力捷丰 S 系列在线式编程器/烧录器

DLL 接口命令说明

V4.23（更新于 2025 年 4 月）

力捷丰科技
Opteeq Technologies

感谢您选择力捷丰。

在第一次使用前，请先熟悉本产品的软硬件，并仔细阅读本使用手册。这样做有助于减少意外情况的发生，避免为您带来不必要的损失。

本手册中的图片均为示例图片，您手中的产品可能会由于型号、生产批次以及适用性等原因而与本手册中的描述有所不同。力捷丰科技不断致力于制造优质、高效、快速的产品，我们将保留随时更改设计、元件及技术工艺的权利。

目录

版本变更说明

版本变更信息如下:

版本号	**版本发布时间**	**发布内容**
V4.1	2024-07-01	第一次发布。
V4.2	2024-08-20	OpteeQ SetExtraData 增加部分说明
V4.21	2024-09-12	修改部分文字错误
V4.22	2024-11-21	修改 OpteeQ_GetDeviceList 函数说明
V4.23	2025-04-23	添加 OpteeQ_ExecCommand 示例

第一章文档说明

1.1 目的

本文档主要介绍力捷丰 S 系列在线式编程器/烧录器（包括 S1、S4、S8）的 DLL 所包含的库函数说明、使用流程、调用方法、注意事项等。帮助用户通过自己的软件调用 DLL 来控制烧录器并完成相应操作。

1.2 准备

1) 支持 win xp，win7(32/64 位)，win8(32/64 位)，win10(32/64 位)的 windows 操作系统。

2) 用户的 PC 机必须安装了力捷丰 S 系列在线式编程器/烧录器的软件。

3) 相关文件 OpteeQ_DLL.dll 和 OpteeQ_DLL.h 必须已经拷贝到用户的 PC 机上。

说明：文件 OpteeQ_DLL.dll 和 OpteeQ_DLL.h 在本公司提供的 U 盘里，路径为 U 盘:\DLL\DLL。

第二章工作流程

烧录器完成烧录的过程主要包括“获取烧录器列表”、“连接烧录器”、“烧录”和“断开”四个步骤。

2.1 获取烧录器列表

在对烧录器执行操作时，用户程序需要知道烧录器的设备路径信息，该路径信息可以通过 OpteeQ_GetDeviceList 函数来获得。当烧录器通过网络接口连接 PC 机时，用户程序需要先调用 OpteeQ_AddNetDevice 函数，将烧录器注册到设备列表里，然后才可以用 OpteeQ_GetDeviceList 函数来获取设备路径。

2.2 连接烧录器

用户程序在执行其它操作之前，PC 机与烧录器之间必须先建立连接。用户程序通过调 OpteeQ_Connect 函数来完成 PC 机与烧录器之间连接。

2.3 烧录

在 PC 机与烧录器已经建立连接的前提下，用户程序可以通过调用 OpteeQ_Run 函数，使烧录器开始烧录。

用户程序通过循环调用 OpteeQ_Ask 函数查询烧录结果。当烧录器完成烧录后，用户程序就会收到烧录结果。在烧录过程中，烧录器无法应答，调用 OpteeQ_Ask 函数会在 0.4 秒等待后自动返回 False。用户应当在延时一段时间后继续执行 OpteeQ_Ask 函数，若烧录完成，则 OpteeQ_Ask 函数立即返回 True，表明烧录过程已经结束并结束循环。如果用户程序在 150 秒内都没有收到烧录结束 True，则判定烧录超时，本次烧录失败。

说明：执行 OpteeQ_Run 函数时必须设置一定的延时（默认值为 0ms，推荐 200ms），否则无法获取正确的烧录完成状态。

2.3.1 S1 烧录流程

当用 S1 烧录器烧录时，OpteeQ_Ask 函数返回 True 表明烧录成功。无需查看 OpteeQ_Ask 函数的通道状态返回值。烧录流程如图 2.1 所示。

