UTFT_SPIflash

Add-on Library for UTFT and SPIflash

Manual

Introduction:

This library is an add-on to UTFT and will not work on its own.
This add-on library also requires the SPIflash library.

This library adds a simple way to load images from SPI flash chip. The images must be contained within the SPIflash file system. Images can be added to the flash chips using the FlashUploader tool supplied with the SPIflash library.

You can always find the latest version of the library at **http://www.RinkyDinkElectronics.com/**

For version information, please refer to **version.txt**.

EXAMPLE DATASETS USED:

These files can be found in the **/SPIflash/tools/FlashUploader/Example Datasets** folder.

Full name	Short name	Minimum Flash Chip Size (Mbits)
Earth_Map.*	EARTH.SFD	32 Mbits
Earth_Map_HR.*	EARTH_HR.SFD	128 Mbits
TestImages_240x320.*	240X320.SFD	8 Mbits
TestImages_240x400.*	240X400.SFD	8 Mbits
TestImages_320x240.*	320X240.SFD	8 Mbits
TestImages_400x240.*	400X240.SFD	8 Mbits
TestImages_480x272.*	480X272.SFD	8 Mbits
TestImages_800x480.*	800X480.SFD	32 Mbits
TestImages_Mono_For_Colordisplays.*	MONO_C.SFD	2 Mbits

The specific dataset required by an example sketch it will be noted in the opening comments of that sketch.

FUNCTIONS:

UTFT_SPIflash(UTFT, SPIflash);

The main class constructor.

Parameters:

```
UTFT:     a reference to an already created UTFT object
SPIflash: a reference to an already created SPIflash object
```

Usage:

```
UTFT_SPIflash myFiles(&myGLCD, &myFlash); // Create an instance of the UTFT_SPIflash class
```

Notes: Remember the ‘&’ in front of the UTFT and SPIflash object names

loadBitmap (fileID, x, y);

Load a color image from the flash chip and display it on the screen.

Parameters:

```
fileID: ID of the file you want to open for reading
x:      x-coordinate of the upper, left corner of where to display the image
y:      y-coordinate of the upper, left corner of where to display the image
```

Usage:

```
myFiles.loadBitmap(12, 0, 0); // Load the image with fileID 12 and display it
```

Notes: Image dimensions are stored in the flash chip file system. No checking is done if the image will fit on the screen. Drawing images outside the screen may cause unpredictable results.

loadBitmap(fileID, x, y, ox, oy, sx, sy);

Load a section of a color image from the flash chip and display it on the screen.

Parameters:

```
fileID: ID of the file you want to open for reading
x:      x-coordinate of the upper, left corner of where to display the image
y:      y-coordinate of the upper, left corner of where to display the image
ox:     x-coordinate of the upper, left corner of the section in the original image
oy:     y-coordinate of the upper, left corner of the section in the original image
sx:     width of the section in pixels
sy:     height of the section in pixels
```

Usage:

```
myFiles.loadBitmap(6, 0, 0, 100, 100, 320, 240); // Load the image with fileID 6 and display a part of it
```

Notes: Original image dimensions are stored in the flash chip file system. No checking is done if the image will fit on the screen or if the section you are trying to display is within the bounds of the image. Drawing images outside the screen may cause unpredictable results.

loadMonoBitmap (fileID, x, y);

Load a monochrome image from the flash chip and display it on the screen.

Parameters:

```
fileID: ID of the file you want to open for reading
x:      x-coordinate of the upper, left corner of where to display the image
y:      y-coordinate of the upper, left corner of where to display the image
```

Usage:

```
myFiles.loadMonoBitmap(14, 50, 50); // Load the image with fileID 14 and display it
```

Notes: Image dimensions are stored in the flash chip file system. No checking is done if the image will fit on the screen. Drawing images outside the screen may cause unpredictable results.
The image will be drawn in the colors set by the UTFT functions setColor() and setBackColor().
Transparent backgrounds are supported for monochrome images.