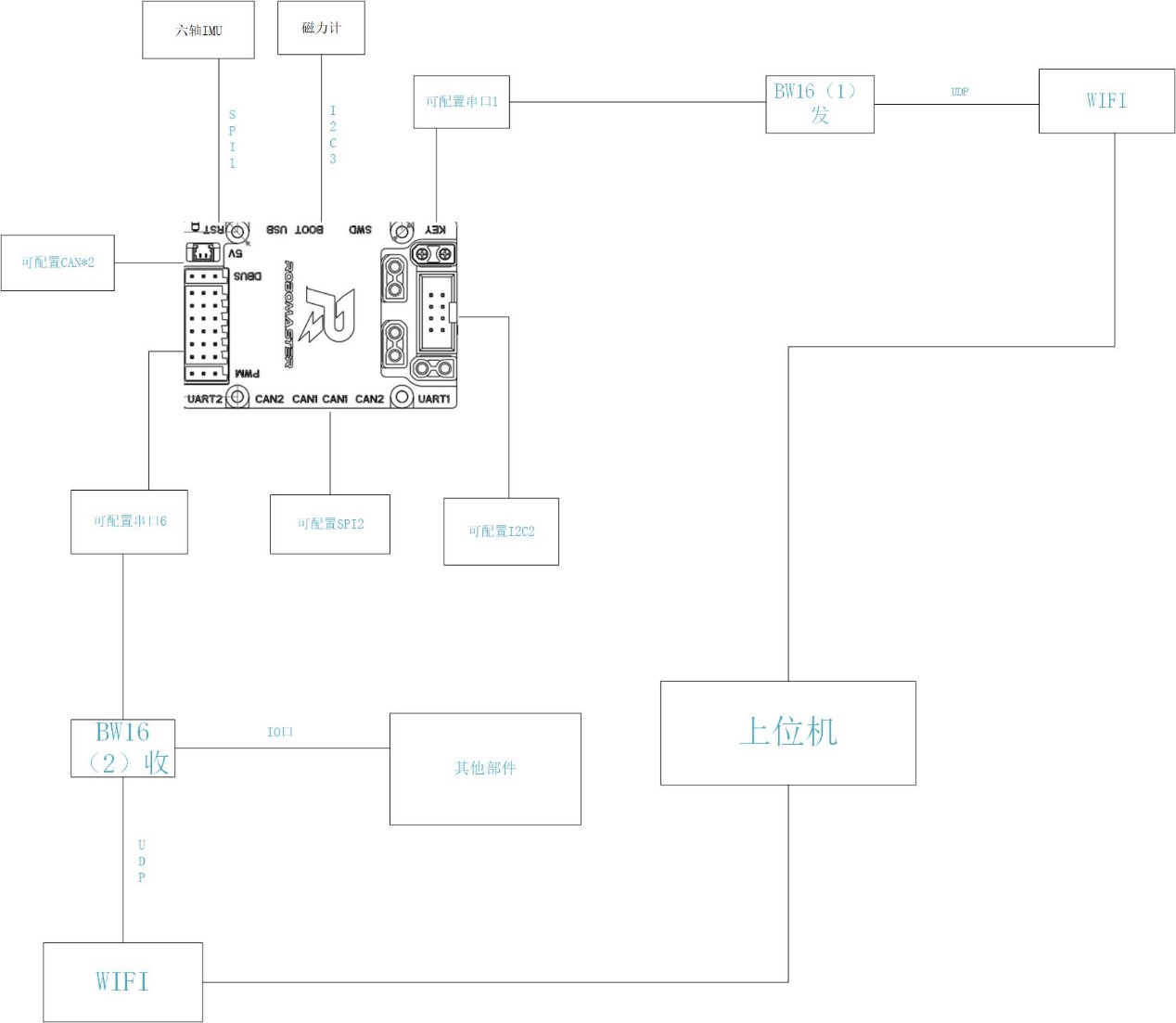
主控板： RoboMaster开发板C型

资源：CAN \*2 UART\*2 DBUS USB I2C SPI



ModBue协议

OSI七层模型

开发思路：

物理层 –>> 链路层 –>> 应用层

物理层为杜邦线连接；

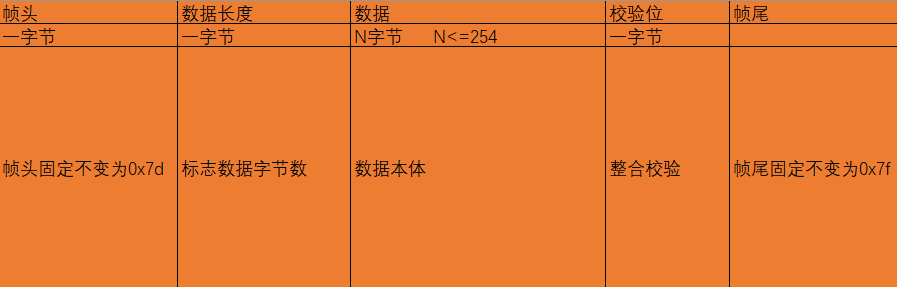
链路层负责封装数据为帧的形式；

应用层负责处理数据；

**链路层：**

进行通信前，首先进行握手，发送方先向接收方发送一段特殊的字符，以查询接收方的接收状态，当接收方做好了接收准备后，向发送方发送一串指令，表示可以开始通信，此时握手结束，进入通信状态。

