

低功耗均速管流量计部分软件设计

专业：通信工程 学生：许思炜 指导老师：王琨

摘要

随着科技水平的提升，人工抄表已经跟不上现代化建设的脚步，并且人工抄表存在着诸多隐患，要想改变这种效率不高的抄表方式，只能通过采用智能水表来实现。与传统的机械水表不同，在新一代的智能水表的研发中，普遍选择使用单片机来做中央处理器，通过各种传感器来采集数据。为了满足智能水表生产厂家和使用用户对智能水表的要求，适应节约用水、合理用水的科学理念，本文设计了一款基于 MSP430 单片机的均速管流量计的累计发送主板，实现了对数据的采集与处理，以及对数据的上传等功能。

本设计主要研究以 MSP430 单片机为主控芯片，由 TM1727 液晶驱动芯片驱动段式 LCD 液晶显示屏，在段式 LCD 液晶显示屏上显示累计流量、温度、压力、瞬时流量、电池电量、流量计工作符号、天线信号强度，通过串口向差压采集电路读取温度值、瞬时流量，采用 Modbus 通信协议由无线通信模块将数据上传。

目前水表已经从居民的屋内转移到屋外，为避免水表遭到他人的破坏，采用电池供电而不采用交流电源供电，因此电池使用时间的长短决定了产品的实用性，本文设计的均速管流量计的累计发送主板在硬件上皆选择低功耗的器件，并且通过软件设置使整个电路的功耗达到最低。

本设计实现的功能亦可用在电、气、暖三表中，最终把四个表连接到同一个网络中，加快智能小区的建设。

关键词：智能远传水表 MSP430 单片机 Modbus 通信协议

目 录

1 绪论	3
1.1 课题背景及研究意义	3
1.2 国内外发展现状	3
1.3 研究思路与主要内容	5
1.4 论文结构	5
2 系统方案的设计	6
2.1 系统的整体硬件结构	6
2.2 单片机的选择	6
2.3 无线通信模块选择	8
3 系统的软件设计	9
3.1 整个系统的总体软件设计	9
3.2 主程序设计	10
3.3 时间程序设计	11
3.4 电池电压采集程序设计	11
3.5 Flash 存储程序设计	12
3.6 显示模块	14
3.7 采集数据的读取程序设计	15
3.8 Modbus 通信协议设计	16
4 系统测试与分析	19
结论	22
参考文献	23
附录一	24

1 绪论

1.1 课题背景及研究意义

水资源对于我们的生命有着特殊的意义，它是一切生命的源泉，是生命生存所需的最重要的物质资源之一，是城市可持续发展的重要保障^[1]。节约用水、合理用水是我们所有人的责任，是避免水资源被滥用的主要途径。水表的出现，让用户与水资源之间的关系产生了衡量方式。

随着我国经济快速发展以及水表“一户一表”制的逐渐推行，传统水表的各种弊端与不足之处已经表露出来，越来越显得与城市的现代化建设不相适应^[2]。首先，入户抄表会打扰到用水用户的日常生活，有可能会侵犯到用水用户的个人隐私，同时也存在着很多的安全隐患，让犯罪分子有了可乘之机；其次，由于抄表的时间间隔相对较短，使得抄表频率较高，而且住房的增多、用水用户的增加，导致抄表工作的工作量变大、效率变低、容易出现失误；再次，由于抄表效率低，使得部分用水用户和用水单位出现拖欠水费等问题，导致自来水公司收费率无法提高、资金回收缓慢；最后，由于用水用户的增加，抄表人员和管理人员也随之增加，导致各自来水公司的支出日益增高，严重限制着各自来水公司的发展。

在电子科技逐渐发达的今日，自动化和智能化是各类仪器仪表研发的主要选择，而具有纯电子数字化计量和远程传输的智能远传水表的研究，可以促进用水单位和用水用户节约用水。与传统的机械水表和人工抄表相比，智能远传水表有以下的优点：

- (1) 不会打扰到用水用户的日常生活和正常作息，减少因抄表而引起的安全隐患；
- (2) 能够随时了解用水用户的用水量，降低偷水、漏水等问题的出现；
- (3) 可以提高抄表效率，避免漏抄、误抄的问题出现，减少水费的拖欠，加快资金的回收；
- (4) 减轻各自来水公司工作人员的工作量，加快公司的发展。

提高智能远传水表的实用性、运行可靠性、使用寿命和安全性是迫切需要解决的问题，而低功耗则是智能远传水表的推广的关键所在。为此，设计了低功耗均速管流量计，解决传统旧水表存在的偷水、漏水等问题，以及适应各种环境。

1.2 国内外发展现状

从水表行业来看，至今为止已经有 100 多年的发展情况。为了更合理的对水

资源进行利用以及控制，各国均是采用部门管理的方式进行输送，这样做也是为了防止居民浪费使用。全球淡水资源仅仅只占地球水资源的 3%还不到，而世界人口有 74 亿的庞大数目，由此更加说明了淡水资源的稀少。所以在这么多人存在的情况下，各个国家对水资源的管理极其重视。从第二次世界大战结束后，西欧和美国在战后经济发展最为迅速，但是所带来的便是以环境作为代价，其中水资源的破坏极其严重，于是在 2000 年的时候，欧盟颁发了《水资源管理框架指导方针》，使欧盟各个成员国依照此管理条例进行对水资源的管理^[3]。在这种情况下，电子水表无疑是最方便的管理工具^[4]。欧洲和北美国家在电子智能水表的市场最为集中和发达，其消费市场也是相当可观。

1.2.1 国外发展现状

电子智能水表在国外市场非常受欢迎，其原因在于管理和使用上非常的方便。国外对电子式又或者说是智能式水表的技术掌握相当成熟，这也是为什么西欧国家和北美国家是智能水表市场最为集中和发达的地方的原因之一。就目前掌握的相关信息，从《环球表计》杂志的报道上报道过这样一则信息：NB-FI (Narrow Band Fidelity, 窄带保真) 技术将会引领整个欧洲国家的市场走向物联网智能水表的发展道路。听惯了 NB-IOT (Narrow Band Internet of Things, 基于蜂窝的窄带物联网) 的技术消息，目前东欧和西欧市场在将 NB-FI 技术应用到物联网智能水表上已经有了成功的案例，这也更加说明了技术在不断地进行革新。目前国外一些国家使用的智能水表门类众多，例如美国的耐普远传水表，该水表是由美洲国家最大的水表公司美国耐普水表集团研发制造；再有就是由日本公司主力研发的远程抄表和自动检测漏水的智能式水表等等。相关消息报道，日本 NTT (Nippon Telegraph and Telephone, 日本电报电话公司) 通信公司、MIRAIT Technologies 公司、美国胜赛斯公司联合开发研制新式水表，这种水表进一步的实现了水表智能化，完成三表统一。

