同济大学

基于 MSP430 的频率测量系统

课题名称	电子信息专业课程设计报告
学院(系)	电子与信息工程学院
专　　业	电子信息工程
年　　级	2020 级
姓　　名	黄锴康
学　　号	1951706
组　　号	1
指导教师	陈　耀
起讫时间	2023.7.3—2023.7.7
成　　绩	

目录

装 订 线

装 订 线

一、设计目的

随着微电子技术和计算机技术的发展，微控制器和嵌入式技术得到了迅猛的发展，它不仅广泛应用于仪器仪表、工业测控、通讯、军事等各个领域，并且已成为日常生活中无处不在的新技术。

本课程设计结合实际应用，基于模拟电路、数字电路、C 语言、计算机原理与技术、嵌入式系统和通信原理等多门专业课程，设计一个嵌入式综合应用系统，培养专业设计与应用能力，为后续创新实践、毕业设计等环节和以后的学习、工作打好坚实的基础。

通过设计理解电压、电阻、频率、温/湿度、距离等相关参数的测量原理、LED/LCD 显示、RS232 通信、蜂鸣器驱动等电路及工作原理，熟悉 MCU O、AD、D/A、定时器、串行通信等相关模块的高级应用及 C 语言程序设计和调试方法，掌握 MCU 集成开发环境、Proteus 虚拟仿真工具的使用，了解多机组网通信的基本原理。

本课程基于 MSP430 实验平台，在设计过程中需要理解实验平台的原理图，以及阅读相关相关芯片的 datasheet，掌握嵌入式系统的开发方法。

二、设计环境

本实验硬件环境为 PC,MSP430-FFTB6638。

本实验软件环境为 Code Composer Studio 6.2(CCS)集成开发环境，串口调试助手等。

三、设计内容

基于 MSP430F6638_FFTB 实验平台，设计频率测量潜入式系统，编写程序、进行调试、实现频率测量与通信系统，利用 MCU 定时器模块相关功能设计实现数字频率计功能，测量范围 100-10000Hz,测量误差≤1%,测量速度≤1 秒。测量结果本地显示(段式 LCD 和多位 7 段 LED 等)，同时通过 UART/RS232 通讯把测量结果传送到 PC 等其他终端，该项目实现以下功能：

- 项目名称、学号、姓名、测量结果等信息的本地显示(段式 LCD、多位 7 段 LED 等)。
- 项目名称、学号、姓名、测量结果等信息通过 UART/RS232 握手通讯(握手信号为本人序号或学号) 传送到 PC。
- 蜂鸣器声响提示项目功能的开始或不同蜂鸣器声响指示不同工作状态，如通信握手的正确与否。
- 通过串口命令实现频率与周期测量的模式转换。

