



Fremont Micro Devices

# FMDTouchTool 使用手册

Rev2.11

[www.fremontmicro.com](http://www.fremontmicro.com)



## 文档修改历史

日期	版本	描述
2021-12-30	2.10	更新 62F13x、62F23x、62F28x 库文件；
2022-04-8	2.11	1. 62F13x、62F23x、62F28x 新增加功能选项：输出原始值、低功耗、软件扫描； 2. 增加 FT32F0XX 范例演示说明

## 目 录

文档修改历史 .....	2
1 概述 .....	5
2 使用说明 .....	5
2.1 软件说明 .....	5
2.2 界面使用说明 .....	6
2.2.1 设置界面 .....	6
2.2.1.1 工程设置框 .....	6
2.2.1.2 MCU 型号选框 .....	6
2.2.1.3 按键选框 .....	7
2.2.1.4 功能选框 .....	7
2.2.1.5 其他选框 .....	7
2.2.1.6 芯片视图框 .....	7
2.2.1.7 资源使用提示框 .....	11
2.2.2 调试界面 .....	12
2.2.2.1 信息输出框 .....	12
2.2.2.2 初始化(不触摸)按钮 .....	12
2.2.2.3 参考电容值输入框 .....	13
2.2.2.4 按键状态与数据显示 .....	13
2.2.2.5 效果验证按钮 .....	14
2.2.2.6 组数据测试按钮 .....	15
2.2.2.7 数据显示方式选项 .....	15
2.2.2.8 数据显示窗口 .....	16
2.2.3 组数据调试界面 .....	16
2.2.3.1 按键分组设置 .....	17
2.2.3.2 组数据测试 .....	17
2.3 调试异常 .....	19
3 范例演示 .....	20
3.1 62F13x、62F23x、62F28x 使用范例 .....	20
3.1.1 按键应用 .....	20
3.1.1.1 工程创建 .....	20

3.1.1.2	测试验证.....	24
3.1.1.3	低功耗的测试验证 .....	25
3.1.1.4	生成应用代码.....	26
3.1.2	库应用说明.....	27
3.2	62F08x、62F211 使用范例 .....	28
3.2.1	按键使用 .....	28
3.2.1.1	工程创建.....	28
3.2.1.2	测试验证.....	29
3.2.1.3	生成应用代码.....	30
3.2.2	滑条、滑轮使用.....	31
3.2.2.1	工程创建.....	31
3.2.2.2	测试验证.....	33
3.2.2.3	生成应用代码.....	33
3.2.3	库应用说明.....	35
3.3	FT32F072xx、FT32F032xx 使用范例.....	36
3.3.1	按键使用 .....	36
3.3.1.1	工程创建.....	36
3.3.1.2	测试验证.....	39
3.3.1.3	生成应用代码.....	39
3.3.2	库应用说明.....	40
联系信息 .....		41

## 1 概述

FMDTouchTool 由辉芒微电子有限公司开发，是用来对 FMD 触摸芯片进行可视化测量开发的工具。其主要特点是可以自动采集、计算触摸阈值，降低触摸程序的开发门槛。

## 2 使用说明

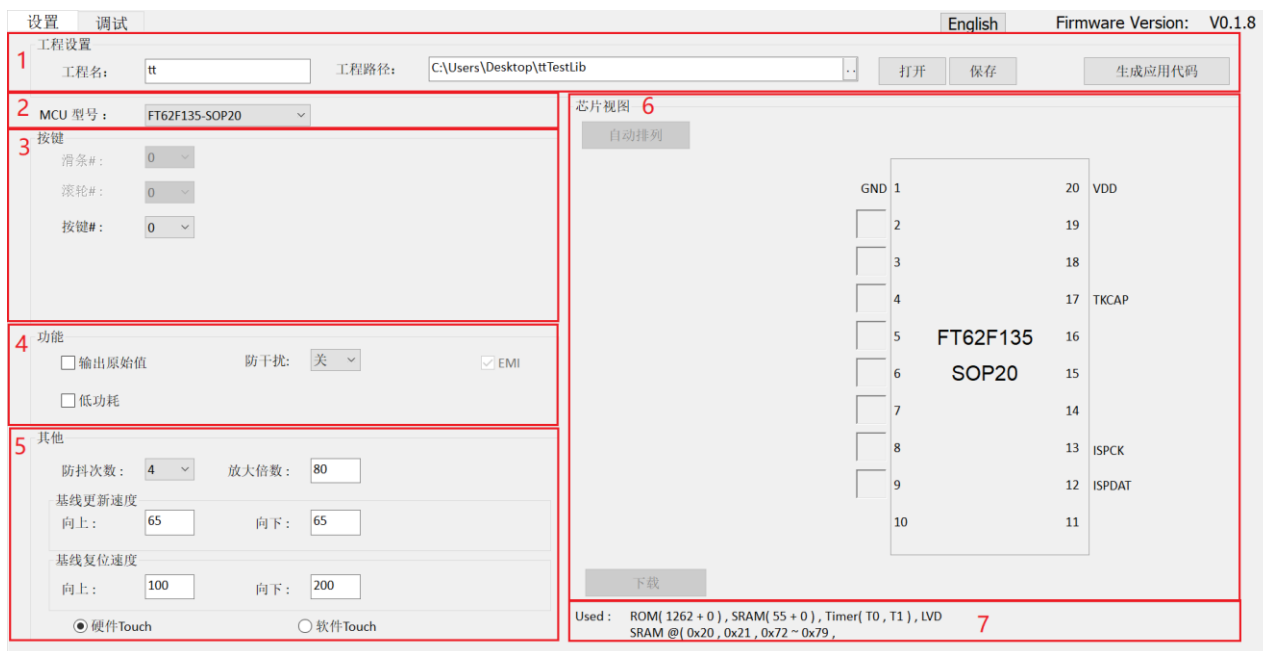
### 2.1 软件说明

打开 FMD 的 IDE 工具包，在工具包目录下找到 FMDTouchTool 应用程序，双击打开即可。

名称	类型	大小
data	文件夹	
doc	文件夹	
inc	文件夹	
testlib	文件夹	
Testtmp	文件夹	
Update	文件夹	
FMDIDE	应用程序	16,791 KB
FMDIDE_internal	应用程序	16,850 KB
FMDTouchTool	应用程序	2,515 KB
OnlineUpdate	应用程序	88 KB
readme	文本文档	1 KB
SciLexer.dll	应用程序扩展	629 KB
Scintilla.dll	应用程序扩展	310 KB
setting.bd	BD 文件	1 KB
SYS	配置设置	1 KB
tlib.dll	应用程序扩展	142 KB

## 2.2 界面使用说明

### 2.2.1 设置界面



#### 2.2.1.1 工程设置框

1. 工程名设置框  
工程名支持中文。
2. 工程路径显示框、“浏览”按钮  
当设置好工程名后，通过“浏览”按钮选择好工程路径则会生成工程设置文件“.tch”；工程路径支持中文。
3. “打开”按钮  
用于选择已有的工程设置文件“.tch”。如果当前有工程设置操作，在点击按钮时则会弹出一个提示信息，询问是否需要保存当前设置后，才会去选择打开文件。

名称	类型	大小
tt	配置设置	5 KB
tt.tch	TCH 文件	1 KB

4. “保存”按钮  
将当前设置参数保存到指定工程以及路径。
5. 生成应用代码按钮  
用于生成触摸库应用工程代码，用触摸调试工具在调试界面测试完所选的通道后，才可点击此按钮。

#### 2.2.1.2 MCU 型号选框

可以选择芯片型号及封装。

### 2.2.1.3 按键选框

1. 滑条#

滑条总个数最多可选两个；每个滑条占用四个触摸引脚；分辨率 16。

2. 滑轮#

滑轮总个数最多可选两个；每个滑轮占用四个触摸引脚；分辨率 32。

3. 按键#

根据目标板的设计选择按键个数。并支持多按键选择，其中：

- 多按键：同时生效按键个数。
- 主按键：多按键时主按键选择，选择此选项时多个按键按下但是此按键未被按下返回 0。
- 禁止按键：选择此选项时多个按键按下此按键同时被按下时返回 0。

### 2.2.1.4 功能选框

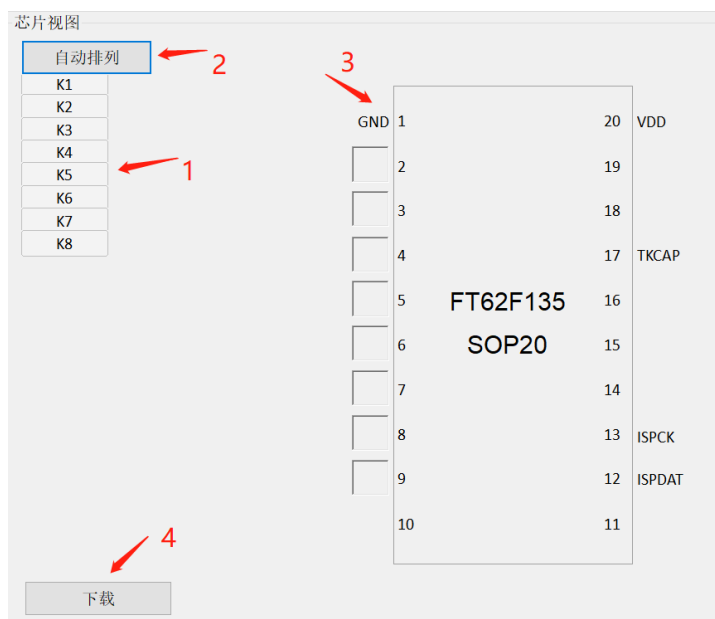
根据不同 MCU 型号，可选择的功能选框不同，详情请查 [范例演示](#) 中的详细设置说明。

### 2.2.1.5 其他选框

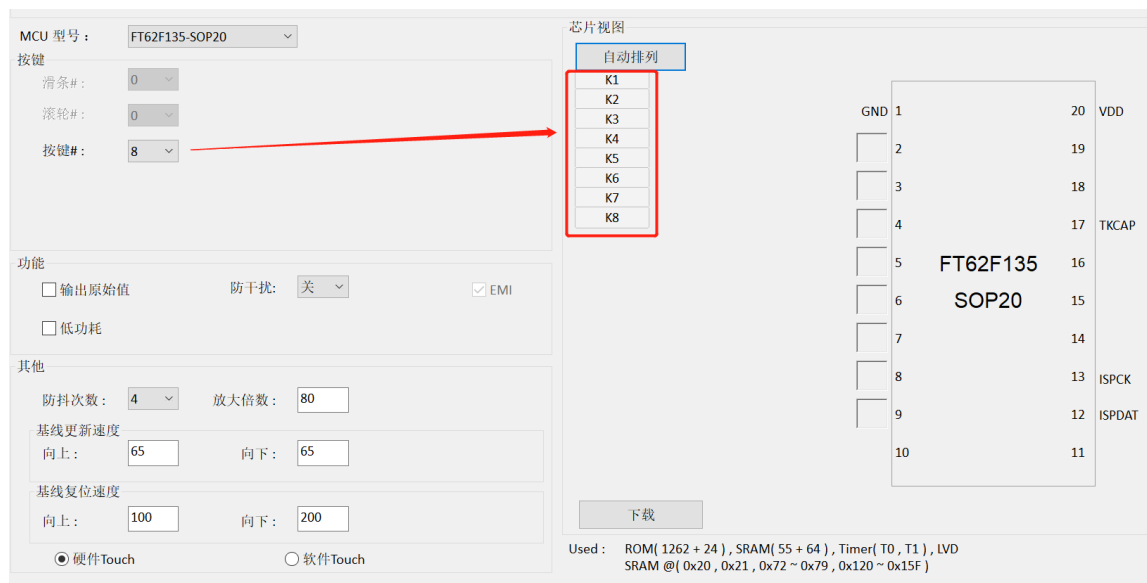
根据不同 MCU 型号，可选择设置的参数不同，详情请查 [范例演示](#) 中的详细设置说明。

