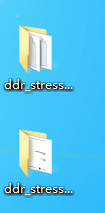
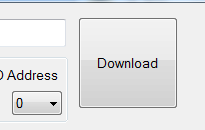
**IMX6 DDR测试工具使用**

DDR测试工具有两个版本，一个是2.6,一个是2.52我一般用2.52版本。

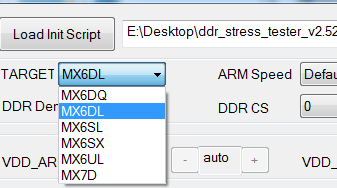


目录名字必须是英文的，否则在运行下载会出现ERROR



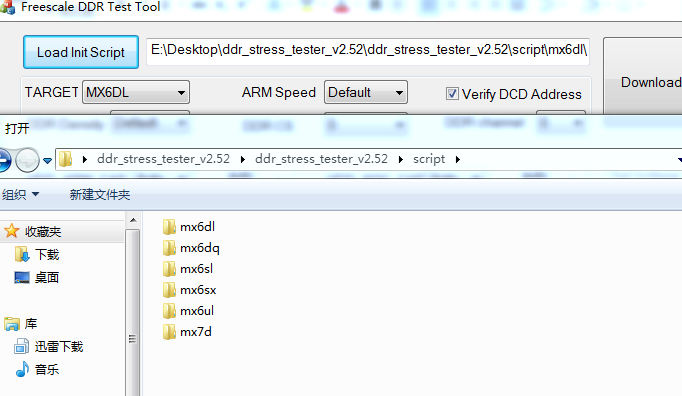
所以整个路径都必须是英文

软件选项

因为IMX6DL和IMX6solo都是一样的,只是IMX6DL比IMX6solo多一个核。记住将开发板拨码开关拨到00001100，这样才行。

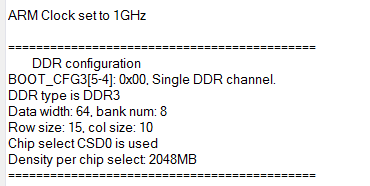
IMX6DL芯片和IMX6solo芯片都选择MX6DL

IMX6Q芯片选择MX6DQ



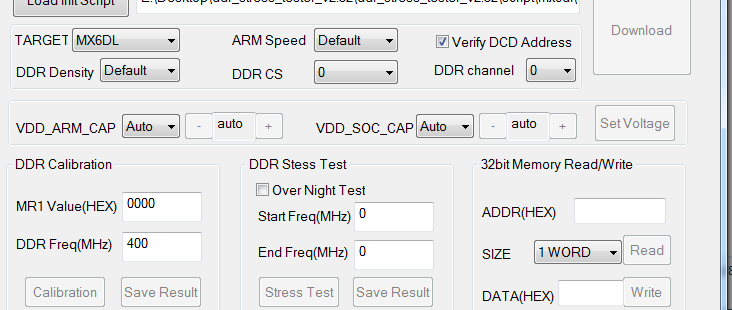
点击Load Init Scpript选择脚本，如果使用的是IMX6dl或者IMX6solo那么就选择MX6dl，如果是IMX6Q就选择MXS6qd。

然后点击DOWNload



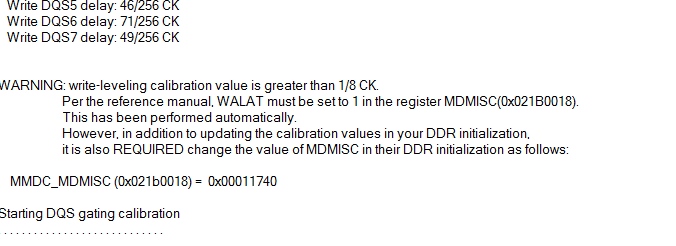
这就是正常访问芯片了。

如果换了个DDR型号的芯片，就要对DDR芯片进行校正。



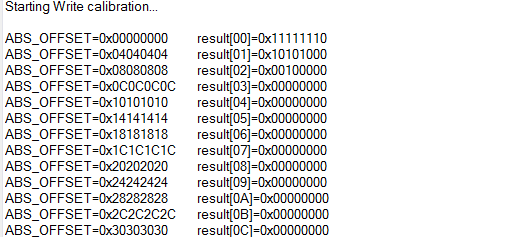
选择400MDDR运行频率

点击运行



进入DDR测试运行状态

最后会生成一个DDROFFSET报表

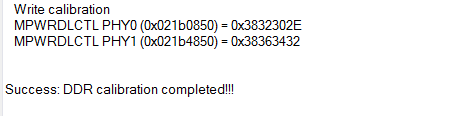


如果你没有换其他型号的内存，这里就已经结束操作了。

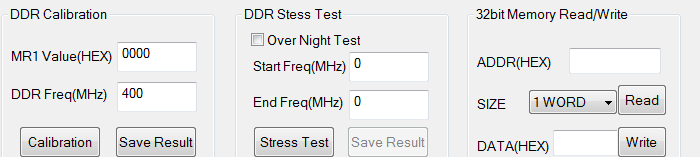
如果你换了其他型号的内存，那么在没有下载程序进开发板之前，就要先执行这个DDR\_stream软件， 生成上面这个ABS\_OFFSET列表。将列表里面的内容提取出来。

