



Serial Displaying Third Party Fonts

FAT16

DOCUMENT DATE: 24th May 2019
DOCUMENT REVISION: 1.1



Description

This application note explains how custom fonts can be used on a Picaso or a Diablo16 display module in the Serial environment. Here is a list of items required to replicate this application:

- Any of the following Picaso touch display modules:

[uLCD-24PTU](#) [uLCD-32PTU](#) [uLCD-43\(PT/PCT\)](#)
[uLCD-28PTU](#) [uLCD-32WPTU](#)

and other superseded touch display modules which support the ViSi environment

- The target module can also be a Diablo16 touch display

[uLCD-35DT](#) [uLCD-70DT](#)

Visit www.4dsystems.com.au/products to see the latest display module products that use the Diablo16 processor.

- [4D Programming Cable](#) or [µUSB-PA5](#)
- [micro-SD](#) (µSD) memory card
- [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

Description	2
Content	2
Application Overview	3
Setup Procedure	4
Adding Fonts	4
<i>Customizing Fonts using the ViSi environment</i>	<i>4</i>
<i>Completing the ViSi project</i>	<i>8</i>
Simulation	8
<i>Clear the Screen</i>	<i>9</i>
<i>Mount the uSD Card</i>	<i>9</i>
<i>Load the Font Files</i>	<i>10</i>
<i>Set the Font ID</i>	<i>11</i>
<i>Send a String</i>	<i>11</i>
<i>The New Line Character</i>	<i>12</i>
<i>Choose another Font</i>	<i>13</i>
<i>Print the Contents of the uSD Card</i>	<i>14</i>
<i>Unmount the uSD Card</i>	<i>16</i>
Proprietary Information	17
Disclaimer of Warranties & Limitation of Liability	17

Application Overview

There are three built in fonts in the Picaso processor:

- 0 or FONT1 = System font
- 1 or FONT2
- 2 or FONT3 = Default font

Users might need more stylish and larger size fonts which is a need addressed in this application. User can import ANSI or UNICODE fonts. The text could be displayed using **Put Character** or **Put String** command.

For the Diablo16 processor, the available system font IDs are:

- 1 for FONT_1 = System 5x7
- 2 for FONT_2 = System 8x8
- 3 for FONT_3 = System 8x12 (Default)
- 4 for FONT_4 = System 12x16
- 5 for FONT_5 = MS San Serif 8x12
- 6 for FONT_6 = Deja Vu Sans Condensed 9pt
- 7 for FONT_7 = Deja Vu Sans 9pt
- 8 for FONT_8 = Deja Vu Sans Bold 9pt
- 9 for FONT_9 = System 3x6
- 10 – Not currently available for SPE Serial, N/A
- 11 for FONT_11 = EGA 8x12 font

NOTE: The **List Filenames** command is the only other command that writes the list of directory directly to the screen. This command is also affected by the Fonts change.

This application note uses a Picaso display module as the sample model, but the procedures are also applicable to Diablo16 display modules. The main difference between a Picaso display and a Diablo16 display with respect to the Serial environment is the bytes used for the serial commands. The serial command bytes for clearing the screen and sending a string, for instance, are different for each graphics processor.

	Picaso	Diablo16
Clear screen	0xFFCD	0xFF82
Put string	0x0018	0x0018

A user with a Diablo16 display module must have to consider this fact when using this application note as a reference.

Setup Procedure

This application note, although written for Serial, requires the use of the ViSi environment to generate the necessary files which will be copied to the uSD card. The display module is then configured as a slave device by loading it with the SPE application. With the uSD card mounted onto the display, the host, which is the Serial Commander in this application note, will then be able to control the display and access the contents of the uSD card.

This application note starts with the creation of a basic ViSi project. Users who want to learn more about the ViSi environment may consult the application note

[ViSi Getting Started - First Project for Picaso and Diablo16](#)

Topics discussed include instructions on how to launch Workshop 4, how to open a ViSi project, how to change the target display, how to create a new ViSi project, how to save a ViSi project, how to connect the target display to the PC, and how to compile and download a program.

Adding Fonts

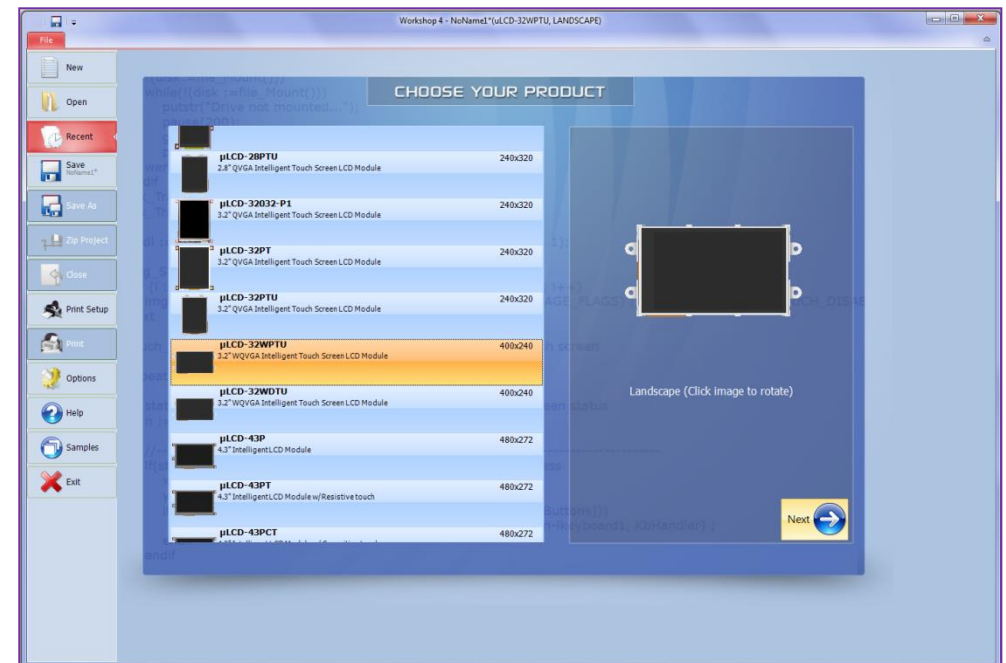
Customizing Fonts using the ViSi environment

Open the Workshop 4 (WS4) IDE and click "Create a new project".



Choose the Display module you want to use. A uLCD-32WPTU will be used for this application.

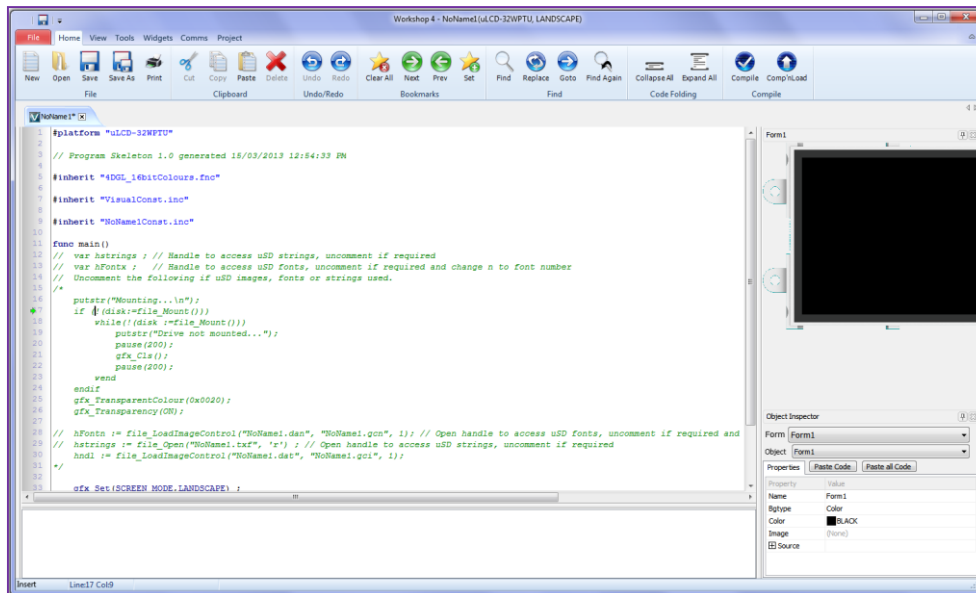
Note: Orientation of the Display is irrelevant here.



