P-Touch

User Manual

Version 0.01

January 16, 2023

6F-6, No.1, Sec. 3, Gongdao 5th Rd., Hsinchu City 30069, Taiwan, R.O.C.

TEL: 886-3-572-8688 www.padauk.com.tw

IMPORTANT NOTICE

PADAUK Technology reserves the right to make changes to its products or to terminate production of its products at any time without notice. Customers are strongly recommended to contact PADAUK Technology for the latest information and verify whether the information is correct and complete before placing orders.

PADAUK Technology products are not warranted to be suitable for use in life-support applications or other critical applications. PADAUK Technology assumes no liability for such applications. Critical applications include, but are not limited to, those that may involve potential risks of death, personal injury, fire or severe property damage.

PADAUK Technology assumes no responsibility for any issue caused by a customer's product design. Customers should design and verify their products within the ranges guaranteed by PADAUK Technology. In order to minimize the risks in customers' products, customers should design a product with adequate operating safeguards.

Table of Contents

Revision History

Revision	Data	Description
0.00	2022/05/05	Preliminary version
0.01	2023/01/16	Other known details bug correct

1. Introduction

P-Touch is a software of program framework generator for PADAUK touch series MCU, which is used for the early development and simulation test of the user's touch scheme. The Touch debugging tool T-Watch is included in the software, which can display the intensity and change of Touch in real time.

Compared with the previous version, P-Touch _ V1.8 has a slightly larger change. The function modules of slider slip and ring are added. The main interface classification, function window and module distribution are optimized and adjusted. The User Manual will describe and introduce P-Touch _ V1.8 in detail.

Fig. 1-1: P - Touch V1.8 icon

P-Touch _ V1.8 must be used with IDE 0.93 and above.

Users can download P-Touch _ V1.8 and IDE from the official website of PADAUK.

Welcome to scan the QR code through QQ and join the PADAUK. Touch MCU Technology Discussion Group.

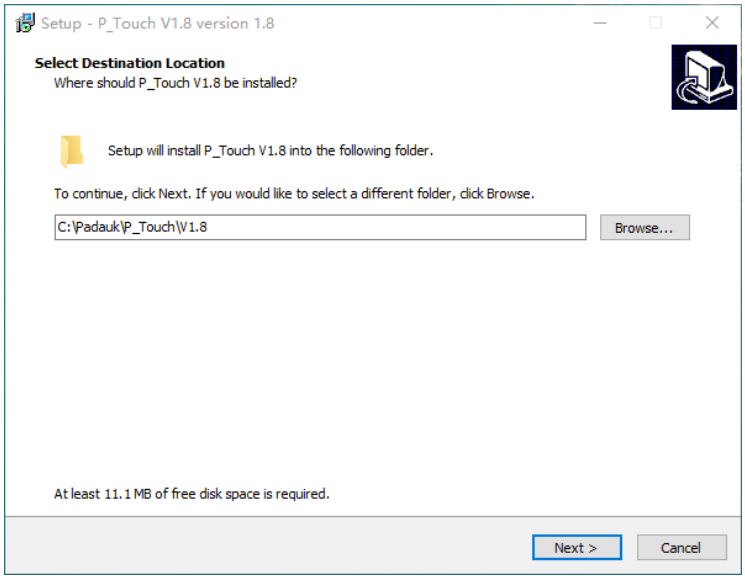
QQ group number: 710107052

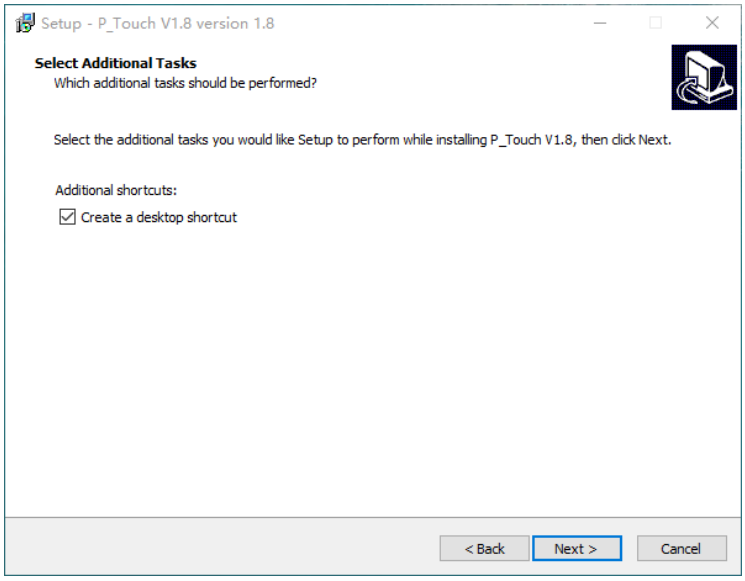
2. Setup P-Touch_V1.8

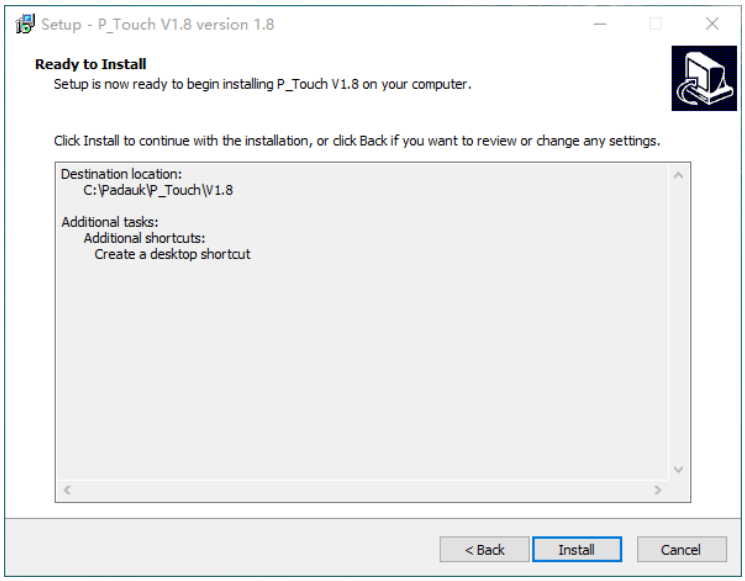
Attention: During the download or installation process, some anti-virus software may misdiagnose. If this occurs, add trust or disable the anti-virus software.

Step1：Download the software installation package
Please download the latest version from the official website of PADAUK.：
http://www.padauk.com.tw/en/technical/index.aspx

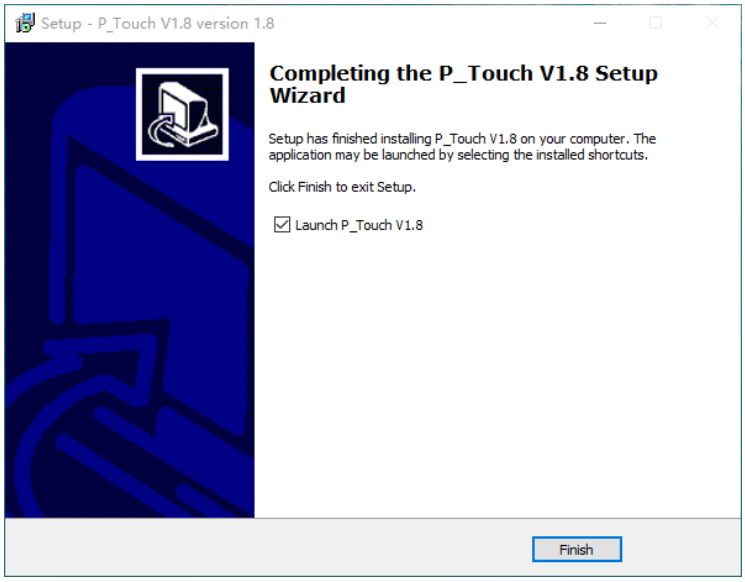
Step2：Double-click the setup software and follow the step prompts step by step;


Setup - P_Touch V1.8 version 1.8
Select Destination Location
Where should P_Touch V1.8 be installed?
Setup will install P_Touch V1.8 into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\Padauk\P_Touch\V1.8
Browse...
At least 11.1 MB of free disk space is required.
Next >
Cancel


Setup - P_Touch V1.8 version 1.8
Select Additional Tasks
Which additional tasks should be performed?
Select the additional tasks you would like Setup to perform while installing P_Touch V1.8, then click Next.
Additional shortcuts:
Create a desktop shortcut
< Back
Next >
Cancel


Setup - P_Touch V1.8 version 1.8
Ready to Install
Setup is now ready to begin installing P_Touch V1.8 on your computer.
Click Install to continue with the installation, or click Back if you want to review or change any settings.
Destination location:
C:\Padauk\P_Touch\V1.8
Additional tasks:
Additional shortcuts:
Create a desktop shortcut
< Back
Install
Cancel

Step3： Setup finished.



3. Introduction to Main Interface of P-Touch _ V1.8

Run P-Touch _ V1.8, as shown below. The software is divided into three functional modules: Touch Project, Slide Project, and Standard List, which will be introduced one by one below.

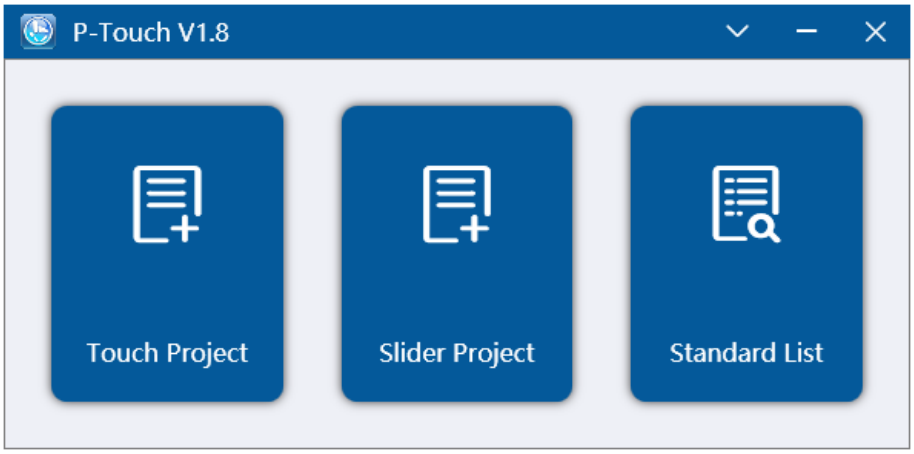


Fig. 3-1: Main Interface of P-Touch _ V1.8

Click the down arrow on the upper right to expand the home page menu to switch the display of software in Chinese and English and the historical record information about software version update.

Fig. 3-2: Home menu

4. Introduction of Touch Project

This module is used to generate touch program framework, please configure according to the arrow index [Summary], [Select Keys], [Parameters] and so on at the top of the software, and then generate programs, users can complete their own functions in the corresponding module in the framework.

4.1. Solution Configuration Description

4.1.1. Summary

Basic project information needs to be set first, including project name and project path; select chip name, package information, application type, whether to enable slider, whether to enable T-Watch and running IDE version; The button on the right side of the package information can be used to view the schematic diagram of the currently selected package. When opening T-Watch, it is necessary to configure the wake-up module and communication port. The introduction of T-Watch will be discussed later. On the right is a brief introduction of the current available touch MCU. For details on MCU, please refer to the specification.

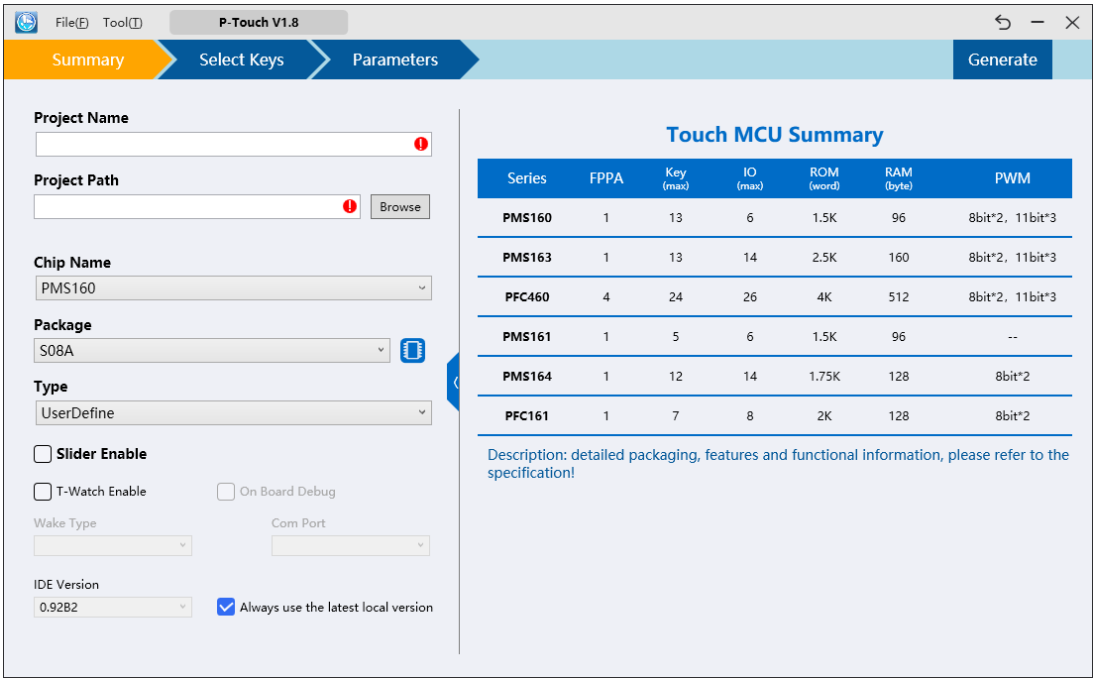


Fig. 4-1: Summary

4.1.2. Select Keys

Working Mode： Including General mode and Power Saving mode:

(1) General mode：Touch sensitive, no sleep, Generally used in applications that do not need to calculate power consumption, such as AC applications;

(2) Power Saving mode：With sleep and wake up function, low standby power consumption, generally used for battery applications;

CS Pin： If the selected chip has multiple external capacitor pins that can be selected, appropriate external capacitor pins can be selected according to requirements;

Select Touch Pins：First click to select the desired touch channel (Step1), and then click the right-moving arrow on its right (Step2). Then all channels selected and used and their initial parameters configuration will be displayed in the list of channels in the middle (Step3)；If Power Saving mode is selected, IO wake up setting can be configured (Step4) ;

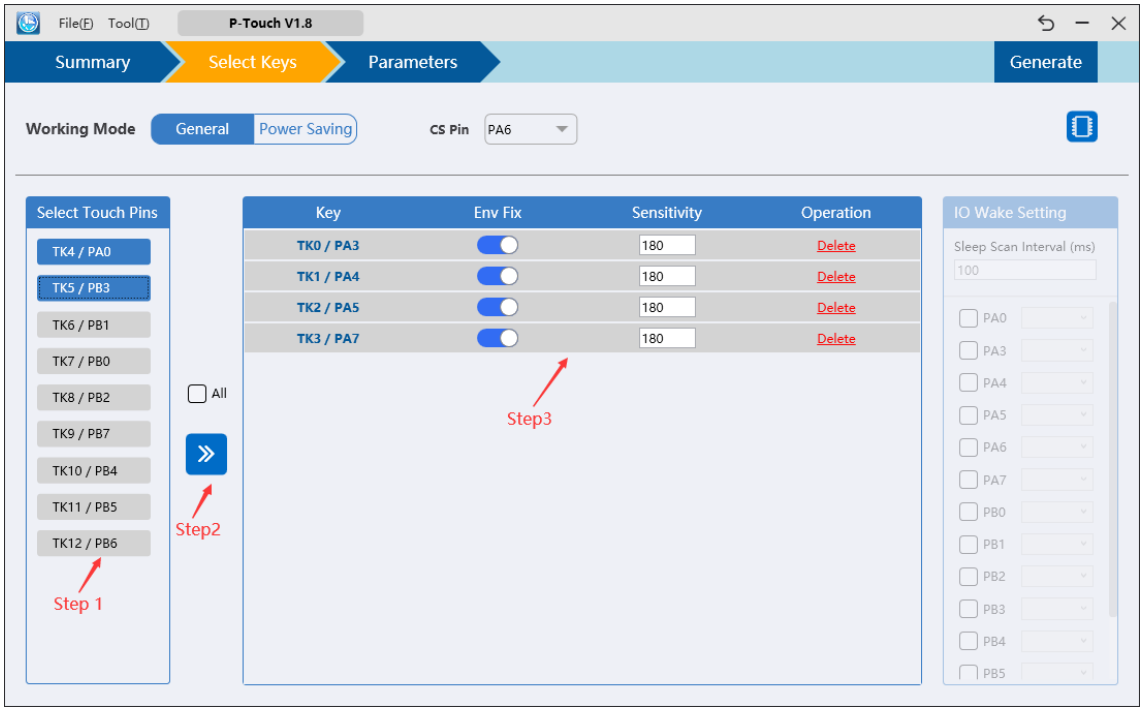


Fig. 4-2: General working mode selection

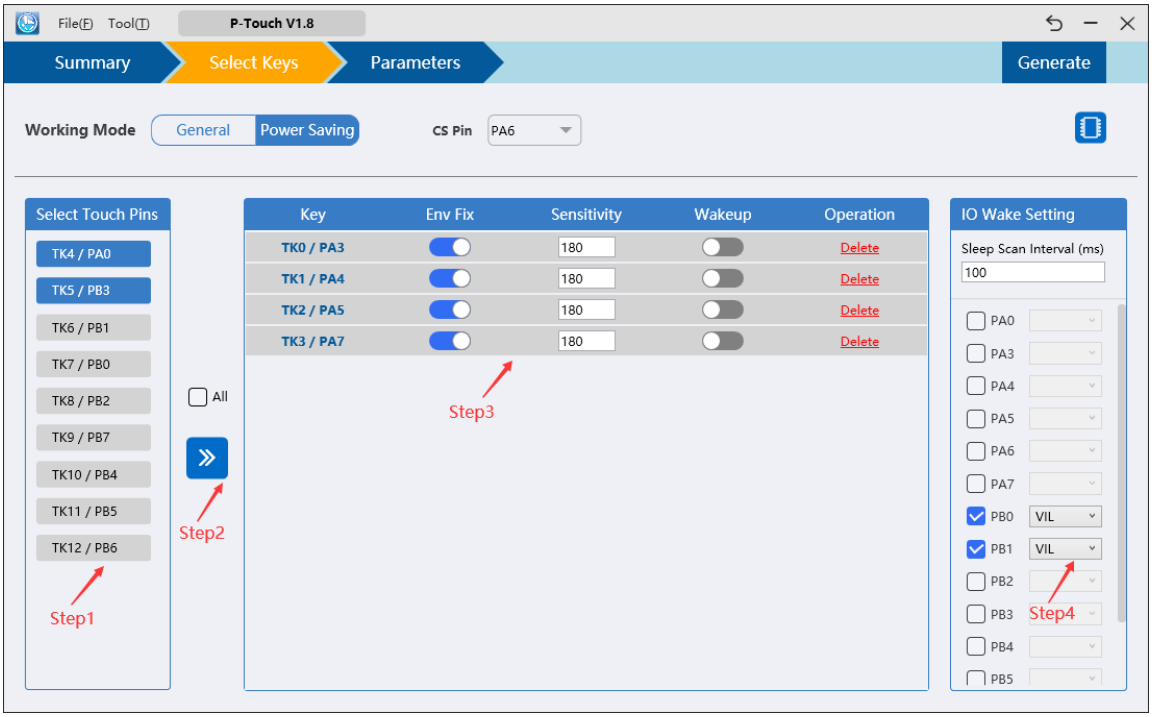


Fig. 4-3: Power Saving working mode selection

Parameters introduction：

Env Fix: Indicates whether the touch environment value will still be repaired in real time when the key is triggered;

