TSMaster 应用笔记

AN0008

How to use the TOSUN C MiniProgram (Advanced Skills)

如何使用同星C小程序模块（高级技巧）

作者:seven

2024-03-25

# Chapter 1 TSMaster 调用外部C/C++ DLL

TSMaster支持调用外部C/C++编译后的DLL，并且与VS2022或gcc等调用方式相同，存在多种调用方式，接下来为大家介绍。

## 动态加载DLL

Windows提供动态加载DLL的方式，需要依赖windows.h头文件，动态加载DLL使用DLL中的函数，一共分为五步：

1. 引用windows.h
2. 加载DLL，获取DLL句柄
3. 声明一个函数指针，从句柄中获取函数地址
4. 调用函数
5. 结束后释放DLL

在TSMaster C小程序中实现该步骤，如下：

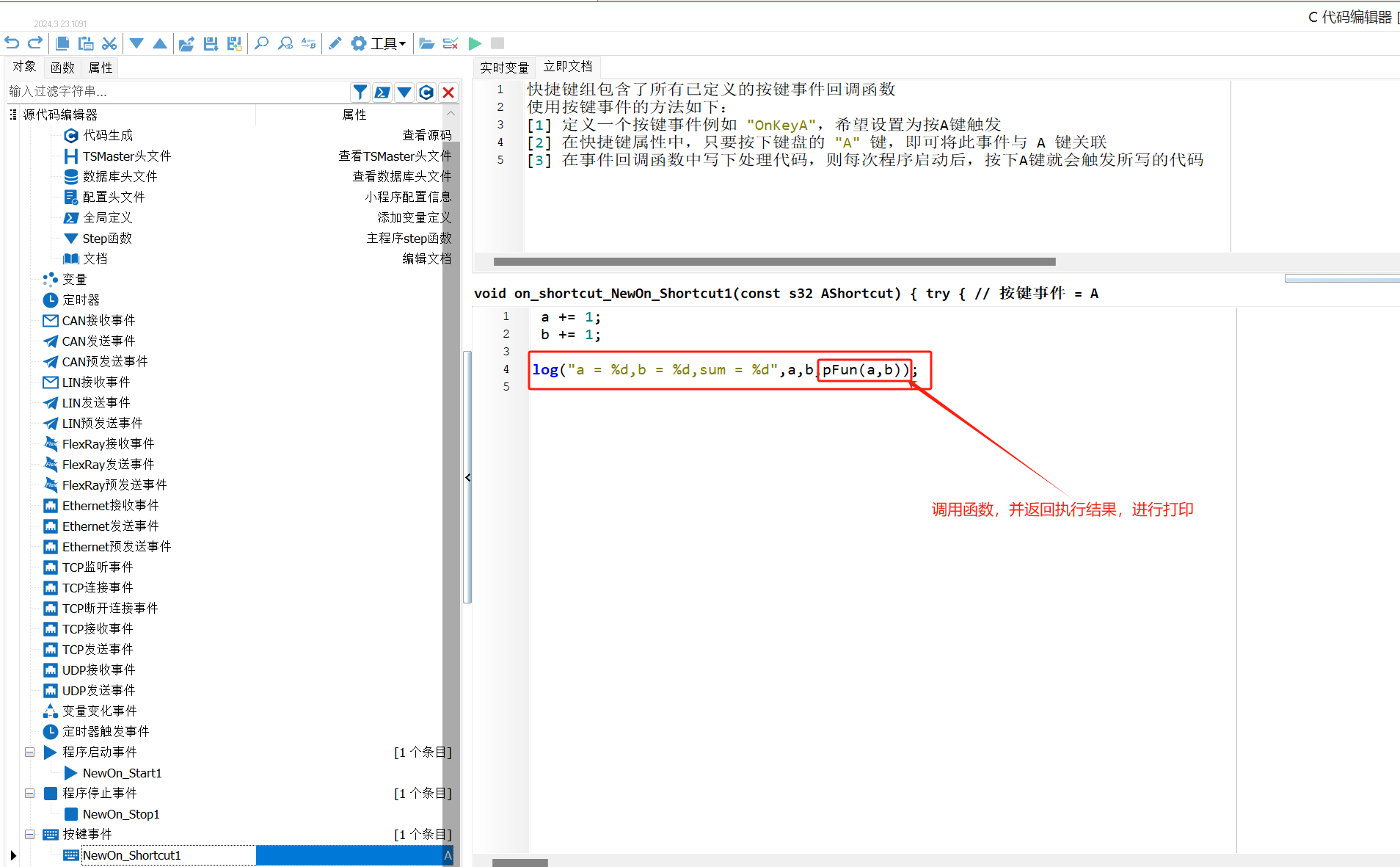
（1）在全局定义中：

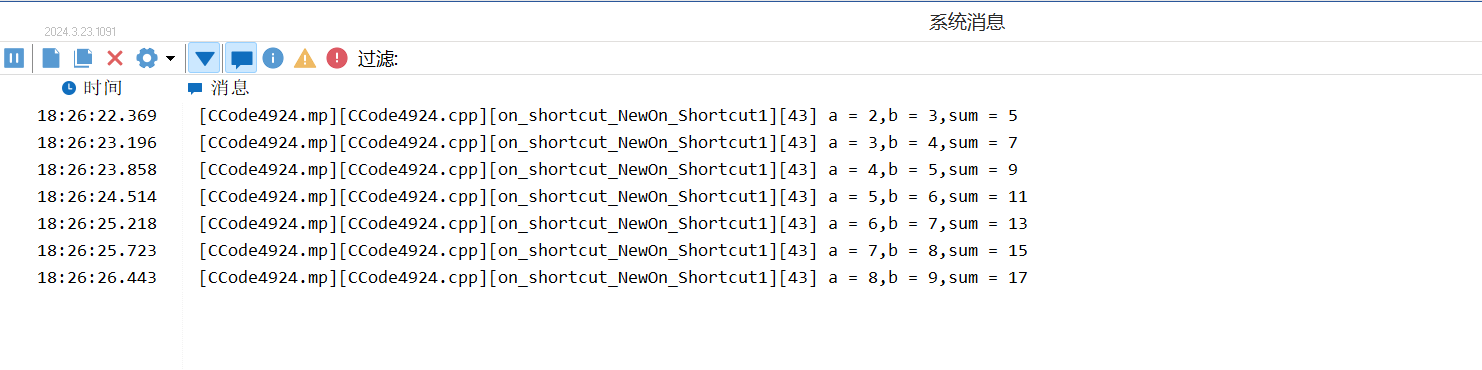


1. 在启动事件中，加载DLL并获取函数地址

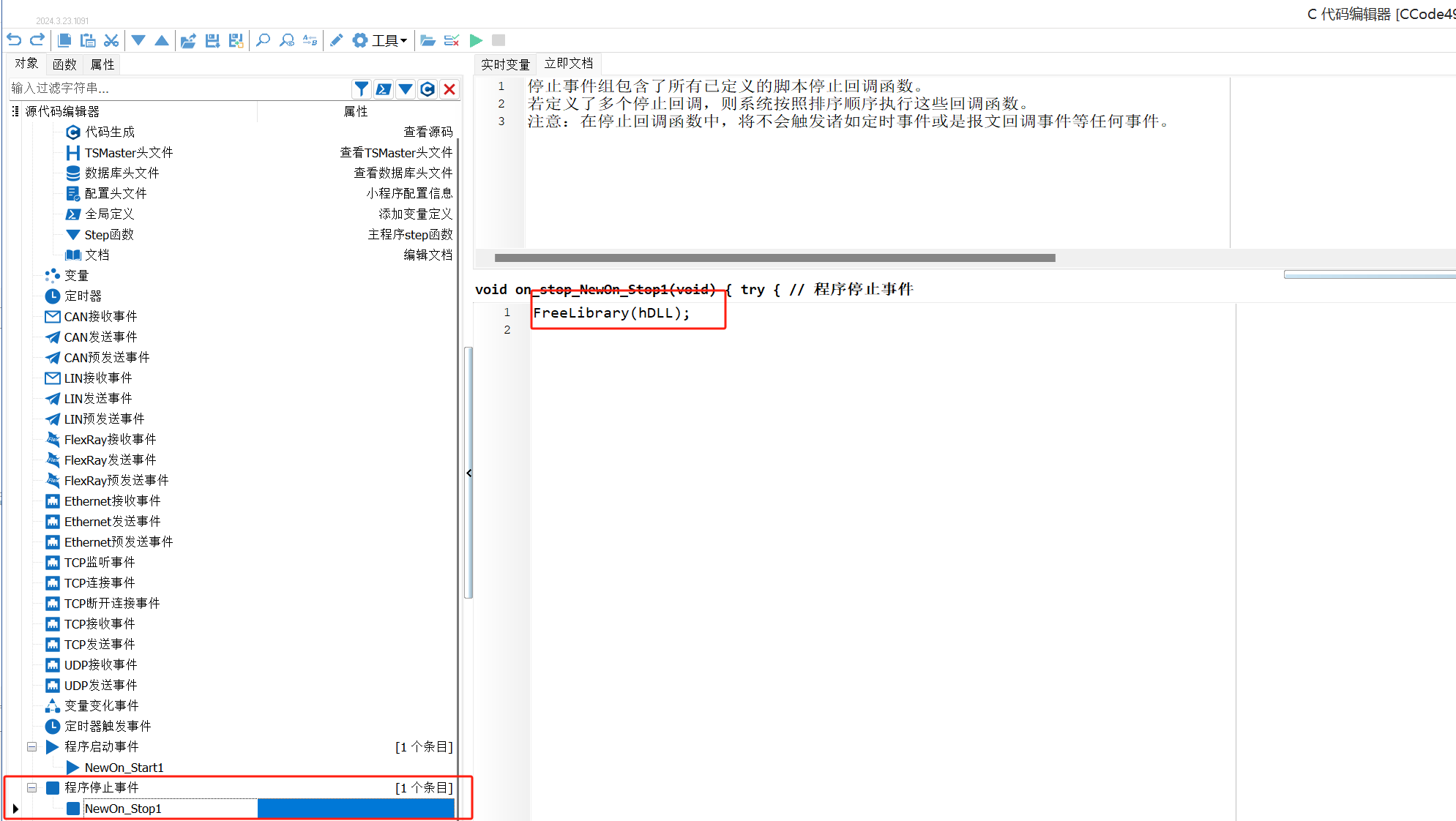


