

## 摘要

在电子技术中，频率是最基本的参数之一，并且与许多电参量的测量方案、测量结果都有十分密切的关系，因此，频率的测量就显得更为重要。测量频率的方法有多种，其中电子计数器测量频率具有精度高、使用方便、测量迅速，以及便于实现测量过程自动化等优点，是频率测量的重要手段之一。数字式频率计的测量原理有两类：一是直接测频法，即在一定闸门时间内测量被测信号的脉冲个数；二是间接测频法即测周期法，如周期测频法。直接测频法适用于高频信号的频率测量，通常采用计数器、数据锁存器及控制电路实现，并通过改变计数器闸门的时间长短在达到不同的测量精度；间接测频法适用于低频信号的频率测量，本设计中使用的就是直接测频法，即用计数器在计算1S内输入信号周期的个数。

数字频率计是数字电路中的一个典型应用，实际的硬件设计用到的器件较多，连线比较复杂，而且会产生比较大的延时，造成测量误差、可靠性差。随着现场可编程单片机的广泛应用，运用单片机控制，将使整个系统大大简化，提高了系统的整体性能和可靠性。

关键词：频率测量；单片机；数据处理

# 目录

摘要 .....	1.....
目录 .....	2.....
一 系统概述.....	3.....
1.系统组成.....	3.....
2.信号处理方法.....	4.....
二 器件简介.....	6.....
1.主控制器 AT89S52 .....	6.....
(1) MSC-51 芯片资源简介.....	错误!未定义书签.....
(2)单片机的引脚.....	错误!未定义书签.....
(3)89S51单机的电源线.....	错误!未定义书签.....
(4)89S51单片机的外接晶体引脚.....	错误!未定义书签.....
(5)89S51单片机的控制线.....	错误!未定义书签.....
(6)89S51单片机复位方式.....	错误!未定义书签.....
2.数码管显示器简介.....	6.....
(1) 数码管的分类.....	错误!未定义书签.....
(2) 数码管的驱动方式.....	错误!未定义书签.....
(3) 数码管参数.....	错误!未定义书签.....
(4) 数码管应用.....	错误!未定义书签.....
(5) 数码管使用的电流与电压.....	错误!未定义书签.....
(6) 恒流驱动与非恒流驱动对数码管的影响.....	错误!未定义书签.....
(7)怎样测量数码管引脚 .....	错误!未定义书签.....
3.元件设计软件 CPLD .....	错误!未定义书签.....
(1) CPLD 简介 .....	错误!未定义书签.....
(2) CPLD 器件特点.....	错误!未定义书签.....
三 系统设计.....	24.....
1.硬件设计.....	错误!未定义书签.....
(1) 信号予处理电路.....	错误!未定义书签.....
(2) 单片机系统.....	错误!未定义书签.....
(3) 数据显示电路.....	错误!未定义书签.....
(4) VHDL 实现 74LS160 功能 .....	错误!未定义书签.....
2.系统软件设计.....	24.....

(1)数据处理过程.....	24.....
(2)系统软件框图.....	25.....
(3)浮点数学运算程序 .....	25.....
(4) 实测结果和误差分析.....	26.....
四 参考文献.....	28.....
附件 1: 程序清单.....	29.....

# 一 系统概述

## 1. 系统组成

频率计由单片机89C51 、信号予处理电路、串行通信电路、测量数据显示电路和系统软件所组成, 其中信号予处理电路包含待测信号放大、波形变换、波形整形和分频电路。系统硬件框图如图1 所示。信号予处理电路中的放大器实现对待测信号的放大, 降低对待测信号的幅度要求; 波形变换和波形整形电路实现把正弦波样的正负交替的信号波形变换成可被单片机接受的TTL/ CMOS 兼容信号; 分频电路用于扩展单片机的频率测量范围并实现单片机频率测量和周期测量使用统一的输入信号。

