PROTEUS-UHD

An Ultra High-Definition Video Overlay

User Manual (UM)

Version V1.00
June 11, 2025

TABLE OF CONTENTS

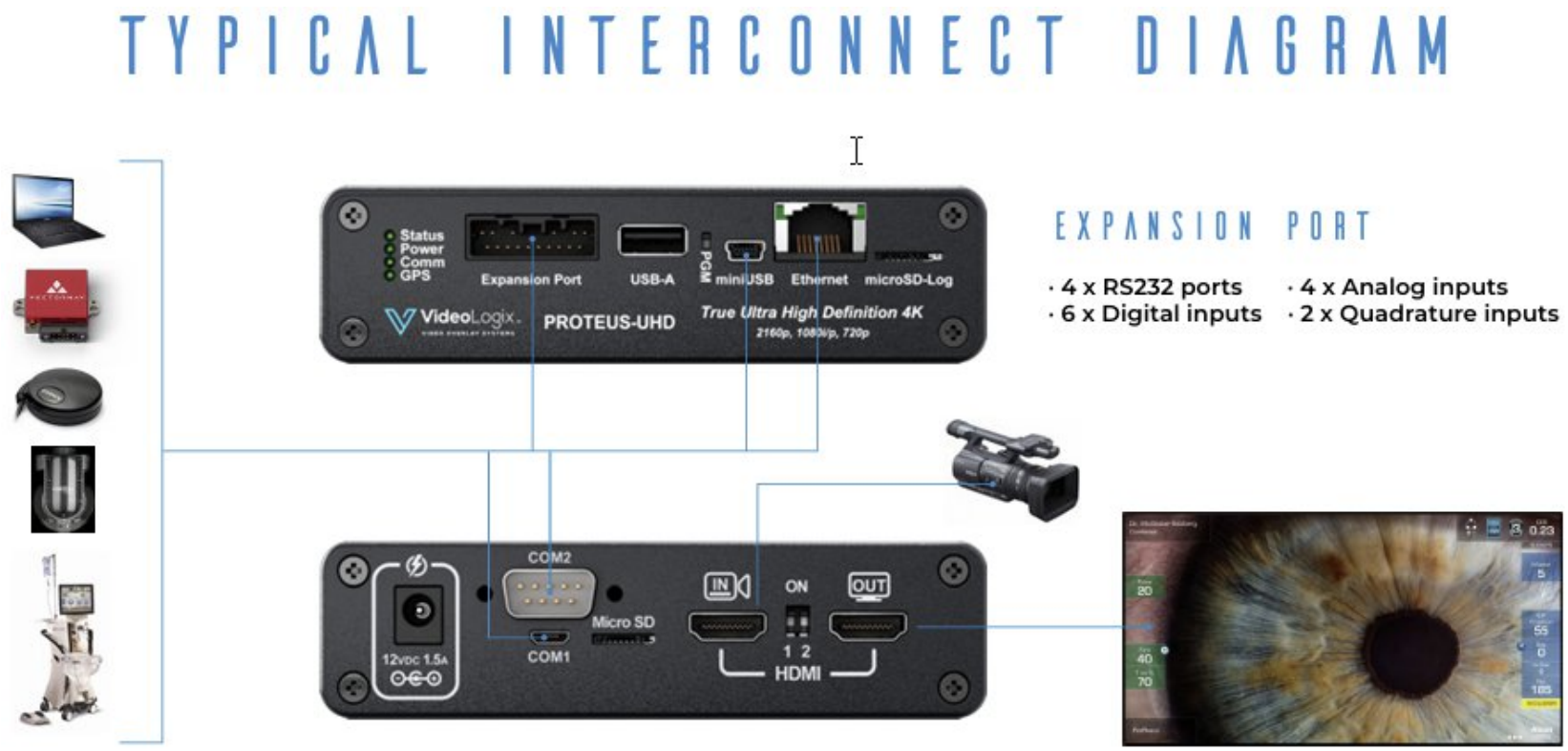
GENERAL OVERVIEW

Video Overlay is a method by which computer-generated images are superimposed on video. Properly transformed images appear as if they are an integral part of the scene without impeding the video of the actual environment.

PROTEUS-UHD is designed for overlaying texts, images and data onto 2K & 4K video. This is useful for broadcasting, surveillance, mapping applications, or real-time monitoring where such overlays provide critical contextual information.

PROTEUS-UHD offers advanced functionality through additional widgets, but its core capability remains straightforward—handling text, image, and GPS data insertion efficiently.

PROTEUS-UHD's support for HDMI 2.1 ensures high-resolution, high-bandwidth video compatibility, making it ideal for modern displays and broadcast systems. Additionally, the fact that it operates independently of a computer simplifies deployment, making it more reliable for stand-alone applications.