Sensitivity: The adjustable range is from 0 to 210. The general default value is 180, but the default value of PMS161 is 150. The higher the value, the higher the sensitivity;

Wakeup： Option in Power Saving mode. If selected, this channel can be used to wake up sleep;

Sleep Scan Interval： In Power Saving mode, the parameters indicates the interval for scanning keys during

sleep. The adjustable range is 50-500ms. The default value is 100ms; If the wake-up speed needs to be improved, the scanning interval can be reduced by a slight adjustment. If the standby current needs to be reduced, the scanning interval can be increased by a slight adjustment.

IO Wake Setting： In Power Saving mode, in addition to waking up the system with touch, you can also set up to wake up the system with external IO. Therefore, the port and potential of IO awakening can be selected here. In general, it is low level awakening. If high level is required to wake up, connect the IO with a pull-down resistor to keep the IO normal low level.

4.1.3. Slider

At present, the program can provide the slider function of 6 keys and 11 levels, among which PMS164 can support up to 12 keys and 23 levels. The relationship between the number of keys (N) and the order of the slider (M) is: M= 2N-1. A single key cannot achieve slider effect. Take two keys and three steps as an example, the state is as follows: press key 1 alone, press key 1 and key 2 at the same time, press key 2 alone; In addition, it should be noted that the slider mode channels, touch channels and the reference channels in the CS coping strategy should not be reused. If you want to use the slider, select Slider Enable in the project's Summary of information, and this column will be added.

Slider type： The slider module is divided into general slider and direction slider. The direction slider is compatible with the general state slider and contains the sliding direction of the slider. The direction slider is only effective when sliding in positive and negative directions.

Sensitivity： The slider sensitivity range and default value are the same as the touch keys, the higher the value is, the higher the sensitivity is.

Wake： If Power Saving module is selected in the key selection bar, the wake-up function can be set under each key of the slider;

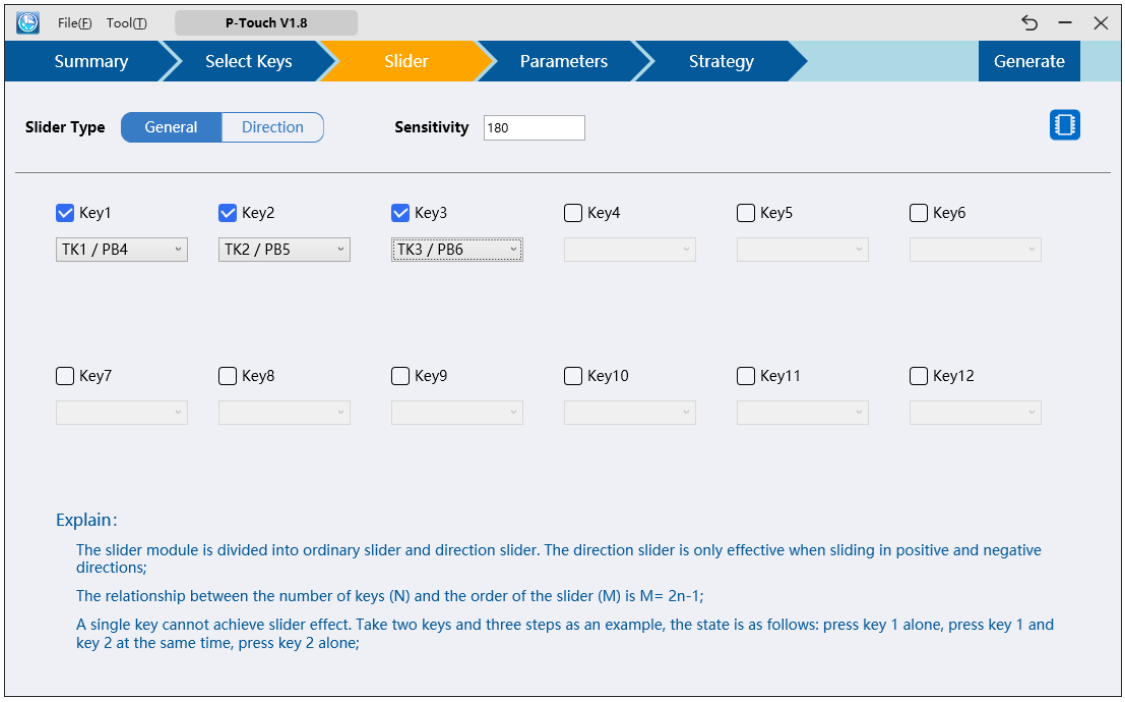


Fig. 4-4: General module of slider type

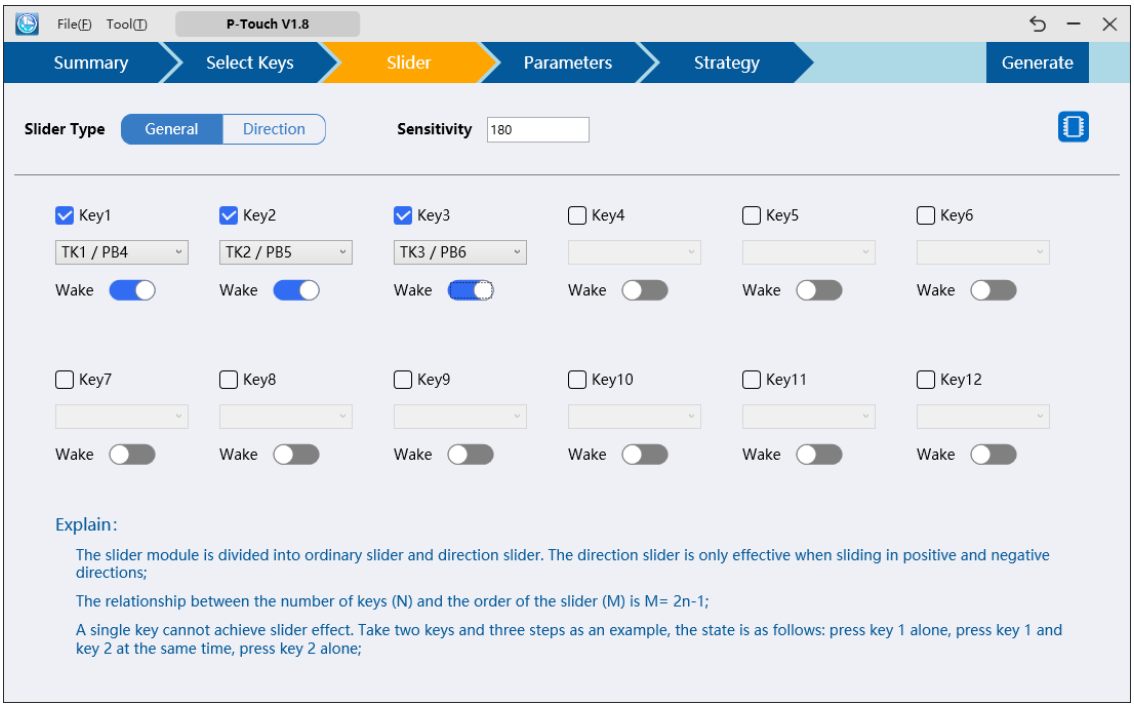


Fig. 4-5: Power Saving module of slider type

4.1.4. Parameters Setting

Due to different chip models, the parameters may be different.

- parameters settings of PMS163, PFC460, PMS164, PMS161 and PFC161:

Touch Register Setting

(1) **TK Scan Clock Source**：That is, the clock that the touch function scans and counts, where the higher the clock frequency, the greater the actual value of the touch. Note that this is not the system clock;

(2) **TK Reference Voltage**：Also known as TK reference voltage, it has an impact on CS capacitor size and touch sensitivity, which can be increased to reduce the external reference capacitance;

(3) **CS Discharge Time**：CS discharge time before touch, generally, the larger the CS capacitor is, the longer the discharge time is needed;

(4) **CS Forced Discharge Time**：CS discharge time extension selection before touch. When the CS value is too large and 128CLK discharge still cannot be fully discharged/charged, you can turn on this option for CS forced discharge/charging, whose efficiency is higher than the CS discharge time value of the previous parameter, the default value is 50us;

Basic Anti-Noise Setting

(1) **TK Filtering Rank**：In the program, the filtering method of sampling and taking the average value is adopted, and the level is from 0 to 6, corresponding to the sampling times of 1, 3, 6, 10, 18, 34 and 66 respectively. The higher the level here, the smoother the filtering and the better the noise resistance, but at the same time the longer the sampling takes, the touch will be slightly slower. The higher the level is, the smoother the filtering is and the stronger the anti-noise capability is. However, the longer the sampling time is, the slower the touch will be. Therefore, the user should select the filtering level according to the actual number of touches and the interference condition, and the default level is 3;

(2) **TK Release Margin**：When a touch is triggered, the touch value is adjusted back to a value below the trigger threshold (TK release margin) before it is considered a touch release. If the value is too large, the touch value

may be triggered and cannot be released. The value range is 0 ~ 20, and the default value is 10;

(3) **Environmental Value Fix**：It is used to touch the environment value to adjust in real time to cope with environmental changes, and it is generally kept on;

(4) **Quick Recovery**：Touch the key quick recovery switch to quickly recover the environment value after the key is released;

(5) **Environmental Value Downward Retarget**：The environment value is difficult to adjust downward. The larger the value is, the slower the adjustment is. A value of 100 means that the data with smaller values are sampled at least 100 times before it is adjusted downward by 1. The value range is 1 ~ 10000, and the default value is 10;

(6) **High Sensitivity**：For the application in the touch environment of the ultra-thick clapboard (the clapboard is larger than 8mm), when the general sensitivity can't meet the requirements, it is necessary to turn on the high sensitivity enable;

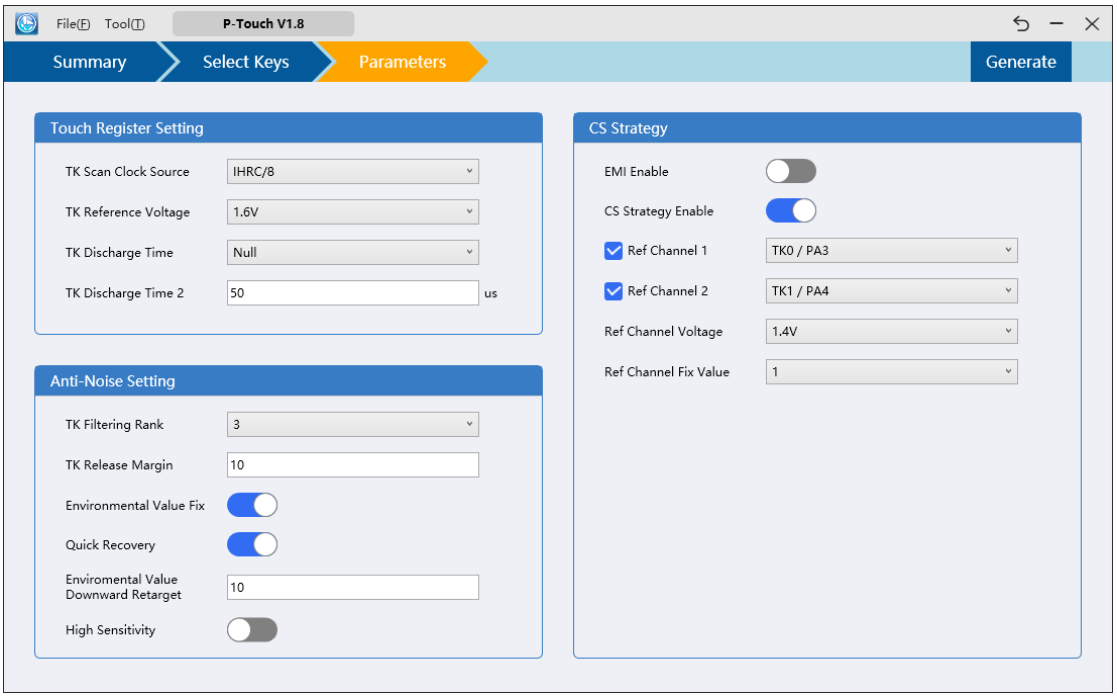


Fig. 4-6: Parameters Setting

CS Strategy Setting

(1) **EMI Enable**：When it is turned on, the IHRC will jump from side to side, so that the measured harmonic power db value will be reduced during the conduction and radiation tests;

(2) **CS Strategy Enable**：This option can be turned on when the product needs to pass the CS test;

(3) **Ref Channel 1/2**：When the interference is large, please open at least one reference channel in sequence; You can select the unused channel or the touch channel hidden in IC (the touch channel hidden in IC is preferred), this channel does not generate touch signal, do not mix with the touch channel in use;

(4) **Ref Channel Voltage**：The selection of touch reference voltage has an impact on the performance of touch sensitivity and the selection of CS capacitance. In theory, the larger the reference voltage is, the higher the sensitivity is. Because the reference key is usually hidden in the IC interior or unused TK foot, so the value is relatively high, in order to prevent overflow, so it is necessary to reduce the level to prevent overflow;

(5) **Ref Channel Fix Value**：Modify the base number of the environment value through the reference channel. The value range is 1 ~ 6, and the default value is 1;

- PMS160 parameters setting:

Touch Register Setting

(1) **TK Clock Setting**：IFC touch clock selection, value selection: 0: 32MHz, 1:16 MHz, default selection: 16 MHz;

(2) **LDO Voltage**：IFC touch LDO voltage selection, value options: 0: 1.8V, 1: reserved, 2: 1.7 V, 3: 1.6V, default option: 1.8 V;

(3) **TK Ref Cap Coefficient**：IFC Touch Reference Capacitance Factor Selection, where selection affects IFC readings, value options: 0: *1, 1: *2, 2: * 3, 3: * 4,…,N: * (N + 1)　N < = 15, default option is 8;

(4) **IFC Model**：IFC mode selection, value range: 0: Module0, 1: Module1, default option: Module 0;

(5) **IFC Pulse Count Frequency Division**：Module0 parameter, IFC pulse count frequency division, value range: 0:/1, 1:/2, 2:/4, 3:/8, 4:/16, 5:/32, 6:/64, 7:/128, 8:/256, default option: 0;

(6) **IFC Pulse Count Upper Division**：Module0 parameter, IFC pulse count upper limit selection, selection here affects IFC reading, value range: 1 ~ 255, default value: 50;

(7) **IFC Reading Time**：Module 1 parameter, IFC reading time selection, unit us, accuracy 10us (how long to start IFC to read the IFC value, the longer the waiting time, the larger the IFC value, affecting the sensitivity), value range: 100 ~ 3000, default value: 1000;

Multi Write Sensitivity Parameters Setting

(1) **Multi Write Paras Enable**：Multi-burn-over parameter enable setting, including burn-over parameters Const _ SEN _ T _ KeyX (sensitivity), Const _ IFC _ Count (Module0 IFC count upper limit), Const _ IFC _ Timing (Module1 IFC reading time);

(2) **Multi Write Paras Max Cnt**：Setting of multiple write parameters maximum count, value range: 1 ~ 8, default value: 3;

Basic Anti-Noise Setting

(1) **TK Filtering Rank**：TK sampling movement filtering level: the larger the value is, the smoother the filtering will be, the longer the sampling time will be, and the slower the button trigger will be. The value options are: 1:1/2, 2:1/4, 3:1/8, and the default value is 2.

