



OTM4001A APPLICATION NOTES

720-channel 6-bit Source Driver with System-on-chip for Color Amorphous TFT-LCDs

Preliminary

Apr. 07,2010

Version 0.4

Table of Contents

	<u>PAGE</u>
1. CPT2.8" 3.0" 3.2" APPLICATION NOTE	3
1.1. CPT2.8" APPLICATION CIRCUIT	3
1.2. CPT3.0" APPLICATION CIRCUIT	4
1.3. CPT3.2" APPLICATION CIRCUIT	5
1.4. INITIAL CODE	6
1.4.1. CPT_2.8" initial code	6
1.4.2. CPT_3.0" initial code	9
1.4.3. CPT_3.2" initial code	12
2. CMO3.0" APPLICATION NOTE	15
2.1. CMO3.0" APPLICATION CIRCUIT	15
INITIAL CODE	16
2.1.1. CMO_3.0" initial code	16
3. HTT3.0" APPLICATION NOTE	19
3.1. APPLICATION CIRCUIT	19
INITIAL CODE	20
3.1.1. HTT_3.0" initial code	20
4. GPT2.5" 2.8" 3.0" APPLICATION NOTE	23
4.1. GPT2.5" APPLICATION CIRCUIT	23
4.2. GPT2.8" APPLICATION CIRCUIT	24
4.3. GPT3.0" APPLICATION CIRCUIT	25
INITIAL CODE	26
4.3.1. GPT2.5" initial code	26
4.3.2. GPT2.8" initial code	29
4.3.3. GPT3.0" initial code	32
5. AUO 3.2" APPLICATION NOTE	35
5.1. AUO 3.2" APPLICATION CIRCUIT	35
INITIAL CODE	36
5.1.1. AUO 3.2" initial code	36
6. BOE 2.8" APPLICATION NOTE	39
6.1. BOE 2.8" APPLICATION CIRCUIT	39
INITIAL CODE	40
6.1.1. BOE 2.8" initial code	40
7. HSD 3.0" APPLICATION NOTE	43
7.1. HSD 3.0" APPLICATION CIRCUIT	43
INITIAL CODE	44
7.1.1. HSD 3.0" initial code	44
8. REVISION HISTORY	48

1. CPT2.8" 3.0" 3.2" APPLICATION NOTE

1.1. CPT2.8" Application Circuit

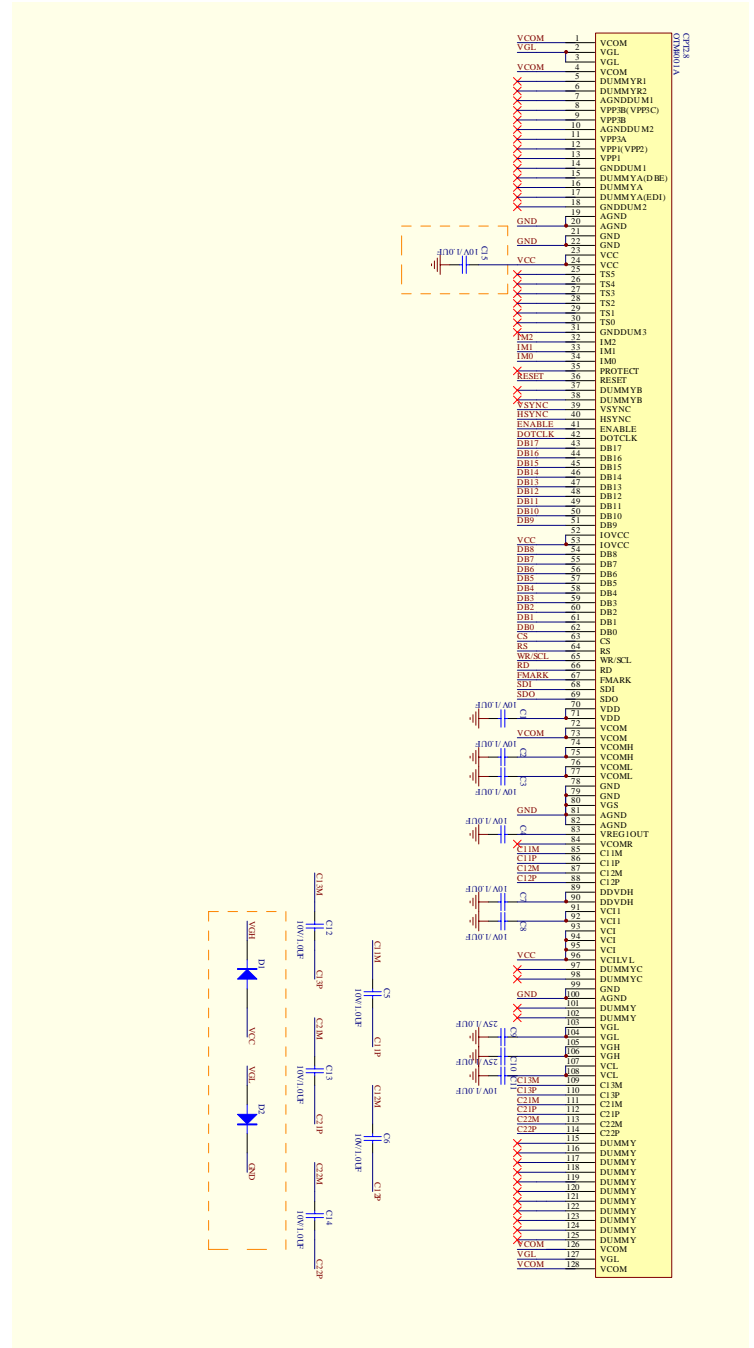
CLAF028JD01A0

IM2	IM1	IM0/ ID	Interface Mode	DB Pin	Colors
0	0	0	80-system 18-bit interface	DB17-0	262,144
0	0	1	80-system 9-bit interface	DB17-9	262,144
0	0	1	80-system 16-bit interface	DB17-10, DB8-1	262,144 see Note 1
0	1	1	80-system 8-bit interface	DB17-10	262,144 see Note 2
1	0	*(ID)	Clock synchronous serial interface	-	65,536
1	1	0	Setting disabled	-	-
1	1	1	Setting disabled	-	-

- Notes: 1. 65,536 colors in one transfer mode
2. 65,536 colors in two transfers mode

Note(*): C15, D1, D2

are not used, but Pls. Keep these pad on FPC.

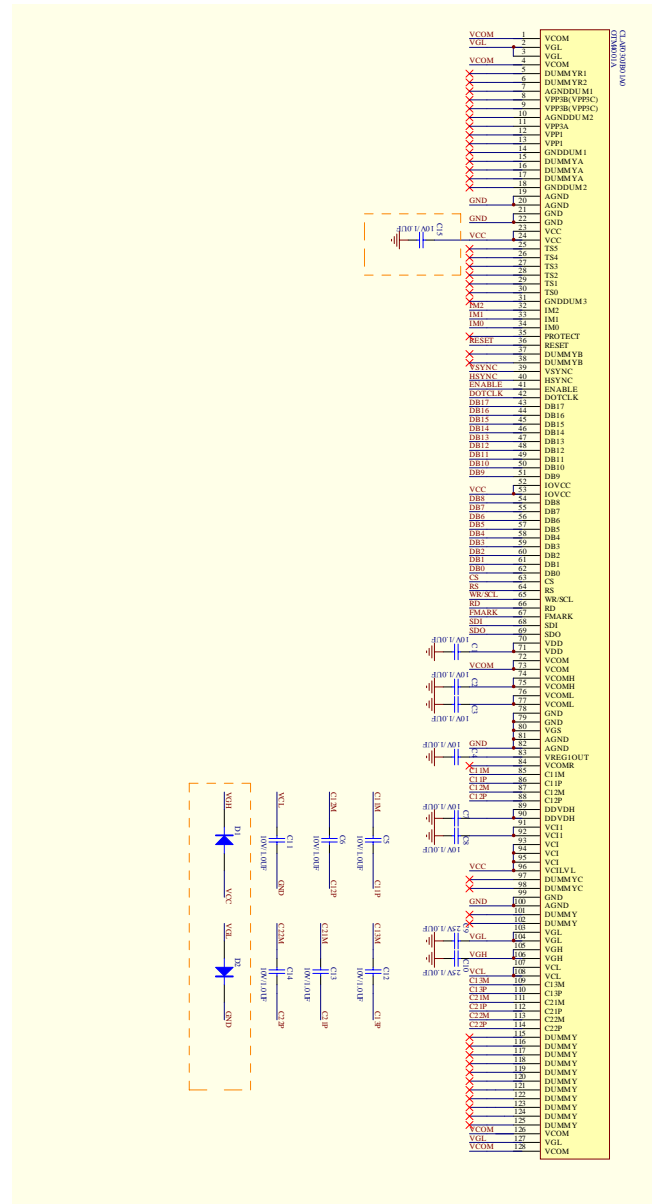


1.2. CPT3.0⁺ Application Circuit

CLAF030JB01A0

IM2	IM1	IM0/ ID	Interface Mode	DB Pin	Colors
0	0	0	80-system 18-bit Interface	DB17-0	262,144
0	0	1	80-system 9-bit Interface	DB17-9	262,144
0	1	0	80-system 18-bit interface	DB17-10, DB8-1	262,144 see Note 1
0	1	1	80-system 8-bit interface	DB17-10	262,144 see Note 2
1	0	*(ID)	Clock synchronous serial interface	-	65,536
1	1	0	Setting disabled	-	-
1	1	1	Setting disabled	-	-

Notes: 1. 65,536 colors in one transfer mode
2. 65,536 colors in two transfers mode



Note(*): C15, D1, D2

are not used, but Pls. Keep
these pad on FPC.

1.3. CPT3.2" Application Circuit

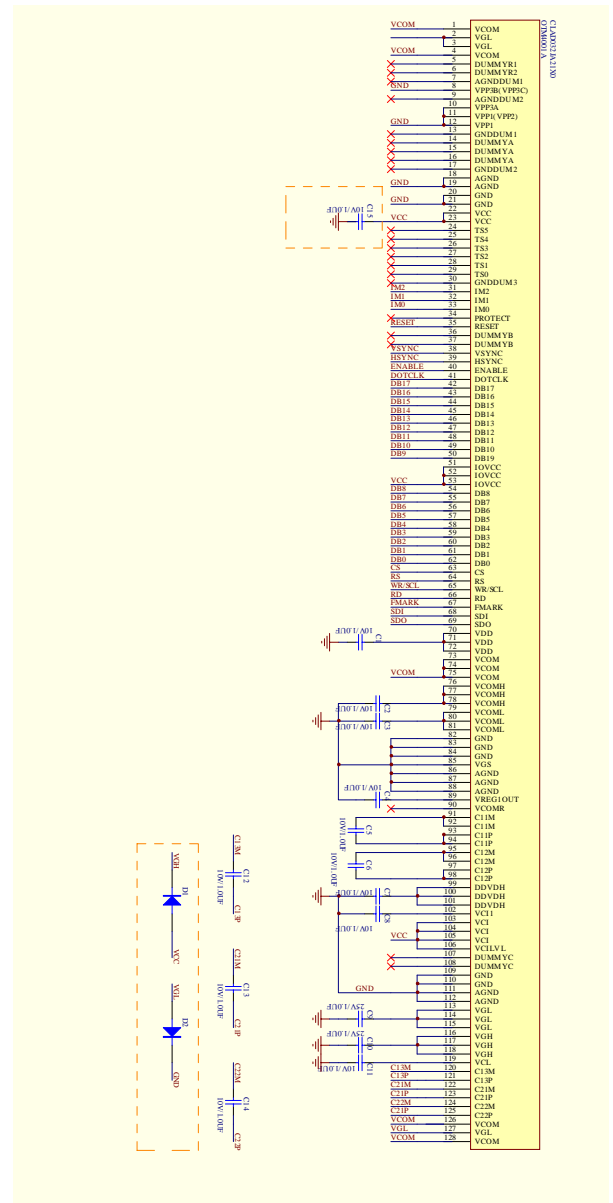
CLAD032JA21X0

IM2	IM1	IM0/ ID	Interface Mode	DB Pin	Colors
0	0	0	80-system 18-bit interface	DB17-0	282,144
0	0	1	80-system 9-bit interface	DB17-9	282,144
0	1	0	80-system 18-bit interface	DB17-10, DB8-1	282,144 see Note 2
0	1	1	80-system 8-bit interface	DB17-10	282,144 see Note 2
1	0	*(ID)	Clock synchronous serial interface	-	65,536
1	1	0	Setting disabled	-	-
1	1	1	Setting disabled	-	-

Notes: 1. 65,536 colors in one transfer mode
2. 65,536 colors in two transfers mode

Note(*): C15, D1, D2

are not used, but Pls. Keep these pad on FPC.