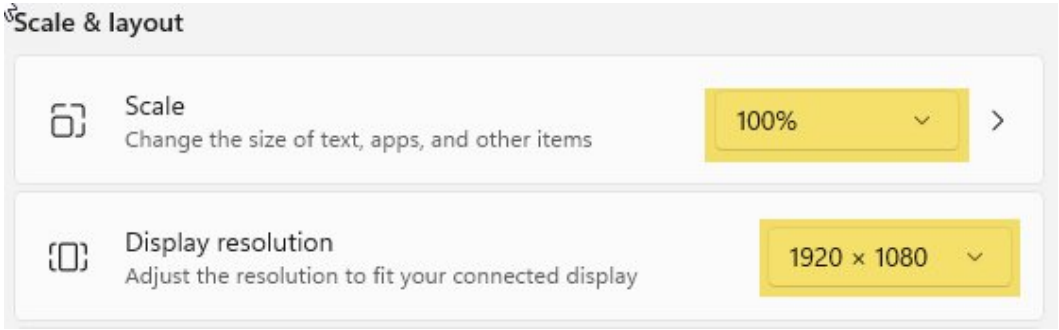


GLOSSARY TERMS

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 includes a micro-SD card) into your PC
2. This micro-SD has a folder called ***Videologix***. Copy this folder into your ***Documents*** folder
3. Confirm folder *Documents**Videologix*** now exists on your PC
4. For ***ProteusWizard*** App to work as WYSIWYG, it is important to set your PC display scale to 100% and resolution to 1920 x 1080 as shown:



For proper operation, the folder **Videologix must remain inside the **Documents** directory. Do not rename or modify its location.*

EXTERNAL INTERFACES

COMMUNICATION

COM PORTS

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

COM PORT	Location		*Expansion port*	Intended to Interface with	Baud rate
COM1	Micro-USB	Rear Panel micro-USB	-	**ProteusApp to configure Proteus**	**921600, N,8,1**
COM2	RS232	Rear Panel DB9	2=RX, 3=TX, 5=GND	Various sensors	4800-921600, N,8,1
COM3	RS232	Internal TB (J12)	1=RX, 3=TX, 2=GND	Various sensors	4800-921600, N,8,1
COM4	RS232	Expansion Port	1=RX, 2=TX, 5=GND	Various sensors	4800-921600, N,8,1
COM5	RS232	Expansion Port	3=RX, 4=TX, 5=GND	Various sensors	4800-921600, N,8,1

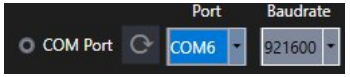
When using a USB serial adaptor to communicate with COM1-5 at 460,800 or 921,600, ensure it is capable of 921.6 Kbps

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



- Connect your PC to *micro-USB* port on the rear panel. Launch *Videologix**ProteusApp** and select *COM Port* as shown:
- Confirm the communication link by pressing the '*Clear Log*' button, then the '*Status*' button. The *Reply* should start with $VL35,,,,,,
- 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 your own Wizard file or choose a sample Wizard file from *Videologix\Tutorial*
- Your monitor should now display the selected Wizard OSD.
- To learn more about Proteus, go through the entire *Tutorial* section.

*Pressing **Alt + h** (On the keyboard attached to PROTEUS) will display all the UHD configurations on your monitor.*

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:

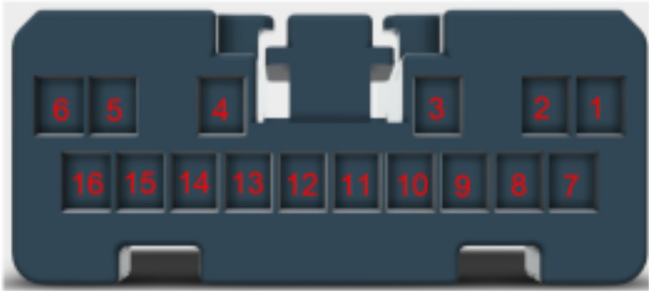
2160p @	23.97 / 24 / 25 / 29.97 / 30 / 50 / 59.94 / 60 HZ
1080p @	23.97 / 24 / 25/ 29.97/ 30 Hz / 60 HZ
1080i @	50 / 60 Hz
720p @	50 / 59.94 / 60 Hz

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	Description
1	COM4 - RX	COM4 UART
2	COM4 - TX	
3	COM5 - RX	COM5 UART
4	COM5 - TX	
5	GND	Ground
6	V5 (optional)	If 5V is supplied for the optical isolators, set internal jumper J14 between pins 1 and 2
7	QUADRATURE INPUT A1	Optically isolated. Alternatively, it can be used as GPI-HW:1 to control timer or turn a widget ON/OFF*
8	QUADRATURE INPUT B1	Optically isolated. Alternatively, it can be used as GPI-HW:2 to control timer or turn a widget ON/OFF*
9	QUADRATURE RESET	Optically isolated. Alternatively, it can be used as GPI-HW:3 to control timer or turn a widget ON/OFF*
10	QUADRATURE INPUT A2	Optically isolated. Alternatively, it can be used as GPI-HW:4 to control timer or turn a widget ON/OFF*
12	QUADRATURE INPUT B2	Optically isolated. Alternatively, it can be used as GPI-HW:5 to control timer or turn a widget ON/OFF*
12	ANALOG INPUT1	Range 0-3.3V
13	ANALOG INPUT2	Range 0-3.3V
14	ANALOG INPUT3	Range 0-3.3V
15	ANALOG INPUT4	Range 0-3.3V
16	IRIG-B INPUT	Can be used as IRIG-B, TLC or general-purpose 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 see 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 set 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**

