Multimedia Files on 4D E-blocks

by David A April 2012

Abstract

The new 4D display E-blocks allow users to run files from an SD card slot located on the module. However getting the boards to access these files can be a little bit tricky, this article is a comprehensive guide on how to access the multimedia files on both the Picaso and Goldelox display modules.

Requirements

Software:

- Professional or Educational version of Flowcode V5 for PICmicros
- Graphics Composer software from 4D

Hardware:

- One of EB071, EB075, EB076.
- Multiprogrammer board
- Micro SD card

Picaso display modules - EB071 & EB076

The EB071 and EB076 E-blocks both use a ‘Picaso’ display module, these modules allow reading pictures and sound files from a file structure rather than just raw data. However the display modules still require a raw data format for displaying videos.

Pictures - FAT16
Sound - FAT16
Video - Raw

Displaying images

- *µSD card - formatted with FAT16 file structure*

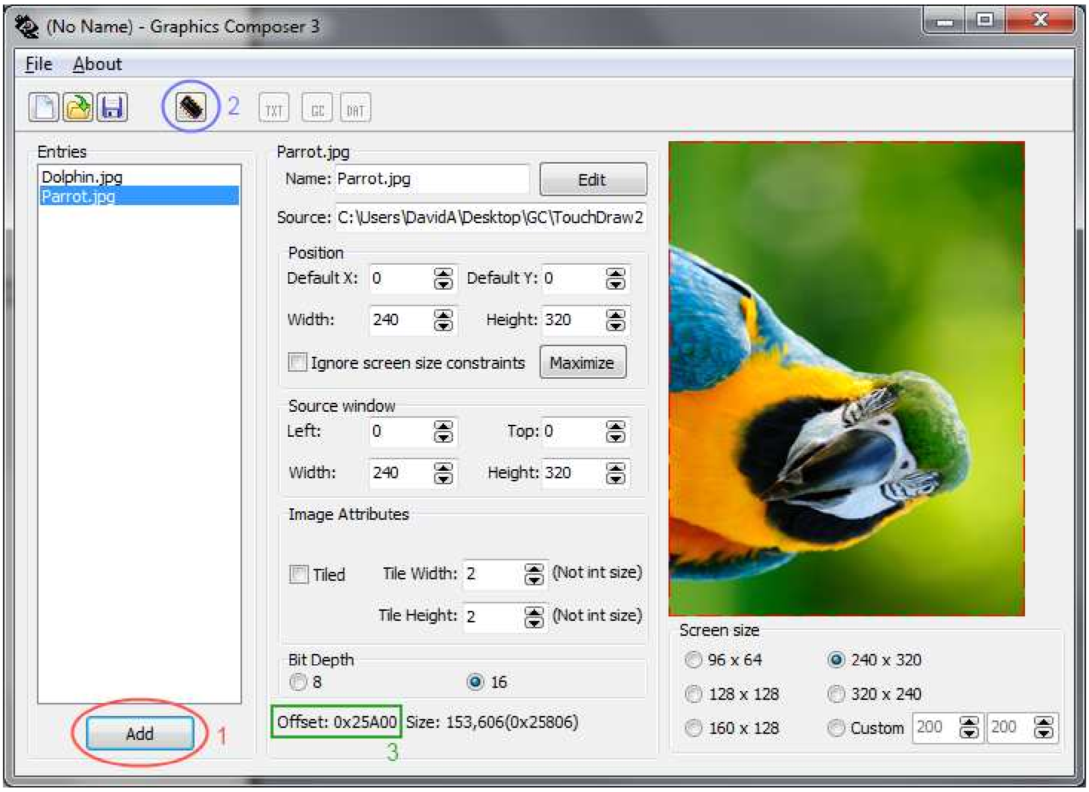
Step 1: Adding the files to the Graphics Composer

The first step is to load up the Graphics Composer software, once you have loaded your software you can then begin to add the images to the project. Click the add button at the bottom (red (1) in *pic.1*) and select the image you wish to add, this can be repeated for as many pictures as you wish to add to the project.

