



武汉芯源半导体有限公司
WUHAN XINYUAN SEMICONDUCTOR CO., LTD

CW32 电容式触摸按键设计指南

应用笔记

版本号：Rev 1.0



前言

CW32 电容式触摸按键设计指南向客户提供一种利用 CW32 内部资源结合软件编程实现电容式触摸按键有效触摸检测的方法。本指南的内容重点在于工作原理、软件检测过程以及调试指引。

利用芯源半导体的 CW32 系列小规模 MCU 的 IO、比较器、定时器、高速高精度内置 RC 时钟源以及高算力等功能，通过检测电路端子电容的微小变化和波动，实现电容式触摸按键功能。其外围电路简单，占用资源比例不高，非常有利于用户在节约 BOM 成本的前提下拓展功能。结合适当的工业化设计，触摸按键比接触式按键更美观、耐磨的同时，还具有防水、抗干扰、寿命长等多种优势。

通过本文，您会了解到如何利用内置电压比较器和内置定时器及软件配合，实现灵活方便的按键检测。本文在介绍标准演示板（如下图）和演示软件的性能参数的同时，还会给出详细的调试建议以及设计参数选择倾向分析，用以帮助客户快速而自信地完成设计并实施调试。



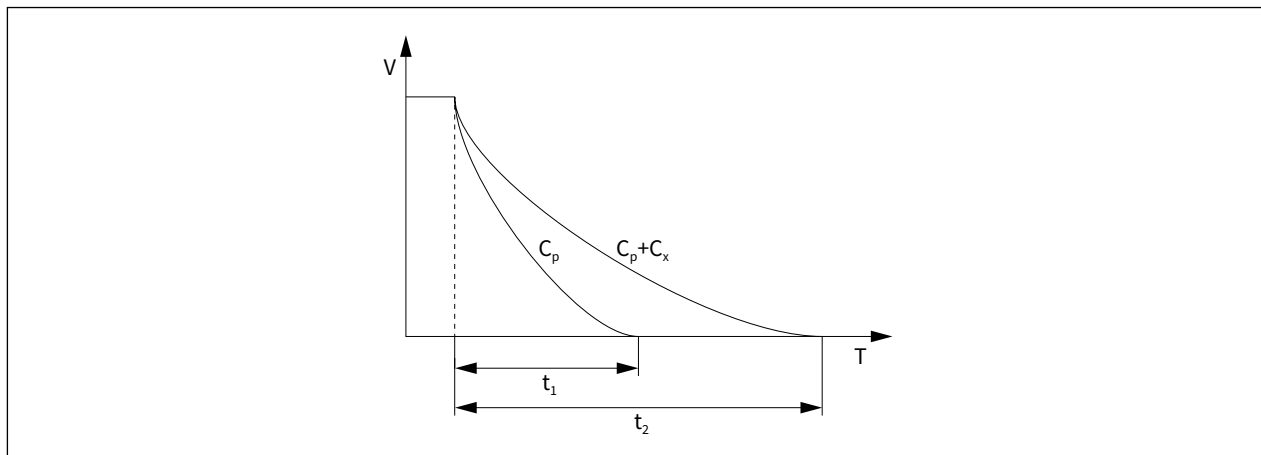
目录

前言	1
1 电容触摸检测基本原理.....	3
2 基于 CW32F003 的触摸按键方案简介.....	4
3 电容触摸检测电路软件过程.....	6
4 触摸参数及选型倾向	7
5 调试指引及性能参考	8
5.1 示例软件框架介绍	8
5.2 调试工具 TD_GetBaseResponseRCT 的使用.....	8
5.3 性能参考	8
6 版本信息	9



