4D Systems

Application Note: 4D-AN-1002

Displaying an Image Using the FAT Controller for SGC Modules

Document Date: 31st October 2011

Document Revision: 1.0

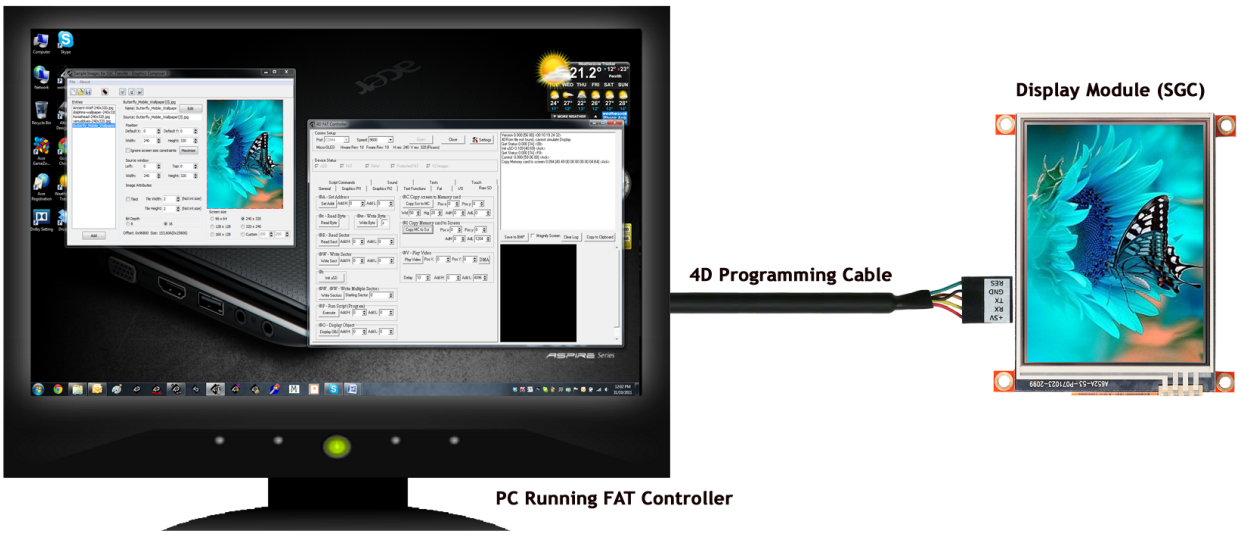
Description

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 (µSD) 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

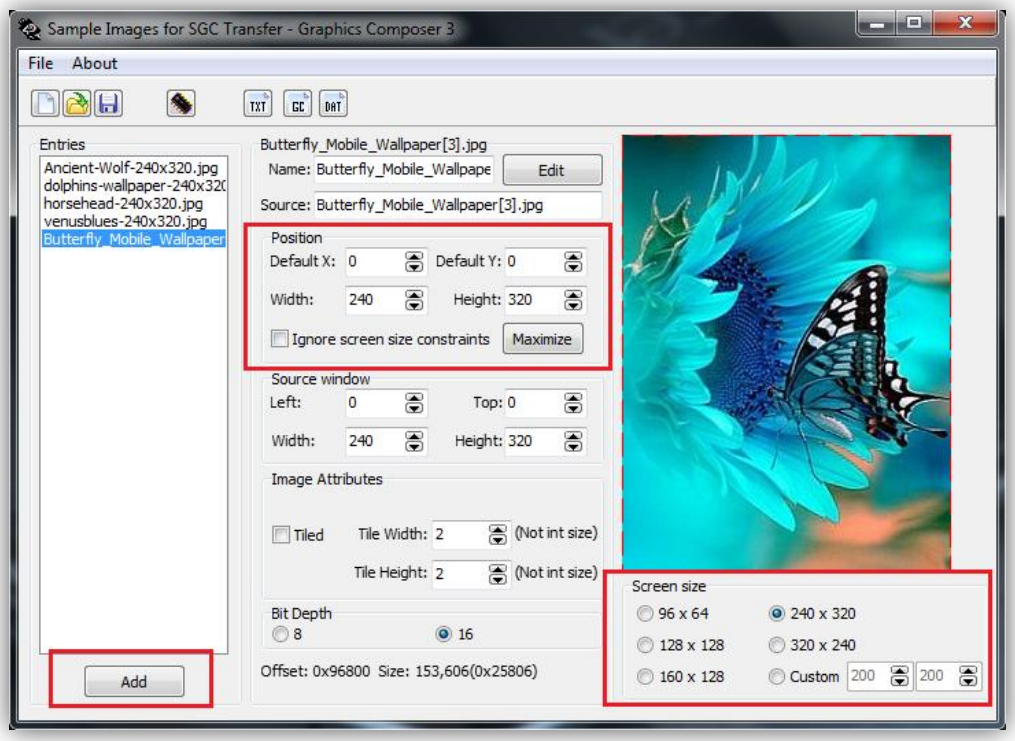
Firstly, you will need to download two software tools from the 4D Systems website titled FAT Controller and Graphics Composer. The FAT Controller is used to establish communication wit hthe SGC module and send the necessary serial commands. The Graphics Composer is used to decompress an image into 4D 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

