a-Si TFT LCD Single Chip Driver with 132RGBx162 Resolution and 262K color

Application Notes

Version: V0.4
Date:Apr. 27 2017

New Vision Micro Inc.

Room 2208 Changhong Building,
China Academ of Science and technology
Development, Nanshan District,SZ

1 NV3021C-01 Chip ID Read

1.1 Chip ID Code

1.11 Chip ID Code 1 (R04h)

	W/R	RS	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Index	W	0	—	—	—	—	—	—	—	—	0	0	0	0	0	1	0	0
Read	R	1	—	—	—	—	—	—	—	—	1	1	0	1	0	1	0	0
Read	R	1	—	—	—	—	—	—	—	—	1	1	0	1	0	1	0	0
Read	R	1	—	—	—	—	—	—	—	—	1	0	0	0	0	0	0	0
Read	R	1	—	—	—	—	—	—	—	—	0	1	1	0	0	1	1	0

ID code for read the register “R04h” and the device data “D4”,“D4,“80”,“66”are read out.

1.12 Chip ID Code 2 (Rdah)

	W/R	RS	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Index	W	0	—	—	—	—	—	—	—	—	1	1	0	1	1	0	1	0
Read	R	1	—	—	—	—	—	—	—	—	1	1	0	1	0	1	0	0

ID code for read the register “Rdah” and the device data “D4”,“D4” are read out.

1.13 Chip ID Code 3 (Rdbh)

	W/R	RS	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Index	W	0	—	—	—	—	—	—	—	—	1	1	0	1	1	0	1	1
Read	R	1	—	—	—	—	—	—	—	—	1	0	0	0	0	0	0	0
Read	R	1	—	—	—	—	—	—	—	—	1	0	0	0	0	0	0	0

ID code for read the register “Rdbh” and the device data “80”,“80” are read out.

1.14 Chip ID Code 4 (Rdch)

	W/R	RS	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Index	W	0	—	—	—	—	—	—	—	—	1	1	0	1	1	1	0	0
Read	R	1	—	—	—	—	—	—	—	—	0	1	1	0	0	1	1	0
Read	R	1	—	—	—	—	—	—	—	—	0	1	1	0	0	1	1	0

ID code for read the register “Rdch” and the device data “66”,“66” are read out.

1.2 Chip ID Program For Reference

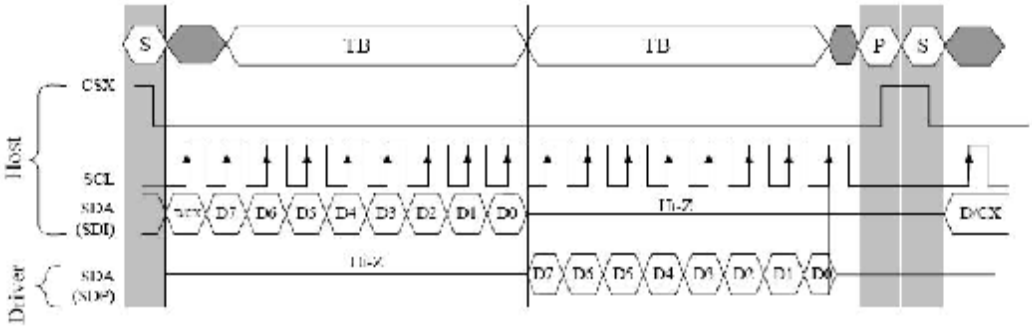
1.21 Program for I80-8bits

```
unsigned char    RData()
{
      unsigned char i;
        P0=0xff;
        cs=0;
        rs=1;
        wr=1;
        rd=1;
        rd=0;
        i=P0;
        rd=1;
        cs=1;
        return i;
}

void main(void)
{
    char rrdata;
        RST=1;
    delayms(20);
        RST=0;
    delayms(50);
        RST=1;
    delayms(50);
    comm_out(0x04);//Read “R04h”
    rrdata=RData(); //0Xd4
    rrdata=RData(); //0xd4
```

```
    rrdata=RData(); //0x80
    rrdata=RData(); //0x66

    comm_out(0xda);//Read “Rdah”
    rrdata=RData(); //0xd4
    rrdata=RData(); //0xd4

    comm_out(0xdb);//Read “Rdbh”
    rrdata=RData(); //0x80
    rrdata=RData(); //0x80

    comm_out(0xdc);//Read “Rdch”
    rrdata=RData(); //0x66
    rrdata=RData(); //0x66

}
```

