



**4D Systems**

---

## **Application Note: 4D-AN-1002**

# **Displaying an Image Using the FAT Controller for SGC Modules**

Document Date: 31<sup>st</sup> October 2011

Document Revision: 1.0

---

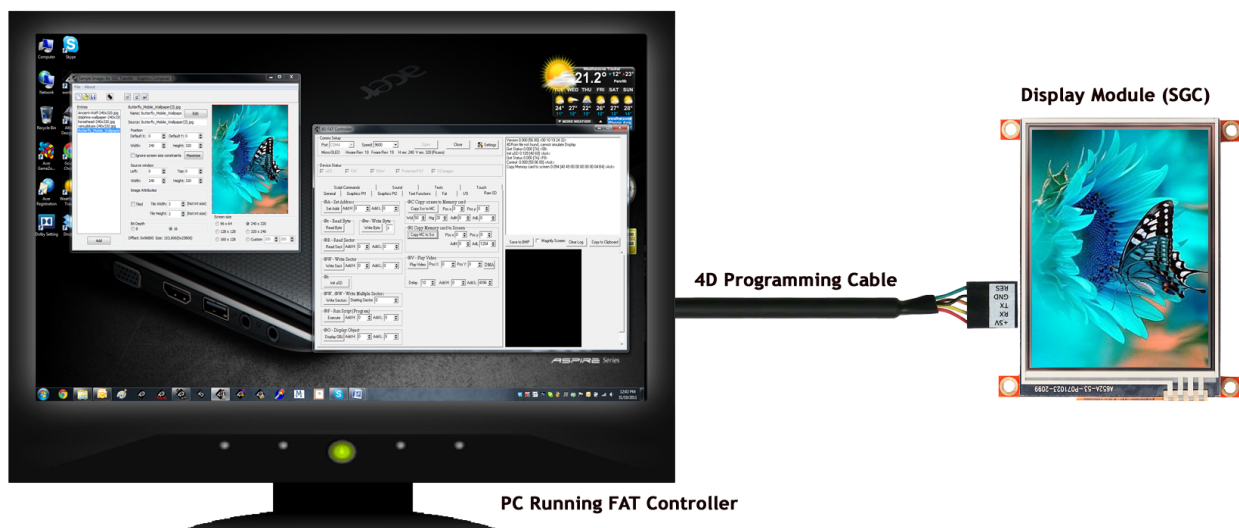
## Description

This Application Note is dedicated to explaining how to display an image on a 4D SGC module using the FAT Controller Software Tool. The FAT Controller is effectively emulating that of a host micro-controller, which is used to control the functionality of the SGC display module. In order to undertake this Application note, the following items are required:

- Any 4D SGC Display Module
  - Windows based PC
  - 4D Programming Cable
  - Micro-SD ( $\mu$ SD) Card
  - FAT Controller Software Tool
  - Graphics Composer Software Tool
- 

## Application Overview

Displaying an image on a 4D Screen Module is one of the most essential tasks to know how to do. For SGC modules, it is possible to test and experiment with images, prior to implementing a host micro-controller to carry out such a task. The FAT Controller is capable of generating serial commands directly to the 4D module via a Windows based PC. This Application Note will explore ways to prepare an image, as well as the necessary steps to display it.