1. 在按键事件中调用函数



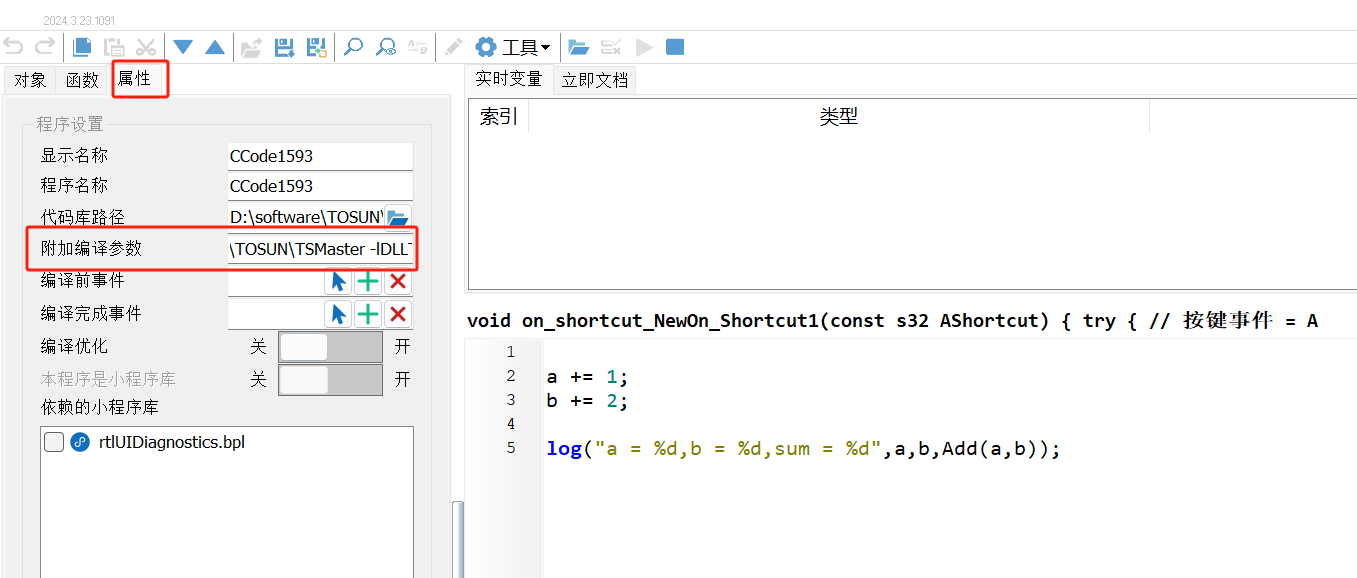


1. 在停止事件中，释放



## 静态加载DLL

静态加载，需要使用TSMaster C小程序中的属性界面中的附加编译参数

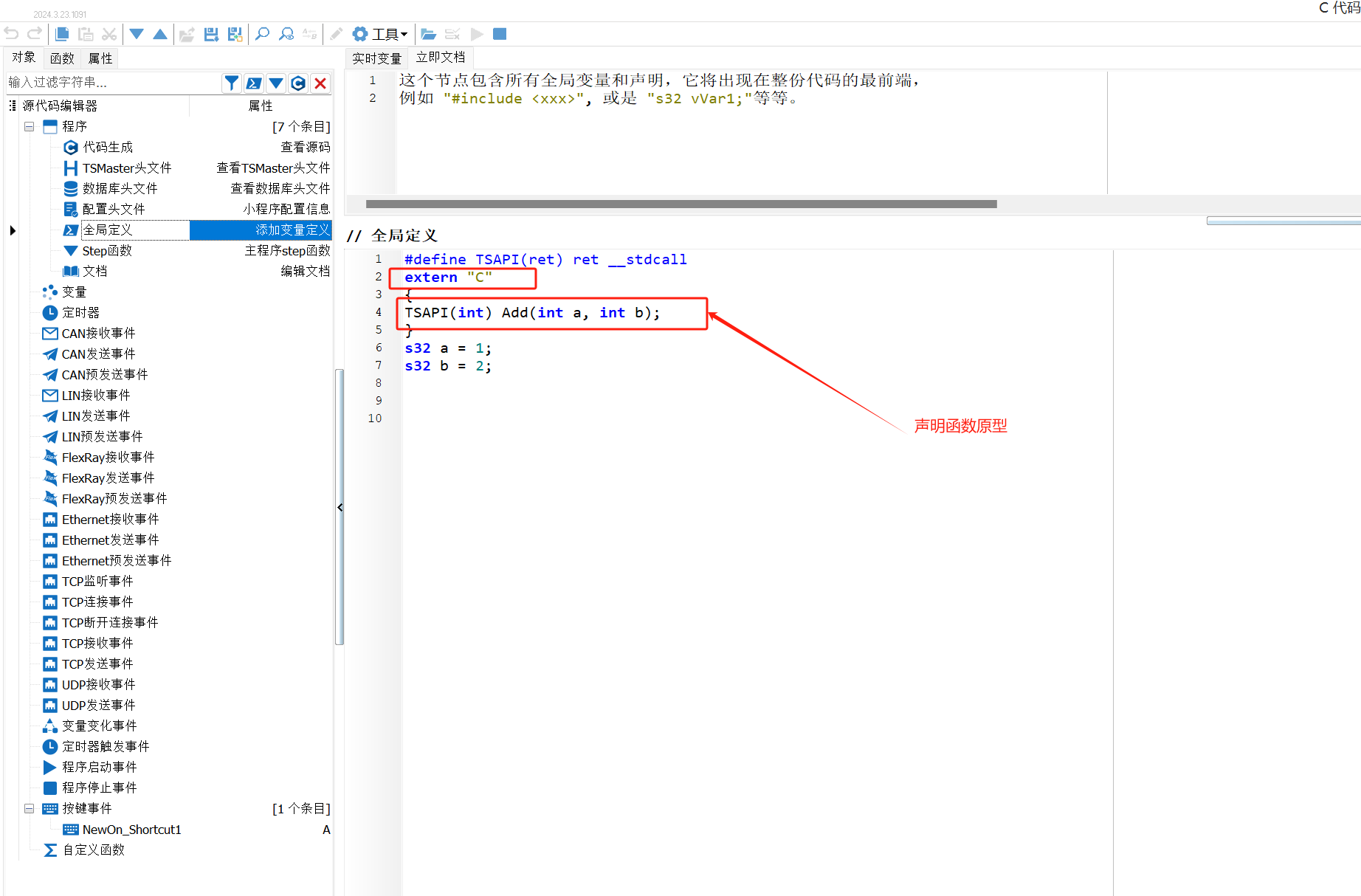


静态加载步骤分为三步：

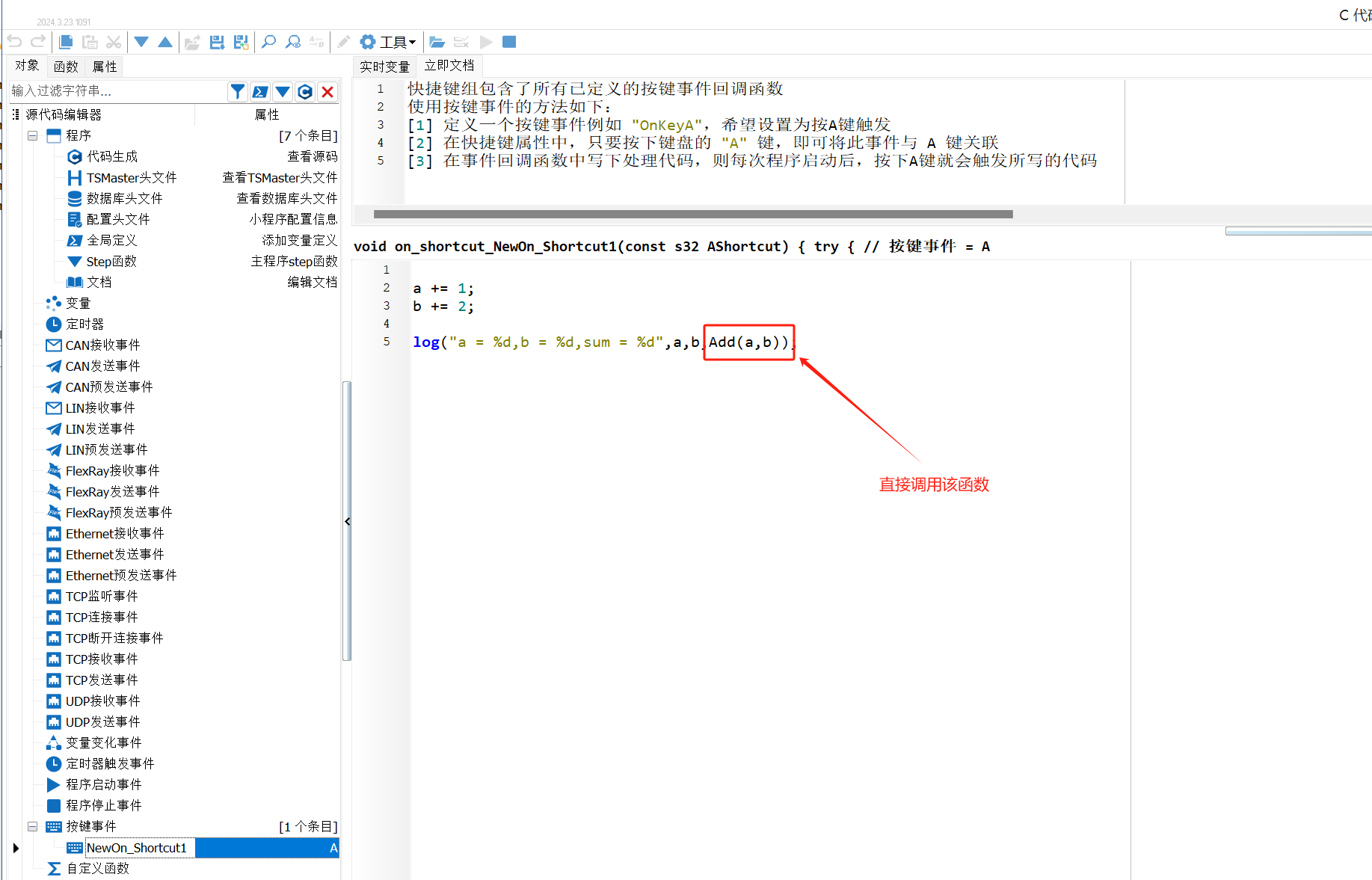
1. 加载dll头文件，或者自行定义D;;中的函数声明
2. 直接调用声明的函数
3. 在附加编译参数中，指向DLL