四、设计方案及流程

4.1 总体设计方案

本系统的原理框图如图 4.1 所示，

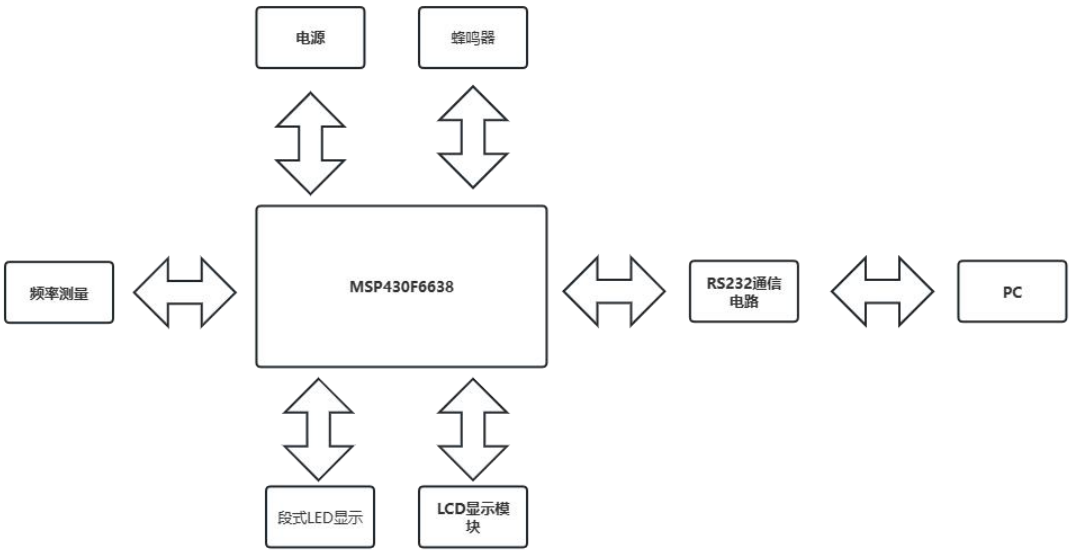

图 4.1 系统原理框图

结合原理框图，本项目主要采用多个中断完成不同任务，根据本项目的信号流以及不同的中断，系统设计如图 4.2 所示，

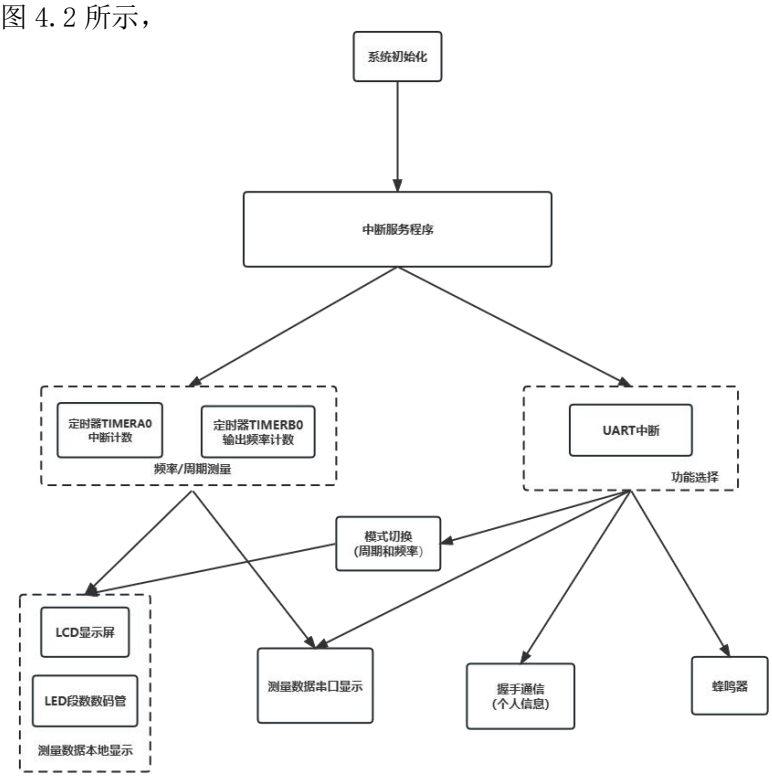

图 4.2 系统设计图

在系统中断、串口以及时钟初始化后，开启中断服务，TIMERA0 和 TIMERB0 负责频率测量，

UART 中断负责不同功能的选择，最后通过不同的外设硬件实现对应的功能。由此可见项目的功能主要通过中断完成，因此主程序的功能十分简单，如图 4.3 所示，

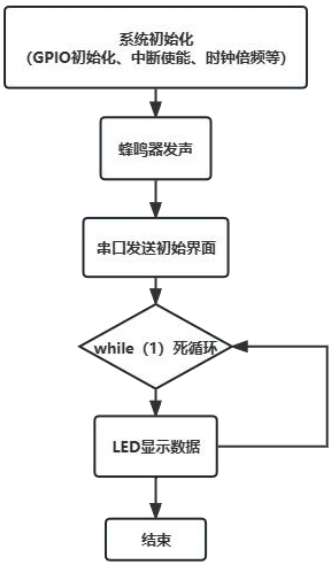

图 4.3 主程序流程图

下面根据不同的模块，分别介绍设计思路与内容。

4.2 频率测量模块

频率测量是基于两个定时器中断实现的。首先分析例程以及观察电路图如图 4.4 所示，在例程中，可以根据光敏传感器，测量电机转速。GK152 为红外光传感器，P2.3 和 P2.2 分别为直流电机和步进电机控制信号，使用 CD4052 四通道选择芯片选择哪个电机的信号，并通过 SN74LVC1

