Sentinel Camera Communication Protocols Specification Document (Draft 2c)

Peter McCarthy
December 07, 2002

The purpose of this draft specification document is to detail a *generic* Sentinel communications protocol to facilitate zero-configuration, 'invisible' networking of camera devices.

Note: This is a confidential document and is for discussion purposes only.

Communication Protocols

The recommended method of communications comprises the exchange of XML messages, using the TCP/IP protocol, between the Sentinel control PC and camera devices.

However, it is recognized at this moment that camera devices typically utilize serial protocols. Therefore, if it is not possible to implement a TCP/IP solution, it is recommended that communications comprise binary message exchanges using the (full-duplex) RS422 protocol.

Consequently, this specification is concerned with defining a serial-based communications protocol.

Communications based on TCP/IP will be supplied in a separate specification document.

Connection Settings

The recommended serial connection settings are as follows:

Baud:	9600
Data Bits:	8
Parity:	None
Stop Bits:	1

It is also advised that all software and hardware handshaking (e.g. RTS/CTS and XON/XOFF) be ignored for the sake of simplicity.

Message Format

To be classified as Sentinel 'compatible', camera devices should support a binary message format based on the following:

0	1..2	3..n	n+1..		..n_1-1	n_1
STX	MsgID	DevID	$Param_1$	.. ..	$Param_m$	ETX

- *STX* represents the start of message character
- *MsgID* represents the message identifier. Message identifiers 01..7F (hexadecimal) are reserved for messages sent from the Sentinel control

PC **to** camera devices. Message identifiers 81..FF are reserved for messages sent **from** devices to the control PC

- *DevID* represents the identifier of the camera device that is to receive the message, or the device that sent the message. Depending on the 'class' of message, this will be either a universally unique identifier or a network identifier (i.e. address)

- $Param_1..Param_m$ represent one or more message parameters. These will depend on the class of message

- *ETX* represents the end of message character

Message Classes

To be classified as Sentinel compatible, camera devices must support the following message classes: *Discovery*, *Description*, and *Control*.

Discovery

In order for a camera device to join and participate in a Sentinel network, a process of discovery must be undertaken. The discovery process represents an exchange of messages between the control PC and a camera device, as shown in Table 1 below:

Control PC		Camera Device
When an *Address Discover* message is received from a camera device, the universally unique identifier (*UUID*) of the device is extracted from the message by the control PC The control PC then sends an *Address Offer* message back to the camera device identified by the *UUID*	←	On power-up, a camera device sends an *Address Discover* message – say – every 10 seconds
	→	When an *Address Offer* message is received from the control PC, the camera device sets its internal *NetID* (i.e. address) 'property' and stops sending *Address Discover* messages

Table 1

Also, the control PC can query – at any time – for the presence of existing camera devices on the network. The message exchange is shown in Table 2 below:

Control PC		Camera Device
The control PC sends a *Query* message to a camera device	→	When a *Query* message is received,

If the control PC does not receive a *Query Result* message from the camera device within a certain time period, further *Query* messages are sent until either a *Query Result* message is received, or the control PC decides that the device is no longer present on the network	←	the camera device replies immediately by sending a *Query Result* message to the control PC

Table 2

As well as providing the means for a camera device to **join** a Sentinel network, the discovery process also enables a device to advertise the fact that it is about to **leave** the network. This in turn enables the control PC to re-configure the Sentinel system appropriately.

Therefore, before a camera device is powered off, the device must attempt to send a *Bye Bye* message to the control PC to indicate that it is about to gracefully leave the network. If it is not possible for a camera device to do this, the control PC will eventually learn that the device is no longer present through the use of the *Query* message (via polling techniques).

Address Discover (81)

The format is:

0	1..2	3.. ..34	35
STX	MsgID	UUID	ETX

- Byte 0 represents the start of message character (ASCII 0x02)
- Bytes 1..2 represent the *Address Discover* message ID. These bytes consist of the hexadecimal characters ASCII 0x38, ASCII 0x31 (i.e. '81')
- Bytes 3..34 represent the 128-bit universally unique ID of the camera device. These bytes consist of 32 hexadecimal ASCII characters
- Byte 35 represents the end of message character (ASCII 0x03)

An example is:

0	1..2	3.. ..34	35
	81	C74DE0038D0911D6A1C50002E321E79C	

This message indicates that a camera device, with a *UUID* of C74DE0038D0911D6A1C50002E321E79C, has joined the Sentinel network and requires a network identifier (i.e. address) to be allocated to it

---

Address Offer (01)

The format is:

0	1..2	3.. ..34	35..38	39
STX	MsgID	UUID	NetID	ETX

- Byte 0 represents the start of message character (ASCII 0x02)
- Bytes 1..2 represent the *Address Offer* message ID. These bytes consist of the hexadecimal characters ASCII 0x30, ASCII 0x31 (i.e. '01')
- Bytes 3..34 represent the 128-bit universally unique ID of the camera device. These bytes consist of 32 hexadecimal ASCII characters
- Bytes 35..38 represent the 16-bit Sentinel network address assigned to the camera device. These bytes consist of 4 hexadecimal ASCII characters
- Byte 39 represents the end of message character (ASCII 0x03)

