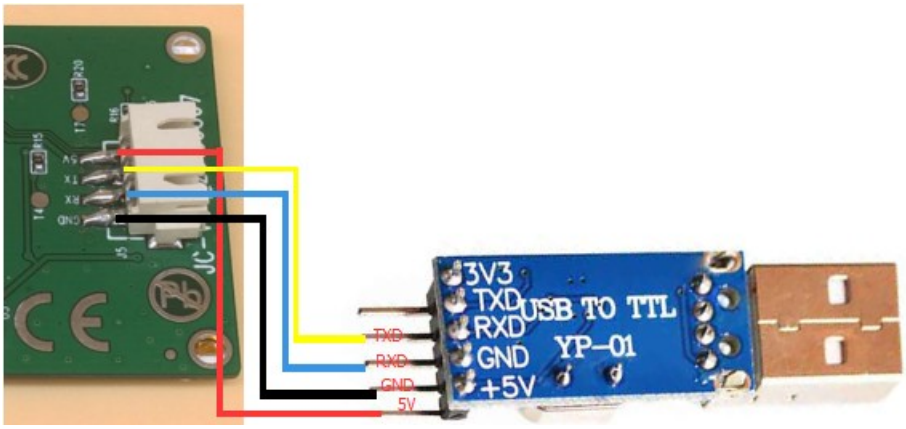
1.Test it by computer

(1) 4 line 5V TX RX GND

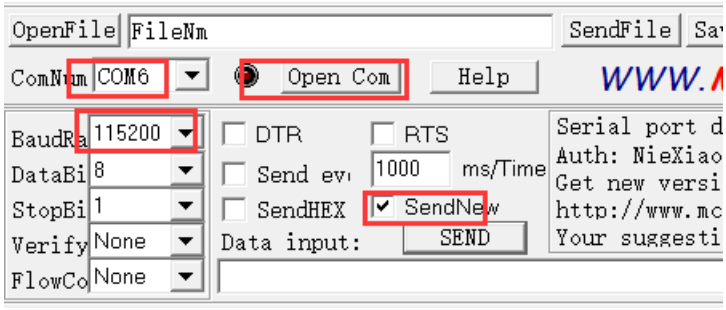
(2) You need a USB to UART module.

(3) How to connect?

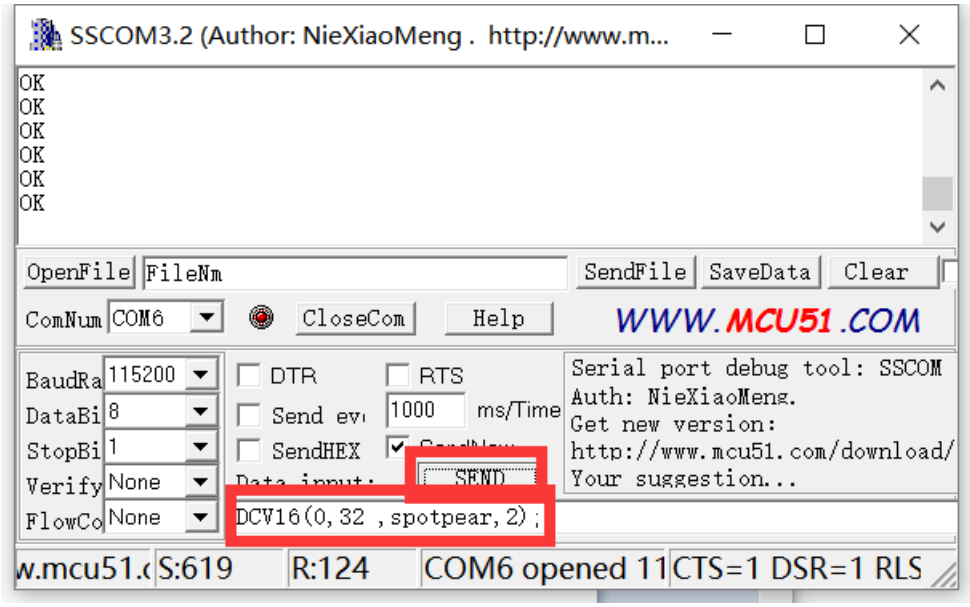


(4) How to install the USB TO TTL module PL2303 driver ?
http://www.spotpear.com/learn/EN/module/USB-TO-TTL-module-PL2303-driver.html

