



ViSi-Genie Using Combined Objects

DOCUMENT DATE: **15th JANUARY 2019**
DOCUMENT REVISION: **1.01**



Description

This Application Note details how to use the **COMBINED OBJECTS** or **INPUT/OUTPUT OBJECTS**.

Most **INPUT OBJECTS** can also function as **OUTPUT OBJECTS**, with the notable exception of Keyboards.

Certain objects need both an input stimuli as well as produce an output event. For example, a slider thumb position may need to be remotely controlled from incoming serial data. A button may need to be animated not only using the touch screen but via serial data.

This application note requires:

- Workshop 4 has been installed according to the document Workshop 4 Installation;
- The user is familiar with the Workshop 4 environment and with the fundamentals of ViSi-Genie, as described in Workshop 4 User Guide and ViSi-Genie User Guide;
- When downloading an application note, a list of recommended application notes is shown. It is assumed that the user has read or has a working knowledge of the topics discussed in these recommended application notes.

A ViSi-Genie project is provided as example to help you along this application note.

Content

Description	2
Content	2
Application Overview	3
Setup Procedure	4
Create a New Project	4
<i>Create a New Project</i>	4
Simulation Procedure	5
<i>Build the Interface</i>	5
<i>Define the Commands</i>	8
Build and Upload the Project	10
Debugger Output	11
<i>Launch the Debugger</i>	11
<i>Use the Project</i>	13
Proprietary Information	17
Disclaimer of Warranties & Limitation of Liability	17

Application Overview

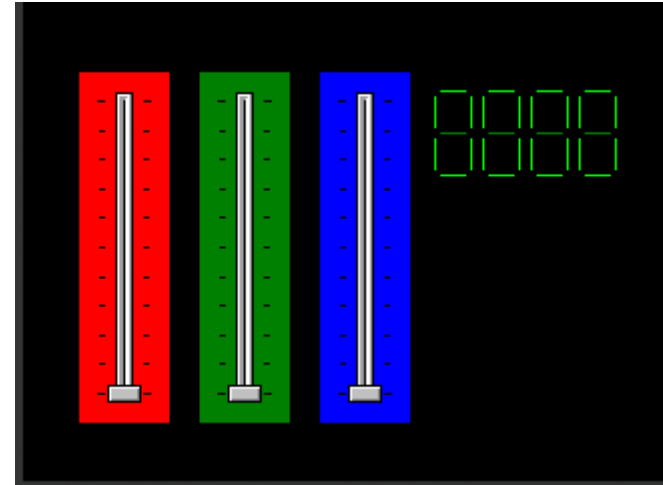
COMBINED OBJECTS bring flexibility to a user interface, acting as both **INPUT** and **OUTPUT**.

Imagine an air conditioning control panel, with three level of authorisation:

- Ground level, everyone can adjust the temperature;
- Management level, some managers can impose a given temperature;
- Technical level, only technicians can override the previous settings.

In this application, three track-bars have different priorities:



- The red track-bar is the master control and updates all the others: it is an **INPUT OBJECT**,
- The green track-bar is the intermediate control and only updates the blue track-bar: it is **COMBINED OBJECT**.
- So is the blue track-bar, as the slave control.
- Finally, the LED digits object only displays the value: it is an **OUTPUT OBJECT**.



Technical level has access to the red track-bar, management level to the green track-bar and everyone to the blue track-bar.

Setup Procedure

This application note comes with a zip file which contains two ViSi-Genie projects.

Name
 4D-AN-00011 DIABLO16.rar
 4D-AN-00011 PICASO.rar

For instructions on how to launch Workshop 4, how to open a ViSi-Genie project, and how to change the target display, kindly refer to the section “**Setup Procedure**” of the application note:

[ViSi Genie Getting Started – First Project for Picaso Displays](#) (for Picaso)

or

[ViSi Genie Getting Started – First Project for Diablo16 Displays](#) (for Diablo16).

Create a New Project

Create a New Project

For instructions on how to create a new ViSi-Genie project, please refer to the section “**Create a New Project**” of the application note

[ViSi Genie Getting Started – First Project for Picaso Displays](#) (for Picaso)

or

[ViSi Genie Getting Started – First Project for Diablo16 Displays](#) (for Diablo16).

