EE403w Section 5
Embedded Audio Engineering
Lab 6 – REV A

Summary:

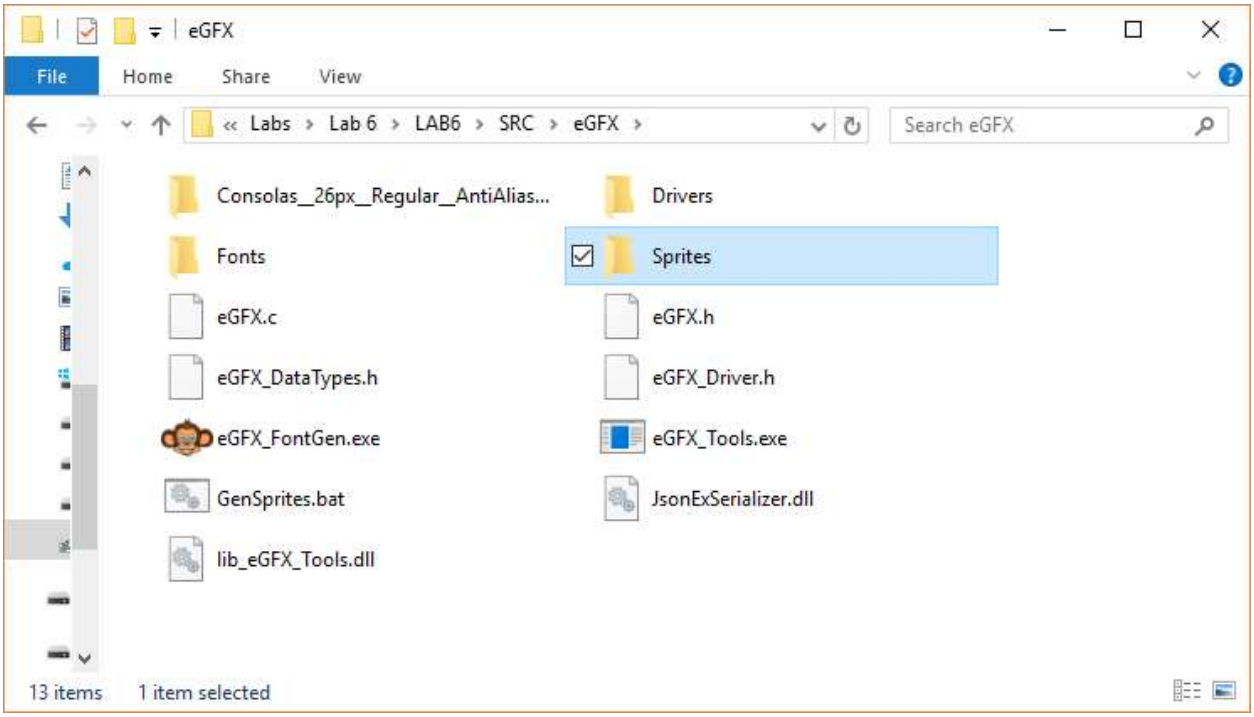
This lab is an example how to read the touch screen and draw some graphics to complete your UI. I have recorded a video and put in on Youtube to guide you through this project.

https://youtu.be/-mnmyhgc8hg

Summary:

1.) Adding custom graphics

It is possible to draw .png files on the screen with the eGFX graphics library. It is as simple as dropping the .png files in a folder and running a special tool. This tool converts the .png files to an **eGFX_ImagePlane** structure that can be copied on to the back buffer. The tool **"eGFX_Tools.exe"** is a command line tool that can scan a directory for files and make a .c and .h file with data structures you can use in your project. There is a batch file in the **"SRC\eGFX"** folder called **"GenSprites.bat"**. It calls the **"eGFX_Tools.exe"** program to look in the **"SRC\eGFX\Sprites\"** folder for all the images it can find and convert them into a C file.

