

# 华中科技大学电子与信息工程系 2012 年 TI 杯电子设计大赛项目总结报告

项目名称: 智能药盒

团队成员: 通信工程 0907 班高盛

通信工程 0907 班胡中天

电子信息 0903 班陈圣

指导教师: 曾喻江

2012 年 7 月 4 日

## 课题名称：智能药盒

### 【摘要】

本项目以 MSP430 单片机和与客户机的通信实现为主。整个项目在 MSP430 单片机上面实现了相关数据显示输出和端口查询输入以及通过有线及无线方式与客户机实现通信等功能。我们的设计思路是单片机可以自己独立运行也可以通过客户机控制单片机的运行行为。

本项目的意义旨在方便用户记忆以及存取药物。通过选用便于集成的单片机系统可以很方便地完成药物管理与提醒等功能。为了方便设计，我们将整个任务分成了硬件设计和软件设计两大部分。其中软件设计又细分为单片机端软件与客户机软件设计两部分。下面分别介绍硬件和软件部分的大致思路。

硬件部分主要在于药盒开关状态采集以及药盒机械结构设计两部分。其中状态采集属于电子部分，采用点触式开关通过上拉电阻默认上拉到高电平并将开关状态输入到单片机的 GPIO 口来实现开关状态采集。药盒的机械结构设计采用抽拉式盒子实现。其中在盒子尾端装入点触式开关来实现使用感应的功能。本产品设计之初本来准备采用子母盒的方式来实现。其中母盒采用较大容量的药盒可以对整个家庭的药品集中进行管理。子盒采用较小的药盒便于随身携带与提醒功能。但因单片机数量限制我们只做了母盒，子盒在程序上可与母盒共享同一套程序，因盒间通信采用蓝牙形式链接因此只需在蓝牙硬件跳线上修改主从设备关系即可区分子母盒而无需修改代码。因此在此以母盒的实现代替子盒的功能。

软件部分单片机端采用中断方式实现定时与功能按键设置等功能，查询方式实现药盒开关状态采集。因为药盒开关可能同时出现多个开关状态，采用中断嵌套方式无法区分嵌套优先级以及嵌套顺序，采用查询方式可以有效避免此类问题。另外通讯可以通过串口方式以及蓝牙方式两种方式与客户机进行通信。截止目前我们的通信方式可以实现单片机的时间设置和新加药品管理等功能。客户机端我们可以调用串口通信程序接受与处理来自单片机的消息。并能够发送管理指令到单片机。

**【关键词】：**单片机 无线通信 信息管理

## 1. 概述

当下,人们对于健康的关注度不断提高。而在生活节奏不断加快的今天,人们经常会因为忙于工作而忘记服用一些药物或保健品。因此出于简便生活、利于健康的目的,我们设计了一款基于 MSP430F149 单片机的智能药盒,它加入了蓝牙模块与显示模块,实现了 pc 端与药盒的蓝牙通信进行信息更新、弹出抽屉的信息显示、闲置时显示时钟信息等功能。

## 2. 设计目标

### 2.1 基本功能

- 用户打开该盒子任意一个储纳间时显示该储纳间中所存放的药物名称、所需服用的药物剂量以及即将过期天数以提醒用户。
- 药盒在抽屉全部关上的时候在显示屏上显示目前时间。时间包括年、月、日与时、分、秒。
- 用户在药盒开机时的图片与欢迎语等信息的显示功能

### 2.2 拓展功能

- 进实现智能药盒的蓝牙通信,可以通过 pc 端使用蓝牙通信设置药盒显示时间。
- 可以在 pc 端通过蓝牙传输提醒信息到药盒,在显示屏上即时显示。

## 3. 团队组成与任务分工

本小组分工的指导思想如下:为了最大程度实现并行性,我们按照模块来划分任务。不同模块之间首先需要将相互之间的接口定义好,定义完成以后不同模块就能够相对地独立工作了。

高盛,1991年11月16日生,2009年考入华中科技大学电子与信息工程系。本次项目实验中,担任队长,主要负责整个项目进度,硬件模块的搭建,各模块方案选择与测试验证,文档的整理,系统的测试,展示网站制作。

胡中天,1990年5月29日生,2009年考入华中科技大学电子与信息工程系。本次项目实验中,主要负责元器件的采购,各硬件模块的焊接,系统的测试,文档的整理,药盒设计与制作,展示网站翻译。

陈圣,1991年1月16日生,2009年考入华中科技大学电子与信息工程系。本次项目实验中,主要负责药盒的采购,PC端软件编程,硬件模块的焊接,文档的整理,药盒设计与制作,展示网站翻译。

。

## 4. 硬件选择与分析

### 1. 单片机的选择

- ♣ 方案一: Ti 赞助的 16 位 MSP430 LaunchPad, 超低功耗, 且由于 Ti 赞助可以向老师借到不用额外的开支, 但是外接接口太少不够用。
- ♣ 方案二: MSP430F149 单片机最小系统板, 额外设立了 12864 液晶接口, 方便使用

显示器件，支持 3.3V 和 5V。

最后选择了方案二。

## 2. 无线通信模块的选择

- ♣ 方案一：zigbee 模块，低功耗低成本，使用简单便捷，但无法直接与电脑通信。
- ♣ 方案二：蓝牙模块 BC04 带转接板后可以直接用在 MSP430 上，且可以与手机、电脑、PDA 等设备配对连接然后通信。

最后选择了方案二。

## 3. 显示模块的选择

- ♣ 方案一：1602 液晶屏，价格便宜，但显示字符数太少，且只能显示英文。
- ♣ 方案二：12864 液晶屏，淘宝大约都在 50 元左右，带中文字库，显示字数较多，带有文字和图形混合显示功能等，功能较 1602 丰富许多。

