

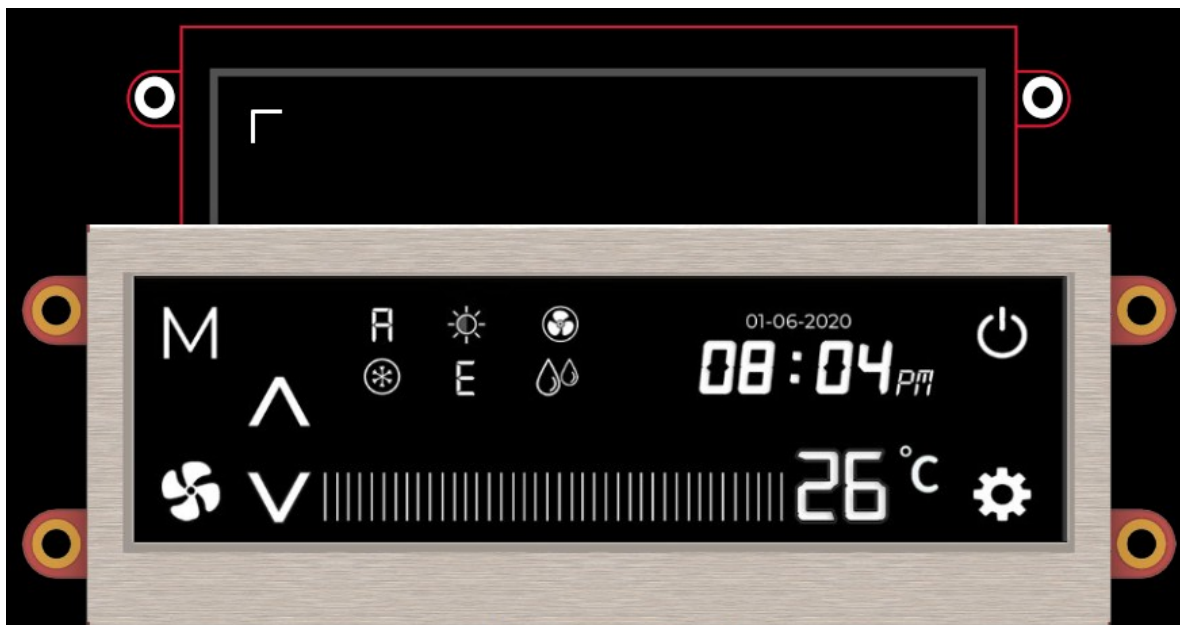


4D SYSTEMS pixxiLCD-13P2-CTP-CLB Display Arduino Platform Evaluation Expansion Board User Guide

[Home](#) » [4D SYSTEMS](#) » 4D SYSTEMS pixxiLCD-13P2-CTP-CLB Display Arduino Platform Evaluation Expansion Board User Guide 



USER GUIDE
pixxiLCD SERIES
pixxiLCD-13P2/CTP-CLB
pixxiLCD-20P2/CTP-CLB
pixxiLCD-25P4/CTP
pixxiLCD-39P4/CTP



pixxiLCD Series

SCREEN SIZE		RESOLUTION	TOUCH TYPE			PROCESSORS	
inches	mm		Non-Touch	Resistive	Capacitive	PIXXI-28	PIXXI-44
1.3	33.02	240 x 240	◆		◆	◆	
2.0	50.80	176 x 220	◆		◆	◆	
2.5	63.50	240 x 240	◆		◆		◆
3.9	99.06	480 x 128	◆		◆		◆

*Also available in Cover Lens Bezel (CLB) version.

VARIANTS:

PIXXI Processor (P2)

PIXXI Processor (P4)

Non Touch (NT)

Capacitive Touch (CTP)

Capacitive Touch with Cover Lens Bezel (CTP-CLB)

This user guide will help you started using the pixxiLCD-XXP2/P4-CTP/CTP-CLB modules along with the WorkShop4 IDE. It also includes a list of essential project examples and application notes.

Contents

1 What's In The Box

2 Introduction

3 System Requirements

3.1 Hardware

3.2 Software

4 Connection Options

4.1 Option A – Using the 4D-UPA

4.2 Option B – Using the 4D Programming Cable

4.3 Option C – Using the uUSB-PA5-II

5 Let WS4 Identify the Display Module

6 Getting Started With A Simple Project

7 Application Notes

8 Reference Documents

9 GLOSSARY

9.1 Hardware

9.2 Software

10 Documents / Resources

10.1 References

11 Related Posts

What's In The Box



Supporting documents, datasheet, CAD step models and application notes are available at
www.4dsystems.com.au



Introduction

This User Guide is an introduction to becoming familiar with the pixxiLCDXXP2/P4-CT/CT-CLB and the software IDE associated with it. This manual should be treated only as a useful starting point and not as a comprehensive reference document. Refer to Application Notes for a list of all the detailed reference documents.

