

PT8M2102 触摸库函数说明

目录

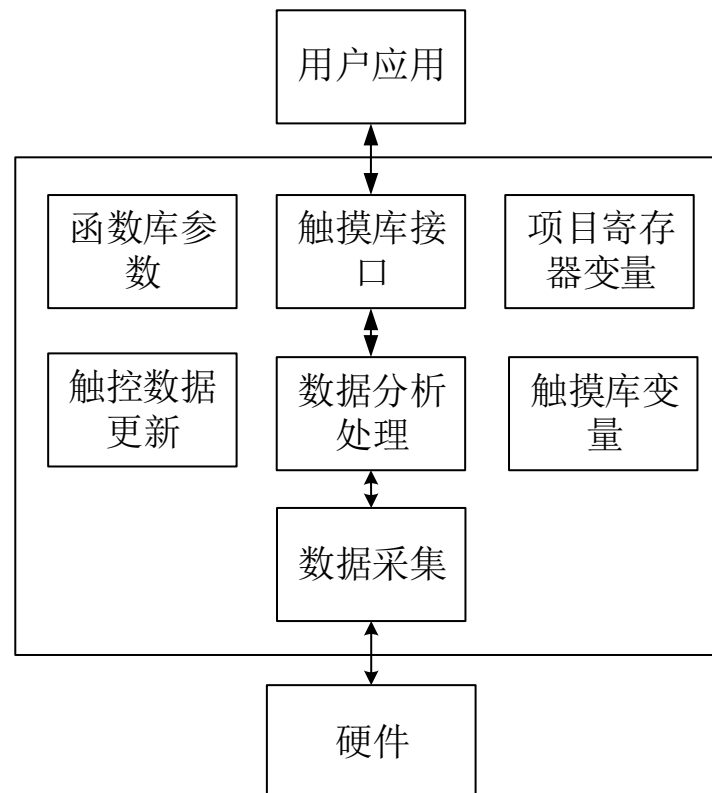
1. 概述.....	3
2. 函数基本架构.....	3
3. 函数资源占用情况.....	3
3.1 RAM	3
3.2 ROM.....	4
3.3 堆栈	4
3.4 IO 端口.....	4
4. 函数接口.....	4
4.1 函数名.....	4
4.2 输入数据	4
4.3 输出数据	4
4.4 文件组成	5
5. 参数定义.....	5
5.1 参数列表	5
5.2 参数配置	6
6. 注意事项.....	7
6.1 低功耗模式.....	7
6.2 按键超时处理	7
6.3 绕线触摸处理	7
6.4 扫描时间间隔	8
7. 应用示例.....	8
8. 历史记录.....	10

1. 概述

PT8M2102 是一款集成了电容式触摸感应模块的 8 位 MTP 单片机。湖南品腾科技提供标准 PT8M2102 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. 函数基本架构

函数组成结构如下：



3. 函数资源占用情况

函数运行过程中占用 RAM、ROM、堆栈、Touch 中断及 IO 资源

3.1 RAM

RAM 主要用于函数运行过程中的临时数据存储，触控算法运行完成后的数据存储。

1, 全局数据：存储地址为 0x72 到 0x95，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，否则会导致触控异常。全局变量具体存储空间与实际使用的 Touch 通道相关，具体 RAM 使用空间请参考下表：

使用通道数	实际使用地址空间	备注
1	0x72~0x89	用户可使用 0x8A~0x95RAM 空间
2	0x72~0x8D	用户可使用 0x8E~0x95RAM 空间

3	0x72~0x91	用户可使用 0x92~0x95 RAM 空间
4	0x72~0x95	
*	OPTION_ONECH_EM_C_EN	使能此功能时，RAM 0x94~0x95 会被函数库占用

2, 局部数据: 存储地址为 0x5C 到 0x71, 触控函数运算过程中的临时变量, 用于算法运算等。用户程序在触控函数未退出时不能改变其存储的数据, 同时每次调用触控函数时其中存储的数据会被改变。
注意: 推荐用户程序不使用此段 RAM, 如要使用此段 RAM 请注意 RAM 作用域(勿在中断内使用)。

