Fast and partial update modes provide an improved user experience when using e-paper displays

Introduction:
E-paper displays offer many advantages over LCD and LED displays, in a wide range of applications. By implementing fast or partial updates, the number of applications grows further. However, fast update and partial update modes may not be suitable for every application, so it is important to understand how they are implemented and the potential consequences.

By Charming Su, Pervasive Displays

The benefits of e-paper ensure it stands out from other display technologies. Analyst, Market Research Future, predicts the market for e-paper will grow at a compound annual growth rate of over 37% through to 2023, at which point its value will be approaching $6 billion.[1]

The technology's appeal lies in its flexibility, both physically and in the applications it can address. It is low power, lightweight and thin, making it ideal in smart labelling applications. It can be removed from power for long periods of time yet remain readable, even in direct daylight. For applications that need these attributes, e-paper is unsurpassed.

There are now numerous examples of e-paper displays that operate without any battery or local power source at all. In these examples power, which is only needed to update the display's contents, is provided over an RF link, typically using either NFC or RFID. These contactless, battery-less tags perfectly demonstrate the main characteristics and benefits of e-paper technology.

The benefits of fast and partial update modes when using e-paper
As with any display technology, the faster e-paper can be updated, the better the user experience. However, what is also common across most electronic devices is, the faster they operate, the more energy they consume. To be more precise, the energy needed to operate is consumed in a shorter period of time; it isn't necessarily true that a slower update results in lower total power consumed, at least not for most display technologies.

However, for e-paper, this is the case. Because e-paper is bi-stable, power is only consumed when something on the display is changing. In its steady state, the display requires no power. This bi-stable operation is why manufacturers are now able to produce battery-less displays with energy harvesting technologies.

To change the image being displayed, the usual sequence involves first inverting the grey level of each pixel in the current image; effectively turning the image into its own "negative". All pixels are then cleared, resetting the display to white, before updating the display with pixel data for the new image. The driving waveforms are pre-programmed in the display's driver IC as lookup tables (LUTs). Only the image data and an update command are needed to complete the screen refresh.

This global-update sequence may seem complicated, and the entire cycle can take a long time to complete. Moreover, inverting and clearing the display makes for cumbersome transitions between images. This may not be a problem if the display is updated infrequently or while unseen by users; a smart-retail application such as a shelf-edge label could be remotely updated before the store opens, for example. However, in an application where the display must be updated "live", the visual effects can impair the user experience.

Developers can use fast and partial update modes to overcome these drawbacks. In each mode, only the pixels that need to change, to display the new image, are changed. Both the old and new images are compared to generate the required waveforms using dedicated partial-update and fast-update LUTs.

When a fast update command is sent with the image data, the fast update LUT compares the old and new images and generates new waveforms for pixels in areas that need to change in a full screen area.

The partial update differs from fast update in that the LUT uses the old and new images to generate waveforms for specific square area of the updated image.

These two modes enhance flexibility for developers to choose a method best suited to their application. A partial update sends less waveform data to the display and can be completed more quickly if the differences between the two images are confined to a small area, such as updating the price displayed by a smart label while the remainder of the label remains the same. If, however, multiple areas of the image contain changes, a fast update may be completed more quickly.

The ability to execute a fast or partial update increases the number of possible applications. For example, e-paper displays are often used with a touch-sensitive panel fitted to the front or electromechanical buttons. These may trigger a screen change when activated. Partial updates would be applicable when a user operates a switch on – or associated with – the screen, for example, showing that a status has been toggled. [2]

The consequences of fast updates for e-paper

The apparent advantages of implementing either fast or partial updates implies it should be a standard mode of operation. However, there are some disadvantages associated with both fast and partial updates to e-paper displays, due to the nature of the technology.

Compared to the standard global update, both modes result in less energy being applied to the display to move the e-ink particles. However, the additional energy consumed for processing, to compare images and calculate the new waveforms, must be considered. Also, when using these modes, the power is mostly on with the DC/DC converter running to take immediate actions. Hence the total energy consumed may be more than for a global update. In addition, e-paper technology is prone to ghosting. This is the name given for a condition where pixel data from a previous image is still slightly visible after an update. Ghosting is more likely to occur when either a fast or partial update is implemented, and it is partly due to the variable nature of the technology. A global update resets each pixel, which removes or reduces the ghosting effect, but because a partial update only changes the state of selected pixels, they are not fully reset. This can degrade the quality of the image or impact the contrast ratio.