### 2.2.1.6 芯片视图框

包括按键列表、自动排列按钮、芯片触摸引脚的位置图、下载按钮。

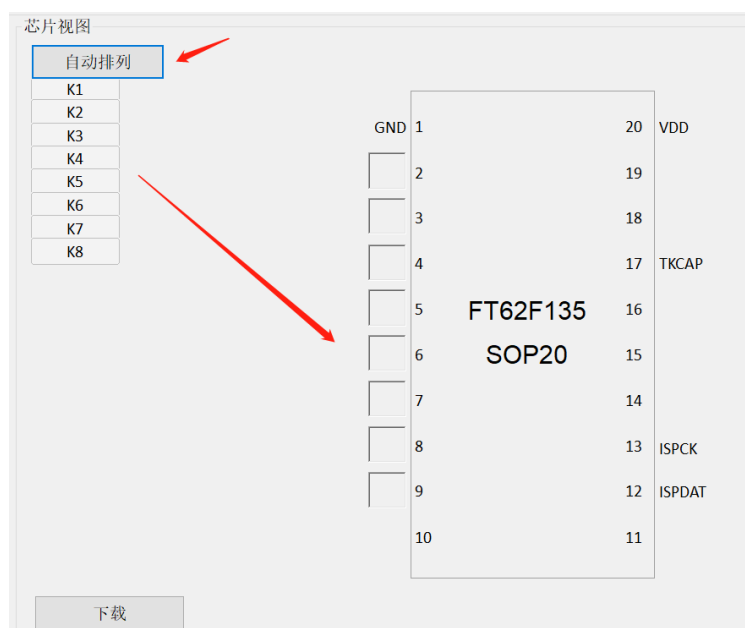


1. 按键列表：由设置的按键个数产生。



### 2. 自动排列按钮:

- 自动排列: 当按键列表未与芯片引脚配置时, 可将按键列表配置到对应的触摸引脚上。

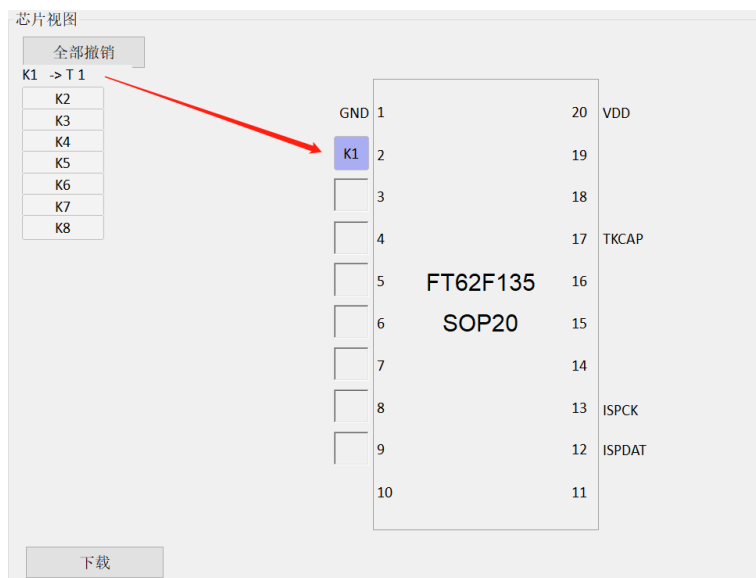


- 全部撤销: 只要有任何一个按键已与芯片引脚配置, 则可通过此按钮将已配置引脚全部撤销。





3. 芯片触摸引脚位置图：用于显示按键与引脚的配置关系，可用鼠标拖动按键，手动放置芯片引脚区域实现配置。

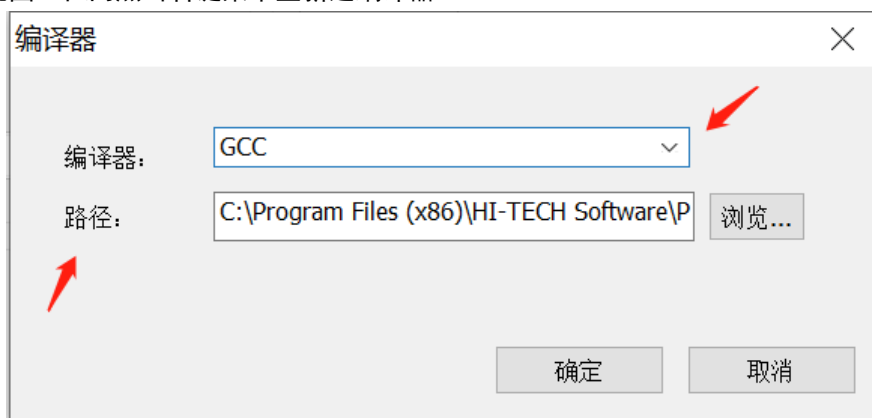


#### 4. 下载按钮

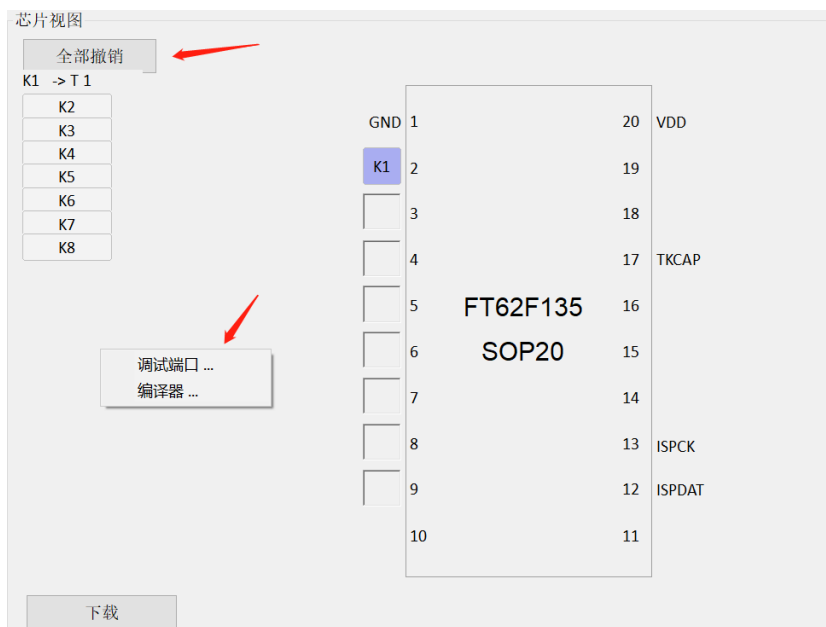
- 弹出“调试端口设置”框，选好调试引脚或自定义调试引脚，按下确定键；如果选择错误，会造成调试异常。
- 当勾选“不再弹出”选框后，不会自动弹出“调试端口设置”框，如需要更改调试端口可在“芯片视图”框内点击右键菜单选调试端口。



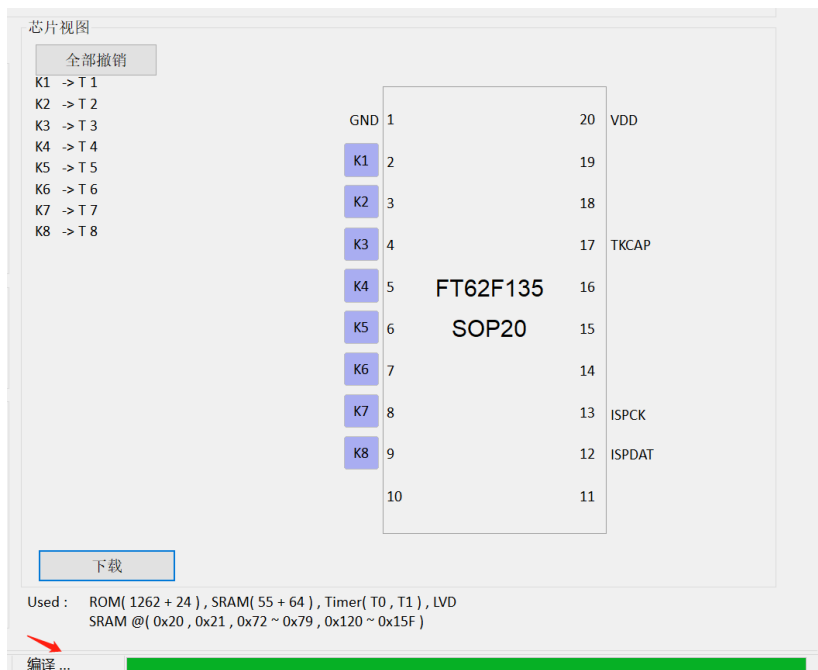
- 如果 IDE 未选择编译器，则会自动弹出“编译器设置”框，如需要更改编译器可在“芯片视图”框内点击右键菜单重新选编译器。



- 需要更改调试端口或更改编译器可在“芯片视图”框内点击右键菜单选择。

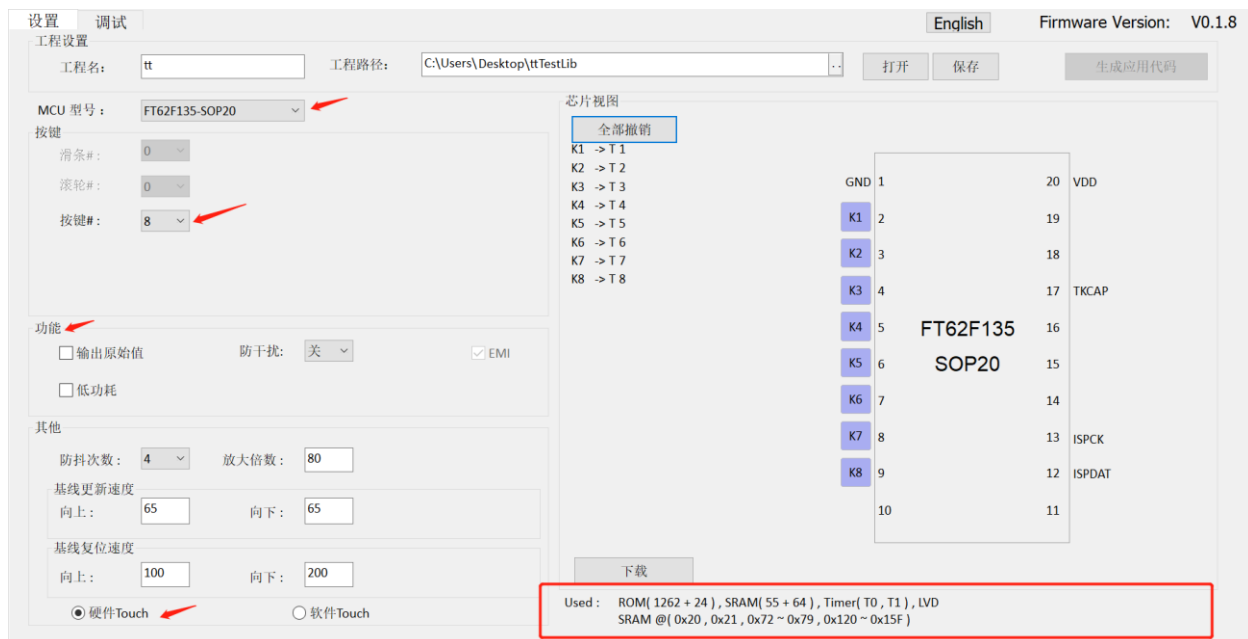


- 下载过程，将工程设置进行保存生成工程配置文件“.ini”，并把测试程序下载到目标板中；下载提示信息在如下图所示。如果下载失败，请确认调试工具与目标板的连接是否正确，或者是否为通讯线过长，通讯不稳定造成的。

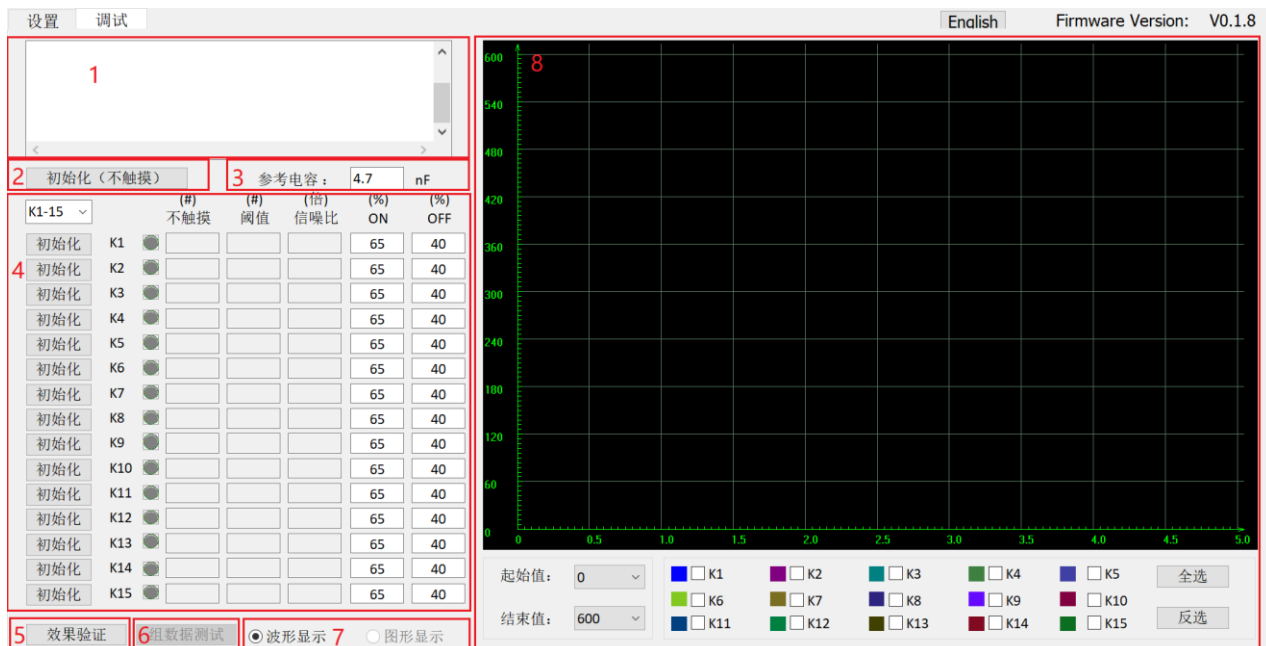


### 2.2.1.7 资源使用提示框

当选择好 MCU 类型，按键个数，功能选项以及扫描方式后，就可以计算出资源的使用情况：ROM、SRAM 使用大小、外设资源的使用以及 SRAM 指定地址的使用等。

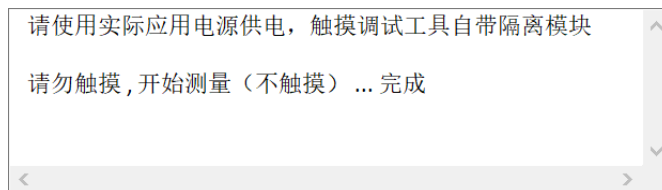


## 2.2.2 调试界面



## 2.2.2.1 信息输出框

用于显示进度提示、测试过程提示等信息。



## 2.2.2.2 初始化(不触摸)按钮

用于测试所有未触摸按键的按键值；如果已初始化完成，则会变成“重新初始化(不触摸)”按钮。



### 2.2.2.3 参考电容值输入框

输入 PCB 线路板所焊的电容值大小，按下“初始化(不触摸)”按钮后，测量出按键数据是否达标，不达标会在信息输出框内提示更换电容值信息。

请使用实际应用电源供电，触摸调试工具自带隔离模块

请勿触摸,开始测量（不触摸）...