(2) **TK Release Margin**：When TK is triggered, the touch value is adjusted back to a value lower than the trigger threshold (TK release margin), which is considered to be touch release. Professional parameters should be adjusted carefully. If the margin is set too large, it may cause that TK can't be released after it is triggered. Value range: 0-20, default value: 10;

(3) **Environmental Value Fix**：Environment value fix main switch, default on;

(4) **Env Fix Count Free**：When free, the environment value is fixed and the count is delayed, that is, the environment value is repaired once relative to the number of times of sampling, the value range is 1 ~ 100, and the default value is 10;

(5) **Env Fix Count Busy**：When busy(pressed), the environment value is fixed and the count is delayed, that is, the environment value is repaired once relative to the number of times of sampling, the value range is 1 ~ 100, and the default value is 10;

(6) **Env Fix Speed Free**：When free, the speed of environment value. The smaller the value is, the faster the repair speed is. Value option: 0: off 1:1/2 2:1/4 3:1/8 4:1/16 5:2/32 6:6/64. Default option: 5;

(7) **Env Fix Count Busy**：When busy (pressed), the speed of environment value. The smaller the value is, the faster the repair speed is. Value option: 0: off 1:1/2 2:1/4 3:1/8 4:1/16 5:2/32 6:6/64.Default option: 5;

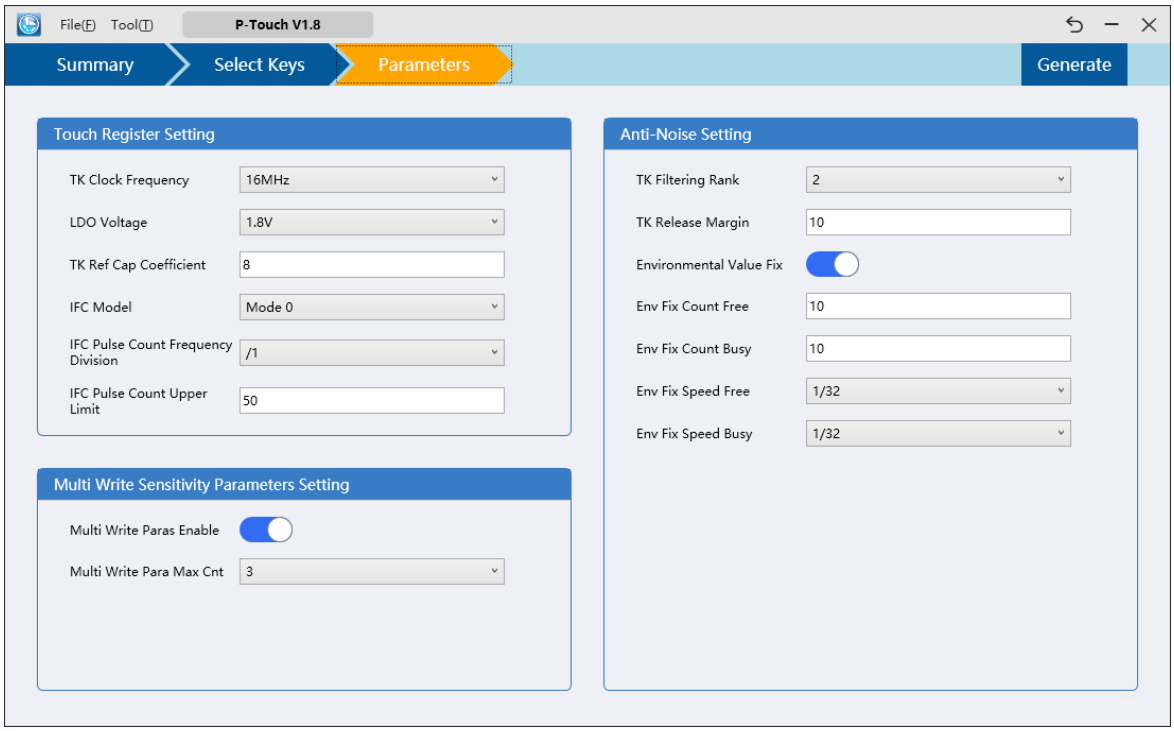


Fig. 4-7: PMS160 Parameters Setting:

4.1.5. Strategy

This strategy only supports PMS164 and PMS161, mainly anti-jamming and extra-long touch applications (such as anti-intercom interference and fish-drift applications).

Anti-jamming Strategy

(1) **Anti-jamming Enable**：Turn on to enable this anti-jamming strategy;

(2) **Ref Channels 1 and 2**：Touch channel environment value repair speed for detecting strong jamming;

(3) **Environment Value Repair Speed**：When the number of times of no interference is greater than the value, the environment value is fixed;

(4) **Environment Value Repair Ratio**：The environment value approaches the actual value at a certain speed. The larger the value is, the slower the repair speed is. Specifically, the difference between the previous environment value and the actual value is divided by 2^n.

(5) **Ref channel jitter threshold**：If the jitter of the actual value of the reference channel exceeds this value, it is judged abnormal.

(6) **Touch depth threshold setting**：Indicates the false trigger threshold when the touch channel is strongly interfered, specifically, it cannot exceed (220-sensitivity of the reference channel) × the value;

Extra Long Touch Strategy

(1) Extra Long Touch Enable：Turn on to enable this extra-long touch strategy;

(2) Abnormal key release detection rate：When the number of times reaching the release point is greater than this value, the environment value is fixed;

(3) Abnormal key release detection sensitivity：A release point is set, and that release position is the difference between the environment value move down from the original release position and the actual value divided by 2 ^ n;

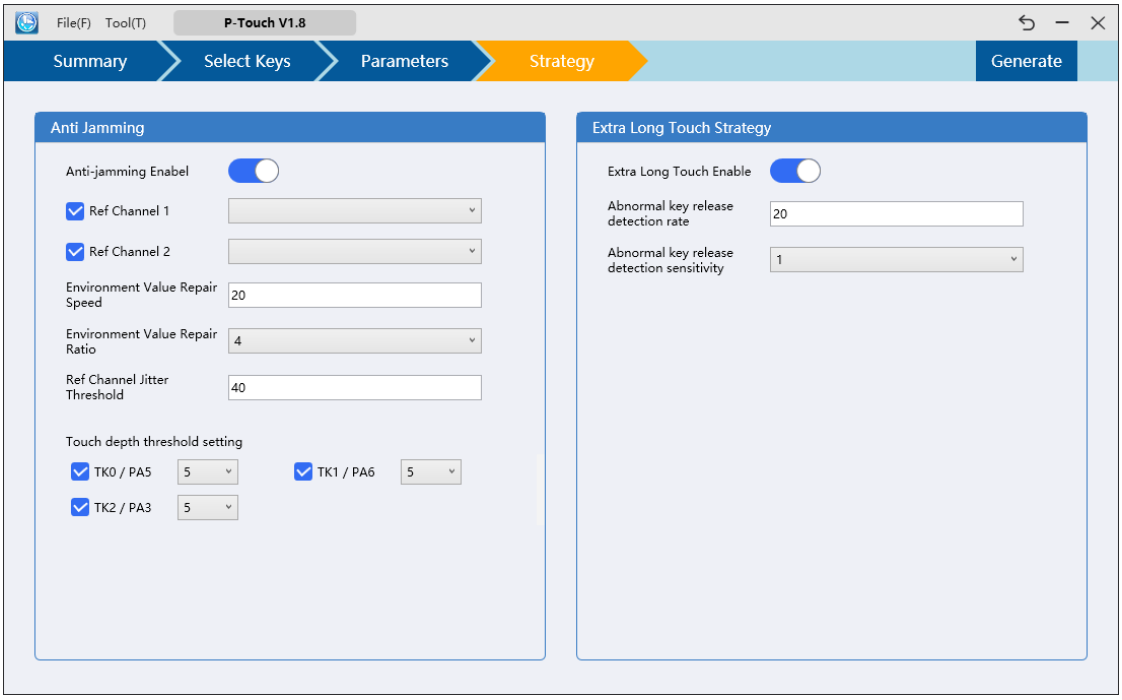


Fig. 4-8: Strategy

4.1.6. Menu Bar Introduction

(1) **File**：Including open configuration and save configuration, after the user set the configuration and parameter generation program framework, an INI format configuration file with the same name as the project will be generated in the program package, which is used to record the configuration and parameters set by the user. Of course, users can also manually click save configuration file to save the current configuration and parameters. Click open configuration to open the Settings in the configuration file with this software;

(2) **Tool**：Click to open the T-Watch touch debugging tool, which will be described later;

4.2. Generate framework of project

As described above, after configuration as required, click the Generate program button in the upper right corner of the software to generate the program framework, and then the following window will pop up to prompt whether to run the program directly with IDE.

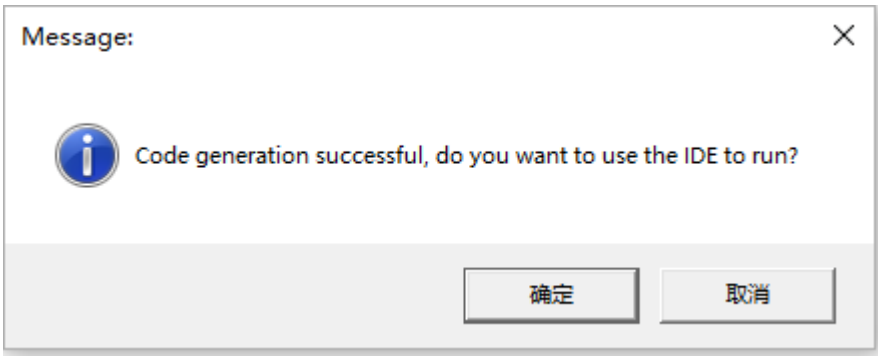


Fig. 4-9: Prompt to generate the code framework

After running the generated project with the IDE, you can write actual engineering programs based on this framework. Each project file is introduced in detail below (for example, we generate a Touch_Demo framework, and the file architecture diagram in the program is as follows):

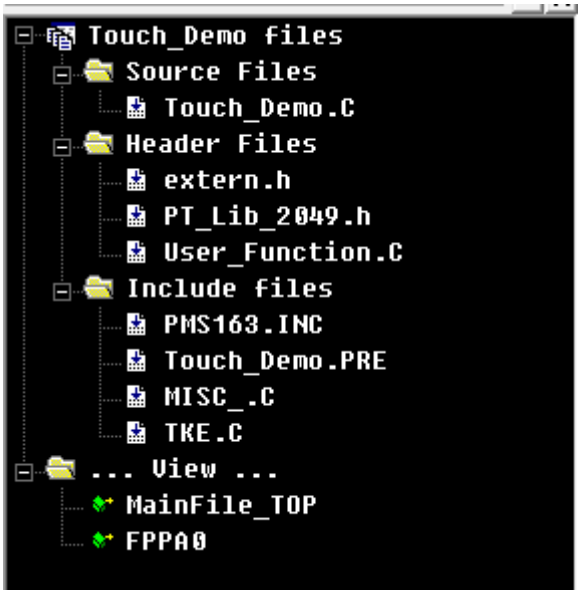


Fig. 4-10：Program file architecture

- The Touch _ Demo. C is a main program file, which can perform relevant operations such as input and output and digital enabling on the used IO pin and select the working mode of the main program;
- Extern.h is an extended program header file that contains chip name, CS pins, system clock, and UART clock information;
- PT_Lib_xxxx.h is the touch library configuration file, which records the used touch channel information, parameter configuration, UART Settings and other information;
- User_function.C is the function compilation file of the user, in which the user can write the required functions according to the corresponding module's own requirements;

Several of the main program files are described in detail below.

4.2.1. Configuration file of touch library

The touch library configuration file PT_Lib_xxxx.h is used to configure touch related settings, including touch pin selection, sensitivity configuration, wake up pin configuration, environment fix parameters and Uart enabling, etc.

Attention：PT_Lib_xxxx.h file description is intended to help users understand the program. The parameter Settings in the file have been automatically configured during P-Touch generates the program. Do not modify the configuration information in this file unless necessary.

1. Touch channel enabling Settings: (Const_EN_CH_T_Key)

Channel enable set to 1: Channel on; Channel enable set to 0 or commented out: Channel off

```
    //T_Key Channel enablement setting
Touch_Channel_Selection:
    Const_EN_CH_T_Key0  =>  1
    Const_EN_CH_T_Key1  =>  1
    Const_EN_CH_T_Key2  =>  1
    Const_EN_CH_T_Key3  =>  1
    Const_EN_CH_T_Key4  =>  1
    Const_EN_CH_T_Key5  =>  1
```

Fig. 4-11: TK channel enablement setting

Attention：

(1) Keep analog input for the I/O corresponding to the enabled T_Key channel and turn off the pull-high resistance. Do not switch to output IO at will.

(2) If you are uncertain about the number of touch channels before running the program, or if you need to close or open the touch channels in the program, you can open all channels that may be used first.

2. Sleep timing scan and wake up Settings: (Const_Wakeup_CH_T_Key)

Touch channel wake up set to 1: enable channel wake up; Touch channel wake up set to 0 or comment out: Turn off channel wake up;

```
\******When the main program is in power-saving mode,the program will sleep.The following wake channels need to be set******/
    //T_KEY wakes up the channel enable setting
Touch_Wakeup_Channel_Selection:
    Const_Wakeup_CH_T_Key0  =>  1
    Const_Wakeup_CH_T_Key1  =>  1
    Const_Wakeup_CH_T_Key2  =>  1
    Const_Wakeup_CH_T_Key3  =>  1
```

Fig.4-12: TK Wakes up the channel enable setting

3. Touch sensitivity: (Const_SEN_T_Key)

Touch channel sensitivity setting, the range is 0~210; The default value is 180, but the default value of PMS161 is 150. The higher the value is, the higher the sensitivity is.

```
    //T_KEY Sensitivity setting
Touch_Sensitivity_Set:
    Const_SEN_T_Key0    =>  180 //Touch sensitivity setting, the greater the value, the higher the sensitivity
    Const_SEN_T_Key1    =>  180 //Option value: 0-210 default: 180
    Const_SEN_T_Key2    =>  180
    Const_SEN_T_Key3    =>  180
    Const_SEN_T_Key4    =>  180
    Const_SEN_T_Key5    =>  180
```

Fig.4-13: TK Sensitivity setting

4. **Environment value fixed enable Settings：(Const_Press_Fix_T_Key)**

   Whether to continue to modify the environment value after touching the button. Set it to 0: continue to fix; set it to 1 or comment it out: do not fix;

```
//Whether to continue to modify the environment value after touching the button//0: continue;   1: close
Const_Press_Fix_T_Key0  =>  0
Const_Press_Fix_T_Key1  =>  0
Const_Press_Fix_T_Key2  =>  0
Const_Press_Fix_T_Key3  =>  0
```

Fig.4-14: Environment value fixed enable Settings

5. **Slider Settings：(Const_En_Slider)**

   Const_En_Slider_A Set to 1: enable the slider. Set to 0: disable the slider

   Slider_T_Key Set the channel and sequence of the slider. Currently, it supports 6 channels at most, and the sensitivity range and default value are the same as those of the touch channel;

```
Touch_Slider_Set:
    Const_En_Slider_A    =>  1   //6Key slider enable
    Slider_T_Key1    =>  'TK0';
    Slider_T_Key2    =>  'TK1';
    Slider_T_Key1    =>  'TK2';
    Slider_T_Key2    =>  'TK3';
    Const_SEN_Slider_A=>    180  //Set the slider sensitivity, Max=210,Min=0,
                                 // the sensitivity should be set,the greater the value, the higher the sensitivity
                                ////0: least sensitive 210: most sensitive default:180
```

Fig.4-15: Slider Settings

6. **Touch register Settings：**

   Due to different MCU models, the optional range of parameters is also different. PMS164 and PMS160 are taken as examples below:

- **PMS164 reference is as follows:**

  (1) Touch clock source settings：(**Const_Touch_Source_CLK**)

  0：reserved；1：reserved；2：IHRC/4；3：IHRC/8；4：IHRC/16；5：IHRC/32；6：IHRC/64；7：IHRC/128；8：ILRC。default:3：IHRC/8；

  (2) CS capacitance reference voltage setting：(**Const_Touch_VRef**)

  0：0.5*VCC；1：0.4*VCC；2：0.3*VCC；3：0.2*VCC

  It has an effect on CS capacitance and touch sensitivity, The highest sensitivity is set to 0, and the lowest sensitivity is set to 3;

  (3) Touch before CS discharge time selection：(**Const_Touch_Discharge**)

  0：reserved；1：CLK_32；2：CLK_64；3：CLK_128；

  (4) Before touching the CS discharge time expansion option：(**Const_Touch_Discharge2**)

