APPLICATION NOTES

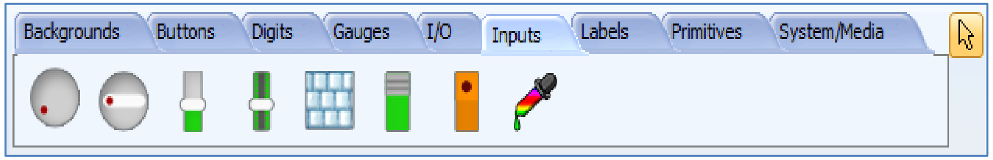
4D SYSTEMS
TURNING TECHNOLOGY INTO ART

ViSi-Genie Inputs

DOCUMENT DATE:	**5th April 2019**
DOCUMENT REVISION:	**1.1**

Description

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



This application note requires:

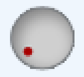
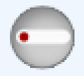
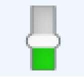
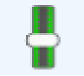
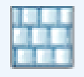
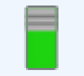
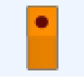
- Workshop 4 has been installed according to the document Workshop 4 Installation;
- The user is familiar with the Workshop 4 environment and with the fundamentals of ViSi-Genie, as described in Workshop 4 User Guide and ViSi-Genie User Guide;
- When downloading an application note, a list of recommended 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.

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

Content

Application Overview

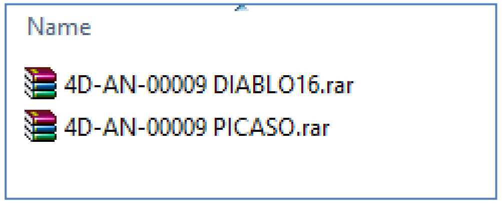
It is often difficult to design a graphical display without being able to see the immediate results of the application code. ViSi-Genie is the perfect software tool that allows the user to see the instant results of their desired graphical layout with this large selection of gauges and meters that can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

-  Knob
-  Rotary switch
-  Slider
-  Track-bar
-  Keyboard
-  DIP-switch
-  Rocker

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-00009 DIABLO16.rar

4D-AN-00009 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).

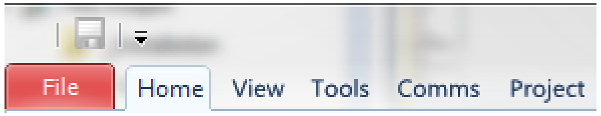
Simulation Procedure

You can load the example…

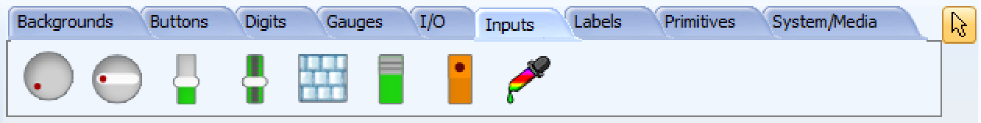
Example: 4D-AN-00009 PICASO – Knob or 4D-AN-00009 DIABLO16 – Knob

…orfollow the procedure described hereafter.

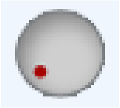
Select the **Home** menu to display the objects:



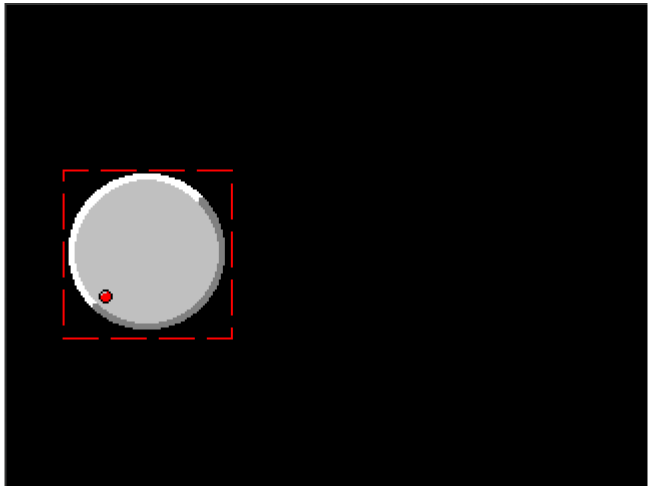
The **Input** object is located on the Inputs pane:



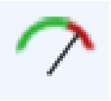
To add an object, first click on the desired icon, here start with the first one, the **Knob**…



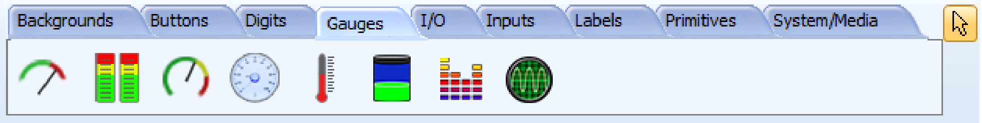
…and then click on the WYSIWYG screen to place it.



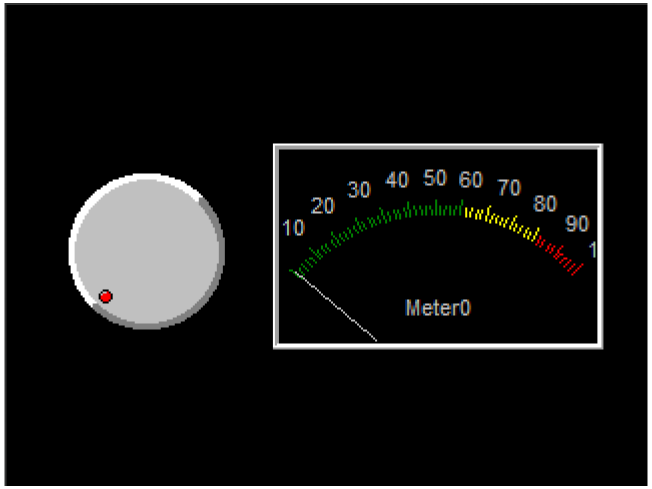
Add a meter as output object…



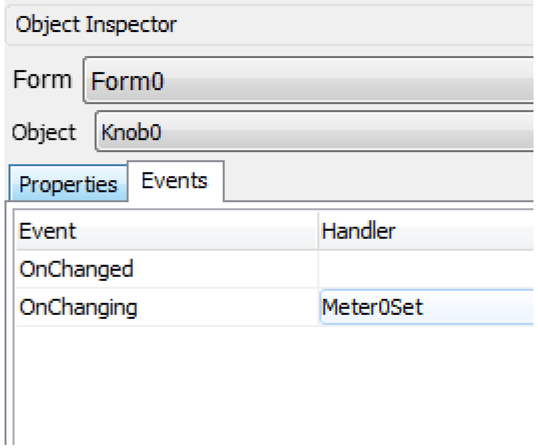
…from the **Gauges** pane…



…and place it on the right of the form:



Define the event **onChanging** for the **Knob0** as **Meter0Set**:



Object Inspector

Form: Form0

Object: Knob0

Properties | Events

Event	Handler
OnChanged	
OnChanging	Meter0Set

Each time the knob is rotated, the meter is updated accordingly.

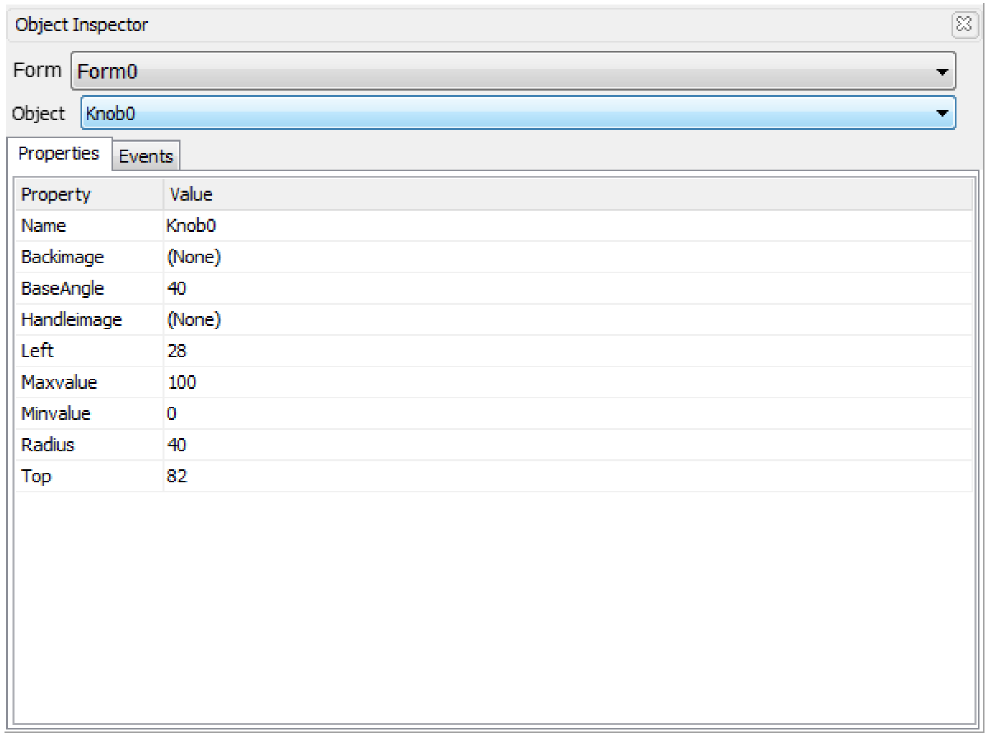
Knob Options

You can load the example…

> ***Example: 4D-AN-00009 PICASO – Knob or 4D-AN-00009 DIABLO16 – Knob***

…or follow the procedure described hereafter.

The options are listed on the Object Inspector:



Object Inspector

Form: Form0

Object: Knob0

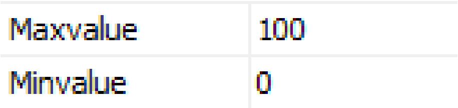
Properties | Events

Property	Value
Name	Knob0
Backimage	(None)
BaseAngle	40
Handleimage	(None)
Left	28
Maxvalue	100
Minvalue	0
Radius	40
Top	82

Standard options are font, colour, size, back image… are not going to be discussed here.

Minimum and Maximum

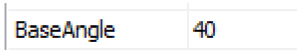
Minimum and maximum are defined by:



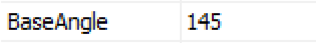
Maxvalue	100
Minvalue	0

Base Angle

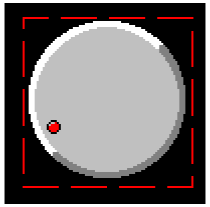
The base angle sets the point 0 for the knob:

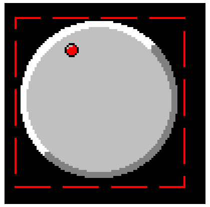


BaseAngle	40



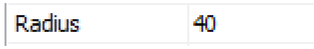
BaseAngle	145



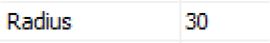


Radius

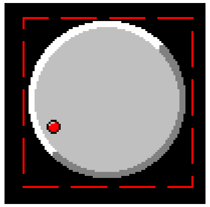
The radius defines the position of the red dot:

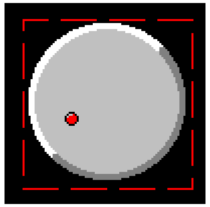


Radius	40



Radius	30





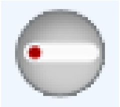
Rotary Switch Options

You can load the example…

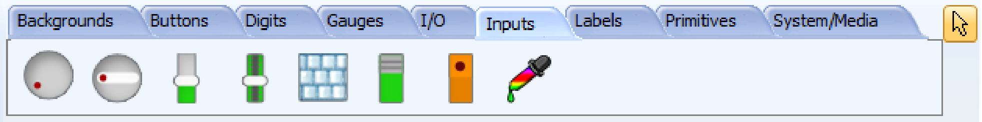
Example: 4D-AN-00009 PICASO – Rotary Switch or 4D-AN-00009 DIABLO16 – Rotary Switch

…or follow the procedure described hereafter.