1.22 Program for SPI

1.221 Timing for 3-SPI



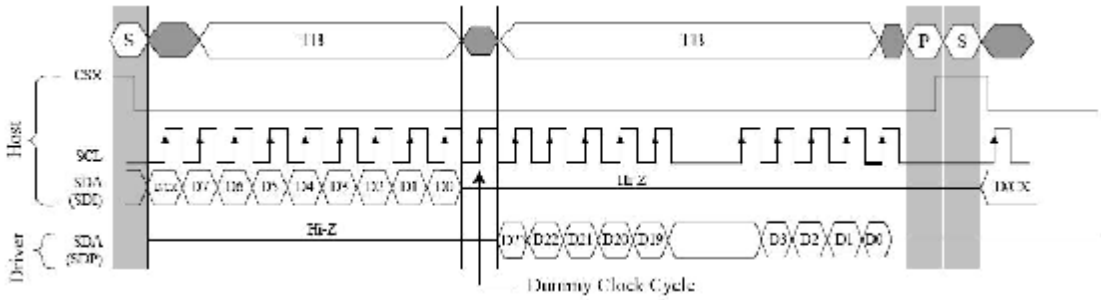
8-bit read for dah, dbh, dch

1.222 Program for 3-SPI (8-bit)

```
void comm_out_rspi(unsigned char data) // 3-SPI  读操作的写指令函数
    {
        unsigned char i = 8;
        cs=0;                                   //片选低有效
        sck=0;
        sda=0;                              // D/C=0，为写指令操作
        sck=1;                              // SCK  上升沿采数据
        sck=0;
```

```
        while(i--)
        {
         if(data&128)
           {
                  sck=0;
                  sda=1;
                  sck=1;          // SCK  上升沿采数据
                  sck=0;
           }
         else
           {
                sck=0;
                sda=0;
                sck=1;      // SCK  上升沿采数据
                sck=0;
                   }
                   data=data<<1; //由高位到低位，一位一位的取出  data
           }
              cs=0;       //不能置  1  或删除不操作。
}

unsigned char read_8()                  //8-bit read for 3-SPI
{
    unsignedchar i,b,c=0,d=0;
     i=8;
     sda=1;                                          //清数据端口
     cs=0;                                           //片选置低有效
     while(i--)
     {
          sck=0;
          sck=1;
          b=sda;                                  //从  SDA  中取出读出的数据存在  b
          d=(0xff&(b<<i));
          c=(c|d);                                //读出的值拼成  8  位的数据
     }
          sck=0;
          cs=1;
          return c;                               //返回读出的  8  位的  ID  数据
}
```

1.223 Timing for 3-SPI (24-bit)



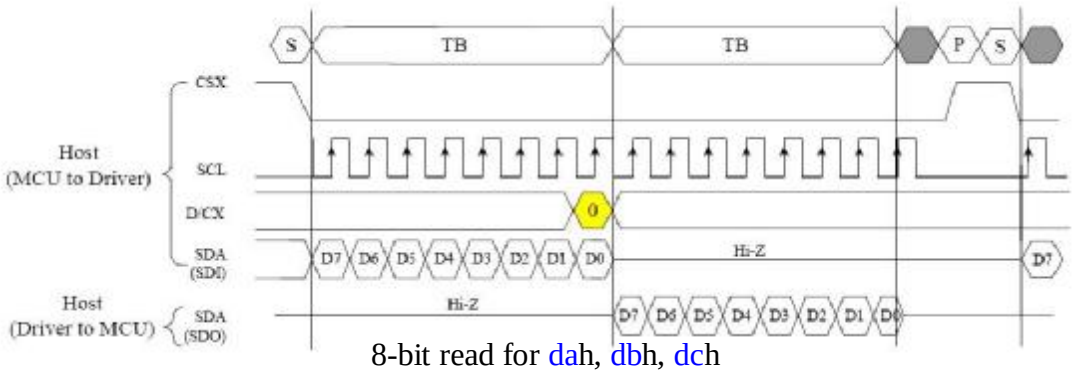
24-bit read for 04h

1.224 Program for 3-SPI (24-bit)

```
unsigned char read_24()       // 24-bit read for 3-SPI
{
    unsigned    char i,b,d=0;
    sda=1;                                    //清数据端口
    cs=0;                                     //片选置低有效

    sck=0;                                    //DUMMY 时钟
    sck=1;                                    //此 DUMMY 不能少

    i=8;
    while(i--)
   {
    sck=0;
    sck=1;
    b=sda;                    //从 SDA 中取出读出的数据存在 b 中
    d=(0xff&(b<<i));
    c1=(c1|d);   //读出的值拼成 8 位的数据，存在全局变量 C1 中
    }

     i=8;
    b=d=0;                            //继续读后面的 8 位 ID 数据
    while(i--)
   {
    sck=0;
    sck=1;
    b=sda;                    //从 SDA 中取出读出的数据存在 b 中
    d=(0xff&(b<<i));
    c2=(c2|d);   //读出的值拼成 8 位的数据，存在全局变量 C2 中
    }
```

```
    i=8;
    b=d=0;                          //再继续读剩余的最后 8 位 ID 的数据
    while(i--)
   {
    sck=0;
    sck=1;
    b=sda;                  //从 SDA 中取出读出的数据存在 b 中
    d=(0xff&(b<<i));
    c3=(c3|d);   //读出的值拼成 8 位的数据，存在全局变量 C3 中
    }

     sck=0;
     cs=1;                  //片选置高
}
```