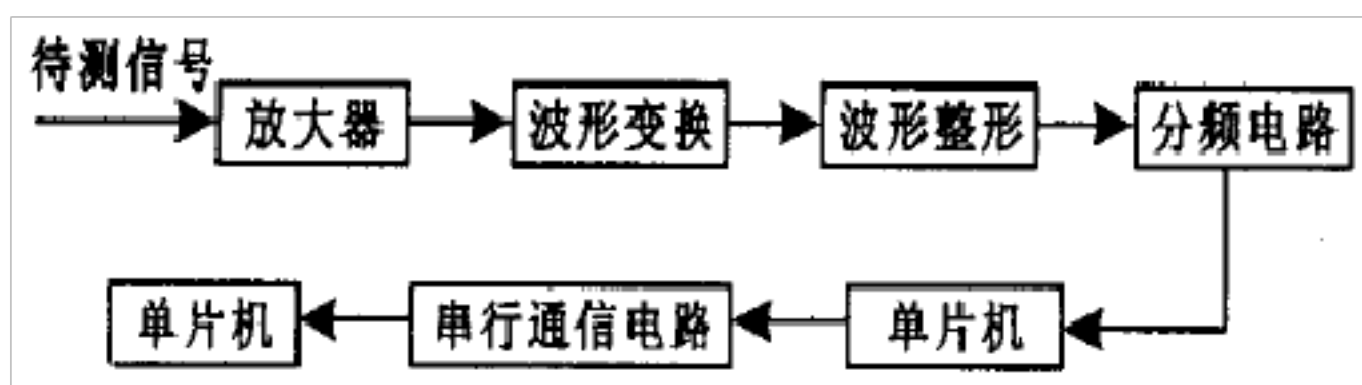


图1 系统硬件框图

系统软件包括测量初始化模块、显示模块、信号频率测量模块、量程自动转换模块、信号周期测量模块、定时器中断服务模块、浮点数格式化模块、浮点数算术运算模块、浮点数到BCD 码转换模块。系统软件框图如图2 所示。

## 2. 信号处理方法

本频率计的设计以AT89C51 单片机为核心, 利用它内部的定时/ 计数器完成待测信号周期/ 频率的测量。单片机AT89C51 内部具有2 个16 位定时/计数器, 定时/计数器的工作可以由编程来实现定时、计数和产生计数溢出中断要求的功能。在构成为定时器时, 每个机器周期加1 (使用12MHz 时钟时, 每1 $\mu$ s 加1) , 这样以机器周期为基准可以用来测量时间间隔。在构成为计数器时, 在相应的外部引脚发生从1 到0 的跳变时计数器加1 , 这样在计数闸门的控制下可以用来测量待测信号的频率。外部输入每个机器周期被采样一次, 这样检测一次从1 到0 的跳变至少需要2 个机器周期(24 个振荡周期) , 所以最大计数速率为时钟频率的1/ 24 (使用12MHz 时钟时, 最大计数速率为500 KHz) 。定时/计数器的工作由相应的运行控制位TR 控制, 当TR置1 , 定时/ 计数器开始计数;当TR 清0 , 停止计数。设计综合考虑了频率测量精度和测量反应时间的要求。例如当要求频率测量结果为 $N$ 位有效数字, 这时如果待测信号的频率为1Hz , 则计数闸门宽度必须大于 $1000s$ 。为了兼顾频率测量精度和测量反应时间的要求, 把测量工作分为两种方法。当待测信号的频率大于100Hz 时, 定时/ 计数器构成为计数器, 以机器周期为基准, 由软件产生计数闸门, 这时要满足频率测量结果为3 位有效数字, 则计数闸门宽度大于1s 即可。当待测信号的频率小于100Hz 时, 定时/ 计数器构成为定时器, 由频率计的预处理电路把待测信号变成方波, 方波宽度等于待测信号的周期。用方波作计数闸门, 当待测信号的频率等于100Hz , 使用12MHz 时钟时的最小计数值为10000 , 完全满足测量精度的要求。

