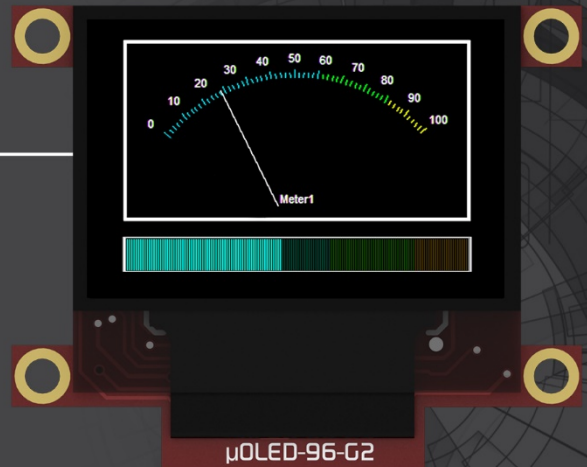


GETTING STARTED MANUAL

uOLED-96-G2

0.96" INTELLIGENT OLED MODULE



This manual will help you get started using the uOLED-96-G2 module along with the WorkShop4 IDE. It also includes a list of essential project examples and application notes.

CONTENTS

03 Introduction

System Requirements

- Hardware Requirements
- Software Requirements

06 Connecting the Display Module to the PC

- Connection Options
- Let WS4 Identify the Display Module

08 Getting Started with a Simple Project

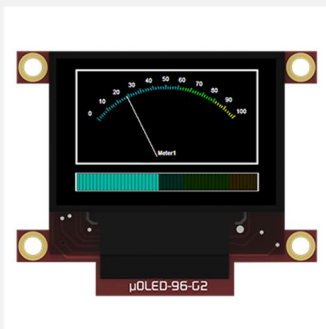
14 Application Notes

15 Reference Documents

17 Glossary

- Hardware
- Software

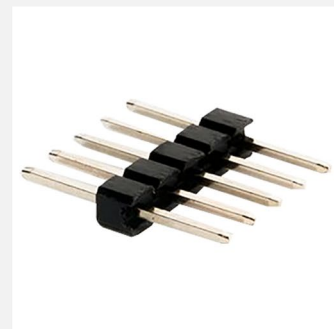
WHAT'S IN THE BOX



uOLED-96-G2



5-way Female-Female
Jumper cable



5-way Male-Male
Adaptor

Supporting documents, datasheet, CAD step models and application notes are available on the 4D Systems website.

INTRODUCTION

This Getting Started Manual is an introduction to the uOLED-96-G2 and the WorkShop4 software IDE associated with it. This manual should be treated only as a useful starting point and not as a comprehensive reference document. Refer to Application Notes for a list of all the detailed reference documents.

In this Getting Started Manual we will briefly focus on the following topics:

- Hardware and Software Requirements
- Connecting the Display Module to your PC
- Getting Started with Simple Projects
- Reference Documents

The **uOLED-96-G2** is designed and manufactured by 4D Systems. It is a compact and visually stunning display module using the latest state of the art Passive Matrix OLED (PMOLED) technology with an embedded GOLDELOX graphics processor that can deliver 'stand-alone' functionality to any project.

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

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

SYSTEM REQUIREMENTS

The following subsections discuss the hardware and software requirements for this manual.

1 Hardware

1.1 Intelligent Display Module

The uOLED-96-G2 intelligent display module and its accessories (5-way cable and male-male adaptor) should be included in the box delivered to you after your purchase from our website or through one of our distributors. Please refer to the section “What’s in the Box” for images of the display module and its accessories.

1.2 4D Programming Cable

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

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

To use any of the programming modules, the corresponding driver must first be installed in the PC. You may refer to the product page of the given module for more information and detailed instruction.

1.3 microSD Card

Workshop4 has built-in widgets that can be used to design your display UI. In order to use these widgets, they should be saved first into the microSD Card along with the other graphic files during the compilation step.

Note: *The microSD Card is optional and is only needed with projects that are utilizing graphical files. Please note as well that not all cards on the market are SPI compatible, and therefore not all cards can be used in 4D Systems products. Buy with confidence, choose the cards recommended by 4D Systems.*

1.4 Windows PC

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

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

2 Software

2.1 Workshop4 IDE

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

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

Workshop4 Environments

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

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

Serial This environment is also provided to transform the display module into a slave serial display module, allowing the user to control the display from any host microcontroller or device with a serial port.