1.225 Timing for 4-SPI (8-bit)



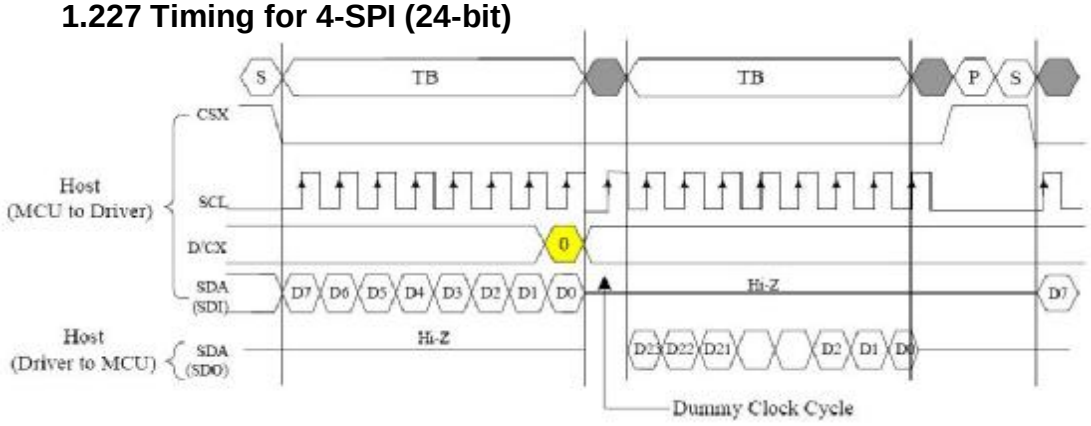
8-bit read for dah, dbh, dch

1.226 Program for 4-SPI (8-bit)

```
void comm_out_rspi(unsigned char data) // 4-SPI 读操作的写指令函数
  {
     unsigned char i = 8;
     cs=0;                                          //片选低有效
     rs=0;                                          //rs=0 为指令
     while(i--)
     {
      if(data&128)
        {
          sck=0;
          sda=1;
          sck=1;          // SCK 上升沿采数据
          sck=0;
```

```
        }
      else
        {
            sck=0;
            sda=0;
            sck=1;     // SCK 上升沿采数据
            sck=0;
          }
          data=data<<1; //由高位到低位，一位一位的取出 data
      }
        cs=0;      //不能置 1 或删除不操作。
}

unsigned char read_8()          //8-bit read for 4-SPI， 同 3-SPI
{
    unsigned   char i,b,c=0,d=0;
    i=8;
    sda=1;                  //清数据端口
    cs=0;                   //片选置低有效
    while(i--)
    {
        sck=0;
        sck=1;
        b=sda;                  //从 SDA 中取出读出的数据存在 b 中
        d=(0xff&(b<<i));
        c=(c|d);                //读出的值拼成 8 位的数据
    }
        sck=0;
        cs=1;
        return c;               //返回读出的 8 位的 ID 数据
}
```

1.227 Timing for 4-SPI (24-bit)



24-bit read for 04h

1.228 Program for 4-SPI (24-bit)

```
unsigned char read_24()       // 24-bit read for 4-SPI, 同 3-SPI
{
    unsigned    char i,b,d=0;
    sda=1;                          //清数据端口
    cs=0;                           //片选置低有效

    sck=0;                         //DUMMY 时钟
    sck=1;                         //此 DUMMY 不能少

    i=8;
    while(i--)
   {
    sck=0;
    sck=1;
    b=sda;                      //从 SDA 中取出读出的数据存在 b 中
    d=(0xff&(b<<i));
    c1=(c1|d);                 //读出的值拼成 8 位的数据，存在全局变量 C1 中
    }

     i=8;
    b=d=0;                          //继续读后面的 8 位 ID 数据
    while(i--)
   {
    sck=0;
    sck=1;
    b=sda;                      //从 SDA 中取出读出的数据存在 b 中
    d=(0xff&(b<<i));
    c2=(c2|d);                 //读出的值拼成 8 位的数据，存在全局变量 C2 中
    }

    i=8;
    b=d=0;                          //再继续读剩余的最后 8 位 ID 的数据
    while(i--)
   {
    sck=0;
    sck=1;
    b=sda;                      //从 SDA 中取出读出的数据存在 b 中
    d=(0xff&(b<<i));
    c3=(c3|d);                 //读出的值拼成 8 位的数据，存在全局变量 C3 中
    }
     sck=0;
```

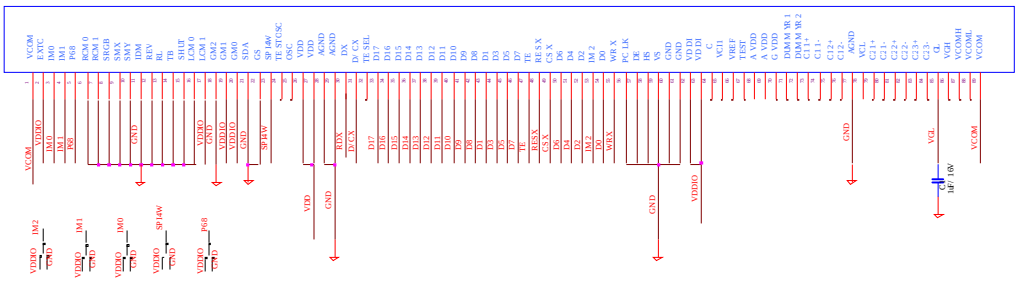
```
    cs=1;                //片选置高
}
```

1.229 Reading for SPI

```
uchar c1,c2,c3;              // For SPI   04h 24bits read

 void main(void)
{
char rrdata;
RST=1;
delayms(20);
RST=0;
delayms(50);
RST=1;
delayms(50);
comm_out_rspi (0xda);//Read “Rdah”
rrdata=RData(); //0xd4
comm_out_rspi (0xdb);//Read “Rdbh”
rrdata=RData(); //0x80
comm_out_rspi (0xdc);//Read “Rdch”
rrdata=RData(); //0x66

comm_out_rspi (0x04);//Read “R04h”
        // c1=0xd4; c2=0x80; c3=0x66
}
```

