

JLX12864G-135-PN 使用说明书

(不带字库 IC)

目 录

序号	内 容 标 题	页码
1	概述	2
2	特点	2
3	外形及接口引脚功能	3~4
4	基本原理	4~5
5	技术参数	5~6
6	时序特性	6~8
7	指令功能及硬件接口与编程案例	8~末页

1. 概述

晶联讯电子专注于液晶屏及液晶模块的研发、制造。所生产 JLX12864G-135 型液晶模块由于使用方便、显示清晰，广泛应用于各种人机交流面板。

JLX12864G-135 可以显示 128 列*64 行点阵单色图片，或显示 8 个/行*4 行 16*16 点阵的汉字，或显示 16 个/行*8 行 8*8 点阵的英文、数字、符号。

本产品可选择带中文字库 IC 与不带中文字库 IC 两种。

2. JLX12864G-135 图像型点阵液晶模块的特性

2.1 结构牢：背光带有挡墙，焊接式 FPC。

2.2 IC 采用矽创公司 UC1701X, 功能强大，稳定性好

2.3 功耗低:1~100mW（关掉背光：[0.3mA@3.3V](#), 打开背光不大于 100mW）；

2.4 显示内容：

(1) 128*64 点阵单色图片，或其它小于 128*64 点阵的单色图片；

(2) 可选用 16*16 点阵或其他点阵的图片来自编汉字，按照 16*16 点阵汉字来计算可显示 8 字*4 行；

(3) 按照 12*12 点阵汉字来计算可显示 10 字*4 行；

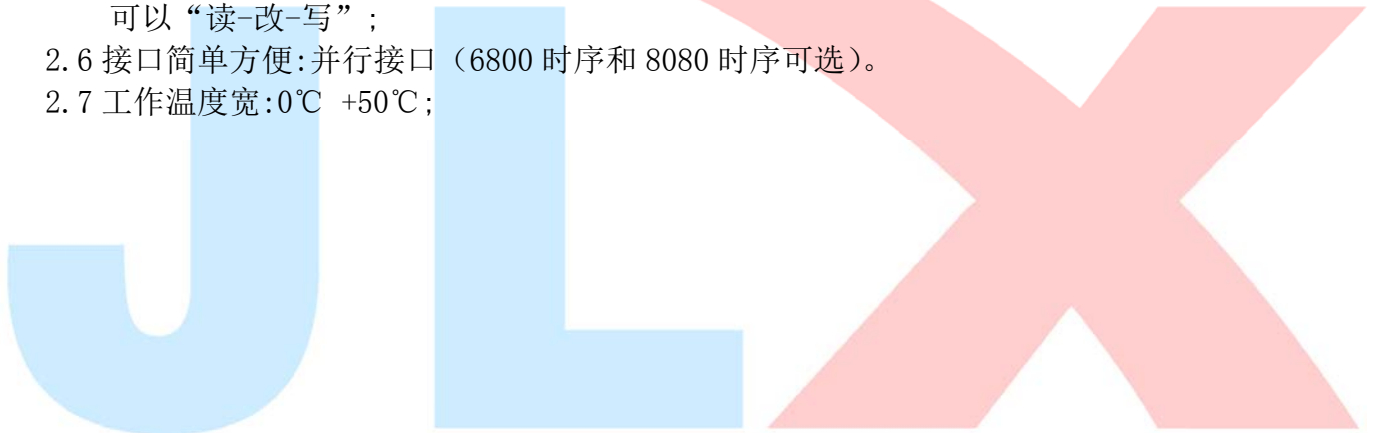
(4) 按照 8*16 点阵汉字来计算可显示 16 字*4 行；

(5) 按照 5*8 点阵汉字来计算可显示 21 字*8 行；

2.5 指令功能强:可软件调对比度、正显/反显转换、行列扫描方向可改（可旋转 180 度使用），可以“读-改-写”；

2.6 接口简单方便:并行接口（6800 时序和 8080 时序可选）。

2.7 工作温度宽:0℃ +50℃；



3. 外形尺寸及接口引脚功能

