ENGINEERING CHANGE ORDER

Number	03-	
Project Engineer	Stephen L. Robinson	Cross Ref. Doc. Type & Number: None
Change Requested By	Stephen L. Robinson	Page 1 of 1

Description of Change

Release of technical documentation for archive:

Sandia National Labs, TASS, May 26, 1999

Reason for Change None

Scope of Change

- ☐ Changes Form, Fit, or Function
- ☐ Other performance enhancement
- ☒ Internal

Documentation Affected

Product Model Number: None

Drawing Number	Old Rev	New Rev
None		

Type of Change

- ☐ New Product
- ☐ Error
- ☐ Design Improvement
- ☒ Additional Info
- ☐ Cost Reduction
- ☐ Conform to Present Practices

Material Disposition

- ☒ None
- ☐ Scrap
- ☐ Rework
  - ☐ Finished Goods
  - ☐ Work In Progress
  - ☐ Stock
- ☐ Running Change

Approvals	Engineering Signature		Date
	Materials Signature		Date
Cost Impact	0	New Comp. Cost	0
Obsol. Impact	0	New Comp. Lead Time	0

Manager's Initials in Appropriate Box

☐ EWS ☐ Hot ☒ Normal

Required Tasks (use attachments if necessary)	Initials	Date
Manufacturing None		
Production None		
Materials None		
Stock Room None		
Sales / Marketing None		
Repair None		
Quality Assurance None		
Other None		

1 VOLT ASSOCIATES, INC.

June 25, 1999

Pelco
Mr. Kevin Carpenter

Re: TASS Standards - Hurley Associates Esprit Project

Dear Kevin:

Per our several conversations regarding the Project for Hurley Associates of Mt. Airy, Maryland to develop a positioning system for the U.S. Air Force TASS Program, I am sending the enclosed document from Sandia National Labs which describes the TASS standard for Interface Control of Pan/tilts and additional devices.

As we have discussed, this opportunity may allow us to sell between 500 and 3000 units of a modified Esprit device over a 3 year period. We may have additional opportunities to sell other conventional products into this program as well. While our initial opportunity may be to provide products using our current P or D control protocol, this document outlines the Sandia/D.O.D. standard for RS-422 Bidirectional control. I am forwarding this document to you for distribution as noted to relevent Pelco Staff (Dave Smith, Steve Robinson, Tom Dodrill, Jim Arbuckle, John Ellenberger, Kim Schneider) who are involved with software, control protocol, Esprit, etc. for their review. I would like this standard taken into consideration with an eye towards inclusion/compatibility with our "Millenium" protocol design.

Outside of this immediate opportunity we are reviewing on behalf Tom Hurley for EER, Inc. and the TASS program, I see additional possibilities for development along these lines for other thermal/visual imaging platforms for the U.S. Border Patrol Project, and other D.O.D. business currently enjoyed by Cohu and Philips. Please ask everyone to review the TASS Protocol document and hold in file for now. More specific details about the Hurley requirement will follow shortly.

Best Regards,

Brian Dolan

Brian Dolan
1 Volt Associates, Inc.

Manufacturers Representatives Serving the Electronic Industry in the Mid Atlantic Region

8709 2nd Avenue
Silver Spring, MD 20910
Tel - 301-565-3930
Fax - 301-585-7734

8500 Gruenwald Lane
Charlotte, NC 28210
Tel - 704-552-8707
Fax - 704-552-8708

35 N. Lincoln Avenue
Newtown, PA 18940
Tel - 215-504-2406
Fax - 215-504-2407

6900 Galaxie Road
Richmond, VA 23228
Tel - 804-261-4141
Fax - 804-264-7477

APPLICATION		REVISIONS			
NEXT ASSY.	USED ON	REV.	DESCRIPTION	DATE	APPROVAL
	TASS	A	Original Edition	95-12-10	
	TASS	B	Add DSP Device	96-1-5	
	TASS	C	Incorporate Cohu Comments	96-1-15	
	TASS	D	Modify Electrical Std. Refs	96-1-23	
	TASS	E	General Restructure	96-3-25	
	TASS	F	Add Message Length, Source	96-5-6	
	TASS	G	Changed tables for group size	98-6-02	
	TASS	H	Added joystick commands, button commands, revised to use standard hexadecimal characters, added size options to the extended message	98-08-31	
	TASS	I	Redefined addressing, added additional preset positions, simplified various messages, changed method to calculate checksum	99-05-26	

DRAFT

ALL SHEETS ARE AT THE CURRENT REVISION LEVEL

ORIGINAL DATE OF DRAWING 95-12-10	Sandia National Laboratories Security Technology Department		
DRAFTSMAN	APPROVAL	Interface Control Document for Control of Pan-Tilt Mounts, Cameras and Other Devices	
CHECKED	APPROVAL		
APPROVAL	APPROVAL		
DESIGN ACTIVITY APPROVAL	SIZE A	ORGN. NO. 05838	DWG. NO. ICD-TASS-001
PROCURING ACTIVITY APPROVAL	SCALE: NONE		SHEET 1 OF 30

DRAFT

INTERFACE CONTROL DOCUMENT FOR CONTROL OF PAN-TILT MOUNTS, CAMERAS, AND OTHER DEVICES

Prepared by:
Sandia National Laboratories
Security Technology Department 5838
P. O. Box 5800
Albuquerque, NM 87185-0780

For information or comments, contact:

Daniel A. Pritchard
(505) 844-7444
dpritch@sandia.gov

Sandia National Laboratories	SIZE A	ORGN. No. 05838	DWG. No. ICD-TASS-001
SCALE NONE	REV. I		SHEET 2 OF 30

CONTENTS

FIGURES AND TABLES

1. PURPOSE AND SCOPE

This document defines a standard for the control of pan-tilt mounts, cameras, and other devices. The purpose of this standard is to facilitate the integration and interoperability of a variety of equipment that can be used for alarm assessment in a security system. To achieve that goal, this document defines the message packets and responses that are used for the command and control of the features of advanced imaging cameras and auxiliary equipment such as infrared (thermal) imaging video cameras, visible waveband closed-circuit television (CCTV) cameras, snapshot video cameras, pan-tilt mounts, digital signal processors, and others.

The main features of this standard are the definition of the command messages and responses between devices in a group of devices that are interconnected by some media (network), and the message header that specifies the addressing of the command messages to assure appropriate delivery. Other features of this standard include methods to perform error checking and retransmission attempts. Not all remotely-controllable devices must implement all commands.

The manufacturer of devices conforming to this protocol should provide documentation that defines what messages and features are implemented and what responses are provided for specific commands.

The scope of this standard is generally limited to the messages used by application programs operating at the application layer level in the International Standards Organization (ISO) Basic Reference Model for Open Systems Interconnect (ISO/OSI). The seven layers of this model for network communications is shown in Table 1. However, some features are included in this document that cross over into other layers, for instance, device addressing . Some functions may also be redundant (such as addressing), depending on the implementation by the application program (Ethernet vs. RS-485).

Definition of the interface between devices corresponding to the physical (PHY) layer in the ISO/OSI model is beyond the scope of this document.

Table 1. Seven-layer ISO/OSI network model.

Layer	Name	Description
7	Application Layer	Type of communication: E-mail, file transfer, client/server.
6	Presentation Layer	Encryption, data conversion, ASCII-to-binary, etc.
5	Session Layer	Starts, stops session. Maintains order.
4	Transport Layer	Ensures delivery of entire file or message.
3	Network Layer	Routes data to different LANs based on network address.
2	Media Access Control (MAC) or Data Link Layer	Transmits packets from node to node based on station address.
1	Physical (PHY) Layer	Electrical signals and cabling.