参考电容不合理，请调整到 3.30nF完成

重新初始化（不触摸）

参考电容：5.6 nF

		(#) 不触摸	(#) 阈值	(倍) 信噪比	(%) ON	(%) OFF
测触摸	K1	774			65	40
测触摸	K2	734			65	40
测触摸	K3	640			65	40
测触摸	K4	681			65	40
测触摸	K5	645			65	40
测触摸	K6	703			65	40
测触摸	K7	756			65	40
测触摸	K8	833			65	40

### 2.2.2.4 按键状态与数据显示

用于显示各按键的测试状态、键值等信息。

1 K1-15

初始化	K1				65	40
初始化	K2				65	40
初始化	K3				65	40
初始化	K4				65	40
初始化	K5				65	40
初始化	K6				65	40
初始化	K7				65	40
初始化	K8				65	40
初始化	K9				65	40
初始化	K10				65	40
初始化	K11				65	40
初始化	K12				65	40
初始化	K13				65	40
初始化	K14				65	40
初始化	K15				65	40

2 3 4 5 6 7

1. 按键列表选择框：当按键个数超过 15Key 时，会弹出选择框，用于切换按键。

	K16-28	(#) 不触摸	(#) 阈值	(倍) 信噪比	(%) ON	(%) OFF
初始化	K16				65	40
初始化	K17				65	40
初始化	K18				65	40
初始化	K19				65	40
初始化	K20				65	40
初始化	K21				65	40
初始化	K22				65	40
初始化	K23				65	40
初始化	K24				65	40
初始化	K25				65	40

## 2. “初始化”、“测触摸”或“重测”按钮

此按钮有三种切换状态：

- “初始化”按钮，未进行初始化时则会提示先初始化。
- “测触摸”按钮，当初始化完成之后，未做触摸测试时则会提示做触摸测试。
- “重测”按钮，已做按键触摸测试，则提示可以重新测试阈值。

## 3. 按键状态显示

此提示灯在测试过程与效果验证过程中变化：

- 在测试过程中，绿灯闪动提示按键正在测试，如果测试数据有效则会显示绿灯，否则为灰色。
- 在效果验证中，红色表示按键按下，绿色按键未按下。

## 4. 不触摸(#)

显示初始化不触摸时的按键值。

## 5. 阈值(#)

由不触摸键值与触摸键值产生，触摸变化量大于此值认为有按键按下。

其公式为：阈值 = (不触摸键值 - 触摸键值) \* (触发点 ON) %

## 6. 信噪比(倍)

当前通道信噪比(建议大于 5)。当信噪比小于 5 时，其阈值会变红，按键效果显示为灰色，测试未通过。此项仅用于警示硬件没达到标准，不会影响触摸库的生成。

## 7. 触摸(ON%)、非触摸(OFF%)

触摸和非触摸触百分比的设定可以手动选择修改；修改百分比从而改变阈值大小，如下图所示：

其中，ON：推荐 65%，建议大于 55%， OFF：推荐 40%。

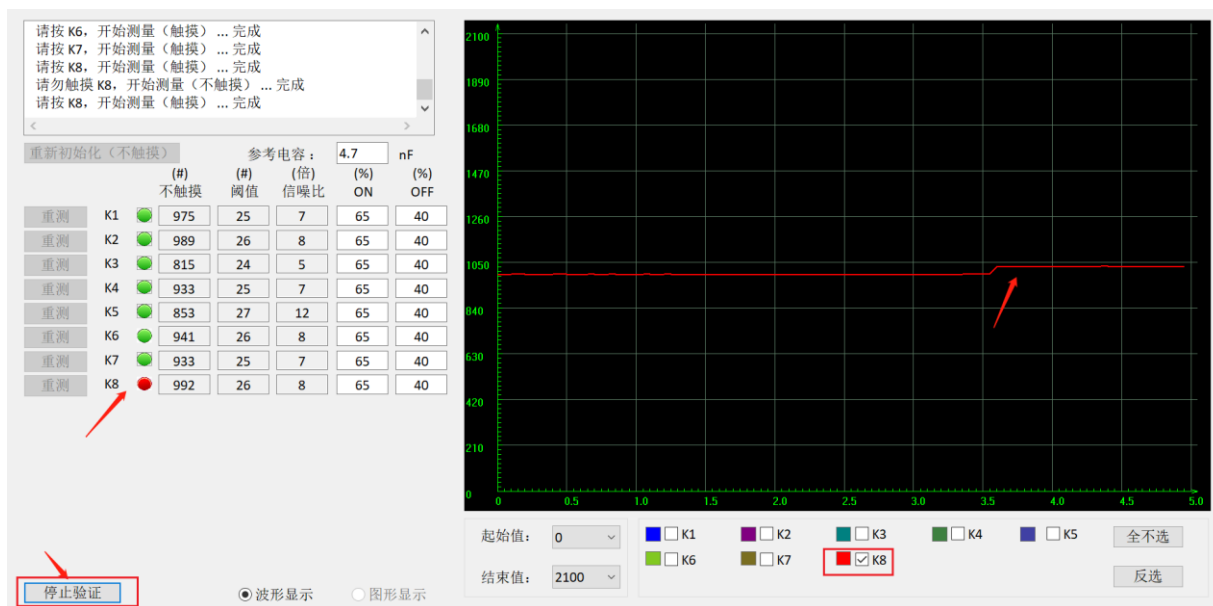
			(#) 不触摸	(#) 閾値	(倍) 信噪比	(%) ON	(%) OFF
重測	K1		975	25	7	65	40
重測	K2		989	26	8	65	40
重測	K3		815	24	5	65	40
重測	K4		933	25	7	65	40
重測	K5		853	27	12	65	40
重測	K6		941	26	8	65	40
重測	K7		933	25	7	65	40
重測	K8		992	26	8	65	40

			(#) 不触摸	(#) 閾値	(倍) 信噪比	(%) ON	(%) OFF
重測	K1		975	21	7	55	40
重測	K2		989	26	8	65	40
重測	K3		815	24	5	65	40
重測	K4		933	25	7	65	40
重測	K5		853	27	12	65	40
重測	K6		941	26	8	65	40
重測	K7		933	25	7	65	40
重測	K8		992	26	8	65	40

### 2.2.2.5 效果验证按钮

用于查看实时按键值的变化，数据会以图形或波形的方式在观察窗口中显示出来。被触摸按键的状态

显示指示灯会从绿色变为红色，如果未达到阈值指示灯为黄色。如下图所示：



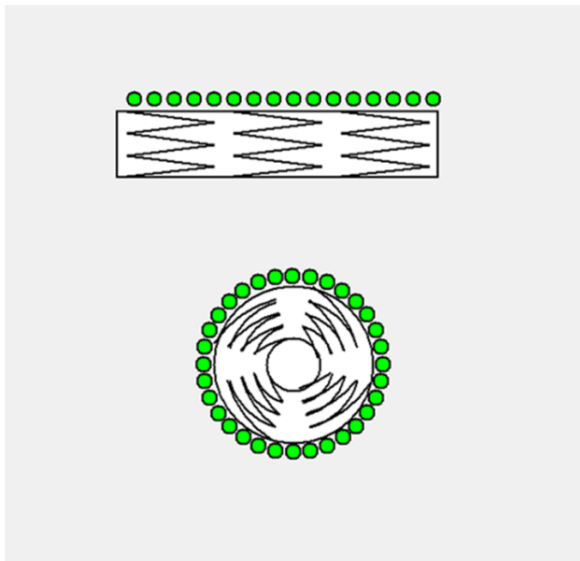
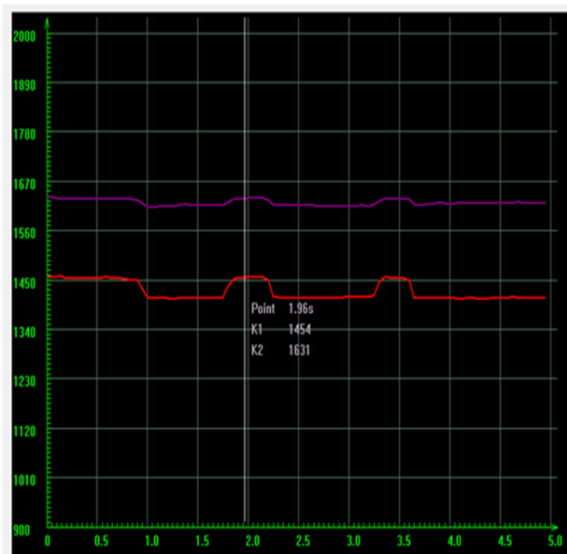
### 2.2.2.6 组数据测试按钮

组数据测试用于测试低功耗数据，只有在设置界面勾选了“低功耗”选项才会显示；并且所有按钮需测试完成才能调试使用。这个按钮也是组数据与原数据调试界面的切换按钮，操作详情查阅 [组数据调试界面](#) 说明。



### 2.2.2.7 数据显示方式选项

数据将以波形或图形的方式显示在观察窗口中，滑轮和滑条可选图形显示。



### 2.2.2.8 数据显示窗口

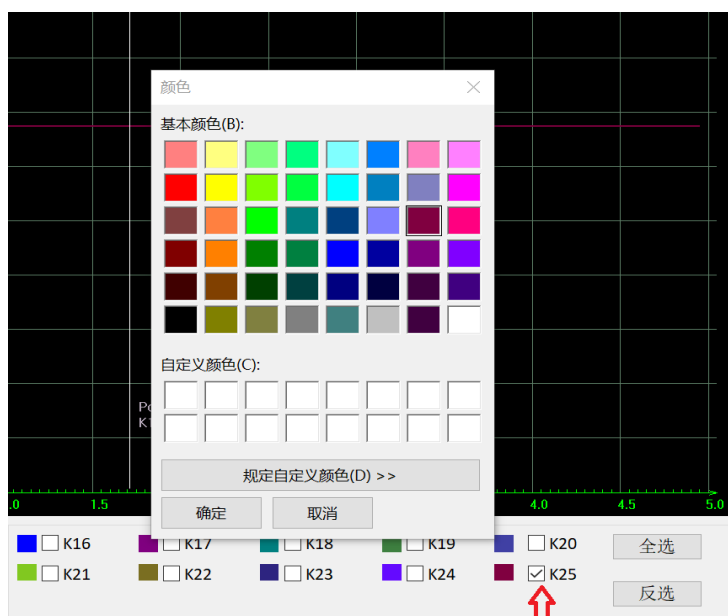
用于显示按键实时数据的曲线或图形。

#### 1. 纵坐标起始、结束值设置

在测试过程或效果验证中，如果观察图形窗口并未看到键值波形，可能因为 Y 轴设置不对；则可根据其值设置 Y 轴的参数，即调节起始值、结束值；如果一直显示 0 表示没有键值输出或错误数据。

#### 2. 选择显示的键值曲线

勾选对应按键，其键值会在观察窗口中显示数据曲线；双击颜色框可以根据喜好选择曲线颜色。



### 2.2.3 组数据调试界面

进入组数据调试界面，要先将按键分组设置后才能进行组数据测试；组数越多，扫描时间越长，功耗越大。



### 2.2.3.1 按键分组设置

#### 1. 自动生成

由工具根据所有按键的测试结果自动生成了分组信息，用户可以直接按下确定按钮。



#### 2. 手动组合

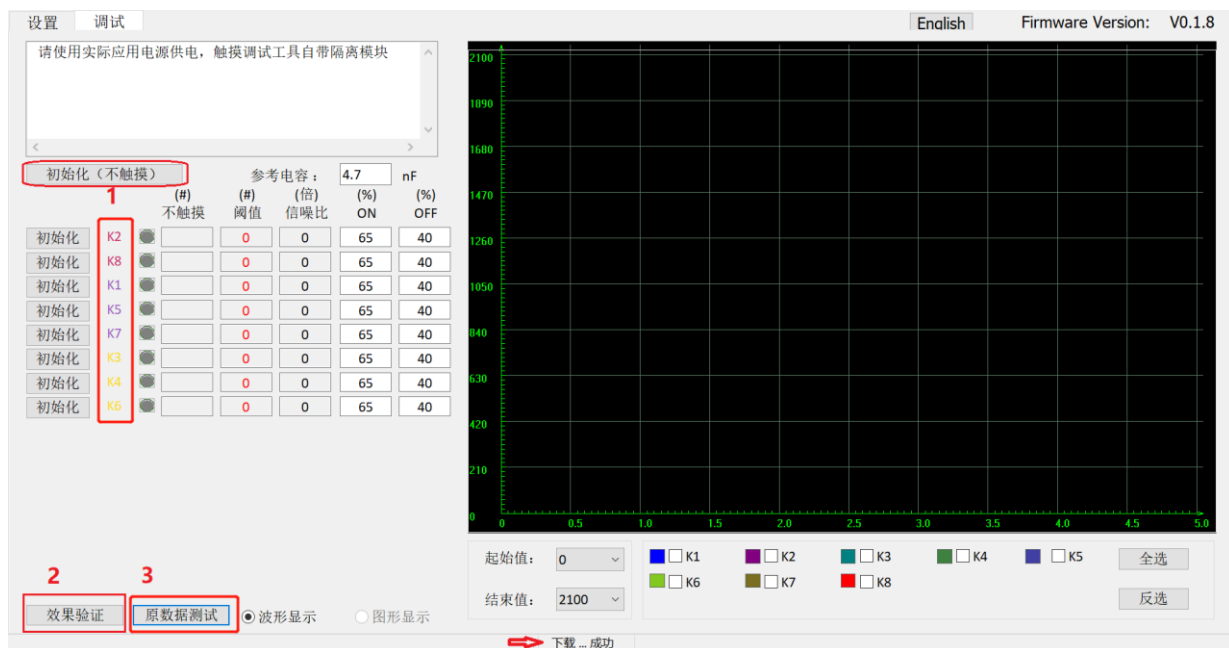
用户可以取消组合，用手动拖到各组的方式组合，但组合必须要达到并联参数设置的条件，然后按确定按钮。



