APPLICATION NOTES

4D SYSTEMS
TURNING TECHNOLOGY INTO ART

ViSi Genie Getting Started – First Project for Picaso Displays

DOCUMENT DATE: **25th APRIL 2020**
DOCUMENT REVISION: **1.3**

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:

- Any of the following 4D Picaso touch display modules:

gen4-uLCD-24PT	**gen4-uLCD-28PT**	**gen4-uLCD-32PT**
uLCD-24PTU	**uLCD-28PTU**	

other superseded modules which support the ViSi-Genie environment

- 4D Programming Cable / uUSBPA5-II
  for non-gen4 displays (uLCD-xxx)
- 4D Programming Cable & gen4-IB / 4D-UPA / gen4-PA
  for gen4 displays (gen4-uLCD-xxx)
- micro-SD (uSD) memory card
- Workshop4 IDE (installed according to the installation document)

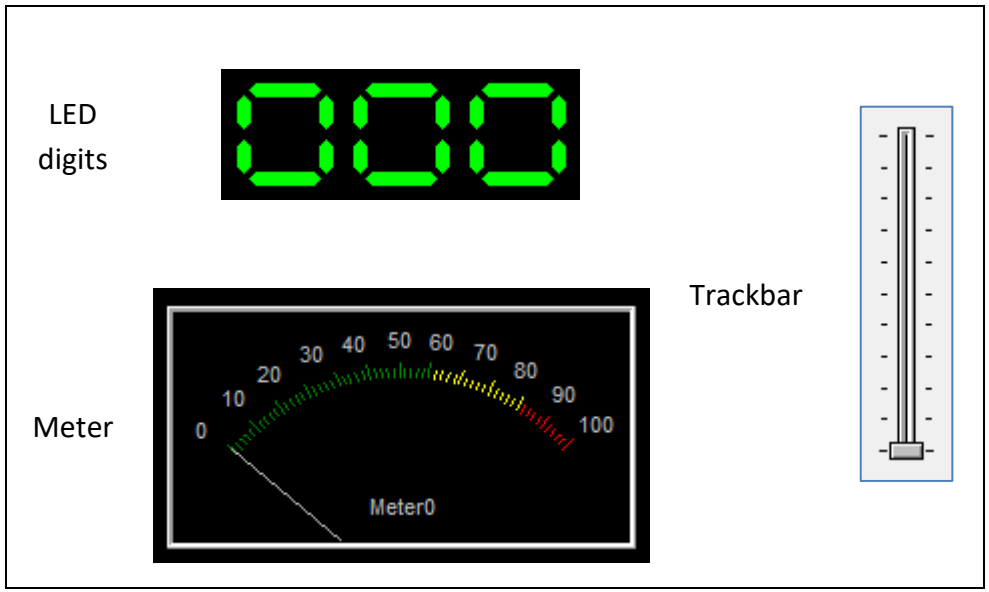
Visit www.4dsystems.com.au/products to see the latest Picaso displays and related products.

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. The following are examples of widgets used in this application note.



The simple project developed in this application note demonstrates basic touch functionality and object interaction. The user moves or touches the trackbar, and the meter and LED digits change their values to correspond with the trackbar's change in status. By default, input objects such as the

trackbar respond to touch. The user can configure an input object to drive an output object such as the meter or the LED digits. The project also illustrates how input objects are configured to send messages to an external host controller and how these messages are interpreted.

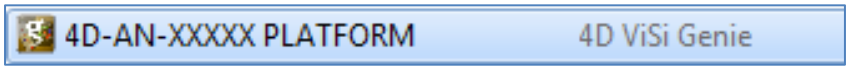
Launch Workshop4

There is a shortcut for Workshop 4 on the desktop.

Launch Workshop 4 by double-clicking on the icon.

Setup Procedure

This document comes with a ViSi-Genie program, with the format:



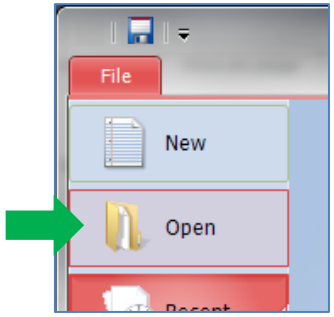
XXXXX refers to the application note number and “PLATFORM” indicates if the Visi-Genie program is setup for PICASO, DIABLO16 or GOLDELOX.

In this application note, the Visi-Genie program is:

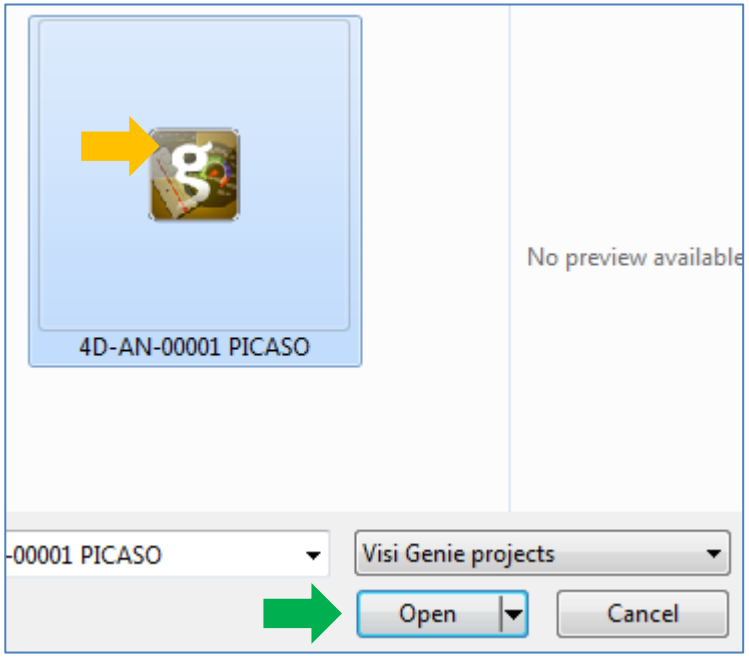
For users who want to learn how to create a ViSi Genie application, proceed to the next section.

Workshop 4 opens and displays the **Recent** page.

To load the existing project, click on Open.

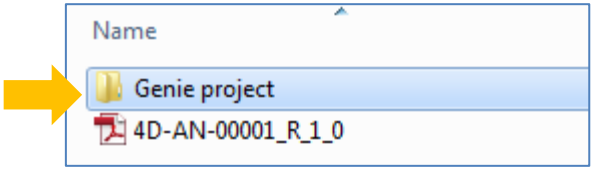


A standard open window asks for a ViSi-Genie project.

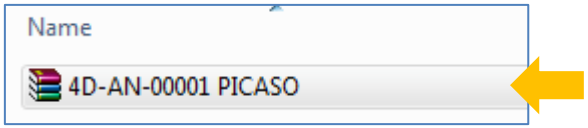


Open a Project from Explorer

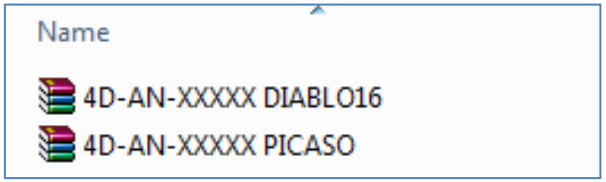
To open a project from explorer, extract the zip file downloaded from the application note site to any folder in the computer. Then open the “Genie project” folder.



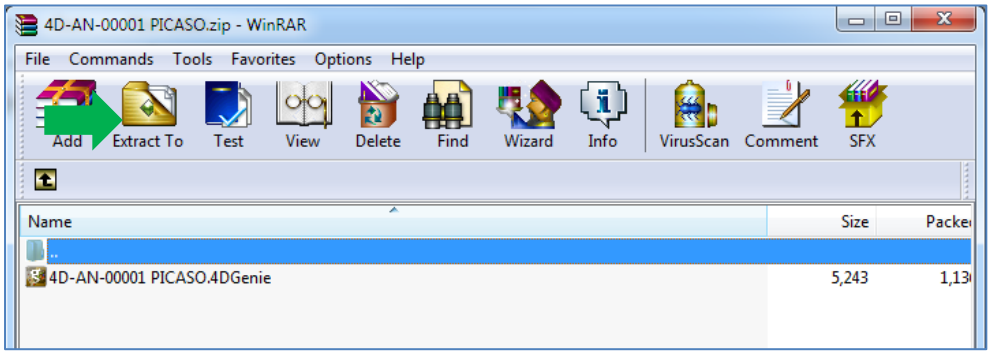
Inside the folder is a zip file of the Genie program.



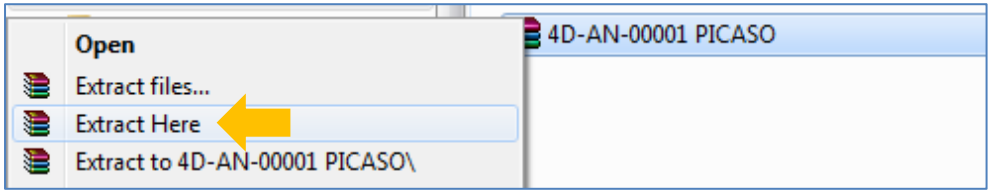
If the application note is multi-platform, it also comes with multiple zip files.



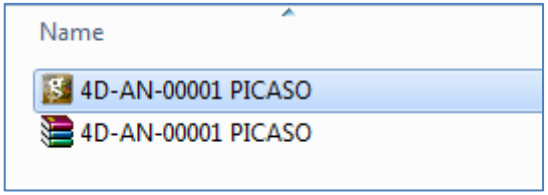
Now, extract the zip file to any folder in the PC. Click on “Extract to”.



Or extract it to the same folder by right-clicking on the zip file then clicking on “Extract Here”.

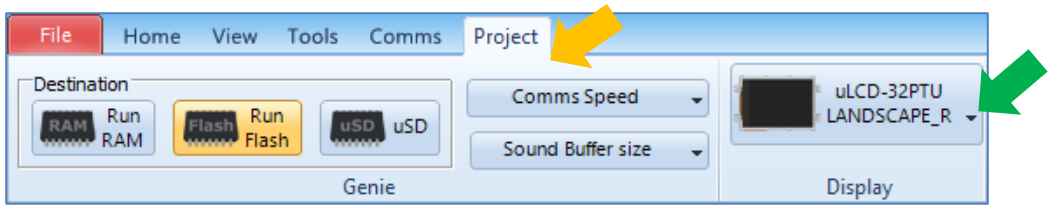


After extracting the contents of the zip file, double-click on the Visi – Genie program.