2. DEFINITIONS AND BLOCK DIAGRAMS

2.1. Definitions

This document defines a software interface protocol and the functions that are available between control units and remote devices as shown in Figure 1.

Applicable Commands Control commands and messages in this protocol only apply if the remote device has that particular capability. For example, if a camera has a fixed focus lens, then it is not necessary to implement commands to adjust the focus. It is suggested that the supplier or manufacturer of the remote device publish a list of implemented commands

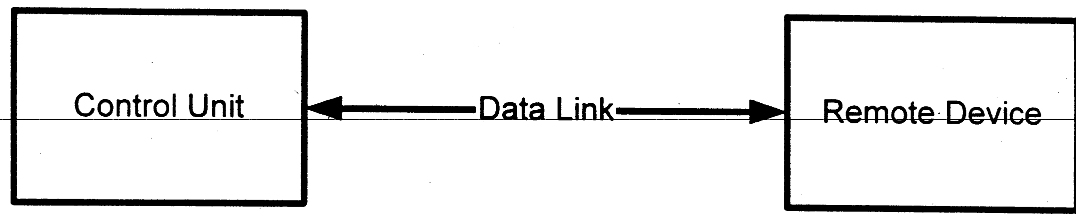

Figure 1. Basic block diagram.

Control Unit A device capable of synthesizing a string of ASCII characters, as documented in the protocol, and sending that message to a device to be controlled. The control unit can be a computer-based Display Control Unit (DCU), a Tactical Automated Security System (TASS) Desktop/Laptop Annunciator (DLA), a separate personal-computer (PC) -based controller, or other device capable of meeting the requirements.

Port An interface on a control unit used to send and receive messages.

Remote Device Equipment that consists of a receiver circuit and one or more electro-mechanical objects to be controlled. The receiver circuit typically translates the received message string of ASCII characters into the appropriate analog or digital signals to control the object. The object to be controlled can be a pan-tilt mount, a closed-circuit television (CCTV) camera, a thermal imaging camera, a combination of a pan-tilt mount and one or more cameras, a digital signal processor (DSP) board, or some auxiliary device located with the PTH or camera. For the remainder of this document, the term "device" includes both the receiver circuit, and all objects and/or functions to be controlled.

Address An identifier for either the control unit, the port, or the remote device. Devices must ignore messages not containing their address.

Data Link The data link is the circuitry, software, and media used to transfer messages from a control unit to a remote device. The data link can be implemented by direct wire connection, RF modem, internet protocol, optical fiber, or virtually any means, providing the link can transmit and receive ASCII characters. The data link is the bridge between the control unit and the device to be controlled, or between control units in a complex system.

Group Control Unit (GCU) A control unit that can be connected through the data link to multiple remote devices using either one port (communications channel) or multiple ports.

Master Control Unit (MCU) In a system with only one control unit, that control unit is called the Master Control Unit. In a system with multiple control units connected through a network, the Master Control Unit is the one designated to have highest priority or supervisory access.

Master/Slave This protocol is defined as a master-slave process. The master (control unit) initiates all communications to the slave (remote device). The slave responds only to the master. This protocol does not allow messages to be sent between slaves. Commands and responses can also be sent between GCUs (including the MCU). Commands and responses are generally sent only between a GCU and devices local to that GCU. This protocol also allows the "pass-through" of commands from one GCU to another GCU and it's local devices.

Hexadecimal Values between 0…15 are specified as 0xNN, where N can be 0…9 and A…F.

NIC Acronym for the Non-Implemented Command character (question mark, "?").

2.2. Configuration Block Diagrams

Figure 2 shows an example of an MCU or GCU with four ports and a single device attached to each port. Figure 3 shows an example of an MCU or GCU configured with many devices attached to a single port. Figure 4 shows an example of an MCU or GCU with multiple ports and devices.

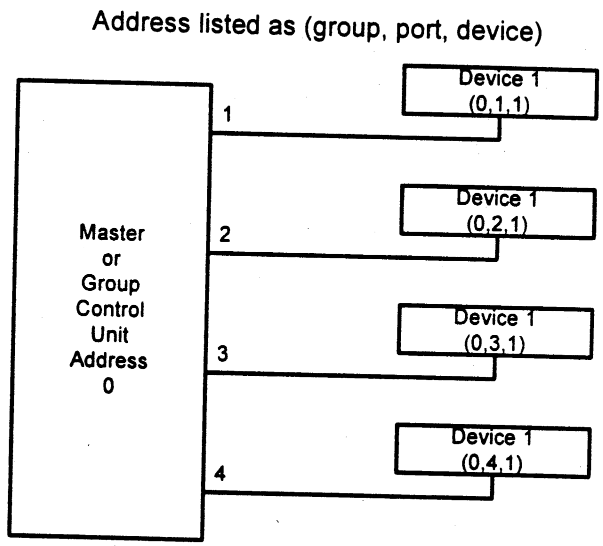

Figure 2. MCU/GCU with four ports and one device per port.

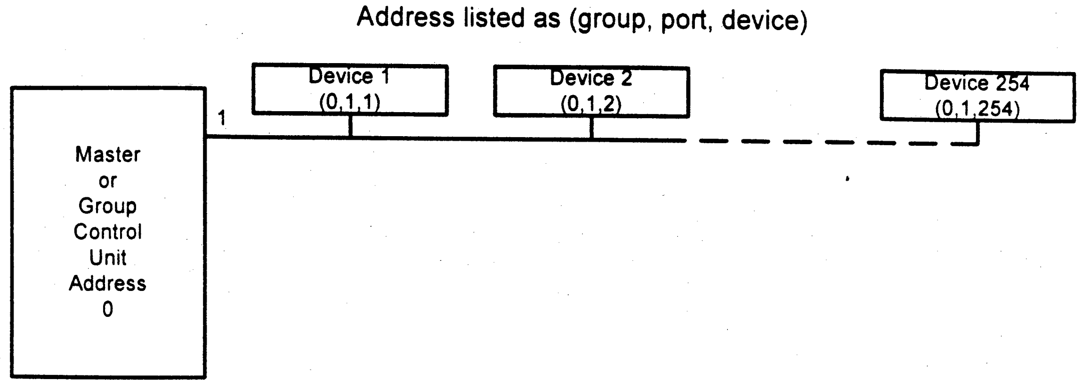

Figure 3. MCU/GCU with one port and many devices.

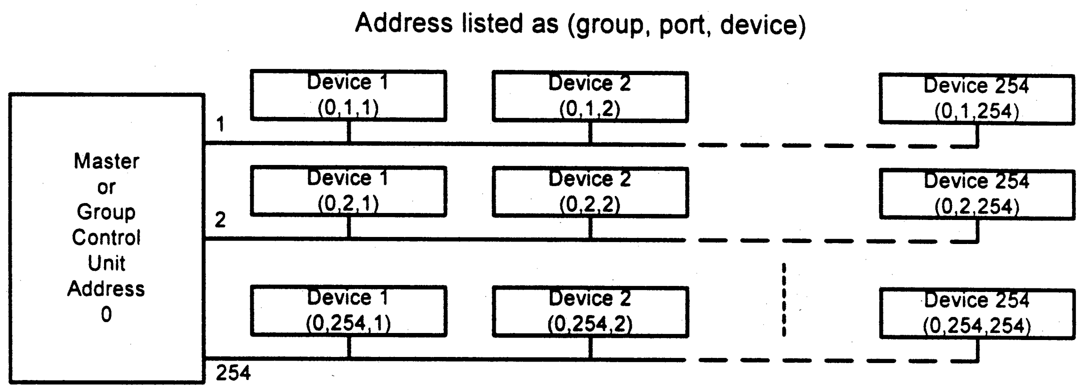

Figure 4. MCU/GCU with many ports and many devices per port.

