JLX12864G-086-PN User Manual

(Without font library IC)

table of Conter

Jinglianxun ElectronicsLCD module JLX12864G-086-PN Update date: 2014-3-19

1. Overview

Jinglianxun Electronics focuses on the R&D and manufacturing of LCD screens and LCD modules. The JLX12864G-086 LCD module produced is due Easy to use, clear display, widely used in various man-machine communication panels.

JLX12864G-086 can display 128 columns*64 rows of dot matrix monochrome pictures, or display 16*16 dot matrix Chinese characters 8*4 rows, or Display 16*4 lines of English, numbers and symbols of 8*16 dot matrix. Or display 21*8 lines of English, numbers, and symbols of 5*8 dot matrix.

This product can choose between IC with Chinese font and IC without Chinese font.

2. The characteristics of JLX12864G-086 image type dot matrix liquid crystal module

2.1 The structure is light, thin, with backlight.

2.2 The IC adopts UC1701X, which has powerful functions and good stability

2.3 Low power consumption: When the voltage is 3.3V, the power consumption is low: 1mW without backlight (3.3V*0.3mA), with backlight not more t (3.3V*45mA);

2.4 Display content:

(1) 128*64 dot matrix monochrome pictures, or other monochrome pictures smaller than 128*64 dot matrix;

(2) You can choose 16*16 dot matrix or other dot matrix pictures to compile Chinese characters, and calculate according to 16*16 dot matrix C Display 8 characters*4 lines;

(3) According to 12*12 dot matrix Chinese characters, it can display 10 characters*4 lines;

(4) According to 8*16 dot matrix Chinese characters, it can display 16 characters*4 lines;

(5) According to 5*8 dot matrix Chinese characters, it can display 21 characters*8 lines;

2.5 Strong command function.

2.6 The interface is simple and convenient: a 4-wire SPI serial interface is adopted.

2.7 Wide working temperature: -20℃-70℃;

Page 3

3. Dimensions and interface pin functions

Figure 1. Dimensions

Page 4

Interface pin function of the module

Table 1: Interface pin function of the module

Lead number	symbol	I/O	name	Features
1	ROM_IN	I	Namely font IC interface (SI) serial data input	1. When choosing a product with a font library, please refer to:
2	ROM_OUT	O	Namely font IC interface (SO) serial data output	(1) Font library IC: JLX-GB2312 manual
3	ROM_SCK	I	I.e. font IC interface (SCLK) Serial clock input	(2) Chinese font library programming of JLX12864G-086-PC Manual
4	ROM_CS	I	Font library IC interface (CS#) chip select input	2. Empty when the font library is not used
5	LEDA	I	Backlight power	Positive backlight power supply, same as VDD voltage (5V or 3.3V)
6	VSS	I	Grounded	0V
7	VDD	I	Circuit power	5V, or 3.3V optional
8	SCLK	I	Serial clock	Serial clock input
9	SDA	I	Serial data	Serial data input
10	RS	I	Register selection signal	H: Data register 0: Instruction register (written as "CD" on the IC data)
11	RESET	I	Reset	Low level reset, after the reset is completed, return to high level, the LCD module starts to work
12	CS	I	Chip Select	Low-level chip select

4. Basic principles

4.1 Liquid crystal screen (LCD)

A 128×64 dot matrix is arranged on the LCD, 128 column signals are connected to the driver IC, and 64 row signals are also connected to the driver IC. IC is bonded on LCD glass (this processing technology is called COG).

4.2 Working electrical diagram:

Figure 2 is the circuit block diagram of the JLX12864G-0086 image dot matrix module, which is composed of the driver IC ST7565R and several resisto

Figure 2: Circuit block diagram of JLX12864G-086-PN image dot matrix LCD module

Page 5

4.3 Backlight parameters

This type of LCD module is equipped with LED backlight. Its performance parameters are as follows:

Working temperature: -20∽+70°C;

Storage temperature: -30∽+80°C;

The backlight panel can choose green or white.

The normal working current is: 24∽60mA (the number of LED lights is 3, and each light is 8-20 mA)

Working voltage: the same as VDD voltage (the voltage of the LED light is 3.0V, because a current limiting resistor has been added to the PCB, it can b Voltage);

5. Technical parameters

5.1 Maximum limit parameters (exceeding the limit parameters will damage the LCD module)

name	symbol	standard value			unit
		The smallest	typical	maximum	
Circuit power	VDD-VSS	-0.3		7.0	V
Operating temperature		-20		+70	°C
Storage temperature		-30		+80	°C

Table 2: Maximum limit parameters

5.2 Direct current (DC) parameters

name	symbol	Test Conditions	standard value			unit
			The smallest	typical value is the	largest	
Operating Voltage	VDD	Choose 3.3V products	2.4	3.3	3.6	V
		Choose 5.0V product	4.0	5.0	5.5	V
Backlight working voltage	VLED		2.9	3.0	3.1	V
Input high level	VIH	-	2.2		VDD	V
Input low level	VI0	-	-0.3		0.6	V
Output high level	VOH	IOH = 0.2mA	2.4		-	V
Output low level	VO0	IO0 = 1.2mA	-		0.4	V
Module working current	IDD	VDD = 3.0V	-		1.0	mA
Backlight working current	ILED	VLED=3.0V (Total 3 LED lights in parallel)	twenty four	45	60	mA

Table 3: Direct current (DC) parameters

6. Read and write timing characteristics

6.1 Sequence diagram of serial interface:

Chip select must be low level when transmitting commands/data. CD (that is, RS) is low level: transmission command, CD (that is, RS) is high level: tra When the SCK rising edge, SDI transmits 1 bit of instruction/data. The high bit D7 is transmitted first, and 8 bits are transmitted as a byte.

Writing Data from CPU to UX1701X (Writing Data from CPU to UX1701X)

Figure 4. Writing Data from CPU to UX1701X (Writing Data from CPU to UX1701X)

6.2 Serial interface: timing requirements (AC parameters):

Timing requirements for writing data to UX1701X:

VDD =2.5~3.3V,Ta = 25℃ Table 4.

project	symbol	Test Conditions	Limit value			unit
			MIN	TYPE	MAX	
4-wire SPI serial port clock cycle (4-line SPI Clock Period)	tCYS8	Pin: SCK	60	-	-	ns
Keep SCK high pulse width (SCK "H" pulse width)	tHPS8	Pin: SCK	15	-	-	ns
Keep SCK low pulse width (SCK "L" pulse width)	tLPS8	Pin: SCK	15	-	-	ns
Address establishment time (Address setup time)	tASS8	Pin: RS	0	-	-	ns
Address hold time (Address hold time)	tAHS8	Pin: RS	0	-	-	ns
Data establishment time (Data setup time)	tDSS8	Pin: SDA	12	-	-	ns
Data retention time (Data hold time)	tDHS8	Pin: SDA	0	-	-	ns
Chip select creation time (Chip Select setup time)	tCSSS8	Pin: CS	5	-	-	ns
Chip select hold time (Chip Select hold time)	tCSHS8	Pin: CS	5	-	-	ns

6.3 Timing requirements for reset after power on (RESET CONDITION AFTER POWER UP):

Figure 5: Sequence of reset after power-on

Table 6: Timing requirements for reset after power-on

project	symbol	Test Conditions	Limit value			unit
			MIN	TYPE	MAX	
Reset hold low time t_{RW}		Pin: RES	3.0 us	-	-	
Reset to internal state delay	tR	Pin: RES And IC internal state	6.0 ms	-	-	

7. Command function:

7.1 Instruction list

Instruction list Table 8.

