



ADAS 应用 API 参考手册

文档版本号: v1.0

发布日期: 2018-11-01

版权所有 © 珠海全志科技股份有限公司 2018。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



、全志和其他全志商标均为珠海全志科技股份有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受全志公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，全志公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保

前言

概述

本文档为 ADAS 开发设计工程师提供 SDK 开发指导。

产品版本

与本文档对应的产品版本。

产品名称	产品版本
V316	ADAS V2.2.1

读者对象

本文档（本指南）主要适用于以下工程师：

使用全志 ADAS 进行开发的软件开发工程师

名词解释

API - 应用程序接口, 全称: Application Programming Interface

SDK - 软件开发包, 全称: Software Development Kit

ADAS - 高级驾驶辅助系统, 全称: Advanced Driver Assistance Systems, ADAS 是利用安装在车上的各式传感器, 在汽车行驶过程中随后感应周围的环境, 收集数据, 进行静态、动态物体的辨识、侦测与追踪, 并结合导航仪地图数据, 进行系统的运算与分析, 从而预先让驾驶者察觉到可能发生的危险, 有效增加汽车驾驶的舒适性与安全性。其核心功能主要为前车碰撞预警 (FCW)、车道偏离预警 (LDW) 等功能。

LDW - 车道偏离警示, 英文全称: Lane Departure Warning

FCW - 前车碰撞预警, 英文全称: Forward Collision Warning

修订记录

版本号	修订日期	修订内容
ADAS V2.2.1	2018-11-01	初稿

Allwinmertechn

目 录

ADAS 应用 API 参考手册.....	1
文档版本号: v1.0.....	1
发布日期: 2018-11-01.....	1
指导, 本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保.....	1
1. 概述.....	4
1.1 概述.....	4
1.2 功能描述.....	4
1.3 ADAS 典型场景指标.....	4
2. ADAS 应用 API 参考.....	6
2.1 AW_AI_ADAS_Init.....	6
2.2 AW_AI_ADAS_UnInit.....	7
2.3 AdasIn 回调函数.....	8
2.4 AdasOut 回调函数.....	10
3. 数据结构参考.....	12
3.1 宏定义.....	12
3.2 关键数据结构.....	12
ADASInData.....	12
ADASOutData.....	15
ADASPara.....	19
4. 调用例程.....	22

1. 概述

1.1 概述

全志ADAS是针对车载系统专门定制开发一款高级智能驾驶辅助系统，可应用于车辆高速公路、城市道路车辆预警。全志ADAS预警具有准确率较高，功耗低等特点，能辅助驾驶员提高行车安全性。

1.2 功能描述

1. 全天候预警，支持昼/夜检测模式
2. 高预警准确率
白天>95%
3. 支持前车碰撞预警
4. 支持车道偏离预警

1.3 ADAS 典型场景指标

场景一：市区内主干道, CPUCOSTLEVEL = 1, 适合 V5



典型场景指标	说明
分辨率	640 x 360 像素
功能	全天候 ADAS FCW（前车碰撞预警）

	LDW（车道偏离预警）
CPU 规格	Quad-core Cortex A7 1.5GHz
单帧 CPU 耗时	前车 2 辆，21.82ms
CPU 消耗	< 32%
内存消耗	79MB
带宽消耗	70MB/s

表1. 3. 1 场景1典型消耗

场景一：市区内主干道, CPUCOSTLEVEL = 0, 适合 V5-V200



典型场景指标	说明
分辨率	640 x 360 像素
功能	全天候 ADAS FCW（前车碰撞预警） LDW（车道偏离预警）
CPU 规格	Quad-core Cortex A7 1.5GHz
单帧 CPU 耗时	前车 2 辆，14.71ms
CPU 消耗	< 24%
内存消耗	79MB
带宽消耗	70MB/s

表1. 3. 2 场景2典型消耗

2. ADAS 应用 API 参考

ADAS 接口函数列表	功能描述
AW_AI_ADAS_Init	初始化 ADAS
AW_AI_ADAS_UnInit	反初始化 ADAS
AdasIn	ADAS 回调输入
AdasOut	ADAS 回调输出

