

ARCSoft [ALGORITHM SDK QNX]

开发手册



©2022 ArcSoft Corporation Limited. All rights reserved.

本档中所有内容为虹软科技股份有限公司专属所有，均受《中华人民共和国著作权法》、《中华人民共和国商标法》、《中华人民共和国专利法》及国际公约中有关著作权、商标权、专利权及 / 或其它财产所有权法律的保护。本档中任何信息均不得解释为设立任何实质内容的要约或承诺，也不得解释为虹软科技股份有限公司有义务进行磋商或达成任何进一步协议，亦不代表任何权利的转让或授予。

本档仅用于技术评估目的。未经虹软科技股份有限公司书面授权，任何人不得变更、复制、分发、转移或以任何其他方式复制本档中包含的信息，否则以侵权论，依法追究相关法律责任。

All content in this document is the exclusive property of ArcSoft Co., Ltd., and is subject to the copyright, trademark, and trademark rights in the Copyright Law of the People's Republic of China, the Trademark Law of the People's Republic of China, the Patent Law of the People's Republic of China and international conventions. Protection of patent rights and/or other property ownership laws. Any information in this document shall not be construed as an offer or promise to establish any substantial content, nor shall it be construed as an obligation to negotiate or reach any further agreement by ArcSoft Co., Ltd., nor represent any transfer or grant of rights. This document is for technical evaluation purposes only. Without the written authorization of ArcSoft Co., Ltd., no one is allowed to change, copy, distribute, transfer or reproduce the information contained in this document in any other way, otherwise, the relevant legal liability will be investigated for infringement.

修订历史

版本	日期	描述	作者
V1.0.0.1	2021-09-03	初稿 for 合众	ArcSoft
V1.0.0.2	2021-09-26	1.授权增加车型信息 2.头文件支持 C 编译	ArcSoft
V1.0.0.3	2021-11-23	增加 oms 支持的部分	ArcSoft
V1.0.0.6	2021-12-13	Dms oms 子功能开关接口 4.5~4.6	ArcSoft
V1.0.0.10	2022-01-19	增加了一些 SDK 的详细说明	ArcSoft

修订历史	3
1. SDK 概述.....	6
1.1 运行平台	6
1.2 系统要求	6
1.3 支持图像格式	6
2. SDK 授权.....	6
2.1 授权流程	6
3. SDK 相关结构体定义.....	6
3.1 算法引擎 ASD_ALG_HANDLE	6
3.2 图片结构体 ASVLOFFSCREEN.....	7
3.3 事件回调函数	7
3.4 DMS 事件型检测结果结构体.....	7
3.5 OMS 事件型检测结果结构体.....	10
4. SDK 功能接口说明.....	11
4.1 引擎初始化与反初始化	11
4.1.1 初始化引擎.....	11
4.1.2 释放引擎.....	12
4.2 FACEID 相关功能	12
4.2.1 人脸注册提取特征.....	13
4.2.2 人脸识别提取特征.....	13
4.2.3 人脸特征比对.....	14
4.3 DMS 送帧.....	14
4.3.1 输入图片数据.....	15
4.4 OMS 送帧	15
4.4.1 输入图片数据.....	15
4.5 设置 DMS 检测开关.....	16
4.5.1 设置接口.....	16
4.6 设置 OMS 检测开关.....	17
4.6.1 设置接口.....	17
4.7 获取 SDK 版本号	17
4.8 全局错误码定义	18

1. SDK 概述

文档描述了 Arcsoft QNX Algorithm SDK 的递交包说明和使用方法。

递交包:

- SDK 头文件定义
- SDK 动态库
- 开发文档
- Samplecode 使用样例

1.1 运行平台

- QNX aarch64le

1.2 系统要求

- QNX 7.0 及以上版本

1.3 支持图像格式

- DMS Camera: 支持 UYVY 格式的 IR 图片, 人脸方向为 0 或 90 度
- OMS Camera: 支持 UYVY 格式的 IR/RGB 图片, 人脸方向朝上

