PLASTIC LOGIC

08. June 2016

Application Note

For Plastic Logic's UC8156 based displays

"Controlling the Active Border"

The behaviour and refresh of the Active Border (AB) can be controlled by using Reg[1Dh]:

(30) VBorder Setting (Index: 1Dh) (Default: 30h)

Action	R/W	D7	D6	D5	D4	D3	D2	D1	D0
Vborder Setting	R/W	VBST[3:0]				VBLV[1:0]		VBINIT	VBEN

VBST[3:0]: Selection VBD transitions.

VBST	VBD Transitions	VBST	VBD Transitions
0000	GS0→GS0	1000	GS2→GS0
0001	GS0→GS1	1001	GS2→GS1
0010	GS0→GS2	1010	GS2→GS2
0011	GS0→GS3	1011	GS2→GS3
0100	GS1→GS0	1100	GS3→GS0
0101	GS1→GS1	1101	GS3→GS1
0110	GS1→GS2	1110	GS3→GS2
0111	GS1→GS3	1111	GS3→GS3

VBLV[1:0]: Selection VBD level during "non-update time".
00: HiZ
01: VCOM
1x: GND

VBINIT: Selection which transitions for initialize update.
0: Use initial transitions. (R13h INITTS)
1: Use VBD transitions.

VBEN: Selection whether to switch the border during next update or not.
0: VBD update Disable.
1: VBD update Enable.

This command can be active only when DBUSY = "0".

Typically the Active Border is already "pre-driven" to a target color by Plastic Logic before shipment.

Register overwrite after Power-on

If you want to keep this color and not update the AB anymore, you need to change the register value after power-up:
Reg[1Dh]=04h -> VBD level during "non-update time" = VCOM
(not doing this might cause the AB to "drift" towards darker or lighter greylevels during updates)

Change color of Active Border

If you like the change the AB's color you need to choose the relevant VBD Transition first, like:
Example 1: AB is currently Black (GS0) and AB should get switched to White (GS3) → GS0→GS3 → VBST='0011'
Example 2: AB is currently White (GS3) and AB should get switched to Black (GS0) → GS3→GS0 → VBST='1100'
In addition you need to set:
VBINIT='1'
VBEN='1'

The actual AB update is done during the next display update triggered by Reg[14h].bit0 (DWTRG).

Example source code:

```
void drive_active_border_black()
{
	spi_write_command_1param(0x1d, 0xC7);

	UC8156_HVs_on();
	UC8156_update_display(FULL_UPDATE);
	UC8156_HVs_off();

	spi_write_command_1param(0x1d, 0x04);
}

void drive_active_border_white()
{
	spi_write_command_1param(0x1d, 0x37);

	UC8156_HVs_on();
	UC8156_update_display(FULL_UPDATE);
	UC8156_HVs_off();

	spi_write_command_1param(0x1d, 0x04);
}
```