

中国科学技术大学

高级软件工程技术论文

基于 ARM- μ C/OS 的智能家居系统



小组成员：苏颖辉、袁、孙

陈、马

二零一零年七月十二日

摘 要

嵌入式系统 (Embedded System) 是以应用为中心, 以计算机技术为基础, 并且软硬件可剪裁, 适用于应用系统对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统。它一般由嵌入式微处理器、外围硬件设备、嵌入式操作系统以及用户的应用程序等四个部分组成, 用于实现对其他设备的控制、监视或管理等功能。

智能家居 (Smart Home) 控制系统是通过综合采用先进的计算机、通信和控制技术, 建立一个由家庭安全防护系统、网络服务系统和家庭自动化系统组成的家庭综合服务与管理集成系统, 从而实现全面的安全防护、便利的通讯网络以及舒适的居住环境的家庭住宅。家居智能化包含的内容比较多, 目前国内的产品有总线制、电力线载波、无线等方式, 在网络连接方面基本上用总线制联网、电话联网或者通过以太网方式来实现。本系统以 S3C2410 嵌入式片上系统和 μ C/OS-II 实时操作系统 (RTOS) 为基础, 实现对门禁和家电控制以及水电气的实时信息反馈。

关键字: 嵌入式系统 智能家居 实时操作系统

ABSTRACT

Embedded system, based on computer technology, is application oriented. The application of Embedded System suits for special purpose which has strict command on function, reliability, expenditure, size and power consumption. Its hardware and software can be trimmed. It usually consists of four parts, that is, embedded microprocessor, peripherals hardware devices, embedded operating system and application software. It is used for control, supervision, management, etc. Smart home control system is designed to provide a integration system, including home security, internet services and home automation. Smart home has a lot of items. The product of home market is mainly based on the bus, power wire and wireless technology. These products use bus, telephone line or Ethernet to connect the internet. The microprocessor unit of our system is S3C2410, the real time operating system(RTOS) is μ C/OS-II. This system is designed to control the access, household appliances and keep the user informed of the reading of water meter, power meter and gas meter.

Key Words: Embedded System/Smart Home/Real Time Operating System

目 录

第一章 引言.....	1
第二章 智能家居系统的工作原理.....	2
第三章 系统实现.....	3
3.1 蓝牙模块的硬件设计.....	3
3.2 蓝牙模块的协议.....	4
第四章 uC/OS-II 操作系统上的系统设计.....	7
4.1 μ C/OS-II 的移植过程.....	7
第五章 功能实现.....	9
第六章 总结.....	11
参考文献.....	12

第一章 引言

随着数字化家用电器的飞速发展,如何对家庭中相对独立的家用电器进行智能化管理和操作?这是人们非常关心的一个问题。

智能家居是以住宅为平台,利用综合布线技术、网络通信技术、安全防范技术、自动控制技术等将家居生活有关的设施集成,构建高效的住宅设施与家庭日程事务的管理系统,提升家居安全性、便利性、舒适性、艺术性,并实现环保节能的居住环境。

全球每年这个领域的市场是 500 亿美元左右,国内目前每年有近 100 亿的市场。在国内,深圳的智能家居在布线方面做得比较好,前瞻性较强,考虑电源、空调、电话、电视、网络等方面较周全,预埋智能布线的观念比较超前。北京的智能家居在考虑功能和地方风格方面做得比较好。上海做得最好的在浦东新区,它的城区规划和小区布置更符合上海这样一个商业化大都市的需求。