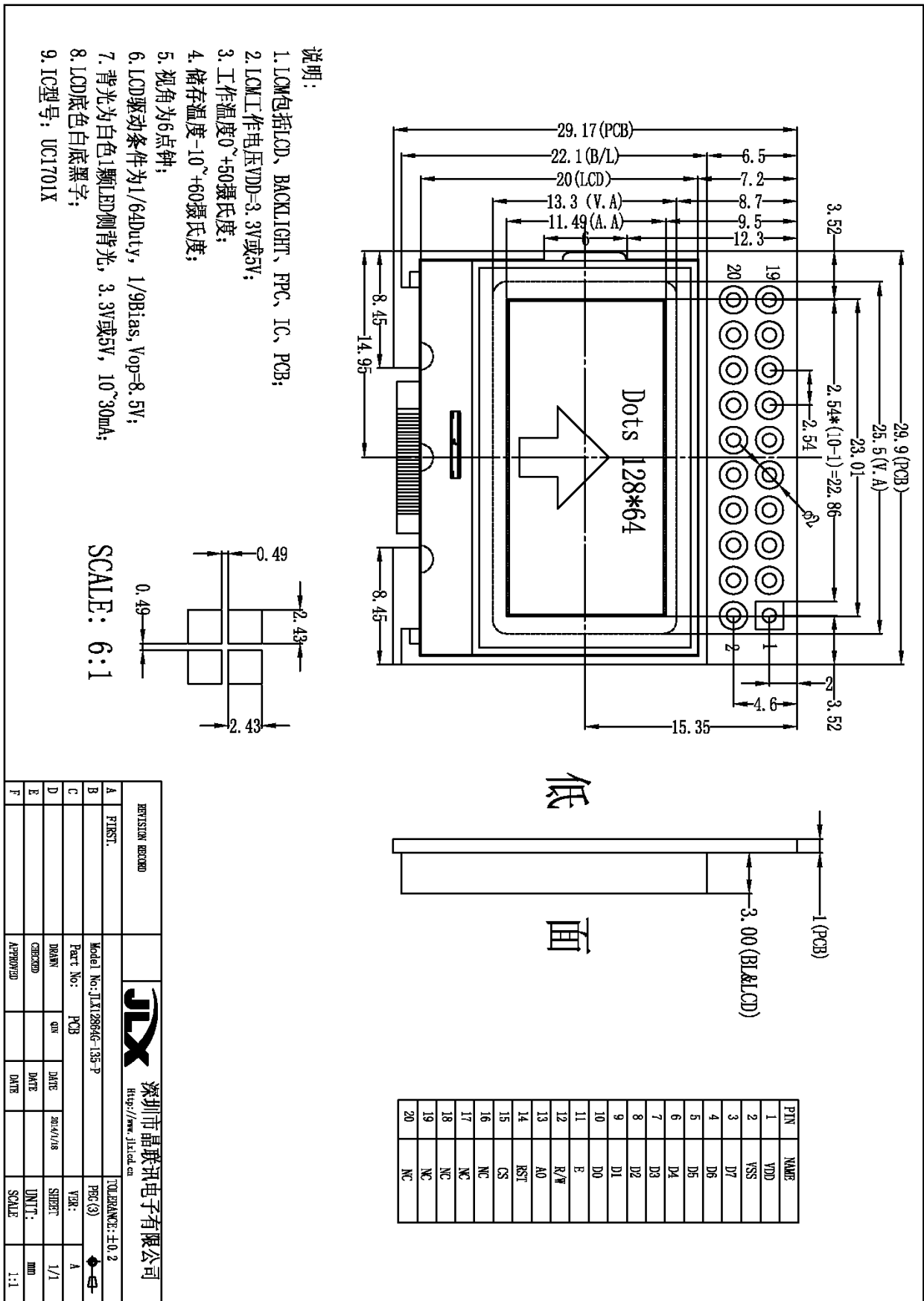


图 1. 外形尺寸

模块的接口引脚功能

表 1: 模块的接口引脚功能

引线号	符号	名称	功能
1	VDD	电路电源	5V, 或 3.3V 可选
2	VSS	接地	0V
3-10	D7-D0	I/O	数据总线 DB7-DB0
11	E(/RD)	6800 时序: 使能 8080 时序: 读	并行接口时并且选择 6800 时序时: 使能信号, 高电平有效. 并行接口时并且选择 8080 时序时: 读数据, 低电平有效.
12	R/W	6800 时序: 读/写 8080 时序: 写	并行接口时并且选择 6800 时序时: H: 读数据 L: 写数据 并行接口时并且选择 8080 时序时: 写数据, 低电平有效.
13	A0(RS)	寄存器选择信号	H: 数据寄存器 0: 指令寄存器
14	RST	复位	低电平复位, 复位完成后, 回到高电平, 液晶模块开始工作
15	CS	片选	低电平片选
16	LEDA	背光电源	背光电源正极, 同 VDD 电压 (5V 或 3.3V)
17	NC	空脚	空脚
18	NC	空脚	空脚
19	NC	空脚	空脚
20	NC	空脚	空脚

4. 基本原理

4.1 液晶屏 (LCD)

在 LCD 上排列着 128×64 点阵, 128 个列信号与驱动 IC 相连, 64 个行信号也与驱动 IC 相连, IC 邦定在 LCD 玻璃上 (这种加工工艺叫 COG)。

4.2 工作电图:

图 2 是 JLX12864G-135 图像点阵型模块的电路框图, 它由驱动 IC UC1701X 及几个电阻电容组成。

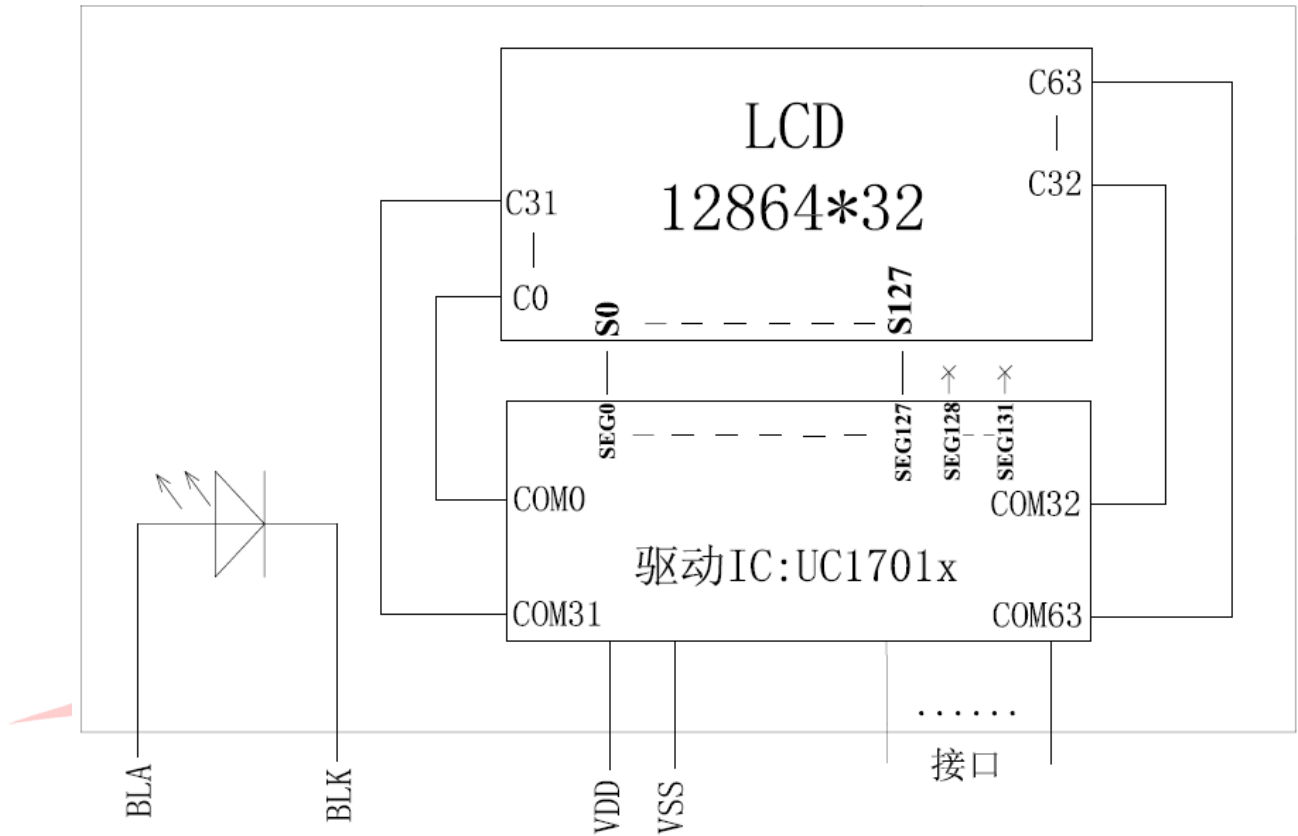


图 2: JLX12864G-135 图像点阵型液晶模块的电路框图

4.3 背光参数

该型号液晶模块带 LED 背光源。它的性能参数如下：
 正常工作电流为：10~15mA (LED 灯数共 1 颗)；
 工作电压：3.0V；

5. 技术参数

5.1 最大极限参数 (超过极限参数则会损坏液晶模块)

