综合案例——电子秤设计与实现

◆ 5.1 项目概述

电子秤是采用现代传感技术、电子技术和计算机技术的一种电子称量装置,能够在实际生活中满足人们对物品称重时"快速、准确、连续、自动"的要求,同时保持相对较低的称量误差,使物品称量符合法制计量管理和过程控制要求。构成电子秤的核心原件有称重传感器、微控制器以及显示模块。其中,通过MCU实现对称重传感器数据的采集是实现电子秤的核心任务,称重传感器 有光电式、液压式、电容式、磁极变形式、振动式、陀螺仪式、电阻应变式等几类, 在电子秤设计中,电阻应变式压力传感器的应用最为广泛。

本项目的设置旨在培养读者综合运用所学理论知识和技能解决实际问题 的能力、动手实践能力、团队协作能力以及文档撰写和管理能力。本项目要求 读者完成硬件选型、电路设计、嵌入式软件开发、文档撰写等具体任务,其任务 难度和工作量适中。电子秤项目涉及如下3个核心业务模块。

1. 获取质量信号

第

5

音

电阻应变式压力传感器是目前应用最为广泛的传感器之一,其基本原理是 电阻的应变效应,即导体受机械形变时,其电阻值发生变化,称为"应变效应"。 构建电子秤的首要任务是通过 MCU 控制压力传感器获取压力信号并根据调校 算法建立传感器输出与物品实际质量之间的函数关系。

2. 显示质量信息

显示物品质量信息是电子秤必须具备的功能,通过将压力信号转变为物品 的质量信息并调用显示设备的 API 函数实现对物品质量信息的显示。本项目 建议采用单片机和嵌入式开发领域常用的 LCD1602 显示器,其成本低廉、易于 控制的优势使其在市场上占有相当的份额。通过设计 MCU 和 LCD1602 显示 器之间的接口电路、移植驱动程序、设计和开发质量信息显示程序,读者需了解 嵌入式产品设计开发的基本流程,熟悉相关工具的使用,掌握嵌入式开发的相 关技术。

3. 电子秤设置模块

功能完备的电子秤需要具备量程设置,显示屏亮度调节,超量程报警等基本功能。本项目要求读者通过电位器调节实现对屏幕亮度的控制,通过按键实现量程设置,通过 LED灯和蜂鸣器实现物品称重时的超量程声光报警。通过实现这些功能,读者能够熟悉 常见元器件的使用方法以及相关控制程序的设计方法。

◆ 5.2 设计目的

本项目旨在培养读者综合应用物联网工程专业的基本知识和基本理论分析解决实际 问题、设计开发嵌入式系统、研究相关课题、使用现代工具、实施项目管理、进行团队协作 以及持续学习的能力。在新工科教育背景下,通过本项目的实施,能够为达到如下教学目 标打下一定的基础。

(1)提高政治觉悟,自觉拥护党的领导,忠于国家,忠于人民,立志发奋读书,努力掌握专业知识,练就过硬的业务素质,成长为对祖国建设有用的高科技人才。

(2)熟悉构建嵌入式系统的常用元器件和传感器,掌握常见电路设计方法,能够使用 相关工具完成硬件方案设计。

- (3) 熟悉嵌入式软件开发方法,熟练使用一种嵌入式开发工具。
- (4) 具备一定的文档撰写能力,能够独立完成项目中相关文档资料的编写和管理。
- (5) 具备一定的自学能力,能够独立收集相关资料,自学相关知识并完成任务。
- (6) 具备一定的团队协作能力,能够和团队成员积极沟通,协作完成项目实施工作。
- (7) 掌握现代项目管理工具,理解现代嵌入式项目的实施流程。
- (8) 掌握一定的工程实施规范,具备一定的人文素养,具有高度的社会责任感。

◆ 5.3 预备知识

模拟电路和数字电路基础、C程序设计基础、嵌入式系统与设计基础、传感器原理与应用基础。

◆ 5.4 系统需求

5.4.1 系统功能性需求

- (1) 设计一个嵌入式系统——电子秤。
- (2) 电子秤能够实时采集物体质量并显示在显示屏。
- (3) 电子秤具备去皮功能。
- (4) 电子秤具备量程设置功能。
- (5) 电子秤具备超量程声光报警功能。