Select the **ViSi** environment

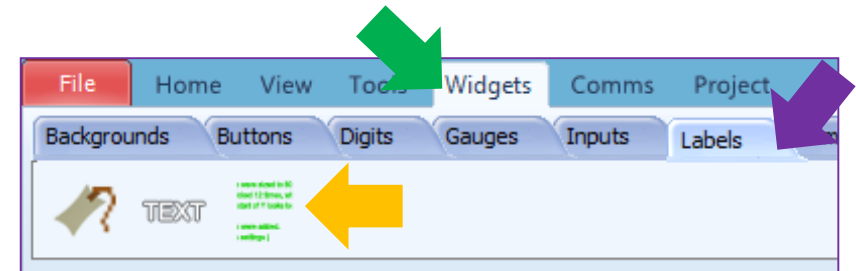


This will open the ViSi development environment window within the WS4 IDE as shown below.

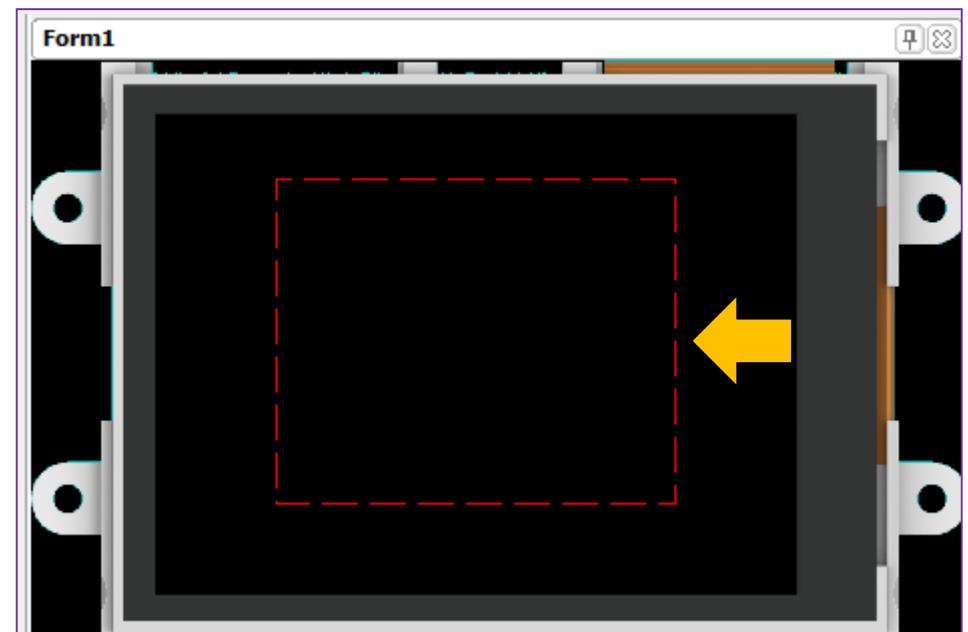


The new project starts as 'noname' project. Save the project as, say AddFont.

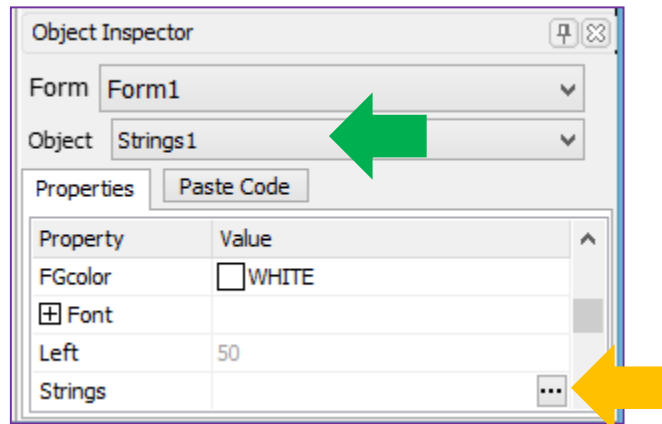
Go to **Widgets**, select **Strings** object under the **Labels** tab.



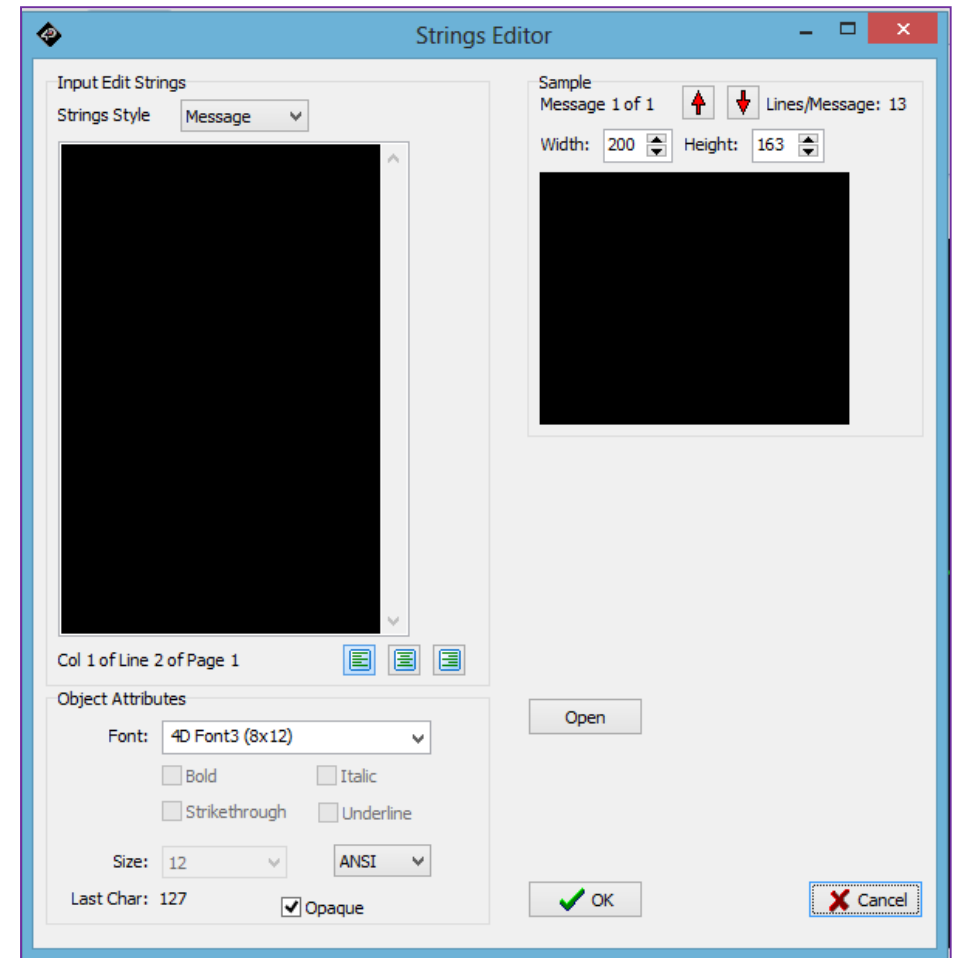
Click on to the WYSIWYG screen to drop the string object. A string object has now been created. This is Strings1.



