PROTEUS-UHD

An Ultra High-Definition Video Overlay

Software Communication Specification (SCS)

Version V1.25

Jan 22, 2026

Contents

1. COMMUNICATION 4
1.1 COMMAND FORMAT 4
1.2 CHECKSUM COMPUTATION 4
1.3 REPLY FORMAT 5
1.4 COMMUNICATION PORTS 5
2. PROTEUSAPP 6
2.1 HOW TO SEND A COMMAND 6
2.2 HOW TO SEND A SCRIPT 7
2.3 FONTS 9
2.3.1 GENERATE FONTS 9
3. TEXT JUSTIFY 10
4. LAYERS 10
5. COMMANDS 11
5.1 $VL01: CLEAR LAYER 11
5.2 $VL06: COPY RECTANGLE AREA 12
5.3 $VL07: PAINT OR ERASE A RECTANGLE AREA 13
5.4 $VL14: DRAW STRING 14
5.5 $VL25: DISPLAY PNG IMAGE 15
5.6 $VL09: ATOMIC COMMANDS 15
5.7 $VL43: SET REGISTER 16
5.8 $VL42: GET REGISTER 16
5.9 $VL30: CONFIGURE REGISTER ATTRIBUTE WHEN & INPUT 17
5.10 $VL44: SET TIME 18
5.11 $VL45: SET DATE 18

5.12 $VL9002: PERFORM POWER ON RESET 18
5.13 $VL31: CONFIGURE COUNT UP/DOWN TIMER 19
5.14 $VL27: CONFIGURE QUADRATURE COUNTERS 20
5.15 $VL20: SAVE MACRO 21
5.16 $VL21: EXECUTE MACRO 21
5.17 $VL22: DELETE MACRO 21
5.18 $VL26: DISPLAY PORTION OF PNG IMAGE 22
5.19 $VL33: ROTATE PNG IMAGE 22
5.20 $VL35: GET STATUS 23
5.21 $VL37: CONFIGURE TPG 24
5.22 $VL38: SELECT TPG PATTERN 24
5.23 $VL40: SAVE FILE 25
5.24 $VL56: DELETE A FILE 25
5.25 $VL41: GET CRC OF A FILE 25
6. APPENDIX A – REGISTER'S NAME & ID 26

1. COMMUNICATION

1.1 COMMAND FORMAT

All commands start with $VL followed by a two-character command ID, a comma, payload (command specific parameters), an asterisk, two-character checksum and end with new line character.

Start	Command ID	,	Payload	*	Checksum	End
$VL	**ii**	**,**	**Command specific parameters**	*****	**CS**	**LF CR**
$VL	07	,	1,100,100,200,200,1,FF0000FF	*	31	0A 0D

An example is shown below:

"$VL07,1,100,100,200,200,1,FF0000FF*31\r\n" // Command ID=07, checksum included (31)
"$VL07,1,100,100,200,200,1,FF0000FF*XX\n" // Command ID=07, checksum excluded (XX)

1.2 CHECKSUM COMPUTATION

The checksum field is the last field in a command and follows the checksum delimiter character "*". The checksum is calculated as an 8-bit XOR of every byte in the command string. Start after the leading $ character. Stop before the trailing * character. The $ and * themselves are not included in the XOR. The hexadecimal values of the most significant and least significant 4 bits of the result is converted to two ASCII characters (0-9, A-F) for transmission. The most significant character is transmitted first.

Example of a C program to compute checksum:

```
char partialCommand [] = "$VL07,1,100,100,200,200,1,FF0000FF", fullCommand[256], checksum=0;

for (int i = 1; i < strlen(partialCommand); i++)
    checksum ^= partialCommand [i];

sprintf (fullCommand,"%s*%02X\r\n", partialCommand, checksum);
```

Although not recommended, checksum can be excluded by replacing it with **XX:**

```
sprintf (fullCommand,"%s*XX\r\n", partialCommand);
```

1.3 Reply Format

The only commands that require a formatted reply are $VL35, $VL41, $VL42.

As an example, reply to the command $VL35 is shown below:

Start	Command ID	,	Reply	,	Checksum	End
$VL	ii	,	Command specific parameters	*	CS	LF CR
$VL	35	,	$VL35,147,V11.00,35,0,0,3840,,,101	*	43	0A 0D

For the remaining commands, UVO returns a single byte (i.e., ACK or NAK) to indicate whether command was accepted or rejected.

Reply character	Description
6 (ACK)	Command was accepted
0 (NAK)	Command was rejected

1.4 Communication Ports

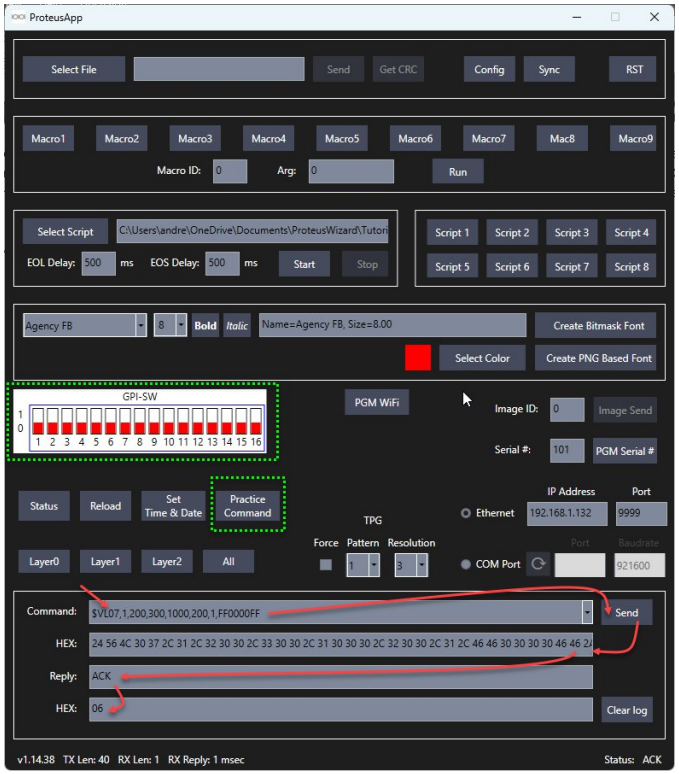
The PROTEUS-UHD system is equipped with four serial ports and one Ethernet port to facilitate communication with external sensors. Refer to the *EXTERNAL INTERFACES* section of the User Manual for detailed instructions on configuring the serial and Ethernet ports.