In this User Guide, we will briefly focus on the following topics:

- Hardware and Software Requirements
- Connecting the Display Module to your PC
- Getting Started with Simple Projects
- Projects using pixxiLCD-XXP2/P4-CT/CT-CLB
- Application Notes
- Reference Documents

The pixxiLCD-XXP2/P4-CT/CT-CLB is part of the Pixxi series of display modules designed and manufactured by 4D Systems. The module features a 1.3" round, 2.0", 2.5" or 3.9 colour TFT LCD display, with optional capacitive touch. It is powered by the feature-rich 4D Systems Pixxi22/Pixxi44 graphics processor, which offers an array of functionality and options for the designer/integrator/user.

Intelligent display modules are low-cost embedded solutions used in various applications in the medical, manufacturing, military, automotive, home automation, consumer electronics, and other industries. In fact, there are very few embedded designs on the market today that do not have a display. Even many consumer white goods and kitchen appliances incorporate some form of display. Buttons, rotary selectors, switches and other input devices are being replaced by more colourful and easier-to-use touch screen displays in industrial machines, thermostats, drink dispensers, 3D printers, commercial applications – virtually any electronic application.

For designers/users to be able to create and design a user interface for their applications that will run on 4D intelligent display modules, 4D Systems provides a free and user-friendly software IDE (Integrated Development Environment) called "Workshop4" or "WS4". This software IDE is discussed in more detail in the section "System Requirements".

System Requirements

The following sub-sections discuss the hardware and software requirements for this manual.

Hardware

1. Intelligent Display Module and Accessories

The pixxiLCD-xxP2/P4-CT/CT-CLB intelligent display module and its accessories (adaptor board and flat flex cable) are included in the box, delivered to you after your purchase from our website or through one of our distributors. Please refer to the section “What’s in the Box” for images of the display module and its accessories.

2. Programming Module

The programming module is a separate device required to connect the display module to a Windows PC. 4D Systems offer the following programming module:

- 4D Programming Cable
- uUSB-PA5-II Programming Adaptor
- 4D-UPA

To use the programming module, the corresponding driver must first be installed in the PC.

You may refer to the product page of the given module for more information and detailed instruction.

NOTE: This device is available separately from 4D Systems. Please refer to the product pages for more information.

3. Media Storage

Workshop4 has built-in widgets that can be used to design your display UI. Most of these widgets are required to be stored in storage device, such as a microSD Card or an external flash, along with the other graphic files during the compilation step.

NOTE: microSD Card and external flash is optional and is only needed with projects that are utilizing graphical files.

Please note as well that not all microSD cards on the market are SPI compatible, and therefore not all cards can be used in 4D Systems products. Buy with confidence, choose the cards recommended by 4D Systems.

4. Windows PC

Workshop4 only runs on Windows operating system. It is recommended to be used on Windows 7 up to Windows 10 but should still work with Windows XP. Some older OS’s such as ME and Vista has not been tested for quite some time, however, the software should still work.

If you want to run the Workshop4 on other operating systems like Mac or Linux, it is recommended to set up a virtual machine (VM) on your PC.

Software

1. Workshop4 IDE

Workshop4 is a comprehensive software IDE for Microsoft Windows that provides an integrated software development platform for all of the 4D family of processors and modules. The IDE combines the Editor, Compiler, Linker and Downloader to develop complete 4DGL application code. All user application code is developed within the Workshop4 IDE.

Workshop4 includes three development environments, for the user to choose based on application requirements or even user skill level- Designer, ViSi–Genie, and ViSi.

Workshop4 Environments

Designer

This environment enables the user to write 4DGL code in its natural form to program the display module.

ViSi – Genie

An advanced environment that doesn’t require any 4DGL coding at all, it is all done automatically for you. Simply lay the display out with the objects you want (similar to ViSi), set the events to drive them and the code is written

for you automatically. ViSi-Genie provides the latest rapid development experience from 4D Systems.

ViSi

A visual programming experience that enables drag-and-drop type placement of objects to assist with 4DGL code generation and allows the user to visualize how the display will look while being developed.

2. Install Workshop4

Download links for the WS4 installer and installation guide can be found on the Workshop4 product page.

Connecting The Display Module To The Pc

This section shows the complete instructions for connecting the display to the PC. There are three (3) options of instructions under this section, as shown in the images below. Each option is specific to a programming module. Follow only the instructions applicable to the programming module that you are using.