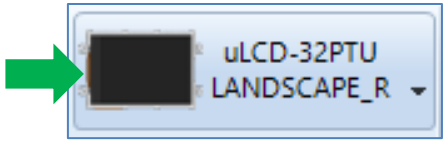


How to change the Target Display

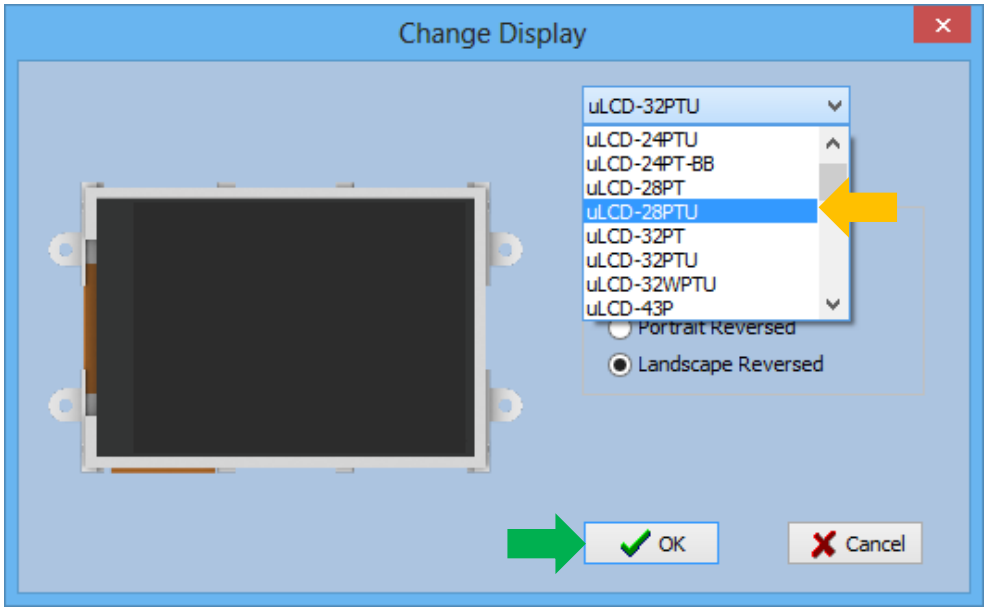
To change the target display, check the type of the screen module by selecting the **Project** menu.



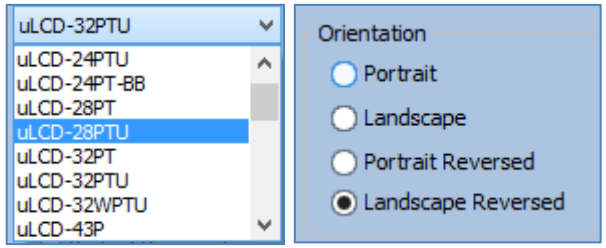
If using a different display module, change the target display module by clicking on the display button.

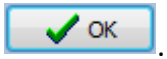


The Change Display window appears.



Select the appropriate screen on the drop-down list and define the orientation.



…and confirm by clicking on 

.

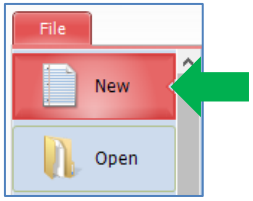
Create a New Project

Create a New Project

Workshop 4 opens and displays the **Recent** page.

To create a new project, there are two options.

Click on the top left-most icon, New.



Or Click on the icon beside Create a new Project.

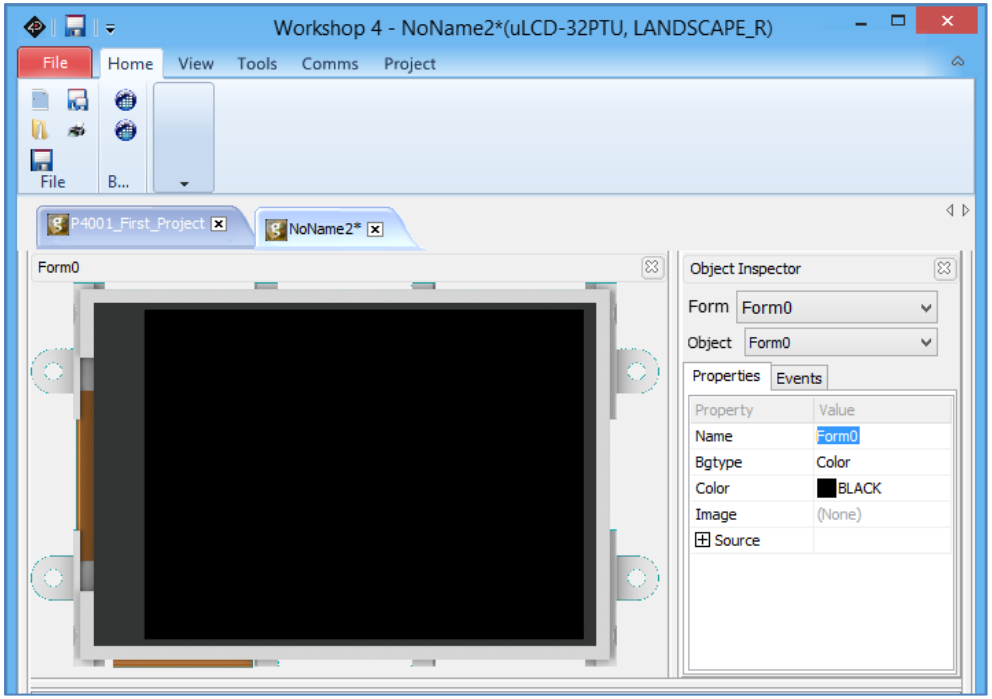
These options update the main window with the selection of the screen.

Select the appropriate screen and preferred orientation. The screen used in this example is a **uLCD-32PTU (landscape reversed orientation)**.

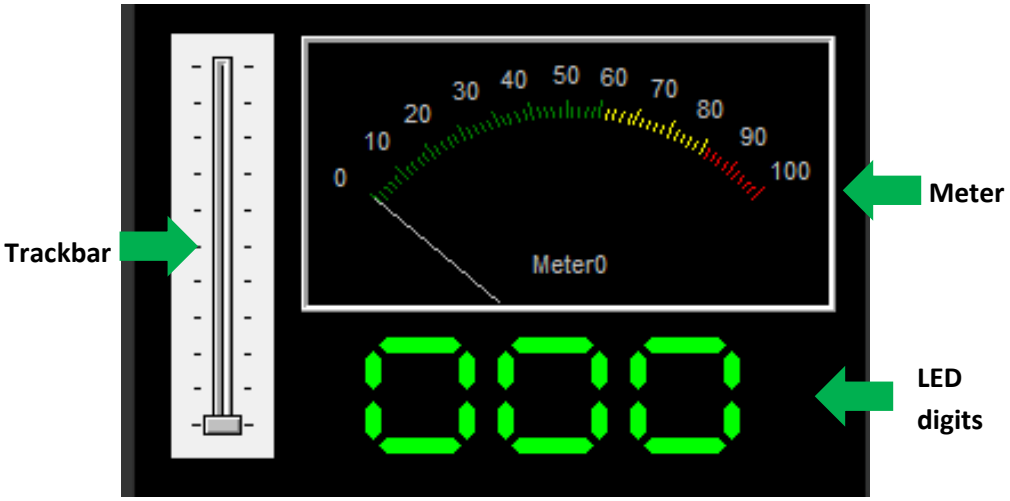
Select ViSi Genie

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.



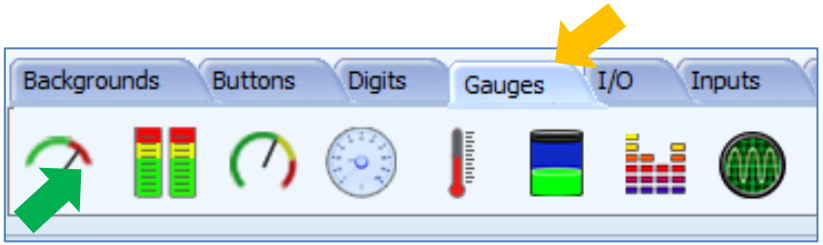
At the end of this section, the user will able to create a form with three objects. The final form will look like as shown below, with the labels excluded.



The procedure for adding each of these objects will now be discussed.

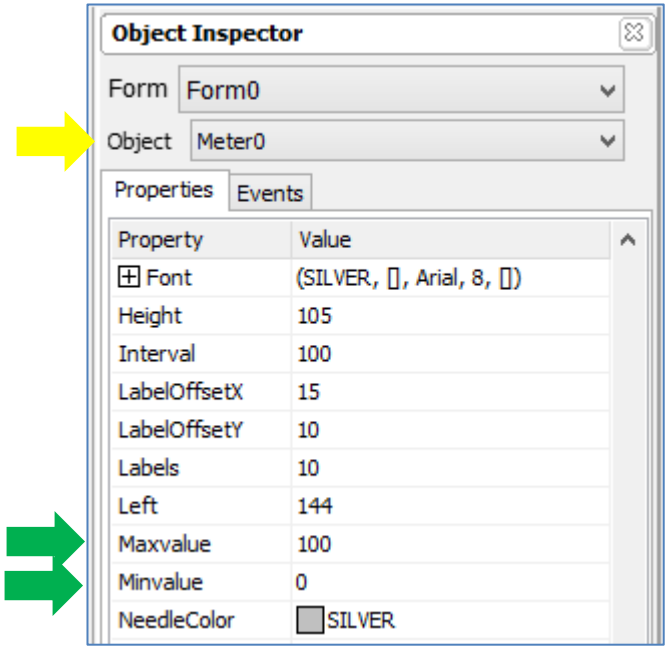
Adding a Meter

The meter changes its value while the trackbar is being moved. To add a meter, go to the **Gauges** pane then click on the **meter** icon.

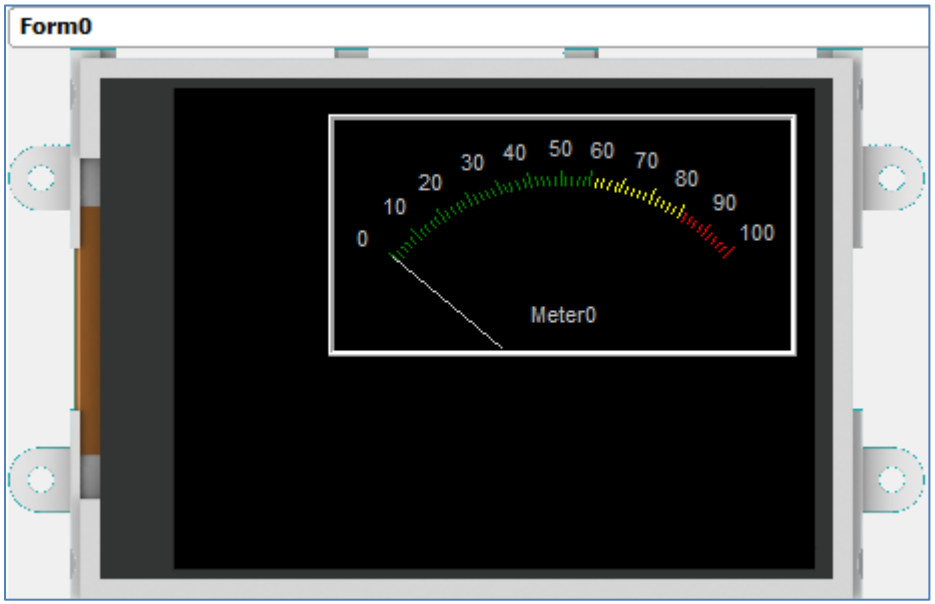


Click on the **WYSIWYG** (What-You-See-Is-What-You-Get) screen to put the meter in place. The WYSIWYG screen simulates the actual appearance of the display module screen.