Many GCUs and devices can be connected to form a network using the protocol defined in this document. Figure 5 shows a block diagram for a fully expanded system. In this configuration, up to 254 devices can be connected to a single communications port on a GCU. Each GCU can have up to 254 ports. Up to 254 group control units can be connected together in a network. One MCU is allowed in the system for sending messages to GCUs or directly to the devices on its own ports to be controlled.

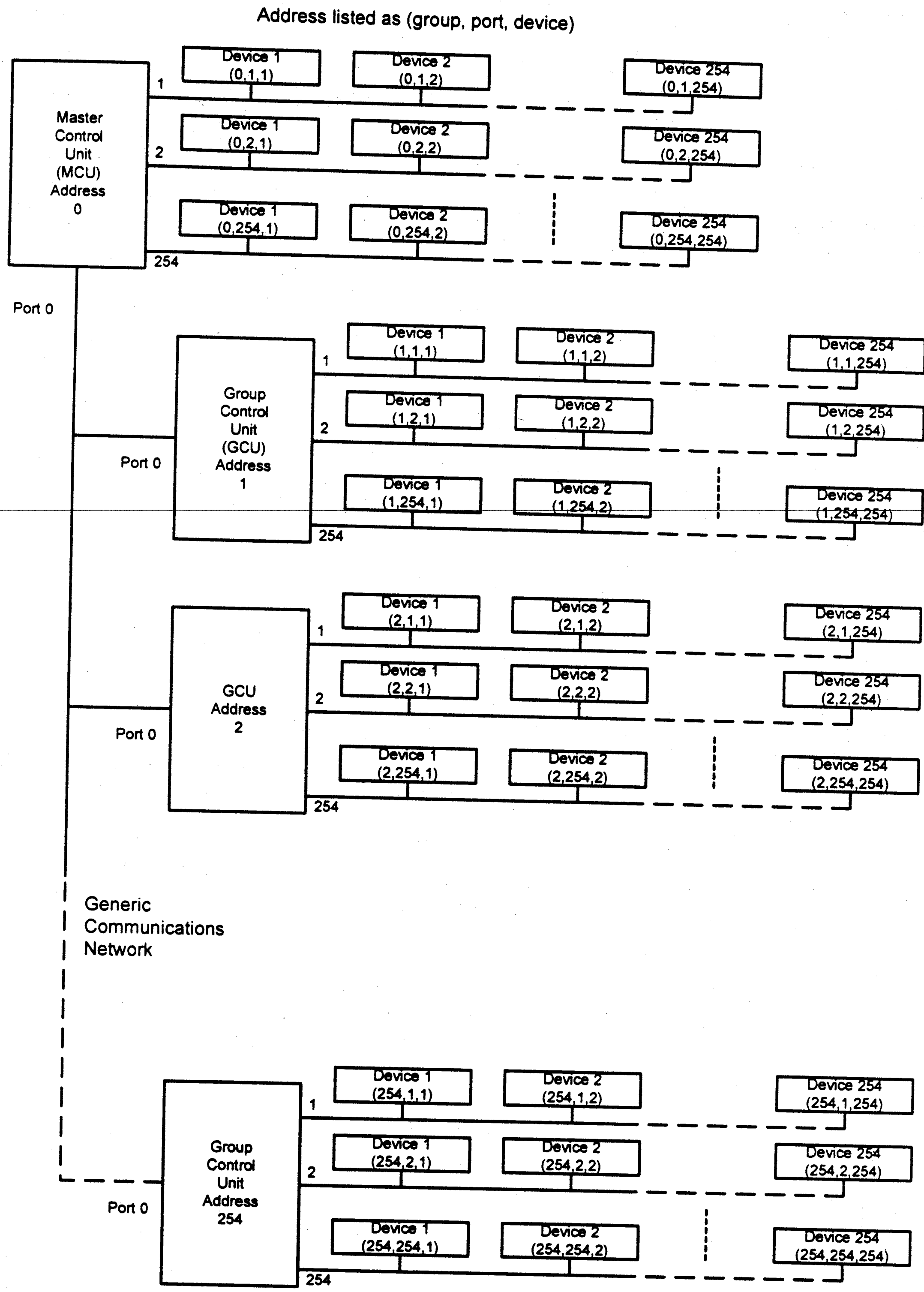

Figure 5. Block diagram of expanded system.

3. STANDARD MESSAGE FORMATS

3.1. Command Message From Control Unit To Remote Device

3.1.1. The standard, fixed command message format is six bytes of header plus one to 256 bytes of payload or command message plus a one byte checksum. See Table 2.

Table 2. Command Message Format

BYTE	DATA	DESCRIPTION
1	0xF8	Start character
2	Destination Group Address	Address of control unit destination address
3	Destination Port Address	Address of port within a control unit
4	Destination Device Address	Address of device on a port
5	Source GCU Address	Group Control Unit address of source of message
6	Length	Length in bytes of command data. Does not include bytes 1 through 6 or the checksum byte.
7 to 7 + length-1	Command data	See tables in following sections.
7 + length	Checksum	Computed on byte 2 through byte 7 + length

3.1.2. Header. The header consists of the first six bytes of the command message. All bytes in the header have hexadecimal values.

3.1.2.1. The first byte of a message always has the value 248 (hexadecimal 0xF8). This is the start character.

3.1.2.2. The second byte of a message indicates the destination Group Control Unit (GCU) address of the specific device for which the message is intended. The reserved addresses are shown in Table 3.

There is a limit of 254 GCUs per system with addresses from 1 to 254 (0x01 to 0xFE). GCU address zero (0) is reserved for the Master Control Unit (MCU). GCU address 255 is a wild-card address for messages that can be broadcast to all GCUs in a complete system.

3.1.2.3. The third byte of the address field is the destination communications port identifier within the GCU.

There is a limit of 254 ports per GCU with addresses from one to 254. Port address zero (0) is used to identify communications between GCUs and an MCU on a separate communications link. Port address 255 is a wild-card address for messages that can be broadcast to all ports in a GCU.

3.1.2.4. The fourth byte of the address field is the destination address of the device connected through the communications port to the GCU.

There is a limit of 254 devices per port, and device numbers can range from one to 254. Device address 0 is used for identifying the MCU/GCU as the destination in a result response message from a remote device. Device address 255 is a wild-card address for messages that can be broadcast to all ports in a GCU.

3.1.2.5. The fifth byte is the GCU address of the source of the message. This allows messaging from the MCU to a GCU, and between GCUs.

3.1.2.6. The sixth byte is be the length of the payload or command string, in bytes, not including the addressing bytes and checksum byte. This allows the receiver to accurately check message length. For example, if the payload or command message is two bytes, then the length byte will have a value of 0x02. If the length of the payload is 256 bytes, the length byte will have a value of zero.

3.1.2.7. All tables below that describe command formats only define the payload. The header and checksum, as defined in Table 2, must always be added.