名称	符号	标准值			单位
		最小	典型	最大	
电路电源	VDD - VSS	-0.3		3.6	V
LCD 驱动电压	V0、VOUT	-0.3		13.5	V
静电电压		-	-	100	V
工作温度		0		+50	°C
储存温度		-10		+60	°C

表 2: 最大极限参数

5.2 直流 (DC) 参数

名称	符号	测试条件	标准值			单位
			MIN	TYPE	MAX	
工作电压	VDD		2.4	3.3	3.6	V
背光工作电压	VLED		2.9	3.0	3.1	V
输入高电平	VIH	-	2.2		VDD	V

输入低电平	V _{I0}	-	-0.3		0.6	V
输出高电平	V _{OH}	I _{OH} = 0.2mA	2.4		-	V
输出低电平	V _{O0}	I _{O0} = 1.2mA	-		0.4	V
模块工作电流	I _{DD}	V _{DD} = 3.0V	-		1.0	mA
背光工作电流	I _{LED}	V _{LED} =3.0V (共 1 颗灯)	10	15	20	mA

表 3: 直流 (DC) 参数

6. 读写时序特性

6.1 并行接口:

从 CPU 写到 UC1701X (Writing Data from CPU to UC1701X)

System Bus Timing for 6800 Series MPU

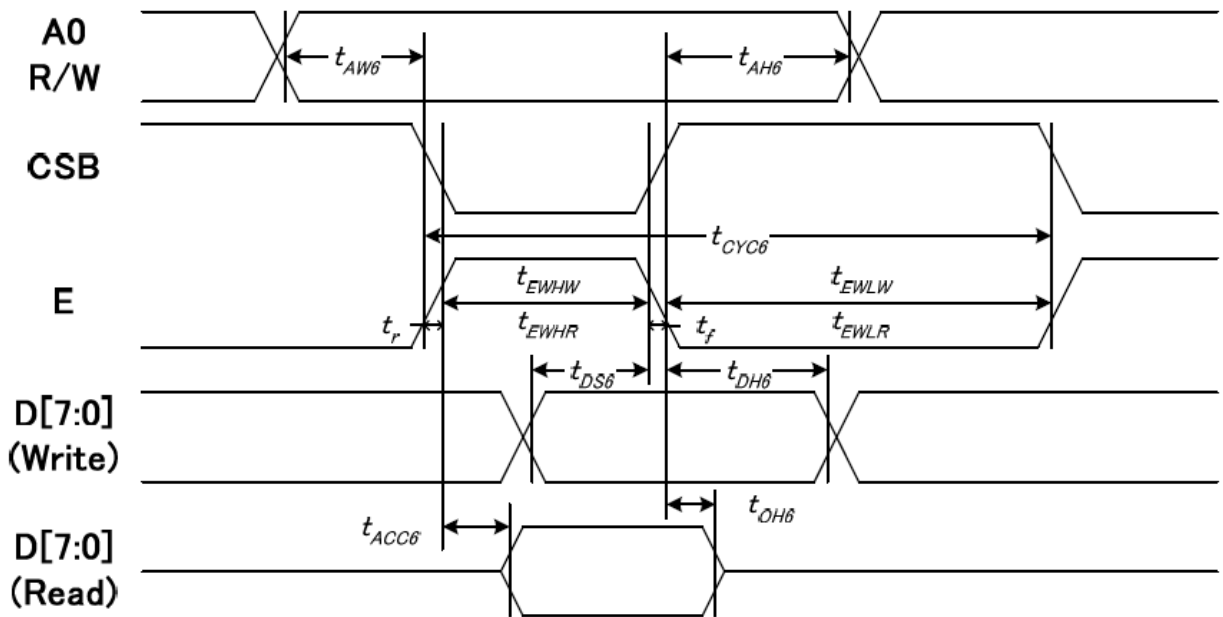


图 5. 从 CPU 写到 UC1701X (Writing Data from CPU to UC1701X)

System Bus Timing for 8080 Series MPU

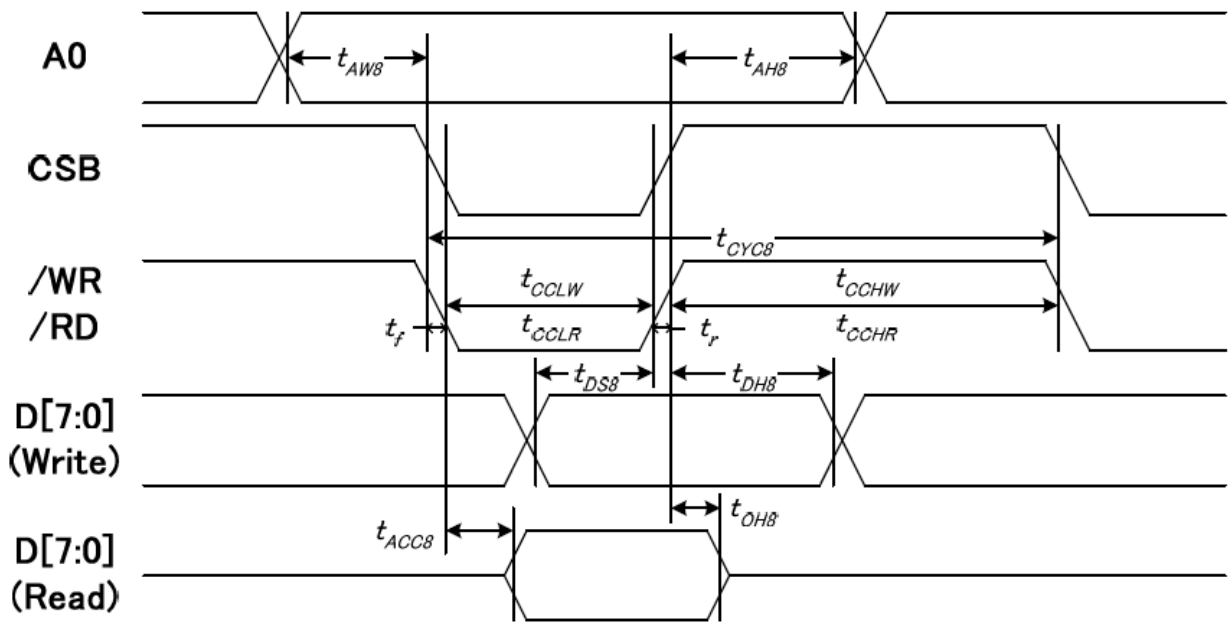


图 6. 从 CPU 写到 UC1701X (Writing Data from CPU to UC1701X)

6.4 并行接口：时序要求 (AC 参数):

写数据到 UC1701X 的时序要求: (8080 系列 MPU)

