

UHF RFID 读写器读标签程序编程指南 v1.0.1

说明

本文档叙述了主机(Host)设备与 UHF RFID 读写器之间如何实现高效通信的编程指南。这里指的主机，包括计算机、嵌入式工业控制板和 M3 和 M4 以上等级高速单片机控制板等用户设备。

通信接口

读写器的通信接口包括：串口、RS485 口和网口等接口。

读取标签速率

在读取多标签的命令的情况下，标签读取的平均速率在 10mS 左右。主机设备与读写器之间的通信，要充分考虑多任务操作系统的实时性。

读取标签的命令

在**命令模式**下，主机(Host)发送包含 0xC1 命令的命令帧给读写器(Reader)，读写器则根据命令帧传递的读取标签的算法、Q 值参数和盘点标签的次数等参数，启动读取标签动作。(读写器的 API 通信格式，请参考相关文档。)

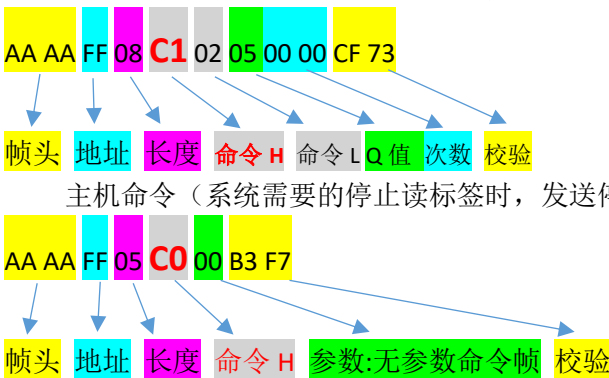
与读取标签命令相关联的指令，包括标签读取命令 0xC1 和停止读取标签 0xC0。因此，主机的程序在接收读写器的应答帧的时候，要对 0xC1 和 0xC0 的命令进行识别和处理。同时，主机程序在解码标签数据的时候，要从读写器的应答帧中，判断正常的状态值 0x00 和无标签超时状态值 0x15。正在状态值 0x00 的应答帧，则从后续的应答帧数据中，按照格式取出数据，其余错误的状态值，则可以丢弃。

一. 对于读写器，有两种方式读取标签：

(1)发命令启动读取标签和发命令停止读取标签：发送命令启动标签连续读取 0xC1，循环读取的次数设置是 0x00，系统需要的停止读标签时，发送停止标签读取命令 0xC0。

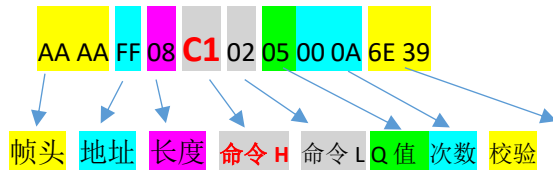
详情请参考：SYS-IoT UHF 读写器及模块通讯协议 V3.0

例如，主机命令（发送命令启动标签连续读取 0xC1，循环读取的次数设置是 0x00）



(2) 发命令启动读取标签和自动停止读取标签：发送命令启动标签连续读取 0xC1，循环读取的次数设置是非 0（≠0x00），读写器在发送设定的轮询次数之后，自动停止读取标签，并且自动主动向主机上报 0xC0 的应答帧数据，以通知主机(Host)，已经结束本次读取标签的操作。