HARDWARE

SOFTWARE

Option A



Display Module



FFC Cable



4D-UPA



Workshop4

Option B



Display Module



FFC Cable



gen4-IB



4D Prog. Cable



Workshop4

Option C



Display Module



FFC Cable



gen4-IB



uUSB-PA5-II



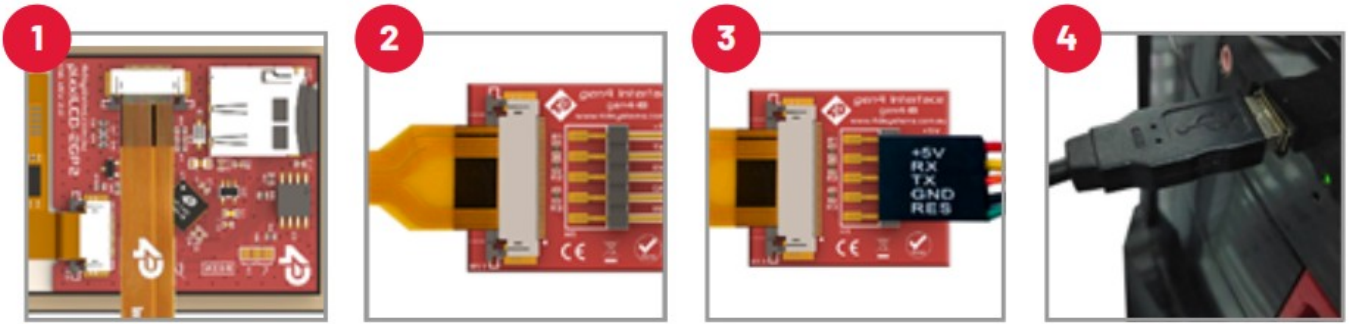
Workshop4

Connection Options

Option A – Using the 4D-UPA

1. Connect one end of the FFC to the pixxiLCD's 15-way ZIF socket with the metal contacts on the FFC facing on the latch.
2. Connect the other end of the FFC to the 30-way ZIF socket on the 4D-UPA with the metal contacts on the FFC facing on the latch
3. Connect the USB-Micro-B Cable to the 4D-UPA.

4. Lastly, connect the other end of the USB-Micro-B Cable to the computer.



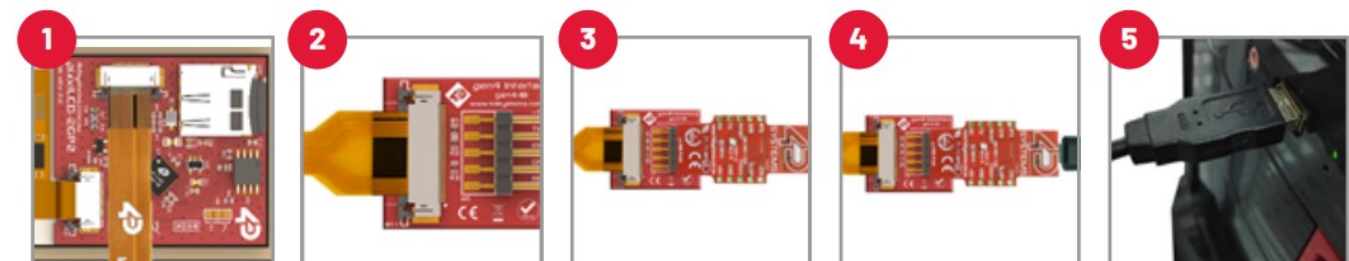
Option B – Using the 4D Programming Cable

1. Connect one end of the FFC to the pixxiLCD's 15-way ZIF socket with the metal contacts on the FFC facing on the latch.
2. Connect the other end of the FFC to the 30-way ZIF socket on the gen4-IB with the metal contacts on the FFC facing on the latch.
3. Connect the 5-Pin female header of the 4D Programming Cable to the gen4-IB following the orientation on both cable and module labels. You can also do this with the assistance of the supplied ribbon cable.
4. Connect the other end of the 4D Programming Cable to the computer.



Option C – Using the uUSB-PA5-II

1. Connect one end of the FFC to the pixxiLCD's 15-way ZIF socket with the metal contacts on the FFC facing on the latch.
2. Connect the other end of the FFC to the 30-way ZIF socket on the gen4-IB with the metal contacts on the FFC facing on the latch.
3. Connect the 5-Pin female header of the uUSB-PA5-II to the gen4-IB following the orientation on both cable and module labels. You can also do this with the assistance of the supplied ribbon cable.
4. Connect a USB-Mini-B Cable to the uUSB-PA5-II.
5. Lastly, connect the other end of the uUSB-Mini-B to the computer.



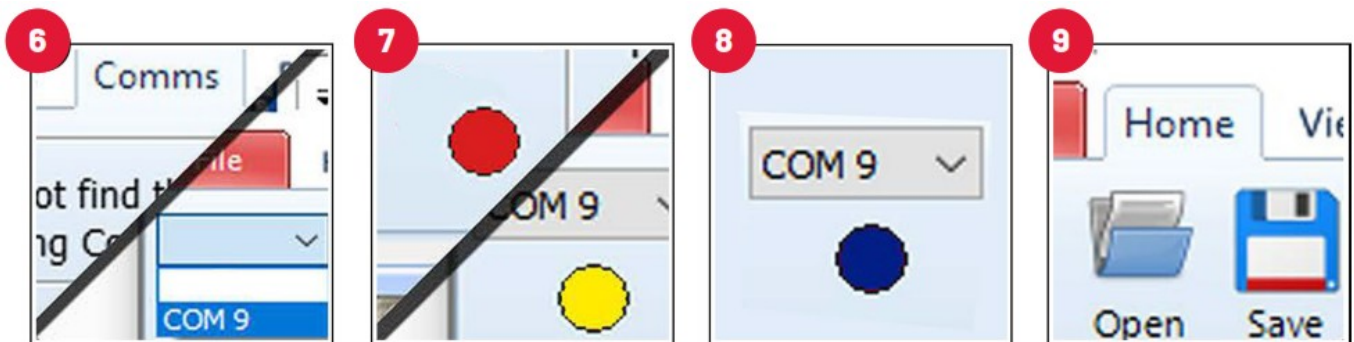
Let WS4 Identify the Display Module

After following the appropriate set of instructions in the previous section, you now need to configure and setup Workshop4 to make sure that it identifies and connects to the correct display module.