Install Workshop4

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

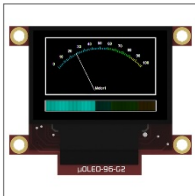
CONNECTING THE DISPLAY MODULE TO THE PC

This section shows the steps for connecting the display to the PC. Please refer to the image below for the complete instructions.

Hardware

Software

Option A



Display Module



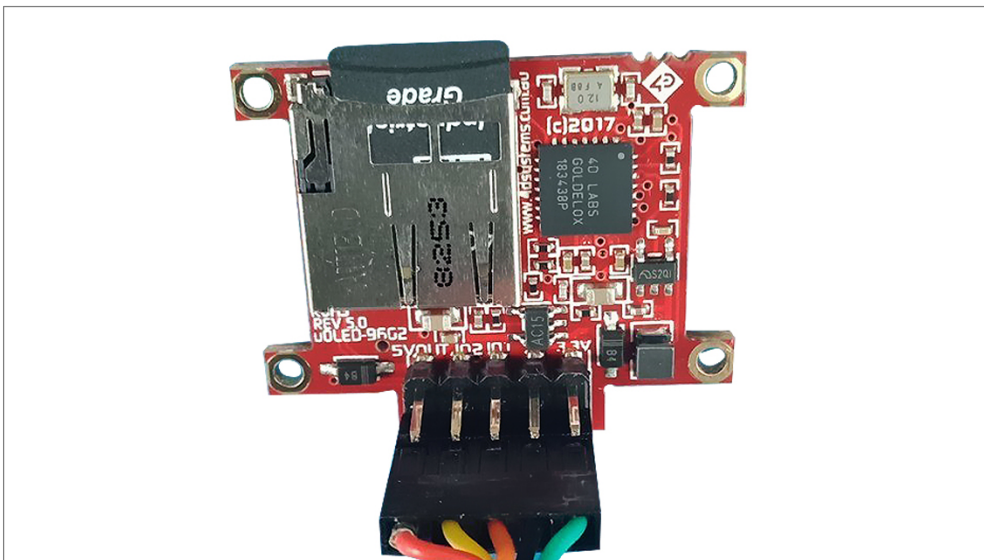
4D Programming Cable



Workshop4

CONNECTION OPTIONS

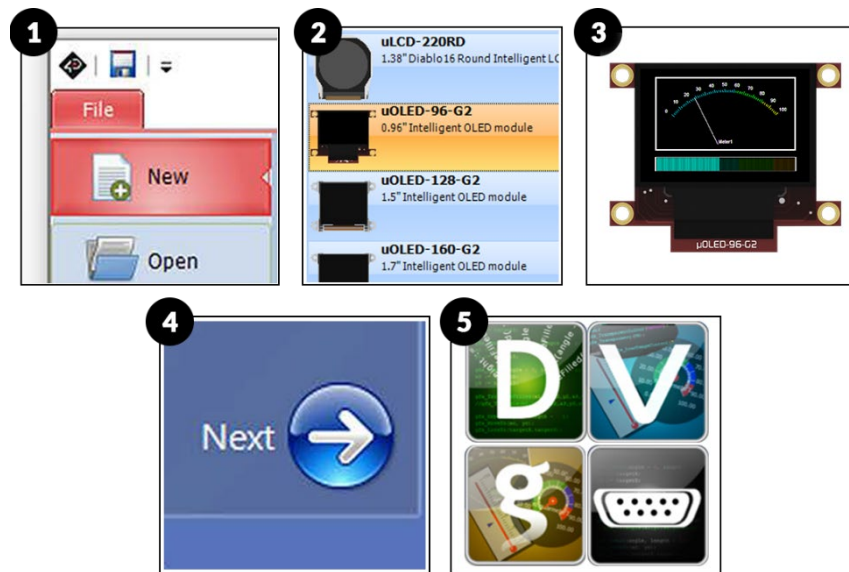
Connect the Programming cable to the display. Please refer to the image below for proper insertion. Pin1 corresponds to 5V input.



