0x00 = 0x1CD
0x1CD + 0x46 = 0x213
0x213 & 0xFF = 0x13
```

# 5 Detailed Command Descriptions

## 5.1 Response 0x01 ($1_{10}$), Standard Extended Response

### 5.1.1 Response format

| Standard Extended Response D_EC_STD_EXT_RESP | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | RESP TYPE | 0x01 | FUTURE USE | FUTURE USE | — |
| 0xFF | — | ACK | 0x01 | 0x00 | 0x00 | — |
| 0xFF | — | NAK | 0x01 | 0x00 | 0x00 | — |

FPN

1. This response is used by the Endura/Atlas project.
2. This response is generated when a command may not be executed (these commands generate a NAK) or when some of the newer commands have been accepted (these may generate a NAK or an ACK).

   In systems prior to Spectra IV, if a command was not understood, a General Response was always sent if the command had a "proper" structure. I.e. a first Sync byte, the correct address, was seven bytes long and had a good checksum. Some of the newer commands in the Spectra IV, do a logical check to see if a command is correct. As of December 2007, the only commands that do this are the commands for setting the time (subsection 5.60).

### 5.1.2 Description

1. RESP TYPE: indicates whether the response is a NAK or an ACK.

   - 0x00 = NAK (D_ECS_STD_EXT_RESP_NAK)
   - 0x01 = ACK (D_ECS_STD_EXT_RESP_ACK)

   The information in the FUTURE USE bytes is specific to the command that is being responded to. If the FUTURE USE bytes are not used for particular response, the bytes must be set to zero.

## 5.2 Command 0x03 ($3_{10}$), Set Preset

### 5.2.1 Command format

| Set Preset D_EC_SET_PRESET | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x03 | 0x00 | PRESET ID | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:

[16] $Header: 01.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
[17] $Header: 03.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$

2.1 0x0003 SET_PRESET

3. This command is used by the Endura/Atlas projects.
4. This command is used and decoded on the:
   ERD97P21, Esprit 3012, Intercept and LRD.
5. The defined presets may be determined by using QUERY DEFINED PRESETS (subsubsection 5.59.17) and its reply of
   QUERY DEFINED PRESETS RESPONSE (subsubsection 5.59.18).
6. A list of typical presets is in Appendix G.

### 5.2.2 Description

Presets can be moved to, set, or cleared.

When a move to preset command is received, the preset position stored for the preset number specified in the command is checked. If the position is not valid, the command is ignored. Otherwise the unit moves to the preset pan, tilt, zoom, and focus positions. Once the preset has been reached, the preset label is displayed on the second video line or where it has been moved through use of the SET 95 menu system.

If any command which causes motion is received during a move to preset, the move will be aborted and the new command will start. These commands are: a motion command, or another move to preset command, Also if the move is not completed within a timeout period, the move is aborted and motion is stopped.

When a SET PRESET command is received, the current pan, tilt, focus, and zoom positions are saved for the preset number specified in the command and the label for that preset becomes whatever is currently on the second video line.

Usually this command will cause the camera system to remember where it is currently pointing. Other times it will cause a specific action to occur. The most common of specific action is a menu call command with either SET PRESET 95 (or SET PRESET 28 in 32 preset mode).

Pre-assigned presets may not be used for position setting. If an attempt to do so is done, then the command is ignored with a General Reply being returned.

1. DATA2 This is the preset number. The range of this value is: 0x01 $\rightarrow$ 0xFF, 1 $\rightarrow$ $255_{10}$. Different camera systems have differing number of preset numbers. Preset 0 is invalid. To get a model specific count of the available presets see the unit's manual.

Spectra III and ExSite saves in addition to the pan/tilt/zoom information for a preset through use of the menu system all items in the appropriate "preset camera setting screens". A total of ten (10) presets may have this special capability. With the Spectra IV this capability has been increased to all available presets.

When the unit has a password enabled, this command will cause the unit to display the password request screen and not proceed until a correct password is entered. There is no way to enter a password directly via D Protocol.

## 5.3 Command 0x05 ($5_{10}$), Clear Preset

### 5.3.1 Command format

| Preset Clear<br>D_EC_CLEAR_PRESET | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x05 | 0x00 | PRESET ID | — |

[18] $Header: 05.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$

FPN

1. This command generates a "General Reply".
2. Command Names as used by the Spectra IV software:
   2.1 `0x0005 CLEAR_PRESET`
3. This command is used by the Endura/Atlas projects.
4. This command is used and decoded on the:
   ERD97P21, Esprit 3012, Intercept and LRD.
5. See SET PRESET (subsection 5.2) for more information about this command.
6. The defined presets may be determined by using QUERY DEFINED PRESETS (subsubsection 5.59.17) and its reply of
   QUERY DEFINED PRESETS RESPONSE (subsubsection 5.59.18).
7. A list of typical presets is in Appendix G.

### 5.3.2 Description

The CLEAR PRESET command makes the stored preset for the preset number specified in the command invalid so that it can not be moved to.

Clears the requested preset's information from the camera system. Does not affect any of the preassigned presets.

It is not necessary to clear a preset before setting it.

Pre-assigned presets may not be cleared.

1. DATA2 This is the preset number. The range of this value is: `0x01` → `0xFF`, $1 \rightarrow 255_{10}$. Different camera systems have differing number of preset numbers. Preset `0` is invalid. To get a model specific count of the available presets see the unit's manual.

When the unit has a password enabled, this command will cause the unit to display the password request screen and not proceed until a correct password is entered. There is no way to enter a password via D Protocol.

## 5.4 Command `0x07` ($7_{10}$), Call Preset

### 5.4.1 Command format

| Call Preset<br>D_EC_MOVE_PRESET | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x07 | 0x00 | PRESET ID | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 `0x0007 GO_TO_PRESET`
   2.2 `0x0007 FLIP`
   2.3 `0x0007 GO_TO_ZERO_PAN`
3. This command is used by the Endura/Atlas projects.

[19] `$Header: 07.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$`

4. This command is used and decoded on the:
   ERD97P21, IRD/ERD, Esprit 3012, Intercept and LRD.
5. See SET PRESET (subsection 5.2) for more information about this command.
6. The defined presets may be determined by using QUERY DEFINED PRESETS (subsubsection 5.59.17) and its reply of
   QUERY DEFINED PRESETS RESPONSE (subsubsection 5.59.18).
7. A list of typical presets are in Appendix G.

### 5.4.2 Description

Causes the camera unit to move, at preset speed, to the requested position.

1. DATA2 This is the preset number. The range of this value is: 0x01 → 0xFF, $1 \rightarrow 255_{10}$. Different camera systems have differing number of preset numbers. Preset 0 is invalid. To get a model specific count of the available presets see the unit's manual.

FPN

### 5.4.3 Special Presets

| Flip (rotate 180° ) D_EC_MOVE_PRESET | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x07 | 0x00 | 0x21 | — |

| Go To Zero Pan ($cal_0$) D_EC_MOVE_PRESET | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x07 | 0x00 | 0x22 | — |

## 5.5 Command 0x09 ($9_{10}$), Set Auxiliary

### 5.5.1 Command format

| Set Auxiliary D_EC_SET_AUX | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | SUB-OP-CODE | 0x09 | 0x00 | AUX ID | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x0009 SET_AUXILIARY
   2.2 0x0109 SET_INDICATOR
3. This command is used by the Endura/Atlas projects.
4. This command is used and decoded on the:
   ERD97P21, IRD/ERD, Esprit 3012, Intercept and LRD.
5. This[21] op-code was changed. The changes are available in the Spectra IV only.

[20] $Header: 09.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
[21] From an e-mail by Robert Sexton dated: Wednesday, September 26, 2007 4:27 PM with a subject of SP4 LED control.

### 5.5.2 Description

Causes an auxiliary function in the camera unit to be activated.

There are two sub-opcodes for this command:

1. 0x00 D_ECS_SET_AUX_RELAY
2. This sub-opcode is used by the Endura/Atlas projects.
   2.1 AUX ID This is the auxiliary number. The range of this value for relay control is: $1 \rightarrow 8$.
3. 0x01 D_ECS_SET_AUX_LED This is allows control of the Spectra IV's single two color LED via D Protocol.
   3.1 Timed on and off functions are effectively independent. If an ON command is sent with a time, the LED will immediately display the requested color for the time specified. If no timed OFF command is sent it will not repeat. Sending a timed OFF will work in the same way. Sending a timed OFF with an "illegal" color will allow blinking the LED on and off at whatever rate and ON color is desired. An ON or OFF with zero time will stop any timing function.
   3.2 Sending an ON command with a time of 0 will result in having the LED turned on constantly.
   3.3 Sending an OFF command with a time of 0 will result in having the LED turned off constantly.
   3.4 Any combination of other than on/off times of zero, will result in a blinking LED.
   3.5 The colors are 0xFE for green, 0xFD for red and 0xFC for amber. (To get amber, it is necessary to alternate rapidly between red and green but the rate is rather slow and is biased 2:1 for green so it blinks noticeably.) Any other color is treated as off.

| | | |
|---|---|---|
| 0xFE | Green | D_ECD_SET_AUX_LED_GREEN |
| 0xFD | Red | D_ECD_SET_AUX_LED_RED |
| 0xFC | Amber | D_ECD_SET_AUX_LED_AMBER |

   3.5.1. DATA1 contains the "rate" of the toggling in 0.1 second increments (0 = permanent, either ON or OFF. An opcode of OFF, with a rate of non-zero means that the LED will go OFF for the specified duration then back ON for the duration that was initially identified in the previous ON value, see the below example.
   3.5.2. DATA2 contains the affected LED by numeric value. Because LEDs are very much specific to the receiving device, and may vary in number and color, the receiving device identifies which LED or color is LED 1, 2, etc.

## 5.6 Command 0x0B ($11_{10}$), Clear Auxiliary

### 5.6.1 Command format

| Clear Auxiliary D_EC_CLEAR_AUX | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | SUB-OP-CODE | 0x0B | 0x00 | AUX ID | — |

[22] $Header: 0B.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x000B CLEAR_AUXILIARY
   2.2 0x010B CLEAR_INDICATOR
3. This command is used by the Endura/Atlas projects.
4. This command is used and decoded on the:
   ERD97P21, IRD/ERD, Esprit 3012, Intercept and LRD.
5. See SET AUXILIARY (subsection 5.5) for more information about this command.

### 5.6.2 Description

Causes an auxiliary function in the camera unit to be deactivated.

1. CMND1 There are two sub-opcodes for this command:
   1.1 0x00 D_ECS_CLEAR_AUX_RELAY This sub-opcode is used by the Endura/Atlas projects.
   1.2 0x01 D_ECS_CLEAR_AUX_LED
2. DATA2 This is the numerically encoded auxiliary number. The range of this value for relay control is: $1 \rightarrow 8$.
3. DATA2 This is the numerically encoded auxiliary number. The range of this value for LED control is:

| | | |
|---|---|---|
| 0xFE | Green | D_ECD_CLEAR_AUX_LED_GREEN |
| 0xFD | Red | D_ECD_CLEAR_AUX_LED_RED |
| 0xFC | Amber | D_ECD_CLEAR_AUX_LED_AMBER |

## 5.7 Command 0x0D ($13_{10}$), Dummy

### 5.7.1 Command format

| Dummy D_EC_DUMMY_1 | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | ACTION | 0x0D | SUB-DEVICE ID | ON/OFF/-TEMP | — |

FPN

1. This command generates a "General Reply" (subsection 4.2) or nothing. See below.
2. Command Names as used by the Spectra IV software:
   2.1 0x000D DUMMY

[23] $Header: 0D.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$

### 5.7.2 Description

Originally an unused command. Because of the method used in decoding commands in the domes, all command opcodes must be defined. Thus, even though this value is usually unused, it must be defined and decoded.

When ACTION (byte 3) is set to 0x00 this command is decoded and a general response is sent except for the ExSite that gives no response at all. Other than sending a response nothing else occurs in any Pelco equipment.

## 5.8 Command 0x0F ($15_{10}$), Remote Reset

### 5.8.1 Command format

| Remote Reset<br>D_EC_RESET | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x0F | 0x00 | 0x00 | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x000F REMOTE_RESET
3. This command is used and decoded on the: ERD97P21, Esprit 3012, Intercept and LRD.
4. This command resets the system. It will take several seconds before the system is ready to resume normal operation.

### 5.8.2 Description

Causes the dome to be reset. This is the same as turning the dome off and then back on.

## 5.9 Command 0x11 ($17_{10}$), Set Zone Start

### 5.9.1 Command format

| Set Zone Start<br>D_EC_ZONE_START | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x11 | 0x00 | ZONE ID | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x0011 SET_ZONE_START
3. This command is used and decoded on the: Esprit 3012 and Intercept.
4. See ZONE SCAN ON (subsection 5.14) for more information about this command.

[24] $Header: 0F.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
[25] $Header: 11.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$

### 5.9.2 Description

This command is used to define the start point of up to eight zones (ZONE ID). Zones are defined by setting a start point and then moving the dome in a clockwise direction (looking down from above the dome or pan/tilt when it is installed in its normal position) to set an end point. Higher numbered zones override lower numbered zones. Zone numbers are in byte 6 (ZONE ID) and are 1 based, not 0 based. I.e. 0x03 = Zone 3.

When the unit has a password enabled, this command will cause the unit to display the password request screen and not proceed until a correct password is entered. There is no way to directly enter a password via a Protocol.

FPN

### 5.9.3 Zones

When a SET ZONE START command (subsection 5.9) is received, the current pan position is saved as the start position for the zone number specified in the command. If the zone was previously defined the end position is invalidated and the new start position is set. Also whatever is displayed on the Zone label (i.e. columns 0 → 19 of write character to screen command 0x15, $21_{10}$ (subsection 5.11) though not displayed directly on some systems) on the logical first video line is saved as the zone label. When a SET ZONE END command (subsection 5.10) is received, the monitor is unlocked How was the monitor locked?[-DGS], and the current pan position is saved as the end position for the zone number specified (ZONE ID) in the command.

Zones extend from the start point clockwise to the end point. This means that if a zone start point is set, the Spectra is panned slightly clockwise, and the zone end point is set, the zone will be small. But if the Spectra is panned slightly counterclockwise between the start and end points, the zone will be almost all the way around the pan circle.

A zone may not be specified that is 360° in size. Starting with the Spectra III the end and start points of the same zone may not be within 1° of each other. On earlier models the start/end points had to be at least 10° apart.

There are commands to turn SET ZONE SCAN ON (subsection 5.14) and SET ZONE SCAN OFF (subsection 5.15). If zone scan has been turned on, during normal pan/tilt operation the current pan position is continuously read. If the current position is within a zone, the label for that zone is displayed on the logical first video line. If the current position is not within any zone, the line is cleared. If the current position is within more than one zone, the label for the highest-numbered zone will be displayed.

**Note**

For units previous to Spectra III SET ZONE SCAN OFF (subsection 5.15) must be sent off before this command is received or the zone programming will not work correctly.

Spectra III and Spectra IV does not require Zone Scan to be off to operate properly.

## 5.10 Command 0x13 ($19_{10}$), Set Zone End

### 5.10.1 Command format

| Set Zone End<br>D_EC_ZONE_END | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x13 | 0x00 | ZONE ID | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:

[26] $Header: 13.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$

2.1 0x0013 SET_ZONE_END

3. This command is used and decoded on the: Esprit 3012 and Intercept.
4. See SET ZONE SCAN START (subsection 5.9) and SET ZONE SCAN ON (subsection 5.14) for more information about this command.

### 5.10.2 Description

This is used to define the end point of up to eight zones. Zones are defined by setting a start point and then moving the dome in a clockwise direction (looking down from above the dome or pan/tilt when it is installed in its normal position) to set an end point. Higher numbered zones override lower numbered zones. Zone numbers are in byte 6 and are 1 based, not 0 based. I.e. 0x04 = Zone 4. See the write up in SET ZONE START (subsection 5.9) for more details on setting zones.

When the unit has a password enabled, this command will cause the unit to display the password request screen and not proceed until a correct password is entered. There is no way to enter a password via a Protocol.

## 5.11 Command 0x15 ($21_{10}$), Write Character To Screen

### 5.11.1 Command format

| Write Character To Screen D_EC_WRITE_CHAR | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | DATA TYPE | 0x15 | SCREEN COLUMN | ASCII CHAR | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:

   2.1 0x0015 WRITE_CHARACTER_TO_SCREEN
3. This command is used and decoded on the: Esprit 3012, Intercept and on the Everest/Atlas projects.
4. There is no protocol command to read out this data via a serial link.

### 5.11.2 Description

1. DATA TYPE is used to indicate the use of this byte.

   1.1 0x00 (D_ECS_WRITE_CHAR_LABEL) is used to indicate that the data is to be displayed on the screen in the label buffer.

   There are several places/data sub-types for the characters to be displayed:

   1.1.1. The parameter in DATA1 (byte 5), SCREEN COLUMN of this command indicates the column to write to. This parameter is interpreted as follows:

   1.1.1.1. Columns 0 → 19 are used to receive ZONE LABELS.

   1.1.1.2. Columns 20 → 39 are used to receive PRESET LABELS.

   1.1.1.3. Columns 40 → 59 are used to receive camera labels. (Unimplemented, reserved for future use.)

[27] $Header: 15.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$

1.1.1.4. Columns 60 → 79 are used to receive the unit's serial number. (Esprit TI (FLIR) units only.)

Starting with Spectra III (and ExSite):

1.1.1. Characters written to these positions are not written directly to the screen. Once the SET ZONE START (opcode 0x11) command is received, the characters are displayed.

1.1.2. If characters are written to these columns and no SET PRESET (opcode subsection 5.2) command is received within 250 milliseconds of receipt of the last character, the characters will be displayed on the screen beginning at the first column of the second row of the display.

With the Spectra IV, following the first command on after a power cycle, characters in the Preset label locations are not displayed except when followed by a SET PRESET (subsection 5.2) command within 250 milliseconds.

1.2 DATA2, ASCII CHAR, is the character to be displayed. The exact representation on the screen will be controlled by the current font that is selected.

### 5.11.3 Sending a serial number to a unit

From an e-mail from:

```
From: Hannen, Craig
Sent: Monday, December 10, 2007 4:51 PM
Subject: FLIR Serialization Protocol
```

Looking at the serialization app I wrote, there's a few steps involved in serializing a FLIR PTZ. The serial number is sent with character designators 60 to 79 (much like a zone label is sent with designator's 0 to 19 and a preset label is sent with designator's 20 to 39). The serial number is sent as if it is a label.

These are the steps involved:

1. Using the app GUI the user will enter the serial number into the only textbox in the main window. Following this the serialization process is started via a button click.

2. The app will first auto-connect to the PTZ at 2400, 4800 or 9600 baud. To AutoBaud and get the Spectra's address, the app sends the D Protocol query command (subsection 5.35) at each bit rate twice before giving up. If none of the queries get a reply, assume there's no PTZ connected and quit the serialization process.

3. Once the app is connected, start sending the serial number one character at a time (using the WRITE CHARACTER TO SCREEN command (subsection 5.11)). For each character command, make sure the PTZ replies to each command and the checksum is correct.

4. When all characters of the serial number have been sent, send the NULL character or 0x00 (once again using the WRITE CHARACTER TO SCREEN command) to indicate to the PTZ the complete serial number has been sent. Notice that we don't send any blank/space characters at the end of the serial number. It's necessary to send spaces at the end of preset/zone labels to write over a previous label, but this is unnecessary in this serial number application.

5. To make certain that the PTZ received the entire string accurately, send the checksum of all of the characters in the string as the last character (this is the sum of all character bytes in the string modulo 256). Once again this checksum is sent via the WRITE CHARACTER TO SCREEN command (subsection 5.11).

The characters allowed to be used in a serial number are 0-9, a-z, A-Z and the dash character (ASCII 0x2D). The GUI prevents the serial number from being longer than 18 characters.

## 5.12 Command 0x17 ($23_{10}$), Clear Screen

### 5.12.1 Command format

| Clear Screen D_EC_CLEAR_SCREEN | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x17 | 0x00 | 0x00 | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x0017 CLEAR_SCREEN
3. This command is used by the Endura/Atlas projects.
4. This command is used and decoded on the: Esprit 3012 and Intercept.

### 5.12.2 Description

Clears all generated characters from the screen.
Does not clear following on-screen data that is displayed with a "CONSTANT" attribute:

1. Active Alarms
2. Azimuth
3. Elevation
4. Direction, Spectra III and following.
5. Zoom Level, Spectra III and following.
6. Date/Time, Spectra IV only.

## 5.13 Command 0x19 ($25_{10}$), Alarm Acknowledge

### 5.13.1 Command format

| Alarm Acknowledge D_EC_ALARM_ACK | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | TYPE | 0x19 | 0x00 | 0x00 | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x0019 ALARM_ACKNOWLEDGE
3. This command, with a TYPE SUB OP-CODE of 0x01, is used by the Endura/Atlas projects.
4. This opcode is used by the ERD97P21.

[28] $Header: 17.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
[29] $Header: 19.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$

### 5.13.2 Description

Usage of the TYPE SUB OPCODE:

1. 0x00: When an alarm is enabled (through a SET 95 menu action) and it occurs, the dome will perform whatever action it is programmed to do (again through a SET 95 menu action), until the alarm is acknowledged. Alarm numbers are in byte 6 and are 1 based not 0 based. Sending an ALARM ACKNOWLEDGE when no alarm is active is harmless.

   The ERD97 has Alarms in the range of 1 to 8. When the alarm number is sent over the communications line the range is 2 to 9. I.e. it is TWO based.

2. 0x01:[30] The response will be the General Response of 4 bytes which includes the Alarm Status data

## 5.14 Command 0x1B ($27_{10}$), Zone Scan On

### 5.14.1 Command format

| Zone Scan On<br>D_EC_ZONE_ON | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x1B | 0x00 | 0x00 | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x001B ZONE_SCAN_ON
3. This command is used and decoded on the: Esprit 3012 and Intercept.
4. See SET ZONE START (subsection 5.9) for more information about this command.

### 5.14.2 Description

Used to turn the displaying of zone labels on.

If zone scan has been turned on, the current pan position is continuously read. If the current position is within a zone, the label for that zone is displayed on the first video line. If the current position is not within any zone, the line is cleared. If the current position is within more than one zone, the label/blanking for the highest-numbered zone will be displayed. For more information about labels, see subsubsection 3.3.1.

FPN

Using[32] the word "Scan" to describe these commands (subsection 5.14 and subsection 5.15) implies saying that the PTZ is expected to scan on its own. These commands simply enable/disable the zone label display. In earlier PTZ's (pre-Spectra III), these commands would enable/disable zone labels without any extra setup in the menus. Spectra III started this 2-tiered idea of enabling zone labels in the menus before they can be enabled with the zone scan on/off commands. A more sensible naming for these commands would be "Zone Labels On" and "Zone Labels Off".

This opcode is used by the following Pelco equipment:

Esprit, Spectra, ExSite

[30] This information comes from an e-mail from Alfio Marrone, dated Friday, January 23, 2004 12:54 PM.
[31] $Header: 1B.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
[32] From: Hannen, Craig. Tuesday, April 18, 2006 10:11 AM

### 5.14.3 Zone Blanking

Zone blanking is used to blank out a defined zone. It is set up in the set 95 menu system and is the only way to blank out an area in the Spectra II and Esprit type systems. It is usable on all of the newer systems by using their set 95 menus.

### 5.14.4 Creating Zone Labels

Many devices have an ability to display labels on the video. Labels that identify the preset or zone being scanned are common. There is a special technique to establish a label that is associated with either a preset or a zone. First, send the label to the receiver/driver using the "Write Character to Screen" (subsection 5.11) command. After the label is on the screen, then set the preset or zone. That will establish the label and associate it with that preset or zone.

- ZONE SCAN ON (subsection 5.14): Used to turn the display of zone labels on.
- ZONE SCAN OFF (subsection 5.15): Used to turn the display of zone labels off.

For the Spectra III, Spectra III SE and ExSite, characters written to these zone addresses positions are not written directly to the screen. Once the SET ZONE START (subsection 5.9) command is received, the characters are displayed.

#### Defining a Zone

1. Send "SET ZONE SCAN OFF" (subsection 5.15)
2. Send characters to positions 0 $\longrightarrow$ 19.
3. Send a "SET ZONE START" (subsection 5.9)
4. Do any required pan motion. On newer units presets may be called.
5. Send a "SET ZONE END" (subsection 5.10)
6. Send a "SET ZONE SCAN ON" (subsection 5.14)

Now when the camera is in one of the eight zones, the zone message will appear. The internal logic of the software, copies what is in the zone label buffer into EEPROM when it gets a "SET ZONE END" (subsection 5.10) command. With the older units, all units prior to the Spectra III, the zone label was always on the first displayed line. With the Spectra III the zone label can be anywhere on the screen and the logic to move it is internal to the Spectra III software.

## 5.15 Command 0x1D ($29_{10}$), Zone Scan Off

### 5.15.1 Command format

| Zone Scan Off<br>D_EC_ZONE_OFF | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x1D | 0x00 | 0x00 | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x001D ZONE_SCAN_OFF
3. See subsection 5.14 for more information about this command.
4. This command is used and decoded on the: Esprit 3012 and Intercept.

[33] $Header: 1D.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$

### 5.15.2 Description

Used to turn the displaying of zone labels off.

## 5.16 Command 0x1F ($31_{10}$), Record Pattern Start

### 5.16.1 Command format

| Set Pattern Start<br>D_EC_START_RECORD | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x1F | 0x00 | PATTERN ID | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x001F SET_PATTERN_START
3. This command is used by the Endura/Atlas projects.
4. This command is used and decoded on the: Esprit 3012 and Intercept.
5. The defined patterns may be determined by using QUERY DEFINED PATTERNS (subsubsection 5.59.19) and QUERY DEFINED PATTERNS RESPNSE (subsubsection 5.59.20).
6. A pattern may be deleted by using DELETE PATTERN (subsubsection 5.59.9).

### 5.16.2 Description

Recording of a pattern starts when a start pattern record command is received. It ends when either an end pattern record command is received, or 60 seconds has elapsed since the start pattern record command was received. The message "PROGRAMMING PATTERN" is put on the first video line. When recording stops (either by command, timeout or when pattern memory fills up on the newer types of PTZ units[35]), the message is cleared.

FPN

Playing of the recorded pattern starts when a start pattern play command is received. When the end of the recorded pattern is reached, playback starts over again at the beginning of the pattern. This continues until any other command is received. If zones have not been enabled, the message "RUNNING PATTERN" is put on the first video line and remains there until playback stops, at which point it is cleared. If zones have been enabled, the "RUNNING PATTERN" message will not be shown. Instead, the zone labels will be shown as the pattern moves through the zones. When playback is stopped, the first video line is cleared.

There are two types of pattern processing: record and playback. Pattern processing occurs once each timer tick (14 times a second[36]). If recording, the current command is stored in the EEPROM/flash. If the current command is not one that can be played back, an illegal command is saved (it will be skipped during playback). If playing back, a command is read out of the EEPROM/flash and decoded. If it is an illegal command, it is skipped.

When the unit has a password enabled, this command will cause the unit to display the password request screen and not proceed until a correct password is entered. There is no way to enter a password via a Protocol.

[34] $Header: 1F.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$

[35] The newer unit types are: Spectra III, Mini Spectra, ExSite and Spectra IV.

[36] Or depending on the model it may be each vertical interrupt (60 or 50 times a second depending on whether the video is NTSC or PAL).

The parameter in byte 6 of this command indicates the pattern number. Used when starting to record a pattern. Spectra III, ExSite and Spectra IV interpret byte 6 as follows:

| PATTERN ID | |
|---|---|
| Value | Action |
| 0 or 1 | Sets/runs pattern 1 |
| 2 | Sets/runs pattern 2 |
| 3 | Sets/runs pattern 3 |
| 4 | Sets/runs pattern 4 |
| 5 | Sets/runs pattern 5 |
| 6 | Sets/runs pattern 6 |
| 7 | Sets/runs pattern 7 |
| 8 | Sets/runs pattern 8 |

Spectra I, Spectra II and Esprit interpret byte 6 as follows:

| PATTERN ID | |
|---|---|
| Value | Action |
| 0 | Sets/runs the single “long pattern” |
| 1 | Sets/runs the first “short pattern” |
| 2 | Sets/runs the second “short pattern” |

FPN

### 5.16.3 Pattern numbers

The method of saving a pattern has changed over time. The original method was to save what the unit was doing every vertical sync time. This resulted in getting 60 (50 with PAL based cameras) records a second. Older units had patterns defined in maximum minutes of run time. With these systems the choice is one one minute pattern (pattern 0) or two one half minute patterns (patterns 1 and 2).

Starting with the Spectra III/ExSite series of units. Saving what the unit was doing each vertical sync time is used, however if the unit is doing the same thing on several vertical sync times in a row, then an eight bit counter is incremented and saved when a change is detected. This Run Length Limited (RLL) technique makes it so that the saved pattern length may be of greater duration. The total duration is now unknown in advance as it depends on the number of **changes** in what the unit is told to do while recording the pattern. (A patent is pending on this method of saving a pattern.)

The numbering and quantity of patterns varies on different units:

1. On many Intercept type units there was one pattern. On other Intercept units, those with out presets, there were no patterns.
2. The Spectra I, Spectra II and Esprit have pattern numbers from 0 $\longrightarrow$ 2.
3. On the Spectra II Lite the only pattern is numbered 0.
4. On the Spectra III the only pattern is numbered 1.
5. With the Spectra III SE and ExSite patterns are numbered in the range of 1 $\longrightarrow$ 4.
6. On the Spectra IV SE the patterns are numbered in the range of 1 $\longrightarrow$ 8.
7. With older low cost Spectra units utilizing the reduced capability types of cameras, there is only one pattern.
8. With low cost Spectra IV units utilizing the reduced capability types of cameras, there are four patterns.

## 5.17 Command 0x21 ($33_{10}$), Record Pattern End

### 5.17.1 Command format

| Set Pattern End<br>D_EC_END_RECORD | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x21 | 0x00 | PATTERN ID | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x0021 SET_PATTERN_STOP
3. This command is used by the Endura/Atlas projects.
4. This command is used and decoded on the: Esprit 3012 and Intercept.
5. See RECORD PATTERN START subsection 5.16 for more information about this command.

### 5.17.2 Description

Stops the current pattern to stop being recorded and have its data written into persistent memory.

## 5.18 Command 0x23 ($35_{10}$), Run Pattern

### 5.18.1 Command format

| Run Pattern<br>D_EC_START_PLAY | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x23 | 0x00 | PATTERN ID | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x0023 RUN_PATTERN
3. This command is used by the Endura/Atlas projects.
4. This command is used and decoded on the: Esprit 3012 and Intercept.
5. The defined patternss may be determined by using QUERY DEFINED PATTERNS (subsubsection 5.59.19) and its response of
   QUERY DEFINED PATTERNS RESPONSE (subsubsection 5.59.20).
6. See RECORD PATTERN START subsection 5.16 for more information about this command.

[37] $Header: 21.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
[38] $Header: 23.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$

### 5.18.2 Description

The parameter in byte 6 of this command indicates the pattern number. Used to get a pattern running. Spectra III, EsSite and Spectra IV interpret byte 6 as follows:

| PATTERN ID | |
|---|---|
| Value | Action |
| 0 or 1 | Sets/runs pattern 1 |
| 2 | Sets/runs pattern 2 |
| 3 | Sets/runs pattern 3 |
| 4 | Sets/runs pattern 4 |
| 5 | Sets/runs pattern 5 |
| 6 | Sets/runs pattern 6 |
| 7 | Sets/runs pattern 7 |
| 8 | Sets/runs pattern 8 |

Spectra I, Spectra II and Esprit interpret byte 6 as follows:

| PATTERN ID | |
|---|---|
| Value | Action |
| 0 | Sets/runs the single “long pattern” |
| 1 | Sets/runs the first “short pattern” |
| 2 | Sets/runs the second “short pattern” |

## 5.19 Command 0x25 ($37_{10}$), Set Zoom Speed

### 5.19.1 Command format

| **Set Zoom Speed D_EC_ZOOM_SPEED** | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x25 | 0x00 | ZOOM SPEED | — |

FPN

1. This command generates a “General Reply” (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x0025 SET_ZOOM_SPEED
3. This opcode is are used by the: ERD97P21.
4. This command is used and decoded on the: Intercept.
5. The current value of the zoom speed may not be read out.

[39] $Header: 25.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$

### 5.19.2 Description

This command accepts values of 0 through 3, in byte 6, to change the speed of the indicated function. Some cameras (such as the X12 camera in the Spectra I) do not support these functions. If the camera does support this function, the command is ignored. 0 is the slowest speed, 3 is the fastest. Because of the vast number of variations in camera capabilities, it is usually best to use the SET 95 menu for this type of control change.

| ZOOM SPEED | | |
|---|---|---|
| Value | Use | |
| 0 | Slowest Speed | D_ECD_ZOOM_SPEED_SLOW |
| 1 | Low Medium Speed | D_ECD_ZOOM_SPEED_MEDIUM |
| 2 | High Medium Speed | D_ECD_ZOOM_SPEED_FAST |
| 3 | Highest speed | D_ECD_ZOOM_SPEED_FASTEST |

**Note:** In the reduced capability type of Spectra IV units Zoom speed may not be changed.

## 5.20 Command 0x27 ($39_{10}$), Set Focus Speed

### 5.20.1 Command format

| **Set Focus Speed D_EC_FOCUS_SPEED** | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x27 | 0x00 | FOCUS SPEED | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x0027 SET_FOCUS_SPEED
3. This opcode is used by the: ERD97P21 and the Intercept.
4. The current value of the focus speed may not be read out.

### 5.20.2 Description

This command accepts values of 0 through 3, in byte 6, to change the speed of the indicated function. Some cameras (such as the X12 camera in the Spectra I) do not support these functions. If the camera does support this function, the command is ignored. 0 is the slowest speed, 3 is the fastest. Because of the vast number of variations in camera capabilities, it is usually best to use the SET 95 menu for this type of control change.

| FOCUS SPEED | | |
|---|---|---|
| Value | Use | |
| 0 | Slowest Speed | D_ECD_FOCUS_SPEED_SLOW |
| 1 | Low Medium Speed | D_ECD_FOCUS_SPEED_MEDIUM |
| 2 | High Medium Speed | D_ECD_FOCUS_SPEED_FAST |
| 3 | Highest speed | D_ECD_FOCUS_SPEED_FASTEST |

[40] $Header: 27.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$

## 5.21 Command 0x29 ($41_{10}$), Reset Camera to Defaults

### 5.21.1 Command format

| Reset camera to defaults D_EC_CAMERA_RESET | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x29 | 0x00 | 0x00 | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x0029 RESET_CAMERA_DEFAULTS
3. The Spectra IV does not acknowledge a CAMERA RESET command. It just does the reset.

### 5.21.2 Description

Resets the camera to its default condition, except that the current phase delay is not changed.

## 5.22 Command 0x2B ($43_{10}$), Auto Focus

### 5.22.1 Command format

| Auto focus D_EC_AUTO_FOCUS | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x2B | 0x00 | AUTO FOCUS CTRL | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x002B AUTO_FOCUS_MODE
3. This command is used by the Endura/Atlas projects.
4. The current value of auto focus may not be read out.

### 5.22.2 Description

If byte 6 is 0 the device automatically controls whether auto focus is on (default) or off. If byte 6 is 1, auto focus is turned off. Other values are ignored.

| AUTO FOCUS CTRL | | |
|---|---|---|
| Value | Use | |
| 0 | Automatic operation (default) | D_ECD_AUTO_FOCUS_AUTO |
| 1 | Auto focus is off | D_ECD_AUTO_FOCUS_OFF |

[41] $Header: 29.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
[42] $Header: 2B.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$

## 5.23 Command 0x2D ($45_{10}$), Auto Iris

### 5.23.1 Command format

| Auto iris<br>D_EC_AUTO_IRIS | | | | | | |
|---|---|---|---|---|---|---|
| 1<br>SYNC | 2<br>ADDR | 3<br>CMND1 | 4<br>CMND2 | 5<br>DATA1 | 6<br>DATA2 | 7<br>CKSM |
| 0xFF | — | 0x00 | 0x2D | 0x00 | AUTO IRIS CTRL | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x002D AUTO_IRIS_MODE
3. This command is used by the Endura/Atlas projects.
4. The current value of auto iris may not be read out.

### 5.23.2 Description

If byte 6 is 0 the device automatically controls whether auto iris is on (default) or off. If byte 6 is 1, auto iris is turned off. Other values are ignored.

| AUTO IRIS CTRL | | |
|---|---|---|
| Value | Use | |
| 0 | Automatic operation (default) | D_ECD_AUTO_IRIS_AUTO |
| 1 | Auto iris is off | D_ECD_AUTO_IRIS_OFF |

## 5.24 Command 0x2F ($47_{10}$), AGC

**Discontinued Command**

In July 2004 the use of this command was discontinued as a protocol item. This function is only supported via the menu system.

This command does not apply to the Esprit TI.

### 5.24.1 Command format

| AGC<br>D_EC_AGC | | | | | | |
|---|---|---|---|---|---|---|
| 1<br>SYNC | 2<br>ADDR | 3<br>CMND1 | 4<br>CMND2 | 5<br>DATA1 | 6<br>DATA2 | 7<br>CKSM |
| 0xFF | — | 0x00 | 0x2F | 0x00 | AGC CTRL | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x002F AGC_MODE
3. The current value of AGC may not be read out.

[43] $Header: 2D.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
[44] $Header: 2F.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$

### 5.24.2 Description

If byte 6 is 0 the device automatically controls whether AGC (automatic gain control) is on or off (default), If byte 6 is 1, AGC is turned off (manual gain). Other values are ignored. Sending an ADJUST GAIN command (subsection 5.32) turns AGC off.

| AGC CTRL | | |
|---|---|---|
| Value | Use | |
| 0 | Automatic operation (default) | D_ECD_AUTO_AGC_AUTO |
| 1 | AGC is off | D_ECD_AUTO_AGC_OFF |

## 5.25 Command 0x31 ($49_{10}$), Backlight Compensation

### 5.25.1 Command format

| Backlight compensation D_EC_BLC | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x31 | 0x00 | BLC CTRL | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x0031 BACKLIGHT_COMPENSATION
3. The current value of BLC may not be read out.

### 5.25.2 Description

If byte 6 is 1, backlight compensation is turned off (default). If byte 6 is 2, backlight compensation is turned on. Other values are ignored.

| BLC CTRL | | |
|---|---|---|
| Value | Use | |
| 1 | BLC is off (default) | D_ECD_AUTO_BLC_OFF |
| 2 | BLC is on | D_ECD_AUTO_BLC_ON |

**Change for Spectra IV** If byte 6 is 0, backlight compensation is turned off (default). If byte 6 is non-zero, backlight compensation is turned on.

| BLC CTRL | |
|---|---|
| Value | Use |
| 0 | BLC is off (default) |
| 1 | BLC is on |

[45] $Header: 31.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$

## 5.26 Command 0x33 ($51_{10}$), Auto White Balance

### 5.26.1 Command format

| Auto white balance D_EC_AWB | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x33 | 0x00 | AWB CTRL | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x0033 AUTO_WHITE_BALANCE
3. The current value of AWB may not be read out.

### 5.26.2 Description

If byte 6 is 1, auto white balance is turned on (default). If byte 6 is 2, auto white balance is turned off. Other values are ignored. Sending an ADJUST WHITE BALANCE command turns auto white balance off. (subsection 5.30, subsection 5.31)

| AWB CTRL | | |
|---|---|---|
| Value | Use | |
| 1 | AWB is on (default) | D_ECD_AUTO_AWB_ON |
| 2 | AWB is off | D_ECD_AUTO_AWB_OFF |

**Change for Spectra IV** If byte 6 is 0, auto white balance is turned on (default). If byte 6 is non-zero, auto white balance is turned off. Sending an ADJUST WHITE BALANCE command turns auto white balance off. (subsection 5.30, subsection 5.31)

| AWB CTRL | |
|---|---|
| Value | Use |
| 0 | AWB is on (default) |
| 1 | AWB is off |

## 5.27 Command 0x35 ($53_{10}$), Enable Device Phase Delay Mode

**Discontinued Command**

In July 2004 the use of this command was discontinued as a protocol item. This function is only supported via the menu system.

This command does not apply to the Esprit TI.

[46] $Header: 33.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
[47] $Header: 35.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

### 5.27.1 Command format

| Enable device phase delay mode D_EC_DEVICE_PHASE | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x35 | 0x00 | 0x00 | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x0035 ENABLE_DEVICE_PHASE_DELAY_MODE

### 5.27.2 Description

When device phase delay is set, the phase delay is set by the device (there may be a manual adjustment). Sending an ADJUST LINE LOCK phase delay command will disable device phase delay mode.

See subsection 5.29.

## 5.28 Command 0x37 ($55_{10}$), Set Shutter Speed

### 5.28.1 Command format

| Set shutter speed D_EC_SHUTTER_SPEED | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x37 | SHUTTER CTRL 1 | SHUTTER CTRL 2 | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x0037 SET_SHUTTER_SPEED

### 5.28.2 Older Shutter Speed Codes

**Spectra II and older Shutter Speed Codes** Byte[49] 5 and byte 6 are the high and low bytes respectively of 1 divided by the shutter speed. The shutter speed is limited internally to the range from $\frac{1}{60}$ second (NTSC) or $\frac{1}{50}$ second (PAL) to $\frac{1}{30000}$ second, corresponding to a sent number range from 60 (or 50) to 30000. If the sent number is, 0 the shutter speed is reset to its default value ($\frac{1}{60}$ or $\frac{1}{50}$ second). If the sent number is 1, the shutter speed is moved to the next faster speed in the shutter speed table (below). If the sent number is 2, the shutter speed is set to the next slower speed in the table. Because of the vast number of variations in camera capabilities, it is usually best to use the SET 95 menu for this type of control change.

[48] $Header: 37.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
[49] Up until Spectra II rev 3.21.

| SHUTTER CTRL | |
|---|---|
| Byte 5,6 | Speed (seconds) |
| 0 | Use default of $\frac{1}{60}$ or $\frac{1}{50}$ |
| 1 | Increment in table |
| 2 | Decrement in table |
| 50 | 50 (PAL) |
| 60 | 60 (NTSC) |
| 100 | 100 |
| 120 | 120 |
| 250 | 250 |
| 500 | 500 |
| 1000 | 1000 |
| 2000 | 2000 |
| 4000 | 4000 |
| 10000 | 10000 |
| 30000 | 30000 |

### 5.28.3 Newer Shutter Speed Codes

With[50] the Spectra III series of units this has been changed as follows: Byte 6 is the only byte processed. Byte 6 is an index into an array of usable shutter speeds. An index of `0x00` represents auto-shutter, and indexes from 1 $\longrightarrow$ 16 represent the supported shutter speeds. The actual shutter speed values vary depending on the exact make and rev of camera installed. Because of the vast number of variations in camera capabilities, it is usually best to use the SET 95 menu for this type of control change.

**Spectra III, Spectra IV, ExSite and Esprit IOP/IOC Shutter Speed Codes** All times in the below list are the denominator of a fraction with 1 being the numerator. I.e. using an index of 9 generates a shutter speed of $\frac{1}{250}$ second.

**Spectra MINI Shutter Speed Codes** All times in the below list are the denominator of a fraction with 1 being the numerator. I.e. using an index of 6 generates a shutter speed of $\frac{1}{500}$ second. The 1/100 shutter speed was added in for the Esprit series starting with rev 3.30.

[50] Starting with Spectra II, rev 3.22, and all Spectras since.

| | Spectra III and newer | | MINI Spectra | |
|---|---|---|---|---|
| | SHUTTER CTRL | | SHUTTER CTRL | |
| Index | NTSC | PAL | NTSC | PAL |
| 0 | Auto Shutter | Auto Shutter | Auto Shutter | Auto Shutter |
| 1 | 2 | 1.5 | 60 | 50 |
| 2 | 4 | 3 | 100 | 100 |
| 3 | 8 | 6 | 120 | 100 |
| 4 | 15 | 12 | 180 | 150 |
| 5 | 30 | 25 | 250 | 250 |
| 6 | 60 | 50 | 500 | 500 |
| 7 | 120 | 100 | 1000 | 1000 |
| 8 | 180 | 150 | 2000 | 2000 |
| 9 | 250 | 250 | 4000 | 4000 |
| 10 | 500 | 500 | 10000 | 10000 |
| 11 | 1000 | 1000 | 30000 | 30000 |
| 12 | 2000 | 2000 | — | — |
| 13 | 4000 | 4000 | — | — |
| 14 | 10000 | 10000 | — | — |
| 15 | 30000 | 30000 | — | — |
| 16 | 100 | reserved | — | — |

## 5.29 Command 0x39 ($57_{10}$), Adjust Line Lock Phase Delay

**Discontinued Command**

In July 2004 the use of this command was discontinued as a protocol item. This function is only supported via the menu system.

This command does not apply to the Esprit TI.

### 5.29.1 Command format

| Adjust line lock phase delay D_EC_ADJUST_PHASE | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | LL CTRL | 0x39 | LL DELAY MSB | LL DELAY LSB | — |

FPN

1. This command generates a “General Reply” (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x0039 ADJUST_LINE_LOCK_PHASE_DELAY
   2.2 0x0139 SET_LINE_LOCK_PHASE_DELAY

[51] $Header: 39.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

### 5.29.2 Description

If byte 3 is 0, byte 5 and byte 6 are the high and low bytes respectively of an unsigned 16-bit number which is the new phase delay. If byte 3 is 1, byte 5 and byte 6 are the high and low bytes respectively of a signed 16-bit number which is the change to the current phase delay. The phase delay is the delay between the zero crossing of the AC power waveform and the line lock signal sent to the camera. It is in units of 1.085 microseconds. The phase delay is limited internally to the range from 0 to 32767. If an attempt is made to set or change the delay to a value outside this range, the delay will be set to the appropriate end of the range. This command disables device phase delay mode (ENABLE PHASE DELAY MODE subsection 5.27).

| LL CTRL/DELAY | | | |
|---|---|---|---|
| Byte | Bytes | — | |
| 3 | 5, 6 | Use | |
| 0 | 16-bit unsigned value | New phase delay value | D_ECS_ADJUST_PHASE_NEW |
| 1 | 16-bit signed value | Phase delay change | D_ECS_ADJUST_PHASE_DELTA |

## 5.30 Command 0x3B ($59_{10}$), Adjust White Balance (R-B)

### 5.30.1 Command format

| Adjust white balance (R-B) D_EC_ADJUST_RB_WB | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | WB CTRL | 0x3B | WB-RB MSB | WB-RB LSB | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x003B ADJUST_WHITE_BALANCE_RB
   2.2 0x013B SET_WHITE_BALANCE_RB

### 5.30.2 Description

If byte 3 is 0, byte 5 and byte 6 are the high and low bytes respectively of an unsigned 16-bit number which is the new red-blue white balance value. If byte 3 is 1, byte 5 and byte 6 are the high and low bytes respectively of a signed 16-bit number which is the change to the current red-blue white balance value. The balance value is limited internally to the range from 192 (0xC0) to 768 (0x300). If an attempt is made to set or change the balance to a value outside this range, the balance will be set to the appropriate end of the range. This command turns off AUTO WHITE BALANCE (subsection 5.26).

[52] $Header: 3B.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

| CTRL/WB-RB VALUE | | | |
|---|---|---|---|
| Byte | Bytes | — | |
| 3 | 5, 6 | Use | |
| 0 | 16-bit unsigned value | New red-blue white balance value | D_ECS_ADJUST_RB_WB_NEW |
| 1 | 16-bit signed value | Red-blue white balance change | D_ECS_ADJUST_RB_WB_DELTA |

## 5.31 Command 0x3D ($61_{10}$), Adjust White Balance (M-G)

### 5.31.1 Command format

| Adjust white balance (M-G) D_EC_ADJUST_MG_WB | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | WB CTRL | 0x3D | WB-MG MSB | WB-MG LSB | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x003D ADJUST_WHITE_BALANCE_MG
   2.2 0x013D SET_WHITE_BALANCE_MG

### 5.31.2 Description

If byte 3 is 0, byte 5 and byte 6 are the high and low bytes respectively of an unsigned 16-bit number which is the new magenta-green white balance value. If byte 3 is 1, byte 5 and byte 6 are the high and low bytes respectively of a signed 16-bit number which is the change to the current magenta-green white balance value. The balance value is limited internally to the range from 192 (0xC0) to 768 (0x300). If an attempt is made to set or change the balance to a value outside this range, the balance will be set to the appropriate end of the range. This command turns off AUTO WHITE BALANCE (subsection 5.26).

| WB CTRL/WB-RB VALUE | | | |
|---|---|---|---|
| Byte | Bytes | — | |
| 3 | 5, 6 | Use | |
| 0 | 16-bit unsigned value | New Magenta-green white balance phase delay value | D_ECS_ADJUST_MG_WB_NEW |
| 1 | 16-bit signed value | Magenta-green white balance change | D_ECS_ADJUST_MG_WB_DELTA |

## 5.32 Command 0x3F ($63_{10}$), Adjust Gain

**Discontinued Command**

[53] $Header: 3D.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
[54] $Header: 3F.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

In July 2004 the use of this command was discontinued as a protocol item. This function is only supported via the menu system.

This command does not apply to the Esprit TI.

### 5.32.1 Command format

| Adjust gain D_EC_ADJUST_GAIN | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | GAIN CTRL | 0x3F | GAIN MSB | GAIN LSB | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x003F ADJUST_GAIN
   2.2 0x013F SET_GAIN

### 5.32.2 Description

If byte 3 is 0, byte 5 and byte 6 are the high and low bytes respectively of an unsigned 16-bit number which is the new gain. If byte 3 is 1, byte 5 and byte 6 are the high and low bytes respectively of a signed 16-bit number which is the change to the current gain. The gain is limited internally to the range from 0 to 448 (0x1C0). If an attempt is made to set or change the gain to a value outside this range, the gain will be set to the appropriate end of the range.

| GAIN CTRL/GAIN VALUE | | | |
|---|---|---|---|
| Byte | Bytes | — | |
| 3 | 5, 6 | Use | |
| 0 | 16-bit unsigned value | New gain value | D_ECS_ADJUST_GAIN_NEW |
| 1 | 16-bit signed value | Gain change | D_ECS_ADJUST_GAIN_DELTA |

## 5.33 Command 0x41 ($65_{10}$), Adjust Auto-Iris Level

### 5.33.1 Command format

| Adjust auto-iris level D_EC_ADJUST_AI_LEVEL | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | AI CTRL | 0x41 | AIL MSB | AIL LSB | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:

[55] $Header: 41.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

2.1 0x0041 ADJUST_AUTO_IRIS_LEVEL

2.2 0x0141 SET_AUTO_IRIS_LEVEL

3. This command is used by the Endura/Atlas projects.

### 5.33.2 Description

If byte 3 is 0, byte 5 and byte 6 are the high and low bytes respectively of an unsigned 16-bit number which the new level. If byte 3 is 1, byte 5 and byte 6 are the high and low bytes respectively of a signed 16-bit number which is the change to the current level. The level is limited internally to the range from 0 to 255 (0xFF). If an attempt is made to set or change the level to a value outside this range, the gain will be set to the appropriate end of the range.

| AWB CTRL/AIL VALUE | | | |
|---|---|---|---|
| Byte | Bytes | — | |
| 3 | 5, 6 | Use | |
| 0 | 16-bit unsigned value | New auto iris level value. This sub-opcode is used by the Endura/Atlas projects. | D_ECS_ADJUST_AI_LEVEL_NEW |
| 1 | 16-bit signed value | Auto iris level change | D_ECS_ADJUST_AI_LEVEL_DELTA |

## 5.34 Command 0x43 ($67_{10}$), Adjust Auto-Iris Peak Value

### 5.34.1 Command format

| Adjust auto-iris peak value D_EC_ADJUST_AI_PEAK | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | AI CTRL | 0x43 | AI PEAK MSB | AI PEAK LSB | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:

2.1 0x0043 ADJUST_AUTO_IRIS_PEAK_VALUE

2.2 0x0143 SET_AUTO_IRIS_PEAK_VALUE

### 5.34.2 Description

If byte 3 is 0, byte 5 and byte 6 are the high and low bytes respectively of an unsigned 16-bit number which the new peak value. If byte 3 is 1, byte 5 and byte 6 are the high and low bytes respectively of a signed 16-bit number which is the change to the current peak. The peak is limited internally to the range from 0 → 255 (0xFF) for older cameras. With the current Hitachi cameras the range is 0 → 127 (0x00 → 0x7F). If an attempt is made to set or change the peak to a value outside this range, the peak will be set to the appropriate end of the range.

[56] $Header: 43.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

| AI CTRL/PEAK VALUE | | | |
|---|---|---|---|
| Byte | Bytes | — | |
| 3 | 5, 6 | Use | |
| 0 | 16-bit unsigned value | New auto iris peak value | D_ECS_ADJUST_AI_PEAK_NEW |
| 1 | 16-bit signed value | Auto iris peak change | D_ECS_ADJUST_AI_PEAK_DELTA |

## 5.35 Command 0x45 ($69_{10}$), Query

### 5.35.1 Command format

| Query D_EC_QUERY | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | QUERY TYPE | 0x45 | 0x00 | 0x00 | — |

FPN

1. This command generates a "Query Reply" (subsection 4.4).
2. Command Names as used by the **Spectra IV** software:
   2.1 0x0045 QUERY
3. This command is used by the Endura/Atlas projects.
4. This command is used and decoded on the: **Esprit 3012**.
5. This command is not supported in **ERD**, **IRD** or **LRD** type units.

### 5.35.2 Description

This command does not utilize the address field. This is so that the address of a unit may be determined programmabily.

The response to the Query command is:

The address field is the address of the device responding to the query. The content of the part number field is dependent on the type and version of the device being programmed, please refer to the table that follows.

The checksum is the 8 bit (modulo 256) sum of the transmitted query command's checksum, the address of the response, and the 15-byte part number.

**Phase in of the part number/model number/serial number readouts.** Note the change over from program number to model number to additions of the serial number.

1. **Spectra** ASCII text string of the **program** number and version of device. The string was internal to the software and this op-code was unsupported. Thus there was no way to readout the value.

   E.G. PG53-0001-R206

2. **Spectra II** ASCII text string of the **program** number and version of device. E.G. PG53-0060-R331

[57] $Header: 45.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

3. **Spectra III** prior to version 1.20. ASCII text string of the **program** number and version of device. E.G. `PG53-0060-R400`
4. **Spectra III** version 1.22 and later. ASCII text string of the device **model** number. E.G. `DD53C22-X`
5. **Spectra IV** prior to rev 2.000 is identical the **Spectra III** versions 1.22 and later.
6. **Spectra IV** versions following rev 2.000, have a shortened part number of the genera form `DD4nn` with `nn` being the maximum optical zoom level.
7. **Esprit** prior to version 3.10. ASCII text string of the **program** number and version of device. E.G. `PG53-0097-R306`
8. **Esprit** version 3.10 and later. ASCII text string of the device **model** number. E.G. `ES31CBW18`
9. All units than those above, have the ASCII text string of the device **model** number. E.G. `ES31CBW18`
10. **Spectra IV** and **Esprit TI** also have the ASCII text string of the device **serial** number available through use of an enhanced SUB OP-CODE of `0x01`. E.G. `123456`

**Esprit model number generation** The **Esprit** combines the following strings to build up the curent model type, all of the resulting model numbers are blank "␣" filed.

| | |
|---|---|
| "`ES30`" | For non-wiper models |
| "`ES31`" | For wiper models |
| "`P`" | For IOC models (IOP models have no special flag) |
| "`C16`" | For 16X camera types |
| "`C22`" | For 22X camera types |
| "`CBW18`" | For 18X CBW camera types |
| "`CBW24`" | For 24X CBW camera types |
| "`CBW35`" | For 35X CBW camera types |
| "`X`" | For PAL models (For NTSC models there is no suffix) |

These are some typical values for X18 **Esprit** models:

| | |
|---|---|
| `ES31CBW18␣␣␣␣␣␣` | `ES31CBW18X␣␣␣␣␣` |
| `ES31PCBW18␣␣␣␣␣` | `ES31PCBW18X␣␣␣␣` |
| `ES30CBW18␣␣␣␣␣␣` | `ES30CBW18X␣␣␣␣␣` |
| `ES30PCBW18␣␣␣␣␣` | `ES30PCBW18X␣␣␣␣` |

Sample model number for a X18 Esprit with wiper: "`ES31PCBW18␣␣␣␣␣`". In this example the ␣ symbol indicates blank (`0x20`) padding being used.

**Camera modules** *vs.* **Esprit model numbers** The Esprit software, starting in 2010 with rev 4.00, supports at least 11 different models of Hitachi camera. Support is for both NTSC and PAL versions:

| Type | Hitachi camera types |
|---|---|
| "`C16`" | VK-S934, VK-934R |
| "`C22`" | VK-S274, VK-274R |
| "`CBW18`" | VK-S454, VK-454R, VK-454N |
| "`CBW24`" | VK-P554 |
| "`CBW35`" | VK-S654, VK-654R, VK-654N |

**Spectra III Part number generation** The **Spectra III** generates a response to a QUERY command which is composed of two or three contatenated fields. These fields are selected from the following table (by selecting one entry from each column) depending on the exact model of **Spectra III**. The resulting string is NULL padded to be 15 bytes long with no trailing delimiter.

| DD53 | TC16 | -X |
|---|---|---|
| | C22 | |
| | M22 | |
| | CBW | |

Sample model number for a X22 Spectra III SE: "DD53C22␣␣␣␣␣␣␣␣". In this example the ␣ symbol indicates NULL (0x00) padding being used.

**Spectra IV Part number generation**

| DD4 | TC16 | -X | |
|---|---|---|---|
| | C22 | | |
| | M22 | | |
| | CBW23 | | |
| | CBW35 | | |
| DD4 | 23 | | |
| | 27 | | |
| | 35 | | |
| DD4H | CBW35 | | Look Up dome |

Sample model number for a X35 Spectra IV SE prior to rev 2.000:
"DD4CBW35␣␣␣␣␣␣␣". In this example the ␣ symbol indicates NULL (0x00) padding being used.
Sample model number for a X35 Spectra IV SE following rev 2.000:
"DD435␣␣␣␣␣␣␣␣␣␣". In this example the ␣ symbol indicates NULL (0x00) padding being used.

**Camera modules** *vs.* **Esprit model numbers** The Spectra IV, starting in 2009, supports 11 different models of Hitachi camera. These are:

| Type | Hitachi camera types |
|---|---|
| "TC16" | Unknown |
| "C22" | VK-S274, VK-S274R |
| "M22" | VK-K274, VK-K274R |
| "CBW23" | VK-S454, VK-S454R, VK-S454N |
| "CBW35" | VK-S654, VK-S654R |
| "23" | VK-S454N (Defeatured) |
| "27" | VK-S654N (Defeatured) , VK-S624 |
| "35" | VK-S654N |

**Esprit TI Model number generation**

| | | | | | |
|---|---|---|---|---|---|
| ES30 | | TI | | | Esprit Thermal Imager |
| | 14 | | | | 14.25mm focal length lens |
| | 35 | | | | 35mm focal length lens |
| | 50 | | | | 50mm focal length lens |
| | | | X | | PAL instead of NTSC video format |
| | | | | 1 | Frame rate of 9 Hz instead of 30 Hz (25 Hz for PAL) rate |

**Esprit TI** sample part numbers:

1. "ES3035TI␣␣␣␣␣␣␣". A unit with 30 Hz frame rate NTSC video output and a 35mm lens.
2. "ES3035TI1␣␣␣␣␣␣". A unit with 9 Hz frame rate NTSC video output and a 35mm lens.
3. "ES3035TIX␣␣␣␣␣␣". A unit with 25 Hz frame rate PAL video output and a 35mm lens.
4. "ES3035TIX1␣␣␣␣␣". A unit with 9 Hz frame rate PAL video output and a 35mm lens.

In these examples the ␣ symbol indicates NULL (0x00) padding being used.

## 5.36 Command 0x47 ($71_{10}$), Preset Scan

### 5.36.1 Command format

| Preset Scan D_EC_PRESET_SCAN | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x47 | 0x00 | DWELL | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x0047 PRESET_SCAN
3. This command is decoded on the: Esprit 3012.

### 5.36.2 Description

This command tells a PTZ to visit every defined preset for a given number of seconds (the number of seconds are defined in DWELL). So this is a method of implementing a preset tour at the PTZ level. **This command is not used as a D command**. It is a 32-bit Coaxitron® only command.

[58] $Header: 47.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

## 5.37 Command 0x49 ($73_{10}$), Set Zero Position

### 5.37.1 Command format

| Set Zero Position D_EC_SET_ZERO | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x49 | 0x00 | 0x00 | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x0049 SET_ZERO_POSITION
3. This command is used by the Endura/Atlas projects.
4. The angular value of where the displayed "zero point" has been moved to may be read out with the command/response pair of QUERY AZIMUTH ZERO (subsubsection 5.59.2) and QUERY AZIMUTH ZERO RESPONSE (subsubsection 5.59.3).

### 5.37.2 Description

This command is used to set the pan position that the unit uses as a zero reference point for the azimuth on-screen display. The unit's current pan position when this command is received becomes the zero reference point for on screen Pan display. This command performs the same function as the "SET AZIMUTH ZERO" menu item.

## 5.38 Command 0x4B ($75_{10}$), Set Pan Position

### 5.38.1 Command format

| Set Pan Position D_EC_SET_PAN | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x4B | PAN MSB | PAN LSB | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x004B SET_PAN_POSITION
3. This command is used by the Endura/Atlas projects.
4. This position may be read out using the command/response pair QUERY PAN POSITION (subsection 5.41) and QUERY PAN POSITION RESPONSE (subsection 5.45).

[59] $Header: 49.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
[60] $Header: 4B.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

### 5.38.2 Description

This command is used to set the pan position of the device. The position is given in hundredths of a degree and has a range from 0 → 35999 (decimal).

Sub-op codes used in CMND1

1. 0x00 is used for setting the pan position.

   Generally these values are interpreted as follows: Zero degrees indicates that the device is pointed horizontally (at the horizon). Ninety degrees indicates that the device is pointed straight down.

FPN

Example: the value to use to set the pan position to 45 degrees is 4500.

Note that the value used here is always the "absolute" pan position. It does not take into account any adjustment to the screen display that may have been made by using the "SET ZERO POSITION", opcode (0x49) command or the "SET AZIMUTH ZERO" menu item. See Appendix D for more information on pan positioning.

## 5.39 Command 0x4D ($77_{10}$), Set Tilt Position

### 5.39.1 Command format

| Set Tilt Position D_EC_SET_TILT | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x4D | TILT MSB | TILT LSB | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:

   2.1 0x004D SET_TILT_POSITION
3. This command is used by the Endura/Atlas projects.
4. This position may be read out using the command/response pair QUERY TILT POSITON (subsection 5.42) and QUERY TILT POSITON RESPONSE (subsection 5.46).

### 5.39.2 Description

This command is used to set the tilt position of the device. The position is given in hundredths of a degree and has a range from 0 → 35999 (decimal).

Sub-op codes used in CMND1

1. 0x00 is used for setting the tilt position.

   Generally these values are interpreted as follows: Zero degrees indicates that the device is pointed horizontally (at the horizon). Ninety degrees indicates that the device is pointed straight down.

FPN

[61] $Header: 4D.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

Examples:

1. The value used to set the tilt position to 45 degrees below the horizon is 4500.
2. The value used to set the tilt position to 45 degrees above the horizon, is 31500.

Note that different equipment will have different ranges of tilt motion. To determine the abilities of a specific piece of equipment, refer to that device's operation manual.

## 5.40 Command 0x4F ($79_{10}$), Set Zoom Position

### 5.40.1 Command format

| Set Zoom Position<br>D_EC_ZOOM | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x4F | ZOOM MSB | ZOOM LSB | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:

   2.1 0x004F SET_ZOOM_POSITION
3. In the Esprit, if Electronic Image Stabilization (EIS) is turned on, this command is ignored with a General Reply being sent back.
4. This position may be read out using the command/response pair QUERY ZOOM POSITION (subsection 5.43) and QUERY ZOOM POSITION RESPONSE (subsection 5.46).

### 5.40.2 Description

This command is used to set the zoom position of the device. The position is given as a ratio based on the device's Zoom Limit setting. The position is calculated as follows:

Position = (desired_zoom_position / zoom_limit) * 65535

Where desired_zoom_position and zoom_limit are given in units of magnification.

FPN

Example: Given that the zoom limit of the device's camera is X184, calculate the value needed to set the zoom position to X5:

Position = (5 / 184) * 65535 = approximately 1781

This works out to: decimal 1781 = 0x06F5 with byte 5 = 0x06 and byte 6 = 0xF5.

Commands in D protocol can not send floating point numbers. Since the result of this division is ALWAYS smaller than 1, we multiply it by 64K and round (or truncate) to get an integer. (Here we rounded the result.) To use this method the maximum zoom value must be known. There is no command to ask the Spectra what the maximum zoom value is.

Maximum zoom limit is set in one of the menus that the Spectra supports. The exact value varies depending on the exact type of camera installed.

[62] $Header: 4F.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

## 5.41 Command 0x51 ($81_{10}$), Query Pan Position

### 5.41.1 Command format

| Query Pan Position D_EC_QUERY_PAN | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x51 | 0x00 | 0x00 | — |

FPN

1. This command generates an "Extended Reply" (subsection 4.3) of QUERY PAN POSITION RESPONSE (subsection 5.45).
2. This command is used by the Endura/Atlas projects.
3. This command generates a QUERY PAN POSITION RESPONSE (subsection 5.45)

### 5.41.2 Description

This command is used to query the current pan position of the device. The response to this command uses opcode 0x59. See QUERY PAN POSITION RESPONSE (subsection 5.45) for more information.

## 5.42 Command 0x53 ($83_{10}$), Query Tilt Position

### 5.42.1 Command format

| Query Tilt Position D_EC_QUERY_TILT | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x53 | 0x00 | 0x00 | — |

FPN

1. This command generates an "Extended Reply" (subsection 4.3) of QUERY TILT POSITION RESPONSE (subsection 5.46).
2. This command is used by the Endura/Atlas projects.
3. This command generates a QUERY TILT POSITION RESPONSE

### 5.42.2 Description

This command is used to query the current tilt position of the device. The response to this command uses opcode 0x5B. See QUERY TILT POSITION RESPONSE (subsection 5.46) for more information.

[63] $Header: 51.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
[64] $Header: 53.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

## 5.43 Command 0x55 ($85_{10}$), Query Zoom Position

### 5.43.1 Command format

| Query Zoom Position D_EC_QUERY_ZOOM | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x55 | 0x00 | 0x00 | — |

FPN

1. This command generates an "Extended Reply" (subsection 4.3) of QUERY ZOOM POSITION RESPONSE (subsection 5.47).
2. This command generates QUERY ZOOM POSITION RESPONSE (subsection 5.47)
3. In Esprit rev 3.71 support was added for the 35X camera. One of the features of the 35X camera is that it has Electronic Image Stabilization (EIS). When EIS is **off**, then the Esprit will generate a QUERY MAGNIFICATION RESPONSE (subsection 5.50) and when EIS is **on** then a "General Reply" (subsection 4.2) will be generated. The Spectra IV always generates a QUERY MAGNIFICATION RESPONSE (subsection 5.50).

### 5.43.2 Description

This command is used to query the current zoom position of the device. The response to this command uses opcode 0x5D. See QUERY ZOOM POSITION RESPONSE (subsection 5.47) for more information.

## 5.44 Command 0x57 ($87_{10}$), Prepare For Download

### 5.44.1 Command format

| Prepare For Download D_EC_DOWNLOAD | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x57 | 0x00 | 0x00 | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:

   2.1 0x0057 DOWNLOAD

### 5.44.2 Description

Places the device into a state where it is prepared to receive a firmware update.

D_EC_DOWNLOAD forces a baud rate of 2400 and expects a reply at the original baud rate. I.e. if the Spectra is running at 4800 baud, when it receives an D_EC_DOWNLOAD command it will send a response at 4800 baud and then reconfigure to operate at 2400 baud.

Use of this command in a download sequence is outlined in Appendix E.

[65] $Header: 55.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
[66] $Header: 57.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

## 5.45 Response 0x59 ($89_{10}$), Query Pan Position Response

### 5.45.1 Response format

| Query Pan Position Response D_EC_PAN_RESP | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x59 | PAN MSB | PAN LSB | — |

FPN

1. This response is an "Extended Reply" (subsection 4.3).
2. This response is the reply to a QUERY PAN POSITION (subsection 5.41)

### 5.45.2 Description

The position is given in hundredths of a degree and has a range from 0 → 35999 (decimal).

FPN

Example: a position value of 4500 indicates 45 degrees.

Note that the value returned is always the "absolute" pan position. It does not take into account any adjustment to the screen display that may have been made by using the "SET ZERO POSITION", opcode (see subsection 5.37) command or the "SET AZIMUTH ZERO" menu item.

For more information about how the interactions between how "SET ZERO POSITION" (subsection 5.37) and "QUERY PAN POSITION" (subsection 5.41) opcodes operate see Appendix D.

## 5.46 Response 0x5B ($91_{10}$), Query Tilt Position Response

### 5.46.1 Response format

| Query Tilt Position Response D_EC_TILT_RESP | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x5B | TILT MSB | TILT LSB | — |

FPN

1. This response is an "Extended Reply" (subsection 4.3).
2. This response is the reply to a QUERY TILT POSITION (subsection 5.42)

### 5.46.2 Description

The position is given in hundredths of a degree and has a range from 0 → 35999 (decimal). Refer to examples listed in description of the "SET TILT POSITION", opcode subsection 5.39.

Tilt angle values comes in in two bytes as degrees times 100 "hungrees".

[67] $Header: 59.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
[68] $Header: 5B.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

| Position | D reads out as | Spectra displays as |
|---|---|---|
| 90° up | 27000 | 90° |
| 45° up | 31500 | 45° |
| Horizontal - 1° | 35900 | 1° |
| Horizontal | 000 | 0° |
| 45° down | 4500 | -45° |
| 90° down | 9000 | -90° |

**Position** Pointing direction of the enclosure/camera

**D reads out as** D protocol returned value for this angle

**Spectra displays as** What is displayed on the Spectra screen

## 5.47 Response 0x5D ($93_{10}$), Query Zoom Position Response

### 5.47.1 Response format

| Query Zoom Position Response D_EC_ZOOM_RESP | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x5D | ZOOM MSB | ZOOM LSB | — |

FPN

1. This response is an "Extended Reply" (subsection 4.3).
2. This response is the reply to a QUERY ZOOM POSITION (subsection 5.43) command

### 5.47.2 Description

The position is given as a ratio based on the device's Zoom Limit setting. This value can be converted into units of magnification by using the following formula:

current_magnification = (position / 65535) * zoom_limit

Where current_zoom_position and zoom_limit are given in units of magnification.

FPN

Example: Given that the zoom limit of the device's camera is X184, position value is 1781, calculate the current magnification:

Current magnification = (1781 / 65535) * 184 = approximately X5.

## 5.48 Command 0x5F ($95_{10}$), Set Magnification

### 5.48.1 Command format

| Set Magnification D_EC_SET_MAG | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | REL/ABS | 0x5F | MAG MSB | MAG LSB | — |

[69] $Header: 5D.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

[70] $Header: 5F.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 `0x005F SET_MAGNIFICATION`
   2.2 `0x015F SCREEN_SET_MAG`
3. This command is used by the Endura/Atlas projects.
4. In the Esprit, if Electronic Image Stabilization (EIS) is turned on, this command is ignored with a General Reply being sent back.
5. This position may be read out using the command/response pair in QUERY MAGNIFICATION (subsection 5.49) and QUERY MAGNIVICATION RESPONSE (subsection 5.50).

### 5.48.2 Description

This command is used to set the zoom position of the device.

The position (MAG MSB and MAG LSB) form a signed, 16-bit, 2's complement value expressing the desired setting, or change in setting, in hundredths of units.

Maximum zoom limit is set in one of the menus that the Spectra supports. The exact value varies depending on the exact type of camera installed.

There is no command to ask the Spectra what the maximum zoom value is.

FPN

Example: A value of 500 means X5.

For example if we want a zoom level of X5 then we have to send a value of 5 * 100 = 500. 500 decimal = `0x01F4` or byte 5 = `0x01` and byte 6 = `0xF4`. The Spectra will not zoom "past the end" of its range as set by the maximum zoom value.

**Changes:** The relative change in magnification was added in to support the Atlas project and is available on the Spectra IV only.

1. REL/ABS = `0x00` indicates that the value in DATA1 and DATA2 represent an absolute change in magnification value.

   That is, a value of 200 indicates a setting of 2X for absolute,

2. REL/ABS = `0x01` indicates that the value in DATA1 and DATA2 represent a relative change in magnification value.

   That is, a value of 200 indicates a setting of 2X for a change of setting of 2.00 for relative.

FPN

If a relative magnification value of 350 is sent when the current setting is 1.5X, the resulting magnification will be 3.5 × 1.5, or 5.25X. The sign indicates the direction of change for relative values and must always be positive for absolute settings. For relative changes, a negative value indicates that the result should be smaller than the current setting. That is, if the value is -3.5 and the current setting is 12X, the result will be 3.43X. In no case will the result be less than 1.00X.

This is the only 1 of 2 D Protocol commands that has a **`signed int`** as DATA1 and DATA2 which are concatenated together.

## 5.49 Command 0x61 ($97_{10}$), Query Magnification

### 5.49.1 Command format

| Query Magnification<br>D_EC_QUERY_MAG | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x61 | 0x00 | 0x00 | — |

FPN

1. This command generates an "Extended Reply" (subsection 4.3) of QUERY MAGNIFICATION RESPONSE (subsection 5.50).
2. This command is used by the Endura/Atlas projects.
3. This opcode was first implemented in Spectra III version 1.14. It is not in version 1.16.
4. In Esprit rev 3.71 support was added for the 35X camera. One of the features of the 35X camera is that it has Electronic Image Stabilization (**EIS**). When EIS is **off**, then the Esprit will generate a QUERY MAGNIFICATION RESPONSE (subsection 5.50) and when EIS is **on** then a "General Reply" (subsection 4.2) will be generated. The Spectra IV always generates a QUERY MAGNIFICATION RESPONSE (subsection 5.50).
5. This command generates QUERY MAGNIFICATION RESPONSE (subsection 5.50)

### 5.49.2 Description

This command is used to query the current zoom position of the device. The response to this command uses opcode 0x63. See QUERY MAGNIFICATION RESPONSE (subsection 5.50) for more information.

## 5.50 Response 0x63 ($99_{10}$), Query Magnification Response

### 5.50.1 Response format

| Query Magnification Response<br>D_EC_MAG_RESP | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x63 | MAG MSB | MAG LSB | — |

FPN

1. This response is an "Extended Reply" (subsection 4.3).
2. This opcode was first implemented in Spectra III version 1.14. It is not in version 1.16.
3. This response is the reply to a QUERY MAGNIFICATION (subsection 5.49) command
4. The Esprit TI generates a General Response, not an Extended Response, because it does not have a lens capable of "zooming".

### 5.50.2 Description

The value returned is given in hundredths of units of magnification.

Example: a value of 500 means X5.

[71] $Header: 61.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
[72] $Header: 63.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

## 5.51 Command 0x65 ($101_{10}$), Activate Echo Mode

### 5.51.1 Command format

| Activate Echo Mode D_EC_ECHO_MODE | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x65 | 0x00 | 0x00 | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x0065 ACTIVATE_ECHO_MODE

### 5.51.2 Description

Places the device into a mode in which characters that are received by the unit are immediately retransmitted. The unit exits this mode when one of the following occurs: more than 100 milliseconds pass without receipt of a character or more than 180 characters have been received.

This command is sent at the current D_EC_SET_BAUD rate.

Use of this command in a download sequence is outlined in Appendix E.

## 5.52 Command 0x67 ($103_{10}$), Set Remote Baud Rate

### 5.52.1 Command format

| Set Remote Baud Rate D_EC_SET_BAUD | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | MODE | 0x67 | 0x00 | BAUD CODE | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x0067 SET_REMOTE_BAUD_RATE
   2.2 0x0167 SET_CURRENT_BAUD_RATE
3. This command is used by the Endura/Atlas projects.

### 5.52.2 Description

Sets the unit's baud rate. Valid values for this command are:

[73] $Header: 65.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
[74] $Header: 67.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

| Download baud rates | | |
|---|---|---|
| Value | Baud | |
| 0 | 2400 | D_ECD_SET_BAUD_2400 |
| 1 | 4800 | D_ECD_SET_BAUD_4800 |
| 2 | 9600 | D_ECD_SET_BAUD_9600 |
| 3 | 19200 | D_ECD_SET_BAUD_19200 |
| 4 | 38400 | D_ECD_SET_BAUD_38400 |
| 5 | 115200 | D_ECD_SET_BAUD_115200 |

1. During a download sequence D_EC_SET_BAUD is always sent at 2400 baud and its response is always sent at 2400 baud. This works because a delay of at least 100 milliseconds has occurred since the last data byte transferred. Thus the Spectra has returned to the "recovery state" of 2400 baud. This implies that there must be no "gap" longer than 100 ms during a download sequence.

2. The unit sends its response to this command before changing its baud. The baud automatically returns to 2400 after 100 milliseconds of no activity when the MODE is 0x00.

3. The Spectra III always sends a General Response for all baud rate changes (CMND1 = 0x00) at 2400 baud and then changes its baud rate to the selected rate.

Use of this command in a download sequence is outlined in Appendix E.

## 5.53 Command 0x69 ($105_{10}$), Start Download

### 5.53.1 Command format

| Start Download D_EC_START_DOWNLOAD | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x69 | 0x00 | 0x00 | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. Command Names as used by the Spectra IV software:
   2.1 0x0069 START_DOWNLOAD

### 5.53.2 Description

Places the device into a state where it is expecting download commands and down load data — instead of "D" protocol commands.

Use of this command in a download sequence is outlined in Appendix E.

**Change with Spectra IV:** This command's actions have been changed with the Spectra IV. Now when this command is received, the Spectra IV starts sending 0x02s to the head end. Earlier versions did not do this if the entire sequence of command outlined in Appendix E had not been previously sent.

(DTE is the GlassKeyboard at 2400 baud, DCE is a Spectra IV running 1.070 software.)

[75] $Header: 69.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