1. Open Workshop4 IDE and create a new project.
2. Select the display module you're using from the list.
3. Select your desired orientation for your project.
4. Click next.
5. Choose a WS4 Programming Environment. Only the compatible programming environment for the display module will be enabled.



6. Click on the COMMS tab, select the COM port the display module is connected to from the dropdown list.
7. Click on the RED Dot to start scanning for the display module. A YELLOW dot will show while scanning. Make sure that your module is connected properly.
8. Lastly, a successful detection will give you a BLUE Dot with the name of the display module shown alongside it.
9. Click on the Home tab to start creating your project.



Getting Started With A Simple Project

After successfully connecting the display module to the PC using your programming module, you can now start creating a basic application. This section shows how to design a simple user interface using the ViSi-Genie environment and utilizing the slider and gauge widgets.

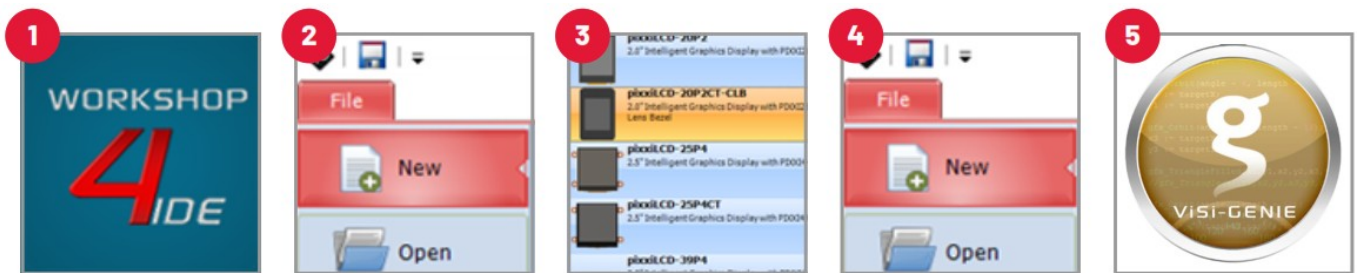
The resulting project consists of a slider (an input widget) controlling a gauge (an output widget). The widgets can also be configured to send event messages to an external host device through the serial port.

Create a New ViSi-Genie Project

You can create a ViSi-Genie project by opening Workshop and by choosing the display type and the environment that you want to work with. This project will be using the ViSi-Genie environment.

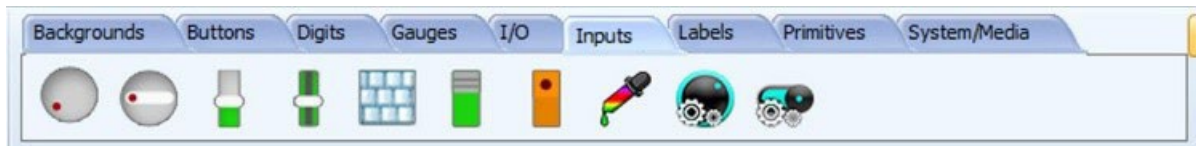
1. Open Workshop4 by double-clicking the icon.
2. Create New Project with the New Tab.

3. Choose your display type.
4. Click Next.
5. Choose ViSi-Genie Environment.

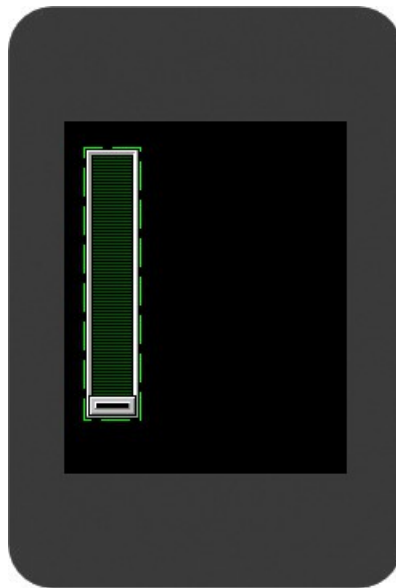


Add a Slider Widget

To add a slider widget, simply click on the Home tab and choose the Inputs Widgets. From the list, you may choose the type of widget that you want to use. In this case, the slider widget is selected.



Simply drag and drop the widget towards the What-You-See-Is-What-You-Get (WYSIWYG) section.