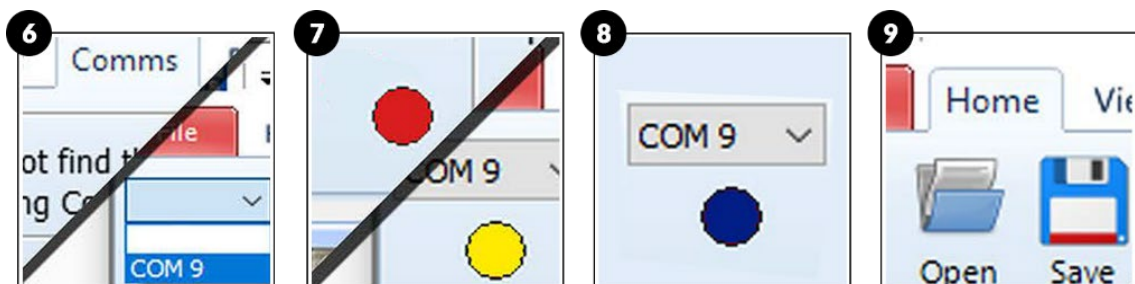
LET WS4 IDENTIFY THE DISPLAY MODULE

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

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



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



GETTING STARTED WITH A SIMPLE PROJECT

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

The resulting project consists of LEDDIGITS and GAUGE widgets. The value of the two widgets shall continuously change randomly, thus the shown digit and level shall change correspondingly.

Note: This sample project can be found in Workshop4 upon installation. Click on Files > Samples > Goldelox ViSi > LEDDIGITS GAUGE.

Create a New ViSi Project to Create a New ViSi Project

You can create a ViSi project by opening Workshop4 and choosing the display type and the environment that you want to work with. This project will be using the ViSi environment.

1. Open Workshop4 by double-clicking the icon.
2. Create New Project with the New Tab.
3. Choose your display type.
4. Click Next.
5. Choose ViSi.

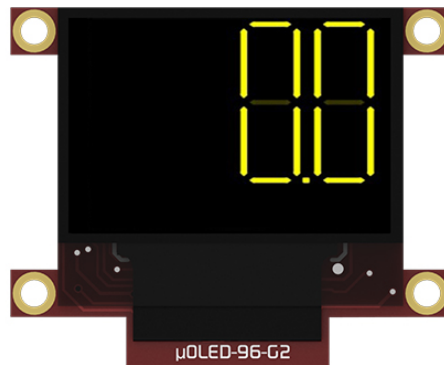


Add a Leddigits Widget

To add a Leddigits widget, simply click on the Widgets Tab and choose the Digits sub-tab. From the list, you may choose the type of widget that you want to use. In this case, the Leddigits widget.

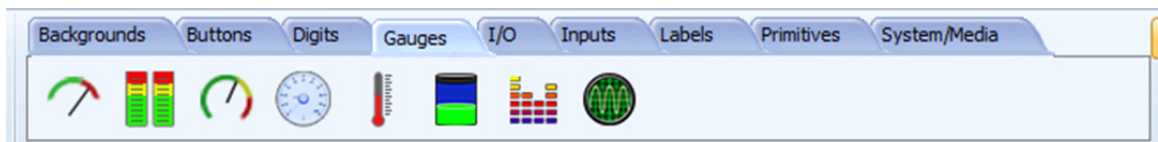


Simply Drag-and-Drop the widget towards the What-You-See-Is-What-You-Get (WYSIWYG) area. You may change the properties (color, number of digits, etc.) of the widget in the Object Inspector window.



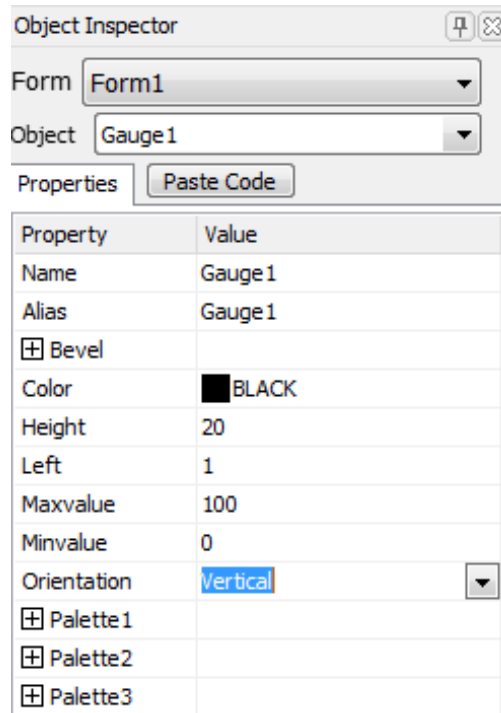
Add a Gauge Widget

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

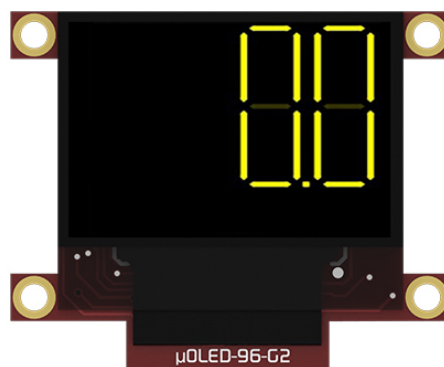


Drag and drop it towards the WYSIWYG area to proceed. You may change the properties (color, number of digits, etc.) of the widget in the Object Inspector window.