最后选择了方案二。

## 4. 药盒的选择与制作

### 药盒的选择

- ♣ 想法一：因为需要有多个抽拉式储纳间的盒子，所以首先想到的是小的药盒，但是找了数家药店不是盒子太小就是没有抽拉式的储纳间的药盒。最后放弃了直接找药盒的想法。
- ♣ 想法二：由于又要考虑到药盒需要匹配单片机的大小，所以又想到了放小杂物的小储物盒。于是去中百仓储寻找。在家居用品区找到了 2 个抽拉式储物盒，但体积有点过于庞大，且储纳间在抽拉的时候有上下左右的松动，药盒开闭状态的感应开关在这种情况下可能被不小心触发，所以只能再寻找合适药盒。
- ♣ 想法三：在对之前找到的储物盒不满意的情况下，我们在超市寻觅替代品时碰巧看到了调味品的架子上也有抽拉式的调味品盒，而且这个盒子抽拉时比较牢靠，在盒子背部还有已经开好的几个孔可以方便我们在后面安装感应开关。于是我们暂时就放下我们对药盒避光性的要求，最后选择了这一款的盒子作为我们的药盒。

### 药盒的制作

- ♣ 想法一：盒子的储纳间抽拉时还是不够稳固，储纳间很容易就会掉下。所以我们把所有储纳间抽出来后，在盒子每格两侧用双面胶粘上一些楔形海绵起到防滑作用，加入后效果不错。
- ♣ 想法二：感应开关一开始买的是清零开关，尝试之后发现需要的按键力度太大且不方便安装。后又去购买了自锁开关，因为相比清零开关自锁开关按键更方便一些且正好可以插在已有的开孔处。在用万用表测试了自锁开关的按下与回复状态时的引脚连通规律后我们将自锁开关插入药盒背面的孔中，将导线直接用焊锡焊到自锁开关的闭合时会连通的两个引脚上。

## 5. 系统硬件设计与实现

### 5.1 主要模块分析

#### 5.1.1 12864 LCD 模块

12864 汉字图形液晶显示模块，可显示汉字及图形，内置 8192 个中文汉字（16\*16 点阵）、128 个字符（8\*16 点阵）及 64\*256 点阵显示 RAM（GDRAM）。

#### 功能特点：

- ♣ 1. 12864 显示内容 128x64 点阵，点大小 0.50x0.50mm<sup>2</sup>，点间距 0.05mm；
- ♣ 2. 内带 8192 中文汉字字库液晶显示模块，串行/并行两用接口；
- ♣ 3. 显示类型：STN 黄绿模式，6:00 视角，正向显示；STN 蓝模式（蓝底白字），6:00 视角，负向显示；FSTN 黑白模式（白底黑字，蓝底黑字），6:00 视角，正向显示；
- ♣ 4. 工作电压：3V 或 5V，不含背光工作电流：3V/1.2mA、5V/2 mA；
- ♣ 5. 工作温度：-20℃~70℃，储存温度：-30℃~80℃；
- ♣ 6. 控制器 ST7920，芯片封装 COB
- ♣ 7. LED 背光：黄绿、蓝、白背光。
- ♣ 8. 多种软件功能：光标显示、画面移位、自定义字符、睡眠模式等。

## 模块引脚说明

### 12864 引脚说明

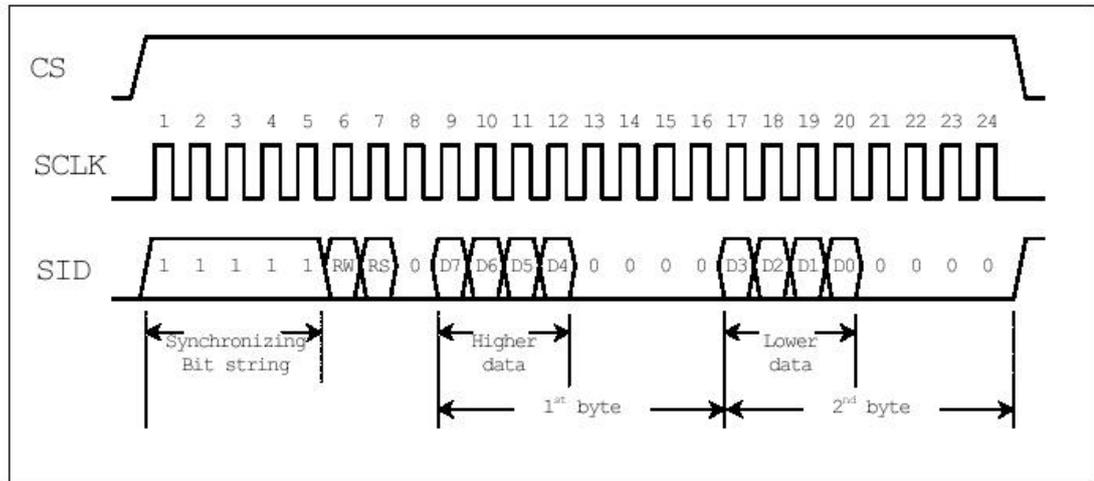
引脚号	引脚名称	方向	功能说明
1	VSS	-	模块的电源地
2	VDD	-	模块的电源正端
3	V0	-	LCD 驱动电压输入端
4	RS (CS)	H/L	并行的指令/数据选择信号串行的片选信号
5	R/W (SID)	H/L	并行的读写选择信号串行的数据口
6	E (CLK)	H/L	并行的使能信号 串行的同步时钟
7	DB0	H/L	数据 0
8	DB1	H/L	数据 1
9	DB2	H/L	数据 2
10	DB3	H/L	数据 3
11	DB4	H/L	数据 4
12	DB5	H/L	数据 5
13	DB6	H/L	数据 6
14	DB7	H/L	数据 7
15	PSB	H/L	并/串行接口选择 H-并行 L-串行
16	NC		空脚
17	/RET	H/L	复位 低电平有效
18	NC		空脚
19	LED_A	-	背光源正极 LED+5V
20	LED_K	-	背光源负极 LED-0V