在TSMaster C小程序中实现该步骤，如下：

1. 在全局定义中：



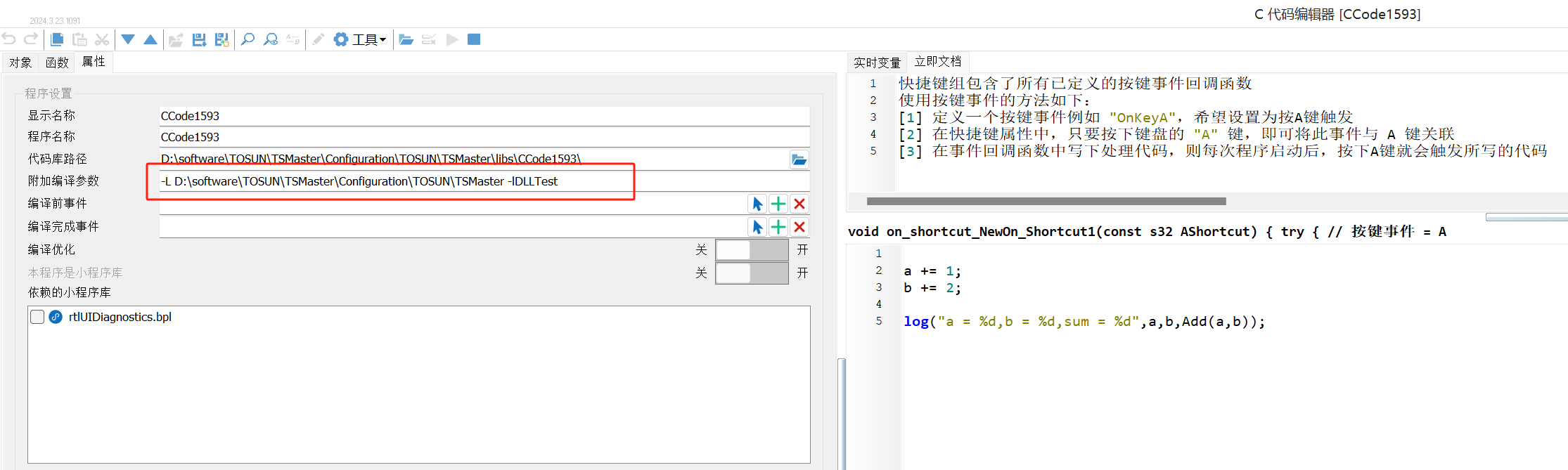
1. 在按键事件中：



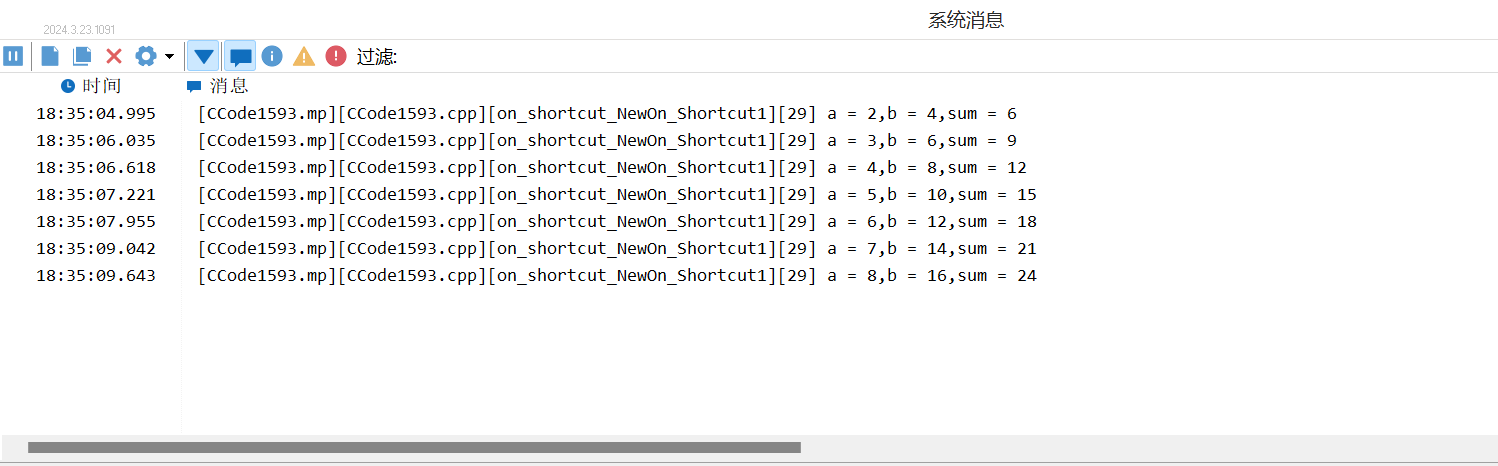
1. 增加编译指令：

指令描述：-L 后面是dll所在的路径，中间存在空格（-L ..\bin）

-l dll的文件名，中间不存在空格 （-lDLLTest）



1. 运行效果：



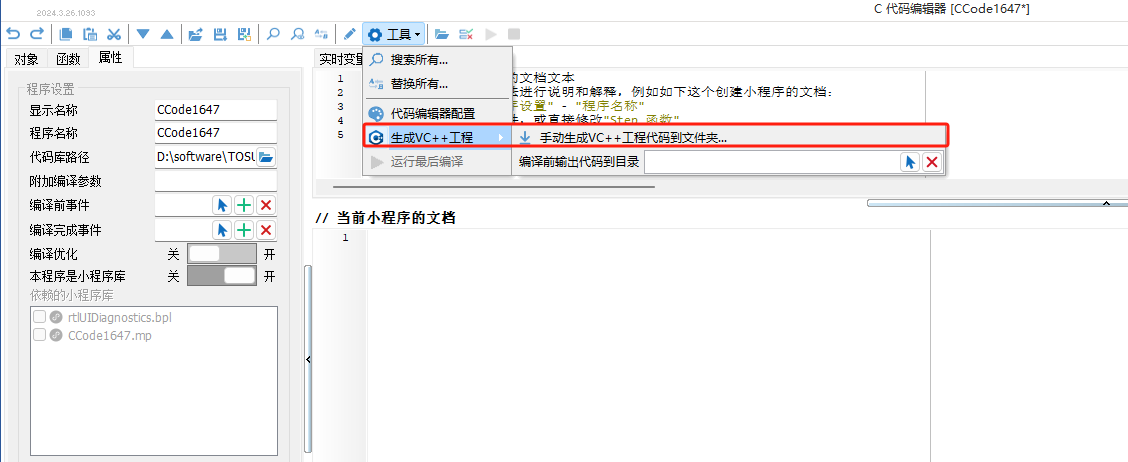
# Chapter 2 VS生成TSMaster小程序

TSMaster提供小程序模式，等同于C\C++开发动态库（DLL），与C\C++DLL相比的好处为，可以直接拖入到TSMaster中使用其中函数。并且TSMaster目前提供两种方式来生成小程序库:

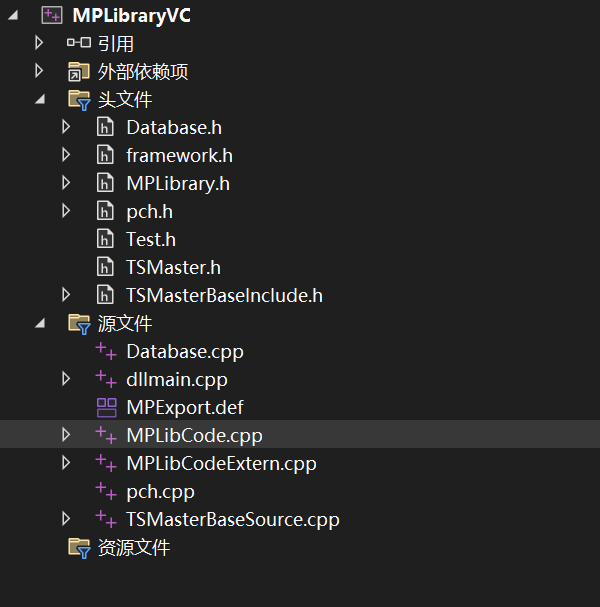
1. 直接在TSMaster C小程序中，创建自定义函数，在属性页面勾选本程序是小程序库，编译后，自定义函数即可被其他小程序调用。如下图所示：



1. 将TSMaster C小程序工程导出成为VS工程，然后按照后续内容进行编写即可，导出成VS工程如下图所示：



## 2.1 VC++工程文件介绍



TSMaster.h：为当前工程提供TSMaster的API；

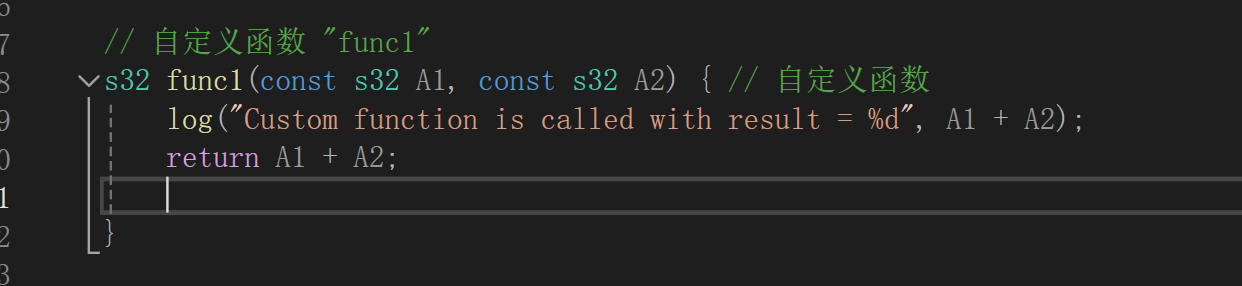
MPLibCode. cpp：函数编写文件，当然也可以新建一个cpp文件来编写，只需要引用对应头文件即可；（该cpp文件命名是根据小程序导出时自动生成）

