# TTY7952B 16Key + IIC
# 规格书 V1.7

产品描述: ..... 2
特色: ..... 2
产品应用范围： ..... 2
封装脚位图: ..... 3
脚位定义: ..... 4
电气特性： ..... 5
1 最大绝对额定值 ..... 5
2 DC/AC 特性：（测试条件为室温=25℃） ..... 5
功能描述： ..... 5
特别说明: ..... 17
示范程序: ..... 19
应用参考线路: ..... 28
封装说明： ..... 31
订购信息 ..... 33

# 产品描述:

此方案提供客户一个简单的 1~16 键 IIC 输出应用。

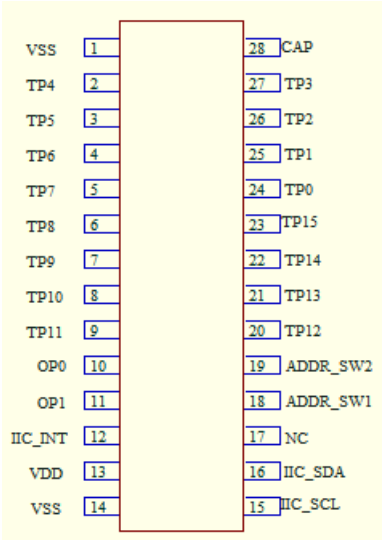
# 特色:

- 程序的独立按键输出模式有两种 Multiple 以及 Single，当选则为 Multiple 时会将所有按下的按键输出，而选择 Single 时只会输出第一个按下的按键，当按键被放开时，才会输出下一个按键。
- 设计有省电模式，适合用在如遥控器等需要长时间待机的应用。
- 有 4 组 Slave address，可由外部开关切换。
- 增加多組睡眠掃描，提高休眠靈敏度。

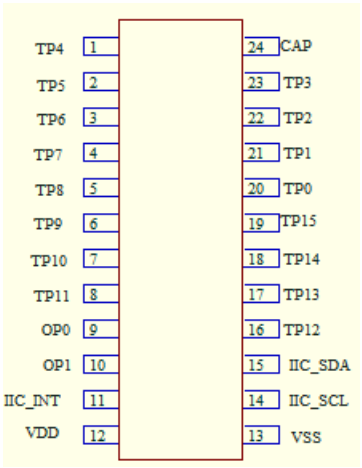
# 产品应用范围:

- 大小家电
- 门禁监控设备
- 消费类电子

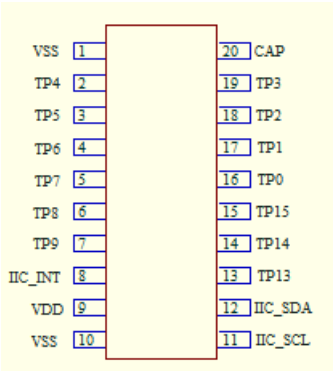
# 封装脚位图:




TTY7952B-16 SSOP28




TTY7952B-16 SSOP24




TTY7952B-13 TSSOP20

Figure5. pin define for IC package

CAP 为量测电容接脚，电容大小约 10nF~39nF。

TP0~TP15 是触摸按键的量测 PAD，TTY7952B 最多可侦测 16 个按键。不使用的按鍵請保持空接。

OP0、OP1 是外部靈敏度選擇腳位:

| OP0 | OP1 | 靈敏度 |
|---|---|---|
| 0 | 0 | 400% |
| 0 | 1 | 200% |
| 1 | 0 | 20% |
| 1 | 1 | 100% |

