

海奇FreeRTOS SDK三方库集成方法

1. 客户私有代码打包成库。
 - 1.1. 源代码如何打包成库文件？
 - 1.2. 打包成的库文件如何使用？
2. 网上开源代码三方库集成

海奇FreeRTOS SDK三方库集成方法

该文档介绍了海奇FreeRTOS SDK中库的集成和使用方法，适合于所有使用海奇SDK进行开发的软件工程师。

1. 客户私有代码打包成库。

1.1. 源代码如何打包成库文件？

1. 所需更改代码的文件如下：

```
finn.wei@hichip01:cstm-code$ tree
.
├── build
│   ├── app-deps.mk          ----- 修改上层依赖关系
│   └── Makefile.link        ----- 修改 链接库
├── components
│   ├── applications
│   │   └── apps-avp
│   │       └── source
│   │           └── main.c ----- 上层调用库相关函数
│   ├── cstm_lib_sample ----- 客户代码包
│   │   ├── cstm_lib_sample.mk ----- 选择客户库所需的依赖关系等
│   │   ├── kconfig          ----- 配置选择项。
│   │   └── source
│   │       ├── cstm_lib_sample.c ----- 源码
│   │       ├── cstm_lib_sample.h ----- 头文件
│   │       └── Makefile ----- 编译组织文件，包括源码位置，头文件嵌入位置，生产的库文件位置等。
│   └── kconfig              ----- 选择客户编译源代码
```

2. build/Makefile.link 的修改：

```
--- a/build/Makefile.link
+++ b/build/Makefile.link
@@ -36,6 +36,7 @@ wholearchivelibs-$(BR2_PACKAGE_PREBUILTS_LIBDSC) += -ldsc
wholearchivelibs-$(BR2_PACKAGE_CMDS) += -lcmds
wholearchivelibs-$(BR2_PACKAGE_LVGL) += -llvgl_app
wholearchivelibs-$(BR2_PACKAGE_MEMTESTER) += -lmemtester
+wholearchivelibs-$(BR2_PACKAGE_CSTM_LIB_SAMPLE) += -lcstm_lib_sample

wholearchivelibs-$(BR2_PACKAGE_PREBUILTS_MP3DEC) += -lmp3ndec -lmp3
ifeq ($(CONFIG_CPU_MIPS32R1),y)
```

3. build/app-deps.mk 的修改：

```

--- a/build/app-deps.mk
+++ b/build/app-deps.mk
@@ -10,6 +10,7 @@ apps_deps-$(BR2_PACKAGE_LVGL) += lvgl
apps_deps-$(BR2_PACKAGE_HC_EXAMPLES) += hc-examples
apps_deps-$(BR2_PACKAGE_LIBAACELD) += libaaceld
apps_deps-$(BR2_PACKAGE_HCFOTA) += hcfota
+apps_deps-$(BR2_PACKAGE_CSTM_LIB_SAMPLE) += cstm_lib_sample
apps_deps-$(BR2_PACKAGE_LIBWS2811) += libws2811
apps_deps-$(BR2_PACKAGE_AD6956F) += ad6956f
apps_deps-$(BR2_PACKAGE_AC6955F) += ac6955f

```

4. components/kconfig 的修改:

```

--- a/components/kconfig
+++ b/components/kconfig
@@ -28,6 +28,7 @@ source "unity/kconfig"
source "memtester/kconfig"
source "freetype/kconfig"
source "pcre/kconfig"
+source "cstm_lib_sample/kconfig"

```

5. components/cstm_lib_sample/kconfig 的修改:

```

1 config BR2_PACKAGE_CSTM_LIB_SAMPLE
2     bool "cstm_lib_sample"
3     default y
4     help
5         cstm add thire own lib into hichip sdk

```

6. components/cstm_lib_sample/source/Makefile 的修改

```

1 include $(BR2_MKENTRY)
2 ifeq ($(skip-makefile),)
3
4 TARGETS := libcstm_lib_sample.a
5
6 all : $(TARGETS) FORCE
7
8 cstm_lib_sample-y := cstm_lib_sample.o
9
10 obj-y = cstm_lib_sample.o
11
12 $(TARGETS): built-in.o
13     rm -f $@
14     $(Q)$ (AR) -rc $@ $^
15
16 built-in.o: FORCE
17     $(Q)$ (MAKE) $(build)=.
18
19 install: FORCE
20     install -D -m 0664 $(TARGETS) $(STAGING_DIR)/usr/lib/$(TARGETS)
21     cp -fr $(srctree)/cstm_lib_sample.h $(STAGING_DIR)/usr/include/
22
23 clean: FORCE
24     $(Q)$ (MAKE) $(clean)=.
25     rm -f $(TARGETS)
26
27 .PHONY: FORCE
28
29 FORCE:
30
31 endif # skip-makefile

```

7. 添加以上代码后, 在 make menuconfig 下, 会有如下选项, 可以根据需要选上:

```
Components
> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkey
feature is selected [ ] feature is excluded

() Prebuilt subdir
[*] prebuilts --->
Applications Configuration --->
- * Newlib --->
[*] kernel --->
[*] Cmsds --->
[ ] pthread
[ ] ffmpeg ----
[ ] hc-examples
[ ] opencore-amr library
[ ] libusb
[ ] hcfota
[ ] liblvg1 ----
[ ] bluetooth ----
[ ] quicklz
[ ] liblzo
[ ] zlib
[ ] mbedtls
[ ] cJSON
[ ] hccast ----
[ ] Unity Test
[*] memtester
[ ] freetype
[ ] pcre
[*] cstm_lib_sample
Host utilities --->
HiChip Libs --->
```

8. 修改完后，需要重新编译：`make all`

1.2. 打包成的库文件如何使用？

1. `components/applications/apps-avp/source` 的修改：

```
--- a/main.c
+++ b/main.c
@@ -14,6 +14,8 @@
#include <freertos/FreeRTOS.h>
#include <freertos/task.h>

+ #include <cstm_lib_sample.h>
+
static void app_main(void *pvParameters);
int main(void)
{
@@ -34,6 +36,8 @@ static void app_main(void *pvParameters)
    setenv("TZ", CONFIG_APP_TIMEZONE, 1);
    tzset();

+    cstm_api_test();
+
    console_init();
    /* console loop */
    console_start();
```

2. 网上开源代码三方库集成

1. 可以参考 `components\cjson` 或者 `components\memtester`，与章节1的区别是源代码的路径不同，其余一样。