MPLibCodeExtern,cpp：函数导出到TSMaster中使用的主要文件，会将函数指针传到TSMaster中使用；（该cpp文件命名是根据小程序导出时自动生成）

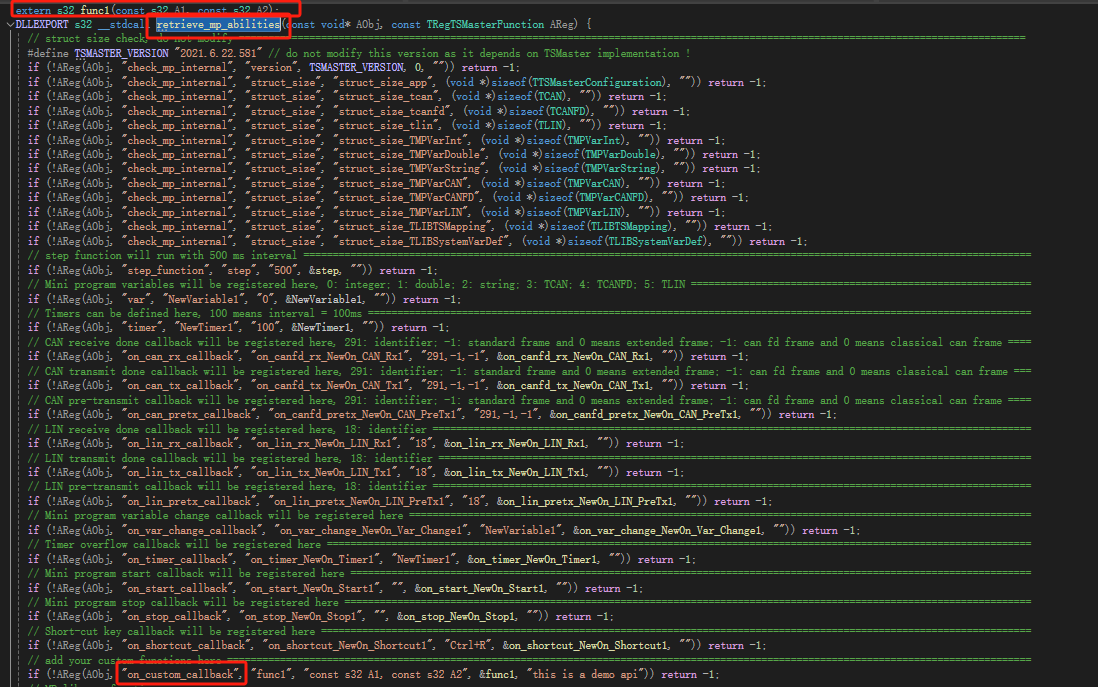
TSMasterBaseSource.cpp：可为每个函数参数添加说明，以及提供使用示例，协助共同开发；

## 2.2 函数编写示例

在MPLibCode.cpp中编写自定义函数，按照正常函数编写即可，如下图，编写一个返回值为int32类型的函数（导出到TSMaster中的函数返回值类型都为s32类型，同时参数类型要是自定义类型的情况下，将导出参数定义为void\*）。

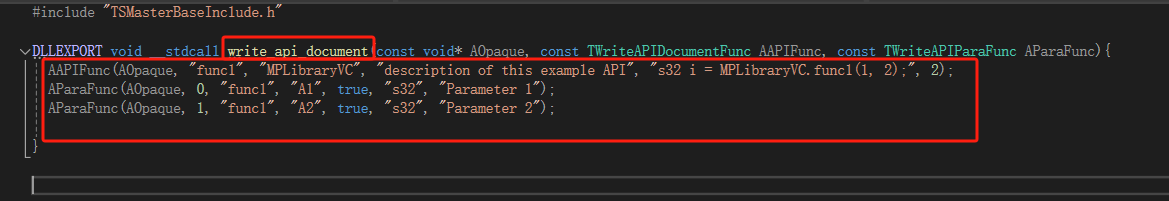


在MPLibCode.cpp中函数编写完成后，将在MPLibCodeExtern.cpp中导出自定义函数，如下图所示，先声明函数，加载extern修饰全局，然后在retrieve\_mp\_abilities函数中，添加函数声明，以及传入函数地址，如下图所示。



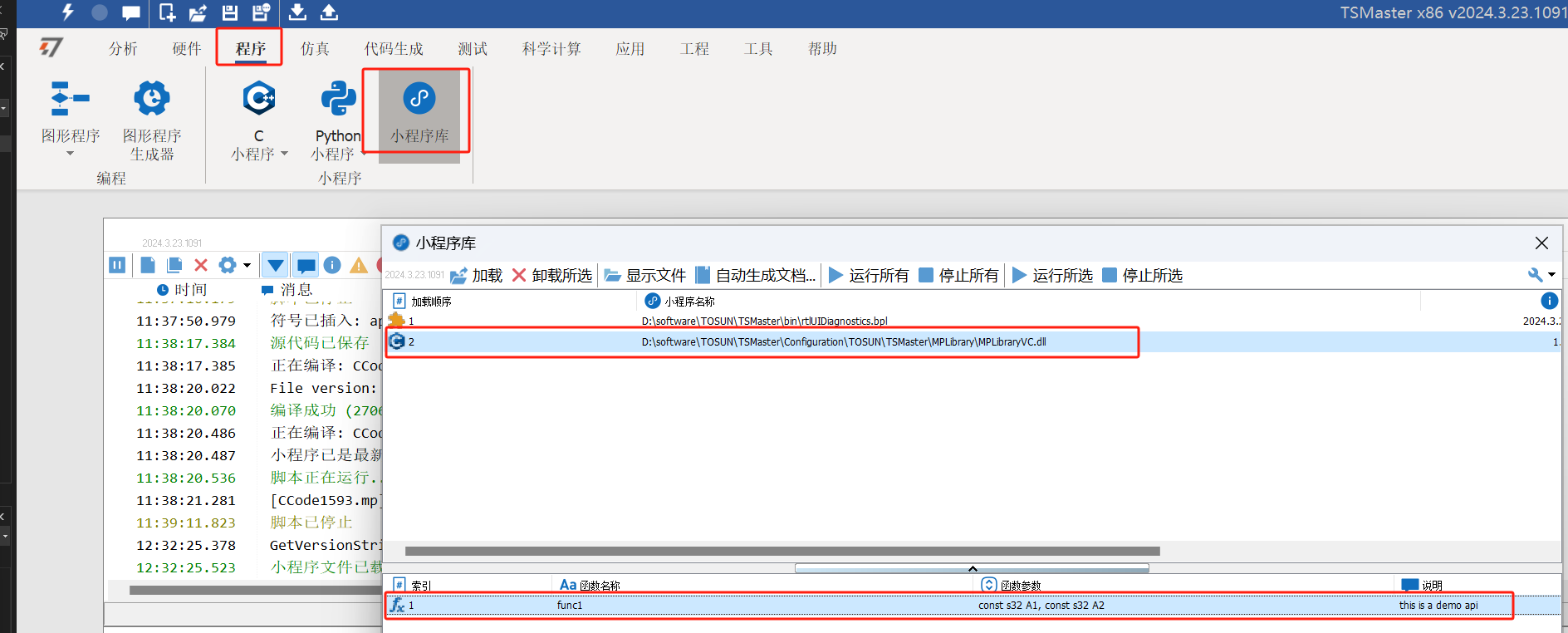
上述两步做完后，即可生成编译代码生成DLL（使用releases生成发布），在TSMasterBaseSource.cpp中编写函数使用说明，更多是为了方便使用人员理解函数参数与函数使用。

