P47101

32x3x64 Full Color Application Notes (for SPI Interface)

Revision History

Version	REVISION DESCRIPTION
X01	First release

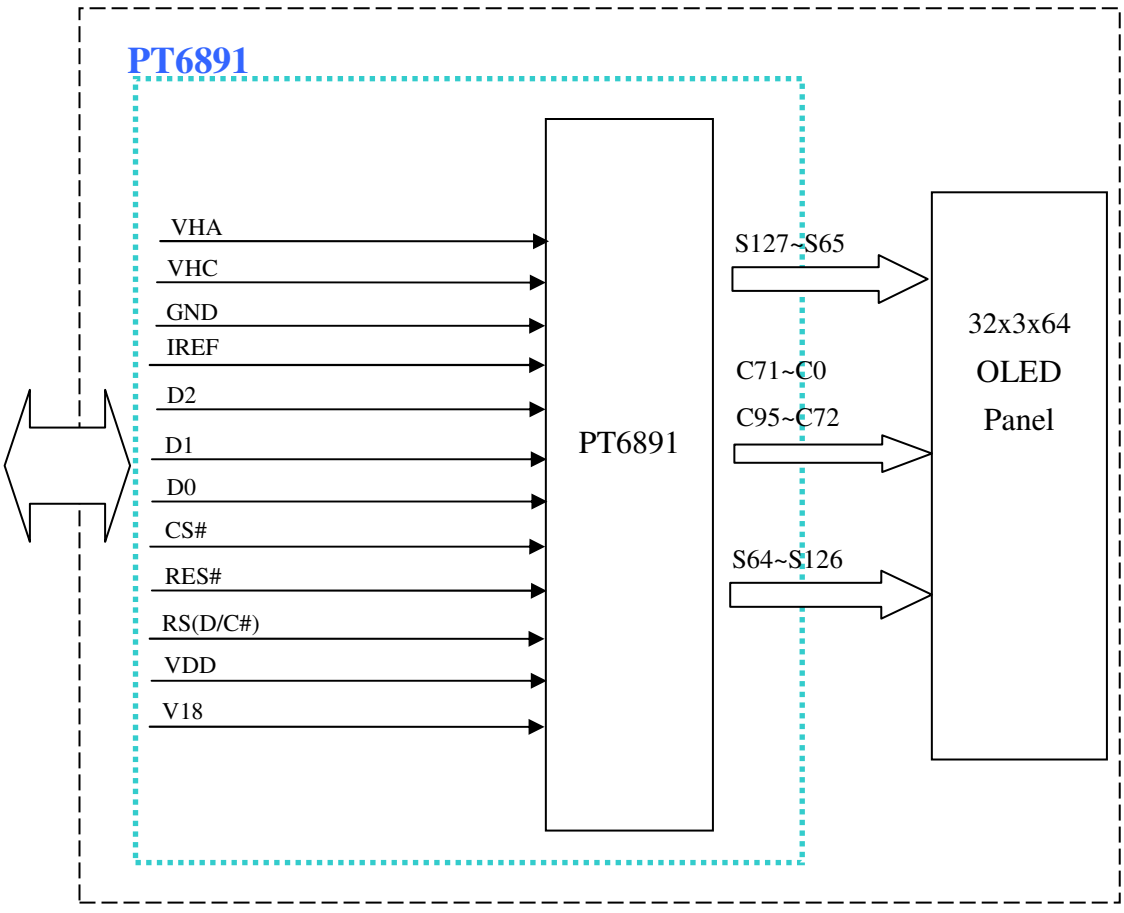
DESCRIPTION

P47101 is a 32x3x64 dot matrix full color passive OLED module with controller for many compact portable applications.