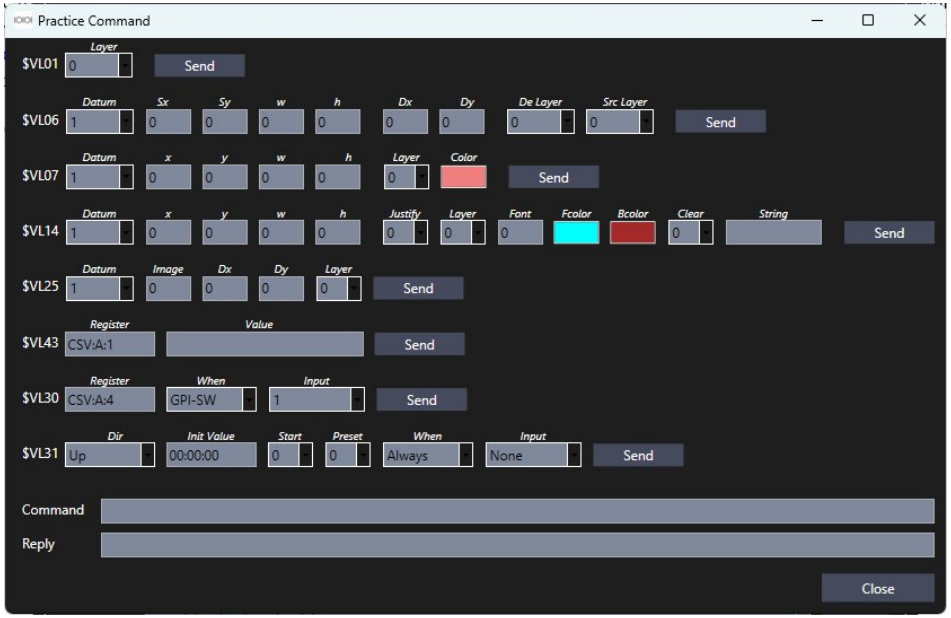
2. PROTEUSAPP

ProteusApp is used to practice sending commands to your PROTEUS-UHD or simulate any sensor with CSV string output.

2.1 HOW TO SEND A COMMAND

1. Follow instructions in *readme.pdf* or **PROTEUS-UM** to establish communication with your PROTEUS-UHD
2. Paste your command i.e. $VL07,1,200,300,1000,200,1,FF0000FF into the *command* box and click *Send,* as shown below on the left. ProteusApp will automatically append checksum and new line characters to your command prior to transmission. The preceding command will draw a red rectangle on your screen.
3. The reply from PROTEUS-UHD will be displayed in the *Reply* and *HEX* boxes.





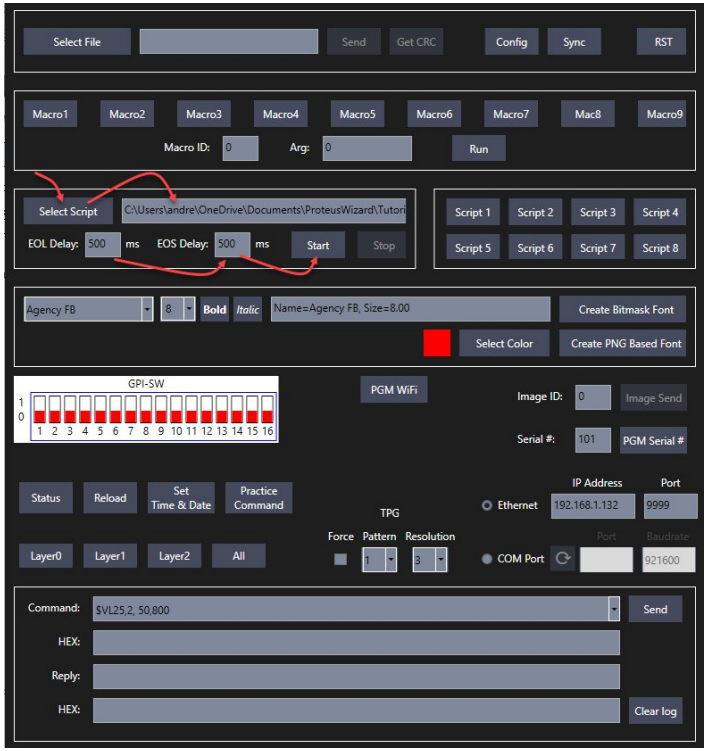
The user can practice building and transmitting common commands by selecting the 'Practice Command' button.
The sixteen-position switch can be used to set or clear any of the GPI-SW inputs as an alternative to typing the $VL43 command.

2.2 HOW TO SEND A SCRIPT

A script file contains multiple commands to be sent to the PROTEUS-UHD in succession. Script execution includes two delays—one between commands (EOL)and one between iterations (EOS).

