P43202

28x3x95 Full Color
Application Notes
(For 4 wire SPI Interface)

Revision History

Version	Content
X01	First release(For 4 wire SPI Interface)

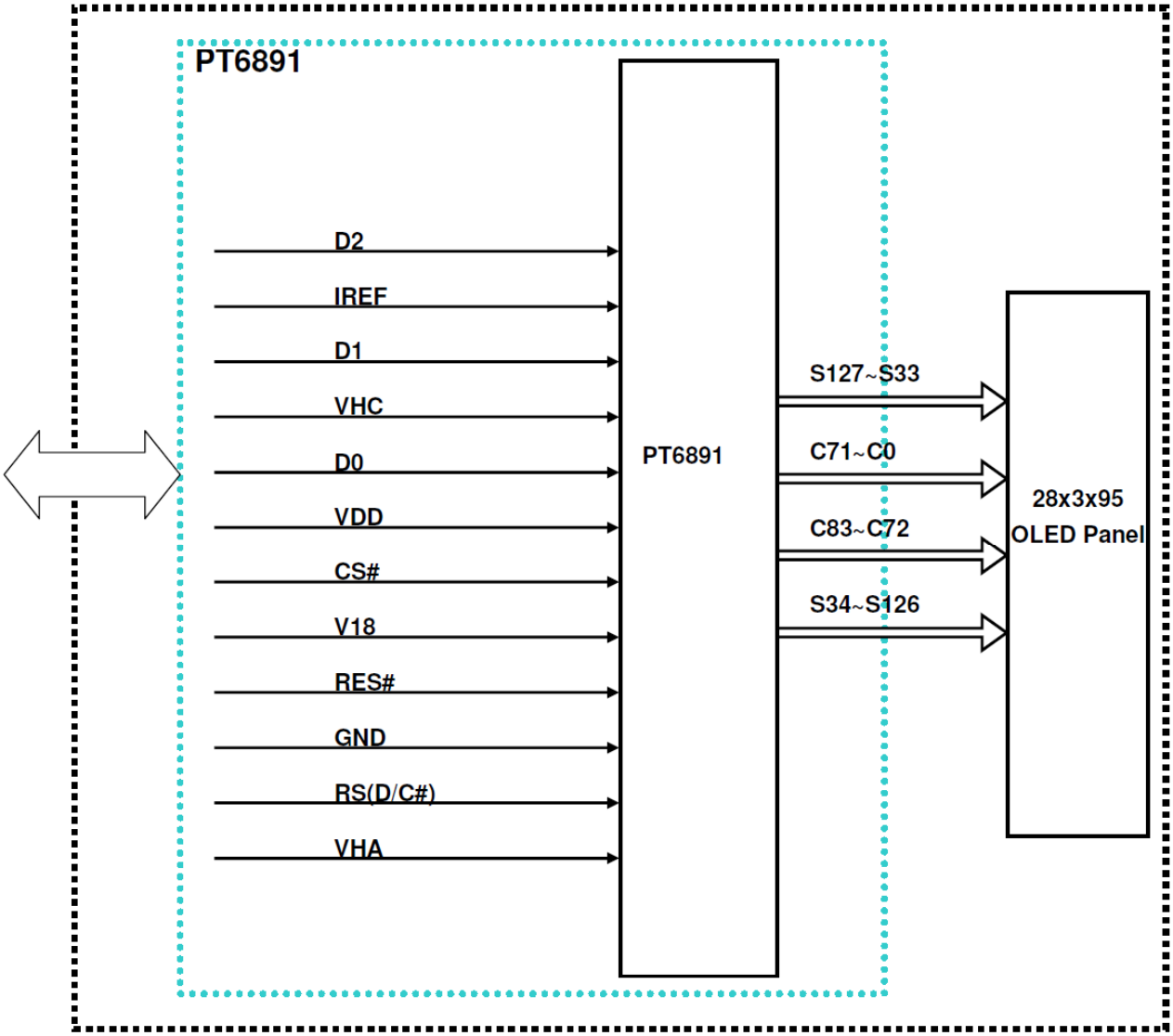
DESCRIPTION

P43202 is a 28x3x95 full color passive OLED module with controller for many compact portable applications.