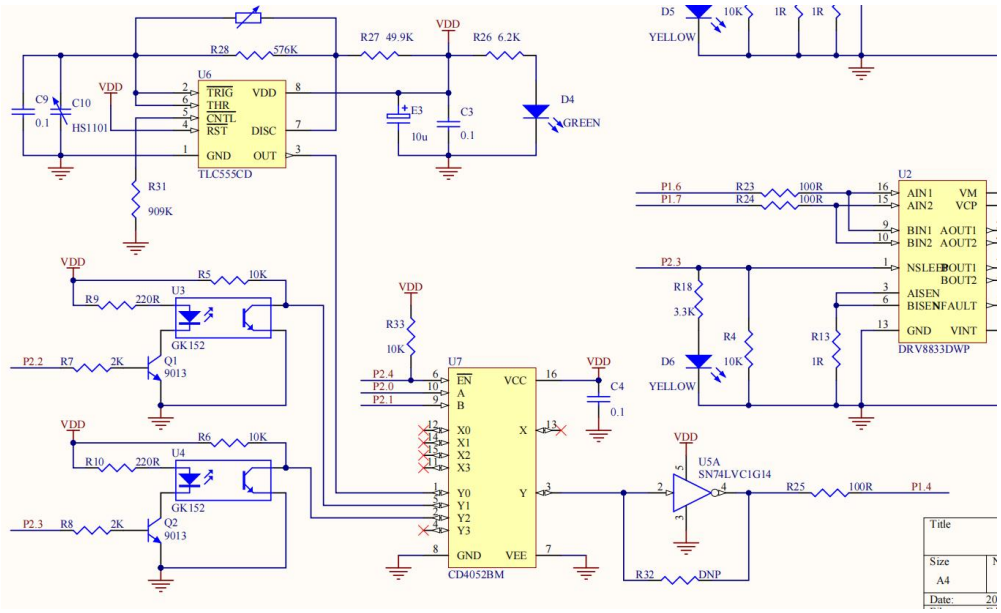

图 4.4 频率测量电路原理图

施密特触发器，使得信号更利于被定时器捕捉，在引脚 P1.4 输出信号。

用 TIMERA0 捕捉 P1.4 的信号，在 TIMERA0 的 ISR 中首先清除中断标志，并且频率计数器 G_DC_SpeedCnt 自加，

与此同时设置 TIMERB 定时器，定时为 1s 的定时器中断，把 G_DC_Speed 作为测量频率输出，同时清空频率计数器和中断标志位。同时在该 ISR 中把测量数据通过 UART 和 LCD 输出。

4.3 串口通信模块

在本项目中将 P3.4 和 P3.5 引脚分别设置为 CTS 和 RTS，在中断服务函数 USCI_A1_ISR 中，通过判断按位读取 UCA1RXBUF 寄存器中信息，分别执行握手信息显示、串口连续输出测量值、蜂鸣器测试以及模式切换四个功能，如果输入信息与预期输入不同，则会输出 error 错误信息。

值得注意的是串口波特率设置机制，本项目中通过 UCA1BRO、UCA1BR1 和 UCA1MCTL 控制串口通信的波特率，时钟频率除以波特率的整数部分，高八位存于 UCA1BR1 中，低八位存于 UCA1BRO 中，UCALMCTL 存储一位后的小数部分。通过对以上寄存器的设置，可将波特率配置为 19200bps。

4.4 显示模块

LED 数码管和段式 LCD 显示二者都用于显示测量结果，通过模 10 计算，得到每一位的数字和数据位数，以此显示数据并且将多余的数字零做灭灯处理。值得注意的是，LED 显示数据采用写入 char 类型数据显示，因此要将 int 类型的数据与存储不同对应数字点亮数据的 char 类型数字对应。

同时由于周期和频率都要显示，周期结果以μs 为单位保留 1 位小数，因此在显示周期时会点亮小数点 dp，以保证数据可读性。

五、设计结果

本项目基于 CCS 完成上述设计，实现频率测量和显示功能，系统握手成功时通过串口显示项目名称、姓名和学号等信息。通过串口通信可以实现模式切换。

同时频率计量程符合要求为 100-10000Hz，测量速度≤1s，为了更好评估测量数据误差，我统计了 10 个测量数据，并把频率和周期误差分别计算，结果如表 5.1 所示，

表 5.1 测量数据误差分析

频率/Hz		周期/μs	
实际值	测量值	实际值	测量值
110	110	9090.9	9090.9
210	208	4761.9	4807.7
510	510	1960.8	1960.8
800	798	1250.0	1253.1
960	960	1041.7	1041.6
1200	1200	833.3	833.3

2200	2197	454.5	455.1
5200	5194	192.3	192.5
8000	7996	125.0	125.0
9000	9000	111.1	111.1
平均误差	0.15%		0.15%

平均误差为0.15%，符合设计要求。

六、思考题

1) 常见的频率测量方法有哪几种？测量精度和哪些因素有关？如测量范围扩展为10~100000Hz，设计方案要作何调整？

答：频率测量方法包括：计数器方法，将信号的脉冲计数在固定的时间窗口内，然后根据计数值计算频率；周期测量方法，即在一个被测信号周期内，测量测量基准时钟个数，然后根据计数时钟个数计算频率。

测量精度受多种因素影响，1.计数器或定时器的分辨率，分辨率越高，测量精度越高。2.计数或周期测量的时间窗口，较长的时间窗口可以提高精度，但也会降低测量速度。3.信号波形稳定性，信号波形的稳定性影响测量结果的准确性。

对于测量范围扩展到10~100000Hz，设计方案需要作以下调整：1.改变计数器或定时器，针对较低频率的测量，可能需要选择较大范围的计数器或定时器，原来的1s定时器可能不适用较高频率的测量。2.显示屏需要扩展显示位数，保证正确读数。

