Fast update Application Guide

Doc. No. PDW001 Rev. **02** (Nov 2025)

Introduction

E-paper displays (or EPDs) feature extremely low power, allowing it to be used in devices once the content has been rendered then no power is needed to maintain the display. The normal and standard refresh cycle for an EPD calls normal update provides the best optical performance.

Normal update satisfies most of the use cases. However, the refresh time cannot meet the interactive scenario and the use case of a device for the Internet of Things. This application guide will introduce the fast update provides an overall introduction including technology, operating approach and limitation to help you understand how to realize it in your applications.

Copyright

龍亭新技股份有限公司 Pervasive Displays Inc.

4F, No. 28, Chuangye Rd., Tainan Science Park, Tainan City 74144, Taiwan (R.O.C.)

Tel: +886-6-279-5399

https://www.pervasivedisplays.com

Content

List of figures

List of tables

General description

1.1. Overview

This section will introduce the materials and drivers that have been used on the products of Pervasive Displays Inc (PDi). After generally understanding the most of knowledge, we will start introducing the driving difference between normal update and fast update. Afterward we will explain how to implement the fast update in different driver solution. At the last section, you will learn about the pros and cons of this technology using e-Paper display (EPD) to be realized in your applications.

Prerequisites:

1. You have worked or tested our EPD modules and the screen has been refreshed successfully.

2. You have a rough understanding of the characteristics of EPD and already known that EPD cannot play animation effects because fast update is a special and non-standard driving approach for EPD.

3. You must follow our instructions to implement fast update from this document, and you are solely responsible for the result of any self-modification.

1.2. E ink imaging film (FPL)

E ink imaging film is also called FPL (Front Panel Laminate) has two main material films for industrial applications that are different from the FPLs used in e-readers (e.g. Amazon Kindle). The name of two FPLs are Aurora and Spectra. It can be subdivided into the following different FPL types which have been mass supplied for vendors to make as EPD modules.

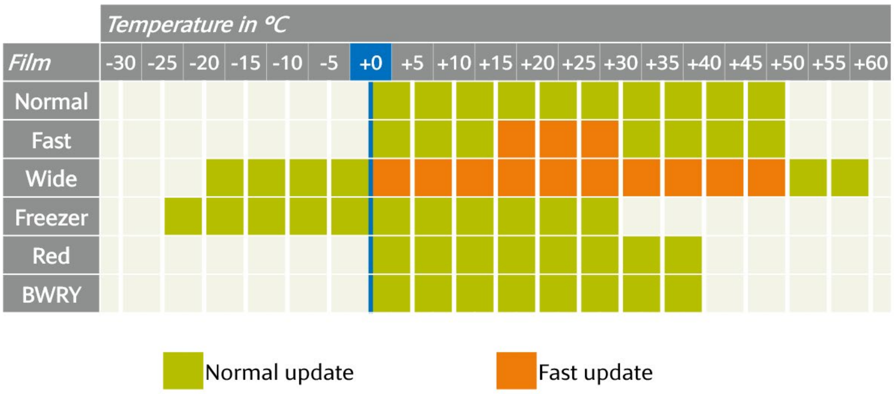


Figure 1 - Temperature ranges for different films

Notes

1. The wide-temperature e-paper modules are designed for consistent optical performance in ambient temperatures from -20 °C to +60 °C with normal update and +0 °C to +50 °C with fast update.

2. The Freezer e-paper modules support -25 °C to +30 °C with normal update.

3. The lower temperature, the slower refresh rate of EPD screen. For this reason,

1.3. Driver IC (CoG) and driving waveform

Driver IC is a timing controller (TCON) to output different sources/data and control the gates per pixel on EPD. It's always bonded on TFT backplane which is also called CoG (Chip on Glass).

The output image on an EPD is realized by controlling the pixel electrode voltage, thus affecting motion of charged particles in the neutral suspension. A voltage sequence applied to the pixel electrode is called a **driving waveform**. Different manufacturer of EPD selects different driver IC and the E ink FPL is always varied batch by batch. Therefore, each design of a driving waveform is completed manually and always need to be tuned from different batch of FPL. Simultaneously, under different temperature conditions, more sets of driving waveform matched with different temperature conditions are required. The motion of particles in an EPD device will be inadequate or overdriven when the driving waveform does not match with the temperature conditions. If so, user will see image ghosting or blurred pixels are easily generated, which may affect the output image quality. The design of the driving waveform will determine the image quality of the EPD.