The object can be dragged to any desired location and resized to the desired dimensions. The **Object Inspector** on the right part of the screen displays all the properties of the newly created meter object named **Meter0**.

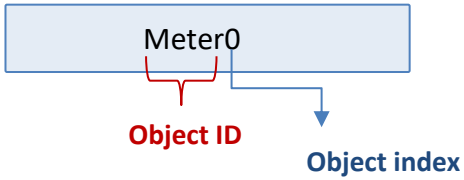


Feel free to experiment with the different properties. Take note of the maximum and minimum values. These correspond to the maximum and minimum values of the trackbar. When finished, the WYSIWYG screen will look similar to that shown below.



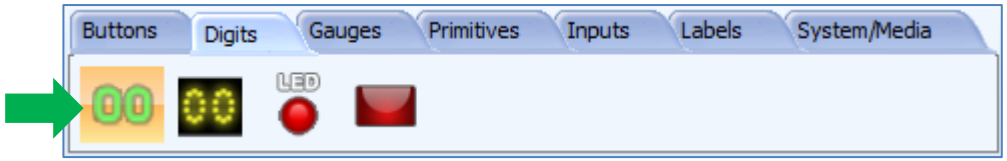
Naming of Objects

Naming is important to differentiate between objects of the same kind. For instance, suppose the user adds another meter to the WYSIWYG screen. This object will be given the name Meter1 – it being the second meter in the program. The third meter will be given the name Meter2, 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.

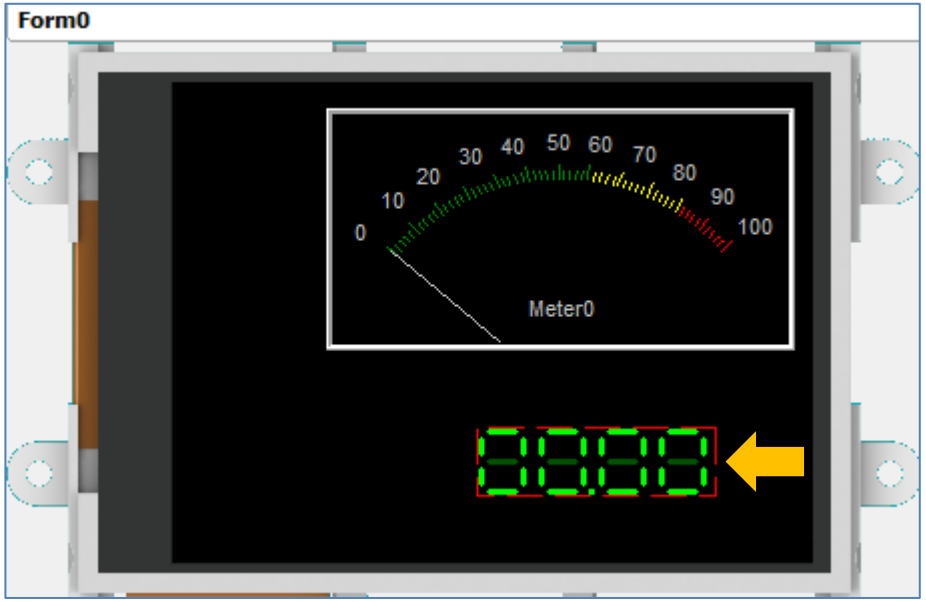


Adding LED Digits

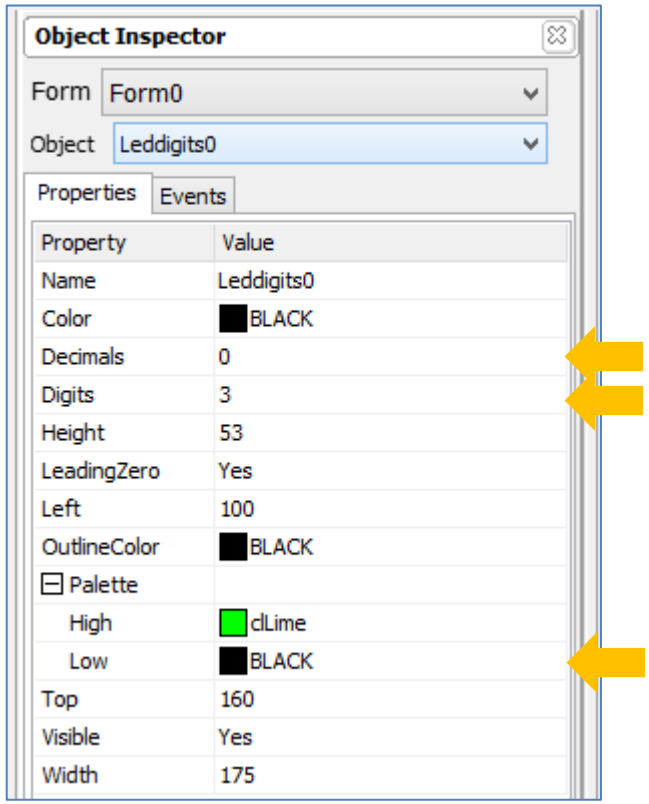
The **LED digits** object updates its value when the value of Meter0 has changed. To add a LED digits object, go to the **Digits** pane and select the first icon.



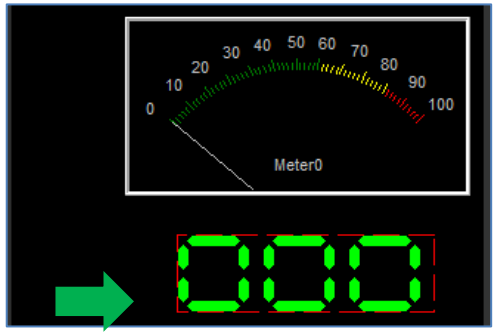
Click on the WYSIWYG screen to place the object.



Go to the Object inspector and set the following property values.

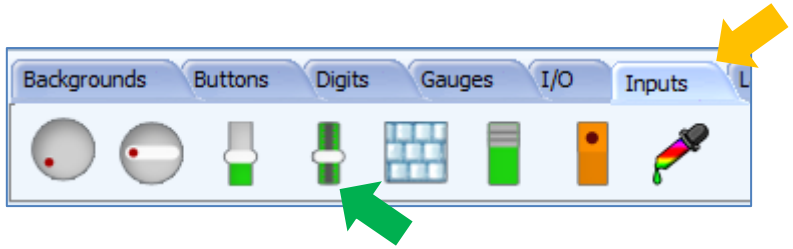


The updated appearance of the LED digits object is shown below.

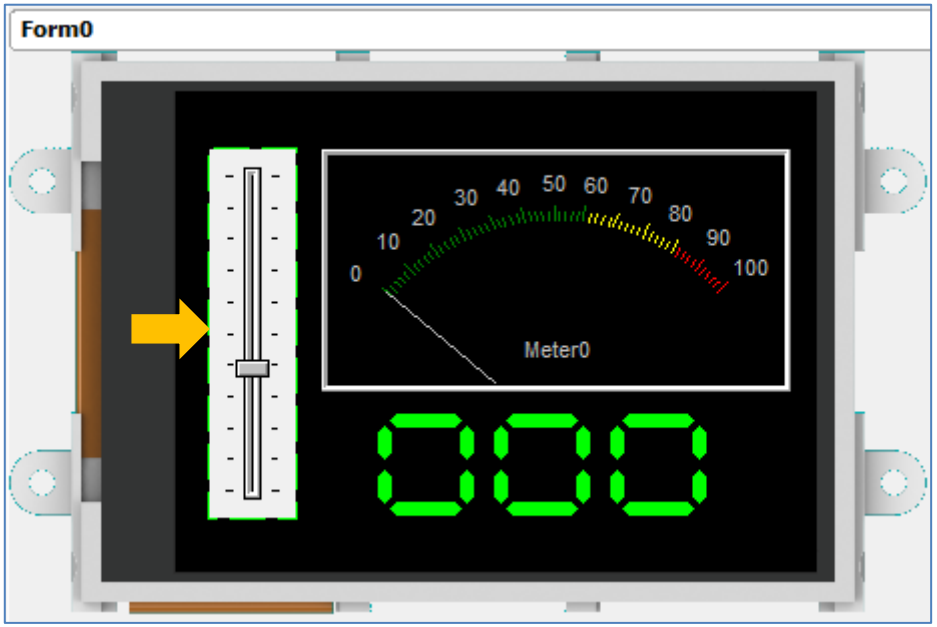


Adding a Trackbar

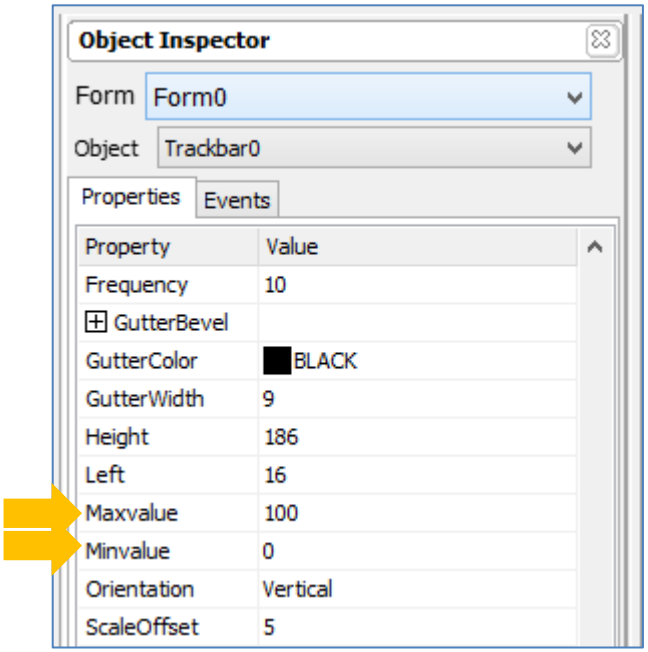
The trackbar responds to the user's touch and drives the meter and LED digits. To add a trackbar, go to the **Inputs** pane and click on the **trackbar** icon.



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



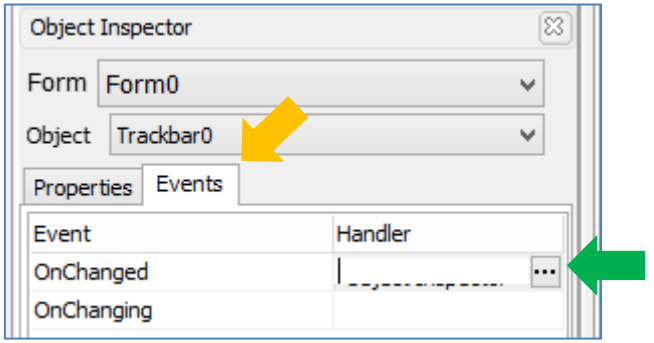
In the **Object Inspector**, the minimum value is 0 and maximum value is 100 by default.



