



Designer or ViSi Playing Sound

DOCUMENT DATE: **7th MAY 2020**
DOCUMENT REVISION: **1.1**



Description

This application note presents and discusses a Designer project for playing a WAV audio file. The 4DGL code of the Designer project can be copied and pasted to an empty ViSi project and it will compile normally. The code can also be integrated to that of an existing ViSi project.

Before getting started, the following are required:

- Any of the following 4D Picaso display modules:

<u>gen4-uLCD-24PT</u>	<u>gen4-uLCD-28PT</u>	<u>gen4-uLCD-32PT</u>
<u>uLCD-24PTU</u>	<u>uLCD-28PTU</u>	<u>uVGA-III</u>

and other superseded modules which support the ViSi Genie environment

- The target module can also be a Diablo16 display

<u>gen4-uLCD-24D series</u>	<u>gen4-uLCD-28D series</u>	<u>gen4-uLCD-32D series</u>
<u>gen4-uLCD-38D series</u>	<u>gen4-uLCD-43D series</u>	<u>gen4-uLCD-50D series</u>
<u>gen4-uLCD-70D series</u>		
<u>uLCD-35DT</u>	<u>uLCD-43D Series</u>	<u>uLCD-70DT</u>

Visit www.4dsystems.com.au/products to see the latest display module products that use the Diablo16 processor. The display module used in this application note is the uLCD-32PTU, which is a Picaso display. This application note is applicable to Diablo16 display modules as well.

- [4D Programming Cable / \$\mu\$ USB-PA5/uUSBPA5-II](#) for non-gen4 displays (uLCD-xxx)
- [4D Programming Cable & gen4-IB / 4D-UPA / gen4-PA](#) for gen4 displays (gen4-uLCD-xxx)
- [micro-SD \(\$\mu\$ SD\)](#) 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	3
Application Overview	3
Setup Procedure	4
Create a New Project	4
Design the Project	4
<i>Preparing the μSD Memory Card</i>	<i>4</i>
<i>Loading an Appropriate WAV File</i>	<i>5</i>
<i>Essential commands for Producing Audio</i>	<i>5</i>
<i>Potential Issues</i>	<i>5</i>
<i>Functions to Combat Issues</i>	<i>5</i>
<i>Complete Application Example</i>	<i>6</i>
<i>Build an External Speaker Circuit (for gen4 Displays)</i>	<i>6</i>
Run the Program	7
Proprietary Information	8
Disclaimer of Warranties & Limitation of Liability	8

Application Overview

The Picaso and Diablo16 are range of display modules equipped with a small speaker and an audio amplifier chip, which is capable of playing a WAV file from the uSD card. This application note will walk through the necessary steps to playback audio from a WAV file source on a 4D Picaso or Diablo16 display module. Once it is known how to produce these sounds, they can then be linked to such actions as a touch, received serial data, or when an image is displayed.



Setup Procedure

For instructions on how to launch Workshop 4, how to open a **Designer** project, and how to change the target display, kindly refer to the section “**Setup Procedure**” of the application note

[Designer Getting Started - First Project](#)

For instructions on how to launch Workshop 4, how to open a **ViSi** project, and how to change the target display, kindly refer to the section “**Setup Procedure**” of the application note

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

Create a New Project

For instructions on how to create a new **Designer** project, please refer to the section “**Create a New Project**” of the application note

[Designer Getting Started - First Project](#)

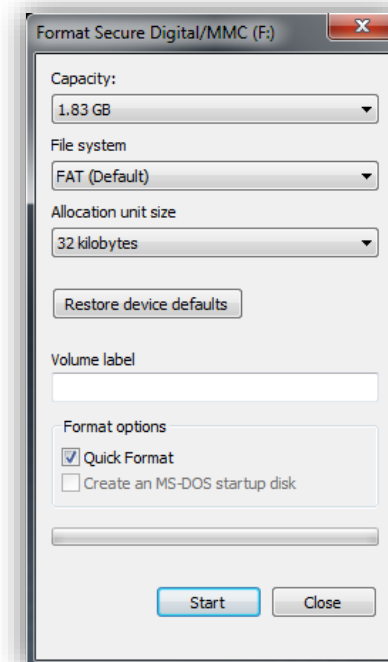
For instructions on how to create a new **ViSi** project, please refer to the section “**Create a New Project**” of the application note

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

Design the Project

Preparing the μ SD Memory Card

In order to play a custom sound through a module, a μ SD card must first be sourced for storing the WAV file. The first step is to format the μ SD to FAT16 or FAT (Default). Insert the card into a PC and right click on the μ SD card and select **Format**. Leave all settings as they appear, or change them accordingly as shown in the snapshot below. Press the **Start** button and the card will now be formatted.