Add a Gauge Widget

To add a gauge widget, go to the Gauges section and choose the gauge type that you want to use. In this case the Coolgauge widget is selected.



Drag and drop it towards the WYSIWYG section to proceed.

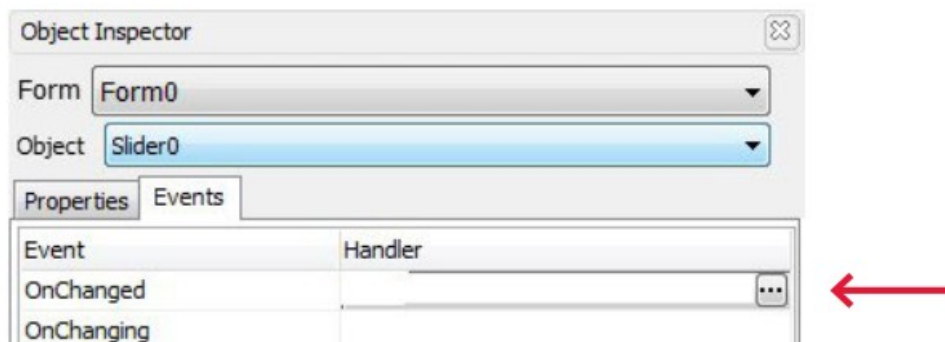


Link the Widget

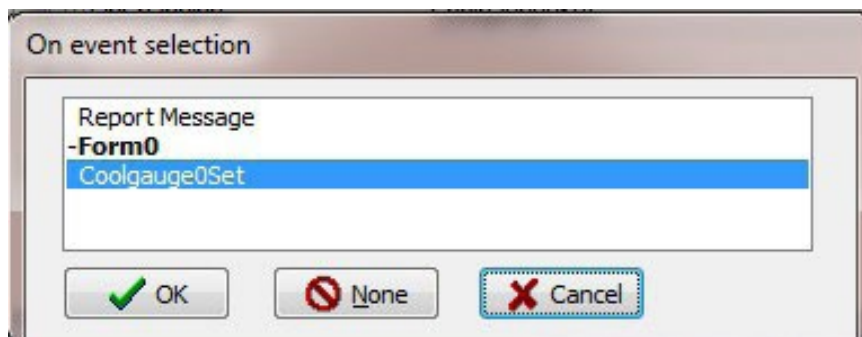
Input widgets can be configured to control an output widget. To do this, just click on the input (in this example, the slider widget) and go to its Object Inspector Section and click the Events Tab.

There are two events available under the events tab of an input widget – OnChanged and OnChanging. These events are triggered by touch actions performed on the input widget.

The OnChanged event is triggered every time an input widget is released. On the other hand, the OnChanging event is continuously triggered while an input widget is being touched. In this example, the OnChanging event is used. Set the event handler by clicking on the ellipsis symbol for the OnChanging event handler.

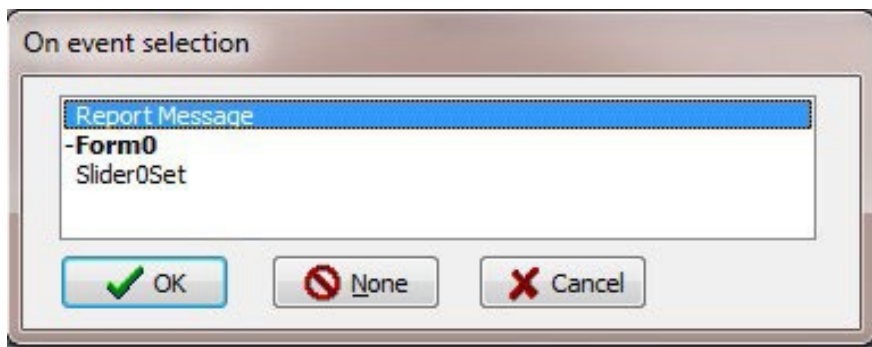


The on-event selection window appears. Select coolgauge0Set, then click OK.



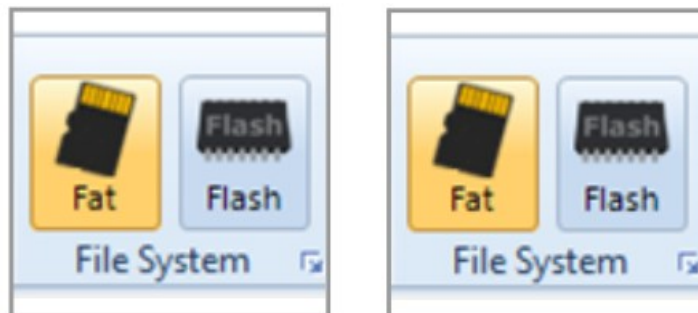
Configure the Input Widget to Send Messages to a Host

An external host, connected to the display module through the serial port, can be made aware of the status of a widget. This can be achieved by configuring the widget to send event messages to the serial port. To do this, set the OnChanged event handler of the slider widget to Report Message.



