QUICK START GUIDE

4D SYSTEMS

TURNING TECHNOLOGY INTO ART

PICASO Modules

Quick Start Guide

Document Date: 17th May 2013
Document Revision: 1.0

QUICK START GUIDE

Contents

1. Introduction

This Quick Start Guide is an introduction to becoming familiar with PICASO driven Display Modules and the development tools associated with them. This guide should be treated only as a useful starting point and not as a comprehensive reference document. The primary aim of this guide is to quickly and effectively teach the essentials to setting up and developing an application on any one of the 4D PICASO display modules. Once the basics are mastered, developing more advanced and involved applications will flow much easier. **Section 7** lists a full range of detailed reference documents, which will be required during various stages of development.

The PICASO belongs to a family of processors powered by a highly optimized soft core virtual engine; **EVE** (Extensible Virtual Engine). EVE is a proprietary, high performance virtual processor, with an extensive byte-code instruction set optimised to execute compiled 4DGL programs. **4DGL** (4D Graphics Language) was specifically developed from ground up for the EVE engine core. It is a high level language, which is easy to learn and simple to understand, yet powerful enough to deliver many embedded graphics applications.

There are 6 different display modules based on the PICASO processor in 4D's product range besides their derivatives. Such as a uLCD-24PTU is based on PICASO processor whereas a uLCD-24PTU-AR is its derivative which includes a set of items (Display Module and Arduino Shield Adaptor) ready to connect the display module to an Arduino Module with the help of available libraries.

uLCD-28PTU

2.8" Intelligent LCD module w/ Touch

uLCD-32PTU

3.2" Intelligent LCD module w/ Touch

uLCD-32WPTU

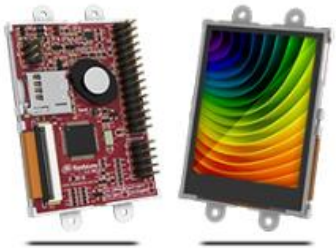
3.2" Wide Intelligent LCD module w/ Touch

uVGA-III

VGA Graphics Controllers for QVGA, VGA, WVGA

uLCD-43 (P/PT/PCT)

microLCD 4.3" PICASO LCD-TFT Display Module (optional Touch)



uLCD-24PTU

2.4" Intelligent LCD module w/ Touch

The list of features that each of the display module has to offer is given under the Generic Features table.

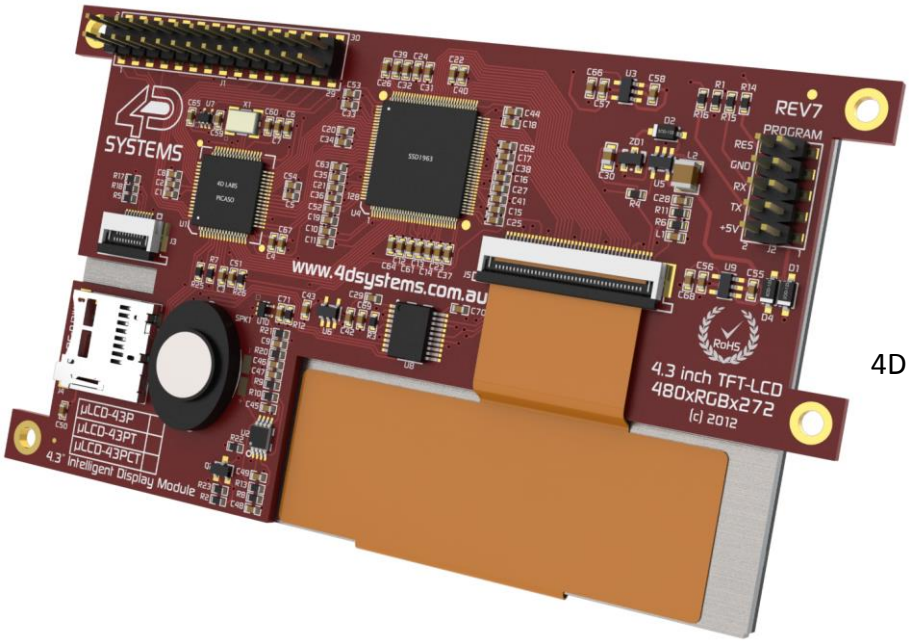
Note: Complete details on individual products are available from the relevant product page on our website.

