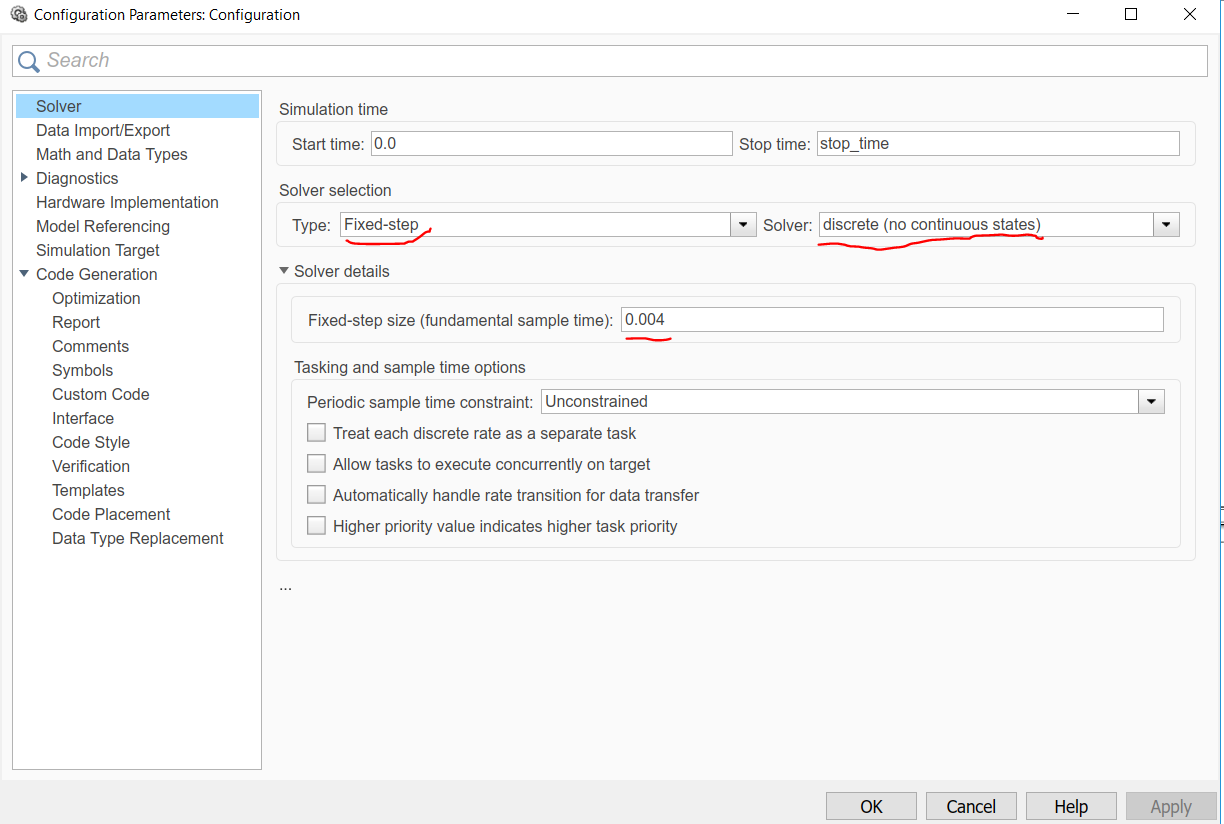
# Simulink模型使用说明

## 模型配置

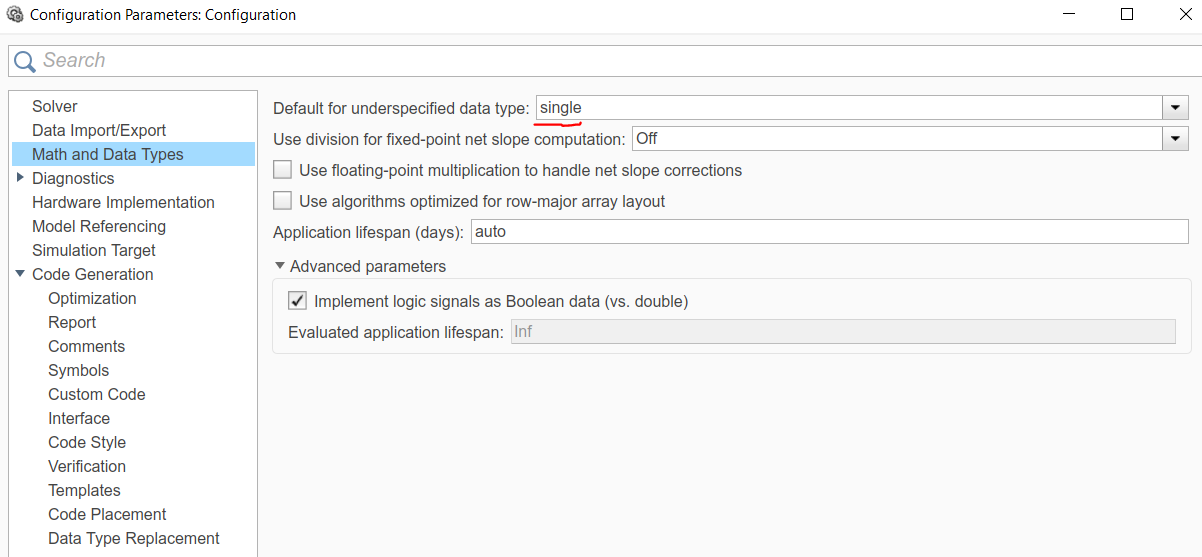
Simulink具有强大的模型仿真能力，同时还支持将仿真模型自动生成C/C++代码的功能，十分方便。当一个算法十分复杂时，手写的算法将变得越来越难维护，这个时候，如果使用Simulink，则会非常便捷和高效。

为了用Simulink模型生成可以在目标硬件上正常运行的代码，我们需要对Simulink模型进行配置。Simulink的仿真非常强大，具有很多复杂的功能，所以模型可配置项也非常多。