

Workshop5 User Manual



WS 5

Manual

Revision 1.0

Copyright © 2024 4D Systems

Content may change at any time. Please refer to the resource centre for latest documentation.

Contents

1. Introduction	5
1.1. Requirements	7
1.2. Installation	8
1.3. Programming Tools Driver Installation	9
1.3.1. CP210x VCP Driver	9
1.3.2. FTDI VCP Driver	11
1.4. MicroSD Card Format	13
2. Setup Procedures	15
2.1. Open Recent Projects	15
2.2. Create a New Project	17
2.3. Environments	19
2.3.1. ViSi	19
2.3.2. Genie	20
2.3.3. Extended	21
2.3.4. RP2350 (Pico SDK)	22
3. Workshop5 Menu	23
3.1. Common Menus	23
3.1.1. Home Menu	23
3.1.1.1. Project Related Buttons	24
3.1.1.2. Code Related Buttons	24
3.1.1.3. Bookmarks Buttons	25
3.1.1.4. Find and Replace Buttons	25
3.1.1.5. Code Folding Buttons	26
3.1.1.6. Compile Buttons	26
3.1.2. Graphics Menu	27
3.1.2.1. Widgets Buttons	27
3.1.3. Comms Menu	29

3.1.4. ViSi Specific Menus	29
3.1.4.1. Tools Menu	29
3.1.4.2. Setup Buttons	30
3.1.4.3. Storage Buttons	30
3.1.4.4. Communication Buttons	30
3.1.4.5. System Buttons	30
3.1.4.6. Projects Menu	31
3.1.4.6.1. Destination Buttons	31
3.1.4.6.2. File System	32
3.1.4.6.3. Display Selection	34
3.1.4.7. Home Menu	34
4. Connect the Module	35
5. Hardware Setup	37
5.1. gen4-uLCD and pixxiLCD Displays	37
5.1.1. Using 4D-UPA	37
5.1.2. Using gen4-PA	37
5.1.3. Using gen4-IB	38
5.1.3.1. gen4-IB with uUSB-PA5-II	38
5.1.3.2. gen4-IB with Programming Cable	38
5.1.4. pixxiLCD Displays	39
5.1.4.1. Using 4D-UPA	39
5.1.4.2. Using gen4-IB with uUSB-PA5-II	39
5.1.4.3. Using gen4-IB with Programming Cable	39
5.2. microLCD and microOLED Displays	40
5.2.1. microLCD Displays	40
5.2.1.1. Using uUSB-PA-5	40
5.2.1.2. Using Programming Cable	40
5.2.2. microOLED	41
5.2.2.1. Using uUSB-PA5-II	41
5.2.2.2. Using Programming Cable	41
6. Workshop5 Widgets	42

7. Revision History	43
8. Legal Notice	44
8.1. Proprietary Information	44
8.2. Disclaimer of Warranties & Limitations of Liabilities	44

1. Introduction

Workshop5 IDE is the latest release from 4D Systems, designed as a successor to the popular Workshop4 IDE. This advanced integrated development environment (IDE) provides a fresh look and enhanced support for graphical programming with 4D display modules, while incorporating the familiar environments and functionality of Workshop4. Workshop5 is engineered to support all existing Workshop4 environments, with significant improvements and new features.

In Workshop4, users were able to work with four main environments – Designer, ViSi, ViSi-Genie, and Serial – as well as two Arduino environments, Basic Graphics and Extended Graphics. Workshop5 consolidates these environments for a streamlined experience:

- **Designer and ViSi (Workshop4)** – Workshop5 combines these into the new **ViSi** environment. Both Designer and ViSi in Workshop5 allowed programming display modules directly in 4DGL code, with ViSi offering a graphical editor or WYSIWYG tool for user interface design.
- **ViSi-Genie** – *often called "Genie"*, Workshop5 aims to support this as the **Genie** environment, retaining the codeless GUI project deployment while adding flexibility with the optional Genie Magic feature, which allows users to incorporate custom 4DGL code.
- **Serial, Arduino Basic/Extended Graphics** – Planned to be a part of Workshop5's **Extended** environment, this mode transforms 4D displays into Serial UART-controlled modules. The Extended environment also aims to support features from Workshop4's Arduino Basic and Extended environments.

With the initial release of Workshop5, users gain access to the **ViSi** environment, which offers a similar experience to the Workshop4's Designer and ViSi. This environment supports all Picaso, Pixxi28, Pixxi44 and Diablo16 display modules.

Additionally, Workshop5's initial release will also support RP2350 display modules with a dedicated environment based on Raspberry Pi's Pico SDK. The environment will provide a complete solution for most users as well as provide an export functionality for advanced users allowing them to continue development in their preferred development environment, such as Visual Studio Code.

Note

Workshop5 will **NOT** support Goldelox display modules and other modules that have reached end-of-life or are not recommended for new designs.

Notably, while Workshop5 provides a robust platform for new projects, it is recommended that users continue to use Workshop4 for any active projects. Workshop4 projects are not compatible with Workshop5.



A visual programming experience suitably called ViSi enables drag-and-drop type placement of widgets to assist with 4DGL code generation.

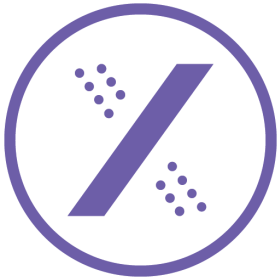
This allows the user to visualize how the display will look while being developed and provides direct control over all display modules powered by 4D Labs processors.



Genie environment doesn't require any 4DGL coding at all, it is all done for you.

Simply lay the display out with the widgets you want, set the events to drive them and the code is written for you automatically. Genie provides the latest rapid development experience from 4D Systems.

An advanced feature, Genie Magic, allowing users to add custom code in addition to the generated Genie code is also available, further expanding the possibilities.



Utilizing the Serial (SPE) libraries for 4D Labs processor used in the display, this environment offers an alternative development experience to 4DGL programming.

Serial (SPE) provides various commands bridging a host controller, such as an Arduino, and the 4D Systems displays and providing direct control over the graphical user interface.



Providing support for 4D Systems' RP2350 display modules, this environment offers an environment similar to **ViSi**. This is perfect for those who want to perfect their graphical designs before moving into the core functionality of their projects.

It allows users to develop applications completely within the IDE or continue to their preferred IDEs which may offer more functionalities to advanced users.

1.1. Requirements

Workshop5 is currently available for 64-bit Windows PCs. Here are the requirements for installing the application:

- Windows 7 and later are supported, older operating systems are not supported (and do not work). Please note that the ARM version of Windows is not supported.
- An Intel Pentium 4 processor or later that's SSE3 capable.
- At least 300 MB of storage space
- Minimum 1GB of RAM (*depends on usage*)

Listed are tools recommended during development:

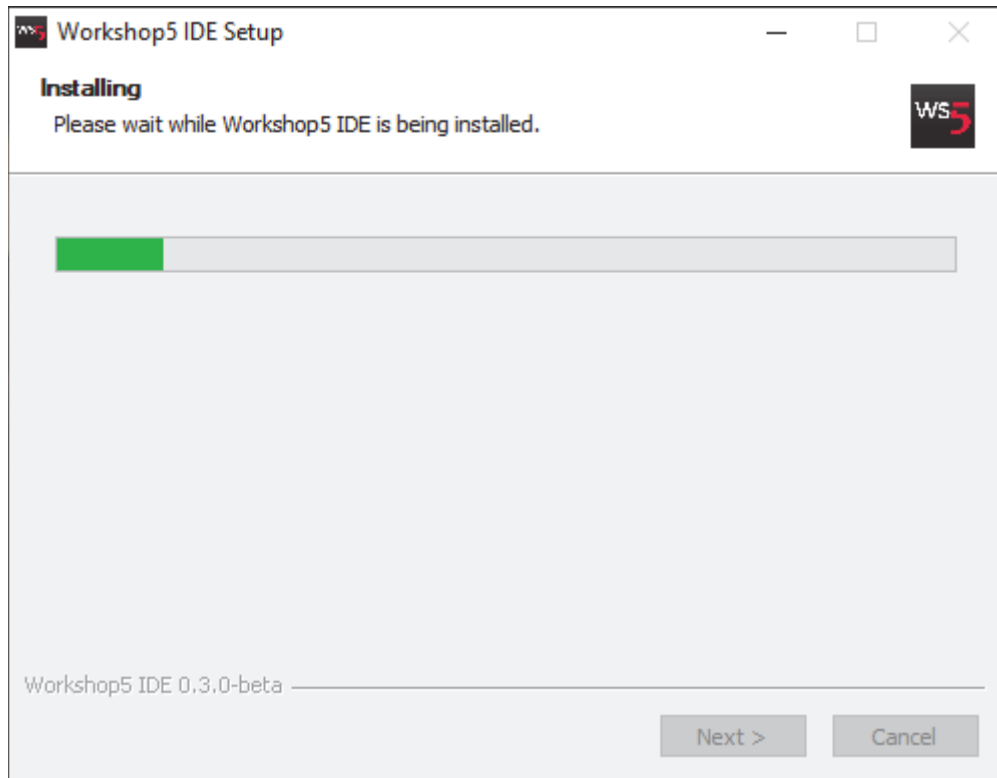
- **4D-UPA**
- **gen4-PA**
- **uUSB-PA5** (w/ **gen4-IB** if using gen4 and pixxiLCD displays)
- **4D Programming Cable** (w/ **gen4-IB** if using gen4 and pixxiLCD displays)

All are available from **4D Systems**. These hardware tools are used to update the PmmC and upload Workshop5 projects to the graphics processors.

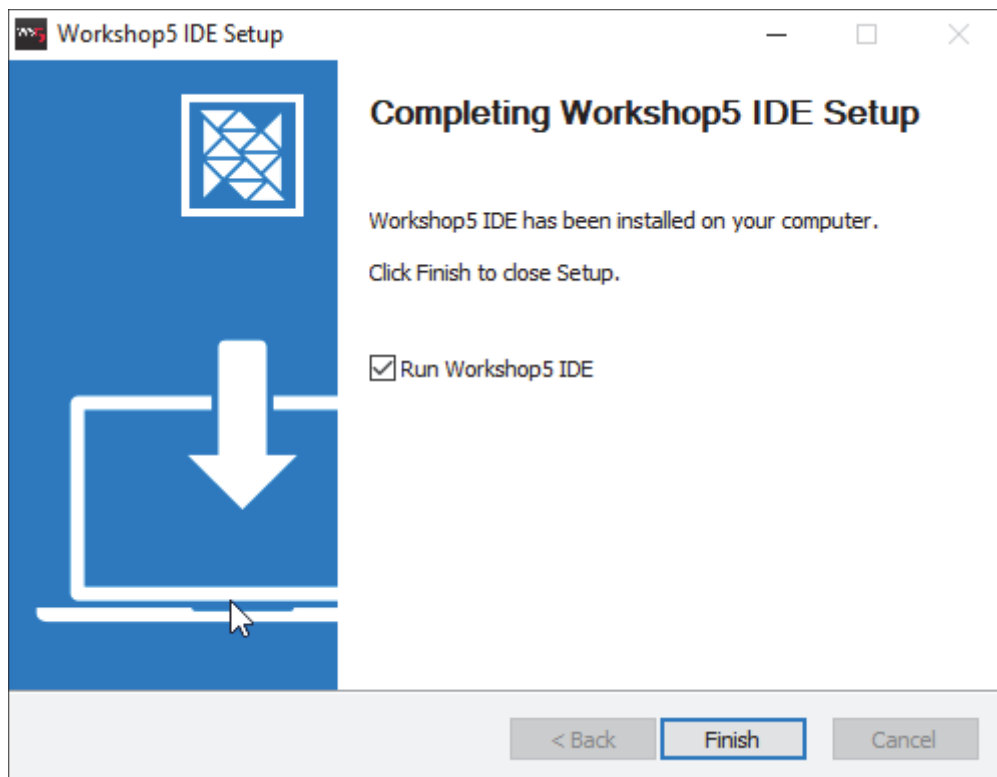
1.2. Installation

After acquiring a copy of the **installer** you should be able to run it simply by double clicking the file.

Simply wait for the installation to finish.



After the installation, you are given an option to run Workshop5 IDE after closing the installation window. If you want to do so, leave a check mark on the checkbox. Click on **Finish**.



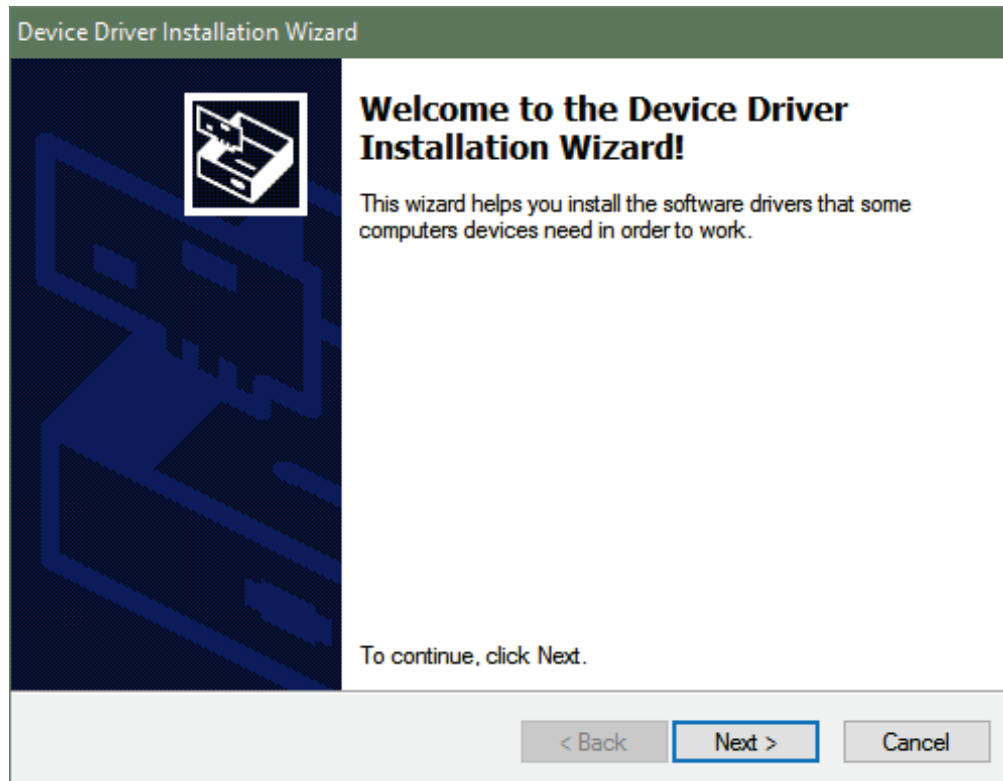
1.3. Programming Tools Driver Installation

You can find the links of the drivers for each recommended USB to TTL programming solutions on their product pages.

