拉伸机UI通讯接口

# 包含文件

Pipe.h

Pipe.cpp

# 示例代码

DICCLIENT.CPP

# 说明

1. 由于读操作会卡线程，所以读写需要在另一个线程中操作；
2. Client可以给拉伸机发送自动运行（:auto）、停止（:stop）、request（：request）三个指令，其中运行和停止Server不会有返回数据；request指令会返回机器的当前状态，**发送请求后务必进行读操作；**
3. 下位机在自动运行前需要手动执行回原点操作，否则没有参考位置，不能执行自动指令；
4. 发送自动后，request几次如果auto=0，则说明拉伸机没有自动运行；
5. 拉伸机和Client需要有断线重连的操作；
6. 如果连接断开，send和read指令都将会退出；
7. 时间是10进制明文；
8. 数据是Float型，需要先将8字节ASCII码合并成4字节BYTE，再将4字节BYTE转成float。

Request指令的返回格式如下：

char data[] = ":request\_pos=xxxxxxxx\_spd=xxxxxxxx\_pre=xxxxxxxx\_auto=x\_time=00:00:00:000\_end";//时间为时：分：秒：毫秒

参考代码如下：

void \_4byteTo8byte(byte src[4], byte output[8])//8字节转4字节按此代码反向操作即可

{

output[0] = src[0] / 16;

output[1] = src[0] % 16;

output[2] = src[1] / 16;

output[3] = src[1] % 16;

output[4] = src[2] / 16;

output[5] = src[2] % 16;

output[6] = src[3] / 16;

output[7] = src[3] % 16;

for (size\_t i = 0; i < 8; i++)

{

if (output[i] < 10 && output[i] >= 0)

{

output[i] += 0x30;

}

else

{

output[i] += 0x37;

}

}

}

typedef union

{

float floatData;

int intData;

unsigned int uintData;

}value\_info;

float bytesToFloat(byte src[], int offset)

{

value\_info info;

info.uintData = src[0 + offset];

info.uintData <<= 8;

info.uintData |= src[1 + offset];

info.uintData <<= 8;

info.uintData |= src[2 + offset];

info.uintData <<= 8;

info.uintData |= src[3 + offset];

return info.floatData;

}