WINSTAR_E-Paper Application Note

E-Paper

Version 2.0
Date: 2011/08/24

WINSTAR_E-Paper Application Note

WINSTAR_E-Paper Application Note

1. RECORD OF REVISION

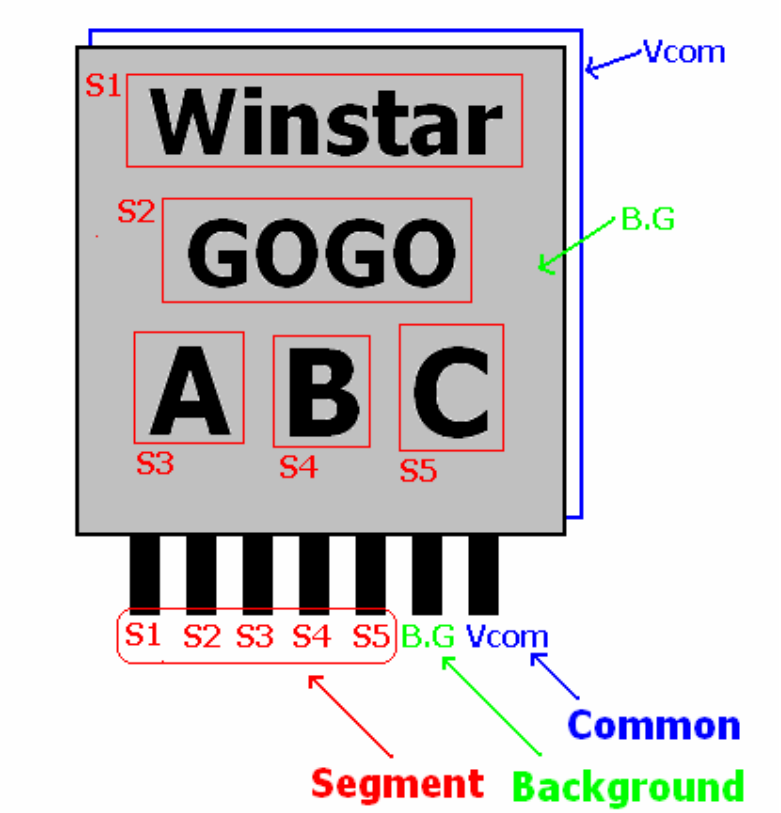
Revision Date	Page	Contents	Editor
2010/7/13	-	New Release	
2011/08/24	-	Modified Reference Circuit	
	-		

2. Features

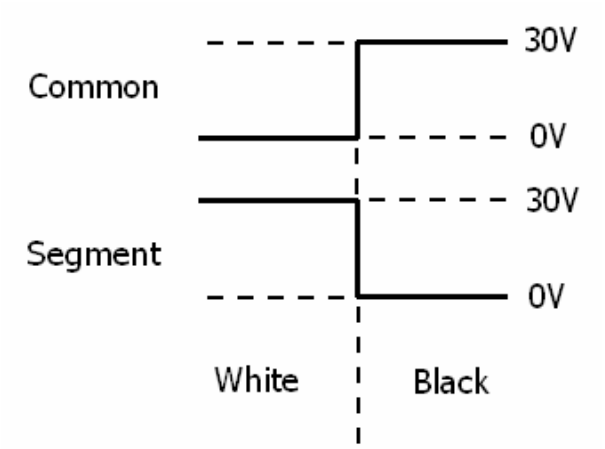
˙Segment output of 160 Channel, 4 voltage levels can be set.
˙Low resistance Segment output of 2 Channel, 4 voltage levels can be set and normally used by Common and Background.
˙ Each Channel voltage level output can be set to high resistance.
˙3 BIT parallel Interface
˙ Segment/Common driving voltage: 15V~40v (E-Paper driving voltage: 30V)
˙ Voltage range: 2.0V~5.5V
˙ IC operation temperature: -10˚C~75˚C (E-paper operation temperature: 0˚C~50˚C)
˙ IC package: Chip on FPC (COF)

3. Driving theory

Basically, E-paper driving pins are divided into three ones, as shown in below picture:



(1) Common: Transparent Conductor in E-Paper Film
(2) Background: A big ICON (segment) for background color of E -Paper and has the same Signal Driving as Segment.
(3) Segment: ICON for display contents of E-Paper
ICON can display black or white only when there is potential difference between Common and Segment.