Command name	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Description
(1) Display on/off (Display on/off)	0	1	0	1	0	1	1	1	0 1	Display on/off: 0XAE: off, 0XAF : on
(2) Display initial line settings (Display start line set)	0	0	1	Display the initial row address, a total of 6 digits						Set the display initial line of the display memory, the settable value is 0X40 ~ 0x7f , representing the first 0 ~ 63 lines for the LCD screen is generally set to 0x40
(3) Page address setting (Page address set)	0	1	0	1	1	Display page address, 4 digits in total				Set the page address. Every 8 lines is a page, 64 lines are divided into 8 Page, the settable value is: 0XB0 ~ 0XB8 correspond to the first A page into the first nine pages, the ninth page is a single line in FIG. Target, this LCD screen does not have this line of icons, so the set value Is 0XB0 ~ 0xB7 respectively first correspondence a this page first to eight pages.
(4) The upper 4 bits of the column address are set	0	0	0	0	1	The upper 4 bits of the column address				The upper 4 bits and the lower 4 bits together form the column address, specifying 128 One of the columns. For example, the 100th column of the LCD module The address hexadecimal is 0x 6 4 , then this instruction consists of 2 Expressed in bytes: 0x1 6 ,0x0 4
Set the lower 4 bits of the column address	0	0	0	0	0	Lower 4 bits of column address				
(5) Read status (Status read)	0	status				0	0	0	0	Parallel port: read the current status of the driver IC, not available for serial port This instruction. This LCD module uses a serial interface and does not have This feature .
(6) Write display data to the LCD screen (Display data write)	1	8-bit display data								Write data from CPU to LCD screen, each bit corresponds to one Dot matrix, 1 byte corresponds to 8 vertical dot matrix
(7) Read the display data of the LCD screen (Display data read)	1	8-bit display data								Parallel port: read the number of dots displayed on the LCD screen according to. This command cannot be used in the serial port. This LCD module uses a serial interface and does not have this function .
(8) Display column address increase or decrease (ADC select)	0	1	0	0	0	0	0	0 1		Display column address increase or decrease: 0xA0 : Conventional: column address from left to right, 0xA1 : Reverse: column address from right to left
(9) Display positive display/reverse display (Display normal/reverse)	0	1	0	1	0	0	1	1	0 1	Display front/reverse display: 0xA6 : Conventional: Positive display 0xA7 : reverse display
(10) Display all dots (Display all points)	0	1	0	1	0	0	1	0	0 1	Show all dots: 0xA4 : Conventional 0xA5 : Display all dot matrix
(11) LCD bias ratio setting (LCD bias set)	0	1	0	1	0	0	0	1	0 1	Set the bias ratio: 0XA2 : BIAS=1/9 (commonly used) 0XA3 : BIAS=1/7
(12) Read-modify-write (Read-modify-write)	0	1	1	1	0	0	0	0	0	0XE0 : "Read- Modify -Write" starts. This LCD module uses a serial interface and does not have this function . Please refer to IC data for details
(13) Exit the above "read-modify-Write" command (End)	0	1	1	1	0	1	1	1	0	0XEE : The above "read-modify-write" instruction ends This LCD module uses a serial interface and does not have this function . Please refer to IC data for details
(14) Software reset (Reset)	0	1	1	1	0	0	0	1	0	0XE2 : Software reset.

(15) Line scan order selection (Common output mode select)	1	1	0	0	0 1	0	0	0		Line scan order selection: 0XC0 : Normal scanning sequence: from top to bottom 0XC8 : Reverse scan order: from bottom to top
(16) Power control (Power control set)	0	0	1	0	1	Voltage operation mode selection Choice, a total of 3 digits				Select the internal voltage supply operation mode: D2, D1, D0 bits respectively correspond to whether the internal boost is turned on (1 means to open, 0 means not to open), is the voltage adjustment circuit Open (1 means to open, 0 means not to open), the voltage follower is No open (1 means open, 0 means not open). Usually 0x2C, 0x2E, 0x2F three The instructions are written immediately in sequence, indicating that the internal upgrades are turned on in turn Voltage, voltage adjustment circuit, voltage follower. You can also single

(17) Select internal resistance ratio	0	0	0	1	0	0	Internal voltage value resistance Set up			Write **0x2F to** open three parts of the circuit at a time. Select the internal resistance ratio (Rb/Ra): can be understood as rough Adjust the contrast value. The setting range is : **0x20 ~ 0x27** , The larger the value, the stronger the contrast, the smaller the lighter
(18) Built-in LCD Voltage mode	0	1	0	0	0	0	0	0	1	Set the internal resistance to fine-tune, which can be understood as fine-tuning the contrast Value, these two instructions need to be used immediately. Upper finger Let 0x81 not change, the following command can set the range
Set voltage value		0	0	6-bit voltage value data, 64 levels from 0 to 63						The range is : **0x00 ~ 0x3F, the** larger the value, the stronger the contrast. The smaller the lighter
(19)Static icon display: switch	0	1	0	1	0	1	1	0	0 1	Switch settings for static icons: **0xAC** : Off, **0xAD** : On. This command works when entering and exiting sleep mode
(20) Boost multiple selection (Booster ratio set)	0	1	1	1	1	1	0	0	0	Choose boost multiple: 00: 2 times, 3 times, 4 times 01: 5 times 11: 6 times. The boost multiplier has been set to 4 outside the module Times, no need to use this instruction
		0	0	0	0	0	0	2 digit setting Boost multiple		
(21) Power saving mode (Power save)										Power saving mode, this is not a command, it is displayed by "(10) All dot matrix", (19)Static icon display: on/off, etc. The instructions are combined into a "power saving function". See IC regulations for details "POWER SAVE" section of grid book
(22) Null instruction (NOP)	0	1	1	1	0	0	0	1	1	No operation
(23) Test	0	1	1	1	1	*	*	*	*	For internal testing, don't use it!

Tips: Please refer to the detailed **IC** information " **UC1701X_V1.3.PDF"** on **11 - 16** instruction page

Detailed explanation of tables and instructions.

Page 9

7.3 Correspondence between dot matrix and DD RAM (display data memory) address

Please pay attention to the definition of page: PAGE, which is not the same as the usual "page", means that 8 lines are a "page" . A 128*64 dot matrix screen is divided into 8 "pages", from the 0th "page" to the 7th "page".

