

RT-Thread ASR 应用开发指南

文件标识: RK-KF-YF-364

发布版本: V1.2.0

日期: 2020-08-11

文件密级: ☐绝密 ☐秘密 ☐内部资料 ☒公开

免责声明

本文档按“现状”提供, 瑞芯微电子股份有限公司(“本公司”, 下同)不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因, 本文档将可能在未经任何通知的情况下, 不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标, 归本公司所有。

本文档可能提及的其他所有注册商标或商标, 由其各自所有者所有。

版权所有 © 2020 瑞芯微电子股份有限公司

超越合理使用范畴, 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。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址: 福建省福州市铜盘路软件园A区18号

网址: www.rock-chips.com

客户服务电话: +86-4007-700-590

客户服务传真: +86-591-83951833

客户服务邮箱: fae@rock-chips.com

前言

概述

本文档主要介绍RT-Thread ASR的应用开发实例。

产品版本

芯片名称	RT Thread 版本
RK2108	RT-Thread v3.1.3 及以上

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

修订记录

版本	作者	日期	修改说明
V1.0.0	马龙昌	2019-09-06	初始发布
V1.0.1	马龙昌	2020-03-30	更新第2章节
V1.1.0	马龙昌	2020-05-27	更新文档格式，更新第2章节
V1.2.0	吴佳健	2020-08-11	更新文档格式，更新配置说明

目录

RT-Thread ASR 应用开发指南

1. 概述
2. ASR应用实例
 - 2.1 代码路径
 - 2.2 配置说明
 - 2.3 思必驰固件配置说明
 - 2.4 声卡接口说明
 - 2.4.1 接口说明
 - 2.4.2 注意事项
3. Display应用实例
 - 3.1 代码路径
 - 3.2 配置说明
 - 3.3 应用开机启动
4. 文件系统打包
5. 固件编译与生成
6. 固件烧录
7. 运行测试
8. 参考文档

1. 概述

本文介绍了Rockchip 应用实例基于 RT-Thread 实现的ASR语音唤醒功能(思必驰语音固件)、并结合ASR语音唤醒功能实现带屏显示语音唤醒交互结果的功能。

通过本文介绍的应用实例，开发者可了解并掌握关于Rockchip RT-Thread开发平台的基于语音唤醒功能、带显示的应用使用功能，以便参考。

2. ASR应用实例

2.1 代码路径

```
1 | ./applications/  
2 | └─ rk_iot_app/asr
```

2.2 配置说明

bsp/rockchip/rk2108目录下，执行scons --menuconfig，参考配置如下：

- 开启DSP配置（选中其中标注*的项，下同）

```
1 | > RT-Thread rockchip rk2108 drivers > Enable DSP >  
2 | [*] Enable DSP  
3 | [*]   Enable firmware loader to dsp   #使能将固件加载到dsp  
4 |       Dsp firmware path (Store firmware data in builtin) --->  
5 | (rkdsp_fw.c) Dsp firmware file name   #此处填入dsp_fw目录下固件文件名  
6 | [ ]   Enable dsp send trace to cm4  
7 | (0)   Config dsp debug uart port
```

这里dsp debug uart的port配置根据实际使用中调试串口端口保持一致。

若这里使用uart0作为M4调试端口，DSP同样也需配置成端口0，否则无法接收DSP端的调试信息。

- 开启codec

```

1  > RT-Thread rockchip rk2108 drivers > Enable Audio > Audio Card >
2  [*] Audio Cards
3  [ ]   Enable Internal ADC audio card
4  [ ]   Enable AUDIOPWM audio card
5  [ ]   Enable I2S Digital Mic audio card
6  [*]   Enable PDM Digital Mic audio card      #选择PDM MIC
7  [ ]   Enable Everest Semi ES7243 audio card
8  [*]   Enable Everest Semi ES8311 audio card #选择es8311 声卡
9  [ ]   Enable Everest Semi ES8388 audio card
10 [ ]   Enable mix audio card with Internal ADC + ES8311
11 [ ]   Enable mix audio card with PDM Mics + ES8388

