



CAA 软件开发包

版本：v1.0

2024. 12. 30

本手册版权供振旺光电公司所有，公司有权利随时变更本手册的内容而无须声明

目录

一. 介绍.....	4
1.1 头文件和库.....	4
1.2 安装方法.....	4
二. 头文件说明.....	5
2.1 枚举类型和结构体定义.....	5
2.1.1 typedef enum _CAA_ERROR_CODE	5
2.1.2 typedef struct _CAA_ID	5
2.1.3 typedef struct _CAA_TYPE	5
2.1.4 typedef struct _CAA_INFO	5
2.2 函数定义.....	6
2.2.1 CAAGetNum	6
2.2.2 CAAGetProductIDs	6
2.2.3 CAACheck.....	6
2.2.4 CAAGetID	7
2.2.5 CAAOpen	7
2.2.6 CAAGetProperty.....	8
2.2.7 CAAMove.....	8
2.2.8 CAAMoveTo.....	9
2.2.9 CAAStop.....	9
2.2.10 CAAMoving.....	10

2.2.11 CAAGetDegree	11
2.2.12 CAACurDegree.....	11
2.2.13 CAAMinDegree	12
2.2.13 CAASetMaxDegree	12
2.2.14 CAAGetMaxDegree	13
2.2.15 CAAGetTemp	13
2.2.16 CAASetBeep.....	14
2.2.17 CAAGetBeep	14
2.2.18 CAASetReverse	15
2.2.19 CAAGetReverse	15
2.2.20 CAAClose.....	16
2.2.21 CAAGetSDKVersion	16
2.2.22 CAAGetFirmwareVersion.....	17
2.2.23 CAAGetSerialNumber.....	17
2.2.24 CAASetID.....	18
2.2.25 CAAGetType	18
2.2.26 CAASendCMD	19
2.2.27 CAASetSerialNumber.....	19
2.3 建议的调用方式.....	19
三. Demo 使用方式.....	20

修改记录

时间	版本	备注
2025.12.30	v1.0	CAA 软件开发包首次修订

一. 介绍

该软件开发包（SDK）描述了一组可以用来操作 CAA 的使用函数，通过 C、C++、C #等开发工具调用，适用于 x86 或 x64 的 Windows，Linux，和 OSX 操作系统。

1.1 头文件和库

头文件：CAA_API.h

windows 的导入库和动态库：CAA_SRC.lib, CAA_SRC.dll;

Linux 的动态库和静态库：libCAA.so, libCAA.a;

OSX 的动态库和静态库：libCAA.dylib, libCAA.a;

1.2 安装方法

在 Windows 中，下载后解压 zip 文件到任何目录，并添加 DLL 的路径的系统环境变量，有时需要注销并重新登录。还可以将 DLL 置于包含应用程序可执行文件的文件夹中。

在 Linux 和 OSX 中都需要先检查 CAA 的规则文件是否存在。检查路径：
/etc/udev/rules.d/，规则文件名称：caa.rules。不存在就把 caa.rules 放里面，之后重启设备。

在 Linux 中，下载后解压 tar.gz 文件到任意目录，把.so 文件或.a 文件的路径添加到使用目录。

在 OSX 中的使用方式与 Linux 的方式一样。

二. 头文件说明

2.1 枚举类型和结构体定义

2.1.1 typedef enum _CAA_ERROR_CODE

```
typedef enum _CAA_ERROR_CODE
{
    CAA_SUCCESS = 0,           // 成功
    CAA_ERROR_INVALID_INDEX,   // 无效USB ID
    CAA_ERROR_INVALID_ID,      // 无效连接ID
    CAA_ERROR_INVALID_VALUE,   // 无效参数
    CAA_ERROR_REMOVED,         // 未找到CAA设备，可能CAA设备已经被移除
    CAA_ERROR_MOVING,          // CAA移动中
    CAA_ERROR_ERROR_STATE,     // CAA当前状态错误
    CAA_ERROR_GENERAL_ERROR,   // 通用错误
    CAA_ERROR_NOT_SUPPORTED,   // 当前版本该接口不支持
    CAA_ERROR_CLOSED,          // 关闭失败
    CAA_ERROR_OUT_RANGE,       // 超过 0- 360范围
    CAA_ERROR_OVER_LIMIT,      // 超过限位
    CAA_ERROR_STALL,           // 堵转
    CAA_ERROR_TIMEOUT,         // 超时
    CAA_ERROR_END = -1
}CAA_ERROR_CODE; // CAA 相关错误码
```

