## EGR-BLDC Emulator代码ACG后的处理

由于EGR-Emulator使用了S-Function，里面涉及了一些基于浮点的函数，如rem函数等。而模拟器执行环境一般不支持浮点数操作，因此在经过Matlab ACG代码生成后，需要手动处理下生成的EGR模型代码。

涉及的代码文件以及相应如下：

1. Power\_BLDC\_EGR\_private.h

将extern void remf\_Outputs\_wrapper(const real\_T\* theta, Boolean\_T \*Hallx, in8\_T \*phase中的real\_T改为***int32\_T***.

1. Power\_BLDC\_EGR\_data.c

将const ConstP\_power\_BLDC\_EGR\_T power\_BLDC\_EGR\_ConstP的值由浮点数6.28转换为LSB=1024后的值，即改为6434.

1. Power\_BLDC\_EGR.h
2. 注释掉行int32\_T DataTypeConversion; /\* '<S14>/Data Type Conversion' \*/
3. 将行real\_T DataTypeConversion1中的real\_T改为int32\_T
4. 将行real\_T rad2deg1的real\_T改为int32\_T
5. 将行real\_T pooled1中的real\_T改为int32\_T
6. Power\_BLDC\_EGR.c

在power\_BLDC\_EGR\_rem函数体中，

1. 注释掉行localDW->DataTypeConversion = (real\_T)rtu\_the…
2. 将remf\_outputs\_wrapper(&localDW->xxx, …)改为remf\_outputs\_wrapper(**&rtu\_the**, …)
3. 将 localDW->DataTypeConversion1 = (int32\_T)(localDW->SFunctionBuilder \* 1024.0);中的\*1024去掉

在power\_BLDC\_EGR\_step函数体中：

1. 在函数体中搜索power\_BLDC\_EGR\_rem，power\_BLDC\_EGR\_DW.rad2deg1= (real\_T)power\_BLDC\_EGR\_DW.rem.DataTypeConversion1\*0.0009765625\* 57.2957795;改为 ***power\_BLDC\_EGR\_DW.rad2deg1 = power\_BLDC\_EGR\_DW.rem.SFunctionBuilder \* 58671 >> 10;***

最后全局搜索下real\_T看看是否还有遗漏之处。

PS:如果模型基本没有改动，可以借助diff工具来进行修改。