Table 3. Specific Group, Port, and Device Addresses

ADDRESS	GCU ID BYTE	PORT ID BYTE	DEVICE ID BYTE
0	Reserved for MCU address	Port ID for MCU/GCU inter-communications	Identifies MCU/GCU as destination device
1	GCU 1	Port 1	Device number 1
2	GCU2	Port 2	Device number 2
…	…	…	…
254	Last GCU on System	Last port on GCU	Last device on port
255	Wild Card address for all GCUs	Wild Card address for all ports on a GCU	Wild Card address for all devices on a port

3.1.3. Payload. The payload of the message consists of at least one byte, up to a maximum of 256 bytes of data. All information is to be represented by hexadecimal ASCII characters with the exception of the binary data message and the extended message formats. Numbers above 9 are to be represented by capital letters (A, B,…F). A hexadecimal number like 0x3AF is represented as 0x33, 0x41, 0x46 (not as 0x33, 0x61, 0x66). The command data is sent as the ASCII codes for the characters shown in the tables. Commands generally consist of 2-byte pairs (2 ASCII characters), but may be more. The binary data (see 3.4.2 below) and the extended message commands have the provision for sending binary data as part of the payload.

3.1.4. Checksum. A checksum is transmitted with every command message. The checksum byte is the last byte of the command message. The checksum is calculated as the straight sum of bytes, ignoring overflow. The checksum is computed on byte 2 through byte 7 + length-1.

3.2. Simple Response Messages From Remote Devices

3.2.1. Every command message is to be responded to with a single ACK character (0x06, or 6_{10}) or a single NAK character (0x15 or 21_{10}) or a single question-mark character ("?", 0x3F, or 63_{10}) at the transmitted bit rate to acknowledge receiving the message and to indicate if the message was received properly. An example this simple handshake is shown in the first example in Figure 6. Note that these simple responses **do not** have a header and checksum.

3.2.2. Each command message is to be responded to by the slave device before another command can be sent.

3.2.3. If the calculated checksum is the same as the transmitted checksum, a single ACK character is sent back from the device to the GCU.

3.2.4. If the calculated checksum is not the same as the transmitted checksum, a NAK character is sent back from the device to the GCU.

3.2.5. If the message contains an invalid, unrecognized, or non-applicable command, the receiving device responds with the "Non-Implemented Command (NIC) character, which is a single question-mark character ("?"), to alert the control unit that the message sent has no meaning to the receiver.

3.2.6. If no ACK, NAK, or "?" is received within a "time-out" period, the message is re-transmitted. It is suggested that the message be transmitted a total of three times, though this may be programmable. After the limit of retries has been reached, a communications fault is to be signaled to the operator by the application software.

3.2.7. The time-out period will be short, typically equal to 3 character times plus 5 milliseconds. For a 9600 bps link, this is approximately 8 milliseconds. Special provisions may be made for slower data links and half duplex connections, such as RF modems, where the control unit transmitter and the device transmitter share the same carrier frequency.

3.2.8. The time-out period for simple responses may be programmable.

3.3. Result Response Messages From Remote Devices

3.3.1. If the command requires a result response, one response will be sent. This is shown in the example in Figure 6.

3.3.2. The result response is sent following the ACK. There may be a time delay between the ACK and the result response message, depending on the command.

3.3.3. The response message generally has ASCII characters to specify status and parameter settings back to the MCU/GCU. Exceptions include the case of extended messages or binary messages (see section 3.4).

3.3.4. The format of the result response message uses the same header, payload, and checksum as the command format in Table 2.

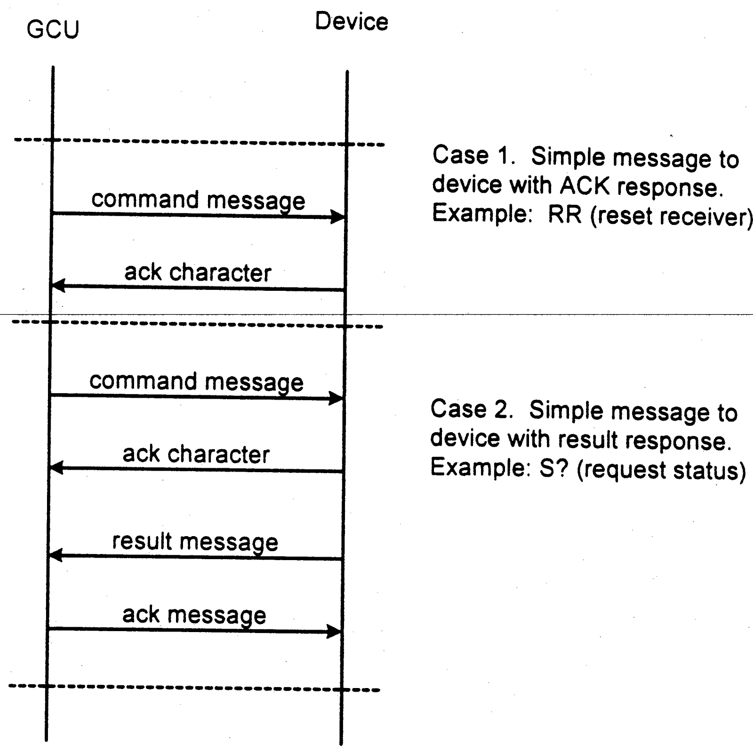

Figure 6. Message flow for simple and result responses to messages.

3.3.5. In the response message, the Destination Group Address and the Source Group Address are exchanged (assuming they are different, that is, that the source of the command was a GCU different from the one that the device is directly connected to). Because of the local communications between the GCU and the device, the GCU implicitly recognizes this message as one being transmitted back to the GCU, therefore the Destination Port Address and the Destination Device Address in the response message are unchanged from the original command message. The GCU can thus use the Destination Group information to determine if this message needs to be retransmitted to another GCU or keep the results local. Table 4 is provided as an example of this exchange.

Table 4. Example of Header Address Bytes for Command and Response Messages

HEADER BYTE	TRANSMIT COMMAND	RESULT RESPONSE
Destination Group	2	5
Destination Port	1	1
Destination Device	12	12
Source Group	5	2

3.3.6. In the diagram shown in Figure 7, a command issued from GCU-2 is passed through GCU-5 to Device 12 on Port 1. The header bytes are reversed according to Table 4 for the result response from the device back to GCU-2.

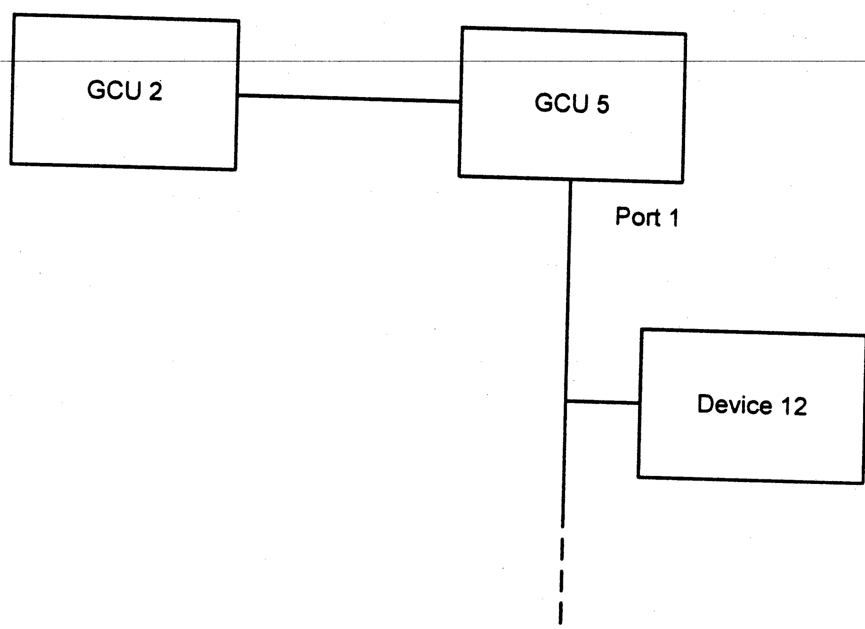

Figure 7. Block diagram of GCU to GCU communication (see Table 4).

