APPLICATION NOTES

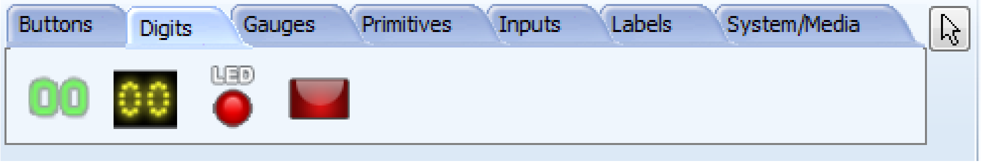
4D SYSTEMS
TURNING TECHNOLOGY INTO ART

ViSi-Genie Digital Displays

DOCUMENT DATE:	**13th April 2019**
DOCUMENT REVISION:	**1.1**

Description

This Application Note explores the possibilities provided by ViSi-Genie for the **Digits** objects:



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.

Four ViSi-Genie projects are provided as examples to help you along this application note.

Content

Application Overview

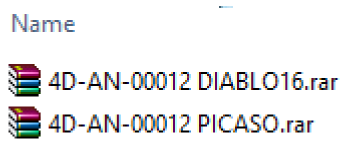
It is often difficult to design a graphical display without being able to see the immediate results of the application code. ViSi-Genie is the perfect software tool that allows the user to see the instant results of their desired graphical layout with this large selection of gauges and meters that can simply be dragged and dropped onto the simulated module display.

- LED digits
- Custom digits
- LED
- User LED

Each object can have properties edited and at the click of a button, all relevant code is produced in the user program. Each feature of ViSi-Genie will be outlined with examples below.

Setup Procedure

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



Name

4D-AN-00012 DIABLO16.rar

4D-AN-00012 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

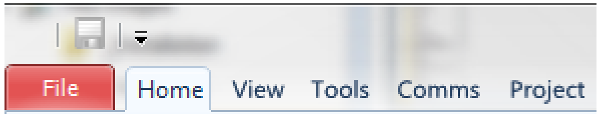
ViSi Genie Getting Started – First Project for Picaso Displays (for Picaso)

or

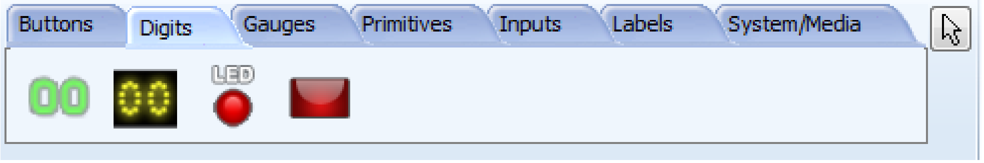
ViSi Genie Getting Started – First Project for Diablo16 Displays (for Diablo16).

Simulation Procedure

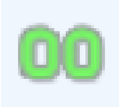
Select the **Home** menu to display the objects:



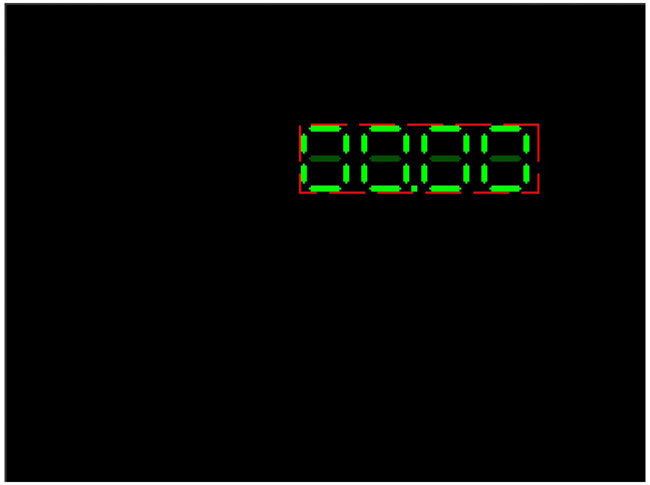
The **Digits** objects are located on the Gauges pane:



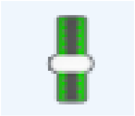
To add an object, first click on the desired icon, here start with the first one, the **LED Digits**…



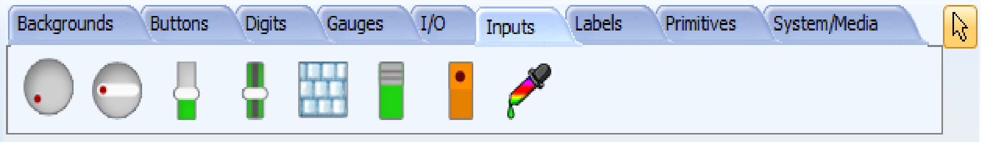
…and then click on the WYSIWYG screen to place it.



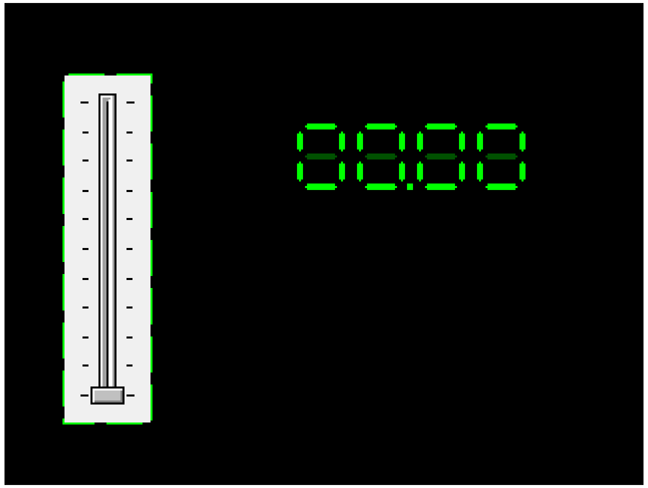
Add a track-bar as input object…



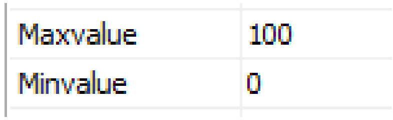
…from the **Inputs** pane…



...and place it on the left of the form:

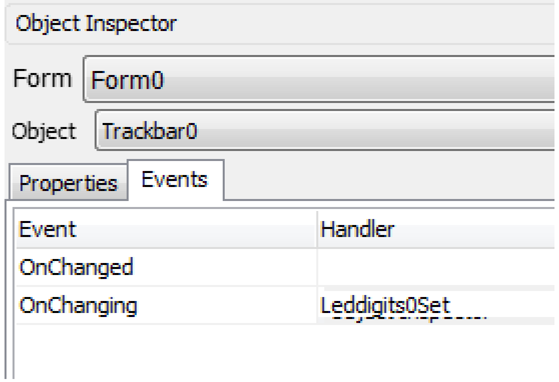


By default, the track-bar has the following minimum and maximum values:



Maxvalue	100
Minvalue	0

Define the event **onChanging** for the **TrackBar0** as **LedDigits0Set**:



Object Inspector

Form: Form0

Object: Trackbar0

Event	Handler
OnChanged	
OnChanging	Leddigits0Set

Each time the cursor on the track-bar is moved, the LED digits display is updated accordingly.

LED Digits Options

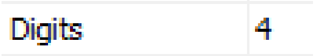
You can load the example...

> ***Example: 4D-AN-00012 PICASO – LED Digits or 4D-AN-00012 DIABLO16 – LED Digits***

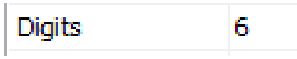
...or follow the procedure described hereafter.

Number of Digits

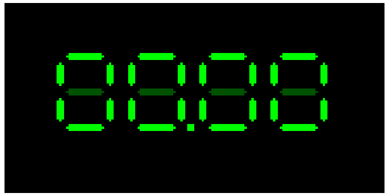
By default, the number of digits is 4.

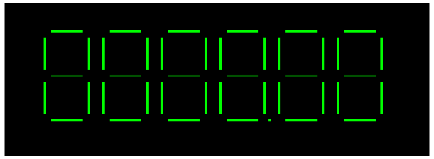


Digits	4



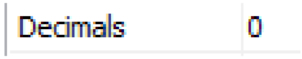
Digits	6



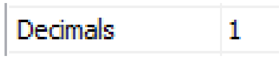


Number of Decimals

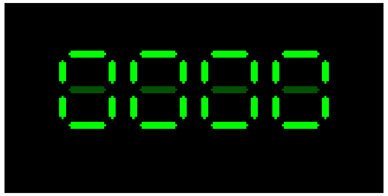
The number of decimals can be adjusted. By default, it is 2.

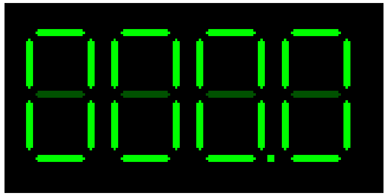


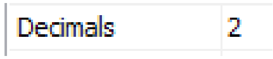
Decimals	0



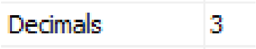
Decimals	1



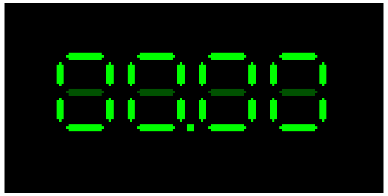


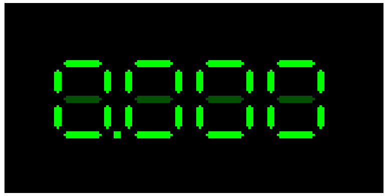


Decimals	2



Decimals	3



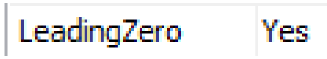


However, the same 1234 figure will be displayed independently of the decimals. Here are the results:

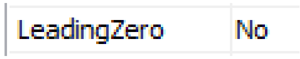
- **1234** if decimals = 0
- **123.4** if decimals = 1
- **12.34** if decimals = 2
- **1.234** if decimals = 3

Leading Zero

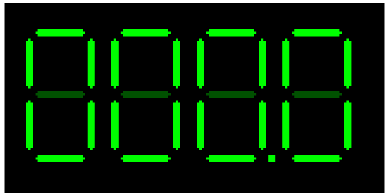
Leading zeroes can be omitted:

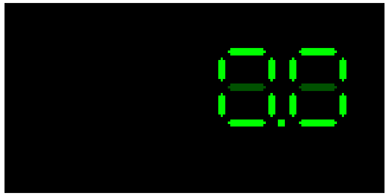


LeadingZero	Yes



LeadingZero	No

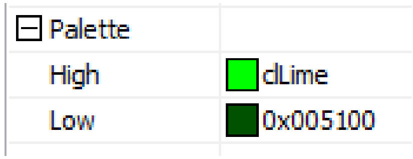




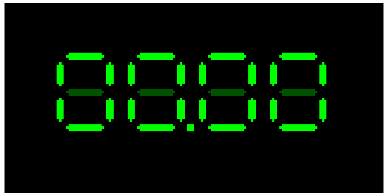
Colours

There are two colours to define:

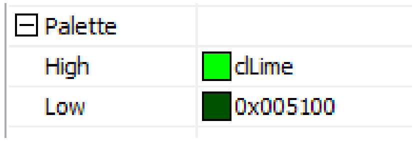
- **High** for segments on,
- **Low** for segments off.