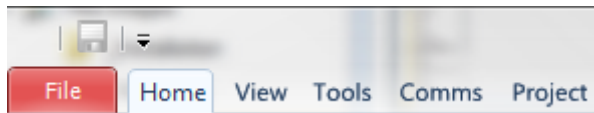
Simulation Procedure

You can load the example...

Example: 4D-AN-00011 PICASO – Combined Objects or 4D-AN-00011 DIABLO16 – Combined Objects

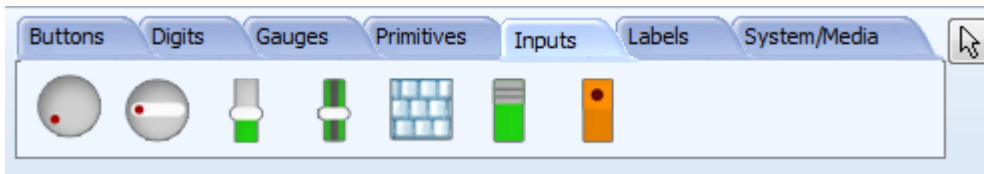
...or follow the procedures described hereafter.

Select the **Home** menu to display the objects:



Build the Interface

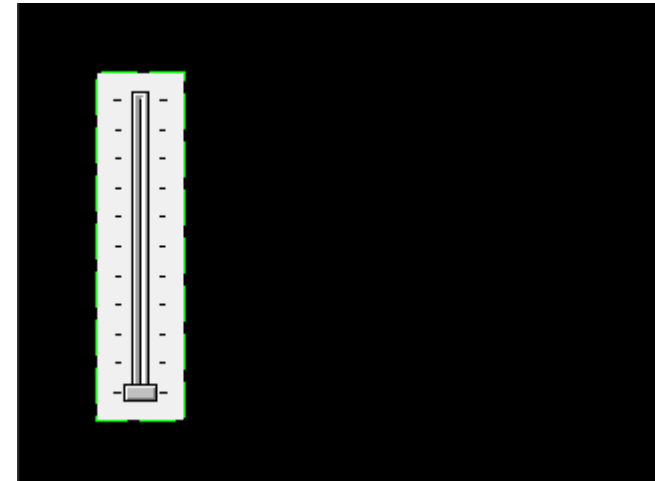
The **TrackBar** object is located on the Inputs pane:



Click first on the **TrackBar** icon...

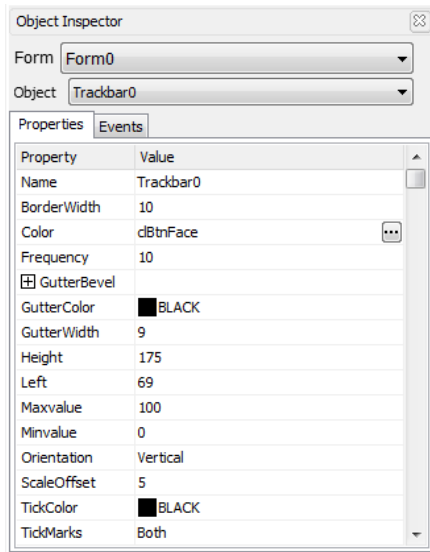



...and then click on the desired location on the form to place it:

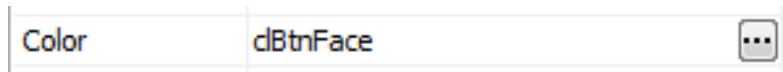


This first track-bar is named **TrackBar0**.