3.3.7. If the GCU receives a proper response, the GCU acknowledges the transaction with a final ACK message. This ACK message includes the header and checksum. This ACK allows the device to clear the transmission link and wait for the next command, or send the next block of data in the case of a multi-block data transmission. The device **does not**, in turn, acknowledge the ACK message, but simply clears the communications link in preparation for receiving the next command.

3.3.8. If an improper result response from the device is received, or there are errors in the message, the GCU sends a final NAK message for this transaction. This NAK includes the header and checksum. The device **does not**, in turn, acknowledge the NAK message, but simply clears the communications link and waits for the next command. The device may, in fact, need to clear the communications link after a time-out period to allow new messages to be received. The GCU may retry the entire message transaction, but this retry and the number of retries is to be determined by the GCU application software. The GCU also has the responsibility to indicate a communications fault to the operator.

3.4. Large Messages To And From Remote Devices

3.4.1. There are two forms of messages to send or receive larger amounts of data to or from a remote device; binary data message and extended message.

3.4.2. If it is desired to send or receive additional data to or from the remote devices, the "X" command for sending unformatted, binary data is provided. This message format can be used for sending or receiving virtually any type of data such as embedded commands. The implementation is device specific and as such must be defined by the supplier of the equipment.

3.4.3. Table 5 defines the message format for a ***binary data message***.

3.4.4. If the remote devices support sending or receiving larger messages or data blocks, such as transferring images or other large amounts of data, the MCU/GCU manages the handshaking and communications.

Table 5. Binary Data Message Command Format

BYTE	DATA	DESCRIPTION
7	X	Binary data message
8 to 8+length (byte 6)	Data block	Up to 255 bytes of data

3.4.5. An ***extended data message*** ("EM" command) allows transferring large data blocks of various sizes.

3.4.6. A two-byte command can be included after the "EM"' command. This two-byte command is not defined in this document and is considered device specific.

3.4.7. Up to 0xFFFF+1 data blocks (65536) can be sent, and each data block can be up to 244 bytes.

3.4.8. A counter keeps track of which data block is currently being sent or received.

3.4.9. Each block transferred must be replied to by an ACK/NAK. The NAK causes a retransmit of the last block. Any ACK/NAK is a signal to the sender to start the transmission of a block. This gives the master the power to perform any pending poll activity in between blocks and also any higher priority message transfer. For this reason the ACK/NAK must include the standard header in the downstream case (the slave needs to see the address). Thus it would be possible for the master to send a command not requiring a reply to the data transmitting/receiving slave. This also requires the slave to have a quite long time-out in EM traffic.

3.4.10. It is possible to have a multi-thread EM communication between the master and several slaves. Since the ACK/NAK signals request of next block transfer it also acts as a 'XON/XOFF' signal. In this way the master can control which slave to be the next to transmit a block. This feature is useful if the slaves are slow in preparing new blocks.

3.4.11. Table 6 defines the format for a message to send large data blocks.

Table 6. Extended Message Command Format

BYTE	DATA	DESCRIPTION
6-7	EM	Extended Message Command
8-9	xx	Two byte command, device specific.
10-13	bbbb	Four hexadecimal characters indicating the total number of data blocks (0000-FFFF). Note that 0000 indicates 65536.
14-17	ssss	Four hexadecimal characters indicating the current block index (0000-FFFF). Note that 0000 indicates 65536.
18 to 18 + nn –1	Data	Maximum of 244 data bytes

4. PROTOCOL DEFINITION

4.1. Electrical Standards, Signal Formats, and Data Rates

4.1.1. Any media and interface standard may be used that allows bidirectional transmission of ASCII characters. These include industry standards such as Electronic Industries Association (EIA) standard EIA-232, EIA-422, or EIA-485 (RS-232, RS-422, RS-485) and other standards such as IEEE 802.3 (Ethernet), IEEE 802.11 (RF Networks).

4.1.2. If an RS-232/RS-422/RS-485 serial data interface is used, the receiving device is to support a data rate in bits per second (bps) of one or more of the following: 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200 bps. For some applications, the rate may be different.

4.1.3. Devices are not required to communicate at all data rates but must operate at the default data rate and format (start, stop & data bits).

4.1.4. The default data rate is 9600 bits per second (bps) with one start bit, eight data bits, one stop bit, and no parity. There are no provisions defined in this document to change start, stop, data, or parity bits.

4.1.5. The control unit may query the receivers to determine maximum data rate for all devices on the link.

4.1.6. The control unit may send commands to receivers to change data communication rates to match other devices on the link.

4.1.7. The data rate may (optionally) be automatically configured with an "auto-rate" character in the command message. The auto-rate detection process can be implemented as outlined in Section 4.7. below.

4.2. General Functions of the Protocol

The general functions included in the protocol consist of control functions for a pan-tilt mount (PTM), a CCTV camera, a thermal imaging camera, a digital signal processing (DSP) device, or any combination of devices. The protocol includes messages to control four auxiliary devices (relays) and return status to the control unit.

The capability to send and receive large data blocks is also included. Though the format and response for this type of message is usually device specific, a provision is included to have variable-length data block messages. Examples of uses for extended messages include transmission of digitized images and downloading new software to reprogram functionality.

4.3. Control Receiver and General Device Control Commands

4.3.1. The receiver circuitry may be integrated into the remote device to be controlled or may be a separate component. General receiver commands are defined in Table 7.

4.3.2. The remote device receiver executes a reset operation upon receipt of a reset command. The actions taken during a reset operation are device dependent. The reset command is intended to allow a remote restart of the device in an attempt to clear an error condition. This command is not intended to completely delete all configuration information such as device number. The device may not be available to accept additional commands immediately following a reset command.

4.3.3. All devices must respond to a message requesting an acknowledge. This is also known as an "awake?" query, or "ping."

4.3.4. A control unit must be able to be program a devices' ID number. This can be accomplished by connecting only one device to the port and issuing the three byte command "#nn" where nn is the device ID represented by two ASCII characters with a range of 01 to FE.

4.3.5. All devices must respond to an ID request message. The response message identifies the unit by including a block of information. The format for the payload of the result response is described in Table 9.

4.3.6. All devices must respond to a state-of-health or status request. The specific response from each remote device may be device dependent.

4.3.7. All devices must respond to a maximum data rate query. The response message specifies the highest data rate at which the device can communicate.

4.3.8. The control unit can specify to the devices the communications rate within the range of possible data rates the devices support.

4.3.9. The control unit can request a block of binary data from the device, or send a block of binary data to the device, if the remote device has the capability or function. This is a device-specific command.

4.3.10. If multiple video cameras or imaging devices are connected to the receiver, the control unit can select the video camera or imaging device for control, for video transmission, and display. The default video selected (V0) should be the primary imaging camera on the remote device. In many cases, this is a thermal imager.

4.3.11. If more than one CCTV camera and/or more than one FLIR is connected to one receiver, then the commands for those devices will have to be preceded by a command (V0-VF) to select the specific device to be the destination for the command.