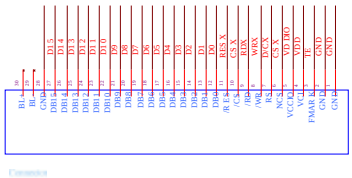
2. CMO1.77 inch Panel

2.1 Application FPC Circuit



TESEL 要悬空	P68=1 6800	P68=0 8080
	IM2=1 Parallel	IM2=0 Serial

IM1	IM0	Parallel interface
0	0	8-bit
0	1	16-bit
1	0	9-bit
1	1	18-bit



GM2	GM1	GM0	分辨率
0	0	0	132RGB*162
0	0	1	128RGB*128
0	1	0	120RGB*160
0	1	1	128RGB*160
1	0	0	130RGB*130
1	0	1	132RGB*132

IM2	IM1	IM0	SPI4W	接口总线	
1	0	0	*	并行 8-bit	(D[7:0])
1	0	1	*	并行16-bit	(D[15:0])
1	1	0	*	并行9-bit	(D[8:0])
1	1	1	*	并行18-bit	(D[17:0])
0	*	*	0	串行3-SPI:	SCL(DCX); SDIO(D0)
0	*	*	1	串行4-SPI:	SCL(DCX); SDIO(D0); RS(WRX)

2.2 CMO 1.77 inch initial code

```
initial_CMO1.77()

    {

  // VCI=2.8V
  //---------- Reset LCD Driver ----------//
      RST = 1;
      delayms(10); // Delay 1ms
      RST = 0;



      delayms(100); // Delay 10ms // This delay time is necessary
      RST = 1;
      delayms(50); // Delay 50 ms
```

```
comm_out(0xb4);
data_out(0x07);

comm_out(0xb1);
data_out(0x1c);

//--------------------INTER REG_EN--------------------------
comm_out(0xff);
data_out(0xa5);


comm_out(0xec);
data_out(0x89);

comm_out(0xed);
data_out(0x25);

comm_out(0xee);
data_out(0x22);

comm_out(0xf6);
data_out(0x10);

comm_out(0xc4);
data_out(0x10);

comm_out(0xc5);
data_out(0x12);

comm_out(0xc6);
data_out(0x0a);

comm_out(0xe0);
data_out(0x00);

comm_out(0xe8);
data_out(0x77);

comm_out(0xe1);
data_out(0x55);

comm_out(0xe7);
data_out(0x22);
```

```
    comm_out(0xe2);
    data_out(0x07);

    comm_out(0xe6);
   data_out(0x07);

    comm_out(0xe3);
    data_out(0x52); //

    comm_out(0xe9);
    data_out(0x25); //

    comm_out(0xe4);
   data_out(0x05; //

    comm_out(0xeb);
   data_out(0x00); //

    comm_out(0xe5);
   data_out(0x01); //

    comm_out(0xea);
   data_out(0x00; //

    comm_out(0x11);
    delayms(10);

    comm_out(0x3a);
    data_out(0x05);

    comm_out(0x36);
    data_out(0xdc);
    delayms(1);

    comm_out(0x29);

}
```

```
void setDispArea(unsigned int s_col,e_col,s_page,e_page)
{
   comm_out(0x2a);         // Set col address
   data_out((s_col &0x0100)>>8);
   data_out(s_col &0x00ff); // Start col address
   data_out((e_col &0x0100)>>8);
   data_out(e_col &0x00ff); // End col address
   comm_out(0x2b);         // Set page address
   data_out((s_page &0x0100)>>8);
   data_out(s_page &0x00ff); // Start col address
   data_out((e_page &0x0100)>>8);
   data_out(e_page &0x00ff); // End col address
   comm_out(0x2c);         // Write Display Data
}

void sleepin (void)
{
      comm_out(0x28);
      delayms(10);
      comm_out(0x10);
      delayms(100);
}

void sleepout (void)
{
      comm_out(0x11);
      delayms(120);
      comm_out(0x29);
      delayms(10);
}
```

3. 来宝1.77 inch Panel

3.1 来宝 1.77 inch initial code

```
initial_来宝 1.77()

{

// VCI=2.8V
//---------- Reset LCD Driver ----------//
RST = 1;
delayms(1); // Delay 1ms
RST = 0;
delayms(10); // Delay 10ms // This delay time is necessary
RST = 1;
delayms(50); // Delay 50 ms


comm_out(0xb4);
data_out(0x07);

comm_out(0xb1);
data_out(0x18);

//----------INTER REG_EN----------
comm_out(0xff);
data_out(0xa5);



comm_out(0xec);
data_out(0x89);

comm_out(0xed);
data_out(0x25);

comm_out(0xee);
data_out(0x22);

comm_out(0xf6);
data_out(0x10);

comm_out(0xc4);
data_out(0x0d);

comm_out(0xc5);
data_out(0x46);

comm_out(0xc6);
data_out(0x1f);  //0e
```

```
comm_out(0xc7);
data_out(0x11);

comm_out(0xe0);    //Gamma
data_out(0x70);

comm_out(0xe8);
data_out(0x07);

comm_out(0xe1);
data_out(0x72);//40

comm_out(0xe7);
data_out(0x02);//04

comm_out(0xe2);
data_out(0x77); //07//77

comm_out(0xe6);
data_out(0x77); //70//77

comm_out(0xe3);
data_out(0x17); //14

comm_out(0xe9);
data_out(0x07); //15

comm_out(0xe4);
data_out(0x00); //02

comm_out(0xeb);
data_out(0x00); //02

comm_out(0xe5);
data_out(0x01); //01

comm_out(0xea);
data_out(0x01); //01

comm_out(0x11);
delayms(10);
comm_out(0x3a);
data_out(0x05);

comm_out(0x36);
data_out(0xdc);

delayms(1);
comm_out(0x29);
}
```

```
void setDispArea(unsigned int s_col,e_col,s_page,e_page)
{
   comm_out(0x2a);         // Set col address
   data_out((s_col &0x0100)>>8);
   data_out(s_col &0x00ff); // Start col address
   data_out((e_col &0x0100)>>8);
   data_out(e_col &0x00ff); // End col address
   comm_out(0x2b);         // Set page address
   data_out((s_page &0x0100)>>8);
   data_out(s_page &0x00ff); // Start col address
   data_out((e_page &0x0100)>>8);
   data_out(e_page &0x00ff); // End col address
   comm_out(0x2c);         // Write Display Data
}

void sleepin (void)
{
      comm_out(0x28);
      delayms(10);
      comm_out(0x10);
      delayms(120);
}

void sleepout (void)
{
      comm_out(0x11);
      delayms(120);
      comm_out(0x29);
      delayms(10);
}
```

