Python接口使用说明

## 一、总体说明

为了保证较大数据负载的传输质量，需要对数据负载进行必要的预处理。因此，与传统的直接透明传输不同，本设计需要用户提供少量的数据元信息。Python接口的设计目标是消除用户代码与底层硬件的耦合，提供简洁直观的编程接口。上层用户可直接通过接口函数驱动底层硬件，无需关心底层驱动的实现细节。本说明对Python接口提供的所有函数进行了定义，并通过应用示例展示了Python接口的调用方法。

## 二、接口函数

Python接口库名称为UAV\_interface，主要包含以下功能：

**函数set\_serial(com):**

该函数用于修改嵌入式设备的串口端口号。传入参数为string型（不区分Windows和Linux系统）。

**函数U\_send(data, length, num):**

该函数用于将用户数据发送到无线设备。三个传入参数分别为：负载（数组，其中元素为十六进制表示的字节），负载长度（即负载的字节数），无人机数量（即负载包含多少架无人机的数据）。

**函数U\_receive():**

该函数用于持续监听是否收到来自无线设备的数据，并做出响应。无需传入参数，需作为线程在后台运行。

**类DataReceivedEvent：**

该类定义了以下两个函数，用于接收事件处理。

**函数add\_handler(handler):**

该函数用于注册事件处理函数。传入参数为事件处理函数名。

**函数notify(data):**

当函数U\_receive()监听到来自无线设备的数据后，该函数用于将数据传至事件处理函数。传入参数为监听到的数据。

**data\_received\_event:**

类DataReceivedEvent的实例化对象。

## 三、应用示例

1、使用前需安装以下Python库：serial，threading和crcmod。

2、数据接收流程：

（1）导入函数U\_receive、set\_serial，对象data\_received\_event和threading库。

（2）根据本地串口映射，对应修改串口端口号。

Linux系统：1710209195841(1)

Windows系统：1710209127013

（3）开启一个线程，使用函数U\_receive()持续监听是否收到来自无线设备的数据。

1710146193400

（4）自定义一个事件处理函数，该函数用于执行接收到来自无线设备的数据后的处理逻辑。下图的示例事件函数执行的逻辑为在收到嵌入式端的数据后直接打印数据。

1710209091903

（5）通过实例化对象data\_received\_event使用函数add\_handler(handler)注册自定义的事件处理函数。

1710145526843

3、数据发送流程：

（1）导入函数U\_send、set\_serial。

（2）根据本地串口映射，对应修改串口端口号。

Linux系统：1710209195841(1)

Windows系统：1710209127013

（3）调用函数U\_send，传入对应的负载、负载长度和无人机数量参数，即可通知无线设备进行发送。

1710146086816