2. Generic Features

Feature	µLCD-24PTU	µLCD-28PTU	µLCD-32PTU	µLCD-32WPTU	µLCD-43P/PT/PCT	µVGA-III
2.4'' LCD TFT display	X					
2.8'' LCD TFT display		X				
3.2'' LCD TFT display			X			
3.2''W LCD TFT display				X		
4.3'' LCD TFT display					X	
RGB 65K true to life colors	X	X	X	X	X	X
240 x RGB x 320 Resolution	X	X	X			X
240 x RGB x 400 Resolution				X		
480 x RGB x 272 Resolution					X	
320 x RGB x 240 |640 x RGB x 480|800 x RGB x 480 Resolutions						X
Integrated 4-Wire Resistive Touch Panel	X	X	X	X	X	
Optional Non-Touch or Capacitive Touch Panel					X	
5 pin interface to any host device: VCC, TX, RX, GND, and RESET	X	X	X	X	X	X
PICASO processor	X	X	X	X	X	X
14KB of Flash memory for user code	X	X	X	X	X	X
14KB of SRAM for user variables	X	X	X	X	X	X
Two Asynchronous hardware serial ports (COM0, COM1), TTL interface, with 300 baud to 600K baud	X	X	X	X	X	X
One I2C interface (Master)	X	X	X	X	X	X
Eight 16 bit timers with 1.0ms resolution	X	X	X	X	X	X
13 General Purpose I/O pins. Upper 8 bits can be used as an I/O Bus for fast 8-bit parallel data transfers	X	X	X	X	X	X
micro-SD memory card adaptor for multimedia storage and data logging purposes. HC memory card support is also available for cards larger than 4GB.	X	X	X	X	X	X
DOS compatible file access (FAT16 format), as well as low level access to card memory	X	X	X	X	X	X
Dedicated PWM Audio pin supports FAT16 audio WAV files and complex sound generation	X	X	X	X	X	X
On-board audio amplifier with a 8Ω speaker for sound generation and WAV file playback	X	X	X	X	X	
Built in extensive 4DGL graphics and system library functions	X	X	X	X	X	X
Display full color images, animations, icons and video clips	X	X	X	X	X	X
Supports all available Windows fonts and characters	X	X	X	X	X	X
One 2 x 15 pin header for I/O expansion	X	X	X	X	X	X
15 pin D-type VGA cable to interface to any external VGA monitor						X
4.0V to 5.5V range operation (single supply).	X	X	X	X	X	X
4 x mounting tabs with 3mm holes for mechanical support	X	X	X	X	X	
4 x 3mm diameter holes within the PCB for mechanical support						X
RoHS Compliant	X	X	X	X	X	X

3. Connecting the Display Module with the PC

The PICASO display modules could only be programmed through 4D's Programming adaptors, such as 4D Programming Cable or uUSB-PA5. Please note, any third party USB to Serial (TTL) converter can damage the display module, using it for programming also voids the warranty.



4D Programming Cable

uUSB-PA5

4D Programming Cable

Powering a PICASO display module is as simple as connecting either a 4D Programming Cable, or uUSB-PA5 to a PC. Please note, the USB ports of the PCs are usually limited to 500mA.
The driver for the Programming adaptor is usually auto detected and installed on the Windows PC. If it isn't installed automatically, user might have to download the driver, from the DOWNLOADs section on the following links, and install it explicitly.

- uUSB-PA5
- 4D Programming Cable

4. Development Environments

Any PICASO display module can be programmed and used in 4 distinct Development Environments categorised in to Standalone or Slave configurations.

- **Standalone Configuration**
  - Designer Environment
  - ViSi Environment

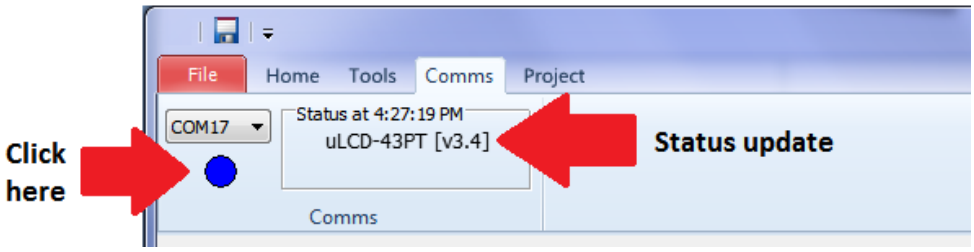
- **Slave Configuration**
  - ViSi Genie Environment
  - Serial Environment

4D Workshop4 IDE is comprehensive software IDE for Microsoft Windows that provides an integrated software development platform with these development environments, for the user to choose based on application requirements or even user skill level.

