

密级状态：绝密() 秘密() 内部资料() 公开(√)

PX3SE Carplay 开发指南

(技术部，第二系统产品部)

文件状态： [] 草稿 [] 正式发布 [√] 正在修改	当前版本：	V1.0
	作 者：	黄国椿
	完成日期：	2017-07-26
	审 核：	王剑辉、蓝斌元
	完成日期：	2017-07-31

福州瑞芯微电子股份有限公司

Fuzhou Rockchips Semiconductor Co., Ltd

(版本所有,翻版必究)

版 本 记 录

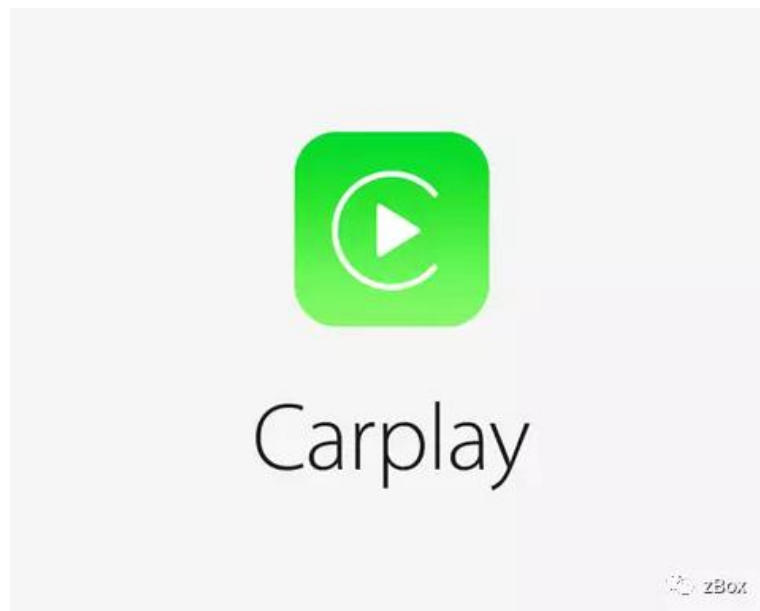
日期	修订版本	修订内容	修改人	核定人
2017-07-26	V1.0	发布初始版本	黄国椿	王剑辉/ 蓝斌元

目 录

一、产品介绍.....	1
二、PX3SE 硬件改动.....	4
1、外接 MFI 芯片.....	4
2、MFI 芯片挂接在 i2c 下的地址.....	4
3、将 OTG 接口的供电脚改为长供电.....	4
三、软件内核.....	5
1、声卡相关配置.....	5
2、USB 相关配置.....	7
四、Carplay 支持包.....	7
五、调试步骤.....	9

一、产品介绍

我司在 PX3SE Linux 平台实现车载方案开发商北京至简科技有限公司推出的手机互联解决方案 z-Link，其可完美支持 iPhone Carplay，为车载研发厂商提供完整的手机互联兼容支持，支持苹果公司原生 Carplay 功能。



等待用户连接界面



iPhone 连接界面



语音识别



拨打电话



地图导航



音乐播放



支持 iPhone 平台



二、PX3SE 硬件改动

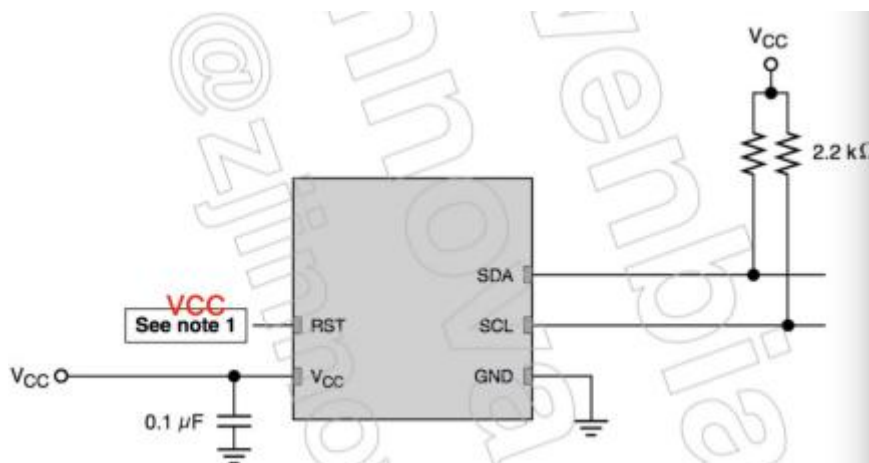
1、外接 MFI 芯片

Carplay 是一款涉及 ios 的手机车机互联产品，必须要进行苹果 MFI 认证，所以需要在 PX3SE 开发板的 I2C 接口外接苹果的 MFI 芯片，在 PX3SE 上已经给 MFI 芯片预留 i2c2 的接口。