Delay	Description
EOL (End of Line)	This is the amount of time ProteusApp pauses after transmitting each command line. This delay can be set to 0.
EOS (End of Script)	The is the amount of time ProteusApp pauses after transmitting the last command line. This delay can be set to 0. If this delay is not 0, ProteusApp will pause for that duration before repeating the entire script

Commands in the script do not require a checksum; ProteusApp automatically appends the checksum before transmission. Follow the steps below to transmit *Document\Videologix\Tutorial\Script Demo4K.txt*. Try running the script using three different EOL and EOS delay values: 0 ms, 25 ms, and 50 ms.



Please ensure that a 4K monitor (3840 × 2160) is connected to Proteus, as the drawings in the ScriptDemo4K script are designed for a 4K display.

The list below shows commands included in ScriptDemo4K.txt file.

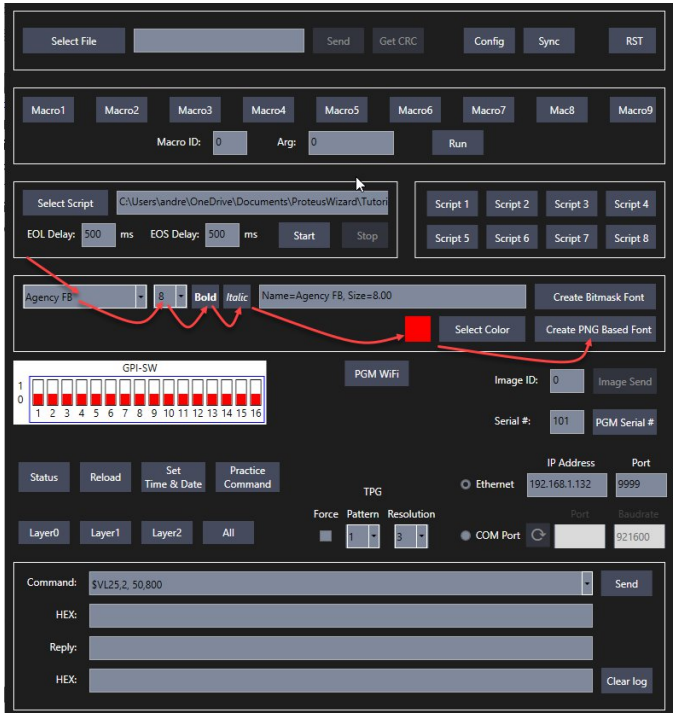
$VL01,A
$VL14,1,100,100,600,100,5,0,11,000000FF,00000000,1,HELLO
$VL14,1,100,200,600,100,5,0,11,0000FF00,00000000,1,WELCOME TO
$VL14,1,100,300,600,100,5,0,11,00FF0000,00000000,1,PROTEUS
$VL14,1,100,400,600,100,5,0,11,00FFFFFF,00000000,1,AN ULTRA HIGH DEFINITION
$VL14,1,100,500,600,100,5,0,11,00000000,00000000,1,VIDEO OVERLAY
$VL07,1,100,600,600,100,1,800000FF
$VL25,1,53,320,800,1
$VL14,1,1100,100,600,100,5,0,10,00FFFFFF,500000FF,1,HELLO
$VL14,1,1100,200,600,100,5,0,10,00FFFFFF,5000FF00,1,WELCOME TO
$VL14,1,1100,300,600,100,5,0,10,00FFFFFF,50FF0000,1,PROTEUS
$VL14,1,1100,400,600,100,5,0,10,00FFFFFF,50000000,1,AN ULTRA HIGH DEFINITION
$VL14,1,1100,500,600,100,5,0,10,00FFFFFF,5000FFFF,1,VIDEO OVERLAY
$VL07,1,1100,600,600,100,1,8000FF00
$VL25,1,63,1320,800,1
$VL14,1,2100,100,600,100,5,0,12,00FFFFFF,50000000,1,HELLO
$VL14,1,2100,200,600,100,5,0,12,00FFFFFF,50000000,1,WELCOME TO
$VL14,1,2100,300,600,100,5,0,12,00FFFFFF,50000000,1,PROTEUS
$VL14,1,2100,400,600,100,5,0,12,00FFFFFF,50000000,1,AN ULTRA HIGH DEFINITION
$VL14,1,2100,500,600,100,5,0,12,00FFFFFF,50000000,1,VIDEO OVERLAY
$VL07,1,2100,600,600,100,1,80FF00FF
$VL25,1,65,2320,800,1
$VL14,1,1100,1200,600,100,5,0,10,00000000,00000000,0,FONT 10 PNG
$VL14,1,1100,1300,600,100,5,0,11,00000000,00000000,0,FONT 11 PNG
$VL14,1,1100,1400,600,100,5,0,12,00000000,00000000,0,FONT 12 PNG
$VL14,1,1100,1500,600,100,5,0,13,00000000,00000000,0,FONT 13 PNG
$VL14,1,1100,1600,600,100,5,0,14,00000000,00000000,0,FONT 14 PNG

2.3 FONTS

PROTEUS-UHD supports any Windows font with full anti-aliasing (subpixel rendering) to ensure smooth, high-quality text rendering. It includes a set of built-in fonts and allows an unlimited number of additional fonts to be loaded. To add additional fonts, please follow the instructions below:

2.3.1 GENERATE FONTS

- Start ProteusApp
- Pick your font style i.e. Arial and a font size i.e. 16
- Select a *Color*
- Press "*Create PNG Based Font*" button
- Browse the folder *Videologix\fonts* and select a font template i.e. iso8859-1 Latin1.txt This file defines a list of characters to include.
- Provide a file name for your font i.e. Aerial16P and press *Save*
- ProteusApp creates two files Aerial16P.FNT & Aerial16P.PNG in the same folder
- Using ProteusWizard, edit asset FontList.txt and add line 15,Aerial16P.fnt which assigns ID 15 to this font. You may assign any ID > 10
- Use ProteusApp to sync new files to your PROTEUS-UHD. See PROTEUS-UM for more details



3. TEXT JUSTIFY

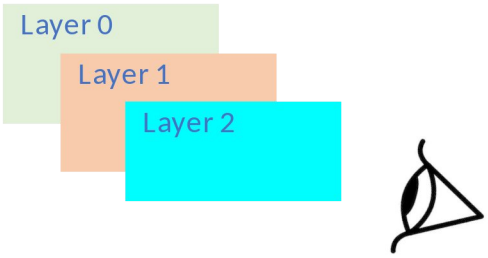
Certain commands allow text to be justified within a rectangular area or graphic object. This is achieved by setting the 'Justify' parameter within the command:

Justify	Description	
1	*Upper Left.*	*UL*
2	*Upper Center.*	*UC*
3	*Upper Right.*	*UR*
4	*Center Left.*	*CL*
5	*Center Center*	*CC*
6	*Center Right.*	*CR*
7	*Lower Left.*	*LL*
8	*Lower Center.*	*LC*
9	*Lower Right.*	*LR*

4. LAYERS

With four independent 4K UHD (3840 × 2160) OSD layers, PROTEUS-UHD allows precise composition of sensor data, text, and PNG images with pixel-level control over position, color, alpha-blending, and brightness.

Layer	Description	Color Format	Max layer size
0	Layer 0	α*RGB32* (ααBBGGRR)	3840 x 2160
1	Layer 1	α*RGB32* (ααBBGGRR)	3840 x 2160
2	Layer 2	α*RGB32* (ααBBGGRR)	3840 x 2160
3	Live video	α*RGB32* (ααBBGGRR)	3840 x 2160



5. COMMANDS

Only the commands $VL01, $VL07, $VL14, and $VL25 are required to create rich text and graphics. It is highly recommended to initially focus on mastering these four commands to effectively achieve your objectives.

5.1 $VL01: CLEAR LAYER

This command clears all content from the specified layer.

Cmnd ID	Payload
01	Layer

Payload	Description	Range
Layer	0 = Layer 0 1 = Layer 1 2 = Layer 2 A = All layers	0..3

Example	
Command:	**Reply:**
$VL01,0 *// Clear layer 0*	ACK
$VL01,2 *// Clear layer 2*	ACK

5.2 $VL06: COPY RECTANGLE AREA

Copies a rectangular area from the source coordinates (Sx, Sy) to the destination coordinates (Dx, Dy).

Cmnd ID	Payload								
06	Datum	Sx	Sy	W	H	Dx	Dy	Dst Layer	Src Layer

Payload	Description	Range
Datum	0 = X,Y references center of the rectangle 1 = X,Y references upper left corner of rectangle	*0..1*
Sx	Source location X	*0..3840*
Sy	Source location Y	*0..2160*
W	Width of Rectangle	*0.. 3840*
H	Height of Rectangle	*0.. 2160*
Dx	Destination location X	*0.. 3840*
Dy	Destination location Y	*0.. 2160*
Dst Layer	destination layer	Layers
Src Layer	Source layer	Layers
Example		
Command: $VL06,1,100,100,200,200,600,300,0,1 // Copy Rect area (100,100,200,200) to (600,300) L1→L0		**Reply:** ACK

5.3 $VL07: PAINT OR ERASE A RECTANGLE AREA

This command allows for painting or erasing a rectangular area.

Cmnd ID	Payload						
07	Datum	X	Y	W	H	Dst Layer	color

Payload	Description	Range
Datum	0 = X,Y references center of the rectangle 1 = X,Y references upper left corner of rectangle	*0..1*
X	Center of Rectangle	*0..3840*
Y	Center of Rectangle	*0..2160*
W	Width of Rectangle	*0.. 3840*
H	Height of Rectangle	*0.. 2160*
Dst Layer	Destination Layer	Layers
color	Fill color. If αα == 0, rectangle area is erased	
Example		
Command: $VL07,0,100,100,200,200,1,00000000 // Erase rectangle $VL07,1,100,100,200,200,1,FF0000FF // Fill rectangle with color Red		**Reply:** ACK ACK

5.4 $VL14: DRAW STRING

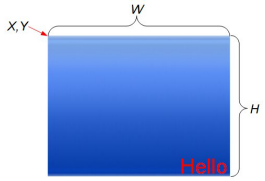
Cmnd ID						Payload						
14	Datum	X	Y	W	H	Justify	Layer	Font	Fcolor	Bcolor	CLR	String

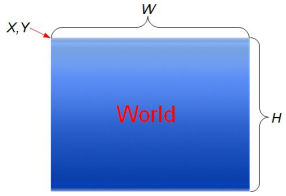
14	1	X	Y	W	H	9	0	2	αRGB32	αRGB32	1	Hello
14	1	X	Y	W	H	5	1	4	αRGB32	αRGB32	1	World
14	1	X	Y	W	H	1	2	7	αRGB32	αRGB32	1	Proteus

