

Raspberry Pi Expansion Module User Manual

52PI-RPI-OLED

Maximum Power at Minimum Size

www.52pi.cn

Raspberry Pi 0.96' OLED Display Module User
Manual

Contact Us

Raspberry Pi Geek Forum

admin@52pi.net

www.52pi.cn

Version: 1.0

Data: 2014.12

Raspberry Pi Geek Forum Copyright Statement:



Raspberry Pi 0.96' OLED Display Module User Manual

The document only describes the information about the product, however, it cannot guarantee the product function and performance. If the document content or the product feature and tech spec included in the document are changed, it will be notified further.

Content in the document might be outdated. Our cooperation cannot promise to update this information.

Some information in the document might be disabled in your local area, that includes product and service. You can consult with the contact and agency in your local area.

Copyright to the document belongs to Raspberry Pi Geek Forum. Users can only use this content after they get authorization from our company or other obligee. However, users cannot copy, paraphrase, or create similar devices or products.

The document's final right to interpret and be used as the final interpretation belongs to Raspberry Pi Geek Forum.

More information:

Get more product and support, please contact Raspberry Pi Geek Forum (www.52pi.cn)

Attention:

Due to technical requirements of components, please do not touch directly connected components. The core board and development system contain static-sensitive devices. Quiet electrical charges can easily accumulate in the human body, and the device cannot detect possible damage to equipment. It is recommended to take anti-static measures, it is recommended not to touch, stored in anti-static effect devices.



Raspberry Pi Expand Module Serial:

Raspberry Pi 0.96' OLED Display Module User Manual

Infinity cascade IO expand module:

This module is designed specifically for the Raspberry Pi IO expansion modules. The module expand 32 IO, Multiple modules can cascade, infinity cascade, infinity GPIO.

I2C GPIO expand module:

This module is designed specifically for the Raspberry Pi IO expansion modules. The Module use I2C bus to connect to Raspberry Pi. The module expand 8 Bidirectional GPIO and wit isolation protection function which can effectively excessive external voltage. There are 8 I2C address, you can choose one of them through setup the jumper. Multiple modules can cascade and maximum cascade 8 modules!

Prototype development module:

The Prototype development module is designed specifically for the Raspberry Pi. The module suitable enthusiasts and user can weld peripheral to the module; The module expand some amphenol connector and some SMT, so the user can finish prototype test easily.

Berryclip expand module:

The BerryClip module is designed specifically for learning how to use the GPIO of Raspberry Pi. There are 6 multiple color LED, 1 button and 1 Buzzer on the module.

Berryclip(DIY) expand module:

The module is not the end product, you need weld them by yourself. The function of the module is the same as BerryClip module.

UNO compatibility module:

The module makes Raspberry Pi compatible with Arduino Uno and many Arduino Shields. The module's GPIO is the same as Arduino Uno and you can choose the voltage of GPIO between 5V or 3V through setup jumper.

T Electric level convert module:

The module expand the GPIO of Raspberry Pi to breadboard. It convert 3.3V electric to 5V electric level, then the Raspberry Pi can connect many 5V electric level peripheral.

IO extraction module:

The module expand all of GPIO of Raspberry Pi to breadboard.

RTC expand module:

The RTC module is specifically designed for Raspberry Pi. There is a 0.2uF Super Capacitor on the board to keep the real time for a long time after the Raspberry Pi

Raspberry Pi 0.96' OLED Display Module User Manual

has power off.

AD/DA expand module:

The AD/DA module is specifically designed for Raspberry Pi. There is 8bit high resolution DAC and ADC on one chip.

I2C electric level conversion expand module:

The I2C electric level conversion module is specifically designed for Raspberry Pi. The module convert the 3.3V of I2C electric level to 5V level or convert 5V to 3.3V.

I2C electric level conversion expand module:

The Serial Port module is specifically designed for Raspberry Pi. The module use Maxim MAX232 chip and DB9 port. So the user don't need to connect Dupont Line and that avoid wrong wiring. User can use this mod

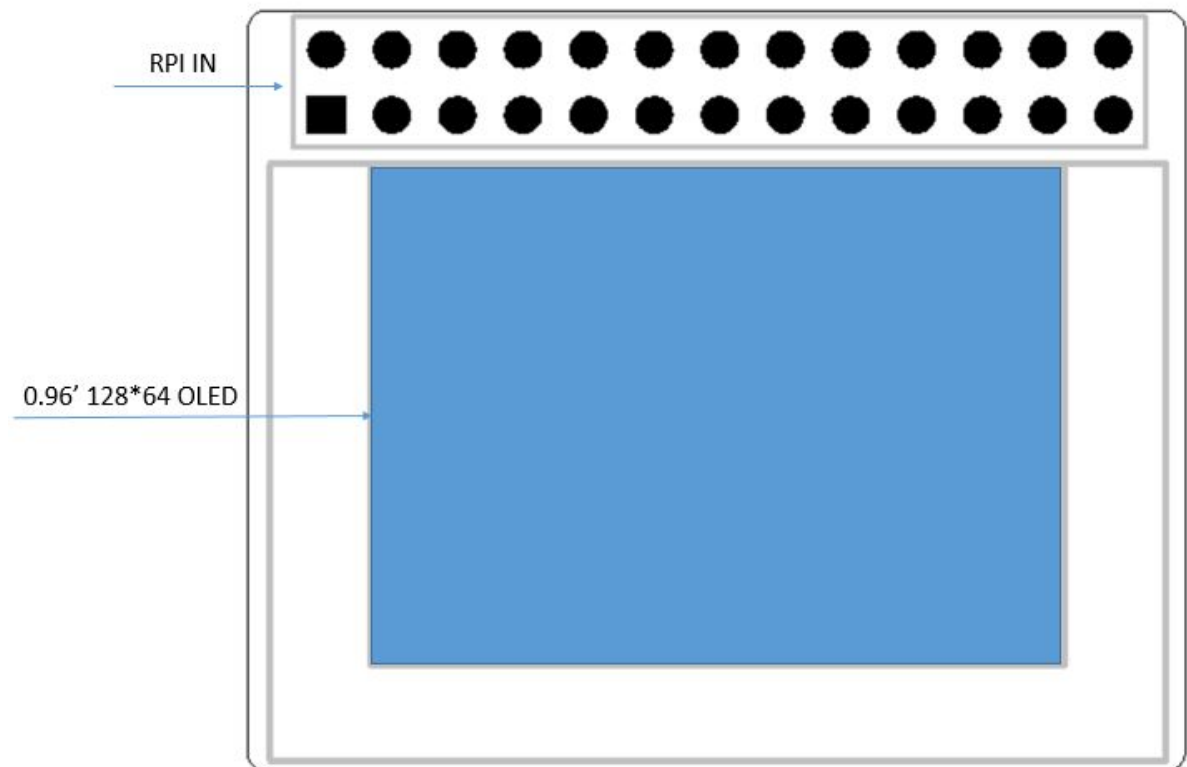
Product name :