FEATURE

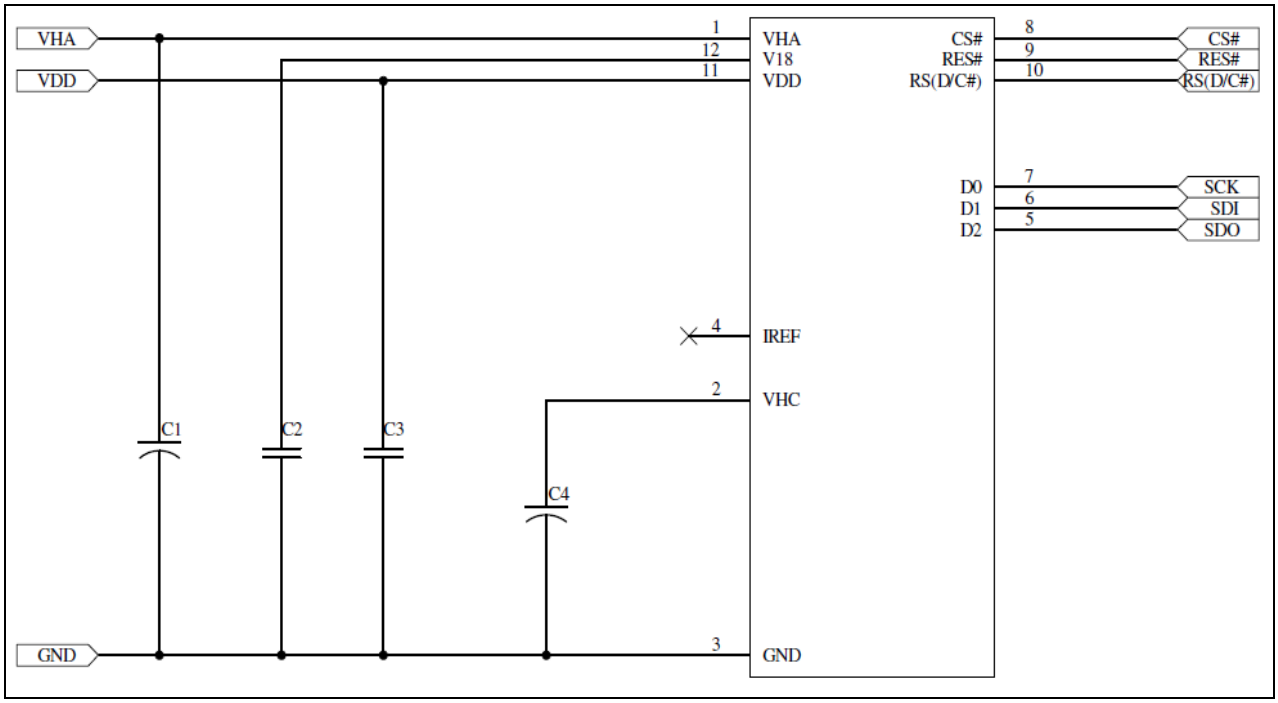
- 32x3x64 dot matrix full color OLED panel.
- Driver IC: PT6891.
- VCC=15V.
- VDD = 2.4V ~ 3.6V.
- 4-wire serial peripheral interface.