1.4. Initial Code

1.4.1. CPT_2.8" initial code

```

void Code( )
{
//----- Power supply condition: VCC=IOVCC=VCI=2.8V -----//
//-----Power on sequence-----//
write_ctrl(0x0606);          write_data(0x0000);
delayms(100);
write_ctrl(0x0007);          write_data(0x0001);
delayms(5);
write_ctrl(0x0110);          write_data(0x0001);
delayms(5);
write_ctrl(0x0100);          write_data(0x17B0);
write_ctrl(0x0101);          write_data(0x0147);
write_ctrl(0x0102);          write_data(0x019D);
write_ctrl(0x0103);          write_data(0x3600);
write_ctrl(0x0281);          write_data(0x0010);
delayms(5);
write_ctrl(0x0102);          write_data(0x01BD);
delayms(5);
write_ctrl(0x0000);          write_data(0x0000);
write_ctrl(0x0001);          write_data(0x0100);
write_ctrl(0x0002);          write_data(0x0100);
write_ctrl(0x0003);          write_data(0x10B0);
write_ctrl(0x0006);          write_data(0x0000);
write_ctrl(0x0008);          write_data(0x0503);
write_ctrl(0x0009);          write_data(0x0001);
write_ctrl(0x000B);          write_data(0x0010);
write_ctrl(0x000C);          write_data(0x0000);
write_ctrl(0x000F);          write_data(0x0000);
write_ctrl(0x0007);          write_data(0x0001);
//-----Panel interface control-----//
write_ctrl(0x0010);          write_data(0x0014);
write_ctrl(0x0011);          write_data(0x0202);
write_ctrl(0x0012);          write_data(0x0300);
write_ctrl(0x0020);          write_data(0x021e);
write_ctrl(0x0021);          write_data(0x0202);
write_ctrl(0x0090);          write_data(0x8000);
write_ctrl(0x0092);          write_data(0x0000);
//-----Voltage setting-----//
write_ctrl(0x0100);          write_data(0x13b0);
delayms(10);
write_ctrl(0x0101);          write_data(0x0147);
delayms(10);
write_ctrl(0x0102);          write_data(0x018D);
delayms(10);
write_ctrl(0x0103);          write_data(0x0200);
delayms(10);
write_ctrl(0x0107);          write_data(0x0000);

```

```

delayms(10);
write_ctrl(0x0110);          write_data(0x0001);
delayms(10);
//-----//

write_ctrl(0x0210);          write_data(0x0000);
write_ctrl(0x0211);          write_data(0x00ef);
write_ctrl(0x0212);          write_data(0x0000);
write_ctrl(0x0213);          write_data(0x018f);
write_ctrl(0x0200);          write_data(0x0000);
write_ctrl(0x0201);          write_data(0x0000);
write_ctrl(0x0280);          write_data(0x0000);
write_ctrl(0x0281);          write_data(0x0000);
write_ctrl(0x0282);          write_data(0x0000);
delayms(10);
//-----Gamma setting-----//

write_ctrl(0x0300);          write_data(0x0010);
write_ctrl(0x0301);          write_data(0x0234);
write_ctrl(0x0302);          write_data(0x0438);
write_ctrl(0x0303);          write_data(0x3704);
write_ctrl(0x0304);          write_data(0x3402);
write_ctrl(0x0305);          write_data(0x1000);
write_ctrl(0x0306);          write_data(0x1403);
write_ctrl(0x0307);          write_data(0x0314);
write_ctrl(0x0308);          write_data(0x0305);
write_ctrl(0x0309);          write_data(0x0305);
write_ctrl(0x030A);          write_data(0x0D06);
write_ctrl(0x030B);          write_data(0x0E02);
write_ctrl(0x030C);          write_data(0x020E);
write_ctrl(0x030D);          write_data(0x060D);
write_ctrl(0x030E);          write_data(0x0403);
write_ctrl(0x030F);          write_data(0x0503);
//-----//

write_ctrl(0x0400);          write_data(0x3500);
write_ctrl(0x0401);          write_data(0x0001);
write_ctrl(0x0404);          write_data(0x0000);
write_ctrl(0x0500);          write_data(0x0000);
write_ctrl(0x0501);          write_data(0x0000);
write_ctrl(0x0502);          write_data(0x0000);
write_ctrl(0x0503);          write_data(0x0000);
write_ctrl(0x0504);          write_data(0x0000);
write_ctrl(0x0505);          write_data(0x0000);
write_ctrl(0x0600);          write_data(0x0000);
write_ctrl(0x0606);          write_data(0x0000);
write_ctrl(0x06F0);          write_data(0x0000);

//----- Orise mode -----//

write_ctrl(0x07F0);          write_data(0x5420);

```

```
write_ctrl(0x07F3);      write_data(0x288b);
write_ctrl(0x07F4);      write_data(0x0022);
write_ctrl(0x07F5);      write_data(0x0011);
write_ctrl(0x07F0);      write_data(0x0000);
```

```
write_ctrl(0x0007);      write_data(0x0173);
delayms(5);
write_ctrl(0x0007);      write_data(0x0061);
delayms(5);
write_ctrl(0x0007);      write_data(0x0173);
delayms(500);
}
```

```
void Pattern_RED( )
{
Signed int j;
write_ctrl(0x0210);      write_data(0x0000); // xstart
write_ctrl(0x0211);      write_data(0x00EF); // xend
write_ctrl(0x0212);      write_data(0x0000); // ystart
write_ctrl(0x0213);      write_data(0x018F); // yend
write_ctrl(0x0200);      write_data(0x0000); // xstart
write_ctrl(0x0201);      write_data(0x0000); // ystart
write_ctrl(0x0202); // write GRAM
for( j =0; j<240*400; j++ )
{
Write_ Data (0xFC,0x00, 0x00); // red
}
}
```

CPU Interface enter sleep mode code

```
void Code( )
{
write_ctrl(0x07F0);      write_data(0x5420);
write_ctrl(0x07DE);      write_data(0x0C00);
write_ctrl(0x07F0);      write_data(0x0000);
write_ctrl(0x0007);      write_data(0x0000); //Display off
delay_ms(50);
write_ctrl(0x0100);      write_data(0x0002); //Set to sleep mode
delayms(500);
}
```

CPU Interface exit sleep and deep stand by mode code

```
void code( )
{
write_ctrl(0x07F0);      write_data(0x5420);
write_ctrl(0x07DE);      write_data(0x0400);
write_ctrl(0x07F0);      write_data(0x0000);
write_ctrl(0x0100);      write_data(0x13B0); // Release state
delay_ms(50);
write_ctrl(0x0007);      write_data(0x0173); // Display on
delayms(500);
}
```


1.4.2. CPT_3.0" initial code

```

void Code( )
{
//----- Power supply condition: VCC=IOVCC=VCI=2.8V -----//
//-----Power on sequence-----//
write_ctrl(0x0606);          write_data(0x0000);
delayms(100);
write_ctrl(0x0007);          write_data(0x0001);
delayms(5);
write_ctrl(0x0110);          write_data(0x0001);
delayms(5);
write_ctrl(0x0100);          write_data(0x17B0);
write_ctrl(0x0101);          write_data(0x0147);
write_ctrl(0x0102);          write_data(0x019D);
write_ctrl(0x0103);          write_data(0x3600);
write_ctrl(0x0281);          write_data(0x0010);
delayms(5);
write_ctrl(0x0102);          write_data(0x01BD);
delayms(5);
write_ctrl(0x0000);          write_data(0x0000);
write_ctrl(0x0001);          write_data(0x0100);
write_ctrl(0x0002);          write_data(0x0100);
write_ctrl(0x0003);          write_data(0x10B0);
write_ctrl(0x0006);          write_data(0x0000);
write_ctrl(0x0008);          write_data(0x0503);
write_ctrl(0x0009);          write_data(0x0001);
write_ctrl(0x000B);          write_data(0x0010);
write_ctrl(0x000C);          write_data(0x0000);
write_ctrl(0x000F);          write_data(0x0000);
write_ctrl(0x0007);          write_data(0x0001);
//-----Panel interface control-----//
write_ctrl(0x0010);          write_data(0x0014);
write_ctrl(0x0011);          write_data(0x0202);
write_ctrl(0x0012);          write_data(0x0300);
write_ctrl(0x0020);          write_data(0x021e);
write_ctrl(0x0021);          write_data(0x0202);
write_ctrl(0x0090);          write_data(0x8000);
write_ctrl(0x0092);          write_data(0x0000);
//-----Voltage setting-----//
write_ctrl(0x0100);          write_data(0x13b0);
delayms(10);
write_ctrl(0x0101);          write_data(0x0147);
delayms(10);
write_ctrl(0x0102);          write_data(0x018D);
delayms(10);
write_ctrl(0x0103);          write_data(0x0200);
delayms(10);
write_ctrl(0x0107);          write_data(0x0000);
delayms(10);

```

```

write_ctrl(0x0110);          write_data(0x0001);
delayms(10);
//-----//
write_ctrl(0x0210);          write_data(0x0000);
write_ctrl(0x0211);          write_data(0x00ef);
write_ctrl(0x0212);          write_data(0x0000);
write_ctrl(0x0213);          write_data(0x018f);
write_ctrl(0x0200);          write_data(0x0000);
write_ctrl(0x0201);          write_data(0x0000);
write_ctrl(0x0280);          write_data(0x0000);
write_ctrl(0x0281);          write_data(0x0000);
write_ctrl(0x0282);          write_data(0x0000);
delayms(10);
//-----Gamma setting-----//
write_ctrl(0x0300);          write_data(0x0010);
write_ctrl(0x0301);          write_data(0x0235);
write_ctrl(0x0302);          write_data(0x0438);
write_ctrl(0x0303);          write_data(0x3804);
write_ctrl(0x0304);          write_data(0x3502);
write_ctrl(0x0305);          write_data(0x1000);
write_ctrl(0x0306);          write_data(0x1403);
write_ctrl(0x0307);          write_data(0x0314);
write_ctrl(0x0308);          write_data(0x0305);
write_ctrl(0x0309);          write_data(0x0302);
write_ctrl(0x030A);          write_data(0x0D06);
write_ctrl(0x030B);          write_data(0x0E02);
write_ctrl(0x030C);          write_data(0x020E);
write_ctrl(0x030D);          write_data(0x060D);
write_ctrl(0x030E);          write_data(0x0203);
write_ctrl(0x030F);          write_data(0x0503);
//-----//

write_ctrl(0x0400);          write_data(0x3500);
write_ctrl(0x0401);          write_data(0x0001);
write_ctrl(0x0404);          write_data(0x0000);
write_ctrl(0x0500);          write_data(0x0000);
write_ctrl(0x0501);          write_data(0x0000);
write_ctrl(0x0502);          write_data(0x0000);
write_ctrl(0x0503);          write_data(0x0000);
write_ctrl(0x0504);          write_data(0x0000);
write_ctrl(0x0505);          write_data(0x0000);
write_ctrl(0x0600);          write_data(0x0000);
write_ctrl(0x0606);          write_data(0x0000);
write_ctrl(0x06F0);          write_data(0x0000);

//----- Orise mode -----//

write_ctrl(0x07F0);          write_data(0x5420);
write_ctrl(0x07F3);          write_data(0x288b);

```

```
write_ctrl(0x07F4);      write_data(0x0022);
write_ctrl(0x07F5);      write_data(0x0011);
write_ctrl(0x07F0);      write_data(0x0000);
```

```
write_ctrl(0x0007);      write_data(0x0173);
delayms(5);
write_ctrl(0x0007);      write_data(0x0061);
delayms(5);
write_ctrl(0x0007);      write_data(0x0173);
delayms(500);
}
```

```
void Pattern_RED( )
{
Signed int j;
write_ctrl(0x0210);      write_data(0x0000); // xstart
write_ctrl(0x0211);      write_data(0x00EF); // xend
write_ctrl(0x0212);      write_data(0x0000); // ystart
write_ctrl(0x0213);      write_data(0x018F); // yend
write_ctrl(0x0200);      write_data(0x0000); // xstart
write_ctrl(0x0201);      write_data(0x0000); // ystart
write_ctrl(0x0202); // write GRAM
for( j =0; j<240*400; j++ )
{
Write_ Data (0xFC,0x00, 0x00); // red
}
}
```

CPU Interface enter sleep mode code

```
void Code( )
{
write_ctrl(0x07F0);      write_data(0x5420);
write_ctrl(0x07DE);      write_data(0x0C00);
write_ctrl(0x07F0);      write_data(0x0000);
write_ctrl(0x0007);      write_data(0x0000); //Display off
delay_ms(50);
write_ctrl(0x0100);      write_data(0x0002); //Set to sleep mode
delayms(500);
}
```

CPU Interface exit sleep and deep stand by mode code

```
void code( )
{
write_ctrl(0x07F0);      write_data(0x5420);
write_ctrl(0x07DE);      write_data(0x0400);
write_ctrl(0x07F0);      write_data(0x0000);
write_ctrl(0x0100);      write_data(0x13B0); // Release state
delay_ms(50);
write_ctrl(0x0007);      write_data(0x0173); // Display on
delayms(500);
}
```