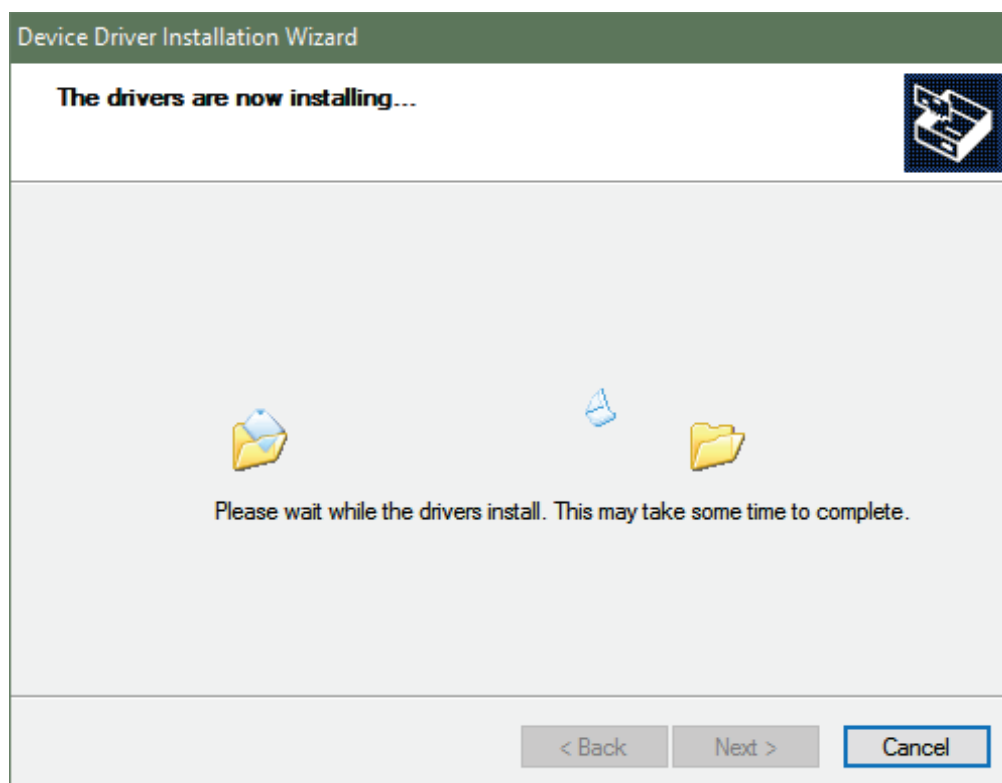
1.3.1. CP210x VCP Driver

This is used for **4D Programming Cable**, **uUSB-PA5-II**, **gen4-PA** and **4D-UPA**. This driver can be downloaded [here](#).

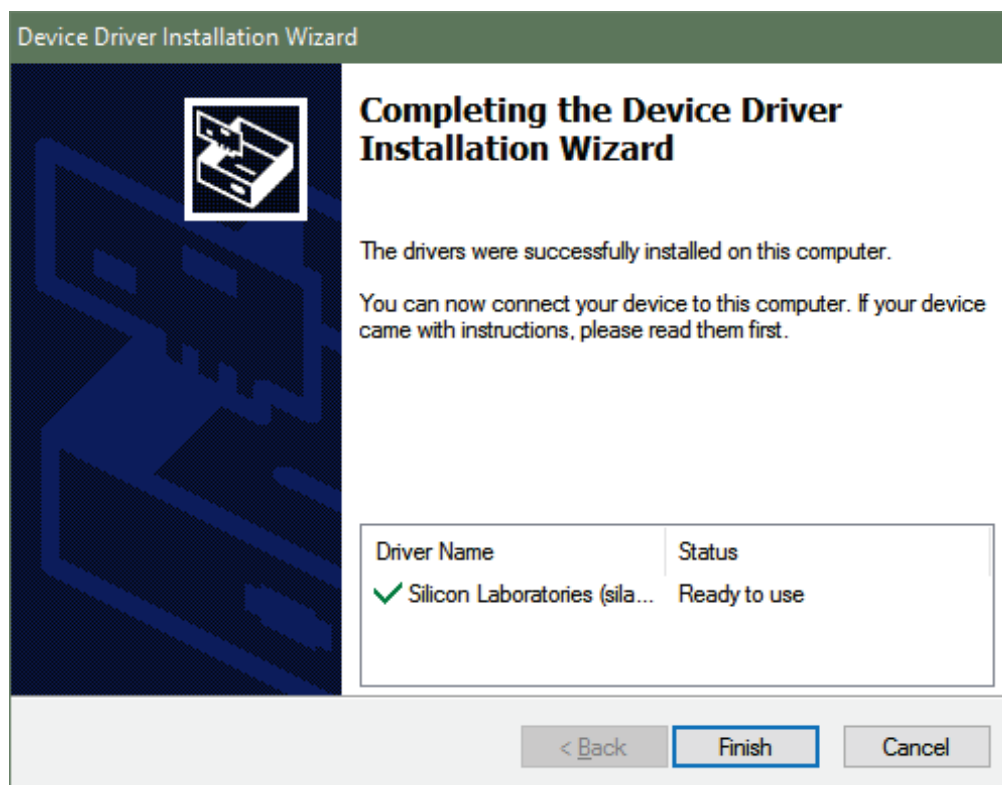
After downloading successfully, save the zipped file into a folder on your system and unzip the file. Launch the appropriate installer based on your computer's architecture.



Continue with the installation prompts by clicking on **Next**.



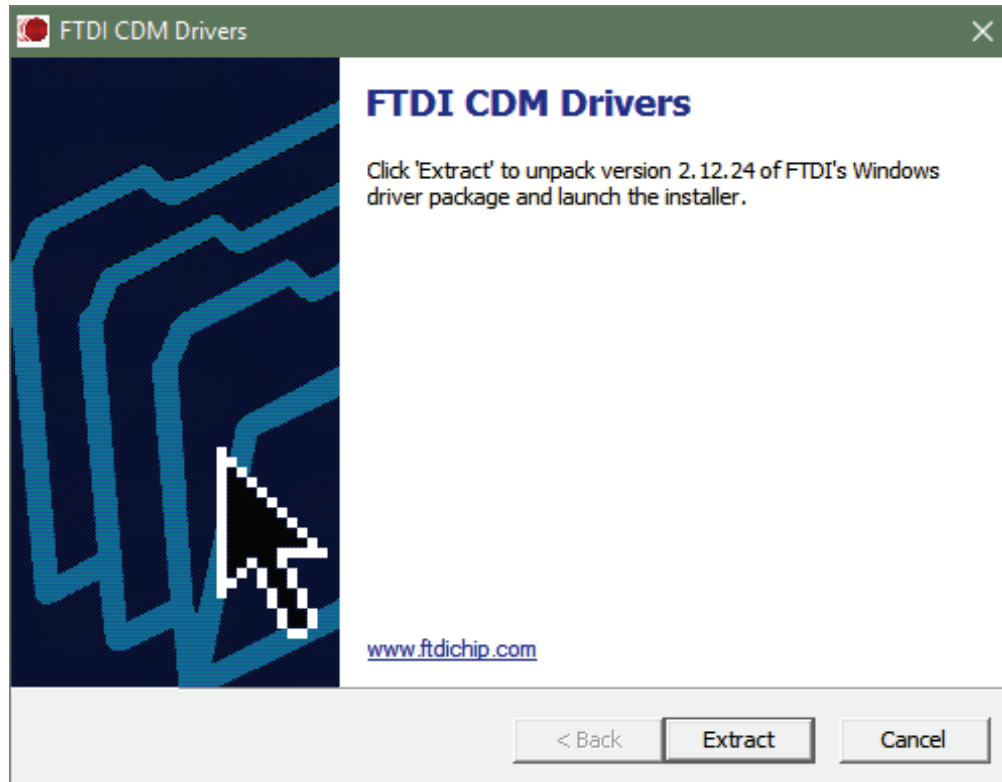
The installation should finish shortly.



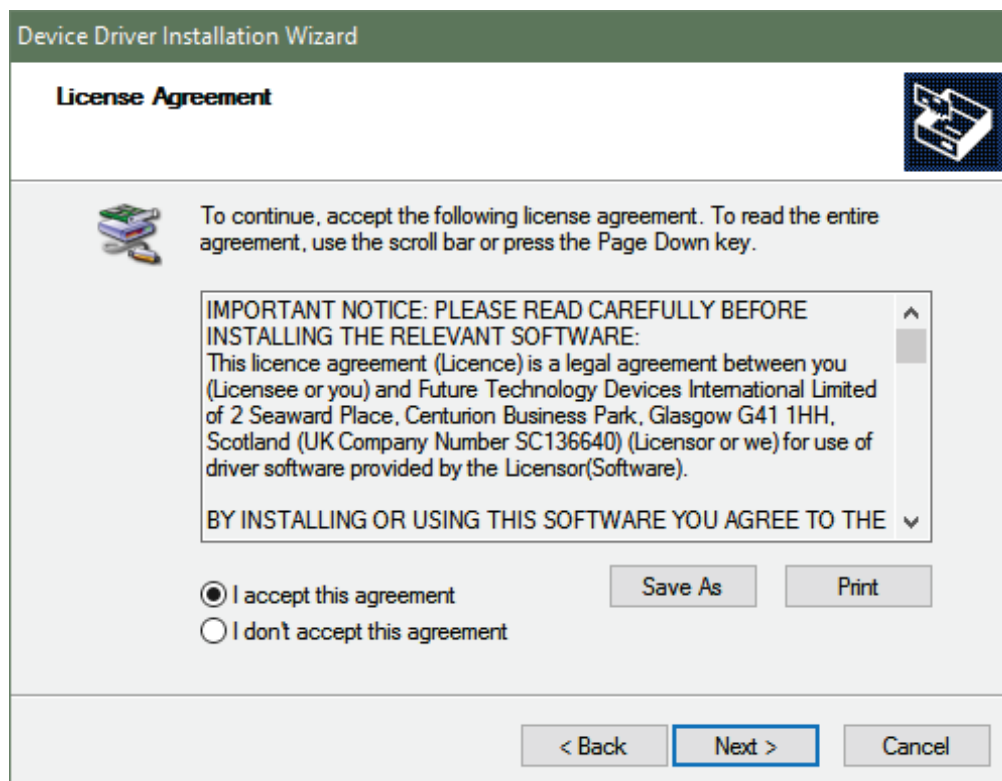
1.3.2. FTDI VCP Driver

The installer for this driver can be found [here](#). This driver is used solely for uUSB-PA5 which has been superseded by the uUSB-PA5-II.

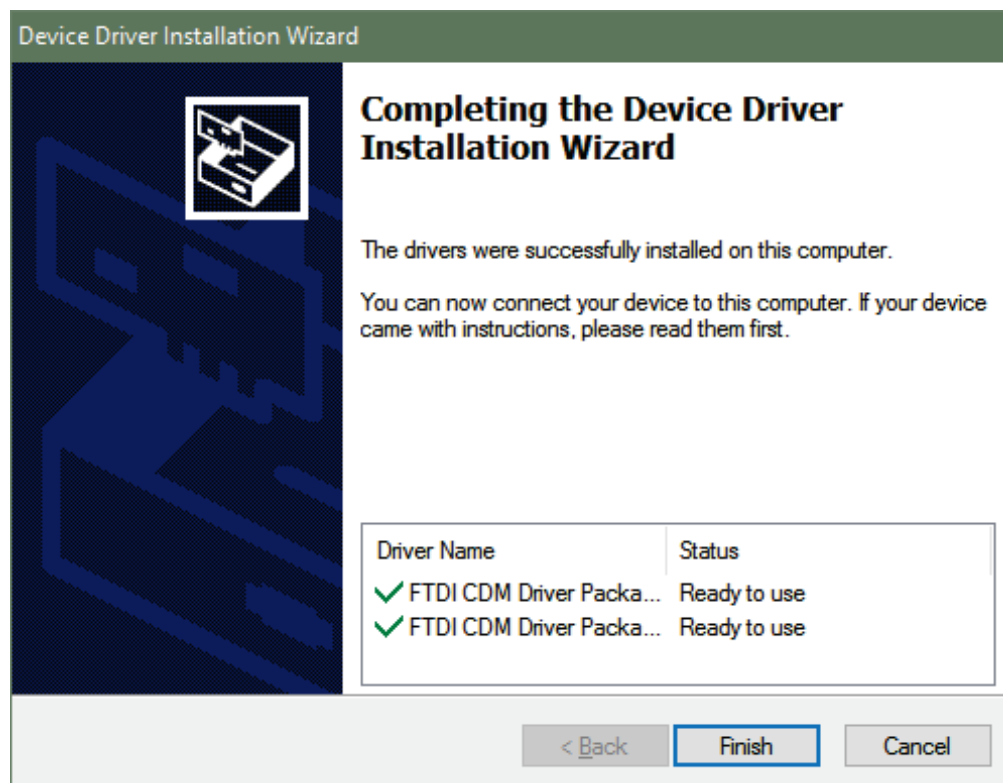
After downloading, save the zipped file into a folder on your system and unzip the file. Launch the installer afterwards.



Accept the agreement and continue with the installation prompts by clicking on Next.

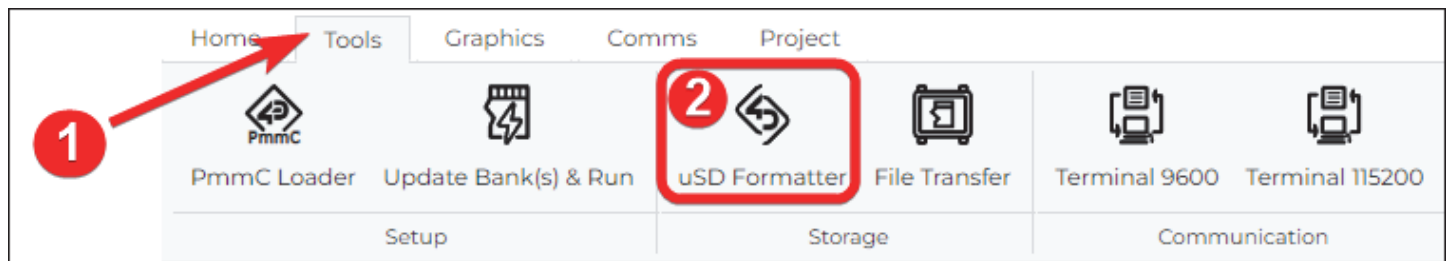


The installation should finish quickly afterwards.