3.2 ROM

ROM: 该函数占用约 650 行 ROM 空间。

3.3 堆栈

TOUCH 函数内部有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. 函数接口

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 函数名

函数名: TOUCH

客户直接调用该函数, 进行触摸相应的算法, 输出按键状态, 每次调用根据当前状态进行状态跳转。

调用方法:

例: CALL TOUCH

4.2 输入数据

LP_EN: Bit 类型数据, 只能使用位操作, 1 表示使能低功耗按键扫描模式。参见低功耗模式

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

4.3 输出数据

LP_AL: Bit 类型数据, 只能使用位操作, 1 表示触控函数允许程序进入低功耗模式。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

4.4 文件组成

Touch 函数库由三个文件组成: TOUCH_SET.inc, TOUCH.inc 和 TOUCH_2102.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(切勿修改!!!)

TOUCH_2102.lib: 触控函数库

注意: TOUCH_SET.inc 需在 TOUCH.inc 之前被包含

5. 参数定义

本节描述函数库提供的参数及设置这些参数的方法

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_KEY_IN_DEBONCE	0x04	按键进入消抖次数
OPTION_KEY_OUT_DEBONCE	0x02	按键退出消抖次数
OPTION_KEYFINGER_THRESHOLD_0	100	使能的第 1 通道触摸阈值
OPTION_RELEASE_THRESHOLD_0	75	使能的第 1 通道释放阈值
OPTION_NOISE_THRESHOLD_0	20	使能的第 1 通道噪声阈值
OPTION_KEYFINGER_THRESHOLD_1	100	使能的第 2 通道触摸阈值
OPTION_RELEASE_THRESHOLD_1	75	使能的第 2 通道释放阈值
OPTION_NOISE_THRESHOLD_1	20	使能的第 2 通道噪声阈值
OPTION_KEYFINGER_THRESHOLD_2	100	使能的第 3 通道触摸阈值
OPTION_RELEASE_THRESHOLD_2	75	使能的第 3 通道释放阈值
OPTION_NOISE_THRESHOLD_2	20	使能的第 3 通道噪声阈值
OPTION_KEYFINGER_THRESHOLD_3	100	使能的第 4 通道触摸阈值
OPTION_RELEASE_THRESHOLD_3	75	使能的第 4 通道释放阈值
OPTION_NOISE_THRESHOLD_3	20	使能的第 4 通道噪声阈值
OPTION_SLP_THRESHOLD	5	低功耗唤醒阈值
OPTION_TOUCH_CHANNEL_SEL	0x0F	Touch 通道功能参数设置, 低 4 位有效, 高 4 位无效, 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。
OPTION_TH_STATE_E	75	二级阈值=75%触摸阈值
OPTION_ONECH_EMG_EN		间隔扫描使能
OPTION_TH_SPACE		开启间隔扫描后, 此参数用于配置调整扫描间隔时间

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 中 OPTION_TOUCH_CHANNEL_SEL 配置表格。

5.2 参数配置

用户只需要根据自己的需求设定以下数据：

1, **OPTION_TOUCH_CHANNEL_SEL**：配置哪些通道为使能的 Touch 通道，有效位为低 4 位。

低 4 位 配置数 据	Touch3	Touch2	Touch1	Touch0	KEY_STATE 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, 配置对应触控通道的阈值，推荐使用以下配置数据：

阈值	设置值	说明
OPTION_KEYFINGER_THRESHOLD	100	$80 \leq \text{OPTION_KEYFINGER_THRESHOLD} \leq 250$
OPTION_RELEASE_THRESHOLD	75	释放阈值 = 75% 触摸阈值
OPTION_NOISE_THRESHOLD	20	噪声阈值 = 20% 触摸阈值

OPTION_TH_STATE_E	75	二级阈值=75%触摸阈值
-------------------	----	--------------

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

3, 配置低功耗触控通道的阈值, 根据选择的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(OPTION_SLP_THRESHOLD)配置关系:

使能通道数	触摸阈值*	推荐低功耗唤醒阈值	说明
1	100	50	唤醒阈值=50% 触摸阈值
2	100	18	唤醒阈值=18% 触摸阈值
3	100	8	唤醒阈值= 8% 触摸阈值
4	100	5	唤醒阈值= 5% 触摸阈值