(6) 电子秤显示屏具备亮度调节功能。

5.4.2 系统性能需求

(1)当用户在秤盘上放置物品时,质量数据需实时显示,不得有太大的延迟和卡顿现象。

(2)质量数据需经过仔细调校,显示结果准确、稳定、可靠。

(3) 去皮、量程设置等操作需灵敏,不得有卡顿、停止响应等问题。

5.4.3 其他需求

(1) 易用性:操作简单易用,对用户友好。

(2)可靠性:系统能够持续工作,避免出现卡顿、操作失灵等问题。

(3) 低功耗:模拟智能设备的嵌入式系统需保持较低功耗。

(4)环境适应性:模拟智能设备的嵌入式系统需能够在常规的监测环境完成称重。

(5) 安全性: 设备运行能够保证程序和数据安全, 杜绝漏电、短路、数据丢失等故障的出现。

(6) 经济性:系统开发成本较低,开发周期较短。

◆ 5.5 硬件设计与实现

5.5.1 电路设计

电子秤的硬件电路主要包括电源电路、按键控制电路、LED指示灯控制电路、蜂鸣器控制电路、LCD1602显示电路以及HX711压力传感器控制电路,其硬件设计框图如图 5-1 所示。



采用常规元器件,按照系统需求,完成硬件电路设计,图 5-2 给出电子秤原理图的一种设计方案,读者完全可以根据所掌握的专业知识,提出其他设计方案。对于初学者而言,可按照本书提出的设计方案认真实施项目,达到锻炼能力的目的。

物联网系统开发综合实验教程



5.5.2 硬件选型

根据硬件设计原理图,制作 PCB,购置元器件。根据图 5-2 的设计图,表 5-1 列出一份物料清单(Bill of Materials,BOM),清单中所列元器件均为常见的元器件,很容易通过 网购渠道获得。

物料名称	规格型号	原理图标号	数量	参考图片
微控制器	STM32F103C8T6 核心板	U4	1	
按键	立式按键,6×6×4.3	K2,K3,K4	3	
LED 报警灯	5mm,白发红	LED1	1	

表 5-1 电子秤物料清单

214

第5章 综合案例——电子秤设计与实现

续表

物料名称	规格型号	原理图标号	数量	参考图片
三极管	S8050	Q2	1	Sector C L L L L L L L L L L L L L L L L L L L
蜂鸣器	有源蜂鸣器	BELL	1	Control of the second s
液晶显示器	LCD1602	U3	1	
电阻器	1K	R4,R9	2	Status.
可调电位器	3362P 电位器,10K	RT1	1	
电源底座	DC005 插头,3.5×1.5	U1	1	
按压式开关	8×8 自锁式按键开关	U6	1	
AD转换模块	HX711	U2	1	
压力传感器	YZC131 微型称重传感模块	U9	1	··· ···

(215)

5.5.3 制作硬件实物

使用设计好的 PCB 或者万用板完成对硬件实物的焊接,实现电子秤硬件设计方案。 建议具备条件的读者使用 Altium Designer 认真设计原理图和 PCB 设计图,完成 PCB 的 制作,尽量采用贴片元件实现硬件电路。不具备条件的读者建议采购穿通式元件,在万用 板上实现硬件电路。

◆ 5.6 嵌入式软件设计与实现

5.6.1 创建项目模板

1. 准备 STM32 库函数

STM32 库函数可通过官方网站进行下载,下载后解压生成如图 5-3 所示的目录结构。

Pro	gram-SSD (D:) > STM32F103 > 标准库	✓ U <> 携	號"标准库"	
^	名称 ^	修改日期	类型	大小
	STM32F10x_StdPeriph_Lib_V3.5.0	2020/12/2 10:01	文件夹	
	an.stsw-stm32054.zip	2020/11/29 18:30	WinRAR ZIP 压缩	21,617 KB

图 5-3 STM32 库函数

2. STM32 库函数简介

ST(意法半导体)为了方便用户开发程序,提供了一套丰富的 STM32 固件库。固件 库就是函数的集合,固件库函数的作用是向下负责与寄存器直接打交道,向上提供用户函 数调用的接口(API)。固件库将这些寄存器底层操作都封装起来,提供一整套接口(API) 供开发者调用,大多数场合下,读者不需要知道某个函数到底操作的是哪个寄存器,只需 要知道调用哪些函数即可。

ARM 公司为了让不同的芯片公司生产的 Cortex-M3 芯片能在软件上基本兼容,和芯片生产商共同提出 ARM Cortex 微控制器软件接口标准 (Cortex Microcontroller Software Interface Standard, CMSIS)。

CMSIS 分为 3 个基本功能层具体如下。

(1) 核内外设访问层:由 ARM 公司提供的访问,定义处理器内部寄存器地址以及功能函数。

(2) 中间件访问层: 定义访问中间件的通用 API,也由 ARM 公司提供。

(3) 外设访问层: 定义硬件寄存器的地址以及外设的访问函数。

CMSIS 层在整个系统中处于中间层,向下负责与内核和各个外设直接打交道,向上 提供实时操作系统用户程序调用的函数接口。如果没有 CMSIS 标准,那么各个芯片公司 就会设计风格各异的库函数,而 CMSIS 标准就是要强制规定,芯片生产公司设计的库函数必须按照 CMSIS 这套规范来设计。

