Subject: Is Rainbow/FrameBuffer/XXXXFB broken?
Posted by Mindtraveller on Sun, 07 Dec 2014 00:54:59 GMT
View Forum Message <> Reply to Message

Recently I've made an attempt to look Rainbows-based and WinFB examples, but they failed to compile.
reference/UWord_FB seems to be broken too.
Is it some kind of upgrade going on? Or framebuffer-based stuff is completely broken in latest revisions?

By the way, I consider making framebuffer-like front-end for e-ink displays. It's 4 bits per pixel and needs manual updates of display. How do you think, will it be possible to implement it without changing any of Ctrl or any other U++ internals?
Thanks.

Subject: Re: Is Rainbow/FrameBuffer/XXXXFB broken?
Posted by mirek on Sun, 07 Dec 2014 10:02:54 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Sun, 07 December 2014 01:54Recently I've made an attempt to look Rainbows-based and WinFB examples, but they failed to compile.
reference/UWord_FB seems to be broken too.
Is it some kind of upgrade going on? Or framebuffer-based stuff is completely broken in latest revisions?

By the way, I consider making framebuffer-like front-end for e-ink displays. It's 4 bits per pixel and needs manual updates of display. How do you think, will it be possible to implement it without changing any of Ctrl or any other U++ internals?
Thanks.

Yes, it might be broken now. There were changes in CtrlCore; so far I have only fixed "canonical" backends. But should be easy to fix (might do it soon).

e-ink: definitely possible. But I would say that due to manual updates, your GUI will have to be changed a bit anyway....

In any case, I would probably start by thinking about some screen buffer (with all pixels) in application and some mechanism how to put that data to e-ink.

4-bits is not a problem at all, just make convert RGBA to monochorome... (maybe you might want some dithering too).

Subject: Re: Is Rainbow/FrameBuffer/XXXXFB broken?
Posted by Mindtraveller on Sun, 07 Dec 2014 16:08:00 GMT
View Forum Message <> Reply to Message

Quote:e-ink: definitely possible. But I would say that due to manual updates, your GUI will have to be changed a bit anyway....
In any case, I would probably start by thinking about some screen buffer (with all pixels) in application and some mechanism how to put that data to e-ink.
Due to the nature of e-ink display, the most tricky part of frame-buffer functionality is partial updates. Very much like you use in Turtle where (I suppose) you update only the refreshed Ctrl pixels. It will be great if it will be possible to use partial updates mechanism from Turtle to optimize rendering speed.
Talking about dithering, yes on-the-fly conversion to 4bpp is not the best option in the sense of efficiency and memory consumption (the code will be executed on ARM CPUs). But it may be the only option which keeps U++ internals intact.