注意只需要修改ABS\_OFFSET列表里面的东西，其他不用动。

内核uboot文件中找到使用ABS\_OFFSET的代码，然后按照DDR\_stream生成的OFFSET对uboot里面的offset进行修改。然后从新编译UBOOT，烧写进内核。



这就是DDR测试成功。



这个是DDR压力测试。

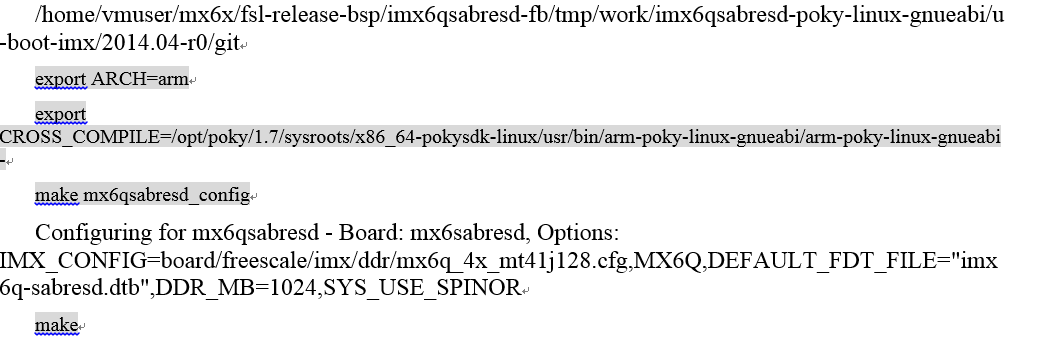
**IMX6 DDR内存大小修改指南**

参考MX6X\_3.14.28\_Uboot\_V1版本资料

下载yocto环境，在yocto的

这个目录下有uboot源码

按照uboot文档的要求



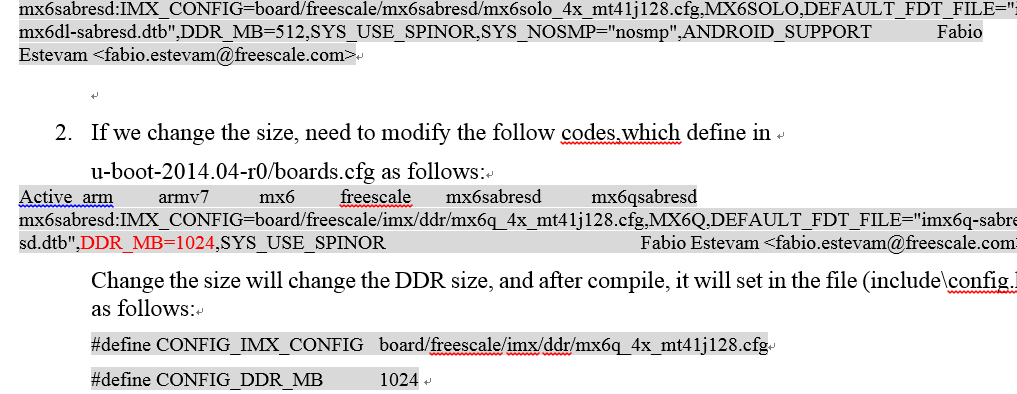
选择IMX6型号然后make就会生成u-boot.imx文件。这个文件就是给MFG下载进开发板使用的uboot文件。