FEATURE

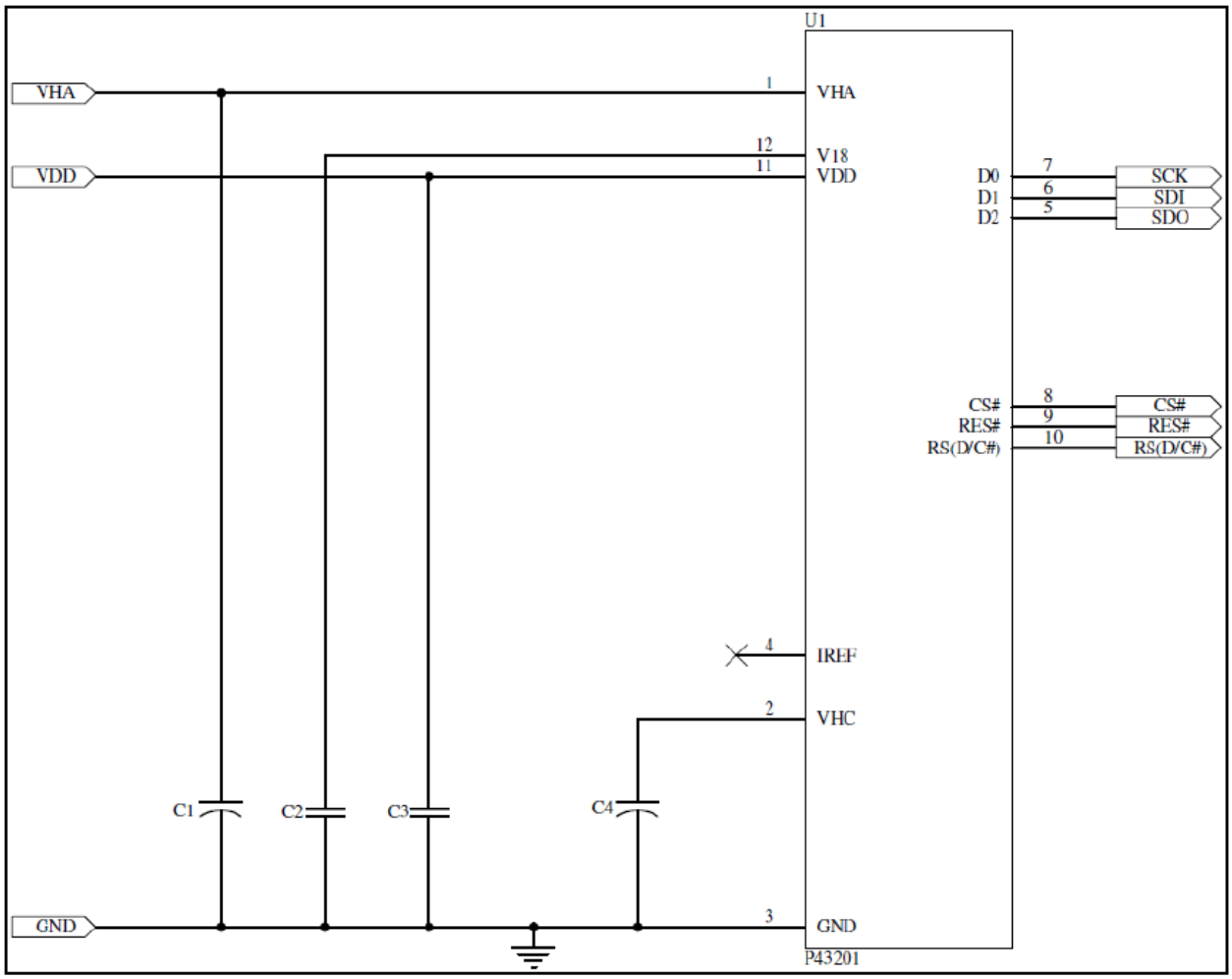
- 28x3x95 dot matrix full color OLED panel.
- Driver IC PT6891.
- VHA = 15V
- VDD = 2.4V~3.6V
- 4 wire serial peripheral interface.

