**【RokidGlassSDK】**

**接口文档**

目录

[1、 项目概述](#_Toc29233)

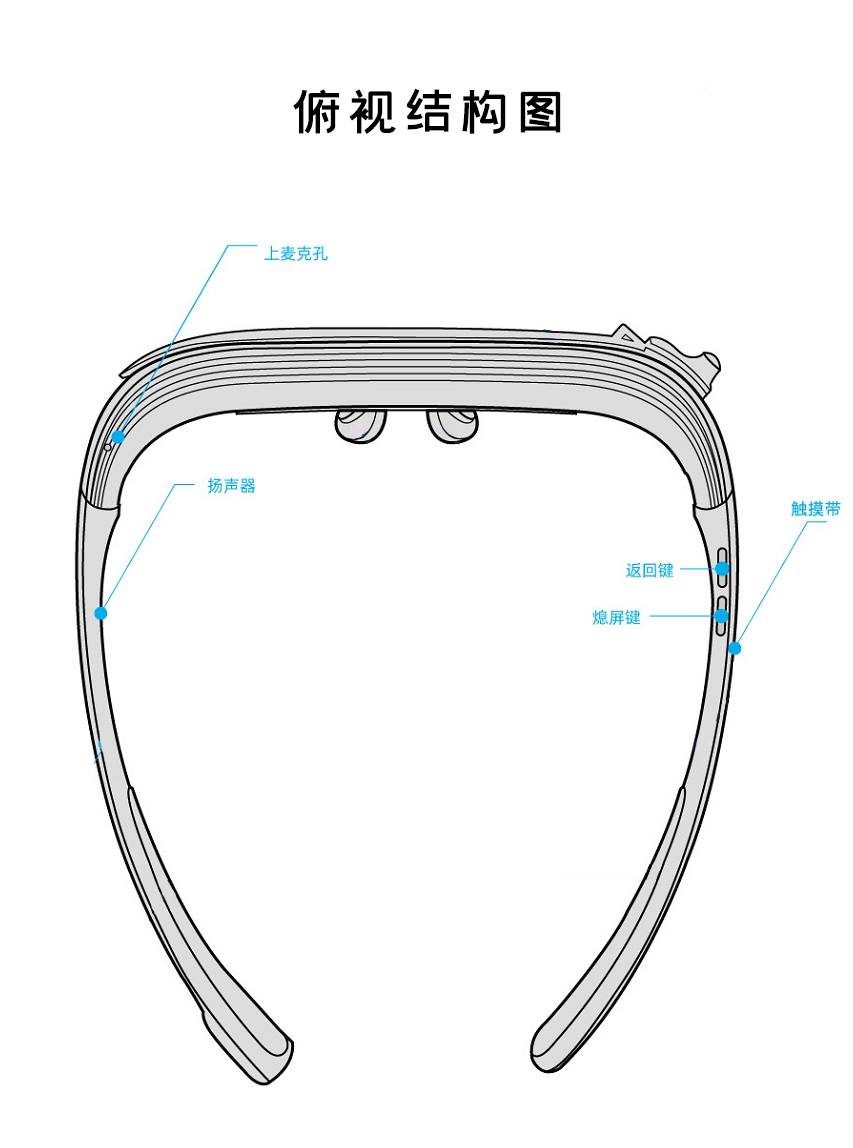
[2、方案框架图](#_Toc30513)