1.4.3. CPT_3.2" initial code

```

void Code( )
{
//----- Power supply condition: VCC=IOVCC=VCI=2.8V -----//
//-----Power on sequence-----//
write_ctrl(0x0606);      write_data(0x0000);
delayms(100);
write_ctrl(0x0007);      write_data(0x0001);
delayms(5);
write_ctrl(0x0110);      write_data(0x0001);
delayms(5);
write_ctrl(0x0100);      write_data(0x17B0);
write_ctrl(0x0101);      write_data(0x0147);
write_ctrl(0x0102);      write_data(0x019D);
write_ctrl(0x0103);      write_data(0x3600);
write_ctrl(0x0281);      write_data(0x0010);
delayms(5);
write_ctrl(0x0102);      write_data(0x01BD);
delayms(5);
write_ctrl(0x0000);      write_data(0x0000);
write_ctrl(0x0001);      write_data(0x0100);
write_ctrl(0x0002);      write_data(0x0100);
write_ctrl(0x0003);      write_data(0x10B0);
write_ctrl(0x0006);      write_data(0x0000);
write_ctrl(0x0008);      write_data(0x0503);
write_ctrl(0x0009);      write_data(0x0001);
write_ctrl(0x000B);      write_data(0x0010);
write_ctrl(0x000C);      write_data(0x0000);
write_ctrl(0x000F);      write_data(0x0000);
write_ctrl(0x0007);      write_data(0x0001);
//-----Panel interface control-----
write_ctrl(0x0010);      write_data(0x0014);
write_ctrl(0x0011);      write_data(0x0202);
write_ctrl(0x0012);      write_data(0x0300);
write_ctrl(0x0020);      write_data(0x021e);
write_ctrl(0x0021);      write_data(0x0202);
write_ctrl(0x0090);      write_data(0x8000);
write_ctrl(0x0092);      write_data(0x0000);
//-----Voltage setting-----
write_ctrl(0x0100);      write_data(0x13b0);
delayms(10);
write_ctrl(0x0101);      write_data(0x0147);
delayms(10);
write_ctrl(0x0102);      write_data(0x018f);
delayms(10);
write_ctrl(0x0103);      write_data(0x0e00);
delayms(10);
write_ctrl(0x0107);      write_data(0x0000);
delayms(10);

```

```

write_ctrl(0x0110);      write_data(0x0001);
delayms(10);
write_ctrl(0x0210);      write_data(0x0000);
write_ctrl(0x0211);      write_data(0x00ef);
write_ctrl(0x0212);      write_data(0x0000);
write_ctrl(0x0213);      write_data(0x018f);
write_ctrl(0x0200);      write_data(0x0000);
write_ctrl(0x0201);      write_data(0x0000);
write_ctrl(0x0280);      write_data(0x0000);
write_ctrl(0x0281);      write_data(0x0010);
write_ctrl(0x0282);      write_data(0x0000);
delayms(10);

```

//-----Gamma setting-----

```

write_ctrl(0x0300);      write_data(0x0000);
write_ctrl(0x0301);      write_data(0x0610);
write_ctrl(0x0302);      write_data(0x0831);
write_ctrl(0x0303);      write_data(0x3108);
write_ctrl(0x0304);      write_data(0x1006);
write_ctrl(0x0305);      write_data(0x0000);
write_ctrl(0x0306);      write_data(0x100f);
write_ctrl(0x0307);      write_data(0x0f10);
write_ctrl(0x0308);      write_data(0x0404);
write_ctrl(0x0309);      write_data(0x0503);
write_ctrl(0x030A);      write_data(0x0f03);
write_ctrl(0x030B);      write_data(0x0f00);
write_ctrl(0x030C);      write_data(0x000f);
write_ctrl(0x030D);      write_data(0x030f);
write_ctrl(0x030E);      write_data(0x0305);
write_ctrl(0x030F);      write_data(0x0403);

```

```

write_ctrl(0x0400);      write_data(0x3500);
write_ctrl(0x0401);      write_data(0x0000);
write_ctrl(0x0404);      write_data(0x0000);
write_ctrl(0x0500);      write_data(0x0000);
write_ctrl(0x0501);      write_data(0x0000);
write_ctrl(0x0502);      write_data(0x0000);
write_ctrl(0x0503);      write_data(0x0000);
write_ctrl(0x0504);      write_data(0x0000);
write_ctrl(0x0505);      write_data(0x0000);
write_ctrl(0x0600);      write_data(0x0000);
write_ctrl(0x0606);      write_data(0x0000);
write_ctrl(0x06F0);      write_data(0x0000);

```

//-----Orise mode -----//

```

write_ctrl(0x07F0);      write_data(0x5420);
write_ctrl(0x07F3);      write_data(0x288b);
write_ctrl(0x07F4);      write_data(0x0022);
write_ctrl(0x07F5);      write_data(0x0011);
write_ctrl(0x07F0);      write_data(0x0000);

```

```

write_ctrl(0x0007);      write_data(0x0173);
delayms(5);
write_ctrl(0x0007);      write_data(0x0061);
delayms(5);
write_ctrl(0x0007);      write_data(0x0173);
delayms(500);
}
void Pattern_RED( )
{
    Signed int j;
write_ctrl(0x0210);      write_data(0x0000); // xstart
write_ctrl(0x0211);      write_data(0x00EF); // xend
write_ctrl(0x0212);      write_data(0x0000); // ystart
write_ctrl(0x0213);      write_data(0x018F); // yend
write_ctrl(0x0200);      write_data(0x0000); // xstart
write_ctrl(0x0201);      write_data(0x0000); // ystart
write_ctrl(0x0202); // write GRAM
for( j =0; j<240*400; j++ )
{
Write_ Data (0xFC,0x00, 0x00); // red
}
}

```

CPU Interface enter sleep mode code

```

void Code( )
{
write_ctrl(0x07F0);      write_data(0x5420);
write_ctrl(0x07DE);      write_data(0x0C00);
write_ctrl(0x07F0);      write_data(0x0000);
write_ctrl(0x0007);      write_data(0x0000); //Display off
delay_ms(50);
write_ctrl(0x0100);      write_data(0x0002); //Set to sleep mode
delayms(500);
}

```

CPU Interface exit sleep and deep stand by mode code

```

void code( )
{
write_ctrl(0x07F0);      write_data(0x5420);
write_ctrl(0x07DE);      write_data(0x0400);
write_ctrl(0x07F0);      write_data(0x0000);
write_ctrl(0x0100);      write_data(0x13B0); // Release state
delay_ms(50);
write_ctrl(0x0007);      write_data(0x0173); // Display on
delayms(500);
}

```

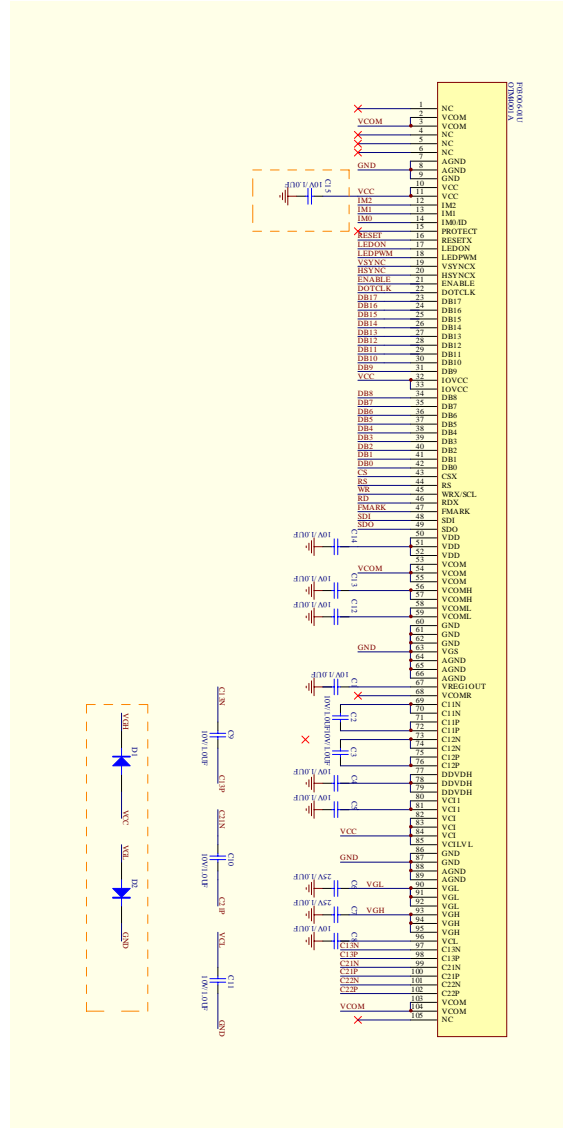
2. CMO3.0" APPLICATION NOTE

2.1. CMO3.0" Application Circuit

F03006-01U

IM2	IM1	IM0/ ID	Interface Mode	DB Pin	Colors
0	0	0	80-system 18-bit Interface	DB17-0	262,144
0	0	1	80-system 9-bit interface	DB17-9	262,144
0	1	0	80-system 16-bit Interface	DB17-10, DB8-1	262,144 see Note 1
0	1	1	80-system 8-bit interface	DB17-10	262,144 see Note 2
1	0	*(ID)	Clock synchronous serial interface	-	65,536
1	1	0	Setting disabled	-	-
1	1	1	Setting disabled	-	-

Notes: 1. 65,536 colors in one transfer mode
2. 65,536 colors in two transfers mode



Note(*)::C15,D1 ,D2

are not used, but Pls. Keep these pad on FPC.

Initial Code

2.1.1. CMO_3.0* initial code

```

void Code( )
{
//----- Power supply condition: VCC=IOVCC=VCI=2.8V -----//
//-----Power on sequence-----//
write_ctrl(0x0606);      write_data(0x0000);
delayms(100);
write_ctrl(0x0007);      write_data(0x0001);
delayms(5);
write_ctrl(0x0110);      write_data(0x0001);
delayms(5);
write_ctrl(0x0100);      write_data(0x17B0);
write_ctrl(0x0101);      write_data(0x0147);
write_ctrl(0x0102);      write_data(0x019D);
write_ctrl(0x0103);      write_data(0x3600);
write_ctrl(0x0281);      write_data(0x0010);
delayms(5);
write_ctrl(0x0102);      write_data(0x01BD);
delayms(5);
write_ctrl(0x0000);      write_data(0x0000);
write_ctrl(0x0001);      write_data(0x0100);
write_ctrl(0x0002);      write_data(0x0100);
write_ctrl(0x0003);      write_data(0x10B0);
write_ctrl(0x0006);      write_data(0x0000);
write_ctrl(0x0008);      write_data(0x0503);
write_ctrl(0x0009);      write_data(0x0001);
write_ctrl(0x000B);      write_data(0x0010);
write_ctrl(0x000C);      write_data(0x0000);
write_ctrl(0x000F);      write_data(0x0000);
write_ctrl(0x0007);      write_data(0x0001);
//-----Panel interface control-----//
write_ctrl(0x0010);      write_data(0x0014);
write_ctrl(0x0011);      write_data(0x0202);
write_ctrl(0x0012);      write_data(0x0300);
write_ctrl(0x0020);      write_data(0x021E);
write_ctrl(0x0021);      write_data(0x0202);
write_ctrl(0x0090);      write_data(0x8000);
write_ctrl(0x0092);      write_data(0x0000);
//-----Voltage setting-----//
write_ctrl(0x0100);      write_data(0x13b0);
delayms(10);
write_ctrl(0x0101);      write_data(0x0147);
delayms(10);
write_ctrl(0x0102);      write_data(0x018D);
delayms(10);
write_ctrl(0x0103);      write_data(0x0200);
delayms(10);
write_ctrl(0x0107);      write_data(0x0000);

```



```

delaysms(10);
write_ctrl(0x0110);      write_data(0x0001);
delaysms(10);
write_ctrl(0x0210);      write_data(0x0000);
write_ctrl(0x0211);      write_data(0x00ef);
write_ctrl(0x0212);      write_data(0x0000);
write_ctrl(0x0213);      write_data(0x018f);
write_ctrl(0x0200);      write_data(0x0000);
write_ctrl(0x0201);      write_data(0x0000);
write_ctrl(0x0280);      write_data(0x0000);
write_ctrl(0x0281);      write_data(0x0000);
write_ctrl(0x0282);      write_data(0x0000);
delaysms(10);
//-----Gamma setting-----//
write_ctrl(0x0300);      write_data(0x0010);
write_ctrl(0x0301);      write_data(0x0237);
write_ctrl(0x0302);      write_data(0x043a);
write_ctrl(0x0303);      write_data(0x3b04);
write_ctrl(0x0304);      write_data(0x3702);
write_ctrl(0x0305);      write_data(0x1000);
write_ctrl(0x0306);      write_data(0x1403);
write_ctrl(0x0307);      write_data(0x0314);
write_ctrl(0x0308);      write_data(0x0304);
write_ctrl(0x0309);      write_data(0x0304);
write_ctrl(0x030A);      write_data(0x0D06);
write_ctrl(0x030B);      write_data(0x0E02);
write_ctrl(0x030C);      write_data(0x020E);
write_ctrl(0x030D);      write_data(0x060D);
write_ctrl(0x030E);      write_data(0x0403);
write_ctrl(0x030F);      write_data(0x0403);
//-----//
write_ctrl(0x0400);      write_data(0x3500);
write_ctrl(0x0401);      write_data(0x0001);
write_ctrl(0x0404);      write_data(0x0000);
write_ctrl(0x0500);      write_data(0x0000);
write_ctrl(0x0501);      write_data(0x0000);
write_ctrl(0x0502);      write_data(0x0000);
write_ctrl(0x0503);      write_data(0x0000);
write_ctrl(0x0504);      write_data(0x0000);
write_ctrl(0x0505);      write_data(0x0000);
write_ctrl(0x0600);      write_data(0x0000);
write_ctrl(0x0606);      write_data(0x0000);
write_ctrl(0x06F0);      write_data(0x0000);
//-----Orise mode -----//
write_ctrl(0x07F0);      write_data(0x5420);
write_ctrl(0x07F3);      write_data(0x288b);
write_ctrl(0x07F4);      write_data(0x0022);
write_ctrl(0x07F5);      write_data(0x0011);
write_ctrl(0x07F0);      write_data(0x0000);

