

# 基于 51 单片机的数字频率计

## 福星电子网

提供单片机学习板，开发板，最小系统板；超声波测距实验应用板，

各类器件仪表，详情请访问网站 <http://www.fxdzw.com>

题 目：基于 51 单片机的数字频率计

# 目 录

<b>第 1 节 引言</b>	<b>2</b>
1.1 数字频率计概述	2
1.2 频率测量仪的设计思路与频率的计算	2
1.3 基本设计原理	3
<b>第 2 节 数字频率计（低频）的硬件结构设计</b>	<b>4</b>
2.1 系统硬件的构成	4
2.2 系统工作原理图	4
2.3 AT89C51 单片机及其引脚说明	5
2.4 信号调理及放大整形模块	7
2.5 时基信号产生电路	7
2.6 显示模块	8
<b>第 3 节 软件设计</b>	<b>12</b>
3.1 定时计数	12
3.2 量程转换	12
3.3 BCD 转换	12
3.4 LCD 显示	12
<b>第 4 节 结束语</b>	<b>13</b>
<b>参考文献</b>	<b>14</b>
<b>附录 汇编源程序代码</b>	<b>15</b>

# 基于 51 单片机的数字频率计

## 第 1 节 引言

本应用系统设计的目的是通过在“单片机原理及应用”课堂上学习的知识，以及查阅资料，培养一种自学的能力。并且引导一种创新的思维，把学到的知识应用到日常生活当中。在设计的过程中，不断的学习，思考和同学间的相互讨论，运用科学的分析问题的方法解决遇到的困难，掌握单片机系统一般的开发流程，学会对常见问题的处理方法，积累设计系统的经验，充分发挥教学与实践的结合。全能提高个人系统开发的综合能力，开拓了思维，为今后能在相应工作岗位上的工作打下了坚实的基础。

### 1.1 数字频率计概述

数字频率计是计算机、通讯设备、音频视频等科研生产领域不可缺少的测量仪器。它是一种用十进制数字显示被测信号频率的数字测量仪器。它的基本功能是测量正弦信号，方波信号及其他各种单位时间内变化的物理量。在进行模拟、数字电路的设计、安装、调试过程中，由于其使用十进制数显示，测量迅速，精确度高，显示直观，经常要用到频率计。

本数字频率计将采用定时、计数的方法测量频率，采用一个 1602A LCD 显示器动态显示 6 位数。测量范围从 1Hz—10kHz 的正弦波、方波、三角波，时基宽度为 1 $\mu$ s, 10 $\mu$ s, 100 $\mu$ s, 1ms。用单片机实现自动测量功能。

基本设计原理是直接利用十进制数字显示被测信号频率的一种测量装置。它以测量周期的方法对正弦波、方波、三角波的频率进行自动的测量。

### 1.2 频率测量仪的设计思路与频率的计算

频率测量仪的设计思路主要是：对信号分频，测量一个或几个被测量信号周期中已知标准频率信号的周期个数，进而测量出该信号频率的大小，其原理如右图 1 所示。

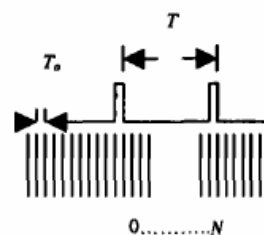


图 1 频率测量原理

若被测量信号的周期为 $T_x$ ，分频数 $m_1$ ，分频后信号的周期为 $T$ ，则： $T=m_1T_x$ 。由图可知： $T=NT_0$ 。

（注： $T_0$ 为标准信号的周期，所以 $T$ 为分频后信号的周期，则可以算出被测量信号的频率 $f_x$ 。）

由于单片机系统的标准频率比较稳定，而是系统标准信号频率的误差，通常情况下很小；而系统的量化误差小于1，所以由式 $T=NT_0$ 可知，频率测量的误差主要取决于 $N$ 值的大小， $N$ 值越大，误差越小，测量的精度越高。

### 1.3 基本设计原理

基本设计原理是直接利用十进制数字显示被测信号频率的一种测量装置。它以测量周期的方法对正弦波、方波、三角波的频率进行自动的测量。

所谓“频率”，就是周期性信号在单位时间（1s）内变化的次数。若在一定时间间隔 $T$ 内测得这个周期性信号的重复变化次数 $N$ ，则其频率可表示为 $f=N/T$ 。其中脉冲形成电路的作用是将被测信号变成脉冲信号，其重复频率等于被测频率 $f_x$ 。时间基准信号发生器提供标准的时间脉冲信号，若其周期为1s，则门控电路的输出信号持续时间亦准确地等于1s。闸门电路由标准秒信号进行控制，当秒信号来到时，闸门开通，被测脉冲信号通过闸门送到计数译码显示电路。秒信号结束时闸门关闭，计数器停止计数。由于计数器计得的脉冲数 $N$ 是在1秒时间内的累计数，所以被测频率 $f_x=N\text{Hz}$ 。