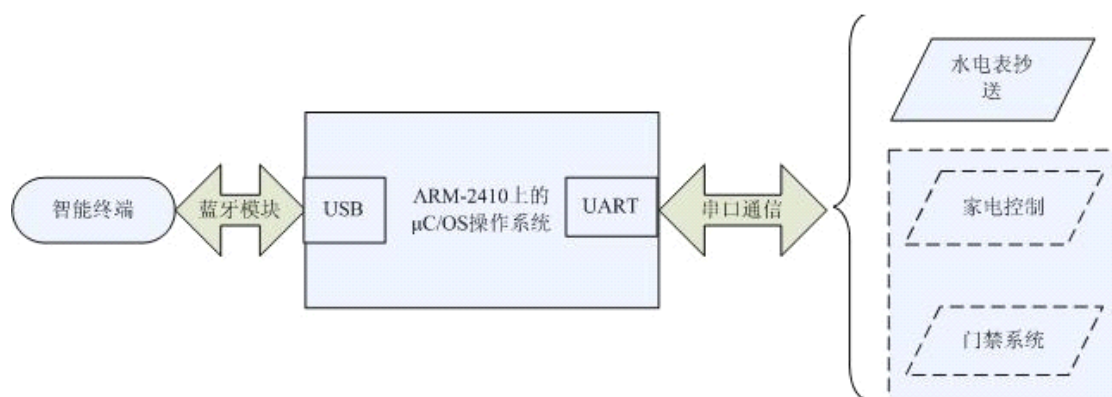
考虑到目前市场上的智能家居系统主要以有线网为主,系统受到布线的限制,本项目希望能从无线控制上突破,设计一套基于 ARM 平台和 Wi-Fi 技术的智能家居控制系统。

本项目的工作重点在于实现用户手持终端与集中控制单元的信号交互。以期能在现实中达到诸如电子门禁、电器自动控制,水、电、煤气表远程抄送等功能。

第二章 智能家居系统的工作原理

本系统是以基于 ARM 内核的 Samsung S3C2410 开发板和 uC/OS 操作系统为平台以及蓝牙技术的一个智能家居控制中心，并通过其实现水电表数据的抄送。

其体系结构如下图所示



智能家电与建立在 uC/OS 操作系统上的控制中心模块通过串口通信，控制中心通过蓝牙与手持终端之间通信。

第三章 系统实现

3.1 蓝牙模块的硬件设计

作为一种近距离无线通信技术规范的蓝牙技术因其低成本、低功耗、小体积而广泛应用于小型移动产品和信息家电产品中，而此类产品大都属于嵌入式产品。因此对用户和开发人员而言，能够快速掌握蓝牙技术在嵌入式领域中的开发方法，让蓝牙技术尽快应用于产品中，具有一定的意义。在此给出一个基于CSR（（英国剑桥硅无线电）公司的蓝牙单芯片 BC01 模块的蓝牙嵌入式开发方法的一般应用框架，读者可依此框架快速开发嵌入式蓝牙产品。

