上下位机通信协议：

ＵＡＲＴ通信，最好以数据包形式发出，包头用来确认数据是否准确

上下位机发送指令每包有效内容4字节：

第一字节：

0x01, 前进指令，后三字节分别给定前进速度、转弯方向和转弯速度，下位机返回0x00表示速度更新成功，任何其他字节均表示

失败（ｐｓ：上位机发送的数据有可能导致两侧轮胎到满速，使下位机无法转弯，此时应优先转弯需求，处理代码放在下位机。

可以定义错误返回代码，比如0xff电机响应错误，0xee收到前进指令但尾数据不全等等）

example: 0x01 0x07 0x00 0x04 含义：前进，前进速度为极限速度的(7/256)，左转，转角为（4/4096\*360度）

example: 0x01 0x2A 0x01 0x0C 含义：前进，前进速度为极限速度的(42/256)，右转，转角为（12/4096\*360度）

example: 0x01 0x07 0x02 0x04 含义：前进，前进速度为极限速度的(7/256)，左转，两侧转速差为极限速度的（4/4096\*360度）

example: 0x01 0x2A 0x03 0x0C 含义：前进，前进速度为极限速度的(42/256)，右转，两侧转速差为极限速度的（12/4096\*360度）

ps:前进速度指的是最慢一侧轮胎的速度,定义这样的转弯规则是因为编码器是４０９６线的，这样可以刚好除开

0x02，获取当前速度，后三字节均为0x00。返回数据应为4字节，分别为错误代码(0x00代表成功，其他均失败)、前进速度、转弯方向

和转弯速度

example: 上位机：0x02 0x00 0x00 0x00

下位机：0x00 0xf0 0x02 0x05 含义：成功，前进速度(240/256),左转，速度差(5/256)

example: 上位机：0x02 0x00 0x00 0x00

下位机：0xff 0xef 0xac 0x34 含义：失败，后三位为随机错误代码（ｐｓ，可定义错误返回代码）

0x0a,开启反馈模式，即每收到一次指令0x01,执行前进指令，并且自动返回四字节数据（即执行0x01指令，返回0x02数据），返回

一字节错误代码0x00代表成功，任何其他代码表示失败。若收到其他指令则正常执行，并同样返回错误代码和当前速度信息

0x0b, 关闭反馈模式，返回一字节错误代码0x00代表成功，任何其他代码表示失败

0xff，强制停机，所有轮胎停止，指示灯（如果有的话）闪烁，返回一字节错误代码，0x00代表强制停机成功，任何其他代码表示失败

例子：

包头0xcc，包尾0xcc:

upper computer: 0xcc 0x01 0x10 0x00 0x00 0xcc //前进

lower computer: 0xcc 0x00 0x00 0x00 0x00 0xcc //成功

upper computer: 0xcc 0x02 0x00 0x00 0x00 0xcc //获取当前速度

lower computer: 0xcc 0x00 0x11 0x02 0x00 0xcc //获取成功，并返回速度值

upper computer: 0xcc 0x0a 0x00 0x00 0x00 0xcc //开启反馈模式

lower computer: 0xcc 0x00 0x00 0x00 0x00 0xcc //成功

upper computer: 0xcc 0x01 0x34 0x01 0x54 0xcc //前进

lower computer: 0xcc 0x00 0x15 0x03 0x23 0xcc //成功，由于处于反馈模式，返回当前实际速度

upper computer: 0xcc 0x11 0xc //未定义的指令、包头包尾损毁

lower computer: 0xcc 0xff 0xff 0xff 0xff 0xcc　 //返回错误代码

upper computer: 0xcc 0x0b 0x00 0x00 0x00 0xcc //关闭反馈模式

lower computer: 0xcc 0x00 0x00 0x00 0x00 0xcc //成功

upper computer: 0xcc 0xff 0x00 0x00 0x00 0xcc //强制停止

lower computer: 0xcc 0x00 0x00 0x00 0x00 0xcc //强制停止成功

ＰＳ：一般情况下下位机会长时间以反馈模式工作，但考虑到代码结构的合理性和以后扩展的需要，还是希望把它封成

单独的模块，实现时要尽量降低反馈模式的工作时长