2.1 AW_AI_ADAS_Init

【描述】

初始化 ADAS

【语法】

```
void AW_AI_ADAS_Init(ADASPara *ptrADASPara, void *dev)
```

【参数】

参数名称	描述	输入/输出
ptrADASPara	ADAS 配置参数	输入
dev	用户参数	输入

【返回值】

参数名称	描述	返回值
void	无	---

【参考头文件】

ADASAPI.h

【例程】

```
ADASPara adasPara;
int width = 640;
int height = 360;
adasPara.frameWidth = width; //视频宽度
adasPara.frameHeight = height; //视频高度
adasPara.roiX = 0; //感兴趣区域 X 坐标
adasPara.roiY = 0; //感兴趣区域 Y 坐标
adasPara.roiH = height; //感兴趣区域高度
adasPara.roiW = width; //感兴趣区域宽度
```



```

adasPara.fps = 30; //视频帧率
adasPara.focalLength = 2800; //焦距 um
adasPara.pixelSize = 3; //SENSOR 单个像元大小 um
adasPara.horizonViewAngle = 120; //水平视场角
adasPara.verticalViewAngle = 90; //垂直视场角
adasPara.vanishX = width/2; //天际消失点 X 坐标
adasPara.vanishY = height/2; //天际消失点 Y 坐标
adasPara.hoodLineDist = 120; //车头距离
adasPara.vehicleWidth = 182; //车辆宽度
adasPara.wheelDist = 160; //车轮距离
adasPara.cameraToCenterDist = 0; //摄像头中心距离
adasPara.cameraHeight = 120; //相机安装高度
adasPara.leftSensity = 1; //左侧车道线检测敏感度
adasPara.rightSensity = 1; //右侧车道线检测敏感度
adasPara.fcwSensity = 1; //前车碰撞预警敏感度
adasPara.awAdasIn = awAdasIn; //ADAS 输入回调函数
adasPara.awAdasOut = awAdasOut; //ADAS 输出回调函数
//初始化 ADAS
AW_AI_ADAS_Init(&adasPara,NULL);

```

【备注】

暂无

2.2 AW_AI_ADAS_UnInit

【描述】

反初始化 ADAS

【语法】

```
void AW_AI_ADAS_UnInit();
```

【参数】

参数名称	描述	输入/输出
---	---	---

【返回值】

参数名称	描述	返回值
void	无	---

【参考头文件】

ADASAPI.h

【例程】

```
//释放 ADAS
AW_AI_ADAS_UnInit();
```

【备注】

暂无

2.3 AdasIn 回调函数

【描述】

ADAS 回调输入

【语法】

```
typedef void (*AdasIn)(ADASInData *ptrADASInData,void *dv);
```

【参数】

参数名称	描述	输入/输出
ptrADASInData	ADAS 输入结构体	输入
dev	用户参数	输入

【返回值】

参数名称	描述	返回值
void	无	——

【参考头文件】

ADASAPI.h

【例程】

```
//ADAS 输入参数
adasInData.ptrFrameData = adasData.ptrImgBuffer; //图像数据
adasInData.gpsSpeed = 80; //GPS 数据, KM/S
adasInData.gpsSpeedEnable = 0; //GPS 速度使能
adasInData.GsensorStop = 1; //加速度传感器停止标记
```

```
adasInData.GsensorStopEnable = 0; //加速度传感器数据停止使能标记
adasInData.sensity = 2; //检测敏感度
adasInData.CPUCostLevel = 1; //CPU 消耗等级
adasInData.luminanceValue = 1500; //光照强度
adasInData.lightSignal = 0; //车辆转向灯信号
adasInData.lightSignalEnable = 0; //信号灯使能信号
adasInData.cameraHeight = 1200; //摄像机离地高度,单位 mm
adasInData.ptrFrameDataArray = (unsigned char *)malloc(width * height*3/2); //彩色图像数
据指针

adasInData.dataType = 2; //彩色数据类型, NV21
adasInData.cameraType = 0; //摄像头类型
adasInData.imageWidth = width; //图像宽度
adasInData.imageHeight = height; //图像高度
adasInData.cameraNum = 1; //摄像机数目
adasInData.Lng = 'E'; //GPS 经度信息
adasInData.LngValue = 113.46;
adasInData.Lat = 'N'; //GPS 纬度
adasInData.LatValue = 22.27;
adasInData.altitude = 20.f; //GPS 海拔值
adasInData.deviceType = 1; //设备类型
strcpy(adasInData.sensorID, "OVXXXX"); //传感器类型
adasInData.networkStatus = 0; //连接类型
//回调输入
void awAdasIn(ADASInData *ptrADASInData,void *dv)
{
    sigset_t set1;
    sigemptyset(&set1);
    sigaddset(&set1, SIGIO);
    pthread_sigmask(SIG_UNBLOCK, &set1, NULL);
    int ret = sigpending(&set1);
    if(bInit == 0)
    {
        printf("image buffer is not ready #####\n");
    }
}
```

```
else
{
    sem_wait(&adasData.semInData);
    pthread_mutex_lock(&adasData.dataTransferLock);
    memcpy(adasData.ptrImgBuffer,adasData.ptrImgGray,adasData.width*asData.height);
    memcpy(ptrADASInData,&adasData.adasInData,sizeof(ADASInData));
    adasData.dataADASIn = 1;
    pthread_mutex_unlock(&adasData.dataTransferLock);
}
}
```