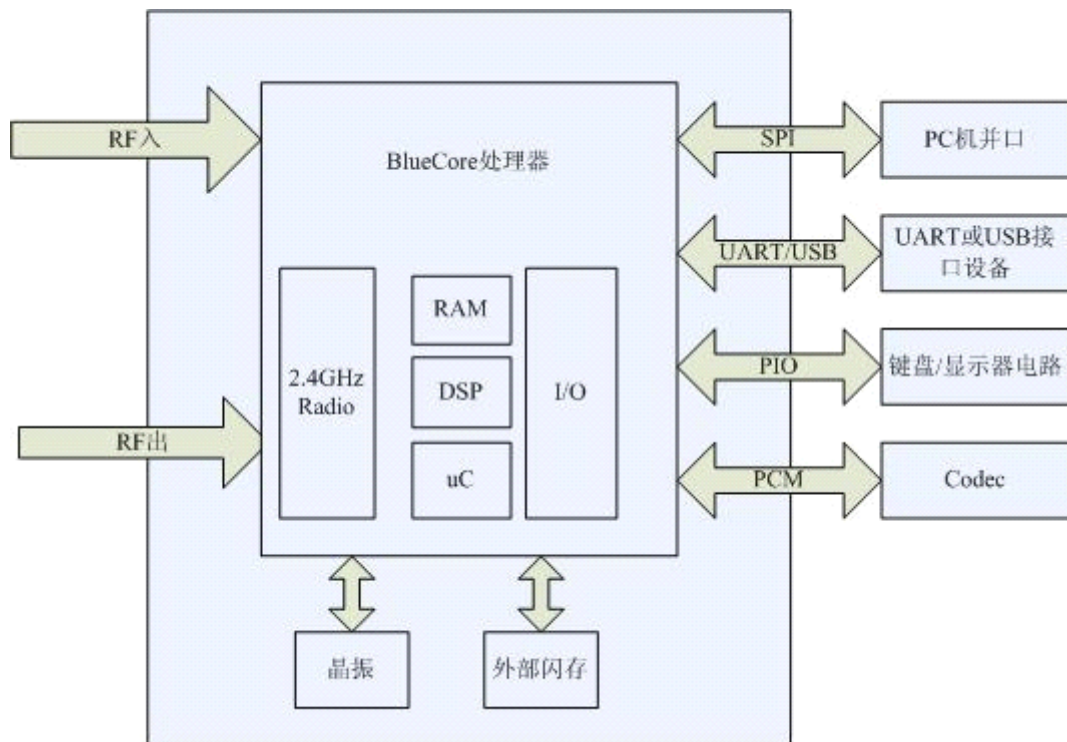


图 1. 嵌入式蓝牙系统硬件结构框图

3.1.1 嵌入式蓝牙系统硬件结构

进行蓝牙产品的开发，一般都是选用现有的蓝牙芯片，目前可选的蓝牙芯片方案有多芯片方案（RF 芯片、基带芯片和微处理器芯片）、双芯片方案（（RF 芯片和含 CPU、RAM 和 Flash 的基带芯片）、单芯片方案（将（RF、基带和 CPU、RAM 及 Flash 集成在一片 CMOS 芯片上）。嵌入式产品一般都要要求低成本、低功耗和小体积，因此单芯片蓝牙系统是最佳的选择方案。CSR

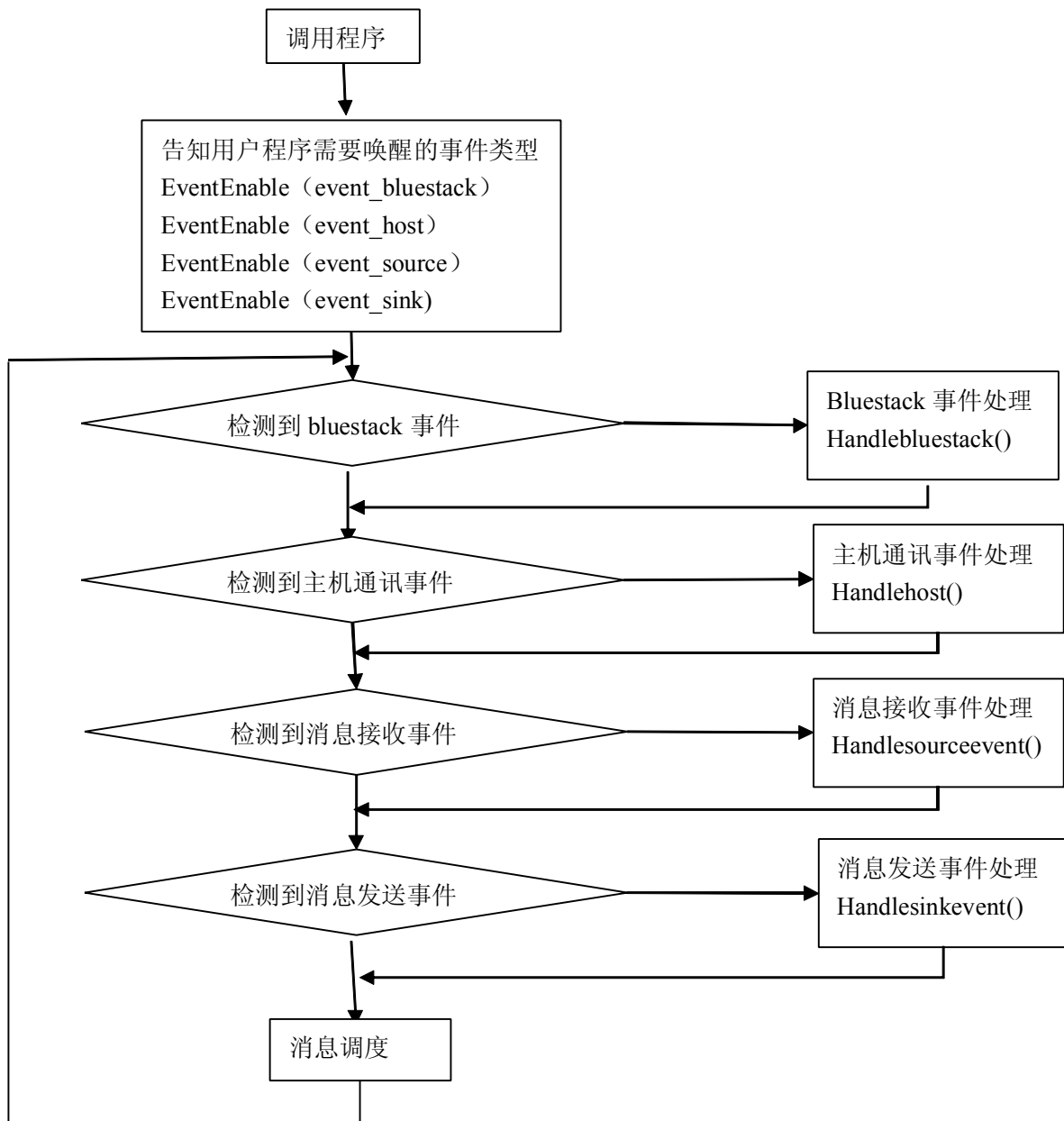
的 BLUECORE01 是蓝牙 2.4GHZ 无线技术在一片 8mm*8mm 的 CMOS 芯片上实现的单芯片无线和基带集成电路, 内有 16 位 RISC 微控制器。BC01 是蓝牙单芯片模块, 内嵌 BLUECORE01, 可将 BLUECORE 作为一个简单的嵌入控制器及蓝牙链路控制器, BC01 能运行链路管理器及以下层的协议栈及高层协议栈和应用程序。该文应用 CSR 的 BC01 蓝牙模块来构建嵌入式蓝牙产品硬件平台, 其硬件结构框图如图 01 所示。其中, 16 位的外部存储器容量高达 4Mb, 主要用于存储 BLUECORE01 设置和程序代码。USB 接口 UART 接口分别为使用 USB 和 RS232 标准与其他数据设备通讯提供一个简单的机制, USB 和 UART 接口总线是共用的, 在同一个蓝牙设备上只能选用其一接口。PCM 接口可与 MOTOROLA 的 MC145483CODEC 芯片直接连接, 用于在 BLUECORE01 和 CODEC 芯片或其他硬件之间传输双向语音通道。SPI 端口是 BLUECORE01 的串行外设接口, 提供一个对内部寄存器、RAM 和外部存储器读写的机制。通讯波特率可高达 5M, 用于装载、校验和调试 BLUECORE01 软件。并行输入输出 PIO 端口是 BLUECORE01 的通用 IO 接口。端口有 8 条可编程双向 PIO。