## 第2节 数字频率计（低频）的硬件结构设计

### 2.1 系统硬件的构成

本频率计的数据采集系统主要元器件是单片机 AT89C51，由它完成对待测信号频率的计数和结果显示等功能，外部还要有分频器、显示器等器件。可分为以下几个模块：放大整形模块、秒脉冲产生模块、换档模拟转换模块、单片机系统、LCD 显示模块。各模块关系图如图 2 所示：

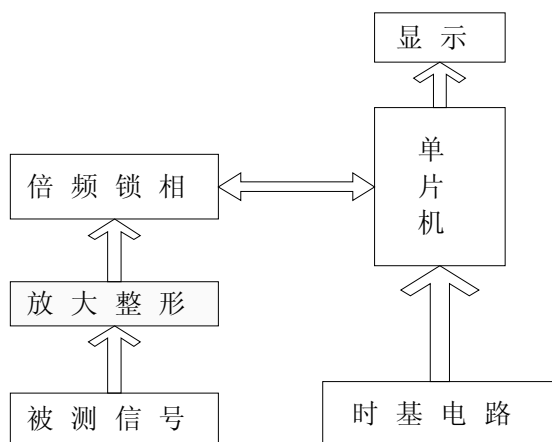


图 2 数字频率计功能模块

### 2.2 系统工作原理图

该系统工作的总原理图如图 3 所示：

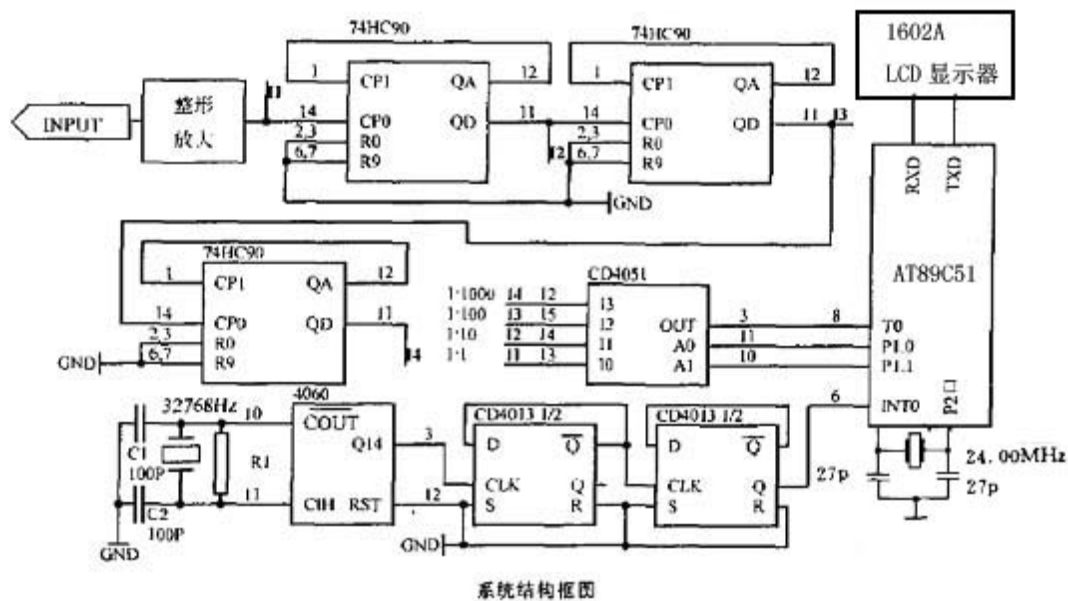


图 3 数字频率计系统工作原理图

## 2.3 AT89C51 单片机及其引脚说明

89C51 是一种高性能低功耗的采用 CMOS 工艺制造的 8 位微控制器，它提供下列标准特征：4K 字节的程序存储器，128 字节的 RAM,32 条 I/O 线，2 个 16 位定时器/计数器，一个 5 中断源两个优先级的中断结构，一个双工的串行口，片上震荡器和时钟电路。

引脚说明：

- $V_{CC}$ : 电源电压

- GND: 地

- P0 口：P0 口是一组 8 位漏极开路型双向 I/O 口，作为输出口用时，每个引脚能驱动 8 个 TTL 逻辑门电路。当对 0 端口写入 1 时，可以作为高阻抗输入端使用。