【备注】

暂无

2.4 AdasOut 回调函数

【描述】

ADAS 输出回调

【语法】

```
typedef void (*AdasOut)(ADASOutData *ptrADASOutData,void *dv)
```

【参数】

参数名称	描述	输入/输出
ptrADASOutData	ADAS 输出结构体	输出
dev	用户参数	输出

【返回值】

参数名称	描述	返回值
void	无	——

【参考头文件】

ADASAPI.h

【例程】

//ADAS 回调输出

```
void awAdasOut(ADASOutData *ptrADASOutData,void *dv)
{
```

```
sigset_t set1;
sigemptyset(&set1);
sigaddset(&set1, SIGIO);
pthread_sigmask(SIG_UNBLOCK, &set1, NULL);
int ret = sigpending(&set1);
if(bInit == 0)
{
    printf("image buffer is not ready #####\n");
}
else
{
    pthread_mutex_lock(&adasData.dataTransferLock);
    if(NULL!=ptrADASOutData)
    {
        memcpy(&adasData.adasOutData, ptrADASOutData, sizeof(ADASOutData));
    }
    adasData.dataBackNum = 1;
    pthread_mutex_unlock(&adasData.dataTransferLock);
    sem_post(&adasData.semOutdata);
}
}
```

【备注】

暂无

3. 数据结构参考

3.1 宏定义

【说明】

ADAS 相关宏定义

【定义】

```
#define MAX_CAR_NUM 10    //最大检测车辆数目
#define MAX_PLATE_NUM 5   //最大检测车牌数目（暂不支持）
#define MAX_INDEX_NUM 30 //车道线条状块分割点的个数最大值
```

3.2 关键数据结构

成员	功能描述
ADASInData	ADAS 输入参数，用于 AdasIn 回调
ADASOutData	ADAS 输出结果，用于 AdasOut 回调
ADASPara	初始化 ADAS 配置参数

ADASInData

【说明】

ADAS 输入参数

【定义】

```
typedef struct _ADASInData
{
    //图像帧数据
    unsigned char *ptrFrameData;

    //GPS 数据
    unsigned char gpsSpeed;//KM/H 千米每小时
    unsigned char gpsSpeedEnable;//1-gps 速度有效 0-gps 速度无效

    //Gsensor 数据
    unsigned char GsensorStop;//
    unsigned char GsensorStopEnable;//
```

//敏感度 每个功能占用两位

```
unsigned int sensity;//01
```

//CPUCostLevel 每个功能占用两位

```
unsigned int CPUCostLevel;
```

//光照强度 LV

```
unsigned int luminanceValue;
```

```
//车辆转向灯信号
```

```
unsigned char lightSignal;//0-无信号 1-左转向 2-右转向
```

```
unsigned char lightSignalEnable;
```

```
//相机高度
```

```
unsigned int cameraHeight;//mm
```

```
//图像帧数据
```

```
unsigned char *ptrFrameDataArray;//彩色图像数据指针
```

```
unsigned char dataType;//0-YUV 1-NV12 2-NV21 3-RGB...
```

```
unsigned char cameraType;//摄像头类型(0-前置摄像头 1-后置摄像头 2-左侧摄像头 3-右侧摄像头...)
```

```
unsigned int imageWidth;
```

```
unsigned int imageHeight;
```

```
unsigned int cameraNum;
```

//GPS 经纬度信息

```
//GPS 经度
```

```
unsigned char Lng;//E W
```

float LngValue;

```
//GPS 维度
```

```
unsigned char Lat;//N S
```

float LatValue;

```
//GPS 海拔
```

```
float altitude;
```

```
//设备类型(0-行车记录仪 1-后视镜...)
```

```
unsigned char deviceType;
```

```
//sensor 类型
```

```
char sensorID[255];
```

```
unsigned char networkStatus;//0-端口 1-连接
```

```
/*  
*****  
*/
```

```
}ADASInData;
```

【变量】

变量	含义	说明
ptrFrameData	图像帧数据	
gpsSpeed	GPS 数据	单位：千米每小时，负数代表 GPS 信号无效
gpsSpeedEnable	GPS 速度使能	1-gps 速度有效 0-gps 速度无效
GsensorStop	加速度传感器停止标记	0-开启 1-停止，默认为 1
GsensorStopEnable	加速度传感器数据停止使能标记	0-关闭 1-使能 默认为 0
sensity	检测敏感度	00 - 低敏感 01 - 中敏感 10 - 高敏感 FCW 占用低 0-1 数据 BIT 位 LDW 占用低 2-3 数据 BIT 位
CPUCostLevel	CPU 消耗等级	0 - CPU 消耗低 1- CPU 消耗中 2- CPU 消耗高 默认为 0
luminanceValue	光照强度	白天：1500 Lux，夜晚：300 Lux,由 ISP 数据制定
lightSignal	车辆转向灯信号	0-无信号 1-左转向 2-右转向
lightSignalEnable	信号灯使能信号	0-关闭 1-开启
cameraHeight	摄像机离地高度	单位 mm

ptrFrameDataArray	彩色图像数据指针	多路 ADAS 升级使用（备用）
dataType	彩色数据类型	0-YUV 1-NV12 2-NV21 3-RGB
cameraType	摄像头类型	0-前置摄像头 1-后置摄像头 2-左侧摄像头 3-右侧摄像头
imageWidth	图像宽度	ptrFrameData 和 ptrFrameDataArray 宽度一致
imageHeight	图像高度	ptrFrameData 和 ptrFrameDataArray 高度一致
cameraNum	摄像机数目	
Lng	GPS 经度	GPS 经度信息（E/W）
LngValue	GPS 经度值	GPS 经纬度信息
Lat	GPS 纬度	GPS 经纬度信息
LatValue	GPS 纬度值	GPS 经纬度信息
altitude	GPS 海拔值	GPS 海拔
deviceType	设备类型	0-行车记录仪 1-后视镜
sensorID	传感器类型	如：“OVxxx”
networkStatus	连接类型	0-端口 1-连接