3.2 蓝牙模块的协议

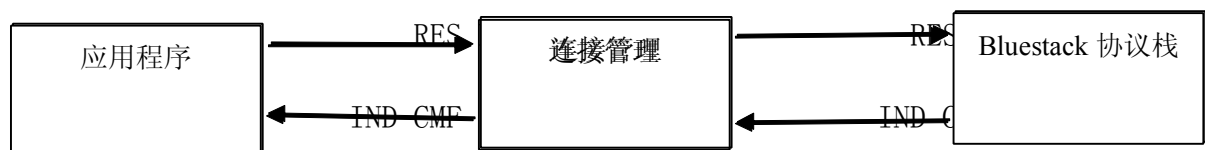
我们主要研究的协议栈包括 CSR 的 BlueCore 协议栈、CCL 的 BlueStack 栈和用户程序。BlueCore 协议栈主要完成链路管理器及以下层协议, BlueStack 主要完成包括 L2CAP、SDP 及 RFCOMM 等的高层协议栈。用户程序主要包括调度程序、消息调度程序、实现蓝牙连接建立过程的连接管理程序等通用程序及实现特定应用模型的最高层的应用程序。

在嵌入式蓝牙系统中, 调度程序检测所有事件并进行处理。CSR 提供了与事件相关的函数 EventEnable(Event_Type)。EventEnable 函数功能是当接收到信号时将应用程序从等待中唤醒。