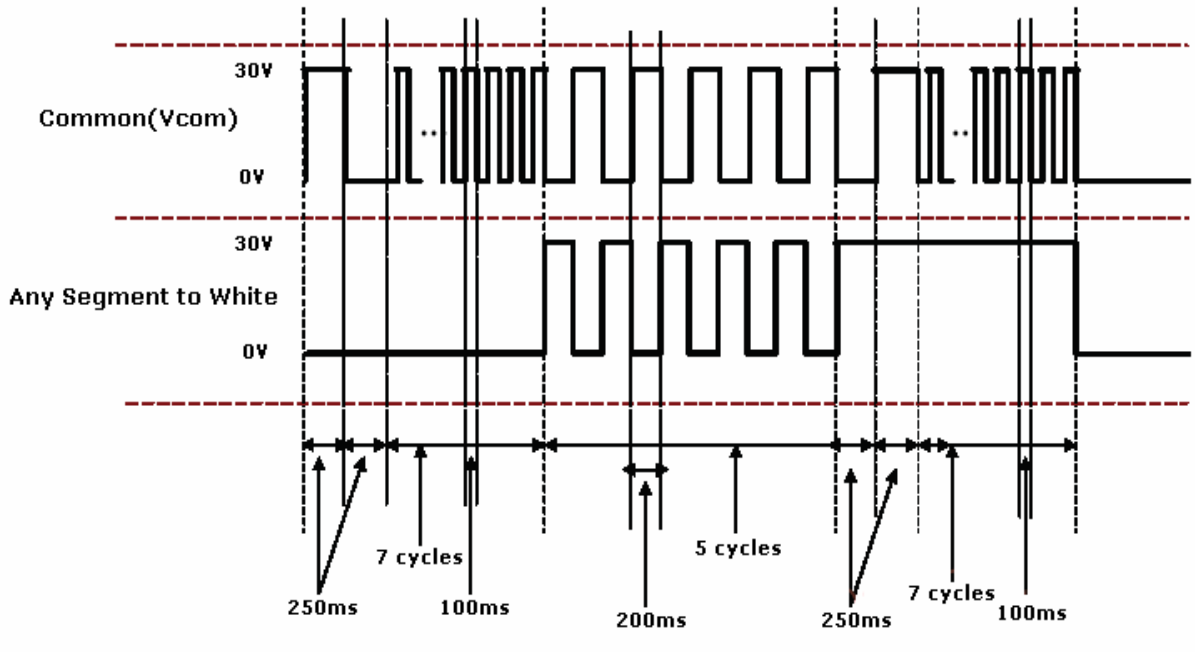


Waveform as shown in picture:
Display White：Common：0V Segment：30V.
Display black：Common：30V Segment：0V

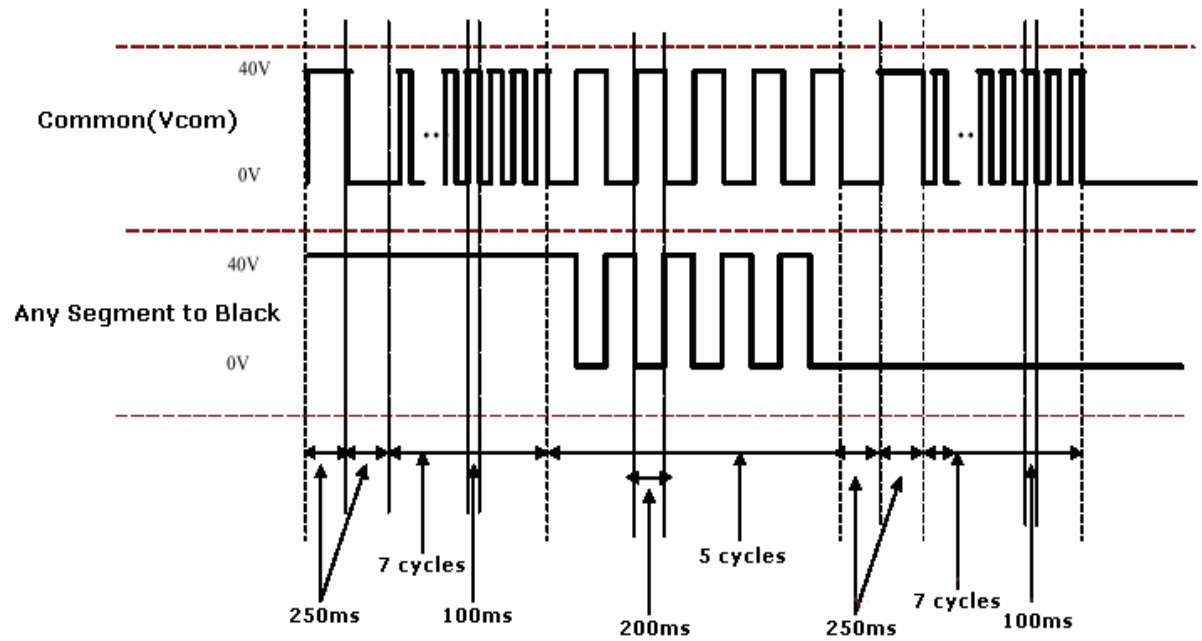
WINSTAR_E-Paper Application Note

One characteristic of E-Paper is that it can display normally if it is successfully drove, even though power supply is removed. But for long time normally displaying, specified waveform should be set for Common and Segment. Winstar found one best waveform which is called 7-5-7 waveform. It can keep displaying for about two months once drove, the waveform is shown as follows:

7-5-7 White Waveform



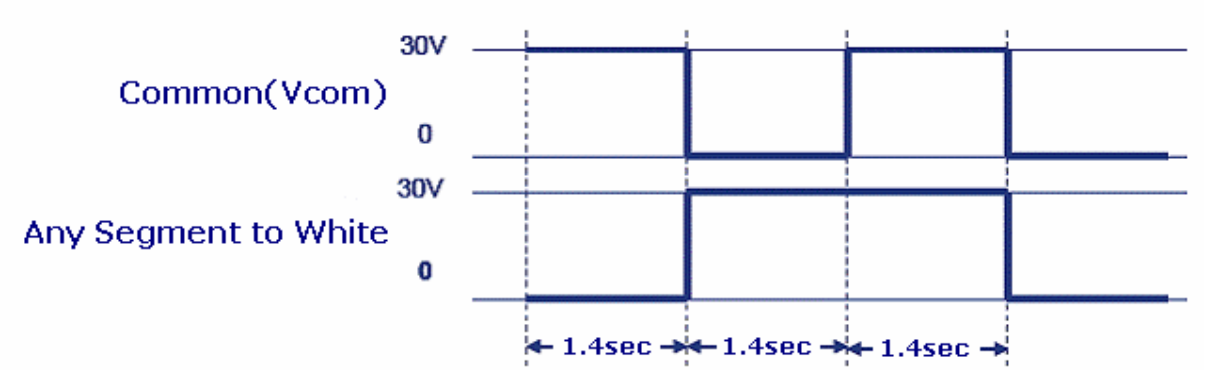
7-5-7 Black Waveform



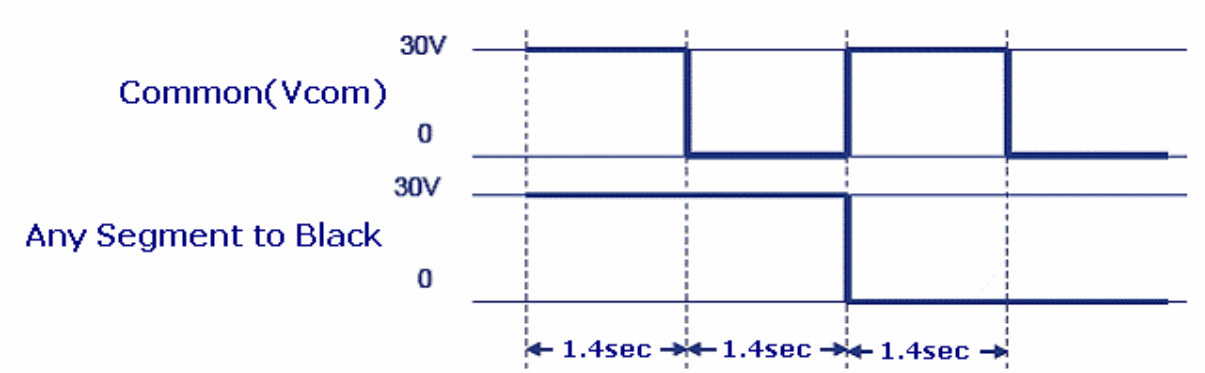
Common and Segment must be input at the same time and it can display for two months as above waveform shows.

As for the products which don't need to long time display but must update very often, Winstar can provide another waveform, it could display about 8 hours once waveform drove, the waveform is shown as follows:

Simple Waveform displays white:



Simple Waveform displays black:



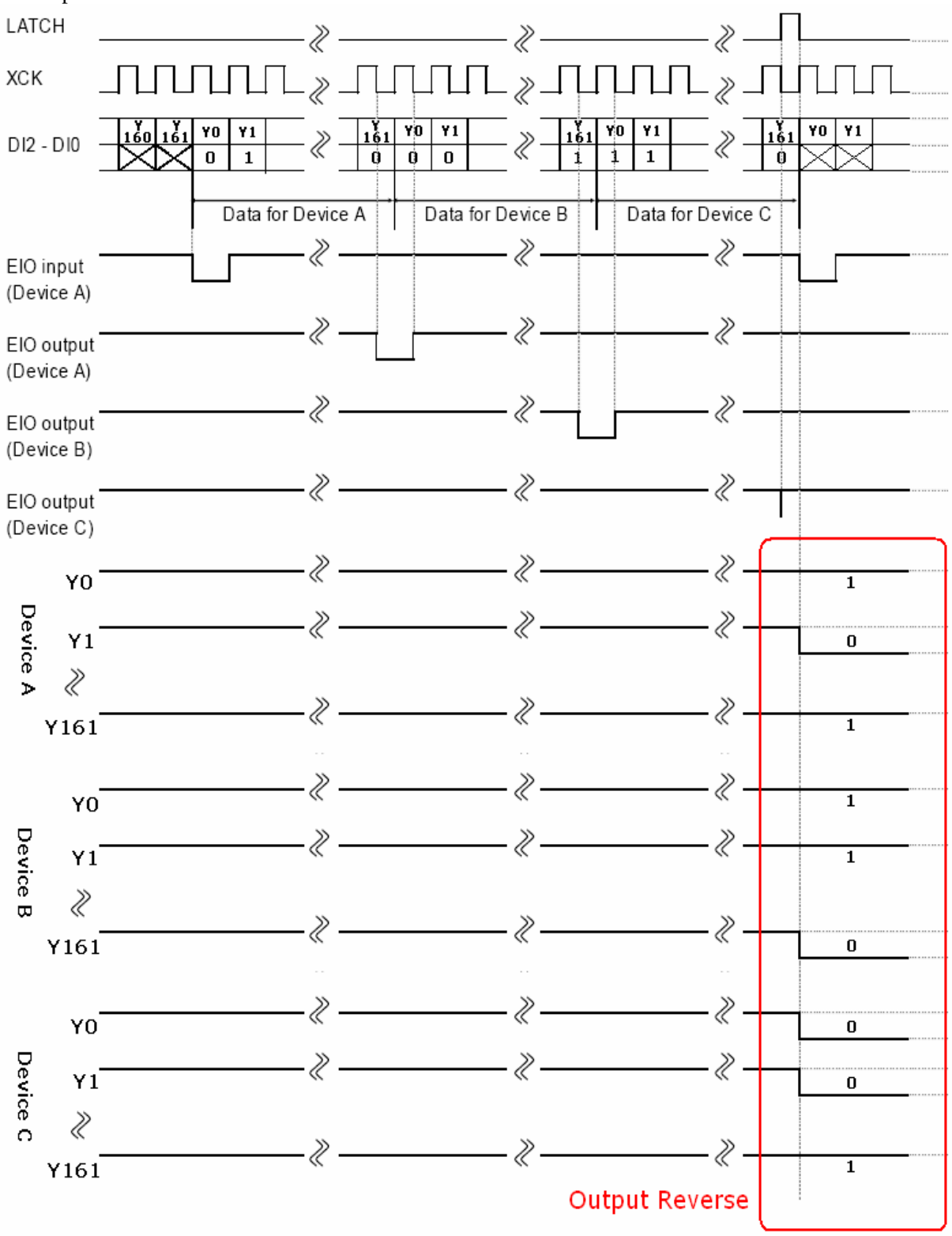
4. DSM04 Introduction

4.1 DSM040 Pin Function explanation:

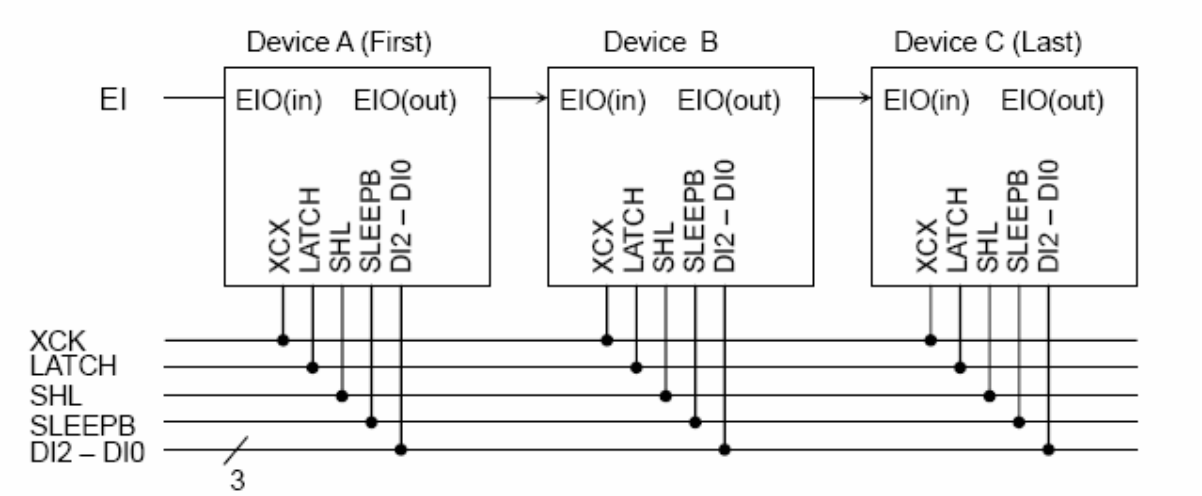
PIN Symbol	I/O	Functions
VDD	I	Output pin of voltage
VSS	I	Ground pin, must be 0V.
V0,V1,V2	I	Input voltage for output voltage level of Driver segment must follow V0>V2>V1>VSS，maximum output voltage is 40V.
EIO1,EIO2	I/O	Driver series connection input/Output pin, it is used when there are more than 2 Driver series connection. When SHL="L", EI01 series connection output, EI02 series connection input. When SHL="H", EI01 series connection input, EI02 series connection output. When LATCH="H" and XCK="L", ，EIO(OUTPUT)="H". After 162 XCK Pulse, EIO will output a negative edge pulse to the next Driver.
XCK	I	Input pin for display data, which can take the data input from DI2~DI0 to driver register.
DI2~DI0	I	Input pin for data input, display data is input from here and then take the display data to Driver register by negative edge signal of XCK pin.
SHL	I	Selection pin for shift direction of data When SHL="L", shift direction from Y161-Y0 When SHL="H", shift direction from Y0-Y161
LATCH	I	Control pin for display data output, negative-edge triggered. When LATCH changes from "H" to "L", the display data input from DI0~DI2 will output Y0~ Y161 from register.
SLEEPB	I	SLEEP Mode, when SLEEPB="L", it goes into sleep mode.
DLY1,DLY0	I	Timing selection pin for Driver output When DLY1, DLY0= [H, L], Output Timing is in automatic mode. When DLY1, DLY0= [H, H], Output Timing is controlled by LATCH, once LATCH triggered, Y0, Y1~Y10, Y11~Y20, Y21~Y30~ Y151~Y160, Y161 will be separately sent.
Y1~Y160	O	Driver Output pin for Segment
Y0,Y161	O	Driver Output pin for Common and Background

