

端口(I/O Port)结构及应用

端口(I/O)是单片机系统中最基本的一个资源，用于外围状态的输入和控制信号的输出。不同结构的 I/O 在使用上是有差异的，所以在使用时，要事先了解该 I/O 的结构和定义，才能正确应用。

1. 端口(I/O Port)结构

中颖公司 SH6xxx 系列单片机中，针对不同的应用，设计了几种不同的 I/O 结构。如：开漏电路(Open Drain)和 CMOS 结构。以下对这些 I/O 的结构和应用分别进行说明。

– CMOS 结构 I/O

中颖公司系列单片机典型的 CMOS I/O 端口是一个标准的双向端口，即可设置为输入端口，又可设置为输出端口。端口在作为输出时，依据具体的应用定位，可以提供大小不等的驱动能力。端口在作为输入时，呈现极高的输入阻抗，对输入信号基本可视为开路或悬浮状态。

这些端口的输入或输出状态完全由用户通过软件设置相应控制寄存器进行控制。并且每个端口都互相独立。

CMOS I/O 端口的结构分为三种。

图 1-1 为 CMOS I/O 结构图 1：

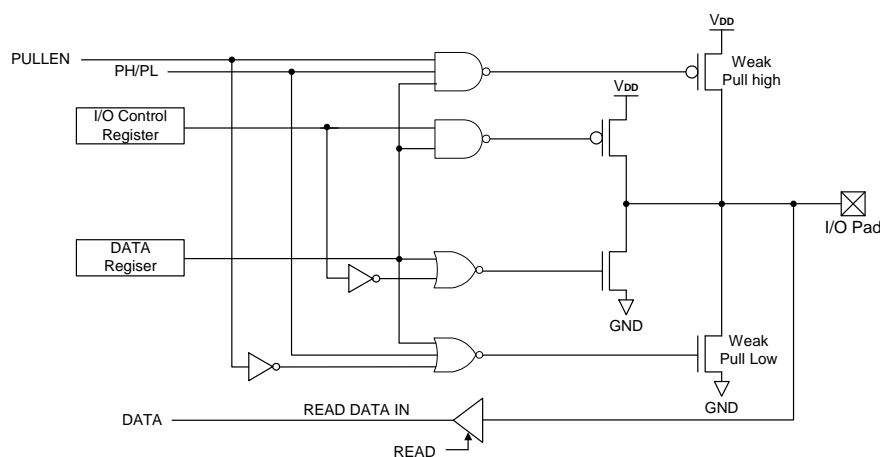


图 1-1 CMOS I/O 结构图 1

特点：

1) 此端口设计中有内建的上拉/下拉 CMOS 电阻（几十至几百千欧姆不等，依单片机型号而定），此功能仅在端口设置为输入状态时才能启用，在

输出状态会自动关闭，且每个端口互相独立。

2) 通过寄存器 PULLEN 控制位选择是否允许上拉/下拉电阻功能；通过寄存器 PH/PL 控制位选择上拉或下拉电阻功能。这些内建的上拉/下拉 CMOS 电阻在设计中可以节省电路板上的外部上拉/下拉电阻及电路板面积。

图 1-2 为 CMOS I/O 结构图 2:

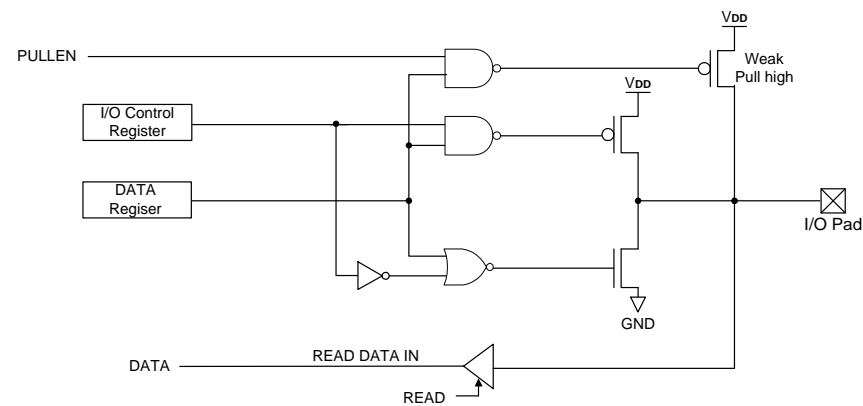


图 1-2 CMOS I/O 结构图 2

此端口设计中只有内建的上拉 CMOS 电阻（几十至几百千欧姆不等，依单片机型号而定），此功能仅在端口设置为输入状态时才能启用，在输出状态会自动关闭，且每个端口互相独立。通过寄存器 PULLEN 控制位选择是否允许上拉电阻功能。

图 1-3 为 CMOS I/O 结构图 3:

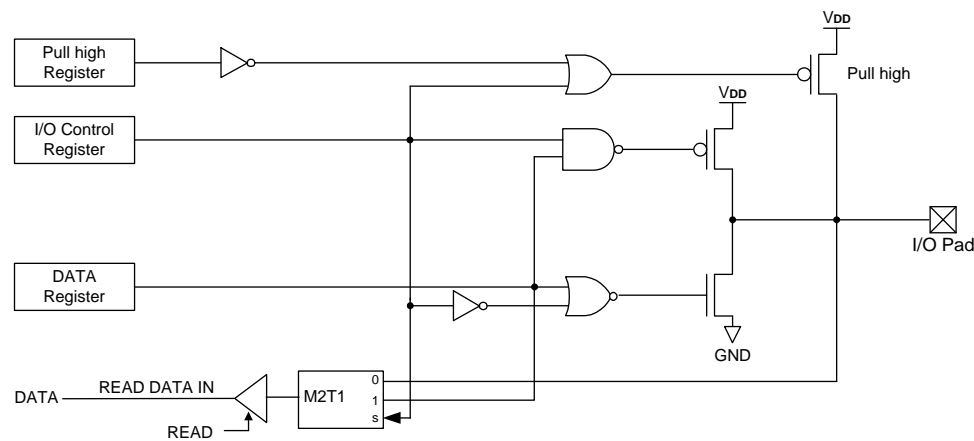


图 1-3 CMOS I/O 结构图 3

此端口设计中有内建的上拉 CMOS 电阻（几十千欧姆），此功能仅在端口设置为输入状态时才能启用，在输出状态会自动关闭，且每个端口互相独立。通过寄存器 PULLEN 控制位选择是否允许上拉电阻功能。