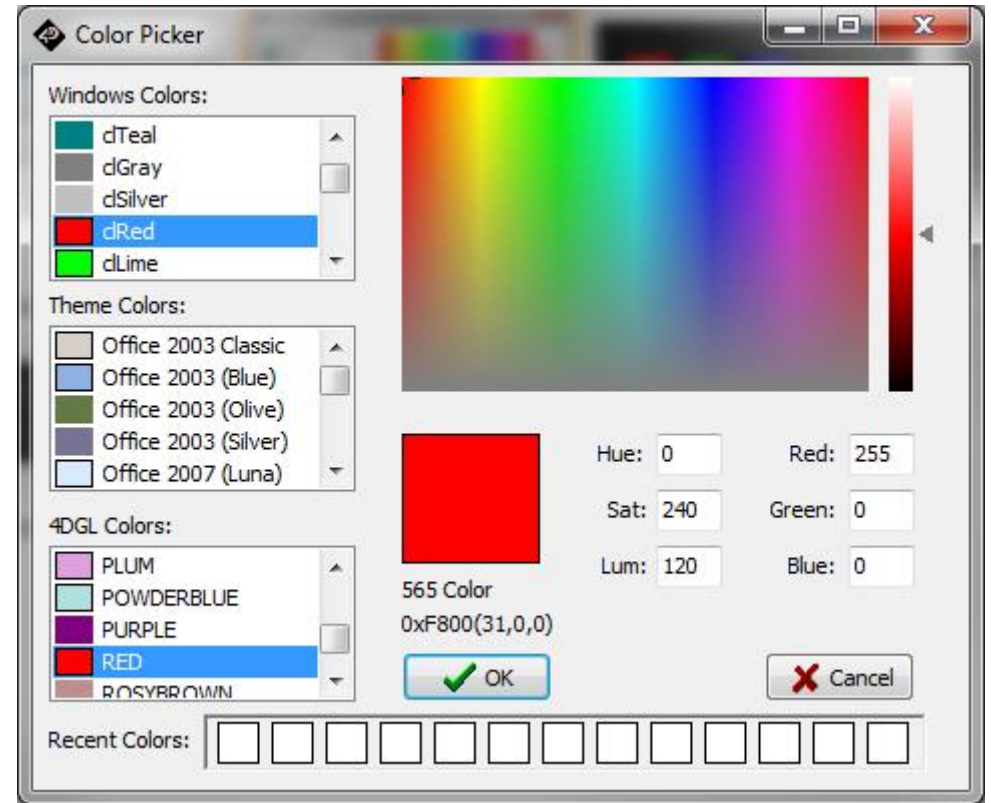
To change the colour of the track-bar, go to the Object Inspector...



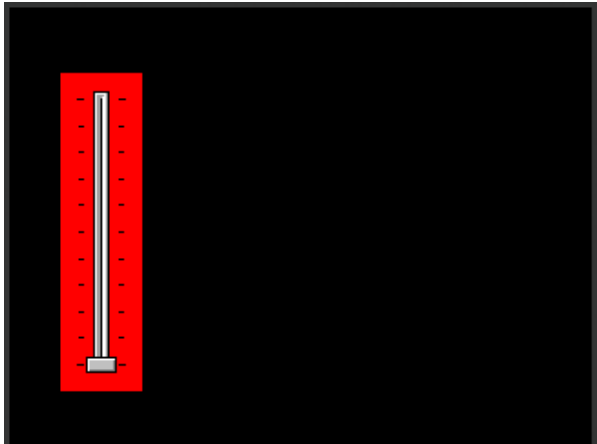
...click on the  on the Colour line...



...a new window Colour Picker is displayed.

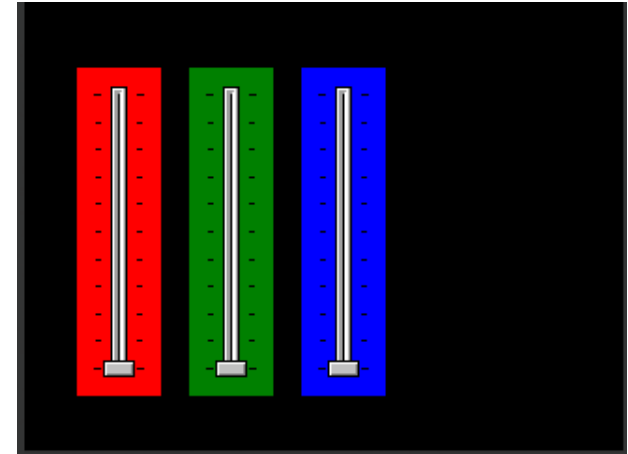


Pick the colour and press **OK** to confirm. The track-bar is red now:

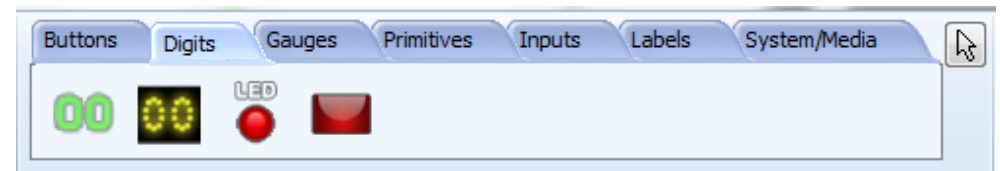


Proceed the same way for the green and blue track-bars, named respectively **TrackBar1** and **TrackBar2**.