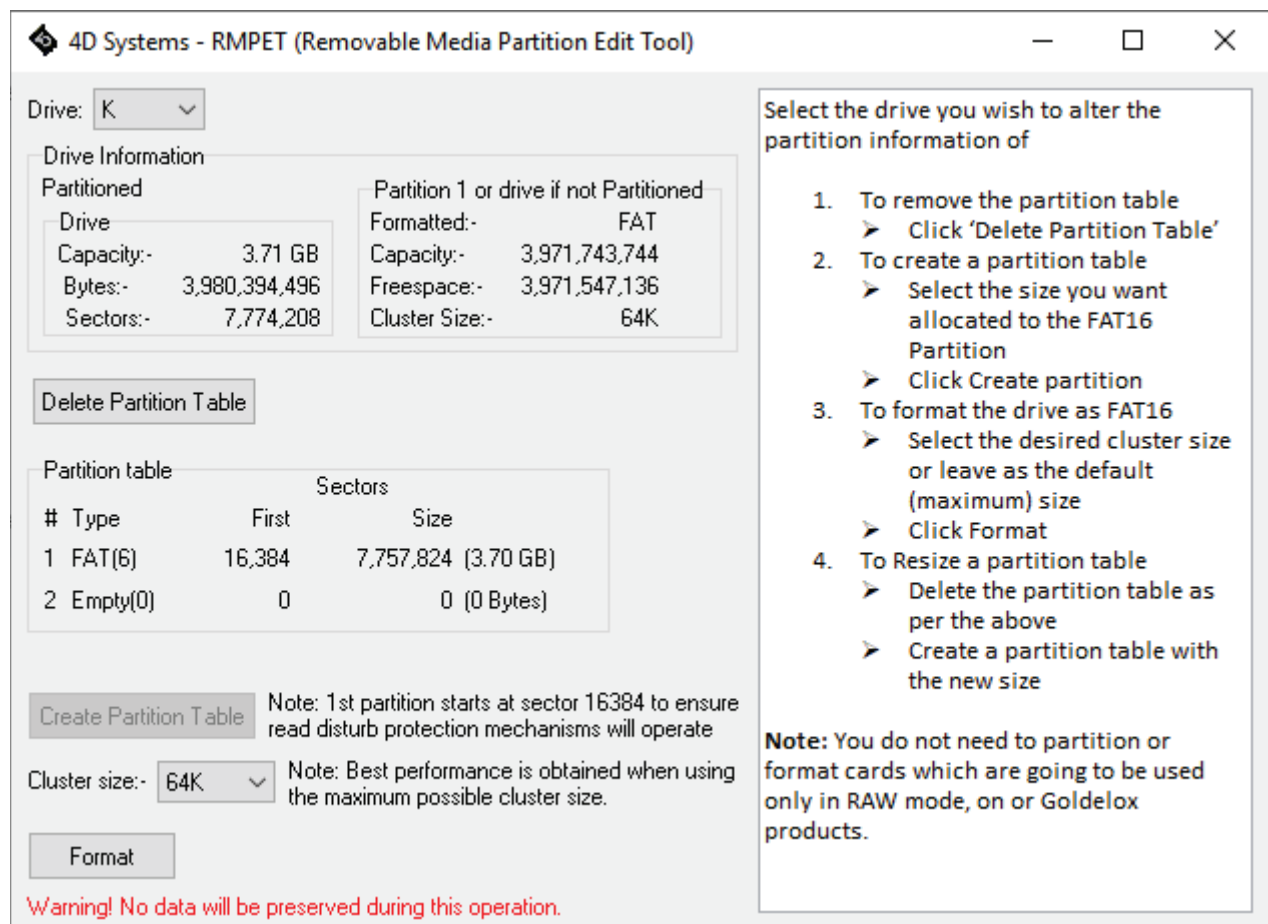


1.4. MicroSD Card Format

The microSD card shall be FAT16-formatted for processors. It is recommended to use Workshop5's built-in utility, uSD Formatter when formatting and partitioning a microSD card. Particularly when using uSD cards with capacity greater than 4GB.



The software provides detailed information on the uSD card's current status an easy way to partition the uSD card while considering the read disturb protection of uSD cards.



Note

Cards **MUST** be formatted FAT16 to work with 4D Systems models. FAT32, exFAT, NTFS etc will not work. For best results, format and partition your microSD cards using uSD Formatter.

For a more detailed instruction on how to use this utility, please refer to 4D Systems' application note titled:

[General Partitioning a microSD into FAT and RAW Components](#)

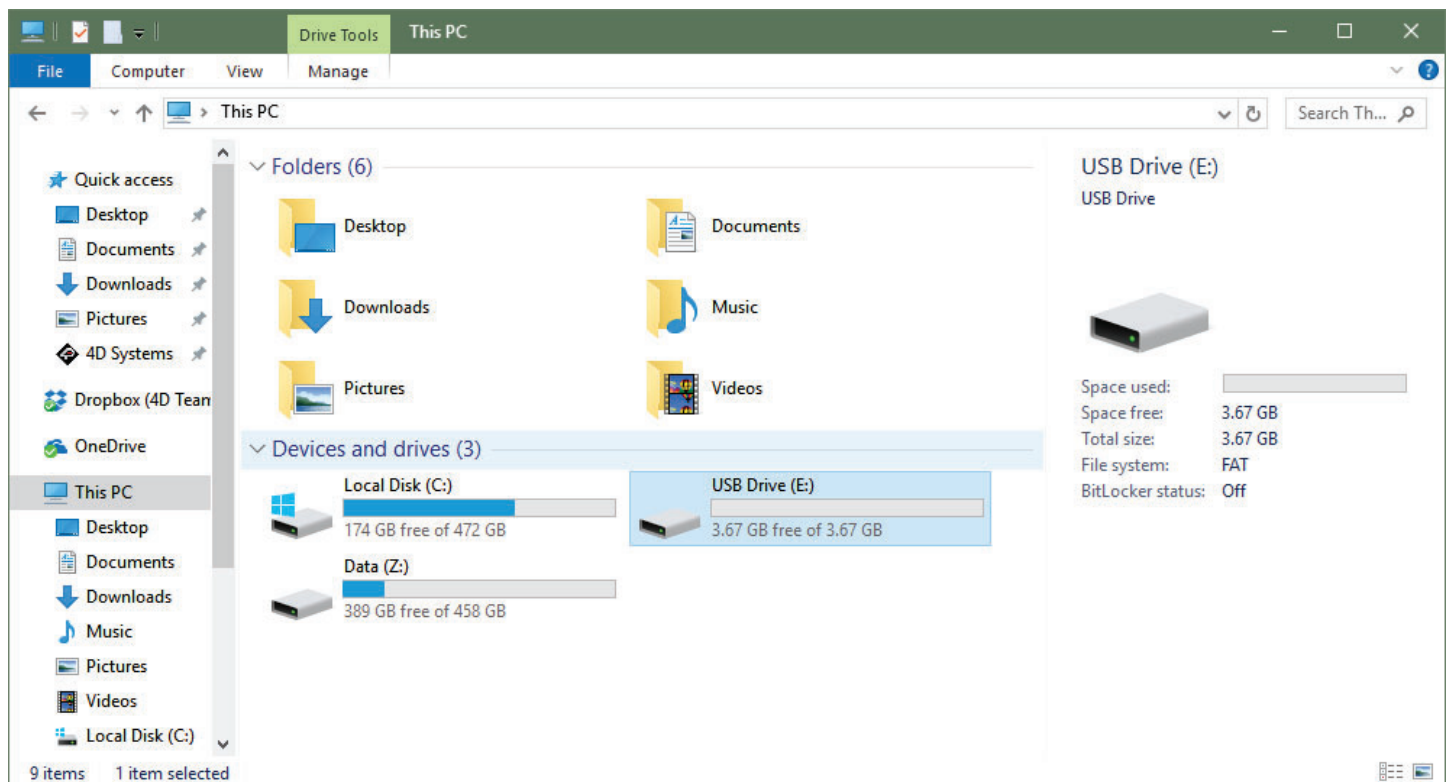
When using RMPet to format and/or Partition your microSD card, it is best to use a microSD to USB adaptor, or a microSD to SD adaptor into a media slot of your Laptop/PC. The cards need to be formatted on your PC, not on the display module themselves. Many types/brands are available, choose one that best suits your hardware setup.



Note

microSD cards **MUST** be SPI compatible, and it is highly recommended to use Industrial Grade cards to prevent corruption over time due to a phenomenon called Read Disturb, which affects NAND Flash memory. 4D Systems offers such cards, available on our website.

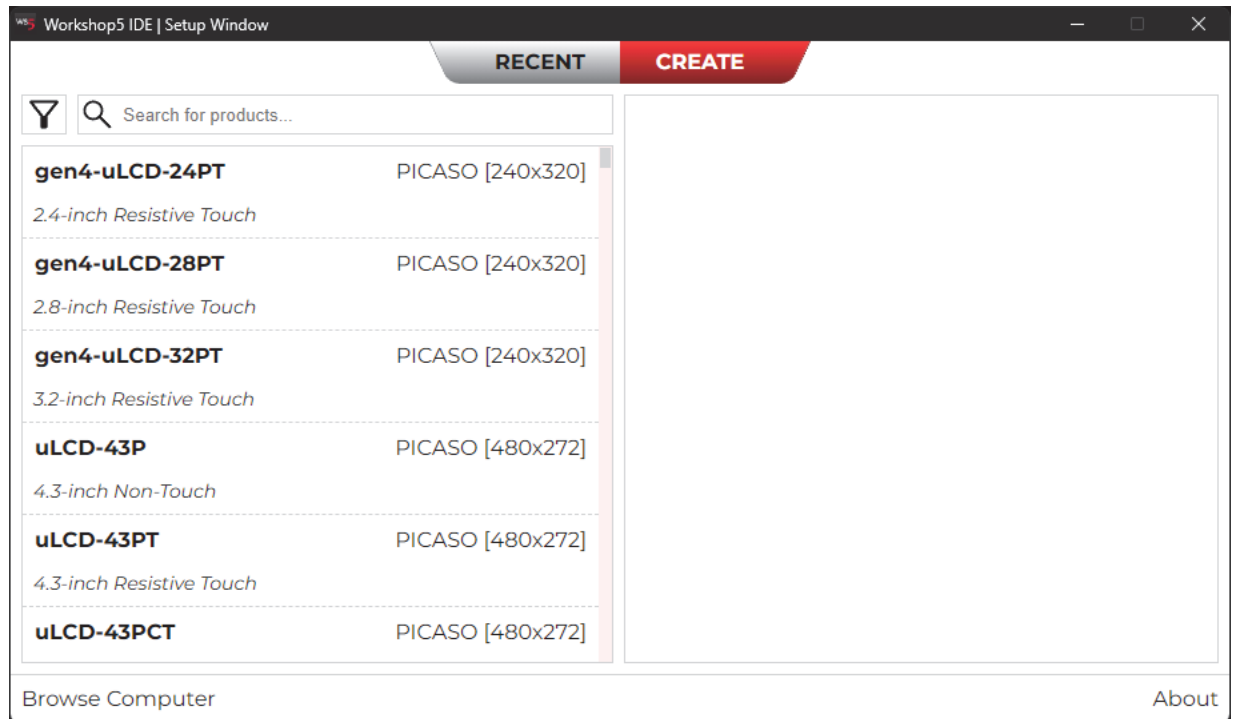
Afterwards, check if the uSD card mounted successfully. Here, it is shown as drive **E:**.



2. Setup Procedures

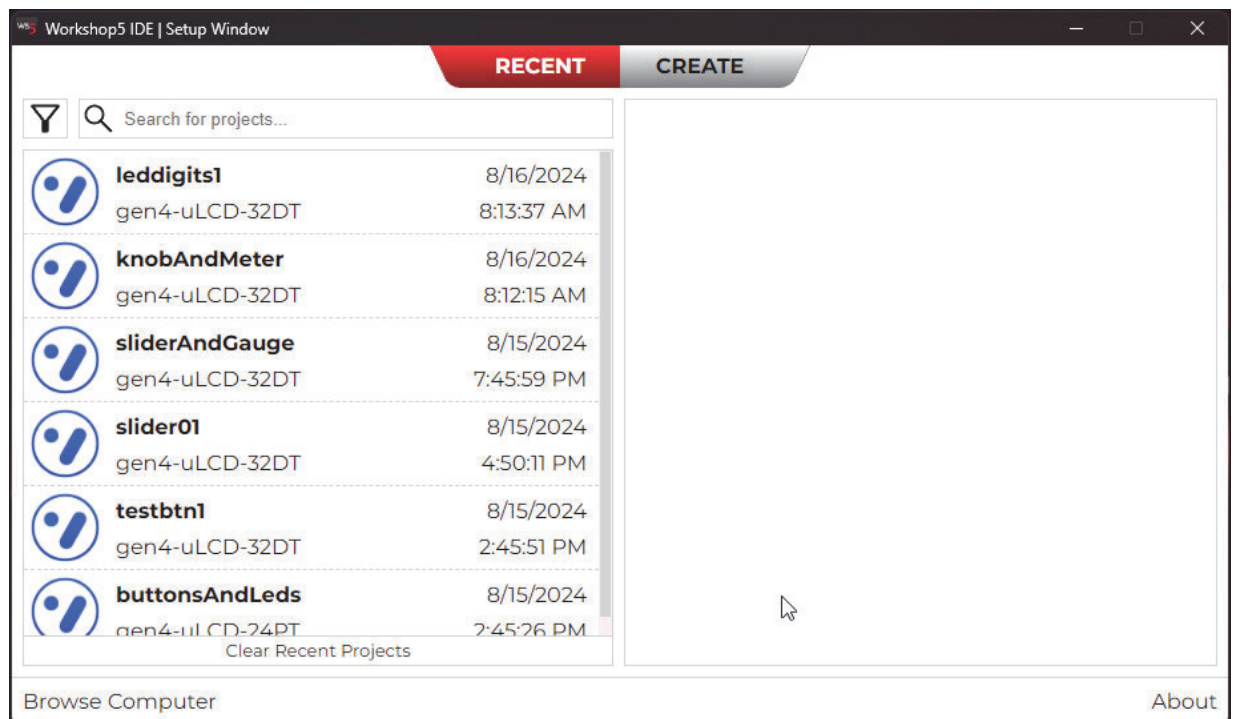
Launch Workshop5 IDE just like any typical Windows application.

At launch, Workshop5 IDE will display the **Setup Window** which defaults to **Recent** tab when there are recent projects available or to **Create** tab when there are no recent projects.