(5) Open sscom32：http://www.spotpear.com/download/LCD/uart/UART-LCD22/Software/sscom32E.rar



(6)
6.1 Test the LCD with following simple CMD:




```
DIR(1);
CIRF(40,80,20,3);
CIR(40,120,20,3);
BOXF(90,30,100,40,1);
BOX(110,40,120,60,1);
DCV32(0,0 ,spotpear,0);
CLR(0);
PL(0,0,220,176,1);
PS(40,40,3);

SBC(1);

DCV16(0,32 ,spotpear,2);
DIR(0);
DCV24(0,0 ,spotpear,0);
```

About above CMD, you can see more detailed introduction later in the article

6.2 How to display a picture ?

->6.2.1 Get a picture （ you need to change your picture to bmp format ）

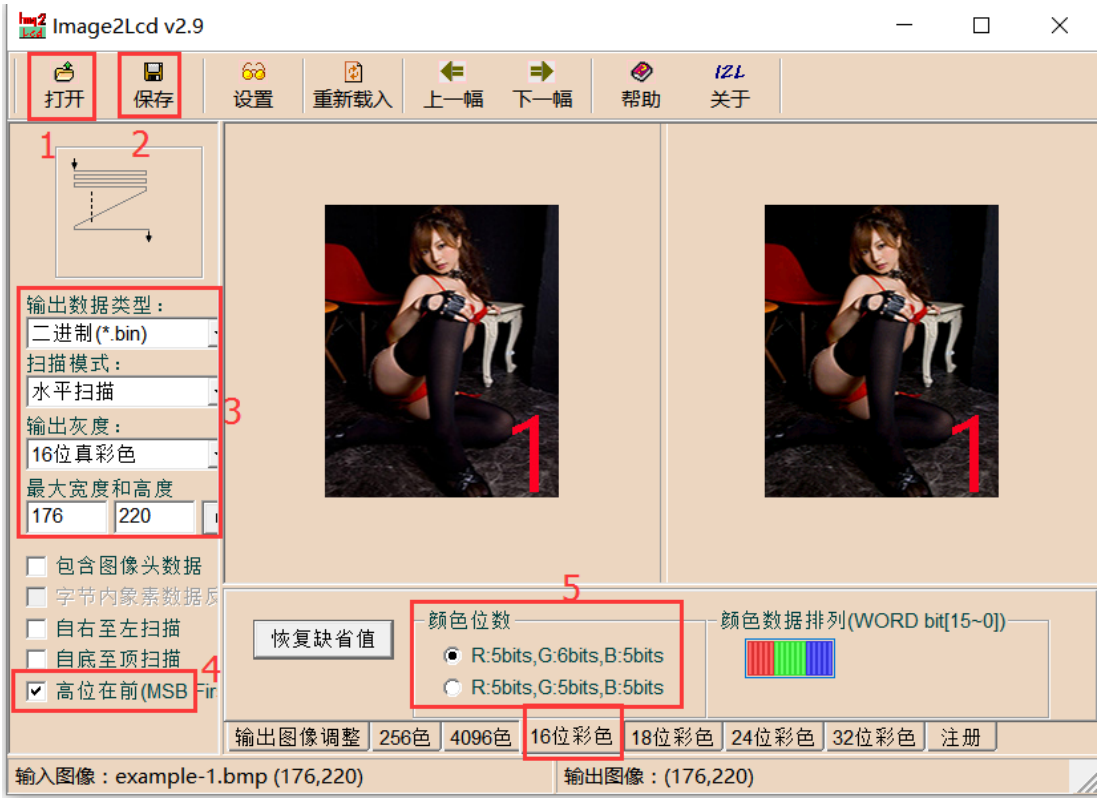
click to download it （ example-1.bmp ）

->6.2.2 change .bmp picture to bin file :