IIC_SDA 是 IIC 的数据输出/输入脚。

IIC_SCL 是 IIC 的時脈输入脚。

# 脚位定义:

| 28pin | 24pin | 20pin | Define | I/O | Pin Description |
|---|---|---|---|---|---|
| - | - | - | RSTB | I | External reset input, active low 50kΩ pull-up(5v) |
| 13 | 12 | 9 | $V_{DD}$ | Power | Positive power supply |
| 14 | 13 | 10 | $V_{SS}$ | Power | Negative power supply, ground |
| 28 | 24 | 20 | CAP | I | Touch sensor input |
| 1 | 13 | 1 | $V_{SS}$ | Power | |
| 10 | 9 | - | OP0 | I | Sensitivity select option 0 |
| 11 | 10 | - | OP1 | I | Sensitivity select option 1 |
| 12 | 11 | 8 | IIC_INT | IO | IIC interrupt pin |
| - | - | - | - | - | |
| 15 | 14 | 11 | IIC_SCL | IO | IIC clock pin |
| 16 | 15 | 12 | IIC_SDA | IO | IIC data pin |
| 18 | - | - | ADDR_SW1 | I | IIC slave address select |
| 19 | - | - | ADDR_SW2 | I | IIC slave address select |
| 24 | 20 | 16 | TP0 | IO/I | touch pad input |
| 25 | 21 | 17 | TP1 | IO/I | touch pad input |
| 26 | 22 | 18 | TP2 | IO/I | touch pad input |
| 27 | 23 | 19 | TP3 | IO/I | touch pad input |
| 2 | 1 | 2 | TP4 | IO/I | touch pad input |
| 3 | 2 | 3 | TP5 | IO/I | touch pad input |
| 4 | 3 | 4 | TP6 | IO/I | touch pad input |
| 5 | 4 | 5 | TP7 | IO/I | touch pad input |
| 6 | 5 | 6 | TP8 | IO/I | touch pad input |
| 7 | 6 | 7 | TP9 | IO/I | touch pad input |
| 8 | 7 | - | TP10 | IO/I | touch pad input |
| 9 | 8 | - | TP11 | IO/I | touch pad input |
| 20 | 16 | - | TP12 | IO/I | touch pad input |
| 21 | 18 | 13 | TP13 | IO/I | touch pad input |
| 23 | 18 | 14 | TP14 | IO/I | touch pad input |
| 24 | 19 | 15 | TP15 | IO/I | touch pad input |

Table1. TTY7952B pin descriptio

# 电气特性：

1 最大绝对额定值

| 参数 | 符号 | 条件 | 值 | 单位 |
|---|---|---|---|---|
| 工作温度 | Top | -—— | -40~+85 | ℃ |
| 存放温度 | $T_{STG}$ | —— | -50~+125 | ℃ |
| 电源电压 | VDD | Ta=25℃ | VSS-0.3~VSS+5.5 | V |
| 输入电压 | $V_{IN}$ | Ta=25℃ | VSS-0.3~VDD+0.3 | V |
| 芯片抗静电强度 HBM | ESD | —— | >5 | KV |
| 备注：VSS 代表系统接地 | | | | |

2 DC/AC 特性：（测试条件为室温=25℃）

| 参数 | 符号 | 测试条件 | 最小值 | 典型值 | 最大值 | 单位 |
|---|---|---|---|---|---|---|
| 工作电压 | VDD | | 2.5 | - | 5.5 | V |
| 系统震荡频率 | F | VDD=5V | - | 4M | - | Hz |
| 工作电流 | $I_{OP}$ | 待机，VDD=3V 输出无负载 | - | 1.1 | - | mA |
| | $I_{OFF}$ | 待机，VDD=3V 输出无负载 | 5.3 | 6.8 | 10.0 | uA |

# 功能描述：

*触摸按键介绍:*

触摸按键是利用测量人体接近导体时产生的电容变化，转换为数值判断的一种方式。此应用中所有的触摸按键都有 **Threshold** 设定参数，用来调整触摸按键的灵敏度。