```
160,    1135:  DTE   1849   707.100414   0.004167 ff
160,    1136:  DTE   1850   707.104551   0.004137 01
160,    1137:  DTE   1851   707.108551   0.004000 00
160,    1138:  DTE   1852   707.112696   0.004145 69
160,    1139:  DTE   1853   707.117035   0.004339 00
160,    1140:  DTE   1854   707.121167   0.004132 00
160,    1141:  DTE   1855   707.125331   0.004164 6a

161,     715: DCE    1856   707.133968   0.008637 ff
161,     716: DCE    1857   707.138108   0.004140 01
161,     717: DCE    1858   707.142277   0.004169 00
161,     718: DCE    1859   707.146444   0.004167 6a
161,     719: DCE    1860   708.772541   1.626097 02 These are a new action
161,     720: DCE    1861   710.396525   1.623984 02
161,     721: DCE    1862   712.020927   1.624402 02
161,     722: DCE    1863   713.645158   1.624231 02
161,     723: DCE    1864   715.269754   1.624596 02
161,     724: DCE    1865   716.894154   1.624400 02
```

## 5.54 Command 0x6B ($107_{10}$), Query Device Type

### 5.54.1 Command format

| Query Device Type<br>D_EC_QUERY_DEV_TYPE | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x6B | 0x00 | 0x00 | — |

FPN

1. This command generates an "Extended Reply" (subsection 4.3) of QUERY DEVICE TYPE RESPONSE (subsection 5.55).

### 5.54.2 Description

This command is used to query the device for information about the hardware platform the device is using and the type of software that is running on the platform. The response to this command uses opcode 0x6D. See QUERY DEVICE TYPE RESPONSE (subsection 5.55) for more information.

## 5.55 Response 0x6D ($109_{10}$), Query Device Type Response

### 5.55.1 Response format

| Query Device Type Response<br>D_EC_DEV_TYPE_REP | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x6D | HARD TYPE | SOFT TYPE | — |

FPN

1. This response is an "Extended Reply" (subsection 4.3).
2. This response is the reply to a QUERY MAGNIFICATION (subsection 5.49) command

[76] $Header: 6B.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
[77] $Header: 6D.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

### 5.55.2 Description

1. The value returned in "byte 5" (HARD TYPE) indicates the software type. Valid values are:
   - 1.1 01 Spectra III Application
   - 1.2 02 Spectra III BIOS
   - 1.3 03 ExSite Application
   - 1.4 04 ExSite BIOS
2. The value returned in "byte 6" (SOFT TYPE) indicates the hardware type. Valid values are:
   - 2.1 0x17 MMC2107 processor.
   - 2.2 0x1E MMC2114 processor

## 5.56 Command 0x6F ($111_{10}$), Query Diagnostic Information

### 5.56.1 Command format

| Query Diagnostic Information D_EC_QUERY_DIAG_INFO | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x6F | 0x00 | 0x00 | — |

FPN

1. This response is an "Extended Reply" (subsection 4.3) of QUERY DIAGNOSTIC INFORMATION RESPONSE (subsection 5.57).
2. This command generates QUERY DIAGNOSTIC INFORMATION RESPONSE (subsection 5.57)

### 5.56.2 Description

This command is used to query the device for diagnostic information. The response to this command uses opcode 0x71. See QUERY DIAGNOSTIC INFORMATION RESPONSE (subsection 5.57) for more information.

1. **Spectra IV** starting with rev 1.072, has a reply using opcode 0x71 (subsection 5.57) having the unit's temperature in it.
2. **Esprit** starting with rev 4.06, has a reply (subsection 5.57) with additonal internal status information.

## 5.57 Response 0x71 ($113_{10}$), Query Diagnostic Information Response

### 5.57.1 Response format

| Query Diagnostic Information Response D_EC_QUERY_DIAG_RESP | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x71 | TEMP | SENSOR ID | — |

[78] $Header: 6F.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
[79] $Header: 71.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

FPN

1. This response is an "Extended Reply" (subsection 4.3).
2. This response is the reply to a QUERY DIAGNOSTIC INFORMATION (subsection 5.56) command.

### 5.57.2 Description

The contents of the message may vary based on the type of device that is being queried. For **Spectra III** the contents of the message are defined as follows:

1. Byte 5, DATA1, of the message is always `0x00` for the **ExSite**, **Esprit**, **Spectra II**, **Spectra III**.
2. Byte 5, DATA1, has been changed in newer versions of the PTZ units:
   - 2.1 **Spectra IV** starting with rev 1.072. DATA1 is now the internal temperature of the unit in degrees F. The range is $0 \rightarrow 255$.
3. Byte 6, DATA2, of the message is a sensor position indicating bit:
   - 3.1 SENSOR ID: Bit 0 of byte 6 is the pan sensor indicator. If the bit is on then the unit is oriented such that the pan sensor is being detected.
   - 3.2 SENSOR ID: Bit 1 of byte 6 is the tilt sensor indicator. If the bit is on then the unit is oriented such that the tilt sensor is being detected.

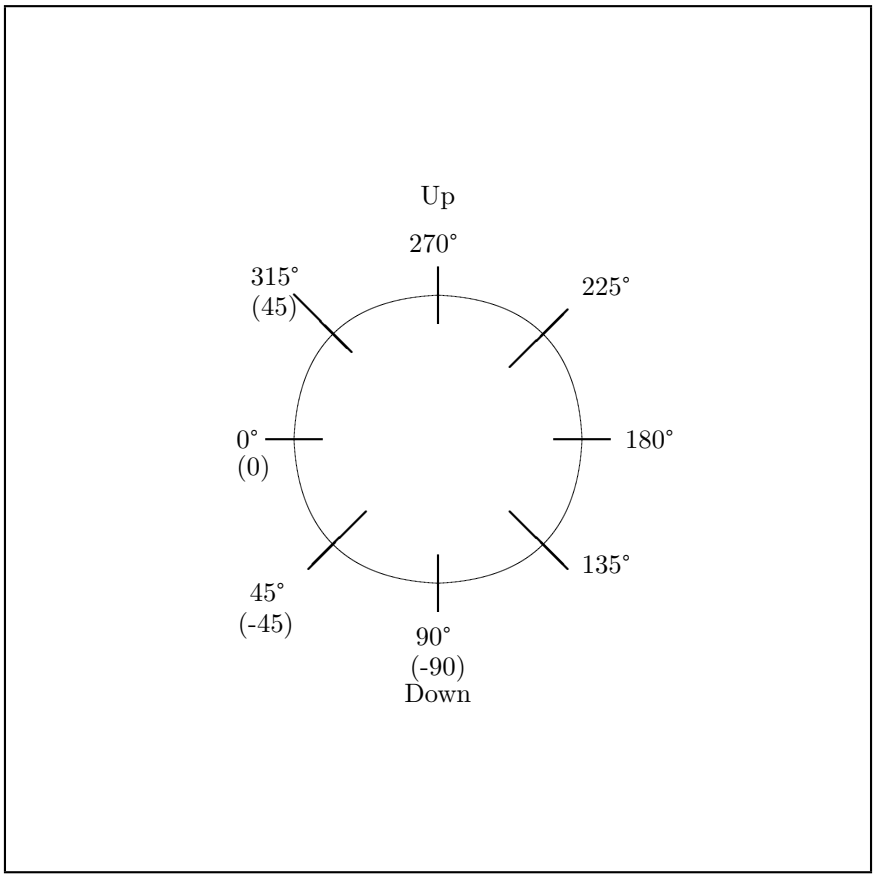



Figure 1: D Protocol Tilt Readout Positions. Values in parentheses are the values displayed on PTZ units.

### 5.57.3 Esprit Rev 4.06 and newer diagnostic responses

<table>
<tr><th colspan="3">0x71, Query Diagnostic Information Response</th></tr>
<tr><th>CMND1</th><th>CMND2</th><th>Use</th></tr>
<tr><td>0x01</td><td>0x6F</td><td>Returns two bytes (or one <code>16 bit unsigned int</code>) of internal status bits as follows, all bits are or'ed together:
<table>
<tr><th>DATA1:</th><th>Internal Data #1</th></tr>
<tr><td>0x80:</td><td>Non-English/English Menus 0 = english, 1 = non-english</td></tr>
<tr><td>0x40:</td><td><code>IR Level state</code> Dusk, 1 = Dark</td></tr>
<tr><td>0x20:</td><td><code>IR Mode state</code> Auto, 1 = off</td></tr>
<tr><td>0x10:</td><td><code>TOL type</code> 0 = T1, 1 = T2</td></tr>
<tr><td>0x08:</td><td><code>FNR state</code> 0 = off, 1 = on</td></tr>
<tr><td>0x04:</td><td><code>Pict Mode</code> 0 = 1, 1 = 2</td></tr>
<tr><td>0x02:</td><td><code>Sure Focus state</code> 0 = off, 1 = on</td></tr>
<tr><td>0x01:</td><td><code>WDR state</code> 0 = off, 1 = on</td></tr>
</table>
<table>
<tr><th>DATA2:</th><th>Internal Data #2</th></tr>
<tr><td>0x80:</td><td><code>EIS Center Frequency</code> 5hz/10hz</td></tr>
<tr><td>0x40:</td><td><code>EIS state</code> 0 = off, 1 = on</td></tr>
<tr><td>0x20:</td><td><code>BLC state</code> 0 = off, 1 = on</td></tr>
<tr><td>0x10:</td><td><code>WDR state</code> 0 = off, 1 = on</td></tr>
<tr><td>0x08:</td><td><code>Auto Iris state</code> 0 = off, 1 = auto</td></tr>
<tr><td>0x04:</td><td><code>Auto Sharpness state</code> 0 = off, 1 = on</td></tr>
<tr><td>0x02:</td><td><code>Auto Focus state</code> 0 = off, 1 = auto</td></tr>
<tr><td>0x01:</td><td><code>AWB state</code> 0 = off, 1 = auto</td></tr>
</table>
</td></tr>
</table>

## 5.58 Command 0x73 ($115_{10}$), Version Information Macro Opcode

### 5.58.1 Command format

| Version Information Macro Opcode D_EC_VERSION_INFO | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| | 0xFF | — | SUB OP-CODE | 0x73 | Various | Various | — |
| | Section | Page | | | | | |
| Cmnd | 5.58.2 | 75 | 0x00 | Request software version number | | | |
| Resp | 5.58.3 | 76 | 0x01 | Application version number | | | |
| Cmnd | 5.58.4 | 76 | 0x02 | Request build number | | | |
| Resp | 5.58.5 | 76 | 0x03 | Build number | | | |
| Cmnd | 5.58.2 | 75 | 0x00 | VERSION_REQUEST | | | |
| Resp | 5.58.3 | 76 | 0x01 | VERSION_RESPONSE | | | |
| Cmnd | 5.58.4 | 76 | 0x02 | BUILD_REQUEST | | | |
| Resp | 5.58.5 | 76 | 0x03 | BUILD_RESPONSE | | | |

FPN

1. These commands generate "General Responses" (subsection 4.2) and "Extended Responses" (subsection 4.3) including a "Standard Everest Response" (subsection 5.1).
2. This command is used by the Endura/Atlas projects.
3. Note: ACK and NAK message types are seven bytes in length. For a detailed description of ACK and NAK type replies refer to subsection 5.1.

### 5.58.2 Command for Request Software Version 0x73 Sub Op-Code 0x00

| Version Information Macro Opcode D_EC_VERSION_INFO D_ECS_VERSION_INFO_MAIN_CPU_VERSION_QRY | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x73 | 0x00 | 0x00 | — |

FPN

1. This command generates an "Extended Reply" (subsection 4.3).
2. Typical useage of this command, and its reply, is shown in Appendix A.
3. This sub-opcode is used by the Endura/Atlas projects.

   3.1 The version number request was added in for the Esprit starting with rev 3.77.

Request the software version number. It comes in as subsubsection 5.58.3.

[80] $Header: 73.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

### 5.58.3 Response to Request Software Version 0x73 Sub Op-Code 0x01

| Version Information Macro Opcode<br>D_EC_VERSION_INFO<br>D_ECS_VERSION_INFO_MAIN_CPU_VERSION_RSP | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x01 | 0x73 | VERSION MSB | VERSION LSB | — |

FPN

1. This response is an "Extended Reply" (subsection 4.3).
2. Typical useage of this command, and its reply, is shown in Appendix A.
3. This response is the reply to a APPLICATION VERSION NUMBER (subsection 5.58 SUB-OPCODE 01) command.
4. This sub-opcode response is used by the Endura/Atlas projects.

The VERSION number is returned as a 16 bit integer in bytes 5 and 6 as either the rev number ×100 or ×1000 for the Spectra IV.

### 5.58.4 Command for Request Build Number 0x73 Sub Op-Code 0x02

| Version Information Macro Opcode<br>D_EC_VERSION_INFO<br>D_ECS_VERSION_MAIN_CPU_BUILD_QRY | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x02 | 0x73 | 0x00 | 0x00 | — |

FPN

1. This command generates an "Extended Reply" (subsection 4.3).
2. Typical useage of this command, and its reply, is shown in Appendix A.

Request the software build number. It comes in as subsubsection 5.58.5.

### 5.58.5 Response to Request Build Number 0x73 Sub Op-Code 0x03

| Version Information Macro Opcode<br>D_EC_VERSION_INFO<br>D_ECS_VERSION_MAIN_CPU_BUILD_RSP | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x03 | 0x73 | BUILD MSB | BUILD LSB | — |

FPN

1. This response is an "Extended Reply" (subsection 4.3).
2. Typical useage of this command, and its reply, is shown in Appendix A.
3. This response is the reply to a REQUEST BUILD NUMBER (subsection 5.58 SUB-OPCODE 03) command.

The BUILD number is returned as an unsigned 16 bit integer in bytes 5 and 6.

## 5.59 Command 0x75 ($117_{10}$), Everest Macro Opcode

### 5.59.1 Command format

| Everest Macro Opcode D_EC_EVEREST | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| | 0xFF | — | SUB OP-CODE | 0x75 | Various | Various | — |
| | Section | Page | | | | | |
| Cmnd | 5.59.2 | 77 | 0x00 | Query azimuth zero offset command | | | |
| Resp | 5.59.3 | 78 | 0x01 | Query azimuth zero offset response | | | |
| Cmnd | 5.59.4 | 78 | 0x02 | Set zoom limit command | | | |
| Cmnd | 5.59.5 | 78 | 0x03 | Query zoom limit command | | | |
| Resp | 5.59.6 | 79 | 0x04 | Query zoom limit response | | | |
| Cmnd | 5.59.7 | 79 | 0x05 | Query alarms command | | | |
| Resp | 5.59.8 | 79 | 0x06 | Query alarms response | | | |
| Cmnd | 5.59.9 | 80 | 0x07 | Delete pattern command | | | |
| Cmnd | 5.59.10 | 80 | 0x08 | Set manual left pan limit command | | | |
| Cmnd | 5.59.11 | 80 | 0x09 | Set manual right pan limit command | | | |
| Cmnd | 5.59.12 | 81 | 0x0A | Set scan left pan limit command | | | |
| Cmnd | 5.59.13 | 81 | 0x0B | Set scan right pan limit command | | | |
| Cmnd | 5.59.14 | 81 | 0x0C | Query limit command | | | |
| Resp | 5.59.15 | 82 | 0x0D | Query limit response | | | |
| Cmnd | 5.59.16 | 82 | 0x0E | Enable/disable limits command | | | |
| Cmnd | 5.59.17 | 82 | 0x0F | Query defined presets command | | | |
| Resp | 5.59.18 | 83 | 0x10 | Query defined presets response | | | |
| Cmnd | 5.59.19 | 83 | 0x11 | Query defined patterns command | | | |
| Resp | 5.59.20 | 84 | 0x12 | Query defined patterns response | | | |

FPN

1. These commands are responses are used by the Endura/Atlas projects.
2. ACK and NAK message types are seven bytes in length. For a detailed description of ACK and NAK type replies refer to subsection 5.1.

   For the Spectra III with SUB OP-CODEs greater than 0x12 a General Response is always generated.

### 5.59.2 Command for Query Azimuth Zero 0x75 Sub Op-Code 0x00

| Query azimuth zero offset command D_EC_EVEREST D_ECS_EVEREST_AZIMUTH_ZERO_OFFSET_QRY | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x75 | 0x00 | 0x00 | — |

[81] $Header: 75.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

FPN

1. This command generates an "Extended Reply" (subsection 4.3) of subsubsection 5.59.3.
2. This sub-opcode is used by the Endura/Atlas projects.

### 5.59.3 Response to Query Azimuth Zero 0x75 Sub Op-Code 0x01

| Query azimuth zero offset response<br>D_EC_EVEREST<br>D_ECS_EVEREST_AZIMUTH_ZERO_OFFSET_RSP | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x01 | 0x75 | OFFSET MSB | OFFSET LSB | — |

FPN

1. This response is an "Extended Reply" (subsection 4.3).
2. This response sub-opcode is used by the Endura/Atlas projects.
3. OFFSET is a 16-bit number in hundredths of degrees.

### 5.59.4 Command for Set Zoom Limit 0x75 Sub Op-Code 0x02

| Set zoom limit command<br>D_EC_EVEREST<br>D_ECS_EVEREST_SET_MAX_MAG | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x02 | 0x75 | LIMIT MSB | LIMIT LSB | — |

FPN

1. The response to this command is ACK or NAK.
2. LIMIT is in hundredths. EG a value of 18400 means x184. Acceptable values are device specific.

### 5.59.5 Command for Query Zoom Limit 0x75 Sub Op-Code 0x03

| Query zoom limit command<br>D_EC_EVEREST<br>D_ECS_EVEREST_MAX_MAG_QRY | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x03 | 0x75 | 0x00 | 0x00 | — |

FPN

1. This command generates an "Extended Reply" (subsection 4.3) of subsubsection 5.59.6.
2. This sub-opcode is used by the Endura/Atlas projects.

### 5.59.6 Response to Query Zoom Limit 0x75 Sub Op-Code 0x04

| Query zoom limit response<br>D_EC_EVEREST<br>D_ECS_EVEREST_MAX_MAG_RSP | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x04 | 0x75 | LIMIT MSB | LIMIT LSB | — |

FPN

1. This response is an "Extended Reply" (subsection 4.3).
2. This response sub-opcode is used by the Endura/Atlas projects.
3. Limit is in hundredths. E.G. a value of 18400 means x184.

### 5.59.7 Command for Query Alarms 0x75 Sub Op-Code 0x05

| Query alarms command<br>D_EC_EVEREST<br>D_ECS_ALARM_QRY | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x05 | 0x75 | 0x00 | 0x00 | — |

FPN

1. This command generates an "Extended Reply" (subsection 4.3) of subsubsection 5.59.8.

### 5.59.8 Response to Query Alarms 0x75 Sub Op-Code 0x06

| Query zoom limit response<br>D_EC_EVEREST<br>D_ECS_ALARM_RSP | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x06 | 0x75 | 0x00 | ALARM BIT MASK | — |

FPN

1. This response is an "Extended Reply" (subsection 4.3).
2. The format of the ALARM BIT MASK is the same as the format of the alarms in the General Response.

### 5.59.9 Command for Delete Pattern 0x75 Sub Op-Code 0x07

| Delete pattern command<br>D_EC_EVEREST<br>D_ECS_DELETE_PATTERN | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x07 | 0x75 | 0x00 | PATTERN NUMBER | — |

FPN

1. This sub-opcode is used by the Endura/Atlas projects.
2. Pattern numbers start at 1. Valid PATTERN NUMBERS for Spectra III are 1 → 4. Valid pattern numbers for Spectra IV are 1 → 8.
3. The response is ACK if the pattern was deleted. The response is NAK if the pattern was not deleted or the PATTERN NUMBER is out of range. The reasons a pattern would not be deleted are if it was not defined or if pattern recording is in progress.

### 5.59.10 Command for Set Manual Left Pan Limit 0x75 Sub Op-Code 0x08

| Set manual left pan limit command<br>D_EC_EVEREST<br>D_ECS_SET_MAN_PAN_LEFT_LIMIT | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x08 | 0x75 | LIMIT MSB | LIMIT LSB | — |

FPN

1. The response is ACK if the LIMIT is in range. The response is NAK if the LIMIT is out of range.
2. Units for LIMIT are hundredths of degrees. The range of values is 0 → 35999.
3. General note regarding setting limits: Setting a limit does not automatically move the camera so that it is within the limits.
4. See also subsubsection 5.59.16.

### 5.59.11 Command for Set Manual Right Pan Limit 0x75 Sub Op-Code 0x09

| Set manual right pan limit command<br>D_EC_EVEREST<br>D_ECS_SET_MAN_PAN_RIGHT_LIMIT | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x09 | 0x75 | LIMIT MSB | LIMIT LSB | — |

FPN

1. The response is ACK if the LIMIT is in range. The response is NAK if the LIMIT is out of range.
2. Units for LIMIT are hundredths of degrees. The range of values is 0 → 35999.
3. See also subsubsection 5.59.16.

### 5.59.12 Command for Set Scan Left Pan Limit 0x75 Sub Op-Code 0x0A

| Set scan left pan limit command<br>D_EC_EVEREST<br>D_ECS_SET_SCAN_PAN_LEFT_LIMIT | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x0A | 0x75 | LIMIT MSB | LIMIT LSB | — |

FPN

1. The response is ACK if the LIMIT is in range. The response is NAK if the LIMIT is out of range.
2. Units for LIMIT are hundredths of degrees. The range of values is 0 → 35999.
3. See also subsubsection 5.59.16.

### 5.59.13 Command for Set Scan Right Pan Limit 0x75 Sub Op-Code 0x0B

| Set scan right pan limit command<br>D_EC_EVEREST<br>D_ECS_SET_SCAN_PAN_RIGHT_LIMIT | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x0B | 0x75 | LIMIT MSB | LIMIT LSB | — |

FPN

1. The response is ACK if the LIMIT is in range. The response is NAK if the LIMIT is out of range.
2. Units for LIMIT are hundredths of degrees. The range of values is 0 → 35999.
3. See also subsubsection 5.59.16.

### 5.59.14 Command for Query Limit 0x75 Sub Op-Code 0x0C

| Query limit command<br>D_EC_EVEREST<br>D_ECS_EVEREST_LIMIT_QRY | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x0C | 0x75 | 0x00 | LIMIT ID | — |

FPN

1. This command generates an "Extended Reply" (subsection 4.3).
2. This sub-opcode is used by the Endura/Atlas projects.
3. Request a limit. The following IDs are used:
   - 3.1 0x00: D_ECD_EVEREST_MAN_LEFT_PAN Manual left pan limit
   - 3.2 0x01: D_ECD_EVEREST_MAN_RIGHT_PAN Manual right pan limit
   - 3.3 0x02: D_ECD_EVEREST_SCAN_LEFT_PAN Scan left pan limit
   - 3.4 0x03: D_ECD_EVEREST_SCAN_RIGHT_PAN Scan right pan limit
4. The response is opcode 0x75, sub opcode 0x0D. (subsubsection 5.59.15)
5. If the ID is not supported by the device the response is NAK.
6. This command generates an "Extended Reply" (subsection 4.3) of subsubsection 5.59.15.

### 5.59.15 Response to Query Limit 0x75 Sub Op-Code 0x0D

| Query limit response<br>D_EC_EVEREST<br>D_ECS_EVEREST_LIMIT_RSP | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x0D | 0x75 | LIMIT MSB | LIMIT LSB | — |

FPN

1. This response is an "Extended Reply" (subsection 4.3).
2. This response sub-opcode is used by the Endura/Atlas projects.
3. This is the response to opcode 0x75, sub opcode 0x0C.
4. Units for LIMIT are hundredths of degrees. The range of values is 0 → 35999.
5. The LIMIT that is returned depends on the LIMIT ID in the command that is being responded to.

### 5.59.16 Command for Enable/Disable Limits 0x75 Sub Op-Code 0x0E

| Enable/Disable limits command<br>D_EC_EVEREST<br>D_ECS_EVEREST_ENABLE_LIMITS | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x0E | 0x75 | 0x00 | NEW LIMITS STATE | — |

FPN

1. These sub-opcodes for commands and responses are used by the Endura/Atlas projects.
2. Enables or disables the manual limits and scan limits. Allowed values for NEW LIMITS STATE are:
   2.1 0x00: D_ECD_EVEREST_ENABLE_LIMITS_DISABLE disables the limits.
   2.2 0x01: D_ECD_EVEREST_ENABLE_LIMITS_ENABLE enables the limits.
3. The response to this command is ACK if NEW LIMITS STATE is a valid value. Otherwise the response is NAK.

### 5.59.17 Command for Query Defined Presets 0x75 Sub Op-Code 0x0F

| Query Defined presets command<br>D_EC_EVEREST<br>D_ECS_EVEREST_DEFINED_PRESETS_QRY | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x0F | 0x75 | 0x00 | PRESET GROUP | — |

FPN

1. This command generates an "Extended Reply" (subsection 4.3) of subsubsection 5.59.18.

2. This sub-opcode is used by the Endura/Atlas projects.
3. This command is used to determine which presets are defined on a device.
4. PRESET GROUP indicates which group of 16 presets are being queried. For Spectra III and Spectra IV preset groups can range from 0 to 15 (0xF). Group 0 covers presets 1 → 16, group 1 covers presets 17 → 32, etc.
5. The response to this command is opcode 0x75, sub opcode 0x10.

### 5.59.18 Response to Query Defined Presets 0x75 Sub Op-Code 0x10

| Query Defined presets response D_EC_EVEREST D_ECS_EVEREST_DEFINED_PRESETS_RSP | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x10 | 0x75 | BITMASK MSB | BITMASK LSB | — |

FPN

1. This response is an "Extended Reply" (subsection 4.3).
2. This is the response to opcode 0x75, sub opcode 0x0F.
3. The bits in the BITMASK indicate which presets are defined on the unit. An on bit indicates the preset is defined. An off bit indicates the preset is not defined. The range of presets represented depends on the preset group parameter of the command that is being responded to. The lowest number preset in the BITMASK is calculated as follows:
   (preset_group x 16) +1

### 5.59.19 Command for Query Defined Patterns 0x75 Sub Op-Code 0x11

| Query Defined patterns command D_EC_EVEREST D_ECS_EVEREST_DEFINED_PATTERNS_QRY | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x11 | 0x75 | 0x00 | PATTERN GROUP | — |

FPN

1. This command generates an "Extended Reply" (subsection 4.3) of subsubsection 5.59.20.
2. This sub-opcode is used by the Endura/Atlas projects.
3. This command is used to determine which patterns are defined on a device.
4. PATTERN GROUP indicates which group of 16 patterns are being queried. For Spectra III the only valid pattern group is 0. Since the maximum number of patterns that Spectra allows is 8, only the 8 least significant bits will ever be set to 1.
5. The response to this command is opcode 0x75, sub opcode 0x12.

### 5.59.20 Response to Query Defined Patterns 0x75 Sub Op-Code 0x12

| Query Defined patterns response<br>D_EC_EVEREST<br>D_ECS_EVEREST_DEFINED_PATTERNS_RSP | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x12 | 0x75 | BITMASK MSB | BITMASK LSB | — |

FPN

1. This response is an "Extended Reply" (subsection 4.3).
2. The response for unsupported SUB OP CODEs is NAK.
3. This response sub-opcode is used by the Endura/Atlas projects.
4. This is the response to opcode 0x75, sub opcode 0x11.
5. The bits in the BITMASK indicate which patterns are defined on the unit. An on bit indicates the pattern is defined. An off bit indicates the pattern is not defined. The range of patterns represented depends on the pattern group parameter of the command that is being responded to. The lowest number pattern in the BITMASK is calculated as follows:
   (pattern_group x 16) +1

## 5.60 Command/Responses 0x77 ($119_{10}$), Time Commands

### 5.60.1 Command format

| Time Commands D_EC_TIMESET_MACRO_OPCODE | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 SYNC | 2 ADDR | 3 CMND1 | 4 CMND2 | 5 DATA1 | 6 DATA2 | 7 CKSM |
| | 0xFF | — | TIME SUB OPCODE | 0x77 | 0x00 | 0x00 | — |
| | Section | Page | | | | | |
| Cmnd | 5.60.3 | 86 | 0x00 | 0x77 | Set seconds and synchronize time | | |
| | | | | | seconds | | |
| Cmnd/- | 5.60.4 | 86 | 0x01 | 0x77 | Report seconds | | |
| Resp | | | | | seconds | | |
| Cmnd | 5.60.5 | 87 | 0x02 | 0x77 | Set hour and minutes | | |
| | | | | | hour | minute | |
| Cmnd/- | 5.60.6 | 87 | 0x03 | 0x77 | Report hour and minutes | | |
| Resp | | | | | hour | minute | |
| Cmnd | 5.60.7 | 87 | 0x04 | 0x77 | Set month and date | | |
| | | | | | month | day | |
| Cmnd/- | 5.60.8 | 88 | 0x05 | 0x77 | Report month and date | | |
| Resp | | | | | month | day | |
| Cmnd | 5.60.9 | 88 | 0x06 | 0x77 | Set year | | |
| | | | | | year | | |
| Cmnd/- | 5.60.10 | 89 | 0x07 | 0x77 | Report year | | |
| Resp | | | | | year | | |

### 5.60.2 Description

The sub-opcodes are assigned such that even values (0x00, 0x02, etc.) are time setting commands, and odd value (0x01, 0x03, etc.) are time reporting sub-opcodes.

Set commands return an ACK if successful, or NAK if not. Report commands will return a NAK if time is not set. In order to assure a successful time setting operation, time and date should be sent top-down, that is, year, month-date, hour-minute, seconds. Similarly, a time request will only be guaranteed to be correct if no intervening transmissions occur.

**Note**

1. These commands/replies are only implemented on the Spectra IV.
2. When D_ECS_SET_SECONDS is received, all updated times are copied over into the active time array.
3. If the upper byte of the data field of SET_YEAR is equal to 0 (00), then 2000 is added to the year value.

[82] $Header: 77.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

### 5.60.3 Command 0x77, Sub Op-Code 0x00, Set seconds and synchronize time

| Time Commands, Set seconds and synchronize time<br>D_EC_TIMESET_MACRO_OPCODE<br>D_ECS_SET_SECONDS | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x77 | 0x00 | NEW SECOND | — |

FPN

1. The response to this command is either an ACK or NAK (subsection 5.1) depending on the validity of the arguments.

| 0x00 Set seconds and synchronize time | |
|---|---|
| byte 5 | 0x00 |
| byte 6 | NEW SECOND to set (0-59) |

On receipt of this command, the receiver clock will be set to the time and date previously transmitted. If an unrelated transmission occurs between this command and other time setting commands, no action will be taken. The response to this command is ACK if the seconds value is in range and time is set. It is NAK if out of range or time is not set.

### 5.60.4 Command 0x77, Sub Op-Code 0x01, Report seconds

| Time Commands, Report seconds<br>D_EC_TIMESET_MACRO_OPCODE<br>D_ECS_GET_SECONDS | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x01 | 0x77 | 0x00 | CURRENT SECOND | — |

FPN

1. This command generates an “Extended Response” (subsection 4.3).

When reporting any segment of the clock data, the current time will be read and held in a buffer until an unrelated transmission occurs. Thus no ambiguities or anomalies will be reported if the time requests occur in succession.

### 5.60.5 Command 0x77, Sub Op-Code 0x02, Set hour and minutes

| Time Commands, Set hour and minutes<br>D_EC_TIMESET_MACRO_OPCODE<br>D_ECS_SET_HR_MINS | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x02 | 0x77 | NEW HOUR | NEW MINUTE | — |

FPN

1. The response to this command is either an ACK or NAK (subsection 5.1) depending on the validity of the arguments.

| 0x02 Set hour and minutes | |
|---|---|
| byte 5 | NEW HOUR to set (0-23) |
| byte 6 | NEW MINUTE to set (0-59) |

Time will always be transmitted in 24-hour format. That is, midnight is 00:00, etc.

### 5.60.6 Response 0x77, Sub Op-Code 0x03, Report hour and minutes

| Time Commands, Report hour and minutes<br>D_EC_TIMESET_MACRO_OPCODE<br>D_ECS_GET_HR_MINS | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x03 | 0x77 | CURRENT HOUR | CURRENT MINUTE | — |

FPN

1. This command generates an "Extended Response" (subsection 4.3).

### 5.60.7 Command 0x77, Sub Op-Code 0x04, Set month and date

| Time Commands, Set month and date<br>D_EC_TIMESET_MACRO_OPCODE<br>D_ECS_SET_MON_DATE | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x04 | 0x77 | NEW MONTH | NEW DATE | — |

FPN

1. The response to this command is either an ACK or NAK (subsection 5.1) depending on the validity of the arguments.

| 0x04 Set month and date | |
|---|---|
| byte 5 | NEW MONTH (1-12) |
| byte 6 | NEW DATE (1-31) |

The date will be out of range if not valid for the month specified. If the month specified is February, the range is limited to 1 — 28 unless the year has been determined to be a leap year, in which case 29 is an acceptable value. If the year has not been set, it is assumed that it is not a leap year.

### 5.60.8 Response 0x77, Sub Op-Code 0x05, Report month and date

| Time Commands, Report month and date<br>D_EC_TIMESET_MACRO_OPCODE<br>D_ECS_GET_MON_DATE | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x05 | 0x77 | CURRENT MONTH | CURRENT DATE | — |

FPN

1. This command generates an "Extended Response" (subsection 4.3).

Month and date are reported in the same format as the corresponding set command.

### 5.60.9 Command 0x77, Sub Op-Code 0x06, Set year

| Time Commands, Set Year<br>D_EC_TIMESET_MACRO_OPCODE<br>D_ECS_SET_YEAR | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x06 | 0x77 | YEAR MSB | YEAR LSB | — |

FPN

1. The response to this command is either an ACK or NAK (subsection 5.1) depending on the validity of the arguments.

The year may be sent as a complete value (i.e. 2006) or as the last two digits (i.e. 06.) If only the last two digits are sent, the century value is assumed to be 2000 and is added to the value sent to determine the year.

### 5.60.10 Response 0x77, Sub Op-Code 0x07, Report year

| Time Commands, Report Year<br>D_EC_TIMESET_MACRO_OPCODE<br>D_ECS_GET_YEAR | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x07 | 0x77 | YEAR MSB | YEAR LSB | — |

FPN

1. This command generates an "Extended Response" (subsection 4.3).

The year reported is always the absolute value, that is, 2006 is always sent as 2006, not 06.

## 5.61 Command 0x79 ($121_{10}$), Screen Move

### 5.61.1 Command format

| Screen Move<br>D_EC_SCREEN_MOVE | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | ABS | 0x79 | PAN $\Delta$ | TILT $\Delta$ | — |
| 0xFF | — | REL | 0x79 | PAN $\Delta$ | TILT $\Delta$ | — |

FPN

1. This command generates a "General Reply" (subsection 4.2).
2. This command is used by the Endura/Atlas projects.

### 5.61.2 Description

For screen moves the REL/ABS SUB OPCODE indicates if this is a relative or absolute move.

1. ABS 0x00 (D_ECS_SET_ABS_MAG) for an absolute move
2. REL 0x01 (D_ECS_SET_REL_MAG) for a relative move

The data bytes are PAN $\Delta$ (byte 5): screen relative pan position, TILT $\Delta$ (byte 6): screen relative tilt position.

The pan/tilt positions are percentage of distance from center of screen to the corresponding edge expressed as a signed, 8-bit 2's complement value. Positive values are right for pan and up for tilt. The command will return a standard response with ACK if values are in range, NAK if not.

This command has been added to the D Protocol to support the Atlas project for use with the Spectra IV only.

[83] $Header: 79.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

This is the only 1 of 2 D Protocol commands that has a **`signed int`** as DATA1 and DATA2 which are concatenated together.

## 5.62 Command 0x7B ($123_{10}$), Video Mode

### 5.62.1 Command format

| Video Mode D_EC_VIDEO_MODE | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| 0xFF | — | 0x00 | 0x7B | Bit Field | Bit Field | — |

FPN

1. This command generates either a "General Response" (subsection 4.2) or an "Extended Response" (subsection 4.3). On PTZ units that do not support this command, the reply will be a "General Reply" which indicates the command is not supported. On PTZ units that do support this command, the reply will be a D Protocol ACK or NAK (subsection 5.1). to indicate which features are supported.
2. This command is in P Protocol and Coaxitron® and is identical in operation when received. However P Protocol does not generate the extended or general reply types. And Coaxitron® is a "one-way protocol". Thus in P Protocol and Coaxitron® this is a "write only" command.
3. This command has been implemented in Spectra IV starting with rev 1.09 only when a progressive scan type of camera is installed. These are VK-S434N, VK-S454N and all VK-S654 type cameras[85]. Future versions of the PTZ systems may support additional camera types.
4. A limited sub-set of this function is supported on the Esprit starting with revision 4.04 and following through use of the Esprit's built in menu system only.
5. Not all progressive scan features will be implemented on all models.

| Used/Implemented on | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Spectra II | Spectra III | Spectra Mini | Spectra IV | Esprit IOP | Esprit TI | ExSite |
| Decoded | No | No | No | 1.09 | No | No | No |
| In Coaxitron® | No | No | No | 1.09 | No | No | No |
| In P | No | No | No | 1.09 | No | No | No |
| General | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Extended | No | No | No | 1.09 | No | No | No |
| Standard | No | No | No | 1.09 | No | No | No |

### 5.62.2 Description

1. Enables various options for Progressive Scan.

   Decoding of this command is done by concatenating DATA1 and DATA2 together and processing them as a single unsigned 16 bit integer.

[84] `$Header: 7B.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:12 PM$`
[85] This list is correct in Fall 2008. It is anticipated that in the future additional camera types will be added to the list.

1.1 `0x0000` `VIDEO_MODE_GET_INFO` Request the current and supported modes without modifying them. See the Video ACK Response, item 2, below for details.

1.2 `0x0001` `VIDEO_MODE_SET_INTERLACED` Request to set the video mode to interlaced sensor scan with interlaced video output.

1.2.1. If this mode is supported the camera will set the mode to interlaced then send a Video Mode ACK reply with updated current mode data (see Video Mode ACK Response, item 2, below).

1.2.2. If this mode is not supported for the camera a NAK will be sent.
Note: video will appear smoothest on pure analog devices and devices not supporting field aligned, progressive scan, segmented frame output when set to either `VIDEO_MODE_INTERLACED` or `VIDEO_MODE_PROGRESSIVE_NOT_ALIGNED` video modes.

1.3 `0x0002` `VIDEO_MODE_SET_PROGRESSIVE_NOT_ALIGNED` Request to set the video mode to progressive sensor scan without field aligned segmented frames.

1.3.1. If this mode is supported the camera will set the mode to interlaced then send a Video Mode ACK Response with updated current mode data (see Video Mode ACK Response below).

1.3.2. If this mode is not supported for the camera a NAK will be sent.
Note: video will appear smoothest on pure analog devices and devices not supporting field aligned, progressive scan, segmented frame output when set to either `VIDEO_MODE_INTERLACED` or `VIDEO_MODE_PROGRESSIVE_NOT_ALIGNED` video modes.

1.4 `0x0004` `VIDEO_MODE_SET_PROGRESSIVE_UPPER_ALIGNED` Request to set the video mode to progressive sensor scan with upper field aligned segmented frames.

Digital devices encoding the video, that always adhere to the temporal order of the fields (upper → lower) when combining them into frames, will get frames that were progressively read from the sensor. When combined properly, digital devices will have the benefit of frames that were read at one time period, thus removing jagged edges seen from objects moving horizontally across the video without progressively scanned sensor data with field aligned output.

Note: For maximum resolution digital devices encoding the video must disable any "deinterlacing" algorithms when using this mode.

1.4.1. If this mode is supported the camera will set the mode to interlaced then send a Video Mode ACK Response with updated current mode data (see Video Mode ACK Response, below).

1.4.2. If this mode is not supported for the camera a NAK will be sent.

1.5 `0x0008` `VIDEO_MODE_SET_PROGRESSIVE_LOWER_ALIGNED` Request to set the video mode to progressive sensor scan with lower field aligned segmented frames.

Digital devices encoding the video, that always adhere to the temporal order of the fields (lower → upper) when combining them into frames, will get frames that were progressively read from the sensor. When combined properly, digital devices will have the benefit of frames that were read at one time period, thus removing jagged edges seen from objects moving horizontally across the video without progressively scanned sensor data with field aligned output.

Note: For maximum resolution digital devices encoding the video must disable any "deinterlacing" algorithms when using this mode.

1.5.1. If this mode is supported the camera will set the mode to interlaced then send a Video Mode ACK Response with updated current mode data (see Video Mode ACK Response, below).

1.5.2. If this mode is not supported for the camera a NAK will be sent.

2. Responses to this command are as follows:

2.1 Video Mode ACK Response

This response uses the two data bytes to convey information about the installed camera capabilities. DATA1 reports the current video mode(s) and DATA2 reports the capabilities of the camera.

Video modes supported (by bit):

2.1.1. `0x01` Interlaced

2.1.2. `0x02` Unaligned Progressive Scan

2.1.3. `0x04` Upper Field Aligned Progressive Scan

2.1.4. `0x08` Lower Field Aligned Progressive Scan

Thus if DATA1 and DATA2 are equal to `0x0103` that would indicate that the camera is running in interlaced mode and supports interlaced and unaligned progressive scan modes.

| **Video Mode ACK Response D_EC_STD_RESPONSE** | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| `0xFF` | — | `0x01` | `0x01` | Bit Field | Bit Field | — |

2.2 Video Mode NAK Response

2.2.1. Video Mode NAK uses standard NAK response with no additional data in the data bytes.

2.2.2. Video Mode NAK is issued when a `VIDEO_MODE_SET` request is not a supported mode or if multiple bits (modes) were set in the command.

| **Video Mode NAK Response D_EC_STD_RESPONSE** | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| SYNC | ADDR | CMND1 | CMND2 | DATA1 | DATA2 | CKSM |
| `0xFF` | — | `0x00` | `0x01` | `0x00` | `0x00` | — |

### 5.62.3 Pelco Questions to Hitachi about these Video Modes