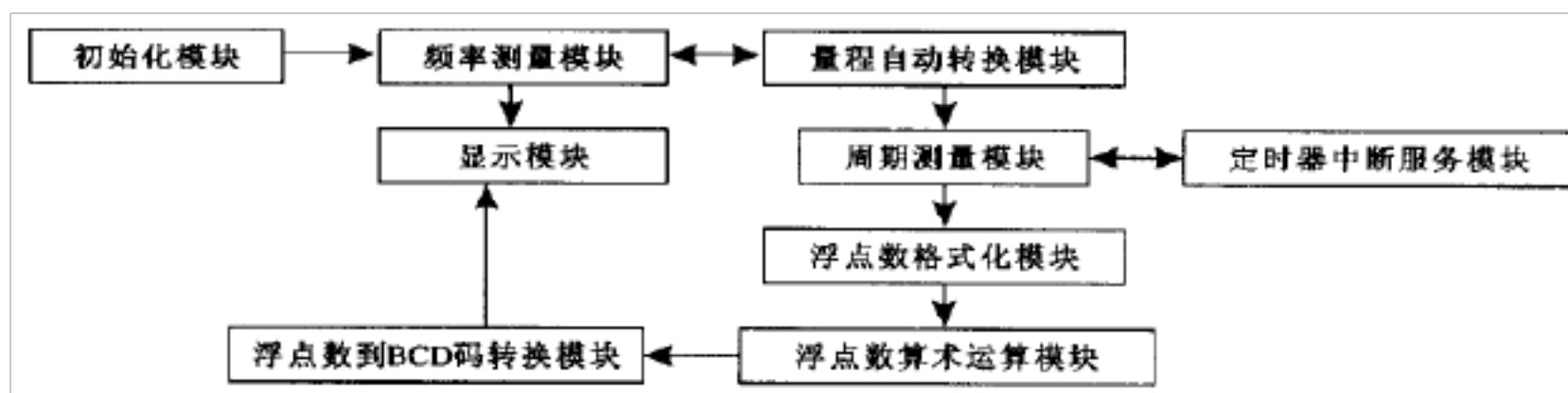


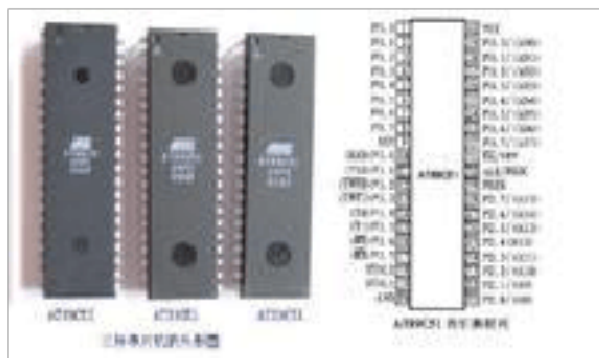
图2 系统软件框图

### 1. 3 频率计的量程自动切换

在使用计数方法实现频率测量时,这时外部的待测信号为定时/计数器的计数源,利用软件延时程序实现计数闸门。频率计的工作过程为:首先定时/计数器的计数寄存器清0,运行控制位TR置1,启动定时/计数器;然后运行软件延时程序,同时定时/计数器对外部的待测信号进行计数,延时结束时TR清0,停止计数;最后从计数寄存器读出测量数据,在完成数据处理后,由显示电路显示测量结果。在使用定时方法实现频率测量时,这时外部的待测信号通过频率计的预处理电路变成宽度等于待测信号周期的方波,该方波同样加至定时/计数器的输入脚。这时频率计的工作过程为:首先定时/计数器的计数寄存器清0,然后检测方波高电平是否加至定时/计数器的输入脚;当判定高电平加至定时/计数器的输入脚,运行控制位TR置1,启动定时/计数器对单片机的机器周期的计数,同时检测方波高电平是否结束;当判定高电平结束时TR清0,停止计数,然后从计数寄存器读出测量数据,在完成数据处理后,由显示电路显示测量结果。测量结果的显示格式采用科学计数法,即有效数字乘以10为底的幂。这里设计的频率计用5位数码管显示测量结果:前3位为测量结果的有效数字;第4位为指数的符号;第5位为指数的值。采用这种显示格式既保证了测量结果的显示精度,又保证了测量结果的显示范围(0.100Hz - 9.99MHz)。量程自动转换的过程由频率计测量量程的高端开始。由于只显示3位有效数字,在测量量程的高端计数闸门不需要太宽,例如在10.0 KHz - 9919 KHz 频率范围,计数闸门宽度为10mS即可。频率计每个工作循环开始时使用计数方法实现频率测量,并使计数闸门宽度为最窄,完成测量后判断测量结果是否具有3位有效数字,如果成立,将结果送去显示,本工作循环结束;否则将计数闸门宽度扩大10倍,继续进行测量判断,直到计数闸门宽度达到1s,这时对应的频率测量范围为100Hz - 999Hz。如果测量结果仍不具有3位有效数字,频率计则使用定时方法实现频率测量。定时方法测量的是待测信号的周期,这种方法只设一种量程,测量结果通过浮点数运算模块将信号周期转换成对应的频率值,再将结果送去显示。这样无论采用何种方式,只要完成一次测量即可,频率计自动开始下一个测量循环,因此该频率计具有连续测量的功能,同时实现量程的自动转换。