FUNCTION BLOCK DIAGRAM



RiTdisplay 28X3x95 OLED Module

APPLICATION CIRCUIT



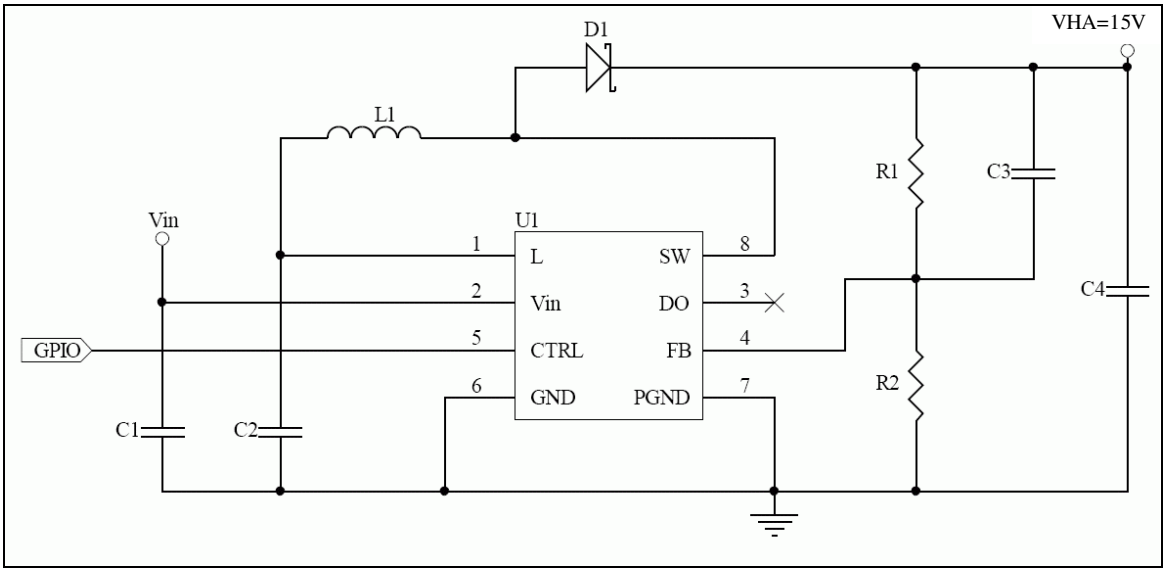
Recommend components:

C1, C4: 4.7uF/25V

C2, C3: 1uF/6.3V

This circuit is for 4 wire SPI interface.

DC-DC application circuit for OLED module



Recommend components:

The C1: 0.1uF/6.3V.

The C2: 4.7 uF/6.3V.

The C3: 22pF/16V.

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

The R1: 1.27M 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 No.	Pin Name	Description	Setting at each interface		
			8080 parallel	4 wire 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.			
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	SDO(OUT)	NA
6	D1		NA	SDI(IN)	NA
7	D0		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#	When the pin is LOW, the chip is reset.			
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.			

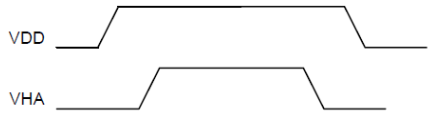
Note

(1) Low is connected to GND

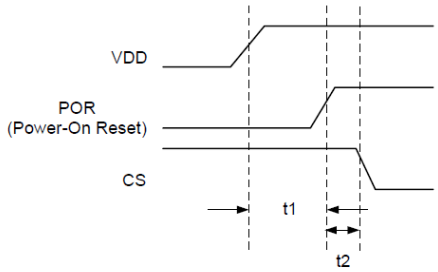
(2) High is connected to VDD

Power ON / OFF Sequence

VDD AND VHA POWER SEQUENCE TIMING



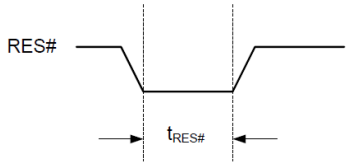
POWER-ON AND RESET TIMING



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



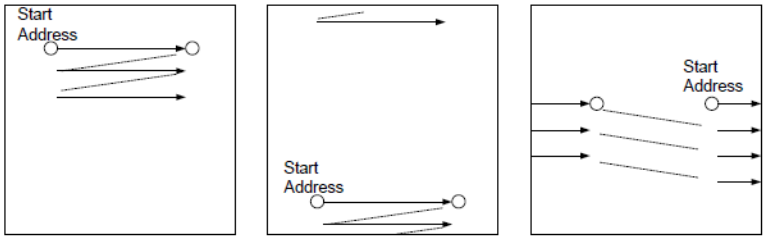
Note: In standby mode, low of RES# cannot reset the chip.