【备注】

ADASAPI.h

ADASOutData

【说明】

ADAS 输出结果

【定义】

```
typedef struct _ADASOutData
```

```
{
```

```
    //版本信息
```

```
    unsigned char *ptrVersion;
```

```
    //左车道线两点
```

```
    unsigned int leftLaneLineX0;
```

```
    unsigned int leftLaneLineY0;
```

```
    unsigned int leftLaneLineX1;
```

```
unsigned int leftLaneLineY1;
```

```
//右车道线两点
```

```
unsigned int rightLaneLineX0;
```

```
unsigned int rightLaneLineY0;
```

```
unsigned int rightLaneLineX1;
```

```
unsigned int rightLaneLineY1;
```

```
//车道线报警信息
```

```
unsigned char leftLaneLineWarn;//0-黄色（未检测到） 1-绿色（检测到） 2-红色（压线报警）
```

3-请重新标定摄像头

```
unsigned char rightLaneLineWarn;//0-黄色（未检测到） 1-绿色（检测到） 2-红色（压线报
```

警）3-请重新标定摄像头

```
//车道线绘图信息===
```

```
int colorPointsNum; //车道线条状块分割点的个数
```

```
unsigned char dnColor; //下方一块的颜色。 1-蓝，2-绿
```

```
unsigned short rowIndex[MAX_INDEX_NUM]; //车道线条状块分割点的行坐标
```

```
unsigned short ltColIndex[MAX_INDEX_NUM]; //车道线条状块左边分割点的列坐标
```

```
unsigned short mdColIndex[MAX_INDEX_NUM]; //车道线条状块中间分割点的列坐标
```

```
unsigned short rtColIndex[MAX_INDEX_NUM]; //车道线条状块右边分割点的列坐标
```

```
//车道线绘图信息==END
```

```
//=====车辆信息
```

```
//车辆位置信息
```

```
unsigned int carX[MAX_CAR_NUM];
```

```
unsigned int carY[MAX_CAR_NUM];
```

```
unsigned int carW[MAX_CAR_NUM];
```

```
unsigned int carH[MAX_CAR_NUM];
```

```
float carTTC[MAX_CAR_NUM];
```

```
float carDist[MAX_CAR_NUM];
```

```
//车辆报警信息
```

```
unsigned char carWarn[MAX_CAR_NUM];
```

```
int carNum;
```

```
//=====车牌信息
```

```
unsigned int plateX[MAX_PLATE_NUM];
```

```
unsigned int plateY[MAX_PLATE_NUM];
```

```
unsigned int plateW[MAX_PLATE_NUM];
```

```
unsigned int plateH[MAX_PLATE_NUM];
```

```
unsigned int plateNum;
```

```
}ADASOutData;
```

【变量】

变量	含义	说明
ptrVersion	版本信息	ADAS 版本信息
leftLaneLineX0	左车道线左上角点 X 坐标	单位：像素
leftLaneLineY0	左车道线左上角点 Y 坐标	单位：像素
leftLaneLineX1	左车道线左下角点 X 坐标	单位：像素
leftLaneLineY1	左车道线左下角点 Y 坐标	单位：像素
rightLaneLineX0	右车道线右上角点 X 坐标	单位：像素
rightLaneLineY0	右车道线右上角点 Y 坐标	单位：像素
rightLaneLineX1	右车道线右下角点 X 坐标	单位：像素
rightLaneLineY1	右车道线右下角点 Y 坐标	单位：像素
leftLaneLineWarn	车道线报警信息	0-黄色（未检测到） 1-绿色（检测到）

		2-红色（压线报警） 3-请重新标定摄像头
rightLaneLineWarn	车道线报警信息	0-黄色（未检测到） 1-绿色（检测到） 2-红色（压线报警） 3-请重新标定摄像头
colorPointsNum	车道线条状块分割点的个数	数组
dnColor	下方一块的颜色	1-蓝，2-绿
rowIndex[MAX_INDEX_NUM]	车道线条状块分割点的行坐标	数组
ltColIndex[MAX_INDEX_NUM]	车道线条状块左边分割点的列坐标	数组
mdColIndex[MAX_INDEX_NUM]	车道线条状块中间分割点的列坐标	数组
rtColIndex[MAX_INDEX_NUM]	车道线条状块右边分割点的列坐标	数组
carX[MAX_CAR_NUM]	车辆起点 X 坐标	数组
carY[MAX_CAR_NUM]	车辆起点 Y 坐标	数组
carW[MAX_CAR_NUM]	车辆宽度	数组
carH[MAX_CAR_NUM]	车辆高度	数组
carWarn[MAX_CAR_NUM]	车辆报警信息	数组， 0-无信号， 1-车道偏离报警 2-前车碰撞预警
carNum	检测到的车辆数目	
plateX	车辆左上角 X 坐标	最大不超过 MAX_PLATE_NUM
plateY	车辆左上角 Y 坐标	最大不超过 MAX_PLATE_NUM
plateW	车辆右下角 X 坐标	最大不超过 MAX_PLATE_NUM
plateH	车辆右下角 Y 坐标	最大不超过 MAX_PLATE_NUM
plateNum	检测车辆数目	最大不超过 MAX_PLATE_NUM