但是以上配置是官方的CPU 为IMX6Q，DDR为美光的1G内存的下使用的uboot。

我现在要修改CPU型号和内存大小：

CPU : IMX6solo

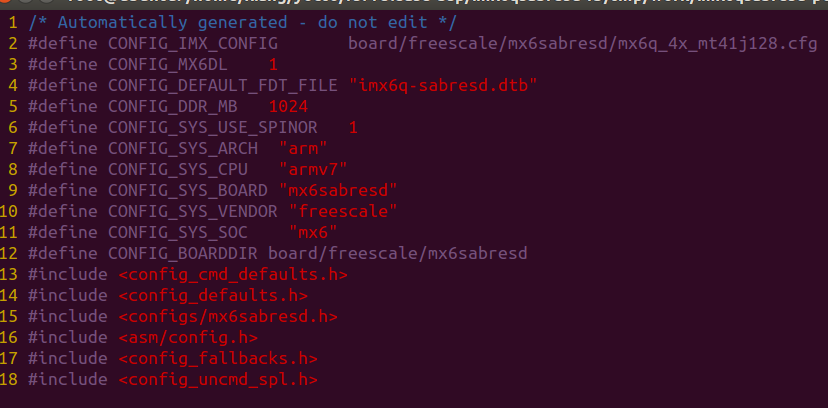
DDR：美光，但是大小为512M



按照上面的文档要求，在uboot目录下找到配置文件。先做IMX6Q的uboot



打开该文件



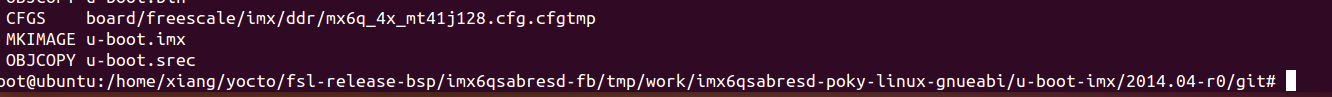
**DDR是1G**

**Dts是识别IMX6Q的dtb**

**官方的配置是IMX6Q的CPU**

按照上面这个配置去先去make mx6qsabresd\_config指定编译哪一个CPU型号

make一下，生成的uboot文件是给IMX6Q使用的。