After adding all the required images and altering the settings for each (if required), it is a good idea to then note down the “offset” (green (3) in *pic.1*) for every image you have added to the project:

Eg:
Picture1: *offset = 0x0*
Picture2: *offset = 0x25A00*
Picture3: *offset = 0x4B400*

Now we need to build this assortment of pictures into a file that is readable by the graphics module, to do this we click the ’Build’ button in the software (blue (2) in *Pic.1*). From there we will be presented with another menu which asks what Build type to use (*Pic. 2)*.



(No Name) - Graphics Composer 3

File About

Entries: Dolphin.jpg, Parrot.jpg

Add

Parrot.jpg

Name: Parrot.jpg Edit

Source: C:\Users\DavidA\Desktop\GC\TouchDraw2

Position
Default X: 0 Default Y: 0
Width: 240 Height: 320
□ Ignore screen size constraints Maximize

Source window
Left: 0 Top: 0
Width: 240 Height: 320

Image Attributes
□ Tiled Tile Width: 2 (Not int size)
Tile Height: 2 (Not int size)

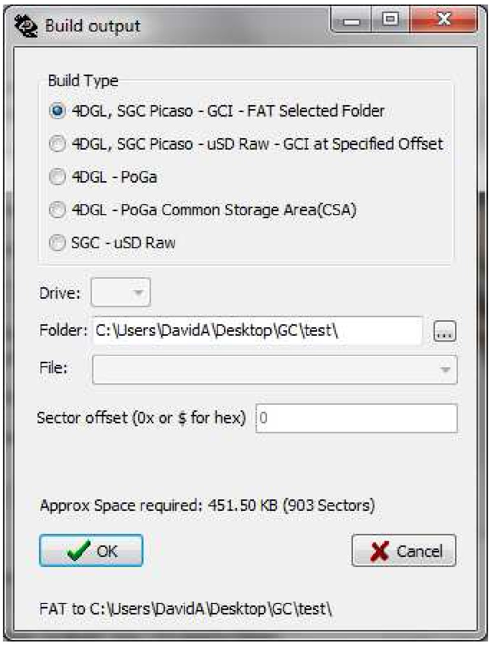
Bit Depth
○ 8 ● 16

Offset: 0x25A00 Size: 153,606(0x25806)

Screen size
○ 96 x 64 ● 240 x 320
○ 128 x 128 ○ 320 x 240
○ 160 x 128 ○ Custom 200 200

Pic 1

Pic 2



Build output

Build Type
● 4DGL, SGC Picaso - GCI - FAT Selected Folder
○ 4DGL, SGC Picaso - uSD Raw - GCI at Specified Offset
○ 4DGL - PoGa
○ 4DGL - PoGa Common Storage Area(CSA)
○ SGC - uSD Raw

Drive:

Folder: C:\Users\DavidA\Desktop\GC\test\

File:

Sector offset (0x or $ for hex) 0

Approx Space required: 451.50 KB (903 Sectors)

OK Cancel

FAT to C:\Users\DavidA\Desktop\GC\test\

For Picaso images we need to select the option:

- **4DGL, SGC Picaso - GCI - FAT Selected Folder**

You can then choose to save it directly to the SD card or onto a folder on your computer and copy the files over afterwards.

Now you should have two pieces of information which we can use to get these displays working in Flowcode.

- Offset for every picture
- Filename of the graphics composer output on the SD card.

***Step 2:* Using with Flowcode**

To display the pictures we added to the Graphics Composer file within our Flowcode project we must use the macro:

G4D_DisplayImageFromFilePicaso

This macro has a few parameters which will all need filling in if we are to get our pictures displayed:

X - This is the X co-ordinate position of where the picture will begin on the screen
Y - This is the Y co-ordinate position of where the picture will begin on the screen
Filename - The filename of the Graphics Composer file on the SD card. (including file extension)
PosH - This is the higher int of the offset for each picture
PosL - This is the lower int of the offset for each picture

Eg for an offset of **0x25A00**: ***PosH = 0x2***
PosL = 0x5A00

Playing sound files

- *µSD card - formatted with FAT16 file structure*

The EB071 and EB076 can also play sounds through either the headphone port (EB071) or the miniature speaker (EB076). The devices can only play **.wav** files however, but getting these files play is significantly easier than displaying images.