调度程序负责管理应用程序事件的处理, 程序流程图如图所示, 此函数是一个对所有事件进行处理的无限循环程序, 应用程序只需简单地调用该函数即可。



软件各层之间的通信采用消息传递机制。每个消息包含消息类型和用户数据，消息类型分为四大类，在消息名后加上扩展名_REQ, _IND, _RES, _CMF 分别表明消息是一个请求、指示、响应还是证实。消息传递过程如图。



应用程序所需完成的功能为接口等的初始化工作、发送消息到下层软件及

对下层响应的消息处理、循环进行调度程序的执行等。

整体来说，软件实现主要包括三部分：控制器接口模块、服务发现模块和数据传输模块。

蓝牙协议体系为控制器接口模块提供了统一的访问接口，即主机控制器接口（HCI）。它为主机提供了直接控制主机控制器（蓝牙模块）的方法和途径。

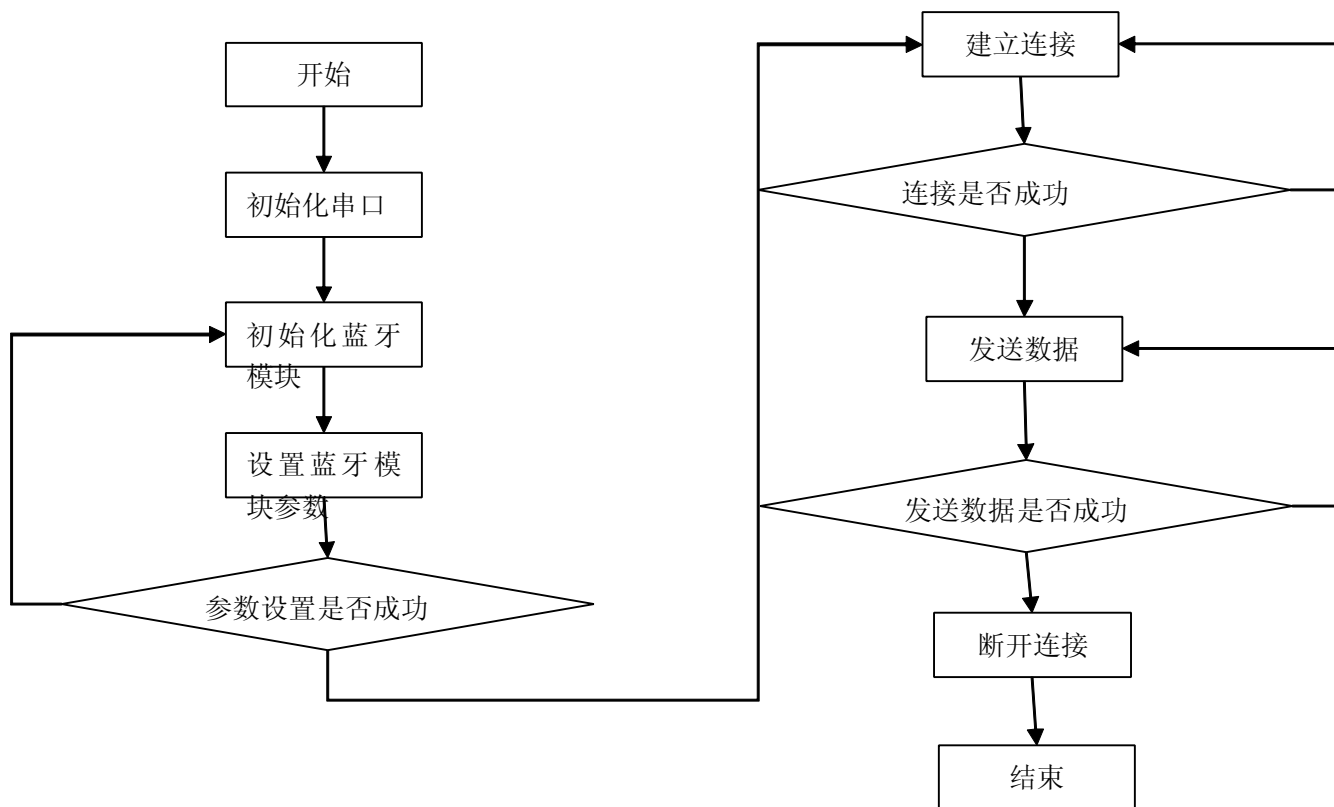
HCI 模块的程序包括位于主机上的 HCI 驱动程序和位于主机控制器（蓝牙模块）上的 HCI 固件。具体运行时由主机调用 HCI 驱动程序中提供的一系列命令来控制主机控制器，并通过主机控制器中的链路管理器发送链路管理协议分组（LMP）实现对本地端和远端蓝牙设备的管理。HCI 模块将实现串口设置，HCI 信息的封装解析传输以及异常处理等功能。