```
From:  Springer, Derek
Sent:  Tuesday, September 30, 2008 11:35 AM
```

Springer, Derek wrote:
Steve,
I'm a little confused about the differences between the 4 different modes available,

1. Interlaced Signal Process Mode,
2. Progressive Signal Process Mode with frame output turned off.
3. Progressive Signal Process Mode with Even (aligned) segmented frame output,
4. Progressive Signal Process Mode with Odd (aligned) segmented frame output, and

I believe I understand all the modes except the last one above, Progressive Signal Process Mode with Frame Output turned off.

The attached pdf has a representation of what I believe the video signal looks like for Progressive Scan with Even alignment, and Interlaced Scan.

I'm not clear what the representation would be for Progressive Scan with Frame Output set to 'Off'.

Would it look like the interlaced scan diagram with 16ms (NTSC) between fields? If so, what is happening with the progressively processed signal information?

Or in this mode is the alignment just not guaranteed (and it may switch back and forth at any given time) but frames (i.e. fields with no time delta between them) do exist in the video output.

Let me know if you, or one of your colleagues, can shed some light on the situation.

Thanks,
Derek Springer
Software Engineer

### 5.62.4 Answer from Hitachi about these Video Modes

<Edited somewhat>

```
From:  Steve Hodgman [mailto:steve.hodgman@hhea.hitachi.com]
Sent:  Wednesday, October 01, 2008 1:47 PM
To:  Springer, Derek
Cc:  Kevin Komatsu
Subject:  Re:  Frame Output Modes (in Progressive Signal Process Mode), OFF
Vs.  Odd/Even Vs Interlaced Signal Process Mode
```

Hi Derek,

OK, I will try to explain the differences among the various output options with our 35X VK-S654 series and the new 23X VK-S454N series cameras.

1. **Signal Processing Modes**

   1.1 **Interlace Signal Processing Mode** (Not available on the Esprit) Figure 2 block marked "Interlaced Sensor Processing".

   In the interlaced processing mode the CCD is operated in interlace scanning mode.

   1.1.1. For each of the "CCD Field" time periods, the CCD will output half of the lines in a combined format.

   1.1.2. For the Odd CCD "field", lines 1 & 2 will be combined, lines 3 & 4, 5 & 6, and so on.

   1.1.3. For the Even CCD "field", lines 2 & 3 will be combined, 4 & 5, 6 & 7, and so on.

   The actual combination takes place during the next "field" when the pixel charges accumulated during the previous "field" are output from the CCD via the horizontal and vertical shift registers.

   In the interlaced processing mode the DSP IC and all of the algorithms will function using only half of the CCD lines during each "field" time period.

   The benefit of interlace processing mode is a theoretical 6db increase in camera sensitivity due to the CCD line combination in interlaced scanning mode.

   1.2 **Progressive Signal Processing Mode** (Default mode for the Esprit) Figure 2 block marked "Progressive Sensor Processing".

   In the progressive processing mode the CCD is operated in progressive scanning mode.

   1.2.1. For each of the "CCD Field" time periods, the CCD will output all of the lines in a separated format. All CCD lines are output in order from 1, 2, 3, 4, and so on exactly the same for each "field".

   1.2.2. In the progressive processing mode the DSP IC and all of the algorithms will function using all of the CCD lines during each "field" time period. The additional CCD lines in progressive processing mode enable more accurate algorithm performance and produce better video.

2. **Output Options**

   As you know, in order to maintain compatibility with all of the installed standard interlaced video monitors, **all of our analog cameras will always output an interlaced signal regardless of internal signal processing mode**.

2.1 **Interlaced Signal Processing Mode** (Not available in the Esprit) Figure 2 block marked "Video Frame Output Off".

In this mode the camera output will follow the interlace CCD output signal. So the Odd interlaced camera output field signal will consist of the CCD lines 1 & 2 combined, lines 3 & 4 combined, 5 & 6, and so on. The Even interlaced camera output field signal will consist of the CCD lines 2 & 3 combined, 4 & 5 combined, 6 & 7, and so on.

2.2 **Progressive Signal Processing Mode**

2.2.1. **Frame Output mode OFF** Figure 2 block marked "Video Frame Output Off".
In this mode all of the CCD lines are processed by the DSP IC and are available at the final progressive-to-interlace converter block.

2.2.1.1. The Odd interlaced camera output field will consist of the Odd CCD lines, 1, 3, 5, 7, and so on from the "first" CCD output "field".

2.2.1.2. The Even interlaced camera output field will consist of the Even CCD lines, 2, 4, 6, 8, and so on, from the "next" CCD output "field".

2.2.1.3. The remaining CCD output lines from each CCD "field" are not output. Whether the "first" CCD output "field" is the "Odd CCD field" or the "Even CCD field" will depend upon the camera mode and the setting in register `$FF54/$144F`.

2.2.1.4. In normal camera output mode the "first" CCD output "field" will be the "Odd CCD field" while the "next" CCD output "field" will be the "Even CCD field".

In the Frame Output, WDR, DSS, Freeze, Reverse, or Mirror modes, the order will be selected by the value in register `$FF54/$144F`.

2.2.2. **Frame Output mode ON — Odd Field**
**Frame Output mode ON — Even Field** Figure 2 block marked "Video Frame Output Odd" and "Video Frame Output Even".
In the Frame Output mode both camera interlaced output fields will come from the same processed CCD "field" output.

2.2.2.1. So the Odd interlaced camera output field will consist of the Odd CCD lines, 1, 3, 5, 7, and so on from the "first" CCD output "field".

2.2.2.2. The Even interlaced camera output field will consist of the Even CCD lines, 2, 4, 6, 8, and so on, from the exact same CCD output "field".

2.2.2.3. The signal from the next CCD output is not output. So you are only seeing the video information from every other CCD "field". Since each camera interlaced output field originated from the same CCD "field" (i.e. the same moment in time), there will be no position displacement between the same objects in each "field" image.

2.2.2.4. If this interlaced output frame is captured by a "full frame" video capture device or "frame" DVR/NVR, the frame stills will be perfectly motionless and clear with double the vertical resolution, even if there was motion in the captured scene. Of course the capture device/DVR/NVR must match the Odd/Even field order or the motion artifacts will be twice as bad.

2.2.2.5. The Odd/Even field order is again selectable by the value in register `$FF54/$144F`.

3. It is somewhat difficult to explain/understand what is happening with the different modes of a camera that has Interlace mode, Progressive mode (with interlaced output), and our Frame Output mode (with field order selection). Also, that is why I put the word "field" in quotes when referring to a progressive scan CCD. The word "field" will have different meanings depending for a progressive scan CCD and an interlace output camera and video monitors.

Anyhow, look this stuff over and let me know if you have any follow up questions.
Regards,
Steve

```
Steve Hodgman
Technical Support Manager
Hitachi America LTD - Home Electronics
Visual Technologies Division
101 West Street - Suite 19
Hillsdale, NJ 07642-1421
(201) 664-9306 (phone)
(800) 526-6241 (toll free)
(201) 664-6143 (fax)
E-Mail:  Steve.Hodgman@hhea.hitachi.com
```

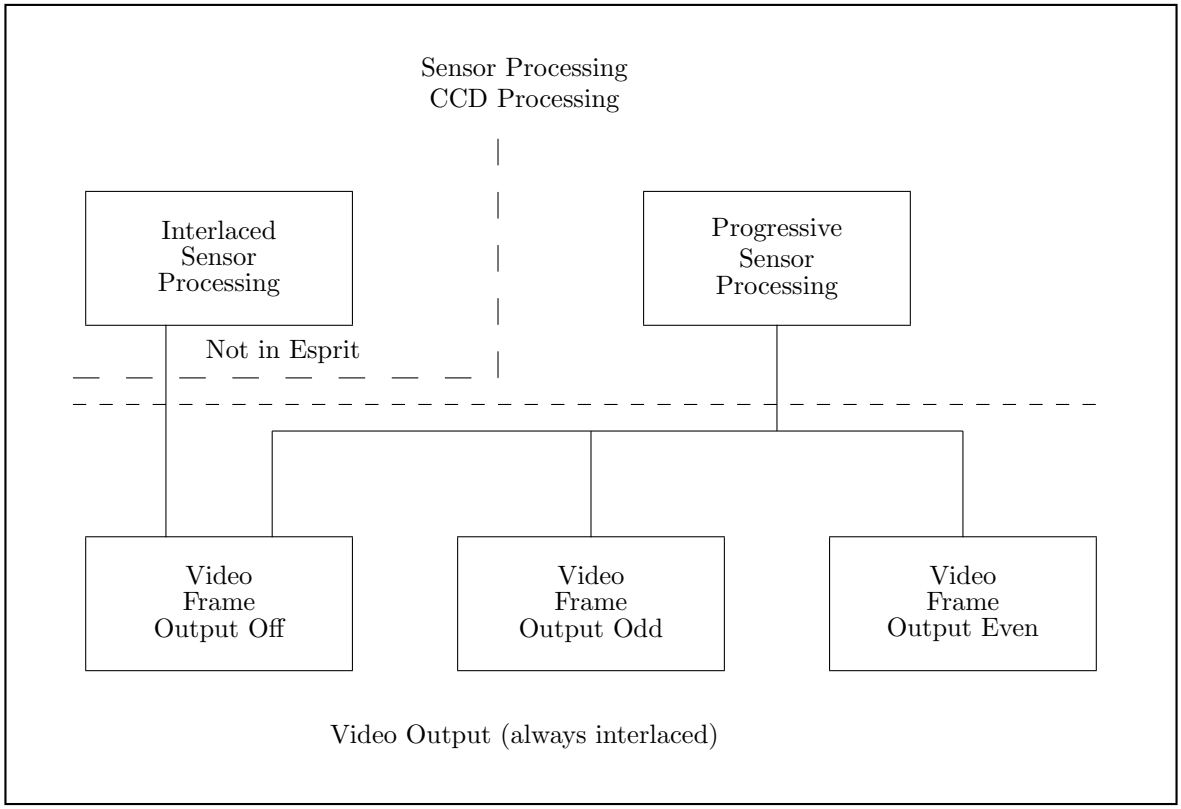
### 5.62.5 Understanding Interlaced/Progressive modes

The exact interlationship between Interlaced and Progressive (and more recently Normal and High Sensitivity[86] ) modes of camera operation is beyond the scope of a protocol document. The following Hitachi Confidential Proprietary documents describe the various features of these modes as the modes apply to each of their camera modules. Hitachi frequently updates the camera module software and so only the information contained in these documents should be used for using these various camera module modes.

When searching Hitachi manuals, look for the following hexadecimal addresses: `1221`, `144F` and `FF54`. Hitachi changed the names of these items from time to time.

| Manual Name | Page |
|---|---|
| `434N_NewComm_E10` | 17 |
| `454N_NewComm_E11` | 17 |
| `P554_Comm_E14` | 54 |
| `P554_NewComm_E12` | 33 |
| `S624N_Comm_E10` | 75 |
| `S624N_NewComm_E10` | 32, 45 |
| `S654_Comm_E131` | 59 |
| `S654_NewComm_E12` | 34 |
| `S654N_Comm_E13` | 83 |
| `S654N_NewComm_E13` | 34, 46 |
| `S654R_Comm_E10_R1` | 55 |
| `S654R_NewComm_E11` | 42 |

[86] Starting with the VK-S654N camera series the terms Interlaced and Progressive were renamed to become High-Sensitivity and Normal modes. (Or it may be the other way around.)




$RCSfile: IntProg.inc,v $

Figure 2: Interlace/Progressive Mode Processing and Video Output

# A Software Revision Reporting

Different versions of the Spectra/Esprit/Exsite series of integrated PTZ units have responded in different ways to query commands for their make, model number and software revision data requests.

## A.1 Spectra II

This is the older format of reply to the query command. In this format the reply consists of the program number which is null padded. It does not give an extended reply to additional software build data.

```
 1  $Header: S2Rev.dat: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:18 AM$
 2  Spectra II, Rev S3.31
 3  DCE      34    11.462325  ff 01 00 45 00 00 46  Part number query
 4                                P  G  5  3  -  0  0  6  0  -  S  3  3  1
 5   DTE     41    11.492682  ff 01 50 47 35 33 2d 30 30 36 30 2d 53 33 33 31 00 50
 6  DCE      59    12.590205  ff 01 01 45 00 00 47  Serial number query
 7                                P  G  5  3  -  0  0  6  0  -  S  3  3  1
 8   DTE     66    12.621544  ff 01 50 47 35 33 2d 30 30 36 30 2d 53 33 33 31 00 51
 9  DCE      84    13.718040  ff 01 00 73 00 00 74  Software rev query
10   DTE     91    13.748785  ff 01 00 74
11  DCE      95    14.846040  ff 01 02 73 00 00 76  Build query
12   DTE    102    14.877727  ff 01 00 76
```

## A.2 Spectra III

In this reply to the query command it begins to respond with its marketing model number and repeats its marketing name instead of a serial number both of which are null padded. Additonal information is available in the special software rev fields. However the range of the software rev is in the format `n.nn`. Thus the rev usually fits in one byte of return data.

```
 1  $Header: S3Rev.dat: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:19 AM$
 2  Spectra III, Rev 1.34
 3  DCE      34    11.462343  ff 01 00 45 00 00 46  Part number query
 4                                D  D  5  3  C  2  2
 5   DTE     41    11.492664  ff 01 44 44 35 33 43 32 32 00 00 00 00 00 00 00 00 de
 6  DCE      59    12.590233  ff 01 01 45 00 00 47  Serial number query
 7                                D  D  5  3  C  2  2
 8   DTE     66    12.620557  ff 01 44 44 35 33 43 32 32 00 00 00 00 00 00 00 00 df
 9  DCE      84    13.718149  ff 01 00 73 00 00 74  Software rev query
10   DTE     91    13.747879  ff 01 01 73 00 86 fb  1.34
11  DCE      98    14.846066  ff 01 02 73 00 00 76  Build query
12   DTE    105    14.875918  ff 01 03 73 00 00 77  0
```

## A.3 Spectra MINI

The reply to the query command consists of its marketing name (this is a PAL type of unit) which is null padded and it does not have a serial number to give out. It does not give an extended reply to additional software build data.

```
 1  $Header: MRev.dat: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:14 AM$
 2  Spectra Mini, Rev 1.23
 3  DCE      12     1.100139  ff 01 00 45 00 00 46  Part number query
 4                                M  I  N  I     1  1  4  R  -  X
```

[87] $Header: Revs.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:13 PM$

```
 5   DTE     19     1.114932  ff 01 4d 49 4e 49 20 31 31 34 52 2d 58 00 00 00 00 01
 6  DCE      37     2.251495  ff 01 01 45 00 00 47  Serial number query
 7                                 M  I  N  I     1  1  4  R  -  X
 8   DTE     44     2.266214  ff 01 4d 49 4e 49 20 31 31 34 52 2d 58 00 00 00 00 02
 9  DCE     185    12.619487  ff 01 00 73 00 00 74  Software rev query
10   DTE    192    12.634248  ff 01 00 74
11  DCE     196    13.770828  ff 01 02 73 00 00 76  Build query
12   DTE    203    13.785538  ff 01 00 76
```

The software rev data is read out using the diagnostic information query (`0x6F` subsection 5.56) opcode: (This data comes from a different dome than the previous readout.)

```
 1  $Header: MRev71.dat: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:14 AM$
 2  Spectra Mini, Rev 1.23
 3  DCE    1875   812.401611 ff 01 00 6f 00 00 70 Diagnostic Information Query
 4                                  1  .  2  3
 5   DTE   1882   812.409717 ff 01 00 71 60 01 31 2e 32 33 97
```

## A.4 Spectra IV

The reply to the query command consists of its marketing name which is null padded and a serial number. In this case the software rev number has been changed to have a format of `n.nnn` and starts to take up two bytes of data.