2、MFI 芯片挂接在 i2c 下的地址

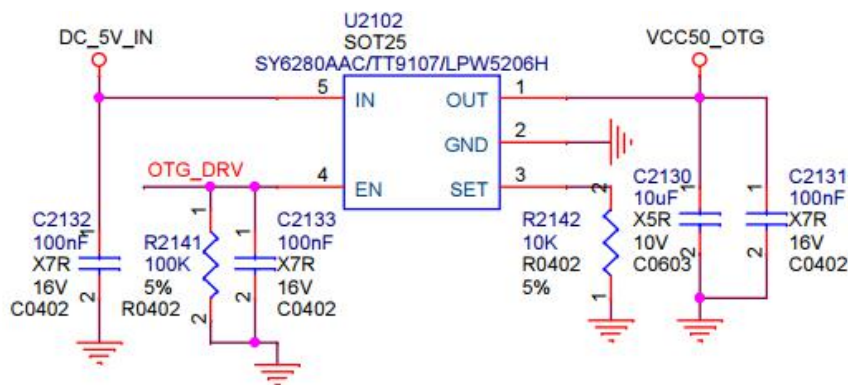
该芯片的 RST 引脚将决定其挂接在 i2c 下的地址，拉高地址为 0x11，拉低地址为 0x10，同时 i2c 的两条通信线 SDA 和 SCL 需要上拉 2.2 千欧的上拉电阻。



3、将 OTG 接口的供电脚改为长供电

iphone 通过 OTG 与 PX3SE 开发板连接进入 Carplay，iphone 作为 host，PX3SE 作为 device，而 iPhone 仍需要供电。

在开发板中将如下器件的第 4、5pin 短接，使其持续使能输出 vcc50_OTG 的供电信号。



三、软件内核

在 SDK 的内核目录 `../kernel/drivers/usb/gadget` 下：

已添加 iphone 认证文件：f_iap.c；

已修改 Carplay 相关通信文件：f_ncm.c；

已在 android.c 中添加对 iap 和 ncm 的支持。

注：目前的 android.c 中只支持 iap 和 ncm，因为使能 iap 和 ncm 后将其关闭或者再去使能其他功能，会导致无法连接 Carplay。

1、声卡相关配置

Carplay 在使用过程中会申请 PX3SE 的音频资源，否则会出现相关应用开启后，Carplay 不断重启连接。

codec 配置：

在 kernel/目录 make menuconfig 配置如下：

Search (CONFIG_SND_RK_SOC_ES8396)

Symbol: SND_RK_SOC_ES8396 [=y]

Type : tristate

Prompt: SoC I2S Audio support for rockchip - ES8396

Location:

-> Device Drivers

-> Sound card support (SOUND [=y])

-> Advanced Linux Sound Architecture (SND [=y])

-> ALSA for SoC audio support (SND_SOC [=y])

同时根据 PX3SE 硬件原理图在板级文件 dts 加入相应的 Codec 配置:

```
rockchip-es8396 {
    compatible = "rockchip-es8396";

    dais {
        dai0 {
            audio-codec = <&es8396>;
            audio-controller = <&i2s0>;
            format = "i2s";
        };
        dai1 {
            audio-codec = <&es8396>;
            audio-controller = <&i2s0>;
            format = "i2s";
        };
    };
};

&i2c2{
    es8396: es8396@10 {
        compatible = "es8396";
        reg = <0x11>;
        spk-con-gpio = <&gpio0 GPIO_D6 GPIO_ACTIVE_HIGH>;
        lineout-con-gpio = <&gpio3 GPIO_C4 GPIO_ACTIVE_HIGH>;
        clocks = <&clk_i2s_2ch_out>;
        clock-names = "mclk";
        status = "disabled";
    };
};
```