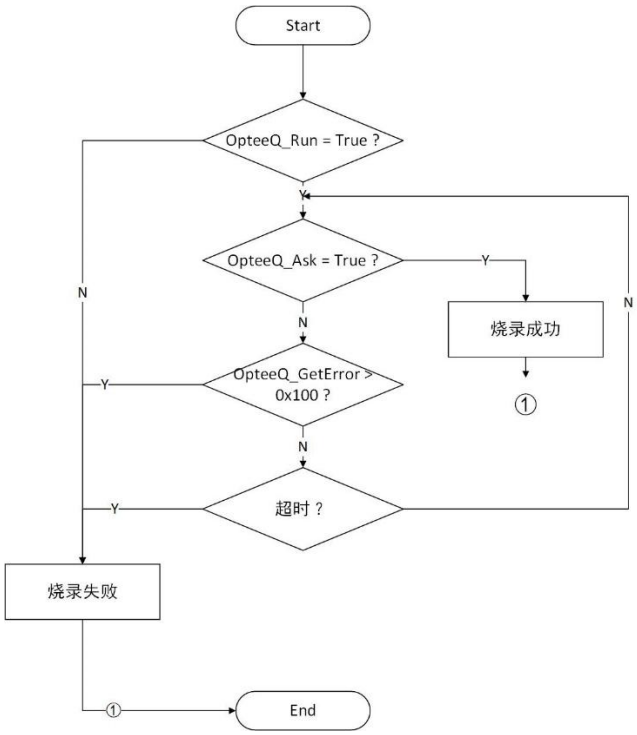

Figure 2.1: S1 烧录流程图

2.3.2 S8 烧录流程

当用 S8 烧录器烧录时，OpteeQ_Ask 函数返回 True 只表明烧录完成。 还要查看 OpteeQ_Ask 函数的通道状态返回值，该值存放在数组中，数组从 0 到 7 对应烧录器通道 1 到通道 8 烧录 结果。烧录流程如图 2.2 所示。

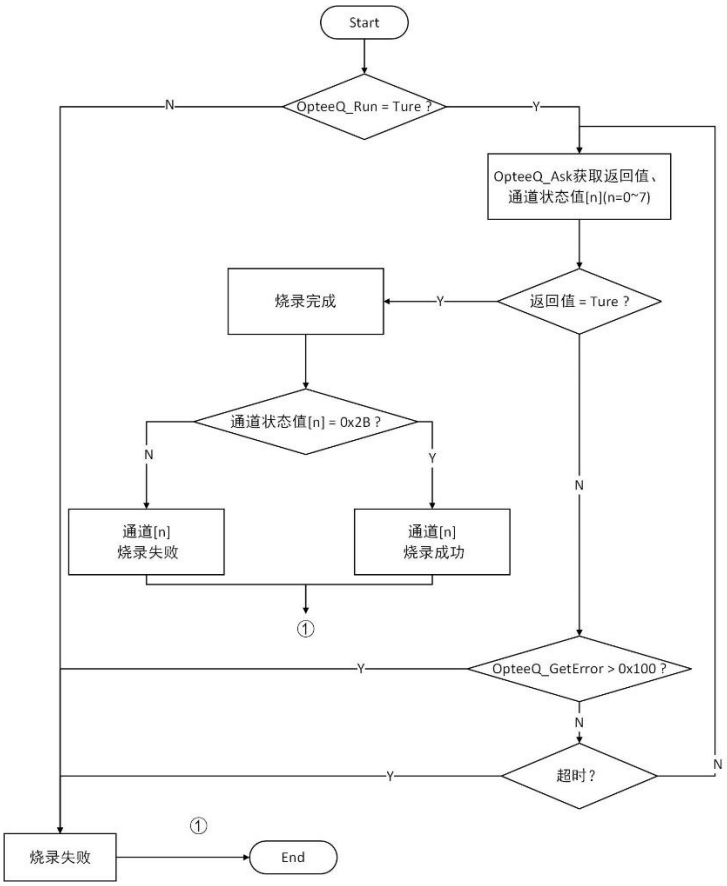

Figure 2.2: S8 烧录流程图

2.4 断开

当用户不需要再对烧录器进行任何操作时，可断开 PC 机与烧录器之间的连接。用户程序通过调 用 OpteeQ_Disconnect 函数来断开 PC 机与烧录器之间的连接。

第三章函数说明

3.1 OpteeQ_AddNetDevice

函数说明: **当烧录器通过网络接口连接 PC 机时，在对烧录器操作之前，必须先调用该函数进行注册。对于刚出厂的烧录器，其默认的 IP 地址为“192.168.0.100”，端口号为“8000”。**

语法:

---

bool OpteeQ_AddNetDevice(IN const char *szIP,IN short sPort);

参数:

---

szIP：烧录器的 IP 地址，格式为字符串，例如：“192.168.0.100”。
sPort：烧录器的端口号，默认值为“8000”。

返回值

True：注册成功。
False：注册失败。

3.2 OpteeQ_Ask

函数说明: 当烧录器进入烧录状态时，用户程序调用此函数以获取烧录结果。用户程序需要循环调用 此函数，因为在烧录过程中烧录器无法应答，在完成烧录之后才会应答。如果烧录器无应答，OpteeQ_Ask 会在大约 0.4 秒后自动返回 false；如果烧录器烧写完成，则返回 true。

语法:

---

bool OpteeQ_Ask(IN const char *szDevPath,OUT DWORD *pdwChannelStatus);

参数:

---

szDevPath：烧录器的设备路径。
pdwChannelStatus：烧录器通道的状态代码数值指针，需要 32 个字节的空间，每个通道的状态占用一个 DWORD，代码数值，请参见表 3.1。

返回值

---

true：断开成功
false：断开失败

3.3 OpteeQ_Connect

函数说明: 该函数用于建立 PC 机与烧录器的连接，从而使用户程序获取烧录器的控制权。在用户程序 执行其它操作前，必须首先建立连接。获取控制权即意味着烧录器将只接受调用了此函数的用户程序的控制。其它的程序此后无法控制烧录器，直到获取控制权的用户程序释放控制权（即调用 OpteeQ_Disconnect 函数）。

语法:

---

bool OpteeQ_Connect(IN const char *szDevPath);

参数:

---

szDevPath：烧录器的设备路径。

返回值

True：连接成功。
False：连接失败。

3.4 OpteeQ_DeleteNetDevice

函数说明: 该函数用于删除采用网络接口的烧录器，调用后 DLL 会释放相应的资源。

语法:

---

bool OpteeQ_DeleteNetDevice(IN const char *szDevPath);

参数:

---

szDevPath：烧录器的设备路径。

返回值

True：删除成功。
False：删除失败。

3.5 OpteeQ_DeletePrj

函数说明：**该函数用于删除烧录器中指定的工程，szPrjName 为需要删除的工程名称。**

语法:

---

bool OpteeQ_DeletePrj(IN const char *szDevPath,IN const char *szPrjName);

参数:

---

szDevPath：烧录器的设备路径。
szPrjName：需要删除的工程名称。

返回值

True：删除工程成功。
False：删除工程失败。

3.6 OpteeQ_Disconnect

函数说明：**该函数用于断开 PC 机与烧录器的连接，同时释放烧录器的控制权。**

语法:

---

bool OpteeQ_Disconnect(IN const char *szDevPath);

参数:

---

szDevPath：烧录器的设备路径。

返回值

True：断开成功。

False：断开失败。

3.7 OpteeQ_GetDeviceList

函数说明: **该函数用于获取当前已注册的烧录器列表，用户通过该函数来检查烧录器是否在线，烧录器的信息以字符串的形式存放在 lDevList[][256]中，lDevList[][256] 是一个二维数组，最大为 8*256 字节，执行函数前必须将 lDevList 内容初始化为零。当烧录器通过网络接口连接 PC 机时，用户程序需要先调用 OpteeQ_AddNetDevice 函数注册。**

语法:

DWORD OpteeQ_GetDeviceList(OUT char lDevList[][256]);

参数:

---

lDevList：返回所有烧录器的设备路径的二维数组。

返回值

当前在线烧录器的个数。

3.8 OpteeQ_GetDeviceName

函数说明：**该函数用于获取烧录器的名称，该名称存放在烧录器的 SD 卡上，更换 SD 卡会导致烧录器名称发生改变。**

语法:

---

bool OpteeQ_GetDeviceName(IN const char *szDevPath,OUT char * pStrDeviceName) ;

参数:

---

szDevPath：烧录器名称的存放路径。
pStrDeviceName：烧录器名称的字符串指针，需要 48 字节的空间。

返回值

True：获取名称成功。
False：获取名称失败。

3.9 OpteeQ_GetError

函数说明: 当其它函数执行失败时，用户程序可通过此函数返回值查询错误原因。

语法:

DWORD OpteeQ_GetError(IN const char *szDevPath,OUT char* szMessage);

参数:

---

szDevPath：烧录器的设备路径。
szMessage：返回的错误提示信息。

返回值

错误代码数值，请参见下表。

Table 3.1: 错误代码数值

序号	定义	数值	说明
1	EM_GETDEVICE	0x00000001	获取设备错误
2	EM_GETWINUSB	0x00000002	获取接口错误
3	EM_GETPIPE	0x00000003	获取通道错误
4	EM_SEND	0x0000000A	发送错误
5	EM_RECV	0x0000000B	接受错误
6	EM_TAG	0x0000000C	命令序列错误
7	STAT_CHECKINGID	0x00000021	正在检查 ID
8	STAT_FORMATING	0x00000022	正在擦除
9	STAT_BLANKCHECKING	0x00000023	正在空白检查
10	STAT_PROGRAMMING	0x00000024	正在烧写
11	STAT_VERIFYING	0x00000025	正在校验
12	STAT_CHIPINITING	0x00000026	正在初始化芯片
13	STAT_LOCKING	0x00000027	正在设置芯片保护
14	STAT_UNLOCKING	0x00000028	正在解除芯片保护

15	STAT_UNKNOWN	0x00000029	未知状态
16	STAT_READY	0x0000002A	就绪
17	STAT_PASS	0x0000002B	烧录成功
18	STAT_FAIL	0x0000002C	烧录失败
19	STAT_DISSELECT	0x0000002D	通道未选择
20	RECV_RUNNING	0x000000FF	正在烧录
21	RECV_TRUE	0x00000101	烧录完成
22	RECV_FALSE	0x00000102	烧录失败
23	RECV_TIMEOUT	0x00000113	烧录超时
24	RECV_INITERROR	0x00000114	初始化失败
25	RECV_FORMATERROR	0x00000115	擦除失败
26	RECV_PROGRAMERROR	0x00000116	烧写失败
27	RECV_CHECKERROR	0x00000117	校验失败
28	RECV_FILEERROR	0x00000118	烧录器文件打开失败
29	RECV_CHIPIDERROR	0x00000119	芯片 ID 错误
30	RECV_RANGEERROR	0x0000011A	烧写地址范围错误
31	RECV_BLANKCHECKERROR	0x0000011B	空白检查错误
32	RECV_LOCKERROR	0x0000011C	设置芯片保护失败
33	RECV_UNLOCKERROR	0x0000011D	解除芯片保护失败
34	RECV_FILE_COVER	0x0000011E	文件已存在
35	EM_FILEOPENERROR	0x00000201	本地文件打开失败
36	EM_FILENAMELENERROR	0x00000202	文件名过长
37	EM_FILEEMPTY	0x00000203	文件为空
38	EM_DEVICEPATHERROR	0x00000204	设备路径错误
39	EM_DEVICELOSTCONNECT	0x00000205	设备未连接

3.10 OpteeQ_GetPrjList

函数说明：**该函数用于查询烧录器的工程列表信息，执行后将把烧录器上的工程名存放到 lPrjList 中，lPrjList 是一个二维数组，最大为 256*48 字节，执行该函数前必须将数组 lPrjList 中的内容初始化为零。**

语法:

---

bool OpteeQ_GetPrjList(IN const char *szDevPath,OUT char lPrjList[][48]);

参数:

---

szDevPath：烧录器的设备路径。
lPrjList：工程名称的二维数组。

返回值

True：获取工程列表成功。
False：获取工程列表失败。

3.11 OpteeQ_GetVersion

函数说明：该函数用于查询烧录器的固件版本信息，版本信息会赋值给 pStrVersion 指针指向的区域。

语法:

---

bool OpteeQ_GetVersion(IN const char *szDevPath,OUT char* pStrVersion);

参数:

---

szDevPath：烧录器的设备路径。

pStrVersion：烧录器版本字符串指针，需要 48 字节的空间。

返回值

True：获取版本信息成功。

False：获取版本信息失败。

3.12 OpteeQ_Run

函数说明: **用户程序通过调用该函数使烧录器开始烧录。**

语法:

---

bool OpteeQ_Run(IN const char*szDevPath,
IN const char* szProjectName,
IN int iTime,
IN unsigned char bChSelBit,
IN const char * pExtraData,
IN int nExtraDataLen);

参数:

---

szDevPath：烧录器的设备路径。
szProjectName：要烧录的工程名称，注意添加“.sct”后缀，例如工程名为“test”，需要改成“test.sct”。 iTeme：延时，默认值为 0，推荐值为 200（单位：毫秒）。
bChSelBit：烧录通道选择参数，bit0~bit7 分别代表通道 1 到通道 8，如果要指定烧录的通 道，可以设置对应的 bit 位为'1'，如果全部为'0'，将会按照工程创建时设定的通道参数来烧录。
pExtraData：附加参数的数据指针，可以指定地址烧写序列号等功能，若不需要请填 0。
nExtraDataLen：附加参数数据的字节数，若不需要请填 0。

返回值

True：烧写命令执行成功，进入烧写状态。
False：烧写命令执行失败。

序列号参数说明:
序列号信息可以在调用 DLL 函数 OpteeQ_Run 时，用参数（pExtraData 和 nExtraDataL-en）来指定序列号。参数 pExtraData 为数据在内存中的起始地址，参数 nExtraDataLen 为数据 总长度（参数数据总长度不要超过 1024字节）。如果在烧录之前调用了 OpteeQ_SetExtraData 设置了参数数据，那么该函数的 pExtraData 和 nExtraDataLen 将无效。

每个序列号用一个结构体来表示，结构体以 1 字节对齐，如下所示:

```
#pragma pack(1)//指定 1 字节对齐
struct SerialNum
{
   DWORD dwCH_Number;//通道号
   DWORD dwAddress;//地址
   DWORD dwLen;//序列长度，单位为字节
   BYTE     bDatas[dwLen]; //序列内容
};
```

每一个序列号需要按照上述结构体来填充，多个结构体可以连在一起。

例如有下面 4 个序列号:

1 号通道在地址 0x00000004 处烧写序列“0x90,0x010,x56,0x78”。该序列号在内存中的结构图，如图 3.1 所示。

1 号通道在地址 0x00001000 处烧写序列“0x90,0x01”。

2 号通道在地址 0x00000004 处烧写序列“0x56,0x78,0x90,0x01,0x12,0x34”。

3 号通道在地址 0x00000008 处烧写序列“0x56,0x78,0x90,0x01”。 那么所有参数数据内容长度为 64，在内存中数据如下所示。

```
{
   0x01,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x90,0x01,
   0x56,0x78,
   0x01,0x00,0x00,0x00,0x00,0x10,0x00,0x00,0x02,0x00,0x00,0x00,0x90,0
   x01,0x02,0x00,
   0x00,0x00,0x04,0x00,0x00,0x00,0x06,0x00,0x00,0x00,0x56,0x78,0x90,0
   x01,0x12,0x34,
   0x03,0x00,0x00,0x00,0x08,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x56,0
   x78,0x90,0x01
}
```

如果该内存起始地址为 0xABCD0000，那么最终参数 nExtraDataLen 赋值为 64，pExtra-Data 赋值为 0xABCD0000。

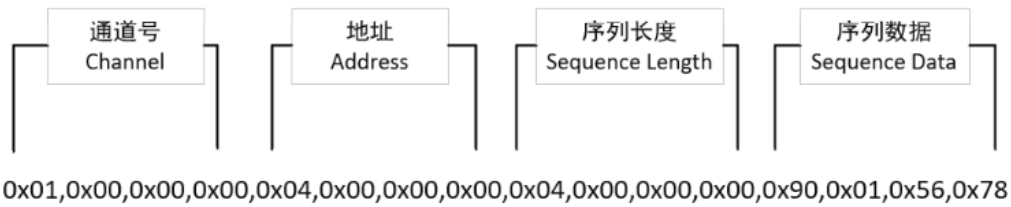

图 3.1 序列号结构图

3.13 OpteeQ_ReadChip

函数说明: **读取芯片上 FLASH 的数据。**

语法:

---

bool OpteeQ_ReadChip(IN const char *szDevPath,
IN BYTE bChNO,
IN const char *szProjectName,
IN const char *szSaveFileName,
IN DWORD dwStartAddress,
IN DWORD dwSize,
OUT DWORD* pdwBytesRecved);

参数:

---

szDevPath：烧录器的设备路径。
bChNO：芯片所在的通道号（1-8），用于 S4/S8 烧录器的通道选择，如果是 S1 烧录器，可以填写为 0。
szProjectName：芯片对应的工程文件名（需要加上后缀名“.sct”）。
szSaveFileName：保存文件的路径。
dwStartAddress：读取内容的起始地址。
dwSize：需要读取的字节数。
pdwBytesRecved：已读取数据大小的指针，用来表示已经读取的字节数，不用时可以填写为 NULL。