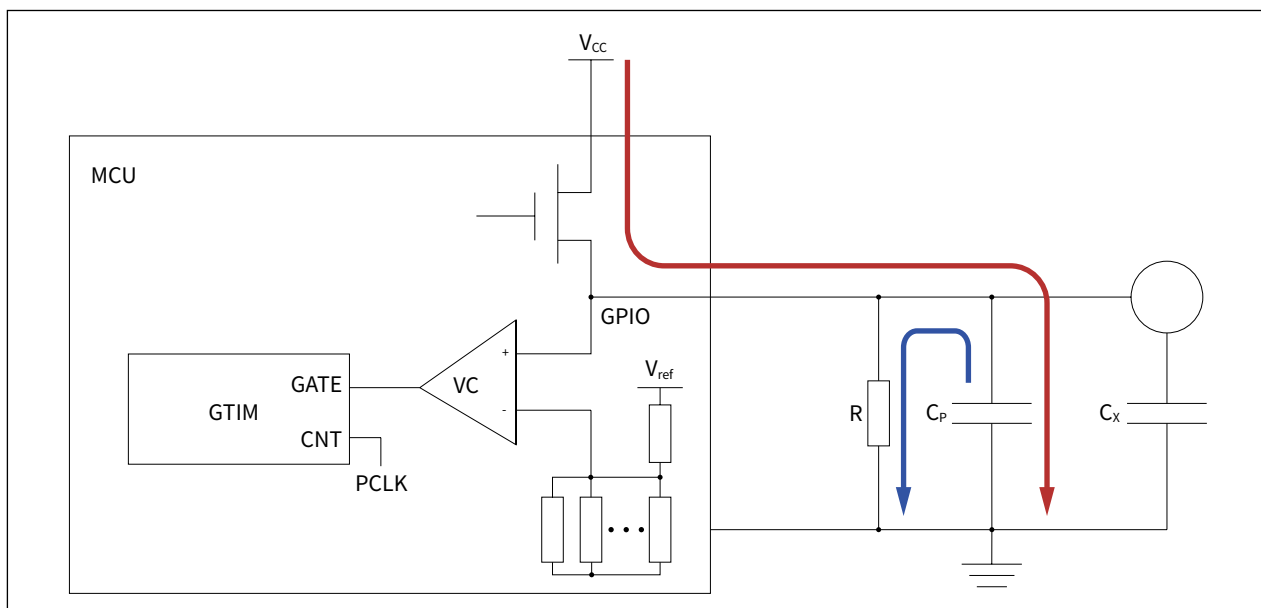
1 电容触摸检测基本原理

独立于电路的金属部件都能够作为电容触摸传感器使用，其原理在于金属部件附近存在手指时，相当于增加了金属部件对地的旁路电容。因此，利用 CW32 系列 MCU 的 IO 口对金属部件充电，并检测电容放电时间的变化，理论上能够辨别金属部件附近是否存在手指按压动作。当无手指存在时，金属部件的电容量为 C_p ，其放电时间为 t_1 ；当存在手指时，增加的旁路电容为 C_x ，此时的放电时间为 t_2 ，如下图所示，可以看出两者之间的放电时间是不一样的：



