

自定义 OTA 通信协议

(1.0.0)

IoT 小胡

2020 年 6 月 12 日

概述

本协议为 **IoT 小胡** 自定义的 STM32 串口升级协议, 仅用于理解 OTA 的原理以及实验用. 还存在很多不足之处, 因此不推荐商用.

数据字段格式

	指令类型	字段	备注	
上位机 (发送)	开始指令	[S,T,A,R,T] + [FILE_SIZE] + [SUM]	[S,T,A,R,T]	5 字节,ASCII 码 , 后同
			[FILE_SIZE]	4 字节, 高字节在前,表示二进制文件的大小
			[SUM]	1 字节,总加和,后同
	异常结束指令	[E,N,D] + [SUM]		
	数据指令	[D,A,T,A] + [SIZE] + [START_ADDR] + [CODE] + [SUM]	[SIZE]	1 字节,表示本条指令的数据
			[START_ADDR]	4 字节,高字节在前,后同
			[CODE]	SIZE 字节的程序
下位机 (发送)	请求指令	[D,A,T,A] + [SIZE] + [START_ADDR] + [SUM]		
	升级成功指令	[O,K] + [SUM]		

讲解:

升级指令最开始由上位机发出,

- 上位机发出升级指令, 会间隔一定的时间发送 **开始指令**
- 当下位机收到上位机发送的开始指令之后, 修改标志位, 重启
- 重启后执行 boot 程序, 自检 2s(如果对实时性要求比较高的话可以通过标志位来完成, 不自检. 设置自检主要用来防止 app 烧录失败, 这样其实只要 bootloader(不需要 app)就能够进行程序升级了)
- 如果在自检过程中收到 **开始指令** 就从中获取升级文件的大小.
 - 如果没有收到, 但是读取标志位为升级模式, 会继续自检更长时间(防止丢包)
 - 如果标志位也没有被修改, 自检也没有收到 **开始指令** 说明没有升级需求, 就正常启动
- 获取到升级文件大小之后就开始根据升级文件的大小向上位机获取数据的 **请求指令**
- 当上位机收到下位机的 **请求指令** 之后就停止发送 **开始指令**, 转而发送 **数据指令**
- 这样下位机一直获取数据直到数据接收完成(因为知道总大小, 所以知道什么时候完成)
- 下位机数据接收完成之后发送 **升级成功指令**, 上位机收到之后就知道结束了.
- 备注: 如果下位机中途收到 **异常结束指令** 可能上位机部分有人终止了升级过程或者出错了, 就可以停止升级过程.