SPECIAL SPECIFICATION

6390

Software System Interface

1. Description. Integrate the new, contractor installed Fiber Optic Dynamic Message Sign System(s), Local Control Unit(s), Lane Control System(s), CCTV Field Equipment, and Radar Vehicle Sensing Devices with the existing Mainframe Computer and software located in the TransGuide Operations Control Center.

Do not install any hardware interface between the field controllers and the TransGuide mainframe.

No solutions other than specified within these specifications are acceptable.

Provide and install materials and equipment as detailed in these Special Specifications and as directed by the Engineer.

Provide controller software that interfaces the data stream and protocol as described below.

2. Fiber Optic Dynamic Message Sign System (DMS). The following describes the software protocol used to communicate between the Texas Department of Transportation (TxDOT) TransGuide Operations Control Center Mainframe (TOCCM) and the Fiber Optic Dynamic Message Sign System (DMS) controller. The protocol is based on sequences of ASCII characters transmitted via a serial data connection between the TOCCM and the DMS Controller.

A. General Command Structure. Data to be exchanged between the TOCCM and the DMS controller is framed into blocks. Each block is comprised of a header, a data body, and end of block information. The header always starts with "SOH" (Start of Header) followed by the address of the controller. The end of the header is a "STX" (Start of Text) to indicate data follows. The data is placed after this STX character, and before the ETX character, which marks the end of the data body. Following the ETX character is a Block Check Character (BBC). Data transmitted to the controller is framed between two "NUL" characters used for synchronization. Data transmitted from the controller is framed between a leading "NUL" character and a trailing "SUB" character.

The following diagram is what a single transmission block looks like:

NUL --- SOH --- ADDR --- STX --- DATA --- ETX ---BCC ---NUL

Where:

NUL--Synchronization character (Hex 00)

SOH--The ASCII Start-of-Header character (Hex 01)

ADDR--Address of the controller determined as:

A controller address is comprised of two parts. The first part of the address is the physical address of the controller. It is a value between 0 and 255.

The second part of the address is the logical address of the controller. It is usually 01, unless the controller is in configuration mode, then the address is 00.

The controller address is sent over the line as 5 ASCII characters.

STX--The ASCII Start-of-Text character (Hex 02), the STX marks the end of the header and the start of the data

DATA--Data for to be exchanged, its length, format, and contents are dependent on the command type (details provided in the command detail section).

ETX--The ASCII End-of-Text character (Hex 03)

BCC--Block check character (to ensure data integrity). The BCC is computed as the arithmetic sum of all the block characters, up to and including the ETX character. When transmitting using a seven-bit data word, only the seven least significant bits are transmitted.

For the following command:

NUL-SOH-00101-STX-E-ETX-BCC- NUL

The BCC would be:

NUL = 00H
SOH = 01H
0 = 30H
0 = 30H
1 = 31H
0 = 30H
1 = 31H
STX = 02H
“E” = 45H
ETX = 03H

=====
13DH

BCC=(13DH).AND.(7FH)=3DH

B. Communication Termination. The TOCCM terminates communication with the DMS controller by using an "EOT"(Hex04) sequence. This sequence signals the controller to be ready to communicate with the computer. The sequence does not include a BCC and comprises only of an "EOT" character framed between two "NUL" characters. An example synchronization character string would be:

NUL-EOT-NUL

C. DMS Controller Modes. The DMS controller does not synchronously send data to the TOCCM, rather, the DMS controller communicates data in one of two fashions:

- A "SELECTING" mode in which the TOCCM requests the DMS controller to receive data. The following command sequence is sent to place the DMS controller in SELECTING mode:

  NUL-SOH-ADDR-SEL-NUL

- A "POLLING" mode in which the TOCCM requests the DMS controller to send data to the TOCCM. The following command sequence is sent to place the DMS controller in POLLING mode:

  NUL-SOH-ADDR-POLL-NUL

Note that there are no BCC characters attached to the above sequences.

D. Command Acknowledgement. Once a sequence has been sent, either from the TOCCM or the DMS controller, the listener always acknowledges the reception of data. If the reception was correctly received, an "ACK" (Hex 06) is sent over the line. If the reception was garbled or lost, a "NAK" (Hex 15) is sent over the line.

The appropriate character is framed between two synchronization characters as follows:

NUL-ACK-NUL to acknowledge command successfully received
NUL-NAK-NUL to acknowledge command not successfully received

E. Command Summary. In general each command starts with a capital letter (its command code), followed by various parameters to form a precise syntax. All commands are embedded between control characters according to the communication protocol discussed in the previous section.

The following commands will be sent from the TOCCM to the DMS controller to control the DMS:

*Display Command
*Status Retrieval
*Lamp Status
*Day/Night Changeover
*Abort Sign

*Clock/Calendar Read Command
*Clock/Calendar Set Command
*Overbrightness Setting
*Echo
*Simulation Control

The details of each of these commands will be further discussed in the following sections.

F. Command Details. The following sections provide the details for each command that is sent between the TOCCM and the DMS controller.

1. DMS Commands Sent From the TOCCM to DMS Controller. The following commands are sent from the TOCCM to the DMS Controller via the communications line.

The structure of the Display Command is as follows:

Offset	Format	Description
1	B	Command function
2	X	subsign number
3	X	X=0: Deferred display 1: Immediate display
4	X	X=0: Steady message 1: Two alternated messages
5	X	X=0: Text 1 steady 1: Text 2 flashing
6	X	Text 1 flashing time ON 3 <= X <= F (.3 sec <=1.5 sec.)
7	X	Text 1 flashing time OFF 3 <= X <= F (.3 sec <= X <= 1.5 sec)
8	X	X=0: Text 2 steady 1: Text 2 flashing
9	X	Text 2 flashing time ON 3 <= X <= F (.3 sec <= X <= 1.5 sec.)
10	X	Text 2 flashing time OFF 3 <= X <= F (.3 sec <= X <= 1.5 sec.)
11	XX	Text 1 time ON if alternated 3 <= XX <= FF (.3 < = XX <= 25.5 sec.)
13	XX	Blank time between Text 1 and Text 2 0 <= XX <= FF (0 <= XX <= 25.5 sec.)

Offset	Format	Description
15	XX	Text 2 time On if alternated 3 <= XX <= FF (.3 <= XX <= 25.5 sec.)
17	XXXX	Display time (overall) 1 <= XXXX <= FFFF FFFF = 65534 min. FFFF = infinite
21	X	Brightness of Text 1 X = 0: Normal X = 1: Overbright
22	X	Brightness of Text 2 X = 0: Normal X=1: Overbright
23	X	Text follows or Controller text X = 0: Text follows X = 0: Programmed text
24	XX	Number of characters in Text 1 (hexadecimal ASCII coded format)
26	XX	Number of characters in Text 2 (hexadecimal ASCII coded format)
28	X.....X or XX	N characters describing Text 1 or programmed text number (see 23)
xx	X.....X or XX	N characters describing Text 2 or programmed text number (see 23)

The structure of the reply sent by the DMS controller to the TOCCM is:

Error Free Case:

Offset	Format	Description
1	sent by the DMS	to the TOCCM Function correctly processed
2	0	No error occurred

Error Occurred Case:

Offset	Format	Description
1	b	Error occurred

Offset	Format	Description
2	X	X = 1 : Unknown function code X = 2 : EEPROM parameter code X = 3 : Reset indicator X = 4 : Syntax error in command X = 5 : Undefined subsign X = 6 : Unknown preprogrammed text X = 7 : Length error in command format X = 8 : Text with non ASCII characters X = E : Time not initialized X = F : CAN board no enabled or fault sensor

2. Status Retrieval. The status retrieval command provides the status of a DMS or it's associated subsign. The structure of the command status retrieval command is:

Offset	Format	Description
1	C	Command function
2	X	Subsign number 0 <= X <= 7

Note: Subsign 0 indicates the full sign.

Offset	Format	Description
1	C	Function correctly processed
2	XXXX	Remaining display time in minutes (e) 0000: Sign blank
6	X	X = 0 : Sign Off X = 1 : Sign loaded X = 2 : Sign loaded in deferred mode X = 3 : Sign lit X = 4 : Sign busy (=lit)
7	X	X = 0 : Normal Operation X = 1 : Loop-back mode X = 2 : Back-up operation X = 3 : Lamp(s) OUT & OFF X = 4 : Lamp(s) OUT & ON X = 5 : No 48 volts X = 6 : Sign Aborted X = 7 : Bad shutter power supply X = 8 : Simulation mode active
8	X	Display from : X = 0 : Central computer X = 1 : Maintenance Terminal X = 2 : Local Control Panel X = 3 : Remote Control Panel
9	X	Day / Night sensor status X = 0 : Normal mode X = 1 : Day mode

Offset	Format	Description
10	X	Overbright sensor status X = 0 : Normal mode X = 1 : Overbright mode
11	X	Day / Night Command X = 0 : Night command X = 1 : Day command
12	X	Overbrightness Command Status X = 0 : Normal command X = 1 : Overbrightness command
13	X	Day / Night Function Status X = 0 : Automatic mode X = 1 : Manual mode
14	X	Overbrightness Function Status X = 0 : Automatic mode X = 1 : Manual mode
15	X	Shutter Service Status X = 0 : No service in progress X = 1 : Service in progress
16	X	Default Display Status X = 0 : No current default display X = 1 : Current default displayed now
17	17	Shutter Power Supply X = 0 : Power 0 X = 1 : Bad power supply
18	X	Local Display Message Active X = 0 : No local display on X <= C : Message number displayed (1-12) X = D : Not used X = E : Not used X = F : Test message displayed

Error Case:

Offset	Format	Description
1	C	Error occurred
2	X	X = 1 : Unknown function code X = 3 : Reset Indicator X = 4 : Syntax error in command X = 5 : Undefined subsign X = 7 : Length error in command format

3. Lamp Status. The Lamp Status command requests the status of the 12 lamps controlled by a lamp control module. A lamp control module is capable of driving up to 12 lamps.