当 P0 口访问外部程序存储器或数据存储器时，它还可设定成地址数据总线复用的形式。在这种模式下，P0 口具有内部上拉电阻。

在 EPROM 编程时，P0 口接收指令字节，同时输出指令字节在程序校验时。程序校验时需要外接上拉电阻。

- P1 口：P1 口是一带有内部上拉电阻的 8 位双向 I/O 口。P1 口的输出缓冲能接受或输出 4 个 TTL 逻辑门电路。当对 P1 口写 1 时，它们被内部的上拉电阻拉升为高电平，此时可以作为输入端使用。当作为输入端使用时，P1 口因为内部存在上拉电阻，所以当外部被拉低时会输出一个低电流 ( $I_{IL}$ )。

- P2 口：P2 是一带有内部上拉电阻的 8 位双向的 I/O 端口。P2 口的输出缓冲能驱动 4 个 TTL 逻辑门电路。当向 P2 口写 1 时，通过内部上拉电阻把端口拉到高电平，此时可以用作输入口。作为输入口，因为内部存在上拉电阻，某个引脚被外部信号拉低时会输出电流 ( $I_{IL}$ )。

P2 口在访问外部程序存储器或 16 位地址的外部数据存储器（例如 MOVX @DPTR）时，P2 口送出高 8 位地址数据。在这种情况下，P2 口使用强大的内部上拉电阻功能当输出 1 时。当利用 8 位地址线访问外部数据存储器时（例 MOVX @R1），P2 口输出特殊功能寄存器的内容。

当 EPROM 编程或校验时，P2 口同时接收高 8 位地址和一些控制信号。

- P3 口：P3 是一带有内部上拉电阻的 8 位双向的 I/O 端口。P3 口的输出缓冲能驱动 4 个 TTL 逻辑门电路。当向 P3 口写 1 时，通过内部上拉电阻把端口拉到高电平，此时可以用作输入口。作为输入口，因为内部存在上拉电阻，某个引

脚被外部信号拉低时会输出电流 ( $I_{IL}$ )。

P3 口同时具有 AT89C51 的多种特殊功能，具体如下表 1 所示：

端口引脚	第二功能
P3.0	RXD (串行输入口)
P3.1	TXD (串行输出口)
P3.2	$\overline{INT0}$ (外部中断 0)
P3.3	$\overline{INT1}$ (外部中断 1)
P3.4	T0 (定时器 0)
P3.5	T1 (定时器 1)
P3.6	$\overline{WR}$ (外部数据存储器写选通)
P3.7	$\overline{RD}$ (外部数据存储器都选通)

表 1 P3 口的第二功能

- RST: 复位输入。当振荡器工作时，RST 引脚出现两个机器周期的高电平将使单片机复位。

- $\overline{ALE}/\overline{PROG}$ : 当访问外部存储器时，地址锁存允许是一输出脉冲，用以锁存地址的低 8 位字节。当在 Flash 编程时还可以作为编程脉冲输出 ( $\overline{PROG}$ )。

一般情况下，ALE 是以晶振频率的 1/6 输出，可以用作外部时钟或定时目的。但也要注意，每当访问外部数据存储器时将跳过一个 ALE 脉冲。

- $\overline{PSEN}$ : 程序存储允许时外部程序存储器的读选通信号。当 AT89C52 执行外部程序存储器的指令时，每个机器周期  $\overline{PSEN}$  两次有效，除了当访问外部数据存储器时， $\overline{PSEN}$  将跳过两个信号。

- $\overline{EA}/V_{PP}$ : 外部访问允许。为了使单片机能够有效的传送外部数据存储器从 0000H 到 FFFH 单元的指令， $\overline{EA}$  必须同 GND 相连接。需要主要的是，如果加密位 1 被编程，复位时 EA 端会自动内部锁存。

当执行内部编程指令时， $\overline{EA}$  应该接到  $V_{CC}$  端。

- XTAL1: 振荡器反相放大器以及内部时钟电路的输入端。

- XTAL2: 振荡器反相放大器的输出端。

在本次设计中，采用 89C51 作为 CPU 处理器，充分利用其硬件资源，结合 D 触发器 CD4013，分频器 CD4060，模拟转换开关 CD4051，计数器 74LS90 等数字处理芯片，主要控制两大硬件模块，量程切换以及显示模块。下面还将详细说明。



## 2.4 信号调理及放大整形模块

放大整形系统包括衰减器、跟随器、放大器、施密特触发器。它将正弦输入信号  $V_x$  整形为同频率方波  $V_o$ ，幅值过大的被测信号经过分压器分压送入后级放大器，以避免波形失真。由运算放大器构成的射级跟随器起阻抗变换作用，使输入阻抗提高。同相输入的运算放大器的放大倍数为  $(R1+R2)/R1$ ，改变  $R1$  的大小可以改变放大倍数。系统的整形电路由施密特触发器组成，整形后的方波送到闸门以便计数。