Graphic Display Data RAM Address Map

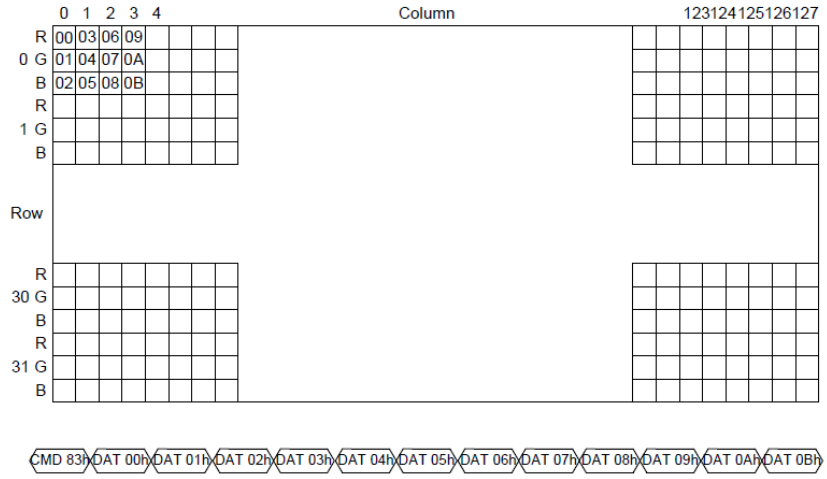
This RAM data Read/Write in RGB / Mono mode.

	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)

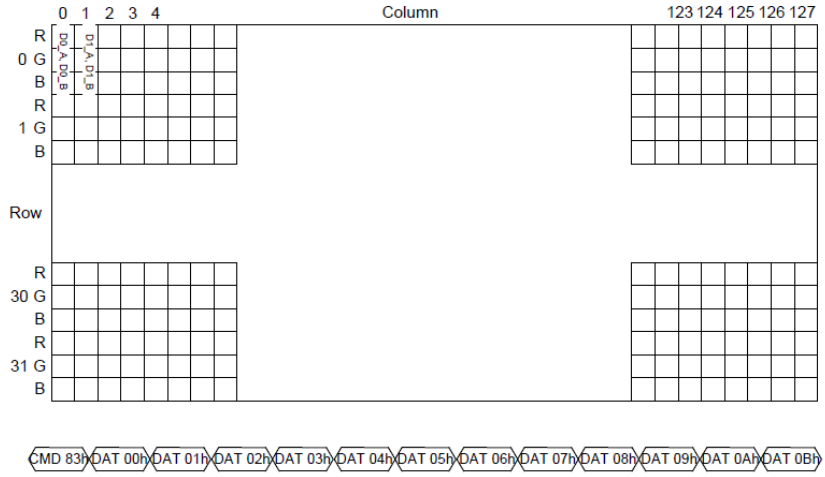
If the pixels of a line travel across the line boundary, the pixel position return to the beginning of the line and continue to traverse the line till the number of pixels satisfy. Then, the pixel position goes to the “column start” of the next line. The following 3 figures illustrate the behavior.



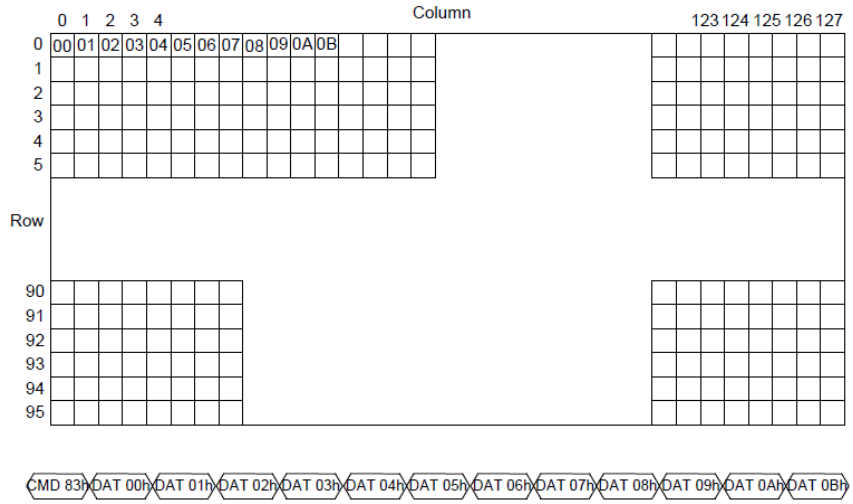
Example: for color mode(RGB666), write data: 00h, 01h, …, 0Bh



