APPLICATION NOTES

4D SYSTEMS
TURNING TECHNOLOGY INTO ART

Serial Picaso Getting Started – The SPE Application

DOCUMENT DATE: **16th APRIL 2019**
DOCUMENT REVISION: **1.1**

Description

This application note explores the possibilities provided by the Serial environment in Workshop for a 4D display to work with an external host controller. In this example, the host is a PC thru the use of the Serial Commander.

Before getting started, the following are required:

- Any of the following 4D Picaso display modules:

gen4-uLCD-24PT	gen4-uLCD-28PT	gen4-uLCD-32PT
uLCD-24PTU	uLCD-32PTU	uVGA-III

  other superseded modules which support the Serial environment
- 4D Programming Cable / uUSB-PA5/uUSB-PA5-II for non-gen4 displays(uLCD-xxx)
- 4D Programming Cable & gen4-PA, / gen4-IB / 4D-UPA for gen4 displays (gen4-uLCD-xxx)
- Workshop 4 IDE (installed according to the installation document)
- When downloading an application note, a list of recommended application notes is shown. It is assumed that the user has read or has a working knowledge of the topics presented in these recommended application notes.

Content

Application Overview

The Picaso processor can be programmed to act as a serial slave device, responding to serial commands received from any host controller. Communication between the slave (the 4D Picaso display) and the host must follow the Serial Protocol, which defines a set of commands with associated parameters. These commands enable the host controller to make the slave device display primitives, text, and images, play audio and video, log data to a micro-SD card, or receive touch events. For more information on the Serial Protocol, refer to the Picaso Serial Command Set Reference Manual.

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. Thru the use of the Serial Commander, the programmer will be acquainted with the Serial Protocol or more specifically the nature and format of the serial commands. This knowledge will then allow the user to properly program any external host controller for the display module.

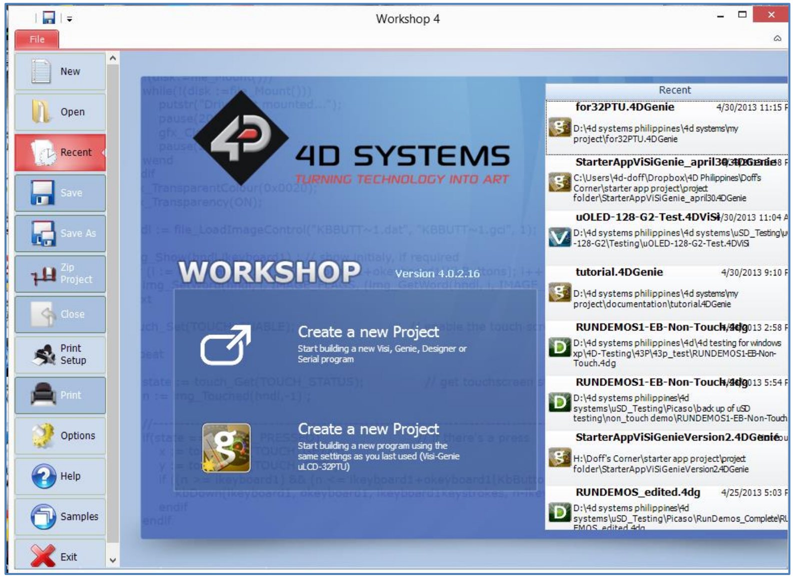
Setup Procedure

Launch Workshop 4

There is a shortcut for Workshop 4 on the desktop. Launch Workshop 4 by double-clicking on the icon.

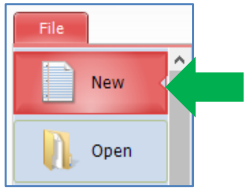
Create a New Project

Workshop 4 opens and displays the **Recent** page.

