PLASTIC LOGIC

# How to integrate VCOM calibration

**Description:** How to integrate VCOM calibration in an E-Paper system.
**Date:** 20 November 2013
**Revision:** v002

## Table of contents

# 1 What is VCOM?

The VCOM voltage is applied to a common electrode for all pixels on top of electrophoretic displays (EPD). If the value is too high, a Plastic Logic display will look lighter; if the voltage is too low the display will appear darker.

As part of the manufacturing process, the **kick-back** voltage is measured: it is a property of each display. The VCOM voltage should match the kick-back value when the display is being driven in a product in order to obtain optimal performance. So for each display manufactured, there is an associated VCOM voltage provided with it.

# 2 What is VCOM calibration?

Each display power supply unit (PSU) also has different characteristics and electronic tolerances which need to be compensated in order to generate an accurate VCOM output voltage. This process is called VCOM calibration, which can be typically done in two ways:

### *Dynamic closed-loop VCOM calibration*

In this mode, the calibration is performed **on the device itself** using an ADC. The software will measure the characteristics of the VCOM PSU and automatically compensate them in a closed-loop fashion.

### *Static open-loop VCOM calibration*

This approach has the advantage of a reduced number of electronics components in the product as no ADC is needed. Instead, the VCOM PSU characteristics are measured **in the factory and** stored for each electronics board. On evaluation systems, the characteristics are stored in an EEPROM, but any non-volatile memory which can be associated with a given PSU can be used in principle.

> ***Note***
>
> The static method is currently the preferred one as it is the most efficient in terms of cost and complexity of the product electronics.

# 3 Static VCOM calibration

## 3.1 Principle

This method can be summarised with the following steps:

1. Measure the PSU calibration data for each board during manufacturing
2. Store the PSU calibration data into the product (i.e. board EEPROM)
3. On the device, retrieve both the display VCOM voltage and PSU data
4. Compute the compensated VCOM voltage and configure the PSU accordingly

The PSU calibration data contains the following values:

**DAC x1**

First DAC register value (usually 25% of the full range)

**DAC y1**

Output VCOM voltage associated with DAC x1

**DAC x2**

Second DAC register value (usually 75% of the full range)

**DAC y2**

Output VCOM voltage associated with DAC x2

**VGPOS**

Positive gate power supply voltage in mV

**VGNEG**

Negative gate power supply voltage in mV

The VCOM voltage compensation is a linear scaling [1] of the provided "ideal" VCOM voltage `V0` using the ideal and measured differences between `VGPOS` and `VGNEG`. This difference is referred to as `VGSWING`. On Plastic Logic Hummingbird Z6 and Z7, the ideal `VGSWING` is 57089mV. So the compensated VCOM voltage `V1` in mV is:

```
V1 = V0 * (VGPOS - VGNEG) / 57089
```

Then the value to use in the VCOM DAC register for a given voltage `reg(v)` is determined using a linear model based on the two pairs of values:

```
dx = x2 - x1
dy = y2 - y1
offset = y1 - (x1 * dy / dx)

reg(v) = (v - offset) * dx / dy
```

## 3.2 Using Plastic Logic reference designs

This is automatically handled in Plastic Logic's reference software, using an first EEPROM on the Hummingbird Z6 and Z7 to store the PSU calibration data and a second EEPROM on the Type18 4.0" displays to store the VCOM voltage (among more display data). With the reference designs, displays and systems without an EEPROM use an SD card instead.

### *Generating the EEPROM blob*

The `mkvcomblob.py` Python script can be used to produce a binary file which can then be programmed into an EEPROM device. It is part of the pldata Python module, which is a PLSDK component, and can be run on any Linux system with Python v2.7 installed (i.e. PC, BeagleBone...):

```
python mkvcomblob.py DAC_X1 DAC_Y1 DAC_X2 DAC_Y2 VGPOS VGNEG
```

For more details regarding `mkvcomblob.py`, please refer to the "Plastic Logic Software Manual".

### *Programming the board EEPROM*

The `plhwtools` command line application can be used to programme an EEPROM device. It is part of the PLSDK and is included with Plastic Logic evaluation kits. This is a sample command to run on BeagleBone:

```
plhwtools -a 0x50 -b /dev/i2c-3 eeprom 24c01 f2e vcom-blob.bin
```

For more details regarding `plhwtools`, please refer to the "Plastic Logic Software Manual".

## 3.3 Measuring the calibration data

For the Plastic Logic reference hardware design boards Hummingbird Z6 and Z7, the VCOM PSU characteristics are measured and programmed into the EEPROMs by Plastic Logic. For an alternative implementation, this process may need to be performed independently.

