

DCS开发手册

DCS开发手册

DCS开发手册

DCS简介

DCS模块设计说明

使用方法

lightduer_dcs.h

lightduer_dcs_alert.h

接口说明

duer_dcs_framework_init

1. 语音输入(voice input)接口

1.1 duer_dcs_voice_input_init

1.2 duer_dcs_on_listen_started

1.3 duer_dcs_stop_listen_handler

2. 语音输出(voice output)接口

2.1 duer_dcs_voice_output_init

2.2 duer_dcs_speech_on_finished

2.3 duer_dcs_speak_handler

3. 扬声器控制(speaker controller)接口

3.1 duer_dcs_speaker_control_init

3.2 duer_dcs_on_volume_changed

3.3 duer_dcs_on_mute

3.4 duer_dcs_get_speaker_state

3.5 duer_dcs_volume_set_handler

3.6 duer_dcs_volume_adjust_handler

3.7 duer_dcs_mute_handler

4. 音频播放器(audio player)接口

4.1 duer_dcs_audio_player_init

4.2 duer_dcs_audio_on_finished

4.3 duer_dcs_audio_play_handler

4.4 duer_dcs_audio_stop_handler

4.5 duer_dcs_audio_seek_handler

4.6 duer_dcs_audio_pause_handler

5. 播放控制 (Playback Controller)接口

5.1 duer_dcs_play_control_cmd_t

5.2 duer_dcs_send_play_control_cmd

6. 闹钟 (Alerts) 接口

6.1 duer_dcs_alert_init

6.2 duer_alert_event_type

6.3 duer_dcs_report_alert_event

6.4 duer_dcs_alert_set_handler

6.5 duer_dcs_alert_delete_handler

6.6 duer_dcs_report_alert

6.7 duer_dcs_get_all_alert

7. 系统 (System) 接口

duer_dcs_sync_state

DCS简介

- DuerOS Conversational Service(以下简称DCS)是DuerOS对外免费开放的智能语音交互服务API。
- DCS是面向开发者（硬件厂商）的一个标准化接入协议，允许开发者使用麦克风、扬声器等客户端能力，通过语音来连接由DuerOS提供的服务(如音乐播放，计时器和闹钟，天气查询，知识解答等)。
- DCS主要包含三部分：

指令(directive)：服务端下发给设备端，设备端需要执行相应的操作，比如播放语音（Speak指令）。

事件(event)：需要设备端上报给服务端，通知服务端在设备端发生的事情，比如音乐播放开始了（PlaybackStarted事件）等。

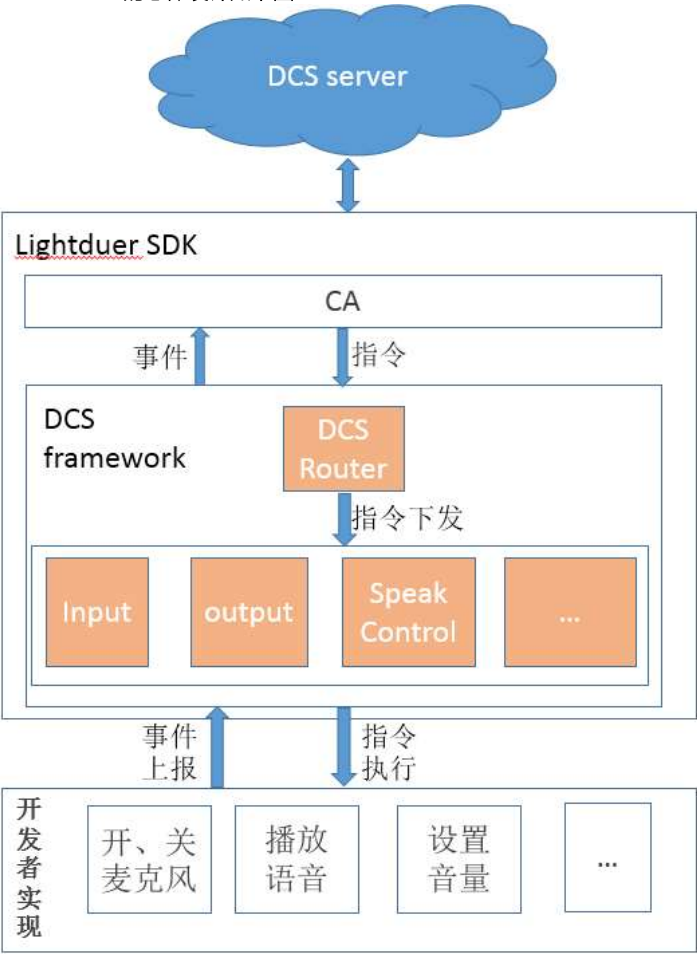
端状态(clientContext)：当前设备上的所有状态，比如当前音量、是不是正在播放音乐、是否有闹钟等。