但是我们只要对其中一些主要配置项进行配置即可，其它配置项一般用默认值就行。下面详细介绍下一个可以直接生成C/C++代码的Simulink模型需要做哪些配置。

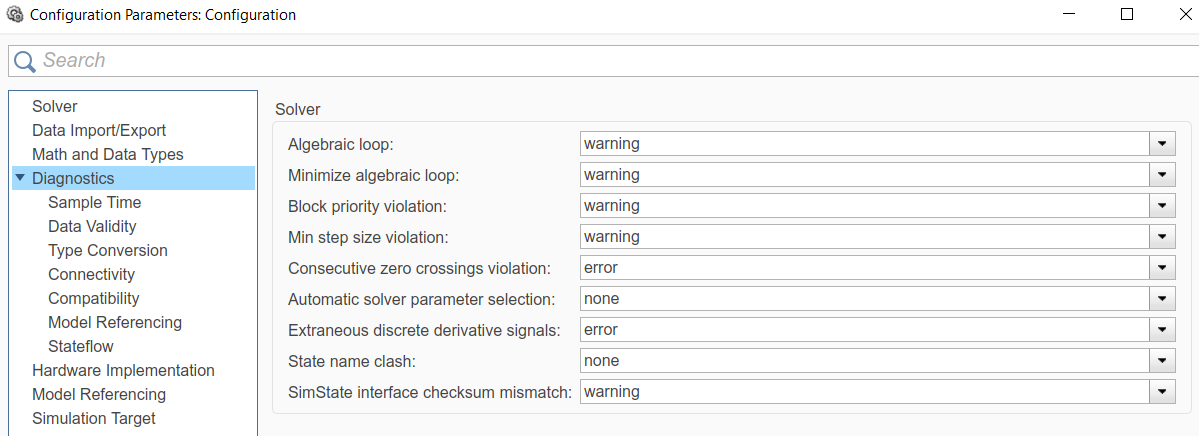
以PID模型为例。打开Model Configuration Parameters，可以看到如下界面。首先对Solver进行配置。画红线的地方是我们需要配置的地方。**Type**配置成Fixed-step，**Solver**配置成Discrete(或者一般ode4)，因为我们实际的系统都是离散系统。**Fixed Step-Size**是系统的执行周期，这里设置成0.004，表示我们的控制器是4ms周期执行。