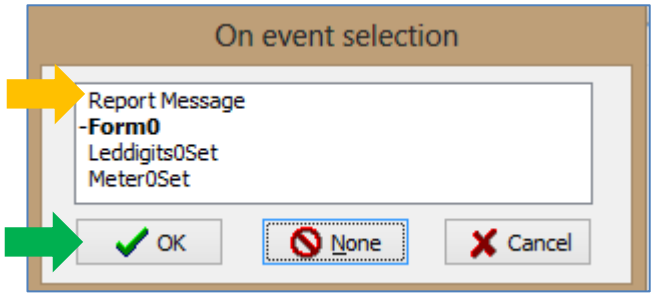
Configuring the Trackbar

The OnChanged Event

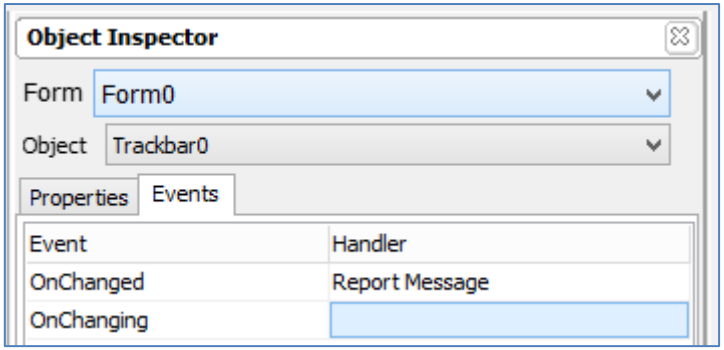
An input object such as the trackbar can be configured to report a message to an external host every time its (trackbar's) status has changed. To do this, click on the Events tab in the object inspector and click on the ... symbol in the **OnChanged** line.



The **On event selection** window appears. Select **Report Message** and click **OK**.



The Events pane is now updated.



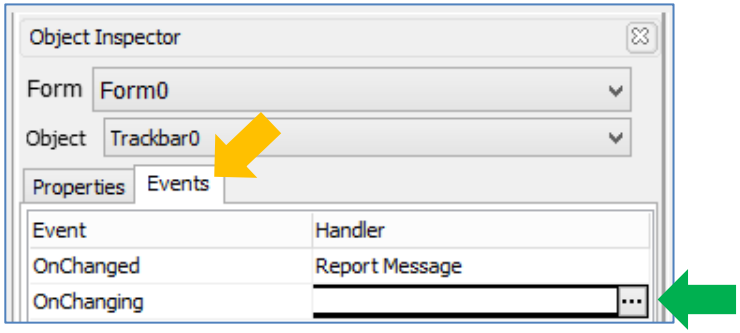
Now every time the trackbar has moved or its status has changed, it sends a message to the external host.

Report Message

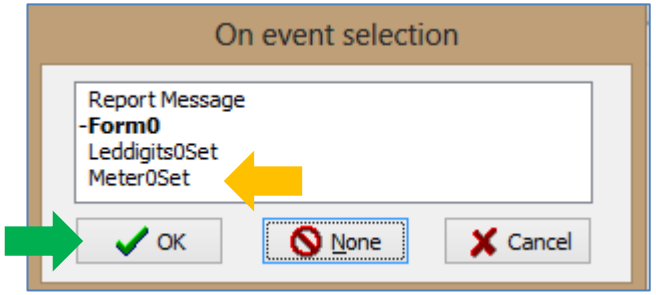
The use of the **Report Message** option in the **On event selection** window is one way by which the display module can communicate with an external host controller. For users who intend to interface the display to an external controller (such as the Arduino), this option allows an input object of the display module to update the host of its status. The message or data sent has a format which the host must understand. A section of this application note is dedicated to explaining this format (called the ViSi-Genie Communication Protocol) used by the display module. Advanced users may refer to the **ViSi Genie Reference Manual**.

The OnChanging Event

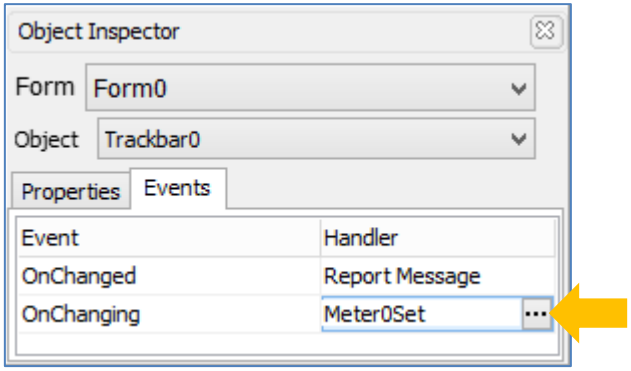
An input object such as the trackbar can also be configured to change the status of another object while its status is changing. To do this, click on the Events tab in the object inspector and click on the ... symbol in the **OnChanging** line.



The **On event selection** window appears. Select **Meter0Set** and click **OK**.



The Events pane is now updated.



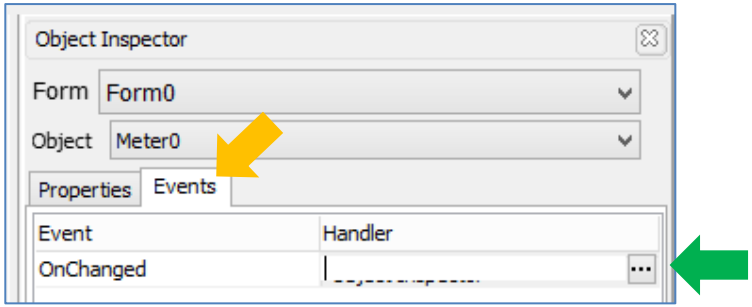
Now while the trackbar is being moved, it constantly sends its values to Meter0. Meter0 receives and displays these values on the fly.

OnChanged vs. OnChanging

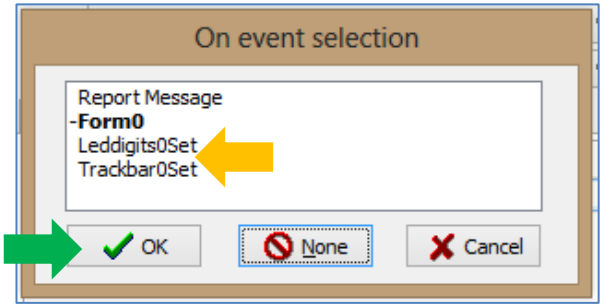
For the OnChanged event, the trackbar will send a value when the stylus or finger moving it is lifted off the screen. Selecting the **OnChanging** event, on the other hand, causes the trackbar to send values while it is being moved (the moving finger or stylus is not lifted off yet). To learn more about OnChanged and OnChanging events, read the application note **ViSi-Genie onChanging and onChanged Events.**

Linking Objects

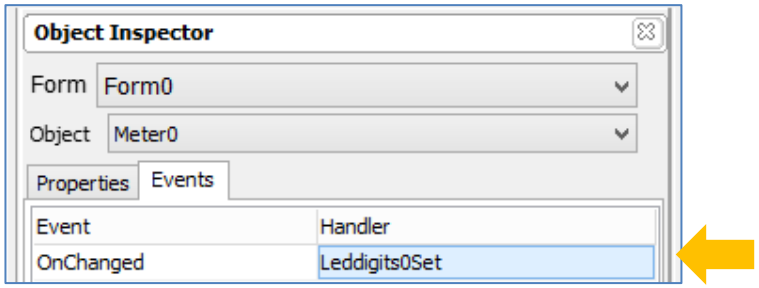
After having linked Trackbar0 to Meter0, it is also possible to further extend the link by configuring Meter0 to send its value to Leddigits0. To do this, click on Meter0 to select it. Select the Events tab in the object inspector and click on the ... symbol in the **OnChanged** line.



The **On event selection** window appears. Select **Leddigit0Set** and click **OK**.



The Events pane is now updated.



Now while Trackbar0 is being moved, it constantly sends its values to Meter0. Meter0 receives and displays these values while simultaneously updating Leddigits0. To learn more about how objects are linked and classified (input/output/combined), refer to the **ViSi Genie User Guide** section **Combining Objects**.

Build and Upload the Project

How to Save the Program

Save the program with the desired file name first.

Connecting gen4 Display Modules

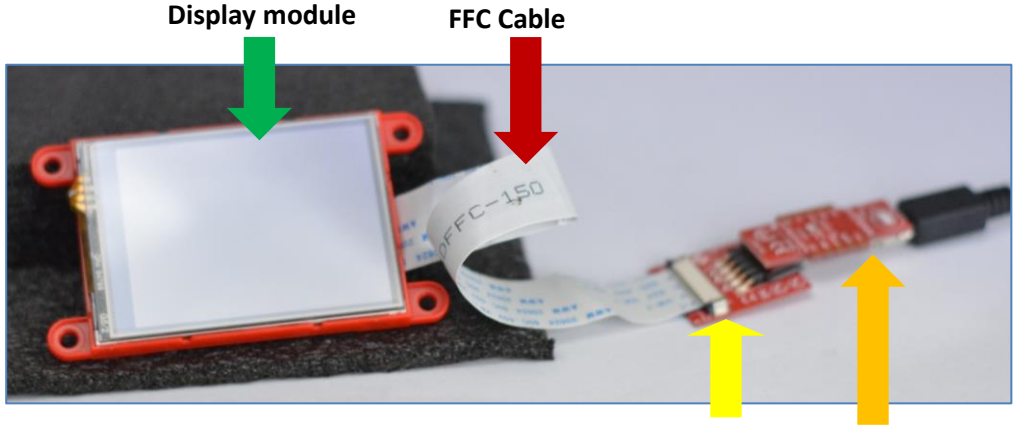
Connect the display module to the PC using a 4D USB Programming Cable, gen4-PA or a µUSB-PA5-II programming adaptor.

Note: Before using the 4D Programming Cable or the µUSB-PA5 adaptor, the drivers need to be installed first. Click any of the hyperlinks to go to their product pages. Follow the instructions on the page for installing the drivers.