Raspberry Pi 0.96' OLED display Module

Description:

Raspberry Pi 0.96' OLED display module is designed by 52pi which is made by 128x64 yellow and blue OLED pixels , diagonal is only 0.96 inch . The controller is SSD1306, communicates via 4-wire SPI, the display have its own backlight, no backlight LED is required ,This reduces the power required to run the OLED and is why the display has such high contrast , extremely wide viewing angle and extremely operating temperature

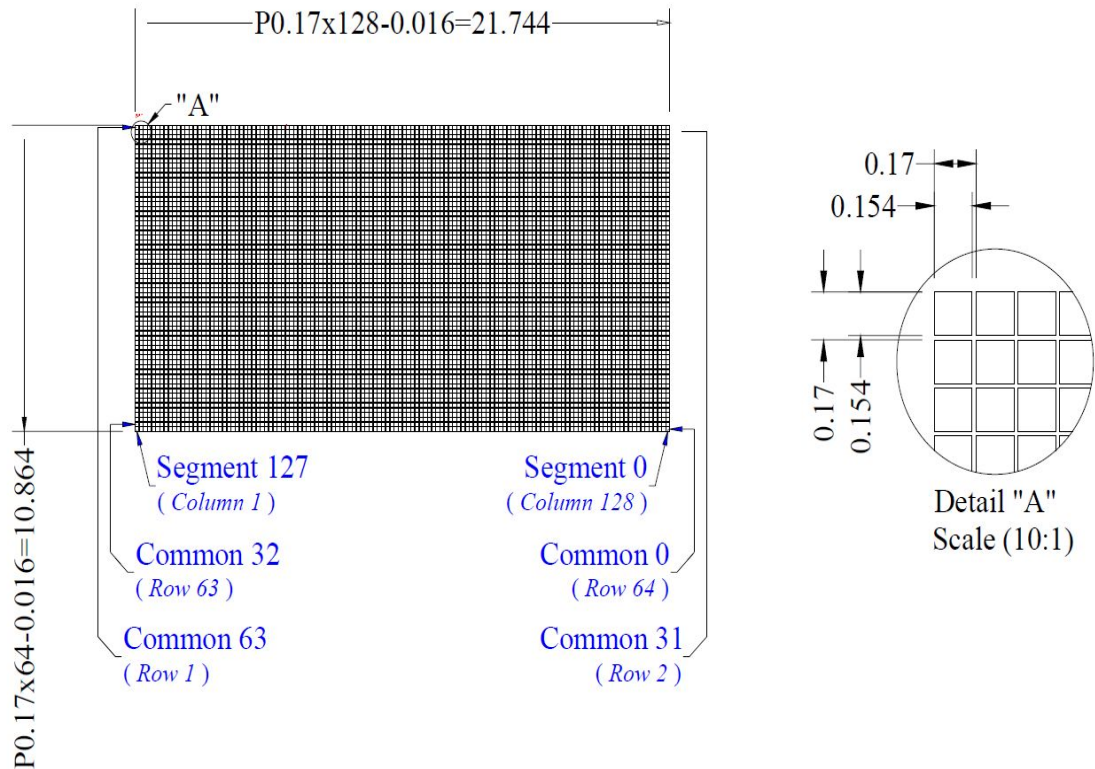
Product Overview:



Mechanical specifications:

- Outline Drawing: 370x300(mm)
- Number of pixels: 128x64
- Panel size:26.70x19.26x1.45(mm)
- Active area: 21.74x10.864(mm)
- Pixel pitch 0.17x0.17(mm)
- Pixel size: 0.154x0.154(mm)
- Weight: 15g

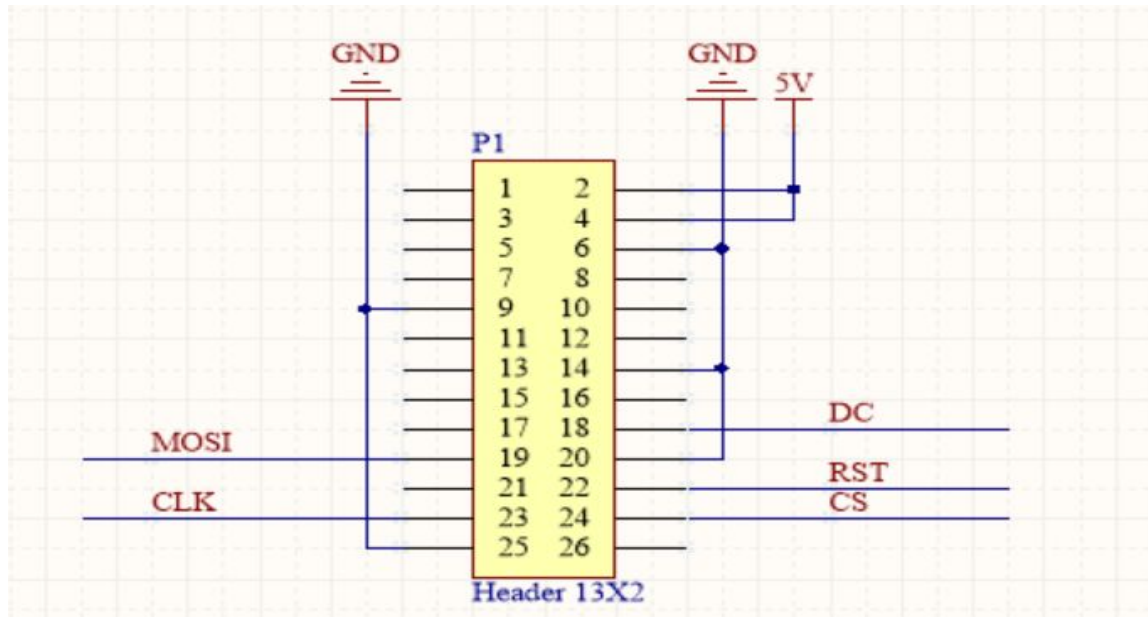
Active area & pixel construction :