4.2 DSM040 Operation theory：

DSM040 can be regarded as a shift register which is series input and parallel output but Y0~Y161 voltage level must be set before. Timing theory and pin connections are shown in below picture:



If 3 Drivers are connected together(as shown in below pict ure), signal “L” should be input into EIO INPUT first and then Driver A will start to work, input data through D12~D10, when XCK is in negative edge, data will be taken to Driver register. While after 162 CLK for XCK, EIO OUTPUT will output a signal “L” to inform Driver B to work. When 3 Drivers are connected together, XCK will take the 162*3=486 data, after all information is taken, data in Driver inside can output to Driver Output when LATCH sends a signal “H”. 162 data should be taken by one Driver as ab ove mentioned.



3 Drivers connections

Voltage level setting for Driver Output (Y0~Y161)

This Driver provides input pins for V0, V1, and V2 voltage level. It can decide which voltage level should be output for Driver by the setting of DI2~DI0

Hold Memory Data			SLEEPB	Driver Output voltage level(Y0~Y161)
DI2	DI1	DI0		
0	X	X	H	High-Z High-Z(High resistance)
1	0	0	H	V0
1	0	1	H	V1
1	1	0	H	V2
1	1	1	H	VSS
X	X	X	L	VSS

Since E-Paper Flim used at present only requires 30V voltage, it will be ok only enter 30V into V0 and select **DI2~DI0=[1,0,0]→V0**，**DI2~DI0=[1,1,1]→VSS,** when SLEEPB is set to “L”, all output from Y0~Y16 is VSS. Nowadays, there are only 0V and 30V for EPAPER waveform level, so only VSS and V0 are useful.

