E-paper Display COG Driver Interface Timing for Wide Temperature of 1.44”,2” and 2.7” EPD with G2 COG and Aurora Ma Film

Description	**Detailed information to design a timing controller for wide temperature 1.44”,2” and 2.7” EPD with G2 COG and Aurora Ma film**
Date	**2015/08/14**
Doc. No.	**4P015-00**
Revision	**03**

	Design Engineering		
	Approval	**Check**	**Design**
	黃 2015.08.14 朝玄	黃 2015.08.14 朝玄	丁 2015.08.14 昭文

4F, No. 28, Chuangye Rd., Tainan Science Park, Tainan City 74144, Taiwan (R.O.C.)
Tel: +886-6-279-5399 Fax: +886-6-505-5300

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
http://www.pervasivedisplays.com

Table of Contents

Revision History....4
Glossary of Acronyms....5
1 General Description....6
1.1 Overview....6
1.2 Input Terminal Pin Assignment....7
1.3 Reference Circuit....9
1.3.1 1.44 inch EPD Reference Circuit....9
1.3.2 2 inch EPD Panel Reference Circuit....10
1.3.3 2.7 inch EPD Panel Reference Circuit....11
1.4 EPD Driving Flow Chart....12
1.5 Controller....13
1.6 SPI Timing Format....14
2 Write to the Memory....19
3 Power On G2 COG Driver....20
4 Initialize G2 COG Driver....21
5 Write Data from the Memory to the EPD....23
5.1 Data Structure....23
5.2 Overall Update Flow....24
5.2.1 Room Temperature Flow (0℃≦T≦50℃)....24
5.2.2 Low Temperature Flow (-20℃≦T＜0℃)....24
5.3 Store a Line of Data in the Buffer....25
5.4 Writing to the Display in Stages....29
5.4.1 Frame Type....29
5.4.2 Block Type....31
5.5 Driving Waveform and Temperature Factor in Stages....38
5.5.1 Room Temperature (0℃≦T≦50℃)....38
5.5.2 Low Temperature (-25℃≦T＜0℃)....39
5.5.3 The Structure of the stage....41
6 Power off G2 COG Driver....47
Appendix: Use G1's PCBA to drive the EPD with G2 Driver IC....48

Revision History

Version	Date	Page (New)	Section	Description
Ver. 01	2015/02/16	All	All	First issued
Ver. 02	2015/08/05	All	All	Modify PDi company address
		All	All	Revise wrong descriptions
		7	1.2	Modify Pin.8 assignment
		15-17	1.6	Register Index -> Command Index Register Data -> Command Data Modify the delay time between /CS high and low
		21	4	Modify delay time of power on sequence Modify OE setting (0x02,0x04) -> (0x02,0x06)
		24	5.2	Renew update flow
		26-28	5.3	Remove checking BUSY when sending each data byte
		31	5.4.2	Renew description for Block Type
		38	5.5.1	Modify Room Temperature setting
		45	5.5.3	Revise Sequence flow from Black -> White to White -> Black
		47	6	Modify delay time of VGL voltage
		48	7	Section 1.3.6 move to Appendix
Ver. 03	2015/08/14	38	5.5.1	Modify Room Temperature setting

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, Driver IC. There are two models: Gen1 (G1) is EOL and Gen2 (G2) which is used in this document
PDI, PDi	Pervasive Displays Incorporated

1 General Description

1.1 Overview

This document explains the interface to the G2 COG Driver to operate the EPD for a Timing Controller based solution using one page of memory buffer. G2 is the most recent EPD driving technology from PDI that offers new features such as breakage detection, lower inrush current, and a lower operation voltage.

The procedure to update display is

1. Store new pattern in memory buffer
2. Power on G2 COG Driver
3. Initialize G2 COG Driver
4. Update display stage by stage
5. Power off G2 COG Driver

Refer to the EPD controller in section 1.5 for the complete procedure. To operate the EPDs for the best sharpness and performance, each update of the panel is divided into a series of stages before the display of the new image pattern is completed. During each stage, frame updates with intermediate image patterns are repeated for a specified period of time. The number of repeated frame during each stage is dependent on the Timing Controller speed. After the final stage, the new pattern is displayed.

Section 1 is an overview and contains supporting information such as the overall theory for updating an EPD, SPI timing for PDI's EPDs, as well as current profiles.

Section 2 describes a method to write to memory buffer. New pattern is stored in the memory buffer and update image in displays.

Section 3 describes how to power on the G2 COG Driver which consists of applying a voltage and generating the required signals for /CS and /RESET.

Section 4 describes the steps to initialize the G2 COG Driver.

Section 5 describes the details on how to update the EPD from the memory buffer, create a line of data, update in stages.

Section 6 describes how to power off the G2 COG Driver, and discharge voltage from EPD to ground, make sure there is no voltage remains in EPD.

1.2 Input Terminal Pin Assignment

No	Signal	I/O	Connected to	Function
1	/CS	I	TCon	Chip Select. Low enable
2	BUSY	O	TCon	When BUSY = High, EPD stays in busy state that EPD ignores any input data from SPI.
3	ID	I	Ground	Set SPI interface
4	SCLK	I	TCon	Clock for SPI
5	SI	I	TCon	Serial input from host Timing Controller to EPD
6	SO	O	TCon	Serial output from EPD to host Timing Controller
7	/RESET	I	TCon	Reset signal. Low enable
8	BORDER_DRIVER or PWRON	-	BORDER or NC	For 1.44" & 2", connect to BORDER. For 2.7", keep open.
9	V_{CL}	C	Capacitor	-
10	C42P	-	NC	Not connected. These two pins are used only with G1 COG Drive IC.
11	C42M	-		
12	C41P	C	Charge-Pump Capacitor	-
13	C41M	C		-
14	C31M	C	Charge-Pump Capacitor	-
15	C31P	C		-
16	C21M	C	Charge-Pump Capacitor	-
17	C21P	C		-
18	C16M	C	Charge-Pump Capacitor	-
19	C16P	C		-
20	C15M	C	Charge-Pump Capacitor	-
21	C15P	C		-

No	Signal	I/O	Connected to	Function
22	C14M	C	Charge-Pump Capacitor	-
23	C14P	C		-
24	C13M	C	Charge-Pump Capacitor	-
25	C13P	C		-
26	C12M	C	Charge-Pump Capacitor	-
27	C12P	C		-
28	C11M	C	Charge-Pump Capacitor	-
29	C11P	C		-
30	V_{COM_DRIVER}	RC	Resistor & Capacitor	The duty cycle of V_{COM_DRIVER} can adjust V_{COM} voltage from source driver IC
31	V_{CC}	P	V_{CC}	Power supply for analog part of source driver
32	V_{DD}	P	V_{DD}	Power supply for digital part of source driver
33	V_{SS}	P	Ground	-
34	V_{GH}	C	Capacitor	-
35	V_{GL}	C	Capacitor	-
36	V_{DH}	C	Capacitor	-
37	V_{DL}	C	Capacitor	-
38	BORDER	I	-	For 1.44" & 2", connect to BORDER_DRIVER For 2.7", connect to V_{DL} via control circuit for white frame border
39	V_{ST}	P	V_{COM_PANEL}	-
40	V_{COM_PANEL}	C	Capacitor	V_{COM} to panel

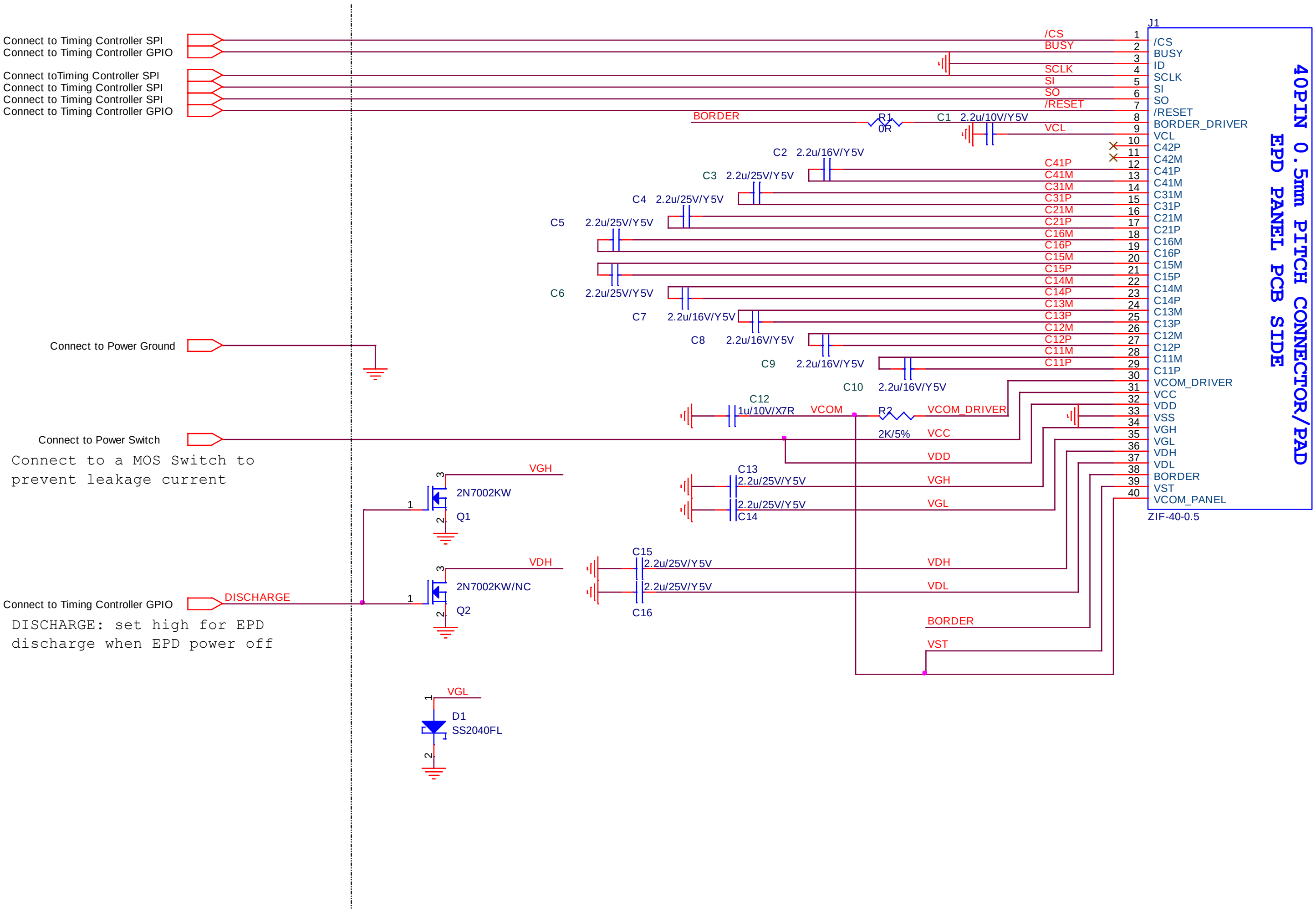
Note:

1. **I**: Input, **O**: Output, **C**: Capacitor, **RC**: Resistor and Capacitor, **P**: Power, **NC**: Not connected (Do not SMT)
2. Recommend to use an independent SPI bus to control the EPD.
3. Around the active area of the EPD is a 0.5mm width blank area called the BORDER. It's connected to V_{DL} (-13V ~ -14V) to keep the border white. After EPD updates with the constant voltage, the border color may degrade to a gray level that is not as white as the active area. Reset the Border per screen update to avoid this phenomenon.

1.3 Reference Circuit

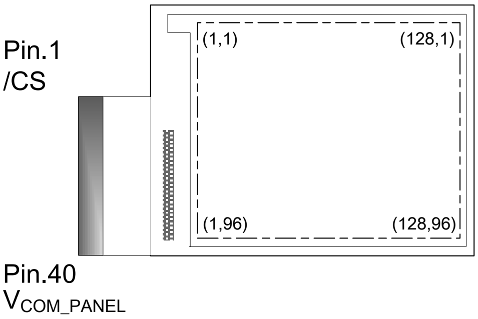
1.3.1 1.44 inch EPD Reference Circuit

INPUT



Note :

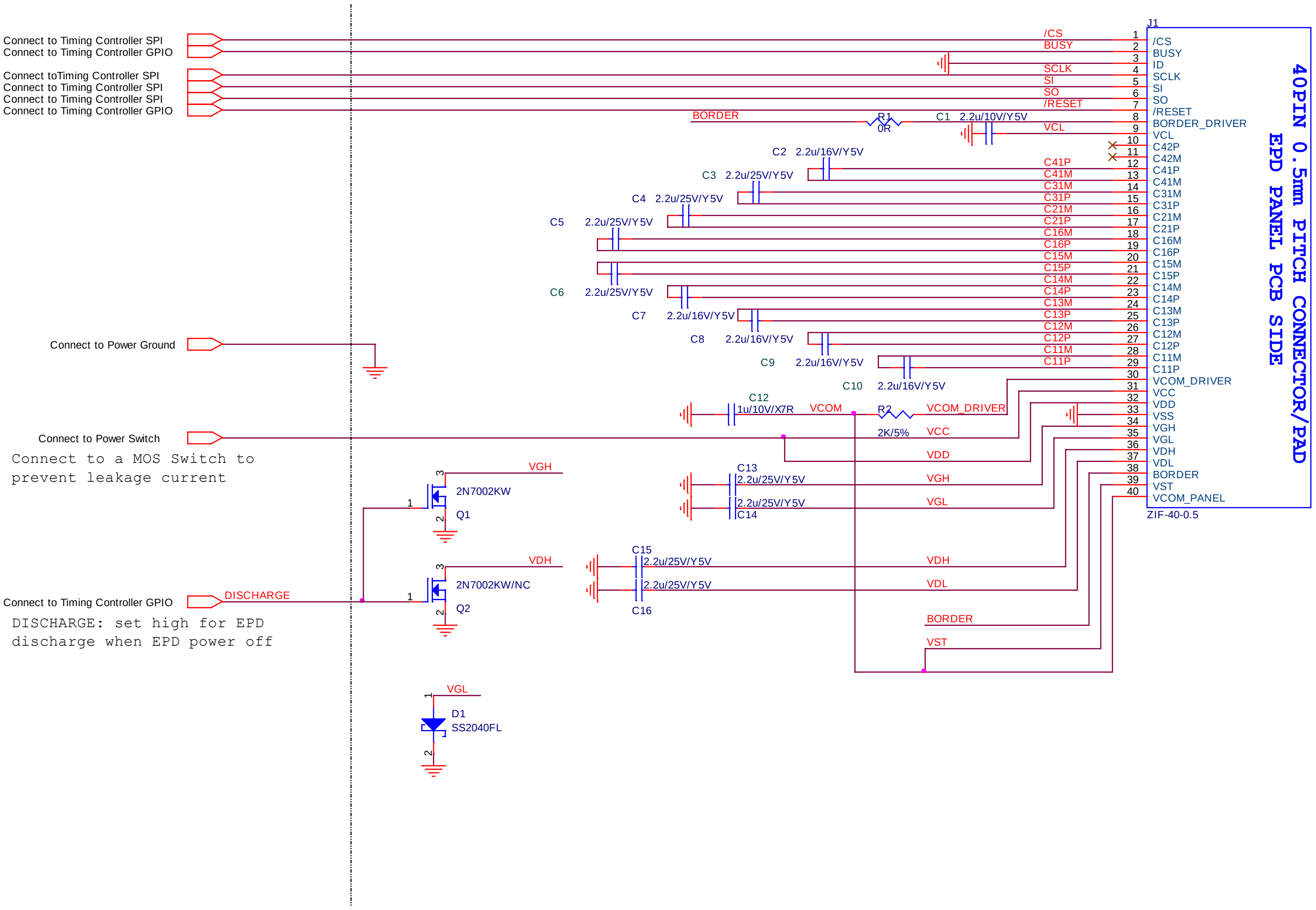
1. V_{DD} and V_{CC} must be discharged promptly after power off
2. 1.44" Pin.1 location



