
Milestone

发布 1.0

MA Yuhai

2020 年 03 月 15 日

1	主要功能特性	1
2	安装与配置	3
2.1	软件授权	3
2.2	软件安装	3
2.3	外部依赖环境配置	3
3	案例教程	5
3.1	运行案例	5
3.2	代码结构剖析	5
4	Milestone 的图形用户界面操作方式	9
4.1	新建模型并生成模板	9
4.2	打开已有的模型头文件	9
4.3	模型模板的代码结构及资源接口	9
5	Milestone 的命令行操作方式	11
5.1	添加模型	11
6	FMI 标准模型接口及其实现	13
7	S-Function 模型接口及其实现	15
8	工具和索引	17

主要功能特性

1. 用于模型在环仿真（MiL）的 C/C++ 代码集成工具
2. 简捷、统一，一次编码同时支持 FMI 及 S-Function 模型接口标准
3. 使用 CMake 构建系统，可自动适配大多数编译器
4. 使用 Qt 图形界面，可运行于大部分操作系统的桌面环境
5. 与来自 IBK 的 MasterSimulator 环境集成发布，便于 FMU 集成仿真
6. 生成平台相关的代码，兼容 Windows 及 Linux
7. 生成的模型可运行在基于 Linux 的半实物仿真（HiL）系统

2.1 软件授权

提供运行系统的 MAC 地址，获取授权文件

2.1.1 Windows

```
ipconfig /all
```

2.1.2 Linux

```
ifconfig
```

2.2 软件安装

工具包解压缩得到主目录结构将授权文件放置在 license 目录中

2.3 外部依赖环境配置

2.3.1 编译器

Windows

Visual Studio 中的 cl 编译器，注意不要使用绿色安装。示例代码在 VS2010 及以上版本中经过测试，但推荐使用 VS2013 及以上的版本，以支持 C99 中的编码习惯。Linux ^^^^ gcc/g++ 或 clang/clang++ 编译器，推荐在系统的包管理器中安装。

2.3.2 CMake

开发工具包执行需要部署 cmake 运行环境，env 中包含相应的安装文件。此外 Windows 下还要部署 VC 运行时环境，Linux 下需要授予 bin 目录下程序执行权限（`chmod 777 ./bin/*`）。

2.3.3 路径配置

在 MasterSim 中选择当前系统中安装的 CMake 路径以及 milestone 可执行文件的路径。

2.3.4 切换界面语言

在 MasterSim 中切换 Milestone 的界面语言，重新启动后生效。

3.1 运行案例

在工具包根目录下执行” mkdir build” //建立单独的构建目录，名称任意，用于将临时文件与工具分开在工具包根目录下执行” cd build” //切换到创建的构建目录在创建的构建目录下执行” cmake ..” //在构建目录下，指定代码目录在上层目录，生成编译工程文件（Windows 下为 MSVC sln，Linux 下为 Makefile）在创建的构建目录下执行” cmake -build .” //执行构建，注意” .” 为当前目录，附加” -config Release” 或” -config Debug” 参数切换 Release 版和 Debug 版，默认为 Debug 版所导出的 fmu 模型在 export 目录中依次在不同系统下执行工具包，将 model 目录（保留已生成的中间文件）或整个工具包复制到其他系统继续构建，将获得同时支持多系统的 fmu 文件

3.2 代码结构剖析

```
1  #ifndef INTERFACE_H__
2  #define INTERFACE_H__
3  //=====
4  // A test case for fmi simulation tools
5  // Copyright (c) 2019 马玉海
6  // All rights reserved.
7  //
8  // Version 1.0
9  //=====
10 #define _CRT_SECURE_NO_WARNINGS
11 #include <math.h>
12 #include <stdio.h>
13 #include <stdlib.h>
14 #include <memory.h>
15 #include <string.h>
16 #include <float.h>
17
18 #define IO_PORT_FLUSH(data_type, var_name) \
```

(下页继续)

(续上页)

```

19     do{\
20         memset(&(p->var_name), 0, sizeof(data_type));\
21     } while(0);
22
23 #ifndef __cplusplus
24 #define FMI_EXPORT
25 #define bool unsigned char
26 #define true 1
27 #define false 0
28 #else
29 #define FMI_EXPORT extern "C"
30 #endif
31
32 #ifndef _WIN32
33 #include <limits.h>
34 #define _MAX_PATH PATH_MAX
35 #define _MAX_FNAME NAME_MAX
36 #define _MAX_EXT NAME_MAX
37 #endif
38 // non-standard interface definition
39 #define FMI_IN
40 #define FMI_OUT
41 #define FMI_PRM
42 typedef const char * fmi_str_ptr;
43
44 FMI_EXPORT void *fmi_instantiate(void);
45 FMI_EXPORT int fmi_initialize(void *);
46 FMI_EXPORT int fmi_doStep(void *);
47 FMI_EXPORT int fmi_reset(void *);
48 FMI_EXPORT void fmi_freeInstance(void *);
49
50 #pragma pack(push, 8)
51 typedef struct _Stru_Data_Controller_To_Plant_ex
52 {
53     double y;
54     double z;
55 }Stru_Data_Controller_To_Plant_ex;
56
57 typedef struct _Stru_Data_Controller_To_Plant_ex1{
58     double y;
59     double z;
60 }Stru_Data_Controller_To_Plant_ex1;
61 typedef struct _Stru_Data_Controller_To_Plant_ex2 {
62     double y;
63     double z;
64 }Stru_Data_Controller_To_Plant_ex2;
65 typedef struct _Stru_Data_Controller_To_Plant_ex3 {
66     double y;
67     double z;
68 }Stru_Data_Controller_To_Plant_ex3;
69
70 typedef struct _Stru_Data_Controller_To_Plant{
71     double F;
72     double x_0;
73     double v_0;
74 }Stru_Data_Controller_To_Plant;

```

(下页继续)

(续上页)

```
75
76 typedef struct _Stru_Data_Plant_To_Controller
77 {
78     double x;
79     double v;
80 }Stru_Data_Plant_To_Controller;
81
82 #pragma pack(pop)
83
84 #endif // INTERFACE_H__
```

Milestone 的图形用户界面操作方式

4.1 新建模型并生成模板

4.2 打开已有的模型头文件

4.3 模型模板的代码结构及资源接口

Milestone 的命令行操作方式

5.1 添加模型

复制 model 内部的模型目录结构（内部 sources 文件夹为必须），实现与模型文件夹同名称的.h 及.cpp 模型代码文件若增加新的模型间接口，在 model/interface.h 中定义接口数据结构体在顶层 CMakeLists.txt 中” foreach (MODEL_NAME controller plant plant_1) # add model to this list” 语句处，将新的模型添加在列表中重新执行上述构建操作，系统将执行增量构建

CHAPTER 6

FMI 标准模型接口及其实现

S-Function 模型接口及其实现

生成 fmu 的过程中，在模型的 sources 目录中也生成了支持 Simulink 导入的 S-Function 接口代码若要生成模型的 S-Function 模块，在顶层 SFcnLists.m 中” model_list = { ‘controller’ , ‘plant’ }; % add model to this list” 语句处，将新的模型添加在列表中启动 MATLAB，将工作路径切换至工具包根目录，运行 SFcnLists.m 脚本，将执行 S-Function 的代码生成和模块构建保存获得的 Simulink 模型模块，以及工作空间中的数据总线定义，分发模型时还需要附加 *.mexw32/*.mexw64 二进制文件，以及模型所需的数据文件

CHAPTER 8

工具和索引

- `genindex`
- `modindex`
- `search`