

# ExSim\_DeT用户手册

2018年12月29日 10:52

## 文件包：

installfile	安装包	
	install	包含对已安装的旧版app及依赖文件进行清理，然后再执行ExSim_DeTail
	ExSim_DeT	如果是未安装过该工具的matlab,可双击此安装包进行安装;
example	例子	
	GKJ1224_15b	光刻机模型，分了7个子系统
用户手册	此说明文档	

## matlab版本限制：

封装的simulink模块对matlab版本有要求，如2016a封装的模块，在matlab2015b中不显示；

## 安装install

1. matlab中切到installfile所在路径；
2. 命令行执行install；

### a. 成功

```
>> install
> Processing ..., Please wait !
> ExSimDeT:installed!
```

屏幕剪辑的捕获时间: 2019-1-4 14:42

### b. 安装成功，显示警告

如下所示警告，指的是：

安装过程中删除了一些文件夹，而之前这些路径在matlab的path中有设置；

警告即告诉用户，matlab已自行将path中对应的路径或其子路径删除。

Remove '安装过程中删除的path中包含的路径' from matlab path for this MATLAB session.

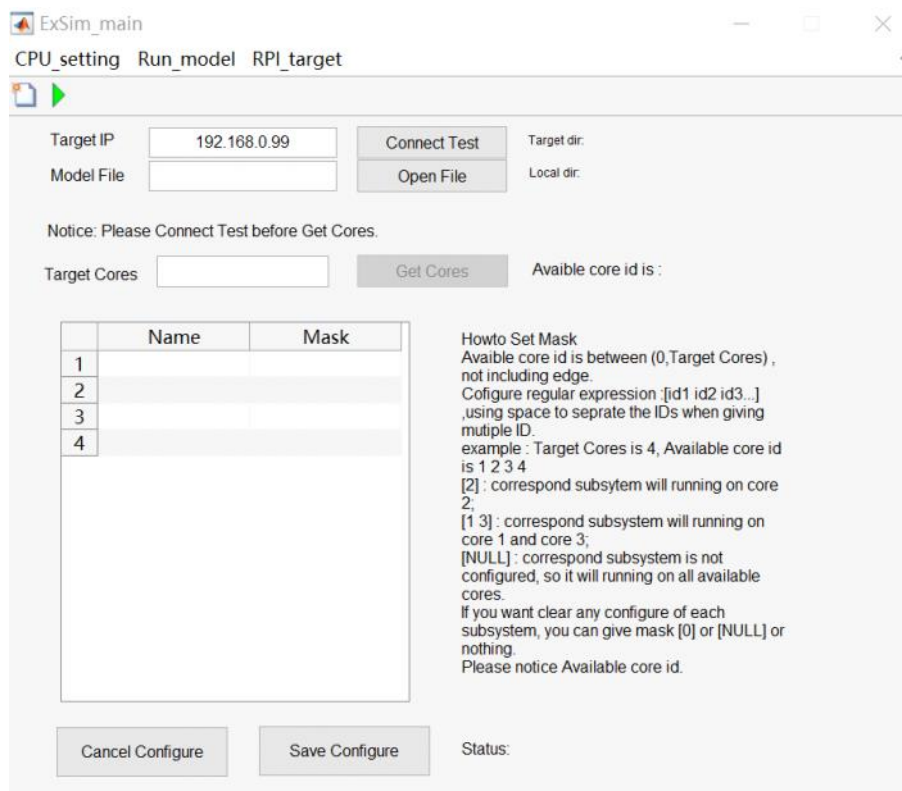
```
> Processing ..., Please wait !
Warning: Removed 'D:\Developer\MATLAB\Add-Ons\Apps\ExSim_DeT\code' from the MATLAB path for this MATLAB session.
See 'doc path' for more information.
> In install (line 15)
Warning: Removed 'D:\Developer\MATLAB\R2015b\blocks\PCIE' from the MATLAB path for this MATLAB session.
See 'doc path' for more information.
> In install (line 49)
fx > ExSimDeT:installed!
```

屏幕剪辑的捕获时间: 2019-1-4 14:43

最后显示installed!即安装成功；

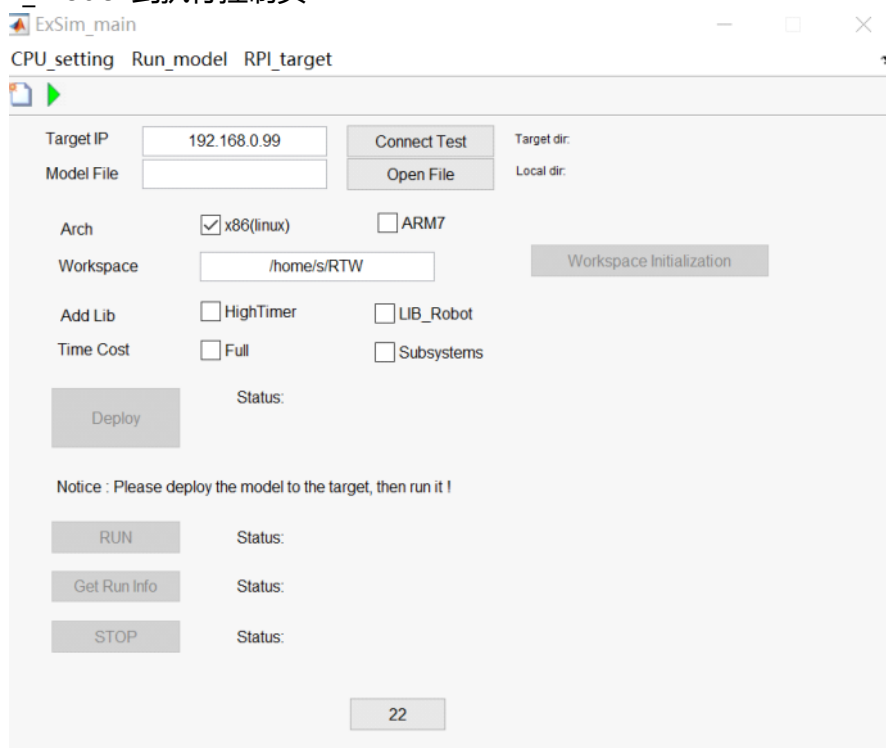
## 运行

1. Matlab -> APPS -> MY APPS，中找到ExSim\_DeT，单击打开，显示如下



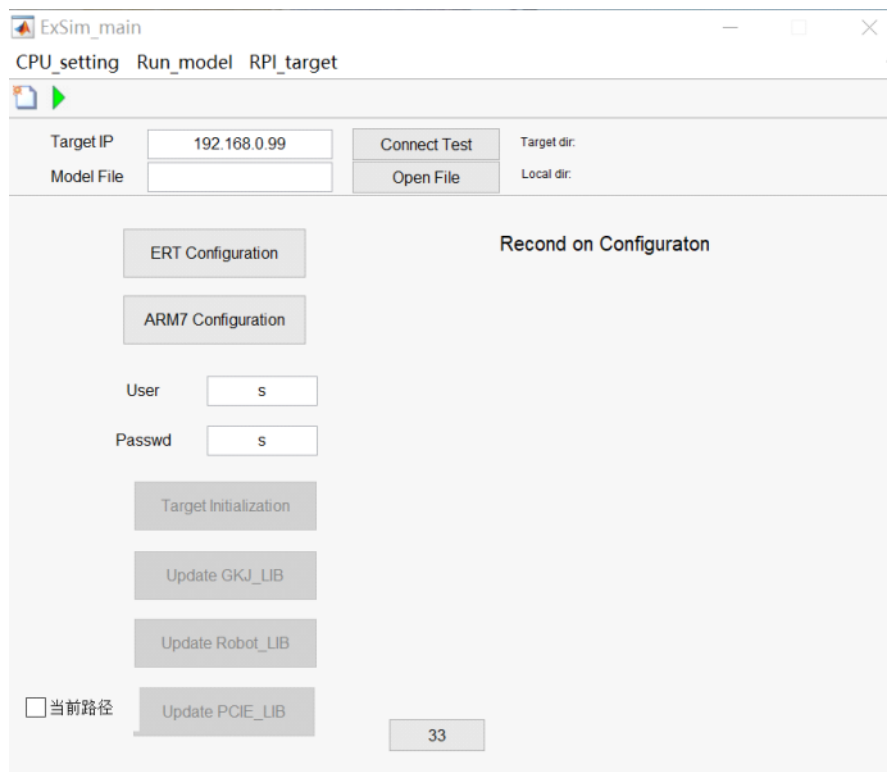
屏幕剪辑的捕获时间: 2019-1-4 14:56

## 点Run\_model 到执行控制页



屏幕剪辑的捕获时间: 2019-1-4 15:03

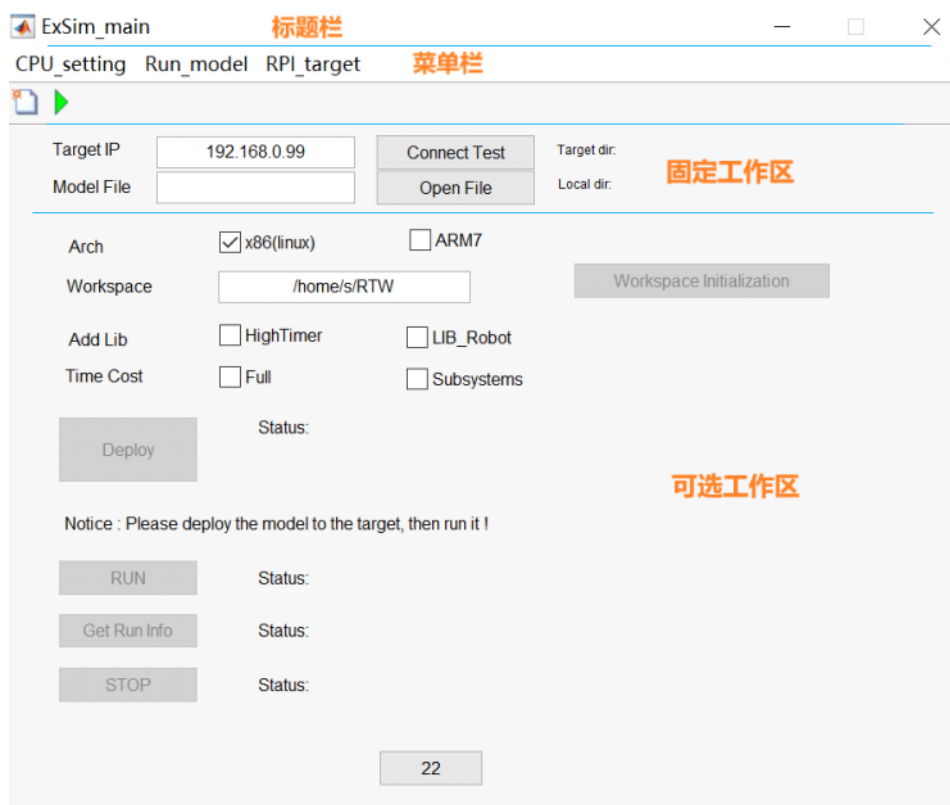
## 点RPI\_target, 到运行时（包括本机+目标机）配置页



屏幕剪辑的捕获时间: 2019-1-4 15:03

## 2. 界面介绍:

### a. 上下分为四部分



标题栏	
菜单栏	切换可选工作区
固定工作区	固定不变的部分，因为每页操作都需要知道当前的模型及目标IP

可选工作区	根据具体操作需要，在菜单栏点击实现三个页面间的切换
-------	---------------------------

b. 可选工作区，分三页，在菜单栏点击进行切换

CPU_setting	子系统分核设置	可选项，有子系统且有分核需求才设置
Run_model	目标机程序运行控制	编译参数可选，架构、库、计时操作等
PRI_target	运行时配置	配置ERTcoder工具链；目标机编译环境