2.1.2 typedef struct _CAA_ID

```
typedef struct _CAA_ID
{
    unsigned char id[8];
}CAA_ID; // CAA SN 或别名
```

2.1.3 typedef struct _CAA_TYPE

```
typedef struct _CAA_TYPE
{
    char type[16];
}CAA_TYPE; // CAA 的类型
```

2.1.4 typedef struct _CAA_INFO

```
typedef struct _CAA_INFO
```

```
{  
    int ID;  
    char Name[64];  
    int MaxStep;//fixed maximum degree  
} CAA_INFO;
```

2.2 函数定义

2.2.1 CAAGetNum

语法：int CAAGetNum();

功能描述：

此接口用于获取已连接的 CAA 设备数量。可在 CAA 设备连接或断开时调用此接口，以刷新设备列表。

返回值：

返回已连接的 CAA 设备数量。

返回值为 1 表示有 1 个 CAA 设备已连接。

2.2.2 CAAGetProductIDs

语法：int CAAGetProductIDs (int* pPIDs)

功能描述：

此接口用于获取 CAA 的产品 ID。

返回值：

返回 1。

2.2.3 CAACheck

语法：int CAACheck (int iVID, int iPID)

功能描述：

此接口用于检查该设备是否是 CAA，VID 的值是 0x03C3，PID 通过 CAAGetProductIDs 接口获取。

返回值：

返回 1 表示设备是 CAA。

返回 0 表示是其他设备。

2.2.4 CAAGetID

语法：CAA_ERROR_CODE CAAGetID (int index, int* ID)

功能描述：

此接口用于获取连接的 CAA ID，index 是 CAA 的序号，通过 GetNum() 的返回值得到 n，index 的范围是 0 ~ n-1。

返回值：

CAA_ERROR_INVALID_ID

CAA_ERROR_GENERAL_ERROR

CAA_ERROR_REMOVED

CAA_SUCCESS

2.2.5 CAAOpen

语法：CAA_ERROR_CODE CAAOpen (int ID)

功能描述：

此接口用于打开该 ID 的 CAA 设备。

返回值：

CAA_ERROR_INVALID_ID

CAA_ERROR_GENERAL_ERROR

CAA_ERROR_REMOVED

CAA_SUCCESS

2.2.6 CAAGetProperty

语法: CAA_ERROR_CODE CAAGetProperty (int ID, CAA_INFO *pInfo)

功能描述:

此接口用于获取 CAA 的属性。该接口包括打开 CAAOpen 的操作。

int ID: 连接的 ID。

CAA_INFO *pInfo: CAA 的信息, 如果没有别名, 则 name 的内容是"CAA"。

返回值:

CAA_ERROR_INVALID_ID

CAA_ERROR_GENERAL_ERROR

CAA_ERROR_REMOVED

CAA_SUCCESS

2.2.7 CAAMove

语法: CAA_ERROR_CODE CAAMove (int ID, float iAngle)

功能描述:

此接口用于控制 CAA 移动 iAngle 度的接口。

int ID: 连接的 ID。

float iAngle: 移动的角度, 范围在 0 ~ 360° 。

返回值:

CAA_ERROR_INVALID_ID

CAA_ERROR_GENERAL_ERROR

CAA_ERROR_REMOVED

CAA_ERROR_OUT_RANGE

CAA_ERROR_CLOSED

CAA_SUCCESS

2.2.8 CAAMoveTo

语法: CAA_ERROR_CODE CAAMoveTo (int ID, float iAngle)

功能描述:

此接口用于控制 CAA 移动到 iAngle 度。

int ID: 连接的 ID。

float iAngle: 移动的角度, 范围在 0 ~ 360° 。

返回值:

CAA_ERROR_INVALID_ID

CAA_ERROR_GENERAL_ERROR

CAA_ERROR_REMOVED

CAA_ERROR_OUT_RANGE

CAA_ERROR_CLOSED

CAA_SUCCESS

2.2.9 CAAStop

语法: CAA_ERROR_CODE CAAStop (int ID)

功能描述:

此接口用于控制 CAA 停止移动。

返回值:

CAA_ERROR_INVALID_ID

CAA_ERROR_GENERAL_ERROR

CAA_ERROR_REMOVED

CAA_ERROR_CLOSED

CAA_SUCCESS

2.2.10 CAAIsMoving

语法: CAA_ERROR_CODE CAAIsMoving (int ID, bool *pbVal, bool* pbHandControl)

