主控机与飞行控制器之间的无线通讯协议

1. 更改历史

	编写	审核	日期	备注
1	吴桐	董伟	2013.3.28	
2				
3				

2. 述语

主控机：或称主控设备，四旋翼飞机的现场中央控制设备。实物为微型计算机与无线通讯设备的 集成。主控设备可通过无线指令控制多台次控机飞行。

飞行控制器：一个由嵌入式系统（单片机/arm 主控系统）与无线通讯构成的对旋翼飞器进行控制的软硬件集合。

3. 概述

本文档为建立合理，高效，稳定的串口通讯协议而编写。此协议只用于四旋翼飞行器主控机与飞行控制器的通讯实施。

本文档旨在为通讯双方制定一个统一的通信规范，本文档主要处理应用层。

本文档作为项目初期实施提供一个完整而易实现的协议，同时兼顾其可扩充性。

4. 物理层

上位机与设备采用 zigbee 无线通讯协议，具体设备采用 xbee 模块实现上位机与执行机构的无线通讯。上位机需要安装相应的驱动。波特率设置为 57600. 链路层由 zigbee 提供，只需合理配置以保证通讯链路的建立。

5. 通信协议

由于通讯数据性质不一，本协议中所有报文采用不定长方式制定。报文格式如图 1 所示。其中

- 起始段： 3 字节，固定为’>*>’。
- 结构长度段： 2 字节，标识当前数据结构的长度。
- 数据包描述： 2 字节，标识数据所属于的数据包。
- 实际数据结构：不定字节数，根据执行命令而定。
- 循环冗余校验段： 2 字节，数据校验（采用 CRC16 方式）。

- 结束段： 3 字节，固定为‘<#<’

图 1 报文格式及数据范例

起始段	结构长度	数据包描述	实际数据结构	循环冗余校验	停止段
>*>	Length	Description	Data	crc	<#<

5.1 系统参数

起始段	结构长度	数据包描述	实际数据结构	循环冗余校验	停止段
>*>	80	p	Data	crc	<#<

系统参数段为提供系统控制器（如 PID）等参数设定而提供的由主控机传向飞行控制器的数据包。由主控器传向飞行控制器。结构长度：80 ； 数据包描述：‘p’。每个参数占用 2 字节，共可传递 40 个（定点小数）参数。

参数上传无返回确认，可采用蜂鸣器鸣叫等方式确认。参数存于控制器缓存，掉电时不保存。

5.2 系统参数的保存

起始段	结构长度	数据包描述	实际数据结构	循环冗余校验	停止段
>*>	80	s	Data	crc	<#<

系统参数段为提供系统控制器（如 PID）等参数设定而提供的由主控机传向飞行控制器的数据包。由主控器传向飞行控制器。结构长度：80 ； 数据包描述：‘s’。每个参数占用 2 字节，共可传递 40 个（定点小数）参数。

参数上传后无返回确认，可采用蜂鸣器鸣叫等方式确认。参数存于控制器闪存，下次开机时自动设定。

5.3 实时控制参数上传

起始段	结构长度	数据包描述	实际数据结构	循环冗余校验	停止段
>*>	24	l	Data	crc	<#<

实时控制参数提供实时上传的控制指令（如姿态指令）或反馈信息（如 VICON 位置信息）。由主控器传向飞行控制器。

结构长度：24；数据包描述：‘l’。

每个参数占用 2 字节，共可传递 12 个（定点小数）变量。

5.3 状态采集

起始段	结构长度	数据包描述	实际数据结构	循环冗余校验	停止段
>*>	24	f	Data	crc	<#<

采集当时飞行器状态等信息，由飞行控制器传向主控机。结构长度：24 ； 数据包描述：‘f’。每个参数占用 2 字节，共可传递 12 个（定点小数）状态变量。通过将实时控制

参数的一个变量作为开关使用，可传递 12N 个变量。

附：CRC16 算法示例

```
unsigned shortcrc_update (unsigned short crc, unsigned char data)
{
        data ^= (crc & 0xff);
        data ^= data << 4;
        return ((((unsigned short) data << 8) | ((crc>>8)&0xff)) ^ (unsigned char) (data >> 4) ^
        (( unsigned short) data << 3));
}

unsigned shortcrc16(void * data,unsigned short cnt)
{
   unsigned shortcrc=0xff;
   unsigned char * ptr=(unsigned char *) data;
   int i;
   for (i=0;i<cnt;i++)
   {
      crc=crc_update(crc,*ptr);
      ptr++;
   }
   return crc;
}
```