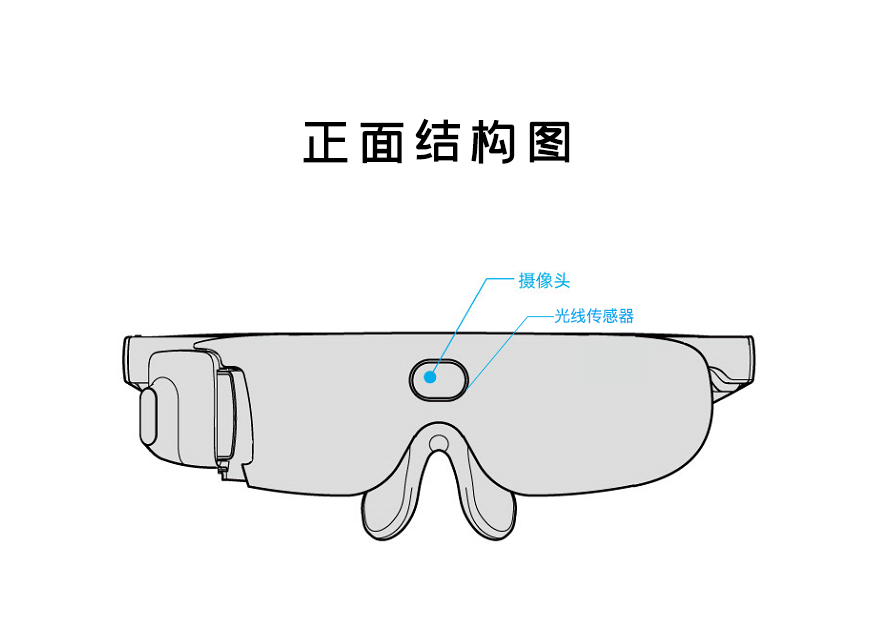
[3、业务接口](#_Toc1757)