In a standard DMS configuration, 6 of these lamps are used as primary lamps, and 6 are associated backup lamps. The structure of the command Lamp status command is:

Offset	Format	Description
1	D	Command function
2	X	Lamp control module number X <= X <= 3

The structure of the reply received from the DMS controller is:

Error Free Case:

Offset	Format	Description
1	D	Function correctly processed
2	X	X = 0 : Test performed X = 1 : Test not performed
3	X	X = 0 : Configuration
4		Analog loop-back X = 0 : With loop-back X = 1 : Without loop-back
5	X	48 vilts X = 0 : No 48 volts X = 1 : 48 volts ok
6	XXXXXXXXXXXX	12 bytes corresponding each to a lamp byte 6 = Lamp 1 .... Byte 18 = Lamp OUT and OFF For each byte: X = 0 : Lamp OK X = 1 : Lamp OUT and OFF X = 2 : Lamp OUT and ON X = 3 : No significant

Error Occurred Case:

Offset	Format	Description
1	d	Error occurred
2	X	X = 1 : Unknown function code X = 3 : Reset indicator X = 4 : Syntax error in command X = 5 : Undefined subsign X = 7 : Length error in command format

4. **Day/Night Changeover.** The Day/Night Changeover command activates the Day/Night change over relay. The brightness can be set to either day or night levels. The structure of the command is as follows:

Offset	Format	Description
1	G	Command function
2	X	Change of Day / Night X = 0 Day -> Night X = 1 Night -> Day
3	X	Control X = 0 Automatic Mode X = 1 Forced Mode

The structure of the reply received from the DMS controller is:

Error Free Case:

Offset	Format	Description
1	G	Function correctly processed
2	0	No error occurred

Error Occurred Case:

Offset	Format	Description
1	g	Error occurred
2	X	X = 1 : Unknown function code X = 3 : Reset indicator X = 4 : Syntax error in command X = 7 : Length error in command format X = B : Day/Night change over impossible

5. Abort Sign. The Abort Sign command is sent by the TOCCM to activate the lamp power supply abort relay, thus blanking the sign. The DMS controller should receive any commands sent after an abort command but the commands should not be processed or displayed. A un-abort (an Abort command with the restore option specified) command must be sent by the TOCCM to restore the power to the lamps (and thus the DMS controller should start normal processing again).

The structure of the Abort Sign command is as follows:

Offset	Format	Description
1	H	Command function
2	X	X = 0 48 V cut off X = 1 48 V restored

The structure of the reply received from the controller is:

Error Free Case:

Offset	Format	Description

Offset	Format	Description
1	H	Function correctly processed
2	0	No error occurred

Error Occurred Case:

Offset	Format	Description
1	h	Error occurred
2	X	X = 1 : Unknown function code X = 3 : Reset indicator X = 4 : Syntax error in command X = 7 : Length error in command format X = C : Abort change over impossible

6. Clock/Calendar Read Command. The Clock/Calendar Read command transmitted by the TOCCM to read the current time and date from the DMS controller's clock/calendar board. The structure of the command is as follows:

Offset	Format	Description
1	L	Command function

The structure of the reply received from the controller is:

Offset	Format	Description
1	L	Command function
2	XX	Seconds (00 <= XX <= 59)
4	XX	Minutes (00 <= XX <= 59)
6	XX	Hours (00 <=XX <= 23) Military Time
8	XX	Day of month (01 <= XX <= 31)
10	XX	Month (1 <= XX <= 12)
12	X	-Day of week (1 <= XX <= 7) CAUTION : Monday = 1
13	X	Initialization X = 0 Not initialized X = 1 Time initialized

Error Occurred Case:

Offset	Format	Description
1	1	Error occurred

Offset	Format	Description
2	X	X = 1 : Unknown function code X = 3 : Rest indicator X = 4 : Syntax error in command X = 7 : Length error in command format X = A : Function not implemented

7. Clock/Calendar Set Command. The Clock/Calendar Set command is issued by the TOCCM to initialize the time and date in the DMS controller's clock/calendar board. The structure of the command is as follows:

Offset	Format	Description
1	M	Command function
2	00	Seconds can not set seconds
4	XX	Minutes (01 <= XX <= 59)
6	XX	Hours (00 <= XX <= 23) Military Time
8	XX	Day of month (1 <= XX <= 31)
10	XX	Month (1 <= XX <= 12)
12	X	Date of Week (1 <= X <= 7) CAUTION: Monday 1

The structure of the reply received from the controller is:

Error Free Case:

Offset	Format	Description
1	M	Function correctly processed
2	0	No error occurred

Error Occurred Case:

Offset	Format	Description
1	m	Error occurred
2	X	X = 1 : Unknown function code X = 3 : Reset indicator X = 4 : Syntax error in command X = 7 : Length error in command line X = A : Function not implemented

8. Overbrightness Setting. The Overbrightness Setting command is issued by the TOCCM to turn on both the normal and backup lamps simultaneously. This is used to obtain a greater brightness (overbrightness mode). The structure of the command is as follows:

Offset	Format	Description
1	R	Command function
2	X	Change of Normal/Overbright X = 0 Normal Mode X = 1 Overbrightness Mode
3	X	Control X = 0 Automatic Mode X = 1 Forced Mode

The structure of the reply received from the controller is:

Error Free Case:

Offset	Format	Description
1	R	Function correctly processed
2	0	No error occurred

Error Occurred Case:

Offset	Format	Description
1	r	Error occurred
2	x	X = 1 : Unknown function code X = 3 : Reset indicator X = 4 : Syntax error in command X = 7 : Length error in command format

9. Echo. The Echo command is issued by the TOCCM to return the message being displayed with all associated parameters to the TOCCM. The structure of the command is as follows:

Offset	Format	Description
1	S	Command function
2	X	Subsign number 0 <= X <= 7

The structure of the reply received from the controller is:

Offset	Format	Description
1	P B Q or D	Last display command sent
2	X	Subsign number
3	X	X = 0 : Deferred display 1 : Immediate display

Offset	Format	Description
4	X	X = 0 : Steady messages 1 : Two alternated messages
5	X	X = 0 : Text 1 steady 1 : Text 1 flashing
6	X	Text 1 flashing time ON 3 <= X <= F (.3 sec <= X <= 1.5 sec)
7	X	Text 1 flashing time OFF 3 <= X <= F(.3 sec <= X <= 1.5 sec)
8	X	X = 0 : Text 2 steady 1 : Text 2 flashing
9	X	Text 2 flashing time ON 3 <= X <= F (.3 sec <= X <= 1.5 sec)
10	X	Text 2 flashing time OFF 3 <= X <= F (.3 sec <= X <= 1.5 sec)
11	XX	Text 1 time ON if alternated 3 <= XX <= FF (.3 <= XX <= 25.5 sec)
13	XX	Blank time between Text 1 and Text 2 0 <= XX <= FF (0 <= XX <= 25.5 sec)
15	XX	Text 2 time ON if alternated 3 <= XX <= FF (.3 <= XX <= 25.5 sec)
17	XXXX	Display time (overall) 1 < XXXX < FFFF FFFE = 65534 min FFFF = infinite
21	X	Brightness of Text 1 X = 0 : Normal X = 1 : Overbright
22	X	Brightness of Text 2 X = 0 : Normal X = 1 : Overbright
23	X	Text follows or Controller est X = @ : Text follows X = 1 : Preprogrammed text
24	XX	Number of characters in Text 1 (hexadecimal ASCII coded format)
26	XX	Number of characters in Text 2 (hexadecimal ASCII coded format)
28	X....X or XX	N characters describing Text 1 or programmed text number (see 23)

Offset	Format	Description
xx	X....X or XX	N characters describing Text 2 or programmed text number

Error Occurred Case:

Offset	Format	Description
1	s	Error occurred
2	X	X = 1 : Unknown function code X = 3 : Reset indicator X = 4 : Syntax error in command X = 5 : Undefined subsign X = 7 : Length error in command format X = D : Subsign off

10. Simulation Control. The simulation command allows the TOCCM to control the simulation control of a controller. When a controller enters simulation mode, if the controller is displaying a message, it blanks the sign. From this point forward, any and all messages to be displayed on the sign will be simulated. The controller will report the messages as being displayed, but the sign will remain blank. The sign status reports if the sign is in simulation mode. When the controller is in simulation mode, the sign will display the current message in the controller, or blank the sign if the controller received a blank sign command.

This command exists so that development efforts can be performed on the TOCCM without presenting information to the field DMS signs. The structure of the command is as follows:

Offset	Format	Description
1	Z	Command function
2	X	Simulation mode X = 0 : Exit simulation mode X = 1 : Lamp and shutter simulation

The structure of the reply received from the controller is:

Error Free Case:

Offset	Format	Description
1	Z	Error occurred
2	0	No error occurred

Error Occurred Case:

Offset	Format	Description
1	z	Error occurred

Offset	Format	Description
2	X	X = 1 : Unknown function code X = 3 : Reset indicator X = 4 : Syntax error in command

G. Error Codes. An error free result of the command is a buffer starting with the same upper case letter as the command issued. If no data is expected from the DMS controller, the response is the upper case letter of the command followed by "0" indicating correct processing.