## Setup Procedure

### FAT Controller & Graphics Composer Software Tools

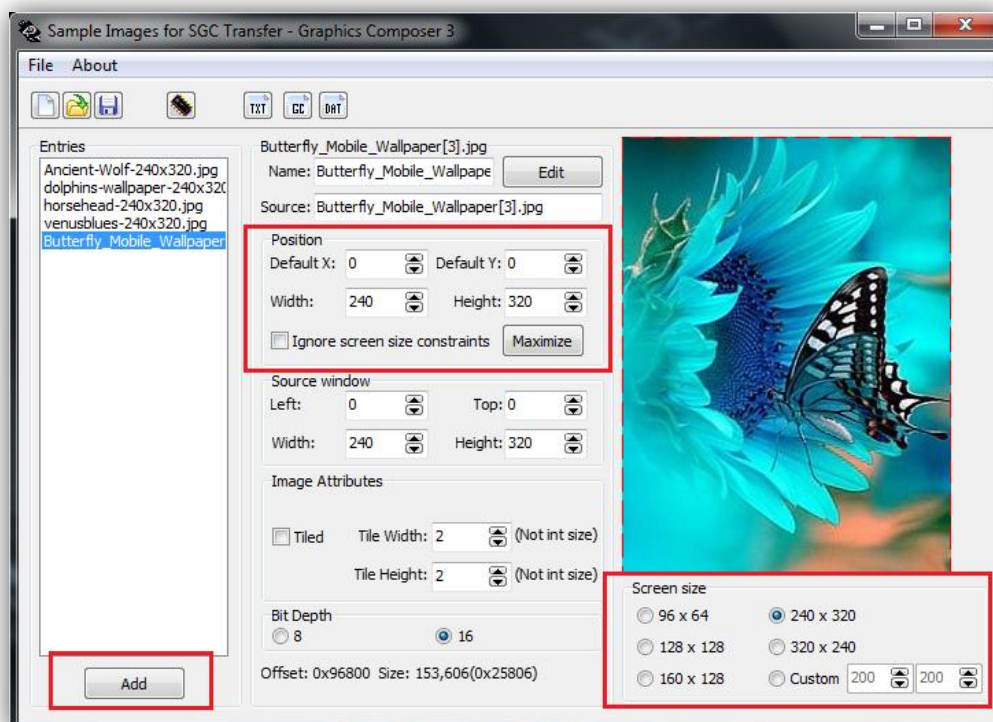
Firstly, you will need to download two software tools from the 4D Systems website titled FAT Controller and Graphics Composer. The FAT Controller is used to establish communication with the SGC module and send the necessary serial commands. The Graphics Composer is used to decompress an image into 4D format, which is essential for correct display functionality. The FAT Controller and the Graphics Composer can be sourced from the following links respectively:

<http://www.4dsystems.com.au/prod.php?id=70>

<http://www.4dsystems.com.au/prod.php?id=50>

### Preparing an Image

Open the Graphics Composer software and begin by adding at least two different images that are the same resolution as the screen you are testing. For example; if you are using a uLCD-32PT(SGC), the resolution will be 240x320; whereas if you are using a uOLED-96-G1(SGC), the resolution will be 96x64. Sample images can be found by doing a simple internet search. Save these somewhere on your computer hard drive. Click on the 'Add' button found in the bottom left hand corner. Ensure that the default X and Y position are set to zero, so that the image is built from the upper left hand corner of the screen. If this is not done, only certain portions of the image will be seen. Select the screen size of the module in the bottom right hand corner.

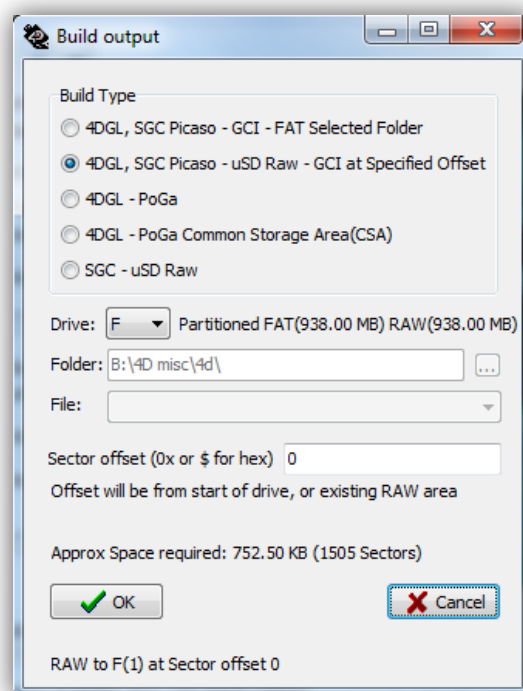


Following this, click the 'build' button indicated by the small chip near the top of the screen.



## Building the Output

The Build Output screen will appear. One of two options can be chosen for this example; '4DGL, SGC Picaso – uSD Raw – GCI at Specified Offset' or 'SGC – uSD Raw'. Different SGC commands would be used in each case, however, in this example, select '4DGL, SGC Picaso – uSD Raw – GCI at Specified Offset'. Enter the Sector offset to zero, as this will starting point for data in the RAW section. In addition, select the Drive where the  $\mu$ SD is stored. Click OK.



