More timothy.p.dell@gmail.com Dashboard Sign Out



Pseudo-random ramblings about programming and other geeky stuff

Friday, 14 March 2014

Low-level Graphics on Raspberry Pi [part X]

Now that we have been gradually building the example program to allow us to do something interesting - how about trying a bit of animation?

```
// rectangle dimensions
```

Blog Archive

- **▶** 2016 (€
- **▶** 2015 (3)
- **2014 (9)**
- ► September (1)
- ▶ April (3)
- March (5)

Programming language (micro)

Low-level Graphics on Raspberry Pi (part X+2)

Low-level Graphics on Raspberry Pi (part X+1)

Low-level Graphics on Raspberry Pi

Restart

- ▶ 2013 (9)
- **≥** 2012 (2)

Code Repository

• Low-level Graphics on RPi

Discussion

- Low-level Graphics on RPi
- Python Programming on RPi
- Java Programming on RP:

Links

- · Raspberry P
- Python

Save as fbtestX.c (complete code in GitHub) - build with make fbtestX. This should give us a moving white rectangle that bounces off the screen sides... Unfortunately the updates are not smooth (at least on most displays) - there is a quite prominent tearing effect.

Linux framebuffer interface does define some methods to overcome this - we could make the framebuffer virtual size double the height of the (smaller) physical one using the FBIOPUT_VSCREENINFO call:

```
// Set variable info
vinfo.xres = 640; // try a smaller resolution
vinfo.yres = 480;
vinfo.xres_virtual = 640;
vinfo.yres_virtual = 960; // double the physical
vinfo.bits_per_pixel = 8;
if (ioctl(fbfd, FBIOPUT_VSCREENINFO, &vinfo)) {
   printf("Error setting variable information.\n");
}

//long int screensize = vinfo.xres * vinfo.yres;
// have to use the virtual size for the mmap...
long int screensize = vinfo.xres_virtual * vinfo.yres_virtual;
```

And change our drawing loop to use the two halves of the virtual buffer using a call to FBIOPAN_DISPLAY for page-flipping tied to a vertical sync using FB_ACTIVATE_VBL:

```
int vs = 0;
// initially show upper half (0) - draw to lower half (1)
int cur_half = 1;
for (i = 0; i < 1000; i++) {

    fill_rect(x, y, 40, 40, 4);

    x = x + dx;
    y = y + dy;

    if ((x < 0) || (x > (vinfo.xres - 40)) {
            dx = -dx;
            x = x + 2 * dx;
    }
    if ((y < 0) || (y > (vinfo.yres - 40)) {
            dy = -dy;
            y = y + 2 * dy;
    }

    // switch page
    vinfo.yoffset = cur_page * vinfo.yres;
    vinfo.activate = FB_ACTIVATE_VBL;
    if (ioctl(fbfd, FBIOPAN_DISPLAY, &vinfo)) {
            printf("Error panning display.\n");
    }
}
```

Unfortunately these calls have not been implemented in the RPi framebuffer driver (does not seem to be in later versions either yet, see also http://www.raspberrypi.org/phpBB3/viewtopic.php?f=67&t=19073). So running the above code (full code) results in repeated output of Error panning display. and the rectangle flashing even worse (as we miss every second screen update by drawing outside of the visible area).

[Continued in next part]

[UPDATE on changes in the fb driver here]

Posted by Unknown at 12:19

Labels: C, graphics, Linux, Kaspberry Pi

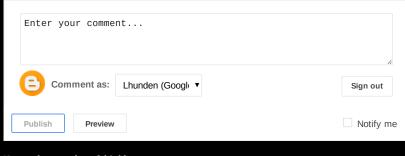
1 comment:



Raspberry Compote / Wednesday, 19 March 2014 at 18:06:00 GM

Switching the ioctl(fbfd, FBIOPAN_DISPLAY, &vinfo) to ioctl(fbfd, FBIOPUT_VSCREENINFO, &vinfo) seems to work slightly better - it does flip the page to the yoffset, but unfortunately it also redraws the console buffer, so one may end up with flashing directory listing or whatever was on display when starting the tester app...

Reply



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

Newer Post Home

Subscribe to: Post Comments (Atom)

Simple theme. Powered by Blogger.