The arrangement direction of DB7--DB0: The data is arranged from bottom to top. The lowest bit D0 is at the top and the highest bit D7 is at the bottom Noodles . Each bit of data corresponds to a dot matrix, usually "1" means to turn on the dot matrix, and "0" means to turn off the dot matrix . As shown in the Shown:

The figure below is taken from the UC1701X IC data, and the best results can be obtained from page 29 of "UC1701X_V1.3.PDF".

7.4 Initialization method

The display program compiled by the user must be initialized at the beginning, otherwise the module cannot be displayed normally. Please refer to the p

7.5 Program example:

The interface diagram between LCD module and MPU (taking 8051 series single-chip microcomputer as an example) is as follows:

Figure 9. Serial interface

```
/* LCD demo program JLX12864G-0088, serial interface!
   The driver IC is: UC1701X
   Written by Ye Jianren, November 22, 2011
   JLX Electronics: Website http://www.jlxlcd.cn;
```

```
*/
#include <reg52.H>
#include <intrins.h>
#include <Ctype.h>

sbit key=P3^4;

sbit cs1=P3^1;
sbit rs=P3^0;
sbit reset=P1^0;
sbit sclk=P3^1;
sbit sid=P3^2;


#define uchar unsigned char
#define uint unsigned int
#define ulong unsigned long


uchar code ascii_table_8x16[95][16];
uchar code ascii_table_5x7[95][5];

uchar code cheng1[]={
/*-- Text: Cheng--*/
/*-- Song Ti 23; The corresponding dot matrix under this font is: width x height = 31x31 --*/
/*-- The height is not a multiple of 8, now adjusted to: width x height=32x32 --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0C,
0xFC,0xFC,0x88,0x00,0x00,0x1C,0x78,0xF0,0xE0,0x00,0x80,0x80,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0xFF,0xFF,0x83,0x83,0x83,0x83,0x83,0x83,0xC3,0xC3,0x03,0x1F,
0xFF,0xFF,0x83,0x03,0x03,0x03,0xC3,0xF3,0xF3,0x63,0x03,0x03,0x00,0x00,0x00,0x00,
0x00,0x00,0xFC,0xFF,0x3F,0x00,0x80,0x00,0x00,0x80,0xFF,0xFF,0x03,0x00,0x00,0x03,
0x9F,0xFF,0xF8,0xF8,0xBE,0x1F,0x07,0x01,0x00,0x00,0xE0,0x20,0x00,0x00,0x20,0x38,
0x1F,0x07,0x01,0x00,0x00,0x01,0x01,0x07,0x07,0x23,0x31,0x18,0x0C,0x0E,0x07,0x03,
0x01,0x01,0x01,0x03,0x07,0x0F,0x0E,0x1C,0x1F,0x3F,0x30,0x00,0x00,0x00,0x00,0x00);

uchar code zhuang1[]={
/*-- Text: State--*/
/*-- Song Ti 12; The corresponding dot matrix under this font is: width x height = 16x16 --*/
0x08,0x30,0x00,0xFF,0x20,0x20,0x20,0x20,0xFF,0x20,0xE1,0x26,0x2C,0x20,0x20,0x00,
0x04,0x02,0x01,0xFF,0x40,0x20,0x18,0x07,0x00,0x00,0x03,0x0C,0x30,0x60,0x20,0x00);

uchar code tai1[]={
/*-- Text: State--*/
/*-- Song Ti 12; The corresponding dot matrix under this font is: width x height = 16x16 --*/
0x00,0x04,0x04,0x04,0x84,0x44,0x34,0x4F,0x94,0x24,0x44,0x84,0x84,0x04,0x00,0x00,
0x00,0x60,0x39,0x01,0x00,0x3C,0x40,0x42,0x4C,0x40,0x40,0x70,0x04,0x09,0x31, 0x00);

uchar code shi1[]={
/*-- text: make --*/
/*-- Song Ti 12; The corresponding dot matrix under this font is: width x height = 16x16 --*/
0x40,0x20,0xF0,0x1C,0x07,0xF2,0x94,0x94,0x94,0xFF,0x94,0x94,0x94,0xF4,0x04,0x00,
0x00,0x00,0x7F,0x00,0x40,0x41, 0x22,0x14,0x0C,0x13,0x10,0x30,0x20,0x61,0x20,0x00);

uchar code yong1[]={
/*-- Text: Use --*/
/*-- Song Ti 12; The corresponding dot matrix under this font is: width x height = 16x16 --*/
0x00,0x00,0x00,0xFE,0x22,0x22,0x22,0xFE,0x22,0x22,0x22,0x22,0xFE,0x00,0x00,
0x80,0x40,0x30,0x0F,0x02,0x02,0x02,0x02,0xFF,0x02,0x02,0x42,0x82,0x7F,0x00,0x00);



uchar code mao_hao[]={
/*-- Text:: (colon) --*/
/*-- Song Ti 12; The corresponding dot matrix under this font is: width x height = 8x16 --*/
0x00,0x00,0x00,0xC0,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x30,0x30,0x00,0x00,0x00);

char code num0[]={
/*-- Text: 0 --*/
/*-- Song Ti 12; The corresponding dot matrix under this font is: width x height = 8x16 --*/
0x00,0xE0,0x10,0x08,0x08,0x10,0xE0,0x00,0x00,0x0F,0x10,0x20,0x20,0x10,0x0F,0x00
};
char code num1[]={
/*-- Text: 1 --*/
```



```
/*-- Song Ti 12; The corresponding dot matrix under this font is: width x height = 8x16 --*/
0x00,0x10,0x10,0xF8,0x00,0x00,0x00,0x00,0x00,0x20,0x20,0x3F,0x20,0x20,0x00,0x00
};
char code num2[]={
/*-- Text: 2 --*/
/*-- Song Ti 12; The corresponding dot matrix under this font is: width x height = 8x16 --*/
0x00,0x70,0x08,0x08,0x08,0x88,0x70,0x00,0x00,0x30,0x28,0x24,0x22,0x21,0x30,0x00
};
char code num3[]={
/*-- Text: 3 --*/
/*-- Song Ti 12; The corresponding dot matrix under this font is: width x height = 8x16 --*/
0x00,0x30,0x08,0x88,0x88,0x48,0x30,0x00,0x00,0x18,0x20,0x20,0x20,0x11,0x0E,0x00
};
char code num4[]={
/*-- Text: 4 --*/
/*-- Song Ti 12; The corresponding dot matrix under this font is: width x height = 8x16 --*/
```

```
0x00,0x00,0xC0,0x20,0x10,0xF8,0x00,0x00,0x00,0x00,0x07,0x04,0x24,0x24,0x3F,0x24,0x00
};

char code num5[]={
/*-- Text: 5 --*/
/*-- Song Ti 12; The corresponding dot matrix under this font is: width x height = 8x16 --*/
0x00,0xF8,0x08,0x88,0x88,0x08,0x08,0x00,0x00,0x19,0x21,0x20,0x20,0x11,0x0E,0x00
};

char code num6[]={
/*-- Text: 6 --*/
/*-- Song Ti 12; The corresponding dot matrix under this font is: width x height = 8x16 --*/
0x00,0xE0,0x10,0x88,0x88,0x18,0x00,0x00,0x00,0x0F,0x11,0x20,0x20,0x11,0x0E,0x00
};
char code num7[]={
/*-- Text: 7 --*/
/*-- Song Ti 12; The corresponding dot matrix under this font is: width x height = 8x16 --*/
0x00,0x38,0x08,0x08,0xC8,0x38,0x08,0x00,0x00,0x00,0x00,0x3F,0x00,0x00,0x00,0x00
};
char code num8[]={
/*-- Text: 8 --*/
/*-- Song Ti 12; The corresponding dot matrix under this font is: width x height = 8x16 --*/
0x00,0x70,0x88,0x08,0x08,0x08,0x88,0x70,0x00,0x00,0x00,0x1C,0x22,0x21,0x21,0x22,0x1C,0x00
};

char code num9[]={
/*-- Text: 9 --*/
/*-- Song Ti 12; The corresponding dot matrix under this font is: width x height = 8x16 --*/
0x00,0xE0,0x10,0x08,0x08,0x08,0x10,0xE0,0x00,0x00,0x00,0x00,0x31,0x22,0x22,0x11,0x0F,0x00
};

/*Write command to LCD module*/
void transfer_command(int data1)
{
        char i;
        cs1=0;
        rs=0;
        for(i=0;i<8;i++)
        {
                sclk=0;
                if(data1&0x80) sid=1;
                else sid=0;
                sclk=1;
                data1=data1<<=1;
        }
}

/*Write data to LCD module*/
void transfer_data(int data1)
{
        char i;
        cs1=0;
        rs=1;
        for(i=0;i<8;i++)
        {
                sclk=0;
                if(data1&0x80) sid=1;
```



```
                else sid=0;
                sclk=1;
                data1=data1<<=1;
        }
}

/*Delay*/
void delay(int i)
{
  int j,k;
  for(j=0;j<i;j++)
  for(k=0;k<110;k++);