项目	符号	测试条件	极限值			单位
			MIN	TYPE	MAX	
地址保持时间	A0	tAHS8	0	--	--	ns
地址建立时间		tASS8	0		--	ns
系统循环时间		tCYC8	240		--	ns
使能“低”脉冲(写)	WR	tCCLW	80	--	--	ns
使能“高”脉冲(写)		tCCHW	80	--	--	ns
使能“低”脉冲(读)	RD	tCCLR	140	--	--	ns
使能“高”脉冲(读)		tCCHR	80	--		ns
写数据建立时间	D0-D7	tDS8	40		--	ns
写数据保持时间		tDH8	0		--	
读时间		tACC8	--		70	
读输出允许时间		tOH8	5		50	ns

写数据到 UC1701X 的时序要求: (6800 系列 MPU)

项目	符号	测试条件	极限值			单位
			MIN	TYPE	MAX	
地址保持时间	A0	tAHS6	0	--	--	ns
地址建立时间		tASS6	0		--	ns
系统循环时间		tCYC6	240		--	ns
使能“低”脉冲(写)	WR	tEWLW	80	--	--	ns
使能“高”脉冲(写)		tEWHW	80	--	--	ns
使能“低”脉冲(读)	RD	tEWLR	80	--	--	ns

使能“高”脉冲(读)		tEWHR	140	--		ns
写数据建立时间	D0-D7	tDS6	40		--	ns
写数据保持时间		tDH6	0		--	
读时间		tACC6	--		70	
读输出允许时间		tOH6	5		50	ns

6.5 电源启动后复位的时序要求 (RESET CONDITION AFTER POWER UP):

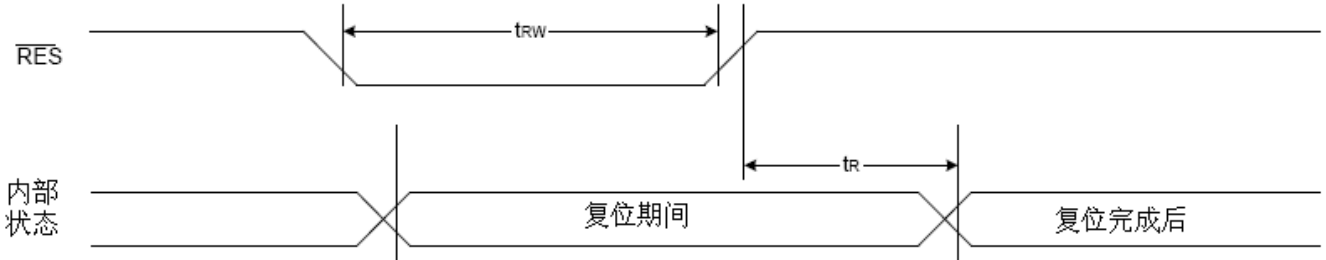


图 7: 电源启动后复位的时序

表 6: 电源启动后复位的时序要求

项目	符号	测试条件	极限值			单位
			MIN	TYPE	MAX	
复位时间	tr		--	--	1.0	us
复位保持低电平的时间	trw	引脚: RES	1.0	--	--	us

7. 指令功能:
7.1 指令表

指令表

表 8.

指令名称	指令码									说明
	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
(1) 显示开/关 (display on/off)	0	1	0	1	0	1	1	1	0 1	显示开/关: 0XAE :关, 0XAF : 开
(2) 显示初始行设置 (Display start line set)	0	0	1	显示初始行地址, 共 6 位						设置显示存储器的显示初始行,可设置值为 0X40~0X7F ,分别代表第 0~63 行, 针对该液晶屏一般设置为 0x60
(3) 页地址设置 (Page address set)	0	1	0	1	1	显示页地址, 共 4 位				设置页地址。每 8 行为一个页, 64 行分为 8 个页, 可设置值为: 0XB0~0XB8 分别对应第一页到第九页, 第九页是一个单独的一行图标, 本液晶屏没有这一行图标, 所以设置值为 0XB0~0XB7 分别对应第一页~第八页。
(4) 列地址高4位设置	0	0	0	0	1	列地址的高 4 位				高 4 位与低 4 位共同组成列地址, 指定 128

列地址低4位设置		0	0	0	0	列地址的低 4 位				列中的其中一列。比如液晶模块的第 100 列地址十六进制为 0x64, 那么此指令由 2 个字节来表达: 0x16, 0x04
(5) 读状态 (Status read)	0	状态				0	0	0	0	并口时: 读驱动 IC 的当前状态,串口时不能用此指令
(6)写显示数据到液晶屏 (Display data write)	1	8 位显示数据								从 CPU 写数据到液晶屏, 每一位对应一个点阵, 1 个字节对应 8 个竖置的点阵
(7)读液晶屏的显示数据 (Display data read)	1	8 位显示数据								并口时: 读已经显示到液晶屏上的点阵数据。串口时不能用此指令
(8) 显示列地址增减 (ADC select)		1	0	1	0	0	0	0	0	显示列地址增减: 1 0xA0: 常规: 列地址从左到右, 0xA1: 反转: 列地址从右到左
(9)显示正显/反显 (Display normal/reverse)	0	1	0	1	0	0	1	1	0	显示正显/反显: 1 0xA6: 常规: 正显 0xA7: 反显
(10)显示全部点阵 (Display all points)	0	1	0	1	0	0	1	0	0	显示全部点阵: 1 0xA4: 常规 0xA5: 显示全部点阵
(11)LCD 偏压比设置 (LCD bias set)	0	1	0	1	0	0	0	1	0	设置偏压比: 1 0XA2: BIAS=1/9 (常用) 0XA3: BIAS=1/7
(12) 读-改-写 (Read-modify-write)	0	1	1	1	0	0	0	0	0	0XE0: “读-改-写” 开始。 列地址的增加: 写入时: 列地址+1 读出时: 列地址不加 详情请参考IC资料第43-44页
(13) 退出上述“读-改-写”指令(End)	0	1	1	1	0	1	1	1	0	0XEE: 上述“读-改-写”指令结束 详情请参考 IC 资料第 43-44 页
(14) 软件复位 (Reset)	0	1	1	1	0	0	0	1	0	0XE2: 软件复位。
(15) 行扫描顺序选择 (Common output mode select)		1	1	0	0	0	0	0	0	行扫描顺序选择: 0XC0:普通扫描顺序: 从上到下 0XC8:反转扫描顺序: 从下到上
(16) 电源控制 (Power control set)		0	0	1	0	1	电压操作模式选择, 共 3 位			选择内部电压供应操作模式: D2、D1、D0 位分别对应内部升压是否打开 (1 为打开, 0 为不打开), 电压调整电路是否打开 (1 为打开, 0 为不打开), 电压跟随器是否打开 (1 为打开, 0 为不打开)。 通常是 0x2C, 0x2E, 0x2F 三条指令按顺序紧接着写, 表示依次打开内部升压、电压调整电路、电压跟随器。也可以单写 0x2F, 一次性打开三部分电路。
(17) 选择内部电阻比例	0	0	0	1	0	0	内部电压值电阻设置			选择内部电阻比例 (Rb/Ra): 可以理解为粗调对比度值。可设置范围为: 0x20~0x27, 数值越大对比度越浓, 越小越淡