There is a further consequence of not fully resetting each pixel following an update. Under direct sunlight, the transistors in the e-paper backplane can experience current leakage because carrying out fast or partial updates still needs power to be supplied to the screen. Over time, this leakage current will have the effect of changing the state of a pixel, which will degrade the quality of the image being displayed. Because a global update keeps refreshing the entire screen (using 4 stages, as explained later) then when power is removed from the screen, the leakage current build-up in the backplane transistors is always cleared. This helps maintain a clear image as there is no current leaked once power is off. For this reason, implementing partial updates in applications that are exposed to direct sunlight may not deliver the best user experience.

Similarly, in applications where the ambient operating temperature is low, the speed of operation of the e-paper can be slower. This can result in a combination of the effects outlined above, including higher operating power and ghosting.

Enabling e-paper displays to support fast and partial update modes

The layered structure of an e-paper display for industrial applications includes the front-panel laminate (FPL) film, a waveform driver IC and a charge pump (Figure 1). The FPL holds the pigment material, which are suspended charged particles arranged in a pixel format. The voltage potential across each pixel determines the orientation of the charged particles. In monochrome displays, the particles present as either black or white, depending on their orientation.

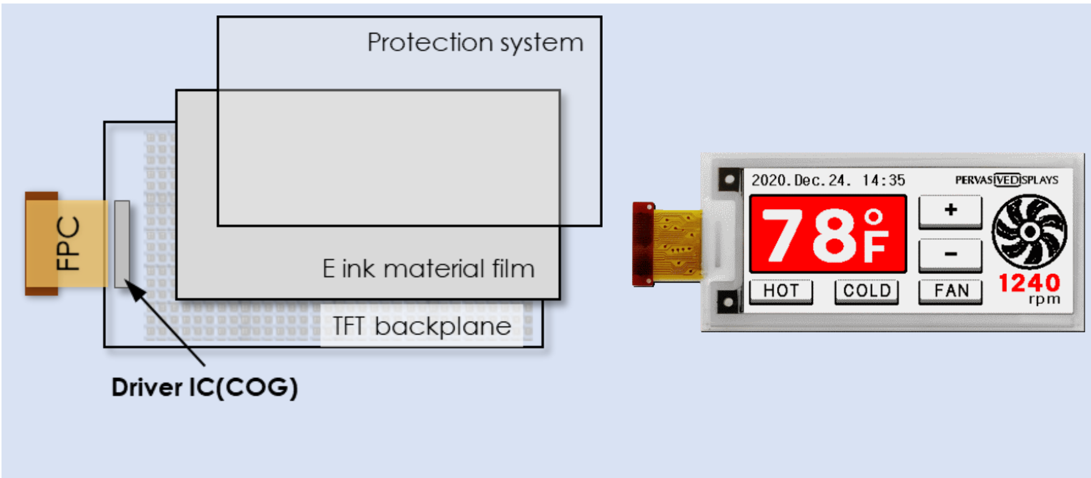
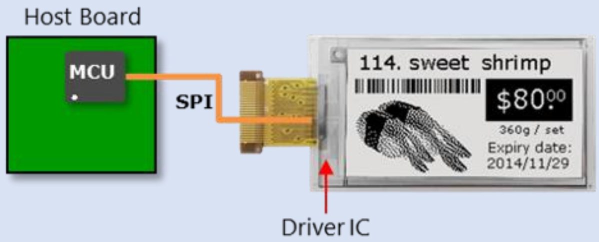
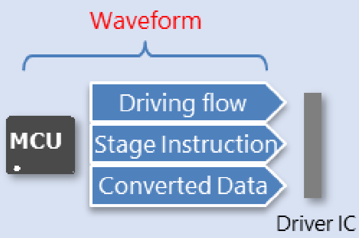
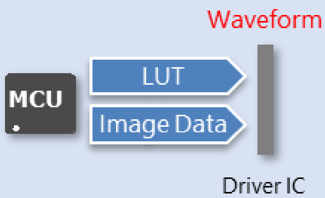


Figure 1 Layered structure of an EPD module

The orientation is controlled by the shape of the waveform applied across the pixel, generated by the driver IC, and the voltage magnitude of the waveform, supplied by the charge pump. The charge pump is effectively a DC-DC buck/boost converter.

As every e-paper display behaves slightly differently, particularly in relation to temperature variations, the waveforms applied by the driver IC are tuned. Together, the driver IC and charge pump manage the way the display is updated and, together, they are referred to as the timing controller. If the waveforms are pre-programmed into the driver IC and it is mounted inside the display (using chip on glass, or COG), along with the charge pump, the display is said to have an internal timing controller (iTC). If the charge pump circuit is arranged on PCBA (host board) out of the COG, it is referred to as an external timing controller(eTC). Often, in the case of an external timing controller, the waveforms will be implemented using a microcontroller.