逻辑工作电压 (VDD) 4.5 5.5V

电源地 (GND) 0V

工作温度 (Ta) 0 60 (常温) / -20 75 宽温

## 接口时序



串行连接时序图

串行数据传送共分三个字节完成

第一字节 串口控制—格式 11111ABC

A 为数据传送方向控制 H 表示数据从 LCD 到 MCU L 表示数据从 MCU 到 LCD

B 为数据类型选择 H 表示数据是显示数据 L 表示数据是控制指令

C 固定为 0

第二字节 (并行)8 位数据的高 4 位—格式 DDDD0000

第三字节 (并行)8 位数据的低 4 位—格式 0000DDDD

### 5.1.2 MSP430F149 最小系统

Ti 公司的 MSP430 系列是一个 16 位的功能强大的超低功耗类型微控制器，该系列将大量的外围模块整合到片内，也特别适合于设计单片系统，同时 MSP430 有丰富的不同类型器件可供选择，给设计者带来很大的灵活性。

#### 功能特点：

- ♣ 1. 超低功耗：MSP430 系列单片机的电源电压采用 1.8~3.6V 低电压，RAM 数据保持方式下耗电仅 0.1uA，活动模式耗电 250pA / MIPS(MIPS: 每秒百万条指令数)，IO 输入端口的漏电流最大仅 50nA，远低于其他系列单片机(一般为 1~10uA)。
- ♣ 2. 采用矢量中断：MSP430 系列单片机采用矢量中断，支持十多个中断源，并可以任意嵌套。用中断请求将 CPU 唤醒只要 6us，通过合理编程，既以降低系统功耗，又可以对外部事件请求作出快速响应。
- ♣ 3. 处理能力强大：MSP430 系列单片机是 16 位单片机，采用了目前流行的、颇受学术界好评的精简指令集(RISC)结构，一个时钟周期可以执行一条指令(传统的 MCS51 单片机要 12 个时钟周期才可以执行一条指令)，使 MSP430 在 8MHz 晶

振工作时，指令速度可达 8MIPS。

- ♣ 4. 高性能模拟技术及丰富的片上外围模块: MSP430 系列单片机结合 TI 的高性能模拟技术,各成员都集成了较丰富的片内外设。视型号不同可能组合有以下功能模块:看门狗(WDT),模拟比较器 A,定时器 A(Timer\_A),定时器 B(Timer\_B),串口 0、1(USART0、1),硬件乘法器,液晶驱动器,10 位,12 位,14 位 ADC,12 位 DAC,12C 总线,直接数据存取(DMA),端口 1-6(P1-P6),基本定时器(Basic Timer)等。
- ♣ 5. 系统工作稳定:上电复位后,首先由 DCO\_CLK 启动 CPU,以保证程序从正确的位置开始执行,保证晶体振荡器有足够的起振及稳定时间。然后软件可设置适当的寄存器的控制位来确定最后的系统时钟频率。如果晶体振荡器在用做 CPU 时钟 MCLK 时发生故障,DCO 会自动启动,以保证系统正常工作。这种结构和运行机制,在目前各系列单片机中是绝无仅有的。另外,MSP430 系列单片机均为工业级器件,运行环境温度为-40~+85°C,运行稳定、可靠性高,所设计的产品适用于各种民用和工业环境。
- ♣ 6. 目前 MSP430 系列有 OTF 型、FLASH 型和 ROM 型 3 种类型的器件,对于 FLASH 型则有十分方便的开发调试环境。因为器件片内有 JTAG 调试接口,还有可电擦写的 FLASH 存储器,因此采用先通过 JTAG 接口下载程序到 FLASH 内,再由 JTAG 接口控制程序运行、读取片内 CPU 状态,以及存储器内容等信息供设计者调试,整个开发(编译、调试)都可以在同一个软件集成环境中进行。这种方式只需要一台 PC 机和一个 JTAG 调试器,而不需要专用仿真器和编程器。开发语言有汇编语言和 C 语言。目前较好的软件开发工具是 IARWORKBENCH V3.10。这种以 FLASH 技术、JTAG 调试、集成开发环境结合的开发方式,具有方便、廉价、实用等优点,在单片机开发中还较为少见。其他系列单片机的开发一般均需要专用的仿真器或编程器。

## MSP430F149 原理图



MSP430F149 原理图

### 5.1.3 BC04 蓝牙模块

BLK-MD-BC04-B 蓝牙模块采用英国 CSR 公司 BlueCore4-Ext 芯片，遵循 V2.1+EDR 蓝牙规范。本模块支持 UART, USB, SPI, PCM, SPDIF 等接口，并支持 SPP 蓝牙串口协议，具有成本低、体积小、功耗低、收发灵敏性高等优点，只需配备少许的外围元件就能实现其强大功能。

#### 功能特点：

- ♣ 蓝牙 V2.1+EDR
- ♣ 蓝牙 Class 2
- ♣ 内置 PCB 射频天线

- ♣ 内置 8Mbit Flash
- ♣ 支持 SPI 编程接口
- ♣ 支持 UART,USB,SPI,PCM 等接口
- ♣ 支持主从一体
- ♣ 支持软硬件控制主从模块
- ♣ 3.3V 电源
- ♣ 支持连接 7 个从设备
- ♣ 通过 REACH、ROHS 认证