```

根据硬件具体情况选择使能声卡。

- 开启APP

```

1  > RT-Thread application
2      Display demo select (Applications disable)  --->
3  [*] rk iot app
4  [*]   system info save to flash
5  [*]   boot app automatically
6  [ ]   network and wlan enable
7  [ ]   Enable aispeech
8      Select asr wake up mode (use xiaodu wake up words)  ---> #选择唤醒词
9  (es8311p) Playback sound card      #播放声卡设置
10 (pdmc) ASR sound card               #录音声卡设置
11 [ ]   dsp get data through vad path
12 [ ] Enable dual-tracking

```

- 开启Audio Server

```

1  > RT-Thread third party package
2  [*] Audio server  --->
3      --- Audio server
4      Compile type (Use static library)  ---> #选择使用静态库

```

2.3 思必驰固件配置说明

思必驰固件使用dsp_fw/rkdsp_fw_speech.c，需在[配置说明](#)基础上修改如下配置：

固件选择

```

1  > RT-Thread rockchip rk2108 drivers > Enable DSP >
2  ...
3  (rkdsp_fw_speech.c) Dsp firmware file name  #此处选择思必驰dsp固件文件
4  ...
5
6  > RT-Thread application
7  ...
8      Select asr wake up mode (use speech wake up words)  ---> #使用思必驰唤醒
    词
9  ...

```

替换setting.ini

使用bsp/rockchip/rk2108/board/common/setting_speech.ini替换
bsp/rockchip/rk2108/board/common/setting.ini，并保持setting.ini名称不变。

修改内存地址

由于思必驰固件较大，需要使用部分SRAM，因此将CPU可用内存SRAM缩小。

```
1 > RT-Thread board config --->
2 (audio_demo_rk2108_v11) the board name of you use #同board目录下定义的板型名称
3 (0x200E0000) the address of sram end #调整SRAM的结束地址到0x200E0000,默认为0x20100000
4 [ ] Enable Cortex M4 JTAG
5 [ ] Enable DSP JTAG
```

2.4 声卡接口说明

声卡接口具体实现在bsp/rockchip/common/drivers/audio/rk_audio.c，使用示例可参考
bsp/rockchip/common/tests/tinyplay.c和bsp/rockchip/common/tests/tinycap.c。

2.4.1 接口说明

rk_audio.c文件内接口不能直接调用，而是通过RT-Thread Audio设备框架进行调用，相关接口有：

接口名称	用途
rt_device_t rt_device_find(const char* name);	根据 Audio 设备名称获取设备句柄
rt_err_t rt_device_open(rt_device_t dev, rt_uint16_t oflags);	通过设备句柄打开或关闭设备
rt_err_t rt_device_control(rt_device_t dev, rt_uint8_t cmd, void* arg);	通过设备句柄操作设备
rt_size_t rt_device_write(rt_device_t dev, rt_off_t pos, const void* buffer, rt_size_t size);	写入音频数据
rt_size_t rt_device_read(rt_device_t dev, rt_off_t pos, void* buffer, rt_size_t size);	获取音频数据
rt_err_t rt_device_close(rt_device_t dev);	关闭音频设备

接口详细说明请查看RT-Thread官方文档 [Audio设备篇](#)。

2.4.2 注意事项

音频读写所需环形内存由调用者维护，即需要调用者申请内存，并在使用结束后释放，申请或释放需要使用rt_malloc_uncache和rt_free_uncache接口，使用其他接口申请的内存可能会因为读写速度的影响导致声音卡顿。

读写音频数据时传入的 `size` 为帧大小，与bytes的关系为 `frames = bytes / (bits >> 3 * channels)`，例如16bits、2ch的情况下，帧字节数为2048，则读写时传递的size应为 `2048 / (16 >> 3 * 2) = 512`。

3. Display应用实例

3.1 代码路径

```
1 | ./applications/  
2 | └─ rk_iot_app  
3 | └─ rk_iot_display
```