FUNCTION BLOCK DIAGRAM



RiTdisplay 32X3X64 OLED Module

APPLICATION CIRCUIT



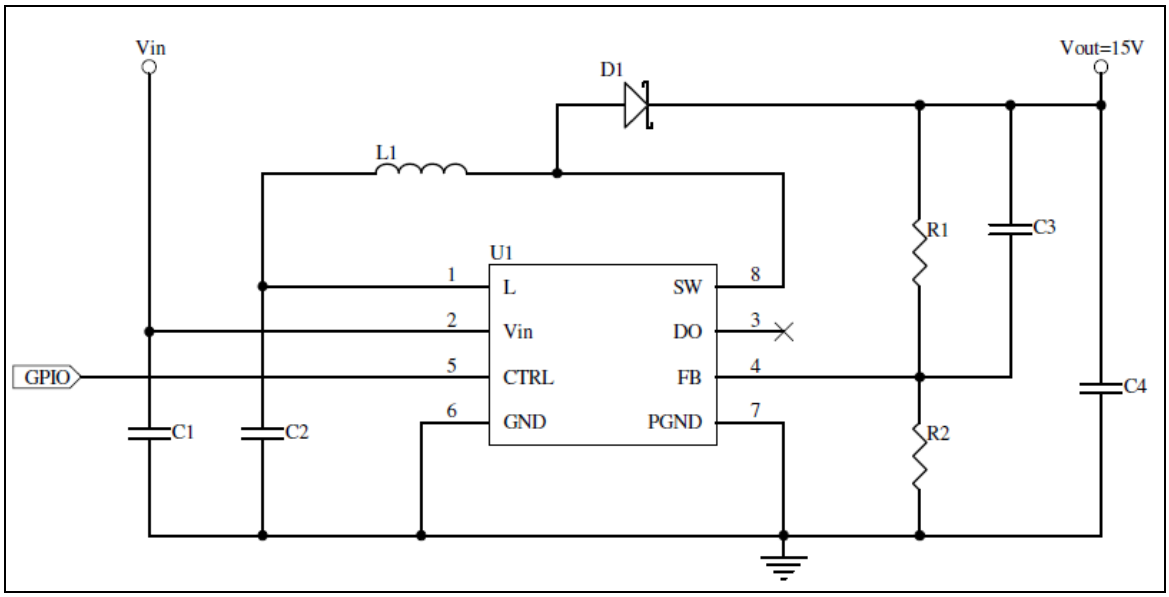
Recommend components:

C1, C4: 4.7uF/25V(0805)

C2, C3: 1uF/6.3V(0603)

This circuit is for 4 wire SPI interface.

DC-DC application circuit for OLED module(For External DC/DC)



Recommend components:

The C1: 0.1uF/25V.

The C2: 4.7uF/25V.

The C3: 22pF/16V.

The C4: 4.7uF/25V Tantalum type capacitor.

The R1: 1.2M ohm1%.

The R2: 113K ohm1%.

The D1: SCHOTTY DIODE.

The L1: 10uH.

The U1: TPS61045.

The R1, R2 and C3 value should be fine tune by customer.

PIN ASSIGNMENTS

PIN NAME	PIN NO.	DESCRIPTION	Setting at each interface		
			8080 parallel	4-SPI	IIC
1	VHA	Power supply for panel driving voltage.			
2	VHC	This is the most positive voltage supply pin of the chip to drive cathode.			
3	GND	Ground pin.			
4	IREF	This is reference current pin. A resistor (typical 100KΩ) connected between this pin and ground determines the reference current for anode driver. When register "IRE"=1, IREF pin will be used.			
5	D2	When SPI mode is selected, D[2] will be the serial data output (SDO), D[1] will be the serial data input (SDI) and D[0] will be the serial clock input (SCK).	NA	SD0(OUT)	NA
6	D1	When SPI mode is selected, D[2] will be the serial data output (SDO), D[1] will be the serial data input (SDI) and D[0] will be the serial clock input (SCK).	NA	SDI(IN)	NA
7	D0	When SPI mode is selected, D[2] will be the serial data output (SDO), D[1] will be the serial data input (SDI) and D[0] will be the serial clock input (SCK).	NA	SCK	NA
8	CS#	This pin is the chip select input. The chip is enabled for MCU communication only when CS# is pulled LOW.	NA	CS#	NA
9	RES#	This pin is low-active reset input. When the pin is LOW, the chip is reset.	NA	RES#	NA
10	RS(D/C#)	This pin is Data/Command control pin.	NA	RS	NA
11	VDD	This is Logic power input.			
12	V18	This is 1.8V power input pin for core logic circuit. If regulator is enabled (GND in LDO_DIS pin), this pin connected to VRO pin. If regulator is disabled (VDD in LDO_DIS pin), this pin is connected to external 1.8V power.			