举一个简单的例子,在使用 STM32 芯片的时候首先要进行系统初始化,CMSIS 规范规定,系统初始化函数名字必须为 SystemInit,所以各个芯片公司开发自己的库函数时必须用 SystemInit 对系统进行初始化。CMSIS 还对各个外设驱动文件的文件名进行了规范。

可在 ST 官网下载 ST 固件库,固件库是不断完善升级的,有不同的版本,电子秤项目 使用 V3.5 版本的固件库 STM32F10x_StdPeriph_Lib_V3.5.0。下面查看 STM32V3.5 库 函数的基本情况。

Libraries 目录下有 CMSIS 和 STM32F10x_StdPeriph_Driver 两个目录,这两个目录 包含固件库核心的所有子目录和文件。其中,CMSIS 目录下是启动文件,STM32F10x_ StdPeriph_Driver 存放的是 STM32 固件库源码文件。源文件目录下面的 inc 目录存放的 是 stm32f10x_xxx.h 头文件,无须改动。src 目录下面放的是 stm32f10x_xxx.c 格式的固 件库源码文件。每一个.c 文件和一个相应的.h 文件对应。这里的文件也是固件库的核 心文件,每个外设对应一组文件。建立工程的时候会用到 Libraries 目录下的文件。

Project 目录下有两个目录。顾名思义,STM32F10x_StdPeriph_Examples 目录下面 存放着 ST 官方提供的固件实例源码,在以后的开发过程中,可以参考修改官方提供的实 例来快速驱动自己的外设,通过网购得到的大多数开发板实例都参考了官方提供的实例 源码,这些源码对后续的学习非常重要。STM32F10x_StdPeriph_Template 目录下面存 放的是工程模板。

Utilities 目录下是官方评估板的一些对应源码,这个可以忽略不看。

根目录下还有一个 stm32f10x_stdperiph_lib_um.chm 文件,这是一个固件库的帮助 文档,该文档非常有用,关于库函数的帮助信息都在该文档中详细记录。

下面着重介绍 Libraries 目录下的几个重要文件。core_cm3.c 和 core_cm3.h 文件位 于\Libraries\CMSIS\CM3\CoreSupport 目录下,这是 CMSIS 核心文件,提供进入 CM3 内核接口,也由 ARM 公司提供,对所有 CM3 内核的芯片都一样,初学者永远都不需要修 改这个文件。

与 CoreSupport 同一级目录中还有一个 DeviceSupport 目录。DeviceSupport\ST\ STM32F10xt 目录下面主要存放一些启动文件以及比较基础的寄存器定义和中断向量定 义的文件。该目录下共有 3 个文件,即 system_stm32f10x.c、system_stm32f10x.h 以及 stm32f10x.h 文件。其中 system_stm32f10x.c 和对应的头文件 system_stm32f10x.h 文 件的功能是设置系统以及总线时钟,该文件中定义了非常重要的 SystemInit()函数,这个 函数在系统启动的时候会被调用,用来设置系统的整个时钟系统。

stm32f10x.h 文件也非常重要,该文件中定义了很多结构体和宏,主要完成对系统寄存器的定义以及封装对内存的操作。

3. 创建项目模板

1) 创建项目目录

创建 D:\STM32F103\My_Elec_Scale 目录作为项目目录,并将 STM32 库函数的

217

Libraries 目录复制到该目录下并重命名为 STM32F10x_FWlib。同时,还需要在该项目 目录下分别创建如下几个目录用来保存各类文件。通俗地讲,项目开发过程中需要建立 不同的目录分门别类地管理项目中的各类文件,如图 5-4 所示。

My_Elec_Scale			
共享 查看			
▮ 〉 此电脑 〉	Program-SSD (D:) > STM32F103 > My_Elec_Sc	 	
*	^ 名称 [^]	修改日期	类型 大小
*	CORE 存放与内核相关的文件	2020/12/2 10:42	文件夹
*	DOC 存放项目文档资料	2021/10/14 21:18	文件夹
*	II HARDWARE 存放项目各种硬件驱动	2020/12/5 15:15	文件夹
03]物联网系统开发	■ OBJ 存放编译后的目标文件	2020/12/20 12:12	文件夹
盘)	STM32F10x_FWlib STM32库函数文件	夹2020/12/4 9:12	文件夹
	SYSTEM 存放系统相关文件	2020/12/4 9:12	文件夹
	USER 用户自己的文件夹	2020/12/20 12:12	文件夹
	工具软件	2020/12/4 9:12	文件夹