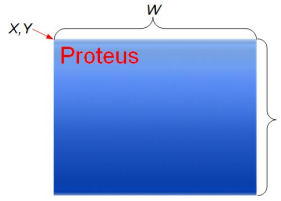
Payload	Description	Range
Datum	0 means X,Y references center of the rectangle area defined by W,H 1 means X,Y references upper left corner of the rectangle area defined by W,H	*0..1*
X	A rectangle area in which to justify the string. Both W & H must be greater than what is required to draw the string	*0..3840*
Y		*0..2160*
W		*0..3840*
H		*0..2160*
Justify	Select text justification within W,H area	*Justify*
Layer	Select destination layer	Layers
Font	Font ID (> 10) used to draw the text	*10..99*
Fcolor	Foreground color. Note αα is ineffective since fonts are anti-aliased	*αRGB32*
Bcolor	Background color is used to clear the area (W, H) if CLR is 1. αα is effective	*αRGB32*
CLR	1 = Clear the rectangle area with Bcolor before printing the string 0 = Do not clear rectangle area. Just add the string to the rectangle area	*0..1*
String	Your text	-

Example

Command:	**Reply:**
$VL14,1,300,100,400,500,9,2,10,000000FF,50FF0000,1,Hello	ACK
$VL14,1,300,100,400,500,5,2,10,000000FF,50FF0000,1,World	ACK
$VL14,1,300,100,400,500,1,2,10,000000FF,30FF0000,1,Proteus	ACK







5.5 $VL25: DISPLAY PNG IMAGE

Cmnd ID	Payload				
25	Datum	Image ID	Dx	Dy	layer

Payload	Description	Range
Datum	0 = X,Y references center of the rectangle 1 = X,Y references upper left corner of rectangle	*0..1*
Image ID	ID	*0..95*
Dx	Destination location X	*0..3840*
Dy	Destination location Y	*0..2160*
Layer	0,1,2	Layers
Example		
Command: $VL25,0,10,100,100,2 // Draw Image #10 @(100,100) on Layer 2 $VL25,1,14,500,500,2 // Draw Image #14 @(500,500) on Layer 2		**Reply:** ACK ACK

5.6 $VL09: ATOMIC COMMANDS

Execute multiple commands all at once. This command guarantees the result will appear into a single frame buffer. There is no limit in number commands included in the payload.

Cmnd ID	Payload
09	Command separated by @

Payload	Description	Range
Command string		*n*
Example		
Command: **$VL09**,**$VL01,0@$VL21,17,0@**$VL07**,1000,400,160,157,0,00000000@		**Reply:** ACK

5.7 $VL43: SET REGISTER

PROTEUS-UHD includes over 200 registers. See APPENDIX A – REGISTER'S NAME & ID for the list of registers.

There are three methods for updating the value of a register:

1. Attached the corresponding sensor such as GPS, IMU, etc. to PROTEUS-UHD
2. Send a CSV string. See Proteus UM for more details
3. Send $VL43 command as described below. Register can be reference using name or ID

Cmnd ID	Payload	
43	Register name or ID	Value

Payload	Description	Range
Register	CSV:A:1 or 1 CSV:A:16 or 16 SYS:GPI-SW:1 or 185 SYS:GPI-SW:16 or 200	*Register Name or ID*
Value	The values assigned to the register	

Example	
Command:	**Reply:**
$VL43,SYS:GPI-SW:1,1 // Set GPI-SW:1 to 1	ACK
$VL43,185,1 // Set GPI-SW:1 to 1	ACK
$VL43,CSV:A:2,Dive#60 // Set CSV:A:2 to Dive#60	ACK
$VL43,2,Dive#60 // Set CSV:A:2 to Dive#60	ACK

5.8 $VL42: GET REGISTER

Cmnd ID	Payload
42	Register

Payload	Description	Range
Register	CSV:H:1 or 113 SYS:GPI-SW:1 or 191	*Register Name or ID*

Example	
Command:	**Reply:**
$VL42,SYS:GPI-SW:1 Or $VL42,191	$VL42,1
$VL42,CSV:A:2 Or $VL42,2	$VL42,Dive#60

5.9 $VL30: CONFIGURE REGISTER ATTRIBUTE WHEN & INPUT

Each register has two associated parameters: **When** and **Control Input**. The **When** parameter defines when the register is displayed. If **When** is set to **GPI**, the **Control Input** field specifies which bit controls the display of the register. When **Control Input** is **1**, the register is displayed; when it is **0**, the register is hidden (removed from display)

Cmnd ID	Payload		
30	Register Name or ID	When	Control Input

Payload	Description	Range
Register	Register name (i.e. GPS:1:TIME) or register number (i.e. 132)	*Name or ID*
When	• Always // Always display this register • GPI-SW // GPI-SW controls whether register is displayed • GPI-HW // GPI-HW controls whether register is displayed • 1HZ // Flash the register at 1Hz • 2HZ // Flash the register at 2Hz • Never // Never display the register	*Enum*
Control Input	• 1 through 16 for GPI-SW 1..16 • 1 through 5 for GPI-HW 1..5 • None	*1..16*

Example	
Command:	**Reply:**
$VL30,CSV:A:1,Always,None // Always display the register	ACK
$VL30,CSV:A:1,Never,None // Don't display the register	ACK
$VL30,CSV:A:1,GPI-HW,5 // Display register if GPI-HW:5 is 1 else remove it	ACK
$VL30,CSV:A:1,GPI-SW,16 // Display register if GPI-SW:16 is 1 else remove it	ACK
$VL30,CSV:A:1,2HZ,None // Flash register at 2HZ	ACK
$VL30,1,2HZ,None // Flash register at 2HZ	ACK

5.10 $VL44: SET TIME

Cmnd ID	Payload
44	Time

Payload	Description	Range
Time	HH:MM:SS	*string*
Example		
Command: $VL44,08:30:45	**Reply:** ACK	