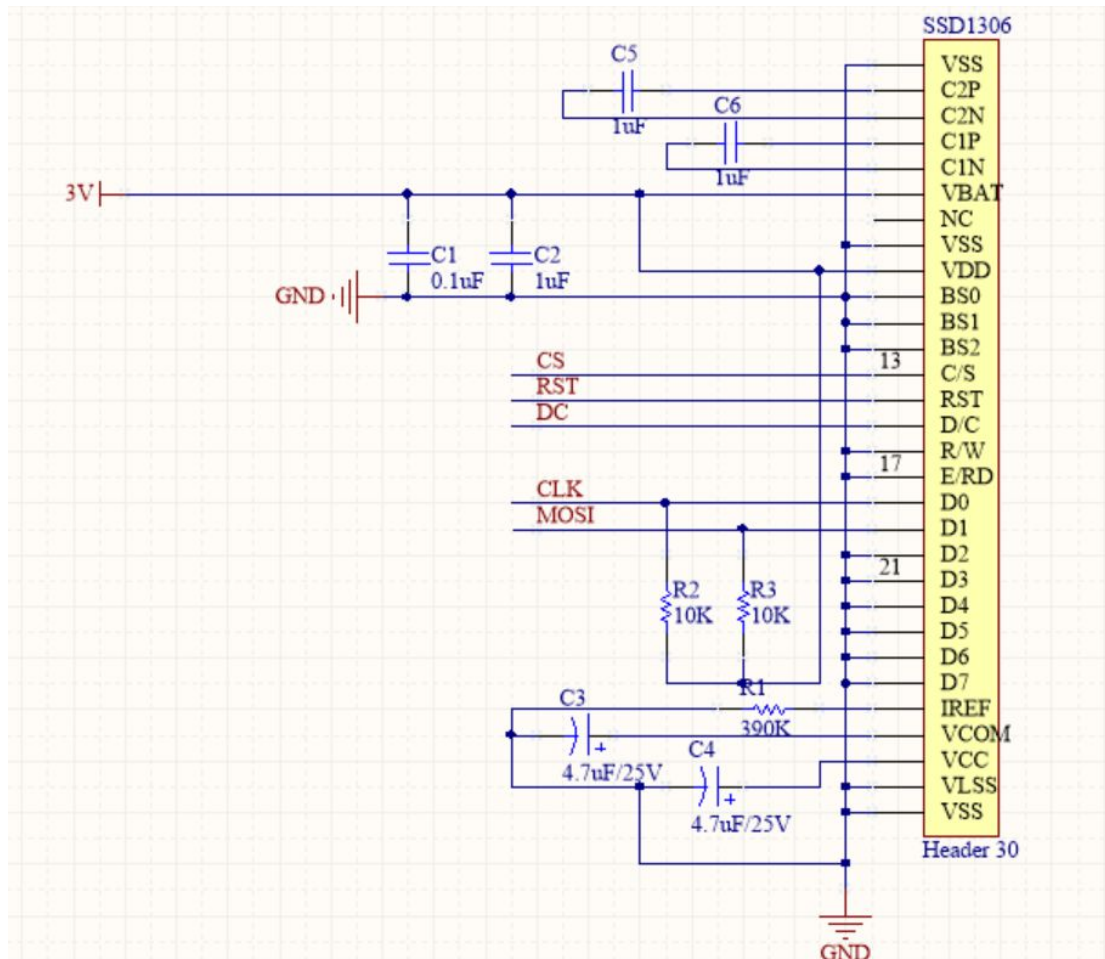
Product Features:

- Ultra low power supply
- Maximum current 100uA
- Keep up current 12.25uA
- Hardware keep up display
- 3.3V work voltage ,([MIN 1.6V] ,[MAX:3.6V])
- Internal DC/DC voltage converter
- SPI driver support , AT 64MHz[TYP] 125MHZ[OB]
- FP frequency: 64fps[TYP] 368fps[OB]