An example is:

0	1..2	3.. ..34	35..38	39
	01	C74DE0038D0911D6A1C50002E321E79C	0001	

This message offers a *NetID* of 0001 to a camera device that has a *UUID* of C74DE0038D0911D6A1C50002E321E79C

Query (02)

The format is:

0	1..2	3	4.. ..n-1	n
STX	MsgID	Mask	Parameters	ETX

- Byte 0 represents the start of message character (ASCII 0x02)
- Bytes 1..2 represent the *Query* message ID. These bytes consist of the hexadecimal characters ASCII 0x30, ASCII 0x32 (i.e. '02')
- Byte 3 represents a 4-bit query parameter mask encoded as a single hexadecimal ASCII character. The mask comprises the following bits:

3	2	1	0
UUID	NetID	unused	unused

  - Bit 3 indicates that a *UUID* is being queried
  - Bit 2 indicates that a *NetID* is being queried
  - Bits 1..0 are unused
- Bytes 4..n-1 represent the query parameter values that must be matched by a camera device if that device is to reply with a *Query Result* message. These values must be packed according to the order in which they appear in the mask
- Byte n represents the end of message character (ASCII 0x03)

Examples are:

0	1..2	3	4.. ..35	36
☺	02	8	C74DE0038D0911D6A1C50002E321E79C	♥

This message queries for the network presence of a camera device that has a *UUID* of C74DE0038D0911D6A1C50002E321E79C

0	1..2	3	4..7	8
☺	02	4	000F	♥

This message queries for the network presence of a camera device that has a *NetID* of 000F

0	1..2	3	4.. ..39	40
☺	02	C	C74DE0038D0911D6A1C50002E321E79C002A	♥

This message queries for the network presence of a camera device that has a *UUID* of C74DE0038D0911D6A1C50002E321E79C **and** a *NetID* of 002A

Query Result (82)

The format is:

0	1..2	3	4.. ..n-1	n
STX	MsgID	Mask	Parameters	ETX

- Byte 0 represents the start of message character (ASCII 0x02)
- Bytes 1..2 represent the *Query Result* message ID. These bytes consist of the hexadecimal characters ASCII 0x38, ASCII 0x32 (i.e. '82')
- Byte 3 represents a 4-bit query parameter mask encoded as a single hexadecimal ASCII character. This is copied from the *Query* message being replied to
- Bytes 4..n-1 represent the query parameters values matched. These are copied from the *Query* message being replied to
- Byte n represents the end of message character (ASCII 0x03)

An example is:

0	1..2	3	4.. ..35	36
☺	82	8	C74DE0038D0911D6A1C50002E321E79C	♥

This message was returned, in response to a *Query* message, by a camera device that has a *UUID* of C74DE0038D0911D6A1C50002E321E79C

Bye Bye (83)

The format is:

0	1..2	3.. ..34	35
STX	MsgID	UUID	ETX

- Byte 0 represents the start of message character (ASCII 0x02)
- Bytes 1..2 represent the *Bye Bye* message ID. These bytes consist of the hexadecimal characters ASCII 0x38, ASCII 0x33 (i.e. '83')
- Bytes 3..34 represent the 128-bit universally unique ID of the camera device. These bytes consist of 32 hexadecimal ASCII characters
- Byte 35 represents the end of message character (ASCII 0x03)

An example is:

0	1..2	3.. ..34	35
☻	83	C74DE0038D0911D6A1C50002E321E79C	♥

This message advertises the fact that a camera device, with a *UUID* of C74DE0038D0911D6A1C50002E321E79C, is about to leave the network

---

Description

When a camera device is attached to a Sentinel network and has been discovered, the control PC will still know very little about that device.

To enable the control PC to learn about what type of camera device has joined the network, and what the device's capabilities are, a device description must be retrieved. This description is stored in the device as a set of read-only 'properties' (see Table 3):

Property	Description	Set	Get
ManufacturerID	The manufacturer of the camera device. This is hard-coded during manufacture, and is an integer value in the range 001..999. A list of valid values will be available in future in a revised specification, and is subject to customer agreement		✓
ModelNo	The model number of the camera device. This is hard-coded during manufacture, and is an integer value in the range 001..999. A list of valid values will be available in future in a revised specification, and is subject to customer agreement		✓
SerialNo	The serial (i.e. product) number of the camera device. This is hard-coded during manufacture, and is an integer value in the range 00000001..99999999		✓
Type	The type of camera device, for example static, PTZ, or dummy. This is hard-coded during manufacture, and is an integer value in the range 01..99, where:		✓

	▪ 01 == Static camera device ▪ 02 == PTZ camera device ▪ 03 == Dummy camera device ▪ 04..99 == reserved		
UUID	The universally unique identifier (UUID) of the camera device. The UUID is a 128-bit integer value, hard-coded during manufacture, and guaranteed never to be duplicated by any other manufacturer for any type of device. This can be generated on any PC running Windows 95/98/Me/2000/XP. A software tool will be provided to generate UUIDs, if required		✓

Table 3

To retrieve the description for a particular camera device, the control PC must send a *Get Description* message.