```

```
write_ctrl(0x0007);      write_data(0x0173);
delayms(5);
write_ctrl(0x0007);      write_data(0x0061);
delayms(5);
write_ctrl(0x0007);      write_data(0x0173);
delayms(500);
```

```
void Pattern_RED( )
{
Signed int j;
write_ctrl(0x0210);      write_data(0x0000); // xstart
write_ctrl(0x0211);      write_data(0x00EF); // xend
write_ctrl(0x0212);      write_data(0x0000); // ystart
write_ctrl(0x0213);      write_data(0x018F); // yend
write_ctrl(0x0200);      write_data(0x0000); // xstart
write_ctrl(0x0201);      write_data(0x0000); // ystart
write_ctrl(0x0202); // write GRAM
for( j =0; j<240*400; j++ )
{
Write_ Data (0xFC,0x00, 0x00); // red
}
}
```

CPU Interface enter sleep mode code

```
void Code( )
{
write_ctrl(0x07F0);      write_data(0x5420);
write_ctrl(0x07DE);      write_data(0x0C00);
write_ctrl(0x07F0);      write_data(0x0000);
write_ctrl(0x0007);      write_data(0x0000); //Display off
delay_ms(50);
write_ctrl(0x0100);      write_data(0x0002); //Set to sleep mode
delayms(500);
}
```

CPU Interface exit sleep and deep stand by mode code

```
void code( )
{
write_ctrl(0x07F0);      write_data(0x5420);
write_ctrl(0x07DE);      write_data(0x0400);
write_ctrl(0x07F0);      write_data(0x0000);
write_ctrl(0x0100);      write_data(0x13B0); // Release state
delay_ms(50);
write_ctrl(0x0007);      write_data(0x0173); // Display on
delayms(500);
}
```

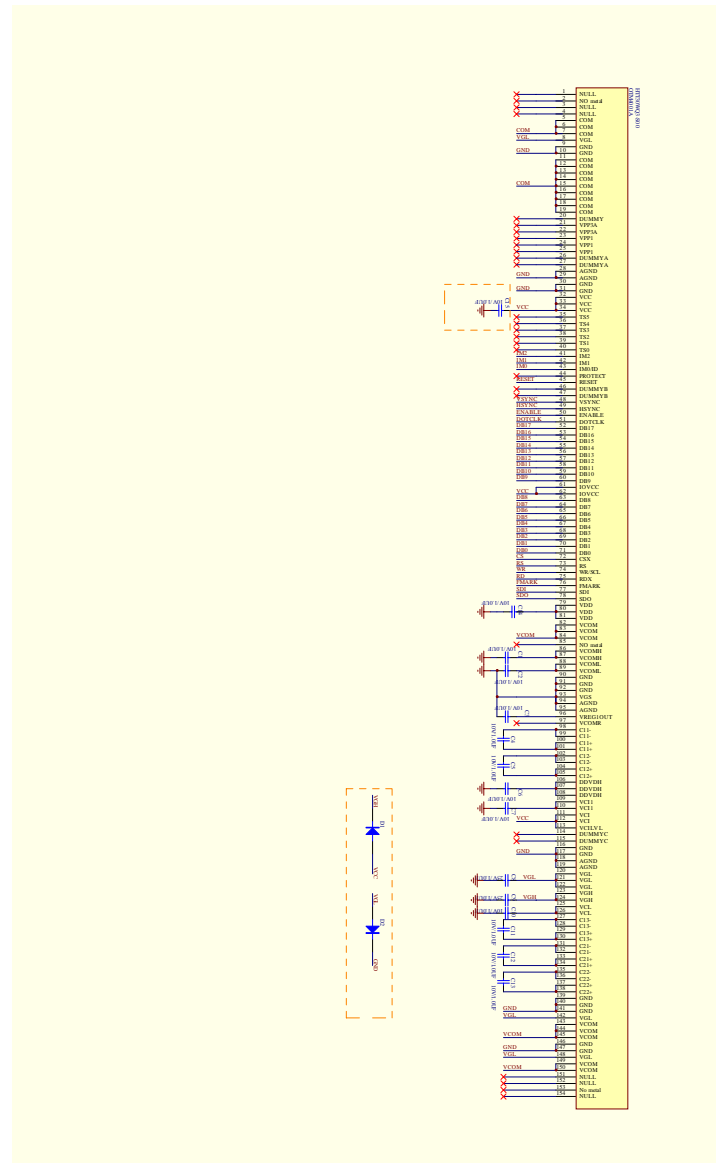
3. HTT3.0” APPLICATION NOTE

3.1. Application Circuit

HTT30WQ3-D00

IM2	IM1	IM0/ ID	Interface Mode	DB Pin	Colors
0	0	0	80-system 18-bit Interface	DB17-0	262,144
0	0	1	80-system 9-bit interface	DB17-9	262,144
0	1	0	80-system 18-bit Interface	DB17-10, DB8-1	262,144 see Note 1
0	1	1	80-system 8-bit interface	DB17-10	262,144 see Note 2
1	0	*(ID)	Clock synchronous serial interface	-	65,536
1	1	0	Setting disabled	-	-
1	1	1	Setting disabled	-	-

Notes: 1. 65,536 colors in one transfer mode
2. 65,536 colors in two transfers mode



Note(*)::C15,D1 ,D2

are not used, but Pls. Keep these pad on FPC.

Initial Code

3.1.1. HTT_3.0" initial code

```
void Code( )
{
//----- Power supply condition: VCC=IOVCC=VCI=2.8V -----//
//-----Power on sequence-----//
write_ctrl(0x0606);      write_data(0x0000);
delayms(100);
write_ctrl(0x0007);      write_data(0x0001);
delayms(5);
write_ctrl(0x0110);      write_data(0x0001);
delayms(5);
write_ctrl(0x0100);      write_data(0x17B0);
write_ctrl(0x0101);      write_data(0x0147);
write_ctrl(0x0102);      write_data(0x019D);
write_ctrl(0x0103);      write_data(0x3600);
write_ctrl(0x0281);      write_data(0x0010);
delayms(5);
write_ctrl(0x0102);      write_data(0x01BD);
delayms(5);
write_ctrl(0x0000);      write_data(0x0000);
write_ctrl(0x0001);      write_data(0x0000);
write_ctrl(0x0002);      write_data(0x0100);
write_ctrl(0x0003);      write_data(0x10A0);
write_ctrl(0x0006);      write_data(0x0000);
write_ctrl(0x0008);      write_data(0x0503);
write_ctrl(0x0009);      write_data(0x0001);
write_ctrl(0x000B);      write_data(0x0010);
write_ctrl(0x000C);      write_data(0x0000);
write_ctrl(0x000F);      write_data(0x0000);
write_ctrl(0x0007);      write_data(0x0001);
//-----Panel interface control-----//
write_ctrl(0x0010);      write_data(0x0010);
write_ctrl(0x0011);      write_data(0x0202);
write_ctrl(0x0012);      write_data(0x0300);
write_ctrl(0x0020);      write_data(0x021e);
write_ctrl(0x0021);      write_data(0x0202);
write_ctrl(0x0090);      write_data(0x8000);
write_ctrl(0x0092);      write_data(0x0000);
//-----Voltage setting-----//
write_ctrl(0x0100);      write_data(0x13b0);
delayms(10);
write_ctrl(0x0101);      write_data(0x0147);
delayms(10);
write_ctrl(0x0102);      write_data(0x018D);
delayms(10);
write_ctrl(0x0103);      write_data(0x2500);
delayms(10);
```

```
write_ctrl(0x0107);    write_data(0x0000);
delayms(10);
write_ctrl(0x0110);    write_data(0x0001);
delayms(10);
//-----//

write_ctrl(0x0210);    write_data(0x0000);
write_ctrl(0x0211);    write_data(0x00ef);
write_ctrl(0x0212);    write_data(0x0000);
write_ctrl(0x0213);    write_data(0x018f);
write_ctrl(0x0200);    write_data(0x0000);
write_ctrl(0x0201);    write_data(0x0000);
write_ctrl(0x0280);    write_data(0x0000);
write_ctrl(0x0281);    write_data(0x0000);
write_ctrl(0x0282);    write_data(0x0000);
delayms(10);
//-----Gamma setting-----//

write_ctrl(0x0300);    write_data(0x0310);
write_ctrl(0x0301);    write_data(0x052f);
write_ctrl(0x0302);    write_data(0x0632);
write_ctrl(0x0303);    write_data(0x3206);
write_ctrl(0x0304);    write_data(0x2105);
write_ctrl(0x0305);    write_data(0x1003);
write_ctrl(0x0306);    write_data(0x1e02);
write_ctrl(0x0307);    write_data(0x021e);
write_ctrl(0x0308);    write_data(0x0805);
write_ctrl(0x0309);    write_data(0x0902);
write_ctrl(0x030A);    write_data(0x0D06);
write_ctrl(0x030B);    write_data(0x0502);
write_ctrl(0x030C);    write_data(0x020D);
write_ctrl(0x030D);    write_data(0x060D);
write_ctrl(0x030E);    write_data(0x0209);
write_ctrl(0x030F);    write_data(0x0508);
//-----//

write_ctrl(0x0400);    write_data(0x3500);
write_ctrl(0x0401);    write_data(0x0001);
write_ctrl(0x0404);    write_data(0x0000);
write_ctrl(0x0500);    write_data(0x0000);
write_ctrl(0x0501);    write_data(0x0000);
write_ctrl(0x0502);    write_data(0x0000);
write_ctrl(0x0503);    write_data(0x0000);
write_ctrl(0x0504);    write_data(0x0000);
write_ctrl(0x0505);    write_data(0x0000);
write_ctrl(0x0600);    write_data(0x0000);
write_ctrl(0x0606);    write_data(0x0000);
write_ctrl(0x06F0);    write_data(0x0000);
//-----Orise mode -----//

write_ctrl(0x07F0);    write_data(0x5420);
write_ctrl(0x07F3);    write_data(0x288D);
write_ctrl(0x07F4);    write_data(0x0022);
```

```
write_ctrl(0x07F5);      write_data(0x0061);
write_ctrl(0x07F0);      write_data(0x0000);
write_ctrl(0x0007);      write_data(0x0173);
delayms(5);
write_ctrl(0x0007);      write_data(0x0061);
delayms(5);
write_ctrl(0x0007);      write_data(0x0173);
delayms(500);
```

```
void Pattern_RED( )
{
    Signed int j;
    write_ctrl(0x0210);      write_data(0x0000); // xstart
    write_ctrl(0x0211);      write_data(0x00EF); // xend
    write_ctrl(0x0212);      write_data(0x0000); // ystart
    write_ctrl(0x0213);      write_data(0x018F); // yend
    write_ctrl(0x0200);      write_data(0x0000); // xstart
    write_ctrl(0x0201);      write_data(0x0000); // ystart
    write_ctrl(0x0202); // write GRAM
    for( j =0; j<240*400; j++ )
    {
        Write_ Data (0xFC,0x00, 0x00); // red
    }
}
```

CPU Interface enter sleep mode code

```
void Code( )
{
    write_ctrl(0x07F0);      write_data(0x5420);
    write_ctrl(0x07DE);      write_data(0x0C00);
    write_ctrl(0x07F0);      write_data(0x0000);
    write_ctrl(0x0007);      write_data(0x0000); //Display off
    delay_ms(50);
    write_ctrl(0x0100);      write_data(0x0002); //Set to sleep mode
    delayms(500);
}
```

CPU Interface exit sleep and deep stand by mode code

```
void code( )
{
    write_ctrl(0x07F0);      write_data(0x5420);
    write_ctrl(0x07DE);      write_data(0x0400);
    write_ctrl(0x07F0);      write_data(0x0000);
    write_ctrl(0x0100);      write_data(0x13B0); // Release state
    delay_ms(50);
    write_ctrl(0x0007);      write_data(0x0173); // Display on
    delayms(500);
}
```