Click on the ellipsis dots of the **Strings** property of the Object Inspector.



The Strings Editor window appears. Select the font and adjust other properties as required.

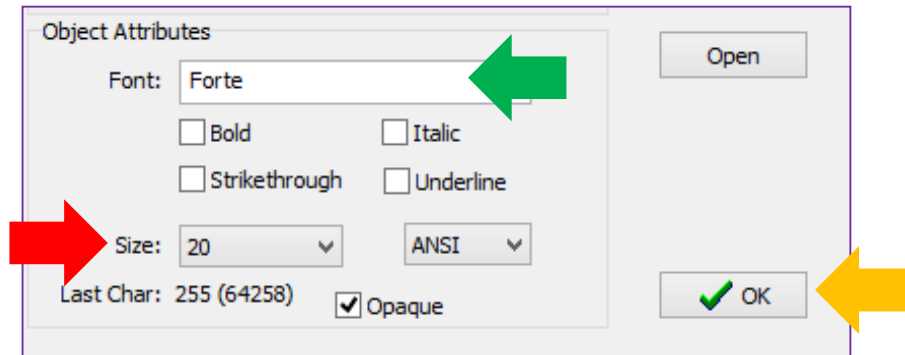


For this application, we have only adjusted the following properties,

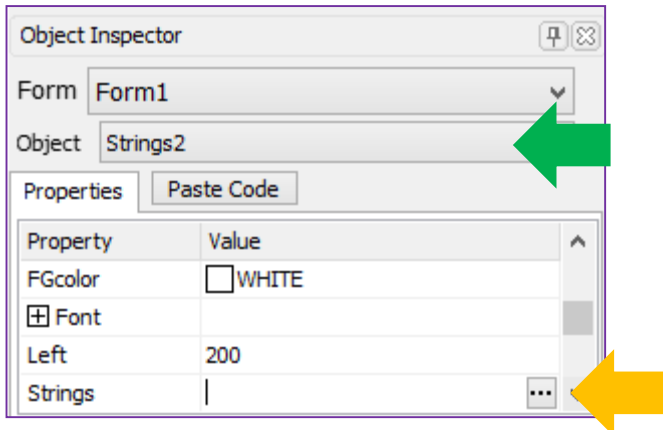
Set Fonts to 'Forte'

Font size is set to 20

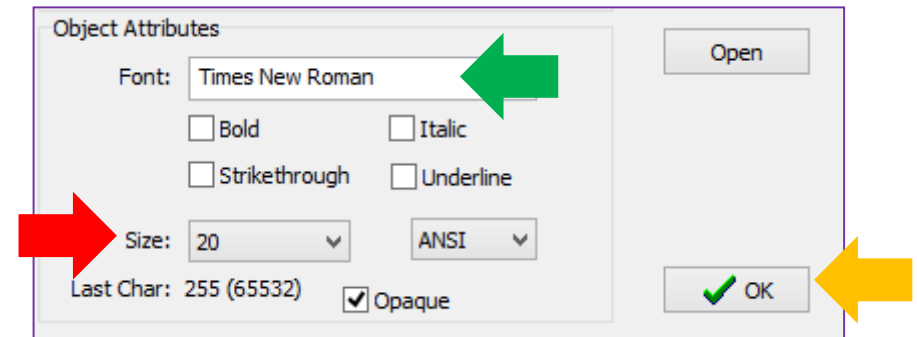
Press OK.



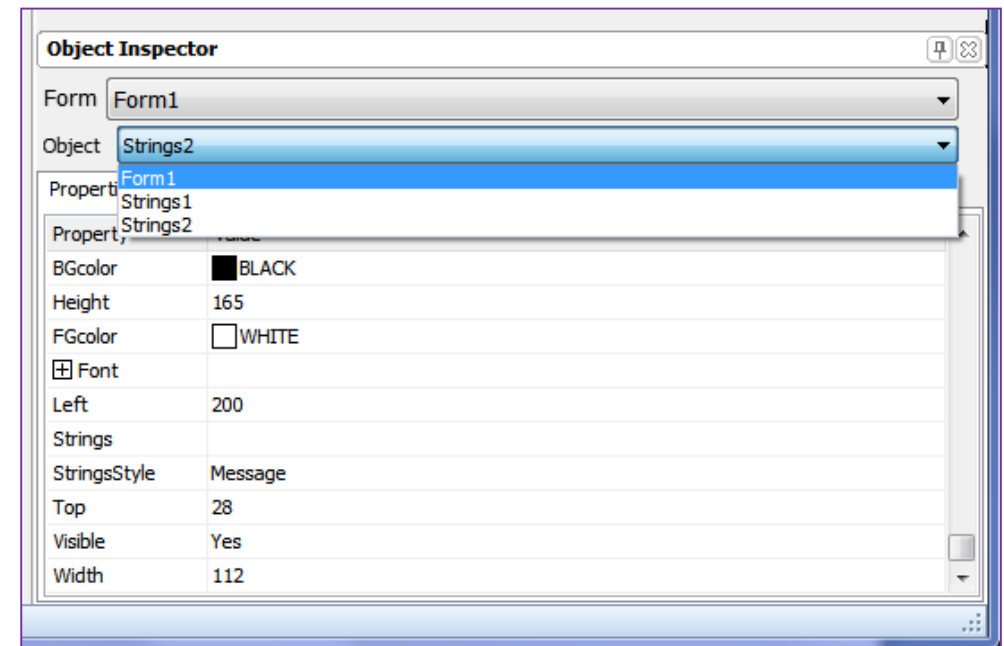
Create another string object. This object will be Strings2.



Set the properties in the Strings Editor window. For Strings2, the properties are:

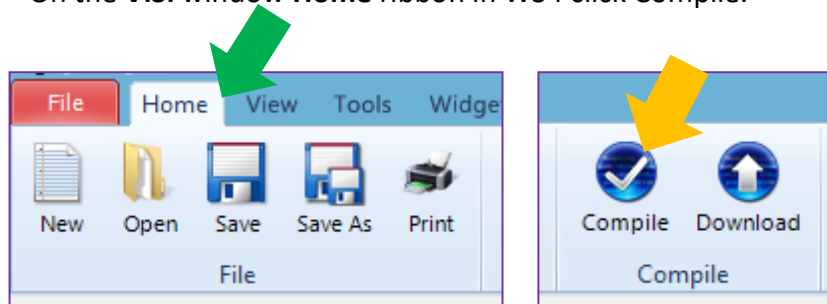


Now, there should be two strings in the object inspector.

