

NEC 芯片 780208 单片机的 FIP 应用

摘要：介绍了日本 NEC 公司 8 位单片机产品 μ PD780208 中的 FIP 显示控制器及显示与键盘扫描的应用方案，给出了该设计方案的原理电路和程序流程。解决了显示和键盘输入独立控制时所造成的 CPU 资源占用较多的问题。

关键词：单片机 μ PD780208 微控制器 FIP 显示控制器

1 概述

日本 NEC 公司生产的 μ PD780208 系列 8 位单片机属于 78K/0 家庭中功能较强的一种系列。根据内部集成的 ROM 和 RAM 容量的不同， μ PD780208 系列分为 5 个型号， μ PD780208 芯片是最高型号。该芯片内部集成了 8 位 78K0CPU 内核、60kB 的 ROM 和 2192B 的 RAM 之外，还有 1 个 FIP 显示驱动/控制器、8 位 A/D 转换器、2 个串行 I/O 口、5 个定时/计数器、3 个定时器输出、1 个可编程时钟输出、1 个可编程蜂鸣器输出、4 个外部中断源、12 个内部中断源和 1 个测试输入。

MPD780208 支持双时钟，其电源电压范围为 2.7~5.5V，可设定 2 种待机模式。

其中的 FIP（Fluorescent Indicator Panel）显示控制/驱动器是 μ PD780208 系列所特有的，它的主要功能有自动读取显示数据，以实现硬件自动显示刷新功能；控制显示 9~40 段和 2~16 位 FIP；通过编程自由设定位信号输出时序；并可编程调节 8 级显示亮度。另外，

μ PD780208 的显示时序中还包含有键盘扫描时序，可输出键盘扫描信号；并具有较高的驱动能力，可直接驱动 FIP 显示器。

本文主要通过笔者在项目中的实际应用，给出将 μ PD780208 中的 FIP 显示控制/驱动器用于显示和键盘扫描结合的应用方案。

2 设计方案

在笔者设计的项目中，要求 μ PD780208 硬件平台有 12 位的 9 段 VFD 显示器和

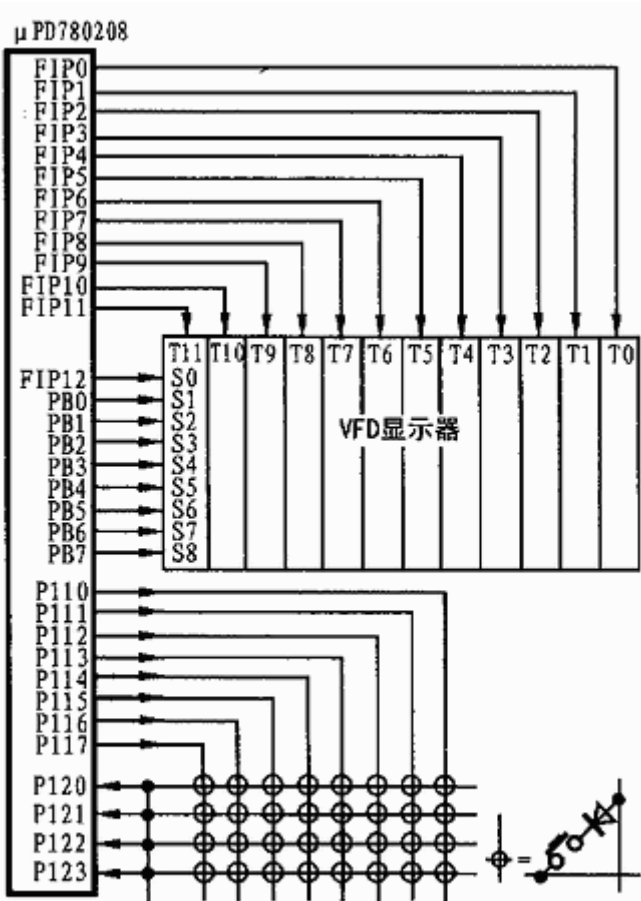


图 1 FIP 显示电路结构原理图

8×4 的键盘输入，如果按照常规设计方案，由于显示控制与键盘扫描控制是相互独立的，键盘扫描需占有一个定时器。而本方案则使用显示控制/驱动器提供的键盘扫描时序来扫描键盘，因而不占用定时资源。这使是该设计方案的优点所在。（系统所用的晶振频率为 4.9152MHz）。