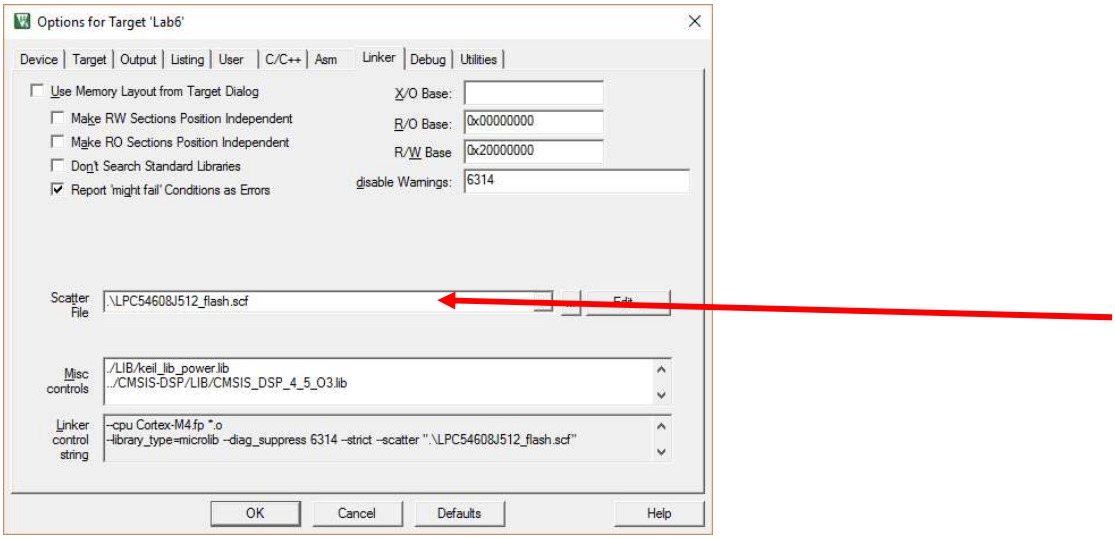


Since the screen on the LPC54608 is 16-bit color, large images can take up a large amount of flash memory. This project enables an special type of external flash memory that is connected to a "SPIFI" port. It is a memory with a serial interface but appears in the normal address space of the microcontroller. This means data from the SPIFI flash can be accessed the same as any other memory with a pointer. The **"GenSprites.bat"** also adds some special decorators to the variable names to make sure the image data gets placed in the SPIFI flash.

To get the Keil build tools to know how to put things into SPIFI flash, there are 2 other modifications to the project files needed

a.)

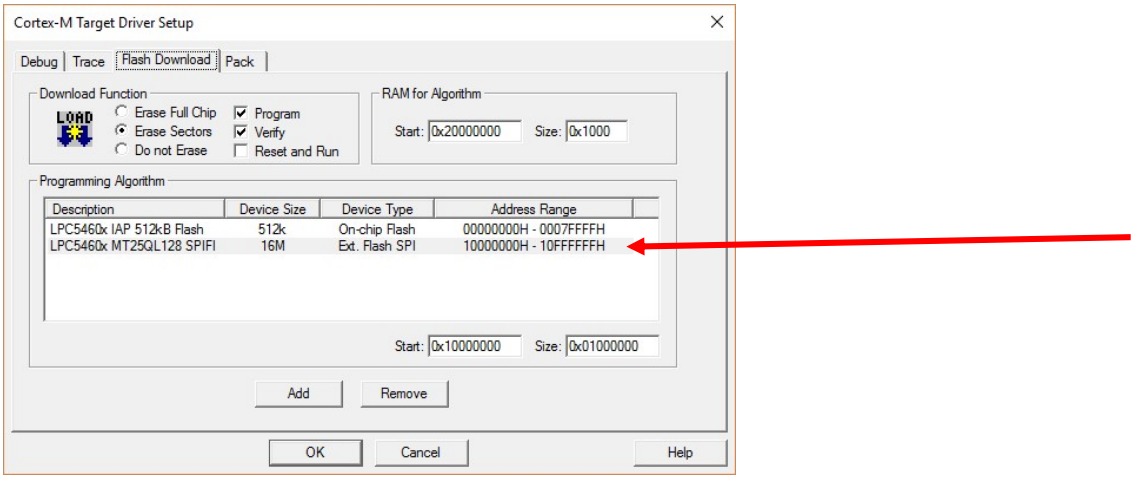
The "LPC54608J512_flash.scf" file (this is what guides the linker on where to put variables) was has edits (compared to previous labs) to create a new "section" for the SPIFI flash.




```
    LR_EROM1 0x10000000 0x010000000  {    ; load region size_region
  ER_EROM1 0x10000000 0x010000000  {  ; load address = execution address
   *.o (ExtFlashSection)
  }
}
```