例如，主机命令 AA AA FF 08 C1 02 05 00 0A 6E 39，其中轮询次数为 0x0A 次。



如下图 1 所示：

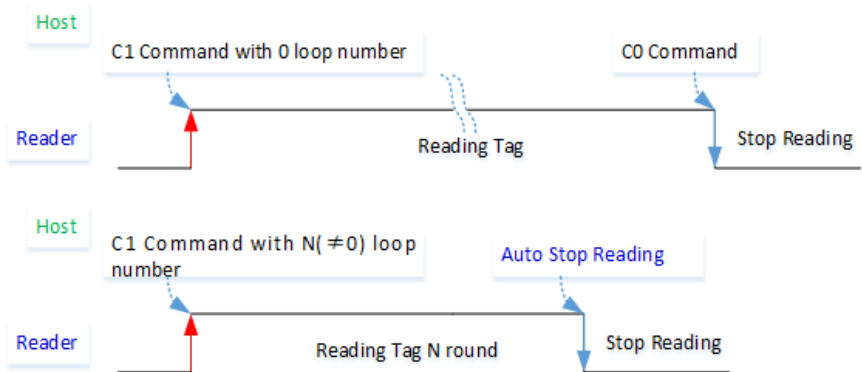
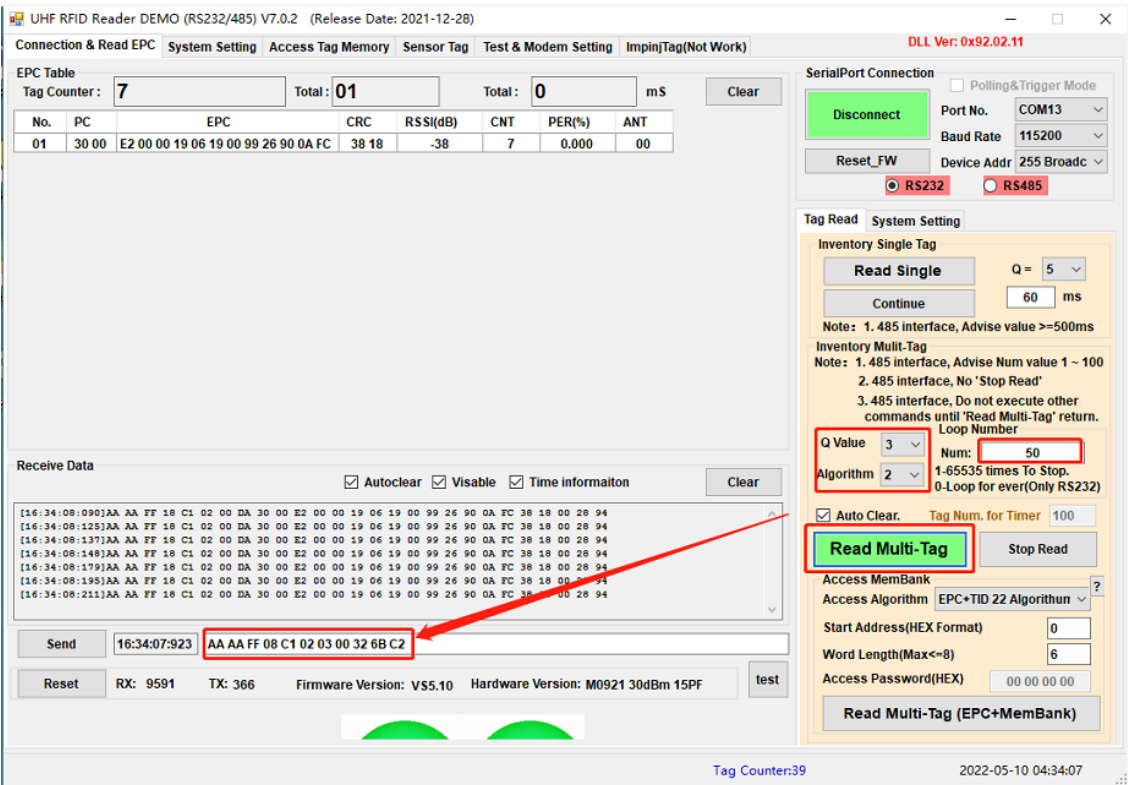


图 1

二. 快速获取指令方法：

针对读标签的指令，我们可以通过 Window 平台下的 DEMO 程序进行快速获取。

例如：我们想获取 Q=3,算法=2,读取 50 次，在点击 Read Multi-Tag 后，在程序 Send 框中获取的 AA AA FF 08 C1 02 03 00 32 6B C2，就是该参数中读标签的指令



解码标签数据

主机的程序，在解码标签数据的过程中，考虑到主机系统的多任务性，首先，建立高优先级的线程，专门负责接收读写器的应答帧数据，先存入循环队列缓冲区(Queue)；再建立另外一个中等优先级线程，专门负责从循环缓冲区队列，读取数据，再进行解析解码，最后再将解码的缓冲区存入标签数据缓冲区。