1.2.2 国内发展现状

我国对水表的应用和制造相比国外晚了几十年，在过去很长的一段时期内，我国的水表行业一直被外国的生产商所垄断，一直到建国以后才开始发展起来。到目前为止，我国大部分居民使用的还是传统的湿式旋翼式水表，这种水表使用寿命久、计量较为准确，但是需要大量的人力来抄读，容易出错，为了摆脱人工抄表带来的弊端，我国一直在积极研制智能水表。到目前为止，我国智能水表每年的产量和消费量将近两千万台，随着 NB-IOT 通信技术的出现，我国智能远传水表将出现飞跃式的发展。

1.3 研究思路与主要内容

根据国内外智能水表的研究进展，将段式 LCD (Liquid Crystal Display, 液晶显示器) 液晶显示屏模块、24 位低噪声温度测量模数转换器、NB-IoT 模块加入到智能水表的设计中，以保证智能水表的数据观察和数据上传与操作。基于上述需求，本课题主要是研究基于 MSP430 的低功耗均速管流量计的累计发送主板。本设计的主要研究内容包括：

(1) 通过串口向差压采集电路读取温度和流速的数据，通过 ADS1248 转换压力的差压值。

(2) 在段式 LCD 液晶显示屏上显示流速、压力、温度、累积流量等数据。

(3) 通过 NB-IoT 模块将水表的数据进行上传与操作。

(4) 能够通过串口对累计发送主板进行设定。

1.4 论文结构

本文主要分为五个章节。

第一章叙述了课题的来源和研究的意义，目前国内外的的发展情况，累计发送主板研究的思路和内容。

第二章描述了系统方案的设计，包括设计的整体硬件结构，硬件模块的选择。

第三章主要介绍了累计发送主板的软件设计，其中包含了累计发送主板的总体软件设计和各个功能模块的软件设计。

第四章讲述系统的测试与分析，其包括显示内容的测试和数据读取的测试，串口调试的操作，以及功耗的叙述。

最后为结论，总结本课题完成的工作和解决的问题。

2 系统方案的设计

2.1 系统的整体硬件结构

低功耗均速管流量计的整体设计分为两个部分，均速管流量计部分和累计发送主板部分。本设计主要设计的是低功耗均速管流量计的累计发送主板，硬件主要由 MSP430F5529 单片机最小系统、用于显示水表信息的段式 LCD 液晶显示屏、用于接收/发送信息的 NB-IoT 技术模块、用于采集压力数据的 ADS1248 组成。图 2-1 是累计发送主板的硬件框图。

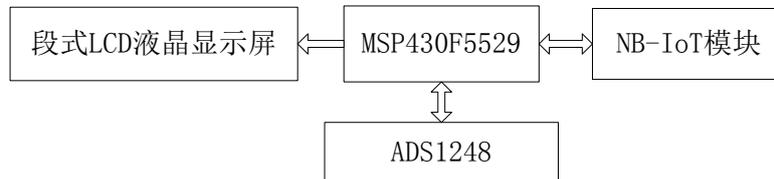


图 2-1 累计发送主板的硬件框图

2.2 单片机的选择

(1) 方案一

STC12C5A60S2 单片机是一款以 51 内核为基础的反应速度快、功耗低、抗干扰能力强的新型单片机。程序与 STC89C51 完全匹配，使用简便，其系统运行速率是 STC89C51 的 12 倍，并且程序存储空间是 STC89C51 的十几倍，比 STC89C51 多了一个串口和 SPI (Serial Peripheral Interface, 串行外设接口) 接口，拥有 STC89C51 所没有的 ADC (Analog-to-Digital Converter, 模/数转换器)。

(2) 方案二

Atmega128 是爱特梅尔公司开发的一款高配置、执行速率高、采用先进的精简指令集、功耗低的 AVR 单片机，具备了 ADC、串口和 SPI 接口等外设，具有较

高的稳定性，并且价格相对低廉，目前已经广泛地使用在仪器仪表、工业操作设备、家庭常用电器等领域。

(3) 方案三

STM32F 系列单片机是一款以 ARM Cortex - M3 为基本架构的低功耗且高性能的处理器。这款处理器不仅有大量容量的闪存存储空间，还拥有 AHB (Advanced High Performance Bus, 高级高性能总线) 和 APB (Advanced Peripheral Bus, 外围总线) 2 种总线来连接各种设备。独特的时钟控制系统和低功耗模式，让用户能够根据自己的需要进行优化。

(4) 方案四

MSP430 单片机是美国 TI (Texas Instruments, 德州仪器) 公司推出的一种基于 RISC (Reduced Instruction Set Computing, 精简指令集) 的具有超低功耗的 16 位工业级混合信号处理器^[5]。可以在 1.8-3.6V 的低电压下工作，通过在 CPU (Central Processing Unit, 中央处理器) 不运行和外部功能模块不工作的时候关闭时钟来减小功耗，并且让单片机在工作模式与睡眠模式之间切换以降低功耗^[6]。MSP430 单片机在芯片内部集成了很多的外围模块，包括用来传输的串行通信总线，常用的各类控制器和多个定时器。

STC12C5A60S2 单片机工作在 3.5-5.5V, Atmega128 单片机工作在 3.5-5.5V, STM32F 系列单片机工作在 2.0-3.6V, 由于采用电池供电，并且需要使用很长的时间，所以不考虑 STC12C5A60S2 和 Atmega128 这两款单片机。MSP430 系列单片机在工作模式下比 STM32F 系列单片机的功耗低，因此选择 MSP430 系列单片机。

MSP430F5529 是 MSP430F5XXX 系列产品的一员。该微控制器配置了集成的 USB (Universal serial Bus, 通用串行总线)，并且在物理层上支持 USB2.0，拥有四个功能各异的 16 位的定时器，一个高性能的 12 位模拟/数字转换器 (ADC)，两个 USCI (Universal serial Communication Interface, 通用的串行通信接口)，硬件乘法器，DMA (Direct Memory Access, 直接内存存取)，具有实时报警功能和时钟模块，以及 63 个 I/O (Input/Output, 输入/输出) 口引脚^[7]。图 2-2 是 MSP430F5529 的功能结构框图。

