PROTEUS-UHD

An Ultra High-Definition Video Overlay

User Manual (UM)

Version V1.7
Feb 6, 2026

TABLE OF CONTENTS

GENERAL OVERVIEW

Video Overlay technology allows text and images to be layered over live video, producing visuals that appear naturally embedded in the scene while preserving a clear view of the original environment.

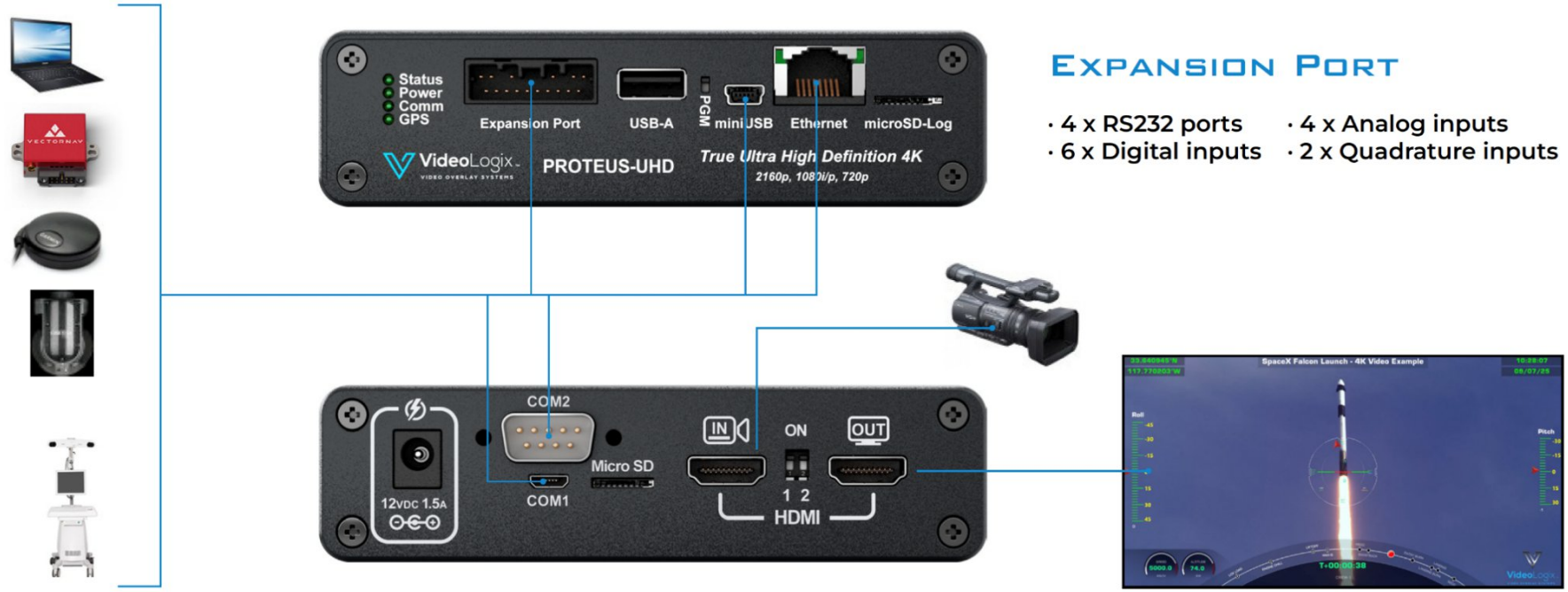
PROTEUS-UHD is designed for overlaying text, images and data onto 2K and 4K live video. This capability is ideal for applications such as broadcasting, surveillance, mapping, and real-time monitoring—where overlays provide critical contextual information.

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

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.

While PROTEUS-UHD offers advanced functionality through additional **widgets** (i.e. ROV situation awareness, sliders, gauges, compass, up/down timer, counters, etc.), its core strength lies in efficient handling of text, image, and geospatial data overlays.

With HDMI 2.1 support, PROTEUS-UHD ensures compatibility with modern high-resolution, high-bandwidth video systems. Its ability to operate **independently of a host computer** simplifies deployment and enhances reliability for **standalone applications**.



GLOSSARY TERMS

Term	Definition
OSD	On Screen Display. ProteusWizard can be used to design your OSD
CSV	Comma-Separated Values. A CSV string consists of a unique header followed by up to 16 comma-separated values
CS	Checksum
SCS	Software Communication Specification
UM	User Manual
GPI	General Purpose Input pin
GPI-HW	Hardware based GPI inputs. All 5 GPI-HW are provided via Expansion Port
GPI-SW	Software based GPI inputs. All 16 GPI-SW are set/clear via command $VL43
WIDGET	A graphic object that is designed to provide a specific piece of information such as time, date, depth, pressure, heading…

INSTALLATION

1. Plug the micro-SD Card Reader (it contains a micro-SD card) into your PC
2. The micro-SD has a folder called ***Videologix***. Copy this folder into the *Windows* ***Documents*** folder
3. Confirm that the folder ***Documents\Videologix*** now exists on your PC
4. Both apps referenced in this document—***ProteusWizard*** and ***ProteusApp***—are in the ***Documents\Videologix*** folder
    - Do not rename the folder.
    - Do not move or modify its location
5. For the ***ProteusWizard*** app to function as WYSIWYG, it is important to set your PC display scale to 100%. Here's how:
    - **Right-click** on your Desktop and select **Display settings**.
    - Scroll down to the **Scale and layout** section.
    - Set **Scale** to **100% (Recommended)**.
    - Close the settings window.

EXTERNAL INTERFACES

COMMUNICATION

COM PORTS

PROTEUS-UHD has **five** serial ports that allow for direct communication with external sensors or systems, making it highly adaptable for various applications.

PORT	Location		Expansion port	Intended to Interface with	Baud rate
COM1	Micro-USB	Rear Panel micro-USB	-	**PC**: Send SCS commands + **Configure Proteus-UHD**	921600, N,8,1
COM2	**RS232**	Rear Panel DB9	2=RX, 3=TX, 5=GND	Various sensors + Send SCS commands	4800-921600, N,8,1
COM3	**RS232**	Internal TB (J12)	1=RX, 3=TX, 2=GND	Various sensors + Send SCS commands	4800-921600, N,8,1
COM4	**RS232**	Expansion Port	1=RX, 2=TX, 5=GND	Various sensors + Send SCS commands	4800-921600, N,8,1
COM5	**RS232**	Expansion Port	3=RX, 4=TX, 5=GND	Various sensors + Send SCS commands	4800-921600, N,8,1
COM6	**RS485**	Expansion Port	14=RX+, 15=RX-, 5=GND	Various sensors + Send SCS commands	4800-921600, N,8,1
COM7	Mini-USB	Front Panel mini-USB	-	**PC**: Send SCS commands + **Update firmware**	921600, N,8,1

When using a USB serial adaptor to communicate with COM2-6 at 460,800 or 921,600, ensure it is capable of 921.6 Kbps

ETHERNET PORT

Ethernet port can be used to send commands defined in SCS *or CSV strings*

- Networking: Static or DHCP IPv4 addressing
- Subnet Mask: Configurable. Default 255.255.255.0
- Default Gateway: 0.0.0.0
- UDP protocol. Port # is hardcoded as 9999

CONFIGURE PROTEUS-UHD

- Follow the setup shown in Figure 1 (Page 19) and adhere to the instructions provided on that page to complete the preparation.
- Press the **Config** button to configure various settings such as baud rate, IP address, CSV headers. When finished, press **OK** to save and exit.
- Press the **Sync** button. When prompted, select any Wizard file such as **Videologix\Tutorial\ Wizard RTC**
- All your configuration settings have been now loaded into Proteus
- To learn more about Proteus, go through the entire *Tutorial* section.

Pressing Alt + h (On the keyboard attached to PROTEUS) will display all configuration settings on the monitor attached to PROTEUS.

VIDEO INTERFACE

HDMI PORT

- Supports HDMI 2.1 video.
- Does *not support* HDMI video with *HDCP*.
- Does not scale video. Output resolution matches input
- In absence of video input, Proteus displays several internal video patterns

VIDEO FRAME RATES

PROTEUS-UHD is compatible with the following video formats:

2160p @	23.97 / 24 / 25 / 29.97 / 30 / 50 / 59.94 / 60 HZ
1080p @	23.97 / 24 / 25/ 29.97/ 30 Hz / 60 HZ
1080i @	50 / 60 Hz
720p @	50 / 59.94 / 60 Hz

VIDEO DELAY

All OSD objects are superimposed into the video stream in real-time, using a process known as "on-the-fly" superimposition. This method ensures that the insertion of graphical elements does not compromise the original video. As a result, the video quality remains intact with no degradation.

Furthermore, the latency introduced by this operation is minimal, with a delay of approximately three horizontal lines from the video input to the video output. This ensures that the video signal is processed and displayed with very little noticeable delay.

