

Dec 2009

ST08B

8通道带自校正功能的容性触摸感应器(改进版)

概述

ST08B 触摸感应器可以用平均电容值作为基准检测感应点的电容变化。它可以通过任何非导电介质来感应电容变化。这样感应模块就可以很好的跟水和灰尘隔离。ST08B 和 ST08 相比有更强的抗干扰性和更好的一致性。这个芯片可以工作在低功耗的环境下，当电源为 5v 时，工作电流为 220ua，待机电流为 10ua 以下，也适用于电池应用。

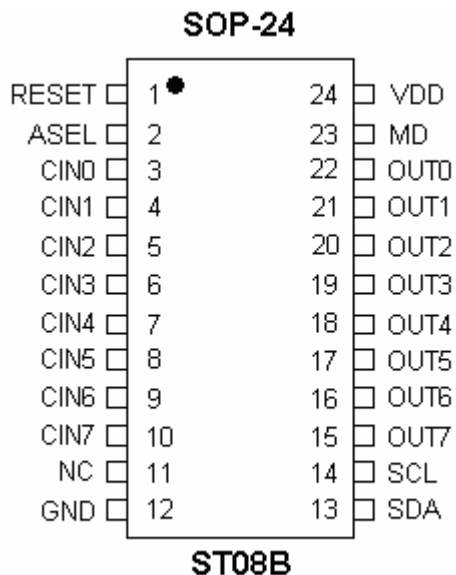
特点

- 带自校正功能的8通道感应芯片
- 可以通过任何非导电介质感应“按键触摸”
- 通过外部电容调整灵敏度
- Open-Drain 的输出形式
- I2C接口
- 工作电压范围：2.1v~6.5v

应用

- 液体高度检测
- 替代开关
- 人体感应检测
- 玩具和互动游戏的人体界面
- 灯开关
- 替代隔膜开关
- 密封的键盘面板

封装



I2C 接口定义

总线的构成及信号类型

I2C总线是由数据线SDA和串行时钟线SCL构成的串行总线，可发送和接收数据。在信息的传输过程中，ST系列触摸芯片是被控器（slaver），主控MCU是主控器（master）。

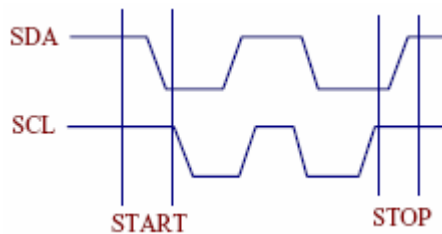
SCL是串行时钟线，在通信过程中始终由主控MCU控制；

SDA是数据线，接上拉电阻，主控器（master）和被控器（slaver）都可以控制SDA线，它们芯片内部是open-drain结构，发送“0”时通过芯片内部的NMOS将SDA线拉低，发送“1”时释放总线由上拉电阻将SDA线拉到高电平。在SCL的低电平时改变状态来发送地址、数据和ACK信号，在SCL的高电平时改变状态来发送开始和结束信号。

开始信号（start）：SCL 为高电平时，SDA 由高电平向低电平跳变，开始传送数据。

结束信号（end）：SCL 为高电平时，SDA 由低电平向高电平跳变，结束传送数据。

应答信号（ACK）：接收数据的 IC 在接收到 8bit 数据后，向发送数据的 IC 发出特定的低电平脉冲，表示已收到数据。主控器向被控器发出一个信号后，等待被控器发出一个应答信号，主控器接收到应答信号后，根据实际情况作出是否继续传递信号的判断。



总线基本操作

ST 系列触摸芯片采用简化的 I2C 总线协议，只有读操作。

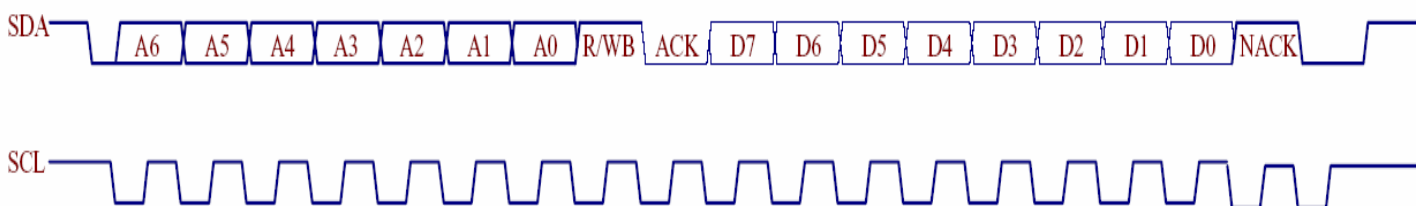
读操作的过程如下：

- 1) 先由主控 MCU 发出 start 信号，主控 MCU 接着送出 8bit 读命令，包括 7bit 地址 A[6:0]和读写控制位 R/WB（高电平表示读命令）。