3. R1 is connected.

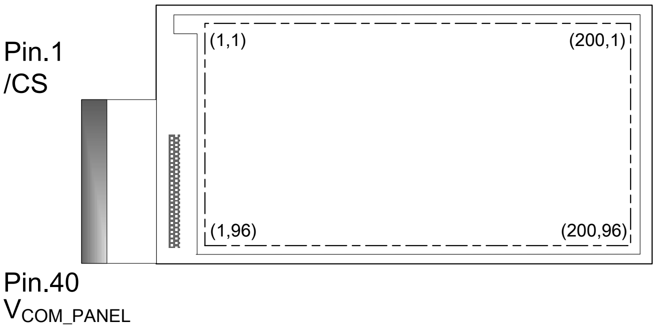
1.3.2 2 inch EPD Panel Reference Circuit

INPUT



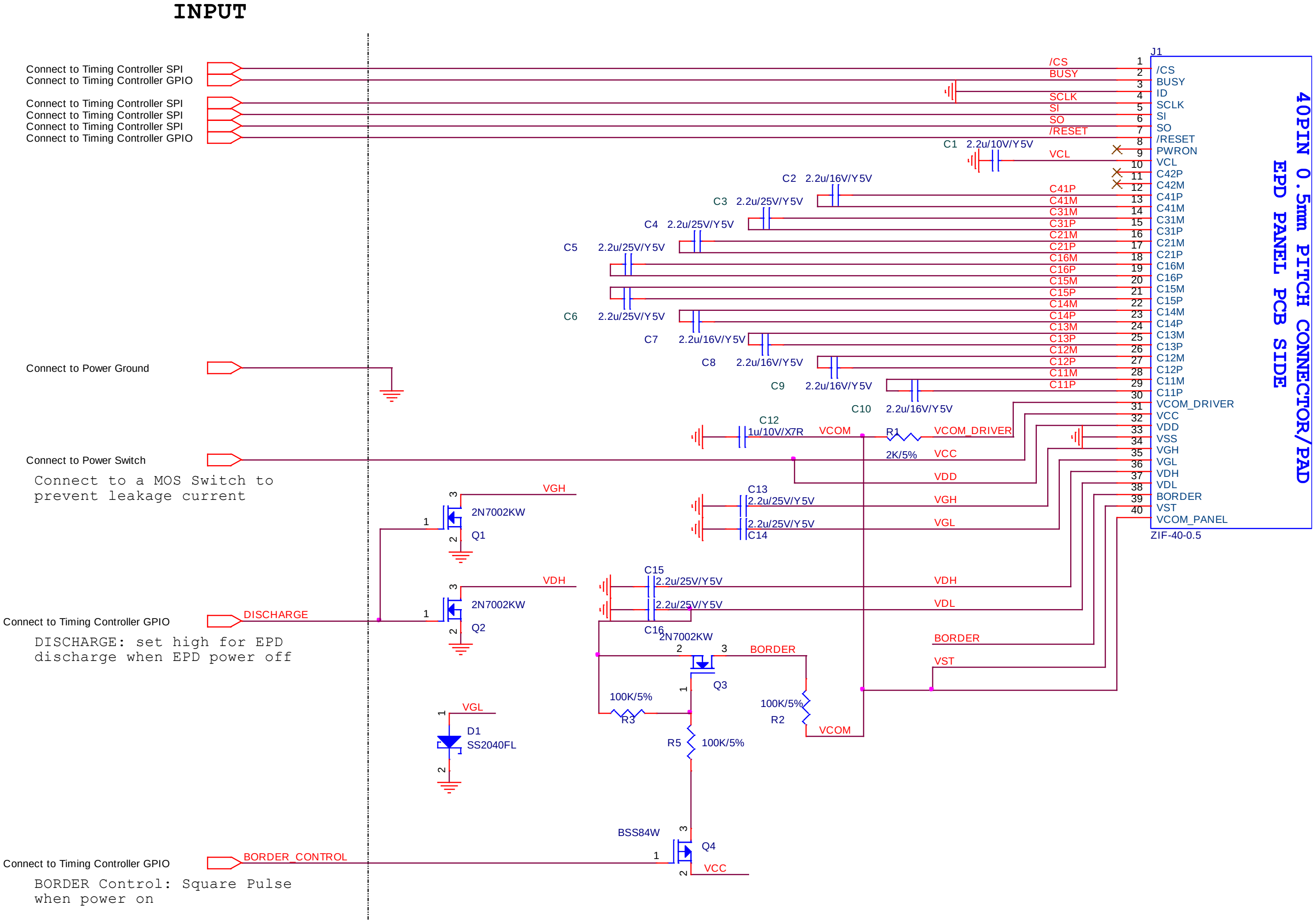
Note :

1. V_{DD} and V_{CC} must be discharged promptly after power off
2. 2" Pin.1 location



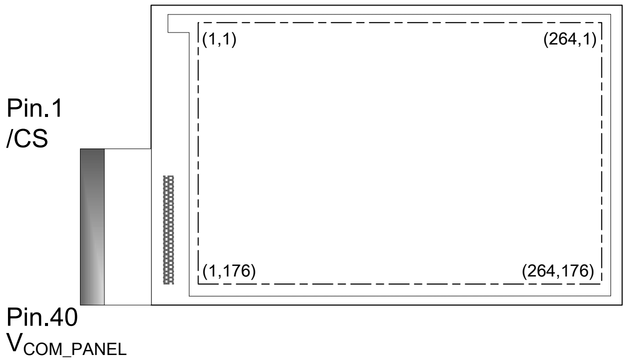
3. R1 is connected.

1.3.3 2.7 inch EPD Panel Reference Circuit



Note :

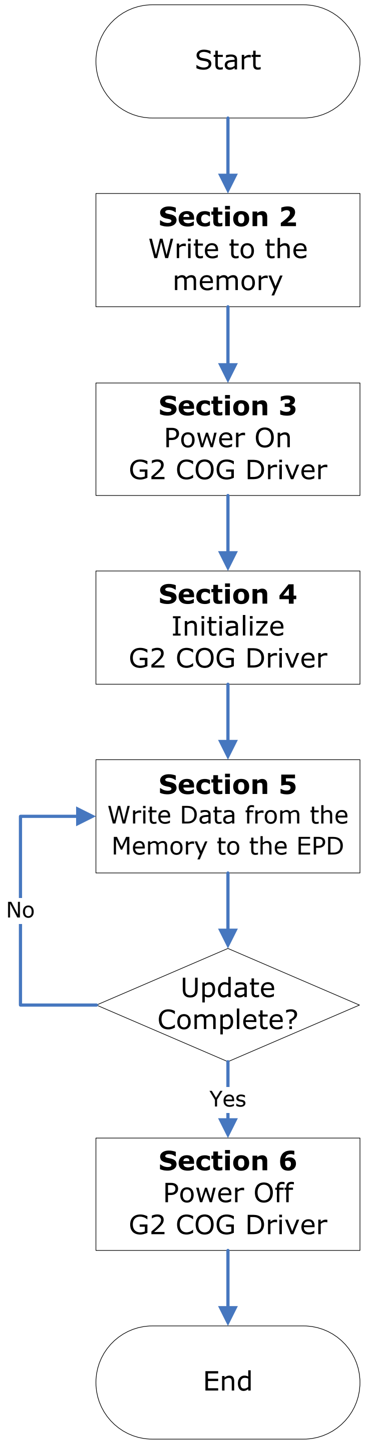
1. V_{DD} and V_{CC} must be discharged promptly after power off
2. 2.7" Pin.1 location



➢ If you are still using our old COG driver G1 and previous circuit, please refer to the Appendix.

1.4 EPD Driving Flow Chart

The flowchart below provides an overview of the actions necessary to update the EPD. We call this is Global Update or Full Update. The steps below refer to the detailed descriptions in the respective sections.



1.5 Controller

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

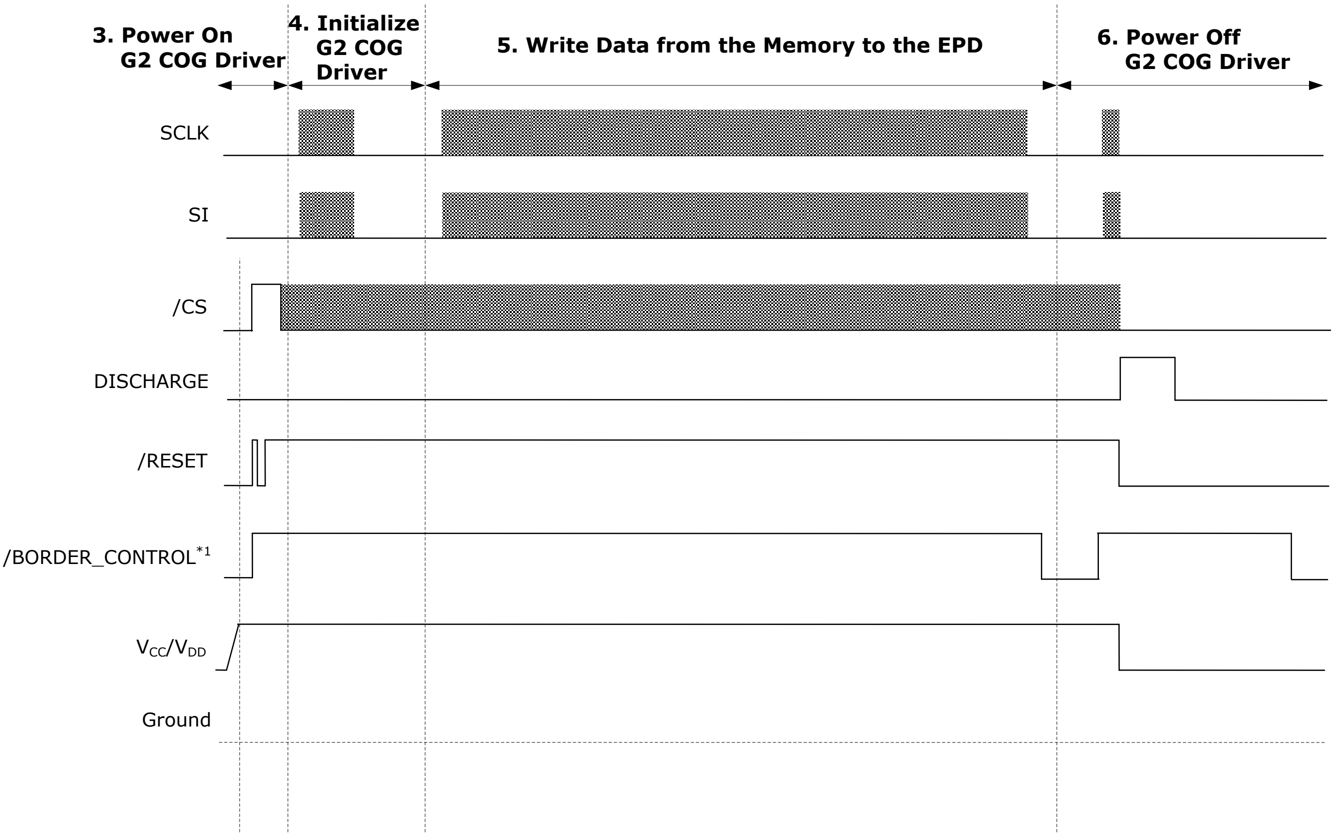
"3. Power On G2 COG Driver",

"4. Initialize G2 COG Driver",

"5. Write data from the Memory to the EPD",

"6. Power Off G2 COG Driver",

The number and title matches a section title in this document.


3. Power On G2 COG Driver
4. Initialize G2 COG Driver
5. Write Data from the Memory to the EPD
6. Power Off G2 COG Driver
SCLK
SI
/CS
DISCHARGE
/RESET
/BORDER_CONTROL*1
VCC/VDD
Ground

Note:

1. /BORDER_CONTROL:

   /BORDER_CONTROL is used to keep a sharp border while taking care of the electronic ink particles. For implement this function, developer needs to use a Timing Controller pin (GPIO) to control this signal.

   (This function is only used for 2.7" EPD Panel)

1.6 SPI Timing Format

SPI commands are used to communicate between the Timing Controller and the G2 COG Driver. The SPI format differs from the standard in that two way communications, and /CS is pulled high then low between Command Index and Command Data. When setting up the SPI timing, please follow the SPI command format and timing in this section and verify the control signals in Section 1.5 then.

The maximum SPI clock speed for G2 COG Driver is 20MHz.

The SPI mode is 0.

- Below is a description of the SPI Format:

SPI (0xI, 0xD$_1$, 0xD$_2$, 0xD$_3$ …)

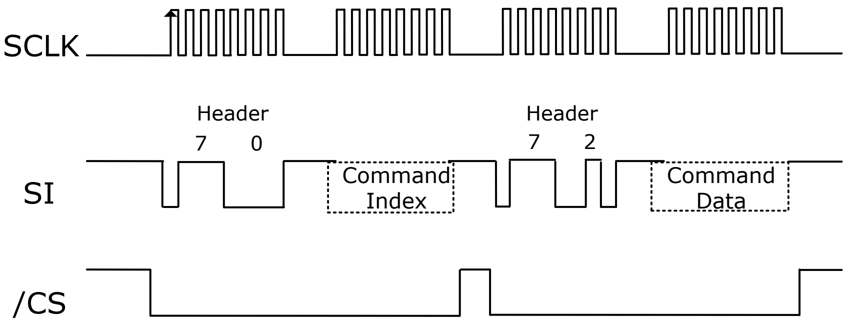
Where:

I is the Command Index and the length is 1 byte
D$_{1\sim n}$ is the Command Data. The Command Data length is from 1 or 110 bytes depending on which Command Index is selected.