## 二 器件简介

### 1.主控制器 AT89S52



#### AT89C51 简介

AT89C51 是一种带 4K 字节闪烁可编程可擦除 [只读存储器](#) (FPEROM—Flash Programmable and Erasable Read Only Memory) 的低电压, 高性能 CMOS 8 位微处理器, 俗称 [单片机](#)。AT89C2051 是一种带 2K 字节闪烁可编程可擦除只读存储器的单片机。单片机的可擦除只读存储器可以反复擦除 100 次。该器件采用 ATMEL 高密度非易失存储器制造技术制造, 与工业标准的 MCS-51 指令集和输出管脚相兼容。由于将多功能 8 位 CPU 和闪烁存储器组合在单个芯片中, ATMEL 的 AT89C51 是一种高效微控制器, AT89C2051 是它的一种精简版本。AT89C 单片机为很多嵌入式控制系统提供了一种灵活性高且价廉的方案。外形及引脚排列如图所示

主要特性：

- 与 MCS-51 兼容
- 4K 字节可编程闪烁存储器
- 寿命：1000 写/擦循环
- 数据保留时间：10 年
- 全静态工作：0Hz-24MHz
- 三级程序存储器锁定
- 128×8 位内部 RAM
- 32 可编程 I/O 线
- 两个 16 位定时器/计数器
- 5 个中断源
- 可编程串行通道
- 低功耗的闲置和掉电模式
- 片内振荡器和时钟电路

管脚说明：

VCC：供电电压。

GND：接地。

P0 口：P0 口为一个 8 位漏级开路双向 I/O 口，每脚可吸收 8TTL 门电流。当 P1 口的管脚第一次写 1 时，被定义为高阻输入。P0 能够用于外部程序数据存储器，它可以被定义为数据/地址的第八位。在 FLASH 编程时，P0 口作为原码输入口，当 FLASH 进行校验时，P0 输出原码，此时 P0 外部必须被拉高。

P1 口：P1 口是一个内部提供上拉电阻的 8 位双向 I/O 口，P1 口缓冲器能接收输出 4TTL 门电流。P1 口管脚写入 1 后，被内部上拉为高，可用作输入，P1 口被外部下拉为低电平时，将输出电流，这是由于内部上拉的缘故。在 FLASH 编程和校验时，P1 口作为第八位地址接收。

P2 口：P2 口为一个内部上拉电阻的 8 位双向 I/O 口，P2 口缓冲器可接收，输出 4 个 TTL 门电流，当 P2 口被写“1”时，其管脚被内部上拉电阻拉

高，且作为输入。并因此作为输入时，P2 口的管脚被外部拉低，将输出电流。这是由于内部上拉的缘故。P2 口当用于外部程序存储器或 16 位地址外部数据存储器进行存取时，P2 口输出地址的高八位。在给出地址“1”时，它利用内部上拉优势，当对外部八位地址数据存储器进行读写时，P2 口输出其特殊功能寄存器的内容。P2 口在 FLASH 编程和校验时接收高八位地址信号和控制信号。

P3 口：P3 口管脚是 8 个带内部上拉电阻的双向 I/O 口，可接收输出 4 个 TTL 门电流。当 P3 口写入“1”后，它们被内部上拉为高电平，并用作输入。作为输入，由于外部下拉为低电平，P3 口将输出电流（ILL）这是由于上拉的缘故。