将文件拷贝到MFG上面。记住要把u-boot.imx改成符合MFG格式的名字



根据MFG要求应该要改成这样的名字，但是不要加%号。这里有个问题？

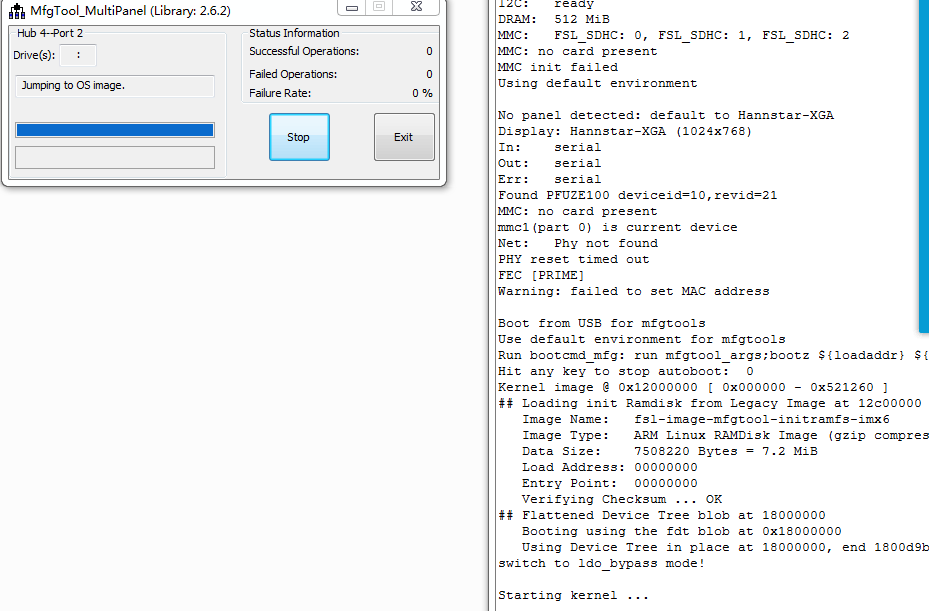
实际的MFG file和fireware文件里面名字是这样的。

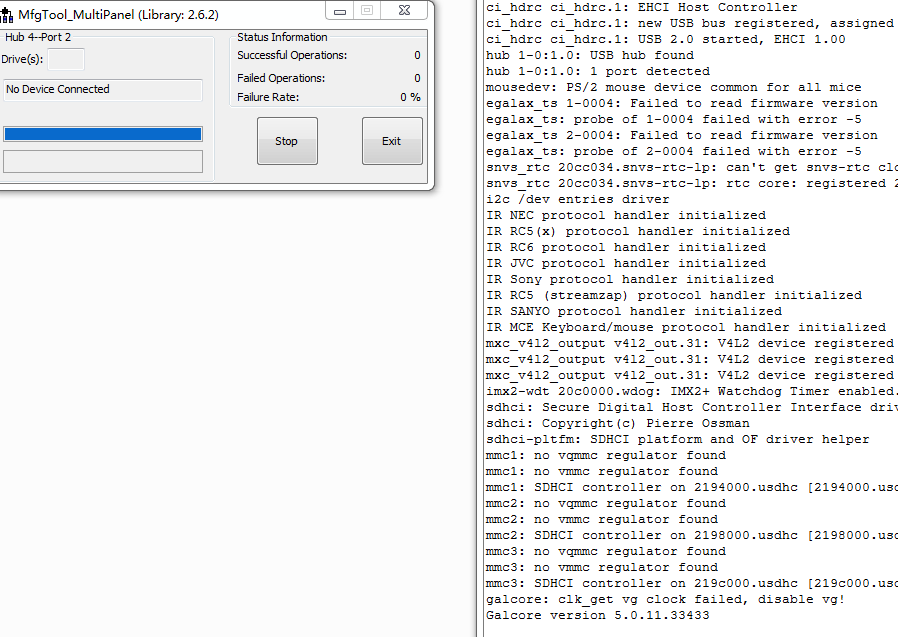
证明了%%里面的内容是可以修改的。MFG索引的应该是%号之前和%之后的内容，两个%之间的内容可以修改。

然后下载程序进开发板。

正常运行，你会发现IMX6DL和IMX6Q开发板之间的uboot可以相互使用。

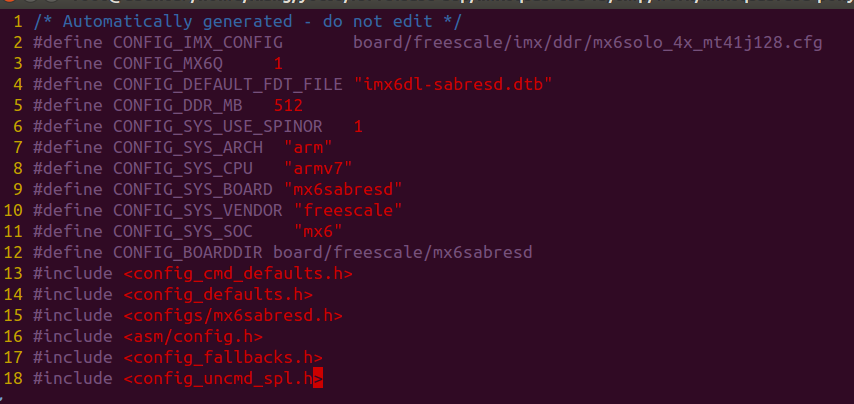
如果将IMX6Q的UBOOT烧写进imx6solo DDR 512M的开发板就会出现下面的错误：





