APPLICATION NOTES

4D SYSTEMS
TURNING TECHNOLOGY INTO ART

ViSi Genie Animated Button

DOCUMENT DATE:	29th April 2019
DOCUMENT REVISION:	1.1

W W W . 4 D S Y S T E M S . C O M . A U

Description

This application note provides a first hands-on example with ViSi-Genie and describes all the steps related to a project.

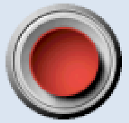
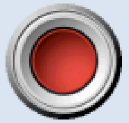
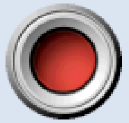
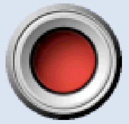
Before getting started, the following are required:

- 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.

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 users to see the instant results of their desired graphical layout with this large selection of gauges and meters (called objects or widgets). The user can simply click on the desired widget to select it and click on the simulated display to place the widget. Below is the animated button widget, an example of the several widgets available in Workshop.

Animated Button Icon		
Animated Button	State 0	
	State 1	
	State 2	
	State 3	
	State 4	

Animated
Button Matrix
Option 1
Option 2
Option 3

The animated button automatically plays a sequence of images when it is pressed and released. The user provides the image to be used for each state and defines the frame delay. The figure to the left shows an animated button with five states or frames.

When configured as momentary, the animated button, when pressed and released, sequentially displays states 0 to 4 then states 4 to 0, with the specified frame delay.

When configured as toggle, the animated button, when pressed and released, plays frames 0 to 4. Another press-and-release is needed for it to play frames 4 back to 0.
When three animated buttons are configured as members of a matrix, only one can be enabled at a time. Enabling a button disables the other two. Animated buttons in a matrix are actually toggle buttons behaving as a group.

The simple project developed in this application note demonstrates the basic use of animated buttons using the three configurations outlined above. Each of these configurations has its own application.

The section **"Identify the Messages"** also discusses the format of the messages coming from and going to an animated button using the GTX Tool in Workshop. An understanding of this section is essential for users who intend to interface the display to an external host.

Setup Procedure

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

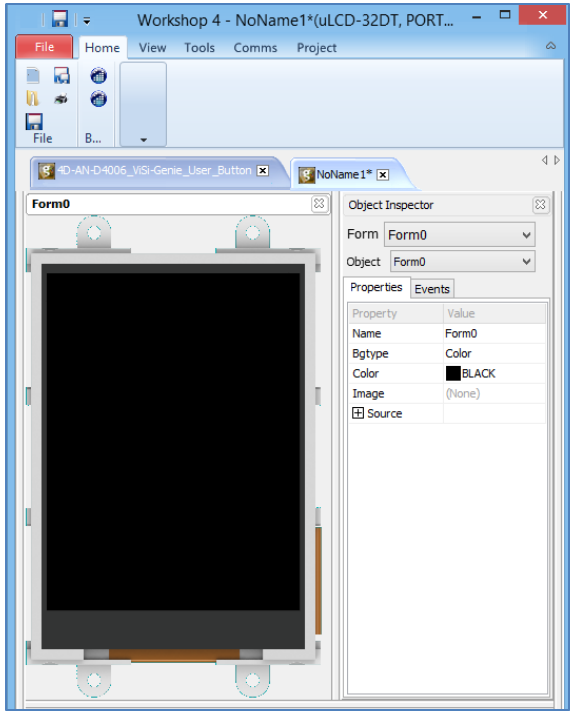
For instructions on how to create a new ViSi-Genie project, please refer to the section "**Create a New Project**" of the application note

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

Design the Project

Everything is now ready to start designing the project. **Workshop 4** displays an empty screen, called **Form0**. A **form** is like a page on the screen. The form can contain **widgets** or **objects** like trackbars, sliders, displays or keyboards.

Below is an empty form.

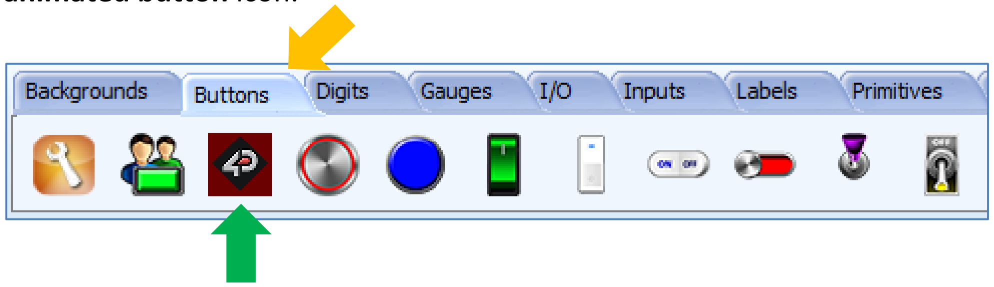


Form: Form0
Object: Form0

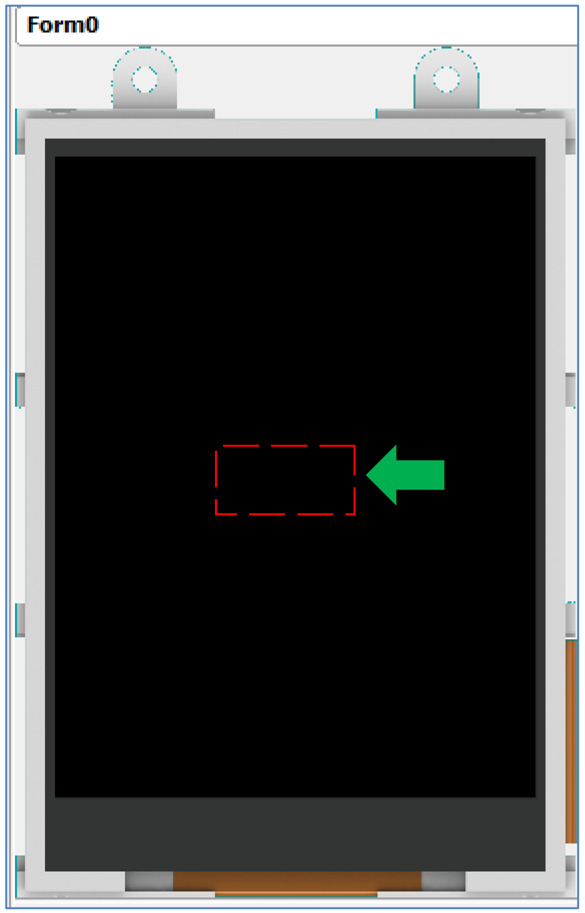
Property	Value
Name	Form0
Bgtype	Color
Color	■ BLACK
Image	(None)
⊞ Source	

Adding a Momentary Animated Button

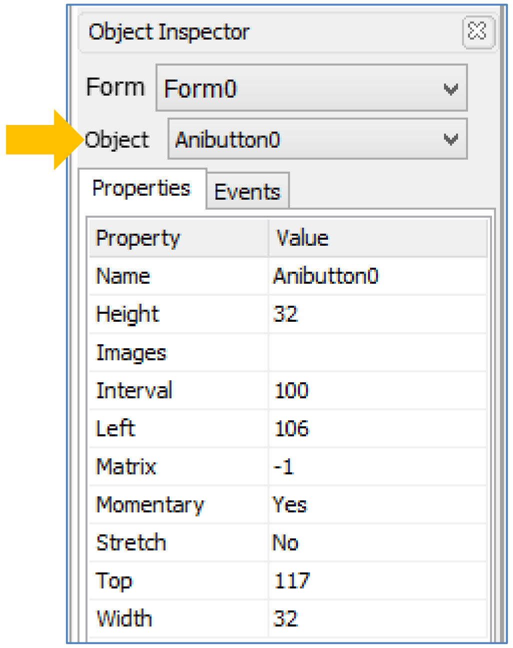
To add an animated button, go to the **Buttons** pane then click on the **animated button** icon.



Click on the **WYSIWYG** (What-You-See-Is-What-You-Get) screen to put the object in place. The WYSIWYG screen simulates the actual appearance of the display module screen.



The object can be dragged to any desired location. The **Object Inspector** on the right part of the screen displays all the properties of the newly created animated button object named **Anibutton0**.



