

Sbus模拟库概览与函数调用说明

梗概

整个库有且仅有一个文件：BTFL_430.h

其中驱动层提供最基础的sbus通信模拟功能，涉及大量的sbus协议与ESP32底层知识，用户若不想深究实现原理其实不必研究代码，直接使用提供的API函数调用实现您想要达到的功能即可。

虽此文重在解释函数调用的方法，但也需担当sbus模拟功能实现的任务，因此此文的前半部分我会较深入地阐释这三个文件各自的函数内容与对应实现的功能的思想方法，后半部分则着重解释API用法与注意事项。想直接了解API调用部分的请点击[这个链接](#)或自行转到“”部分。

那么现在，让我们开始吧。

前提要点

本文所诉直接默认用户已了解sbus协议的基本知识（数据格式、传输要求等等）与ESP32系列芯片在Arduino上开发的知识点（多任务调度、面向对象方法、外设控制等等），要求用户至少对上诉知识点至少有一些大体上的认知，能做到遇见不懂的部分，知道该向哪个方向搜寻资料并补充。

以下为关于sbus协议的重要资料文档，务必仔细阅读并理解方可继续下面的学习。另ESP32资料请自行搜寻。

1. [Futaba S-BUS controlled by mbed | Mbed](#)
2. [一文看懂Sbus协议 | CSDN](#)

数据类型简解

```
1 struct ReturnVal{
2     errCodeType errCode = BTFL_OK;
3 };
```

用例

```
1 ReturnVal RTN;
2 RTN = btfl.setTXpin(12, 6);
3 if(RTN.errCode != BTFL_OK){
4     //该函数出现错误，应当进行错误处理
```

```
5 //不同的错误代码意味着出现对应不同的错误
6 }
```

BTFL_OK: 正常

BTFL_NOFOUND: 输入参数值未对应要求

BTFL_threadERR: 新建多线程异常

API详解

由于我们的目标是使用ESP32硬件资源模拟出一个SBus通讯串口向主控发送经过调制的控制信息，因此在API接口层面，我们至少需要提供以下几个API函数给用户以进行必要的参数设置：

```
1 ReturnVal btfl.setTXpin(uint8_t pinNum, uint8_t pinUless)
```

功能: 设置Sbus输出引脚

参数:

- pinNum: sbus协议输出引脚编号，默认值为12。
- pinUless: 在整个工程项目中一定不会使用的引脚号，否则会产生串扰；默认值为6；

返回值: ReturnVal型结构体

```
1 ReturnVal btfl.allInit(void)
```

功能: 初始化

参数: 无

返回值: ReturnVal型结构体

```
1 ReturnVal btfl.startTaskLoop(void)
```

功能: 开始sbus数据输出（不能停止）

参数：无

返回值：RetVal型结构体

```
1 RetVal btfl.flyUnlock(void)
```

功能：解锁无人机飞控

参数：无

返回值：RetVal型结构体



飞控成功解锁后，btfl.flyUnlock_Flag的值会由0变为1，在该值变化后之后才可以进一步调用

btfl.flyDirection()函数配置对应通道值，否则会出现错误

```
1 RetVal btfl.flyLock(void)
```

功能：锁定无人机飞控

参数：无

返回值：RetVal型结构体

```
1 RetVal btfl.flyDirection(channelType channelFlag, percentType percentage)
```

功能：设置通道值

参数：

- channelType：选择对应想改变数值的通道，有且仅有以下参数可选：
 - Forward_and_Back
 - Left_and_Right
 - Up_and_Down

- rLeft_and_rRight
- AUX1
- percentage:
 - 若参数一选择Up_and_Down，则参数二值范围0~1000；
 - 若参数一选择其他，则参数二范围-1000~1000；
 - 若参数二的值超出范围，则用户输入值无效，自动改为0

返回值：ReturnVal型结构体

```
1 ReturnVal fineTuningChangeValue(channelType channelFlag, uint16_t value)
```

功能：设置通道误差值

参数：

- channelType：选择对应想改变数值的通道，有且仅有以下参数可选：
 - Forward_and_Back
 - Left_and_Right
 - Up_and_Down
 - rLeft_and_rRight
 - AUX1
- value：误差值（整数）

返回值：ReturnVal型结构体