EXPANSION PORT

PIN ASSIGNMENT

Pin#	Signal	Alternative	Description
1	COM4 - RX		COM4 UART
2	COM4 - TX		
3	COM5 - RX		COM5 UART
4	COM5 - TX		
5	GND		Ground
6	V5 (optional)		The 5 V supply is intended to power the internal optical isolators used for the quadrature counters. If the 5 V is provided by the customer, reposition the internal jumper **J14** to pins **1–2**
7	QUADRATURE INPUT A1	GPI-HW:1	All 5 inputs are optically isolated inside Proteus. As a GPI-HW*, it can be used to control the ON/OFF state of any graphic widget or to control a count up/down timer.
8	QUADRATURE INPUT B1	GPI-HW:2	
9	QUADRATURE RESET	GPI-HW:3	
10	QUADRATURE INPUT A2	GPI-HW:4	
12	QUADRATURE INPUT B2	GPI-HW:5	
12	ANALOG INPUT1		Range 0-3.3V
13	ANALOG INPUT2		Range 0-3.3V
14	ANALOG INPUT3	Configurable via ProteusApp	See table below
15	ANALOG INPUT4		
16	IRIG-B/SMPTE INPUT		This input can be used for IRIG-B or SMPTE input.

Pin 14, 15	Function
AN3, AN4	ANALOG INPUT3 and ANALOG INPUT4
RS485+, RS485-	Serial port RS485 (COM6)
CAN+, CAN-	Control Area Network (CAN)
SMPTE+, SMPTE-	Time code SMPTE
GPI-HW6, GPI-HW7	General Purpose Inputs
GPI-HW6, IRIG-B	General purpose input and IRIG-B input.

* *With no signal applied to the GPI-HW pins, their default state is logic 1 (5V). To set the state to logic 0, the input pins must be connected to ground.*

CSV STRINGS

The most common method of sending data to PROTEUS-UHD is through comma-separated value (CSV) formatted strings. A CSV string consists of a unique header followed by up to 16 comma-separated values and a checksum.

$Header,VAL1,VAL2,VAL3,VAL4,VAL5,VAL6,VAL7,VAL8,VAL9,VAL10,VAL11,VAL12,VAL13,VAL14,VAL15,VAL16*CS

$	Signifies the start of the string and it is optional
Header	Unique string header. *ProteusApp* is used to define all eight string headers
VALn	Each string contains up to maximum 16 value (VALn) delimited by commas
*	The asterisk serves as a checksum delimiter
CS	The checksum field contains two ASCII characters which indicate the hexadecimal value of the checksum

The table below shows various CSV strings supported by Proteus-UHD:

Type	String includes	String Format	Example	Parsed VALn saved in
CSV1	Unique header + Comma Separated VALn + CS	$HEADER,VAL1,VAL2,,,,,VALn*CS	$STEVE,45,315,200,100*64	Registers 1..128
CSV2	Unique header + Comma Separated VALn	$HEADER,VAL1,VAL2,,,	$BRIAN,45,315,200,100	Registers 1..128
CSV3 **SSV3**	$ + Comma Separated VALn $ + Space Separated VALn	$VAL1,VAL2,VAL3,,, $VAL1 VAL2 VAL3	$45,315,200,100 $45 315 200 100	Registers 65..80 Registers 81..96
CSV4 **SSV4**	Comma separated VALn Space separated VALn	VAL1,VAL2,VAL3,,, VAL1 VAL2 VAL3	45,315,200,100 45 315 200 100	Registers 97..112 Registers 113..128

- Proteus can be configured (using *ProteusApp*) to receive up to 8 unique CSV strings (A, B, C, D, E, F, G, H)
- Upon reception of a CSV string, Proteus verifies checksum & parses the string
- Based on the unique string header, the parsed values (VAL1 - VAL16) are stored in Registers # 1-128 as shown below
- Any widgets linked (via *ProteusWizard*) to these registers will automatically get updated

$Header-A	**VAL1**	**VAL2**	**VAL3**	**VAL4**	**VAL5**	**VAL6**	**VAL7**	**VAL8**	**VAL9**	**VAL10**	**VAL11**	**VAL12**	**VAL13**	**VAL14**	**VAL15**	**VAL16**
Register →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

$Header-B	**VAL1**	**VAL2**	**VAL3**	**VAL4**	**VAL5**	**VAL6**	**VAL7**	**VAL8**	**VAL9**	**VAL10**	**VAL11**	**VAL12**	**VAL13**	**VAL14**	**VAL15**	**VAL16**
Register →	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

$Header-C	**VAL1**	**VAL2**	**VAL3**	**VAL4**	**VAL5**	**VAL6**	**VAL7**	**VAL8**	**VAL9**	**VAL10**	**VAL11**	**VAL12**	**VAL13**	**VAL14**	**VAL15**	**VAL16**
Register →	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48

$Header-D	**VAL1**	**VAL2**	**VAL3**	**VAL4**	**VAL5**	**VAL6**	**VAL7**	**VAL8**	**VAL9**	**VAL10**	**VAL11**	**VAL12**	**VAL13**	**VAL14**	**VAL15**	**VAL16**
Register →	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64

$Header-E	VAL1	VAL2	VAL3	VAL4	VAL5	VAL6	VAL7	VAL8	VAL9	VAL10	VAL11	VAL12	VAL13	VAL14	VAL15	VAL16
Register →	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80

$Header-F	VAL1	VAL2	VAL3	VAL4	VAL5	VAL6	VAL7	VAL8	VAL9	VAL10	VAL11	VAL12	VAL13	VAL14	VAL15	VAL16
Register →	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96

$Header-G	VAL1	VAL2	VAL3	VAL4	VAL5	VAL6	VAL7	VAL8	VAL9	VAL10	VAL11	VAL12	VAL13	VAL14	VAL15	VAL16
Register →	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112

$Header-H	VAL1	VAL2	VAL3	VAL4	VAL5	VAL6	VAL7	VAL8	VAL9	VAL10	VAL11	VAL12	VAL13	VAL14	VAL15	VAL16
Register →	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128

USE CASES

$Header-A,1,22,333,4444,55555,666666,7777777,88888888

$Header-A	VAL1	VAL2	VAL3	VAL4	VAL5	VAL6	VAL7	VAL8
Register →	1	2	3	4	5	6	7	8
Values →	1	22	333	4444	55555	666666	7777777	88888888

$Header-H,This,is,an,Example,,123,,7

$Header-H	VAL1	VAL2	VAL3	VAL4	VAL5	VAL6	VAL7	VAL8
Register →	113	114	115	116	117	118	119	120
Values →	This	is	an	Example		123		7

NMEA 0183

All NMEA 0183 strings such $GPRMC, $GPGGA, $PTSAG, $GPWPL, $GPGSA, $GPGSV, $GPGGL, $SDDPT, $SDDBT, $WIMTW, $WIMWV, $VNINS, $VNIMU, $VNYPR, $PTNTHPR, $HCHDG, $HCHDT, $HCC, $DBS, $PCIT, $PCIPR are CSV formatted string. PROTEUS-UHD intrinsically supports many NMEA strings.

Users may come across a NMEA string that is not supported by PROTEUS-UHD. For example, $PTCF. To configure PROTEUS-UHD to receive this string, use *ProteusApp* and set **CSV Header A** to $PTCF. Upon reception of $PTCF CSV string, PROTEUS-UHD parses the string and parsed values (VAL1..VAL6) are sequentially stored in Registers # 1-6 as shown below:

$PTCF,HHH.H,T,+RRR.R,+PPP.P,+rrr.rr,+ppp.pp

$PTCF	VAL1	VAL2	VAL3	VAL4	VAL5	VAL6
Register →	1	2	3	4	5	6
Values →	HHH.H	T	+RRR.R	+PPP.PP	+rrr.rr	+ppp.pp

DISPLAYING CSV VALUES

Follow *ProteusWizard* and *Tutorial* section to learn how to display CSV values

LINKING WIDGET TO CSV VALUES

Any of the 128 CSV values (registers) can be linked to a widget. When register value changes, the widget is automatically updated. Follow *ProteusWizard* for a step-by-step guide on linking widgets to registers.

SETTING CSV VALUES

CSV registers can be assigned to new values using three different methods. The steps below demonstrate how to set **CSV:H:6** (i.e., VAL6 of the CSV string H) to **123**. Refer to SCS for more details

1. Send the corresponding CSV string i.e. $Header-H,This,is,an,Example,,**123**,,7
2. Send command $VL43,**CSV:H:6**,**123**
3. Send command $VL43,**118**,**123**

See Appendix B for the derivation of value 118.

READING CSV VALUES

CSV registers can be read using two different methods. The steps below show how to read the value of **CSV:H:6** (i.e., VAL6 of the CSV string H). Refer to SCS for more details.