Hardware conection

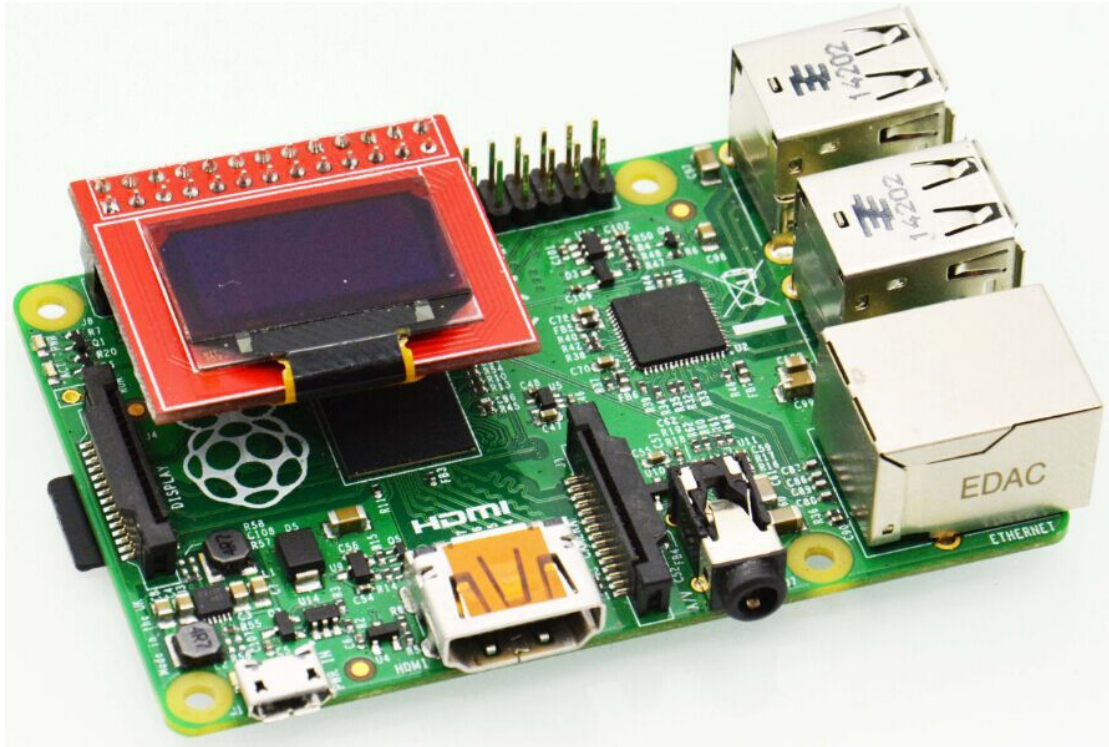


Schematic diagram:



Method Of Use:

Directly mounted on raspberry pi(A、 B、 B+)and Pi 2



Example and test code:

```
#include <stdio.h>
#include <stdlib.h>

#include <wiringPi.h>

#define byte    unsigned char
#define word    unsigned int
#define dword  unsigned long

#define XLevelL      0x00
#define XLevelH      0x10
#define XLevel      ((XLevelH&0x0F)*16+XLevelL)
#define Max_Column  128
#define Max_Row      64
#define Brightness  0xCF

#define X_WIDTH 128
#define Y_WIDTH 64

//=====

void LCD_Init(void);
void LCD_CLS(void);
```

Raspberry Pi 0.96' OLED Display Module User Manual

```
void LCD_P6x8Str(byte x,byte y,byte ch[]);
void LCD_P8x16Str(byte x,byte y,byte ch[]);
void LCD_P14x16Ch(byte x,byte y,byte N);
void LCD_Fill(byte dat);
void Draw_BMP(byte bmp[]);
```

//字符库

```
const unsigned char F6x8[][6] =
{
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 }, // sp
    { 0x00, 0x00, 0x00, 0x2f, 0x00, 0x00 }, // !
    { 0x00, 0x00, 0x07, 0x00, 0x07, 0x00 }, // "
    { 0x00, 0x14, 0x7f, 0x14, 0x7f, 0x14 }, // #
    { 0x00, 0x24, 0x2a, 0x7f, 0x2a, 0x12 }, // $
    { 0x00, 0x62, 0x64, 0x08, 0x13, 0x23 }, // %
    { 0x00, 0x36, 0x49, 0x55, 0x22, 0x50 }, // &
    { 0x00, 0x00, 0x05, 0x03, 0x00, 0x00 }, // '
    { 0x00, 0x00, 0x1c, 0x22, 0x41, 0x00 }, // (
    { 0x00, 0x00, 0x41, 0x22, 0x1c, 0x00 }, // )
    { 0x00, 0x14, 0x08, 0x3E, 0x08, 0x14 }, // *
    { 0x00, 0x08, 0x08, 0x3E, 0x08, 0x08 }, // +
    { 0x00, 0x00, 0x00, 0xA0, 0x60, 0x00 }, // ,
    { 0x00, 0x08, 0x08, 0x08, 0x08, 0x08 }, // -
    { 0x00, 0x00, 0x60, 0x60, 0x00, 0x00 }, // .
    { 0x00, 0x20, 0x10, 0x08, 0x04, 0x02 }, // /
    { 0x00, 0x3E, 0x51, 0x49, 0x45, 0x3E }, // 0
    { 0x00, 0x00, 0x42, 0x7F, 0x40, 0x00 }, // 1
    { 0x00, 0x42, 0x61, 0x51, 0x49, 0x46 }, // 2
    { 0x00, 0x21, 0x41, 0x45, 0x4B, 0x31 }, // 3
    { 0x00, 0x18, 0x14, 0x12, 0x7F, 0x10 }, // 4
    { 0x00, 0x27, 0x45, 0x45, 0x45, 0x39 }, // 5
    { 0x00, 0x3C, 0x4A, 0x49, 0x49, 0x30 }, // 6
    { 0x00, 0x01, 0x71, 0x09, 0x05, 0x03 }, // 7
    { 0x00, 0x36, 0x49, 0x49, 0x49, 0x36 }, // 8
    { 0x00, 0x06, 0x49, 0x49, 0x29, 0x1E }, // 9
    { 0x00, 0x00, 0x36, 0x36, 0x00, 0x00 }, // :
    { 0x00, 0x00, 0x56, 0x36, 0x00, 0x00 }, // ;
    { 0x00, 0x08, 0x14, 0x22, 0x41, 0x00 }, // <
    { 0x00, 0x14, 0x14, 0x14, 0x14, 0x14 }, // =
    { 0x00, 0x00, 0x41, 0x22, 0x14, 0x08 }, // >
    { 0x00, 0x02, 0x01, 0x51, 0x09, 0x06 }, // ?
    { 0x00, 0x32, 0x49, 0x59, 0x51, 0x3E }, // @
```