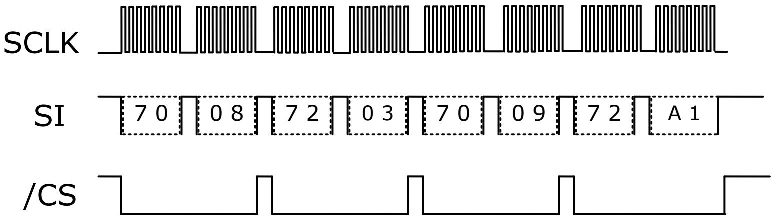
Command Index	Number Bytes of Command Data
0x01	8
0x02	1
0x03	1
0x04	1
0x05	1
0x07	1
0x08	1
0x09	1
0x0A	< 110
0x0B	1
0x0F	1

- Before sending the Command Index, the SPI (SI) must send a 0x**70**(header of Command Index).
- Likewise, the SPI (SI) must send a 0x**72**(header of Command Data) prior to the Command Data. The flow chart and detailed description can be found in the next page.
- Number Bytes of Command Index (0x0A) depends on which panel size is used.

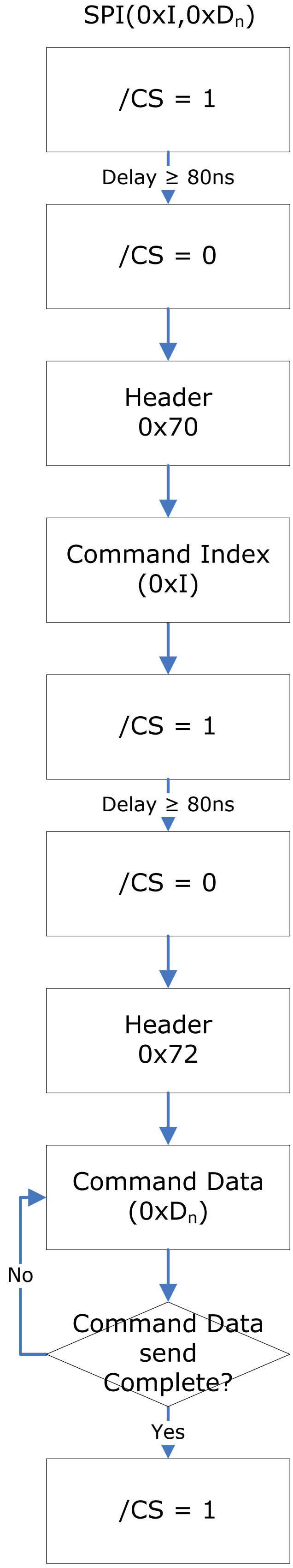
- SPI write command signals and flowchart(SPI):



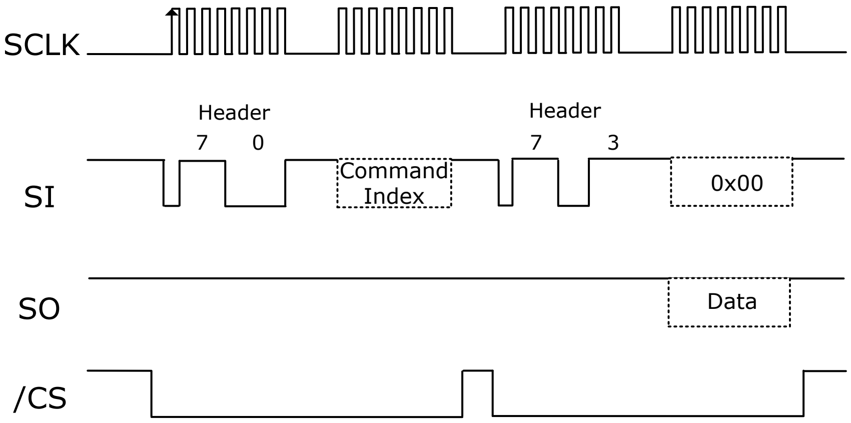
Example:
Send two continue SPI write commands
SPI(0x08,0x03) and SPI(0x09, 0xA1)



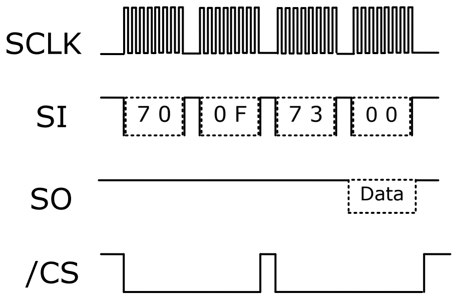
If Command Data is larger than two bytes, you must input data continuously without setting Command Index again.



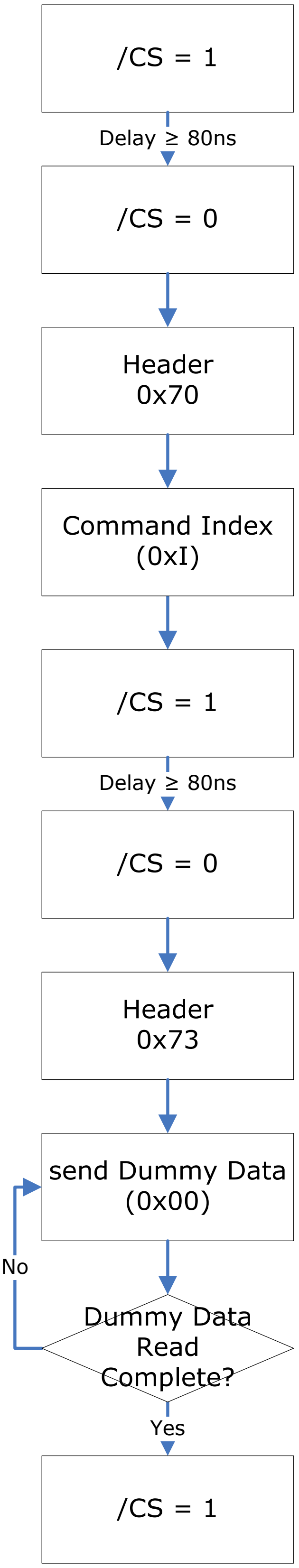
- SPI read command signals and flowchart(SPI_R):



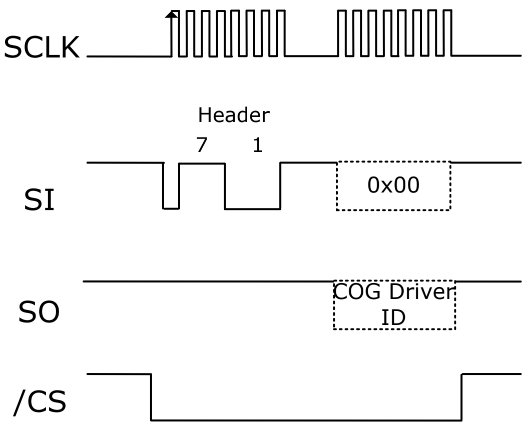
Example:
Send one SPI read command
SPI_R(0x0F,0x00)

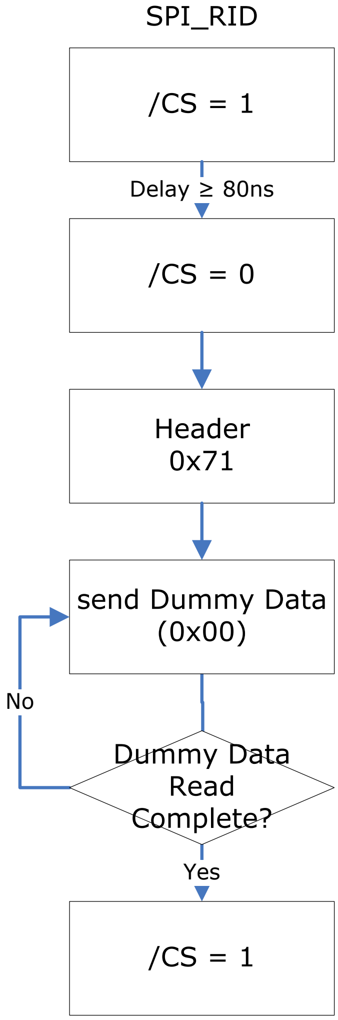


SPI_R(0xI,0xD$_n$)

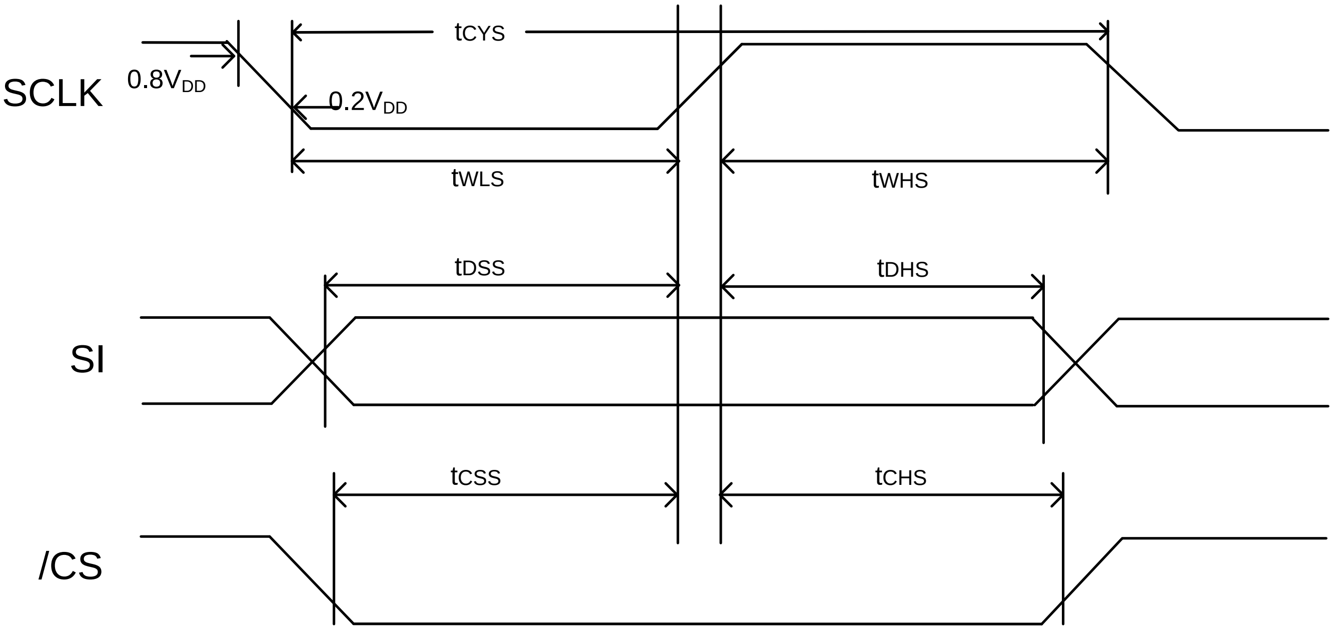


- SPI read COG ID and flowchart(SPI_RID):





- SPI command timing (mode 0)



VCC = 2.3 to 3.6V **Temp = -25 to 50℃**

Item	Signal	Symbol	Min.	Typ.	Max.	Unit	Remark
Serial clock cycle	SCLK	tCYS	50	-	-	ns	
SCLK high pulse width	SCLK	tWHS	25	-	-	ns	
SCLK low pulse width	SCLK	tWLS	25	-	-	ns	
Data setup time	SI	tDSS	12	-	-	ns	
Data hold time	SI	tDHS	12	-	-	ns	
CSB setup time	/CS	tCSS	12	-	-	ns	
CSB hold time	/CS	tCHS	20	-	-	ns	

2 Write to the Memory

Before powering on G2 COG Driver, the developer should write the new pattern to image buffer, either SRAM or flash memory. The image pattern must be converted to a 1 bit bitmap format (Black/White) prior to writing.

One buffer space should be allocated to store new pattern. The new pattern will be written to the EPD. The table below list the buffer space size required for each EPD size.

EPD size	Image resolution(pixels)	image Buffer (bytes)
1.44”	128 x 96	1,536
2"	200 x 96	2,400
2.7”	264 x 176	5,808

3 Power On G2 COG Driver

This flowchart describes power on sequence for the G2 COG Driver.

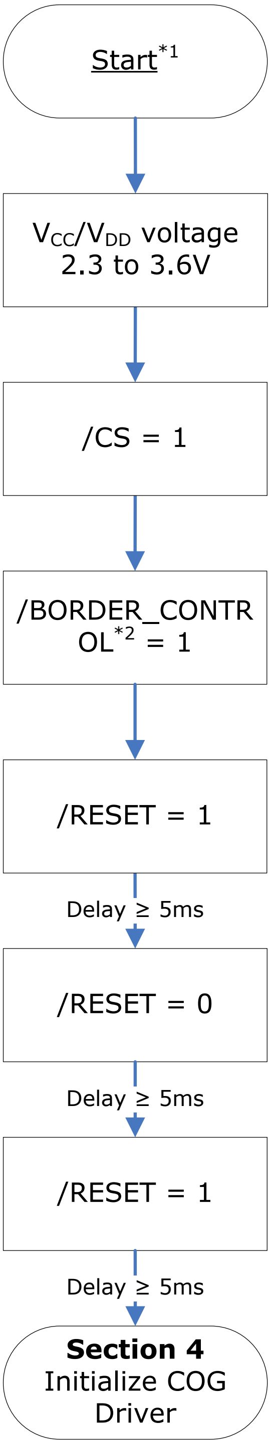
1. Start :

   Initial State:

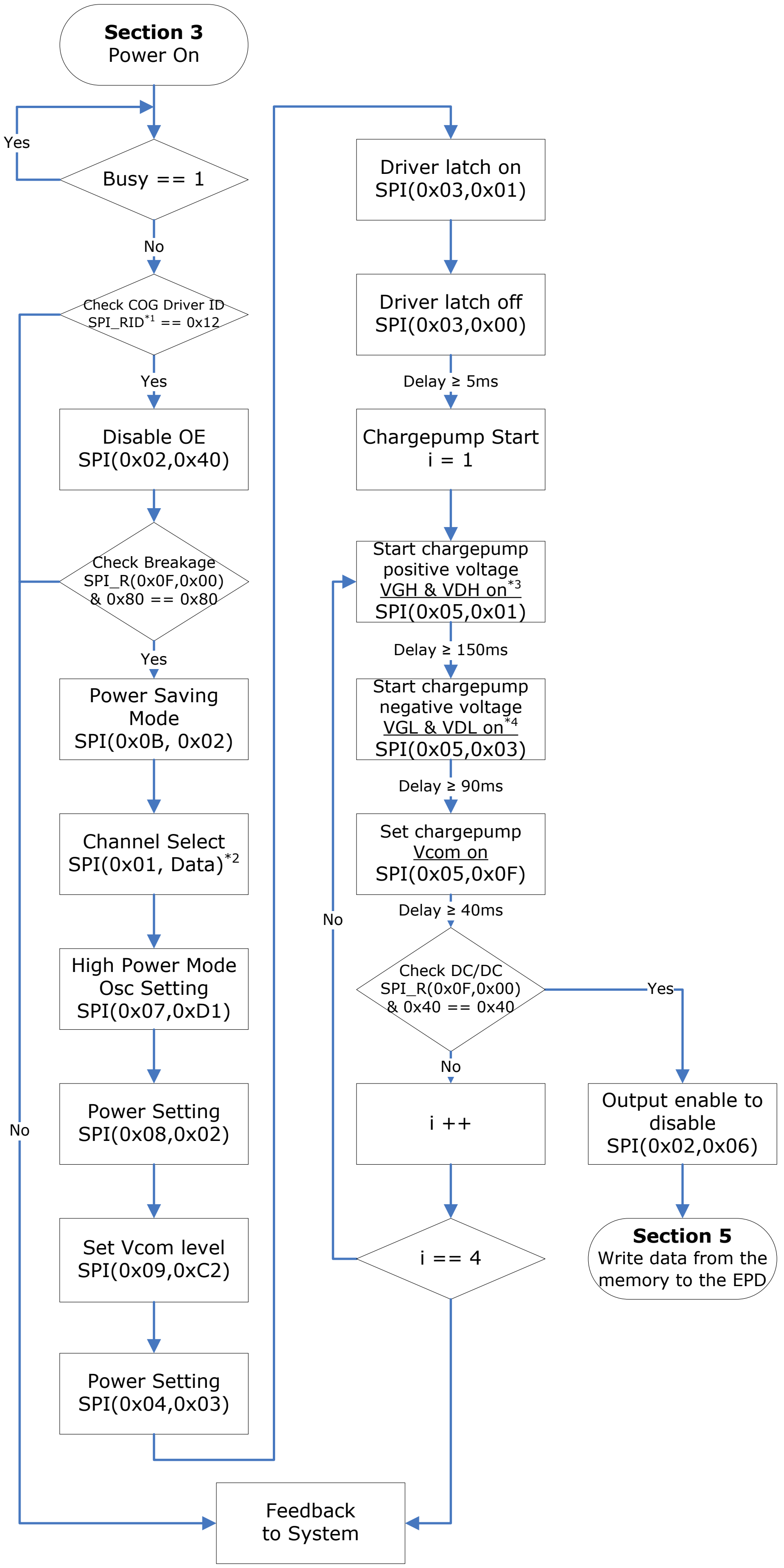
   $V_{CC}/V_{DD} = 0$

   /RESET, /CS, /BORDER_CONTROL*2, SI, SCLK = 0

2. /BORDER_CONTROL:
   GPIO pin for 2.7" EPD Panel to reset the Border.



4 Initialize G2 COG Driver



Note:

1. SPI timing format (Refer to ”1.6 SPI Timing Format” for detail)
    - SPI read COG Driver ID command: SPI_RID
        - ■ G1 COG Driver ID is 0x11
        - ■ G2 COG Driver ID is 0x12
2. SPI(0x01, Data):
    - Different by each size
        - ■ 1.44”: SPI(0x01, (0x0000,0000,000F,FF00))
        - ■ 2”: SPI(0x01, (0x0000,0000,01FF,E000))
        - ■ 2.7”: SPI(0x01, (0x0000,007F,FFFE,0000))
    - Take 2” for example, to send first byte protocol (0x70) before Register Index (0x01), and then send second byte protocol (0x72) before Register Data (0x0000,0000,01FF,E000).

3. Should measure VGH >12V and VDH >8V
4. Should measure VGL <-12V and VDL <-8V

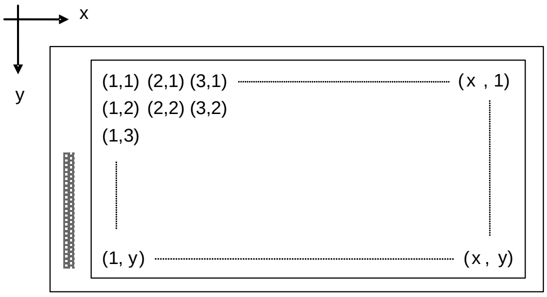
5 Write Data from the Memory to the EPD

This section describes how data should be sent to the G2 COG Driver which will update the display. The G2 COG Driver uses a buffer to store a line of data and then writes to the display.

5.1 Data Structure

- EPD Resolutions

EPD size	Image resolution(pixels)	X	Y
1.44”	128 x 96	128	96
2"	200 x 96	200	96
2.7”	264 x 176	264	176



- Data components
  - One Bit – A bit can be W (White), B (Black) or N (Nothing) bit. Using the N bit to keep the pixel for the same color if new bit is same as previous one and also avoid overdriving the pixel to reduce ghosting phenomenon.
  - One Dot/pixel is comprised of 2 bits.
  - One Line is the number of dots in a line.

    For example:

    ■ The 1.44” uses 128 Dots to represent 1 Line.

    ■ The 2” uses 200 Dots to represent 1 Line.

    ■ The 2.7” uses 264 Dots to represent 1 Line.

    ■ The G2 COG Driver uses a buffer to write one line of data (Mapping) – interlaced. The table below is a Line data of 2" including Odd data bytes, Scan bytes and Even data bytes.

Data Bytes	Scan bytes	Data Bytes
1st – 25th (Odd)	1st - 24th	26th – 50th (Even)
{D(199,y),D(197,y), D(195, y), D(193, y)}…{D(7,y),D(5,y), D(3,y), D(1,y)}	{S(96),S(95), S(94),S(93)}…{S(4),S(3), S(2), S(1)}	{D(2,y),D(4,y), D(6,y), D(8,y)}…{D(194,y),D(196,y), D(198,y), D(200,y)}

  - One frame of data is the number of lines * rows.

  For example:

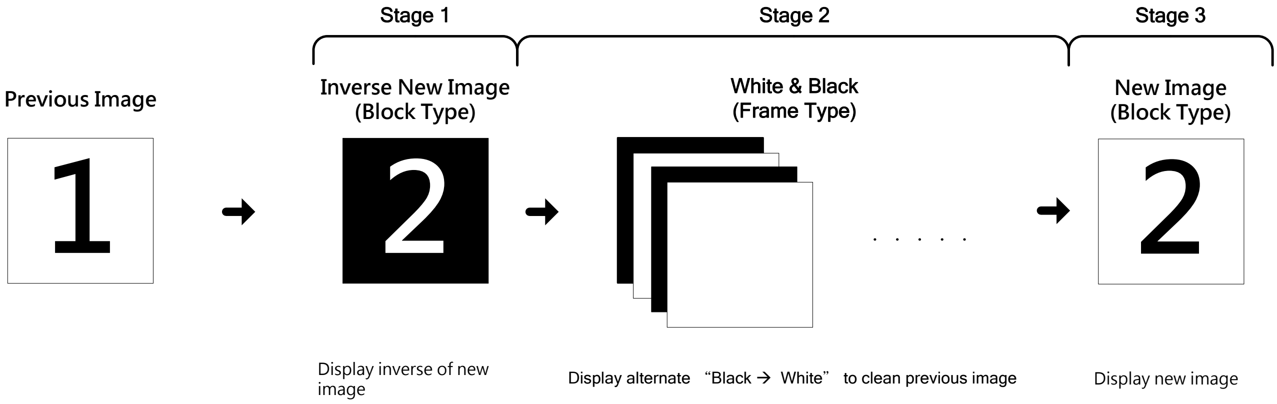
    ■ The 1.44” frame of data is 96 lines * 128 dots.

    ■ The 2” frame of data is 96 lines * 200 dots.

    ■ The 2.7” frame of data is 176 lines * 264 dots.

5.2 Overall Update Flow

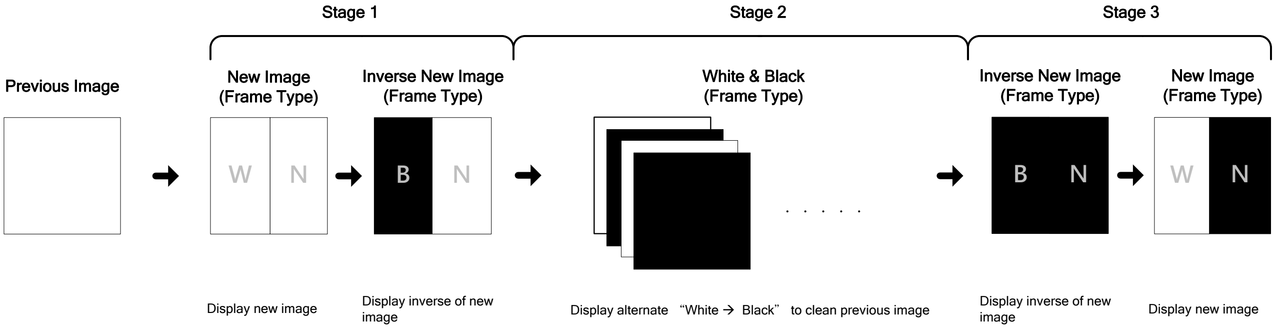
5.2.1 Room Temperature Flow (0℃≦T≦50℃)



Note:

(1) If the ghosting is at unacceptable level, users can increase the Repeat cycle (in table of section 5.5.1) or the time (Tx) for each White and Black of Stage 2.

5.2.2 Low Temperature Flow (-20℃≦T<0℃)



	New image W	New image B
Input W	W	N

	New image W	New image B
Input B	B	N

	New image W	New image B
Input B	B	N

	New image W	New image B
Input W	W	N

Note:

(1) If the ghosting is at unacceptable level, users can increase the Repeat cycle (in table of section 5.5.1) or the time (T_x) for each White and Black of Stage 2.

(2) Highly recommend updating above full cycle every 6 hours to get good ghosting level and optical performance.

(3) For -25℃ to -20℃, please contact us for more information.

5.3 Store a Line of Data in the Buffer

This section describes the details of how to send data to the G2 COG Driver. The G2 COG Driver uses a buffer to update the display line by line.

- 1.44” Input Data Order

Note :

1. Turn on OE :
   Output data from COG driver to panel.

2.

Data	bit1	bit0	Input

D(x,y)
x = 1~128
y = 1~96
	1	1	Black (B)

	1	0	White (W)
	0	0	Nothing (N)

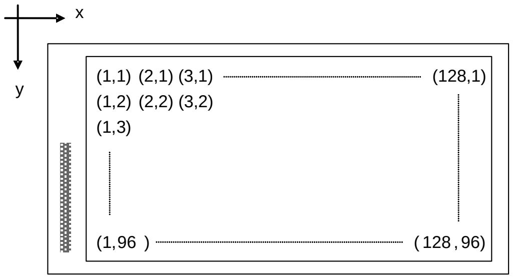
Example:
D(127,y) = Black (B) = 11
D(125,y) = White (W)= 10
D(123,y) = Nothing(N) = 00
D(121,y) = Black (B) = 11
→ 1st Data Byte= 11,10,00,11

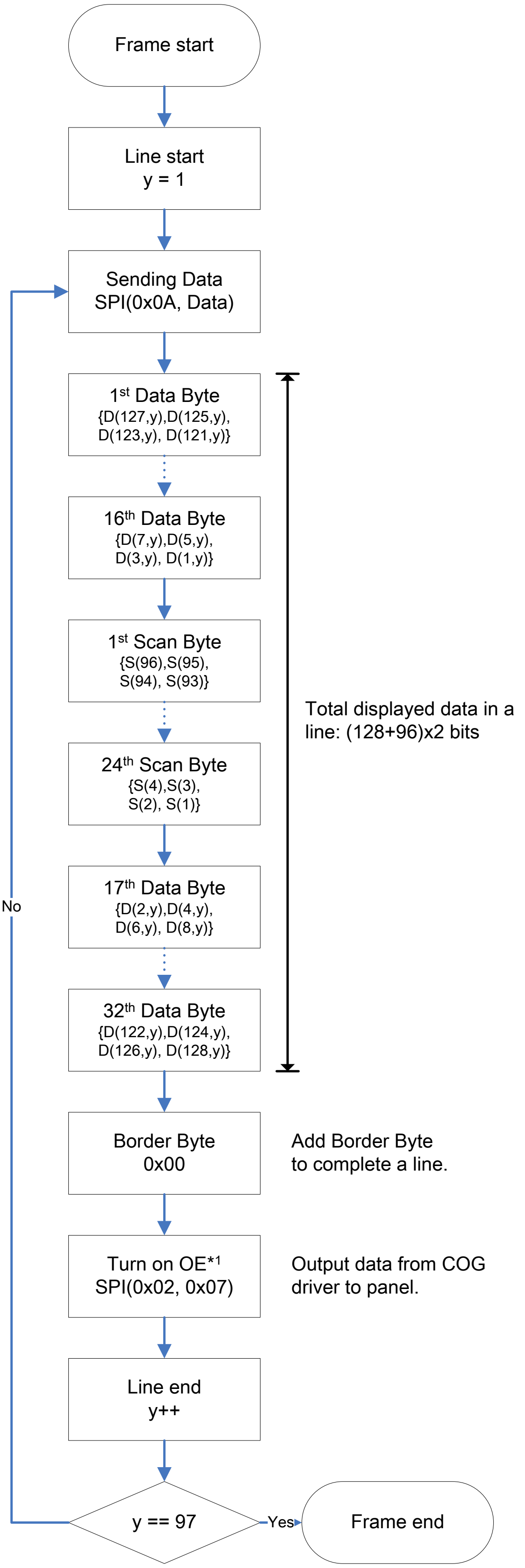
Scan	bit1	bit0	Input
S(1) ~S(96)	1	1	Scan on
	0	0	Scan off

Example:
When y = 2,
→ Only S(2) is Scan on (11) while others are Scan off (00). The image represented by Data Bytes will be displayed on 2nd horizontal line (i.e. Dot(1,2) ~ Dot(128,2)).

S(1)	= Scan off	= 00
S(2)	= Scan on	= 11
S(3)	= Scan off	= 00
S(4)	= Scan off	= 00
	:	
S(96)	= Scan off	= 00

→ 1st ~ 23rd Scan Byte = 00,00,00,00
→ 24th Scan Byte = 00,00,11,00





Total displayed data in a line: (128+96)x2 bits

Add Border Byte to complete a line.

Output data from COG driver to panel.

• 2″ Input Data Order

Note :

1. Turn on OE :
   Output data from COG driver to panel.

2.