4.1 Designer

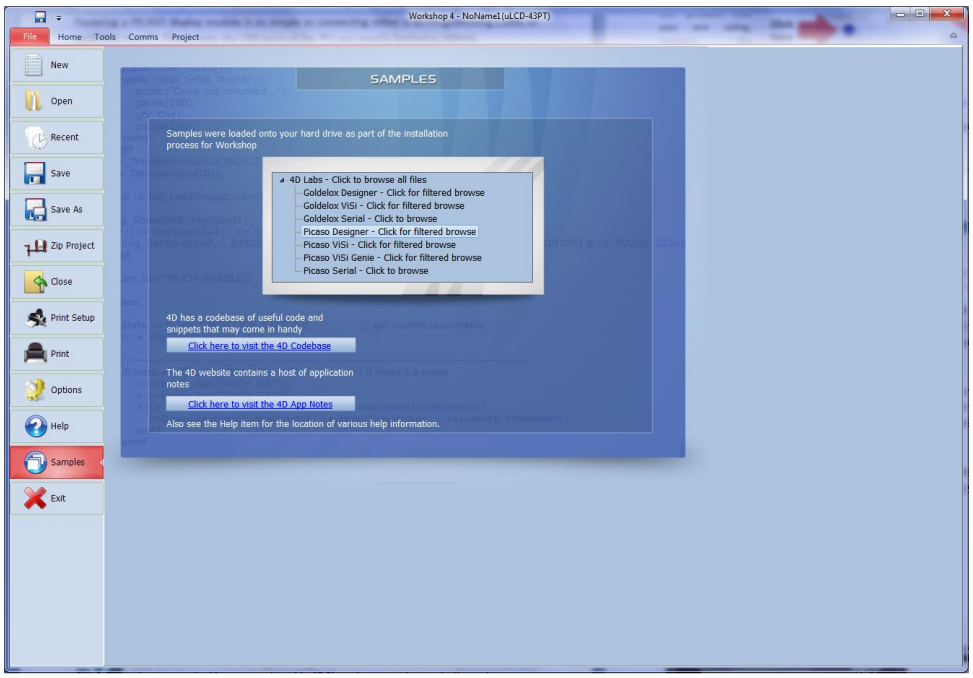
This environment enables the user to write 4DGL code in its natural form to program the display module. The user gets full control of all the available resources on the module such as, GPIOs, I2C and 2xUART ports. The display module can communicate with external devices as a standalone device.

To get started with this environment, you need to make sure that R34 PmmC file or above is installed on the display module. You could check it by connecting the display module with the PC, start a new project, select the product and the Designer Environment, connect the module with the PC through the 4D Programming adaptor, select the com port and click on to Status light under the **Comms** menu, you will see a status update as shown in this image.



4.1.1 Testing a Designer Sample Program

You can find the sample programs for the Designer category from **Samples** under File menu on the 4D Workshop4 IDE. Click on to Picaso Designer section and browse for the WORM.4DG file here,
...\4D Labs\Picaso Designer\PICASO - GRAPHICS



Now, Go to Project menu, select your product ID and set the Destination to Flash or Ram. Go back to Home, click Compile and make sure there are no errors. Click Download to load the program on to the module. You should see a program running on the screen.

Note: Some of the Designer programs require uSD card for proper execution.

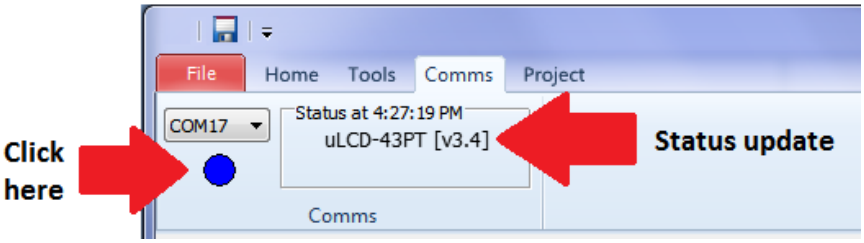
All the reference documentation that is required to proceed with 4DGL programming on a PICASO display module is available under the Designer category, under the DOWNLOADs tab, on the 4D_Wotkshop4_IDE product page.

You can also find a number of application notes on our website based on PICASO Designer category to introduce you to different aspects of developing an application in the PICASO Designer Environment.

4.2 ViSi

ViSi provides a visual programming experience that 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. The user gets full control of all the available resources on the module such as, GPIOs, I2C and 2xUART ports. The display module can communicate with external devices as a standalone device.

