**背景简介：**

SFTP是SSH File Transfer Protocol的缩写，安全文件传送协议。可以为传输文件提供一种安全的加密方法。sftp 与 ftp 有着几乎一样的语法和功能。

SFTP 是 SSH的一个子功能，在SSH软件包中，已经包含了一个叫作SFTP(Secure File Transfer Protocol)的安全文件传输子系统。

OpenSSH 是 SSH （Secure SHell） 协议的免费开源实现，但该开源代码一般用于比较大型，资源丰富的系统上，比如通用linux操作系统，windows操作系统，代码庞大，资源要求高。所以不适合移植到我们嵌入式系统中。

好在网上有个开源项目实现了基于嵌入式，RTOS，资源受限的系统上的SSH，该开源项目名为wolfssl(www.wolfssl.com)，

该SSH项目名为wolfSSH，其依赖的SSL协议为wolfssl。下载地址: https://www.wolfssl.com/download/

该开源软件已移植到DJYOS上，其中wolfssh-1.3.0 为应用程序，wolfssl-3.15.7 为SSL协议的实现。即wolfssh-1.3.0需要调用wolfssl-3.15.7中的代码。

**使用说明：**

既然sftp为安全文件传输协议，就会有客户端和服务器端，现在我们的DJYOS的wolfssh-1.3.0已实现了客户端和服务器端，只需要在Component Config中把wolfssh-1.3.0勾上，在应用代码中，通常是在djy\_main()函数上面增加以下shell命令安装语句

ADD\_TO\_ROUTINE\_SHELL(sftpd,sftp\_server,"start the sftp server,usage: sftpd -p port\n example: sftpd -p 22222");

ADD\_TO\_ROUTINE\_SHELL(sftpc,sftp\_client,"start the sftp client,usage: sftpc -h host -p port -u user -P password\n example: sftpc -h 192.168.0.1 -p 2323 -u user -P user");

即可同时支持SFTP服务器端和客户端，也可以只安装其中一个。注意，这时只是支持sftpd和sftpc命令，但系统中还未运行这两个服务。

注意：现在DJYOS创建事件Djy\_EvttRegist时，默认栈大小为0x1000，但由于sftp 相对比较大，需要的栈空间目前至少为0x9000，服务器端的设置在djysrc/third/wolfssh-1.3.0/examples/echoserver/echoserver.c sftp\_server\_create()调用的Djy\_EvttRegist()，目前已设置为0x9000。客户端由于是在shell命令中，所以创建shell命令的线程其栈大小需设置为0x9000，位于djysrc/component/shell/shell.c 的ModuleInstall\_Shell()调用的Djy\_EvttRegist()，如果为0x1000，需要改为0x9000。

**服务器端使用说明：**

如果需要拉起sftp服务器即sftpd，需要在命令行中输入sftpd [-p 22222]， 其中-p 22222为设置其监听端口号，可以不设置，使用默认端口，如果需要sftpd跟随系统一起运行，可以直接在应用代码中调用sftp\_server()函数，而不需要安装该命令。

服务器即我们板子运行的SFTP作为服务器，供远程SFTP客户端连接，服务器需要配置的参数主要有三个，在源码目录wolfssh-1.3.0/wolfssh 的test.h 文件中， 有如下定义。

#define serverKeyRsaPemFile "/efs/server-key-rsa.pem" /\*locate in wolfssh-1.3.0/keys \*/

#define serverKeyEccDerFile "/efs/server-key-ecc.der" /\*locate in wolfssh-1.3.0/keys \*/

#define serverKeyRsaDerFile "/efs/server-key-rsa.der" /\*locate in wolfssh-1.3.0/keys \*/

#define SFTP\_SERVER\_DEFAULT\_LISTEN\_PORT 22222 // ssh server listen port

#define SFTP\_SERVER\_USERNAME\_PASSWORD\_STRINGS "jill:upthehill\n jack:fetchapail\n" //username and password

其中serverKeyRsaPemFile，serverKeyEccDerFile, serverKeyRsaDerFile 为加密解密所需要的文件。必须把该三个文件放到板子上相应目录。比如现在定义的是/efs/ 目录。在wolfssh-1.3.0/keys 中可以找到这三个文件。

SFTP\_SERVER\_DEFAULT\_LISTEN\_PORT 为sftp服务器监听端口号，现在默认为22222.

SFTP\_SERVER\_USERNAME\_PASSWORD\_STRINGS 为客户端连接我们需要输入的用户名密码，格式为username:password\n 其中冒号和换行符必不可少。

由于sftpd只是ssh功能的一部分，所以运行sftpd时，作为ssh远程登陆(类似telnet功能)的服务器端也是支持的。客户端，比如windows上的SecureCRT可以通过快速连接中选择协议SSH2，输入sftpd 中设置好的用户名密码，和板子的IP地址，即可登陆。

在windows上可以用filezilla做客户端。

**客户端使用说明：**

即我们板子上运行的是sftp客户端sftpc， 必须安装为命令形式，在命令行执行：

sftpc -h host -p port -u user -P password

例子: sftpc -h 192.168.0.1 -p 2323 -u user -P user

其中-h 为远程主机地址， -p 为远程sftp服务器(ssh) 监听的端口号， -u 为用户名，-P[大写] 为密码。

在windows上可以用freesshd 做服务器端。

参考资料：

wolfssh 是一个应用，主要功能是ssh，但其包含了sftp功能，我们项目上服务器端进程名字为sftpd，只是一开始我们主要取其sftp的功能而忽略了ssh(安全外壳，远程登陆)本身的功能。 Wolfssl是一个开源项目，主要实现wolfssl这个ssl协议，是一个加解密协议，wolfssh的实现需要用到wolfssl。

wolfssl开源项目官网 www.wolfssl.com

下载地址: https://www.wolfssl.com/download/

如果在linux平台下编译，可以通过 wolfssh-1.3.0目录下的build\_makefile 来编译，注意需要修改build\_makefile中的SSL\_SOURCE\_DIR 指向wolfssl-3.15.7， make -f build\_makefile生成的只是sftp客户端程序，如果需要编译服务器端，需修改BUILD\_TARGET\_SRC为examples/echoserver/ echoserver.c。 和目标文件名BUILD\_TARGET

更多功能和编译请参考官方README文件，在wolfssh-1.3.0目录下。