# Ittd接口及定义说明

根据设计文档定义，Ittd是运行在嵌入式Linux下的服务程序，负责监听Itt的连接请求，接收、解析和处理相应的命令，并把处理结果返回Itt，Itt是运行在Windows下的程序，负责用户的命令请求发起和处理结果显示，**由于目前只是实现了通讯机制，也就是对指定长度的数据进行发送和接收，**所以本文不涉及具体命令的处理，以下就Ittd的接口及定义进行说明。

Itt

Windows

Ittd

Linux

Command

1. 源代码文件IQTest.h、IQTest.c

这两个文件位于IQDebug\IQDebug目录下，其中IQTest.h是Ittd和Itt都要使用的公共头文件，包含通讯所需的一些结构及宏定义，IQTest.c是Linux下运行的程序源代码，编译命令为“gcc IQTest.c -o IQTtest -lpthread”，运行命令为“./IQTest 11111”，“11111”是启动服务的监听端口，然后Itt就可以与Ittd进行连接并通讯了。通过这两个源代码文件，我们主要是了解接口的使用方法，实际应用时可以将相应接口嵌入其他程序使用。

1. 结构及宏定义

#define IQ\_TEST\_DATA 1 /\* 命令 \*/

#define MAX\_DATA\_BUFSZ 1024 \* 1024 /\* 接收数据缓冲大小 \*/

typedef struct {

int cmd; /\* 命令 \*/

int len; /\* 负载数据长度，不包括iq\_test\_msg\_t头长度 \*/

int xorcheck; /\* 头校验 \*/

} iq\_test\_msg\_t;

iq\_test\_msg\_t是Ittd和Itt通讯使用的数据头，对于通讯机制来说，只负责指定数据的发送接收，并不关心具体数据，**所以这个数据头与用户没有太多关系，用户应当自己定义数据结构。**此处的定义IQ\_TEST\_DATA是通讯使用的命令，该处存在的作用在于后面可能会需要扩充通讯机制，例如超大的文件或数据传输，需要发送方及接收方进行协商的，例如定义命令IQ\_TEST\_PACKET，对于该类型的数据分多次接收处理。

MAX\_DATA\_BUFSZ定义双方一次数据传输所需的最大缓冲大小，可根据实际情况调整。

1. 接口说明

1、int iq\_test\_start(unsigned short ipport);

用户使用前应首先调用该接口启动服务侦听，之后Itt就可以进行连接了，ipport为监听端口，成功返回0；

2、int iq\_test\_stop(void);

用户使用完应调用该接口结束服务，之后用户不能再连接和进行数据传输；

3、int iq\_test\_set\_callback(data\_pcallback pcallback);

用户调用该接口设置数据回调处理函数，通过网络有数据被收到时，调用该函数处理；

4、int iq\_test\_data\_send(void \*puser, void \*pbuf, int len);

用户调用该接口向Itt发送数据，puser表示对应的Itt连接（因为同时可能存在多个Itt连接），puser为NULL时将数据发送给所有已连接的Itt端，pbuf和len表示需要发送的数据指针和长度；

1. 数据回调处理的例子

static void iq\_test\_data\_callback(void \*puser, void \*pbuf, int len)

{

printf("data callback user %p data %p len %d\n", puser, pbuf, len);

/\* 原样回写 \*/

iq\_test\_data\_send(puser, pbuf, len);

}

当收到Itt发送的数据时，该函数被调用，puser表示对应的Itt连接，此处直接将数据原样发回到该Itt连接。**如果是一个异步处理，应当先保存puser指针，等处理结束后再调用接口iq\_test\_data\_send发送数据。**