}

void waitkey()
{
repeat:         if(key==1)
                        goto repeat;
                        else
                        delay(40);
}

/*LCD module initialization*/
void initial_lcd()
{
        cs1=0;
        reset=0;                /*Low level reset*/
        delay(100);
        reset=1;                        /*Reset complete*/
        delay(20);
        transfer_command(0xe2);                 /*Soft reset*/
        delay(5);
        transfer_command(0x2c); /*Boost step poly 1*/
        delay(5);
        transfer_command(0x2e); /*Boost step poly 2*/
        delay(5);
```

```
        transfer_command(0x2f); /*Boost step poly 3*/
        delay(5);
        transfer_command(0x23); /*Coarse adjustment of contrast, adjustable range 0x20 ~ 0x27*/
        transfer_command(0x81); /*Fine-tuning the contrast*/
        transfer_command(0x28); /*0x1a, fine-tuning the value of the contrast, the setting range is 0x00 ~ 0x3f*/
        transfer_command(0xa2); /*1/9 Bias ratio (bias)*/
        transfer_command(0xc8); /*Line scanning order: from top to bottom*/
        transfer_command(0xa0); /*Column scanning order: from left to right*/
        transfer_command(0x40); /*Start line: the first line starts*/
        transfer_command(0xaf); /*Open display*/
        cs1=1;
}


void lcd_address(uchar page,uchar column)
{
        cs1=0;
        column=column-1;                                        //The first column we usually talk about is column 0 in the LCD driver IC. So subtract 1 here.
        page=page-1;
        transfer_command(0xb0+page);                    //Set the page address. Each page is 8 lines. The 64 lines of a screen are divided into 8 pages. What we usually call page 1 is page 0 in the LCD driver IC, so
Subtract 1*/ here
        transfer_command(((column>>4)&0x0f)+0x10);      //Set the upper 4 bits of the column address
        transfer_command(column&0x0f);                          //Set the lower 4 bits of the column address
}

/*Full screen clear screen*/
void clear_screen()
{
        unsigned char i,j;
        cs1=0;
        for(i=0;i<9;i++)
        {
                lcd_address(1+i,1);
                for(j=0;j<132;j++)
                {
                        transfer_data(0x00);
                }
```



```
        }
        cs1=1;
}


//==================display a piture of 128*64 dots==================
void full_display()
{
        int i,j;
        for(i=0;i<8;i++)
        {
                cs1=0;
                lcd_address(i+1,0);
                for(j=0;j<128;j++)
                {
                        transfer_data(0xff);
                }
        }
}


/*Display 32x32 dot matrix images, Chinese characters, rare characters or other icons in 32x32 dot matrix*/
void display_graphic_32x32(uchar page,uchar column,uchar *dp)
{
        uchar i,j;
        cs1=0;
        for(j=0;j<4;j++)
        {
                lcd_address(page+j,column);
                for (i=0;i<31;i++)
                {
                        transfer_data(*dp);                     /*Write data to LCD, the column address will automatically increase by 1 after each 8-bit data is written*/
                        dp++;
                }
        }
        cs1=1;
}

/*Display 16x16 dot matrix images, Chinese characters, rare characters or other icons in 16x16 dot matrix*/
void display_graphic_16x16(uchar page,uchar column,uchar *dp)
{
        uchar i,j;

        cs1=0;
        for(j=0;j<2;j++)
        {
                lcd_address(page+j,column);
                for (i=0;i<16;i++)
                {
                        transfer_data(*dp);                     /*Write data to LCD, the column address will automatically increase by 1 after each 8-bit data is written*/
                        dp++;
                }
        }
        cs1=1;
}
```

```
/*Display 8x16 dot matrix image, ASCII, or 8x16 dot matrix self-made characters, other icons*/
void display_graphic_8x16(uchar page,uchar column,uchar *dp)
{
	uchar i,j;
	cs1=0;
	for(j=0;j<2;j++)
	{
		lcd_address(page+j,column);
		for (i=0;i<8;i++)
		{
			transfer_data(*dp);			/*Write data to LCD, the column address will automatically increase by 1 after each 8-bit data is written*/
			dp++;
		}
	}
	cs1=1;
}
```



```
void display_string_8x16(uint page,uint column,uchar *text)
{
	uint i=0,j,k,n;
	cs1=0;
	while(text[i]>0x00)
	{
		if((text[i]>=0x20)&&(text[i]<=0x7e))
		{
			j=text[i]-0x20;
			for(n=0;n<2;n++)
			{
				lcd_address(page+n,column);
				for(k=0;k<8;k++)
				{
					transfer_data(ascii_table_8x16[j][k+8*n]);/*Display 5x7 ASCII words on the LCD, y is the page address, x is the column address, and finally is the data*/
				}
			}
			i++;
			column+=8;

		}
		else
		i++;
	}
}

void display_string_5x7(uint page,uint column,uchar *text)
{
	uint i=0,j,k;
	cs1=0;
	while(text[i]>0x00)
	{
		if((text[i]>=0x20)&&(text[i]<0x7e))
		{
			j=text[i]-0x20;
			lcd_address(page,column);
			for(k=0;k<5;k++)
			{
				transfer_data(ascii_table_5x7[j][k]);/*Display 5x7 ASCII words on the LCD, y is the page address, x is the column address, and finally is the data*/
			}
			i++;
			column+=6;
		}
		else
		i++;
	}
}

void main(void)
 {

	while(1)
	{
		initial_lcd();
		clear_screen();				//clear all dots
		display_graphic_32x32(1,1,cheng1);	/*On page 1, column 49 shows a single Chinese character "成"*/
		//delay(2000);
		waitkey();
		clear_screen();				//clear all dots
		display_graphic_16x16(5,1,zhuang1);	/*On page 5, the first column shows a single Chinese character "状"*/
		display_graphic_16x16(5,(1+16),tai1);	/*On the 5th page, the 17th column shows a single Chinese character "state"*/
		display_graphic_8x16(5,(1+16*2),mao_hao);	/*On page 5, column 25 shows a single character ":"*/
		display_graphic_16x16(5,(1+16*2+8),shi1);	/*On page 5, column 41 displays a single Chinese character "使"*/
		display_graphic_16x16(5,(1+16*3+8),yong1);	/*On page 5, column 49 shows a single Chinese character "用"*/
		display_graphic_8x16(5,(89),num0);		/*On page 5, column 89 shows a single number "0"*/
		display_graphic_8x16(5,(89+8*1),num0);		/*On page 5, column 97 shows a single number "0"*/
		display_graphic_8x16(5,(89+8*2),mao_hao);	/*On page 5, column 105 displays a single character ":"*/
		display_graphic_8x16(5,(89+8*3),num0);		/*On page 5, column 113 shows a single number "0"*/
		display_graphic_8x16(5,(89+8*4),num0);		/*On page 5, column 121 shows a single number "0"*/
		waitkey();
		//delay(2000);
		clear_screen();				//clear all dots
```