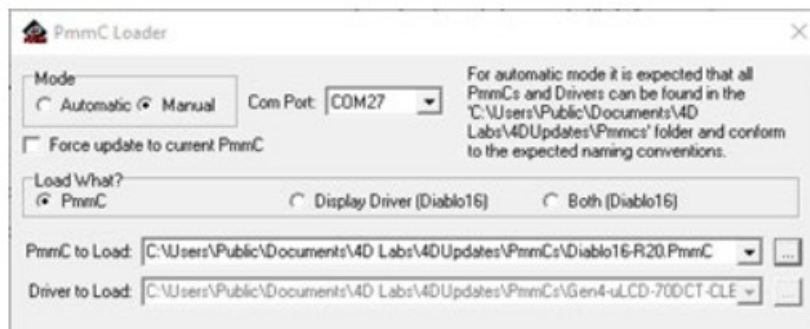
microSD Card / On-board Serial Flash Memory

On Pixxi display modules, the graphics data for the widgets can be stored to the microSD card/On-board Serial Flash Memory, which will be accessed by the graphics processor of the display module during runtime. The graphics processor will then render the widgets on the display.



The appropriate PmmC must also be uploaded to the Pixxi module to use the respective storage device. The PmmC for microSD card support has the suffix "-u" while the PmmC for on-board serial flash memory support has the suffix "-f".

To manually upload the PmmC, click the Tools Tab, and select the PmmC Loader.



	pixxiLCD-20P2CT-CLB-f-A-R15.PmmC	6/25/2020 12:32 PM	PMMC File	125 KB
	pixxiLCD-20P2CT-CLB-u-A-R15.PmmC	6/26/2020 4:53 AM	PMMC File	125 KB
	pixxiLCD-20P2-f-A-R15.PmmC	6/25/2020 12:31 PM	PMMC File	125 KB
	pixxiLCD-20P2T-f-A-R15.PmmC	6/25/2020 12:32 PM	PMMC File	125 KB
	pixxiLCD-20P2T-u-A-R15.PmmC	6/26/2020 4:53 AM	PMMC File	125 KB
	pixxiLCD-20P2-u-A-R15.PmmC	6/26/2020 4:54 AM	PMMC File	125 KB

Build and Compile the Project

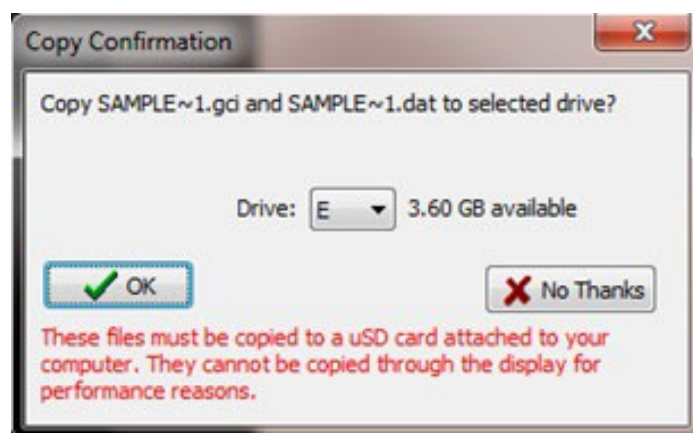
To Build/Upload the project, click the (Build) Copy/Load icon.



Copy the Required Files to the microSD Card / On-board Serial Flash Memory

microSD card

WS4 generates the required graphics files and will prompt you for the drive to which the microSD card is mounted. Make sure that the microSD card is properly mounted to the PC, then select the correct drive in the Copy Confirmation window, as shown in the image below.

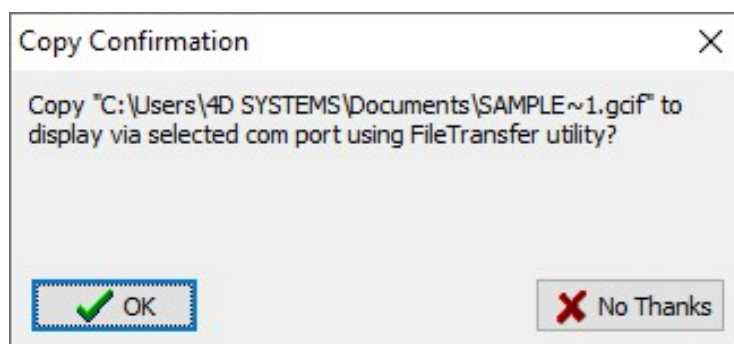


Click OK after the files are transferred to the microSD card. Unmount the microSD Card from the PC and insert it to the display module's microSD Card slot.