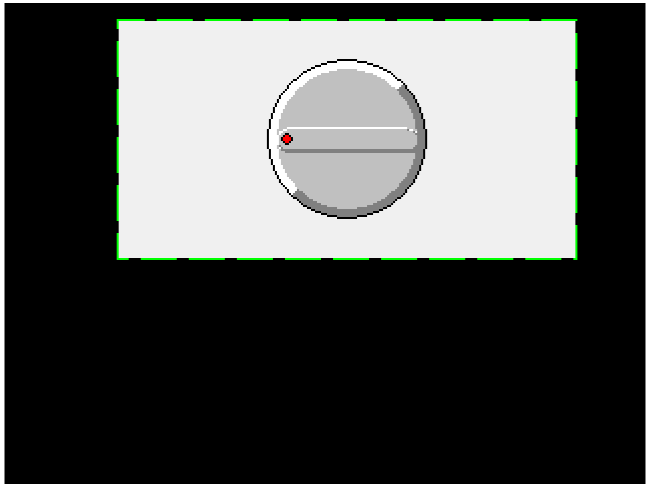
Start with a new project and select the Rotary Switch…



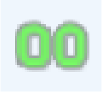
…from the Inputs pane…



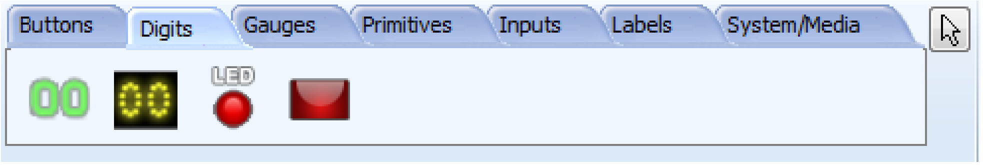
…and then click on the WYSIWYG screen to place it.



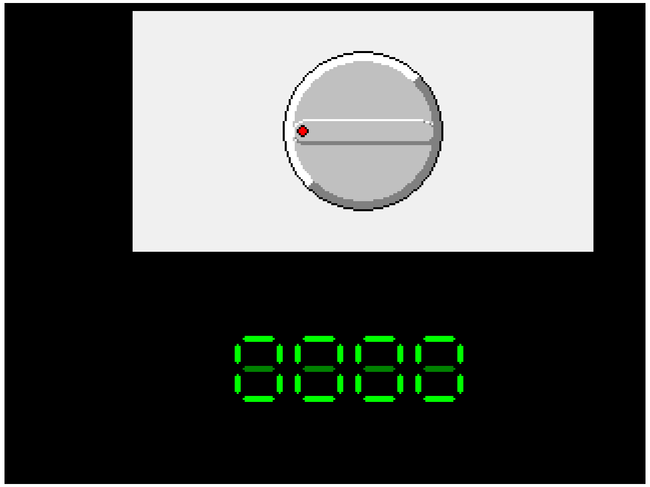
Add a LED digits object as output object…



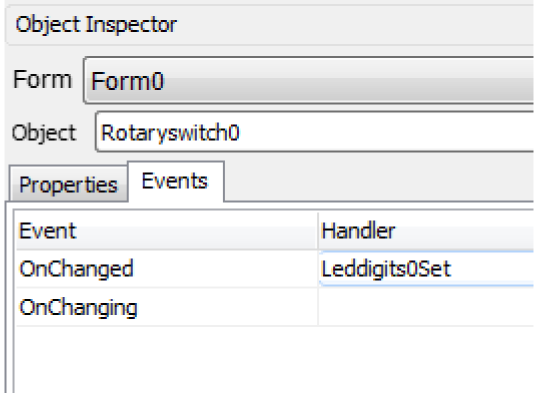
…from the **Digits** pane…



…and place it on the bottom of the form:



Define the event **onChanging** for the **RotarySwitch0** as **LedDigits0Set**:



Object Inspector

Form: Form0

Object: Rotaryswitch0

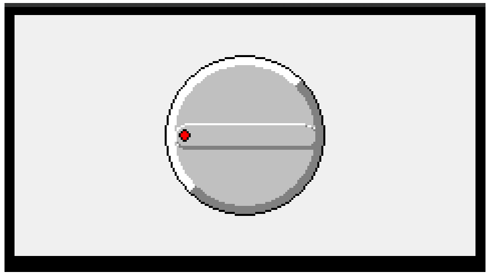
Properties | Events

Event	Handler
OnChanged	Leddigits0Set
OnChanging	

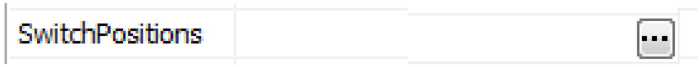
Each time the rotary switch is changed, the LED digits are updated accordingly.

Define the Positions

The rotary switch has no position, so the first step is to define positions.

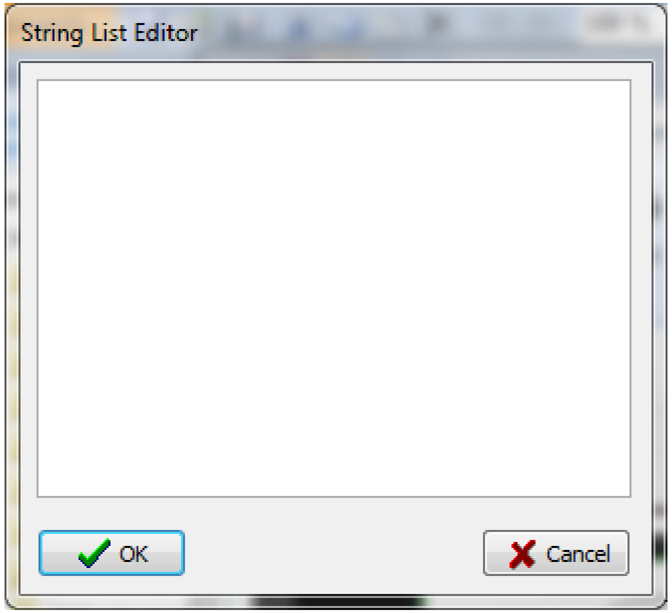


Select the line **SwitchPositions**…



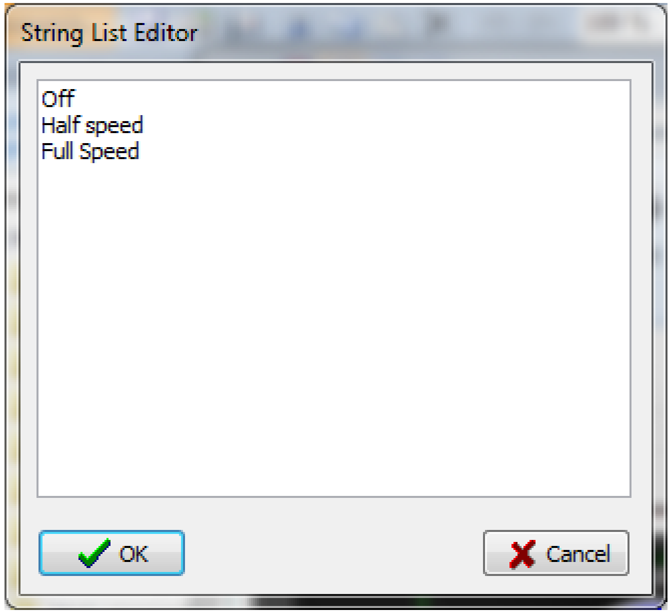
SwitchPositions ...

…and click on the ... to display a new window:



The labels of the different positions of the rotary switch are defined in the **String List Editor** window.

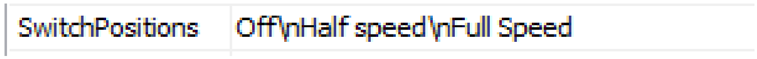
Enter each position on a separate line…



Off
Half speed
Full Speed

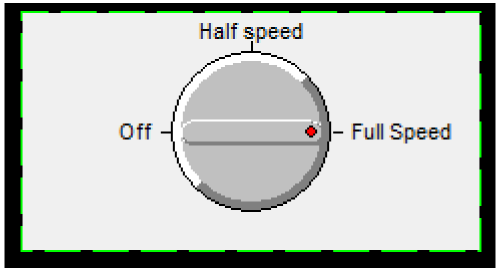
…and click ✓ OK to confirm.

The SwitchPositions line is updated with the values…



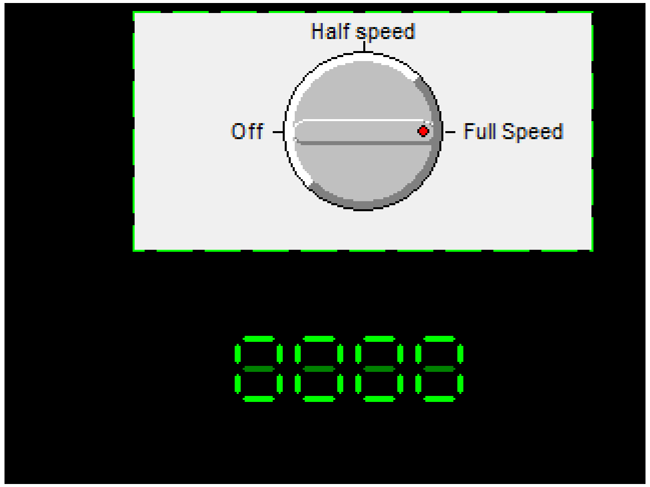
SwitchPositions Off\nHalf speed\nFull Speed

…as the rotary switch on the form:

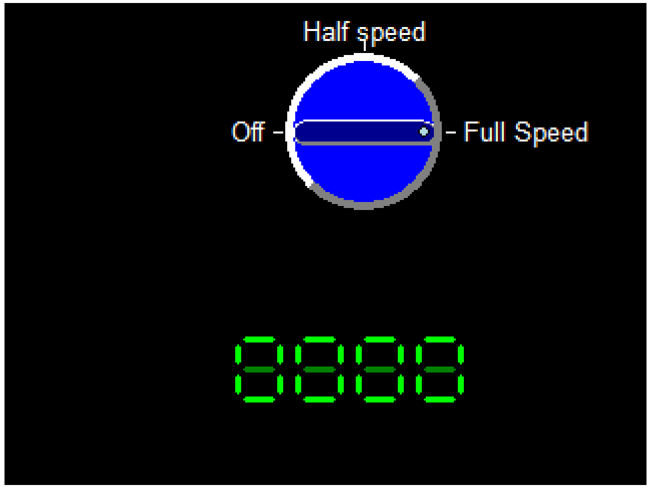


Improve the Interface

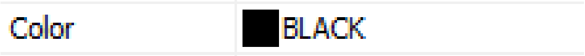
The screen isn't especially nice!



How to reach that nicer one?