开漏结构(Open Drain) I/O

图 1-4 是开漏结构的 I/O 端口结构：

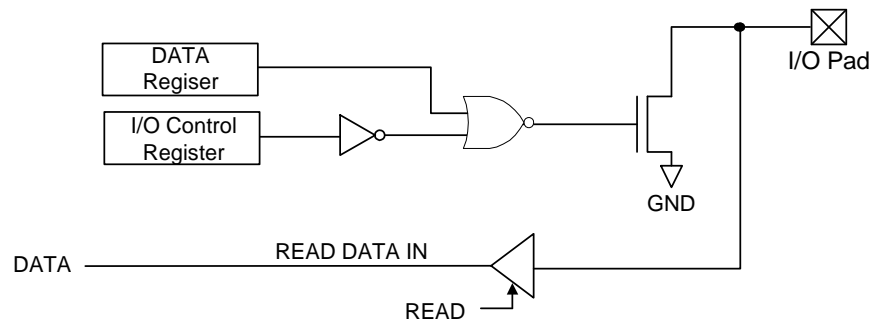


图 1-4 开漏结构 I/O 结构图

开漏结构端口输出“0”时，在引脚输出低电平。而输出“1”时，在引脚输出表现为高阻抗。如果要取得高电平则需要使用外部上拉电阻，如图 1-5。此外部高电平可以高于或低于单片机的工作电压，使用户可以根据外围电路需求取得需要的输出控制电平。同时这一特性可允许多个开漏结构端口并联使用，而不会造成冲突。

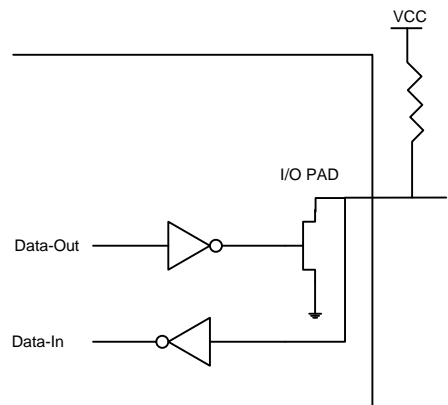


图 1-5 开漏结构端口的应用

2. 端口(I/O Port)的操作

每个端口都有独立的输入/输出状态控制寄存器（PxCR）。其中 x 对应 A, B, C, D 等不同端口。通过用户软件对这些状态控制寄存器写入 0 或 1，就可以独立控制每一个端口的输入/输出状态，0 为输入状态，1 为输出状态。一旦向端口输入/输出状态控制寄存器（PxCR）写入数据后，端口将一直保持用户设定的状态，直至下一次被改写。

端口在系统复位后(上电复位、外部 RESET、低电压检测复位及看门狗溢出复位)的初始状态为输入悬空状态（数据寄存器复位为 0、PxCR 中数据复位为 0）。为了避免单片机复位时，端口的不确定状态对外部电路的工作状态产生影响，对处

于输入悬空状态的端口，可以通过在外部加上拉/下拉电阻加以解决, 使在系统复位开始到端口被重新设置期间，外部电路都能保持在正确的工作状态。

不使用的 I/O 端口不可以处于悬空输入状态。建议设置为输入，同时使用内部或外部上拉或下拉电阻，或设置为输出低电平。

相关寄存器介绍：

端口数据寄存器

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$08	PA. 3	PA. 2	PA. 1	PA. 0	读/写	端口 A 数据寄存器

此寄存器除了可直接控制输出” 0” 或” 1” 于端口外，在大部分产品中还可以用于控制各个端口上拉或下拉电阻的开与关。

端口模式控制寄存器

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$16	PACR. 3	PACR. 2	PACR. 1	PACR. 0	读/写	端口 A 控制寄存器 写入 0：设置为输入端口 写入 1：设置为输出端口

这个寄存器用于控制端口的输入或输出，每个 bit 都可独立地设为输入或输出。

上拉/下拉电阻控制寄存器

地址	第 3 位	第 2 位	第 1 位	第 0 位	读/写	说明
\$13	PULLEN	PH/PL	X	X	读/写	端口上拉/下拉电阻控制 PULLEN=0：关闭上拉/下拉电阻功能 PULLEN=1：开启上拉/下拉电阻功能 PH/PL=0：选择下拉电阻 PH/PL=1：选择上拉电阻

此寄存器与端口数据寄存器配合控制上拉电阻的开与关。

下面以实例来说明 I/O 端口的设定。

用户可根据不同的应用选择开启上拉电阻或下拉电阻，下面以选择 PORT A 为例说明上下拉电阻的设定。

设 PORT A 为输入打开下拉

步骤 一 、 对端口模式寄存器 PA3OUT 写 “0” 把端口设为输入

步骤 二 、 对上下拉电阻控制寄存器 PULLEN 和 PH/PL 分别写 “1” 和

“0”

步骤 三 、 对端口数据寄存器写 “0”

例 2-1：PORTA 设值

```
PORTA      EQU      08H      ;端口数据寄存器
PULLCTL    EQU      15H      ;上(下)拉电阻控制寄存器
PACR       EQU      13H      ;端口模式寄存器

:
:

LDI        PACR      , 00H      ;设端口 A 为输入
LDI        PULLCTL   , 08H      ;
LDI        PORTA     , 00H      ;打开端口 A 的下拉电阻
```

端口的读写操作

每个端口都有独立的端口数据寄存器（Px）。其中 x 对应 A，B，C，D 等不同端口。

当端口为输入状态时，对端口数据寄存器（Px）进行读操作可以直接获得引脚上的外部电平状态，对端口数据寄存器（Px）进行写操作则只会改写数据寄存器的内容而不会影响引脚的外部电平状态。