If an error occurred, the lower case letter of the command is followed by an error code. This indicates that the command has not been executed. The following tablelists the possible errors that may occur:

Error Code	Description
1	Unknown function code
2	EEPROM parameter error
3	Reset indicator
4	Syntax error in command
5	undefined subsign
6	Unknown preprogrammed text
7	Length error in command format
8	Text with non ASCII characters
9	Unknown MCL (lamp) module
A	Function not implemented
B	Day/Night change over impossible
C	Abort change over impossible
D	Subsign off
E	Time not initialized
F	CAN board not enabled or faulty sensor

3. CCTV Field Equipment. The following describes the software protocol used to communicate between the TransGuide Operations Center (TOC) and the CCTV Field Equipment. The protocol is based on sequences of ASCII characters transmitted via a serial data connection between the TOC and the Video Cameras.

A. General Command Structure. The communication between the TOC and the CCTV Field Equipment can be one-way in either direction or two-way, depending on the specific command. Commands are sent to the CCTV Field Equipment using ASCII characters and one of the following formats:

1. *$<CODE><TYPE>[DATA]<carriage return><line feed>* where each character in the string is OR'ed with 80hex to set the most significant bit.

2. <CODE>[DATA]

   The CCTV Field Equipment is controlled by sending one or more commands terminated by a <carriage return>. The commands are not be executed by the camera until the <carriage return> is received. After execution, a <carriage return> is sent back to the TOC to acknowledge completion.

   Note: [DATA] represents 1 or more parameters required by the command. Some commands do not require data. No space is needed before the 1st parameter; if two or more parameters are transmitted, the parameters must be separated by a space.

   Important Note: In cases where the host does not monitor responding carriage returns from the controller, the host will not know whether a command line has been completed. Though the controller normally can receive new commands while it is processing previous commands, there is an exception when the controller is processing a preset position recall. During a preset recall, all incoming characters are ignored except for the ASCII abort character, <CtrlX> (hex 18) which tells the controller to abort the recall. If the host unit does not monitor the controller to see whether a recall is complete, it should send a <CtrlX> preceding all command lines. This way, if a recall was in progress, it will be aborted and then the controller will accept the command; if no recall was in progress, <CtrlX> will be ignored. The TOC utilizes this paradigm.

B. Command Summary. The following commands are defined in this document.

- Shutter speed
- Auto black balance
- Auto white balance
- Auto white balance select
- Color bars on
- Gain select
- Menu
- DTL select
- IRIS control
- Title number
- Title up (display the title)
- Title down (turn off the title)
- Title set
- Stop/Start Command
- Auto Setup/File
- Manual Integration
- Backlight Compensation
- Color/Monochrome
- Auto Integration
- Auto Focus
- IRIS data

- Analog data
- Enable pan/tilt table
- Disable pan/tilt table
- Turn wiper on
- Turn wiper off
- Turn camera on
- Turn camera off
- Extend lens to 2X
- Retract lens to 1X
- Set all controls to no motion
- Set pan
- Set tilt
- Set zoom
- Set focus
- Abort command
- Move to preset position
- Move to specified pan, tilt, zoom, focus position
- Store the current position into preset
- Store the specified pan, tilt, zoom, focus into preset
- Query camera for it's pan, tilt, zoom, focus position
- Camera titling
- Set sector limit
- Set sector label
- Delete sector data
- Delete all sector data

C. Camera Commands without Data. Camera Commands without data are sent to the CCTV Field Equipment using the following format:

$<CODE><TYPE><carriage return><line feed>

where

$ will prefix each command,
CODE -- Two characters which specify the command code
(i.e. What the command is),
TYPE -- Two characters which specify the type of command,
<carriage return> <linefeed> will terminate each command, and
each ASCII character transmitted will be logical OR'ed with '80' hex before being sent to the camera by the TOC.

Notes:

- The examples in this document omit the prefix, suffix, and the OR'ing of bits.
- A <carriage return> terminates the set of one or more commands.
- A <carriage return> is returned upon command execution.

1. **Shutter Speed.** The Shutter Speed command is sent by the TOC to change the shutter speed of the camera. The command is structured using the following data:

Code	Type	Description
01	00	Off
	01	Low
	02	Middle
	03	Fast
	04	Auto Shutter

2. **Auto Black Balance.** The Auto Black Balance command is sent by the TOC to enable the auto black balance feature of the camera. The command has the following format:

Code	Type
03	01

3. **Auto White Balance.** The Auto White Balance command is sent by TOC to enable the auto white balance feature of the camera. The command has the following format:

Code	Type
04	01

4. **Auto White Balance Select.** The Auto White Balance select command is sent by the TOC to change the current settings of the auto black and white balance. The command has the following format:

Code	Type	Description
05	00	Off
	01	A
	02	B

5. **Color Bars On.** The Color Bars On command is sent by the TOC to display the color bars. The command has the following format:

Code	Type	Description
08	00	Off
	01	On

6. **Gain Select.** The Gain Select is sent by the TOC to change the gain setting on the camera. The command has the following format:

Code	Type	Description
0A	00	-3 dB
	01	0 dB
	02	+6 dB
	03	+9 dB

Code	Type	Description
	04	+15 dB
	05	+18

7. DTL Select. The DTL command is sent by the TOC to change the DTL setting on the camera. The command has the following format:

Code	Type	Description
17	00	off
	01	-3 dB
	02	0 dB
	03	+3 dB

8. IRIS Select. The IRIS command is sent by the TOC to change the IRIS setting on the camera. The command has the following format:

Code	Type	Description
1B	00	IRIS cap
	01	(Reserved)
	02	Auto
	03	Auto Adjust
	04	Manual

9. Title Number. The Title Number command is sent by the TOC to change the title currently being displayed on the camera. The command has the following format:

Code	Type	Description
1E	00	Page 1
	01	Page 2
	02	Page 3
	03	Page 4
	04	Off

10. Menu. The Menu command is sent by the TOC to display the camera's menu of internal commands. The command has the following format:

Code	Type	Description
0D	00	Menu

11. Menu Up. The Menu Up command is sent by the TOC to have the camera move its current cursor position up when the camera's internal menu is displayed. The command has the following format:

Code	Type	Description
20	01	Up

This command causes the camera to move its current cursor up one position.

12. **Menu Down.** Menu Down command is sent by the TOC to have the camera move its current cursor position down when the camera's internal menu is displayed. The command has the following format:

Code	Type	Description
21	01	Down

This command causes the camera to move its current cursor down one position.

13. **Menu Set.** The Menu Set command is sent by the TOC to inform the camera to set the current selection for the current cursor position. The command has the following format:

Code	Type	Description
22	01	Set

14. **Stop/Start Command.** The Start/Stop command, used in conjunction with the Title set command, is sent by the TOC to have the camera update the title text. The command has the following format:

Code	Type	Description
2C	00	Stop
	01	Start

15. **Auto Setup/File.** The Auto Setup/File command is sent by the TOC to return the camera to default conditions. The command has the following format:

16. **Manual Integration Settings.** The Manual Integration Setting is sent by the TOC to change the integration level of the camera to adjust for low light. The command has the following format:

Code	Type	Description
60	00	Off
	02	1/30
	03	1/15
	04	1/8
	05	1/4

17. **Back Light Compensation.** The Back Light Compensation command is used to enable/disable Back Light compensation to adjust for a bright background. The command has the following format:

Code	Type	Description
61	00	Disable Compensation
	01	Enable Compensation

18. **Auto Integration.** The Auto Integration command is used to enable/disable Auto Integration to adjust for a low light. The command has the following format:

Code	Type	Description
62	00	Disable Auto Integration
	01	Enable Auto Integration

19. Auto Focus. The Auto Focus command is used to enable/disable Auto Focus. The command has the following format:

Code	Type	Description
63	00	Disable Auto Focus
	01	Enable Auto Focus

20. Color/Monochrome. The Auto Color/Monochrome command is used to enable/disable Auto Color/Monochrome. The command has the following format:

Code	Type	Description
64	00	Disable Auto Color/Monochrome (Color only)
	01	Enable Auto Color/Monochrome

D. Camera Commands (Requiring Data). Camera Commands requiring data are sent to the CCTV Field Equipment using the following format:

$<CODE><TYPE>[DATA]<carriage return><line feed>

where

- $ will prefix each command,
- CODE -- Two characters which specify the command code i.e. What the command is),
- TYPE -- Two characters which specify the type of command,
- DATA -- Varies based on command being sent. The format for the data will be further described in the following sections.
- <carriage return> <linefeed> will terminate each command, and
- each ASCII character transmitted will be logical OR'ed with '80' hex before being sent to the camera by the TOC

NOTE: The examples in this document omit the prefix, suffix, and this OR'ing of bits. The command string shall have the following format.

1. IRIS Control. The IRIS control command is issued by the TOC to either open or close the camera IRIS. The format of the command is as follows:

Code	Type	Data	Description
27	04	0000	(open) to 03FF (close)

Note: First send 1B04, then 270403FF to go to manual mode and close iris.

2. Analog. The Analog command is issued by the TOC to change various gain settings. The format of the command is as follows:

Code	Type	Data	Description

Code	Type	Data	Description
28	00	0000	(open) to 03FF (close) Red Gain
	01	0000	(open) to 03FF (close) Blue Gain
	02	0000	(open) to 03FF (close) Green Gain
	03	0000	(open) to 03FF (close) M Ped
	04	0000	(open) to 03FF (close) BPed

E. Pan/Tilt/Zoom and Titling Commands. The commands defined in this section have the following format:

<CODE>[DATA]

Where

- Each command consists of either one or two letters,
- the commands are CASE SENSITIVE,
- any numerical parameters needed for the command are:
  - specified by the string of ASCII digits representing the decimal value. For example, a value of 26 is sent as a '2' followed by a '6', (i.e. hex 32, hex 36).
  - Where a command has more than one parameter, the parameters shall be separated by a space.
  - The numerical parameters follow the command letter(s). A space may be used between the command and the first parameter, but it is not required.
- To complete the command line, a <carriage return> is placed at the end of the ASCII string.