ST 芯片只有 3 个地址可供选择：当 ASEL 端口为高电平时，地址为 62h，即下图中 A[6:0]=62h；当 ASEL 端口电平为 VDD/2 时，地址为 64h；当 ASEL 端口为低电平时，地址为 61h。

所以当 ASEL 端口为高电平时，主控 MCU 送出的 8bit 读命令应为 C5h；当 ASEL 端口电压为 VDD/2 时，主控 MCU 送出的 8bit 读命令应为 C9h；当 ASEL 端口为低电平时，主控 MCU 送出的 8bit 读命令应为 C3h

- 2) 如果 ST 芯片正确收到读命令，就会发出低电平的 ACK 信号。如果主控 MCU 没有收到 ACK 信号，说明通信出现故障。
- 3) 接下来 ST 芯片会送出 8-bit 按键信息 D[7:0]，低电平表示有按键，高电平表示无按键。
- 4) 最后主控 MCU 发出 NACK 信号，表示通信结束。
- 5) SDA 信号规范：只有在 SCL 下降沿的时候，SDA 的数据才可以发生变化。
- 6) SCL 信号规范：SCL 信号 H 或 L 持续时间至少 5US,SCL 的频率最快不超过 100K。



数据传送波特率

ST08B 芯片的允许最大通信速率

| | |
|--------|----------|
| 芯片内部时钟 | 建议最大的波特率 |
| 3.3us | 100K |

管脚说明

| 编号 | 名称 | I/O | 功能 |
|----|-------|-----|--|
| 1 | RESET | I | 复位（低电平有效） |
| 2 | ASEL | I | I2C 接口地址选择（端口电压为VDD/2时，地址设置为64H 接 VDD，地址设置为62H， 接GND，地址设置为61H） |
| 3 | CIN0 | I | 通道0的感应电容输入 |
| 4 | CIN1 | I | 通道1的感应电容输入 |
| 5 | CIN2 | I | 通道2的感应电容输入 |
| 6 | CIN3 | I | 通道3的感应电容输入 |
| 7 | CIN4 | I | 通道4的感应电容输入 |
| 8 | CIN5 | I | 通道5的感应电容输入 |
| 9 | CIN6 | I | 通道6的感应电容输入 |
| 10 | CIN7 | I | 通道7的感应电容输入 |
| 11 | NC | | 保留端口 |
| 12 | GND | I | GND |
| 13 | SDA | I/O | I2C接口的数据端口,内部上拉 |
| 14 | SCL | I | I2C 接口的时钟端口 |
| 15 | OUT7 | O | 通道7的输出 |
| 16 | OUT6 | O | 通道6的输出 |
| 17 | OUT5 | O | 通道5的输出 |
| 18 | OUT4 | O | 通道4的输出 |
| 19 | OUT3 | O | 通道3的输出 |
| 20 | OUT2 | O | 通道2的输出 |
| 21 | OUT1 | O | 通道1的输出 |
| 22 | OUT0 | O | 通道0的输出 |
| 23 | MD | I | 设置芯片的工作模式 |
| 24 | VDD | I | 电源 |

工作模式

| MD输入 | 数字灵敏度设置 | 输出逻辑方式 |
|---------|---------|--------|
| 二分之一VDD | 高灵敏度 | 直接输出 |
| VDD | 普通灵敏度 | 直接输出 |
| GND | 普通灵敏度 | 锁存输出 |

***芯片灵敏度**

由数字灵敏度设置和CIN电容值决定，数字灵敏度设置一般建议设置为普通灵敏度。只有在应用环境外部干扰很小，并且对灵敏度有较高要求的场合，才需要将数字灵敏度设置成高灵敏度。

***输出逻辑方式**

芯片的输出管脚是open-drain方式，有高阻和低电平两种状态。当输出逻辑方式选为直接输出方式时，无触摸时是高阻态，有触摸时为低电平；当输出逻辑方式选为锁存输出时，初始输出为高阻态，触摸动作会使输出转态。

***初始化时间**

芯片复位之后会读取外部电容值做为判断基准值。此过程大约300ms左右。

***自校正功能**

芯片在检测到有按键时，会停止自校正大约30s左右。其余的时间内不停的进行自校正，根据外部环境的漂移来调整基准值。

***省电模式**

没有按键并且SDA端口一直保持高电平的情况下，大约30秒钟芯片会进入省电模式。将SDA端口接低电平或者每次读取I2C的时间间隔不超过30秒钟，都可以使芯片不进入省电模式。

