**卡拉扬项目ATS设计**

一、通讯方式:单板ATS是UART,总装ATS是BLE。

二、关键点:1、需要调通BLE数据透传。

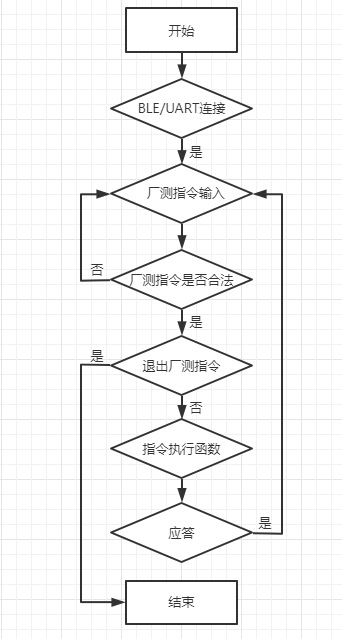
2、需要调通UART数据通信。

3、需要编写指令解析函数。

4、需要编写指令执行函数。

三、基本框架

程序运行起来，默认进入ATS模式，通过发16进制的指令(如0x900100)，当BLE或者UART连接上时，通过BLE或者UART发送指令，只有在输入的指令合法时，就会执行厂测指令对应的函数。假若输入退厂测指令，则退出ATS。流程图如下：



四、重要函数

1、函数用途：BLE接收数据处理函数。

void ats\_server\_write\_req\_ind(uint8\_t att\_idx, uint8\_t con\_idx, uint16\_t length, uint8\_t const \*value)

{

    if(att\_idx == ATS\_SVC\_IDX\_RX\_VAL)

    {

        SimplePeripheral\_processATSCmdEvt(value,length);    //调用解析命令函数

    }

    else if (att\_idx == ATS\_SVC\_IDX\_RX\_NTF\_CFG)

    {

        LS\_ASSERT(length == 2);

        memcpy(&ats\_cccd\_config, value, length);

    }

    else

    {

    }

}

2、函数用途：UART接收数据处理回调函数。

void ats\_uart\_cb(uint8\_t value)

{

    ats\_uart\_buff[ats\_index++] = value ;

    if(ats\_index < 3)

    {

        uart\_drv\_enable\_rec\_it(U\_ATS);

        return ;

    }

    if(ats\_uart\_buff[2] == (ats\_index - 3))              //通过判断字节长度来知道数据接受完成

    {

        SimplePeripheral\_processATSCmdEvt(ats\_uart\_buff , ats\_index);

        // //将接收到的数据再发回给串口调试助手

        // HAL\_UART\_Transmit\_IT(&uart\_driver\_arr[U\_ATS].USART\_X\_Handle,ats\_uart\_buff,ats\_index);

        ats\_uart\_recv\_enable();

    }

    else{

        uart\_drv\_enable\_rec\_it(U\_ATS);       //每次接受一个字节，保存在ats\_uart\_rx\_buf数组里面

    }

}

3、函数用途：BLE把执行函数得到的数据回给上位机。

ats\_send\_notification(notify,length);

4、函数用途：UART把执行函数得到的数据回给上位机。

HAL\_UART\_Transmit\_IT(&uart\_driver\_arr[U\_ATS].USART\_X\_Handle,notify,length);

5、函数用途：区分BLE和UART工作。

void ble\_uart\_notify(uint8\_t length)

{

    if(connect\_id != 0xff)

    {

        ats\_send\_notification(notify,length);

    }

    else

    {

        HAL\_UART\_Transmit\_IT(&uart\_driver\_arr[U\_ATS].USART\_X\_Handle,notify,length);

    }

}

6、函数用途：通过传入的指令和指令长度，函数内部通过switch-case来执行对应指令函数。

void SimplePeripheral\_processATSCmdEvt(const uint8\_t \*buf , uint8\_t length)