Result is:



The **LedDigits** object is located on the Inputs pane:

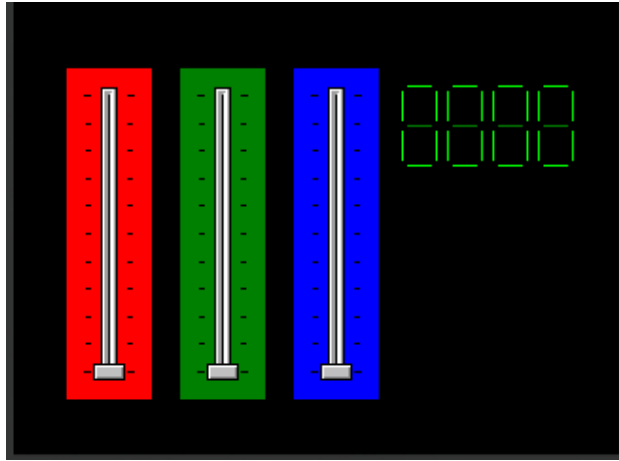


Click first on the **LedDigits** icon...



..and place it on the screen.

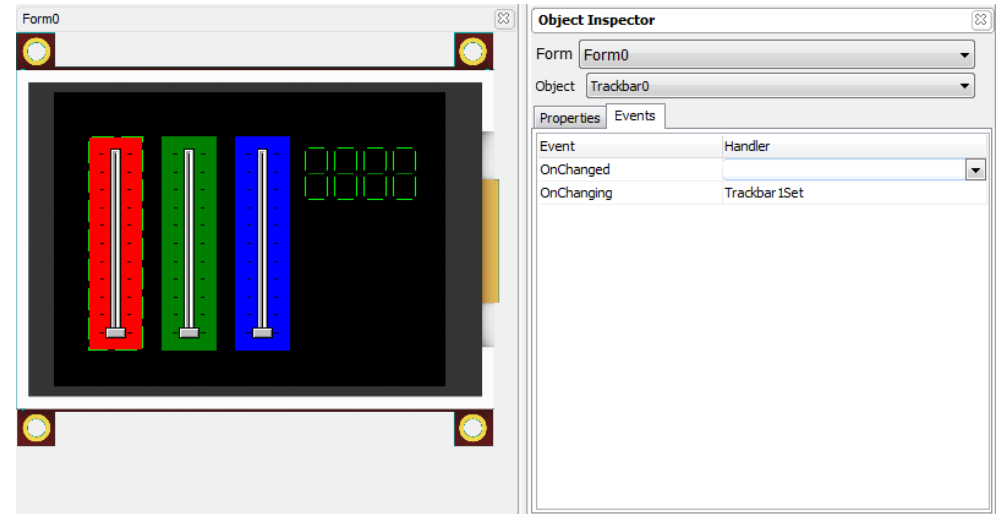
Final result is:



Define the Commands

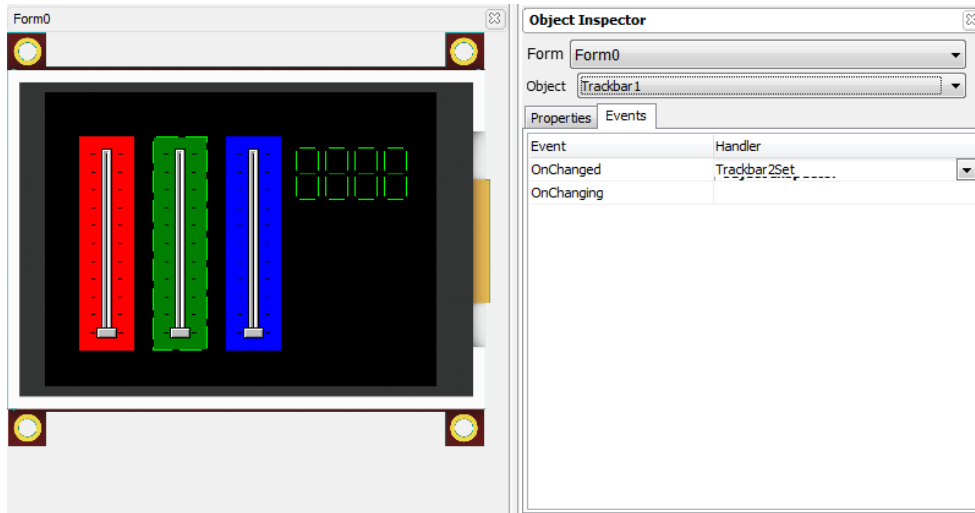
The red track-bar **TrackBar0** is the master control.