The generic instructions below describe how to collect the necessary information. To minimise errors in the resulting generated VCOM voltage when the data is used, the accuracy of the measured voltages should be of the order of **1mV for the VCOM PSU** output and **10mV for the gate PSU**.

The measurements have to be made while the display high-voltage PSU is turned on. Depending on the system, this may involve running some software or electrically overriding a signal to force the PSU to be enabled. If little manual control is available on a final product, then a continuous update can be run. The VCOM switch can be left open, if possible.

### *VCOM PSU characteristics*

1. Turn the display power on.
2. Write a first value into the VCOM DAC register, usually 63 for an 8-bit DAC or 127 for a 9-bit DAC; this value is `x1`.
3. Measure the output voltage of the VCOM PSU in mV; this gives `y1`.
4. Write a second value into the VCOM DAC register, usually 190 or 381 respectively for 8-bit and 9-bit DACs; this is `x2`.
5. Measure the output voltage of the VCOM PSU in mV; this gives `y2`.

### *Gate PSU characteristics*

6. While the display power is left on, measure positive gate voltage `VGPOS`.
7. Similarly, measure the negative gate voltage `VGNEG`.
8. Turn the display power supply off.

## 3.4 Example using a BeagleBone and a HBZ6 board

This procedure is a fully functional example based on the following set of hardwdare:

- BeagleBone with Plastic Logic standard software build + 5V PSU
- Either USB-UART or Ethernet interface to connect to the kit from a PC
- Hummingbird Z6 (HBZ6) electronics board + ribbon cable
- Type18 4.0" Plastic Logic display

The objective is to programme the Z6 EEPROM with measured calibration values. This can then be tested on a Parrot board with the same Z6 board and any Type18 display. For this to be useful, the display EEPROM should also be programmed as it should contain the ideal VCOM voltage to be used with the display.

The calibration is achieved using command line tools available on the evaluation kit such as `plhwtools`. It makes use of I2C control interfaces to communicate with the hardware. The BeagleBone has several I2C busses, so `plhwtools` needs to be configured with the one to use. It can either be passed to the `-b` option, or set in the PLSDK configuration file in `~/.plsdk.ini` to simplify the command line syntax:

```
[libplhw]

i2c-bus=/dev/i2c-3
```

Then the following commands can be run in a console to gather the calibration measurements using the TI TPS65185 HV-PMIC available on the HBZ6 board:

```
# 1. stop the EPD driver and turn on the HV manually
/etc/init.d/epdc stop
plhwtools tps65185 active

# 2. write the first DAC register value (127 is 1/4 of full range)
plhwtools tps65185 vcom 127

# 3. measure the first voltage on the HBZ6 VCOM test point (1mV accuracy)
# for this example: 4185 mV

# 4. write the second DAC register value (381 is 3/4 of full range)
plhwtools tps65185 vcom 381

# 5. measure the second voltage on the HBZ6 VCOM test point (1mV accuracy)
# for this example: 12528 mV

# 6. measure the VGPOS voltage on the HBZ6 VGP test point (10mV accuracy)
# for this example: 25090 mV

# 7. measure the VGNEG voltage on the HBZ6 VGN test point (10mV accuracy)
# for this example: -33860 mV

# 8. turn the HV off
plhwtools tps65185 standby
```

Finally, to generate the VCOM blob and programme it into the HBZ6 EEPROM using the command line utilities from the PLSDK and still on a BeagleBone:

```
# generate the VCOM EEPROM blob (using default file name vcom-blob.bin)
# Important: The DAC register values are in hexadecimal: 127=0x7F, 381=0x17D
python /usr/lib/python2.7/pldata/mkvcomblob.py 7F 4185 17D 12528 25090 -33860

# optionally, create a backup of the current VCOM EEPROM contents
plhwtools -a 0x50 eeprom 24c01 e2f vcom-blob-backup.bin

# programme the VCOM EEPROM chip with the generated data
plhwtools -a 0x50 eeprom 24c01 f2e vcom-blob.bin
```

The EEPROM can now be used with the reference software which supports the static VCOM PSU calibration. For example, the `pl-mcu-epd` software on a Parrot board will print these messages:

```
psu-data: dac[0x7f]=4185, dac[0x17d]=12528, vgpos=25090, vgneg=-33860
vcom    : input: 5124, scaled: 5310, DAC reg: 0xA1
```

PLASTIC LOGIC

### *Automating the procedure*

While the command line tools are convenient for a manual procedude as described above, it is also possible to directly use the `libplhw` library and the `pldata.mkvcomblob` Python module to automate the procedure within a Python script and a specific C application.

---

1 Linear scaling is an approximation which gives acceptable results when the scaling factor is less than about 2% (i.e. 200mV for a 10V VCOM).