After the command line has been completed by the unit, a <carriage return> will be sent back to the host to acknowledge completion.

IMPORTANT NOTE: The commands in this section DO NOT 1) start with the '$' character, 2) end with a <line feed>, 3) nor is each ASCII character logical OR'ed with '80' hex.

1. Enable Pan/Tilt Table. The Enable Pan/Tilt Table command is issued by the TOC to enable the pan/tilt table. The structure of the command is as follows:

Code	Data
M	none

a. Pan/Tilt Table. Camera motion is controlled by four voltages corresponding to the velocity of pan, tilt, zoom, and focus which are set using the PTZF commands described in the table below. The commands have a single numerical parameter determining the voltage to which the output is set. The value of the parameter is in the range [0, 32767]. 16383 is the value at which no motion occurs. As the value moves away from 16383, the speed of motion increases. The direction of motion is determined by whether the value is greater or less than 16383 as follows:

Command	Less Than (<) 16383	Greater Than (>) 16383
P – Pan	Right	Left
T – Tilt	Down	Up
Z – Zoom	In (Tele)	Out (Wide)
F – Focus	Near	Far

2. **Disable Pan/Tilt.** The Disable Pan/Tilt command is issued by the TOC to disable the pan/tilt table. The structure of the command is as follows:

Code	Data
m	none

3. **Turn Wiper On.** The Turn Wiper On command is issued by the TOC to turn the wiper (located on the outside of the camera housing) on. The structure of the command is as follows:

Code	Data
W	none

4. **Turn Wiper Off.** The Turn Wiper Off command is issued by the TOC to turn the wiper (located on the outside of the camera housing) off. The structure of the command is as follows:

Code	Data
w	none

5. **Turn Camera On.** The Turn Camera On command is issued by the TOC to turn the camera on. The structure of the command is as follows:

Code	Data
V	none

This command shall be acknowledged by a <carriage return>.

6. **Turn Camera Off.** The Turn Camera Off command is issued by the TOC to turn the camera off. The structure of the command is as follows:

Code	Data
v	none

This command shall be acknowledged by a <carriage return>, the camera shall be reset even if Power On/Off is not supported by the camera and the camera IRIS shall be closed even if Power On/Off is not supported by the camera.

7. **Extend Lens to 2X.** The Extend Lens To 2X command is issued by the TOC to engage the 2X lens extender. The structure of the command is as follows:

Code	Data

Code	Data
X	none

8. **Retract Lens TO 1X.** The Retract Lens to 1X command is issued by the TOC to disengage the 2X lens extender. The structure of the command is as follows:

Code	Data
x	none

9. **Set All Controls To No Motion.** The Set All Controls To No Motion command is issued by the TOC to command the camera to stop (NO MOTION). This should not be confused with moving the camera to a home position. The structure of the command is as follows:

Code	Data
R	none

10. **Abort Command.** The Abort command is issued by the TOC to abort the command currently being processed by the camera. The structure of the command is as follows:

Code	Data
CtrlX	none

Note that if no command is in progress, the abort command is ignored.

NOTE: When received, all commands will be removed from the queue and all motion will stop. Even if no commands are in the queue, the Abort command is acknowledged with a <carriage return>.

11. **Move to Preset Position.** The Move To Preset Position command is sent by the TOC to instruct the camera to move to a predefined preset position. The structure of the command is as follows:

Code	Data	
C	n	where: 0 <= n <= 63

12. **Move To Specified Pan, Tilt, Zoom, Focus Position.** The Move To Specified Pan, Tilt, Zoom, Focus Position command is sent by the TOC to instruct the camera to move to a specified condition. The structure of the command is as follows:

Code	Data
CX	<p> <*t*> <*z*> <*f*> Where <p> <*t*> <*z*> <*f*> are described in Section 3.6.

13. **Store the Current Position Into Preset.** The Store the Current Position Into Preset command is sent by the TOC to instruct the camera to store the current position in the specified preset position. The structure of the command is as follows:

Code	Data	
E	n	where: 0 <= n <= 63

14. **Store Specified Pan, Tilt, Zoom, Focus Into Preset.** The Store Specified Pan, Tilt, Zoom, Into Preset command is sent by the TOC to instruct the camera to store the specified settings into the specified preset. The structure of the command is as follows:

Code	Data
EX	*n P T Z F*

where:

0<= n <= 63
<p> <*t*> <*z*> <*f*> are described in Section 3.6.

15. **Query Camera for its Pan, Tilt, Zoom, Focus Position.** The Query Camera for its Pan, Tilt, Zoom, Focus Position command is sent by the TOC to instruct the camera to report its current Pan, Tilt, Zoom, and Focus position. The structure of the command is as follows:

Code	Data
ptzf	none

The camera will respond with the following data:

<p> <*t*> <*z*> <*f*> *: Where* <p> <*t*> <*z*> <*f*> are described in Section 3.6.

16. **Camera Titling.** The Camera Titling command is sent to define the title for the camera. The title is included in the image transmitted by the camera.

Code	Data
T	<T|B> <1|2> <Title Text>

Where:

<T|B> specifies the top or bottom position for the title.
<1|2> specifies line 1 or line 2 of the title.
<Title Text> is up to 20 characters of text.

17. **Set Sector Limit.** The Set Sector Limit command is sent to define the sector limit for a specified sector ID of the camera.

Code	Data
S<L|R>	*n*

Where:

<L|R> specifies the left or right limit for the sector.

<L|R> specifies the left or right limit for the sector.
0 <= n <= 15, sector ID.

18. **Set Sector Label.** The Set Sector Label command is sent to define the sector label for a specified sector ID of the camera. The label is included in the image transmitted by the camera.

Code	**Data**
ST	*n* <Label Text>

Where:

0 <= n <= 15, sector ID.
<Label Text> is up to 24 characters of text.

19. **Delete Sector Data.** The Delete Sector Data command is sent to delete the sector data, i.e. sector limits and sector label, for a specified sector ID of the camera.

Code	**Data**
SD	*n*

Where:

0 <= n <= 15, sector ID.

20. **Delete All Sector Data.** The Delete All Sector Data command is sent to delete all the sector data, i.e. sector limits and sector labels, for all the sector IDs of the camera.

Code	**Data**
SX	none

F. PTZF Voltages. Pan, Tilt, Zoom and Focus voltages are in the range [0, 4095] where 0 typically represents the center position of the full range of motion of a given function. The voltages are actually 12 bit two's complement values, so in essence, the range is [-2048, 2047] which when viewed as unsigned numbers progress from 2048 to 4095, then from 0 to 2047. The following table defines the voltages for Pan, Tilt, Zoom and Focus.

	Position	Voltage
Pan	0 degrees (left)	2400
	174.9 degrees (center)	4095
	175 degrees (center)	0
	360 degrees (right)	1607
Tilt	-90 degrees (down)	2250
	-30.1 degrees (center)	4095
	-30 degrees (center)	0
	40 degrees (up)	1855
Zoom	0% (out/wide)	1825

	50%	0
	50.1%	4095
	100% (full optical zoom)	2600
	2x digital zoom	2400
	4x digital zoom	2200
Focus	Focus In	2500
	Center	4095
	Center	0
	Focus Out	1820

4. Lane Control System. The following describes the software protocol used to communicate between the TransGuide Operations Control Center Mainframe (TOCCM) and the Lane Control Signals (LCS) controller. The protocol is based on sequences of ASCII characters transmitted via a serial data connection between the TOCCM and the LCS Controller.

A. General Command Structure. Data to be exchanged between the TOCCM and the LCS controller is framed into blocks. Each block is comprised of a header, a data body, and end of block information. The header always starts with "SOH" (Star of Header) followed by the address of the controller. The end of the header is a "STX" (Start of Text) to indicate data follows. The data is placed after this STX character, and before the ETX character, which marks the end of the data body. Following the ETX character is a Block Check Character (BBC). Data transmitted to the controller is framed between two "NUL" characters used for synchronization. Data transmitted from the controller is framed between a leading "NUL" character and a trailing "SUB" character.

The following diagram is what a single transmission block looks like:

NUL SOH ADDR STX DATA ETX BCC NUL

Where:

NUL -- Synchronization character (Hex 00)

SOH -- The ASCII Start-of-Header character (Hex 01)

ADDR -- Address of the controller determined as :

A controller address is comprised of two parts. The first part of the address is the physical address of the controller. It is value between 0 and 255.

The second part of the address is the logical address of the controller. It is usually 01, unless the controller is in configuration mode, then the address is 00.

The controller address is sent over the line as five ASCII characters.

STX --The ASCII Start-of-Text character (Hex 02), the STX marks the end of the header and the start of the data

DATA -- Data for to be exchanged, its length, format, and contents are dependent on the command type. The general format of the data area is:

<Name><parameter 1><parameter 2>...<parameter n>

Where:

<Name> --- identifies the command. It is formed using only upper case letters, 1 to 8 characters in length.

<Parameters> - are character blocks constructed with:

A mnemonic which is the parameter identifier (using 1 to 8 alphanumerical characters)

the character '='

a set of alphanumerical characters which constitutes the parameter argument.

The parameter separators are formed with one or more space characters (20h). The number parameters used can vary, and is not limited. Several parameters can be used in the same command multiple times. Inside a parameter, no space character are allowed (specifically between the mnemonic and "=" sign or between '=" sign and the arguments). When text is entered, double quotes " are used to avoid any confusion. The relative position of the different parameters is generally not important.