3.2 配置说明

显示实例相关配置如下：

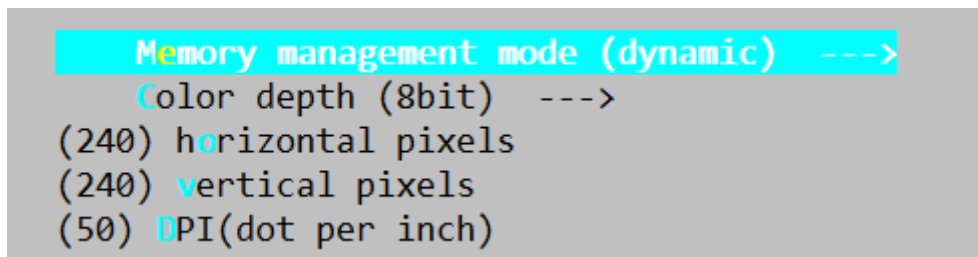
进入工程目录，执行**menuconfig**命令：

```
1 | usr@host:~/path to sdk/$ cd bsp/rockchip/rk2108/  
2 | usr@host:~/path to sdk/bsp/rockchip/rk2108$ scons --menuconfig
```

- LittlevGL组件的配置：

```
1 | Location:  
2 |   -> RT-Thread Components  
3 |     -> System  
4 |       -> LittlevGL2RTT: The LittlevGl gui lib adapter RT-Thread  
      (RT_USING_LITTLEVGL2RTT [=y])  
5 |         -> LittlevGL2RTT Component Options
```

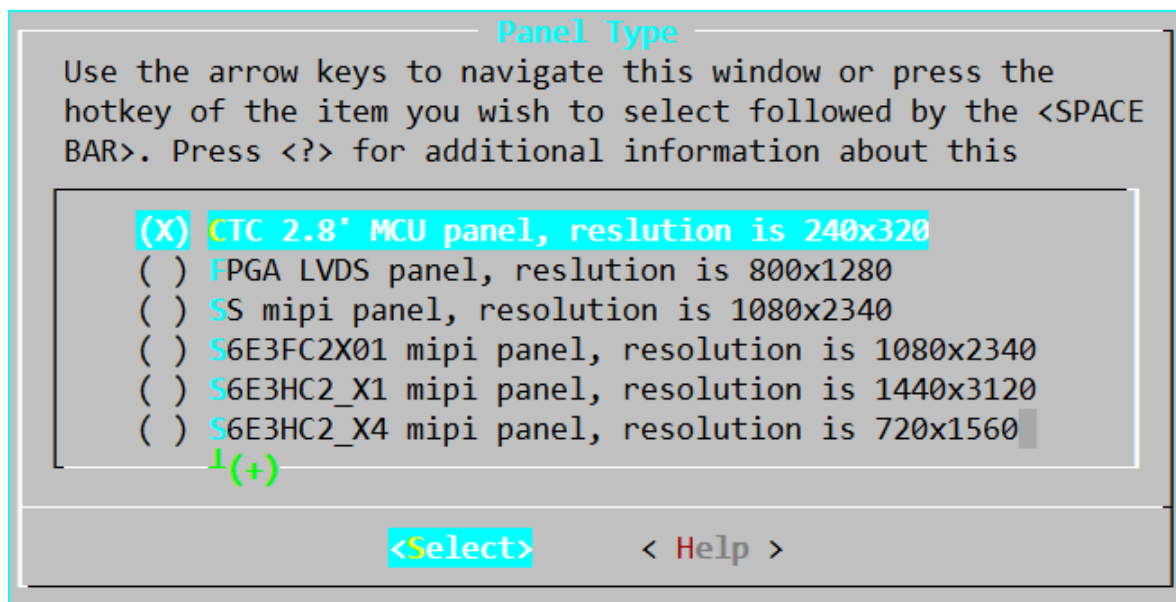
具体配置如上图所示：使用MCU屏，配置成宽高240*240像素，8bit色深。



按照以下配置，选择240x320 MCU屏幕：

```
1 | Location:  
2 |   -> RT-Thread rockchip common drivers  
3 |     Panel Type (CTC 2.8' MCU panel, reslution is 240x320) --->
```