2.1 硬件连接

图 1 所示是 μ PD780208 的 FIP 显示电路的结构原理图。应用时将显示端口 FIP0～FIP11 接 VFD 显示器的位控制器 T0～T11，FIP12 和 P80～P87 接 VFD 显示器的段控制端 S0～S8。 μ PD780208 的显示驱动器的驱动能力很强，因而可以直接连接 VFD 而无需驱动电路。端口 P110～P117 接键盘的 8 根扫描线，端口 P120～P123 接键盘的 4 根回读线。

2.2 初始化设置与资源分配

合理地设置显示模式寄存器 DSPM0 和 DSPM1 可将显示方式设置为 12 位 9 段模式，这时 FIP 显示 RAM 地址为 FA60H～FA6BH 和 FA70H～FA7BH。FIP 控制器的时序图如图 2 所示。图中 $n=11$ ，TDSP 为 1 位显示周期

（ $2048/4.9152\text{MHz}=416.7\mu\text{s}$ ），TKS 为键盘扫描周期（ $\text{TKS}=\text{TDSP}$ ），TCYT 为完整显示周期（ $\text{TCYT}=\text{TDSP}\times(12+1)$ ），TDIG 为位信号脉冲宽度（可编程改变）。

在 FIP 显示周期中，当处在键盘时序时，键盘扫描标志 KSF 被置 1，其它时间被清零。当 KSF 置 1 时，将产生键盘扫描中断 INTKS。在中断处理程序中进行键盘扫描和回读可用软件来设计。键盘扫描必须在 KSF 为 1 时进行，否则会造成显示混乱。1 次 INTKS 的时间是 $416.7\mu\text{s}$ ，这段时间可能来不及扫描完所有键。因此，该方案采用 1 次中断扫描一半键盘，2 次中断完成一遍扫描的方式。

键盘的消抖动问题在扫描中解决，方法是当同一位置连续 3 次检测到信号，即在 4 个 TCYT 间隔（ $4\times\text{TCYT}=21.667\text{ms}$ ）中信号一直有效时，才判断该键被按下，否则判为抖动干扰。

3 软件设计

3.1 相关的数据结构和变量

使用的数据结构和变量的定义说明如下：

```
char PreKeyStatus[8]; /*前一次 8 列键的状态存入数组的低 4 位*/
```

```
char CurKeyStatus[8]; /*当有 8 列键的状态存入数组的低 4 位*/
```

```
char ScanData; /*当前用于扫描的数据*/
```

