**工程由\MY7020-EVM\_Alpha\_V1.0\4-软件资料\Linux\FSBL\src\fsbl-v1.0中根据7045的Freertos资料建立而来，此图1目录下的为已经可以在7020下完成主核freertos功能（led1（mio29）闪烁）。**

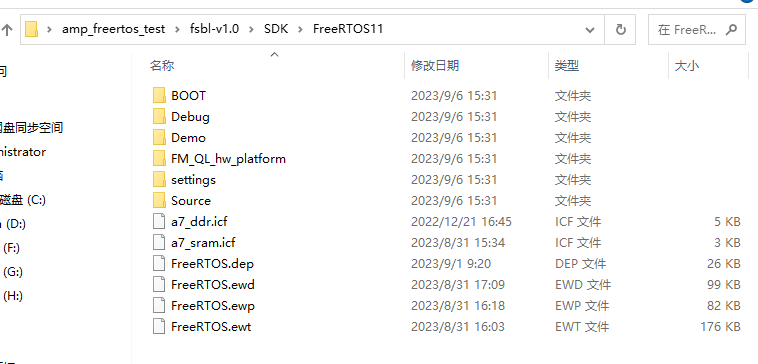


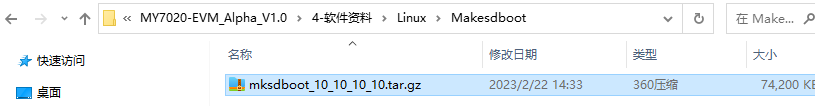
图1

**现根据《linux+freertos双系统调试记录-v1.1.pdf》和《3-4 PS端裸机与FreeRTOS案例开发手册》以及《FMQL系列开发平台AMP教程》对工程进行如下修改，ps端系统按照以前的方法正常加载，pl端随后加载也能正常运行的功能。**