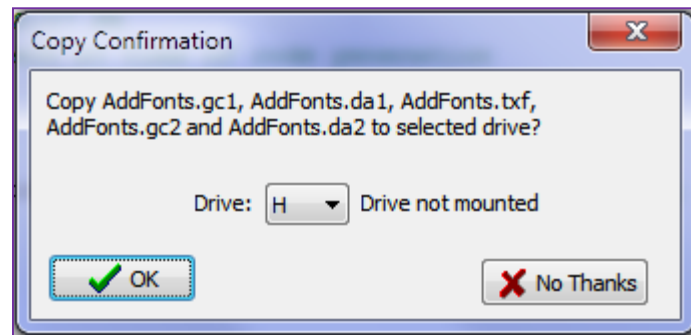


Completing the ViSi project

- Now, insert the uSD card in the uSD card reader on the PC/Notebook.
- Make sure the uSD card is FAT (aka FAT16) formatted.
- On the **ViSi** window **Home** ribbon in WS4 click Compile.



After clicking on **Compile**, you will be prompted to select and confirm the drive to where the font files will be saved. Click **OK** to confirm and start copying.



Press OK to start writing the Font files to the uSD card. Once the writing is completed, remove the uSD card from the PC/Notebook and insert it in to the display module.

Simulation

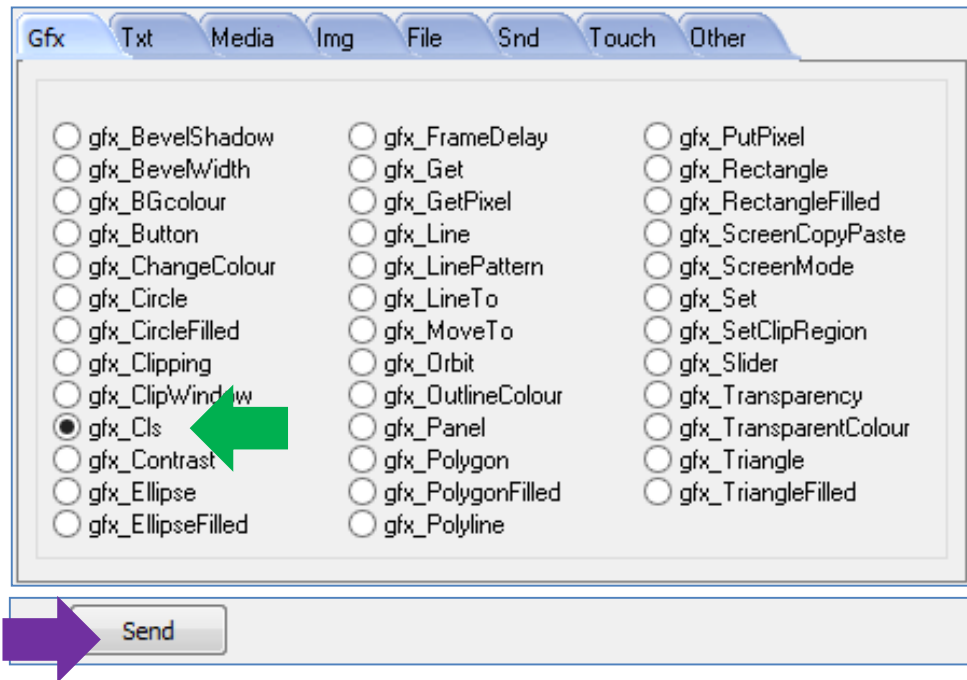
The display must be configured as a slave device first before it can be controlled by a host. For instructions on how to launch Workshop 4, how to connect the display module to the PC, and how to configure the display as a slave device, kindly refer to the section “**Setup Procedure**” of any of the application notes below. Choose according to your display module’s processor.

[Serial Picaso Getting Started - the SPE Application](#)

[Serial Diablo16 Getting Started - the SPE Application](#)

These application notes also introduce the user to the Serial Protocol through the use of the Serial Commander

Clear the Screen



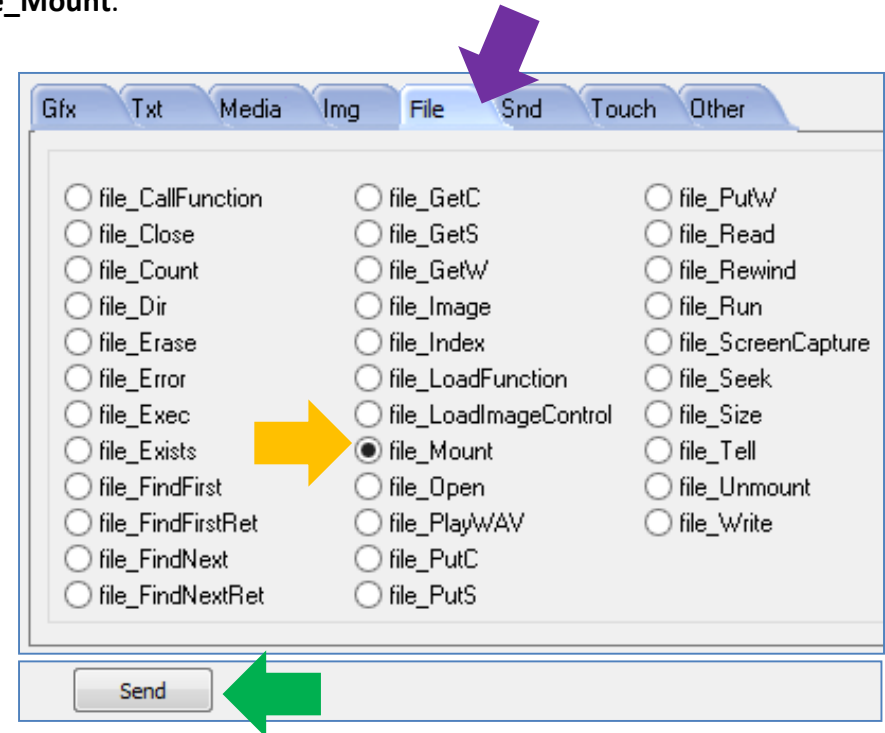
The bytes sent and received are:

```
gfx_Cls[FFCD ] 0.020 (ACK)
```

The screen should now be cleared.

Mount the uSD Card

Mount the uSD card on to the display module. Go to the **File** tab and select **file_Mount**.