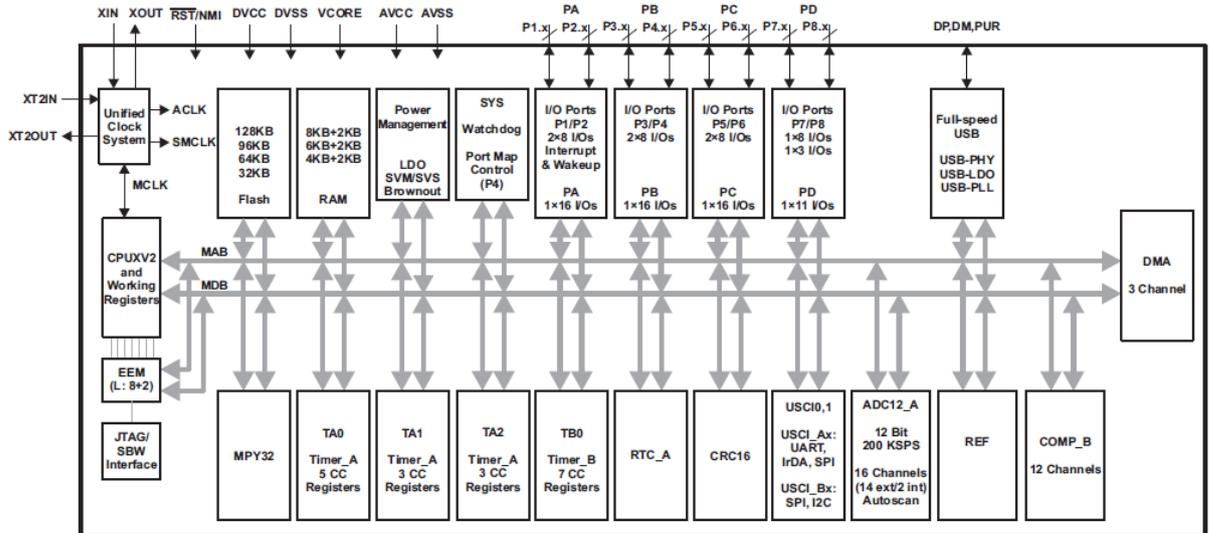


图 2-2 MSP430F5529 的功能结构框图

2.3 无线通信模块选择

(1) 方案一

GPRS (General Packet Radio Service, 通用分组无线服务) 技术是 GSM (Global System for Mobile Communication, 全球移动通信系统) 移动通信中采用的一种移动数据业务, 是对 GSM 的改进。常用的 GPRS 模块有 SIM900A 和 SIM800A 两种, 这两种模块皆可完成语音通话、短信和彩信的收发、GPRS 数据的传输, 下载速率有 85.6Kbps, 而上传速率有 42.8Kbps。

(2) 方案二

NB-IOT 技术是华为公司主推的一项新型的物联网技术。目前市场上已出现 BC95 NB-IOT 模块, 该模块目前能实现点对点的短信接收与发送功能, 支持文本和 PDU (Protocol Data Unit, 协议数据单元) 两种模式。

NB-IOT 与 GPRS 相比:

1) NB-IOT 的上传容量比 GPRS 大了 50~100 倍, 可以使连接的设备数目提升 50~100 倍, 为物联网的发展起着非常重要的作用。

2) NB-IOT 技术在通信范围上有着百倍左右的提升, 而且穿墙能力更加强大, 即使是地下停车场, 地下室、地下排水管道等地面下较深的地方也能正常的接收到信号。

3) SIM900A 模块在通信的过程中需要 2A 左右的电流, 而华为推出的 NB-IOT 芯片工作时的电流才 50mA, 并且不工作的时候可以进入睡眠模式, 只需要 4 μ A 来维持, 为采用电池供电的设备提供了更长的使用时间。

4) 在造价方面, GPRS 的芯片需要 50 块钱左右, 而华为的 NB-IOT 芯片在大规模商用时有望达到 10 块钱以下。NB-IOT 技术可以直接应用在当前网络基站上, 不需要另外建设新的基站。

5) NB-IOT 独自使用一个新的频段, 不会与其他的业务复用频段, 可以保证其稳定性。

本课题采用方案二来实现。

3 系统的软件设计

3.1 整个系统的总体软件设计

基于 MSP430F5529 的低功耗均速管流量计的累计发送主板选择按功能模块来设计的设计方式, 将每一个功能独立实现之后再联合调试, 让程序能够一目了然, 提高了程序的可读性, 后期能够对程序进行功能的扩充和更改。程序的整体设计思路为: 单片机接通电源之后, 先初始化液晶显示屏和 ADS1248, 以及单片机内部的各个模块, 然后进入低功耗工作模式, 如果中断源产生中断, 就进入相关的中断函数里面执行对应的程序, 执行完后回到低功耗模式里。图 3-1 为累计发送主板的总体软件设计流程图。

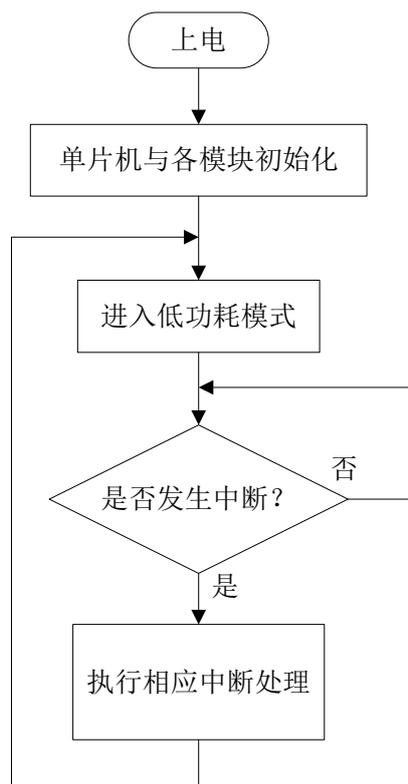


图 3-1 总体软件设计流程图

3.2 主程序设计

单片机接通电源程序复位后，主程序通过从上往下处理的方式，先初始化系统时钟、初始化 2 个串口、初始化定时器、初始化 RTC（Real Time Clock，实时时钟）、初始化 ADC12、初始化 ADS1248，然后从单片机自带的 Flash（Flash EEPROM，闪存）存储空间中获取智能水表的设置参数，对 TM1727 芯片进行初始化操作以控制液晶显示屏，打开中断的总开关，刷新液晶显示屏上的显示内容，最后进入低功耗工作模式等待中断源产生中断。因为在程序中有多个中断源，所以要提前考虑好中断的优先级和是否需要中断嵌套的问题。图 3-2 是主程序流程图。

