

Application Note

for

4.2" Spectra

with

Internal Tcon Driver IC(iTC)

(OTP LUT)

| | |
|--------------------|---|
| Description | Interface for the 4.2" Spectra EPD |
| Date | 2016/9/27 |
| Doc. No. | |
| Revision | 01 |

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

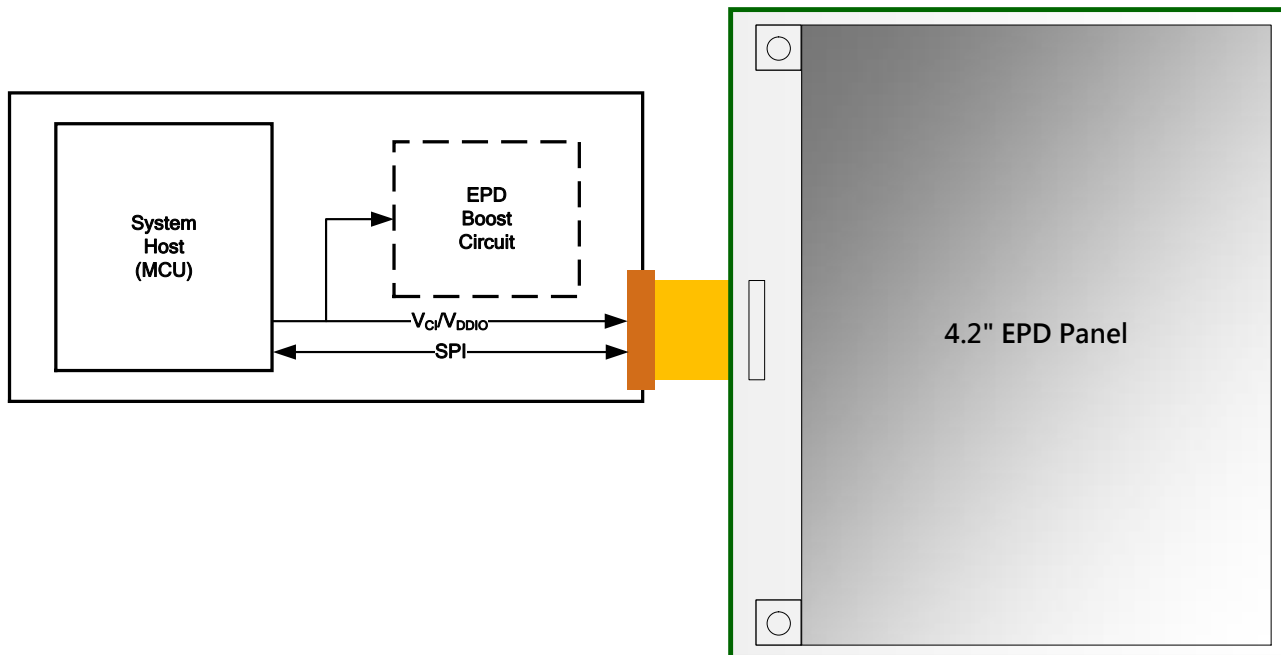
Table of Contents

| | | |
|-----|------------------------------|----|
| 1 | General Description | 3 |
| 1.1 | Overview | 3 |
| 1.2 | Panel drawing | 4 |
| 1.3 | EPD Driving Flow Chart | 5 |
| 1.4 | Overall Waveform | 6 |
| 1.5 | SPI Timing Format | 7 |
| 2 | Power on COG driver | 10 |
| 3 | Initialize COG Driver | 11 |
| 3.1 | Initial flow chart | 11 |
| 4 | Input image to the EPD | 12 |
| 5 | Send updating command..... | 14 |
| 6 | Turn-off DC/DC | 15 |
| | Revision History..... | 17 |
| | Glossary of Acronyms..... | 18 |