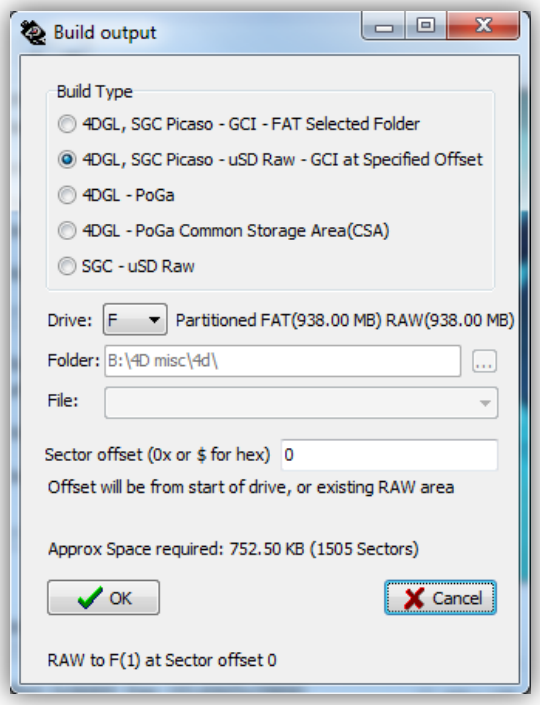
Open the Graphics Composer software and begin by adding at least two different images that are the same resolution as the screen you are testing. For example; if you are using a uLCD-32PT(SGC), the resolution will be 240x320; whereas if you are using a uOLED-96-G1(SGC), the resolution will be 96x64. 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 µSD 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 µSD, the FAT Controller can be used to test simulation with a host microcontroller and verify results by displaying the images on the screen.