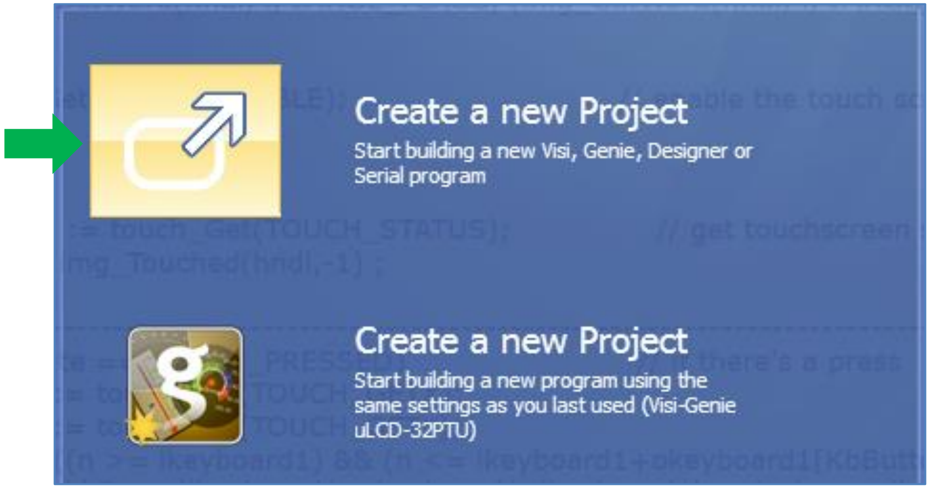

Workshop 4
File
New
Open
Recent
Save
Save As
Zip Project
Close
Print Setup
Print
Options
Help
Samples
Exit
4D SYSTEMS
TURNING TECHNOLOGY INTO ART
WORKSHOP
Version 4.0.2.16
Create a new Project
Start building a new Visi, Genie, Designer or Serial program
Create a new Project
Start building a new program using the same settings as you last used (Visi-Genie uLCD-32PTU)
Recent
for32PTU.4DGenie
StarterAppViSiGenie_april30.4DGenie
uOLED-128-G2-Test.4DViSi
tutorial.4DGenie
RUNDEMOS1-EB-Non-Touch.4dg
RUNDEMOS1-EB-Non-Touch.4dg
StarterAppViSiGenieVersion2.4DGenie
RUNDEMOS_edited.4dg

To create a new project, there are two options.

Click on the top left-most icon, New.



Or Click on the icon beside Create a new Project.



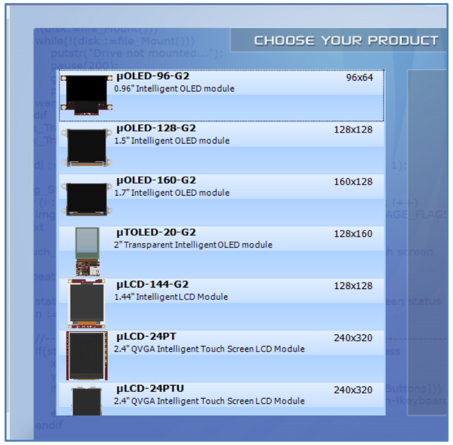
Create a new Project

Start building a new Visi, Genie, Designer or Serial program

Create a new Project

Start building a new program using the same settings as you last used (Visi-Genie uLCD-32PTU)

The Choose-Your-Product window appears.