注：PX3SE 的 SDK 开发板有 V10 和 V11 两个版本，其音频的硬件有差异，对于 V10 版本的开发板，需要在内核：

```
git revert 3e5d9a49bb5ba80dd617a8b740b9626a0ed0b279
```

2、USB 相关配置

Enable IPV6 protocol:

Search (CONFIG_IPV6)

Symbol: IPV6 [=y]

Type : tristate

Prompt: The IPv6 protocol

Location:

-> Networking support (NET [=y])

-> Networking options

-> TCP/IP networking (INET [=y])

USB 模式的软件切换

Host 模式:

```
echo 1 > /sys/devices/10180000.usb/driver/force_usb_mode
```

Device 模式:

```
echo 2 > /sys/devices/10180000.usb/driver/force_usb_mode
```

iphone 的 Carplay 连接时会把 PX3SE 软件切换到 device 模式，退出 Carplay 连接时软件会切回 host 模式。

四、Carplay 支持包

PX3SE 的 SDK 中 Carplay 支持包被指定在工程根目录下的路径为../device/rockchip/px3-se/carplay:

carplay

└─ airplayutil

└─ lib

```
|   ├── libAirPlay.so
|   ├── libAirPlayStart.so
|   ├── libAirPlaySupport.so
|   ├── libAirPlayWait.so
|   ├── libdns_sd.so
|   └── libRKpx3SE.so
└── mdnsd
    ├── siri.png
    ├── unz-link.sh
    └── z-link
        └── z-link.sh
```

z-link.sh: 使能 iap 和 ncm 并配置 usb 和网络相关脚本文件;

mdnsd: 要从 iphone 中获取到认证后的所有数据;

z-link: 就是认证和连接 Carplay 的主程序;

lib: Carplay 认证和通信所需要的库;

siri.png: Carplay 在 qt 界面的应用图标;

unz-link.sh: 退出 Carplay 应用时恢复 adb 功能并将 PX3SE 切换成 device;

airplayutil: 该文件默认没有拷贝到制作的根文件系统, 用户需要的话自行拷贝

到../buildroot/output/target/bin/目录下, 主要用于测试 i2c 是否和 MFI 芯片通信正常, 使用方法见调试步骤;

在该目录下, SDK 编译过程中, 会执行 rk_make.sh 脚本并根据功能配置包 package_config.sh 将 Carplay 支持包的内容拷贝到需要制作的根文件系统中:

```
...
if [[ $enable_carplay =~ "yes" ]];then
    echo "enable carplay"
    cp $(pwd)/carplay/siri.png $BUILDROOT_TARGET_PATH/etc/
    cp $(pwd)/carplay/lib/*.so $BUILDROOT_TARGET_PATH/lib/
```

```
cp $(pwd)/carplay/z-link* $BUILDROOT_TARGET_PATH/bin/
cp $(pwd)/carplay/unz-link.sh $BUILDROOT_TARGET_PATH/bin/
cp $(pwd)/carplay/mdnsd $BUILDROOT_TARGET_PATH/bin/
fi
...
```

五、调试步骤

- ### 1、使用 mfi 测试工具查看 i2c 是否与 mfi 芯片正常通信:

[illegible]

- 其中会得到 4 个字节的 mfi id: 0x0 0x0 0x2 0x0, 以及 908 bytes 数据。
- ## 2、执行 Carplay 的配置脚本并查看 usb0 和 lo

```
source z-link.sh
```

ifconfig

lo Link encap:Local Loopback

inet addr:127.0.0.1 Mask:255.0.0.0

inet6 addr: ::1/128 Scope:Host

UP LOOPBACK RUNNING MTU:65536 Metric:1

RX packets:0 errors:0 dropped:0 overruns:0 frame:0

TX packets:0 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:0

RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

usb0 Link encap:Ethernet HWaddr 16:8C:0C:0B:32:ED

inet6 addr: fe80::4859:5aff:fe42:efab/64 Scope:Link

UP BROADCAST MULTICAST MTU:1500 Metric:1

RX packets:0 errors:0 dropped:0 overruns:0 frame:0

TX packets:0 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

3、查看是否启动 mdnsd 进程:

ps | grep mdnsd

4、执行 z-link 即可启动 Caplay 连接应用。