When a device receives this message, it replies by sending a *Description* message back to the control PC.

Get Description (11)

The format is:

0	1..2	3..6	7
STX	MsgID	NetID	ETX

- Byte 0 represents the start of message character (ASCII 0x02)
- Bytes 1..2 represent the *Get Description* message ID. These bytes consist of the hexadecimal characters ASCII 0x31, ASCII 0x31 (i.e. '11')
- Bytes 3..6 represent the 16-bit Sentinel network address assigned to the camera device. These bytes consist of 4 hexadecimal ASCII characters only
- Byte 7 represents the end of message character (ASCII 0x03)

An example is:

0	1..2	3..6	7
	11	0042	

This message requests the description from a camera device that has a *NetID* of 0042

Description (91)

The format is:

0	1..2	3..6	7.. ..22	23
STX	MsgID	NetID	Description	ETX

This can be broken down further into the following format:

0	1..2	3..6	7..9	10..12	13..20	21..22	23
STX	MsgID	NetID	ManufacturerID	ModelNo	SerialNo	Type	ETX

- Byte 0 represents the start of message character (ASCII 0x02)
- Bytes 1..2 represent the *Description* message ID. These bytes consist of the hexadecimal characters ASCII 0x39, ASCII 0x31 (i.e. '91')
- Bytes 3..6 represent the 16-bit Sentinel network address assigned to the camera device. These bytes consist of 4 hexadecimal ASCII characters
- Bytes 7..9 represent the device manufacturer ID. These consist of 3 numerical ASCII characters
- Bytes 10..12 represent the device model number. These consist of 3 numerical ASCII characters
- Bytes 13..20 represent the device serial number. These consist of 8 numerical ASCII characters
- Bytes 21..22 represent the device type. These consist of 2 numerical ASCII characters
- Byte 23 represents the end of message character (ASCII 0x03)

An example is:

0	1..2	3..6	7..9	10..12	13.. ..20	21..22	23
[illegible]	91	007B	01	002	00008417	01	[illegible]

This message was returned from a (static) camera device that has a *NetID* of 007B

Control

Once the discovery and description steps are complete, the Sentinel control PC can then control the camera device. Control is defined as four sub-steps: *Configuration* and *Dynamics*, *Diagnostics*, and *Telemetry*.

Configuration

Each camera device must store a number of read-only and read-write configuration 'properties', that can be set and/or got by the control PC. These are detailed in Table 4 below:

Property	Description	Set	Get
NetID	The network identifier (i.e. address) allocated to the camera device by the control PC. This is established during the *Discovery* step, and can only be read during subsequent steps. This is a 16-bit integer value		✓
Locale	The geographical location (i.e. site number) where the camera device is currently installed. This is an integer value in the range	✓	✓

	0001..9999		
SocketNo	The rail socket into which the camera device is plugged. This is an integer value in the range 001..999	✓	✓
InputNo	The matrix video input into which the camera device is connected. This is an integer value in the range 001..999	✓	✓

Table 4

To get configuration properties, the control PC must send a *Get Configuration* message to a particular camera device.

When a device receives a *Get Configuration* message, it replies immediately with a *Configuration* message

To set configuration properties, the control PC must send a *Set Configuration* message.

Get Configuration (21)

The format is:

0	1..2	3..6	7..8	9
STX	MsgID	NetID	Mask	ETX

- Byte 0 represents the start of message character (ASCII 0x02)
- Bytes 1..2 represent the *Get Configuration* message ID. These bytes consist of the hexadecimal characters ASCII 0x32, ASCII 0x31 (i.e. '21')
- Bytes 3..6 represent the 16-bit Sentinel network address assigned to the camera device. These bytes consist of 4 hexadecimal ASCII characters
- Bytes 7..8 represent an 8-bit property value mask encoded as 2 hexadecimal ASCII characters. The mask comprises the following bits:

7	6	5	4	3..0
Address	Locale	SocketNo	InputNo	unused

  - Bit 7 indicates that the *Address* property value is required
  - Bit 6 indicates that the *Locale* property value is required
  - Bit 5 indicates that the *SocketNo* property value is required
  - Bit 4 indicates that the *InputNo* property value is required
  - Bits 3..0 are unused
- Byte 9 represents the end of message character (ASCII 0x03)

An example is:

0	1..2	3..6	7..8	9
	21	0007	30	

This message requests the *SocketNo* **and** *InputNo* configuration property values from a camera device that has a *NetID* of 0007

Configuration (A1)

The format is:

0	1..2	3..6	7..8	9.. ..n-1	n
STX	MsgID	NetID	Mask	Properties	ETX

- Byte 0 represents the start of message character (ASCII 0x02)
- Bytes 1..2 represent the *Configuration* message ID. These bytes consist of the hexadecimal characters ASCII 0x41, ASCII 0x31 (i.e. 'A1')
- Bytes 3..6 represent the 16-bit Sentinel network address assigned to the camera device. These bytes consist of 4 hexadecimal ASCII characters
- Bytes 7..8 represent an 8-bit property value mask encoded as 2 hexadecimal ASCII characters. This is copied from the *Get Configuration* message being replied to
- Bytes 9..n-1 represent the properties values requested. These values are packed according to the order in which they appear in the mask
- Byte n represents the end of message character (ASCII 0x03)

An example is:

0	1..2	3..6	7..8	9..14	15
	A1	0007	30	015012	

This message has been sent in response to a *Get Configuration* message. The camera device has a *NetID* of 0007, and has returned values for its *SocketNo* **and** *InputNo* properties as 015 and 012 respectively

---

Set Configuration (22)

The format is:

0	1..2	3..6	7..8	9.. ..n-1	n
STX	MsgID	NetID	Mask	Properties	ETX

- Byte 0 represents the start of message character (ASCII 0x02)
- Bytes 1..2 represent the *Set Configuration* message ID. These bytes consist of the hexadecimal characters ASCII 0x32, ASCII 0x32 (i.e. '22')
- Bytes 3..6 represent the 16-bit Sentinel network address assigned to the camera device. These bytes consist of 4 hexadecimal ASCII characters
- Bytes 7..8 represent an 8-bit property value mask encoded as 2 hexadecimal ASCII characters. The mask comprises the following bits:

7	6	5	4	3..0
unused	Locale	SocketNo	InputNo	unused

- Bit 7 is unused
- Bit 6 indicates that the *Locale* property value is to be set
- Bit 5 indicates that the *SocketNo* property value is to be set
- Bit 4 indicates that the *InputNo* property value is to be set
- Bits 3..0 are unused

- Bytes 9..n-1 represent the properties values to be set. These values are packed according to the order in which they appear in the mask
- Byte n represents the end of message character (ASCII 0x03)

An example is:

0	1..2	3..6	7..8	9.. ..18	19
;	22	00A5	40	3379002032	^

This message sets the *Locale* (i.e. site number), *SocketNo*, and *InputNo* configuration property values – for a camera device with *NetID* 00A5 – to 3379, 002, and 032 respectively

---

Dynamics

Each camera device must store a number of read-only dynamics 'properties' that can be got by the control PC. These are detailed in Table 5 below:

Property	Description	Set	Get
PresetRange	The range of user/application definable Sentinel presets (see Table 7) supported by the camera device. This range can mirror the entire range (i.e. 01..F7), or can be a sub-range instead (e.g. 01..20) The range is expressed as a big-endian 16-bit integer, where the lower range value is stored in the hi-byte and the upper range value is stored in the lo-byte		✓
Ncx	The number of sensor elements (sels) in the horizontal (x) direction of the camera device's CCD		✓
Ncy	The number of sensor elements (sels) in the vertical (y) direction of the camera device's CCD		✓
Dx	The dimension of a sensor element (sel) in the horizontal (x) direction, measured in millimetres (mm)/sel		✓
Dy	The dimension of a sensor element (sel) in the vertical (y) direction, measured in millimetres (mm)/sel		✓

Table 5

In order to deliver future surveillance functionality within the Sentinel system, the control PC will need to model camera devices according to the classic 'pinhole camera' model. This then enables accurate camera device *calibration* to be performed. The properties *Ncx*, *Ncy, Dx*, and *Dy*, otherwise known as 'intrinsic camera constants', should be well known by the manufacturer. These values will enable the control PC to calculate a camera device's *intrinsic* (internal) and *extrinsic* (external) calibration.

Further details regarding camera calibration will be defined in a future revision of this specification.

Get Dynamics (23)

The format is:

0	1..2	3..6	7..10	11
STX	MsgID	NetID	Mask	ETX

- Byte 0 represents the start of message character (ASCII 0x02)
- Bytes 1..2 represent the *Get Dynamics* message ID. These bytes consist of the hexadecimal characters ASCII 0x32, ASCII 0x33 (i.e. '23')
- Bytes 3..6 represent the 16-bit Sentinel network address assigned to the camera device. These bytes consist of 4 hexadecimal ASCII characters
- Bytes 7..10 represent a 16-bit property value mask encoded as 4 hexadecimal ASCII characters. The mask comprises the following bits:

15	14..0
PresetRange	unused

  - Bit 15 indicates that the *PresetRange* property value is required
  - Bits 14..0 are unused
- Byte 11 represents the end of message character (ASCII 0x03)

An example is:

0	1..2	3..6	7..10	11
	23	0015	8000	

This message requests the *PresetRange* dynamics property value from a camera device that has a *NetID* of 0015

Dynamics (A2)