  This is a newly added parameter，when CS is too large, the discharge CLK_128 still cannot fully discharge/charge, can be turned on this option, CS forced discharge/charge. Higher discharge/charging efficiency than a Const_Touch_Discharge (US), ranging from 0 to 1000 us.

```
//Actual touch key Settings
Const_Touch_Source_CLK  =>  3;  //Touch clock selection
                                //0:reserved, 1:reserved, 2:IHRC/4, 3:IHRC/8,
                                //4:IHRC/16, 5:IHRC/32, 6:IHRC/64, 7:IHRC/128, 8:ILRC
                                //default:3
Const_Touch_VRef        =>  0;  //0:0.5*VCC, 1:0.4*VCC, 2:0.3*VCC, 3:0.2*VCC
                                //CS capacitance reference voltage setting 0.2VCC-0.5VCC
                                //It has an effect on CS capacitance and touch sensitivity
                                //Raise here to reduce the external reference capacitance
                                //default:0
Const_Touch_Discharge   =>  0;  //Touch before CS discharge time selection, the longer the time, the cleaner the discharge
                                //0:reserved, 1:CLK_32, 2:CLK_64, 3:CLK_128
                                //default: 3
Const_Touch_Discharge2  =>  50; //Before touching the CS discharge time expansion option, when CS is too large,
                                //the discharge CLK_128 still cannot fully discharge/charge, can be turned on this option, CS forced discharge/charge
                                //Higher discharge/charging efficiency than a Const_Touch_Discharge (US), ranging from 0 to 1000 us
                                //default: 50
```

Fig.4-16: PMS164 Touch register Settings

- **PMS160 reference is as follows:**

(1) IFC touch clock settings：(**Const_IFC_CLK**)
Option value: 0:32MHz, 1:16MHz
Default: 1

(2) IFC touch LDO voltage settings：(**Const_IFC_LDO**)
Option value: 0:1.8V, 1: reserved, 2: 1.7V, 3: 1.6V
Default: 0

(3) IFC touch reference capacitance coefficient：(**Const_IFC_CAP**)
Option value: 0:*1, 1:*2, 2:*3, 3:*4, …… ,N:*(N+1) N<=15
Default: 8

(4) IFC module selection：(**Const_IFC_Module**)
Option value: 0: Module0, 1: Module1
Default: 0

(5) IFC pulse count frequency division：(**Const_IFC_Scalar**)
Module0 parameter，Value range: 0:/1, 1:/2, 2:/4, 3:/8, 4:/16, 5:/32, 6:/64, 7:/128, 8:/256
Default: 0

(6) IFC pulse count upper limit: (**Const_IFC_Count**)
Module0 parameter，Value range: 1~255
Default: 50

(7) IFC reading timing：(**Const_IFC_Timing**)
Module1 parameter，Value range: 100 ~ 3000
Default: 1000

```
    //Actual touch key Settings
    Const_IFC_CLK   =>  1;  //IFC Touch clock selection
                            //Option value: 0:32MHz, 1:16MHz
                            //default: 1
    Const_IFC_LDO   =>  0;  //IFC touch LDO voltage selection
                            //0:1.8V, 1:reserved, 2:1.7V, 3:1.6V
                            //default: 0
    Const_IFC_CAP   =>  8;  //IFC touch reference capacitance coefficient selection, this selection affects IFC reading
                            //Option value: 0:*1, 1:*2, 2:*3, 3:*4, …… ,N:*(N+1) N<=15
                            //default: 8
    Const_IFC_Mode  =>  0;  //IFC Module selection
                            //Option value:   0: Module0, 1: Module1
                            //default: 0
C_IFC_Mode0:
    Const_IFC_Scalar=>  0;  //IFC pulse count frequency division
                            //value range:  0:/1, 1:/2, 2:/4, 3:/8, 4:/16, 5:/32, 6:/64, 7:/128, 8:/256
                            //default: 0
    Const_IFC_Count =>  50;//IFC pulse count upper limit selection, where selection affects IFC reading
                            //value range: 1~255
                            //default: 50
```

Fig.4-17: PMS160 Touch register Settings

7. Touch environment correction and anti-interference Settings

(1) Touch sampling once required filter level Settings:（**Const_T_Key_Smooth_Rank**）

The filtering level is 1 ~ 6, corresponding to the sampling times of 1, 3, 6, 10, 18, 34 and 66 respectively. TK sampling filter level, the larger the value, the smoother the filter, and the longer the sampling time, the default value is 3.

(2) Touch release margin settings:（**T_Key_Release_Margin**）

When touch is triggered, it is considered as touch release only when the touch value is lower than the trigger threshold (touch release margin)；This value needs to be adjusted carefully, and cannot be set too large, otherwise, it may not be released after touch triggering; the value range is 0 ~ 20, and the default value is 10;

(3) Touch key environment value fix master switch:（**Const_Env_Fix**）

Set to 1: turn on the environment value correction general enable; set to 0: turn off the environment value correction general enable;

(4) Touch key environment value fast recover switch:（**Const_Fast_Recover**）

Fast recovery of environment value after release of touch key

Set to 1: enable. 0: Disable.

(5) Touch key environment value fix gradient down:（**Const_Env_Dw_Fix_Cnt**）

Option value:10~10000; The smaller the value is, the faster the correction is; the larger the value is, the slower the correction is;

A value of 100 means that at least 100 samples of small data will be sampled before it starts to be reduced by 1;

```
//------Touch environment correction and anti-interference Settings-----
Touch_Ref_and_Noise_Set:
    Const_T_Key_Smooth_Rank =>  3   //Tk sampling filter level, the larger the value, the smoother the filter, and the longer the sampling time
                                    //Option value: 1:4 times, 2:6 times, 3:10 times, 4:18 times, 5:34 times, 6:66 times,
                                    //default: 3
    T_Key_Release_Margin    =>  10  //Tk release margin. When Tk triggers,
                                    //the touch value is considered to be released only when the touch value is recalled to a certain value below the trigger threshold (Tk release margin)
                                    //Professional parameters, carefully adjusted, if the margin is set too large, may lead to TK triggered, unable to release
                                    //Option value:0-20
                                    //default:10
    Const_Env_Fix           =>  1   //Environmental value correction enable
                                    //Option value:0:close, 1:open
                                    //default:1
    Const_Fast_Recover      =>  1   //Touch button quick recovery switch (quick recovery of environment value after key release)
                                    //Option value:1:open 0:close
                                    //default:1
    Const_Env_Dw_Fix_Cnt    =>  10  //The larger the value, the slower the correction.
                                    //A value of 100 means that at least 100 samples of small data will be sampled before it starts to be reduced by 1
                                    //Option value:10 - 10000
                                    //default:10
```

Fig.4-18: Environment value correction and anti-interference Settings

PMS160 Reference Section:

```
Const_Env_Fix_Count_Free=>  10  //When free, the environment value is fixed and the count is delayed,
                                //That is, the environment value is repaired once relative to the number of times of sampling
                                //Value range: 1 ~ 100
                                //Default value: 10
Const_Env_Fix_Count_Busy=>  10  //When busy(pressed), the environment value is fixed and the count is delayed,
                                //That is, the environment value is repaired once relative to the number of times of sampling
                                //Value range: 1 ~ 100
                                //Default value: 10
Const_Env_Fix_Speed_Free=>  5   //When free, the speed of environment value.
                                //The smaller the value is, the faster the repair speed is.
                                //Value option: 0:off 1:1/2 2:1/4 3:1/8 4:1/16 5:1/32 6:1/64    //12disable
                                //Default option: 5
Const_Env_Fix_Speed_Busy=>  5   //When busy (pressed), the speed of environment value.
                                //The smaller the value is, the faster the repair speed is.
                                //Value option: 0:off 1:1/2 2:1/4 3:1/8 4:1/16 5:1/32 6:1/64    //12disable
                                //Default option: 5
```

Fig.4-19: PMS160 Environment fixed some parameters

(6) When free, the environment value is fixed and the count is delayed：（**Const_Env_Fix_Count_Free**）

The environment value is repaired once relative to the number of times of sampling, the value range is 1 ~ 100

Default value: 10

(7) When busy(pressed), the environment value is fixed and the count is delayed: (Const_Env_Fix_Count_Busy)

The environment value is repaired once relative to the number of times of sampling, the value range is 1 ~ 100

Default value: 10

(8) When free, the speed of environment value：（**Const_Env_Fix_Speed_Free**）

The smaller the value is, the faster the repair speed is. Value option: 0: off 1:1/2 2:1/4 3:1/8 4:1/16 5:2/32 6:6/64. //12 disable

Default option: 5;

(9) When busy (pressed), the speed of environment value：（**Const_Env_Fix_Speed_Busy**）

The smaller the value is, the faster the repair speed is. Value option: 0: off 1:1/2 2:1/4 3:1/8 4:1/16 5:2/32 6:6/64. //12 disable

Default option: 5;

8. Anti - noise reference key setting

(1) CS coping strategy setting（**Const_Touch_Noise_Strategy**）

It mainly deals with power interference, Option value：0 - 3, which correspond to off and on anti-noise strategy 1- on anti-noise strategy 3 respectively;

Anti-noise strategy 1 and anti-noise strategy 3 are reserved;

Anti-noise strategy 2 The response strategy for CS testing requires 1 or 2 reference channels. When the interference is strong, open at least one reference channel and open it in sequence.

(2) Reference key level settings：（**Const_Touch_Noise_Vref**）

Because the reference key is usually hidden in the IC interior or unused TK foot, so the value is relatively high, in order to prevent overflow, so it is necessary to reduce the level to prevent overflow.

Option value：0：0.5*VCC；1：0.4*VCC；2：0.3*VCC；3：0.2*VCC, default value: 3

(3) Reference channel modify cardinality：（**Const_T_Key_Noise_Fix_Cn**）

Modify the cardinality of the environment value with the anti-noise button, Option value：1-6, default value:1

```
//Anti - noise reference key setting
Const_Touch_Noise_Strategy  =>  2    //It mainly deals with power interference
                                     //Option value, 0-3 Correspond to off and on anti-noise strategy 1- on anti-noise strategy 3 respectively
                                     //Anti-noise strategy 1 Reserved
                                     //Anti-noise strategy 2 The response strategy for CS testing requires 1 or 2 reference channels
                                     //Anti-noise strategy 3 Reserved
Const_T_Key_Noise_Ref_1 =>  'TK1'
Const_T_Key_Noise_Ref_2 =>  'TK2'
                                     //When the interference is large, please open at least one reference channel,
                                     //and open it in order :Const_T_Key_Noise_Ref_1>Const_T_Key_Noise_Ref_2
                                     //Note: The reference channel set here does not generate touch signal. Do not mix it with the use of the touch channel
Const_Touch_Noise_VRef      =>  3;   //Refer to key level
                                     //Because the reference key is usually hidden in the IC interior or unused TK foot, so the value is relatively high,
                                     //in order to prevent overflow, so it is necessary to reduce the level to prevent overflow
                                     //Generally lower than the voltage set by Const_Touch_VRef
                                     //Option value, 0:0.5*VCC, 1:0.4*VCC, 2:0.3*VCC, 3:0.2*VCC
                                     //default:1
Const_T_Key_Noise_Fix_Cn=>  1        //Modify the cardinality of the environment value with the anti-noise button
                                     //Option value, 1-6
                                     //default:1
```

Fig.4-20：Anti - noise reference key setting

9. Parameter setting of upper computer

(1) Enable the upper computer:（**Const_EN_Uart**）

When the upper computer needs to be started, please enable the Const _ EN _ Uart. If the upper computer is not used, please disable it;

(2) Setting of wake up mode of upper computer:（**Const_Uart_Wakeup_Module**）

0: low voltage wake-up, the communication port needs to be short-circuited to the ground for more than 0.5s;

1: T _ Watch wake-up, click the connection button on the upper computer to send a wake-up signal to the chip for wake-up;

(3) Baud rate of upper computer：（**UART_BaudRate**）

UART baud rate range: 9600 ~ 56000, the default value is 38400. Note that if the baud rate value is changed, the baud rate on T_Watch needs to be changed as well.

(4) Communication port of upper computer：（**Interrupt_Uart**）

Select the UART interrupt and transmission port here, and the optional ports of different chips will be different. PA5 and PB0 interrupts are not supported during simulation, and PA5 is recommended for actual IC burning; in addition, if PMS160 is selected for the chip, the on-board debugging mode can be enabled, which will be introduced in the subsequent use of the upper computer;

```
//-------Parameter setting of upper computer---------------
T_Watch_Set:
    Const_EN_Uart       =>  1       //Uart Enable, 0:Uart Disable   1:Uart Enable
                                    //When you need to connect to the computer, please put Const_EN_Uart on the power,
                                    //please divide the power without the upper computer
                                    //The parameters of the upper computer are invalid except for energy
                                    //default:0
    Const_Uart_Wakeup_Mode=>1       //Upper computer wake-up mode: 0: low level wake-up (> 0.5S) 1: upper computer T-watch wake-up
                                    //default:1
    UART_BaudRate       =>  38400   //UART Baudrate:9600-56000
                                    //default:38400
    Interrupt_Uart      =>  2       //UART interrupt and transmission port selection
                                    //0:PA0  1:PB5
                                    //2:PB0  3:PA4
```

Fig.4-21：Parameter setting of upper computer

Attention：If the software program has chosen to enable the UART communication port and you want to change this setting in your program, in addition to changing Interrupt_Uart, you need to enable input, pull-high, and digital input for the IO in the FPPA_IO_Set function.

10. Touch test switch

```
//--------Touch test switch--------------
    T_Key_Debug         =>  1       //Open to see the open channel touch signal in the Log
    Continuous_Debugging=>  1       //On continuous output measurement signal,
                                    //when off only signal jump will output measurement signal to the LOG window (such as touch)
    Disable_Debug_Var   =>  1       //When T_KEY_DEBUG is 1, if the variable memory allocation failed (i.e., insufficient RAM),
                                    //you can open Disable_Debug_Var to release part of RAM to solve the problem , default:0
//-----------------------------------------------
```

Fig.4-22：Touch test switch settings

(1) Touch test main switch：（**T_Key_Debug**）

Open to see the open channel touch signal in the Log;

(2) Continue scan switch：（**Continuous_Debugging**）

On continuous output measurement signal, when off only signal jump will output measurement signal to the LOG window (such as touch);

```
T_Key_Debug:
T_Key_Signal=0000
K1-K12TL:TK1  TK2  TK3  TK4  TK5  TK6  TK7  TK8  TK9  TK10 TK11 TK12
K1-K12R1:触摸实时值--> Touch Real-Time Value
K1-K12R1:1D2  1E1  1D3  000  000  000  000  000  000  000  000
K1-K12Rf:触摸环境/参考值 -->Touch Environment /Reference Value
K1-K12Rf:1D2  1E1  1D3  1D3  000  000  000  000  000  000  000  000

Debug_Num=1 , T_Key_Signal=0000
K1-K12TL:TK1  TK2  TK3  TK4  TK5  TK6  TK7  TK8  TK9  TK10 TK11 TK12
K1-K12R1:1D3  1E1  1D3  1D3  000  000  000  000  000  000  000  000
K1-K12Rf:1D2  1E1  1D3  1D3  000  000  000  000  000  000  000  000

Debug_Num=2 , T_Key_Signal=0000
K1-K12TL:TK1  TK2  TK3  TK4  TK5  TK6  TK7  TK8  TK9  TK10 TK11 TK12
K1-K12R1:1D3  1E1  1D3  1D3  000  000  000  000  000  000  000  000
K1-K12Rf:1D2  1E1  1D3  1D3  000  000  000  000  000  000  000  000

Debug_Num=3 , T_Key_Signal=0000
K1-K12TL:TK1  TK2  TK3  TK4  TK5  TK6  TK7  TK8  TK9  TK10 TK11 TK12
K1-K12R1:1D3  1E1  1D3  1D3  000  000  000  000  000  000  000  000
K1-K12Rf:1D2  1E1  1D3  1D3  000  000  000  000  000  000  000  000

Debug_Num=4 , T_Key_Signal=0000
K1-K12TL:TK1  TK2  TK3  TK4  TK5  TK6  TK7  TK8  TK9  TK10 TK11 TK12
K1-K12R1:1D3  1E1  1D3  1D3  000  000  000  000  000  000  000  000
K1-K12Rf:1D2  1E1  1D3  1D3  000  000  000  000  000  000  000  000

Debug_Num=5 , T_Key_Signal=0000
K1-K12TL:TK1  TK2  TK3  TK4  TK5  TK6  TK7  TK8  TK9  TK10 TK11 TK12
K1-K12R1:1D3  1E1  1D3  1D3  000  000  000  000  000  000  000  000
K1-K12Rf:1D2  1E1  1D3  1D3  000  000  000  000  000  000  000  000

Debug_Num=6 , T_Key_Signal=0000
K1-K12TL:TK1  TK2  TK3  TK4  TK5  TK6  TK7  TK8  TK9  TK10 TK11 TK12
K1-K12R1:1D3  1E1  1D3  1D3  000  000  000  000  000  000  000  000
K1-K12Rf:1D2  1E1  1D3  1D3  000  000  000  000  000  000  000  000
```

Fig.4-23：Debug Log Shows

(3) Release part of RAM：（**Disable_Debug_Var**）

When T_KEY_DEBUG is 1, if the variable memory allocation failed (i.e., insufficient RAM), you can open Disable_Debug_Var to release part of RAM to solve the problem;

Attention：When burning, the above three enable must be closed.