【参考头文件】

ADASAPI.h

【备注】**ADASPara****【说明】**

ADAS 初始化参数

【定义】

//ADAS 初始化参数

typedef struct ADASPara

{

//图像宽度,高度

unsigned int frameWidth;

unsigned int frameHeight;

//=====抠图位置

//int cropEnable;

int roiX;

int roiY;

int roiH;

int roiW;

//video 帧率

unsigned int fps;

//camera parameter 相机参数

unsigned int focalLength;//um

unsigned int pixelSize;//um

unsigned int horizonViewAngle;

unsigned int verticalViewAngle;

/**/ADD**/

unsigned int vanishX;//天际消失点坐标

unsigned int vanishY;

unsigned int hoodLineDist;//车头距离 cm

unsigned int vehicleWidth;//车辆宽度 cm

```
unsigned int wheelDist;//车轮距离 cm
unsigned int cameraToCenterDist;//摄像头中心 cm
```

```
unsigned int cameraHeight;//相机高度 cm
```

```
unsigned int leftSensity;//0-1-2-3
```

```
unsigned int rightSensity;//0-1-2-3
```

```
unsigned int fcwSensity;//0-1-2-3
```

```
//初始化回调函数
```

```
AdasIn awAdasIn;
```

```
AdasOut awAdasOut;
```

```
}ADASPara;
```

【变量】

变量	含义	说明
frameWidth	图像宽度	单位：像素
frameHeight	图像高度	单位：像素
roiX	感兴趣区域 X 坐标	单位：像素
roiY	感兴趣区域 Y 坐标	单位：像素
roiH	感兴趣区域高度	单位：像素
roiW	感兴趣区域宽度	单位：像素
fps	视频帧率	25/30 FPS
focalLength	焦距	相机参数, 单位: um
pixelSize	SENSOR 单个像元大小	相机参数, 单位: um
horizonViewAngle	水平视场角	单位：度
verticalViewAngle	垂直视场角	单位：度
vanishX	天际消失点 X 坐标	单位：像素
vanishY	天际消失点 Y 坐标	单位：像素
hoodLineDist	车头距离	单位：厘米
vehicleWidth	车辆宽度	单位：厘米
wheelDist	车轮距离	单位：厘米
cameraToCenterDist	摄像头中心距离	单位：厘米
cameraHeight	相机安装高度	单位：厘米
leftSensity	左侧车道线检测敏感度	0-1-2-3 四档，默认为 1
rightSensity	右侧车道线检测敏感度	0-1-2-3 四档，默认为 1

fcwSensity	前车碰撞预警敏感度	0-1-2-3 四档，默认为 1
awAdasIn	ADAS 输入回调函数	参考 AdasIn 回调
awAdasOut	ADAS 输出回调函数	参考 AdasOut 回调