5.11 $VL45: SET DATE

Cmnd ID	Payload
45	Time

Payload	Description	Range
Time	MM/DD/YY	*string*
Example		
Command: $VL45,09:17:25	**Reply:** ACK	

5.12 $VL9002: PERFORM POWER ON RESET

Perform a POR

Cmnd ID	Payload
9002	-

Payload	Description	Range
-		-
Example		
Command: $VL9002	**Reply:** None	

5.13 $VL31: CONFIGURE COUNT UP/DOWN TIMER

Cmnd ID	Payload					
31	direction	preset	start	reset	when	input

Payload	Description	Range
direction	1 = Up 0 = Down	*0-1*
preset	Initial value when reset is asserted	*HH:MM:SS*
start	Start/stop. 1 = Start, 0 = Stop Start/Stop can also be controlled via command $VL43 to set the associated GPI-SW bit	*0-1*
When	• Always • GPI-SW • GPI-HW • 1HZ • 2HZ • Never	*1..6*
Input	• 1 through 16 for GPI-SW • 1 through 5 for GPI-HW • None	*1..16* *1..5* *None*

Example

Command:		**Reply:**
$VL31,0,00:30:45,0,1,Always,None	// Count Down, stop, preset to 00:30:45, always display	ACK
$VL31,0,00:30:45,1,0,Always,None	// Count Down, start, always display	ACK
$VL31,0,00:30:45,1,0,2HZ,None	// Count Down, start, flash timer at 2Hz	ACK
$VL31,0,00:30:45,1,0,Never,None	// Do not display the timer	ACK
$VL31,1,00:30:45,1,0,GPI-SW,1	// Count Down, start, display when GPI-SW:1 is set to 1	ACK
$VL43,SYS:GPI-SW:1,1	// Set GPI-SW:1 to 1	ACK
$VL43,191,1	// Set GPI-SW:1 to 1	ACK

5.14 $VL27: CONFIGURE QUADRATURE COUNTER

This command allows the quadrature counter parameters (Slope, Intercept) to be defined as an alternative to using ProteusApp.

Cmnd ID	Payload			
27	which	slope	intercept	reset

Parm	Description	Range
Which	0 = Quadrature counter 1, 0 = Quadrature counter 2	*0,1*
slope	Slope used to generate mapped value	-
Intercept	Intercept used to generate mapped value	-
Reset	0 = Do not reset the counter. 1 = Reset the counter	*0,1*
Command:		
$VL27,0,1.033,0.00456,0*XX // Counter 0: Slope=1.033, Intercept = 0.00456, Do not reset		
$VL27,1,1.033,0.00456,1*XX // Counter 1: Slope=1.033, Intercept = 0.00456, Reset the counter		

5.15 $VL20: SAVE MACRO

A macro is a series of commands that are executed one after the other in the same order. Macros may have argument.

Cmnd ID	Payload	
20	Macro ID	commandArray []

Payload	Description	Range
Macro ID	Assigned ID	0..63
commandArray []	Group of commands	ASCII
Example		
Command: (*Command below is Macro#10 with 3 arguments ^1, ^2, ^3*) $VL20,10,$VL14,1,300,100,400,500,9,2,10,000000FF,50FF0000,1,^1@$VL43,CSV:A:2,^1@$VL25,^3,100,100,1@		**Reply:** ACK

5.16 $VL21: EXECUTE MACRO

Cmnd ID	Payload	
21	Macro ID	Arguments

Payload	Description	Range
Macro ID	ID	*0..63*
Arguments	If there is no argument, provide at least one i.e., 0	*ASCII*
Example		
Command $VL21,10,0 // Execute Macro #10. No argument $VL21,11,100,200,My,Name,Proteus // Execute Macro #11 with 5 arguments: 100, 200, My, Name, Proteus		**Reply:** ACK ACK

5.17 $VL22: DELETE MACRO

Cmnd ID	Payload
22	Macro ID

Payload	Description	Range
Macro ID	Assigned ID	0..63
Command: $VL22,10 // Delete Macro #10 from memory only		**Reply:** ACK

5.18 $VL26: DISPLAY PORTION OF PNG IMAGE

Display portion of an image that was stored on microSD at a specific location.

Cmnd ID	Payload					
26	Image ID	Dx	Dy	layer	#line	Dir

Payload	Description	Range
Image ID	ID	*0..95*
Dx	Destination location X	*0..3839*
Dy	Destination location Y	*0..2159*
Layer	0	Layers
#line	Number line to display	*n*
Dir	T = From Top, B = From Bottom	*n*
Example		
Command: $VL26,10,100,100,1,7,T // Draw only 7 lines from TOP	**Reply:** ACK	

5.19 $VL33: ROTATE PNG IMAGE

Cmnd ID	Payload						
33	Image ID	Angle	Dst.X	Dst.Y	Image: Axis of Rotation X	Image: Axis of Rotation Y	Dst Layer

It takes 2.2msec to rotate an image of 100 x 100 pixel.