Page 17


```
                display_string_8x16(1,1,"0123456789abcdef");/*Display string on page 1, column 1*/
                display_string_8x16(3,1,"~`!@#$%^&*() _-+=");/*Display string on page *, column **/
                display_string_5x7(5,1,"! #$%&()*+,-./01234");
                display_string_5x7(6,1,"56789:;<=>?@ABCDEFGHI");
                display_string_5x7(7,1,"JKLMNOPQRSTUVWXYZ[\]^");
                display_string_5x7(8,1,"_`abcdefghijklmnopqrs");
                waitkey();

                //delay(2000);
        }
}
```

```
//Longitudinal modulus, suitable for LCD modules with driver ICs such as ST7565P, ST7565R, ST7567, UC1701X, KS0108, etc.

char code ascii_table_8x16[95][16]={

//Bold 8x16 dot matrix ASCII code dot matrix data, national standard read from "JLX-GB2312" type font IC.
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  //--(ie "space") ASCII code: 0X20
0x00,0x00,0x38,0xFC, 0xFC,0x38,0x00,0x00, 0x00,0x00,0x00,0x00,0x0D, 0x0D,0x00,0x00,0x00,0x00,  //-!-  ASCII code: 0X21
0x00,0x0E,0x1E,0x00, 0x00,0x1E,0x0E,0x00, 0x00,0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00,0x00,  //-"-
0x20,0xF8,0xF8,0x20, 0xF8,0xF8,0x20,0x00, 0x02,0x0F,0x0F,0x02, 0x0F,0x0F,0x02,0x00,  //-#-
0x38,0x7C,0x44,0x47, 0x47,0xCC,0x98,0x00, 0x06,0x0C,0x08,0x38, 0x38,0x0F,0x07,0x00,  //-$-
0x30,0x30,0x00,0x80, 0xC0,0x60,0x30,0x00, 0x0C,0x06,0x03,0x01, 0x00,0x0C,0x0C,0x00,  //-%-
0x80,0xD8,0x7C,0xE4, 0xBC,0xD8,0x40,0x00, 0x07,0x0F,0x08,0x08, 0x07,0x0F,0x08,0x00,  //-&-
0x00,0x10,0x1E,0x0E, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00,  //-'-
0x00,0x00,0xF0,0xF8, 0x0C,0x04,0x00,0x00, 0x00,0x00,0x03,0x07, 0x0C,0x08,0x00,0x00,  //-(-
0x00,0x00,0x04,0x0C, 0xF8,0xF0,0x00,0x00, 0x00,0x00,0x08,0x0C, 0x07,0x03,0x00,0x00,  //-)-

0x80,0xA0,0xE0,0xC0, 0xC0,0xE0,0xA0,0x80, 0x00,0x02,0x03,0x01, 0x01,0x03,0x02,0x00,  //-*-  ASCII code: 0X2A
0x00,0x80,0x80,0xE0, 0xE0,0x80,0x80,0x00, 0x00,0x00,0x00,0x03, 0x03,0x00,0x00,0x00,  //-+-
0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x10,0x1E, 0x0E,0x00,0x00,0x00,  //-,-
0x80,0x80,0x80,0x80, 0x80,0x80,0x80,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00,  //---
0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x0C, 0x0C,0x00,0x00,0x00,  //-.-
0x00,0x00,0x00,0x80, 0xC0,0x60,0x30,0x00, 0x0C,0x06,0x03,0x01, 0x00,0x00,0x00,0x00,  //-/-
0xF8, 0xF8, 0x0C, 0xC4, 0x0C, 0xF8, 0xF0, 0x00, 0x03, 0x07, 0x0C, 0x08, 0x0C, 0x07, 0x03, 0x00,  //-0-  ASCII code: 0X30
0x00,0x10,0x18,0xFC, 0xFC,0x00,0x00,0x00, 0x00,0x08,0x08,0x0F, 0x0F,0x08,0x08,0x00,  //-1-
0x08,0x0C,0x84,0xC4, 0x64,0x3C,0x18,0x00, 0x0E,0x0F,0x09,0x08, 0x08,0x0C,0x0C,0x00,  //-2-
0x08,0x0C,0x44,0x44, 0x44,0xFC,0xB8,0x00, 0x04,0x0C,0x08,0x08, 0x08,0x0F,0x07,0x00,  //-3-

0xC0,0xE0,0xB0,0x98, 0xFC,0xFC,0x80,0x00, 0x00,0x00,0x00,0x08, 0x0F,0x0F,0x08,0x00,  //-4-  ASCII code: 0X34
0x7C,0x7C,0x44,0x44, 0x44,0xC4,0x84,0x00, 0x04,0x0C,0x08,0x08, 0x08,0x0F,0x07,0x00,  //-5-
0xF0,0xF8,0x4C,0x44, 0x44,0xC0,0x80,0x00, 0x07,0x0F,0x08,0x08, 0x08,0x0F,0x07,0x00,  //-6-
0x0C, 0x0C, 0x04, 0x84, 0xC4, 0x7C, 0x3C, 0x00, 0x00, 0x00, 0x0F, 0x0F, 0x00, 0x00, 0x00, 0x00,  //-7-
0xB8,0xFC,0x44,0x44, 0x44,0xFC,0xB8,0x00, 0x07,0x0F,0x08,0x08, 0x08,0x0F,0x07,0x00,  //-8-
0x38,0x7C,0x44,0x44, 0x44,0xFC,0xF8,0x00, 0x00,0x08,0x08,0x08, 0x0C,0x07,0x03,0x00,  //-9-
0x00,0x00,0x00,0x30, 0x30,0x00,0x00,0x00, 0x00,0x00,0x00,0x06, 0x06,0x00,0x00,0x00,  //-:-
0x00,0x00,0x00,0x30, 0x30,0x00,0x00,0x00, 0x00,0x00,0x08,0x0E, 0x06,0x00,0x00,0x00,  //-;-
0x00,0x80,0xC0,0x60, 0x30,0x18,0x08,0x00, 0x00,0x00,0x01,0x03, 0x06,0x0C,0x08,0x00,  //-<-
```