Object Inspector

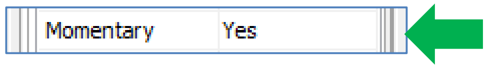
Form: Form0

Object: Anibutton0

Properties | Events

Property	Value
Name	Anibutton0
Height	32
Images	
Interval	100
Left	106
Matrix	-1
Momentary	Yes
Stretch	No
Top	117
Width	32

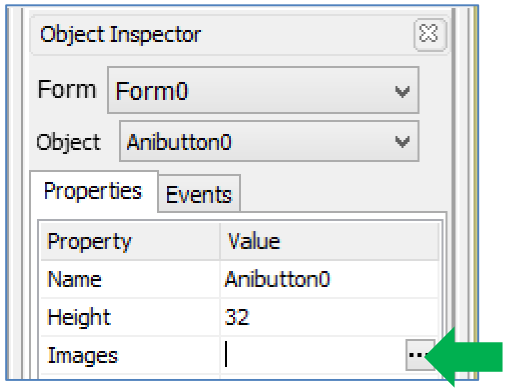
By default, an animated button is a momentary button. Note that the property "Momentary" is set to "Yes".



Momentary	Yes

Defining the States of a Momentary Animated Button

A momentary animated button, when touched and released, sequentially displays all the states from first to last then from last to first with the specified frame delay. It can be conveniently used for navigating between forms, starting and stopping a timer, and navigating between the frames of a video or a user images object. Here the momentary animated button is used to navigate to the next form. To define the states, click on the ellipsis dots of the Images property line of the object inspector.



Object Inspector

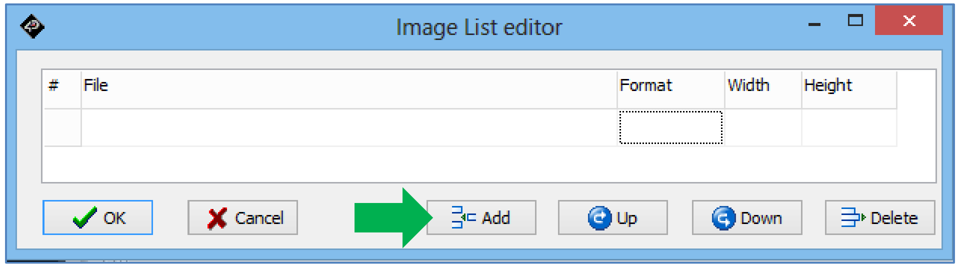
Form: Form0

Object: Anibutton0

Properties | Events

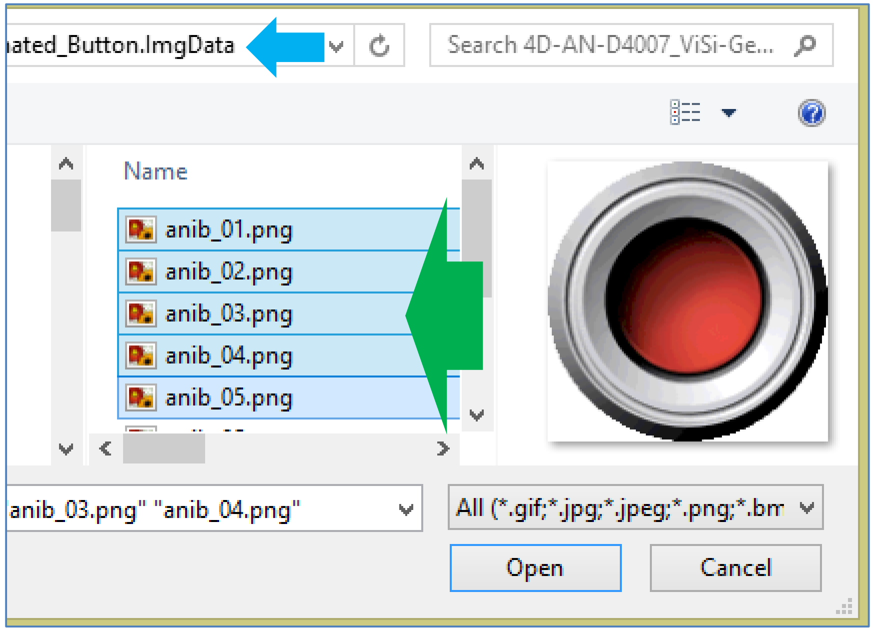
Property	Value
Name	Anibutton0
Height	32
Images	

The Image List editor window appears. Click on the Add button to browse for the image files.

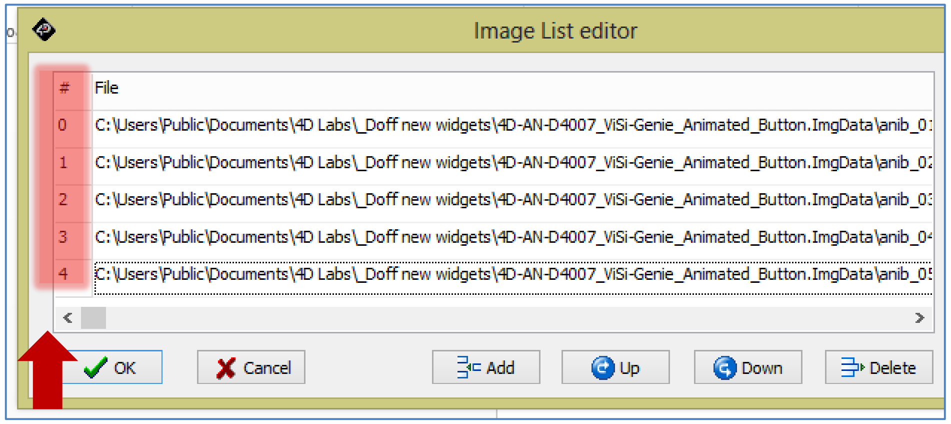


A standard open window appears and asks for image files. Multiple files can be selected. The figure below shows the five image files used for Anibutton0. All the image files used in this demo are saved in the **".ImgData"** folder, which is automatically generated by Workshop when the demo project is compiled. Alternatively the image files are available in the folder

"\Documents\4D Labs\Animated Buttons\Animated\Push". This folder is generated during Workshop installation. After having selected the desired image files, click on the Open button.

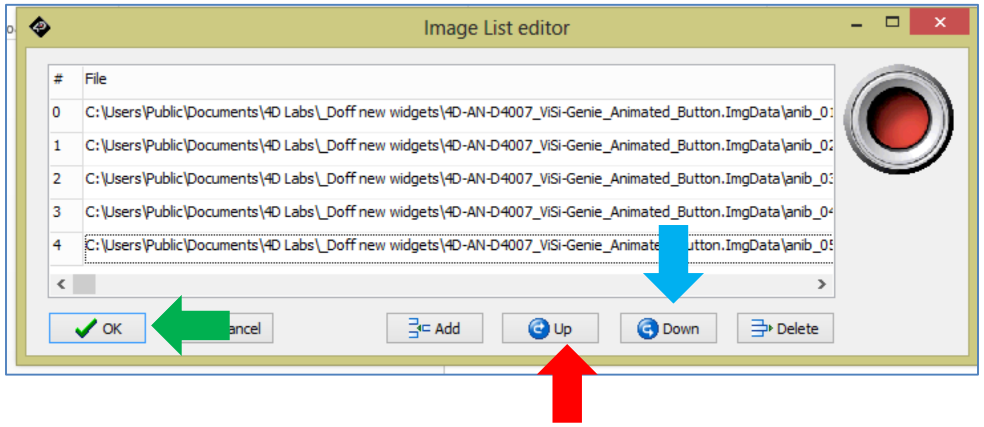

ated_Button.ImgData
Search 4D-AN-D4007_ViSi-Ge...
Name
anib_01.png
anib_02.png
anib_03.png
anib_04.png
anib_05.png
anib_03.png" "anib_04.png"
All (*.gif;*.jpg;*.jpeg;*.png;*.bm
Open
Cancel

The Image List editor window is again displayed. The left-most column lists the image numbers or the states.