过了jump OS后就无法下载内核和文件系统了。

所以使用IMX6solo DDR大小为512M时。就需要对config.h文件进行修改了。

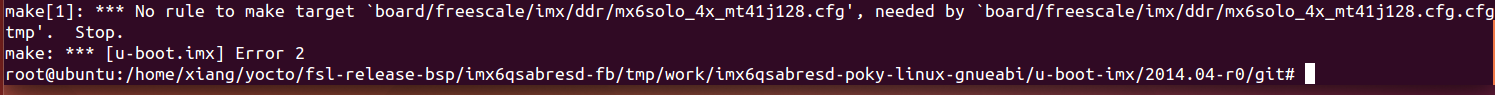


**DDR大小512M**

**这里用imx6dl的设备树，因为dl和solo是兼容的**

**这里一定要写成CPU的型号**

开始make编译



编译报错。因为你最先make 配置的是IMX6Q的CPU型号，现在你要用IMX6solo，那么你就要重新指定make配置型号。

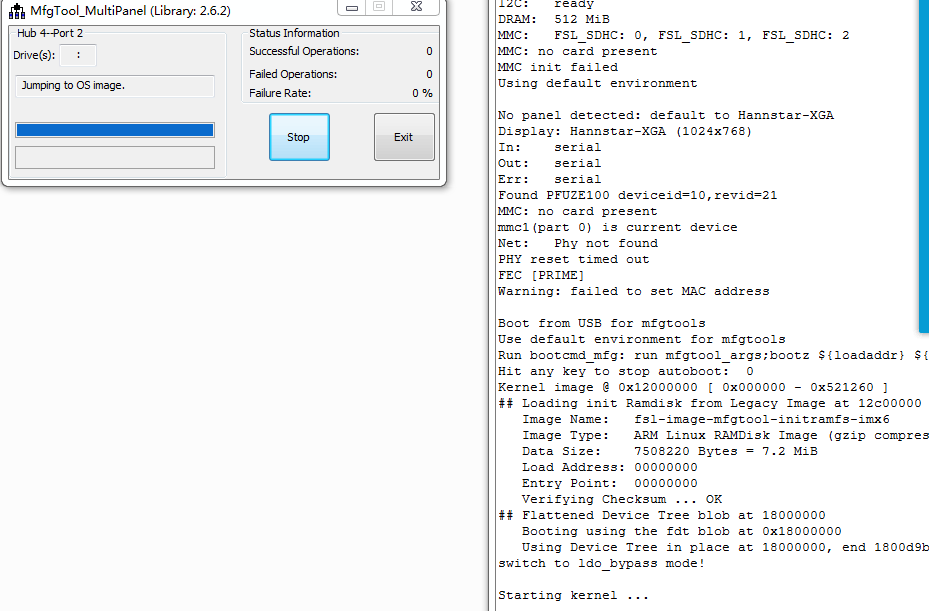
这里一定要写成mx6solosabresd\_config

而不是mx6qsabresd\_confi

或者 mx6dlsabresd\_confi

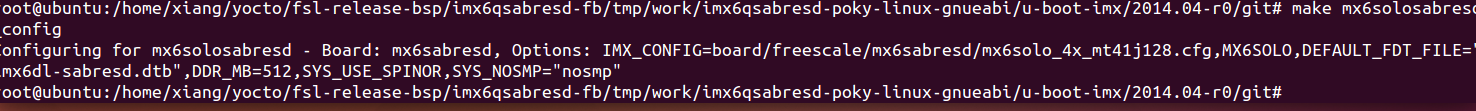
make mx6solosabresd\_config

否则在下载程序的时候会出现和上面一样的情况



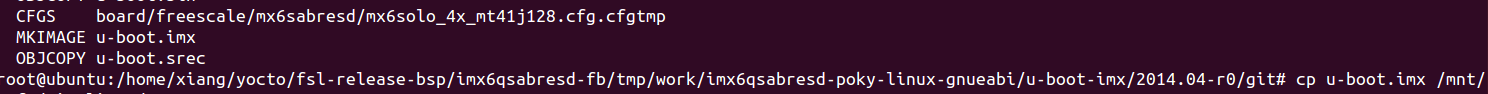
过了jump OS就无法继续正常下载了

执行make mx6solosabresd\_config之后，linux终端会回显



然后你在make

**这样就对了**

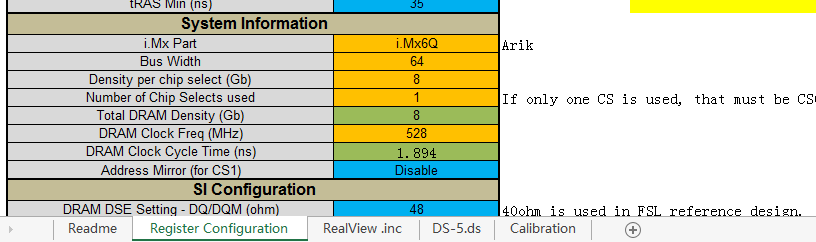


这样就是正常的IMX6solo单核，DDR大小512M的配置，下载进开发板就可以了。

**用IMX6solo单核对ddr 512M内存大小进行调试**

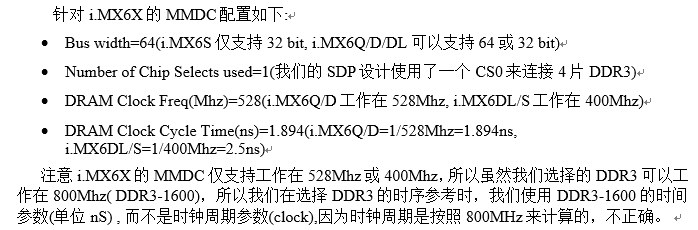
我们现在将主板换成IMX6solo单核，DDR还是美光的，只是将1G改成了512M，我们用DDR\_Strem软件对512M 两片128M\*16DDR进行压力测试。

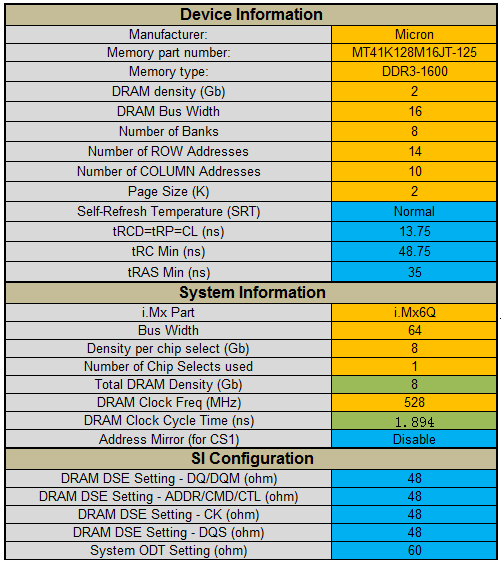
1. 我们先打开I.MX6DQSDL DDR3 Script Aid V0.10 excel格式的脚本文件 