<table>
<tr><td></td><td colspan="2">
</td></tr>
<tr><td></td><td>eTC (external Tcon)</td><td>iTC (internal Tcon)</td></tr>
<tr><td>Driver type</td><td>
</td><td>
</td></tr>
<tr><td>Driving waveform</td><td>Controlled by MCU</td><td>Embedded in driver IC (CoG)</td></tr>
<tr><td>Customization / Design flexibility</td><td>Higher</td><td>Lower</td></tr>
<tr><td>Design-in effort</td><td>Normal</td><td>Easier</td></tr>
<tr><td>Power consumption</td><td>Lower</td><td>Normal</td></tr>
<tr><td>Explanation</td><td>Developers can control the flow and each stage using the MCU.
Developers need to send converted image data at each specific stage.</td><td>The driving waveforms (LUTs, lookup tables) have been pre-programmed in the driver IC. Developers just need to send image data and update command to complete the screen refresh.
If developers need special support to change the waveforms, extra LUTs can be sent externally.</td></tr>
</table>

Implementing fast updates with e-paper displays

As explained above, the waveforms used to drive the display's pixels are fundamental to the user experience and overall performance of the display. Because of this, most e-display manufacturers will develop and implement their own driver waveforms.

Both global updates (also referred to as full update or global refresh) and fast updates (or local update or fast refresh) are supported by Pervasive Displays (Table 1). As outlined earlier, a global update will impact every pixel on the display, while a fast update only impacts the pixels that need to change, in order to display the new image. With a fast update the full image is sent to the driver IC, which carries out the image data comparison. A partial update requires an area (window) to be defined and the system side sends just the image data specific to that area, but uses the same driving waveform to carry out the update. Typically, eReaders use the partial update approach. The following description will focus on global update and fast update.

There are effectively two approaches taken by manufacturers in order to implement updates and it relates to the number of image buffers used. This system-level design choice has an impact on both how updates are implemented and the overall performance.

In systems that use two buffers, the current image is stored in one buffer and the new image is stored in the second buffer. They are then compared on a pixel-by-pixel basis. The result of this comparison is the new image to be displayed. In systems that use a single buffer, the buffer always holds the new image and in this case no comparison is carried out.

A global update using two image buffers is a 4-stage process (depending on the EPD design) and involves the following steps:

Stage 1: Inverting the current image
Stage 2: Displaying a completely white image
Stage 3: Inverting the new image
Stage 4: Displaying the new image

This process is used because it causes more of the pigment material to migrate in the pixel, which reduces the ghosting effect and also extends the lifetime of the FPL.

In systems with a single image buffer the global update process is reduced to just two stages:

Stage 1: The display is toggled between black and white
Stage 2: The new image is written to the screen

This process resets every pixel on the display and reduces the time taken to update the display by 50%.

Implementing partial updates only requires a system architecture with two image buffers, however an extra process is needed to determine which of the pixels remain unchanged and should maintain their current state. In order for the driver IC to implement this, it needs a step that compares the current and new images.

By comparison, a fast update is effectively a single-stage process, with the new image data being sent directly to the display. However, as outlined above, this doesn't allow for any pixel compensation or resetting. If the driver waveforms do not include the comparison process, implementing fast updates could result in ghosting and a reduction of the lifetime of the display, due to overdriving the particles that are always the same colour in a pixel. This will impact the display's performance or even cause permanent damage to the display.

Optimized fast and partial updates
Using two image buffers allows each pixel in the current and new images to be compared. The actions taken will depend on the image data, as outlined in Table 2, below.

Fast or Partial Update		Previous Image	
		Black	White
New Image	White	**White**	**No Changes**
	Black	**No Changes**	**Black**

Table 2: A summary of the actions taken during a partial update image comparison

Where the action is No Changes, it means the pixel colour data tells the waveform driver not to change the content. In order to make the fast update effective, the driver waveform applied to these pixels needs to remain consistent; no change is necessary. This means that every pixel has one of three possible actions; change to black, change to white, or do not change.

If an external timing controller configuration is being used, the microcontroller must first carry out the pixel comparison and then convert the actions into image data. This image data is then sent to the external timing controller (Figure 2). The green colours below indicate that No Changes data is sent to the driver IC.