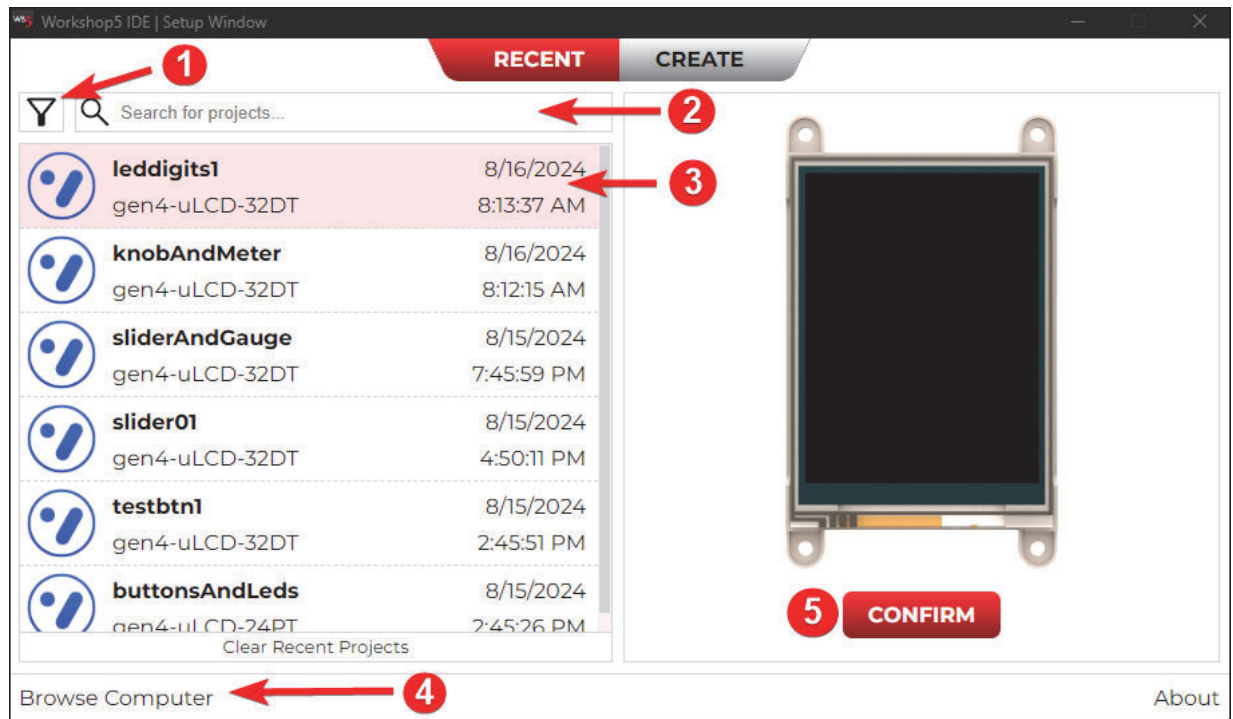
2.1. Open Recent Projects

If this is the first time the Workshop5 IDE has been launched, it will display the **Create** tab, as shown above; otherwise, the list of recent projects will be shown instead.

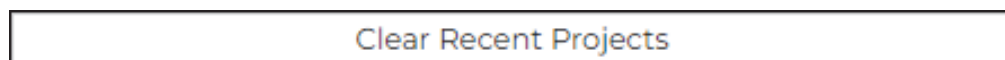


To open a recent project can be open using the procedure below.

1. Filter the recent project by environment.
2. Search recent project using the textbox.
3. Select the recent project from the list.
4. Use browse project button open file explorer and navigate to the recent project.
5. Click **CONFIRM** button to open the project.



To clear the recent project list, click the **Clear Recent Project** button.



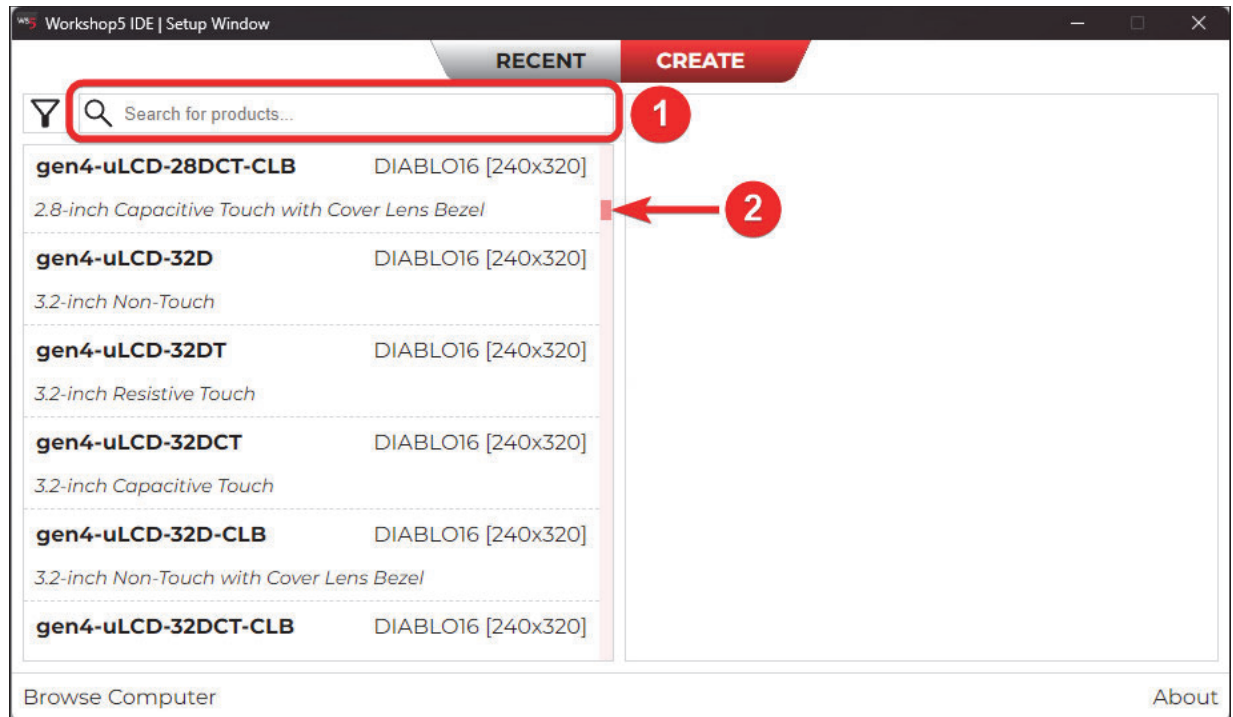
2.2. Create a New Project

To start a new project, navigate to the **Create** tab.



There are two (2) options that you can use to select your target display.

1. Use the search for products text box.
2. Use slider or mouse wheel.

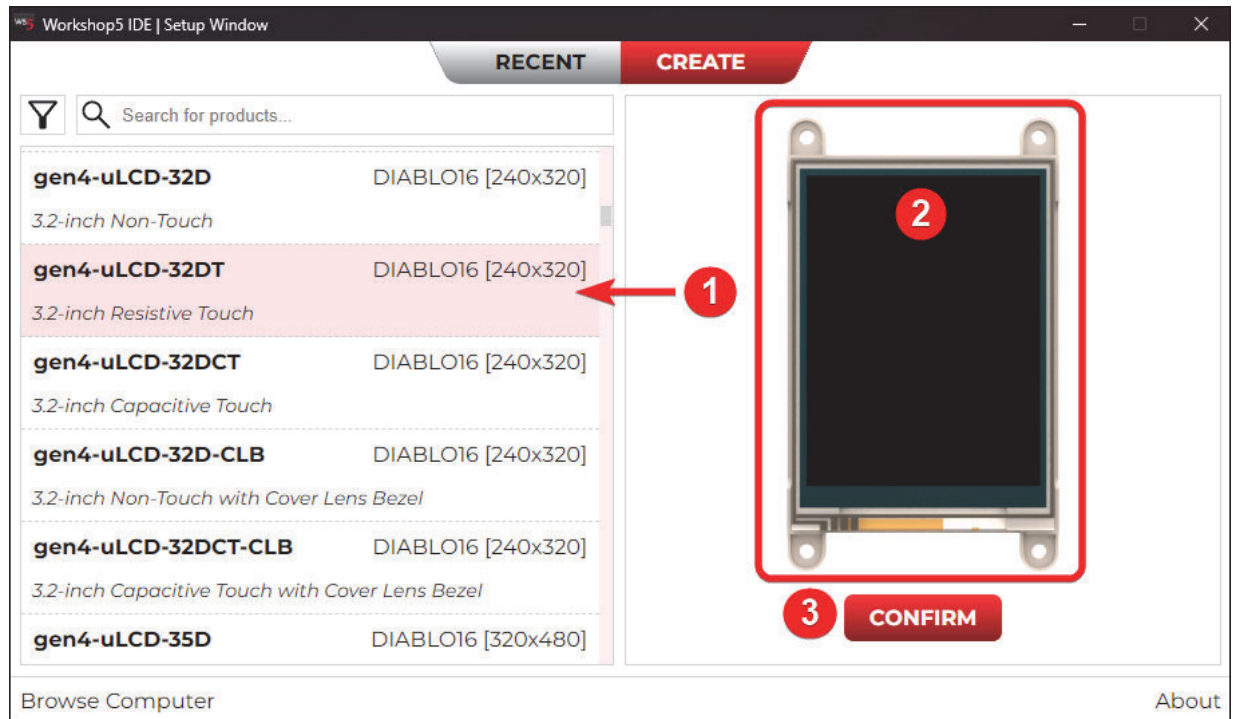


You can also click the filter button and select the processor of your target display.



After finding your target display, you can follow the steps below to proceed.

1. Select your target display.
2. Click the display image to select your desired display orientation.
3. Click the Confirm.

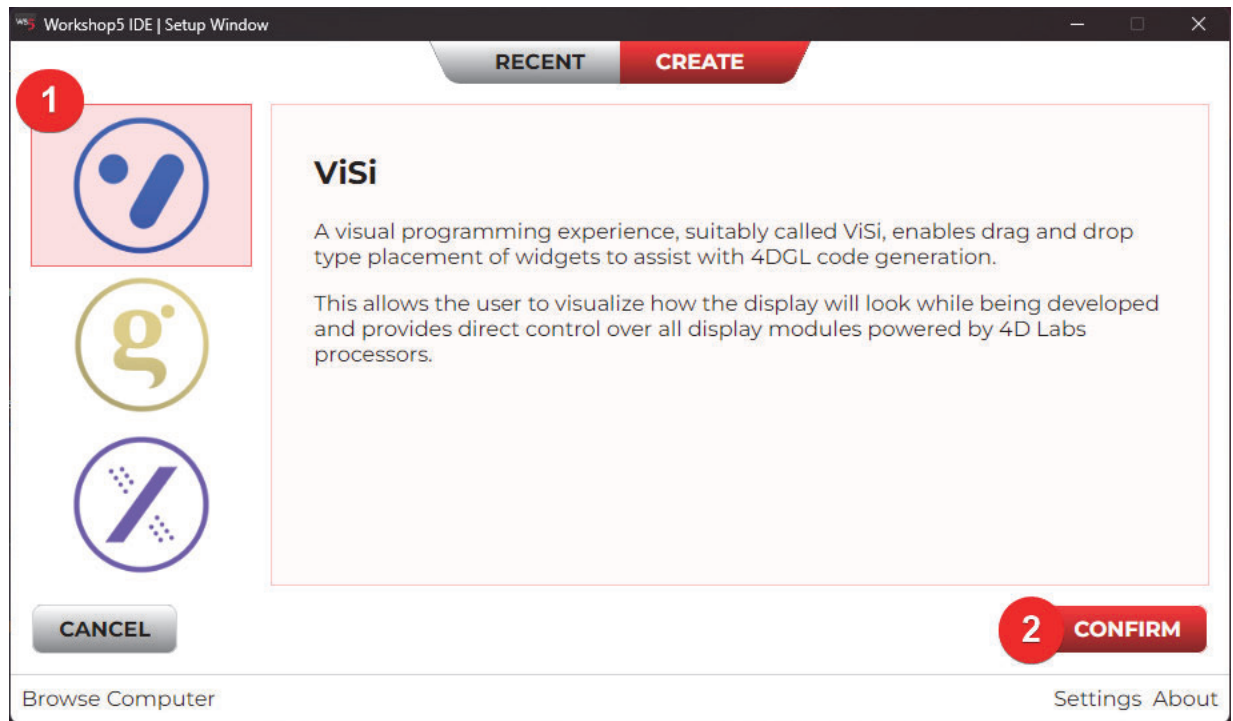


After selecting the display, the environment needs to be selected. Depending on your display you may be prompted to select a **Workshop5 Environment** (when the display primarily uses a 4D processor) or the compatible environment will open automatically (when using RP2350 display).

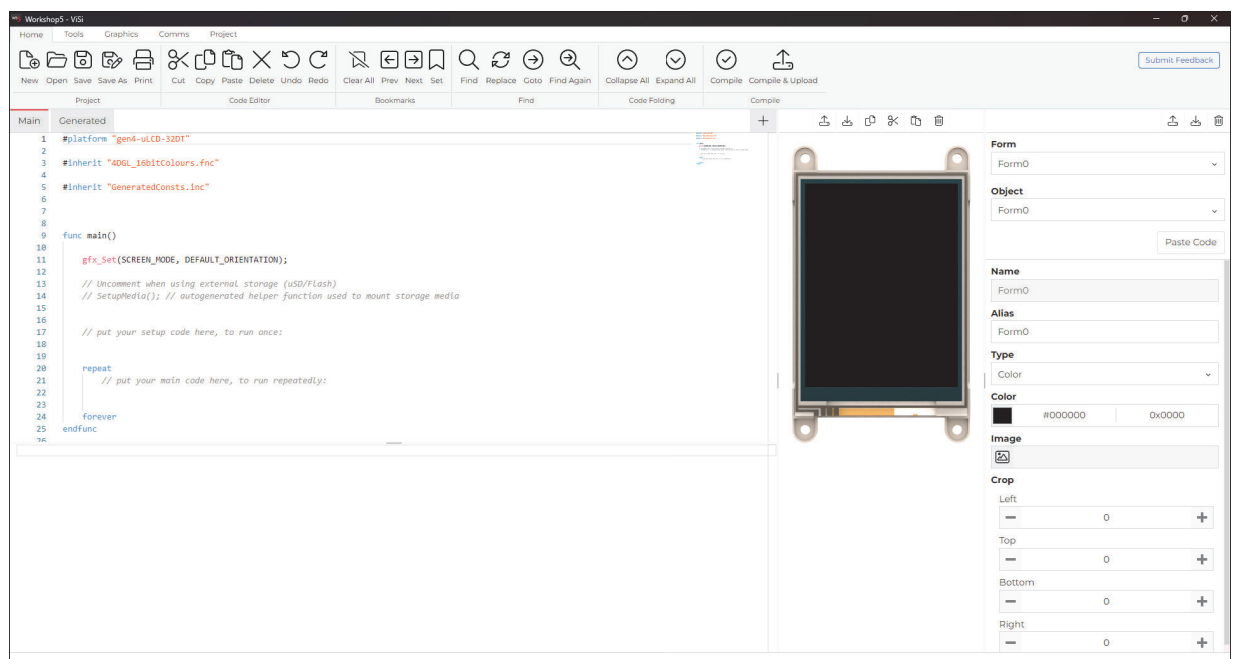
2.3. Environments

2.3.1. ViSi

ViSi environment enables drag-and-drop type placement of objects to assist with with 4DGL code generation and programming.