A response without error has the same structure as the command.

ETX -- The ASCII End-of-Text character (Hex 03)

BCC -- Block check character (to ensure data integrity). The BCC is computed as the arithmetic sum of all the block characters, up to and including the ETX character. When transmitting using a seven-bit data word, only the 7 least significant bits are transmitted.

For the following command:

NUL-SOH-00101-STX-E-ETX-BCC-NUL

the BCC would be:

NUL = 00H
OH = 01H

0 = 30H
0 = 30H
1 = 31H
0 = 30H
1 = 31H

STX = 02H
"E" = 45H
ETX = 03H

===
13DH

BCC = (13DH).AND.(7FH) = 3DH

1. **Communication Termination.** The TOCCM terminates communication with the LCS controller by using an "EOT" (Hex 04) sequence. This sequence signals the controller to be ready to communicate with the computer. The sequence does not include a BCC and comprises only of an "EOT" character framed between 2 "NUL" characters. An example synchronization character string would be:

   NUL-EOT-NUL

2. **LCS Controller Modes.** The LCS controller does not synchronously send data to the TOCCM, rather, the LCS controller communicates data in one of two fashions:

   * A "SELECTING" mode in which the TOCCM requests the LCS controller to receive data. The following command sequence is sent to place the LCS controller in SELECTING mode:

     NUL-SOH-ADDR-SEL-NUL

   * A "POLLING" mode in which the TOCCM requests the LCS controller to send data to the TOCCM. The following command sequences sent to place the LCS controller in POLLING mode:

     NUL-SOH-ADDR-POLL-NUL

   Note that there are no BCC characters attached to the above sequences.

3. **Command Acknowledge.** Once a sequence has been sent, either from the TOCCM or the LCS controller, the listener always acknowledges the reception of data. If the reception was correctly received, an "ACK" (Hex 06) is sent over the line. If the reception was garbled or lost, a "NAK" (Hex 15) is sent over the line. The appropriated character is framed between 2 synchronization characters as follows:

   NUL-ACK-NUL to acknowledge command successfully received NUL-NAK-NUL to acknowledge command not successfully received

B. Command Summary. In general each command starts with a capital letter (its command code), followed by various parameters to form a precise syntax. All commands are embedded between control characters according to the communication protocol discussed in the previous section.

The following commands will be sent from the TOCCM to the LCS controller to control the LCS:

The following commands will be sent from the TOCCM to the LCS controller to control the LCS:

* Activation Command

* Status Request

The details of each of these commands will be further discussed in the following sections.

C. Command Details. The following sections provide the details for each command that is sent between the TOCCM and the LCS controller.

1. LCS Commands Sent From The TOCCM to LCS Controller. The following commands are sent from the TOCCM to the LCS Controller via the communication line.

a. Activation Command. The ANM (Activate Numeric Mode) command is issued by the TOCCM to activate a message on a sign. This command defines the display parameters for one or several Display Units. This command defines which symbols to be displayed specifying the preprogrammed page numbers. The response to the ANM command from the LCS controller is:

ANM

The different parameters for the command are:

ID : The access to this command is protected. This parameter is required

ID=idf/pwd

where:

ID ---- is the parameter identifier
= ----- is a separator
idf --- is a 1 to 8 character string (origin identifier)
/ ----- is a separator and is required whether or not a password is used
pwd --- is a 0 to 8 character string (password)

BC: This parameter defines which type of brightness the controller will use (global for all the Display Units). The syntax is:

BC=A (Automatic)
or
BC=N (Night)
or
BC=D (Day)

In the case where BC=A, the result of the command corresponds to the value of the photocell status. If the BC parameter is not used, the current display brightness value is not modified.

AU: This parameter specifies the Display Unit. This parameter may be used more than once in the same ANM command. In this case, each AU parameter effects all the succeeding parameters up to the next AU parameter.

DSP: This parameter defines the display with numerical mode. The syntax of this command is:

DSP-xxx

where:

xxx is a 1 to 3 numerical character string which represents a preprogrammed message number.

LAMPPWR: This parameter controls the power going to the lamp transformer and thus the lamps for the sign. This is a manual way via a software call to turn off power to the lamps. The syntax is:

LAMPPWR=ON	Supply power to the lamps.
Or	
LAMPPWR=OFF	Turn off power to the lamps.

SIMU: This parameter controls the controller's simulation function. When the controller enters simulation mode, all Display Units are turned off. The controller continues to report the status of the Display Units just as if they were lit. The syntax is:

SIMU=0	No simulation
or	
SIMU=1	Lamp simulation

Examples:

Command: ANM ID-ATMS/sigma BC=A AU=1 DSP=001 AU=2 DSP=001 AU=3 DSP=005 AU=4 DSP=006

Response: ANM

Command: ANM ID=ATMS/sigma LAMPPWR=ON

Response: ANM

Command: ANM ID=ATMS/sigma SIMU=1

Response: ANM

b. Status Request. The Status Request command causes the LCS controller to send the status details of Display Units requested to the TOCCM. Utilizing this command, the TOCCM can request various parameters status from the LCS controller. The different parameters include:

ID: The access to this command is protected. This parameter is required.

ID=idf/pwd

where:

ID ---- is the parameter identifier
= ----- is a separator
idf --- is a 1 to 8 character string (origin identifier)
/ ----- is a separator and is required whether or not a password is used
pwd --- is a 0 to 8 character string (password)

BSC: Request that the current level of the brightness status be returned. The syntax is:

BSC=?

SIMU: Request that the current status of simulation mode function be returned. The syntax is:

SIMU=?

AU: Request the status of a specified AU when used in conjunction with the DSP and LMP parameters. The syntax is:

AU=?

DSP: Request that the current symbol being displayed, or 0 to indicate the sign is blank, be returned. The syntax is:

DSP=?

LMP: Request the current lamps out of service for each symbol for the specified AU be returned. According to the parameters which are present in the command, the following parameters are returned.

BCS: Reports the type of control and the current brightness level for the sign. The syntax is:

BCS=m/v

where:

m is the type of control. (A is for Automatic and M is for Manual) '/' is a separator

v is the brightness level. (N is for Night and D is for Day)

SIMU: Reports the controller current simulation mode. The syntax is:

SIMU=0
(No simulation mode review)

or

SIMU=1
(Simulation mode active)

AU: Report the Display Unit. The syntax is:

AU=n

where:

n is a Display Unit number

DSP: Reports what symbol is being displayed. The syntax is:

DSP=n

where:

n is the preprogrammed symbol number.

IMP: Reports the lamp which are out of service for the Display Unit and symbol (DSP). The syntax is:

LMP=lamp 1,lamp 2,...,lamp n

or

LMP=OK

Examples:

Command: ST ID=ATMS/sigma BCS? SIMU=? AU=1 LMP=? AU=2 LMP=?

Response: ST ID=ATMS/sigma BCS=A/D SIMU=0 Au=1 DSP=1 LMP=1,2 AU=2 DSP=2 LMP=OK

Command: ST ID=ATMS/sigma AU=1 DSP=? LMP=?

Response: ST ID=ATMS/sigma AU=1 DSP=3 LMP=8

D. Error Codes. If the command cannot be executed, the LCS should send the following response to the TOCCM:

<Name><error code>:<error location>

where:

<Name> is the command name in lower case letters

<error code> is a 1 to 5 numerical character string which defines the error type encountered

: character is a separator

<error location> shows the character number (calculated from the first character of the name command) where the error was detected.

The following is a list of error codes that the LCS should transmit:

1001 - Command name too long (> 8 characters)
1002 - Parameter name too long (> 8 characters)
1003 - Parameter argument too long
1004 - Unknown command
1005 - Command not implemented
1006 - ID name too long (> 8 characters)
1007 - ID name too short (no name provided)
1008 - ID password too long (> 8 characters)
1009 - Invalid name/password
1010 - Required ID parameter not supplied
1011 - Access denied
1012 - ID name unknown
1013 - Invalid ID number
1014 - Unknown AU number
1015 - Undefined AU number
1016 - Number of argument is incorrect
1017 - Value out of range
1018 - Too many characters
1019 - Too many pages
1020 - Required DSP parameter not supplied
1021 - Argument length incorrect
1022 - Wrong parameter name
1023 - Invalid argument

5. Local Control Unit. The following describes the software protocol used to communicate between the TransGuide Operations Control Center Mainframe (TOCCM) and the Local Control Units (LCUs). The TOCCM executes a piece of software known as the Local Control Unit Master (LCUM) which communicates to the LCUs installed remotely in the field. The protocol is based on sequences of ASCII data transmitted via a serial data connection between the TOCCM and the LCUs.

A. General Command Structure. The LCUM and LCUs communicate via fixed format messages containing both ASCII and binary formats. These messages have the following generic format:

SOH ID TYPE DATA LRC ETX

Where:

SOH -- The ASCII Start-of-Header character (Hex 01)

ID --- Five characters long field which defines the LCU that the message is addressed to or sent from. Each LCU will have a simple name of the form Unnnn, where nnnn is the decimal representation of a 16 bit number that the LCU considers its name. The name U0000 is reserved for use in the Power Up message.

TYPE -- One character and defines the type of message (details provided in the command detail section).

DATA --- This field may be N/A, and its length, format, and contents are dependent on the message type (details provided in the command detail section).

LRC --- One character longitudinal redundancy check field. The LRC is determined by performing a modulo-256 sum of the eight data bits in all preceding characters in the message (including the SOH).

