



# QUICK START GUIDE

# 4D SYSTEMS

*TURNING TECHNOLOGY INTO ART*

## GOLDELOX Modules

### QUICK START GUIDE

Document Date: 23<sup>rd</sup> May 2013  
Document Revision: 1.0

# Contents

---

<b>1. Introduction .....</b>	<b>3</b>
<b>2. Generic Features.....</b>	<b>4</b>
<b>3. Connecting the Display Module with the PC .....</b>	<b>5</b>
<b>4. Development Environments.....</b>	<b>6</b>
4.1. Designer:.....	6
4.2. ViSi: .....	8
4.3. Serial: .....	10
<b>5. Switching and selecting a suitable 'Environment' .....</b>	<b>13</b>
<b>6. Legal Notice.....</b>	<b>14</b>
<b>7. Contact Information .....</b>	<b>14</b>

## 1. Introduction

This Quick Start Guide is an introduction to becoming familiar with GOLDELOX 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 GOLDELOX 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 GOLDELOX 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.

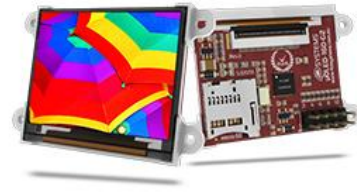
There are 5 different display modules based on the GOLDELOX processor in 4D's product range besides their derivatives. Such as uOLED-128-G2 is based on GOLDELOX processor whereas a 4Display-Shield-128 is its derivative ready to connect the display module to an Arduino Board with the help of available libraries.



uOLED-96-G2  
0.96" Intelligent OLED module



uOLED-128-G2  
1.5" Intelligent OLED module



uOLED-160-G2  
1.7" Intelligent OLED module



uLCD-144-G2  
1.44" Intelligent LCD module



uTOLED-20-G2  
2.0" Intelligent Transparent OLED module

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

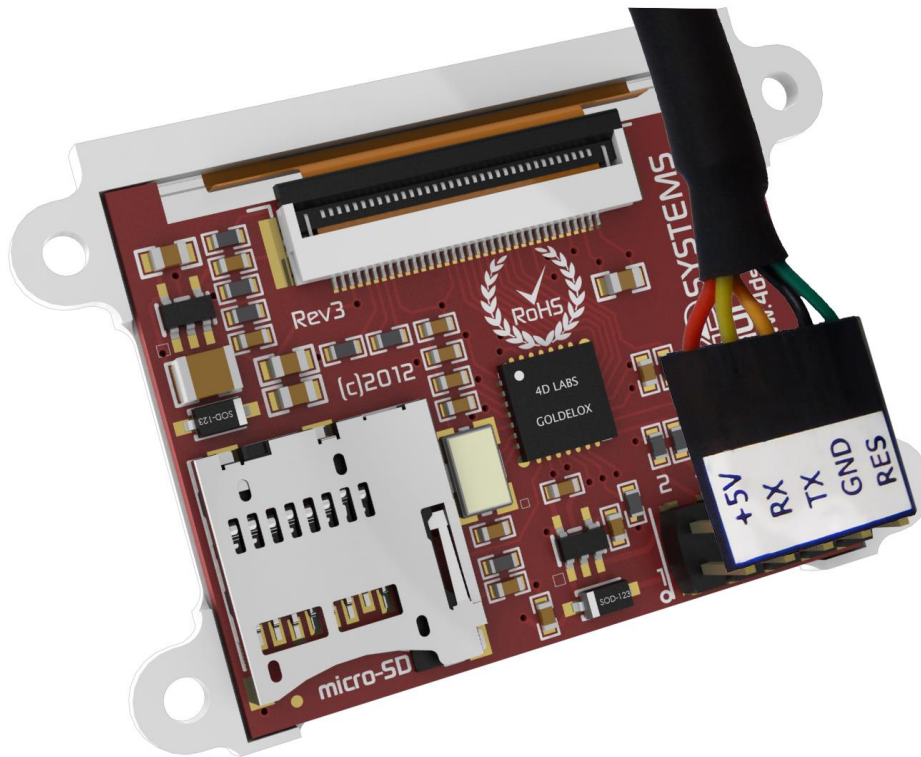
**Note:** Complete details on individual products are available from the relevant product page on our website.