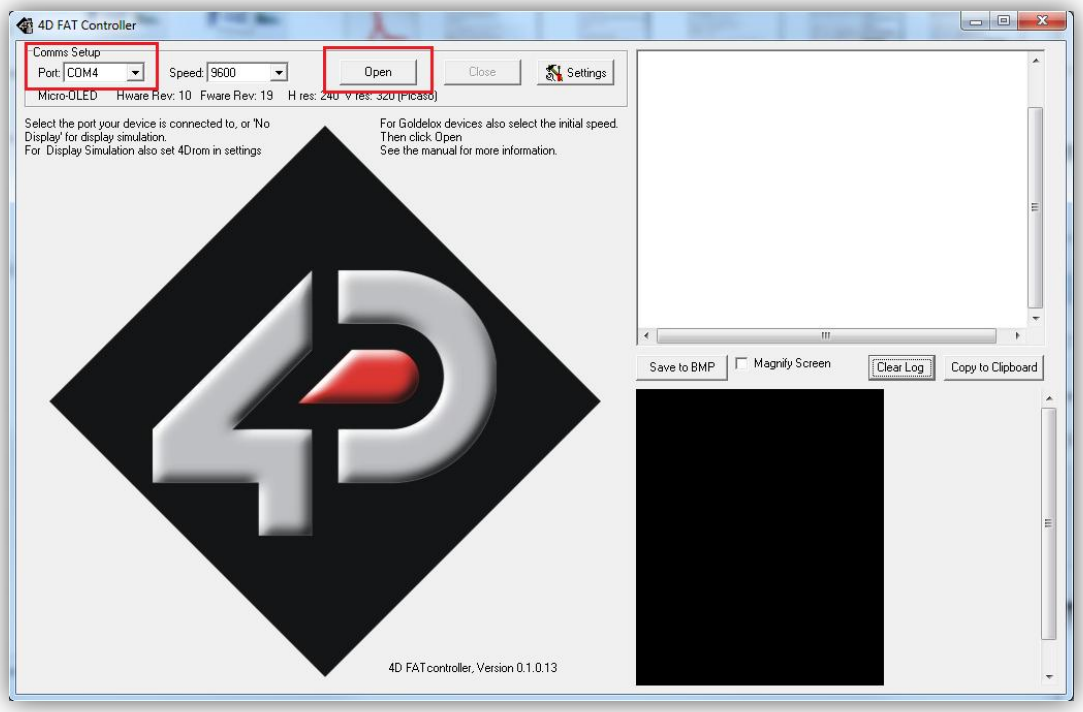
Simulation Procedure

PC Connectivity

Load the µSD 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.

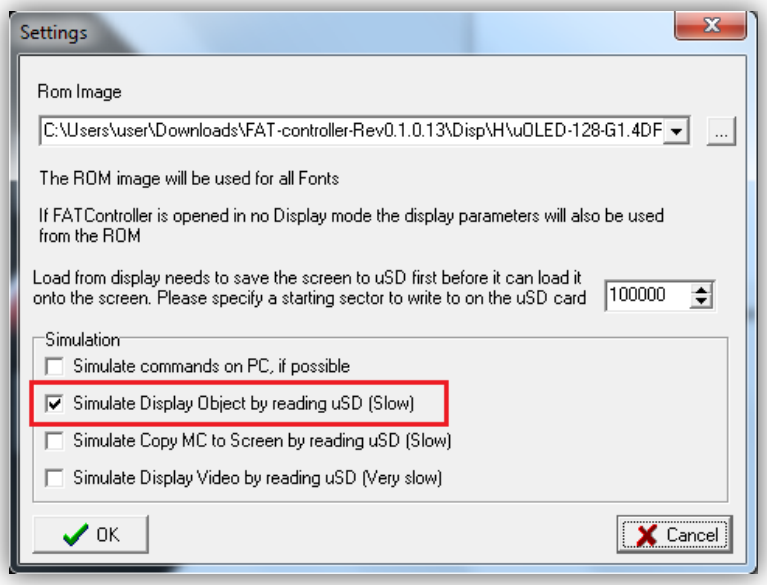


An assortment of settings will now be displayed, that allow the user to send and receive commands from the screen; just like what would be done with a separate host processor.