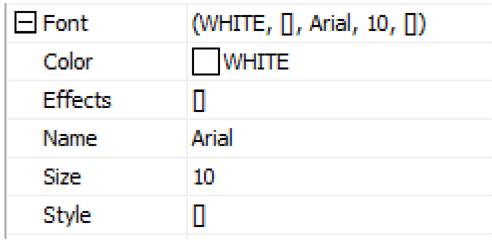


Change first the colour of the back-ground to black…



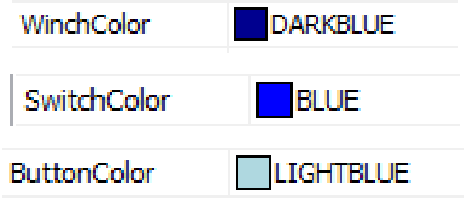
Color	■ BLACK

…then the colour of the text to white…



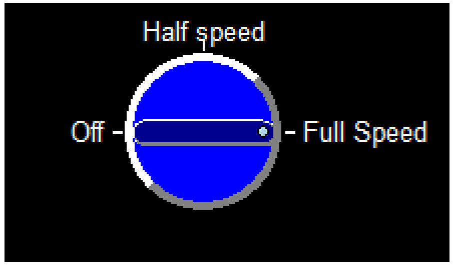
⊟ Font	(WHITE, [], Arial, 10, [])
Color	□ WHITE
Effects	[]
Name	Arial
Size	10
Style	[]

…and finally the colours of the different parts of the rotary switch to shades of blue…



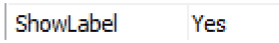
WinchColor	🟦 DARKBLUE
SwitchColor	🟦 BLUE
ButtonColor	□ LIGHTBLUE

…to obtain a much more pleasant interface:

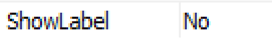


Other Options

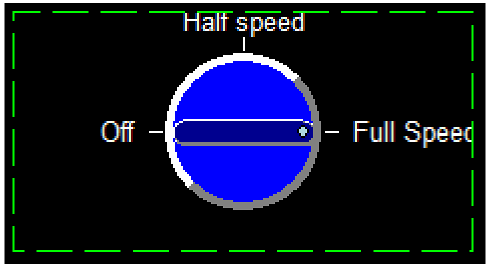
Labels can be hidden:

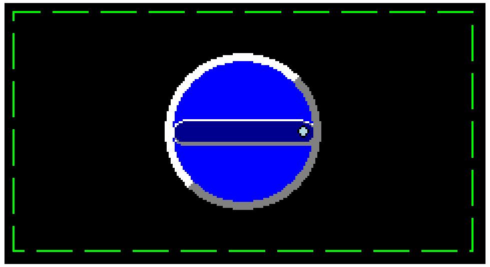


ShowLabel	Yes

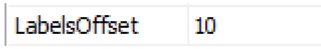


ShowLabel	No

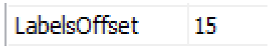




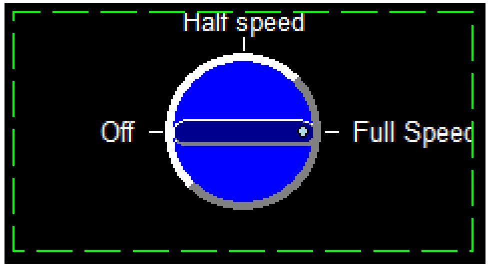
Offset can be changed:

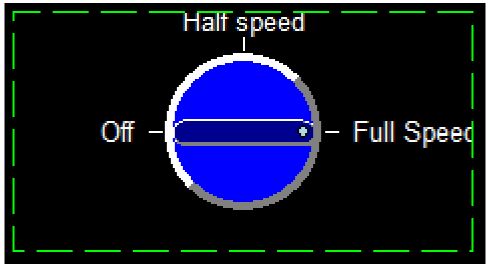


LabelsOffset	10

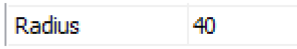


LabelsOffset	15

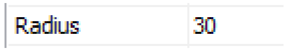




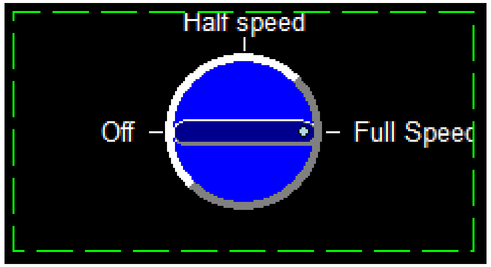
Radius can be modified…

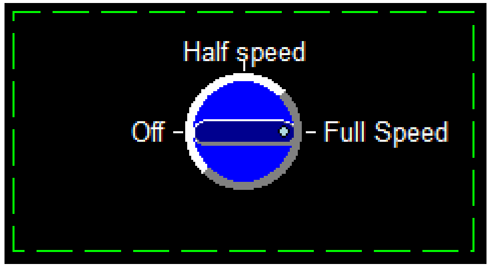


Radius	40

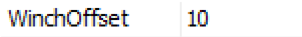


Radius	30

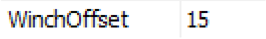




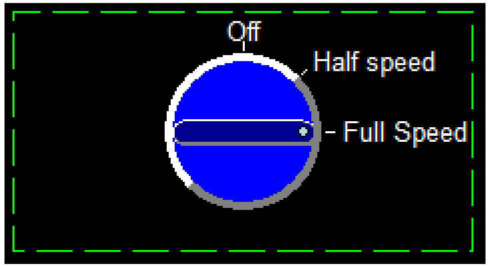
…as the size of the winch:

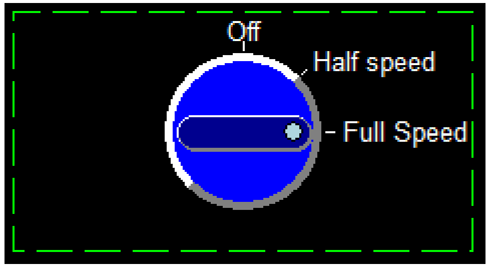


WinchOffset	10



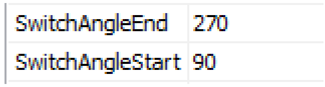
WinchOffset	15



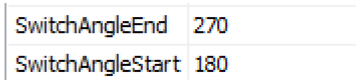


Angle Options

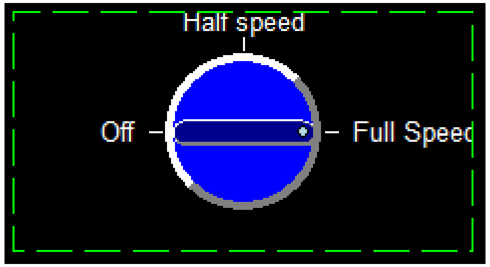
Two angles of the rotary switch can be set:

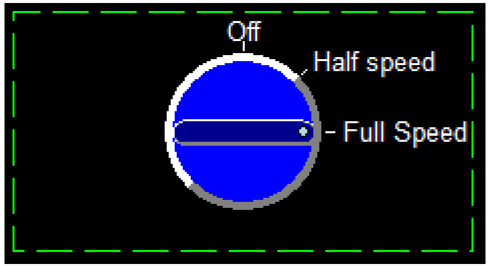


SwitchAngleEnd	270
SwitchAngleStart	90

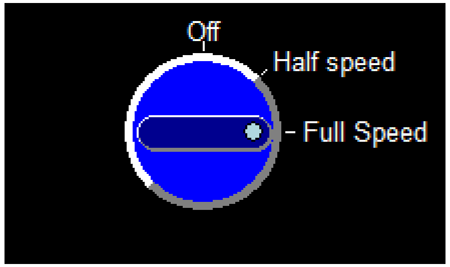


SwitchAngleEnd	270
SwitchAngleStart	180

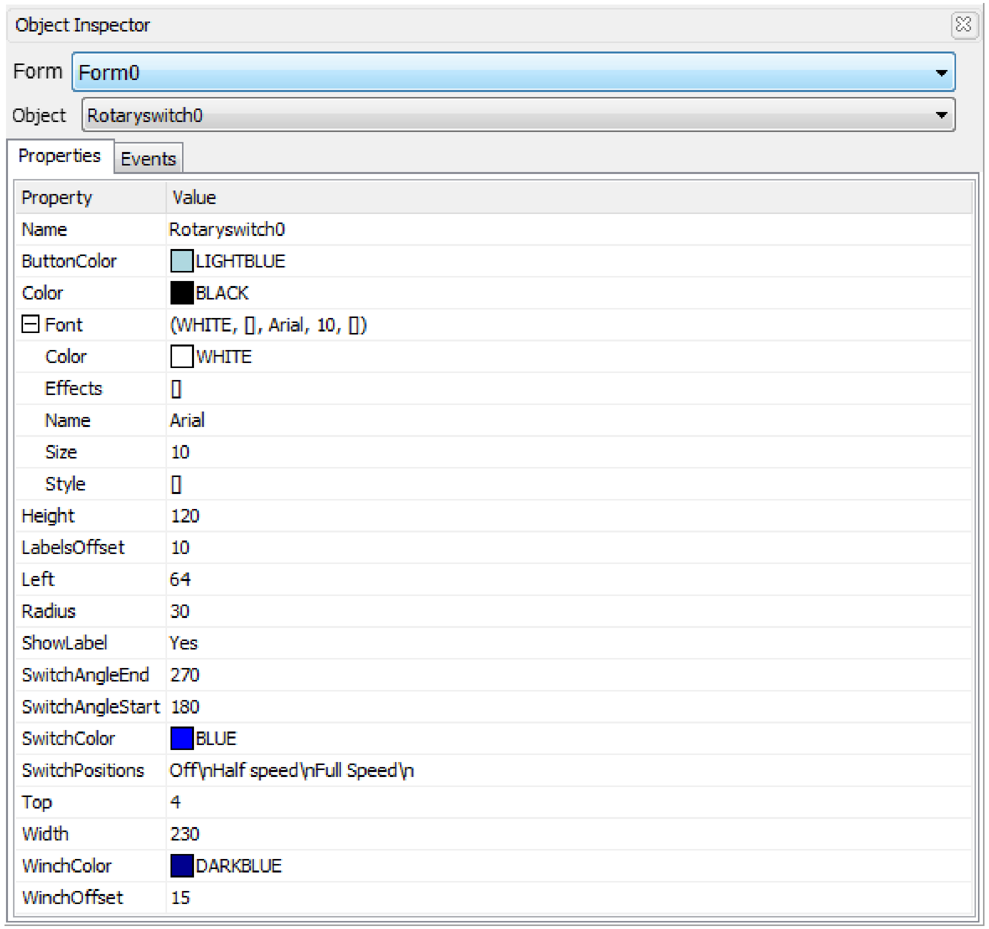




Here is the final result…



…with the properties associated:



Form Form0

Object Rotaryswitch0

Property	Value
Name	Rotaryswitch0
ButtonColor	LIGHTBLUE
Color	BLACK
⊟ Font	(WHITE, [], Arial, 10, [])
Color	WHITE
Effects	[]
Name	Arial
Size	10
Style	[]
Height	120
LabelsOffset	10
Left	64
Radius	30
ShowLabel	Yes
SwitchAngleEnd	270
SwitchAngleStart	180
SwitchColor	BLUE
SwitchPositions	Off\nHalf speed\nFull Speed\n
Top	4
Width	230
WinchColor	DARKBLUE
WinchOffset	15

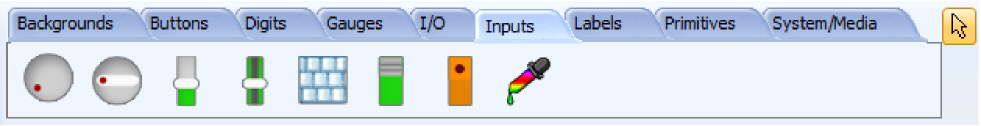
Slider Options

You can load the example…

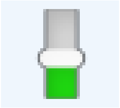
Example: 4D-AN-00009 PICASO – Slider or 4D-AN-00009 DIABLO16 - Slider

…orfollow the procedure described hereafter.