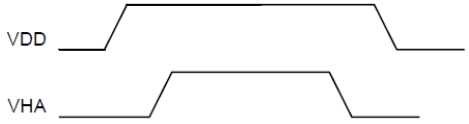
Note

(1) Low is connected to GND

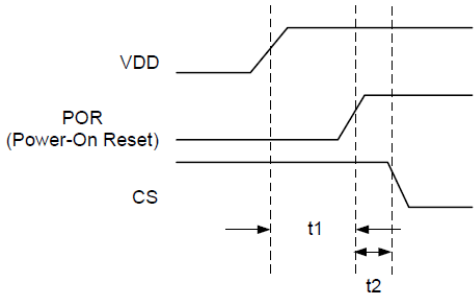
(2) High is connected to VDD

Power ON / OFF Sequence

VDD AND VHA POWER SEQUENCE TIMING



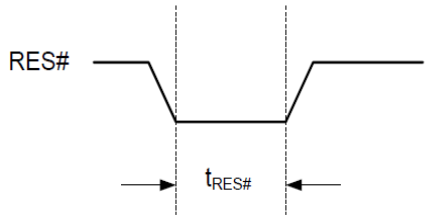
POWER-ON AND RESET TIMING



Item	Unit (µs)
t1	600
t2	100

MINIMUM LOW-ACTIVE PULSE WIDTH REQUIRED FOR RES# PIN

Minimum Pulse Width for RES# Pin	Min.	Description
$t_{RES\#}$	T_{OSC} * 40	$t_{RES\#}$ ➔ Valid RES# signal



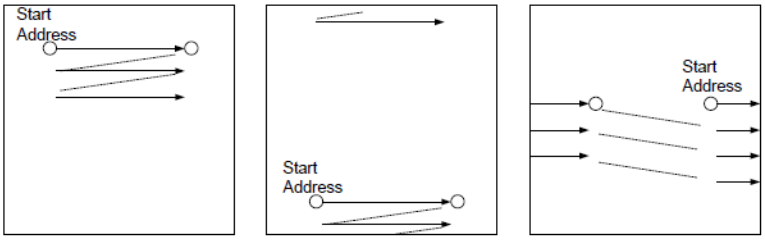
Note: In standby mode, low of RES# cannot reset the chip.

Graphic Display Data RAM Address Map

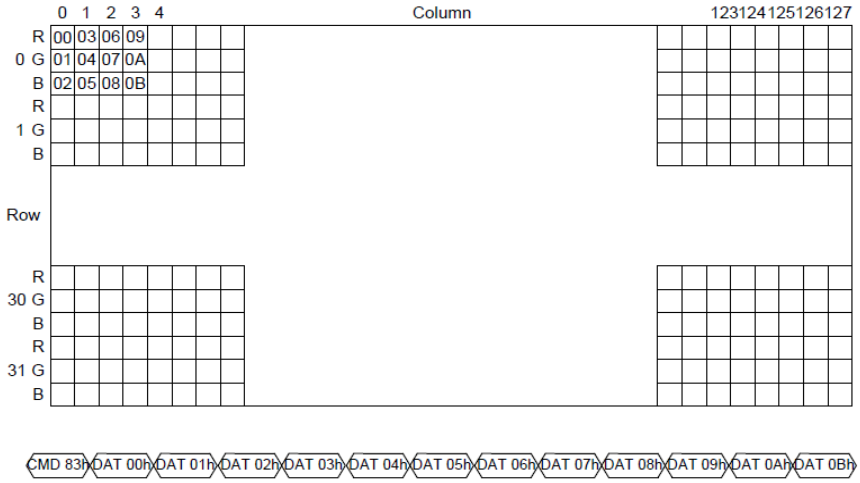
This RAM data Read/Write in RGB / Mono mode.

	D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]/SDA	D[0]/SCK	Function
1st	*	*	R[5:0]						8 bits bus/color mode(RGB666) Serial interface(D7~D0)/color mode(262k colors)
2nd	*	*	G[5:0]						
3rd	*	*	B[5:0]						
1st	R[4:0]					G[5:3]			8 bits bus/color mode(RGB565) Serial interface(D7~D0)/color mode (262k colors)
2nd	G[2:0]		B[4:0]						
1st	*	*	DATA[5:0]						8 bits bus/mono mode(64 Gray)
1st	P7	P6	P5	P4	P3	P2	P1	P0	8 bits bus/mono mode(2 Gray)