4. GPT2.5" 2.8" 3.0" APPLICATION NOTE

4.1. GPT2.5" Application Circuit

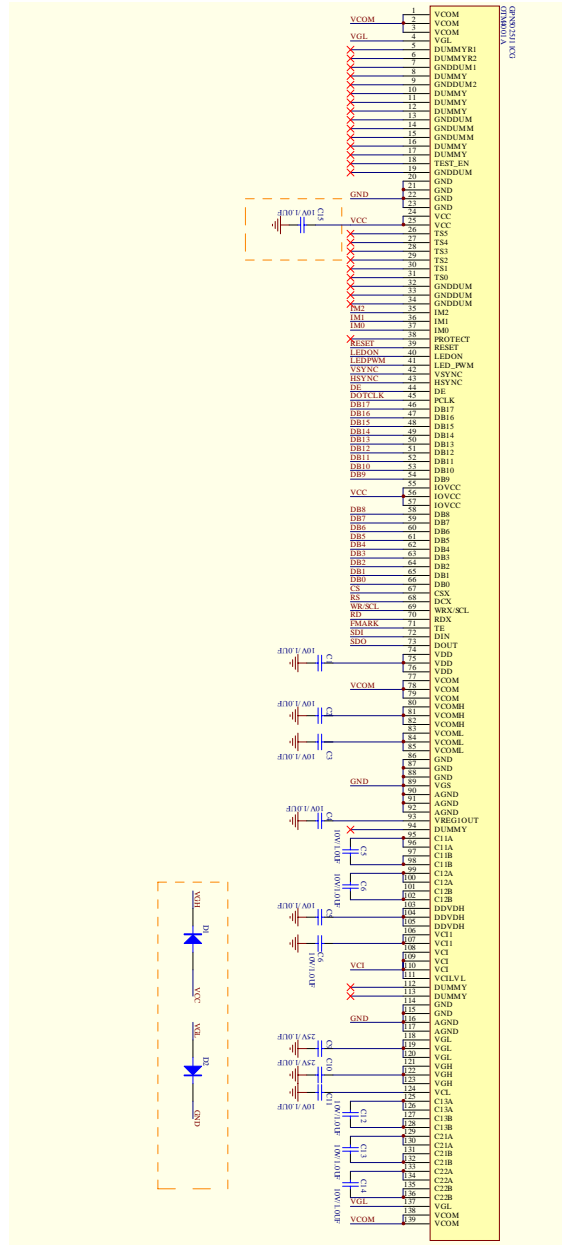
GPN5025J11CG

IM2	IM1	IM0/ ID	Interface Mode	DB Pin	Colors
0	0	0	80-system 18-bit interface	DB17-0	282,144
0	0	1	80-system 9-bit interface	DB17-9	282,144
0	1	0	80-system 18-bit interface	DB17-10, DB8-1	282,144 see Note 1
0	1	1	80-system 8-bit interface	DB17-10	282,144 see Note 2
1	0	*(ID)	Clock synchronous serial interface	-	65,536
1	1	0	Setting disabled	-	-
1	1	1	Setting disabled	-	-

Notes: 1. 65,536 colors in one transfer mode
2. 65,536 colors in two transfers mode

Note(*): C15, D1, D2

are not used, but Pls. Keep these pad on FPC.



4.3. GPT3.0⁺ Application Circuit

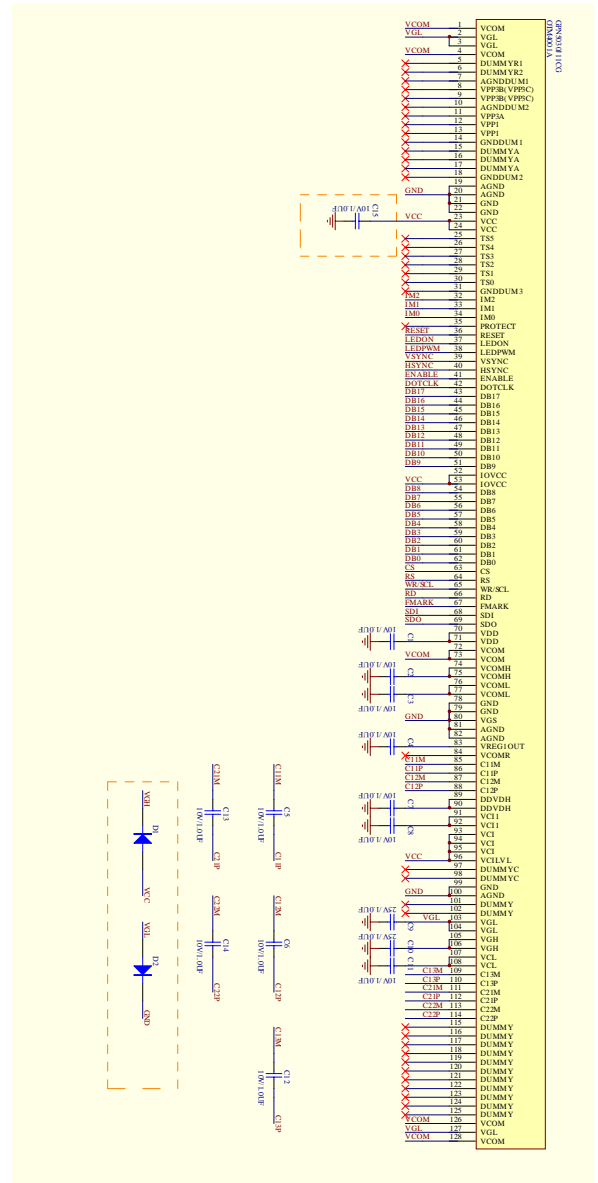
GPN5030J11CG

IM2	IM1	IM0/ I D	Interface Mode	DB Pin	Colors
0	0	0	80-system 18-bit interface	DB17-0	262,144
0	0	1	80-system 9-bit interface	DB17-9	262,144
0	1	0	80-system 18-bit interface	DB17-10, DB8-1	262,144 see Note 1
0	1	1	80-system 8-bit interface	DB17-10	262,144 see Note 2
1	0	*(ID)	Clock synchronous serial interface	-	65,536
1	1	0	Setting disabled	-	-
1	1	1	Setting disabled	-	-

Notes: 1. 65,536 colors in one transfer mode
2. 65,536 colors in two transfers mode

Note(*)::C15,D1 ,D2

are not used, but Pls. Keep these pad on FPC.



Initial Code

4.3.1. GPT2.5" initial code

```

void Code( )
{
//----- Power supply condition: VCC=IOVCC=VCI=2.8V -----//
//-----Power on sequence-----//
    write_ctr(0x0606);        write_data(0x0000);
    delayms(100);
    write_ctr(0x0007);        write_data(0x0001);
    delayms(5);
    write_ctr(0x0110);        write_data(0x0001);
    delayms(5);
    write_ctr(0x0100);        write_data(0x17B0);
    write_ctr(0x0101);        write_data(0x0147);
    write_ctr(0x0102);        write_data(0x019D);
    write_ctr(0x0103);        write_data(0x3600);
    write_ctr(0x0281);        write_data(0x0010);
    delayms(5);
    write_ctr(0x0102);        write_data(0x01BD);
    delayms(5);
    write_ctr(0x0000);        write_data(0x0000);
    write_ctr(0x0001);        write_data(0x0100);
    write_ctr(0x0002);        write_data(0x0100);
    write_ctr(0x0003);        write_data(0x10B0);
    write_ctr(0x0006);        write_data(0x0000);
    write_ctr(0x0008);        write_data(0x0202);
    write_ctr(0x0009);        write_data(0x0001);
    write_ctr(0x000B);        write_data(0x0010);
    write_ctr(0x000C);        write_data(0x0000);
    write_ctr(0x000F);        write_data(0x0000);
    write_ctr(0x0007);        write_data(0x0001);
//-----Panel interface control-----
    write_ctr(0x0010);        write_data(0x0014);
    write_ctr(0x0011);        write_data(0x0301);
    write_ctr(0x0012);        write_data(0x0300);
    write_ctr(0x0020);        write_data(0x021E);
    write_ctr(0x0021);        write_data(0x0202);
    write_ctr(0x0090);        write_data(0x8000);
    write_ctr(0x0092);        write_data(0x0000);
//-----Voltage setting-----
    write_ctr(0x0100);        write_data(0x12b0);
    delayms(10);
    write_ctr(0x0101);        write_data(0x0147);
    delayms(10);
    write_ctr(0x0102);        write_data(0x01BE);
    delayms(10);
    write_ctr(0x0103);        write_data(0x2900);
    delayms(10);
    write_ctr(0x0107);        write_data(0x0000);

```

```
delaysms(10);
write_ctr(0x0110);    write_data(0x0001);
delaysms(10);
write_ctr(0x0210);    write_data(0x0000);
write_ctr(0x0211);    write_data(0x00ef);
write_ctr(0x0212);    write_data(0x0000);
write_ctr(0x0213);    write_data(0x018f);
write_ctr(0x0200);    write_data(0x0000);
write_ctr(0x0201);    write_data(0x0000);
write_ctr(0x0280);    write_data(0x0000);
write_ctr(0x0281);    write_data(0x0004);
write_ctr(0x0282);    write_data(0x0000);
delaysms(10);
```

```
//-----Gamma setting-----
```

```
write_ctr(0x0300);    write_data(0x0101);
write_ctr(0x0301);    write_data(0x092e);
write_ctr(0x0302);    write_data(0x0F2b);
write_ctr(0x0303);    write_data(0x2e0F);
write_ctr(0x0304);    write_data(0x2709);
write_ctr(0x0305);    write_data(0x0101);
write_ctr(0x0306);    write_data(0x1802);
write_ctr(0x0307);    write_data(0x0318);
write_ctr(0x0308);    write_data(0x0607);
write_ctr(0x0309);    write_data(0x0406);
write_ctr(0x030A);    write_data(0x0E07);
write_ctr(0x030B);    write_data(0x0E00);
write_ctr(0x030C);    write_data(0x000E);
write_ctr(0x030D);    write_data(0x070E);
write_ctr(0x030E);    write_data(0x0604);
write_ctr(0x030F);    write_data(0x0706);
```

```
//-----//
```

```
write_ctr(0x0400);    write_data(0x3500);
write_ctr(0x0401);    write_data(0x0001);
write_ctr(0x0404);    write_data(0x0000);
write_ctr(0x0500);    write_data(0x0000);
write_ctr(0x0501);    write_data(0x0000);
write_ctr(0x0502);    write_data(0x0000);
write_ctr(0x0503);    write_data(0x0000);
write_ctr(0x0504);    write_data(0x0000);
write_ctr(0x0505);    write_data(0x0000);
write_ctr(0x0600);    write_data(0x0000);
write_ctr(0x0606);    write_data(0x0000);
write_ctr(0x06F0);    write_data(0x0000);
```

```
//-----Orise mode -----//
```

```
write_ctr(0x07F0);    write_data(0x5420);
write_ctr(0x07F3);    write_data(0x280F);
write_ctr(0x07F4);    write_data(0x0022);
write_ctr(0x07F5);    write_data(0x0001);
```

```
write_ctrl(0x07F0);      write_data(0x0000);
write_ctrl(0x0007);      write_data(0x0173);
delayms(5);
write_ctrl(0x0007);      write_data(0x0061);
delayms(5);
write_ctrl(0x0007);      write_data(0x0173);
delayms(500);

}

void Pattern_RED( )
{
    Signed int j;
    write_ctrl(0x0210);      write_data(0x0000); // xstart
    write_ctrl(0x0211);      write_data(0x00EF); // xend
    write_ctrl(0x0212);      write_data(0x0000); // ystart
    write_ctrl(0x0213);      write_data(0x018F); // yend
    write_ctrl(0x0200);      write_data(0x0000); // xstart
    write_ctrl(0x0201);      write_data(0x0000); // ystart
    write_ctrl(0x0202); // write GRAM
for( j =0; j<240*400; j++ )
{
    Write_ Data (0xFC,0x00, 0x00); // red
}
}

CPU Interface enter sleep mode code

void Code( )
{
    write_ctrl(0x07F0);      write_data(0x5420);
    write_ctrl(0x07DE);      write_data(0x0C00);
    write_ctrl(0x07F0);      write_data(0x0000);
    write_ctrl(0x0007);      write_data(0x0000); //Display off
    delay_ms(50);
    write_ctrl(0x0100);      write_data(0x0002); //Set to sleep mode
    delayms(500);
}
}
```

CPU Interface exit sleep and deep stand by mode code

```
void code( )
{
    write_ctrl(0x07F0);      write_data(0x5420);
    write_ctrl(0x07DE);      write_data(0x0400);
    write_ctrl(0x07F0);      write_data(0x0000);
    write_ctrl(0x0100);      write_data(0x12B0); // Release state
    delay_ms(50);
    write_ctrl(0x0007);      write_data(0x0173); // Display on
    delayms(500);
}
}
```