## Save Destination

You will be prompted to save the file in a destination folder. Choose somewhere convenient to locate. Only the text files will be saved at this location. By default, files are saved in the destination folder where the Graphics Composer is saved.

## Test Simulation

Now that the images are stored on the  $\mu$ SD, the FAT Controller can be used to test simulation with a host microcontroller and verify results by displaying the images on the screen.

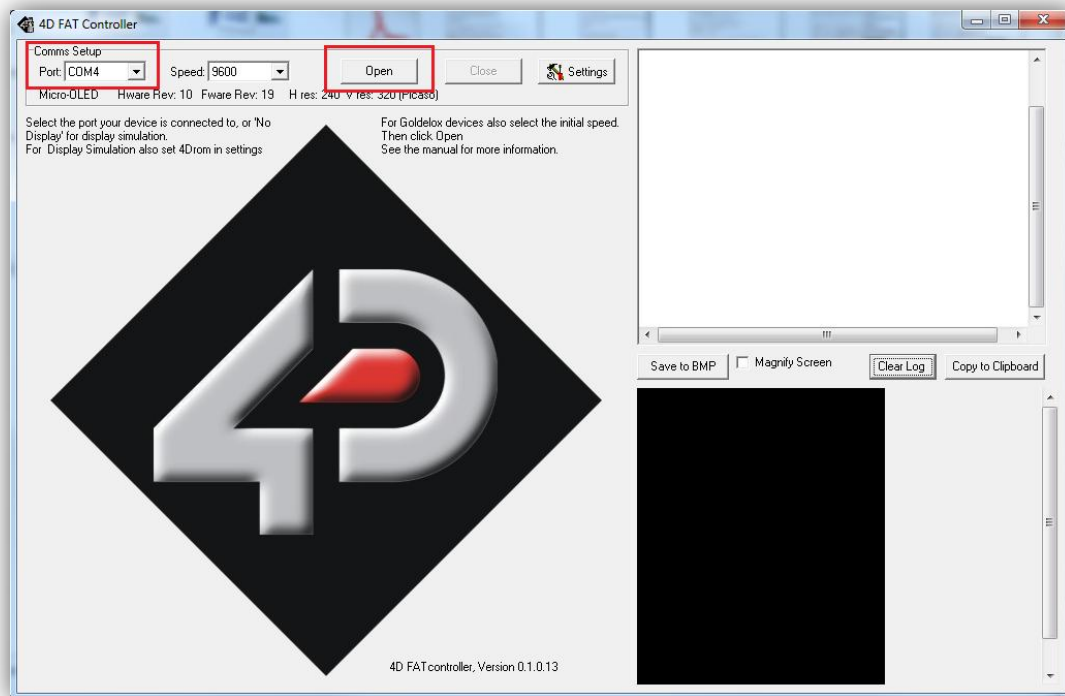
## Simulation Procedure

### PC Connectivity

Load the  $\mu$ SD card into the screen module and plug the module into the computer using a 4D Programming Cable or, USB cable and either a uUSB-CE5 or uUSB-MB5 converter (All of which can be purchased from the products page on the 4D website).

<http://www.4dsystems.com.au/products.php>

Open the FAT Controller software and select the COM PORT that the Module is connected to; then click Open.



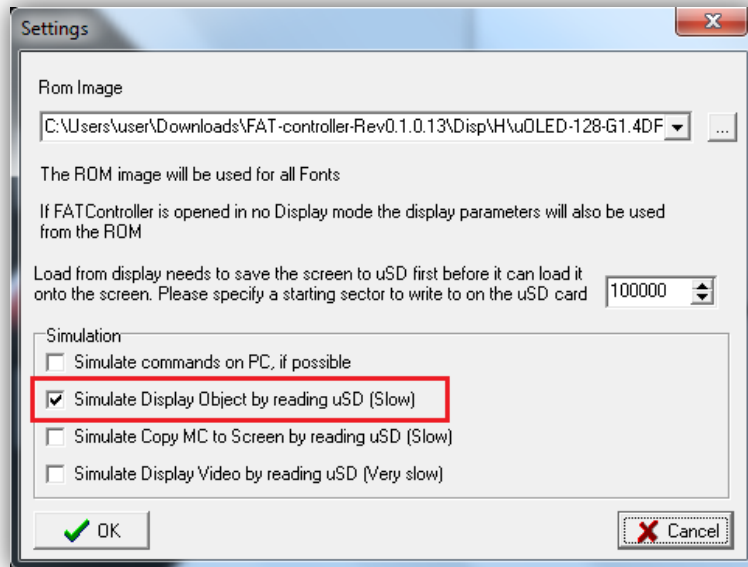
An assortment of settings will now be displayed, that allow the user to send and receive commands from the screen; just like what would be done with a separate host processor.

