

力捷丰 SG 系列在线式编程器/烧录器

DLL 接口命令说明

V2.2.1（更新于 2025 年 2 月）



Copyright © 2025 Opteeq Technologies Co., Ltd

修订记录

版本号	修订者	发布时间	发布内容
V2.0	C	2024-2-20	删除新版本不支持内容
V2.1	K	2024-4-11	新增 QueryAllState 、 ReceiveDevLog 、 StopDevLog、 ChangeLogFile 函数说明
V2.1.1	K	2024-7-23	修正了 getfilelist 及 getchecksum 的示例
V2.1.2	K	2024-7-30	更新了 filecrc32 的示例
V2.1.3	K	2024-8-16	更新了 readchip 的示例，增加了部分函数返回值类型，修改部分函数说明
V2.1.4	K	2024-11-11	更新函数格式
V2.2.0	K	2024-12-18	更新示例格式，增加命令执行流程,更新说明
V2.2.1	K	2025-02-14	更新 setreplystring 脚本示例， 新增 OpteeQ_GetErrorMsg， OpteeQ_SendCommand，get_proj_CRC， get_proj_checksum

文档约定

1. “力捷丰®”、“OpteeQ®”及相关标识是无锡力捷丰科技有限公司的注册商标。
2. 本手册中的图片均为示例，产品由于型号、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。我们保留更改设计、元件及技术工艺的权利。
3. 本文档以及其修改权、更新权、最终解释权和文档所包含的示例图片等均为无锡力捷丰科技有限公司所有，未经许可不得以任何方式发表、转载、引用。
4. 本文档中出现的所有其他产品名称、服务名称或产品标识属于其各自所有者。
5. 文档版本以官网展示为最新版本。

目 录

1.	文档说明	1
1.1.	目的	1
1.2.	环境	1
1.2.1.	Windows.....	1
1.2.2.	Linux.....	2
1.2.3.	范围	2
2.	工作流程	3
2.1.	连接烧录器	3
2.2.	烧录	4
2.3.	断开	5
2.4.	命令执行	6
3.	函数说明	7
3.1.	OpteeQ_Connect	7
3.2.	OpteeQ_Disconnect	8
3.3.	OpteeQ_GetCommandReply	9
3.4.	OpteeQ_ExecuteScript	10
3.5.	OpteeQ_GetErrorMsg	11
3.6.	OpteeQ_GetFile	12
3.7.	OpteeQ_QueryState	13
3.8.	OpteeQ_QueryAllState	14
3.9.	OpteeQ_SendCommand	15
3.10.	OpteeQ_SendFile	16
3.11.	OpteeQ_SetLogLevel	17
3.12.	OpteeQ_StartLogToFile	18
3.13.	OpteeQ_ReceiveDevLog	19
3.14.	OpteeQ_StopDevLog	20
3.15.	OpteeQ_ChangeLogFile	21
3.16.	OpteeQ_StopExecute.....	22
3.17.	OpteeQ_StopLog.....	23
4.	数据定义	24
4.1.	通道号定义	24
4.2.	烧录器通道状态定义	24
5.	烧录器命令	25
5.1.	说明	25

5.1.1.	Lua 脚本执行流程说明	25
5.1.2.	使用示例	26
5.2.	Lua 脚本命令格式	27
5.2.1.	vccon	27
5.2.2.	vccoff	28
5.2.3.	setvcc	29
5.2.4.	setvpp	30
5.2.5.	getvcc	31
5.2.6.	vppon	32
5.2.7.	vppoff	33
5.2.8.	delayms	34
5.2.9.	loadproject	35
5.2.10.	loadalgorithm	36
5.2.11.	clearalgorithm	37
5.2.12.	runproject	38
5.2.13.	RunFile	39
5.2.14.	setloglevel	40
5.2.15.	setdm	41
5.2.16.	clrdm	42
5.2.17.	getdm	43
5.2.18.	setreplystring	44
5.2.19.	readchip	45
5.2.20.	setautouninit	46
5.2.21.	ReadEmmc	47
5.2.22.	getchecksum	49
5.2.23.	filecrc32	50
5.2.24.	get_proj_CRC	51
5.2.25.	get_proj_checksum	52
5.2.26.	delete	53
5.2.27.	getversion	54
5.2.28.	getfilelist	55
6.	第三方软件调用	56
6.1.	LabView 调用	56
6.1.1.	准备工作	56
6.1.2.	制作示例程序	56
6.2.	C 调用示例	62
6.3.	C#调用示例	63

6.4. Python 调用示例	64
6.5. 命令行调用参数说明	65
7. 帮助与支持	66

1. 文档说明

1.1. 目的

本文档主要介绍力捷丰 SG 系列在线式编程器/烧录器 dll 的接口说明。具体内容包含库函数说明、使用流程、调用方法、注意事项等，用户可通过自己的软件调用 dll 来控制烧录器并完成相关操作。

1.2. 环境

目前支持在 Windows 以及 Linux 系统上调用相关参数。相关非系统的 dll 文件及 so 文件可以在购买设备附带的 U 盘或软件安装完成后的安装文件夹中找到。

1.2.1. Windows

用户在 Windows 系统上使用时，需要调用的 dll 文件有：

- OpteeQDLLx64.dll（64 位）
- OpteeQDLL.dll（32 位）

在 Windows 系统上使用时，需要依赖的系统 dll 文件有：

- libcrypto-3.dll
- WS2_32.dll
- KERNEL32.dll
- MSVCP140.dll
- SHLWAPI.dll
- VCRUNTIME140.dll
- api-ms-win-crt-runtime-l1-1-0.dll
- api-ms-win-crt-stdio-l1-1-0.dll
- api-ms-win-crt-file-system-l1-1-0.dll
- api-ms-win-crt-string-l1-1-0.dll
- api-ms-win-crt-time-l1-1-0.dll

- api-ms-win-crt-heap-l1-1-0.dll
- api-ms-win-crt-convert-l1-1-0.dll

1.2.2. Linux

用户在 Linux 系统上使用时，在加载 libOpteeQ.so 之前需要先手动加载以下三个 so 文件：

- libcrypto.so
- libssh2.so
- libssl.so

其他系统所需要的系统库可以通过 ldd libOpteeQ.so 命令查看。

1.2.3. 范围

本文档适用环境：

- OpteeQDLL.dll 文件版本为 V1.6 及以上版本
- 芯片对应算法（.so）为 V2.0 及以上版本

本文档适用的烧录器型号有：

- SG400
- SG400-M
- T3016

2. 工作流程

烧录器使用的整体工作流程如下图所示。

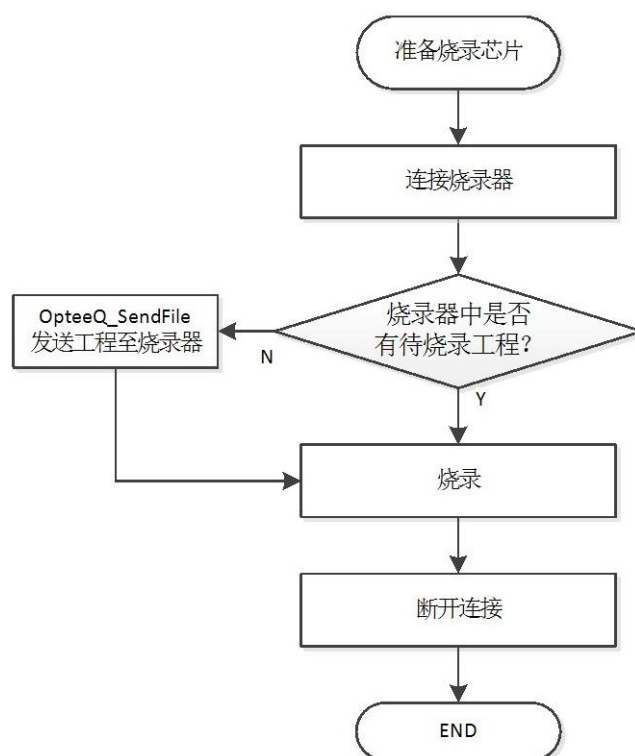


图 2-1 工作流程

说明：若您使用的 DLL 版本小于 V1.11 且设备连接成功后超过 10 分钟没有数据交互，烧录器会断开连接。若要保持设备在线状态可通过添加定期查询通道状态的命令来实现。