4.3.2. GPT2.8" initial code

```

void Code( )
{
//----- Power supply condition: VCC=IOVCC=VCI=2.8V -----//
//-----Power on sequence-----//
    write_ctrl(0x0606);        write_data(0x0000);
    delaysms(100);
    write_ctrl(0x0007);        write_data(0x0001);
    delaysms(5);
    write_ctrl(0x0110);        write_data(0x0001);
    delaysms(5);
    write_ctrl(0x0100);        write_data(0x17B0);
    write_ctrl(0x0101);        write_data(0x0147);
    write_ctrl(0x0102);        write_data(0x019D);
    write_ctrl(0x0103);        write_data(0x3600);
    write_ctrl(0x0281);        write_data(0x0010);
    delaysms(5);
    write_ctrl(0x0102);        write_data(0x01BD);
    delaysms(5);
    write_ctrl(0x0000);        write_data(0x0000);
    write_ctrl(0x0001);        write_data(0x0100);
    write_ctrl(0x0002);        write_data(0x0100);
    write_ctrl(0x0003);        write_data(0x10B0);
    write_ctrl(0x0006);        write_data(0x0000);
    write_ctrl(0x0008);        write_data(0x0202);
    write_ctrl(0x0009);        write_data(0x0001);
    write_ctrl(0x000B);        write_data(0x0010);
    write_ctrl(0x000C);        write_data(0x0000);
    write_ctrl(0x000F);        write_data(0x0000);
    write_ctrl(0x0007);        write_data(0x0001);
//-----Panel interface control-----//
    write_ctrl(0x0010);        write_data(0x0011);
    write_ctrl(0x0011);        write_data(0x0301);
    write_ctrl(0x0012);        write_data(0x0300);
    write_ctrl(0x0020);        write_data(0x021E);
    write_ctrl(0x0021);        write_data(0x0202);
    write_ctrl(0x0090);        write_data(0x8000);
    write_ctrl(0x0092);        write_data(0x0000);
//-----Voltage setting-----//
    write_ctrl(0x0100);        write_data(0x12b0);
    delaysms(10);
    write_ctrl(0x0101);        write_data(0x0147);
    delaysms(10);
    write_ctrl(0x0102);        write_data(0x01BE);
    delaysms(10);
    write_ctrl(0x0103);        write_data(0x2B00);
    delaysms(10);
    write_ctrl(0x0107);        write_data(0x0000);
    delaysms(10);

```

```
write_ctrl(0x0110);    write_data(0x0001);
delayms(10);
write_ctrl(0x0210);    write_data(0x0000);
write_ctrl(0x0211);    write_data(0x00ef);
write_ctrl(0x0212);    write_data(0x0000);
write_ctrl(0x0213);    write_data(0x018f);
write_ctrl(0x0200);    write_data(0x0000);
write_ctrl(0x0201);    write_data(0x0000);
write_ctrl(0x0280);    write_data(0x0000);
write_ctrl(0x0281);    write_data(0x0007);
write_ctrl(0x0282);    write_data(0x0000);
delayms(10);
```

//-----Gamma setting-----

```
write_ctrl(0x0300);    write_data(0x0101);
write_ctrl(0x0301);    write_data(0x0929);
write_ctrl(0x0302);    write_data(0x0F2c);
write_ctrl(0x0303);    write_data(0x2c0F);
write_ctrl(0x0304);    write_data(0x2909);
write_ctrl(0x0305);    write_data(0x0101);
write_ctrl(0x0306);    write_data(0x1904);
write_ctrl(0x0307);    write_data(0x0419);
write_ctrl(0x0308);    write_data(0x0605);
write_ctrl(0x0309);    write_data(0x0403);
write_ctrl(0x030A);    write_data(0x0E06);
write_ctrl(0x030B);    write_data(0x0E00);
write_ctrl(0x030C);    write_data(0x000E);
write_ctrl(0x030D);    write_data(0x060E);
write_ctrl(0x030E);    write_data(0x0304);
write_ctrl(0x030F);    write_data(0x0506);
```

//-----//

```
write_ctrl(0x0400);    write_data(0x3500);
write_ctrl(0x0401);    write_data(0x0001);
write_ctrl(0x0404);    write_data(0x0000);
write_ctrl(0x0500);    write_data(0x0000);
write_ctrl(0x0501);    write_data(0x0000);
write_ctrl(0x0502);    write_data(0x0000);
write_ctrl(0x0503);    write_data(0x0000);
write_ctrl(0x0504);    write_data(0x0000);
write_ctrl(0x0505);    write_data(0x0000);
write_ctrl(0x0600);    write_data(0x0000);
write_ctrl(0x0606);    write_data(0x0000);
write_ctrl(0x06F0);    write_data(0x0000);
```

//-----Orise mode -----//

```
write_ctrl(0x07F0);    write_data(0x5420);
write_ctrl(0x07F3);    write_data(0x280F);
write_ctrl(0x07F4);    write_data(0x0022);
write_ctrl(0x07F5);    write_data(0x0001);
write_ctrl(0x07F0);    write_data(0x0000);
write_ctrl(0x0007);    write_data(0x0173);
```

```
    delaysms(5);
    write_ctrl(0x0007);        write_data(0x0061);
    delaysms(5);
    write_ctrl(0x0007);        write_data(0x0173);
    delaysms(500);
}
}
```

```
void Pattern_RED( )
{
    Signed int j;
    write_ctrl(0x0210);        write_data(0x0000); // xstart
    write_ctrl(0x0211);        write_data(0x00EF); // xend
    write_ctrl(0x0212);        write_data(0x0000); // ystart
    write_ctrl(0x0213);        write_data(0x018F); // yend
    write_ctrl(0x0200);        write_data(0x0000); // xstart
    write_ctrl(0x0201);        write_data(0x0000); // ystart
    write_ctrl(0x0202); // write GRAM
for( j =0; j<240*400; j++ )
{
    Write_ Data (0xFC,0x00, 0x00); // red
}
}
}
```

CPU Interface enter sleep mode code

```
void Code( )
{
    write_ctrl(0x07F0);        write_data(0x5420);
    write_ctrl(0x07DE);        write_data(0x0C00);
    write_ctrl(0x07F0);        write_data(0x0000);
    write_ctrl(0x0007);        write_data(0x0000); //Display off
    delay_ms(50);
    write_ctrl(0x0100);        write_data(0x0002); //Set to sleep mode
    delaysms(500);
}
}
```

CPU Interface exit sleep and deep stand by mode code

```
void code( )
{
    write_ctrl(0x07F0);        write_data(0x5420);
    write_ctrl(0x07DE);        write_data(0x0400);
    write_ctrl(0x07F0);        write_data(0x0000);
    write_ctrl(0x0100);        write_data(0x12B0); // Release state
    delay_ms(50);
    write_ctrl(0x0007);        write_data(0x0173); // Display on
    delaysms(500);
}
}
```

4.3.3. GPT3.0" initial code

```

void Code( )
{
//----- Power supply condition: VCC=IOVCC=VCI=2.8V -----//
//-----Power on sequence-----//
    write_ctrl(0x0606);        write_data(0x0000);
    delayms(100);
    write_ctrl(0x0007);        write_data(0x0001);
    delayms(5);
    write_ctrl(0x0110);        write_data(0x0001);
    delayms(5);
    write_ctrl(0x0100);        write_data(0x17B0);
    write_ctrl(0x0101);        write_data(0x0147);
    write_ctrl(0x0102);        write_data(0x019D);
    write_ctrl(0x0103);        write_data(0x3600);
    write_ctrl(0x0281);        write_data(0x0010);
    delayms(5);
    write_ctrl(0x0102);        write_data(0x01BD);
    delayms(5);
    write_ctrl(0x0000);        write_data(0x0000);
    write_ctrl(0x0001);        write_data(0x0100);
    write_ctrl(0x0002);        write_data(0x0100);
    write_ctrl(0x0003);        write_data(0x10B0);
    write_ctrl(0x0006);        write_data(0x0000);
    write_ctrl(0x0008);        write_data(0x0202);
    write_ctrl(0x0009);        write_data(0x0001);
    write_ctrl(0x000B);        write_data(0x0010);
    write_ctrl(0x000C);        write_data(0x0000);
    write_ctrl(0x000F);        write_data(0x0000);
    write_ctrl(0x0007);        write_data(0x0001);
//-----Panel interface control-----
    write_ctrl(0x0010);        write_data(0x0011);
    write_ctrl(0x0011);        write_data(0x0301);
    write_ctrl(0x0012);        write_data(0x0300);
    write_ctrl(0x0020);        write_data(0x021E);
    write_ctrl(0x0021);        write_data(0x0202);
    write_ctrl(0x0090);        write_data(0x8000);
    write_ctrl(0x0092);        write_data(0x0000);
//-----Voltage setting-----
    write_ctrl(0x0100);        write_data(0x12b0);
    delayms(10);
    write_ctrl(0x0101);        write_data(0x0147);
    delayms(10);
    write_ctrl(0x0102);        write_data(0x01BE);
    delayms(10);
    write_ctrl(0x0103);        write_data(0x2B00);
    delayms(10);
    write_ctrl(0x0107);        write_data(0x0000);
    delayms(10);

```



```
write_ctrl(0x0110);    write_data(0x0001);
delayms(10);
write_ctrl(0x0210);    write_data(0x0000);
write_ctrl(0x0211);    write_data(0x00ef);
write_ctrl(0x0212);    write_data(0x0000);
write_ctrl(0x0213);    write_data(0x018f);
write_ctrl(0x0200);    write_data(0x0000);
write_ctrl(0x0201);    write_data(0x0000);
write_ctrl(0x0280);    write_data(0x0000);
write_ctrl(0x0281);    write_data(0x0007);
write_ctrl(0x0282);    write_data(0x0000);
delayms(10);
```

//-----Gamma setting-----//

```
write_ctrl(0x0300);    write_data(0x0101);
write_ctrl(0x0301);    write_data(0x0927);
write_ctrl(0x0302);    write_data(0x0F2a);
write_ctrl(0x0303);    write_data(0x2a0F);
write_ctrl(0x0304);    write_data(0x2709);
write_ctrl(0x0305);    write_data(0x0101);
write_ctrl(0x0306);    write_data(0x1A04);
write_ctrl(0x0307);    write_data(0x041A);
write_ctrl(0x0308);    write_data(0x0605);
write_ctrl(0x0309);    write_data(0x0403);
write_ctrl(0x030A);    write_data(0x0E04);
write_ctrl(0x030B);    write_data(0x0E00);
write_ctrl(0x030C);    write_data(0x000E);
write_ctrl(0x030D);    write_data(0x040E);
write_ctrl(0x030E);    write_data(0x0304);
write_ctrl(0x030F);    write_data(0x0506);
```

//-----//

```
write_ctrl(0x0400);    write_data(0x3500);
write_ctrl(0x0401);    write_data(0x0001);
write_ctrl(0x0404);    write_data(0x0000);
write_ctrl(0x0500);    write_data(0x0000);
write_ctrl(0x0501);    write_data(0x0000);
write_ctrl(0x0502);    write_data(0x0000);
write_ctrl(0x0503);    write_data(0x0000);
write_ctrl(0x0504);    write_data(0x0000);
write_ctrl(0x0505);    write_data(0x0000);
write_ctrl(0x0600);    write_data(0x0000);
write_ctrl(0x0606);    write_data(0x0000);
write_ctrl(0x06F0);    write_data(0x0000);
```

//-----Orise mode -----//

```
write_ctrl(0x07F0);    write_data(0x5420);
write_ctrl(0x07F3);    write_data(0x280F);
write_ctrl(0x07F4);    write_data(0x0022);
write_ctrl(0x07F5);    write_data(0x0001);
write_ctrl(0x07F0);    write_data(0x0000);
```

```

write_ctrl(0x0007);      write_data(0x0173);
delayms(5);
write_ctrl(0x0007);      write_data(0x0061);
delayms(5);
write_ctrl(0x0007);      write_data(0x0173);
delayms(500);

}

```

```

void Pattern_RED( )
{
    Signed int j;
    write_ctrl(0x0210);      write_data(0x0000); // xstart
    write_ctrl(0x0211);      write_data(0x00EF); // xend
    write_ctrl(0x0212);      write_data(0x0000); // ystart
    write_ctrl(0x0213);      write_data(0x018F); // yend
    write_ctrl(0x0200);      write_data(0x0000); // xstart
    write_ctrl(0x0201);      write_data(0x0000); // ystart
    write_ctrl(0x0202); // write GRAM
    for( j =0; j<240*400; j++ )
    {
        Write_ Data (0xFC,0x00, 0x00); // red
    }
}

```

CPU Interface enter sleep mode code

```

void Code( )
{
    write_ctrl(0x07F0);      write_data(0x5420);
    write_ctrl(0x07DE);      write_data(0x0C00);
    write_ctrl(0x07F0);      write_data(0x0000);
    write_ctrl(0x0007);      write_data(0x0000); //Display off
    delay_ms(50);
    write_ctrl(0x0100);      write_data(0x0002); //Set to sleep mode
    delayms(500);
}

```

CPU Interface exit sleep and deep stand by mode code

```

void code( )
{
    write_ctrl(0x07F0);      write_data(0x5420);
    write_ctrl(0x07DE);      write_data(0x0400);
    write_ctrl(0x07F0);      write_data(0x0000);
    write_ctrl(0x0100);      write_data(0x12B0); // Release state
    delay_ms(50);
    write_ctrl(0x0007);      write_data(0x0173); // Display on
    delayms(500);
}