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 $VL42,118,**123**

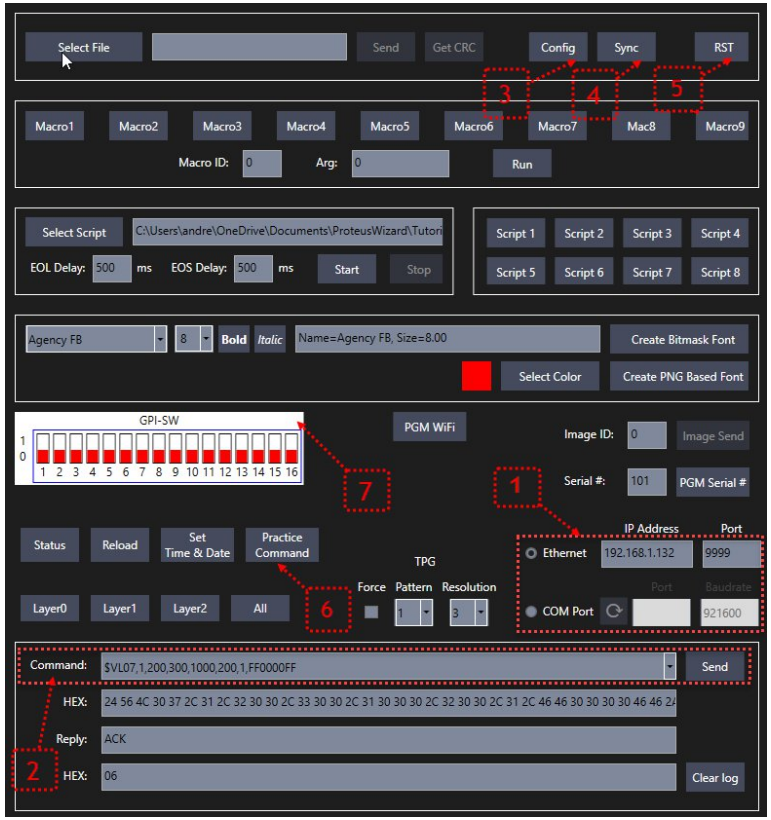
PROTEUSAPP

PURPOSE

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

To **Configure** Proteus and/or **Download** OSD, you must connect to COM1(**micro-USB)** port located on the rear panel. Baud rate for this COM port is 921600.

MAIN FEATURES



Feature	Description
1	When **configuring** Proteus, select the COM1 (micro-USB) port located on the rear panel. This port is always fixed at 921600. Upon selecting the 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 COM port i.e. COM1 (micro-USB), COM2 (DB9) ,COM3 (Internal) ,COM4 & COM5 (expansion port). Baud rate for COM2,3,4,5 is configurable.
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*

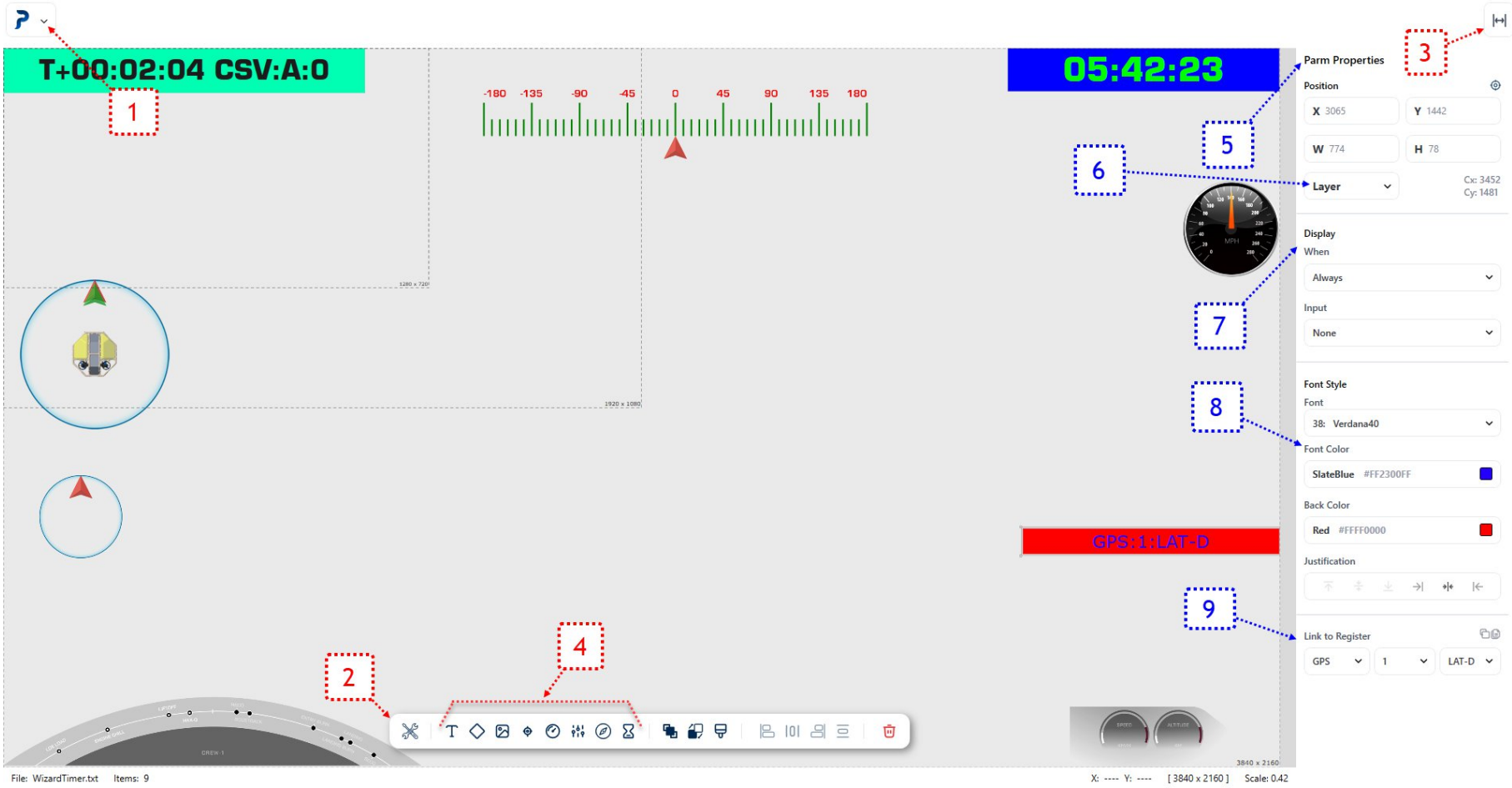
See section ProteusApp in the SCS document to learn how to send commands to PROTEUS-UHD