11. Main function description and related register definition

```
Lib_H_Bottom:
//***********************************///
//  Description of main functions in the library
//***********************************//
//T_Key_Channel_Setting T_Keyx  //Manual T_KEY port setting, X denotes constant
//Env_Fix   T_Keyx              //Force the repair environment value to the current actual value, where x represents a constant
//void  TK_Init_Auto(void);     //Automatically initialize the T_KEY_VREF and Const_EN_CH_T_KEYX registers
                                //If you change channels halfway, use the T_KEY_SCAN_REG register,
                                //and the corresponding Const_EN_CH_T_KEYX should be set to 1
//void  T_Key_Scan(void);       //T_KEY scan function (non-blocking step-by-step polling)
//void  T_Key_Data_Ref_Initial(void);   //Initializes the first touch environment modifier
//void  Get_T_Key_Signal(void); //According to the TK scan results, the key is determined and the key signal is given T_Key1_Signal - T_Key12_Signal
//void  Uart_Auto(void);        //UART is automatically initialized
//void  Sleep_Mode(void);       //Sleep mode, wake according to Const_Wakeup_CH_T_Keyx and Const_EN_CH_T_Keyx
//T_Key_Scan_Reg                //TK scan register (16 bits)
//T_Key_Signal                  //Key flag register (16 bits)
//-----------------------------------*/
```

Fig.4-24：Description of main functions in the touch library

- **T_Key_Scan_Reg** —— TK scan register (16 bits)

Bit	Initial Value	R/W	Description
15	0	-	Reserved（Set to 0）
14	0	-	Reserved（Set to 0）
13	0	-	Reserved for use as an interrupt flag
12	0	R/W	T_Key12 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key12 Set to 1 to be effective）
11	0	R/W	T_Key11 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key11 Set to 1 to be effective）
10	0	R/W	T_Key10 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key10 Set to 1 to be effective）
9	0	R/W	T_Key9 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key9 Set to 1 to be effective）
8	0	R/W	T_Key8 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key8 Set to 1 to be effective）
7	0	R/W	T_Key7 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key7 Set to 1 to be effective）
6	0	R/W	T_Key6 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key6 Set to 1 to be effective）
5	0	R/W	T_Key5 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key5 Set to 1 to be effective）
4	0	R/W	T_Key4 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key4 Set to 1 to be effective）
3	0	R/W	T_Key3 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key3 Set to 1 to be effective）
2	0	R/W	T_Key2 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key2 Set to 1 to be effective）
1	0	R/W	T_Key1 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key1 Set to 1 to be effective）
0	0	-	T_Key query scan alternately end flag bit, reserved for Scan _ End flag

Attention：The T _ Keyx terminal to be used in the program is valid only when it is set to 1 in the PT _ Lib _ xxxx. H file;

This register is used only when the T _ Keyx pin needs to be changed halfway through the process. In general, only the TK _ Init _ Auto () function needs to be called during initialization. The system automatically initializes the T _ Key _ Scan _ Reg register based on Const _ T _ Key _ VRef and Const _ EN _ CH _ T _ Keyx, so it is not necessary to set the T _ key _ Scan _ Reg register.

- **T_Key_Signal —— Key Flag Register（16Bits）**

Bit	Initial Value	R/W	Description
15	0	-	Reserved（Set to 0）
14	0	-	Reserved（Set to 0）
13	0	-	Reserved for use as an interrupt flag
12	0	R/W	T_Key12 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key12 Set to 1 to be effective）
11	0	R/W	T_Key11 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key11 Set to 1 to be effective）
10	0	R/W	T_Key10 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key10 Set to 1 to be effective）
9	0	R/W	T_Key9 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key9 Set to 1 to be effective）
8	0	R/W	T_Key8 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key8 Set to 1 to be effective）
7	0	R/W	T_Key7 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key7 Set to 1 to be effective）
6	0	R/W	T_Key6 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key6 Set to 1 to be effective）
5	0	R/W	T_Key5 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key5 Set to 1 to be effective）
4	0	R/W	T_Key4 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key4 Set to 1 to be effective）
3	0	R/W	T_Key3 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key3 Set to 1 to be effective）
2	0	R/W	T_Key2 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key2 Set to 1 to be effective）
1	0	R/W	T_Key1 Scan and energize，0/1：Disable/Enable（Const_EN_CH_T_Key1 Set to 1 to be effective）
0	0	-	T_Key query scan alternately end flag bit, reserved for Scan _ End flag

12. PMS164 and PMS161 coping strategy description

It is mainly suitable for anti-jamming and extra-long touch applications, such as anti-intercom interference and fish-drift applications).

Anti-jamming Strategy:

(1) Anti-jamming Enable：（**ANTIJAM_Enable**）

(2) Anti-jamming reference channels 1 and 2：（**Const_T_Key_Antijam_Ref_1**、**Const_T_Key_Antijam_Ref_2**）

(3) Environment Value Repair Speed：（**CK_Fix_Count**）

When the number of times of no interference is greater than the value, the environment value is fixed, the range is 0~210;

(4) Environment Value Repair Ratio：（**Antijam_Fix_Speed**）

The environment value approaches the actual value at a certain speed. The larger the value is, the slower the repair speed is. Specifically, the difference between the previous environment value and the actual value is divided by 2^n.

(5) Ref channel jitter threshold：（**ANTIJAM_REFERENCE_VALUE**）

If the jitter of the actual value of the reference channel exceeds this value, it is judged abnormal.

(6) Touch depth threshold setting：（**ANTIJAM_REFERENCE_SNECEx**）

Indicates the false trigger threshold when the touch channel is strongly interfered, specifically, it cannot exceed (220-sensitivity of the reference channel) × the value;

Extra Long Touch Strategy：

(1) Extra Long Touch Enable：（**LongLong_Touch_Release_Enable**）

(2) Abnormal key release detection rate：（**LongLong_Touch_Fix_Count**）

When the number of times reaching the release point is greater than this value, the environment value is fixed;

(3) Abnormal key release detection sensitivity：（**LongLong_Touch_Fix_Sensi**）

A release point is set, and that release position is the difference between the environment value move down from the original release position and the actual value divided by 2 ^ n;

```
//Anti-interference parameter setting
ANTIJAM_Enable              => 1        //Anti-jamming countermeasure switch  1:Enable  0:Disable
Const_T_Key_Antijam_Ref_1   => 'TK9'
Const_T_Key_Antijam_Ref_2   => 'TK10'
CK_Fix_Count                => 20       //Environmental restoration rate,Range:0-255
Antijam_Fix_Speed           => 4        //Environment value repair ratio, the environment value to the actual value at a rate of 1/(2^n) difference
ANTIJAM_REFERENCE_VALUE     => 40       //Refer to setting the jitter threshold of the channel

ANTIJAM_REFERENCE_SNECE1    => 5
ANTIJAM_REFERENCE_SNECE2    => 5
ANTIJAM_REFERENCE_SNECE3    => 5

//-------------------------------------
//Extra Long Touch Strategy,
LongLong_Touch_Release_Enable   => 1        //Ultra long touch release is enabled
LongLong_Touch_Fix_Count        => 20       //Abnormal key release detection rate
LongLong_Touch_Fix_Sensi        => 1        //Abnormal key release detection the greater the sensitivity, the more sensitive,
                                            //too large easy to mistakenly release,Recommended values[1-4]
```

Fig.4-25：PMS164 and PMS161 coping strategy description

13. Description of PMS160 parameter multi write

(1) Multiple write parameter enable：（**En_Mult_Write_Para**）

Multiple firing parameters Const_SEN_T_KeyX(Sensitivity)、Const_IFC_Count(Count limit of Module0 IFC)、Const_IFC_Timing(Reading time of Module1 IFC)

(2) Setting of maximum allowable multiple write times:（**Mult_Write_Para_MaxCnt**）

Value range: 1-8, default value: 3;

(3) Current multiple write times：（**Mult_Write_Para_Cnt**）

Current multiple write times，0 means no multiple write，1 means the first multiple write 1.

value range: 0-Mult_Write_Para_MaxCnt，default value: 0

```
//Parameter multiple write settings
En_Mult_Write_Para      =>  1       //Multiple parameter enable settings,
                                    //Refiring parameters Const_SEN_T_KeyX(Sensitive)、Const_IFC_Count(Count limit of Mode0 IFC)、Const_IFC_Timing(Reading time of Mode1 IFC)
                                    //Default option: 0         Note: Other parameters cannot be modified; otherwise, multiple write may fail
Mult_Write_Para_MaxCnt  =>  3       //Setting of maximum allowable multiple write times
                                    //Value range : 1-8
                                    //default value : 3
Mult_Write_Para_Cnt     =>  0       //Current multiple write times, 0 means no multiple write, 1 means the first multiple write 1
                                    //value range, 0-Mult_Write_Para_MaxCnt
                                    //default: 0
```

Fig.4-26：PMS160 parameter multiple write settings

4.2.2. User function file

User function file User_Function. C is mainly used for IO and user variable initialization, the realization of the user-defined function module.

1. Definition and initialization of user variable：

Users can Define their own variables under Variable_Define, and initialize the variable under Variable_Init(void) function.

```
UserFile_TOP:
//*******************************************************//
//User's Variable Define
Variable_Define:
//Insert variable define Code

//User's variable initialization Settings.
void    Variable_Init(void)
{
    //Insert variable Initial Code
}
```

Fig.4-27: Variable definition and initialization

2. IO initialization：

The IO_Init(void) function is used by the customer to configure the input state and digital/analog enablement of touch IO and other IO. For unused IO, set it to output low or input pull-high (which is good for system stability and power saving).

```
//IO Settings, part of IO p-touch has been set
void    IO_Init(void)
{
    //---------------------------
    PA      =   0b0000_0000;
    PAC     =   0b1101_0111;    //1:output   0:input
    PAPH    =   0b0000_1000;    //1:pull high 0:not pull high
    PAPL    =   0b0000_0000;    //1:pull low 0:not pull low
    PB      =   0b0000_0000;
    PBC     =   0b0000_1111;
    PBPH    =   0b0000_0000;
    PBPL    =   0b0000_0000;
    PC      =   0b0000_0000;
    PCC     =   0b1111_1111;
    PCPH    =   0b0000_0000;
//  PCPL    =   0b0000_0000;

    PADIER  =   0b0000_1000;    /*CS pin and corresponding T_key are set to analog I/O(set to 0),
                                and UART_IO is set to digital I/O(set to 1) when using host computer.*/
    PBDIER  =   0b0000_0000;
    PCDIER  =   0b0000_0000;

    //---------------------------
    //User's Other IO Define
    //---------------------------
}
```

Fig.4-28：IO initialization settings

3. User-defined function

The user-defined function is written in the T _ Key _ Func (void), and the corresponding key function is written in the corresponding modules. The enable configuration is in the PT Lib xxxx.h file.

Take T _ Key1 as an example, as shown in the following figure. If the program jumps into the if statement, it indicates that the touch key is triggered; If the program jumps into the elseif statement, the touch key is released or there is no touch.

```
void    T_Key1_Func(void)
{
    if(T_Key1_Signal==1 && Pre_T_Key1_Release==1)
    {
        Pre_T_Key1_Release  =   0;
    //-----------------------------------------//

        //User can add code

    //-----------------------------------------//
    }
    elseif(T_Key1_Signal==0)
    {
        Pre_T_Key1_Release  =   1;
    }
}
```

Fig.4-29：Key function setting

Other user functions can be written in the User _ other _ Func (void) function;

```
void    User_other_Func(void)
{
    //User's Code
}
```

Fig.4-30：Other function settings

4. Slider module procedures

The slider module is divided into general slider and direction slider; the direction slider is divided into positive direction and negative direction.

6 keys 11 levels is our maximum feasible range of slider modules. The relationship between the level of the slider m and the number of keys n is: m= 2n-1. A single key cannot achieve slider effect. Take the lowest two keys and three steps as an example, the three levels are: press key 1 alone, press key 1 and key 2 at the same time, press key 2 alone.

```
Void    Slider_Function(void)
{
    byte    shift   =   SliderA_Gear_Range & 0x1f;
    switch(shift)   //Actions are generated according to the gear signal
    {
    case 1:
        //User's Code
        break;
    case 2:
        //User's Code
        break;
    case 3:
        //User's Code
        break;
    case 4:
        //User's Code
        break;
    case 5:
        //User's Code
        break;
    case 6:
        //User's Code
        break;
    case 7:
        //User's Code
        break;
    default:
    case 0: //Release the slider
        //User's Code
        break;

    }
}
```

Fig.4-31: Slider Settings

5. Writing Sleep Function in Power Saving Mode

#define Sleep_Condition (0)：Enter power saving sleep mode conditions

If the user selects the power saving mode, the configuration of number is set to 1 by default after the program is generated, and the user can add the required sleep judgment conditions, for example, **#define Sleep_Condition** (PA.0==0). In the normal working mode, the number in the bracket is 0, which represents no sleep; the program setting before the power saving mode can be completed in the Pre _ sleep _ set (void) function;

```
//Enter Power Saving Mode Conditions
#define Sleep_Condition (1)         //For example (pa.0 ==0 & variable T = 100) 1 represents unconditional
//Action before entering power saving mode
void    Pre_sleep_set(void)
{
    nop;
    //Action before entering power saving mode, such as turning off lights, etc
    //The wake condition is set in pt_lib.h
}
```

Fig.4-32: Pre-sleep function settings

There are two ways to wake up: touch key and IO. The actions are different according to different ways. Users can set the actions according to their needs.

```
//Wake up action
void    After_wakeup_set(void)
{
    if(Wakeup_Signal==1)
    {
        Wakeup_Signal = 0;  //Clear it in time, otherwise it will affect the next power-saving mode
                            //Touch wake-up, which channel can be read by Pre_T_Key_Signal if necessary
    }
    else
    {
        //IO Wake up
    }
}
```

Fig.4-33：Wake up action settings

6. Other registers and variables

(1) Pre_T_Key_Release　Key release flag register（16 bits）;

Bit	Initial Value	R/W	Description
15	0	-	Reserved (Set to 0)
14	0	-	Reserved (Set to 0)
13	0	-	Reserved (Set to 0)
12	0	R/W	Pre_T_Key12_Release
11	0	R/W	Pre_T_Key11_Release
10	0	R/W	Pre_T_Key10_Release
9	0	R/W	Pre_T_Key9_Release
8	0	R/W	Pre_T_Key8_Release
7	0	R/W	Pre_T_Key7_Release
6	0	R/W	Pre_T_Key6_Release
5	0	R/W	Pre_T_Key5_Release
4	0	R/W	Pre_T_Key4_Release
3	0	R/W	Pre_T_Key3_Release
2	0	R/W	Pre_T_Key2_Release
1	0	R/W	Pre_T_Key1_Release
0	0	R/W	Reserved (Set to 0)

Attention： This register is not read or written to the library and is only used in the **T_Key_Function()** function of user_function.c. Users can perform read and write operations by themselves.

(2) T_Key1_Data_Ref - T_Key12_Data_Ref（word）：TK1-TK12 Environment value variable（R/W）;

(3) T_Key1_Data_Real - T_Key12_Data_Real（word）：TK1-TK12 Real value variable（R/W）;

(4) SliderA_Gear_Range（byte）：Slider gear，range1-11（R/W，It will not be cleared automatically）;

4.2.3. User main project file

1. Work Mode：normal mode and power saving mode

The main program mode state is selected in **P-Touch** and then generated in the program. Once the working mode is selected, this mode cannot be changed twice. If you want to switch mode state, please generate a new program in **P-touch**.

```
//-------Main program mode selection------------------
Const_Work_Mode =>  1   //Range 1-2
                        //1 :   mode 1  Normal mode : Touch sensitive, no dormancy, can be added timer
                        //2 :   mode 2  Power saving mode : With sleep, can wake up, can use the timer
```

Fig.4-34：Working mode setting

2. Function description

In normal mode: non-sleep mode, the initialization functions include **IO_Init()**, **Variable_Init()**, and **TK_Init_Auto()**;The main functions include **T _ Key _ Scan ()**, **T _ Key _ Warning ()**, **T _ key _ Process ()**, **T _ key _ Function ()** and **User _ other _ Function ()**, the subfunctions should be changed under the **User function.C** file.

