



USER MANUAL

Workshop4 IDE USER GUIDE

Document Revision: 2.1
Document Date: 29th July 2017

Table of Contents

1. Description	4
2. Workshop4	5
3. Create a New Project.....	6
4. Choose Your 4D Environment	10
4.1. Designer	11
4.2. ViSi	12
4.3. ViSi-Genie.....	13
4.4. Serial.....	14
4.5. Create System File.....	15
4.6. Create Text File	16
5. Choose Your Arduino Compatible Environment	17
5.1. Basic Graphics	18
5.2. Extended Graphics	18
5.3. Genie Graphics.....	18
6. Common File Menu	19
6.1. File-Related Buttons	20
6.2. Print-Related Buttons.....	23
6.3. Miscellaneous Buttons.....	24
7. Designer Specific Menus.....	28
7.1. Home Menu	28
7.1.1. File-Related Buttons.....	28
7.1.2. Code-Related Buttons	28
7.1.3. Bookmark Buttons	29
7.1.4. Find and Replace Buttons.....	29
7.1.5. Code Folding Buttons	30
7.1.6. Compile Buttons	31
7.2. Tools Menu.....	32
7.3. Comms Menu.....	32

7.4. Project Menu	33
7.4.1. Destination	33
7.4.2. Enhancements	33
7.4.3. Display Selection	34
8. ViSi Specific Menus	36
8.1. View Menu	36
8.2. Widgets Menu	37
9. ViSi-Genie Specific Menus	38
9.1. Home Menu	38
9.1.1. File-Related Buttons.....	38
9.1.2. Build Button.....	38
9.1.3. Objects Pane.....	39
9.2. View Menu	39
9.3. Tools Menu.....	40
9.4. Comms Menu.....	41
9.5. Project Menu	41
9.5.1. Genie Options	42
9.5.2. Parameters	43
9.5.3. Display Selection	44
10. Basic and Extended Graphics Specific Menus	45
11. Connect the Module	46
12. Insert the Micro-SD Card	47
13. Revision History	48
14. Legal Notice	49
15. Contact Information	49

1. Description

This user guide provides an introduction to Workshop4, the 4D Labs integrated development environment. Workshop4 supports multiple development environments for the user, to cater for different user requirements and skill level.



The **Designer** environment enables the user to write 4DGL code in its natural form to program 4D Labs processors of choice.



A visual programming experience, suitably called **ViSi**, enables drag-and-drop type placement of objects to assist with 4DGL code generation and allows the user to visualise how the display will look while being developed.



An advanced environment called **ViSi-Genie** doesn't require any 4DGL coding at all, it is all done automatically. Simply lay the display out with the objects required, set the events to drive them and the code is written for the user automatically. ViSi-Genie provides the latest rapid development experience from 4D Labs. (Not available for Goldelox)



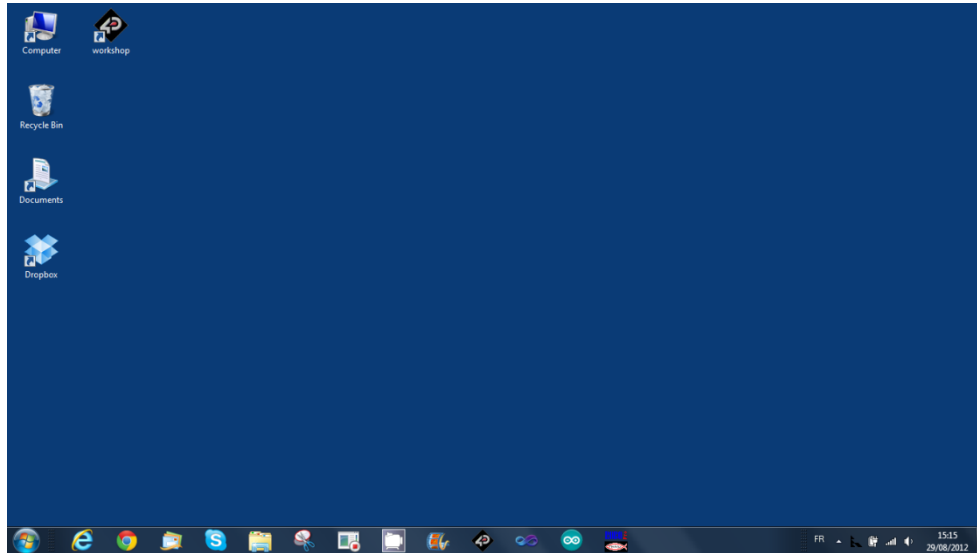
A **Serial** environment is also provided to transform display modules powered by 4D Labs processors into a slave serial module, allowing the user to control the display from any host microcontroller or device with a serial port.

Additionally, Workshop4 also offers Arduino compatible environments that allows the user to easily create a project with both a 4D and an Arduino product. More details can be found in [Choose Your Arduino Compatible Environment](#).

To install Workshop4, please refer to the document [Workshop4 Installation](#).

2. Workshop4

There is an alias for 4D Workshop on the desktop:



Launch 4D Workshop by double-clicking on the icon:



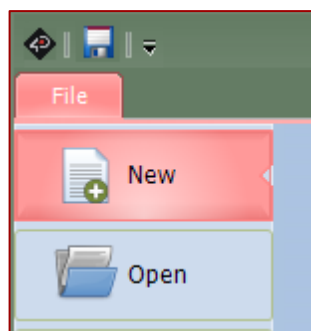
3. Create a New Project

Workshop4 opens and displays the **Recent** page:

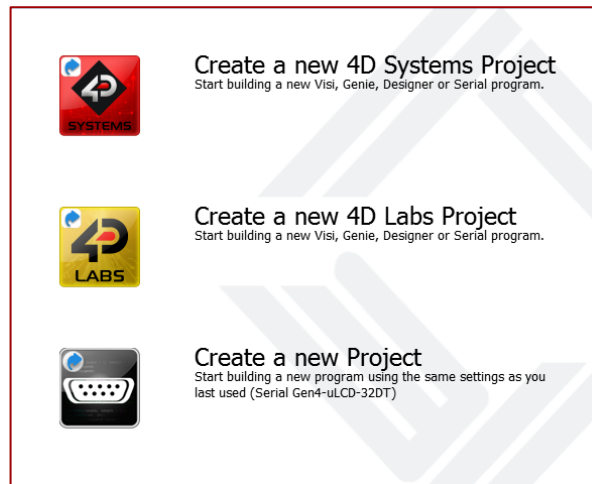


To create a new program, there are multiple options:

- Click on the top left-most icon **New**

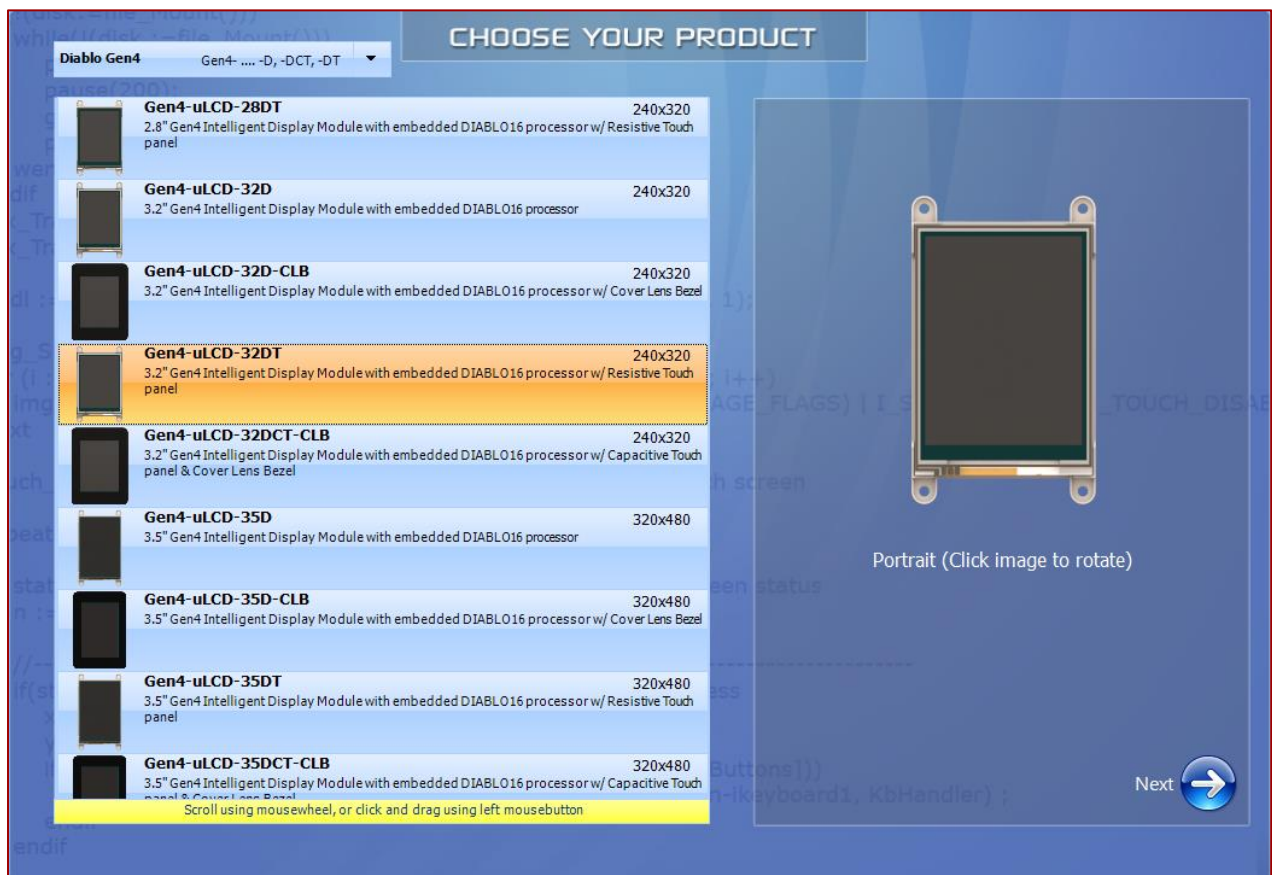


- Click on the icon close to **Create a new 4D Systems Project** on top
- Click on the **Create a new Project** to create a project instance based on your last project settings.

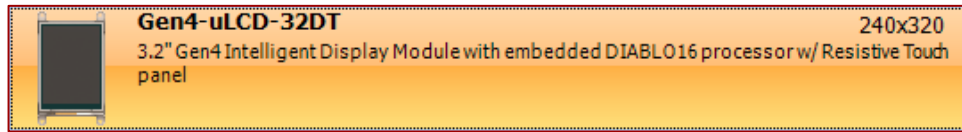


Note: Another option is displayed on the image above. **Create a new 4D Labs Project** is an upcoming Workshop4 feature which allows to create a project with a customized display module powered by 4D Labs' processors

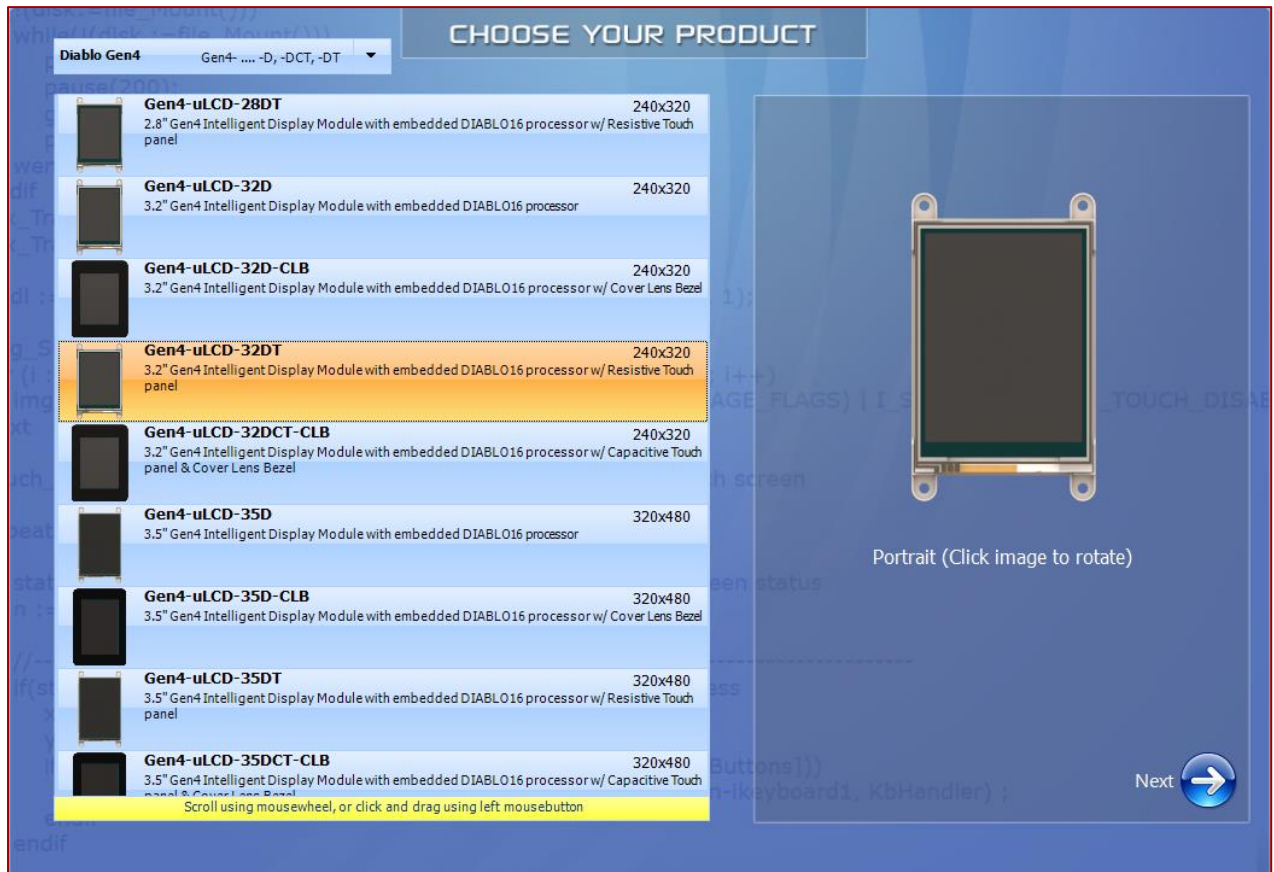
Both of the first two options update the main window with the selection of the screen:



Select the screen, here the gen4-uLCD-32DT



The selected screen is displayed:



Orientation is portrait by default. To set it to landscape, just click on the image of the screen to rotate it:

CHOOSE YOUR PRODUCT

Diablo Gen4 Gen4- -D, -DCT, -DT

	Gen4-uLCD-28DT 2.8" Gen4 Intelligent Display Module with embedded DIABLO16 processor w/ Resistive Touch panel	240x320
	Gen4-uLCD-32D 3.2" Gen4 Intelligent Display Module with embedded DIABLO16 processor	240x320
	Gen4-uLCD-32D-CLB 3.2" Gen4 Intelligent Display Module with embedded DIABLO16 processor w/ Cover Lens Bezel	240x320
	Gen4-uLCD-32DT 3.2" Gen4 Intelligent Display Module with embedded DIABLO16 processor w/ Resistive Touch panel	240x320
	Gen4-uLCD-32DCT-CLB 3.2" Gen4 Intelligent Display Module with embedded DIABLO16 processor w/ Capacitive Touch panel & Cover Lens Bezel	240x320
	Gen4-uLCD-35D 3.5" Gen4 Intelligent Display Module with embedded DIABLO16 processor	320x480
	Gen4-uLCD-35D-CLB 3.5" Gen4 Intelligent Display Module with embedded DIABLO16 processor w/ Cover Lens Bezel	320x480
	Gen4-uLCD-35DT 3.5" Gen4 Intelligent Display Module with embedded DIABLO16 processor w/ Resistive Touch panel	320x480
	Gen4-uLCD-35DCT-CLB 3.5" Gen4 Intelligent Display Module with embedded DIABLO16 processor w/ Capacitive Touch panel & Cover Lens Bezel	320x480

Scroll using mousewheel, or click and drag using left mousebutton

Landscape (Click image to rotate)

Next

Press **Next** to proceed:



4. Choose Your 4D Environment

The main window now asks for the environment for the project:



Four main environment options are available:

- Designer,
- ViSi,
- ViSi-Genie,
- Serial

...and two editor options:

- Create System File,
- Create Text File.

