

PT8P2104 触摸库使用手册

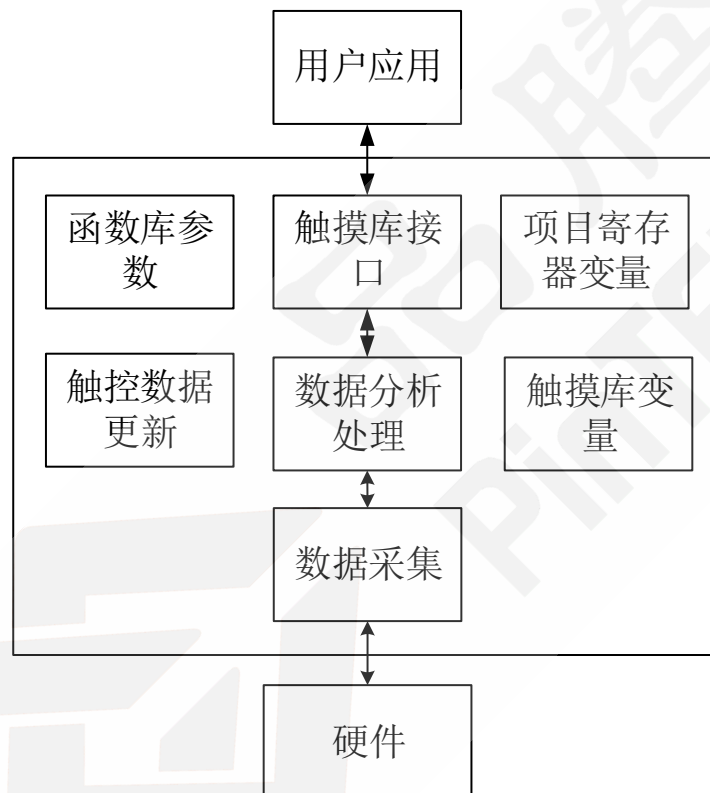
1. 概述.....	3
2. 触摸库架构.....	3
3. 触摸库资源占用情况.....	3
3.1 RAM.....	3
3.2 ROM.....	4
3.3 堆栈.....	4
3.4 IO 端口.....	4
4. 触摸库接口.....	4
4.1 汇编库接口.....	4
4.1.1 函数名.....	4
4.1.2 输入数据.....	4
4.1.3 输出数据.....	5
4.1.4 文件组成.....	5
4.2 C 库接口.....	5
4.2.1 函数名.....	5
4.2.2 输入数据.....	5
4.2.3 输出数据.....	6
4.2.4 文件组成.....	6
5. 参数定义.....	6
5.1 参数列表.....	6
5.2 参数配置.....	7
6. 注意事项.....	8
6.1 中断处理函数.....	8
6.2 低功耗模式.....	8
6.3 按键超时处理.....	8
6.4 二次触摸功能.....	9
6.5 WDT 处理.....	9
7. 应用示例.....	9
8. 历史记录.....	10

1. 概述

PT8P2104 是一款集成了电容式触摸感应模块的 8 位 OPT 单片机。湖南品腾提供标准 PT8P2104 触控算法函数库(TOUCH)，方便客户快速进行程序开发。本文主要介绍触控函数(TOUCH)的资源利用情况、函数使用方法及函数使用注意事项。

2. 触摸库架构

函数组成结构如下：



3. 触摸库资源占用情况

介绍触摸库运行过程中 RAM、ROM、堆栈、Touch 中断及 IO 资源占用情况。

3.1 RAM

RAM 主要用于库函数数据存储。

1, 全局数据:

存储触控算法数据，用户程序在任何时间都不能改变其存储的数据，否则会导致触摸功能异常。全局变量占用空间与实际使用的触摸通道相关。

汇编工程内存使用情况请参考下表，需要注意 RAM 地址分配：

使能通道	标准按键	二次触摸	用户可用 RAM
1	0x5C~0x77	0x58~0x77	0x30~0x5B/0x57
2	0x55~0x77	0x4D~0x77	0x30~0x54/0x4C
3	0x4E~0x77	0x42~0x77	0x30~0x4D/0x41
4	0x47~0x77	不支持	0x30~0x46