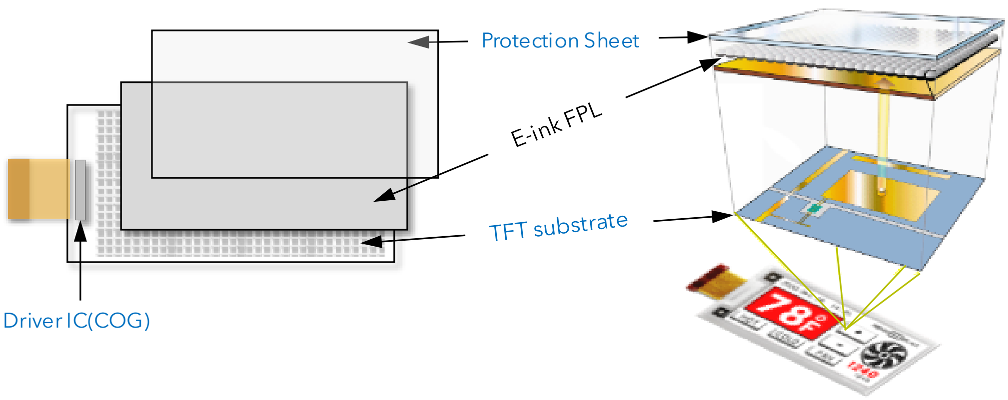


Figure 2 - Layer structure of EPD module

Charge pump, a kind of DC-to-DC converter (DC/DC) that uses capacitors for energetic charge storage to raise or lower voltages. A positive charge pump and a negative charge pump to provide adjustable regulated output voltages. EPD needs charge pumps to step up the voltages to supply different voltage levels to drive different colour pigments in EPD module. When the DC/DC is embedded in the CoG and driving waveform is pre-programmed in CoG, we call it internal timing controller (iTC), and similarly, if the DC/DC circuit is arranged outside the EPD module and driving waveform is controller from MCU, we call it external timing controller (eTC).

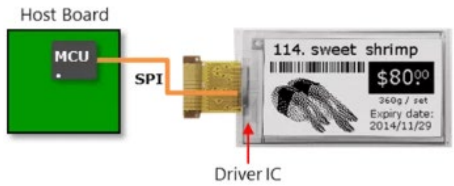
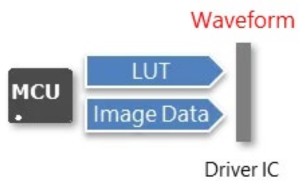
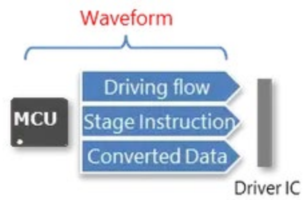
<table>
<tr><th>Driving Method</th><th>iTC</th><th>eTC</th></tr>
<tr><td>EPD with host board</td><td colspan="2">
</td></tr>
<tr><td>Waveform Location</td><td>
</td><td>
</td></tr>
<tr><td>Driving Waveform</td><td>Embedded in driver IC (CoG)</td><td>Controlled by MCU</td></tr>
<tr><td>Customization / Design Flexibility</td><td>Normal</td><td>Higher</td></tr>
<tr><td>Design-in Effort</td><td>Easier</td><td>Normal</td></tr>
</table>

Table 1 - iTC and eTC comparison

Notes

1. PDi provides both approaches of timing controller for EPD modules. The eTC model is PDi's unique structure, which is different from the vendors on the market.
2. The eTC EPDs have gradually declined or reached end-of-life. Pervasive Displays has discontinued further development of eTC model EPDs after 2025.
3. The driving waveforms in this document are 1-bit greyscale which means black and white colours only.

1.4. Normal and fast update

Every EPD manufacturer or design house (vendor) will design their own driving waveforms or get the existing waveforms from E ink company directly to drive their produced EPD modules. Because of this, you will find the effect of screen flashing varies from every vendor. Not only that, but the optical quality performance of each vendor is different, the power consumption either.

To ensure optimal image quality, Pervasive Displays customises the driving waveforms for each batch of iTC screens and saves them in the driver IC.

In this section, we will start introducing normal update and fast update. Note that PDi's waveform stages may be different from what you've seen from other vendors. PDi always fine tune the driving waveforms and LUTs of EPD modules by own technologies and patents.

- **Normal update** (also known as "Global update", "Full update" or "Global refresh") to update display from original image to new image and looks like every pixel of entire display has been refreshed and updated with flickering effect.
- **Fast update** (also known as "Fast refresh") is the process to update display from original image to new image directly and the new image is always a full image but looks like only the pixels to be changed are refreshed and the rest of pixels are not changed. fast update still keeps sending full new image and driver IC will work on the image data comparison to carry out the local changes.
- **Partial update** (also known as "Local update" or "Partial refresh"). The difference between fast update and partial update is partial update will need to define an area (window) to be updated and system side will send image data of the specific area only (not a full image) then using the same driving waveform to carry out the local update.

	Original image on screen	**Final image on screen**
	1 2 3 4	1 2 3 5
Update mode	**Image data**	**Updated on screen**
Normal	1 2 3 5	1 2 3 5
Fast	1 2 3 5	1 2 3 5

Table 2 - Normal and fast update

Notes

1. In this application guide, we will introduce fast update only.

2. The main difference between normal update and fast update is fast update will reduce the driving stages and shorten the stage time to carry out faster refresh than a normal update. Find the comparison in Table 4.