触摸阈值*: 多通道时取最大的触摸阈值

6. 注意事项

6.1 低功耗模式

低功耗模式是通过使能芯片 STOP 模式和 WDT 唤醒功能交替进行达到降低整个系统功耗的目的。低功耗模式流程如下:

- 1, 主程序调用 Touch 函数
- 2, Touch 函数检测所有通道都为稳定无按键状态, 则 Touch 函数设置 LP_AL=1, 跳转到第 3 步;
- 3, 主程序允许进入低功耗且主程序检测到 LP_AL=1, 跳转到第 4 步, 否则不进入低功耗模式, 返回第 1 步;
- 4, 主程序设置 LP_EN=1, 将 WDT 设置为溢出唤醒模式, 进入 STOP, 等待唤醒
- 5, 唤醒后返回第 1 步.

6.2 按键超时处理

超时按键是为防止系统上电后有触控通道一直处于按键状态无法正常工作的情况。

按键超时处理流程如下:

- 1, 主程序检测到按键信息时启动计时, 如果按键信息撤除后清除计时
- 2, 当时间超过设定的按键超时时间时按键信息仍然没有撤除则设置相应的 TH_KEY_EXPIRE 位
- 3, 下次调用 Touch 函数时, 函数会重新获取所有应通道的状态信息并获取基值。

注意:

- 1, TH_KEY_EXPIRE 值的设置与 OPTION_TOUCH_CHANNEL_SEL 选择的通道相关
e.g. OPTION_TOUCH_CHANNEL_SEL=0x06, 则 TH_KEY_EXPIRE 只能赋值为 0x02,0x04,0x06。
- 2, 推荐开启按键超时功能;
- 3, 用户可不支持按键超时处理, 此时请勿对 TH_KEY_EXPIRE 进行操作。

6.3 绕线触摸处理

当 PWM 输出线与电源线缠绕时需要开启二级触摸选项(#define STATE_E_TOUCH_EN), 此时支持绕

线完成后现次触摸时及时响应触摸按键。

注意：1,默认二级触摸选项关闭；

2.此项只支持单按键，多按键时自动忽略此选项。

二级触摸选项开启示例：

```
#define STATE_E_TOUCH_EN
#define OPTION_TH_STATE_E      75 ;(=释放阈值)
```

6.4 扫描时间间隔

OPTION_ONECH EMC_EN: 间隔扫描时间使能或关闭功能。库不会在每次调用时进行 Touch 的转换，只有在库内部计数次数达到后才会进行一轮 Touch 的转换，转换完成后内部的计数会被清零，重新开始计数。

OPTION_TH_SPACE: 间隔时间调整参数，可通过增大或减小参数达到增加或减少间隔时间。

注意：1, 当开启的通道数为 4 时此功能会被自动屏蔽

2, 开启此功能后会占用 RAM(0x94、0x95)，即 RAM(0x94、0x95)不能再用作其它用途。

7.应用示例

流程：1, 设置通道数，配置相关阈值

2, 调用 Call 函数

3, 如果低功耗达到条件，设置低功耗并睡眠，等待唤醒回到 2

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

```
*****
;
; 程序启动
    ORG    0x000          ; processor reset vector
    CLR    STATUS
    CLRWDT
    BSET   WDTE
    JMP    MAIN          ; go to beginning of program
    ORG    0x008
    JMP    INT_FUN
    ORG    0x00A
*****
;
; 中断函数
INT_FUN:
    ;添加中断处理程序
    RETI
*****
;主函数入口
```


MAIN:

;添加初始化程序

MAIN_LOOP:

CALL TOUCH

LP_MODE:

;可添加主程序低功耗条件判断

BCLR LP_EN

BTSNZ LP_AL

JMP MAIN_CLR_WDT

BSET LP_EN

;进入 STOP 模式

BCLR LDOEN

MOV A , #0xC0

MOV PCON , A

STOP ; stop command clear WDT and stop system clock

NOP

MOV A , #0xA4

MOV PCON , A

BSET LDOEN

MAIN_DOUT_PUT:

;添加主函数数据处理

MAIN_CLR_WDT:

CLRWDT

JMP MAIN_LOOP

8. 历史记录

版本号	修改记录	发布日期
V1.0	初版	2020-06-24