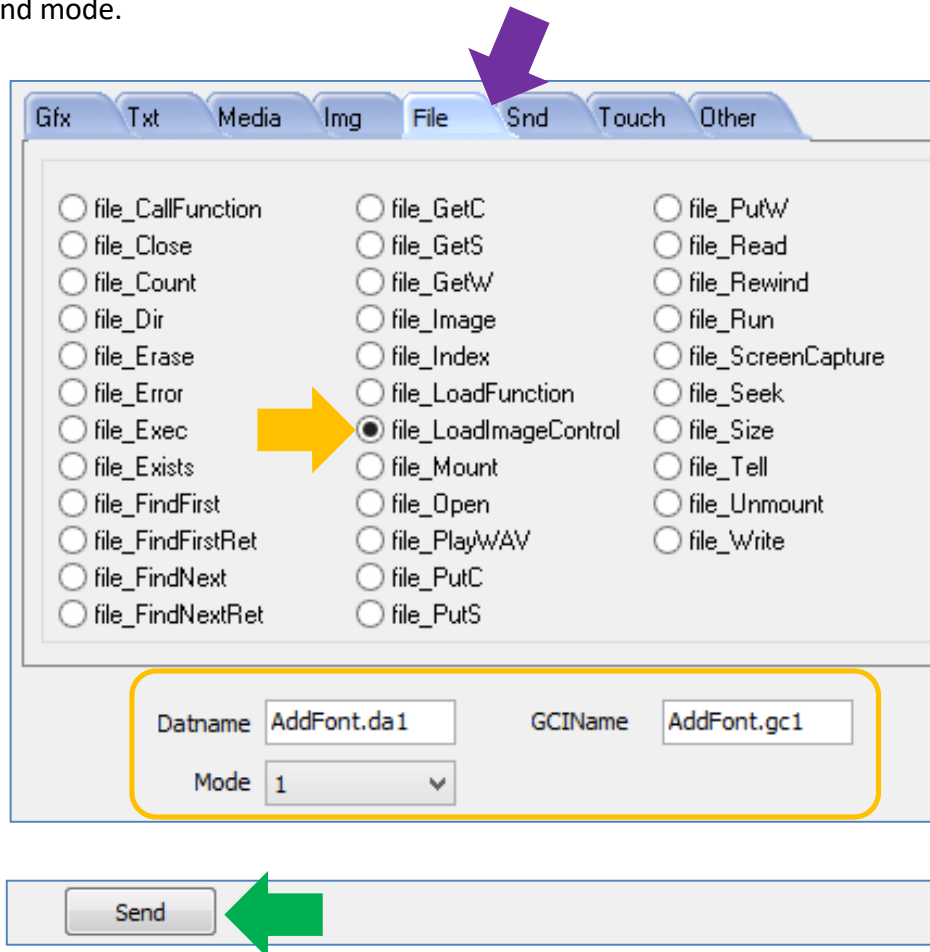


The bytes sent and received are:

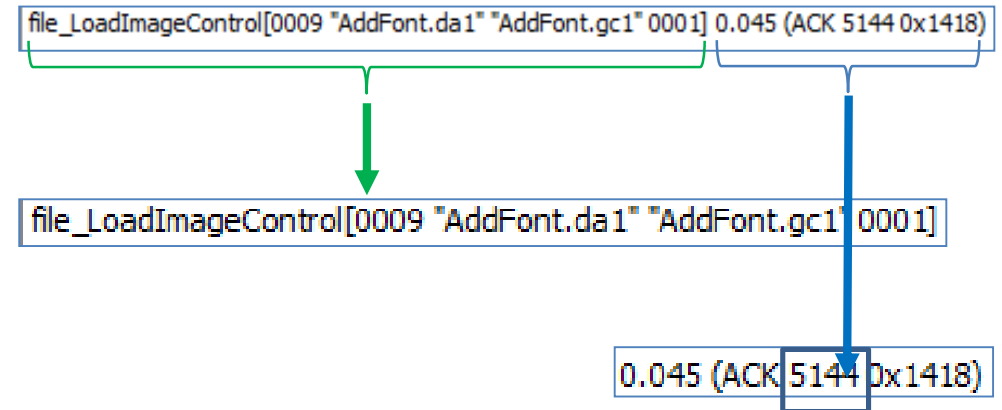
```
file_Mount[FF03 ] 0.110 (ACK 5447 0x1547)
```

Load the Font Files

In the **File** tab, select **file_LoadImageControl** then enter the font file names and mode.



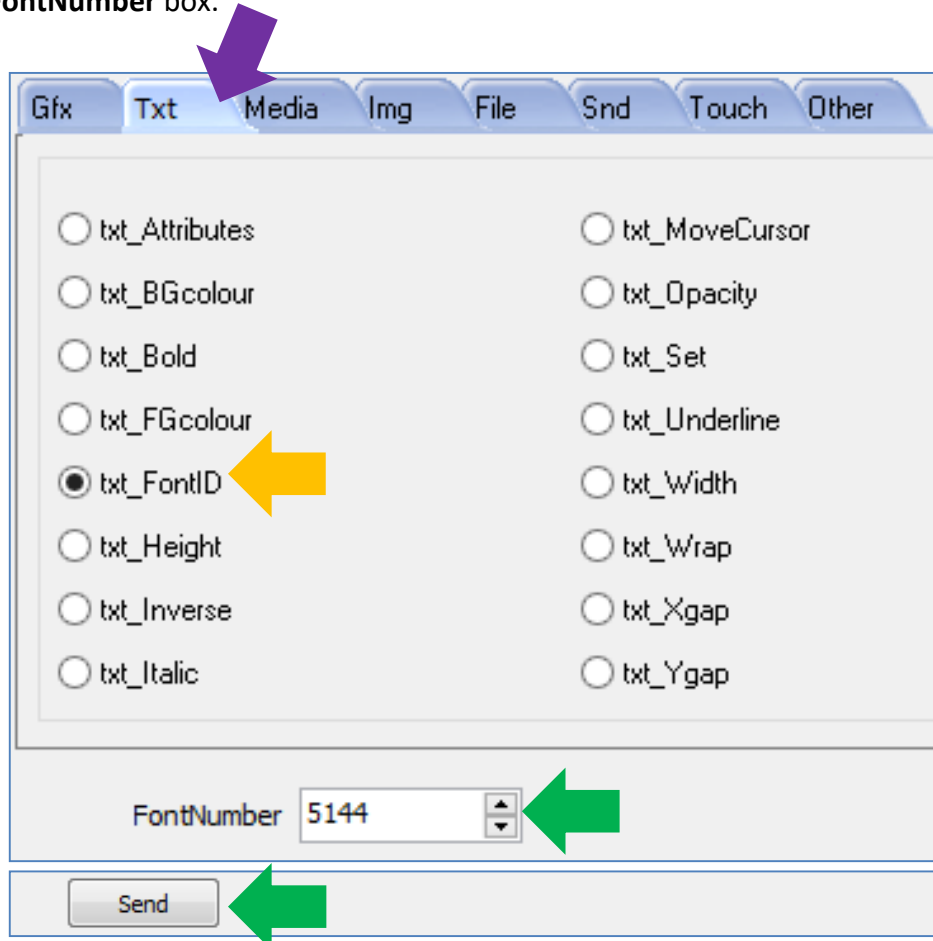
The bytes sent and received are:



Take note of the reply from the display – “5144” in decimal or “0x1418” in hexadecimal. This is the 16-bit file handle. **It could be different in your case.**

Set the Font ID

Go to the **Txt** tab and select **txt_FontID**. Enter the file handle into the **FontNumber** box.