Set the *Orientation* option to Vertical.



The WYSIWYG area should now look like this:



Programming the events

To control the widgets in the ViSi environment, 4DGL is used. Please paste the following code to the code editor area.

```
#inherit "4DGL_16bitColours.fnc"
#inherit "VisualConst.inc"
#inherit "LedDigitsGaugeConst.inc"
#inherit "LedDigitsDisplayGoldelox.inc"

func main()
  var numx ;
  print("Starting\n") ;
  while(!media_Init())
    putstr("Drive not mounted...");
    pause(200);
    gfx_Cls();
    pause(200);
  wend

  media_SetAdd(iLeddigits1H, iLeddigits1L) ;// point to the
  Leddigits1 image
  media_Image(0, 0) ;      // show all digits at 0, only do this
  once

  repeat
    numx := ABS ( RAND() % 1000) ;

    ledDigitsDisplay(numx, iLeddigits1H, iLeddigits1L, 0, 0,
4, 2, 32, 1);

    numx /= 10 ;
    media_SetAdd(iGauge1H, iGauge1L) ;      // point to the
  Gauge1 image
    media_VideoFrame(0, 108, numx) ;      // where numx is 0
  to 100 (for a displayed 0 to 100)

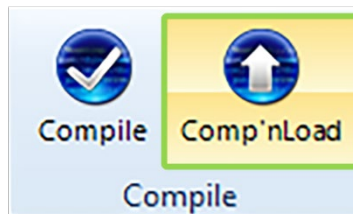
    pause(250) ;
  forever

endfunc
```

This code sets both widgets to change in value every 250ms. To know more about 4DGL, please see References section.

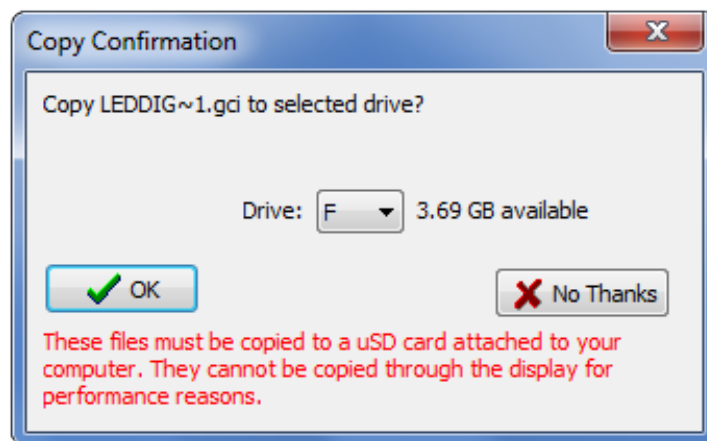
Build and Compile the Project

To Build/Upload the project, click the Comp'nLoad icon.



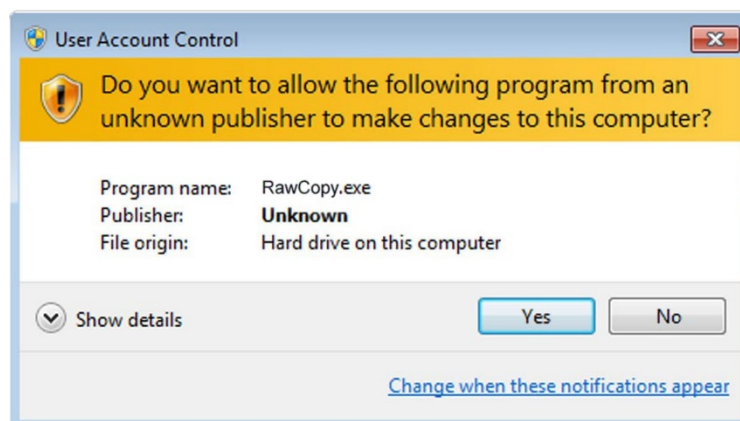
Copy the Required Files to the microSD Card

In order to display widgets on the display, it needs a microSD card to store graphic files like the leddigits and gauge widgets. To do this, make sure that you have a microSD card inserted in one of your PC ports and the display is properly connected.