Data	bit1	bit0	Input

D(x,y)
x = 1~200
y = 1~96
	1	1	Black (B)

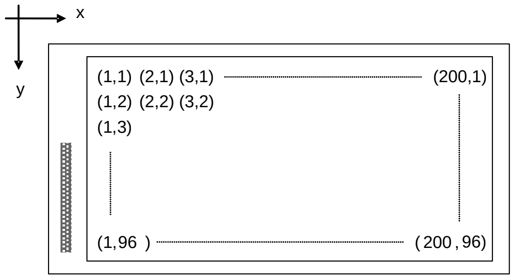
	1	0	White (W)
	0	0	Nothing (N)

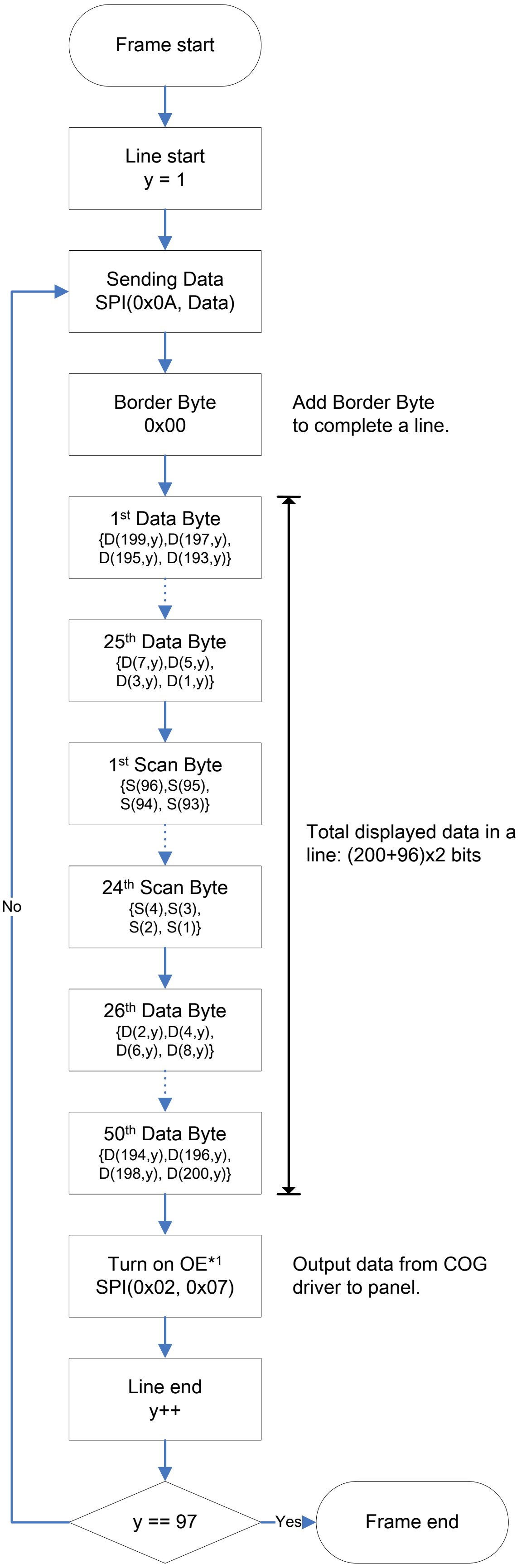
Example:
D(199,y) = Black (B) = 11
D(197,y) = White (W)= 10
D(195,y) = Nothing (N) = 00
D(193,y) = Black (B) = 11
→ 1st Data Byte= 11,10,00,11

Scan	bit1	bit0	Input
S(1) ~S(96)	1	1	Scan on
	0	0	Scan off

Example:
When y = 2,
→ Only S(2) is Scan on (11) while others are Scan off (00). The image represented by Data Bytes will be displayed on 2nd horizontal line (i.e. Dot(1,2) ~ Dot(200,2)).

S(1) = Scan off = 00
S(2) = Scan on = 11
S(3) = Scan off = 00
S(4) = Scan off = 00
:
S(96) = Scan off = 00
→ 1st ~ 23rd Scan Byte = 00,00,00,00
→ 24th Scan Byte = 00,00,11,00





Add Border Byte to complete a line.

Total displayed data in a line: (200+96)x2 bits

Output data from COG driver to panel.

• 2.7” Input Data Order

Note :

1. Turn on OE :
   Output data from COG driver to panel.

2.

Data	bit1	bit0	Input

D(x,y)
x = 1~264
y = 1~176
	1	1	Black (B)

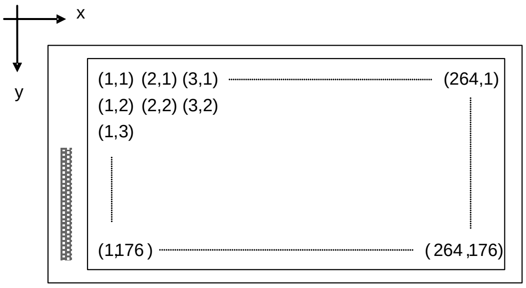
	1	0	White (W)
	0	0	Nothing (N)

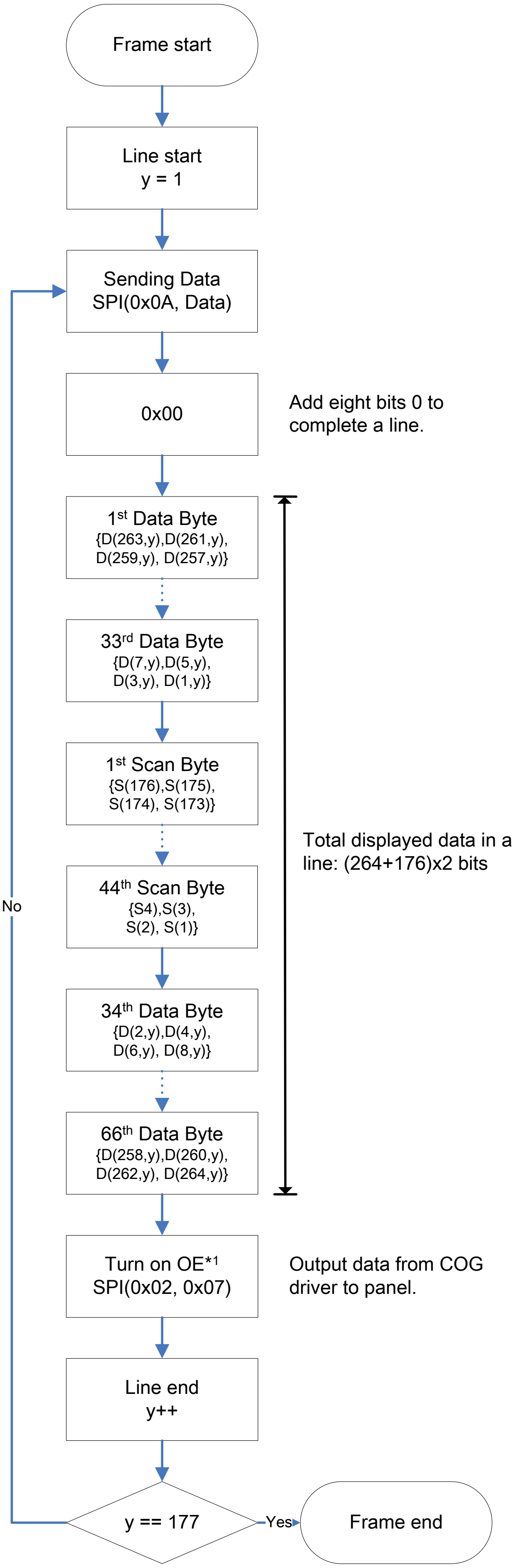
Example:
D(263,y) = Black (B) = 11
D(261,y) = White (W)= 10
D(259,y) = Nothing (N) = 00
D(257,y) = Black (B) = 11
→ 1st Data Byte= 11,10,00,11

Scan	bit1	bit0	Input
S(1) ~S(176)	1	1	Scan on
	0	0	Scan off

Example:
When y = 2,
→ Only S(2) is Scan on (11) while others are Scan off (00). The image represented by Data Bytes will be displayed on 2nd horizontal line (i.e. Dot(1,2) ~ Dot(264,2)).

S(1) = Scan off = 00
S(2) = Scan on = 11
S(3) = Scan off = 00
S(4) = Scan off = 00
:
S(176) = Scan off = 00
→ 1st ~ 43rd Scan Byte = 00,00,00,00
→ 44th Scan Byte = 00,00,11,00

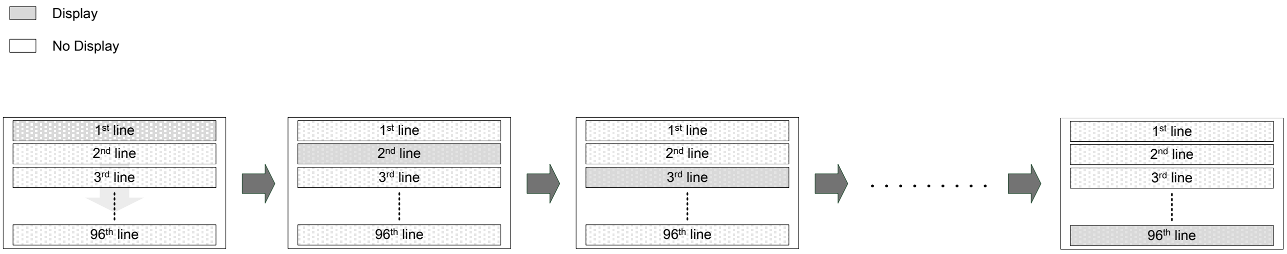



Frame start
Line start
y = 1
Sending Data
SPI(0x0A, Data)
0x00
Add eight bits 0 to complete a line.
1st Data Byte
{D(263,y),D(261,y), D(259,y), D(257,y)}
33rd Data Byte
{D(7,y),D(5,y), D(3,y), D(1,y)}
1st Scan Byte
{S(176),S(175), S(174), S(173)}
44th Scan Byte
{S4),S(3), S(2), S(1)}
34th Data Byte
{D(2,y),D(4,y), D(6,y), D(8,y)}
66th Data Byte
{D(258,y),D(260,y), D(262,y), D(264,y)}
Total displayed data in a line: (264+176)x2 bits
Turn on OE*1
SPI(0x02, 0x07)
Output data from COG driver to panel.
Line end
y++
y == 177
No
Yes
Frame end

5.4 Writing to the Display in Stages

This section contains the method to write to the display in stages. Rewrite the frame during each stage. There are two different ways, Block Type and Frame type, to scan the display.

5.4.1 Frame Type



Note:

The image frame needs a continuous scan and send data of line from top to bottom line by line to complete a frame update. Repeat the frame update until fulfilling the defined driving time (T_1, T_2) at the table of section 5.51.

- **Frame Type Flow**

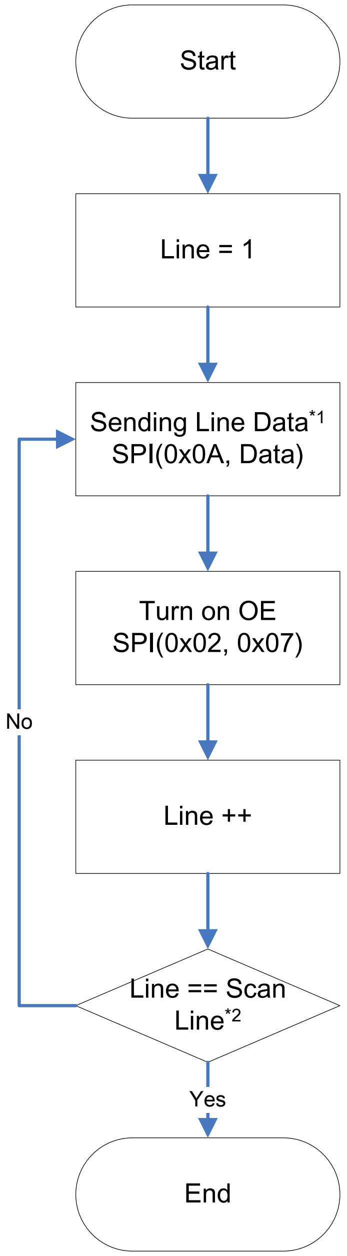
Note:

1. Line Data :

Size	Line Data (include Border Byte)
1.44"	57 Bytes
2"	75 Bytes
2.7"	111 Bytes

2. Scan Line :

Size	Scan Line
1.44"	97
2"	97
2.7"	177



5.4.2 Block Type

This type drives the portion lines of an EPD. It can reduce the refresh or recharge time of the line. Therefore need to shift the portion lines (*Block*) in a fixed number of lines (*Step*) until finish all lines (*First of repeat number*). The last Repeat number needs to drive "Nothing" at the last *Step* in each *Block*. The Nothing is included in *Block*. In this type of scan, you need to decide three variables. *Repeat number*, *Block* and *Step*, use these three variables to complete a stage.

***Block*:**

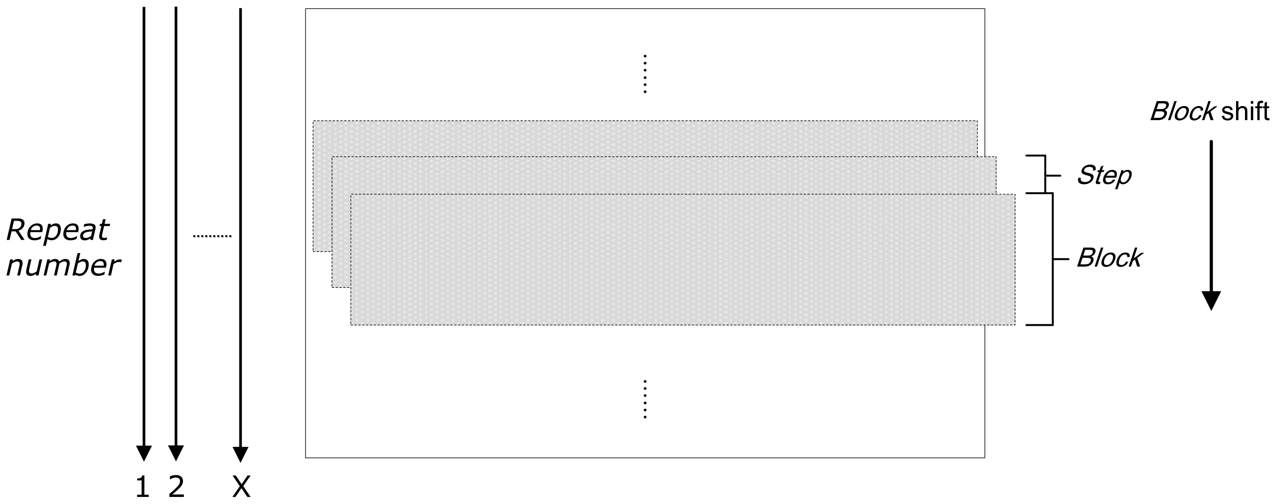
Block is a portion of total lines of EPD. It can decide the refresh or recharge time of each line. A smaller *Block* gets the fast refresh or recharge time of each line, whereas a bigger *Block* would be slower. And the number of *Block* should be divisible by the number of *Step*.

***Step*:**

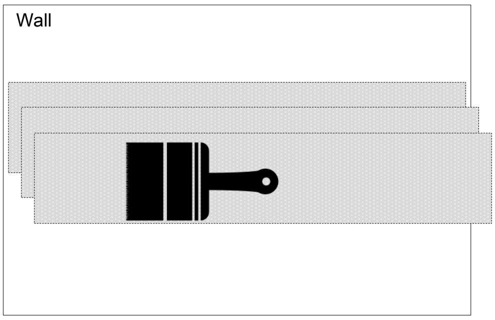
Step means the *Block* will be shifted how many lines. It decides each line will be scanned how many times. A smaller *Step* gets more scan times, whereas a bigger *Step* would be less. And the number of total lines should be divisible by the number of *Step*.