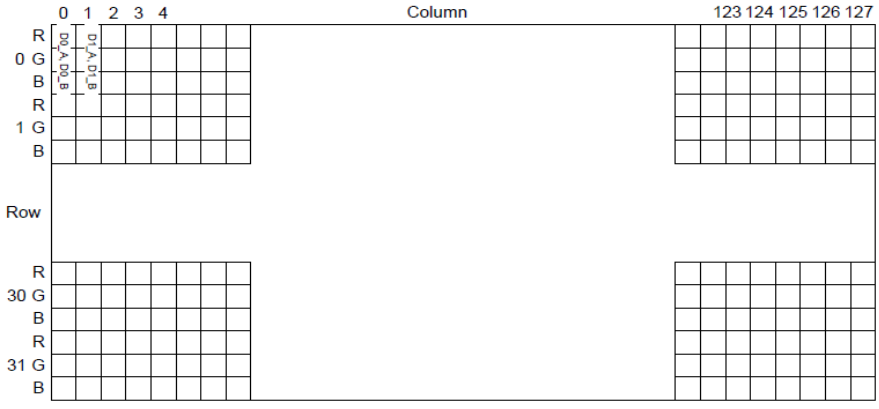
If the pixels of a line travel across the line boundary, the pixel position return to the beginning of the line and continue to traverse the line till the number of pixels satisfy. Then, the pixel position goes to the “column start” of the next line. The following 3 figures illustrate the behavior.

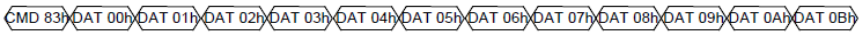


Example: for color mode(RGB666), write data: 00h, 01h, …, 0Bh

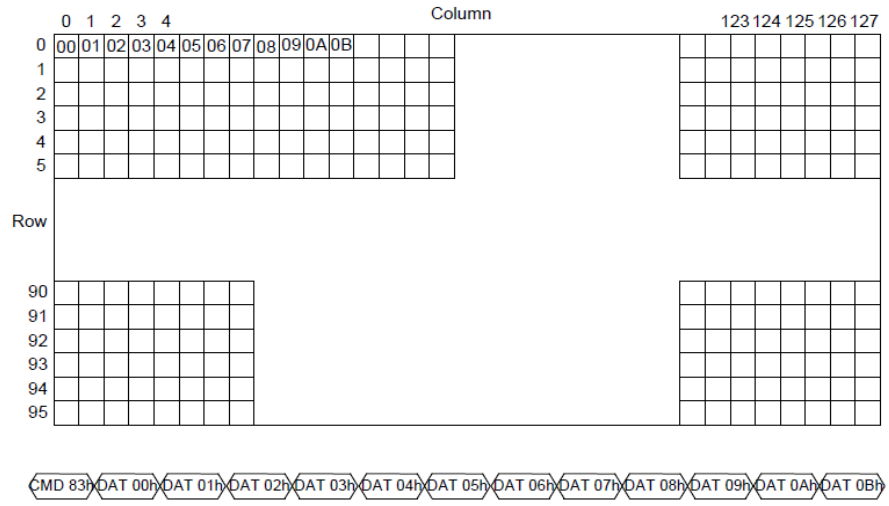


Example: for color mode(RGB565), write data: D0_A, D0_B, …, D4_A, D4_Bh
RGB565(R[4:0], G[5:0], B[4:0]) be transferred to RGB666(R[4:0],1’b0, G[5:0], B[4:0], 1’b0).