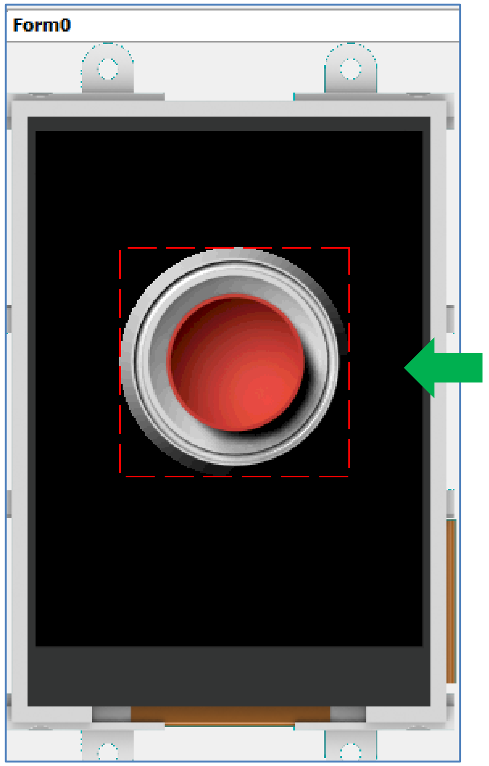
#	File
0	C:\Users\Public\Documents\4D Labs_Doff new widgets\4D-AN-D4007_ViSi-Genie_Animated_Button.ImgData\anib_0
1	C:\Users\Public\Documents\4D Labs_Doff new widgets\4D-AN-D4007_ViSi-Genie_Animated_Button.ImgData\anib_0
2	C:\Users\Public\Documents\4D Labs_Doff new widgets\4D-AN-D4007_ViSi-Genie_Animated_Button.ImgData\anib_0
3	C:\Users\Public\Documents\4D Labs_Doff new widgets\4D-AN-D4007_ViSi-Genie_Animated_Button.ImgData\anib_0
4	C:\Users\Public\Documents\4D Labs_Doff new widgets\4D-AN-D4007_ViSi-Genie_Animated_Button.ImgData\anib_0

Rearrange the order by selecting an image and pressing the up or down button. When done, click on the OK button.



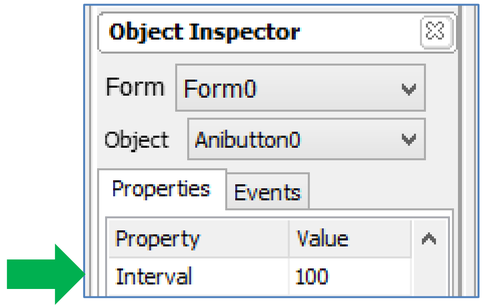
#	File
0	C:\Users\Public\Documents\4D Labs_Doff new widgets\4D-AN-D4007_ViSi-Genie_Animated_Button.ImgData\anib_0
1	C:\Users\Public\Documents\4D Labs_Doff new widgets\4D-AN-D4007_ViSi-Genie_Animated_Button.ImgData\anib_0
2	C:\Users\Public\Documents\4D Labs_Doff new widgets\4D-AN-D4007_ViSi-Genie_Animated_Button.ImgData\anib_0
3	C:\Users\Public\Documents\4D Labs_Doff new widgets\4D-AN-D4007_ViSi-Genie_Animated_Button.ImgData\anib_0
4	C:\Users\Public\Documents\4D Labs_Doff new widgets\4D-AN-D4007_ViSi-Genie_Animate utton.ImgData\anib_0

The WYSIWYG screen is updated with the initial state displayed. Take time to experiment with the different properties.



Defining the Frame Delay of an Animated Button

The property “Interval” defines the frame delay.

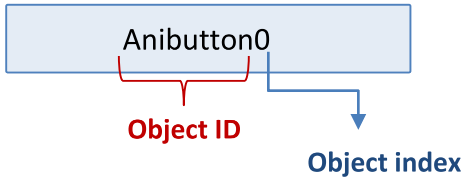


Property	Value
Interval	100

The default interval is 100 milliseconds. The reciprocal of the frame delay is the frame rate, which is 10 frames per second in this example.

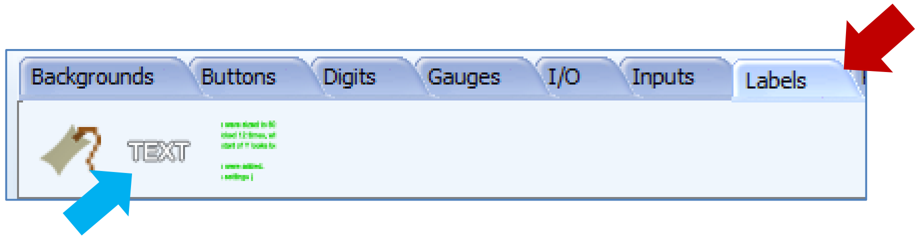
Naming of Objects

Naming is important to differentiate between objects of the same kind. For instance, suppose the user adds another animated button object to the WYSIWYG screen. This object will be given the name Anibutton1– it being the second animated button object in the program. The third animated button will be given the name Anibutton2, and so on. An object’s name therefore identifies its kind and its unique index number. It has an ID (or type) and an index.

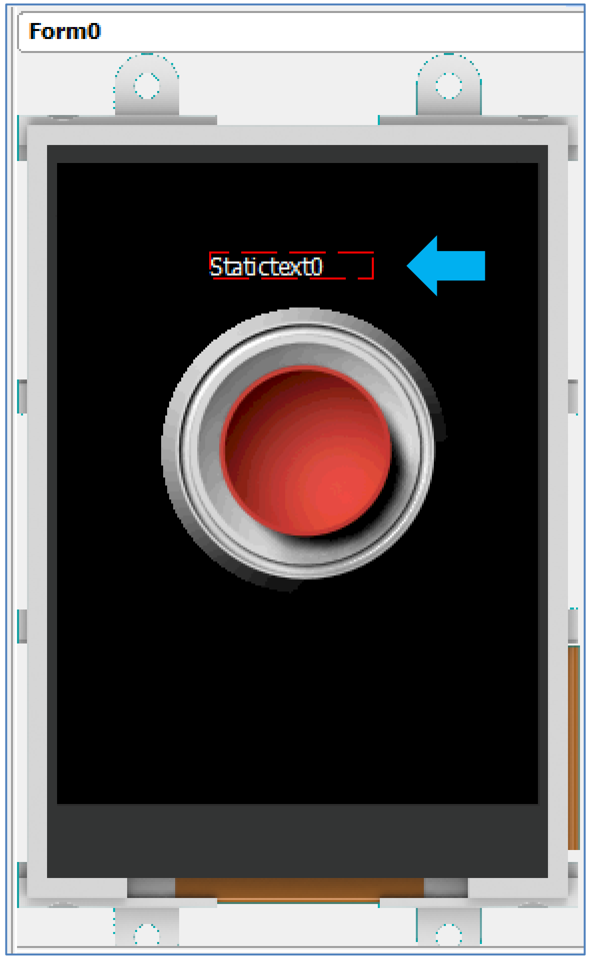


Adding a Static Text

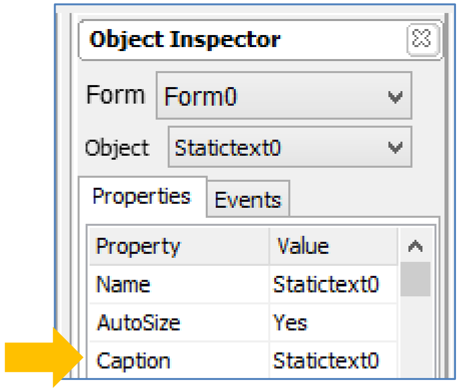
To add a static text, go to the **Labels** pane and click on the **static text** icon.



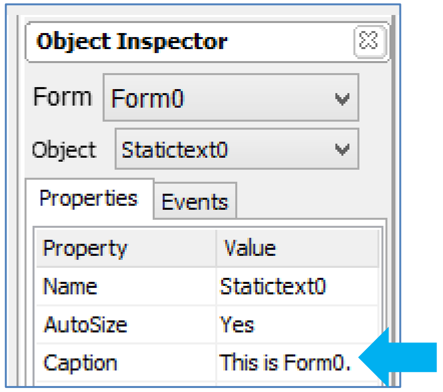
Click on the WYSIWYG screen to place the object. Drag it to any desired location.



Change the caption in the object inspector.

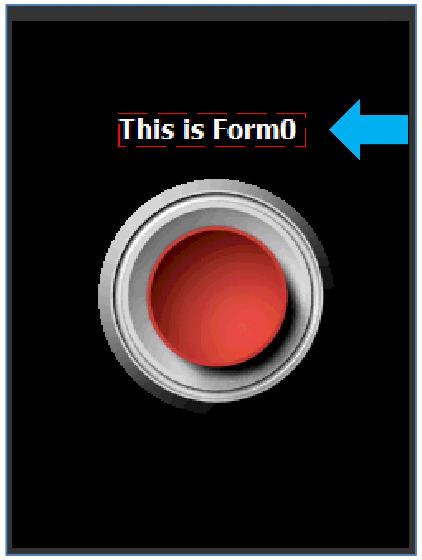


Property	Value
Form	Form0
Object	Statictext0
Name	Statictext0
AutoSize	Yes
Caption	Statictext0



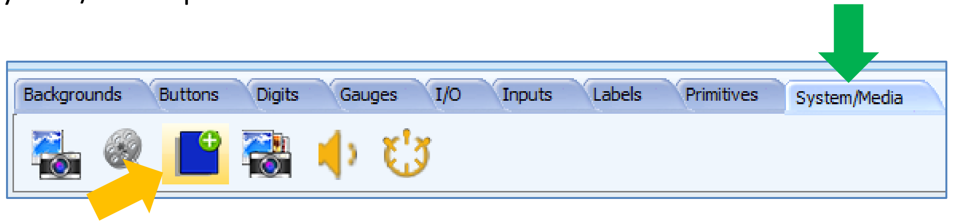
Property	Value
Form	Form0
Object	Statictext0
Name	Statictext0
AutoSize	Yes
Caption	This is Form0.