在两个蓝牙模块进行数据传输之前，需要查询远程蓝牙设备是否提供所需的服务。蓝牙规范中定义了 SDP 协议实现了查询设备信息、服务类型和服务特征。用户可以从邻近的服务中选一个可用的，然后就可以在蓝牙设备间建立相应的连接了。

其工作流程为：客服端通过主机控制接口命令搜索周围的蓝牙服务器，此时，客服端通过链路维护模块与服务器之间建立了一条 SDP 链路。当 SDP 链路建立完成后，客服端根据需要的服务向服务端发送相应的服务搜索请求 PDU，服务器收到请求 PDU 后，将此 PDU 发送至 SDP 解析服务器，解析服务器收到后与本身的服务记录表里面各服务的 UUID(Universally Unique Identifier)进行比较，如有相同，则表明其能为客服端提供所需要的服务。然后发送响应 PDU 给客服端，客服端收到响应 PDU，发现服务器能提供所需服务。此时客服端发送查询协议描述符列表属性的请求 PDU，服务器接收到请求 PDU 后将其送到解析服务器，解析服务器在服务记录中查找协议描述符列表属性值，如果有，则返回包含此属性值的响应 PDU 给客服端，没有的不用发送。至此，客服端已经通过 SDP 从服务器得到它所想要的信息。

数据收发模块的功能依次包括：（1）在两个蓝牙设备之间建立 L2CAP 信道连接；（2）配置最大传输单（MTU），清除超时（Flush Timeout）和服务质量（QoS）的参数；（3）在 L2CAP 层进行数据交换；（4）数据收发结束后，断开设备间的连接。

在传输数据时总的流程图如图



第四章 μ C/OS-II 操作系统上的系统设计

4.1 μ C/OS-II 的移植过程

虽然 μ C/OS-II 在设计时,已经考虑到了在不同的处理器上的移植问题,但在移植过程中,用户仍需修改部分代码,这些代码主要是和处理器相关的文件,即 OS_CPU.H, OS_CPU_C.C 和 OS_CPU_A.ASM。

要在 OS_CPU.H 中编写与处理器和编译器相关的代码,包括用#define 语句定义的,与处理器相关的常数、宏以及数据类型。在 OS_CPU_C.C 中,则需要进行堆栈初始化和相应函数的编写。共有 6 个函数: OSTaskStkInit (), OSSTaskCreateHook (), OSTaskDelHook (), OSTaskSwHook (), OSTaskStatHook (), OStimeTickHook ()。其中后面 5 个是 Hook 函数,又称为钩子函数,主要是用来对 μ C/OS-II 进行功能扩展的。这些钩子函数为用户定义函数,由操作系统调用相应的 Hook 函数去执行,在一般情况下,它们都可以不加代码,实现为空函数即可。而函数 OSTaskStkInit () 要对堆栈进行初始化,在进行堆栈初始化以后, OSTaskStkInit () 返回新的堆栈栈顶指针。

OS_CPU_A.ASM 中应编写 4 个函数:

OS_SStarHighRdy ();

OSCtxSw ();

OSTick_ISR ();

OSIntCtsSw ();

OS_SStarHighRdy () 被多任务系统启动函数 OS_Start () 调用,负责获取新任务的堆栈指针,并从堆栈指针中恢复新任务的所有处理器寄存器,运行优先级最高的就绪任务。

OSCtxSw () 是任务级的任务切换函数,因此,中断服务子程序、陷阱或异常处理的向量地址都必须指向 OSCtxSw ()。 OSCtxSw () 函数负责将当前任务对应的处理器的寄存器保存到堆栈中,并将任务中需要恢复的寄存器的寄存器从堆栈中恢复出来。