选择regisert配置

1. 修改execl里面的参数





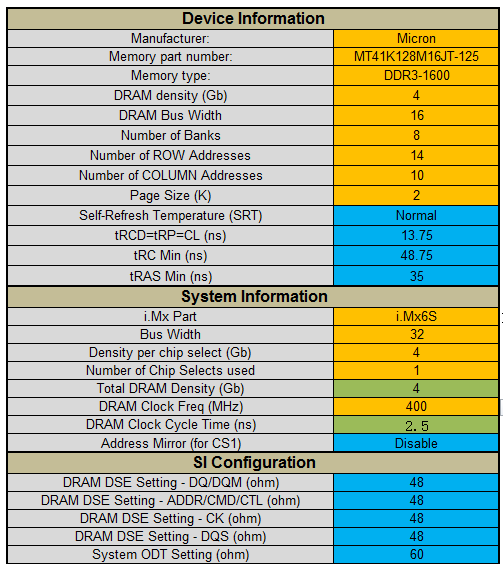
因为IMX6S只支持32位

因为IMX6S只支持32

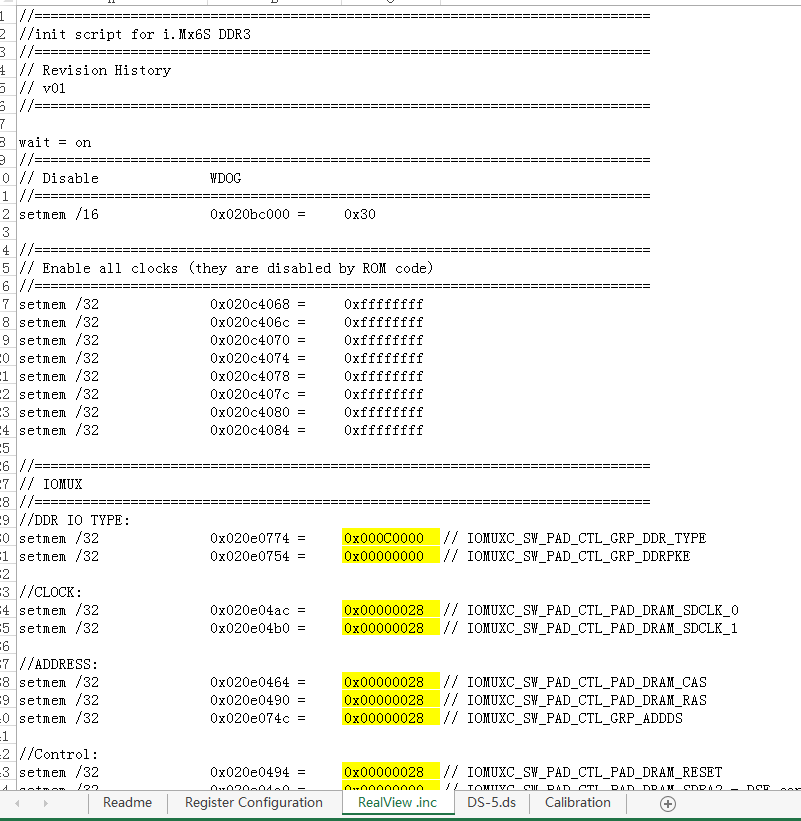
改为IMX6S

改为4Gb因为Gb是位，所以实际内存大小要除以/8

修改后就是这样

这样就修改完毕了

3.然后选择RealView



记住一定要屏蔽dcd

选择excel脚本选项

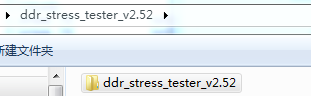


这个文件就是根据前面register Configuration选项自动设置的脚本文件。

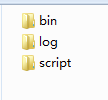
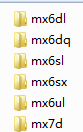
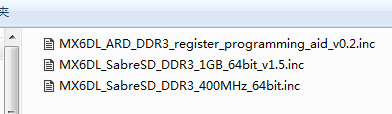