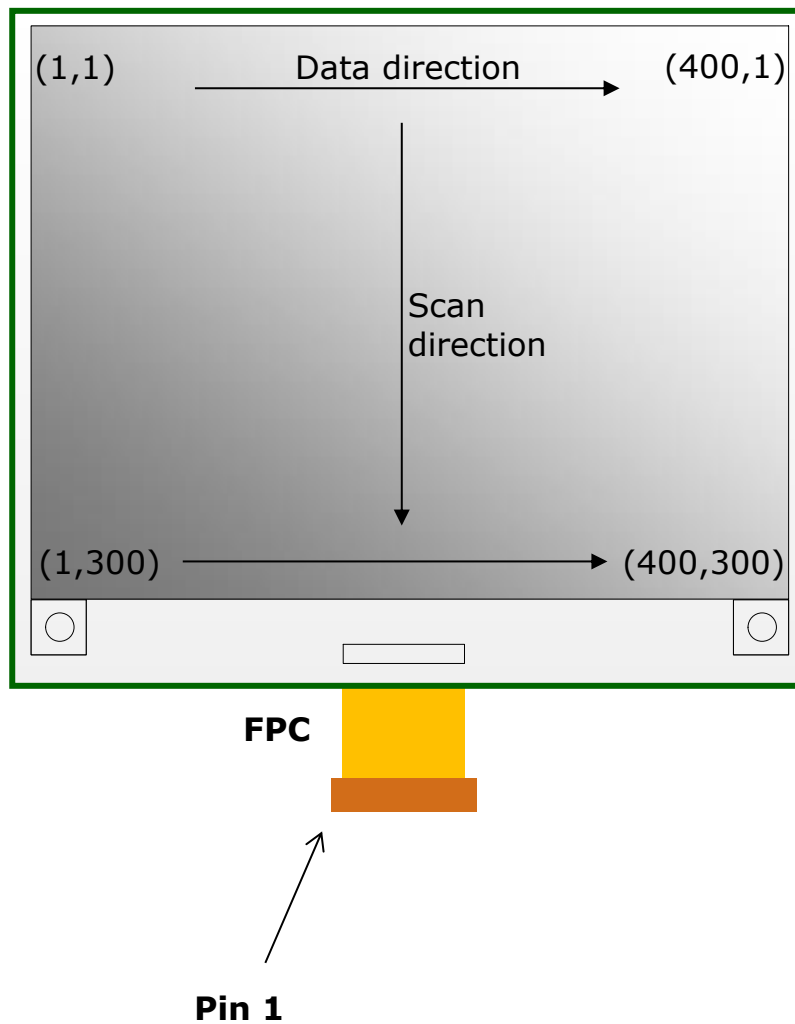
1 General Description

1.1 Overview

The document introduces how to drive the 4.2" EPD **with OTP LUT**. 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.