ETX --- The ASCII End-of-Text character (Hex 03)

B. Command Summary. The following messages will be sent from the LCUM to the LCU:

* Download Message
* Poll Message
* Run Diagnostics Message
* Reset Message

The following messages will be sent from the LCU to the LCUM:

* Power Up Message
* Poll Response (Data) Message
* Diagnostic Results Message

The details of each of these messages will be further discussed in the following sections.

C. Command Details. The following sections provide the details for each message that is sent between the LCUM and the LCUs.

1. LCUM to LCU Messages. The following messages are sent from the LCUM to the LCU. The LCUM is a software module executing on the TOCCM and transmitting the following messages to each of the LCUs configured into the TransGuide network.

a. Download Message. When the LCUM receives the LCU's power up message, or if the LCUM determines that the configuration of the LCU has changed (including device failures), the LCUM will send a download message to the LCU. This message contains the LCU's ID and the configuration of the LCU's 24 input lines (non-trap device, upstream & matching device, downstream & matching device).

The ID, TYPE and DATA areas of the message are structured as follows:

ID: “Unnnn” (LCU specific)
TYPE:”L” (Hex 4B)
DATA: :NAME:LA:LB:LC:...:LX:

Where:

NAME: A five character ID, in the same format as the ID field, to be used by the LCU in its messages to the LCUM.

Lx: A two character field that represents the type of each of the 24 lines, in one of five formats: "NT", "Ux", "Dx", or "FL", or "XX", where:

"NT" (Hex 4D54) is a non-trap device line. "Ux" (Hex 55xx) is the upstream line of a pair, whose downstream partner is line x, whose values range from "A" (Hex 41) to "X" (Hex 58).

"Dx" (Hex 44x) is the downstream line of a par, whose upstream partner is line x, whose values range from "A" (Hex 41) to "X" (Hex 58).

"FL" (Hex 464B) is a failed non-trap, upstream, and downstream line. Note: If an upstream or downstream line is marked as failed, the partner line will be marked as "NT".

"XX" (Hex 5858) is a line that is currently unused.

Example:
<SOH>U0001LU0001UBDAUDDCUFDEUHDGUJDIULDKUN
DMUPDOFLNTUTDSNTNTNTNT<LRC><ET>

This LCU contains 10 trap pairs on lines 1-20 (with a failure on line 17, forcing line 18 to Non-Trap) and four non-trap devices on lines 21 - 24.

Example:
<SOH>U0001LU0001UKULUMUNUOUPUQURUSUTDADBDC
DDDEDFDGDHDUDHBTBTXXXX<LRC><ETX>

This LCU contains 10 trap pairs on lines 1-20, but paired as (1 & 11), (2 &12), (3 & 13), etc., two non-trap devices on lines 21 and 22, and lines 23 and 24 are unused.

Note that the overall message length is 62 characters long.

b. Poll Message. The LCUM will poll each LCU at regular intervals (this interval is under software control on the TOCCM and is nominally set at 20 seconds). The interval can be software adjusted within a range of 10 to 60 seconds.

The ID, TYPE, and DATA areas of the message are structured as follows:

ID: "Unnnn" (LU specific)
TYPE: "P" (Hex 50)
|DATA: :SERIAL:

Where:

SERIAL: A 16-bit cyclical serial number, identifying the specific poll for tracking.

Example: <SOH>U0001P<xy><LRC><ETX>

Note that the overall message length in 11 characters long.

c. Run Diagnostics Message. The LCUM can request the LCU to run one or more of its set of diagnostics. The ID, TYPE and DATA areas of the message are structured as follows:

ID: "Unnnn" (LCU specific)
TYPE: "R" (Hex 52)
DATA: :TESTS:

Where:

TESTS:

A 16-bit mask, where a set bit indicates the corresponding diagnostic should be run.

Example: <SOH>U0001R<<0110000000000000>><LRC><ETX>

This example requests the LCU to run two diagnostics.

Note that the overall message length is 11 characters long.

d. Reset Message. When error conditions are detected on a loop device, the LCUM can request the LCU to reset the line. The ID, TYPE and DATA areas of the message are structured as follows:

ID: "Unnnn" (LCU specific)
TYPE:"S" (LCU specific)
DATA: :LINE:

Where:

LINE: A one- character field that represents the line to be reset. Values from "A" (Hex 41) to "X" (Hex 58), the same line identification used in the Download Message.

Example: <SOH>U0001SB<LRC><ETX>

This example requests a reset on the second line.

Note that the overall message length is 10 characters long.

2. LCU to LCUM Messages. The following messages are sent from the LCUs to the LCUM.

a. Power Up Message. The Power Up message is sent by the LCU when it powers up to inform the LCUM that the LCU needs to be downloaded. The message is sent in response to a Poll Messages until the LCU receives a Download Message.

The ID, TYPE and DATA areas of the message are structured as follows:

ID: "U0000" (Hex 5530303030)
TYPE: "U" (Hex 55)

DATA: LCU name, a 16 bit binary number. The first 12 bits are latched for each LCU, with the last four bits being set from a flywheel on the LCU case. The ASCII decimal version of this number shall be used in the ID field of all further messages.

Example: <SOH>U0000U<xx><LRC><ETX>

Note that the overall message length is 11 characters long.

b. Poll Response (Data) Message. The LCU shall respond to each poll with a Data message (unless a Run Diagnostic message has been received, in which case the poll response is the Diagnostic Results message). The serial number field of the Data message will be transferred from the Poll message being answered.

The ID, TYPE and DATA areas of the message are structured as follows:

ID: "Unnnn" (LCU specific)
TYPE: "D" (Hex 44)
DATA: :SERIAL"DELTAT:CA:CB:CC:...CX:

Where:

SERIAL: A 16 bit cyclical serial number, taken from the received Poll Message.

DELTAT: A 32 bit integer, representing the time the LCU has been accumulating the counts, in milliseconds.

Cx: A field containing two 32 bit subfields. The format of the field depends on the configuration of the corresponding line from the download message. Lines configured as "NT" or "Ux" contain "vvvvoooo". Lines configured as "Dx" contain "ttttssss". Lines configured as "FL" contain "FAILFAIL". Lines configured as "XX" contain "XXXXXXXX". If the LCU detected a problem with a line or resent the line, the field will not contain data but will contain a flag and status code, "nnnncccc", where:

vvvv is the 32-bit integer count of vehicles for the line or pair.

oooo is the 32-bit integer count of total time, in milliseconds, that the device was occupied for the line or pair.

tttt is the 32-bit integer count of vehicles that contributed to the speed time count.

ssss is the 32-bit integer count of total elapsed time between the pair of devices for the pair.

FAILFAIL (hex 4641494B4641494B) is filler for failed lines.

XXXXXXXX (Hex 5858585858585858) is filler for unused lines.

nnnn is a 32-bit flag, set to all ones (Hex FFFFFFFF) marking the field as status instead of data.

cccc is a 32-bit filed of status codes. The list of possible code values is provided in Subarticle 5.4 of this document.

Example:

<SOH>U0001D<xy><time><vvva>oooa><tttb>
<sssb><vvvc><oooc><tttd><sssd><vvve><oooe>
<tttf><sssf><vvvg>>ooog><ttth><sssh><vvvi>
<oooi><tttj><sssj><vvvk><oook><tttl><sssl>
<vvvm><ooom><tttn><sssn><vvvo><oooo><tttp>
<sssp><FAILFAIL<vvvr><ooor><vvvs><ooos>
<tttt><ssst><vvvu><ooou><vvvv><ooov><vvvw>
<ooow><vvvx><ooox><LRC><ETX>

This example uses the configuration from the first download example and matching the Poll Message example.

Example:

<SOH><U0001D<xy><time><vvva><oooa> <vvvb><oob>
<vvvc><oooc><vvvd><oood><vvve><oooe><vvvf>
<ooof><vvvg><ooog><vvvh><oooh><vvvi><oooi>
<vvvj><oooj><tttk><sssk><tttl><sssl><nnnm>
<cccm><tttn><sssn><ttto><ssso><tttp><sssp>
<tttq><sssq><tttt><sssr><ttts><ssss><tttt>
<ssst><vvvu><ooou><vvvv><ooov>
XXXXXXXXXXXXXXX<LRC><ETX>

This example uses the configuration from the second download example, matching the Poll Message example, but with a status code on line M.

Note that the overall message length is 207 characters long.

c. Diagnostic Results Message. If the LCU receives a Run Diagnostics message, the LCU shall run the specified self-diagnostic tests and report the results with the Diagnostic Results message in response to the next Poll Message. The LCU shall respond to each poll with a Data Message. The serial number field of the Results message will be transferred from the Poll message being answered.

The ID, TYPE and Data areas of the message are structured as follows:

ID: "Unnnn" (LCU specific)
TYPE: "G" (Hex 47)
DATA: :SERIAL:R1:R2:R3:...:R16:

Where:

SERIAL: A 16-bit cyclical serial number, taken from the received Poll Message.

Rn: A field containing two 32-bit subfields, representing the results of the corresponding diagnostic. Fields for diagnostics that were not run should be cleared to binary 0.

Example:
<SOH><U0001G<xy><000a><000a><rrrb><rrrb><rrrc><rrrc>
<000d><000d><000e><000e><000f><000f><000g><000g>
<000h><000h><000i><000i><000j><000j><000k><000k>
<0001><0001><000m><000m><000n><000n><000o><000o>
<000p><000p><LRC><ETX>

Note that this example matches the poll message example previously provided.

Note that the overall message length is 107 characters long.

6. Radar Vehicle Sensing Device. The following describes the software protocol used to communicate between the TransGuide Operations Center (TOC) and the Radar Vehicle Sensing Devices if the devices are not connected to Local Control Units via the use of Dual Loop emulation cards. The protocol is based on sequences of ASCII characters transmitted via a serial data connection between the TOC and the devices.