2. SDK 授权

2.1 授权流程

目前通过 Android 端离线授权 jar 获取 license 字符串。具体可参考 Android 授权 SDK 文档。

如果没有获得授权或者授权失败, 请通过项目联系解决, 否则 SDK 将无法执行。

3. SDK 相关结构体定义

3.1 算法引擎 ASD_ALG_HANDLE

ASD_ALG_HANDLE 算法引擎结构定义:

```
typedef void* ASD_ALG_HANDLE;
```

3.2 图片结构体 ASVLOFFSCREEN

ASVLOFFSCREEN 图片输入结构定义:

```
typedef struct __tag_ASVL_OFFSCREEN
{
    MUInt32 u32PixelFormat;        // 图片格式
    MInt32 i32Width;               // 图片宽度
    MInt32 i32Height;             // 图片高度
    MUInt8* ppu8Plane[4];         // ppu8Plane[0]图片数据指针
    MInt32 pi32Pitch[4];          // pi32Pitch[0]图片间距, 当前为 2* i32Width
}ASVLOFFSCREEN, *LPASVLOFFSCREEN;
```

3.3 事件回调函数

```
typedef void (*ArcEvent)(ASD_EVENT_TYPE eventType, void *msg, int32_t msgLen);
```

通过在初始化引擎的接口中设置该回调函数来获取报警结果。

其中:

DMS 摄像头所产生的事件中, eventType 为 ASD_EVENT_DMS;

OMS 摄像头所产生的事件中, eventType 为 ASD_EVENT_OMS;

```
typedef enum tagASD_EVENT_TYPE {
    ASD_EVENT_DMS = 0,                // DMS 事件报警
    ASD_EVENT_OMS,                   // OMS 事件报警
    ASD_EVENT_MAX,
}ASD_EVENT_TYPE;
```

3.4 DMS 事件型检测结果结构体

```
typedef struct tagASD_DMS_RESULT {
    ASD_ALARM_TYPE alarmType;        // 报警类型
    union {
        ASD_DANGER_ACTION dangerAction;    // 危险动作
        ASD_EXPRESSION_TYPE expressionType; // 表情
    } alarmDetail;
    ASD_DISTRACT_LEVEL distractLevel;    // 分心等级
    ASD_DROWSY_LEVEL drowsyLevel;       // 疲劳等级
};
```

```
ASD_GAZE_AREA    alarmGazeArea;           // 视线区域
ASD_ACTION_TYPE  actionType;              // 头部动作
ASD_GESTURE_TYPE  gestureType;            // DMS 手势
ASD_HEAD_POSE_DATA headPoseData;          // 头部姿态
} ASD_DMS_RESULT;
```

其中报警类型用 `alarmType` 表示，具体包括如下类型：

```
typedef enum tagASD_ALARM_TYPE {
    ASD_ALARM_TYPE_NONE,
    ASD_ALARM_TYPE_DISTRACT,              // 分心
    ASD_ALARM_TYPE_DROWSY,                // 疲劳
    ASD_ALARM_TYPE_DANGER_ACTION,         // 危险动作
    ASD_ALARM_TYPE_EXPRESSION,            // 表情
} ASD_ALARM_TYPE;
```

其中：

如果报警类型为 `DISTRACT`，则具体的分心等级在 `distractLevel` 字段中获取，最大 4 级；

如果报警类型为 `DROWSY`，则具体的疲劳等级在 `drowsyLevel` 字段中获取，最大 4 级；

如果报警类型为 `DANGER_ACTION`，则具体的危险动作在 `dangerAction` 字段中获取；

`ASD_DANGER_ACTION` 类型定义：

```
typedef enum tagASD_DANGER_ACTION{
    ASD_DANGER_ACTION_INVALID = 0,
    ASD_DANGER_ACTION_SMOKE,              // 抽烟
    ASD_DANGER_ACTION_CALL,               // 打电话
    ASD_DANGER_ACTION_NONE,               // 无危险动作
} ASD_DANGER_ACTION;
```