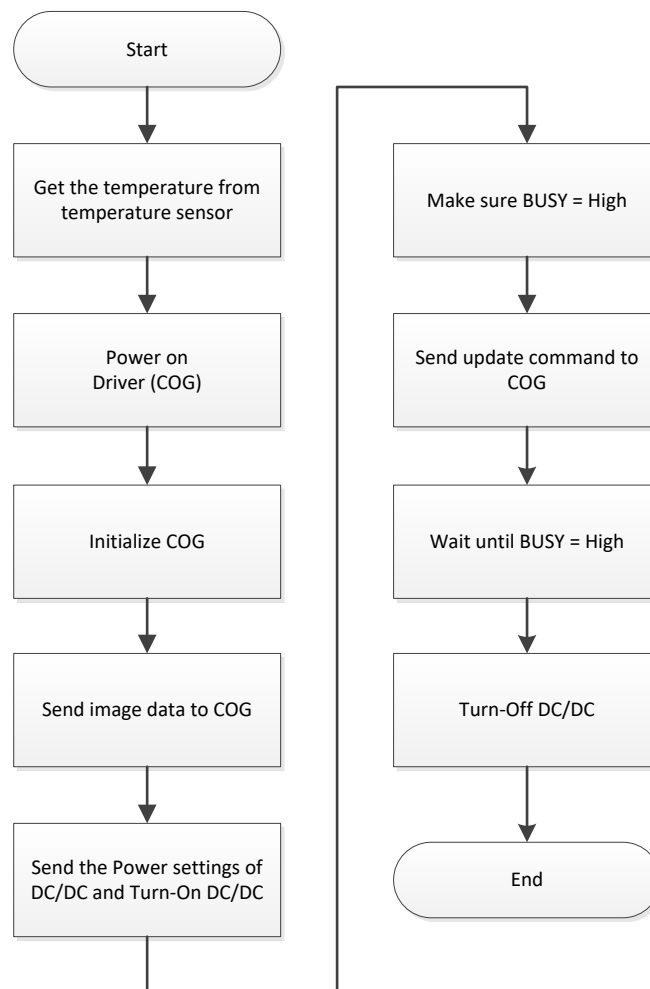


1.2 Panel drawing



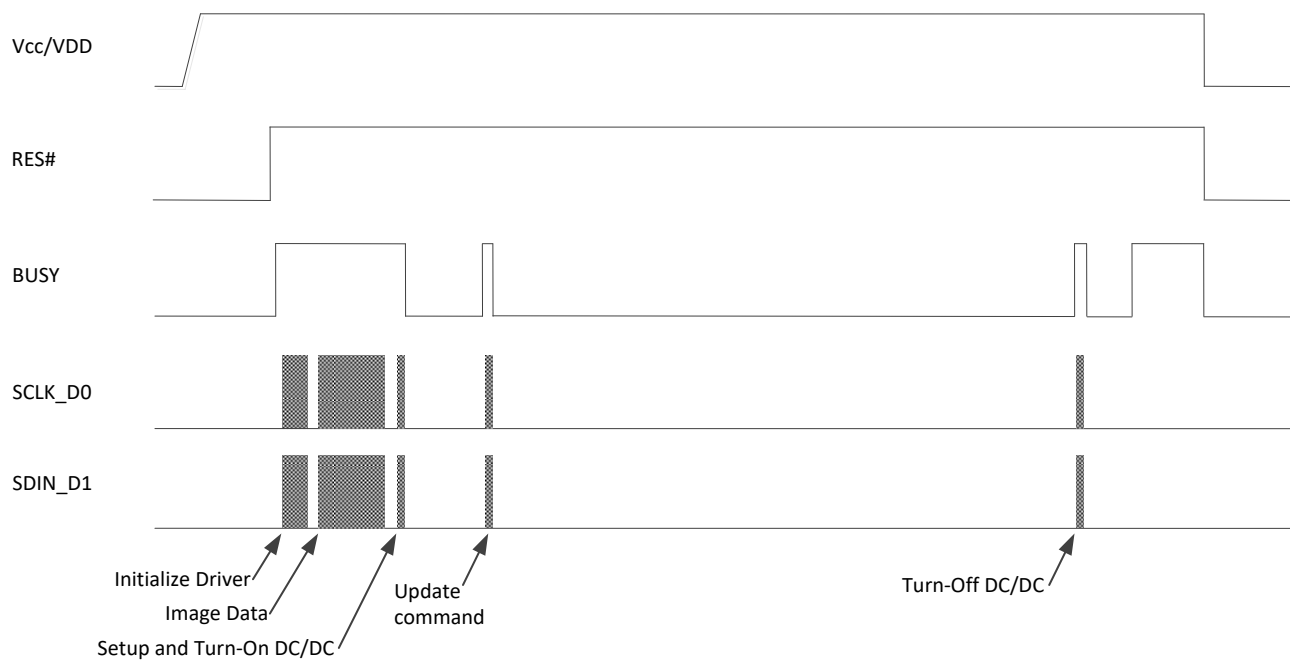
1.3 EPD Driving Flow Chart

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.



1.4 Overall Waveform

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



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:

SPI(0xI, 0xD₁, 0xD₂, 0xD₃, ...)