Download this software to change .bmp to .bin file

http://www.spotpear.com/download/LCD/uart/UART-LCD22/Software/Image2Lcd 2.9.rar

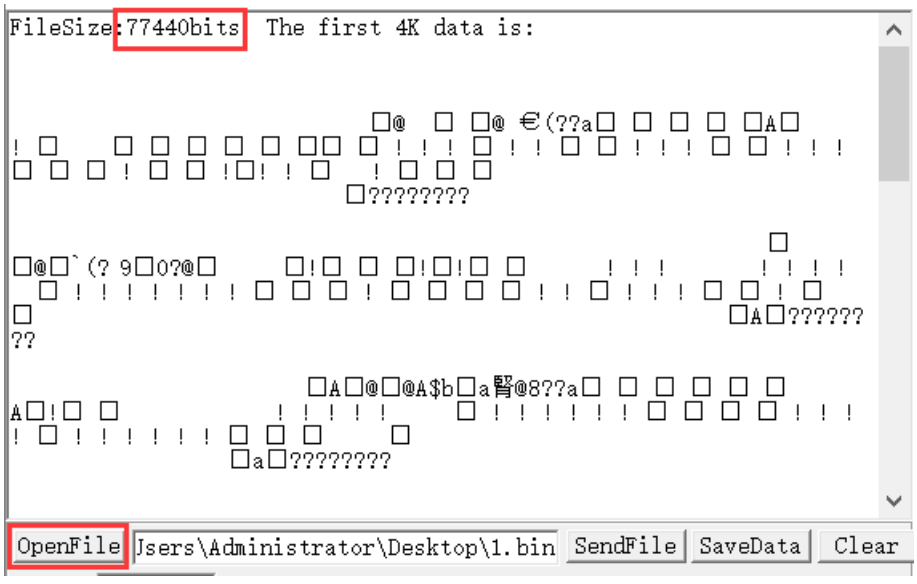
1->open ; set as 3,4,5; 2->save. You will get the .bin file



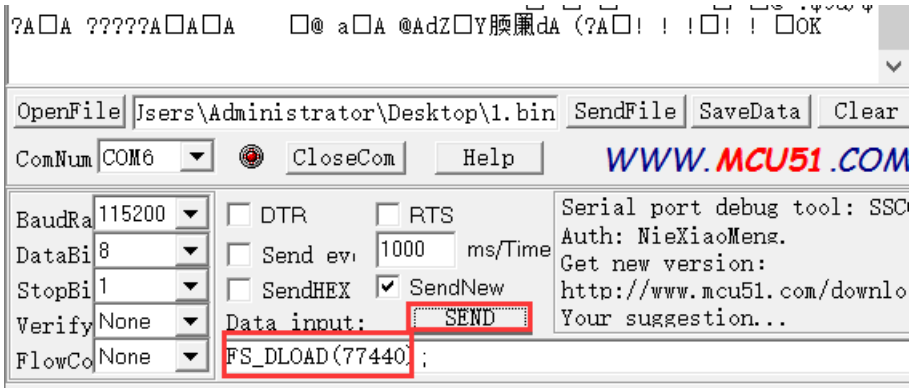
or you click to get it （ The file “1.bin” is made by us ）

->6.2.3 display the picture

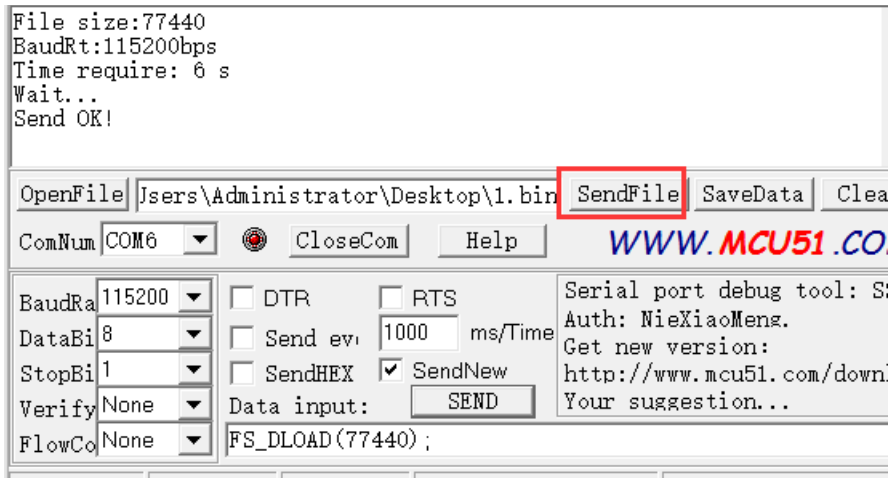
Use SSCOM3.2 to open the .bin file， you will see the file Size 77440bits （220x176 picture）