Threshold 依照按键的按压深度来做调整，数值越小越灵敏，但也越容易受到噪声干扰，需根据实际应用的数据来调整。

## *IIC 协定*:

IC 使用 IIC 数据传输协议，两线式总线 SCL、SDA 来读写数据。INT 脚位用来通知 Master 有按键状态变化。

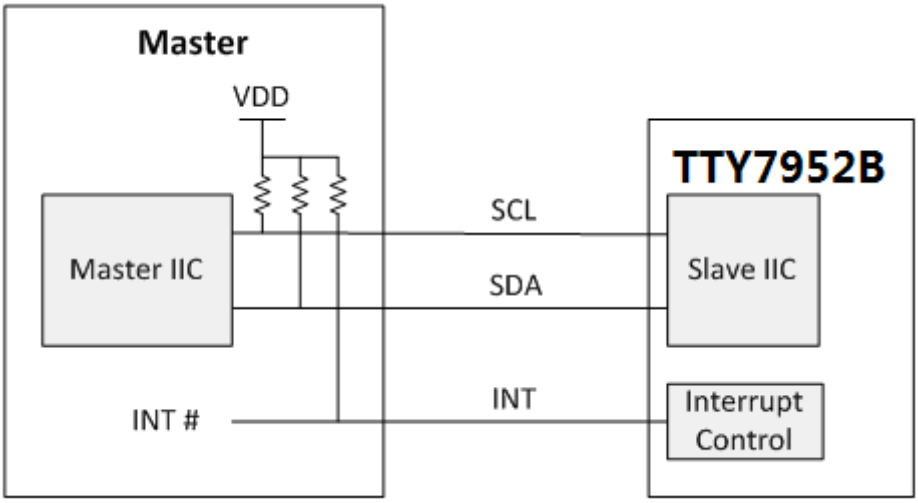



Figure6. IIC connect for master and TTY7952B

INT 在无按键状态变化时为 High，当有按键状态变化时，INT 脚位会拉 Low 100ms。若 Slave 接收到 Slave address 则会清除回复为 High。

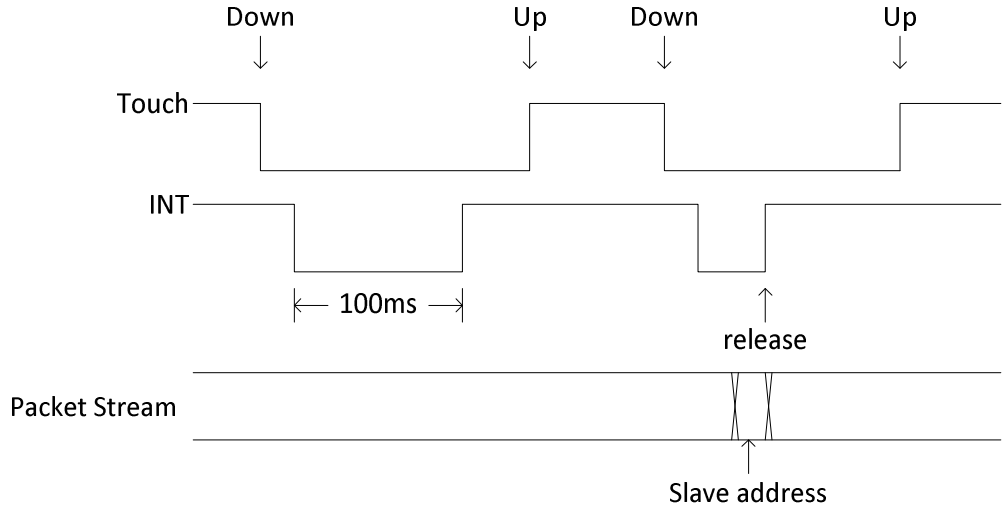



Figure7. INT pin describe

## Switching Characteristics

| Symbol | Description | Min | Max | Units |
|---|---|---|---|---|
| FSCL | SCL clock frequency. | 0 | 100 | KHz |
| THDSTA | Hold time(repeated) star condition. After this period, the first clock pulse is generated. | 4.0 | - | us |
| TLOW | Low period of the SCL clock. | 4.7 | - | us |
| THIGH | High period of the SCL clock. | 4.0 | - | us |
| TSUSTA | Set-up time for a repeated start condition. | 4.7 | - | us |
| THDDAT | Data hold time. | 0 | - | us |
| TSUDAT | Data set-up time. | 250 | - | ns |
| TSUSTO | Set-up time for stop condition. | 4.0 | - | us |
| TBUF | Bus free time between a stop and start condition. | 4.7 | - | us |
| TSPI | Pulse width of spikes are suppressed by the input filter. | 0 | 50 | ns |
| TSPT | Slave processor time | 10 | 75 | us |