4. IVO1.77 inch Panel

4.1 IVO 1.77 inch initial code

```
initial_IVO 1.77()

{

// VCI=2.8V
//---------- Reset LCD Driver ----------//
RST = 1;
delayms(1); // Delay 1ms
RST = 0;
delayms(10); // Delay 10ms // This delay time is necessary
RST = 1;
delayms(50); // Delay 50 ms


comm_out(0xb4);
data_out(0x07);

comm_out(0xb1);
data_out(0x18);

//----------INTER REG_EN----------
comm_out(0xff);
data_out(0xa5);



comm_out(0xec);
data_out(0x89);

comm_out(0xed);
data_out(0x25);

comm_out(0xee);
data_out(0x22);

comm_out(0xf6);
data_out(0x10);

comm_out(0xc4);
data_out(0x0d);

comm_out(0xc5);
data_out(0x46);

comm_out(0xc6);
data_out(0x1f);    //0e
```

```
comm_out(0xc7);
data_out(0x47);

comm_out(0xe0);         // Gamma
data_out(0x70);//70

comm_out(0xe8);
data_out(0x07);//07

comm_out(0xe1);
data_out(0x32);//40 72

comm_out(0xe7);
data_out(0x03);//04

comm_out(0xe2);
data_out(0x77); //07//77

comm_out(0xe6);
data_out(0x77); //70//77

comm_out(0xe3);
data_out(0x17); //14//07

comm_out(0xe9);
data_out(0x77); //15

comm_out(0xe4);
data_out(0x00); //02

comm_out(0xeb);
data_out(0x05); //02

comm_out(0xe5);
data_out(0x00); //01

comm_out(0xea);
data_out(0x00); //01

comm_out(0x11);
delayms(10);

comm_out(0x3a);
data_out(0x05);

comm_out(0x36);
data_out(0xdc);

delayms(1);
comm_out(0x29);
}
```

```
void setDispArea(unsigned int s_col,e_col,s_page,e_page)
{
   comm_out(0x2a);         // Set col address
   data_out((s_col &0x0100)>>8);
   data_out(s_col &0x00ff); // Start col address
   data_out((e_col &0x0100)>>8);
   data_out(e_col &0x00ff); // End col address
   comm_out(0x2b);         // Set page address
   data_out((s_page &0x0100)>>8);
   data_out(s_page &0x00ff); // Start col address
   data_out((e_page &0x0100)>>8);
   data_out(e_page &0x00ff); // End col address
   comm_out(0x2c);         // Write Display Data
}

void sleepin (void)
{
      comm_out(0x28);
      delayms(10);
      comm_out(0x10);
      delayms(120);
}

void sleepout (void)
{
      comm_out(0x11);
      delayms(120);
      comm_out(0x29);
      delayms(10);
}
```

5. CPT 1.77 inch Panel

5.1 CPT 1.77 inch initial code

```
void NV3021_CPT177_Initial(void)
{
//-------------------- Reset LCD Driver ------------------//
comm_out(0xb4);
data_out(0x07);
comm_out(0xb1);
data_out(0x1c);
//--------------------INTER REG_EN----------------------------

comm_out(0xff);
data_out(0xa5);
comm_out(0xec);
data_out(0x89);
comm_out(0xed);
data_out(0x25);
comm_out(0xee);
data_out(0x22);
comm_out(0xf6);
data_out(0x10);
comm_out(0xc4);
data_out(0x0e);
comm_out(0xc5);
data_out(0x32);
comm_out(0xc6);
data_out(0x0f); //0e
//comm_out(0xc7);
//data_out(0x03);

comm_out(0xe0);
data_out(0x00);
comm_out(0xe8);
data_out(0x77);
comm_out(0xe1);
data_out(0x55);
comm_out(0xe7);
data_out(0x22);
comm_out(0xe2);
data_out(0x07);
comm_out(0xe6);
data_out(0x07);
comm_out(0xe3);
data_out(0x52);
comm_out(0xe9);
data_out(0x25);
comm_out(0xe4);
data_out(0x00);
```

```
comm_out(0xeb);
data_out(0x00);
comm_out(0xe5);
data_out(0x00);
comm_out(0xea);
data_out(0x00);
comm_out(0x11);
Delay_ms(100);
comm_out(0x3a);
data_out(0x05);
comm_out(0x36);
data_out(0x88);
comm_out(0x29);


}

        void setDispArea(unsigned int s_col,e_col,s_page,e_page)
        {
          comm_out(0x2a);         // Set col address
          data_out((s_col &0x0100)>>8);
          data_out(s_col &0x00ff); // Start col address
          data_out((e_col &0x0100)>>8);
          data_out(e_col &0x00ff); // End col address
          comm_out(0x2b);         // Set page address
          data_out((s_page &0x0100)>>8);
          data_out(s_page &0x00ff); // Start col address
          data_out((e_page &0x0100)>>8);
          data_out(e_page &0x00ff); // End col address
          comm_out(0x2c);         // Write Display Data
        }

        void sleepin (void)
        {
              comm_out(0x28);
              delayms(10);
              comm_out(0x10);
              delayms(120);
        }

        void sleepout (void)
        {
              comm_out(0x11);
              delayms(120);
              comm_out(0x29);
              delayms(10);
        }
```