需先配置RGB与VOP相关驱动。



- 相关驱动配置:

```
1  -> RT-Thread rockchip rk2108 drivers
2      [*] Enable RGB
3      [*] Enable VOP
```

因io复用的关系，若使用显示功能需关闭codec相关的配置。

- 关闭codec:

```
1  > RT-Thread rockchip rk2108 drivers
2      > Enable Audio --->
3          Enable Audio Card --->
4              [ ] Enable es8388 audio card
```

- 版型配置:

```
1  -> RT-Thread board config
2      -> Board Type
3          (X) Enable RK2108B_EVB board
```

- ASR唤醒通路配置:

```
1  -> RT-Thread rockchip rk2108 drivers
2      [*] Enable PCM
3      [*] Enable I2STDM
4      [*] Enable I2STDM0
5          Enable Audio Card --->
6              [*] Audio Cards
7                  [*] Enable internal adc audio card
8                  [ ] Enable digital mic audio card
9                  [ ] Enable es8388 audio card
```

- 开启app:


```
1 | -> RT-Thread application
2 | [*] IoT display enable
3 | [*] rk iot app
```

3.3 应用开机启动

RT-Thread 开机启动应用的代码在applications/start_app/application.c中，test_thread_entry接口。

4. 文件系统打包

bsp/rockchip/rk2108目录下，执行 `./mkroot.sh resource/userdata/normal`，即会在Image目录下生成root.img文件。

该脚本将bsp/rockchip/rk2108/resource/userdata/normal目录下的文件或目录制作成Fat12格式的文件系统，大小为setting.ini中设定的root分区 PartSize的大小。

5. 固件编译与生成

bsp/rockchip/rk2108目录下执行

```
1 | ./build.sh
```

编译无误后，固件将自动调用./mkimage.sh生成固件。

固件路径在当前Image/Firmware.img。

6. 固件烧录

地址	名称	文件
0x00000000	LoaderToDDR	Image/rk2108_db_loader.bin
0x00000000	firmware	Image/Firmware.img
0x0000xxxx	root	Image/root.img

注：0x0000xxxx 为root分区的在setting.ini中设置的PartOffset偏移地址。

文件系统仅需烧写一次，后续可以只烧写loader和Firmware。

root.img 在Display实例中可不烧录。

7. 运行测试

固件烧录成功，运行启动，对着EVB板说出如下命令词：（思必驰固件）

```
1  "xiao you xiao you" ,  
2  "da kai re shui qi" ,  
3  "guan bi re shui qi" ,  
4  "tiao gao wen du" ,  
5  "tiao di wen du" ,  
6  "si shi wu du" ,  
7  "liu shi wu du" ,  
8  "qi shi wu du" ,  
9  "da kai yi jun" ,  
10 "guan bi yi jun" ,  
11 "da kai chen yu" ,  
12 "guan bi chen yu" ,  
13 "da kai wan yu" ,  
14 "guan bi wan yu" ,  
15 "wo yao xi zao" ,  
16 "da kai su re bu shui" ,  
17 "guan bi su re bu shui",
```

可看到经过DSP处理，并成功唤醒后在ASR实例中会播放如下内容的应答提示音；

在显示应用实例中，在MCU显示屏上显示如下内容的应答字符。

```
1  "我在/在的/请吩咐/您请说" （随机出现）  
2  "已为您打开热水器",  
3  "已为您关闭热水器",  
4  "已为您调高温度",  
5  "已为您调低温度",  
6  "已为您调至45°",  
7  "已为您调至65°",  
8  "已为您调至75°",  
9  "已为您打开抑菌",  
10 "已为您关闭抑菌",  
11 "已为您打开晨浴",  
12 "已为您关闭晨浴",  
13 "已为您打开晚浴",  
14 "已为您关闭晚浴",  
15 "好的, 已为您打开热水器",  
16 "已为您打开速热补水",  
17 "已为您关闭速热补水",
```

8. 参考文档

1. [Rockchip_Developer_Guide_RT-Thread_Display_CN.md](#)
2. [Rockchip_Developer_Guide_RT-Thread_Display_APP_CN.md](#)