### 3. 操作介绍

下文从以下几方面介绍软件具体操作

注：Config\_Local和代码生成，无需连接目标机即可操作，  
其余操作都需要连接目标机；

Connect Test	连接目标机操作
Config_Local	开发主机配置，ert和Arm7
Config_target	目标机配置，编译环境，自定义库
代码生成	模型生成代码，以example介绍
模型仿真	x86目标机运行example,仿真
分核设置	为各子系统配置运行核，以example介绍
测试子系统时间	example，以Time Cost为例，介绍编译参数选择
可选编译参数	介绍各个编译选项的作用
附：子系统创建	介绍模型中子系统的创建

## Connect Test: 登录目标机root用户

### 1. Fatal主要原因：

a. 显示状态： Connection timed out

目标机IP ping不通；

★ 注意：ping不通会导致matlab阻塞比较长的时间（不超过10s），阻塞直到超时退出；所以，如果不确定网络是否正常，使用该软件前请先在cmd中测试目标机ping通成功。

b. 显示状态： Connection refused

可以ping通，但目标机的ssh-server功能配置有问题。

c. 显示状态： note:if time more than 1s,input ‘y,then Enter in command line.

如果再ping通确认成功后，test显示该状态。原因是，系统认为第一次连接该目标机，需要安全确认。

会一直阻塞matalb,此时需要在命令行输入相应的指令

i. 失败指令：在命令行中输入 回车。

完成后显示： "WARNING-POTENTIAL SECURITYBREACH! .....Connection anandoned."（字符串较长，label中只显示到第一行， ‘！’ 号结束），按键

## GetCores失效

按下回车键enter，即取消连接操作。

```
The new rsa2 key fingerprint is:
ssh-rsa 2048 fb:65:6c:52:7c:f9:7d:53:d4:64:32:5e:0a:13:85:b1
If you were expecting this change and trust the new key,
enter "y" to update PuTTY's cache and continue connecting.
If you want to carry on connecting but without updating
the cache, enter "n".
If you want to abandon the connection completely, press
Return to cancel. Pressing Return is the ONLY guaranteed
safe choice.
Update cached key? (y/n, Return cancels connection) Connection abandoned.
```

屏幕剪辑的捕获时间: 2018-12-29 11:50

ii. 成功指令见2-b;

## 2. Succeeded: 成功后GetCores等按键生效

a. 显示 login succeeded,192.168.0.99: /root

成功登录root 用户

b. Note:more than1s...系统认为第一次连接该目标机，在命令行中输入 'y' ,回车。

完成后显示: login succeeded,%IP%:WARNING...../root, 按键GetCores生效

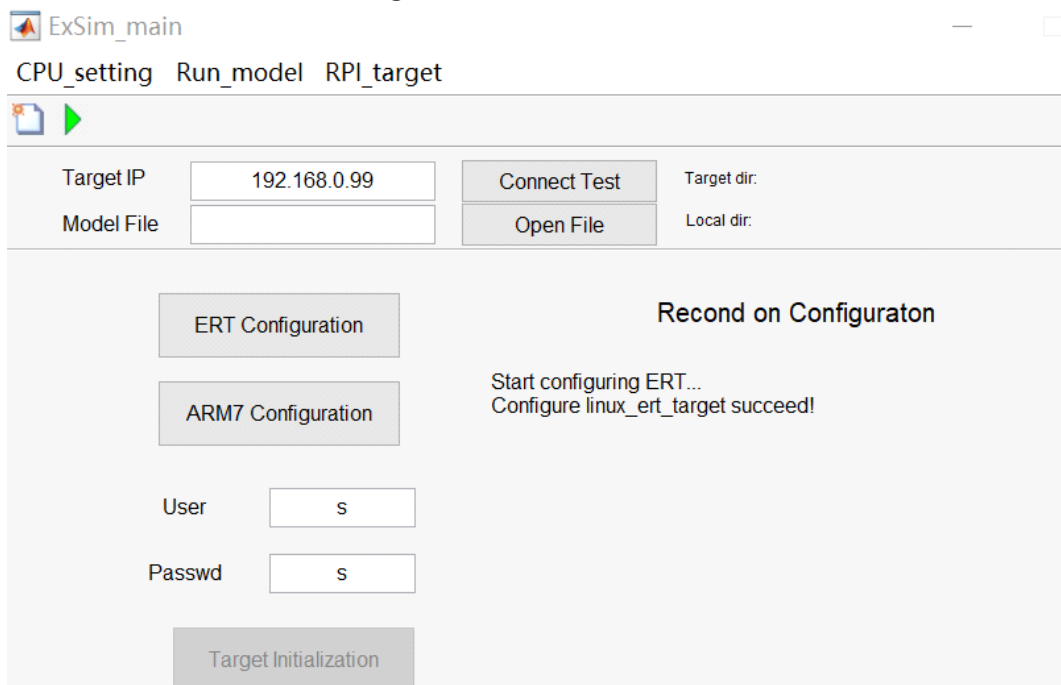
'y' 和 'n' 的区别:

'y'	更新key,下次直接login;
'n'	此次login, 但不更新key,下次连接仍需确认操作

## Config\_Local 执行需要的配置即可

1. ERT Configuration 配置ertcoder;

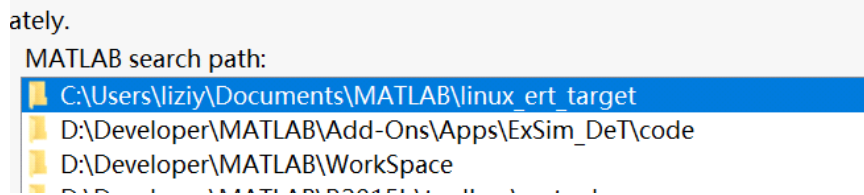
切换到RPI\_target页，点击ERTConfiguration按键;



屏幕剪辑的捕获时间: 2019-1-4 15:47

如图所示, 执行成功;

在matlab path 中可以看到新增Linux\_ert\_target路径



屏幕剪辑的捕获时间: 2019-1-4 15:47

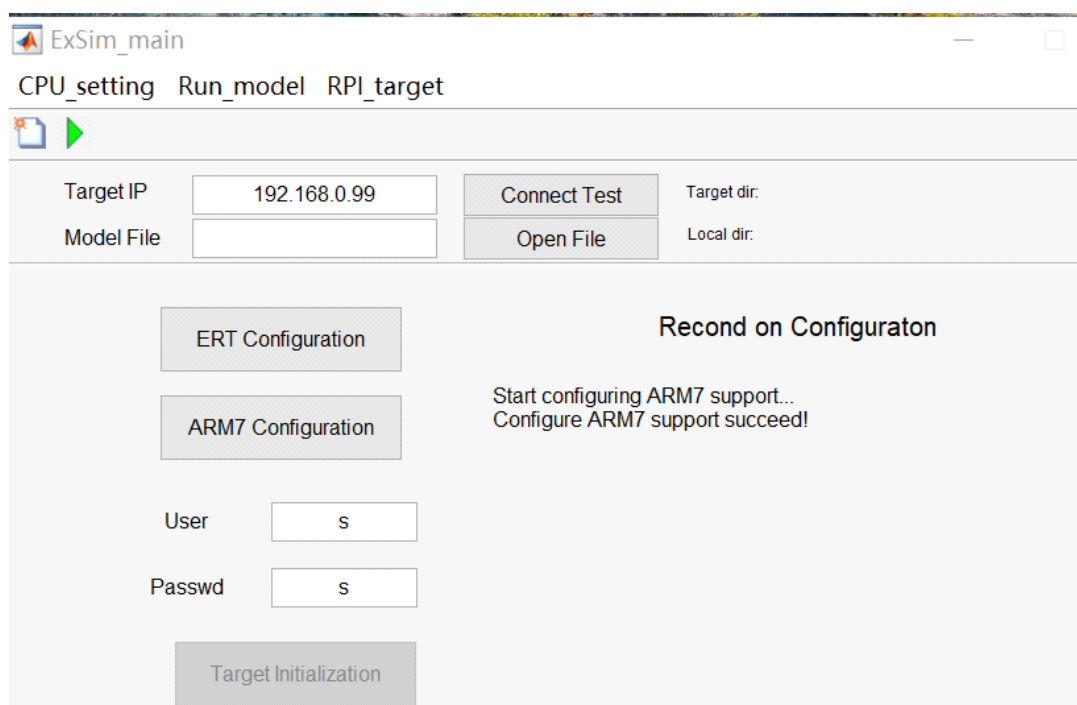
## 2. Arm7 Configuration 配置Windows-arm7交叉编译环境

切换到RPI\_target页, 点击ARM7 Configuration按钮;

当弹出系统提示: 允许此应用对设备进行更改时, 选 '是'