Experimentation with the other properties of the static text is left to the user as an exercise. The WYSIWYG screen is updated accordingly.

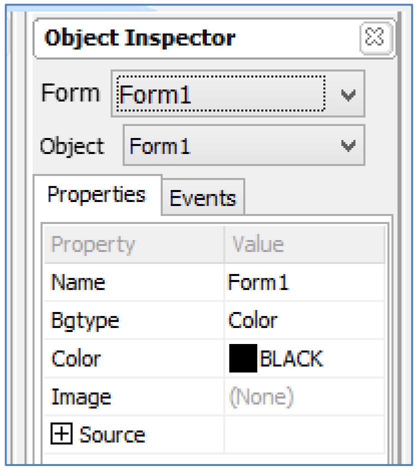


Adding a New Form

A new form is added the purpose of which is to demonstrate the use of an animated button configured for toggling. To add a new form, go to the System/Media pane and select the form icon.



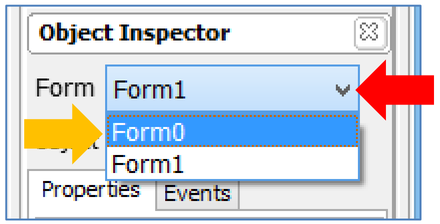
A new blank form, named Form1, is generated.



Form: Form1
Object: Form1

Property	Value
Name	Form1
Bgtype	Color
Color	■ BLACK
Image	(None)
⊞ Source	

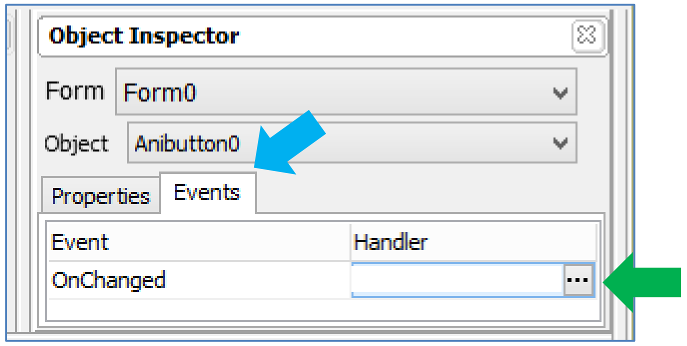
To navigate back to Form0 in Workshop, click on the drop-down menu arrow of the object inspector as shown below and select Form0.



Do this to navigate between forms.

Linking Forms

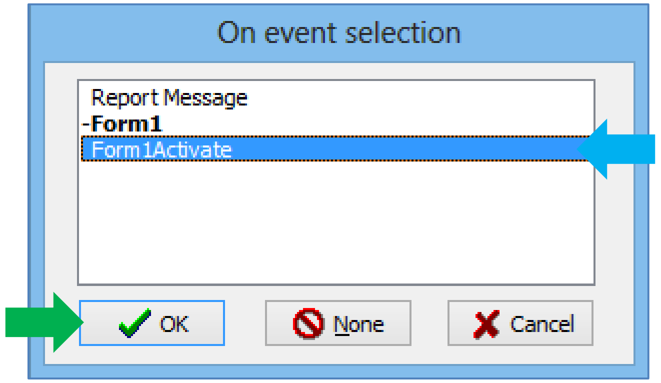
Now that there are two forms, a button in Form0 can be used to display Form1 (and vice-versa) when the program runs. Here Anibutton0 of Form0 is configured as a navigation button. In Form0, select Anibutton0, go to the object inspector, and select the Events tab. Click on the ellipsis dots of the OnChanged event.



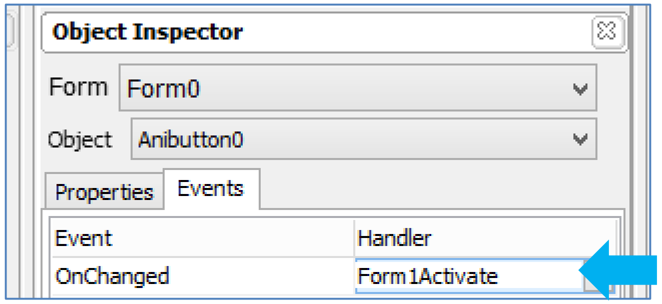
Form: Form0
Object: Anibutton0

Event	Handler
OnChanged	

The On event selection window appears. Choose "Form1Activate" and click OK.



The object inspector is updated accordingly.



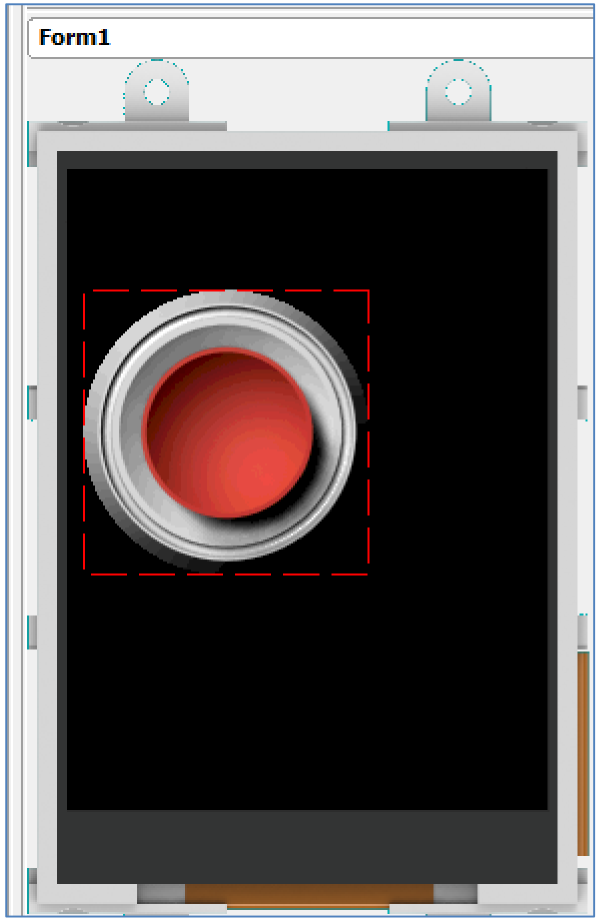
Event	Handler
OnChanged	Form1Activate

Form: Form0
Object: Anibutton0

Adding a Toggle Animated Button

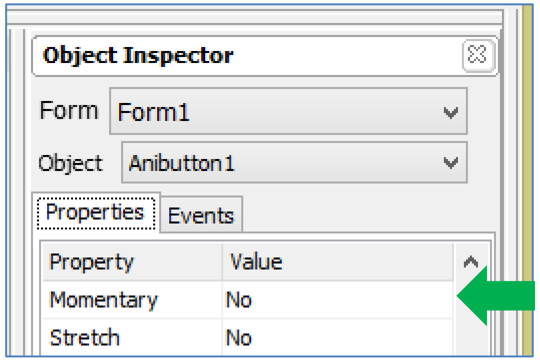
Defining the States of a Toggle Animated Button

Another animated button (Anibutton1) is created in Form1. This button will be configured as a toggle button. Anibutton1, when pressed and released, displays all the frames (from first to last). Another press-and-release is needed for it to play the frames backwards (from the last to first). Anibutton1 of Form1 is similar to Anibutton0 of Form0. The only difference is the configuration of the property "Momentary".



Configuring an Animated Button as a Toggle Button

To configure Anibutton1 to behave as a toggle button, go to the object inspector and set the momentary property to "No".