Step 1: Adding the file to the SD card

This is a simple case of adding the **.wav** file you wish to use in your project onto the µSD card, just a simple copy onto the device will work, you just need to note down the filename of the **.wav** file you are going to use.

Step 2: Using the file with Flowcode

To use the **.wav** file we added to the SD card within our Flowcode project we must use the macro:

G4D_PlayAudioFromFilePicaso

This macro has a couple of parameters which will all need filling in if we are to get our pictures displayed:

Play_Option - This option dictates how the file is played, please see the gLCD Help file for more info.
Filename - The filename of the **.wav** file on the SD card. (including file extension).

All display modules - EB071, EB075 & EB076

Picaso & Goldelox - playing video files

- *µSD card - formatted with raw file structure*

All the E-blocks play video in the same way with one small difference which I will go over in this section. There are a number of formats which the video playback supports:

AVI, WMV, VOB, MPG, GIF

Step 1: **Adding the files to the Graphics Composer**

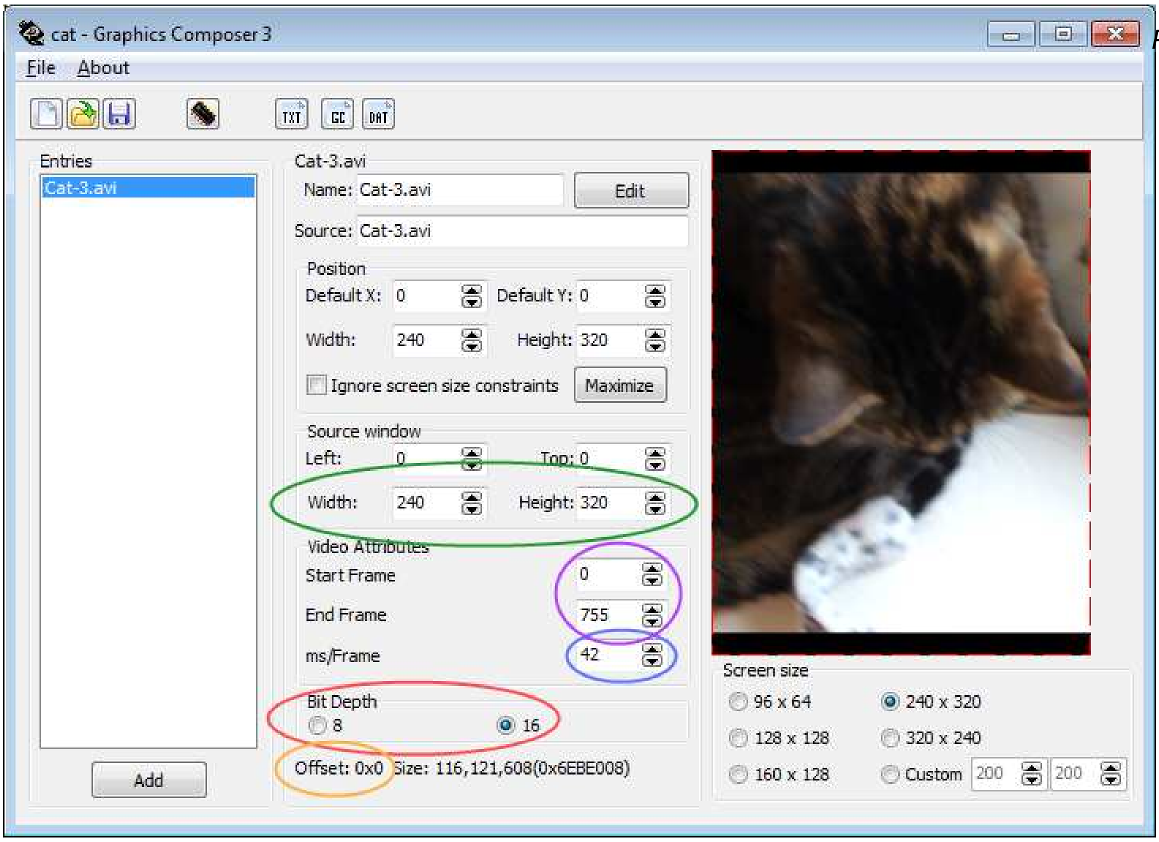
This is very similar to adding pictures to the project, with a few differences. The first is the number of attributes you need to note down from the Graphics Composer software (*Pic. 3)*.