Table3. AC characteristics of the IIC SDA and SCL pins for vdd

## Timing Waveform

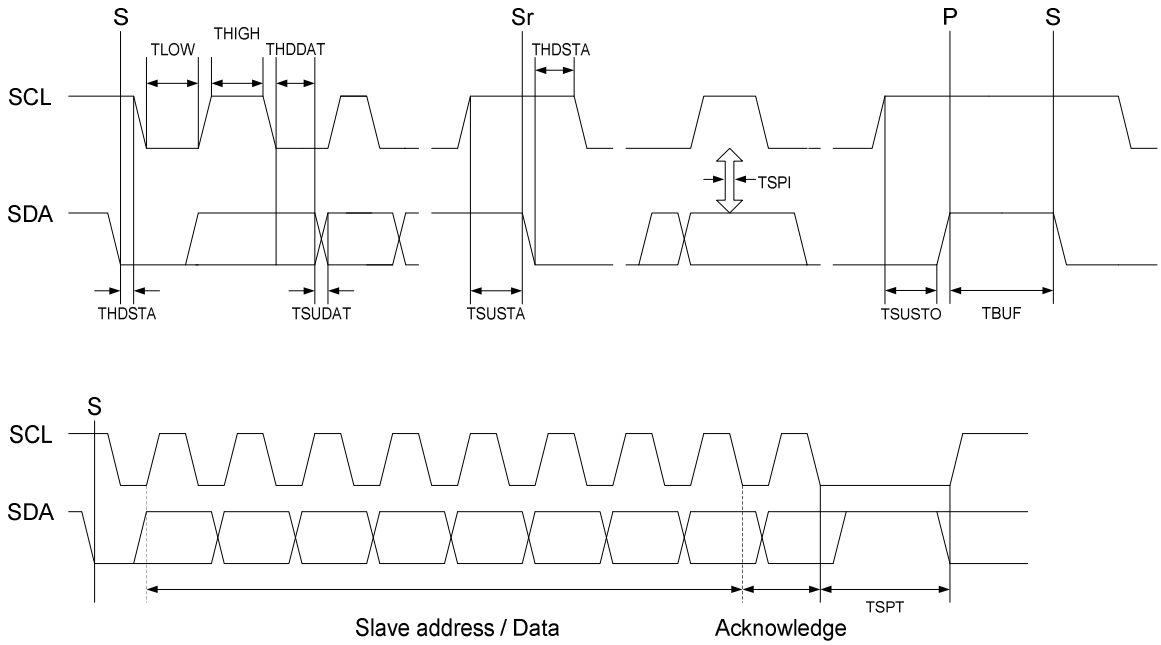



Figure8. Definition for timing for fast/standard mode on the IIC

## Packet Stream

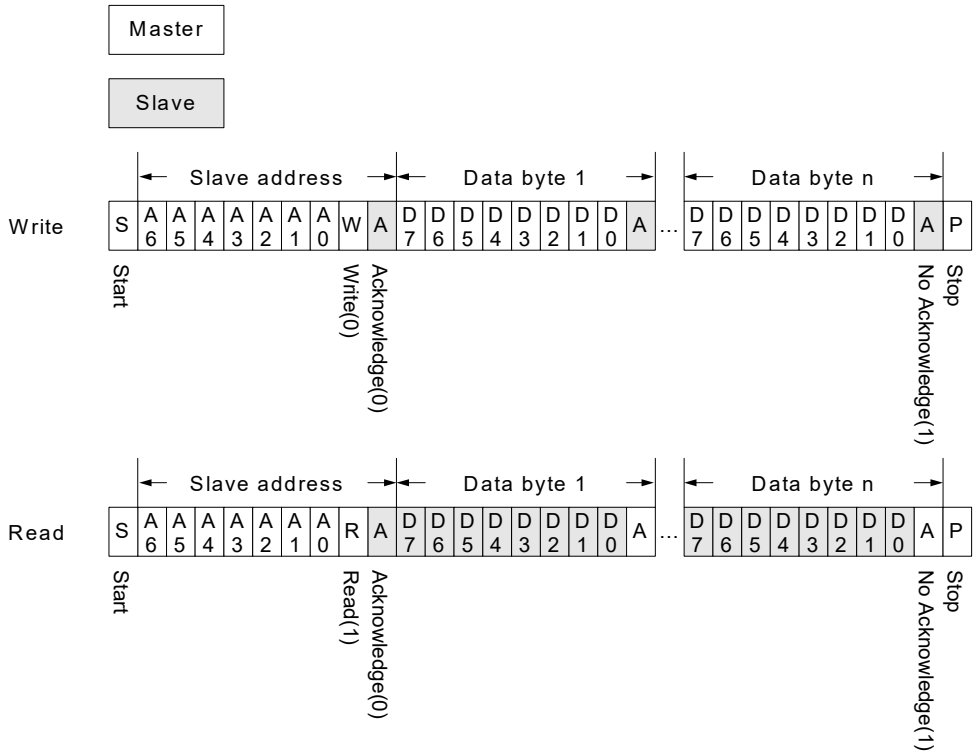



Figure9. Write / Read byte form I2C

## Slave address

可透过 ADDR_SW1，ADD_SW2 脚位做切换，如下表

| ADDR_SW1 | ADDR_SW2 | Slave address (A6-A0) | Write (A6-A0,R) | Read (A6-A0,R) |
|---|---|---|---|---|
| 0 | 0 | 50H | A0H | A1H |
| 0 | 1 | 51H | A2H | A3H |
| 1 | 0 | 52H | A4H | A5H |
| 1 | 1 | 53H | A6H | A7H |

ADDR_SW1，ADDR_SW2 空接时 Slave address 为 53H，如果封装没有这两个脚位默认为 53H。

## *Data Stream:*

软件设计为 **16 键应用模式**。写入数据以 3 Data Bytes 为一组资料串流。当一笔数据串流写入完成后，系统会将数据覆写并进行**系统重设**。若写入被中断并重新写入，则前一笔数据会被放弃。

在每次写入完一组 3 Data Bytes 后，若要再次写入下一组设定，需要送出 Stop 讯号结束当前数据传输，再重新写入下一组设定。

# 16 Key Application mode

**Write Data**

### 1. Setting commands

| Data byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1 | IICM=1 | CT=0 | KOM | VREFS | PSM | LDO | KRT | |
| 2 | Key Num | | | | SA | KAT | | |
| 3 | Key Off Num | | | | - | MCKS | | |

**IICM**

IIC 数据模式选择。

| **IICM** | **IIC Mode** |
|---|---|
| 0 | Reserve |
| 1 | Application mode (Define) |

**CT**

在 Wheel application 模式，写入数据区分成应用设定以及阀值设定，当 CT 为 0 时是写入应用设定，当 CT=1 时是写入阀值设定。

| **CT** | **Custom Threshold** |
|---|---|
| 0 | Setting commands |
| 1 | Custom threshold commands |

**KOM**

按键输出模式，有多个按键输出以及单一按键输出两种模式。此选项是对普通按键的输出设定，滑条按键则不受影响。单一按键输出模式时只会输出第一个被按下的按键，当按键放开后才会承认其它按键。

| **KOM** | **Key Output Mode** |
|---|---|
| 0 | Multiple |
| 1 | Single (Define) |