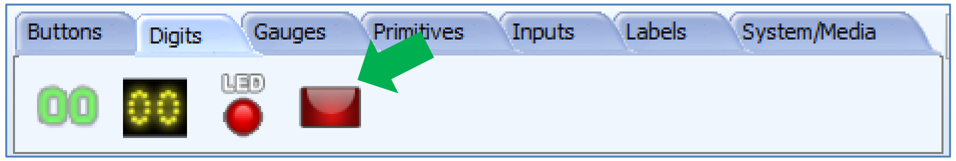


Form	Form1
Object	Anibutton1

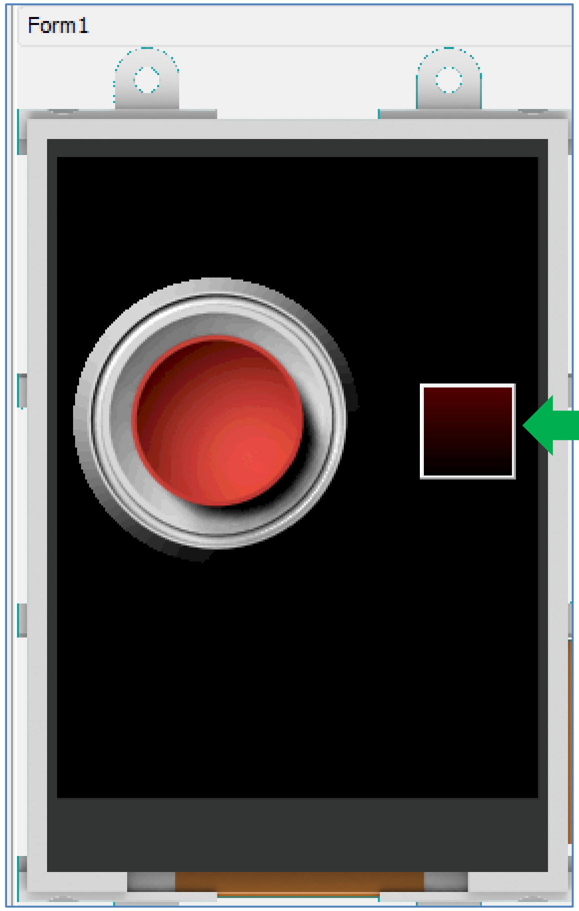
Property	Value
Momentary	No
Stretch	No

Adding a User LED

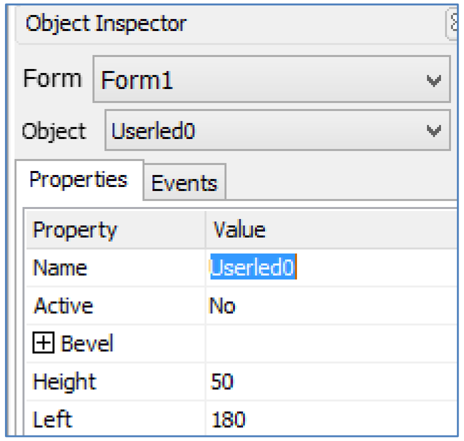
To add a user LED, go to the Digits pane and select the user LED icon.



Click on the WYSIWYG screen to place it.



The object can be dragged and resized. The properties can be edited in the Object Inspector. Userled0 of Form1 has the following properties:



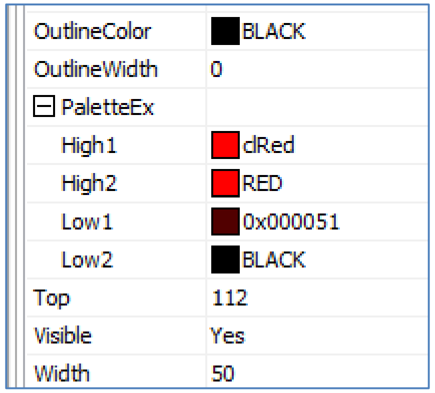
Object Inspector

Form: Form1

Object: Userled0

Properties | Events

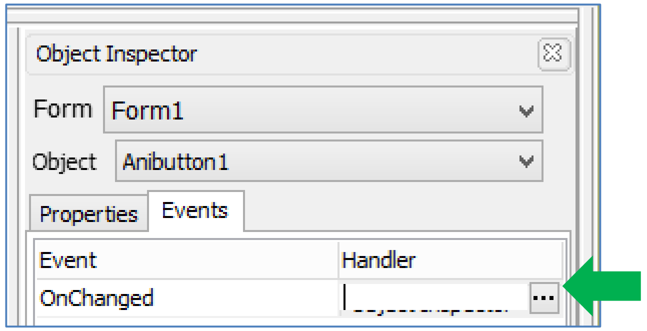
Property	Value
Name	Userled0
Active	No
⊞ Bevel	
Height	50
Left	180



OutlineColor	⬛ BLACK
OutlineWidth	0
⊟ PaletteEx	
High1	🟥 dRed
High2	🟥 RED
Low1	🟥 0x000051
Low2	⬛ BLACK
Top	112
Visible	Yes
Width	50

Linking Objects

Now that Anibutton1 is a toggle button, it can be used to control Userled0. Go to Anibutton1's object inspector and click on the ellipsis dots of the onChanged event.



Object Inspector

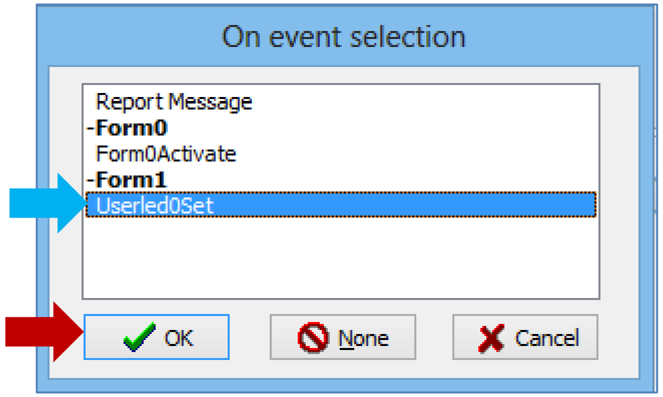
Form: Form1

Object: Anibutton1

Properties | Events

Event	Handler
OnChanged	...

The On event selection window appears. Select Userled0Set and click OK.



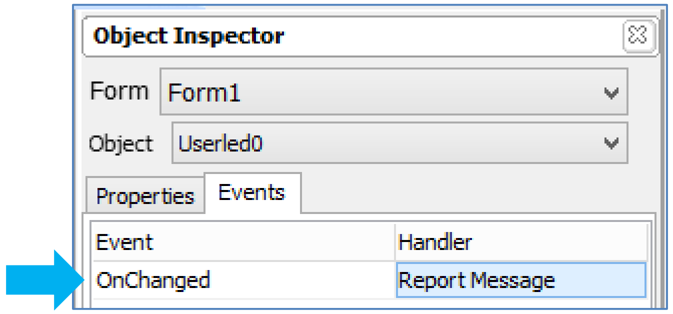
On event selection

Report Message
-**Form0**
Form0Activate
-**Form1**
Userled0Set

When the program runs, Anibutton1 will toggle Userled0.

Configuring the User LED to Report a Message

Userled0 can be configured to report a message to an external host controller when its status has changed. Go to its object inspector and configure the onChanged event as shown below.



Object Inspector

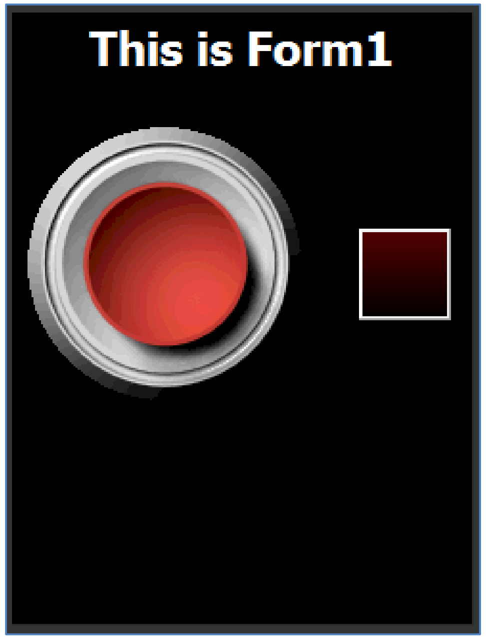
Form: Form1

Object: Userled0

Properties | Events

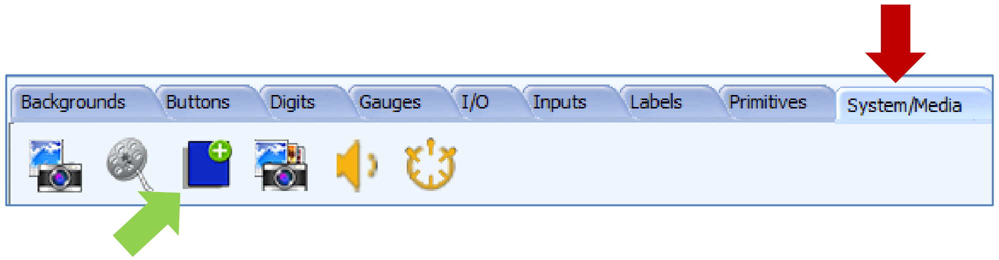
Event	Handler
OnChanged	Report Message

Add another static text object to the form. When done, the form should look like as shown below.