1) Red Circle - Bit Depth
2) Blue Circle - ms/Frame
3) Green Circle - Width & Height of video
4) Orange Circle - Offset
5) Purple Circle - Start & End Frames

These will all be needed when we come to add the video to Flowcode.

Now we need to 'Build' the project, this time however in our Build Types we select the:

	4DGL - SGC Picaso - uSD Raw - GCI at Specific Offset	**- <u>EB071 & EB076</u>**
or	***SGC - uSD Raw***	- **<u>EB075</u>**



Entries
Cat-3.avi

Cat-3.avi
Name: Cat-3.avi
Source: Cat-3.avi

Position
Default X: 0 Default Y: 0
Width: 240 Height: 320
☐ Ignore screen size constraints

Source window
Left: 0 Top: 0
Width: 240 Height: 320

Video Attributes
Start Frame 0
End Frame 755
ms/Frame 42

Bit Depth
○ 8 ● 16

Offset: 0x0 Size: 116,121,608(0x6EBE008)

Screen size
○ 96 x 64 ● 240 x 320
○ 128 x 128 ○ 320 x 240
○ 160 x 128 ○ Custom 200 200

Pic 3

When building the project you also will specify a "**Sector Offset**", the default for this is zero, however if you do change it then you will also need to make a note of this value.

As the data is raw data, you must put it directly on the µSD card, so this must be plugged in and selected from the drop down **Drive** menu. Once this is done we are ready to move onto the Flowcode program.

Step 2: Using with Flowcode

To display the pictures we added to the Graphics Composer file within our Flowcode project we must use the macro:

G4D_DisplayVideoFromCard

This macro has a few parameters which will all need filling in if we are to get our pictures displayed:

X - The X co-ordinate position of where the picture will begin on the screen
Y - The Y co-ordinate position of where the picture will begin on the screen
Width - The specific width of the video. (Green Circle)
Height - The specific height of the video. (Green Circle)
ColourMode - 0 = 8bit, 1 = 16bit. (Red Circle)
Delay - Delay time between frames (Blue Circle)
Frames - Difference between start and end frames (Purple Circle)
AddH - This is the higher byte of the offset for each picture
AddM - This is the middle byte of the offset for each picture
AddL - This is the low byte of the offset for the video

The offset is split into 3 bytes for the address, the upper, middle and lower bytes, to calculate what offset your video is on, you will need two values:

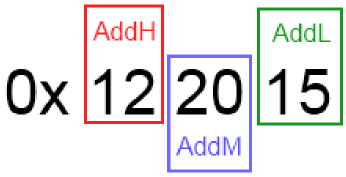
- Video offset in Graphics Composer (Orange Circle)
- **Sector offset** from the Build menu specified by the user

Adding these two values together will get you the offset for your Flowcode program. Most of the time this will be zero, unless you have multiple videos in the Graphics Composer or have manually set the offset to be anything other than zero.

Example: A user has two videos in their project, the first is located at offset 0x0 in the Graphics Composer project file, the second is located at 0x2015. The user also manually changes the offset when they Build the project to be 0x120000.

Video offset + Sector offset = Total offset
0x120000 + 0x2015 = 0x122015

To use this video in your Flowcode program you would then add the following values:



AddH = 0x12

AddM = 0x20

AddL = 0x15

Goldelox display modules - EB075

Displaying pictures

To display pictures on your EB075 E-blocks you need to follow the same instructions as with the Picaso modules, but with a couple of small differences.

1) The µSD card must have a **RAW** file format for the display to read it.

2) When selecting a ‘Build Output’ from the Graphics Composer software we now need to use a different output: ‘**SGC - uSD Raw’**

3) The offset is in three separate bytes instead of two integers, this offset is calculated in a similar manner to the offset calculation needed when playing video (shown above).

Further reading

Below are some links to other resources and articles on related subjects, and technical documentation relating to the hardware used for this project...

Flowcode:	http://www.matrixmultimedia.com/flowcode.php
Learning Centre:	http://www.matrixmultimedia.com/lc_index.php
User Forums:	http://www.matrixmultimedia.com/mmforums