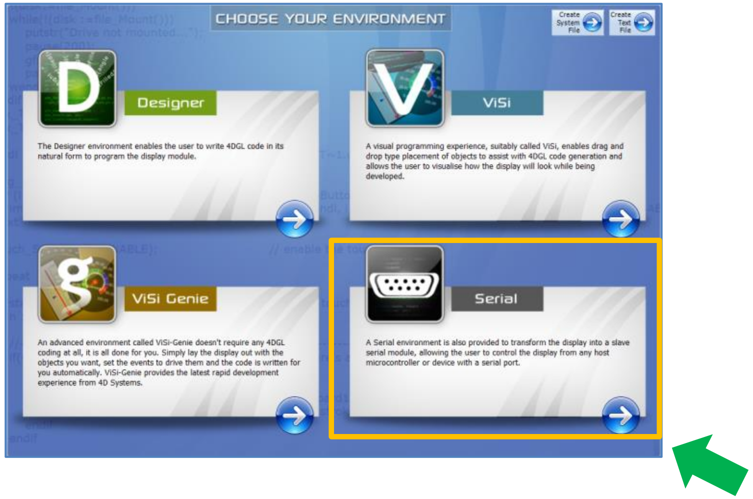


CHOOSE YOUR PRODUCT

µOLED-96-G2 0.96" Intelligent OLED module	96x64
µOLED-128-G2 1.5" Intelligent OLED module	128x128
µOLED-160-G2 1.7" Intelligent OLED module	160x128
µTOLED-20-G2 2" Transparent Intelligent OLED module	128x160
µLCD-144-G2 1.44" Intelligent LCD Module	128x128
µLCD-24PT 2.4" QVGA Intelligent Touch Screen LCD Module	240x320
µLCD-24PTU 2.4" QVGA Intelligent Touch Screen LCD Module	240x320

Select the appropriate screen and preferred orientation. The screen used in this example is the **uLCD-32PTU (Portrait orientation)**.

Select Serial



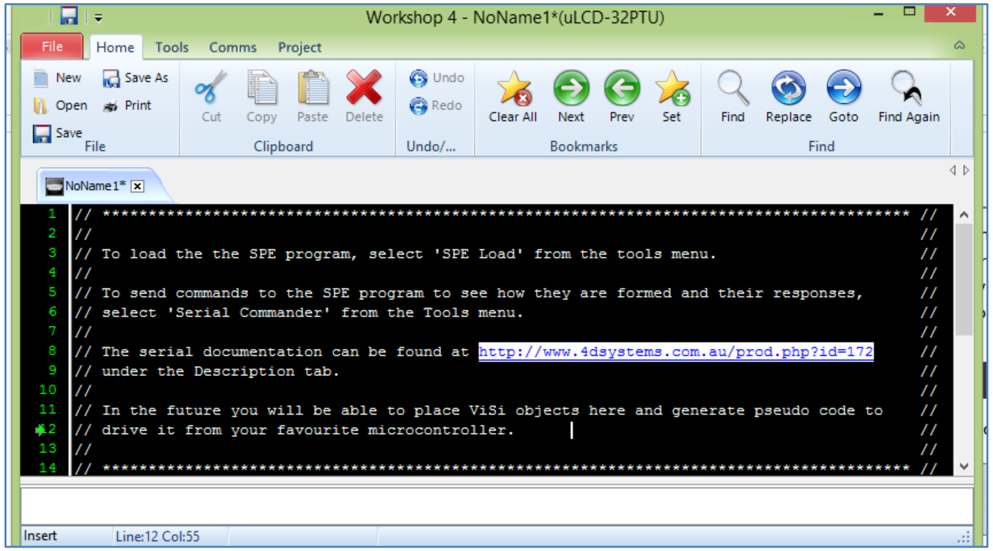
The Designer environment enables the user to write 4DGL code in its natural form to program the display module.

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 for you. Simply lay the display out with the objects you want, 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.

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

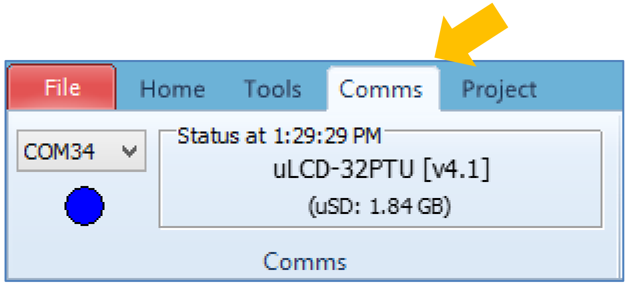
A new file opens and displays some information.



```
// **************************************************************************************** //
//                                                                                          //
// To load the the SPE program, select 'SPE Load' from the tools menu.                      //
//                                                                                          //
// To send commands to the SPE program to see how they are formed and their responses,      //
// select 'Serial Commander' from the Tools menu.                                           //
//                                                                                          //
// The serial documentation can be found at http://www.4dsystems.com.au/prod.php?id=172     //
// under the Description tab.                                                               //
//                                                                                          //
// In the future you will be able to place ViSi objects here and generate pseudo code to    //
// drive it from your favourite microcontroller.                                            //
//                                                                                          //
// **************************************************************************************** //
```

Connect the Display Module

Connect the display module to the PC using a 4D USB Programming Cable or a µUSB-PA5 programming adaptor. Go to the Comms menu to check if the module is detected. Below is an example of how the Comms tab will look like if a uLCD-32PTU is connected to the PC.