当端口为输出状态时，对端口数据寄存器（Px）进行写操作则会改写数据寄存器的内容，同时在引脚上加以输出。对端口数据寄存器（Px）进行读操作则分为两种：

- 1) 对于如“CMOS I/O 结构 3”中所示的端口，在输出状态下执行读操作时，读入的数据是数据寄存器（Px）中保存的数据，而非引脚上的电平状态。
 - 2) 对于如“CMOS I/O 结构 1 ， 2”中所示的端口，在输出状态下执行读操作时，读入的数据是引脚上的外部电平状态，而非数据寄存器（Px）内数据。
- 针对上述状况(在输出状态下的读操作的数据来源存在差异)，程序中直接对 I/O 端口进行逻辑指令操作需要慎重处理,因为 I/O 逻辑操作指令的执行过程是先执行读操作，再执行逻辑运算，最后执行写操作，也称为 Read-Modify-Write 指令。

端口的输出驱动能力

每个端口的输出驱动能力由组成 I/O 结构的 P 沟道和 N 沟道场效应管决定。在输出为空负载时，输出高电平接近 VDD，输出低电平接近 0V。依不同型号，其驱动能力从几毫安至上百毫安不等（请参考具体型号单片机规格书）。在一些低功耗设计的应用场合中，具备较强输出驱动能力的端口在电路中可以作

为部分外围器件的电源使用，输出“1”开始供电，输出“0”，切断外围电路供电，达到省电目的(Power down)。

中颖公司 SH69, SH67 系列单片机中, I/O 端口都内置静电 (ESD) 保护电路，以保护单片机不受静电损伤及干扰程序运行。

3. 特殊应用实例

I/O 端口用于市电（交流电）过零检测

单片机的 I/O 端口存在着内置的 ESD 保护二极管，其作用在于将施于 I/O 端口的外部输入电压钳位在 $V_{DD} \sim 0V$ 范围之内，以保护单片机不被损坏。在确保输入电流不超过 1mA 时，此特性可以用于市电（交流电）的过零检测。电路如下图 1-6：

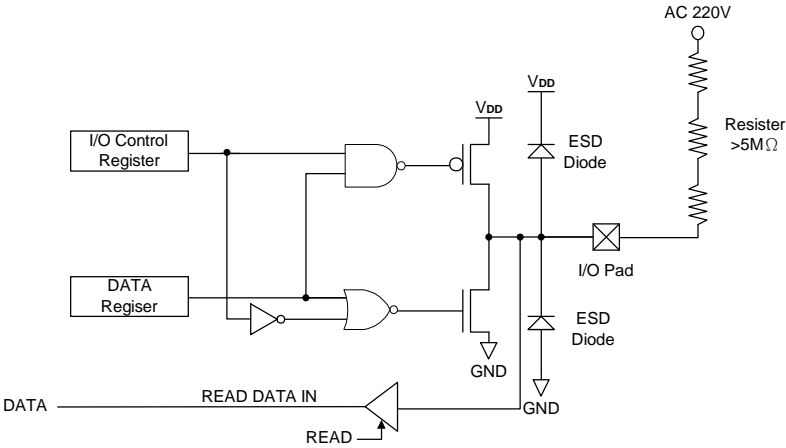


图 1-6 I/O 端口用于市电（交流电）过零检测原理示意图

将 I/O 端口和市电（交流电）通过 5 兆~10 兆欧姆的限流电阻直接连接，当市电高于单片机工作电压 V_{DD} 时，I/O 端口通过 ESD 保护二极管使输入电压钳位在 V_{DD} 处，I/O 读入状态为 1。当市电由高于 $1/2 V_{DD}$ 转为低于 $1/2 V_{DD}$ 时，I/O 读入状态也由 1 变 0，此时可以认为是交流电的过零状态。当市电电压低于 0V 时，I/O 端口通过 ESD 保护二极管使输入电压钳位在 0V 处，I/O 读入状态为 0。

这样就可以通过读入 I/O 端口数据的变化得到交流电的过零点。程序可以通过端口中断或定时扫描端口的处理方式来实现交流电的过零检测功能。

应用中需注意：

1. 交流电一般会掺杂较大的噪声，中断或扫描的处理中要考虑增加去抖动处理才不会多计入过零点。
2. 限流电阻应采用多个电阻串联，以防单个电阻的耐压不够导致损坏限流电