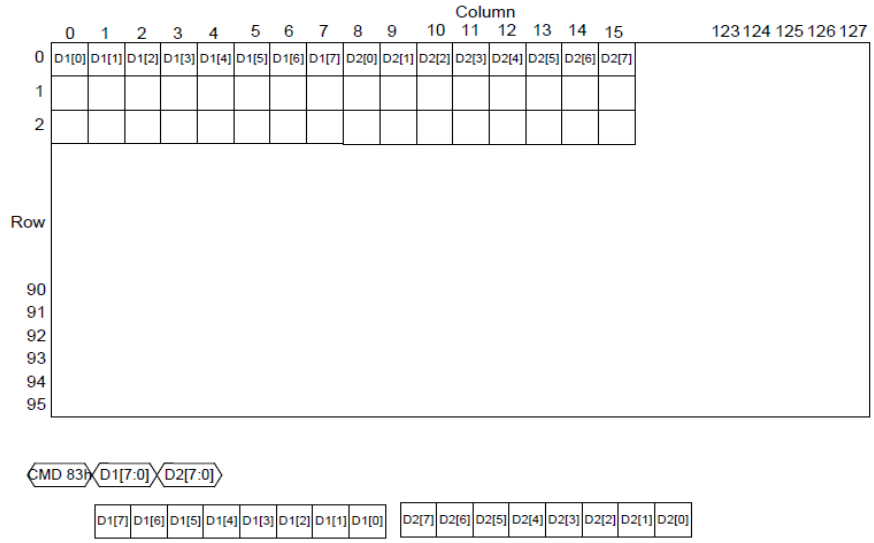




Example: mono mode(64 gray), write data: 00h, 01h, …, 0Bh



Example: mono mode(2 gray), write data: D1, D2



Application Initial Setting

```
/*32x3x64 OLED driver program */
/* The more detail of SPI sequence please refer the PT6891 datasheet */
void initial(void)
{
comm_out(0xa5);//Software Reset

comm_out(0x58);//Display ON/OFF(Type_1 OFF)

comm_out(0xb5);//Voltage/Current Control
data_out(0x87);// Enable internal IREF

comm_out(0x30);//Color Mode(CM=00(RGB: 666; 262K colors))

comm_out(0x28);//G Gamma table setting
data_out(0x00);//0
data_out(0x04);//1
data_out(0x08);//2
data_out(0x0c);//3
data_out(0x10);//4
data_out(0x14);//5
data_out(0x18);//6
data_out(0x1c);//7
data_out(0x20);//8
data_out(0x24);//9
//====================
data_out(0x28);//10
data_out(0x2c);//11
data_out(0x30);//12
data_out(0x34);//13
data_out(0x38);//14
data_out(0x3c);//15
data_out(0x40);//16
data_out(0x44);//17
data_out(0x48);//18
data_out(0x4c);//19
//====================
```

```
data_out(0x50);//20
data_out(0x54);//21
data_out(0x58);//22
data_out(0x5c);//23
data_out(0x60);//24
data_out(0x64);//25
data_out(0x68);//26
data_out(0x6c);//27
data_out(0x70);//28
data_out(0x74);//29
//====================
data_out(0x78);//30
data_out(0x7c);//31
data_out(0x80);//32
data_out(0x84);//33
data_out(0x88);//34
data_out(0x8c);//35
data_out(0x90);//36
data_out(0x94);//37
data_out(0x98);//38
data_out(0x9c);//39
//====================
data_out(0xa0);//40
data_out(0xa4);//41
data_out(0xa8);//42
data_out(0xac);//43
data_out(0xb0);//44
data_out(0xb4);//45
data_out(0xb8);//46
data_out(0xbc);//47
data_out(0xc0);//48
data_out(0xc4);//49
//====================
data_out(0xc8);//50
data_out(0xcc);//51
data_out(0xd0);//52
data_out(0xd4);//53
```

```
data_out(0xd8);//54
data_out(0xdc);//55
data_out(0xe0);//56
data_out(0xe4);//57
data_out(0xe8);//58
data_out(0xec);//59
//====================
data_out(0xf0);//60
data_out(0xf4);//61
data_out(0xf8);//62
data_out(0xfc);//63

comm_out(0x2b);//Gamma Table Update

//----------------------------------
comm_out(0x29);//B Gamma table setting
data_out(0x00);//0
data_out(0x04);//1
data_out(0x08);//2
data_out(0x0c);//3
data_out(0x10);//4
data_out(0x14);//5
data_out(0x18);//6
data_out(0x1c);//7
data_out(0x20);//8
data_out(0x24);//9
//====================
data_out(0x28);//10
data_out(0x2c);//11
data_out(0x30);//12
data_out(0x34);//13
data_out(0x38);//14
data_out(0x3c);//15
data_out(0x40);//16
data_out(0x44);//17
data_out(0x48);//18
data_out(0x4c);//19
```

```
//====================
data_out(0x50);//20
data_out(0x54);//21
data_out(0x58);//22
data_out(0x5c);//23
data_out(0x60);//24
data_out(0x64);//25
data_out(0x68);//26
data_out(0x6c);//27
data_out(0x70);//28
data_out(0x74);//29
//====================
data_out(0x78);//30
data_out(0x7c);//31
data_out(0x80);//32
data_out(0x84);//33
data_out(0x88);//34