### FAT Controller Settings

Firstly, click on the 'Settings' button in the upper middle of the screen.

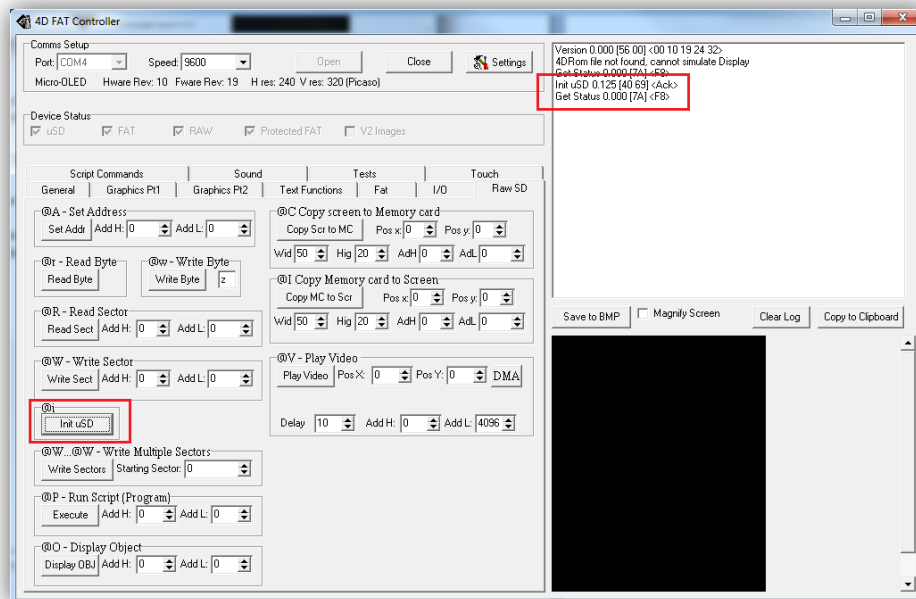


Uncheck the 3<sup>rd</sup> box down; 'Simulate Copy MC to Screen by reading  $\mu$ SD (Slow)'. This will improve the speed of the data transfer. Click OK.

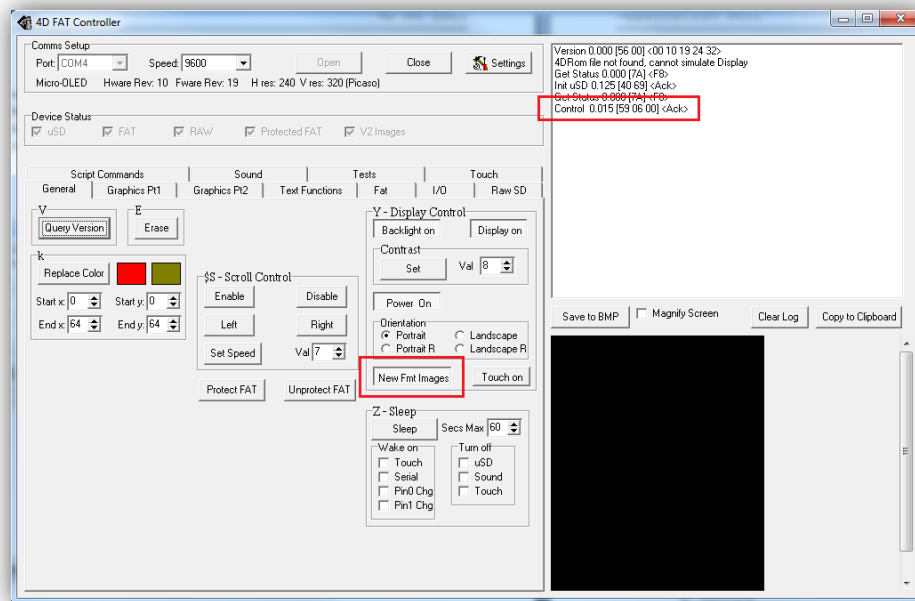


## µSD Card Initialisation

Initialise the µSD card by first selecting the 'Raw SD' Tab, then clicking on the 'Init µSD' button found in the middle on the left hand side. Two commands should appear in the log as indicated in the diagram below.