Raspberry Pi 0.96' OLED Display Module User Manual

```
{ 0x00, 0x7C, 0x12, 0x11, 0x12, 0x7C }, // A
{ 0x00, 0x7F, 0x49, 0x49, 0x49, 0x36 }, // B
{ 0x00, 0x3E, 0x41, 0x41, 0x41, 0x22 }, // C
{ 0x00, 0x7F, 0x41, 0x41, 0x22, 0x1C }, // D
{ 0x00, 0x7F, 0x49, 0x49, 0x49, 0x41 }, // E
{ 0x00, 0x7F, 0x09, 0x09, 0x09, 0x01 }, // F
{ 0x00, 0x3E, 0x41, 0x49, 0x49, 0x7A }, // G
{ 0x00, 0x7F, 0x08, 0x08, 0x08, 0x7F }, // H
{ 0x00, 0x00, 0x41, 0x7F, 0x41, 0x00 }, // I
{ 0x00, 0x20, 0x40, 0x41, 0x3F, 0x01 }, // J
{ 0x00, 0x7F, 0x08, 0x14, 0x22, 0x41 }, // K
{ 0x00, 0x7F, 0x40, 0x40, 0x40, 0x40 }, // L
{ 0x00, 0x7F, 0x02, 0x0C, 0x02, 0x7F }, // M
{ 0x00, 0x7F, 0x04, 0x08, 0x10, 0x7F }, // N
{ 0x00, 0x3E, 0x41, 0x41, 0x41, 0x3E }, // O
{ 0x00, 0x7F, 0x09, 0x09, 0x09, 0x06 }, // P
{ 0x00, 0x3E, 0x41, 0x51, 0x21, 0x5E }, // Q
{ 0x00, 0x7F, 0x09, 0x19, 0x29, 0x46 }, // R
{ 0x00, 0x46, 0x49, 0x49, 0x49, 0x31 }, // S
{ 0x00, 0x01, 0x01, 0x7F, 0x01, 0x01 }, // T
{ 0x00, 0x3F, 0x40, 0x40, 0x40, 0x3F }, // U
{ 0x00, 0x1F, 0x20, 0x40, 0x20, 0x1F }, // V
{ 0x00, 0x3F, 0x40, 0x38, 0x40, 0x3F }, // W
{ 0x00, 0x63, 0x14, 0x08, 0x14, 0x63 }, // X
{ 0x00, 0x07, 0x08, 0x70, 0x08, 0x07 }, // Y
{ 0x00, 0x61, 0x51, 0x49, 0x45, 0x43 }, // Z
{ 0x00, 0x00, 0x7F, 0x41, 0x41, 0x00 }, //[
{ 0x00, 0x55, 0x2A, 0x55, 0x2A, 0x55 }, // 55
{ 0x00, 0x00, 0x41, 0x41, 0x7F, 0x00 }, // ]
{ 0x00, 0x04, 0x02, 0x01, 0x02, 0x04 }, // ^
{ 0x00, 0x40, 0x40, 0x40, 0x40, 0x40 }, // _
{ 0x00, 0x00, 0x01, 0x02, 0x04, 0x00 }, // '
{ 0x00, 0x20, 0x54, 0x54, 0x54, 0x78 }, // a
{ 0x00, 0x7F, 0x48, 0x44, 0x44, 0x38 }, // b
{ 0x00, 0x38, 0x44, 0x44, 0x44, 0x20 }, // c
{ 0x00, 0x38, 0x44, 0x44, 0x48, 0x7F }, // d
{ 0x00, 0x38, 0x54, 0x54, 0x54, 0x18 }, // e
{ 0x00, 0x08, 0x7E, 0x09, 0x01, 0x02 }, // f
{ 0x00, 0x18, 0xA4, 0xA4, 0xA4, 0x7C }, // g
{ 0x00, 0x7F, 0x08, 0x04, 0x04, 0x78 }, // h
{ 0x00, 0x00, 0x44, 0x7D, 0x40, 0x00 }, // i
{ 0x00, 0x40, 0x80, 0x84, 0x7D, 0x00 }, // j
{ 0x00, 0x7F, 0x10, 0x28, 0x44, 0x00 }, // k
```

Raspberry Pi 0.96' OLED Display Module User Manual

```
{ 0x00, 0x00, 0x41, 0x7F, 0x40, 0x00 }, // l
{ 0x00, 0x7C, 0x04, 0x18, 0x04, 0x78 }, // m
{ 0x00, 0x7C, 0x08, 0x04, 0x04, 0x78 }, // n
{ 0x00, 0x38, 0x44, 0x44, 0x44, 0x38 }, // o
{ 0x00, 0xFC, 0x24, 0x24, 0x24, 0x18 }, // p
{ 0x00, 0x18, 0x24, 0x24, 0x18, 0xFC }, // q
{ 0x00, 0x7C, 0x08, 0x04, 0x04, 0x08 }, // r
{ 0x00, 0x48, 0x54, 0x54, 0x54, 0x20 }, // s
{ 0x00, 0x04, 0x3F, 0x44, 0x40, 0x20 }, // t
{ 0x00, 0x3C, 0x40, 0x40, 0x20, 0x7C }, // u
{ 0x00, 0x1C, 0x20, 0x40, 0x20, 0x1C }, // v
{ 0x00, 0x3C, 0x40, 0x30, 0x40, 0x3C }, // w
{ 0x00, 0x44, 0x28, 0x10, 0x28, 0x44 }, // x
{ 0x00, 0x1C, 0xA0, 0xA0, 0xA0, 0x7C }, // y
{ 0x00, 0x44, 0x64, 0x54, 0x4C, 0x44 }, // z
{ 0x14, 0x14, 0x14, 0x14, 0x14, 0x14 } // horiz lines
};
```