b.)

Another flash programming algorithm was added to the debugger configuration to allow the tool to program data into the SPIFI flash.

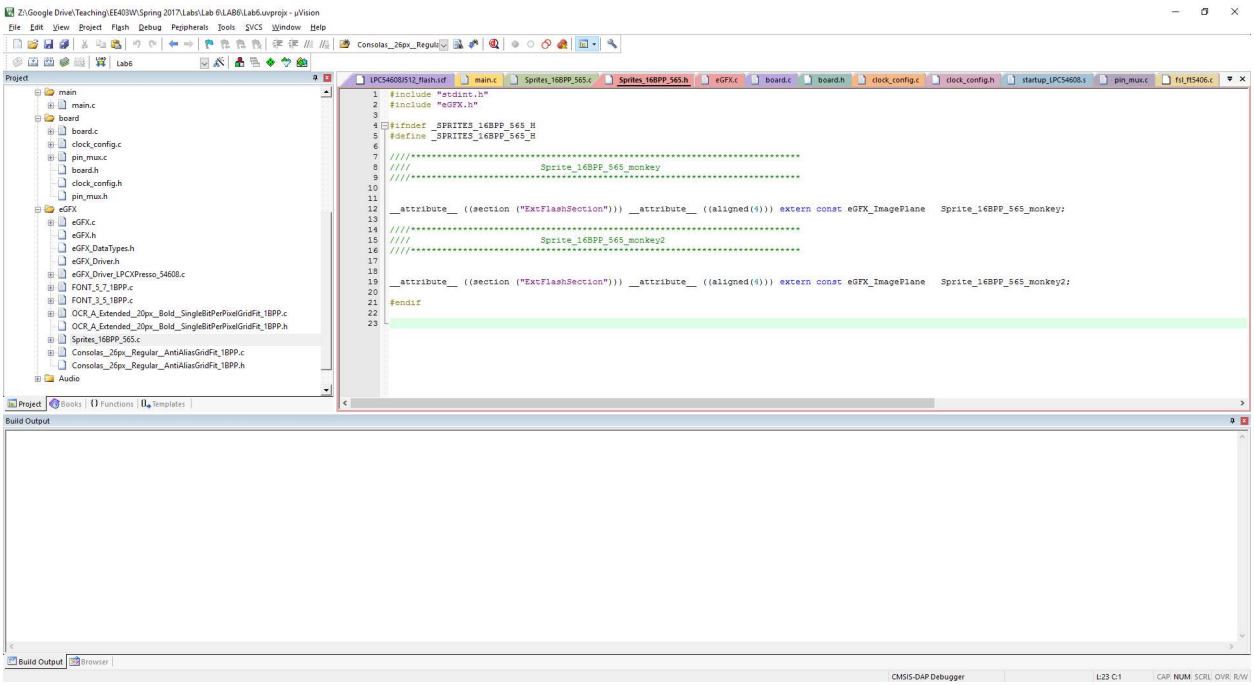


Once this is done, any variable with the decorator :

```
__attribute__ ((section ("ExtFlashSection")))
```

Gets put in this external SPIFI flash. Not to worry though, all of this process is automated.