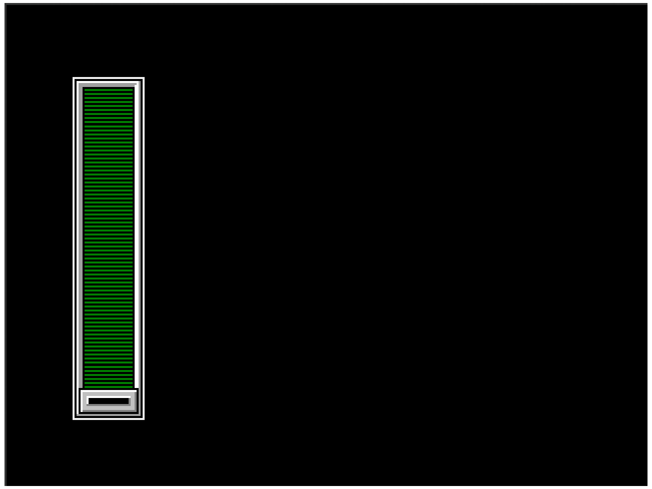
The **Slider** object is located on the Inputs pane:



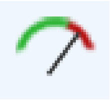
To add an object, first click on the desired icon, here start with the first one, the **Slider**…



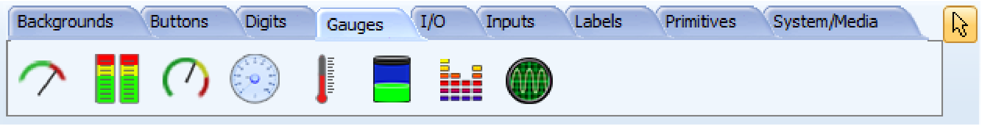
…and then click on the WYSIWYG screen to place it.



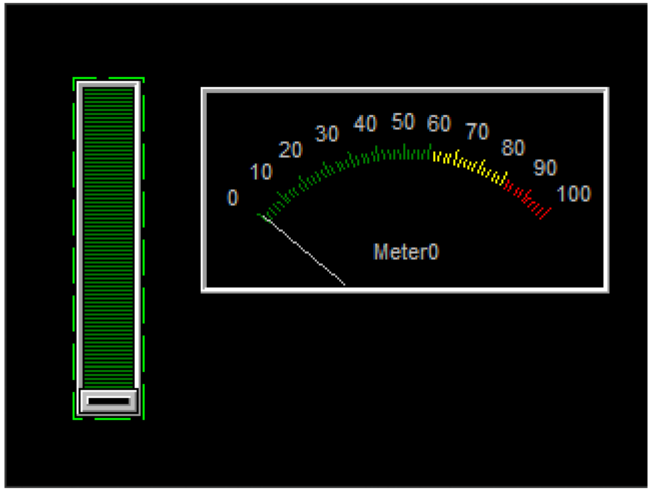
Add a meter as output object…



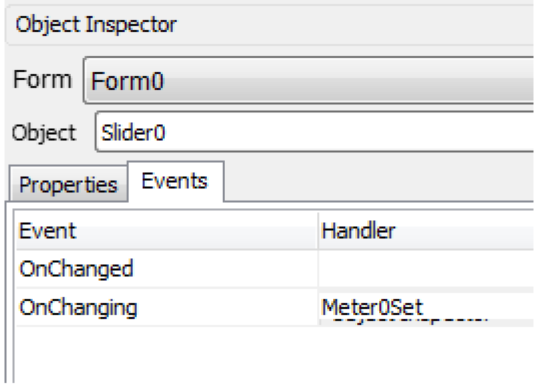
…from the **Gauges** pane…



…and place it on the right of the form:



Define the event **onChanging** for the **Slider0** as **Meter0Set**:



Object Inspector

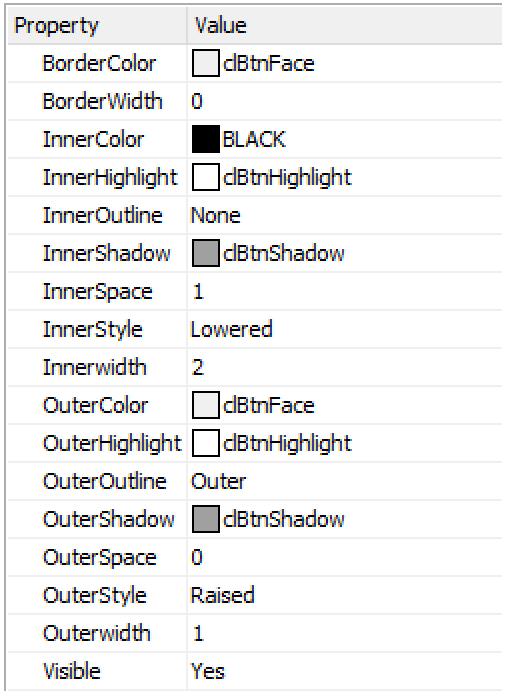
Form: Form0

Object: Slider0

Event	Handler
OnChanged	
OnChanging	Meter0Set

Each time the slider is moved, the meter is updated accordingly.

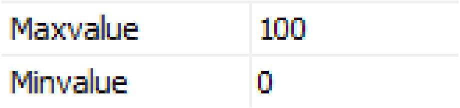
The bevel offers many options to customise its presentation:



Property	Value
BorderColor	□ clBtnFace
BorderWidth	0
InnerColor	■ BLACK
InnerHighlight	□ clBtnHighlight
InnerOutline	None
InnerShadow	▩ clBtnShadow
InnerSpace	1
InnerStyle	Lowered
Innerwidth	2
OuterColor	□ clBtnFace
OuterHighlight	□ clBtnHighlight
OuterOutline	Outer
OuterShadow	▩ clBtnShadow
OuterSpace	0
OuterStyle	Raised
Outerwidth	1
Visible	Yes

Minimum and Maximum

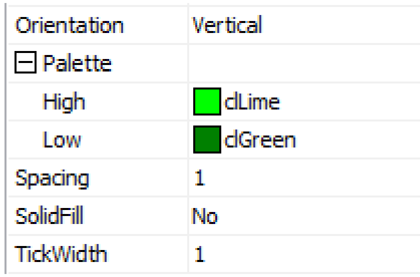
Minimum and maximum are defined by:



Maxvalue	100
Minvalue	0

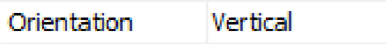
Slider Options

The options related to the slider include:

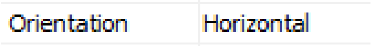


Orientation	Vertical
⊟ Palette	
High	🟩 clLime
Low	🟩 clGreen
Spacing	1
SolidFill	No
TickWidth	1

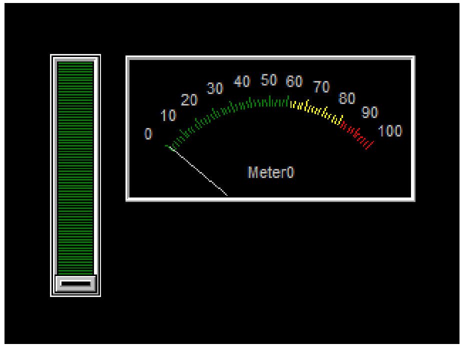
Orientation can be vertical or horizontal:

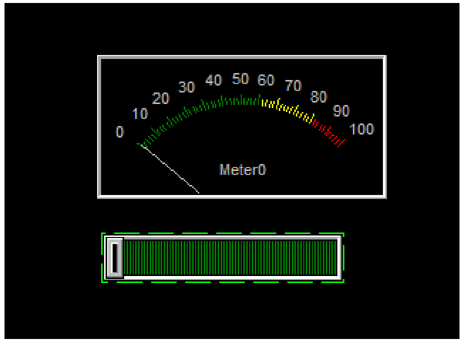


Orientation	Vertical

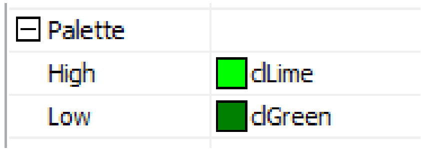


Orientation	Horizontal





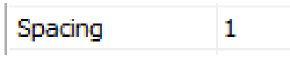
There are two colours to define: one for high, another for low:



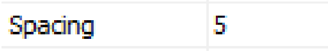
⊟ Palette	
High	🟩 clLime
Low	🟩 clGreen

Choosing two contrasted colours provides an instant feed-back on the position selected.

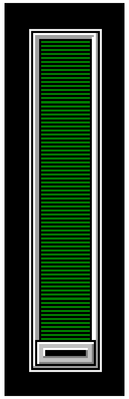
Spacing can be adjusted…

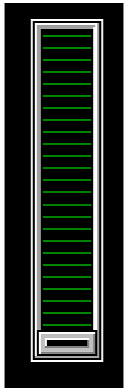


Spacing	1

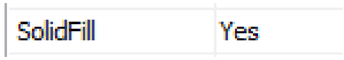


Spacing	5

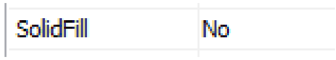




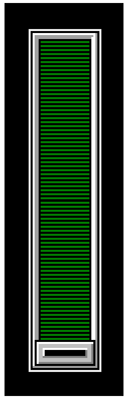
…and set to solid or no…

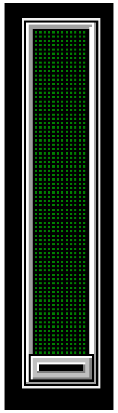


SolidFill	Yes

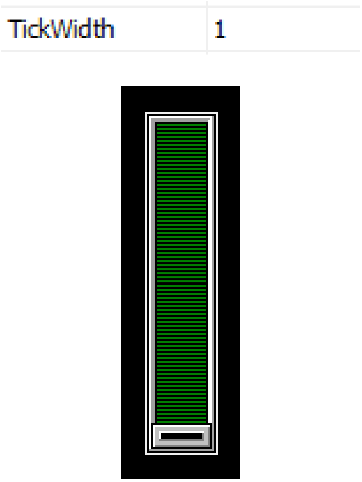


SolidFill	No

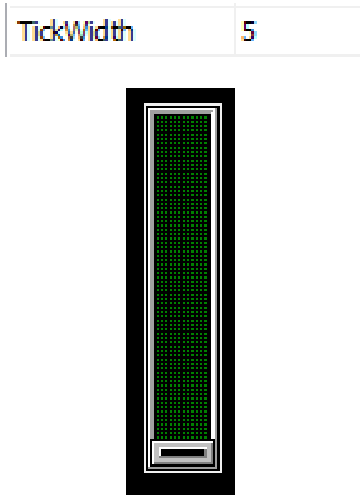




...and ticks size changed:



TickWidth	1



TickWidth	5

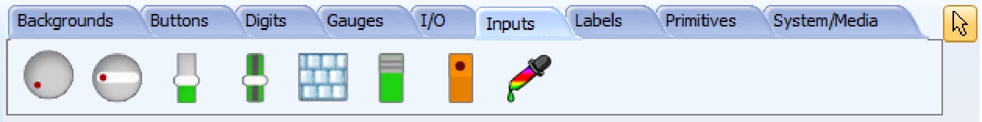
Track-Bar Options

You can load the example...

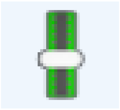
Example: 4D-AN-00009 PICASO – Track Bar or 4D-AN-00009 DIABLO16 – Track Bar

...orfollow the procedure described hereafter.

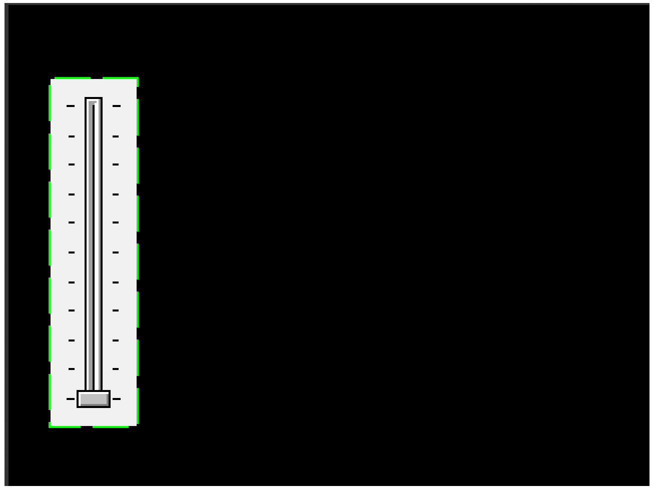
The **Track-bar** object is located on the Inputs pane:



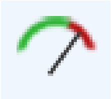
To add an object, first click on the desired icon, here start with the first one, the **TrackBar**...