const unsigned char F8X16[] =

```
{
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, // 0
  0x00, 0x00, 0x00, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x33, 0x30, 0x00, 0x00, 0x00, // 1
  0x00, 0x10, 0x0C, 0x06, 0x10, 0x0C, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, // 2
  0x40, 0xC0, 0x78, 0x40, 0xC0, 0x78, 0x40, 0x00, 0x04, 0x3F, 0x04, 0x04, 0x3F, 0x04, 0x04, // 3
  0x00, 0x70, 0x88, 0xFC, 0x08, 0x30, 0x00, 0x00, 0x00, 0x18, 0x20, 0xFF, 0x21, 0x1E, 0x00, 0x00, // 4
  0xF0, 0x08, 0xF0, 0x00, 0xE0, 0x18, 0x00, 0x00, 0x00, 0x21, 0x1C, 0x03, 0x1E, 0x21, 0x1E, 0x00, // 5
  0x00, 0xF0, 0x08, 0x88, 0x70, 0x00, 0x00, 0x00, 0x1E, 0x21, 0x23, 0x24, 0x19, 0x27, 0x21, 0x10, // 6
  0x10, 0x16, 0x0E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, // 7
  0x00, 0x00, 0x00, 0xE0, 0x18, 0x04, 0x02, 0x00, 0x00, 0x00, 0x00, 0x07, 0x18, 0x20, 0x40, 0x00, // 8
  0x00, 0x02, 0x04, 0x18, 0xE0, 0x00, 0x00, 0x00, 0x00, 0x40, 0x20, 0x18, 0x07, 0x00, 0x00, 0x00, // 9
  0x40, 0x40, 0x80, 0xF0, 0x80, 0x40, 0x40, 0x00, 0x02, 0x02, 0x01, 0x0F, 0x01, 0x02, 0x02, 0x00, // 10
  0x00, 0x00, 0x00, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01, 0x01, 0x1F, 0x01, 0x01, 0x01, 0x00, // 11
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0xB0, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00, // 12
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, // 13
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x30, 0x30, 0x00, 0x00, 0x00, 0x00, // 14
  0x00, 0x00, 0x00, 0x00, 0x80, 0x60, 0x18, 0x04, 0x00, 0x60, 0x18, 0x06, 0x01, 0x00, 0x00, 0x00, // 15
  0x00, 0xE0, 0x10, 0x08, 0x08, 0x10, 0xE0, 0x00, 0x00, 0x0F, 0x10, 0x20, 0x20, 0x10, 0x0F, 0x00, // 16
  0x00, 0x10, 0x10, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x20, 0x3F, 0x20, 0x20, 0x00, 0x00, // 17
  0x00, 0x70, 0x08, 0x08, 0x08, 0x88, 0x70, 0x00, 0x00, 0x30, 0x28, 0x24, 0x22, 0x21, 0x30, 0x00, // 18
  0x00, 0x30, 0x08, 0x88, 0x88, 0x48, 0x30, 0x00, 0x00, 0x18, 0x20, 0x20, 0x20, 0x11, 0x0E, 0x00, // 19
  0x00, 0x00, 0xC0, 0x20, 0x10, 0xF8, 0x00, 0x00, 0x00, 0x07, 0x04, 0x24, 0x24, 0x3F, 0x24, 0x00, // 20
};
```

Raspberry Pi 0.96' OLED Display Module User Manual

0x00,0xF8,0x08,0x88,0x88,0x08,0x08,0x00,0x00,0x19,0x21,0x20,0x20,0x11,0x0E,0x00,//521
0x00,0xE0,0x10,0x88,0x88,0x18,0x00,0x00,0x00,0x0F,0x11,0x20,0x20,0x11,0x0E,0x00,//622
0x00,0x38,0x08,0x08,0xC8,0x38,0x08,0x00,0x00,0x00,0x00,0x3F,0x00,0x00,0x00,//723
0x00,0x70,0x88,0x08,0x08,0x88,0x70,0x00,0x00,0x1C,0x22,0x21,0x21,0x22,0x1C,0x00,//824
0x00,0xE0,0x10,0x08,0x08,0x10,0xE0,0x00,0x00,0x00,0x31,0x22,0x22,0x11,0x0F,0x00,//925
0x00,0x00,0x00,0xC0,0xC0,0x00,0x00,0x00,0x00,0x00,0x30,0x30,0x00,0x00,0x00,//:26
0x00,0x00,0x00,0x80,0x00,0x00,0x00,0x00,0x00,0x80,0x60,0x00,0x00,0x00,0x00,//;27
0x00,0x00,0x80,0x40,0x20,0x10,0x08,0x00,0x00,0x01,0x02,0x04,0x08,0x10,0x20,0x00,//<28
0x40,0x40,0x40,0x40,0x40,0x40,0x00,0x04,0x04,0x04,0x04,0x04,0x04,0x00,//=29
0x00,0x08,0x10,0x20,0x40,0x80,0x00,0x00,0x00,0x20,0x10,0x08,0x04,0x02,0x01,0x00,//>30
0x00,0x70,0x48,0x08,0x08,0x08,0xF0,0x00,0x00,0x00,0x30,0x36,0x01,0x00,0x00,//?31
0xC0,0x30,0xC8,0x28,0xE8,0x10,0xE0,0x00,0x07,0x18,0x27,0x24,0x23,0x14,0x0B,0x00,//@32
0x00,0x00,0xC0,0x38,0xE0,0x00,0x00,0x00,0x20,0x3C,0x23,0x02,0x02,0x27,0x38,0x20,//A33
0x08,0xF8,0x88,0x88,0x88,0x70,0x00,0x00,0x20,0x3F,0x20,0x20,0x20,0x11,0x0E,0x00,//B34
0xC0,0x30,0x08,0x08,0x08,0x38,0x00,0x07,0x18,0x20,0x20,0x20,0x10,0x08,0x00,//C35
0x08,0xF8,0x08,0x08,0x08,0x10,0xE0,0x00,0x20,0x3F,0x20,0x20,0x20,0x10,0x0F,0x00,//D36
0x08,0xF8,0x88,0x88,0xE8,0x08,0x10,0x00,0x20,0x3F,0x20,0x20,0x23,0x20,0x18,0x00,//E37
0x08,0xF8,0x88,0x88,0xE8,0x08,0x10,0x00,0x20,0x3F,0x20,0x00,0x03,0x00,0x00,0x00,//F38
0xC0,0x30,0x08,0x08,0x08,0x38,0x00,0x00,0x07,0x18,0x20,0x20,0x22,0x1E,0x02,0x00,//G39
0x08,0xF8,0x08,0x00,0x00,0x08,0xF8,0x08,0x20,0x3F,0x21,0x01,0x01,0x21,0x3F,0x20,//H40
0x00,0x08,0x08,0xF8,0x08,0x08,0x00,0x00,0x00,0x20,0x20,0x3F,0x20,0x20,0x00,0x00,//I41
0x00,0x00,0x08,0x08,0xF8,0x08,0x08,0x00,0xC0,0x80,0x80,0x80,0x7F,0x00,0x00,0x00,//J42
0x08,0xF8,0x88,0xC0,0x28,0x18,0x08,0x00,0x20,0x3F,0x20,0x01,0x26,0x38,0x20,0x00,//K43
0x08,0xF8,0x08,0x00,0x00,0x00,0x00,0x00,0x20,0x3F,0x20,0x20,0x20,0x20,0x30,0x00,//L44
0x08,0xF8,0xF8,0x00,0xF8,0xF8,0x08,0x00,0x20,0x3F,0x00,0x3F,0x00,0x3F,0x20,0x00,//M45
0x08,0xF8,0x30,0xC0,0x00,0x08,0xF8,0x08,0x20,0x3F,0x20,0x00,0x07,0x18,0x3F,0x00,//N46
0xE0,0x10,0x08,0x08,0x08,0x10,0xE0,0x00,0x0F,0x10,0x20,0x20,0x20,0x10,0x0F,0x00,//O47
0x08,0xF8,0x08,0x08,0x08,0x08,0xF0,0x00,0x20,0x3F,0x21,0x01,0x01,0x01,0x00,0x00,//P48
0xE0,0x10,0x08,0x08,0x08,0x10,0xE0,0x00,0x0F,0x18,0x24,0x24,0x38,0x50,0x4F,0x00,//Q49
0x08,0xF8,0x88,0x88,0x88,0x88,0x70,0x00,0x20,0x3F,0x20,0x00,0x03,0x0C,0x30,0x20,//R50
0x00,0x70,0x88,0x08,0x08,0x08,0x38,0x00,0x00,0x38,0x20,0x21,0x21,0x22,0x1C,0x00,//S51
0x18,0x08,0x08,0xF8,0x08,0x08,0x18,0x00,0x00,0x00,0x20,0x3F,0x20,0x00,0x00,0x00,//T52
0x08,0xF8,0x08,0x00,0x00,0x08,0xF8,0x08,0x00,0x1F,0x20,0x20,0x20,0x20,0x1F,0x00,//U53
0x08,0x78,0x88,0x00,0x00,0xC8,0x38,0x08,0x00,0x00,0x07,0x38,0x0E,0x01,0x00,0x00,//V54
0xF8,0x08,0x00,0xF8,0x00,0x08,0xF8,0x00,0x03,0x3C,0x07,0x00,0x07,0x3C,0x03,0x00,//W55
0x08,0x18,0x68,0x80,0x80,0x68,0x18,0x08,0x20,0x30,0x2C,0x03,0x03,0x2C,0x30,0x20,//X56
0x08,0x38,0xC8,0x00,0xC8,0x38,0x08,0x00,0x00,0x00,0x20,0x3F,0x20,0x00,0x00,0x00,//Y57
0x10,0x08,0x08,0x08,0xC8,0x38,0x08,0x00,0x20,0x38,0x26,0x21,0x20,0x20,0x18,0x00,//Z58
0x00,0x00,0x00,0xFE,0x02,0x02,0x02,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7F,0x40,0x40,0x40,0x00,//[59
0x00,0x0C,0x30,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x06,0x38,0xC0,0x00,//\60
0x00,0x02,0x02,0x02,0xFE,0x00,0x00,0x00,0x00,0x40,0x40,0x40,0x7F,0x00,0x00,0x00,//]61
0x00,0x00,0x04,0x02,0x02,0x02,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,//^62
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,//_63