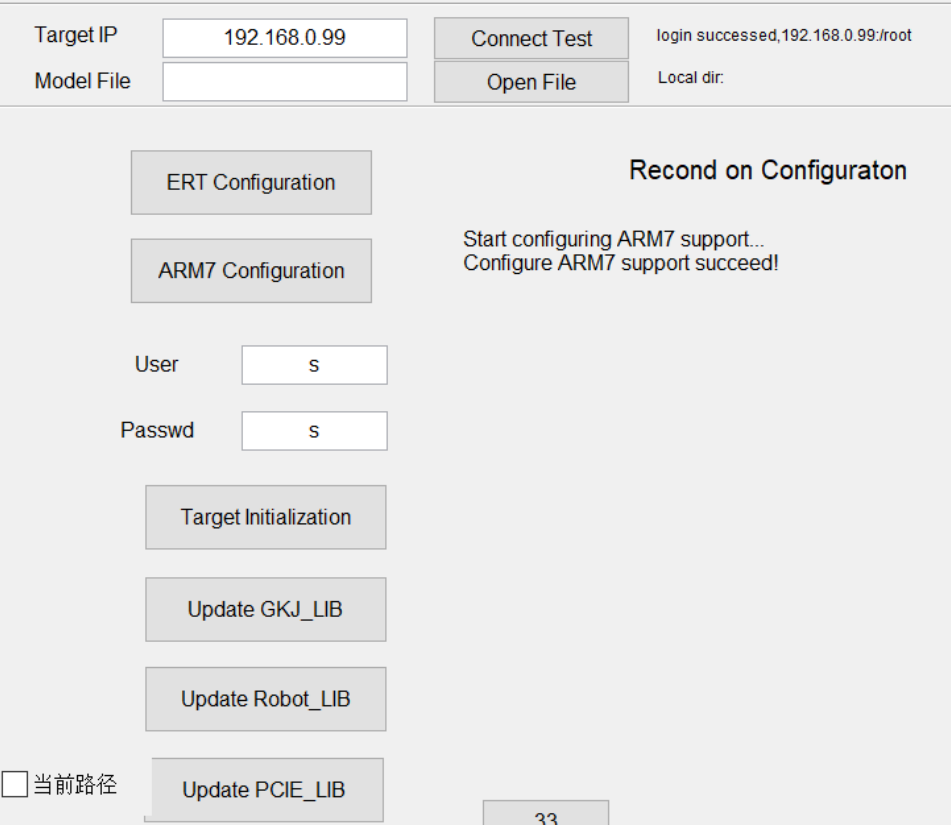
屏幕剪辑的捕获时间: 2019-1-4 16:00



如图，配置成功。

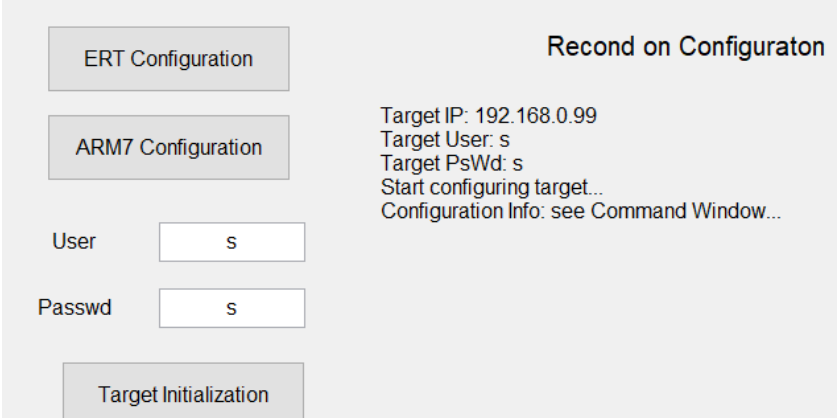
Config\_target 配置目标机，需连接成功

- 1. 连接成功后，目标机配置按键变为enable，可更改目标机用户名和密码



- 2. Target Initialization 配置目标机编译初始环境：

注：这个配置的目的是使目标机编译的依赖库，匹配开发主机的matlab版本；  
如果不确定上下是否一致，请执行此操作



更详细的执行信息在命令行中显示



```
Command Window
> Configuring Hightimer.
> Enabling user access to the EV3 USB port.
> Enabling user access to the NXT USB port.
> Checking cpu governor configuration.
> rc.local already set cpu governor to performance mode.
> Configuration of RPIt successfully completed.
> NOTE: please reboot your target for changes to take effect.
fx >>
```

屏幕剪辑的捕获时间: 2019-1-4 16:10

配置成功, 无需reboot;

### 3. 更新自定义库操作

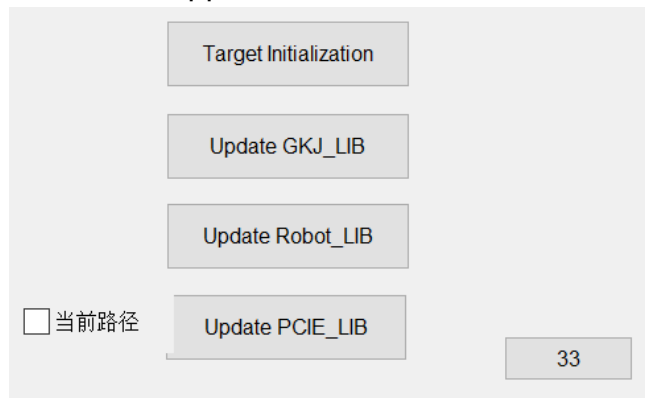
步骤2中, 已经配置了GKJ、Robot、和PCIE库;

界面中三个Update键的作用是, 无需重新安装app,来更新Libs

只替换app路径下包含lib 的压缩包后, 点击对应按键, 即可更新目标机的对应库包;

PCIE键前的' 当前路径 '勾选, 的作用是, 用户在调用了block块之后, 进行了修改, 在模型路径下生成了新的wrapper.c, 勾选"当前路径", 并Update,可快速更新库; 主要是考虑用户需要经常修改block的情况

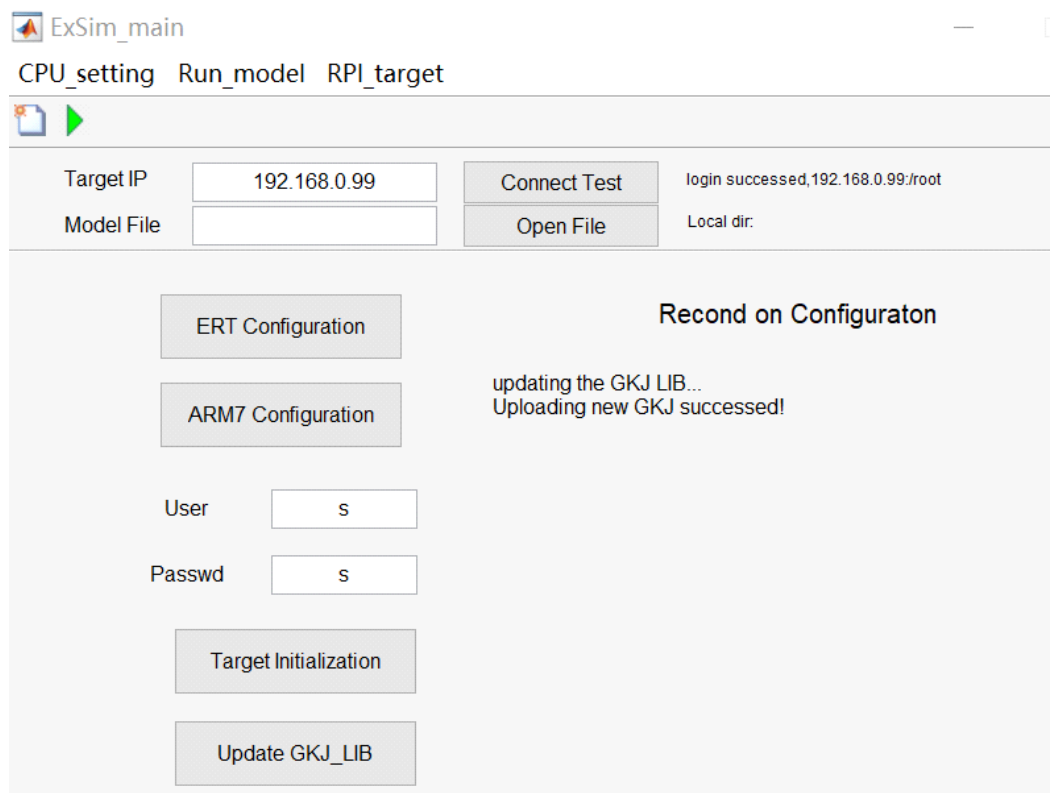
但此操作并不修改app路径下的库包, 可以不勾选, 点Update,恢复安装路径下的库包;



屏幕剪辑的捕获时间: 2019-1-4 16:14

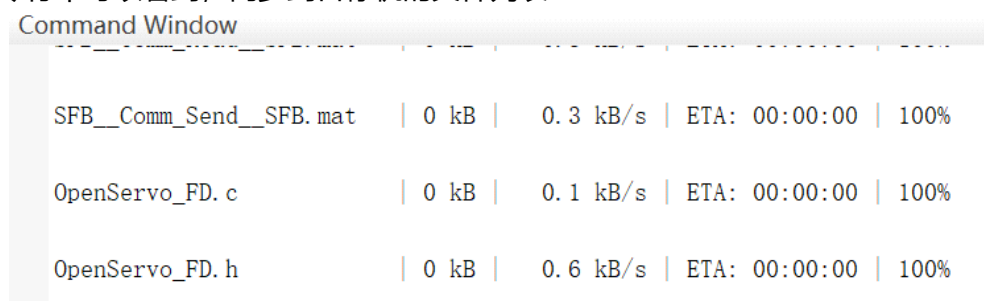
操作效果:





屏幕剪辑的捕获时间: 2019-1-4 16:18

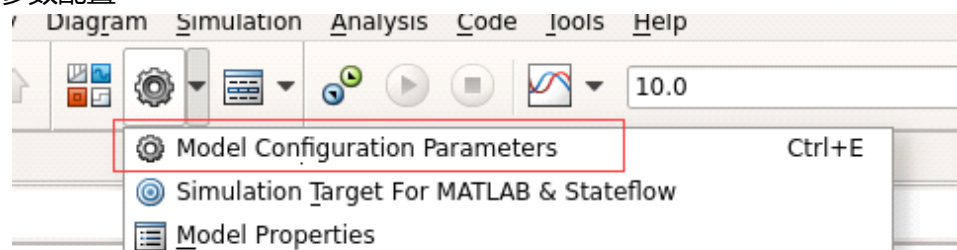
在命令行中可以看到，同步到目标机的文件列表



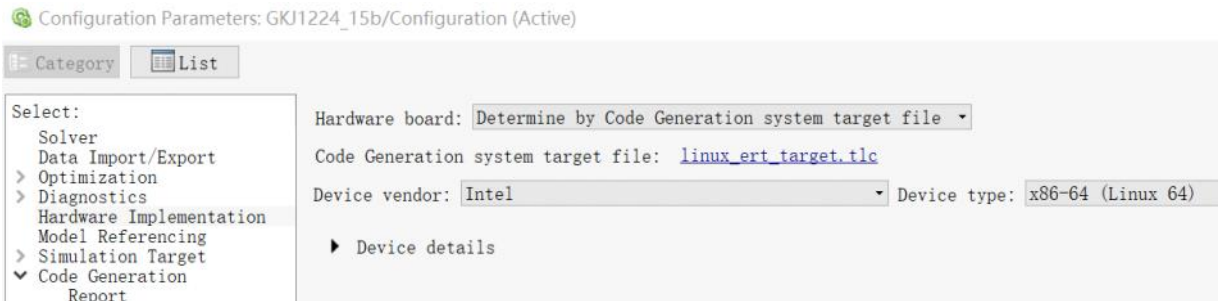
屏幕剪辑的捕获时间: 2019-1-4 16:19

## 代码生成：此操作需先完成配置ERT

### 1. 模型参数配置

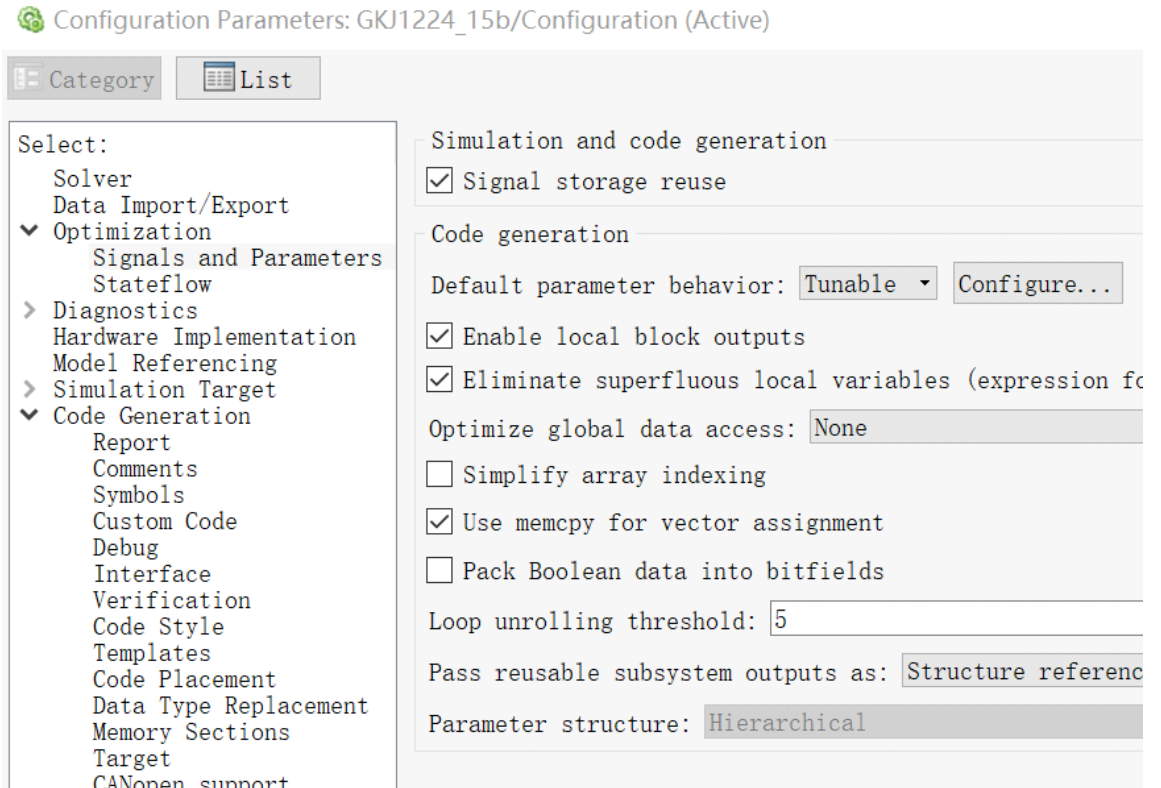


#### a. Hardware Implementation



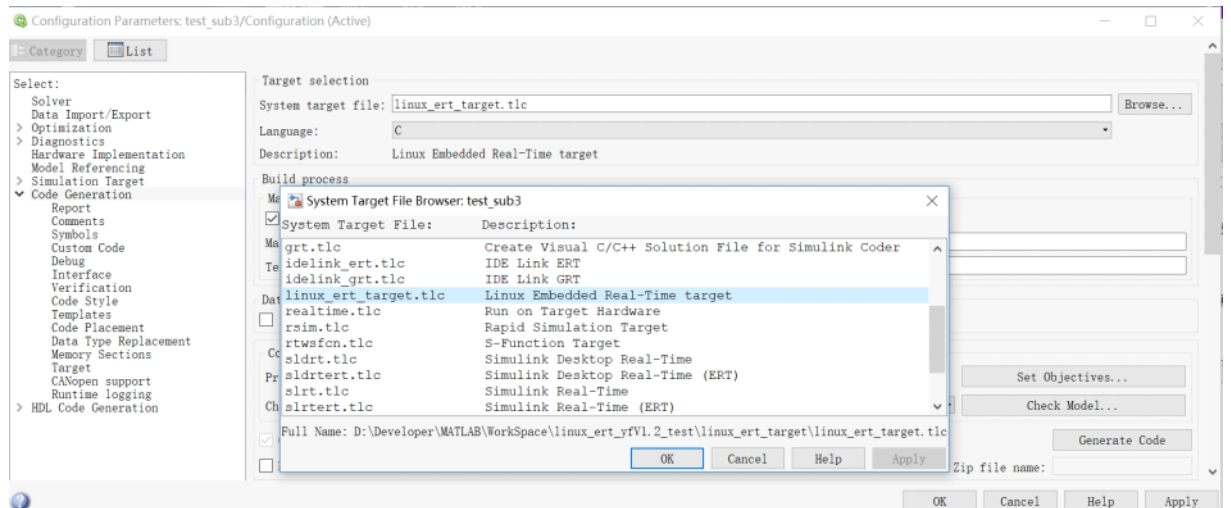
屏幕剪辑的捕获时间: 2019-1-4 16:24

## b. Optimization -> Signals and Parameters 选择Tunable

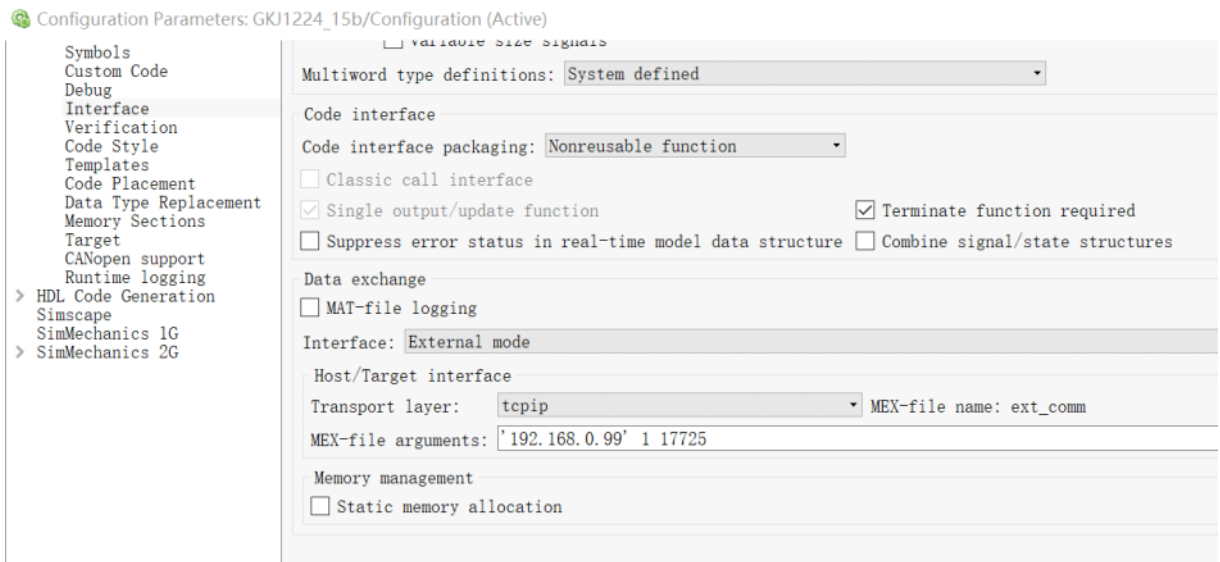


屏幕剪辑的捕获时间: 2019-1-4 16:28

## c. Code Generation: Browser 选择tlc



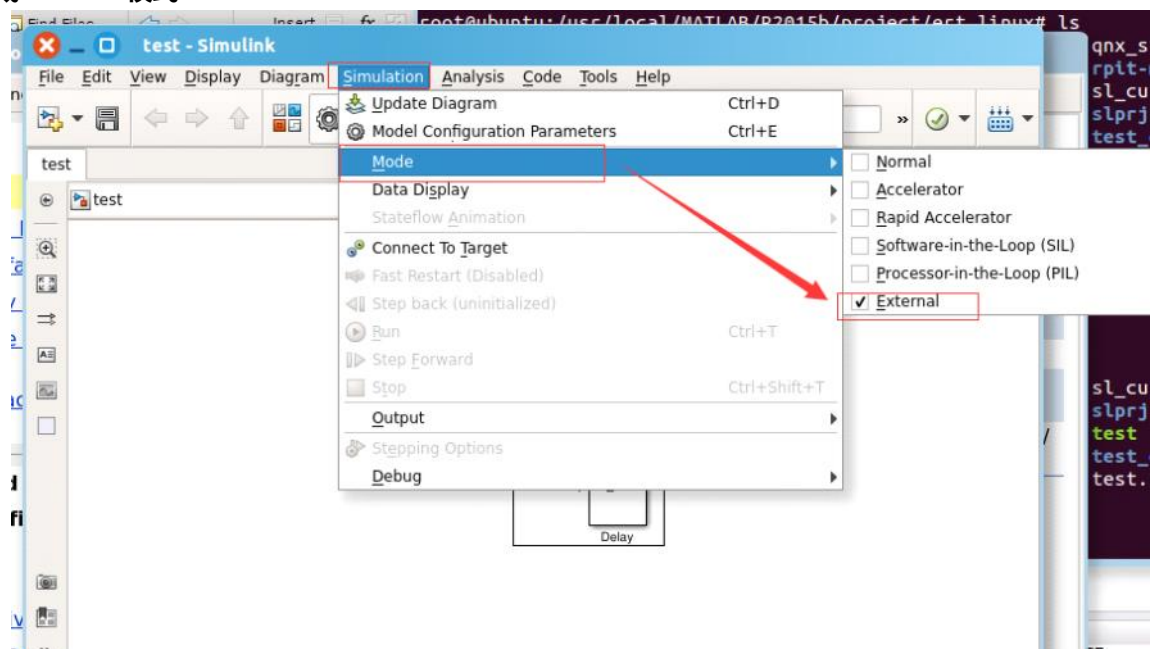
## d. Code Generation -> Interface



屏幕剪辑的捕获时间: 2019-1-4 16:25

## 2. 仿真设置：外部模式

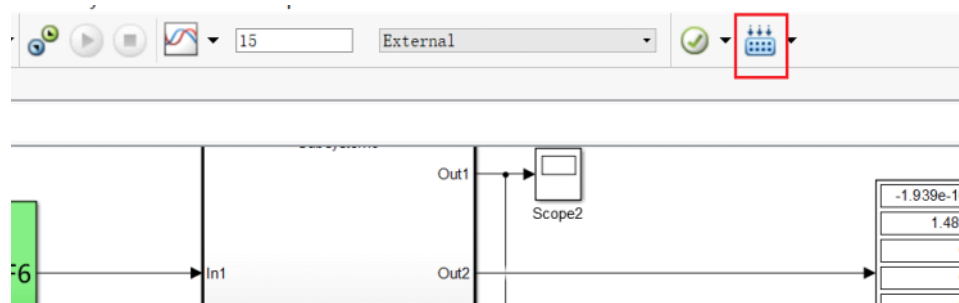
### 设置成External模式



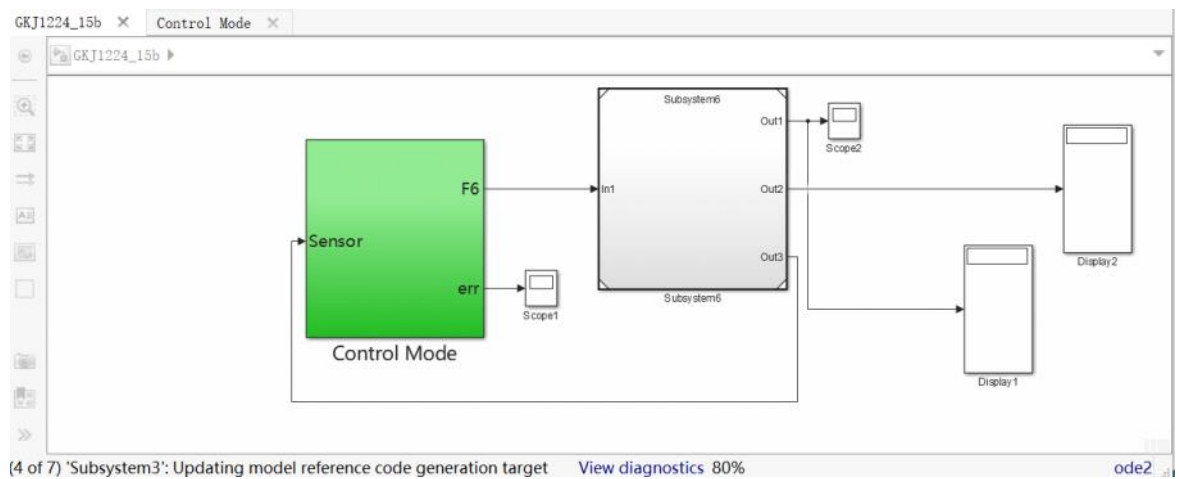
图片来自其他，只为说明操作方法

## 3. 点build生成代码 (快捷键ctrl+B)

**注意：生成代码之前要保存所有设置，否则仿真时会报错模型与目标中不匹配。**



## 4. 点击模型下方View diagnostics 可以查看生成进展；生成失败会自动弹出



屏幕剪辑的捕获时间: 2019-1-4 16:40

```

GKJ1224_15b
### Writing source file GKJ1224_15b_data.c
### Writing header file rtmodel.h
.
### Writing source file ert_main.c
### Writing header file logger.h
### Writing source file logger.c
### TLC code generation complete.
### Generating TLC interface API.
### Creating data type transition file GKJ1224_15b_dt.h
### Creating project marker file: rtw_proj.tmw
### Using toolchain: MinGW64 v4.x | gmake (64-bit Windows)
### Creating
'D:\Developer\MATLAB\Workspace\GKJmodel\GKJ20181225_split6\GKJ1224_15b_linux_ert_rtw\