DCS模块设计说明

为了简化开发者的工作，lightduer SDK中提供了DCS framework，主要提供以下功能：

- 1. 接收并解析server端下发的指令，调用接口（由开发者实现）完成相应操作，例如播放音乐、设置闹钟等。
- 2. 为开发者提供接口，实现事件和端状态的上报

DCS framework的总体设计如下图：



使用方法

根据DCS协议，DCS framework主要分为语音输入、语音输出、播放控制等7个接口（具体参照接口说明部分）。开发者可以根据需要，选择自己需要使用的接口。DCS 要求的功能，DCS framework中有的已经实现，有的提供了函数给开发者调用（例如事件上报），还有一些需要开发者自己实现（例如音频播放）。DCS framework提供了两个头文件：

lightduer_dcs.h

该头文件中声明了dcs framework提供给开发者或需要开发者实现的所有函数（除了闹钟接口相关函数）。

lightduer_dcs_alert.h

由于闹钟功能比较独立，只有部分开发者可能会用到，所以把闹钟接口相关的函数单独放到了该文件中。我们提供了demo来展示如何使用或实现相关的函数，可供开发者参考。

接口说明

要使用DCS framework，开发者首先需要调用duer_dcs_framework_init函数进行初始化。

duer_dcs_framework_init

该函数用于初始化DCS framework。

参数

none

返回值

none

1. 语音输入(voice input)接口

该接口定义语音输入相关的功能，如果设备有麦克风，应该实现该接口，接入语音输入的能力。该接口包括设备端进行语音请求（上传语音），服务端下发开始监听、停止监听指令等。