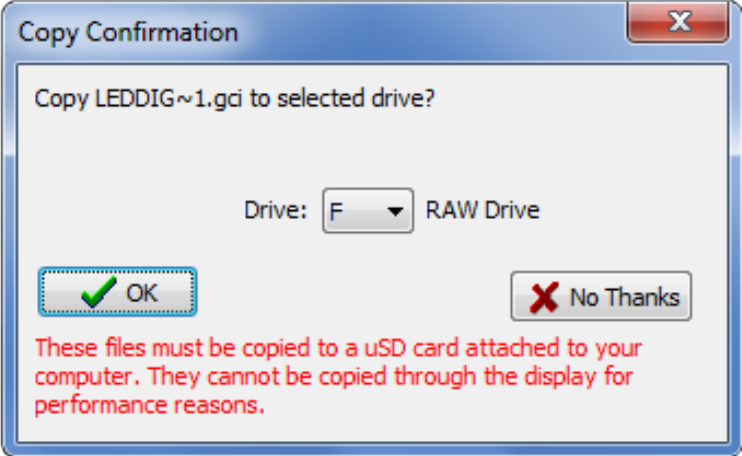


There will be a pop-up window asking for the microSD card that you want to write into. Choose your microSD card and click on OK. Click "Yes" for the User Account Control window and wait for the program to load.

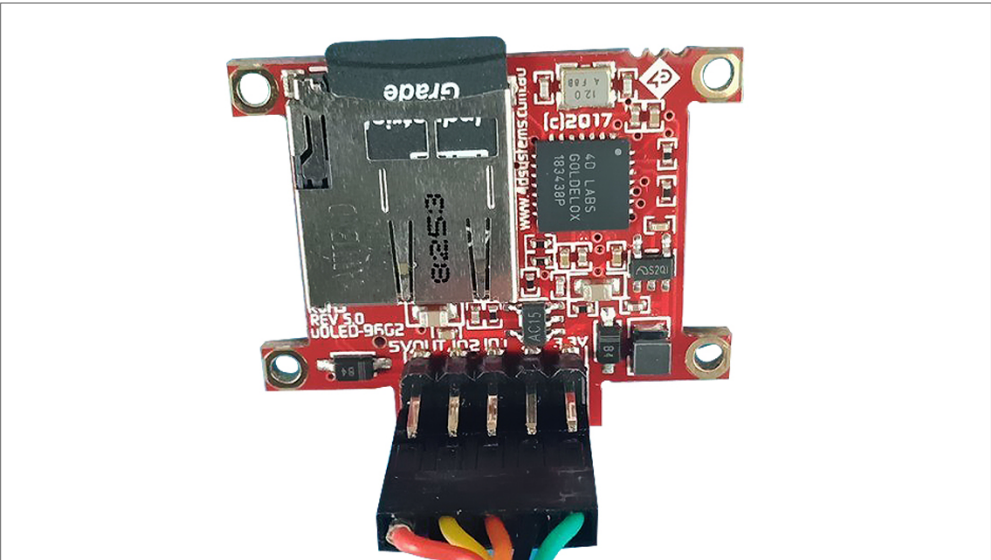
Note: The ViSi application writes the graphics files into the microSD card in RAW format for GOLDELOX display modules. In other words, the FAT formatting will be removed. Please make sure you backup any important files/data you have on the microSD card before proceeding.



You may upload your graphics files after the conversion of the FAT format to RAW. It will also be indicated in the pop-up window that the microSD Card is already in the RAW format. Before proceeding, make sure that you have selected the right microSD Card.



Unmount the microSD Card from the PC and insert it to the display's microSD Card slot.