When the cursor of **TrackBar0** is moved, the **onChanging** event is raised and the Set command is sent to the green track-bar **TrackBar1** along with the value:



The red track-bar is an **INPUT OBJECT**.

The green track-bar **TrackBar1** is the intermediate control.
When the cursor of **TrackBar1** is moved and released, the **onChanged** event is raised and the Set command is sent to the blue track-bar **TrackBar2** along with the value:

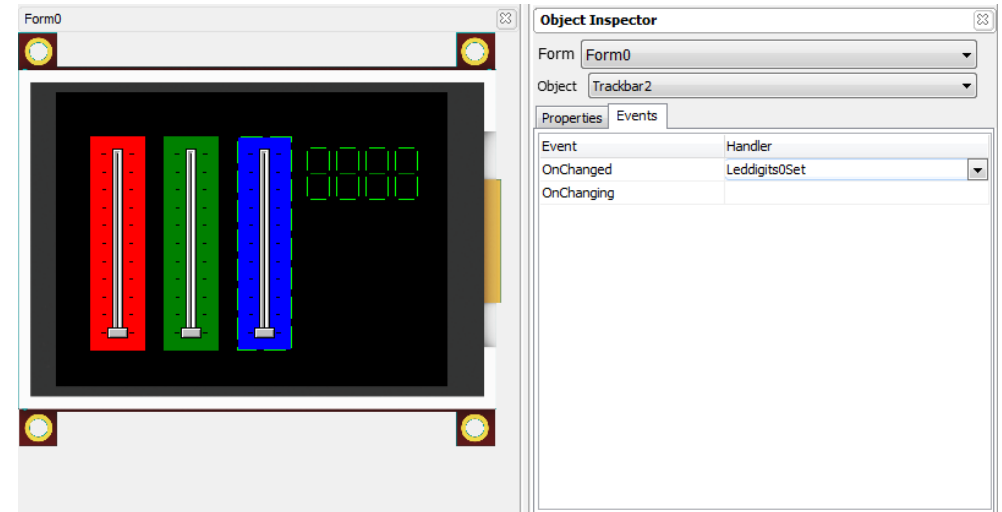


The green track-bar is updated on two different ways:

- When the user move and release the cursor, the track-bar records the new value: it acts as an **INPUT OBJECT**;
- When it received a command from the red track-bar: it acts as an **OUTPUT OBJECT**.

The green track-bar is a **COMBINED OBJECT**.

The blue track-bar is the slave control.
It has no event.

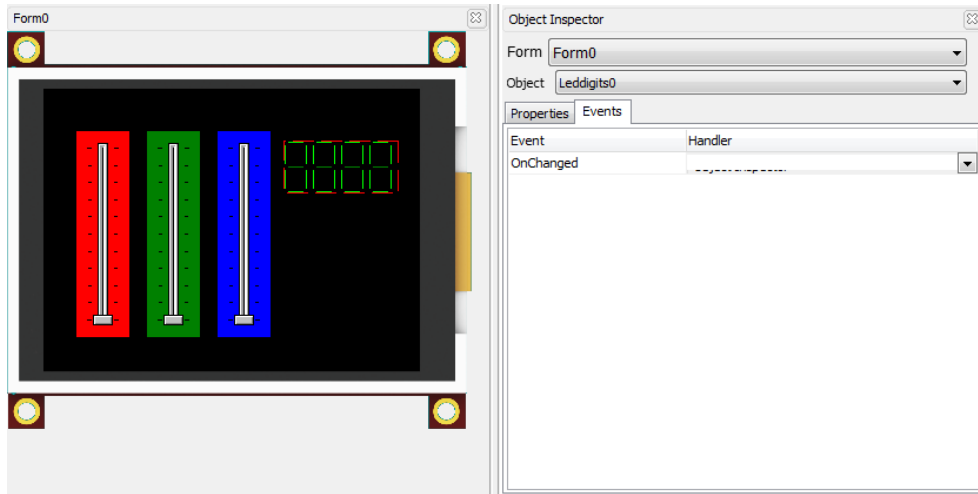


The blue track-bar is updated on two different ways:

- When the user move and release the cursor, the track-bar records the new value: it acts as an **INPUT OBJECT**;
- When it received a command from the red track-bar: it acts as an **OUTPUT OBJECT**.