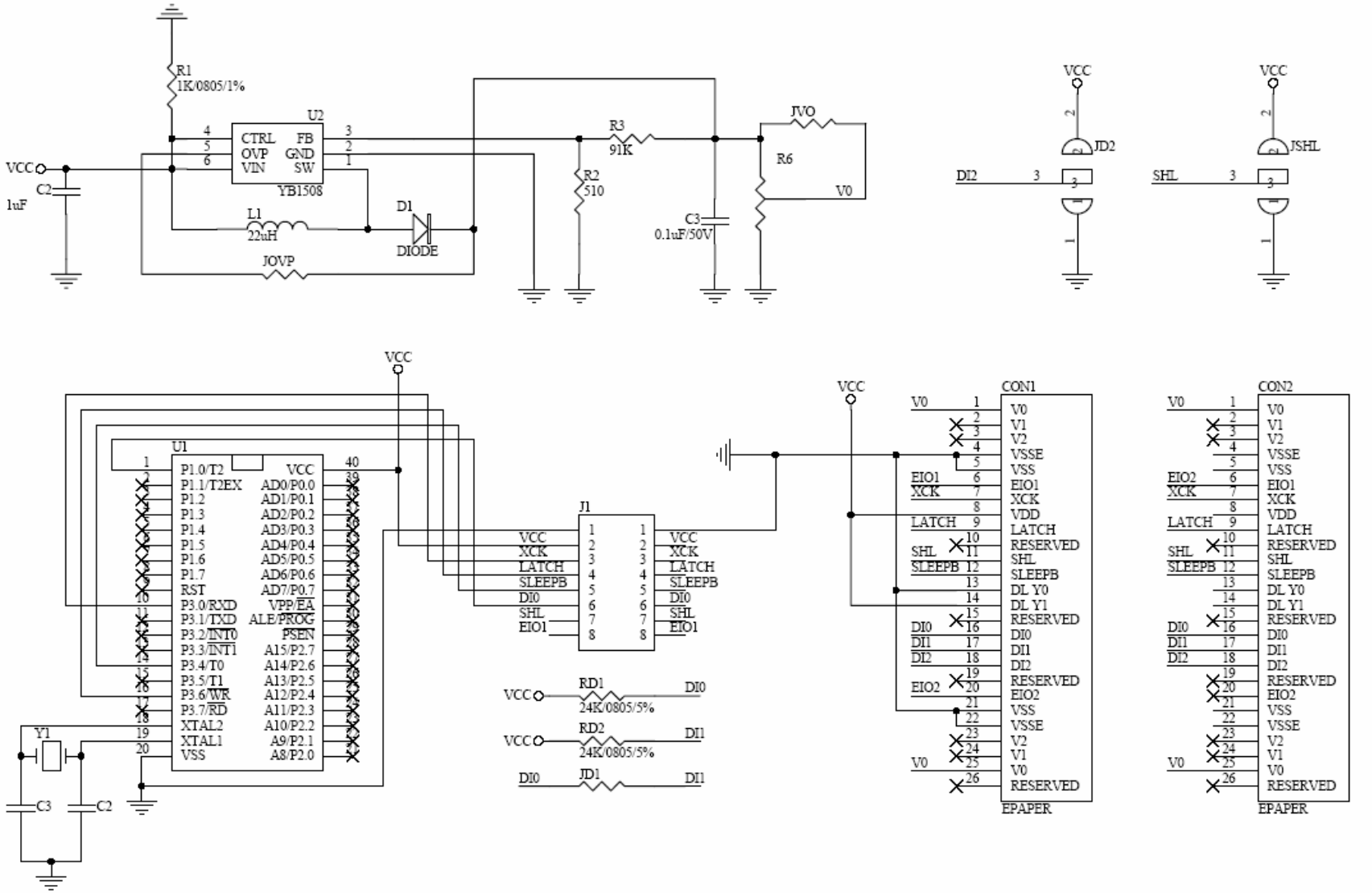
Setting for shift direction of data:

Shit direction of data is mainly set by SHL, as shown in below form.

SHL	EIO1	EIO2	XCK clock(Negative-Edge triggered)						
			1clock	2lock	3lock	……	160clock	161clock	162clock
L	Output	Input	Y161	Y160	Y159	……	Y2	Y1	Y0
H	Input	Output	Y0	Y1	Y2	……	Y159	Y160	Y161