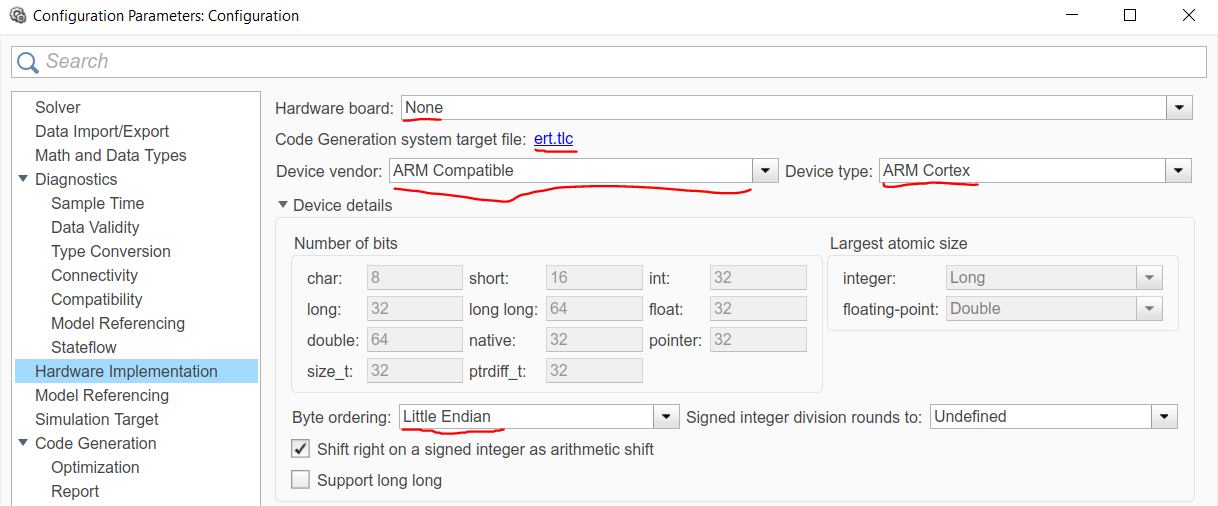
第二个界面是Math and Data Type，这里只需要配置一个选项**Default for underspecified data type**，配置成single，因为我们希望模型默认是使用single作为数据类型。如果追求更高精度，处理器的性能足够的话，也可以用double类型。