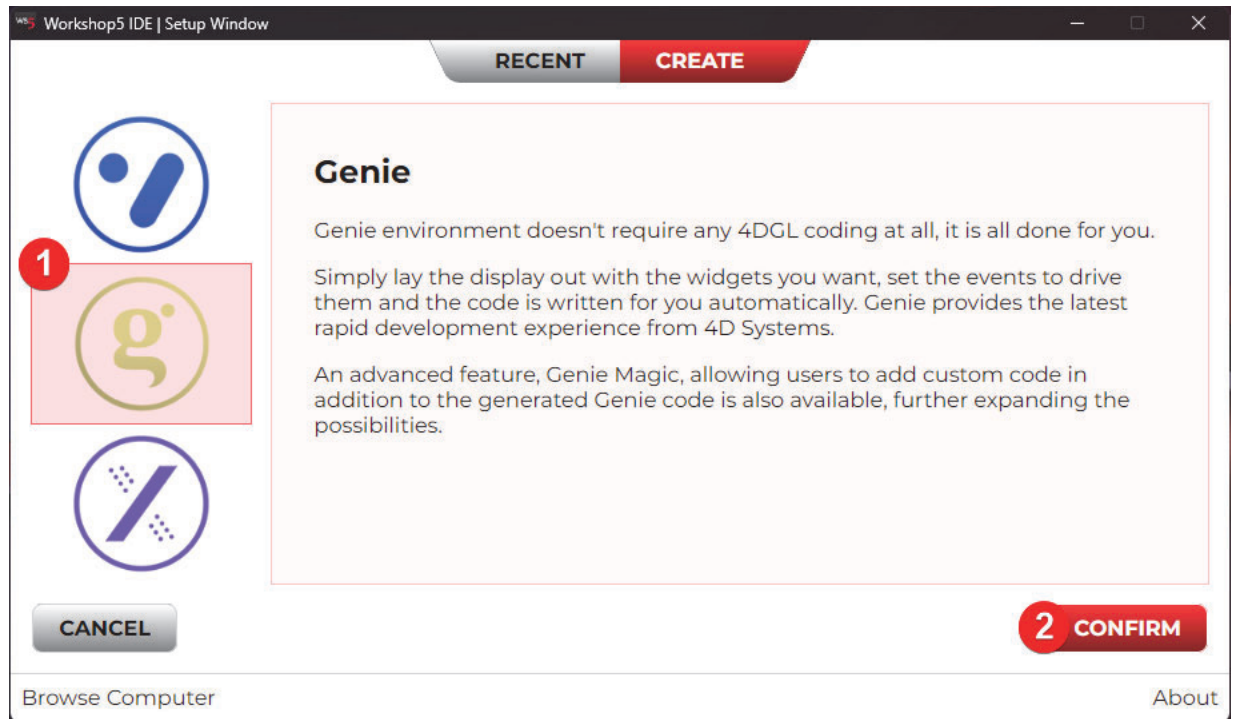
ViSi is an excellent software tool that lets users instantly view the results of their graphical layouts. It includes a variety of widgets that can be easily placed onto the simulated module display. Each object's properties can be edited, and with a single click, all relevant 4DGL code for that object is generated in the user program. Users can then write additional 4DGL code around these objects to utilize them as desired.



To learn more, please refer to the [Workshop5 ViSi User Manual](#)

2.3.2. Genie

The **Genie** environment offers a code-free environment for deploying GUI projects, primarily for use with other host microcontrollers via the Serial UART interface. Additionally, it provides an option for **Genie Magic** feature, which allows the integration of custom 4DGL code into various parts of the Genie program.



Genie provides the user with a feature-rich rapid development environment, second to none.

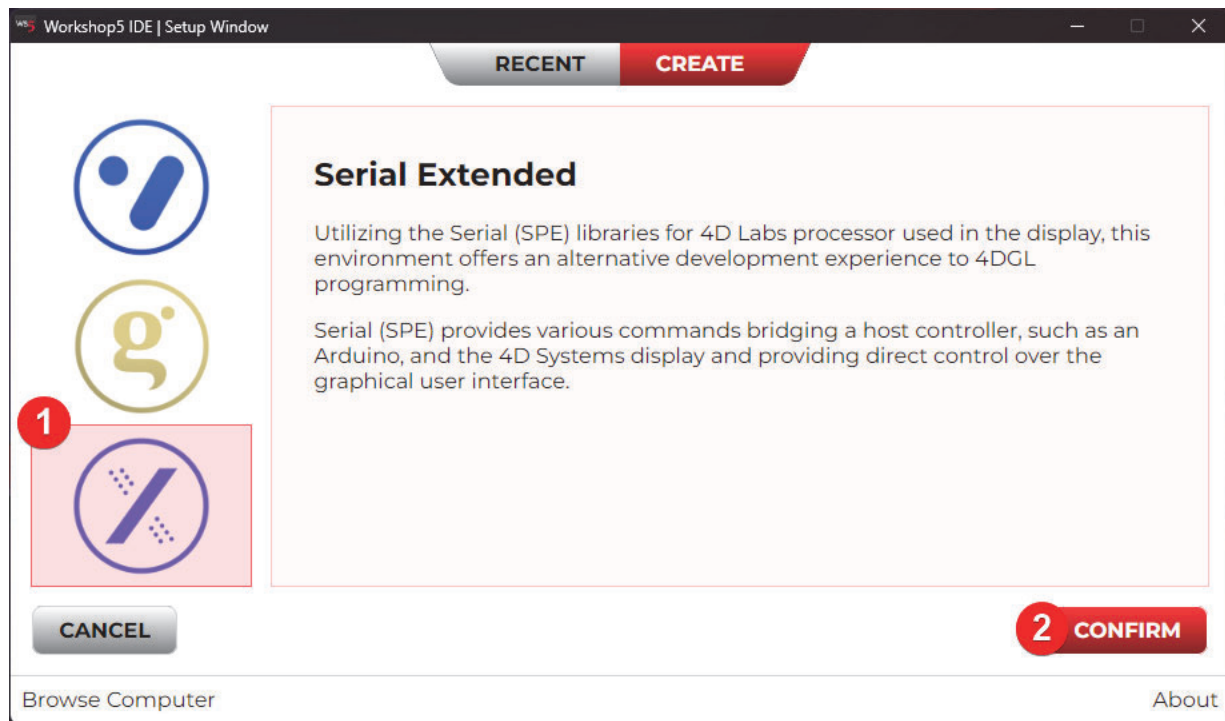
The user can select and position the necessary objects on the display, without needing to write any code. Each object comes with adjustable parameters and configurable events to animate and control other objects or interact with external devices or host controller.

Note

This environment will become available in a future version of Workshop5 IDE. For a similar experience, please check out Workshop4 ViSi Genie instead. This will provide a similar experience however, please keep in mind that projects are incompatible.

2.3.3. Extended

Extended environment turns the 4D display modules to a Serial UART controlled display and supports Arduino compatible environment.



Once the display module is configured by the Extended Environment (by downloading what is called the SPE application to the module), commands can be sent from the user's host microcontroller to display text and widgets.

The Extended Environment should not be taken as being basic in terms of its capabilities, as it has the full 4DGL command set behind it, but available from the Host rather than from programming the display module itself using the Workshop5 IDE.

Please refer to the following reference manuals for a complete listing of all the supported Serial commands for each processor

- [DIABLO-16 Serial Command Set Reference Manual](#)
- [PIXXI Serial Command Set Reference Manual](#)
- [PICASO Serial Command Set Reference Manual](#)
- [GOLDELOX Serial Command Set Reference Manual](#)

Note

This environment will become available in a future version of Workshop5 IDE. For a similar experience, please check out Workshop4 Arduino Extended Graphics instead. This will provide a similar experience however, please keep in mind that projects are incompatible.

2.3.4. RP2350 (Pico SDK)

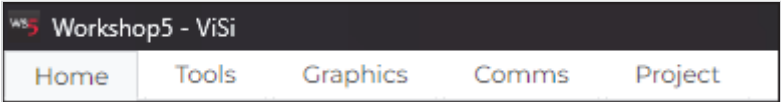
Workshop5 provides a code and graphics editor for RP2350 based modules. All application code can also be developed within the Workshop5 IDE, easily coupling it with the design, so is a one-stop shop for development with these modules.

The Workshop5 IDE utilises Raspberry Pi's Pico SDK to handle the compiling and linking of RP2350 projects, and using the **Graphics4D** library.

3. Workshop5 Menu

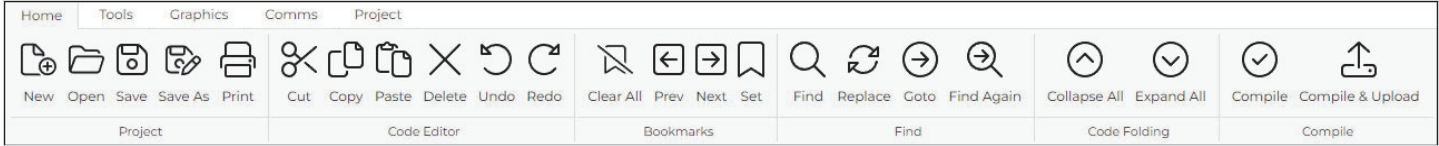
3.1. Common Menus

Workshop5 IDE has five (5) menus for all environments.



3.1.1. Home Menu

The Home menu is the main menu.

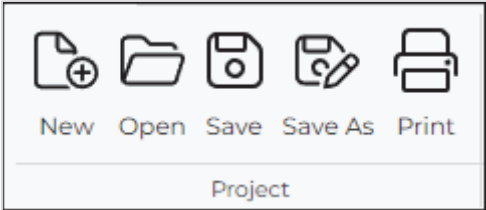







This menu contains the following grouped objects.

- Project related buttons
- Code releted buttons
- Bookmarks buttons
- Find and Replace buttons
- Compile buttons

3.1.1.1. Project Related Buttons

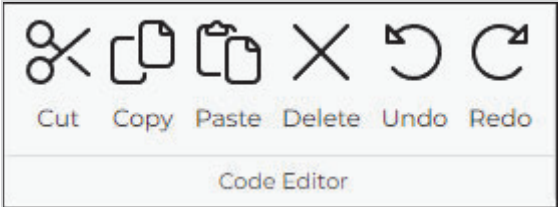
The Project related buttons includes command such as New, Open, Save, Save As and Print.



Button	Description
 New	Create a New Project by selecting the target display module and development environment.
 Open	Open a standard Open file window to browse and load an existing project.
 Save	Save all modified projects/files.
 Save As	Save a copy of a previously saved project/file and give it a new name.
 Print	Open a Print window that can be used to print the project.

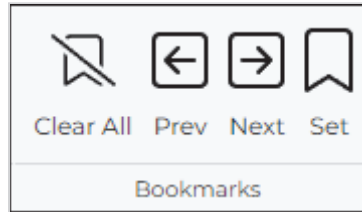
3.1.1.2. Code Related Buttons

The Code related buttons includes standard Windows commans such as Cut, Copy, Paste, Delete, Undo and Redo.

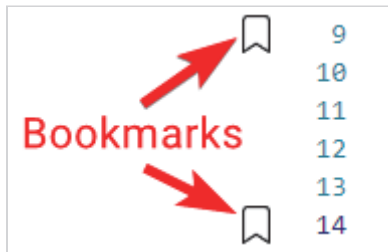


3.1.1.3. Bookmarks Buttons

The Bookmarks buttons includes Clear All, Prev, Next and Set.



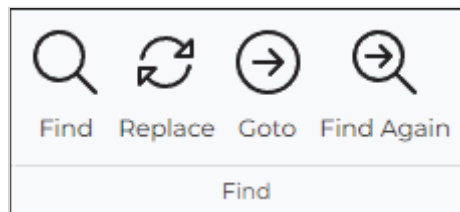
Bookmarks are shown close to the line number:



Bookmarks are especially useful for large projects.

3.1.1.4. Find and Replace Buttons

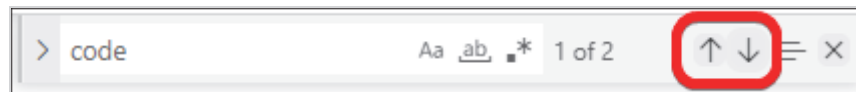
The Find and Replace buttons includes Find, Replace, Goto and Find Again.



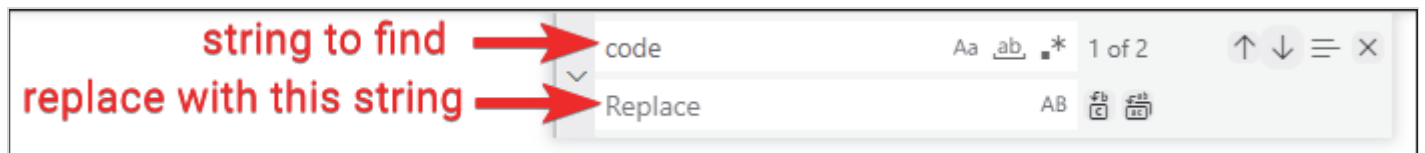
The **Find** button prompts for a string and highlights it in the code:



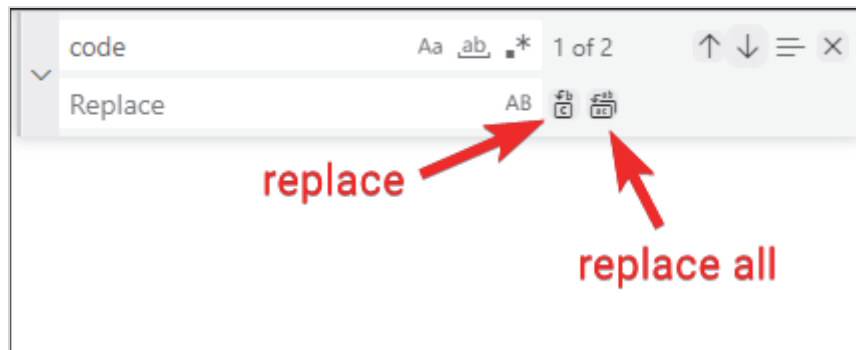
Use the up and down arrows to look for the previous and next occurrence.



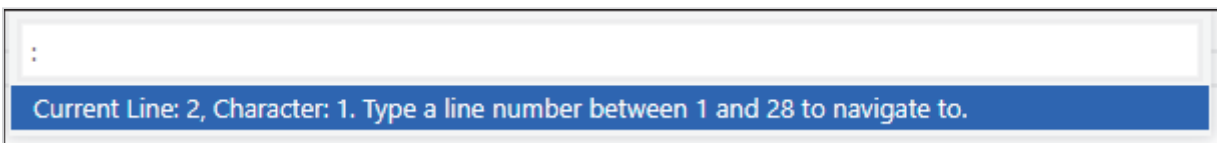
The **Replace** button searches for a string and changes it with another string.



