

Nokia5110 液晶屏完全新手学习笔记(一)

3 人收藏此文章, [我要收藏](#) 发表于 1 年前, 已有 3735 次阅读 共 1 个评论

小 Jay 已经很久没有接触嵌入式了, 这些天在学习 51, 还故意买了块 Nokia 5110 液晶屏来玩玩(因为便宜吧, 嘻嘻~)。

一开始, 小 Jay 不才, 看 5110 芯片资料一头雾水, 囧! 不过呢, 小 Jay 还是很努力的, 经过几番研究, 终于有些眉目, 其实 5110 也并不难的。此文提供给和我一样的鸟哥参考, 希望能帮得上忙。(*^__^*)
嘻嘻

声明:

- 1、本文章用于自己的知识回顾和提供给有需要的初学者, 高手请飘过~
- 2、本文章的内容均从“诺基亚 5110 液晶屏资料驱动芯片 PCD8544 中文数据手册.pdf”中, 结合自己的理解, 整理并简化出来的知识, 更多详细内容请查看芯片手册。另外, 内容有不对的地方, 望指正。

现在我们一步一步来:

一、硬件链接:

首先，看一下硬件（仅供参考，读者已对自己的硬件很熟悉，可以直接跳过。）：



图1

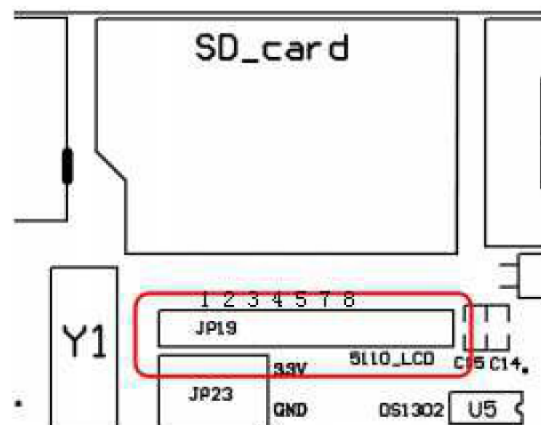


图2

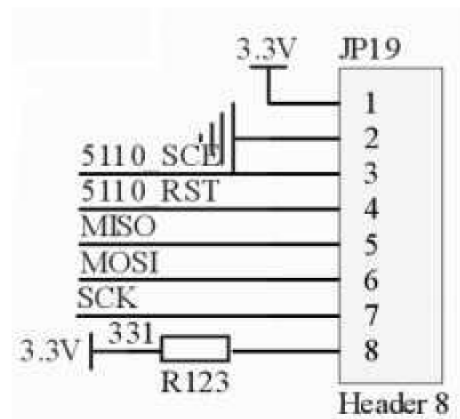


图3



图4

说明：

图1：排针从左至右1到8位。

图2：5110接口在板子中的区域。

图3：8位排针对应连接5110液晶屏相应的引脚。

图4：5110液晶显示屏对应8位排针JP19的插入方向。

再看一下我的板子的原理图：

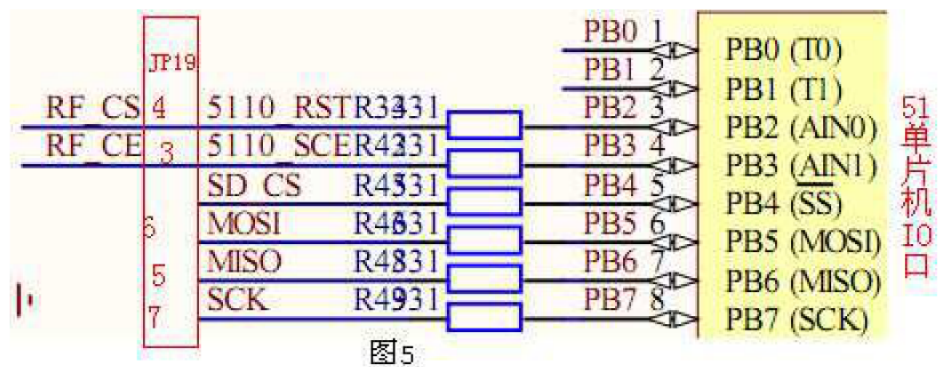


图 5 中，显示了 8 位排针 JP19（即 5110 液晶屏的引脚）其中的 5 根与 51 单片机（本人的为 STC89C516RD+）的 PB I/O 口的连接方式：

```

5110_RST    <---> PB2 (P1.2)
5110_SCE    <---> PB3 (P1.3)
MOSI(sdin)  <---> PB5 (P1.5)
MISO(dc)    <---> PB6 (P1.6)
SCK         <---> PB7 (P1.7)