```
 1  $Header: S4Rev.dat: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:20 AM$
 2  Spectra IV, Rev 1.050
 3  DCE      34    11.470258  ff 01 00 45 00 00 46  Part number query
 4                                 D  D  4  C  B  W  3  5
 5   DTE     41    11.500216  ff 01 44 44 34 43 42 57 33 35 00 00 00 00 00 00 00 47
 6  DCE      59    12.598173  ff 01 01 45 00 00 47  Serial number query
 7                                 5  7  7  9  8  4  2
 8   DTE     66    12.628027  ff 01 35 37 37 39 38 34 32 00 00 00 00 00 00 00 00 c2
 9  DCE      84    13.726064  ff 01 00 73 00 00 74  Software rev query
10   DTE     91    13.756231  ff 01 01 73 04 1a 93  1.050
11  DCE      98    14.854007  ff 01 02 73 00 00 76  Build query
12   DTE    105    14.884018  ff 01 03 73 00 01 78  1
```

## A.5 Spectra IV TC16

The reply to the query command consists of its marketing name which is null padded and a serial number. In this case the software rev number has been changed to have a format of `n.nnn` and starts to take up two bytes of data.

```
 1  $Header: s4tc16.dat: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:20 AM$
 2  Spectra IV, TC-16, Rev 1.07
 3   DTE     34    10.365612  ff 01 00 45 00 00 46  Part number query
 4                                 D  D  4  T  C  1  6
 5  DCE      41    10.394552  ff 01 44 44 34 54 43 31 36 00 00 00 00 00 00 00 00 01
 6   DTE     59    11.518137  ff 01 01 45 00 00 47  Serial number query
 7                                 7  4  5  6  7  4  7
 8  DCE      66    11.547364  ff 01 37 34 35 36 37 34 37 00 00 00 00 00 00 00 00 c0
 9   DTE     84    12.670377  ff 01 00 73 00 00 74  Software rev query
10  DCE      91    12.699736  ff 01 01 73 04 2e a7  1.070
11   DTE     98    13.822615  ff 01 02 73 00 00 76  Build query
12  DCE     105    13.851712  ff 01 03 73 00 01 78  1
```

## A.6 Spectra IV Horizon

The reply to the query command consists of its marketing name which is null padded and a serial number. In this case the software rev number has been changed to have a format of `n.nnn` and starts to take up two bytes of data.

```
 1  $Header: s4h.dat: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:19 AM$
 2  Spectra IV Horizon, rev 1.071
 3   DTE     34    10.430965  ff 01 00 45 00 00 46  Part number query
 4                                D  D  4  H  3  5
 5  DCE      41    10.438667  ff 01 44 44 34 48 33 35 00 00 00 00 00 00 00 00 b3
 6   DTE     59    11.583296  ff 01 01 45 00 00 47  Serial number query
 7                                8  4  5  4  0  8  0
 8  DCE      66    11.591231  ff 01 38 34 35 34 30 38 30 00 00 00 00 00 00 00 b5
 9   DTE     84    12.735633  ff 01 00 73 00 00 74  Software rev query
10  DCE      91    12.743471  ff 01 01 73 04 2f a8  1.071
11   DTE     98    13.886954  ff 01 02 73 00 00 76  Build query
12  DCE     105    13.894669  ff 01 03 73 00 01 78  1
```

## A.7 Esprit

### A.7.1 Esprit CBW24 PAL

In this reply to the query command is the marketing name which is blank padded and repeats its marketing name instead of a serial number. For the software revison commands, it returns with an Endura NAK to indicate that it does not support the command.

```
 1  $Header: EsRev.dat: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:09 AM$
 2  Esprit 24X, Rev 3.36 PAL, With wiper
 3  DCE      34    10.558229  ff 01 00 45 00 00 46  Part number query
 4                                E  S  3  1  C  B  W  2  4  X
 5   DTE     41    10.588681  ff 01 45 53 33 31 43 42 57 32 34 58 20 20 20 20 20 7d
 6  DCE      59    11.709581  ff 01 01 45 00 00 47  Serial number query
 7                                E  S  3  1  C  B  W  2  4  X
 8   DTE     66    11.742013  ff 01 45 53 33 31 43 42 57 32 34 58 20 20 20 20 20 7e
 9  DCE      84    12.862889  ff 01 00 73 00 00 74  Software rev query
10   DTE     91    12.892847  ff 01 00 01 00 00 02  Endura Nak
11  DCE      98    14.014217  ff 01 02 73 00 00 76  Build query
12   DTE    105    14.045763  ff 01 00 01 00 00 02  Endura Nak
```

### A.7.2 Esprit CBW NTSC

This is similar to the older Esprit response (subsubsection A.7.1), however it has an additional "`a`" character to indicate that this 23X camera is a newer model (type "`N`") with more lines of resolution. Also additional information is available starting at rev Esprit 3.86. The additional information is the camera data with DIP switch settings and an SMR number if applicable.

```
 1  # Esprit 23X NTSC IOP rev 3.86, SMR 1-11UXTS
 2  # $Header: ES18a.dat: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:08 AM$
 3   DTE      1     0.000000 ff 01 00 00 00 00 01
 4  DCE       8     0.033389 ff 01 00 01
 5   DTE     12     1.027500 ff 01 00 45 00 00 46 Part number query
 6                               E  S  3  1  C  B  W  1  8  X
 7  DCE      19     1.065135 ff 01 45 53 33 31 43 42 57 31 38 61 20 20 20 20 20 89
 8   DTE     37     2.053855 ff 01 01 45 00 00 47 Serial number query
 9                               E  S  3  1  C  B  W  1  8  X
```

```
10  DCE     44     2.087545 ff 01 45 53 33 31 43 42 57 31 38 61 20 20 20 20 20 8a
11   DTE    62     3.080182 ff 01 02 45 00 00 48 Camera and DIP switch query
12                              E  0  0  5  -  0  1  0  4     0  0     0  1
13  DCE     69     3.116516 ff 01 45 30 30 35 2d 30 31 30 34 20 30 30 20 30 31 16
14   DTE    87     4.106563 ff 01 03 45 00 00 49 SMR, if any, query
15                              S  M  R     1  -  1  1  U  X  T  5
16  DCE     94     4.143677 ff 01 53 4d 52 20 31 2d 31 31 55 58 54 35 20 20 20 b2
17   DTE   162     7.186588 ff 01 00 73 00 00 74 Software rev query
18  DCE    169     7.220552 ff 01 01 73 01 82 f8 3.86
19   DTE   176     8.215364 ff 01 02 73 00 00 76 Software build query
20  DCE    183     8.271880 ff 01 00 01 00 00 02 Endura NAK reply
21   DTE   190     9.244689 ff 01 00 6f 00 00 70 Mini-Spectra software query
22  DCE    197     9.274875 ff 01 00 70          General response
```

### A.7.3 Esprit CBW PAL

This is similar to the older Esprit response (subsubsection A.7.1), however it has an additional "a" character to indicate that this 23X camera is a newer model (type "N") with more lines of resolution. Also additional information is available starting at rev Esprit 3.86. The additional information is the camera data with DIP switch settings and an SMR number if applicable.

```
 1  # Esprit 23X PAL IOP rev 3.86, SMR 1-11UXTS
 2  # $Header: ES18xa.dat: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:08 AM$
 3   DTE    12     1.027267 ff 01 00 45 00 00 46 Part number query
 4                              E  S  3  1  C  B  W  1  8  X  a
 5  DCE     19     1.056938 ff 01 45 53 33 31 43 42 57 31 38 58 61 20 20 20 20 c1
 6   DTE    37     2.054585 ff 01 01 45 00 00 47 Serial number query
 7                              E  S  3  1  C  B  W  1  8  X  a
 8  DCE     44     2.089324 ff 01 45 53 33 31 43 42 57 31 38 58 61 20 20 20 20 c2
 9   DTE    62     3.081902 ff 01 02 45 00 00 48 Camera type and DIP switch query
10                              E  0  1  5  -  0  1  0  4     0  0     0  1
11  DCE     69     3.114570 ff 01 45 30 31 35 2d 30 31 30 34 20 30 30 20 30 31 17
12   DTE    87     4.109273 ff 01 03 45 00 00 49 SMR number query, if there
13                              S  M  R     1  -  1  1  U  X  T  5
14  DCE     94     4.142982 ff 01 53 4d 52 20 31 2d 31 31 55 58 54 35 20 20 20 b2
15   DTE   162     7.190236 ff 01 00 73 00 00 74 Software rev query
16  DCE    169     7.224439 ff 01 01 73 01 82 f8 3.86
17   DTE   176     8.217060 ff 01 02 73 00 00 76 Software build query, no reply
18   DTE   183     9.244398 ff 01 00 6f 00 00 70 Mini-Spectra Software rev query
19  DCE    190     9.277514 ff 01 00 70          General response reply
```

### A.7.4 Esprit CBW PAL

This is similar to the older Esprit response (subsubsection A.7.1), Additional information is available starting at rev Esprit 3.86. The additional information is the camera data with DIP switch settings and an SMR number if applicable.

```
 1  # Esprit 24X PAL IOP rev 3.86, SMR 1-11UXTS
 2  # $Header: Es24x.dat: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:08 AM$
 3   DTE    12     1.007127 ff 01 00 45 00 00 46
 4                              E  S  3  1  C  B  W  2  4  X
 5  DCE     19     1.016280 ff 01 45 53 33 31 43 42 57 32 34 58 20 20 20 20 20 7d
 6   DTE    37     2.013908 ff 01 01 45 00 00 47
 7                              E  S  3  1  C  B  W  2  4  X
 8  DCE     44     2.025306 ff 01 45 53 33 31 43 42 57 32 34 58 20 20 20 20 20 7e
 9   DTE    62     3.021745 ff 01 02 45 00 00 48
10                              1  0  1  1  -  0  1  0  5     0  2     0  1
```

```
11  DCE     69     3.033189 ff 01 31 30 31 31 2d 30 31 30 35 20 30 32 20 30 31 02
12   DTE    87     4.027591 ff 01 03 45 00 00 49
13                                  S  M  R     1  -  1  1  U  X  T  5
14  DCE     94     4.039407 ff 01 53 4d 52 20 31 2d 31 31 55 58 54 35 20 20 20 b2
15   DTE   162     7.049995 ff 01 00 73 00 00 74
16  DCE    169     7.058781 ff 01 01 73 01 82 f8
17   DTE   176     8.058290 ff 01 02 73 00 00 76
18  DCE    183     8.066562 ff 01 00 01 00 00 02
19   DTE   190     9.067144 ff 01 00 6f 00 00 70
20  DCE    197     9.075385 ff 01 00 70
```

## A.8 Esprit TI

In this reply to the query command is the marketing name which is null padded and it sends out its serial number. For the software revison commands, it returns with an Endura NAK to indicate that it does not support the command.

```
 1  $Header: TIRev.dat: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:23 AM$
 2  Esprit TI, Rev unknown
 3  DCE      34    11.463211  ff 01 00 45 00 00 46  Part number query
 4                                E  S  3  0  1  4  T  I
 5   DTE     41    11.497948  ff 01 45 53 33 30 31 34 54 49 00 00 00 00 00 00 00 44
 6  DCE      59    12.591196  ff 01 01 45 00 00 47  Serial number query
 7                                6  7  2  4  3  7  0
 8   DTE     66    12.621723  ff 01 36 37 32 34 33 37 30 00 00 00 00 00 00 00 00 b5
 9  DCE      84    13.711300  ff 01 00 73 00 00 74  Software rev query
10   DTE     91    13.745056  ff 01 00 01 00 00 02  Endura Nak
11  DCE      98    14.831403  ff 01 02 73 00 00 76  Build query
12   DTE    105    14.862998  ff 01 00 01 00 00 02  Endura Nak
```

## A.9 ExSite

The ExSite responds with its marketing model number which is blank padded, and does not have a serial number to respond with. The software rev information is the same as that provided by the Spectra III.

```
 1  $Header: ExRev.dat: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:09 AM$
 2  ExSite, Rev 1.13
 3  DCE      34    10.462784  ff 01 00 45 00 00 46  Part number query
 4                                I  P  S  X  M  3  0  C  2  2
 5   DTE     41    10.492636  ff 01 49 50 53 58 4d 33 30 43 32 32 20 20 20 20 20 82
 6  DCE      59    11.582862  ff 01 01 45 00 00 47  Serial number query
 7                                I  P  S  X  M  3  0  C  2  2
 8   DTE     66    11.612609  ff 01 49 50 53 58 4d 33 30 43 32 32 20 20 20 20 20 83
 9  DCE      84    12.702991  ff 01 00 73 00 00 74  Software rev query
10   DTE     91    12.733001  ff 01 01 73 00 71 e6  1.13
11  DCE      98    13.823069  ff 01 02 73 00 00 76  Build query
12   DTE    105    13.852974  ff 01 03 73 00 00 77  0
```

## A.10 ExSite, fixed mount, PAL

The Fixed ExSite responds with a modified version of the marketing model number which is blank padded, but does not have a serial number to respond with. This is for an ExSite Fixed mount with a CBW23, PAL camera installed. The software rev information is the same as that provided by the Spectra III.

```
1  $Header: FExRevP.dat: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:10 AM$
2  Fixed ExSite rev 1.13
```

```
 3  DCE      34    10.463435  ff 01 00 45 00 00 46  Part number query
 4                                X  M  3  1  C  B  W  2  3  X
 5   DTE     41    10.493028  ff 01 58 4d 33 31 43 42 57 32 33 58 20 20 20 20 20 89
 6  DCE      59    11.583435  ff 01 01 45 00 00 47  Serial number query
 7                                X  M  3  1  C  B  W  2  3  X
 8   DTE     66    11.612976  ff 01 58 4d 33 31 43 42 57 32 33 58 20 20 20 20 20 8a
 9  DCE      84    12.703515  ff 01 00 73 00 00 74  Software rev query
10   DTE     91    12.732950  ff 01 01 73 00 71 e6  1.13
11  DCE      98    13.823617  ff 01 02 73 00 00 76  Built query
12   DTE    105    13.853314  ff 01 03 73 00 00 77  0
```

## A.11 ExSite, fixed mount, NTSC

The Fixed ExSite responds with a modified version of its marketing model number which is blank padded, but does not have a serial number to respond with. This is for an ExSite Fixed mount with a C22, NTSC camera installed. The software rev information is the same as that provided by the Spectra III.

```
 1  $Header: FExRevN.dat: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:10 AM$
 2  Fixed ExSite rev 1.13
 3    DTE     34    10.558073  ff 01 00 45 00 00 46  Part number query
 4                                 X  M  3  0  C  2  2
 5   DCE      41    10.587717  ff 01 58 4d 33 30 43 32 32 20 20 20 20 20 20 20 20 f6
 6    DTE     59    11.710315  ff 01 01 45 00 00 47  Serial number query
 7                                 X  M  3  0  C  2  2
 8   DCE      66    11.739750  ff 01 58 4d 33 30 43 32 32 20 20 20 20 20 20 20 20 f7
 9    DTE     84    12.860571  ff 01 00 73 00 00 74  Software rev query
10   DCE      91    12.890114  ff 01 01 73 00 71 e6  1.13
11    DTE     98    14.014062  ff 01 02 73 00 00 76  Build query
12   DCE     105    14.043369  ff 01 03 73 00 00 77  0
```

# B Camera Characteristics

Although not a part of the protocol, in many places it is convenient to know how a Pelco PTZ units operates in order to understand these protocols.

## B.1 Focusing

The default focus mode is auto mode (the mode after a dome or camera reset). This means that the device (the dome) controls whether auto-focus is turned on or off. There is also an always off mode. The mode is set/changed by a command (subsection 5.22) or menu option.

When in auto mode, the camera starts out with auto-focus on. If the receiver receives a focus near or far command, auto-focus is turned off. It stays off until a pan or tilt command is received or the dome or camera is reset (subsection 5.21) or the unit has panned at least 15° .

## B.2 Iris

The default iris mode is auto mode (the mode after a dome or camera reset). This means that the device (the dome) controls whether auto-iris is turned on or off. There is also an always off mode. The mode is set/changed by a command (subsection 5.23) or menu option.

When in auto mode, the camera starts out with auto-iris on. If the receiver receives an iris open or close command, auto-iris is turned off. It stays off until the device pans or tilts more than 15° from the position where auto iris was turned off or the dome or camera is reset (subsection 5.21).

If the Spectra is in auto mode for auto iris and also in auto mode for AGC (see below), the following happens. If the iris is all the way open and the Spectra receives an iris stop command and then an iris open command within one second of receiving the stop, then AGC is turned off and the camera gain is slowly increased until an iris stop is received or the upper gain limit (subsection 5.34) is increased.

If the gain has been increased by an auto iris command and a close iris command is received, the gain is slowly decreased until it reaches the value it had when the gain started to increase. Then AGC is turned back on and the iris is closed.

## B.3 Speed Ramping

To avoid abrupt speed changes (which could cause clunking noises or even motor stalling), the speeds (angular velocities in degrees/second) are ramped up or down. A command that causes a speed change (such as a motion command or a move to preset) does not set the speed directly. Instead, it sets a desired speed and direction. The ramping task compares the current speed (degrees/second) and direction to the desired speed and direction and calculates a new current speed and direction. This new speed is calculated to keep the angular acceleration (degrees/second/second) approximately constant.

## B.4 Zooming

When zooming in (tele), the camera first does optical zoom, meaning that the elements of the lens move to do the zooming. After reaching maximum optical zoom (the lower zoom limit), the camera then does electronic zooming. Electronic zooming involves no lens movement, until it reaches a preset limit (the higher zoom limit). If the receiver receives a zoom stop command after reaching the lower zoom limit, and then another zoom in command within one second of receiving the stop, the receiver increases electronically to the higher zoom limit.

[88] $Header: CC.inc: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:04 AM$

When zooming out (wide), the camera zooms out through the electronic zoom range and then the optical zoom range. It does not stop at the lower zoom limit.

# C Various Pan and Tilt speeds of Pelco Products

**Note**

1. Internally in Pelco PTZ units, there is a table of speeds *vs.* speed step. Depending on the actual unit, the table may be specified in tenths, hundredth or other format for ° /sec of angular motion. In these tables, the values have been converted to decimal ° /sec.
2. The tables in this section have the actual value of pan/tilt speeds that are selected when that index number is sent to the individual unit.
3. The tables are arranged in eight lines of eight columns with each position in the body of the table representing one indexed speed table value.
4. To aid in reading the hexadecimal value used in D Protocol, following each ° /sec value is its hexadecimal step value.
   4.1 When a speed is represented with one decimal place (x.0) that indicates that the basic table is defined in tenths of a degree.
   4.2 For the Esprit pan speed step 44 (`0x02C`) would give a pan speed of 16.4° /sec.

| | | |
|---|---|---|
| | 16.4 `0x2C` | |

   4.3 When a speed is represented with two decimal places (x.00) that indicates that the basic table is defined in hundredths of a degree.
   4.4 For the Spectra IV pan speed step 44 (`0x2C`) the pan speed will be 13.90° /sec.

| | | |
|---|---|---|
| | 13.90 `0x2C` | |

5. The Pan speed index is in byte 5 of the D and P Protocol command.
6. The Tilt speed index is in byte 6 of the D and P Protocol command.
7. The special case of "pan turbo" speed is not listed in these tables. It is represented by an index value of `0x40` ($64_{10}$) and represents the fastest that the unit may reliably rotate.
8. There is no "tilt turbo" speed. The highest tilt speed is represented by `0x3F` ($63_{10}$). Higher values usually result in "no speed", i.e. a stop.
9. Tilt speeds tend to be about half of the pan speed for each index value.

[89] `$Header: Speeds.inc: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:21 AM$`

## C.1 Spectra IV, Pan and Tilt Speeds

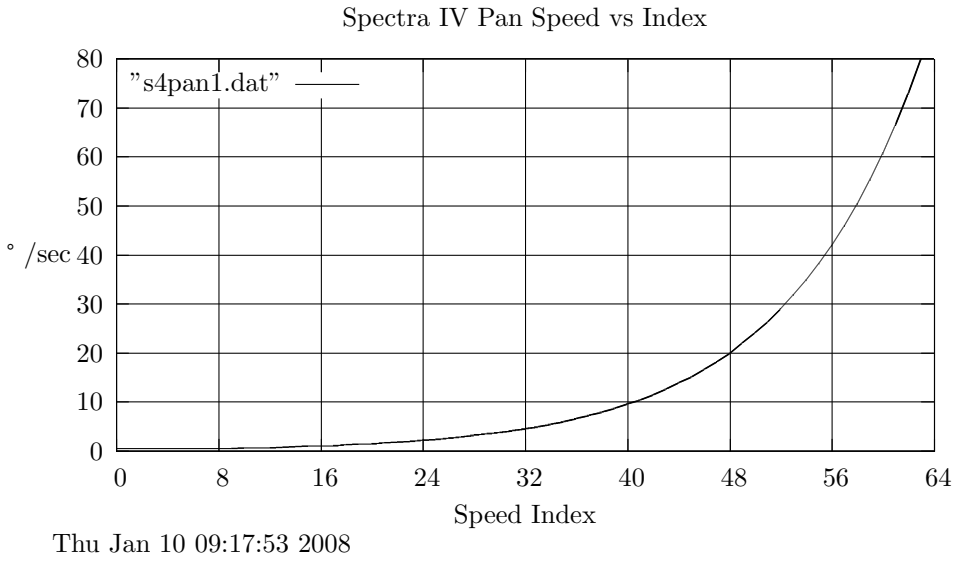



Figure 3: Spectra IV Pan Speeds

| Step | Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.50 0x00 | 0.50 0x01 | 0.50 0x02 | 0.50 0x03 | 0.50 0x04 | 0.50 0x05 | 0.50 0x06 | 0.50 0x07 |
| 8 | 0.50 0x08 | 0.50 0x09 | 0.60 0x0A | 0.70 0x0B | 0.70 0x0C | 0.80 0x0D | 0.90 0x0E | 1.00 0x0F |
| 16 | 1.00 0x10 | 1.10 0x11 | 1.30 0x12 | 1.40 0x13 | 1.50 0x14 | 1.70 0x15 | 1.80 0x16 | 2.00 0x17 |
| 24 | 2.20 0x18 | 2.40 0x19 | 2.60 0x1A | 2.90 0x1B | 3.20 0x1C | 3.50 0x1D | 3.80 0x1E | 4.20 0x1F |
| 32 | 4.62 0x20 | 5.00 0x21 | 5.50 0x22 | 6.00 0x23 | 6.60 0x24 | 7.30 0x25 | 8.00 0x26 | 8.70 0x27 |
| 40 | 9.60 0x28 | 10.50 0x29 | 11.50 0x2A | 12.60 0x2B | 13.90 0x2C | 15.20 0x2D | 16.70 0x2E | 18.30 0x2F |
| 48 | 20.00 0x30 | 22.00 0x31 | 24.10 0x32 | 26.40 0x33 | 29.00 0x34 | 31.80 0x35 | 34.90 0x36 | 38.20 0x37 |
| 56 | 41.90 0x38 | 46.00 0x39 | 50.40 0x3A | 55.30 0x3B | 60.70 0x3C | 66.50 0x3D | 72.90 0x3E | 80.00 0x3F |

Table 1: Spectra IV Pan Speed Table

[90] $Header: S4Speeds.inc: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:20 AM$

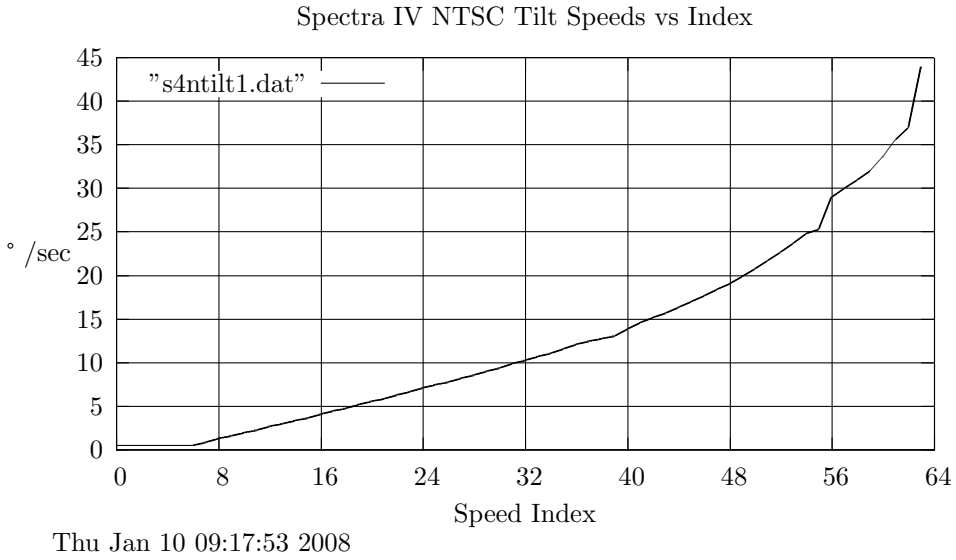



Figure 4: Spectra IV NTSC Tilt Speeds

| Step | Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.50 0x00 | 0.50 0x01 | 0.50 0x02 | 0.50 0x03 | 0.50 0x04 | 0.50 0x05 | 0.50 0x06 | 0.90 0x07 |
| 8 | 1.30 0x08 | 1.60 0x09 | 2.00 0x0A | 2.30 0x0B | 2.70 0x0C | 3.00 0x0D | 3.40 0x0E | 3.70 0x0F |
| 16 | 4.10 0x10 | 4.50 0x11 | 4.80 0x12 | 5.20 0x13 | 5.60 0x14 | 5.90 0x15 | 6.30 0x16 | 6.70 0x17 |
| 24 | 7.10 0x18 | 7.50 0x19 | 7.80 0x1A | 8.20 0x1B | 8.60 0x1C | 9.00 0x1D | 9.40 0x1E | 9.90 0x1F |
| 32 | 10.30 0x20 | 10.70 0x21 | 11.10 0x22 | 11.60 0x23 | 12.10 0x24 | 12.50 0x25 | 12.80 0x26 | 13.10 0x27 |
| 40 | 13.90 0x28 | 14.60 0x29 | 15.20 0x2A | 15.70 0x2B | 16.40 0x2C | 17.00 0x2D | 17.70 0x2E | 18.40 0x2F |
| 48 | 19.10 0x30 | 19.90 0x31 | 20.80 0x32 | 21.70 0x33 | 22.70 0x34 | 23.70 0x35 | 24.80 0x36 | 25.30 0x37 |
| 56 | 29.00 0x38 | 30.00 0x39 | 31.00 0x3A | 32.00 0x3B | 33.60 0x3C | 35.60 0x3D | 37.00 0x3E | 44.00 0x3F |

Table 2: Spectra IV NTSC Tilt Speeds

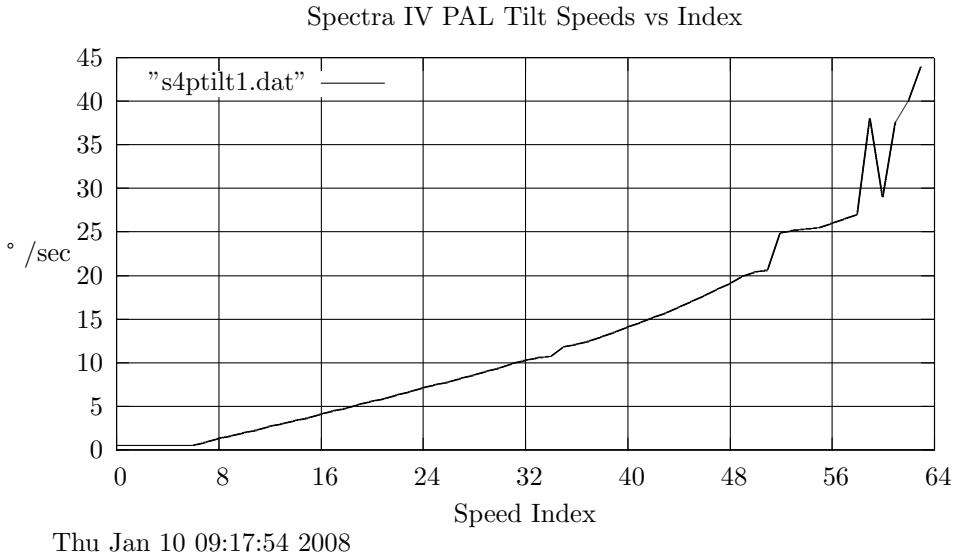



Figure 5: Spectra IV PAL Tilt Speeds

| Step | Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.50 0x00 | 0.50 0x01 | 0.50 0x02 | 0.50 0x03 | 0.50 0x04 | 0.50 0x05 | 0.50 0x06 | 0.90 0x07 |
| 8 | 1.30 0x08 | 1.60 0x09 | 2.00 0x0A | 2.30 0x0B | 2.70 0x0C | 3.00 0x0D | 3.40 0x0E | 3.70 0x0F |
| 16 | 4.10 0x10 | 4.50 0x11 | 4.80 0x12 | 5.20 0x13 | 5.60 0x14 | 5.90 0x15 | 6.30 0x16 | 6.70 0x17 |
| 24 | 7.10 0x18 | 7.50 0x19 | 7.80 0x1A | 8.20 0x1B | 8.60 0x1C | 9.00 0x1D | 9.40 0x1E | 9.90 0x1F |
| 32 | 10.30 0x20 | 10.60 0x21 | 10.70 0x22 | 11.80 0x23 | 12.10 0x24 | 12.50 0x25 | 13.00 0x26 | 13.50 0x27 |
| 40 | 14.10 0x28 | 14.60 0x29 | 15.20 0x2A | 15.70 0x2B | 16.40 0x2C | 17.00 0x2D | 17.70 0x2E | 18.40 0x2F |
| 48 | 19.10 0x30 | 19.90 0x31 | 20.40 0x32 | 20.60 0x33 | 24.90 0x34 | 25.20 0x35 | 25.30 0x36 | 25.50 0x37 |
| 56 | 26.00 0x38 | 26.50 0x39 | 27.00 0x3A | 38.00 0x3B | 29.00 0x3C | 37.60 0x3D | 40.00 0x3E | 44.00 0x3F |

Table 3: Spectra IV PAL Tilt Speeds

## C.2 Esprit, version PG53-0096-0404

In the Esprit, starting with rev 4.04, there is only one speed table. The table is shared between pan and tilt. To get the tilt speed value, take the pan speed value and divide it by 2. This was done to reduce the required memory.

[91] $Header: Estbl4.inc: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:09 AM$

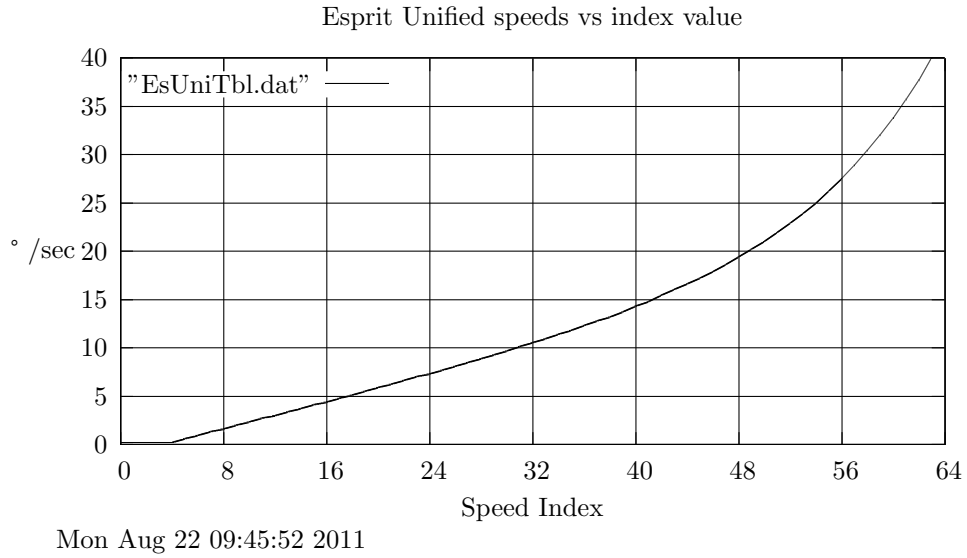



Figure 6: Esprit PG53-0096-0404 Pan/Tilt Speeds

| Step | Value | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | .2 0x00 | .2 0x01 | .2 0x02 | .2 0x03 | .2 0x04 | .6 0x05 | .9 0x06 | 1.3 0x07 |
| 8 | 1.6 0x08 | 2.0 0x09 | 2.3 0x0A | 2.7 0x0B | 3.0 0x0C | 3.4 0x0D | 3.7 0x0E | 4.1 0x0F |
| 16 | 4.4 0x10 | 4.8 0x11 | 5.1 0x12 | 5.5 0x13 | 5.9 0x14 | 6.2 0x15 | 6.6 0x16 | 7.0 0x17 |
| 24 | 7.3 0x18 | 7.7 0x19 | 8.1 0x1A | 8.5 0x1B | 8.9 0x1C | 9.3 0x1D | 9.7 0x1E | 10.1 0x1F |
| 32 | 10.5 0x20 | 10.9 0x21 | 11.4 0x22 | 11.8 0x23 | 12.3 0x24 | 12.8 0x25 | 13.2 0x26 | 13.7 0x27 |
| 40 | 14.3 0x28 | 14.8 0x29 | 15.4 0x2A | 16.0 0x2B | 16.6 0x2C | 17.2 0x2D | 17.9 0x2E | 18.6 0x2F |
| 48 | 19.4 0x30 | 20.2 0x31 | 21.0 0x32 | 21.9 0x33 | 22.9 0x34 | 23.9 0x35 | 25.0 0x36 | 26.2 0x37 |
| 56 | 27.5 0x38 | 28.9 0x39 | 30.4 0x3A | 32.0 0x3B | 33.8 0x3C | 35.7 0x3D | 37.7 0x3E | 40.0 0x3F |

Table 4: Pan and Tilt speeds Esprit, version PG53-0096-0404

# D Interpreting Pan D Readout Replies

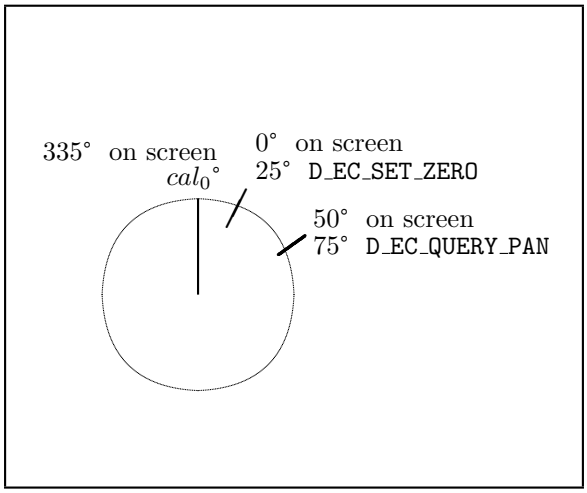



Figure 7: D Protocol Pan Readout Positions

In the below example several variables and functions are used:

1. Protocol Command Values:
   1.1 `SDcmnd1` and `SDcmnd2`, UNSIGNED CHARs are used to hold commands to the Spectra.
2. Calculated Intermediate Values:
   2.1 `HR_offset` is a SIGNED INT which holds the results of asking the Spectra what the Azimuth offset is.
   2.2 `HR_temp` is a SIGNED LONG which holds the result of modifying the reported value from the Spectra by the offset.
3. Macros/defines used:
   3.1 `D_EC_EXTENDED_REPLY_LENGTH` is the length of a D Protocol reply that contains Azimuth or Elevation data. It is currently 7 (seven).
   3.2 `DREPLY_DATA1` with an index value of 5 and
   3.3 `DREPLY_DATA2` which has an index value of 6.
4. Arrays used:
   4.1 `Dreply` is a 7 UNSIGNED CHAR to receive the Spectra reply into. The two positions used here are:
      4.1.1. `DREPLY_DATA1` with an index value of 5 and
      4.1.2. `DREPLY_DATA2` which has an index value of 6.
5. Functions called:

[92] `$Header: PTdata.inc: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:17 AM$`

5.1 `SCheckSumD()`

5.2 `GetDReply()`

6. The results are in two UNSIGNED CHARs:

6.1 `HpanU` this is the upper half of the pan angle when modified by the Set Azimuth Zero value.

6.2 `HpanL` this is the lower half of the pan angle when modified by the Set Azimuth Zero value.