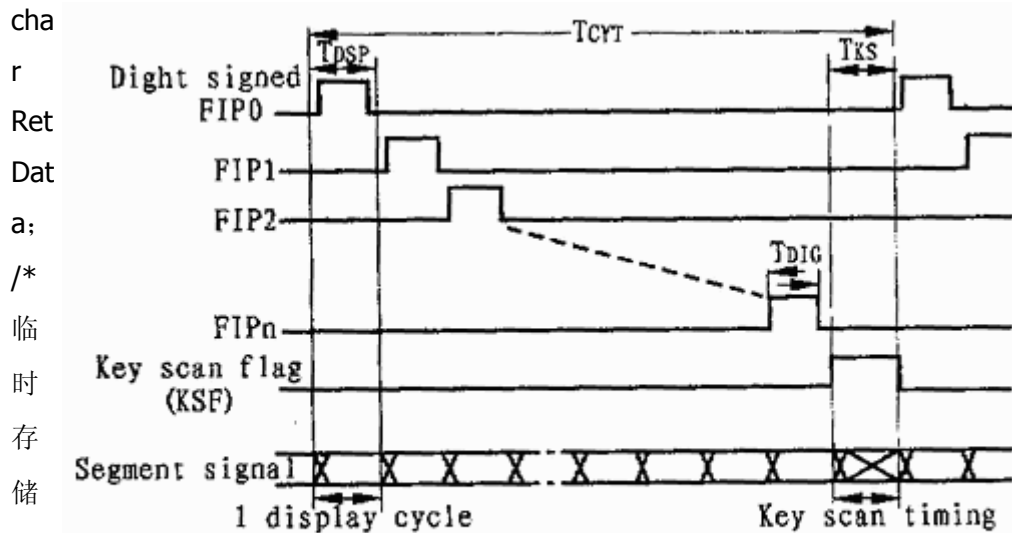


图 2 FIP 控制器时序图

数据*/

char KeyBuff[10]; /*键盘缓冲区*/

char Khead; /*键盘缓冲区头指针*/

char Krear; /*键盘缓冲区尾指针*/

char ChatterCount; /*消抖动计数器*/

char ScanEnd; /*一遍扫描结束标志*/

char KeyChanged; /*按键改变标志*/

const char DisplayCode[0][2]; /*0~9 的显示编码, 每个编码占 2 字节*/

3.2 程序初始化

进行显示方式设置、I/O 端口设置、中断系统设置初始化的具体程序段如下:

DI (); /*关中断*/

DSPM0=0x00; /*9 段显示, 段显示方式, 系统时钟 4.9152MHz*/

DSPM1=0xB3; /*23 位显示, 亮度为 2/16, 显示周期为 416.7μs*/

PM12=0x0F; /*P12 端口的低四位用于键盘回读, 设为输出模式*/

KSIF=0; /*清除中断请求标志*/

KSMK=0; /*清除中断屏蔽标志, 允许 INTKS 中断*/

Khead=9; /*初始化键盘缓冲区*/

Krear=0; /*消抖动计数器置初值*/

Scandata=0x01; /*键盘扫描数据置初值*/

ScanFinish=0; /*一遍扫描结束标志置初值*/

```
KeyChanged=0; /*按键改变标志置初值*/
```

```
EI (); /*开中断*/
```

3.3 显示程序

下面的程序可实现在指定位置显示一个数字的功能。其中能数 **digit** 为要显示的数字，范围是 0~9；**position** 为显示位置，从右向左与 0~11 相对应。

```
void DisplayDigit(char digit,char position)
```

