



# Serial Arduino Displaying Images from the uSD Card RAW

DOCUMENT DATE: **22<sup>nd</sup> April 2019**  
DOCUMENT REVISION: **1.1**



## Description

This application note explores the possibilities provided by the Serial environment in Workshop for a Goldelox display module to work with an Arduino host. In this example, the host is an Arduino Uno board. The host can also be an Arduino Mega 2560 or Due. Ideally, the application described in this document should work with any Arduino board that supports software serial or with at least one UART serial port. [See specifications of Arduino boards here.](#)

Before getting started, the following are required:

- Any of the following 4D Goldelox display modules:

<a href="#">uOLED-96-G2</a>	<a href="#">uOLED-128-G2</a>	<a href="#">uOLED-160-G2</a>
<a href="#">uLCD-144-G2</a>		

or any superseded module that supports the Serial environment. Visit [www.4dsystems.com.au](http://www.4dsystems.com.au) to see the latest products using the Goldelox graphics processor.

- [4D Programming Cable](#) or [uUSB-PA5](#)
- [Workshop 4 IDE](#) (installed according to the installation document)
- [micro-SD \(uSD\)](#) memory card
- [4D Arduino Adaptor Shield](#) (optional) or connecting wires
- [Arduino IDE](#)

- 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

<b>Description</b> .....	<b>2</b>
<b>Content</b> .....	<b>3</b>
<b>Application Overview</b> .....	<b>3</b>
<b>Setup Procedure</b> .....	<b>4</b>
<b>Program the Arduino Host</b> .....	<b>4</b>
<i>Set the Address of an Object</i> .....	<i>5</i>
<i>Show the Object</i> .....	<i>5</i>
<b>Connect the Arduino Host to the Display Module.</b> .....	<b>6</b>
<b>Proprietary Information</b> .....	<b>7</b>
<b>Disclaimer of Warranties &amp; Limitation of Liability</b> .....	<b>7</b>

## Application Overview

This application note shows how to program an Arduino host so that it can control a Goldelox display module (acting as a slave device) and make it display images. The images are contained in a graphics file saved on a  $\mu$ SD card mounted on the display module. The Arduino host sends serial commands to the display module to access and display the images. The graphics file is of a format that is readable by 4D-LAB's processors. This graphics file is generated using the Workshop 4 IDE.

## Setup Procedure

The following are the steps involved in creating this project:

1. **Generate the graphics file** using Workshop 4 ViSi. Copy the generated graphics file to the uSD card and mount the uSD card to the display module.
2. Configure the display module as a serial slave device by **loading it with the SPE application**.
3. **Program the Arduino host**. Here the Goldelox-Serial-Arduino library will be used in writing the sketch for the Arduino host.
4. **Connect the Arduino host to the display module**. The Arduino board will be the master. By communicating with the slave display module, it will be able to access and display the images stored inside the graphics file on the uSD card of the display module.

For steps 1 and 2, refer to the application note [Serial Displaying Images from the uSD Card RAW](#). Serial Displaying Images from the uSD Card RAW shows how the Serial Commander is used to access and show the images saved on the uSD card of the display module.

For step 4, refer to the application note [Serial Connection to an Arduino Host](#).

## Program the Arduino Host

A thorough understanding of the application note [Serial Connection to an Arduino Host](#) is required before attempting to proceed further beyond this point. [Serial Connection to an Arduino Host](#) provides the basic information that a user needs to be able to get started with the Serial Environment and Arduino. The following is a list of the topics discussed in [Serial Connection to an Arduino Host](#).

- How to download and install the Serial-Arduino library (choose a library according to your display module's processor)
- How to modify the library for Arduino Due (due to a Due bug reported by a forum user)
- How to define the serial port to be used for talking to the display
- How to set the baud rate
- The Error Handling Routine
- How to set the Timeout Limit
- How to reset the Arduino Host and the Display
- How to let the Display Start Up
- How to set the Screen Orientation
- How to Clear the Screen
- The uSD Card Mount Routine
- How to enable message logging to the Serial Monitor of the Arduino IDE

Discussion of any of these topics is avoided in other Serial-Arduino application notes unless necessary. Users are encouraged to read [Serial Connection to an Arduino Host](#) first.