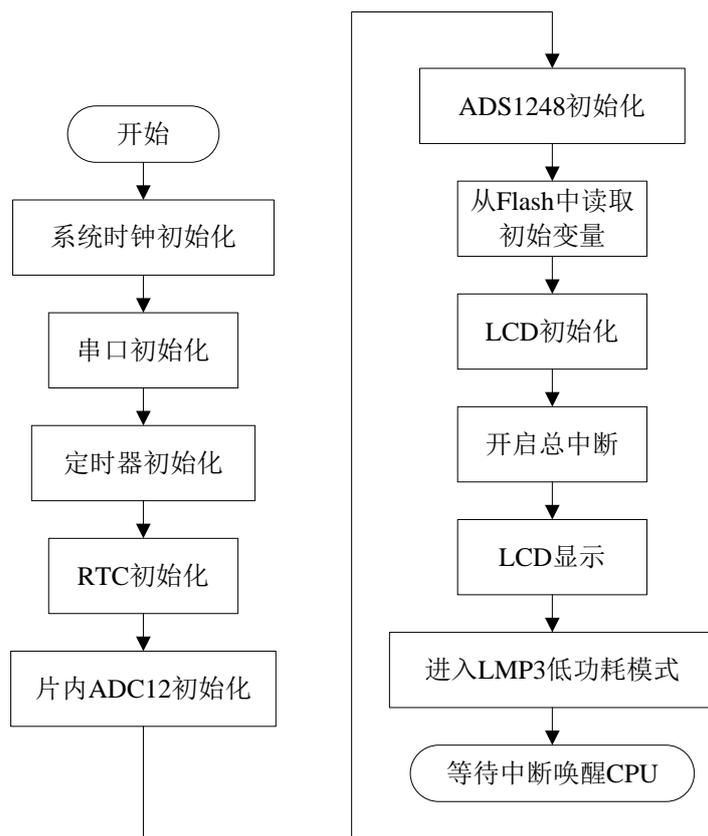


图 3-2 主程序流程图

3.3 时间程序设计

MSP430F5529 的 RTC_A 模块可以作为一个通用的 32 位计数器(计数器模式), 或者作为一个集成的实时时钟(日历模式)。在计数器模式中, RTC_A 包含了 2 个互不干扰的 8 位定时器, 能够级联形成 16 位定时器/计数器, 这 2 个定时器能够采用软件的方式来读取和写入。日历模式能够在小于 31 天的月份里自动更正日历, 并且包括闰年校正。该模块不需要设置就能够准确地提供 1s 的时间间隔, 并且提供若干个寄存器存放当前的日期和时间。当程序执行到 RTC 初始化的时候, 将预先设置的年月日时分秒写入年月日时分秒相对应的寄存器中, 当 RTC 中断的时候将年月日时分秒对应的寄存器数值读出, RTC 的年月日时分秒寄存器可以通过串口用 Modbus 通信协议指令修改。

3.4 电池电压采集程序设计

ADC12_A 模块支持快速的 12 位模数转换。该模块实现了一个 12 位 SAR

(Successive Approximation Register, 逐次逼近寄存器) 核心, 模拟输入多路复用器, 具有参考电压发生器、采样及转换所需的时序控制电路和 16 个转换结果缓冲及控制寄存器。转换结果缓存与控制寄存器器在没有任何 CPU 干预的情况下, 允许最多达 16 个独立的 ADC 采样进行转换与保存。

由于对采集到的电池电压的准确度要求不是非常高, MSP430 单片机自带的 ADC 模块就能够胜任, 因此不需要外接 ADC 芯片。在单片机接通电源完成 ADC12 的初始化之后, 立刻对电池进行电压采集, 之后每隔半个月采集一次电池的电压, 当电池电压低于 3.1V 的时候就要对电池电压过低报警标志位赋值, 并向上位机报警。图 3-3 为设计电池电压采集的流程图。

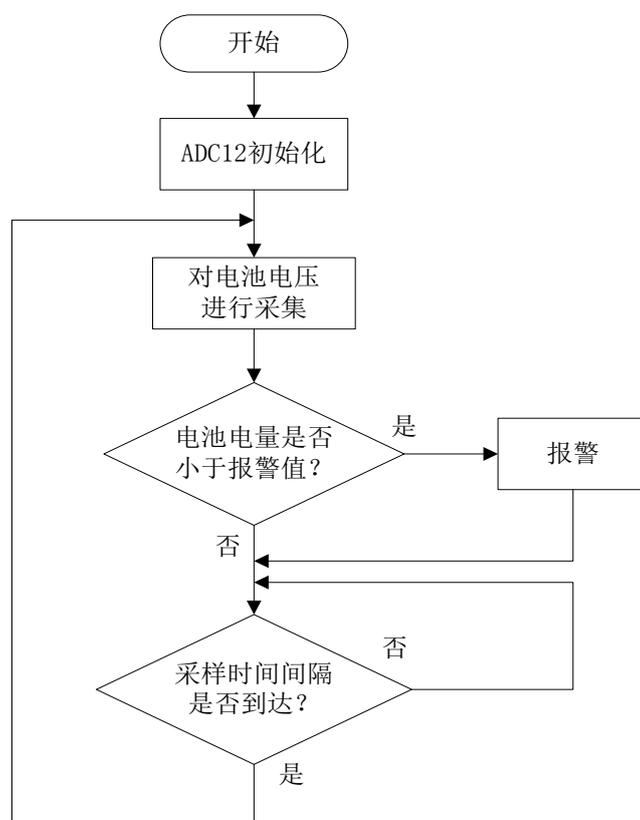


图 3-3 电池电压采集流程图

3.5 Flash 存储程序设计

MSP430FLASH 型单片机必然存在一个 Flash 存储器模块。虽然每个类型的单片机的 flash 存储空间大小不一样, 而且地址空间并不相同, 但是必然包含主存储器和信息存储器, MSP430F5529 的主存储器分为 256 个部分, 每个部分是 512 字节, 信息存储器分为 4 个部分, 每个部分是 128 字节。MSP430F5529 的 Flash

存储器可以通过 JTAG (Joint Test Action Group, 联合测试工作组) 端口、SBW (Spy-Bi-Wire, 二线制编程) 接口、BSL (Bootstrap Loader, 引导装载器) 接口或者由 CPU 在程序中编程。

CPU 可以对 Flash 存储器进行单字节、单字和长字写入操作。主存储器可以一次性全部擦除, 也可以通过一个部分一个部分的单独擦除。进行读操作的时候可以直接从地址中获取数据, 写入操作和擦除操作的时候要先对 Flash 进行解锁在执行写入操作或擦除操作。

在各个功能模块完成初始化之后, 从 Flash 中读出各个参数的初始值, 通过 RTC 中断分别存储每月数据和整点数据, 在每月 1 日存储一次当前累计的流量和温度、压力, 以及保存的时间, 在每个整点存储一次当前累计的流量和温度、压力的数值, 以及存储时间。每月数据共有 36 个月的数据, 整点数据共有 24 个小时的数据, 采用堆栈式结构, 超过溢出。这些数据可以通过串口读出。每月数据是每个月存一次, 所以使用频率不高, 当写满 36 次数据之后, 把数据保存到 RAM 中, 然后从第二个数据开始复制到原本的地址中, 把最新一次的数据写在第 36 次的地址中。整点数据是每小时保存一次数据, 因此使用频率较高, 而 Flash 存储器的典型使用寿命在一万次左右, 用 24 个部分的空间来保存数据。当写满 24 次数据的时候, 从第二次的的数据开始复制到第二个部分的地址中, 把最新的数据写到第二个部分地址的第 24 次地址中, 然后擦除第一部分的数据, 以此类推, 在 24 个部分的空间中循环保存, 来保存更多的数据。Flash 初始化的时候还要找到每月数据和整点数据下一次要写入的地址。图 3-4 是 Flash 数据存储的流程图。