Each option opens a customised environment with specific commands and controls. Different projects using different modules and environments can be opened simultaneously on Workshop4. The toolbar ribbon will adapt to the project tab that is selected, to suit its environment.

4.1. Designer



The **Designer** environment enables the user to write 4DGL code in its natural form to program the display module. 4DGL is a graphics oriented language allowing rapid application development, and the syntax structure was designed using elements of popular languages such as C, Basic, Pascal and others.

```

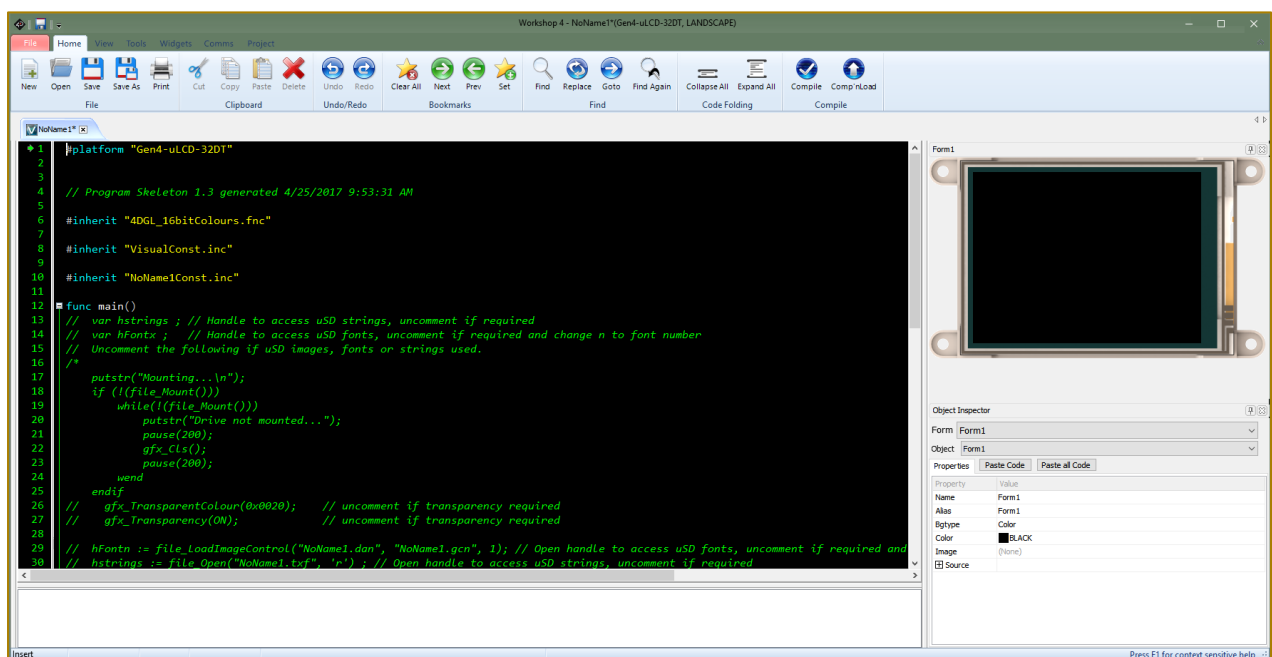
Workshop 4 - NoName1*(Gen4-uLCD-32DT)
File Home Tools Comms Project
New Open Save Save As Print Cut Copy Paste Delete Undo Redo Clear All Next Prev Set Find Replace Goto Find Again Collapse All Expand All Compile Comp nLoad
File Clipboard Undo/Redo Bookmarks Find Code Folding Compile
NoName1* [x]
1 #platform "Gen4-uLCD-32DT"
2
3
4 #inherit "4DGL_16bitColours.fnc"
5
6 func main()
7
8     gfx_ScreenMode(LANDSCAPE) ; // change manually if orientation change
9
10    print("Hello World") ; // replace with your code
11
12    repeat // maybe replace
13    forever // this as well
14
15 endfunc
16
Insert Press F1 for context sensitive help ...
  
```

To learn more about Designer, please refer to the 4DGL Internal Functions Reference Manual and 4DGL Programmers Reference Manual on the Workshop4 product [page](#) and to related [application notes](#).

4.2. ViSi



A visual programming experience, suitably called **ViSi**, enables drag-and-drop type placement of objects to assist with 4DGL code generation and allows the user to visualise how the display will look while being developed.

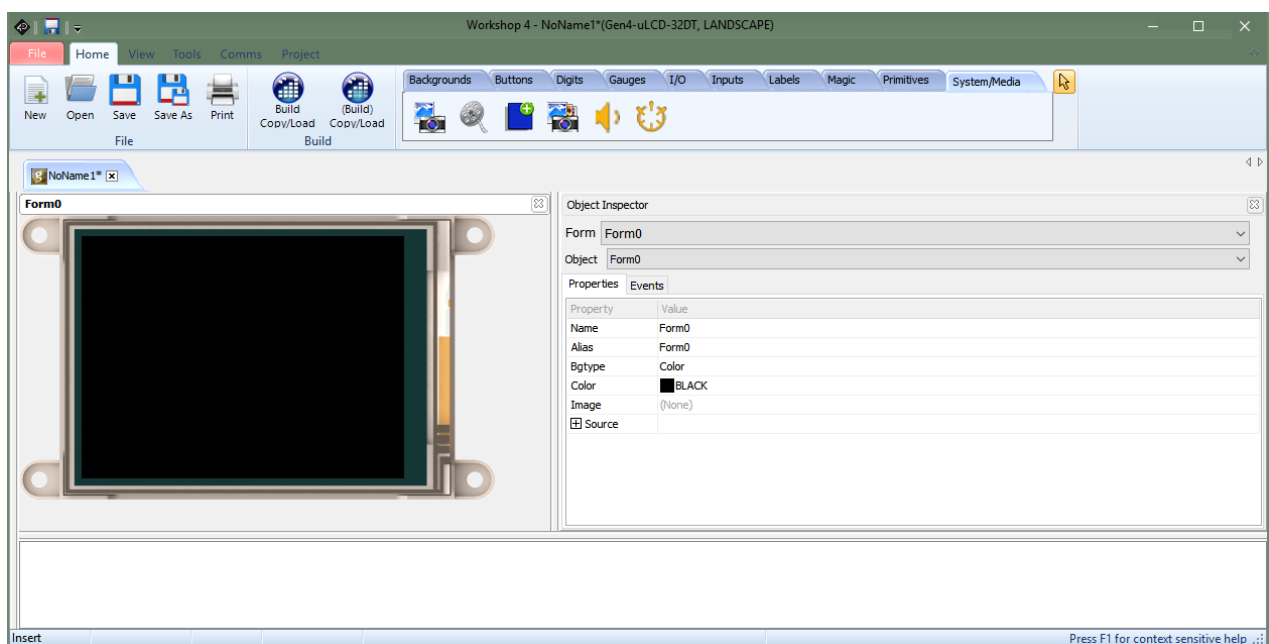


To learn more about ViSi, please refer to the ViSi User Guide on the Workshop4 product [page](#) and to related [application notes](#).

4.3. ViSi-Genie

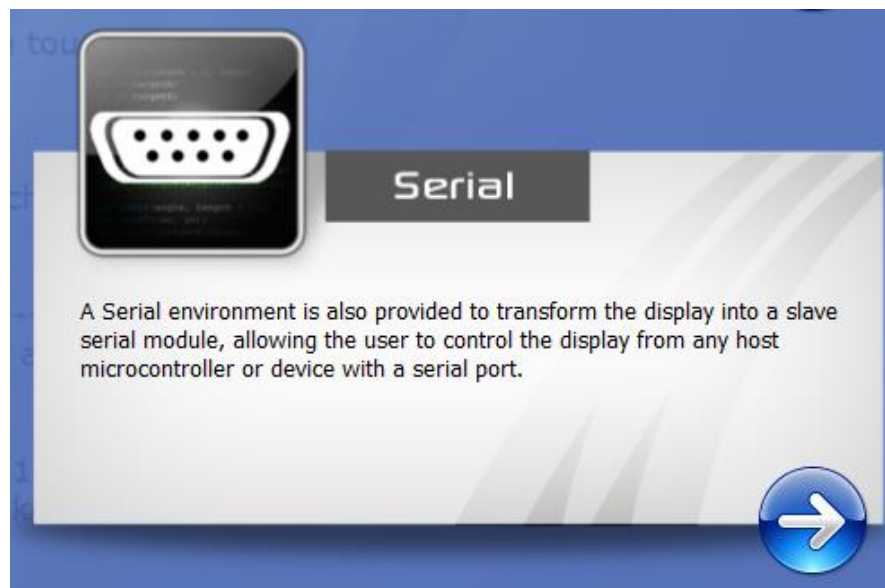


An advanced environment called **ViSi-Genie** doesn't require any 4DGL coding at all, it is all done for the user. By simply laying the display out with the wanted objects and setting the events to drive them, the code is written automatically. ViSi-Genie provides the latest rapid development experience from 4D Labs.



To learn more about ViSi-Genie, please refer to the ViSi-Genie User Guide on the Workshop4 product [page](#) and to related [application notes](#).

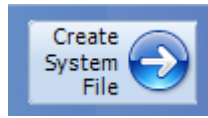
4.4. Serial



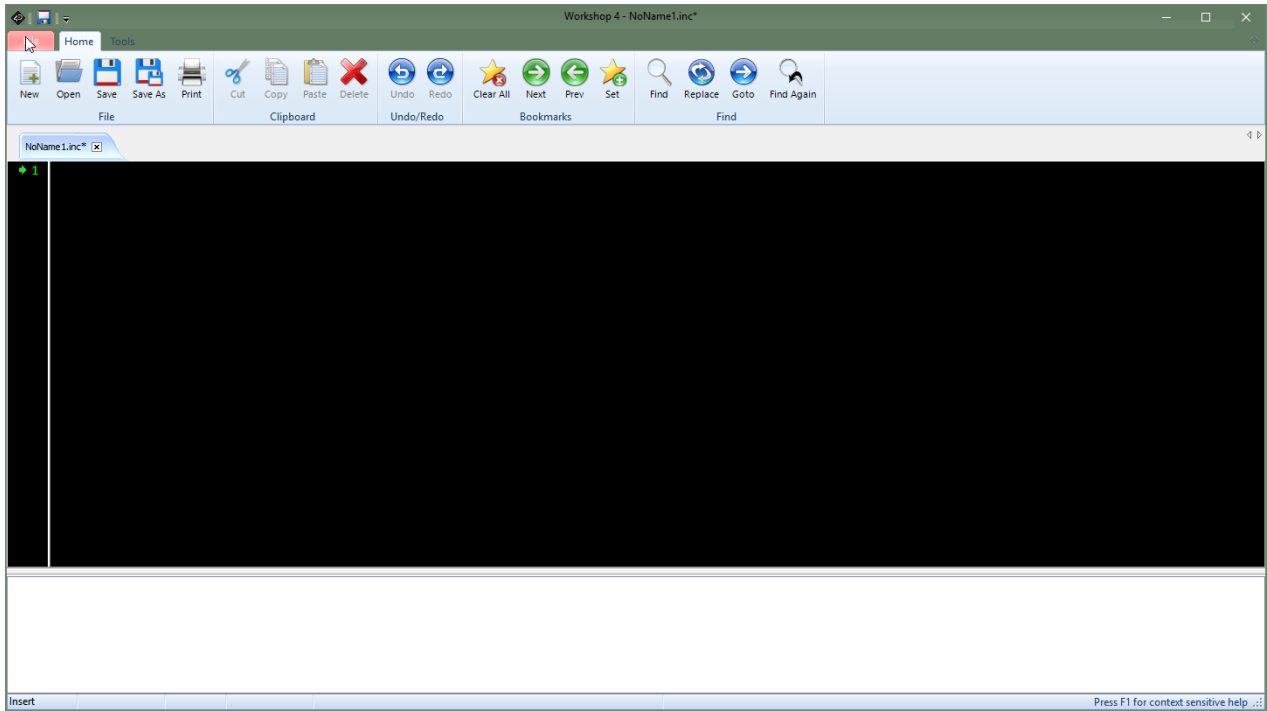
The **Serial** environment is provided to transform the module into a slave serial module, allowing communication and control from virtually any host micro-controller or device with a serial port.

For detail on this environment and steps on how to make your display module run under the Serial environment, please refer to the Serial Command Set Reference Manual on the Workshop4 product [page](#) and to related [application notes](#).