1. Send command $VL42,**CSV:H:6**
2. Send command $VL42,**118**

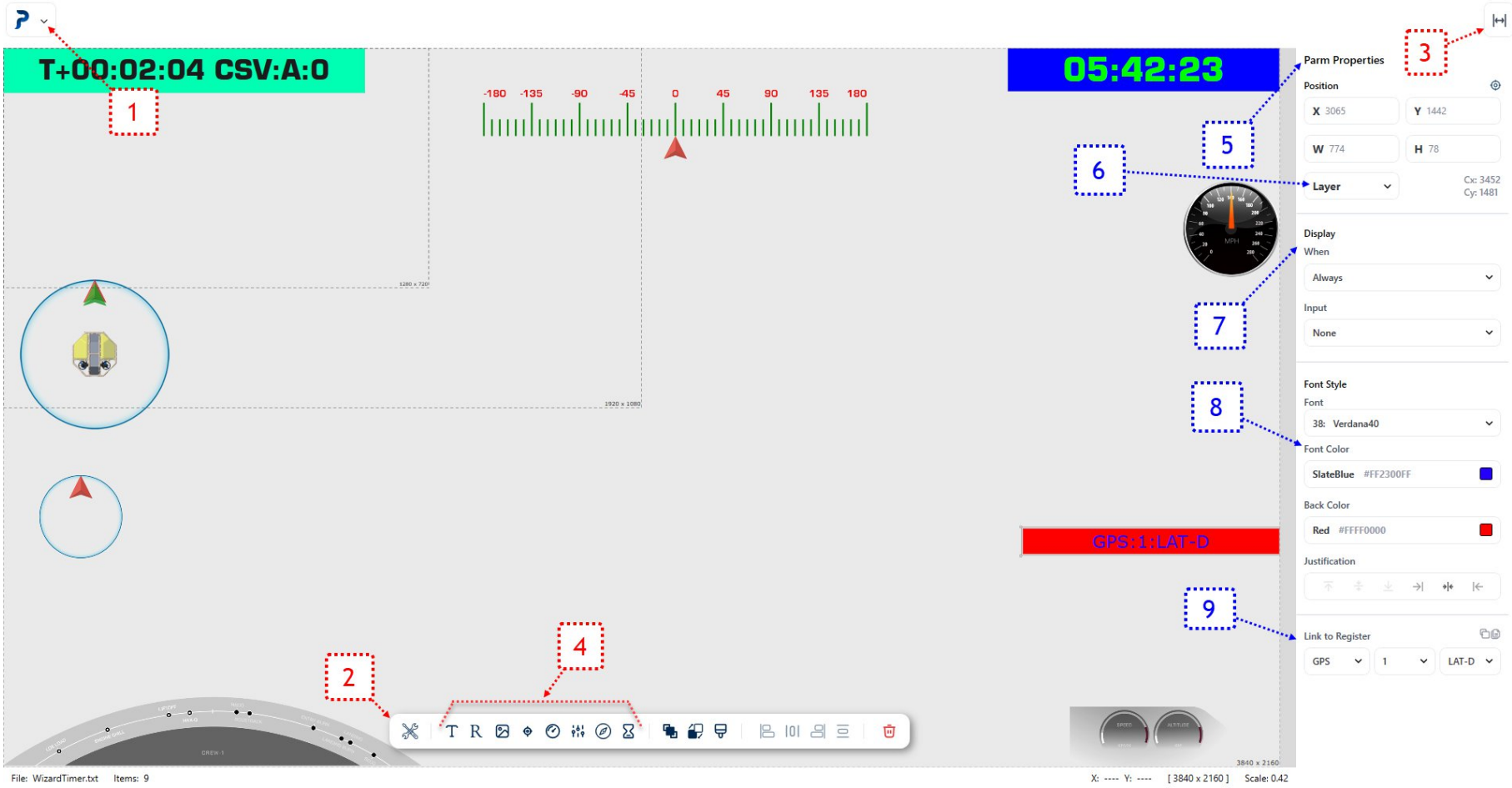
Proteus replies by sending $VL42,**123**

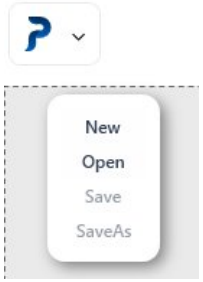
PROTEUSWIZARD

PURPOSE

ProteusWizard is a WYSWYG app for designing your OSD. It allows users to drag and drop graphic widgets like Text, Parameter, Image, Gauge, Slider, Compass, Up/Down Timer into the canvas and adjust their properties i.e. font, color, position, etc. ProteusWizard has a fix **3840 x 2160** canvas with two Dashed Rectangle Outlines depicting **1920 x 1080** and **1280 x 720** video resolutions.

MAIN FEATURES



Feature	Description
1	File operation i.e. **Open** a Wizard design, **Save** current design, etc.  Naming files associated with **ProteusWizard** with the prefix **Wizard** provides several benefits: 1. **Consistency** – Ensures all related files are easily identifiable and grouped together. 2. **Avoids Confusion** – Reduces the chance of mixing ProteusWizard files with unrelated files
2	Before beginning your OSD you must define three classes of assets. Assets serve as the foundational building blocks of your OSD. • TextList.txt This text file contains a list of all available texts • FontList.txt This text file contains a list of all available fonts • ImageList.txt This text file contains a list of all available PNG images
3	Click this icon to zoom canvas to match your video input resolution i.e. **3840 x 2160**, **1920 x 1080** or **1280 x 720** The zoom-in function allows users to position objects more accurately on 1920×1080 and 1280×720 canvas.
4	The “Insert Widget” group allows users to add the following widgets to the canvas: • Text • Register • Image • ROV • Gauge • Slider • Compass • Timer Once a widget is inserted into the canvas, use the Property box [5] to customize it according to your needs.

5

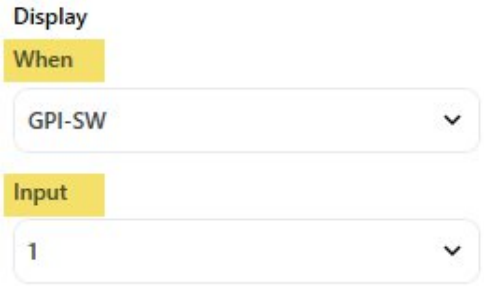
Property boxes allow users to customize individual widgets. Customization options include:

- Position widget on canvas
- Linking widget to a register [9]
- When and how to display the widget [7]
- Select font and assig foreground and background colors to the font [8]
- Which layer to display the widget [6]

6

Select a layer to display the widget. Widgets that overlap must be assigned to different layers to ensure proper rendering

7

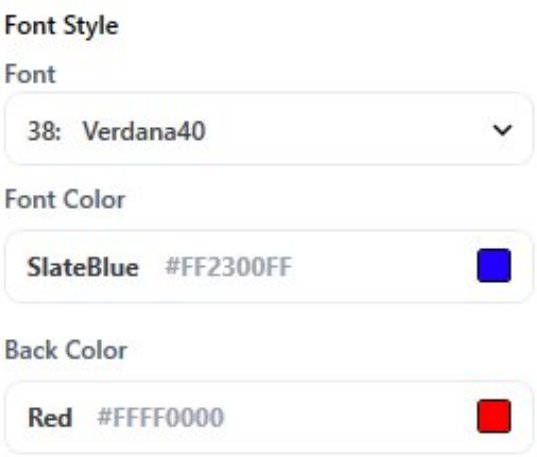


This dialog allows you to specify the conditions under which a widget is displayed.

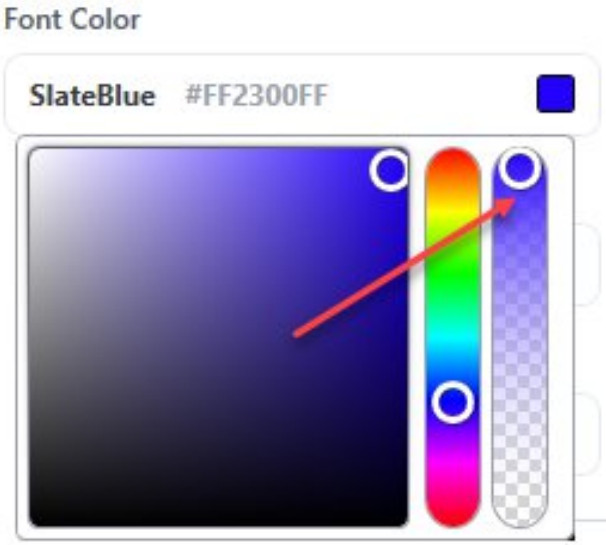
“When”	**“Input”**	**Result**
Always	-	Always display the widget
GPI-HW	1..5	Display widget only if GPI-HW {1,2,3,4,5} is 1
GPI-SW	1..16	Display widget only if GPI-SW {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16} is 1
1HZ or 2HZ	-	Flash widget @ 1HZ or 2HZ
Never	-	Never display the widget on video

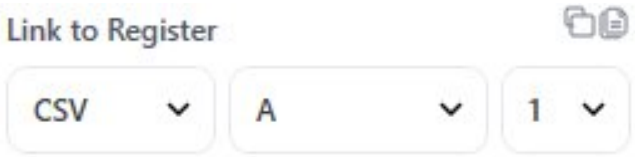
See SCS for how to set or reset GPI-SW

8



This dialog assigns foreground and background colors to the text.
Colors are defined using a 32-bit ARGB format, where **A** represents the alpha (transparency) channel.
If a background color is not needed, set **A** to **0**, as shown below:



9	 This is the most important dialog. It enables users to link any of over 200 available registers to a widget. Registers are organized hierarchically into **Group : Subgroup : Value**. • The available groups include **CSV**, **GPS**, **IMU**, **NMEA**, **SYS** • In this section, we will focus on the **CSV** group • Within **CSV** groups, there are 8 subgroups (**A, B, C, D, E, F, G, H)** • Each **CSV** string has a maximum of 16 Values: **CSV:A:1** is the 1st value in CSV string A **CSV:A:2** is the 2nd value in CSV string A **CSV:A:16** is the 16th value in CSV string A