GKJ1224_15b.mk' ...
### Successful completion of Real-Time Workshop build procedure for model:
GKJ1224_15b
### Adapting code from file GKJ1224_15b.mk
### Adapting code from file GKJ1224_15b.mk

Build process completed successfully

```

屏幕剪辑的捕获时间: 2019-1-4 16:42

如图，生成代码成功

## 5. 可以确认，操作是否使用了正确的生成工具链

注意：当前路径下有同名tmf文件，系统默认使用第一个工具链；

```

.
### Processing Template Makefile:
C:\Users\liziy\Documents\MATLAB\linux_ert_target\linux_ert_target.tmf
### Creating GKJ1224_15b.mk from
C:\Users\liziy\Documents\MATLAB\linux_ert_target\linux_ert_target.tmf
### Successful completion of Real-Time Workshop build procedure for model:
GKJ1224_15b
### Adapting code from file GKJ1224_15b.mk
### Adapting code from file GKJ1224_15b.mk

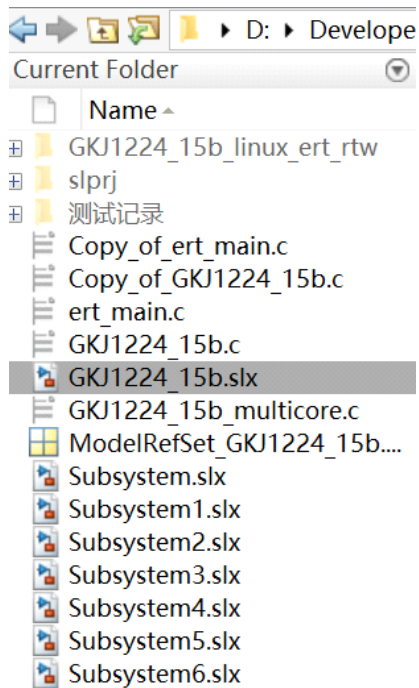
Build process completed successfully

```

屏幕剪辑的捕获时间: 2019-1-5 9:34

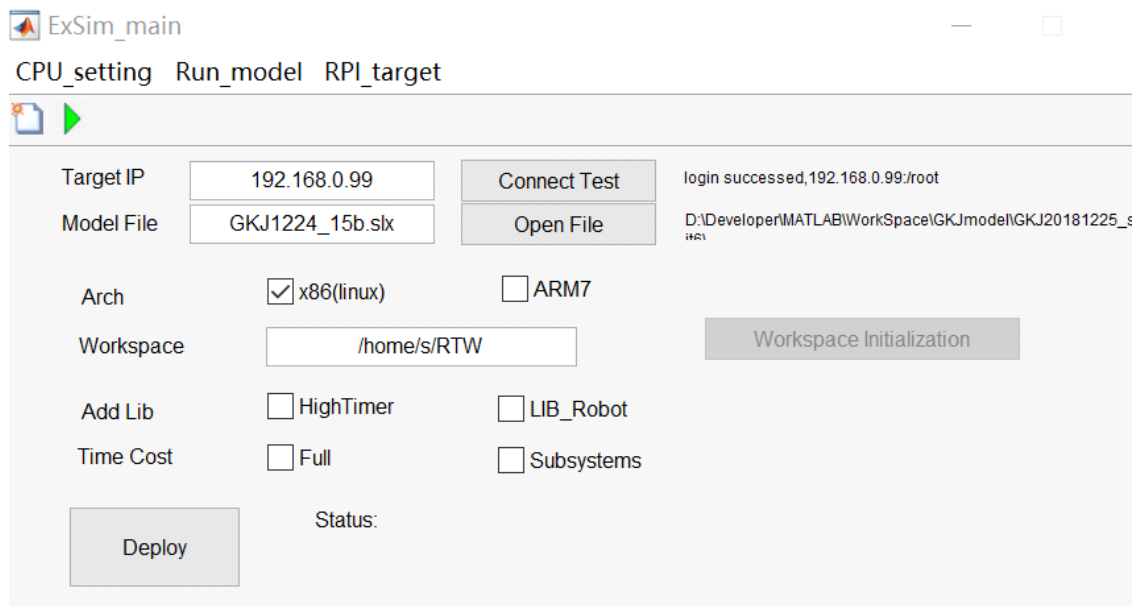
## 模型仿真：以GKJ1224\_15b.slx

### 1. 确定matlab当前路径在模型路径下



屏幕剪辑的捕获时间: 2019-1-4 16:44

## 2. APP打开该模型



屏幕剪辑的捕获时间: 2019-1-4 16:45

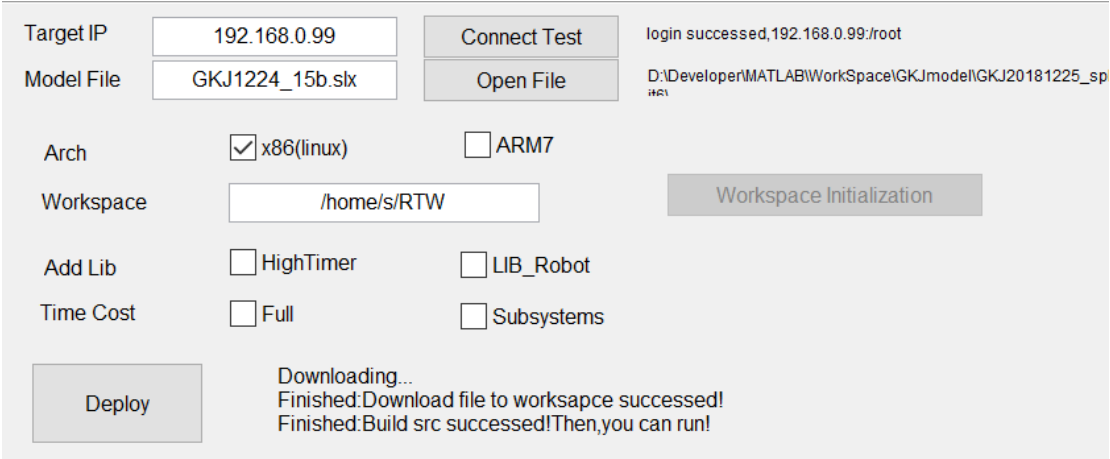
如果有子系统，会自动显示分核设置信息，如果，为分核，Mask 均为Null

	Name	Mask
1	Subsystem	4
2	Subsystem1	1
3	Subsystem2	3
4	Subsystem3	2
5	Subsystem4	1
6	Subsystem5	2
7	Subsystem6	4