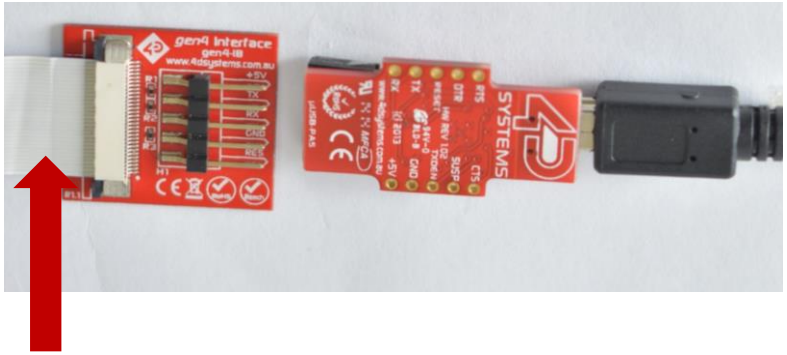
FFC Cable Connected to the gen4-display (Silver Side up)

Using the µUSB-PA5 Programming Adaptor

Complete setup:



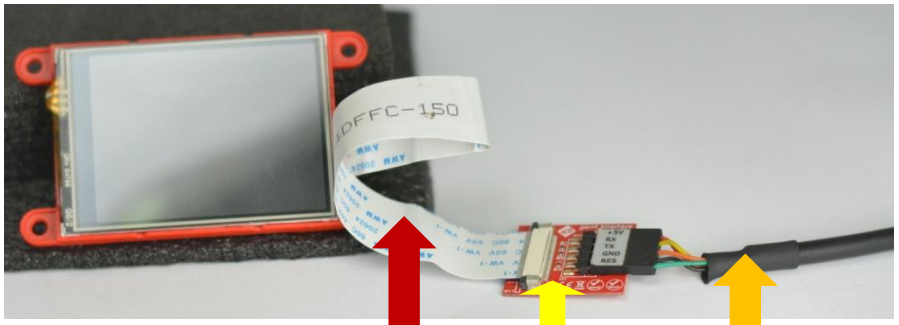
Check the orientation.



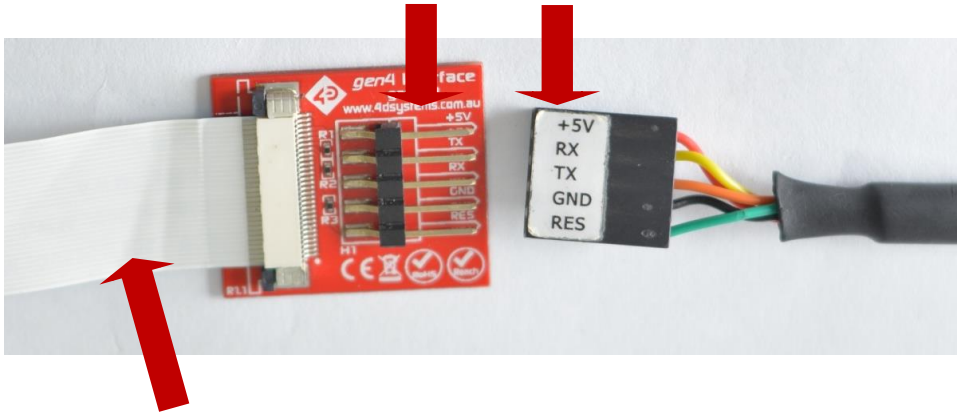
FFC Cable Connected to the Display Module (Silver Side up)

Using the 4D USB Programming Cable

Complete setup:

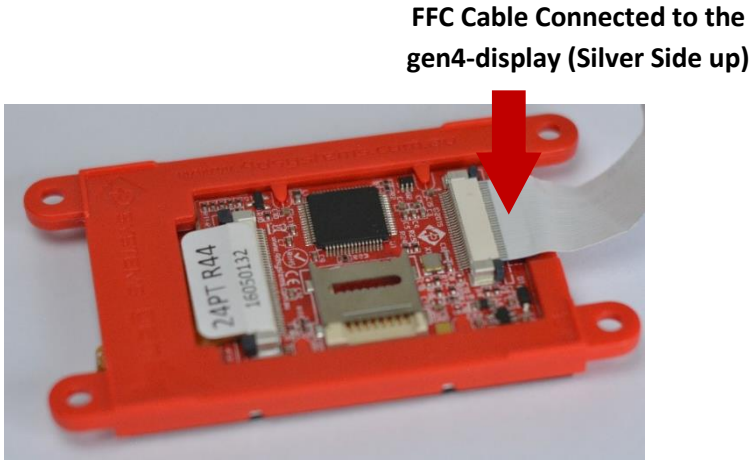


Check orientation:



Using the 4D-UPA

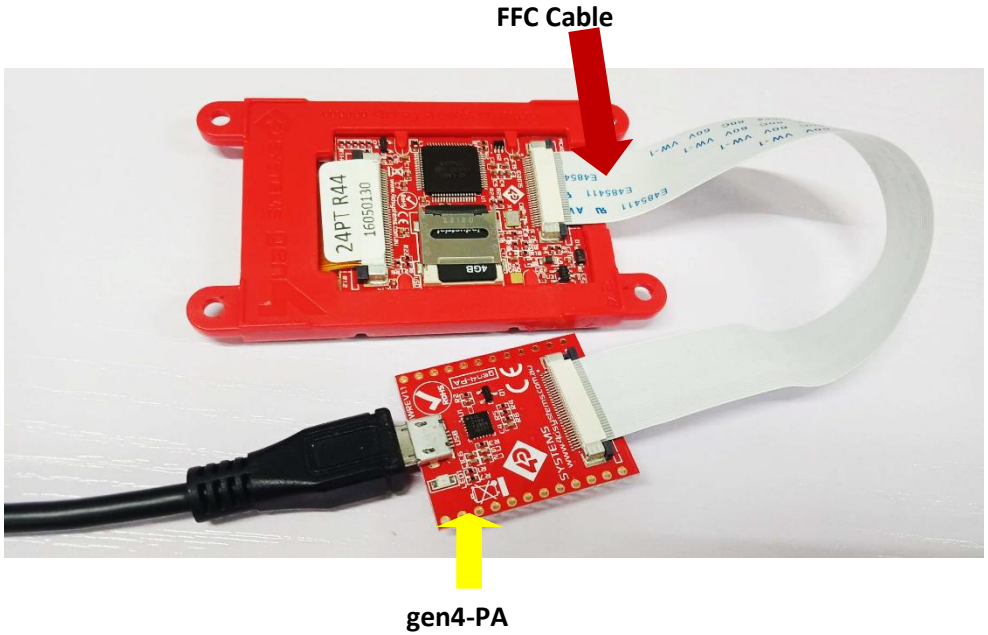
Complete setup:



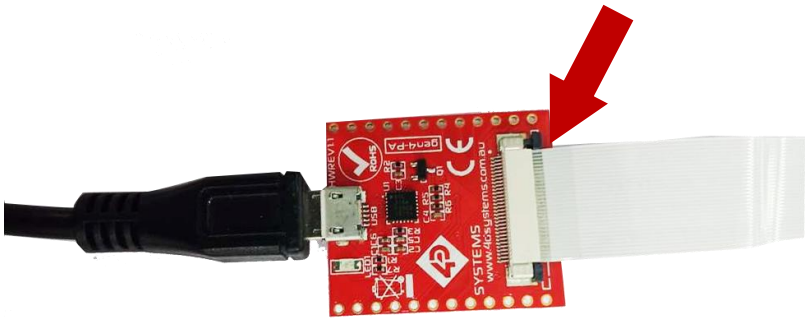
Check Orientation:

Using the gen4-PA

Complete setup:



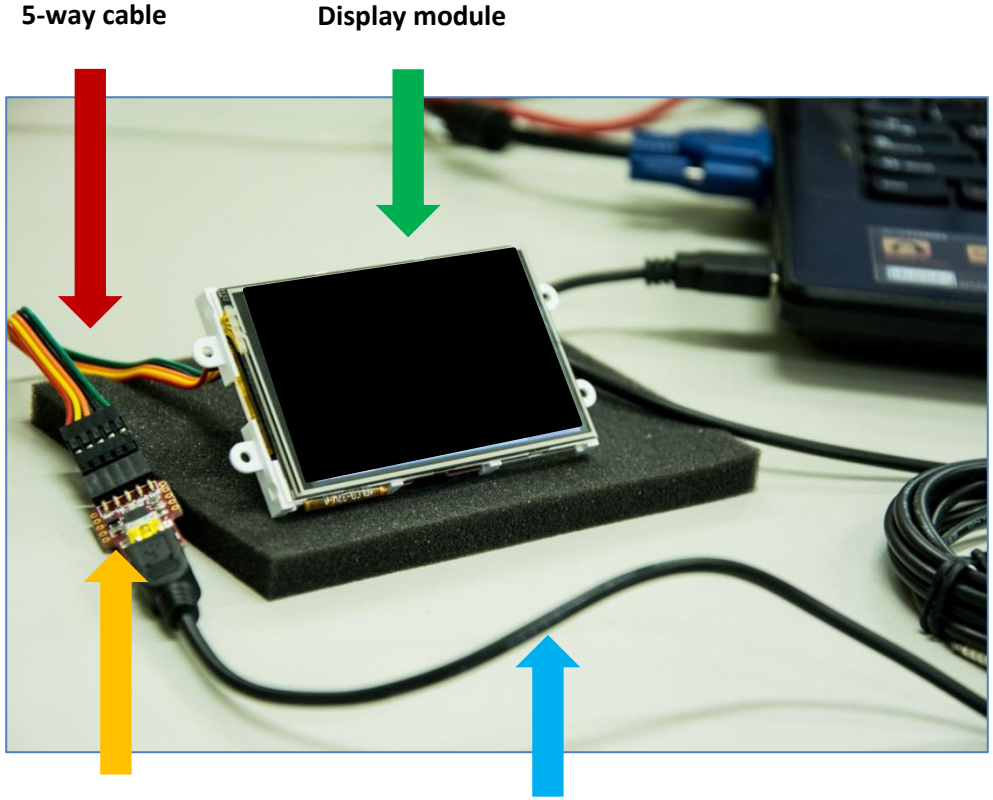
Check Orientation:

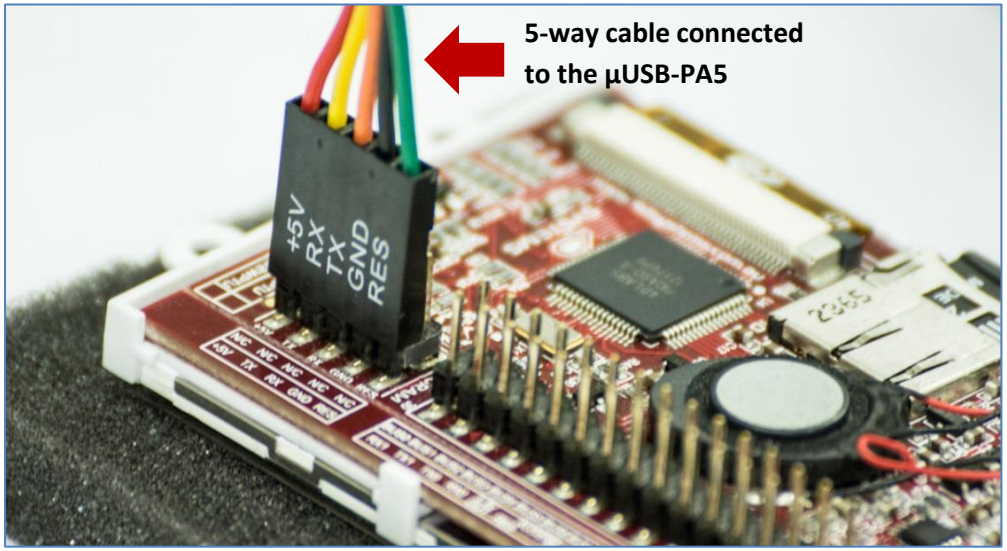


Connecting non-gen4 Display Modules

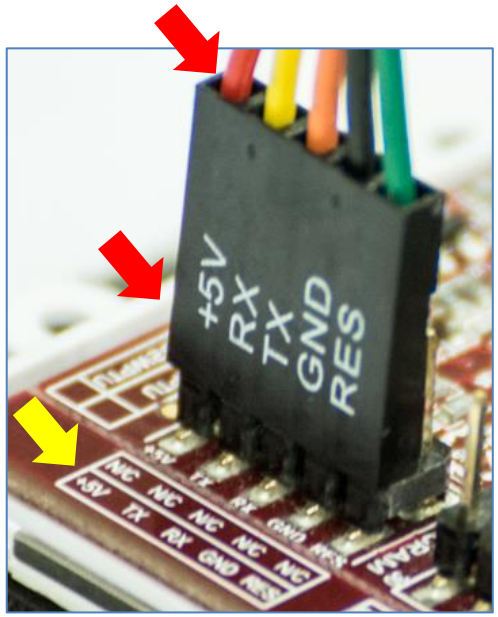
Using the µUSB-PA5-II Programming Adaptor