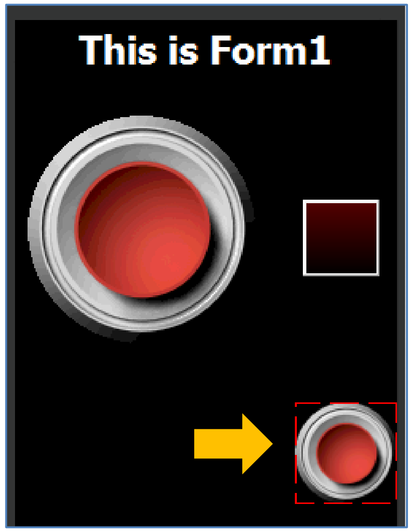
Create a Button Matrix

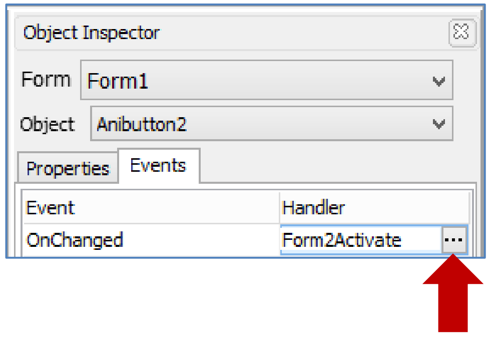
Add another form to the project – Form2.



This form will contain three animated buttons which form a matrix. Buttons in a matrix behave such that only one button can be enabled at a time. Enabling another button disables all the other buttons. A matrix of buttons can be conveniently used as a GUI menu.

First, create a navigation button to link Form1 to Form2. In Form1, a momentary animated button similar to Anibutton0 in Form0 is added.




Object Inspector

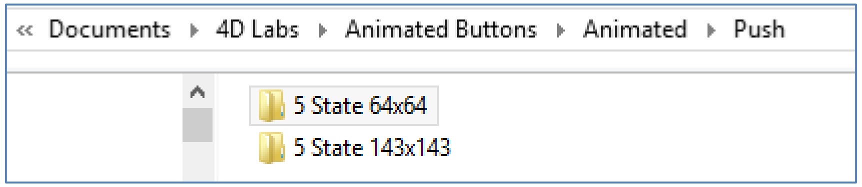
Form: Form1

Object: Anibutton2

Properties | Events

Event	Handler
OnChanged	Form2Activate

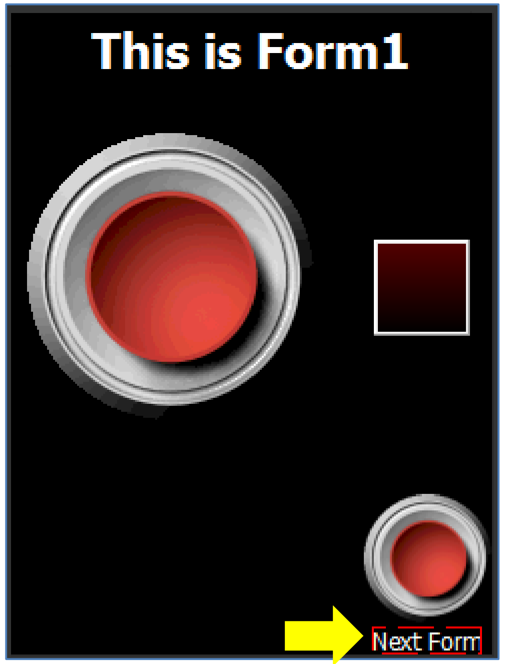
Note that for Anibutton0 and Anibutton1, the image size is 143 pixels by 143 pixels. For Anibutton2, the image size is 64 pixels by 64 pixels. These files are in the Workshop folder shown below.


« Documents ▸ 4D Labs ▸ Animated Buttons ▸ Animated ▸ Push

5 State 64x64

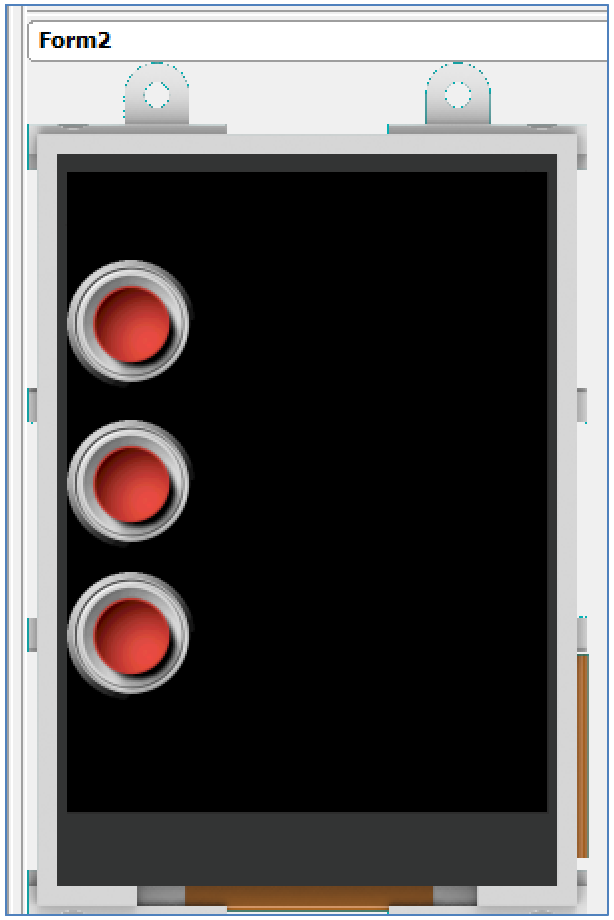
5 State 143x143

Add another static text object to label Anibutton2.



Adding Three Animated Buttons

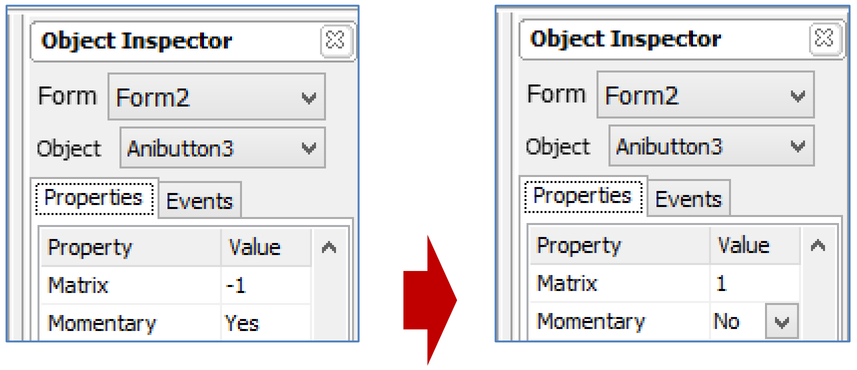
Add three animated buttons to Form2. These are Anibutton3, Anibutton4, and Anibutton5.



These buttons are similar to Anibutton2 in Form1 except that these are configured as toggle buttons.

Configuring the Three Animated Buttons as a Matrix

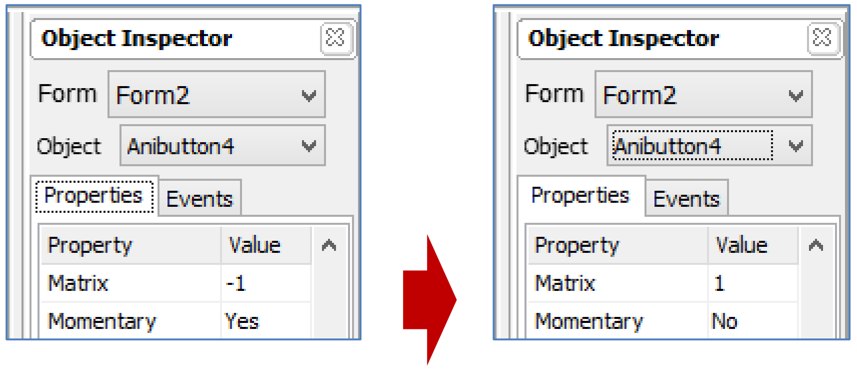
To group these buttons into a matrix, edit the Matrix property in the object inspector.