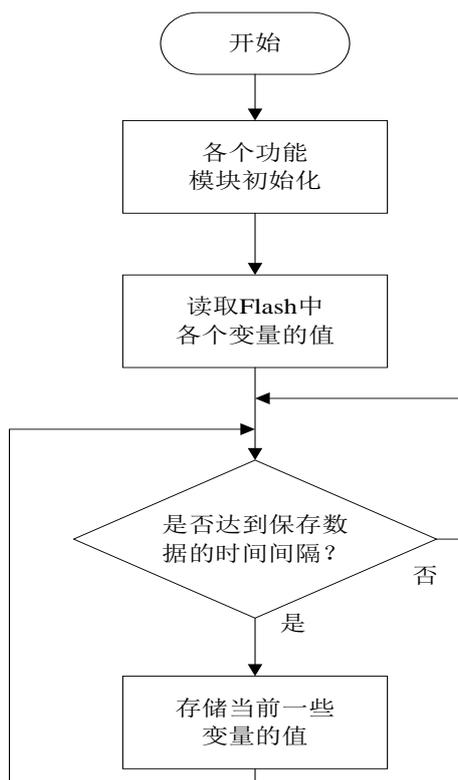


图 3-4 Flash 数据存储流程图

3.6 显示模块

目前市场上用于智能仪表的液晶显示屏有 LCD 和 LED(Light Emitting Diode, 发光二极管) 两种, 其中 LED 的功耗比 LCD 的功耗大很多, 因而采用 LCD 液晶显示屏^[8]。根据对显示需求的分析, 最终采用了段式 LCD 液晶显示屏和要显示的内容, 图 3-5 是段式 LCD 液晶显示屏的效果图, 显示参数包括:

- (1) 累计流量: 指用户使用的用水总量;
- (2) 瞬时流量: 指用户在用水时的流速;
- (3) 温度与压力: 指水表的温度和水管的水压, 温度和压力共用显示部分, 通过软件来切换;
- (4) 电池电量符号: 指电池剩余多少电量;
- (5) 天线信号强度符号: 指天线的信号强弱;
- (6) 流量计工作符号: 当用户用水的时候显示。

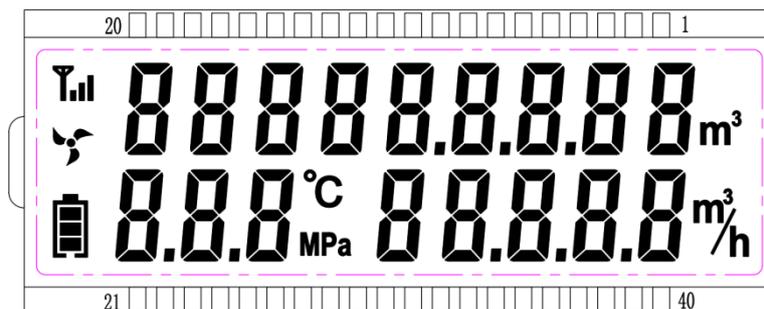


图 3-5 段式 LCD 液晶显示屏效果图

在本次设计中，采用了 TM1727 这款 LCD 显示驱动芯片使得显示操作更加便捷。该芯片里面具有单片机 2 线串行接口、数据锁存器、LCD 驱动等电路，最大能够使能 80 个段码，能够驱动 3V 和 5V 的液晶显示屏。一切的数据都采用串行总线来传输。图 3-6 为 TM1727 与段式 LCD 液晶显示屏的硬件连接电路图：

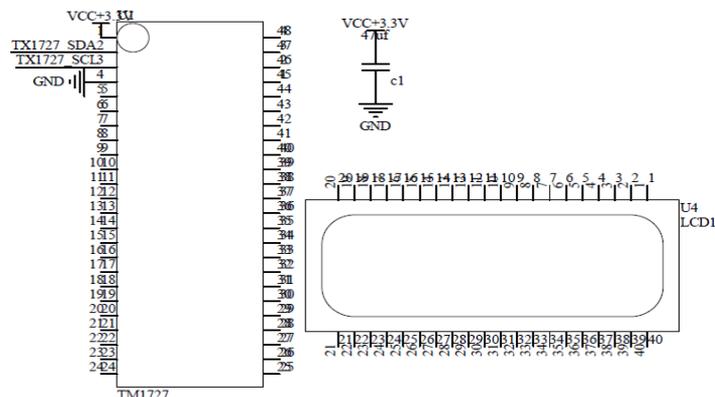


图 3-6 TM1727 与段式 LCD 液晶显示屏的硬件连接电路图

MSP430F5529 单片机本身不带 LCD 段式液晶驱动模块，TM1727 采用了低功耗 CMOS 工艺，因此通过 SDA 和 SCL 这两根串行线使 TM1727 芯片驱动液晶显示屏会比 MSP430F5529 直接驱动液晶显示屏的功耗小。当有数据要在显示屏上更新的时候，把 TM1727 从待机模式唤醒，把液晶显示屏上要显示的数据刷新之后，继续进入待机模式以降低功耗。液晶显示的过程是先发送起始信号，然后发送要写入数据的地址，等待芯片的应答，发送要写入的数据，等待芯片的应答，最后发送停止信号。

3.7 采集数据的读取程序设计

MSP430F5529 单片机的 2 个 USCI 能够配置成 UART(Universal Asynchronous Receiver/Transmitter, 通用异步收发传输器) 模式和 SPI 模式，可以同时使用 2 个 UART 和 2 个 SPI，其中一个 UART 接口采用 Modbus 通信协议 RTU (Remote

Terminal Unit, 远程终端单元) 模式向差压采样电路读取温度值和瞬时流量值, 读取到的是 4 字节浮点数, 另一个 UART 接口可以通过上位机对水表的参数进行设置, 或者读取累计流量、压力、温度、瞬时流量等数据, 通过一个 SPI 总线与 ADS1248 进行通信, 对 ADS1248 进行设置并且读取转换结果。在串口和定时器初始化完成之后, 当定时器进入中断时通过串口向差压采样电路发送读取数据的指令, 等到进入串口接收中断的时候, 把接收到的数据保存下来, 进行校验和数据处理^[9]。图 3-7 是串口读取数据的流程图。