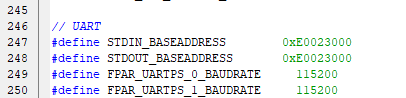
1. **开发板参数简介**
2. **启动方式**

**SD卡启动，拨码开关为1010（1~4键）**

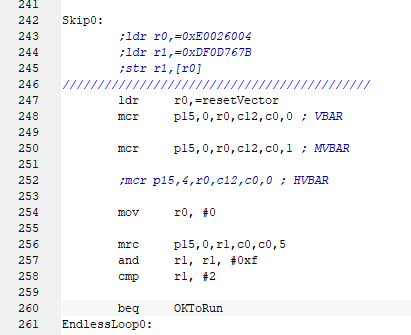
1. **SD卡装载系统**

****

1. **iar工程配置**
2. **修改工程fmsh\_ps\_parameters.h**



1. **修改cstartup.s**

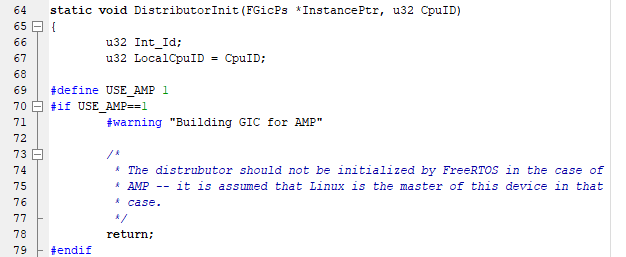


1. **修改mian.c(添加USE\_AMP的声明)**



1. **修改fmsh\_gic.c(添加USE\_AMP的声明)**





1. **修改在FreeRTOS\_tick\_config.c（添加此行）**

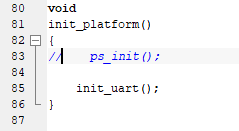


修改cache.c（注释此行）

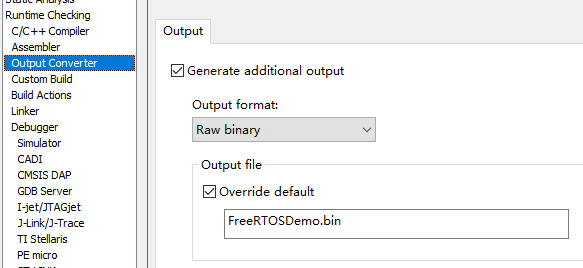


1. **修改platform.c**

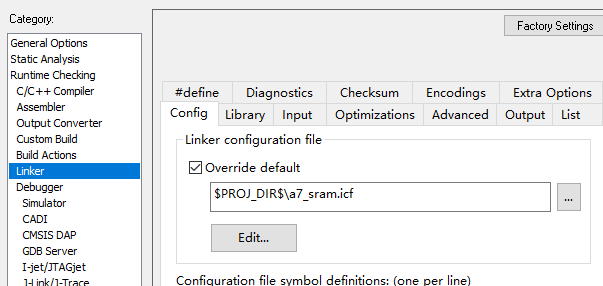
注释掉ps\_init();注释掉的作用是可以从flash和SD卡启动。

****

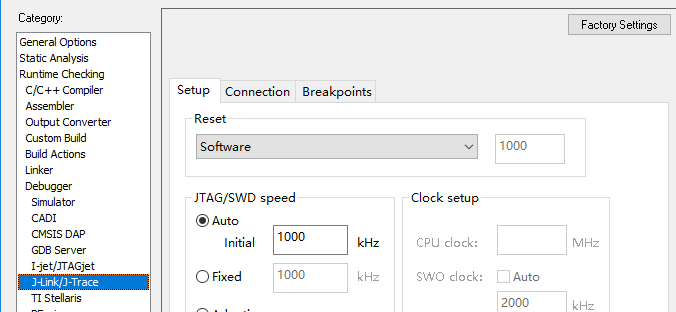
1. **修改Freertos工程配置**
   1. 修改Output Converter->output format为Raw binary



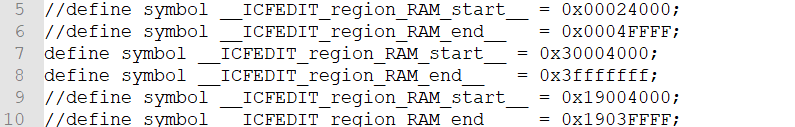
* 1. **修改Linker**



* 1. **修改J-Link/J-Trace**



1. **修改a7\_sram.icf**







其中上下地址保持一致，以0x30开头则不要用0x19开头，相反也是一样，不能混用。

1. **Bootloader配置**
2. **修改bootarg参数**

****

**setenv bootargs 'console=ttyPS0,115200 mem=767M@0x100000 maxcpus=2 loglevel=8 noinitrd root=/dev/mtdblock3 rootfstype=jffs2 rootwait rw clk\_ignore\_unused'**

1. **修改sdboot参数**

****

**setenv sdboot 'mmc rescan && echo Copying zImage/dtb from SD to RAM... && setenv bootargs 'console=ttyPS0,115200 noinitrd earlyprintk root=/dev/mmcblk0p2 rootwait rw maxcpus=2';load mmc 0:2 ${load\_addr} ${bootdir}/${kernel\_image} && load mmc 0:2 ${loadaddr} ${bootdir}/${dtb\_image} && bootz ${load\_addr} - ${loadaddr}'**

1. **amp.c文件修改**

****

这里需要使能uart0和uart1

1. **工程运行过程**
2. **Uboot启动参数查看**

**U-Boot 2018.07-fmsh-gcfcc872 (Feb 17 2023 - 15:26:30 +0000) FMSH PSOC**

**CPU: FMSH PSOC FMQL20S, Revision Code:1**

**Model: FMSH PSOC QL Demo Board**

**Board: FMSH PSOC Board**

**DRAM: 1023 MiB**

**Flash: 0 Bytes**

**NAND: 0 MiB**

**MMC: dwmmc@e0043000: 0, dwmmc@e0044000: 1**

**Loading Environment from SPI Flash... SF: Detected n25q128 with page size 256 Bytes, erase size 64 KiB, total 16 MiB**

**OK**

**In: serial@e0004000**

**Out: serial@e0004000**

**Err: serial@e0004000**

**Net:**

**Warning: ethernet@e0047000 (eth0) using random MAC address - 4a:cf:f5:de:cc:79**

**eth0: ethernet@e0047000**

**Hit <SPACE> key to stop autoboot in 2s**

**fmsh> pri**

**arch=arm**

**baudrate=115200**

**board=fmql**

**board\_name=fmql**

**boot\_targets=env eth qspi usb nand nor**

**bootaddr=0x101100**

**bootargs=console=ttyPS0,115200 mem=767M@0x100000 maxcpus=2 loglevel=8 noinitrd root=/dev/mtdblock3 rootfstype=jffs2 rootwait rw clk\_ignore\_unused**

**bootargs\_vx=gem(0,0)host:vxWorks.st h=192.168.0.51 e=192.168.0.50:ffffff00 g=192.168.0.1 u=target pw=vxTarget f=0x0 tn=fmsh\_fmql**

**bootcmd=run boottestcount || run luoji ; run $modeboot || run distro\_bootcmd**

**bootcount=0**

**bootdelay=2**

**bootdir=/boot**

**bootenv=uEnv.txt**

**boottest=0**

**boottestcount=if test $boottest = 1; then setexpr bootcount ${bootcount} + 1; saveenv; echo boottest=1 bootcount: ${bootcount}; fi;**