```

5. AUO 3.2" APPLICATION NOTE

5.1. AUO 3.2" Application Circuit

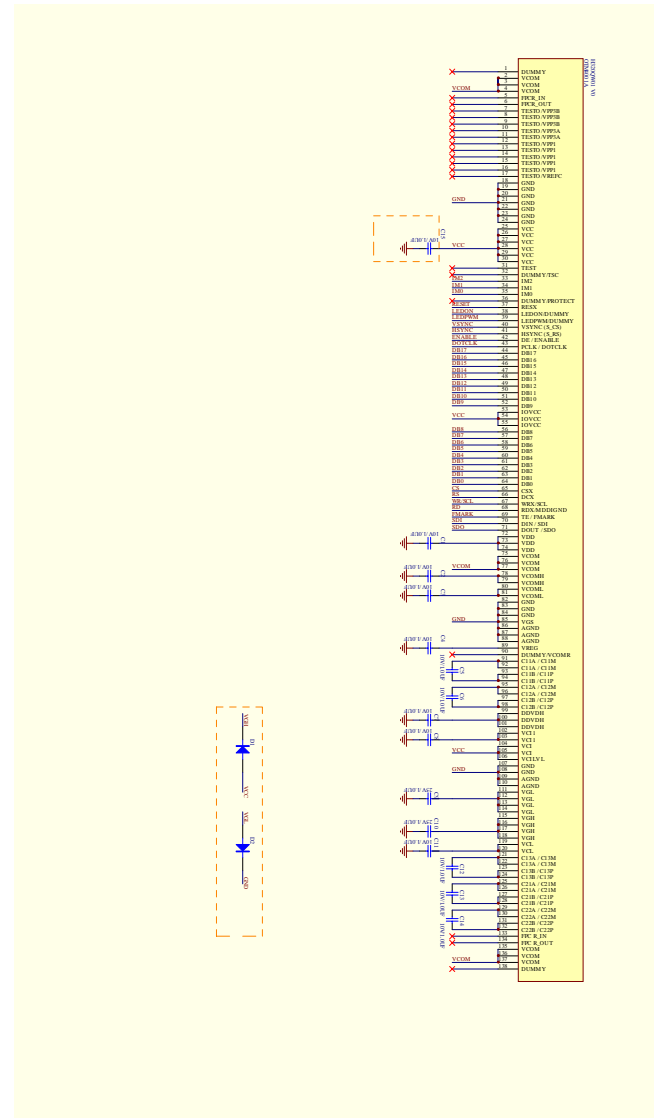
H320QW01 V0

IM2	IM1	IM0/ ID	Interface Mode	DB Pin	Colors
0	0	0	80-system 18-bit interface	DB17-0	282,144
0	0	1	80-system 9-bit interface	DB17-9	282,144
0	1	0	80-system 18-bit interface	DB17-10, DB8-1	282,144 see Note 1
0	1	1	80-system 8-bit interface	DB17-10	282,144 see Note 2
1	0	*(ID)	Clock synchronous serial interface	-	65,536
1	1	0	Setting disabled	-	-
1	1	1	Setting disabled	-	-

Notes: 1. 65,536 colors in one transfer mode
2. 65,536 colors in two transfers mode

Note(*): :C15,D1 ,D2

are not used, but Pls. Keep these pad on FPC.



Initial Code

5.1.1. AUO 3.2" initial code

```

void Code( )
{
//----- Power supply condition: VCC=IOVCC=VCI=2.8V -----//
//-----Power on sequence-----//
write_ctrl(0x0000);      write_data(0x0000);
write_ctrl(0x0001);      write_data(0x0100);
write_ctrl(0x0002);      write_data(0x0100);
write_ctrl(0x0003);      write_data(0x10B0);
write_ctrl(0x0006);      write_data(0x0000);
write_ctrl(0x0008);      write_data(0x0503);
write_ctrl(0x0009);      write_data(0x0001);
write_ctrl(0x000B);      write_data(0x0010);
write_ctrl(0x000C);      write_data(0x0000);
write_ctrl(0x000F);      write_data(0x0000);
write_ctrl(0x0007);      write_data(0x0001);
//-----Panel interface control-----//
write_ctrl(0x0010);      write_data(0x0014);
write_ctrl(0x0011);      write_data(0x0202);
write_ctrl(0x0012);      write_data(0x0300);
write_ctrl(0x0020);      write_data(0x021e);
write_ctrl(0x0021);      write_data(0x0202);
write_ctrl(0x0090);      write_data(0x8000);
write_ctrl(0x0092);      write_data(0x0000);
//-----Voltage setting-----//
write_ctrl(0x0100);      write_data(0x12b0);
delayms(10);
write_ctrl(0x0101);      write_data(0x0147);
delayms(10);
write_ctrl(0x0102);      write_data(0x010E);
delayms(10);
write_ctrl(0x0103);      write_data(0x0200);
delayms(10);
//-----//
write_ctrl(0x0210);      write_data(0x0000);
write_ctrl(0x0211);      write_data(0x00ef);
write_ctrl(0x0212);      write_data(0x0000);
write_ctrl(0x0213);      write_data(0x018f);
write_ctrl(0x0200);      write_data(0x0000);
write_ctrl(0x0201);      write_data(0x0000);

write_ctrl(0x0280);      write_data(0x0000);
write_ctrl(0x0281);      write_data(0x0001);
write_ctrl(0x0282);      write_data(0x0000);
delayms(10);
//-----Gamma setting-----//
write_ctrl(0x0300);      write_data(0x0010);
write_ctrl(0x0301);      write_data(0x0235);

```

```

write_ctrl(0x0302);      write_data(0x0437);
write_ctrl(0x0303);      write_data(0x3704);
write_ctrl(0x0304);      write_data(0x3502);
write_ctrl(0x0305);      write_data(0x1000);
write_ctrl(0x0306);      write_data(0x1603);
write_ctrl(0x0307);      write_data(0x0316);
write_ctrl(0x0308);      write_data(0x0306);
write_ctrl(0x0309);      write_data(0x0303);
write_ctrl(0x030A);      write_data(0x0D04);
write_ctrl(0x030B);      write_data(0x0F00);
write_ctrl(0x030C);      write_data(0x000F);
write_ctrl(0x030D);      write_data(0x040D);
write_ctrl(0x030E);      write_data(0x0303);
write_ctrl(0x030F);      write_data(0x0603);
//-----//

```

```

write_ctrl(0x0400);      write_data(0x3500);
write_ctrl(0x0401);      write_data(0x0001);
write_ctrl(0x0404);      write_data(0x0000);
write_ctrl(0x0600);      write_data(0x0000);
write_ctrl(0x0606);      write_data(0x0000);
write_ctrl(0x06F0);      write_data(0x0000);
//----- Orise mode -----//

```

```

write_ctrl(0x07F0);      write_data(0x5420);
write_ctrl(0x07F3);      write_data(0x288A);
write_ctrl(0x07F4);      write_data(0x0022);
write_ctrl(0x07F5);      write_data(0x0001);
write_ctrl(0x07F0);      write_data(0x0000);

```

```

write_ctrl(0x0007);      write_data(0x0173);
delayms(500);
}
void Pattern_RED( )
{
    Signed int j;
write_ctrl(0x0210);      write_data(0x0000); // xstart
write_ctrl(0x0211);      write_data(0x00EF); // xend
write_ctrl(0x0212);      write_data(0x0000); // ystart
write_ctrl(0x0213);      write_data(0x018F); // yend
write_ctrl(0x0200);      write_data(0x0000); // xstart
write_ctrl(0x0201);      write_data(0x0000); // ystart
write_ctrl(0x0202); // write GRAM
for( j =0; j<240*400; j++ )
{
Write_ Data (0xFC,0x00, 0x00); // red
}
}

```

CPU Interface enter sleep mode code

```

void Code( )
{
write_ctrl(0x07F0);      write_data(0x5420);
write_ctrl(0x07DE);      write_data(0x0C00);
}

```

```
write_ctrl(0x07F0);      write_data(0x0000);
write_ctrl(0x0007);      write_data(0x0000); //Display off
delay_ms(50);
write_ctrl(0x0100);      write_data(0x0002); //Set to sleep mode
delayms(500);
}
```

CPU Interface exit sleep and deep stand by mode code

```
void code( )
{
write_ctrl(0x07F0);      write_data(0x5420);
write_ctrl(0x07DE);      write_data(0x0400);
write_ctrl(0x07F0);      write_data(0x0000);
write_ctrl(0x0100);      write_data(0x12B0); // Release state
delay_ms(50);
write_ctrl(0x0007);      write_data(0x0173); // Display on
delayms(500);
}
```


Initial Code

6.1.1. BOE 2.8" initial code

```
void Code( )
{
//----- Power supply condition: VCC=IOVCC=VCI=2.8V -----//
//-----Power on sequence-----//
    write_ctrl(0x0606);        write_data(0x0000);
    delayms(100);
    write_ctrl(0x0007);        write_data(0x0001);
    delayms(5);
    write_ctrl(0x0110);        write_data(0x0001);
    delayms(5);
    write_ctrl(0x0100);        write_data(0x17B0);
    write_ctrl(0x0101);        write_data(0x0147);
    write_ctrl(0x0102);        write_data(0x019D);
    write_ctrl(0x0103);        write_data(0x3600);
    write_ctrl(0x0281);        write_data(0x0010);
    delayms(5);
    write_ctrl(0x0102);        write_data(0x01BD);
    delayms(5);
    write_ctrl(0x0000);        write_data(0x0000);
    write_ctrl(0x0001);        write_data(0x0100);
    write_ctrl(0x0002);        write_data(0x0100);
    write_ctrl(0x0003);        write_data(0x10B0);
    write_ctrl(0x0006);        write_data(0x0000);
    write_ctrl(0x0008);        write_data(0x0202);
    write_ctrl(0x0009);        write_data(0x0001);
    write_ctrl(0x000B);        write_data(0x0010);
    write_ctrl(0x000C);        write_data(0x0000);
    write_ctrl(0x000F);        write_data(0x0000);
    write_ctrl(0x0007);        write_data(0x0001);
//-----Panel interface control-----//
    write_ctrl(0x0010);        write_data(0x0010);
    write_ctrl(0x0011);        write_data(0x0301);
    write_ctrl(0x0012);        write_data(0x0300);
    write_ctrl(0x0020);        write_data(0x021E);
    write_ctrl(0x0021);        write_data(0x0202);
    write_ctrl(0x0090);        write_data(0x8000);
    write_ctrl(0x0092);        write_data(0x0000);
//-----Voltage setting-----//
    write_ctrl(0x0100);        write_data(0x12b0);
    delayms(10);
    write_ctrl(0x0101);        write_data(0x0147);
    delayms(10);
    write_ctrl(0x0102);        write_data(0x01BE);
    delayms(10);
    write_ctrl(0x0103);        write_data(0x2400);
    delayms(10);
    write_ctrl(0x0107);        write_data(0x0000);
}
```



```
delaysms(10);
write_ctrl(0x0110);          write_data(0x0001);
delaysms(10);
//-----//
write_ctrl(0x0210);          write_data(0x0000);
write_ctrl(0x0211);          write_data(0x00ef);
write_ctrl(0x0212);          write_data(0x0000);
write_ctrl(0x0213);          write_data(0x018f);
write_ctrl(0x0200);          write_data(0x0000);
write_ctrl(0x0201);          write_data(0x0000);
write_ctrl(0x0280);          write_data(0x0000);
write_ctrl(0x0281);          write_data(0x0000);
write_ctrl(0x0282);          write_data(0x0000);
delaysms(10);
//-----Gamma setting-----//
write_ctrl(0x0300);          write_data(0x0100);
write_ctrl(0x0301);          write_data(0x072E);
write_ctrl(0x0302);          write_data(0x0131);
write_ctrl(0x0303);          write_data(0x3001);
write_ctrl(0x0304);          write_data(0x2E07);
write_ctrl(0x0305);          write_data(0x0001);
write_ctrl(0x0306);          write_data(0x1F01);
write_ctrl(0x0307);          write_data(0x011F);
write_ctrl(0x0308);          write_data(0x0406);
write_ctrl(0x0309);          write_data(0x0103);
write_ctrl(0x030A);          write_data(0x0b02);
write_ctrl(0x030B);          write_data(0x0B00);
write_ctrl(0x030C);          write_data(0x000B);
write_ctrl(0x030D);          write_data(0x020b);
write_ctrl(0x030E);          write_data(0x0301);
write_ctrl(0x030F);          write_data(0x0604);
//-----//
write_ctrl(0x0400);          write_data(0x3100);
write_ctrl(0x0401);          write_data(0x0001);
write_ctrl(0x0404);          write_data(0x0000);
write_ctrl(0x0500);          write_data(0x0000);
write_ctrl(0x0501);          write_data(0x0000);
write_ctrl(0x0502);          write_data(0x0000);
write_ctrl(0x0503);          write_data(0x0000);
write_ctrl(0x0504);          write_data(0x0000);
write_ctrl(0x0505);          write_data(0x0000);
write_ctrl(0x0600);          write_data(0x0000);
write_ctrl(0x0606);          write_data(0x0000);
write_ctrl(0x06F0);          write_data(0x0000);
//-----Orise mode-----//
write_ctrl(0x07F0);          write_data(0x5420);
write_ctrl(0x07F3);          write_data(0x280F);
write_ctrl(0x07F4);          write_data(0x0022);
write_ctrl(0x07F5);          write_data(0x0001);
```

```

        write_ctrl(0x07F0);        write_data(0x0000);
//-----//
        write_ctrl(0x0007);        write_data(0x0173);
        delayms(5);
        write_ctrl(0x0007);        write_data(0x0061);
        delayms(5);
        write_ctrl(0x0007);        write_data(0x0173);
        delayms(500);
    }

void Pattern_RED( )
{
    Signed int j;
    write_ctrl(0x0210);            write_data(0x0000); // xstart
    write_ctrl(0x0211);            write_data(0x00EF); // xend
    write_ctrl(0x0212);            write_data(0x0000); // ystart
    write_ctrl(0x0213);            write_data(0x018F); // yend
    write_ctrl(0x0200);            write_data(0x0000); // xstart
    write_ctrl(0x0201);            write_data(0x0000); // ystart
    write_ctrl(0x0202); // write GRAM
for( j =0; j<240*400; j++)
{
    Write_ Data (0xFC,0x00, 0x00); // red
}
}