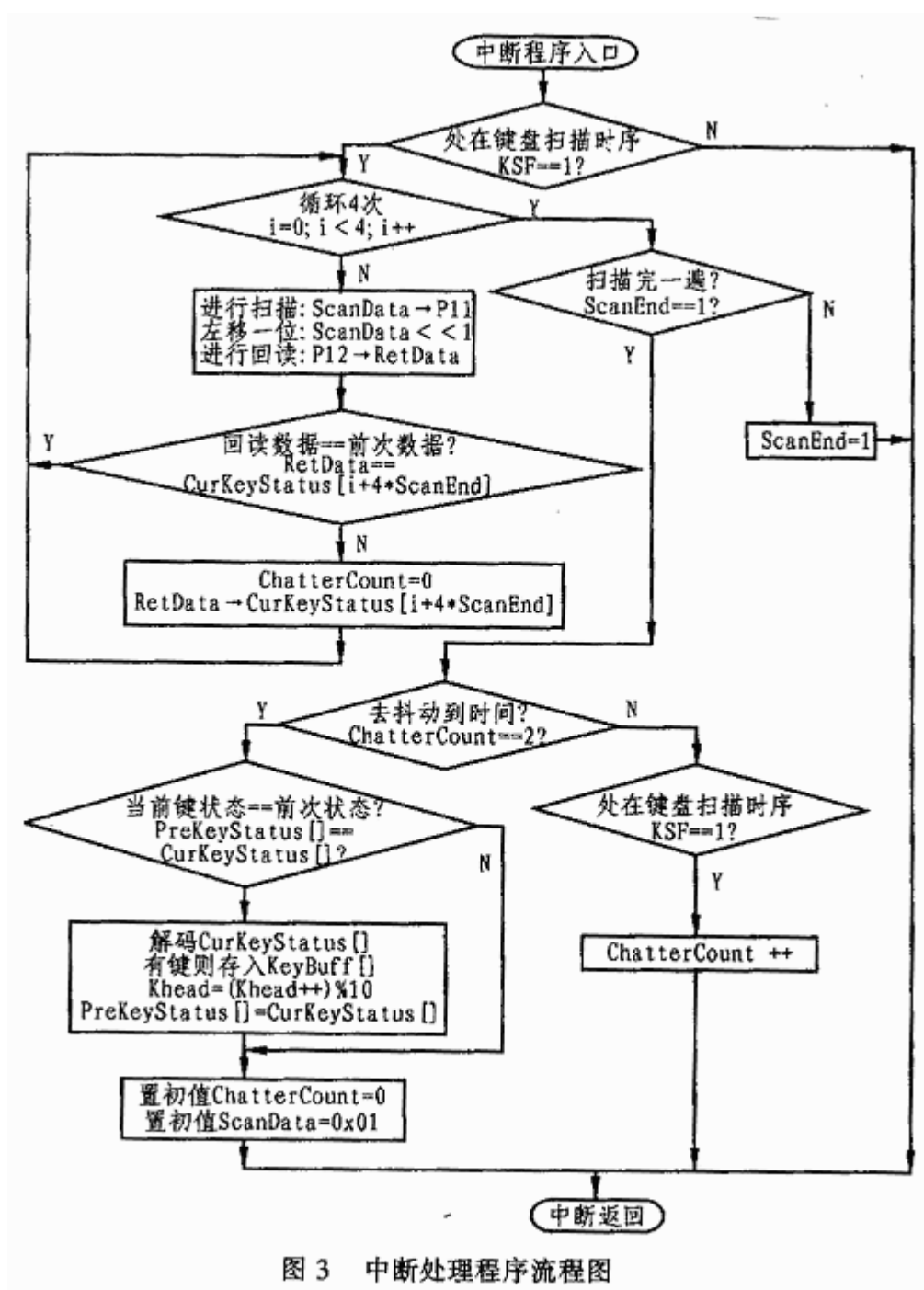
```
{if((digit<10)&&(digit>=0)&&(position>=0)&&(position<12) /*在显示范围之内*/
```

```
{pokeb(0xFA60+position,DisplayCode[digit][1]); /*编码的低字节*/
```

```
{pokeb(0xFA70+position,DisplayCode[digit][0]); /*编码的高字节*/
```

```
}
```

中断处理程序流程如图 3 所示。



4 小结

在其他单片机系统中，通常由软件定时中断来逐位刷新显示和扫描键盘，这样将增加软件的复杂性，并且会占用较多的 CPU 资源。而 μ PD780208 所具有的 FIP 显示控制/驱动器可实现硬件的自动显示刷新功能，在初始位置完成后，软件的工作只是将显示数据按要求的格式放入 FIP 显示 RAM 区而无需额外编程，由于定时刷新显示的作由硬件完成，因此在很大程序上降低了 CPU 的资源占用率。同时，由于 FIP 显示时序中提供有键盘扫描时序，从而进一步简化了软件编程，并且减少了定

时中断占用。这样，在单片机的资源并不丰富的资源条件下，FIP 显示控制/驱动器的优越性得到了充分体现。