【备注】

ADASAPI.h

Allwinmertechn

4.调用例程

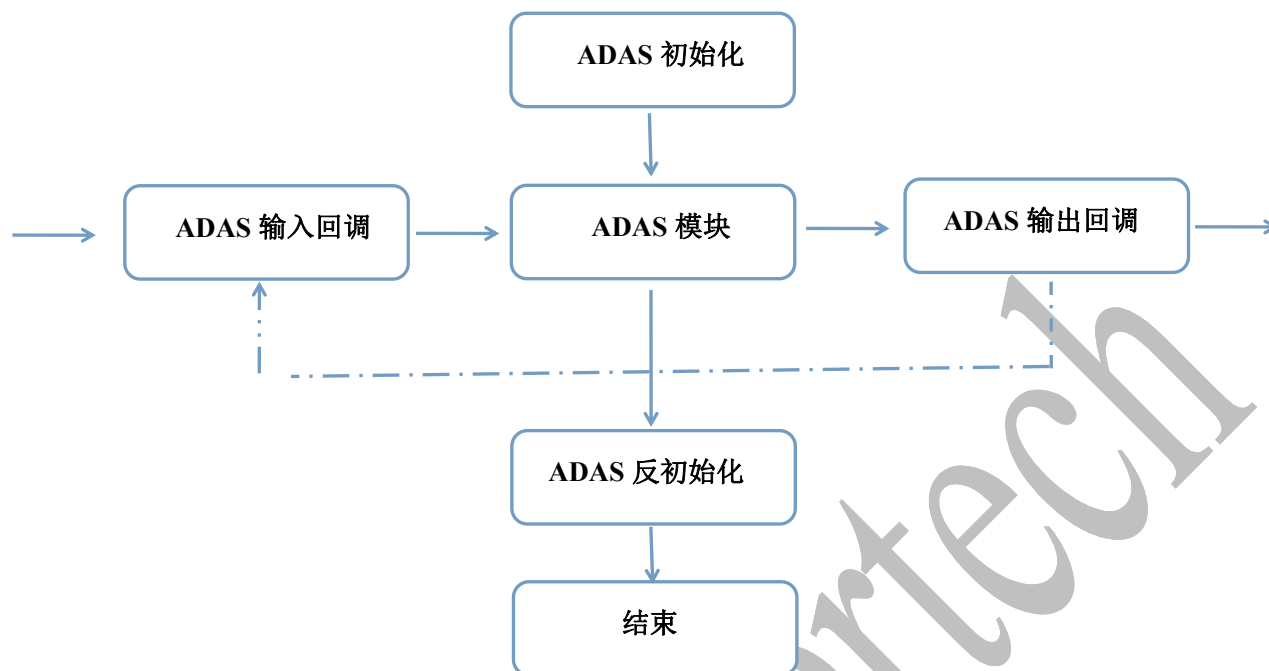


图 4.1 ADAS 调用流程

//以 NV21 为例，ADAS 的主要流程如下：

```
#include "ADASAPI.h"
#include "string.h"
#include "stdlib.h"
#include "stdio.h"
#include <pthread.h>
#include <semaphore.h>
#include <signal.h>
#include <unistd.h>

#define TEST_TIME 1
#ifdef TEST_TIME
#include <sys/time.h>

int timeval_subtract(struct timeval* result, struct timeval* x, struct timeval* y)
{
    int nsec;

    if ( x->tv_sec>y->tv_sec )
        return -1;
```



```
        if ( (x->tv_sec==y->tv_sec) && (x->tv_usec>y->tv_usec) )
            return -1;

        result->tv_sec = ( y->tv_sec-x->tv_sec );
        result->tv_usec = ( y->tv_usec-x->tv_usec );

        if (result->tv_usec<0)
        {
            result->tv_sec--;
            result->tv_usec+=1000000;
        }

        return 0;
    }
#endif

//是否开始视频传入
int bInit = 0;

//ADAS 数据结构体
typedef struct ADASData
{
    ADASInData adasInData; //ADAS 输入数据结构体
    ADASOutData adasOutData; //ADAS 输出数据结构体
    ADASPara adasPara;      //ADAS 配置参数结构体

    int dataADASIn;          //ADAS 输入完成标记, 完成 1, 初始化为 0

    //线程相关
    pthread_cond_t dataInCond;
    pthread_mutex_t dataInLock;
    pthread_cond_t dataTransferCond;
```

```
pthread_mutex_t dataTransferLock;

unsigned char *ptrImgBuffer;
unsigned char *ptrImgGray;//图像灰度数据
int width;    //图像宽度
int height;   //图像高度

int dataBackNum;

//信号量相关
sem_t semIndata;
sem_t semOutdata;
}ADASData;

ADASData adasData;

//ADAS 输入回调
void awAdasIn(ADASInData *ptrADASInData,void *dv)
{
    sigset_t set1;
    sigemptyset(&set1);
    sigaddset(&set1, SIGIO);
    pthread_sigmask(SIG_UNBLOCK, &set1, NULL);
    int ret = sigpending(&set1);

    if(bInit == 0)
    {
        usleep(40000);
        //printf("image buffer is not ready #####\n");
    }
    else
    {
        sem_wait(&adasData.semIndata);
        pthread_mutex_lock(&adasData.dataTransferLock);
```

```
memcpy(adasData.ptrImgBuffer,adasData.ptrImgGray,adasData.width * adasData.height);
//将视频数据刷新给 ADAS 输入 buffer
memcpy(ptrADASInData,&adasData.adasInData,sizeof(ADASInData));
adasData.dataADASIn = 1;
pthread_mutex_unlock(&adasData.dataTransferLock);
}
}

//ADAS 输出回调
void awAdasOut(ADASOutData *ptrADASOutData,void *dv)
{
    sigset_t set1;
    sigemptyset(&set1);
    sigaddset(&set1, SIGIO);
    pthread_sigmask(SIG_UNBLOCK, &set1, NULL);
    int ret = sigpending(&set1);
    if(bInit == 0)
    {
        usleep(40000);
        //printf("image buffer is not ready #####\n");
    }
    else
    {
        pthread_mutex_lock(&adasData.dataTransferLock);

        //将 ADAS 执行结果传送给全局变量
        if(NULL!=ptrADASOutData)
        {
            memcpy(&adasData.adasOutData, ptrADASOutData, sizeof(ADASOutData));
        }

        adasData.dataBackNum = 1;
        pthread_mutex_unlock(&adasData.dataTransferLock);
        sem_post(&adasData.semOutdata);
    }
}
```

```
}  
}  
  
//计算和前车的距离  
float awDistVehicleBottomY(int vanishY,int bottomY,int srcHeight,float cameraHeight)  
{  
    int carWidth = 1.7;//(单位: m)  
    float k,b;  
    float w;  
    float dist;  
    k = -0.2548*(cameraHeight/100)+0.703;//输入单位:cm  
    b = -k*(float)vanishY;  
    w = k*(float)bottomY+b;  
    dist = 567.2069/(720.0*w/(float)srcHeight+3.0285)-0.5;//(单位: m)  
    return dist;  
}  
  
//主函数  