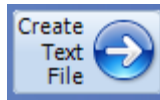
4.5. Create System File



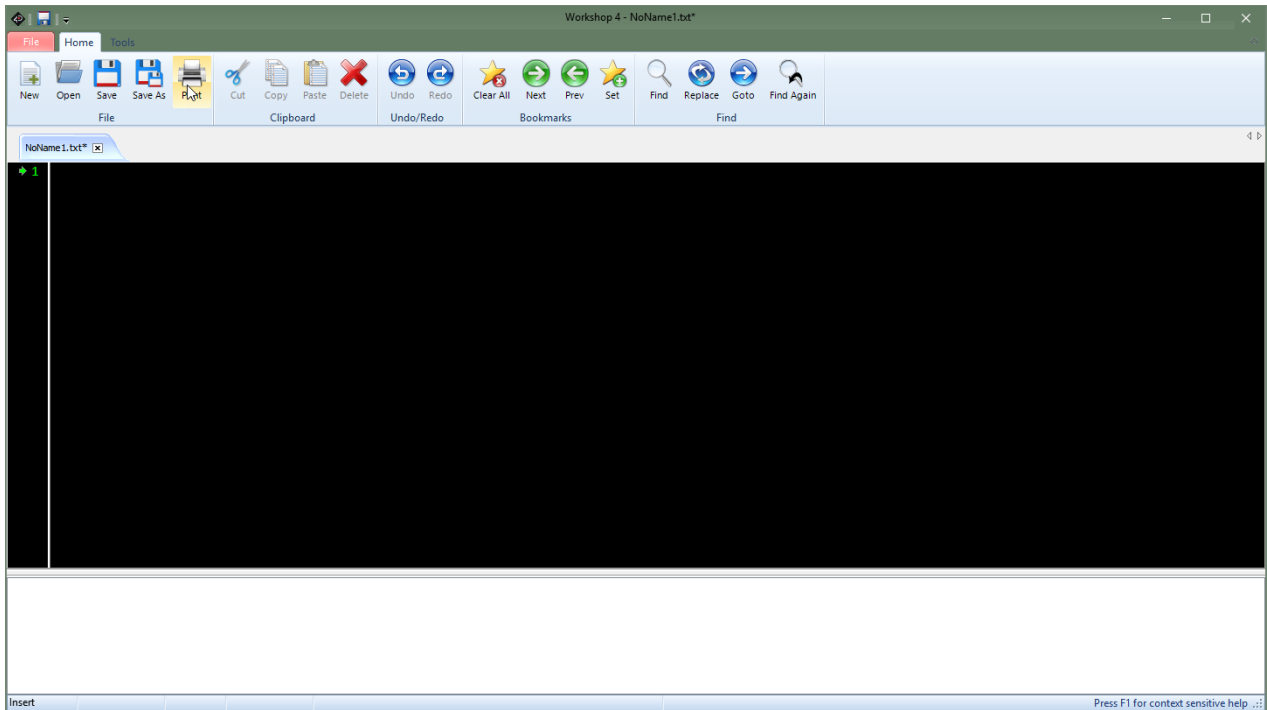
The **Create System File** option provides an editor for all 4DGL-related projects, so a user can create or edit a 4DGL Include file, 4DGL Library file, a Function or System file. These can then be included in the user's 4DGL code.



4.6. Create Text File

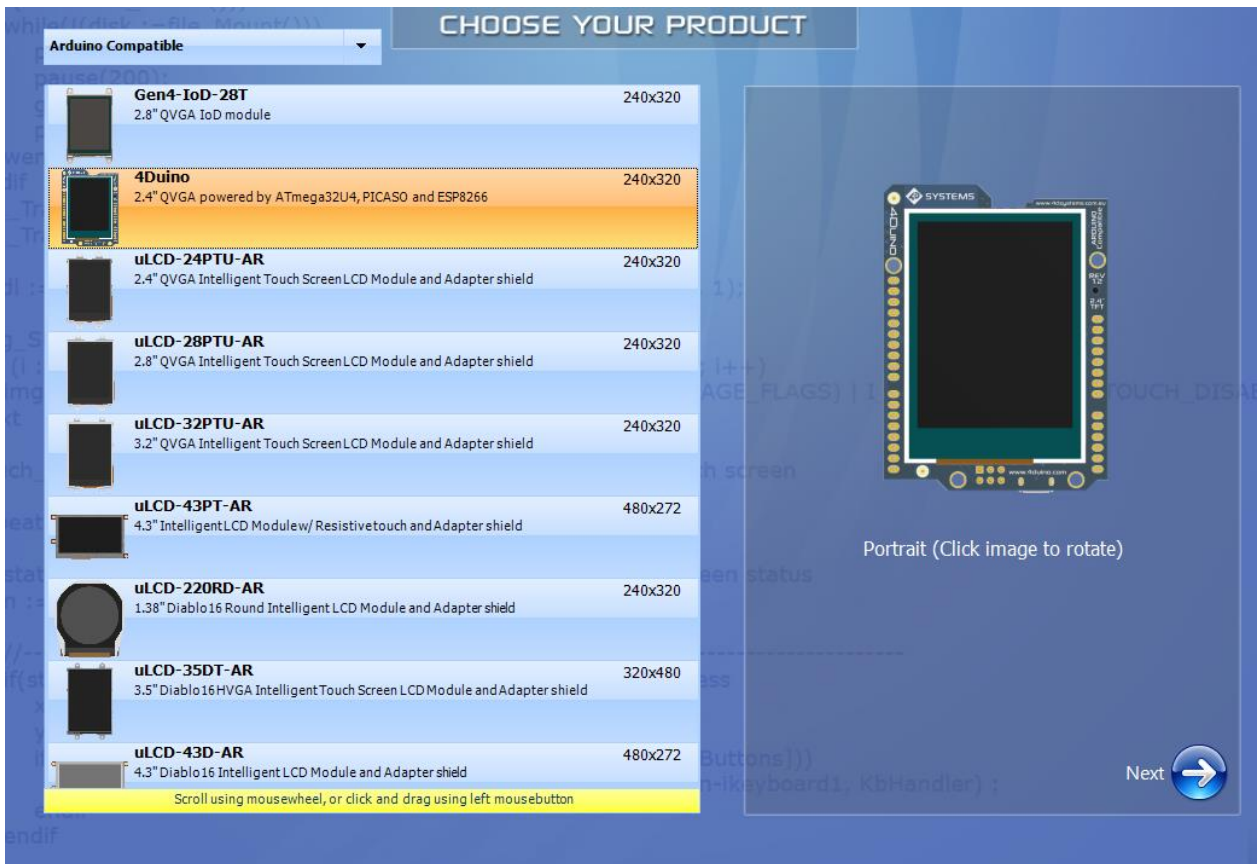


The **Create Text File** environment features a plain editor, suitable for writing basic documentation, application notes, data files or anything else requiring plain text files.



5. Choose Your Arduino Compatible Environment

When choosing a 4D Systems product, you can see that there is a product group named **Arduino Compatible** which includes both gen4-IoD-28T and 4Duino. It also includes a wide range of 4D Systems' display modules setup with an Arduino board.



Workshop4 allows the users to write their Arduino code and customize their 4D graphics user interface at the same time in a single integrated development environment.



Basic Graphics

The Arduino compatible Basic Graphics environment enables the user to write Arduino code directly to program this Arduino compatible module. It requires no uSD card and allows graphics primitives to be dragged and dropped on the screen and placed in your code. It utilises the Serial SPE library for the processor used in the display, and therefore embraces the full set of Serial SPE Commands are available to the User, to produce the Graphical User Interface required.





Extended Graphics

A visual programming experience, suitably called Arduino compatible Extended Graphics, enables drag-and-drop type placement of Workshop4 objects to assist with Arduino code generation and allows the user to visualise how the display will look while being developed. A uSD card will be required in order to hold the graphics. It utilises the Serial SPE library for the processor used in the display, and therefore embraces the full set of Serial SPE Commands are available to the User, to produce the Graphical User Interface required.





Genie Graphics

An advanced environment called Arduino compatible Genie Graphics utilises the latest VISI-Genie technology, which is available for other 4D Systems Intelligent Display Modules. It utilises the genieArduino library, and provides rapid development of a GUI due to the powerful tools and widgets Workshop4 provides. Simply lay the display out with the widgets you want, set the events to drive the GUI, then utilise the genie functions to connect the Arduino sketch to read or write information to those objects from the Atmel processor. This is a very clean and rapid development environment.

In the meantime you can continue developing Genie Arduino applications using the current tools



5.1. Basic Graphics



The Arduino compatible Basic Graphics environment enables the user to write Arduino code directly to program this Arduino compatible module. It requires no uSD card and allows graphics primitives to be dragged and dropped on the screen and placed in your code. It utilizes the Serial SPE library for the processor used in the display, and therefore embraces the full set of Serial SPE Commands available to the User, to produce the Graphical User Interface required.

5.2. Extended Graphics



A visual programming experience, suitably called Arduino compatible Extended Graphics, enables drag-and-drop type placement of Workshop4 objects to assist with Arduino code generation and allows the user to visualize how the display will look while being developed. A uSD card will be required in order to hold the graphics. It utilizes the Serial SPE library for the processor used in the display, and therefore embraces the full set of Serial SPE Commands available to the User, to produce the Graphical User Interface required.

5.3. Genie Graphics



An upcoming feature is an advanced environment called Arduino compatible Genie Graphics which utilizes the popular ViSi-Genie technology that is available in most 4D Systems Intelligent Displays. It utilizes the genieArduino library, and allows rapid development of GUI due to the powerful tools and widgets Workshop4 provides. Simply lay the display out with the objects you want, set the events to drive them and then utilize the genie functions inside the Arduino sketch to read or write information to those objects from the Atmel processor. This is a very clean and rapid development environment.

In the meantime, you can continue developing Genie Arduino applications using the current tools.

6. Common File Menu

The **File** Menu is the first menu and common to all environments...

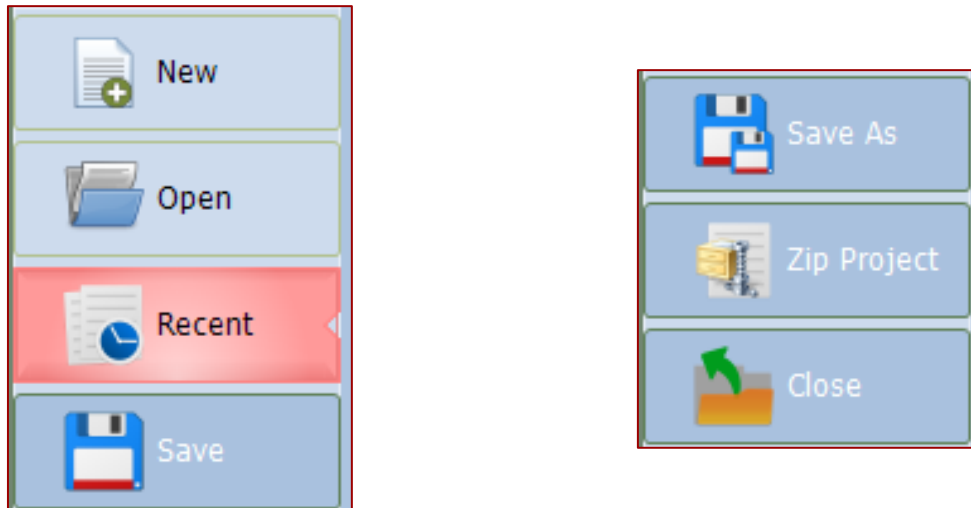


...with various buttons relating to the project that is open (or greyed out if no project is open):

- File-related buttons,
- Print-related buttons,
- And miscellaneous buttons, such as Help, Options and Samples.

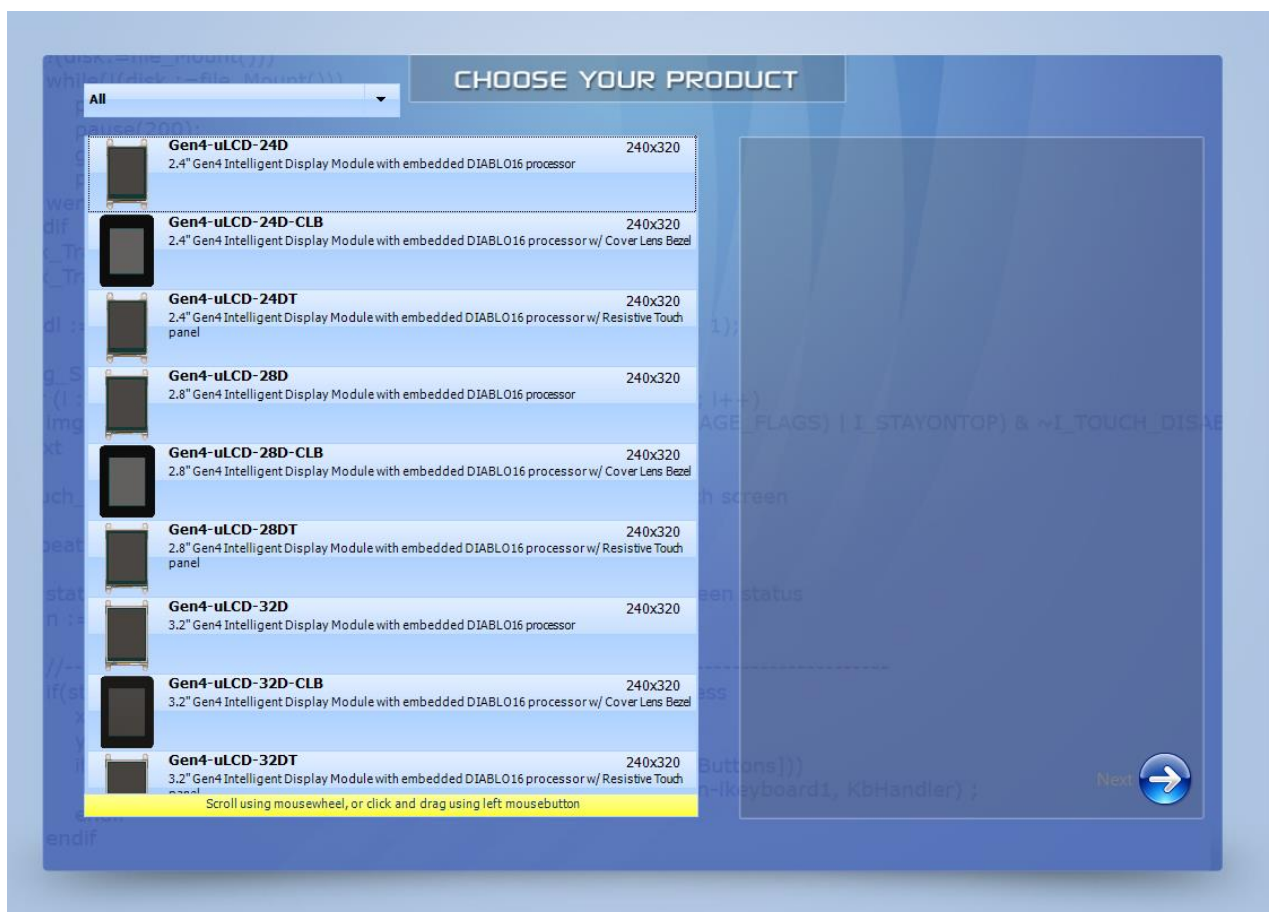
6.1. File-Related Buttons

The buttons include all the actions related to projects: New, Open, Recent, Save, Save As, Zip Project and Close project.