To use interrupts, select the interrupt switch during initialization.

```
    IO_Init();      //IO Initial
    Variable_Init();
    //  Insert Initial Code
//  En_Interrupt;   //Turn on global interrupt  //Use this notation to break an interrupt when using UART
//  Dis_Interrupt;  //Close the interrupt       //Use this way of writing when using UART
    TK_Init_Auto(); //Initializes Tk and automatically configures the register according to the default constant Const_EN_CH_T_Keyx
                    //If you change channels halfway, use the T_KEY_SCAN_REG register,
                    //and the corresponding Const_EN_CH_T_KEYX should be set to 1

    while (1)
    {
//      .wdreset;
        //-------------------------------------------------------------//
        //mode 2    Power saving mode : Need to set wake-up
        T_Key_Scan();           //Scan T_Key
        if(T_Key_Scan_End)      //Touch key polling is over
        {
            T_Key_Process();        //Process the TKEY data and determine the touch status
                                    //(if the slider is open, determine the slider status together)
            T_Key_Function();       //Touch keys set up after command


            //----------------------
            User_other_Func();  //The user's function is called here.
            //----------------------
        }
        //-------------------------------------------------------------------------------//
//      ...
//      wdreset;

    }
```

Fig.4-35：Description of functions in working mode

In power-saving mode, there are more sleep and wake functions than in normal mode, including the subfunctions **Pre_sleep_set()**, **Sleep_Module()**, and **After_wake_up_set()**.the subfunctions should be changed under the **User function.C** file.

```
if(T_Key_Signal==0 && Pre_T_Key_Signal==0 && Sleep_Condition)   //No keys and no need to connect UART to send data automatically into the power-saving mode
{                   //Other restrictions can be added,Like to turn off the lights: if(T_Key_Signal==0 && LED==0)
    Pre_sleep_set();//First turn off PWM and other power consumption items
    Sleep_Mode();   //This is blocked sleep, only to wake up to jump out of sleep
    After_wakeup_set();//First turn off PWM and other power consumption items
}
```

Fig.4-36：Description of sleep and wake up functions

4.2.4. Use of T-Watch

T-watch is the upper computer for touch sensitivity debugging, which can intuitively observe the state changes of touch keys. The following is a detailed introduction to the use of T-Wacth:

1. Firstly, when P-Touch is used to generate the touch program frame, T-Watch is selected to be enabled, and wake up mode and appropriate communication port are set;

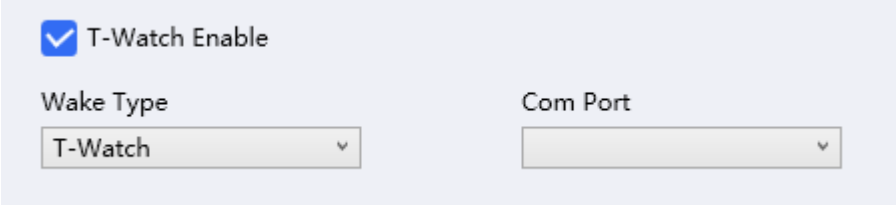


Fig.4-37：Enable T-Watch Settings

Attention：For the engineering scheme, only PA0 is supported for Uart communication during simulation, and the actual chip can support PA0, PA5 and PB0;

2. Select Tool in the P-Touch menu and click to open T-Watch. As shown in the figure below, Uart is used to connect the touch simulation board;

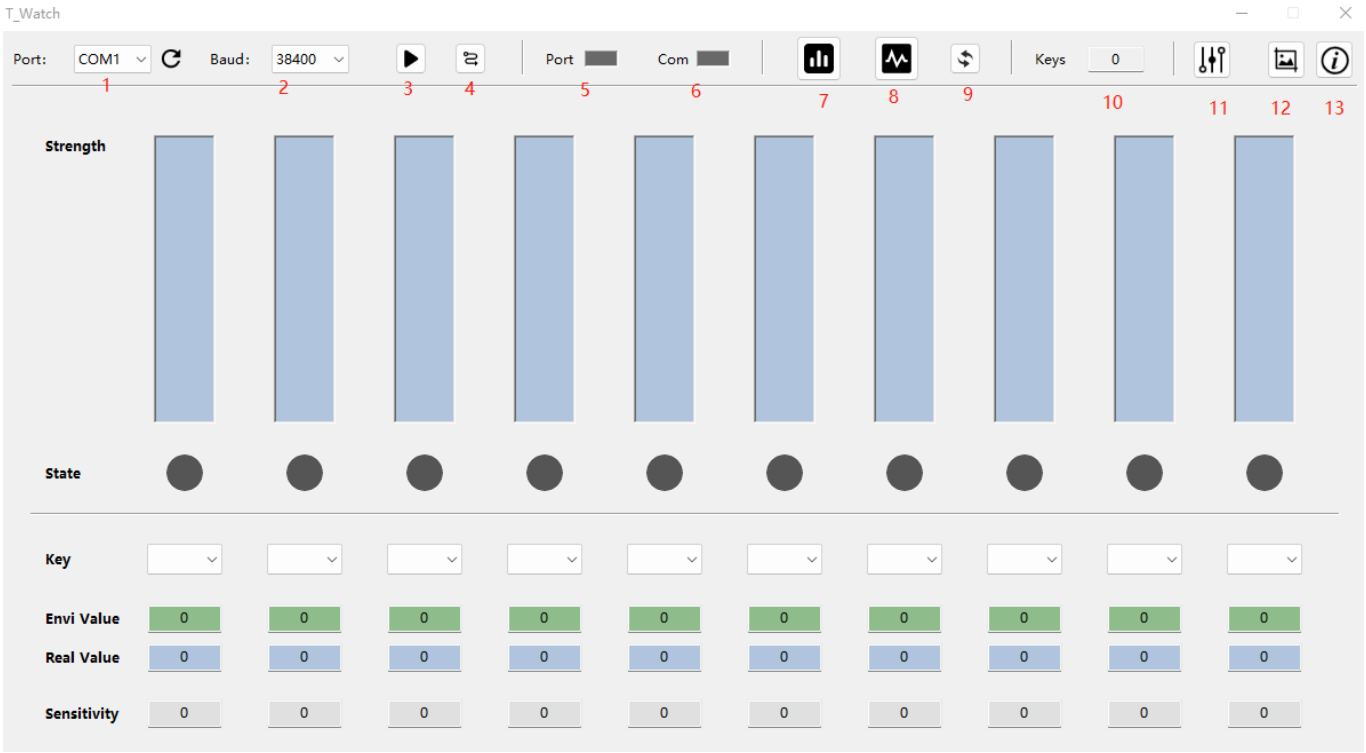


Fig.4-38：T-Watch Initial Interface

The following describes the red label functions of the main menu in the figure:

Mark 1 shows the communication port; In general, after the computer is connected to the UART, the T-Watch will automatically identify the serial port number. If the serial port number is not identified, click the refresh button next to it to obtain it again;

Mark 2 shows the baud rate of serial communication. The default setting is 38400. If you need to change it, please be sure to keep the baud rate consistent with that in the program framework generated by P-touch; otherwise, communication will not be possible;

Mark 3 is the start button, which means to open the communication serial port. If the configuration and line are

normal, the communication conditions will be available after clicking;

Mark 4 only works when the wake-up mode is set to T-WACTH. After clicking, a signal will be sent to wake up the MCU. If low voltage wake is selected, this button function can be ignored;

Mark 5 is used to show whether the communication port of the computer is connected properly. If it is normal, it will show green;

Mark 6 is used to display the communication status between T-Watch and MCU. If the display flashes yellow, it indicates that the communication is normal;

Mark 7 display mode 1 and the intensity display above;

Mark 8 display mode 2, in the form of waveform display;

Mark 9 is the initialization display, click it will re-initialize communication;

Mark 10 shows the number of open touch channels in the program;

Mark 11 PMS160 debugging and setting module on board can be used only after serial communication is enabled;

Mark 12 is used to quick screenshot, picture automatically saved on the user's computer desktop;

Mark 13 is the operation instruction of T-Watch;

Attention：When using UART, 1K ohm resisters need to be connected in series to TX terminal, and only CP2101 and CH340(G) modules are supported. For details, please refer to the instructions in T-Watch.

3. The following two display modes are introduced

(1) Display 1: Intensity display

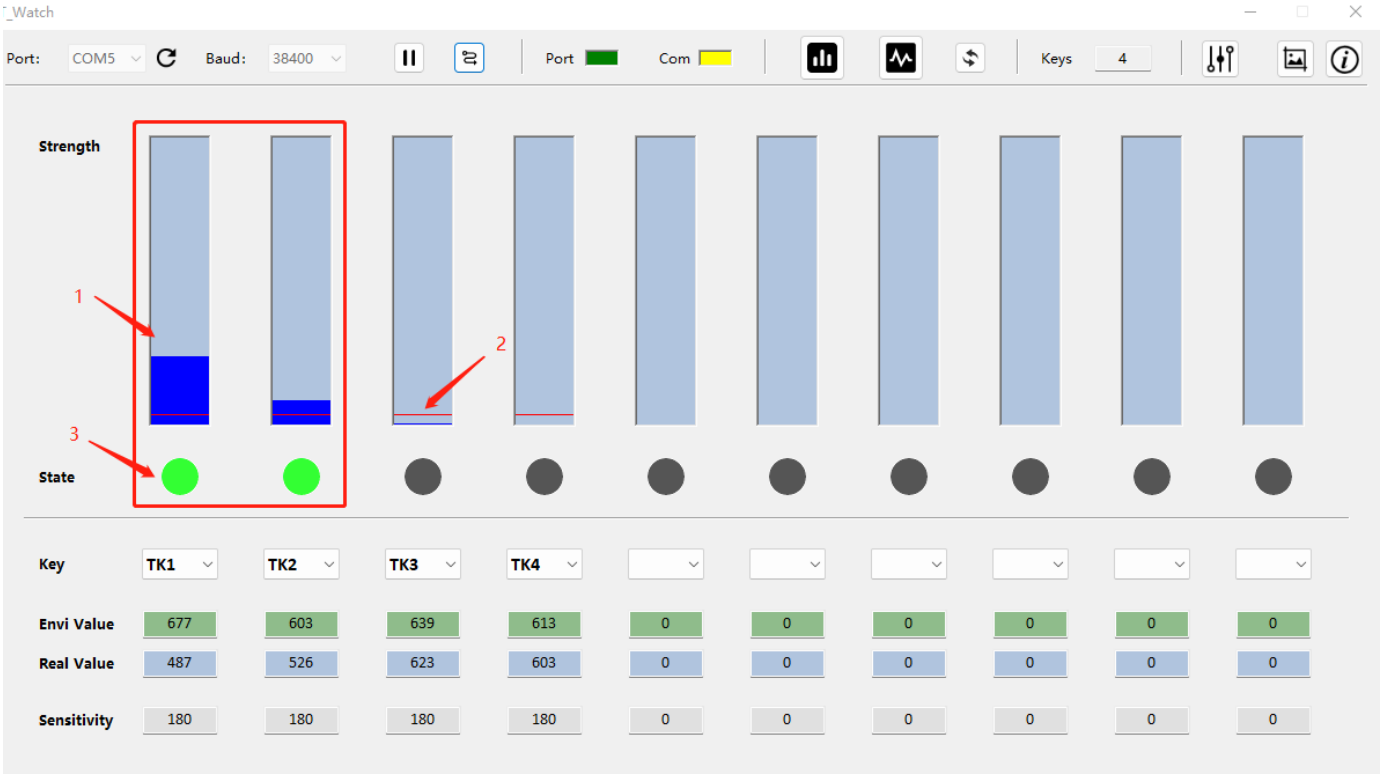


Fig.4-39：Trigger intensity display

The intensity display interface can judge whether the touch button is triggered and the trigger state in real time according to the sensitivity and other parameter options set in the program. When there is an external key, the blue column status bar marked 1 in the figure will rise, and the higher its length is, the higher the touch intensity is. When its height exceeds the height of the red line marked 2, the key status light marked 3 will be on, indicating that the key is triggered; The corresponding key channel, real-time environmental value,

real-time actual value and sensitivity are displayed below each status bar. When the height exceeds the height of the red line 2, the button status indicator 3 turns on, indicating that the button is triggered.

(2) Display 2:Graphical display

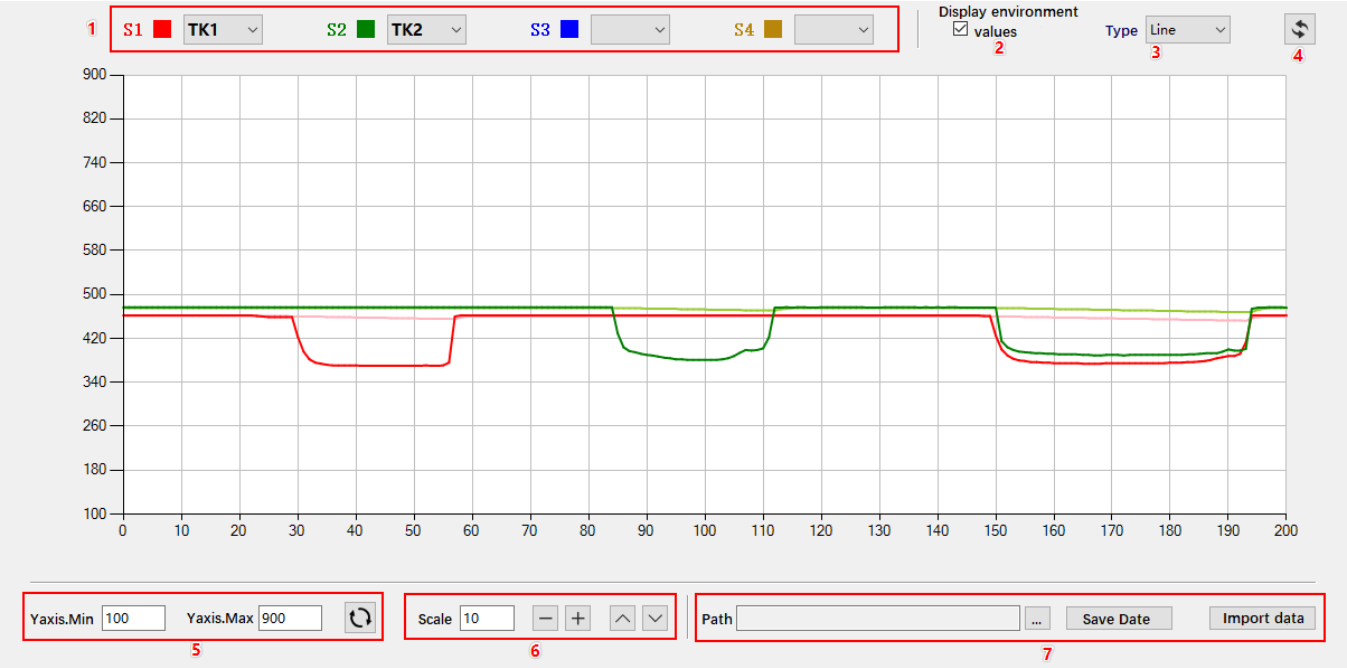


Fig.4-40：Display of trigger graphical

Mark 1 is the channel selection currently displayed. Different colors correspond to the linear color of the display area below. At the same time, touch data of up to four channels can be displayed;

Mark 2 shows the environment value, and the linear color of the environment value of each channel is a relatively dark color;

Mark 3 is the display style switch, which can choose line, spline, and point and bar display;

Mark 4 is data refresh display, the graph will be displayed again after clicking;

Mark 5 is used to debug the Y-axis starting value. By default, the program will set a relatively appropriate initial starting value according to your initial touch value after running. Of course, you can also modify your own starting value.

Mark 6 is used to fine-tune the upper and lower position of the graph display and the size scaling of the graph;

Mark 7 can save touch data or open it with T-watch to display touch data;

4. Use of PMS160 On-board Debugging Function

(1) To use PMS160 for on-board debugging, you need to enable T _ Watch in P _ Touch and check the on-board debugging function. At present, only PMS160 supports this function;

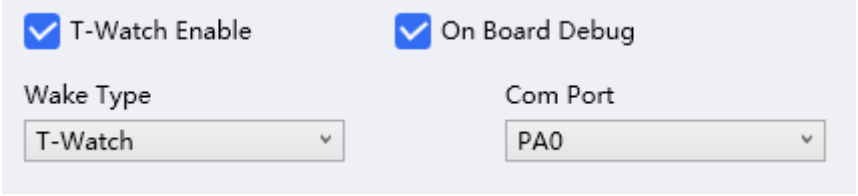
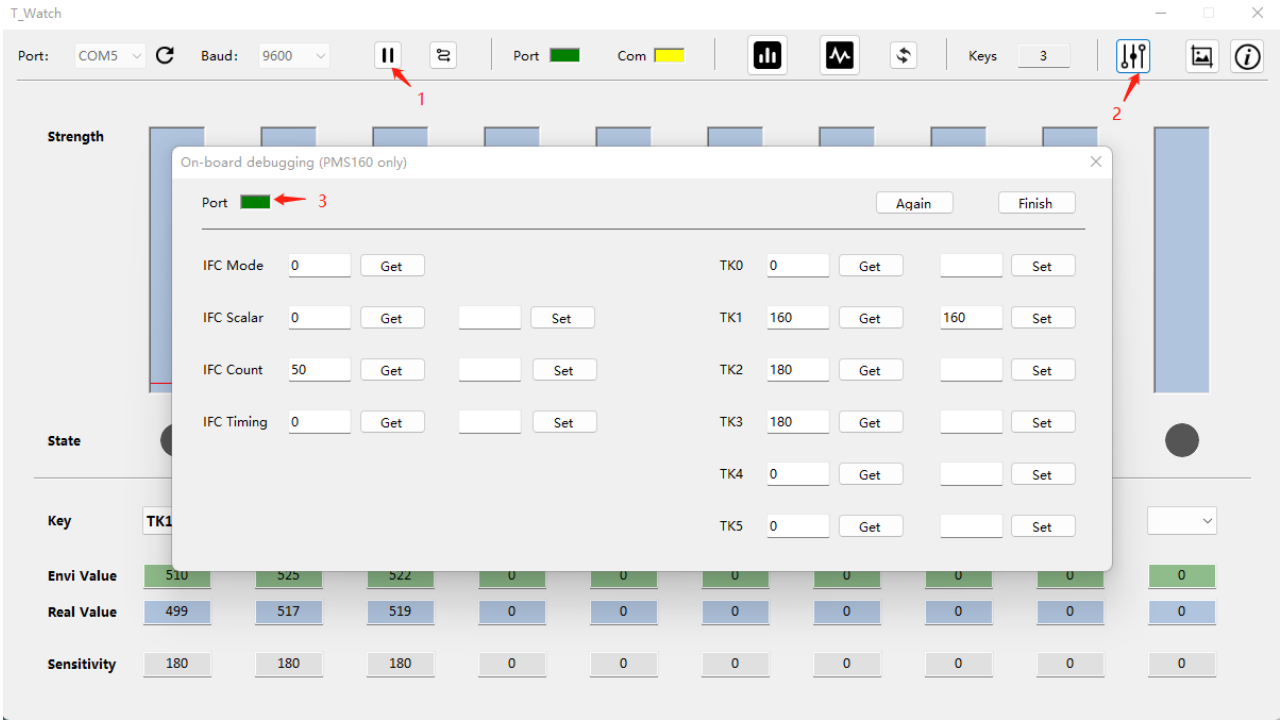


Fig.4-41：Enable the onboard debugging function of PMS160

(2) If this function is required to modify parameters, first ensure that the serial port communication is enabled (marked 1 in the following picture), pay special attention to check whether the communication baud rate is

consistent, and open the debugging window of PMS160 in T_Watch (marked 2 in the following picture). At this point, T_Watch is in wait mode, and then the test board is powered on. After this process is complete, the test board will communicate with T_Watch to determine whether to enter in-board debugging mode. If the communication indicator in the board debugging window changes from grey to green (mark 3 in the figure below), it indicates that the on-board debugging mode has entered. Users can obtain parameters through Get and modify parameters through Set;



4.3. Coping strategy to CS testing

4.3.1. Overview

In order to help relevant touch products pass the CS test, this method is specially developed;

4.3.2. PCB Layout

It is required to follow the relevant instructions in the annex.