When you run the “GenSprites.bat” file, you get a .c/.h file that looks like this:

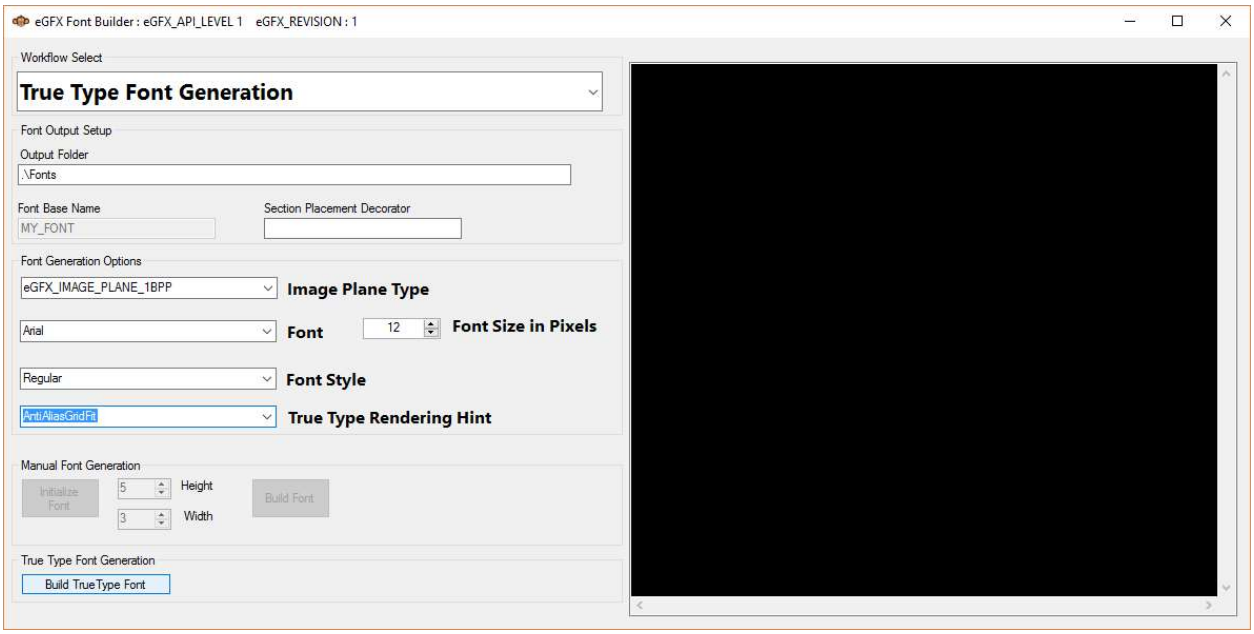


In the figure above, we have 2 image planes that we can later use with the “eGFX_Blit” function. It is easy to add more. Just drop .png images into the Sprites folder and re-run “GenSprites.bat”

2.) Adding custom fonts

A font in eGFX is a data structure that is a collection of eGFX_Imageplanes for each character. There is a program in the **“SRC\eGFX”** folder called **“eGFX_FontGen.exe”**

Just run the program and put in settings like the image below.



There will be .c/.h files you can add to project in the **“eGFX/Fonts”** folder. You can play with the settings to see what you get. The section placement Decorator should be

```
__attribute__ ((section ("ExtFlashSection")))
```

to put the font into the SPIFI flash. You do not have to do this unless your main flash memory in the LPC54608 starts to fill up. This is generally only a problem with large fonts.

If you want “smooth” anti-aliased fonts you can select “eGFX_IMAGE_PLANE_16BPP” but this takes up quite a bit of FLASH. It is good to use SPIFI in this case. Leave as “eGFX_IMAGE_PLANE_1BPP” for a smaller memory footprint.

Watch the youtube video for more details.