阻。

3. 开漏结构的端口因为没有对 VDD 的 ESD 保护二极管，所以不能作此应用。

交流电过零点开关灯泡的应用实例

1) 电路设计

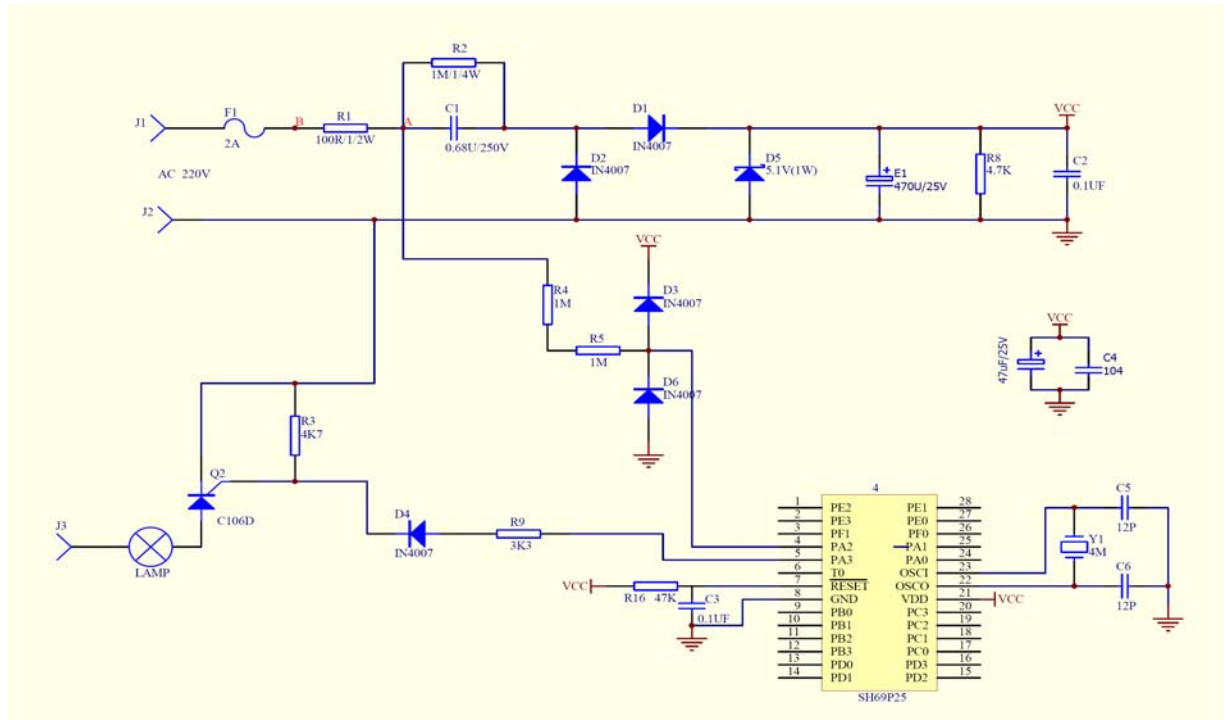


图 1-7 交流电过零点开关灯泡电路

电路采用交流电源，4M 晶振，交流电通过阻容降压（R2 和 C1）后，通过二极管 D1 和 D2 进行半波整流，然后再经稳压管 D5 稳压到 5.1V，提供 IC 工作的电压。

可控硅 C106D 是单向可控硅，当芯片检测到交流电过零时，通过 PA. 3 口控制可控硅打开/关闭灯泡，图中电阻 R3 和二极管 D4 起保护作用，J3 接于图中 B 点。

交流电经过保险管 F1 和限流电阻 R1 后，再经过两个 1MΩ 串联的限流电阻 R4 和 R5 流入 PA. 2 口，这样流入 PA. 2 口的最大电流仅约为 0.16mA，远远小于该 I/O 口的最大灌电流；直接接到 PA. 2 口上的两个二极管 D3 和 D6（通常芯片 I/O 引脚内部已经有这两个二极管，此例中为了更加安全，在外部也增加了这两个二极管），可以保证输入到 PA. 2 口的电压值在芯片的正常工作范围内。

例 3-1：交流电过零点开关灯泡

程序主要功能：在 TIMERO 中检测交流过零信号，每当检测到过零时，对可控硅控制口输出进行取反，从而控制灯泡在交流过零时开关。

```
*****  
;  
; 系统寄存器的定义  
*****
```

```

IE            EQU            00H            ;中断使能标志
IRQ           EQU            01H            ;中断请求标志
TMO           EQU            02H            ;定时器 0 模式寄存器
TLO           EQU            04H            ;定时器 0 装入/计数值低 4 位
THO           EQU            05H            ;定时器 0 装入/计数值高 4 位
PORTA         EQU            08H            ;PORT A 数据寄存器
SETTING       EQU            15H

;*****
;                                ;BIT3:端口内部上/下拉电阻使能控制
;                                ;BIT2:端口内部上/下拉电阻控制
;                                ;BIT1:端口 B 端口 C 上升沿/下降沿中断控制
PAOUT         EQU            16H            ;PORT A 输入/输出状态控制寄存器
WDT           EQU            1EH            ;看门狗定时器
;*****
;                                用户寄存器的定义
;*****
AC_BACKUP     EQU            20H            ;各中断中备份累加器 AC 的值
WORK_FLAG0    EQU            21H            ;BIT0:交流过零标志
;                                ;BIT1~3:未使用
PORTA_B       EQU            22H            ;PORTA 口备份寄存器
;*****
;                                向量地址区域
;*****
                ORG            00H
                JMP            RESET            ;跳转到复位服务子程序的入口地址
                RTNI            ;保留
                JMP            TIMEROINT        ;跳转到定时器 0 服务子程序的入口地址
                RTNI            ;保留
                RTNI            ;端口 B&C 中断服务子程序的入口地址
;*****
;                                定时器 0 中断服务子程序
;*****
TIMEROINT:
                STA            AC_BACKUP, 00H        ;保护累加器 AC 的值
                ANDIM          IRQ, 1011B            ;清除定时器 0 中断请求标志
PASS_ZERO_CHECK_PART:
;                                ;过零检测部分
                ANDIM          PAOUT, 1011B        ;设置 PA. 2 做为输入口
                NOP
                NOP

```



```

        LDA        PORTA , 00H                ;读取 PA. 2 的值
        BA2        PASS_ZERO_CHECK_PART10
        ANDIM      WORK_FLAG0, 1110B          ;设置本次读取 PA. 2 的状态为 0
        JMP        PASS_ZERO_CHECK_PART99

PASS_ZERO_CHECK_PART10:
        LDA        WORK_FLAG0, 00H            ;检测 PA. 2 口上次状态
        BAO        PASS_ZERO_CHECK_PART99
        ORIM       WORK_FLAG0, 0001B          ;设置交流过零标志

        EORIM      PORTA_B, 1000B             ;打开/关闭可控硅输出
        STA        PORTA, 00H
        ORIM       PAOUT, 1000B
        JMP        PASS_ZERO_CHECK_PART99

PASS_ZERO_CHECK_PART99:
TIMER0INT_OUT:
        LDI        IE, 0100B                  ;打开定时器 0 中断使能标志:进入中断后, 硬件
会自                                           ;动清除中断使能标志, 所以中断服务子程序执
                                           ;行后
                                           ;要手动恢复中断使能标志

        LDA        AC_BACKUP, 00H             ;恢复累加器 AC 的值
        RTNI

;*****
;                程序初始化部分
;*****

RESET:
        LDI        IE, 00H                    ;清除中断使能标志
        LDI        TM0, 0FH                    ;设置定时器 0 的模式, 此处设定为 1 分频
        LDI        TL0, 06H                    ;设定定时器 0 的定时时间, 工作频率为 4MHZ, 此
处                                           ;设定定时时间为 250 微秒

        LDI        TH0, 00H                    ;先装入定时器 0 计数值低 4 位, 再装入定时器 0
计                                           ;数值高 4 位

        LDI        PORTA, 00H                  ;设置 PORTA 口为低电平
        LDI        PAOUT, 1011B                ;设置 PA. 3 作为输出口, PA. 2 作为输入口
        LDI        PORTA_B, 00H                ;初始化 PORTA 口备份寄存器
        LDI        AC_BACKUP, 00H              ;初始化累加器 AC 的备份寄存器