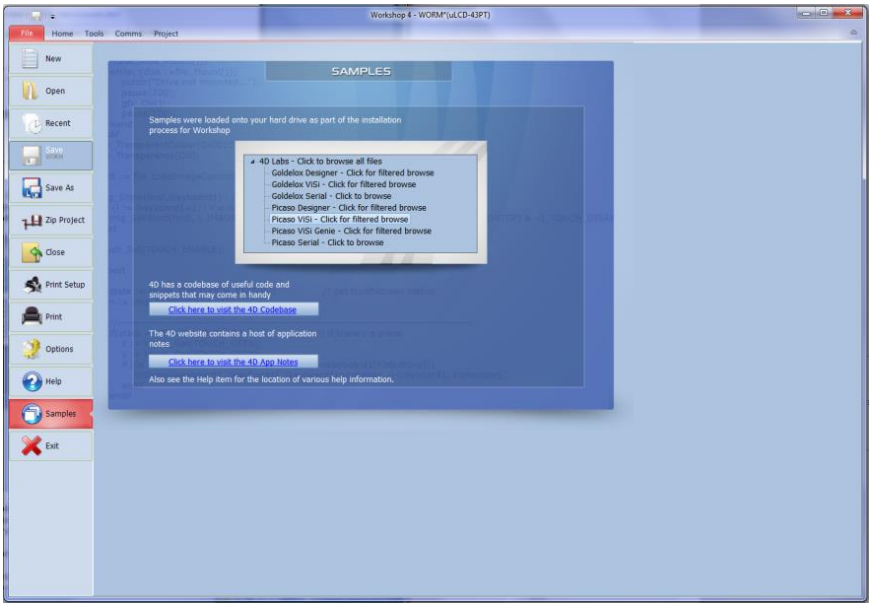
To get started with this environment, you need to make sure that R34 PmmC file or above is installed on the display module. You could check it by connecting the display module with the PC, start a new project, select the product and the ViSi Environment, connect the module with the PC through the 4D Programming adaptor, select the com port and click on to Status light under the **Comms** menu, you will see a status update as shown in this image.



ViSi program requires a FAT (aka FAT16 formatted) uSD card to store the resources. After dragging and dropping the objects and building screen layout, user can simply click “Paste code” to add the necessary code to the command area. The code can be easily modified to handle the properties and events associated with the objects.

4.2.1 Testing a ViSi Sample Program

You can find the sample programs for the ViSi category from **Samples** under File menu on the 4D Workshop4 IDE. Click on to Picaso ViSi section and browse for the FANCYBUTTONS.4DViSi file here,
... \4D Labs\Picaso Visi



Insert a uSD card in to the PC, format it with FAT(aka FAT16) using Windows formatting tool.

Now, Go to Project menu in the 4D Workshop4 IDE, select your product ID and set the Destination to Flash or Ram. Go back to Home, click Compile and you will be prompted to copy some files to the uSD card. Select the Drive and press OK. Check the compiler log at the bottom and make sure there are no errors. Click Download to load the program on to the module. You will see the display module displaying "Drive not mounted..." until you plug the uSD card in to the display module. Once you have inserted the uSD card, you should see the program running on the display screen.

ViSi also requires 4DGL programming. All the reference documentation that is required to proceed with 4DGL programming on a PICASO display module is available under the ViSi category, under the DOWNLOADs tab, on the 4D Wotkshop4 IDE product page

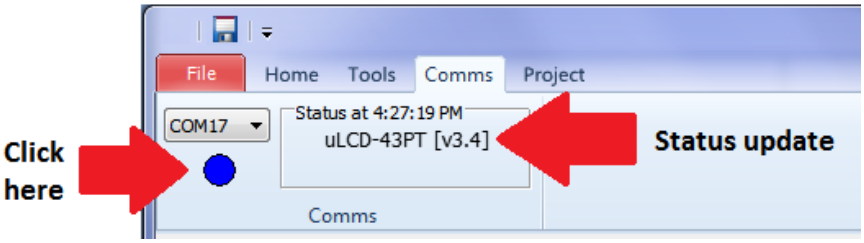
You can also find a number of application notes on our website based on PICASO ViSi category to introduce you to different aspects of developing an application in the PICASO ViSi Environment.

Note: You may wish to use CLOCK.4DViSi file here **... \4D Labs\Picaso Visi** for the uVGA-III module as it doesn't support Touch.