如果报警类型为 `EXPRESSION`，则具体的表情类型在 `expressionType` 字段中获取；

`ASD_EXPRESSION_TYPE` 类型定义：

```
typedef enum {
    ASD_EXPRESSION_TYPE_INVALID = 0,
    ASD_EXPRESSION_TYPE_HAPPY,            // 高兴
    ASD_EXPRESSION_TYPE_ANGRY,            // 愤怒
    ASD_EXPRESSION_TYPE_SURPRISE,         // 惊讶
    ASD_EXPRESSION_TYPE_SAD,              // 哭泣
    ASD_EXPRESSION_TYPE_NORMAL,           // 平静
} ASD_EXPRESSION_TYPE;
```


视线区域结果在 alarmGazeArea 字段中获取;

ASD_GAZE_AREA 类型定义:

```
typedef enum tagASD_GAZE_AREA{  
    ASD_GAZE_AREA_INVALID = 0,                // 无效值  
    ASD_GAZE_AREA_MAIN_DRIVING_WINDSCREEN,     // 主驾驶前风挡  
    ASD_GAZE_AREA_CENTER_CONSOLE,             // 中控屏幕  
    ASD_GAZE_AREA_MAIN_DRIVING_SCREEN,        // 仪表屏幕  
    ASD_GAZE_AREA_LEFT_REARVIEW,              // 左后视镜  
    ASD_GAZE_AREA_RIGHT_REARVIEW,            // 右后视镜  
    ASD_GAZE_AREA_OTHER,                      // 其他区域  
} ASD_GAZE_AREA;
```

手势结果在 gestureType 字段中获取, DMS 目前只支持 MUTE 手势;

ASD_GESTURE_TYPE 类型定义:

```
typedef enum tagASD_GESTURE_TYPE{  
    ASD_GESTURE_TYPE_INVALID = 0,  
    ASD_GESTURE_TYPE_MUTE,                    // 比"嘘"手势, 支持主驾和副驾  
    ASD_GESTURE_TYPE_VSIGN,                   // v 手势  
    ASD_GESTURE_TYPE_OK,                      // OK 手势  
    ASD_GESTURE_TYPE_THUMBUP,                 // 点赞  
    ASD_GESTURE_TYPE_CIRCLE_CW,               // 食指顺时针旋转  
    ASD_GESTURE_TYPE_CIRCLE_CCW,             // 食指逆时针旋转  
    ASD_GESTURE_TYPE_SWIPE_LEFT,              // 手掌向前然后左移  
    ASD_GESTURE_TYPE_SWIPE_RIGHT,            // 手掌向前然后右移  
    ASD_GESTURE_TYPE_SWIPE_FORWARD,           // 手掌向后然后从前往后移动  
    ASD_GESTURE_TYPE_SWIPE_BACKWARD,         // 手掌向前然后从后往前移动  
} ASD_GESTURE_TYPE;
```

头部动作结果在 actionType 字段中获取;

ASD_ACTION_TYPE 类型定义:

```
typedef enum tagASD_ACTION_TYPE{  
    ASD_ACTION_TYPE_NONE = 0,  
    ASD_ACTION_TYPE_NOD,                      // 点头  
    ASD_ACTION_TYPE_SHAKE,                    // 摇头  
} ASD_ACTION_TYPE;
```

头部姿态数据在 headPoseData 字段中获取。

ASD_HEAD_POSE_DATA 类型定义:

```
typedef struct tagASD_HEAD_POSE_DATA {  
    MFloat roll;  
    MFloat yaw;  
    MFloat pitch;  
    MFloat headPoseX;  
    MFloat headPoseY;  
    MFloat headPoseZ;  
} ASD_HEAD_POSE_DATA;
```