## 2. Generic Features

Feature	μOLED-96-G2	μOLED-128-G2	μOLED-160-G2	μLCD-144-G2	μTOLED-20-G2
0.96" PMOLED display	X				
1.5" PMOLED display		X			
1.7" PMOLED display			X		
1.44" LCD TFT display				X	
2.0" TOLED display					X
RGB 65K true to life colours	X	X	X	X	X
96 x RGB x 64 Resolution	X				
128 x RGB x 128 Resolution		X		X	
160 x RGB x 128 Resolution			X		
128 x RGB x 160 Resolution					X
Touch Panel					
10 pin interface to any host device: 3.3Vout, IO2, GND, IO1, RESET, GND, RX, TX, +5V, 5V OUT	X	X	X	X	X
GOLDELOX processor	X	X	X	X	X
10KB of Flash memory for user code	X	X	X	X	X
510 Bytes of SRAM for user variables	X	X	X	X	X
Asynchronous hardware serial port (COM0), TTL interface, with 300 baud to 600K baud	X	X	X	X	X
General Purpose I/O pin(s). Supports Digital I/O, A/D conversion with 8/10 bit resolution, complex sound generation, dedicated RTTTL tune engine, multi switch joystick/buttons and Dallas 1-wire	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
Display full colour images, animations, icons and video clips	X	X	X	X	X
Supports all available Windows fonts and characters	X	X	X	X	X
4.0V to 5.5V range operation (single supply).	X	X	X	X	X
Four mounting tabs with holes for mechanical support.	X	X	X	X	X
RoHS Compliant	X	X	X	X	X

### 3. Connecting the Display Module with the PC

The GOLDELOX 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.



**uUSB-PA5**



**4D Programming Cable**

Powering a GOLDELOX 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 GOLDELOX display module can be programmed and used in 3 distinct Development Environments categorised in to Standalone or Slave configurations.

- **Standalone Configuration**
  - Designer Environment
  - ViSi Environment
- **Slave Configuration**
  - Serial Environment

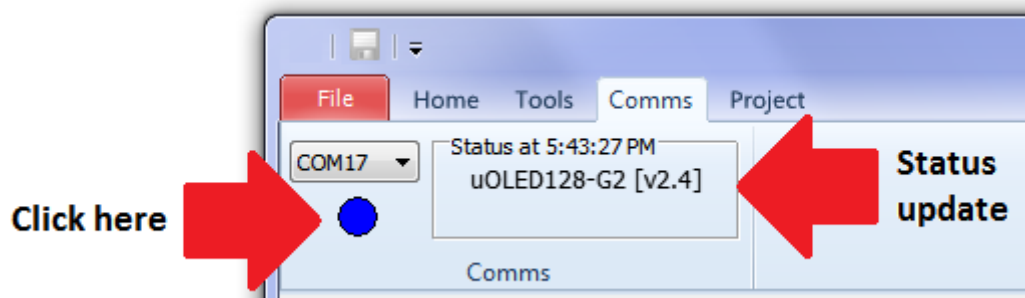
[4D Workshop4 IDE](#) is comprehensive software IDE for Microsoft Windows that provides an integrated software development platform with these development environments, for the user to choose based on application requirements or even user skill level.