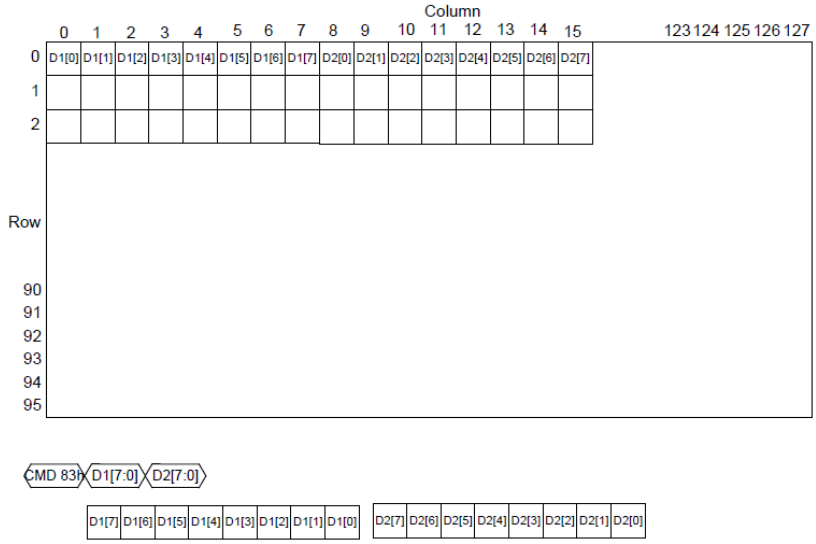
Example: for color mode(RGB565), write data: D0_A, D0_B, …, D4_A, D4_Bh
RGB565(R[4:0], G[5:0], B[4:0]) be transferred to RGB666(R[4:0],1’b0, G[5:0], B[4:0], 1’b0).