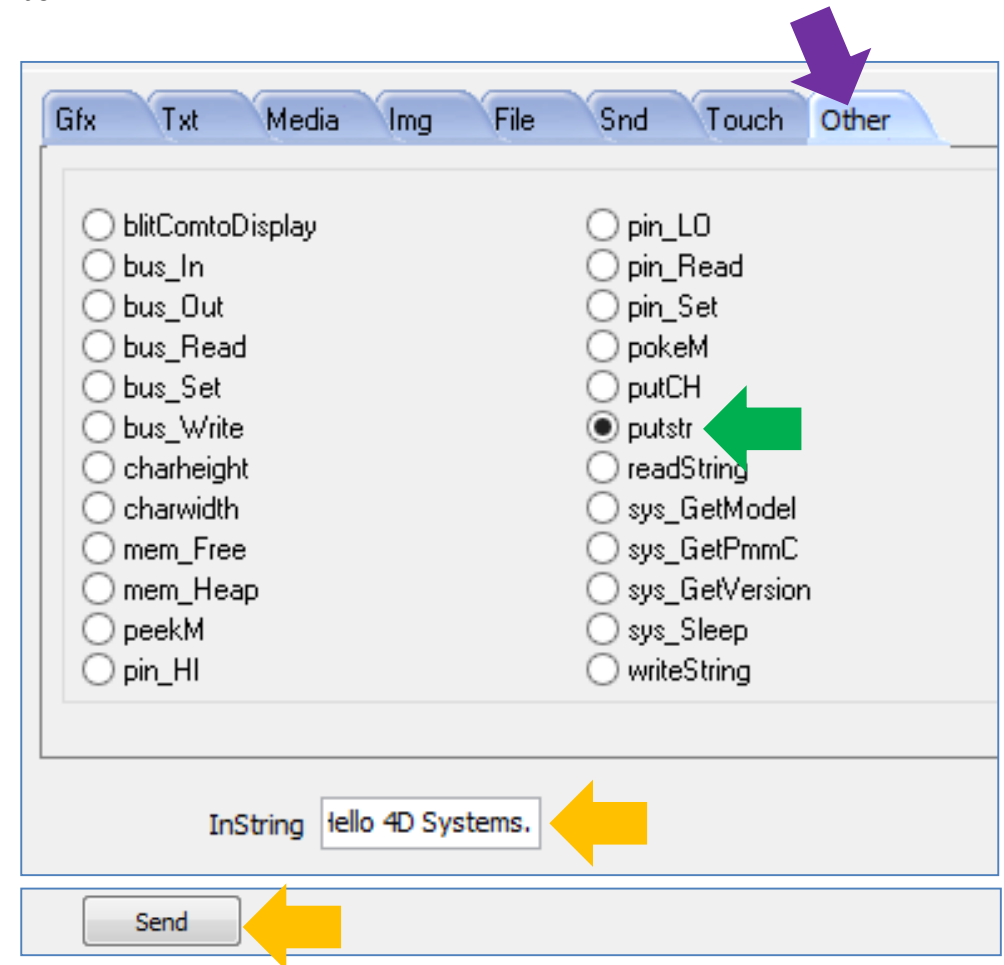


The bytes sent and received are:

```
txt_FontID[FFE5 1418] 0.009 (ACK 2 0x0002)
```

Send a String

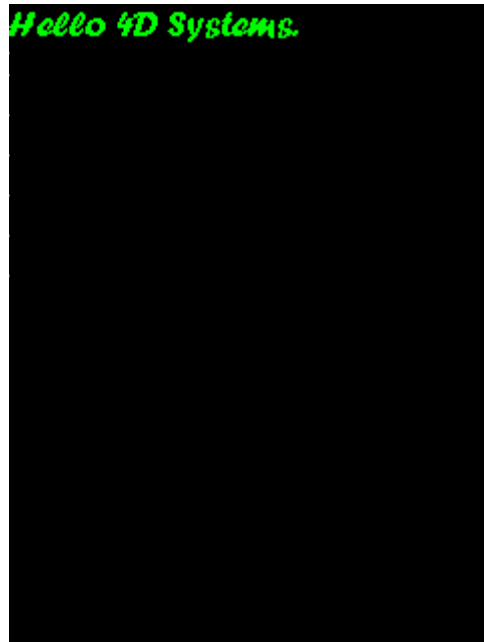
Go to the **Other** tab and select **putstr**. Input "Hello 4D Systems." to the **InString** box.



The bytes sent and received are:

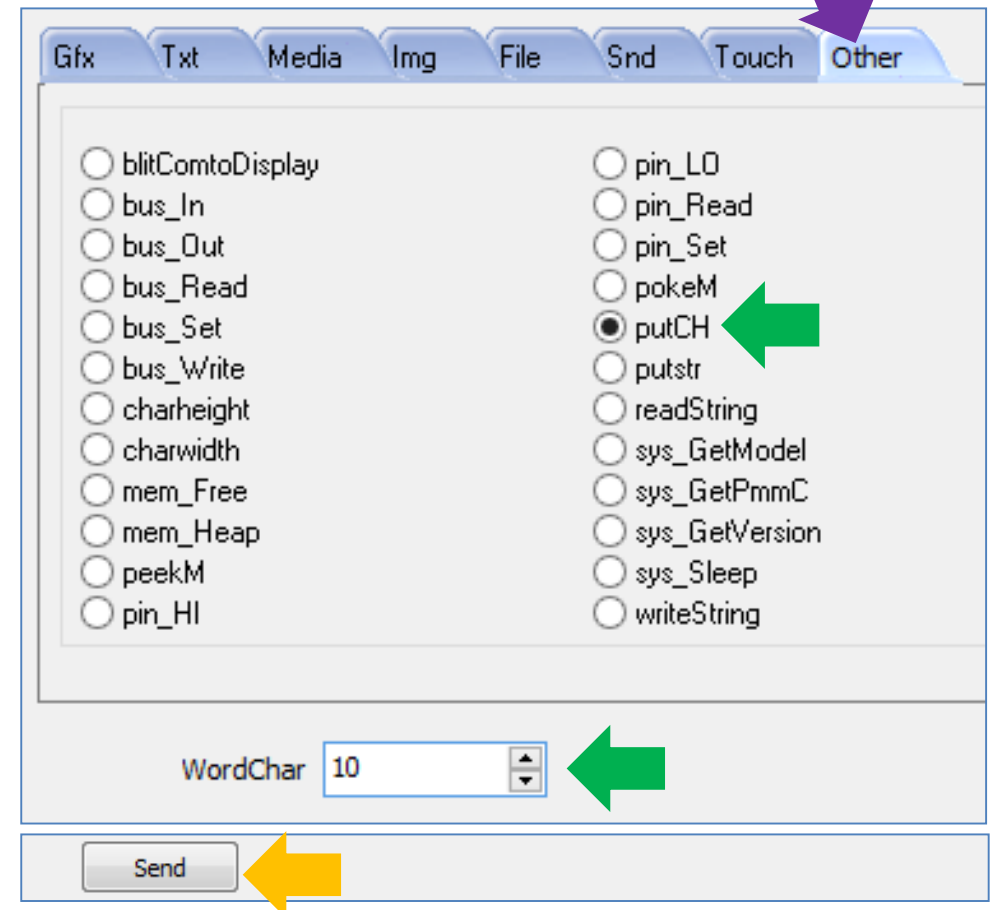
```
putstr[0018 "Hello 4D Systems." ] 0.047 (ACK 17 0x0011)
```

The string "Hello 4D Systems." should now be shown on the screen.



The New Line Character

The ASCII code for the new line character is 0x0A in hexadecimal or 10 in decimal. In the tab **Other**, select **putCH**. Input the value "10" to the **WordChar** box and click on the send button.