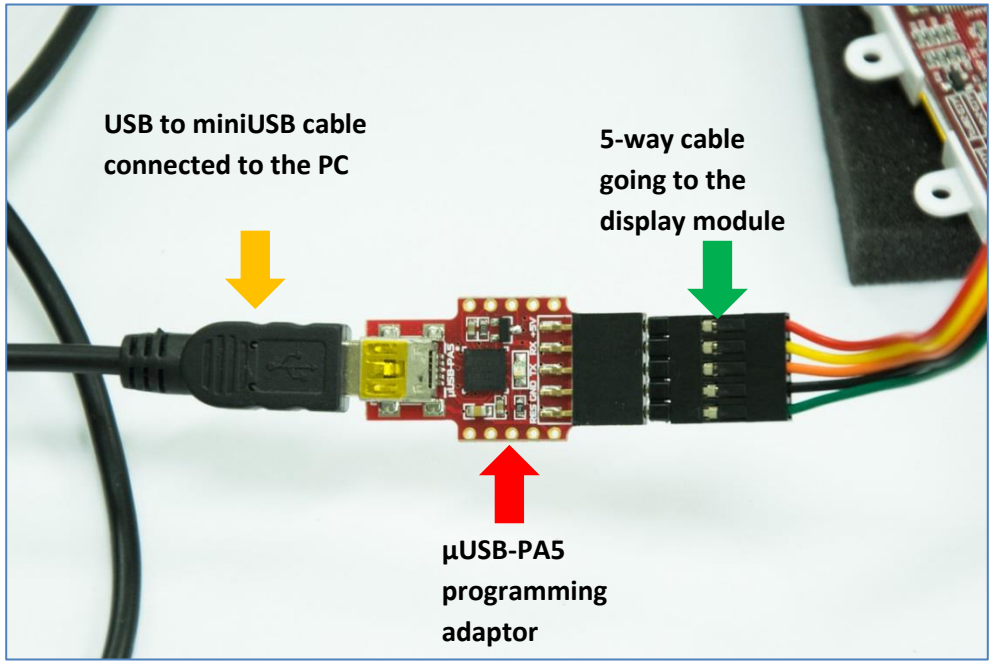
Complete setup:



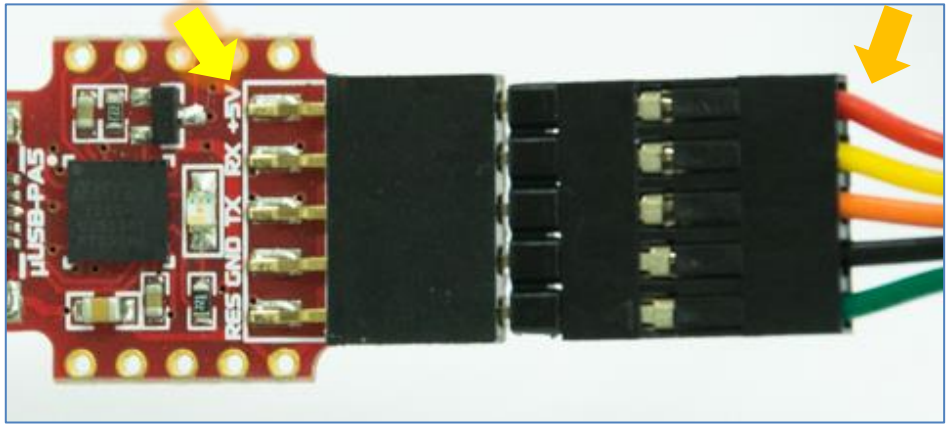


Check the orientation.

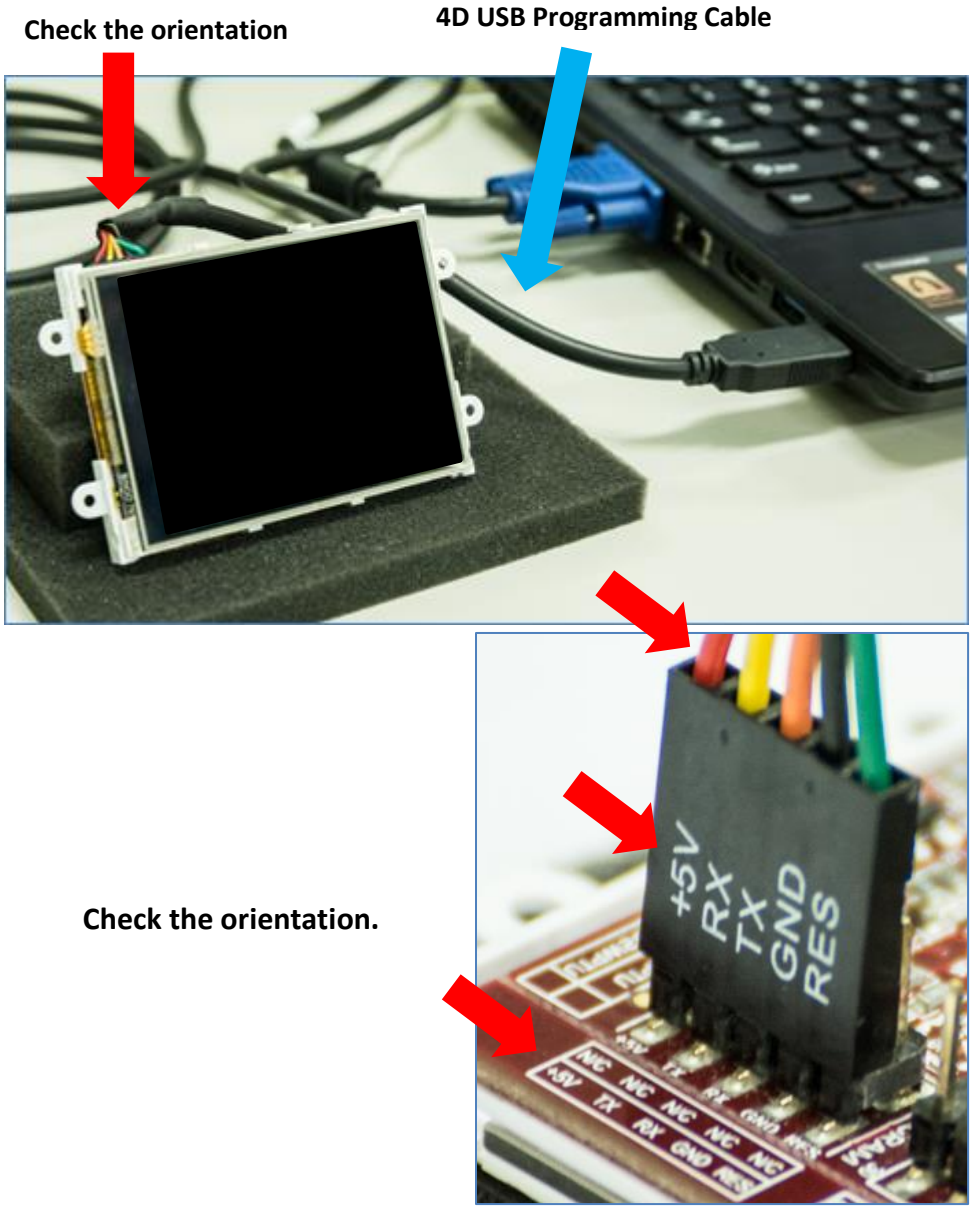




Check the orientation.

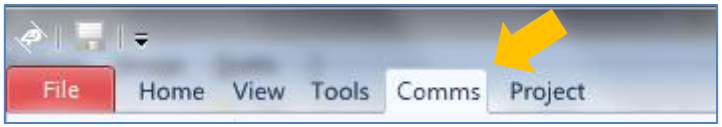


Using the 4D USB Programming Cable

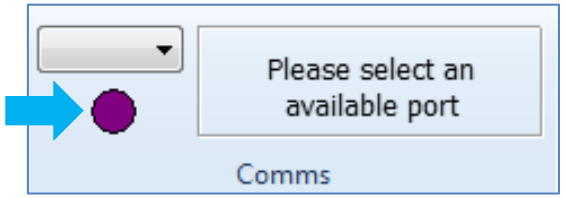


Check if the Display Module is Detected by the PC

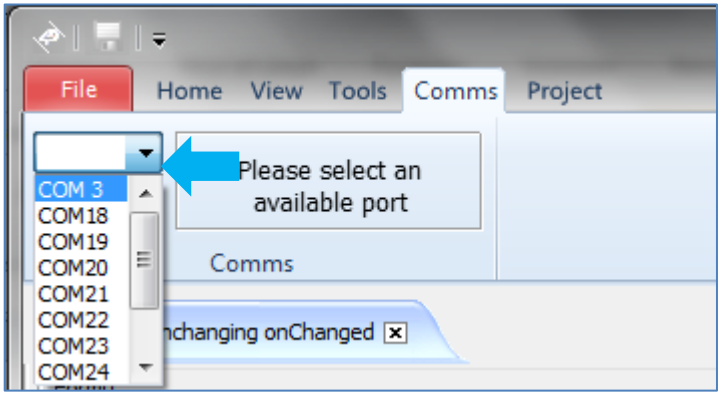
Go to the Comms menu to check if the module is detected.



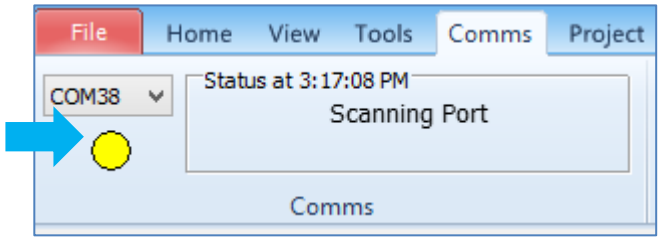
The violet light mentions no programming module is currently connected.