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



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



Application Initial Setting

```
//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(0x01);//1
data_out(0x02);//2
data_out(0x03);//3
data_out(0x04);//4
data_out(0x05);//5
data_out(0x06);//6
data_out(0x07);//7
data_out(0x08);//8
data_out(0x09);//9
//====================
data_out(0x0a);//10
data_out(0x0b);//11
data_out(0x0d);//12
data_out(0x0f);//13
data_out(0x11);//14
data_out(0x13);//15
data_out(0x16);//16
data_out(0x18);//17
data_out(0x1b);//18
data_out(0x1d);//19
//====================
data_out(0x20);//20
```

```
data_out(0x23);//21
data_out(0x26);//22
data_out(0x29);//23
data_out(0x2c);//24
data_out(0x30);//25
data_out(0x33);//26
data_out(0x37);//27
data_out(0x3a);//28
data_out(0x3e);//29
//====================
data_out(0x42);//30
data_out(0x46);//31
data_out(0x4a);//32
data_out(0x4e);//33
data_out(0x53);//34
data_out(0x57);//35
data_out(0x5c);//36
data_out(0x60);//37
data_out(0x65);//38
data_out(0x6a);//39
//====================
data_out(0x6f);//40
data_out(0x74);//41
data_out(0x79);//42
data_out(0x7e);//43
data_out(0x83);//44
data_out(0x89);//45
data_out(0x8e);//46
data_out(0x94);//47
data_out(0x9a);//48
data_out(0x9f);//49
//====================
data_out(0xa5);//50
data_out(0xab);//51
data_out(0xb1);//52
data_out(0xb7);//53
data_out(0xbe);//54
data_out(0xc4);//55
```

```
data_out(0xca);//56
data_out(0xd1);//57
data_out(0xd8);//58
data_out(0xde);//59
//====================
data_out(0xe5);//60
data_out(0xec);//61
data_out(0xf3);//62
data_out(0xfa);//63

comm_out(0x2b);//Gamma Table Update

//----------------------------------
comm_out(0x29);//B Gamma table setting
data_out(0x00);//0
data_out(0x01);//1
data_out(0x02);//2
data_out(0x03);//3
data_out(0x04);//4
data_out(0x05);//5
data_out(0x06);//6
data_out(0x07);//7
data_out(0x08);//8
data_out(0x09);//9
//====================
data_out(0x0a);//10
data_out(0x0b);//11
data_out(0x0d);//12
data_out(0x0f);//13
data_out(0x11);//14
data_out(0x13);//15
data_out(0x16);//16
data_out(0x18);//17
data_out(0x1b);//18
data_out(0x1d);//19
//====================
data_out(0x20);//20
data_out(0x23);//21
```

```
data_out(0x26);//22
data_out(0x29);//23
data_out(0x2c);//24
data_out(0x30);//25
data_out(0x33);//26
data_out(0x37);//27
data_out(0x3a);//28
data_out(0x3e);//29
//====================
data_out(0x42);//30
data_out(0x46);//31
data_out(0x4a);//32
data_out(0x4e);//33
data_out(0x53);//34
data_out(0x57);//35
data_out(0x5c);//36
data_out(0x60);//37
data_out(0x65);//38
data_out(0x6a);//39
//====================
data_out(0x6f);//40
data_out(0x74);//41
data_out(0x79);//42
data_out(0x7e);//43
data_out(0x83);//44
data_out(0x89);//45
data_out(0x8e);//46
data_out(0x94);//47
data_out(0x9a);//48
data_out(0x9f);//49
//====================
data_out(0xa5);//50
data_out(0xab);//51
data_out(0xb1);//52
data_out(0xb7);//53
data_out(0xbe);//54
data_out(0xc4);//55
data_out(0xca);//56
```

```
data_out(0xd1);//57
data_out(0xd8);//58
data_out(0xde);//59
//====================
data_out(0xe5);//60
data_out(0xec);//61
data_out(0xf3);//62
data_out(0xfa);//63

comm_out(0x2b);//Gamma Table Update

//----------------------------------
comm_out(0x2a);//R Gamma table setting
data_out(0x00);//0
data_out(0x01);//1
data_out(0x02);//2
data_out(0x03);//3
data_out(0x04);//4
data_out(0x05);//5
data_out(0x06);//6
data_out(0x07);//7
data_out(0x08);//8
data_out(0x09);//9
//====================
data_out(0x0a);//10
data_out(0x0b);//11
data_out(0x0d);//12
data_out(0x0f);//13
data_out(0x11);//14
data_out(0x13);//15
data_out(0x16);//16
data_out(0x18);//17
data_out(0x1b);//18
data_out(0x1d);//19
//====================
data_out(0x20);//20
data_out(0x23);//21
data_out(0x26);//22
```

```
data_out(0x29);//23
data_out(0x2c);//24
data_out(0x30);//25
data_out(0x33);//26
data_out(0x37);//27
data_out(0x3a);//28
data_out(0x3e);//29
//====================
data_out(0x42);//30
data_out(0x46);//31
data_out(0x4a);//32
data_out(0x4e);//33
data_out(0x53);//34
data_out(0x57);//35
data_out(0x5c);//36
data_out(0x60);//37
data_out(0x65);//38
data_out(0x6a);//39
//====================
data_out(0x6f);//40
data_out(0x74);//41
data_out(0x79);//42
data_out(0x7e);//43
data_out(0x83);//44
data_out(0x89);//45
data_out(0x8e);//46
data_out(0x94);//47
data_out(0x9a);//48
data_out(0x9f);//49
//====================
data_out(0xa5);//50
data_out(0xab);//51
data_out(0xb1);//52
data_out(0xb7);//53
data_out(0xbe);//54
data_out(0xc4);//55
data_out(0xca);//56
data_out(0xd1);//57
```

```
data_out(0xd8);//58
data_out(0xde);//59
//====================
data_out(0xe5);//60
data_out(0xec);//61
data_out(0xf3);//62
data_out(0xfa);//63

comm_out(0x2b);//Gamma Table Update
//---------------------------------
comm_out(0xe0);//COM Number
data_out(0x1b);

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
```

```
comm_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(0x1c);//R
data_out(0x09);//G
data_out(0x12);//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(0x21);

comm_out(0x82);//Return Length Setting
data_out(0x5e);
```

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

for(i=0;i<28;i++)
 {
  for(j=0;j<95;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(0x21);

comm_out(0x82);//Return Length Setting
data_out(0x5e);

comm_out(0x83);//Display Data Write

for(i=0;i<28;i++)
 {
  for(j=0;j<95;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(0x21);

comm_out(0x82);//Return Length Setting
data_out(0x5e);

comm_out(0x83);//Display Data Write

for(i=0;i<28;i++)
 {
  for(j=0;j<95;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(0x21);
```

```
comm_out(0x82);//Return Length Setting
data_out(0x5e);

comm_out(0x83);//Display Data Write

for(i=0;i<28;i++)
 {
   for(j=0;j<95;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(0x2b);//R
data_out(0x0e);//G
data_out(0x1a);//B
}
```

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

```
Brightness_mode2 (void);
{
comm_out(0xb8);//Brightness
data_out(0x1c);//R
data_out(0x09);//G
data_out(0x12);//B
}
```

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

```
Brightness_mode3 (void);
{
comm_out(0xb8);//Brightness
data_out(0x10);//R
data_out(0x04);//G
data_out(0x0b);//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(0x01);//G
data_out(0x07);//B
}
```

Thank You