4.3.12. The number of auxiliary relays providing switch-closure outputs defined by this document is four (0…3).

Table 7. General Device Control Commands

COMMAND	DESCRIPTION	RESPONSE	COMMENTS
#NN	Set device ID	ACK	NN = new device ID (two ASCII characters, range 01 to FE)
AW	Request awake status (ping)	ACK	One byte response
B?	Request maximum data rate	ACK + Result	Response is C0…CF. See below.
C0 C1 C2 C3 C4 C5 C6	Set communication data rate	ACK	0 = 1200 bits per second 1 = 2400 bits per second 2 = 4800 bits per second 3 = 9600 bits per second 4 = 19,200 bits per second 5 = 38,400 bits per second 6 = 57,600 bits per second
C7 C8…CF			7 = 115,200 bits per second Reserved for future data rates
D?	Request ID information	ACK + Result	See Table 9 for response format
EM	Extended message to (or from) device	ACK	See Section 3.4 for method of transferring data blocks.
L0	Open Relay 0	ACK	
L1	Open Relay 1	ACK	
L2	Open Relay 2	ACK	
L3	Open Relay 3	ACK	
L8	Close Relay 0	ACK	
L9	Close Relay 1	ACK	
LA	Close Relay 2	ACK	
LB	Close Relay 3	ACK	
L?	Auxiliary Relay Status Request	ACK + Result	See Table 8 for result format
RR	Reset Receiver	ACK	
SN	Self Test On	ACK	Initiate a self-test operation
SX	Self Test Off	ACK	Cancels self-test mode
S?	Request State of Health or status from device	ACK + Result	Multi-byte response. Result is generally dependent on the device.
V0…VF	Select Video	ACK	Select video output 0…F (1 of 16)
X	Binary data to device	ACK	See Table 5 for format

Table 8. Auxiliary Relay Status Results Message

BYTE	DATA	DESCRIPTION
7-8	L0…LF	Results indicate which relays are on (1) or off (0). For example: 0 = all relays off 1 = relay 0 on 2 = relay 1 on 3 = relays 0 and 1 on F = all relays on

4.4. Identification Response Message

4.4.1. Table 9 defines the message returned from a Request ID request.

Table 9. Device Identification Response Message

BYTE	DATA	DESCRIPTION
7-8	ID	ID Response
9-10	Protocol Revision Level	Two ASCII characters (leading space character)
11..30	ASCII character string	Device name, padded with 0x20 (space) if necessary
31-50	ASCII character string	Serial number of device, padded with 0x20 (space) if necessary

4.5. Maximum Data Rate Response Message

4.5.1. In Table 7, the C0…CF messages define the set of returned values from a Maximum Data Rate request.

4.6. Changing Device Addresses

4.6.1. The device ID may be programmable over the data link. The following process may be used: A single device can be connected to a control unit and the "#NN" message transmitted using wild-card addressing to set the device ID to a new address, where "#" is character 0x23 and NN is the new device ID number.

4.6.2. The group ID is not changeable over the data link.

4.6.3. Automatic device ID configuration with multiple devices on a port is possible with a process that is to be determined.

4.7. Automatic Data Rate Detection Process (Optional)

4.7.1. If the communication is implemented using a serial data link, a process for automatically sensing the data rate can be implemented by measuring the width (in time) of the first pulse of the $F8_H$ data byte. (First byte in command message)

4.7.2. For EIA-232 signals, the first bit (the start bit) is a logic zero or positive voltage.

4.7.3. The 0xF8 byte is transmitted least significant bit (LSB) first, providing three more logic-zero bits (three positive-voltage bits for EIA-232), for a total of four positive-voltage bits before the signal line goes to a negative voltage to transmit the remaining bits of the 0xF8 byte.

4.7.4. The bit rate can be calculated from the time duration of this initial pulse.

4.7.5. For EIA-422 or EIA-485, a similar procedure may be used.

4.7.6. In the event there is insufficient hardware to automatically sense and set the data rate, a software configuration process may be used to query each device to determine its maximum data rate, then set the link to operate at a rate compatible with all devices on that link.

4.7.7. Special provisions for inserting or connecting new devices onto a bus or a data link that has previously been reconfigured for a data rate higher than the default data rate are to be determined.

4.8. Automatic Revision Level Configuration

4.8.1. Automatic revision level configuration is a process used to request from a device the specific command subset that the device has implemented. This can be accomplished by using the extended message (EM) command with a subcommand. It is recommended that the subcommand "IC" be reserved for returning a comma-separated list of Implemented Commands from the device to the MCU/GCU. The "IC" command is a general (not device specific) command.

4.9. Polling of Devices For Status

4.9.1. A specific process for polling of devices is to be determined.

5. SPECIFIC DEVICE COMMANDS

5.1. Camera and Thermal Imager Control Commands

5.1.1. The control unit can turn the imaging device on and off.

5.1.2. The control unit can command the camera to focus near and far, adjust the focus speed, and enable/disable auto-focus.

5.1.3. The control unit can adjust the lens iris or aperture or set automatic modes.

5.1.4. The control unit can command the camera to zoom in, out and adjust the zoom speed.

5.1.5. The control unit can command the lens to a specific zoom and focus position, with up to 12-bit resolution. Each 4-bit nibble is represented by a hexadecimal ASCII character (for example, 0x4BF). See Table 11 for the position parameter format.

5.1.6. The control unit can set the digital zoom parameters.

5.1.7. The control unit can read zoom and focus information from the camera..

5.1.8. The control unit can change the color balance of the image from the camera.

5.1.9. The control unit can initiate a self-test operation. Results are device-dependent.

5.1.10. The control unit can select IR cut filter on/off, monochrome on/off, and backlight compensation on/off.

5.1.11. The control unit can select live or freeze-frame video.

5.1.12. The control unit can enable and disable automatic contrast and brightness.

5.1.13. The control unit can set the contrast and brightness with up to 12-bit resolution. See Table 12 for the format.

5.1.14. The control unit can set the field of view to wide or narrow.

5.1.15. The control unit can set the polarity to black or white hot.

5.1.16. The control unit can initiate a non-uniformity correction cycle.

5.1.17. The control unit can turn on, turn off, or change the reticle/graticule overlay on the display.

5.1.18. The control unit can read contrast, brightness, and other status from the device.

5.1.19. The camera and thermal imager control commands are shown in Table 10.

Table 10. Camera and Thermal Imager Control Commands

COMMAND	DESCRIPTION	RESPONSE	COMMENTS
BN	Backlight Compensation On	ACK	
BX	Backlight Compensation Off	ACK	
B*v2v1v0*	Set Brightness (Level)	ACK	See Table 12 for format
C*v2v1v0*	Set Contrast (Gain)	ACK	See Table 12 for format
CN	Power On	ACK	
CX	Power Off	ACK	
F0…F9	Focus Speed	ACK	F0 = slowest, F0 = fastest
FA	Auto-focus On	ACK	
FF	Focus Far	ACK	Send FS to stop focus
FN	Focus Near	ACK	Send FS to stop focus
FS	Focus Stop	ACK	
FX	Auto-focus Off	ACK	
HB	Set Polarity Black Hot	ACK	Selects black = hot video polarity
HW	Set Polarity White Hot	ACK	Selects white = hot video polarity
IA	Automatic Contrast and Brightness Mode	ACK	
IM	Manual Contrast and Brightness Mode	ACK	
IV	Live Image	ACK	
IZ	Freeze Image	ACK	
I?	Request Imager Status	ACK + Result	See Table 13 for status format
J0…JF	Iris/Aperture Setting	ACK	Manual, 16 steps, open to closed
KN	IR Cut Filter On	ACK	
KX	IR Cut Filter Off	ACK	
LN	Narrow Field-of-View	ACK	Selects narrow field-of-view
LW	Wide Field-of-View	ACK	Selects wide field-of-view
MN	Monochrome On	ACK	Puts camera in monochrome mode
MX	Monochrome Off	ACK	Puts camera in color mode
NC	Insert Shutter	ACK	Initiates non-uniformity correction
R0	Reticle/Graticule Off	ACK	
R1..R9	Reticle Select 1-9	ACK	
TM	Enable Test Mode	ACK + Result	Initiates a self-test operation
TX	Exit Test Mode	ACK	
U0…U9	Auto Exposure Mode	ACK	Sets auto exposure mode, 0 = auto
V*z2z1z0f2f1f0*	Lens Go-To Command	ACK	Sets lens zoom and focus position See Table 11 for format
V?	Lens Position Request	ACK + Result	See Table 11 for result format
W0…W9	White Balance	ACK	