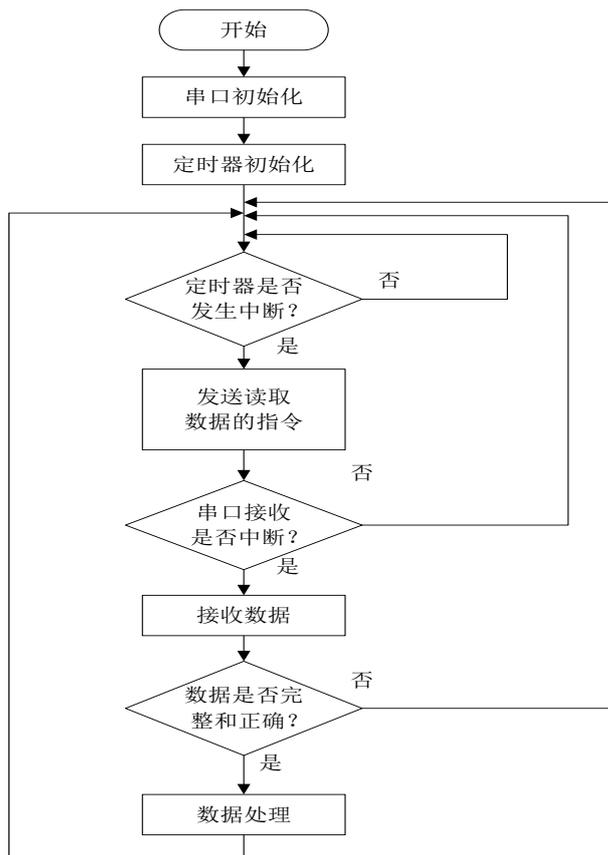


图 3-7 串口读取数据流程图

3.8 Modbus 通信协议设计

Modbus 通信协议是施耐德电气旗下的莫迪康公司创造的用来操作电子设备的一种开放的协议^[10]。通过这个协议, 可以让各个厂家生产的电子设备通过互联网互相通信, 在计算机终端上对各个设备进行统一的操作^[11]。Modbus 通信协议规范了上位机如何发送指令对电子设备进行询问, 和电子设备回复上位机的返回指令, 还有如何检测指令在通信的过程中是否出错。Modbus 通信协议提供了一些常用的操作指令。

Modbus 通信协议有 ASCII (American Standard Code for Information

Interchange, 美国信息交换标准代码) 模式和 RTU 模式 2 种传输方式, 本设计中采用的是 RTU 模式^[12]。RTU 模式在数据帧传输时不会有间隔, 在同样的串口通信速率时, 数据传送所需时间比 ASCII 模式所需的时间短。ASCII 模式采用 LRC (Longitudinal Redundancy Check, 纵向冗余校验) 方式来检测命令的正确与否, 而 RTU 模式采用 CRC (Cyclic Redundancy Code, 循环冗余校验) 方式。

以本设计所用到的读操作和写操作为例。

上位机对下位机发送的读操作指令包括: 设备地址、功能码、要读的寄存器起始地址的高位、要读的寄存器起始地址的低位、要读的寄存器个数的高位、要读的寄存器个数的低位、CRC 校验的低字节、CRC 校验的高字节。

下位机对上位机读操作指令的回复为: 设备地址、功能码、返回的数据长度 N、N 个数据、CRC 校验的低字节、CRC 校验的高字节。

上位机对下位机发送的写操作指令包括: 设备地址、功能码、要写的寄存器起始地址的高位、要写的寄存器起始地址的低位、要写的寄存器个数的高位、要写的寄存器个数的低位、要写的寄存器长度 N、N 个数据、CRC 校验的低字节、CRC 校验的高字节。

下位机对上位机写操作指令的回复为: 设备地址、功能码、要写的寄存器起始地址的高位、要写的寄存器起始地址的低位、要写的寄存器个数的高位、要写的寄存器个数的低位、CRC 校验的低字节、CRC 校验的高字节。

收到第一个数据的时候先判断是不是发给本设备的, 不是就不做响应, 是才把数据接收下来。在接收到完整的一帧数据后, 先对数据进行 CRC 校验, 如果 CRC 校验不通过, 返回 CRC 校验错误, CRC 校验通过的判断功能码是否正确, 功能码不正确的返回功能码错误, 功能码正确的判断寄存器地址和寄存器个数是否超出范围, 超出的返回寄存器地址错误, 没超出的按功能码执行对应的操作并返回。图 3-8 是 Modbus 通信协议实现的流程图。

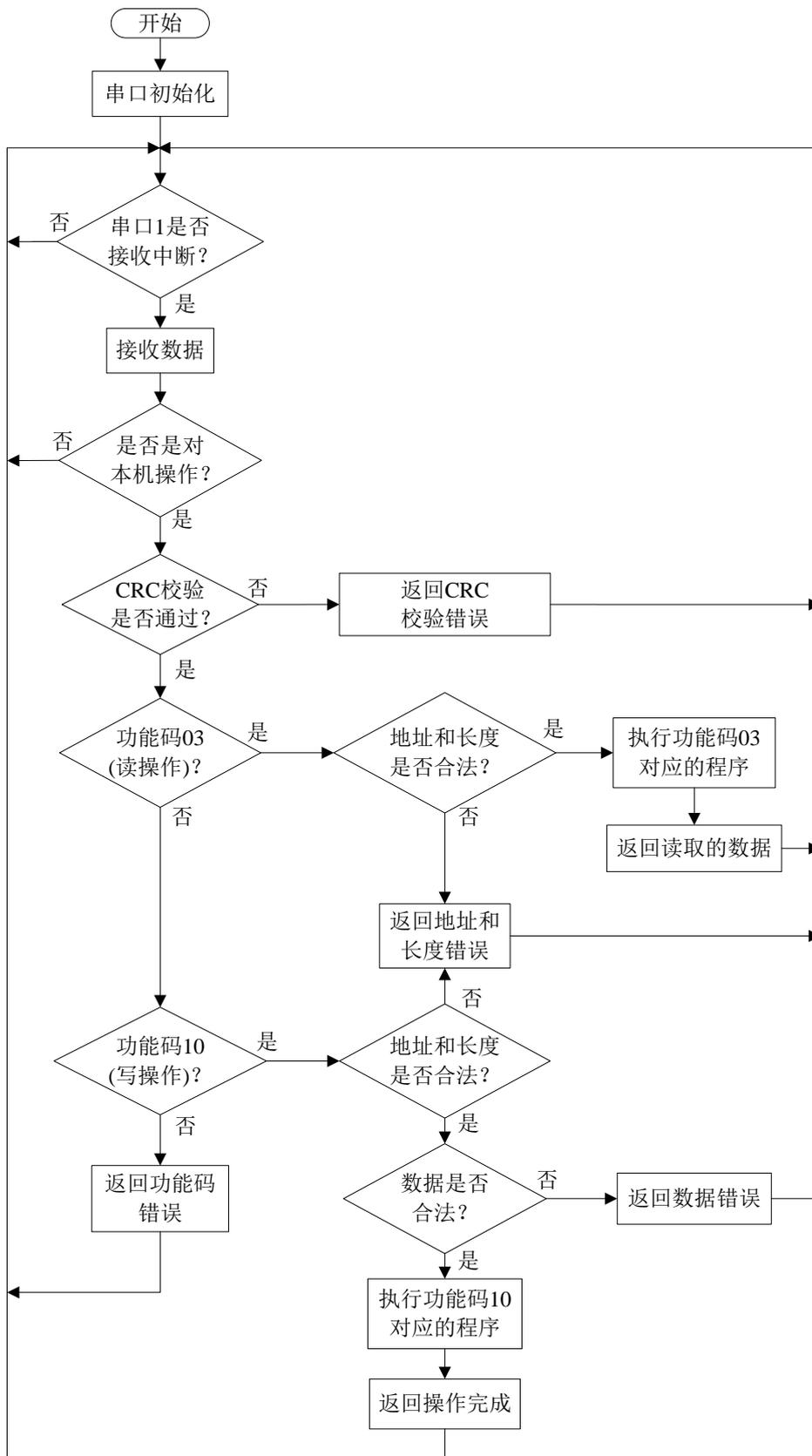


图 3-8 Modbus 通信协议实现的流程图

4 系统测试与分析

在硬件完成之后，对累计发送主板进行逐步测试。在调试的过程中用到了MSP430F5529LaunchPad 开发板，通过板上的 eZ-FET lite 模块能够实现在线仿真，为调试程序提供了很大的便利。图 4-1 是 MSP430F5529LaunchPad 开发板的实物图。