### 4.1. Designer:



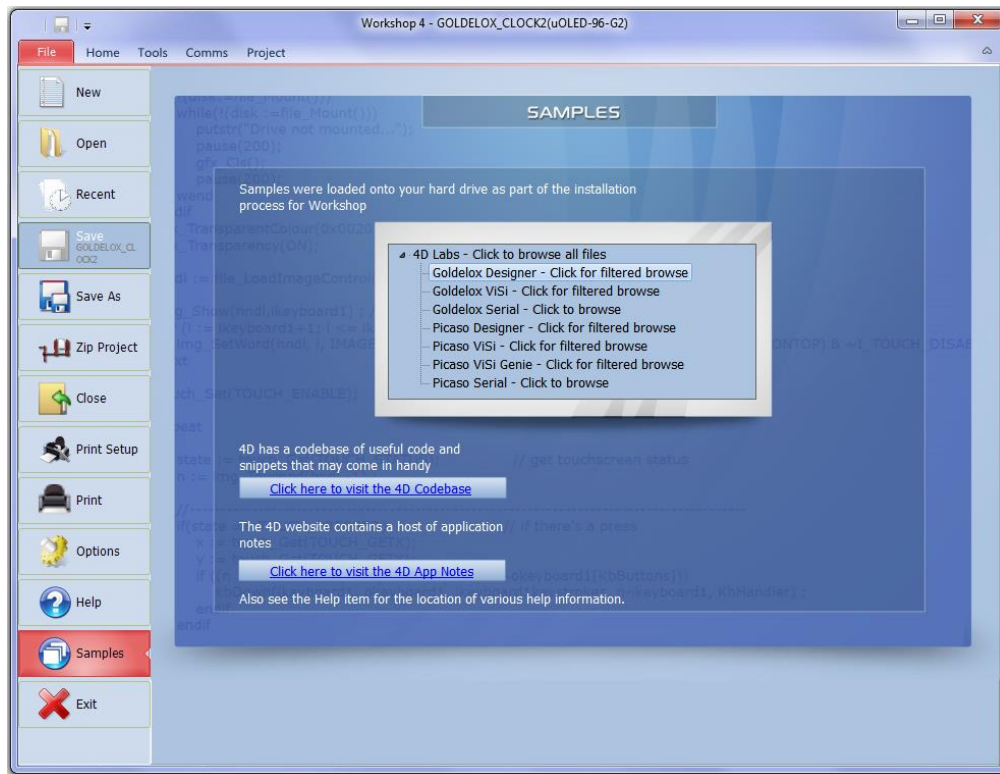
This environment enables the user to write 4DGL code in its natural form to program the display module. The user gets full control of the available GPIOs that could be configured as Digital I/O, Analog input, complex sound generator, multi switch joystick/buttons or Dallas 1-wire. The UART port is also available to communicate with the external serial devices.

To get started with this environment, user need to make sure that R24 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 GOLDELOX Designer section and browse for the GOLDELOX\_CLOCK2.4DG file here, ...\\4D Labs\Goldelox Designer\GENERAL EXAMPLES






Now, Go to Project menu and select your product ID. 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 GOLDELOX display module is available under the Designer category, under the DOWNLOADS tab, on the [4D Workshop4 IDE](#) product page.

**Designer Environment Documentation**

 <p><b>4DGL Programmers Reference Manual</b> Rev 5.2</p>	 <p><b>GOLDELOX 4DGL Internal Functions Reference Manual</b> Rev 6.4</p>
 <p><b>PICASO 4DGL Internal Functions Reference Manual</b> Rev 6.10</p>	

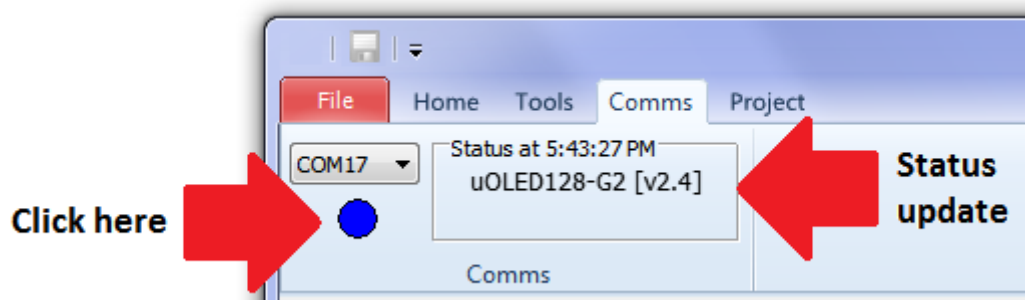
You can also find a number of [application notes](#) on our website based on GOLDELOX Designer category to introduce you to different aspects of developing an application in the GOLDELOX Designer Environment.

## 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 the available GPIOs that could be configured as Digital I/O, Analog input, complex sound generator, multi switch joystick/buttons or Dallas 1-wire. The UART port is also available to communicate with the external serial devices.

To get started with this environment, user need to make sure that R24 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 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.

**Note:** ViSi writes the data to the uSD card in RAW form for the GOLDELOX modules as GOLDELOX processor doesn't support FAT file system. User should make backup of any useful data on the uSD card before approving to copy the necessary data to the uSD card.