表 1 内存使用情况

C 工程触摸库 RAM 由编译系统自动分配，用户不需要关注 RAM 具体地址分配问题。

3.2 ROM

ROM 资源使用如下表：

编号	功能	ROM 占用
1	按键(开启低功耗)	单键约 400*14bits 多键约 500*14bits
2	二次触摸功能	80*14bits
3	低功耗功能	默认开启低功耗，关闭低功耗 ROM 减少 70~100*14bits

表 1 ROM 使用情况

3.3 堆栈

TOUCH 函数内部包含一级函数调用，占用一级堆栈。

3.4 IO 端口

根据客户需求占用不同的引脚，最多可选择 4 个 IO 作为触控通道。

4. 触摸库接口

本节介绍函数对外接口，包括函数名、输入数据、输出数据及文件组成。

4.1 汇编库接口

4.1.1 函数名

函数名：TOUCH

用户程序直接调用 TOUCH 函数，触摸库函数会采集数据进行算法处理并输出触摸按键值及相应的状态。

调用方法：

例：CALL TOUCH

4.1.2 输入数据

LP_EN: Bit 类型数据, 仅支持位操作, 1 表示开启低功耗扫描模式。参考低功耗模式

TH_KEY_EXPIRE: Byte 类型数据, 低四位数据有效, 用户检测到某一通道的按键超时, 配置该寄存器相应位。参见按键超时处理

FIRST: Bit 类型数据, 仅支持位操作, 此位清零强制复位触摸库程序。

4.1.3 输出数据

LP_AL: Bit 类型数据, 仅支持位操作, 1 表示触控函数允许程序进入低功耗扫描模式。参考低功耗模式

KEY_STATE: Byte 类型数据, 低四位数据有效. 参考参数定义 OPTION_TOUCH_CHANNEL_SEL 描述部分。

0001B: 使能的第 1 通道有按键发生

0010B: 使能的第 2 通道有按键发生

0100B: 使能的第 3 通道有按键发生

1000B: 使能的第 4 通道有按键发生

4.1.4 文件组成

Touch 函数库由三个文件组成: TOUCH_SET.inc, TOUCH.inc 和 xxx.lib

TOUCH_SET.inc: 用户需要修改的触控函数定义

TOUCH.inc: 定义触控函数 RAM 及参数定义(请勿修改!!!)

xxx.lib: 触控函数库

注意: TOUCH_SET.inc 需在 TOUCH.inc 之前被包含

4.2 C 库接口

4.2.1 函数名

函数名: void touch(void);

函数在 touch.h 文件中进行申明。

客户直接调用该函数, 触摸库函数会采集数据并进行算法处理并输出触摸按键值及相应的状态。

调用方法:

例: touch();

4.2.2 输入数据

SET_LP_EN: 宏, 调用此宏定义表示开启低功耗扫描模式。参考低功耗模式

TOUCH_RESET: 宏, 调用此宏定义将复位触摸模块

key_expire: unsigned char 类型数据, 低四位数据有效, 用户检测到某一通道的按键超时, 配置该寄存器相应位。参见按键超时处理

4.2.3 输出数据

key_flag: unsigned char 类型数据, 低四位数据有效. 考参数定义

[OPTION_TOUCH_CHANNEL_SEL](#) 描述部分。

LP_AL_STA: 宏, 获取触摸模块是否允许进入低功耗扫描模式

4.2.4 文件组成

Touch 函数库由四个文件组成: touch.h, TOUCH_SET.inc, TOUCH.inc 和 xxx.alib

TOUCH_SET.inc: 用户需要修改的触控函数定义

TOUCH.inc: 定义触控函数 RAM 及参数定义(请勿修改!!!)

xxx.alib: 触控函数库

注意:

TOUCH_SET.inc, TOUCH.inc 与 xxx.alib 需放在工程目录的 lib 文件夹下

touch.h 需要在用户需要调用触摸库的 C 文件中包含。

5. 参数定义

本章节列举触摸库的可配置参数, 并说明参数的功能及配置时注意事项。

5.1 参数列表

函数参数定义如下表:

参数名	默认值	说明备注
OPTION_TOUCH_CHANNEL_SEL	0x0F	Touch 通道功能参数设置, 每一比特对应一个通道。有效配置为 0x01 ~0x0F, 不允许配置为其它数据。
OPT_ESTATE_TH_AGAIN	0	防缠绕功能
OPT_TOUCH_SENSITIVITY_SEL	6	灵敏度等级有效值 1~10, 值越大灵敏度越高
OPTION_KEYFINGER_THRESHOLD_0	200	使能的第 1 通道触摸阈值, 根据项目实际情况进行调整
OPTION_RELEASE_THRESHOLD_0	150	使能的第 1 通道释放阈值, 根据项目实际情况进行调整
OPTION_NOISE_THRESHOLD_0	40	使能的第 1 通道噪声阈值, 根据项目实际情况进行调整
OPTION_KEYFINGER_THRESHOLD_1	160	使能的第 2 通道触摸阈值, 根据项目实际情况进行调整
OPTION_RELEASE_THRESHOLD_1	120	使能的第 2 通道释放阈值, 根据项目实际情况进行调整
OPTION_NOISE_THRESHOLD_1	32	使能的第 2 通道噪声阈值, 根据项目实际情况进行调整
OPTION_KEYFINGER_THRESHOLD_2	160	使能的第 3 通道触摸阈值, 根据项目实际情况进行调整
OPTION_RELEASE_THRESHOLD_2	120	使能的第 3 通道释放阈值, 根据项目实际情况进行调整
OPTION_NOISE_THRESHOLD_2	32	使能的第 3 通道噪声阈值, 根据项目实际情况进行调整
OPTION_KEYFINGER_THRESHOLD_3	150	使能的第 4 通道触摸阈值, 根据项目实际情况进行调整
OPTION_RELEASE_THRESHOLD_3	112	使能的第 4 通道释放阈值, 根据项目实际情况进行调整
OPTION_NOISE_THRESHOLD_3	30	使能的第 4 通道噪声阈值, 根据项目实际情况进行调整
OPT_CMOD_CHECK_FUN	0	外接 Cmod 灵敏度调节功能, 有效值 0,1,2

OPT_CMOD_PIN_LOCATION	2	外接 Cmod 脚选择，有效值 1,2
OPT_LP_MODE_ENABLE	1	低功耗功能
OPT_SLP_THRESHOLD0	-	逻辑通道 0 低功耗唤醒阈值
OPT_SLP_THRESHOLD1	-	逻辑通道 1 低功耗唤醒阈值
OPT_SLP_THRESHOLD2	-	逻辑通道 2 低功耗唤醒阈值
OPT_SLP_THRESHOLD3	-	逻辑通道 3 低功耗唤醒阈值
OPTION_KEY_IN_DEBONCE	0x04	按键有效消抖次数
OPTION_KEY_OUT_DEBONCE	0x02	按键释放消抖次数
OPT_ESTA_RELEASE_DB	0x08	防缠绕按键释放消抖次数
OPT_ESTA_BASE_UP_DB	0x04	防缠绕基值更新消抖次数

注意：使能的通道编号与芯片物理 Touch 通道的对应关系，可参考 5.2 参数配置。

5.2 参数配置

用户只需要根据自己的需求设置以下数据：

1, **OPTION_TOUCH_CHANNEL_SEL**: 配置使能的 Touch 通道，有效位为低 4 位。

低 4 位 配置	TH3	TH2	TH1	TH0	KEY_STAT E 有效位	说明
0001				Y	Bit0	Touch0 为使能通道 0
0010			Y		Bit0	Touch1 为使能通道 0
0011			Y	Y	Bit1, 0	Touch0 为使能通道 0, Touch1 为使能通道 1
0100		Y			Bit0	Touch2 为使能通道 0
0101		Y		Y	Bit1, 0	Touch0 为使能通道 0, Touch2 为使能通道 1
0110		Y	Y		Bit 1, 0	Touch1 为使能通道 0, Touch2 为使能通道 1
0111		Y	Y	Y	Bit 2, 1, 0	Touch0 为使能通道 0, Touch1 为使能通道 1, Touch2 为使能通道 2
1000	Y				Bit0	Touch3 为使能通道 0
1001	Y			Y	Bit 1, 0	Touch0 为使能通道 0, Touch3 为使能通道 1
1010	Y		Y		Bit 1, 0	Touch1 为使能通道 0, Touch3 为使能通道 1
1011	Y		Y	Y	Bit 2, 1, 0	Touch0 为使能通道 0, Touch1 为使能通道 1, Touch3 为使能通道 2
1100	Y	Y			Bit 1, 0	Touch2 为使能通道 0, Touch3 为使能通道 1
1101	Y	Y		Y	Bit 2, 1, 0	Touch0 为使能通道 0, Touch2 为使能通道 1, Touch3 为使能通道 2
1110	Y	Y	Y		Bit 2, 1, 0	Touch1 为使能通道 0, Touch2 为使能通道 1, Touch3 为使能通道 2
1111	Y	Y	Y	Y	Bit 3, 2, 1, 0	Touch0 为使能通道 0, Touch1 为使能通道 1, Touch2 为使能通道 2, Touch3 为使能通道 2