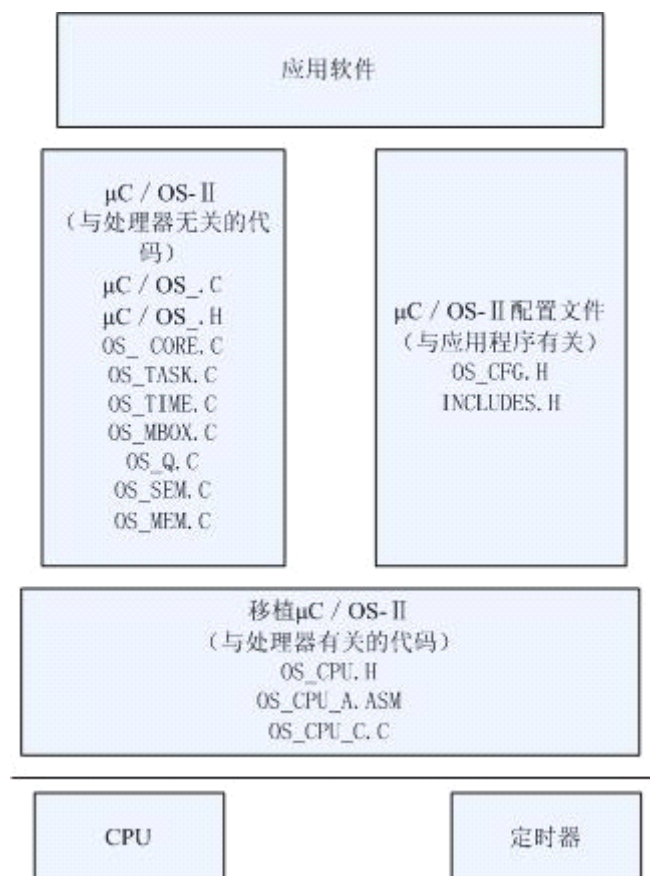
OSTick_ISR () 是系统时钟节拍中断服务函数。它是由定时中断产生,主要负责在进入时保存处理器的寄存器,完成任务时切换,退出时恢复寄存器并返回。

OSIntCtsSw () 是中断级的任务切换函数,它的绝大多数代码和 OSCtxSw () 一样,区别在于 OSTick_ISR () 中已经保存了处理

数中保存处理器的寄存器。

完成上述工作后， $\mu\text{C}/\text{OS-II}$ 就可以运行在 S3C2410A 硬件开发平台上了。

以下为 uc/os 系统结构图



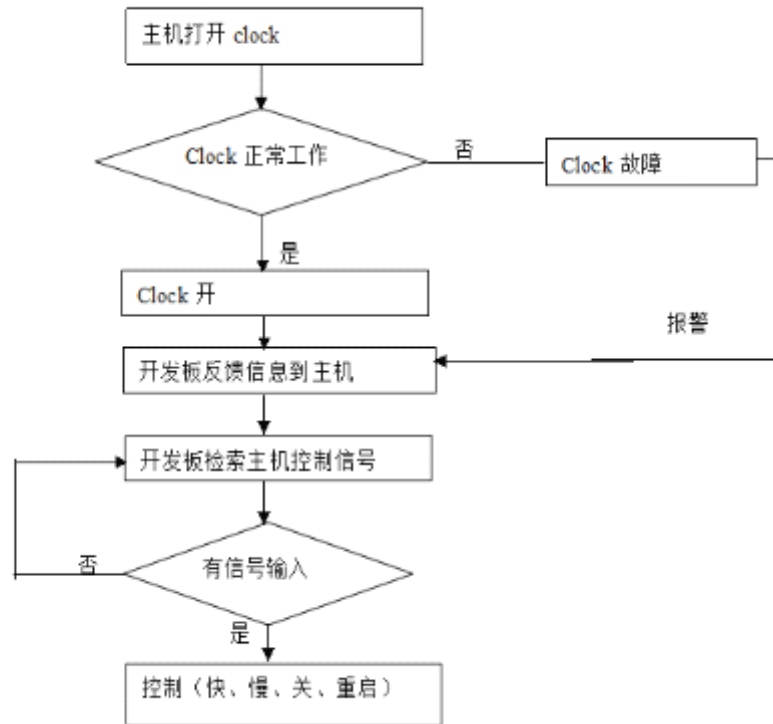
操作系统的相关数据类型定义

```
typedef unsigned char    BOOLEAN;
typedef unsigned char    INT8U;
typedef signed char      INT8S;
typedef unsigned int     INT16U;
typedef signed int       INT16S;
typedef unsigned long    INT32U;
typedef signed long      INT32S;
typedef float            FP32;
typedef double           FP64;
typedef unsigned int     OS_CPU_SR; // Define size of CPU status register (PSR
= 32 bits)
```