Payload	Description	Range
Image ID	0..10%	*0-63*
Rotate angle	0..359%	*0-359*
Dst.X	Display (frame buffer) coordinate where to place the axis of rotation	0..3840
Dst.Y		0..2160
Image: Axis of Rotation X	Axis of rotation relative to image's Upper Left Corner	*0 .. less than image width*
Image: Axis of Rotation Y		*0 .. less than image height*
Dst Layer		Layers
Example		
Command: $VL33,37,45,1920,1080,87,102,2	**Reply:** ACK	

5.20 $VL35: GET STATUS

Cmnd ID	Payload
35	0

Reply Parm	Description		Range
Video Format	150 = 4K@60p 147 = 4K@30p 145 = 4K@24p 112 = 2K@60p 111 = 2K@50p 109 = 2K@30p	107 = 2K@24p 11 = 2K@60i 10 = 2K@50i 50 = 720@60p 49 = 720@50p	
Version	Firmware version		*String*
Core Temp	FPGA core temperature		*0..100*
coldboot	@POR, coldboot will be set to 1. This flag is reset after the first status reply is sent to host.		*0-1*
Monitor_down	1 = Monitor is down i.e., not connected, powered down, cable removed		*0-1*
Horz Resolution	For 4K, this value will be 3840 For 2K, this value will be 1920		*1920 or 3840*
Example			
Command: $VL35,0		**Reply** $VL35,150,V3.7,45,1,3840,,,	

5.21 $VL37: CONFIGURE TPG

This command Configure Internal TPG (Test Pattern Generator) for desire resolution, frame rate and patten. The pattern will automatically be displayed when no video input is detected.

Cmnd ID	Payload		
37	Resolution	Frame rate	Pattern

Payload	Description	Range
Resolution	3: 720px1280, 4: 720px1680, 5: 1920x1080p, 6: 2560x1080p, 7: 3840 x 2160, 9: 1920x1080i	*3-18*
Frame rate	1: 24Hz, 2: 25Hz, 3: 30Hz, 4: 50Hz, 5: 60Hz, 6: 100Hz, 7:120Hz	*1-7*
Pattern	1: Color bars, 10: Checkerboard, 11: Cross hatch, 12: Noise	*1,10,11,12*
Example		
Command:		**Reply**
$VL37,7,3,12	// 4K@30 + Noise	ACK
$VL37,7,5,12	// 4K@60 + Noise	ACK

Internal TPG can also be configured at powerup via file **tpg_conf.txt**. Parameters shall be stored in the file in order of *Resolution Pattern Frame rate*.

5.22 $VL38: SELECT TPG PATTERN

Select a TPG pattern to display when no video input is detected.

Cmnd ID	Payload
38	Pattern

Payload	Description	Range
Pattern	TPG Pattern. (After power up, pattern switches to default set by command $VL37)	1,10,11,12
Example		
Command:		**Reply**
$VL38,12		ACK

5.23 $VL40: SAVE FILE

This command can be used to save font, image, xml, or text files onto microSD card. For example:

Cmnd ID	Payload	
40	Filename	File content

Payload	Description	Range
Filename	For example: imageList.txt, fontList.txt, imageName.png, fontName.fnt	*ASCII*
File content	For example, byte 34 is sent as two bytes 33, 34	*ASCII-HEX*
Example		
Command: $VL40,bubble.png,89504E470D0A…	**Reply** ACK	

5.24 $VL56: DELETE A FILE

Cmnd ID	Payload
56	File name

Payload	Description	Range
File Name	rov.png	*string*
Example		
Command: $VL56,rov.png	**Reply** ACK	

5.25 $VL41: GET CRC OF A FILE

Compute the 32-bit CRC of the requested file.

Cmnd ID	Payload
41	File name

Payload	Description	Range
Image Name	i.e. bubble.png	*string*
Example		
Command: $VL41,bubble.png	**Reply:** $VL41,1234BCEF	

6. APPENDIX A – REGISTER'S NAME & ID

User owned registers															
CSV String A		CSV String B		CSV String C		CSV String D		CSV String E		CSV String F		CSV String G		CSV String H	
Name	ID	Name	ID	Name	ID	Name	ID	Name	ID	Name	ID	Name	ID	Name	ID
CSV:A:1	1	CSV:B:1	17	CSV:C:1	33	CSV:D:1	49	CSV:E:1	65	CSV:F:1	81	CSV:G:1	97	CSV:H:1	113
CSV:A:2	2	CSV:B:2	18	CSV:C:2	34	CSV:D:2	50	CSV:E:2	66	CSV:F:2	82	CSV:G:2	98	CSV:H:2	114
CSV:A:3	3	CSV:B:3	19	CSV:C:3	35	CSV:D:3	51	CSV:E:3	67	CSV:F:3	83	CSV:G:3	99	CSV:H:3	115
CSV:A:4	4	CSV:B:4	20	CSV:C:4	36	CSV:D:4	52	CSV:E:4	68	CSV:F:4	84	CSV:G:4	100	CSV:H:4	116
CSV:A:5	5	CSV:B:5	21	CSV:C:5	37	CSV:D:5	53	CSV:E:5	69	CSV:F:5	85	CSV:G:5	101	CSV:H:5	117
CSV:A:6	6	CSV:B:6	22	CSV:C:6	38	CSV:D:6	54	CSV:E:6	70	CSV:F:6	86	CSV:G:6	102	CSV:H:6	118
CSV:A:7	7	CSV:B:7	23	CSV:C:7	39	CSV:D:7	55	CSV:E:7	71	CSV:F:7	87	CSV:G:7	103	CSV:H:7	119
CSV:A:8	8	CSV:B:8	24	CSV:C:8	40	CSV:D:8	56	CSV:E:8	72	CSV:F:8	88	CSV:G:8	104	CSV:H:8	120
CSV:A:9	9	CSV:B:9	25	CSV:C:9	41	CSV:D:9	57	CSV:E:9	73	CSV:F:9	89	CSV:G:9	105	CSV:H:9	121
CSV:A:10	10	CSV:B:10	26	CSV:C:10	42	CSV:D:10	58	CSV:E:10	74	CSV:F:10	90	CSV:G:10	106	CSV:H:10	122
CSV:A:11	11	CSV:B:11	27	CSV:C:11	43	CSV:D:11	59	CSV:E:11	75	CSV:F:11	91	CSV:G:11	107	CSV:H:11	123
CSV:A:12	12	CSV:B:12	28	CSV:C:12	44	CSV:D:12	60	CSV:E:12	76	CSV:F:12	92	CSV:G:12	108	CSV:H:12	124
CSV:A:13	13	CSV:B:13	29	CSV:C:13	45	CSV:D:13	61	CSV:E:13	77	CSV:F:13	93	CSV:G:13	109	CSV:H:13	125
CSV:A:14	14	CSV:B:14	30	CSV:C:14	46	CSV:D:14	62	CSV:E:14	78	CSV:F:14	94	CSV:G:14	110	CSV:H:14	126
CSV:A:15	15	CSV:B:15	31	CSV:C:15	47	CSV:D:15	63	CSV:E:15	79	CSV:F:15	95	CSV:G:15	111	CSV:H:15	127
CSV:A:16	16	CSV:B:16	32	CSV:C:16	48	CSV:D:16	64	CSV:E:16	80	CSV:F:16	96	CSV:G:16	112	CSV:H:16	128

