**基于STM32平台移植Free Modbus通信协议**

1. 准备材料

* Free Modbus v-1.50版本代码
* STM32标准库
* Keil V-5开发软件

1. 基本原理

串口发送中断至少有两种方式，第一种，数据寄存器空中断，只要数据寄存器为空并且中断屏蔽位置位，那么中断就会发生；第二种，发送完成中断，若数据寄存器的数据发送完成并且中断屏蔽位置位，那么中断也会发送。我非常建议各位使用串口发送完成中断。Free-Modbus多使用RS485通信中，从机要么接收要么发送，多数情况下从机处于接收状态，要有数据发送时才进入发送状态。进入发送状态时，数据被一个一个字节发送出去，当最后一个字节被发送出去之后，从机再次进入接收状态。如果使用发送寄存器为空中断，还需要使用其他的方法才可以判断最后一个字节的数据是否发送完成。如果使用数据寄存器为空中断，那么将很有可能丢失最后一个字节。

Modbus协议中没有明显的开始符和结束符，而是通过帧与帧之间的间隔时间来判断的。如果在指定的时间内，没有接收到新的字符数据，那么就认为收到了新的帧。接下来就可以处理数据了，首当其冲的就是判断帧的合法性。Modbus通过时间来判断帧是否接受完成，自然需要单片机中的定时器配合

1. 移植过程

* 将Free-Modbus文件中，modbus整个文件拷贝到新的工程目录下，同时还有demo/BARE/port文件夹拷贝新的工程目录；
* 对port文件夹中的文件(portserial.c和porttimer.c)进行编辑，主要包括串口、定时器基本信息的配置；
* 针对Modbus不同的功能码，编写对应通信代码；

1. 核心思想：

* 控制器端口初始化配置，管脚配置信息；
* 串口波特率、数据位、校验位配置；485接口还有控制端管脚配置；
* 定时器具体参数配置，通过时间间隔来判断数据整完整性；
* Modbus通信协议具体实现，数据请求和应答信号，遵从对应的数据协议格式；

1. 参考链接：

* <https://www.amobbs.com/forum.php?mod=viewthread&tid=5491615&highlight=freemodbus>
* <https://blog.csdn.net/golf_research/article/details/52240739?t=1515683902153>

1. 注意点：

* 实际上modbus协议原本是用在plc上的，plc的第一个数据起始地址就是1。 你看你现在设置为0，但是第一个数据无法读出，因为你从modbusPoll发出去看上去是从0开始读取的，但是freemodbus转换后起始地址是1，读取数据函数里读取哪个地址使用的地址是iRegIndex = ( int )( usAddress - REG\_INPUT\_START );所以导致你第一个（0号）数据无法读取，你改成REG\_INPUT\_START = 1，那么其实地址就是你数组的buf[0]数据了。
* <https://blog.csdn.net/byxdaz/article/details/78335633?locationNum=5&fps=1>

1. 其他

* Free-Modbus只能使用从机功能。Free-Modbus更适合嵌入式系统，虽然例子中也有WIN32的例子，如果想要做PC机程序并实现主机功能，推荐使用另一个Modbus库——NMODBUS，使用C#开发。