返回值

True：文件读取成功。
False：文件读取失败。

3.14 OpteeQ_ReadFile

函数说明: **读取烧录器上指定路径的文件，并保存在本地文件夹里。**

语法:

---

bool OpteeQ_ReadFile(IN const char *szDevPath,
IN const char* szTargetFileName,
IN const char *szSaveFileName,
IN bool bCover,
OUT DWORD* pdwAllByte,
OUT DWORD* pdwReadedByte);

参数:

---

szDevPath：烧录器的设备路径。
szTargetFileName：读取文件的路径。
szSaveFileName：保存文件的路径。
bCover：是否覆盖, True 表示覆盖，当保存文件已存在时会覆盖当前文件；False 表示不覆盖，当保存文件已存在时，会返回 False。
pAllByte：数据指针，用来表示文件总大小，不用时可以填写为 NULL。
pReadedByte：已读取数据大小的指针，用来表示已经读取的字节数，不用时可以填写为 NULL。

返回值

True：文件读取成功。
False：文件读取失败。

3.15 OpteeQ_SendFile

函数说明: **该函数用于发送一个工程文件到烧录器上，可以通过检查 pAllByte 及 pSendedByte 来查询 发送的进度。文件名最大长度为 15 个字符，只支持“sct”、“bin”及“lic”三种文件类型。**

语法:

DWORD OpteeQ_SendFile(IN const char *szDevPath,
IN const char *szFileName,
IN bool bCover,
OUT DWORD* pdwAllByte = NULL,
OUT DWORD* pdwSendedByte = NULL);

参数:

---

szDevPath：烧录器的设备路径。
szFileName：需要发送的文件的绝对路径（包含文件名）。
bCover：是否覆盖。
pAllByte：文件总字节数指针。
pSendedByte：已发送字节数指针。

返回值

True：发送文件成功。
False：发送文件失败。

3.16 OpteeQ_SetDeviceName

函数说明: **该函数用于设置烧录器的名称，该名称存放在烧录器的 SD 卡上，更换 SD 卡会导致烧录器名称发生改变。**

语法:

---

bool OpteeQ_SetDeviceName(IN const char *szDevPath,IN const char * szDevice- Name) ;

参数:

---

szDevPath：烧录器的设备路径。
szDeviceName：烧录器名称，不能超过 48 个字节。

返回值

True：设置名称成功。
False：设置名称失败。

3.17 OpteeQ_SetExtraData

函数说明: **该函数用于设置烧录器烧录的额外参数数据。**

语法:

---

bool OpteeQ_SetExtraData(IN const char* szDevPath,
IN int nChNumber,
IN DWORD dwAddress,
IN const char* szData) ;

参数:

---

szDevPath：烧录器的设备路径。
nChNumber：烧录器通道号，1~8 表示设置其中一个通道，0 表示所有通道。
dwAddress：额外参数数据的烧录地址。
szData：烧录的额外参数数据，参数以 16 进制字符串形式传入，字符串长度必须是 2 的倍数，如：“12345678ABCD”，表示 6 个字节的数据 0x12,0x34,0x56,0x78,0xAB,0xCD。

返回值

True：设置成功。
False：设置失败。
可以多次调用来设置多个额外参数数据，在调用完 OpteeQ_Run 函数后，所有设置的额外参数数据会被清空，程序在下一次调用 OpteeQ_Run 之前需要重新设置额外参数数据。
在版本 2.1.3.3 中新增了一项机制，即当遇到设置失败的情况时，DLL（动态链接库）会自动清空所有已经设置的数据，并强制确保下一次的烧录操作失败。

3.18 OpteeQ_SetNetDeviceIp

函数说明: **用于修改烧录器的 IP 地址。成功调用此函数后，需要重启一下烧录器来实现 IP 地址的修改。**

语法:

---

bool OpteeQ_SetNetDeviceIp(IN const char *szDevPath, IN const char*ip);

参数:

---

szDevPath：烧录器的设备路径。
ip：烧录器的新 IP 地址，格式为字符串，例如：“192.168.0.100”。

返回值

True：设置成功。
False：设置失败。

3.19 OpteeQ_StartLogToFile

函数说明: **调用该函数后，DLL 会生成运行日志，并写到文件 szFilePath 中。**

语法:

---

bool OpteeQ_StartLogToFile(IN const char* szFilePath);