4.3 ViSi-Genie

An advanced environment that doesn't require any 4DGL coding, it is all done automatically for you. Simply lay the display out with the objects you want (similar to ViSi), set the events to drive them and the code is written for you automatically. ViSi -Genie provides the latest rapid development experience from 4D Systems. The user cannot access the GPIOs and I2C port. There is only one UART port that is connected to the Host.

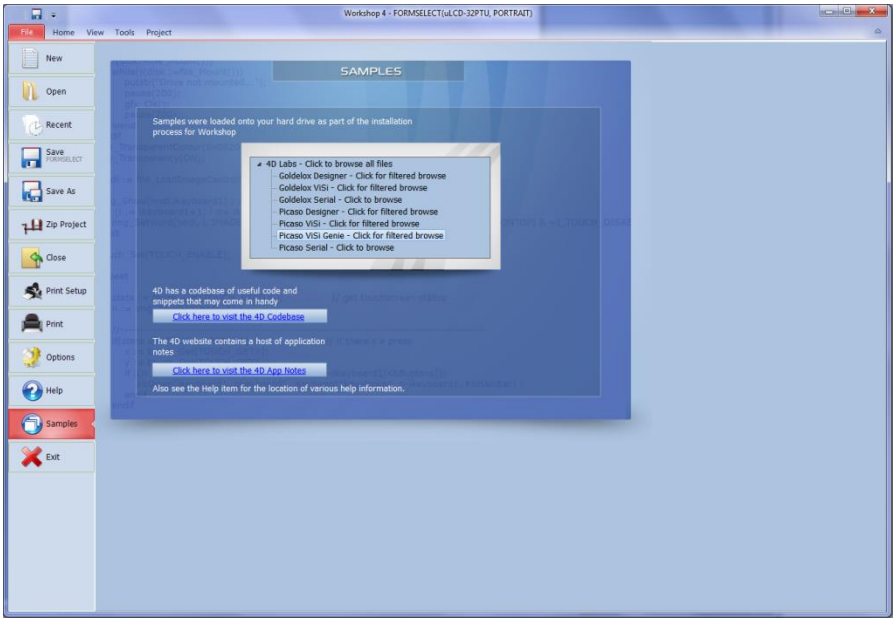
To get started with this environment, you need to make sure that R34 PmmC file or above is installed on the display module. You could check it by connecting the display module with the PC, start a new project, select the product and the ViSi-Genie Environment, connect the module with the PC through the 4D Programming adaptor, select the com port and click on to Status light under the **Comms** menu, you will see a status update as shown in this image.

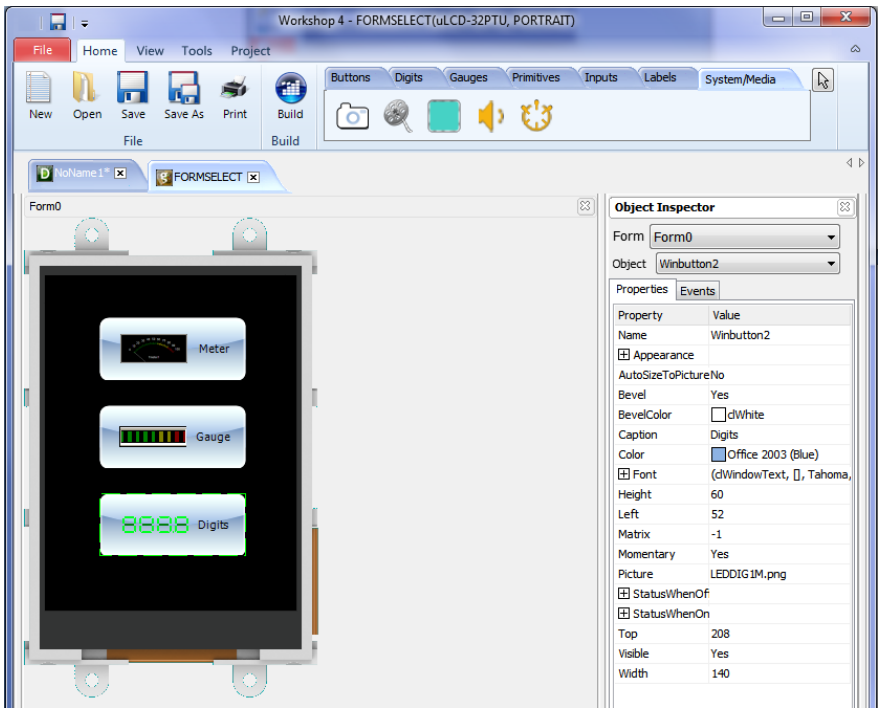


ViSi-Genie program requires a FAT (aka FAT16) formatted uSD card to store the resources. The Display module in this environment can trigger events on its own or accept commands from the host controller.