Where:

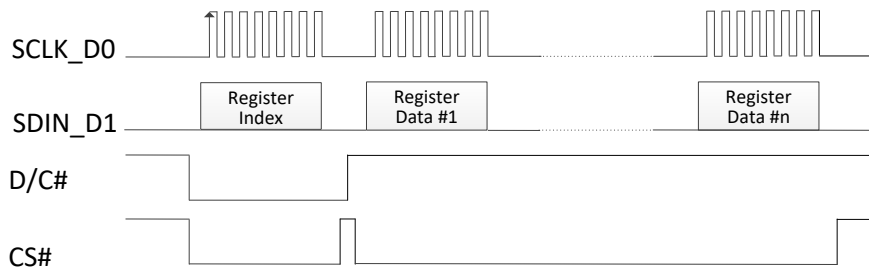
I is the Register Index and the length is 1 byte

D_{1~n} is the Register Data. The Register Data length varies from 0 to 15000 bytes depending on which Register Index is selected.

| Register Index | Number Bytes of Register Data |
|----------------|-------------------------------|
| 0x00 | 1 |
| 0x02 | 0 |
| 0x04 | 0 |
| 0x06 | 3 |
| 0x10 | 15000 |
| 0x12 | 0 |
| 0x13 | 15000 |
| 0x50 | 1 |
| 0x61 | 4 |
| 0xe0 | 1 |
| 0xe3 | 1 |
| 0xe5 | 1 |

- When SPI sends the Index, the D/C# have to pull LOW. When sends the data, the D/C# have to pull HIGH. The next page is the detail flow chart.

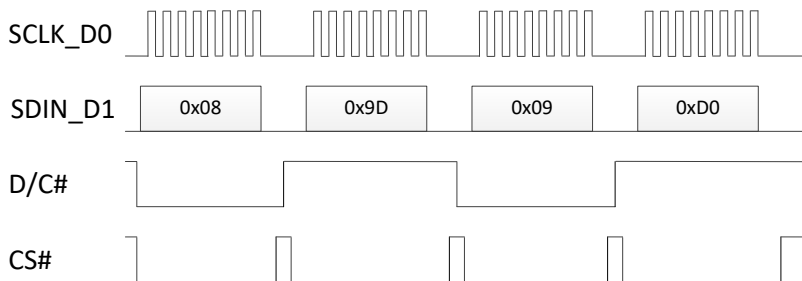
- SPI command signals and flowchart:



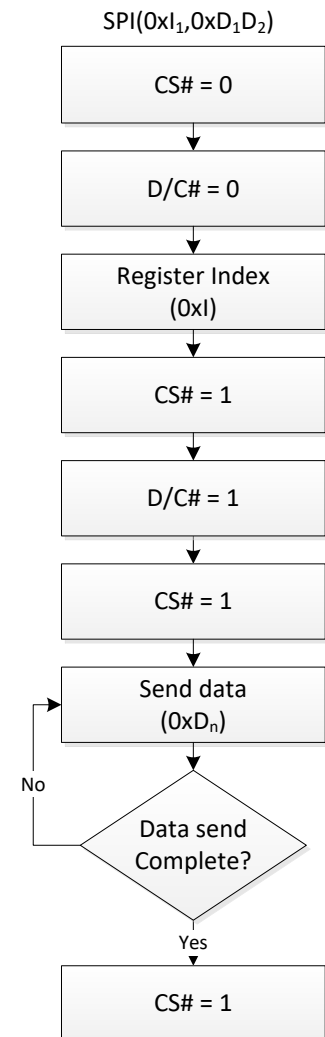
For example:

To send two SPI commands:

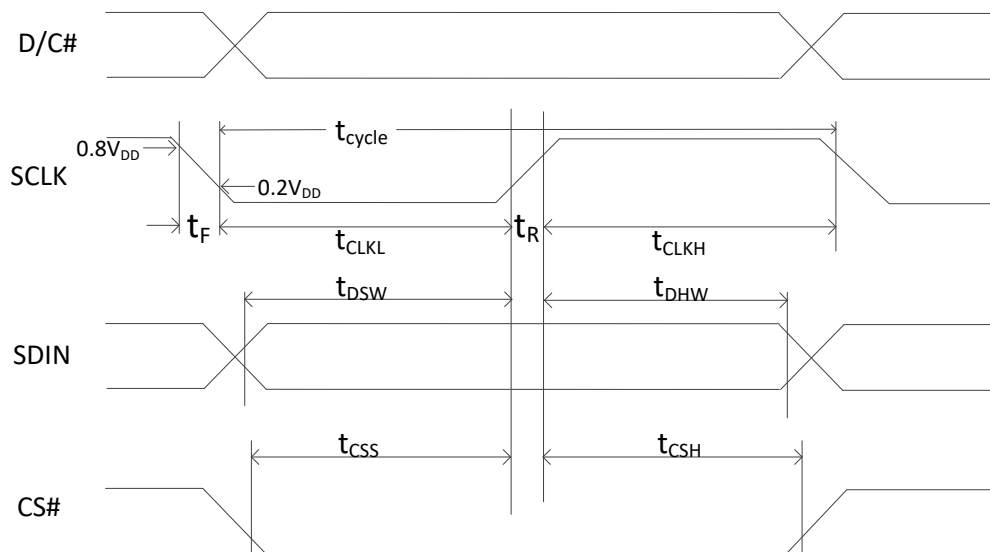
SPI((0x08,0x9D) and SPI(0x09, 0xD0)



If register data is larger than two bytes, you must input data continuously without setting Register Index again.



- SPI command timing



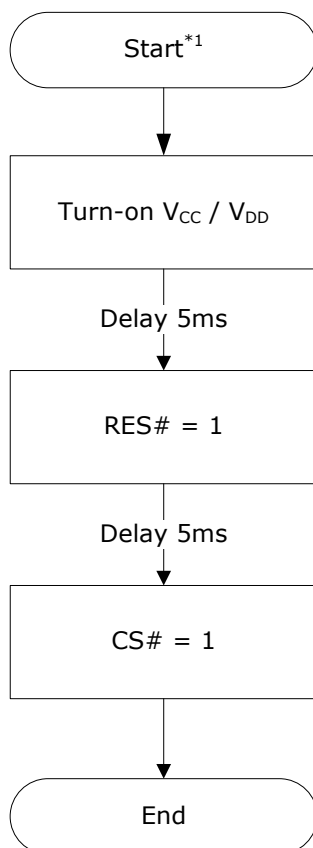
VCC = 2.3 to 3.6V

Temp = 0 to +50°C

| Item | Symbol | Min. | Typ. | Max. | Unit | Remark |
|------------------------|-------------|------|------|------|------|--------|
| Clock Cycle Time | t_{cycle} | 100 | - | - | ns | |
| Chip Select Setup Time | t_{CSS} | 60 | - | - | ns | |
| Chip Select Hold Time | t_{CSH} | 65 | - | - | ns | |
| Write Data Setup Time | t_{DSW} | 30 | - | - | ns | |
| Write Data Hold Time | t_{DHW} | 30 | - | - | ns | |
| Clock Low Time | t_{CLKL} | 35 | - | - | ns | |
| Clock High Time | t_{CLKH} | 35 | - | - | ns | |
| Rise Time [20% ~ 80%] | t_R | - | 5 | - | ns | |
| Fall Time [20% ~ 80%] | t_F | - | 5 | - | ns | |

2 Power on COG driver

This flowchart describes power sequence for driver chip.



Note:

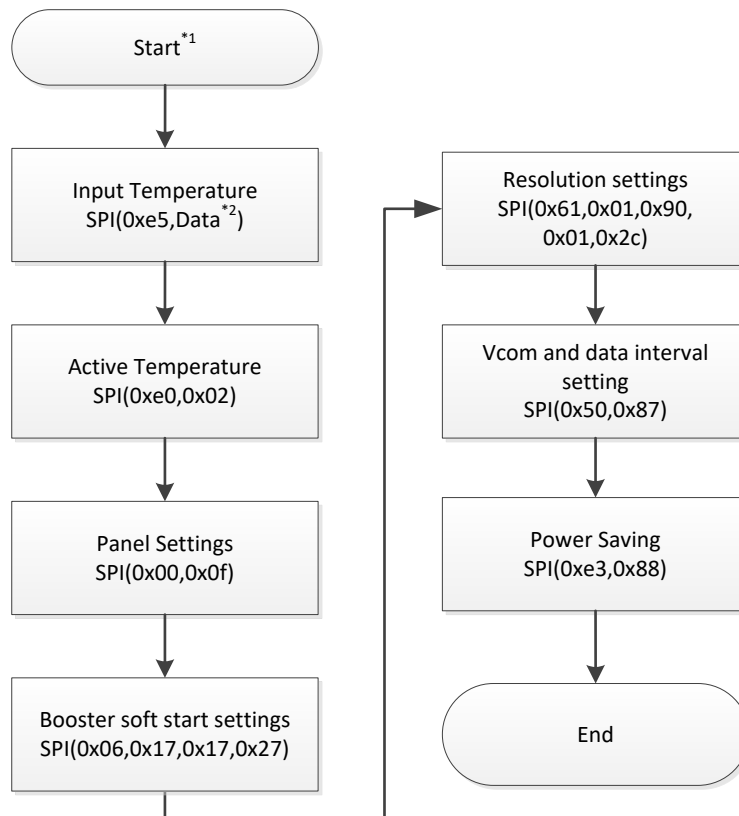
1. Start

Initial State:

VCC/VDD, RES#, CS#, SDIN, SCLK = 0

3 Initialize COG Driver

3.1 Initial flow chart



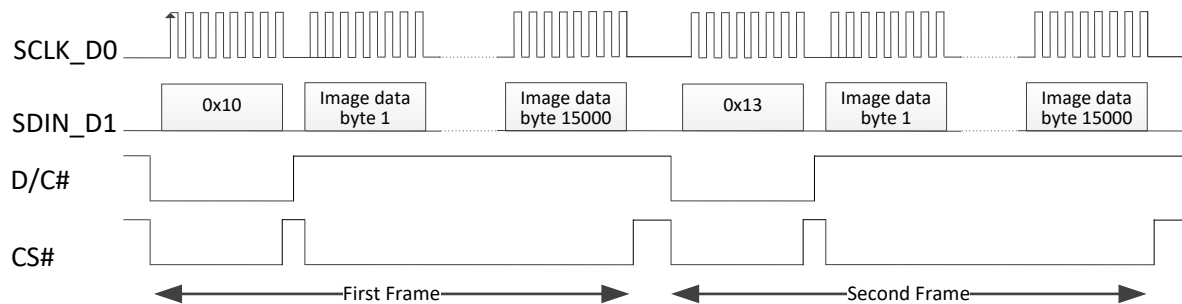
Note:

1. Start
Follow the end of the power on sequence
2. The data 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 complementn of temperature value)
example:

| temperature value | data |
|-------------------|------|
| 25°C | 0x19 |
| 5°C | 0x05 |
| -5°C | 0xFB |

4 Input image to the EPD

This section describes how many data should be sent to 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**.



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

| Data Byte | D[7] | D[6] | D[5] | D[4] | D[3] | D[2] | D[1] | D[0] |
|-----------|------|--------|--------|--------|--------|--------|--------|--------|
| Pixel | P[n] | P[n+1] | P[n+2] | P[n+3] | P[n+4] | P[n+5] | P[n+6] | P[n+7] |

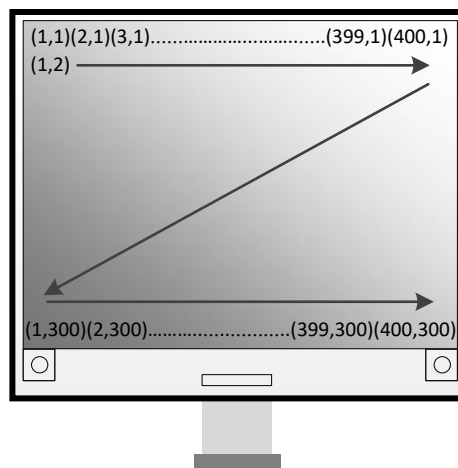


Image data input sequence :
 Line1:(1,1)>(2,1)>...>(400,1)>
 Line2:(1,2)>(2,2)>...>(400,2)>
 ...
 Line300:>(400,300)

Total : 1 x 400 x 300
 = 120,000 bits
 = 15,000 Bytes

- First Frame

The frame is the "black" frame. The data "1" represents the black color pixel and the data "0" represents both white and red color pixel.

