## Application Note
for
small size Monochrome EPD
with
iTC (OTP LUT)

<table>
<thead>
<tr>
<th>Description</th>
<th>Interface for the small size Mono. EPD with the iTC(OTP LUT)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Date</strong></td>
<td>2018/10/22</td>
</tr>
<tr>
<td><strong>Doc. No.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Revision</strong></td>
<td>01</td>
</tr>
</tbody>
</table>

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

Tel: +886-6-279-5399       Fax: +886-6-270-5857
# Table of Contents

1. General Description ........................................................................................................... 3  
   1.1 Overview ..................................................................................................................... 3  
   1.2 Panel drawing .............................................................................................................. 4  
      1.54-inch EPD .................................................................................................................. 4  
      2.13-inch EPD ................................................................................................................. 5  
      2.66-inch EPD .................................................................................................................. 6  
      2.7-inch EPD ................................................................................................................... 7  
      2.87-inch EPD .................................................................................................................. 8  
      4.2-inch EPD ................................................................................................................... 9  
   1.3 EPD Driving Flow Chart .............................................................................................. 10  
   1.4 Overall Waveform ...................................................................................................... 11  
   1.5 SPI Timing Format ...................................................................................................... 12  

2. Power on COG driver ...................................................................................................... 14  

3. Set environment temperature and PSR .......................................................................... 15  

4. Input image to the EPD .................................................................................................. 16  

5. Send updating command .............................................................................................. 18  

5. Turn-off DC/DC ............................................................................................................ 19  

Revision History ............................................................................................................... 21  

Glossary of Acronyms ...................................................................................................... 22
1. General Description

1.1 Overview

The document introduces how to drive the small size EPD with OTP LUT. The “Small size” includes 1.54”, 2.13”, 2.66”, 2.7”, 2.87” and 4.2”. The EPD use single driver and that embedded T-con. The major control interface of the driver is SPI. The host sends both the setting commands and the display image to driver through the SPI bus.

![Diagram of SPI System](attachment:image.png)
1.2 Panel drawing

1.54-inch EPD

- Data direction from (1,1) to (152,1)
- Scan direction from (1,152) to (152,152)
- FPC
- Pin 1
2.13-inch EPD

[(1,1) Data direction (104,1)]

Scan direction

(1,212) → (104,212)

FPC

Pin 1
2.66-inch EPD

Scan direction

(1,1) Data direction → (152,1)

(1,296) → (152,296)

FPC

Pin 1
2.7-inch EPD

Data direction: (1,1) → (176,1)

Scan direction: (1,264) → (176,264)

FPC

Pin 1
2.87-inch EPD

(1,1) Data direction → (128,1)

Scan direction

(1,296) → (128,296)

FPC

Pin 1
4.2-inch EPD

Data direction

Scan direction

FPC

Pin 1

(1,1) → (400,1)

(1,300) → (400,300)
1.3 **EPD Driving Flow Chart**

The flowchart below provides an overview of the necessary actions to update the EPD. The steps below refer to the detailed descriptions in the respective sections.

```
Start

Get the temperature from temperature sensor

Power on Tcon (COG)

Input temperature to COG

Send image data to COG

Send the power settings of DC/DC and Turn-On DC/DC

Make sure BUSY = High

Send update command to COG

Wait until BUSY = High

Turn-Off DC/DC

End
```
1.4 **Overall Waveform**

The diagram below provides a signal control overview during an EPD update cycle.

![Overall Waveform Diagram]
1.5 SPI Timing Format

SPI commands are used to communicate between the MCU and the COG Driver. The SPI format used differs from the standard in that two-way communications are not used, and CS is pulled high then low between clocks. When setting up the SPI timing, PDI recommends verify both the SPI command format and SPI command timing in this section.

The maximum clock speed of the display is 10MHz.
- Below is a description of the SPI Format:
  
  \[ \text{SPI}(0xI, 0xD_0, 0xD_1, 0xD_2, \ldots) \]

  Where:
  
  - \( I \) is the Register Index and the length is 1 byte
  - \( D_{0\sim n} \) is the Register Data. The Data length is variable by different Register Index.

- SPI command signals and flowchart:

  ![SPI Timing Format Diagram]

  SPI command signals and flowchart:

  - SPI\((0xI_1,0xD_{0\sim D_n})\)
  - D/C# = 0
  - CS# = 0
  - Register Index \((0xI_1)\)
  - CS# = 1
  - D/C# = 1
  - CS# = 0
  - Send data \((0xD_{0\sim n})\)
  - CS# = 1
  - Data send Complete?

  For example:
  
  To send two SPI commands:
  
  SPI\((0x08,0x9D)\) and SPI\((0x09, 0xD0)\)

  If register data is more than one byte, the CS# pulse is necessary between each data byte.
SPI command timing

- **D/C#**
- **SCLK**
- **SDIN**
- **CS#**

### VCC = 2.3 to 3.6V, Temp = 0 to +50°C

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock Cycle Time</td>
<td>$t_{cycle}$</td>
<td>100</td>
<td>-</td>
<td>-</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>Chip Select Setup Time</td>
<td>$t_{CSS}$</td>
<td>60</td>
<td>-</td>
<td>-</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>Chip Select Hold Time</td>
<td>$t_{CSH}$</td>
<td>65</td>
<td>-</td>
<td>-</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>Write Data Setup Time</td>
<td>$t_{DSW}$</td>
<td>30</td>
<td>-</td>
<td>-</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>Write Data Hold Time</td>
<td>$t_{DHW}$</td>
<td>30</td>
<td>-</td>
<td>-</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>Clock Low Time</td>
<td>$t_{CLKL}$</td>
<td>35</td>
<td>-</td>
<td>-</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>Clock High Time</td>
<td>$t_{CLKH}$</td>
<td>35</td>
<td>-</td>
<td>-</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>Rise Time [20% ~ 80%]</td>
<td>$t_R$</td>
<td>-</td>
<td>5</td>
<td>-</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>Fall Time [20% ~ 80%]</td>
<td>$t_F$</td>
<td>-</td>
<td>5</td>
<td>-</td>
<td>ns</td>
<td></td>
</tr>
</tbody>
</table>
2. Power on COG driver

This flowchart describes power sequence for driver chip.