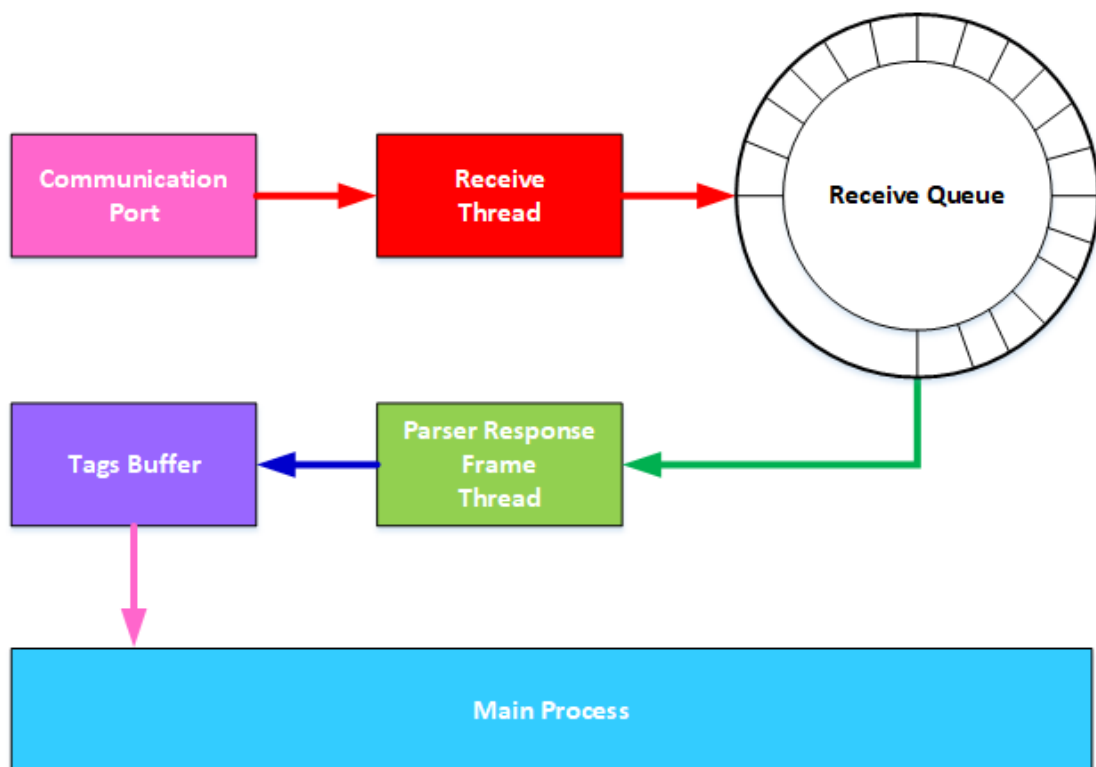


图 2

1. 由于读取标签的平均速率在 10ms 左右，标签连续上传的时间间隔在 3~20ms 的范围内变化。因此，建议主机程序先将通信口（RS232/485/网口）上，接收到的读写器应答帧数据，先无条件存储到一个循环缓冲区队列(Queue)当中。**不建议**，一边接收读写器应答帧数据，一边解析解码应答帧。否则，要考虑主机系统的多任务性，能否及时接收标签的应答数据，而不至于丢数据。
 - 对于 Windows/Linux 操作系统的主机，要先建立一个相对较高的优先级线程，从通信口接收数据，优先存入循环缓冲区队列。
 - 对于单片机主机程序，在串口中断接收数据，先将读写器应答帧数据存入循环缓冲区队列。
2. 接收循环缓冲区队列大小建议要大于 1K 以上；
3. 建立应答帧解析线程，从接收循环缓冲区队列中，读取数据，按照通信格式，进行通信格式的解析。
4. 应答帧解析线程，建议用状态机的方式解析；如图 3 所示。