FAT Controller Settings

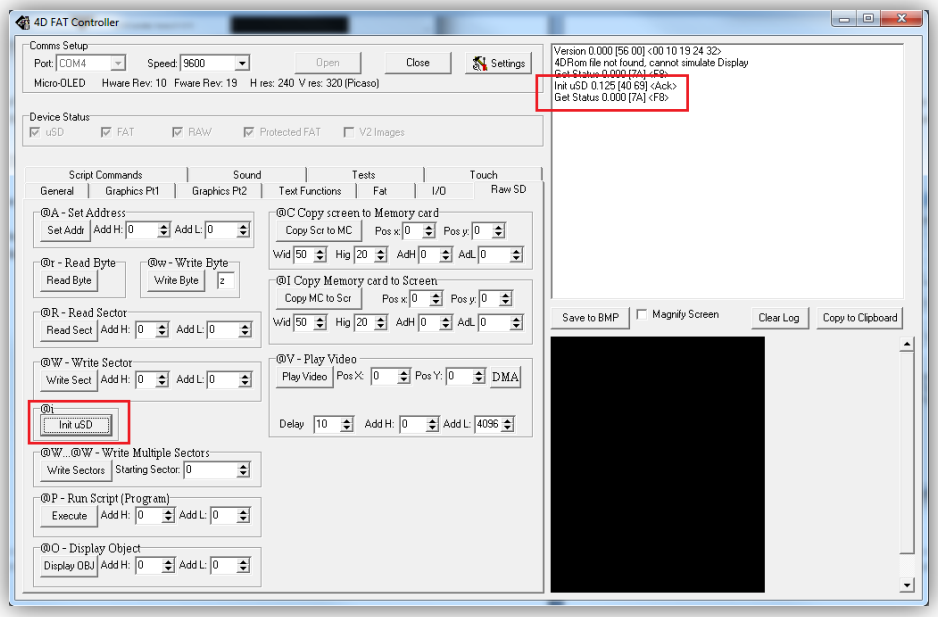
Firstly, click on the 'Settings' button in the upper middle of the screen. 

Uncheck the 3rd box down; 'Simulate Copy MC to Screen by reading µSD (Slow)'. This will improve the speed of the data transfer. Click OK.

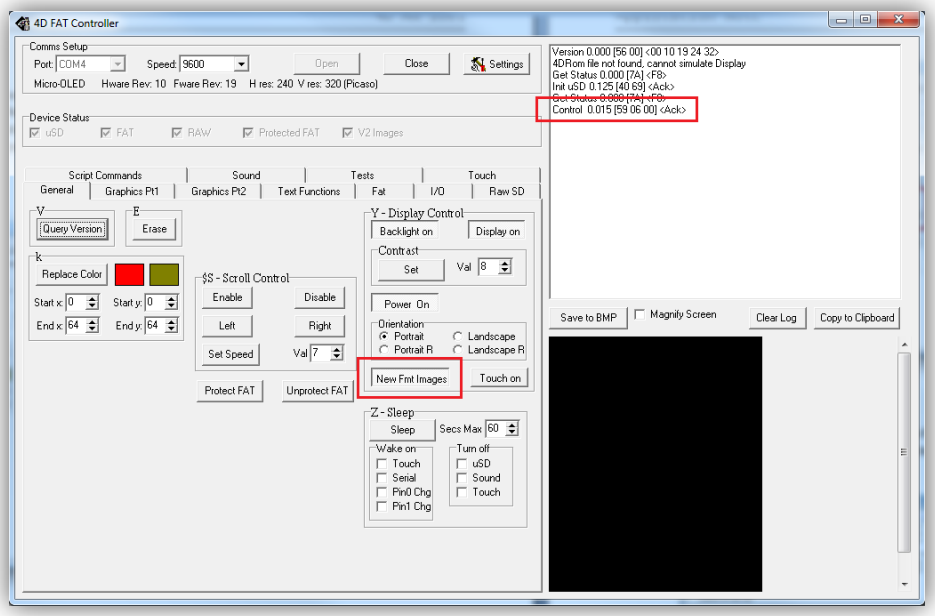


µSD Card Initialisation

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.