6. HSD 1.77 inch Panel

6.1 hsd 1.77 inch initial code

```
Initial_hsd1.77();
{
comm_out(0xb4);
data_out(0x07);    //d2=1 frame inversion    07
comm_out(0xb1);
data_out(0x1C);    //frame rate    1c
//--------------------INTER REG_EN----------------------------
comm_out(0xff);
data_out(0xa5);

comm_out(0xec);
data_out(0x89);

comm_out(0xed);
data_out(0x15);    //25
comm_out(0xee);
data_out(0x12);    //22
comm_out(0xf6);
data_out(0x10);
comm_out(0xc4);
data_out(0x1a); //VRH
comm_out(0xc5);
data_out(0x35); //vcom    33 36
comm_out(0xc6);
data_out(0x0a); //vdv

comm_out(0xe0); //Gamma
data_out(0x00);
comm_out(0xe8);
data_out(0x77);
comm_out(0xe1);
data_out(0x55);
comm_out(0xe7);
data_out(0x22);
comm_out(0xe2);
data_out(0x07);
comm_out(0xe6);
data_out(0x07);
comm_out(0xe3);
data_out(0x52);
comm_out(0xe9);
data_out(0x25);
comm_out(0xe4);
data_out(0x00);
comm_out(0xeb);
data_out(0x00);
comm_out(0xe5);
```

```
data_out(0x00);
comm_out(0xea);
data_out(0x00);    //Gamma

comm_out(0x11);
delayms(120);
comm_out(0x3a);
data_out(0x05);
comm_out(0x36);
data_out(0x98);
comm_out(0x29);
}

void setDispArea(unsigned int s_col,e_col,s_page,e_page)
{
   comm_out(0x2a);          // Set col address
   data_out((s_col &0x0100)>>8);
   data_out(s_col &0x00ff); // Start col address
   data_out((e_col &0x0100)>>8);
   data_out(e_col &0x00ff); // End col address
   comm_out(0x2b);          // Set page address
   data_out((s_page &0x0100)>>8);
   data_out(s_page &0x00ff); // Start col address
   data_out((e_page &0x0100)>>8);
   data_out(e_page &0x00ff); // End col address

   comm_out(0x2c);          // Write Display Data
}

void sleepin (void)
{
      comm_out(0x28);
      delayms(10);
      comm_out(0x10);
      delayms(120);
}
void sleepout (void)
{
      comm_out(0x11);
      delayms(120);
      comm_out(0x29);
      delayms(10);
}
```

7. TM 1.77 inch Panel

7.1 tm 1.77 inch initial code

```
Initial_tm1.77()
   {

  comm_out(0xb4);
  data_out(0x07);         //frame Inversion
  // data_out(0x00);         //line Inversion



  comm_out(0xb1);
  data_out(0x18);        //120hz

  comm_out(0xff);
  data_out(0xa5);        //inter_en


comm_out(0xec);
data_out(0x89);
  comm_out(0xed);
  data_out(0x25);        //chp


  comm_out(0xee);
  data_out(0x22);        //chp

  comm_out(0xf6);
  data_out(0x10);

  comm_out(0xc4);
  data_out(0x12);        //vrh

  comm_out(0xc5);
  data_out(0x22);        //vcm

  comm_out(0xc6);
  data_out(0x0a);        //vdv


  comm_out(0xe2);
  data_out(0x07);    //kp5    kp4

  comm_out(0xe6);
  data_out(0x70);    //kn1    kn0

  comm_out(0xe3);
  data_out(0x30); //rp1    rp0

  comm_out(0xe9);
```

```
    data_out(0x03); // rn1    rn0

    comm_out(0xe4);
    data_out(0x00); // vrp0

    comm_out(0xeb);
    data_out(0x00); //    vrn1

    comm_out(0xe5);
    data_out(0x06); //vrp1

    comm_out(0xea);
    data_out(0x03); // vrn0


    comm_out(0xe1);
    data_out(0x02);    //kp3    kp2

    comm_out(0xe7);
    data_out(0x20); //kn3    kn2

    comm_out(0xe0);
    data_out(0x02); //kp1 kp0

    comm_out(0xe8);
    data_out(0x20); //kn5    kn4



    comm_out(0x11); //sleep out
    Delay_ms(50);
    comm_out(0x3a);
    data_out(0x05);

    comm_out(0x36);
    data_out(0xd4);// ss gs bgr
    Delay_ms(50);

    comm_out(0x29);//display on


}


setDispArea(unsigned int s_col,e_col,s_page,e_page)
{
   comm_out(0x2a);          // Set col address
   data_out((s_col &0x0100)>>8);
   data_out(s_col &0x00ff);    // Start col address
   data_out((e_col &0x0100)>>8);
   data_out(e_col &0x00ff);    // End col address
   comm_out(0x2b);          // Set page address
   data_out((s_page &0x0100)>>8);
```

```
   data_out(s_page &0x00ff);  // Start col address
   data_out((e_page &0x0100)>>8);
   data_out(e_page &0x00ff);  // End col address
   comm_out(0x2c);          // Write Display Data
}


                void sleepin (void)
                {
                      comm_out(0x28);
                      delayms(10);
                      comm_out(0x10);
                      delayms(120);
                }

                void sleepout (void)
                {
                      comm_out(0x11);
                      delayms(120);
                      comm_out(0x29);
                      delayms(10);
                }
```