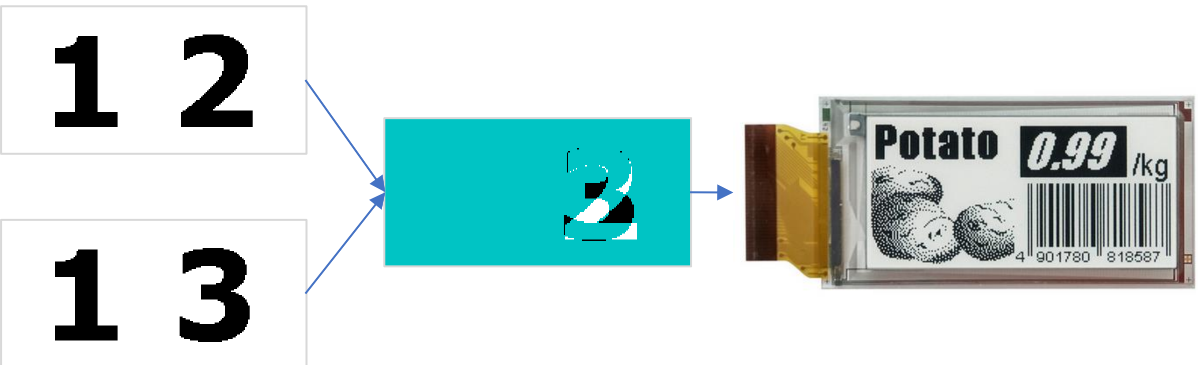


Figure 2: Image data sent to an e-paper display with an external timing controller

If a display with an internal timing controller is used, the information needed to drive the display is stored inside the driver IC's look-up tables (LUTs). This LUT data is pre-programmed by the display manufacturer and includes cycle times, drive options, parameters and more. In this case, the microcontroller simply needs to send the previous image first followed by the new image to the display and the internal timing controller handles the comparison process to carry out a fast update internally (Figure 3). There is no comparison process required by the microcontroller. Using a LUT for fast updates is the key to the iTC model.

Figure 3: Image data sent to an e-paper display with an internal timing controller, configured to implement fast updates.

Fast update performance using an external timing control system can be improved by executing the update multiple times. The number of fast refreshes that can be carried out depends on the cycle time required and the system time available. The time allocated is referred to as the stage time and it is related to the frame time of the display, which is the time it takes to update all of the pixels. For example, a dot-matrix display with a resolution of 200 by 96 dots, supplied by Pervasive Displays, has a frame of 200 x 96 and a recommended stage time of at least 400ms.

The frame time can be calculated, based on the time the microcontroller needs to refresh the entire screen once. In order to achieve a stage time of at least 400ms with a frame time of 55ms, the frame should be written at least eight times (55 x 8 = 440). If the stage time is lower than the recommended minimum there is more chance that the application will experience ghosting or a degradation in image quality.

When using a display with an internal timing controller (iTC), the stage time is defined by the manufacturer. In most cases, the number of frames that can be carried out in a stage will be higher than when using a display with an external timing controller.

In order to extend the display's lifetime, improve overall performance or reduce ghosting, it is recommended that a global update be carried out at a regular interval. This interval will either be based on the amount of time since the last global update, or the number of fast or partial updates that have been carried out since the last global update. Based on experimental data, Pervasive Displays recommends a global update is carried out after 15 fast updates, or 10 seconds of idle state for displays with an eTC driver. For e-Paper displays with an iTC driver, the recommended maximum can be increased to 50 fast updates or even more when using Pervasive Displays' special tuned LUT.

Another way of improving system-level performance is to use an image library for graphics that are likely to be subject to fast or partial updates. For example, if the application is an interactive user interface featuring some button controls, the image library could contain fast or partial updates for each possible state of the button. This also applies to common features, such as a progress bar, sliders or scroll bars.

Conclusion

When using e-paper displays, the use of fast or partial updates can greatly improve the user experience. It overcomes one of the common criticisms of e-paper, the time required to display updates.

In applications where only a small part of the display needs to change on a regular basis, partial updates can have a large and positive impact. However, it is important to understand the potential consequences of using partial updates and how best to implement them.

For further information, please contact your local Pervasive Displays representative.

The following reference videos from Pervasive Displays can be found on their YouTube channel Pervasive Displays Inc.

Partial update demonstrations
https://www.youtube.com/watch?v=enzUbiSWenQ&ab_channel=PervasiveDisplaysInc.

Thermostat application demonstration
https://www.youtube.com/watch?v=0jLfi62LT08&ab_channel=PervasiveDisplaysInc.

Smart home application
https://www.youtube.com/watch?v=Yh7yO8vpefU&ab_channel=PervasiveDisplaysInc.

Human machine interface (HMI) use case
https://www.youtube.com/watch?v=9fqRE6idp7U&ab_channel=PervasiveDisplaysInc.

Text references:

1) https://www.marketresearchfuture.com/reports/e-paper-display-market-2189
2) https://www.youtube.com/watch?v=0jLfi62LT08