Enter FS_DLOAD(77440);

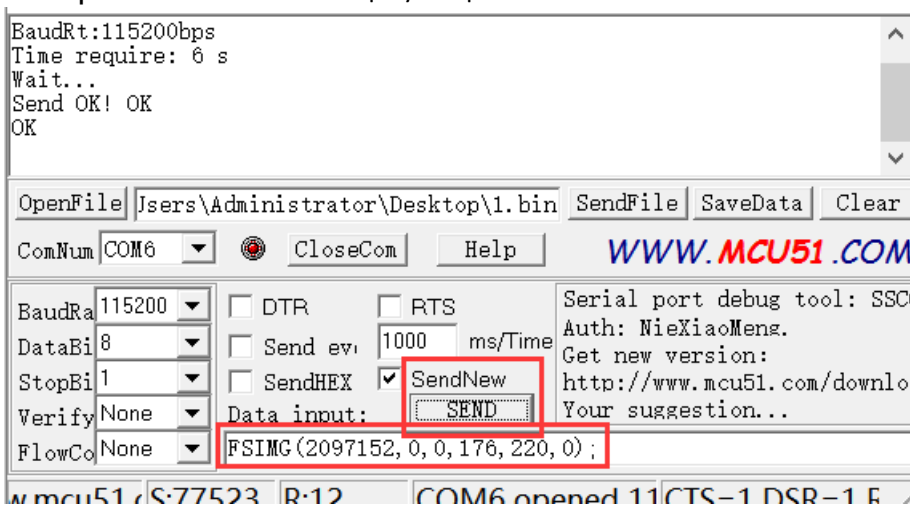


Then send file



Enter FSIMG(2097152,0,0,176,220,0);

And press "SEND"to display the picture



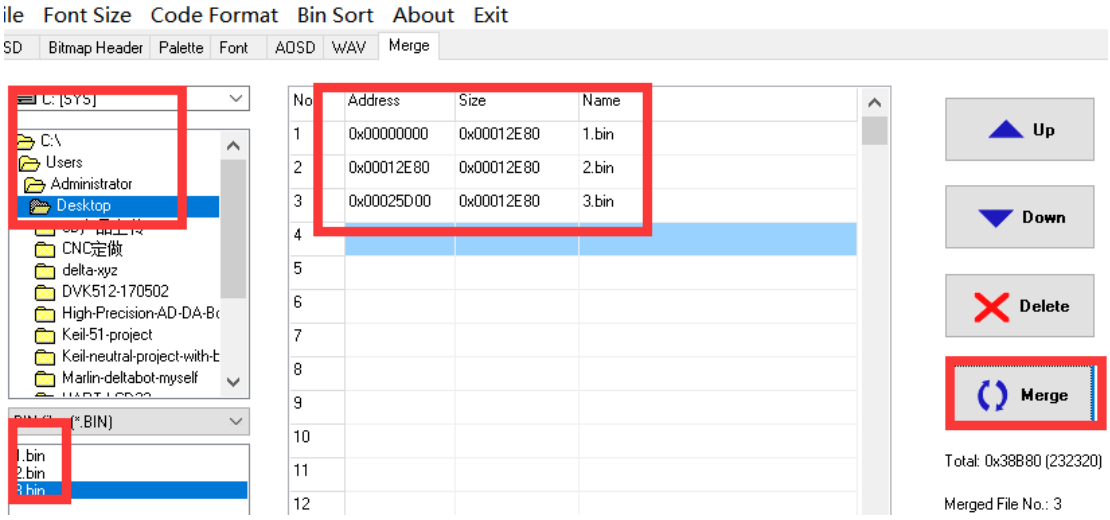
If no picture showed，you need to press "SEND" again

->6.2.4 Download 3 pictures

Do "->6.2.1" and "->6.2.2" to get file "2.bin" and "3.bin" from "example-2.bmp" "example-3.bmp"

Put "1.bin" "2.bin" "3.bin" together to "Pic.BIN" by this software :

http://www.spotpear.com/download/LCD/uart/UART-LCD22/Software/EzOSD_v015T9.rar



Do "->6.2.3" to download "Pic.BIN" (change file "1.bin" to "Pic.BIN") and show "example-1.bmp"