屏幕剪辑的捕获时间: 2019-1-4 16:46

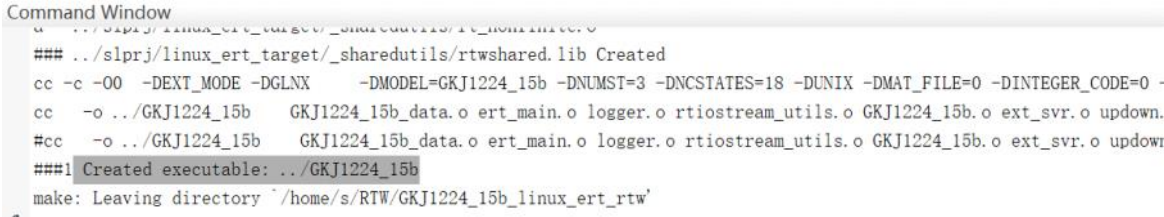
3. Deploy部署到目标机

编译成功



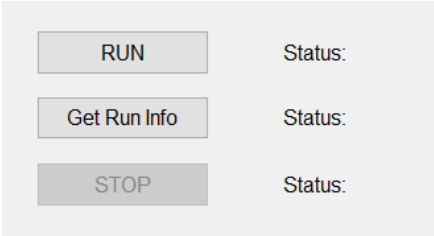
屏幕剪辑的捕获时间: 2019-1-5 9:40

命令行详细信息



屏幕剪辑的捕获时间: 2019-1-5 9:41

此时，RUN 键使能



屏幕剪辑的捕获时间: 2019-1-5 9:55

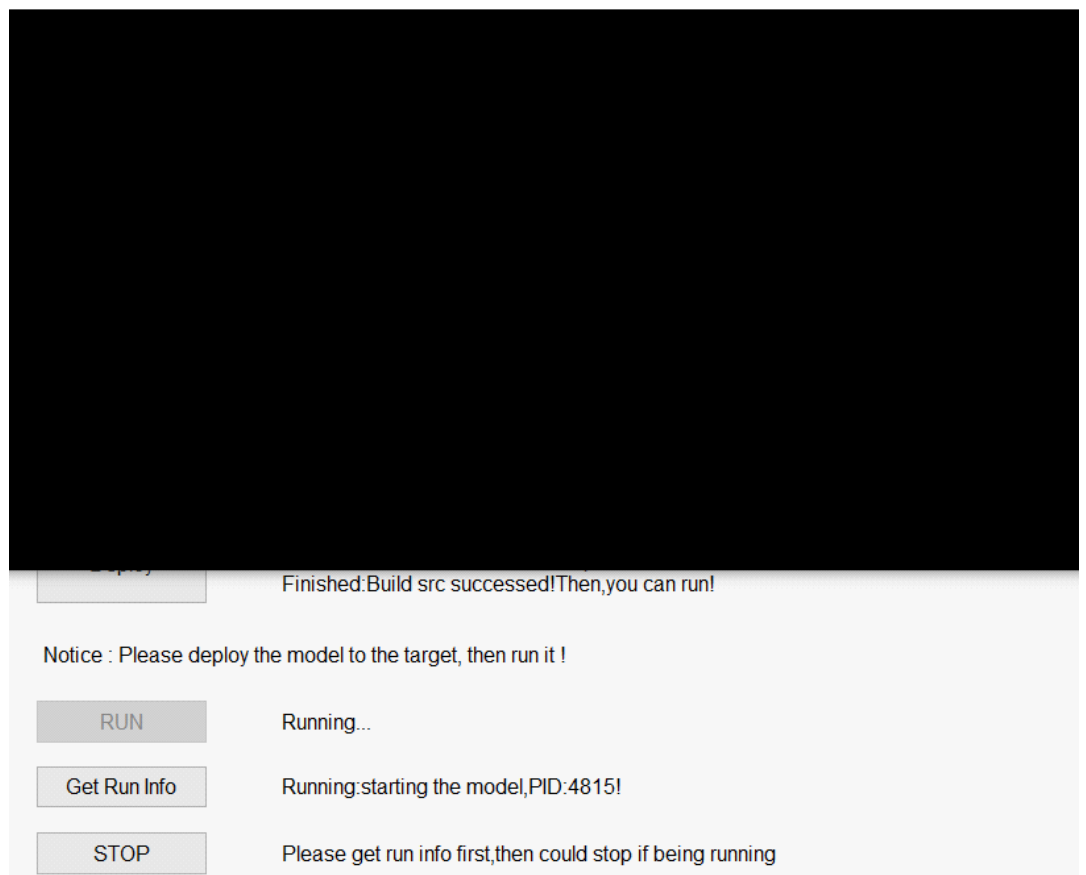
4. 目标机程序运行

点RUN，弹出DOS窗口

如果不确定，可以点Get Run Info 获取PID(进程号)

如图，目标机程序的ID = 4815；


```
V132\cmd.exe - plink -pw root root@192.168.0.99 "cd /home/s/RTW && ./GKJ1224_15b -w"
```

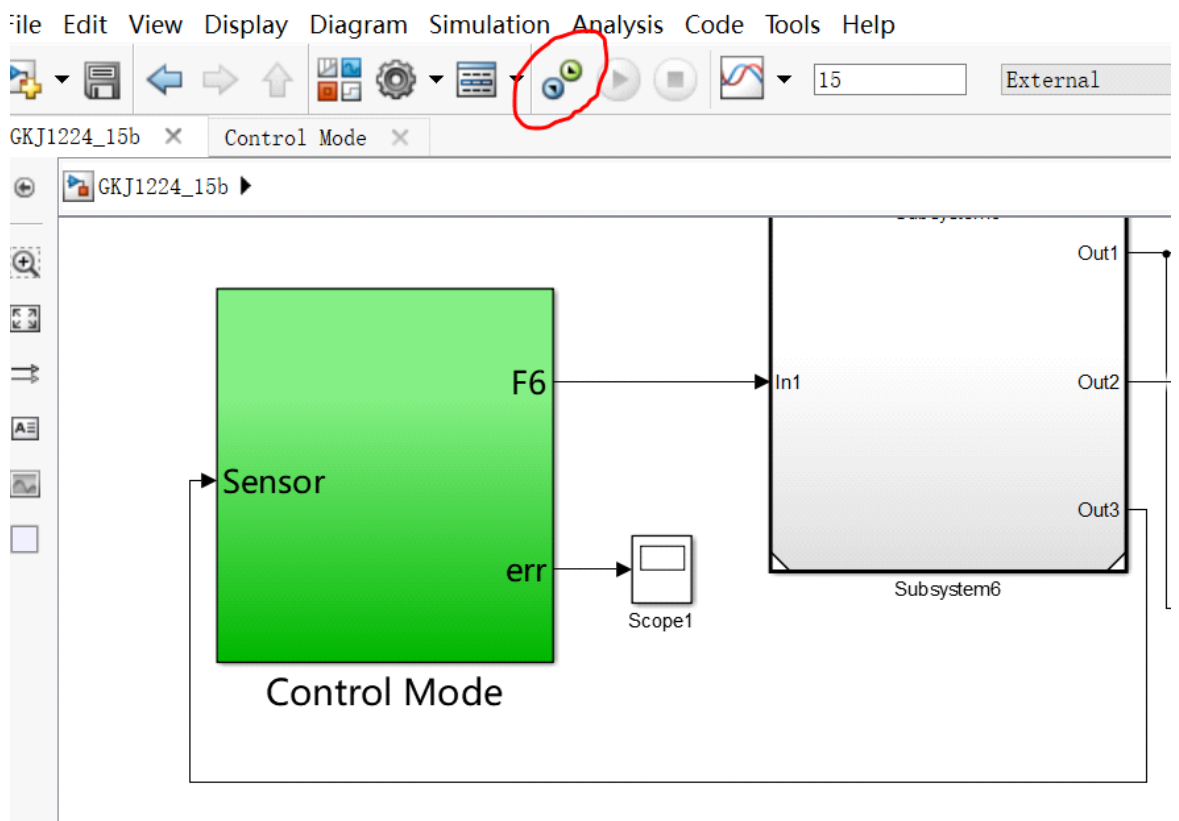


屏幕剪辑的捕获时间: 2019-1-5 9:58

## 5. 模型联接目标机

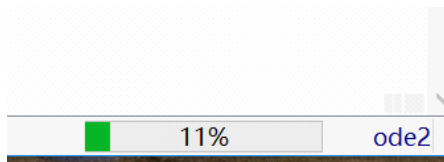
点击Connect to target,

 GKJ1224\_15b - Simulink

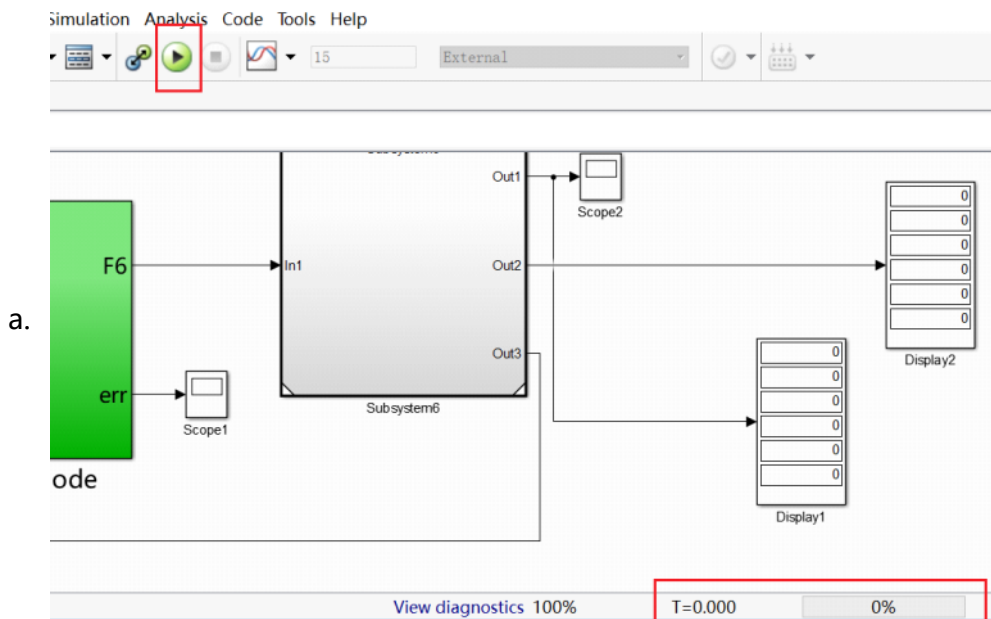


右下角看连接操作进度.