P3 口也可作为 AT89C51 的一些特殊功能口，如下表所示：

口管脚 备选功能

P3.0 RXD（串行输入口）

P3.1 TXD（串行输出口）

P3.2 /INT0（外部中断 0）

P3.3 /INT1（外部中断 1）

P3.4 T0（记时器 0 外部输入）

P3.5 T1（记时器 1 外部输入）

P3.6 /WR（外部数据存储器写选通）

P3.7 /RD（外部数据存储器读选通）

P3 口同时为闪烁编程和编程校验接收一些控制信号。

RST：复位输入。当振荡器复位器件时，要保持 RST 脚两个机器周期的高电平时间。

ALE/PROG：当访问外部存储器时，地址锁存允许的输出电平用于锁存地址的地位字节。在 FLASH 编程期间，此引脚用于输入编程脉冲。在平时，ALE 端以不变的频率周期输出正脉冲信号，此频率为振荡器频率的 1/6。因此它可用作对外部输出的脉冲或用于定时目的。然而要注意的是：每当用作外部数据存储器时，将跳过一个 ALE 脉冲。如想禁止 ALE 的输出可在 SFR8EH 地址上置 0。此时，ALE 只有在执行 MOVX，MOVC 指令是 ALE 才起作用。另外，



该引脚被略微拉高。如果微处理器在外部执行状态 ALE 禁止，置位无效。

/PSEN：外部程序存储器的选通信号。在由外部程序存储器取指期间，每个机器周期两次/PSEN 有效。但在访问外部数据存储器时，这两次有效的/PSEN 信号将不出现。

/EA/VPP：当/EA 保持低电平时，则在此期间外部程序存储器（0000H-FFFFH），不管是否有内部程序存储器。注意加密方式 1 时，/EA 将内部锁定为 RESET；当/EA 端保持高电平时，此间内部程序存储器。在 FLASH 编程期间，此引脚也用于施加 12V 编程电源（VPP）。

XTAL1：反向振荡放大器的输入及内部时钟工作电路的输入。

XTAL2：来自反向振荡器的输出。

振荡器特性：

XTAL1 和 XTAL2 分别为反向放大器的输入和输出。该反向放大器可以配置为片内振荡器。石晶振荡和陶瓷振荡均可采用。如采用外部时钟源驱动器件，XTAL2 应不接。有余输入至内部时钟信号要通过一个二分频触发器，因此对外部时钟信号的脉宽无任何要求，但必须保证脉冲的高低电平要求的宽度。

芯片擦除：

整个 PEROM 阵列和三个锁定位的电擦除可通过正确的控制信号组合，并保持 ALE 管脚处于低电平 10ms 来完成。在芯片擦操作中，代码阵列全被写“1”且在任何非空存储字节被重复编程以前，该操作必须被执行。

此外，AT89C51 设有稳态逻辑，可以在低到零频率的条件下静态逻辑，支持两种软件可选的掉电模式。在闲置模式下，CPU 停止工作。但 RAM，定时器，计数器，串口和中断系统仍在工作。在掉电模式下，保存 RAM 的内容并且冻结振荡器，禁止所用其他芯片功能，直到下一个硬件复位为止。

串口通讯

单片机的结构和特殊寄存器，这是你编写软件的关键。至于串口通信需要用到那些特殊功能寄存器呢，它们是 SCON，TCON，TMOD，SCON 等，各代表什么含义呢？

SBUF 数据缓冲寄存器这是一个可以直接寻址的串行口专用寄存器。有

朋友这样问起过“为何在串行口收发中,都只是使用到同一个寄存器 SBUF? 而不是收发各用一个寄存器。”实际上 SBUF 包含了两个独立的寄存器,一个是发送寄存,另一个是接收寄存器,但它们都共同使用同一个寻址地址—99H。CPU 在读 SBUF 时会指到接收寄存器,在写时会指到发送寄存器,而且接收寄存器是双缓冲寄存器,这样可以避免接收中断没有及时的被响应,数据没有被取走,下一帧数据已到来,而造成的数据重叠问题。发送器则不需要用到双缓冲,一般情况下我们在写发送程序时也不必用到发送中断去外理发送数据。操作 SBUF 寄存器的方法则很简单,只要把这个 99H 地址用关键字 sfr 定义为一个变量就可以对其进行读写操作了,如 `sfr SBUF = 0x99;`当然你也可以用其它的名称。通常在标准的 `reg51.h` 或 `at89x51.h` 等头文件中已对其做了定义,只要用 `#include` 引用就可以了。