在TSMasterBaseSource.cpp中，增加AAPIFunc,来指定函数参数个数以及函数示例，下面对应函数参数个数，为使用者提供函数参数说明以及参数类型，方便使用者正确调用。

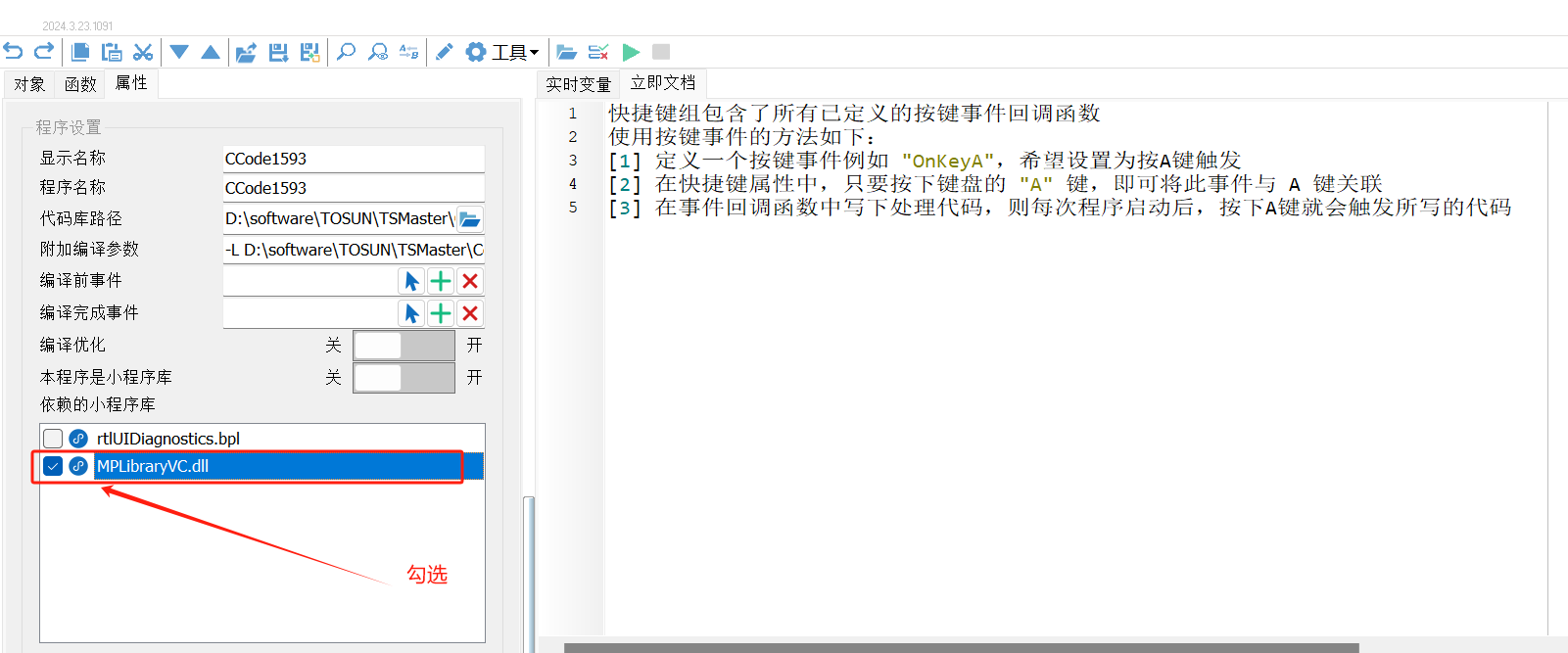


## 2.3 函数调用

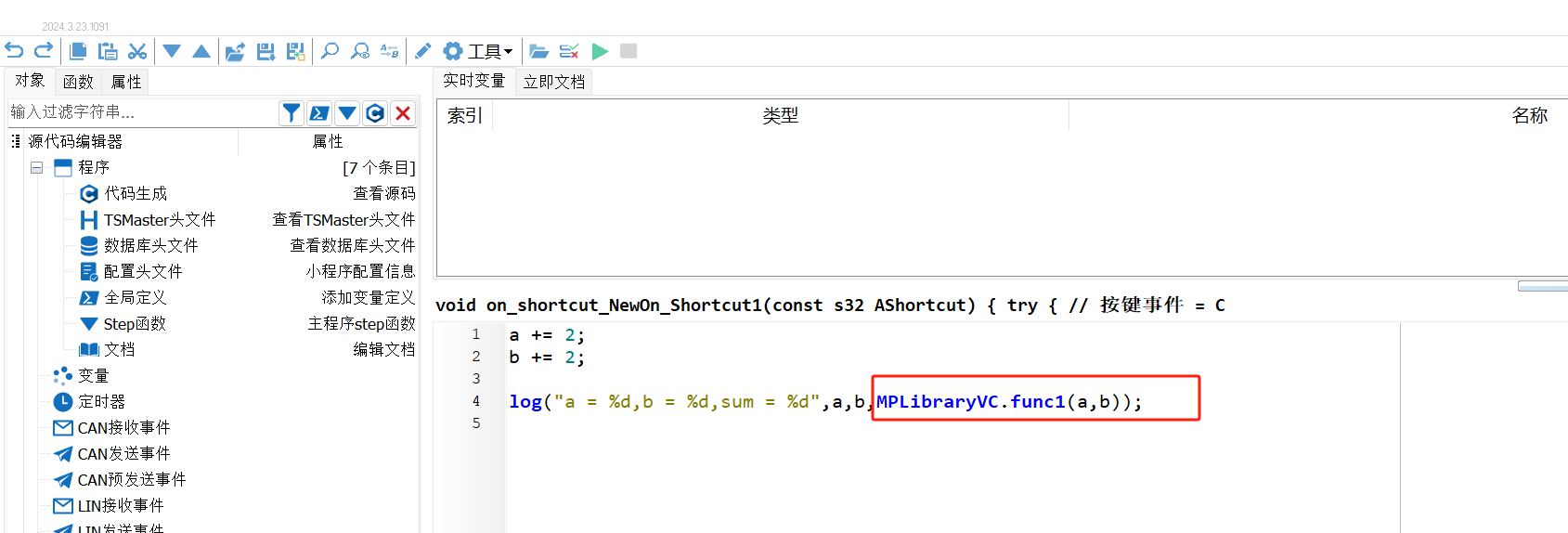
将生成的动态库直接拖入TSMaster软件中，在TSMaster小程序库中即可看到对应函数与说明，如下图。