Object Inspector
Form Form2
Object Anibutton3
Properties Events

Property	Value
Matrix	-1
Momentary	Yes

Object Inspector
Form Form2
Object Anibutton3
Properties Events

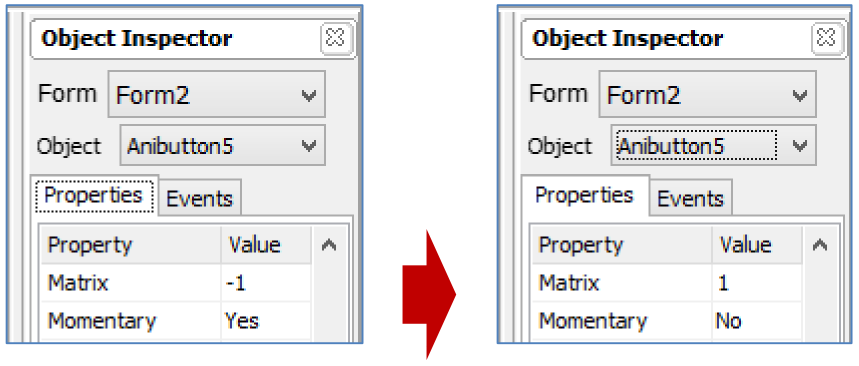
Property	Value
Matrix	1
Momentary	No


Object Inspector
Form Form2
Object Anibutton4
Properties Events

Property	Value
Matrix	-1
Momentary	Yes

Object Inspector
Form Form2
Object Anibutton4
Properties Events

Property	Value
Matrix	1
Momentary	No


Object Inspector
Form Form2
Object Anibutton5
Properties Events

Property	Value
Matrix	-1
Momentary	Yes

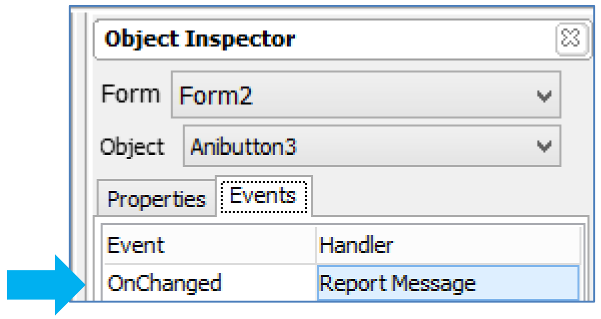
Object Inspector
Form Form2
Object Anibutton5
Properties Events

Property	Value
Matrix	1
Momentary	No

It is important that all buttons grouped share the same matrix number, otherwise pressing on one button will not release the other buttons of the group.

Configuring the Three Animated Buttons to Report a Message

An animated button can be configured to report a message to an external host controller when its status has changed. Go to the object inspector of Anibutton3 and configure the onChanged event as shown below.

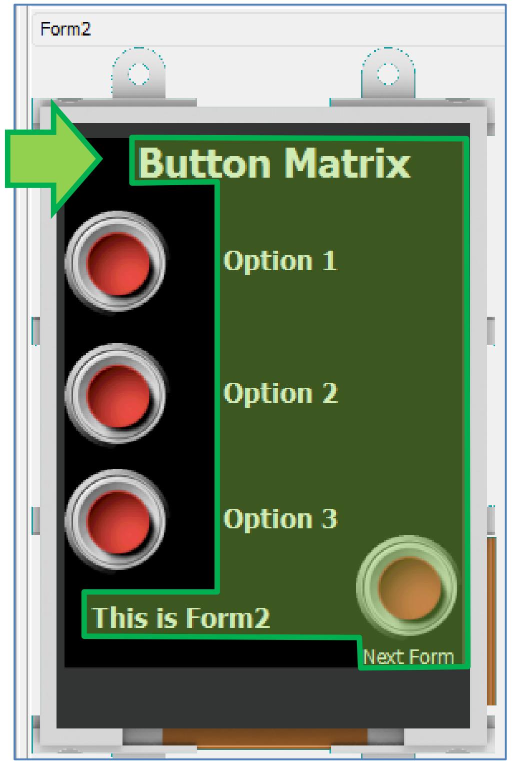

Object Inspector
Form Form2
Object Anibutton3
Properties Events

Event	Handler
OnChanged	Report Message

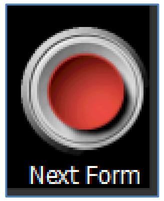
Do the same for Anibutton4 and Anibutton5.

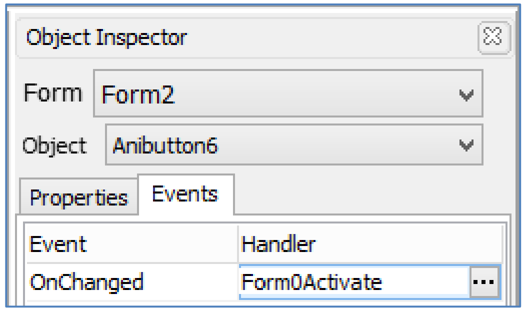
Complete the Form

Form2 of this project has six static text objects and a navigation animated button. Follow the instructions described earlier to add these objects.



Note that Anibutton6 is linked to the first form – Form0.





Object Inspector

Form: Form2

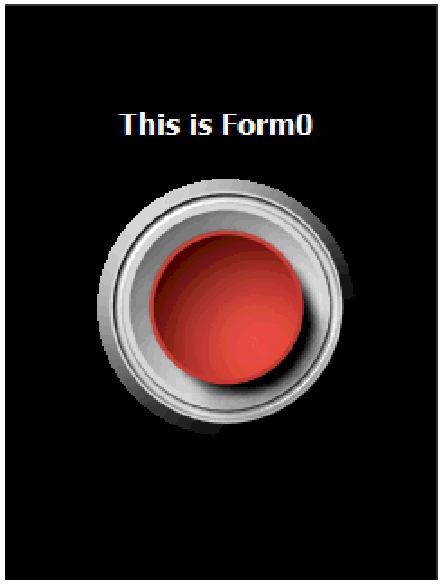
Object: Anibutton6

Properties | Events

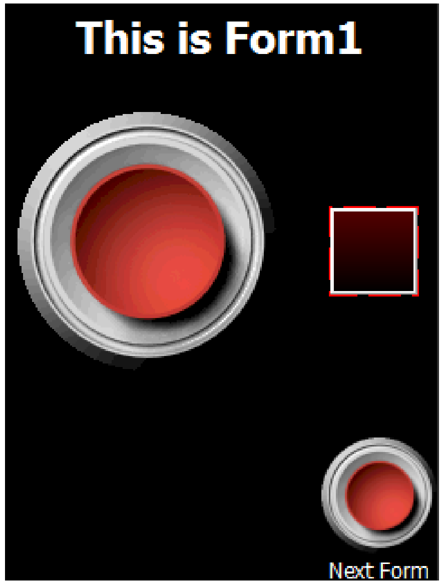
Event	Handler
OnChanged	Form0Activate

The Complete Project

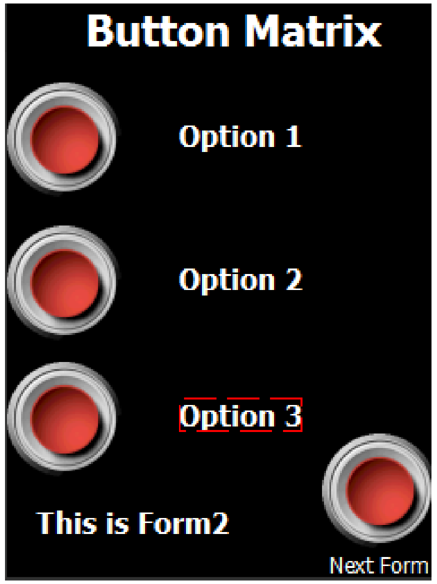
The project now has three forms. In Form0 is Anibutton0 which is a momentary button used for displaying the next form – Form1.



In Form1 are Anibutton1 and Anibutton2. Anibutton1 is a toggle button used to control a user LED. Anibutton2 is a momentary navigation button used to display the next form – Form2.



In Form2 are Anibutton3, Anibutton4, Anibutton 5, and Anibutton 6. Anibuttons 3 to 5 are toggle buttons that form a matrix while Anibutton6 is a momentary button for navigating back to the first form.



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.

Identify the Messages

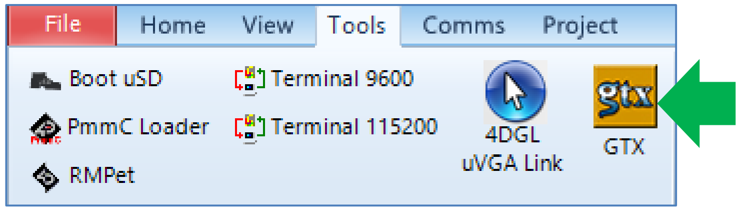
The display module is going to receive and send messages from and to an external host. This section explains to the user how to interpret these messages. An understanding of this section is necessary for users who intend to interface the display to a host. The ViSi Genie Reference Manual is recommended for advanced users.