void main(int argc, char *argv[])  
{  
    int currentFrames = 0;  
    int totalFrames = 250;  
#ifdef TEST_TIME  
    struct timeval starttime, endtime, difftime;  
#endif  
    if(argc < 4)  
    {  
        printf("testAdas *.yuv w h #\n");  
        return;  
    }  
    int width  = atoi(argv[2]);  
    int height = atoi(argv[3]);  
    printf("image width = %d, height = %d\n", width, height);  
    FILE *fileNode = fopen(argv[1], "rb");
```

```
if(NULL == fileNode)
{
    printf("test adas yuv file is not exist!\n");
    return;
}

//配置 ADAS 初始化参数
adasData.adasPara.frameWidth = width; //视频宽度
adasData.adasPara.frameHeight = height; //视频高度
adasData.adasPara.roiX = 0; //感兴趣区域 X 坐标
adasData.adasPara.roiY = 0; //感兴趣区域 Y 坐标
adasData.adasPara.roiH = height; //感兴趣区域高度
adasData.adasPara.roiW = width; //感兴趣区域宽度
adasData.adasPara.fps = 30; //视频帧率
adasData.adasPara.focalLength = 2800; //焦距 um
adasData.adasPara.pixelSize = 3; //SENSOR 单个像元大小 um
adasData.adasPara.horizonViewAngle = 120; //水平视场角
adasData.adasPara.verticalViewAngle = 90; //垂直视场角
adasData.adasPara.vanishX = width/2; //天际消失点 X 坐标
adasData.adasPara.vanishY = height/2; //天际消失点 Y 坐标
adasData.adasPara.hoodLineDist = 120; //车头距离
adasData.adasPara.vehicleWidth = 182; //车辆宽度
adasData.adasPara.wheelDist = 160; //车轮距离
adasData.adasPara.cameraToCenterDist = 0; //摄像头中心距离
adasData.adasPara.cameraHeight = 120; //相机安装高度
adasData.adasPara.leftSensity = 1; //左侧车道线检测敏感度
adasData.adasPara.rightSensity = 1; //右侧车道线检测敏感度
adasData.adasPara.fcwSensity = 1; //前车碰撞预警敏感度
adasData.adasPara.awAdasIn = awAdasIn; //ADAS 输入回调函数
adasData.adasPara.awAdasOut = awAdasOut; //ADAS 输出回调函数

int vanishY = height/2 + 10; //消失线 Y 坐标位置

//线程控制变量
pthread_mutex_init(&adasData.dataInLock, NULL);
```

```
pthread_cond_init(&adasData.dataInCond, NULL);
pthread_mutex_init(&adasData.dataTransferLock, NULL);
pthread_cond_init(&adasData.dataTransferCond, NULL);

//视频相关
adasData.ptrImgBuffer = (unsigned char *)malloc(width * height);
adasData.ptrImgGray = (unsigned char *)malloc(width * height);
adasData.width = width;
adasData.height = height;
adasData.dataBackNum = 1;

//ADAS 输入参数
adasData.adasInData.ptrFrameData = adasData.ptrImgBuffer; //图像数据
adasData.adasInData.gpsSpeed = 80; //GPS 数据, KM/S
adasData.adasInData.gpsSpeedEnable = 0; //GPS 速度使能
adasData.adasInData.GsensorStop = 1; //加速度传感器停止标记
adasData.adasInData.GsensorStopEnable = 0; //加速度传感器数据停止使能标记
adasData.adasInData.sensity = 2; //检测敏感度
adasData.adasInData.CPUCostLevel = 1; //CPU 消耗等级
adasData.adasInData.luminanceValue = 1500; //光照强度
adasData.adasInData.lightSignal = 0; //车辆转向灯信号
adasData.adasInData.lightSignalEnable = 0; //信号灯使能信号
adasData.adasInData.cameraHeight = 1200; //摄像机离地高度,单位 mm
adasData.adasInData.ptrFrameDataArray = (unsigned char *)malloc(width * height*3/2); //彩色图像数据指针
adasData.adasInData.dataType = 2; //彩色数据类型, NV21
adasData.adasInData.cameraType = 0; //摄像头类型
adasData.adasInData.imageWidth = width; //图像宽度
adasData.adasInData.imageHeight = height; //图像高度
adasData.adasInData.cameraNum = 1; //摄像机数目
adasData.adasInData.Lng = 'E'; //GPS 经度信息
adasData.adasInData.LngValue = 113.46;
adasData.adasInData.Lat = 'N'; //GPS 纬度
adasData.adasInData.LatValue = 22.27;
```

```
adasData.adasInData.altitude = 20.f; //GPS 海拔值
adasData.adasInData.deviceType = 1; //设备类型
strcpy(adasData.adasInData.sensorID, "OVXXXX"); //传感器类型