COMMAND	DESCRIPTION	RESPONSE	COMMENTS
Z0…Z9	Zoom Speed	ACK	Z0 = slowest, Z9 = fastest
ZA…ZF	Digital Zoom	ACK	Additional digital zoom settings
ZI	Zoom In	ACK	Send ZS to stop zooming in
ZO	Zoom Out	ACK	Send ZS to stop zooming out
ZS	Zoom Stop	ACK	

Table 11. Lens Position Set Command and Result Message Format

BYTE	DATA	DESCRIPTION
7	V	V = go-to command
8	z2 (MS nibble)	Zoom position, bits 8-11 + 0x30 for 0-9 and + 0x41 for A-F
9	z1	Bits 4-7 + 0x30 for 0-9 and + 0x41 for A-F
10	z0 (LS nibble)	Bits 0-3 + 0x30 for 0-9 and + 0x41 for A-F
11	f2 (MS nibble)	Focus position, bits 8-11 + 0x30 for 0-9 and + 0x41 for A-F
12	f1	Bits 4-7 + 0x30 for 0-9 and + 0x41 for A-F
13	f0 (LS nibble)	Bits 0-3 + 0x30 for 0-9 and + 0x41 for A-F

Table 12. Thermal Imager Contrast and Brightness Message Format

BYTE	DATA	DESCRIPTION
7	B or C	Brightness or Contrast
8	v2 (MS nibble)	Brightness or contrast value bits 8-11 + 0x30 for 0-9 and + 0x41 for A-F
9	v1	bits 4-7 + 0x30 for 0-9 and + 0x41 for A-F
10	v0 (LS nibble)	bits 0-3 + 0x30 for 0-9 and + 0x41 for A-F

5.2. Imager Settings Response Message

5.2.1. Table 13 defines the results message format from a request for imager status. The results return settings of gain (contrast), brightness (level), and other parameters.

Table 13. Imager Status Response Message

BYTE	DATA	DESCRIPTION
7	S	S = Status returned
8	c2 (MS nibble)	Contrast value Bits 8-11 + 0x30 for 0-9 and + 0x41 for A-F
9	c1	Bits 4-7 + 0x30 for 0-9 and + 0x41 for A-F
10	c0 (LS nibble)	Bits 0-3 + 0x30 for 0-9 and + 0x41 for A-F
11	b2	Brightness value Bits 8-11 + 0x30 for 0-9 and + 0x41 for A-F
12	b1	Bits 4-7 + 0x30 for 0-9 and + 0x41 for A-F
13	b0	Bits 0-3 + 0x30 for 0-9 and + 0x41 for A-F
14	0x30-0x3F	LS nibble is 4 bits of status (0 or 1) Bit 0 = Field of View Wide (0) or Narrow (1) Bit 1 = Polarity White (0) or Black Hot (1) Bit 2 = Auto Contr/Bright Off (0) or On (1) Bit 3 = Test Mode Inactive (0) or Active (1)

5.3. Pan/Tilt Mount Commands

5.3.1. The control unit can specify the mount to pan left, right, or stop paning. The speed is adjustable.

5.3.2. The control unit can specify the pan/tilt mount to tilt down, up, or stop tilting. The speed is adjustable.

5.3.3. The control unit can specify the pan/tilt mount to program a preset position.

5.3.4. The control unit can specify the pan/tilt mount to move to a preset position.

5.3.5. The control unit can request status relating to moving to a preset position.

5.3.6. The number of preset positions defined in this protocol is 256 with a range from 00…FF. No minimum number of presets is mandatory, but any device that has programmable presets should implement the presets in order, beginning from 00. If an access is made to an unimplemented position, the reply shall be a NIC. The implemented positions should have "home" as a default value.

5.3.7. The control unit can specify the pan-tilt mount to move to a specific azimuth and elevation position determined by two 12-bit values, each represented by three hexadecimal ASCII characters. The mapping of the 12-bit position values to absolute pan or tilt position angles is device dependent. The manufacturer should specify the mapping of the values to position angles.

5.3.8. The control unit can request the current position from the pan-tilt mount. The position data is formatted as a 12-bit value represented by three hexadecimal ASCII characters for both azimuth and elevation.

5.3.9. Table 15 shows the format for the command to direct the pan-tilt mount to go to a position and the result of a position request (P?) command.

5.3.10. The control unit can specify the pan speed for manual moves. For some applications, this may be combined with the tilt speed command.

5.3.11. The control unit can specify the tilt speed for manual moves.

5.3.12. The control unit can specify the speed for auto-moves.

5.3.13. The speed range for manual and automatic moves is 0...15, where 0 may be the slowest speed or a time-dependent proportional rate, and 15 is the maximum speed for that type of move (manual or automatic).

5.3.14. The control unit can command the pan/tilt mount to begin an automatic scanning sequence. The preset positions to be used as the end points of the auto-scan motion are PA and PB.

5.3.15. PA is the left preset and PB is the right preset with reference to the camera's point of view. The auto scan then moves between these left and right positions.

5.3.16. The control unit can initiate a de-icing cycle, program the deicing interval, and turn on and off the automatic de-icing process.

5.3.17. The pan/tilt command format is shown in Table 14.

Table 14. Pan/Tilt Mount Commands

COMMAND	DESCRIPTION	RESPONSE	COMMENTS
A0…AF	Set Auto-Move Speed	ACK	A0 = slowest, AF = maximum
AS	Begin Auto-Scan Mode	ACK	Auto-scans between positions PA and PB. Use PS or TS to stop.
E0…EF	Set Speed for Tilt Manual Moves	ACK	E0 = slow/proportional, EF = maximum
GI	Perform De-icing cycle	ACK	
GN	Auto De-Icing On	ACK	
GX	Auto De-icing Off	ACK	
G0…G9	Set De-icing Interval	ACK	G0 = 0.5 minute G1 = 1 minute G2 = 2 minutes G3 = 4 minutes G4 = 8 minutes G5 = 15 mins. G6 = 30 mins. G7 = 1 hour G8 = 2 hours G9 = 4 hours
HO	Go to the "Home" Position	ACK	Home position is azimuth = 0 and elevation = zero degrees (level)
M?	Move Status Request	ACK + Result	See Table 16 for result format
NO	North Offset	ACK	Set current azimuth angle = north
P?	Current Position Query	ACK + Result	See Table 15 for format
Pa2a1a0e2e1e0	Go to Pan-Tilt Position	ACK	Go to specific pan and tilt position. See Table 15 for format
PA	Store Auto-Scan Position A	ACK	
PB	Store Auto-Scan Position B	ACK	
PK	Park	ACK	Go to "park" or storage position
PL	Pan Left	ACK	At constant speed. Send PS to stop.
PR	Pan Right	ACK	At constant speed. Send PS to stop
PS	Pan Stop	ACK	
Pnn	Go To Preset nn	ACK	nn = 00…FF (256 presets) Note: Three byte command
Snn	Store Current Position as Preset nn	ACK	nn = 00…FF (256 presets) Note: Three byte command
S0…SF	Set Speed for Pan (and Optional Tilt) Manual Moves	ACK	S0 = slow/proportional, SF = maximum
TD	Tilt Down	ACK	At constant speed. Send TS to stop.
TS	Tilt Stop	ACK	
TU	Tilt Up	ACK	At constant speed. Send TS to stop.