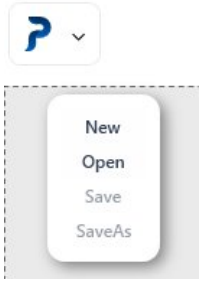
PROTEUSWIZARD

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

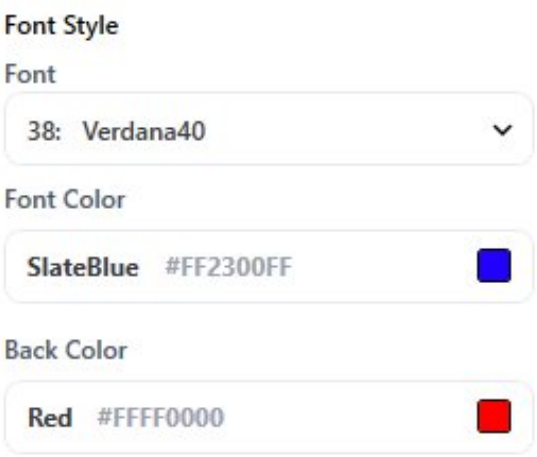
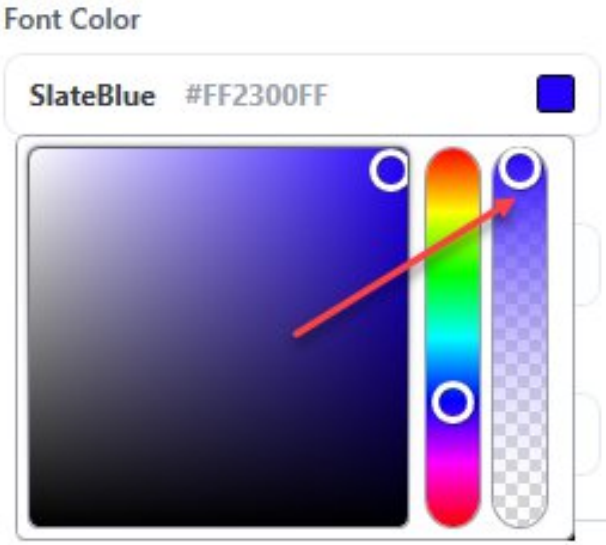


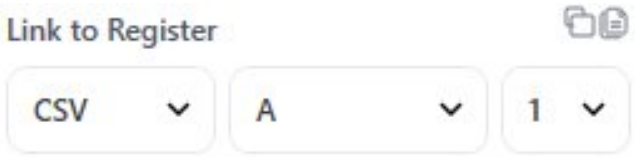
Feature	Description
1	File operation i.e. open a **Wizard** design, save current design, etc.  Naming files associated with **ProteusWizard** with the prefix **"Wizard"** provides several benefits: 1. **Consistency** – Ensures all related files are easily identifiable and grouped together. 2. **Avoids 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** This functionality ensures that your OSD fits perfectly within your video input.
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.

<table>
<tr><td>5</td><td>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]</td></tr>
<tr><td>6</td><td> 

Select a layer to display the widget. Widgets that overlap must be assigned to different layers to ensure proper rendering</td></tr>
<tr><td>7</td><td> 

This dialog allows you to specify the conditions under which a widget will be displayed.<table><tr><th>When</th><th>Description</th></tr><tr><td>Always</td><td>Always display the widget</td></tr><tr><td>GPI-HW</td><td>Display widget only if GPI-HW {1,2,3,4,5} is 1</td></tr><tr><td>GPI-SW</td><td>Display widget only if GPI-SW {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16} is 1</td></tr><tr><td>1HZ or 2HZ</td><td>Flash widget @ 1HZ or 2HZ</td></tr><tr><td>Never</td><td>Never display the widget</td></tr></table>
See SCS for how to set or reset GPI-SW</td></tr>
</table>

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 Tutorial section how to use ProteusWizard to design various OSD

TUTORIAL

Follow the instructions below to prepare Proteus for running tutorials:

1. Connect a micro-USB cable from Proteus (micro-USB port on the rear panel) to your PC.
2. Connect Proteus HDMI Output to a monitor capable of displaying 2K or 4K resolution
3. Apply power to Proteus & monitor
4. Launch *ProteusApp*
5. Select COMx (associated with micro-USB cable) and set the baud rate to 921600
6. Press *Clear Log* and *Status* to confirm communication link is working. You should see $VL35,147,V1.00,49,1,0,3840,, in the *Reply* box
7. If your monitor can ***only*** display 2K, press *Config* button and set 'Video Pattern Res' to 1080p
8. Press *Ok* to save & exit
9. Press *Sync*
10. Monitor shall display a crosshatch test pattern
11. Launch *ProteusWizard* app

DISPLAY TIME, DATE & TITLES

Step	On App	Description
1	ProteusWizard	Open file *Videologix\Tutorial\Wizard RTC*
2	ProteusWizard	Navigate Canvas and become familiar with the design
3	ProteusApp	Press *Sync* to update Proteus. When prompted, select file *Wizard RTC* Time & Date should now be displayed
4	ProteusApp	Press '*Set Time & Date*' button to adjust the clock

DISPLAY GPS DATA

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

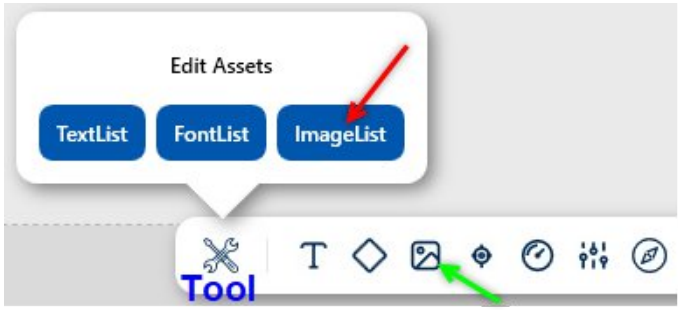
Step	On App	Description
1	ProteusWizard	If your video is 2K, open file *Videologix\Tutorial\Wizard GPS-2K*
2	ProteusWizard	If your video is 4K, open file *Videologix\Tutorial\Wizard GPS-4K*
3	ProteusWizard	Navigate Canvas and become familiar with the design
4	ProteusApp	Press *Config* and set **COM2*** to 4800 & CSV1
5	ProteusApp	Press *OK* to save & exit
6	ProteusApp	Press *Sync* to update Proteus. When prompted, select file *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

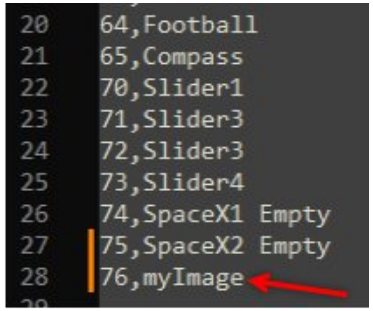
**Alternatively, you can connect your GPS modem to COM4 or COM5 on Expansion Port or COM3 (internal)*

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 *Wizard SpaceX*
4	ProteusApp	Two SpaceX images should appear on your monitor
5	ProteusWizard	Click on the Image icon shown below with a green arrow. Select image "**65 – Compass**" as Image ID
6	ProteusWizard	For this Compass, set *Display:When* = GPI-SW and *Display:Input* = 1 and save the canvas
7	ProteusApp	Press *Sync* to update Proteus. When prompted, select file *Wizard SpaceX*
8	ProteusApp	Toggle GPI-SW1 on and off several times. Each time, the Compass image should appear when switched on and disappear when switched off.
9	ProteusWizard	If you need to display your own image, i.e. **myImage.PNG:** a. Add your image to the folder *Videologix\Images* b. Press Tool icon and select *ImageList* as shown below c. Add line "**76 – myImage**" to the file as shown below. The 76 is an arbitrary ID assigned to this image d. Save file and exit. e. To add your new image to the Canvas, press the *Image* icon shown below with a green arrow. f. Set Image ID to "**76 – myImage**" g. Save Canvas.
10	ProteusApp	Press *Sync* to update Proteus. When prompted, select file *Wizard SpaceX*
-	-	Your new image should appear on your monitor

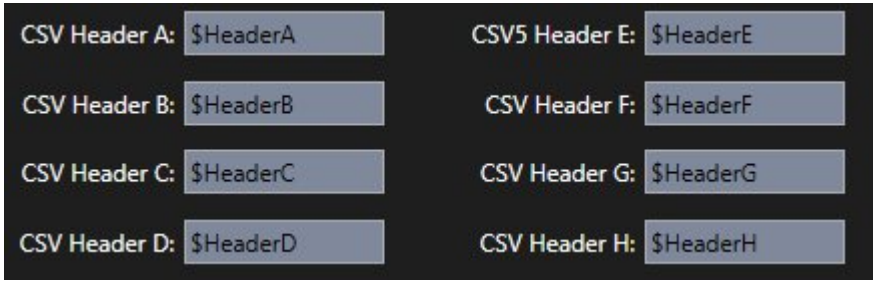




DISPLAY CSV DATA

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 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 *Wizard CSV-2K/4K* based on your monitor.
7	ProteusApp	Press *Select Script* button and open file *Videologix\Tutorial\Script CSV.txt*
8	ProteusApp	Enter 500 in *EOL Delay* and 0 *EOS Delay*
9	ProteusApp	Press *Start*
10	-	CSV fields should be sequentially updated every 500 msec
12	-	Press ESC on the keyboard attached to Proteus to reload the OSD screen
11	ProteusApp	Enter 5 in *EOL Delay* and 0 *EOS Delay*
13	ProteusApp	Press *Start*
14	-	CSV fields should be updated at once