The blue track-bar is also a **COMBINED OBJECT**.

The last object, LEDDigit0, only displays the value it receives from the blue track-bar. It has no event.



The LED digit is an **OUTPUT OBJECT**.

Build and Upload the Project

For instructions on how to build and upload a ViSi-Genie project to the target display, please refer to the section “**Build and Upload the Project**” of the application note

[ViSi Genie Getting Started – First Project for Picaso Displays](#) (for Picaso)

or

[ViSi Genie Getting Started – First Project for Diablo16 Displays](#) (for Diablo16).

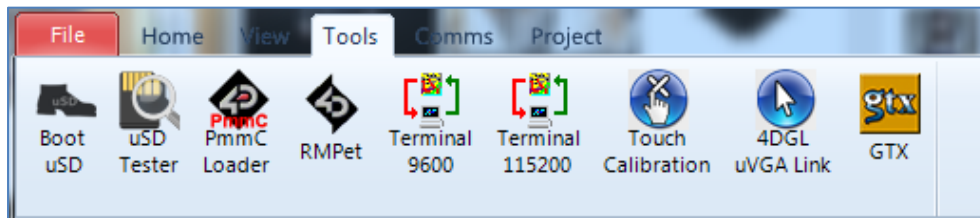
The uLCD-32PTU and/or the uLCD-35DT display modules are commonly used as examples, but the procedure is the same for other displays.

Debugger Output

Setting the handler to Message sends the values to the debugger, **Genie Test Executor** or GTX.

Launch the Debugger

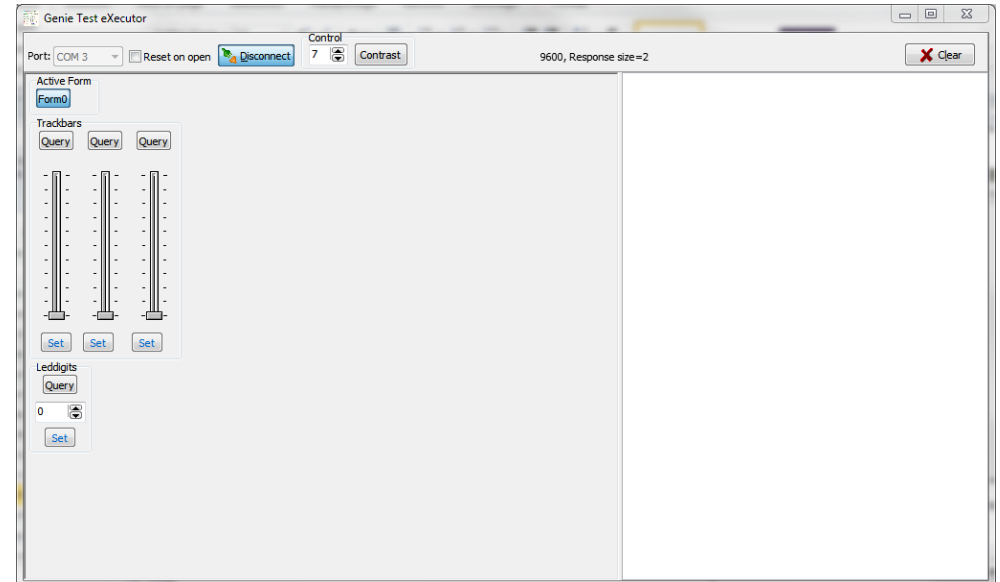
To launch the **Genie Test Executor** or GTX, select the **Tools** menu...



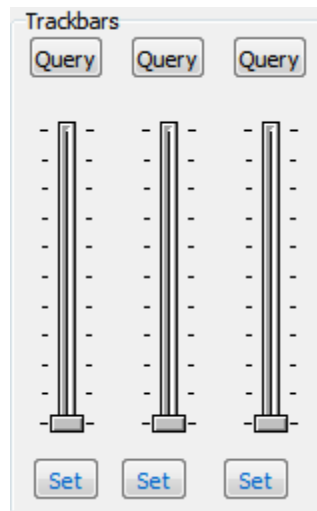
...and then click on the **GTX** button.