2.1. 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。用户程序通过调用 OpteeQ_Connect 函数来完成 PC 机与烧录器之间连接。

2.2. 烧录

通道独立，可支持不同芯片同时进行烧录。在 PC 机与烧录器已经建立连接的前提下，用户程序可以通过函数，使烧录器开始烧录。烧录流程如下：

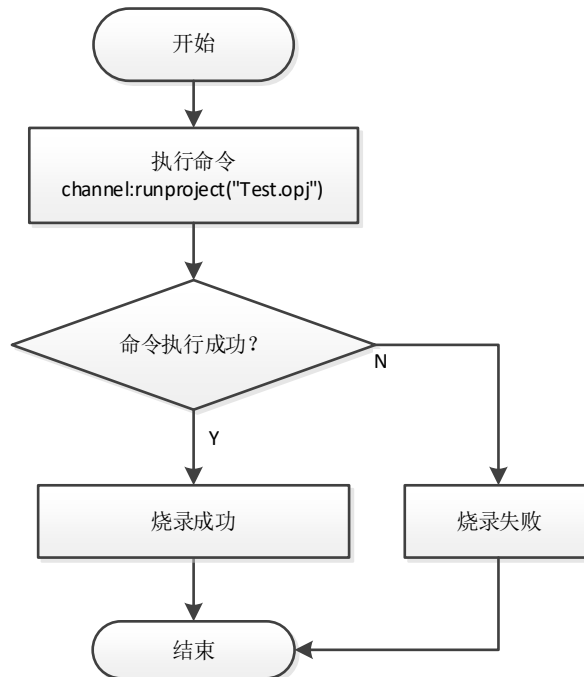


图 2-2 烧录流程

- 若您使用的 DLL 版本低于 V1.11，在调用 dll 文件进行烧录操作的过程中，如果烧录完成后没有断开烧录器过十分钟后再对烧录器进行发送烧录命令会失败，需要重新连接烧录器才能正常运行烧录，您也可以采取每十秒调用 OpteeQ_QueryState 或者 OpteeQ_QueryAllState 查询通道状态来保证程序一直稳定运行。

- 调用 OpteeQ_QueryState 或者 OpteeQ_QueryAllState 查询通道状态，如果状态为 STATUS_BUSY，需要继续查询，并等待状态不为 STATUS_BUSY。

2.3. 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。

说明：当连续烧录时如烧录器出现连接不上的情况，可以尝试在连接（OpteeQ_Connect）前执行断开连接（OpteeQ_Disconnect）的操作。

2.4. 命令执行

通过 OpteeQ_ExecuteScript 函数执行命令的流程如下：

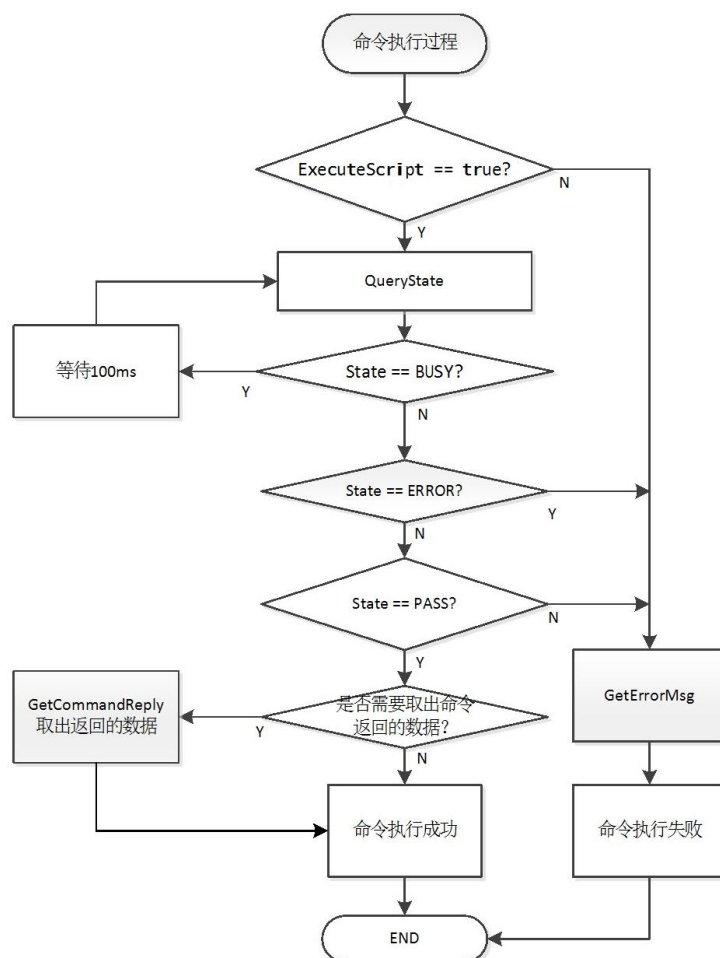


图 2-3 命令执行流程

- 调用 OpteeQ_ExecuteScript 发送烧录命令（runproject）。
- 如果状态为 STATUS_PASS，说明命令执行成功；如果为 STATUS_ERROR，说明发生错误了，可以通过调用 OpteeQ_GetErrorMsg 来获取错误消息字符串。
- 调用 OpteeQ_ExecuteScript 发送 getchecksum 获取校验值之前需先查询烧录的通道状态为 STATUS_PASS，才能执行下一步，以防止还未烧录成功就发送 getchecksum 命令，导致获取校验值失败。

3. 函数说明

3.1. OpteeQ_Connect

函数说明：连接烧录器。

语法

```
bool OpteeQ_Connect(const char *strDev)
```

示例

```
1. bool res = OpteeQ_Connect("LAN192.168.0.100:10023");
```

参数

strDev: 烧录器句柄

返回值

true: 连接成功

false: 连接失败

3.2. OpteeQ_Disconnect

函数说明：断开烧录器。

语法

```
bool OpteeQ_Disconnect (const char *strDev)
```

示例

```
1. bool res = OpteeQ_Disconnect("LAN192.168.0.100:10023");
```

参数

strDev: 烧录器句柄

返回值

true: 断开成功

false: 断开失败

3.3. OpteeQ_GetCommandReply

函数说明：获取命令运行后返回的数据。

语法

```
bool OpteeQ_GetCommandReply(const char *strDev,
                             TARGET eTarget,
                             unsigned char *pBuffer,
                             unsigned int nBufferSize,
                             unsigned int *pnBufferRecvSize,
                             unsigned int nTimeout)
```

示例

```
1. unsigned char buff[1024];
2. unsigned int RecvSize;
3. bool res = OpteeQ_GetCommandReply("LAN192.168.0.100:10023", CH1, buff, 1024, &RecvSize, 100);
```

参数

strDev: 烧录器句柄

eTarget: 通道号, 详见[通道号定义](#)

pBuffer: 用于存放输出数据的缓存

nBufferSize: 缓存空间字节数

pnBufferRecvSize: 实际输出的数据字节数

nTimeout: 超时时间 (ms)

返回值

true: 获取成功

false: 获取失败

3.4. OpteeQ_ExecuteScript

函数说明：向烧录器发送命令。

语法

```
bool OpteeQ_ExecuteScript(const char *devpath,  
                           TARGET eTarget,  
                           const char *strCommandString)
```