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 *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 *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*
7	-	Gauge should be animated

DISPLAY SOFTWARE CONTROLLED COUNT UP TIMER

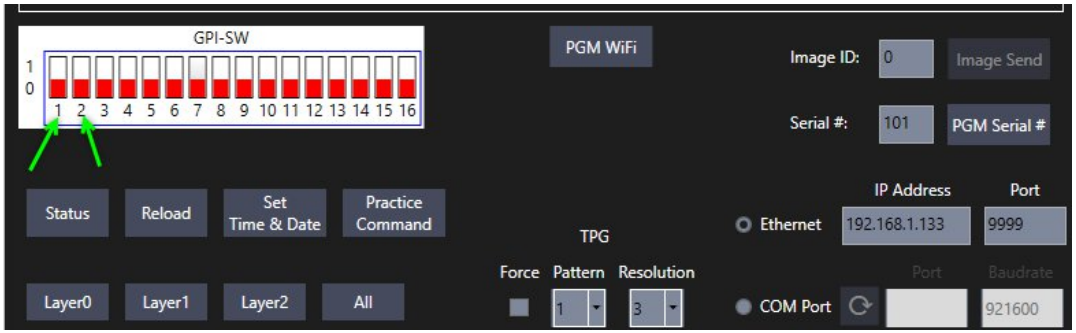
In this exercise, we will display a count up timer that can be controlled by software GPI.

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 *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** or send command $VL43,SYS:GPI-SW:1,0
6	ProteusApp	To reset the timer, set **GPI-SW:2** to **1** or send command$VL43,SYS:GPI-SW:2,1

DISPLAY SOFTWARE CONTROLLED COUNTDOWN 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 *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 or send command $VL43,SYS:GPI-SW:1,0
6	ProteusApp	To reset the timer, set **GPI-SW:2** to **1** as shown below or send command $VL43,SYS:GPI-SW:2,1



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 *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

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.

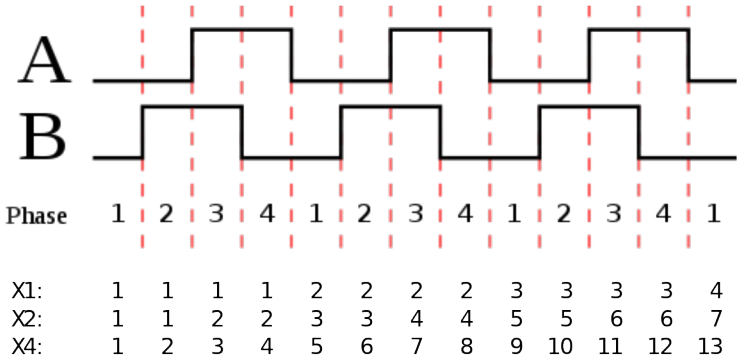
Step	App	Description
1	ProteusWizard	Open file *Videologix\Tutorial\Wizard TIMERUP-H*
2	ProteusWizard	Navigate Canvas and become familiar with the design
3	ProteusApp	Press *Sync* to update Proteus. When prompted, select file *Wizard TIMERUP-HW*
4	ProteusApp	To start the timer, connect **GPI-HW:1** to 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

SEND COMMAND VIA ETHERNET

Step	App	Description
1	ProteusApp	Configure your Proteus as described in *Configure proteus and load osd*
2	ProteusApp	Select *Ethernet* and enter Proteus *IP address.* Enter *9999* for *Port*
3	ProteusApp	Press *Select Script* button and open file *Videologix\Tutorial\Script Demo4K*
4	ProteusApp	Enter 5 in *EOL Delay* field and 0 in *EOS Delay* field
5	ProteusApp	Press *Start* , *All*, *Start* buttons in sequence and demo should appear on the screen.

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*



Refer to the following registers to display quadrature counter values:

Name	ID
SYS:QUAD:MAP1	166
SYS:QUAD:MAP2	167
SYS:QUAD:RAW1	168
SYS:QUAD:RAW2	169

DISPLAY ANALOG INPUTS

- Four Analog inputs. See expansion port for pin assignment.
- Input range 0..3.3V
- 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*

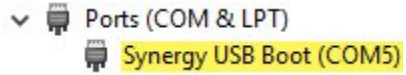
Refer to the following registers to display the Analog values:

Name	ID
SYS:ANLG:MAP1	158
SYS:ANLG:MAP2	159
SYS:ANLG:MAP3	160
SYS:ANLG:MAP4	161
SYS:ANLG:RAW1	162
SYS:ANLG:RAW2	163
SYS:ANLG:RAW3	164
SYS:ANLG:RAW4	165