And use the replace (Enter key) and replace all button as an option.

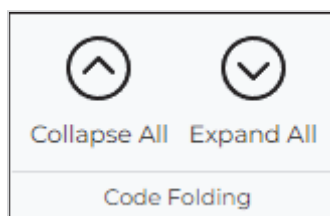


The **Goto** button prompts for a line number.



3.1.1.5. Code Folding Buttons

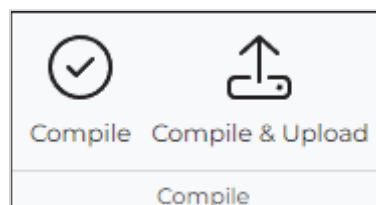
The Code folding buttons includes Collapse All and Expand All.



This is especially useful for large projects.

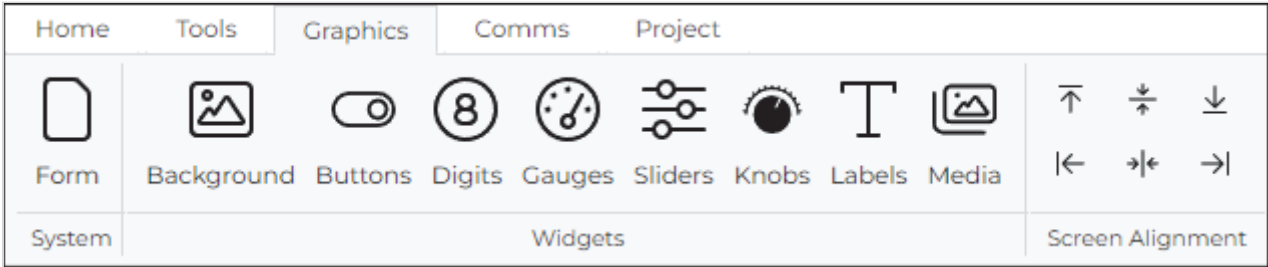
3.1.1.6. Compile Buttons

The **Compile** button launches the compilation of the project, can also be accessed with shortcut key Ctrl + F9. The **Compile & Upload** compiles and uploads the project to the screen, which has a shortcut key of F9.



3.1.2. Graphics Menu

The **Graphics** menu includes all the tools needed for designing your project.



It contains the following grouped buttons:

- System buttons
- Widgets buttons
- Screen alignment buttons

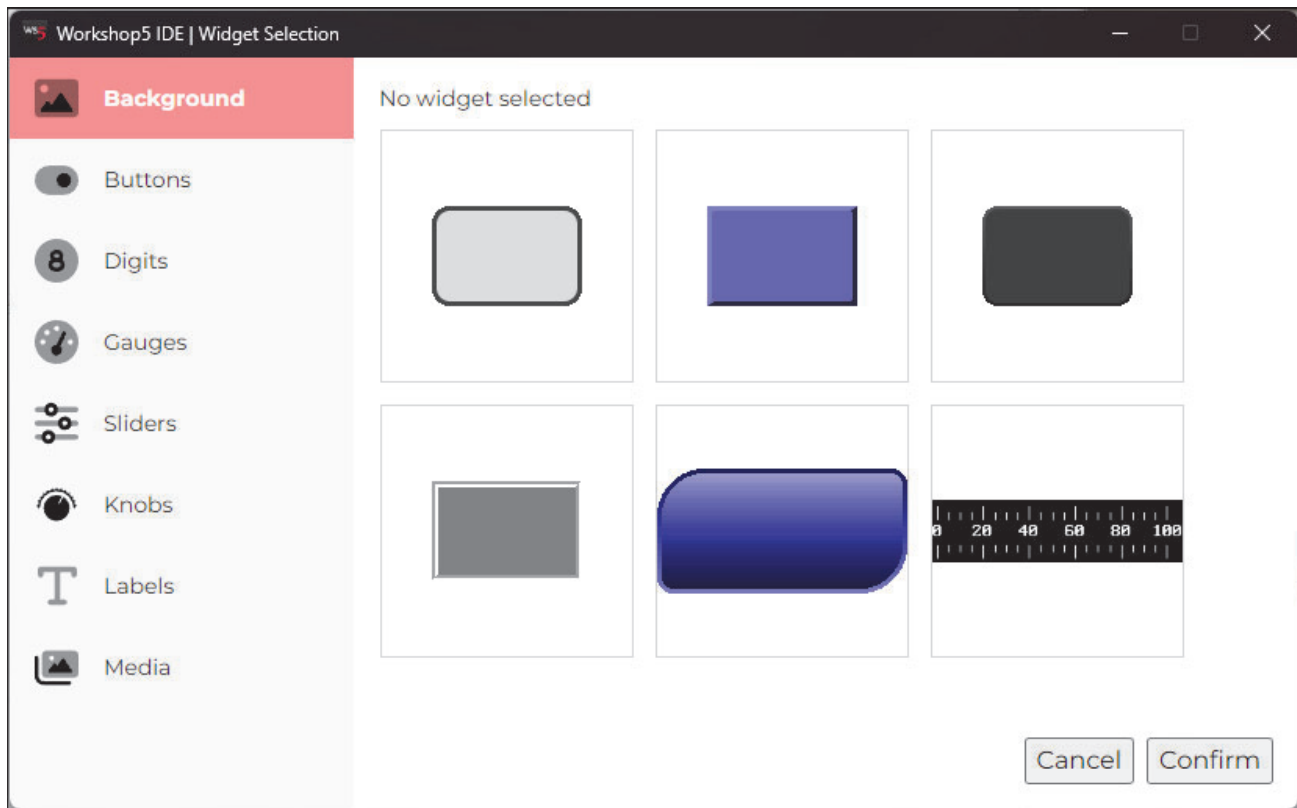
3.1.2.1. Widgets Buttons

Workshop5 introduces a new set of widgets and adapts some from Workshop4, providing support for both the current internal widgets available in Pixxi28, Pixxi44, and Diablo16 displays, and new designs for GCI-based widgets loaded from external storage (such as microSD or external flash chips).

The **widgets** buttons includes all the objects that you can use for your design like Background, Buttons, Digits, Gauges, Sliders, Knobs, Labels, and Media.



If you click any of this widgets buttons will open the **Widget Selection** window where you can choose the widgets that you want to put on the display.



The widget selection window will only show widgets that are supported by the current project processor.

There are three (3) types of widgets that can be used:

- GCI Widgets
- Internal Widgets
- Inherent Widgets

GCI Widgets

The **GCI Widgets** also known as Graphics Composer Image (GCI) widgets, are widgets that are generated by the graphics composer program used in Workshop5 IDE. A GCI widget is composed of a set of frames representing each value.

Internal Widgets

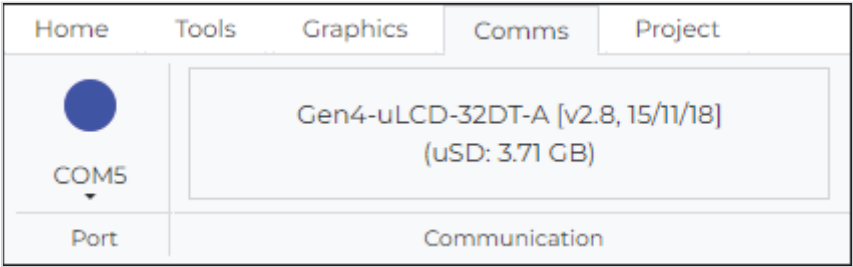
The **Internal Functions (PmmC) Widgets** provides a way to deliver high quality graphical widgets without using any external memory storage device. This type of widget is generated during runtime by executing widget functions included in the PmmC. This widget type is ideal for standalone applications that requires light graphics.

Inherent Widgets

The **Inherent Widgets** are the latest breakthrough in the 4D display graphics. It provides a way to deliver high quality graphical widgets without the use of microSD card. These widgets are available for Pixxi and Diablo16 processors only.

3.1.3. Comms Menu

This menu oversees the communication port:



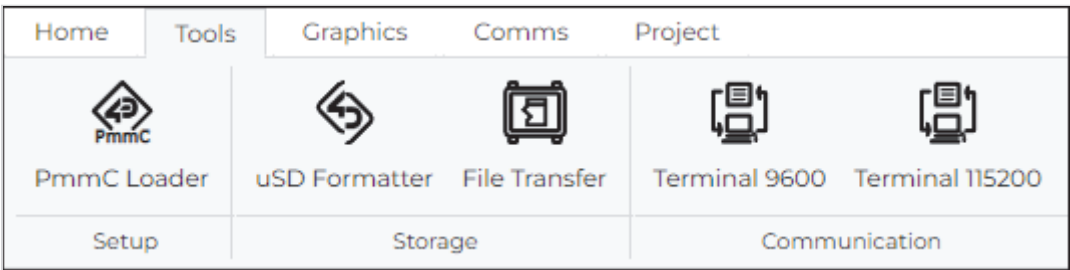
The use of this menu is described at the section [Connect the Module](#).

3.1.4. ViSi Specific Menus

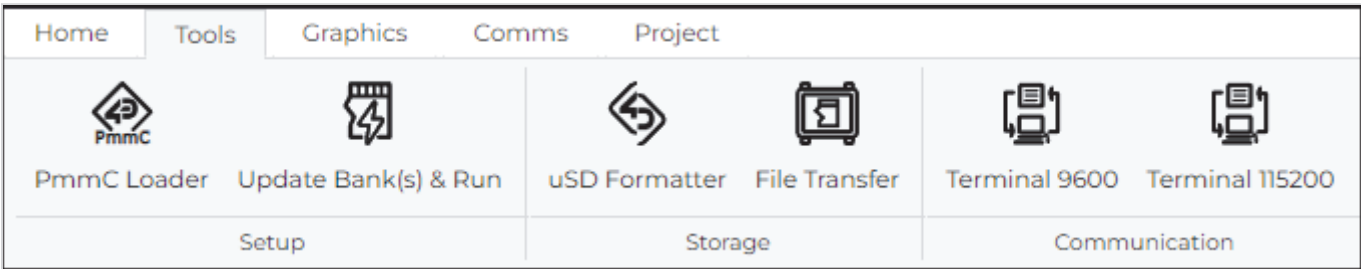
3.1.4.1. Tools Menu

This menu includes all tools and utilities useful for the current project processor.

PICASO AND PIXXI



DIABLO




Tools menu contains the following grouped buttons:

- Setup buttons
- Storage buttons
- Communication buttons



3.1.4.2. Setup Buttons

The Setup buttons includes **PmmC Loader**

Button	Description
 PmmC Loader	Open the PmmC Loader tool to update the PmmC/Driver for the target display.



3.1.4.3. Storage Buttons

The storage buttons includes tools microSD card formatter and file transfer utility.

Button	Description
 uSD Formatter	Open the RMPet tool to partition the uSD card. The uSD card must be mounted to the PC.
 File Transfer	Open the File Transter utility allowing read and write operations to a display modules' uSD or external flash via USB-to-Serial connection.

3.1.4.4. Communication Buttons

The communication buttons includes the terminal tool to communicate to the display module with the current selected port with two (2) different baud rates.

Button	Description
 Terminal 9600	Open the Terminal tool and connect to the currently selected port at 9600 baud.
 Terminal 115200	Open the Terminal tool and connect to the currently selected port at 115200 baud.

3.1.4.5. System Buttons

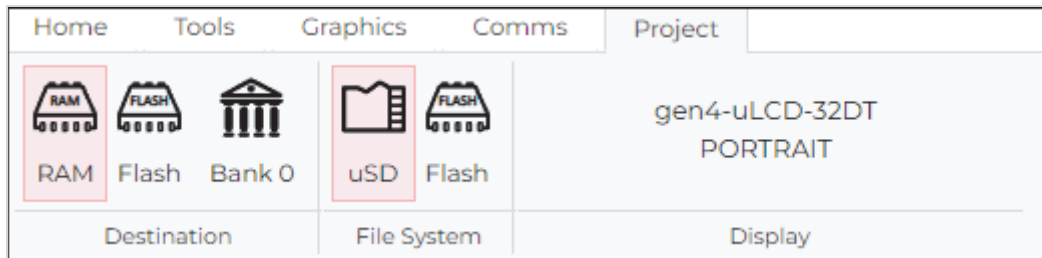
On the systems button includes **Form** button which is use like a page on the screen.



You can add multiple forms to put different widgets and navigate through each form.

3.1.4.6. Projects Menu

The Project menu for ViSi provides additional toggles for the designation of the widgets/media which are used in the Project, as to if they reside in microSD card storage, or Flash memory.

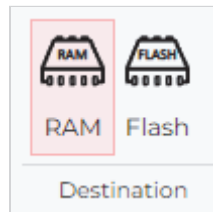


This menu contains the following grouped widgets.

- Destination
- File System
- Display Selection

3.1.4.6.1. Destination Buttons

The following options are for selecting the destination of the compiled code (not images/multimedia etc).

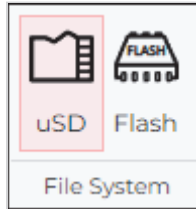


Select the **Destination**:

- **RAM** means the display must be connected during build and that the program will be downloaded to the display's RAM memory once compiled. If **RAM** is chosen as the destination, the program is lost when the display is turned off. This really is for testing while developing, to prevent the Flash cycles being used.
 - **Flash** means the display must be connected during build and that the program will be downloaded to the display's flash memory once compiled. If **Flash** is chosen as the destination, the program is retained and will be available after power cycling. This should be the default for normal use.
-

3.1.4.6.2. File System

This group sets the File System selection, which is to state where the multimedia (widgets, images, etc) will be stored on the display module. Not all display modules have both options, as many will either have a microSD card, or Flash Memory – not both. You will need to select the appropriate one based on your display module.



Selecting **uSD** will target the multimedia to be stored on the microSD card. Selecting **Flash** will target the multimedia to be stored on Flash Memory.



Note

Depending on the Widget used in the Project, will determine what type of storage it requires. There are Internal Widgets, Inherent Widgets, and GCI Widgets.

Internal (PmmC) widgets are based in the PmmC and take up code space of the processor (which is the Flash of the processor itself).

Inherent widgets take up External Flash space on PIXXI based modules, or Flash Bank 5 space on DIABLO modules. Both also take up a little code space on the Processor. Note External Flash is an external chip to the processor, often found on PIXXI based modules. Flash Banks are DIABLO specific and are internal to the DIABLO processor itself.

GCI widgets take up External Flash Space or microSD space on PIXXI-, or microSD space on DIABLO-.

To clarify:

If you have **Internal Widgets** used in your project:

These will be stored on the Processor itself alongside your project code, irrespective if you select **uSD** or **Flash** as the File System, as they are stored internal of the processor.