4.3.1 Testing a ViSi-Genie Sample Program

You can find the sample programs for the ViSi-Genie category from **Samples** under File menu on the 4D Workshop4 IDE. Click on to Picaso ViSi-Genie section and browse for the FORMSELECT.4DGenie file here,**... \4D Labs\PICASO Visi Genie**



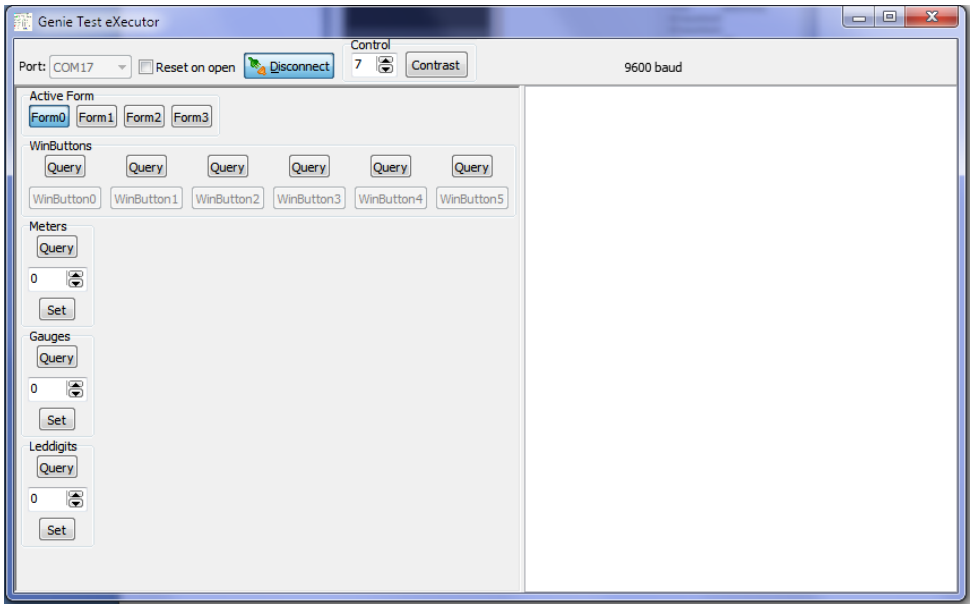


Insert a uSD card in to the PC, format it with FAT(aka FAT16) using Windows formatting tool.

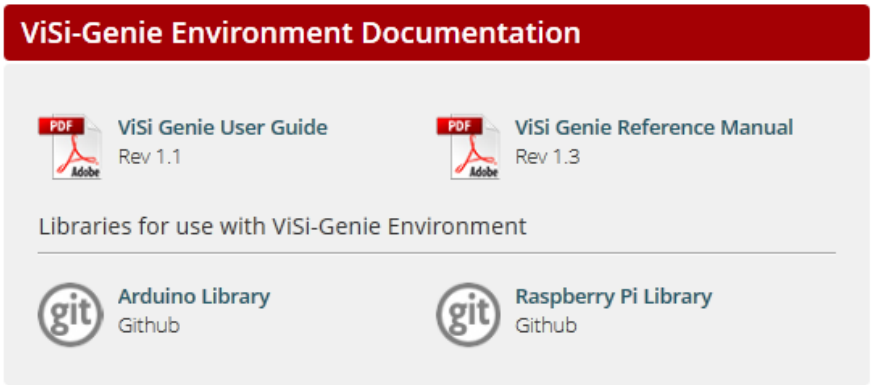
Now, Go to Project menu in the 4D Workshop4 IDE, select your product ID and set the Destination to Flash or Ram. Go back to Home, click Compile and you will be prompted to copy some files to the uSD card. Select the Drive and press OK. Check the compiler log at the bottom and make sure there are no errors. Click Download to load the program on to the module. You will see the display module displaying “Drive not mounted...” until you plug the uSD card in to the display module. Once you have inserted the uSD card, you should see the program running on the display screen.

Note: Select 640x480 or above resolution for the uVGA-III module to run the FORMSELECT.4DGenie correctly. Please also note, uVGA-III doesn’t support Touch.

Now, to simulate the host controller, run the software tool under **Tools** menu called GTX.