data_out(0x8c);//35
data_out(0x90);//36
data_out(0x94);//37
data_out(0x98);//38
data_out(0x9c);//39
//====================
data_out(0xa0);//40
data_out(0xa4);//41
data_out(0xa8);//42
data_out(0xac);//43
data_out(0xb0);//44
data_out(0xb4);//45
data_out(0xb8);//46
data_out(0xbc);//47
data_out(0xc0);//48
data_out(0xc4);//49
//====================
data_out(0xc8);//50
data_out(0xcc);//51
data_out(0xd0);//52
```

```
data_out(0xd4);//53
data_out(0xd8);//54
data_out(0xdc);//55
data_out(0xe0);//56
data_out(0xe4);//57
data_out(0xe8);//58
data_out(0xec);//59
//====================
data_out(0xf0);//60
data_out(0xf4);//61
data_out(0xf8);//62
data_out(0xfc);//63

comm_out(0x2b);//Gamma Table Update

//----------------------------------
comm_out(0x2a);//R Gamma table setting
data_out(0x00);//0
data_out(0x04);//1
data_out(0x08);//2
data_out(0x0c);//3
data_out(0x10);//4
data_out(0x14);//5
data_out(0x18);//6
data_out(0x1c);//7
data_out(0x20);//8
data_out(0x24);//9
//====================
data_out(0x28);//10
data_out(0x2c);//11
data_out(0x30);//12
data_out(0x34);//13
data_out(0x38);//14
data_out(0x3c);//15
data_out(0x40);//16
data_out(0x44);//17
data_out(0x48);//18
```

```
data_out(0x4c);//19
//====================
data_out(0x50);//20
data_out(0x54);//21
data_out(0x58);//22
data_out(0x5c);//23
data_out(0x60);//24
data_out(0x64);//25
data_out(0x68);//26
data_out(0x6c);//27
data_out(0x70);//28
data_out(0x74);//29
//====================
data_out(0x78);//30
data_out(0x7c);//31
data_out(0x80);//32
data_out(0x84);//33
data_out(0x88);//34
data_out(0x8c);//35
data_out(0x90);//36
data_out(0x94);//37
data_out(0x98);//38
data_out(0x9c);//39
//====================
data_out(0xa0);//40
data_out(0xa4);//41
data_out(0xa8);//42
data_out(0xac);//43
data_out(0xb0);//44
data_out(0xb4);//45
data_out(0xb8);//46
data_out(0xbc);//47
data_out(0xc0);//48
data_out(0xc4);//49
//====================
data_out(0xc8);//50
data_out(0xcc);//51
```

```
data_out(0xd0);//52
data_out(0xd4);//53
data_out(0xd8);//54
data_out(0xdc);//55
data_out(0xe0);//56
data_out(0xe4);//57
data_out(0xe8);//58
data_out(0xec);//59
//====================
data_out(0xf0);//60
data_out(0xf4);//61
data_out(0xf8);//62
data_out(0xfc);//63

comm_out(0x2b);//Gamma Table Update
//----------------------------------
comm_out(0xe0);//COM Number
data_out(0x1f);

comm_out(0xe1);//Display Row Setting
data_out(0x00);

comm_out(0xe2);//Display Column Setting
data_out(0x00);

comm_out(0xe6);//Dummy Scan

comm_out(0x44);//Clock Divider

comm_out(0x4c);//OSC Trimming
data_out(0x08);

comm_out(0xd8);//COM Pulse Width
data_out(0x25);
data_out(0x01);
```

```
comm_out(0xe5);//Blank Period
data_out(0x0a);

comm_out(0xb4);//SEG EVEN/ODD Swap
data_out(0x00);

comm_out(0xb3);//SEG Output Type
omm_out(0x68);//Vertical and Horizontal Mirror

comm_out(0xbc);//Cathode Scan Direction

comm_out(0xbe);//Anode Trimming