adasData.adasInData.networkStatus = 0; //连接类型

//初始化 ADAS
AW_AI_ADAS_Init(&adasData.adasPara, NULL);

usleep(5000);

sem_init(&adasData.semIndata, 0, 0);
sem_init(&adasData.semOutdata, 0, 0);

sigset_t set1;
sigemptyset(&set1);
sigaddset(&set1, SIGIO);
pthread_sigmask(SIG_BLOCK, &set1, NULL);
sigpending(&set1);
unsigned char *pNV21Buf = (unsigned char *)malloc(width*height*3/2);
int frameNum = 0;
//遍历视频
while(1)
{
    //读取 NV21 文件
    int flag = fread(pNV21Buf, sizeof(unsigned char), width*height, fileNode);
    if (flag == 0)
    {
        break;
    }
    fread(pNV21Buf + width*height, sizeof(unsigned char), width*height/2, fileNode);
    if(adasData.dataBackNum == 1)
    {
        pthread_mutex_lock(&adasData.dataTransferLock);
        memcpy(adasData.ptrImgGray, pNV21Buf, width*height);
    }
}
```

```
        pthread_mutex_unlock(&adasData.dataTransferLock);
        adasData.dataBackNum = 0;
    }

    bInit = 1;
#ifdef TEST_TIME
    gettimeofday(&starttime, 0);
#endif
    //发送 ADAS 回调输入信号
    sem_post(&adasData.semIndata);
    //等待 ADAS 结束信号
    sem_wait(&adasData.semOutdata);
#ifdef TEST_TIME
    gettimeofday(&endtime, 0);
    timeval_subtract(&difftime, &starttime, &endtime);
    printf("ADAS process cost time = %f ms\n", difftime.tv_usec/1000.f);
#endif
    //ADAS 输出结果
    pthread_mutex_lock(&adasData.dataTransferLock);

    //输出车辆检测信息
    for (int i = 0; i < adasData.adasOutData.carNum; i++)
    {
        if (adasData.adasOutData.carW[i] != adasData.adasOutData.carH[i])
        {
            goto EXIT;
        }
        float dist = awDistVehicleBottomY(vanishY, adasData.adasOutData.carY[i] +
adasData.adasOutData.carH[i], width, 120);
        printf("%d:ADAS result: x[%d], y[%d], width[%d], height[%d], dist[%f]\n",
frameNum,
        adasData.adasOutData.carX[i],
        adasData.adasOutData.carY[i],
        adasData.adasOutData.carW[i],
```



```
        adasData.adasOutData.carH[i],
        dist);

    //输出车辆报警信息
    //无信号: 0x0
    //车道偏离报警信号: 0x1,
    //保持车距报警信号: 0x2
    printf("ADAS warn flag = %d\n", adasData.adasOutData.carWarn[i]);
}

//左侧车道线
printf("ADAS lane detect: sx[%d], sy[%d], ex[%d], ey[%d],\n",
        adasData.adasOutData.leftLaneLineX0,
        adasData.adasOutData.leftLaneLineY0,
        adasData.adasOutData.leftLaneLineX1,
        adasData.adasOutData.leftLaneLineY1);

//右侧车道线
printf("ADAS lane detect: sx[%d], sy[%d], ex[%d], ey[%d],\n",
        adasData.adasOutData.rightLaneLineX0,
        adasData.adasOutData.rightLaneLineY0,
        adasData.adasOutData.rightLaneLineX1,
        adasData.adasOutData.rightLaneLineY1);

//车道线报警信息
printf("ADAS lane detect: leftLaneLineWarn[%d], rightLaneLineWarn[%d]\n",
        adasData.adasOutData.leftLaneLineWarn, //左车道线报警信息
        adasData.adasOutData.rightLaneLineWarn); //右车道线报警信息

pthread_mutex_unlock(&adasData.dataTransferLock);
frameNum++;
}
EXIT:
printf("=====ADAS EXIT=====\\n");
```

```
if(NULL != fileNode)
{
    fclose(fileNode);
    fileNode = NULL;
}
//ADAS 反初始化
AW_AI_ADAS_UnInit();
//释放线程相关
pthread_mutex_destroy(&adasData.dataInLock);
pthread_cond_destroy(&adasData.dataInCond);
pthread_mutex_destroy(&adasData.dataTransferLock);
pthread_cond_destroy(&adasData.dataTransferCond);
//释放图像缓存
free(adasData.ptrImgGray);
free(adasData.ptrImgBuffer);
}
```