```

这样，我们知道了用 51 单片机中的 I/O 口和 5110 引脚的链接方式。

二、引脚

接线图请根据程序里的定义连接，51 的程序默认接线为：

```
sbit  sce = P1^3;  //片选
```

```

sbit    res = P1^2;  //复位,0 复位

sbit    dc  = P1^6;  //1 写数据, 0 写指令

sbit    sdin = P1^5; //数据

sbit    sclk = P1^7; //时钟

```

看了这些默认接线，可能有个疑问：“这些位变量(引脚)究竟有什么用呢？”，带着这个疑问，请看 5110 芯片手册中的引脚说明（重点看红色区域）：

符号	描述
R0~R47	LCD行驱动输出
C0~C83	LCD列驱动输出
V _{SS1} , V _{SS2}	地
VDD1, VDD2	电源电压
VLCD1, VLCD2	LCD电源电压
T1	测试点1 输入
T2	测试点2 输出
T3	测试点3 输入/输出
T4	测试点4 输入
SDIN	串行数据输入端
SCLK	串行时钟输入端
D/ \overline{C}	数据/命令
\overline{SCE}	芯片使能
OSC	振荡器
\overline{RES}	外部复位输入端
Dummy1,2,3,4	没连接

从这个引脚说明中，我们知道了不同的引脚，对应的功能。那么，如何使用这些引脚来控制 5110 工作呢？

三、指令

5110 液晶显示屏 (PCD8544 芯片) , 是通过发送指令和写入数据 RAM 来控制 and 显示数据的。

指令格式分为两种模式：

- 1、如果 D/C (模式选择) 置为低(为 0) , 即位变量 $dc = 0$, 为发送指令模式 , 那么接下来发送的 8 位字节解释为命令字节。
- 2、如果 D/C 置为高 , 即 $dc = 1$; 为写入数据 RAM 模式 , 接下来的字节将存储到显示数据 RAM。

注意：

- 1、每一个数据字节存入之后 , 地址计数自动递增。 在数据字节最后一位期间会读取 D/C 信号的电平。
- 2、每一条指令可用任意次序发送到 PCD8544。 **首先传送的是字节的 MSB (高位) 。**

指令集：

表1 指令集

指令	D/C	命令字								描述
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
(H = 0 or 1)										
NOP	0	0	0	0	0	0	0	0	0	空操作
功能设置	0	0	0	1	0	0	PD	V	H	掉电控制：进入模式： 扩展指令设置 (H)
写数据	1	D7	D6	D5	D4	D3	D2	D1	D0	写数据到显示 RAM
(H = 0)										
保留	0	0	0	0	0	0	1	X	X	不可使用
显示控制	0	0	0	0	0	1	D	0	E	设置显示配置
保留	0	0	0	0	1	X	X	X	X	不可使用
设置RAM的Y地址	0	0	1	0	0	0	Y ₂	Y ₁	Y ₀	设置RAM的Y地址 0 ≤ Y ≤ 5
设置RAM的X地址	0	1	X ₆	X ₅	X ₄	X ₃	X ₂	X ₁	X ₀	设置RAM的X地址 0 ≤ X ≤ 83
(H = 1)										
保留	0	0	0	0	0	0	0	0	1	不可使用

表2 表1中的符号说明

BIT	0	1
PD	芯片是活动的	芯片处于掉电模式
V	水平寻址	垂直寻址
H	使用基本指令集	使用扩展指令集
D and E		
00	显示空白	
10	普通模式	
01	开所有显示段	
11	反转映象模式	
TC1 and TC0		
00	VLCD 温度系数 0	
01	VLCD 温度系数 1	
10	VLCD 温度系数 2	
11	VLCD 温度系数 3	

结合表 1 和表 2，指令集不难读懂。现在以红色横线所标识的指令来举例说明：

第一条（红色横线 1），功能设置指令：

首先，D/C 为 0，表示现在是指令模式，然后从表 2 看：

若要“使用基本指令集”，则 $PD = 0$, $V = 0$, $H = 0$ ，那么相应地 $DB7 \sim DB0$ 分别为 $00100000b$ ，即 $0x20$ 。也就是说，发送 $0x20$ 就能设置液晶“使用基本指令集”。

同理，使用“使用扩展指令集”，指令值为 $0x21$ 。

第二条（红色横线 2），显示控制：

显示模式有 4 种，分别是：显示空白，普通模式，开所有显示段，反转映象模式。这些模式分别使用 D and E 来组合控制。

这样，我们控制显示为“普通模式”，那么 D 要为 1， E 要为 0，这时 $DB7 \sim DB0$ 分别为 $0000\ 1100b$ ，即指令值为 $0x0C$ 。

第三条（红色横线 3），写入数据指令：

写入数据，首先 D/C 必须为 1，表示现在是写入数据 RAM 模式， $DB7 \sim DB0$ 就是相应的数据值。

到这里，我们知道了 5110 液晶屏是使用这样的指令集来控制 and 显示数据的。那么，我们如何把指令值写进去呢？

一篇发不完，请看笔记二。