## 项目概述

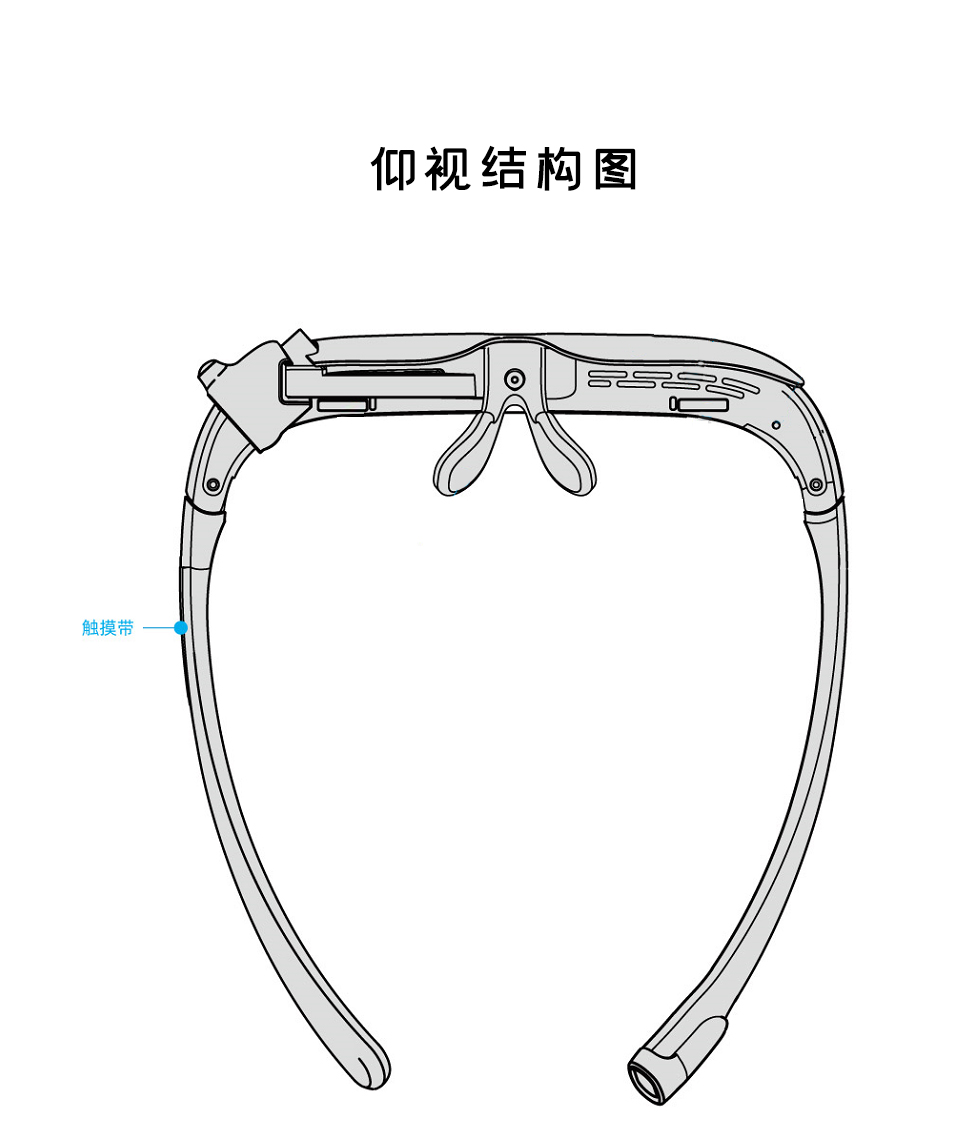
1.1 概述

Rokid Glass SDK适用于Window平台，Rokid Glass作为Dongle，通过TypeC接口连接到Window平台设备。Window平台可以接收到眼镜的IMU、Touch、Key、Light Sensor、Proximity Sensor、Camera、Audio等数据。下面是眼镜的结构图，指示眼镜Dongle中相关设备的位置示意图。









1.2 设备介绍

**麦克风陈列：**眼镜中采用了双Mic设计进行拾音，有利于AI语音算法引擎更好进行语音识别。系统返回的数据默认16k采样率，16bit位深，2声道。

**触摸带**：支持手指滑动，具有短按，长按，向前滑动，向后滑动事件。

**距离感应器**：感应障碍物的距离。主要用于眼镜的佩戴识别。

**光线传感器**：感应外界光线强度，单位为Lux

**Type-C接口**：Type-C是一种既可以应用于PC(主设备)又可以应用于外部设备(从设备，如手机)的接口类型，是最新的USB接口外形标准。

**摄像头**：RGB摄像头。

**按键**：分为灭屏键，和返回键。

**IMU姿态传感器**：姿态传感器是基于MEMS技术的高性能三维运动姿态测量系统。它包含三轴陀螺仪、三轴加速度计，三轴电子罗盘等运动传感器，通过内嵌的低功耗ARM处理器得到经过温度补偿的三维姿态与方位等数据。利用基于四元素的三维算法和特殊数据融合技术，实时输出以四元数、欧拉角表示的零漂移三维姿态方位数据。Rokid有两种姿态数据，GameRotation和Rotation。Rotation是九轴融合数据，返回的值为绝对坐标。GameRotation是六轴融合数据，返回的值为相对坐标。

**加速度传感器**：加速度传感器是一种能够测量加速度的传感器。

**磁力计传感器**：测试磁场强度和方向，用于定位设备方位。  
**陀螺仪传感器**：测量设备自身的旋转运动。

## 2、SDK方案框架图

SDK

USB协议层

USB Type-C

Rokid Glass Device

C++

SDK框架设计图

Rokid Glass SDK设计主要依赖于USB协议和Win32 API接口进行包装，将RokidGlass Device上设备的能力映射到Window平台。Rokid Glass SDK设计采用了回调的设计模式，调用者不需要过多参与到线程，内存等技术环节的代码设计就可以获得设备的数据，便于开发者快速接入Rokid AR Glass，进行行业相关业务逻辑的开发设计。

## 3、业务接口

|  |
| --- |
| **Change List：**  V 0.1.1:  1、支持Touch/Key Event接口  V 0.1.2   1. 增加IMU Raw/Light/Proximity Sensor 接口   V 1.0.0   1. 增加Camera 接口 2. 增加Audio接口 3. 增加版本信息接口 |