3. The partial update driving needs special configuration and driving approach to define the partial window (the area to be changed) before sending the specific size of image data.

4. In order to preserve optimal image quality, Pervasive Displays has decided to deprecate partial update. Partial update is just a bit faster than fast update but raises image quality issues with ghosting more often.

1.5. Buffers

Update can use one or two buffers.

- **Two buffers**: pre-storing the original image in memory (1st buffer) and compares with the new image (2nd buffer) to get the target data bytes to be refreshed on EPD. Once the image is updated on EPD, the new image will be defined as original one.
- **One buffer**: refreshing the new image on EPD directly without compensating from the original image.

Assume we will refresh EPD screen from image 1 to image 2 as follows:

Image 1 (old)	Image 2 (new)
1	2

Table 3 - Old and new images

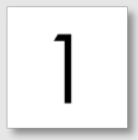
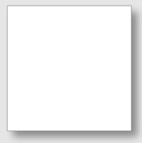
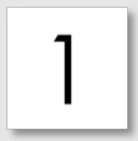
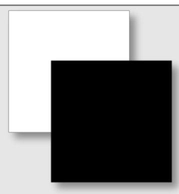
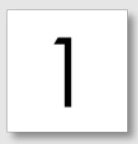
Refresh mode	Image 1	Stage 1	Stage 2	Stage 3	Stage 4
Normal update (two buffers)					
	Original image	Inverse original image	Full white image	Inverse new image	New image
	Normal update features full refresh on every pixel. Each pixel needs to be compensated (inversed from original image and inversed from new image) and reset (full white screen) to keep almost the same moving distance for every particle in the capsule of FPL. It helps for improving ghosting (will introduce later) effect and extend the lifetime of FPL.				
Normal update (one buffer)				N/A	
	Original image	Toggle black/white	New image		
	In this case, when new image is sent to driver IC, the waveform will toggle between full black and full white images continuously. After completed with the stage 1 period, the waveform will send targeted image to complete driving period of the stage 2. This approach will reduce half of driving time and achieve the compensation with reset of every particle in the capsule of FPL too.				
Fast update (two buffers)			N/A		
	Original image	New image			
	As you can see the approach in this case, the fast update looks like it is sending new image to screen directly. However, there is neither compensation nor reset stages in between. Such operations will cause the particles in FPL to be overdriven in an unbalanced state. Over a long period of time, ghosting image may appear, and at worst, the EPD module will have a short lifetime or even cause damage that cannot be recovered to original optical performance.				

Table 4 - Refresh modes and buffers

In the later sections, we will introduce how to correct drive the waveform for fast update.

- Why fast update needs two buffers?

Fast update needs to compare with original image and new image to get different image data and sends the appropriate driving data bytes to EPD. Either storing images in the Flash or allocating two memory banks to implement the data comparison. iTC and eTC will use different approaches. Find Section 2 for more details.

- What will happen if incorrect driving or overdriving with the EPD?

You will see similar vertical lines (see the picture) or the ghosting images.

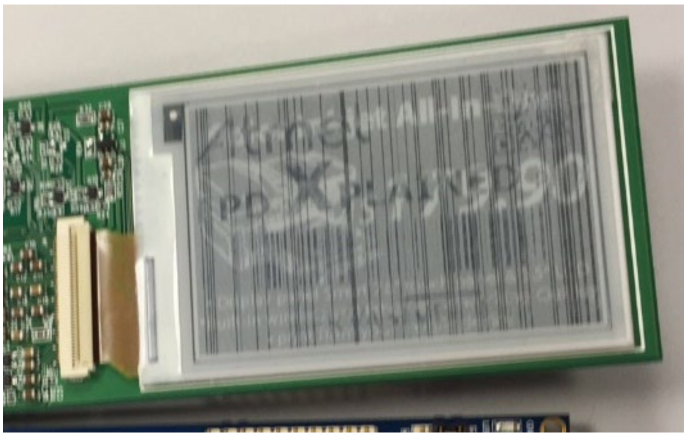


Figure 3 - Screen with vertical lines

Other consequences include

- Overlapping images obviously after powering off
- Reduce the lifetime of EPD
- Hard to clear the ghosting images permanently
- Abnormal behaviours in lower temperature

1.6. Ghosting image

Ghosting is the effect of seeing artifacts of a previous image on the display. EPDs are by nature prone to ghosting effects if the driving waveform was not well-behaved implemented. The ghosting image looks like a grey shade of the previous object is embedded in the new object.

This example is what we expected to see the "2" was changed from "B" properly.