(18)	内部设置液晶电压模式	0	1	0	0	0	0	0	0	1	设置内部电阻微调, 可以理解为 微调 对比度值, 此两个指令需紧接着使用。上面一条指令 0x81 是不改的, 下面一条指令可设置范围为: 0x00~0x3F , 数值越大对比度越浓, 越小越淡
	设置的电压值		0	0	6 位电压值数据, 0~63 共 64 级						
(19)静态图标显示: 开/关		0	1	0	1	0	1	1	0	0 1	静态图标的开关设置: 0xAC : 关, 0xAD : 开。 此指令在进入及退出睡眠模式时起作用
(20) 升压倍数选择 (Booster ratio set)		0	1	1	1	1	1	0	0	0	选择升压倍数: 00: 2 倍, 3 倍, 4 倍 01: 5 倍 11: 6 倍。本模块外部已设置升压倍数为 4 倍, 不必使用此指令
(21) 省电模式 (Power save)											省电模式, 此非一条指令, 是由“(10)显示全部点阵”、(19)静态图标显示: 开/关等指令合成一个“省电功能”。详细看 IC 规格书第 47 页“POWER SAVE”
(22)空指令 (NOP)		0	1	1	1	0	0	0	1	1	空操作
(23) 测试 (Test)		0	1	1	1	1	*	*	*	*	内部测试用, 千万别用!

7.3 点阵与 DD RAM 地址的对应关系

请留意页的定义: PAGE, 与平时所讲的“页”并不是一个意思, 在此表示 **8 个行就是一个“页”**, 一个 128*64 点阵的屏分为 8 个“页”, 从第 0 “页”到第 7 “页”。

DB7--DB0 的排列方向: 数据是从下向上排列的。最低位 D0 是在最上面, 最高位 D7 是在最下面。每一位 (bit) 数据对应一个点阵, 通常“1”代表点亮该点阵, “0”代表关掉该点阵。 如下图所示:

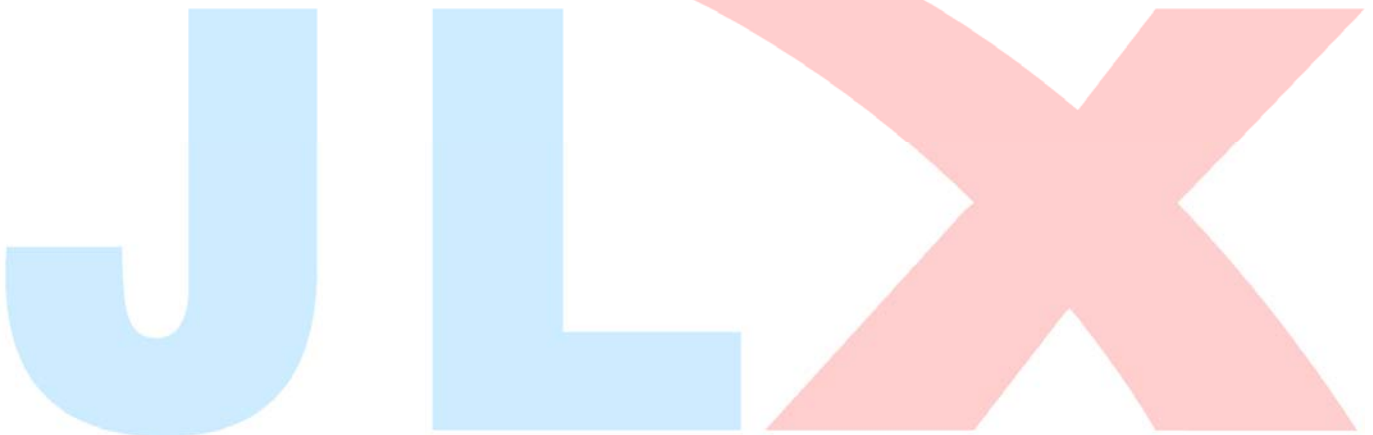
D0	0	1	1	1		0
D1	1	0	0	0		0
D2	0	0	0	0		0
D3	0	1	1	1		0
D4	1	0	0	0		0
-						

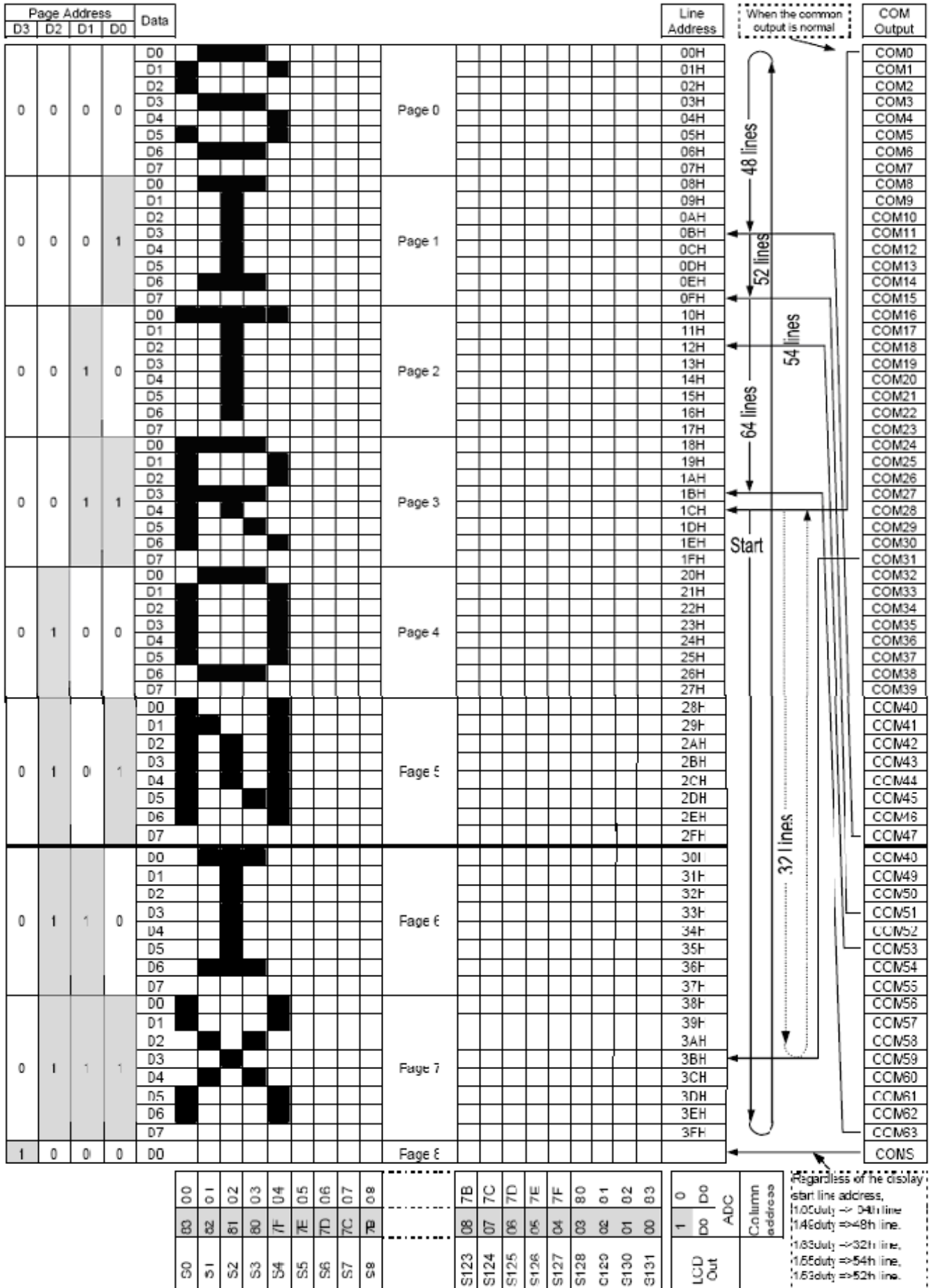
Display data RAM
(显示数据存储器)