通信以数据帧格式进行发送



帧头为固定字节0x7d，代表SOF为帧的起始标志；

其次为数据长度储存数据字节数，要求数据字节数不得大于254；

数据分装成字节形式，int和float等多字节数据拆成字节形式包装；

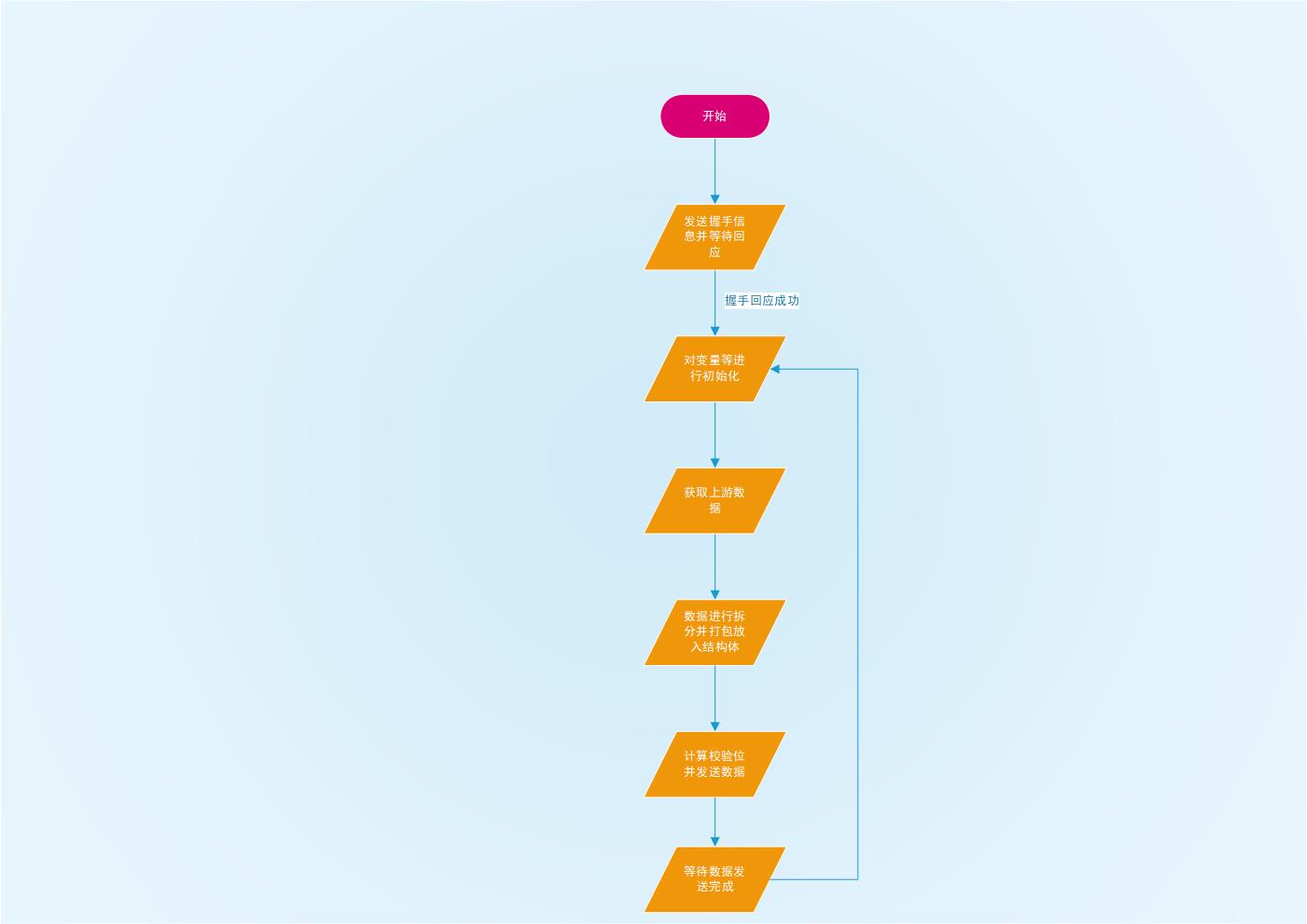
校验采取双重校验，首先比对数据长度和实际数据程度能否对上，其次进行整合累加校验，把数据每个字节相加后取前四位和后四位组成校验码放入校验位，接收端进行同样操作以验证数据传输是否有误。