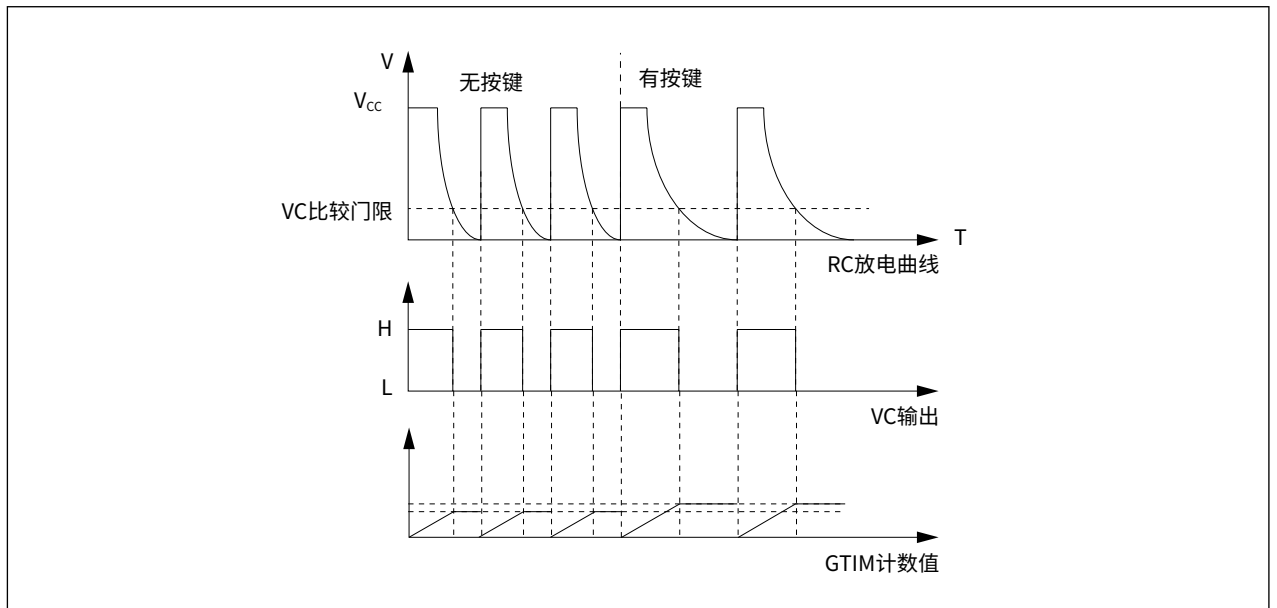
2 基于 CW32F003 的触摸按键方案简介

由于 CW32F003 集成了电压比较器 VC 和定时器，因此触摸按键方案可以通过软件来实现，其实现的原来框图如下所示：



其过程如下：

1. GTIM 配置为门控计数方式，计数源为芯片内部的 PCLK 时钟。
2. VC 比较器的同相端配置为按键的接口，反相端配置为参考，参考的来源为芯片的 V_{CC} 通过内部电阻网络分压得到，VC 比较器输出极性不反转。
3. GPIO 口配置为数字输出，输出高电平对电容充电。由于电容容值比较小，充电电流较大（图中红色箭头所示），电容上的电压很快达到 V_{CC} 。
4. GTIM 的计数器 CNT 清 0，GPIO 口配置为输入高阻态，电容上的电荷基本通过 R 泄放（图中蓝色箭头所示），需要一定的时间，此时电容上的电压要比 VC 比较器的反相端的电压高，VC 输出高电平，是 GTIM 的门控信号有效，GTIM 进行计数。
5. 当电容上的电压降低到比 VC 比较器的反相端的参考电压低时，VC 输出低电平，GTIM 停止计数，同时 VC 比较将产生一个中断信号，此时读取 GTIM 的 CNT 的计数值，和判决门限比较可以判断是否发生触摸按键的事件。如下图所示：



3 电容触摸检测电路软件过程

在范例程序中，软件定时（用定时器中断实现）对每个被测 IO 充电并检测放电时间 N 次，N 次循环检测后，将统计结果提交滤波器状态机，得到按键当前状态。每次检测的具体过程如下：

1. 将 IO 口置高 2 个机器周期，此时金属部件及电容 C 对 GND 的电压被充高到 V_{CC} 。
2. 将 IO 口配置为电压比较器输入模式，此时 IO 口状态切换为高阻输入状态，金属部件及电容 C 通过对 GND 的旁路电阻 R 放电，端子电压变化曲线为标准的 RC 放电曲线。
3. 软件记录循环定时器（GTIM 最高主频运行）的当前值，并等待电压比较器的输出翻转（电压比较器被配置为与某电压门限比较）。
4. 电压比较器输出翻转后立即记录循环定时器当前值，并结合前次记录的时间记录输出结果。