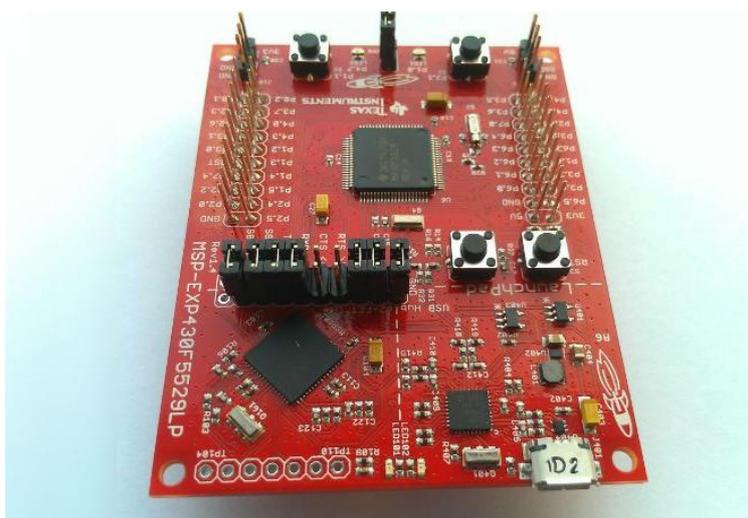


图 4-1 MSP430F5529LaunchPad 开发板

调试过程：

(1) 首先测试能否显示数据：在 LCD 液晶显示屏上能否看到累计流量、瞬时流量、温度/水压、电池电量、流量计工作符号、天线强度符号。在没有接上差压采集电路的情况下，累计流量、瞬时流量、温度/水压都应该为 0，电池电量有显示，流量计工作符号不显示。图 4-2 是 LCD 液晶显示屏显示图。



图 4-2 LCD 液晶显示屏显示图

(2) 测试能否读取到差压采集电路的数据：接上差压采集电路，看 LCD 液晶显示屏上的数值是否有变化，在瞬时流量不为 0 的情况下，流量计工作符号应该显示。图 4-3 是 LCD 液晶显示屏显示数据图。

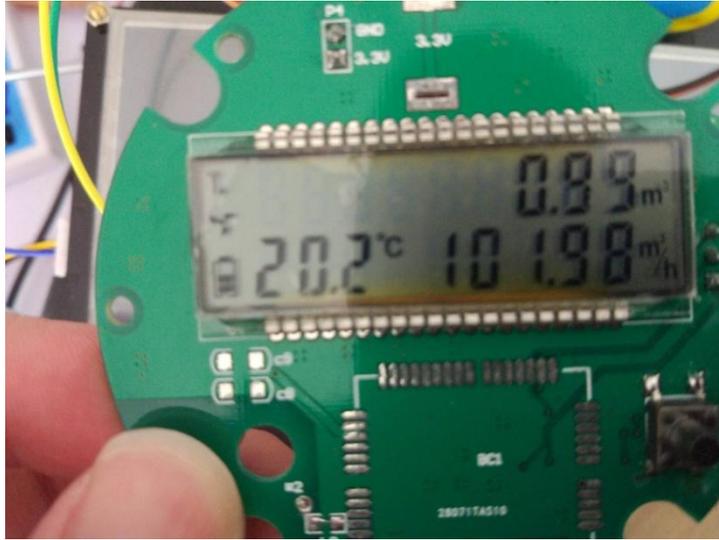


图 4-3 LCD 液晶显示屏显示数据图

(3) 测试能否通过串口进行通信：累计发送主板通过 USB 转 TTL 模块与电脑进行通信，电脑通过串口调试助手向累计发送主板发送命令，累计发送主板对命令做出回应。电脑对累计发送主板发送读指令，累计发送主板返回数据的过程如图 4-4 所示。

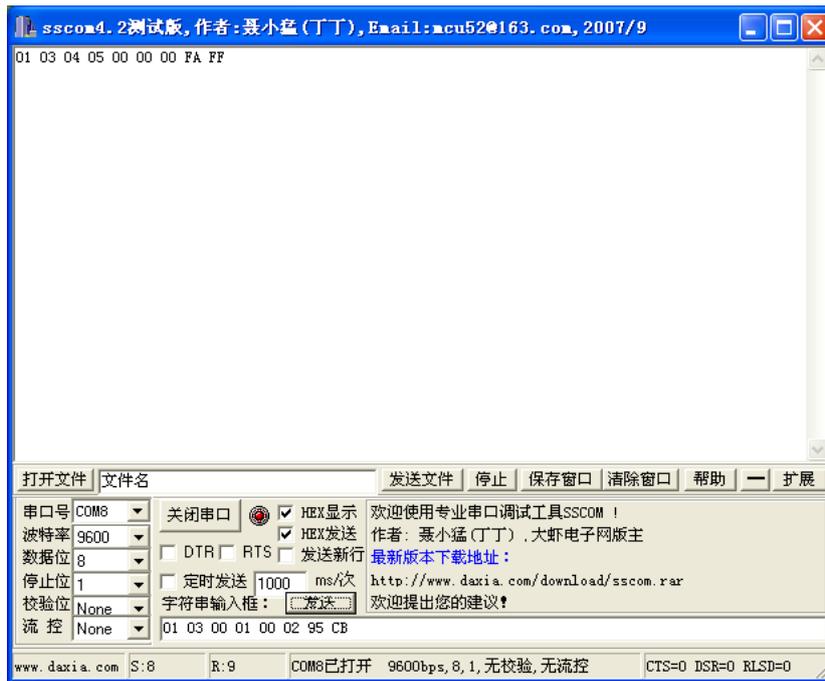


图 4-4 电脑与累计发送主板的通信图

(4) 整个电路的最小电流出现在单片机和 ADS1248 处于低功耗的时候，电流为 $347\mu\text{A}$ ，如图 4-5 所示。整个电路的最大电流出现在 ADS1248 工作的时候，ADS1248 每 10 秒工作一次，电流为 $1169\mu\text{A}$ ，如图 4-6 所示。



图 4-5 电路最小电流图



图 4-6 电路最大电流图

结论

本课题是根据生产厂家的要求所提出的设计，直接应用于工厂生产与用户使用，具有一定的研究价值。

低功耗的实现不仅是硬件上要挑选低功耗的芯片和电路，软件上也要相应的做低功耗的处理。当单片机不工作的时候，将时钟频率关掉或者选用最低需求的时钟频率，并且关掉所有没用到的外设，使单片机不工作的时候处于低电流状态。

