

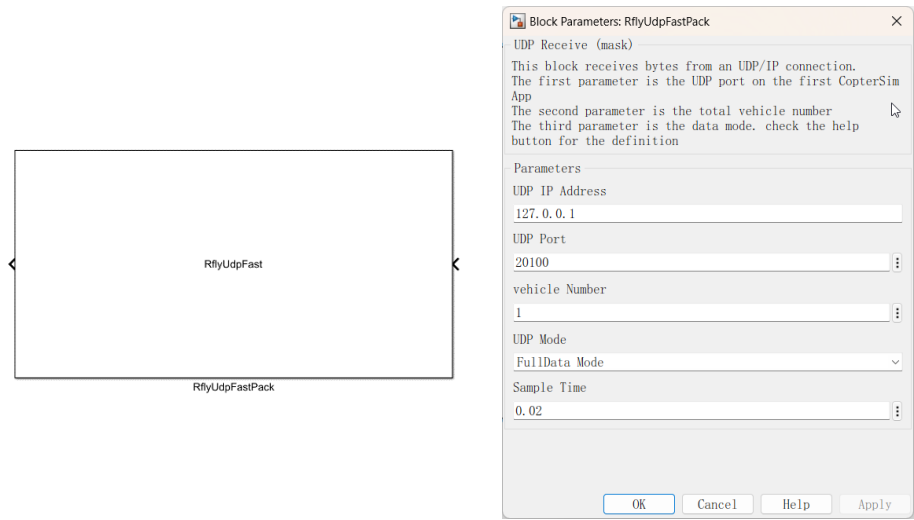
# 1、实验名称及目的

**MATLAB 集群接口模型封装实验：**在 MATLAB 的 C++ S 函数通信模块具有效率高：  
C/C++语言直接实现，比 m 语言高效。**运算小：**Simulink 自带 UDP 模块输入输出都是高维（几百维）的 unit8 向量，在飞机增多时，整个项目维度急剧扩张；而 S 函数方式直接输出 double 型的位置、速度等数据，维度低（几维），运算量小。**延迟低：**Simulink 自带的 UDP 模块为了防止数据丢失，会将收到数据全都保存在缓存中，依次调用，这样当外部程序发送频率大于 Simulink 的运行频率是，就会产生大的延迟；而编写 S 函数的方法则能避免本问题。**更可靠：**Simulink 自带的 UDP 模块每个仿真步长读取一个数据，如果外部程序发送频率小于 Simulink 运行频率，则 Simulink 没有读到数据，会输出 0，导致运算出错；而自己编写 S 函数则可以避免本问题。**扩展性强：**S 函数可以轻易扩展为串口、TCP、共享内存、MAVLink 等其他通信协议

RflySim 平台的 MATLAB 集群接口模块采用 C++/S-functions 混编的形式，可以轻松实现 Simulink 控制其他系统。本实验将讲解如何将编写完成的 C++文件如何封装成 Simulink 模块。

# 2、实验效果

封装完成的 Simulink 模块如下：



# 3、文件目录

文件夹/文件名称	说明
RflyUdpFast.cpp	使用 C++编写的 Simulink 的 S 函数源文件。

# 4、运行环境

(所用软件版本须注明)

序号	软件要求	硬件要求	
		名称	数量

1	Windows 10 及以上版本	笔记本/台式电脑 <sup>①</sup>	1
2	RflySim 平台最新版		

①：推荐配置请见：<https://doc.rflysim.com/1.1InstallMethod.html>

## 5、实验步骤

### Step 1:

请学习本平台的 PPT 资料的“**第 04 讲-载具运动建模与仿真.pdf**”的第 1.4~1.5 小节内容配置 MATLAB，使其满足 MEX 编译环境。部署完成后，在 MATLAB 命令行输入 `mex -setup` 将出现如下显示：



```

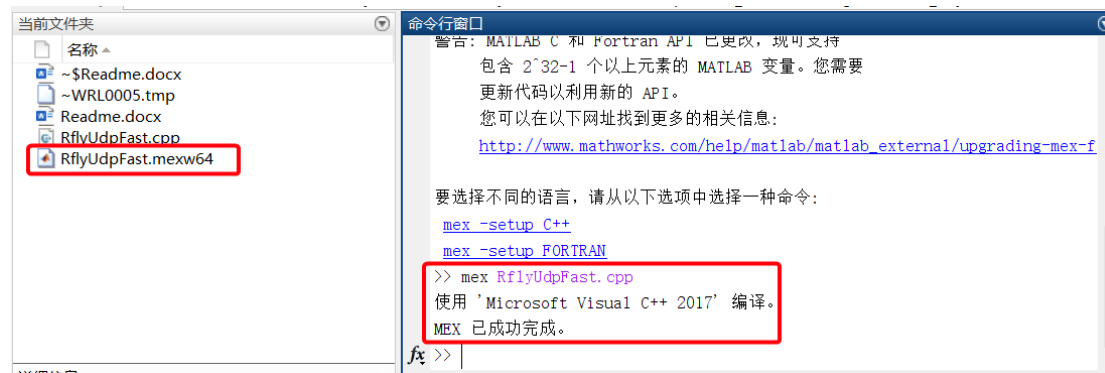
>> mex -setup
MEX 配置为使用 'Microsoft Visual C++ 2017 (C)' 以进行 C 语言编译。
警告: MATLAB C 和 Fortran API 已更改, 现可支持
包含 2^32-1 个以上元素的 MATLAB 变量。您需要
更新代码以利用新的 API。
您可以在以下网址找到更多的相关信息:
http://www.mathworks.com/help/matlab/matlab\_external/upgrading-mex-files-to-use-64-bit-api.ht

要选择不同的语言, 请从以下选项中选择一种命令:
mex -setup C++
mex -setup FORTRAN
fx >>