由于输入的信号幅度是不确定、可能很大也有可能很小，这样对于输入信号的测量就不方便了，过大可能会把器件烧毁，过小可能器件检测不到，所以在设计中采用了这个信号调理电路对输入的波形进行阻抗变换、放大限幅和整形，信号调理部分电路具体实现电路原理图和参数如下图 4 所示：

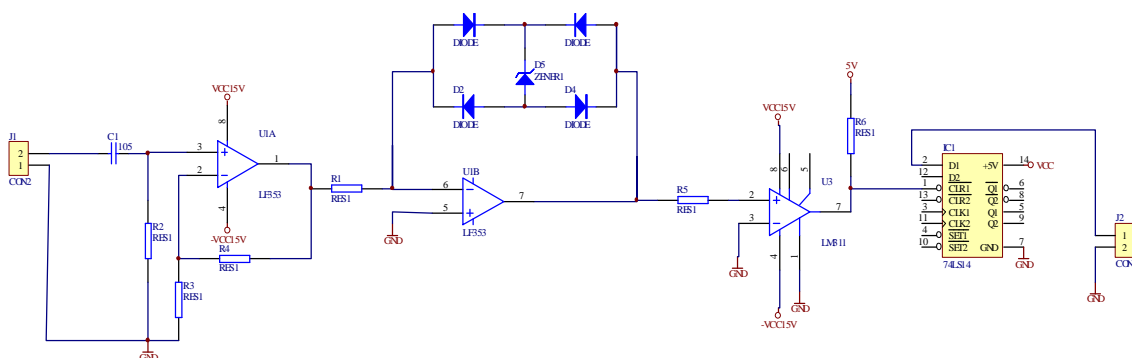


图 4

## 2.5 时基信号产生电路：

CD4013-----双上升沿 D 触发器 ， 引脚及功能见如下图 5：

CD4013 由两个相同的、相互独立的数据型触发器构成。每个触发器有独立的数据置位复位时钟输入和 Q 及 Q 非输出。此器件可用作移位寄存器，且通过将 Q 非输出连接到数据输入，可用作计数器和触发器。在时钟上升沿触发时，加在 D 输入端的逻辑电平传送到 Q 输出端。置位和复位或复位线上的高电平完成。

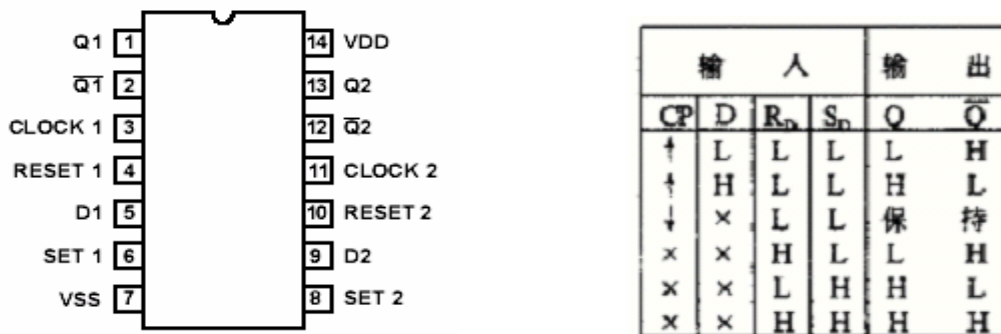


图 5 CD4013 芯片引脚用功能图

CD4060-----14 位二进制串行计数器，引脚及功能见如下图 6:

CD4060 由一振荡器和 14 极二进制串行计数器位组成，振荡器的结构可以是 RC 或晶振电路。CR 为高电平时，计数器清零且振荡器使用无效，所有的计数器位均为主从触发器 CP1 非（和 CP0）的下降沿计数器以二进制进行计数，在时钟脉冲线上使用施密特触发器对时钟上升和下降时间无限制。