省电模式下，按键采样时间间隔是500ms左右。检测到有按键后会马上进入正常工作模式。

***按键反应时间**

正常工作模式下可以检测到的按键频率大于10次/秒。

额定值*

工作温度..... -40℃ to +85℃

存储温度 -50℃ to +15 0℃

最大工作电压..... 6.5V

管脚的容限电压VDD+0.3v

直流输出电流..... 10.0 mA

***注意:** 超出上述额定值可能导致芯片工作不正常并且导致芯片的永久损坏。

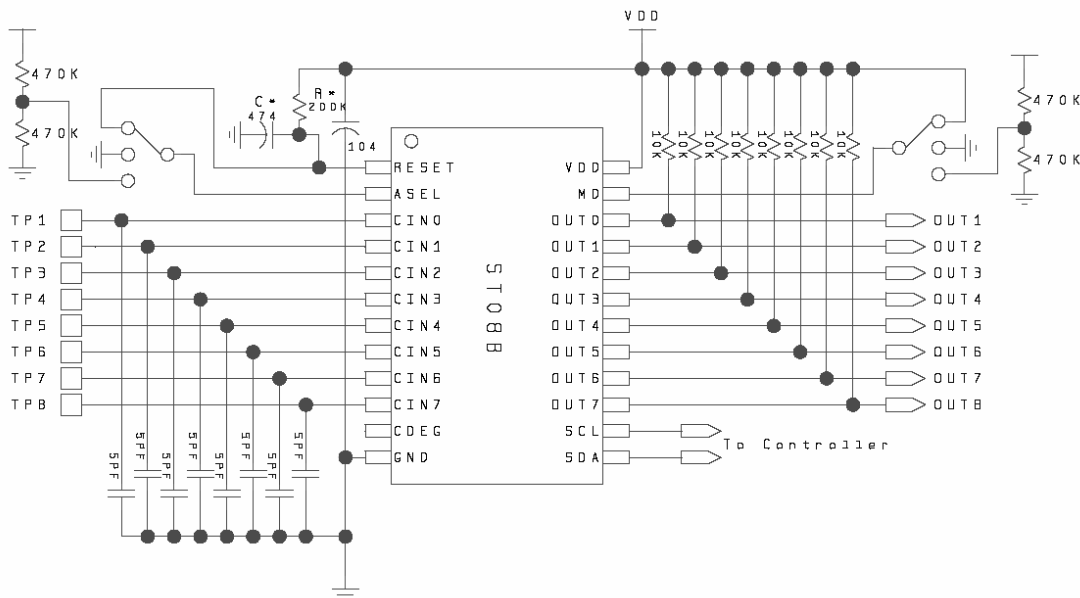
电气特性

T_A = 25℃

| Characteristics | Symbol | Condition | Min | Typ | Max | Units |
|------------------------------------|-----------------|--------------------------------------|-----|------------|-----|-------|
| Operating Voltage | Vdd | | 2.1 | | 6.5 | V |
| Normal Current consumption | I _{nd} | VDD=5.0V | | 220 | | uA |
| Idle Current consumption | I _{id} | VDD=5.0v | | | 10 | uA |
| Output Impedance (open drain) | Z _o | delta Cs > 0.2pF delta Cs < 0.2pF | | 20 100M | | Ohm |

| | | | | | | |
|-------------------------------|----------|-----------------|---|-----|------|----|
| Output Sink Current | Isk | VDD=5V, Rb=500K | | | 10.0 | mA |
| Input capacitance range | Cs | | 0 | | 80 | pF |
| Minimum detective capacitance | delta_Cs | Cs = 5pF | | 0.2 | | pF |