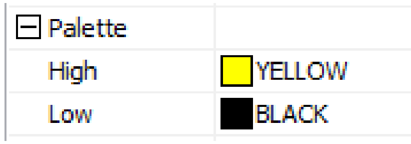
⊟ Palette	
High	🟩 clLime
Low	🟩 0x005100



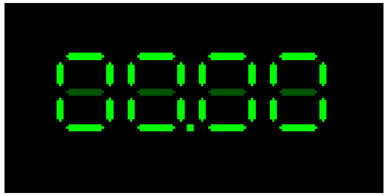
To obtain a high contrast yellow on black display, just define **High** as yellow and **Low** as black:

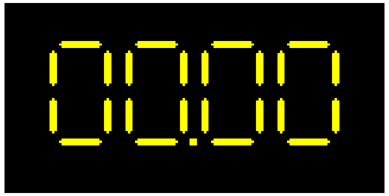


⊟ Palette	
High	🟩 clLime
Low	🟩 0x005100



⊟ Palette	
High	🟨 YELLOW
Low	⬛ BLACK





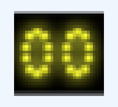
Custom Digits Options

You can load the example…

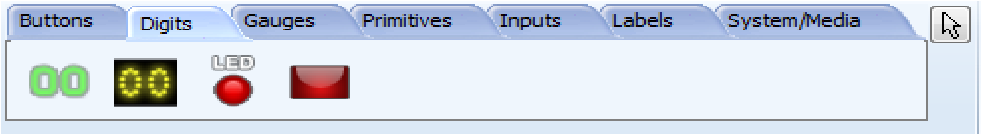
Example: 4D-AN-00012 PICASO – Custom Digits or 4D-AN-00012 DIABLO16 – Custom Digits

…or follow the procedure described hereafter.

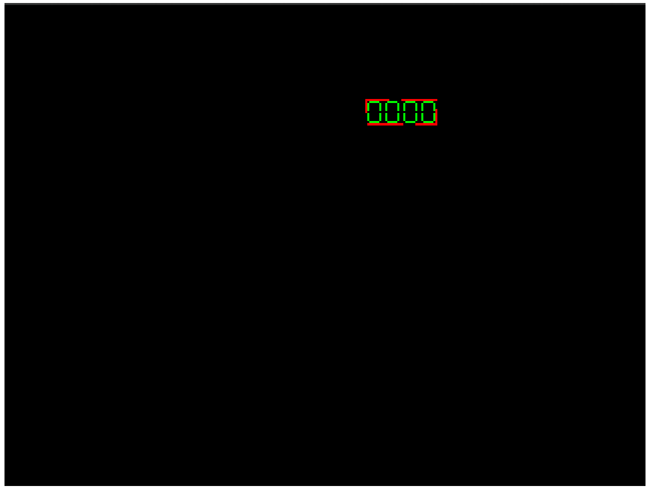
Start with a new project and select the Custom Digits…



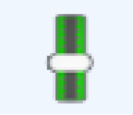
…from the **Digits** pane…



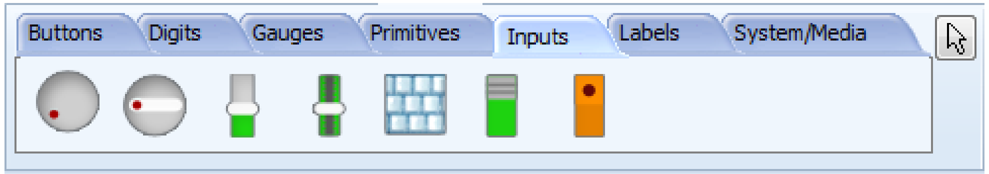
…and then click on the WYSIWYG screen to place it.



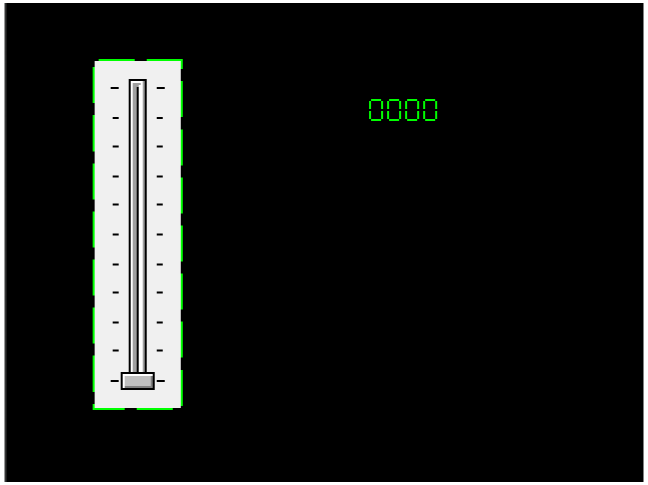
Add a track-bar as input object…



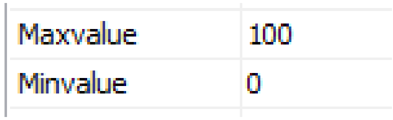
…from the **Inputs** pane…



…and place it on the left of the form:

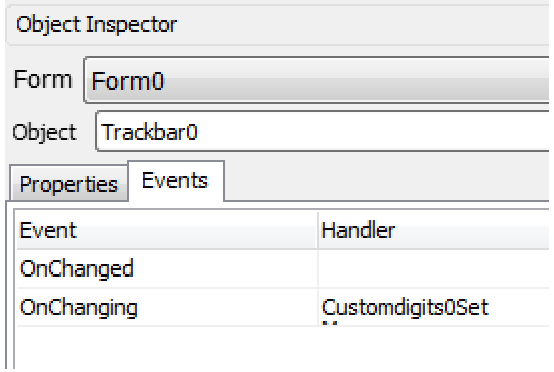


By default, the track-bar has the following minimum and maximum values:



Maxvalue	100
Minvalue	0

Define the event **onChanging** for the **TrackBar0** as **CustomDigits0Set**:


Object Inspector

Form: Form0

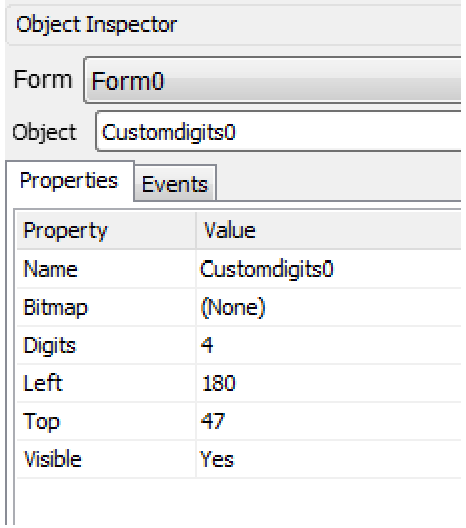
Object: Trackbar0

Properties | Events

Event	Handler
OnChanged	
OnChanging	Customdigits0Set

Each time the cursor on the track-bar is moved, the custom digits display is updated accordingly.

The options provided by the **CustomDigits** object are limited:


Object Inspector

Form: Form0

Object: Customdigits0

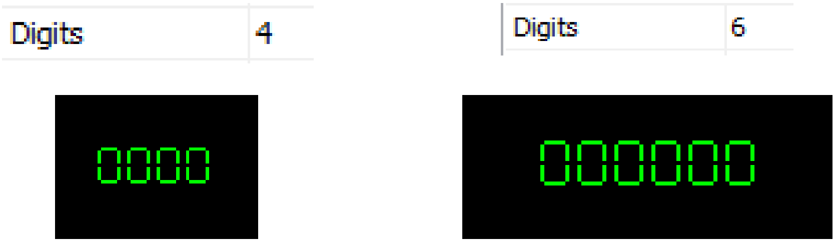
Properties | Events

Property	Value
Name	Customdigits0
Bitmap	(None)
Digits	4
Left	180
Top	47
Visible	Yes

The **CustomDigits** object is perfect for basic display.

Number of Digits

By default, the number of digits is 4.


Digits 4

Digits 6

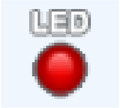
LED Options

You can load the example…

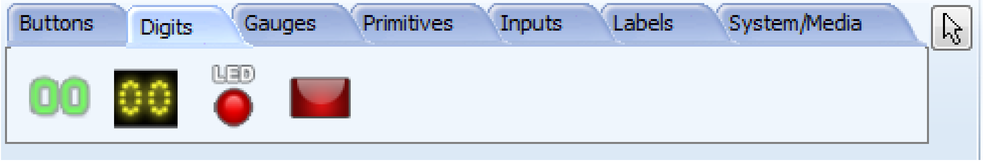
Example: 4D-AN-00012 PICASO – LED or 4D-AN-00012 DIABLO16 – LED

…or follow the procedure described hereafter.

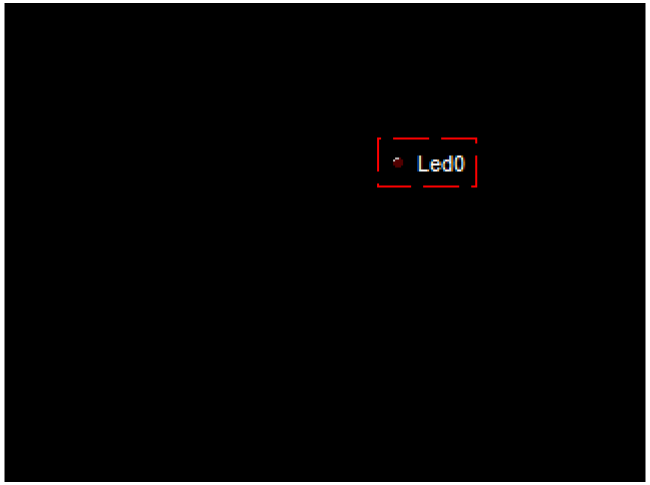
Start with a new project and select the LED object…



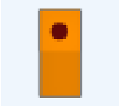
…from the **Digits** pane…



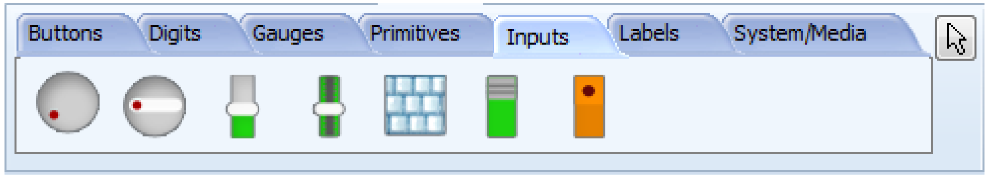
…and then click on the WYSIWYG screen to place it.