功能框图:

BC04 功能框图

应用电路图：

BC04应用电路图

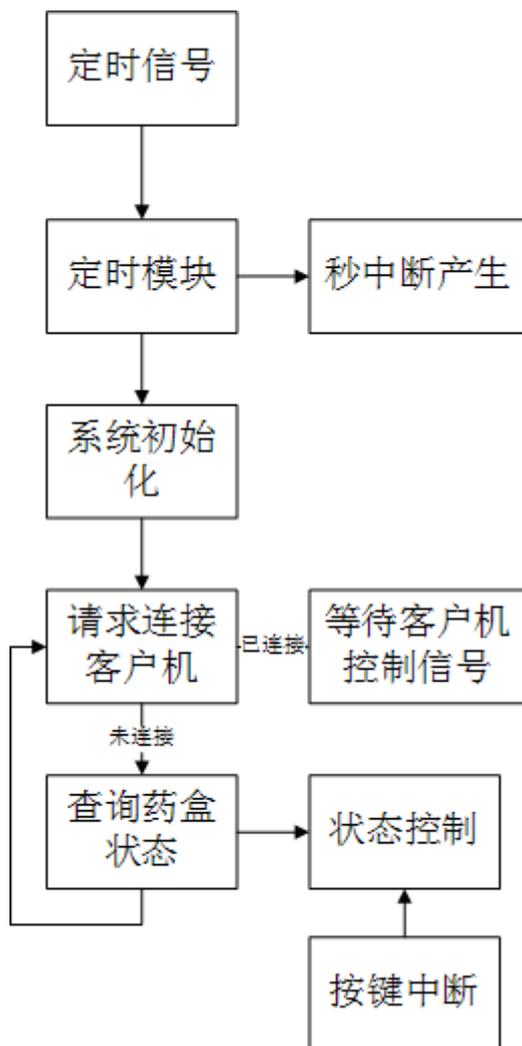
管脚功能描述：

管脚号	名称	类型	功能描述
1	UART-TX	CMOS输出	串口数据输出
2	UART-RX	CMOS输入	串口数据输入
3	UART-CTS	CMOS输入	串口清除发送
4	UART-RTS	CMOS输出	串口请求发送
5	PCM-CLK	双向	PCM时钟
6	PCM-OUT	CMOS输出	PCM数据输出
7	PCM-IN	CMOS输入	PCM数据输入
8	PCM-SYNC	双向	PCM数据同步
9	AIO(0)	双向	可编程模拟输入输出口
10	AIO(1)	双向	可编程模拟输入输出口
11	RESETB	CMOS输入	复位/重启键（低电平复位）

12	3.3V	电源输入	+3.3V电源
13	GND	地	地
14	NC	输出	NC（请悬空）

15	USB-DN	双向	USB数据负
16	SPI-CSB	CMOS输入	SPI片选口
17	SPI-MOSI	CMOS输入	SPI数据输入
18	SPI-MISO	CMOS输出	SPI数据输出
19	SPI-CLK	CMOS输入	SPI时钟口
20	USB-DP	双向	USB数据正
21	GND	地	地
22	GND	地	地
23	PIO(0)	双向	可编程输入/输出口(0)
24	PIO(1)	输出	状态指示LED口
25	PIO(2)	输出	主机中断指示口
26	PIO(3)	输入	记忆清除键(短按)，恢复默认值按键(长按3s)
27	PIO(4)	输入	软/硬件主从设置口；置低(或悬空)为硬件设置主从模式，置高电平3.3V为软件设置主从模式
29	PIO(6)	双向	可编程输入/输出口(6)
30	PIO(7)	双向	可编程输入/输出口(7)
31	PIO(8)	双向	可编程输入/输出口(8)
32	PIO(9)	双向	可编程输入/输出口(9)
33	PIO(10)	双向	可编程输入/输出口(10)
34	PIO(11)	双向	可编程输入/输出口(11)

## 6. 软件设计方案



### 1. 定时模块:

定时采用单片机自带的计时器 TIMERA 实现。其中 TIMERA 采用增计数模式。通过设置 CCRO 来确定 TIMERA 产生中断的周期。TIMERA 在本程序中每秒种产生一次中断，用以确定整个系统的时间。

### 2. 显示模块:

本系统显示信息采用 12864 点阵液晶显示器实现。其中 12864 自带了相关的汉子字库，因此向其输入汉子显示比较容易，只需调用相关的 DISPCGROM 函数直接向其输入汉子即可显示且显示速度较快。图片显示需要先打开其中的特殊指令模式。然后向屏幕输入需要绘图的地址起始和结束的位置。并一位位地向其输入点阵命令来绘制图片。因为输入命令的时候显示器无法关闭，因此可在显示器上面观察到横向的屏幕刷新消息。每次绘图大约需要 12S。

### 3. GPIO 控制模块。

因为需要扩展可控制的外设，需要向 GPIO 口输入相应的信息以控制单片机的行为。

GPIO 控制的调用主要步骤为设置使用的端口，打开使用的端口，设置端口的方向，向端口输入控制信息这几步。在主程序中只需调用控制 GPIO 口的函数程序即可自动向 GPIO 口输入信息。

#### 4. GPIO 读取模块。

因本程序的 GPIO 口读取均是按键读取。因此按键读取模块需要添加相关的去抖动函数。在本程序中去抖动采用延时判断的方式实现。即先读取相应 GPIO 口的状态信息，然后延时一定时间再次读取同一 GPIO 口的状态信息。如果经过比较两者相等。则输出该信息供程序使用。