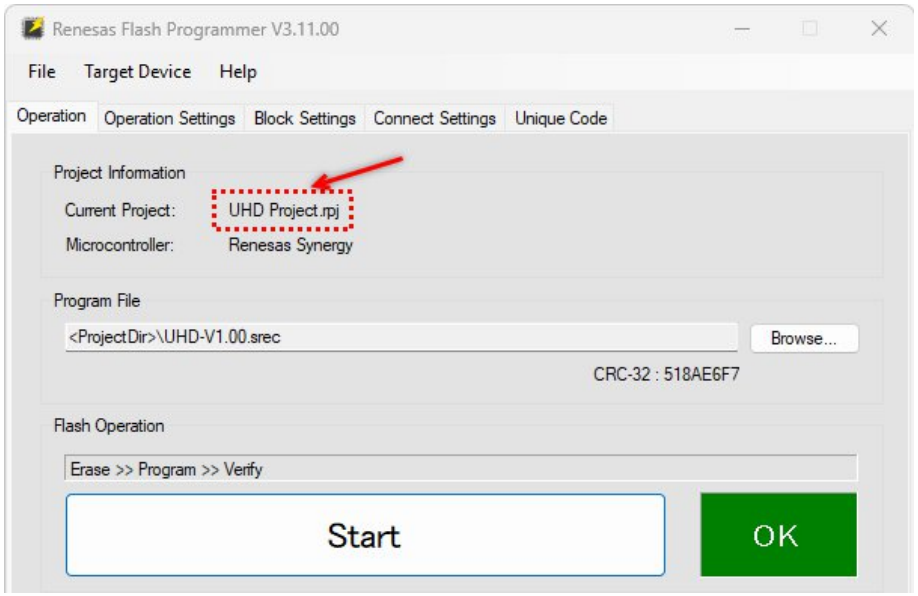
APPENDIX A – UPDATING FIRMWARE

This section assumes you have already installed '*Renesas Flash Programmer*' described in Appendix B.

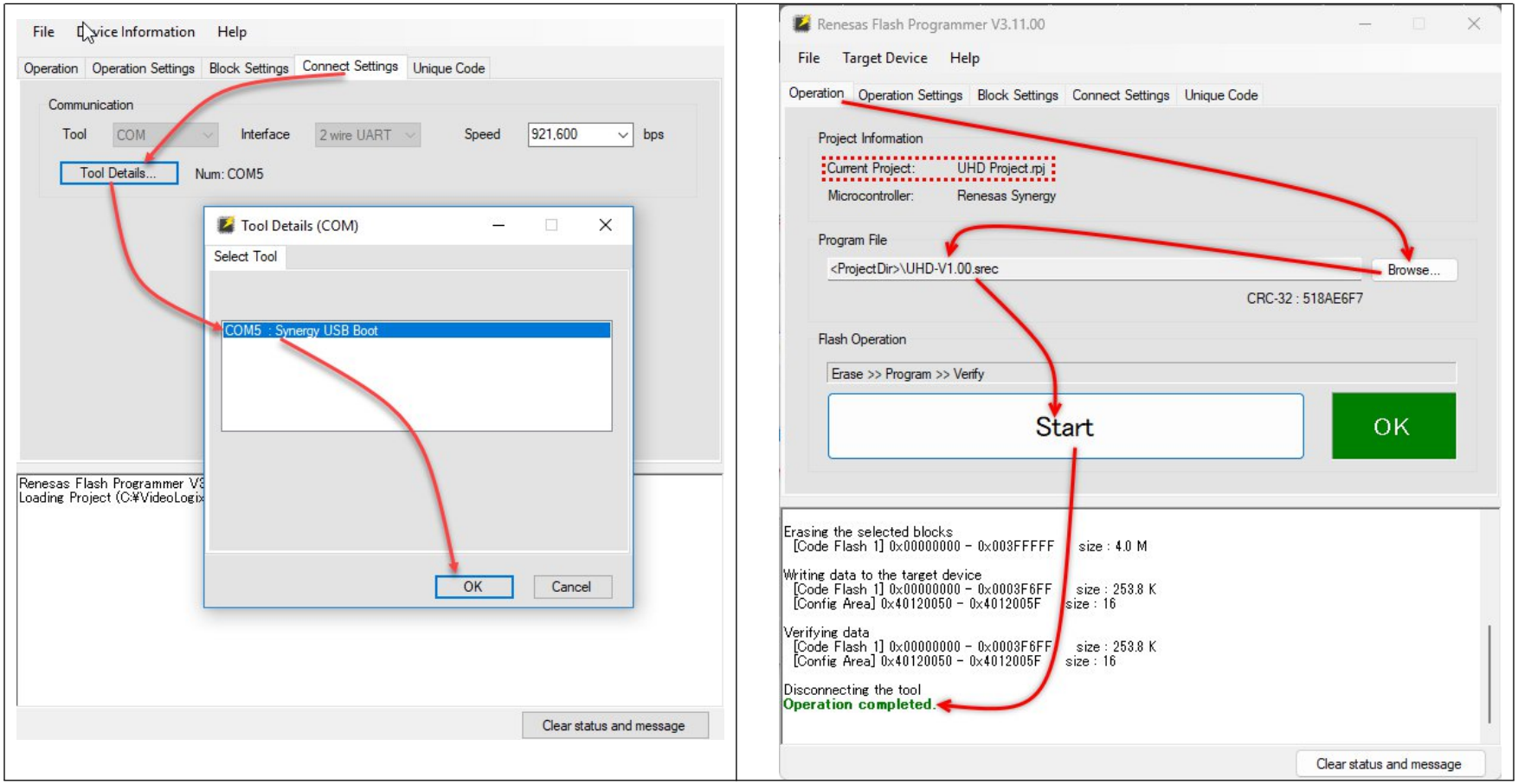
1. Use the ejector pin and set **PGM** switch (located on the front panel) to the lower position
2. Cycle power to PROTEUS-UHD
3. Connect USB cable from your PC to mini-USB port in the front panel
4. Your PC should acknowledge PROTEUS-UHD with a beep. Alternatively, the Device Manager will add the following:



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.