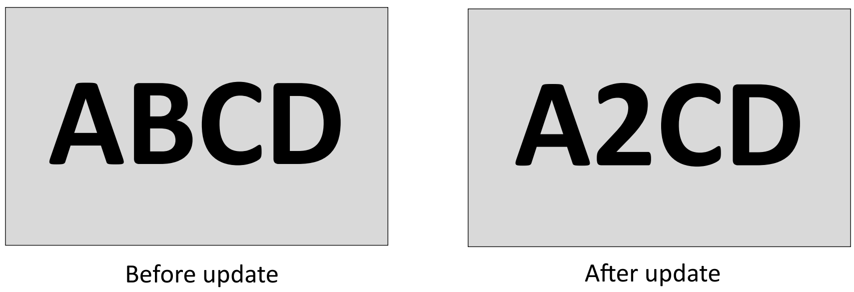


Figure 4 – Update patterns

When you see the phenomenon like the example below after new image is updated, it is the ghosting effect. Sometimes, the colours are mixed.

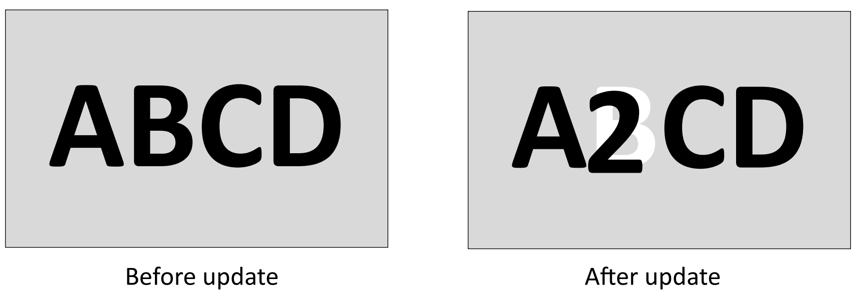


Figure 5 – Ghosting phenomenon after image is updated

To avoid seeing the ghosting image, following the instructions in section 2 will reduce or eliminate this phenomenon.

1.7. Power considerations

Being faster, fast update requires less power than normal update.

The following experiment was run with 2.66” EPD with wide temperature and embedded fast update connected to a Raspberry Pi Pico RP2040. The frame-buffer was in MCU SRAM and the fonts in MCU Flash. The same images were displayed at the same ambient temperature of 25 °C. The Nordic Power Profiler Kit II recorded the power consumption.