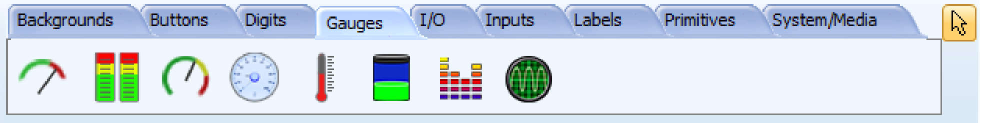
...and then click on the WYSIWYG screen to place it.



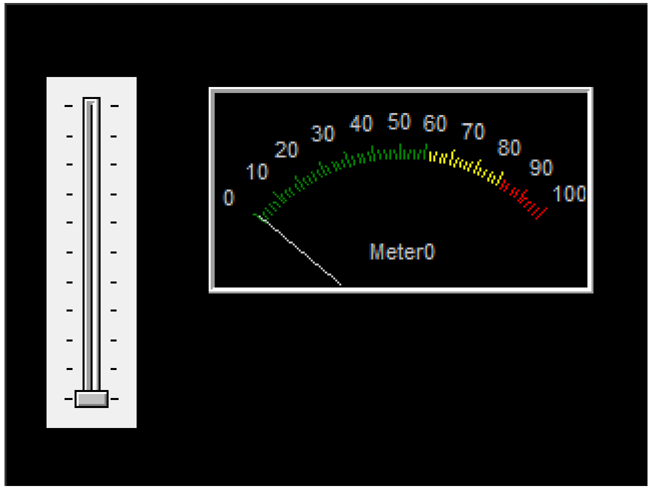
Add a meter as output object...



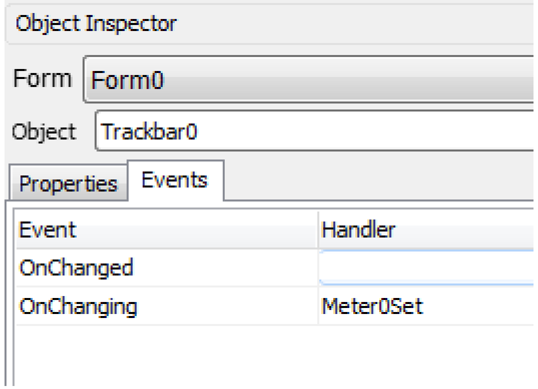
...from the **Gauges** pane...



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



Define the event **onChanging** for the **TrackBar0** as **Meter0Set**:



Object Inspector

Form Form0

Object Trackbar0

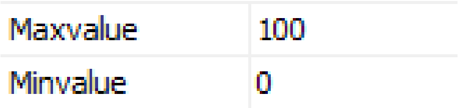
Properties Events

Event	Handler
OnChanged	
OnChanging	Meter0Set

Each time the track-bar is moved, the meter is updated accordingly.

Minimum and Maximum

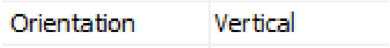
Minimum and maximum are defined by:



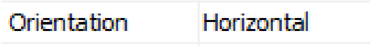
Maxvalue	100
Minvalue	0

Orientation Option

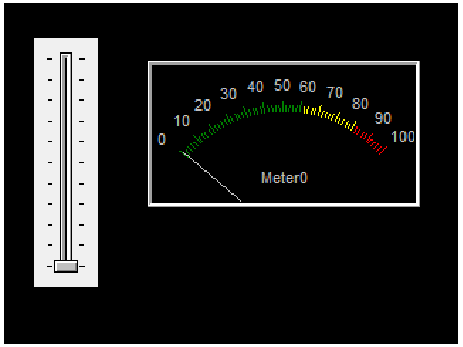
Orientation can be vertical or horizontal:

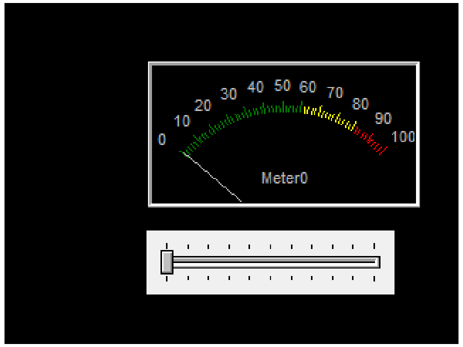


Orientation	Vertical



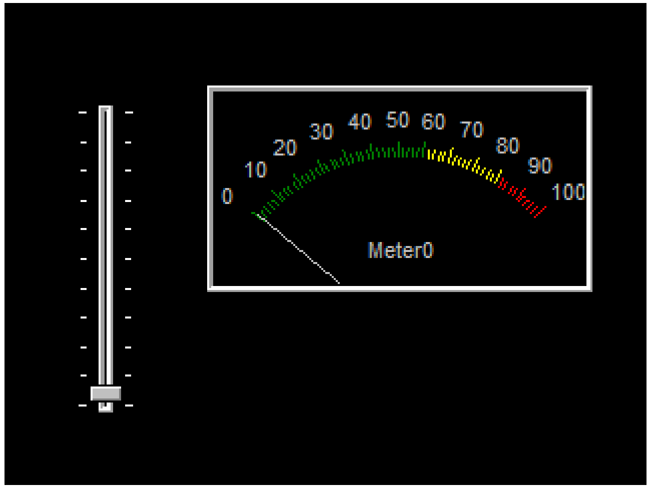
Orientation	Horizontal



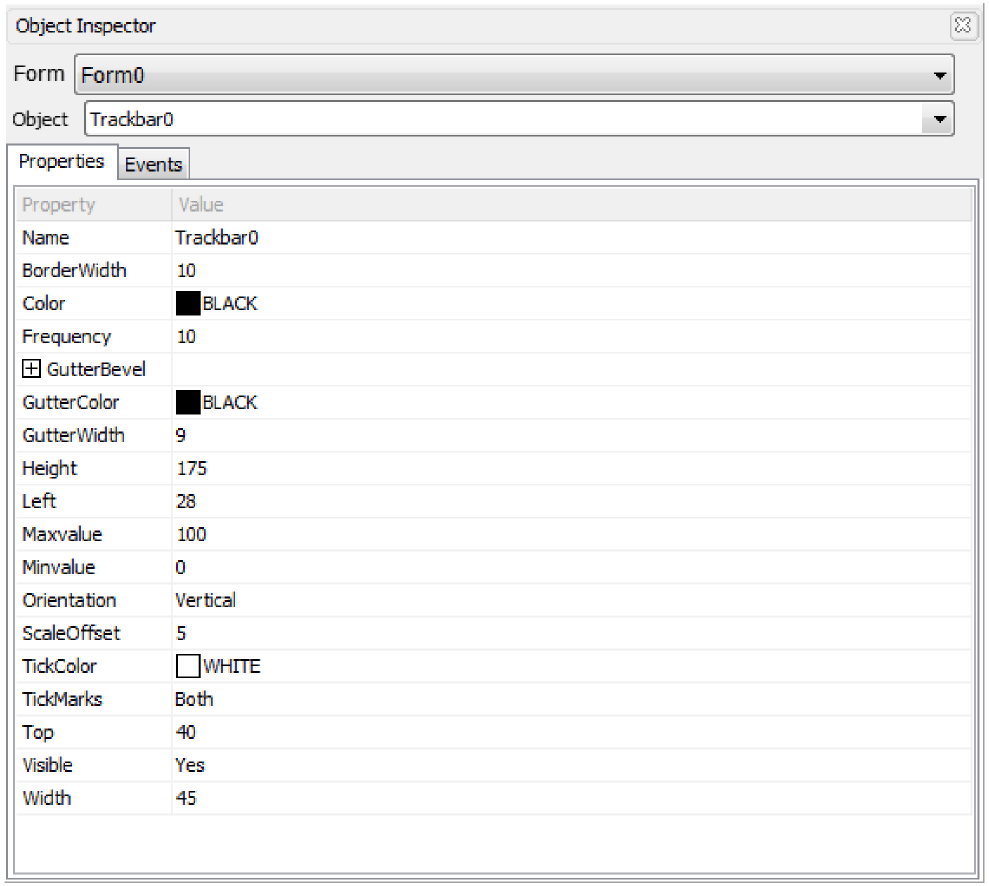


Colours Options

Changing the colours options to obtain this...



…requires coordinating the colour of the object, **Color**, and the colour of the ticks, **TickColor**:



Object Inspector

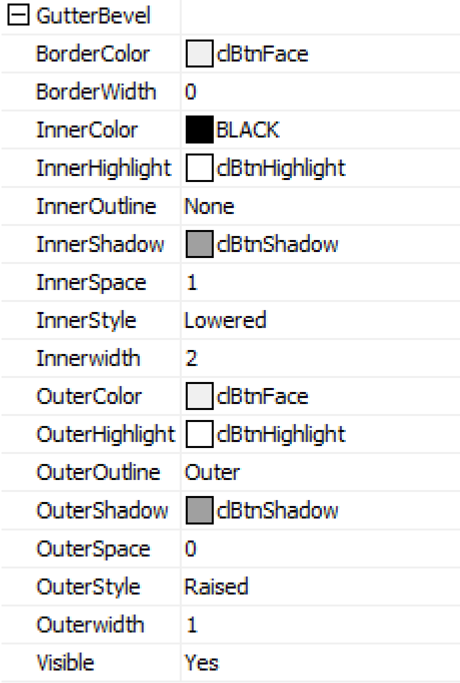
Form: Form0

Object: Trackbar0

Property	Value
Name	Trackbar0
BorderWidth	10
Color	■ BLACK
Frequency	10
⊞ GutterBevel	
GutterColor	■ BLACK
GutterWidth	9
Height	175
Left	28
Maxvalue	100
Minvalue	0
Orientation	Vertical
ScaleOffset	5
TickColor	□ WHITE
TickMarks	Both
Top	40
Visible	Yes
Width	45

Gutter Options

The gutter offers many options to customise its presentation:



⊟ GutterBevel	
BorderColor	□ clBtnFace
BorderWidth	0
InnerColor	■ BLACK
InnerHighlight	□ clBtnHighlight
InnerOutline	None
InnerShadow	▨ clBtnShadow
InnerSpace	1
InnerStyle	Lowered
Innerwidth	2
OuterColor	□ clBtnFace
OuterHighlight	□ clBtnHighlight
OuterOutline	Outer
OuterShadow	▨ clBtnShadow
OuterSpace	0
OuterStyle	Raised
Outerwidth	1
Visible	Yes

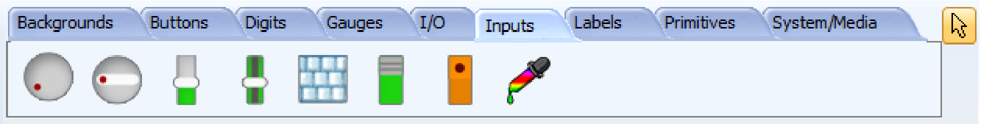
Keyboard Options

You can load the example…

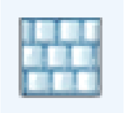
Example: 4D-AN-00009 PICASO – Keyboard or 4D-AN-00009 DIABLO16 - Keyboard

…orfollow the procedure described hereafter.