|  |
| --- |
| **SENSORID**   * V 1.0.0   指示sensor的类型  **Syntax**  Typedef enum SENSORID {  ACCELEROMETER,  MAGNETIC,  GYROSCOPE,  };  **字段说明**   * + ACCELEROMETER：加速度传感器   + MAGNETIC: 磁力计传感器   + GYROSCOPE: 陀螺仪传感器   **KEY\_CODE**   * V 1.0.0   按键事件的类型  **Syntax**  Typedef enum KEY\_CODE {  BACK\_KEY = 4,  POWER\_KEY = 16,  };  **字段说明**   * + BACK\_KEY：返回键   + POWER\_KEY: 熄屏键   **TOUCH\_EVENT**   * V 1.0.0   触摸事件的类型  **Syntax**  Typedef enum TOUCH\_EVENT{  SHORT\_PRESS,  LONG\_PRESS,  FORWARD\_SLIDE,  BACKWARD\_SLIDE  };  **字段说明**   * + SHORT\_PRESS：触摸带短按   + LONG\_PRESS：触摸带长按   + FORWARD\_SLIDE：向前滑动   + BACKWARD\_SLIDE：向后滑动   **VIDEO\_TYPE**   * V 1.0.0   Camera返回数据的图片格式  **Syntax**  Typedef enum VIDEO\_TYPE {  YUV,  RGB,  JPEG  };  **字段说明**   * + YUV：YUV格式的图片   + RGB：RGB格式的图片   + JPEG：JPEG格式的图片   **VideoFormat**   * V 1.0.0   Camera返回数据基本信息的结构体  **Syntax**  Typedef struct \_VideoFormat{  int type;  long height;  long width;  }VideoFormat;  **字段说明**   * + type：VIDEO\_TYPE表示的图片格式   + height：Camera返回图片的高度   + width：Camera返回图片的宽度   **AudioFormat**   * V 1.0.0   Audio返回数据基本信息的结构体  **Syntax**  Typedef struct \_AudioFormat{  int channels;  long sample;  long bitsPerSample;  }AudioFormat;  **字段说明**   * + channels：Audio的通道个数   + sample：Audio的采样率   + bitsPerSample：Audio的采用位深   **GlassInitial**   * V 0.1.0   SDK初始化函数  **Syntax**  GlassHandle GlassInitial(  GlassEvent \*callback  );  **参数列表**   * callback 用户实现的回调函数接口，用于处理glass产生的事件   **返回值**  成功则返回glass 示例指针，失败返回NULL。 |
|  |

|  |
| --- |
| **GlassOpen**   * V 0.1.0   打开glass外设  负责打开glass设备具柄，所有眼镜操作和回调后要在glass打开后才能正常工作。  **Syntax**  bool GlassOpen(  GlassHandle\*instance  );  **参数列表**   * instance 用户创建的glass sdk实例指针   **返回值**  成功则返回True，，失败返回False。 |

|  |
| --- |
| **GlassStartCapture**   * V 1.0.0   启动Rokid Glass设备中的Camera Preview  初始化并且启动Camera，Camera的preview数据通过回调函数onCaptureEvent将图像数据返回给调用者。  **Syntax**  void GlassStartCapture(  GlassHandle\*instance  );  **参数列表**   * instance 用户创建的glass sdk实例指针   **GlassStopCapture**   * V 1.0.0   停止Rokid Glass设备中的Camera Preview  **Syntax**  void GlassStopCapture(  GlassHandle\*instance  );  **参数列表**   * instance 用户创建的glass sdk实例指针   **GlassStartAudio**   * V 1.0.0   启动Rokid Glass设备中的Mic array  初始化并且启动麦克风阵列，眼镜中有两个麦克风，格式分别为16bit/16000sample/2channel。  **Syntax**  void GlassStartAudio(  GlassHandle\*instance  );  **参数列表**   * instance 用户创建的glass sdk实例指针   **GlassGetAudioFrameSize**   * V 1.0.0   获取AR Glass中Mic array一帧中的字节长度  **Syntax**  int GlassGetAudioFrameSize(  GlassHandle\*instance  );  **参数列表**   * instance 用户创建的glass sdk实例指针   **返回值**  Audio PCM一帧的字节长度。  **GlassGetAudioFormat**   * V 1.0.0   获取AR Glass中audio设备的基本信息，主要为通道个数，采样率和采样位深，对应的结构体是AudioFormat，字段信息可以查看AudioFormat的介绍。  **Syntax**  void GlassGetAudioFormat(  GlassHandle\*instance,  AudioFormat\* audioFormat  );  **参数列表**   * instance 用户创建的glass sdk实例指针 * audioFormat 需要获取的audio格式信息   **GlassWaitEvent**   * V 0.1.0   等待眼镜产生事件，直到发生错误或者眼镜离线  **Syntax**  Void GlassWaitEvent(  GlassHandle\*instance  );  **参数列表**   * instance 用户创建的glass sdk实例指针   **返回值**  该函数会进入忙等待状态，调用该函数将阻塞当前线材直到异常返回。 |