功能描述:

此接口用于检查 CAA 是否在移动。

bool *pbVal: true 移动, false 不移动。

bool* pbHandControl: 是否是手柄控制。

返回值:

CAA_ERROR_INVALID_ID

CAA_ERROR_GENERAL_ERROR

CAA_ERROR_STALL

CAA_ERROR_TIMEOUT

CAA_ERROR_OVER_LIMIT

CAA_ERROR_REMOVED

CAA_ERROR_CLOSED

CAA_SUCCESS

2.2.11 CAAGetDegree

语法: CAA_ERROR_CODE CAAGetDegree (int ID, float* piAngle)

功能描述:

此接口用于获取当前的角度。

float* piAngle: 当前角度。

返回值:

CAA_ERROR_INVALID_ID

CAA_ERROR_REMOVED

CAA_ERROR_CLOSED

CAA_SUCCESS

2.2.12 CAACurDegree

语法: CAA_ERROR_CODE CAACurDegree (int ID, float iAngle)

功能描述:

此接口用于设置当前角度为 iAngle。

float iAngle: 设置的角度。

返回值:

CAA_ERROR_INVALID_ID

CAA_ERROR_MOVING

CAA_ERROR_GENERAL_ERROR

CAA_ERROR_REMOVED

CAA_ERROR_CLOSED

CAA_SUCCESS

2.2.13 CAAMinDegree

语法: ~~CAA_ERROR_CODE CAAMinDegree(int ID, float* piAngle)~~

功能描述:

此接口不再使用。

返回值:

CAA_ERROR_INVALID_ID

CAA_ERROR_REMOVED

CAA_ERROR_CLOSED

CAA_SUCCESS

2.2.13 CAASetMaxDegree

语法: CAA_ERROR_CODE CAASetMaxDegree (int ID, float iAngle)

功能描述:

此接口用于设置 CAA 移动的最大角度。

float iAngle: 最大角度。

返回值:

CAA_ERROR_INVALID_ID

CAA_ERROR_MOVING

CAA_ERROR_GENERAL_ERROR

CAA_ERROR_REMOVED

CAA_ERROR_CLOSED

CAA_SUCCESS

2.2.14 CAAGetMaxDegree

语法: `CAA_ERROR_CODE CAAGetMaxDegree (int ID, float* piAngle)`

功能描述:

此接口用于获取 CAA 可移动的最大角度。

`float* piAngle`: 最大角度。

返回值:

`CAA_ERROR_INVALID_ID`

`CAA_ERROR_MOVING`

`CAA_ERROR_GENERAL_ERROR`

`CAA_ERROR_REMOVED`

`CAA_ERROR_CLOSED`

`CAA_SUCCESS`

2.2.15 CAAGetTemp

语法: ~~`CAA_ERROR_CODE CAAGetTemp (int ID, float* pfTemp)`~~

功能描述:

此接口弃用。

返回值:

`CAA_ERROR_INVALID_ID`

`CAA_ERROR_MOVING`

`CAA_ERROR_GENERAL_ERROR`

`CAA_ERROR_REMOVED`

`CAA_ERROR_CLOSED`

CAA_SUCCESS

2.2.16 CAASetBeep

语法: CAA_ERROR_CODE CAASetBeep (int ID, bool bVal)

功能描述:

此接口用于设置 CAA 的蜂鸣器开关。仅可以控制静止到移动的蜂鸣器状态, 错误情况不可控制。

bool bVal: true 开, false 关。

返回值:

CAA_ERROR_INVALID_ID

CAA_ERROR_MOVING

CAA_ERROR_GENERAL_ERROR

CAA_ERROR_REMOVED

CAA_ERROR_CLOSED

CAA_SUCCESS

2.2.17 CAAGetBeep

语法: CAA_ERROR_CODE CAAGetBeep (int ID, bool* pbVal)

功能描述:

该接口用于获取 CAA 的蜂鸣器状态。

bool *bVal: true 开, false 关。

返回值:

CAA_ERROR_INVALID_ID

CAA_ERROR_MOVING

CAA_ERROR_GENERAL_ERROR

CAA_ERROR_REMOVED

CAA_ERROR_CLOSED

CAA_SUCCESS

2.2.18 CAASetReverse

语法: CAA_ERROR_CODE CAASetReverse (int ID, bool bVal)