Refer to Appendix C for the list of register name & ID
See the Tutorial section for instructions on using ProteusWizard to design various OSDs

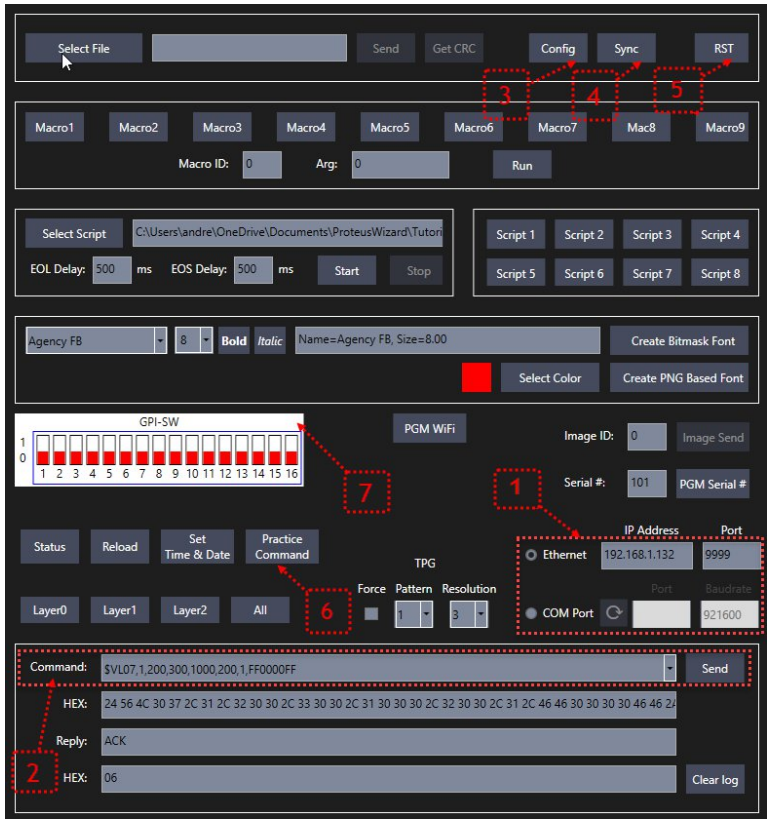
PROTEUSAPP

PURPOSE

- **Download** your OSD design (ProteusWizard output) into the PROTEUS-UHD
- **Configure** COM2,3,4,5 baud rates, Ethernet port IP mode & address, define CSV string headers, configure analog & quadrature inputs, etc.
- **Exercise** sending commands to PROTEUS-UHD
- **Simulate** various sensors and schedule periodic data transmission to PROTEUS-UHD

To **Download** your OSD design or **Configure** PROTEUS-UHD, follow *Figure 1* and adhere to the instructions provided on that page.

MAIN FEATURES



Feature	Description
1	When **configuring** Proteus, connect your PC to COM1 (micro-USB) located on the rear panel. This port is always fixed at 921600. Upon selecting this COM port, immediately press **Clear log** followed by **Status** button to confirm the communication link. When sending commands (defined in SCS) or simulating sensors (sending CSV strings), select Ethernet or any COM port.
2	Enter a command and press **Send**. The ProteusApp automatically appends CRC and transmits the command via selected port. Proteus should respond with an ACK (0x06)
3	Press **Config** button to configure Proteus. Refer to features 8,9,10 below for more details.
4	Once the Configuration is complete, you must press **Sync** button to send the new configuration to PROTEUS-UHD.
5	Press **RST** button to reset Proteus
6	This button allows users to practice building and sending various common commands
7	The sixteen-position switch is used to set/clear any of the GPI-SW (software-based GPI) as an alternative to sending $VL43 command
8	Configure COM2,3,4,5 baud rate and data format expected on the corresponding COM port.
9	Configure Ethernet port settings. Keep in mind, Proteus can only be configured via COM1 (micro-USB). Once configured, only then can you select Ethernet port.
10	Proteus allows up to 8 custom CSV strings each having up to 16 values. Enter the headers of your custom CSV strings. For more detail on CSV strings, refer to *CSV strings*

See section ProteusApp in the **SCS** *document to learn how to send commands to PROTEUS-UHD*

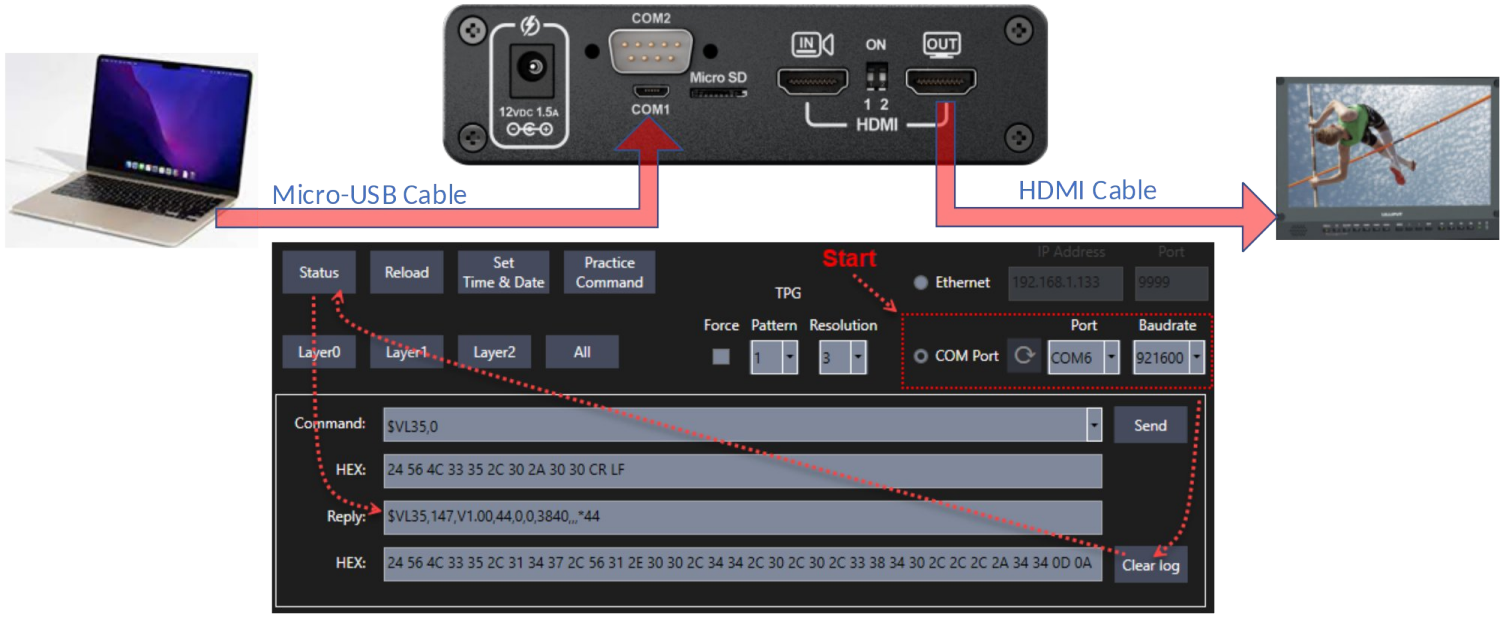
TUTORIAL

Follow the eight steps below to **Configure** Proteus or **Prepare** it for the tutorials:

1. Connect **micro-USB*** cable from PC to Proteus **COM1** (micro-USB on the rear panel). *Do not connect to mini-USB on the front panel*
2. Connect Proteus HDMI Output to a monitor capable of displaying 2K or 4K resolution (Preferably 4K)
3. Apply power to Proteus & monitor. The video should appear on your monitor within 8 seconds
4. Launch *ProteusApp*
5. Select COMx (associated with micro-USB cable) and set the baud rate to 921600
6. Press **Clear Log** and **Status** buttons to confirm communication link is working. You should see $VL35,147,V1.00,.. in the **Reply** box
7. If your monitor can *only* display 2K, press the **Config** button and set **Video Pattern Res** to 1080p. Press **Ok** to save & exit
8. Press **Sync** button. When prompted, select any Wizard file i.e. **Videologix\Tutorial\ Wizard RTC.** Monitor shall now display selected OSD

Launch *ProteusWizard* to proceed with the tutorials.