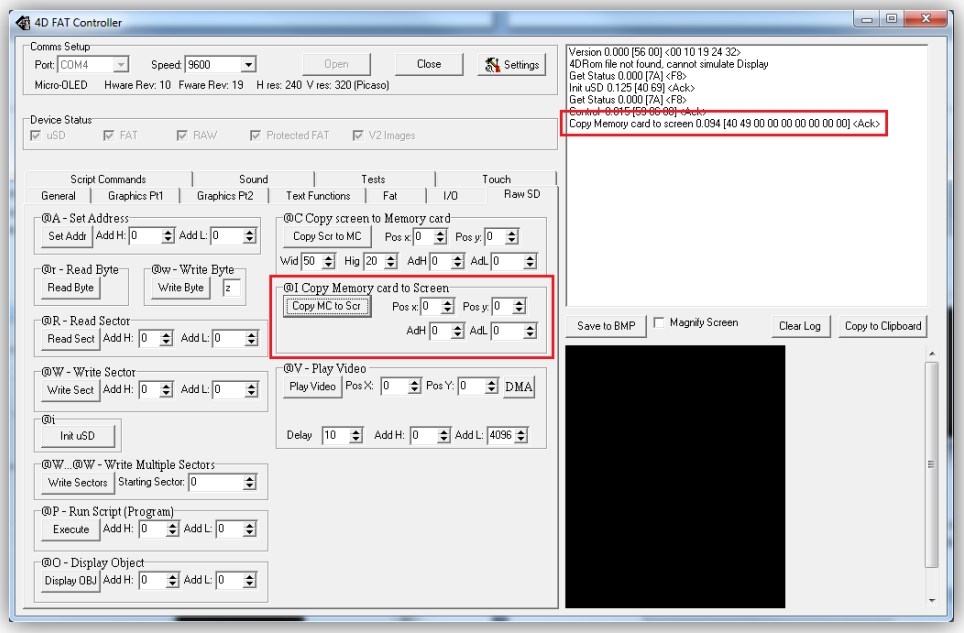


Now click on the 'General' Tab and select the 'New Fmt Images' button found in the central area of the screen. The 'New Fmt Images' button is specific to the '4DGL, SGC Picaso – uSD Raw – GCI at Specified Offset' option that was selected before. If the 'SGC – uSD Raw' setting was selected, then the 'New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



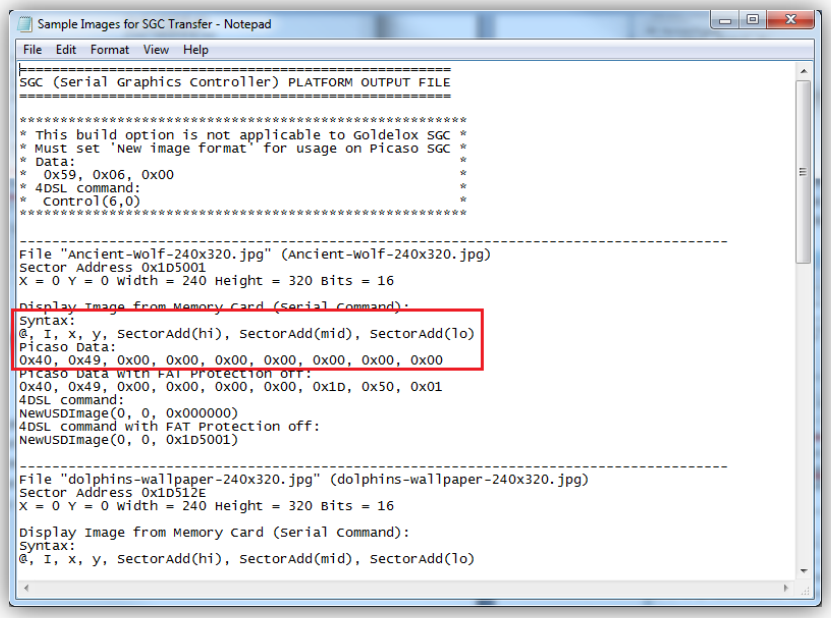
Copy Memory Card to Screen

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.