WINSTAR_E-Paper Application Note

5. Application Circuit



6. Reference Initial code :

```
// Include files
#include <c8051f340.h>          // Header file for SiLabs c8051f34x
#include <INTRINS.H>                // KEIL FUNCTION


// for E-paper
sbit XCK=P3^0;    //RS
sbit LATCH=P3^7;          //RW
sbit SLEEPB=P3^4;         //E
sbit Din=P1^0;

// for E-paper
unsigned char    vcom = 0;
unsigned char    vseg = 0;

void SYSCLK_Init(void);
void main(void);
void delay_us(unsigned char);
void delay_ms(unsigned long);
void Write_data1();
void Write_data2();
void Write_data3();
void Write_data4();

unsigned char
temp[21]={0xff,0xff,0xff,0xff,0xa5,0xd1,0xa3,0xd1,0xbf,0x1f,0xb2,

        0xd3,0x4b,0xf4,0xbe,0x1f,0x69,0x74,0xff,0xff,0x80};


//==================================
void SYSCLK_Init (void)
{
        xdata int i = 0;
        CLKMUL        = 0x80;
        for (i = 0; i < 20; i++);       // Wait 5us for initialization
        CLKMUL        |= 0xC0;
        while ((CLKMUL & 0x20) == 0);
        OSCICN        = 0x83;
}

//==================================
void    WF757()
{
        SLEEPB = 0;
```

```
        delay_ms(100);

        vcom = 0;
        vseg = 0;
        XCK = 0;
        LATCH = 0;
        Din = 0;
        delay_ms(100);
        SLEEPB = 1;
        delay_ms(350);
        delay_ms(350);
        delay_ms(350);

}

//=====================================================
================
//    Write data to E-paper
//=====================================================
================
void WRITE_WF()                         //7-5-7 Waveform
{
      unsigned char i;
        Write_data1();
        delay_ms(260);    //250ms

        Write_data1();
        delay_ms(360);    //350ms


        for(i=0;i<13;i++)
        {

          Write_data1();
          delay_ms(100);    //100ms
        }


        for(i=0;i<10;i++)
        {
          Write_data2();
          delay_ms(210);
        }

      for(i=0;i<2;i++)

        {
```

```
        Write_data3();
        delay_ms(260);
        }

        for(i=0;i<14;i++)
        {

        Write_data3();
        delay_ms(100);
        }

        Write_data4();
        delay_ms(350);

        vcom=0;
        vseg=0;
  }

void Write_data1()
{
        unsigned char i,j,dsp_data;


        for(i=0;i<20;i++)
        {
          dsp_data=temp[i];
          for(j=0;j<8;j++)
                {

                        if(dsp_data&0x80)
                        Din=1;
                        else
                        Din=0;
                        dsp_data<<=1;
                        XCK=1;
                        nop_();
                        XCK=0;
                }
       }

          dsp_data=temp[i];
          nop_();

          if(dsp_data&0x80) Din=1;
          else Din=0;

          XCK=1;
```

```
        nop_();
        XCK=0;

        nop_();

        XCK=1;
        LATCH=1;
        Din=vcom;
      _nop_();
        vcom=!vcom;
        XCK=0;
      _nop_();
        LATCH=0;
      _nop_();
}

void Write_data2()
{
        unsigned char i,j;

        for(i=0;i<20;i++)
        {
              for(j=0;j<8;j++)
              {
                    XCK=1;
                    Din=vseg;
                    nop_();
                    XCK=0;
              }
      }
      for(i=20;i<21;i++)
      {
              XCK=1;
              Din=vseg;
              nop_();
              XCK=0;
              vseg=!vseg;
      }

      nop_();
      XCK=1;
      LATCH=1;
      Din=vcom;
      nop_();
      vcom=!vcom;

      XCK=0;
```

```
        nop_();
        LATCH=0;
}

void Write_data3()
{
        unsigned char i,j,dsp_data;


        for(i=0;i<20;i++)
        {
                dsp_data=~temp[i];
                nop_();

                for(j=0;j<8;j++)
                {

                        if(dsp_data&0x80)
                        Din=1;
                        else
                        Din=0;
                        dsp_data<<=1;
                        XCK=1;
                        nop_();
                        XCK=0;
                }
        }

        dsp_data=~temp[i];
        nop_();

        if(dsp_data&0x80) Din=1;
        else Din=0;
        XCK=1;
        nop_();
        XCK=0;


        nop_();

        XCK=1;
        LATCH=1;
        Din=vcom;
        nop_();
        vcom=!vcom;
        XCK=0;
        nop_();
```

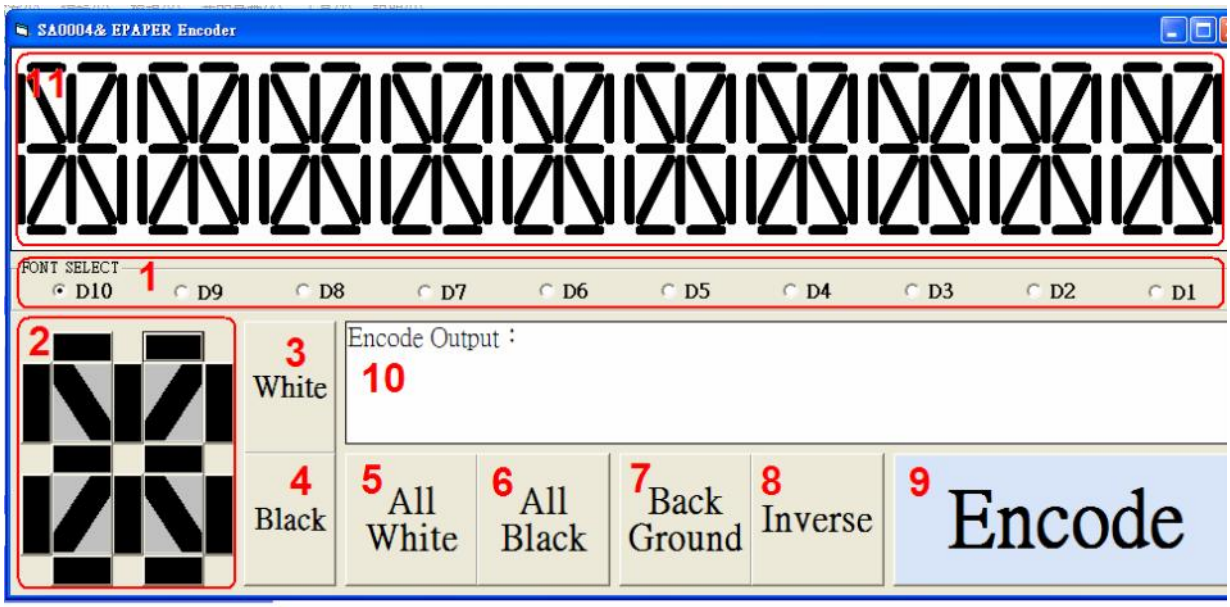
```
        LATCH=0;
}

void Write_data4()
{
        unsigned char i,j,dsp_data;

        for(i=0;i<20;i++)
        {
          dsp_data=0xFF;
          for(j=0;j<8;j++)
                {

                        if(dsp_data&0x80)
                        Din=1;
                        else
                        Din=0;
                        dsp_data<<=1;
                        XCK=1;
                        nop_();
                        XCK=0;
                }
      }

          dsp_data=0xFF;
          nop_();

          if(dsp_data&0x80) Din=1;
          else Din=0;
          XCK=1;
          nop_();
          XCK=0;


       _ nop_();

          XCK=1;
          LATCH=1;
          Din=vcom;
          nop_();
          vcom=!vcom;
          XCK=0;
         _nop_();
          LATCH=0;
         _nop_();
}
```

```
//=====================================================
================
//DELAY
//========================================== ==========
================
void delay_us(unsigned char us_value)        //delay about 10us
{
        while(us_value--)
        {
           _nop_();_nop_();_nop_();_nop_();
        }
}

void delay_ms(unsigned int ms_value)         //delay about 1ms
{
        unsigned int x;
        for(x=0;x<ms_value;x++)
        {
                delay_us(100);
        }
}
//-------------------------------------------------------------------------
// Main Routine
//-------------------------------------------------------------------------
void main(void)
{
  //    SYSCLK_Init ();
        WF757();
        delay_ms(100);
        WRITE_WF();

        while (1)
                {       }
}
```