SCON 串行口控制寄存器通常在芯片或设备中为了监视或控制接口状态,都会引用到接口控制寄存器。SCON 就是 51 芯片的串行口控制寄存器。它的寻址地址是 98H,是一个可以位寻址的寄存器,作用就是监视和控制 51 芯片串行口的工作状态。51 芯片的串口可以工作在几个不同的工作模式下,其工作模式的设置就是使用 SCON 寄存器。它的各个位的具体定义如下:

SM0 SM1 SM2 REN TB8 RB8 TI RI

SM0、SM1 为串行口工作模式设置位,这样两位可以对应进行四种模式的设置。串行口工作模式设置。

SM0 SM1 模式 功能 波特率

0 0 0 同步移位寄存器  $f_{osc}/12$

0 1 1 8 位 UART 可变

1 0 2 9 位 UART  $f_{osc}/32$  或  $f_{osc}/64$

1 1 3 9 位 UART 可变

在这里只说明最常用的模式 1, 其它的模式也就一一略过, 有兴趣的朋友可以找相关的硬件资料查看。表中的  $f_{osc}$  代表振荡器的频率, 也就是晶振的频率。UART 为 (Universal Asynchronous Receiver) 的英文缩写。

SM2 在模式 2、模式 3 中为多处理机通信使能位。在模式 0 中要求该位为 0。

REM 为允许接收位，REM 置 1 时串口允许接收，置 0 时禁止接收。REM 是由软件置位或清零。如果在一个电路中接收和发送引脚 P3.0, P3.1 都和上位机相连，在软件上有串口中断处理程序，当要求在处理某个子程序时不允许串口被上位机来的控制字符产生中断，那么可以在这个子程序的开始处加入 REM=0 来禁止接收，在子程序结束处加入 REM=1 再次打开串口接收。大家也可以用上面的实际源码加入 REM=0 来进行实验。

TB8 发送数据位 8，在模式 2 和 3 是要发送的第 9 位。该位可以用软件根据需要置位或清除，通常这位在通信协议中做奇偶位，在多处理机通信中这一位则用于表示是地址帧还是数据帧。

RB8 接收数据位 8，在模式 2 和 3 是已接收数据的第 9 位。该位可能是奇偶位，地址/数据标识位。在模式 0 中，RB8 为保留位没有被使用。在模式 1 中，当 SM2=0，RB8 是已接收数据的停止位。

TI 发送中断标识位。在模式 0，发送完第 8 位数据时，由硬件置位。其它模式中则是在发送停止位之初，由硬件置位。TI 置位后，申请中断，CPU 响应中断后，发送下一帧数据。在任何模式下，TI 都必须由软件来清除，也就是说在数据写入到 SBUF 后，硬件发送数据，中断响应（如中断打开），这时 TI=1，表明发送已完成，TI 不会由硬件清除，所以这时必须用软件对其清零。

RI 接收中断标识位。在模式 0，接收第 8 位结束时，由硬件置位。其它模式中则是在接收停止位的半中间，由硬件置位。RI=1，申请中断，要求 CPU 取走数据。但在模式 1 中，SM2=1 时，当未收到有效的停止位，则不会对 RI 置位。同样 RI 也必须要靠软件清除。常用的串口模式 1 是传输 10 个位的，1 位起始位为 0，8 位数据位，低位在先，1 位停止位为 1。它的波特率是可变的，其速率是取决于定时器 1 或定时器 2 的定时值（溢出速率）。AT89C51 和 AT89C2051 等 51 系列芯片只有两个定时器，定时器 0 和定时器 1，而定时器 2 是 89C52 系列芯片才有的。

波特率在使用串口做通讯时，一个很重要的参数就是波特率，只有上下位机的波特率一样时才可以进行正常通讯。波特率是指串行端口每秒内可以传输的波特位数。有一些初学的朋友认为波特率是指每秒传输的字节数，如