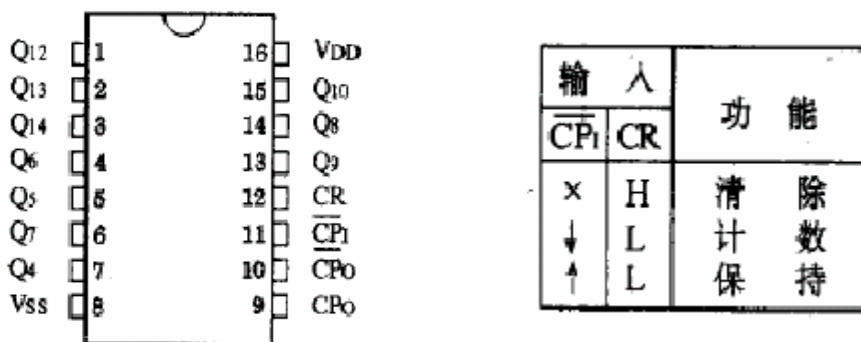
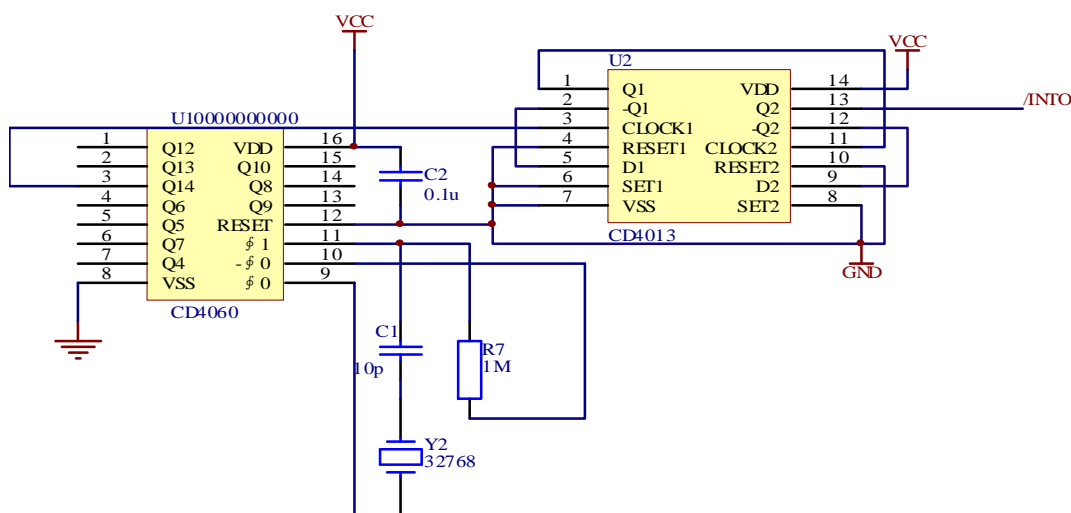


图 6 CD4060 芯片引脚用功能图

时基信号的产生原理:

本电路采用 32768HZ 晶体振荡器，利用 CD4060 芯片经过 14 级分频得到 2HZ 的信号 ( $32768/2^{14}$ )，在经过 CD4013 双 D 触发器经过二分频得到 0.5HZ 的方波，即输出秒脉冲信号使单片机进行计数。



图七 秒脉冲产生电路原理图

## 2.6 显示模块

1602 基本技术:

1)、主要功能

- A、 40 通道点阵 LCD 驱动;
- B、 可选择当作行驱动或列驱动;
- C、 输入/输出信号: 输出, 能产生  $20 \times 2$  个 LCD 驱动波形; 输入, 接受控制器送

出的串行数据和控制信号, 偏压(V1~V6);

D、通过单片机控制将所测的频率信号读数显示出来。

## 2)、技术参数

### 2.1) 极限参数表

名称	符号	标准值			单位
		MIN	TYPE	MAX	
电路电源	VDD - VSS	-0.3		7.0	V
LCD 驱动电压	VDD - VEE	VDD - 13.5		VDD + 0.3	V
输入电压	VIN	-0.3		VDD + 0.3	V
静电电压		-	-	100	V
工作温度		-20		+70	° C
储存温度		-30		+80	° C

### 2.2) 电参数表

名称	符号	测试条件	标准值			单位
			MIN	TYPE	MAX	
输入高电平	VIH	-	2.2		VDD	V
输入低电平	VIL	-	-0.3		0.6	V
输出高电平	VOH	IOH = 0.2mA	2.4		-	V
输出低电平	VOL	IOL = 1.2mA	-		0.4	V
工作电流	IDD	VDD = 5.0V		2.0		mA
液晶驱动电压	VDD-VEE	Ta = 0° C		4.9		V
		Ta = 25° C		4.7		
		Ta = 50° C		4.5		

## 3)、时序特性表

项目	符号	测试条件	标准值			单位
			MIN	TYPE	MAX	
允许时间周期	TCYCE	5.1a 5.1b	1000			ns
允许脉冲宽度, 高电平	PWEH		450	--	--	ns
允许上升和下降时间	tEr tEf		--	--	25	ns
地址建立时间	tAS		140	--	--	ns
数据延迟时间	tDDR		--	--	320	ns
数据建立时间	tDSW		195	--	--	ns
数据保持时间	tH		10	--	--	ns
DATA HOLD TIME	tDHR		20	--	--	ns
地址保持时间	tAH		10	--	--	ns