示例

```
1. bool res = OpteeQ_ExecuteScript("LAN192.168.0.100:10023", CH1, "channel:run-  
project(\"test.opj\")");
```

参数

devpath: 烧录器的设备路径

eTarget: 通道号，详见[通道号定义](#)

strCommandString: 命令行字符串

返回值

true: 发送成功

false: 发送失败

说明：

1. 命令行字符串定义可参考[烧录器命令](#)。
2. 可通过 [OpteeQ_GetCommandReply](#) 函数获取命令返回值。

3.5. OpteeQ_GetErrorMsg

函数说明：查询错误消息字符串，当通道状态为 ERROR 时，可以通过该函数来获取错误消息字符串。

语法

```
bool OpteeQ_GetErrorMsg(const char *strDev,  
                        TARGET eTarget,  
                        OUT char *strErrorstring)
```

示例

```
1. char strErr[256] = { 0 };  
2. if (OpteeQ_GetErrorMsg("LAN192.168.0.100:10023", CH1, strErr))  
3. {  
4.     printf("Error msg:%s", strErr);  
5. }
```

参数

strDev: 烧录器句柄

eTarget: 通道号，详见[通道号定义](#)

strErrorstring: 用于输出错误信息，需要预留至少 256 字节空间

返回值

true: 查询成功

false: 查询失败

3.6. OpteeQ_GetFile

函数说明：从烧录器上读取文件（读取工程文件/日志文件）。

语法

```
bool OpteeQ_GetFile(const char *strDev,  
                    char *strTargetFile,  
                    char *strLocalFile)
```

示例

```
1. bool res = OpteeQ_Get-  
   File("LAN192.168.0.100:10023", "/proj/test.opj", "D:\\project\\test.opj");
```

参数

strDev：烧录器句柄

strTargetFile：需要读取的文件名，需要带绝对路径，常用文件如：

- 工程文件：/proj/xxx.opj
- 算法文件：/so/xxx.so
- 日志文件：/log/xxx.log

strLocalFile：目标文件名。本地保存的默认路径+文件名如：

D:/OpteeQ_Management_Tool/project/xxx.opj

返回值

true：读取成功

false：读取失败

3.7. OpteeQ_QueryState

函数说明：查询单个通道状态。

语法

```
unsigned int OpteeQ_QueryState(const char *strDev,  
                              TARGET eTarget)
```

示例

```
1. unsigned int state = OpteeQ_QueryState("LAN192.168.0.100:10023", CH1);
```

参数

strDev: 烧录器句柄

eTarget: 通道号，详见[通道号定义](#)

返回值

烧录器的状态码，详见[烧录器通道状态定义](#)，状态码为 0 表示获取状态码失败

3.8. OpteeQ_QueryAllState

函数说明：查询所有通道状态（需要 DLL 版本不低于 V1.9）。

语法

```
bool OpteeQ_QueryAllState(const char *strDev,  
                          unsigned int dwState[20])
```

示例

```
1. unsigned int states[20];  
2. bool res = OpteeQ_QueryAllState("LAN192.168.0.100:10023", states);
```

参数

strDev: 烧录器句柄

dwState: 状态值数组，返回 n+1 个状态值，包括内核和 n 个通道的状态，状态码详见[烧录器通道状态定义](#)，状态码为 0 表示获取状态码失败

返回值

true: 查询成功

false: 查询失败

3.9. OpteeQ_SendCommand

函数说明：获取设备信息。可通过 [OpteeQ_GetCommandReply](#) 接口获取到返回数据。

语法

```
bool OpteeQ_SendCommand(const char *strDev,  
                        TARGET eTarget,  
                        const char *strCommandString)
```

示例

```
1. bool res = OpteeQ_SendCommand("LAN192.168.0.100:10023", KER-  
    NAL, "CCMD GetSN");
```

参数

strDev：烧录器句柄

eTarget：通道号（此处需填 KERNAL），详见[通道号定义](#)

strCommandString：获取设备信息命令行字符串

- 设备 SN：CCMD GetSN
- 设备版本号：CCMD GetTarVersion

返回值

true：获取成功

false：获取失败

3.10. OpteeQ_SendFile

函数说明：向烧录器发送文件（如发送工程）。

语法

```
bool OpteeQ_SendFile(const char *strDev,  
                     char *strLocalFile,  
                     char *strTargetFile)
```

示例

```
1. bool res = OpteeQ_SendFile("LAN192.168.0.100:10023", "D:\\proj-  
   ject\\test.opj", "/proj/test.opj");
```

参数

strDev：烧录器句柄

strLocalFile：本地文件文件名，需要带绝对路径如：

“D:/OpteeQ_Management_Tool/project/xxx.opj”

strTargetFile：默认路径+目标文件名，常用文件如：

- 工程文件： /proj/xxx.opj
- 算法文件： /so/xxx.so
- 日志文件： /log/xxx.log

返回值

true：发送成功

false：发送失败

3.11. OpteeQ_SetLogLevel

函数说明：设置 DLL 日志记录等级，等级范围为 0~5，分别为：调试(0)、消息(1)、警告(2)、错误(3)、系统(4)、不记录(5)。等级越小，记录的信息就越多。

语法

```
bool OpteeQ_SetLogLevel (int nLevel)
```

示例

```
1. bool res = OpteeQ_SetLogLevel(2);
```

参数

nLevel: 日志等级

返回值

true: 设置成功

false: 设置失败

3.12. OpteeQ_StartLogToFile

函数说明： 设置 DLL 日志数据保存的文件，默认的日志文件名是 OpteeQDLL.log，如果需要更改，可以调用函数重新指定日志文件。

语法

```
bool OpteeQ_StartLogToFile (const char *szFilePath)
```

示例

```
1. bool res = OpteeQ_StartLogToFile("OpteeQDLL1.log");
```

参数

szFilePath: 日志文件名称

返回值

true: 设置成功

false: 设置失败

3.13. OpteeQ_ReceiveDevLog

函数说明：获取烧录器实时日志报文并保存在指定文件中（需要 DLL 版本不低于 V1.9）

语法

```
bool OpteeQ_ReceiveDevLog(const char *strDev,  
                           char *strLogPath)
```

示例

```
1. bool res = OpteeQ_Receive-  
   DevLog("LAN192.168.0.100:10023", "D:\\log\\log1.log");
```

参数

strDev：烧录器句柄

strLogPath：日志保存位置

返回值

true：获取成功

false：获取失败

3.14. OpteeQ_StopDevLog

函数说明：停止获取烧录器实时日志（需要 DLL 版本不低于 V1.9）。

语法

```
bool OpteeQ_StopDevLog(const char *strDev)
```

示例

```
1. bool res = OpteeQ_StopDevLog("LAN192.168.0.100:10023");
```

参数

strDev: 烧录器句柄

返回值

true: 停止成功

false: 停止失败

3.15. OpteeQ_ChangeLogFile

函数说明：更改指定烧录器输出的实时日志存储位置（需要 DLL 版本不低于 V1.9）。

语法

```
bool OpteeQ_ChangeLogFile(const char *strDev,  
                           char *strLogPath)
```

示例

```
1. bool res = OpteeQ_ChangeLog-  
   File("LAN192.168.0.100:10023", "D:\\log\\log2.log");
```

参数

strDev：目标烧录器句柄

strLogPath：更改后日志保存位置

返回值

true：更改成功

false：更改失败

3.16. OpteeQ_StopExecute

函数说明：强行停止当前执行脚本。

语法

```
bool OpteeQ_StopExecute(const char* strDev,  
                        TARGET eTarget)
```

示例

```
1. OpteeQ_StopExecute("LAN192.168.0.100:10023", CH1);
```

参数

strDev：烧录器句柄

eTarget：通道号，详见[通道号定义](#)

