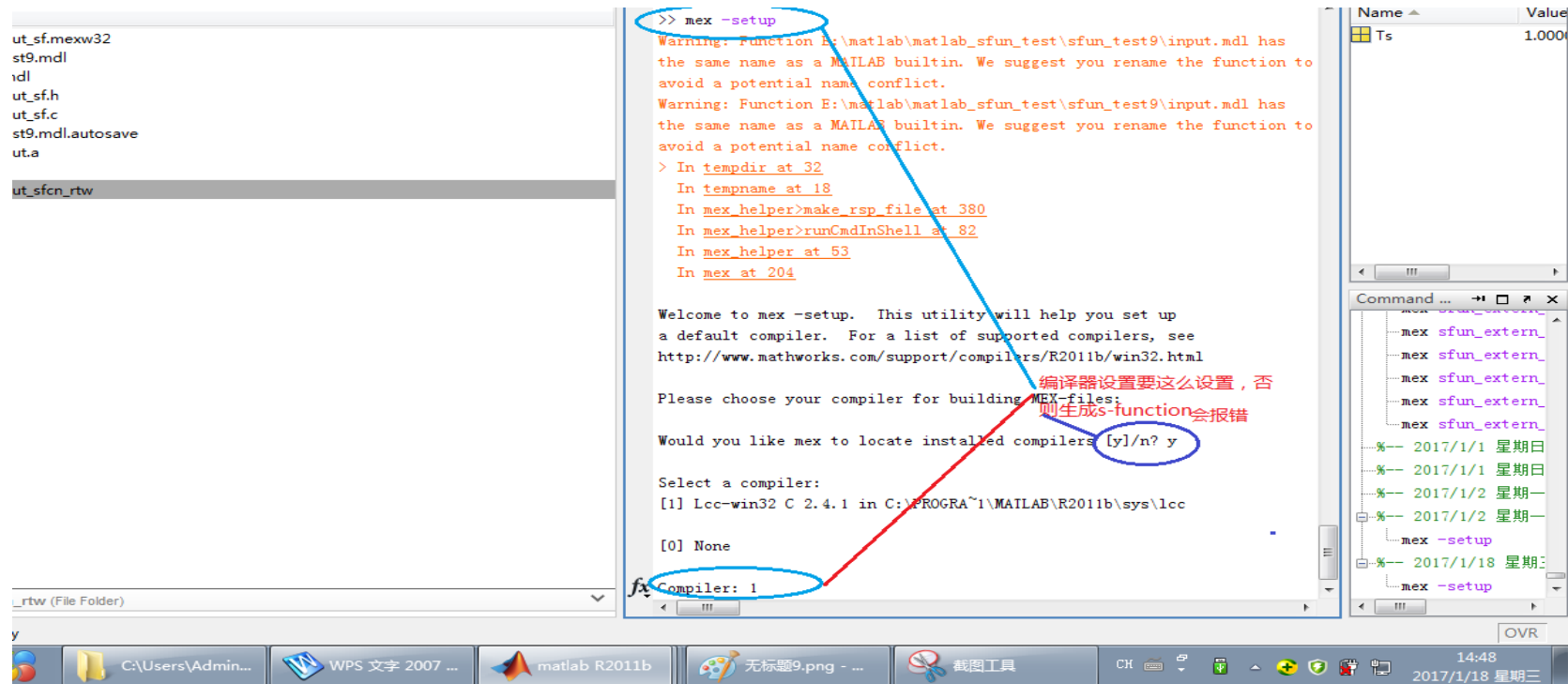


## Matlab 模型生成 S-function 及手动编译过程

### 1. 首先在 Matlab2014B 中设置 mex 编译器



## 2. 不正确设置 mex 编译器报的故障

编译器设置不正确报的故障  
编译器应该选择Lcc-win32 C 2.4.1

Subsystem build: sfun\_test9

Message	Source	Reported By	Summary
Build Error	sfun_test9	Simulink	Error(s) encountered while building model "RecInput"