COM0						
COM1						
COM2						
COM3						
COM4						
-						

Liquid crystal display
(液晶屏)

下图摘自 UC1701X IC 资料, 可通过 “UC1701X_V1.3.PDF” 之第 29 页获取最佳效果。





7.4 初始化方法

用户所编的显示程序, 开始必须进行初始化, 否则模块无法正常显示, 过程请参考程序

点亮液晶模块的步骤

硬件准备:
开发板 (或专门设计的主板)、单片机、电源、连接线、仿真器或程序下载器 (又名烧录器)

正确地接线
根据说明书正确地与开发板连接, 连接的线包括: 液晶模块电源线、背光电源线、IO端口 (接口)
IO端口包括: 并口时: CS、RESET、RW、E、RS、D0—D7, 串口时: CS、SCLK、SDA、RESET、RS

编写软件
背光给合适的直流电可以点亮, 但液晶屏里面没有程序, 只给电不能让液晶屏显示 (我们通常说“点亮”), 程序须另外编写, 并烧录 (下载) 到单片机里液晶模块才能工作。

7.5 程序举例:

液晶模块与 MPU (以 8051 系列单片机为例) 接口图如下:

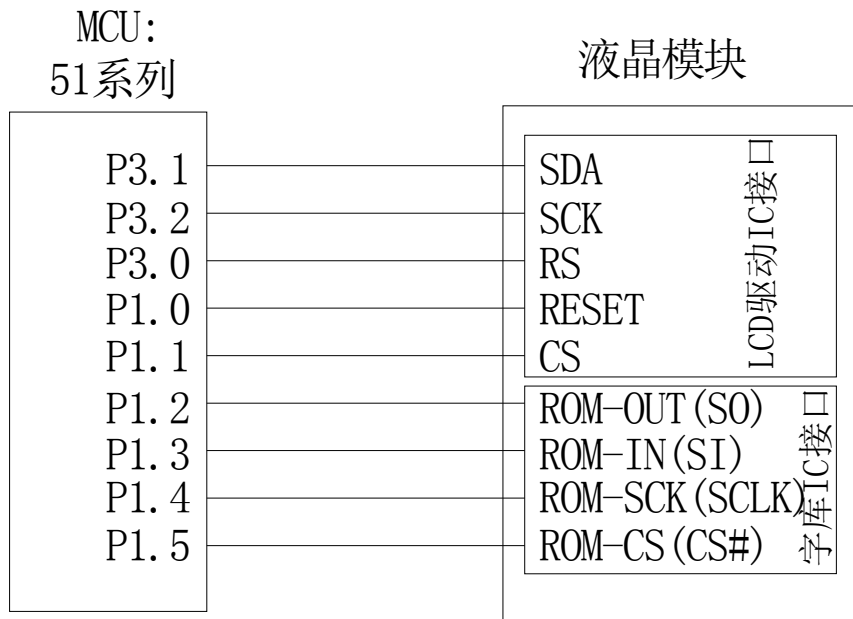


图 9. 并行接口

```

/* Test program for JLX12864G-135
   Driver IC is UC1701X(or competible)
   Programmed by Ken, Dec. 24, 2010
   JLX electronic Co., ltd,
   http://www.jlxlcd.cn;
   */
#include <reg51.H>

sbit cs1=P1^1;
sbit reset=P1^0;
sbit rs=P3^0;
sbit sclk=P3^1;
sbit sid=P3^2;//sid 就是 SDA, 串行数据信号
    
```

```

void Transfer_data(int data1);
void Transfer_command(int data1);
unsigned char code graphic1[];
    
```

```

void Delay(int i);
void Delay1(int i);
void DisplayGraphic1();
void Initial_Lcd();
void clear_screen();
void Switch();
    
```

```
//=====main program=====
void main(void)
{
    int i, j, k;
    Initial_Lcd();
    while(1)
    {
        clear_screen();    //clear all dots
        DisplayGraphic1(); //display a picture of 128*64 dots
        Switch();
    }
}

//=====initial
void Initial_Lcd()
{
    reset=0;                //Reset the chip when reset=0
    Delay(50);
    reset=1;
    Delay(50);
    Transfer_command(0xe2);    /*软复位*/
    Transfer_command(0x2c);    /*升压步聚 1*/
    Transfer_command(0x2e);    /*升压步聚 2*/
    Transfer_command(0x2f);    /*升压步聚 3*/
    Transfer_command(0x23);    /*粗调对比度, 可设置范围 20~27*/
    Transfer_command(0x81);    /*微调对比度*/
    Transfer_command(0x28);    /*微调对比度的值, 可设置范围 0~63*/
    Transfer_command(0xa2);    /*1/9 偏压比 (bias) */
    Transfer_command(0xc8);    /*行扫描顺序: 从上到下*/
    Transfer_command(0xa0);    /*列扫描顺序: 从左到右*/
    Transfer_command(0xaf);    /*开显示*/
}

//=====clear all dot martrics=====
void clear_screen()
{
    unsigned char i, j;
    for(i=0; i<9; i++)
    {
        cs1=0;
        Transfer_command(0xb0+i);
        Transfer_command(0x10);
        Transfer_command(0x00);
        for(j=0; j<132; j++)
        {
```

```

        Transfer_data(0x00);
    }
}

//=====display a picture of 128*64 dots=====
void DisplayGraphic1()
{
    unsigned char *address;    //define pointer
    int i,j;
    address=graphic1;    //data pointer:point to "graphic[]",one-dimensional array
    for(i=0;i<8;i++)
    {
        cs1=0;
        Transfer_command(0xb0+i);    //set page address,
        Transfer_command(0x10);
        Transfer_command(0x00);
        for(j=0;j<128;j++)
        {
            Transfer_data(*address);
            address++;
        }
    }
}

/*=====写指令=====*/
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;
        Delay1(2);
        sclk=1;
        Delay1(2);
        data1=data1<<=1;
    }
}

/*-----写数据-----*/
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;
    }
}

/*=====延时=====*/
void Delay(int i)
{
    int j,k;
    for(j=0;j<i;j++)
    for(k=0;k<990;k++);
}

/*=====延时=====*/
void Delay1(int i)
{
    int j,k;
    for(j=0;j<i;j++)
    for(k=0;k<10;k++);
}

/*-----等按键-----*/
void Switch()
{
    repeat:
    if (P2&0x01) goto repeat;
    else Delay(1);
    if (P2&0x01) goto repeat;
    else;
}
//-----

unsigned char code graphic1[]={
/*-- 调入了一幅图像: d:\My Documents\My Pictures\多字.bmp --*/
/*-- 宽度 x 高度=128x64 --*/
0x00,0x00,0x00,0x00,0x7E,0x2A,0x2A,0x2A,0x2A,0x2A,0x2A,0x7E,0x00,0x00,0x00,0x00,
0x02,0xFE,0x92,0x92,0x92,0xFE,0x12,0x11,0x12,0x1C,0xF0,0x18,0x17,0x12,0x10,0x00,
```