A. General Command Structure. The protocol defines a set of messages using ASCII characters. The ASCII code assigns letters, numbers, punctuation marks, and other common characters to the decimal numbers 0 to 127.

ASCII Codes (Decimal) for Select Characters

Character	Line Feed			Carriage Return (‘/r’)				Space			‘~’	
Code	10			13				32			126	
Character	‘0’	‘1’	‘2’	‘3’	‘4’	‘5’	‘6’	‘7’	‘8’	‘9’		
Code	48	49	50	51	52	53	54	55	56	57		
Character	‘A’	‘B’	‘C’	‘D’	‘E’	‘F’	‘G’	‘H’	‘I’	‘J’	‘K’	‘L’	‘M’
Code	65	66	67	68	69	70	71	72	73	74	75	76	77
Character	‘N’	‘O’	‘P’	‘Q’	‘R’	‘S’	‘T’	‘U’	‘V’	‘W’	‘X’	‘Y’	‘Z’
Code	78	79	80	81	82	83	84	85	86	87	88	89	90
Character	‘a’	‘b’	‘c’	‘d’	‘e’	‘f’	‘g’	‘h’	‘i’	‘j’	‘k’	‘l’	‘m’
Code	97	98	99	100	101	102	103	104	105	106	107	108	109
Character	‘n’	‘o’	‘p’	‘q’	‘r’	‘s’	‘t’	‘u’	‘v’	‘w’	‘x’	‘y’	‘z’
Code	110	111	112	113	114	115	116	117	118	119	120	121	122

For example, the character '0' is assigned the decimal number 48. Each message of the protocol consists of a string of ASCII characters. For example, the 3-character string "XA\r" is sent by an application to request the next event data. (The "\r" character is used here and by some programming languages to represent an ASCII "carriage return". In the ASCII code, a "carriage return" is assigned the decimal number 13.)

When using a polling communication paradigm, the messages can be divided into two groups: requests and responses. Each request-response pair shares the same header substring. For example, both the event data request and response begin with the substring "XA". The header is at the beginning of each message.

The header is followed by the payload and footer as diagrammed below for the response message: "XA01CB3DC5100AF00370~\r\r".

Format of Example Message

	Header	Payload	Footer	
			Checksum	Terminator
Example	"XA"	"01CB3DC5100AF00370"		"~\r\r"

The payload is the data portion of the message. The footer is the used to validate and terminate the message. Validation is performed using a checksum on critical information within a message. Some messages (like the "XA" response) have no checksum. However, when used, the checksum is a 4-character hexadecimal string formulated by adding the numerical ASCII codes of all the characters in a critical substring of the message. As an example, if a checksum was calculated on the payload substring above then the checksum would be determined in the following manner:

	'0'	'1'	'C'	'B'	'3'	'D'	'C'	'5'	'1'	'0'	'0'	'A'	'F'	'0'	'0'	'3'	'7'	'0'
Checksum	= 48	+49	+67	+66	+51	+68	+67	+53	+49	+48	+48	+65	+70	+48	+48	+51	+55	+48

= 999 (Decimal)
= 3E7 (Hexadecimal)
= "03E7" (4-character hexadecimal string)

Termination of a message is indicated by a terminator substring. For request messages, the terminator substring is "\r". For response messages, the terminator substring is a "~\r\r". The device shall return the terminator substring string, however, some devices like an internal CDPD modem will strip off the "~\r" and so only a "\r" will be observed by the receiving application.

The following list of messages is supported. The third column indicates which messages use a checksum.

Message Description	Header	Checksum in Footer
Get Time Interval Data	"XD"	Yes, in Response Only
Get Presence Data	"X1"	No
Get Event Data	"XA"	No
Get Time	"SB"	No

Get Time Interval	"SJS00008E0008"	Yes, in Response Only
Get Baud Rate	"SJS0000970004"	Yes, in Response Only
Get Classification Lengths	"SJS0200000028"	Yes, in Response Only
Set Time	"S4"	No
Set Time Interval	"SKS00008E0008"	Yes, in Request Only
Set Baud Rate	"SKS0000970004"	Yes, in Request Only
Set Classification Lengths	"SKS0200000028"	Yes, in Request Only

B. Retrieving Time Interval Data. Store volume, average speed, occupancy, and vehicle classification statistics for each lane within the device based on a user-configured time-interval, and request this information using the message "XD\r". The "XD\r" request message will retrieve the most recently generated time interval data packet.

In order to retrieve time interval traffic data previously generated on the device, the "XD" header of a request must be followed by a 4-character index substring. For example, if time interval data is being generated every 5 minutes and the time is now 12:07pm, then sending an "XD0002\r" request will retrieve the traffic data generated at 12:00pm. Furthermore, the traffic data generated at 12:05pm can be retrieved by sending any of the following requests: "XD\r", "XD0000\r", or "XD0001\r".

The time interval for traffic data aggregation can be anything from 5 seconds to 1 month. The time interval is specified by an 8-character substring that contains a hexadecimal number representing the number of seconds in the interval.

The time interval is requested using the "SJS00008E0008\r" command. If the time interval is 1 hour, the response will be "SJ00000E100196~\r\r" (See figure below). The time interval substring is comprised of the first 8-characters of the response following the "SJ". For the given example, the time interval substring "00000E10" corresponds to 3600 seconds (1 hour) when converted to decimal. A 4-character checksum is appended to the time interval substring. The checksum is calculated on the characters of the time interval substring.

Format of Get Time Interval Response Message

	Header	Payload	Footer	
		Time Interval	Checksum	Terminator
Length	13	8	4	3
Indices	1-13	14-21	22-25	26-28
Example	"SJS00008E0008"	"00000E10"	"0196"	"~\r\r"
Value		1 hour		

To set the time interval, the time interval substring should first be formulated. The four-character checksum is then calculated. In this case, the checksum is calculated starting with the "S" following the "SK" at the beginning of the message. So for the example in the figure below the checksum is calculated on the substring "S00008E00080000001E". In this example the value of the time interval when converted to decimal is 30 seconds.

Format of Set Time Interval Request Message

	Header	Payload	Footer	
		Time Interval	Checksum	Terminator
Length	13	8	4	3
Indices	1-13	14-21	22-25	26-28
Example	"SKS00008E0008"	"0000001E"	"03EE"	"~\r\r"
Value		30 seconds		

In the event of a success, the response will be "SKSuccess~\r\r". In the event of a failure, the response will be "SKFailure~\r\r".

The device must be capable of storing information in flash (non-volatile) data backup which can buffer up to 2480 intervals.

A successful response to a time interval data request will be as follows:

Example of Successful XD Response

"XD000000B4100000032004B00660333008F003D200000032004B00660333008F003D300000032004B00660333008F003D400000032004B00660333008F003D500000032004B00660333008F003D600000032004B00660333008F003D700000032004B00660333008F003D800000032004B00660333008F003D3062~\r\r"

In the event of a failure, one of three error messages will be returned: empty, invalid, or failure. "XDEmpty~\r\r" is returned if no interval data exists in SRAM or FLASH. "XDInvalid~\r\r" is returned if the index is not valid because it is too large or malformed. "XDFailure~\r\r" is returned if there was a failure while retrieving the interval data from memory.

The payload of a successful response starts after the "XD" header. The 4-character index substring is not present in the response message. As an alternative for sequence verification, the timestamp substring can be checked.

The payload starts with an 8-character **TIMESTAMP** substring that represents the number of seconds from Jan 1, 2000 at 00:00am based on the PC clock's UTC time. In the above example of a successful XD response, the timestamp substring is "000000B4". This timestamp translates to Jan 1, 2000 at 00:03am UTC.

The payload of the timestamp substring is followed by up to 8 lane data substrings. The number of lane data substrings in the response depends upon the number of lanes configured on the device. For instance, if only 3 lanes are configured then only 3 lane data substrings will be returned (See below). Each lane data substring is 29 characters long.

Example Format of XD Response

	Header	Payload				Footer	
		Timestamp	Lane Data 1	Lane Data 2	Lane Data 3	Checksum	Terminator
Length	2	8	29	29	29	4	3

The 29 characters of each lane data substring are grouped into 7 segments as diagramed below. In the successful response above, all 8 of the lane data substrings happen to be exactly the same, except for the (underlined) lane IDs. The lane data substring "200000032004B00660333008F003D" for lane 2 is used as the example below:

Format of Lane Data Substring

	Lane ID	Volume	Average Speed	Occupancy %	Small Class %	Medium Class %	Large Class %
Length	1	8	4	4	4	4	4
Indices	1	2-9	10-13	14-17	18-21	22-25	26-29
Example Substring	'2'	"00000032"	"004B"	"0066"	"0333"	"008F"	"003D"
Decimal Value	2	50 vehicles	75 mph	10.0%	80.0%	14.0%	6.0%

The 1-character **LANE ID** indicates the position of the lane. The closest lane configured is labeled with ID 1. The next closest lane configured is labeled with ID 2, and so on until the last lane.

The 8-character **VOLUME** substring contains a hexadecimal integer that represents the total number of vehicles detected in the lane, during the time interval. In the above figure, the example volume substring "00000032" translates to 50 vehicles.

The 4-character **AVERAGE SPEED** substring contains a hexadecimal integer that represents the average speed vehicles traveled in the lane, during the interval. The speed units are miles per hour or kilometers per hour depending upon whether English or metric units are selected. If English units are assumed, then the example speed substring of the above figure translates to 75 mph.