```

### Step 2:

设置 MATLAB 工作目录位于 RflyUdpFast.cpp 文件目录下，在命令行 `mex RflyUdpFast.cpp`，即可生成 `mex RflyUdpFast.mexw64` 文件。



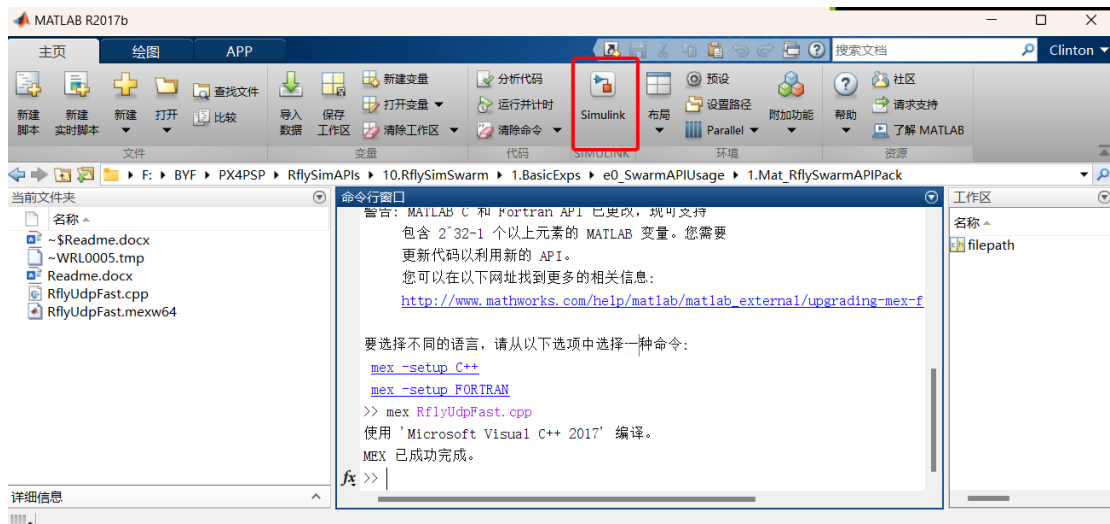
```

>> mex RflyUdpFast.cpp
使用 'Microsoft Visual C++ 2017' 编译。
MEX 已成功完成。
fx >>

```

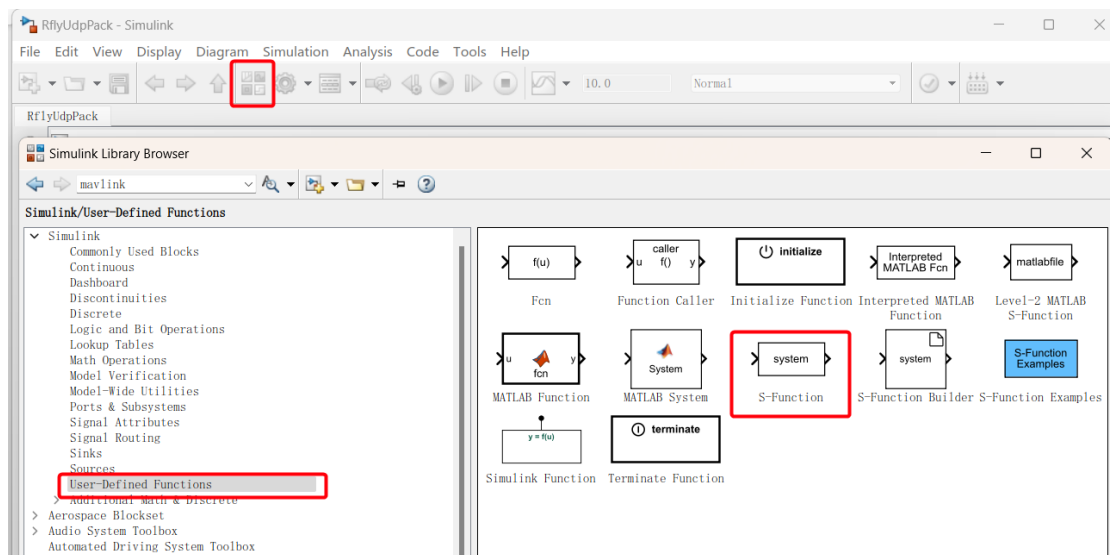
### Step 3:

打开 Simulink，新建保存文件名为：RflyUdpPack.slx。



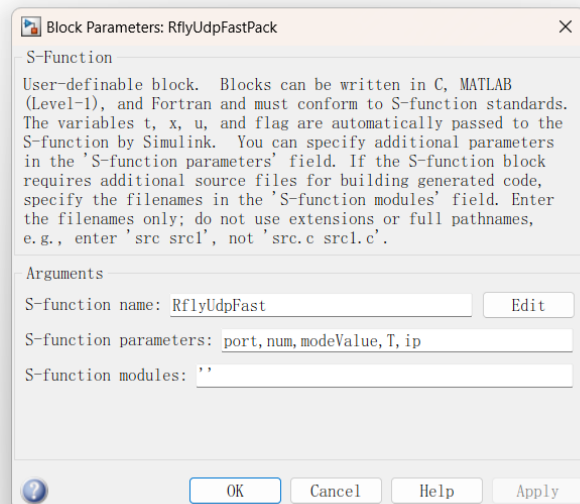
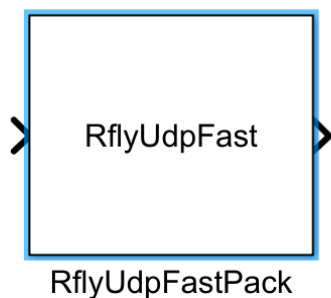
## Step 4:

在 Simulink 的 Library Browser->User-Defined Function 中选择“S-Function”模块，拖入新建的文件 RflyUdpPack.slx 中。



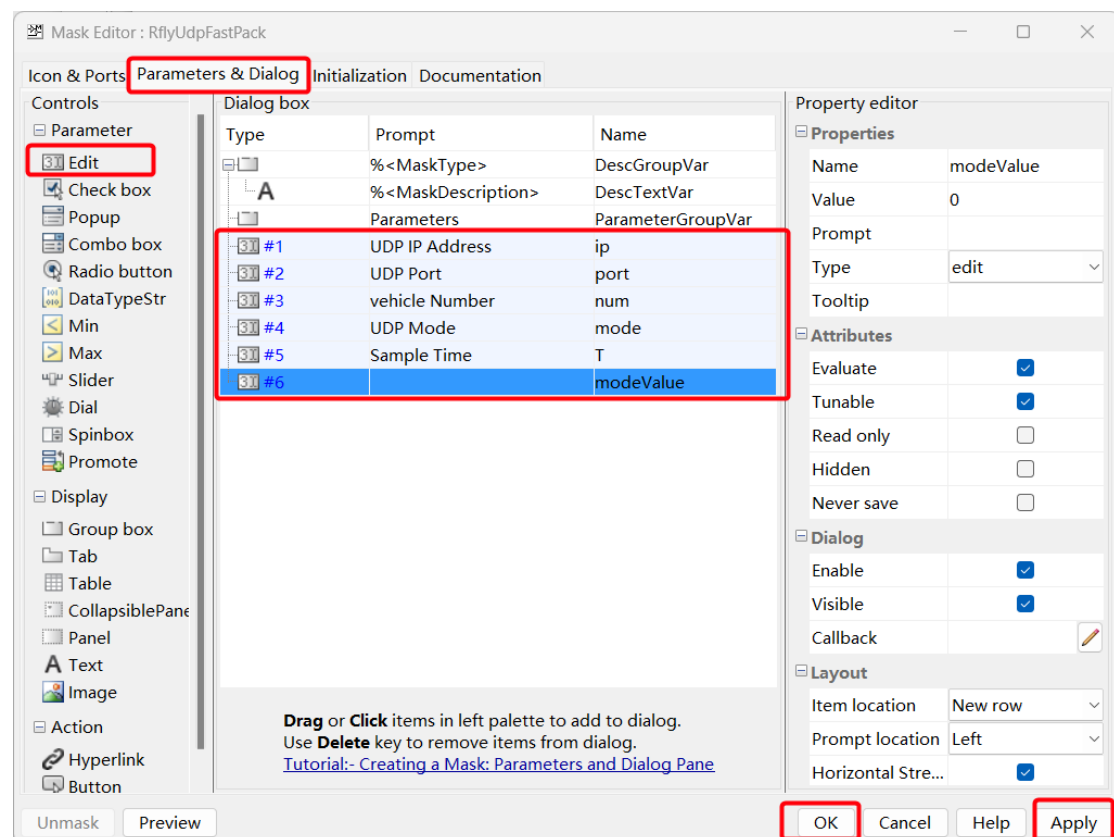
## Step 5:

重命名该模块为 RflyUdpFastPack，并双击该模块打开后，按照下图输入参数：



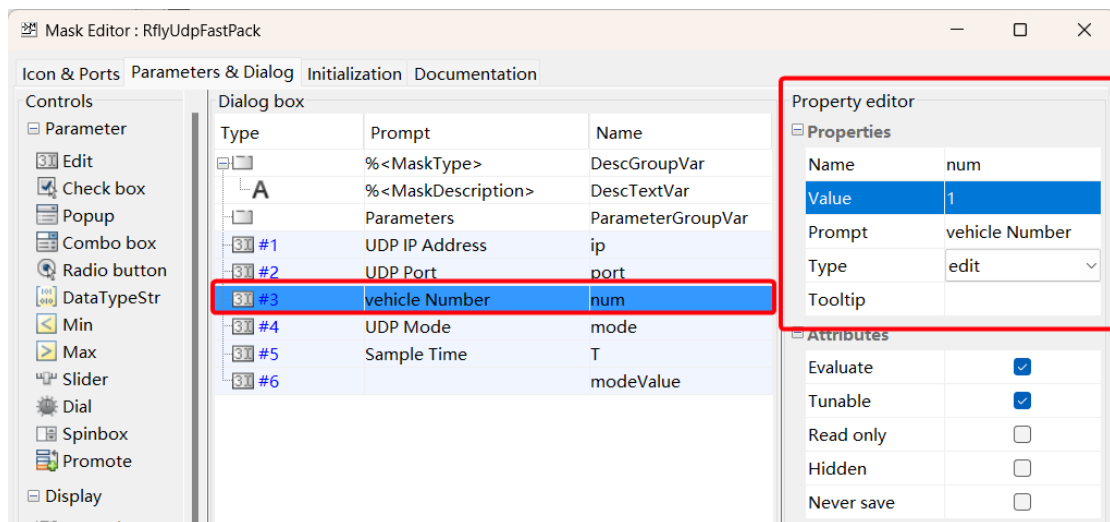
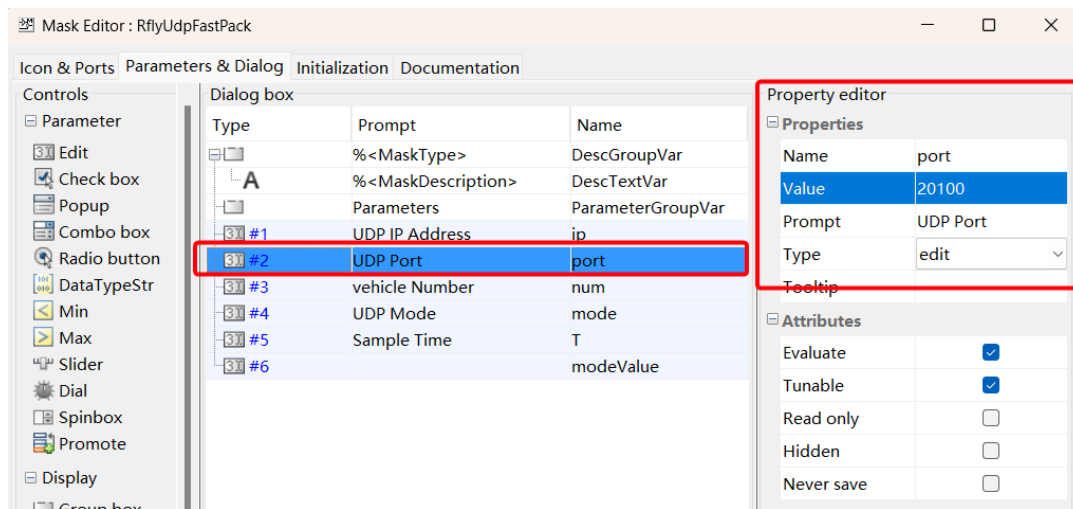
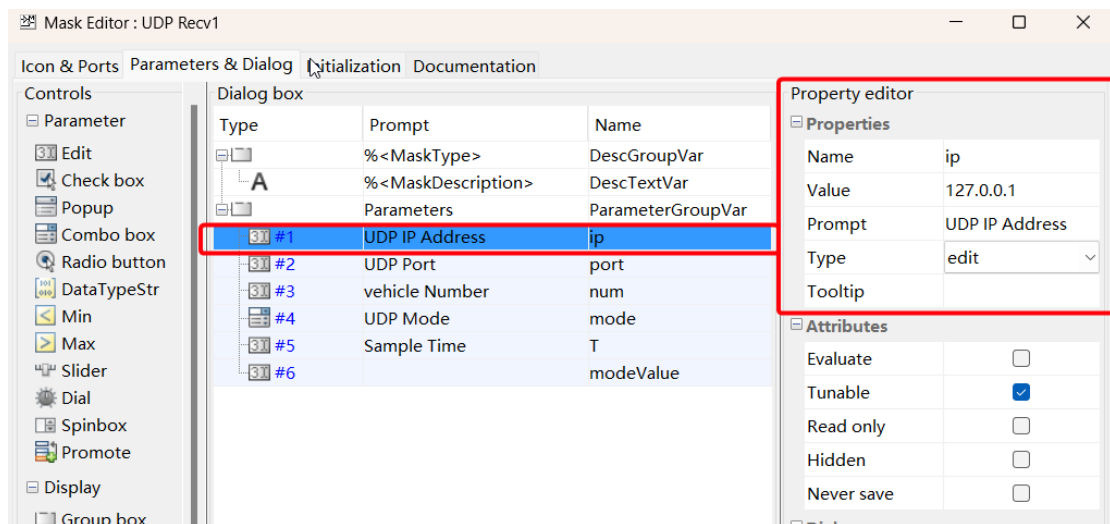
## Step 6:

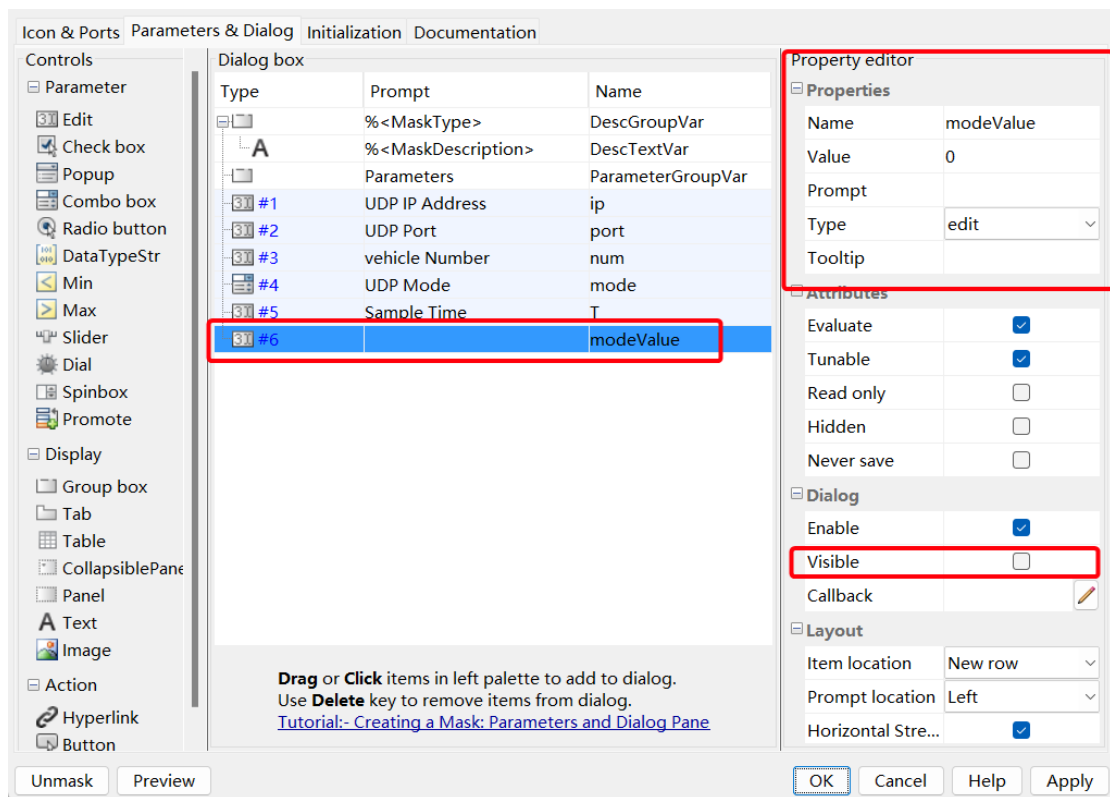
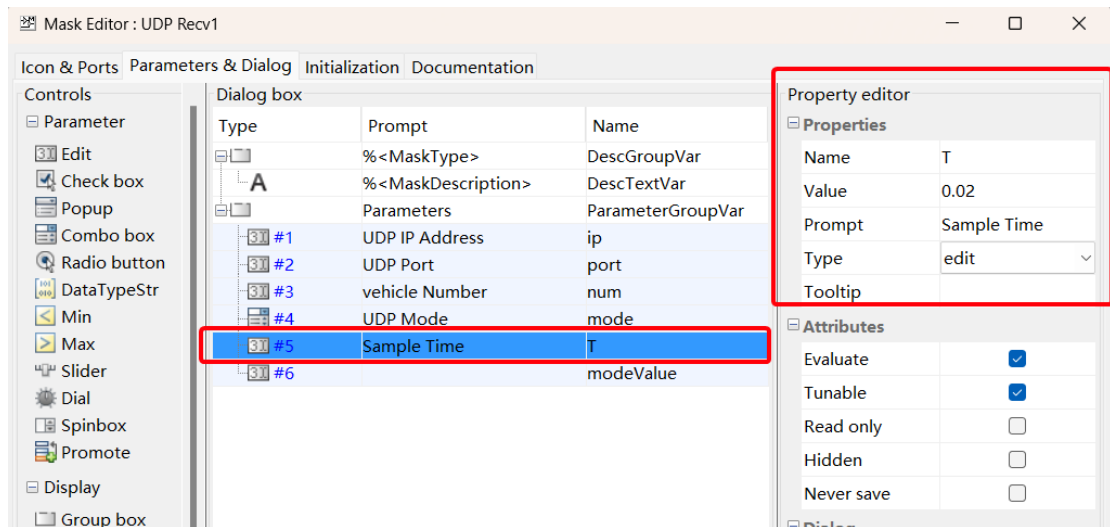
在该模块上鼠标右击选择 **Mask->Ceate Mask**，或者选中该模块按下 **Ctrl+M** 键。即可弹出如下对话框，按照图中进行设置：