这个名字随便改，只要保证后缀是.inc就可以了。

4.然后将这个文件放在DDR\_stream文件的指定路径下



1.打开一个DDR\_stress版本

3.IMX6DL和IMX6solo是兼容的，打开IMX6DL目录

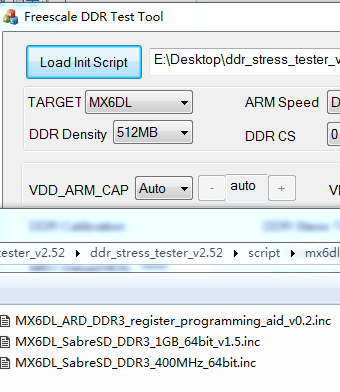
2.打开脚本目录

4.将生成的DDR 512M脚本放进来

我把保存脚本的txt文档改成了MX6DL\_SabreSD\_DDR3\_400MHz\_64bit.inc

其实这个脚本里面就是DDR3 大小512M 速度400M ，16位宽总线的参数

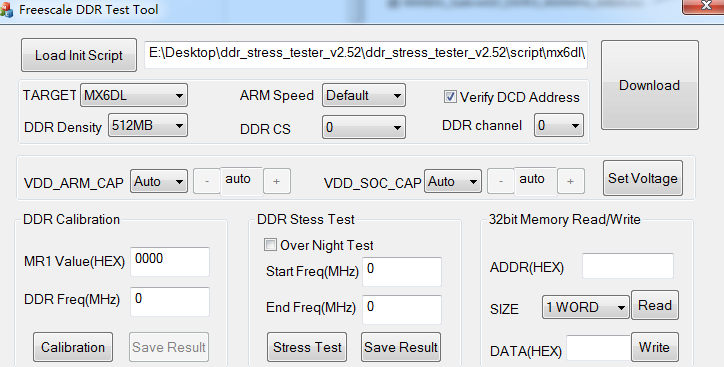
5.打开ddr\_stream调色软件,给开发办插上USBotg线,,拨码开关选择下载模式



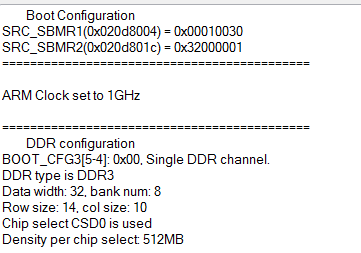
点击

载入自己刚才生成的DDR 脚本文件

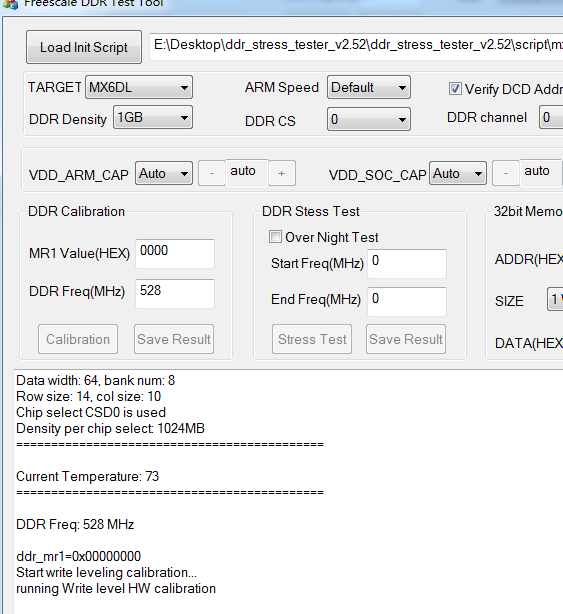
选择IMX6DL DDR选择512M 然后点击Downland



成功载入脚本就是下面这样



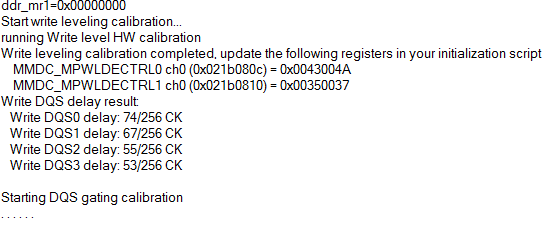
如果你选择的主板和DDR型号或者数量和设置的脚本不一样



程序就会卡在这里

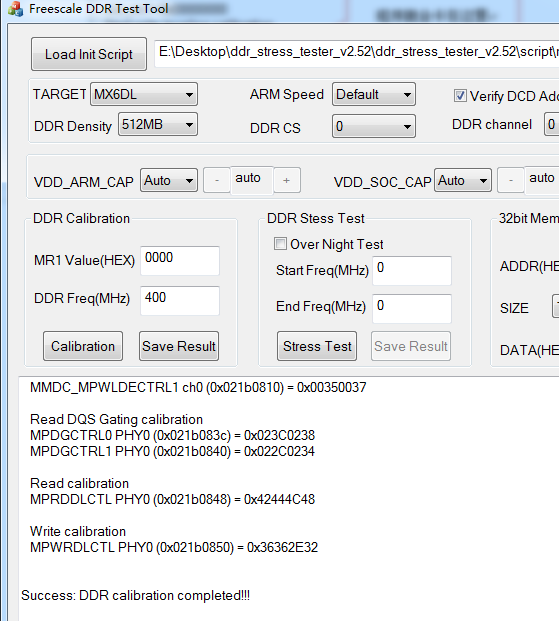
点击Download可以加载成功，但是点击DDR calibration就无法运行通过