***Repeat number*:**

Repeat number is numbers of *Block* moving from top to bottom of display.



Block Type update is similar to painting a wall by a brush. The width of brush is *Block*. Every time, you paint horizontally then move down certain distance, *Step.* Then paint again until finish the whole wall. If the painting is not thick enough, you will repeat above action again until getting thick enough painting. The repetition is *Repeat number*.



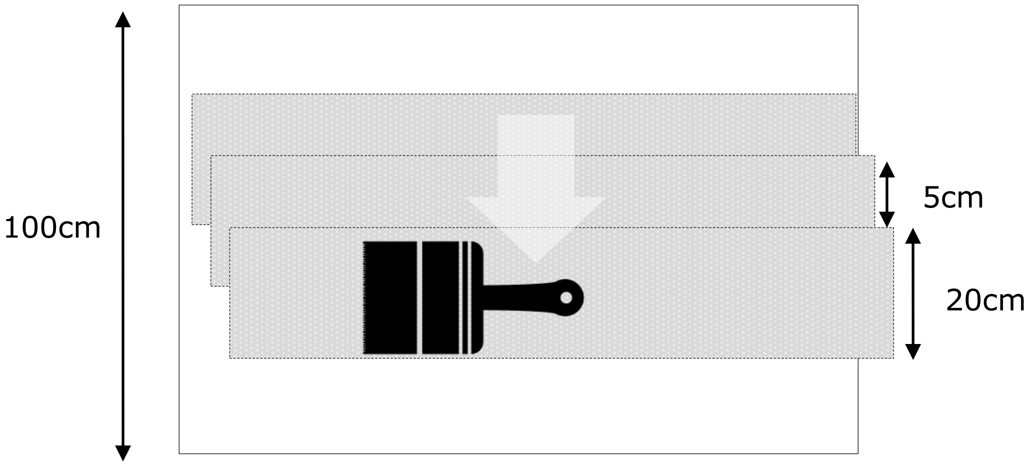
Example:

A 100cm height wall, a 20cm width brush (*Block)*, we move down brush in 5cm (*Step)*. We must ensure the painting on the whole wall is same thickness.

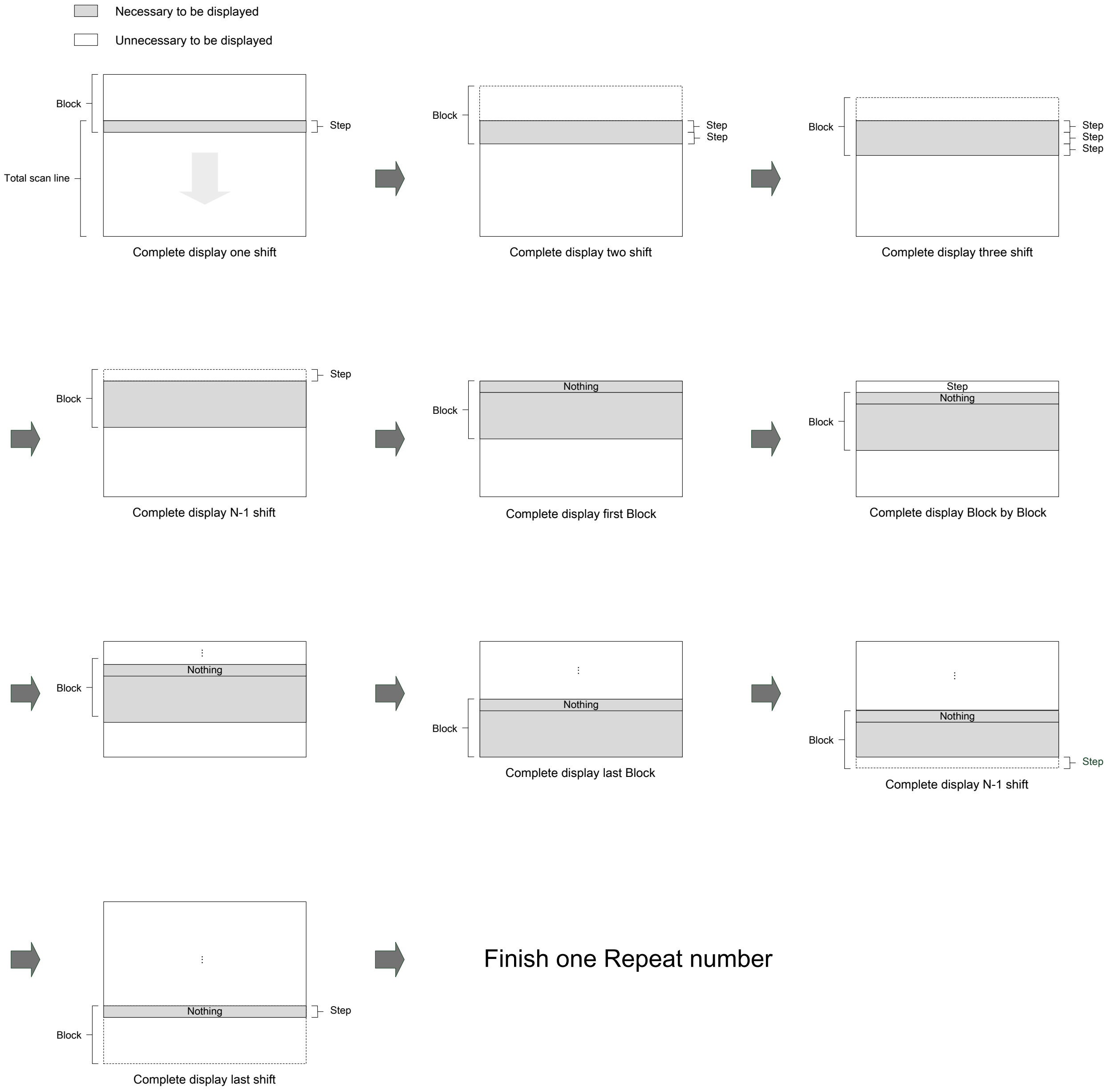
We paint the wall by this order.

(First shift) 0~5 → 0~10 → 0~15 → 0~20 → 5~25 → … → 80~100 → 95~100(Last shift)

In this case, every cm of this wall is painted twice. If twice is not enough. We can repeat from First *Step* to Last *Step* one more time to get sufficient thickness.



- **Repeat number 1 ~ N**


Necessary to be displayed
Unnecessary to be displayed
Block
Step
Total scan line
Complete display one shift
Block
Step
Step
Complete display two shift
Block
Step
Step
Step
Complete display three shift
Step
Block
Complete display N-1 shift
Nothing
Block
Complete display first Block
Step
Nothing
Block
Complete display Block by Block
Block
Nothing
Nothing
Block
Complete display last Block
Nothing
Block
Step
Complete display N-1 shift
Nothing
Step
Block
Complete display last shift
Finish one Repeat number

Note:

In Block, the last height of Step to send Nothing data bytes.

• Block Type Flow

Note:

1. Line Data:

Size	Line Data (include Border Byte)
1.44"	57 Bytes
2"	75 Bytes
2.7"	111 Bytes

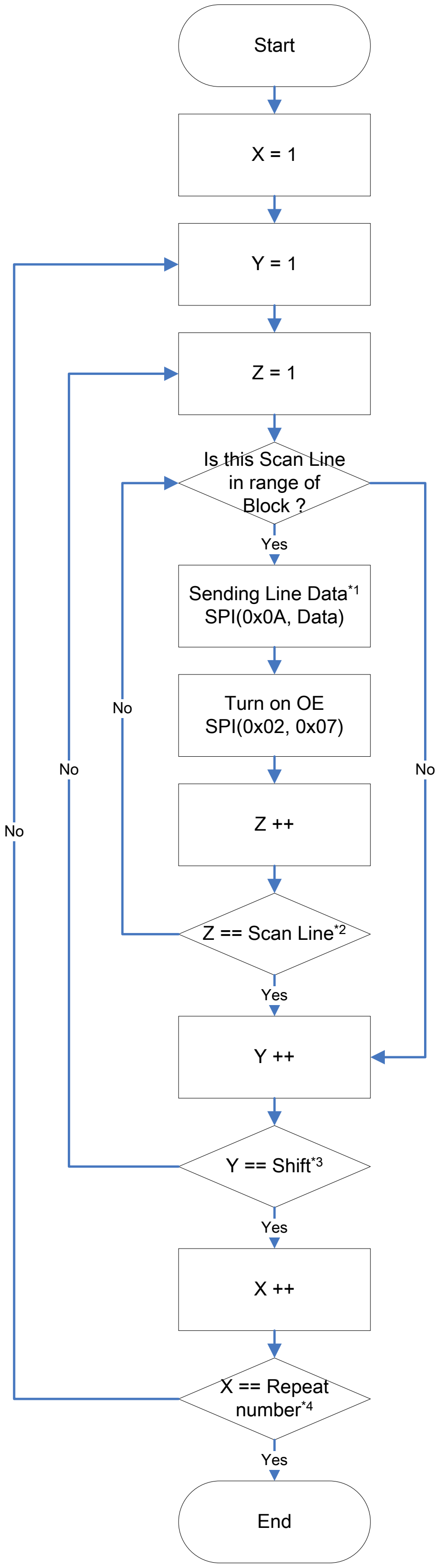
2. Scan Line:

Size	Scan Line
1.44"	97
2"	97
2.7"	177

3. Shift:
Total numbers for finish one Repeat number.

$$Display = \frac{\text{Total line} + [Block - Step]}{Step}$$

4. Repeat number:
Block moving from first Shift to last Shift of display.

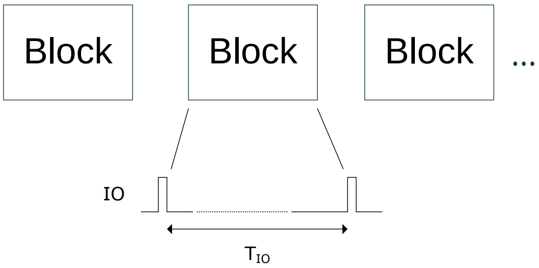


- **Method to Select a suitable *Block/Step/Repeat number***

Procedure:

1. Determine the suitable width of ***Block***

   First set a random value for width of *Block*, this value must be an integer. Use a free GPIO signal (IO) from Timing Controller, measure the period of IO to get T_{IO}*2 by oscilloscope. If the value is not quite close to Block time, we must adjust the value for width of *Block*



$$T_{IO}^{*2} \leq \text{Block time}^{*1}$$

Note:

1. Block time: Complete update one *Block* to display on panel, the value is defined in section 5.5.1
2. T_{IO} must be less than Block time
   (PDi recommend T_{IO} close to Block time)

2. Determine the suitable width of ***Step***

   Use T_{IO}, *Block* and $T_{DRVLINE}$*1 three parameters to calculate *Step* by below formula, *Step* must be an integer.

   Both *Block* and Total scan line should be divisible by *Step*. In case *Step* is hard to divisible *Block*, please go back to procedure 1 to set a new value for width of *Block*.

   If *Block** T_{IO} is less than $T_{DRVLINE}+T_{IO}$, *Step* must be set to"1". Than please use the new formula in procedure 3 to determine ***Repeat number***.

$$\mathit{Step} \leq \frac{\mathit{Block} * T_{IO}}{T_{DRVLINE}^{*1} + T_{IO}}$$

Note:

1. $T_{DRVLINE}$: The value is defined in section 5.5.1

3. Determine the suitable width of ***Repeat number***

   Use T_{IO}, *Block*, $T_{DRVLINE}$*1 and *Repeat number* four parameters to calculate *Step* by below formula

   If image contrast ratio is unacceptable, first decrease the width of *Step* if it's not 1, than increase *Repeat number* to get higher contrast ratio.

$$\textbf{\textit{Repeat number}} \geq \frac{T_{DRVLINE}{}^{*1}}{[\textit{Block} - 1] * T_{IO}}$$

Note:

1. $T_{DRVLINE}$: The value is defined in section 5.5.1

Example:

1. For 2.7 inch (resolution: 264*176) with PDi jig, first we set 6 for width of Block, we can get T_{IO} = 2.6ms by oscilloscope, but the value is not quite close to Block time. Then we set 24 for width of Block, assume we can get T_{IO} = 10.4ms by oscilloscope. Which is same as defined Block time.

   Block = **24**

2. Get Block time and $T_{DRVLINE}$ from section 5.5.1, T_{IO} = 10.4ms and $T_{DRVLINE}$ = 250ms @25°C

$$\textit{Step} \leq \frac{24 * 10.4}{250 + 10.4} \Rightarrow \textit{Step} \leq \frac{249.6}{260.4}$$

249.6(numerator) is less than 260.4(denominator), *Step* = **1**

$$\textbf{\textit{Repeat number}} \geq \frac{250}{23 * 10.4} \Rightarrow \textbf{\textit{Repeat number}} \geq \frac{250}{239.2}$$

According to the new formula, suitable width of *Repeat number* = **1**$_{\text{(min)}}$

- **Formula for calculate Total update time**

$$\text{Time of one } \textit{Repeat number} = \frac{\text{Total line}}{\textit{Step}} * T_{IO}$$

Stage 1: T_{stage1} = Time of one Repeat number * Repeat number

Stage 2: T_{stage2} = (T_1 + T_2) * repeat cycle

Stage 3: T_{stage3} = Time of one Repeat number * Repeat number

➢ Total update time = $T_{stage1} + T_{stage2} + T_{stage3}$

Example:

Use 2.7 inch for example, parameters as below (@25 °C; PDi jig)

T_{IO}	Block	Step	Total line	Repeat number	T_1	T_2	Repeat cycle
10.4ms	24	1	176	1	155ms	155ms	2