连接成功状态：Run键由灰变亮，右下角T=0,0%等待开始；

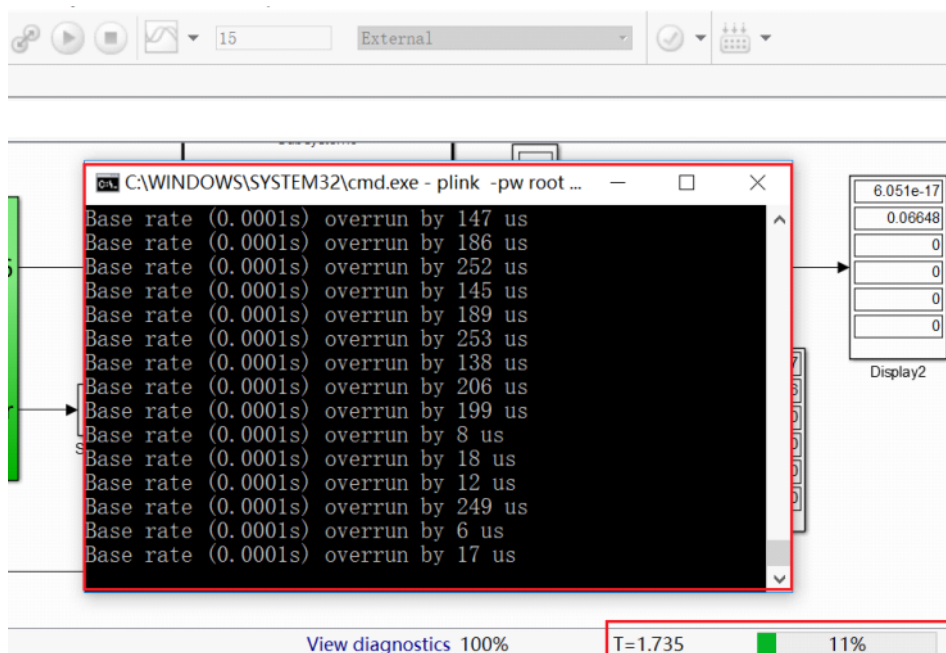


## 6. RUN开始仿真

由于目标机程序执行./ModelName -w,是等待模式，所以在模型连接目标机成功后再点击RUN，模型才开是仿真。

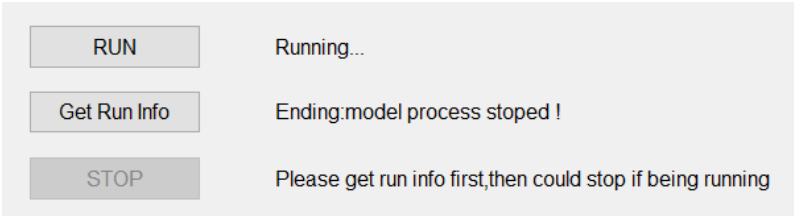
点RUN，可以看到右下角的时间，和进度条 显示仿真进度；

DOS窗口显示目标机程序执行打印信息，默认为周期执行超出时间(给个周期超出才打印)



## 7. 由于系统设置的时间15s,所以仿真自行结束；

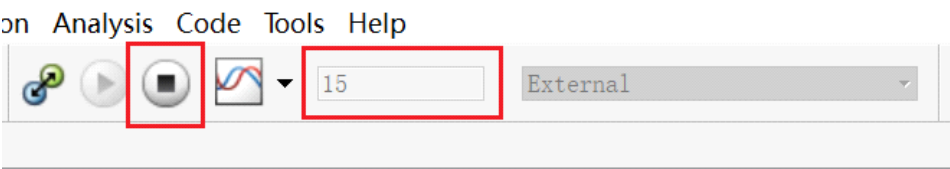
点Get Run Info 可知目标机程序以结束；此时可点RUN，重复步骤4~6进行仿真；



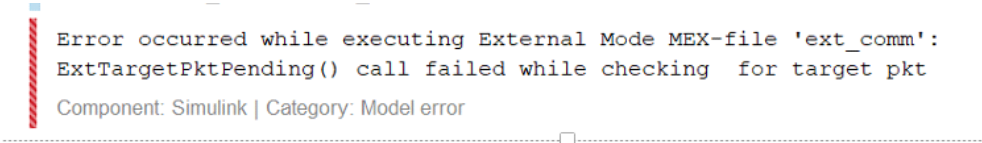
屏幕剪辑的捕获时间: 2019-1-5 10:18

注意：这里设的仿真时间，只是按所设solver情况下的时间；如果发生超时，仿真时间并不等于15s,即并不表示实际执行实际。

8. 执行过程中，或时间设置了inf,可点击Stop停止仿真



由于目标机程序为“-w”等待模式，再模型中点Stop,目标机程序即结束退出；  
或在先APP中点STOP，模型端会错误退出；



## 分核设置

1. 打开模型后，显示子系统信息

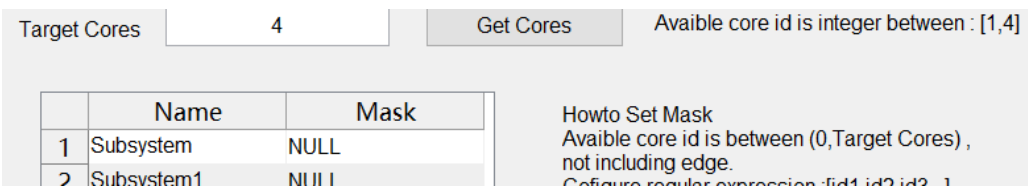
	Name	Mask
1	Subsystem	NULL
2	Subsystem1	NULL
3	Subsystem2	NULL
4	Subsystem3	NULL
5	Subsystem4	NULL
6	Subsystem5	NULL
7	Subsystem6	NULL

屏幕剪辑的捕获时间: 2019-1-5 10:49

如上图所示，此模型有7个子系统，由于还未设置分核，所以各子系统对应的掩码值都为NULL，NULL等价于设0；表示不指定核；

2. 获取核数Get Cores

在设置分核之前，必须先获取目标机的可用核数；  
即确定了分核设置的可用范围；



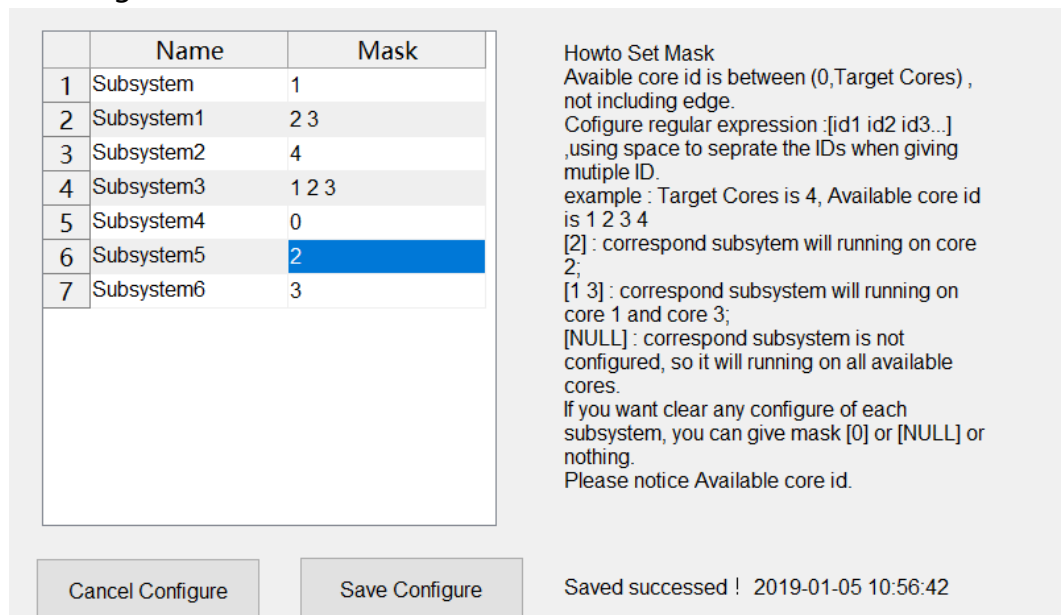
屏幕剪辑的捕获时间: 2019-1-5 10:53

如上图，目标机可用核数为4，则掩码值设置范围为[1,4];可设一个或多个核，如：  
1 或 2 3 ，空格分隔

具体规则可参考 界面中Howto Set Mask

### 3. 保存设置

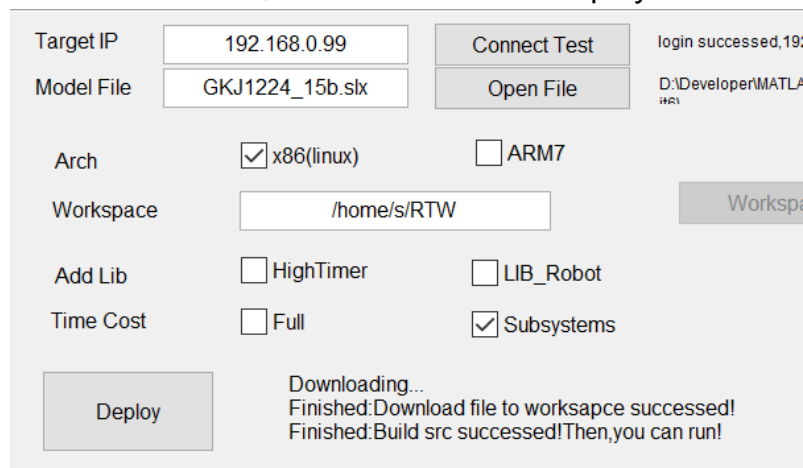
点Save Configure,显示保存成功, 即时间戳;