The 4-character **OCCUPANCY %** substring contains a hexadecimal integer that must be converted to decimal, multiplied by 100, and divided by 1024 to derive the percentage of time lane was occupied during the interval. In the above figure, the example occupancy % substring "0066" translates to 10.0% when rounded.

The 4-character **SMALL CLASS %** substring contains a hexadecimal integer that must be converted to decimal, multiplied by 100, and divided by 1024 to derive the percentage of vehicles whose lengths were classified as small. In the above figure, the example small class % substring "0333" translates to 80.0% when rounded.

The 4-character **MEDIUM CLASS %** substring contains a hexadecimal integer that must be converted to decimal, multiplied by 100, and divided by 1024 to derive the

percentage of vehicles whose lengths were classified as medium. In the above figure, the example medium class % substring "008F" translates to 14.0% when rounded.

The 4-character **LARGE CLASS %** substring contains a hexadecimal integer that must be converted to decimal, multiplied by 100, and divided by 1024 to derive the percentage of vehicles whose lengths were classified as large. In the above figure, the example large class % substring "003D" translates to 6.0% when rounded.

The lane data substrings are followed by the checksum and terminator in the footer as shown in the "Example Format of XD Response". The checksum is calculated on the payload portion of the packet only. In the "Example of Successful XD Response" figure, the checksum substring is "3062".

C. Retrieving Presence Data. To query the device whether vehicles are present in the lanes being monitored, transmit the message request "X1\r". The response from the device shall contain a four-character payload substring that contains a hexadecimal integer. The lower 8 bits of the binary representation of this integer indicate the presence of a vehicle in a particular lane. The least significant bit corresponds to the lane closest to the sensor and the most significant bit corresponds to lane farthest from the sensor. If the bit is set then a vehicle is present in that lane.

To illustrate, suppose that a device monitoring a 4-lane highway is queried for presence and responds with the string "X1000A~\r\r".

Converting "000A" to its binary equivalent yields "00001010" for the lower 8 bits.

The following figure shows how these 8 bits indicate the presence of a vehicle in lanes 1-8.

Example Parsing of Presence Information

	MSB							LSB
Lane	8	7	6	5	4	3	2	1
Bit Value	0	0	0	0	1	0	1	0
Presence	No	No	No	No	Yes	No	Yes	No

D. Retrieving Event Data. Whenever a vehicle leaves the device's detection zone a vehicle event is generated. A vehicle event message is created and stored for every vehicle leaving the device's detection zone. The maximum number of events that the device shall buffer is 10.

To query the device whether vehicles events have been recorded in the lanes being monitored, transmit the command message "XA\r". Only one event can be acquired per request. The first event recorded in the buffer, is the first event returned by the device (First In First Out). Once an event has been requested, it is removed from the device's buffer. If the event buffer is empty, the response is "XAEmpty~\r\r". If an event is found, the response message will look something like "XA01CB3DC51AF00370~\r\r". This response is used as the example below:

Format of Event Data Response Message

	Header	Payload					Footer
		Timestamp	Lane ID	Duration	Speed	Class ID	Terminator
Length	2	8	1	4	4	1	3
Indices	1-2	3-10	11	12-15	16-19	20	21-23
Example	"XA"	"01CB3DC5"	'1'	'AF'	'0037'	'0'	"~\r\r"
Value		20:54:02:092 UTC	1	437.5 ms	55 mph	Small	

The 8-character **TIMESTAMP** substring contains a hexadecimal integer that indicates the number of 2.5 ms increments since the beginning of the day (UTC time). For example, if the timestamp field contains the string "01CB3DC5", then the event occurred at 20:54:02:092 UTC time. (This is true as long as the device time is UTC time.)

The 1-character **LANE ID** indicates the position of the lane in device configuration. The closest lane configured is labeled with ID 1. The next closest lane configured is labeled with ID 2, and so on until the last lane.

The 4-character **DURATION** substring contains a hexadecimal integer that indicates the number of 2.5 ms increments that the vehicle was present in the detection zone. A duration substring of "00AF" translates to 175 increments, or 437.5 ms.

The 4-character **SPEED** substring contains a hexadecimal integer that indicates the speed assigned to the vehicle. A speed substring of "0037" equates to a speed of 55 mph, if the units are English.

The 1-character **CLASS ID** indicates the classification of the detected vehicle. The possible classifications are '0', '1', and '2'; corresponding to small, medium, and large vehicles, respectively. The classification lengths shall be adjustable.

E. Retrieving and Setting the Time. Device clock time is maintained by a hardware timer. This timer provides a resolution of 2.5ms for event data, and 1 second for time-interval traffic data. The clock records time as the number of seconds since Jan 1, 2000 at 00:00am UTC time using a 32-bit integer.

In the event of a power cycle or other irregularity, an application can re-synchronize (set) the seconds count using an "S4" request message. Similarly, an application can retrieve the current seconds count on the sensor using an "SB" request message. The payload for both the "S4" set request and "SB" get response is an 8-character timestamp substring that represents the 32-bit seconds count integer in hexadecimal.

To translate the seconds count to a date and time, the zero-reference point ("00000000" = Jan 1, 2000 at 00:00am UTC) should be used. As an example, a seconds count of "074554BD" translates to Nov 12, 2003 at 20:29:49 UTC. Since the seconds count is a 32-bit integer it will not rollover until the year 2068.

To set the date and time on the device to Nov 12, 2003 at 20:29:49 UTC, send an "S4074554BD\r" request message. In the event of a success, an "S4Success~\r\r" response message will be returned. In the event of a failure, an "S4Failure~\r\r" response message will be returned.

To check the updated time, send an "SB\r" request message. If successful and the date and time is currently Nov 12, 2003 at 20:30:00 UTC, the response will "SB074554C8 \r". In the event of a "SB" request failure the response will be "SBFailure~\r\r".

F. Retrieving and Setting the Baud Rate. The baud rate shall be controllable for the 4 communication ports of the device: RS-232, RS-485, Expansion A (Internal CDPD Modem), and Expansion B. The rates are specified using a 4-character baud rate control substring as specified below. Each character of the substring represents the baud rate of one of the ports.

Format of Baud Control String

	1st Character	2nd Character	3rd Character	4th Character
Port	Expansion B	RS-232	Expansion A	RS-485
Default Code Value	'1'	'0'	'1'	'4'
Default Rate	19.2 Kbps	9600 bps	19.2 Kbps	115.2 Kbps

Baud Rate Encoding Table

ASCII Value	Rate
'0'	9600 bps
'1'	19.2 Kbps
'2'	38.4 Kbps
'3'	57.6 Kbps
'4'	115.2 Kbps
'5'	230.4 Kbps
'6'	460.8 Kbps
'7'	921.6 Kbps
'8'-'F'	Reserved

The baud rate control substring is retrieved using the "SJS0000970004\r" request message. If the baud rate control substring is "1414", then the response will be "SJ141400CA~\r\r". The baud rate control string is composed of the first 4 characters of the response following the "SJ". A 4-character checksum substring and the terminator are appended after the baud rate control string. The checksum is the hex representation of the sum of the ASCII values of the baud rate control string.

To change the baud rate for a specific port, the baud rate control string should first be formulated. A four-character checksum is then appended. In this case, the checksum is calculated starting with the "S" following the "SK" at the beginning of the message. For example, to set the baud rate control string to "1014", the command "SKS00009700041014030D~\r\r" would be sent. Here the checksum "030D " is

calculated on the substring “S00009700041014”. In the event of a success the response will be “SKSuccess~\r\r”. In the event of a failure, the response will be “SKFailure~\r\r”.

G. Retrieving and Setting the Classification Lengths. Specify the classification lengths in feet or in decimeters. There are three classification bins: Class 0 (Small), Class 1 (Medium), Class 2 (Large). A classification bin is defined by specifying a minimum and maximum length of vehicles that will be lumped into that bin. Bins must not overlap each other, or the results will not be as expected.

The classification lengths are specified by a 40-character string. The format of this string is shown below:

Classification Lengths Specification String

	Class 0 Min	**Class 0 Max**	**Reserved**	**Class 1 Min**	**Class 1 Max**	**Reserved**	**Class 2 Min**	**Class 2 Max**
Length	4	4	8	4	4	8	4	4
Example	“0000”	“000A”	“00000000”	“000B”	“001E”	“00000000”	“001F”	“0032”

The classification length specification is requested using the “SJS0200000028\r” command. If the classification lengths are: Class 0 [0,10 ft], Class 1 [11,30 ft], Class 2 [31,50 ft], the response will be:
“SJ0000000A00000000000B001E00000000001F003207D5~\r\r”.

The first 40-characters of the response following the “SJ” is the classification lengths specification string. The string contains 6 hexadecimal numbers that are the minimum and maximum lengths in each classification bin. A four-character checksum is appended after the classification lengths specification string. The checksum is calculated on the characters in the classification lengths specification string.

To set the classification lengths, the specification string should first be formulated. A four-character checksum is then appended. For a set command, the checksum is calculated starting with the “S” following the “SK” at the beginning of the packet. In the following example the checksum “0A03” was calculated on the substring:
“S020000002800000016000000000017002800000000002903E8”.

Suppose the classification lengths need to be changed so that: Class 0 [0,22 ft], Class 1 [23,40 ft], Class 2 [41,1000 ft]. In this case, the request message
“SKS020000002800000016000000000017002800000000002903E80A03~\r\r”
should be sent. In the event of a success the response will be “SKSuccess~\r\r”. In the event of a failure, the response will be “SKFailure~\r\r”.

7. Measurement and Payment. The work performed and materials furnished in accordance with this Item shall not be measured nor paid for directly, but shall be considered subsidiary to the governing specifications for the items of construction in which these materials are used.