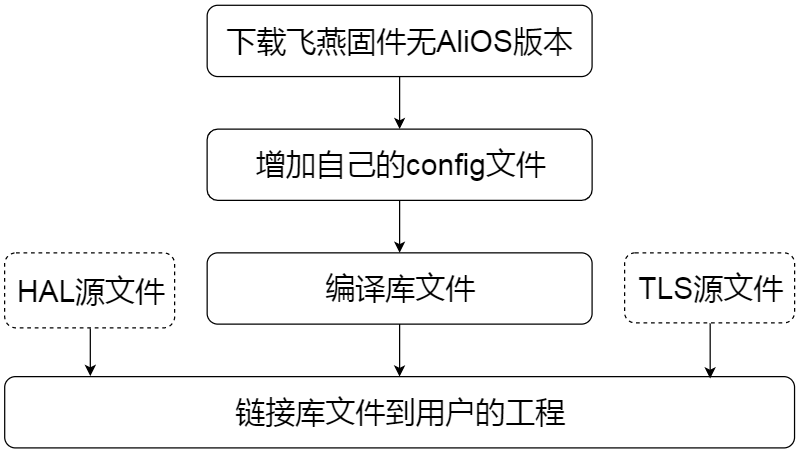
飞燕C-SDK适配FreeRTOS帮助文档

操作流程

整体集成流程如下图，其中HAL和TLS最好以源文件的形式在用户工程进行编译，否则如果在飞燕的工程中编译可能需要解决很多头文件的依赖问题。



移植步骤（以飞燕固件无AOS的V1.3.0版本为例）

第一步 下载代码

飞燕固件无AOS的版本V1.3.0，下载地址参考飞燕帮助文档：

https://help.aliyun.com/document_detail/135278.html?spm=a2c4g.11186623.6.641.6ef444e3es97dK

- 无AliOS Things的SDK（基于Link Kit V2.3.0）
  如果您基于其他操作系统开发（例如Linux、FreeRTOS、Windows等），请选择该版本SDK。

版本号	下载方式	SDK（无AliOS Things）
V1.3.0	zip压缩包	生活物联网平台SDK（V1.3.0）
	git链接	git clone git@code.aliyun.com:living_platform/ali-smartliving-devi

第二步 编译飞燕SDK的代码

配置交叉编译器路径

文件build-rules/settings.mk中修改TOOLCHAIN_DLDIR := /home/mytoolchain 配置编译器的文件夹所在的路径，然后修改build-rules/funcs.mk里面的函数Relative_TcPath增加编译器的相对路径：

```
define
( \
    case $(1) in \
        xtensa-lx106-elf-gcc ) \
            echo "gcc-xtensa-lx106-linux/main/bin" ;; \
        arm-none-eabi-gcc ) \
            echo "gcc-arm-none-eabi-linux/main/bin" ;; \
        nds32le-elf-gcc ) \
            echo "bin" ;; \
    esac \
)
endef
```

增加config文件

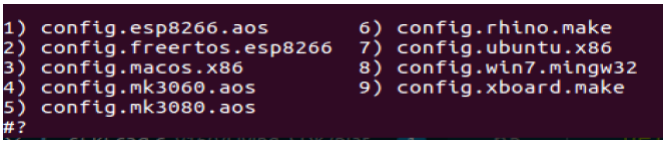
参考src/board/文件夹里面的config增加一个自己的config，例如增加config.freertos.esp8266文件，内容如下：

```
#添加芯片平台相关的配置
CONFIG_ENV_CFLAGS += \
    -DBOARD_ESP8266 -u call_user_start \
    -fno-inline-functions \
    -ffunction-sections \
```

```
    -fdata-sections \
    -mlongcalls \
    -Wl,-static

#配置编译器选项
CONFIG_ENV_CFLAGS += -Os

#配置不需要编译的子模块
CONFIG_src/ref-impl/tls         :=
CONFIG_src/ref-impl/hal         :=
CONFIG_examples                 :=
CONFIG_tests                    :=
CONFIG_src/tools/linkkit_tsl_convert :=

#交叉编译器的前缀，这里不要带路径
CROSS_PREFIX := xtensa-lx106-elf-
```

编译库文件

首先根据功能需要配置SDK的功能，有两种方法配置SDK的功能，方法一是直接编辑根目录下面的make.settings文件，方法二是执行make menuconfig进行配置，配置会修改make.settings文件。
然后编译SDK，先执行make clean，然后执行make reconfig选择你的配置，如下图输入数字2就是选择config.freertos.esp8266配置



然后执行make，如果没有编译错误生成的文件在output/release文件夹下面，用户将库文件output/release/lib/libiot_sdk.a链接到自己的工程中，同时将目录output/release/include下面的头文件放置到自己的工程并配置头文件路径，在用户的应用程序中include "iot_import.h"即可，这个头文件会包含其他SDK的头文件，注意如果修改了文件config.freertos.esp8266都需要再次执行make reconfig才能生效。

常用make相关命令：

命令	解释
make distclean	清除一切构建过程产生的中间文件，使当前目录仿佛和刚刚clone下来一样
make	使用默认的或已选中的平台配置文件平台配置文件开始编译
make env	显示当前编译配置，非常有用，比如可显示交叉编译链，编译CFLAGS等

命令	解释
make reconfig	弹出多平台选择菜单，用户可按数字键选择，然后根据相应的硬件平台配置开始编译
make config	显示当前被选择的平台配置文件
make menuconfig	以图形化的方式编辑和生成功能配置文件make.settings
make help	打印帮助文本

补充说明

如果希望将hal在飞燕的工程中编译，需要先注释掉config.freertos.esp8266中的CONFIG_src/ref-impl/hal :=
同时将hal源文件放在文件夹src/ref-impl/hal/os/freertos文件夹中，依赖的头文件放到prebuild/freertos/include文件夹中，飞燕固件V1.6.0之前的版本这个头文件依赖有个问题，需要修改根目录下面的project.mk里面的变量IMPORT_DIR的值改为IMPORT_DIR := $(TOP_DIR)/prebuilt，建议直接在用户自己的工程中编译HAL和TLS的代码，这样可以减飞燕SDK的编译报错。

第三步 链接库文件到用户工程

1. 为了减少编译SDK的错误，在config.freertos.esp8266里面增加了下面两行内容，就是不生成tls和hal的库，这个时候使用用户工程里面的tls，并在用户自己的工程里面实现HAL_xxx一系列函数

```
CONFIG_src/ref-impl/tls          :=
CONFIG_src/ref-impl/hal          :=
```

2. 将example/linkkit/linkkit_example_solo.c放到用户的工程中编译，然后启动一个任务执行linkit_main就可将整个SDK运行起来了
3. 解决适配过程的编译错误
4. 解决适配的运行错误，例如coap出错检查HAL_UDP_xxx.c是否适配好等

关于用户适配HAL的说明

1. 如果是linux系统，可以直接参考src/ref-impl/hal/os/ubuntu目录下面的c文件，大部分文件可以直接使用；
2. 如果不是linux的系统，例如freertos系统，参考src/ref-impl/hal/os/ubuntu下面的文件实现各个HAL函数的功能；
3. 配网相关的HAL函数比较多，关于HAL的说明可以参考飞燕官方帮助文档：https://help.aliyun.com/document_detail/155232.html?spm=a2c4g.11186623.6.670.67dc2c0b5kANxM