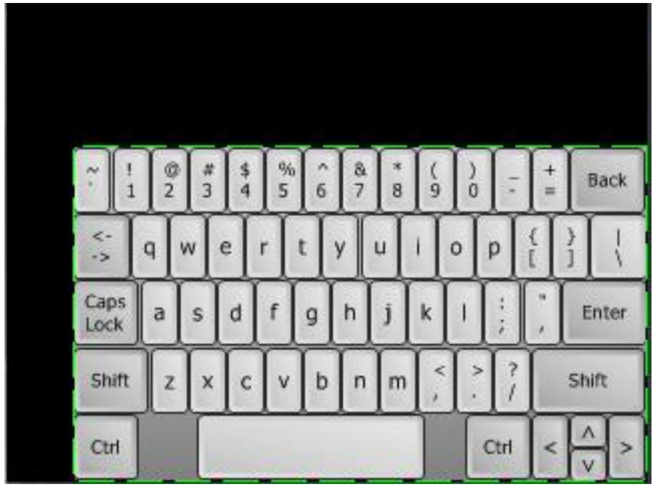
To add a keyboard, select the **Inputs** pane…



…then the **Keyboard** object…

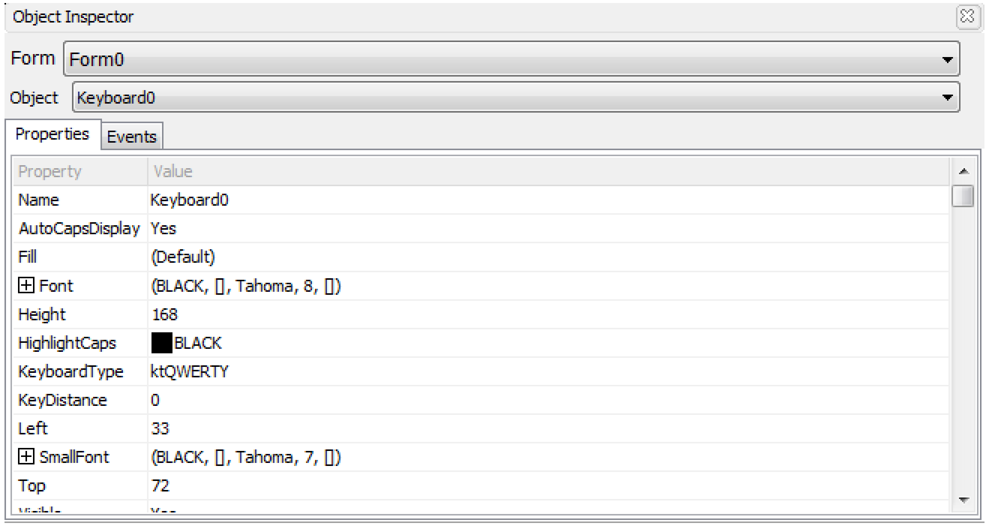


…and finally click on the screen to place it:



Select the Keyboard Type

The object inspector shows all the properties of the keyboard:



Object Inspector

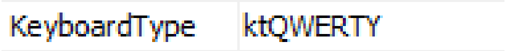
Form: Form0

Object: Keyboard0

Properties | Events

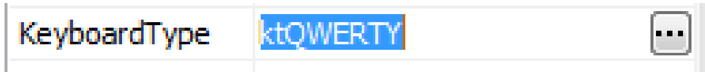
Property	Value
Name	Keyboard0
AutoCapsDisplay	Yes
Fill	(Default)
⊞ Font	(BLACK, [], Tahoma, 8, [])
Height	168
HighlightCaps	■ BLACK
KeyboardType	ktQWERTY
KeyDistance	0
Left	33
⊞ SmallFont	(BLACK, [], Tahoma, 7, [])
Top	72
Visible	Yes

To select the keyboard, we need to change the keyboard type:



KeyboardType	ktQWERTY

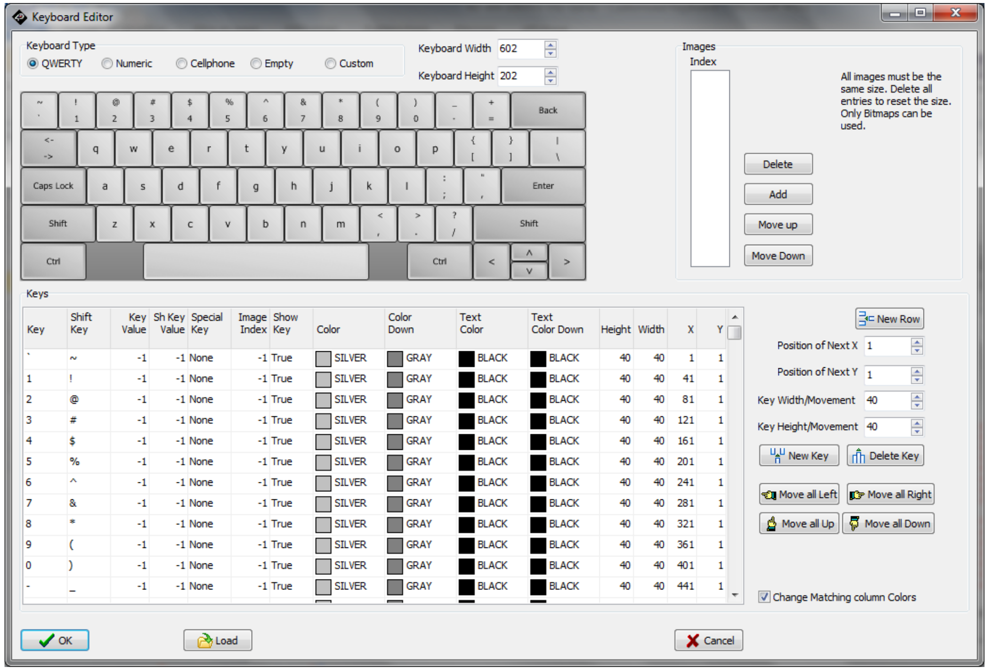
To do so, click on the **KeyboardType** line:



KeyboardType	ktQWERTY	…

A … symbol appears. Click on it.

A new window appears: the Keyboard Editor.



Keyboard Editor

Keyboard Type: ● QWERTY ○ Numeric ○ Cellphone ○ Empty ○ Custom

Keyboard Width 602

Keyboard Height 202

Images

Index

All images must be the same size. Delete all entries to reset the size. Only Bitmaps can be used.

Keys

Key	Shift Key	Key Value	Sh Key Value	Special Key	Image Index	Show Key	Color	Color Down	Text Color	Text Color Down	Height	Width	X	Y
`	~	-1	-1	None	-1	True	SILVER	GRAY	BLACK	BLACK	40	40	1	1
1	!	-1	-1	None	-1	True	SILVER	GRAY	BLACK	BLACK	40	40	41	1
2	@	-1	-1	None	-1	True	SILVER	GRAY	BLACK	BLACK	40	40	81	1
3	#	-1	-1	None	-1	True	SILVER	GRAY	BLACK	BLACK	40	40	121	1
4	$	-1	-1	None	-1	True	SILVER	GRAY	BLACK	BLACK	40	40	161	1
5	%	-1	-1	None	-1	True	SILVER	GRAY	BLACK	BLACK	40	40	201	1
6	^	-1	-1	None	-1	True	SILVER	GRAY	BLACK	BLACK	40	40	241	1
7	&	-1	-1	None	-1	True	SILVER	GRAY	BLACK	BLACK	40	40	281	1
8	*	-1	-1	None	-1	True	SILVER	GRAY	BLACK	BLACK	40	40	321	1
9	(	-1	-1	None	-1	True	SILVER	GRAY	BLACK	BLACK	40	40	361	1
0	)	-1	-1	None	-1	True	SILVER	GRAY	BLACK	BLACK	40	40	401	1
-	_	-1	-1	None	-1	True	SILVER	GRAY	BLACK	BLACK	40	40	441	1

Position of Next X 1

Position of Next Y 1

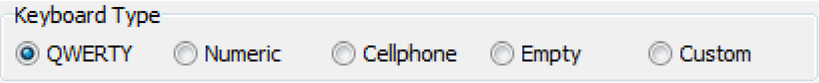
Key Width/Movement 40

Key Height/Movement 40

☑ Change Matching column Colors

Delete
Add
Move up
Move Down
New Row
New Key
Delete Key
Move all Left
Move all Right
Move all Up
Move all Down
OK
Load
Cancel
~ ` ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 _ - + = Back
<- -> q w e r t y u i o p { [}] | \
Caps Lock a s d f g h j k l : ; " ' Enter
Shift z x c v b n m < , > . ? / Shift
Ctrl Ctrl < ^ v >

On top of the window, different keyboard types are proposed:

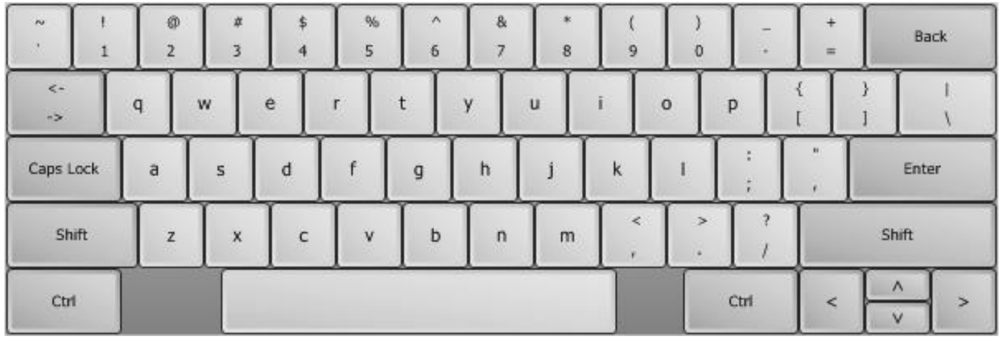


Keyboard Type

● QWERTY ○ Numeric ○ Cellphone ○ Empty ○ Custom

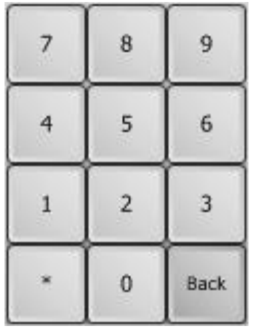
ViSi-Genie comes with various defined ready-to-use keyboards:

- QWERTY keyboard, by default,



~ ` ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 _ - + = Back
<- -> q w e r t y u i o p { [}] | \
Caps Lock a s d f g h j k l : ; " ' Enter
Shift z x c v b n m < , > . ? / Shift
Ctrl Ctrl < ^ v >

- Cell-phone keyboard

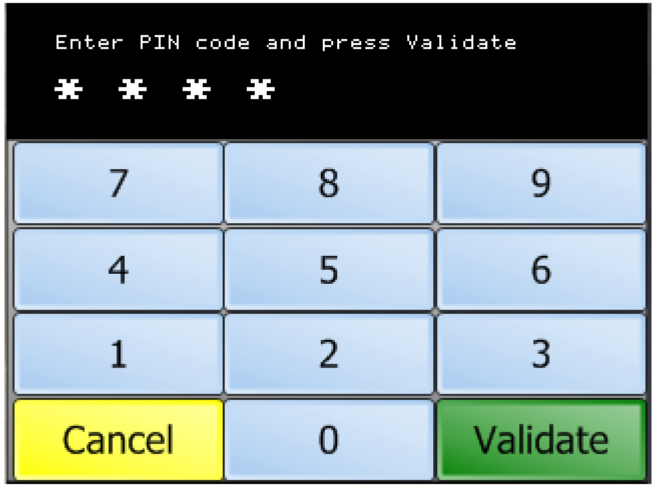


- Numeric keyboard

- And a customised keyboard.