Page 18


```
0x00,0x20,0x20,0x20, 0x20,0x20,0x20,0x20,0x00, 0x00,0x01,0x01,0x01, 0x01,0x01,0x01,0x01,0x00,  //-=-

0x00,0x08,0x18,0x30, 0x60,0xC0,0x80,0x00, 0x00,0x08,0x0C,0x06, 0x03,0x01,0x00,0x00,  //->-  ASCII code: 0X3E
0x18,0x1C,0x04,0xC4, 0xE4,0x3C,0x18,0x00, 0x00,0x00,0x00,0x0D, 0x0D,0x00,0x00,0x00,  //-?-
0xF0,0xF0,0x08,0xC8, 0xC8,0xF8,0xF0,0x00, 0x07,0x0F,0x08,0x0B, 0x0B,0x0B,0x01,0x00,  //-@-
0xE0,0xF0,0x98,0x8C, 0x98,0xF0,0xE0,0x00, 0x0F,0x0F,0x00,0x00, 0x00,0x0F,0x0F,0x00,  //-A-  ASCII code: 0X41
0x04,0xFC,0xFC,0x44, 0x44,0xFC,0xB8,0x00, 0x08,0x0F,0x0F,0x08, 0x08,0x0F,0x07,0x00,  //-B-
0xF0,0xF8,0x0C,0x04, 0x04,0x0C,0x18,0x00, 0x03,0x07,0x0C,0x08, 0x08,0x0C,0x06,0x00,  //-C-
```

```
0x04,0xFC,0xFC,0x04, 0x0C,0xF8,0xF0,0x00, 0x08,0x0F,0x0F,0x08, 0x0C,0x07,0x03,0x00,   //-D-
0x04,0xFC,0xFC,0x44, 0xE4,0x0C,0x1C,0x00, 0x08,0x0F,0x0F,0x08, 0x08,0x0C,0x0E,0x00,   //-E-
0x04,0xFC,0xFC,0x44, 0xE4,0x0C,0x1C,0x00, 0x08,0x0F,0x0F,0x08, 0x00,0x00,0x00,0x00,   //-F-
0xF0,0xF8,0x0C,0x84, 0x84,0x8C,0x98,0x00, 0x03,0x07,0x0C,0x08, 0x08,0x07,0x0F,0x00,   //-G-

0xFC, 0xFC, 0x40, 0x40, 0x40, 0xFC, 0xFC, 0x00, 0x0F, 0x0F, 0x00, 0x00, 0x00, 0x0F, 0x0F, 0x00,   //-H-   ASCII code: 0X48
0x00,0x00,0x04,0xFC, 0xFC,0x04,0x00,0x00, 0x00,0x00,0x08,0x0F, 0x0F,0x08,0x00,0x00,   //-I-
0x00,0x00,0x00,0x04, 0xFC,0xFC,0x04,0x00, 0x07,0x0F,0x08,0x08, 0x0F,0x07,0x00,0x00,   //-J-
0x04,0xFC,0xFC,0xC0, 0xE0,0x3C,0x1C,0x00, 0x08,0x0F,0x0F,0x00, 0x01,0x0F,0x0E,0x00,   //-K-
0x04,0xFC,0xFC,0x04, 0x00,0x00,0x00,0x00, 0x08,0x0F,0x0F,0x08, 0x08,0x0C,0x0E,0x00,   //-L-
0xFC, 0xFC, 0x38, 0x70, 0x38, 0xFC, 0xFC, 0x00, 0x0F, 0x0F, 0x00, 0x00, 0x00, 0x0F, 0x0F, 0x00,   //-M-
0xFC, 0xFC, 0x38, 0x70, 0xE0, 0xFC, 0xFC, 0x00, 0x0F, 0x0F, 0x00, 0x00, 0x00, 0x0F, 0x0F, 0x00,   //-N-
0xF8,0xFC,0x04,0x04, 0x04,0xFC,0xF8,0x00, 0x07,0x0F,0x08,0x08, 0x08,0x0F,0x07,0x00,   //-O-
0x04,0xFC,0xFC,0x44, 0x44,0x7C,0x38,0x00, 0x08,0x0F,0x0F,0x08, 0x00,0x00,0x00,0x00,   //-P-
0xF8,0xFC,0x04,0x04, 0x04,0xFC,0xF8,0x00, 0x07,0x0F,0x08,0x0E, 0x3C,0x3F,0x27,0x00,   //-Q-

0x04,0xFC,0xFC,0x44, 0xC4,0xFC,0x38,0x00, 0x08,0x0F,0x0F,0x00, 0x00,0x0F,0x0F,0x00,   //-R-
0x18,0x3C,0x64,0x44, 0xC4,0x9C,0x18,0x00, 0x06,0x0E,0x08,0x08, 0x08,0x0F,0x07,0x00,   //-S-
0x00, 0x1C, 0x0C, 0xFC, 0xFC, 0x0C, 0x1C, 0x00, 0x00, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x00, 0x00,   //-T-
0xFC,0xFC,0x00,0x00, 0x00,0xFC,0xFC,0x00, 0x07,0x0F,0x08,0x08, 0x08,0x0F,0x07,0x00,   //-U-
0xFC,0xFC,0x00,0x00, 0x00,0xFC,0xFC,0x00, 0x01,0x03,0x06,0x0C, 0x06,0x03,0x01,0x00,   //-V-
0xFC, 0xFC, 0x00, 0x00, 0x00, 0xFC, 0xFC, 0x00, 0x07, 0x0F, 0x0E, 0x03, 0x0E, 0x0F, 0x07, 0x00,   //-W-
0x0C, 0x3C, 0xF0, 0xE0, 0xF0, 0x3C, 0x0C, 0x00, 0x0C, 0x0F, 0x03, 0x01, 0x03, 0x0F, 0x0C, 0x00,   //-X-
0x00, 0x0C, 0x7C, 0xC0, 0xC0, 0x7C, 0x3C, 0x00, 0x00, 0x00, 0x08, 0x0F, 0x0F, 0x08, 0x00, 0x00,   //-Y-
0x1C, 0x0C, 0x84, 0xC4, 0x64, 0x3C, 0x1C, 0x00, 0x0E, 0x0F, 0x09, 0x08, 0x08, 0x0C, 0x0E, 0x00,   //-Z-
0x00,0x00,0xFC,0xFC, 0x04,0x04,0x00,0x00, 0x00,0x00,0x0F,0x0F, 0x08,0x08,0x00,0x00,   //-[-

0x38,0x70,0xE0,0xC0, 0x80,0x00,0x00,0x00, 0x00,0x00,0x00,0x01, 0x03,0x07,0x0E,0x00,   //-\-
0x00,0x00,0x04,0x04, 0xFC,0xFC,0x00,0x00, 0x00,0x00,0x08,0x08, 0x0F,0x0F,0x00,0x00,   //-]-
0x08,0x0C,0x06,0x03, 0x06,0x0C,0x08,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00,   //-^-
0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x20,0x20,0x20,0x20, 0x20,0x20,0x20,0x20,   //-_-
0x00,0x00,0x03,0x07, 0x04,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00,   //-`-
0x00,0xA0,0xA0,0xA0, 0xE0,0xC0,0x00,0x00, 0x07,0x0F,0x08,0x08, 0x07,0x0F,0x08,0x00,   //-a-   ASCII code: 0X61
0x04,0xFC,0xFC,0x20, 0x60,0xC0,0x80,0x00, 0x00,0x0F,0x0F,0x08, 0x08,0x0F,0x07,0x00,   //-b-
0xC0,0xE0,0x20,0x20, 0x20,0x60,0x40,0x00, 0x07,0x0F,0x08,0x08, 0x08,0x0C,0x04,0x00,   //-c-
0x80,0xC0,0x60,0x24, 0xFC,0xFC,0x00,0x00, 0x07,0x0F,0x08,0x08, 0x07,0x0F,0x08,0x00,   //-d-
0xC0,0xE0,0xA0,0xA0, 0xA0,0xE0,0xC0,0x00, 0x07,0x0F,0x08,0x08, 0x08,0x0C,0x04,0x00,   //-e-

0x40,0xF8,0xFC,0x44, 0x0C,0x18,0x00,0x00, 0x08,0x0F,0x0F,0x08, 0x00,0x00,0x00,0x00,   //-f-
```



```
0xC0,0xE0,0x20,0x20, 0xC0,0xE0,0x20,0x00, 0x27,0x6F,0x48,0x48, 0x7F,0x3F,0x00,0x00,   //-g-