#### 5. UART 通信模块

UART 通信采用预定义波特率的方式通过中断方式进行接收，查询方式发送数据来完成。数据发送底层采用单字节发送的方式逐次发送数据。在顶层采用调用底层发送函数通过循环发送的方式实现一整句话的发送。其中程序需要调用通信模块的时候需要先用 `sprintf` 函数将数据转换成数据串的形式并存在程序 `BUFFER` 中然后在调用语句发送模块实现句子的收发。

#### 6. 蓝牙无线通信模块

蓝牙无线通信也采用串口的方式来实现。主要涉及到对蓝牙模块状态的中断控制与中断接受和对其消息的清除与模块指令的发送。本模块需要调用 GPIO 控制模块和 GPIO 接收模块来协调完成。

#### 7. 按键采集模块

本模块主要用来判定药盒的开关是否属于开启的状态。因此按键采集仅需调用 GPIO 状态读取模块即可实现。本模块的作用主要是输出已开启的药盒序号供主程序判定使用。

#### 8. 主函数

主函数作为本单片机的核心函数主要采用调用各个模块的方式来实现。进入主函数的时候函数会调用初始化函数来初始化所有的单片机模块。之后会开启总中断并利用中断的方式实现通信与开关状态判定。通过 `WHILE(1)` 循环对药盒的按键循环查询，用于判断药盒的状态并根据药盒的状态调用相应的处理模块来实现对药盒整体功能的控制。

### 7. PC 端软件设计

#### 1. 串口通信定义

串行接口是一种可以将接受来自 CPU 的并行数据字符转换为连续的串行数据流发送出去，同时可将接受的串行数据流转换为并行的数据字符供给 CPU 的器件。一般完成这种功能的电路，我们称为串行接口电路。

## 2. 串口通信原理

串口通信 (Serial Communications) 的概念非常简单, 串口按位 (bit) 发送和接收字节。尽管比按字节 (byte) 的并行通信慢, 但是串口可以在使用一根线发送数据的同时用另一根线接收数据。它很简单并且能够实现远距离通信。比如 IEEE488 定义并行通行状态时, 规定设备线总长不得超过 20 米, 并且任意两个设备间的长度不得超过 2 米; 而对于串口而言, 长度可达 1200 米。典型地, 串口用于 ASCII 码字符的传输。通信使用 3 根线完成, 分别是地线、发送、接收。由于串口通信是异步的, 端口能够在 1 根线上发送数据同时在另一根线上接收数据。其他线用于握手, 但不是必须的。串口通信最重要的参数是波特率、数据位、停止位和奇偶校验。对于两个进行通信的端口, 这些参数必须匹配。

a, 波特率: 这是一个衡量通信速度的参数。它表示每秒钟传送的 bit 的个数。例如 300 波特表示每秒钟发送 300 个 bit。当我们提到时钟周期时, 我们就是指波特率例如如果协议需要 4800 波特率, 那么时钟是 4800 Hz。这意味着串口通信在数据线上的采样率为 4800 Hz。通常电话线的波特率为 14400, 28800 和 36600。波特率可以远远大于这些值, 但是波特率和距离成反比。高波特率常常用于放置的很近的仪器间的通信, 典型的例子就是 GPIB 设备的通信。

b, 数据位: 这是衡量通信中实际数据位的参数。当计算机发送一个信息包, 实际的数据不会是 8 位的, 标准的值是 5、7 和 8 位。如何设置取决于你想传送的信息。比如, 标准的 ASCII 码是 0~127 (7 位)。扩展的 ASCII 码是 0~255 (8 位)。如果数据使用简单的文本 (标准 ASCII 码), 那么每个数据包使用 7 位数据。每个包是指一个字节, 包括开始/停止位, 数据位和奇偶校验位。由于实际数据位取决于通信协议的选取, 术语“包”指任何通信的情况。

c, 停止位: 用于表示单个包的最后一位。典型的值为 1, 1.5 和 2 位。由于数据是在传输线上定时的, 并且每一个设备有其自己的时钟, 很可能在通信中两台设备间出现了小小的不同步。因此停止位不仅仅是表示传输的结束, 并且提供计算机校正时钟同步的机会。适用于停止位的位数越多, 不同时钟同步的容忍程度越大, 但是数据传输率同时也越慢。

d, 奇偶校验位: 在串口通信中一种简单的检错方式。有四种检错方式: 偶、奇、高和低。当然没有校验位也是可以的。对于偶和奇校验的情况, 串口会设置校验位 (数据位后面的一位), 用一个值确保传输的数据有偶个或者奇个逻辑高位。例如, 如果数据是 011, 那么对于偶校验, 校验位为 0, 保证逻辑高的位数是偶数个。如果是奇校验, 校验位为 1, 这样就有 3 个逻辑高位。高位和低位不真正的检查数据, 简单置位逻辑高或者逻辑低校验。这样使得接收设备能够知道一个位的状态, 有机会判断是否有噪声干扰了通信或者是否传输和接收数据是否不同步。

### 2. LabVIEW 简介

LabVIEW 是一种程序开发环境, 由美国国家仪器 (NI) 公司研制开发的, 类似于 C 和 BASIC 开发环境, 但是 LabVIEW 与其他计算机语言的显著区别是: 其他计算机语言都是采用基于文本的语言产生代码, 而 LabVIEW 使用的是图形化编辑语言 G 编写程序, 产生的程序是框图的形式。

基于 LabVIEW 的 PC 机与单片机的串口通信

### 3. 程序框图

进入框图程序设计界面, 在设计区的空白处单击鼠标右键, 显示函数选板 (Functions)。

(1) 添加一个配置串口函数: 编程 (Programming) → 仪器 I/O (Instrument I/O) → 串口 (Serial) → VISA 配置串口 (VISA Configure Serial Port)。