If you want to display example-2.bmp and example-3.bmp， You need to change to enter

FSIMG(2174592,0,0,176,220,0);

to display example-2.bmp

And enter

FSIMG(2252032,0,0,176,220,0);

to display example-3.bmp

PS:display picture， the starting address is 2097152，each adding picture need add 77440，So you can see

FSIMG(2097152,0,0,176,220,0); ->picture 1
FSIMG(2174592,0,0,176,220,0); ->picture 2
FSIMG(2252032,0,0,176,220,0); ->picture 3
FSIMG(2329472,0,0,176,220,0); ->picture 4
FSIMG(2406912,0,0,176,220,0); ->picture 5
FSIMG(2484352,0,0,176,220,0); ->picture 6

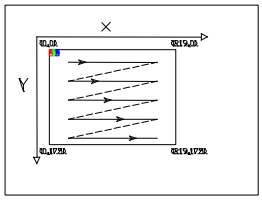
.....

2.Software Description

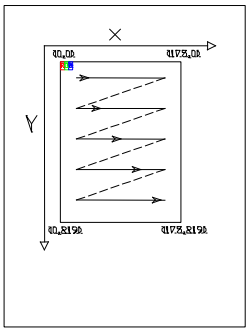
(1)Pre-knowledge

1 Display memory：DDRAM:176*220*2=77440

2 Memory address：

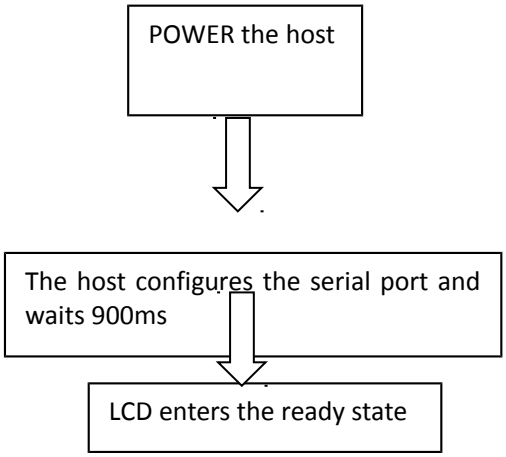


Horizontal screen mode DDRAM address arrangement



Vertical screen mode DDRAM address arrangement

(2) Initialize the LCD



(3) UART command introduction

	CMD	***DESCRIPTION***
Reset	RESET; This command allows the module to enter the software reset, receive this command, the module's peripheral components and system parameters will be restored to the power value.	
Get the version information instructions for the module	VER; Through the VER; you can get the firmware version of this module information, and displayed on the screen	
Set the baud rate	BPS(bps);	The default baud rate is 115200 when