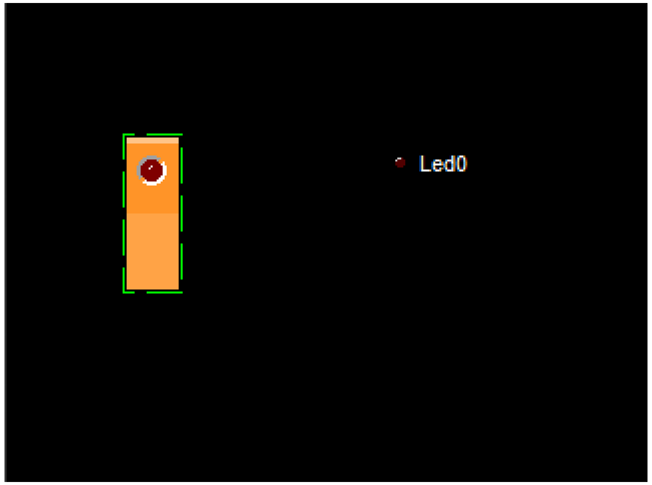
Add a rocker switch as input object…



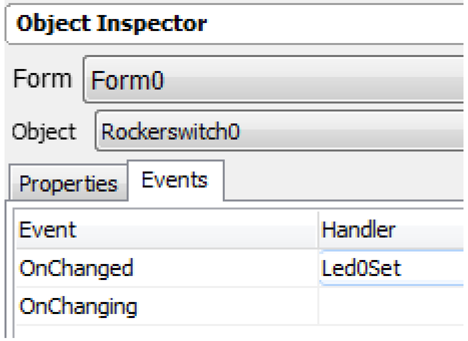
…from the **Inputs** pane…



…and place it on the left of the form:



Define the event **onChanged** for the **RockerSwitch0** as **Led0Set**:



Object Inspector

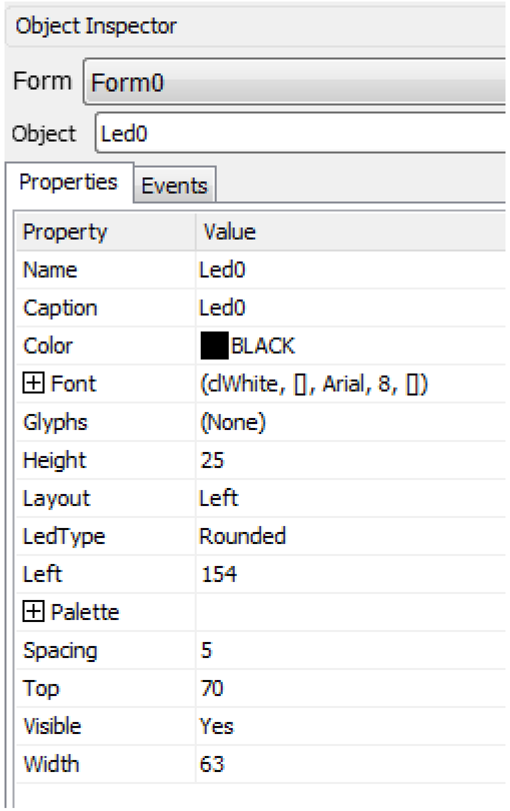
Form Form0

Object Rockerswitch0

Properties Events

Event	Handler
OnChanged	Led0Set
OnChanging	

Each time the switch is turned on and off, the LED is updated accordingly. The options include parameters for the LED and for the caption:



Object Inspector

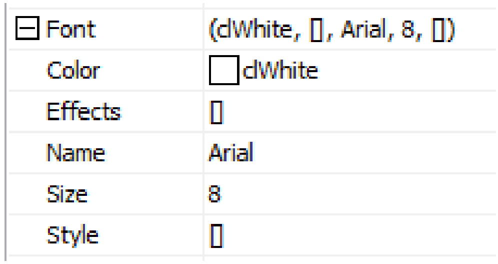
Form: Form0

Object: Led0

Property	Value
Name	Led0
Caption	Led0
Color	■ BLACK
⊞ Font	(clWhite, [], Arial, 8, [])
Glyphs	(None)
Height	25
Layout	Left
LedType	Rounded
Left	154
⊞ Palette	
Spacing	5
Top	70
Visible	Yes
Width	63

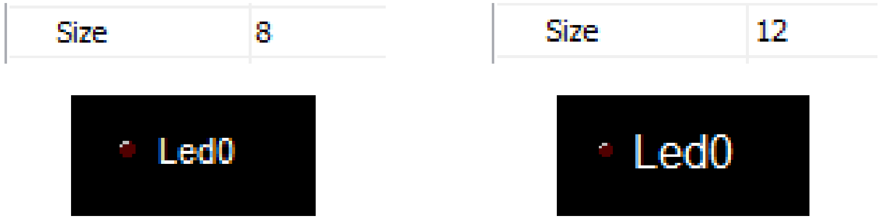
Font Options

The font options provide the usual parameters:



⊟ Font	(clWhite, [], Arial, 8, [])
Color	□ clWhite
Effects	[]
Name	Arial
Size	8
Style	[]

For example, the size can be adjusted:

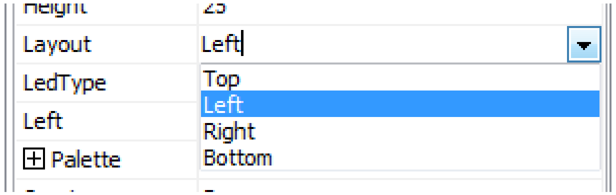


Size	8

Size	12

Layout Options

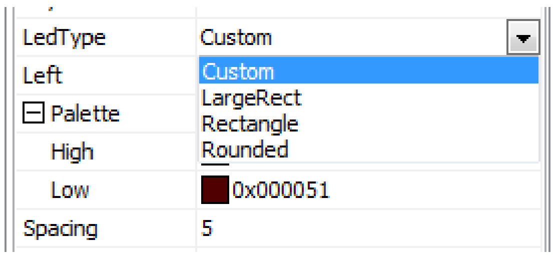
The LED can be placed anywhere compared to the caption…