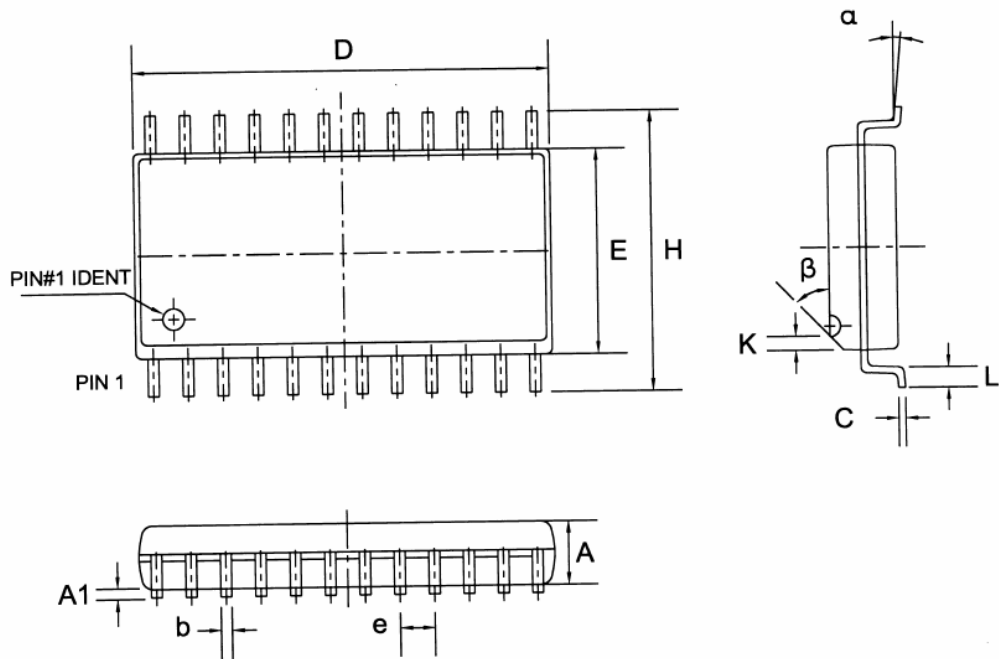
应用线路



注意:

1. RESET为低电平复位，内部有上拉电阻。如果客户对复位时间有特殊要求可以外部接上拉电阻和到GND的电容，当上电比较慢时（大约50MS），C*和R*建议用470NF和200K，普通上电时（10MS以内），用100NF和200K即可。
2. ASEL是I2C通信的地址设置端口，如果系统中只有一片ST08B，该端口接RESET端口即可。此时，主机读命令应设为C5H。
3. CIN0~CIN7为灵敏度设置端口，外接到地电容。电容取值的范围是0pf到50pf。
4. MD接VDD时对应普通灵敏度非锁存输出模式；MD接GND时对应普通灵敏度锁存输出；MD接1/2VDD对应高灵敏度非锁存输出。
5. OUT0~OUT7接MCU的IO端口时建议串联500~1K左右的电阻
6. SCL和SDA和MCU的IO端口建议串联500~1K左右的电阻。如果不使用I2C通信，SDA接低电平可以使芯片不进入省电模式

封装型式 (SOP-24)



| Symbol | Dimensions In Millimeters | | | Dimensions In Inches | | |
|----------|---------------------------|-------|-------|----------------------|-------|-------|
| | Min | Nom | Max | Min | Nom | Max |
| A | 2.35 | 2.50 | 2.65 | 0.093 | 0.098 | 0.104 |
| A1 | 0.10 | 0.20 | 0.30 | 0.004 | 0.008 | 0.012 |
| b | -- | 0.40 | -- | -- | 0.016 | -- |
| C | -- | 0.25 | -- | -- | 0.010 | -- |
| D | 15.10 | 15.40 | 15.70 | 0.594 | 0.606 | 0.618 |
| E | 7.35 | 7.50 | 7.65 | 0.289 | 0.295 | 0.301 |
| e | -- | 1.27 | -- | -- | 0.050 | -- |
| H | 10.15 | 10.45 | 10.75 | 0.400 | 0.411 | 0.423 |
| K | -- | 0.50 | -- | -- | 0.020 | -- |
| L | 0.60 | 0.80 | 1.00 | 0.024 | 0.031 | 0.039 |
| α | 0° | -- | 8° | 0° | -- | 8° |
| β | -- | 45° | -- | -- | 45° | -- |

附录：
MCU 读取 st08B 芯片的示例程序，用 8051 汇编语言编写

```
ReadKey:
    ;;先发送 addr 寄存器的内容, 做为 8bit 读命令
    mov    bitnum, #8
    clr    sda    ;;发送 start 信号
    lcall   Nop10 ;;延时
ReadKey_1:
    clr    scl
    mov    a,    addr
    rlc    a
    mov    addr, a
    mov    sda, c
    lcall   Nop10
    setb   scl
    lcall   Nop10
    djnz   bitnum, ReadKey_1
    ;; 发送 8bit 读命令结束, 开始读是否有 ACK 信号
    clr    scl
    lcall   Nop10
    setb   scl
    lcall   Nop10
    lcall   Nop10
    mov    c,    sda
    jnc    ReadKey_2
    LCALL   Error    ;; 没有收到 ACK 信号, 执行出错处理程序
ReadKey_2:
    ;;开始读 st08B 送出的 8bit 按键数据, 结果存到 key 寄存器
    mov    bitnum, #8
ReadKey_3:
    clr    scl
    lcall   Nop10
    setb   scl
    lcall   Nop10
    mov    c,    sda
    mov    a,    key
    rlc    a
    mov    key,  a
    djnz   bitnum, ReadKey_3
    ;; 按键数据接收完毕
    clr    scl
    lcall   Nop10
    setb   scl
    lcall   Nop10
    clr    scl
    lcall   Nop10
    setb   scl
    ret
```