触摸检测过程的相关代码如下：

```
uint32_t TouchKey_GetValue(uint8_t key, uint8_t ref)
{
    uint32_t CurTime;
    //VC1 切换通道
    CW_VC1->CR0_f.INP = key;           // 设置按键通道
    CW_VC1->DIV_f.DIV = ref;           // 设置按键比较的参考比例
    // 获取放电时间
    CW_GPIOB->DIR &= ~(1UL<<8)>>key); // 按键端口输出，对电容充电
    __NOP();
    __NOP();
    CW_GTIM->CNT = 0x0000;             // 计数器清零
    CW_GPIOB->DIR |= ((1UL<<8)>>key); // 按键端口输入高阻
    while((CW_VC1->SR_f.FLTV) == 1);  // 等到放电到比较点
    CurTime = CW_GTIM->CNT;           // 获取放电时间
    return CurTime;
}
```



4 触摸参数及选型倾向

为了保证检测流程顺利执行，需要选择每一个触摸按键的基础电容 C 和放电电阻 R 以及比较器参考门限 V 。DEMO 中，这三个参数一般为 $C=4.7\text{pF}$ ， $R=51\text{K}\Omega$ ， $V=9/64\text{VDD}$ 。

C 和 R 的值，以及比较器参考门限 V 均可根据实际电路测试结果进行调整，调整考量如下：

1. C 的容量增加会令放电时间更长，在检测程序中将会需要更多的机器周期等待比较器翻转。
2. C 的容量增加会显著增强电路稳定性但对检测灵敏度没有大的影响。
3. R 的阻值增加会令放电时间更长，在检测程序中将会需要更多的机器周期等待比较器翻转。
4. R 的阻值增加会降低电路稳定性（高阻易受环境干扰）但对检测灵敏度有明显帮助。
5. 比较器参考门限 V 过高会降低检测灵敏度，但能节约检测时间。门限 V 过低会削弱抗干扰能力并浪费检测时间。



5 调试指引及性能参考

5.1 示例软件框架介绍

示例软件占用 1 个基本定时器，利用定时中断并在中断服务程序中执行按键检测过程、定时周期 10 毫秒。

每次进入中断服务程序后，顺序扫描 M 个触摸按键的 RC 响应。

顺序扫描 N 次后，将 RC 响应结果数据提交滤波器状态机。

滤波器状态机输出按键状态结果。

5.2 调试工具 TD_GetBaseResponseRCT 的使用

示例软件提供一组标定工具来测量当前环境的 RC 响应，执行过程如下：

1. 在没有手指按下的情况下，执行 TD_GetBaseResponseRCT，函数的参数用于选择对应 IO，返回值作为该按键的基础时长 T_B 。
2. 在有手指按下的情况下，执行 TD_GetBaseResponseRCT，函数的参数用于选择对应 IO，返回值作为该按键的信号时长 T_S 。

注 1：每一个按键（IO）的 T_B 和 T_S 都应被单独收集并作为滤波器状态机的参数使用。

注 2：各种温湿度条件下的 T_B 和 T_S 都应该在实验室中被采集并用于影响滤波器状态机的参数。

注 3：比较器门限 V 也是可以针对每一个触摸按键单独选择的，如果某个按键的 T_B 和 T_S 无法实现明显的差异，调节 C 、 R 和 V 将是唯一有效的途径。

另：由于本例利用了高阻态及小信号检测技术，触摸按键的布线要求尽量保持独立性，其金属部件、与 IO 的连线以及 RC 电路周围要尽量避免与其它电路并列共存，否则将大幅提高参数选择及调试难度直至无法完成。

5.3 性能参考

参数	CW32F003F4P7	CW32F003E4P7
最大按键数量	14	14
代码量（不含 TD_GetBaseResponseRCT）	2524 字节	
RAM 开销	624 字节	
示例最大 CPU loading	4.48%	
最大资源占用	GTIM, VC1, VC2	



6 版本信息

表 6-1 文档修订信息

日期	版本	变更信息
2023-06-16	Rev 1.0	初始发布