Height	25
Layout	Left
LedType	
Left	
⊞ Palette	

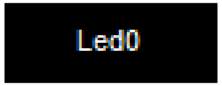
LED Options

The LED can have different types:

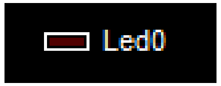


LedType	Custom
Left	Custom LargeRect Rectangle Rounded
⊟ Palette	
High	
Low	■ 0x000051
Spacing	5

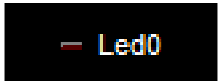
Custom



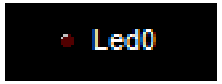
LargeRect



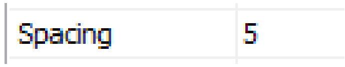
Rectangle



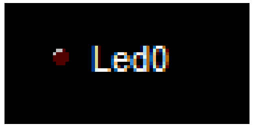
Rounded

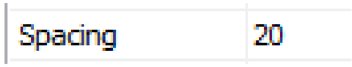


The **Spacing** parameter defines the distance between the LED and the caption:

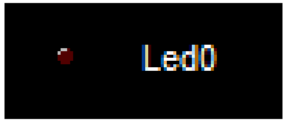


Spacing	5



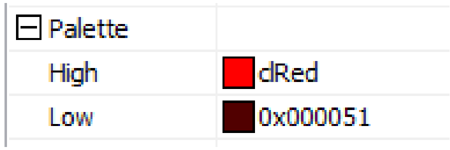


Spacing	20



Palette Options

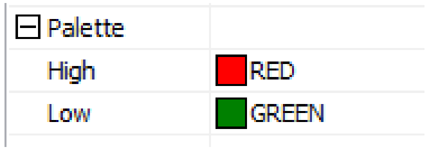
Two palettes need to be defined, one for each state:



⊟ Palette	
High	🟥 clRed
Low	■ 0x000051

- **High**, when the LED is on;
- **Low**, when the LED is off.

In the following example, the LED is red when on or high; and green when off or low:



⊟ Palette	
High	🟥 RED
Low	🟩 GREEN

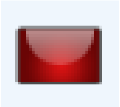
User LED Options

You can load the example…

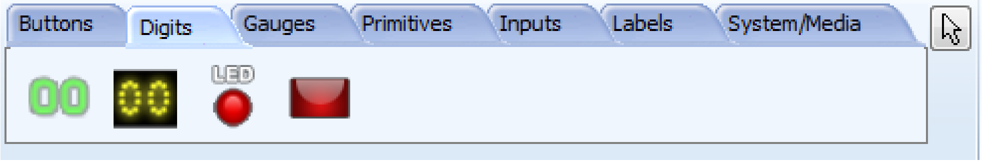
Example: 4D-AN-00012 PICASO – User LED or 4D-AN-00012 DIABLO16 – User LED

…or follow the procedure described hereafter.

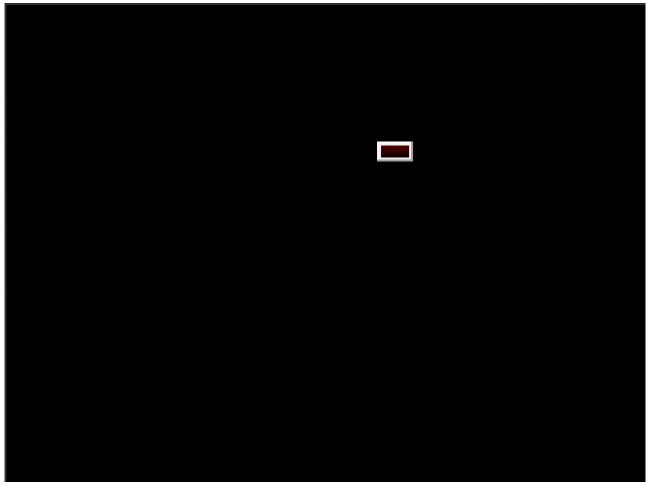
Start with a new project and select the LED object…



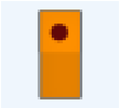
…from the **Digits** pane…



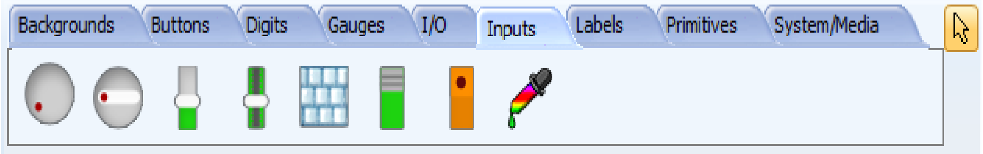
…and then click on the WYSIWYG screen to place it.



