

## 基于 ARM 处理器 AT91S 的 M2M 终端设计

### 引言

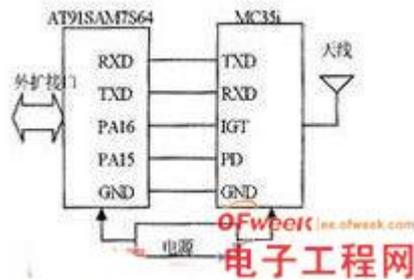
目前,对输油管道、电力装置、油井等进行远程监控主要采用人工巡逻的方式,这种方式存在实时性差、成本高、浪费人力资源、无法对环境恶劣的地区进行监控、可能出现误报等缺点。随着工业领域现代化水平的提高和通信技术的发展,这种生产方式亟待得到改善。

M2M 是指应用无线移动通信技术,实现机器与机器、机器与人之间数据通信和交流的一系列技术及其组合的总称。GPRS 具有覆盖地域广、通信距离远、网络可靠性高等优点。随着 GPRS 的推广和应用,以及基于 GPRS 的 M2M 产品的推出,采用 GPRS 技术来解决上述问题成为一种较好的方案。

嵌入式系统是当前国内外研究的热点之一。采用嵌入式系统技术设计一种可以解决远程监控领域数据传输问题的通用终端是必要的。本文重点阐述了基于嵌入式 ARM 处理器,结合 GPRS 模块 MC35i 设计 M2M 终端的总体设计方案。

### 系统硬件组成

M2M 终端的硬件核心为 GPRS 通信模块 MC35i 和 ARM 处理器 AT91SAM7S64,它们的接口设计如图 1 所示。M2M 终端对外留有 2 个 RS232 串口,通过发送预先定义好的数字指令,可以实现 M2M 终端的启动、关闭等。当 M2M 终端与 Internet 建立连接后,只需把封装好的数据通过串口发送到 M2M 终端,由其完成向监控中心发送数据的功能。



### GPRS 模块

GPRS 模块是实现 M2M 终端平台的核心部分,设计中采用西门子公司 GPRS 通信模块 MC35i。本模块包括 CPU 接口电路、SIM 卡接口电路和 MC35i 外围电路,MC35i 支持 GPRS 的四种编码协议 CS-1、CS-2、CS-3 和 CS-4,理论上最高传输速率可达 172kbps。它具备完整的 GSM 和 GPRS 功能,可以广泛应用在相关的 M2M 数据传输平台上。MC35i 提供了标准 AT 命令界面和一个 RS232 接口,用于与外部应用系统连接。

## ARM 处理器

本数据传输平台处理器采用 ATMEL 公司生产的 32 位 ARM7TDMI 体系结构处理器 AT91SAM7S64, 该芯片采用 3.3V 电压进行供电, 支持低功耗模式, 它具有 64kB 的 FLASH 和 16kB 的 SRAM, 具备丰富的外围设备资源, 其中包括 3 个 UART 通信串口和 1 个 USB2.0 全速设备, 利用本芯片完成系统功能的同时可以实现系统的高性价比。CPU 模块硬件电路包括: 时钟电路、JTAG 接口电路, RS232 接口电路等。

## 电源设计

系统采用 9V 电源进行供电, 经过 TI 公司 LDO 降压芯片 UA7805 进行一次降压, 使电压降至 5V。GPRS 模块 MC35i 要求的电源电压为 3.3V~4.8V, 而且要求电源必须能够提供 2A 的尖峰电流, 因此需要将 5V 电源通过一个 0.7V 压降的肖特基二极管 1N5819 后输入 MC35i, 同时需要接入耐压 25V、1000 $\mu$ F 的电解电容, 以为 MC35i 提供足够的尖峰电流, 将 5V 电源经过 TI 公司电源模块 REG1117-3.3 将电压降至 3.3V, 为微处理器及其他芯片提供电源。