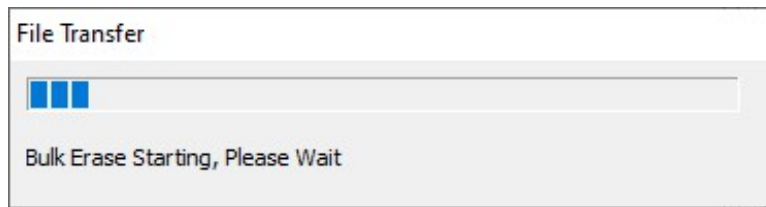
On-board Serial Flash Memory

When selecting the Flash Memory as the destination for the graphics file, make sure that no microSD card is connected in the module

A Copy Confirmation window will pop-up as shown in the message below.



Click OK, and a File Transfer window will pop-up. Wait for the process to end and the graphics will now show on the display module.

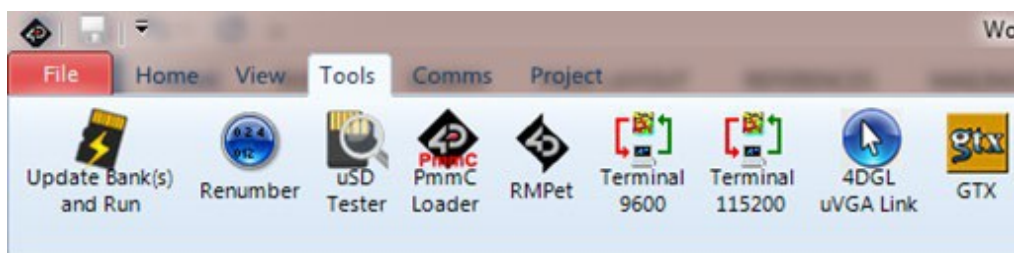


Test the Application

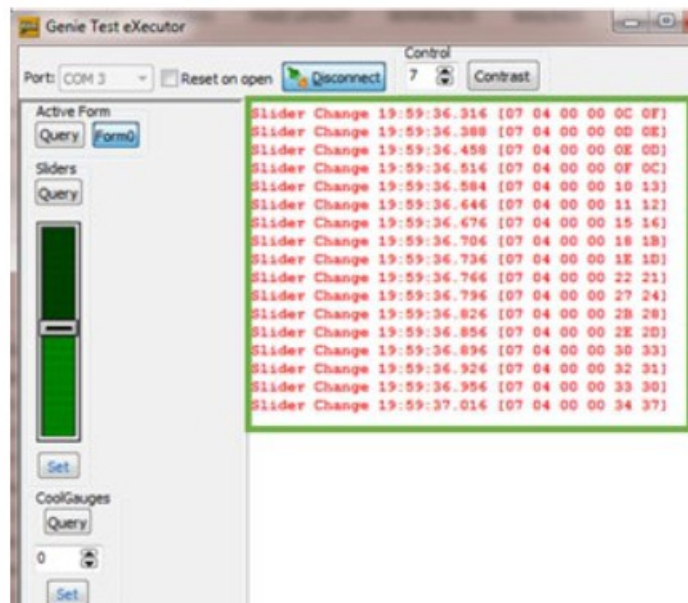
The application should now run on the display module. The slider and gauge widgets should now be shown. Start touching and moving the thumb of the slider widget. A change in its value should also result to a change in the value of the gauge widget, since the two widgets are linked.

Use the GTX Tool to Check the Messages

There is a tool in WS4 used for checking the event messages being sent out by the display module to the serial port. This tool is called "GTX", which stands for "Genie Test eXecutor". This tool can also be thought of as a simulator for an external host device. The GTX tool can be found under the Tools section. Click on the icon to run the tool.



Moving and releasing the thumb of the slider will cause the application to send event messages to the serial port. These messages will then be received and be printed by the GTX Tool. For more information on the details of the communication protocol for ViSiGenie applications, refer to the ViSi-Genie Reference Manual. This document is described in the section "Reference Documents".



Messages
received by
the host

Application Notes

App Note	Title	Description	Supported Environment
4D-AN-00117	Designer Getting Started – First Project	This application note shows how to create a new project using the Designer Environment. It also introduces the basics of 4DGL(4D Graphics Language).	Designer
4D-AN-00204	ViSi Getting Started – First Project for Pixxi	This application note shows how to create a new project using the ViSi Environment. It also introduces the basics of 4DGL(4D Graphics Language and the basic use of the WYSIWYG (What-You-See-Is-What-You-Get) screen.	ViSi
4D-AN-00203	ViSi Genie Getting Started – First Project for Pixxi Displays	The simple project developed in this application note demonstrates basic touch functionality and object interaction using the ViSi-Genie Environment. The project illustrates how input objects are configured to send messages to an external host controller and how these messages are interpreted.	ViSi-Genie

Reference Documents

ViSi-Genie is the environment recommended for beginners. This environment doesn't necessarily involve coding, which makes it the most user-friendly platform among the four environments.

However, ViSi-Genie has its limitations. For users wanting more control and flexibility during application design and development, the Designer, or ViSi environments are recommended. ViSi and Designer allow users to write the code for their applications.

The programming language used with 4D Systems graphics processors is called "4DGL". Essential reference documents that can be utilized for further study of the different environments are listed below.