8、CPT 1.44 Panel initial code

```
void Initial_cpt144(void)    //cpt 1.44
{


comm_out (0xb4);
data_out (0x07);
comm_out (0xb1);
data_out (0x18);
//--------------------INTER REG_EN----------------------------

comm_out (0xff);
data_out (0xa5);

comm_out(0xec);
data_out(0x89);
comm_out (0xed);
data_out (0x25);
comm_out (0xee);
data_out (0x22);
comm_out (0xf6);
data_out (0x10);
comm_out (0xc4);
data_out (0x03);//vrh
```

```
comm_out (0xc5);
data_out (0x7f);//72
data_out (0x3e); //1e

comm_out (0xc6);
data_out (0x3f); //0e vdv=3f
comm_out (0xc7);
data_out (0x03);//11//bd

comm_out (0xe0);
data_out (0x70);//70
comm_out (0xe8);
data_out (0x07);//07
comm_out (0xe1);
data_out (0x12);//40 72
comm_out (0xe7);
data_out (0x03);//04

comm_out (0xe2);
data_out (0x77); //07//77
comm_out (0xe6);
data_out (0x77); //70//77

comm_out (0xe3);
data_out (0x07); //14//07
comm_out (0xe9);
data_out (0x73); //15

comm_out (0xe4);
data_out (0x00); //02
comm_out (0xeb);
data_out (0x03); //02

comm_out (0xe5);
data_out (0x00); //01
comm_out (0xea);
data_out (0x00); //01

comm_out (0x11);
delayms(10);
comm_out (0x3a);
data_out (0x05);
comm_out (0x36);
data_out (0xdc);

delayms(10);
comm_out (0x29);
}
```

```
setDispArea(unsigned int s_col,e_col,s_page,e_page)
{
   comm_out(0x2a);          // Set col address
   data_out((s_col &0x0100)>>8);
   data_out(s_col &0x00ff);    // Start col address
   data_out((e_col &0x0100)>>8);
   data_out(e_col &0x00ff);    // End col address
   comm_out(0x2b);          // Set page address
   data_out((s_page &0x0100)>>8);
   data_out(s_page &0x00ff);  // Start col address
   data_out((e_page &0x0100)>>8);
   data_out(e_page &0x00ff);  // End col address
   comm_out(0x2c);          // Write Display Data
}
```

```
void sleepin (void)
{
      comm_out(0x28);
      delayms(10);
      comm_out(0x10);
      delayms(120);
}

void sleepout (void)
{
      comm_out(0x11);
      delayms(120);
      comm_out(0x29);
      delayms(10);
}
```

9、HSD 1.44 Panel initial code

```
void Initial_cpt144(void)    //cpt 1.44
{

comm_out (0xb4);
data_out (0x00);
comm_out (0xb1);
data_out (0x18);
//--------------------INTER REG_EN---------------------------
comm_out (0xff);
data_out (0xa5);
comm_out(0xec);
data_out(0x89);

comm_out (0xed);
data_out (0x25);

comm_out (0xee);
data_out (0x22);

comm_out (0xf6);
data_out (0x10);

comm_out (0xc4);
data_out (0x03);

comm_out (0xc0);
data_out (0x12);

comm_out (0xc5);
data_out (0x72);
data_out (0x1e);

comm_out (0xc6);
data_out (0x1e);

comm_out (0xe0);
data_out (0x00);
comm_out (0xe8);
data_out (0x07);
comm_out (0xe1);
data_out (0x12);
comm_out (0xe7);
data_out (0x03);
comm_out (0xe2);
data_out (0x77);
comm_out (0xe6);
data_out (0x77);

comm_out (0xe3);
data_out (0x07);
```

```
comm_out (0xe9);
data_out (0x73);
comm_out (0xe4);
data_out (0x00);
comm_out (0xeb);
data_out (0x07);

comm_out (0xe5);
data_out (0x00);
comm_out (0xea);
data_out (0x00);

comm_out (0x11);
delayms(10);
comm_out (0x3a);
data_out (0x05);
comm_out (0x36);
data_out (0xdc);

delayms(10);
comm_out (0x29);
}

setDispArea(unsigned int s_col,e_col,s_page,e_page)
{
   comm_out(0x2a);          // Set col address
   data_out((s_col &0x0100)>>8);
   data_out(s_col &0x00ff);    // Start col address
   data_out((e_col &0x0100)>>8);
   data_out(e_col &0x00ff);    // End col address
   comm_out(0x2b);          // Set page address
   data_out((s_page &0x0100)>>8);
   data_out(s_page &0x00ff);  // Start col address
   data_out((e_page &0x0100)>>8);
   data_out(e_page &0x00ff);  // End col address
   comm_out(0x2c);          // Write Display Data
}

                    void sleepin (void)
                    {
                          comm_out(0x28);
                          delayms(10);
                          comm_out(0x10);
                          delayms(120);
                    }
                    void sleepout (void)
                    {
                          comm_out(0x11);
                          delayms(120);
                          comm_out(0x29);
                          delayms(10);
                    }
```