```

CPU Interface enter sleep mode code

```

void Code( )
{
    write_ctrl(0x07F0);            write_data(0x5420);
    write_ctrl(0x07DE);            write_data(0x0C00);
    write_ctrl(0x07F0);            write_data(0x0000);
    write_ctrl(0x0007);            write_data(0x0000); //Display off
    delay_ms(50);
    write_ctrl(0x0100);            write_data(0x0002); //Set to sleep mode
    delayms(500);
}

```

CPU Interface exit sleep and deep stand by mode code

```

void code( )
{
    write_ctrl(0x07F0);            write_data(0x5420);
    write_ctrl(0x07DE);            write_data(0x0400);
    write_ctrl(0x07F0);            write_data(0x0000);
    write_ctrl(0x0100);            write_data(0x12B0); // Release state
    delay_ms(50);
    write_ctrl(0x0007);            write_data(0x0173); // Display on
    delayms(500);
}

```

7. HSD 3.0" APPLICATION NOTE

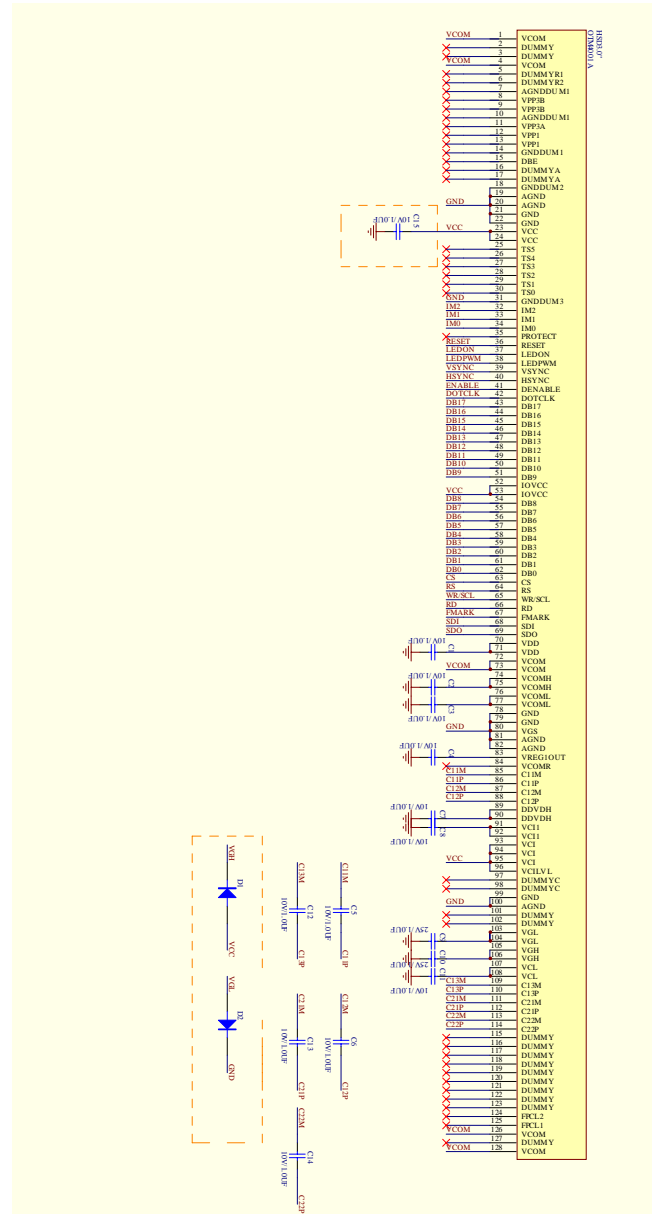
7.1. HSD 3.0" Application Circuit

IM2	IM1	IM0/ ID	Interface Mode	DB Pin	Colors
0	0	0	80-system 18-bit interface	DB17-0	282,144
0	0	1	80-system 9-bit interface	DB17-9	282,144
0	1	0	80-system 18-bit interface	DB17-10, DB8-1	262,144 see Note 1
0	1	1	80-system 8-bit interface	DB17-10	262,144 see Note 2
1	0	*(ID)	Clock synchronous serial interface	-	65,536
1	1	0	Setting disabled	-	-
1	1	1	Setting disabled	-	-

Notes: 1. 65,536 colors in one transfer mode
2. 65,536 colors in two transfers mode

Note(*)::C15,D1 ,D2

are not used, but Pls. Keep these pad on FPC.



Initial Code

7.1.1. HSD 3.0" initial code

```

void Code( )
{
//----- Power supply condition: VCC=IOVCC=VCI=2.8V -----//
//-----Power on sequence-----//
write_ctrl(0x0606);          write_data(0x0000);
delayms(100);
write_ctrl(0x0007);          write_data(0x0001);
delayms(5);
write_ctrl(0x0110);          write_data(0x0001);
delayms(5);
write_ctrl(0x0100);          write_data(0x17B0);
write_ctrl(0x0101);          write_data(0x0147);
write_ctrl(0x0102);          write_data(0x019D);
write_ctrl(0x0103);          write_data(0x3600);
write_ctrl(0x0281);          write_data(0x0010);
delayms(5);
write_ctrl(0x0102);          write_data(0x01BD);
delayms(5);
write_ctrl(0x0000);          write_data(0x0000);
write_ctrl(0x0001);          write_data(0x0100);
write_ctrl(0x0002);          write_data(0x0100);
write_ctrl(0x0003);          write_data(0x10A0);
write_ctrl(0x0008);          write_data(0x0503);
write_ctrl(0x0009);          write_data(0x0001);
write_ctrl(0x000B);          write_data(0x0010);
write_ctrl(0x000C);          write_data(0x0000);
write_ctrl(0x000F);          write_data(0x0000);
write_ctrl(0x0007);          write_data(0x0001);
//-----Panel interface control-----//
write_ctrl(0x0010);          write_data(0x0013);
write_ctrl(0x0011);          write_data(0x0202);
write_ctrl(0x0012);          write_data(0x0300);
write_ctrl(0x0020);          write_data(0x021e);
write_ctrl(0x0021);          write_data(0x0202);
write_ctrl(0x0090);          write_data(0x8000);
write_ctrl(0x0092);          write_data(0x0000);
//-----Voltage setting-----//
write_ctrl(0x0100);          write_data(0x11b0);
delayms(10);
write_ctrl(0x0101);          write_data(0x0147);
delayms(10);
write_ctrl(0x0102);          write_data(0x018D);
delayms(10);
write_ctrl(0x0103);          write_data(0x1600);
delayms(10);
write_ctrl(0x0107);          write_data(0x0000);
delayms(10);

```

```

write_ctrl(0x0110);          write_data(0x0001);
delayms(10);
//-----//
write_ctrl(0x0210);          write_data(0x0000);
write_ctrl(0x0211);          write_data(0x00ef);
write_ctrl(0x0212);          write_data(0x0000);
write_ctrl(0x0213);          write_data(0x018f);
write_ctrl(0x0200);          write_data(0x0000);
write_ctrl(0x0201);          write_data(0x0000);
write_ctrl(0x0280);          write_data(0x0000);
write_ctrl(0x0281);          write_data(0x0000);
write_ctrl(0x0282);          write_data(0x0000);
delayms(10);
//-----Gamma setting-----//
write_ctrl(0x0300);          write_data(0x1004);
write_ctrl(0x0301);          write_data(0x231e);
write_ctrl(0x0302);          write_data(0x2521);
write_ctrl(0x0303);          write_data(0x2125);
write_ctrl(0x0304);          write_data(0x1e23);
write_ctrl(0x0305);          write_data(0x0410);
write_ctrl(0x0306);          write_data(0x1006);
write_ctrl(0x0307);          write_data(0x0610);
write_ctrl(0x0308);          write_data(0x0205);
write_ctrl(0x0309);          write_data(0x0104);
write_ctrl(0x030A);          write_data(0x0d05);
write_ctrl(0x030B);          write_data(0x0d01);
write_ctrl(0x030C);          write_data(0x010d);
write_ctrl(0x030D);          write_data(0x050d);
write_ctrl(0x030E);          write_data(0x0401);
write_ctrl(0x030F);          write_data(0x0502);
//-----//

write_ctrl(0x0400);          write_data(0x3500);
write_ctrl(0x0401);          write_data(0x0001);
write_ctrl(0x0404);          write_data(0x0000);
write_ctrl(0x0500);          write_data(0x0000);
write_ctrl(0x0501);          write_data(0x0000);
write_ctrl(0x0502);          write_data(0x0000);
write_ctrl(0x0503);          write_data(0x0000);
write_ctrl(0x0504);          write_data(0x0000);
write_ctrl(0x0505);          write_data(0x0000);
write_ctrl(0x0600);          write_data(0x0000);
write_ctrl(0x0606);          write_data(0x0000);
write_ctrl(0x06F0);          write_data(0x0000);
//-----Orise mode-----//
write_ctrl(0x07F0);          write_data(0x5420);
write_ctrl(0x07F3);          write_data(0x208d);
write_ctrl(0x07F4);          write_data(0x0022);
write_ctrl(0x07F5);          write_data(0x0061);

```

```

write_ctrl(0x07F0);          write_data(0x0000);

write_ctrl(0x0007);          write_data(0x0173);
delayms(5);
write_ctrl(0x0007);          write_data(0x0061);
delayms(5);
write_ctrl(0x0007);          write_data(0x0173);
delayms(500);
void Pattern_RED( )
{
    Signed int j;
write_ctrl(0x0210);          write_data(0x0000); // xstart
write_ctrl(0x0211);          write_data(0x00EF); // xend
write_ctrl(0x0212);          write_data(0x0000); // ystart
write_ctrl(0x0213);          write_data(0x018F); // yend
write_ctrl(0x0200);          write_data(0x0000); // xstart
write_ctrl(0x0201);          write_data(0x0000); // ystart
write_ctrl(0x0202); // write GRAM
for( j =0; j<240*400; j++ )
{
Write_ Data (0xFC,0x00, 0x00); // red
}
}

```

CPU Interface enter sleep mode code

```

void Code( )
{
write_ctrl(0x07F0);          write_data(0x5420);
write_ctrl(0x07DE);          write_data(0x0C00);
write_ctrl(0x07F0);          write_data(0x0000);
write_ctrl(0x0007);          write_data(0x0000); //Display off
delay_ms(50);
write_ctrl(0x0100);          write_data(0x0002); //Set to sleep mode
delayms(500);
}

```

CPU Interface exit sleep and deep stand by mode code

```

void code( )
{
write_ctrl(0x07F0);          write_data(0x5420);
write_ctrl(0x07DE);          write_data(0x0400);
write_ctrl(0x07F0);          write_data(0x0000);
write_ctrl(0x0100);          write_data(0x11B0); // Release state
delay_ms(50);
write_ctrl(0x0007);          write_data(0x0173); // Display on
delayms(500);
}

```

DISCLAIMER :

The information appearing in this publication is believed to be accurate.

Integrated circuits sold by ORISE Technology are covered by the warranty and patent indemnification provisions stipulated in the terms of sale only. ORISE Technology makes no warranty, express, statutory implied or by description regarding the information in this publication or regarding the freedom of the described chip(s) from patent infringement. FURTHERMORE, ORISE Technology MAKES NO WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PURPOSE. ORISE Technology reserves the right to halt production or alter the specifications and prices at any time without notice. Accordingly, the reader is cautioned to verify that the Application Note and other information in this publication are current before placing orders. Products described herein are intended for use in normal commercial applications. Applications involving unusual environmental or reliability requirements, e.g. military equipment or medical life support equipment, are specifically not recommended without additional processing by ORISE Technology for such applications. Please note that application circuits illustrated in this document are for reference purposes only.

8. REVISION HISTORY

Date	Revision #	Description	Page
Apr. 07,2010	0.4	Add GPT2.5" AN Add BOE2.8" AN	48
Mar.08.2010	0.3	Add CPT2.8" AN Add HSD3.0" AN	40
Jan.11.2010	0.2	Add CMO3.0" initial Code Add HTT3.0" initial Code Add AUO3.2" initial Code	28
OCT. 29, 2009	0.1	Original	22