command		the system is powered on.
Clear command	CLR(c);	Note that the range of c is 0 to 15, and if the value of c exceeds 15, the system will not respond to the CMD, and the range of c values will look at the following color list.
	CLR for the script, c for the clear use of the background color, the specific code see the following color list. If you want to fill the screen with black, then CLR (0);	
LCD control CMD	LCDON (on_off) ;	On_off parameters only 0 or 1, the system ignores other parameters.
	LCDON for the script, on_off, respectively, that start or turn off the LCD. Such as LCDON (1); that start LCD, LCDON (0); turn off the LCD.	
Display the LCD	FSIMG(addr,x , y , w,h,mode);	When Mode is 1, the white background of the picture will not be displayed. This mode is used to overlay the icon and the background image. Addr is the flash start address for storing pictures, starting at 2097152
	FSIMG for the script, addr for the picture stored in the flash address, x, y for the picture to be displayed on the screen above the starting position, w for the picture width, h for the picture height, mode for the picture display: 1 for the transparent display , 0 is normal display. Such as FSIMG (2097152,0,0,240,400,1); that from the 2097152 FLASH address removed 240 * 400 pictures and 0,0 position transparent display.	
Image download to FLASH command	FS_DLOAD(SIZE);	Picture will be downloaded to the FLASH 2M high storage space, so from 2M (2097152 position to start storing pictures) a total of 2M This command supports the merger of the picture programming, does not support a single picture file programming.
	FS_DLOAD is the script, and SIZE is the total size of the picture to be downloaded. Such as FS_DLOAD (192000); that 192000 bytes of pictures downloaded to the flash, the total size of the picture can not exceed 2097152 bytes, if the SIZE assignment greater than 2097152 bytes, the system only to identify 2097152 bytes.	
	SDIMG for the script, x, y for the picture to be displayed at the beginning of the screen position, w, h were the width and height of the picture, 'name' for the file name, currently only supports English name. SDIMG (0,0,240,400, '6.bin'); that is, the SD card stored 6.bin file in the module 0,0 position display	
Vertical and horizontal screen switch CMD	DIR(H_V);	The LCD is displayed by default for DIR (0); for vertical screen
	Such as DIR (0); for vertical screen. DIR (1); for horizontal screen	
Set the brightness of the backlight	BL (p);	After the system is powered on, the brightness of the backlight is 20
	where BL is the instruction code, p is the brightness value of the backlight, the adjustment range is: 0 ~ 255, where 0 is full display, 255 is off display.	
Draw points	PS (x, y, c);	This instruction does not apply to large areas of speculation, if there is a need to recommend built-in internal modules
	where PS is the instruction code, x, y is the starting position of the display, c is the color of the point	
Draw lines	PL (x1, y1, x2, y2, c)	Note that the range of c is 0 to 15, and if the value of c exceeds 15, the system will ignore this operation.
	where PL is the instruction code, x1, y1 is the starting point, x2, y2 is the position of the end point, c is the color of the line	
Draw box	BOX (x1, y1, x2, y2, c)	
	where BOX is the instruction code, x1, y1, the position of the starting point, x2, y2 is the position of the end point, c is the color of the box	
Draw box with Filled color	BOXF (x1, y1, x2, y2, c);	

	where BOXF is the instruction code, x1, y1, the position of the starting point, x2, y2 is the position of the end point, c is the color of the box	
Draw a circle	CIR (x, y, r, c);	
	where CIR is the instruction code, x, y is the center of the circle, r is the radius of the circle, c is the circle color	
Draw a circle with Filled color	CIRF (x, y, r, c);	
	where CIRF is the instruction code, x, y is the center of the circle, r is the radius of the circle, c is the color of the circle	
Set background color	SBC (c);	
	where SBC is the instruction code, c is the background color value, and c ranges from 0 to 63.	
Display 16 high character With background color instruction	DCV16 (x, y, * str, c);	
	where DCV16 is the instruction code, x, y is the starting position of the character, * str is the pointer of the character, c is the color of the character	
Display 24 high character With background color instruction	DCV24 (x, y, * str, c);	
	where DCV24 is the instruction code, x, y is the starting position of the character, * str is the pointer of the character, c is the color of the character	
Display 32 high character With background color instruction	DCV32 (x, y, * str, c);	
	where DCV32 is the instruction code, x, y is the starting position of the character, * str is the pointer of the character, c is the color of the character	

Color list

black	0
red	1
green	2
blue	3
yellow	4
cyan-blue	5
purple	6
gray	7
Light gray	8
brown	9
Dark green	10
Navy blue	11
Dark yellow	12
Orange	13
Light red	14
white	15

(4)

Each instruction must end with a semicolon symbol, and each operation must end with a newline symbol.

semicolon symbol ;