Loading an Appropriate WAV File

Now that the μ SD card is formatted, it is ready to have the desired WAV file loaded onto it. Use any audio editing software to save the sound as a 16kHz Mono WAV file to your μ SD card. Try and get as close to this frequency as possible, however, the extreme most thresholds are 22kHz and 12kHz. Anything outside these boundaries won't work. Insert the μ SD card into the module and open Workshop to begin code development.

Essential commands for Producing Audio

To play a sound file at the most basic level, all that is required is mounting the drive and playing the WAV file. The following extract shows the necessary commands:

```
file_Mount();  
file_PlayWAV("filenameWAV");
```

Potential Issues

However, these commands alone are prone to improper use and potential errors. It is recommended that the following issues be addressed when dealing with these commands.

- μ SD not inserted
- File not on the μ SD card
- Incorrect volume level
- Not enough time to play the whole duration of the file

Functions to Combat Issues

Here is a list of functions available to combat these problems:

- file_Mount()
- file_Exists(fname)
- snd_Volume(var)
- snd_Playing()

Simple Example

The next snippet of code takes into account such possible issues:

```
while(!file_Mount())  
wend  
  
if(file_Exists("filenameWAV"))  
    snd_Volume(100);  
    file_PlayWAV("filenameWAV");  
    while(snd_Playing());  
endif
```

Complete Application Example

Finally, a fully scripted example is shown that includes error messages for any potential issues that may be encountered:

```
#platform "uLCD-24PTU"

/*****
* Filename: Sound.4dg
* Created: 3rd November 2011
* Author: 4D team
* Description: playing a WAV sound
*****/
#inherit "4DGL_16bitColours.fnc"

func main()
repeat

txt_Set(FONT_SIZE, FONT1);
putstr("Mounting...");
while(!file_Mount())
putstr("Drive not mounted...");
pause(200);
gfx_Cls();
pause(200);
wend
print("Drive mounted.");

if(file_Exists("CHIMESWAV"))
snd_Volume(100);
file_PlayWAV("CHIMESWAV");
while(snd_Playing());
else
putstr("CHIMESWAV not found");
endif

forever
endfunc
```

For detailed instructions on how to manipulate sound with your Picaso or Diablo16 device, please refer to the separate documents:

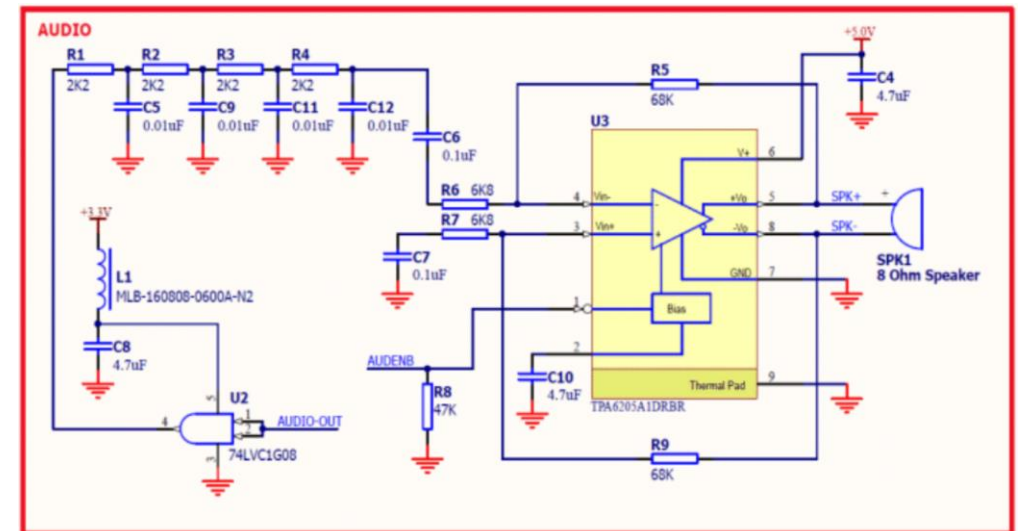
[Picaso Internal Functions Manual](#)

or

[Diablo16 Internal Functions Manual.](#)

Build an External Speaker Circuit (for gen4 Displays)

Unlike the non-gen4 modules, the gen4 Displays do not have an on board speaker. Users may consult this circuit as a reference for building their own.



Run the Program

For instructions on how to save a **Designer** project, how to connect the target display to the PC, how to select the program destination, and how to compile and download a program, please refer to the section “**Run the Program**” of the application note

[Designer Getting Started - First Project](#)

For instructions on how to save a **ViSi** project, how to connect the target display to the PC, how to select the program destination, and how to compile and download a program, please refer to the section “**Run the Program**” of the application note

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

The uLCD-32PTU and uLCD-35DT display modules are commonly used as examples, but the procedure is the same for other displays.

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.