帧尾为结束位，固定字节为0x7f，代表EOF为帧的结束标志。

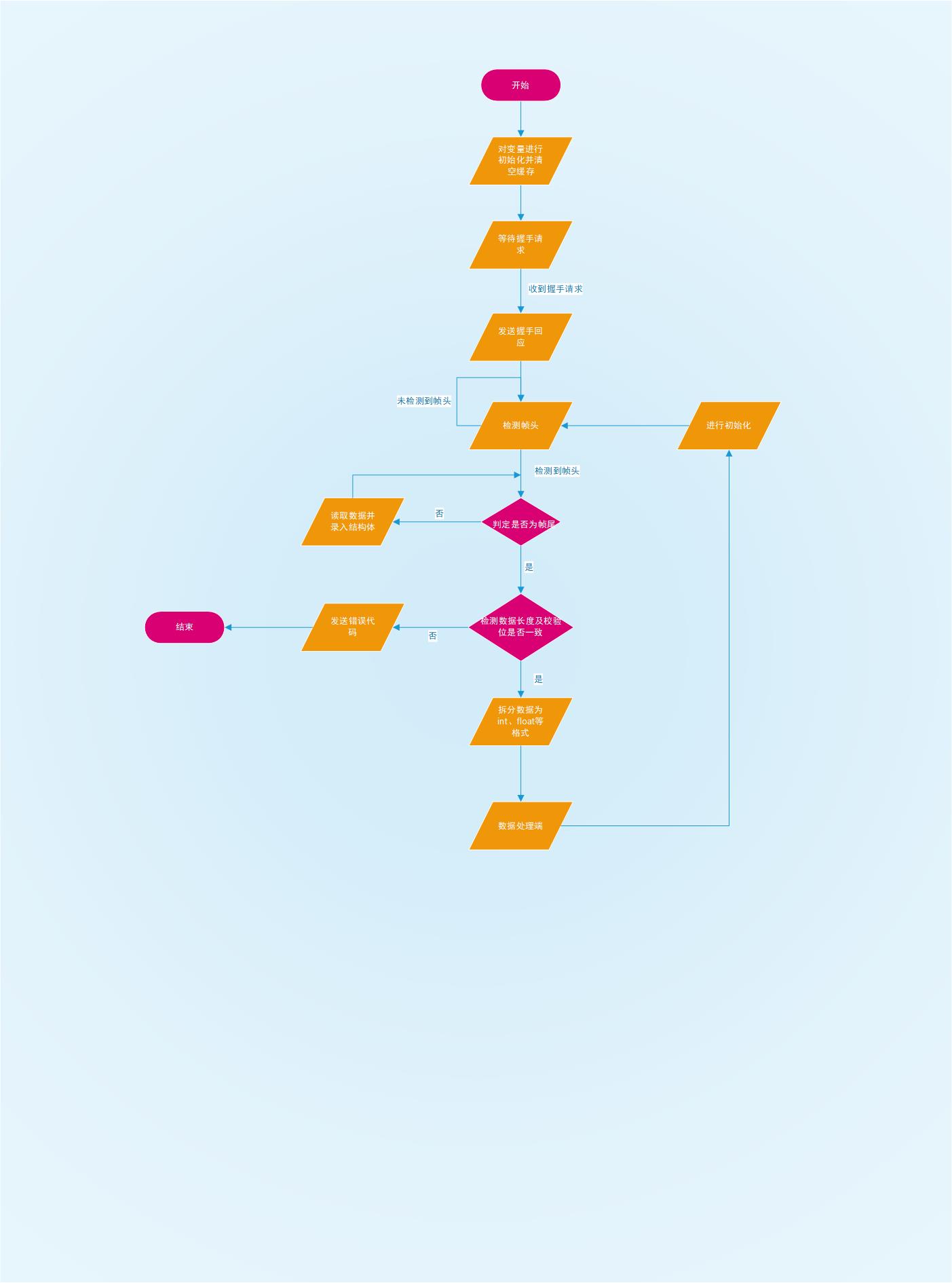
**应用层：**

首先发送端进行发送数据帧的初始化，数据帧以结构体形式进行打包把帧头、帧尾和数据长度放入，将产生好的数据拆解为字节形式放入结构体内并计算校验位。然后把包放入字节数据并发送，然后等待发送结束。

接收端先进行初始化，清空串口缓存和复位数据变量，握手成功后开始循环检测，直到检测到帧头，接着依次按照数据帧格式放入数据结构体，每读取到一个字节都和帧尾进行比对，当读取到帧尾则停止读取，进行数据处理。首先进行数据长度检测，如果数据长度不对，则删除此数据并发送错误代码。若数据长度检测无误则根据数据计算校验码和数据帧内的校验码进行比对，有错则删除数据并发送错误代码，无误则进行数据解析，先依照约定的数据位次，对数据字节进行分割，分割为int、float等形式。分割完成后传入新的结构体发送给下一个处理程序。



发送流程图



接收流程图

报错机制：当检测到错误直接返回不同的值使上级函数做出不同的处理方式

发送端：

数据过大，（大于254）

长时间未得到握手回应，握手失败。

接收端：

握手成功后长时间未收到数据

数据本体有误

数据传输良率过低

长时间检测不到帧尾

缓存溢出

BW16 串口

其与电脑通过ch340沟通 对应pin为serial-tx/rx 在程序中为Serial1

LOG Tx/Rx不与串口芯片相连 标号为Serial