5.4. Pan-Tilt Mount Go-To Position Command and Response Message

5.4.1. Table 15 shows the format for the command to direct the pan-tilt mount to go to a position and the result of a position (P?) request message.

Table 15. Pan - Tilt Mount Go To Command and Response Message Format

BYTE	DATA	DESCRIPTION
7	P	P = position command or status request
8	a2 (MS nibble)	Azimuth (pan) position, bits 8-11 + 0x30 for 0-9 and + 0x41 for A-F
9	a1	bits 4-7 + 0x30 for 0-9 and + 0x41 for A-F
10	a0 (LS nibble)	bits 0-3 + 0x30 for 0-9 and + 0x41 for A-F
11	e2 (MS nibble)	Elevation (tilt) position, bits 8-11 + 0x30 for 0-9 and + 0x41 for A-F
12	e1	bits 4-7 + 0x30 for 0-9 and + 0x41 for A-F
13	e0 (LS nibble)	bits 0-3 + 0x30 for 0-9 and + 0x41 for A-F

5.5. Move to Position Status Response Message

5.5.1. Table 16 contains the response format for the move status (M?) request command.

Table 16. Move-to-Position Status Response Message Format2

BYTE	DATA	DESCRIPTION
7	M	Position Status
8	A S P	A = Active, unit is moving to position S = Stopped P = Park position

5.6. Auxiliary Functions and Device Commands

5.6.1. These commands describe functions that may be available in a separate device or may be integrated with a camera. For this document, these functions are generally referred to as digital signal processor (DSP) functions and commands.

5.6.2. The control unit can turn the DSP power on and off.

5.6.3. The control unit can put the DSP into a low power consumption or sleep state.

5.6.4. The control unit can set the compression level for video compression in the DSP as one of 10 levels (0…9). The mode of compression is device specific.

5.6.5. The control unit can download new information into the DSP such as software. This can be implemented using the EM command as defined in Section 3.4.

5.6.6. The control unit can command the DSP to digitize, compress using the current compression level and send back an image. See Figure 8 for an explanation of the image transmission process.

5.6.7. The control unit can command the DSP to digitize a new image, compute the changes, compress using the current compression level, and send back the changes.

5.6.8. The control unit can initiate a self test mode, if applicable. The actions during self test mode are considered device dependent.

5.6.9. The DSP command format is shown in Table 17.

Table 17. Digital Signal Processor Commands

COMMAND	DESCRIPTION	RESPONSE	COMMENTS
DC	Digitize New Image and Send Changes	ACK + Result	See Section 3.4 for procedure. Uses the last compression factor.
DI	Digitize Image	ACK	Use as reference image for DC
DL	Low Power Sleep	ACK	
DN	DSP Power On	ACK	
DS	Digitize & Send Image	ACK + Result	See Section 3.4 for procedure. Use as reference image for DC. Uses the last compression factor.
DT	Perform Self Test	ACK	Other responses TBD
DX	DSP Power Off	ACK	
D0…D9	Set Compression Factor	ACK	D0 = low compression (high quality) D9 = high compression (low quality)

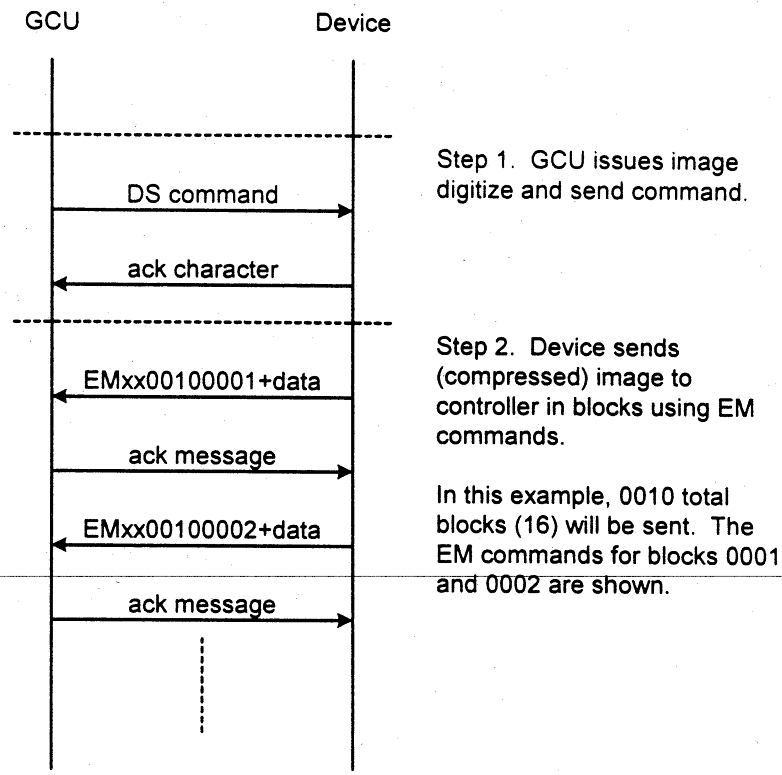

Figure 8. Image transmission process description.

5.7. Joystick Specific Commands

5.7.1. The control unit can command the pan/tilt mount to move up, down, left, or right based on the position of a joystick control lever.

5.7.2. The speed control of the movement is controllable over a range of 0-99 discrete speeds. The value of zero (0) indicates stop. The value of 99 indicates maximum speed. The maximum speed is not defined and may be device dependent.

5.7.3. Table 18 defines the message format for a joystick control message.

Table 18. Joystick Control Command Format

BYTE	DATA	DESCRIPTION
7	J	Joystick Control Command
8	L or R	L = move left, R = move right
9	ps1	Pan speed, most significant digit (0-9, 0x30 – 0x39)
10	ps0	Pan speed, least significant digit (0-9, 0x30 - 0x39)
		NOTE: ps1/ps0 = 00 indicates pan stop
11	U or D	U = move up, D = move down
12	ts1	Tilt speed, most significant digit (0-9, 0x30 - 0x39)
13	ts0	Tilt speed, least significant digit (0-9, 0x30 - 0x39)
		NOTE: ts1/ts0 = 00 indicates tilt stop

5.8. Button/Keypad Specific Commands

5.8.1. The control unit can simulate the pressing/releasing of buttons on a keypad unique to the device being controlled.

5.8.2. The button/keypad specific command provides 100 unique key codes (0..99) for general purpose keypad input and device control.

5.8.3. The control unit can indicate press or release of the key.

5.8.4. Table 19 defines the message format for a button or keypad simulation control message.

Table 19. Button/Keypad Control Message Format

BYTE	DATA	DESCRIPTION
7	B	Button Command
8	kc1	Key code most significant digit (0..9, 0x30 - 0x39)
9	kc0	Key code least significant digit (0..9, 0x30 - 0x39)
10	P or R	'P' = press, 'R' = release

6. COMMUNICATIONS ERROR HANDLING

6.1. Controller Responsibility

6.1.1. The controller is responsible for determining when a communications fault has occurred. See Section 3. above.