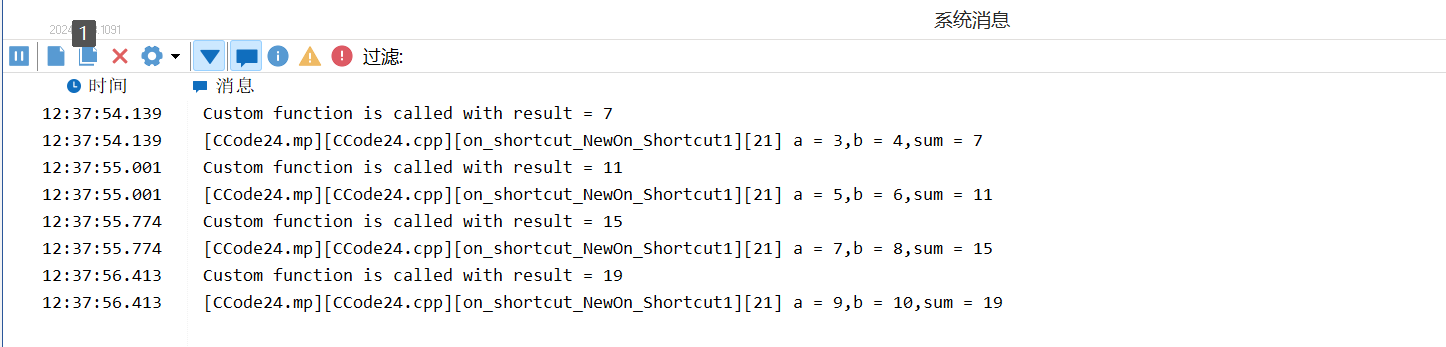


在TSMaster C小程序中即可直接使用DLL中的函数，不过在使用前需要勾选依赖DLL，在C小程序属性页面，选择依赖的小程序库，如下图所示。



在按键事件中，快速验证是否生效，如下图。



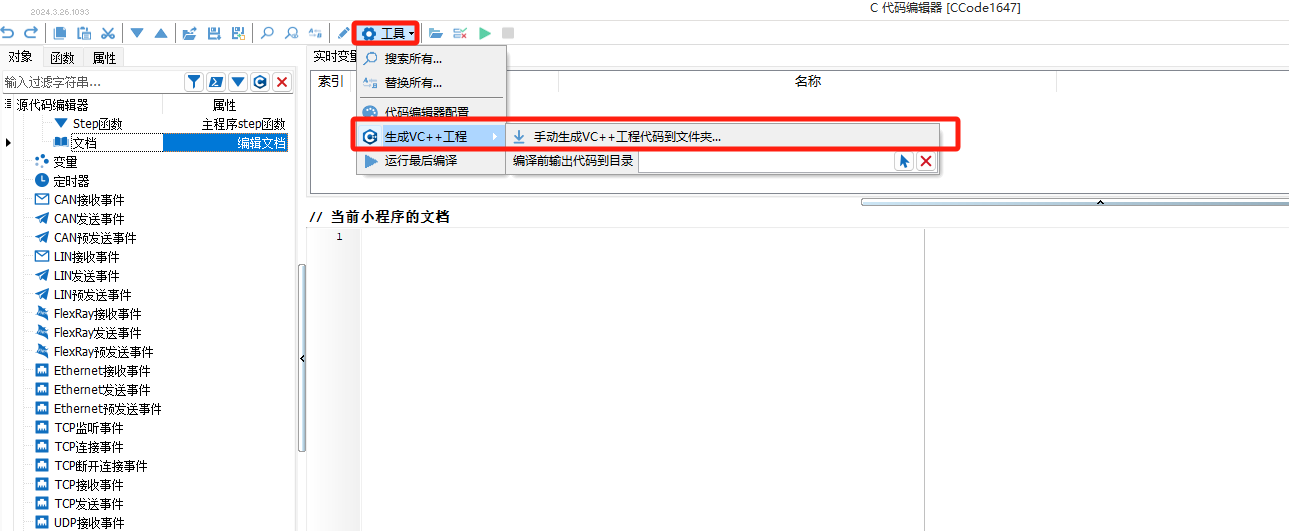


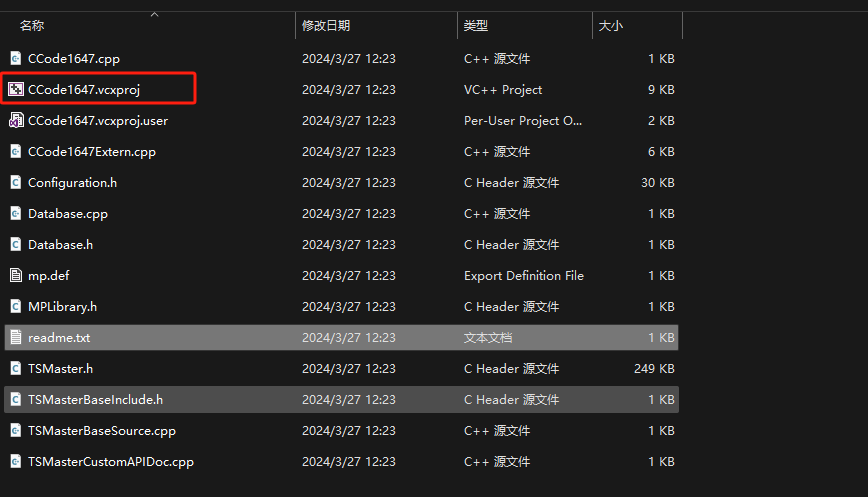
# Chapter 3 C小程序代码调试

目前TSMaster C小程序不支持单步调试，但TSMaster C小程序提供将小程序工程导出为VS工程，然后再VS工程中结合TSMaster进行单步调试。

## 导出C小程序工程

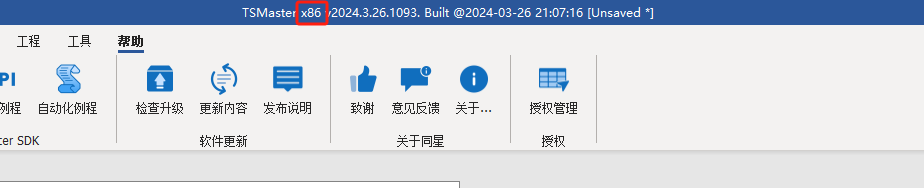
点击C小程序中的工具按钮，选择生成VC++工程->手动生成vc++工程到文件夹，选择文件夹后，将导出文件生成在选择的文件夹中，运行\*.vcxproj文件,如下图所示：