|  |
| --- |
| **GlassClose**   * V 0.1.0   SDK关闭函数  负责回收sdk分配的资源，执行后回释放相关内存。  **Syntax**  bool GlassClose(  GlassHandle\*instance  );  **参数列表**   * instance 用户创建的glass sdk实例指针   **返回值**  始终返回成功 |

|  |
| --- |
| **GlassEvent**   * V 1.0.0   SDK 回调函数结构体  当glass正确初始化之后，会通过回调事件的方式返回glass当前的状态  **Syntax**  typedef struct {  void (\*onRotationVectorEvent)(unsigned long timeStamp,float \*quaternion);  void (\*onGameRotationVectorEvent)(unsigned long timeStamp, float\* quaternion);  void (\*onRawSensorEvent)(long timeStamp,int sensorID, float\* data, int status);  void (\*onKeyEvent)(int keyCode, bool status);  void (\*onTouchEvent)(int event, int len);  void (\*onProximitySensorEvnet)(bool state);  void (\*onLightSensorEvent)(int value);  void (\*onCaptureEvent)(double timestamp, VideoFormat\* format, BYTE\* pBuf, long size);  void (\*onAudioEvent)(BYTE\* pBuf, long frames);  }GlassEvent;  **说明**   * onRotationVectorEvent Rotation Vector 回调事件，返回眼镜当前姿态   参数列表   * + - timestamp 当前时间的时间戳 单位是us     - quaternion 姿态4元数 * onGameRotationVectorEvent   Game Rotation Vector 回调事件，返回眼镜当前相对姿态  参数列表   * + - timestamp 当前时间的时间戳 单位是us     - quaternion 姿态4元数 * onRawSensorEvent   RawSensor回调事件，返回眼镜当前传感器的原始数据  参数列表   * + - timestamp 当前时间的时间戳 单位是us     - sensorID 传感器编号 0:加速度 1:磁力计 2:陀螺仪     - data 传感器X、Y、Z轴的数据     - status 传感器校准 0:未校准 1:可信度低 2:已校准 * onKeyEvent   物理按键回调事件，返回眼镜当前按键的状态  参数列表   * + - KeyCode 按键编号 0x4:返回键 0x10:休眠键     - status 按键状态 0x0:释放 0x1:按下 * onTouchEvent   touch bar回调事件，返回眼镜当前touch bar状态  参数列表   * + - Event 事件编号 2:短按 3:长按 4:前滑 5:后滑     - len 表示滑动长度 0-64 * onProximitySensorEvnet   接近传感器回调事件，返回眼镜佩戴状态  参数列表   * + - state 眼镜佩戴状态 true：佩戴 false：未佩戴 * onLightSensorEvent   光线传感器回调事件，返回当前环境光线强度  参数列表  value 环境光强度，单位lux   * onCaptureEvent   Camera previe回调事件，返回当前preview的图片信息  参数列表   * + - timestamp 当前时间的时间戳 单位是us     - format 图片格式，宽和高     - pBuf 图片的二进制数据，回调完成后SDK会自动释放内存     - size 图片二进制数据数组的长度 * onAudioEvent   Audio返回的数据流回调函数  参数列表   * + - pBuf audio返回的二进制pcm流     - frames audio返回多少帧; Note:audio的数组长度需要frames \* frameSize. |