```

```

LDI      WORK_FLAG0, 00H          ;初始化 WORK_FLAG0 寄存器
LDI      IRQ, 00H                 ;清除中段请求标志
LDI      IE, 0100B                ;打开定时器 0 中断使能标志
JMP      MAIN_LOOP_PART           ;跳转到主程序部分

;*****
;                               主程序部分
;*****

MAIN_LOOP_PART:
LDI      IE, 0100B                ;打开定时器 0 中断使能标志
LDI      WDT, 0FH                 ;清看门狗寄存器，防止看门狗溢出
NOP
HALT                                ;进入 HALT 低功耗模式
NOP
NOP
JMP      MAIN_LOOP_PART
END                                ;程序结束

```

2) 程序运行时过零检测口 PA. 2 和可控硅控制口 PA. 3 的波形如图 1-8。

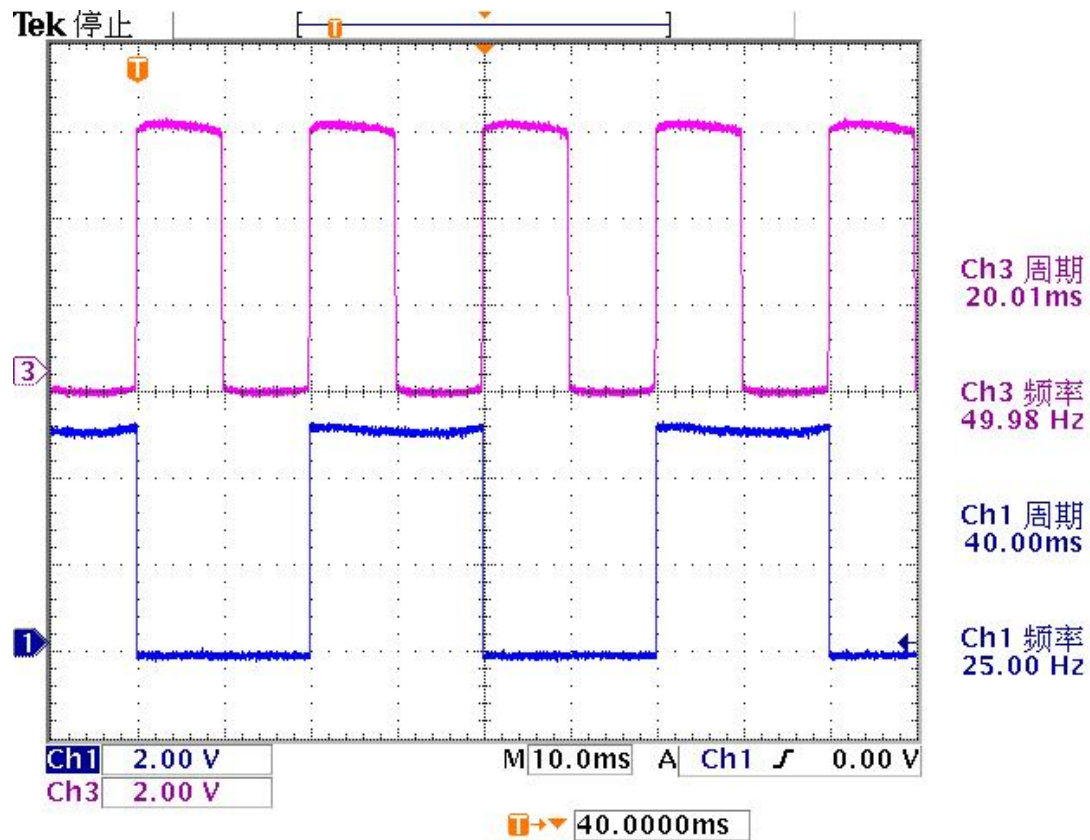


图 1-8 PA. 2 & PA. 3 波形图

图中 CH3 波型为过零检测口 PA. 2 的输入波型, CH1 的波型为可控硅控制口 PA. 3 的输出波型。

从图中可以看到过零信号的周期约为 20.01ms, 频率约为 49.98HZ, 可控硅控制口 PA. 3 输出波型的周期为 40.00ms, 频率为 25.00HZ, 与程序设计的要求完全符合。

用 I/O 测量电阻值

在日常生活中, 我们时常需要通过温度来控制一些对象, 而温度参数大都通过 A/D 模块采样来获得。对于高精度的控制对象, 我们肯定需要通过 A/D 模块采样来求得参数, 在一些精度不需要太高, 采样参数很少, 而且芯片成本要低 (没有 AD 模块) 的情况下, 我们可以通过 I/O 来测量参数。例如房间空气调节器或者空调器遥控器, 需要做一个实际环境温度的测量, 在精度不要求很高的情况下, 完全可以通过 I/O 来测量温度参数。

1) 测量原理

首先我们熟悉一下温度传感器。通常温度传感器是负系数的, 即温度升高, 电阻值降低, 如果温度降低, 则传感器的电阻值增大。然后我们可以通过阻容 RC 测量充放电时间来间接求得电阻值。考虑到环境变化下, 电阻电容的参数都会有偏移, 所以我们采用比较的方式来测量电阻值。电路中我们采用一个高精度的金属膜电阻作为参考电阻, 然后用所测试的电阻的充电时间同参考电阻的充电时间比较来求取电阻值。