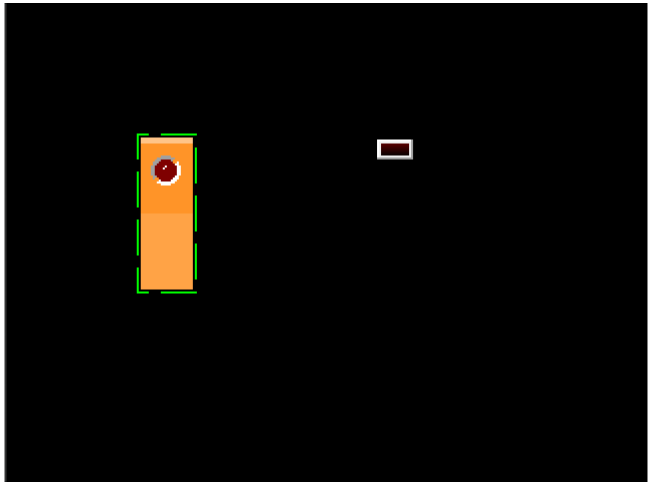
Add a rocker switch as input object…



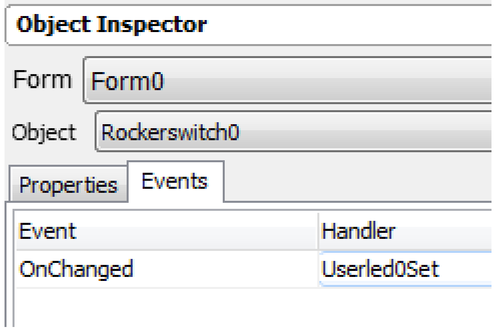
…from the **Inputs** pane…



…and place it on the left of the form:



Define the event **onChanged** for the **RockerSwitch0** as **UserLed0Set**:



Object Inspector

Form: Form0

Object: Rockerswitch0

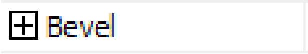
Properties | Events

Event	Handler
OnChanged	Userled0Set

Each time the switch is turned on and off, the LED is updated accordingly.

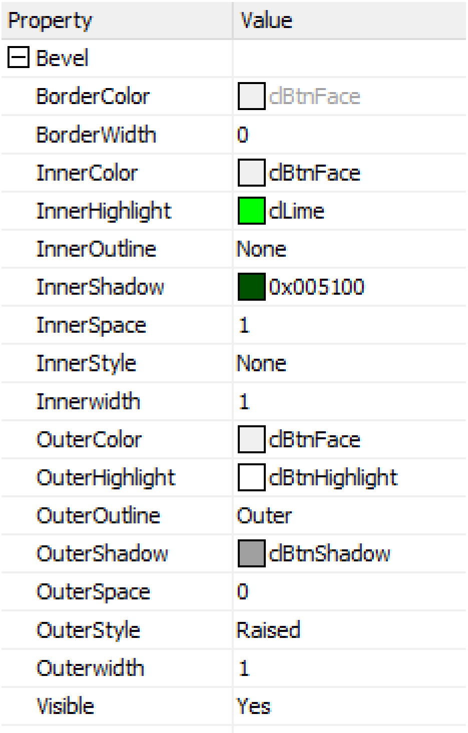
Bevel Options

All the options related to the bevel are under the **Bevel** line:



⊞ Bevel

Click on the ⊞ to show them all:

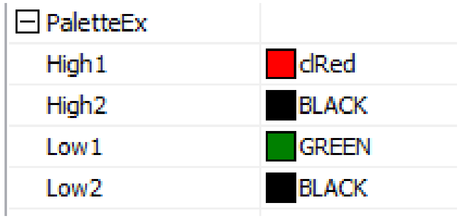


Property	Value
⊟ Bevel	
BorderColor	□ clBtnFace
BorderWidth	0
InnerColor	□ clBtnFace
InnerHighlight	🟩 clLime
InnerOutline	None
InnerShadow	■ 0x005100
InnerSpace	1
InnerStyle	None
Innerwidth	1
OuterColor	□ clBtnFace
OuterHighlight	□ clBtnHighlight
OuterOutline	Outer
OuterShadow	▨ clBtnShadow
OuterSpace	0
OuterStyle	Raised
Outerwidth	1
Visible	Yes

The standard Windows **Open** file appears and asks for a video:

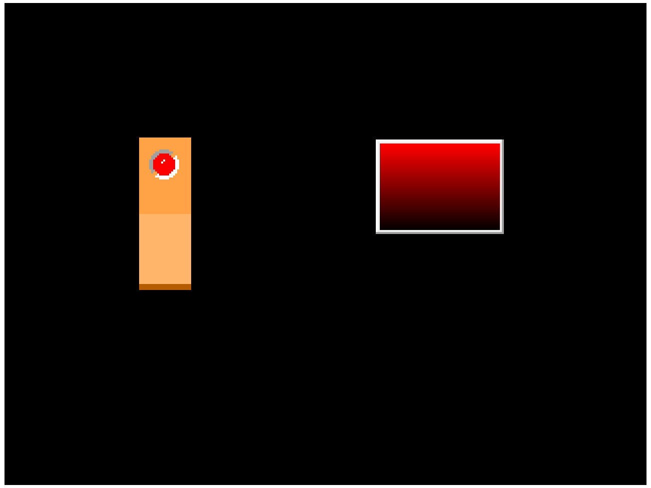
Palette and Colours Options

Two palettes need to be defined, one for each state:

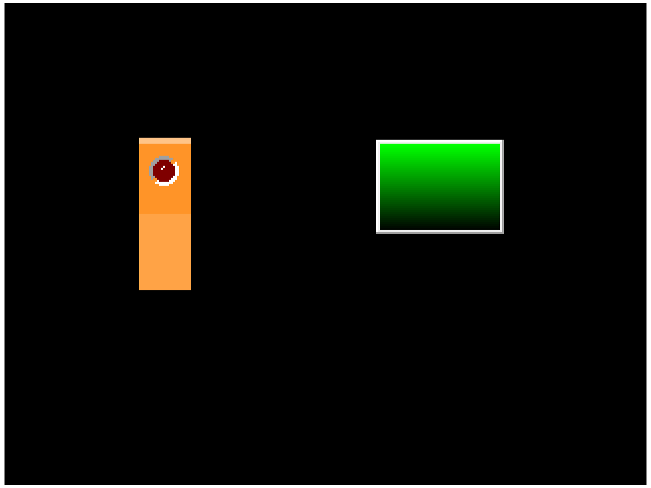


⊟ PaletteEx	
High1	🟥 dRed
High2	⬛ BLACK
Low1	🟩 GREEN
Low2	⬛ BLACK

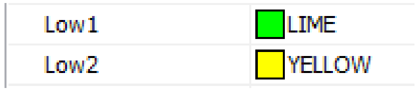
- **High**, when the LED is on:



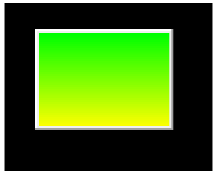
- **Low**, when the LED is off:



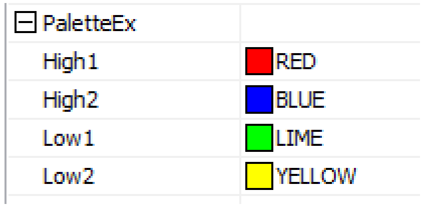
- Each palette consists of two colours, one for the top and the other for the bottom, as in this example:



Low1	🟩 LIME
Low2	🟨 YELLOW



-
- The example includes the following presentation:



⊟ PaletteEx	
High1	🟥 RED
High2	🟦 BLUE
Low1	🟩 LIME
Low2	🟨 YELLOW

-

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).

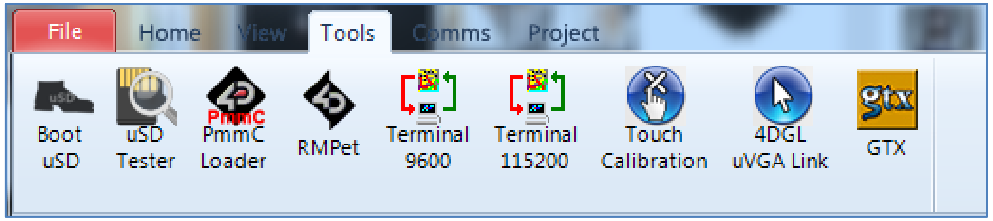
The uLCD-32PTU and/or the uLCD-35DT display modules are commonly used as examples, but the procedure is the same for other displays

Debugger Output

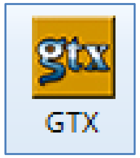
Setting the handler to Message sends the values to the debugger, **Genie Test Executor** or GTX.

Launch the Debugger

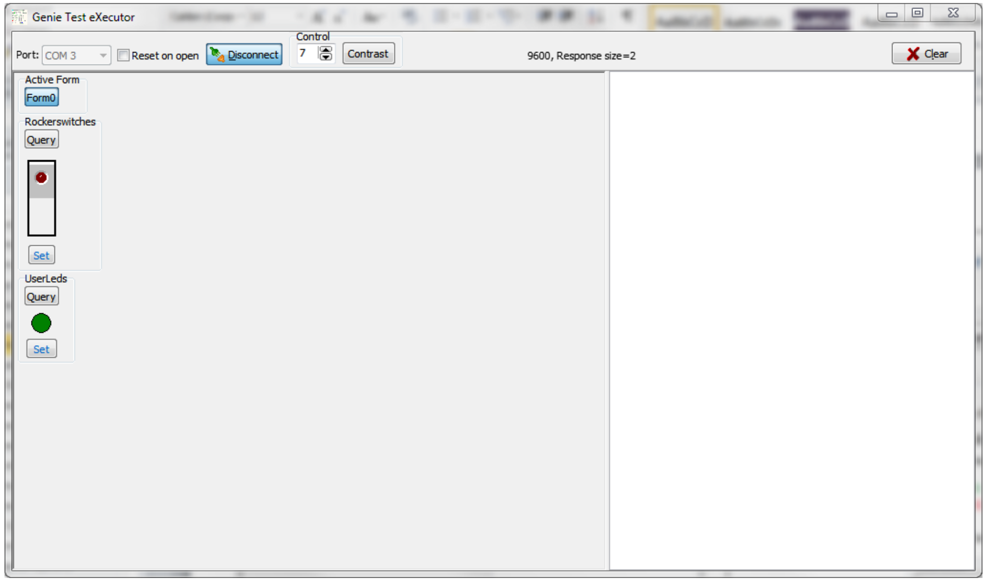
To launch the **Genie Test Executor** or GTX, select the **Tools** menu…



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

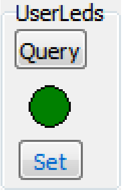


A new screen appears, with the form and objects we have defined previously, here from the LED User example:



Setting the Value of the User LED

The circle is dark green for off:

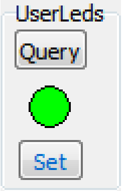


Press Set to send the command to the screen module.

The right part of the GTX window displays the command sent with value **00 00** and the successful acknowledgement **06**:

```
Set Userled Value 14:59:14.517 [01 13 00 00 00 12]
ACK 14:59:14.548 [06]
```

Click on the circle to switch it on: it turns light green.



Press Set to send the command to the screen module.

The right part of the GTX window displays the command sent with value **00 01** and the successful acknowledgement **06**:

```
Set Userled Value 14:59:14.517 [01 13 00 00 01 13]
ACK 14:59:14.548 [06]
```

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.