semicolon symbol \r\n

(5)Programming examples：

STM32F103RBT6:

```
void uart_init(u32 bound){
        //    GPIO_InitTypeDef GPIO_InitStructure;
          USART_InitTypeDef USART_InitStructure;
          NVIC_InitTypeDef NVIC_InitStructure;
         RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1|RCC_APB2Periph_GPIOA|RCC_APB2Periph_AFIO,
ENABLE);//USART1_TX   PA.9
        GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9;
        GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
        GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
        GPIO_Init(GPIOA, &GPIO_InitStructure);
        //USART1_RX      PA.10
        GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10;
        GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
        GPIO_Init(GPIOA, &GPIO_InitStructure);
       //Usart1 NVIC ÅäÖÃ
        NVIC_InitStructure.NVIC_IRQChannel = USART1_IRQn;
          NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority=3 ;
          NVIC_InitStructure.NVIC_IRQChannelSubPriority = 3;            //
          NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
          NVIC_Init(&NVIC_InitStructure);
          USART_InitStructure.USART_BaudRate = bound;//
          USART_InitStructure.USART_WordLength = USART_WordLength_8b;
          USART_InitStructure.USART_StopBits = USART_StopBits_1;
          USART_InitStructure.USART_Parity = USART_Parity_No;
          USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
          USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;
        USART_Init(USART1, &USART_InitStructure);
        USART_ITConfig(USART1, USART_IT_RXNE, ENABLE);//
        USART_Cmd(USART1, ENABLE);
    }
          void UartSend(char * databuf) //
          {
                u8 i=0;
                while (1)
                {
      if (databuf[i]!=0)//
{
      USART_SendData(USART1, databuf[i]); //
      while(USART_GetFlagStatus(USART1, USART_FLAG_TXE) == RESET){}; //
      i++;
```

```
    }
    else return;
    }
}

int main(void)
{
    SystemInit();//
    delay_init(72);     //
    uart_init(115200);  //
    delay_ms(500);
    for(;;)
    {
    UartSend("SBC(15);DIR(0);FSIMG(2329472,0,0,176,220,0);DIR(1);SBC(10);\r
\n");
        CheckBusy();
        UartSend("DC32(0,0,'spotpear',1);\r\n");
        CheckBusy();
        UartSend("DC24(0,32,'spotpear1',2);\r\n");
        CheckBusy();
        UartSend("DC24(0,56,'spotpear2',4);\r\n");
        CheckBusy();
        UartSend("DC16(0,80,'spotpear3',3);\r\n");
        CheckBusy();
        UartSend("DC16(0,96,'spotpear4',1);\r\n");
        CheckBusy();
        UartSend("DC16(0,112,'spotpear5',1);\r\n");
        CheckBusy();
        UartSend("PS(10,10,14);\r\n");
        CheckBusy();

        UartSend("BOX(120,140,150,160,3);\r\n");
        CheckBusy();
        UartSend("CIRF(70,150,20,1);\r\n");
        CheckBusy();
    while(1);
    }
 }
```

3. Test it by Raspberry Pi/Arduino/51

You can find the demo code at here
http://www.spotpear.com/download/LCD/uart/UART-LCD22/Demo-Code/

3.1 If you use it at raspberry pi. You need connect like this

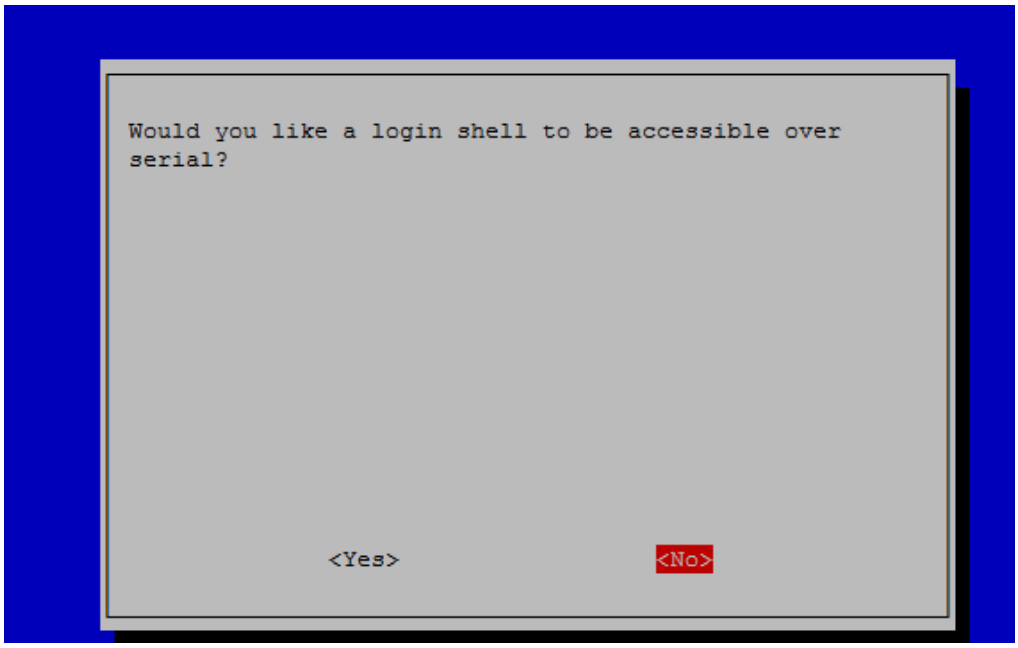
Before you run the program, you need to configure to disable the UART