本设计实现的功能有：

(1) 自动抄表功能：通过上位机能随时获取设备的当前工作状态和各种数据；

(2) 校时功能：设备的时钟能够通过上位机进行校准；

(3) 断电数据保护功能：当设备断电的时候，数据能够保存不消失；

(4) 设备管理功能：能够管理设备参数（如出厂编号，传感器的检定，报警参数等）；

(5) 数据管理功能：用户用水量、报警阈值等数据的初始化，用水量的统

计,用水量、水压、瞬时流速等数据的查询。

本设计的特点是:

(1) 采用上位机通过网络实现水表抄表的高度自动化,减少抄表的误抄、漏抄、安全隐患等问题,大大降低了人力、物力和财力的使用;

(2) 在发生异常情况的时候,能够主动向上位机发送报警信息;

(3) 本设计使用电池供电,并且采用了低功耗的工作方式,在不发生故障的情况下,能正常工作好几年。

智能远传水表已成为研究的热点,该设计还有一些问题需要研究解决:

(1) 找出可以继续降低功耗的地方;

(2) 随着电子科技的发展,芯片的成本会有所下降,尽可能选择低成本高性价比的芯片来完成该设计;

(3) 深入研究智能远传水表,将其特点应用在更广泛的领域内。

参考文献

- [1] 秦晓燕,李勇飞.节约用水 合理利用水资源[J].四川建筑,2007,27(6):16-17.
- [2] 袁汶雯,高锋,陈隆.水表集抄系统的低功耗设计[J].电子技术应用,2002,28(10):38-41.
- [3] 佚名.欧洲和美国水资源管理的经验[J].水利水电快报,2010,31(11):8.
- [4] Wang Jiahui,Liu Xiaodan,Zeng Lei,et al.The Design and Implementation of a Wireless Meter Reading System[C].//The Tenth International Conference on Electronic Measurement & Instruments. IEEE,2011:115-120.
- [5] 曹磊.MSP430 单片机 C 程序设计与实践[M].北京:北京航空航天大学出版社,2007:1-2.
- [6] 吴雅楠,扈国华,张士文.基于 MSP430 单片机的网络智能电子水表的设计[J].自动化仪表,2008,29(04):49-51.
- [7] 任保宏,徐科军.MSP430 单片机原理与应用[M].北京:电子工业出版社,2014:35-41.

- [8] 俞骏豪. 基于 MSP430 单片机的远传智能水表的设计与实现[D]. 河北:华北电力大学控制与计算机工程学院, 2015.
- [9] 蒋剑锋, 吴晓红. 色谱仪控制系统的 Modbus 从站设计与实现[J]. 科学技术与工程, 2012, 12(31):8344-8354.
- [10] 戈瑞. 基于现场总线的协议转换器设计与应用[D]. 湖北:华中科技大学控制科学与工程系, 2012.
- [11] Sun Gang, Wu Wenbiao, Zheng Wengang, et al. Auto-meter reading system for irrigation water based on MODBUS[J]. Transactions of the CSAE, 2008, 24(S2):76-80.
- [12] 刘紫燕, 冯亮, 詹志辉. Modbus 协议 RTU 模式与 TCP 模式的通信转换设计[J]. 科学技术与工程, 2013, 13(18):5191-5196.

附录一

部分源程序

```
#include <msp430.h>
#include <string.h>
#include "UART.h"
#include "CLOCK.h"
#include "LCD.h"
```

```

#include "RTC.h"
#include "Communication.h"
#include "ADC.h"
#include "Flash.h"
#include "ADS1248.h"
const char SendSpeed[8]={0x01,0x03,0x00,0x01,0x00,0x02,0x95,0xCB};//
发送读取流速的命令
const char SendTemp[8]={0x01,0x03,0x00,0x07,0x00,0x02,0x75,0xCA};//发
送读取温度的命令
uint Sendflag=0;//发送标记位，1为流速，0为温度
float RxValue = 0;//读取流速/温度的值
float electr = 0;//电压
uchar Wflag[1]={0xf0};//数据写入Flash中的数据头
int dFlag = 0;//压力转换完成的标志位
signed long tData[5] = {0};//多次转换压力的值
int datt=0;//压力转换次数
uint itp=0;//温度/压力显示间隔
uint adtime = 0;//电池采集间隔
uint pretime = 0;//压力采集间隔
uint speedflag=0;//水流符号标志，0为不亮，1为亮
float ADC_Result = 0;//ADC12转换的数值
uint RxNum = 0;//接收数据的长度
int len_flag = 0;//应改收到的数据长度
uchar UCA1_OFF = 0;//串口1开关，0为开，1为关
uchar UCA1_RX_Flag = 0;//接收完一条完整的命令，1为有，0为没有
uchar Flash_backup[1024] = {0};//备份到RAM中
uint FlashWriteFlag = 0;//Flash写入标志位，防止多次写入，0为未写入，1
为写入
int main(void) {
    WDTCTL = WDTPW | WDTHOLD;    // Stop watchdog timer
    IoInit();//IO口初始化
    CLOCKInt();//时钟初始化
    UARTInit();//UART串口初始化
    TimerAInit();//定时器TA初始化函数

```

```

ADCInit(); //ADC 初始化
InitSPI(); //SPI 串口初始化
InitDevice(); //ADS1248 接口初始化
RTCInit(); //RTC 初始化
FlashInit(); //FLASH 地址初始化
_EINT(); //开启中断
InitConfig(); //ads1248 配置初始化
ParaInit(); //参数初始化
while(1)
{
    electr = AdtoElectricity(ADC_Result); //电池电压计算
    if(electr<EAV_L) {EA_L_flag = 1;} //电池电压低报警
    else {EA_L_flag = 0;}
    if (dFlag)
    {
        pressurei = MeanADtoPressure2(tData, Kpre, Bpre); //压力计算
        if(pressurei>PAV_H) {PA_H_flag = 1;} //压力高报警
        else {PA_H_flag = 0;}
        if(pressurei<PAV_L) {PA_L_flag = 1;} //压力低报警
        else {PA_L_flag = 0;}
        dFlag=0;
    }
    if(disFlag==0)
    {
        if(speed>=0) //正向流速
        {
            if(count>=10000000) //累计总量达 9999999.99 归零
            {
                count = count-10000000;
            }
        }
        LCD_Display(count, temperature, pressurei, speed, 34, electr, speedflag);
        //显示数据
        speed = 0;
        speedflag = 0;
    }
}

```

```

        temperature = 0;
    }
    else//反向流速
    {

        if(countN<=-1000000)//累计总量达-999999.99 归零
        {
            countN = countN+1000000;
        }
LCD_Display(countN, temperature, pressurei, speed, 34, electr, speedflag);
//显示数据

        speed = 0;
        speedflag = 0;
        temperature = 0;
    }
}
if(UCA1_RX_Flag == 1)//收到一条完整的命令
{
    Modbus();
    UCA1_RX_Flag = 0;
}
DataManage();
__bis_SR_register(LPM3_bits + GIE); //进入低功耗
}
}

```