0x04,0xFC,0xFC,0x40, 0x20,0xE0,0xC0,0x00, 0x08,0x0F,0x0F,0x00, 0x00,0x0F,0x0F,0x00,   //-h-
0x00,0x00,0x20,0xEC, 0xEC,0x00,0x00,0x00, 0x00,0x00,0x08,0x0F, 0x0F,0x08,0x00,0x00,   //-i-
0x00,0x00,0x00,0x00, 0x20,0xEC,0xEC,0x00, 0x00,0x30,0x70,0x40, 0x40,0x7F,0x3F,0x00,   //-j-
0x04,0xFC,0xFC,0x80, 0xC0,0x60,0x20,0x00, 0x08,0x0F,0x0F,0x01, 0x03,0x0E,0x0C,0x00,   //-k-
0x00,0x00,0x04,0xFC, 0xFC,0x00,0x00,0x00, 0x00,0x00,0x08,0x0F, 0x0F,0x08,0x00,0x00,   //-l-
0xE0, 0xE0, 0x60, 0xC0, 0x60, 0xE0, 0xC0, 0x00, 0x0F, 0x0F, 0x00, 0x07, 0x00, 0x0F, 0x0F, 0x00,   //-m-
0x20,0xE0,0xC0,0x20, 0x20,0xE0,0xC0,0x00, 0x00,0x0F,0x0F,0x00, 0x00,0x0F,0x0F,0x00,   //-n-
0xC0,0xE0,0x20,0x20, 0x20,0xE0,0xC0,0x00, 0x07,0x0F,0x08,0x08, 0x08,0x0F,0x07,0x00,   //-o-

0x20,0xE0,0xC0,0x20, 0x20,0xE0,0xC0,0x00, 0x40,0x7F,0x7F,0x48, 0x08,0x0F,0x07,0x00,   //-p-
0xC0,0xE0,0x20,0x20, 0xC0,0xE0,0x20,0x00, 0x07,0x0F,0x08,0x48, 0x7F,0x7F,0x40,0x00,   //-q-
0x20,0xE0,0xC0,0x60, 0x20,0xE0,0xC0,0x00, 0x08,0x0F,0x0F,0x08, 0x00,0x00,0x00,0x00,   //-r-
0x40,0xE0,0xA0,0x20, 0x20,0x60,0x40,0x00, 0x04,0x0C,0x09,0x09, 0x0B,0x0E,0x04,0x00,   //-s-
0x20,0x20,0xF8,0xFC, 0x20,0x20,0x00,0x00, 0x00,0x00,0x07,0x0F, 0x08,0x0C,0x04,0x00,   //-t-
0xE0,0xE0,0x00,0x00, 0xE0,0xE0,0x00,0x00, 0x07,0x0F,0x08,0x08, 0x07,0x0F,0x08,0x00,   //-u-
0x00,0xE0,0xE0,0x00, 0x00,0xE0,0xE0,0x00, 0x00,0x03,0x07,0x0C, 0x0C,0x07,0x03,0x00,   //-v-
0xE0, 0xE0, 0x00, 0x80, 0x00, 0xE0, 0xE0, 0x00, 0x07, 0x0F, 0x0C, 0x07, 0x0C, 0x0F, 0x07, 0x00,   //-w-
0x20,0x60,0xC0,0x80, 0xC0,0x60,0x20,0x00, 0x08,0x0C,0x07,0x03, 0x07,0x0C,0x08,0x00,   //-x-
0xE0,0xE0,0x00,0x00, 0x00,0xE0,0xE0,0x00, 0x47,0x4F,0x48,0x48, 0x68,0x3F,0x1F,0x00,   //-y-
```

```
0x60,0x60,0x20,0xA0, 0xE0,0x60,0x20,0x00, 0x0C,0x0E,0x0B,0x09, 0x08,0x0C,0x0C,0x00,   //-z-   //
0x00,0x40,0x40,0xF8, 0xBC,0x04,0x04,0x00, 0x00,0x00,0x00,0x07, 0x0F,0x08,0x08,0x00,   //-{-
0x00,0x00,0x00,0xBC, 0xBC,0x00,0x00,0x00, 0x00,0x00,0x00,0x0F, 0x0F,0x00,0x00,0x00,   //-|-
0x00,0x04,0x04,0xBC, 0xF8,0x40,0x40,0x00, 0x00,0x08,0x08,0x0F, 0x07,0x00,0x00,0x00,   //-}-
0x08,0x0C,0x04,0x0C, 0x08,0x0C,0x04,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00,   //-~-   ASCII code: 0X7E

};



char code ascii_table_5x8[95][5]={
/*All ASCII list: 5x8 dot matrix*/
0x00,0x00,0x00,0x00,0x00,//--//space
0x00,0x00,0x4f,0x00,0x00,//-!-
0x00,0x07,0x00,0x07,0x00,//-"-
0x14,0x7f,0x14,0x7f,0x14,//-#-
0x24,0x2a,0x7f,0x2a,0x12,//-$-
0x23,0x13,0x08,0x64,0x62,//-%-
0x36,0x49,0x55,0x22,0x50,//-&-
0x00,0x05,0x07,0x00,0x00,//-'-
0x00,0x1c,0x22,0x41, 0x00,//-(-
0x00,0x41, 0x22,0x1c,0x00,//-)-
0x14,0x08,0x3e,0x08,0x14,//-*-
0x08,0x08,0x3e,0x08,0x08,//-+-
0x00,0x50,0x30,0x00,0x00,//-,-
0x08,0x08,0x08,0x08,0x08,//---
```



```
0x00,0x60,0x60,0x00,0x00,//-.-
0x20,0x10,0x08,0x04,0x02,//-/-
0x3e,0x51,0x49,0x45,0x3e,//-0-
0x00,0x42,0x7f,0x40,0x00,//-1-
0x42,0x61,0x51,0x49,0x46,//-2-
0x21,0x41,0x45,0x4b,0x31,//-3-
0x18,0x14,0x12,0x7f,0x10,//-4-
0x27,0x45,0x45,0x45,0x39,//-5-
0x3c,0x4a,0x49,0x49,0x30,//-6-
0x01,0x71,0x09,0x05,0x03,//-7-
0x36,0x49,0x49,0x49,0x36,//-8-
0x06,0x49,0x49,0x29,0x1e,//-9-
0x00,0x36,0x36,0x00,0x00,//-:-
0x00,0x56,0x36,0x00,0x00,//-;-
0x08,0x14,0x22,0x41, 0x00,//-<-
0x14,0x14,0x14,0x14,0x14,//-=-
0x00,0x41, 0x22,0x14,0x08,//->-
0x02,0x01,0x51,0x09,0x06,//-?-
0x32,0x49,0x79,0x41, 0x3e,//-@-
0x7e,0x11,0x11,0x11,0x7e,//-A-
0x7f,0x49,0x49,0x49,0x36,//-B-
0x3e,0x41, 0x41, 0x41, 0x22,//-C-
0x7f,0x41, 0x41, 0x22,0x1c,//-D-
0x7f,0x49,0x49,0x49,0x41,//-E-
0x7f,0x09,0x09,0x09,0x01,//-F-
0x3e,0x41,0x49,0x49,0x7a,//-G-
0x7f,0x08,0x08,0x08,0x7f,//-H-
0x00,0x41, 0x7f,0x41, 0x00,//-I-
0x20,0x40,0x41, 0x3f,0x01,//-J-
0x7f,0x08,0x14,0x22,0x41,//-K-
0x7f,0x40,0x40,0x40,0x40,//-L-
0x7f,0x02,0x0c,0x02,0x7f,//-M-
0x7f,0x04,0x08,0x10,0x7f,//-N-
0x3e,0x41, 0x41, 0x41, 0x3e,//-O-
```

```
0x7f,0x09,0x09,0x09,0x06,//-P-
0x3e,0x41,0x51,0x21,0x5e,//-Q-
0x7f,0x09,0x19,0x29,0x46,//-R-
0x46,0x49,0x49,0x49,0x31,//-S-
0x01,0x01,0x7f,0x01,0x01,//-T-
0x3f,0x40,0x40,0x40,0x3f,//-U-
0x1f,0x20,0x40,0x20,0x1f,//-V-
0x3f,0x40,0x38,0x40,0x3f,//-W-
0x63,0x14,0x08,0x14,0x63,//-X-
0x07,0x08,0x70,0x08,0x07,//-Y-
0x61,0x51,0x49,0x45,0x43,//-Z-
0x00,0x7f,0x41, 0x41, 0x00,//-[-
0x02,0x04,0x08,0x10,0x20,//-\-
```