Use the GTX Tool to Analyse the Messages

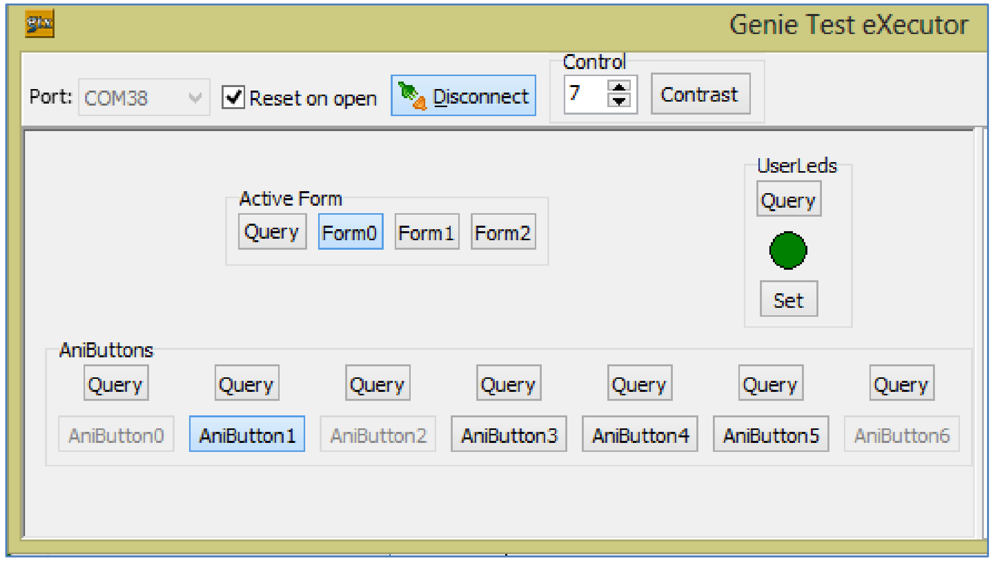
Using the GTX or **Genie Test eXecutor** tool is the first option to get the messages sent by the screen to the host. Here the PC will be the host. The GTX tool is a part of the Workshop 4 IDE. It allows the user to receive, observe, and send messages from and to the display module. It is an essential debugging tool.

Launch the GTX Tool

Under Tools menu click on the GTX tool button.



A new window appears, with the forms, animated buttons, and the user LED created previously.



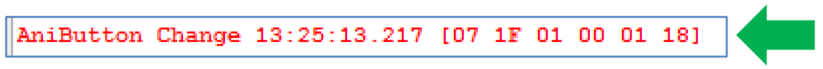
The Animated Button

Report Event

When the program starts, navigate to Form1 (the second form) by pressing the button on Form0 (the first form) of the display module screen.

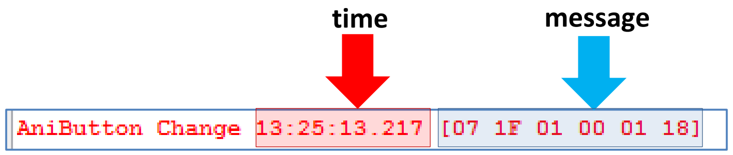
When Form1 is displayed, toggle the user LED by playing with Anibutton1.

Observe the white area on the right part of the GTX tool window. It displays the message received from the display module.



```
AniButton Change 13:25:13.217 [07 1F 01 00 01 18]
```

The actual message bytes are those inside the brackets. These values are in hexadecimal. The figure preceding the actual message is the computer time at which the message is sent. A label is also included to tell the observer what the message represents.



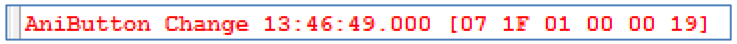
```
AniButton Change 13:25:13.217 [07 1F 01 00 01 18]
```

The message received is formatted according to the following pattern:

Command	Object Type	Object Index	Value MSB	Value LSB	Checksum
07	1F	01	00	01	18
REPORT_EVENT	Animated button	Second	0x0001		

The message is from Anibutton0, and it contains the hexadecimal value “0x0001”. When a toggle button is enabled, the value is **0x0001** or simply **1** in decimal. When a toggle button is disabled, the value is **0x0000** or simply **0** in decimal. Verify this by toggling the button back to its very first state.

The message sent when the animated button has just been disabled is shown below.



```
AniButton Change 13:46:49.000 [07 1F 01 00 00 19]
```

The checksum is a means for the host to verify if the message received is correct. This byte is calculated by XOR’ing all bytes in the message from (and including) the CMD or command byte to the last parameter byte. Then, the result is appended to the end to yield the checksum byte. If the message is correct, XOR’ing all the bytes (including the checksum byte) will give a result of zero. Checking the integrity of a message using the checksum byte shall be handled by the host.

Input and Output Objects

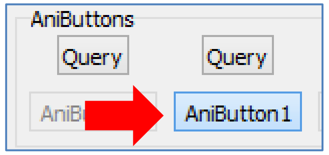
Remember that when designing Form1 earlier, Userled0 was configured to report a message when its status has changed. Anibutton1 on the other hand was only configured to toggle Userled0. The user therefore might have expected the message to be similar to:

UserLED Change 17:23:26.552 [07 13 00 00 01 32]

The object ID of a user LED is 0x13. See section 3.3 of the ViSi Genie Reference Manual for a full list of object IDs. The above is not the case however since a user LED is classified as an output object. Only input objects such as an animated button can initiate an event. The message therefore is actually from Anibutton1, although it is Userled0 that was configured to report a message. To learn more about the input-output classification of Genie objects, refer to section 7 of the ViSi-Genie User Guide.

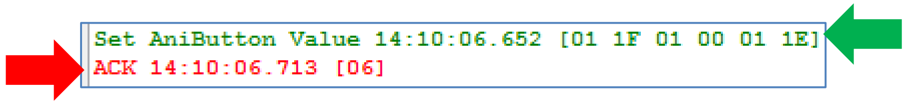
Toggle an Animated Button using the GTX Tool

In the GTX tool window, click on the button indicated below.



Note that Anibutton1 in Form1 of the display module screen has changed its state. Also, the white area on the right displays

- in **green** the messages sent to the display module
- and in **red** the messages received from the display module



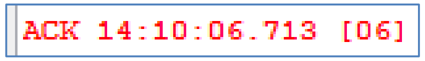
```
Set AniButton Value 14:10:06.652 [01 1F 01 00 01 1E]
ACK 14:10:06.713 [06]
```

The message sent is formatted according to the following pattern:

Command	Object Type	Object Index	Value MSB	Value LSB	Checksum
01	1F	01	00	01	1E
WRITE_OBJ	Animated button	Second	0x0001		

The message stands for "Write to the second animated button object of the display module the value **0x0001**".

ACK = 0x06 as shown below



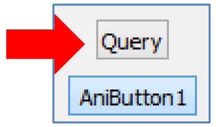
```
ACK 14:10:06.713 [06]
```

is an acknowledgment from the display module which means that it has understood the message.

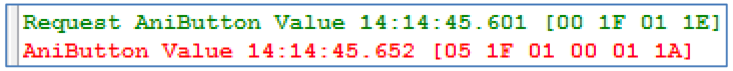
Poll the Animated Button

If the animated button (or any other output object linked to it) was not configured to report a message when its status has changed, it is still possible (although sometimes not advisable in practical applications) to know its current status (enabled or disabled) by polling.

To do be able to do this, click on the Query button.



Messages are sent to and received from the display module.



```
Request AniButton Value 14:14:45.601 [00 1F 01 1E]
AniButton Value 14:14:45.652 [05 1F 01 00 01 1A]
```

The messages are formatted according to the following pattern:

Command	Object Type	Object Index	Value MSB	Value LSB	Checksum
00	1F	01	-	-	1E
READ_OBJ	Animated button	Second	N/A		
05	1F	01	00	01	1A
REPORT_OBJ	Animated button	Second	0x0001		

The host sends a READ_OBJ command specifically asking for the value (or status) of the second animated button object. The display module then responds with the current value of that animated button. Communication between a 4D display module programmed with a ViSi-Genie application and an external host controller must follow the ViSi-Genie Communications Protocol, which is defined in the ViSi Genie Reference Manual

Proprietary Information

Disclaimer of Warranties & Limitation of Liability