2) 如何测量电机的转速？直流/步进电机的控制方式有何不同？

答：工程中有多种方法完成电机测速：通过编码器测量脉冲数、霍尔传感器测量转速等。在本项目中使用光电传感器测量转速，通过光电传感器未被遮挡的信号次数，得出电机转速，由于电机表盘孔洞为3个，因此实际频率为测量频率的三分之一。

电机的控制方式：直流电机通常通过调整电压或电流来控制转速和方向。可以使用PWM（脉宽调制）技术来控制直流电机的转速；步进电机是通过依次激活电机的不同绕组来实现转动的。控制步进电机通常需要使用专门的驱动器和控制器，通过发送脉冲信号来控制电机的步进角度和转速。

3) 如何实现电机转速的闭环恒速控制？

答：首先，需要选择合适的传感器来实时测量电机的转速。其次需要闭环控制器，通常使用PID（比例-积分-微分）控制器或其他适合的控制算法。PID控制器根据实际转速和目标转速之间的误差来计算控制信号，并输出到电机驱动器。同时，根据实际应用的要求，调节PID控制器的参数，以实现稳定的闭环控制和快速的响应。将传感器反馈的转速信号与设定的目标转速进行比较，计算误差，并将误差作为输入送入控制器。

4) 电子信息工程项目管理通常会涉及哪些内容？

答：电子信息工程项目管理涉及多个方面，以下是常见的内容：

项目计划：制定项目计划，明确项目的目标、范围、时间表、资源需求等，规划项目的整体布局。

风险管理：识别项目风险，评估风险的概率和影响，制定风险应对措施，以及监控和控制项目风险。

质量管理：确保项目交付的产品或成果符合质量标准，制定质量管理计划和质量检查措施，进行质量控制和质量保证。

成本管理：制定项目预算，跟踪项目的成本支出，控制成本，确保项目在预算范围内完成。

采购管理：对项目需要的设备、材料和服务进行采购，管理供应商关系，确保采购的质量和交付。

沟通管理：建立良好的项目沟通机制，与项目干系人保持有效的沟通，及时传递项目信息。

时间管理：制定项目进度计划，跟踪项目进度，确保项目按时完成。

5) 电子信息的专业工程实践除技术因素和项目管理外，还应考虑哪些非技术因素的影响？

答：电子信息专业工程的非技术因素包括：

经济因素：包括项目预算、投资回报率、成本效益等。在项目实践中，需要合理控制成本，确保项目的经济可行性。

文化因素：不同地区、国家的文化差异可能会影响项目实践的方式和沟通效果。考虑文化差异，进行有效的跨文化沟通，有助于项目的顺利进行。

环境因素：包括自然环境和社会环境。在项目实践中需要考虑项目对环境的影响，采取相应的环保措施。比如在有些国家就倾向于环保智能设备的开发。

市场因素：市场需求和竞争状况会影响项目的方向和市场营销策略。在项目实践中需要对市场进行充分调研和分析。

人力资源：项目团队的组成和能力对项目的成功至关重要。考虑人力资源的需求和配置，进行有效的团队管理和人才培养。

6) 温度测量常可用哪几种方法?各自特点?

答：热敏电阻：热敏电阻是一种电阻，其电阻值随着温度的变化而变化。特点是精度较高、响应速度快，但需要外部电路进行线性化处理。

热电偶：热电偶是由两种不同金属组成的回路，当两端温度不一致时，会产生电势差。热电偶的优点是测量范围广，适用于高温和低温测量，但精度相对较低。

红外线测温：利用物体发射的红外线能量来测量其表面温度。特点是非接触式测温，适用于测量高温、不规则形状的物体。

线性温度传感器：线性温度传感器是基于温度对其电特性的影响，输出线性电压或电流信号。常见的线性温度传感器有 LM35、LM75 等。

7) RS485 和 RS232 通信有何异同？如何实现 RS485 差分串行通信？

答：两者差别的主要体现在以下几点：

电压级别：RS485 是差分信号通信标准，数据传输通过电压的正负变化来表示 0 和 1。而 RS232 通过正负电平的变化来表示 0 和 1。

距离和速率：由于 RS485 采用差分信号传输，通信距离通常可达几千米。而 RS232 通信距离通常较短，一般在几十米以内。在速率方面，RS485 通常支持更高的数据传输速率。

要实现 RS485 差分串行通信，需要使用 RS485 收发器和支持 RS485 通信的串口控制器。RS485 收发器负责将串口的单端信号转换为差分信号，并进行数据发送和接收。将连接 RS485 收发器的 A 线和 B 线分别到总线上的两端，构成一个差分信号总线。将 RS485 收发器的 DE（Driver Enable）引脚和 RE（Receiver Enable）引脚连接到控制器的控制信号，以实现发送和接收的切换。