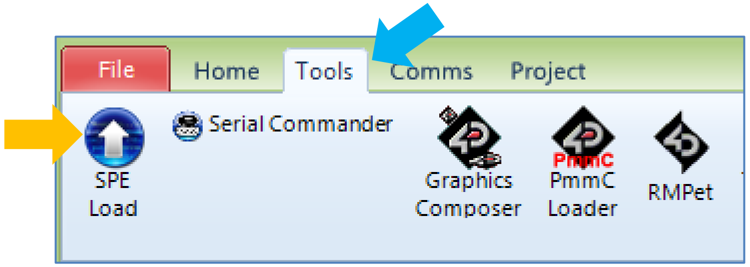
Status at 1:29:29 PM
uLCD-32PTU [v4.1]
(uSD: 1.84 GB)

If not familiar with how to actually connect a 4D display to the PC using a 4D USB programming cable or a uUSB-PA5 and with how to update the firmware, the user should consult the document:

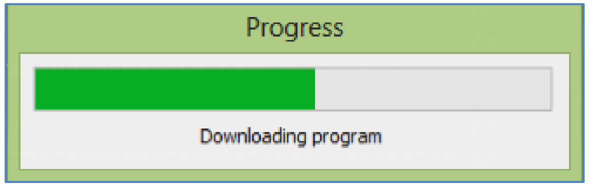
General How to Update the PmmC for Picaso

Load the SPE Application

Under Tools menu, click on the SPE Load icon.



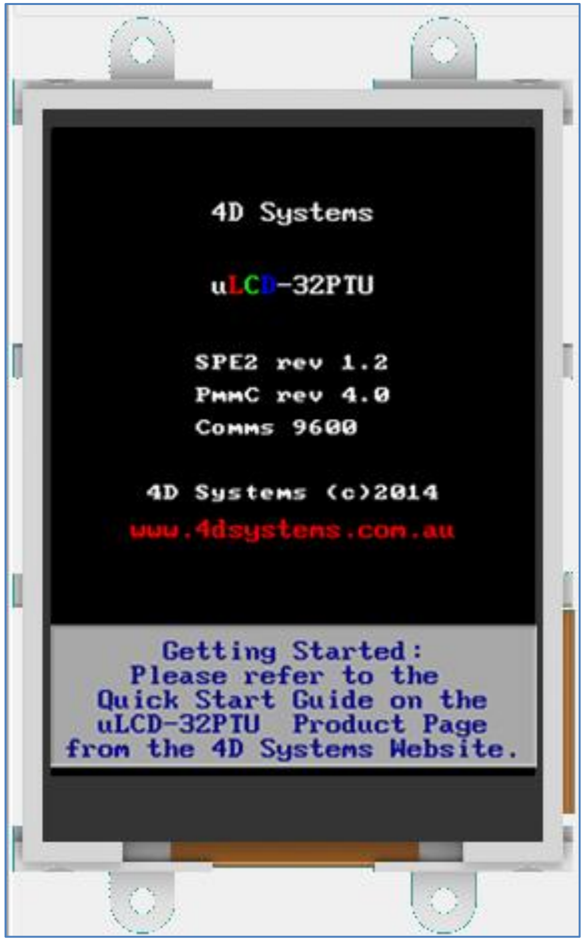
The application is now downloaded to the display module.



Progress

Downloading program

The SPE splash screen should now be shown on the display module.