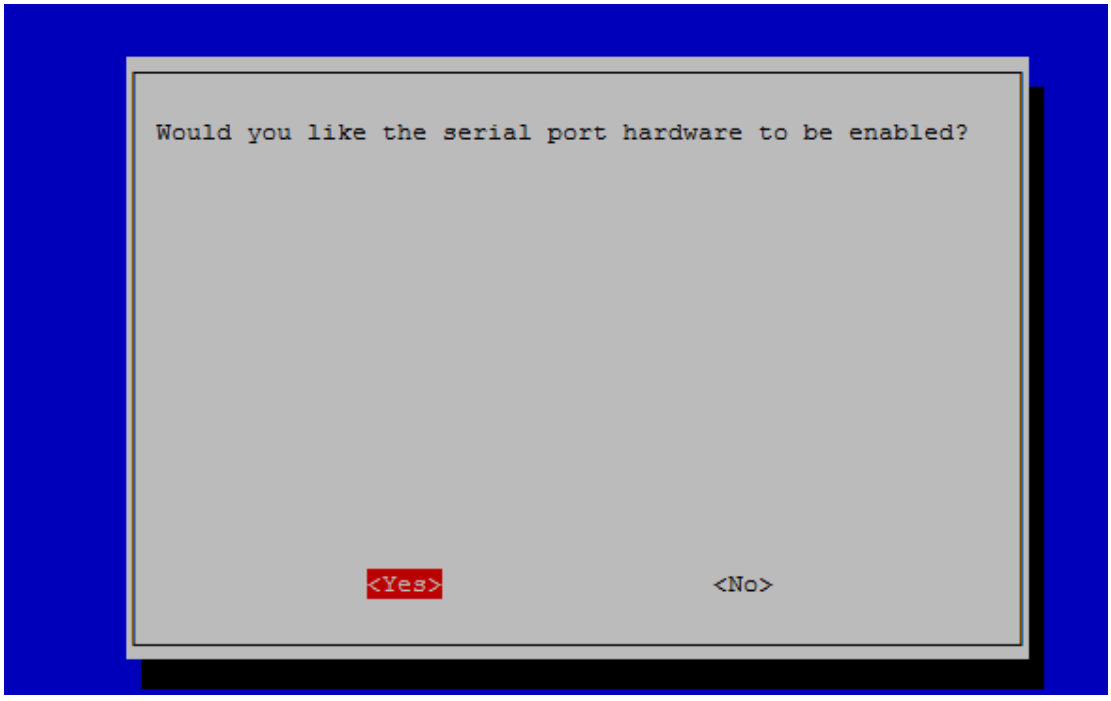
```
sudo raspi-config
```

```
 1 Change User Password          Change password for the default u
 2 Hostname                      Set the visible name for this Pi
 3 Boot Options                  Configure options for start-up
 4 Localisation Options          Set up language and regional sett
 5 Interfacing Options           Configure connections to peripher
 6 Overclock                     Configure overclocking for your P
 7 Advanced Options              Configure advanced settings
 8 Update                        Update this tool to the latest ve
 9 About raspi-config            Information about this configurat
```

```
┤ Raspberry Pi Software Configuration Tool (raspi-config) ├

   P1 Camera                     Enable/Disable connection to the
   P2 SSH                        Enable/Disable remote command lin
   P3 VNC                        Enable/Disable graphical remote a
   P4 SPI                        Enable/Disable automatic loading
   P5 I2C                        Enable/Disable automatic loading
   P6 Serial                     Enable/Disable shell and kernel m
   P7 1-Wire                     Enable/Disable one-wire interface
   P8 Remote GPIO                Enable/Disable remote access to G
```





After reboot. On the Terminal, enter

```
sudo wget http://www.spotpear.com/download/LCD/uart/UART-LCD22/Demo-Code/UARTLCD22-RPi3.tar.gz
sudo tar xvf UARTLCD22-RPi3.tar.gz
cd UARTLCD22-RPi3/
sudo rm UART
sudo make
sudo ./UART
```

3.2 If you use it at ARDUINO. You need connect like this