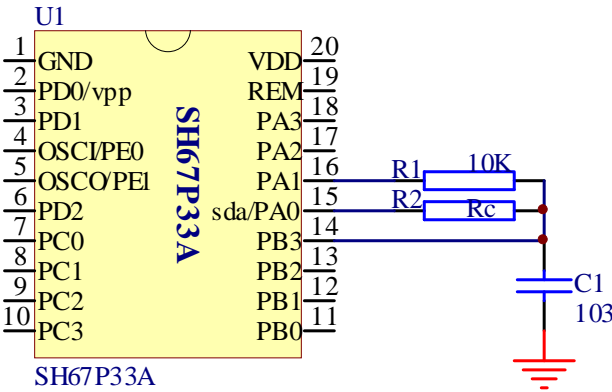


图 1-9 SH67P33A I/O 测温原理图

在实际应用中, 我们要考虑到测量的误差, 所以我们可以求取参数时采用采样多次后求取平均值的方法来降低误差。通常情况下我们采样 10-20 次左右就可以了。在求取了时间参数后, 我们就可以采用运算或者查表的方式来求取电阻

值。同时我们采用芯片内部定时器来计数以减少误差，这比软件计数要准备和精度高。

RC 测温度基本原理是，通过电阻向 C 充电，用 TMR0，或 TMR1，或其它办法检查充电的时间，有了时间就算出电阻值，然后就可以查表计算出温度。由于单个 RC 电路要受 R、C 的误差限制，一般的是用分别用 2 个电阻（一个是热敏电阻，一个是标准电阻）对同一个 C 充电。

2 个电阻分别接一个 I/O 口，分别对 C 充电。在一个端口作充电时间测量的时候，另一个端口要设置成输入，这样这个端口就成高阻状态而减少对测量端口的干扰。在充电测试完毕后一定要将 RC 彻底的放电再进行下一次的充电时间测量。

公式是： $R_{\text{热敏}} = T_{\text{热敏充电时间}} * R_{\text{标准}} / T_{\text{标准充电时间}}$

可以看出，公式排除了电容 C 的误差影响，精度提高了，并且对电容的选择要求就低了。对标准电阻要选择温度系数低的金属膜精密电阻。

设计过程中要注意电阻、电容值的选择，保证充电时间要落在记时器里面。为了保证计算方便，一般的是标准电阻选所测温度对应的热敏电阻值范围内的中间值。

另：C 在一次充电完成后要及时彻底放电。在 R、C 连接点并一个 I/O 口，C 充电时候，I/O 口设成输入并记录电平跳高，放电就设成输出并置 0

2) 计算方法：

$$T_{\text{ref}} = k \times R_f \times C \quad (1)$$

$$T_{\text{rt}} = k \times R_t \times C \quad (2)$$

由方程式（1）、（2）可得到：

$$T_{\text{ref}} / T_{\text{rt}} = R_f / R_t$$

由此可以算出 R_t 的值，再经查表即可得到温度值。

3) 用 RC 充放电测温之局限性：

由于随着 VDD 变动电阻值也相应变化，因此当 VDD 下降时温度测量误差将会增大，而且在高温情况下误差较明显。

4) 测量精度的控制

时间的测量我们采用芯片内部的定时器技术来获取。在测量之前我们可以先估算一下测试的结果值，然后确定需要多少位寄存器来计数。

5) 程序设计

下面例程是按照上面原理图来测试图中的 R_c ，例程中 AD 采样的精度设置为 12 位。参照上面 1-9 原理图计算，参考电阻的时间参数大概是： $10K \times 10^4 \times 10^{-12} = 100\mu s$ 。所以将定时器设计成 1us 计数，考虑到测试电阻值的大小，设计为 3 个寄存器保留参数值。这样精度就是 12 位了。

例 2-3 用 I/O 测量电阻

```
;;;;;;;;;;;;;system
define ;;;;;;;;;;;;;;

    IE            EQU        00H        ;中断允许标志
    IRQ           EQU        01H        ;中断请求标志
    PA            EQU        08H        ;PORT A 数据寄存器
    PB            EQU        09H        ; PORTB 数据寄存器
    PC            EQU        0AH        ; PORT C 数据寄存器
    PD            EQU        0BH        ; PORT D 数据寄存器
    PE            EQU        0CH        ; PORT E 数据寄存器
    PACR          EQU        16H        ; PORT A 输入输出控制寄存器
    PBCR          EQU        17H        ; PORT B 输入输出控制寄存器
    PCCR          EQU        18H        ; PORT C 输入输出控制寄存器
    PDCR          EQU        19H        ; PORT D 输入输出控制寄存器
    PECR          EQU        1AH        ; PORT E 输入输出控制寄存器
    PMOD          EQU        12H        ;BIT0-BIT2: 载波输出时钟预分频设定
                                ;BIT3: 下拉允许控制位

    DPL           EQU        10H        ;数据指针低位
    DPM           EQU        11H        ;数据指针中间位
    DPH           EQU        12H        ;数据指针高位
    INX           EQU        0FH        ;间接索引寄存器

;;;;;;;;;;;;; ram
define ;;;;;;;;;;;;;;

    ACCBK         EQU        20H        ;备份 ACC 寄存器
    DELAYL        EQU        21H        ;延时函数输入参数低位
    DELAYM        EQU        DELAYL+1   ;延时函数输入参数中位
    DELAYH        EQU        DELAYL+2   ;延时函数输入参数高位

    TEMP         EQU        24H        ;临时使用寄存器
```