All the reference documentation and ViSi-Genie libraries required to build a ViSi-Genie application on a PICASO display module are available under the ViSi Genie category, under the DOWNLOADs tab, on the 4D Wotkshop4 IDE product page

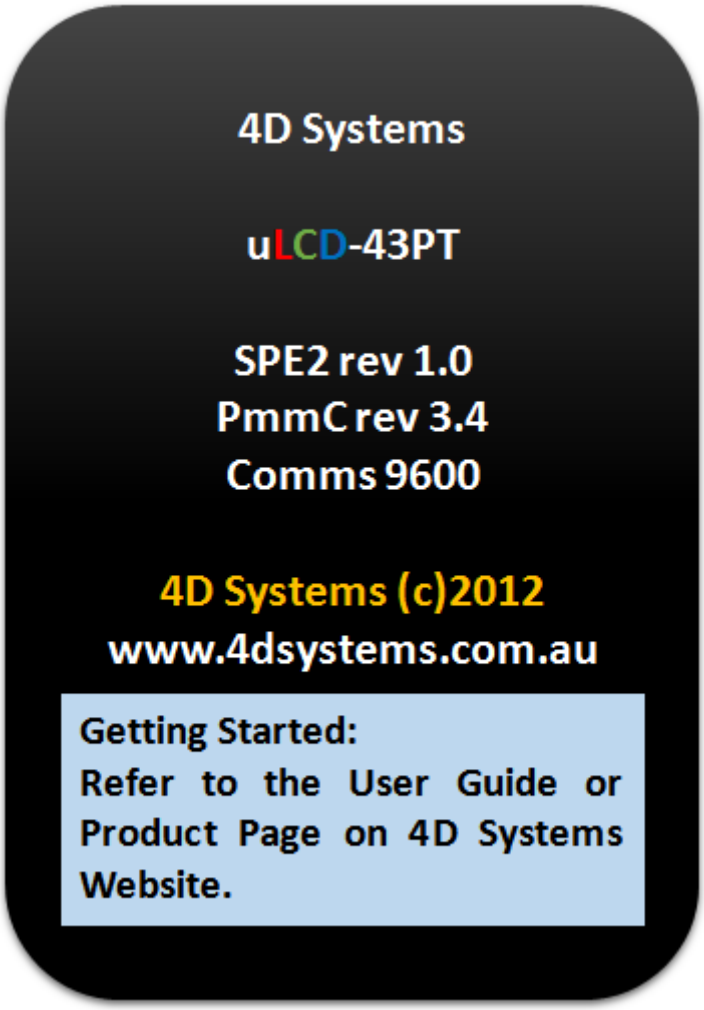


You can also find a number of application notes on our website based on PICASO ViSi category to introduce you to different aspects of developing an application in the PICASO ViSi Environment.

4.4 Serial

This environment also transforms the display module into a slave serial device, allowing the user to control the display from any host microcontroller or device with a serial port. The user cannot access the I2C port. There is only one UART port; that is connected to the Host. The GPIOs are user accessible in this environment.

The display module gets loaded with a 4DGL application called SPE (Serial Platform Emulator) to transform it in to a serial slave device. It's loaded from the factory by default. So, when you power a display module off the box, you would see splash screen similar to this,