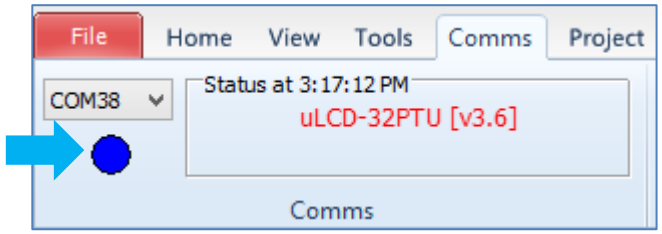
With the display module connected to the 4D USB programming cable (or µUSB-PA5), plug the cable into the USB port. Click on the drop-down list and select the COM port allocated to the cable. The product pages for the programming cable and µUSB-PA5 have instructions on how to determine the allocated COM port.



The light turns yellow while the connection is being established:



Finally, the light goes blue when the connection is established.



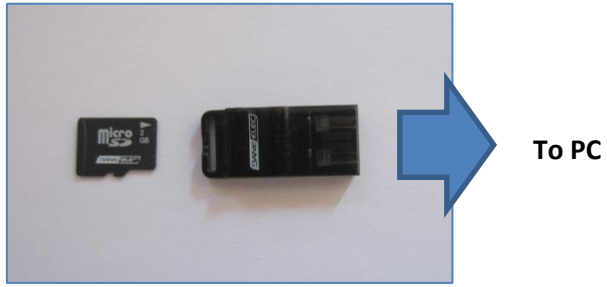
Note that the model of the target display and its current PmmC are printed in red font, which means that a later version of the PmmC is available in Workshop. In this case, the outdated PmmC on the target display is version 3.6. The latest available version is 3.8. For instructions on how to update the PmmC of the display refer to the application note **General How to Update the PmmC for Picaso**.

The light turns red when no module is attached to the selected port:
If the connected target display cannot be detected, double check all connections, ensure that the drivers are correctly installed, and verify the correct COM port allocation for the programming module. Check continuity of the 5-way cable and try replacing the USB-to-miniUSB cable (if using a uUSB-PA5) as well. Some USB-to-miniUSB cables transfer power only and not data.

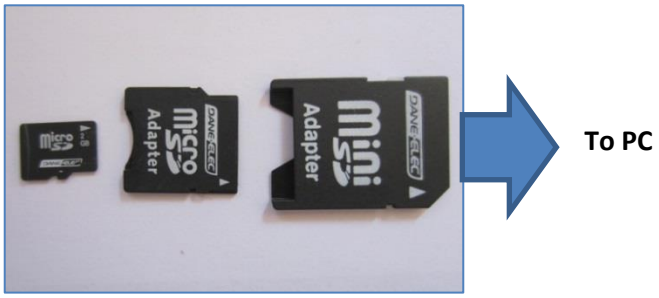
Insert the uSD Card to the PC

For Picaso display modules, a µSD card shall be FAT16-formatted, and partition can't exceed 4 GB.

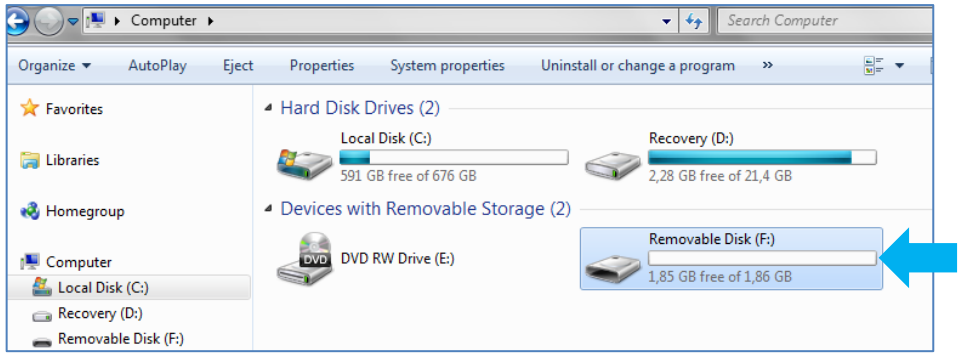
Insert the µSD card into the USB adaptor and plug the USB adaptor into a USB port of the PC.



OR insert the µSD card into a µSD to SD card converter and plug the SD card converter into the SD card slot of the PC.

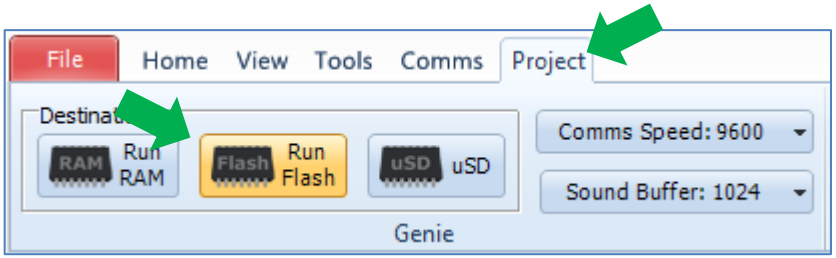


Check if the µSD card is mounted, here it is mounted as drive E:



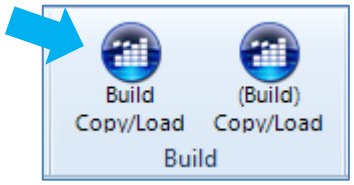
Program Destination

Choose the destination of the project. Select the **Project** menu and click on **Flash** as the destination.

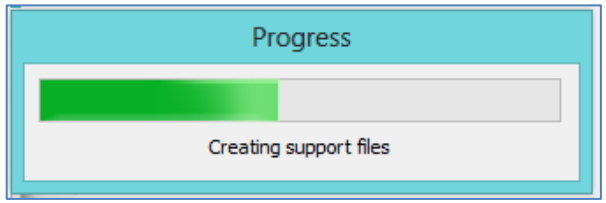


Compile and Download

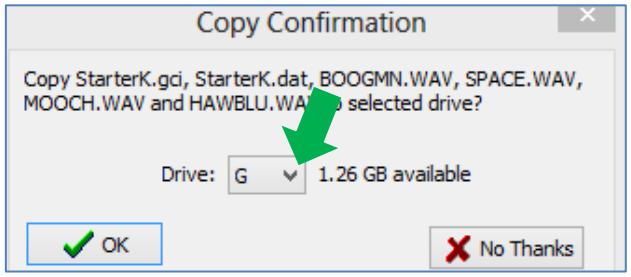
After making sure that the device is detected, go to the **Home** menu and click on the **Build Copy/Load** button. Clicking on the left icon always builds and copies the graphics files to the uSD card and downloads the program to the display module. Use this icon to be sure that the graphics files are always up-to-date, i.e., they include the latest changes made.



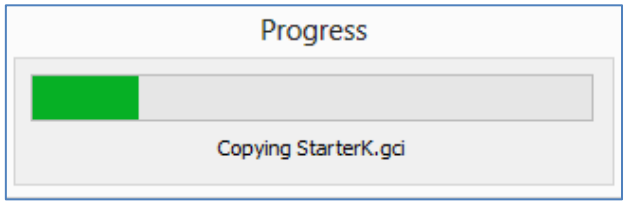
Workshop now builds and downloads the program to the display module.



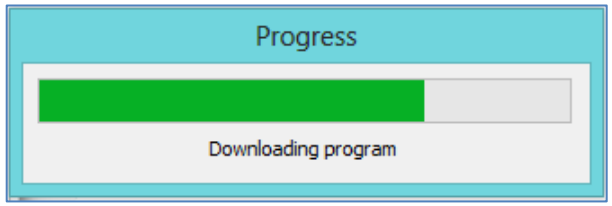
Workshop will prompt the user for the µSD card. Select the drive on the drop down list then click on OK.



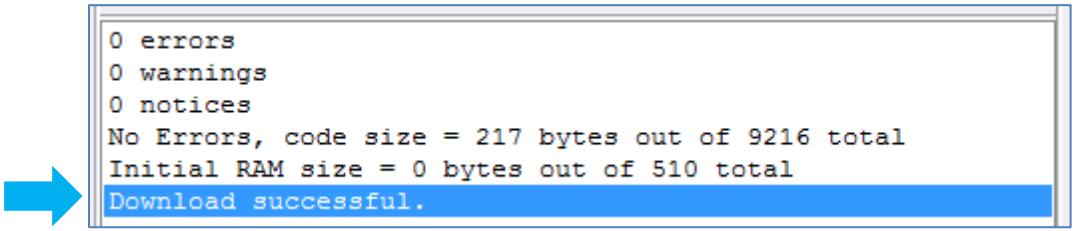
A progress bar is displayed while the necessary files are being copied to the µSD card. Workshop copies two files to the µSD card –the GCI and the DAT files. The GCI file contains the graphics and the DAT file contains a list of the objects inside the GCI file. These files will be accessed by the program when the display module is turned on.



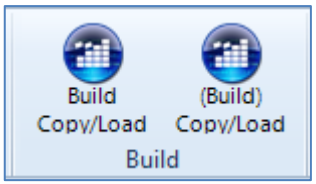
Now Workshop downloads the program to the flash memory of the display module.



Finally, the message box displays the code size and confirms that the download to the flash memory has been successful.



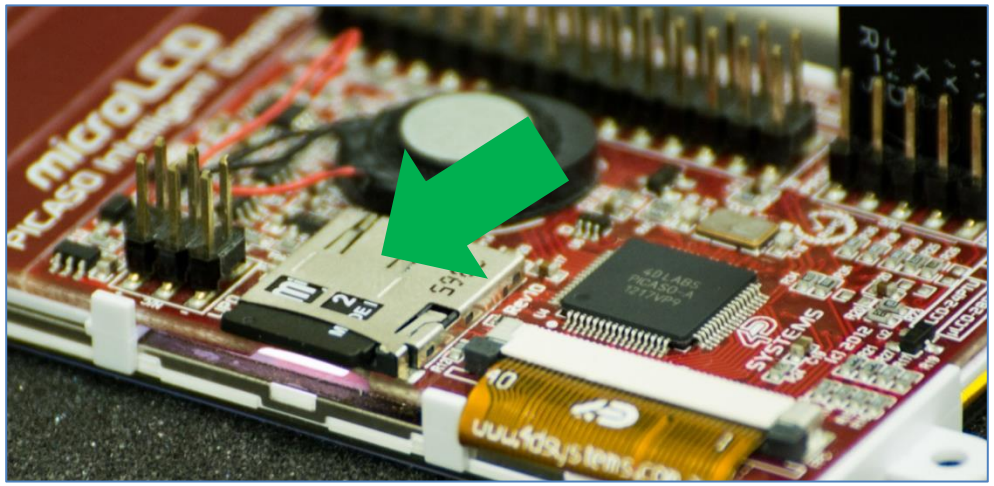
Note: There are two Build-Copy/Load buttons. The left button forces a build of the graphics files all the time. The right button builds the graphics files only when Workshop detects any change made on the WYSIWYG screen and/or the Object Inspector. When no changes are detected, clicking on the right button will simply cause Workshop to copy the graphics files to the uSD card and load the program to the display.