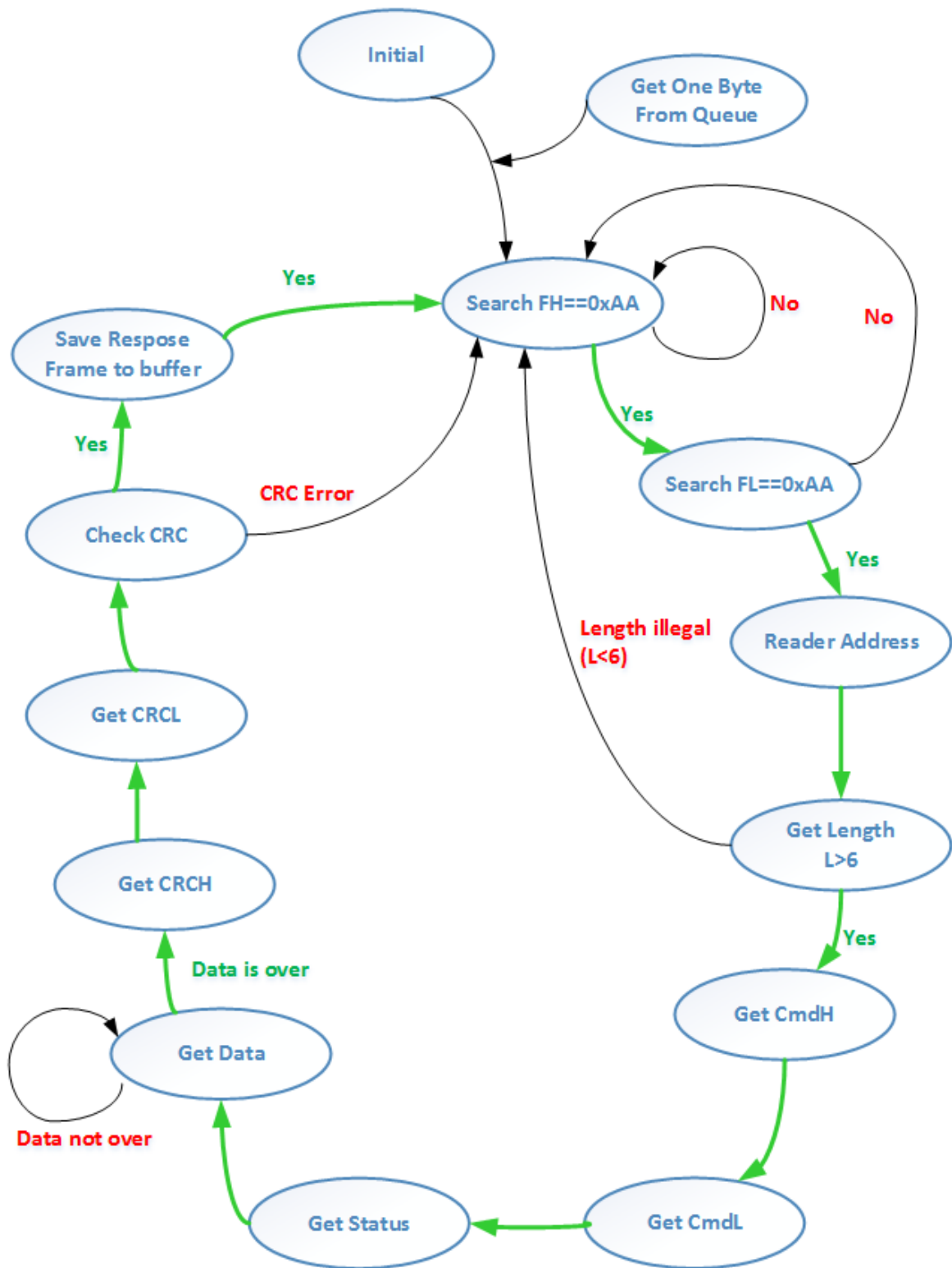


图 3

5. 主程序从标签数据缓冲区中，取出标签数据，做进一步处理。

>Host: AA AA FF 08 C1 00 05 00 BC 44 4C //QV=5, InvNumber = 0x00BC=188
 >Reader: AA AA FF 17 C1 00 00 CF 30 00 00 00 00 00 00 00 00 10 00 00 2D E3 C5 32 E8
 >Reader: AA AA FF 17 C1 00 00 CF 30 00 00 00 00 00 00 00 00 10 00 00 2D E3 C5 32 E8
 >Reader: AA AA FF 06 C1 00 15 8E C0

```
>Reader: AA AA FF 18 C1 00 00 BC 30 00 E2 00 30 09 28 11 01 46 11 20 A5 20 23 98 00 4D 56
```

[illegible]

```
[2022-05-27 16:22:49.651]# RECV ASCII>
:Reader Initial OK!

:-----Application(V94.02.00.20)-----
```