如果你选择的脚本和主板CPU的信号和DDR是匹配的



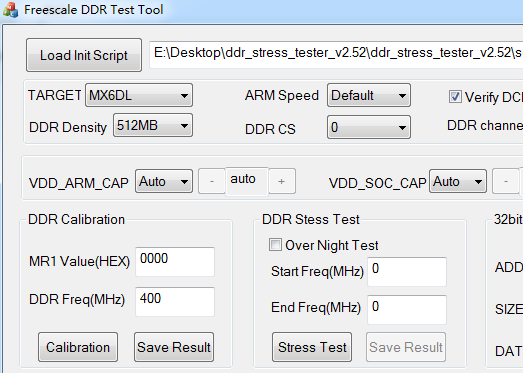
在进行DDR calibration 也就是DDR校准的时候，会成功执行到Starting DQS信号测试

等待一个不太长的时间



DDR成功校准

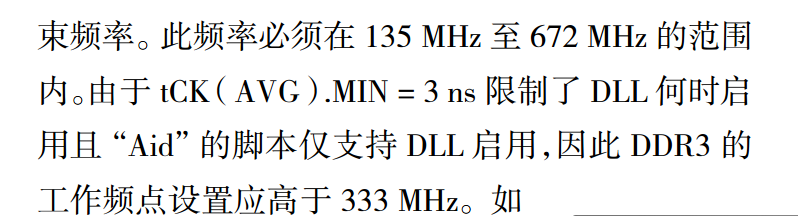
6.然后对DDR进行压力测试

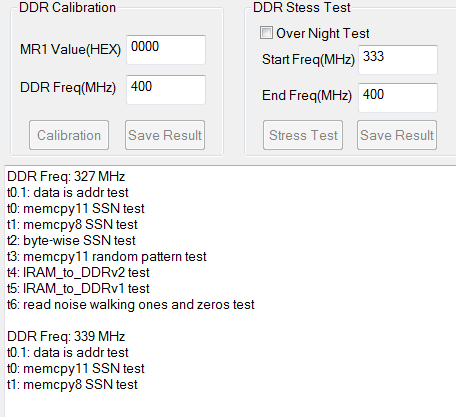


DDR压力测试

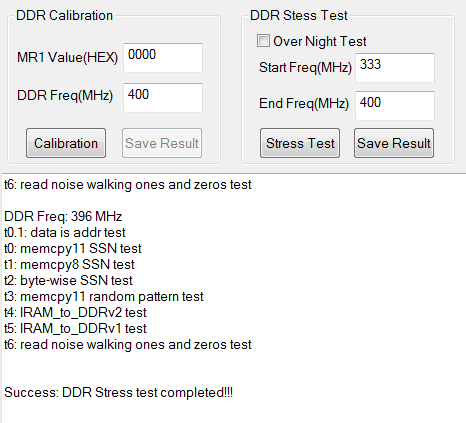
DDR校准测试

DDR压力测试频率标准



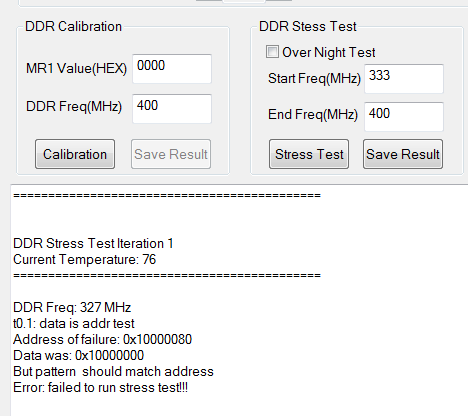


这就是DDR压力测试



DDR 3内存333M～400M速率压力测试成功。

如果我重启主板，先进行压力测试可以吗？



这就是重启主板后先Download脚本，然后不经过DDR校准，直接进入DDR压力测试，发送错误。所以必须先进行DDR校准，然后才能进行DDR压力测试，顺序不能乱。