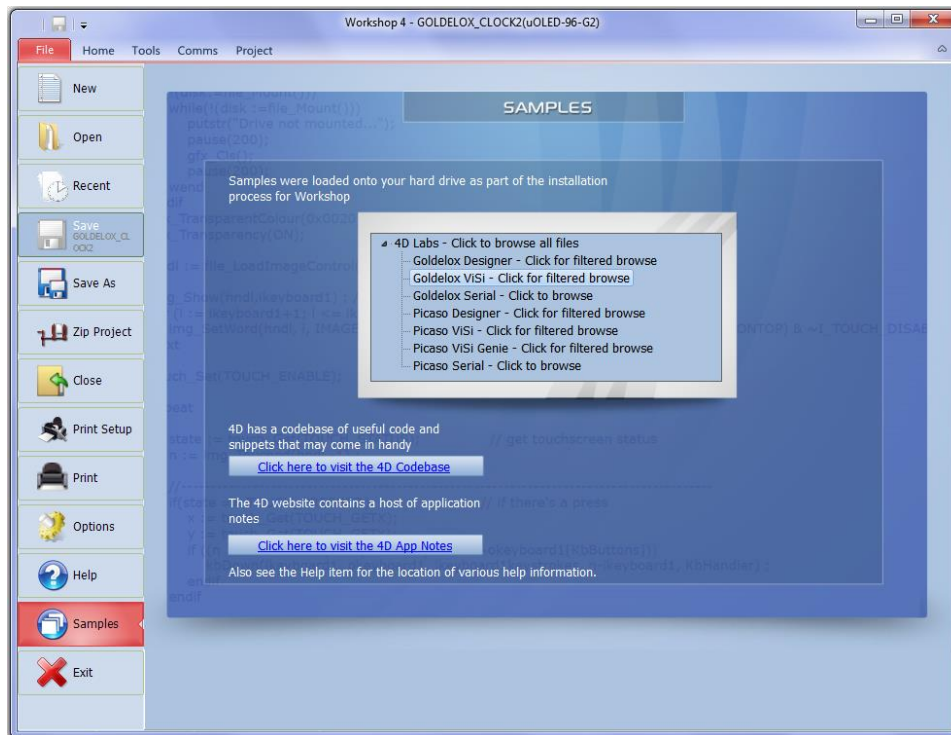
**Note:** If you have a file system on the uSD card, Workshop4 IDE will notify you to get permission before proceeding.

### 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 GOLDELOX ViSi section and browse for the LEDDIGITSGAUGE.4DViSi file here,  
... \4D Labs\GOLDELOX ViSi

**Note:** User might have to scale down the LED digits and the Slider objects in the LEDDIGITSGAUGE.4DViSi demo for the uOLED-96-G2 module as they would not fit 96x64 resolution as is.









Insert a uSD card in to the PC. Now, Go to Project menu in the 4D Workshop4 IDE, select your product ID. 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 GOLDELOX display module is available under the ViSi category, under the DOWNLOADS tab, on the [4D Workshop4 IDE](#) product page

**ViSi Environment Documentation**

 <b>ViSi User Guide</b> Rev 1.0	 <b>4DGL Programmers Reference Manual</b> Rev 5.2
 <b>GOLDELOX 4DGL Internal Functions Reference Manual</b> Rev 6.4	 <b>PICASO 4DGL Internal Functions Reference Manual</b> Rev 6.10

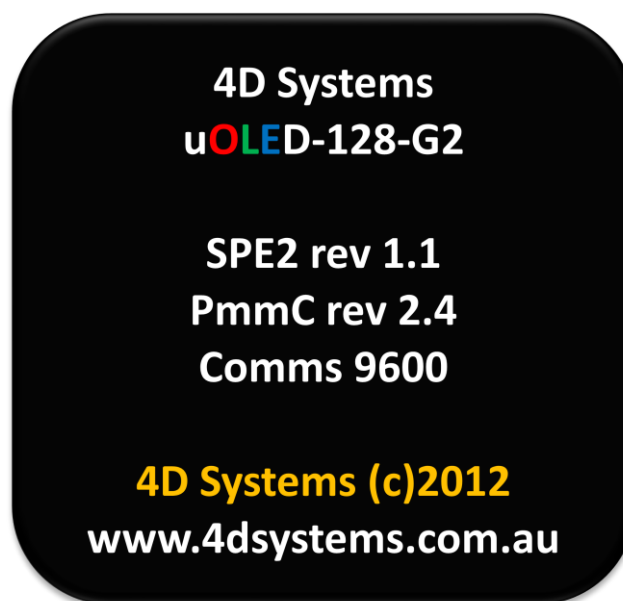
You can also find a number of [application notes](#) on our website based on GOLDELOX ViSi category to introduce you to different aspects of developing an application in the GOLDELOX ViSi Environment.