返回值

true：执行成功

false：执行失败

3.17. OpteeQ_StopLog

函数说明：停止记录 DLL 日志。

语法

```
bool OpteeQ_StopLog()
```

示例

```
1. OpteeQ_StopLog();
```

参数

无

返回值

true: 设置成功

false: 设置失败

4. 数据定义

4.1. 通道号定义

```
typedef enum{  
    KERNAL,          \\0  
  
    CH1,              \\1  
  
    CH2,              \\2  
  
    CH3,              \\3  
  
    CH4,              \\4  
  
    ...  
  
    CH16,             \\16  
  
}TARGET;
```

不同设备对应通道号如下：

- SG400-M: 0、1、2
- SG400: 0、1、2、3、4
- T3016: 0、1、2、3、4、...、16

4.2. 烧录器通道状态定义

```
# define STATUS_BUSY      1 \\ 忙  
  
# define STATUS_READY     2 \\ 就绪，一般上电后为就绪状态  
  
# define STATUS_ERROR     3 \\ 错误  
  
# define STATUS_PASS      4 \\ 成功
```

5. 烧录器命令

5.1. 说明

文档中的“OpteeQ_ExecuteScript”命令中包含了“strCommandString(命令行字符串)”参数，本文档对“命令行字符串”的格式及用法提供了相关说明。

烧录软件通过“OpteeQ_ExecuteScript”接口，接收烧录管理软件命令（通道号说明详见[通道号定义](#)）。

5.1.1. Lua 脚本执行流程说明

OpteeQ_ExecuteScript 适用于算法 SO 版本不低于 V2.0，并且调用的 OpteeQDLL 版本不低于 V1.6 的场景。

烧录器通过 DLL 中的函数 OpteeQ_ExecuteScript，接收 Lua 命令并执行（通道号设置为对应的通道），详见[Lua 脚本命令格式](#)。

此接口只能发送 Lua 命令，Lua 是一种轻量级的高级编程语言，它具有简单、快速和可嵌入型的特点。

Lua 脚本执行流程图详见[命令执行](#)，执行的具体流程如下：

1. 烧录管理软件通过 OpteeQ_QueryState 接口，轮询当前守护进程的状态（接口返回值说明：1 表示“busy”，2 表示“ready”，3 表示“error”，4 表示“pass”）。如果当前状态不是“busy”，则进行下一步，否则请等待。
2. 烧录管理软件通过 dll 函数 OpteeQ_ExecuteScript 发送命令到烧录器，命令详见[Lua 脚本命令格式](#)。
3. 继续通过 OpteeQ_QueryState 接口，轮询当前命令执行的状态。
4. 当守护进程状态为“pass”或“error”时，使用 OpteeQ_GetCommandReply 接口查询命令返回的数据或错误原因。

5.1.2. 使用示例

下面以 checksum 命令为例展示命令调用及获取返回值的方法：

1. 调用 checksum 获取上次烧录的 checksum 值：

```
1. OpteeQ_ExecuteScript("LAN192.168.0.100:10023",CH1,"channel:setreplystring(channel:checksum())")
```

2. 通过 OpteeQ_GetCommandReply 获取返回值：

```
1. # 调用 dll 接口 OpteeQ_GetCommandReply, 获取 checksum 的值。
2. unsigned char checksumbuff[1024];
3. unsigned int RecvSize;
4. OpteeQ_GetCommandReply("LAN192.168.0.100:10023",CH1,checksum-
    buff,1024, &RecvSize,100);
```

说明：Lua 脚本中不区分单引号、双引号，本文档中以双引号为例。

5.2. Lua 脚本命令格式

5.2.1. vccon

命令描述：停止记录 DLL 日志。

语法

```
bool OpteeQ_StopLog()
```

示例

```
1. OpteeQ_StopLog();
```

参数

无

返回值

true: 设置成功

false: 设置失败

5.2.2. vccoff

命令描述：关闭 VCC 输出。

语法

channel:vccoff()

示例

1. channel:vccoff()

参数

无

返回值

无

5.2.3. setvcc

命令描述：设置 VCC 电压。调用此接口，并不会直接输出电压到 VCC 上，需要配合 vccon 一起使用。此电压也是 IO 口的幅值电压。

语法

```
channel:setvcc(int nVol)
```

示例

```
1. channel:setvcc(5000)
```

参数

nVol: VCC 电压，单位是毫伏

返回值

无

5.2.4. setvpp

命令描述：设置 VPP 电压。

语法

channel:setvpp(int nVol)

示例

1. channel:setvpp(5000)

参数

nVol: 设置 VPP 电压，单位是毫伏

返回值

无

5.2.5. getvcc

命令描述：检测 VCC 电压。获取当前 VCC 的电压值，单位毫伏，并把值返回。

语 法

```
unsigned int channel:getvcc()
```

示 例

```
1. vcc_val = channel:getvcc()
```

参 数

无

返回值

返回值为电压值，单位 mV

5.2.6. vppon

命令描述：开启 VPP 电压输出。

语法

```
bool channel:vppon()
```

示例

```
1. channel:vppon()
```

参数

无

返回值

true: 成功

false: 失败

5.2.7. vppoff

命令描述：关闭 VPP 输出。

语法

bool channel:vppoff()

示例

1. channel:vppoff()

参数

无

返回值

true: 成功

false: 失败

5.2.8. delayms

命令描述：延时。延时一段时间（毫秒）。

语法

```
void channel:delayms(unsigned int nms)
```

示例

```
1. channel:delayms(1000)
```

参数

nms: 延时的时长，单位毫秒

返回值

无

5.2.9. loadproject

命令描述：加载芯片工程，但不执行工程。

语法

```
bool channel:loadproject(string filename)
```

示例

```
1. channel:loadproject("test.opj")
```

参数

filename：工程名。需要先通过 [OpteeQ_SendFile](#) 把工程发送到设备上

返回值

true：成功

false：失败

5.2.10. loadalgorithm

命令描述：加载算法。

语法

```
bool channel:loadalgorithm(string Algofile,  
                           string Chipname)
```

示例

```
1. channel:loadalgorithm("test.so", "W25QJ16")
```

参数

Algofile：算法文件名。优先从指定工程路径下寻找算法文件，没找到再从烧录器的算法文件路径下找

Chipname：芯片名称。需要与算法文档里的芯片名对应

返回值

true：成功

false：失败

5.2.11. clearalgorithm

命令描述：卸载算法。卸载已加载的算法，清空 Lua 虚拟机。

语法

```
bool channel:clearalgorithm()
```

示例

```
1. channel:clearalgorithm()
```

参数

无

返回值

true: 成功

false: 失败

5.2.12. runproject

命令描述：执行或烧录工程文件。需确保烧录器中存在烧录工程，若无可使用命令 [OpteeQ_SendFile](#) 命令将工程文件发送到烧录器上。

说明：通道进程版本升至 V1.1.17 以上可兼容老版本工程（.oprj）的命令发送。

语法

```
bool channel:runproject(string proj)
```

示例

```
1. channel:runproject("test.opj")
```

参数

proj: 工程名称

返回值

true: 成功

false: 失败

5.2.13. RunFile

命令描述：执行文件。以 Lua 解释器运行指定文件。

语法

```
bool channel:runfile(string filename)
```

示例

```
1. channel:runfile("test.sct")
```

参数

filename：要运行的脚本文件名称

返回值

true：成功

false：失败

5.2.14. setloglevel

命令描述：设置可输出烧录器日志的等级，只有当输出日志等级大于等于设置的等级，才会输出此日志信息。

语法

```
void channel:setloglevel(int n)
```

示例

```
1. channel:setloglevel(1)
```

参数

n: 输出日志等级（出厂默认值为 1）

0-debug

1-info

2-warn

3-error

4-fault

返回值

无

5.2.15. setdm

命令描述：设置添加的动态数据内容。要烧录的内容格式为以字符串形式表示十六进制数，低地址数据在前，高地址数据在后，长度为 2 的倍数。

语法

```
bool channel:setdm(unsigned int address,  
                    string value)
```