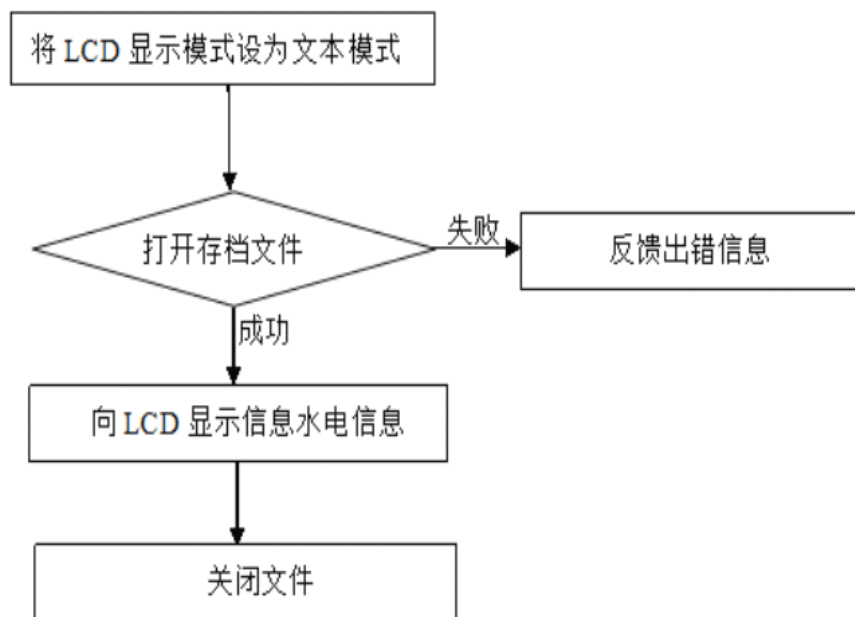
第五章 功能实现

系统采用一个时钟控制程序进行测试，来模拟水电表的抄送功能。其工作流程如图

- Clock 信号控制



系统对水电表信息的读取流程如下图



水电表信息读取部分代码

```
loadsystemParam();  
LCD_printf("Create task on uCOS-II...\n");  
OSTaskCreate(Uart1_Task, (void *)0, &Uart1_Stk[STACKSIZE-1],  
Uart1_Task_Prio);  
OSTaskCreate(Meter_Task, (void *)0, (OS_STK *)  
&Meter_Stack[STACKSIZE-1], Meter_Task_Prio);  
OSAddTask_Init(1);  
LCD_printf("Starting uCOS-II...\n");  
LCD_printf("Entering graph mode...\n");  
LCD_ChangeMode(DspGraMode);
```

第六章 总结

本系统采用了比较成熟的理论技术，保证了系统安全可靠，性能稳定。做为一个模拟系统，其可以推广对整个智能家居系统提供技术上支持。最后，感谢翟建芳老师的指导和软件学院其他老师的大力支持。只有在他们的全力支持下我们才有可能完成对整个智能家居系统的了解、建立和完善。

参考文献

- [1]. Jean J Labrosse. . 《嵌入式实时操作系统 μ C/OS- II（第 2 版）》. 北京:北京航空航天大学出版社, 2006.
- [2]. 于明, 范书瑞, 曾祥烨. 《ARM 9 嵌入式系统设计与开发教程》. 北京:电子工业出版社, 2006.
- [3]. 雷小俊, 李伟. 《SQLite 在嵌入式 Web 服务器中的应用》. 信息技术, 2006(6):128-130.
- [4]. 杨大千, 等.《基于 ARM9 和 Linux 的嵌入式动态 Web 技术设计与实现》. 工业控制计算机, 2006.