### Set the Address of an Object

Use the method “**media\_SetAdd(...)**” to be able to access an object inside the uSD card. An object can be accessed by using its 32-bit address in the uSD card. The prototype for **media\_SetAdd(...)** is

```
void media_SetAdd(word Hiword, word Loword);
```

where the argument “Hiword” is the high word of the 32-bit address, and the argument “Loword” is the low word of the 32-bit address. To access **Image1** for example, write

```
Display.media_SetAdd(0, 0);
```

To access **Image2**, write

```
Display.media_SetAdd(0, 0x2000);
```

Note that the addresses of the objects are saved as defined constants in an include file of the ViSi project. For more information, see [Serial Displaying Images from the uSD Card RAW](#).

### Show the Object

Immediately after having set the address of an object, use the method “**media\_Image(...)**” to show it. The prototype for this method is

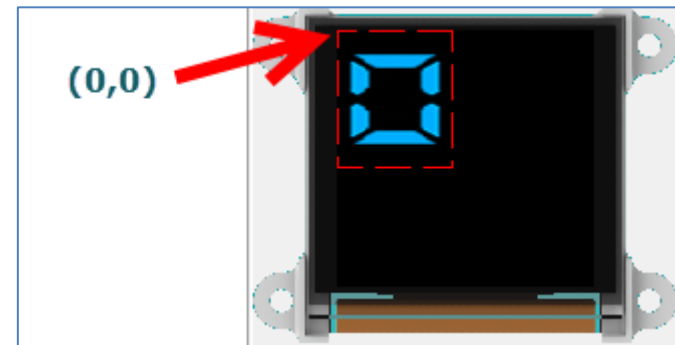
```
void media_Image(word X, word Y);
```

where the arguments “X” and “Y” are the x and y coordinates of the location at which the object will be displayed. To display **Image1** at the top-left corner of the display module, for example, write,

```
Display.media_SetAdd(0, 0);
```

```
Display.media_Image(0, 0);
```

The code above will show **Image1** at the top-left corner of the display module, like as shown below.

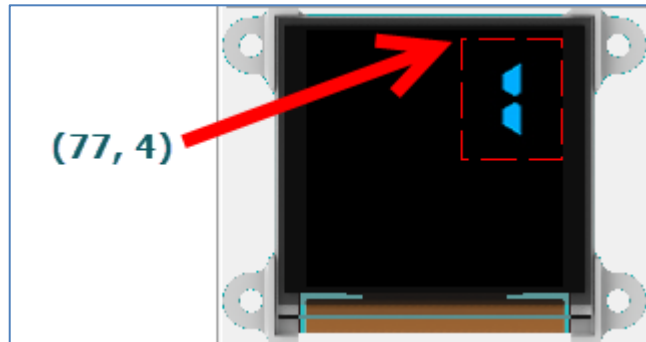


To display **Image2** at the top-middle part of the display module, write,

```
Display.media_SetAdd(0, 0x2000);
```

```
Display.media_Image(77, 4);
```

The code above will show **Image2** at the top-middle part of the display module, like as shown below.



## Connect the Arduino Host to the Display Module.

Refer to the section **“Connect the Display Module to the Arduino Host”** of the application note [Serial Connection to an Arduino Host](#) for the following topics:

- Using the New 4D Arduino Adaptor Shield (Rev 2.00)
  - Definition of Jumpers and Headers
  - Default Jumper Settings
  - Change the Arduino Host Serial Port
  - Power the Arduino Host and the Display Separately
- Using the Old 4D Arduino Adaptor Shield (Rev 1)
- Connection Using Jumper Wires

Note that in the attached sketch, the Arduino Uno uses a software serial port (TX pin is pin 10, RX pin is pin 11) to communicate with the display module. It uses the hardware serial port Serial0 to communicate with the Serial Monitor of the Arduino IDE.

## 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.