Stage 1: T_{stage1} = 1830.4 * 1 = 1830.4ms

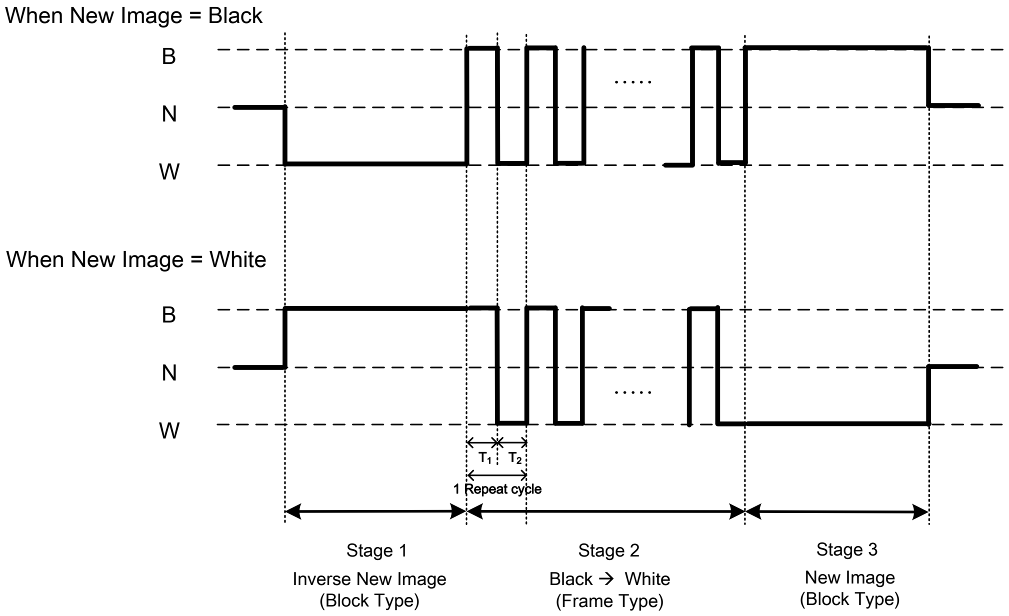
Stage 2: T_{stage2} = (155 + 155) * 2 = 620ms

Stage 3: T_{stage3} = 1830.4 * 1 = 1830.4ms

➢ Total update time = $T_{stage1} + T_{stage2} + T_{stage3}$ = **4280.8ms**

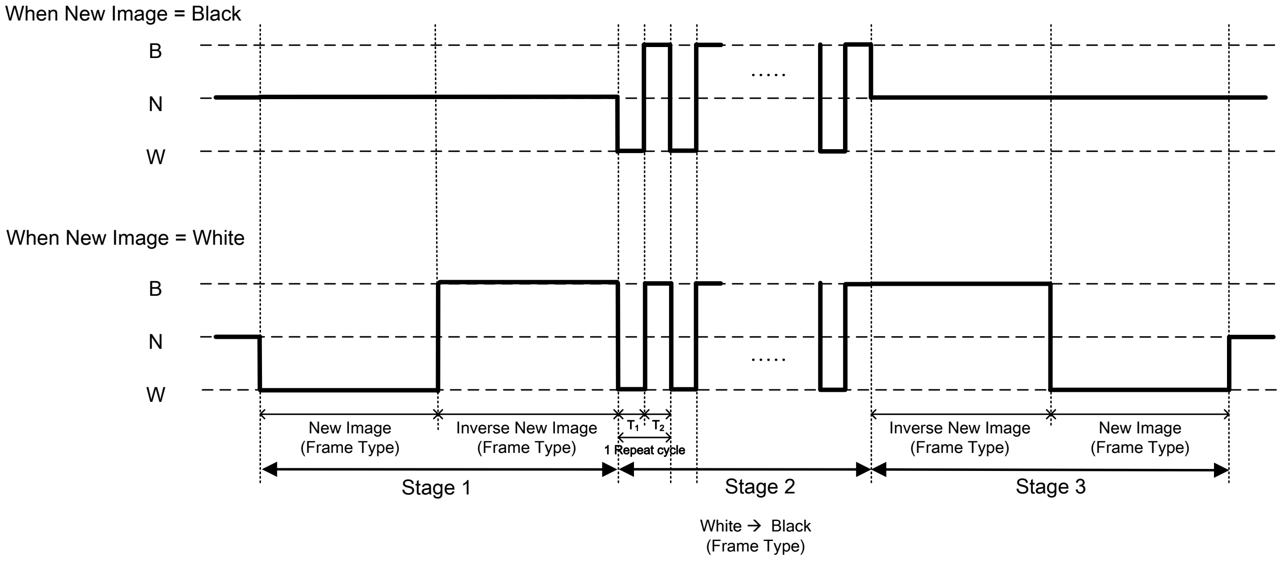
5.5 Driving Waveform and Temperature Factor in Stages

5.5.1 Room Temperature (0℃≦T≦50℃)



Temperature Range (℃)	Panel Size (inch)	Stage 1		Stage 2			Stage 3	
		Block time (ms)	TDRVLINE (ms)	T_1 (ms)	T_2 (ms)	Repeat cycle	Block time (ms)	TDRVLINE (ms)
T ＞ 45	2.7	4.9	29.9	280	280	1	4.9	29.9
	2	2.7	13.2	250	250	1	2.7	13.2
45 ≧ T ＞ 40	2.7	9.3	103	280	280	1	9.3	103
	2	3.2	19.4	250	250	1	3.2	19.4
40 ≧ T ＞ 35	2.7	9.3	103	280	280	2	9.3	103
	2	5.0	45	250	250	2	5.0	45
35 ≧ T ＞ 30	2.7	14.4	66.9	280	280	2	14.4	66.9
	2	7.3	95.4	250	250	2	7.3	95.4
30 ≧ T ＞ 20	2.7	14.4	127	305	305	2	14.4	127
	2	12	129	280	280	2	12	129
20 ≧ T ＞ 10	2.7	26.4	148	455	455	3	26.4	148
	2	14.4	182	320	320	3	14.4	182
10 ≧ T ≧ 0	2.7	35	730	450	450	4	35	730
	2	23.6	250	350	350	4	23.6	250

5.5.2 Low Temperature (-25℃≦T＜0℃)



1.44" Temperature Factor

Temperature Range (℃)	Stage 1		Stage 2			Stage 3		Total Time (Sec)
	New Image (ms)	Inverse New Image (ms)	T1 (ms)	T2 (ms)	Repeat cycle	Inverse New Image (ms)	New Image (ms)	
0 ≧ T ＞ -5	245	1960	950	950	12	245	1960	27.2
-5 ≧ T ＞ -10	445	3560	1250	1250	9	445	3560	30.5
-10 ≧ T ＞ -15	745	5960	1650	1650	12	745	5960	53
-15 ≧ T ＞ -20	1045	8360	2250	2250	12	1045	8360	72.8
≦ -20 (TBD)	1345	10760	2540	2540	13	1345	10760	90.2

2.0" Temperature Factor

Temperature Range (℃)	Stage 1		Stage 2			Stage 3		Total Time (Sec)
	New Image (ms)	Inverse New Image (ms)	T_1 (ms)	T_2 (ms)	Repeat cycle	Inverse New Image (ms)	New Image (ms)	
0 ≧ T ＞ -5	245	1960	950	950	12	245	1960	27.2
-5 ≧ T ＞ -10	445	3560	1250	1250	9	445	3560	30.5
-10 ≧ T ＞ -15	745	5960	1650	1650	12	745	5960	53
-15 ≧ T ＞ -20	1045	8360	2250	2250	12	1045	8360	72.8
≦ -20 (TBD)	1345	10760	2540	2540	13	1345	10760	90.2

2.7" Temperature Factor

Temperature Range (℃)	Stage 1		Stage 2			Stage 3		Total Time (Sec)
	New Image (ms)	Inverse New Image (ms)	T_1 (ms)	T_2 (ms)	Repeat cycle	Inverse New Image (ms)	New Image (ms)	
0 ≧ T ＞ -5	545	4360	1080	1080	10	545	4360	31
-5 ≧ T ＞ -10	745	5960	1580	1580	9	745	5960	41.8
-10 ≧ T ＞ -15	1045	8360	2180	2180	10	1045	8360	62.4
-15 ≧ T ＞ -20	1345	10760	2580	2580	9	1345	10760	70.6
≦ -20 (TBD)	1680	13440	2680	2680	11	1680	13440	89.2

Note:

1. Line time of PDi Jig is 0.432ms. (@2.7 inch EPD Panel)
2. The guaranteed operation temperature range is listed in PDI EPD datasheets. The temperature range listed in this document is only for timing controller programming reference.
3. This table is tested with PDI jig.
4. If the ghosting is at unacceptable level, the EPD can be rewritten more cycles.

5.5.3 The Structure of the stage

- **Room Temperature** (0℃≦T≦50℃)

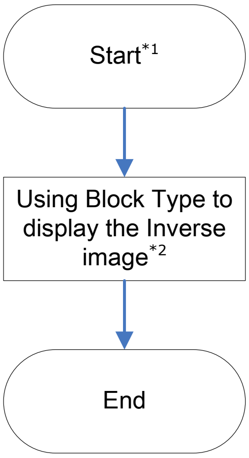
Stage 1 Inverse New Image

Note:

1. Start :
   Follow the end of COG initial sequence.

2. Inverse image :
   Input the inverse image of the new pattern.

3. New :
   The ”New” mean the image that user wants to show on EPD.



Inverse	Data		Border
New*3	B	W	
Input	W	B	N

Stage 2 White & Black

Note:

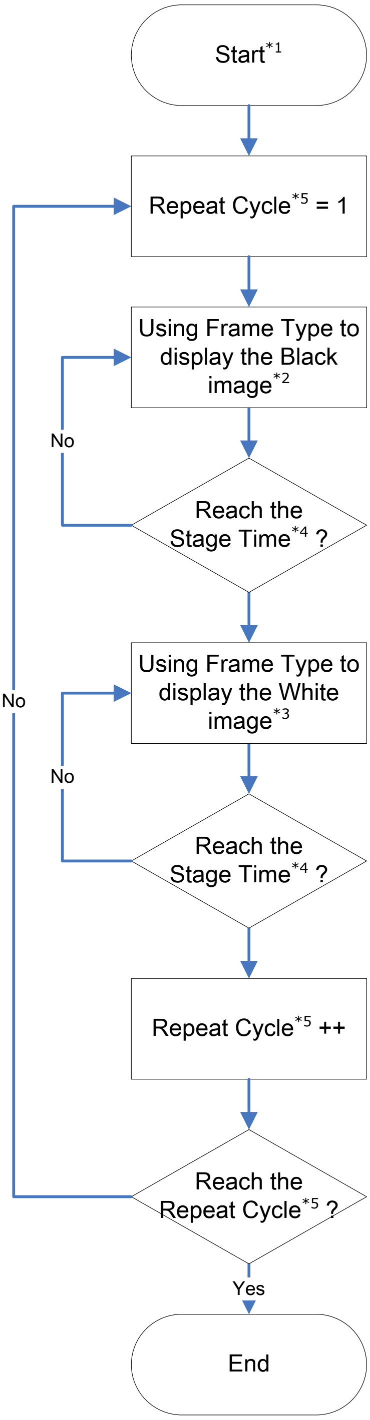
1. Start :
Follow the end of the Stage 1.

2. Black image :
Input the Black image by Frame Type. The Stage Time is different at different temperature.

3. White image :
Input the White image by Frame Type. The Stage Time is different at different temperature.

4. Stage Time :
Total display time for White image and Black image.

5. Repeat Cycle :
White image and Black image repeat times. The repeat times is different at different temperature.


Start*1
Repeat Cycle*5 = 1
Using Frame Type to display the Black image*2
No
Reach the Stage Time*4 ?
Using Frame Type to display the White image*3
No
No
Reach the Stage Time*4 ?
Repeat Cycle*5 ++
Reach the Repeat Cycle*5 ?
Yes
End

Black	Data		Border
Input	B	B	N

White	Data		Border
Input	W	W	N

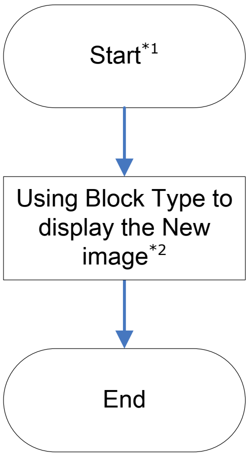
Stage 3 New Image

Note:

1. Start :
   Follow the end of the Stage 2.

2. New image :
   Input the New image of the new pattern.

3. New :
   The "New" mean the image that user wants to show on EPD.



Inverse	Data		Border
New*3	B	W	
Input	B	W	N

▪ Low Temperature (-25℃≦T<0℃)

Stage 1

• New Image

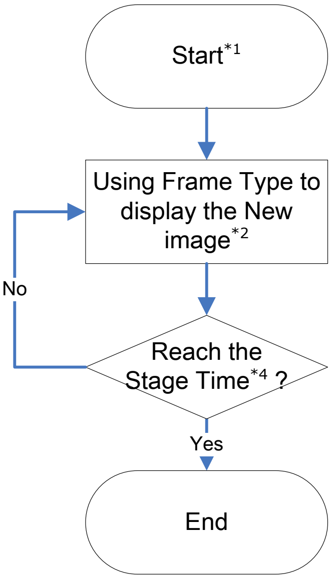
Note:

1. Start :
   Follow the end of COG initial sequence.

2. New image :
   Input the New image of the new pattern.

3. New :
   The "New" mean the image that user wants to show on EPD.

4. Stage Time :
   The New image display times.



	Data		Border
New[3]	B	W	
Input	N	W	N

• Inverse New Image

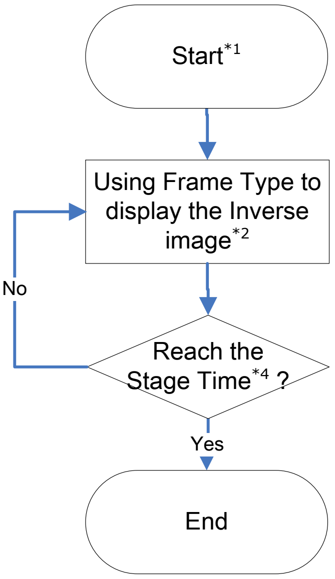
Note:

1. Start :
   Follow the end of New Image at Stage 1.

2. Inverse image :
   Input the inverse image of the new pattern.

3. New :
   The "New" mean the image that user wants to show on EPD.

4. Stage Time :
   The Inverse image display times.



Inverse	Data		Border
New[3]	B	W	
Input	N	B	N

Stage 2 White & Black

Note:

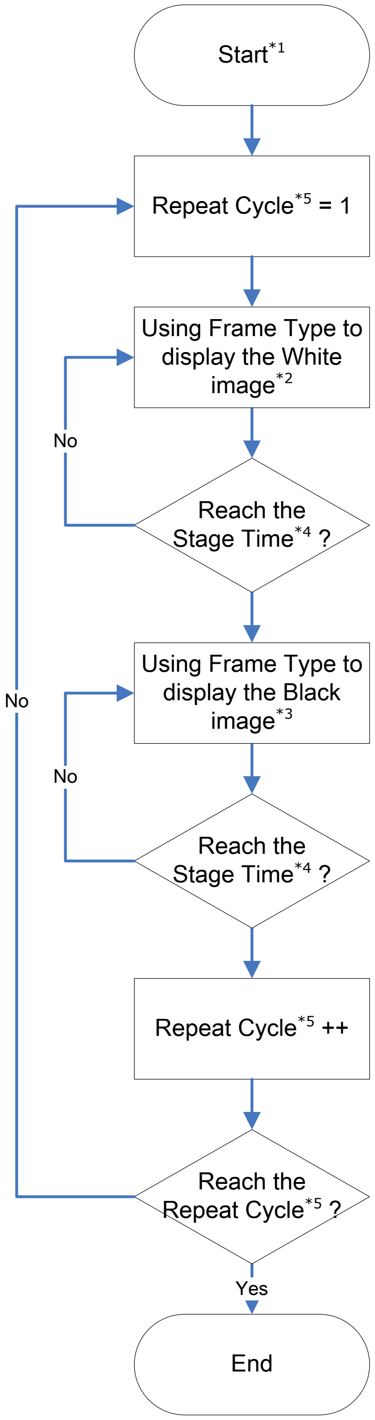
1. Start :
   Follow the end of the Stage 1.