Capacitive screen PCB

4.3.3. Notes

The following instructions are required for software parameters:

(1) CS coping strategy enablement requires opening and selecting at least one reference channel.

(2) The reference voltage of the reference channel shall be less than the reference voltage of Tk. If there is no special requirement for the correction base of the reference channel, please keep the default value of 1.

(3) The quick recovery switch must be turned off.

(4) The sensitivity of the relevant TK touch channel should not be too high, which is generally recommended (90~180).

(5) The CS capacitor should not be too large, generally about 10NF is recommended.

Special instructions:

The CS test does not use the power saving mode. If the power saving mode is used, do not sleep frequently to avoid too much influence. Generally, the system standby (no operation, no output) about 30 seconds before entering the sleep mode.

When the CS capacitance of the chip is greater than 10nF and the CS coping strategy is enabled, the touch register should be changed to 64CLK or 128CLK.

5. Slider and Slip Ring Scheme

This module is used to generate slider and slip ring program frame, please configure according to the arrow index [Summary], [Select Keys], [Parameters], etc., and then generate program, users can complete their own functions in the corresponding module in the framework.

5.1. Description of the scheme configuration

5.1.1. Summary

Set basic project information first, including project name and project path; select chip name and package information; Currently, only PFC460 is supported (Multiple FPPA can be enabled or not as required) and the running IDE version; the button to the right of package information allows you to view the currently selected package diagram. When the upper computer is turned on, the communication port needs to be configured. The introduction of the upper computer will be discussed later. On the right is a brief introduction to the use of slider and slip ring library.

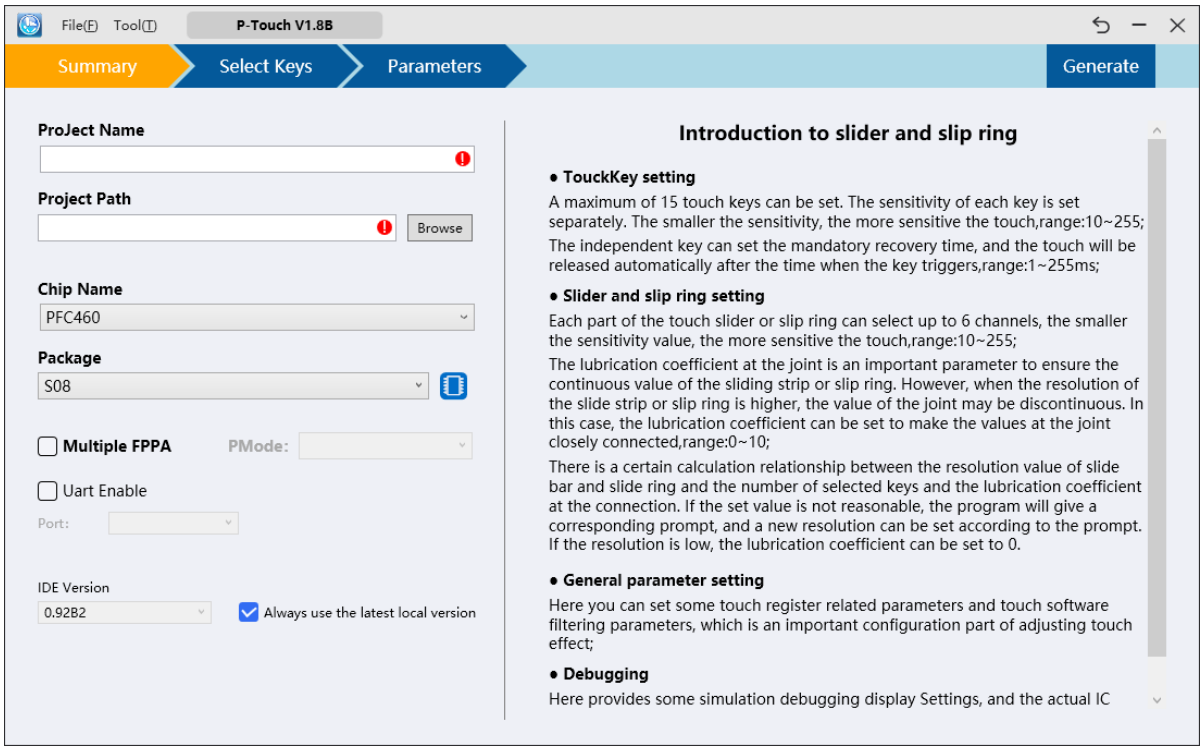


Fig.5-1：Slider and slip ring scheme interface

5.1.2. Select Keys

First, select a pin from the pin list(Step1), and then click the corresponding Move Right button(Step2), set the channel as a slider, ring, or key (Step 3). Finally, set the resolution of the slider or slip ring and lubrication coefficient of joint (Step4)； Please refer to the following notes when selecting keys:

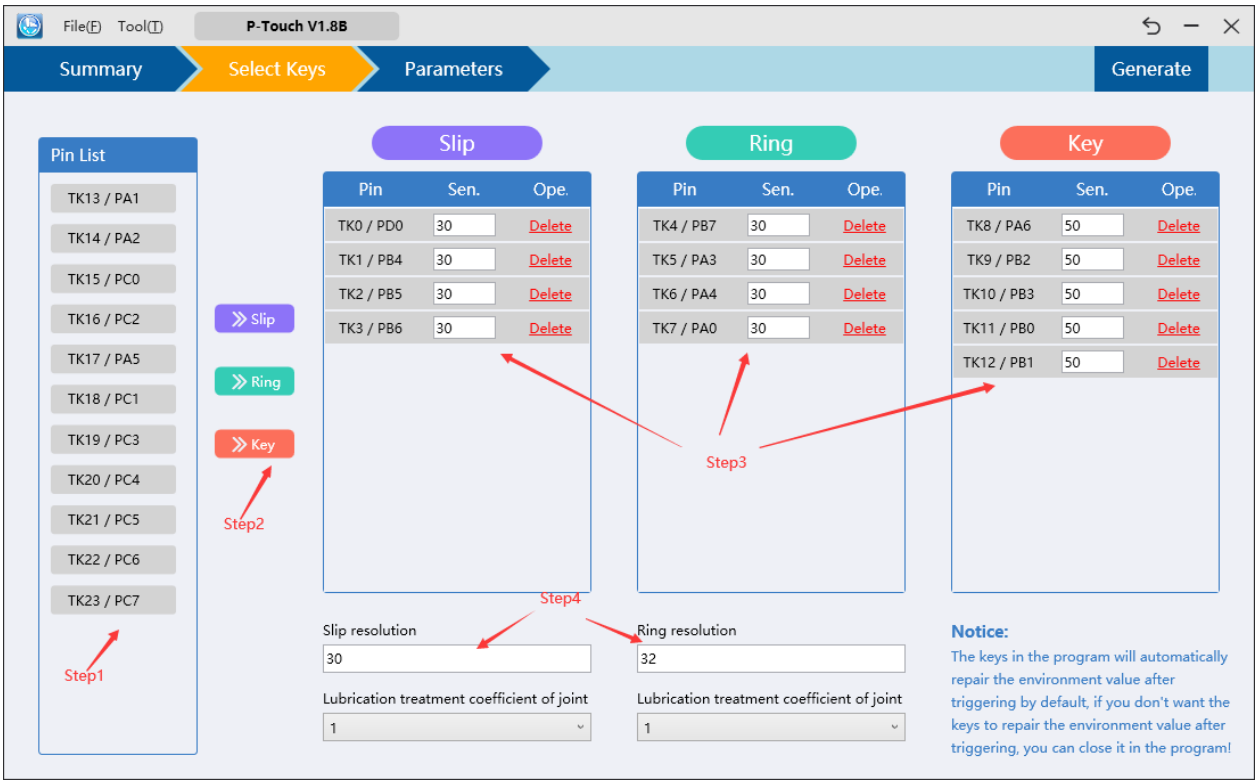


Fig.5-2：keys selection

Precautions:

Slider and slip ring Settings:

Each part of the slider or slip ring can select up to 6 channels. The smaller the sensitivity value is, the more sensitive the touch is. value range：10~255；

Lubrication coefficient at the joint is an important parameter to ensure the continuity of the slider or slip ring in sliding. However, when the resolution of the slider or slip ring is high, the value of the slider or slip ring at the joint may be discontinuous. At this time, the value of the joint can be closely connected by adjusting the lubrication coefficient, the value range: 0~10;

There is a certain calculation relation between the resolution of slider and slip ring and the number of keys selection and the lubrication coefficient of joint. If the set value is unreasonable, the program will give a corresponding suggestion, and a resolution can be reset according to the suggestion. When the resolution is low, the lubrication coefficient can be set to 0;

Key Settings:

A maximum of 15 touch keys can be set, and the sensitivity of each key is set separately. After the key is triggered, it will be automatically repaired by default. If you do not want to automatically repair it, you can set it in the program. The smaller the sensitivity value is, the more sensitive the touch is. Value range: 10 ~ 255.

5.1.3. Parameters

General Setting

(1) **TK Scan Clock Source**：That is, the clock that the touch function scans and counts, where the higher the clock frequency, the greater the actual value of the touch. Note that this is not the system clock;

(2) **TK Reference Voltage**：Also known as TK reference voltage, it has an impact on CS capacitor size and touch sensitivity, which can be increased to reduce the external reference capacitance;

(3) **CS Discharge Time**：CS discharge time before touch, generally, the larger the CS capacitor is, the longer the discharge time is needed;

(4) **System Clock**：Set the system clock frequency.

(5) **Smooth Rank**：The filter times are equal to 2 plus 2 ^ (smooth rank) . For example, if the smooth rank is 2, the filtering times are 6.

(6) **Dithering allowable range**：If the dither exceeds the setting range and is lower than the sensitivity, the environment value will be corrected;

(7) **CS Pin**：The number of filters equals 2 is equal to the number of filters equals 2 to the power of filters plus 2;

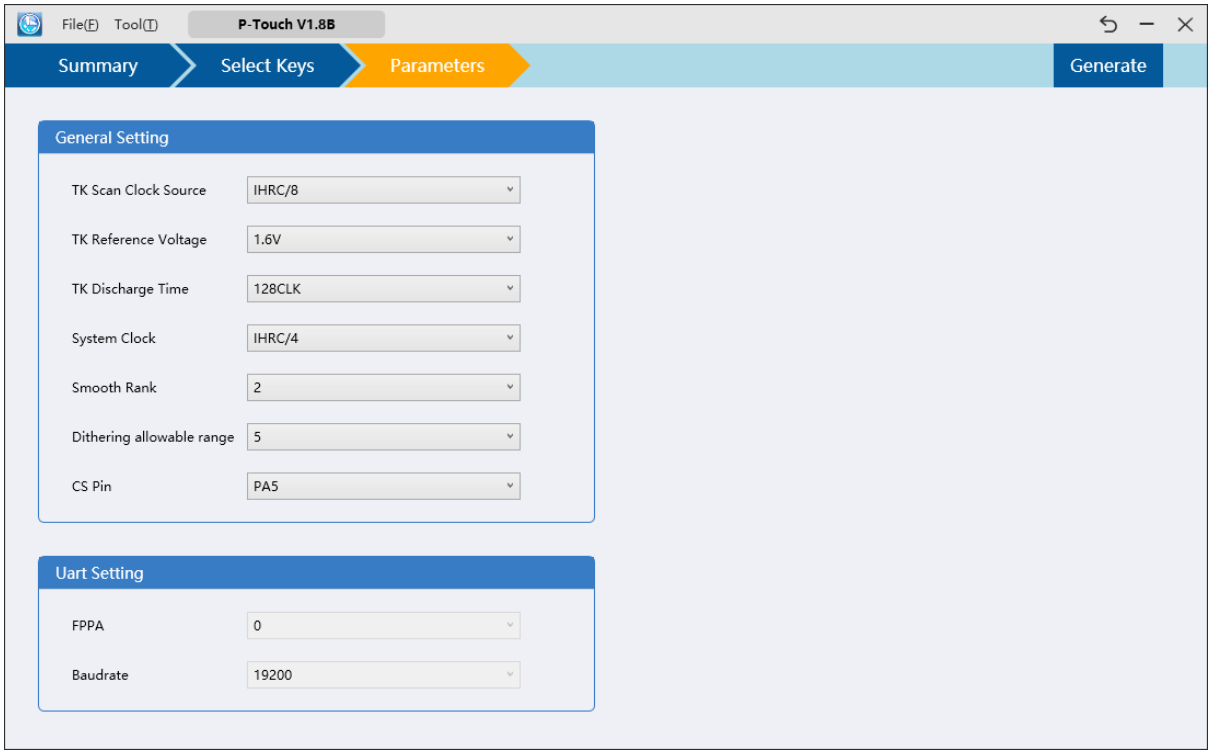


Fig.5-3：Parameter setting

5.2. Generate framework of project

As described above, after configuration as required, click the Generate program button in the upper right corner of the software to generate the program framework, and then the following window will pop up to prompt whether to run the program directly with IDE.

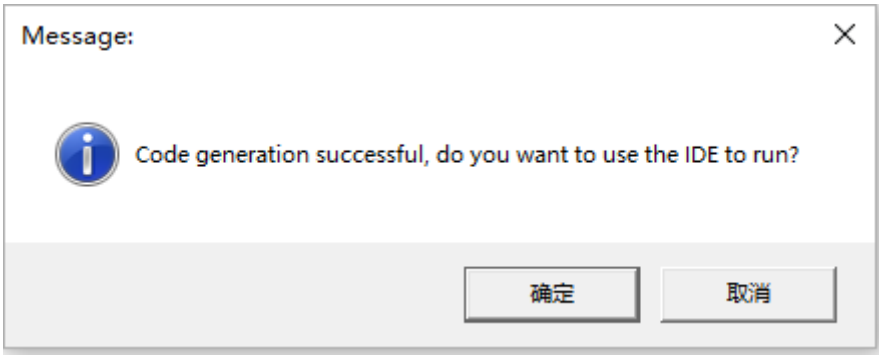


Fig.5-4：Prompt to generate the code framework

After running the generated project with the IDE, you can write actual engineering programs based on this framework. Each project file is introduced in detail below (for example, we generate a Touch_Demo framework, and the file architecture diagram in the program is as follows):

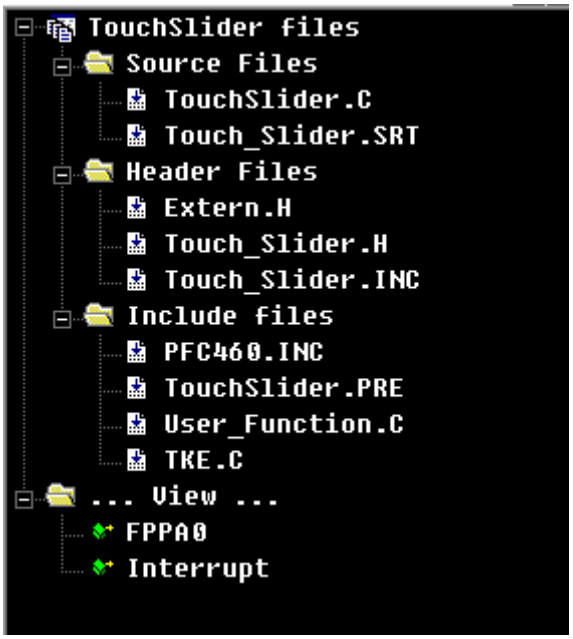


Fig.5-5：Program file architecture

- The Touch _ Demo. C is a main program file, mainly including some initialization functions and functional function calls;
- Touch_slide. H is the configuration file of the slider, slip ring and key, which records the touch channel information, parameter configuration, UART Settings and other information used.
- User_function.C is the function compilation file of the user, in which the user can write the required functions according to the corresponding module's own requirements;

5.2.1. Slider and slip ring library configuration file

Touch library configuration file the Touch_Slider. H file is used to configure settings related to the slider, slip ring, and keys, including channel selection, sensitivity configuration, general parameter configuration, and Uart enablement, etc.