```
// Get pan angle offset from zero
SDcmnd1 = D_ECS_EVEREST_AZIMUTH_ZERO_OFFSET_QRY;
SDcmnd2 = D_EC_EVEREST;      // This is an Everest op-code
SCheckSumD(YES_REPLY);
GetDReply(D_EC_EXTENDED_REPLY_LENGTH);  // Put it in the reply buffer
HR_offset = ((DReply[DREPLY_DATA1]*256) + DReply[DREPLY_DATA2]);

// Get pan angle SDcmnd1, SDdata1 and SData2 don't change anymore
SDcmnd1 = 0x00;
SDcmnd2 = D_EC_QUERY_PAN;   // What is the current azimuth reading
SCheckSumD(YES_REPLY);
GetDReply(D_EC_EXTENDED_REPLY_LENGTH);  // Put it in the reply buffer

// Pan angle comes in in two bytes as degrees times 100 "hungrees"
// Value has to be rounded (i.e. that is why there is a "+ 50" here)
//
// The pan angle reported by an D_EC_QUERY_PAN command is not
// offset by the D_EC_SET_ZERO command. But the on screen display
// is. So here we have to modify the reported output by the
// changed pan offset value.
//
// If D_EC_SET_ZERO has been used to set pan zero to 25 degrees,
// and the on-screen display is now reading 50 degrees in pan,
// then the reply from a D_EC_QUERY_PAN command will be 75 degrees.
// In general we should have the angle reported to the outside
// world match what is seen on the screen. Thus there is logic to
// request the actual offset value and to use that in modifying
// the reported value so that it matches the on-screen value.
//
HR_temp  = ((DReply[DREPLY_DATA1]*256) + DReply[DREPLY_DATA2]);

HR_temp -= HR_offset;           // Get difference of real vs display
if (HR_temp < 0)                // Too small
{
    HR_temp += 36000;           // Yep, let it wrap up
}
HR_temp += 50;                  // Round
HR_temp /= 100;                 // Convert from hungrees to decimal
HpanU    = (unsigned char) (HR_temp/256);
HpanL    = HR_temp & 0xFF;
```

# E Spectra Download Process Used Via The 422/485 Port

10/31/2002
Eric Bopp[94]
Original name = "`sp3 download.doc`"

## E.1 Background

At the start of the process the download application (Windows, Palm or iPaq) does not know whether the Spectra application is running or if the unit is already at the "Ready for download" prompt. So the first step is to make sure we are at the "Ready for download" prompt. Additionally, our specification says that the user should not be able to set the baud used by the download application. This means that the download applilcation will have to test communications at all possible baud rates. The possible initial baud rates are as follows: if the BIOS is running the baud will be 2400, if the application is running the baud could be 2400, 4800 or 9600 (as set by the DIP switches).

## E.2 Getting the unit into "Ready for download" mode

Send a "D" protocol message containing the "download" opcode (`0x57`, `D_EC_DOWNLOAD` subsection 5.44). Cycle through the possible bauds (2400, 4800, 9600) starting with 2400, until a valid response is received. Once we have acknowledgement that this message has been received by the unit, we can be sure that the unit is at the "Ready for download" prompt and that the baud is 2400. (subsubsection E.5.1, lines 14 ⟶ 26 and subsubsection E.5.2, lines 16 ⟶ 26.) (`D_EC_DOWNLOAD` *subsection 5.44 forces a baud rate of 2400 and expects a reply at the original baud rate. I.e. if the Spectra is running at 4800 baud, when it receives an* `D_EC_DOWNLOAD` *subsection 5.44 command it will send a response at 4800 baud and then reconfigure to operate at 2400 baud.*)

| Baud Codes | |
|---|---|
| Value | Baud |
| 0 | 2,400 |
| 1 | 4,800 |
| 2 | 9,600 |
| 3 | 19,200 |
| 4 | 38,400 |
| 5 | 115,200 |

## E.3 New Opcodes

Three new "D" protocol opcodes have been added for the purpose of determining the optimum baud for the download and for starting the download process. The following is a brief description of each opcode.

1. **"Activate Echo Mode"** message (opcode `0x65`, `D_EC_ECHO_MODE` subsection 5.51). Sending this command puts the 422/485 port into a state where any character that is received is immediately retransmitted. The unit automatically comes out of this state when one of the following happens: more than 100 milliseconds pass without a character being received or more than 180 characters having been received. (*This command is sent at the current* `D_EC_SET_BAUD` *subsection 5.52, rate.*)

2. **"Set Baud",** `D_EC_SET_BAUD` message (opcode `0x67` subsection 5.52). Sending this command changes the baud at which the unit communicates at on the 422/485 port. The unit does not change its baud

[93] `$Header: Dnldr.inc: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:05 AM$`
[94] Additions by Eric Hamilton in November 2006

until after it has sent a response to this message. The unit automatically falls back to 2400 baud if no characters are received for 100 milliseconds. (Note that this fallback condition does not apply once the unit begins the download process).

3. **"Start Download",** `D_EC_START_DOWNLOAD` message (opcode `0x69` subsection 5.53). This message starts the download process.

## E.4 Determining the optimum baud for downloading

The next step is to determine the maximum baud that can be reliably used for the download. The Spectra supports 2400, 4800, 9600, 19200, 38400 and 115200 baud rates for downloads via the 422/485 port. The following is the step-by-step process for determining the optimum baud for downloading. The term "recovery state" is used in the description below. "Recovery state" is defined as waiting more than 100 milliseconds and resetting the baud rate to 2400.

1. Start testing the communications link at 2400 baud. (subsubsection E.5.1, lines 27 ⟶ 29, subsubsection E.5.2, lines 28 ⟶ 30.) (*This assumes that the unit is in the "Ready for download" state described in subsection E.2, above.*)

2. Send the "Set Baud" (`0x67`, `D_EC_SET_BAUD` subsection 5.52) message to set the baud to rate to be tested. If a valid response is received, the application should change its baud to the rate being tested and go to step 3. If a valid response is not received go to the recovery state, then resend the "Set Baud" message. Note that retry counts for the purpose of dropping to a lower baud do not need to be implemented for this action because we are always transmitting 2400 baud (the lowest baud). (`D_EC_SET_BAUD` *subsection 5.52 is always sent at 2400 baud and its response is always sent at 2400 baud. This works because a delay of at least 100 milliseconds has occurred since the last data byte transferred. Thus the Spectra has returned to the "recovery state" of 2400 baud.*)

| subsubsection E.5.1 | subsubsection E.5.2 | |
|---|---|---|
| Lines | Lines | |
| 50, 51 | 49, 50 | 68, 69 |
| 58, 59 | 88, 89 | 103, 104 |
| 65, 66 | 116, 117 | 127, 128 |

3. Send the "Activate Echo Mode" message (`0x65`, `D_EC_ECHO_MODE` subsection 5.51). If a valid response is received go to step 4, otherwise increment the retry count, go to the recovery state and do one of the following: (subsubsection E.5.1, lines 31, 32)

   3.1 If the maximum number of retries have occurred then step back to next lower baud and go to step 6.

   3.2 Retry by going back to step 2.

4. Send the test packet. Go to step 5. (subsubsection E.5.1, lines 34 ⟶ 48)

| subsubsection E.5.1 | subsubsection E.5.2 | |
|---|---|---|
| Lines | Lines | |
| 53 ⟶ 57 | 32 ⟶ 48 | 52 ⟶ 66 |
| 61 ⟶ 64 | 71 ⟶ 86 | 91 ⟶ 101 |
| 68 ⟶ 72 | 106 ⟶ 114 | 119 ⟶ 125 |

   A typical 142 byte test packet is (In ASCII):

```
S345000002C0006D20F000CF000000800010FFFFFF00C7FFFFFFFF80FFFFFFFF00FFFFF
FF8FF63F724F00E72007C2470123CE00264070F7401C2E0DB746F2C8687046003960732
```

5. Receive the echoed back message and confirm that there are no errors. Go to the recovery state (this needs to be done regardless of success or failure). If there are no errors in the echo back data: if you are at the maximum baud then go to step 6 otherwise increment to the next baud and go to step 2. If there are errors in the echo back data or a timeout occurs then increment the retry count and do one of the following:

| subsubsection E.5.1 | subsubsection E.5.2 | |
|---|---|---|
| Lines | Lines | |
| 53 ⟶ 57 | 32 ⟶ 48 | 52 ⟶ 66 |
| 61 ⟶ 64 | 71 ⟶ 86 | 91 ⟶ 101 |
| 68 ⟶ 72 | 106 ⟶ 114 | 119 ⟶ 125 |

   5.1 if the maximum number of retries have occurred then begin the download at the next lower baud.

   5.2 Retry by going back to step 2.

6. Send the "Set Baud" (`0x67`, `D_EC_SET_BAUD` subsection 5.52) message to set the units baud to the "current baud" (the highest baud the passed the tests above). Repeat until a valid response is received. Once a valid response is received then go to step 7. (subsubsection E.5.1, lines 73, 74)

7. Send the "Start Download" (`0x69`, `D_EC_START_DOWNLOAD` subsection 5.53) message. Repeat until a valid response is received. Once a valid response is received then continue the download using the algorithm that is used on the RJ-45 port (Appendix F). (subsubsection E.5.1, lines 76, 77, subsubsection E.5.2, lines 130 ⟶ 136. The RJ-45 algorithm's operation is shown in subsubsection E.5.1, lines 79 ⟶ 89.)

## E.5 Typical Download Captures

The following download captures were all done:

1. On a Spectra III SE,
2. Model `DD53CBW`,
3. Running software Version 1.34.000,
4. BIOS Version 1.12.0000,
5. With font version 3.03 loaded.
6. Initial Spectra communications parameters were to operate at 9600 baud.
7. The head end, a PC, was communicating at 9600 baud.
8. The capture software was running at 2400 baud.
9. Both downloads would have been successful, but were intentionally stopped as the only reason for doing them was to show how the start of a download sequence operates.
10. Many items of importance do not appear in the captures. This is because the actual events occur at data rates other than 2400 baud and only unintellageble junk was captured.
11. Other parts of the communication were intentionally deleted as having no importance to gaining an understanding of the basics of the download sequence.

A typical pair of lines from the capture file are (these came from subsubsection E.5.1, extra blanks have been removed):

```
A   B    C    D    E      F        G    H                    I
65 145, 202: DCE  365 4.398797 0.512087 ff 01 00 67 00 01 69 D_EC_SET_BAUD 4800
66 145, 164:  DTE 372 4.437848 0.004167 ff 01 00 69
```

**A** This is the sequential line number of the listing.

**B** This is the message number. This number is incremented every other time the "direction" of data changes.

**C** Byte number within "this" direction of data.

**D** Direction of data flow.

- `DCE`: this is for all data coming from the PC.
- `DTE`: this is for all data coming from the TXB, or Spectra.

**E** Overall total byte number from both directions.

**F** Time, in seconds, from when the first byte was detected by the capture software.

**G** Time, in seconds, from the preceding byte.

**H** The actual message. Usually these are arranged in full messages.

Many times, the data is being collected at the "wrong" baud rate, it is not always practable to try and group the data properly.

On long messages, only the start and end have been displayed.

**I** When it seems appropriate, a comment of what is happening on this line.

### E.5.1 Download with the baud rate limited to 2400 baud

```
 1  $ Id: makesecs.l,v 1.8 2006-05-01 08:31:47-07 Hamilton Exp Hamilton $
 2  $Header: DnldX01.out: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:05 AM$
 3  FTS capture buffer (11/2/2006 9:27:49 AM)
 4  Event 1 (11/2/2006 9:18:58.625999 AM) through
 5  Event 2,150 (11/2/2006 9:19:11.900819 AM)
 6
 7  Capture between the Headend and the TXB
 8  Baud rate = 2400
 9  Using Perrin's computer
10  DCE = PC
11  DTE = TXB
12  Download would have been a success if I hadn't stopped it.
13
14       1,       1: DCE       1     0.000000   0.000000 ff
15       1,       2: DCE       2     0.405675   0.405675 ff 01 00 57 00 00 58 EC_DOWNLOAD
16       1,       9: DCE       9     0.751281   0.320688 0f
17       1,      10: DCE      10     0.755436   0.004155 20
18       1,      11: DCE      11     0.759574   0.004138
19       1,      12: DCE      12     0.759585   0.000011 00
20       1,      13: DCE      13     0.763561   0.003976 f2
21       1,      14: DCE      14     1.178100   0.414539 38
22       1,      15: DCE      15     1.182207   0.004107 f6
23
24       1,       1:  DTE     16     1.196256   0.004107 e8
25       1,       2:  DTE     17     1.217455   0.021199 ff 01 00 58
26
27       2,      16: DCE      21     1.419371   0.189416 ff 01 00 67 00 00 68 EC_SET_BAUD 2400
28       2,      23: DCE      28     1.966221   0.521932 ff 01 00 67 00 00 68 EC_SET_BAUD 2400
```

```
29       2,       6:  DTE     35     2.005346   0.004140 ff 01 00 68
30
31       3,      30: DCE      39     2.082392   0.064417 ff 01 00 65 00 00 66 EC_ECHO_MODE
32       3,      10:  DTE     46     2.121102   0.004141 ff 01 00 66
33
34       4,      37: DCE      50     2.154551   0.020845 53
35       4,      38: DCE      51     2.158692   0.004141 33
36       4,      39: DCE      52     2.162913   0.004221 34
37       4,      40: DCE      53     2.166999   0.004086 35
38       4,      14:  DTE     54     2.168523   0.004086 53
39       5,      41: DCE      55     2.171165   0.004086 30
40       5,      15:  DTE     56     2.172742   0.001577 33
41  <Echo Data>
42     141,     177: DCE     327     2.736086   0.000766 33
43     141,     151:  DTE    328     2.739513   0.003427 36
44     142,     178: DCE     329     2.740252   0.000739 32
45     142,     152:  DTE    330     2.743679   0.000739 30
46     142,     153:  DTE    331     2.747845   0.004166 37
47     142,     154:  DTE    332     2.752013   0.004168 33
48     142,     155:  DTE    333     2.756178   0.004165 32
49
50     143,     179: DCE     334     3.013095   0.256917 ff 01 00 67 00 01 69 EC_SET_BAUD 4800
51     143,     156:  DTE    341     3.051882   0.004166 ff 01 00 69
52
53     144,     186: DCE     345     3.129271   0.064783 0f
54     144,     187: DCE     346     3.133430   0.004159 80
55     144,     188: DCE     347     3.137571   0.004141
56     144,     189: DCE     348     3.137582   0.000011 00
57     144,     190: DCE     349     3.141554   0.003972 f5
58     144,     191: DCE     350     3.758199   0.616645 ff 01 00 67 00 01 69 EC_SET_BAUD 4800
59     144,     160:  DTE    357     3.797116   0.004168 ff 01 00 69
60
61     145,     198: DCE     361     3.874267   0.064653 1f
62     145,     199: DCE     362     3.878412   0.004145 80
63     145,     200: DCE     363     3.882569   0.004157 08
64     145,     201: DCE     364     3.886710   0.004141 f5
65     145,     202: DCE     365     4.398797   0.512087 ff 01 00 67 00 01 69 EC_SET_BAUD 4800
66     145,     164:  DTE    372     4.437848   0.004167 ff 01 00 69
67
68     146,     209: DCE     376     4.514868   0.064524 0f
69     146,     210: DCE     377     4.519030   0.004162 80
70     146,     211: DCE     378     4.523167   0.004137
71     146,     212: DCE     379     4.523177   0.000010 00
72     146,     213: DCE     380     4.527153   0.003976 f5
73     146,     214: DCE     381     5.278667   0.751514 ff 01 00 67 00 00 68 EC_SET_BAUD 2400
74     146,     168:  DTE    388     5.317610   0.004164 ff 01 00 68
75
76     147,     221: DCE     392     5.394915   0.064728 ff 01 00 69 00 00 6a EC_START_DOWNLOAD
77     147,     172:  DTE    399     5.433703   0.004167 ff 01 00 6a
78
79     148,     228: DCE     403     5.466989   0.020732 03
80     148,     176:  DTE    404     5.480918   0.020732 01
81
82     149,     229: DCE     405     5.501813   0.020732 53 30 30 . . . 31 42 0d
83     149,     177:  DTE    422     5.766859   0.198586 01
84
85     150,     246: DCE     423     5.787506   0.020647 53 33 34 . . . 35 34 0d
86     150,     178:  DTE    566     6.393469   0.004141 01
87
88  <Much more boring data>
89
90  There were a total of     2149 bytes transferred
91
92  There were a total of     1961 DCE bytes transferred
93  The first DCE byte came in at 0.000000 seconds from the start of data collection
94  The last DCE byte was at 13.265195 seconds from the start of data collection
95
96  There were a total of      188 DTE bytes transferred
97  The first DTE byte came in at 1.196256 seconds from the start of data collection
98  The last DTE byte was at 12.654480 seconds from the start of data collection
```

### E.5.2 Download with the baud rate not limited

```
$ Id: makesecs.l,v 1.8 2006-05-01 08:31:47-07 Hamilton Exp Hamilton $
$Header: DnldX02.out: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:05 AM$
FTS capture buffer (11/2/2006 1:37:52 PM)
Event 1 (11/2/2006 1:35:38.932693 PM) through
Event 2,084 (11/2/2006 1:36:33.261134 PM)

Capture between the headend and the Spectra (No TXB Installed)
Baud RaTE = 2400
uSING pERRIN'S COMPUTER
DCE = PC
DTE = Spectra
Download would have been a success if I hadn't stopped it.

         1     0.000000   0.000000
         2     1.320357   0.000000
      1,       1: DCE       3    43.262720   0.000000 ff
      1,       2: DCE       4    43.567539   0.304819 ff 01 00 57 00 00 58 EC_DOWNLOAD
      1,       9: DCE      11    43.812538   0.220249 0f
      1,      10: DCE      12    43.816683   0.004145 20
      1,      11: DCE      13    43.820842   0.004159 00
      1,      12: DCE      14    43.824981   0.004139 f2
      1,      13: DCE      15    44.038343   0.213362
      1,      14: DCE      16    44.038355   0.000012 00
      1,      15: DCE      17    44.043236   0.004881 ff

      1,       1:  DTE     18    44.046312   0.004881 e8

      2,      16: DCE      19    44.259022   0.004881 ff 01 00 67 00 00 68 EC_SET_BAUD 2,400
      2,      23: DCE      26    44.696419   0.004881 ff 01 00 67 00 00 68 EC_SET_BAUD 2,400
      2,       2:  DTE     33    44.726129   0.004794 ff 01 00 68

      3,      30: DCE      37    44.802797   0.064169 ff 01 00 65 00 00 66 EC_ECHO_MODE
      3,       6:  DTE     44    44.832328   0.004308 ff 01 00 66

      4,      37: DCE      48    44.865685   0.020859 53
      4,      38: DCE      49    44.869825   0.004140 33
      4,      10:  DTE     50    44.870243   0.004140 53
      5,      39: DCE      51    44.873992   0.004140 34
      5,      11:  DTE     52    44.874409   0.000417 33
<Echo Data>
    142,     176: DCE     325    45.443080   0.002238 37
    142,     148:  DTE    326    45.445008   0.002238 30
    143,     177: DCE     327    45.447220   0.002238 33
    143,     149:  DTE    328    45.449174   0.001954 37
    144,     178: DCE     329    45.451387   0.002213 32
    144,     150:  DTE    330    45.453342   0.002213 33
    144,     151:  DTE    331    45.457509   0.004167 32

    145,     179: DCE     332    45.712016   0.254507 ff 01 00 67 00 01 69 EC_SET_BAUD 4,800
    145,     152:  DTE    339    45.741545   0.004141 ff 01 00 69

    146,     186: DCE     343    45.818314   0.064269 1f
    146,     187: DCE     344    45.822463   0.004149 80
    146,     188: DCE     345    45.826622   0.004159 08
    146,     189: DCE     346    45.830761   0.004139 f5

    146,     156:  DTE    347    45.833114   0.004139 0f
    146,     157:  DTE    348    45.837274   0.004160 a8

    147,     190: DCE     349    45.849799   0.012525 21
    147,     158:  DTE    350    45.852040   0.012525 41
<Echo Data at 4800 baud>
    216,     259: DCE     487    46.136253   0.001510 a4
    216,     227:  DTE    488    46.139078   0.002825 a4
    217,     260: DCE     489    46.140580   0.001502 45
    217,     228:  DTE    490    46.143243   0.001502 a5

    218,     261: DCE     491    46.378995   0.001502 ff 01 00 67 00 02 6a EC_SET_BAUD 9,600
```

```
 69     218,     229:  DTE    498    46.408550   0.004624 ff 01 00 6a
 70
 71     219,     268: DCE     502    46.486509   0.065461 38
 72     219,     269: DCE     503    46.490474   0.003965 fe
 73
 74     219,     233:  DTE    504    46.493965   0.003965 f8
 75
 76     220,     270: DCE     505    46.501421   0.003965 86
 77     220,     234:  DTE    506    46.502619   0.001198 86
 78  <Echo Data at 9600 Baud>
 79     252,     302: DCE     569    46.634772   0.002784 39
 80     252,     266:  DTE    570    46.636003   0.001231 39
 81     253,     303: DCE     571    46.638913   0.002910 62
 82     253,     267:  DTE    572    46.640294   0.002910 62
 83     254,     304: DCE     573    46.643080   0.002910 b9
 84     254,     268:  DTE    574    46.644468   0.001388 63
 85     255,     305: DCE     575    46.647220   0.002752 ff
 86     255,     269:  DTE    576    46.648628   0.002752 ff
 87
 88     256,     306: DCE     577    46.868240   0.002752 ff 01 00 67 00 03 6b EC_SET_BAUD 19,200
 89     256,     270:  DTE    584    46.897795   0.004637 ff 01 00 6b
 90
 91     257,     313: DCE     588    46.975009   0.064713 c6
 92     257,     274:  DTE    589    46.978704   0.064713 fe
 93     258,     314: DCE     590    46.982403   0.064713 4b
 94     258,     275:  DTE    591    46.983003   0.000600 4b
 95     259,     315: DCE     592    46.986403   0.003400 42
 96     259,     276:  DTE    593    46.986857   0.003400 42
 97  <Echo Data at 19,200 Baud>
 98     274,     330: DCE     622    47.048190   0.003671 42
 99     274,     291:  DTE    623    47.048896   0.000706 42
100     275,     331: DCE     624    47.052349   0.003453 b6
101     275,     292:  DTE    625    47.053054   0.003453 42
102
103     276,     332: DCE     626    47.269593   0.003453 ff 01 00 67 00 04 6c EC_SET_BAUD 38,400
104     276,     293:  DTE    633    47.299071   0.004550 ff 01 00 6c
105
106     277,     339: DCE     637    47.376078   0.064507 fb
107     277,     297:  DTE    638    47.377958   0.064507 6f
108     278,     340: DCE     639    47.380266   0.064507 42
109     278,     298:  DTE    640    47.382123   0.001857 42
110  <Echo Data at 38,400 Baud>
111     285,     347: DCE     653    47.408503   0.002395 18
112     285,     305:  DTE    654    47.410066   0.002395 8c
113     286,     348: DCE     655    47.412720   0.002395 c6
114     286,     306:  DTE    656    47.414018   0.001298 dc
115
116     287,     349: DCE     657    47.626025   0.212007 ff 01 00 67 00 05 6d EC_SET_BAUD 115,200
117     287,     307:  DTE    664    47.655450   0.004141 ff 01 00 6d 95
118
119     288,     356: DCE     669    47.734128   0.001205 2a
120     288,     312:  DTE    670    47.736889   0.001205 b4
121  <Echo data at 115,200 Baud>
122     289,     357: DCE     671    47.738085   0.001205 3a
123     289,     358: DCE     672    47.741861   0.001205 e1
124     289,     313:  DTE    673    47.741870   0.000009 07
125     289,     314:  DTE    674    47.745815   0.003945 ff
126
127     290,     359: DCE     675    47.952224   0.206409 ff 01 00 67 00 05 6d EC_SET_BAUD 115,200
128     290,     315:  DTE    682    47.981571   0.004142 ff 01 00 6d
129
130     291,     366: DCE     686    48.058383   0.064313 ff
131     291,     367: DCE     687    48.071317   0.012934 ff
132     291,     368: DCE     688    48.358495   0.287178 95
133     291,     369: DCE     689    48.362479   0.003984 59
134  <Download Data at 115,200 Baud>
135     291,     373: DCE     693    48.379545   0.003959 94
136     291,     374: DCE     694    48.383494   0.003949 53
137
138  There were a total of     2082 bytes transferred
139
140  There were a total of     1764 DCE bytes transferred
```

```
141  The first DCE byte came in at 43.262720 seconds from the start of data collection
142  The last DCE byte was at 54.328441 seconds from the start of data collection
143
144  There were a total of      318 DTE bytes transferred
145  The first DTE byte came in at 44.046312 seconds from the start of data collection
146  The last DTE byte was at 47.994070 seconds from the start of data collection
147
```

### E.5.3 Details of the data start of a download

This is the start of the data transfer of download data. Note that the data is in "Motorola S Record" format. Each record starts with a "S" and ends with a `cr` (`0x0D`) byte.

This is a type "S0" record with 06 pairs of data bytes following it.

```
  1     151,     172:  DTE    608   100.440350   0.004141 ff 01 00 6a
  2
  3     152,     437: DCE     612   100.473695   0.020844 03
  4
  5     152,     176:  DTE    613   100.487434   0.020844 01
  6
  7                                                 S  0  0  6  0  0  0  0  4  8  4  4  5  2  1  B  cr
  8     153,     438: DCE     614   100.508216   0.020844 53 30 30 36 30 30 30 30 34 38 34 34 35 32 31 42 0d
  9
 10     153,     177:  DTE    631   100.771970   0.197179 01
 11
 12                                                 S  3  4  5  0  0  0  0  0  0  0  0  0  0  0  0  0  6  4  3  0  0  0
0  0  7
 13     154,     455: DCE     632   100.792786   0.020816 53 33 34 35 30 30 30 30 30 30 30 30 30 30 30 30 36 46 43 30 30
30 30 30 37
 14
```

### E.5.4 Fatal timeout errors in the middle of a download

On this fatal timeout, caused by unplugging the download cable, note that a total of 9 retrys, lines 4 $\longrightarrow$ 12, were performed before quitting.

```
  1     178,    3980: DCE    4181   116.208871   0.004165 41
  2     178,    3981: DCE    4182   116.213013   0.004142 b0
  3
  4     178,     202:  DTE   4183   116.733497   0.004142 02
  5     178,     203:  DTE   4184   117.243159   0.509662 02
  6     178,     204:  DTE   4185   117.753472   0.510313 02
  7     178,     205:  DTE   4186   118.263612   0.510140 02
  8     178,     206:  DTE   4187   118.773576   0.509964 02
  9     178,     207:  DTE   4188   119.283576   0.510000 02
 10     178,     208:  DTE   4189   119.793392   0.509816 02
 11     178,     209:  DTE   4190   120.303705   0.510313 02
 12     178,     210:  DTE   4191   120.814018   0.510313 02
```

### E.5.5 Recovery following a timeout in the middle of a download

This shows the recovery to a forced time out in the middle of a download.

On line 2 is the last fully "good" message. It received a normal `LACK`. Then we have the start of the message, line 6, that was truncated in the middle by disconnecting the download cable.

In the timeout area, lines 8 ⟶ 13, there are several `LNACK`s, which are sent about every .5 second. When the cable is reconnected, line 16, the system starts to resend the message that was "broken" (line 6). It will have an `LACK` sent and the system continues on with no further problems.

```
 1                                                    S  3  4  5  0  0  0  0  0  4  0  0  9
 2    348,    6552: DCE    6955   223.858009   0.020653 53 33 34 35 30 30 30 30 30 34 30 30 39
 3    348,     404:  DTE   7098   224.463734   0.004167 01
 4
 5                                                    S  3  4  5  0  0  0  0  0  4  4  0  1
 6    349,    6695: DCE    7099   224.484555   0.004167 53 33 34 35 30 30 30 30 30 34 34 30 31
 7
 8    349,     405:  DTE   7158   225.238842   0.513285 02
 9    349,     406:  DTE   7159   225.748973   0.510131 02
10    349,     407:  DTE   7160   226.259048   0.510075 02
11    349,     408:  DTE   7161   226.769102   0.510054 02
12    349,     409:  DTE   7162   227.279102   0.510000 02
13    349,     410:  DTE   7163   227.789076   0.509974 02
14
15                                                   S  3  4  5  0  0  0  0  0  4  4  0  1
16    350,    6754: DCE    7164   227.809892   0.020816 53 33 34 35 30 30 30 30 30 34 34 30 31
17    350,     411:  DTE   7307   228.415584   0.004139 01
```

### E.5.6 Ending sequence of a download

This is the normal end of a download sequence. Note here the sending of a "`S7`" record, line 7, to indicate the end by the PC. Which gets an immediate `LACK`, line 9, before the full message has been received.

The full last message is finished on line 12. Then the PC sends a `LTERM`, line 14, command to finish up the whole process. Which is given an `LACK` before the Spectra updates its flash and reboots.

A full interpreted dump of a typical, here the last, download data message is in Table 5.