六、课程体会

在本项目中，我学习到了嵌入式系统的开发流程，从最开始的阅读电路原理图，查阅相关芯片的 datasheet，再到编写中断服务程序，实现不同模块的功能，最后是通过 debug 查找程序中的错误以及实际操作测试，修改代码逻辑。完整的开发流程使得我对于嵌入式系统开发有了更深的认识。

在本次调试过程中，我深刻认识到了嵌入式系统的复杂性，首先是代码的复杂性，由于中断服务的存在，代码中存在大量的全局变量，这使得代码编写尤为困难，尤其是编译器并不能很好的跳转变量命名的情况下，因此在开发过程中需要尤为注意代码命名和使用的规范，比如将声明文件和实现分别存放与 h 文件和 c 文件中等。其次是硬件系统的复杂性，MCU 作为一个整体，需要执行不同的任务，当个别任务对系统时钟进行修改后，将会影响其他模块的使用，比如在调试蜂鸣器时，由于系统时钟倍频 20MHz，导致单独出现了单独使用蜂鸣器没有问题，而整体使用不正常工作的现象，因此在开发过程中要尤为注意模块对于系统的操作。

总的来说，本次课程让我受益匪浅。除了学习了丰富的技术知识，更重要的是培养了分析问题、解决问题和团队合作的能力。我对电子信息工程的兴趣和热情进一步加深，期待在未来的工程实践中能够有所贡献。

七、参考文献

[1] Texas Instruments Incorporated ． MSP430F663x Mixed-Signal Microcontrollers datasheet[M]，2023.

[2] 德研电科．MSP430F6638_FFTB 实验指导书(MSP430F6638_DemoV2.0.)[M]，2013.

[3] 德研电科．MSP430F6638 一体化测试程序_使用说明书[M]，2013.

[4] 德研电科．MSP430F6638_FFTBV2.0 版电路原理图，2013.

[5] 傅强，杨艳．Launchpad 口袋实验平台指导书（电子版）[M]．TI 半导体技术有限公司,2013．

[6] 杨艳，傅强．从零开启大学生电子设计之路--基于 MSP430 LaunchPad 口袋实验平台[M]．北京：北京航空航天大学出版社，2014．

[7] 杭州艾研信息技术有限公司．MSP430 口袋实验套件 AY-G2PL KIT[EB/OL]. (2019-12-18)[2023-7-1]. http://www.hpati.com/ay_scm_pack/product_33.html

[8] 杭州艾研信息技术有限公司．AY-G2PL KIT_用户手册[M],2014.

[9] Texas Instruments Incorporated． MSP430 LaunchPad Value Line Development kit[EB/OL]. [2023-7-1]. http://www.ti.com/tool/msp-exp430g2

[10] TI 半导体技术有限公司．MSP-EXP430G2 LaunchPad 实验板用户指南 (Rev. C)[M]，2012．

[11] 徐安，陈耀，陆杰等．微控制器原理与应用实验教程[M]．北京：科学出版社，2009．

[12] 徐安，陈耀，方春华．微控制器原理与应用[M]. 北京：科学出版社，2006．

[13] ATMEL Corporation．AT89C52 datasheet[M]，2002．
[14] Philips Semiconductor．P87C51X2/52X2/54X2/58X2 80C51 8-bit micro controller family[M]，2002．
[15] 张毅刚，杨智明,付宁．基于 Proteus 的单片机课程的基础实验与课程设计[M]．北京：人民邮电出版社，2012．
[16] 肖看，李群芳．单片机原理、接口及应用——嵌入式系统技术基础（第 2 版）[M]．北京：清华大学出版社，2010．
[17] 徐爱钧，徐阳．ARM 嵌入式应用技术-基于 Proteus 虚拟仿真[M]．北京：航空航天大学出版社，2012．
[18] 刘刚,王立香．基于 MSP430 单片机和 LabVIEW 的温度监控系统设计[J]．科技创新与应用
[19] 江海波,郭建强,曹继承,等．基于 MSP430F6638 的无线智能组网的传感器采集系统的硬件研究[J]．自动化技术与应用
[20] 谢海武,严桂林,魏学刚,等．一款基于 MSP430F6638 的时钟及温度检测数据显示电路[J]．物联网技术．
[21] 陈小桥,李希希,李哲,等．基于 FPGA 和 MSP430 的频率特性测试仪[J]．实验室研究与探索．