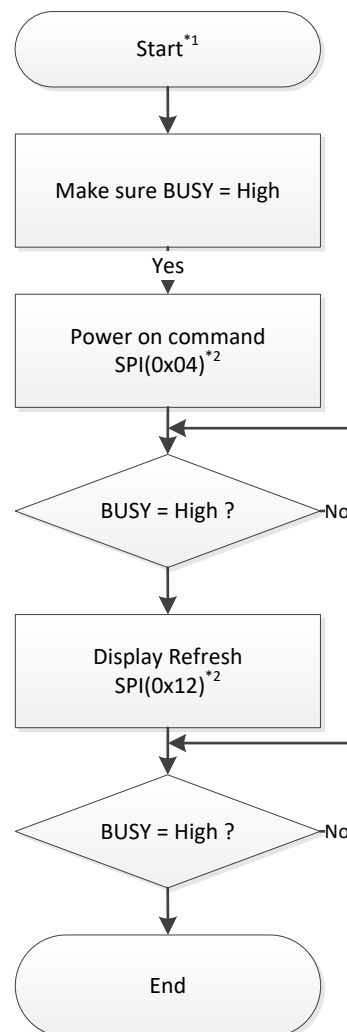
| Data | Pixel Color |
|------|-------------|
| 1 | Black |
| 0 | White/Red |

- Second Frame

The frame is the "Red" frame. The data "1" represents the red color pixel and the data "0" represents both black and white color pixel.

| Data | Pixel Color |
|------|-------------|
| 1 | Red |
| 0 | White/Black |

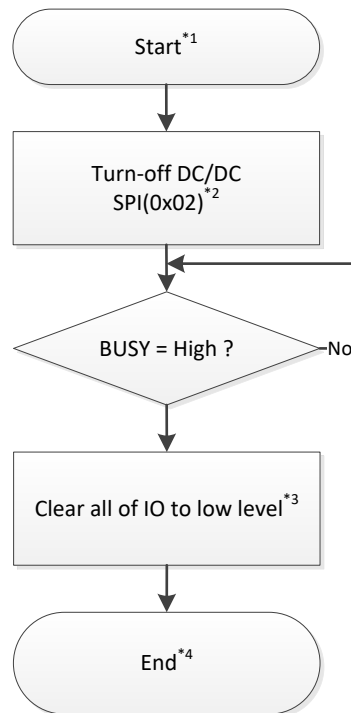
5 Send updating command



Note:

1. Start
Follow the end of the input image sequence
2. This register has no data, just need to send the index.

6 Turn-off DC/DC



Note:

1. Start
Follow the end of the send updating command
2. This register has no data, just need to send the index.
3. VCC/VDD, RES#, CS#, SDIN and SCLK
4. Finished the all of the steps for update the 4.2" EPD

Copyright

Pervasive Displays Incorporated All rights reserved.

This document is the exclusive property of Pervasive Displays Inc. (PDI) and shall not be reproduced or copied or transformed to any other format without prior permission of PDI. (PDI Confidential)

本資料為龍亭新技股份有限公司專有之財產，非經許可，不得複製、翻印或轉變成其他形式使用。

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

<http://www.pervasivedisplays.com>

Revision History

| Version | Date | Page (New) | Section | Description |
|---------|-----------|---------------|---------|-------------|
| 01 | 2016/1/27 | | | First issue |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

Glossary of Acronyms

| | |
|-----------|---|
| EPD | Electrophoretic Display (e-Paper Display) |
| EPD Panel | EPD |
| TCon | Timing Controller |
| FPL | Front Plane Laminate (e-Paper Film) |
| SPI | Serial Peripheral Interface |
| COG | Chip on Glass |
| PDI, PDi | Pervasive Displays Incorporated |