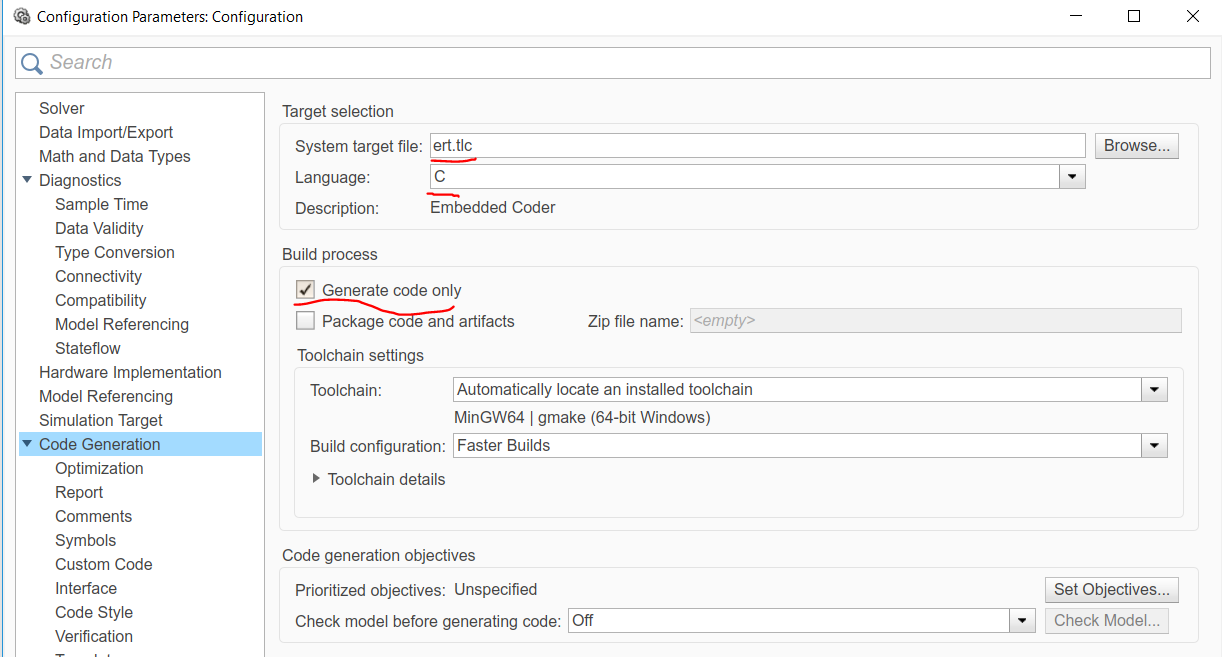
第三个页面是Diagnostics，这个页面主要是配置模型编译的时候，哪些情况是error，哪些情况是warning。一般用默认配置就行。如果想要使得模型变得更加健壮，可以对这里进行一些配置。



Hardware Implementation里面，首先**Hardware board**选择None，然后**Code Generation system target file**选择ert.tlc(Embedded Coder)，表示生成嵌入式代码。也可以选择其他的system target file来生成比如桌面程序代码。然后**Device vender**选择ARM Compatible，**Device type**选择ARM Cortex，这里也可以根据实际需要，选择不同的芯片类型。**Byte ordering**选择默认小端即可。



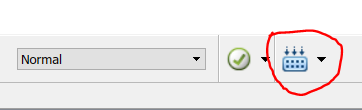
Code Generation设置代码生成相关的配置项。首先**System target file**选择ert.tlc，表示使用Embedded Coder生成代码。**Language**选择C(或者C++)。**Generate Code Only**勾上，这样就只会生成代码，会少生成一些不相关的文件。



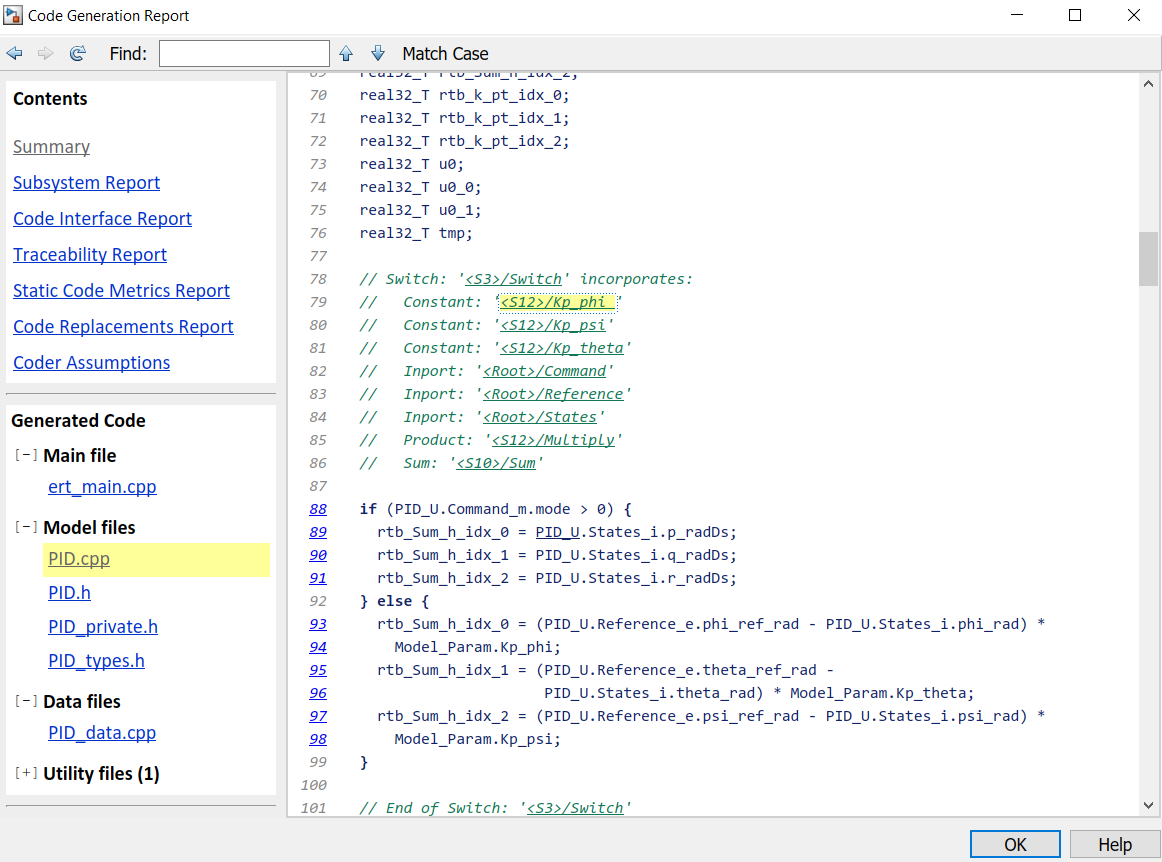
在Code Generation下面还有一些小的选项，一般都是默认配置就好。比如Optimization，可以设置生成代码的优化等级。其它的选项可以根据需要设置。