```
 1                                                     S 3  4        0  1  2  cr
 2    2149,  263519: DCE  265728  1352.954333   0.004115 53 33 34 . . . 30 31 32 0d Full msg in FULL_MESSAGE
 3
 4    2149,    2210:  DTE 265871  1353.559452   0.015250 01
 5
 6                                                    S  7  0  5  0
 7    2150,  263662: DCE  265872  1353.580198   0.020746 53 37 30 35 30
 8
 9    2150,    2211:  DTE 265877  1353.598199   0.004138 01
10
11                                                   0  0  1  C  0  0  0  3  9  cr
12    2151,  263667: DCE  265878  1353.601076   0.004138 30 30 31 43 30 30 30 33 39 0d
13
14    2151,  263677: DCE  265888  1353.642613   0.004138 85
15
16    2151,    2212:  DTE 265889  1353.656534   0.013921 01
17
```

| Bytes | **Data** | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $1 \rightarrow 16$ | S | 3 | 4 | 5 | 0 | 0 | 0 | 1 | F | F | 3 | 0 | F | F | 0 | 0 |
| — | 53 | 33 | 34 | 35 | 30 | 30 | 30 | 31 | 46 | 46 | 43 | 30 | 46 | 46 | 30 | 30 |
| $17 \rightarrow 32$ | 0 | 0 | F | F | F | F | 0 | 0 | 0 | 0 | F | F | C | 3 | 3 | C |
| — | 30 | 30 | 46 | 46 | 46 | 46 | 30 | 30 | 30 | 30 | 46 | 46 | 43 | 33 | 33 | 43 |
| $33 \rightarrow 48$ | 3 | C | C | 3 | 3 | 3 | C | C | 3 | 3 | C | C | 3 | 3 | C | C |
| — | 33 | 43 | 43 | 33 | 33 | 33 | 43 | 43 | 33 | 33 | 43 | 43 | 33 | 33 | 43 | 43 |
| $49 \rightarrow 64$ | 3 | 3 | C | C | 0 | F | F | 0 | 0 | F | F | 0 | F | 0 | 0 | F |
| — | 33 | 33 | 43 | 43 | 30 | 46 | 46 | 30 | 30 | 46 | 46 | 30 | 46 | 30 | 30 | 46 |
| $65 \rightarrow 80$ | F | 0 | 0 | F | C | C | 3 | 3 | C | C | 3 | 3 | C | C | 3 | 3 |
| — | 46 | 30 | 30 | 46 | 43 | 43 | 33 | 33 | 43 | 43 | 33 | 33 | 43 | 43 | 33 | 33 |
| $81 \rightarrow 96$ | C | C | 3 | 3 | 3 | C | C | 3 | C | 3 | 3 | C | 0 | 0 | F | F |
| — | 43 | 43 | 33 | 33 | 33 | 43 | 43 | 33 | 43 | 33 | 33 | 43 | 30 | 30 | 46 | 46 |
| $97 \rightarrow 112$ | F | F | 0 | 0 | 0 | 0 | F | F | F | F | 0 | 0 | 0 | 0 | 0 | 0 |
| — | 46 | 46 | 30 | 30 | 30 | 30 | 46 | 46 | 46 | 46 | 30 | 30 | 30 | 30 | 30 | 30 |
| $113 \rightarrow 128$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| — | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 |
| $129 \rightarrow 143$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | cr | |
| — | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 30 | 31 | 32 | 0d | |

Table 5: An interpreted version of the last data record of a download sequence for Spectra III, version 1.34, "`FULL_MESSAGE`"

# F Firmware Downloader Implementation (RJ-45 Download Protocol)

The software discussed in this section[95] is implemented in the Spectra III. As other units are designed and built, and equivalent functionality will be included. Commands are shown in Table 6.

The firmware downloader is designed to receive formatted Motorola S-records via the given STREAM interface and write the data portion of said records to display RAM. The records are received via a proprietary protocol exchange defined as follows:

1. Wait for the `LCCHAN` (clear channel) signal.
2. Allocate a buffer to receive the S-record data. If successful, transmit `LACK`, otherwise transmit `LALLOC` and restart.
3. Receive a record from the STREAM. Retries are allowed by utilizing `LNACK`. If the receive fails transmit an `LCOMM` and restart. If `LTERM` is received, go to step 6.
4. Write the received record to display RAM.
5. Transmit an `LACK` and go to line 3.
6. Load the routine ROMEXEC into RAM and execute it. This routine retrieves the received records from display RAM and writes them to flash.

Note that errors encountered during a firmware update may result in a application reboot with invalid code space. This has been addressed via LOADER_MAIN, which acts as an application backup in the case of code space corruption.

| | | |
|---|---|---|
| `LACK` | `0x01` | Acknowledgement |
| `LNACK` | `0x02` | Negative acknowledgement |
| `LCCHAN` | `0x03` | Clear channel |
| `LERASE` | `0x81` | Erase failure |
| `LWRITE` | `0x82` | Write failure |
| `LALLOC` | `0x83` | Memory allocation failure |
| `LCOMM` | `0x84` | Communications failure |
| `LTERM` | `0x85` | Termination request |
| `LCHIP` | `0x86` | Invalid chip id |

Table 6: RJ-45 Port Download Command Set

[95] This protocol description has been copied from the source code for the Spectra III. A slightly different process is used on the ExSite, or at least there are additional commands in the protocol (Table 7).

## F.1 ExSite Changes

For the ExSite the responses have been expanded:

| | | |
|---|---|---|
| LACK | 0x01 | Acknowledgement |
| LNACK | 0x02 | Negative acknowledgement |
| LCCHAN | 0x03 | Clear channel |
| LCONT | 0x04 | Continue Download |
| LDONE | 0x05 | All Done |
| LERASE | 0x81 | Erase failure |
| LWRITE | 0x82 | Write failure |
| LALLOC | 0x83 | Memory allocation failure |
| LCOMM | 0x84 | Communications failure |
| LTERM | 0x85 | Termination request |
| LCHIP | 0x86 | Invalid chip ID |

Table 7: RJ-45 Port Download Command Set, ExSite version

# G Typical Predefined Presets

In the following table, preset numbers in parentheses are the numbers when operating in "32 preset mode". Presets without parentheses are not available in 32 preset mode. When in 32 preset mode, if limit stops are turned off then presets 23 → 26 may be used as normal presets.

| PRESET ID | Name | Set/Call | Use |
|---|---|---|---|
| $0x21/33_{10}$ | PRESET_FLIP | Call | Standard in all units, causes a 180° turn. |
| $0x22/34_{10}$ | PRESET_ZERO | Call | Standard in all units, causes pan to move to calibrate 0 (*cal_0*). |
| $0x54/84_{10}$ | PRESET_AUX1 | Set<br>Call | Turns on AUX 1 with the ExSite only.<br>Turns off AUX 1 with the ExSite. |
| $0x55/85_{10}$ | PRESET_AUX2 | Set<br>Call | Turns on AUX 2 with the ExSite only.<br>Turns off AUX 2 with the ExSite. |
| $0x55/85_{10}$ | — | Call<br>Set | "Black Hot" choice with the Esprit TI<br>No effect with the Esprit TI |
| $0x56/86_{10}$ | PRESET_WIPER | Call | Turns on/off the wiper with ExSite only. |
| | — | Call | "White Hot" choice with the Esprit TI |
| | — | Call | Toggles inversion of video **only** on PTZ units with Spectra IV units with 23X and 35X cameras installed. Does not invert controls. I.e. left/right pan do the "backwards thing" when the video is inverted. Originally an SMR (SMR 1-1B9GVC) planned for Spectra IV rev 2.01 and following. |
| | — | Set | No effect with the Esprit TI or Spectra IV. |
| $0x57/87_{10}$ | PRESET_WASHER | Call | Turns on the Wash cycle on the ExSite only. |
| $0x57/87_{10}$ | — | Call<br>Set | "Color 1" choice with the Esprit TI<br>No effect with the Esprit TI |
| $0x58/88_{10}$ ($0x15/21_{10}$) | PRESET_IR_FILTER_IN | Call | Used with CBW type cameras. (subsection G.1) |
| $0x58/88_{10}$ ($0x15/21_{10}$) | — | Call<br>Set | "Rain 1" choice with the Esprit TI<br>No effect with the Esprit TI |
| $0x59/89_{10}$ ($0x16/22_{10}$) | PRESET_IR_FILTER_OUT | Call | Used with CBW type cameras. |
| $0x59/89_{10}$ ($0x16/22_{10}$) | — | Call<br>Set | Force an FFC Calibration cycle in the Esprit TI.<br>No effect with the Esprit TI |

*Continued on the next page.*

[96] $Header: Presets.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:13 PM$

*Continued from the previous page.*

| PRESET ID | Name | Set/Call | Use |
|---|---|---|---|
| 0x5A/$90_{10}$ (0x17/$23_{10}$) | PRESET_MANUAL_LEFT_LIMIT | Set | Standard in all units, sets a left hand motion limit. |
| | | Call | No effect |
| | | Clear | Clears this limit |
| 0x5B/$91_{10}$ (0x18/$24_{10}$) | PRESET_MANUAL_RIGHT_LIMIT | Set | Standard in all units, sets a right hand motion limit. |
| | | Call | No effect |
| | | Clear | Clears this limit |
| 0x5C/$92_{10}$ (0x19/$25_{10}$) | PRESET_SCAN_LEFT_LIMIT | Set | Standard in all units, sets a left hand scanning limit. |
| | | Call | No effect |
| | | Clear | Clears this limit |
| 0x5D/$93_{10}$ (0x1A/$26_{10}$) | PRESET_SCAN_RIGHT_LIMIT | Set | Standard in all units, sets a right hand scanning limit. |
| | | Call | No effect |
| | | Clear | Clears this limit |
| 0x5E/$94_{10}$ (0x1B/$27_{10}$) | PRESET_RESET | Set | Reserved for future use. |
| | | Call | On Esprit units only, starting with rev 3.70, will cause the WatchDog timer to time out and reset the unit. |
| 0x5F/$95_{10}$ (0x1C/$28_{10}$) | PRESET_MENU_MODE | Set | Standard in all units, accesses the menu system. |
| | | Call | Ignored, to get into menu mode a set **must** be used. |
| 0x60/$96_{10}$ (0x1D/$29_{10}$) | PRESET_STOP_SCAN | Set | Standard in all units, stops automatic scans. |
| | | Call | Same as a Set command |
| 0x61/$97_{10}$ (0x1E/$30_{10}$) | PRESET_RANDOM_SCAN | Set | Standard in all units, starts random scanning. |
| | | Call | Same as a Set command |
| 0x62/$98_{10}$ (0x1F/$31_{10}$) | PRESET_FRAME_SCAN | Set | Standard in all units, starts frame scanning. |
| | | Call | Same as a Set command |
| 0x63/$99_{10}$ (0x20/$32_{10}$) | PRESET_AUTO_SCAN | Set | Standard in all units, starts automatic scanning. |
| | | Call | Same as a Set command |

## G.1 IR Cut Filter

How I can turn on/off the IR-filter[97]?

With cameras that have an IR cut-filter and we can capture the same scene in Black and White with no "noise" artifacts we can limit the source to the color burst and its phase/saturation/etc properties.

[97] From an e-mail by Derek Springer, 24APR06.

With a camera capable of C/BW such as, the 23x or 18x, Spectra III, the IR cut filter can be put into manual mode Via the menus...

```
<MAIN>
  |
  |--<DOME SETTINGS>
       |
       |--<CAMERA>
            |
            |--IR CUT FILTER --- Set to "OFF" for manual control
```

Once the camera is set to “OFF” for IR cut filter control, Preset 89 can be used to switch the cut filter out and put the camera into black and white mode. Preset 88 puts the IR cut filter back in for normal color operation. If the “noise” artifacts are no longer present in the same scene [i.e. we don’t now have black and white image artifacts] we would then be able to detect that the artifacts are coming from the color properties....

# H IRD/ERD information

Note from Craig Hannen about the ways that the IRD2024/ERD2200 processes commands and presets. The name **IRD** stands for **I**nternal **R**eciever **D**river and the name **ERD** stands for **E**nvironmental **R**eciever **D**river, i.e. usable outside.

```
From:       Hannen, Craig
Sent:       Friday, August 26, 2005 5:05 PM
To:         Hamilton, Eric
Subject:    IRD2024/ERD2200 Preset and Command Information
```

..., there are a small set of commands that are supported in the IRD/ERD. All commands are received via Coaxitron® (Standard or Extended). Pan/tilt/lens control is done at the bit level. The Coaxitron® Bit for each pan/tilt/lens control is transferred directly to the control line that corresponds to the bit. Since the IRD/ERD controls fixed speed pan and tilts, all speed information is ignored.

The IRD/ERD has no support for preset positioning, but there is support for the preset commands that invoke scanning. The preset commands that are supported in Extended Coaxitron® are:

1. **Preset 96** — Manual Scan (turns off the next three scan modes)
2. **Preset 97** — Random Scan (scans for 0-60 seconds; pauses for 0-60 seconds; restarts in a Random direction; repeat)
3. **Preset 98** — Continuous Scan (in other words; scan forever)
4. **Preset 99** — 30 minutes of Continuous Scanning followed by Random Scanning

Unlike many of our receivers, PTZ control will not take the The IRD/ERD out of scan mode. Only Preset 96 (or the Standard Coaxitron® manual scan command) will do this.

To support presets 97 and 99 in Standard Coaxitron®, the auto scan command is used. When the The IRD/ERD receives the first Standard Coaxitron® auto scan command, it goes into Random Scan mode (Preset 97 above) . It's important that any auto scan command is followed by an all stop command (I have found that not all Standard Coaxitron® transmitters work this way, even though they should). The all stop command is used as a command separator. After the all stop command, the next auto scan command enables the Preset 99 scan described above. A third auto scan command puts the The IRD/ERD back in Random Scan mode. The Standard Coaxitron® manual scan command will turn off either scan mode.

The The IRD/ERD supports 2 auxes (aux 1 is a 5V, 20mA open collector output and aux 2 is a relay). In Standard Coaxitron® the auxes are momentary (auxes are switched on with the corresponding aux command and turned off via an all stop command). In Extended Coaxitron® auxes can be momentary or latching. The Extended aux 1 and aux 2 on commands will momentarily switch auxes on for 100 milliseconds. The Extended aux 3 and aux 4 on/off commands will latch auxes 1 and 2 respectively.

The list of predefined constants in the source code is small. Here's the list:

```
PresetGo    equ 07h   ;extended preset go command.
SetAux      equ 09h   ;extended set auxiliary command.
ClrAux      equ 0Bh   ;extended clear auxiliary command.
SetManual   equ 96    ;extended command - manual mode.
SetRandom   equ 97    ;extended command - random mode.
SetAllAuto  equ 98    ;extended command - infinite auto.
SetAuto     equ 99    ;extended command - 30 minute auto then to random.
```

[98] `$Header: IrdErd.inc: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:12 AM$`

The The IRD/ERD checks the validity of a command by counting the number of bits in the command. If the number of bits is not 15, 16 or 32, the command is discarded.

I think this sums up just about everything anyone would ever want to know about the IRD/ERD. If I left out anything important, let me know.

Craig Hannen
channen@pelco.com
1-800-289-9100 (2395)

## H.1 ERD97P21-U

The ERD97P21-U uses P Protocol and a reduced set of commands. It does support the normal P Protocol set of motion commands of pan, tilt, iris, telephoto and focus. In addition it supports only the following commands:

| Name | Value |
|---|---|
| Commands | |
| EC_SET_PRESET | 3 |
| EC_CLEAR_PRESET | 5 |
| EC_MOVE_PRESET | 7 |
| EC_SET_AUX | 9 |
| EC_CLEAR_AUX | 11 |
| EC_RESET | 15 |
| EC_ACK_ALARM | 25 |
| EC_ZOOM_SPEED | 37 |
| EC_FOCUS_SPEED | 39 |
| Special Presets | |
| PRESET_STOP_SCAN | 96 |
| PRESET_RANDOM_SCAN | 97 |
| PRESET_FRAME_SCAN | 98 |
| PRESET_AUTO_SCAN | 99 |
| If enabled only | |
| EC_STATUS | 0xFD |

Table 8: ERD97P21-U commands and presets

# I LRD Information

The **LRD** is a receiver driver designed for use with legacy pan/tilt units. There are three different basic models. These are:

1. **LRDA41**, Coaxitron® only control.
2. **LRD41C11**, Coaxitron® with P and D Protocol control for fixed speed Pan/Tilt units without presets.
3. **LRD41C12**, Coaxitron® with P and D Protocol control for variable speed Pan/Tilt units without presets.
4. **LRD41C21**, Coaxitron® with P and D Protocol control for fixed speed Pan/Tilt units with presets.
5. **LRD41C22**, Coaxitron® with P and D Protocol control for variable speed Pan/Tilt units with presets.

*This section is directly from the manual for the LRD41C21/22 C557M (10/99) page 20.*

Auto and random scan can be operated in either of two ways, depending on the type of control unit you have. One method involves using the AUTO/MAN switch (or keys) if your control unit has these functions. The other method involves setting presets.

- AUTO/MAN Switch: The Random Scan and Auto Scan functions are controlled by the same momentary switch on the control panel labeled AUTO and MAN. The first activation of the switch to the AUTO position will put the pan/tilt into Random Scan. In Random Scan operation, the pan/tilt will travel between the preset limits with a random scan period of about 2 to 30 seconds, and a random dwell period of between 2 to 30 seconds.

  At the completion of a dwell period, another random scan period is started. The direction of this scan period is also randomly determined. When a pan limit is reached, scan direction is reversed automatically.

  A second activation of the AUTO switch will put the pan/tilt into continuous duty Auto Scan (limit switch to limit switch). After approximately $\frac{1}{2}$ hour of auto scan, the circuit will reset to random scan. Commanding AUTO while in RANDOM mode causes a shift to AUTO mode and starts the $\frac{1}{2}$-hour timer. Similarly, commanding AUTO while in AUTO mode causes a shift to the RANDOM mode and zeros the $\frac{1}{2}$-hour timer.

- Presets: Auto and random scan also can be started by programming presets. The presets will work only when your system is configured for Extended (32-bit) Coaxitron® mode.

## I.1 Advantages of Random Scan

Because scan direction, scan period and dwell period are unpredictable, unauthorized activities or intrusions are discouraged.

Because of the reduced duty cycle, gear train wear, cable fatigue, drive motor wear and temperature rise are reduced. These factors all contribute to higher system reliability and increased equipment life.

[99] `$Header: LRD.inc: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:13 AM$`

# J Strange Usages of Pelco Protocols by Competitors

Many of Pelco's competitors have unexpected concepts of how to implement commands in our protocols. These are representative discoveries from various data captures.

## J.1 Dedicated Micros Sprite 2, P Protocol

This data capture was made on 29MAY07 at Pelco using a Dedicated Micros Sprite 2 Digital Video Recorder. In this sample data capture a baud rate of 4800 was used as was no parity.

It should be noted that when a command is initiated that:

1. A "Set Focus Speed" command is sent.
2. A "Set Zoom Speed" command is sent.
3. Then the correct command is sent. Here it is a "Focus Near" command.
4. Following the original command, the original command is resent about every 800 ms, until the function is deselected.
5. A "Set Focus Speed" command is sent.
6. A "Set Zoom Speed" command is sent.
7. And last a "Stop" is sent.

**Note**

In the above sequence of commands:

1. The camera address is set to zero. This is probably because I could not figure out how to set the address in the DMS software.
2. That there are no replies from the camera. This is because a camera was not connected to the DMS unit. However this probably did not affect the order or quantity of the commands, because in no case were there any retransmissions of command. Except for resending the "action command", which is happening at a very consistent rate. There are a few special commands where the action command is not resent even though there is no reply from the camera.
3. This pattern of command sending is used for most commands that are sent.

```
1169:  DTE   1173   373.789791   4.166629 a0 00 00 25 00 01 af 2b   Set Zoom Speed step 1
1177:  DTE   1181   373.809788   0.005209 a0 00 00 27 00 01 af 29   Set Focus Speed step 1
1185:  DTE   1189   374.041003   0.215962 a0 00 02 00 31 31 af 0d   Focus Near
1193:  DTE   1197   374.857747   0.801957 a0 00 02 00 31 31 af 0d   Focus Near, 800 ms later
1201:  DTE   1205   375.674076   0.801750 a0 00 02 00 31 31 af 0d   Focus Near, 800 ms later
1209:  DTE   1213   376.490403   0.801748 a0 00 02 00 31 31 af 0d   Focus Near, 800 ms later
1217:  DTE   1221   377.306940   0.801958 a0 00 02 00 31 31 af 0d   Focus Near, 800 ms later
1225:  DTE   1229   378.123266   0.801748 a0 00 02 00 31 31 af 0d   Focus Near, 800 ms later
1233:  DTE   1237   378.342809   0.204964 a0 00 00 25 00 01 af 2b   Set Zoom Speed step 1
1241:  DTE   1245   378.374890   0.017291 a0 00 00 27 00 01 af 29   Set Focus Speed step 1
1249:  DTE   1253   378.592981   0.203300 a0 00 00 00 31 31 af 0f   Stop
```

[100] $Header: ProtoUse.inc: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:17 AM$

# K About the Joystick Report

The original source for the Joystick Report has been misplaced[102]. There are several "Xerox" copies of one of the originals. The overall findings of the Joystick Report are that a non-linear speed table resident on the dome, pan/tilt, should be used for the best control of a dome or pan/tilt device. For this to work correctly the commands received by the dome, **must** be generated in a linear manner. I.e. as the joy stick is moved further and further from the center the values generated must monotonically increase in a linear manner. Since humans perceive most things in a non-linear manner, the non-linear portion of the control loop must be inside the dome and must be matched to the dome's internal/physical characteristics. A suggested table of non-linear dome speeds is shown in Table 9 of the Joystick Report, it is plotted in Figure 8. Motion control of the Spectra series of domes improved significantly after the recommendations of the Joystick Report were adopted. (The Esprit series of pan/tilt units always have had the improved speed tables suggested by the Joystick Report.)

## K.1 "Ideal" speeds from sheet 6 of the Joystick Report

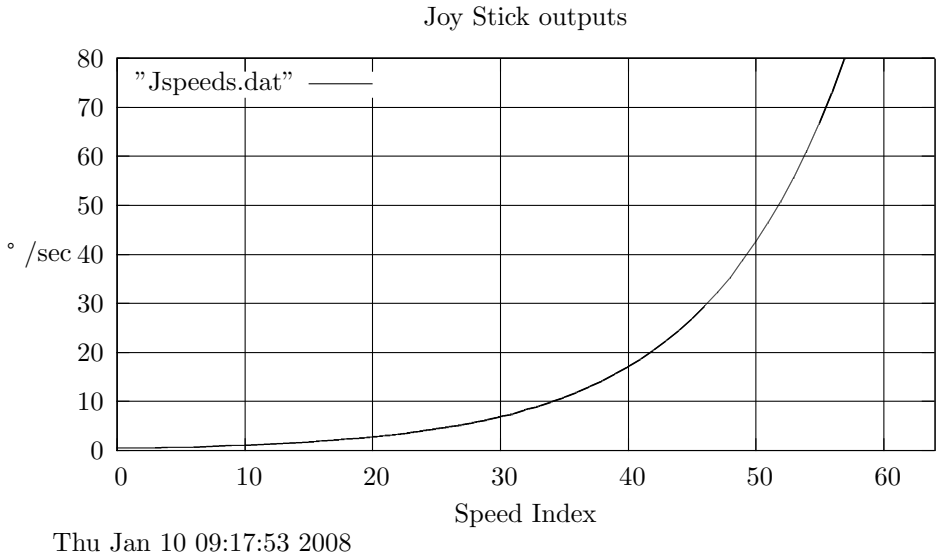



Figure 8: Speeds as plotted from the JoyStick Report

[101] `$Header: JS.inc: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:12 AM$`
[102] The last known original was "misplaced" by Eric Hamilton sometime in this century or the last.
[103] `$Header: JsSpeeds.inc: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:13 AM$`

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 005 | 30 | 016 | 59 | 063 | 88 | 224 |
| 2 | 005 | 31 | 017 | 60 | 063 | 89 | 246 |
| 3 | 005 | 32 | 017 | 61 | 069 | 90 | 246 |
| 4 | 005 | 33 | 019 | 62 | 069 | 91 | 269 |
| 5 | 005 | 34 | 019 | 63 | 075 | 92 | 269 |
| 6 | 005 | 35 | 021 | 64 | 075 | 93 | 295 |
| 7 | 005 | 36 | 021 | 65 | 083 | 94 | 295 |
| 8 | 005 | 37 | 023 | 66 | 083 | 95 | 323 |
| 9 | 006 | 38 | 023 | 67 | 090 | 96 | 323 |
| 10 | 006 | 39 | 025 | 68 | 090 | 97 | 353 |
| 11 | 007 | 40 | 025 | 69 | 099 | 98 | 353 |
| 12 | 007 | 41 | 027 | 70 | 099 | 99 | 387 |
| 13 | 007 | 42 | 027 | 71 | 108 | 100 | 387 |
| 14 | 007 | 43 | 030 | 72 | 108 | 101 | 424 |
| 15 | 008 | 44 | 030 | 73 | 119 | 102 | 424 |
| 16 | 008 | 45 | 033 | 74 | 119 | 103 | 464 |
| 17 | 009 | 46 | 033 | 75 | 130 | 104 | 464 |
| 18 | 009 | 47 | 036 | 76 | 130 | 105 | 508 |
| 19 | 010 | 48 | 036 | 77 | 142 | 106 | 508 |
| 20 | 010 | 49 | 040 | 78 | 142 | 107 | 556 |
| 21 | 011 | 50 | 040 | 79 | 156 | 108 | 556 |
| 22 | 011 | 51 | 044 | 80 | 156 | 109 | 609 |
| 23 | 012 | 52 | 044 | 81 | 171 | 110 | 609 |
| 24 | 012 | 53 | 048 | 82 | 171 | 111 | 667 |
| 25 | 013 | 54 | 048 | 83 | 187 | 112 | 667 |
| 26 | 013 | 55 | 052 | 84 | 187 | 113 | 730 |
| 27 | 014 | 56 | 052 | 85 | 205 | 114 | 730 |
| 28 | 014 | 57 | 057 | 86 | 205 | 115 | 800 |
| 29 | 016 | 58 | 057 | 87 | 224 | 116 | 800 |

Table 9: Speed values from sheet 6 of “The Joystick Report”, September 19, 1997

# L D and P Protocol defines

```
/// $Description: D Protocol command names $
/// $Workfile: DProto.h$
/// Copyright(&copy;) by Pelco, 2007
/// $Header: DProto.h: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:06 AM$
#define doxygen_dproto_h

#ifndef DPROTO_H
#define DPROTO_H

///
/// D Protocol command format, all commands are 7 bytes long as follows:
/// - Sync 0xFF, not in checksum
/// - Address, "0" based, i.e. 0 = address zero, 1 = address 1
/// - CMND1, first byte of a bit encoded command
/// - CMND2, second byte of a bit encoded command or all of a numeric command
/// - DATA1, pan speed index or additional data for a numeric command
/// - DATA2, tilt speed index or additional data for a numeric command
/// - CKSM, least significant byte of the arithmetic sum of
///   bytes 2, 3, 4, 5 and 6
#define doxygen_dprotocol_format

#define D_SYNC  0 ///< Sync 0xFF, not in checksum
#define D_ADDR  1 ///< Address, "0" based, i.e. 0 = address zero, 1 = address 1
#define D_CMND1 2 ///< first byte of a bit encoded command
#define D_CMND2 3 ///< second byte of a bit encoded command or all of a numeric command
#define D_DATA1 4 ///< pan speed index or additional data for a numeric command
#define D_DATA2 5 ///< tilt speed index or additional data for a numeric command
#define D_CKSM  6 ///< least significant byte of the arithmetic sum of bytes 2, 3, 4, 5 and 6.


// D command types
#define D_BIT_ENCODED 0x01 ///< All bit encoded commands have bit 0 = 0

// Pan/Tilt fixed speed steps
#define FIXED_SPEED          50 ///< Gives about 25 degree sec in pan
#define EXSITE_FIXED_SPEED   56 ///< Gives about 25 degree sec in pan

#define D_EC_SYNC           0xFF    // D protocol synch byte


// D command bits for bit encoded commands in Dcmnd1 and Dcmnd2
//
// The next five definations are rarly implemented in moderen units. That
// includes all Spectras, Esprits, ExSites, etc.
//
#define D_SENSE         0x80 ///< Dcmnd1
#define D_RESERVED1     0x40 ///< Dcmnd1
#define D_RESERVED2     0x20 ///< Dcmnd1
#define D_AUTO_MANUAL   0x10 ///< Dcmnd1
#define D_CAMERA_ON_OFF 0x80 ///< Dcmnd1

#define D_CLOSE   0x04    ///< Dcmnd1
#define D_OPEN    0x02    ///< Dcmnd1
#define D_NEAR    0x01    ///< Dcmnd1

#define D_FAR     0x80    ///< Dcmnd2
#define D_WIDE    0x40    ///< Dcmnd2, Identical usage in P Protocol
#define D_TELE    0x20    ///< Dcmnd2, Identical usage in P Protocol
```

```
59
60 #define D_DOWN    0x10    ///< Dcmnd2 plus Ddata2, Identical usage in P Protocol
61 #define D_UP      0x08    ///< Dcmnd2 plus Ddata2, Identical usage in P Protocol
62 #define D_LEFT    0x04    ///< Dcmnd2 plus Ddata1, Identical usage in P Protocol
63 #define D_RIGHT   0x02    ///< Dcmnd2 plus Ddata1, Identical usage in P Protocol
64
65
66 // Standard command length
67 #define D_EC_COMMAND_LENGTH          7 ///< Length of D protocol commands
68
69 // Reply lengths
70 #define D_EC_GENERAL_REPLY_LENGTH    4 ///< Length of most replies
71 #define D_EC_EXTENDED_REPLY_LENGTH   7 ///< Length of all non-EC_QUERY querys
72 #define D_EC_QUERY_REPLY_LENGTH     18 ///< Length of an EC_QUERY reply
73
74
75
76 ///
77 /// Command IDs
78 ///
79 /// Naming conventions:
80 /// - D = D Protocol
81 /// - EC = Extended command opcode in CMND2
82 /// - ECD = Extended command data in DATA2
83 /// - ECS = Extended command sub-opcode in CMND1
84 ///
85 /// Unless otherwise specified all D_EC_.... values are in CMND2
86 ///
87 #define doxygen_dprotocol_naming
88
89 // New with Spectra III/ExSite
90 #define D_EC_STD_EXT_RESP    0x01    ///< Extended reply identifier
91 // EC_STD_EXT_RESP sub opcodes
92 #define D_ECS_STD_EXT_RESP_ACK     0x01 ///< Extended reply ACK
93 #define D_ECS_STD_EXT_RESP_NAK     0x00 ///< Extended reply NAK
94
95
96 // Basic list from Spectra I and Intercept
97 #define D_EC_SET_PRESET      0x03    ///< 03 Set a preset
98 #define D_EC_CLEAR_PRESET    0x05    ///< 05 Clear a preset
99 #define D_EC_MOVE_PRESET     0x07    ///< 07 Move to preset
100
101 #define D_EC_SET_AUX         0x09    ///< 09 Turn on an aux
102 // Sub opcodes for CMND1
103 #define D_ECS_SET_AUX_RELAY   0x00    ///< Turn on an aux
104 #define D_ECS_SET_AUX_LED     0x01    ///< Turn on a LED
105 // Data codes for DATA2
106 #define D_ECD_SET_AUX_LED_GREEN  0xFE ///< Access the greem LED
107 #define D_ECD_SET_AUX_LED_RED    0xFD ///< Access the red LED
108 #define D_ECD_SET_AUX_LED_AMBER  0xFC ///< Access the red/green LED which makes an amber colog
109
110 #define D_EC_CLEAR_AUX       0x0B    ///< 11 Turn off an aux
111 // Sub opcodes for CMND1
112 #define D_ECS_CLEAR_AUX_RELAY 0x00    ///< Turn off an aux
113 #define D_ECS_CLEAR_AUX_LED   0x01    ///< Turn off a LED
114 // Data codes for DATA2
115 #define D_ECD_CLEAR_AUX_LED_GREEN  0xFE ///< Access the greem LED
116 #define D_ECD_CLEAR_AUX_LED_RED    0xFD ///< Access the red LED
117 #define D_ECD_CLEAR_AUX_LED_AMBER  0xFC ///< Access the red/green LED which makes an amber colog
118
```

```
119  #define D_EC_DUMMY_1         0x0D    ///< 13 Dummy command
120  #define D_EC_RESET           0x0F    ///< 15 Reset
121  #define D_EC_ZONE_START      0x11    ///< 17 Set zone start
122  #define D_EC_ZONE_END        0x13    ///< 19 Set zone end
123
124  #define D_EC_WRITE_CHAR      0x15    ///< 21 Write character on screen
125  #define D_ECS_WRITE_CHAR_LABEL            0 ///< Write into the label buffer
126  #define D_ECS_WRITE_CHAR_TITLE_LABEL      1 ///< (Future Use) Write into the title buffer
127  #define D_ECS_WRITE_CHAR_MESSAGE_LABEL    2 ///< (Future Use) Write into the message field, not saved in
EEPROM/Flash
128  #define D_ECS_WRITE_CHAR_CONTROLLER_IP    3 ///< Write into the controller IP buffer
129  #define D_ECS_WRITE_CHAR_CONTROLLER_MODEL 4 ///< Write into the controller model name buffer
130
131  #define D_EC_CLEAR_SCREEN    0x17    ///< 23 Clear the screen
132  #define D_EC_ALARM_ACK       0x19    ///< 25 Acknowledge alarms
133  #define D_EC_ZONE_ON         0x1B    ///< 27 Set zone scan on
134  #define D_EC_ZONE_OFF        0x1D    ///< 29 Set zone scan off
135  #define D_EC_START_RECORD    0x1F    ///< 31 Start pattern record
136  #define D_EC_END_RECORD      0x21    ///< 33 End pattern record
137  #define D_EC_START_PLAY      0x23    ///< 35 Start pattern play
138
139  #define D_EC_ZOOM_SPEED      0x25    ///< 37 Set zoom speed
140  // Data codes for DATA2
141  #define D_ECD_ZOOM_SPEED_SLOW    0x00 ///< The lowest zoom speed
142  #define D_ECD_ZOOM_SPEED_MEDIUM  0x01 ///< A medium zoom speed
143  #define D_ECD_ZOOM_SPEED_FAST    0x02 ///< A fast zoom speed
144  #define D_ECD_ZOOM_SPEED_FASTEST 0x03 ///< The fastest zoom speed
145
146  #define D_EC_FOCUS_SPEED     0x27    ///< 39 Set focus speed
147  // Data codes for DATA2
148  #define D_ECD_FOCUS_SPEED_SLOW    0x00 ///< The lowest focus speed
149  #define D_ECD_FOCUS_SPEED_MEDIUM  0x01 ///< A medium focus speed
150  #define D_ECD_FOCUS_SPEED_FAST    0x02 ///< A fast focus speed
151  #define D_ECD_FOCUS_SPEED_FASTEST 0x03 ///< The fastest focus speed
152
153  #define D_EC_CAMERA_RESET    0x29    ///< 41 Reset the camera
154
155  #define D_EC_AUTO_FOCUS      0x2B    ///< 43 Set auto focus to device, off, or on
156  // Data codes for DATA2
157  #define D_ECD_AUTO_FOCUS_AUTO 0x00   ///< Set focus mode to automatic
158  #define D_ECD_AUTO_FOCUS_OFF  0x01   ///< Turn off auto focus
159
160  #define D_EC_AUTO_IRIS       0x2D    ///< 45 Set auto iris to device, off, or on
161  // Data codes for DATA2
162  #define D_ECD_AUTO_IRIS_AUTO 0x00    ///< Set iris mode to automatic
163  #define D_ECD_AUTO_IRIS_OFF  0x01    ///< Turn off auto iris
164
165  #define D_EC_AGC             0x2F    ///< 47 Set AGC to device, off, or on
166  // Data codes for DATA2
167  #define D_ECD_AGC_AUTO       0x00    ///< Set AGC to automatic operation
168  #define D_ECD_AGC_OFF        0x01    ///< Set AGC to off
169
170  #define D_EC_BLC             0x31    ///< 49 Set backlight compensation off or on
171  // Data codes for DATA2
172  #define D_ECD_BLC_OFF        0x01    ///< Set BLC to off
173  #define D_ECD_BLC_ON         0x02    ///< Set BLC to on
174
175  #define D_EC_AWB             0x33    ///< 51 Set auto white balance off or on
176  // Data codes for DATA2
177  #define D_ECD_AWB_ON         0x01    ///< Set AWB to on
```