Page 21


```
0x00,0x41, 0x41, 0x7f,0x00,//-]-
0x04,0x02,0x01,0x02,0x04,//-^-
0x40,0x40,0x40,0x40,0x40,//-_-
0x01,0x02,0x04,0x00,0x00,//-`-
0x20,0x54,0x54,0x54,0x78,//-a-
0x7f,0x48,0x48,0x48,0x30,//-b-
0x38,0x44,0x44,0x44,0x44,//-c-
0x30,0x48,0x48,0x48,0x7f,//-d-
0x38,0x54,0x54,0x54,0x58,//-e-
0x00,0x08,0x7e,0x09,0x02,//-f-
0x48,0x54,0x54,0x54,0x3c,//-g-
0x7f,0x08,0x08,0x08,0x70,//-h-
0x00,0x00,0x7a,0x00,0x00,//-i-
0x20,0x40,0x40,0x3d,0x00,//-j-
0x7f,0x20,0x28,0x44,0x00,//-k-
0x00,0x41,0x7f,0x40,0x00,//-l-
0x7c,0x04,0x38,0x04,0x7c,//-m-
0x7c,0x08,0x04,0x04,0x78,//-n-
0x38,0x44,0x44,0x44,0x38,//-o-
0x7c,0x14,0x14,0x14,0x08,//-p-
0x08,0x14,0x14,0x14,0x7c,//-q-
0x7c,0x08,0x04,0x04,0x08,//-r-
0x48,0x54,0x54,0x54,0x24,//-s-
0x04,0x04,0x3f,0x44,0x24,//-t-
0x3c,0x40,0x40,0x40,0x3c,//-u-
0x1c,0x20,0x40,0x20,0x1c,//-v-
0x3c,0x40,0x30,0x40,0x3c,//-w-
0x44,0x28,0x10,0x28,0x44,//-x-
0x04,0x48,0x30,0x08,0x04,//-y-
0x44,0x64,0x54,0x4c,0x44,//-z-
0x08,0x36,0x41, 0x41, 0x00,//-{-
0x00,0x00,0x77,0x00,0x00,//-|-
0x00,0x41, 0x41, 0x36,0x08,//-}-
0x04,0x02,0x02,0x02,0x01,//-~-
};
```