### 4.3. Serial:



This environment 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. There is only one UART port; that is connected to the Host. The GPIOs are user accessible in this environment too, but restricted to Joysick and Beep commands.

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,

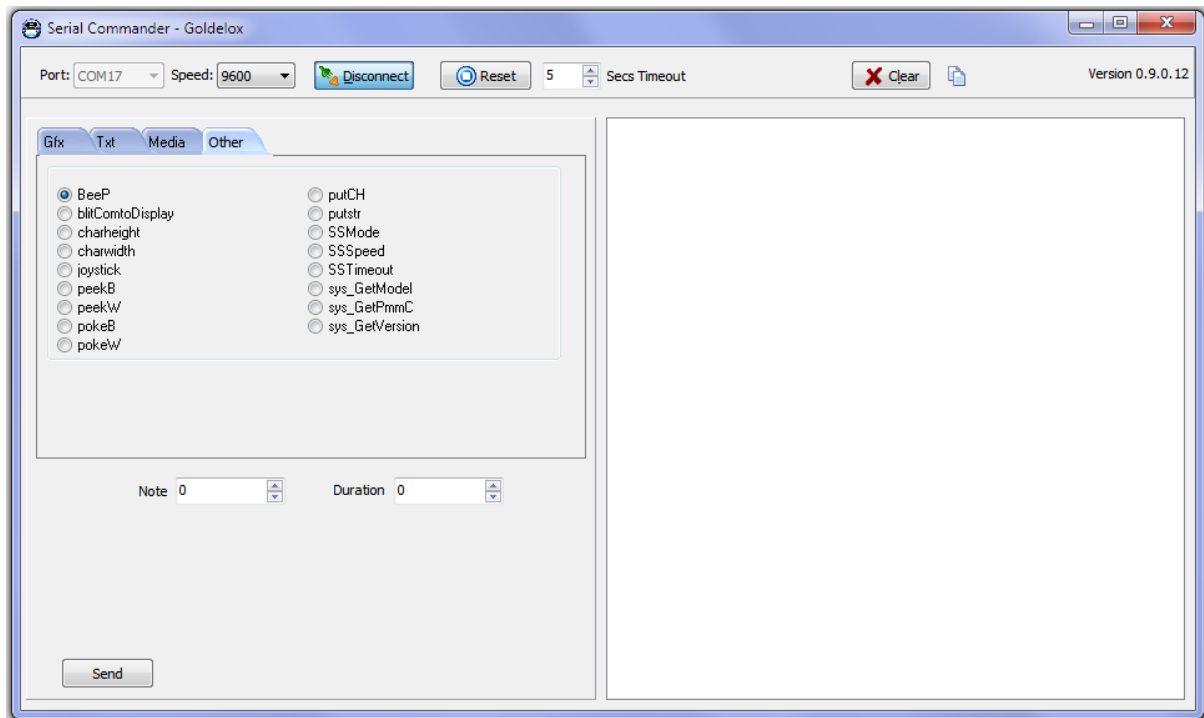


The splash screen displays the model of the display module, SPE revision, PmmC revision and start-up SPE baud rate.

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 GOLDELOX 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 GOLDELOX Serial category to introduce you to different aspects of developing an application in the GOLDELOX Serial Environment.

## 5. Switching and selecting a suitable 'Environment'

A number of times, user is not sure which development environment 'best' suits his requirement. The GPIOs behave differently in each environment. The command set reveals in depth details of how the GPIOs could be used.

For example: A user decides to add a display module to his existing project where he wants to use the analog input but wish to use the module in Serial Slave environment. User will have to make a compromise between Serial Environment or the analog Input as analog input is not supported in the Serial Environment. User can either go with Serial Environment, use the host controller to take care of the analog input part or use the module in Designer or ViSi Environment to make the most of the built-in analog input 4DGL commands.

To switch the platform, simply hook up the display module with the PC through 4D Programming adaptor and open the application in Designer or ViSi 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 make 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.

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.

## 7. Contact Information

For Technical Support: [support@4dsystems.com.au](mailto:support@4dsystems.com.au)

For Sales Support: [sales@4dsystems.com.au](mailto:sales@4dsystems.com.au)

Website: [www.4dsystems.com.au](http://www.4dsystems.com.au)

**Copyright 4D Systems Pty. Ltd. 2000-2013.**