## Step 7:

设置 ip、port、vehicle Number、T、modeValue 数据的属性如下图所示：

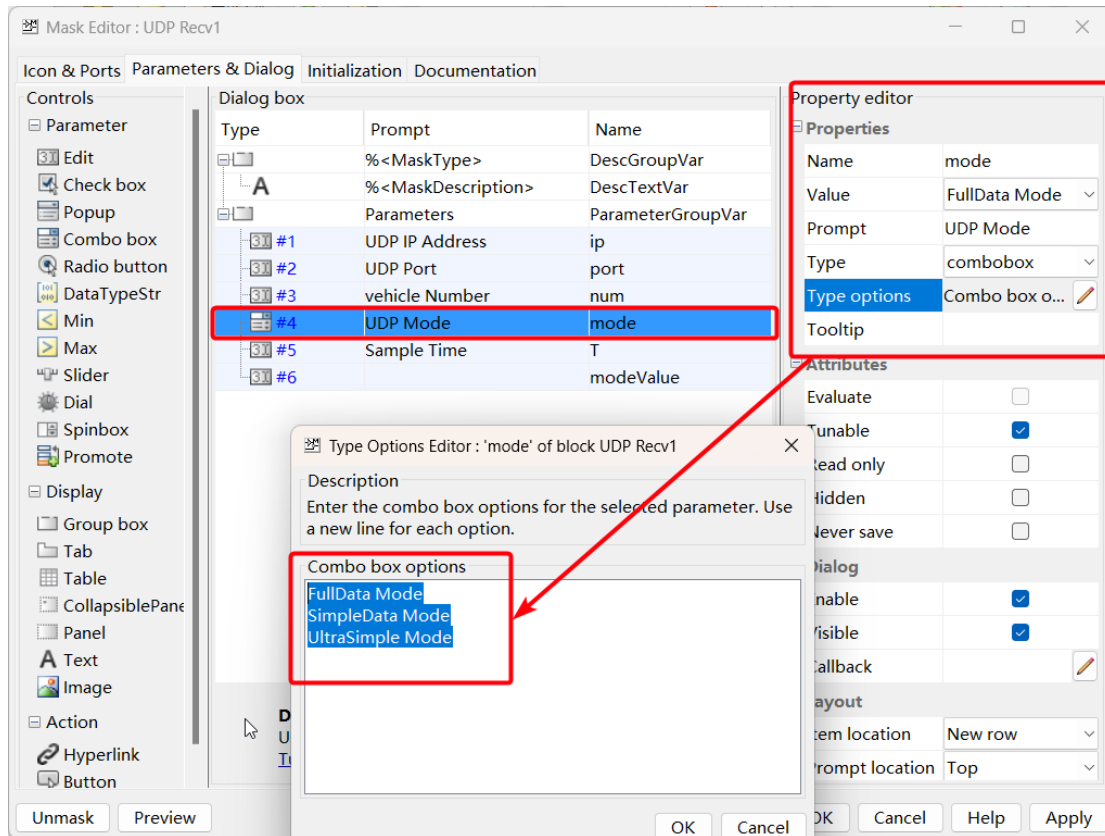




设置 UDP Mode 参数属性的 Type 为 “combobox”，在 Type option 中的点击小铅笔图标，在弹出的对话框中输入：

FullData Mode  
SimpleData Mode  
UltraSimple Mode

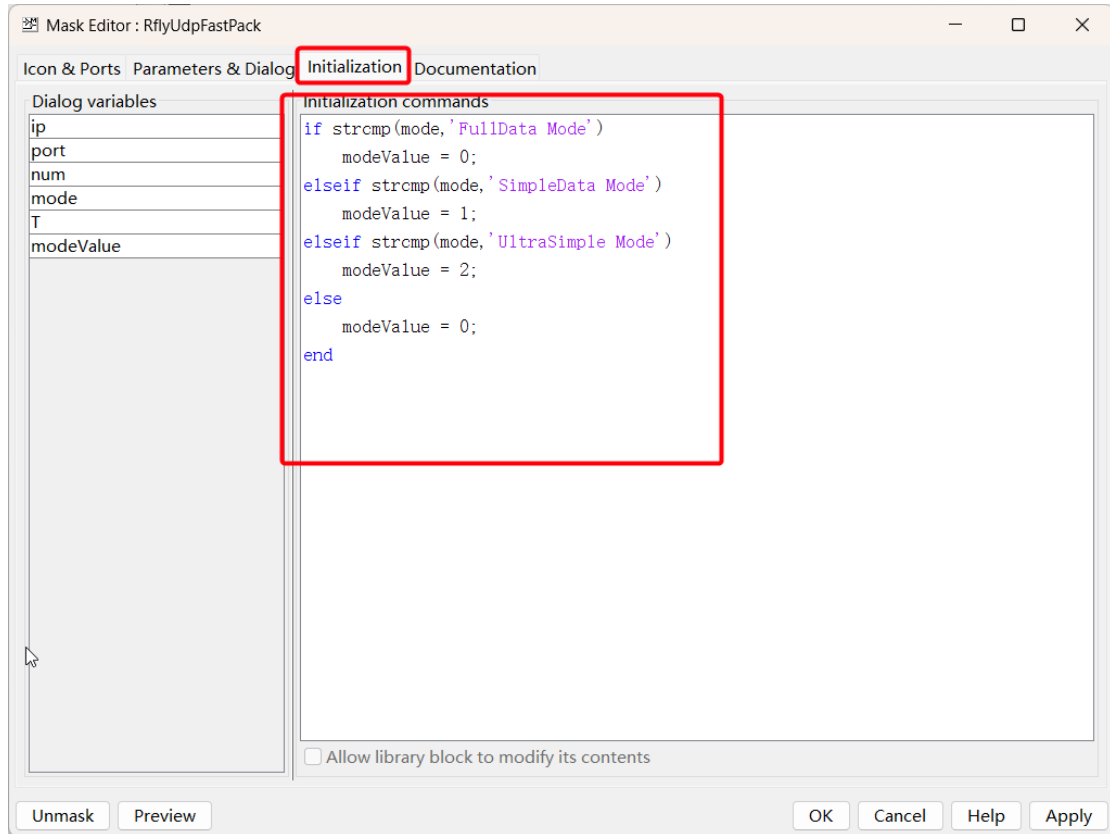
点击 OK 关闭该对话框。



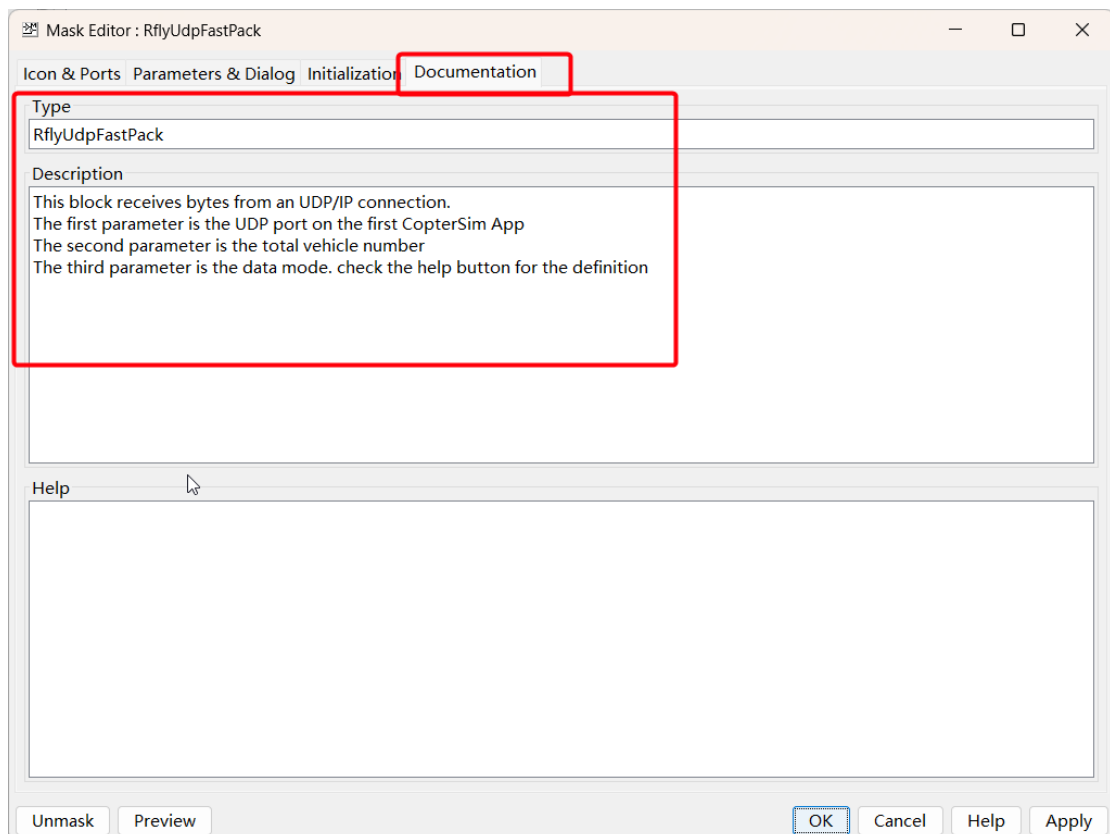
## Step 8:

进入该模型的 Initialization 中，在其中输入如下代码：

```
if strcmp(mode,'FullData Mode')
    modeValue = 0;
elseif strcmp(mode,'SimpleData Mode')
    modeValue = 1;
elseif strcmp(mode,'UltraSimple Mode')
    modeValue = 2;
else
    modeValue = 0;
end
```



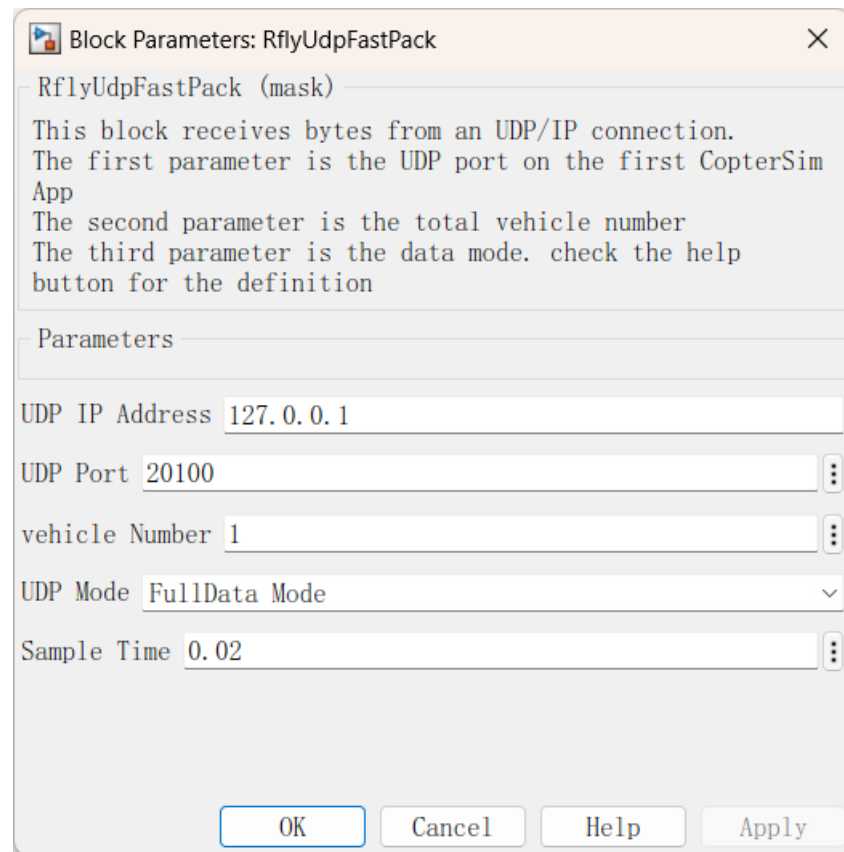
进入该模型的 **Documentation** 中，进行如下设置，该设置只是用于显示解释该模型的作用使用，不进行设置该模块也可正常使用。





## Step 9

完成上述所有设置之后，点击该模型右下角的 **Apply** 和 **OK** 后，即可保存该模块的设置。双击该模块即可弹出如下界面。



## 6、参考资料

[1]. S 函数官方 PDF 文档: [https://www.mathworks.com/help/pdf\\_doc/simulink/index.html](https://www.mathworks.com/help/pdf_doc/simulink/index.html)

---

Documentation Examples Functions Blocks Videos Answers

---

### PDF Documentation for Simulink

[Simulink Getting Started Guide](#)  
[Simulink User's Guide](#)  
[Simulink Graphical User Interface](#)  
[Simulink Reference](#)  
[Developing S-Functions](#)  
[Simulink Release Notes](#)

**Modeling Guidelines**  
[Modeling Guidelines for High-Integrity Systems](#)  
[Modeling Guidelines for Code Generation](#)  
[MathWorks Automotive Advisory Board Control Algorithm Modeling Guidelines Using MATLAB, Simulink, and Stateflow](#)