APPLICATION NOTES

App Note	Title	Description	Supported Environment
<u>4D-AN-00117</u>	Designer Getting Started - First Project	This application note shows how to create a new project using the Designer Environment. It also introduces the basics of 4DGL (4D Graphics Language).	Designer
<u>4D-AN-00118</u>	ViSi Getting Started - First Project for Goldelox	This application note shows how to create a new project using the ViSi Environment. It also introduces the basics of 4DGL (4D Graphics Language and the basic use of the WYSIWYG (What-You-See-Is-What-You-Get) screen.	ViSi
<u>4D-AN-00086</u>	Serial Goldelox Getting Started - The SPE Application	This application note shows how to configure a 4D display to act as a serial slave and how to use the Serial Commander (one of the several tools available in Workshop) to send commands to the display. This knowledge will then allow the user to properly program any external host controller for the display module.	Serial
<u>4D-AN-00058</u>	Designer or ViSi How to Draw Circles and Rectangles	This application note shows how to program a 4D display module in the Designer environment to make it draw circles and rectangles on the screen.	Designer, ViSi
<u>4D-AN-00059</u>	Designer or ViSi How to Draw Circles and Rectangles	This application note shows how to program a 4D display module in the Designer environment to make it draw circles and rectangles on the screen.	Designer, ViSi
<u>4D-AN-00060</u>	Designer or ViSi How to Draw Triangles and Polygons	This application note shows how to program a 4D display module in the Designer environment to make it draw triangles and polygons on the screen.	Designer, ViSi

REFERENCE DOCUMENTS

The following are documents to further enhance the user's ViSi environment knowledge and to start learning about 4DGL programming:

4DGL Programmer Reference Manual

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

Quick Start Guide

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.

Goldelox Embedded Graphics Processor

The Goldelox is a custom embedded graphics controller designed to interface with many popular OLED and LCD display panels. This document lists and describes the specifications and features of the Goldelox processor.

Goldelox Processor Internal Functions Reference Manual

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

Goldelox Processor Serial Commands Reference Manual

The Goldelox Processor can be programmed to act as a 'SERIAL SLAVE' device, responding to the Serial commands sent from virtually any Host Controller. This document shows lists the commands applicable for the Goldelox processor.

uOLED-96-G2 Datasheet

This document lists the detailed information about the uOLED-96-G2.

Note: For more information on the Workshop4 Software in General or the other Environments available in Workshop4, please refer to the Workshop4 User Guide, available from the 4D Systems website.

NOTES



GLOSSARY

Hardware

1. **4D Programming Cable** – The 4D Programming Cable is a USB to Serial-TTL UART converter cable. The cable provides a fast and simple way to connect all of the 4D devices that require TTL level serial interface to USB.
2. **Embedded System** – is a programmed controlling and operating system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts.
3. **Female Header** – is a connector attached to a wire, cable, or piece of hardware, having one or more recessed holes with electrical terminals inside.
4. **Micro USB cable** – a type of cable used to connect the display to a computer.
5. **Processor** – is a small chip that resides the 4D Systems displays. Its basic job is to receive input and provide the appropriate output.
6. **Programming Adaptor** – used for programming 4D Systems display modules, interfacing to a breadboard for prototyping, interfacing to Arduino and Raspberry Pi interfaces
7. **microSD Card** – a type of removable flash memory card used for storing information.

Software

1. **Comm Port** – a serial communication used to connect devices such as your display.
2. **Driver** – is a group of files that enable one or more hardware devices to communicate with the display's processor
3. **Firmware** – a specific class of computer software that provides the low-level control for the device's specific hardware.
4. **GUI** – It is a form of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation.
5. **Image Files** – are graphics files generated upon program compilation that should be saved into the microSD Card.
6. **Object Inspector** – a section in Workshop4 where the user can change the properties of a certain widget. This is where the widgets customization and Events configuration happen.
7. **PmmC** - The PmmC contains the low level micro-code information (analogy of that of a soft silicon) which define the characteristics and functionality of the PICASO processor. The chip-resident 4DGL functions are part of the PmmC configuration file and future proofing and enhancements are available via PmmC updates. The PmmC file is programmed into the module via the COM port.
8. **Programming Environment** – is a type of programming means on which the users wants to use. In Workshop4, the four environments are Designer, ViSi-Genie, ViSi, and Serial.
9. **Widget** – graphical objects in Workshop4.
10. **WS4 IDE** – a comprehensive software IDE for Microsoft Windows that provides an integrated software development platform for all of the 4D family of processors and modules. The IDE combines the Editor, Compiler, Linker and Downloader to develop complete 4DGL application code. All user application code is developed within the Workshop4 IDE.
11. **WYSIWYG** – What-You-See-Is-What-You-Get

Visit our website at: www.4dsystems.com.au

Technical Support: www.4dsystems.com.au/support

Sales Support: sales@4dsystems.com.au