The format is:

0	1..2	3..6	7..10	11.. ..n-1	n
STX	MsgID	NetID	Mask	Properties	ETX

- Byte 0 represents the start of message character (ASCII 0x02)
- Bytes 1..2 represent the *Dynamics* message ID. These bytes consist of the hexadecimal characters ASCII 0x41, ASCII 0x32 (i.e. 'A2')
- Bytes 3..6 represent the 16-bit Sentinel network address assigned to the camera device. These bytes consist of 4 hexadecimal ASCII characters
- Bytes 7..10 represent a 16-bit property value mask encoded as 4 hexadecimal ASCII characters. This is copied from the *Get Dynamics* message being replied to
- Bytes 11..n-1 represent the properties values requested. These values are packed according to the order in which they appear in the mask
- Byte n represents the end of message character (ASCII 0x03)

An example is:

0	1..2	3..6	7..10	11..14	15
	A2	0015	8000	0120	

This message has been sent in response to a *Get Dynamics* message. The camera device has a *NetID* of 0015, and has returned a value of 0120 for its *PresetRange* property. This indicates that the camera device supports a Sentinel preset sub-range of 01..20 (i.e. 1..32)

Diagnostics

To facilitate service and fault management, it should be possible for the control PC to get and/or reset diagnostics 'properties' similar to those shown in Table 6 below:

Property	Description	Get	Reset
TotalStartStops	The total number of times that the PTZ camera device has moved, from a stationary position and back again, since manufacture. This is a 32-bit integer value	✓	
Total360°Rotations	The total number of 360° rotations performed by the PTZ camera device since manufacture. This is a 32-bit integer value	✓	
Flags	An array of bits that indicate any data communications errors, mechanical failures, etc, experienced by the camera device	✓	✓

Table 6

To get diagnostics properties, the control PC must send a *Get Diagnostics* message to a particular camera device.

When a camera device receives a *Get Diagnostics* message, it sends a *Diagnostics* message to the control PC containing diagnostic values of interest.

To reset diagnostics properties, the control PC must send a *Reset Diagnostics* message instead.

Get Diagnostics (31)

The format is:

0	1..2	3..6	7..8	9
STX	MsgID	NetID	Mask	ETX

- Byte 0 represents the start of message character (ASCII 0x02)
- Bytes 1..2 represent the *Get Diagnostics* message ID. These bytes consist of the hexadecimal characters ASCII 0x33, ASCII 0x31 (i.e. '31')
- Bytes 3..6 represent the 16-bit Sentinel network address assigned to the camera device. These bytes consist of 4 hexadecimal ASCII characters
- Bytes 7..8 represent an 8-bit property value mask encoded as 2 hexadecimal ASCII characters. The mask comprises the following bits:

7	6	5..0
TotalStartStops	Total360°Rotations	unused

  - Bit 7 indicates the *TotalStartStops* property value is required
  - Bit 6 indicates the *Total360°Rotations* property value is required
  - Bits 5..0 are unused
- Byte 9 represents the end of message character (ASCII 0x03)

An example is:

0	1..2	3..6	7..8	9
	31	00AB	40	

This message requests the *Total360°Rotations* diagnostics property value from a camera device that has a *NetID* of 00AB

Diagnostics (B1)

The format is:

0	1..2	3..6	7..8	9.. ..n-1	n
STX	MsgID	NetID	Mask	Properties	ETX

- Byte 0 represents the start of message character (ASCII 0x02)
- Bytes 1..2 represent the *Diagnostics* message ID. These bytes consist of the hexadecimal characters ASCII 0x42, ASCII 0x31 (i.e. 'B1')
- Bytes 3..6 represent the 16-bit Sentinel network address assigned to the camera device. These bytes consist of 4 hexadecimal ASCII characters

- Bytes 7..8 represent an 8-bit property value mask encoded as 2 hexadecimal ASCII characters. This is copied from the *Get Diagnostics* message being replied to

- Bytes 9..n-1 represent the properties values requested. These values are packed according to the order in which they appear in the mask

- Byte n represents the end of message character (ASCII 0x03)

An example is:

0	1..2	3..6	7..8	9.. ..16	17
☺	B1	0016	40	00003D86	♥

This message was sent by a camera device, which has a *NetID* of 0016. It indicates that the *Total360°Rotations* since manufacture is 000019FB (i.e. 15750 decimal)

Telemetry

The requirements for telemetry control are split into two distinct areas: *Default* and *Extended*.

Default

PTZ camera devices should be capable of supporting basic telemetry operations such as pan, tilt, zoom, (manual) focus, auto-pan on/off, and so on.