0x20, 0x21, 0x2E, 0xE4, 0x00, 0x42, 0x42, 0xFE, 0x42, 0x42, 0x42, 0x02, 0xFE, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xF8, 0x48, 0x48, 0x48, 0x48, 0xFF, 0x48, 0x48, 0x48, 0x48, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0xE2, 0x12, 0x0A, 0x06, 0x02, 0x00, 0x80, 0x00, 0x00, 0x00, 0x08, 0x08, 0x08, 0xFF, 0x08, 0x08, 0xF8, 0x00, 0xF8, 0x08, 0x08, 0x08, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0xFE, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x00, 0x00, 0x00, 0x00, 0xFE, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x00, 0x00, 0x00, 0x7F, 0x25, 0x25, 0x25, 0x25, 0x7F, 0x00, 0x00, 0x7F, 0x25, 0x25, 0x25, 0x25, 0x7F, 0x00, 0x08, 0x1F, 0x08, 0x08, 0x04, 0xFF, 0x05, 0x81, 0x41, 0x31, 0x0F, 0x11, 0x21, 0xC1, 0x41, 0x00, 0x00, 0x00, 0x00, 0x7F, 0x20, 0x10, 0x00, 0x7F, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x40, 0x38, 0x00, 0x00, 0x00, 0x0F, 0x04, 0x04, 0x04, 0x04, 0x3F, 0x44, 0x44, 0x44, 0x44, 0x4F, 0x40, 0x70, 0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x41, 0x81, 0x7F, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x00, 0x40, 0x20, 0x18, 0x07, 0x00, 0x20, 0x40, 0x3F, 0x00, 0x7F, 0x10, 0x10, 0x10, 0x3F, 0x00, 0x00, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x3F, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x00, 0x40, 0x30, 0x0F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10, 0x61, 0x06, 0xE0, 0x18, 0x84, 0xE4, 0x1C, 0x84, 0x65, 0xBE, 0x24, 0xA4, 0x64, 0x04, 0x00, 0x00, 0x00, 0x00, 0x7E, 0x2A, 0x2A, 0x2A, 0x2A, 0x2A, 0x2A, 0x7E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0x09, 0x49, 0x59, 0xE9, 0x49, 0x49, 0x49, 0xE9, 0x59, 0x49, 0x4F, 0x00, 0x00, 0x10, 0x61, 0x06, 0xE0, 0x18, 0x84, 0xE4, 0x1C, 0x84, 0x65, 0xBE, 0x24, 0xA4, 0x64, 0x04, 0x00, 0x00, 0x00, 0x00, 0x7E, 0x2A, 0x2A, 0x2A, 0x2A, 0x2A, 0x2A, 0x7E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3E, 0x2A, 0xEA, 0x2A, 0x2A, 0x2A, 0xEA, 0x2A, 0x3E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x20, 0x22, 0x22, 0x22, 0x22, 0xE2, 0x22, 0x22, 0x22, 0x22, 0x22, 0x20, 0x20, 0x00, 0x40, 0x40, 0x4F, 0x49, 0x49, 0xC9, 0xCF, 0x70, 0xC0, 0xCF, 0x49, 0x59, 0x69, 0x4F, 0x00, 0x00, 0x04, 0x04, 0xFF, 0x00, 0x01, 0x00, 0xFF, 0x41, 0x21, 0x12, 0x0C, 0x1B, 0x61, 0xC0, 0x40, 0x00, 0x00, 0x7F, 0x25, 0x25, 0x25, 0x25, 0x7F, 0x00, 0x00, 0x7F, 0x25, 0x25, 0x25, 0x25, 0x7F, 0x00, 0x40, 0x30, 0x0F, 0x82, 0x42, 0x32, 0x0F, 0x02, 0x02, 0x02, 0xFF, 0x02, 0x02, 0x02, 0x02, 0x00, 0x04, 0x04, 0xFF, 0x00, 0x01, 0x00, 0xFF, 0x41, 0x21, 0x12, 0x0C, 0x1B, 0x61, 0xC0, 0x40, 0x00, 0x00, 0x7F, 0x25, 0x25, 0x25, 0x25, 0x7F, 0x00, 0x00, 0x7F, 0x25, 0x25, 0x25, 0x25, 0x7F, 0x00, 0x20, 0x21, 0x22, 0x2C, 0x20, 0x3F, 0x20, 0x20, 0x20, 0x3F, 0x28, 0x24, 0x23, 0x20, 0x20, 0x00, 0x10, 0x08, 0x04, 0x03, 0x00, 0x40, 0x80, 0x7F, 0x00, 0x00, 0x01, 0x02, 0x0C, 0x18, 0x00, 0x00, 0x02, 0x02, 0x7E, 0x45, 0x45, 0x44, 0x7C, 0x00, 0x7C, 0x44, 0x45, 0x45, 0x7E, 0x06, 0x02, 0x00, 0x00, 0x80, 0x40, 0x60, 0x50, 0x48, 0x44, 0xC3, 0x44, 0x48, 0x50, 0x70, 0x60, 0x20, 0x00, 0x00, 0x40, 0x44, 0x24, 0x24, 