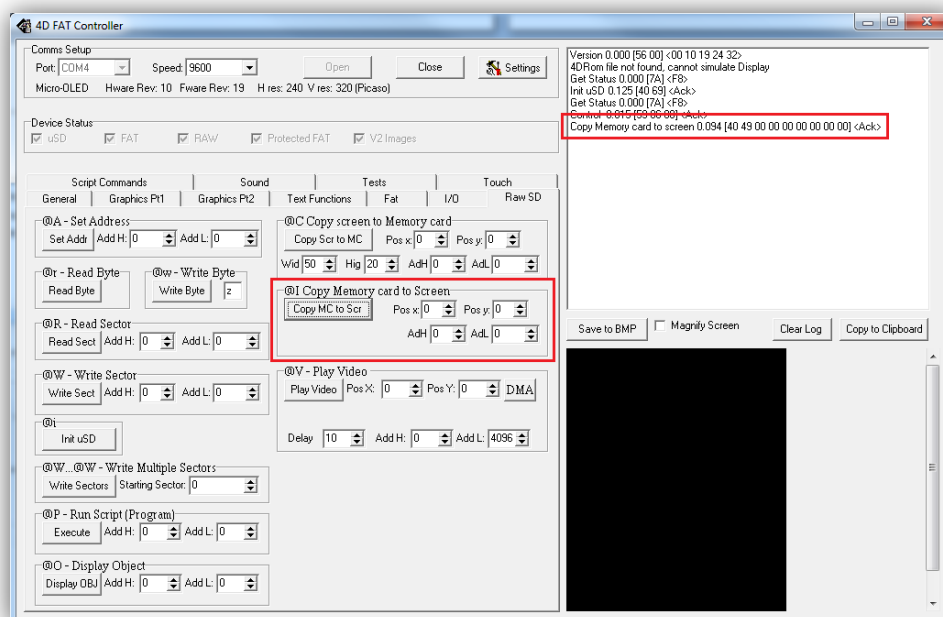


Now click on the 'General' Tab and select the 'New Fmt Images' button found in the central area of the screen. The 'New Fmt Images' button is specific to the '4DGL, SGC Picaso – uSD Raw – GCI at Specified Offset' option that was selected before. If the 'SGC – uSD Raw' setting was selected, then the 'New Fmt Images' will be replaced with 'Old Fmt Images'. For this example however, follow the steps shown.



## Copy Memory Card to Screen

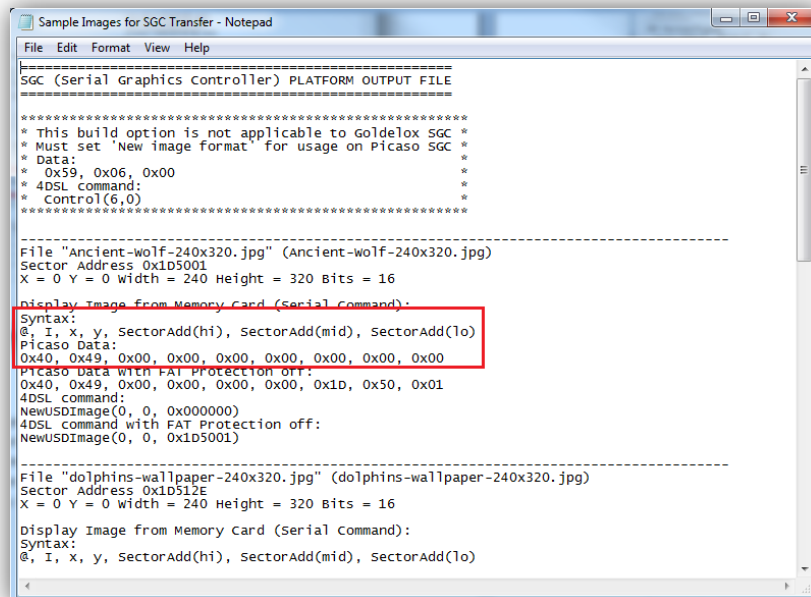
Go back to the 'Raw SD' Tab and draw your attention to the 'Copy Memory card to Screen' Section as indicated below. Set the X and Y variables to zero, for the same reason as mentioned before. The Address High (AdH) and Address Low (AdL) need to be set based on the results of the GC file generated from the Graphics Composer. Open the location of the GC file and open the .txt that was created that is the same name as the GC file. This can be found where the file was saved earlier.