A new screen appears, with the form and objects we have defined previously:

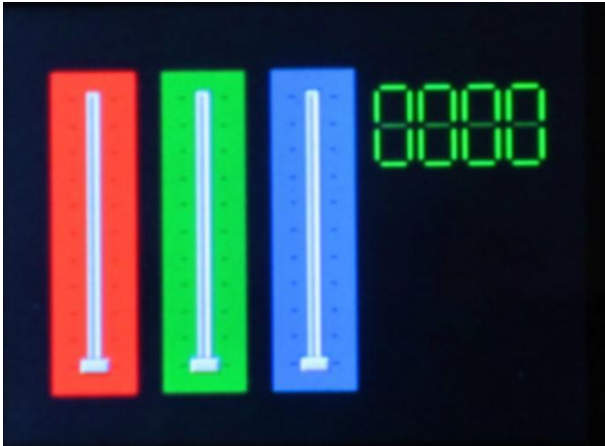


The **Query** buttons read the value from each track-bard and display them on the debugger:

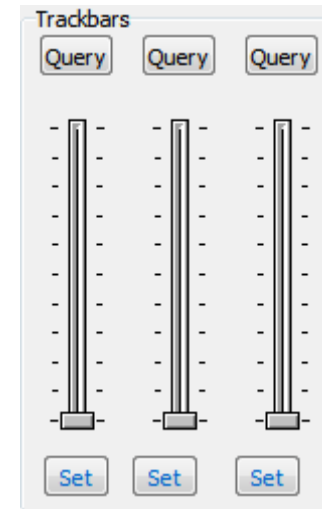


Use the Project

Initial values are all set to zero:



Press the **Query** buttons. The debugger shows:



The right part of the debugger shows the messages sent and received for each track-bar:

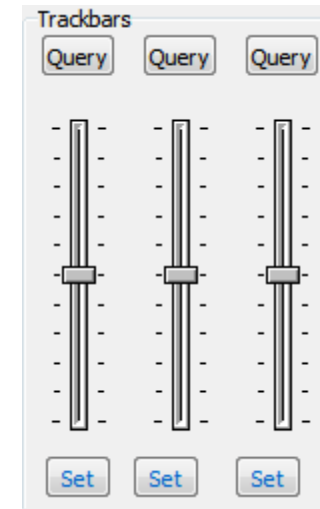
```
Request Trackbar Value 15:26:37.299 [00 05 00 05]  
Trackbar Value 15:26:37.331 [05 05 00 00 00 00]  
Request Trackbar Value 15:26:37.799 [00 05 01 04]  
Trackbar Value 15:26:37.830 [05 05 01 00 00 01]  
Request Trackbar Value 15:26:38.267 [00 05 02 07]  
Trackbar Value 15:26:38.298 [05 05 02 00 00 02]
```

The hexadecimal value of **00 00** stands for 00 in decimal.

Move the leftmost track-bars to 50 on the display module: all the other track-bars are updated.



Press the **Query** buttons. The debugger now displays:



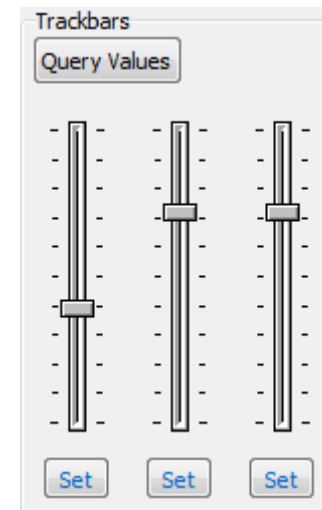
The right part of the debugger shows the messages sent and received for each track-bar:

```
Request Trackbar Value 15:21:43.067 [00 05 00 05]  
Trackbar Value 15:21:43.114 [05 05 00 00 32 32]  
Request Trackbar Value 15:21:43.567 [00 05 01 04]  
Trackbar Value 15:21:43.582 [05 05 01 00 32 33]  
Request Trackbar Value 15:21:45.641 [00 05 02 07]  
Trackbar Value 15:21:45.673 [05 05 02 00 32 30]
```

The hexadecimal value of **00 32** stands for 50 in decimal.
Move the central track-bar on the display module: only the rightmost track-bar is updated.