图 5-4 项目目录结构

将库函数 STM32F10x_StdPeriph_Lib_V3.5.0\Project\STM32F10x_StdPeriph_ Template 下的如下 5 个文件复制到新建的 USER 目录下,如图 5-5 所示。

stimszeriox_stapenpn	_Template						
共享 查看							
📕 « STM32F103 >	« STM32F103 > STM32函数库V3.5 > STM32F10x_StdPeriph_Lib_V3.5.0 > Project > STM32F10x_StdPeriph_Template						
^	^	修改日期	类型	大小			
	EWARM	2020/12/2 10:01	文件夹				
	HITOP	2020/12/2 10:01	文件夹				
*	MDK-ARM	2020/12/2 10:01	文件夹				
*	RIDE	2020/12/2 10:01	文件夹				
*	TrueSTUDIO	2020/12/2 10:01	文件夹				
义	🗋 main.c	2011/4/4 19:03	C Source File	8 KB			
项目申报指南-课题	Release_Notes.html	2011/4/6 18:15	HTML 文档	30 KB			
10	stm32f10x_conf.h	2011/4/4 19:03	C/C++ Header F	4 KB			
用综合实训ppt	stm32f10x_it.c	2011/4/4 19:03	C Source File	5 KB			
	stm32f10x_it.h	2011/4/4 19:03	C/C++ Header F	3 KB			
	system_stm32f10x.c	2011/4/4 19:03	C Source File	36 KB			

图 5-5 复制模板文件

2) 新建工程

(1) 打开 Keil5,选择 Project→New µVision Project...菜单项,如图 5-6 所示。

(2) 建立工程,保存在 My_Elec_Scale/USER/Project 下,然后选择 CPU 型号,我们用的是 STM32F103C8,如图 5-7 所示。

3) 设置工程

(1) 在左侧的 Project 视图中将 Target1 重命名为 My_Elec_Scale(单击 Target1 的 名称,待选中后直接输入新的名称)。



图 5-6 新建工程菜单

Select Device for Target 'Target 1' X
CPU Vendor: STMicroelectronics Device: STM32F103C8 Toolset: ARM Search: P\$ STM32F0 Series P\$ STM32F10 Series P\$ STM32F101 P\$ STM32F101 P\$ STM32F102 P\$ STM32F103C6 STM32F103C6 STM32F103C6 STM32F103C8 P\$ STM32F103C8 P\$ STM32F
UK Cancel Help
图 5-7 选择 CPU 型号

(2) 右击项目名称 My_Elec_Scale 选择 Manage Project Items 菜单项,弹出 Manage Project Items...菜单,如图 5-8 所示。



(3) 在弹出的窗口中,可以在 Project Targets 栏内修改项目名称,在 Group 栏内创建 分组,在 Files 栏内为每个分组添加相关文件。

(4) 创建一个名为 USER 的 Group,在该 Group 内添加..\ My_Elec_Scale\USER 目 录下的 main.c,stm32f10x_it.c 以及 system_stm32f10x.c 这 3 个文件,如图 5-9 所示。

nage Project Items oject Items Folders/Extensions	Books			
Project Targets: K + 4	Groups: USER HARDWARE SYSTEM CORE FWLb		Files: X main.c. stm3210x_jt.c. system_stm3210x.c	
Set as Current Target	1		Add Files Add Files as Image	
	OK	Cancel	1	Help

图 5-9 创建分组、添加文件

(5) 创建 FWLib 分组,并在该分组添加操作 STM32 各类外设的库函数,即将目录.. \My_Elec_Scale\STM32F10x_FWlib\STM32F10x_StdPeriph_Driver\src 下即将用到的 外设文件全部添加即可,如图 5-10 所示。

	(200 20 4 1		
	USER HARDWARE SYSTEM CORE, FWLb		misc.c stm32110c_adc.c stm32110c_bkp.c stm32110c_can.c stm32110c_cec.c stm32110c_cec.c stm32110c_cec.c stm32110c_dec.c stm32110c_dec.c stm32110c_dec.c stm32110c_fan.c stm32110c_fan.c stm32110c_fan.c stm32110c_fan.c stm32110c_fan.c stm32110c_fan.c stm32110c_fan.c stm32110c_fan.c stm32110c_fan.c stm32110c_pai.c stm3210c_pai.c stm32	
			stm32f10x_rtc.c stm32f10x_sdio.c	<u> </u>
Set as Current Target	1		Add Files	

图 5-10 创建 FWLib 分组

(6) 在 CORE 这一 Group 内,添加两个文件。第一个文件是位于..\My_Elec_Scale\ STM32F10x_FWlib\CMSIS\CM3\CoreSupport\目录下的 core_cm3.c,第二个是目录..