4D Systems

uLCD-32PTU

SPE2 rev 1.2
PmmC rev 4.0
Comms 9600

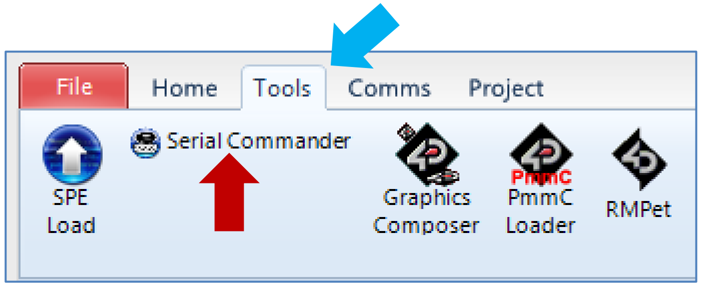
4D Systems (c)2014
www.4dsystems.com.au

Getting Started:
Please refer to the
Quick Start Guide on the
uLCD-32PTU Product Page
from the 4D Systems Website.

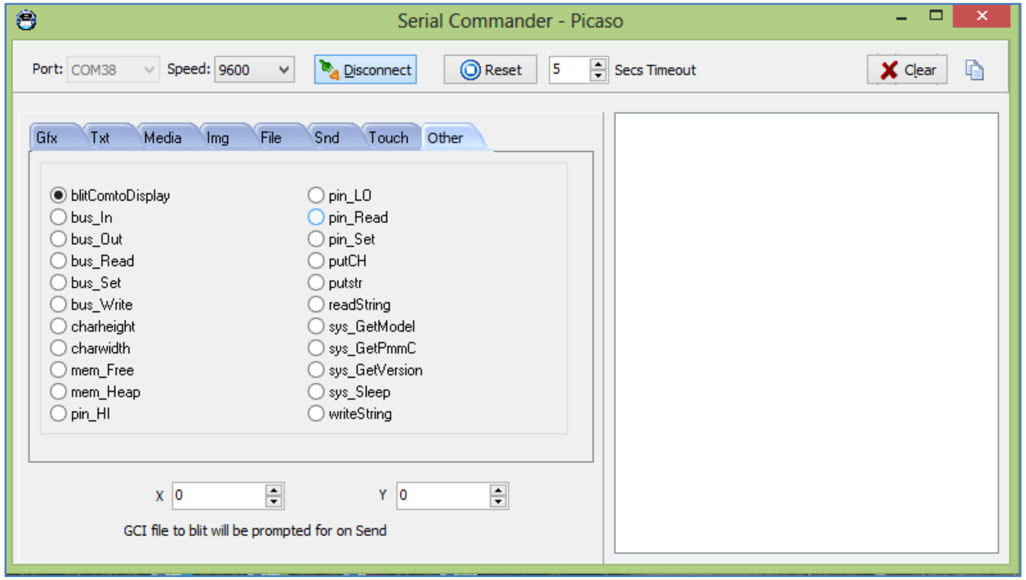
The default baud rate is 9600 and the latest firmware is loaded on to the display. To learn how to modify the default settings, refer to **Section 2.2** (**Additional Configuration Parameters for Serial Communication**) of the Picaso Serial Command Set Reference Manual.

Launch the Serial Commander

The **Serial Commander** icon is found under the **Tools** menu in Workshop. Note that the Serial Commander is available in the Serial environment **only**.



The Serial Commander window appears.



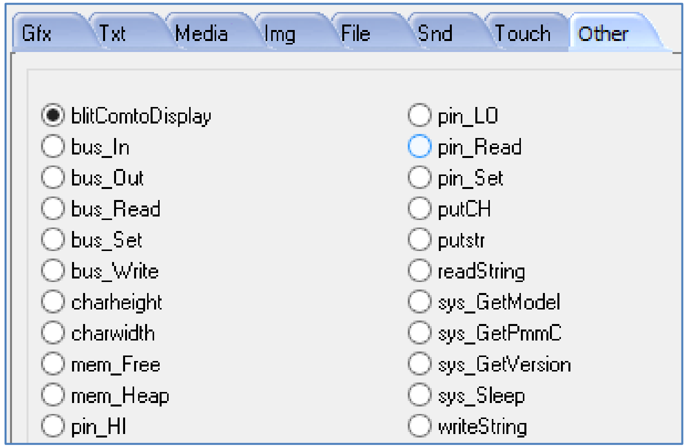
Port: COM38 Speed: 9600 5 Secs Timeout

● blitComtoDisplay
○ bus_In
○ bus_Out
○ bus_Read
○ bus_Set
○ bus_Write
○ charheight
○ charwidth
○ mem_Free
○ mem_Heap
○ pin_HI
○ pin_LO
○ pin_Read
○ pin_Set
○ putCH
○ putstr
○ readString
○ sys_GetModel
○ sys_GetPmmC
○ sys_GetVersion
○ sys_Sleep
○ writeString