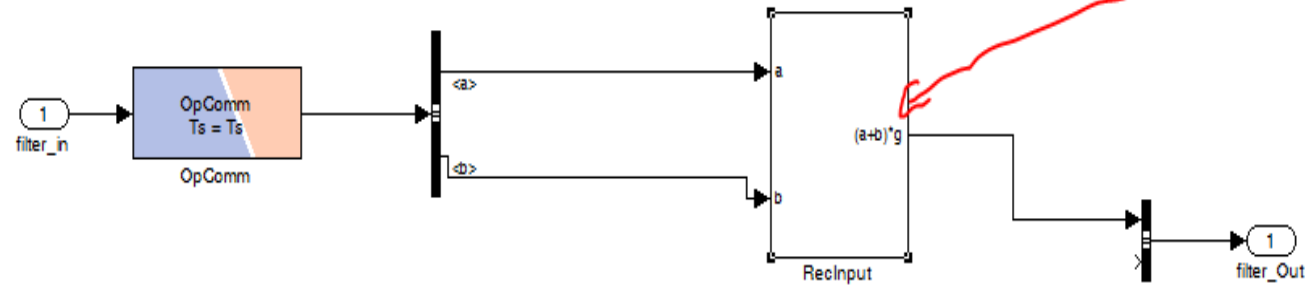
sfun\_test9

Error(s) encountered while building model "RecInput"