示例

1. # 需要在地址 0x100 开始烧录 6 个字节的数据{0x12,0x34,0x56,0x78,0xab,0xcd}
2. channel:setdm(0x100, "12345678ABCD")

参数

address: 动态数据的首地址

value: 动态数据内容（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: 成功

false: 失败

5.2.16. clrdm

命令描述：清除动态数据。清除通道中已设置的动态数据内容。

语法

```
bool channel:clrdm()
```

示例

```
1. channel:clrdm()
```

参数

无

返回值

true: 成功

false: 失败

5.2.17. getdm

命令描述：获取当前已设置的动态数据，并把结果放在应答数据中。

可通过 [OpteeQ_GetCommandReply](#) 接口获取到结果。

语 法

```
bool channel:getdm()
```

示 例

```
1. channel:getdm()
```

参 数

无

返回值

true: 成功

false: 失败

5.2.18. setreplystring

命令描述：设置应答数据。将要返回字符串放在应答数据中，可通过 [OpteeQ_GetCommandReply](#) 接口的 pBuffer 获取到结果，常与其他 lua 命令搭配使用，获取命令的返回值。

语法

```
bool channel:setreplystring(char *reply)
```

示例

1. #获取上次烧录的 checksum 值
2. checksum = channel:getchecksum()
3. #将 checksum 值作为应答数据返回，可通过 [OpteeQ_GetCommandReply](#) 接口的 pBuffer 获取返回值
4. channel:setreplystring(checksum)

参数

reply: 返回字符串

返回值

true: 成功

false: 失败

5.2.19. readchip

命令描述：读取芯片数据。通过工程读取目标芯片 flash 内容，可通过 [OpteeQ_GetCommandReply](#) 接口获取到返回数据。

语法

```
bool channel:readchip(char *project,  
                      unsigned int start,  
                      unsigned int size)
```

示例

```
1. channel:readchip("template.opj",0,10)
```

参数

project: 可烧录的工程（工程脚本中，必须包含“dev:Init()”）

start: 读取的起始地址

size: 读取的大小，单位字节

返回值

true: 成功

false: 失败

5.2.20. setautouninit

命令描述：输出信号置为高阻态。设置执行完成工程后，是否自动执行 UnInit 将信号输出状态置为高阻态。

语 法

channel:setautouninit(bool en)

示 例

1. channel:setautouninit(true)

参 数

true：执行完 runproject 接口后，会自动执行 UnInit 接口，把 FPGA 恢复到默认状态并关闭 VCC、VPP 输出

false：执行完 runproject 接口后结束

返回值

无

5.2.21. ReadEmmc

命令描述：读取 emmc 数据内容，并上传到 ftp server。

此命令为 SG400-M 专属命令。

语法

```
bool channel:ReadEmmc(string prj,  
                      string read_json)
```

示例

```
1. channel:ReadEmmc('read_test.opj',{'unit':"sector","read ac-  
tion":[{"part":"user data area","start":"0","size":"0x100","tar-  
get file":"ftp:/read.bin"}]})
```

参数

prj: 支持读取数据命令的工程（有判断 readflag 变量以及
"dev:parse_read_json(readjson)")

json: 包含要读取信息的 json 格式字符串，其格式为：

{"unit":"sector","read action":[{"part":"user data
area","start":"0","size":"0x100","target file":"ftp:/read.bin"}]}其参数说明
如下：

1. "unit": 类型为字符串。取值“sector”表示此 json 字符所有的 start 和 size 信息以 sector (512B) 为单位。取值“byte”表示此 json 字符中所有的 start 和 size 信息以 byte 为单位。
2. "part": 类型为字符串，表示所属的分区。可选值为：“boot 1”，“boot 2”，“user data area”，“RPMB”，“GPP1”，“GPP2”，“GPP3”，“GPP4”。
3. "size": 类型为字符串。表示要操作的范围。其值由“0-9”，“a-f”，“A-F”或者为“all”，表示全片。不限十进制数还是十六进制。
4. "start": 类型为字符串。表示 bin 文件操作的起始地址，其值必须是由“0-9”，“a-f”，“A-F”。不限十进制数还是十六进制。
5. "target file": 类型为字符串，存放读取的内容，以“ftp:/"开头，表示上传到 ftp server。

返回值

true: 成功

false: 失败

5.2.22. **getchecksum**

命令描述：获取上次烧录的 checksum 值，可通过 [OpteeQ_GetCommandReply](#) 接口获取到返回数据。

语法

```
unsigned int channel:getchecksum()
```

示例

```
1. checksum = channel:getchecksum()  
2. # 将 checksum 值作为应答数据返回  
3. channel:setreplystring(checksum)
```

参数

无

返回值

返回 checksum 值

5.2.23. filecrc32

命令描述：计算工程文件的 crc32 值，可通过
[OpteeQ_GetCommandReply](#) 接口获取到返回值。

语法

```
unsigned int channel:filecrc32(string filename)
```

示例

```
1. crc = channel:filecrc32("/proj/test.opj")
```

参数

filename: 需要计算的文件名称，需要包含绝对路径

返回值

返回值为文件 CRC 值

5.2.24. get_proj_CRC

命令描述：获取创建工程所用源文件的 crc32 值，可通过 [OpteeQ_GetCommandReply](#) 接口获取到返回值。

语法

```
string get_proj_CRC(string projname)
```

示例

```
1. crc = get_proj_CRC("test.opj")
```

参数

projname: 工程名称（已发送至烧录器中的工程）

返回值

成功返回创建工程所使用源文件的 CRC 值，若使用了多个源文件则会
将多个 CRC 值一起返回

失败返回 nil

5.2.25. get_proj_checksum

命令描述：获取工程中 opd 文件的 checksum 值，可通过 [OpteeQ_GetCommandReply](#) 接口获取到返回值。

语法

```
string get_proj_checksum(string projname)
```

示例

```
1. checksum = get_proj_checksum("test.opj")
```

参数

projname: 工程名称（已发送至烧录器中的工程）

返回值

成功返回工程 opd 文件的 checksum 值，若在烧录过程中未添加动态数据，此值与烧录后获取的 checksum 一致

失败返回 nil

5.2.26. delete

命令描述：删除设备上的文件。此命令仅允许用于删除日志、工程、算法文件。

语法

```
bool channel:delete(string file_name)
```

示例

```
1. channel:delete("/proj/test.opj")
```

参数

file_name: 需要删除的文件名称，需要包含绝对路径

- 工程文件路径 "/proj/"
- 算法文件路径 "/so/"
- 日志文件路径 "/log/"

返回值

true: 成功

false: 失败

5.2.27. **getversion**

命令描述：获取通道进程的版本号。可通过
[OpteeQ_GetCommandReply](#) 接口获取到返回值。

语 法

```
string channel:getversion()
```

示 例

```
1. channel:getversion()
```

参 数

无

返回值

返回值为软件版本号，格式%d.%d.%d，例如：1.1.13

5.2.28. getfilelist

命令描述：获取设备目录下文件列表。

语法

```
string channel:getfilelist(string dir)
```

示例

```
1. channel:getfilelist("/proj")
```

参数

dir：获取文件目录，dir 值如下：

- proj：返回工程目录下的文件列表
- so：返回工程目录下的文件列表
- log：返回日志目录下的文件列表

返回文件以换行符隔开

返回值

成功返回具体内容

失败返回空字符串

6. 第三方软件调用

6.1. 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：\Demo。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。

6.1.1. 准备工作

1. 安装 32 位 2013 以上版本 LabView。
2. 关闭 LabView 软件。
3. 将 OpteeQ_DLL 文件夹复制到 LabView 安装目录的 user.lib 文件夹下。

