Makefile 與 .config 全面解說文件

# Makefile 完整逐段說明文件（搭配 .config）

---

## 🔹 基礎設定與變數載入  
```makefile  
CONFIG\_FILE ?= .config  
-include $(CONFIG\_FILE)  
PROJ ?= MST9U  
PARA ?= 0  
```  
- `CONFIG\_FILE ?= .config`：若未事先指定 `CONFIG\_FILE`，則預設使用 `.config` 作為設定檔。  
- `-include $(CONFIG\_FILE)`：包含 `.config`，即使不存在也不會報錯，通常用來定義各種 build 環境變數。  
- `PROJ`：專案代號，預設為 MST9U（主控 SoC 名稱）。  
- `PARA`：控制是否啟用 CodeSonar 掃描，預設為 0（未啟用）。

---

## 🔹 Build Options（建構選項）  
```makefile  
ifeq ($(CHIP\_FAMILY),MST9U)  
VERSION ?= Retail  
else  
VERSION ?= Debug  
endif  
```  
- 根據晶片家族 `CHIP\_FAMILY` 設定版本型態，MST9U 對應 Retail（正式量產版），其他情況為 Debug（除錯版本）。

```makefile  
AP\_NAME = AP  
BL\_NAME = BL  
AP\_COMPRESS\_NAME = AP\_C  
MERGE\_NAME = MERGE  
RES\_NAME = RES  
BL\_BIN = $(ROOT)/boot/$(PROJ)/BOOT.bin  
```  
- `AP\_NAME`, `BL\_NAME`：指定 AP/Bootloader 代號。  
- `MERGE\_NAME`：合併映像檔的名稱。  
- `RES\_NAME`：Resource 資料名稱。  
- `BL\_BIN`：Bootloader 的輸出位置。

```makefile  
ifeq ($(BUILD\_TARGET),BLOADER\_SYSTEM)  
 AP\_NAME = BOOT  
endif  
```  
- 如果是建構 Bootloader 系統，則改名為 BOOT。

---

## 🔹 工具鏈與編譯器設定  
```makefile  
ifeq ($(OS\_TYPE),nos\_aeon)  
CROSSCOMPILE = aeon-  
AEON\_FLAG = -march=$(AEON\_TYPE) -mhard-div -mhard-mul -EL -mredzone-size=4  
else  
$(error "NOT supported OS\_TYPE, Please config OS\_TYPE=nos\_aeon")  
endif  
```  
- `CROSSCOMPILE`：設為交叉編譯器前綴字，如 aeon-gcc、aeon-ld 等。  
- `AEON\_FLAG`：針對 aeon 平台的架構選項（endianness、除法、乘法、redzone 等）。

```makefile  
ifeq ($(PARA), 1)  
CC = cpptestscan --cpptestscanProjectName=UTProject aeon-gcc ...  
else  
CC = aeon-gcc  
endif  
```  
- 根據 `PARA` 是否啟用靜態分析（CodeSonar）來決定使用哪一種 `gcc` 包裝。

---

## 🔹 資料夾結構與路徑設定  
```makefile  
ROOT = .  
BINDIR = ./Bin\_$(PROJ)  
OBJDIR = ./Obj\_$(PROJ)  
BINPATH = $(BINDIR)  
OBJPATH = $(OBJDIR)  
```  
- 指定工作目錄位置，分別對應到最終 binary 與 object file 的儲存位置。

```makefile  
ifeq ($(CHIP\_FAMILY),MST9U)  
COREDIR ?= ./core  
COREDIR\_F ?= $(CURDIR)/core  
endif  
```  
- `COREDIR` 是來源程式的目錄。  
- `COREDIR\_F` 是其絕對路徑。

---

## 🔹 編譯旗標與警告設定  
```makefile  
CC\_MTOPTS += -DUSE\_SW\_I2C=1 -DCOMB\_3D -DBOOTLOADER\_BANK\_NUM  
WARN\_FLAGS = -Wall -Wextra -Wcast-align -Wpointer-arith -Wundef  
YOGA\_GLOBAL\_CFLAGS = -pipe -fno-exceptions -ffunction-sections $(WARN\_FLAGS)  
```  
- 使用 `-D` 定義巨集給 C 編譯器。  
- 各種 `-W...` 代表啟用多項警告訊息，提升開發品質。

```makefile  
ifeq ($(VERSION),Debug)  
 CC\_OPTS = $(CC\_OPTS0) -O0 -ggdb  
else  
 CC\_OPTS = $(CC\_OPTS0) -O2  
endif  
```  
- Debug 模式會使用不最佳化與 gdb 支援，Retail 模式則使用最佳化等級 `-O2`。

---

## 🔹 編譯與物件檔關係  
```makefile  
SRC\_C = $(filter %.c, $(SRC\_FILE))  
OBJ\_C = ${SRC\_C:%.c=$(OBJPATH)/%.o}  
```  
- 篩選出 `.c` 檔後轉換為 `.o` 物件檔放入 `OBJPATH`。

---

## 🔹 Build 規則與目標  
```makefile  
all : $(PROJ)  
$(PROJ): setup ap mbedtls merge  
```  
- 最終目標是透過執行 `make` 建構出 `ap`, `merge`，並在 `setup` 準備好編譯環境。

```makefile  
ap: $(AP\_ELF) $(AP\_BIN) $(AP\_SYM)  
```  
- 表示 `ap` 這個目標取決於這些產出檔。

---

## 🔹 RSA 加密與合併  
```makefile  
$(MERGE\_BIN): $(AP\_BIN)  
 $(BinIDPackFiles) -BIGENDIAN -CRC16ENABLE ...  
```  
- 將已產生的 `.bin` 加上校驗與額外 metadata 後產生合併檔案。

```makefile  
ifeq ($(RSA\_SIGN),y)  
 rsa\_sign ...  
endif  
```  
- 若啟用 RSA 簽章，則執行 `rsa\_sign` 產生 `.sig` 檔，並與 `.bin` 合併。

---

## 🔹 Clean 與 Stack 分析  
```makefile  
clean:  
 rm -f $(OBJPATH)/\*.o ...  
realclean:  
 rm -rf Bin\_\* Obj\_\*  
```  
- `clean` 清除中間檔案，`realclean` 刪除整個輸出資料夾。

```makefile  
checkstack:  
 $(OBJDUMP) -d $(AP\_ELF) | scripts/checkstack.pl aeon  
```  
- 檢查函式堆疊使用量，有助於嵌入式系統最佳化。

---

## 🔹 相依性處理與自動化  
```makefile  
-include $(OBJ\_S:.o=.d) $(OBJ\_C:.o=.d)  
```  
- 包含 `.d` 檔案支援 make 的自動重新編譯功能。

```makefile  
# $(call make-depend-compile,source,object,depend)  
```  
- 定義宏讓每一個編譯動作能產生對應的 `.d` 相依檔。

---

## 🧾 .config 補充內容  
範例：  
```makefile  
CHIP\_FAMILY = MST9U  
OS\_TYPE = nos\_aeon  
BUILD\_TARGET = MAIN\_AP\_SYSTEM  
AEON\_TYPE = aeonR2  
```  
- 對應到 makefile 中各處邏輯，例如 VERSION, AEON\_FLAG, build rule 的判斷條件。