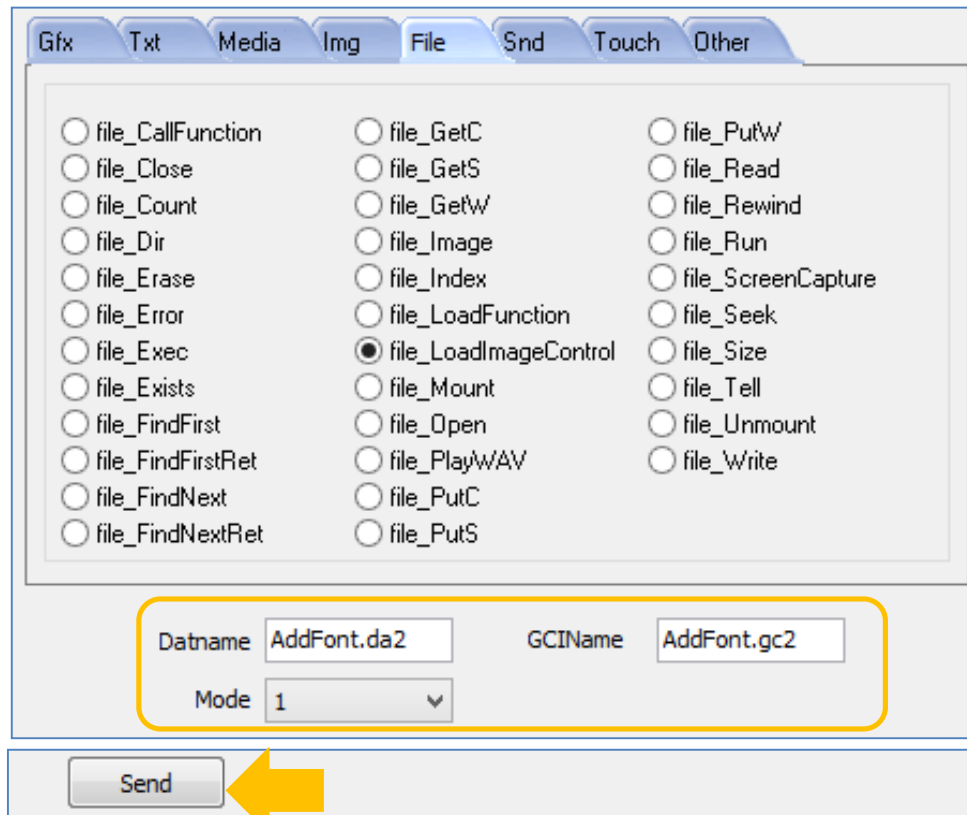


The bytes sent and received are:

```
putCH[FFFE 000A] 0.007 (ACK)
```

Choose another Font

Choose a new font by loading the correct files.

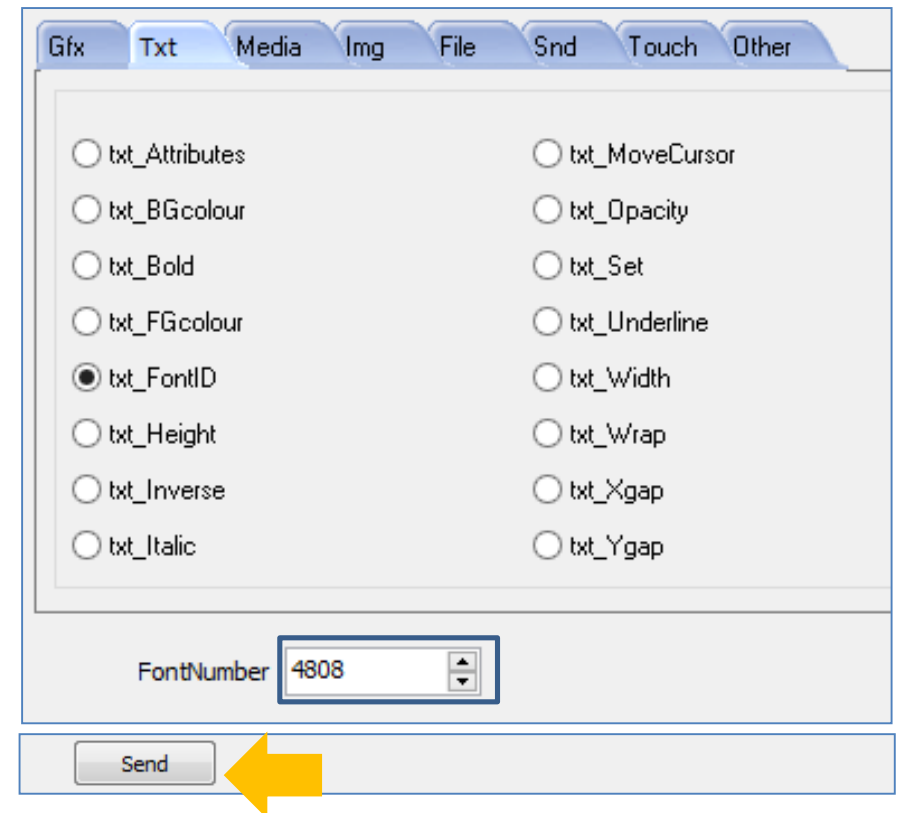


The bytes sent and received are:

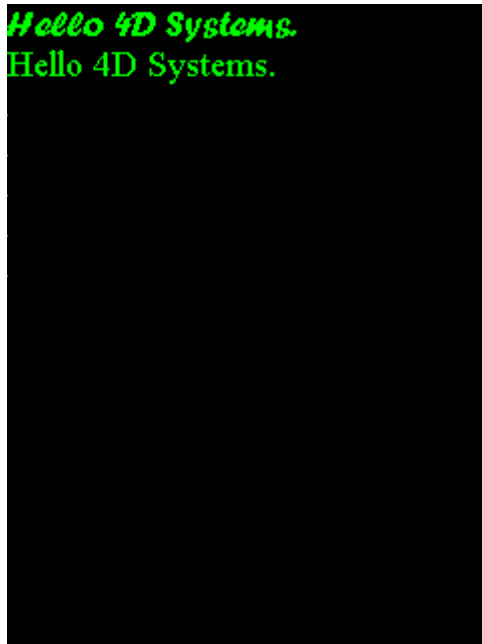
```
file_LoadImageControl[0009 "AddFont.da2" "AddFont.gc2" 0001] 0.045 (ACK 4808 0x12C8)
```

```
0.045 (ACK 4808 0x12C8)
```

Again, take note of the returned value, which is the 16-bit file handle. This value may be different in your case. Use this value when setting the new font ID.