## Extracting the Sector Offset Hex Address

Included in the .txt file are various details about each image created. From Here the AdH and AdL values can be extracted. Refer to the first image listed and draw your attention to the first line of hexadecimal values. See the highlighted region.



```

Sample Images for SGC Transfer - Notepad
File Edit Format View Help
=====
SGC (Serial Graphics Controller) PLATFORM OUTPUT FILE
=====
*****
* This build option is not applicable to Goldebox SGC *
* Must set 'New image format' for usage on Picaso SGC *
* Data:
* 0x59, 0x06, 0x00
* 4DSL command:
* Control(6,0)
*****

-----
File "Ancient-wolf-240x320.jpg" (Ancient-wolf-240x320.jpg)
Sector Address 0x1D5001
X = 0 Y = 0 width = 240 Height = 320 Bits = 16

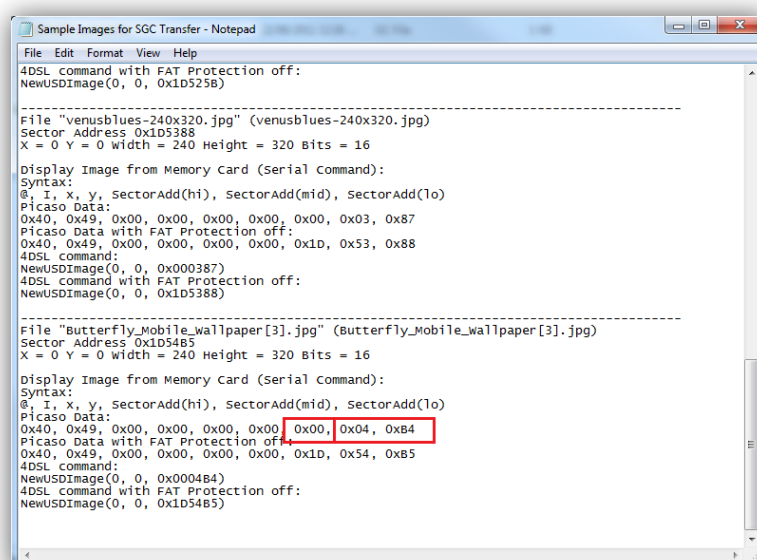
Display Image from Memory Card (Serial Command):
Syntax:
@, I, x, y, SectorAdd(hi), SectorAdd(mid), SectorAdd(lo)
Picaso Data:
0x40, 0x49, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
Picaso Data with FAT Protection off:
0x40, 0x49, 0x00, 0x00, 0x00, 0x00, 0x1D, 0x50, 0x01
4DSL command:
NewUSDImage(0, 0, 0x000000)
4DSL command with FAT Protection off:
NewUSDImage(0, 0, 0x1D5001)

-----
File "dolphins-wallpaper-240x320.jpg" (dolphins-wallpaper-240x320.jpg)
Sector Address 0x1D512E
X = 0 Y = 0 width = 240 Height = 320 Bits = 16

Display Image from Memory Card (Serial Command):
Syntax:
@, I, x, y, SectorAdd(hi), SectorAdd(mid), SectorAdd(lo)