装 订 线

八、主要程序清单

附录 1：main（）

程序主函数，将多个模块内容整合

```
#include <msp430f6638.h>
#include <stdio.h>
#include "string.h"
#include "cd4052.h"
#include "DRV8833.H"
#include "ADC.h"
#include "dc_motor.h"
#include "Timer.h"
#include "Segment_LCD.h"
#include "HAL_PMM.H"
#include "HAL_UCS.H"
#include "Frequency_dection.h"
#include "Uart.h"
#include "LCD.h"
#include "tm1638.h"


extern uint16_t results[8];
#define CPU_F ((double)1000000)
#define delay_us(x) __delay_cycles((long)(CPU_F*(double)x/1000000.0))
#define delay_ms(x) __delay_cycles((long)(CPU_F*(double)x/1000.0))
uint8_t ADC_FLAG = 0;
uint8_t Update = 0; //数码管更新标志位
uint16_t Pot_ADC_Result = 0 ;
extern unsigned int G_DC_Speed;
extern double DC_Times ;
extern unsigned char str_rx[20];
extern int flag_Timesend;

char SpeedStr[20];
const char tab[] =
{0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F,0x77,0x7C,0x39,0x5E,0x79,0x7
```

```
1};
/* Private function prototypes -----------------------------------------------*/
void Up_ClockFor_20MHZ(void);
void SPI_GPIO_Init(void);
void System_Init(void);
void TFTLCD_InitialDisplay(void);
void LED_Display(void);
//void uintToString(unsigned int value, char* str, int strSize);
void Message_Judge(void);
int main(void)
{
    System_Init();


     while(1)
     {
         LED_Display();
         LCD_Display();

      }
}
void LCD_Display()
{
     if(!flag_Timesend)
        {
        Write_Data(G_DC_Speed);
        }
        else
        {
        Write_Data(DC_Times);
        }


}




void Up_ClockFor_20MHZ(void)
{
    SetVCore(PMMCOREV_3);                       // Set Vcore to accomodate for max.
allowed system speed
    UCSCTL3 |= SELREF_2;                        // Set DCO FLL reference = REFO
    UCSCTL4 |= SELA_2;                          // Set ACLK = REFO
    Init_FLL_Settle(20000, 630);                // Set system clock to max (20MHz)
}

void System_Init(void)
{
      WDTCTL = WDTPW + WDTHOLD;    // Stop WDT

      //TFTLCD 显示
      SPI_GPIO_Init();
      Init_TFTLCDConfig();
      //TFTLCD_InitialDisplay();
```

```
    Up_ClockFor_20MHZ();          // 时钟倍频到 20MHz

    //段式 LCD 初始化
    Init_TS3A5017DR();        // Configure TS3A5017DR  IN1 and IN2
    Init_lcd();              // LCD 初始化
    Backlight_Enable();       // 打开背光*/
    LcdGo(1);                // 打开液晶模块
    LCD_Clear();             // 清屏

    //DC_MOTOR 相关模块初始化
    CD4052_Configure();
    DRV8833_Init();
    Step_Timer_Init();           // 配置控制步进电机的定时器
    TIM_Capture_Config();        // 配置输入捕获通道
    TIM_Update_Config();         // 配置定时器，定时更新捕获的数据

    //串口使能
    Init_UartConfig();

    //蜂鸣器初始化
    Beep_Init();
    int i=0;
                for(;i<3;i++)
                {
                Beep_One();
                    delay_ms(1000);

                }
    Uart_Out("Syetem Initial Successd\n");
    Uart_Out("1951706\n");
    Uart_Out("1.The System Information\n");
    Uart_Out("2.The Measure Result\n");
    Uart_Out("3.Beep Test\n");
    Uart_Out("4.Mode Change\n");

    //LED 初始化
    init_TM1638();

    TFTLCD_InitialDisplay();


    __bis_SR_register(GIE);      // 使能中断
}
void TFTLCD_InitialDisplay(void)
{
    TFTLCD_Clear(GREEN);//LCD 刷屏 GREEN
    SPI_Delay(20000);
    TFTLCD_ShowString(0,0,"1951706");//显示字符串
    SPI_Delay(20000);
    TFTLCD_ShowString(100,250,"Huang Kaikang");//显示字符串
    SPI_Delay(20000);
    TFTLCD_ShowString(200,0,"黄锴康");//显示字符串
    SPI_Delay(20000);
}
```