#### 4)、引脚和指令功能

##### 4.1) 模块引脚功能表

引线号	符号	名称	功能
1	Vss	接地	0V
2	VDD	电路电源	5V±10%
3	VEE	液晶驱动电压	保证 VDD-VEE=4.5~5V 电压差
4	RS	寄存器选择信号	H: 数据寄存器 L: 指令寄存器
5	R/W	读/写信号	H: 读 L: 写
6	E	片选信号	下降沿触发, 锁存数据
7   14	DB0   DB7	数据线	数据传输

##### 4.2) 寄存器选择功能表

RS	R/W	操作
0	0	指令寄存器(IR)写入
0	1	忙标志和地址计数器读出
1	0	数据寄存器(DR)写入
1	1	数据寄存器读出

(注: 忙标志为"1"时, 表明正在进行内部操作, 此时不能输入指令或数据, 要等内部操作结束, 即忙标志为"0"时。)

##### 4.3) 指令功能

格式: RS R/W DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

共 11 种指令: 清除, 返回, 输入方式设置, 显示开关, 控制, 移位, 功能设置, CGRAM 地址设置,

置, DDRAM 地址设置, 读忙标志, 写数据到 CG/DDRDRAM, 读数据由 CG/DDRDRAM。

#### 5)、显示位与 DD RAM 地址的对应关系

显示位序号		1	2	3	4	5	.....	40
DD RAM 地址(HEX)	第一行	00	01	02	03	04	..	27
	第二行	40	41	42	43	44	.....	67

#### 6)、初始化方法

用户所编的显示程序, 开始必须进行初始化, 否则模块无法正常显示, 下面介绍两种初始化方法;

##### 6.1 利用内部复位电路进行初始化

下面指令是在初始化过程中执行的。

(1)清屏(DISPLAY CLEAR);

(2)功能设置(FUNCTION SET);

DL = 1: 8Bit 接口数据;

N = 0: 1行显示; F = 0: 5×7dot 字形;

(3)显示开/关控制(DISPLAY ON/OFF CONTROL)

D = 0: 显示关; C = 0: 光标关; B = 0: 消隐关

(4)输入方式设置(ENTRY MODE SET )

I/D = 1: (增量):      S = 0: 无移位:

6.2) 软件复位

如果电路电源不能满足复位电路的要求的话,那么初始化就要用软件来实现,过程如下:

八位接口初始化流程图



## 第 3 节 软件设计

主要能通过编写软件来控制硬件完成以下各模块的功能：

### 3.1 定时读数

### 3.2 量程转换

### 3.3 BCD 转换

### 3.4 LCD 显示的功能

单片机当C/T=1时为计数方式，多路开关与定时器的外部引脚连通，外部计数脉冲由引脚输入。当外部信号由1至0跳变时，计数器加1，此时T0成为外部事件的计数器。由于确认一次由1至0的跳变要用24个振荡器周期，所以计数器的计数频率为单片机内部计数器频率的1/24。

当C / T=0时为定时方式，对单片机内部计数器进行m2分频后，计数器的实际计数频率为单片机内部频率凡的1/m2，

当GATE=0时，反相器输出为1，或门输出为1，打开与门，使定时器的启动仅受TR0端信号电平的控制。

在此种情况下，INT0引脚的电平变化对或门不起作用。TR0=1时接通控制开关，计数脉冲加到计数器上，每来一个计数脉冲，计数器加1，只有当TR0=0时，控制开关断开，计数器停止计数。

当GATA=0时，若TR0=1，或门、与门全部打开，外部信号电平通过INT0引脚直接控制定时器的启动和关闭。输入高电平时允许计数，否则停止计数。

根据定时器的结构原理，若我们将 GATE 位、TR0 均设为 '1'，INT0 端输入被测频率信号，当被测信号的高电平到来时，开始计数；当被测信号的低电平到来时，计数器停止计数，此时 TLO、TH0 的数据就是相应的 N 值。

## 结 束 语

数字频率计是计算机、通讯设备、音频视频等科研生产领域不可缺少的测量仪器。在进行模拟、数字电路的设计、安装、调试过程中，由于其使用十进制数显示，测量迅速，精确度高，显示直观，会被经常使用到。

通过本次课程的设计，不但加深我对在课程上所学到的单片机理论知识的认识和理解，重新让自己认识到了这门学科的在应用方面的广阔前景，并且通过知识与应用于实践的结合更加丰富了自己的知识。扩展了知识面，不但掌握了本专业的相关知识，而且对其他专业的知识也有所了解，而且较系统的掌握单片机应用系统的开发过程，因而自身的综合素质有了全面的提高。