7. Encoder software

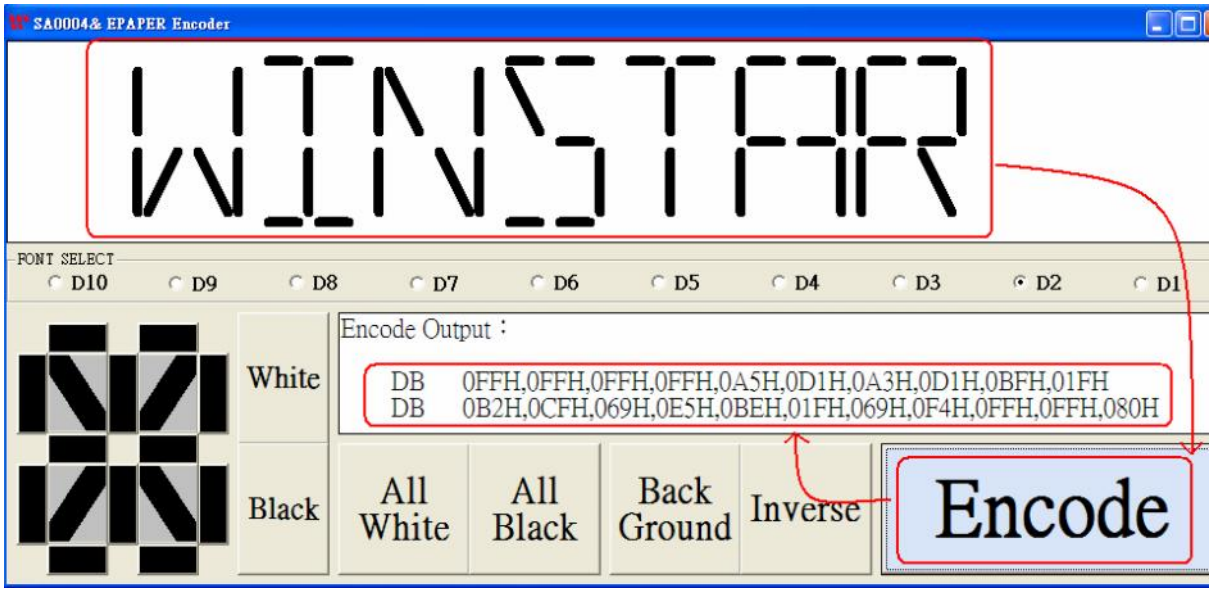
SA0004 & 16x10 EPAPER encoding software manual
First, Operate SA0004& EPAPER Encoder software

Second， Explanation of all functions of SA0004& EPAPER Encoder software
1：It shows the selects 16segment from D1~D10
2：Edit each 16segment
3：The whole 16segment display white



4：The whole 16segment display black
5：Full screen is white
6：Full screen is black
7：Edit the background, black and white on the exchange.
8：Screen displays inversed. (Black changes to white and white changes to black)
9：Encode
10：Encode output
11：EPAPER Edit screen

Third， explanation of Encode
After finishing editing screen, enter the “Encod e” and it will display “Code” in Encode Output.



Forth， paste the Encode output into the program.