### 2.2.3.2 组数据测试

切换进入组数据调试界面时，工具会自动下载组数据测试程序，下载成功后可以进入测试，相同按键

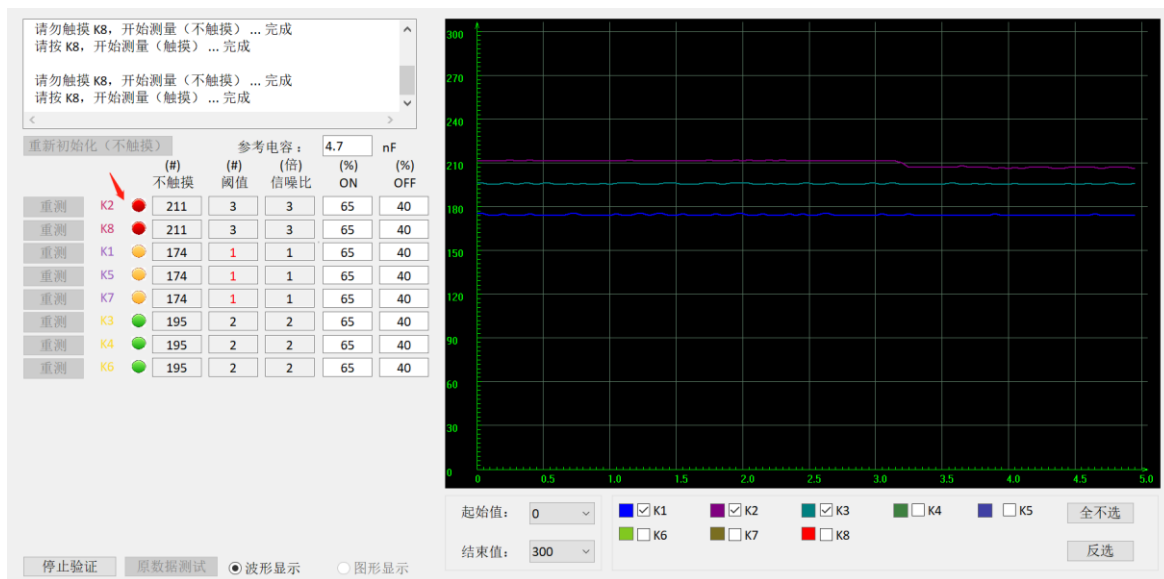
组的颜色相同。



1. 按下“初始化(不触摸)”按钮，工具会自动计算出按键组的不触摸值、阈值、信噪比值，并在信息框中显示：预估低功耗电流值；操作完此步骤可以返回设置界面生成应用代码。



2. 可进行效果验证。



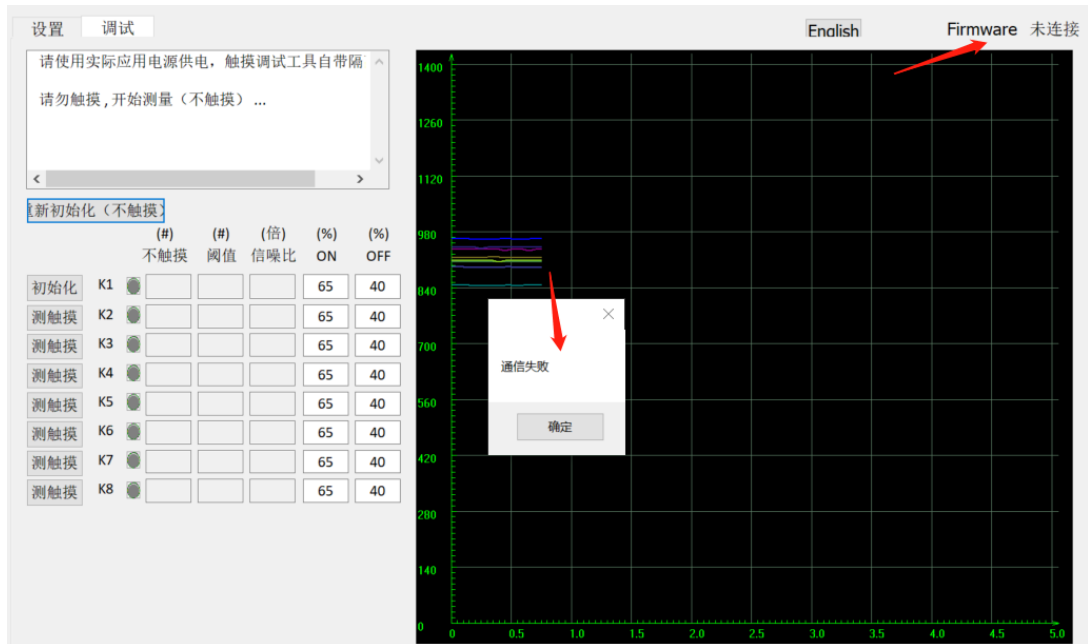
3. 可通过“原数据测试”按钮切回原数据测试调试界面进行测试。

## 2.3 调试异常

1. 提示“读取数据超时，请检查是否连接调试板”，请确认设置的调试引脚是否正确；或者重新将测试程序下载到目标板。



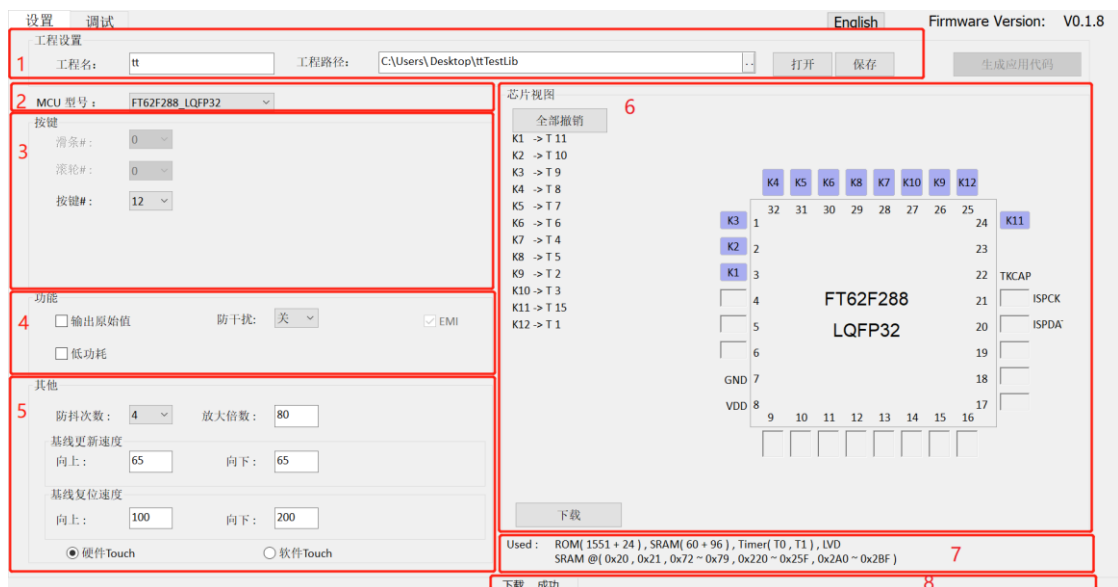
2. 调试过程中提示“通讯失败”，则是出现通讯中断，请重新连接调试工具。



### 3 范例演示

#### 3.1 62F13x、62F23x、62F28x 使用范例

##### 3.1.1 按键应用



##### 3.1.1.1 工程创建

1. 设置工程名称，选择保存路径。
2. MCU 型号  
FT62F288-LQFP32
3. 按键：  
普通触摸按键：12key
4. 功能选项

功能

☐ 输出原始值      防干扰: 主动 ▾      ☒ EMI

☒ 低功耗      静态采样率: 8 ▾      ☐ 快速扫键

- 输出原始值: 为无算法库的选择, 只输出采样值; 采用的是硬件扫描, 此库应用只需设好按键个数以及设置对应的按键 IO 端口 (无需做调试), 就可以生成应用代码。

工程设置

工程名: tt      工程路径: C:\Users\Desktop\TestLib      打开      保存      生成应用代码

MCU 型号: FT62F288\_LQFP32

按键

滑条#: 0 ▾

滚轮#: 0 ▾

按键#: 12 ▾

功能

☒ 输出原始值      防干扰: 关      ☒ EMI

☒ 低功耗

其他

防抖次数: 4      放大倍数: 80

基线更新速度

向上: 65      向下: 65

基线复位速度

向上: 100      向下: 200

☐ 硬件Touch      ☐ 软件Touch

芯片视图

全部撤销

K1 → T11  
K2 → T10  
K3 → T9  
K4 → T8  
K5 → T7  
K6 → T6  
K7 → T4  
K8 → T5  
K9 → T2  
K10 → T3  
K11 → T15  
K12 → T1

FT62F288 LQFP32

下载

Used: ROM( 603 + 36 ), SRAM( 18 + 24 ), Timer( T0, T1 ), LVD, CycleTime( 3200 us,

- 防干扰: 为了提高触摸在特定干扰下的稳定性的功能。三个选项: 关闭、主动跳频、被动跳频。

功能

☐ 输出原始值      防干扰: 主动 ▾      ☒ EMI

☒ 低功耗      静态采样率:      ☐ 快速扫键

关  
主动跳频  
被动跳频

- 快速扫键: 用于多键时加快扫描速度。  
当按键个数  $\geq 12$ Key 时, 并且防干扰打开, 此项才可选择。

按键

滑条#: 0 ▾

滚轮#: 0 ▾

按键#: 12 ▾

功能

☐ 输出原始值      防干扰: 主动 ▾      ☒ EMI

☒ 低功耗      静态采样率: 8 ▾      ☐ 快速扫键

当按键个数  $< 12$ Key 时, 即便防干扰打开, 此选项也不可选。

按键

滑条#: 0

滚轮#: 0

按键#: 11

功能

☐ 输出原始值

防干扰: 主动

☒ EMI

☒ 低功耗

静态采样率: 8

- EMI: 默认选择。
- 低功耗: 如需实现低功耗, 必选。
- 静态采样率: 为低功耗的睡眠时间, 与功耗电流有关系。

功能

☐ 输出原始值

防干扰: 主动

☒ EMI

☒ 低功耗

静态采样率: 8

☐ 快速扫键

## 5. 其他

其他

防抖次数: 4

放大倍数: 80

基线更新速度

向上: 65

向下: 65

基线复位速度

向上: 100

向下: 200

☒ 硬件Touch

☐ 软件Touch

- 防抖次数: 按键生效防抖次数, 取值范围 1~6。
  - 放大倍数: 数据扩大倍数, 取值范围 30~250。
  - 基线更新速度: 数值设置越大, 更新速度越慢, 取值范围 1~127。
  - 基线复位速度: 数值设置越大, 更新速度越慢, 取值范围 1~255。
  - 硬件 Touch: 采用的是硬件扫描方式, 其会占用 TIM0,TIM1 定时器资源。
  - 软件 Touch: 采用的是软件扫描方式, 不会占用 TIM0,TIM1 定时器资源。
- 此选项不支持低功耗, 防干扰不可选, 默认支持防干扰, 如下图所示:

功能

☐ 输出原始值

防干扰:

关

☒ EMI

☐ 低功耗

其他

防抖次数:

4

放大倍数:

80

基线更新速度

向上:

65

向下:

65

基线复位速度

向上:

100

向下:

200

☐ 硬件Touch
 ☒ 软件Touch

### 6. 芯片引脚配置

根据目标板上的按键定义与芯片触摸引脚对应配置。

芯片视图

全部撤销

K1 -> T 11

K2 -> T 10

K3 -> T 9

K4 -> T 8

K5 -> T 7

K6 -> T 6

K7 -> T 4

K8 -> T 5

K9 -> T 2

K10 -> T 3

K11 -> T 15

K12 -> T 1

K4

K5

K6

K8

K7

K10

K9

K12

K3

K2

K1

K11

32

31

30

29

28

27

26

25

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

24

23

22

21

20

19

18

17

FT62F288

LQFP32

GND

VDD

TKCAP

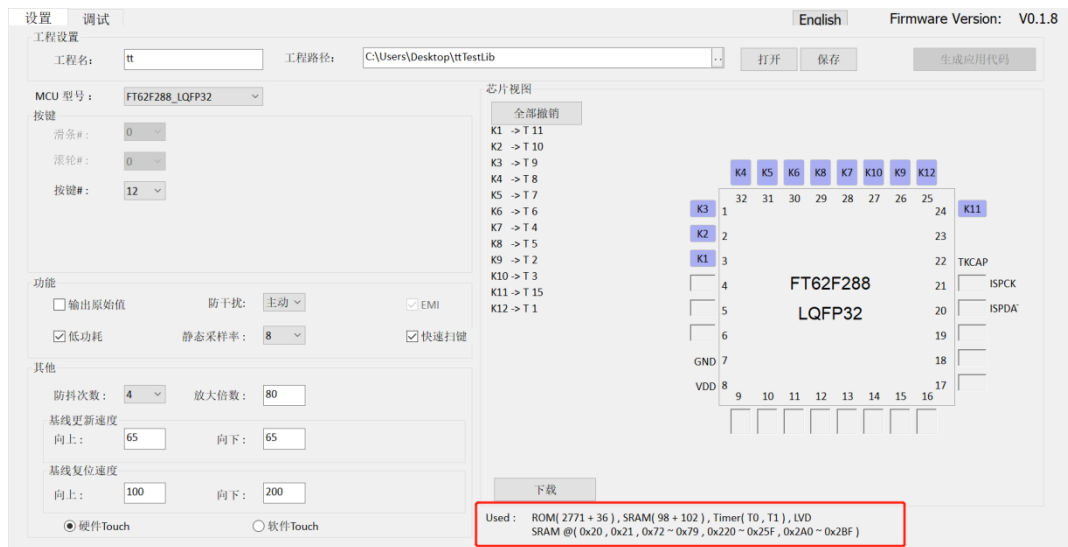
ISPCK

ISPDA

下载

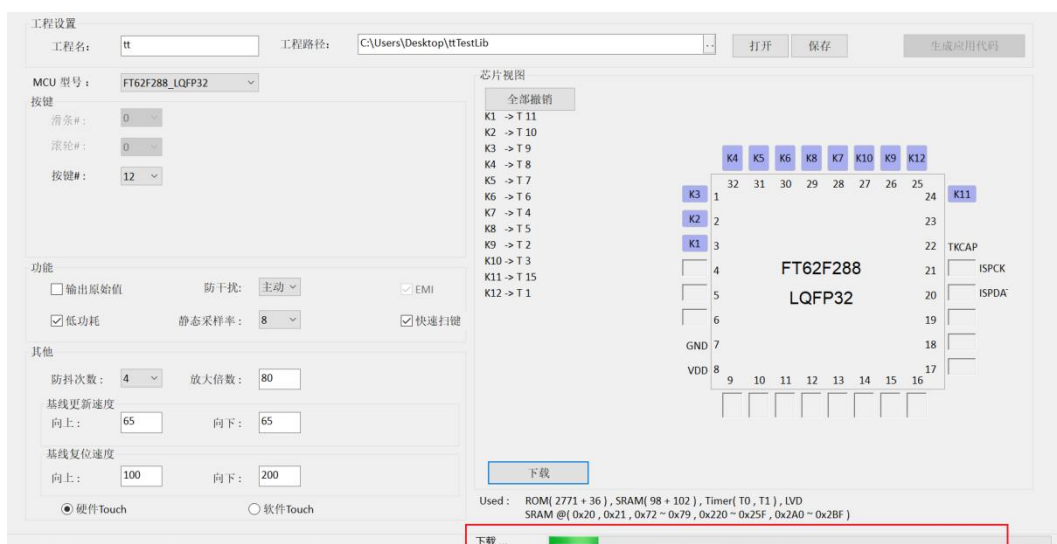
### 7. 资源使用

功能选项不同，按键个数不同其资源使用都不相同。



## 8. 下载按钮

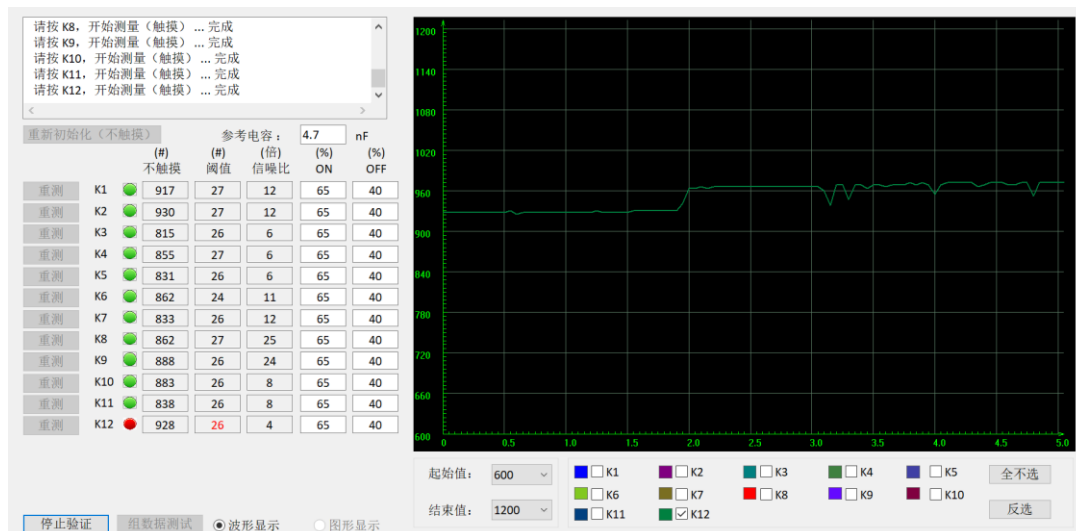
将工程设置进行保存并下载测试程序到目标板中；如果下载失败，请确认调试工具与目标板的连接是否正确；或者是否为通讯线过长，通讯不稳定造成的。



### 3.1.1.2 测试验证

切换至调试界面，按下“初始化(不触摸)”按钮，并测试按键完成，再进行效果验证，如下图所示：





### 3.1.1.3 低功耗的测试验证

当选择了低功耗选项，按“组数据测试”进入低功耗的测试验证。

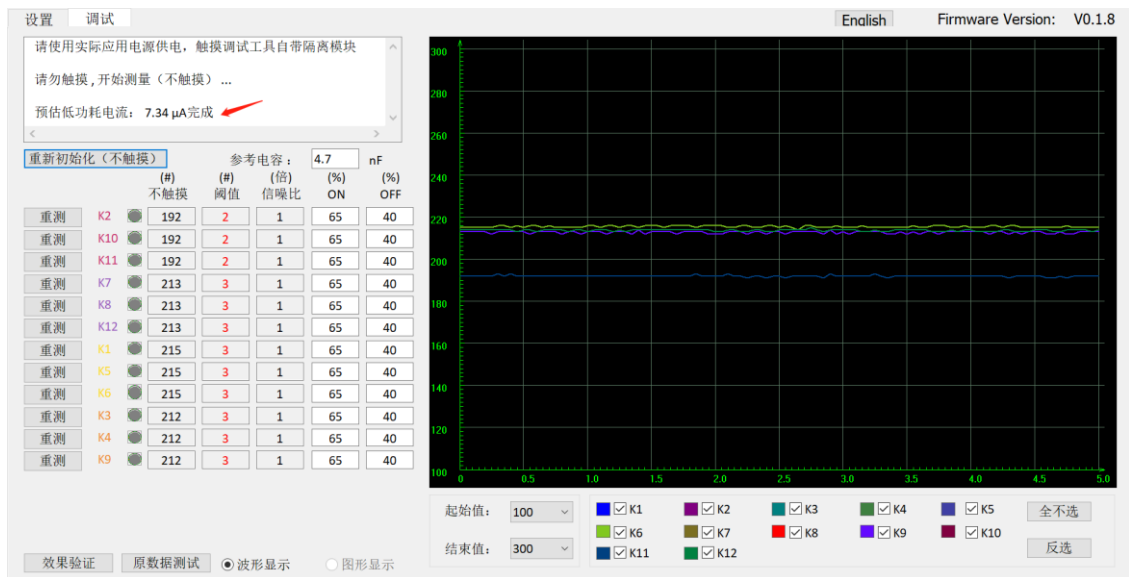
#### 1. 按键分组设置

自动生成，按下“确定”按钮。



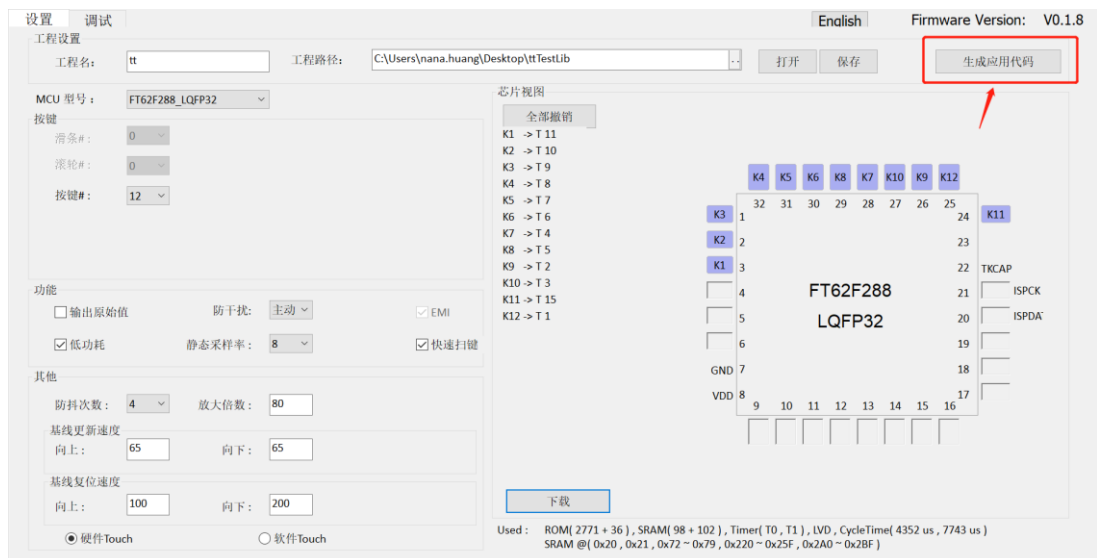
#### 2. 测试验证

按下“初始化(不触摸)”按钮，生成数据并预估出功耗电流。



### 3.1.1.4 生成应用代码

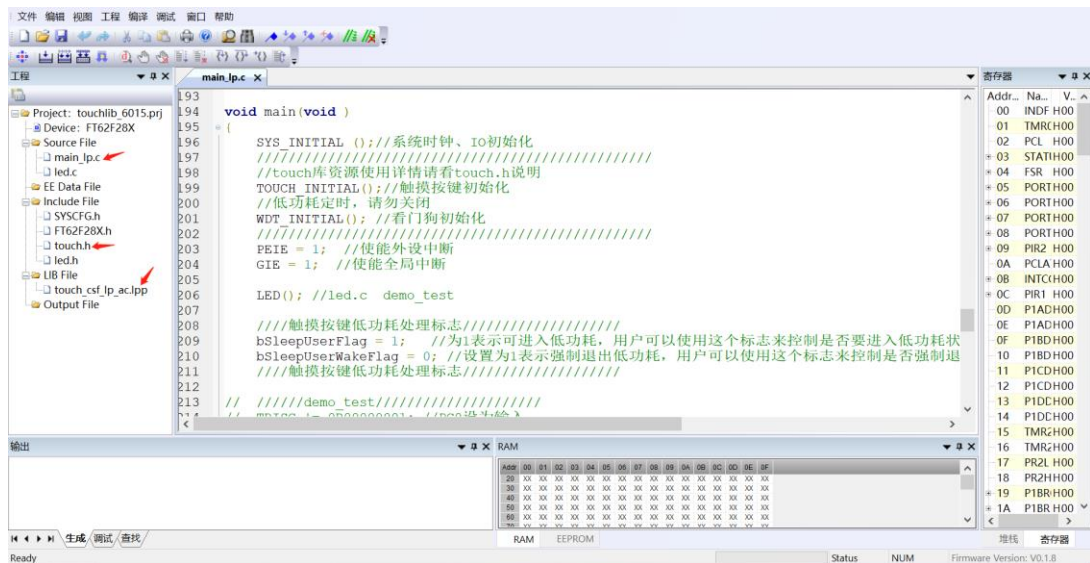
返回设置界面，点击生成应用代码。



1. 在所设置的工程路径下生成一个应用代码文件 touchlib\_xxxx (例:touchlib\_6015);

名称	类型	大小
touchlib_6015	文件夹	
tt	配置设置	6 KB
tt.tch	TCH 文件	1 KB

2. 打开工程



- main.c(main\_lp.c 低功耗)为主函数文件。
- touch\_xxx.lpp 为触摸库，根据功能选项选择不同的库文件。
- touch.h 为触摸库使用的头文件，其部分参数就是由调试工具产生的。

### 3.1.2 库应用说明

注：详细的库应用请查阅相关的 MCU 的库应用说明书。

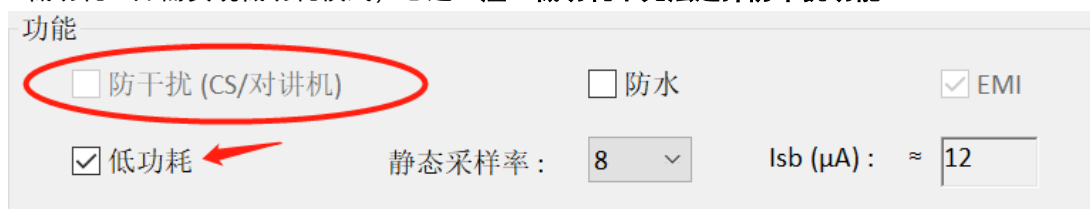
## 3.2 62F08x、62F211 使用范例

### 3.2.1 按键使用



#### 3.2.1.1 工程创建

1. 设置工程名称、选择保存路径。
2. MCU 型号  
选择以 FT62F087 为例。
3. 按键  
普通触摸按键数量：8，使用触摸通道分别为 1、2、3、4、5、6、7、8。  
多按键个数：2。
4. 功能选项
  - 防干扰：如需过 CS，必选。
  - 防水：FT62F08x 为硬件防水功能，默认选择。
  - EMI：默认选择。
  - 低功耗：如需实现低功耗模式，必选。注：低功耗下无法选择防干扰功能。