经过这次一个较完整的产品设计和制作过程，对于认识到自己在知识方面存在的不足，明确今后的学习方向是非常有益的，为将来的的就业提前打下了坚实的基础。在设计过程中，得到了我的指导老师的悉心指导与帮助，还有其他老师和同学的大力支持和协助，在此一并表示衷心的感谢。

## 参考文献

- [1]李光飞 ， 楼苗然主编. 51系列单片机. 北京：北京航空航天大学出版社，2003
- [2]黄正瑾编著. CPLD系统设计技术入门与应用. 北京：电子工业出版社，2002
- [3]谢自美编著. 电子线路设计·实验·测试. 华中理工大学出版社，2002
- [4]陈永甫编著. 电子电路智能化设计. 实例与应用. 北京：电子工业出版社，2002. 8
- [5]康华光主编. 电子技术 基础(第四版). 北京：高等教育出版社，1999



## 附录 汇编源程序代码

```

RS    BIT    P2.0 ;P3.4 脚接 RS 端
      RW    BIT    P2.1 ;P3.5 脚接 R_W 端
      E    BIT    P2.2 ;P3.3 脚接 E 端
      ORG 0000H

      LJMP MAIN
      ORG 002BH
      LJMP IT1

MAIN:
      MOV    P0, #01H ;清除屏幕
      ACALL ENABLE
      MOV    P0, #01H ;清除屏幕
      ACALL ENABLE
      MOV    P0, #01H ;清除屏幕
      ACALL ENABLE
      MOV    P0, #38H ;8 位点阵方式
      ACALL ENABLE
      MOV    P0, #0cH ;开显示
      ACALL ENABLE
      MOV    P0, #06H ;移动光标
      ACALL ENABLE
      MOV    P0, #80H ;显示位置
      ACALL ENABLE
      mov    p0, #80h ;第一行的位置
      call  enable
      mov    dptr, #date
      call  write3
      mov    p0, #0c0h ;第二行的位置
      call  enable
      MOV SP, #7FH
      CLR CY
      mov r6, #00
      mov r2, #00
      MOV R3, #00
      MOV TMOD, #15H
      MOV TLO, #00H
      MOV TH0, #00H
      mov th1, #high(65536-50000)
      mov tl1, #low(65536-50000) ;setb ET1
cha1: SETB TR1 ;LCALL

```

XI ANSHI

```

SETB TR0
mov 27h, #00h
CLR C
MOV C, P1.6
MOV 27H.0, C
CLR C
MOV C, P1.7
MOV 27H.1, C
mov a, 27h
ANL A, #03H
MOV 17H, A
    
```

```

CHA: JBC TF1, JINWEI ;益处进位
      JMP CHA
    
```

JINWEI: CLR TR1

```

mov th1, #high(65536-50000)
mov tl1, #low(65536-50000)
SETB TR1
INC R3
MOV A, R3
CJNE A, #20, CHA
CLR TR0
CLR TR1
MOV R3, #00
MOV R2, TH0
MOV R6, TL0
I call zhuan
LCALL write1
MOV TL0, #00H
MOV TH0, #00H
jmp cha1
    
```

; JNB P3.2, \$

ZHUAN:

```

MOV A, R2
CLR C
MOV 20H, #00H
MOV 21H, #00H
MOV 22H, #00H
MOV 24H, #00H
MOV 25H, #00H
MOV R3, #10H
NEXT: RLC A
MOV R2, A
MOV A, 20H
ADDC A, 20H
    
```

```

DA A
MOV 20H, A
MOV A, 21H
ADDC A, 21H
DA A
MOV 21H, A
MOV A, 22H
ADDC A, 22H
DA A
MOV 22H, A
MOV A, R2
DJNZ R3, NEXT
MOV A, R6
clr c
MOV R3, #08H
NEXT1: RLC A
MOV R6, A
MOV A, 24H
ADDC A, 24H
DA A
MOV 24H, A
MOV A, 25H
ADDC A, 25H
DA A
MOV 25H, A
mov a, r6
DJNZ R3, NEXT1
clr c
mov a, 24h
addc a, 20h
da a
mov 20h, a
mov a, 25h
addc a, 21h
da a
mov 21h, a
mov a, 22h
addc a, #00h
da a
mov 22h, a
mov a, 22h
anl a, #0fh
mov 31h, a
mov a, 22h
    
```

```

    anl a, #0f0h
    swap a
    mov 30h, a
    mov a, 21h
    anl a, #0f0h
    swap a
    mov 32h, a
    mov a, 21h
    anl a, #0fh
    mov 33h, a
    mov a, 20h
    anl a, #0f0h
    swap a
    mov 34h, a
    mov a, 20h
    anl a, #0fh
    mov 35h, a
    ret
; *****
;
; LCD 显示
; *****
;

