

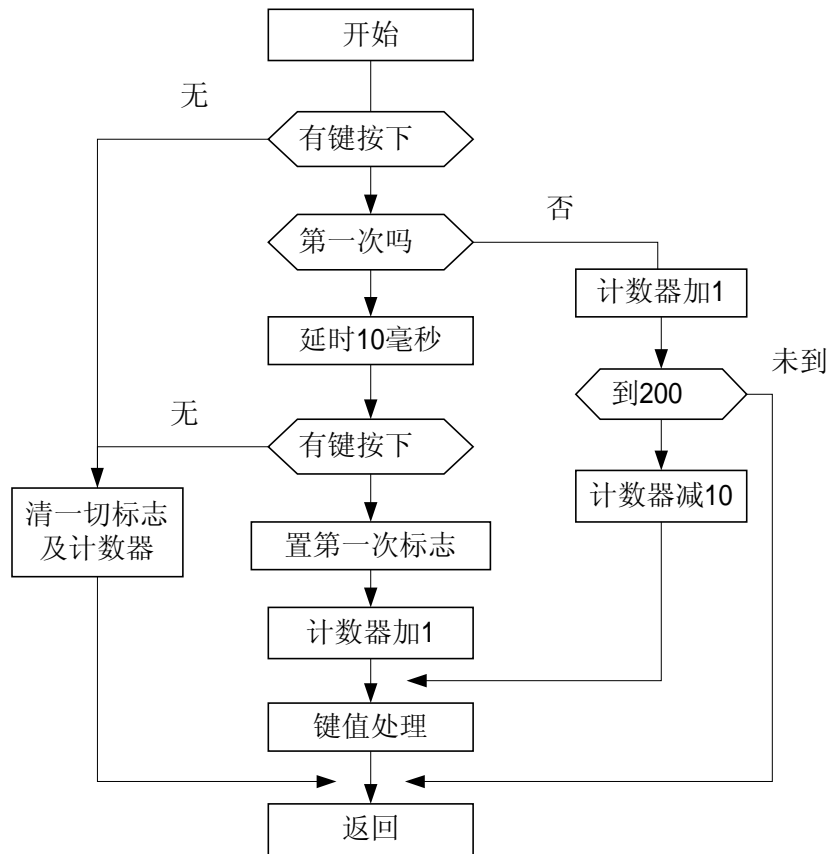
# 一种常用键盘程序的设计

摘要 本文介绍了一种工业设备中常见的键盘方案的实现思路 给出了程序框图和用 C51 语言实现的源程序

关键词 连加 减 扫描 标志

工业控制设备中常有这样一种键盘设计要求 使用 '+' '-' 键控制显示数值 要求按一次 '+' 键使显示值加 1 按一次 '-' 键使显示值减 1 如果连续按 '+' 号键超过一定时间 如 2 秒 则显示值将很快地增加 如 1 秒钟加 10 个数字 减号键也是如此 这样 就可以用很少的键完成多位数的输入工作

单片机中设计这种键盘可以用多种方法来实现 关键在于设计一个正确的程序结构 下面提供一种经过实际使用的设计方法及具体程序 用 C51 实现程序的框图如下



程序工作时 不断地扫描键盘 第一次扫描到有键按下后如常规键盘一样 进行键值处理 不同的是它并不等待键盘释放再退出键盘程序 而是直接退出键盘程序 返回主程序 继续工作 当又一次执行到键盘程序 并再次检测到键还被按着 就不再直接去键值处理程序 而是将一个计数器加 1 直接返回主程序 发此循环 直到计数器中的值到了一个定值 如 200 表示键盘程序已被执行了 200 次 如果键还被按着 说明用户有连加 或连减

要求 程序即将计数器减去一个数值(如 10) 然后进行键值处理 这样 以后键盘程序每执行 10 次 就执行一次键值处进程序 实现了第一次启动时间较长 以后快速连续动作的要求

实现时 首先定义一个 Bit 型的全局标志 KeyMark 用以确定是否是第一次被扫描到 定义方法如下 bit KeyMark; 在键盘子程序中 定义一个静态的变量 KeyCounter 用作计数器 定义方法如下 static uchar KeyCounter;

以下是该程序

```

/*P1.7,P1.6 分别为加减功能 P1.5 切换功能*/
#include <reg51.h>
#include <absacc.h>
#define uint unsigned int
#define uchar unsigned char
#define Qdsj 200 /*影响首次启动连加 减 功能的时间 具体数值根据程序而定*/
#define Ljsj 10 /*影响连加 减 的速度*/
bit KeyMark; /*第一次有键按下的标志*/
uchar KeySwitch=1; /*全程变量,根据切换键决定做什么运算*/
uchar x1,x2,x3,x4;
void Key()
{
    static uchar KeyCounter; /*键值计数器*/
    uchar KeyValue;
    P1=P1|0xe0; /*按位或 11100000B*/
    KeyValue=P1; /*取 P1 口的值*/
    KeyValue=~KeyValue; /*按位取反*/
    KeyValue=KeyValue&0xe0; /*按位与 11100000B*/
    if(!KeyValue) /*如无键按下,则前三位必为 1,取反后必为 0,则 Key=0*/
    {
        KeyCounter=0;
        KeyMark=0;
        return;
    }
    if(KeyMark) /*第一次吗?*/
    {
        KeyCounter++; /*不是第一次(KeyMark 已是 1 了)*/
        if(Qdsj==KeyCounter) /*连续按着已有 Qdsj 次了*/
        {
            KeyCounter-=Ljsj; /*减去 Ljsj 次*/
            KeyProcess(KeyValue); /*键值处理*/
        }
        else /* 如果没有到 Qdsj*/
        {
            return ;
        }
    }
}

```

```

else          /*第一次检测到有键按下*/
{
    delay(10);    /*延时 10 毫秒*/
    P1=P1|0xe0;
    KeyValue=P1;
    KeyValue=~KeyValue;
    KeyValue=KeyValue&0xe0;
    if(!KeyValue)
    {
        KeyCounter=0;
        KeyMark=0;
        return;
    }
    KeyMark=1;          /*置标志*/
    KeyProcess(KeyValue); /*键值处理*/
} /*if* end /
} /*key* end /

/*以下主程序*/
void main(void)
{
    uchar x1,x2,x3,x4; /*定义 4 个变量,根据 KeySwitch 决定它们的值*/
    void (*DispBuffer)();
    DispBuffer=0xde00;
    for(;;)
    {
        Key();
        DispBuffer();
    }
} /*本程序将键值送入显示缓冲区 在 0xde00 处有显示程序*/

```

以上方法 只能用在较为简单的程序中 如果程序本身循环时间较长 或可能在某种状态下处于等待状态 就不能用这种方法了 但是很多工控程序 键盘操作和控制操作的时间往往是分开的 在执行键盘操作时 不需要也不能进行控制 这样可以给键盘操作一个专门的循环 等到键盘操作结束后再转入控制循环 这样就可以利用这个程序了 如果不能这样做 可以用定时器做 参考如上的流程图 很容易就能实现

[平凡的单片机](#)

[E-mail:czlyzhj@cn99.com](mailto:czlyzhj@cn99.com)