```
178 #define D_ECD_AWB_OFF        0x02    ///< Set AWB to off
179
180 #define D_EC_DEVICE_PHASE    0x35    ///< 53 Set device phase delay
181 #define D_EC_SHUTTER_SPEED   0x37    ///< 55 Set shutter speed
182
183 #define D_EC_ADJUST_PHASE    0x39    ///< 57 Adjust phase delay
184 // Sub opcodes that go into the CMND1 byte
185 #define D_ECS_ADJUST_PHASE_NEW    0x00 ///< New phase delay in DATA1, DATA2
186 #define D_ECS_ADJUST_PHASE_DELTA  0x01 ///< Phase delay adjustment in DATA1, DATA2
187
188 #define D_EC_ADJUST_RB_WB    0x3B    ///< 59 Adjust red-blue white balance
189 // Sub opcodes that go into the CMND1 byte
190 #define D_ECS_ADJUST_RB_WB_NEW   0x00 ///< New red-blue white balance in DATA1, DATA2
191 #define D_ECS_ADJUST_RB_WB_DELTA 0x01 ///< Red-blue white balance adjustment in DATA1, DATA2
192
193 #define D_EC_ADJUST_MG_WB    0x3D    ///< 61 Adjust magenta-green white balance
194 // Sub opcodes that go into the CMND1 byte
195 #define D_ECS_ADJUST_MG_WB_NEW    0x00 ///< New magenta-green white balance in DATA1, DATA2
196 #define D_ECS_ADJUST_MG_WB_DELTA  0x01 ///< Magenta-green white balance adjustment in DATA1, DATA2
197
198 #define D_EC_ADJUST_GAIN     0x3F    ///< 63 Adjust camera gain
199 // Sub opcodes that go into the CMND1 byte
200 #define D_ECS_ADJUST_GAIN_NEW    0x00 ///< New camera gain in DATA1, DATA2
201 #define D_ECS_ADJUST_GAIN_DELTA  0x01 ///< Camera gain adjustment in DATA1, DATA2
202
203 #define D_EC_ADJUST_AI_LEVEL 0x41    ///< 65 Adjust auto-iris level
204 // Sub opcodes that go into the CMND1 byte
205 #define D_ECS_ADJUST_AI_LEVEL_NEW   0x00 ///< New auto-iris level in DATA1, DATA2
206 #define D_ECS_ADJUST_AI_LEVEL_DELTA 0x01 ///< Auto-iris level adjustment in DATA1, DATA2
207
208 #define D_EC_ADJUST_AI_PEAK  0x43    ///< 67 Adjust auto_iris peak value
209 // Sub opcodes that go into the CMND1 byte
210 #define D_ECS_ADJUST_AI_PEAK_NEW   0x00 ///< New auto_iris peak value in DATA1, DATA2
211 #define D_ECS_ADJUST_AI_PEAK_DELTA 0x01 ///< Auto_iris peak value adjustment in DATA1, DATA2
212
213
214
215
216 // New with Spectra II
217 #define D_EC_QUERY           0x45    ///< 69 Get the program number
218 // Sub opcodes for D_EC_QUERY, new with Spectra IV and Esprit TI
219 #define D_ECS_QUERY_PART_NUMBER   0  ///< 18 byte reply with the part number
220 #define D_ECS_QUERY_SERIAL_NUMBER 1  ///< 18 byte reply with the serial number
221
222 #define D_EC_PRESET_SCAN     0x47    ///< 71 Preset scan
223
224
225
226 // New with Spectra III
227 #define D_EC_SET_ZERO        0x49    ///< 73 Set pan home position
228 #define D_EC_SET_PAN         0x4B    ///< 75 Set absolute pan position
229 #define D_EC_SET_TILT        0x4D    ///< 77 Set absolute tilt position
230 #define D_EC_SET_ZOOM        0x4F    ///< 79 Set the zoom position
231 #define D_EC_QUERY_PAN       0x51    ///< 81 Return the absolute pan position
232 #define D_EC_QUERY_TILT      0x53    ///< 83 Return the absolute tilt position
233 #define D_EC_QUERY_ZOOM      0x55    ///< 85 Return the zoom position
234 #define D_EC_DOWNLOAD        0x57    ///< 87 Reboot into download mode
235 #define D_EC_PAN_RESP        0x59    ///< 89 Query pan resp
236 #define D_EC_TILT_RESP       0x5B    ///< 91 Query tilt resp
237 #define D_EC_ZOOM_RESP       0x5D    ///< 93 Query zoom resp
```

```
238 #define D_EC_SET_MAG         0x5F    ///< 95 Set the magnification value
239 #define D_EC_QUERY_MAG       0x61    ///< 97 Return the current magnification
240 #define D_EC_MAG_RESP        0x63    ///< 99 Query magnification response
241
242 #define D_EC_ECHO_MODE           0x65 ///< 101 Used for download only
243
244 #define D_EC_SET_BAUD            0x67 ///< 103 Used for download only
245 // Data opcodes that go into DATA2
246 #define D_ECD_SET_BAUD_2400      0x00 ///< Select 2400 baud for downloading
247 #define D_ECD_SET_BAUD_4800      0x01 ///< Select 4800 baud for downloading
248 #define D_ECD_SET_BAUD_9600      0x02 ///< Select 9600 baud for downloading
249 #define D_ECD_SET_BAUD_19200     0x03 ///< Select 19200 baud for downloading
250 #define D_ECD_SET_BAUD_38400     0x04 ///< Select 38400 baud for downloading
251 #define D_ECD_SET_BAUD_115200    0x05 ///< Select 115200 baud for downloading
252
253 #define D_EC_START_DOWNLOAD      0x69 ///< 105 Used for download only
254 #define D_EC_QUERY_DEV_TYPE      0x6B ///< 107 Returns the CPU chip type
255 #define D_EC_QUERY_DEV_TYPE_RESP 0x6D ///< 109 Response to EC_QUERY_DEV_TYPE
256 #define D_EC_QUERY_DIAG_INFO     0x6F ///< 111 Obtain diag info
257 #define D_EC_QUERY_DIAG_INFO_RESP 0x71 ///< 113 Response to EC_QUERY_DIAG_INFO
258
259
260
261 // New for Everest
262 // All unimplemted Everest op-codes generate either a 4 or 7 byte response.
263 // If the unit generating the response is a newer Esprit, then a 7 byte
264 // Everest NAK response is usually generated. Otherwise a 4 byte general
265 // response is generated.
266 #define D_EC_VERSION_INFO        0x73 ///< 115 Get software revs for Everest
267
268 /* EC_VERSION_INFO sub opcodes, these go into the CMND1 byte */
269 #define D_ECS_VERSION_INFO_MAIN_CPU_VERSION_QRY   0x00
270 #define D_ECS_VERSION_INFO_MAIN_CPU_VERSION_RSP   0x01
271 #define D_ECS_VERSION_INFO_MAIN_CPU_BUILD_QRY     0x02
272 #define D_ECS_VERSION_INFO_MAIN_CPU_BUILD_RSP     0x03
273
274 #define D_EC_EVEREST             0x75 ///< 117 Special op-code for Everest
275
276 /* EC_EVEREST sub opcodes, these go into the CMND1 byte */
277 #define D_ECS_EVEREST_AZIMUTH_ZERO_OFFSET_QRY     0x00
278 #define D_ECS_EVEREST_AZIMUTH_ZERO_OFFSET_RSP     0x01
279 #define D_ECS_EVEREST_SET_MAX_MAG                 0x02
280 #define D_ECS_EVEREST_MAX_MAG_QRY                 0x03
281 #define D_ECS_EVEREST_MAX_MAG_RSP                 0x04
282 #define D_ECS_EVEREST_ALARM_QRY                   0x05
283 #define D_ECS_EVEREST_ALARM_RSP                   0x06
284 #define D_ECS_EVEREST_DELETE_PATTERN              0x07
285 #define D_ECS_EVEREST_SET_MAN_PAN_LEFT_LIMIT      0x08
286 #define D_ECS_EVEREST_SET_MAN_PAN_RIGHT_LIMIT     0x09
287 #define D_ECS_EVEREST_SET_SCAN_PAN_LEFT_LIMIT     0x0A
288 #define D_ECS_EVEREST_SET_SCAN_PAN_RIGHT_LIMIT    0x0B
289
290 #define D_ECS_EVEREST_LIMIT_QRY                   0x0C
291 // ECD = Extended command data, in DATA2 for sub-opcode 0x0C
292 #define D_ECD_EVEREST_LIMIT_MAN_LEFT_PAN          0x00
293 #define D_ECD_EVEREST_LIMIT_MAN_RIGHT_PAN         0x01
294 #define D_ECD_EVEREST_LIMIT_SCAN_LEFT_PAN         0x02
295 #define D_ECD_EVEREST_LIMIT_SCAN_RIGHT_PAN        0x03
296
297 #define D_ECS_EVEREST_LIMIT_RSP                   0x0D
```

```
298
299 #define D_ECS_EVEREST_ENABLE_LIMITS              0x0E
300 // Enable limits constants, in DATA2 for sub-opcode 0x0E
301 #define D_ECD_EVEREST_ENABLE_LIMITS_DISABLE      0x00
302 #define D_ECD_EVEREST_ENABLE_LIMITS_ENABLE       0x01
303
304 #define D_ECS_EVEREST_DEFINED_PRESETS_QRY        0x0F
305 #define D_ECS_EVEREST_DEFINED_PRESETS_RSP        0x10
306 #define D_ECS_EVEREST_DEFINED_PATTERNS_QRY       0x11
307 #define D_ECS_EVEREST_DEFINED_PATTERNS_RSP       0x12
308
309
310 // Spectra IV added in a capability to have clock time inside a PTZ unit.
311 #define D_EC_TIMESET_MACRO_OPCODE 0x77   ///< 119 Time control opcode, for CMND2
312
313 // These sub values are in the CMND1 position.
314 // All "GET" codes generate a seven byte response.
315 // All of the "SET" codes generate a four byte response.
316 #define D_ECS_SET_SECONDS         0x00   ///< Loads in all values and a seconds value
317 #define D_ECS_GET_SECONDS         0x01   ///< Reads out the current seconds count
318 #define D_ECS_SET_HR_MIN          0x02   ///< Sets in the Hour/Min Value
319 #define D_ECS_GET_HR_MIN          0x03   ///< Reads out the current Hour/Min Value
320 #define D_ECS_SET_MON_DATE        0x04   ///< Sets in Month/Day value
321 #define D_ECS_GET_MON_DATE        0x05   ///< Reads out the current Month/Day value
322 #define D_ECS_SET_YEAR            0x06   ///< Sets in the Year value
323 #define D_ECS_GET_YEAR            0x07   ///< Reads out the current Year value
324
325
326
327 // For Atlas the following commands were added
328 #define D_EC_SCREEN_MOVE          0x79   ///< 121 Command to move relative (uses % of field of vision)
329 // Sub opcodes for CMND1
330 #define D_ECS_SET_ABS_MAG         0x00   ///< Set mag as an absolute value
331 #define D_ECS_SET_REL_MAG         0x01   ///< Set mag relative to current value
332
333
334
335 // Special Presets
336 #define PRESET_FLIP               33 ///< Standard in all units, causes a 180 degree turn
337 #define PRESET_ZERO               34 ///< Standard in all units, causes a pan to calibrate 0
338
339 #define PRESET_ERROR              71 ///< Unknown use
340
341 #define PRESET_MANUAL_UP_LIMIT    82 ///< Unknown use
342 #define PRESET_MANUAL_DOWN_LIMIT  83 ///< Unknown use
343 #define PRESET_AUX1               84 ///< Used with ExSite
344 #define PRESET_AUX2               85 ///< Used with ExSite
345 #define PRESET_WIPER              86 ///< Used with ExSite
346 #define PRESET_WASHER             87 ///< Used with ExSite
347 #define PRESET_AUTO_FOCUS_ON      88 ///< Unknown use
348 #define PRESET_IR_FILTER_IN       88 ///< Used with CBW type cameras
349 #define PRESET_AUTO_FOCUS_OFF     89 ///< Unknown use
350 #define PRESET_IR_FILTER_OUT      89 ///< Used with CBW type cameras
351
352 #define PRESET_MANUAL_LEFT_LIMIT  90 ///< Standard in all units, sets a left hand motion limit
353 #define PRESET_MANUAL_RIGHT_LIMIT 91 ///< Standard in all units, sets a right hand motion limit
354 #define PRESET_SCAN_LEFT_LIMIT    92 ///< Standard in all units, sets a left hand scanning limit
355 #define PRESET_SCAN_RIGHT_LIMIT   93 ///< Standard in all units, sets a right hand scanning limit
356 #define PRESET_RESET              94 ///< Unknown use, probably unimplemented
357 #define PRESET_WATCHDOG_RESET     94 ///< Unknown use, probably unimplemented
```

```
358  #define PRESET_MENU_MODE          95 ///< Standard in all units, accesses the menu system
359  #define PRESET_STOP_SCAN          96 ///< Standard in all units, stops automatic scans
360  #define PRESET_RANDOM_SCAN        97 ///< Standard in all units, starts random scanning
361  #define PRESET_FRAME_SCAN         98 ///< Standard in all units, starts frame scanning
362  #define PRESET_AUTO_SCAN          99 ///< Standard in all units, starts automatic scanning
363
364  #define PRESET_POSCAL            100 ///< Might be used with ExSite
365
366  #endif // #ifndef DPROTO_H
```

```
1  /// $Description: P Protocol command names $
2  /// $Workfile: PProto.h$
3  /// Copyright(&copy;) by Pelco, 2007
4  /// $Header: PProto.h: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:16 AM$
5  #define doxygen_pproto_h
6
7  #ifndef PPROTO_H
8  #define PPROTO_H
9
10 ///
11 /// P Protocol command format, all commands are 8 bytes long as follows:
12 /// - STX 0xA0
13 /// - Address, "1" based, i.e. 0 = address one
14 /// - CMND1, first byte of a bit encoded command
15 /// - CMND2, second byte of a bit encoded command or all of a numeric command
16 /// - DATA1, pan speed index or additional data for a numeric command
17 /// - DATA2, tilt speed index or additional data for a numeric command
18 /// - ETX 0xAF
19 /// - CKSM, xor of all bytes between and including STX --> ETX
20 #define doxygen_pprotocol_format
21
22 #define P_START 0 ///< Sync 0xFF, not in checksum
23 #define P_ADDR  1 ///< Address, "1" based, i.e. 0 = address one
24 #define P_CMND1 2 ///< First byte of a bit encoded command
25 #define P_CMND2 3 ///< Second byte of a bit encoded command or all of a numeric command
26 #define P_DATA1 4 ///< Pan speed index or additional data for a numeric command
27 #define P_DATA2 5 ///< Tilt speed index or additional data for a numeric command
28 #define P_END   6 ///< End of data
29 #define P_CKSM  7 ///< Least significant byte of the XOR bytes 0 ---> 6
30
31
32 // P command types
33 #define P_BIT_ENCODED 0x01 ///< All bit encoded commands have bit 0 = 0
34
35
36 // P command bits for bit encoded commands in Pcmnd1 and Pcmnd2
37 #define P_CLOSE   0x08    ///< Pcmnd1
38 #define P_OPEN    0x04    ///< Pcmnd1
39 #define P_NEAR    0x02    ///< Pcmnd1
40 #define P_FAR     0x01    ///< Pcmnd1
41
42 #define P_WIDE    0x40    ///< Pcmnd2, Identical usage for D Protocol
43 #define P_TELE    0x20    ///< Pcmnd2, Identical usage for D Protocol
44
45 #define P_DOWN    0x10    ///< Pcmnd2 plus Pdata2, Identical usage for D Protocol
46 #define P_UP      0x08    ///< Pcmnd2 plus Pdata2, Identical usage for D Protocol
47 #define P_LEFT    0x04    ///< Pcmnd2 plus Pdata1, Identical usage for D Protocol
48 #define P_RIGHT   0x02    ///< Pcmnd2 plus Pdata1, Identical usage for D Protocol
49
50
51 // Standard command length
52 #define P_EC_COMMAND_LENGTH     8   ///< Length of P protocol commands
53
54 // All replies are one byte long
55 #define P_STX             0xA0   ///< Start of text
56 #define P_ETX             0xAF   ///< End of text
57 #define P_ACK             0xA2   ///< Acknowledgment for a good message
58 #define P_NAK             0xAA   ///< Negative Acknowledgment for a bad message
59
60 // Supported numerically encoded commands from the Spectra I and Intercept
```

```
 61  // All of these commands perform the same as they would in D Protocol.
 62  // Most of the commented out commands, work with D Protocol.
 63  #define P_EC_SET_PRESET       3    ///< Set a preset
 64  #define P_EC_CLEAR_PRESET     5    ///< Clear a preset
 65  #define P_EC_MOVE_PRESET      7    ///< Move to preset
 66  #define P_EC_SET_AUX          9    ///< Turn on an aux
 67  #define P_EC_CLEAR_AUX       11    ///< Turn off an aux
 68  //#define P_EC_DUMMY_1       13    ///< Dummy command
 69  #define P_EC_RESET           15    ///< Reset
 70  #define P_EC_ZONE_START      17    ///< Set zone start
 71  #define P_EC_ZONE_END        19    ///< Set zone end
 72  #define P_EC_WRITE_CHAR      21    ///< Write character on screen
 73  #define P_EC_CLEAR_SCREEN    23    ///< Clear the screen
 74  //#define P_EC_ALARM_ACK     25    ///< Acknowledge alarms
 75  #define P_EC_ZONE_ON         27    ///< Set zone scan on
 76  #define P_EC_ZONE_OFF        29    ///< Set zone scan off
 77  #define P_EC_START_RECORD    31    ///< Start pattern record
 78  #define P_EC_END_RECORD      33    ///< End pattern record
 79  #define P_EC_START_PLAY      35    ///< Start pattern play
 80  #define P_EC_ZOOM_SPEED      37    ///< Set zoom speed
 81  //#define P_EC_FOCUS_SPEED   39    ///< Set focus speed
 82  #define P_EC_CAMERA_RESET    41    ///< Reset the camera
 83  #define P_EC_AUTO_FOCUS      43    ///< Set auto focus to device, off, or on
 84  #define P_EC_AUTO_IRIS       45    ///< Set auto iris to device, off, or on
 85  //#define P_EC_AGC           47    ///< Set AGC to device, off, or on
 86  #define P_EC_BLC             49    ///< Set backlight compensation off or on
 87  #define P_EC_AWB             51    ///< Set auto white balance off or on
 88  //#define P_EC_DEVICE_PHASE  53    ///< Set device phase delay
 89  #define P_EC_SHUTTER_SPEED   55    ///< Set shutter speed
 90  #define P_EC_ADJUST_PHASE    57    ///< Adjust phase delay
 91  #define P_EC_ADJUST_RB_WB    59    ///< Adjust red-blue white balance
 92  #define P_EC_ADJUST_MG_WB    61    ///< Adjust magenta-green white balance
 93  //#define P_EC_ADJUST_GAIN   63    ///< Adjust camera gain
 94  #define P_EC_ADJUST_AI_LEVEL 65    ///< Adjust auto-iris level
 95  #define P_EC_ADJUST_AI_PEAK  67    ///< Adjust auto_iris peak value
 96  //#define P_EC_QUERY       0x45    ///< Get the program number
 97  #define P_EC_PRESET_SCAN   0x47    ///< Preset scan
 98  // none of the newer commands are available
 99
100  #endif // #ifndef PPROTO_H
101
```

# M Special Modification Request (SMRs) commands and replies

## M.1 SMR-X Command 0xF1 ($241_{10}$), Set Speed Multiplier

### M.1.1 Command format

| Set Speed Multiplier D_EC_xxx | | | | | | | |
|---|---|---|---|---|---|---|---|
| Byte | 1 SYNC | 2 ADDR | 3 CMND1 | 4 CMND2 | 5 DATA1 | 6 DATA2 | 7 CKSM |
| | 0xFF | — | 0xF1 | 0x00 | 0x00 | SPEED MULTIPLIER | — |

FPN

1. This command generates a "General Response".
2. See subsection M.2 for more information about this command.
3. This information came from several e-mails between Craig Hannen and Chien-Min Huang dated 21APR06 and an additional e-mail between Craig Hannen and morison.6@osu.edu dated 07SEP06.

| Used/Implemented on | | | | | | |
|---|---|---|---|---|---|---|
| | Spectra II | Spectra III | Spectra MINI | Spectra IV | Esprit | ExSite |
| Decoded | No | SMR | No | SMR | SMR | No |
| In P | No | No | No | No | No | No |
| General | Yes | Yes | Yes | Yes | Yes | Yes |

### M.1.2 Description

1. SPEED MULTIPLIER See subsection M.2 for a full description of the use of this command.

   **Example:** to set a speed multiplier of 1, FF 01 00 F1 00 01 F3 would be sent.

[104] $Header: SMR-X-F1.inc: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:20 AM$

## M.2 SMR-X Command 0xF3 ($243_{10}$), Set Minimum Variable Speed

### M.2.1 Command format

| Set Minimum Variable Speed<br>D_EC_xxx | | | | | | | |
|---|---|---|---|---|---|---|---|
| Byte | 1<br>SYNC | 2<br>ADDR | 3<br>CMND1 | 4<br>CMND2 | 5<br>DATA1 | 6<br>DATA2 | 7<br>CKSM |
| | 0xFF | — | 0xF3 | 0x00 | 0x00 | MIN VAR SPEED | — |

FPN

1. This command generates a "General Response".
2. See subsection M.1 for more information about this command.
3. This information came from several e-mails between Craig Hannen and Chien-Min Huang dated 21APR06 and an additional e-mail between Craig Hannen and morison.6@osu.edu dated 07SEP06.

| Used/Implemented on | | | | | | |
|---|---|---|---|---|---|---|
| | Spectra II | Spectra III | Spectra MINI | Spectra IV | Esprit | ExSite |
| Decoded | No | SMR | No | SMR | SMR | No |
| In P | No | No | No | No | No | No |
| General | Yes | Yes | Yes | Yes | Yes | Yes |

### M.2.2 Description

1. MIN VAR SPEED The minimum variable speed represents the rotational speed in 10ths of degrees per second.

   **Example:** To set a min variable speed of 1° /sec, `FF 01 00 F3 00 0A FE` would be sent.

### M.2.3 Calculations

1. The way the SMR-X calculates max preset pan speed is:

```
speed = max_speed - min_var_speed;       // max_speed is 360 deg/s for
                                         // Spectra III SE and 250 deg/s
                                         // for regular Spectra III.
speed = (speed * distance) / 18000;      // speed is made proportional to
                                         // the angular pan distance of travel
                                         // to the preset.
                                         // Max angular pan distance is 18000
                                         // (or 180 hundredths of degrees).
speed = speed + min_var_speed;           // speed is offset by min_var_speed
speed = (speed * speed_multiplier) / 10; // speed is scaled by speed_multiplier/10.
```

2. The way the SMR-X calculates max preset tilt speed is similar:

```
speed = max_speed - min_var_speed;       // In this case max_speed is
                                         // 200 deg/s for any Spectra III.
speed = (speed * distance) / 9000;       // speed is made proportional to the
                                         // angular tilt distance of travel
                                         // to the preset.
                                         // Max angular tilt distance is 9000
                                         // (or 90 hundredths of degrees).
speed = speed + min_var_speed;           // speed is offset by min_var_speed
speed = (speed * speed_multiplier) / 10; // speed is scaled by speed_multiplier/10.
```

[105] $Header: SMR-X-F3.inc: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:21 AM$

As can be seen max preset speed (or "speed" in the above) is scaled by the distance to the preset position as well as the `speed_multiplier` that's defined in the D Protocol command (subsection M.1). This SMR only uses this speed calculation in association with the set-pan-position (subsection 5.38) and set-tilt-position (subsection 5.39) commands (not the go-to-preset (subsection 5.4) command).

# N Time Setting Commands in D Protocol

Robert D. Sexton
Sr. Software Engineer
Pelco
March 2, 2006

## N.1 Overview

The Pelco D Protocol does not include any remote time setting commands, simply because a real-time clock has never before been implemented. For Spectra III+[107], such a feature is included and requires an extension in the protocol to accommodate this.

A time setting function must be handled as a set of commands. This is because the structure of D packets includes only four bytes of data, including the command identifier. Effectively, only two bytes (16 bits) of data can be passed to or from the receiver per transmission.

One solution to this limitation is effectively to extend the command value to 16 bits, rather than the 8 bits of the standard commands. This method has been employed in the Version Information Macro Opcode, and the Everest Macro Opcode commands. These commands employ a primary opcode (`0x73` and `0x75` respectively) and a set of sub-opcodes to expand the utility of the command. The opcode is transmitted as data byte 2, the sub-opcode as byte 1, and any data as bytes 3 and 4. Where data is returned, the same opcode is used with a different sub-opcode. Where the command sends data to the receiver, the responses are `ACK` or `NACK` in the extended response format. It must also be noted that the opcodes are limited to odd numbers, the least significant bit must be binary 1 to identify it as an opcode as opposed to a motion command.

## N.2 Proposed Time Setting Commands

According to the D Protocol document, V4.1 dated April 11, 2005, the last assigned commands are the two described above. The opcode value next in line is `0x77`. It is proposed that this value be reserved for time setting and reading functions. The sub-opcodes will be assigned such that even values (`0x00`, `0x02`, etc.) will be time setting commands, and odd value (`0x01`, `0x03`, etc.) will be time reporting sub-opcodes.

## N.3 Sub-opcode usage

| | |
|---|---|
| `0x00` | Set seconds and synchronize time |
| `0x01` | Report seconds |
| `0x02` | Set hour and minutes |
| `0x03` | Report hour and minutes |
| `0x04` | Set month and date |
| `0x05` | Report month and date |
| `0x06` | Set year |
| `0x07` | Report year |

Set commands return an `ACK` if successful, or `NACK` if not. Report commands will return a `NACK` if time is not set. In order to assure a successful time setting operation, time and date should be sent top-down, that is, year, month-date, hour-minute, seconds. Similarly, a time request will only be guaranteed to be correct if no intervening transmissions occur.

[106] `$Header: TimeCmnd.inc: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:23 AM$`
[107] Now named the Spectra IV.

| 0x00 Set seconds and synchronize time | |
|---|---|
| byte 3 | 0x00 |
| byte 4 | Seconds to set (0-59) |

On receipt of this command, the receiver clock will be set to the time and date previously transmitted. If an unrelated transmission occurs between this command and other time setting commands, no action will be taken. The response to this command is ACK if the seconds value is in range and time is set. It is NACK if out of range or time is not set.

| 0x01 Report seconds | |
|---|---|
| byte 3 | 0x00 |
| byte 4 | Current value of seconds. |

When reporting any segment of the clock data, the current time will be read and held in a buffer until an unrelated transmission occurs. Thus no ambiguities or anomalies will be reported if the time requests occur in succession.

| 0x02 Set hour and minutes | |
|---|---|
| byte 3 | Hour to set (0-23) |
| byte 4 | Minutes to set (0-59) |

Time will always be transmitted in 24-hour format. That is, midnight is 00:00, etc.

| 0x03 Report hour and minutes | |
|---|---|
| byte 3 | Current hour. |
| byte 4 | Current minute. |

| 0x04 Set month and date | |
|---|---|
| byte 3 | Month (1-12) |
| byte 4 | Date (1-31) |

The date will be out of range if not valid for the month specified. If the month specified is February, the range is limited to 1 — 28 unless the year has been determined to be a leap year, in which case 29 is an acceptable value. If the year has not been set, it is assumed that it is not a leap year.

| 0x05 Report month and date | |
|---|---|
| byte 3 | Current month. |
| byte 4 | Current date. |

Month and date are reported in the same format as the corresponding set command.

| 0x06 Set year | |
|---|---|
| byte 3 | MSB of year. |
| byte 4 | LSB of year. |

The year may be sent as a complete value (i.e. 2006) or as the last two digits (i.e. 06.) If only the last two digits are sent, the century value is assumed to be 2000 and is added to value sent to determine the year.

| `0x07` Report year | |
|---|---|
| byte 3 | MSB of year. |
| byte 4 | LSB of year. |

The year reported is always the absolute value, that is, 2006 is always sent as 2006, not 06.

# O Future Protocol Directions

**Note**

This is from a note used on the Spectra IV project.

Debug and Special Purpose Extensions for D Protocol
Robert D. Sexton
Sr. Software Engineer
Pelco
April 4, 2006

## O.1 Overview

This document describes a method of extension and declares a debug command set space.

The D Protocol is a master-slave system that sends commands embedded in a packet consisting of a non-exclusive flag character followed by an address character, four data characters and a checksum. The data characters consist of (in transmission order) a sub-command, command and two data characters.

The D Protocol requires that non-PTZ commands be odd numbered. This is required to distinguish PTZ commands from other commands. Command values are assigned from `0x01` → `0x75` and `0xFD` and `0xFF`. In most commands the sub-command is `0x00`. In a few, it is utilized to expand the command capabilities.

## O.2 Proposed Debug Command Set Space

To accommodate the need for varying debug uses, it is proposed that the command number range from `0xF1` to `0xFB` be reserved for developer determined uses. Commands implemented in this range must be installed solely for developmental purposes and should not be included in any product release. It is not necessary that these commands be formally documented, nor is it required that they be consistently applied from project to project. Such commands as are determined to have future permanent utility should be remapped to the special purpose command space.

## O.3 Proposed Special Purpose Command Set Space

To accommodate the need for application or platform specific extensions to the D Protocol, it is proposed that the command number range from `0xE1` to `0xEF` be reserved for platform specific uses. Commands implemented in this range must be formally documented and where used on other platforms, must be consistently applied. Once implemented and documented, these commands may not be redefined. However, it is not required or expected that they be implemented on every platform.

## O.4 General Implementation Notes

Debug and Special Purpose commands should be implemented using the sub-command value. This greatly expands the number of commands possible (1536 for debug and 2048 for special purpose.) The response to these commands should be in the extended format with the command number one greater than the received command. In the debug space, this requirement may be relaxed to invoke any necessary response the developer requires. In the special purpose space, the extended response must be used to minimize the possibility of causing equipment errors in a master unit not designed for the result.

[108] `$Header: Future.inc: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:11 AM$`

# P List of Effective Sections

1. Header: 01.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
2. Header: 03.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
3. Header: 05.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
4. Header: 07.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
5. Header: 09.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
6. Header: 0B.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
7. Header: 0D.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
8. Header: 0F.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
9. Header: 11.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
10. Header: 13.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
11. Header: 15.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
12. Header: 17.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
13. Header: 19.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
14. Header: 1B.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
15. Header: 1D.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
16. Header: 1F.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
17. Header: 21.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
18. Header: 23.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
19. Header: 25.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
20. Header: 27.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
21. Header: 29.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
22. Header: 2B.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
23. Header: 2D.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
24. Header: 2F.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
25. Header: 31.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
26. Header: 33.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:45 PM$
27. Header: 35.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
28. Header: 37.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
29. Header: 39.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
30. Header: 3B.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
31. Header: 3D.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
32. Header: 3F.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
33. Header: 41.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
34. Header: 43.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
35. Header: 45.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
36. Header: 47.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
37. Header: 49.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
38. Header: 4B.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
39. Header: 4D.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
40. Header: 4F.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
41. Header: 51.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
42. Header: 53.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
43. Header: 55.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
44. Header: 57.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
45. Header: 59.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
46. Header: 5B.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
47. Header: 5D.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
48. Header: 5F.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
49. Header: 61.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$

[109]File was generated by FixGlo.l and refit.bat

50. Header: 63.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
51. Header: 65.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
52. Header: 67.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
53. Header: 69.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
54. Header: 6B.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
55. Header: 6D.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
56. Header: 6F.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
57. Header: 71.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
58. Header: 73.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
59. Header: 75.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
60. Header: 77.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
61. Header: 79.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
62. Header: 7B.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:12 PM$
63. Header: CBytes.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:12 PM$
64. Header: CC.inc: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:04 AM$
65. Header: ChngeLog.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
66. Header: Commands.inc: Revision: 3: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$
67. Header: DBits.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:12 PM$
68. Header: DBits.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:12 PM$
69. Header: Dbytes.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:12 PM$
70. Header: Dbytes.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:12 PM$
71. Header: Intro.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:13 PM$
72. Header: Labels.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:13 PM$
73. Header: PBits.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:13 PM$
74. Header: PBits.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:13 PM$
75. Header: Pbytes.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:13 PM$
76. Header: Pbytes.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:13 PM$
77. Header: PDF.tex: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:13 PM$
78. Header: Presets.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:13 PM$
79. Header: PTdata.inc: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:17 AM$
80. Header: Response.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:13 PM$
81. Header: Revs.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 01:37:13 PM$
82. Header: Sp3Down.inc: Revision: 1: Author: ehamilton: Date: 01/25/2012 11:29:21 AM$

# Q Change Log

1. March 2008, revision 5.0.1:
   - 1.1 Several spelling and grammar fixes.
   - 1.2 Added in information about reading out the internal temperature of a Spectra IV.
   - 1.3 Added in information about features of the Esprit series of PTZs gained from reading the "Esprit Software Release" log for software revision 3.80.
2. June 2008, revision 5.0.2:
   - 2.1 Several spelling and grammar fixes.
   - 2.2 Added in cross references between reply types and their definitions.
3. March 2010, revision 5.1.1:
   - 3.1 Added hyper links to the document.
   - 3.2 Added new information about the query command (subsection 5.35).
   - 3.3 Minor english fixes through out.
4. April 2010, revision 5.1.2:
   - 4.1 Documented op-code `0x7B`, which was missed earlier. It has been in shipping products for at least 6 months.
5. August 2011, revision 5.2.0:
   - 5.1 Incorporate many spelling, grammer and similar problems.
   - 5.2 Add an Effective Sections, part.
6. September 2011, revision 5.2.1:
   - 6.1 Incorporate spelling fixes.
   - 6.2 Added a note for opcodes 0x5F and 0x79 that their arguments are the only ones in D Protocol that are signed.
   - 6.3 Added in a speed plot and table for the current Esprit.
7. February 2012, revision 5.2.2:
   - 7.1 Incorporate spelling fixes.
   - 7.2 Made it clear which numbers are in decimal with many externally invisible internal fixes.
   - 7.3 Put into Surround instead of using RCS for source control.

[110] `$Header: ChngeLog.inc: Revision: 2: Author: ehamilton: Date: 02/20/2012 02:32:46 PM$`

# Index