**VREFS**

按键扫描的参考电压。

| VREFS | |
|---|---|
| 0 | 1/2 VDD |
| 1 | 2/3VDD (Define) |

**PSM**

省电模式，无按键 4 秒后进入睡眠模式。

| PSM | Power Save Mode |
|---|---|
| 0 | Disable(Define) |
| 1 | Enable |

**LDO**

Touch IP 內部 LDO 開關

| LDO | |
|---|---|
| 0 | Disable(Define) |
| 1 | Enable |

**KRT**

长按键复位时间设定，在按键按下开始计时，时间到自动复位。

若按键输出发送变化则重新开始计时。

| KRT | | Key Reset Time |
|---|---|---|
| 0 | 0 | Disable |
| 0 | 1 | 15 second (Define) |
| 1 | 0 | 30 second |
| 1 | 1 | 60 second |

**SA**

掃描平均模式

| SA | Scan Average |
|---|---|
| 0 | Disable(Define) |
| 1 | Enable |

**KAT**

按键消抖时间。

| KAT | | | Key Acknowledge Times |
|---|---|---|---|
| 2 | 1 | 0 | |
| 0 | 0 | 0 | 3 times |
| 0 | 0 | 1 | 4 times (Define) |
| 0 | 1 | 0 | 5 times |
| 0 | 1 | 1 | 6 times |
| 1 | 0 | 0 | 7 times |
| 1 | 0 | 1 | 8 times |
| 1 | 1 | 0 | 9 times |
| 1 | 1 | 1 | 10 times |

**Key Num**

**按鍵數設定**

| Key Num | | | | Key Number |
|---|---|---|---|---|
| 3 | 2 | 1 | 0 | |
| 0 | 0 | 0 | 0 | 1 key |
| 0 | 0 | 0 | 1 | 2 keys |
| 0 | 0 | 1 | 0 | 3 keys |
| 0 | 0 | 1 | 1 | 4 keys |
| 0 | 1 | 0 | 0 | 5 keys |
| 0 | 1 | 0 | 1 | 6 keys |
| 0 | 1 | 1 | 0 | 7 keys |
| 0 | 1 | 1 | 1 | 8 keys |
| 1 | 0 | 0 | 0 | 9 keys |
| 1 | 0 | 0 | 1 | 10 keys |
| 1 | 0 | 1 | 0 | 11 keys |
| 1 | 0 | 1 | 1 | 12 keys (Define) |
| 1 | 1 | 0 | 0 | 13 keys |
| 1 | 1 | 0 | 1 | 14 keys |
| 1 | 1 | 1 | 0 | 15 keys |
| 1 | 1 | 1 | 1 | 16 keys |

**Key Off Num**

多按键重置设定，最多为 16 Keys。

| **Key Off Num** | | | | **Key Off Number** |
|---|---|---|---|---|
| 3 | 2 | 1 | 0 | |
| 0 | 0 | 0 | 0 | Disable (Define) |
| 0 | 0 | 0 | 1 | 2key reset |
| 0 | 0 | 1 | 0 | 3 keys reset |
| 0 | 0 | 1 | 1 | 4 keys reset |
| 0 | 1 | 0 | 0 | 5 keys reset |
| 0 | 1 | 0 | 1 | 6 keys reset |
| 0 | 1 | 1 | 0 | 7 keys reset |
| 0 | 1 | 1 | 1 | 8 keys reset |
| 1 | 0 | 0 | 0 | 9 keys reset |
| 1 | 0 | 0 | 1 | 10 keys reset |
| 1 | 0 | 1 | 0 | 11 keys reset |
| 1 | 0 | 1 | 1 | 12 keys reset |
| 1 | 1 | 0 | 0 | 13 keys reset |
| 1 | 1 | 0 | 1 | 14 keys reset |
| 1 | 1 | 1 | 0 | 15 keys reset |
| 1 | 1 | 1 | 1 | 16 keys reset |