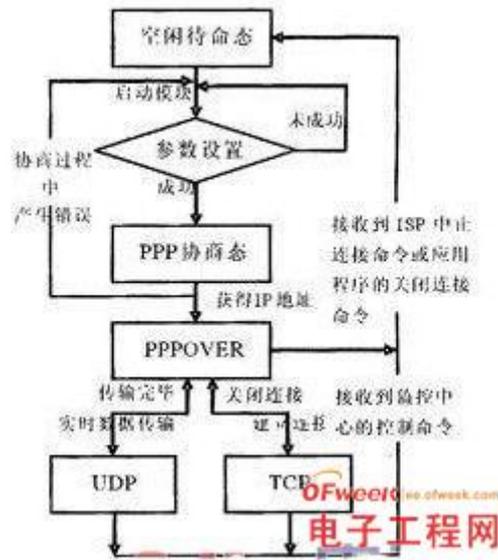
## 嵌入式软件设计

采用嵌入式实时操作系统可以更合适、有效地利用 CPU 的资源, 简化应用软件的设计, 缩短系统开发时间, 更好地保证系统的实时性和可靠性。FreeRTOS 是在 Sourceforge 网站上发布的微内核嵌入式实时操作系统, 它是完全免费的操作系统, 具有源码公开、可移植、可裁减、调度策略灵活的特点。作为一个轻量级的操作系统, FreeRTOS 提供的功能包括: 任务管理、时间管理、信号量、消息队列、内存管理等

FreeRTOS 支持优先级和轮换时间片两种调度算法, 可根据用户需要设置为可剥夺型内核或不可剥夺型内核。本文基于 FreeRTOS 给出了 M2M 终端实现 PPP 协商以及数据封装传输的软件设计方案。

## 基于状态机设计 GPRS 连接任务

由于网络和信号较弱等原因, 可能导致节点与 GPRS 网络连接的失败, 采用基于状态机的结构设计方法对各个阶段产生的错误进行处理, 能保障模块与 GPRS 网络建立可靠连接。程序状态机如图 2 所示。



**空闲待命态：**此时 MC35i 处在离线关闭状态，节点处在低功耗模式下，系统复位后处于此状态。

**GPRS 参数设置态：**处理器控制启动 MC35i 模块后进入此状态，通过发送 AT 命令对模块及必要的网络参数进行设置，为使各个参数均设置成功，软件设计中增加了容错重试机制。

**PPP 协商态：**GPRS 参数设置完成后，通过发送 AT\*99\*\*\*1#命令开始 MC35i 模块与 GPRS 网络 ISP(网络服务提供商)的 PPP 协商软件设计中采用 LCPHandler() 函数完成 LCP 协商，PAPHandler() 完成认证，由 IPCPHandler() 完成 IPCP 协商，如果最后获得 ISP 和本节点的 IP 地址，则进入 PPPOVER 态，此后就能进行数据的传输了，由于 GPRS 网络等原因，PPP 协商有时会失败，此时应重启 MC35i 模块，再按照状态机流程重新连接。