Raspberry Pi 0.96' OLED Display Module User Manual

```
if(dat&0x80){digitalWrite(12, 1);} //LCD_SDA=1;
else{digitalWrite(12, 0);} //LCD_SDA=0;
digitalWrite(14, 1); //LCD_SCL=1;
;;;
digitalWrite(14, 0); //LCD_SCL=0; 下降沿送走数据
dat<<=1;
}
digitalWrite(10, 1); //LCD_CS=1;
}
void LCD_WrCmd(unsigned char cmd)
{
    unsigned char i=8;

    digitalWrite(10, 0); //LCD_CS=0;
    digitalWrite(5, 0); //LCD_DC=0;
    digitalWrite(14, 0); //LCD_SCL=0;

    while(i--)
    {
        if(cmd&0x80){digitalWrite(12, 1);} //LCD_SDA=1;
        else{digitalWrite(12, 0);} //LCD_SDA=0;
        digitalWrite(14, 1); //LCD_SCL=1;
        ;;;
        digitalWrite(14, 0); //LCD_SCL=0;;
        cmd<<=1;;
    }
    digitalWrite(10, 1); //LCD_CS=1;
}
void LCD_Set_Pos(unsigned char x, unsigned char y)
{
    LCD_WrCmd(0xb0+y);
    LCD_WrCmd(((x&0xf0)>>4)|0x10);
    LCD_WrCmd((x&0x0f)|0x00);
}
void LCD_Fill(unsigned char bmp_dat)
{
    unsigned char y,x;

    for(y=0;y<8;y++)
    {
        LCD_WrCmd(0xb0+y);
        LCD_WrCmd(0x01);
        LCD_WrCmd(0x10);
    }
}
```

Raspberry Pi 0.96' OLED Display Module User Manual

```
        for(x=0;x<X_WIDTH;x++)
            LCD_WrDat bmp_dat);
    }
}
void LCD_CLS(void)
{
    unsigned char y,x;
    for(y=0;y<8;y++)
    {
        LCD_WrCmd(0xb0+y);
        LCD_WrCmd(0x01);
        LCD_WrCmd(0x10);
        for(x=0;x<X_WIDTH;x++)
            LCD_WrDat(0);
    }
}
void LCD_DLY_ms(unsigned int ms)
{
    unsigned int a;
    while(ms)
    {
        a=1335;
        while(a--);
        ms--;
    }
    return;
}
void LCD_Init(void)
{
    digitalWrite(14, 1); //LCD_SCL=1;
    digitalWrite(10, 1); //LCD_CS=1; //预制 SLK 和 CS 为高电平
    digitalWrite(6, 0); //LCD_RST=0;
    delay(50); //LCD_DLY_ms(50);
    digitalWrite(6, 1); //LCD_RST=1;

    //从上电到下面开始初始化要有足够的时间，即等待 RC 复位完毕

    LCD_WrCmd(0xae); //--turn off oled panel
    LCD_WrCmd(0x00); //---set low column address
    LCD_WrCmd(0x10); //---set high column address
    LCD_WrCmd(0x40); //--set start line address Set Mapping RAM Display Start Line (0x00~0x3F)
    LCD_WrCmd(0x81); //--set contrast control register
    LCD_WrCmd(0xcf); // Set SEG Output Current Brightness
```