装 订 线

```
void LED_Display(void)
{
    //Write_allLED(0x00);
    unsigned int number;
    if(!flag_Timesend)
    {
        number = G_DC_Speed;
    }
    else
    {
        number = DC_Times;
    }
    unsigned int NumSpeed[6];  // 用于存储每个数字的数组
    int digits = 0;
    if (number == 0) {
        digits = 1;  // 对于数字 0，视为一位数
    }
    else{
    int temp = number;
    while (temp != 0){
        int digit = temp % 10;
        temp /= 10;
        NumSpeed[digits] = digit;
        digits++;
    }
    }
    int i = 0;
    if(digits==3)
        Write_DATA(0,0x00);
    for (; i < digits; i++) {
        if(digits - i <= 4)
            Write_DATA((4+i-digits)*2,tab[NumSpeed[digits - i - 1]]);

    }
}


void SPI_GPIO_Init(void)
{
  P8SEL |= BIT4+BIT5+BIT6;                  //Bit=1:Peripheral module function is
selectedfor the pin
  P8DIR |=  BIT4+BIT5;                     //Port configured as output
  P8DIR &=  BIT6;                          //MISO 配置成输入

  // Configure TS3A5017DR  IN1 and IN2
  P3DIR |= BIT4 + BIT5;   //P3.4 : IN1 ; P3.5 : IN2   set as output
  P3OUT |= BIT4;          //IN1 = 1
  P3OUT &= ~BIT5;         //IN2 = 0
  P3DIR |= BIT7;
  P3OUT |= BIT7;

}
```

MSP430_INT.C

中断服务程序

```
/**

********************************************************************************
  * @文件名 MSP430_INT.c
  * @作者   DY
  * @版本   V1.0
  * @日期   03/12/2012
  * @摘要   简要描述文件的内容

********************************************************************************
  * @Copyright (c)2012,上海德研电子科技有限公司
  * @All right reserved.
  */

/* Includes
------------------------------------------------------------------*/
#include "msp430f6638.h"
#include "stdint.h"

#include "Frequency_dection.h"
#include "Segment_LCD.h"
#include "Uart.h"
#include "math.h"
#include "Beep.h"
extern int8_t ADC_FLAG;

extern unsigned int G_DC_Speed;  //DC_Motor 直流电机"1 秒内 CCR3 采样到的频率次数
extern unsigned int G_DC_SpeedCnt;
extern char* receivedString[20];
//extern unsigned int F_1s_Frequency_DC_Motor;//判断是否为计时器结束计时
/* Private typedef -----------------------------------------------------------*/

/* Private define ------------------------------------------------------------*/
#define   Num_of_Results   8
#define   Num_of_Str       20
#define CPU_F ((double)1000000)
#define delay_us(x) __delay_cycles((long)(CPU_F*(double)x/1000000.0))
#define delay_ms(x) __delay_cycles((long)(CPU_F*(double)x/1000.0))
/* Private macro -------------------------------------------------------------*/
int index_str = 0;

int flag_Uartsend = 0;//////////////////yhy
int flag_Timesend = 1;
double DC_Times ;
volatile unsigned int results[Num_of_Results];
volatile unsigned char str_rx[Num_of_Str];
unsigned char CharSpeed[6];  // 用于存储每个数字的数组
unsigned char CharTimes[6];  // 用于存储每个数字的数组

/*!
*函数功能：ADC 中断服务函数
*输入参数：无
*输出参数：无
*返回值：  无
```

```
*/
void uintToString(void)
{
    unsigned int number = G_DC_Speed;
    int digits = 0;
    int cnt = 1;

    int temp1 = number;
    int temp2 = number;

    while (temp1 != 0)
    {
    int digit = temp1 % 10;
        temp1 /= 10;
        digits++;
    }

    while (temp2 != 0)
    {
    int i = temp2 % 10;
        temp2 /= 10;
        CharSpeed[digits - cnt] = '0' + i;
        cnt++;
    }

    CharSpeed[digits] = '\0';

}

void doubleToString(void)
{
    unsigned int number =round(DC_Times);
    int digits = 0;
    int cnt = 1;

    int temp1 = number;
    int temp2 = number;

    while (temp1 != 0)
    {
    int digit = temp1 % 10;
        temp1 /= 10;
        digits++;
    }

    while (temp2 != 0)
    {
    int i = temp2 % 10;
        temp2 /= 10;
        CharTimes[digits - cnt] = '0' + i;
        cnt++;
    }

    CharTimes[digits] = '\0';

}
```