Customise the Keyboard

The customisation of the keyboard is detailed in the application note **ViSi-Genie Customised Keyboard** with a step-by-step guide to define a PIN code keyboard:



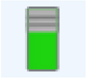
DIP-Switch Options

You can load the example…

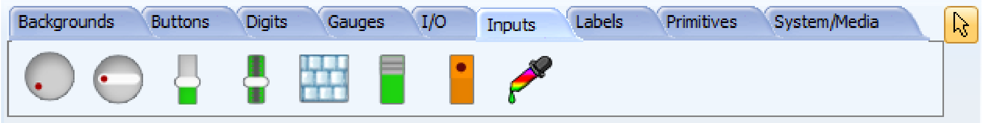
Example: 4D-AN-00009 PICASO – DIP Switch or 4D-AN-00009 DIABLO16 – DIP Switch

…orfollow the procedure described hereafter.

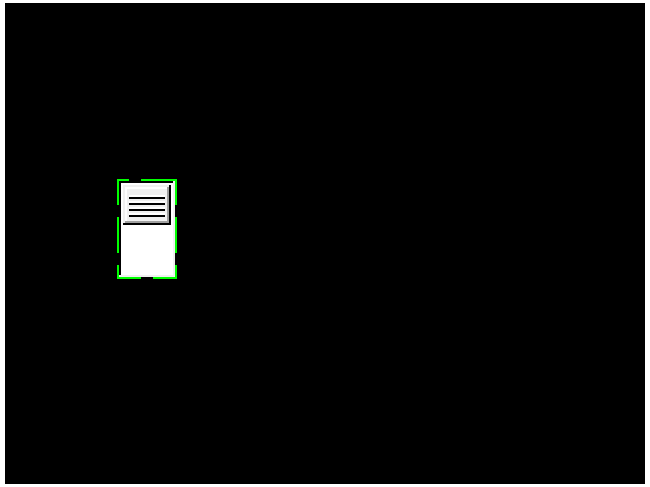
Start with a new project and select the DIP-switch…



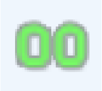
…from the **Inputs** pane…



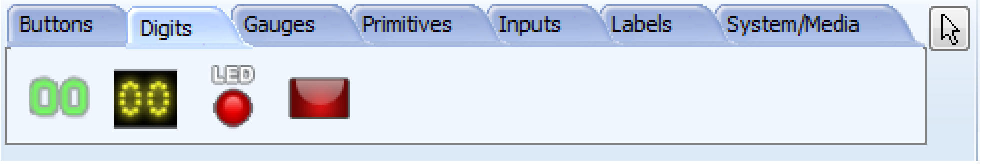
…and then click on the WYSIWYG screen to place it.



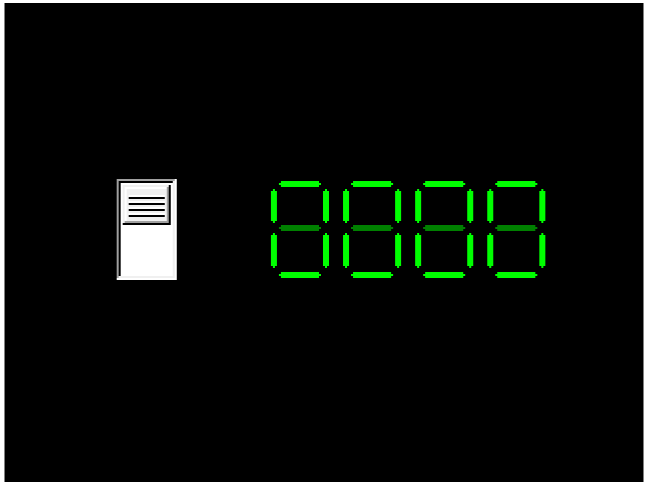
Add a LED digits object as output object…



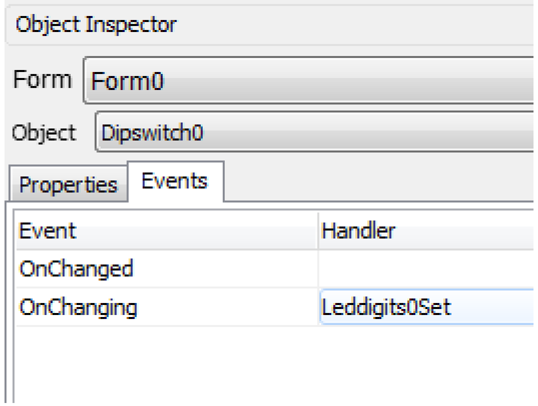
…from the **Digits** pane…



…and place it on the bottom of the form:



Define the event **onChanging** for the **DipSwitch0** as **LedDigits0Set**:



Object Inspector

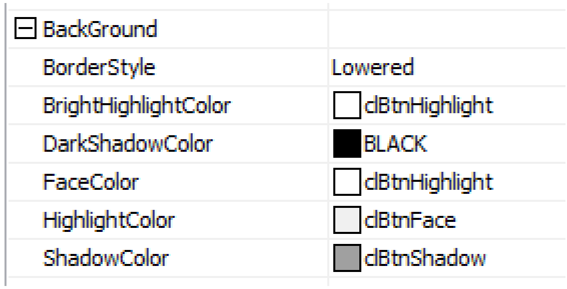
Form: Form0

Object: Dipswitch0

Properties | Events

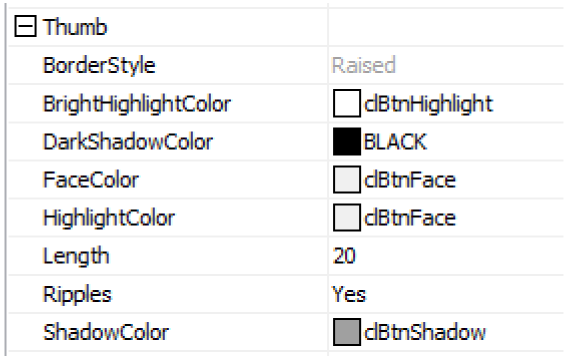
Event	Handler
OnChanged	
OnChanging	Leddigits0Set

Each time the DIP-switch is changed, the LED digits are updated accordingly. The options relative to colours for the different parts of the DIP-switch are defined under **Background**…



⊟ BackGround	
BorderStyle	Lowered
BrightHighlightColor	□ clBtnHighlight
DarkShadowColor	■ BLACK
FaceColor	□ clBtnHighlight
HighlightColor	□ clBtnFace
ShadowColor	▨ clBtnShadow

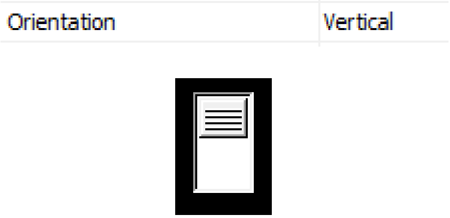
…and **Thumb**:



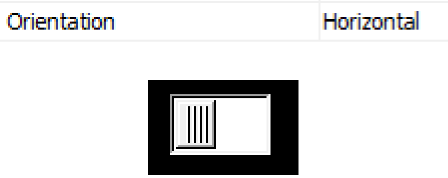
⊟ Thumb	
BorderStyle	Raised
BrightHighlightColor	□clBtnHighlight
DarkShadowColor	■BLACK
FaceColor	□clBtnFace
HighlightColor	□clBtnFace
Length	20
Ripples	Yes
ShadowColor	▨clBtnShadow

Orientation Option

The DIP-switch can be vertical or horizontal:



Orientation	Vertical

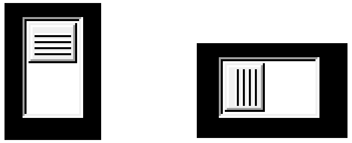


Orientation	Horizontal

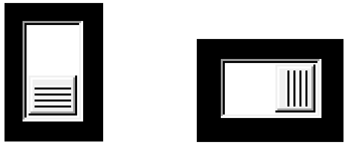
Define the Orientation

The DIP-switch has two positions by default, with:

- 0 as value when the switch is on topmost of leftmost position

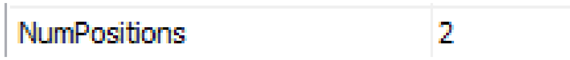


- 1 as value when the switch is on bottommost or rightmost position



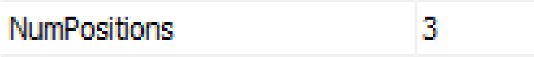
Define the Positions

To define the number of positions, select the line **NumPositions**…



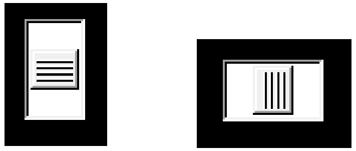
NumPositions	2

…and enter the number of positions desired, here 3:

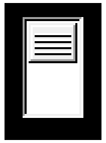
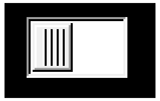
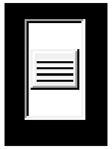
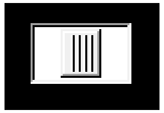
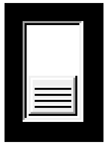
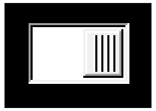


NumPositions	3

This adds an intermediate position:



Numbering is always the same, starting at 0 when the switch is on topmost or leftmost position, and increasing when going down or right:

Value	Vertical	Horizontal
0		
1		
2		

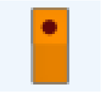
Rocker Options

You can load the example…

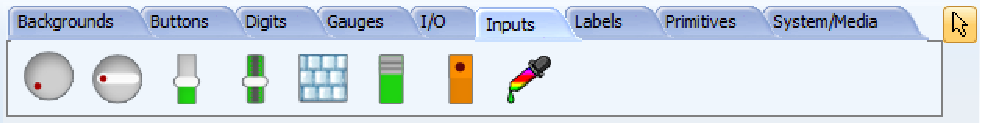
Example: 4D-AN-00009 PICASO – Rocker Switch or 4D-AN-00009 DIABLO16 – Rocker Switch

…or follow the procedure described hereafter.

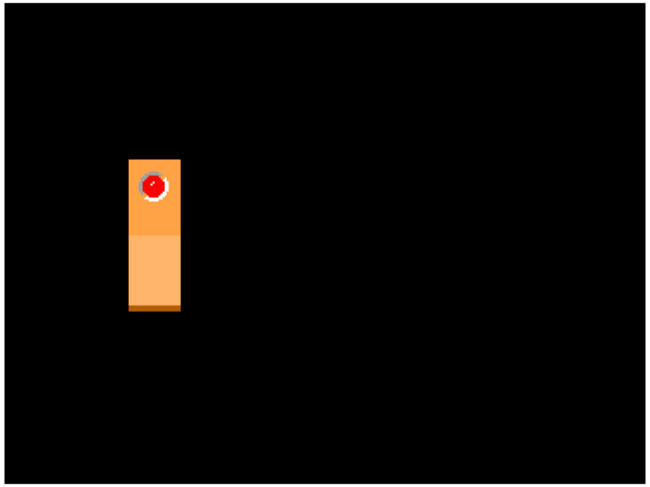
Start with a new project and select the rocker switch…



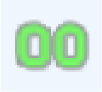
…from the **Inputs** pane…



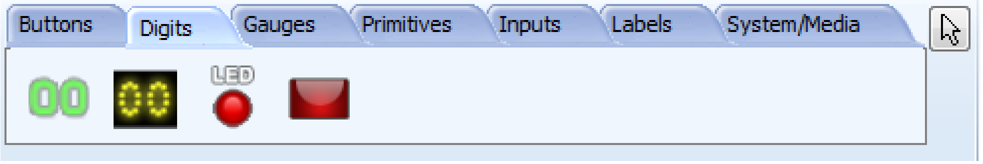
…and then click on the WYSIWYG screen to place it.