X 0 Y 0

GCI file to blit will be prompted for on Send

Note how the commands are categorized. These commands are documented in the Picaso Serial Command Set Reference Manual.



● blitComtoDisplay
○ bus_In
○ bus_Out
○ bus_Read
○ bus_Set
○ bus_Write
○ charheight
○ charwidth
○ mem_Free
○ mem_Heap
○ pin_HI
○ pin_LO
○ pin_Read
○ pin_Set
○ putCH
○ putstr
○ readString
○ sys_GetModel
○ sys_GetPmmC
○ sys_GetVersion
○ sys_Sleep
○ writeString

The Serial Protocol

The Serial Protocol is made up of commands and parameters that are sent over the Serial Port in byte format to the display module. Each command is unique and has a specific set of parameters associated with it. Some commands expect a response from the display module (e.g., when the host is trying to detect the current touch status of the display). Some commands do not require a specific response (e.g., the command for clearing the screen). For the latter type, the display module will reply with an acknowledgement once the command is successfully executed. For commands that require a specific response, the display module may reply with a varying number of bytes, depending on the command and what the response is. Each command sent to the display will require a certain amount of time before the response is sent. This depends on the command and the operation that has to be performed. Commands should only be sent and their response received, before another command is sent. If two commands are sent before the first response is received, incorrect operation may follow.

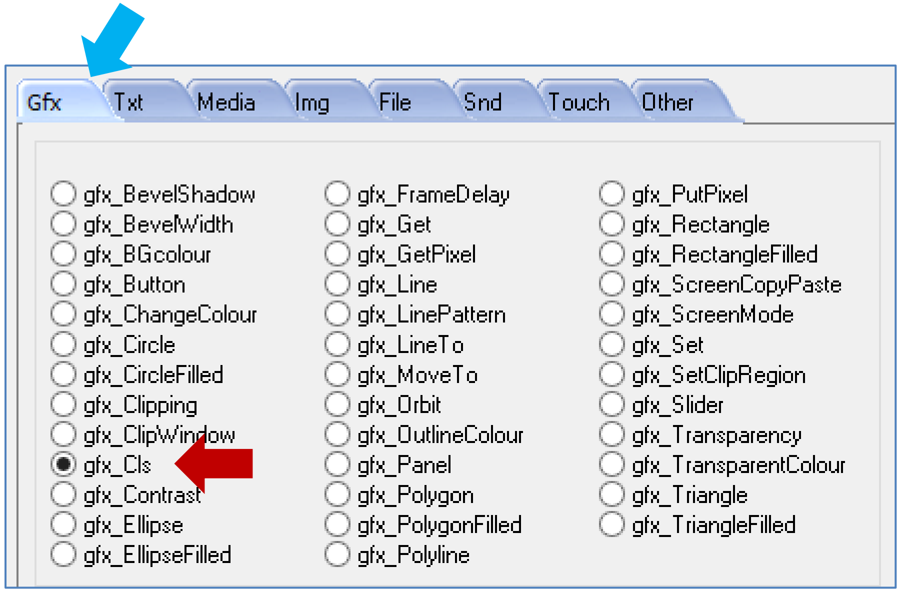
The following sections will now demonstrate the use of four serial commands.

Control the Display

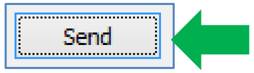
The display module is going to receive and send messages to the host. This section explains to the user how to interpret these messages. An understanding of this section is necessary for the user to be able to properly program the host controller.