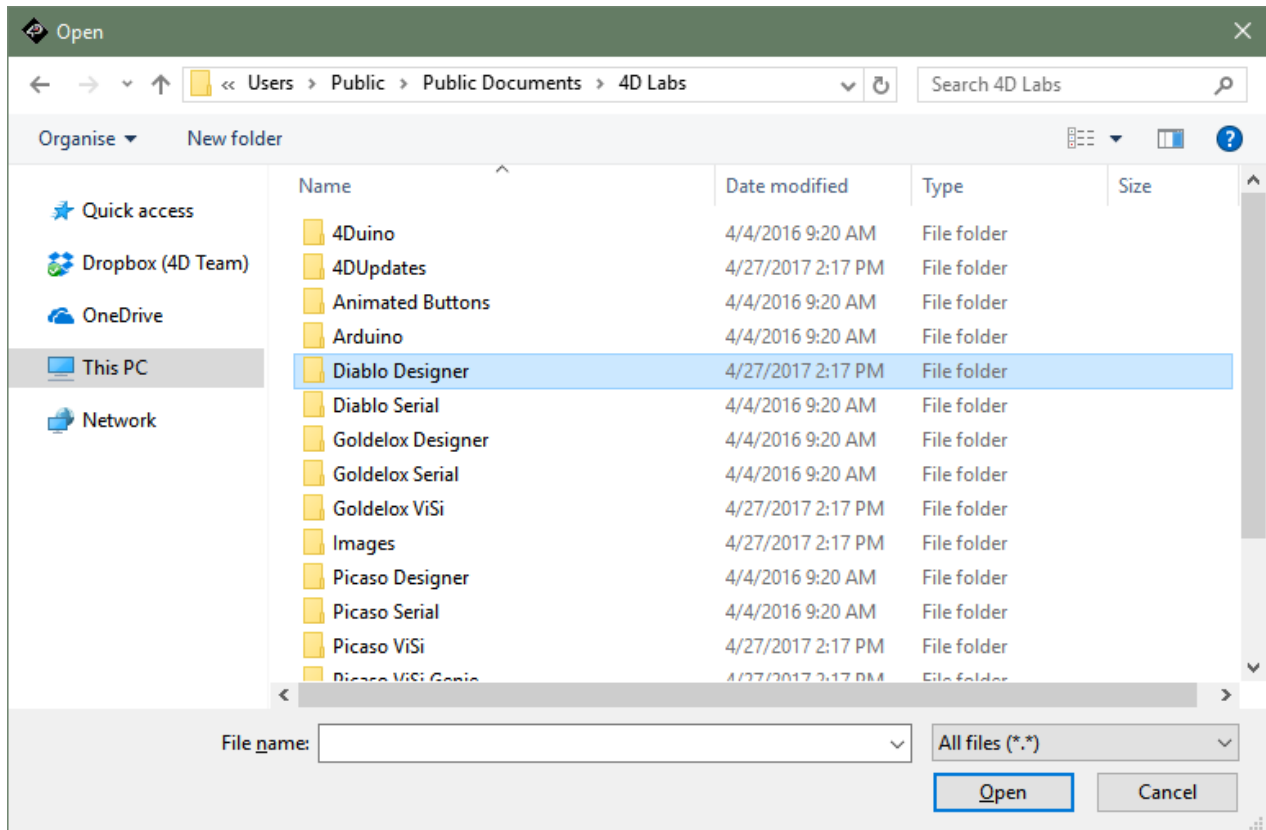


The options include:

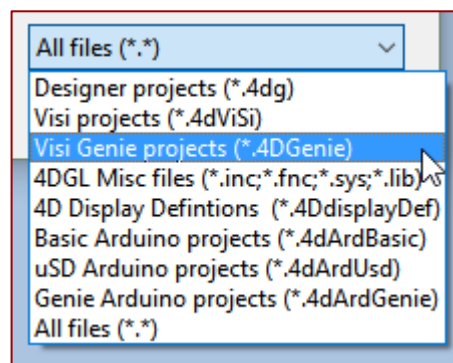
Click **New** to create a new project. A dialogue window asks for the screen and the kind of project.



Click **Open** to browse and open an existing project. A standard Open window opens.



Select **ViSi-Genie projects** among the different kinds listed below to load a ViSi-Genie project:

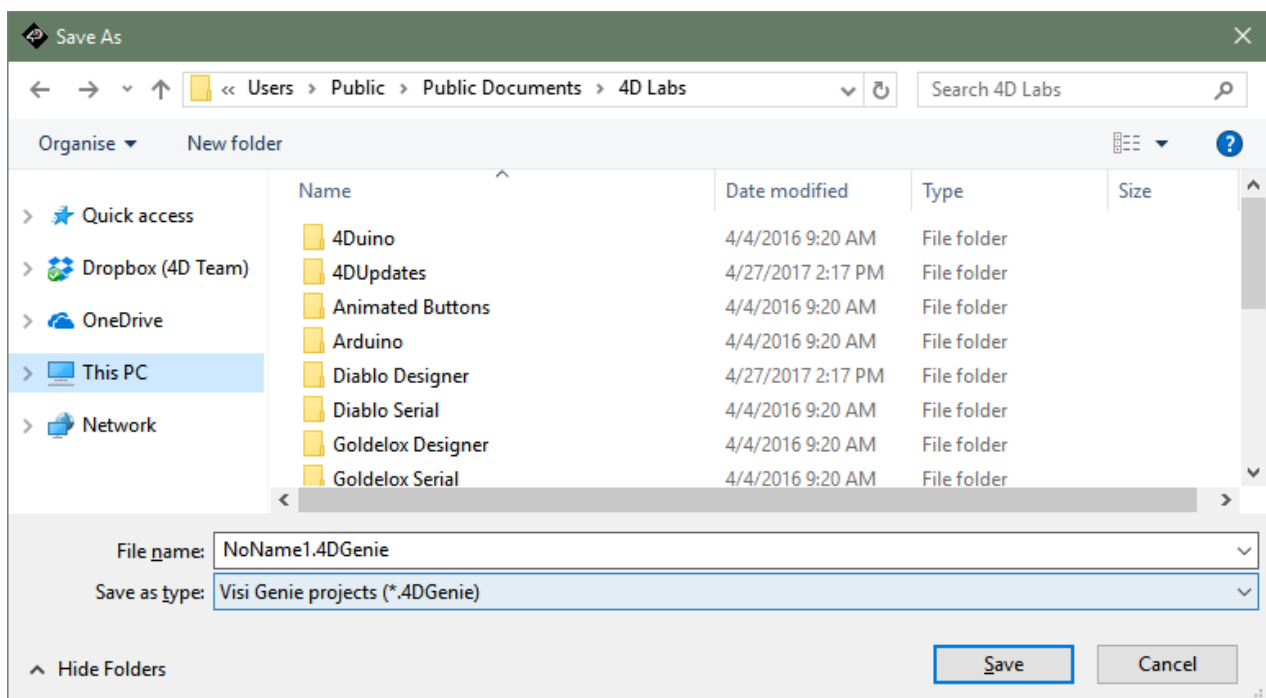


Click **Recent** to list the recently accessed files, and click on the project to open it.



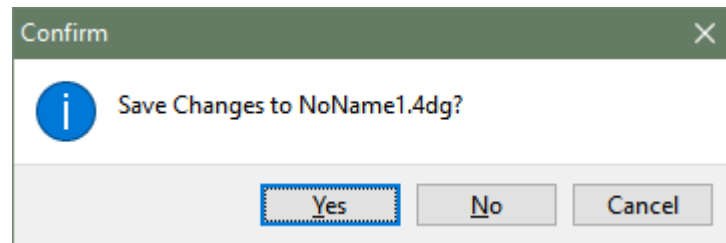
Click **Save** to save the modified projects.

Click **Save As** to create a copy of already saved project and give it a new name. A standard Save window opens and asks for the location and the name:



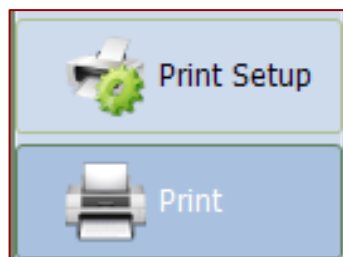
Click **Zip Project** to make a compressed file out of the project. This is especially useful when the project is large and contains pictures and videos.

Click **Close** to close the current project. You will be prompted to save the changes to any modified project.

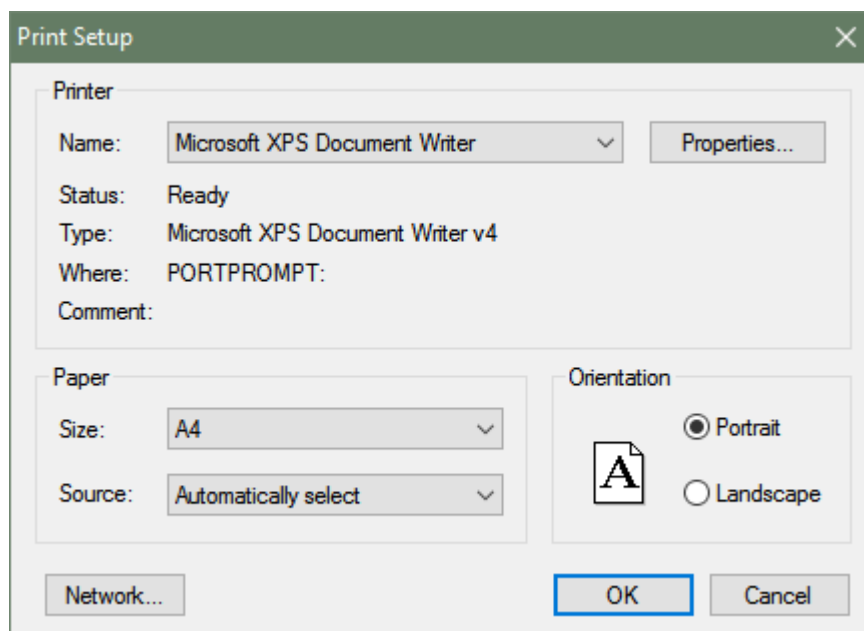


6.2. Print-Related Buttons

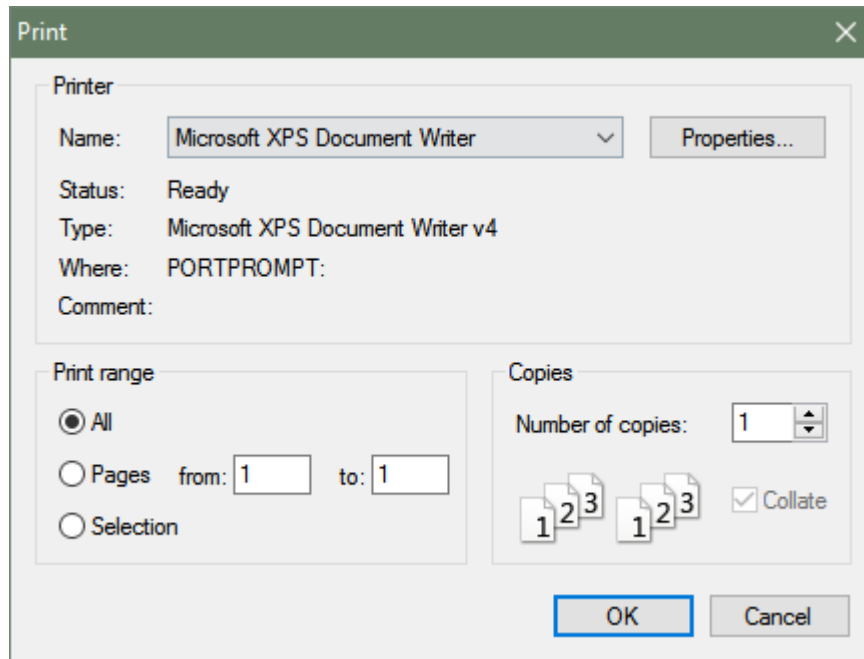
The buttons include all the actions related to print:



Click **Print Setup** to setup the printer:

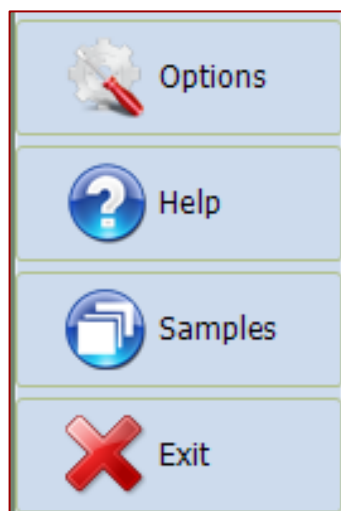


Click **Print** to print the project:

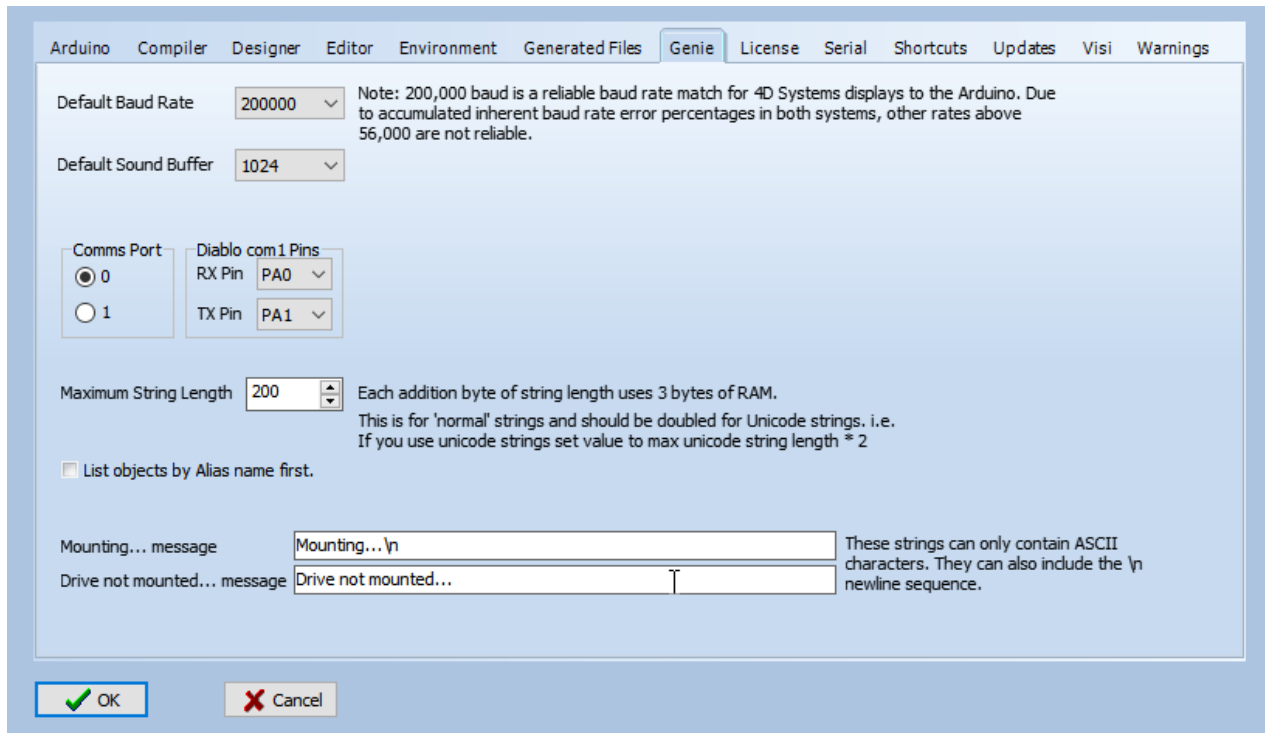


6.3. Miscellaneous Buttons

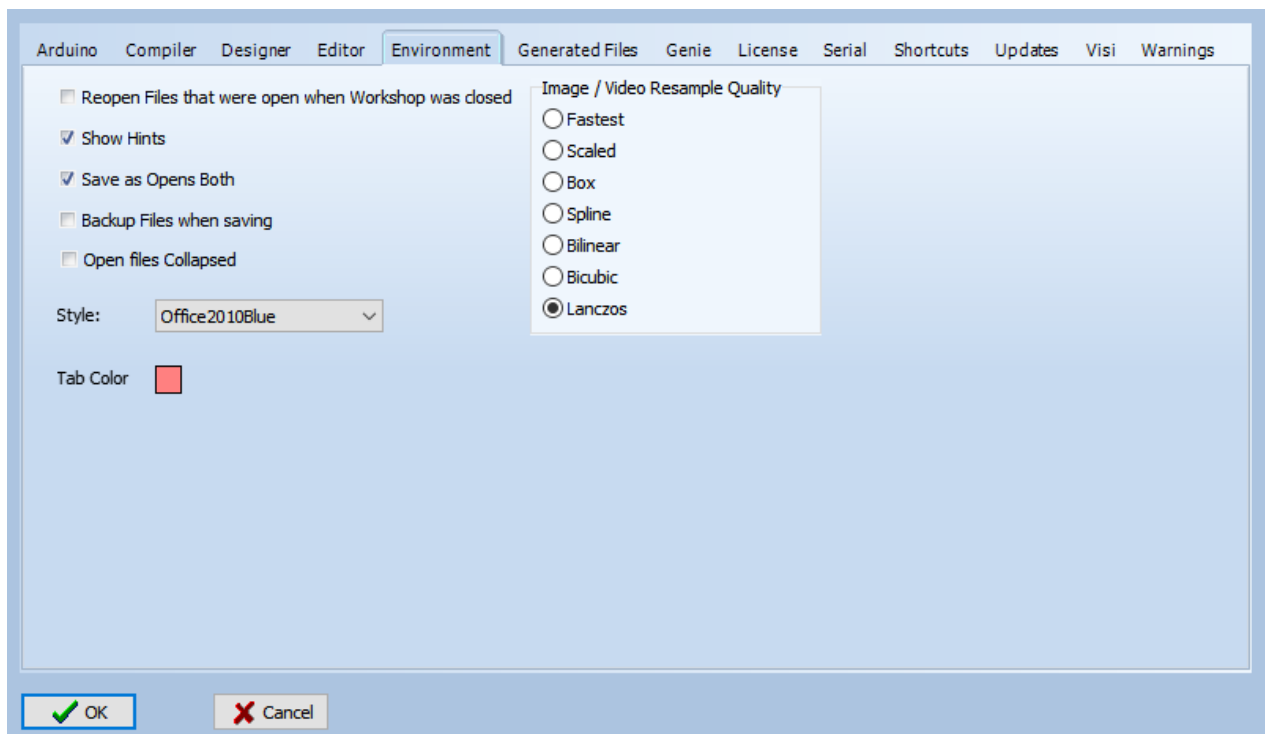
The Miscellaneous buttons include **Options**, **Help**, **Samples** and **Exit**.



Click **Options** to set the options for the current project:



Options include different panes. Select "Environment".

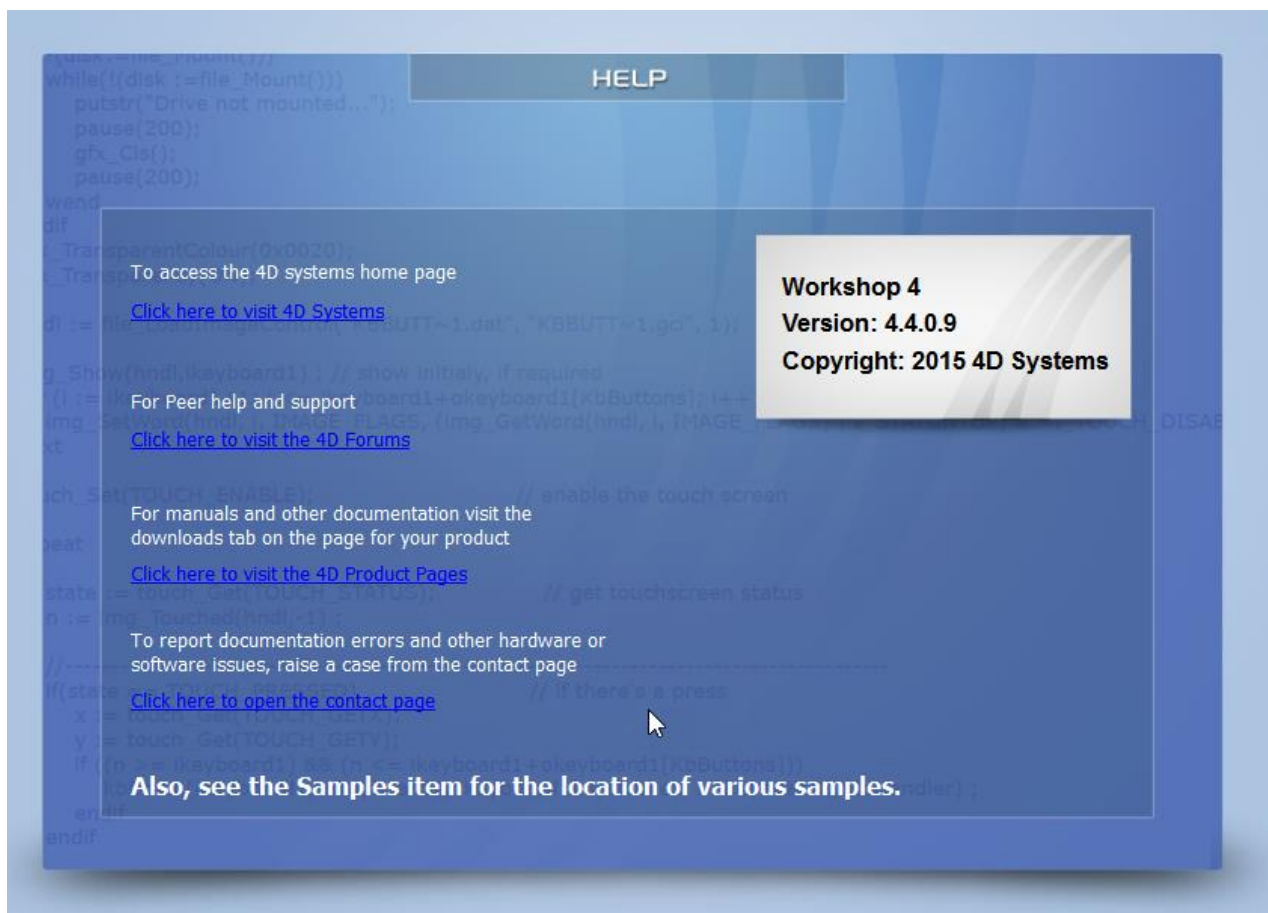


Select the **Style** among different possibilities.

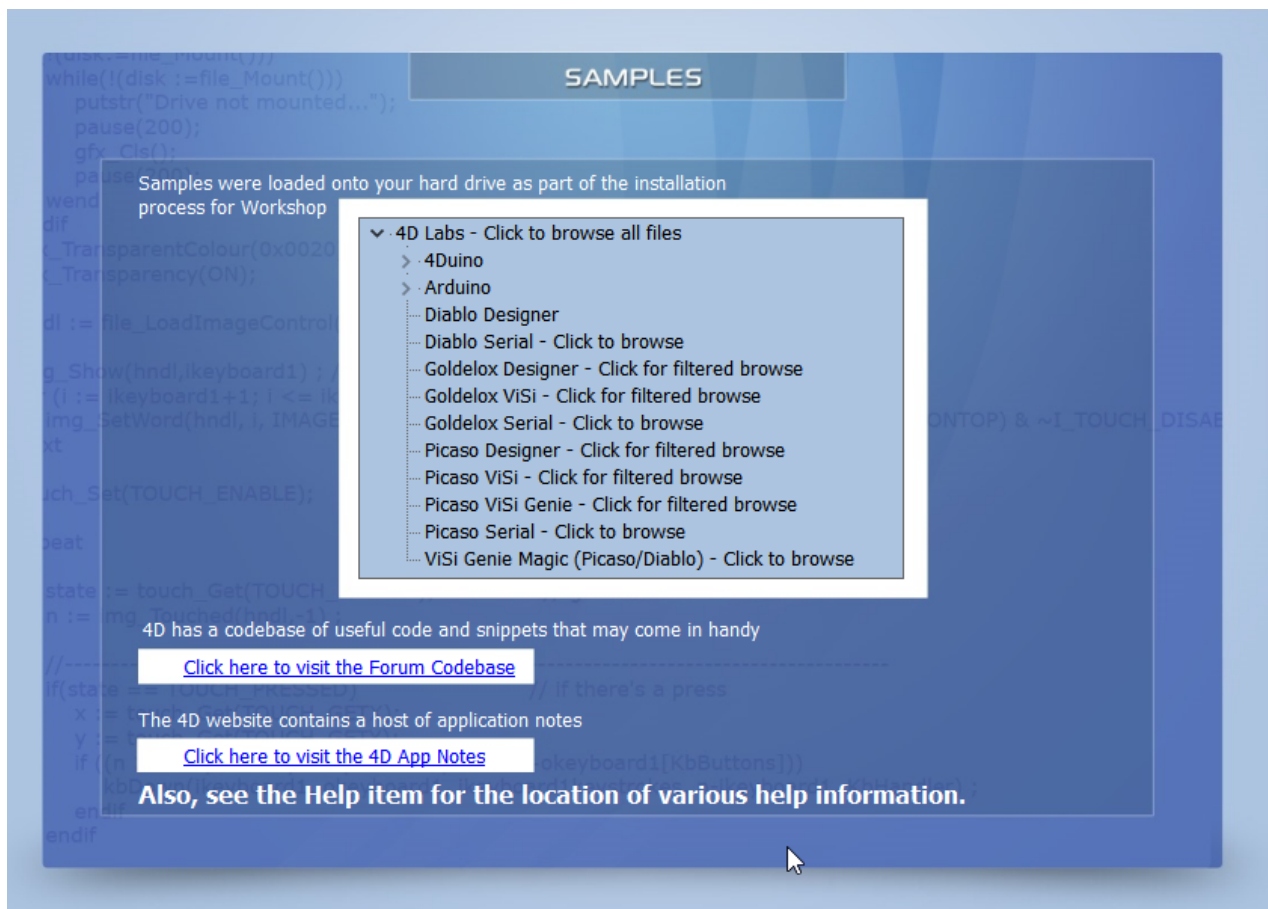
The **Image / Video Resample Quality** selection brings different options:

- Fastest
- Scaled
- Box
- Spline
- Bilinear
- Bicubic
- Lanczos

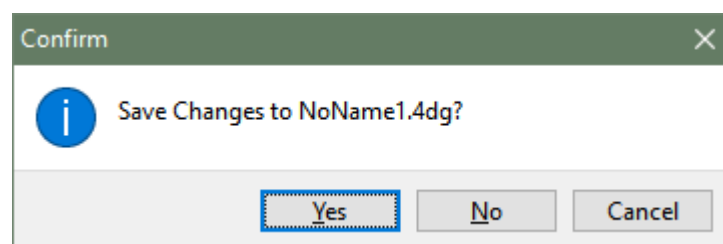
Click **Help** to access help, with links to the 4D website:



Click **Samples** to access examples from 4D, with pre-defined filters:

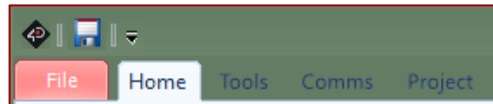


Click **Exit** to close Workshop4. You will be prompted to save the changes to any modified project.



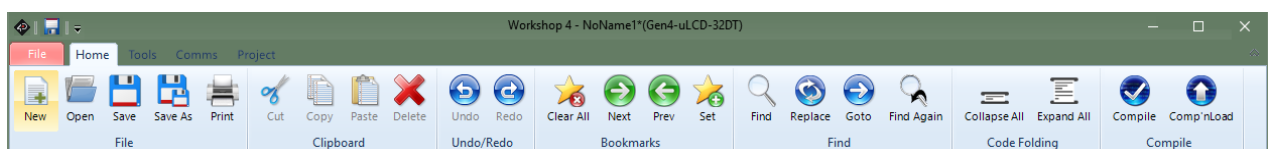
7. Designer Specific Menus

The Designer environment includes five menus:



7.1. Home Menu

The **Home** menu is the main menu.



...with three groups of buttons:

- File-related commands,
- Build command,
- And the objects pane.

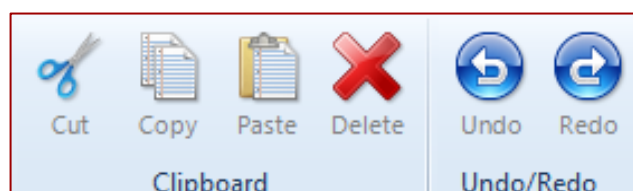
7.1.1. File-Related Buttons

The file-related buttons include the same commands as seen in the File menu: New, Open, Save, Save As and Print.



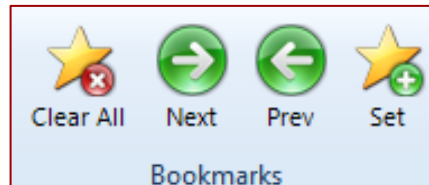
7.1.2. Code-Related Buttons

The code related buttons include the standard Windows commands of Cut, Copy, Paste, Delete, Undo and Redo.



7.1.3. Bookmark Buttons

The bookmark buttons include **Set** a bookmark, go to **Next** or **Previous** bookmark and **Clear All** bookmarks.



Bookmarks are shown close to the line number:



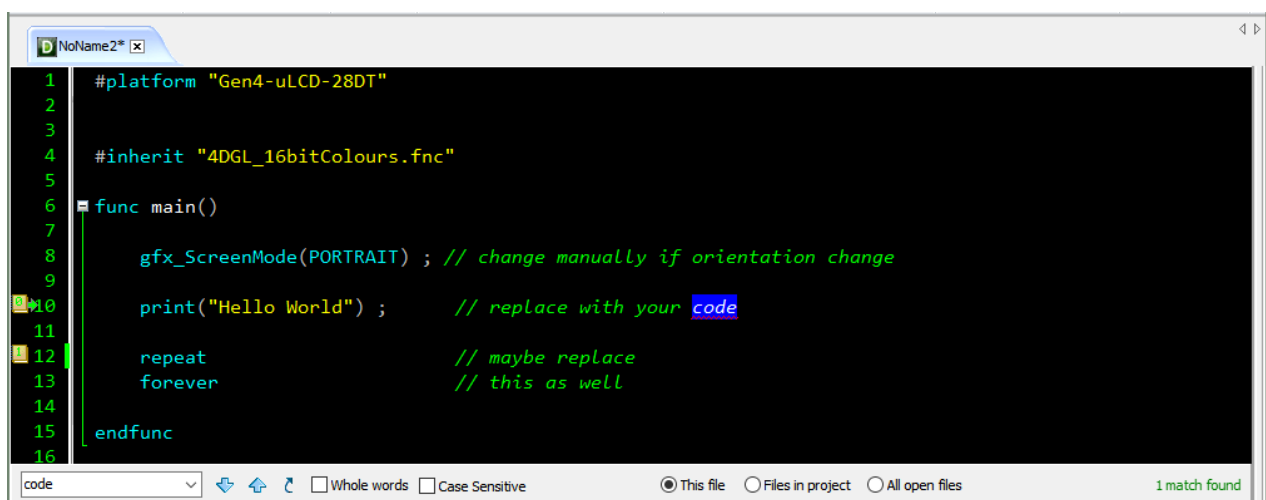
Bookmarks are especially useful for large projects.