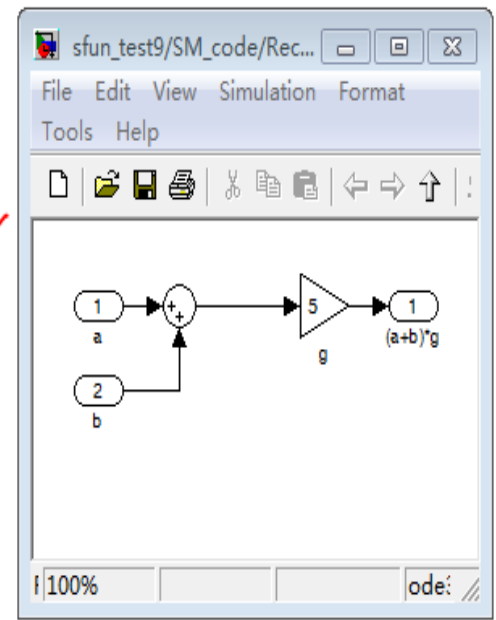
Open Help Close

14:57 2017/1/18 星期三

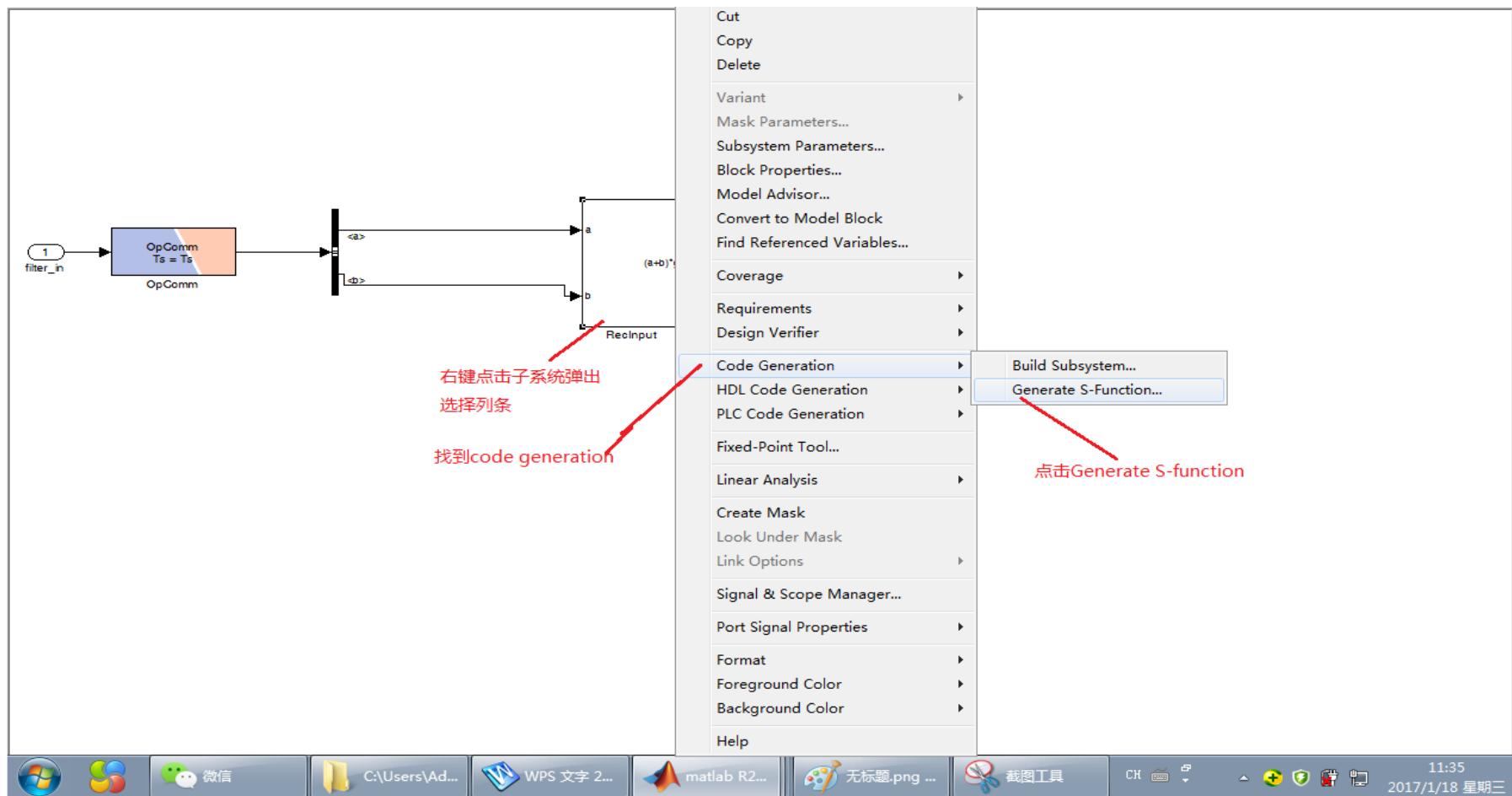
### 3.把模型封装成子系统



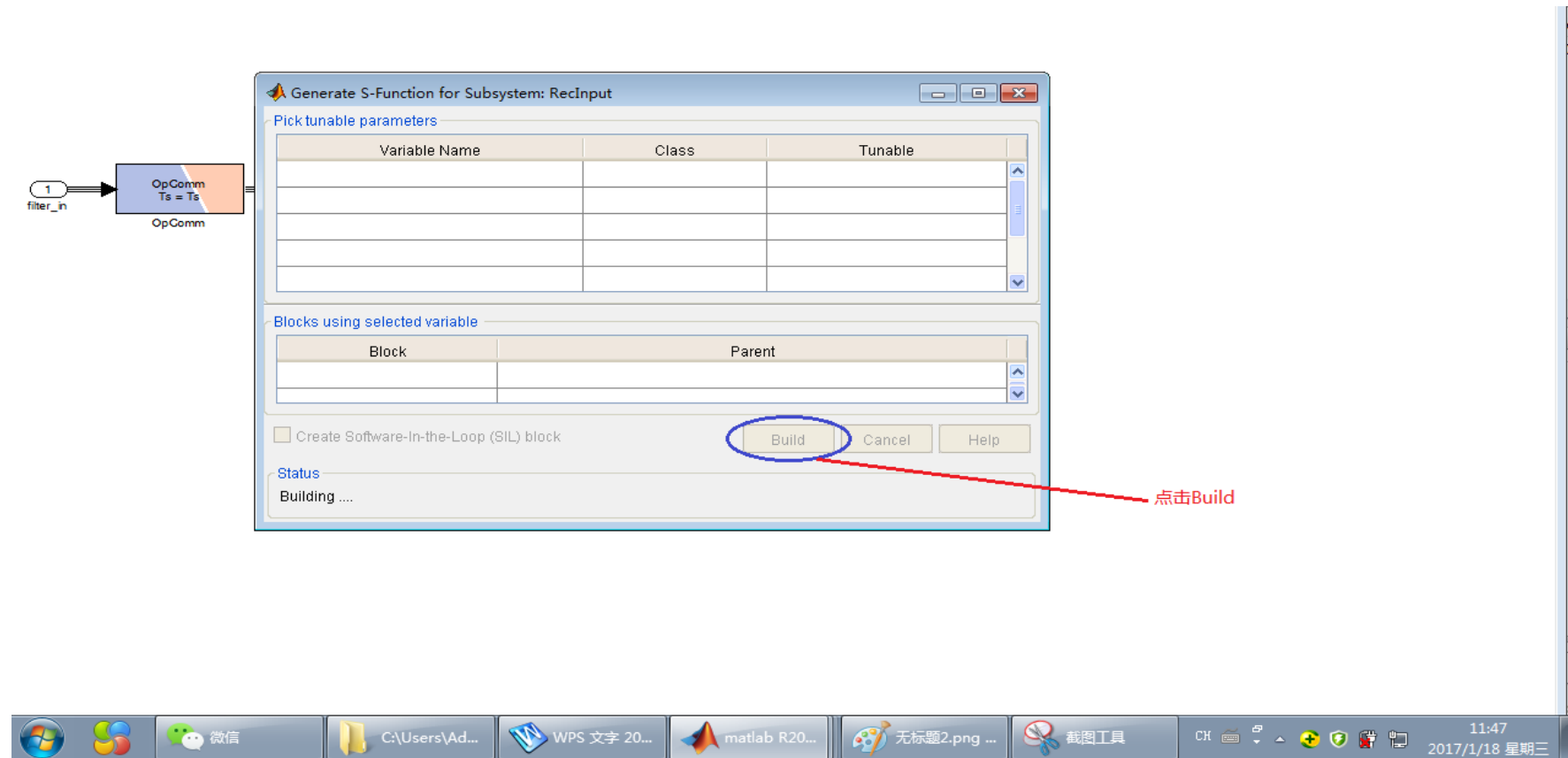