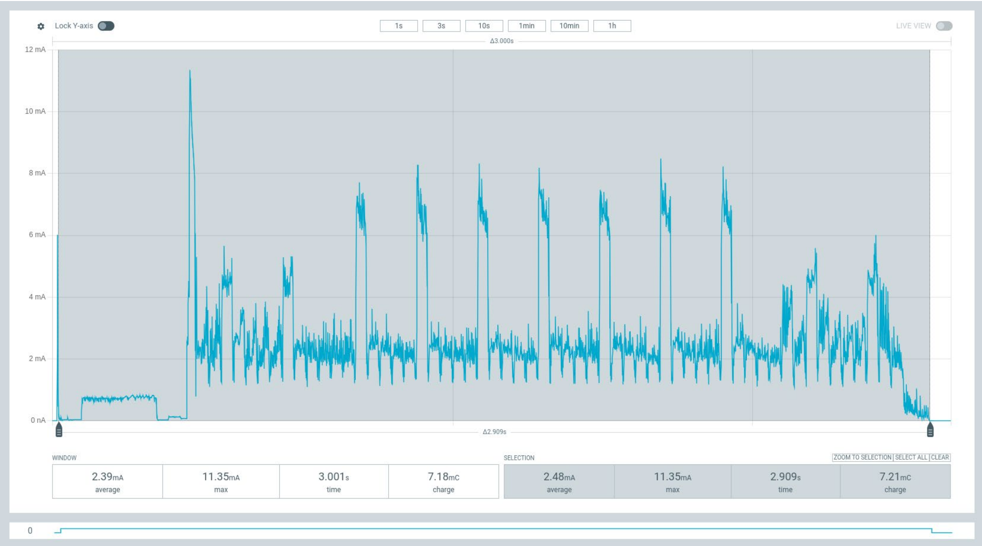


Figure 6 - Power trace for normal update

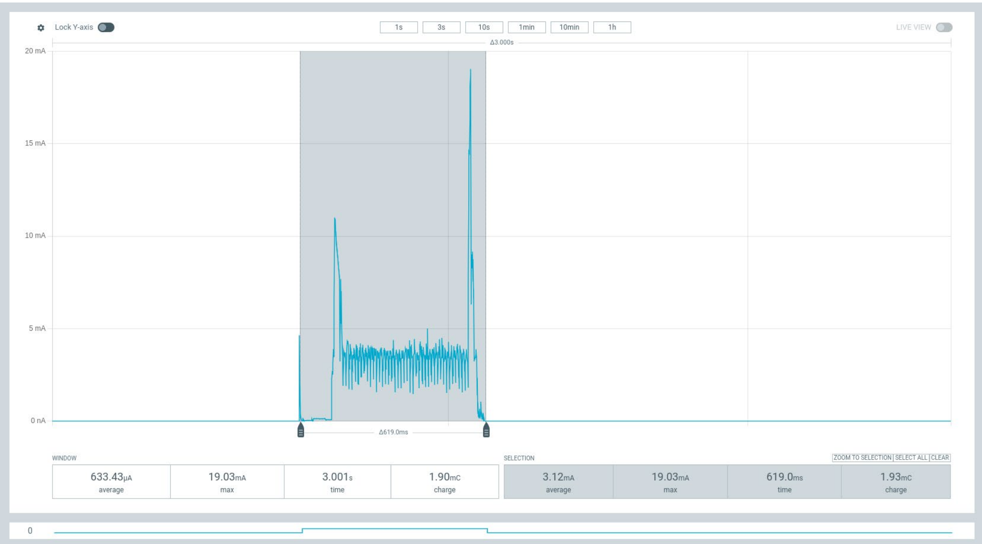


Figure 7 - Power trace for fast update

Screen	Refresh			Sleep		Day	Year
Technology	Current	Duration	Energy	Energy	Energy	Energy	Energy
	mA	*s*	*mAs*	*mAs*	*mAs*	*mAh*	*mAh*
EPD 2.66" BW Fast update	2.459	0.610	1.500	0.001	92	0.026	9.3
EPD 2.66" BW Normal	3.110	4.000	12.440	0.001	158	0.044	16.0

Table 1 - Comparison of screen technologies based on 6 updates per day

For the same EPD 2.66” with wide temperature and embedded fast update, fast update goes 6.6x faster and requires 88% less energy than normal update.

Implementing fast update

1.8. Data comparison of fast update

In section 1, we've understood the fast update needs two data buffers to store image data. One is for original (previous) image and the other is for new image.

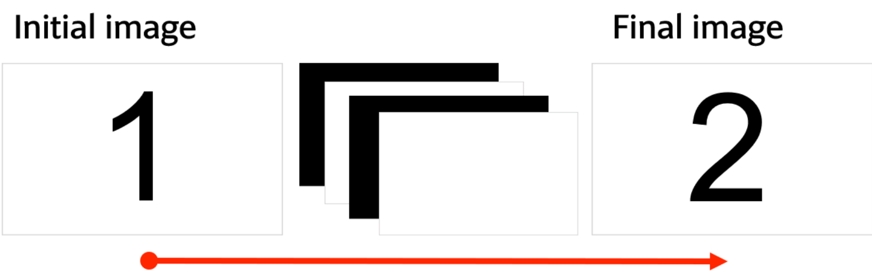
For data comparison, find the table below:

Fast update		**Previous Image**	
		Black	White
New Image	White	**White**	**Nothing** **(do not change)**
	Black	**Nothing** **(do not change)**	**Black**

Table 5 - Data comparison for fast update

- If the new data byte is same as previous data byte, send "Nothing" data byte.
- If the new data byte is different from the previous data byte, send the new data byte.
- Once new image is updated on EPD, the result image becomes the original one in order to be compared with upcoming new image.

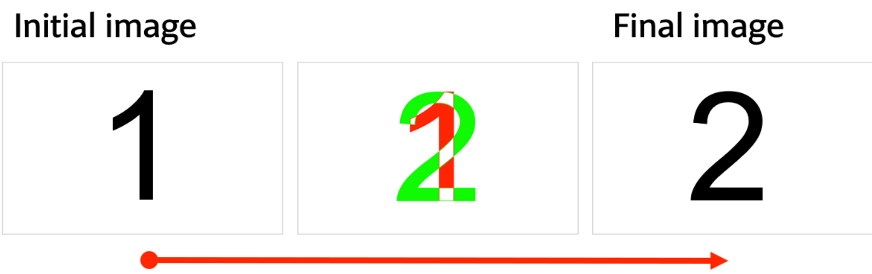
The key is to ask driver to send Nothing data byte for the targeted pixel if the existing colour will not be changed. The driver will keep the same voltage level to maintain the particles at the same pixel position. Find below is an example to compare the images pixel by pixel and get different data bytes that will be driven.



Full cycle

- Image to black
- Black to white
- White to image

Table 6 - Normal update image comparison



Only changes are applied

- White to black
- Black to white
- Unchanged

Figure 8 – Fast update image comparison

1.9. Preparing images

1.9.1. Sending image data to eTC driver

For an eTC EPD to prepare the image data, MCU side needs to get the original image, prepare the new image and get the compared result to generate the (1) Black data bytes, (2) White data bytes and (3) Nothing data bytes. Afterwards, according to the data bytes, MCU sends the corresponding data bytes to eTC driver pixel by pixel to complete a stage driving.