7.1.4. Find and Replace Buttons

The find and replace buttons provide the basic features for code.

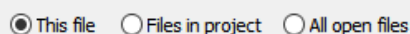


The **Find** button prompts for a string and highlights it in the code:

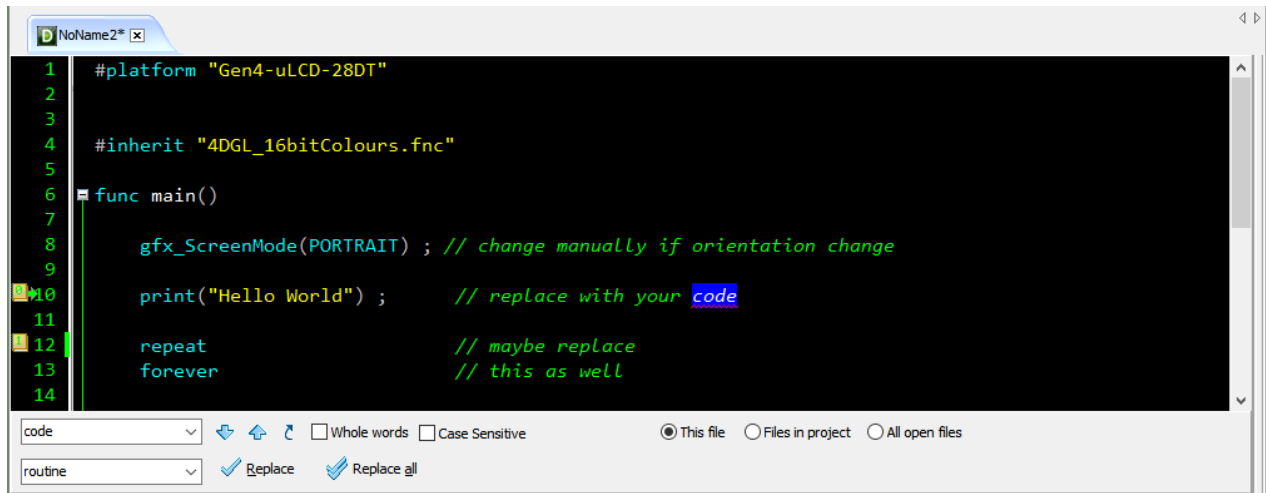


```
1 #platform "Gen4-uLCD-28DT"
2
3
4 #inherit "4DGL_16bitColours.fnc"
5
6 func main()
7
8     gfx_ScreenMode(PORTRAIT) ; // change manually if orientation change
9
10    print("Hello World") ; // replace with your code
11
12    repeat // maybe replace
13    forever // this as well
14
15 endfunc
16
```

Use the up and down arrows to look for the previous and next occurrence. Check **Whole Words** and **Case Sensitive**. Choose between **This file** and **Files in progress** and **All Open Files**.

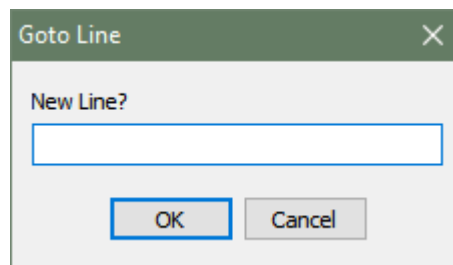


The **Replace** button searches for a string and exchanges it with another string:



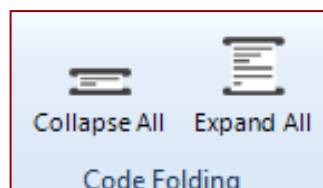
Same options as for Find apply.

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



7.1.5. Code Folding Buttons

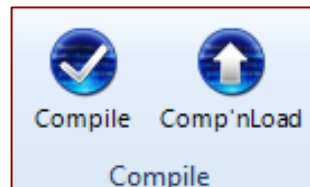
The code folding buttons allow to collapse or expand a function:



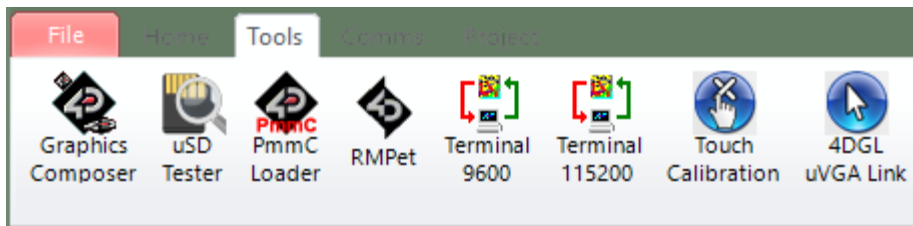
This is especially useful for large projects.

7.1.6. Compile Buttons

The **Compile** button launches the compilation of the project while the **Comp'nLoad** compiles and uploads the project to the screen.



7.2. Tools Menu

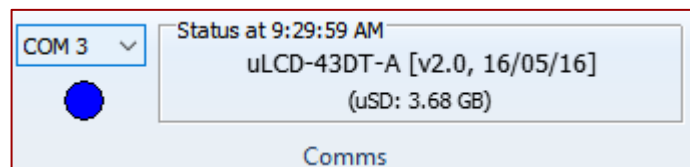


All the tools and utilities are grouped here:

- Click **PmmC Loader** to start the PmmC Loader utility.
- Click '**Terminal connect 9600**' to open the currently selected com port at 9600 baud in the Terminal program.
- Click '**Terminal connect 115200**' to open the currently selected com port at 115200 baud in the Terminal program.
- Click **Touch Calibration** to calibrate the touch on the display. This tool is not available for all display modules.
- Click **4DGL uVGA Link** to open a window which can interact with the uVGA, enabling a mouse and keyboard to be used with the uVGA(GFX) module. Demo code is available from Workshop4.
- Click **Graphics Composer** to open the Graphics Composer tool. This tool is used for creating graphics for a Designer project. If graphics is needed for a Designer project, it is recommended to use ViSi instead.
- Click **uSD Tester** to test the uSD card mounted on the display. Before clicking this button, make sure that the uSD card is mounted on the display module.
- Click **RMPet** to partition the uSD card. The uSD card must be mounted to the PC.

7.3. Comms Menu

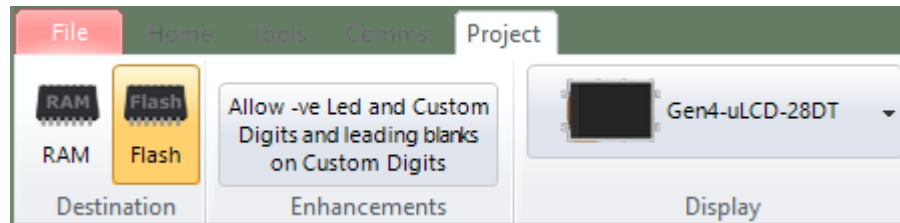
This menu is in charge of the communication port:



The use of this menu is described at the section Connect the Module.

7.4. Project Menu

The **Project** menu includes different parameters and options...

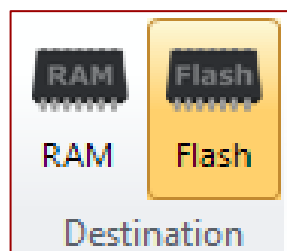


...with three groups of buttons:

- Destination,
- Display selection.
- And Enhancements

7.4.1. Destination

The first group includes the options for the destination:

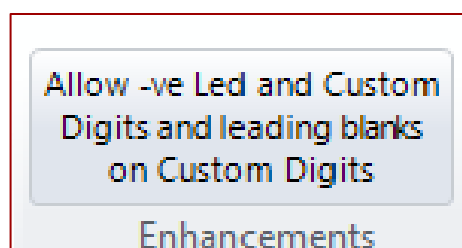


Select the **Destination** among two possibilities:

- **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.
- **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.

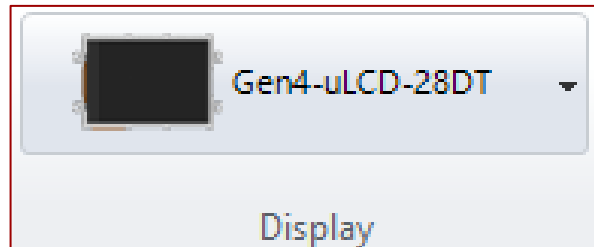
7.4.2. Enhancements

The second section contains a button for enabling the use of negative values for LED digits and custom digits objects and for enabling the use of leading blanks on custom digit objects.

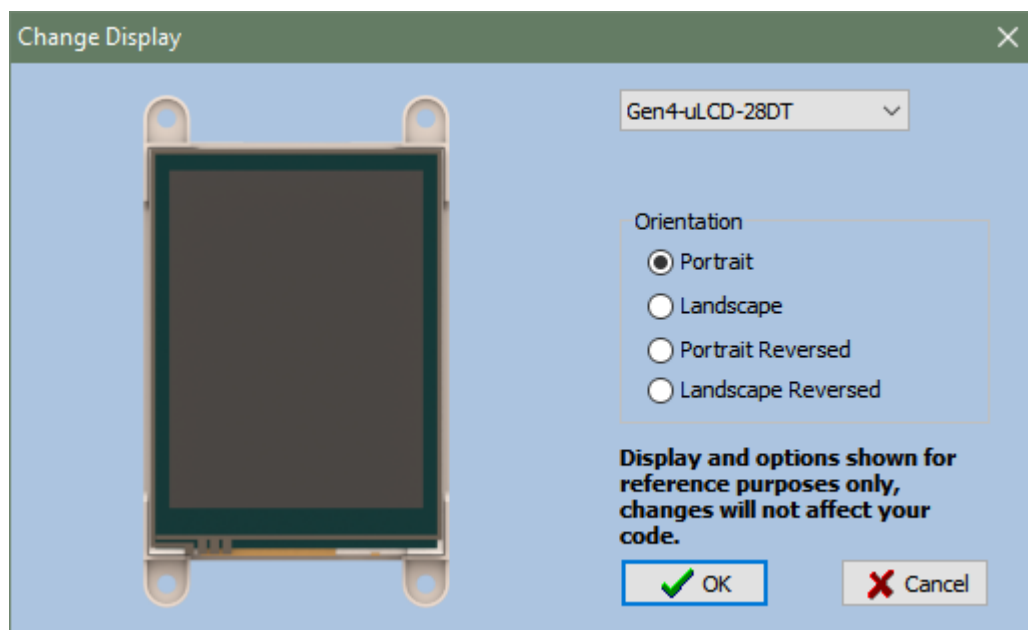


7.4.3. Display Selection

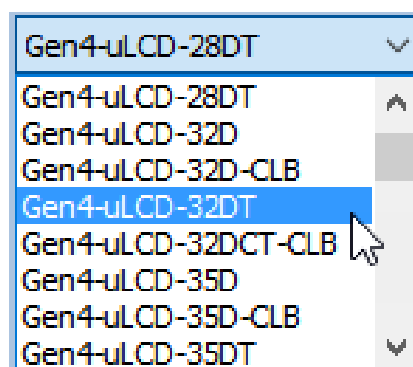
The third section allows to select the screen. Clicking on the button...



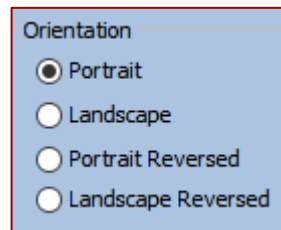
...opens a new window to select the screen:



Select the screen from the drop-down list:



Define the orientation among the four options:

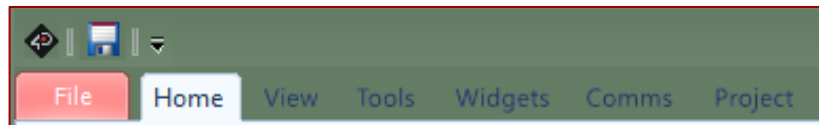


A dialog box titled "Orientation" with a light blue background and a thin red border. It contains four radio button options: "Portrait" (selected), "Landscape", "Portrait Reversed", and "Landscape Reversed".

Confirm by clicking on  or deny by clicking on .

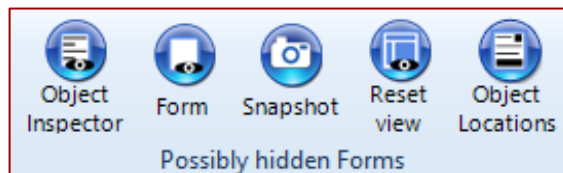
8. ViSi Specific Menus

The ViSi environment includes all the menus available with the Designer environment plus two additional menus: **View** and **Widgets**.

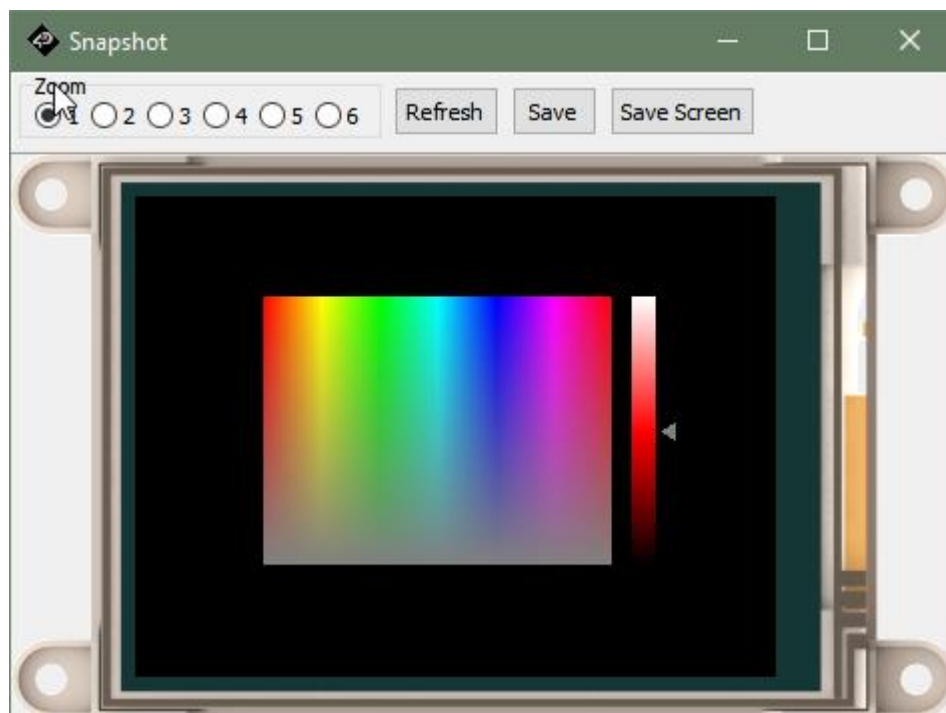


8.1. View Menu

The **View** menu includes one important tool for visualising the form:



Click on **Snapshot** to open a specific window of the form to enable a 1:1 screenshot of the display to me made.