装 订 线

装 订 线

```
#pragma vector=ADC12_VECTOR
__interrupt void ADC12ISR (void)
{
  static unsigned char index = 0;

  switch(__even_in_range(ADC12IV,34))
  {
  case  0: break;                           // Vector  0:  No interrupt
  case  2: break;                           // Vector  2:  ADC overflow
  case  4: break;                           // Vector  4:  ADC timing overflow
  case  6: break;                           // Vector  6:  ADC12IFG0
  case  8: break;                           // Vector  8:  ADC12IFG1
  case 10: break;                           // Vector 10:  ADC12IFG2
  case 12: break;                           // Vector 12:  ADC12IFG3
  case 14: break;                           // Vector 14:  ADC12IFG4
  case 16: break;                           // Vector 16:  ADC12IFG5
  case 18:                                  // Vector 18:  ADC12IFG6
  results[index] = ADC12MEM6;                 // Move results
  index++;                                  // Increment results index, modulo; Set
Breakpoint1 here

  if (index == 8)
  {
    index = 0;
    ADC_FLAG = 1;
  }
  break;
  case 20: break;                           // Vector 20:  ADC12IFG7
  case 22: break;                           // Vector 22:  ADC12IFG8
  case 24: break;                           // Vector 24:  ADC12IFG9
  case 26: break;                           // Vector 26:  ADC12IFG10
  case 28: break;                           // Vector 28:  ADC12IFG11
  case 30: break;                           // Vector 30:  ADC12IFG12
  case 32: break;                           // Vector 32:  ADC12IFG13
  case 34: break;                           // Vector 34:  ADC12IFG14
  default: break;
  }
}



/*!
*函数功能：TIMER1 中断服务函数-用来实现步进电机，每一步之间的时间间隔
*输入参数：无
*输出参数：无
*返回值：   无
*/

// Timer1 A0 interrupt service routine
#pragma vector=TIMER1_A0_VECTOR
__interrupt void TIMER1_A0_ISR(void)
{
```

```
}

/*!
*函数功能：TIMER0 中断服务函数-用来实现步进电机，每一步之间的时间间隔
*输入参数：无
*输出参数：无
*返回值：   无
*note:     Timer0_A5 CC1-4, TA
*/

#pragma vector = TIMER0_A1_VECTOR
__interrupt void Capture_Input_ISR(void)
{
  TA0CCTL3 &= ~CCIFG;//清除中断标志
  G_DC_SpeedCnt++;
}

/*!
*函数功能：Timer0_B7 中断服务函数,用来定时更新捕获计数
*输入参数：无
*输出参数：无
*返回值  ：无
*note    : Timer0_B7 CC0
*/
#pragma vector=TIMERB0_VECTOR
__interrupt void Update_Count_ISR(void)
{
    G_DC_Speed = G_DC_SpeedCnt;
    double D_Speed = G_DC_Speed;
    DC_Times = 1000000/D_Speed;
    LCD_Clear();



    if(flag_Uartsend) {
    if(!flag_Timesend)
    {
        uintToString();
            Uart_Out("The Frequency of No.1  is ");
            Uart_Out(CharSpeed);
            Uart_Out(" Hz\n");}
    else{
        doubleToString();
            Uart_Out("The Time of No.1 is ");
            Uart_Out(CharTimes);
            Uart_Out(" us\n");}
    }
    //CharSpeed = "0";
    G_DC_SpeedCnt=0;
    TB0CCTL0 &= ~CCIFG;//清除中断标志
 }
```

```
// Echo back RXed character, confirm TX buffer is ready first
#pragma vector=USCI_A1_VECTOR
__interrupt void USCI_A1_ISR(void)
{

  switch(__even_in_range(UCA1IV,4))
  {
  case 0:break;                             // Vector 0 - no interrupt
  case 2:                                   // Vector 2 - RXIFG

     //while (!(UCA1IFG & UCRXIFG));
          if(UCA1RXBUF == '1')
            {
            Uart_Out("Success\n");
            Uart_Out("Frequency Dector\n");
            Uart_Out("1951706\n");
            Uart_Out("Huang Kaikang\n");
            flag_Uartsend = 0;
            }
            else if(UCA1RXBUF == '2')
            {
            flag_Uartsend = 1;

            }
            else if(UCA1RXBUF == '3')
            {
            uintToString();
                Uart_Out("Beep\n");
                //Uart_Out(CharSpeed);
                int i=0;
                for(;i<3;i++)
                {
                Beep_One();
                    delay_ms(1000);

                }

                flag_Uartsend = 0;
            }
            else if(UCA1RXBUF == '4'){


            doubleToString();
            if(flag_Timesend)
            {
                flag_Timesend = 0;
                Uart_Out("Frequency Mode");

            }
            else
            {
                flag_Timesend = 1;
                Uart_Out("Time Mode");

            }
```

```
            flag_Uartsend = 0;

            }
            else{
            Uart_Out("Error\n");


            }



      break;

  case 4:break;                           // Vector 4 - TXIFG
  default: break;
  }
  }


#pragma vector=USCI_B1_VECTOR
__interrupt void USCI_B1_ISR(void)
{
  volatile unsigned int i;

  switch(__even_in_range(UCB1IV,4))
  {
    case 0: break;                         // Vector 0 - no interrupt
    case 2:                                // Vector 2 - RXIFG
      break;
    case 4:                                // Vector 4 - TXIFG
      break;
    default: break;
  }
}
```

装 订 线