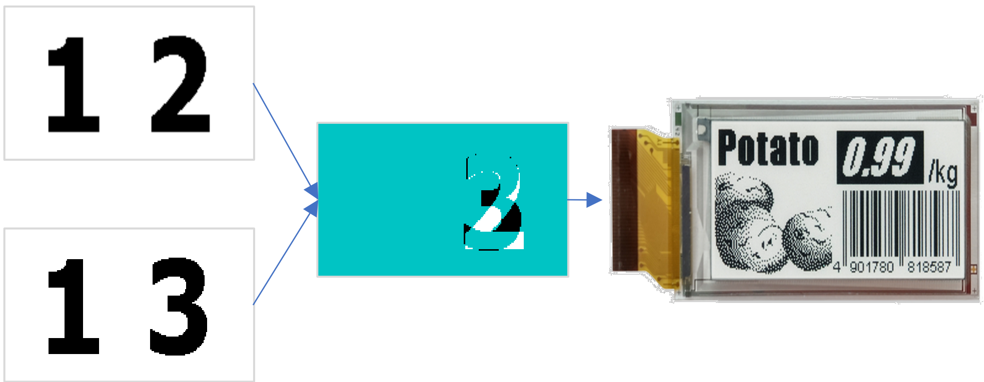


Figure 9 - Image data sent to an eTC driver

Developer always has to convert the output data bytes in advance from original and new images before sending to eTC driver.

1.9.2. Sending image data to iTC driver

The Look-Up Tables (LUTs) is an array to store the driving parameters of timings, data bytes, colour options, cycle times and more. LUTs have been programmed in the iTC driver IC of EPD, hence developer just needs to send image data with very few commands to realize the fast update.

Figure 10 - Image data sent to an iTC driver

The comparison works have been defined in the LUTs. Developer will need to send previous and new images together with the update command for applying fast update LUT to iTC driver.

1.10. Frame time and stage time (for eTC only)

The EPD module described in this document is an active dot-matrix display. It is interlaced of several sources and gates design. Per a resolution of 200 x 96 EPD, it is composed of 96 sources and 200 gates also means one line has 200 dots. One frame is from `(1,1)` to `(200, 96)` which is a complete screen size.

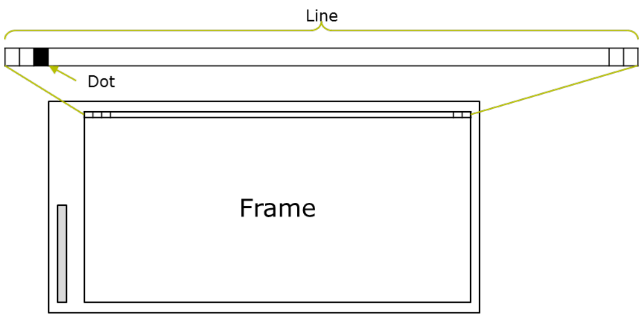


Figure 11 - Frame, line and dot of an EPD

A frame time is the duration to complete one update frame and a stage time is a total duration to update the defined number of frames.

EPD needs the particles in ink to be driven with the targeted voltage level for multiple times to push it at the targeted position steadily. The more update frames in an update stage, the better optical performance and particles fixity.

1.10.1. Determine the frame time of eTC driver

Per PDi's experiments, we recommend the stage time needs **400 ms** above to achieve the more stable and better visual experience.

For eTC EPDs, in terms of suggested 400 ms of stage time, you have to send frames in a stage as more as you can. You can get the frame time of MCU from your code then you will understand how many frames at least to be sent in a stage. For example, if you get MCU frame time is 55 ms from your test code, you have to send 8 frames at least (55 * 7 = 385 ms, 55 * 8 = 440 ms) to complete a stage update.

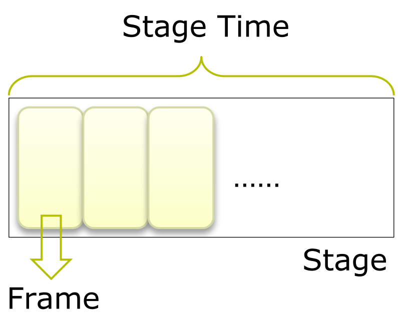


Figure 12 - Frames in a stage for eTC driver

You will easily see ghosting image, decaying quality or fading effect if sending fewer update frames in suggested stage time.

1.10.2. Set the stage time of iTC driver

iTC driver has defined the duration of a stage time which is programmed in the driver IC. It will continue to run the update frames according to its performance until sufficing the defined stage time. In most cases, the number of frames of iTC will be greater than that of eTC in a defined stage. Because of this, the optical performance of iTC driver is usually better than eTC driver in a fixed stage time.

1.10.3. Flowchart for fast update

The charged particles in the ink material are in unbalanced state while fast refreshing continuously. They need either compensation or inversed image to clean the ghosting effect, maintain the optical performance and extend the lifetime of EPD module. With regards to this, you have to take count of fast update times or the EPD has been stopped refreshing for a period of time without any changes (the diamond shape in red below) and place a standard normal update timely. Find the figure below is the suggested EPD driving flow chart of Global and fast update.