标准 9600 会被误认为每秒种可以传送 9600 个字节，而实际上它是指每秒可以传送 9600 个二进位，而一个字节要 8 个二进位，如用串口模式 1 来传输那么加上起始位和停止位，每个数据字节就要占用 10 个二进位，9600 波特率用模式 1 传输时，每秒传输的字节数是  $9600 \div 10 = 960$  字节。51 芯片的串口工作模式 0 的波特率是固定的，为  $f_{osc}/12$ ，以一个 12M 的晶振来计算，那么它的波特率可以达到 1M。模式 2 的波特率是固定在  $f_{osc}/64$  或  $f_{osc}/32$ ，具体用那一种就取决于 PCON 寄存器中的 SMOD 位，如 SMOD 为 0，波特率为  $f_{osc}/64$ ，SMOD 为 1，波特率为  $f_{osc}/32$ 。模式 1 和模式 3 的波特率是可变的，取决于定时器 1 或 2（52 芯片）的溢出速率。那么我们怎么去计算这两个模

式的波特率设置时相关的寄存器的值呢？可以用以下的公式去计算。

$$\text{波特率} = (2\text{SMOD} \div 32) \times \text{定时器 1 溢出速率}$$

上式中如设置了 PCON 寄存器中的 SMOD 位为 1 时就可以把波特率提升 2 倍。通常会使用定时器 1 工作在定时器工作模式 2 下，这时定时值中的 TL1 做为计数，TH1 做为自动重装值，这个定时模式下，定时器溢出后，TH1 的值会自动装载到 TL1，再次开始计数，这样可以不用软件去干预，使得定时更准确。在这个定时模式 2 下定时器 1 溢出速率的计算公式如下：

$$\text{溢出速率} = (\text{计数速率}) / (256 - \text{TH1})$$

上式中的“计数速率”与所使用的晶体振荡器频率有关，在 51 芯片中定时器启动后会在每一个机器周期使定时寄存器 TH 的值增加一，一个机器周期等于十二个振荡周期，所以可以得知 51 芯片的计数速率为晶体振荡器频率的 1/12，一个 12M 的晶振用在 51 芯片上，那么 51 的计数速率就为 1M。通常用 11.0592M 晶体是为了得到标准的无误差的波特率，那么为何呢？计算一下就知道了。如我们要得到 9600 的波特率，晶振为 11.0592M 和 12M，定时器 1 为模式 2，SMOD 设为 1，分别看看那所要求的 TH1 为何值。代入公式：

$$11.0592\text{M}$$

$$9600 = (2 \div 32) \times ((11.0592\text{M}/12) / (256 - \text{TH1}))$$

$$\text{TH1} = 250$$

12M

$$9600 = (2 \div 32) \times ((12M/12) / (256 - TH1))$$

$$TH1 \approx 249.49$$

上面的计算可以看出使用 12M 晶体的时候计算出来的 TH1 不为整数，而 TH1 的值只能取整数，这样它就会有一定的误差存在不能产生精确的 9600 波特率。当然一定的误差是可以在使用中被接受的，就算使用 11.0592M 的晶体振荡器也会因晶体本身所存在的误差使波特率产生误差，但晶体本身的误差对波特率的影响是十分之小的，可以忽略不计。

## DAC0832 芯片：

DAC0832 是 8 分辨率的 D/A 转换集成芯片。与微处理器完全兼容。这个 DA 芯片以其价格低廉、接口简单、转换控制容易等优点，在单片机应用系统中得到广泛的应用。D/A 转换器由 8 位输入锁存器、8 位 DAC 寄存器、8 位 D/A 转换电路及转换控制电路构成。

DAC0832 的主要特性参数如下：

- \* 分辨率为 8 位；
- \* 电流稳定时间 1 $\mu$ s；
- \* 可单缓冲、双缓冲或直接数字输入；
- \* 只需在满量程下调整其线性度；
- \* 单一电源供电（+5V~+15V）；
- \* 低功耗，200mW。

DAC0832 结构：

- \* D0~D7：8 位数据输入线，TTL 电平，有效时间应大于 90ns（否则锁存器的数据会出错）；

- \* ILE: 数据锁存允许控制信号输入线, 高电平有效;

- \* CS: 片选信号输入线 (选通数据锁存器), 低电平有效;