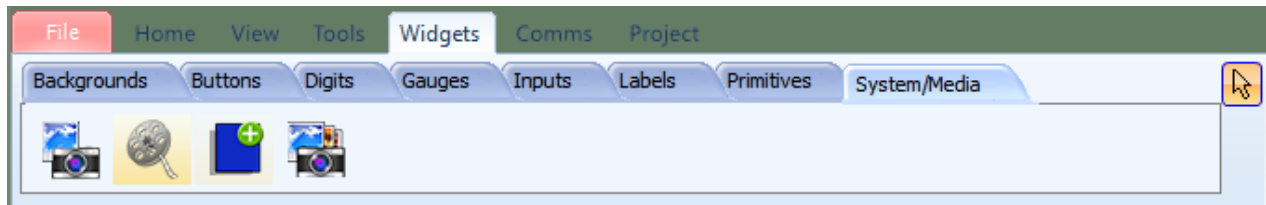
This window provides a zoom up to 4 times. The **Save** button allows to save the screen as an image.

Object Inspector, Form, and Reset View, all relate to the menus and tool bars used in Workshop4. These toolbars and menus can be moved and detached from the side of Workshop4. Object Inspector and Form will bring to front the relevant toolbar when clicked. If required, the toolbars can be reset back to their default location by clicking the Reset View button.

Object Locations enables the user to copy the locations/coordinates of objects on the display, to the clipboard.

8.2. Widgets Menu

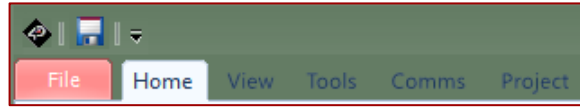
The **Widgets** menu includes the objects pane with all the objects available to build the interface:



The objects –also called widgets–are detailed in the [ViSi User Guide](#).

9. ViSi-Genie Specific Menus

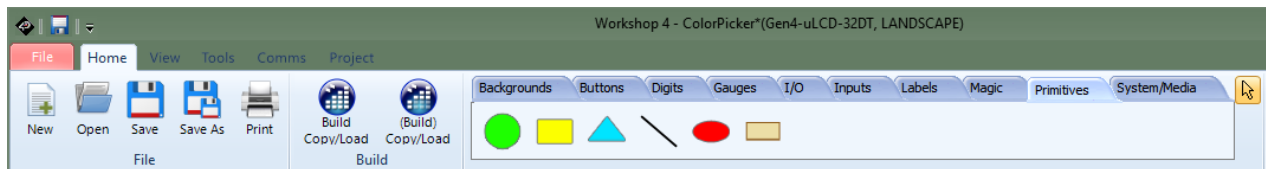
ViSi-Genie includes five menus with specific ribbons and options.



ViSi-Genie is codeless and thus completely different from the previous code-based environments, all the menus relating to ViSi-Genie are detailed.

9.1. Home Menu

The **Home** menu is the main menu.



...with three groups of buttons:

- File-related commands,
- Build command,
- And the objects pane.

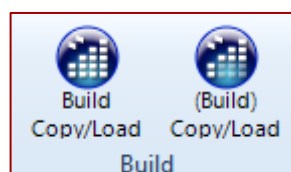
9.1.1. File-Related Buttons

The file-related buttons include the same commands as seen in the File menu: New, Open, Save, Save As and Print.



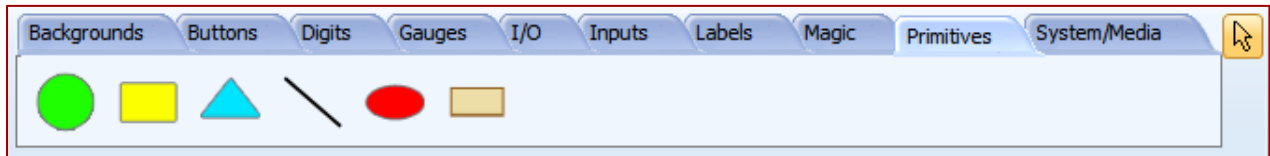
9.1.2. Build Button

The **Build** button launches the compilation and the upload of the project to the screen.



9.1.3. Objects Pane

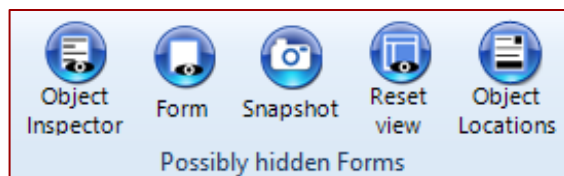
The objects pane includes all the objects available to build the interface:



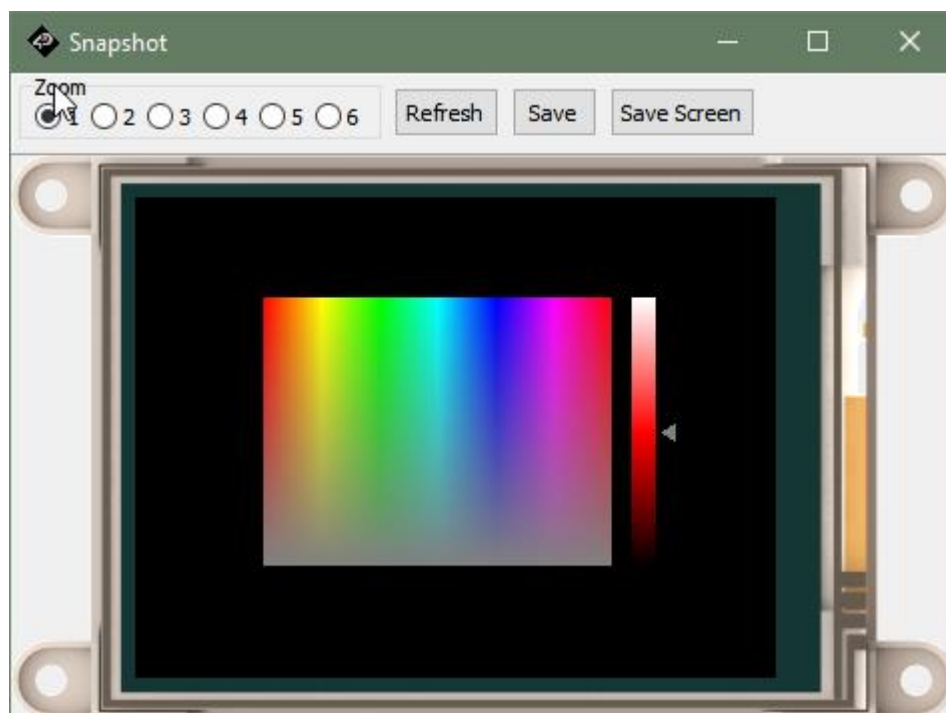
- The objects are detailed on the [VISi-Genie User Guide](#).

9.2. View Menu

The **View** menu includes one important tool for visualising the form:

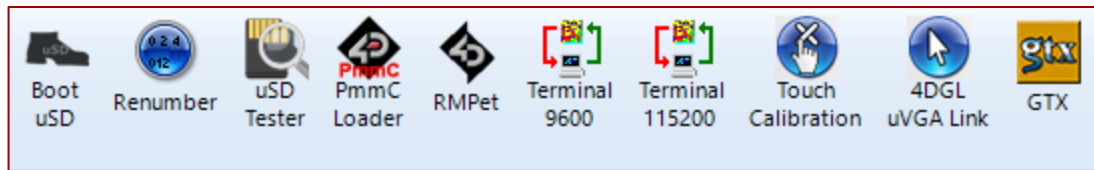


Click on **Snapshot** to open a specific window of the form.



This window provides a zoom up to 6 times. The **Save** button allows to save the screen as an image.

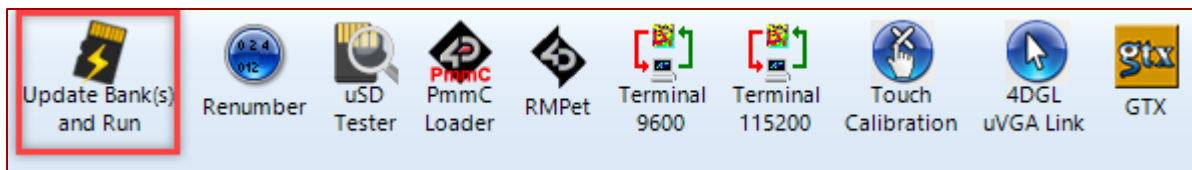
9.3. Tools Menu



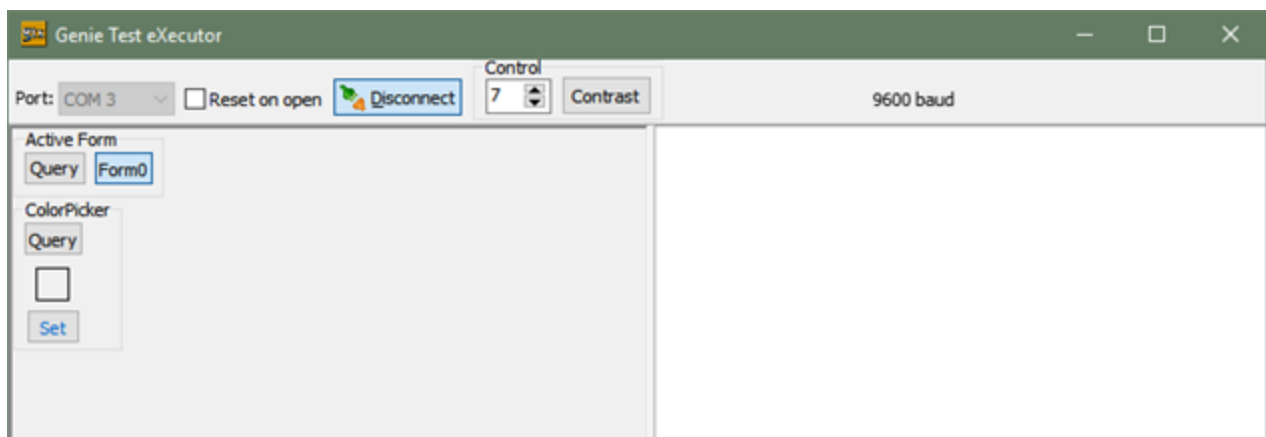
All the tools and utilities are grouped here:

- Click **Boot uSD** to upload the boot-loader to the screen.
- Click **PmmC Loader** to start the PmmC Loader utility.
- Click '**Terminal connect 9600**' to open the currently selected com port at 9600 baud in the Terminal program.
- Click '**Terminal connect 115200**' to open the currently selected com port at 115200 baud in the Terminal program.
- Click **Touch Calibration** to calibrate the touch on the screen. This tool is not available for all display modules.
- Click **4DGL uVGA Link** to open an interactive window to use mouse/keyboard with the uVGA(GFX) module.
- Click **GTX** to launch the Genie Test Executor debugger.

For Diablo16 display modules, the **Boot uSD** icon is replaced with the **Update Bank(s) and Run** icon. For more information on the **Update Bank(s) and Run** option, refer to section [Genie Options](#).

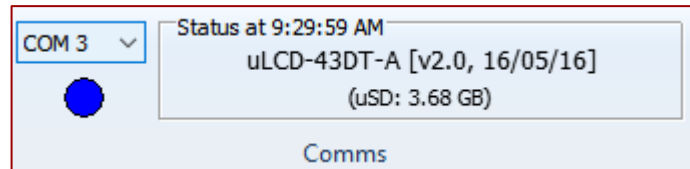


Clicking the GTX button, a new screen will appear with the form and objects defined in the project:



9.4. Comms Menu

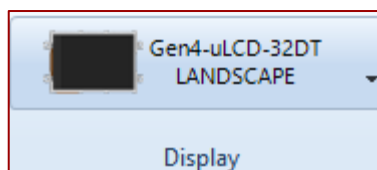
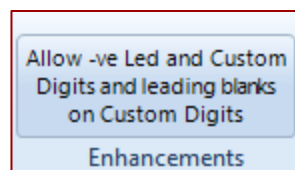
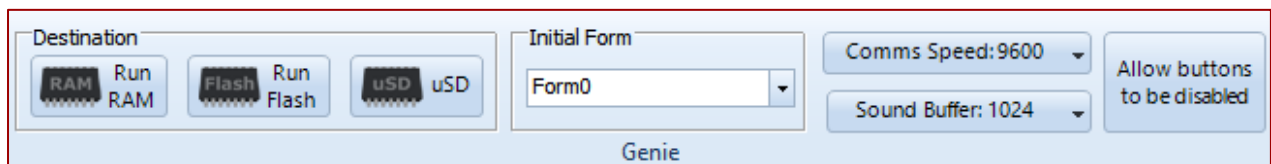
This menu is in charge of the communication port:



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

9.5. Project Menu

The **Project** menu includes different parameters and options...

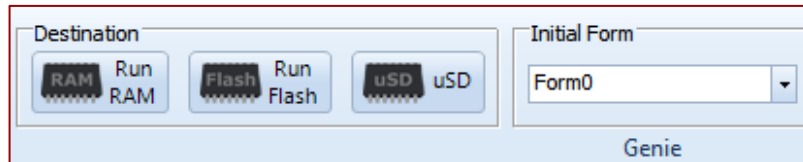


...with three groups of buttons:

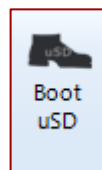
- Options for Genie,
- Parameters and Enhancements
- And display selection.

9.5.1. Genie Options

For Picaso display modules, the first group includes the options for Genie:

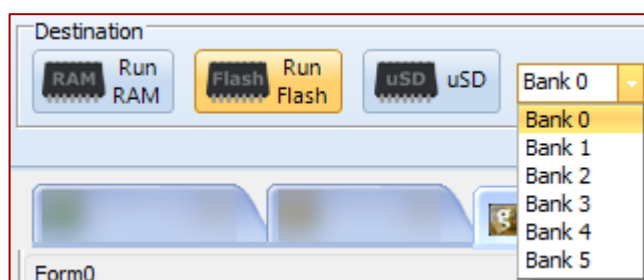


- Select the **Destination** among three possibilities:
 - **Run RAM** - The display must be connected during Build and the program will be downloaded to the display's flash memory once compiled. The user's application will be stored in Flash, but will be run from RAM.
 - **Run Flash** - The display must be connected during Build and the program will be downloaded to the display's flash memory once compiled. The user's application will be stored and run from flash, this uses less memory on the display, but makes programs run slightly slower.
 - **uSD** - The user's application will be built and copied to the uSD card. From the uSD card the application is loaded into RAM and run from there. This option requires the Boot uSD program to be uploaded to the display's flash, as seen in the menu **Tools**. This Boot uSD program loads the user's application from the uSD card at startup, and executes it.



- The initial form section allows the user to set which form will show upon boot up.

For Diablo16 display modules, the first group looks like as shown below.

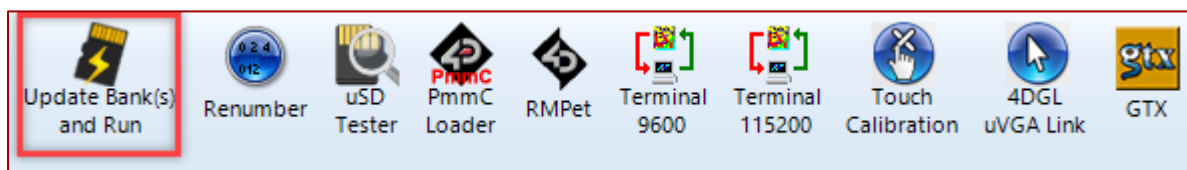


The additional drop-down menu allows the user to specify the target destination flash bank of the ViSi-Genie program. The Diablo16 processor has six flash banks (Bank 0 to Bank 5), each of which has a capacity of 32 kB.