5. 其他选项
  - 防抖次数：按键生效防抖次数。
  - 采样频率：1 秒钟按键扫描次数。
  - 长按保护时间：按键有效时长，按键按下超过此时间，触摸自动复位，按键失效。无长按按键时建议选择 4s，有长按按键时建议根据长按时长，自定义长按保护时间。

- ## 6. 芯片引脚配置

芯片视图

全部撤销

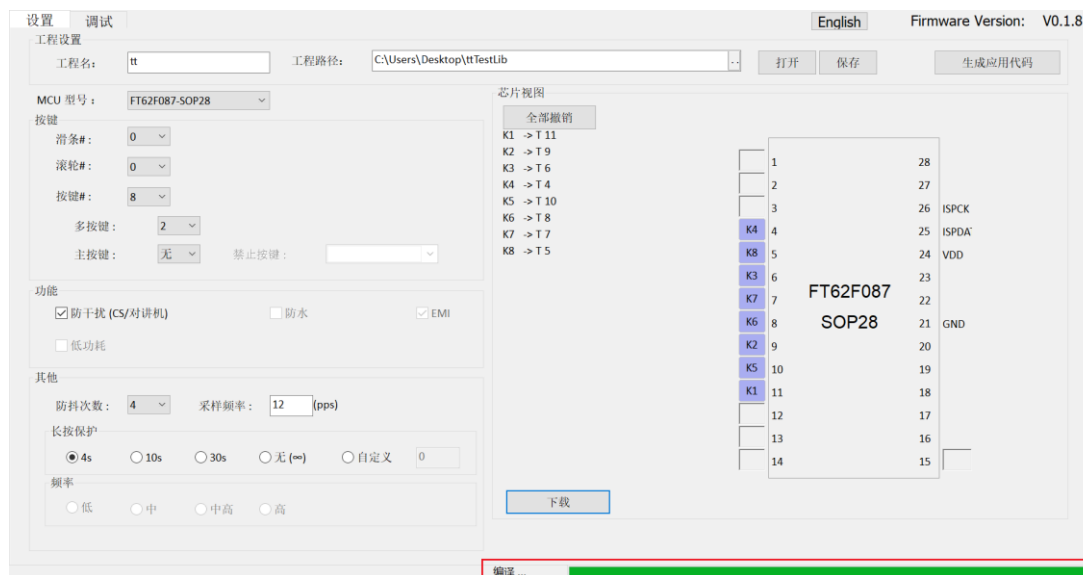
K1 -> T 11  
K2 -> T 9  
K3 -> T 6  
K4 -> T 4  
K5 -> T 10  
K6 -> T 8  
K7 -> T 7  
K8 -> T 5

1 28  
2 27  
3 26 ISPCK  
K4 4 25 ISPDAT  
K8 5 24 VDD  
K3 6 23  
K7 7 22  
K6 8 21 GND  
K2 9 20  
K5 10 19  
K1 11 18  
12 17  
13 16  
14 15

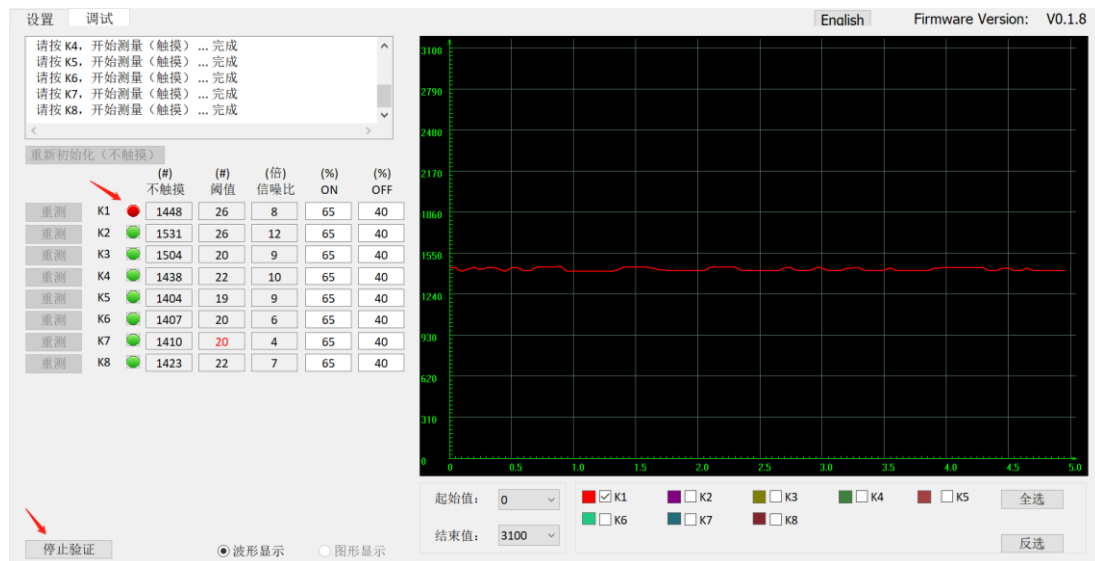
FT62F087  
SOP28

下载

- 点击“下载”按钮，将工程设置进行保存并下载测试程序到目标板中；如果下载失败，请确认调试工具与目标板的连接是否正确；或者是否为通讯线过长，通讯不稳定造成的。



切换置调试界面，按下“初始化(不触摸)”按钮，并测试按键完成，再进行效果验证，如下图所示：



### 3.2.1.3 生成应用代码

返回“设置”界面，点击“生成应用代码”。



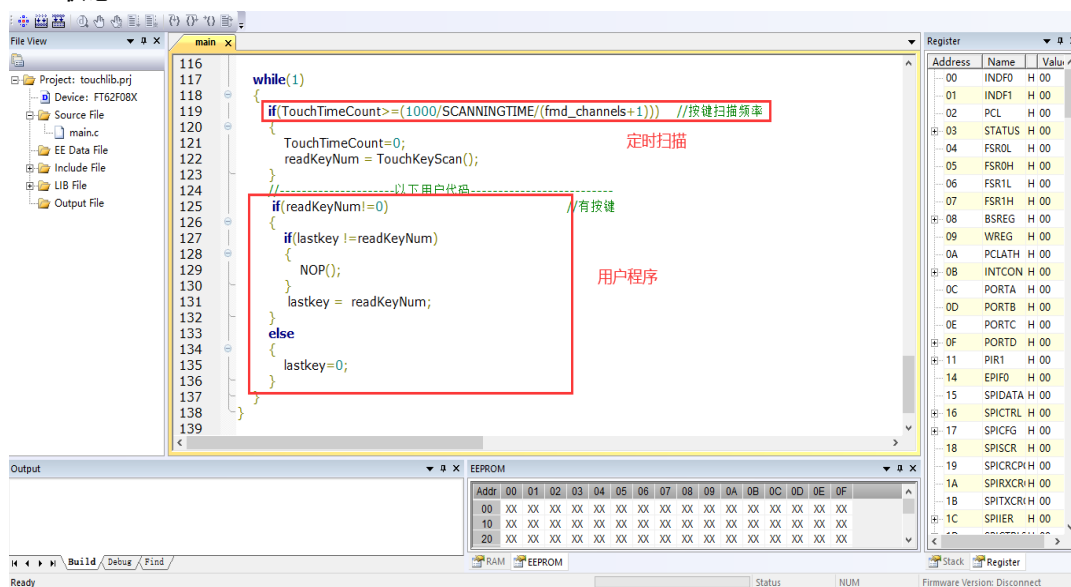
1. 在所设置的工程路径下生成一个应用代码文件 touchlibxx (例:touchlib06);

名称	类型	大小
touchlib06	文件夹	
tt	配置设置	6 KB
tt.tch	TCH 文件	1 KB

2. 打开工程

- 用户程序处加入按键判断代码，编写用户程序(定时扫描部分请勿修改，否则会影响触摸效果)。
- 低功耗程序增加了看门狗唤醒部分程序代码，用户选择合适的时间进入睡眠程序，并通过看门狗定时唤醒，确定是否有按键按下，如果有按键按下则跳入主函数执行，否则一直在看门狗睡眠

状态。



### 3.2.2 滑条、滑轮使用



#### 3.2.2.1 工程创建

1. 设置工程名称、选择保存路径。
2. MCU 型号  
选择以 FT62F087 为例。
3. 滑轮触摸通道为 1、2、3、4；  
滑条触摸通道为 5、6、7、8；  
即每个滑轮或者滑条占用四个触摸通道。
4. 功能选项
  - 防干扰：不支持过 CS，不可选。