参数:

---

szFilePath：日志文件路径。

返回值

True：成功。
False：失败。

3.20 OpteeQ_StopLog

函数说明: **调用该函数后，DLL 会停止生成运行日志。**

语法:

---

bool OpteeQ_StopLog();

参数:

无

返回值

True：成功。

False：失败。

3.21 OpteeQ_ExecCommand

函数说明: **让烧录器执行一条命令，烧录器正在执行命令时，该函数会阻塞，直到烧执行完命令后，函数才会返回。当有返回数据时，可以通过参数来获取烧录器返回的数据。**

语法:

---

bool OpteeQ_ExecCommand(IN const char* szDevPath,
IN const char* strCommand,
IN int nTimeOut,
IN OUT BYTE* pRecvBuffer,
IN DWORD dwRecvBufferSize,
OUT DWORD* pdwBytesRecved);

参数:

---

szDevPath：烧录器的设备路径。
strCommand：命令字符串。
nTimeOut：命令执行等待超时时间(单位：秒)。
pRecvBuffer：如果命令后返回数据，会存入该 pRecvData。
dwRecvBufferSize：表示 pRecvData 的最大缓存字节数，当返回数据字节数大于 dwRecvBuffer-Size 时返回 False。
pdwBytesRecved：表示命令实际返回的字节数，可设置为 0，表示不使用。

返回值

True：成功。
False：失败。

3.21.1 示例 1：PCMD GetFileCRC32

函数说明: **获取烧录器上文件的 CRC32 值。**

示例

```
DWORD dwCRC32 = 0;
OpteeQ_ExecCommand(devList[0],
        "PCMD GetFileCRC32 -file \\BIN\\test.bin",
        60,
        (BYTE*)&dwCRC32,
        sizeof(dwCRC32),
        0);
printf("CRC32 is 0x%08X\n", dwCRC32);
```

3.21.2 示例 2：PCMD GetLastCheckSum

函数说明: **获取烧录器在烧录时自动保存的 CheckSum 值。**

示例

```
DWORD dwLastCheckSum[8] = {0};
OpteeQ_ExecCommand(devList[0],
        "PCMD GetLastCheckSum",
        60,
        (BYTE*)&dwLastCheckSum,
        sizeof(dwLastCheckSum),
        0);
for(int i=0;i < 8;i ++)
{
        printf("CH%d CheckSum is 0x%08X\n", i + 1, dwLastCheckSum[i]);
}
```

第四章LabView 调用

本章介绍如何通过 LabView 来控制烧录器。提供的 VI 示例程序，放在本公司提供的 U 盘里，路径为 U 盘：\DLL\Example\LabView。用户也可以直接调用本公司提供的 VI 示例程序，跳过本章节。

Figure 4.1: LabView

第五章C 调用示例

如果想用 C 来控制烧录器，可以参考提供的 C 示例程序。本公司提供的 C 代码 DLL 调用，放在本公司提供的 U 盘里，路径为 U 盘：\DLL\Example\C。

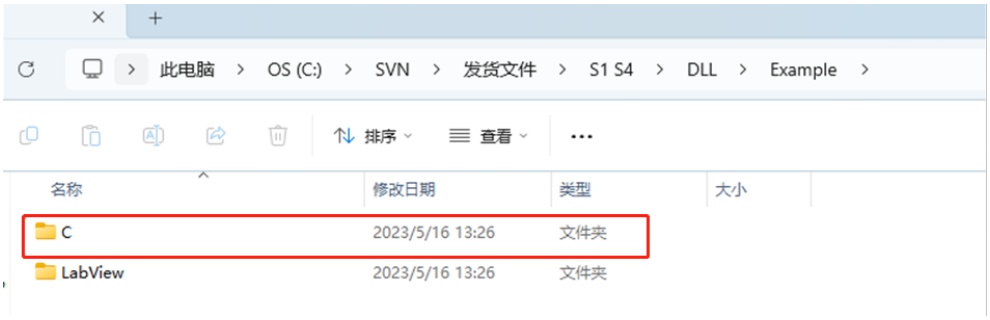

Figure 5.1: C

第六章帮助与支持

若您在使用本手册或本公司产品时出现问题或者疑问，可以联系我们获得帮助，我们的联系方式如下：

电话：400-002-3598　　0510-81813667

网址：www.opteeq.com

邮件：contact@opteeq.com