说明：OpteeQ_DLL 文件夹在本公司提供的 U 盘里，路径为 U 盘：\Demo。

6.1.2. 制作示例程序

6.1.2.1. 连接，断开

打开 LabView，单击上方“文件”菜单，选择“新建 VI”（如下图所示）。



图 6-1 新建 VI

将刚新建的 VI 切换至前面板窗口。在空白处单击鼠标右键，选择“Express”->“按钮与开关”->“文本按钮”（如下图所示），并拖动到前面板上，重复添加 3 个“文本按钮”，并将它们的名称分别修改为“连接”、“断开”、“烧录”。

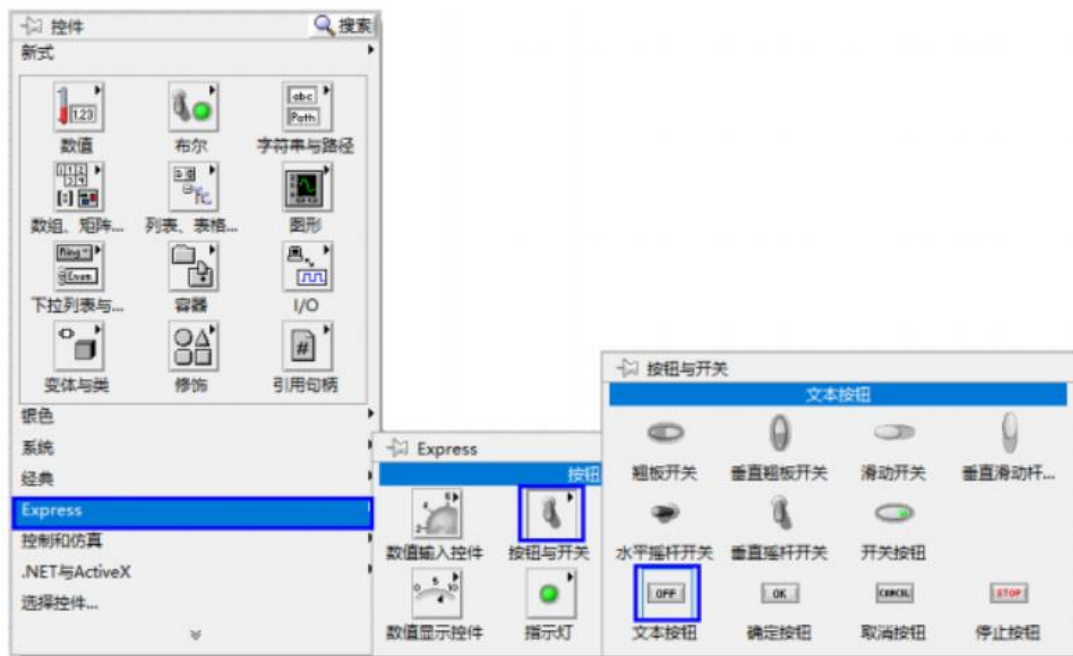


图 6-2 添加文本按钮

在前面板窗口的空白处，单击鼠标右键，选择“布尔”->“圆形指示灯”，并拖动到前面板上，重复添加 5 个“圆形指示灯”。并将它们的名称修改为“连接中”、“通道一”、“通道二”、“通道三”、“通道四”。



图 6-3 添加圆形指示灯

对控件的大小及位置做适当修改，调整后的效果如下图所示。

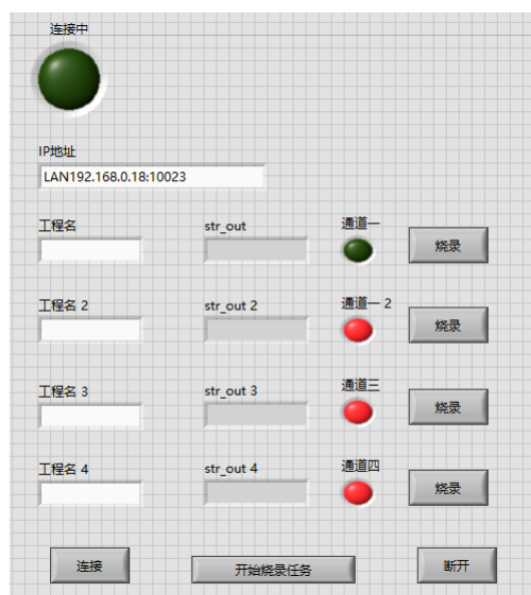


图 6-4 前面板界面

切换至程序框图窗口。在空白处单击鼠标右键，选择“用户库”->“S2G_OpteeQDLL”->“Vis”。可以单击 Vis 窗口左上角的“图钉”把 Vis 窗口固定，以便后续使用。

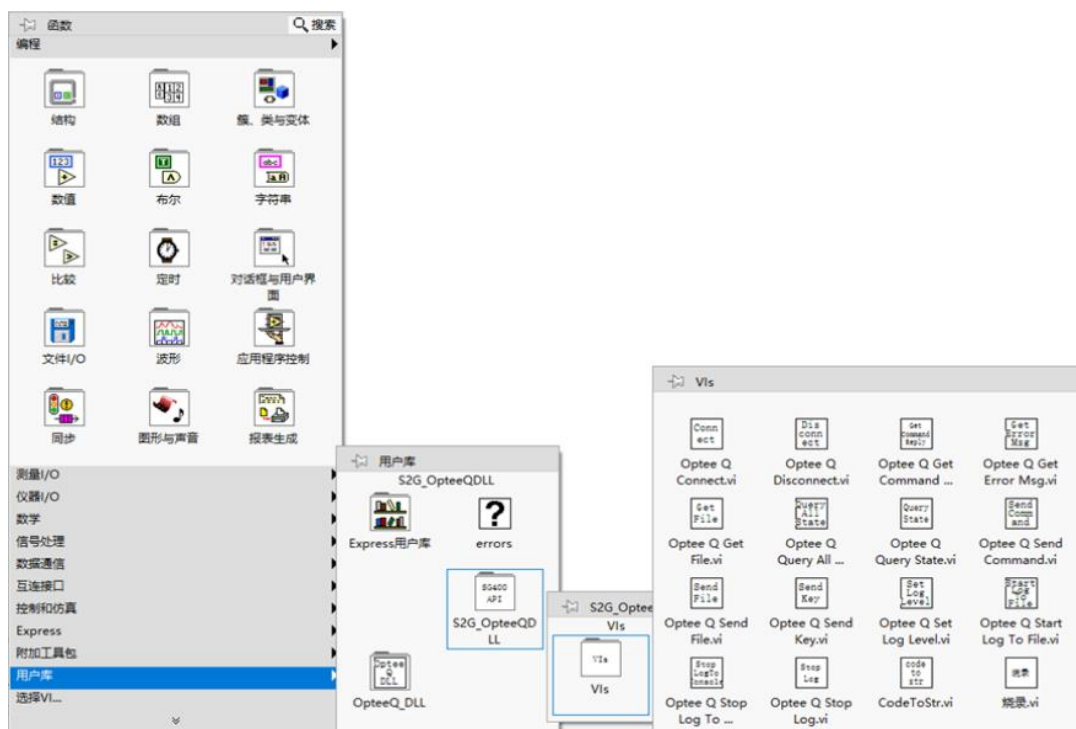


图 6-5 选择 Vis

在程序框图窗口的空白处，单击鼠标右键，选择“结构”->“条件结构”，并拖动到程序框图中。将“连接”按钮与刚添加的“条件结构”相连接。

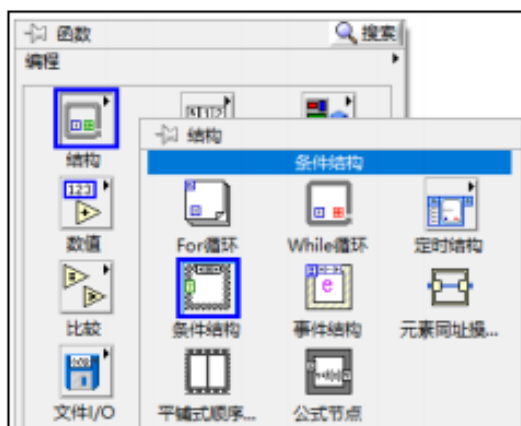


图 6-6 选择条件结构

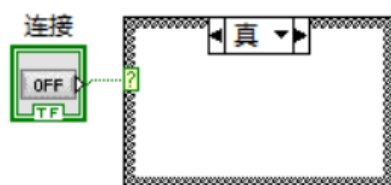


图 6-7 连接条件结构

调用连接函数(connect),并添加相应条件如:IP 地址默认 LAN192.168.0.100, 端口号 10023, 断开只需调用 (disconnect) 函数即可。