data_out(0x08);

comm_out(0xb8);//Brightness(VHA=15V)
data_out(0x21);//R
data_out(0x14);//G
data_out(0x1c);//B

comm_out(0xc8);//PRE-CHARGE PERIOD
data_out(0x0a);

comm_out(0xd0);//PRE-CHARGE CURRENT
data_out(0x00);

cleanDDR();//Clear the whole DDRAM

comm_out(0x43);//INT Setting

comm_out(0x5b);//Display ON/OFF(Normally display)

}
```

```
void cleanDDR(void)
{
int i,j;

comm_out(0x80);//Row Address Setting
data_out(0x00);

comm_out(0x81);//Column Address Setting
data_out(0x00);

comm_out(0x82);//Return Length Setting
data_out(0x7f);

comm_out(0x83);//Display Data Write

for(i=0;i<32;i++)
 {
  for(j=0;j<128;j++)
  {
  data_out(0x00);
  data_out(0x00);
  data_out(0x00);
  }
 }
}
write_red_data(void)
{
int i,j;

comm_out(0x80);//Row Address Setting
data_out(0x00);

comm_out(0x81);//Column Address Setting
data_out(0x40);

comm_out(0x82);//Return Length Setting
data_out(0x3f);
```

```
comm_out(0x83);//Display Data Write

for(i=0;i<32;i++)
 {
  for(j=0;j<64;j++)
  {
  data_out(0x3f);
  data_out(0x00);
  data_out(0x00);
  }
 }
}

write_green_data(void)
{
int i,j;

comm_out(0x80);//Row Address Setting
data_out(0x00);

comm_out(0x81);//Column Address Setting
data_out(0x40);

comm_out(0x82);//Return Length Setting
data_out(0x3f);

comm_out(0x83);//Display Data Write

for(i=0;i<32;i++)
 {
  for(j=0;j<64;j++)
  {
  data_out(0x00);
  data_out(0x3f);
  data_out(0x00);
  }
```

```
 }
}

write_blue_data(void)
{
int i,j;

comm_out(0x80);//Row Address Setting
data_out(0x00);

comm_out(0x81);//Column Address Setting
data_out(0x40);

comm_out(0x82);//Return Length Setting
data_out(0x3f);

comm_out(0x83);//Display Data Write

for(i=0;i<32;i++)
 {
  for(j=0;j<64;j++)
  {
  data_out(0x00);
  data_out(0x00);
  data_out(0x3f);
  }
 }
}


write_white_data(void)
{
int i,j;

comm_out(0x80);//Row Address Setting
data_out(0x00);
```

```
comm_out(0x81);//Column Address Setting
data_out(0x40);

comm_out(0x82);//Return Length Setting
data_out(0x3f);

comm_out(0x83);//Display Data Write

for(i=0;i<32;i++)
  {
    for(j=0;j<64;j++)
    {
    data_out(0x3f);
    data_out(0x3f);
    data_out(0x3f);
    }
  }
}
```

For 120 cd/m^2 setting, user could follow the below setting.

```
Brightness_mode1 (void);
{
comm_out(0xb8);//Brightness
data_out(0x2c);//R
data_out(0x18);//G
data_out(0x27);//B
}
```

For 100 cd/m^2 setting, user could follow the below setting.

```
Brightness_mode2 (void);
{
comm_out(0xb8);//Brightness
data_out(0x21);//R
data_out(0x14);//G
data_out(0x1c);//B
}
```

For 80 cd/m^2 setting, user could follow the below setting.

```
Brightness_mode3 (void);
{
comm_out(0xb8);//Brightness
data_out(0x0d);//R
data_out(0x0d);//G
data_out(0x12);//B
}
```

For 70 cd/m^2 setting, user could follow the below setting.

```
Brightness_mode4 (void);
{
comm_out(0xb8);//Brightness
data_out(0x0b);//R
data_out(0x09);//G
data_out(0x0e);//B
}
```

Thank You