The right button is useful for loading a single Genie application to multiple displays and uSD cards. The left button is the better choice when the user wants to make sure that the graphics files are updated all the time. Note that for larger ViSi-Genie programs, Worskhop may take some time to build the graphics files.

Insert the µSD Card to the Display Module

Properly disconnect the µSD card from the PC and plug it to the µSD Card slot of the display module. The project now starts and runs on the screen.



Play with the project and observe how the objects interact.

Identify the Messages

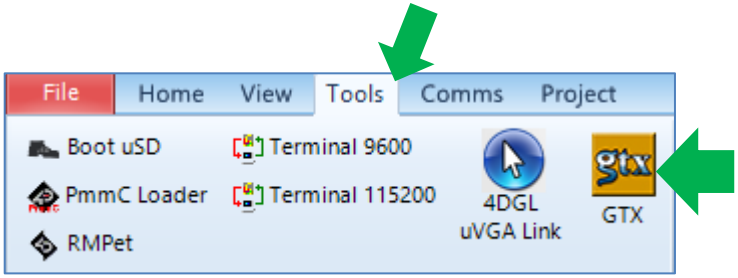
The display module is going to generate and send messages to an external host. This section explains to the user how to interpret these messages. An understanding of this section is necessary for users who intend to interface the display to a host. The **ViSi Genie Reference Manual** is recommended for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

Use the GTX Tool to Analyse the Messages

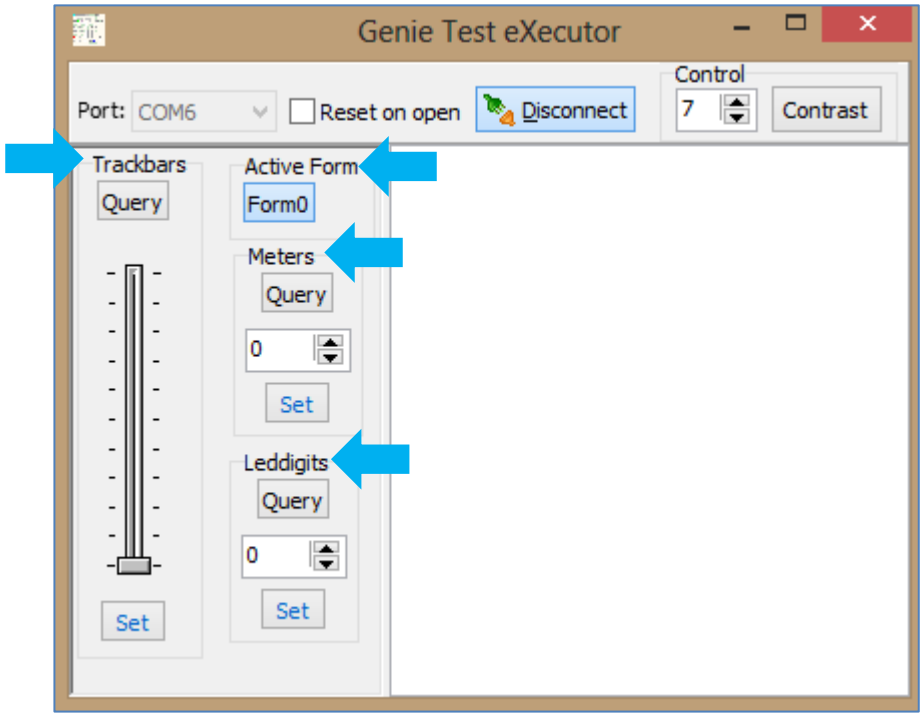
Using the GTX or **Genie Test eXecutor** tool is the first option to get the messages sent by the screen to the host. Here the PC will be the host. The GTX tool is a part of the Workshop 4 IDE. It allows the user to receive, observe, and send messages from and to the display module. It is an essential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

Launch the GTX Tool

Under Tools menu click on the GTX tool button.



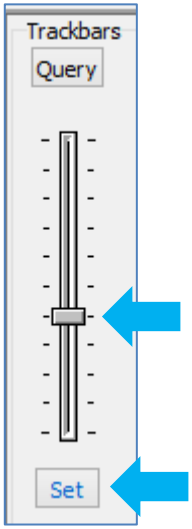
A new window appears, with the form and objects created previously.



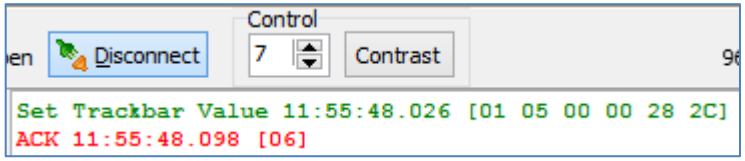
The Trackbar

Change the Status of the Trackbar

In the GTX tool window, use the mouse to move the slider of the trackbar and press **Set**. On the display module, note that the slider of the trackbar has moved.



Also, messages are sent to and received from the display module.



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

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.

The message sent is formatted according to the following pattern:

Command	Object Type	Object Index	Value MSB	Value LSB	Checksum
01	05	00	00	28	2C
WRITE_OBJ	Trackbar	First	0x0028		

The message stands for “Write to the first trackbar object on the display module the value 0x0028.” Converting the hexadecimal value 0x0028 to decimal will yield the value 40.

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.

ACK = 0x06 as shown below

```
ACK 11:55:48.098 [06]
```

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

Message from the Trackbar

Remember that Trackbar0 was configured to **Report a Message** when its status has changed. Now move the trackbar on the display module with a stylus or a finger. Observe the message sent by the display module to the PC.

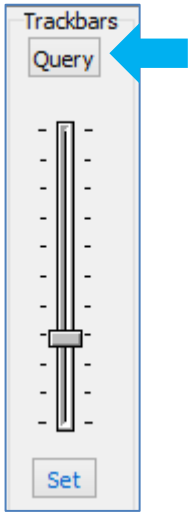
```
Trackbar Change 15:52:53.178 [07 05 00 00 50 52]
```

The message from a trackbar is formatted according to the following pattern:

Command	Object Type	Object Index	Value MSB	Value LSB	Checksum
07	05	00	00	50	52
REPORT_EVENT	Trackbar	First	0x0050		

Interrogate the Display for the Status of the Trackbar

Suppose the trackbar object is not configured to report an event when it has moved. The PC can ask the display module for the value of the trackbar by sending a message. Now on the display module randomly move the trackbar. In the GTX tool window press Query.



Observe the message area.

```
Request Trackbar Value 15:57:23.893 [00 05 00 05]
Trackbar Value 15:57:23.927 [05 05 00 00 1D 1D]
```

The PC sends a request message. The display module replies with the current value of the trackbar. The messages sent and received are formatted according to the following patterns.

Command	Object Type	Object Index	Value MSB	Value LSB	Checksum
00	05	00	-	-	05
READ_OBJ	Trackbar	First	Not applicable		
05	05	00	00	1D	1D
REPORT_OBJ	Trackbar	First	0x001D		

REPORT_EVENT vs. REPORT_OBJ

It is important to take note of the difference between REPORT_EVENT and REPORT_OBJ. **REPORT_EVENT** occurs if the user selects the event of a widget or object in Workshop to be "Report Message". There is no need for the host to ask the display module for the status or value of the object. The object independently sends its current status since it was configured to do so. Whereas **REPORT_OBJ** is a result of the user doing a read of an object from the host, using the query button.

Experimentation with the different objects using the GTX tool is now left to the user as an exercise. The following tables are shown below as a reference. Consult the **ViSi Genie Reference Manual** for more information.

Command	Code	Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter N	Checksum
READ_OBJ	0x00	Object ID	Object Index	-	-	-	Checksum
WRITE_OBJ	0x01	Object ID	Object Index	Value (msb)	Value(lsb)	-	Checksum
WRITE_STR	0x02	String Index	String Length	String (1 byte chars)			Checksum
WRITE_ STRU	0x03	String Index	String Length	String (2 byte chars)			Checksum
WRITE_ CONTRAST	0x04	Value	-	-	-	-	Checksum
REPORT_OBJ	0x05	Object ID	Object Index	Value (msb)	Value(lsb)	-	Checksum
REPORT_EVENT	0x07	Object ID	Object Index	Value (msb)	Value(lsb)	-	Checksum
WRITE_MAGIC_BYTES	0x08	Object Index	Length	Array (1 byte values)			Checksum
WRITE_MAGIC_DBYTES	0x09	Object Index	Length	Array (2 byte values)			Checksum
REPORT_MAGIC_EVENT_BYTES	0x0A	Object Index	Length	Array (1 byte values)			Checksum
REPORT_MAGIC_EVENT_DBYTES	0x0B	Object Index	Length	Array (2 byte values)			Checksum

This table is found in section **Protocol Definitions** of the **ViSi Genie Reference Manual**.

Object	ID	Input	Output	Notes
Dipswitch	0 (0x00)	✓	✓	
Knob	1 (0x01)	✓	✓	
Rockerswitch	2 (0x02)	✓	✓	
Rotaryswitch	3 (0x03)	✓	✓	
Slider	4 (0x04)	✓	✓	
Trackbar	5 (0x05)	✓	✓	
Winbutton	6 (0x06)	✓	✓	
Angularmeter	7 (0x07)		✓	
Coolgauge	8 (0x08)		✓	
Customdigits	9 (0x09)		✓	
Form	10 (0x0A)		✓	Used to set the current form
Gauge	11 (0x0B)		✓	
Image	12 (0x0C)			Displayed as part of form, no method to alter
Keyboard	13 (0x0D)	✓		Keyboard inputs are always single bytes and are unsolicited
Led	14 (0x0E)		✓	
Leddigits	15 (0x0F)		✓	
Meter	16 (0x10)		✓	
Strings	17 (0x11)		✓	
Thermometer	18 (0x12)		✓	
Userled	19 (0x13)		✓	
Video	20 (0x14)		✓	
Statictext	21 (0x15)			Displayed as part of form, no method to alter
Sound	22 (0x16)		✓	
Timer	23 (0x17)		✓	
Spectrum	24 (0x18)		✓	
Scope	25 (0x19)		✓	
Tank	26 (0x1A)		✓	
UserImages	27 (0x1B)		✓	
PinOutput	28 (0x1C)		✓	
PinInput	29 (0x1D)	✓		
4Dbutton	30 (0x1E)	✓	✓	
AniButton	31 (0x1F)	✓	✓	
ColorPicker	32 (0x20)	✓	✓	
UserButton	33 (0x21)	✓	✓	
MagicObject	34 (0x22)		✓	This object is only available with Workshop4 PRO
SmartGauge	35 (0x23)		✓	This object is only available with Workshop4 PRO
SmartSlider	36 (0x24)	✓	✓	This object is only available with Workshop4 PRO
SmartKnob	37 (0x25)	✓	✓	This object is only available with Workshop4 PRO

This table is found in **Object Summary Table** of the **ViSi Genie Reference Manual**.

Proprietary Information

Disclaimer of Warranties & Limitation of Liability