0xF4, 0x5C, 0x57, 0x54, 0x54, 0x5C, 0x54, 0x24, 0x64, 0xC4, 0x44, 0x00, 0x00, 0x00, 0xFF, 0x91, 0x91, 0xFF, 0x00, 0x00, 0xFF, 0xC1, 0x41, 0x49, 0x51, 0xCF, 0x00, 0x00, 0x20, 0x10, 0x10, 0x88, 0x84, 0x4F, 0x52, 0xA2, 0x22, 0x52, 0x4A, 0x86, 0x80, 0x80, 0x80, 0x00, 0x08, 0xF4, 0x57, 0x54, 0xFC, 0x54, 0xF0, 0x42, 0xA2, 0x1E, 0x02, 0xA2, 0x62, 0x3E, 0x00, 0x00, 0x02, 0x04, 0xD8, 0x08, 0x00, 0x08, 0x08, 0x08, 0xFF, 0x08, 0x08, 0x08, 0xF8, 0x00, 0x00, 0x00, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0xF9, 0x4A, 0x4C, 0x48, 0x48, 0xC8, 0x08, 0x08, 0x08, 0x00, 0x00, 0x20, 0x2C, 0x24, 0x64, 0x74, 0xAD, 0xA6, 0xE4, 0x34, 0x24, 0x24, 0x2C, 0x24, 0x00, 0x00, 0x00, 0x40, 0x40, 0x44, 0x44, 0x44, 0x44, 0x7F, 0x44, 0x44, 0x44, 0x46, 0x44, 0x60, 0x40, 0x00, 0x04, 0x04, 0x44, 0x64, 0x57, 0x4D, 0x45, 0x65, 0x25, 0x25, 0x35, 0xE4, 0x44, 0x04, 0x04, 0x00, 0x40, 0x30, 0x0F, 0x20, 0x40, 0x3F, 0x00, 0x00, 0x7F, 0x21, 0x16, 0x08, 0x36, 0x61, 0x20, 0x00, 0x80, 0x81, 0x41, 0x42, 0x22, 0x12, 0x0E, 0x03, 0x02, 0x42, 0x82, 0x7E, 0x00, 0x01, 0x00, 0x00, 0x80, 0x7F, 0x02, 0x02, 0x5F, 0x82, 0x7F, 0x0A, 0x09, 0x09, 0x09, 0xFF, 0x09, 0x09, 0x09, 0x00, 0x02, 0xFE, 0x01, 0x80, 0x41, 0x21, 0x11, 0x0D, 0x03, 0x05, 0x09, 0x11, 0x31, 0x61, 0x21, 0x00, 0x40, 0x40, 0x20, 0x10, 0x0C, 0x03, 0x00, 0x00, 0x20, 0x40, 0x40, 0x3F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x24, 0x24, 0x25, 0x15, 0x15, 0x0D, 0xFE, 0x04, 0x0D, 0x17, 0x14, 0x24, 0x64, 0x24, 0x00,

```

0x88, 0xC8, 0xB8, 0x8F, 0xE8, 0x88, 0x88, 0x20, 0x1C, 0x0B, 0xE8, 0x08, 0x08, 0x18, 0x08, 0x00,
0x40, 0x20, 0xF8, 0x0F, 0x82, 0x60, 0x1E, 0x14, 0x10, 0xFF, 0x10, 0x10, 0x10, 0x10, 0x00, 0x00,
0x02, 0xC2, 0xF2, 0x4E, 0x42, 0xC2, 0x00, 0xFA, 0x4A, 0x4A, 0xFE, 0x4A, 0x4A, 0xFA, 0x02, 0x00,
0x40, 0x20, 0xF8, 0x0F, 0x82, 0x60, 0x1E, 0x14, 0x10, 0xFF, 0x10, 0x10, 0x10, 0x10, 0x00, 0x00,
0x00, 0x44, 0x44, 0x54, 0x55, 0xFE, 0x54, 0x54, 0x54, 0xFE, 0x55, 0x54, 0xF4, 0x44, 0x40, 0x00,
0x00, 0xFE, 0x02, 0xF2, 0x12, 0x12, 0xF2, 0x00, 0xF2, 0x12, 0x1A, 0xD6, 0x12, 0xF2, 0x02, 0x00,
0x00, 0x00, 0xFF, 0x91, 0x91, 0xFF, 0x00, 0x00, 0xFF, 0xC1, 0x41, 0x49, 0x51, 0xCF, 0x00, 0x00,
0x20, 0x10, 0x10, 0x88, 0x84, 0x4F, 0x52, 0xA2, 0x22, 0x52, 0x4A, 0x86, 0x80, 0x80, 0x80, 0x00,
0x08, 0x08, 0x08, 0x08, 0xFF, 0x04, 0x84, 0x60, 0x18, 0x06, 0x01, 0x06, 0x18, 0xE0, 0x40, 0x00,
0x00, 0x00, 0xFF, 0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0xFF, 0x01, 0x01, 0x01, 0x01, 0x01, 0x00,
0x01, 0x00, 0x3F, 0x88, 0x88, 0x5F, 0x44, 0x2B, 0x12, 0x1E, 0x23, 0x22, 0x42, 0xC3, 0x40, 0x00,
0x00, 0x00, 0xFF, 0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0xFF, 0x01, 0x01, 0x01, 0x01, 0x01, 0x00,
0x40, 0x20, 0x11, 0x09, 0x07, 0xFF, 0x01, 0x01, 0x01, 0xFF, 0x05, 0x09, 0x33, 0x60, 0x20, 0x00,
0x20, 0x1F, 0x00, 0x3F, 0x10, 0x09, 0x45, 0x40, 0x2F, 0x20, 0x10, 0x0F, 0x10, 0x2F, 0x60, 0x00,
0x40, 0x30, 0x0F, 0x20, 0x40, 0x3F, 0x00, 0x00, 0x7F, 0x21, 0x16, 0x08, 0x36, 0x61, 0x20, 0x00,
0x80, 0x81, 0x41, 0x42, 0x22, 0x12, 0x0E, 0x03, 0x02, 0x42, 0x82, 0x7E, 0x00, 0x01, 0x00, 0x00

```

```
};
→
```