If you have **Inherent Widgets** in your project:

If you have the **uSD** option selected for the File System, these widgets are unable to be used as they cannot run from microSD card. If you have **Flash** selected, then these widgets will be stored in External Flash on PIXXI based modules, or in a Flash Bank for DIABLO based modules (inside the DIABLO processor).

If you have **GCI Widgets** in your project:

If **uSD** is selected as the File System, then the widgets will be stored on the microSD card for both PIXXI or DIABLO based modules. If **Flash** is selected as the File System, then the widgets will take up External Flash space on PIXXI-, and for DIABLO it is currently not supported but will be coming soon.



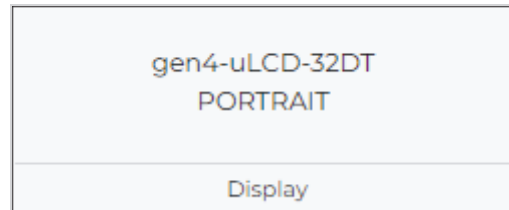
Note

GCI widgets can become large, and External Flash is limited to typically 16MB for 4D Display modules, so there is a very real chance that GCI widgets may not fit in External Flash, and other widget types such as Internal or Inherent may need to be utilised.

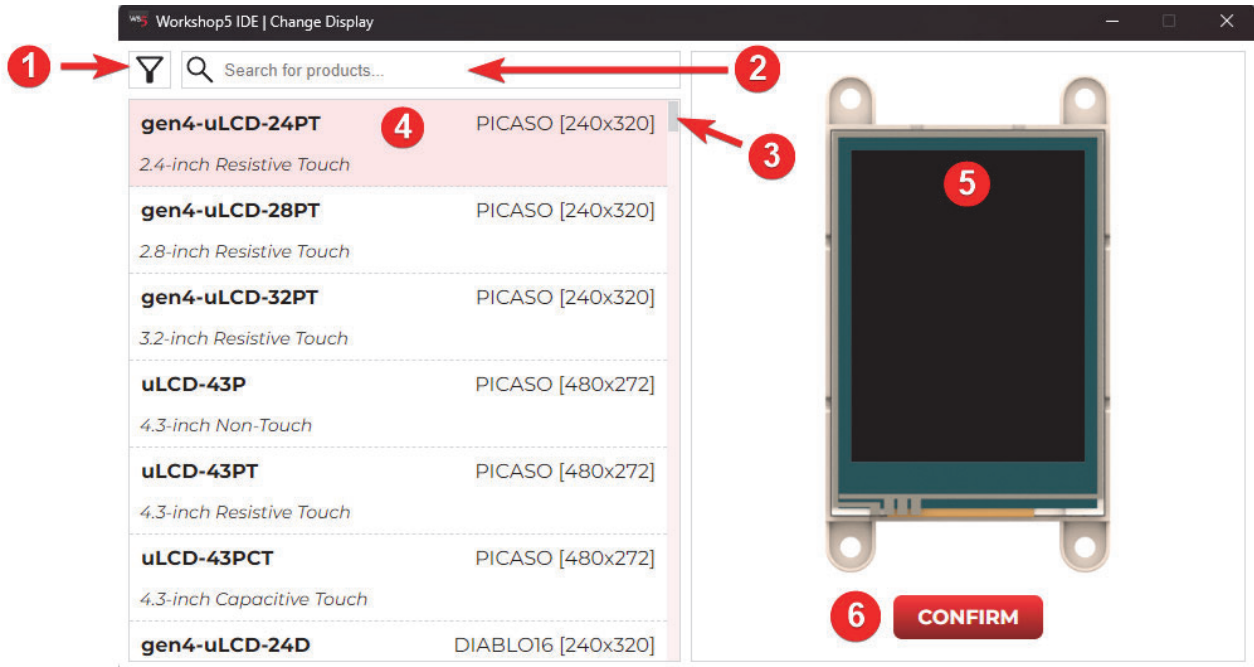
It is important to understand the different widget types **BEFORE** designing your project. Be sure to know what the intended storage File System is that will be used, and therefore which widget type is the most appropriate for your intended project.

3.1.4.6.3. Display Selection

The last section allows selecting the screen, useful for converting a program from one display module to another.



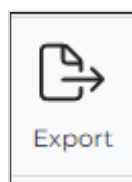
Click the current display to open the Change Display Window.



1. Click filter button to filter the displays by processor.
2. Use the search for product text box to find your target display.
3. Use the slider or scroll the mouse wheel to find your target display.
4. Click the target display to select.
5. Click the image to change display orientation.
6. Click **CONFIRM** button to apply the changes.

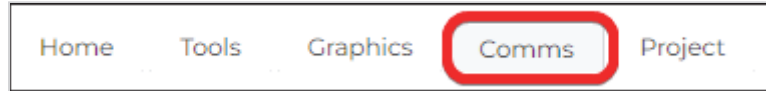
3.1.4.7. Home Menu

The pico environment has additional **Export** button on home menu compile related groups.

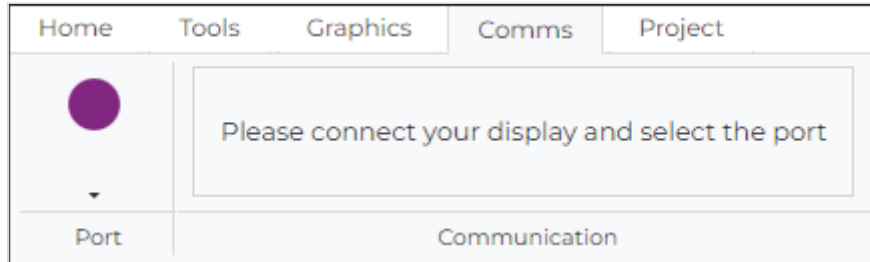


4. Connect the Module

Connect the module to a USB port with the 4D Systems programming cable and select the **Comms** menu:



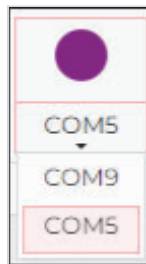
Above the Comms section, the **violet** light mentions no module is currently connected.



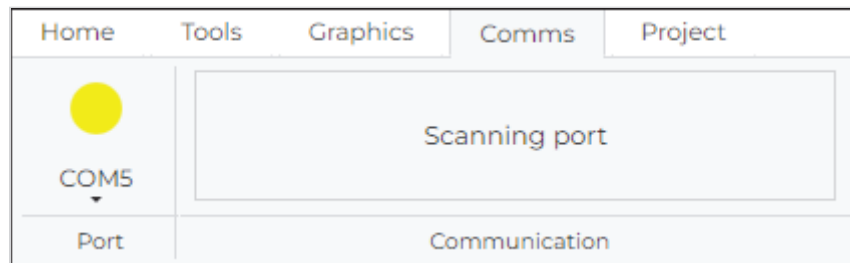
Connect the 4D Systems programming cable/adaptor to the module and plug the cable into the USB port. Click on the drop-down list.



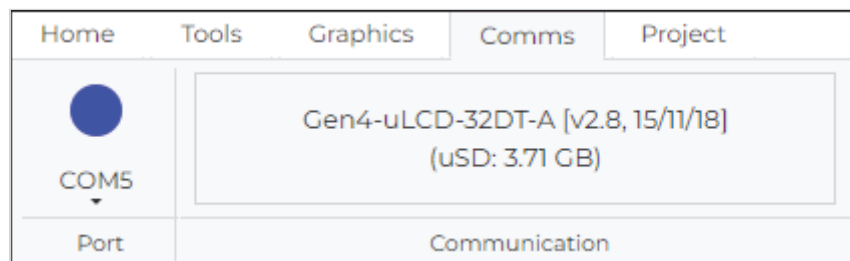
Select the COM port relating to the 4D Programming cable/adaptor.



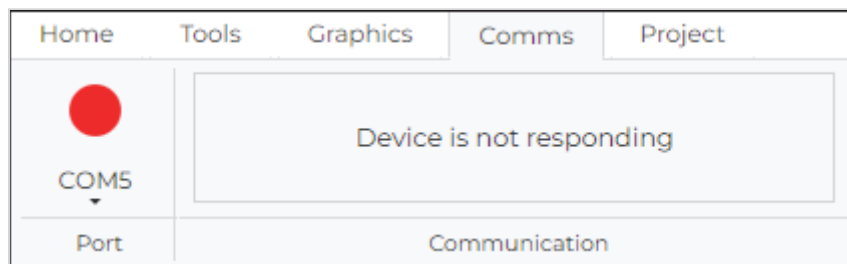
The light turns **yellow** while the connection is being established:



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



The light turns **red** when no module is attached to the selected port:



5. Hardware Setup

4D Systems display modules are classified as gen4-uLCD, microLCD, microOLED and pixxiLCD displays.

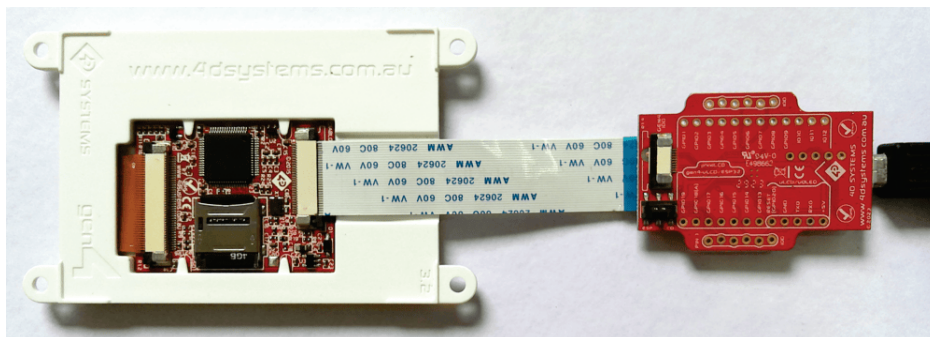
5.1. gen4-uLCD and pixxiLCD Displays

The gen4 series was designed specifically for ease of integration and use, with careful consideration for space requirements and functionality.

The gen4 series of Integrated Display Modules features a 30 pin ZIF/FFC/FPC socket, designed for a 30 pin FPC cable, for easy and simple connection to an application or mother board, or for connecting to accessory boards for a range of functionality advancements.

5.1.1. Using 4D-UPA

The 4D-UPA is a universal programmer that can be easily connected to the display using a 30-pin Flexible Flat Cable.

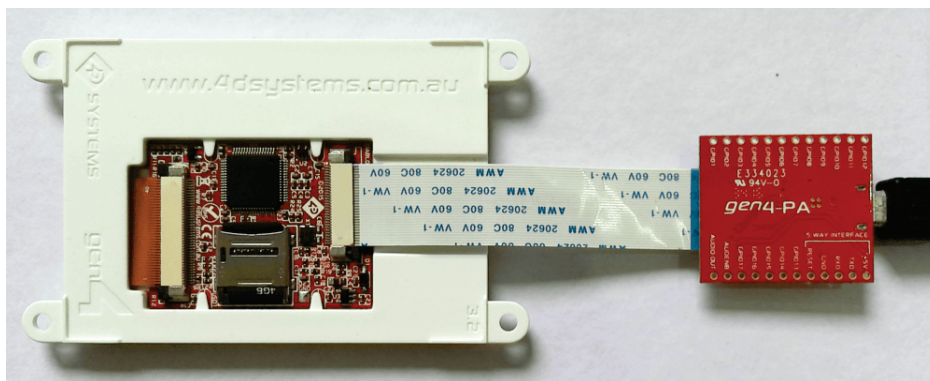


Warning

Be careful in unlocking the 30-pin FPC socket from the display and 4D-UPA to prevent it from breaking. Also be careful to knock off components near the socket.

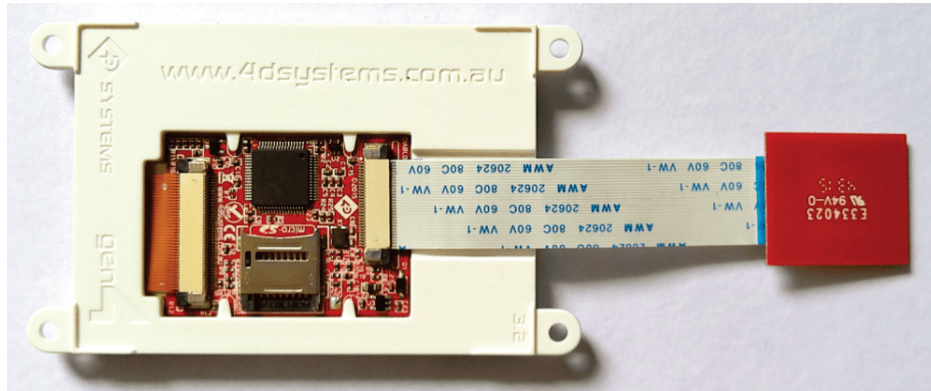
5.1.2. Using gen4-PA

The gen4-PA is a programming adaptor for the gen4 display module that also breaks out the signals found on the 30-way FFC cable coming from the gen4 display module.



5.1.3. Using gen4-IB

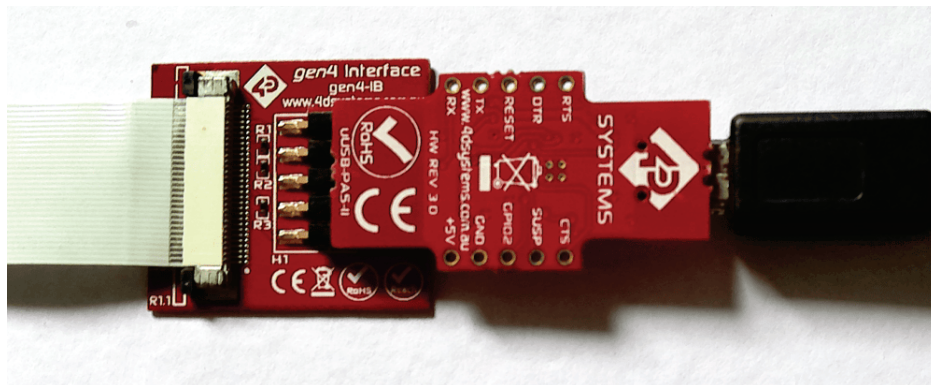
The gen4-IB can be connected to the gen4 display with a 30 pin Flexible Flat Cable.



Then, the gen4-IB can be connected using uUSB-PA5-II or the Programming Cable.