[2]. RflyUdpFast.cpp 代码解析:

```
RflyUdpFast.cpp X
C: > PX4PSP > RflySimAPIs > SimulinkSwarmAPI > SimulinkDemo > RflyUdpFast.cpp > E
1  /* RflySim Simulink UDP Interface *****
2  /* Xunhua Dai, 2020/05/03, Beihang University *****
3
4  #define S_FUNCTION_NAME RflyUdpFast
5  #define S_FUNCTION_LEVEL 2
6  #define nonblockingsocket(s) {unsigned long ctl = 1;ioctlsocket(s, CTL_NONBLOCK, &ctl);}
7  #define _WINSOCK_DEPRECATED_NO_WARNINGS 1
8
9  #define MAXLEN 65536
10
11  #include "simstruc.h"
12  #include <math.h>
13  #include <stdio.h>
14  #include <string.h>
15  #include <winsock2.h>
16  #pragma comment(lib, "ws2_32.lib")
17
18  #define PAYLOAD_LEN_SHORT_SHORT 112
19
20  typedef signed char      int8_t;
21  typedef short            int16_t;
22  typedef int              int32_t;
23  typedef long long        int64_t;
24  typedef unsigned char    uint8_t;
25  typedef unsigned short   uint16_t;
26  typedef unsigned int      uint32_t;
27  typedef unsigned long long uint64_t;
28
29  struct outHILStateData {
30      uint32_t time_boot_ms; //Timestamp of the message
31      uint32_t copterID;     //Copter ID, start from 1
32      int32_t GpsPos[3];     //Estimated GPS positionE-lat&long: deg*1e7, alt: m*1e3
33      int32_t GpsVel[3];     //Estimated GPS velocity, NED, m/s*1e2->cm/s
34      int32_t gpsHome[3];    //Home GPS position, lat&long: deg*1e7, alt: m*1e3 and
35      int32_t relative_alt;  //alt: m*1e3 and up is positive
36      int32_t hdg;           //Course angle, NED,deg*1000, 0~360
37      int32_t satellites_visible; //GPS Raw data, sum of satellite
38      int32_t fix_type;      //GPS Raw data, Fixed type, 3 for fixed
39      int32_t resrveInt;     //Int, reserve for the future use
40      float AngEular[3];     //Estimated Euler angle, unit: rad
41      float localPos[3];     //Estimated locoal position, NED, unit: m
42      float localVel[3];     //Estimated locoal velocity, NED, unit: m/s
43      float pos_horiz_accuracy; //GPS horizontal accuracy, unit: m
44      float pos_vert_accuracy; //GPS vertical accuracy, unit: m
45      float resrveFloat;     //float,reserve for the future use
46  }
47
48  typedef struct _netDataShortShort {
49      int tg;
50      int len;
51      char payload[PAYLOAD_LEN_SHORT_SHORT];
52      _netDataShortShort() {
53          memset(payload, 0, sizeof(payload));
54      }
55  } netDataShortShort;
56
57  struct outHILStateShort {
58      int checksum;
59      int32_t gpsHome[3]; //Home GPS position, lat&long: deg*1e7, alt:
60      float AngEular[3]; //Estimated Euler angle, unit: rad
61      float localPos[3]; //Estimated locoal position, NED, unit: m
62      float localVel[3]; //Estimated locoal velocity, NED, unit: m/s
63  }
```

1. 定义区: S函数名字, 层级, 导入头文件, 自定义变量等

2. 数据接收结构体, 获取Pixhawk转发的MAVLink飞控状态

3. 其他通信接口结构体

```

194  /* mdlCheckParameters, check parameters, this routine is called later
195  #define MDL_CHECK_PARAMETERS
196  static void mdlCheckParameters(SimStruct *S)
197  {
198      /* Basic check : All parameters must be real positive vectors
199      double *dPort, *dUdpMode, *numVehicle;
200      int iPort, iUdpMode, iNumVehicle;

```

4. S函数模块参数校验函数

```

404  /* mdlStart - initialize work vectors *****
405  #define MDL_START
406  static void mdlStart(SimStruct *S)
407  {
408      WSADATA wsa_data;
409      int status;
410      SInfo *info;
411
412      /* get buffer size */
413      //int iBufSize = (int)floor((*mxGetPr(ssGetSFcnParam(S,2)))+0.5);
414      int iBufSize =240;
415      /* retrieve pointer to pointers work vector */
416      void **PWork = ssGetPWork(S);

```

5. 仿真开始函数

```

241  /* mdlInitializeSizes - initialize the sizes array *****
242  static void mdlInitializeSizes(SimStruct *S)
243  {
244      //int iBufSize;
245
246      ssSetNumSFcnParams(S,5); /* number of expected pa
247
248      /* Check the number of parameters and then calls mdlCheckParameters to see
249      if (ssGetNumSFcnParams(S) == ssGetSFcnParamsCount(S))
250      { mdlCheckParameters(S); if (ssGetErrorStatus(S) != NULL) return; } else r

```

6. 初始化函数

```

447  /* mdlOutputs - compute the outputs *****
448  static void mdlOutputs(SimStruct *S, int_T tid)
449  {
450      //int iBufSize = (int)floor((*mxGetPr(ssGetSFcnParam(S,2)))+0.5);
451      //UNUSED(tid);
452      UNUSED_ARG(tid);
453      int iBufSize =240;
454      double *numVehicle=(double *)mxGetPr(ssGetSFcnParam(S,1));
455      int iNumVehicle = (int)floor((*numVehicle)+0.5);
456
457      double *dUdpMode = (double *)mxGetPr(ssGetSFcnParam(S,2));
458      int iUdpMode = (int)floor((*dUdpMode)+0.5);

```

7. 更新输出函数

```

569  #define MDL_UPDATE
570  /* Function: mdlUpdate =====
571  * Abstract:
572  *      xdot = Ax + Bu
573  */
574  static void mdlUpdate(SimStruct *S, int_T tid)
575  {
576      UNUSED_ARG(tid);
577      double *numVehicle=(double *)mxGetPr(ssGetSFcnParam(S,1));
578      int iNumVehicle = (int)floor((*numVehicle)+0.5);
579
580      double *dUdpMode = (double *)mxGetPr(ssGetSFcnParam(S,2));
581      int iUdpMode = (int)floor((*dUdpMode)+0.5);

```

8. 更新状态函数

## 7、常见问题

1. 无