To control default telemetry operations, the control PC must send an appropriate message to a particular camera device. Default telemetry messages are categorized as follows:

- *Set Preset*. Set a preset position

- *Goto Preset*. Go to a pre-defined preset position, at a specified speed

- *Auto Pan*. Start/stop auto-panning (between left and right preset positions), at a specified speed

- *Pan Tilt Zoom Focus*. Start/stop pan, tilt, zoom, and (manual) focus at specified speeds

The following table (Table 7) defines the various speed and preset ranges for the aforementioned telemetry operations (expressed in hexadecimal):

Telemetry Operation	Speed Range	Preset Range
Set Preset	N/A	01..FF ▪ 01..F7 == user/application defined ▪ F8 == auto-pan left limit ▪ F9 == auto-pan right limit ▪ FA == auto-park ▪ FB..FF == reserved
Goto Preset	00..FF	01..FF

	▪ 01..FE == user/application defined ▪ FF == default	
Auto Pan	00..FF ▪ 01..FE == user/application defined ▪ FF == default	F8..F9
Pan Tilt Zoom Focus	00..FF 00..FF 00..FF N/A	N/A N/A N/A N/A

Table 7

Although 255 speeds are specified for expandability purposes, initially there will only be a need for a small number of speeds utilized. For example:

- 00 == stop
- 01 == slow
- 02 == medium
- 03 == fast
- 04 == very fast

Speed FF is reserved, and represents 'default' speed. This applies only to *Goto Preset* and *Auto Pan* messages, for camera devices that do not support variable speed preset selection and auto-pan operations.

Also note that telemetry operations such as auto-focus and auto-flip should be considered 'always on'. Therefore, no extra messages will be required to control these operations.

When a camera device receives a default telemetry message, it replies with a *Telemetry Result* message to indicate the success/failure of the requested operation. Contained within the message will be a result code that indicates either success or a reason for failure. A list of result codes is shown in Table 8 below:

Result Code	Description
00	Success: Operation performed without error
01..FE	Failure: Operation specific error codes. These will be defined in a future revision of this specification
FF	Failure: General error

Table 8

Set Preset (41)

The format is:

0	1..2	3..6	7..8	9
STX	MsgID	NetID	PresetNo	ETX

- Byte 0 represents the start of message character (ASCII 0x02)

- Bytes 1..2 represent the *Set Preset* message ID. These bytes consist of the hexadecimal characters ASCII 0x34, ASCII 0x31 (i.e. '41')

- Bytes 3..6 represent the 16-bit Sentinel network address assigned to the camera device. These bytes consist of 4 hexadecimal ASCII characters

- Bytes 7..8 represent the preset number, and consist of 2 hexadecimal ASCII characters

- Byte 9 represents the end of message character (ASCII 0x03)

An example is:

0	1..2	3..6	7..8	9
	41	00CA	0F	

This message requests that a camera device, with a *NetID* of 00CA, sets preset 0F according to its current position

---

Goto Preset (42)

The format is:

0	1..2	3..6	7..8	9..10	11
STX	MsgID	NetID	PresetNo	Speed	ETX

- Byte 0 represents the start of message character (ASCII 0x02)

- Bytes 1..2 represent the *Goto Preset* message ID. These bytes consist of the hexadecimal characters ASCII 0x34, ASCII 0x32 (i.e. '42')

- Bytes 3..6 represent the 16-bit Sentinel network address assigned to the camera device. These bytes consist of 4 hexadecimal ASCII characters

- Bytes 7..8 represent the preset number, and consist of 2 hexadecimal ASCII characters

- Bytes 9..10 represent the preset speed, and consist of 2 hexadecimal ASCII characters

  If these bytes are set to ASCII 0x46, ASCII 0x46 (i.e. 'FF'), a device-specific default speed is requested instead

- Byte 11 represents the end of message character (ASCII 0x03)

An example is:

0	1..2	3..6	7..8	9..10	11
	42	0038	05	04	

This message requests that a camera device, with a *NetID* of 0038, 'goes to' preset 05 at a speed of 04

Auto Pan (43)

The format is:

0	1..2	3..6	7..8	9
STX	MsgID	NetID	Speed	ETX

- Byte 0 represents the start of message character (ASCII 0x02)
- Bytes 1..2 represent the *Auto Pan* message ID. These bytes consist of the hexadecimal characters ASCII 0x34, ASCII 0x33 (i.e. '43')
- Bytes 3..6 represent the 16-bit Sentinel network address assigned to the camera device. These bytes consist of 4 hexadecimal ASCII characters
- Bytes 7..8 represent the auto-pan speed, and consist of 2 hexadecimal ASCII characters. If these bytes are set to ASCII 0x30, ASCII 0x30 (i.e. '00'), auto-pan will be stopped. Otherwise, auto-pan will be started at the requested speed

  If these bytes are set to ASCII 0x46, ASCII 0x46 (i.e. 'FF'), a device-specific default speed is requested instead
- Byte 9 represents the end of message character (ASCII 0x03)

An example is:

0	1..2	3..6	7..8	9
☻	43	0006	02	♥

This message requests that a camera device, with a *NetID* of 0006, starts auto-panning at a speed of 02

Pan Tilt Zoom Focus (44)

The format is:

0	1..2	3..6	7..8	9..10	11..12	13..14	15	16
STX	MsgID	NetID	Mask	Pan Speed	Tilt Speed	Zoom Speed	Focus On/Off	ETX

- Byte 0 represents the start of message character (ASCII 0x02)
- Bytes 1..2 represent the *Pan Tilt Zoom Focus* message ID. These bytes consist of the hexadecimal characters ASCII 0x34, ASCII 0x34 (i.e. '44')
- Bytes 3..6 represent the 16-bit Sentinel network address assigned to the camera device. These bytes consist of 4 hexadecimal ASCII characters
- Bytes 7..8 represent an 8-bit PTZF mask encoded as 2 hexadecimal ASCII characters. The mask comprises the following bits:

7	6	5	4	3	2	1	0
Pan Left	Pan Right	Tilt Up	Tilt Down	Zoom In	Zoom Out	Focus Near	Focus Far

  - Bit 7 indicates a pan left operation is requested
  - Bit 6 indicates a pan right operation is requested

- Bit 5 indicates a tilt up operation is requested
- Bit 4 indicates a tilt down operation is requested
- Bit 3 indicates a zoom in operation is requested
- Bit 2 indicates a zoom out operation is requested
- Bit 1 indicates a focus near operation is requested
- Bit 0 indicates a focus far operation is requested

Combinations of bits can be set in order to perform simultaneous pan, tilt, zoom, and focus operations

- Bytes 9..10 represent the pan speed, encoded as 2 hexadecimal ASCII characters. If these bytes are set to ASCII 0x30, ASCII 0x30 (i.e. '00'), the current pan operation will be stopped. Otherwise, the pan operation will be started at the requested speed
- Bytes 11..12 represent the tilt speed, encoded as 2 hexadecimal ASCII characters. If these bytes are set to ASCII 0x30, ASCII 0x30 (i.e. '00'), the current tilt operation will be stopped. Otherwise, the tilt operation will be started at the requested speed
- Bytes 13..14 represent the zoom speed, encoded as 2 hexadecimal ASCII characters. If these bytes are set to ASCII 0x30, ASCII 0x30 (i.e. '00'), the current zoom operation will be stopped. Otherwise, the zoom operation will be started at the requested speed
- Byte 15 represents focus on/off. If this byte is set to ASCII 0x30 (i.e. '0'), the current focus operation will be stopped. Otherwise, if set to ASCII 0x30 (i.e. '1'), the requested focus operation will be started
- Byte 16 represents the end of message character (ASCII 0x03)

Examples are:

0	1..2	3..6	7..8	9..10	11..12	13..14	15	16
	44	0014	84	04	00	01	0	ETX

This message requests that a camera device – that has a *NetID* of 0014 - pans left at a speed of 04, and zooms out at a speed of 01

0	1..2	3..6	7..8	9..10	11..12	13..14	15	16
	44	0102	6A	02	03	00	1	ETX

This message requests that a camera device – that has a *NetID* of 0102 - pans right at a speed of 02, tilts up at speed 03, stops zooming in, and focuses near

0	1..2	3..6	7..8	9..10	11..12	13..14	15	16
	44	0025	00	00	00	00	0	ETX

This message requests that a camera device – that has a *NetID* of 0025 – stops all PTZF operations immediately

Telemetry Result (C1)

The format is:

0	1..2	3..6	7..8	9
STX	MsgID	NetID	ResultCode	ETX

- Byte 0 represents the start of message character (ASCII 0x02)
- Bytes 1..2 represent the *Telemetry Result* message ID. These bytes consist of the hexadecimal characters ASCII 0x43, ASCII 0x31 (i.e. 'C1')
- Bytes 3..6 represent the 16-bit Sentinel network address assigned to the camera device. These bytes consist of 4 hexadecimal ASCII characters
- Bytes 7..8 represent the result code, encoded as 2 hexadecimal ASCII characters (see Table 6)
- Byte 9 represents the end of message character (ASCII 0x03)

An example is:

0	1..2	3..6	7..8	9
☻	C1	00B7	00	♥

This message has been returned by a camera device, with a *NetID* of 00B7, and indicates that the previous telemetry operation succeeded

---

Extended

As well as supporting basic (default) telemetry control, PTZ camera devices must also enable very accurate control of pan, tilt, zoom, and focus. This is purely for the purpose of delivering future surveillance functionality within the Sentinel system.

Therefore, it is proposed that the position (i.e. the pose) of a camera device be specified as 'motionstep' counts. Motionsteps can be thought of as logical units of movement and, as a benchmark, 1 motionstep count == 0.01^{o} (approximately). It is up to the camera device to map logical motionstep counts for pan, tilt, zoom, and focus, into their physical motorstep equivalents.