1. Touch key setting（T_Keyx_Set）

First, configure the key channel, and then set the sensitivity of each corresponding channel; By default, the program will automatically repair the environment value after the key is triggered; If you do not want to repair it, you can disable the repair in the program and cancel the comment of the corresponding key Disable _ Press _ Fix _ T _ Key;

```
T_Keyx_Set:         //Touch key (T_Keyx) setting
.C_T_Keyx   PB1,PA2,PA3,PA4,PA5  //Independent touch keys, up to 15 touch keys can be selected
ENUM
{
    //Touch key (T_Keyx)Sensitivity setting
    C_Sen_T_Key1    =   50,             //Touch sensitivity, the smaller the value,the higher the sensitivity; Range (10 ~ 255)
    C_Sen_T_Key2    =   50,
    C_Sen_T_Key3    =   50,
    C_Sen_T_Key4    =   50,
    C_Sen_T_Key5    =   50,
//  Disable_Press_Fix_T_Key1    =   1,  //After opening, close the key to trigger the repair, often used for long touches
//  Disable_Press_Fix_T_Key2    =   1,
//  Disable_Press_Fix_T_Key3    =   1,
//  Disable_Press_Fix_T_Key4    =   1,
//  Disable_Press_Fix_T_Key5    =   1,
}   ;
```

Fig.5-6：Touch key (T_Keyx) setting

2. DrawSlip setting（Draw_Slip_Set）

First, configure the slider channel, and then set the sensitivity of each corresponding channel, slider resolution and lubrication coefficient of joint;

```
Draw_Slip_Set:  //DrawSlip setting
.C_Draw_Slip    PB4,Pb5,PB6,PB7 //DrawSlip setting
ENUM
{
    //DrawSlip Sensitivity setting
    C_Sen_Draw_Slip_T1  =   30,     //Touch slider sensitivity, the smaller the value, the higher the sensitivity;Range (10 ~ 255)
    C_Sen_Draw_Slip_T2  =   30,
    C_Sen_Draw_Slip_T3  =   30,
    C_Sen_Draw_Slip_T4  =   30,
    //-----------------------
    C_Draw_Slip_Resolution= 30, //Slider resolution
    C_Draw_Slip_Libe            =   1   //Connection lubrication treatment, low resolution can be set to 0
    //-----------------------
}   ;
```

Fig.5-7：DrawSlip setting

3. SlipRing setting（Slip_Ring_Set）

First, configure the slip ring channel, and then set the sensitivity of each corresponding channel, slip ring resolution and lubrication coefficient of joint;

```
Slip_Ring_Set:  //SlipRing setting
.C_Slip_Ring    PB5,PA0,PB3 //SlipRing touch keys can be selected. A maximum of 6 keys can be selected. A 4 keys is recommended
ENUM
{
    //SlipRing Sensitivity setting
    C_Sen_Slip_Ring_T1  =   30,     //SlipRing sensitivity, the smaller the value, the higher the sensitivity;Range (10 ~ 255)
    C_Sen_Slip_Ring_T2  =   30,
    C_Sen_Slip_Ring_T3  =   30,
    C_Sen_Slip_Ring_T4  =   30,
    //------------------------
    C_Slip_Ring_Resolution= 32, //SlipRing resolution
    C_Slip_Ring_Libe            =   1   //Connection lubrication treatment, low resolution can be set to 0
    //------------------------
}   ;
```

Fig.5-8：SlipRing setting

4. General setting

(1) Touch clock（C_Touch_Source_CLK）

Value range: 0:ILRC, 1:IHRC/2, 2:IHRC/4, 3:IHRC/8, 4:IHRC/16, 5:IHRC/32, 6:IHRC/64, 7:IHRC/128 (1:reserved)

Default: 3

(2) Touch reference voltage（C_Touch_VRef）

Value range: 0:0.8*Touch Power(TP for short), 1:0.7*TP, 2:0.6*TP, 3:0.5*TP, 4:0.4*TP, 5:0.3*TP, 6:0.2*TP

Default: 0

The selection of touch reference voltage has an impact on the performance of touch sensitivity and the selection of CS capacitance. The higher the theoretical reference voltage is, the higher the sensitivity is;

(3) CS discharge time（C_Touch_Discharge）

Value range: 0: reserved, 1:CLK_32, 2:CLK_64, 3:CLK_128

Default: 3

CS discharge time selection before touch, the longer the time, the cleaner the discharge;

(4) Filter rating（C_Smooth_Rank）

Optional range: 1~6，Number of filter = 2^Filter rating + 2;

(5) Jitter allowed range（C_Shake_Rang）

Beyond the range and below the sensitivity, the environment value will start to correct;

```
General_Set:    //General setting
ENUM
{
    //------------------------
    C_Touch_Source_CLK= 3,          //Touch clock selection
                                    //Value range: 0:ILRC, 1:IHRC/2, 2:IHRC/4, 3:IHRC/8, 4:IHRC/16, 5:IHRC/32, 6:IHRC/64, 7:IHRC/128 (1:reserved)
                                    //Default: 3
    C_Touch_VRef            =   0,  //The selection of touch reference voltage has an impact on the performance of touch sensitivity and the selection of CS capacitance.
                                    //The higher the theoretical reference voltage is, the higher the sensitivity is
                                    //Value range: 0:0.8*Touch Power(TP), 1:0.7*TP, 2:0.6*TP, 3:0.5*TP, 4:0.4*TP, 5:0.3*TP, 6:0.2*TP
                                    //Default: 0
    C_Touch_Discharge   =   2,  //CS discharge time selection before touch, the longer the time, the cleaner the discharge
                                //0:reserved, 1:CLK_32, 2:CLK_64, 3:CLK_128
                                //Default: 3
    C_Smooth_Rank   =   2,      //Filter rating, Number of filter = 2^Filter rating + 2
    C_Shake_Rang    =   5,      //Jitter allowed range, beyond the range and below the sensitivity, the environment value will start to correct
    //------------------------
}   ;
```

Fig.5-9：General parameter setting

5. Upper computer setting

If the upper computer is enabled on the software, the following upper computer parameters are configured according to the settings. The System_Clock is consistent with the system clock; En _ Uart is enabled by the upper computer; Since the PFC460 is a multi-core chip, FPPA means setting the Uart location; Baud _ Rate is the Uart baud rate. If modification is required, pay attention to the baud rate on the synchronous upper computer software；UART_Out sets the communication port;

```
Debug_Set:  //Debug/debug Settings
ENUM
{
    //-----------------------
    //C_Debug_printf    =   1,  //Debug printf Touch the slider/slip ring value
    //-----------------------
    System_Clock    =   4000000,    //  SYSCLK,Used at UART
    En_Uart     =   1,
    FPPA            =   0,                      //  FPPA where Uart is located
    Baud_Rate   =   19200,              //  Uart baud rate
    //-----------------------
}   ;
UART_Out        BIT PA.0
```

Fig.5-10：Upper computer setting

5.2.2. User function files

User function file User_Function. C is mainly used for IO and user variable initialization, the realization of the user-defined function module.

1. IO initialization：

```
Void    IO_Init(void)
{
//  PA  =   0x00;
//  PAC =   0x00;
//  PAPH=   0x00;
//  PAPL=   0x00;
//  PADIER= 0x00;
}
```

Fig.5-11：IO initialization settings

2. Custom slider, slip ring, key functions:

In the function T _ Key _ Function (void), the trigger signals of the slider, the slip ring and the key are continuously scanned, and the user can customize his own function in the judgment of each trigger and release condition;

Customize key function settings

```
if(T_Key_Signal)    //T_Key_Signal.1~T_Key_Signal.15    Used to store key signal
{
    if(T_Key_Signal.1)  //Key1 trigger
    {
    }
    else    //release
    {
    }
    if(T_Key_Signal.2)  //Key2 trigger
    {
    }
    else    //release
    {
    }
    if(T_Key_Signal.3)  //Key3 trigger
    {
    }
    else    //release
    {
    }
}
else            //All key release
{
}
```

Fig.5-12：Customize key function settings

Custom slider function: The function is written in the corresponding Case;

```
if(Draw_Slip_Press)         //The slider triggers the flag
{
    switch(Draw_Slip_Shift)
    {
    Case 1: NULL;
        break;
    Case 2: NULL;
        break;
    Case 3: NULL;
        break;
    Case 4: NULL;
        break;
    }
}
else    //Release the slider
{
    NULL;
}
```

Fig.5-13：Custom slider function

Custom slip ring function: The function is written in the corresponding Case;

```
if(Slip_Ring_Press)          //Slip ring trigger mark
{
    switch(Slip_Ring_Shift)
    {
    Case 1: NULL;
        break;
    Case 2: NULL;
        break;
    Case 3: NULL;
        break;
    Case 4: NULL;
        break;
    Default: NULL;
        break;
    }
}
else    //SlipRing release
{
    NULL;
}
```

Fig.5-14：Custom slip ring function

5.2.3. User main project file

The main project file mainly contains some initialization functions and the scanning function of slip ring, slider and key;

```
IO_Init();
Uart_Auto();
TK_Init_Auto();
Timer16_Init();


//  CLKMD.En_WatchDog   =   1;      // WatchDog Enable
while (1)
{
    T_Key_Scan();
    Uart_Auto();
    T_Key_Function();
    //      ...
    wdreset;
}
```

Fig.5-15：Main project file

5.2.4. Use of the upper computer

The upper computer in slider and slip ring scheme is used to display the trigger status of each module intuitively;

1. First, when p-touch_V1.8 is used to generate the program frame, the upper computer is selected to be enabled, and the appropriate communication port is set, and the low voltage mode is adopted by default to wake up.

Fig.5-16：Settings of Uart Enable

2. Select **Tools** in the P-Touch _ V1.8 menu, and click to open the upper computer.As shown in the figure below, Uart is used to connect the slider and slip ring simulation board. M1 and M2 are two trigger display effects.

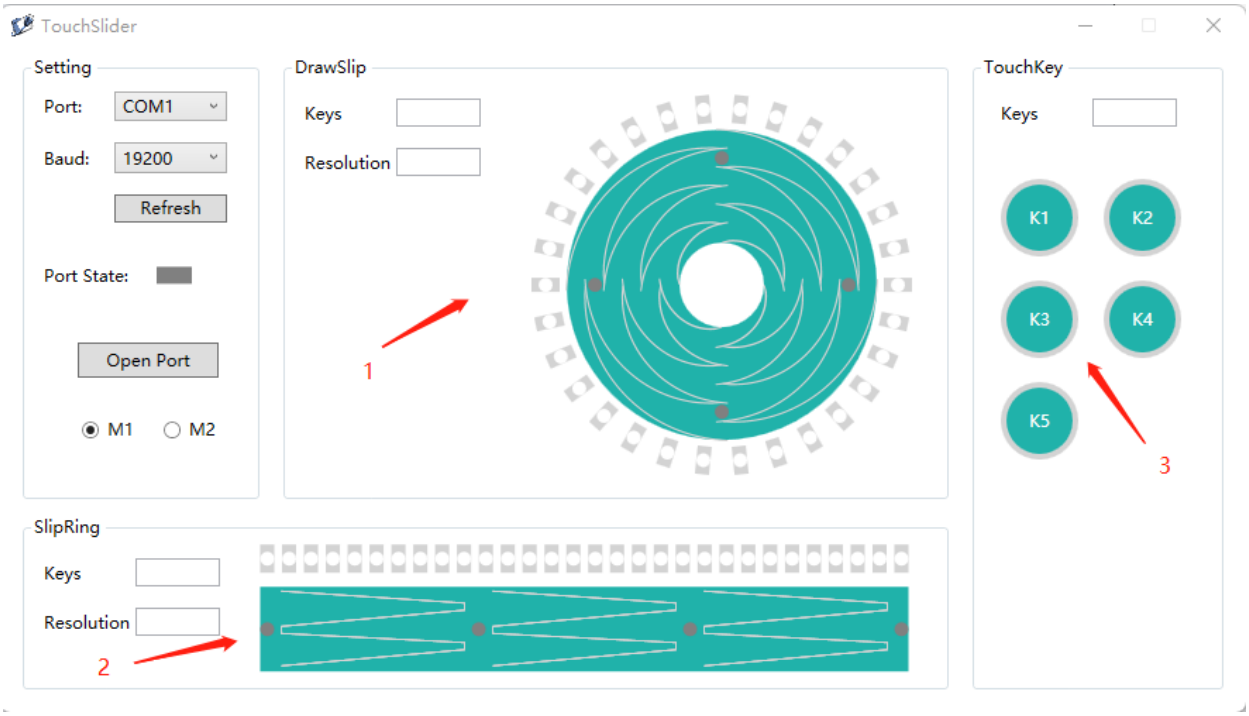


Fig.5-17：Upper computer of slider and slip ring

Mark 1 is the display area of slip ring trigger;

Mark 2 is the display area of slider trigger;

Mark 2 is the display area of key ring trigger;

3. Trigger display effect:

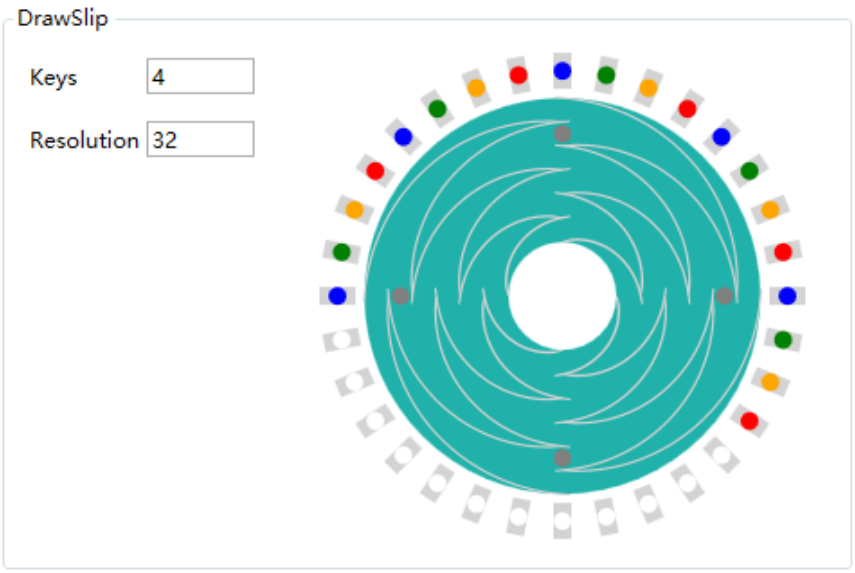


Fig.5-18：Slip ring trigger display

Fig.5-19：Slider trigger display

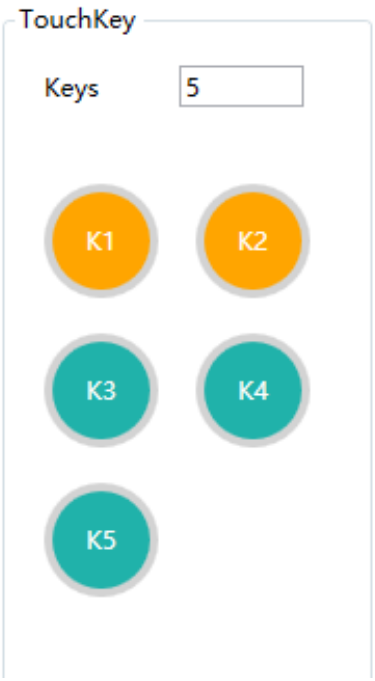


Fig.5-20：Key trigger display

6. Standard Product Selection Manual

The following is the standard product selection manual. Users can use it directly according to its function description if it meets their needs. In addition, they can download the corresponding PDK burning file by themselves.

Attention：Before using standard product program, please verify the function in detail. PADAUK Technology does not assume any software responsibility.

Touch Series

Standard Product Selection Manual

Please verify the function of the standard program before using, the company does not bear any software responsibility! 2020/12/04

Product	Keys	Output	Package	Describe	CheckSum	PDK
XDT8001C-S08B	1	PWM	SOP8	Single key monochromatic light	0x1D2474	
XDT8002B-S08B	2	CMOS	SOP8	Double key switch type - Flip mode	0x3124F5	
XDT8101B-U06	1	CMOS	SOT23-6	Single-key switch Type - Indicates the direct mode	0x4539A4	
XDT8101B-2N06	1	CMOS	DFN-6	Single-key switch Type - Indicates the direct mode	0x4539A4	
XDT8102B-U06	1	CMOS	SOT23-6	Single key switch type - Flip mode	0xCC0B5C	
XDT8103-S08B	2	CMOS	SOP8	Double key Switch Type - Direct mode	0xE43510	
XDT8104-S08B	2	CMOS	SOP8	Double key switch type - Flip mode	0x1404A0	
XDT8105-EY10	3	CMOS	ESSOP10	Double key Switch Type - Direct mode	0xA50624	
XDT8106-EY10	3	CMOS	ESSOP10	Double key switch type - Flip mode	0xAC3B6A	
XDT8107-EY10	4	CMOS	ESSOP10	BCD output	0xFD1245	
XDT8108-EY10	1	PWM	ESSOP10	Single key two-tone light	0x653E03	