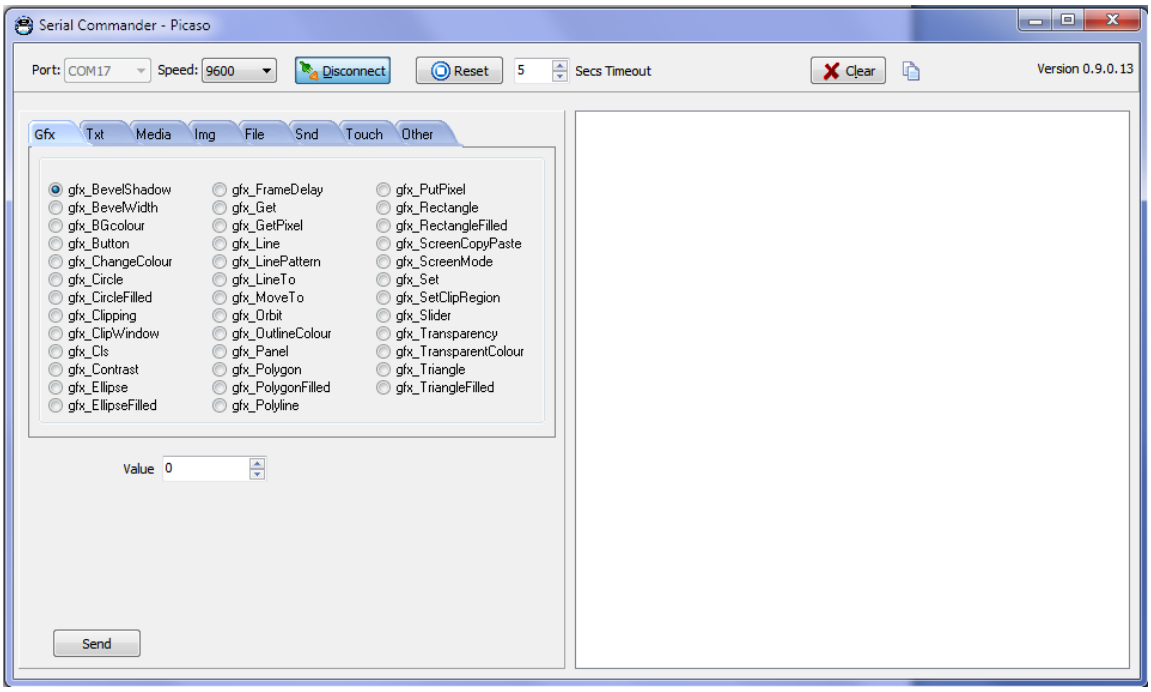


The splash screen displays the model of the display module, SPE revision, PmmC revision and start-up SPE baud rate.

If you have purchased a Display Module only for the 'Serial Environment', you may not need 4D's Programming adaptor as the display modules are factory configured for the Serial Environment. However, it's always good to have one so you could try out other Environments in case you are not too sure of which environment to use before making the purchase. Also, 4D Programming adaptor could be used to test the display module through the Serial Commander, which is a test tool for the module configured for the Serial Environment.

Note: SPE updates are only released if a bug is found and fixed or if there are new update, which doesn't happen very often.

To test the display module in Serial Environment, connect the module with the PC through 4D programming adaptor, open a new project, select 'Serial' environment, go to 'Tools' menu and run Serial Commander.



Now, you can send serial commands to the display module and see the results on the display screen.

Note: Once you have setup the display module in Serial Environment, you don't need any development or coding on the module's side. All the development would be done on the host side.

All the reference documentation and Serial libraries required to build a Serial application on a PICASO display module are available under the Serial category, under the DOWNLOADs tab, on the 4D Wotkshop4 IDE product page

Serial Environment Documentation

PICASO Serial Command Set Reference Manual
Rev 1.12

GOLDELOX Serial Command Set Reference Manual
Rev 1.3

Libraries for use with PICASO Serial Environment

Arduino Serial Library
Github

C Serial Library
Github

Pascal Serial Library
Github

PicAxe Serial Library
Github

Libraries for use with GOLDELOX Serial Environment

Arduino Serial Library
Github

C Serial Library
Github

Pascal Serial Library
Github

PicAxe Serial Library
Github

You can also find a number of application notes on our website based on PICASO Serial category to introduce you to different aspects of developing an application in the PICASO Serial Environment.

5. Switching and selecting a suitable 'Environment'

A number of times, user is not sure which development environment 'best' suits his requirement. Sometimes a display module might be enough to take the control of the whole application without the need of an external controller hence cutting the cost significantly.

For example: A user decides to add a display module to his existing project where a Microcontroller retrieves some data from a sensor through serial interface and controls 5x GPIOs. Now, although he can add a display module in ViSi-Genie environment and send status updates to the display module to be displayed on the screen, he can also completely replace his Microcontroller and use the display module in ViSi environment as a standalone device to communicate with the sensor over serial port directly and control 5x GPIOs. This only requires understanding of 4DGL programming language and the user could make cost savings.

For example: A user has an application where he has to do floating point calculations and display them on the screen, he can do that using any development environment and an external processor to do floating point calculations. The easiest way would be to use the ViSi-Genie environment that provides built-in objects to display the numbers and text in windows fonts and you don't even have to do 4DGL coding. So the external processor sends updates to the display module in ViSi-Genie Environment and the information gets displayed.

To switch the platform, simply hook up the display module with the PC through 4D Programming adaptor and open the application in Designer, ViSi or ViSi-Genie Application and program it to the display module. Or, open a new project in the Serial Environment and program SPE to the module to configure it for Serial Environment. Please note, the base PmmC file for any of the development environment remains the same.

6. Legal Notice

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.

Disclaimer of Warranties & Limitation of Liability

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.

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.

7. Contact Information

For Technical Support: support@4dsystems.com.au

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

Website: www.4dsystems.com.au