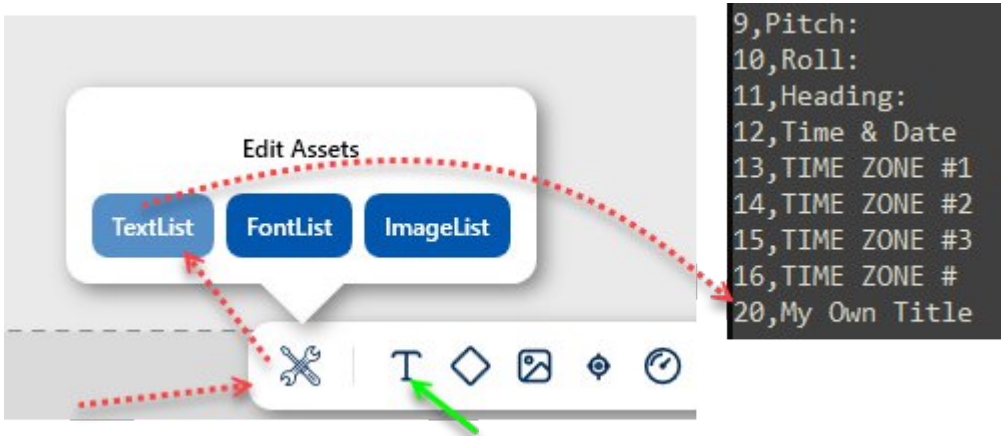
Figure 1



*Proteus-UHD uses the ***FT234XD*** chip for its micro-USB interface.*
The necessary driver should install automatically via Windows Update when the system is connected to the internet.
If the driver is not installed automatically, you can download and install it manually from the FTDI website: https://ftdichip.com/drivers/

DISPLAY TEXT, TIME & DATE

Step	On App	Description
1	ProteusWizard	Open file **Videologix\Tutorial\Wizard RTC**
2	ProteusWizard	Navigate Canvas and become familiar with the design
3	ProteusWizard	If you need to display your own Text, i.e. "**My Own Title**" a. Press Tool icon and select **TextList** as shown below b. Add line "**20 – My Own Title**" to the file as shown below. The 20 is an arbitrary ID assigned to this text c. Save **TextList** file and exit. d. To add this title to the Canvas, press the **Text** icon shown below with a green arrow. e. In the **Text Property** pane, select **20 – My Own Title** as **Text ID** f. Proceed to select your desire font, color, and position **Save** Canvas.
4	ProteusApp	Press **Sync** to update Proteus. When prompted, select file **Videologix\Tutorial\Wizard RTC**
5	ProteusApp	Title, Time & Date should now be displayed
6	ProteusApp	Press **Set Time & Date** button to adjust the clock
7	-	See Appendix C to learn how to edit Text ID 1 through 9 on-the-fly



DISPLAY IMAGES

In this exercise, we will display several PNG images.

Step	App	Description
1	ProteusWizard	Open file **Videologix\Tutorial\Wizard SpaceX**
2	ProteusWizard	Navigate Canvas and become familiar with the design
3	ProteusApp	Press **Sync** to update Proteus. When prompted, select file **Videologix\Tutorial\Wizard SpaceX**
4	-	Two SpaceX images should appear on your monitor
5	ProteusWizard	Click on the **Image** icon shown below with a green arrow. In **Image Property** pane, select "**65: Compass**" as **Image ID**
6	ProteusApp	Press **Sync** button to update Proteus. When prompted, select file **Wizard SpaceX**. Compass image should appear now.
7	ProteusWizard	To display your own image, i.e. **myImage.PNG** g. Place your image into the folder **Videologix\Images** h. Press Tool icon and select **ImageList** as shown below i. Add line "**76 – myImage**" to the file as shown below. The 76 is an arbitrary ID assigned to this image j. Save **ImageList** file and exit. k. To add the new image to the Canvas, press the **Image** icon shown below with a green arrow. l. In the **Image Property** pane, select **76 – myImage** as **Image ID** m. **Save** Canvas.
8	ProteusApp	Press **Sync** button to update Proteus. When prompted, select file **Wizard SpaceX.** Your new image should appear on your monitor.

DISPLAY GPS DATA

In this exercise, we will configure Proteus to display GPS data.

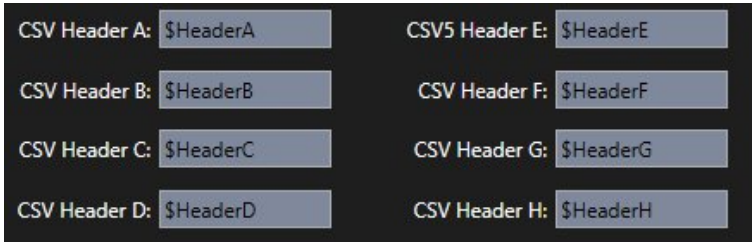
Step	On App	Description
1	ProteusWizard	If your video is 2K, open file **Videologix\Tutorial\Wizard GPS-2K**
2	ProteusWizard	If your video is 4K, open file **Videologix\Tutorial\Wizard GPS-4K**
3	ProteusWizard	Navigate Canvas and become familiar with the design
4	ProteusApp	Press **Config** and set **COM2*** to 4800 & CSV1
5	ProteusApp	Press **OK** to save & exit
6	ProteusApp	Press **Sync** to update Proteus. When prompted, select file **Videologix\Tutorial\Wizard CSV-2K/4K** based on your monitor.
7	-	Connect your GPS modem to **COM2*** located in the rear panel
8	-	GPS data fields should be updated every second
9	ProteusApp	If you don't have a GPS modem, press **Select Script** button and open file **Script GPS-RMC**
10	ProteusApp	Enter 1000 in both **EOL Delay** and **EOS Delay** fields
11	ProteusApp	Press **Start**
12	-	GPS data fields should be updated every second
13	ProteusApp	Press **Stop** to terminate simulation

**Alternatively, you can connect your GPS modem to COM4-COM6 on Expansion Port or COM3 (internal)*

DISPLAY DATA FROM CSV STRING

In this exercise, we will configure Proteus to display data from a custom CSV string.

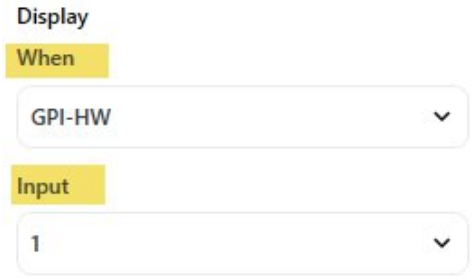
Step	App	Description
1	ProteusWizard	If your video is 2K, open file **Videologix\Tutorial\Wizard CSV-2K**
2	ProteusWizard	If your video is 4K, open file **Videologix\Tutorial\Wizard CSV-4K**
3	ProteusWizard	Navigate Canvas and become familiar with the design
4	ProteusApp	Press **Config** and enter **$HeaderA** through **$HeaderH** in the fields **CVS Header A** through **CVS Header H** as shown below
5	ProteusApp	Press **OK** to save & exit
6	ProteusApp	Press **Sync** to update Proteus. When prompted, select file **Videologix\Tutorial\Wizard CSV-2K/4K** based on your monitor.
7	-	CSV data should appear on your monitor
8	ProteusApp	Press **Select Script** button and open file **Videologix\Tutorial\Script CSV.txt**
9	ProteusApp	Enter 500 in **EOL Delay** and 0 **EOS Delay**
10	ProteusApp	Press **Start**
12	-	CSV fields should be sequentially updated every 500 msec
11	-	Press ESC on the keyboard attached to your Proteus to reload the OSD screen
13	ProteusApp	Enter 5 in **EOL Delay** and 0 **EOS Delay**
14	ProteusApp	Press **Start**
15	-	All CSV fields should be updated nearly at once
16	ProteusApp	Alternatively, you can update any individual CSV field, such as **CSV:B:1**, by copying and pasting any of the three example commands below into the **Command** box and clicking the **Send** button: $HeaderB,Hello World $VL43,CSV:B:1,Hello World $VL43,17,Hello World *(See Appendix B for the derivation of value 17)* Consecutive CSV fields can be updated in a single command by separating their values with commas. The example below shows how to update fields **CSV:B:1 through CSV:B:7** in a single command: $HeaderB,Hello World,NextValue2,NextValue3,NextValue4,NextValue5,NextValue6,NextValue7 $VL43,CSV:B:1,Hello World,NextValue2,NextValue3,NextValue4,NextValue5,NextValue6,NextValue7 $VL43,17,Hello World,NextValue2,NextValue3,NextValue4,NextValue5,NextValue6,NextValue7



TOGGLE THE OVERLAY ON AND OFF VIA HARDWARE

The overlays can be toggled on or off using one of two methods: either through **software control** or via a **hardware enable signal.**

The selection must be made during the overlay design stage via **ProteusWizard**. Select each object and use the dialog shown below to define which hardware pin will toggle the overlay object on and off. The dialog below gives control to hardware GPI input #1 (pin #7) on the *Expansion port*.

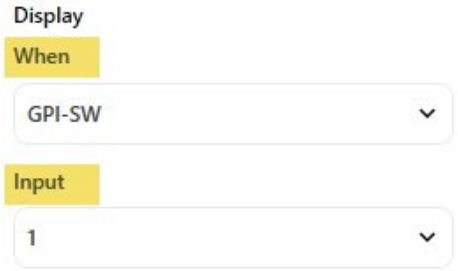


Connect HW-GPI:**1** (pin #7) to ground (pin #5) to turn overlay object OFF and disconnect from ground to turn overlay object ON.

TOGGLE THE OVERLAY ON AND OFF VIA SOFTWARE

The overlays can be toggled on or off using one of two methods: either through **software control** or via a **hardware enable signal.**

The selection must be made during the overlay design stage via **ProteusWizard**. Select each object and use the dialog below to define which software pin will toggle the overlay object on and off. The dialog below gives control to software GPI input #1.



As shown in the table below, set software GPI input #1 to logic **1** to display the overlay object, and set it to logic **0** to hide the overlay object:

To turn overlay object...	Send command		Alternative method using ProteuApp
ON	$VL43,SYS:GPI-SW:1,1	Or $VL43,191,1	Toggle **GPI-SW** DIP switch 1 to UP position
OFF	$VL43,SYS:GPI-SW:1,0	Or $VL43,191,0	Toggle **GPI-SW** DIP switch 1 to DOWN position

See Appendix B for the derivation of value 191.