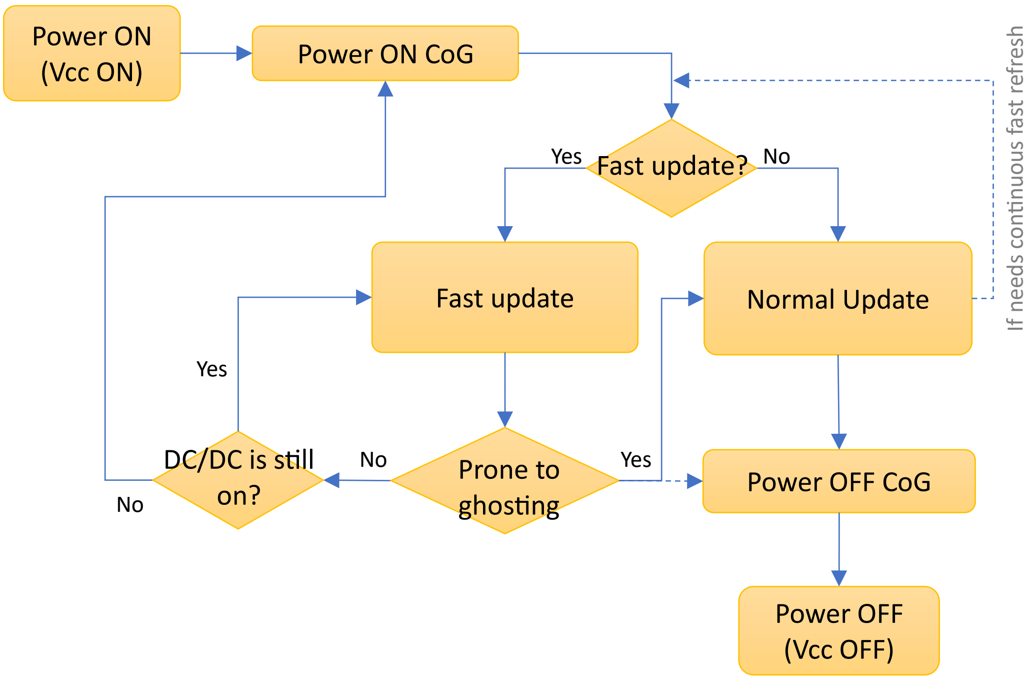


Figure 13 Driving flow chart of Global and fast update

Based on our experiments, we highly recommend running a normal update or power off the CoG to wait until next cycle is coming:

1. After 50 to 100 fast updates
2. 30 seconds without any screen changes (*e.g.* idle state)
3. Ghosting effect starts to appear on screen
4. Totally different template or scene

Point 2 depends on the use case. All the criteria also depend on your acceptance of image quality or tolerance of ghosting images.

iTC driver features shorter period of power on or off the CoG. If the duty cycle of your product is low, we would suggest you could consider power off the CoG, and then turn on the CoG per new fast update to reduce the standby current. The duration of power ON + OFF CoG of iTC driver is around 100ms or less mostly.

1.11. Graphic Library

To implement fast update, you will need to send images continuously but most of the time the pixels to be changed are usually in small areas or from some fixed objects. If there is no graphic generating tool or image production engine to handle these small changes, the output effect will be greatly reduced and will increase the system load.

There are already many graphic libraries on the market that are dedicated to LCD and touch panel. However, they need to support 1-bit grayscale and to provide a driver to manage a double frame-buffer for fast update.

Pervasive Displays has developed the **Pervasive Displays Library Suite** (or PDLS) specifically for the e-paper screens, extension boards and development kits.

PDLS provides a high-level interface to drive the e-paper screens with normal and fast update, supports capacitive touch and haptic feed-back. It is developed in C++ and can be easily ported to any SDK.

It includes the text and graphic primitives, including multilingual fonts.

Figure 14 – Multilingual support

It also provides advanced libraries such as graphics with clock, gauge and histogram; GUI with label, button, check-box, slider or keyboard; codes with bar- and QR-codes; files to save and retrieve pictures as files; and serial to generate and print header files on a serial console.

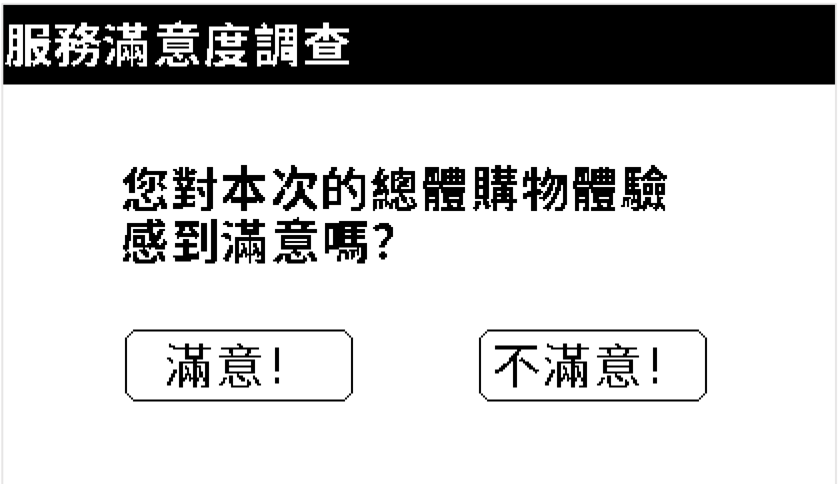


Figure 15 – A human-machine interface in Chinese

PDLS comes in four editions; open-source Basic edition with selected features; pre-compiled Evaluation edition with multilingual fonts, graphics, GUI and serial; Commercial edition with source code; and Viewer edition to simulate an e-paper screen and touch on a PC.