2. White image :
   Input the White image by Frame Type. The Stage Time is different at different temperature.

3. Black image :
   Input the Black image by Frame Type. The Stage Time is different at different temperature.

4. Stage Time :
   Total display time for White image and Black image.

5. Repeat Cycle :
   White image and Black image repeat times. The repeat times is different at different temperature.


Start*1
Repeat Cycle*5 = 1
Using Frame Type to display the White image*2
No
Reach the Stage Time*4 ?
Using Frame Type to display the Black image*3
No
No
Reach the Stage Time*4 ?
Repeat Cycle*5 ++
Reach the Repeat Cycle*5 ?
Yes
End

Black	Data		Border
Input	W	W	N

White	Data		Border
Input	B	B	N

Stage 3

• Inverse New Image

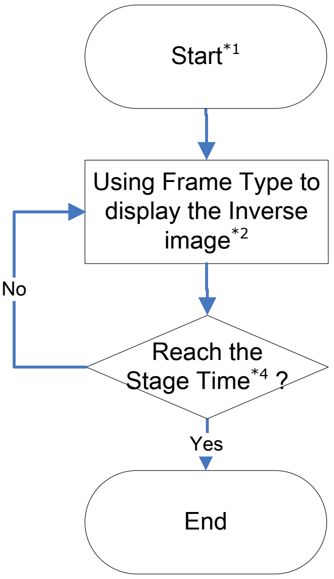
Note:

1. Start :
   Follow the end of Stage 2.

2. Inverse image :
   Input the inverse image of the new pattern.

3. New :
   The ”New” mean the image that user wants to show on EPD.

4. Stage Time :
   The Inverse image display times.


Start[*1]
Using Frame Type to display the Inverse image[*2]
Reach the Stage Time[*4] ?
No
Yes
End

Inverse	Data		Border
New[*3]	B	W	
Input	N	B	N

• New Image

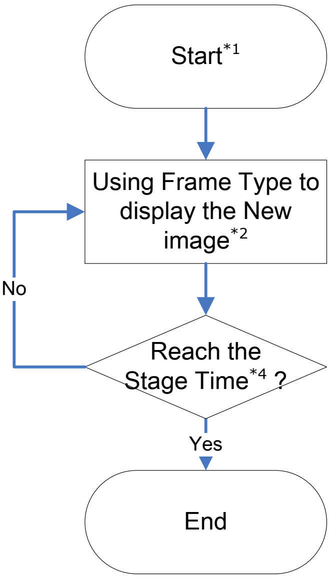
Note:

1. Start :
   Follow the end of Inverse New Image at Stage 3.

2. New image :
   Input the New image of the new pattern.

3. New :
   The ”New” mean the image that user wants to show on EPD.

4. Stage Time :
   The New image display times.


Start[*1]
Using Frame Type to display the New image[*2]
Reach the Stage Time[*4] ?
No
Yes
End

	Data		Border
New[*3]	B	W	
Input	N	W	N

6 Power off G2 COG Driver

1. Nothing Frame :
   Write a frame data

Panel size	Scan Line
1.44"	96
2"	96
2.7"	176

   Whose Data Bytes are 0x00. Scan Bytes operate normally.
   Scan lines are still turned on sequentially. This frame will make the image more uniform. Turn on OE SPI(0x02, 0x07) at the end of each line.
   For 1.44" & 2", need to set Border Byte to complete 1st Data Byte.
   Border Byte = 0x00 for 1.44" & 2"

2. Dummy Line :
   A line whose all Data Bytes are 0x00 and Scan Bytes are 0x00. Turning on OE SPI(0x02, 0x07) to complete this Dummy Line. Clear the register data before power off. Detail of data input is on page 26 ~ page 28.
   (This function is only used in 2.7")

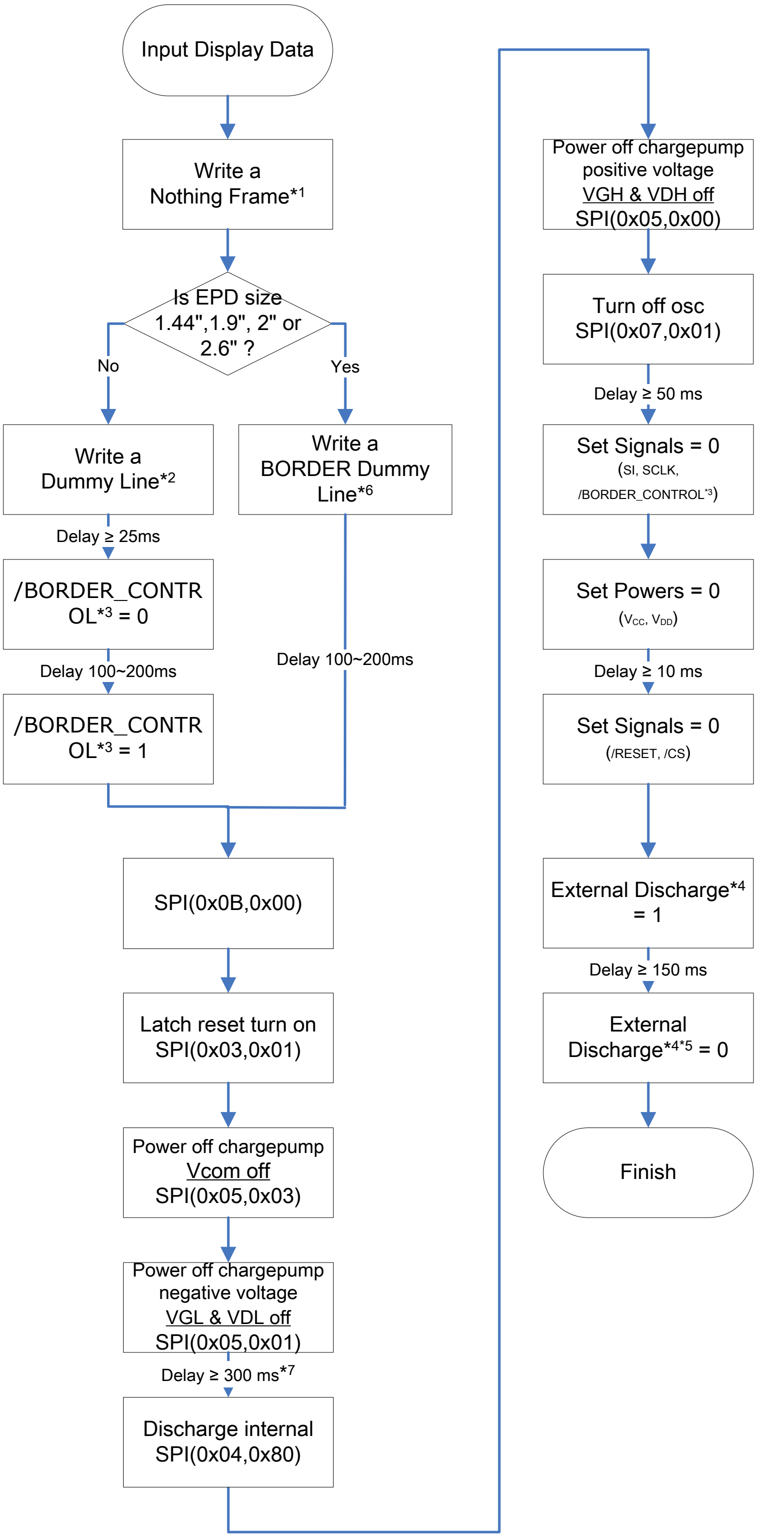
3. /BORDER_CONTROL :
   When = 0, the /BORDER_CONTROL is ON and write to white. When = 1,the /BORDER_CONTROL is OFF.
   (This function is only used in 2.7")

4. External Discharge :
   For implement this function, users need to use a DISCHARGE pin from microcontroller to control.(refer to the reference circuit)
   This is important to avoid vertical lines.

5. If you use the flash memory for pattern store, please recheck flash in this phase and verify the old image flash is erased.

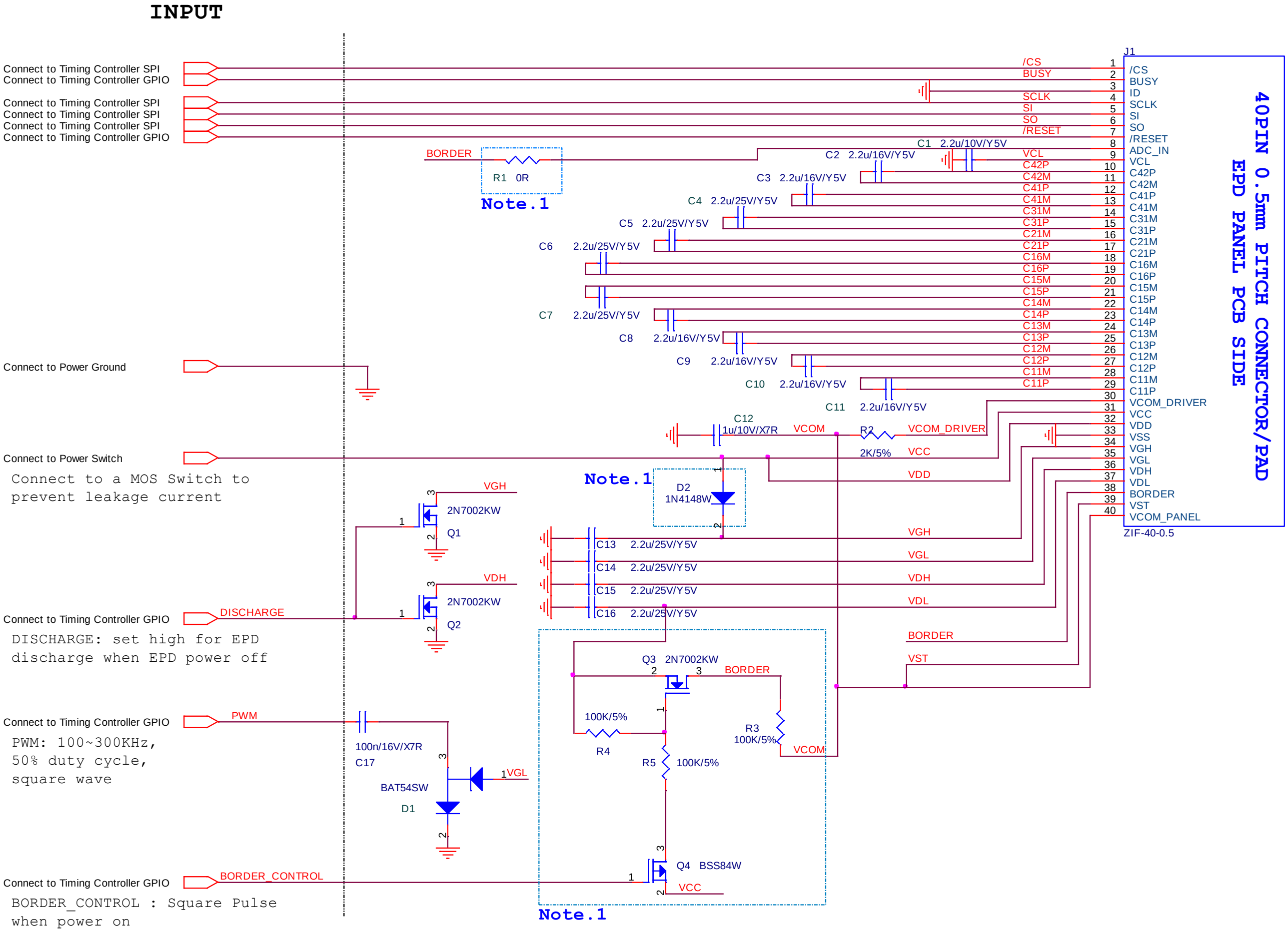
6. BORDER Dummy Line :
   Set Border Byte = 0xAA and write to white. A line whose all Data Bytes are 0x00 and Scan Bytes are 0x00. Then must to set SPI(0x02,0x07) in the end of line for turn on output enable of COG Driver to control border and clear the register data before power off. Detail of data input is on page 26 ~ page 28.
   (This function is only used in 1.44" & 2")

7. Delay time of VGL and VDL :
   Please ensure to place the accurate delay time here until VGL is discharged to GND by measured by oscilloscope.


Input Display Data
Write a Nothing Frame*1
Is EPD size 1.44",1.9", 2" or 2.6" ?
No
Yes
Write a Dummy Line*2
Delay ≥ 25ms
/BORDER_CONTROL*3 = 0
Delay 100~200ms
/BORDER_CONTROL*3 = 1
Write a BORDER Dummy Line*6
Delay 100~200ms
SPI(0x0B,0x00)
Latch reset turn on SPI(0x03,0x01)
Power off chargepump Vcom off SPI(0x05,0x03)
Power off chargepump negative voltage VGL & VDL off SPI(0x05,0x01)
Delay ≥ 300 ms*7
Discharge internal SPI(0x04,0x80)
Power off chargepump positive voltage VGH & VDH off SPI(0x05,0x00)
Turn off osc SPI(0x07,0x01)
Delay ≥ 50 ms
Set Signals = 0 (SI, SCLK, /BORDER_CONTROL*3)
Set Powers = 0 (VCC, VDD)
Delay ≥ 10 ms
Set Signals = 0 (/RESET, /CS)
External Discharge*4 = 1
Delay ≥ 150 ms
External Discharge*4*5 = 0
Finish

Appendix: Use G1's PCBA to drive the EPD with G2 Driver IC

Below is the reference circuit if you have arranged the PCBA that drove the EPD with G1 Driver IC already.



Note:

1. Hardware setting for different size:

	R1	Q3,Q4,R3,R4,R5	D2
1.44 inch & 2 inch	Mounted	No Mounted	Mounted
2.7 inch	No Mounted	Mounted	No Mounted

If users want to drive the EPD with G2 Driver IC by the current PCBA (i.e. the reference circuit above). Below items are the steps needed to do.

- Keep hardware unchanged as above.
  - ■ Keep Resistor R1 open.
  - ■ Keep BORDER CONTROL (Q3, Q4, R3, R4, and R5) circuit mounted.
- Modify SPI data as the following sections described.
- Disable the Timing Controller GPIO pin, PWM. Keep PWM signal as either 1 or 0.
- No matter what size EPD is, use same power off sequence of section 6.