建立子系统



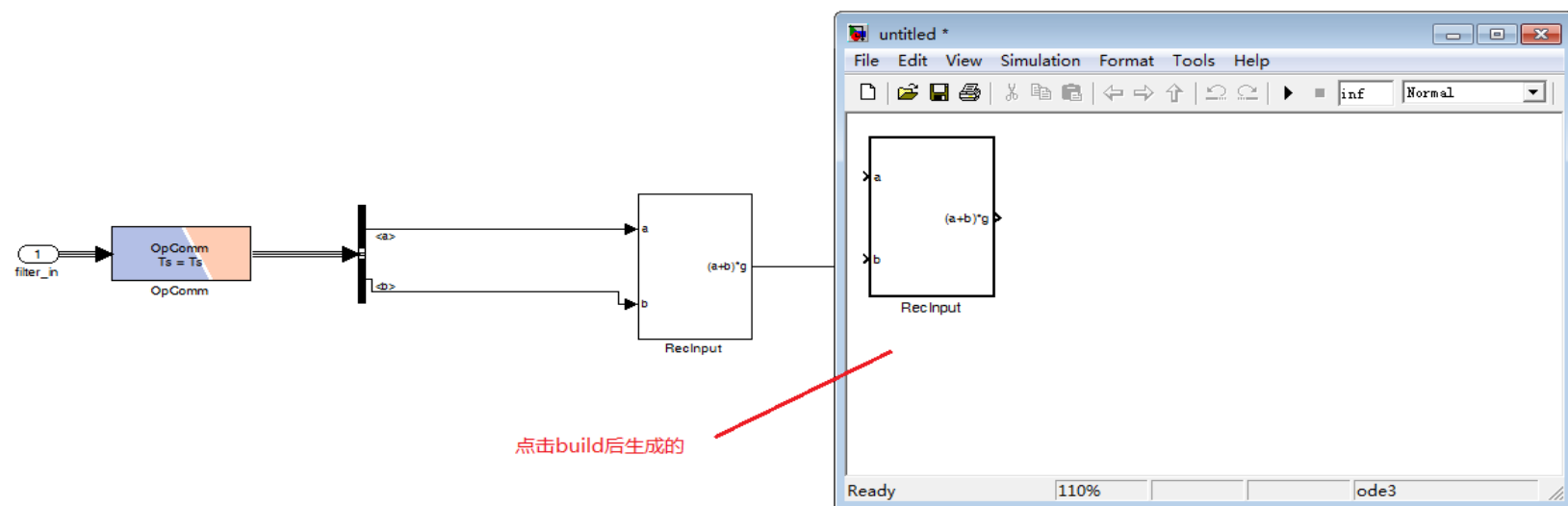
#### 4.编译封装好的模块