(2) 添加 4 个数值常量: 编程 (Programming) → 数值 (Numeric) → 数值常量 (Numeric Constant), 值分别为 4800 (波特率)、8 (数据位)、0 (校验位, 无)、1 (停止位)。

(3) 添加一个 While 循环结构: 编程 (Programming) → 结构 (Structures) → While 循

环 (While Loop)。

(4) 添加一个关闭串口函数: 编程 (Programming) → 仪器 I/O (Instrument I/O) → 串口 (Serial) → VISA 关闭 (VISA Close)。

(5) 在 While 循环结构中添加一个条件结构: 编程 (Programming) → 结构 (Structures) → 条件结构 (Case Structure)。

(6) 在条件结构中添加一个顺序结构: 编程 (Programming) → 结构 (Structures) → 层叠式顺序结构 (Stacked Sequence Structure)。

将其帧 (Frame) 设置为 4 个 (序号 0-3)。设置方法: 选中 Stacked Sequence Structures 上边框, 单击鼠标右键, 执行在后面添加帧 (Add Frame After) 选项 3 次。

(7) 在顺序结构的 Frame 0 中添加一个串口写入函数: 编程 (Programming) → 仪器 I/O (Instrument I/O) → 串口 (Serial) → VISA 写入 (VISA Write)。

(8) 将控件“发送数据 (十六进制)”的图标拖入顺序结构的 Frame 0 中, 分别将确定按钮 (OK Button)、停止按钮 (Stop Buffon) 的图标拖入循环结构中。

(9) 将 VISA 资源名称 (VISA resource name) 函数的输出端口分别与串口配置 (VISA Configure Serial Port) 函数、串口写入 (VISA Write) 函数 (在顺序结构 Frame 0 中)、串口关闭 (VISA Close) 函数的输入端口 VISA 资源名称 (VISA resource name) 相连。

(10) 将数值常量 4800、8、0、1 分别与 VISA 配置串口 (VISA Configure Serial Port) 函数的输入端口波特率 (baud rate)、数据位 (data bits)、奇偶 (parity)、停止位 (stop bits) 相连。

(11) 右键选择循环结构的条件端子 , 设置为“真时停止 (Stop if True)”, 图标变为 。将停止按钮 (Stop Buffon) 与循环结构的条件端子  相连。

(12) 将确定按钮 (OK Button) 与条件结构的选择端子? 相连。

(13) 将函数“发送数据 (十六进制)”与串口写入 (VISA Write) 函数的输入端口写入缓冲区 (write buffer) 相连。

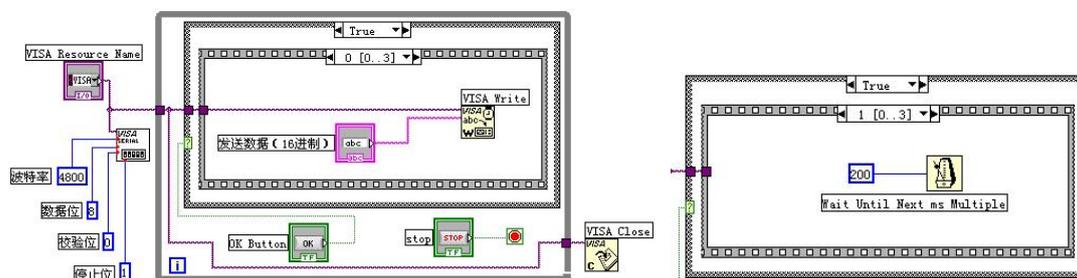
连接好的框图程序如下左图所示。

(14) 在顺序结构的 Frame 1 中添加一个时钟函数: 编程 (Programming) → 定时 (Timing) → 等待下一个整数倍毫秒 (Wait Until Next ms Multiple)。

(15) 在顺序结构的 Frame 1 中添加一个数值常量: 编程 (Programming) → 数值 (Numeric) → 数值常量 (Numeric Constant), 将值改为 200 (时钟频率值)。

(16) 在顺序结构的 Frame 1 中将数值常量 (值为 200) 与等待下一个整数倍毫秒 (Wait Until Next ms Multiple) 函数的输入端口毫秒倍数 (millisecond multiple) 相连。

连接好的框图程序如下右图所示。



框图程序连线 1

框图程序连线 2

(17) 在顺序结构的 Frame 2 中, 添加一个串口字节数函数: 编程 (Programming) → 仪器 I/O (Instrument I/O) → 串口 (Serial) → VISA 串口字节数 (VISA Bytes at Serial Port), 标签为“Property Node”。

(18) 在顺序结构的 Frame 2 中, 添加一个串口读取函数: 编程 (Programming) → 仪器 I/O

(Instrument I/O) → 串口 (Serial) → VISA 读取 (VISA Read)。

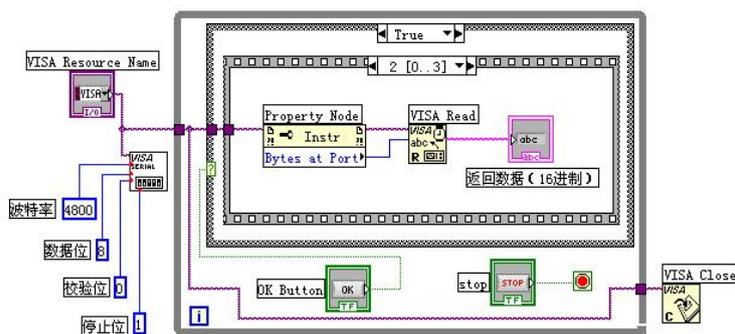
(19) 将控件“返回数据 (十六进制)”的图标拖入顺序结构的 Frame 2 中。