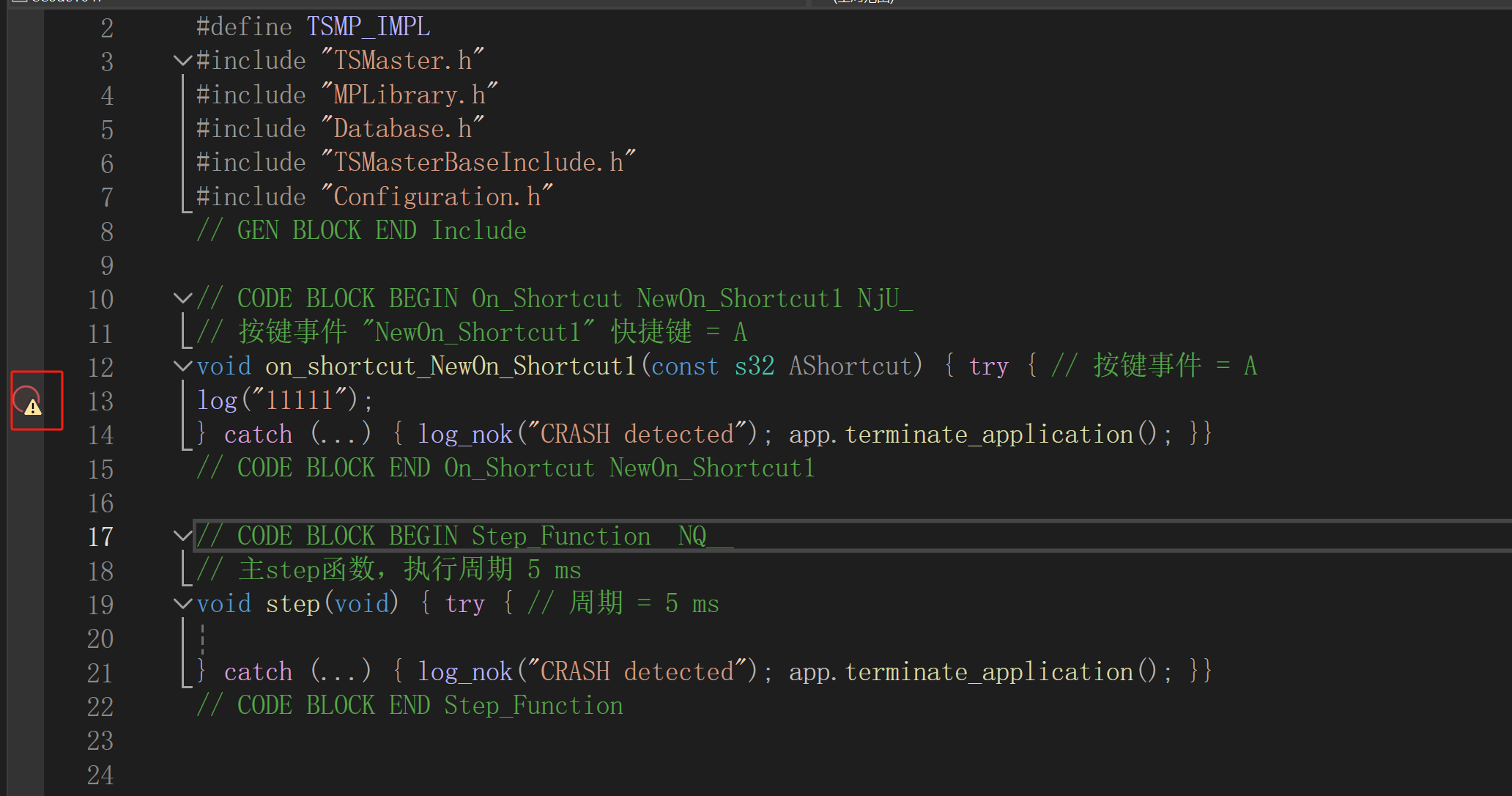


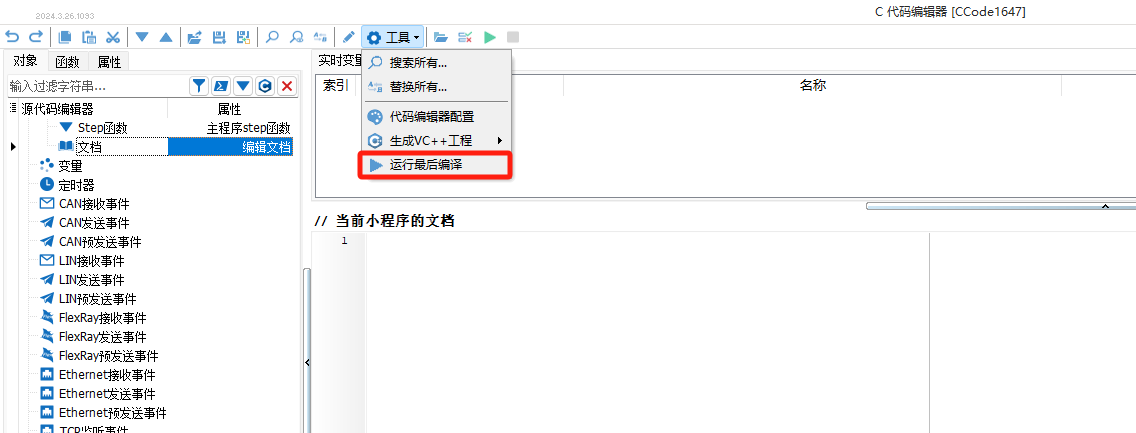
## 单步调试

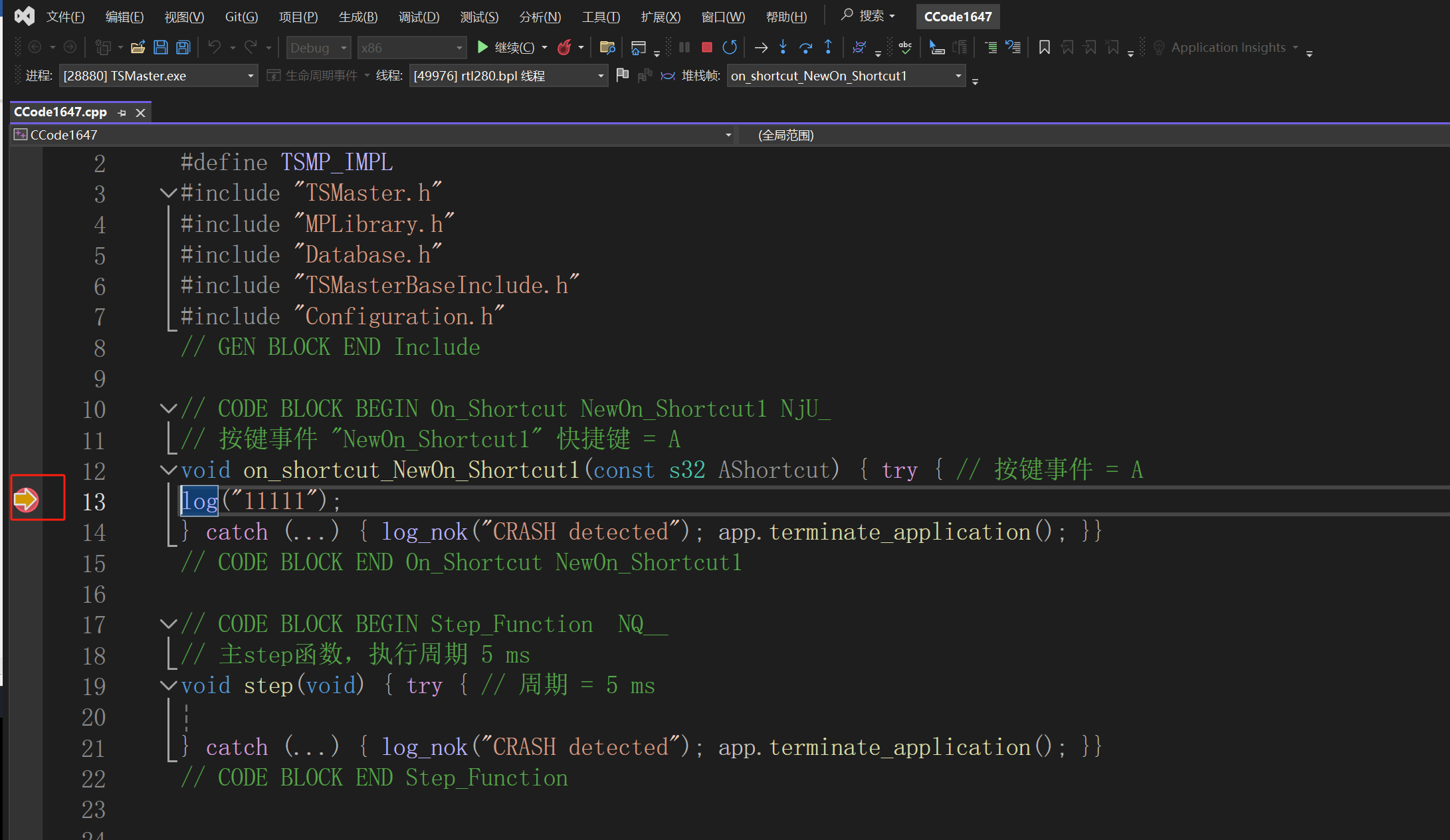
注意，编译工程，需要选择debug模式编译，并且需要指定x86/x64(根据运行的TSMaster是x86版本还是x64版本选择对应的模式进行编译)进行编译，通过TSMaster工程最上方的版本信息即可确定当前TSMaster是x86还是x64版本，如下图。



成功编译后，直接运行VS工程，此时断点显示无法进入，是因为没有跟TSMaster进行关联，因此无法调试，只需要在C小程序工具按钮中，点击运行最后编译，此时将发现VS工程中的断点可以进行调试。如下图所示：







如上图所示，对应的按钮事件执行，进入了VS工程中，进行单步调试成功。

# Chapter 4 特殊说明

特殊情况：

1.在使用TSMaster C小程序时，调用类似socket函数时，在编译时会存在函数未定义的问题，只需要在C小程序属性页面的附加编译指令加上 -lwsock32即可。