屏幕剪辑的捕获时间: 2019-1-5 10:57

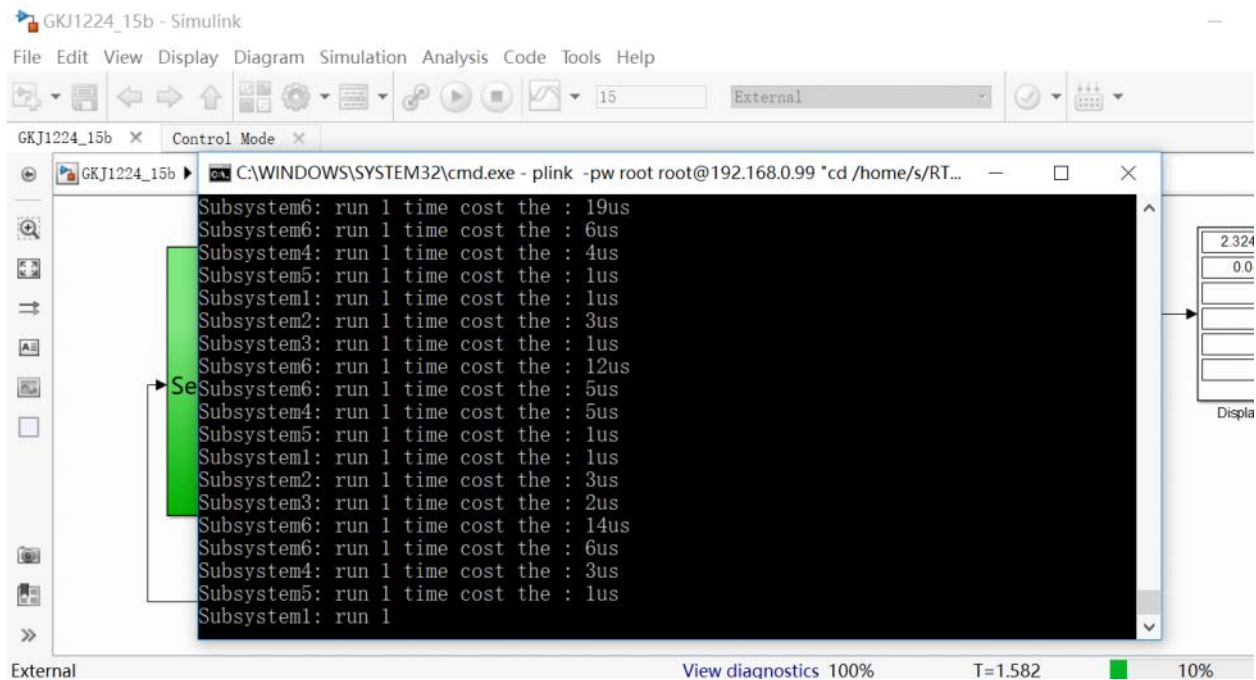
## 测试子系统时间

1. 确认系统含有子系统, 无关分核, 都可以测试;
2. 选择编译选项Time Cost中的substime, 然后点Deploy



屏幕剪辑的捕获时间: 2019-1-5 11:02

3. 执行仿真操作, 结果如下, 输出每个子系执行时间



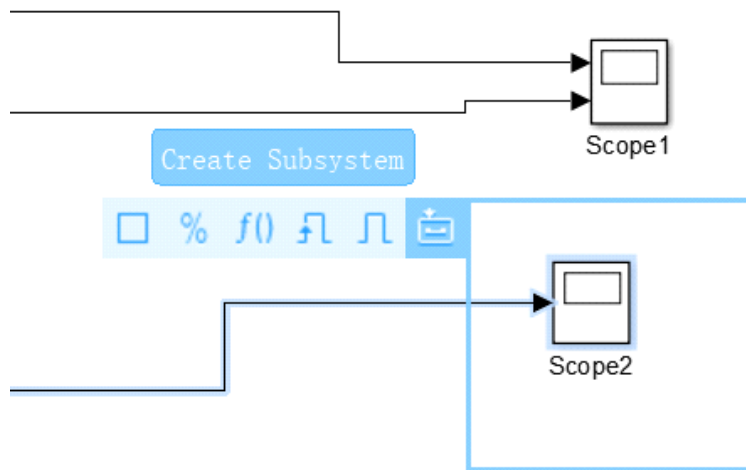
## 可选编译参数

Arch	目标机平台架构	互斥选项
	X86	
	ARM7	
Add Lib	自定义扩展库	可多选，PCIE和GKJ后台自动支持
	HighTimer	福东HT时钟驱动
	Robot	韩冰机器人库
Time Cost	测试执行时间工具	可多选，都不选为默认，输出overrun超时时间
	FullTime	测试周期执行时间
	Substime	测试各个子系统执行时间

## 附：子系统创建

ModelReference一般是在多人协作时，最后集成模型调用各个工程师的模块时使用，可以方便的进行模块修改，然后在主模型中更新舒心即可，而且可以将子模块生成单独的源文件和头文件。

- 1) 在matlab模型界面中选中要生成subsystem的部分，右键->create subsystem from selection,或如下快捷方式



生成一个子系统

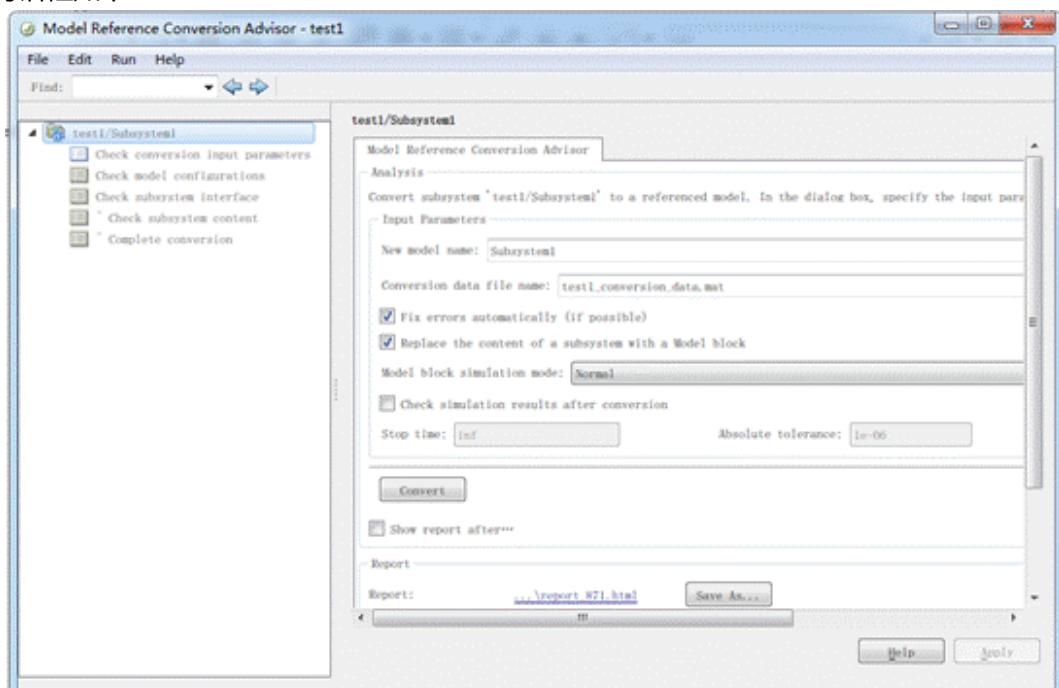


## 2) 转换子系统为ReferencedModel

上图生成的子系统Subsystem, 右键

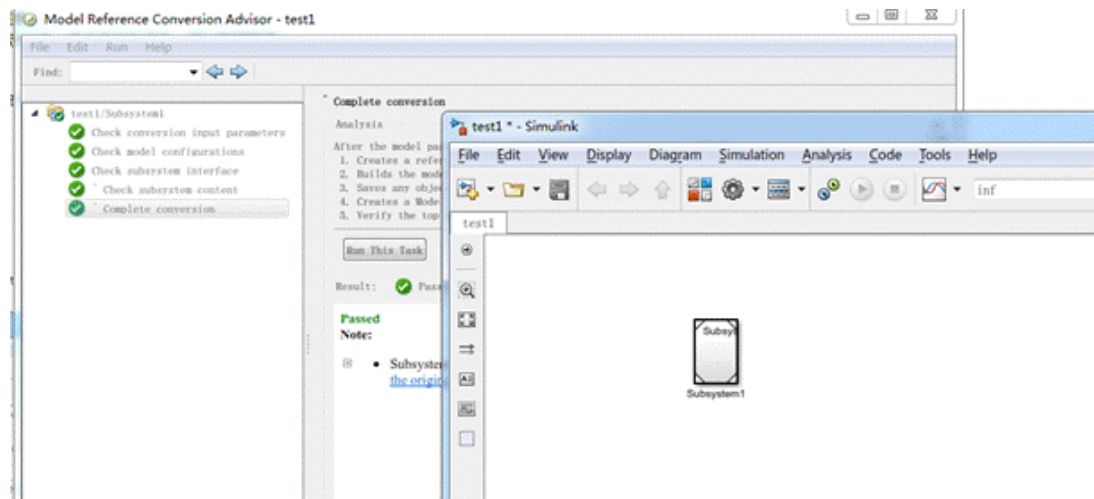
——>Model Reference&Subsystem——>convert subsystem to ——>Referenced Model

弹出对话框如下:



勾选上Fix errors automatically (if possible), 点击Convert, 开始转换,

如果左侧各个步骤到Complete conversion都通过 (绿色背景对勾, 如下图), 做转换成功



完成后，子系统矩形框四角多斜线。方框中定义了实际的子系统模型名，对应slx文件可在本地文件找到。

如果转换过程中出错并停止（对应步骤前显示红色x），可以根据错误提示，修改模型相关参数或设置，然后继续convert，直至转换成功。