**UDP 数据传输态：**当程序采用 UDP 方式进行数据传输时，程序进入此状态，通过 xDataTrsmtTask() 任务进行数据的 UDP/IP 封装和解析。

**TCP 数据传输态：**当节点调用 uip\_cionnect() 函数与监控中心建立连接后，程序进入 TCP 数据传输态，进行基于 TCP 的数据传输。

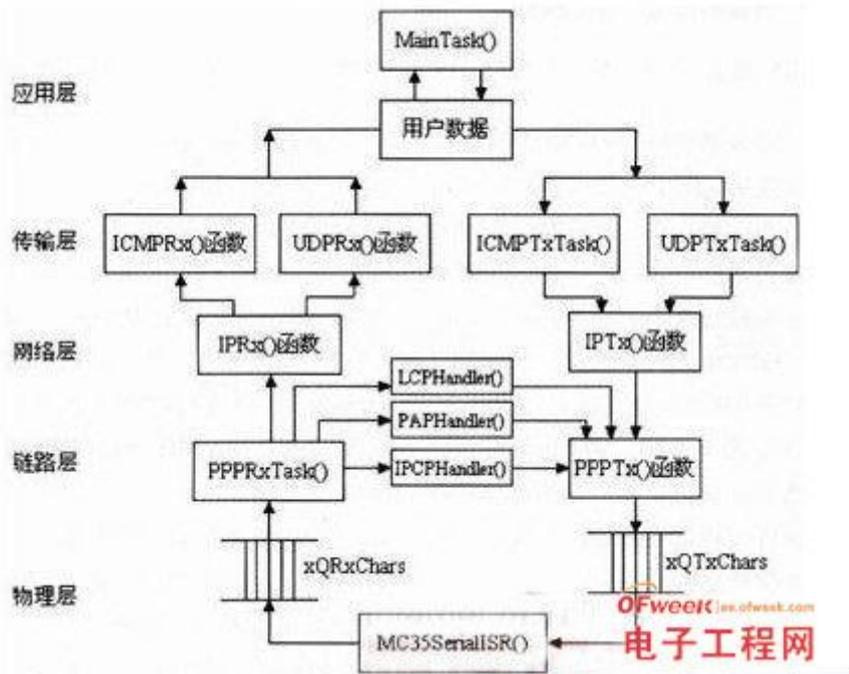
## 数据的封装和传输

通过 GPRS 进行数据的传输需要经过 Internet 网络进行中转，因而传输的数据封装必须进行 TCP/IP 协议。文中利用软件进行了数据封装，需要传输的数据经过传输层 UDP 协议头封装，然后是 IP 协议头的封装，最后进行 PPP 协议的封装。

MC35i 将接收到的数据透明地传输到 Internet 网络中。通过 Internet 网络路由器中转，最终将数据传输到监控中心。接收端对接收到的数据按照相应的层次进行解析，从而确定数据的目标程序。

系统软件设计采用分层的结构，从底到上分别为：串口驱动层(物理层)、PPP协议层(链路层)、IP协议层(网络层)、UDP协议和ICMP层(传输层)以及应用层。在移植好的LwIP协议栈中，通过在各层中建立相应功能的线程，实现数据的封装。底层软件为上层软件提供函数支持，上层软件利用底层软件完成应用程序的编写和实现。软件采用自底向上的设计方法逐步实现系统中各个函数的功能，各部分函数实现均采用模块化的设计方法。每个任务对应一个模块

对每个任务单独进行设计后，最终由FreeRTOS操作系统统一管理，通过采用信号量和邮箱的方式实现多个任务之间的通信，软件各部分主要函数之间的关系如图3所示。



在MC35SerialISR()中将接收的数据存放到xQRxChars队列中后，发送SemMC35Rx信号量来激活PPPRxTask()任务，通过对接收数据的解析，确定数据包的类型，然后由相应的函数对接收数据进行处理。

如果接收的数据是应用程序的数据，将由IPRx()函数判断目标主机是否正确，再经过传输层解析数据从而判定对数据处理的应用程序。最后由应用程序解析数据并执行相应的功能，如将数据通过串口发送到主机、向数据采集系统发送控制命令、接收数据采集系统的数据并发送等。当接收队列中所有数据均处理完毕后，延时250ms如果还没有接收到数据，则任务通过等待信号量SemMC35Rx将自己挂起。数据的发送过程是一个相反的过程。

应用程序根据需要的功能建立UDPTxTask()或ICMPTxTask()任务，并将数据发送到xAPPTxQ队列中。相应的任务再调用IPTx()和PPPTx()函数进行数据的封装并将数据发送到xqTxChar队列中，从而唤醒MC35SerialISR()中断程序将数据通过串口发送到MC35i中进行传输。为提高系统的实时性，本文中FreeRTOS

采用可剥夺内核方式进行调度。采用 FreeRTOS 操作系统对任务进行管理简化了软件的编写难度，同时提高了程序的可读性和可移植性。

## 总结

基于 GPRS M2M 产品的无线数据传输以及远程监控系统是目前国内外研究的热点。本文采用完全免费的操作系统和 TCP/IP 协议栈给出的系统设计方案具备成本低、性能好、可升级等优点，为远程监控系统相关领域的数据传输提供了一个可行的设计方案。