```

The Hex values must be converted to decimal and then entered into the FAT Controller. The 3<sup>rd</sup> last set of Hex is the AdH value, whilst the 2<sup>nd</sup> last and last combined make up the AdL value. An example follows:



```

Sample Images for SGC Transfer - Notepad
File Edit Format View Help
=====
4DSL command with FAT Protection off:
NewUSDImage(0, 0, 0x1D5258)

-----
File "venusblues-240x320.jpg" (venusblues-240x320.jpg)
Sector Address 0x1D5388
X = 0 Y = 0 width = 240 Height = 320 Bits = 16

Display Image from Memory Card (Serial Command):
Syntax:
@, I, x, y, SectorAdd(hi), SectorAdd(mid), SectorAdd(lo)
Picaso Data:
0x40, 0x49, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x87
Picaso Data with FAT Protection off:
0x40, 0x49, 0x00, 0x00, 0x00, 0x00, 0x1D, 0x53, 0x88
4DSL command:
NewUSDImage(0, 0, 0x000387)
4DSL command with FAT Protection off:
NewUSDImage(0, 0, 0x1D5388)

-----
File "butterfly_mobile_wallpaper[3].jpg" (butterfly_mobile_wallpaper[3].jpg)
Sector Address 0x1D54B5
X = 0 Y = 0 width = 240 Height = 320 Bits = 16

Display Image from Memory Card (Serial Command):
Syntax:
@, I, x, y, SectorAdd(hi), SectorAdd(mid), SectorAdd(lo)
Picaso Data:
0x40, 0x49, 0x00, 0x00, 0x00, 0x00, 0x00, 0x04, 0xB4
Picaso Data with FAT Protection off:
0x40, 0x49, 0x00, 0x00, 0x00, 0x00, 0x1D, 0x54, 0xB5
4DSL command:
NewUSDImage(0, 0, 0x0004B4)
4DSL command with FAT Protection off:
NewUSDImage(0, 0, 0x1D54B5)

```



- ## Displaying Images at Varying Sector Offsets

4D FAT Controller

Comms Setup  
 Port: COM4 Speed: 9600 [Open] [Close] [Settings]  
 Micro-OLED Hwrev: 10 Fwrev: 19 H res: 240 V res: 320 [Picaso]

Device Status  
☒ uSD ☒ FAT ☒ RAW ☒ Protected FAT ☒ V2 Images

Script Commands  
 General Graphics P1 Graphics P2 Text Functions Tests I/O Touch Raw SD

@A - Set Address  
 Set Addr: Add H: 0 Add L: 0  
 @R - Read Byte Read Byte  
 @W - Write Byte Write Byte z  
 @R - Read Sector Read Sect: Add H: 0 Add L: 0  
 @W - Write Sector Write Sect: Add H: 0 Add L: 0  
 @i Init uSD  
 @W - Write Multiple Sectors Write Sectors Starting Sector: 0  
 @P - Run Script (Program) Execute Add H: 0 Add L: 0  
 @O - Display Object Display OBJ: Add H: 0 Add L: 0

@C Copy screen to Memory card  
 Copy Scr to MC Pos x: 0 Pos y: 0  
 Wid: 50 Hig: 20 AdH: 0 AdL: 0  
 @I Copy Memory card to Screen  
 Copy MC to Scr Pos x: 0 Pos y: 0  
 AdH: 0 AdL: 1204

@V - Play Video  
 Play Video Pos X: 0 Pos Y: 0 DMA  
 Delay 10 Add H: 0 Add L: 4096

Console Output:  
 Init uSD 0.125 [40 63] <ack>  
 Get Status 0.000 [7A] <FB>  
 Control 0.015 [59 06 00] <ack>  
 Copy Memory card to screen 0.094 [40 49 00 00 00 00 00 00] <ack>  
 Copy Memory card to screen 0.515 [40 49 00 00 00 01 00 20] <No response>  
 Copy Memory card to screen 0.500 [40 49 00 00 00 00 00 00] <No response>  
 Copy Memory card to screen 0.515 [40 49 00 00 00 00 00 00] <No response>  
 CDM4 Closed  
 Version 0.016 [56 00] <00 10 19 24 32>  
 ADROM file not found, cannot simulate Display  
 Get Status 0.000 [7A] <FB>  
 Init uSD 0.125 [40 63] <ack>  
 Get Status 0.000 [7A] <FB>  
 Control 0.000 [59 06 00] <ack>  
 Copy Memory card to screen 0.093 [40 49 00 00 00 00 00 00] <ack>  
 Copy Memory card to screen 0.093 [40 49 00 00 00 00 01 20] <ack>  
 Copy Memory card to screen 0.094 [40 49 00 00 00 00 02 5a] <ack>  
 Copy Memory card to screen 0.094 [40 49 00 00 00 00 03 87] <ack>  
 Copy Memory card to screen 0.094 [40 49 00 00 00 00 04 84] <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.