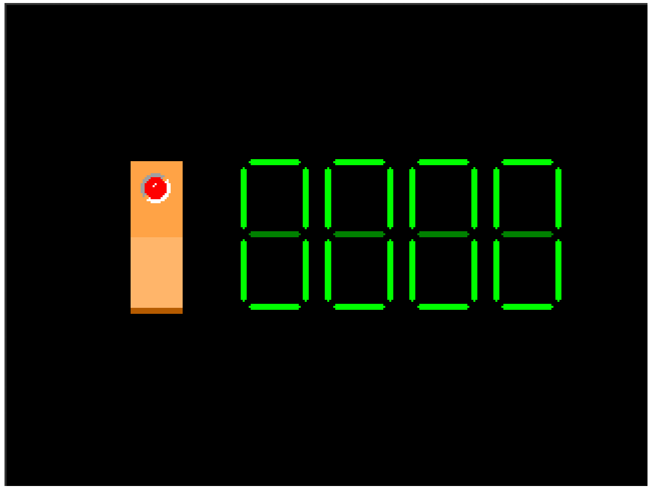
Add a LED digits object as output object…



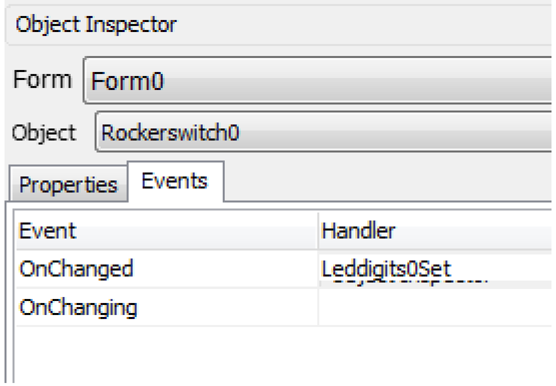
…from the **Digits** pane…



…and place it on the bottom of the form:



Define the event **onChanged** for the **RockerSwitch0** as **LedDigits0Set**:



Object Inspector

Form Form0

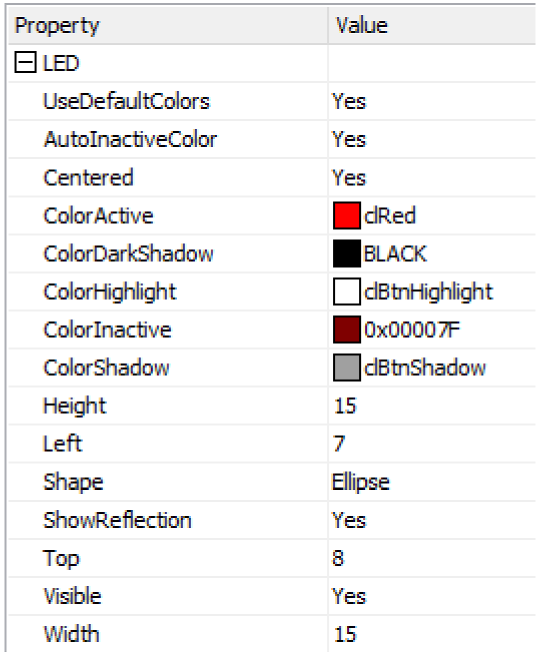
Object Rockerswitch0

Properties Events

Event	Handler
OnChanged	Leddigits0Set
OnChanging	

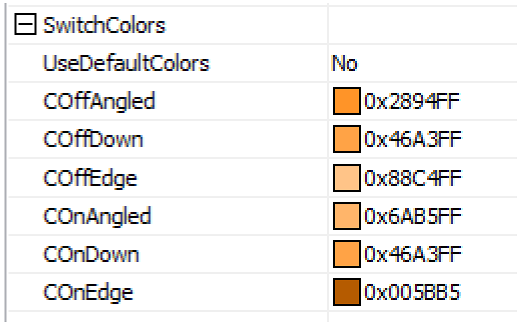
Each time the rocker switch is changed, the LED digits are updated accordingly.

The options relative to colours for the different parts of the rocker switch are defined under **LED**…



Property	Value
⊟ LED	
UseDefaultColors	Yes
AutoInactiveColor	Yes
Centered	Yes
ColorActive	dRed
ColorDarkShadow	BLACK
ColorHighlight	dBtnHighlight
ColorInactive	0x00007F
ColorShadow	dBtnShadow
Height	15
Left	7
Shape	Ellipse
ShowReflection	Yes
Top	8
Visible	Yes
Width	15

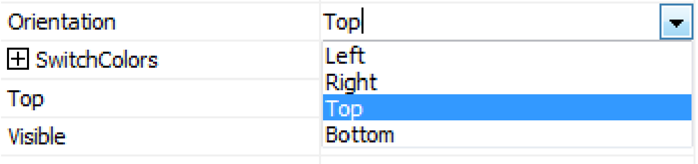
…and **SwitchColours**:



⊟ SwitchColors	
UseDefaultColors	No
COffAngled	0x2894FF
COffDown	0x46A3FF
COffEdge	0x88C4FF
COnAngled	0x6AB5FF
COnDown	0x46A3FF
COnEdge	0x005BB5

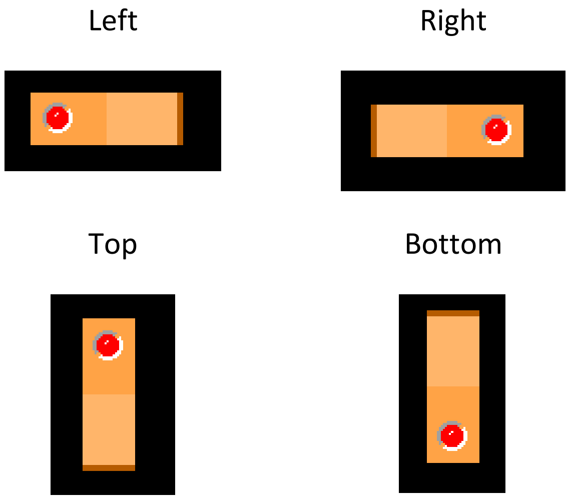
Define the Orientation

The rocker switch has 4 different orientations:

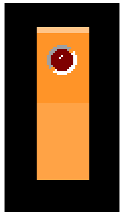
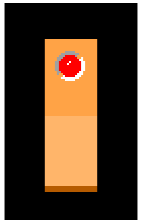


Orientation	Top
⊞ SwitchColors	
Top	
Visible	

The orientation refers to the position of the LED.



The rocker switch has two positions, off and on, and thus two values:

Value	Position	
0		Off
1		On

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).

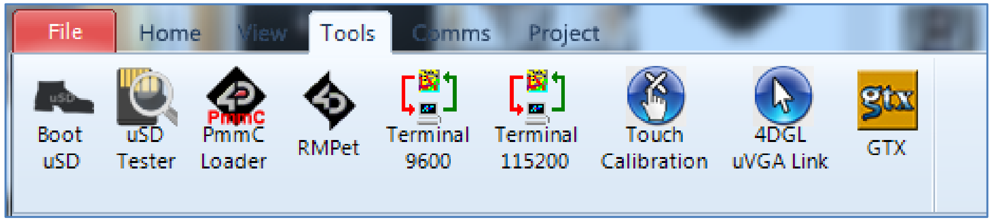
The uLCD-32PTU and/or the uLCD-35DT display modules are commonly used as examples, but the procedure is the same for other displays.

Debugger Output

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

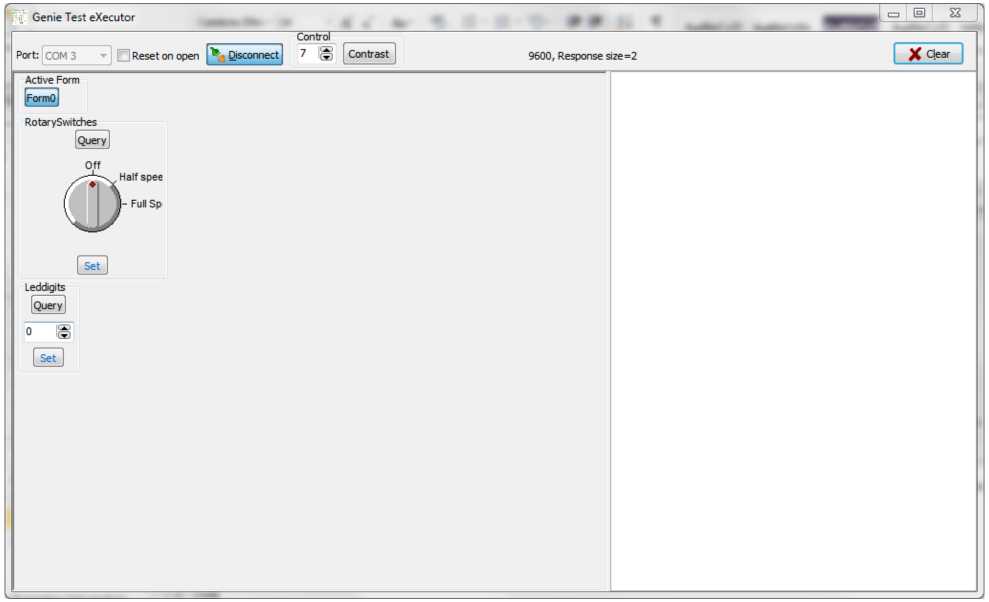
Launch the Debugger

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 on the rotary switch example:



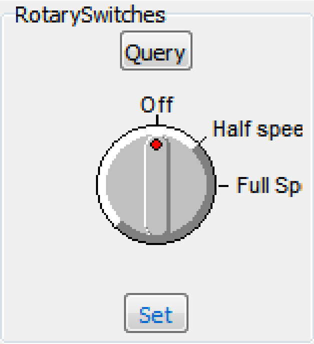
9600, Response size=2

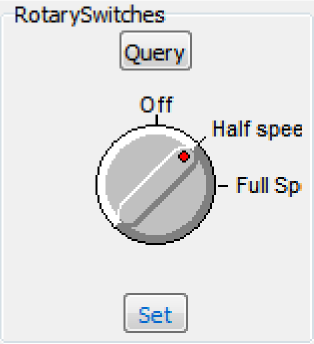
Note the labels are displayed on the debugger window exactly as they are on the screen module.

Use the Debugger

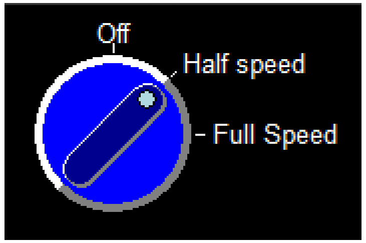
Setting a Value to the Track-Bar

To set a value for the track-bar, move the rotary switch to Half speed…





…and then click on Set. The screen module redraws the rotary switch and the LED digits with the new value:

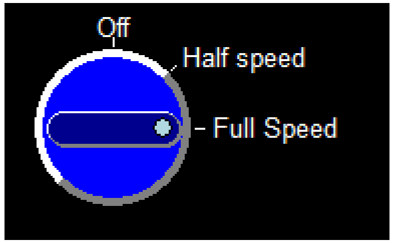


The right part of the GTX window displays the command sent with position **00 01**and the acknowledgement **06**:

```
Set Rotaryswitch Value 14:40:07.929 [01 03 00 00 01 03]
ACK 14:40:08.007 [06]
```

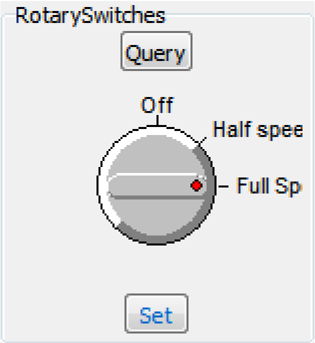
Reading a Value from the Track-Bar

On the screen module, place the rotary switch on the Full speed position:



To read the value of the rotary switch, click now on Query.

The debugger shows the new value corresponding to the second position Full Speed…



…and the right part of the GTX window displays the command sent and the answer, here **0002**:

```
Request Rotaryswitch Value 14:41:36.472 [00 03 00 03]
Rotaryswitch Value 14:41:36.519 [05 03 00 00 02 04]
```

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.