Clear the Screen

In the **Serial Commander** window, click on the **Gfx** (or Graphics) tab and select **gfx_Cls**.

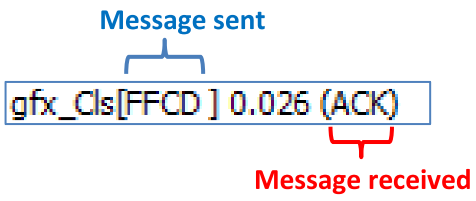


Click on the send button.



The display module is now cleared.

Also, messages are sent to and received from the display module. The value inside the brackets is the message sent to the display module. The value inside the parentheses is the message received from the display module.

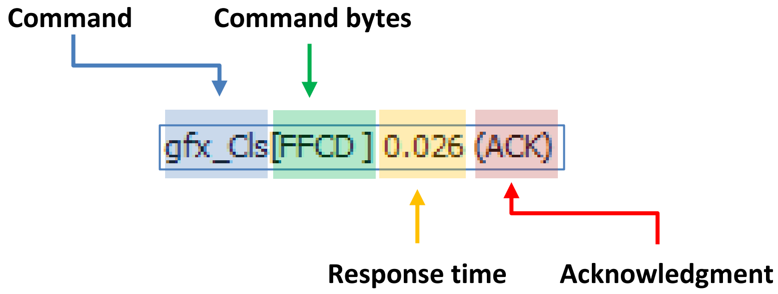


Message sent

gfx_Cls[FFCD] 0.026 (ACK)

Message received

The equivalent byte values for the **gfx_Cls** command are **0xFF** (MSB) and **0xCD** (LSB). The display module responds with an acknowledgment (**ACK**) after **0.026** seconds. The byte value for **ACK** is **0x06** and is not shown.

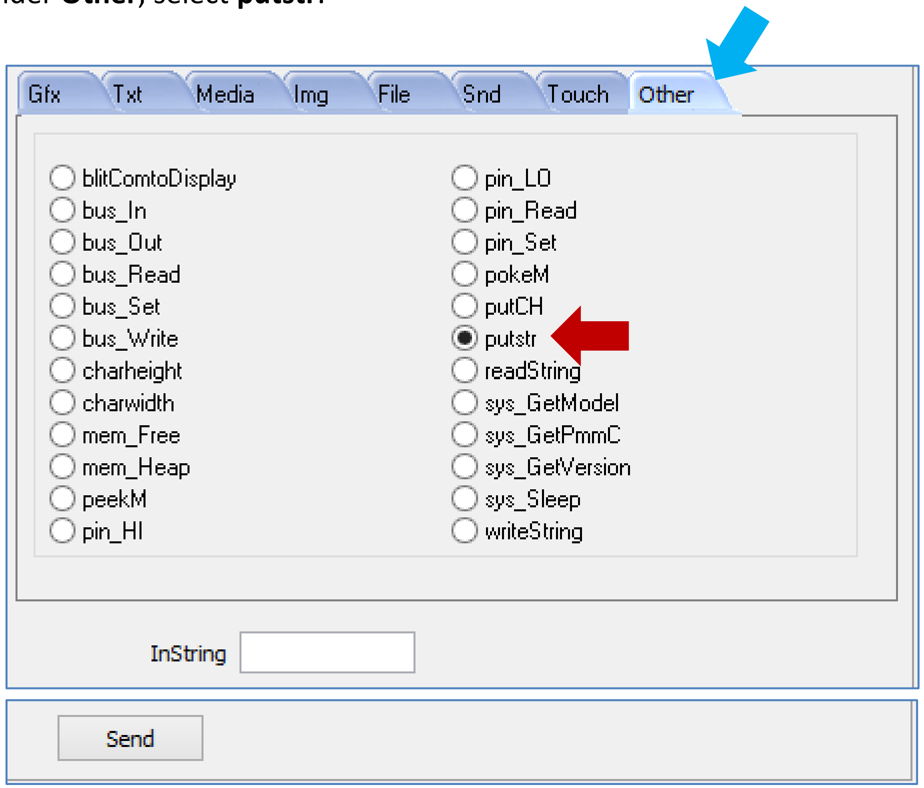


```
gfx_Cls[FFCD ] 0.026 (ACK)
```

For more information on the **gfx_Cls** command, refer to section **5.2.1 Clear Screen** of the Picaso Serial Command Set Reference Manual.