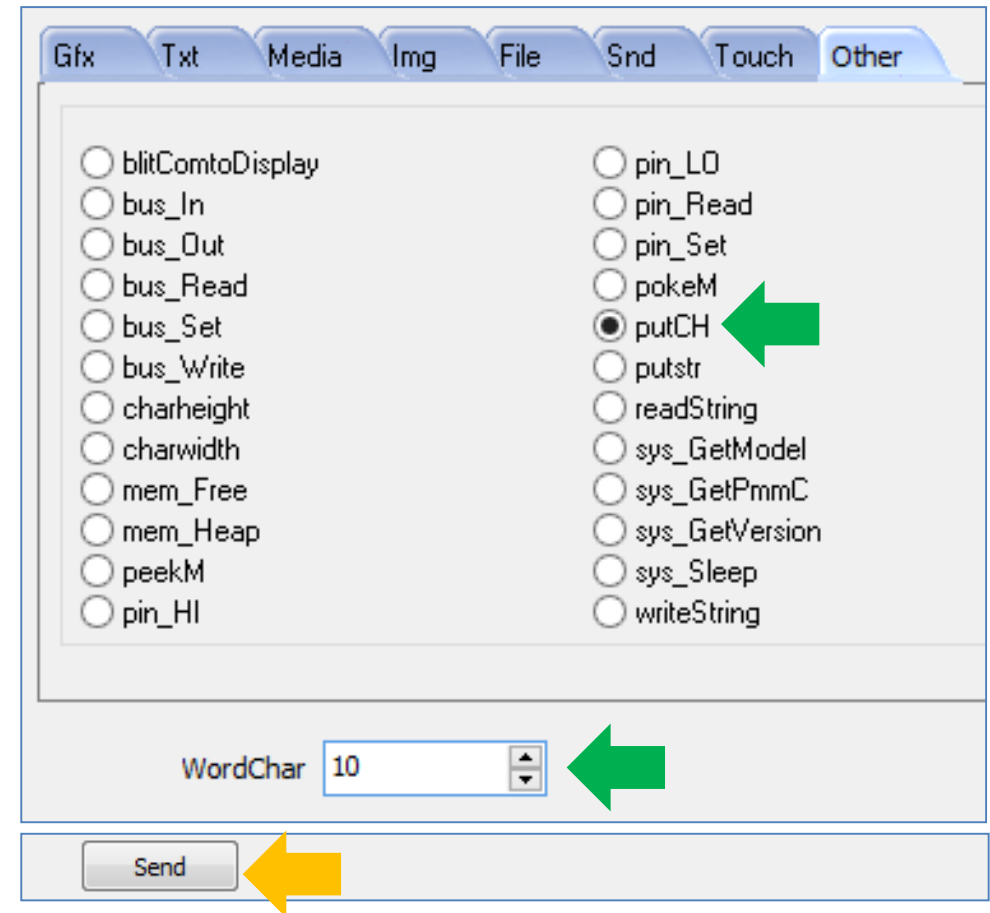


When successful, the new string should now be in a new line and using the new font.

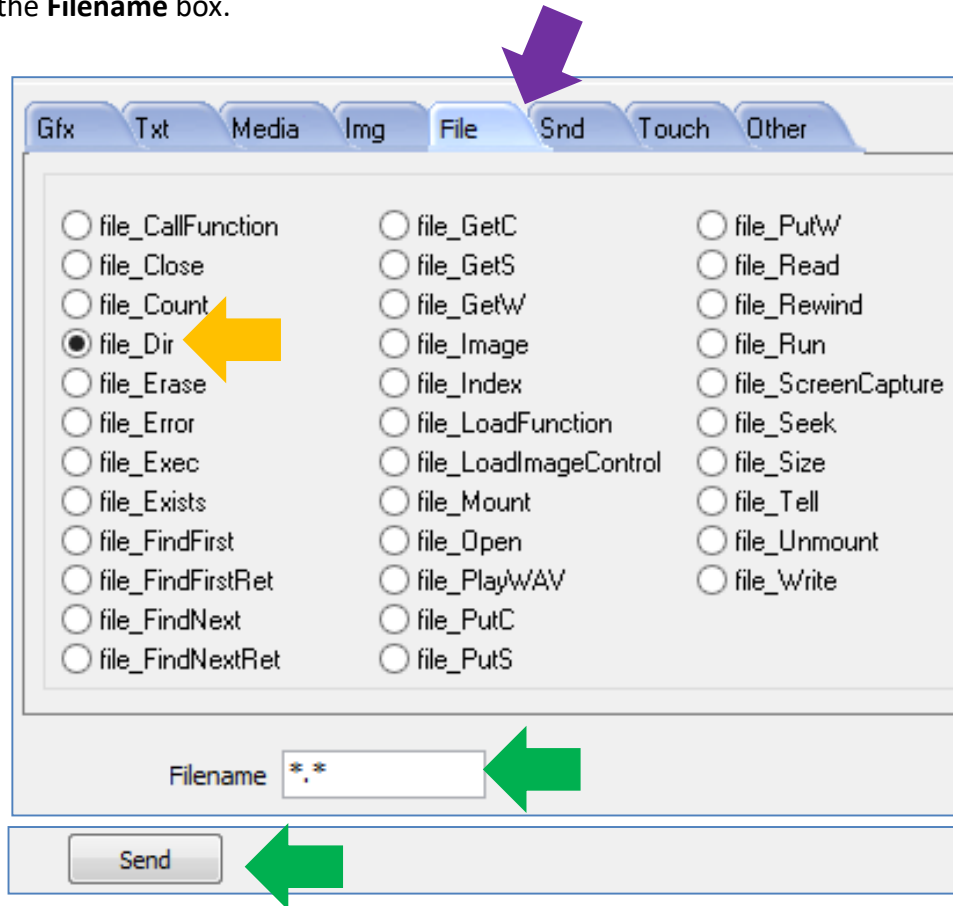


Print the Contents of the uSD Card

Send another new line character to the display.



Go to the **File** tab and select **file_Dir**. Input the wildcard character **"*.*"** to the **Filename** box.



The bytes sent and received are:

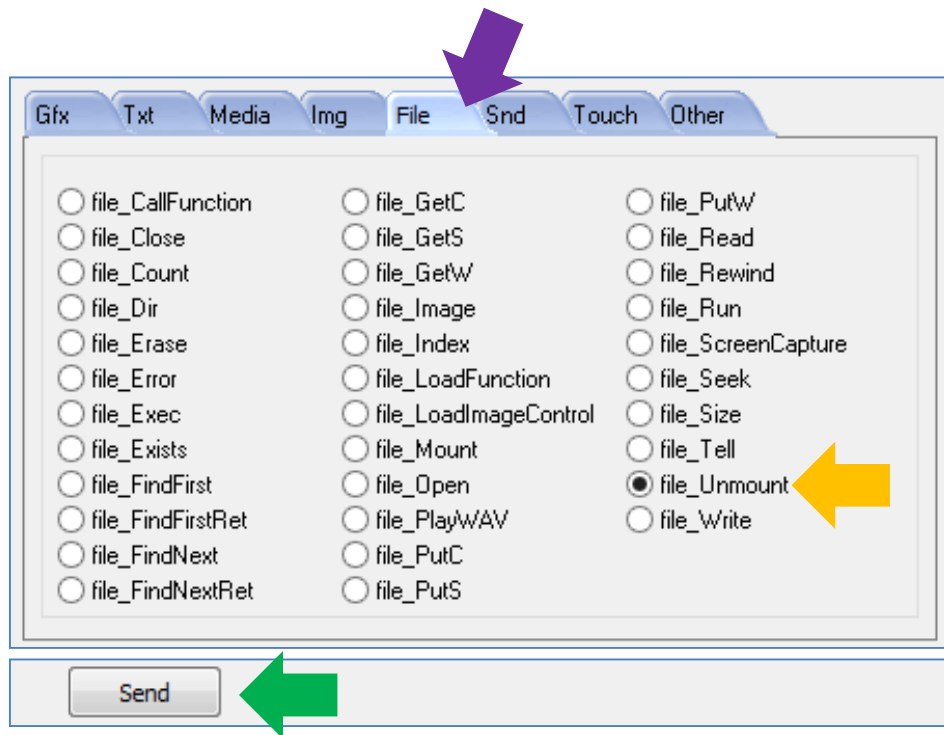
```
file_Dir[0002 "*. *"] 0.096 (ACK 5 0x0005)
```

The file names should now be printed on the display.

```
Hello 4D Systems.
Hello 4D Systems.
ADDFONT.GC1
ADDFONT.DA1
ADDFONT.TXF
ADDFONT.GC2
ADDFONT.DA2
```

Unmount the uSD Card

To properly unmount the uSD card, go to the **File** tab and select **file_Unmount**. Then click on the send button.



The bytes sent and received are:

```
file_Unmount[FF02] 0.004 (ACK)
```


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.