由于头部姿态数据需要单帧结果返回，所以 ASD_DMS_RESULT 结果会单帧进行回调。

3.5 OMS 事件型检测结果结构体

OMS 默认打开乘客危险动作，手势和表情检测，结果通过回调函数触发 ASD_EVENT_OMS 事件，并在事件消息体内返回。当打开全车人脸属性、遗留生物和遗留贵重物品检测开关后，相应的检测结果也会以回调的方式返回。下面是 OMS 事件消息结构体定义：

```
typedef struct tagASD_OMS_RESULT {  
    int32_t iFaceCount;                // 乘员个数  
    ASD_OCCUPANT_INFO occupantInfos[ASD_MAX_FACE_NUM]; // 乘员信息  
    ASD_GESTURE_TYPE gestureType;      // OMS 手势类型  
    ASD_BABY_RESULT babyResult;        // 活体检测结果  
    uint32_t hasResidue;                // 0:无遗留物, 1:有遗留物  
    uint32_t hasPet;                   // 0:无宠物遗留, 1:有宠物遗留  
} ASD_OMS_RESULT;
```

其中，ASD_OCCUPANT_INFO 结构体定义如下：

```
typedef struct tagASD_OCCUPANT_INFO {  
    ASD_ALARM_TYPE alarmType;          // 报警类型  
    union {  
        ASD_DANGER_ACTION dangerAction; // 危险动作  
        ASD_EXPRESSION_TYPE expressionType; // 表情类型  
    } alarmDetail;  
    ASD_SEAT_ID seatid;                // 座位编号  
    int32_t age;                       // 乘员年龄
```

```
int32_t gender; // 乘员性别
} ASD_OCCUPANT_INFO;
ASD_SEAT_ID 类型定义:
typedef enum tagASD_SEAT_ID{
    ASD_SEAT_ID_UNKNOWN = 0x0,
    ASD_SEAT_ID_CO_DRIVER = 0x1, // 副驾驶
    ASD_SEAT_ID_REAR_LEFT = 0x2, // 后排左
    ASD_SEAT_ID_REAR_MID = 0x3, // 后排中
    ASD_SEAT_ID_REAR_RIGHT = 0x4, // 后排右
    ASD_SEAT_ID_DRIVER = 0x5, // 驾驶位
} ASD_SEAT_ID;
```

4. SDK 功能接口说明

4.1 引擎初始化与反初始化

SDK 调用算法之前需初始化引擎，调用算法结束（如进程退出）需反初始化引擎。

4.1.1 初始化引擎

接口定义

```
int32_t ASD_InitAlgHandle(ASD_ALG_HANDLE* pHandle, ASD_INIT_PARAM *
pParam, ArcEvent pFunc);
```

接口说明

此接口用于初始化算法引擎。

输入参数

参数名	类型	参数说明	参数类型
pHandle	ASD_ALG_HANDLE*	待初始化的引擎指针	OUT
pParam	ASD_INIT_PARAM	初始化参数	IN
pFunc	ArcEvent	事件报警回调函数	IN

其中初始化参数定义如下：

```
typedef struct tagASD_INIT_PARAM {
    ASD_CAM_MASK camMask; // 所需初始化的功能引擎设置
    int32_t dms_CamAngle; // dms 摄像头角度，仅支持 0 或 90 度
```

```
}ASD_INIT_PARAM;
```

其中 `camMask` 用于控制是否需要初始化 DMS 和 OMS 相关处理引擎，若未初始化 DMS 引擎，则 DMS 送帧、报警功能将不可用；若未初始化 OMS 引擎，则 OMS 送帧、报警，乘员年龄性别，活体检测等功能将不可用。其可选值为：

```
ASD_CAM_MASK_NONE           // 都不初始化
ASD_CAM_MASK_DMS             // 初始化 DMS 功能相关引擎
ASD_CAM_MASK_OMS             // 初始化 OMS 功能引擎
ASD_CAM_MASK_ALL             // 初始化所有功能引擎
```

返回值

返回值	说明
0	成功
非 0	参考 4.8 节错误码定义

4.1.2 释放引擎

接口定义

```
void ASD_UninitAlgHandle(ASD_ALG_HANDLE pHandle);
```

接口说明

此接口用于释放算法引擎。

输入参数

参数名	类型	参数说明	参数类型
pHandle	ASD_ALG_HANDLE	初始化返回的引擎指针	IN

4.2 FaceID 相关功能

FaceID 的功能主要是人脸注册和人脸识别。

人脸注册时，调用人脸注册提取特征的接口，获取到人脸的特征，然后存储在文件或者数据库中。

人脸识别时，调用识别提取特征的接口，获取当前的人脸特征，并与之前同 ID 的特征进行比对（如果没有 ID，则需要按照顺序和之前所有的特征进行比对），如果达到设定的阈值，即认为是识别到，否则认为未识别到本人。相似度阈值设定，会在 `sample code` 中体现，并在需要变更时通知，客户也可以根据实际测试场景调整该阈值。目前两张 IR 图比对的相似度阈值，建议是 0.75。