- \* WR1: 数据锁存器写选通输入线, 负脉冲 (脉宽应大于 500ns) 有效。

由 ILE、CS、WR1 的逻辑组合产生 LE1, 当 LE1 为高电平时, 数据锁存器状态随输入数据线变换, LE1 的负跳变时将输入数据锁存;

- \* XFER: 数据传输控制信号输入线, 低电平有效, 负脉冲 (脉宽应大于 500ns) 有效;

- \* WR2: DAC 寄存器选通输入线, 负脉冲 (脉宽应大于 500ns) 有效。由 WR1、XFER 的逻辑组合产生 LE2, 当 LE2 为高电平时, DAC 寄存器的输出随寄存器的输入而变化, LE2 的负跳变时将数据锁存器的内容打入 DAC 寄存器并开始 D/A 转换。

- \* IOUT1: 电流输出端 1, 其值随 DAC 寄存器的内容线性变化;

- \* IOUT2: 电流输出端 2, 其值与 IOUT1 值之和为一常数;

- \* Rfb: 反馈信号输入线, 改变 Rfb 端外接电阻值可调整转换满量程精度;

- \* Vcc: 电源输入端, Vcc 的范围为 +5V~+15V;

- \* VREF: 基准电压输入线, VREF 的范围为 -10V~+10V;

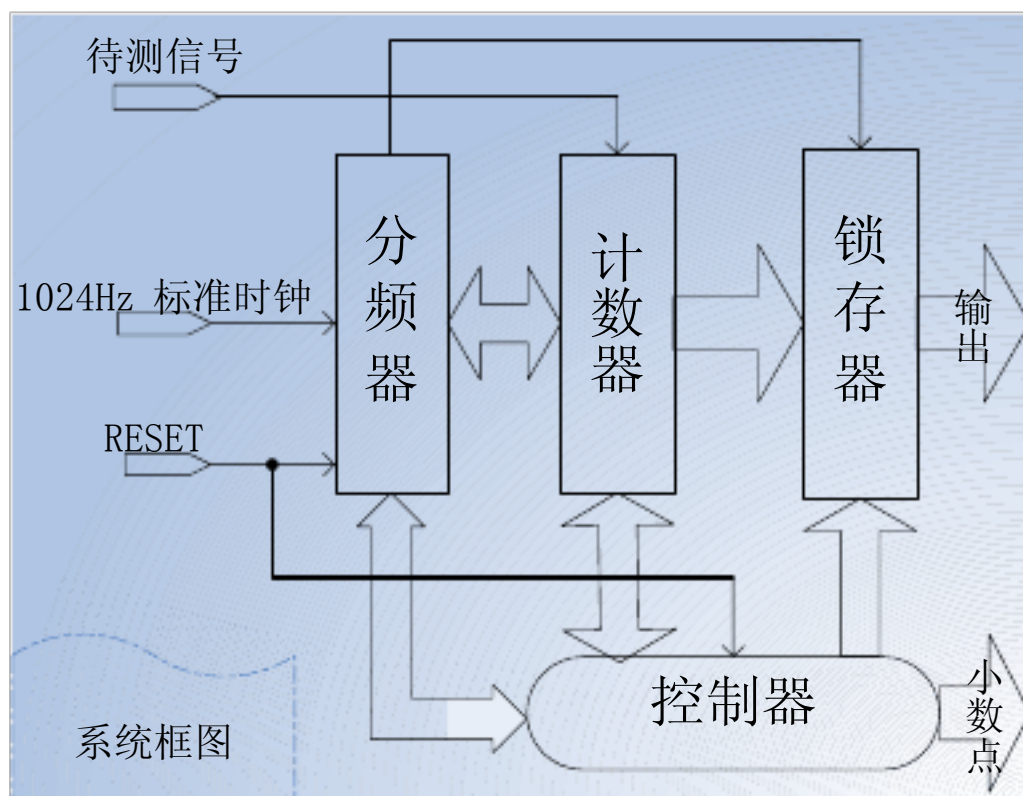
- \* AGND: 模拟信号地

- \* DGND: 数字信号地

DAC0832 的工作方式:

根据对 DAC0832 的数据锁存器和 DAC 寄存器的不同的控制方式, DAC0832 有三种工作方式: 直通方式、单缓冲方式和双缓冲方式。

## 二、系统总体设计



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/966200215110011004>