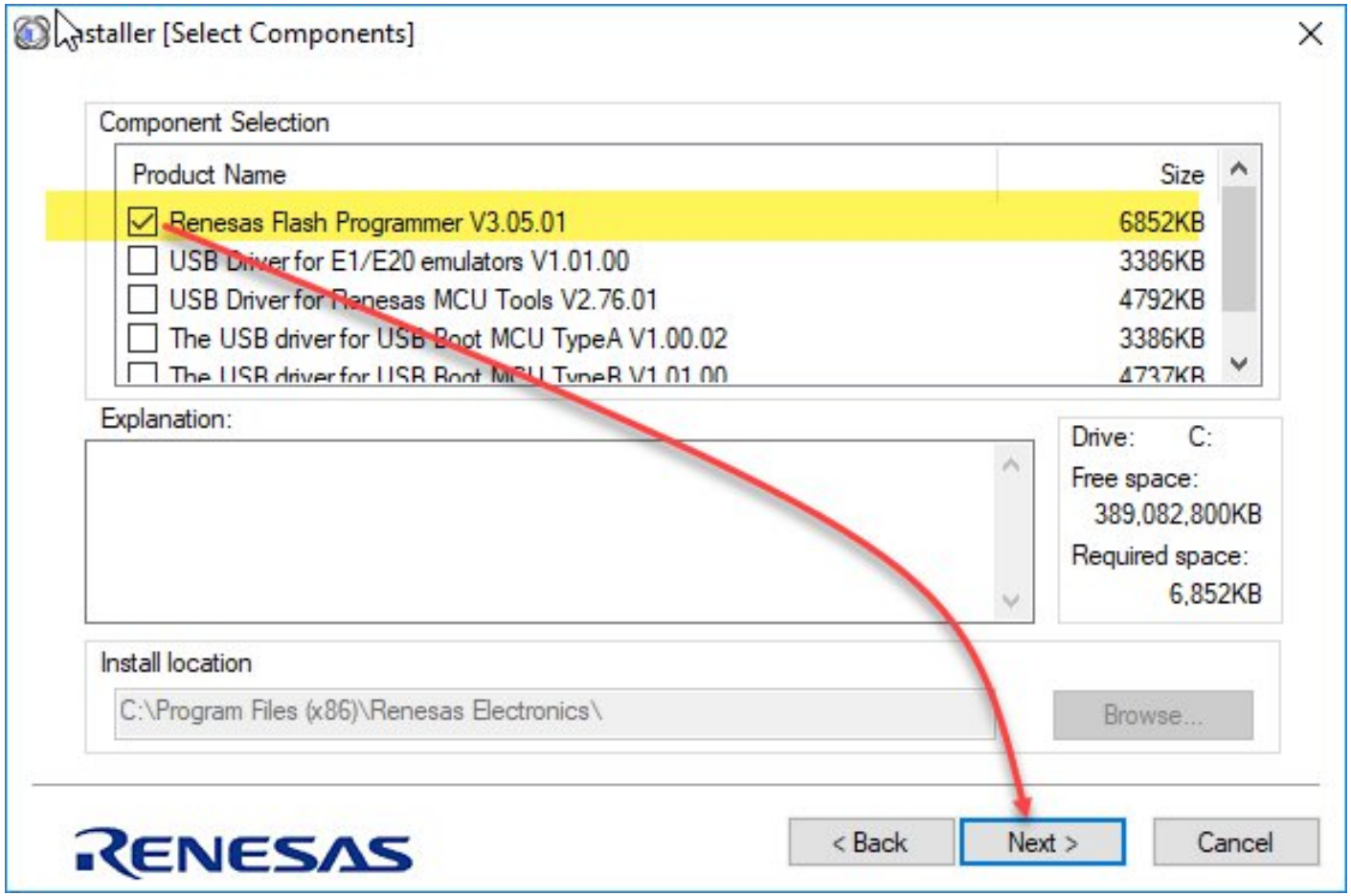
6. Follow the instructions below to load the firmware into Proteus.



7. After 'Operation is completed', set **PGM** switch (located on the front panel) to the upper position
8. Cycle power to PROTEUS and you are done.

APPENDIX B – INSTALL RENESAS FLASH PROGRAMMER

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



APPENDIX C – REGISTER'S NAME & ID

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

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

System Register		NMEA	
Name	ID	Name	ID
SYS:INFO:IP MODE	201	NMEA:MWV:ANGLE	210
SYS:INFO:IP ADDRESS	202	NMEA:MWV:REFERENCE	211
SYS:INFO:SUBNET	203	NMEA:MWV:SPEED	212
SYS:INFO:GATEWAY	204	NMEA:MWV:UNIT	213
SYS:INFO:MAC	205	NMEA:DBT:DEPTH	214
SYS:INFO:SNTP TIME	206	NMEA:DBS:DEPTH	215
SYS:INFO:SNTP DATE	207	NMEA:DPT:DEPTH	216
SYS:INFO:SNTP IP	208	NMEA:DPT:OFFSET	217
SYS:INFO:FILENAME	209	NMEA:DPT:RANGE	218
		NMEA:PCIPR:PITCH	219
		NMEA:PCIPR:ROLL	220
		NMEA:PCIT:TILT	221
		NMEA:HCC:HEADING	222