Raspberry Pi 0.96' OLED Display Module User Manual

```
LCD_WrCmd(0xa1);/--Set SEG/Column Mapping      0xa0 左右反置 0xa1 正常
LCD_WrCmd(0xc8);/--Set COM/Row Scan Direction  0xc0 上下反置 0xc8 正常
LCD_WrCmd(0xa6);/--set normal display
LCD_WrCmd(0xa8);/--set multiplex ratio(1 to 64)
LCD_WrCmd(0x3f);/--1/64 duty
LCD_WrCmd(0xd3);/--set display offset   Shift Mapping RAM Counter (0x00~0x3F)
LCD_WrCmd(0x00);/--not offset
LCD_WrCmd(0xd5);/--set display clock divide ratio/oscillator frequency
LCD_WrCmd(0x80);/--set divide ratio, Set Clock as 100 Frames/Sec
LCD_WrCmd(0xd9);/--set pre-charge period
LCD_WrCmd(0xf1);/--Set Pre-Charge as 15 Clocks & Discharge as 1 Clock
LCD_WrCmd(0xda);/--set com pins hardware configuration
LCD_WrCmd(0x12);
LCD_WrCmd(0xdb);/--set vcomh
LCD_WrCmd(0x40);/--Set VCOM Deselect Level
LCD_WrCmd(0x20);/--Set Page Addressing Mode (0x00/0x01/0x02)
LCD_WrCmd(0x02);/--
LCD_WrCmd(0x8d);/--set Charge Pump enable/disable
LCD_WrCmd(0x14);/--set(0x10) disable
LCD_WrCmd(0xa4);/-- Disable Entire Display On (0xa4/0xa5)
LCD_WrCmd(0xa6);/-- Disable Inverse Display On (0xa6/a7)
LCD_WrCmd(0xaf);/--turn on oled panel
LCD_Fill(0x00); //初始清屏
LCD_Set_Pos(0,0);
}
```

```
//=====
//函数名: LCD_P6x8Str(unsigned char x,unsigned char y,unsigned char *p)
//功能描述: 写入一组标准 ASCII 字符串
//参数: 显示的位置 (x,y), y 为页范围 0~7, 要显示的字符串
//返回: 无
//=====
void LCD_P6x8Str(unsigned char x,unsigned char y,unsigned char ch[])
{
    unsigned char c=0,i=0,j=0;
    while (ch[j]!='\0')
    {
        c=ch[j]-32;
        if(x>126){x=0;y++;}
        LCD_Set_Pos(x,y);
        for(i=0;i<6;i++)
            LCD_WrDat(F6x8[c][i]);
        x+=6;
    }
}
```

Raspberry Pi 0.96' OLED Display Module User Manual

```
    j++;
  }
}
//=====
//函数名: LCD_P8x16Str(unsigned char x,unsigned char y,unsigned char *p)
//功能描述: 写入一组标准ASCII 字符串
//参数: 显示的位置 (x,y), y 为页范围0~7, 要显示的字符串
//返回: 无
//=====

void LCD_P8x16Str(unsigned char x,unsigned char y,unsigned char ch[])
{
    unsigned char c=0,i=0,j=0;

    while (ch[j]!='\0')
    {
        c =ch[j]-32;
        if(x>120){x=0;y++;}
        LCD_Set_Pos(x,y);
        for(i=0;i<8;i++)
            LCD_WrDat(F8X16[c*16+i]);
        LCD_Set_Pos(x,y+1);
        for(i=0;i<8;i++)
            LCD_WrDat(F8X16[c*16+i+8]);
        x+=8;
        j++;
    }
}

//=====
//函数名: LCD_P14x16Ch(unsigned char x,unsigned char y,unsigned char N)
//功能描述: 输出汉字字符串
//参数: 显示的位置 (x,y), y 为页范围0~7, 要显示的字符串
//返回: 无
//=====

void LCD_P14x16Ch(unsigned char x,unsigned char y,unsigned char N)
{
    unsigned char wm=0,ii = 0;
    unsigned int adder=28*N;
    LCD_Set_Pos(x, y);
    for(wm = 0;wm < 14;wm++)
    {
        LCD_WrDat(F14x16[adder]);
    }
}
```

Raspberry Pi 0.96' OLED Display Module User Manual

```
        adder += 1;
    }
    LCD_Set_Pos(x,y + 1);
    for(wm = 0;wm < 14;wm++)
    {
        LCD_WrDat(F14x16[adder]);
        adder += 1;
    }
}
//=====
//函数名: void Draw_BMP(byte x,byte y)
//功能描述: 显示BMP 图片 128×64
//参数: 起始点坐标(x,y),x 的范围0~127, y 为页的范围0~7
//返回: 无
//=====
void Draw_BMP(byte bmp[])
{
    byte x,y;
    word ii=0;

    for(x=0;x<128;x++)
        for(y=0;y<8;y++)
        {
            LCD_Set_Pos(x,y);
            LCD_WrDat(bmp[ii++]);
            if(ii>=720)
                return;
        }
}
```

```
void main(void)
{
    unsigned char i=0;
    wiringPiSetup();
    pinMode(10, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(14, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(6, OUTPUT);
```

Raspberry Pi 0.96' OLED Display Module User Manual

```
LCD_Init();  
LCD_P8x16Str(0,0,"WWW.52pi.net");  
LCD_P6x8Str(0,2,"RaspberryPi 0.96 OLED");  
LCD_P8x16Str(0,4,"admin@52pi.net");  
  
}}
```