Suggested motionstep ranges are:

- *Pan*. This has a motionstep range of 0..35999 (representing continuous $0.00^{o}..359.99^{o}$ coverage)
- *Tilt*. This has a motionstep range of 0..8999 (representing $0.00^{o}..89.99^{o}$ coverage)
- *Zoom*. This has a motionstep range of 0..1023 (representing zoom in through to zoom out)
- *Focus*. This has a motionstep range of 0..255 (representing focus near through to focus far)

To enable accurate control of pan, tilt, zoom and focus, the following messages are proposed:

- *Get PTZF* Pose. Enables the control PC to retrieve the current pan, tilt, zoom, and focus pose for a particular camera device (expressed as motionstep counts)
- *PTZF Pose*. This message is sent by a camera device, to the control PC, in response to a *Get PTZF Pose* request message
- *Set PTZF Pose*. Enables the control PC to set the pan, tilt, zoom, and focus pose for a particular camera device, by specifying respective motionstep counts

Obviously, to achieve this fine-grained level of positioning functionality, camera devices must be capable of managing intensive message exchanges with the control PC.

Get PTZF Pose (51)

The format is:

0	1..2	3..6	7	8
STX	MsgID	NetID	Mask	ETX

- Byte 0 represents the start of message character (ASCII 0x02)
- Bytes 1..2 represent the *Get PTZF Pose* message ID. These bytes consist of the hexadecimal characters ASCII 0x35, ASCII 0x31 (i.e. '51')
- Bytes 3..6 represent the 16-bit Sentinel network address assigned to the camera device. These bytes consist of 4 hexadecimal ASCII characters
- Byte 7 represents a 4-bit motionstep count mask encoded as a single hexadecimal ASCII character. The mask comprises the following bits:

3	2	1	0
Pan	Tilt	Zoom	Focus

  - Bit 3 indicates that the *Pan* motionstep count is required
  - Bit 2 indicates that the *Tilt* motionstep count is required
  - Bit 1 indicates that the *Zoom* motionstep count is required
  - Bit 0 indicates that the *Focus* motionstep count is required

  Combinations of bits can be set in order to perform composite pan, tilt, zoom, and focus requests

- Byte 8 represents the end of message character (ASCII 0x03)

An example is:

0	1..2	3..6	7	8

☺	51	012A	F	♥

This message requests the *Pan*, *Tilt*, *Zoom*, and *Focus* motionstep counts, from the camera device that has a *NetID* of 012A

PTZF Pose (D1)

The format is:

0	1..2	3..6	7	8.. ..n-1	n
STX	MsgID	NetID	Mask	Motionstep Counts	ETX

- Byte 0 represents the start of message character (ASCII 0x02)
- Bytes 1..2 represent the *Get PTZF Pose* message ID. These bytes consist of the hexadecimal characters ASCII 0x44, ASCII 0x31 (i.e. 'D1')
- Bytes 3..6 represent the 16-bit Sentinel network address assigned to the camera device. These bytes consist of 4 hexadecimal ASCII characters
- Byte 7 represents a 4-bit motionstep count mask encoded as a single hexadecimal ASCII character. This is copied from the *Get PTZF Pose* message being replied to
- Bytes 8..n-1 represent 16-bit motionstep counts. These are encoded as hexadecimal ASCII characters, and are packed according to order in which they appear in the mask
- Byte n represents the end of message character (ASCII 0x03)

An example is:

0	1..2	3..6	7	8..11	12..15	16..19	20..23	24
☺	D1	0038	F	13D3	05F0	0024	001E	♥

This message was sent from a camera device, with a *NetID* of 0038, reporting its *Pan*, *Tilt*, *Zoom*, and *Focus* motionstep counts as 13D3 (50.75^{o}), 05F0 (15.2^{o}), 0024, and 001E respectively

---

Set PTZF Pose (52)

The format is:

0	1..2	3..6	7	8.. ..n-1	n
STX	MsgID	NetID	Mask	Motionstep Counts	ETX

- Byte 0 represents the start of message character (ASCII 0x02)
- Bytes 1..2 represent the *Get PTZF Pose* message ID. These bytes consist of the hexadecimal characters ASCII 0x35, ASCII 0x32 (i.e. '52')
- Bytes 3..6 represent the 16-bit Sentinel network address assigned to the camera device. These bytes consist of 4 hexadecimal ASCII characters

- Byte 7 represents a 4-bit motionstep count mask encoded as a single hexadecimal ASCII character

- Bytes 8..n-1 represent 16-bit motionstep counts. These are encoded as hexadecimal ASCII characters, and are packed according to order in which they appear in the mask

- Byte n represents the end of message character (ASCII 0x03)

An example is:

0	1..2	3..6	7	8..11	12..15	16
	52	0038	C	6207	03E8	

This message was sent to a camera device, with a *NetID* of 0038, and controls the *Pan* and *Tilt* pose by supplying motionstep counts of 6207 (250.95^{o}) and 03E8 (10.0^{o}) respectively.

Note: If the pose of a camera device is controlled through extended telemetry messages, auto-focus and auto-flip need to be intelligently disabled by the camera device. Subsequently, when telemetry is controlled via default telemetry messages, auto-focus and auto-flip must then be re-enabled again