**cpu=armv7**

**dfu\_mmc=run dfu\_mmc\_info && dfu 0 mmc 0**

**dfu\_mmc\_info=set dfu\_alt\_info ${kernel\_image} fat 0 1\\;${devicetree\_image} fat 0 1\\;${ramdisk\_image} fat 0 1**

**distro\_bootcmd=for target in ${boot\_targets}; do run ${target}boot; done**

**dtb\_image=fmql-evm.dtb**

**emmcboot=mmc dev 1 && echo Copying zImage/dtb from eMMC to RAM... && setenv bootargs 'console=ttyPS0,115200 noinitrd earlyprintk root=/dev/mmcblk1p2 rootwait rw';load mmc 1:2 ${load\_addr} ${bootdir}/${kernel\_image} && load mmc 1:2 ${loadaddr} ${bootdir}/${dtb\_image} && bootz ${load\_addr} - ${loadaddr}**

**env\_size=0x10000**

**envboot=mmc rescan && if run loadbootenv; then echo Importing environment from ${bootenv}; env import -t ${load\_addr} $filesize; fi && boot**

**ethact=ethernet@e0047000**

**ethboot=echo boot from remote ${serverip}:${remote\_file} &&tftpboot ${loadaddr} ${remote\_file} &&bootvx**

**fdt\_high=0x10000000**

**fdtcontroladdr=3ffa78e0**

**fileaddr=19000000**

**filesize=21a0**

**fit\_image=image.ub**

**fit\_size=0x800000**

**flash\_off=0x100000**

**initrd\_high=0x10000000**

**ipaddr=192.168.0.50**

**jtagboot=echo TFTPing FIT to RAM... && tftpboot ${load\_addr} ${fit\_image} && bootm ${load\_addr}**

**kernel\_image=zImage**

**load\_addr=0x2000000**

**loadaddr=0x1000000**

**loadbootenv=load mmc 0:1 ${load\_addr} ${bootenv}**

**modeboot=sdboot**

**nandboot=echo Copying FIT from NAND flash to RAM... && nand read ${load\_addr} ${flash\_off} ${fit\_size} && bootm ${load\_addr}**

**nor\_flash\_off=0xE2100000**

**norboot=echo Copying FIT from NOR flash to RAM... && cp.b ${nor\_flash\_off} ${load\_addr} ${fit\_size} && bootm ${load\_addr}**

**psboot=echo Copying FIT to RAM by lauterbach... && bootm ${load\_addr}**

**remote\_file=vxWorks**

**sdboot=mmc rescan && echo Copying zImage/dtb from SD to RAM... && setenv bootargs console=ttyPS0,115200 noinitrd earlyprintk root=/dev/mmcblk0p2 rootwait rw maxcpus=2;load mmc 0:2 ${load\_addr} ${bootdir}/${kernel\_image} && load mmc 0:2 ${loadaddr} ${bootdir}/${dtb\_image} && bootz ${load\_addr} - ${loadaddr}**

**serverip=192.168.0.51**

**soc=fmsh**

**spiboot=echo Copying FIT from SPI flash to RAM... && sf probe && sf read ${load\_addr} ${flash\_off} ${fit\_size} && bootm ${load\_addr}**

**stderr=serial@e0004000**

**stdin=serial@e0004000**

**stdout=serial@e0004000**

**usbboot=if usb start; then echo Copying FIT from USB to RAM... && load usb 0 ${load\_addr} ${fit\_image} && bootm ${load\_addr};fi**

**vendor=fmsh**

**Environment size: 3054/131068 bytes**

1. **卸载linux下led驱动**

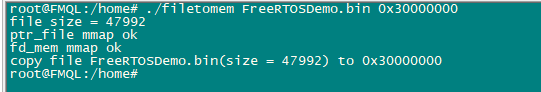
echo leds > /sys/bus/platform/drivers/leds-gpio/unbind

****

使用此指令后，my7020-EVM开发板led1指示灯灭掉。

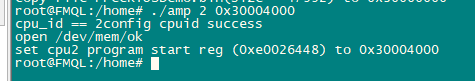
1. **加载freertos到0x3000\_0000**

./filetomem FreeRTOSDemo.bin 0x30000000



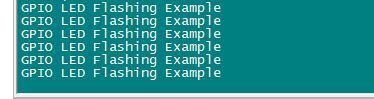
1. **使用amp程序运行FreeRTOS.bin**

./amp 2 0x30004000

****

1. **程序运行情况**

在执行完./amp 2 0x30004000指令之后，开发板led1指示灯闪烁，7020有一个uart1串口引出，看到Freertos的代码打印到此串口上。



注：uart1（RS232）接线关系为2、3交叉连接，5直连。