

Raspberry Compote

Pseudo-random ramblings about programming and other geeky stuff

Wednesday, 27 January 2016

Low-level Graphics on Raspberry Pi [images]

One of the recurring [themes](#) regarding Raspberry Pi graphics is the 16-bit RGB 5:6:5 display mode vs typically 24-bit images. The '565' mode is the default on HDMI and possibly the only one available on TFT add-on displays like this [one from Adafruit](#).

There are tools that can be used to display images in the framebuffer - which do a suitable color-space conversion and do seem to work with the '565' display mode. However for simple and speedy display of images it would in many cases be useful to be able to convert the image to '565' beforehand. The usual suspect tools [Netpbm](#) and [Imagemagick](#) do not seem to include an option to convert to '565'. Only tool I know for sure has this is [GIMP](#) which is a fairly big interactive GUI application and might not suit (there is some level of scripting support though which might be worth checking out). As a side note [IrfanView](#) for Windows can read '565' raw images fine (but does not seem to have an option to save in this mode).

Fellow RPi forum user [bullwinkle](#) has [found out](#) that [ffmpeg](#) supports this mode and - while more typically used for stitching together images to form a video - can also output image files:

```
ffmpeg -vcodec png -i input.png -vcodec rawvideo -f rawvideo \
-pix_fmt rgb565 output.raw
```

I created a simple 'quick and dirty' [conversion program](#) for converting a 24-bit PPM image file to 16-bit 'raw' RGB 565 format. The colorspace conversion is fairly simple: take the top 5 bytes of 8-bit red value $r \gg 3$ (bitwise shift right, $8 - 5 = 3$), top 6 of green, top 5 of blue - and stitch back together into one 16-bit RGB $rgb = (r \ll 11) + (g \ll 5) + b$:

```
...
for (y = 0; y < height; y++) {
    for (x = 0; x < width; x++) {
        if (fread(rgb, 3, 1, fp) == 1) {
            unsigned char r = rgb[0];
            unsigned char g = rgb[1];
            unsigned char b = rgb[2];
            unsigned short rgb565 = ((r >> 3) << 11) +
                                     ((g >> 2) << 5) +
                                     (b >> 3);
            unsigned char c1 = (rgb565 & 0xFF00) >> 8;
            unsigned char c2 = (rgb565 & 0xFF);
            fwrite(&c2, 1, 1, stdout);
            fwrite(&c1, 1, 1, stdout);
        }
    }
}
...
```

Compile with:

```
#include <linux/kd.h>
#include <stdint.h>
#include "vcio.h"
#include <time.h>

// 'global' variables to store s
int fbfd = 0;
char *fbp = 0;
struct fb_var_screeninfo vinfo;
struct fb_fix_screeninfo finfo;

...

size = 0;
...

... 200
```

Blog Archive

- ▼ 2016 (6)
 - March (1)
 - February (4)
 - ▼ January (1)
 - [Low-level Graphics on Raspberry Pi \(images\)](#)
- 2015 (3)
- 2014 (9)
- 2013 (9)
- 2012 (2)

Code Repository

- [Low-level Graphics on RPi](#)

Discussion

- [Low-level Graphics on RPi](#)
- [Python Programming on RPi](#)
- [Java Programming on RPi](#)

Links

- [Raspberry Pi](#)
- [Python](#)

```
gcc -O2 -o ppmtorgb565 ppmtorgb565.c
```

And starting off with a 24-bit PNG file use `pngtoppm` from the `Netpbm` package to convert PNG->PPM, pipe the output to the new tool and save the final output to a file:

```
pngtopnm /path/to/image24.png | /path/to/ppmtorgb565 > image565.raw
```

Or maybe even direct to framebuffer (if 'disk' space is more valuable than run speed):

```
pngtopnm /path/to/image24.png | /path/to/ppmtorgb565 > /dev/fb1
```

...as (almost) everything in Linux is a file ;)

~~Disclaimer: I have not yet tested this on RPi — just on a PC running a Debian VM — will update after properly testing (or seeing comments on the RPi forum in the aforementioned thread).~~ This has now been succesfully tested on RPi. My test images can be found in [GitHub](#) too (along a PPM to 24-bit RGB '888' converter).

[Continued in next part [More palette](#)]

Posted by [Unknown](#) at [13:04](#)



Labels: [C](#), [graphics](#), [Linux](#), [Raspberry Pi](#)

No comments:

Post a Comment

Note: only a member of this blog may post a comment.

Enter your comment...



Comment as:

Lhunden (Google+ ▼)

Sign out

Publish

Preview

☐ Notify me

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)

Simple theme. Powered by [Blogger](#).