Press the **Query** buttons. The debugger now displays:

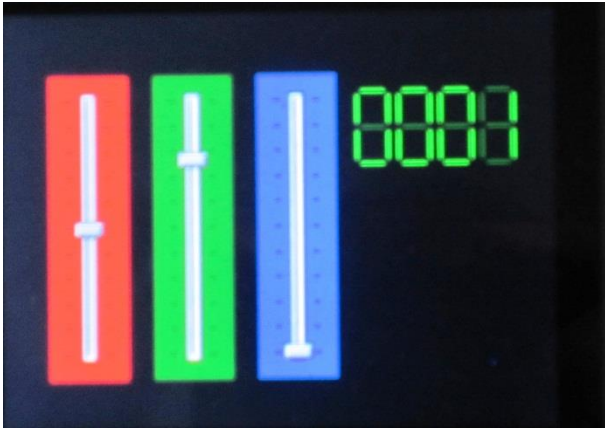


The right part of the debugger shows the messages sent and received for each track-bar:

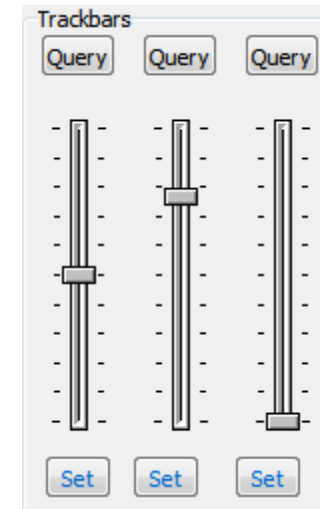
```
Request Trackbar Value 15:16:07.366 [00 05 00 05]  
Trackbar Value 15:16:07.412 [05 05 00 00 32 32]  
Request Trackbar Value 15:16:08.083 [00 05 01 04]  
Trackbar Value 15:16:08.130 [05 05 01 00 4D 4C]  
Request Trackbar Value 15:16:08.536 [00 05 02 07]  
Trackbar Value 15:16:08.567 [05 05 02 00 4D 4F]
```

The hexadecimal value of **00 4D** stands for 77 in decimal.

Move the rightmost track-bar on the display module: the other track-bars remain unchanged.



Press the **Query** buttons. The debugger now displays:



The right part of the debugger shows the messages sent and received for each track-bar:

```
Request Trackbar Value 15:16:49.160 [00 05 00 05]  
Trackbar Value 15:16:49.192 [05 05 00 00 32 32]  
Request Trackbar Value 15:16:49.706 [00 05 01 04]  
Trackbar Value 15:16:49.753 [05 05 01 00 4D 4C]  
Request Trackbar Value 15:16:50.596 [00 05 02 07]  
Trackbar Value 15:16:50.627 [05 05 02 00 00 02]
```

The hexadecimal value of **00 00** stands for 00 in decimal.

Proprietary Information

The information contained in this document is the property of 4D Systems Pty. Ltd. and may be the subject of patents pending or granted, and must not be copied or disclosed without prior written permission.

4D Systems endeavours to ensure that the information in this document is correct and fairly stated but does not accept liability for any error or omission. The development of 4D Systems products and services is continuous and published information may not be up to date. It is important to check the current position with 4D Systems.

All trademarks belong to their respective owners and are recognised and acknowledged.

Disclaimer of Warranties & Limitation of Liability

4D Systems makes no warranty, either expresses or implied with respect to any product, and specifically disclaims all other warranties, including, without limitation, warranties for merchantability, non-infringement and fitness for any particular purpose.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

In no event shall 4D Systems be liable to the buyer or to any third party for any indirect, incidental, special, consequential, punitive or exemplary damages (including without limitation lost profits, lost savings, or loss of business opportunity) arising out of or relating to any product or service provided or to be provided by 4D Systems, or the use or inability to use the same, even if 4D Systems has been advised of the possibility of such damages.

4D Systems products are not fault tolerant nor designed, manufactured or intended for use or resale as on line control equipment in hazardous environments requiring fail – safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of the product could lead directly to death, personal injury or severe physical or environmental damage ('High Risk Activities'). 4D Systems and its suppliers specifically disclaim any expressed or implied warranty of fitness for High Risk Activities.

Use of 4D Systems' products and devices in 'High Risk Activities' and in any other application is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless 4D Systems from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any 4D Systems intellectual property rights.