1.1 duer_dcs_voice_input_init

该函数用于初始化voice input接口，如果开发者需要使用语音输入相关的功能，则必须先调用该函数。

参数

none

返回值

none

1.2 duer_dcs_on_listen_started

该函数用于发送ListenStarted事件给DCS服务端。当用户开始发起语音请求时（在麦克风打开后，上传语音之前），应该调用该函数通知DCS。

参数

none

返回值

int	说明
DUER_OK	执行成功
DUER_ERR_FAILED	执行失败

1.3 duer_dcs_stop_listen_handler

该函数需要开发者实现，当DCS framework收到StopListen指令后，会调用该函数用于关闭麦克风。

注意：该函数需要立即返回，不能被阻塞

参数

none

返回值

none

2. 语音输出(voice output)接口

该接口定义语音输出相关的功能，如果设备有扬声器，则应该实现该接口，接入语音输出的能力。

2.1 duer_dcs_voice_output_init

该函数用于初始化voice output接口，如果开发者需要使用语音输出相关的功能，则必须先调用该函数。

参数

none

返回值

none

2.2 duer_dcs_speech_on_finished

该函数用于向DCS服务端发送SpeechFinished事件。当设备播放完server端发送的语音后，应该调用该函数通知DCS。

参数

none

返回值

none

2.3 duer_dcs_speak_handler

开发者需要实现该函数，当DCS framework收到server端下发的Speak指令后，会调用该函数播放语音。

注意：该函数需要立即返回，不能被阻塞

参数

char *	说明
url	需要播放的语音资源的url

返回值

none

3. 扬声器控制(speaker controller)接口

该接口定义扬声器控制相关的功能，如音量的调节、静音设置等等。

3.1 duer_dcs_speaker_control_init

该函数用于初始化speaker controller接口，如果开发者需要使用扬声器控制相关的功能，则必须先调用该函数。

参数

none

返回值

none

3.2 duer_dcs_on_volume_changed

当设备上的音量发生变化时，调用该函数进行事件上报。

参数

none

返回值

int	说明
DUER_OK	事件上报成功
DUER_ERR_FAILED	事件上报失败

3.3 duer_dcs_on_mute

当设备发生静音/取消静音等操作时，调用该函数进行事件上报。

参数

none

返回值

int	说明
DUER_OK	事件上报成功
DUER_ERR_FAILED	事件上报失败

3.4 duer_dcs_get_speaker_state

开发者需要实现该函数，用于DCS framework获取当前播放控制器的状态，包括音量、是否静音等。

注意：该函数需要立即返回，不能被阻塞

参数

名称	类型	说明
volume	int	用来存储当前音量值
is_mute	bool *	用来存储当前静音状态

返回值

none

3.5 duer_dcs_volume_set_handler

开发者需要实现该函数，当DCS server端下发设置音量的指令后，DCS framework会调用该函数进行音量设置。

注意：该函数需要立即返回，不能被阻塞

参数

名称	类型	说明
volume	int	要设置的音量绝对值，[0~100]

返回值

none

3.6 duer_dcs_volume_adjust_handler

开发者需要实现该函数，当DCS server端下发调整音量的指令后，DCS framework会调用该函数调整音量。

注意：该函数需要立即返回，不能被阻塞

参数

名称	类型	说明
volume	int	音量调整相对值，[-100, 100]

返回值

none

3.7 duer_dcs_mute_handler

开发者需要实现该函数，当DCS server端下发设置或取消静音的指令后，DCS framework会调用该函数设置或取消静音。

注意：该函数需要立即返回，不能被阻塞

参数

名称	类型	说明
is_mute	bool	true代表设置静音，false代表取消静音

返回值

none

4. 音频播放器(audio player)接口

该接口提供了播放音频相关的函数，通过这些函数可以控制设备端上音频资源的播放。

4.1 duer_dcs_audio_player_init

该函数用于初始化audio player接口，如果开发者需要使用音频播放器，则必须先调用该函数。

参数

none

返回值

none

4.2 duer_dcs_audio_on_finished

当一个audio播放完毕后，开发者调用该函数通知DCS framework。

参数

none

返回值

none

4.3 duer_dcs_audio_play_handler

由开发者调用其media播放器来实现该函数，当DCS server下发audio play指令时，DCS framework会调用该函数来播放audio。

注意：该函数需要立即返回，不能被阻塞

参数

名称	类型	说明
url	const char *	需要播放的audio的url

返回值

none

4.4 duer_dcs_audio_stop_handler

由开发者来实现该函数，当DCS server下发stop指令时，DCS framework会调用该函数来停止audio的播放。

注意：该函数需要立即返回，不能被阻塞

参数

none

返回值

none

4.5 duer_dcs_audio_seek_handler

由开发者来实现该函数，用来实现从特定的位置开始播放audio。例如一个audio在播放的过程中，用户进行了语音交互，则audio需要暂停播放（通过调用duer_dcs_audio_pause_handler），等语音交互结束后，DCS framework会调用duer_dcs_audio_seek_handler进行续播。

注意：该函数需要立即返回，不能被阻塞

参数

名称	类型	说明
url	const char *	需要播放的audio的url
offset	int	从该位置开始播放

返回值

none

4.6 duer_dcs_audio_pause_handler

由开发者来实现该函数，用来暂停当前播放的audio。

注意：该函数需要立即返回，不能被阻塞

参数

none

返回值

int型：当前audio的播放位置，调用duer_dcs_audio_seek_handler进行续播时作为参数使用。

5. 播放控制 (Playback Controller)接口

该接口用来支持用户通过控制按钮等方式来控制音乐播放（上一首、下一首、播放、暂停等）。

5.1 duer_dcs_play_control_cmd_t

该枚举类型定义了播放控制事件。

```
1. typedef enum {
2.     DCS_PLAY_CMD, // 当用户按了设备端上的播放按钮时，上报该事件
3.     DCS_PAUSE_CMD, // 当用户按了设备端上的暂停按钮时，上报该事件
4.     DCS_PREVIOUS_CMD, // 当用户按了设备端上的上一首按钮时，上报该事件
5.     DCS_NEXT_CMD, // 当用户按了设备端上的下一首按钮时，上报该事件
6.     DCS_PLAY_CONTROL_EVENT_NUMBER,
7. } duer_dcs_play_control_cmd_t;
```