10. BOE 1.77 initial code

```
void NV3021C_BOE177_Initial(void)//for BOE 1.77
{

//VCI=2.8V
//----------------Star Initial Sequence-------//

comm_out(0xb4);
data_out(0x07);    //d2=1 frame inversion
comm_out(0xb1);
data_out(0x18);    //frame rate
//--------------------INTER REG_EN----------------------------

comm_out(0xff);
data_out(0xa5);

comm_out(0xec);
data_out(0x89);
comm_out(0xed);
data_out(0x15);
comm_out(0xee);
data_out(0x12);
comm_out(0xf6);
data_out(0x10);
comm_out(0xc4);
data_out(0x0e); //VRH
comm_out(0xc5);
data_out(0x5c); //vcom
comm_out(0xc6);
data_out(0x0e); //vdv

comm_out(0xe0);
data_out(0x00); //gamma
comm_out(0xe8);
data_out(0x77);
comm_out(0xe1);
data_out(0x57);
comm_out(0xe7);
data_out(0x02);
comm_out(0xe2);
data_out(0x01);
comm_out(0xe6);
data_out(0x46);
comm_out(0xe3);
data_out(0x67);
comm_out(0xe9);
data_out(0x10);
comm_out(0xe4);
data_out(0x02);
comm_out(0xeb);
data_out(0x02);
comm_out(0xe5);
```

```
data_out(0x00);
comm_out(0xea);
data_out(0x00);

comm_out(0x11);
Delay_ms(120);
comm_out(0x3a);
data_out(0x05);
comm_out(0x36);
data_out(0x80);//80
comm_out(0x29);
}
```

11. Truly 1.77 initial code

```
void NV3021C_Truly177_Initial(void)//for truly 1.77
{

//VCI=2.8V
//----------------Star Initial Sequence-------//
comm_out(0xb4);
data_out(0x07);    //d2=1 frame inversion    07
comm_out(0xb1);
data_out(0x1c);    //frame rate    1c
//--------------------INTER REG_EN----------------------------
comm_out(0xff);
data_out(0xa5);

comm_out(0xec);
data_out(0x89);
comm_out(0xed);
data_out(0x25);
comm_out(0xee);
data_out(0x22);
comm_out(0xf6);
data_out(0x10);
comm_out(0xc4);
data_out(0x10); //VRH      1c 10 0c
comm_out(0xc5);
data_out(0x38); //vcom
comm_out(0xc6);
data_out(0x0e); //vdv

comm_out(0xe0); //Gamma
data_out(0x00);
comm_out(0xe8);
data_out(0x77);
```

```
comm_out(0xe1);
data_out(0x55);
comm_out(0xe7);
data_out(0x22);
comm_out(0xe2);
data_out(0x07);
comm_out(0xe6);
data_out(0x03);
comm_out(0xe3);
data_out(0x50);
comm_out(0xe9);
data_out(0x05);
comm_out(0xe4);
data_out(0x00);
comm_out(0xeb);
data_out(0x00);
comm_out(0xe5);
data_out(0x00);
comm_out(0xea);
data_out(0x00);    //Gamma

comm_out(0x11);
delayms(120);
comm_out(0x3a);
data_out(0x05);
comm_out(0x36);
data_out(0x80); //
comm_out(0x29);
}

setDispArea(unsigned int s_col,e_col,s_page,e_page)
{
   comm_out(0x2a);          // Set col address
   data_out((s_col &0x0100)>>8);
   data_out(s_col &0x00ff);    // Start col address
   data_out((e_col &0x0100)>>8);
   data_out(e_col &0x00ff);    // End col address
   comm_out(0x2b);          // Set page address
   data_out((s_page &0x0100)>>8);
   data_out(s_page &0x00ff);  // Start col address
   data_out((e_page &0x0100)>>8);
   data_out(e_page &0x00ff);  // End col address
   comm_out(0x2c);          // Write Display Data
}


  sleep_in(void)
{
      comm_out(0x28);//display off
      Delay_ms(50);
      comm_out(0x10);//sleep in
      Delay_ms(120);
}
```

```
sleep_out(void)
{
      comm_out(0x11);//sleep out
      Delay_ms(120);
      comm_out(0x29);//display on
      Delay_ms(50);
}



setDispArea(unsigned int s_col,e_col,s_page,e_page)
{
   comm_out(0x2a);          // Set col address
   data_out((s_col &0x0100)>>8);
   data_out(s_col &0x00ff);    // Start col address
   data_out((e_col &0x0100)>>8);
   data_out(e_col &0x00ff);    // End col address
   comm_out(0x2b);          // Set page address
   data_out((s_page &0x0100)>>8);
   data_out(s_page &0x00ff);  // Start col address
   data_out((e_page &0x0100)>>8);
   data_out(e_page &0x00ff);  // End col address
   comm_out(0x2c);          // Write Display Data
}


  sleep_in(void)
{
      comm_out(0x28);//display off
      Delay_ms(120);
      comm_out(0x10);//sleep in
      Delay_ms(50);
}

sleep_out(void)
{
      comm_out(0x11);//sleep out
      Delay_ms(120);
      comm_out(0x29);//display on
      Delay_ms(50);
}
```

12、Revision History

Version	Data	Description
V0.1	2016/11/30	Original
V0.2	2016/12/26	Update SCH and cpt,ctc 1.77 code
V0.3	2016/12/29	Changed ID=D4,80,66
V0.4	2017/04/27	Update hsd1.77, boe1.77 code