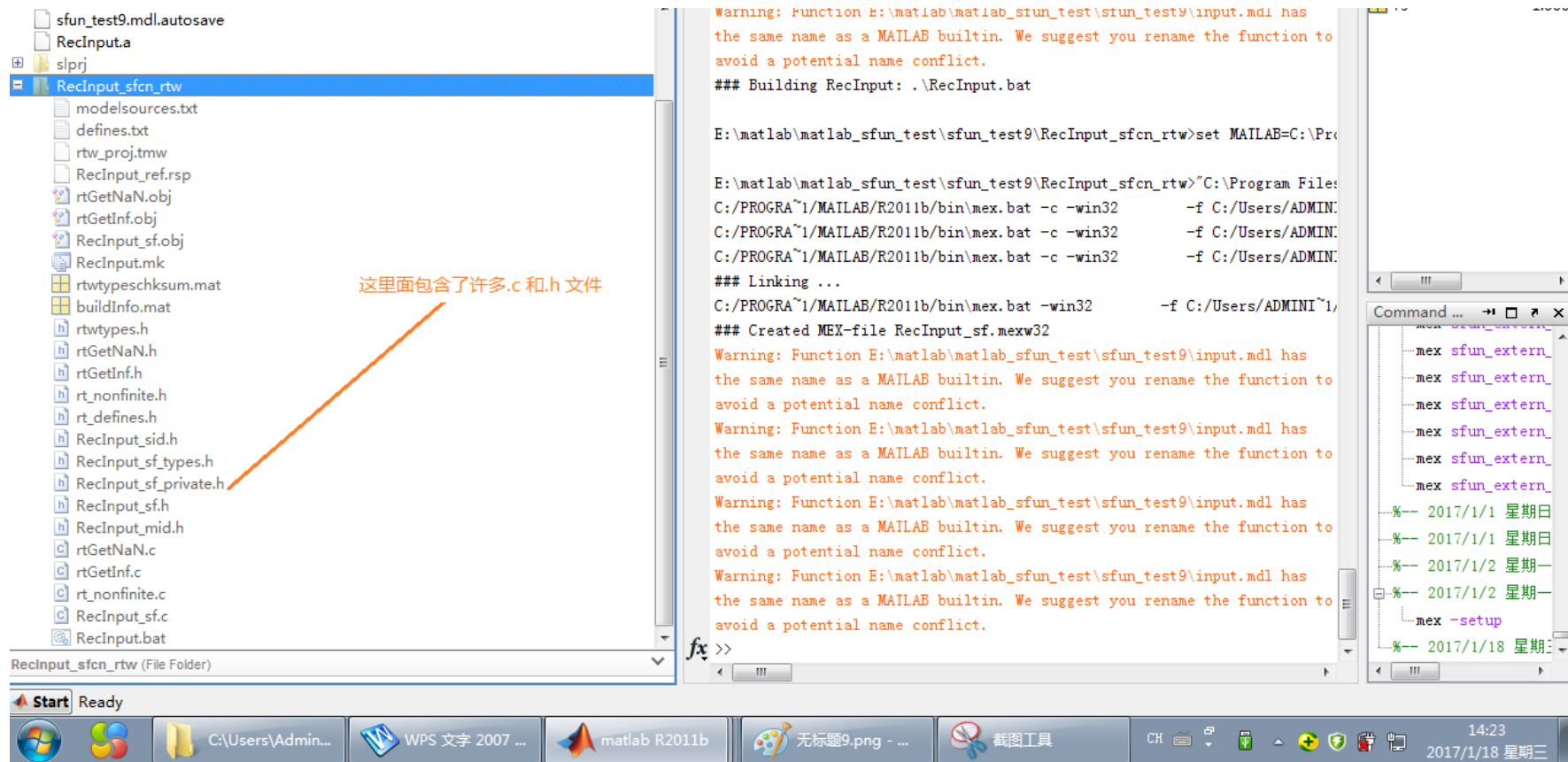
5. 点击 Generation S-function 函数后在弹出的画面上点击 Build。