功能

☐ 防干扰 (CS/对讲机) ☐ 防水 ☒ EMI

☐ 低功耗

- 防水：FT62F08x 为硬件防水功能，可选。

- EMI：默认选择。

- 低功耗：如需实现低功耗，必选。

#### 5. 其他选项

- 防抖次数：按键生效防抖次数。

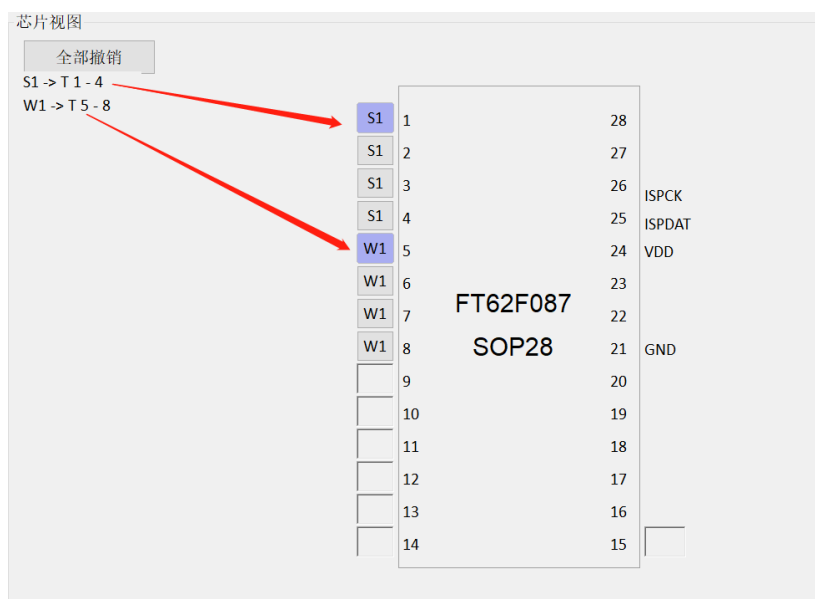
- 采样频率：1 秒钟按键扫描次数。

- 长按保护时间：按键有效时长，按键按下超过此时间，触摸自动复位，按键失效。无长按按键时建议选择 4s，有长按按键时建议根据长按时长，自定义长按保护时间。

- 频率：单频段按键扫描频率选择(FT62F08x)，可以手动选择按键振荡频率的大小，选择抗干扰或低功耗时，自动选择此选项失效(不建议手动选择此选项)。

#### 6. 芯片引脚配置

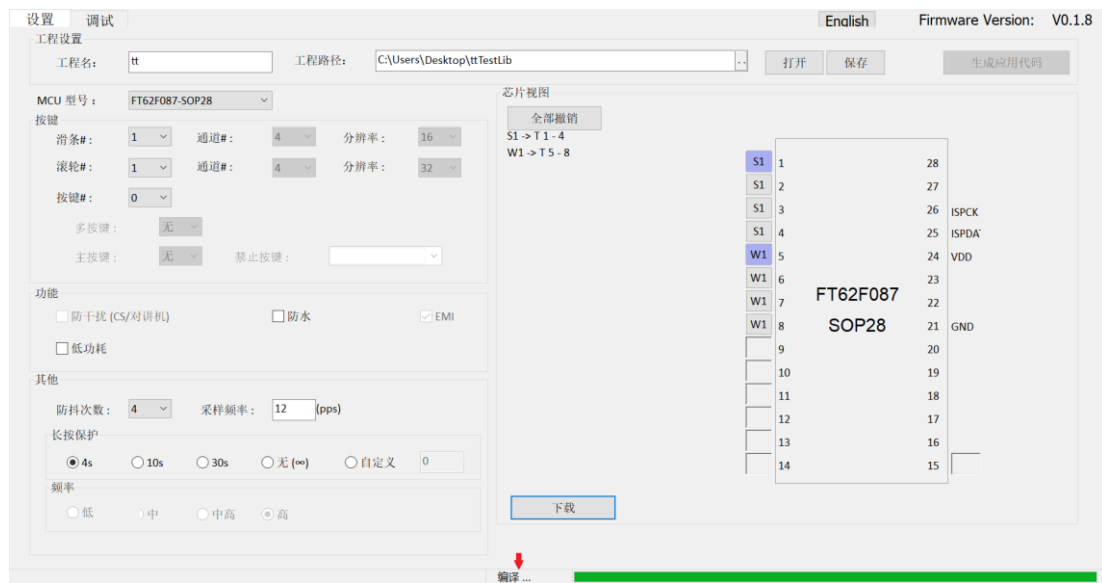
根据目标板上的按键定义与芯片触摸引脚对应配置。



#### 7. 下载按钮

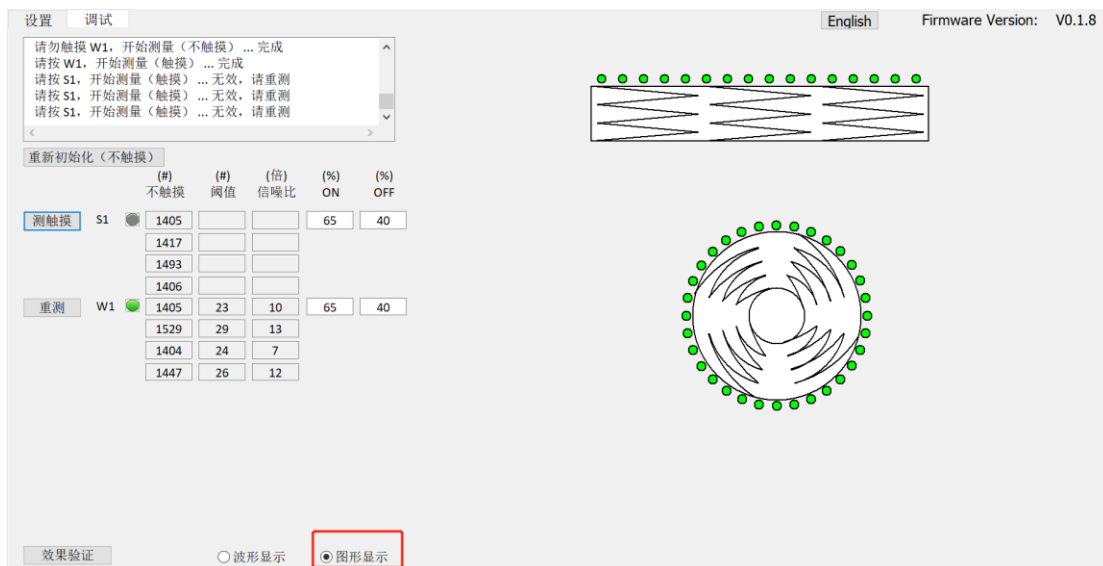
点击“下载”按钮，将工程设置进行保存并下载测试程序到目标板中；如果下载失败，请确认调试工具与目标板的连接是否正确；或者是否为通讯线过长，通讯不稳定造成的。





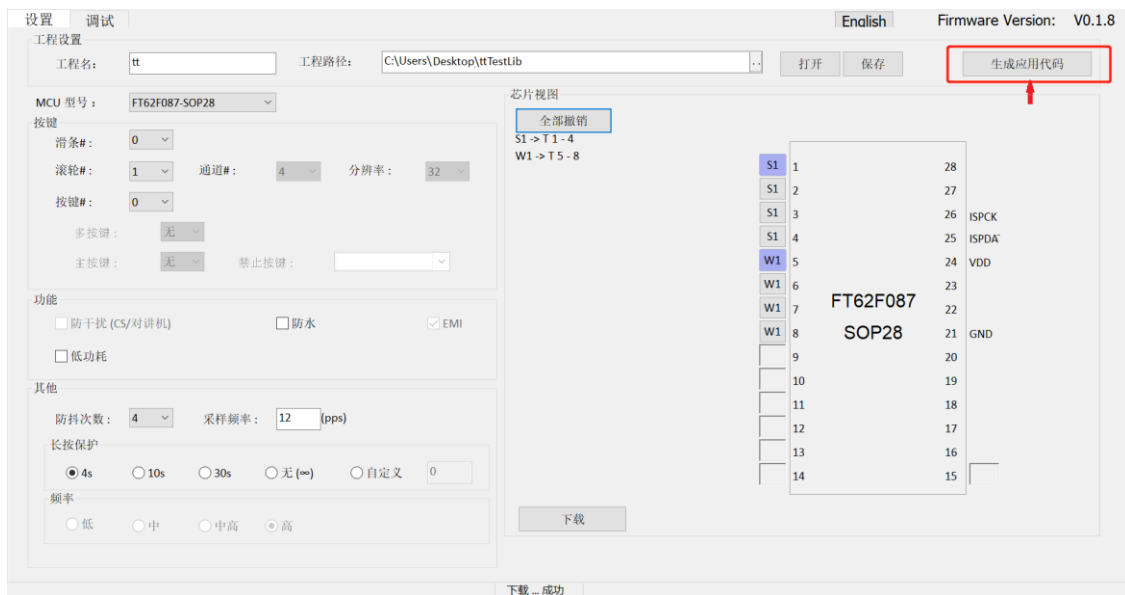
### 3.2.2.2 测试验证

切换至调试界面，按下“初始化(不触摸)”按钮，并测试所有按键，再进行效果验证，如下图所示：



### 3.2.2.3 生成应用代码

返回设置界面，点击生成应用代码。

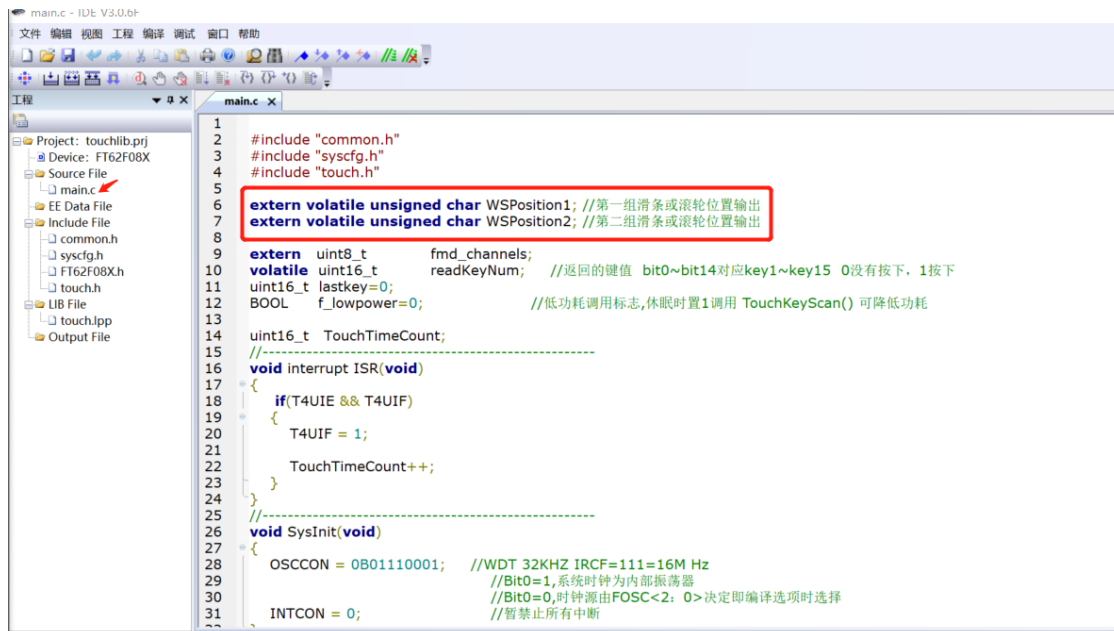


1. 在所设置的工程路径下生成一个应用代码文件 touchlib06 (例:touchlib06);

名称	类型	大小
touchlib06	文件夹	
tt	配置设置	6 KB
tt.tch	TCH 文件	1 KB

2. 打开工程

在调用库文件后访问 WSPosition1, WSPosition2, 寄存器及可获得当前滑轮或者滑条位置信息。



### 3.2.3 库应用说明

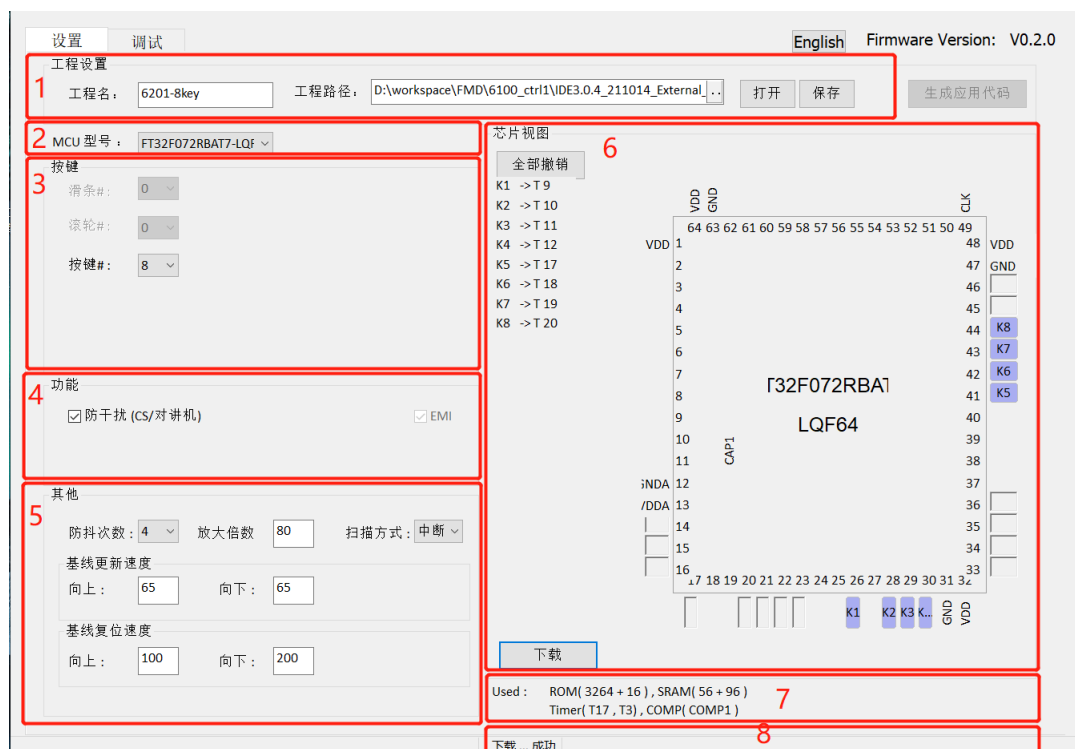
```

//定义所使用的按键 Bit 0~14 对应 key1~key15, 0 不使用, 1 使用
#define IO_TOUCH_ATTR 0x00ff
//单位(秒), 长按保护时间设置, 0 为关闭长按保护功能
#define LONGKEYPROTECT 4
//防抖次数连续多少次按下才认为按键按下
#define ANTISHAKETIMES 1
//1 秒扫描按键的次数
#define SCANNINGTIME 12
//(FT62F08X)频率选择, 此参数请勿修改
#define CSTYPE 4
//多按键个数(程序中最多支持多少个按键同时按下, 超过此数量按键按下按键返回值为 0)
#define MAXIMUMNUMBEROFKEYS1 1
//主按键(多按键个数大于 1 此数据有效, 0 关闭主按键功能任意小于等于多按键数量的按键按下有效, 其他值, 当选择的按键按下且按下按键数量为多按键选择个数是才返回按键值)
#define PRIMARYKEY1 1
//禁止按键(主按键有效时, 此数据有效, 0 不选择 其他值: 对应的禁止按键按下时, 多按键返回 0)
#define FORBIDDENKEY1 0x0000
//触摸低功耗时看门狗开门狗唤醒时间, 比如选择 8, 看门狗一秒钟唤醒 8 次, 约为 128ms 唤醒一次。
#define WAKEUPFREQUENCY 8
//(FT62F08X)是否开启硬件防水 0 不开 1 开
#define ANTI_WATER 0
//KEY ON 触摸触发百分比可手动选择修改(越小触摸越灵敏, 推荐 65%, 建议大于 55%)。
#define KEYn_ON 65
//KEY OFF 触摸放手触发百分比可手动选择修改(修改时需小于 KEYn_ON 推荐值 40%)
#define KEYn_OFF 40
//滑轮滑条选择(bit1bit0:第一组滑条滚轮选择;bit3bit2:第二组滑条滚轮选择;00 不用,01 滚轮,10 滑条)。
#define WHEELADDSILER 0x06
//第一组滑条或滚轮使用按键
const uint8_t FWHEELSILERB[4] = {0,1,2,3};
//第二组滑条或滚轮使用按键
const uint8_t SWHEELSILERB[4] = {4,5,6,7};

```

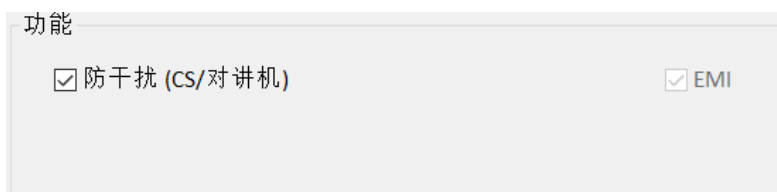
### 3.3 FT32F072xx、FT32F032xx 使用范例

#### 3.3.1 按键使用



##### 3.3.1.1 工程创建

1. 设置工程名称，选择保存路径。
2. MCU 型号  
FT32F072RBA1-LQFP64
3. 按键：  
普通触摸按键：8key
4. 功能选项
  - 防干扰：为了提高触摸在特定干扰下的稳定性的功能。