TEMPL	EQU	25H	
TEMPH	EQU	26H	
DPL_BK	EQU	27H	;移位函数临时寄存器
DPM_BK	EQU	DPL_BK+1	;移位函数临时寄存器
PARA_BYTE_BK	EQU	DPL_BK+2	;移位函数临时寄存器
PARA_BYTE	EQU	DPL_BK+3	;移位函数输入参数，移位字节
PARA_BIT	EQU	DPL_BK+4	;移位函数输入参数，移位位数
TEMPSUB	EQU	DPL_BK+5	;移位函数临时寄存器
PBBUF	EQU	2DH	;PORTB 输出缓冲寄存器
TOSUM	EQU	2EH	;TIMER0 溢出计数器
ADTIME	EQU	2FH	;AD 采样计数器
REFER_TOL	EQU	30H	;参考电阻当次采样定时器低位寄存器
REFER_TOH	EQU	31H	;参考电阻当次采样定时器高位寄存器
REFER_TOSUM	EQU	32H	;参考电阻当次采样定时器溢出寄存器
TEST_TOL	EQU	34H	;测试电阻当次采样定时器低位寄存器
TEST_TOH	EQU	35H	;测试电阻当次采样定时器高位寄存器
TEST_TOSUM	EQU	36H	;测试电阻当次采样定时器溢出寄存器
REFER_SUML	EQU	38H	;参考电阻采样平均值低位
REFER_SUMM	EQU	39H	;参考电阻采样平均值中位
REFER_SUMH	EQU	3AH	;参考电阻采样平均值高位
TEST_SUML	EQU	3CH	;测试电阻采样平均值低位
TEST_SUMM	EQU	3DH	;测试电阻采样平均值中位
TEST_SUMH	EQU	3EH	;测试电阻采样平均值高位
DIVL	EQU	44H	;除数
DIVM	EQU	DIVL+1	
DIVH	EQU	DIVL+2	
BDIV_PL	EQU	DIVL+3	
BDIVL	EQU	DIVL+4	;被除数
BDIVM	EQU	DIVL+5	
BDIVH	EQU	DIVL+6	
RESULT_PL	EQU	DIVL+7	;结果的小数部分
RESULTL	EQU	DIVL+8	;商的整数低位

```

RESULTM      EQU      DIVL+9      ;商中整数间位
RESULTH      EQU      DIVL+0AH    ;商的整数高位
;:::::::::::: port
define ;::::::::::::
REFER      EQU      02H      ;PA1
TEST       EQU      01H      ;PA0
SAMPLE     EQU      08H      ;PB2
ADPORTCR    EQU      PBCR      ;AD 采样端口控制寄存器
ADPORT      EQU      PB        ;AD 采样端口
IETO       EQU      04H
SCALE1     EQU      0111B
SCALE8     EQU      0100B
SCALE2048  EQU      0000B

      ORG  0000H
      JMP  RESET              ;复位入口地址
      JMP  ONLYRET
      JMP  TOISR              ;定时器 T0 中断入口地址
      JMP  ONLYRET
      JMP  PBCISR            ;PBC 中断入口地址

PBCISR:
      STA  ACCBK, 00H
      JMP  TORET

TOISR:
      STA  ACCBK, 00H      ;备份 ACC 寄存器
      LDI  WDT, 08H
      ADIM  TOSUM, 01H      ;定时器溢出寄存器加 1

TORET:
      LDI  IRQ, 00H      ;清除中断请求标志
      LDI  IE, IETO      ;允许定时器溢出中断
      LDA  ACCBK, 00H      ;恢复 ACC 寄存器的值

ONLYRET:
      RTNI              ;退出中断子程序

RESET:
      NOP              ;上电稳定空跑 4 条指令

```

```

NOP
NOP
NOP
LDI    WDT, 08H           ; 清除 watchdog 计数
LDI    IRQ, 00H           ;清除中断请求标志
LDI    IE, 00H            ;不允许中断

LDI    REM, 0H            ;初始化设置端口
LDI    PACR, 0FH
LDI    PBCR, 0FH
LDI    PCCR, 0FH
LDI    PDCR, 06H
LDI    PECR, 0FH
LDI    PA, 00H
LDI    PB, 00H
LDI    PC, 00H
LDI    PD, 00H
LDI    PE, 00H
LDI    PMOD, 0101B        ;不允许下拉电阻

CLEARRAM:                 ;清 0  RAM 020H-4FH 区
LDI    DPL, 00H
LDI    DPM, 02H
LDI    DPH, 00H
?CLRINX:
LDI    INX, 00H
ADIM    DPL, 01H
BNC     ?CLRINX
ADIM    DPM, 01H
ANDIM   DPM, 07H
SBI     DPM, 05H
BNC     ?CLRINX

ADSAMPLE:
LDI    ADPORTCR, 1111B    ;设定采样端为输出
LDI    ADPORT, 0000B      ;输出低电平放电

LDI    WDT, 08H           ;清除 watchdog 计数

```


LDI	PACR, 1111B	;将电容 C 彻底放电
LDI	PA, 00H	
CALL	PARA10MS	
LDA	ADPORT, 00H	;检测是否放电完毕，没有继续放电
BA2	\$-3	
LDI	ADPORTCR, 1011B	;设定采样端为输入
LDI	ADPORT, 0000B	;输入端拉低
LDI	TOSUM, 00H	;清 0 定时器溢出计数器
LDI	WDT, 08H	;清除 watchdog 计数
LDI	PACR, 0010B	;将参考电阻端为输出，其它端口输入
LDI	PA, 0010B	;参考电阻端输出高电平充电
LDI	TOM, SCALE1	;定时器预分频设为 1us
LDI	TOL, 00H	;定时器清 0
LDI	TOH, 00H	
LDI	IE, IET0	;允许定时器中断
NOP		
LDA	ADPORT, 00H	;检测是否充电完毕
BA2	\$+2	;充电完就再检测一次确认
JMP	\$-3	;没有就继续充电
NOP		
LDA	ADPORT, 00H	;检测是否充电完毕
BA2	\$+2	;充电完毕就读取参数
JMP	\$-7	;没有就继续充电
LDA	TOH, 00H	;保存充电参数
STA	REFER_TOH, 00H	
LDA	TOL, 00H	
STA	REFER_TOL, 00H	
ADDM	REFER_SUML, 00H	
LDA	REFER_TOH, 00H	
ADCM	REFER_SUMM, 00H	
LDA	TOSUM, 00H	
STA	REFER_TOSUM, 00H	
ADCM	REFER_SUMH, 00H	
LDI	TOSUM, 00H	
LDI	ADPORTCR, 1111B	