DISPLAY SOFTWARE CONTROLLED COUNT UP TIMER

In this exercise, we will display a count up timer that can be controlled by software GPI.

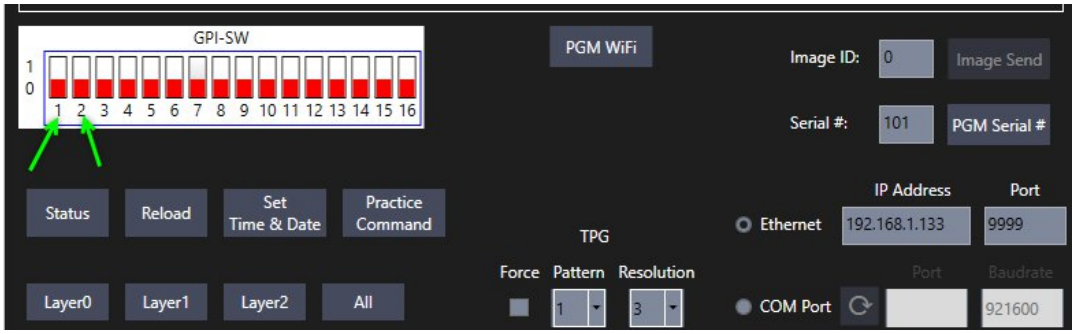
Step	App	Description
1	ProteusWizard	Open file **Videologix\Tutorial\Wizard TIMERUP-SW**
2	ProteusWizard	Navigate Canvas and become familiar with the design
3	ProteusApp	Press **Sync** to update Proteus. When prompted, select file **Videologix\Tutorial\Wizard TIMERUP-SW**
4	ProteusApp	To start the timer, set **GPI-SW:1** to **1** as shown below with green arrow or send command $VL43,SYS:GPI-SW:1,1
5	ProteusApp	To pause the timer, set **GPI-SW:1** to **0**. Alternatively send command $VL43,SYS:GPI-SW:1,0 or $VL43,191,0
6	ProteusApp	To reset the timer, set **GPI-SW:2** to **1**. Alternatively send command $VL43,SYS:GPI-SW:2,1 or $VL43,192,1

See Appendix B for the derivation of values 191 & 192.

DISPLAY SOFTWARE CONTROLLED COUNT DOWN TIMER

In this exercise, we will display a countdown timer that can be controlled by software GPI.

Step	App	Description
1	ProteusWizard	Open file **Videologix\Tutorial\Wizard TIMERDN-SW**
2	ProteusWizard	Navigate Canvas and become familiar with the design
3	ProteusApp	Press **Sync** to update Proteus. When prompted, select file **Videologix\Tutorial\Wizard TIMERDN-SW**
4	ProteusApp	To start the timer, set **GPI-SW:1** to **1** as shown below with green arrow or send command $VL43,SYS:GPI-SW:1,1
5	ProteusApp	To pause the timer, set **GPI-SW:1** to **0** as shown below. Alternatively send command $VL43,SYS:GPI-SW:1,0 or $VL43,191,0
6	ProteusApp	To reset the timer, set **GPI-SW:2** to **1** as shown below. Alternatively send command $VL43,SYS:GPI-SW:2,1 or $VL43,192,1



DISPLAY HARDWARE CONTROLLED COUNT UP TIMER

In this exercise, we will display a count up timer that can be controlled by hardware GPI.
Note, with no signal applied to the GPI-HW pins, their default state is 5V. Therefore, connect GPI-HW:1 and GPI:HW:2 to ground prior to running this demo.

Step	App	Description
1	ProteusWizard	Open file **Videologix\Tutorial\Wizard TIMERUP-H**
2	ProteusWizard	Navigate Canvas and become familiar with the design
3	ProteusApp	Press **Sync** to update Proteus. When prompted, select file **Videologix\Tutorial\Wizard TIMERUP-HW**
4	ProteusApp	To start the timer, connect **GPI-HW:1** to 5V or float as shown below with green arrow
5	ProteusApp	To pause the timer, connect **GPI-HW:1** to ground
6	ProteusApp	To reset the timer, connect **GPI-HW:2** to 5V or float

DISPLAY HARDWARE CONTROLLED COUNT DOWN TIMER

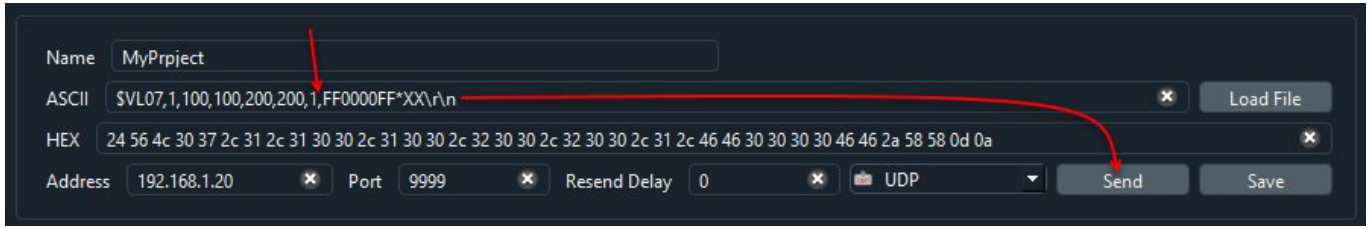
In this exercise, we will display a countdown timer that can be controlled by hardware GPI.
Note, with no signal applied to the GPI-HW pins, their default state is 5V. Therefore, connect GPI-HW:1 and GPI:HW:2 to ground prior to running this demo.

Step	App	Description
1	ProteusWizard	Open file **Videologix\Tutorial\Wizard TIMERDN-HW**
2	ProteusWizard	Navigate Canvas and become familiar with the design
3	ProteusApp	Press **Sync** to update Proteus. When prompted, select file **Videologix\Tutorial\Wizard TIMERDN-HW**
4	ProteusApp	To start the timer, connect **GPI-HW:1** to 5V or float as shown below with green arrow
5	ProteusApp	To pause the timer, connect **GPI-HW:1** to ground
6	ProteusApp	To reset the timer, connect **GPI-HW:2** to 5V or float

SEND COMMAND VIA ETHERNET USING PROTEUSAPP

Step	App	Description
1	-	Configure your Proteus as described in *Configure proteus-uhd*
2	ProteusApp	Press **Config** and set **IP Mode**, **IP Address**, **IP Subnet** to your desired values. Press OK to save and exit
3	ProteusApp	Select **Sync** to transmit new Config setting to Protues
4	ProteusApp	Select **Ethernet** and enter Proteus **IP address**. Enter **9999** for **Port**
5	ProteusApp	Press **Select Script** button and open file **Videologix\Tutorial\Script Demo4K**
6	ProteusApp	Enter 5 in **EOL Delay** field and 0 in **EOS Delay** field
7	ProteusApp	Press **Start** , **All**, **Start** buttons in sequence and demo should appear on the screen.

SEND COMMAND VIA ETHERNET USING PACKET SENDER APP

Step	App	Description
1	-	Configure your Proteus as described in *Configure proteus-uhd*
2	ProteusApp	Press **Config** and set **IP Mode**, **IP Address**, **IP Subnet** to your desired values. Press OK to save and exit
3	ProteusApp	Select **Sync** to transmit new Config setting to Protues
4	Packet Sender	Free utility *Packet Sender* can be used to send commands to Proteus. Using the figure below as a reference, configure Packet Sender with the IP address assigned in Step 2 above.  Copy & paste command $VL07,1,100,100,200,200,1,FF0000FF*XX\n into **ASCII** box and click **Send** as shown below:  A red rectangular box should appear on the upper left corner of your monitor attached to Proteus.

DISPLAY ROV SITUATION AWARENESS

In this exercise, we will display a ROV widget and link it to Proteus built-in tilt sensor and e-compass respectively.

Step	App	Description
1	ProteusWizard	If your video is 2K, open file **Videologix\Tutorial\Wizard ROV-2K**
2	ProteusWizard	If your video is 4K, open file **Videologix\Tutorial\Wizard ROV-4K**
3	ProteusWizard	Navigate Canvas and become familiar with the design
4	ProteusApp	Press **Sync** to update Proteus. When prompted, select file **Videologix\Tutorial\Wizard ROV-2K/4K** based on your monitor.
5	ProteusApp	Press **Select Script** button and open file **Script GPS-RMC**
6	ProteusApp	Enter 1000 in both **EOL Delay** and 0 in **EOS Delay** fields
7	ProteusApp	Press **Start**. GPS data fields should be updated every second
8	ProteusApp	Grap Proteus unit and move it around. ROV widget should reflect tilt sensor position and e-compass
9	ProteusApp	Press **Stop** to terminate simulation
10	ProteusApp	Press **Select Script** button and open file **Script ROV-2K/4K** based on your monitor
11	ProteusApp	Enter 100 in **EOL Delay** and 0 in **EOS Delay** fields
12	ProteusApp	Press **Start**. Slider & gauge should start animating
13	ProteusApp	Press **Stop** to terminate simulation

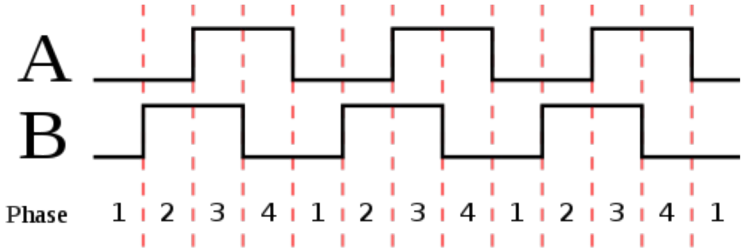
DISPLAY SPEEDOMETER GAUGE

A typical speedometer consists of a needle positioned at the center of the gauge. In this exercise, we will display a speedometer gauge and link it to 1st value of the CSV string A (CSV:A:1)

Step	App	Description
1	ProteusWizard	Open file **Videologix\Tutorial\Wizard Guage**
2	ProteusWizard	Navigate Canvas and become familiar with the design
3	ProteusApp	Press **Sync** to update Proteus. When prompted, select file **Videologix\Tutorial\Wizard Guage**
4	ProteusApp	Press **Select Script** button and open file **Videologix\Tutorial\Script Gauge**
5	ProteusApp	Enter 50 in **EOL Delay** and 0 **EOS Delay**
6	ProteusApp	Press **Start.** Gauge should be animated.