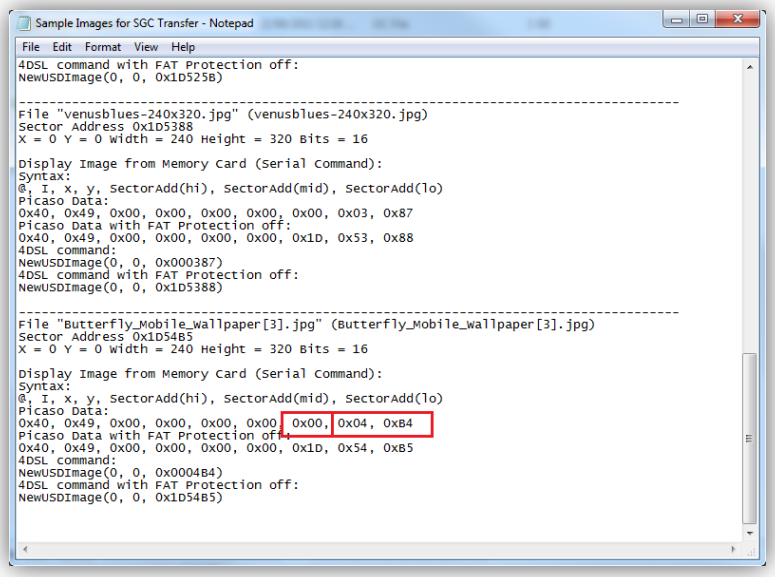


Extracting the Sector Offset Hex Address

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.



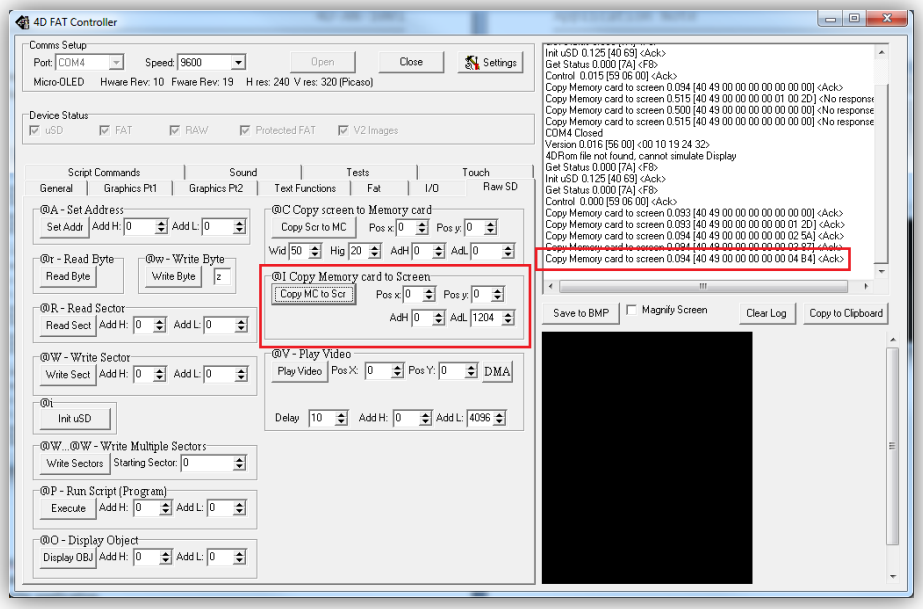
The Hex values must be converted to decimal and then entered into the FAT Controller. The 3rd last set of Hex is the AdH value, whilst the 2nd last and last combined make up the AdL value. An example follows:



- AdH = 0x00 (Hex) = 0 (dec)
- AdL = 0x04B4 (Hex) = 1204 (dec)
- Converting hexadecimal to decimal can be done using your computer's scientific calculator or searching for an internet website application.

Displaying Images at Varying Sector Offsets

Insert these values back into the FAT Controller and click the 'Copy MC to Scr' button. The image should now be displayed. By entering the various image addresses found in the .txt file, each different image can be displayed. You will notice that these addresses are spaced at equal intervals apart.



Proprietary Information

Disclaimer of Warranties & Limitation of Liability