ViSi-Genie Reference Manual

ViSi-Genie does all the background coding, no 4DGL to learn, it does it all for you. This document covers the ViSi-Genie functions available for the PIXXI, PICASO and the DIABLO16 Processors and the communications protocol used known as the Genie Standard Protocol.

4DGL Programmer Reference Manual

4DGL is a graphics oriented language allowing rapid application development. An extensive library of graphics, text and file system functions and the ease of use of a language that combines the best elements and syntax structure of languages such as C, Basic, Pascal, etc. This document covers the language style, the syntax and flow control.

Internal Functions Manual

4DGL has a number of internal functions that can be used for easier programming. This document covers the internal (chip-resident) functions available for the pixxi Processor.

pixxiLCD-13P2/P2CT-CLB Datasheet

This document contains detailed information about the pixxiLCD-13P2/P2CT-CLB integrated display modules.

pixxiLCD-20P2/P2CT-CLB Datasheet

This document contains detailed information about the pixxiLCD-20P2/P2CT-CLB integrated display modules.

pixxiLCD-25P4/P4CT Datasheet

This document contains detailed information about the pixxiLCD-25P4/P4CT integrated display modules.

pixxiLCD-39P4/P4CT Datasheet

This document contains detailed information about the pixxiLCD-39P4/P4CT integrated display modules.

Workshop4 IDE User guide

This document provides an introduction to Workshop4, 4D Systems' integrated development environment.

NOTE: For more information about Workshop4 in general, please refer to the Workshop4 IDE User Guide, available at www.4dsystems.com.au

GLOSSARY

Hardware

1. 4D Programming Cable – The 4D Programming Cable is a USB to Serial-TTL UART converter cable. The cable provides a fast and simple way to connect all of the 4D devices that require TTL level serial interface to USB.
2. Embedded System – A programmed controlling and operating system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts.
3. Female Header – A connector attached to a wire, cable, or piece of hardware, having one or more recessed holes with electrical terminals inside.
4. FFC – Flexible flat cable, or FFC, refers to any variety of electrical cable that is both flat and flexible. It used to connect the display to a programming adaptor.
5. gen4 – IB – A simple interface that converts the 30-way FFC cable coming from your gen4 display module, into the common 5 signals used for programming and interfacing to 4D Systems products.
6. gen4-UPA – A universal programmer designed to work with multiple 4D Systems display modules.
7. Micro USB cable – A type of cable used to connect the display to a computer.
8. Processor – A processor is an integrated electronic circuit that performs the calculations that run a computing device. Its basic job is to receive input and provide the appropriate output.
9. Programming Adaptor – Used for programming gen4 display modules, interfacing to a breadboard for prototyping, interfacing to Arduino and Raspberry Pi interfaces.
10. Resistive Touch Panel – A touch-sensitive computer display composed of two flexible sheets coated with a resistive material and separated by an air gap or microdots.
11. microSD Card – A type of removable flash memory card used for storing information.
12. uUSB-PA5-II – A USB to Serial-TTL UART bridge converter. It provides the user with multi baud rate serial data up to 3M baud rate, and access to additional signals such as flow control in a convenient 10 pin 2.54mm (0.1”) pitch Dual-In-Line package.
13. Zero Insertion Force – The part where the Flexible Flat cable is inserted to.

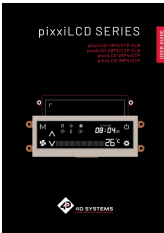
Software

1. Comm Port – A serial communication port or channel used to connect devices such as your display.
2. Device Driver – A particular form of software application that is designed to enable interaction with hardware devices. Without the required device driver, the corresponding hardware device fails to work.
3. Firmware – A specific class of computer software that provides the low-level control for the device's specific hardware.
4. GTX Tool – Genie Test Executor debugger. A tool used to check the data sent and received by the display.
5. GUI – A form of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation.
6. Image Files – Are graphics files generated upon program compilation that should be saved into the microSD Card.
7. Object Inspector – A section in Workshop4 where the user can change the properties of a certain widget. This is where the widgets customization and Events configuration happen.
8. Widget – Graphical objects in Workshop4.
9. WYSIWYG – What-You-See-Is-What-You-Get. The Graphics Editor Section in Workshop4 where the user can drag and drop widgets.

Visit our website at: www.4dsystems.com.au
 Technical Support: www.4dsystems.com.au/support
 Sales Support: sales@4dsystems.com.au

Copyright © 4D Systems, 2022, All Rights Reserved.
 All trademarks belong to their respective owners and are recognised and acknowledged.

Documents / Resources

	<p>4D SYSTEMS pixxiLCD-13P2-CTP-CLB Display Arduino Platform Evaluation Expansion Board [pdf] User Guide</p> <p>pixxiLCD-13P2-CTP-CLB, Display Arduino Platform Evaluation Expansion Board, Platform Evaluation Expansion Board, Evaluation Expansion Board, pixxiLCD-13P2-CTP-CLB, Expansion Board</p>
---	---

References

- [4D Systems – Turning Technology Into Art](#)
- [Support – 4D Systems](#)