Sensor owned registers

GPS		Vector Nav IMU		Real Time Clock		System Register					
Name	ID	Name	ID	Name	ID	Name	ID	Name	ID	Name	ID
GPS:1:ALT	129	IMU:VN:HEADING	143	SYS:RTC:TIME	158	SYS:IMU:HEADING	161	SYS:DEV:1	178	SYS:GPI-SW:1	191
GPS:1:COG	130	IMU:VN:PITCH	144	SYS:RTC:DATE	159	SYS:IMU:PITCH	162	SYS:DEV:2	179	SYS:GPI-SW:2	192
GPS:1:SPEED	131	IMU:VN:ROLL	145	SYS:UTC:OFFSET	160	SYS:IMU:ROLL	163	SYS:DEV:3	180	SYS:GPI-SW:3	193
GPS:1:TIME	132	IMU:VN:HEIGHT	146			SYS:ANLG:MAP1	164	SYS:DEV:4	181	SYS:GPI-SW:4	194
GPS:1:DATE	133	IMU:VN:TIME	147			SYS:ANLG:MAP2	165	SYS:DEV:5	182	SYS:GPI-SW:5	195
GPS:1:LAT_D	134	IMU:VN:DATE	148			SYS:ANLG:MAP3	166	SYS:GPI-HW:1	183	SYS:GPI-SW:6	196
GPS:1:LON_D	135	IMU:VN:LAT_D	149			SYS:ANLG:MAP4	167	SYS:GPI-HW:2	184	SYS:GPI-SW:7	197
GPS:1:LAT_DM	136	IMU:VN:LON_D	150			SYS:ANLG:RAW1	168	SYS:GPI-HW:3	185	SYS:GPI-SW:8	198
GPS:1:LON_DM	137	IMU:VN:LAT_DM	151			SYS:ANLG:RAW2	169	SYS:GPI-HW:4	186	SYS:GPI-SW:9	199
GPS:1:LAT_DMS	138	IMU:VN:LON_DM	152			SYS:ANLG:RAW3	170	SYS:GPI-HW:5	187	SYS:GPI-SW:10	200
GPS:1:LON_DMS	139	IMU:VN:LAT_DMS	153			SYS:ANLG:RAW4	171	SYS:GPI:1HZ	188	SYS:GPI-SW:11	201
GPS:1:EASTING	140	IMU:VN:LON_DMS	154			SYS:QUAD:MAP1	172	SYS:GPI:2HZ	189	SYS:GPI-SW:12	202
GPS:1:NORTHING	141	IMU:VN:EASTING	155			SYS:QUAD:MAP2	173	SYS:INFO:KEY	190	SYS:GPI-SW:13	203
GPS:1:HEADING180	142	IMU:VN:NORTHING	156			SYS:QUAD:RAW1	174			SYS:GPI-SW:14	204
		IMU:VN:HEADING180	157			SYS:QUAD:RAW2	175			SYS:GPI-SW:15	205
						SYS:COUNTER:1	176			SYS:GPI-SW:16	206
						SYS:COUNTER:2	177				

Sensor owned registers					
System Register		NMEA		Misc	
Name	ID	Name	ID	Name	ID
SYS:INFO:IP MODE	207	NMEA:MWV:ANGLE	216	SYS:LTC:TIME	229
SYS:INFO:IP ADDRESS	208	NMEA:MWV:REFERENCE	217	SYS:TIMER:1	230
SYS:INFO:SUBNET	209	NMEA:MWV:SPEED	218	SYS:TIMER:2	231
SYS:INFO:GATEWAY	210	NMEA:MWV:UNIT	219	SYS:IRIB:TIME	232
SYS:INFO:MAC	211	NMEA:DBT:DEPTH	220	SYS:IRIB:DATE	233
SYS:INFO:SNTP TIME	212	NMEA:DBS:DEPTH	221	SYS:GEO:EVENT	234
SYS:INFO:SNTP DATE	213	NMEA:DPT:DEPTH	222	SYS:IMU:HEADING180	235
SYS:INFO:SNTP IP	214	NMEA:DPT:OFFSET	223		
SYS:INFO:FILENAME	215	NMEA:DPT:RANGE	224		
		NMEA:PCIPR:PITCH	225		
		NMEA:PCIPR:ROLL	226		
		NMEA:PCIT:TILT	227		
		NMEA:HCC:HEADING	228		