(20) 将 VISA 串口字节数 (VISA Bytes at Serial Port) 函数的输出端口 VISA 资源名称 (VISA resource name) 与 VISA 读取 (VISA Read) 函数的输入端口 VISA 资源名称 (VISA resource name) 相连。

(21) 将 VISA 串口字节数 (VISA Bytes at Serial Port) 函数的输出端口 Number of bytes at Serial port 与串口读取 (VISA Read) 函数的输入端口字节总数 (byte count) 相连。

(22) 将 VISA 读取 (VISA Read) 函数的输出端口读取缓冲区 (read buffer) 与控件“返回数据 (十六进制)”的输入端口相连。

连接好的框图程序如下图所示。



框图程序连线 3

(23) 在顺序结构的 Frame 3 中, 添加两个局部变量: 编程 (Programming) → 结构 (Structures) → 局部变量 (Local Variable)。

选择局部变量, 单击鼠标右键, 在弹出的快捷菜单的 (Select Item) 选项下, 为局部变量分别选择对象: “返回数据 (十六进制)” 和 “发送数据 (十六进制)”, 将其读写属性设置为 “转换为读取 (Change To Read)”。

(24) 在顺序结构的 Frame 3 中, 添加一个比较函数: 编程 (Programming) → 比较 (Comparison) → 等于? (Equal ?)。

(25) 在顺序结构的 Frame 3 中, 添加一个条件结构: 编程 (Programming) → 结构 (Structures) → 条件结构 (Case Structure)。

(26) 将局部变量 “返回数据 (十六进制)” 和 “发送数据 (十六进制)” 分别与比较函数等于? (Equal ?) 的输入端口 x 和 y 相连。

(27) 将比较函数等于? (Equal ?) 的输出端口  $x=y?$  与条件结构的选择端子? 相连。

(28) 在条件结构的真 (True) 选项中, 添加一个字符串常量: 编程 (Programming) → 字符串 (String) → 字符串常量 (String Constant), 将其值改为 “通信正常!”。

(29) 将控件 “通信状态” 拖入条件结构中。

(30) 将字符串常量 “通信正常!” 与控件 “通信状态” 的输入端口相连。

(31) 在条件结构的假 (False) 选项中, 添加一个字符串常量, 将其值改为 “通信异常!”。

(32) 在条件结构的假 (False) 选项中, 添加一个局部变量, 为局部变量选择对象 “通信状态”, 属性默认为: “写”。

(33) 将字符串常量 “通信异常!” 与局部变量 “通信状态” 相连。

#### 4. 前面板设计

在前面板设计区空白处单击鼠标右键, 显示控件选板 (Controls)。

(1) 添加一个字符串输入控件: 控件 (Controls) → 新式 (Modern) → 字符串与路径 (String

& Path) → 字符串输入控件 (String Control), 将标签改为“发送数据 (十六进制)”, 在该控件上单击鼠标右键, 在弹出的快捷菜单中选择“十六进制显示 (Hex Display)”。

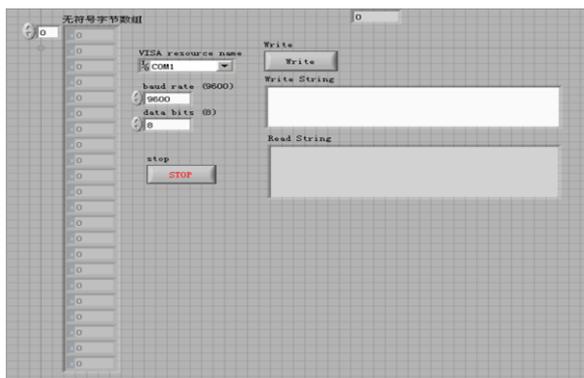
(2) 添加一个字符串显示控件: 控件 (Controls) → 新式 (Modern) → 字符串与路径 (String & Path) → 字符串显示控件 (String Indicator), 将标签改为“返回数据 (十六进制)”, 在该控件上单击鼠标右键, 在弹出的快捷菜单中选择“十六进制显示 (Hex Display)”。

(3) 添加一个字符串显示控件: 控件 (Controls) → 新式 (Modern) → 字符串与路径 (String & Path) → 字符串显示控件 (String Indicator), 将标签改为“通信状态”。

(4) 添加一个串口资源检测控件: 控件 (Controls) → 新式 (Modern) → I/O → VISA 资源名称 (VISA resource name); 单击控件箭头, 选择串口号, 如 ASRL1:或 COM1。

(5) 添加一个确定按钮控件: 控件 (Controls) → 新式 (Modern) → 布尔 (Boolean) → 确定按钮 (OK Button), 将标题改为“发送”。

(6) 添加一个停止按钮控件: 控件 (Controls) → 新式 (Modern) → 布尔 (Boolean) → 停止按钮 (Stop Button), 将标题改为“关闭”。设计的程序前面板如右图所示。



程序前面板

## 7. 系统测试与结果分析

### 1. 药盒部分的测试

药盒通过自锁开关能够实现判断抽屉是否抽出, 同时经过单片机可以被判断出为哪一编号药盒, 且同时开启2个以上药盒时出现提示。

### 2. PC 端软件部分的测试

能够实现字符串的传输, 转译的16位进制数无纰漏。

### 3. 电池盒部分的测试

4个5号电池串联, 无短路。

### 4. 电子屏与单片机的测试

能够正确显示字符和数字, 按下控制按钮后, 电子屏能显示相应正确的数据。

## 8. 设计小结与心得体会

### 1. 设计小结

在本次硬件课程设计中, 大家齐心协力, 将我们的智能药盒打造得拥有以下性能特点:

- 1) 显示药物信息 告知用户拿取的药物名称 避免出现误食
- 2) WIFI 联机更新药品信息功能 实现远程操控和更新信息
- 3) 低功耗中央控制单元 引领进入低碳时代
- 4) 药物过期提醒 一切为了用户的健康
- 5) 电源管理与低压提醒 方便用户及时更换电池 只有细心才能留住客户

## 2. 心得体会

本学期硬件课程设计这门专业实验课以 TI 杯电子设计大赛的形式，老师命题，学生可以选择老师的题目也可以自己命题，并且组队操作其他的事情（包括设计总体方案、硬件电路、软件设计、焊接、调试等工作）。趣味性强，同时也可以学到很多东西。我们认为，在这学期的实验中，在收获知识的同时，还收获了阅历，收获了成熟，在此过程中，我们通过查找大量资料，请教老师，以及不懈的努力，不仅培养了独立思考、动手操作的能力，在各种其它能力上也都有了提高。更重要的是，在实验课上，我们学会了很多学习的方法。而这是日后最实用的，真的是受益匪浅。要面对社会的挑战，只有不断的学习、实践，再学习、再实践。

本来我们组采用的单片机是 MSP430 LAUNCHPAD，但是它的端口过少，不能满足我们完成多种功能，所以我们组决定改用 MSP430F149。MSP430F149支持汇编语言和 C 语言两种语言，组长高盛最擅长的就是 C 语言，他拥有非常良好的 C 语言编程风格。不过不管怎样，这些都是一种锻炼，一种知识的积累，能力的提高。完全可以把这个当作基础东西，只有掌握了这些最基础的，才可以更进一步，取得更好的成绩。很少有人会一步登天吧。永不言弃才是最重要的。而且，这对于我们的将来也有很大的帮助。以后，不管有多苦，我想我们都能变苦为乐，找寻有趣的事情，发现其中珍贵的事情。就像中国提倡的艰苦奋斗一样，我们都可以在实验结束之后变的更加成熟，会面对需要面对的事情。与队友的合作更是一件快乐的事情，只有彼此都付出，彼此都努力维护才能将作品做的更加完美。而团队合作也是当今社会最提倡的。

单片机的选择虽然决定了，但由于为了复习备考，在时间非常紧缺的情况下，我们为了其他专业课程的考试，我们停下了手上的工作。但是，我们是不会因为学业的繁忙而放弃我们的项目，放弃我们决定参加 TI 杯的最初梦想。这因为这一点，我们觉得，不管咱们的智能药盒做得如何，它都会是我们的骄傲。因为，我们没有放弃。相信以后我们会以更加积极地态度对待我们的学习、对待我们的生活。我们的激情永远不会结束，相反，我们会更加努力，努力的去弥补自己的缺点，发展自己的优点，去充实自己，只有在了解自己的长短之后，我们会更加珍惜拥有的，更加努力的去完善它，增进它。只有不断的测试自己，挑战自己，才能拥有更多的成功和快乐！To us, happiness equals success!快乐至上，享受过程，而不是结果！认真对待每一个实验，珍惜每一分一秒，学到最多的知识和方法，锻炼自己的能力，这个是我们在硬件课程设计上学到的最重要的东西，也是以后都将受益匪浅的！

我们组成员均认为实验性的课程不仅趣味性强，而且还锻炼能力，可以学到很多东西，并记忆深刻，不会像其他科目一样，考完就会忘记。同时，在与老师和同学的交流过程中，互动学习，将知识融会贯通。通过这次课程设计，我们还明白了一个道理，那就是人在高强度的压力下，只要不被打垮，那效率会成倍的增加。好多同学抱怨时间过于紧迫，而不能完美的完成原先的计划。可是我们却不这么认为，相反，我们觉得即使有足够的时间，项目也不会比现在做得更好。因为，在时间充裕的情况下，人都会有惰性，于是大家都有些拖沓。而在最后的集训时，所有人都高度的集中，我们想，也只有在这种良好的状态下才能作出好的东西。

我们3人还在一个方面体会颇深，那就是团队合作与管理。首先，团队目标一定要明确、要数字化、有计划有步骤、有时间限制。团队目标是团队主管根据团队的实际情况而制定的通过团队拼搏能完成的目标。每当组内成员未能及时完成任务，组长都会及时提醒和督促，才使计划得以按时完成，项目进度才可更新。其次，团队合作必然有分工，不能所有人都参杂在同一任务下。每个人都会有他自己的想法，若是大家对同一任务都有涉及那么往往做不出什么成绩，因为每一个人都想按他自己的思路来行事，这样大家就会花大量的时间和精力去阐述自己的思想，说服别人同意自己的观点。第三，团队合作需要调节群众情绪。一开始，

大家老说好难实现功能之类的消极想法，弄得所有人没有积极性。后来，其他组的同学鼓励我们，说我们已经做得很不错，我们才有信心继续工作。虽然给团队树立信心的工作并不是我们组内成员完成的，但是，从其他组组员对我们的影响上我能判断出，团队的情绪是多么的重要。

## 9. 参考文献

1. 《LabVIEW 工程实践技术》 主编 付家才 化学工业出版社
2. 《LabVIEW 高级程序与虚拟仪器工程应用》 雷振山 魏丽 编著 中国铁道出版社
3. 《MSP430 系列单片机 C 语言程序设计与开发》 胡大可 编著 北京航空航天大学出版社
4. 《基于 MSP430 系列单片机的智能无功补偿控制器》 陈茂勇 郭西进 电力建设
5. 《MSP430 系列 16 位低功耗单片机原理与实践》 沈建华 杨艳琴 编著北京航空航天大学出版社
6. 《MSP430 系列超低功耗单片机及应用》 杨平 王威 国外电子测量技术