When **Run Flash** is selected, the destination of the ViSi-Genie program is the bank specified in the drop-down menu. In this case, the display module needs to be connected to the PC during build time. The program will then be downloaded to the selected bank and it will run from there. Take note however, that, after the display module is power cycled, the program in Bank 0 always runs first.

On the other hand, when **uSD** is selected, the ViSi-Genie program will be copied to the uSD card. This option does not require the display module to be connected to the PC during build time. However, this option requires the **Update**

Bank(s) and Run program to be downloaded to Bank 0 of the display module's flash memory. The **Update Bank(s) and Run** program button is found under the Tools menu.



This **Update Bank(s) and Run** program checks the uSD card for ViSi-Genie program files and copies them to their destination flash banks. After this check-and-copy process, the program in Bank 1 is executed by default. The **Update Bank(s) and Run** program can be modified to run another bank besides Bank 1 if desired after the check-and-copy process.

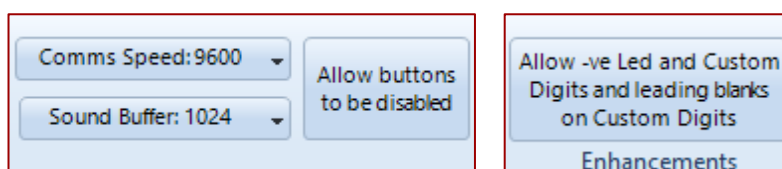
Note that **Update Bank(s) and Run** program stores the time and date information of ViSi-Genie program files (for all banks) in Bank 0. Every time that the display module is power cycled, **Update Bank(s) and Run** in Bank 0 always runs first and checks the time and date information of the ViSi-Genie program files present in the uSD card. By default, if the time and date information of a ViSi-Genie program file in the uSD card is different from that of the last program file uploaded to the specified bank, **Update Bank(s) and Run** automatically updates the bank.

It is also possible to modify the **Update Bank(s) and Run** program such that it updates the target banks only if the corresponding ViSi-Genie program files in the uSD card have a newer time and date information.

For more information, refer to relevant application notes found on the application notes [page](#).

9.5.2. Parameters

The second group includes advanced parameters:

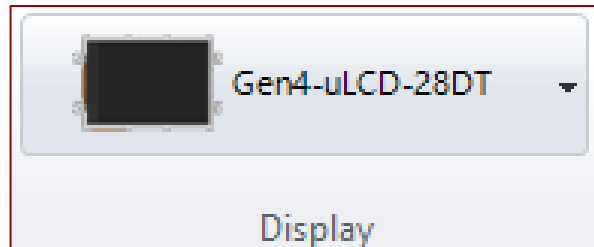


- **Comms speed** is the baud rate at which the serial command interface operates
- Define **Sound buffer size** for the amount of memory set aside for buffering wav (sound) files. For simple sound files 1024 bytes should be enough. For complicate sound files to be played whilst video is displaying may need as much as 4096 bytes.
- Button objects have the ability to be shown and hidden accordingly by the host controller. To enable this, click on the **“Allow buttons to be disabled”** button. Then use the GTX tool to see the appropriate commands for enabling and disabling the buttons.
- The right-most button allows for enabling the use of negative values for LED digits and custom digits objects and for enabling the use of leading blanks on custom digit objects.

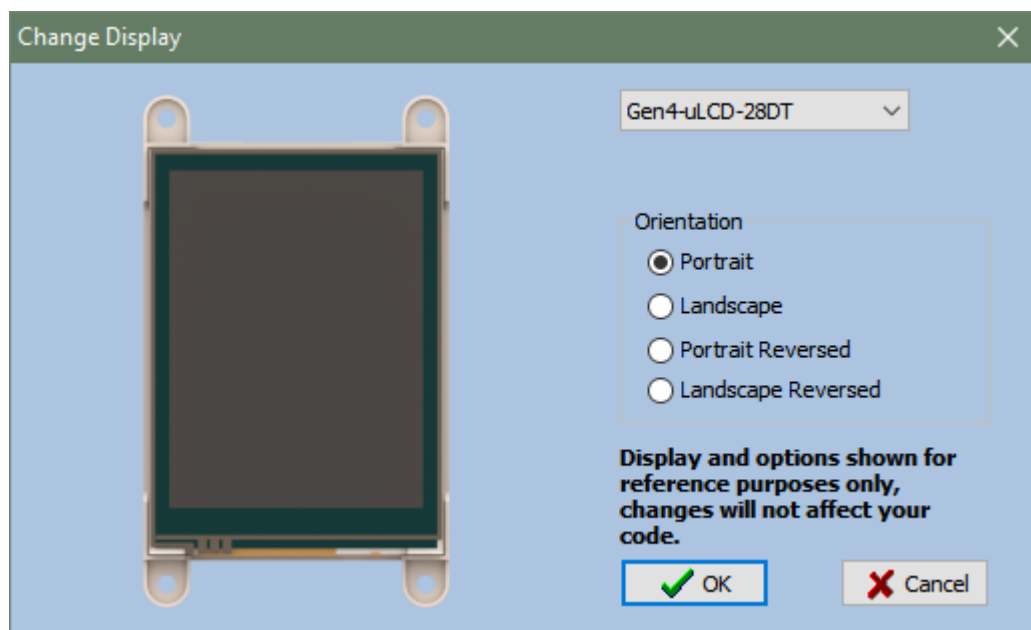
9.5.3. Display Selection

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

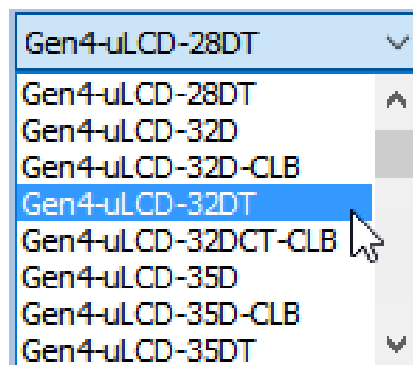
Clicking on the button...



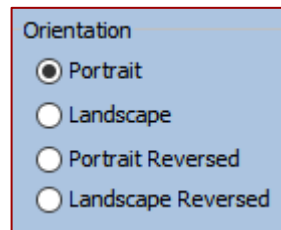
...opens a new window to select the screen:



Select the screen from the drop-down list:



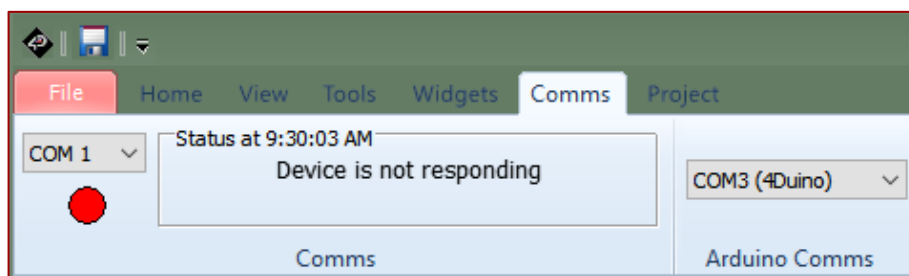
Define the orientation among the four options:



Confirm by clicking on  or deny by clicking on .

10. Basic and Extended Graphics Specific Menus

The Arduino compatible Basic and Extended Graphics environments include all the menus available with the ViSi Environment with some additional options for **Arduino Comms** which can be found under **Comms** tab.



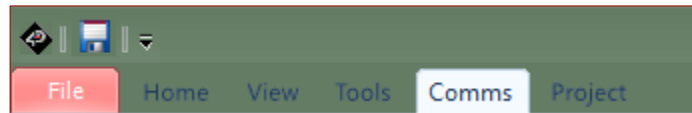
The Arduino Comms refers to the COM port that the Arduino board is currently using.

The main difference between the Basic and Extended Arduino environments lies on the available widgets.

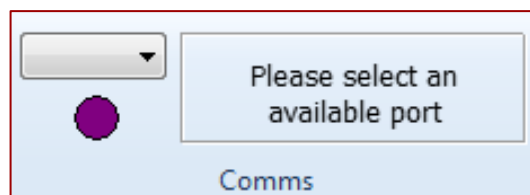
Since the Basic Environment is designed to allow users to create projects without the need for a uSD card, it only allows the user to use primitive shapes and objects in the WYSIWYG window. The Extended Graphics on the other hand gives additional support for 4D Graphics.

11. Connect the Module

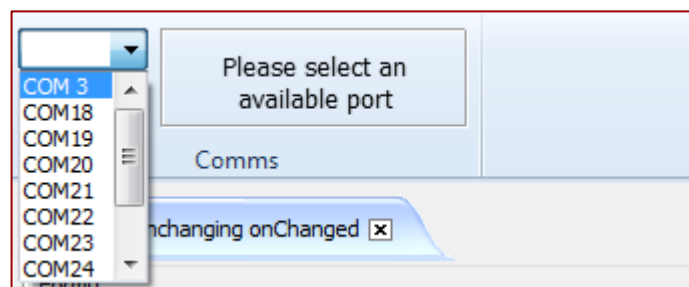
Connect the module to a USB port with the 4D Systems programming cable and select the **Comms** menu:



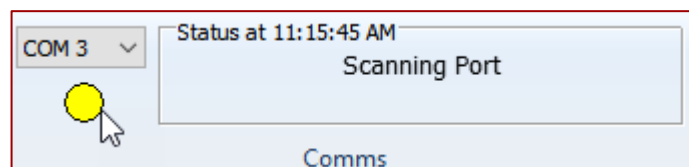
Above the **Comms** section, the **violet** light mentions no module is currently connected.



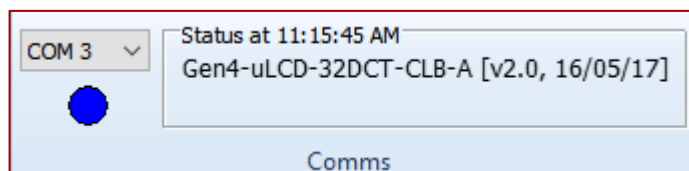
Connect the 4D Systems programming cable/adaptor to the module and plug the cable into the USB port. Click on the drop-down list and select the COM port relating to the 4D Programming cable/adaptor.



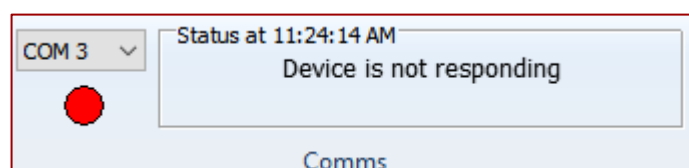
The light turns **yellow** while the connection is being established:



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



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

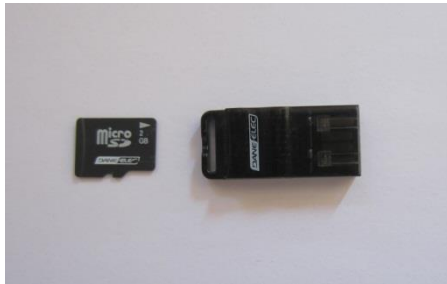


12. Insert the Micro-SD Card

For Picaso, Picaso Lite, and Diablo16, the micro-SD card shall be FAT16-formatted. Partition can't exceed 4 GB. For Goldelox, the micro-SD card shall not be formatted at all, it requires the SD card to be RAW.

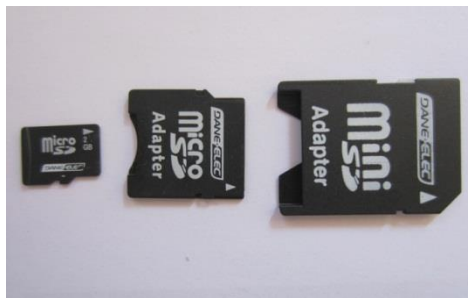
To connect the micro-SD card, either:

- Insert the micro-SD card into the USB adaptor and plug the USB adaptor into an USB port of the PC.

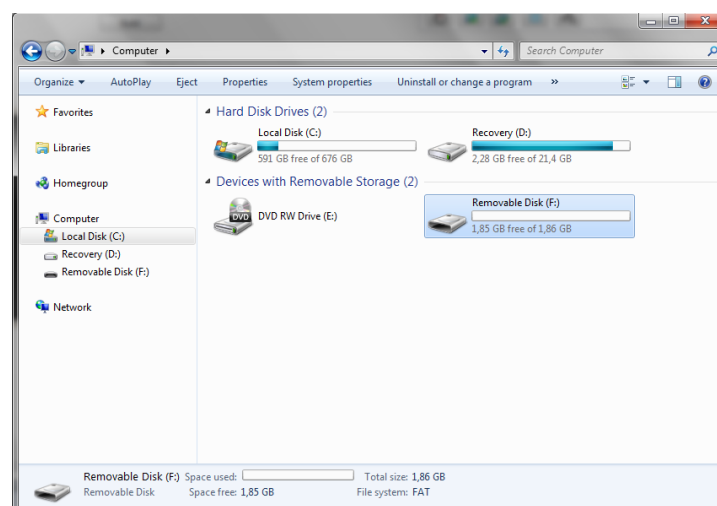


Or

- Insert the micro-SD card into a micro-SD to SD card converter and plug the SD card converter into the SD card slot of the PC.



Check the micro-SD card is mounted, here as drive E:.



For Goldelox, if prompted to format the SD card, click no/cancel. Leave the card unformatted and Workshop4 will handle the rest.

13. Revision History

Revision	Revision Content	Revision Date
1.0	First Release	Nov 19, 2012
1.1	Typos on Page 4 fixed	Dec 17, 2012
1.2	Added new content for Serial and fixed incorrect document references	Feb 4, 2013
1.3	Amended details about Micro-SD card	July 5, 2013
1.4	Amended details about Program Destination	Mar 11, 2014
2.0	Updated formatting and contents	May 1, 2017
2.1	Added information on target flash banks	July 29, 2017
S 2.1	Updated hyperlinks, 4D Systems	July 29, 2017

14. Legal Notice

Proprietary Information

The information contained in this document is the property of 4D Labs Semiconductors and may be the subject of patents pending or granted, and must not be copied or disclosed without prior written permission.

4D Labs Semiconductors 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 Labs Semiconductors products and services is continuous and published information may not be up to date. It is important to check the current position with 4D Labs Semiconductors. 4D Labs Semiconductors 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.

Disclaimer of Warranties & Limitation of Liability

4D Labs Semiconductors 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 Labs Semiconductors range of products, however the quality may vary.

In no event shall 4D Labs Semiconductors 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 Labs Semiconductors, or the use or inability to use the same, even if 4D Labs Semiconductors has been advised of the possibility of such damages.

4D Labs Semiconductors 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 Labs Semiconductors and its suppliers specifically disclaim any expressed or implied warranty of fitness for High Risk Activities.

Use of 4D Labs Semiconductors' 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 Labs Semiconductors from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any 4D Labs Semiconductors intellectual property rights.

15. Contact Information

For Technical Support: www.4dsystems.com.au/contact

For Sales Support: sales@4dsystems.com.au

Website: www.4dsystems.com.au

Copyright 4D Labs Semiconductors 2000-2017.