# 一、Uboot源码vscode工程创建

# 二、顶层Makeile分析

 AS = $(CROSS\_COMPILE)as

 AS = arm-linux-gnueabihf-as

2、变量值

ARCH = arm

CPU = armv7

BOARD = mx6ullevk

VENDOR = freescale

SOC = mx6

CPUDIR = arch/arm/cpu/armv7

BOARDDIR = freescale/mx6ullevk

# 三、编译处理过程

## make distclean

##  make mx6ull\_14x14\_ddr512\_emmc\_defconfig

 make xxx\_defconfig的规则：

%config: scripts\_basic outputmakefile FORCE

 $(Q)$(MAKE) $(build)=scripts/kconfig $@

 1、目标为scripts\_basic

scripts\_basic:

 $(Q)$(MAKE) $(build)=scripts/basic

展开以后就是：

 make -f ./scripts/Makefile.build obj= scripts/basic

 build定义在scripts\Kbuild.include

build := -f $(srctree)/scripts/Makefile.build obj

build := -f ./scripts/Makefile.build obj

2、outputmakefile，

 因为KBUILD\_SRC，因此outputmakefile无效。

 最终要执行的命令

 make -f ./scripts/Makefile.build obj=scripts/basic

* Make -f ./scripts/Makefile.build obj=scripts/kconfig xxx\_defconfig

第一条命令：

src = scripts/basic

kbuild-dir = ./scripts/basic

kbuild-file = ./scripts/basic/Makefile

include ./scripts/basic/Makefile

\_\_build: $(if $(KBUILD\_BUILTIN),$(builtin-target) $(lib-target) $(extra-y)) \

 $(if $(KBUILD\_MODULES),$(obj-m) $(modorder-target)) \

 $(subdir-ym) $(always)

 @:

\_\_build: $(builtin-target) $(lib-target) $(extra-y) $(subdir-ym) $(always)

 @:

scripts/basic/fixdep

\_\_build: scripts/basic/fixdep

因此需要编译出scripts/basic/fixdep，对应的.c是scripts/basic/fixdep.c，

第二条命令：

Make -f ./scripts/Makefile.build obj=scripts/kconfig xxx\_defconfig

src = scripts/kconfig

kbuild-dir = ./scripts/ kconfig

kbuild-file = ./scripts/ kconfig/Makefile

include ./scripts/kconfig/Makefile

%\_defconfig: $(obj)/conf

 $(Q)$< $(silent) --defconfig=arch/$(SRCARCH)/configs/$@ $(Kconfig)

xxx\_defconfig: scripts/kconfig/conf

 $(Q)$< $(silent) --defconfig=arch/$(SRCARCH)/configs/$@ $(Kconfig)

 scripts/kconfig/conf --defconfig=arch/../configs/ xxx\_defconfig Kconfig

###  make V=1 -j12

 默认目标

PHONY := \_all

\_all:

\_all: all

all: $(ALL-y)

ALL-y += u-boot.srec u-boot.bin u-boot.sym System.map u-boot.cfg binary\_size\_check

u-boot.bin: u-boot-nodtb.bin FORCE

 $(call if\_changed,copy)

u-boot-nodtb.bin: u-boot FORCE

 $(call if\_changed,objcopy)

 $(call DO\_STATIC\_RELA,$<,$@,$(CONFIG\_SYS\_TEXT\_BASE))

 $(BOARD\_SIZE\_CHECK)

u-boot: $(u-boot-init) $(u-boot-main) u-boot.lds FORCE

 $(call if\_changed,u-boot\_\_)

u-boot-init := $(head-y)

u-boot-main := $(libs-y)

head-y= arch/arm/cpu/armv7/start.o

libs-y=保存源码的目录

libs-y += lib/ -》lib/built-in.o

libs-y保存大量的built-in.o

u-boot就是将start.o 和大量的built-in.o链接在一起。

假如

CONFIG\_ONENAND\_U\_BOOT =y

ALL-$(CONFIG\_ONENAND\_U\_BOOT)

ALL-y += u-boot-onenand.bin

# 四、链接

 链接脚本为u-boot.lds，uboot链接首地址为0X87800000，



在mx6\_common.h文件中设置CONFIG\_SYS\_TEXT\_BASE=0X87800000