4.2.1 人脸注册提取特征

接口定义

```
int32_t ASD_FaceRegister(ASD_ALG_HANDLE pHandle, ASVLOFFSCREEN* pImage,  
ASD_CAM_MASK camType, char* pFeatureReg);
```

接口说明

此接口用于注册人脸提取特征，仅支持单人脸注册。

输入参数

参数名	类型	参数说明	参数类型
pHandle	ASD_ALG_HANDLE	算法引擎	IN
pImage	ASVLOFFSCREEN*	输入图片结构体	IN
camType	ASD_CAM_MASK	摄像头标识： ASD_CAM_MASK_DMS 或者 ASD_CAM_MASK_OMS	IN
pFeatureReg	char*	提取的人脸特征值	OUT

返回值

返回值	说明
0	成功
非 0	参考 4.8 节错误码定义

4.2.2 人脸识别提取特征

接口定义

```
int32_t ASD_FaceRecognition(ASD_ALG_HANDLE pHandle, ASVLOFFSCREEN*  
pImage, ASD_CAM_MASK camType, char* pFeatureRegcog);
```

接口说明

此接口用于注册人脸提取特征，可支持多人脸识别，提取图片中最大人脸的特征。

输入参数

参数名	类型	参数说明	参 数 类 型
pHandle	ASD_ALG_HANDLE	算法引擎	IN

pImage	ASVLOFFSCREEN*	输入图片结构体	IN
camType	ASD_CAM_MASK	摄像头标识： ASD_CAM_MASK_DMS 或者 ASD_CAM_MASK_OMS	IN
pFeatureRegcog	char*	提取的人脸特征值	OUT

返回值

返回值	说明
0	成功
非 0	参考 4.8 节错误码定义

4.2.3 人脸特征比对

接口定义

```
int32_t ASD_FaceMatch(ASD_ALG_HANDLE pHandle, const char* pFeatureReg, const char* pFeatureRegcog, double* similarity);
```

接口说明

此接口用于比较两个人脸特征值的相似度。

输入参数

参数名	类型	参数说明	参数类型
pHandle	ASD_ALG_HANDLE	算法引擎指针	IN
pFeatureReg	char*	已注册的人脸特征值	IN
pFeatureRegcog	char*	待识别的人脸特征值	IN
similarity	Double*	0-1 之间，越大相似度越大	IN/OUT

返回值

返回值	说明
0	成功
非 0	参考 4.8 节错误码定义

4.3 DMS 送帧

DMS 部分通过此接口输入连续的数据帧，涉及的功能包括：驾驶员的分心、疲劳、危险动作、表情、手势、视线区域、头部动作、头部姿态数据，结果通过回调函数，以

ASD_EVENT_DMS 事件报警方式输出。

4.3.1 输入图片数据

接口定义

```
int32_t ASD_PushDmsFrame(ASD_ALG_HANDLE pHandle, ASVLOFFSCREEN* pImage,  
int32_t speed);
```

接口说明

此接口用于给算法引擎送帧，用于 DMS 检测。

当有报警事件的时候，检测结果会通过回调 ArcEvent 的方式返回。

输入参数

参数名	类型	参数说明	参数类型
pHandle	ASD_ALG_HANDLE	算法引擎	IN
pImage	ASVLOFFSCREEN*	数据帧	IN
speed	int32_t	当前车速	IN

返回值

返回值	说明
0	成功
非 0	参考 4.8 节错误码定义

4.4 OMS 送帧

OMS 部分通过此接口输入连续的数据帧，涉及的功能包括：OMS 的与占位乘员相关的危险动作、表情、年龄性别检测结果，以及手势、遗留生物和遗留贵重物品检测结果。通过回调函数，以 ASD_EVENT_OMS 事件报警方式输出。