Start*1

Turn-on VCC / VDD

Delay 5ms

RES# = 1

Delay 10ms

Res# = 0

RES# = 1

Soft-reset SPI(0x00, 0x0E)

Delay 5ms

End

Note:

1. Start

Initial State:
VCC/VDD, RES#, CS#, SDIN, SCLK = 0
3. Set environment temperature and PSR

![Diagram of setting process]

**NOTE:**

1. **start**: Follow the end of the power on sequence
2. **TSSET**: is the temperature value and unit is degree of Celsius.
   - The highest bit of the data represents positive/negative in temperature.
   - if it’s positive, the data = (temperature value)
   - if it’s negative, the data = (2’s complement of temperature value)
   - example:
     - temperature value      data
     - 25°C                   0x19
     - 5°C                    0x05
     - -5°C                   0xFB
3. **PSR**: there are 2 bytes’ data to send.
   - 4.2” : 0x0F,0x89
   - Other Size : 0xCF,0x8D
4. **Input image to the EPD**

This section describes how to send the image data into the COG driver which will update the display. EDP need to receive both First and Second frame data each updating. The index of the first frame is **0x10** and the second frame is **0x13**.

![Diagram showing image data input sequence and EPD size](image)

**Note 1:** \( n = (N \times M)/8 \)

The data of image frame, one bit represents 1 pixel. (e.g. the first byte represents the 1\(^{st}\) ~ 8\(^{th}\) pixels of the first line, the second byte represents the 9\(^{th}\) ~ 16\(^{th}\) pixels of the first line, ... and so on).

|-----------|------|------|------|------|------|------|------|------|

Image data input sequence:
- Line 1: (1,1) > (2,1) > ... > (N,1) >
- Line 2: (1,2) > (2,2) > ... > (N,2) >
- ...
- Line M: ... 

= (N \times M) bits
= \((N \times M)/8\) Bytes

<table>
<thead>
<tr>
<th>EPD size</th>
<th>N</th>
<th>M</th>
<th>Total bytes/frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.54&quot;</td>
<td>152</td>
<td>152</td>
<td>2,888</td>
</tr>
<tr>
<td>2.13&quot;</td>
<td>104</td>
<td>212</td>
<td>2,756</td>
</tr>
<tr>
<td>2.66&quot;</td>
<td>152</td>
<td>296</td>
<td>5,624</td>
</tr>
<tr>
<td>2.7&quot;</td>
<td>176</td>
<td>264</td>
<td>5,808</td>
</tr>
<tr>
<td>2.87&quot;</td>
<td>128</td>
<td>296</td>
<td>4,736</td>
</tr>
<tr>
<td>4.2&quot;</td>
<td>400</td>
<td>300</td>
<td>15,000</td>
</tr>
</tbody>
</table>
• First Frame
The frame is the "black" frame. The data "1" represents the black color pixel and the data "0" represents the white color pixel.

<table>
<thead>
<tr>
<th>Data</th>
<th>Pixel Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Black</td>
</tr>
<tr>
<td>0</td>
<td>White</td>
</tr>
</tbody>
</table>

• Second Frame
The second frame need to be fed K bytes 0x00. (K: reference Total bytes/frame on previous page.)
5. Send updating command

Note:
1. Start
   Follow the end of the input image sequence
2. This register does not have data, just need send the index.
5. Turn-off DC/DC

Start

Turn-off DC/DC
SPI(0x02)\(^2\)

BUSY = High ?
Yes
Set RES# to floating

Clear CS#, SDIN and SCLK to low level

Cut-off the Vcc/Vdd of COG

Set the BUSY pin to output 0\(^3\)

Delay 150ms

Set RES# back to output 0 from floating

End\(^4\)

Note:
1. Start
   Follow the end of the send updating command
2. This register does not have data, just need send the index.
3. BSUY should originally be input pin but it needs to be as output pin and output 0v now.
4. Finished the all of the steps for update the EPD
## Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Page (New)</th>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>2018/7/11</td>
<td></td>
<td>First issue</td>
<td></td>
</tr>
</tbody>
</table>

This document is the exclusive property of PDI and shall not be reproduced or copied or transformed to any other format without prior permission of PDI. (PDI Confidential) 本資料為龍亭新技專有之財產，非經許可，不得複製、翻印或轉變成其他形式使用。
### Glossary of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPD</td>
<td>Electrophoretic Display (e-Paper Display)</td>
</tr>
<tr>
<td>EPD Panel</td>
<td>EPD</td>
</tr>
<tr>
<td>TCon</td>
<td>Timing Controller</td>
</tr>
<tr>
<td>FPL</td>
<td>Front Plane Laminate (e-Paper Film)</td>
</tr>
<tr>
<td>SPI</td>
<td>Serial Peripheral Interface</td>
</tr>
<tr>
<td>COG</td>
<td>Chip on Glass</td>
</tr>
<tr>
<td>PDI, PDi</td>
<td>Pervasive Displays Incorporated</td>
</tr>
</tbody>
</table>