Print a String

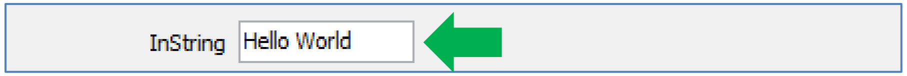
Under **Other**, select **putstr**.



○ blitComtoDisplay
○ bus_In
○ bus_Out
○ bus_Read
○ bus_Set
○ bus_Write
○ charheight
○ charwidth
○ mem_Free
○ mem_Heap
○ peekM
○ pin_HI
○ pin_LO
○ pin_Read
○ pin_Set
○ pokeM
○ putCH
● putstr
○ readString
○ sys_GetModel
○ sys_GetPmmC
○ sys_GetVersion
○ sys_Sleep
○ writeString

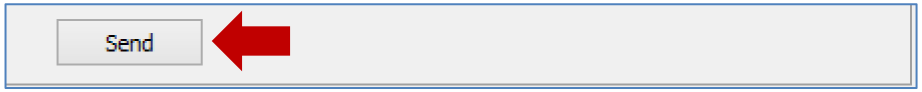
InString

Input the string "Hello World" into the InString box.

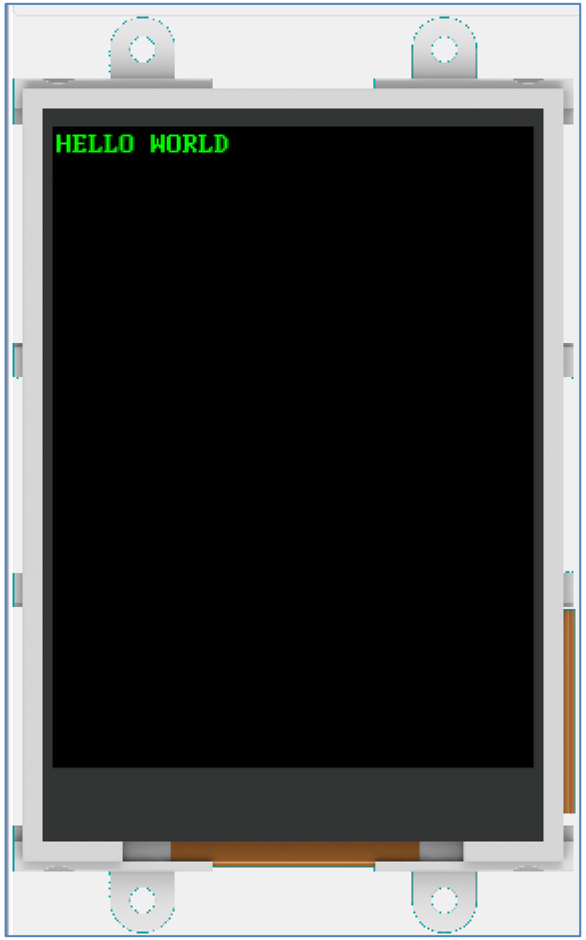


InString: Hello World

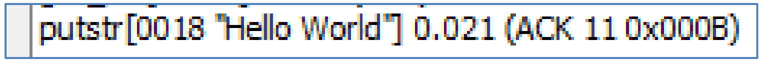
Click on the Send button.



The string “Hello World” should now be printed on the screen.



The message area displays the messages sent to and received from the display module.



```
putstr[0018 "Hello World"] 0.021 (ACK 11 0x000B)
```

The putstr command is sent with the user-defined string as the parameter. Refer to section 5.1.3 Put String of the Picaso Serial Command Set Reference Manual for more information. Experimentation with the different commands using the Serial Commander is now left to the user as an exercise.

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.

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

Disclaimer of Warranties & Limitation of Liability

4D Systems makes no warranty, either expresses 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.