4.4.1 输入图片数据

接口定义

```
int32_t ASD_PushOmsFrame(ASD_ALG_HANDLE pHandle, ASVLOFFSCREEN*  
pImage);
```

接口说明

此接口用于给算法引擎送帧，用于 OMS 检测。

当有报警事件的时候，检测结果会通过回调 ArcEvent 的方式返回。

输入参数

参数名	类型	参数说明	参数类型
pHandle	ASD_ALG_HANDLE	算法引擎	IN
pImage	ASVLOFFSCREEN*	数据帧	IN

返回值

返回值	说明
0	成功
非 0	参考 4.8 节错误码定义

4.5 设置 DMS 检测开关

通过此接口设置打开或关闭 DMS 的算法子功能，默认是打开全部 DMS 算法的检测。

4.5.1 设置接口

接口定义

```
int32_t ASD_SetDmsDetectSwitch(ASD_ALG_HANDLE pHandle, ASD_DETECT_MASK mask);
```

接口说明

此接口用于设置 DMS 子功能检测开关。

输入参数

参数名	类型	参数说明	参数类型
pHandle	ASD_ALG_HANDLE	算法引擎	IN
mask	ASD_DETECT_MASK	默认为 ASD_MASK_DMS_ALL 可选 ASD_MASK_DISTRACT ASD_MASK_DROWSY ASD_MASK_DANGE_ACTION ASD_MASK_NODSHAKE ASD_MASK_GAZEAREA ASD_MASK_EXPRESSION ASD_MASK_GESTURE	IN

返回值

返回值	说明
-----	----

0	成功
非 0	参考 4.8 节错误码定义

4.6 设置 OMS 检测开关

OMS 的有些检测不是一直需要做的，比如年龄性别、遗留生物、遗留贵重物品检测，因此 OMS 的默认检测开关为（ ASD_OMS_MASK_DANGE_ACTION | ASD_OMS_MASK_EXPRESSION | ASD_OMS_MASK_GESTURE ）。其他检测开关在特定条件下才需要打开检测，例如收到开门信号或关门信号。

4.6.1 设置接口

接口定义

```
int32_t ASD_SetOmsDetectSwitch(ASD_ALG_HANDLE pHandle, ASD_DETECT_MASK mask);
```

接口说明

此接口用于设置 OMS 检测开关。

输入参数

参数名	类型	参数说明	参数类型
pHandle	ASD_ALG_HANDLE	算法引擎	IN
mask	ASD_DETECT_MASK	默认为 ASD_MASK_OMS_DEFAULT 可选： ASD_OMS_MASK_DANGE_ACTION ASD_OMS_MASK_EXPRESSION ASD_OMS_MASK_GESTURE ASD_OMS_MASK_BABY ASD_OMS_MASK_PET ASD_OMS_MASK_RESIDUE ASD_OMS_MASK_AGE_GENDER	IN

返回值

返回值	说明
0	成功
非 0	参考 4.8 节错误码定义

4.7 获取 SDK 版本号

接口定义

```
const char* ASD_GetSDKVersion();
```

接口说明

此接口用于获取 SDK 的版本号，方便检查版本更新是否替换成功。

输入参数

无。

返回值

返回 SDK 版本号。

4.8 全局错误码定义

错误码	说明
0	执行正确
-50000	内部引擎错误
-50001	参数校验失败
-50002	授权校验失败
-50003	内存错误
-50010	DMS 错误
-50011	DMS 初始化错误
-50020	OMS 错误
-50021	OMS 初始化错误
-50100	FACEID 错误
-50101	FACEID 初始化错误
-50102	人脸检测错误
-50103	多人脸错误
-50104	人脸质量错误
-50105	人脸质量-角度错误

-50106	人脸质量-遮挡错误
-50107	人脸质量-光线错误
-50108	人脸质量-其他错误
-50109	人脸活体错误
-50110	人脸识别错误
-50111	无人脸错误
-50112	人脸属性错误
-50113	人脸特征值错误