LDI	ADPORT, 0000B	
LDI	WDT, 08H	
LDI	PACR, 1111B	
LDI	PA, 00H	
CALL	PARA10MS	
LDA	ADPORT, 00H	
BA2	\$-3	
LDI	ADPORTCR, 1011B	
LDI	ADPORT, 0000B	
LDI	TOSUM, 00H	
LDI	WDT, 08H	
LDI	PACR, 0001B	
LDI	PA, 0001B	
LDI	TOL, 00H	
LDI	TOH, 00H	
NOP		
LDA	ADPORT, 00H	
BA2	+\$2	
JMP	\$-3	
NOP		
LDA	ADPORT	
BA2	+\$2	
JMP	\$-7	
LDA	TOH, 00H	
STA	TEST_TOH, 00H	
LDA	TOL, 00H	
STA	TEST_TOL, 00H	
ADDM	TEST_SUML, 00H	
LDA	TEST_TOH, 00H	
ADCM	TEST_SUMM, 00H	
LDA	TOSUM, 00H	
STA	TEST_TOSUM, 00H	
ADCM	TEST_SUMH, 00H	
ADIM	ADTIME, 01H	;AD 采样技术其加 1
SBI	ADTIME, 01H	;判断是否是第一次采样

BAZ	ADSAMPLE	;是第一次就不求平均值
SBI	ADTIME, 15	;是否已经采样了 15 次
BC	TEST_OK	;是就退出采样
LDI	DPL, REFER_SUML & 0FH	;求参考电阻充电时间平均值
LDI	DPM, (REFER_SUML >> 4) & 0FH	
LDI	PARA_BYTE, 03H	
LDI	PARA_BIT, 01H	
CALL	SHRC_N	
LDI	DPL, TEST_SUML & 0FH	;求测试电阻充电时间平均值
LDI	DPM, (TEST_SUML >> 4) & 0FH	
LDI	PARA_BYTE, 03H	
LDI	PARA_BIT, 01H	
CALL	SHRC_N	
JMP	ADSAMPLE	;继续下一次采样

TEST_OK:

LDA	REFER_SUML, 00H	;用除法求取测试电阻同参考电阻的比值
STA	DIVL, 00H	
LDA	REFER_SUMM, 00H	
STA	DIVM, 00H	
LDA	REFER_SUMH, 00H	
STA	DIVH, 00H	
LDA	TEST_SUML, 00H	
STA	BDIVL, 00H	
LDA	TEST_SUMM, 00H	
STA	BDIVM, 00H	
LDA	TEST_SUMH, 00H	
STA	BDIVH, 00H	
CALL	DIV	

TOHALT:

LDI	T0M, SCALE2048	; 定时器预分频设为 2ms
LDI	T0L, 06H	;设定每 500ms 采样一次
LDI	T0H, 00H	
LDI	WDT, 08H	; 清除 watchdog 计数
LDI	IRQ, 00H	
LDI	IE, IET0	

BNC	\$+2	
LDI	TEMPSUB, 08H	;如果进位 C=1， ztempsub=8
SBIM	DPL, 01H	;返回上一寄存器
EOR	DPL, 00H	
SBCM	DPM, 00H	
LDA	TEMPSUB, 00H	;将高一字节的低位加入
ADDM	INX, 00H	
SBIM	PARA_BYTE, 01H	;移位字节数减 1
BAZ	?SHRNEXT	;结果=0 就转到移位下一位
ADIM	DPL, 02H	
EOR	DPL, 00H	
ADCM	DPM, 00H	
JMP	?SHRLOOP	
?SHRNEXT:		
SBIM	PARA_BIT, 01H	;移位数减 1
BAZ	SHFRET	;结果=0 就退出函数
LDA	DPL_BK, 00H	;恢复要移位寄存器的地址
STA	DPL, 00H	
LDA	DPM_BK, 00H	
STA	DPM, 00H	
LDA	PARA_BYTE_BK, 00H	;恢复要连续移位的字节数
STA	PARA_BYTE, 00H	
JMP	?SHRFIRST	
SHFRET:		
RTNI		
PARA10MS:		;延时 10ms 参数
LDI	DELAYL, 02H	
LDI	DELAYM, 0EH	
LDI	ELAYH, 04H	
TIME_DELAY:		;调用函数延时 7X+2 条指令周期
LDI	WDT, 08H	; 清除 watchdog 计数
SBIM	DELAYL, 01H	;
EOR	DELAYL, 00H	;
SBCM	DELAYM, 00H	;
EOR	DELAYM, 00H	;
SBCM	DELAYH, 00H	;
BC	TIME_DELAY	;

```

RTNI                                     ;

;=====

;FunctionName:DIV

;Input:

;Output:RESULT_PL, RESULTL, RESULTM, RESULTH

;Temporary:

;CalledSubroutine:

;Description:除法子程序，精确到一个小数字

;=====

DIV:
    LDI        WDT, 08H                ;CLEAR WATCHDOG TIMER
    LDI        RESULTL, 00H
    LDI        RESULTM, 00H
    LDI        RESULTH, 00H
    LDI        BDIV_PL, 00H
    LDI        RESULT_PL, 00H
?DSUB:
    LDA        DIVL, 00H
    SUBM       BDIV_PL, 00H
    LDA        DIVM, 00H
    SBCM       BDIVL, 00H
    LDA        DIVH, 00H
    SBCM       BDIVM, 00H
    EOR        BDIVM, 00H
    SBCM       BDIVH, 00H
    BNC        DIV_RET
    ADIM       RESULT_PL, 01H
    EOR        RESULT_PL, 00H
    ADCM       RESULTL, 00H
    EOR        RESULTL, 00H
    ADCM       RESULTM, 00H
    EOR        RESULTM, 00H
    ADCM       RESULTH, 00H
    JMP        ?DSUB
DIV_RET:
    RTNI

END

```