5.2 duer_dcs_send_play_control_cmd

该函数用来上报播放控制事件。例如，如果用户按了下一首的按键，则应上报DCS_NEXT_CMD事件，DCS server收到该事件后，会下发一个新的audio给设备进行播放。

参数

名称	类型	说明
cmd	duer_dcs_play_control_cmd_t	需要上报的事件类型

返回值

int	说明
DUER_OK	事件上报成功
DUER_ERR_FAILED	事件上报失败
DUER_ERR_MEMORY_OVERFLOW	内存不足，事件上报失败

6. 闹钟 (Alerts) 接口

该接口提供设置、取消闹钟的功能。闹钟功能的使用流程如下：

1. 用户通过“今天晚上8点提醒我喝水”之类的话术，由设备传给云端
2. 云端解析后下发设置闹钟的指令，其中包括token、url、time等内容，开发者需要把这些信息保存到本地，并设定本地闹钟
3. 闹钟触发后，设备播放2中url对应的音频，并删除本地存储的该闹钟的一些信息

6.1 duer_dcs_alert_init

该函数用于初始化alert接口，如果开发者需要使用闹钟功能，则必须先调用该函数。

6.2 duer_alert_event_type

该枚举类型定义了闹钟事件类型。

```
1. typedef enum {
2.     SET_ALERT_SUCCESS, // 设置闹钟成功时上报该事件
3.     SET_ALERT_FAIL, // 设置闹钟失败时上报该事件
4.     DELETE_ALERT_SUCCESS, // 删除闹钟成功时上报该事件
5.     DELETE_ALERT_FAIL, // 删除闹钟失败时上报该事件
6.     ALERT_START, // 闹钟触发时上报该事件
7.     ALERT_STOP, // 闹钟停止时上报该事件
8. } duer_alert_event_type;
```

6.3 duer_dcs_report_alert_event

该函数用于上报闹钟事件。当闹钟设置（或删除）成功（或失败）、闹钟触发或被停止时，都应该上报相应的事件给DCS server，然后DCS server会下发相应的语音给设备。例如上报SET_ALERT_SUCCESS事件后，设备会收到类似“成功为您设置了闹钟”之类的语音播报。

参数

名称	类型	说明
token	const char *	该闹钟的token
type	duer_alert_event_type	需要上报的事件类型

6.4 duer_dcs_alert_set_handler

开发者需要实现该函数，用于设置闹钟。

注意：该函数需要立即返回，不能被阻塞

参数

名称	类型	说明
token	const char *	该闹钟的token，是一个闹钟的 唯一 标识
time	const char *	闹钟的触发时间，ISO 8601格式，根据这个time，通过NTP或RTC校准时间，然后设置定时器
type	const char *	闹钟的类型，TIMER或ALARM

返回值

none

6.5 duer_dcs_alert_delete_handler

开发者需要实现该函数，用于删除闹钟。

注意：该函数需要立即返回，不能被阻塞

参数

名称	类型	说明
token	const char *	要删除的闹钟的token

返回值

none

6.6 duer_dcs_report_alert

开发者在duer_dcs_get_all_alert函数中使用该函数，将现在设备上所有的闹钟信息逐个上报给DCS framework.

参数

名称	类型	说明
alert_array	baidu_json *	用于存储所有的闹钟相关信息，该参数由duer_dcs_get_all_alert函数传入
token	const char *	一个闹钟的token
type	const char *	闹钟的类型，TIMER或ALARM
time	const char *	闹钟的触发时间，ISO 8601格式

返回值

none

6.7 duer_dcs_get_all_alert

该函数需要由开发者实现，DCS framework通过调用该函数获取当前设备上所有的闹钟信息。开发者可以在该函数中循环调用duer_dcs_report_alert函数，来实现这些信息的上报。

参数

名称	类型	说明
alert_array	baidu_json *	用于存储所有的闹钟相关信息

返回值

none

7. 系统 (System) 接口

该接口包含了一些系统级别的接口，DCS framework已经做了相应接口的实现，开发者只需要在设备启动时调用duer_dcs_sync_state函数，把设备上当前的状态上报给DCS server端即可。

duer_dcs_sync_state

该函数用于当设备启动时把设备的状态发送给DCS server。

参数

none

返回值

none