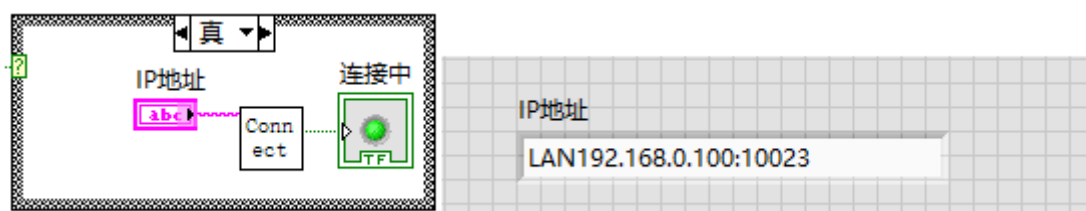


图 6-8 连接条件

6.1.2.2. 烧录

在前面板窗口的空白处，右击选择“布尔”->“确定”按钮，并拖到前面板上，在程序框图空白处添加条件结构，并将按钮和条件框图连接在一起如下图 6-9 所示。在条件框图中添加 ExecuteScript 函数（通过 ExecuteScript 函数发送烧录

指令) 如图 6-10。

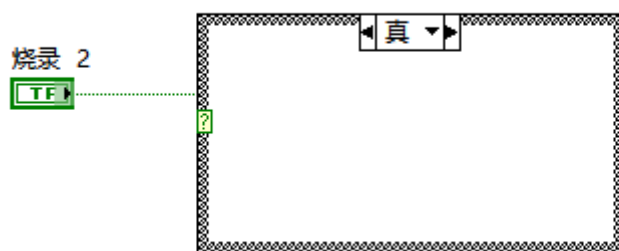


图 6-9 烧录 1

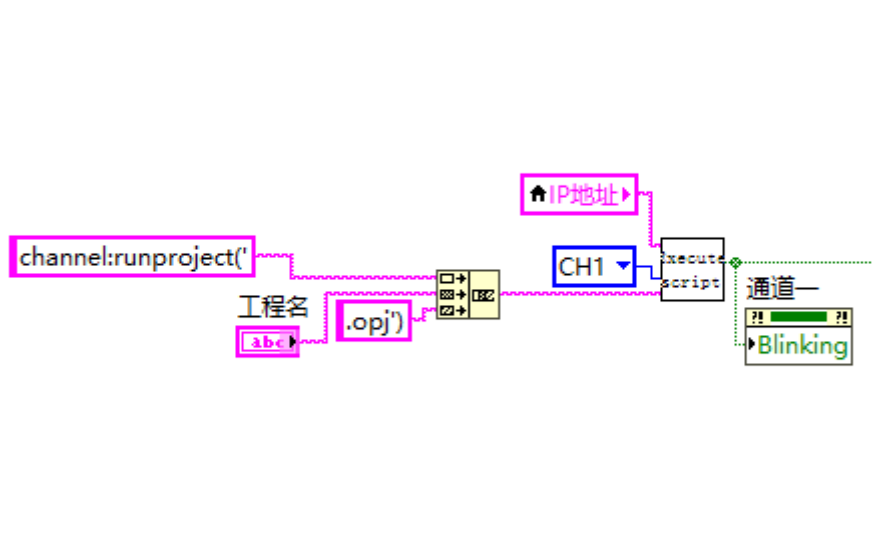


图 6-10 烧录 2

ExecuteScript 函数通过 IP 地址、通道号、以及命令行来进行烧录，烧录的命令行示例如下：channel:runproject('test.opj')，最后输出即可。

6.1.2.3. checksum 值获取

在前面板窗口的空白处，右击选择“布尔”->“确定”按钮，并拖到前面板上，添加字符串显示控件在程序框图空白处添加条件结构，并将按钮和条件框图连接在一起。在条件结构框图中调用 ExecuteScript 函数发送执行获取 checksum 值的命令（channel:getchecksum()），接着用 GetCommandReply 获取执行命令后产生的数据，输出为字符串。