## 代码生成

模型配置好后，点击**Build Model**编译生成代码。



代码编译完成后，会生成一个报告，如下所示。里面列出了所有生成的代码文件，点击文件能够浏览代码。点击对应的注释段还能跳转到这段代码对应模型的模块。源文件在Build目录下的PID\_ert\_rtw和slpj目录下。



## 代码合入

## 日志仿真

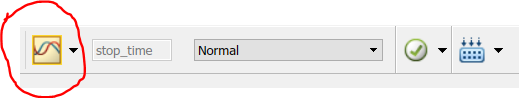
日志仿真是Simulink模型一个王牌级别的功能，可以做到一次飞行，多次仿真的目的。

为了获得Simulink可以用的日志数据，我们首先调用log\_parse.m文件将px4解析的日志文件变为Simulink可以识别的timeseirals数据。timeserials数据可以用Data Inspector进行查看。

有了数据以后，打开Debug.slx模型。将stop\_time赋值为reference.phi\_ref\_rad.Time(end),这样就将仿真的时间跟日志数据的时间设为一致，点击运行即可。

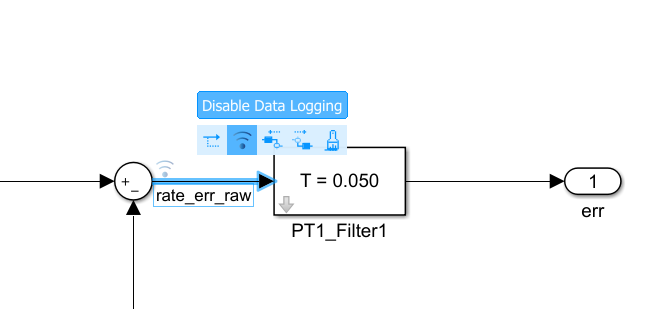
## 查看数据

Simulink支持多种数据查看方式，常用的如Data Inspector，如下图所示，点击图标





左边是所有记录的数据，勾选以后可以在右边显示出来。若要对某个信号进行Log,只需单击对应的信号线，选择Start Logging，然后会在信号线上看到一个信号波的标志，表示该信号会被记录下来。信号名称默认跟信号线上的名称一致，也可以人为指定名称。



第二种方式是添加Scope查看信号。如图所示，如果要查看err信号，添加一个scope既可。