6. 点击完 Build 后需要等等十多秒成功弹出如下的画面, 并保存 untitled 文件中就是生成的 S-function 模型。

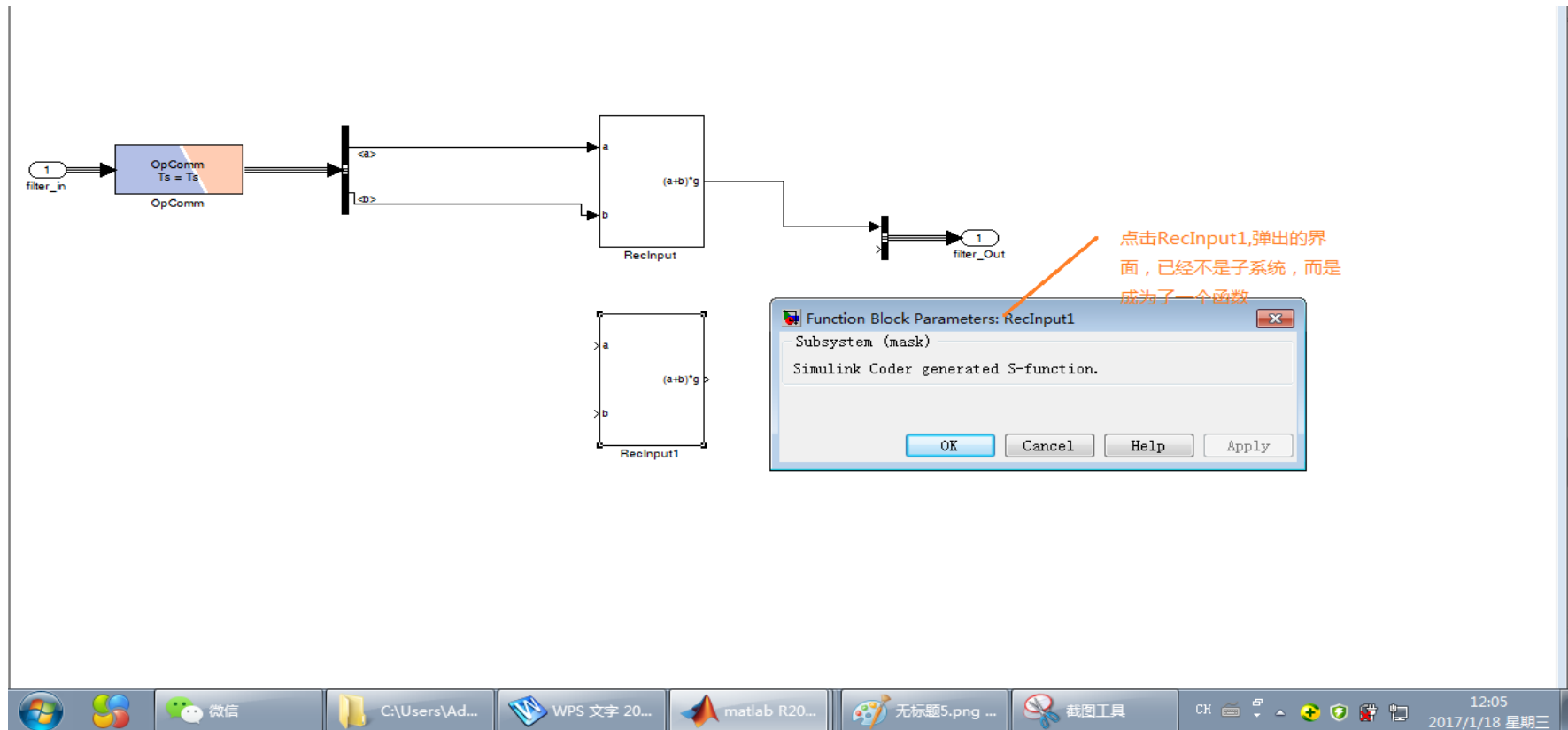


[illegible]





9.把生成的 S-function 模型拷贝到原来所建的 matlab 模型文件中，双击这个模型可以查看这个模块已经变为 S-function



10.把 S-function 模块生成过程中产生的文件夹的内容都拷贝到 linux 系统下的 ma 文件中，并且修改其中 opsfun\_linux32\_v1.1.mk 中指定的位置，注意：此处的 linux 和 C++编译器均需要 32 位。C++编译器版本不能太高。

```
#-----  
# USER-DEFINED PARAMETERS  
#-----  
# Name of static library to be generated. Do not write file extension.  
LIB_NAME = SimInput  
# Source files. Do not forget to write all of them. Ex: SRC = source1.c source2.c  
SRC = srcinput_sf.c  
# Matlab version. Ex: For R2011b, MATLAB_VERSION = v7.13  
MATLAB_VERSION = v7.13  
# For intel compiler, RTLAB_INTEL_COMPILER = 1. For gcc compiler, RTLAB_INTEL_COMPILER = 0  
RTLAB_INTEL_COMPILER = 1  
#-----  
OBJS = $(SRC:.c=.o)  
  
MATLAB_INCLUDES = -I/usr/matlab/$(MATLAB_VERSION)/simulink/include -I/usr/matlab/$(MATLAB_VERSION)/extern/include -I/usr/matlab/$(MATLAB_VERSION)/rtw/c/src -I/usr/matlab/$(MATLAB_VERSION)/rtw/c/src/matrixn  
INCLUDES = -I. $(MATLAB_INCLUDES)  
LIBS =  
LIBPATHS =  
DEFINES = -DUSE_RTMODEL -DRT=RT -DRTOS1EQ=1 -DMULTITASKING=0  
  
#ifeq ($(RTLAB_INTEL_COMPILER),1)  
#   CC = opicc  
#   LD = opxiar  
#   CC_OPTS = -c -O2 -xHost -falign-functions=2 -diag-disable remark, warn, cpu-dispatch  
#   LD_OPTS =  
#else  
#   CC = gcc  
#   LD = ar  
#   CC_OPTS = -c -O2 -ffast-math -mtune=i686 -march=i686 -falign-loops=2 -falign-jumps=2 -falign-functions=2  
#   LD_OPTS =  
#endif  
  
CFLAGS = $(CC_OPTS) $(DEFINES) $(INCLUDES)  
LDFLAGS = $(LD_OPTS) $(LIBPATHS) $(LIBS)  
  
$(LIB_NAME).a: $(OBJS)  
    $(LD) rc $@ $* $(LDFLAGS)  
    @ranlib -v $@  
  
%.o: %.c  
    $(CC) -o $@ $(CFLAGS) $<  
  
clean:
```

11.在 ma 文件夹下打开命令窗口，输入 `make -f opsfun_linux32_v1.1.mk`,完成编译生成对应的库文件，由 matlab 模型生成的 s-function 手动编译完成