DISPLAY QUADRATURE COUNTERS

- Two Quadrature counters. Compatible with Mechanical, Hall effect & Optical rotary encoders. See expansion port for pin assignment.
- The counters are 32-bits wide.
- Configurable line resolution $x1$, $x2$, $x4$. Resolution can be defined in ProteusApp
- RESET pin
- Raw counter can be converted to any unit (distance, speed, etc.) using $map = m * raw + b$
- Slope and intercept for each channel can be set by pressing the **Config** button in ProteusApp.
- Slope & intercept can also be set in real-time using command $VL27. See "PROTEUS-UHD SCS.pdf" for more detail.



X1:	1	1	1	1	2	2	2	2	3	3	3	3	4
X2:	1	1	2	2	3	3	4	4	5	5	6	6	7
X4:	1	2	3	4	5	6	7	8	9	10	11	12	13

Refer to the following registers to display quadrature counter values:

Name	ID
SYS:QUAD:MAP1	172
SYS:QUAD:MAP2	173
SYS:QUAD:RAW1	174
SYS:QUAD:RAW2	175

DISPLAY ANALOG INPUTS

- Four Analog inputs. See expansion port for pin assignment.
- Input range 0..3.30V
- Internal low pass RC filter (24Ω, 5600pF)
- 12-Bit ADC. Analog signals are **Sampled** at 1KHz. Each ADC **Sample** is an average of 4 consecutive (2µs apart) samples
- The **raw** values are average of 64 most recent **Samples**.
- The **raw** values can be converted to any unit (distance, speed, etc.) using **map** $= m * raw + b$
- Slope and intercept for each channel can be set by pressing the **Config** button in ProteusApp

Refer to the following registers to display the Analog values:

Name	ID
SYS:ANLG:MAP1	164
SYS:ANLG:MAP2	165
SYS:ANLG:MAP3	166
SYS:ANLG:MAP4	167
SYS:ANLG:RAW1	168
SYS:ANLG:RAW2	169
SYS:ANLG:RAW3	170
SYS:ANLG:RAW4	171

REALTIME TEXT EDITING

Text IDs 1 through 10 (if displayed) can be edited in real time using **Alt + 1** through **Alt + 10**. Revised text is automatically saved to **TextList.txt**.

WATCHDOG TIMER

The watchdog timer automatically clears the video overlay data if all communications stop. Use **ProteusApp** (Config button) to set the watchdog timer value in seconds. For example, if the value is set to **5 seconds**, the screen will automatically clear **5 seconds** after all communication stops. A watchdog value of **0** disables the timer.

APPENDIX A – UPDATING FIRMWARE

Firmware updates may involve two separate files:

- **UHD-Vx.xx.srec** — Main application firmware (where *x.xx* refers to the version number)
- **BOOT.bin** — Bootloader file

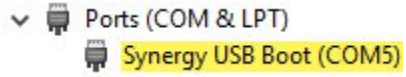
After completing both firmware updates, press Alt + h. The last line on the attached display should read Version = [V1.xx] [V1.yy]

xx corresponds to the .srec firmware version
yy corresponds to the BOOT.bin firmware version

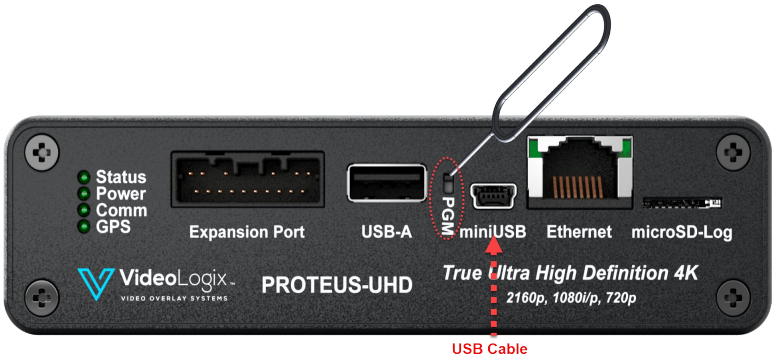
UPDATING UHD-VX.XX.SREC

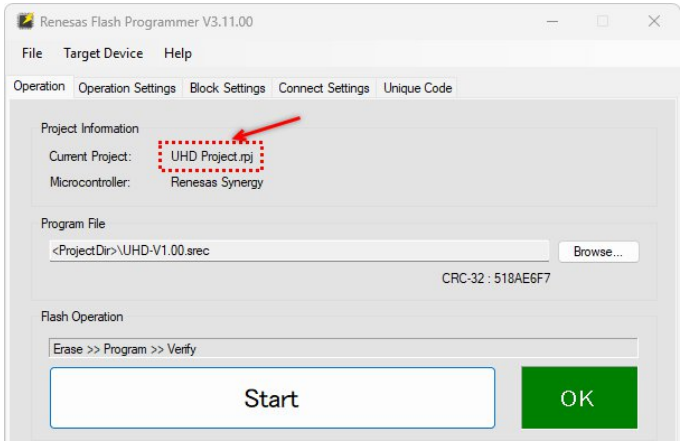
This section assumes you have already installed **Renesas Flash Programmer** described in Install Renesas Flash Programmer.

1. Use the ejector pin and set **PGM** switch (located on the front panel) to the lower position
2. Cycle power to PROTEUS-UHD
3. Connect the USB cable from your PC to the **miniUSB** port on the front panel
4. Your PC should acknowledge PROTEUS-UHD with a beep. Alternatively, the Device Manager will add the following:

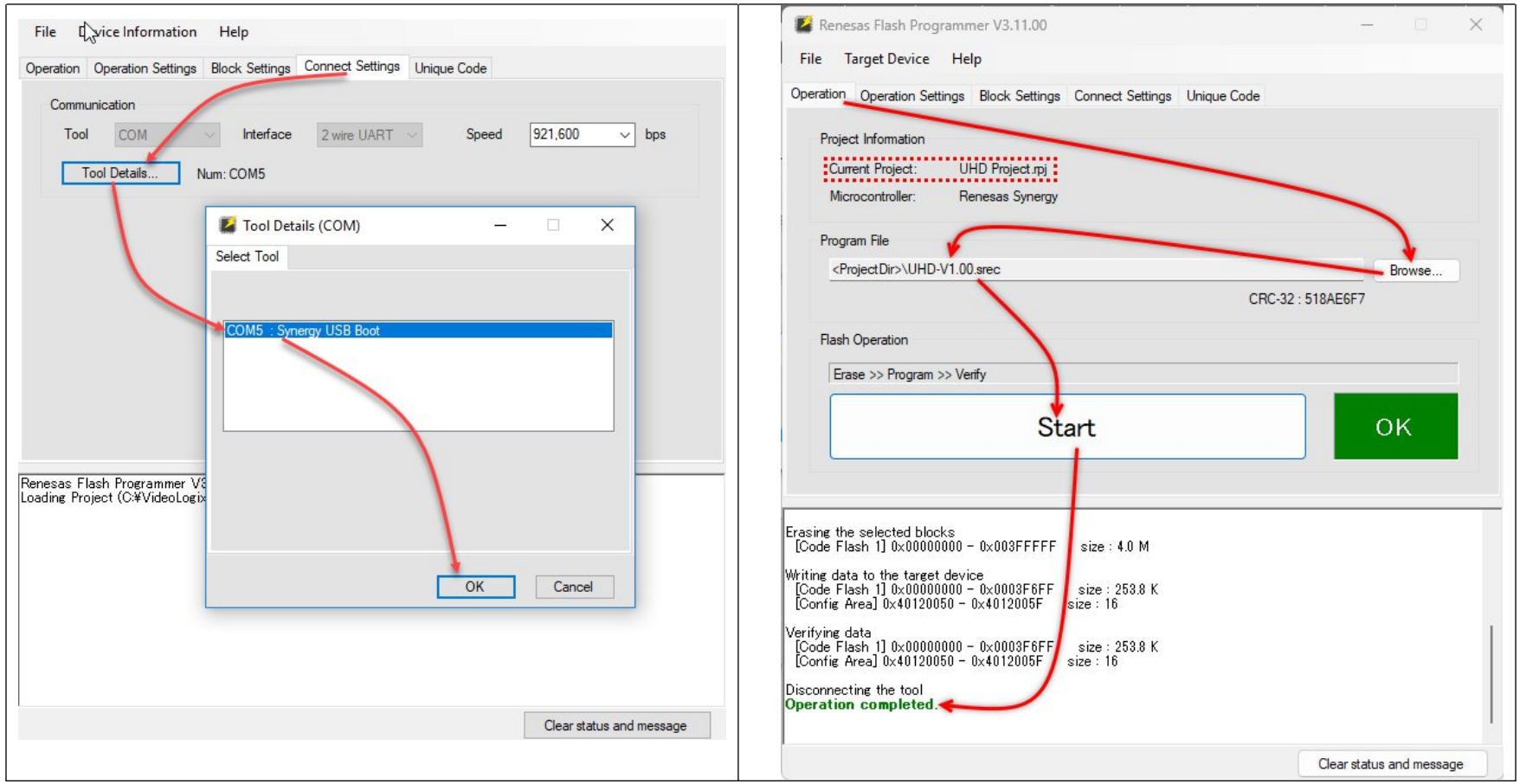


5. Start **Renesas Flash Programmer**. As shown below, ensure **Current Project** is **UHD Project.rpj**. If confirmed, go to step 6.
   If not, go to File, Open Project and load it from folder ***Documents\Videologix\Utilities\UHD Project.rpj***
   To avoid repeating this step in the future, go to File and Save Project.





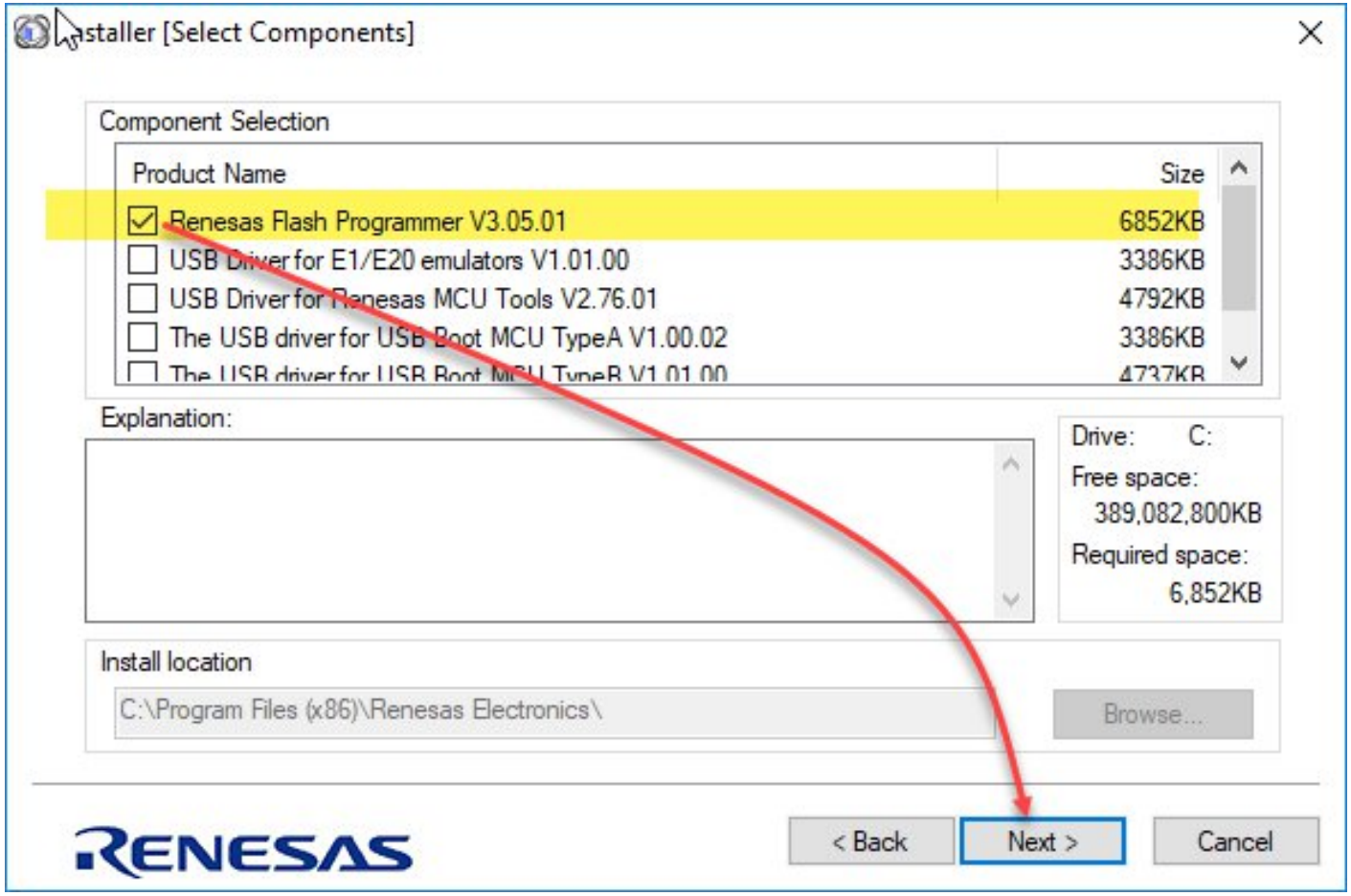
6. Follow the instructions below to load the firmware into Proteus.



7. After '**Operation is completed**', use the ejector pin and set **PGM** switch (located on the front panel) to the upper position
8. Cycle power to PROTEUS-UHD.

INSTALL RENESAS FLASH PROGRAMMER

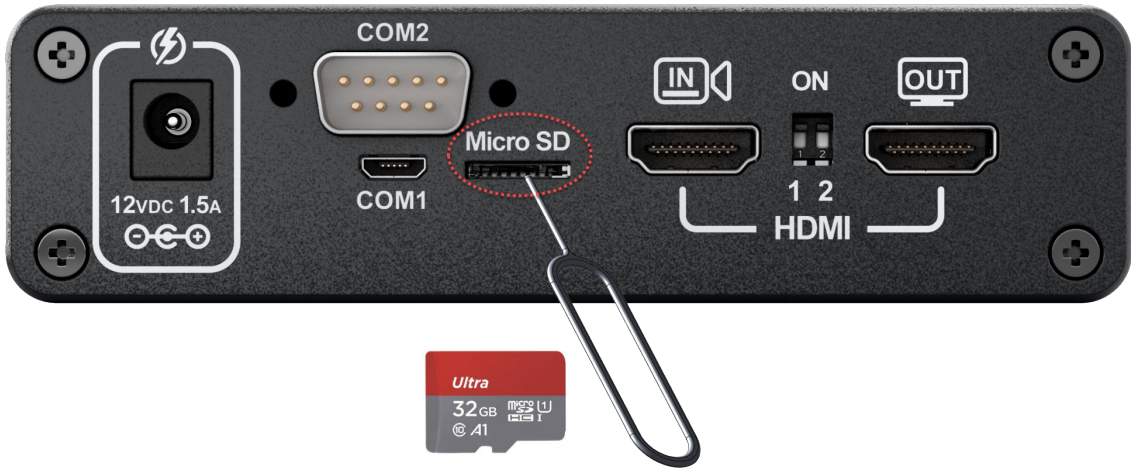
Go to folder **Documents\Videologix\Utilities** and launch program **Renesas_Flash_Programmer_Package_V30501**
Follow the instructions below:



UPDATING BOOT.BIN

To update the **BOOT.bin** file on the Proteus-UHD:

1. Use the ejector pin to remove the micro-SD card located in the rear panel.
2. Insert the micro-SD card into your PC using a micro-SD card reader.
3. Copy the new **BOOT.bin** file onto the root directory of the micro-SD card.
4. Safely eject the micro-SD card from your computer and re-insert it back into the rear panel of the Proteus-UHD.
5. Apply power to the Proteus-UHD to complete the update



APPENDIX B – 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		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		

APPENDIX C - KEYBOARD COMMANDS

Keyboard command	Description
Alt + 1 or Ctrl + 1	Edit user text ID#1
Alt + 2 or Ctrl + 2	Edit user text ID#2
Alt + 3 or Ctrl + 3	Edit user text ID#3
Alt + 4 or Ctrl + 4	Edit user text ID#4
Alt + 5 or Ctrl + 5	Edit user text ID#5
Alt + 6 or Ctrl + 6	Edit user text ID#6
Alt + 7 or Ctrl + 7	Edit user text ID#7
Alt + 8 or Ctrl + 8	Edit user text ID#8
Alt + 9 or Ctrl + 9	Edit user text ID#9
ESC	Refresh screen
Alt + h or Ctrl + h	Display system configuration
Alt + s or Ctrl + s	Start displaying sensor data stream arriving on COM2, COM3, COM4, COM5, mini-USB, Ethernet
Alt + t or Ctrl + t	Stop displaying sensor data stream arriving on COM2, COM3, COM4, COM5, mini-USB, Ethernet
Alt + Ctrl + Shift + I	Display Network parameters (MAC address, IP address, Subnet, Gateway)