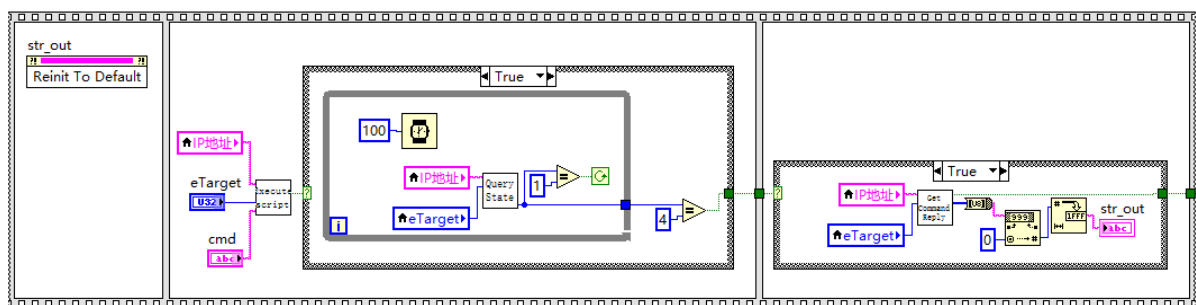


图 6-11checksum 值获取

6.2. 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：\Demo。

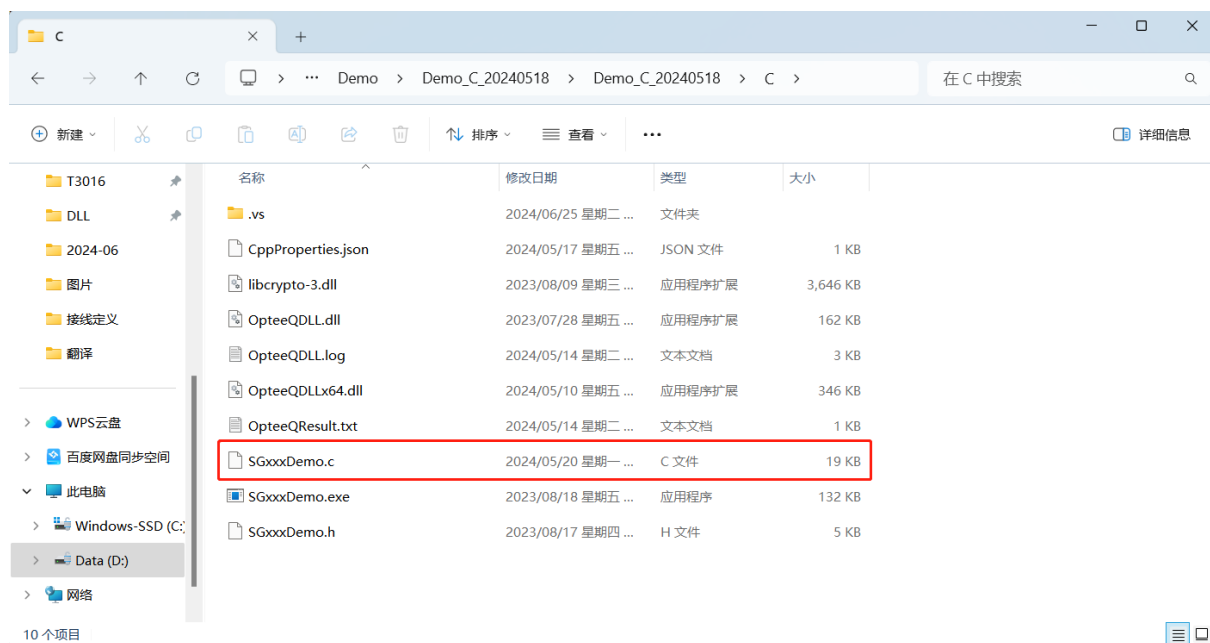


图 6-12C 示例程序

6.3. 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：\Demo。

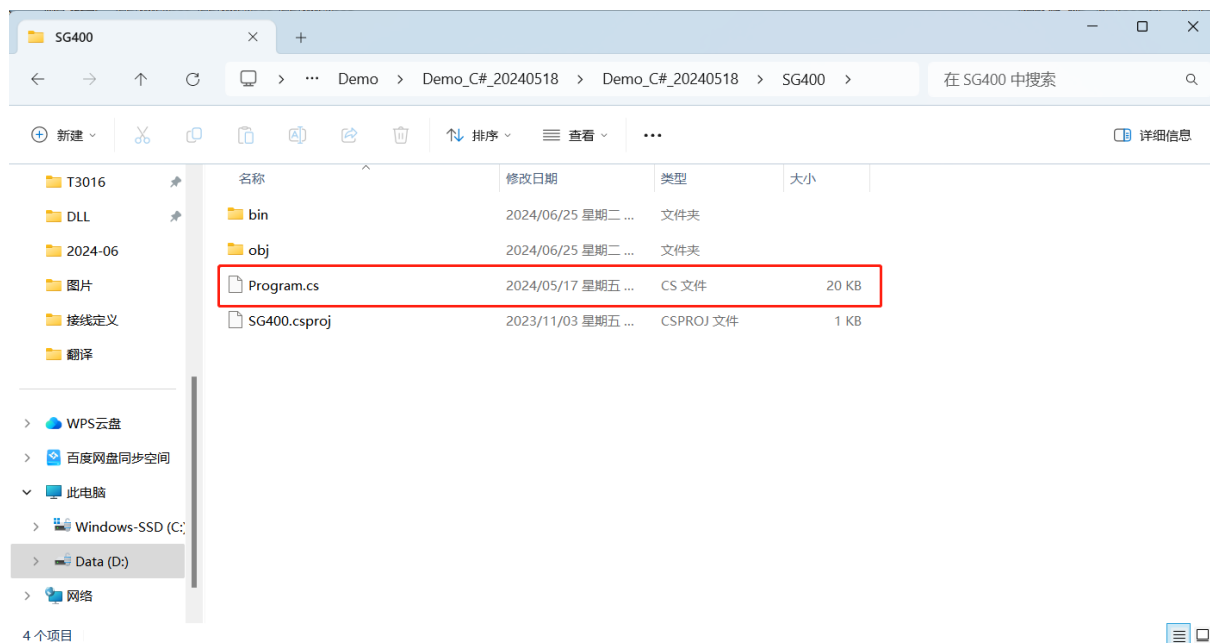


图 6-13C#示例程序

本 Demo 支持 Microsoft Visual Studio 2019 以上版本直接打开。

6.4. Python 调用示例

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

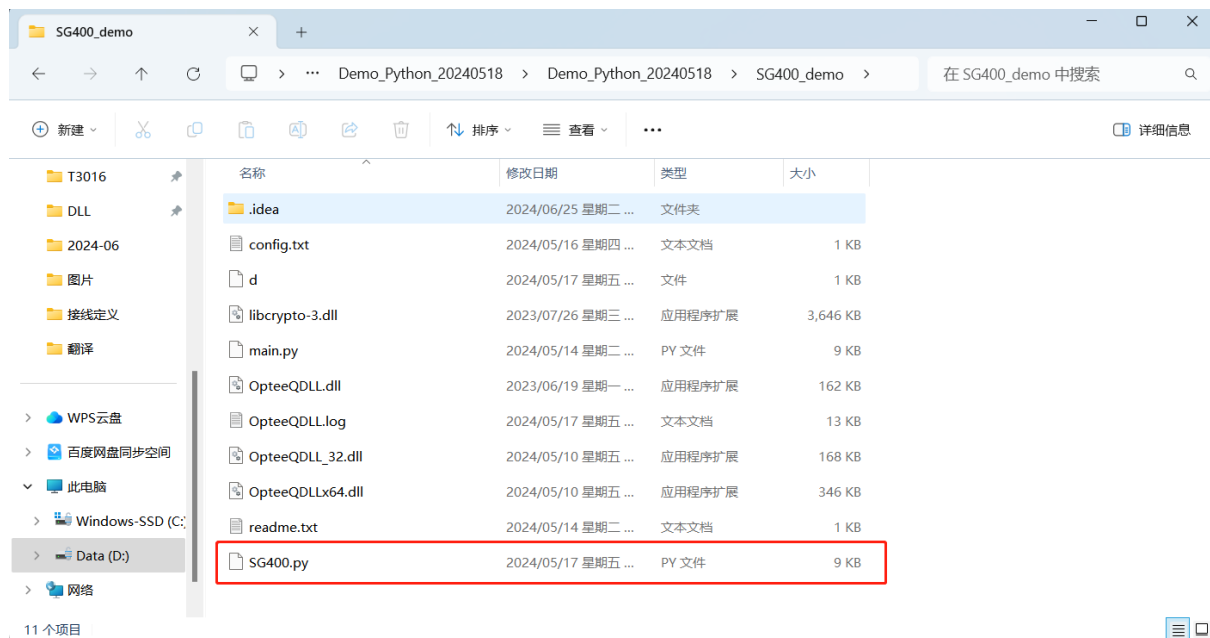


图 6-14 Python 示例程序

工程运行方法：

可用以下命令运行程序

```
SG400.py:python.exe SG400.py -D SG400 -L 192.168.0.18 -C 1 -P W25Q1.oprj
```

参数说明可查看 [命令行调用参数说明](#)。

6.5. 命令行调用参数说明

SG400 提供命令行调用方式进行烧录，调用所需的“OpteeQCMD.exe”可在 OpteeQ 烧录管理软件（版本需要高于 V2.0.7）安装文件夹中找到。

命令示例：

```
"OpteeQCMD.exe -D SG400 -L 192.168.0.100 -P TC212.opj -C 0x3"
```

此命令表示使用网络接口通信，IP 地址为 192.168.0.100，烧录工程为 TC212，不需要烧写序列号，烧录器选择的通道为 1，2。

参数说明：

1) -D

用来指定烧录器设备类型，后面必须跟上烧录器类型，例如“-D SG400”。

2) -L

表示指定烧录器 IP，后面必须跟上烧录器的 IP 地址，例如“-L 192.168.0.100”。

3) -P

用来指定烧录的工程名称，后面必须跟上工程名称，例如“-P TC212.opj”。请确保工程文件已经下发至烧录器。

4) -S

可选参数，用来指定加载的序列号文件路径。例如“-S 1.txt”。如果不指定文件，报错。

5) -C

用来选择烧录器通道号，通道 1 到 4 对应 1 个字节的 BIT1 到 BIT4，例如“-C 0x9”代表烧录器选择通道 1 和通道 4 进行烧录。

7. 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



专业的汽车电子工具链

Professional Tool Chain for Automotive Electronics

OPTEEQ Technologies Co., Ltd

电话: 400 002 3598

0510 8181 3667

邮箱: contact@opteeq.com

网址: www.opteeq.com