2, 触摸通道的阈值，推荐使用以下配置数据：

阈值	设置值	说明
----	-----	----

OPTION_KEYFINGER_THRESHOLD	200	$100 \leq \text{OPTION_KEYFINGER_THRESHOLD} \leq 250$
OPTION_RELEASE_THRESHOLD	160	释放阈值=80%触摸阈值
OPTION_NOISE_THRESHOLD	40	噪声阈值=20%触摸阈值

注意: 触摸阈值(OPTION_KEYFINGER_THRESHOLD*)最大有效配置值为 250;

3, 配置低功耗唤醒阈值: 根据选择的触摸通道和触摸阈值配置不同的低功耗唤醒阈值。

推荐低功耗唤醒阈值:

$\text{OPT_SLP_THRESHOLD} = (\text{OPT_RELEASE_THRESHOLD} + \text{OPT_NOISE_THRESHOLD}) / 2$

例:

$\text{OPTION_RELEASE_THRESHOLD} = 150$

$\text{OPT_NOISE_THRESHOLD} = 30$

$\text{OPT_SLP_THRESHOLD} = (150 + 30) / 2 = 90$

6. 注意事项

6.1 中断处理函数

中断处理函数必需清除 T2IF 标志位。

6.2 低功耗模式

低功耗模式是通过芯片 STOP 模式和看门狗(WDT)唤醒功能交替运行达到降低系统功耗的目的。低功耗模式流程如下:

- 1, 主程序调用 Touch 函数
- 2, 检测 LP_AL=1 且主程序允许进入低功耗, 跳转到第 3 步; 否则不进入低功耗模式, 返回第 1 步;
- 3, 进入 STOP 等待唤醒;
- 4, 唤醒后返回第 1 步。

6.3 按键超时处理

按键超时是为防止系统工作时有触摸通道异常一直处于按键有效无法正常工作的情况。

按键超时处理流程如下:

- 1, 主程序检测到按键值时启动计时, 按键无效时清除计时
- 2, 当计时器超过设定的时间按键仍然有效则设置相应的 TH_KEY_EXPIRE 位
- 3, 下次调用 Touch 函数时, 函数会复位相应通道的状态。

注意:

1, TH_KEY_EXPIRE 值的设置与 OPTION_TOUCH_CHANNEL_SEL 选择的通道相关

例: OPTION_TOUCH_CHANNEL_SEL=0x06, 则 TH_KEY_EXPIRE 有效赋值为 0x02,0x04,0x06。

2, 推荐开启按键超时功能;

3, 如果系统不支持按键超时处理, 请勿对 TH_KEY_EXPIRE 进行操作。

6.4 二次触摸功能

PWM 输出或电源与触摸盘连接线缠绕时需要开启二次触摸选项(`TOUCH_STATE_TOUCH_AGAIN`), 此时支持绕线完成后再次触摸时及时响应触摸按键。

注意: 1.默认二次触摸选项关闭;

2.此项最多同时支持 3 个触摸按键。

二次触摸选项开启示例:

```
#define TOUCH_STATE_TOUCH_AGAIN
```

6.5 WDT 处理

触摸库程序包含清除看门狗(CLRWDT)指令

7. 应用示例

流程: 1, 设置通道数, 配置相关阈值

2, 主循环调用 Call 函数

3, 如果低功耗达到条件, 设置低功耗并睡眠等待唤醒, 再跳转到 4; 未达到直接跳到 4

4, 如果有任意超长按键产生设置超长按键

5, 判断是否需要输出数据, 需要则输出数据然后返回 2, 否则直接返回 2

详细请参考 Demo 程序.

8. 历史记录

版本号	修改记录	发布日期
V1.0	初版	2024-12-03
V1.1	1, 修改部分文字描述 2, 更新章节 4 触摸库接口部分 3, 更新文件目录 4, 更新 5.1 参数列表: 增加 OPT_TOUCH_SENSITIVITY_SE	2025-03-07
V1.2	1, 增加 6.5 WDT 处理	2025-03-21