The Knowledge Base provides a user manual with commented examples; a reference manual with all the functions and their respective parameters; applications notes and reference designs.

Please contact Pervasive Displays for more information on PDLS.

Pros and cons of fast update

1.12. Pros of fast update

- Better user experience

In most of the cases, the flashing (toggling) among inversed, black and white images of normal update causes observer is unconformable requires keeping looking at the screen. The direct fast update on screen has eliminated this issue and provides better visual experience.

- Faster update time

In the case of normal update, typically, there are four image stages to complete an update cycle. Excluding the power-on and power-off stages, the refresh time of fast update is 1/4 of normal update.

- Less power to move particles in FPL

Since there is only a portion of pixels are moved (changed) from fast update, the power applied on FPL is less than that from normal update if we based on one refresh.

- Integrate diverse applications

Fast update features quick refresh and visual experience which helps integrate for example: smart home, IoT device, industrial appliance, retail's shelf talker, information signage and more. When integrating with touch panel on top, the applications can be extended to HMI, HVAC, portable device, badge and any device for user interaction. Thanks to EPD's bi-stable technology and low power consumption, the EPD will maintain the content without power consuming to fulfil the low duty cycle scenario and is also able to react shortly.

1.13. Cons of fast update

- Ghosting of previous image

Because the characteristic of FPL material (varies from batch by batch), it's difficult to make the optical quality of every EPD identical and the reflectance of either black or white may be different. In this case, user will see the output phenomenon like the ghosting effect, degraded image or decayed contrast ratio.

- More power budget for calculation

Even though less energy is needed to drive the particles in FPL for Fast update, you will need more calculations to count the coordination and image data from inputs, and it may need more power than ordinary case (*i.e.* normal update) to prepare the data for EPD. Note that the power is ON mostly and DC/DC is also up to take immediate actions for Fast update, and every pixel is refreshed or compared from colour image data or nothing data bytes. Hence, excluding the power periods, it is possible that requires more total power budget to finish a complete cycle of Fast update.

- Current leakage under direct sunlight

Because EPD uses TFT backplane with ITO electrodes, TFT would be occurred current leakage under sunlight illumination. TFT can't hold voltage and the data is getting down when current leakage happens. Once this effect is happened, user will easily see blurred images or decayed optical quality. normal update will always refresh every pixel of entire EPD with four stages per update cycle and power off the EPD every time to stop the leakage occurs. Nevertheless, the Fast update is just to refresh the pixels to be updated with one stage.

Due to this feature and there is current remains on the TFT (not always power off CoG/EPD), it's easy to see the current leakage and ghosting happened on the unchanged pixels/area of EPD under direct sunlight illumination when Fast update continuously. For this reason, Fast update is not suitable to perform under direct sunlight. It's better to select semi-outdoor or shielding environment to implement Fast update.

- Slow refresh in low temperature

Per EPD's characteristic, the lower ambient temperature, the slower refresh rate of screen changes. The particles will need more powerful driving, but the refresh time is still slower and slower due to the characteristics of EPD and lower temperature. It will affect the user's visual experience and reduce the convenience of operation.

Appendix

1.14. Glossary of acronyms

Acronym	Meaning
COG	Chip on Glass, Driver IC
EoL	End of life, product discontinued
EPD	Electrophoretic Display, e-Paper Display
eTC	External Timing controller
FPL	E ink material film, Front Panel Laminate
HMI	Human machine interface
HVAC	Heating, ventilation and air conditioning
iTC	Internal timing controller
ITO	Indium Tin Oxide
LUT	Look-Up Table
MCU	Microcontroller unit
PDI, PDi	Pervasive Displays Incorporated
PDLS	Pervasive Displays Library Suite
TFT	Thin-Film Transistor

1.15. Revision history

Version	Date	Page	Description
1	2013-10-18	All	Initial release
2	2014-12-01	6, 9, 11 5 10 12	Added FAQ Added Note Added Figure 2.2 Added suggested stage time description
3	2021-01-07	All	Rewrite the whole document Add iTC driver for fast update
2	2025-11-25	All	Updated to EPDK and EXT3.1 Added power section

Pervasive Displays Inc. 4F, No. 28, Chuangye Rd.,Tainan Science Park, Tainan City (74144) Taiwan (R.O.C.) T: (+886)(6) 2795399