功能描述:

此接口用于设置 CAA 的反转。

bool bVal: true 反转, false: 正转。

返回值:

CAA_ERROR_INVALID_ID

CAA_ERROR_MOVING

CAA_ERROR_GENERAL_ERROR

CAA_ERROR_REMOVED

CAA_ERROR_CLOSED

CAA_SUCCESS

2.2.19 CAAGetReverse

语法: CAA_ERROR_CODE CAAGetReverse (int ID, bool* pbVal)

功能描述:

此接口用于获取当前 CAA 反转状态。

bool* pbVal: true 反转, false: 正转

返回值:

CAA_ERROR_INVALID_ID

CAA_ERROR_MOVING

CAA_ERROR_GENERAL_ERROR

CAA_ERROR_REMOVED

CAA_ERROR_CLOSED

CAA_SUCCESS

2.2.20 CAAClose

语法: CAA_ERROR_CODE CAAClose (int ID)

功能描述:

此接口用于断开当前 CAA 的连接, 断开后 CAA 会立刻停止移动。

返回值:

CAA_ERROR_INVALID_ID

CAA_ERROR_MOVING

CAA_ERROR_GENERAL_ERROR

CAA_ERROR_REMOVED

CAA_ERROR_CLOSED

CAA_SUCCESS

2.2.21 CAAGetSDKVersion

语法: char* CAAGetSDKVersion()

功能描述:

此接口用于获取 CAA 的 SDK 的版本。

返回值:

当前的 SDK 版本。

2.2.22 CAAGetFirmwareVersion

语法: CAA_ERROR_CODE CAAGetFirmwareVersion (int ID, unsigned char *major, unsigned char *minor, unsigned char *build)

功能描述:

此接口用于获取 CAA 的设备的固件版本。

返回值:

CAA_ERROR_INVALID_ID

CAA_ERROR_MOVING

CAA_ERROR_GENERAL_ERROR

CAA_ERROR_REMOVED

CAA_ERROR_CLOSED

CAA_SUCCESS

2.2.23 CAAGetSerialNumber

语法: CAA_ERROR_CODE CAAGetSerialNumber (int ID, CAA_SN* pSN)

功能描述:

此接口用于获取 CAA 设备 SN。

返回值:

CAA_ERROR_INVALID_ID

CAA_ERROR_MOVING

CAA_ERROR_GENERAL_ERROR

CAA_ERROR_REMOVED

CAA_ERROR_CLOSED

CAA_SUCCESS

2.2.24 CAASetID

语法: CAA_ERROR_CODE CAASetID (int ID, CAA_ID alias)

功能描述:

此接口用于设置 CAA 设备的别名。

返回值:

CAA_ERROR_INVALID_ID

CAA_ERROR_MOVING

CAA_ERROR_GENERAL_ERROR

CAA_ERROR_REMOVED

CAA_ERROR_CLOSED

CAA_SUCCESS

2.2.25 CAAGetType

语法: CAA_ERROR_CODE CAAGetType (int ID, CAA_TYPE* pCAAType)

功能描述:

此接口用于获取 CAA 设备的类型。

CAA_TYPE* pCAAType: CAA 的类型。例如: CAA-M54

返回值:

CAA_ERROR_INVALID_ID

CAA_ERROR_MOVING

CAA_ERROR_GENERAL_ERROR

CAA_ERROR_REMOVED

CAA_ERROR_CLOSED

CAA_SUCCESS

2.2.26 CAASendCMD

语法: ~~CAA_ERROR_CODE CAASendCMD (int ID, unsigned char* buf, int size, bool bRead = false, unsigned char* readBuf = 0)~~

功能描述:

此接口弃用。

2.2.27 CAASetSerialNumber

语法: ~~CAA_ERROR_CODE CAASetSerialNumber (int ID, CAA_SN* pSN)~~

功能描述:

此接口弃用。

2.3 建议的调用方式

1. 获取 CAA 的连接数量 -> CAAGetNum()
2. 获取 CAA 的 ID -> CAAGetID
3. 获取 CAA 设备的属性-> CAAGetProperty
4. 打开 CAA -> CAAOpen
5. 设置 CAA 移动的位置 -> CAAMove
6. 关闭 CAA -> CAAClose

三. Demo 使用方式

解压 `caa_demo.zip`，具体使用方式见 `caa_demo` 文件夹中的 `README.md`。