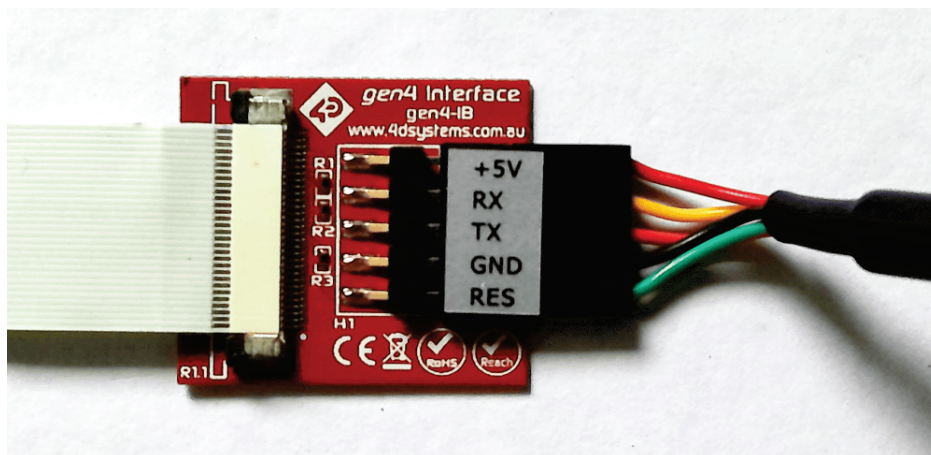
5.1.3.1. gen4-IB with uUSB-PA5-II

The uUSB-PA5-II can be connected to the gen4-IB as shown.



5.1.3.2. gen4-IB with Programming Cable

The programming cable can be connected to the gen4-IB as shown.

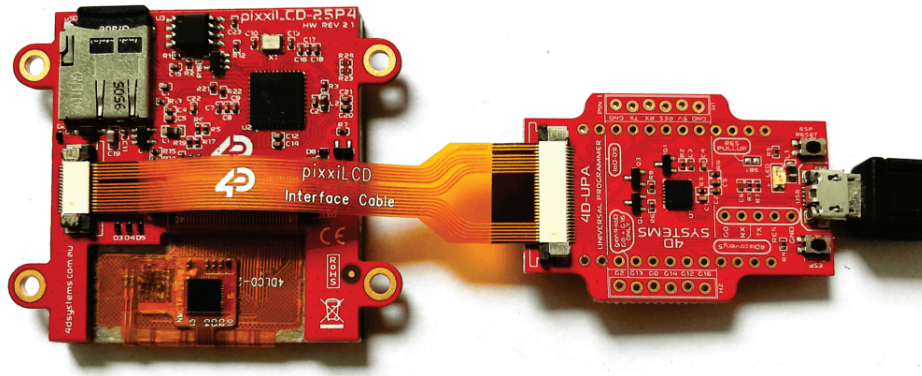


5.1.4. pixxiLCD Displays

The pixxiLCD series use a 15-way 0.5mm pitch ZIF Socket interface which can be connected to 4D-UPA or gen4-IB.

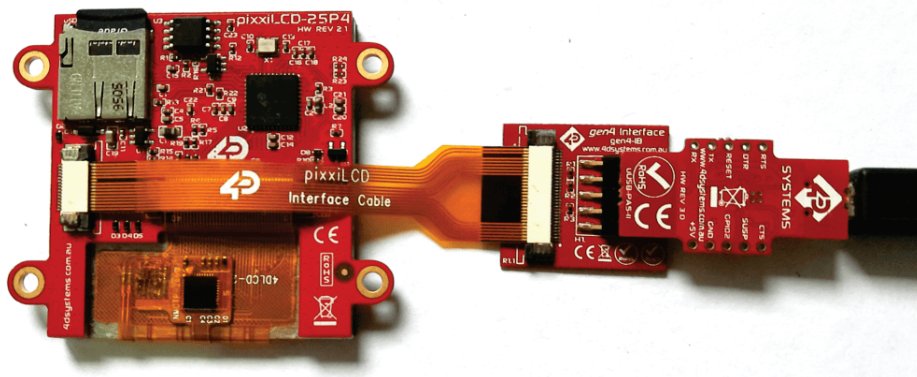
5.1.4.1. Using 4D-UPA

The 4D-UPA can be connected to the pixxiLCD display through a 15-way-to-30-way cable.



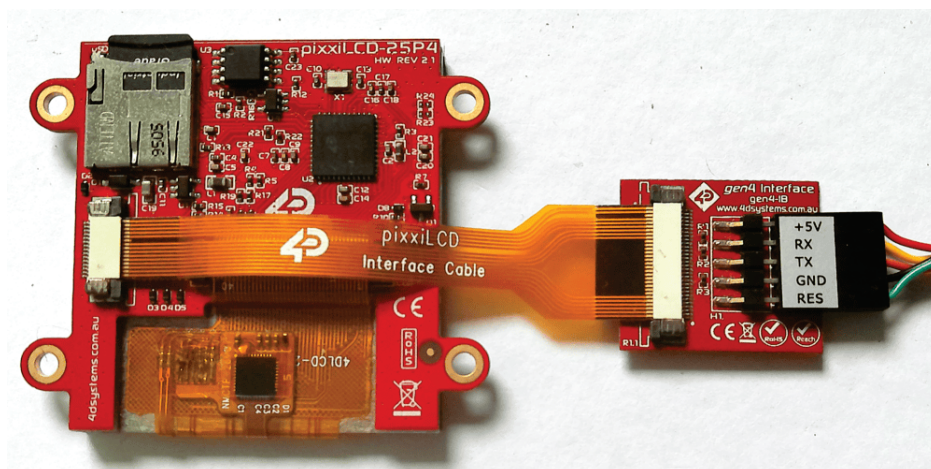
5.1.4.2. Using gen4-IB with uUSB-PA5-II

The gen4-IB with uUSB-PA5-II can be connected to the pixxiLCD display through a 15-way-to-30-way cable.



5.1.4.3. Using gen4-IB with Programming Cable

The gen4-IB with programming cable can be connected to the pixxiLCD display through a 15-way-to-30-way cable.



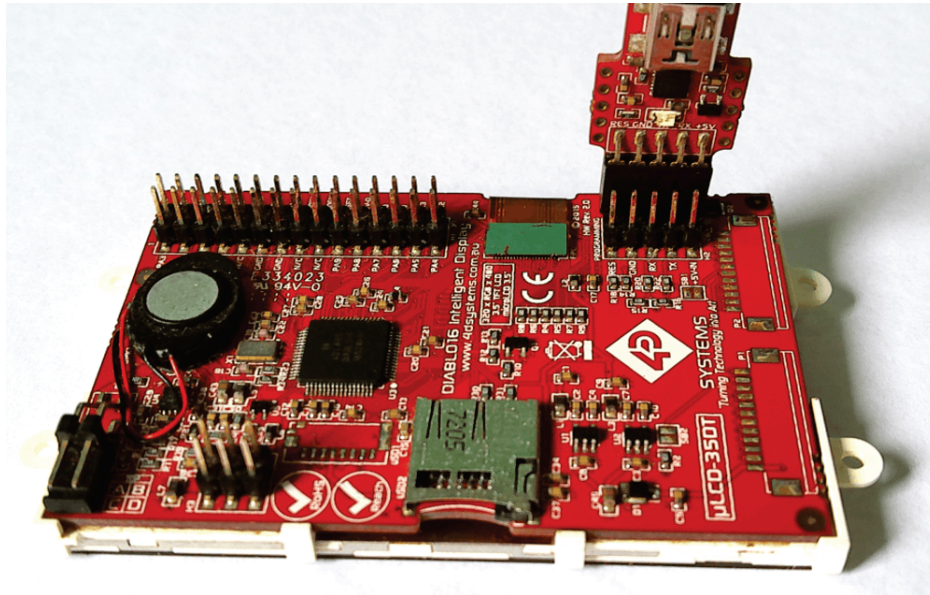
5.2. microLCD and microOLED Displays

5.2.1. microLCD Displays

The microLCD series modules was designed with versatility in mind, allowing microUSB-PA5-II and programming cable to connect to the 5-pin header pins for easy access.

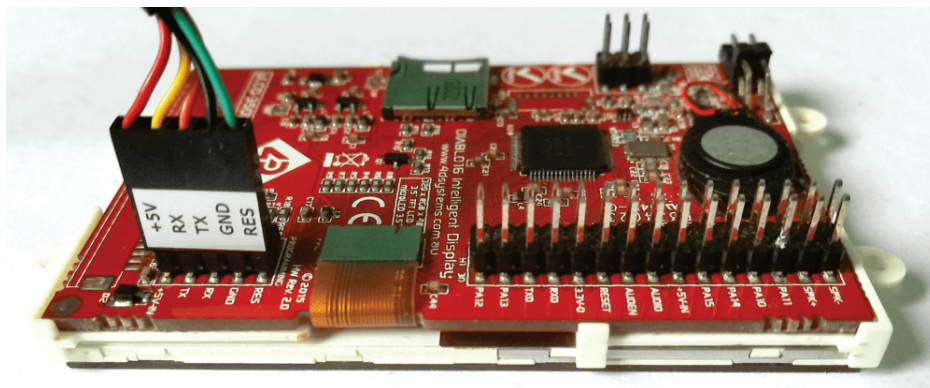
5.2.1.1. Using uUSB-PA-5

The uUSB-PA5-II can be connected as shown.



5.2.1.2. Using Programming Cable

The programming cable can be connected as shown.

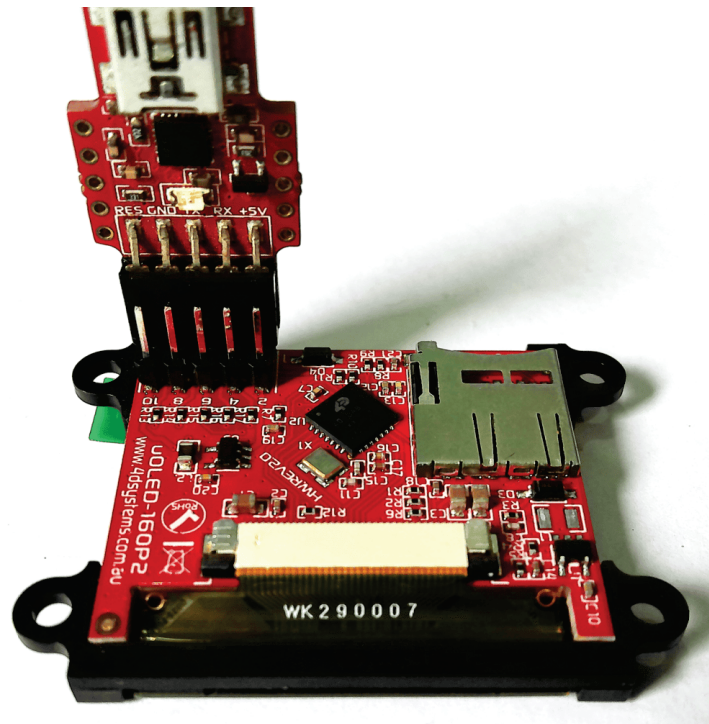


5.2.2. microOLED

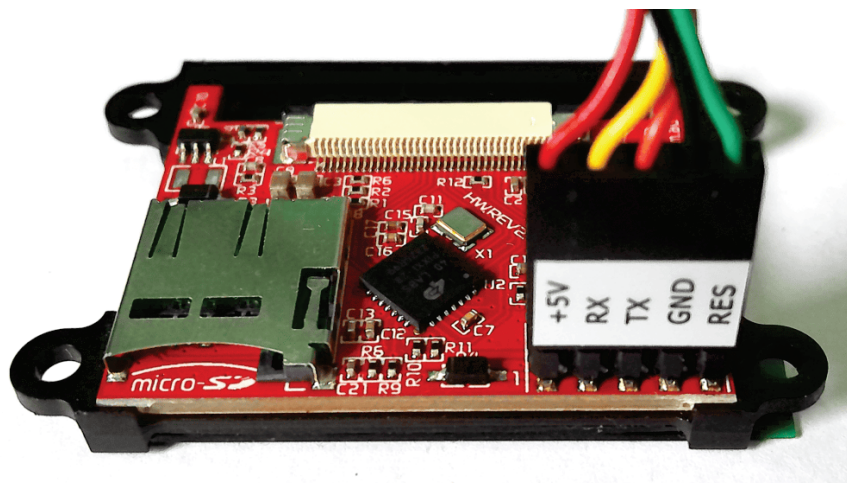
The microOLED series can be connected using the 10-pin header interface.

5.2.2.1. Using uUSB-PA5-II

The uUSB-PA5-II can be connected as shown.



5.2.2.2. Using Programming Cable



6. Workshop5 Widgets

Workshop5 provides a drag n' drop design tool for most of its powerful development environments. This allows user to easily utilise three types of widgets in their projects:

- GUI Widgets
- Internal Functions (PmmC) Widgets
- Inherent Widgets

This offers increased flexibility in creating applications depending on the hardware requirement. These widgets include the following: buttons, sliders, knobs, gauges, LED indicator, LED digit, strings, static labels, media, and images. Additional non-GUI widgets are also available that allows the user to add miscellaneous functionalities like sound, hardware I/O, or supporting resources like fonts, depending on the target processor and environment used. All these widgets are available in the Widgets Menu of the ViSi and ViSi-Genie environment of Workshop5 IDE. Given its high configurability, these widgets can be customized directly on the What You See Is What You Get (WYSIWYG) screen or through the **Object Inspector**.

7. Revision History

<div><div></div>Document Revision</div>		
Revision Number	Date	Content
1.0	04/11/2024	Initial Release

8. Legal Notice

8.1. Proprietary Information

The information contained in this document is the property of 4D Systems Pty. Ltd. and may be the subject of patents pending or granted, and must not be copied or disclosed without prior written permission. 4D Systems endeavours to ensure that the information in this document is correct and fairly stated but does not accept liability for any error or omission. The development of 4D Systems products and services is continuous and published information may not be up to date. It is important to check the current position with 4D Systems. 4D Systems reserves the right to modify, update or makes changes to Specifications or written material without prior notice at any time.

All trademarks belong to their respective owners and are recognised and acknowledged.

8.2. Disclaimer of Warranties & Limitations of Liabilities

4D Systems makes no warranty, either expressed or implied with respect to any product, and specifically disclaims all other warranties, including, without limitation, warranties for merchantability, non-infringement and fitness for any particular purpose.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

Images and graphics used throughout this document are for illustrative purposes only. All images and graphics used are possible to be displayed on the 4D Systems range of products, however the quality may vary.

In no event shall 4D Systems be liable to the buyer or to any third party for any indirect, incidental, special, consequential, punitive or exemplary damages (including without limitation lost profits, lost savings, or loss of business opportunity) arising out of or relating to any product or service provided or to be provided by 4D Systems, or the use or inability to use the same, even if 4D Systems has been advised of the possibility of such damages.

4D Systems products are not fault tolerant nor designed, manufactured or intended for use or resale as on line control equipment in hazardous environments requiring fail – safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of the product could lead directly to death, personal injury or severe physical or environmental damage ('High Risk Activities'). 4D Systems and its suppliers specifically disclaim any expressed or implied warranty of fitness for High Risk Activities.

Use of 4D Systems' products and devices in 'High Risk Activities' and in any other application is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless 4D Systems from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any 4D Systems intellectual property rights.