- 快速扫键：用于多键时加快扫描速度。  
当按键个数  $\geq 12$ Key 时，并且防干扰打开，此项才可选择。

按键

滑条#: 0

滚轮#: 0

按键#: 12

功能

☒ 防干扰 (CS/对讲机)
 ☐ 快速扫键
 ☒ EMI

当按键个数 <12Key 时，即便防干扰打开，此选项也不可选。

按键

滑条#: 0

滚轮#: 0

按键#: 8

功能

☒ 防干扰 (CS/对讲机)
 ☐ 快速扫键
 ☒ EMI

- EMI：默认选择。

### 5. 其它

其他

防抖次数: 4

放大倍数: 80

扫描方式: 中断

基线更新速度

向上: 65

向下: 65

基线复位速度

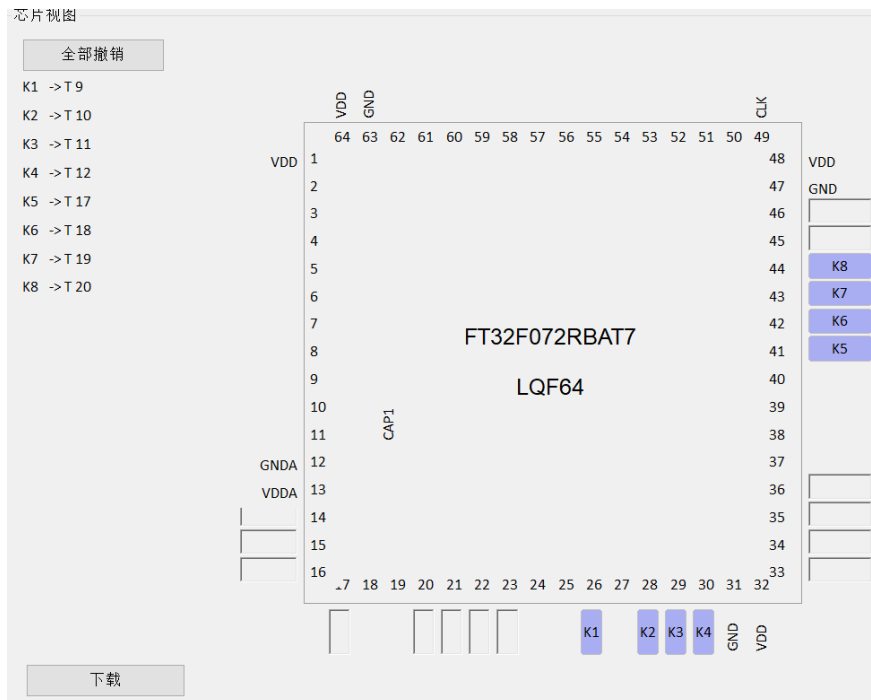
向上: 100

向下: 200

- 防抖次数：按键生效防抖次数，取值范围 1~6。
- 放大倍数：数据扩大倍数，取值范围 30~250。
- 扫描方式：中断或者轮询。
- 基线更新速度：数值设置越大，更新速度越慢，取值范围 1~127。
- 基线复位速度：数值设置越大，更新速度越慢，取值范围 1~255

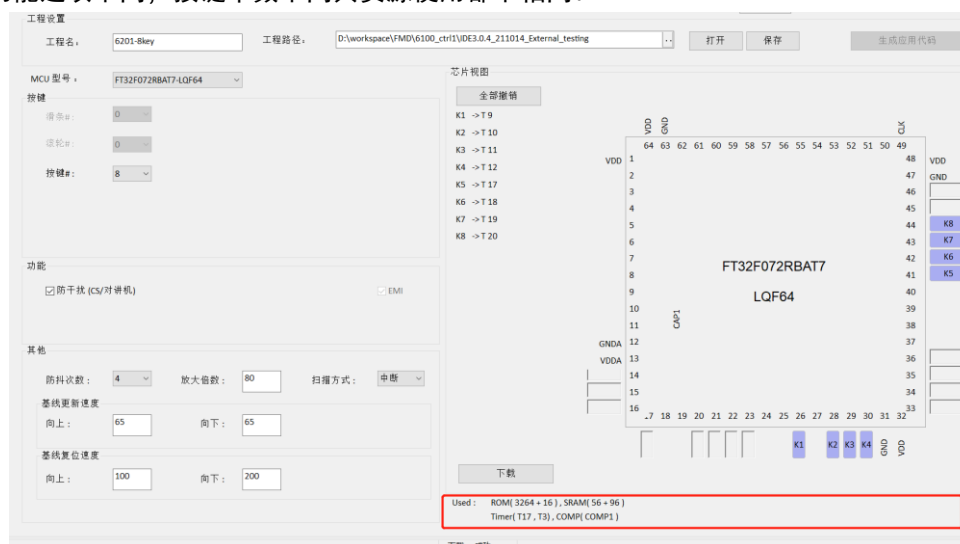
## 6. 芯片引脚配置

根据目标板上的按键定义与芯片触摸引脚对应配置。



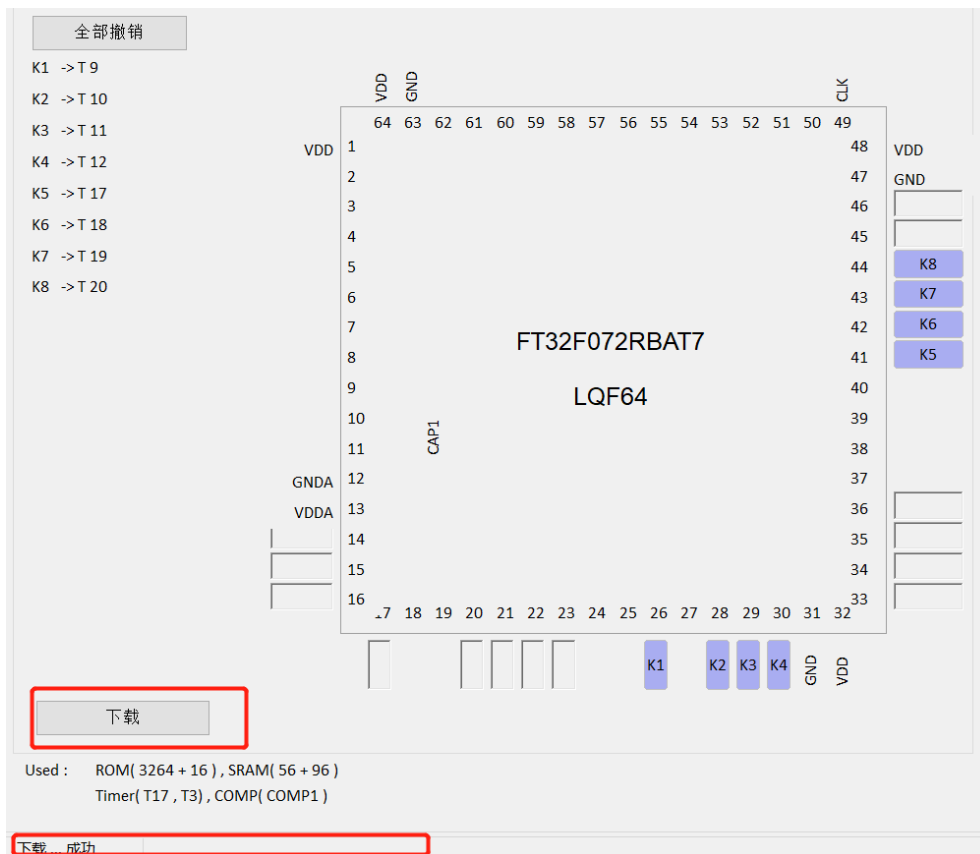
## 7. 资源使用

功能选项不同，按键个数不同其资源使用都不相同。



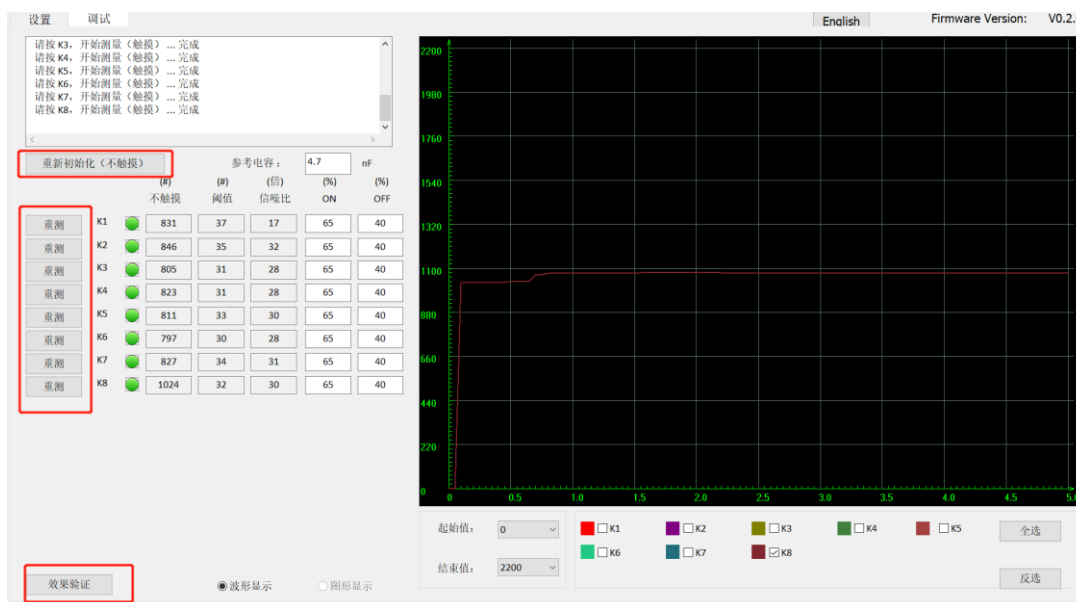
## 8. 程序下载

将工程设置进行保存并下载测试程序到目标板中；如果下载失败，请确认调试工具与目标板的连接是否正确；或者是否为通讯线过长，通讯不稳定造成的。



### 3.3.1.2 测试验证

切换至调试界面，按下“初始化(不触摸)”按钮，并测试按键完成，再进行效果验证，如下图所示：



### 3.3.1.3 生成应用代码

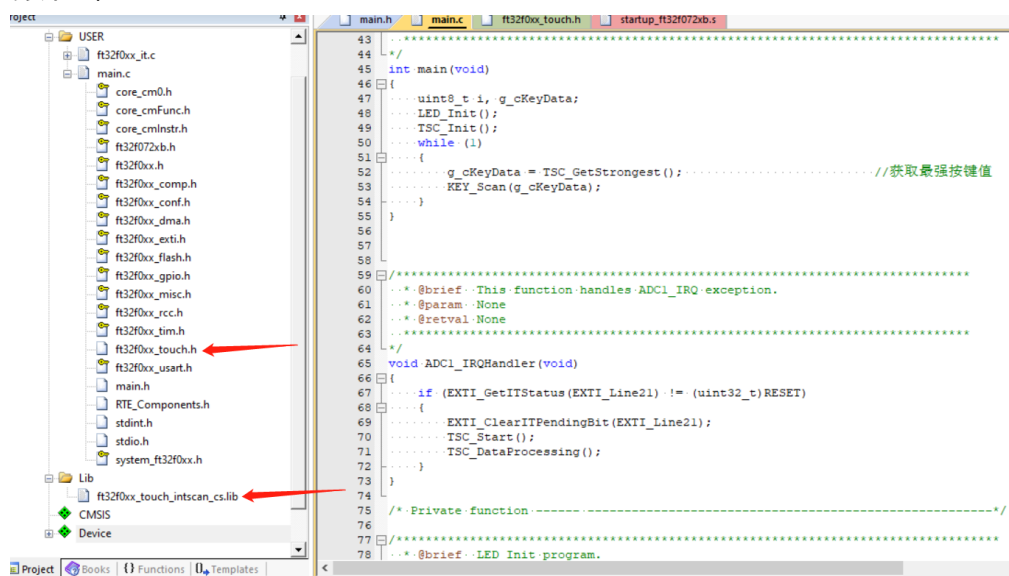
返回设置界面，点击生成应用代码。



1. 在所设置的工程路径下生成一个应用代码文件 touchlib\_xxxx (例:touchlib\_6201);

touchlib_6201	2022/4/14 17:29	文件夹
Update	2022/4/7 11:06	文件夹
6100-8key.ini	2022/3/31 15:20	配置设置
6100-8key.tch	2022/3/23 16:46	TCH 文件
6201-8key.ini	2022/4/14 17:29	配置设置
6201-8key.tch	2022/4/14 16:13	TCH 文件

2. 打开工程



- main.c 为主函数文件。
- ft32f0xx\_touch\_xx\_xx.lib 为触摸库，根据功能选项选择不同的库文件。
- ft32f0xx\_touch.h 为触摸库使用的头文件，其部分参数就是由调试工具产生的。

### 3.3.2 库应用说明

注：详细的库应用请查阅相关的 MCU 的库应用说明书。





**联系信息**

**Fremont Micro Devices Corporation**

#5-8, 10/F, Changhong Building  
Ke-Ji Nan 12 Road, Nanshan District,  
Shenzhen, Guangdong, PRC 518057  
Tel: (+86 755) 8611 7811  
Fax: (+86 755) 8611 7810

**Fremont Micro Devices (HK) Limited**

#16, 16/F, Block B, Veristrong Industrial Centre,  
34-36 Au Pui Wan Street, Fotan, Shatin, Hong Kong SAR  
Tel: (+852) 2781 1186  
Fax: (+852) 2781 1144

<http://www.fremontmicro.com>

\*Information furnished is believed to be accurate and reliable. However, Fremont Micro Devices Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents of other rights of third parties, which may result from its use. No license is granted by implication or otherwise under any patent rights of Fremont Micro Devices Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Fremont Micro Devices Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of Fremont Micro Devices Corporation. The FMD logo is a registered trademark of Fremont Micro Devices Corporation. All other names are the property of their respective owners.