```

ENABLE:

```

    CLR    RS
    CLR    RW
    CLR    E
    ACALL  DELAY
    SETB   E
    RET

```

```

write1:                                ; 写数据
    MOV    10H, #00H
    MOV    11H, #00H
    MOV    12H, #00H
    MOV    13H, #00H
    MOV    14H, #00H
    MOV    15H, #00H
    MOV    16H, #00H
    mov    p0, #0ch
    call   enable
    mov    p0, #0c0h ; 第二行的位置
    call   enable
    MOV    A, 17H
    XRL   A, #02H

```

```
JNZ     XI AN1
MOV     40H, #00H
MOV     41H, #00H
MOV     42H, #00H
MOV     43H, 30H
MOV     44H, 31H
MOV     45H, 32H
JMP     XI AN

XI AN1: MOV     A, 17H
        XRL     A, #01H
        JNZ     XI AN2
        MOV     40H, #00H
        MOV     41H, #00H
        MOV     42H, 30H
        MOV     43H, 31H
        MOV     44H, 32H
        MOV     45H, 33H
        JMP     XI AN

XI AN2: MOV     A, 17H
        XRL     A, #00H
        JNZ     XI AN3
        MOV     40H, #00H
        MOV     41H, 30H
        MOV     42H, 31H
        MOV     43H, 32H
        MOV     44H, 33H
        MOV     45H, 34H
        JMP     XI AN

XI AN3: MOV     A, 17H
        XRL     A, #03H
        JNZ     XI AN
        MOV     40H, 30H
        MOV     41H, 31H
        MOV     42H, 32H
        MOV     43H, 33H
        MOV     44H, 34H
        MOV     45H, 35H
XI AN:  mov     r0, #40h
        mov     a,   @r0
        JZ      L20
        MOV     10H, #0FFH
```

```

MOV    15H, #0FFH
mov    dptr, #date1
movc   a, @a+dptr
call   write2

L20: INC    R0
      MOV    A, 10H
      CJNE   A, #00H, PP
      mov    a, @r0
      JZ     L21
PP:   MOV    11H, #0FFH
      MOV    15H, #0FFH
      mov    a, @r0
      mov    dptr, #date1
      movc   a, @a+dptr
      call   write2

L21: INC    R0
      MOV    A, 11H
      CJNE   A, #00H, PP2
      mov    a, @r0
      JZ     L22
PP2:  MOV    12H, #0FFH
      MOV    15H, #0FFH
      mov    a, @r0
      mov    dptr, #date1
      movc   a, @a+dptr
      call   write2
      MOV    A, 15H
      CJNE   A, #0FFH, L22
      MOV    A, #00H
      mov    dptr, #XIAOSHU DI AN
      movc   a, @a+dptr
      call   write2

L22:  INC    R0
      MOV    A, 12H
      CJNE   A, #00H, PP3
      mov    a, @r0
      JZ     L23
PP3:  MOV    13H, #0FFH
      mov    a, @r0
      mov    dptr, #date1

```

```

    movc    a,@a+dptr
    call    write2

L23: INC    R0
    MOV     A, 13H
    CJNE   A, #00H, PP4
    mov     a, @r0
    JZ     L24
PP4:  MOV     14H, #0FFH
    mov     a, @r0
    mov     dptr, #date1
    movc   a,@a+dptr
    call   write2

L24: INC    R0
    mov     a, @r0
    mov     dptr, #date1
    movc   a,@a+dptr
    call   write2
    MOV     A, 15H
    CJNE   A, #00H, KHZ
    MOV     A, #00H
    MOV     dptr, #DANWEI1
    JMP     QUSHU
KHZ: MOV     dptr, #DANWEI2
QUSHU: movc  a,@a+dptr
    call   write3
    mov     r5, #4
I3:  mov     a, #20h
    call   write2
    djnz   r5, I3
    ret

write2:
    mov     p0, a
    setb   rs
    CLR    rw
    clr    e
    call   delay
    setb   e
    ret

write3: mov     r1, #00h    ;写数据
a2:  mov     a, r1

```

```

    movc    a,    @a+dptr
    call   write2
    inc    r1
    cjne   a, #00h, a2
    ret

delay:
    mov    r7,    #255
d1:  mov    r6,    #255
d2:  djnz   r6,    d2
     djnz   r7,    d1
     ret

date:  db "the frequency is", 00h
date1: db 30h, 31h, 32h, 33h, 34h, 35h, 36h, 37h, 38h, 39h
DANWEI1: DB "HZ          ", 00h
DANWEI2: DB "KHZ         ", 00h
XIAOSHUDIAN: DB ". ", 00h

```