MCKS

| **MCKS** | | | Modulation clock selector | 当开启睡眠时，如设定的是偶数除频，则程序自动切换为奇数除频。 |
|---|---|---|---|---|
| 2 | 1 | 0 | | |
| 0 | 0 | | RC8M/2(Define) | RC8M/3 |
| 0 | 0 | 1 | RC8M/3 | |
| 0 | 1 | 0 | RC8M/4 | RC8M/5 |
| 0 | 1 | 1 | RC8M/5 | |
| 1 | 0 | 0 | RC8M/8 | |
| 1 | 0 | 1 | RC8M/16 | |
| 1 | 1 | 0 | RC8M/32 | |
| 1 | 1 | 1 | OSCL | OSCL |

Note:开启睡眠时，MCKS 需要选择 RC8M/3，RC8M/5，才能保证睡眠稳定。

## 2. Custom threshold commands

阀值则是设定按键承认的门坎。分为按键阀值，睡眠唤醒阀值两种。

**Item**

选择切换不同的写入参数的设定。

| Item | | Item |
|---|---|---|
| 0 | 0 | TPx setting |
| 0 | 1 | Sleep setting |
| 1 | 0 | Reference Adjust Speed |
| 1 | 1 | - |

- **TPx Setting**

| Data byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1 | IICM=1 | CT=1 | Item=0 | | TP Num | | | |
| 2 | TPx Threshold M | | | | TPx Threshold L | | | |
| 3 | | | | | TPx Threshold H | | | |

TPx Threshold： 按键承认阀值。(Define : 010H) (max: FFFH ,min: 001H)

**按键承认阀值越小灵敏度越高**，越大灵敏度越低。预设的阀值为 010H，建议的最小值为 008H，若调整到 008H 按键灵敏度仍然不够，则建议加大 CS 电容，CS 电容的值则建议小于 39nF。

**TP Num**

数据写入的按键编号。

| TP NUM | | | | TP Number |
|---|---|---|---|---|
| 3 | 2 | 1 | 0 | |
| 0 | 0 | 0 | 0 | TP0 |
| 0 | 0 | 0 | 1 | TP1 |
| 0 | 0 | 1 | 0 | TP2 |
| 0 | 0 | 1 | 1 | TP3 |
| 0 | 1 | 0 | 0 | TP4 |
| 0 | 1 | 0 | 1 | TP5 |
| 0 | 1 | 1 | 0 | TP6 |
| 0 | 1 | 1 | 1 | TP7 |
| 1 | 0 | 0 | 0 | TP8 |
| 1 | 0 | 0 | 1 | TP9 |
| 1 | 0 | 1 | 0 | TP10 |
| 1 | 0 | 1 | 1 | TP11 |
| 1 | 1 | 0 | 0 | TP12 |
| 1 | 1 | 0 | 1 | TP13 |
| 1 | 1 | 1 | 0 | TP14 |
| 1 | 1 | 1 | 1 | TP15 |

- **Sleep Setting**

**Group 1**

| Data byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1 | IICM=1 | CT=1 | Item=1 | | 0 | | | |
| 2 | TPSLP Threshold M | | | | TPSLP Threshold L | | | |
| 3 | | | | | TPSLP Threshold H | | | |

TPSLP Threshold : 省电模式唤醒阀值。(Define : 002H)(max: FFFH ,min: 001H)

| Data byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1 | IICM=1 | CT=1 | Item=1 | | 1 | | | |
| 2 | TP7 | TP6 | TP5 | TP4 | TP3 | TP2 | TP1 | TP0 |
| 3 | TP15 | TP14 | TP13 | TP12 | TP11 | TP10 | TP9 | TP8 |

**TP0~TP15:睡眠扫描通道设定(Define:000FH)**

**0000H=扫描所有使用中的按键。**

**Group 2**

| Data byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1 | IICM=1 | CT=1 | Item=1 | | 2 | | | |
| 2 | TPSLP Threshold M | | | | TPSLP Threshold L | | | |
| 3 | | | | | TPSLP Threshold H | | | |

TPSLP Threshold : 省电模式唤醒阀值。(Define : 002H) (max: FFFH ,min: 001H)

| Data byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1 | IICM=1 | CT=1 | Item=1 | | 3 | | | |
| 2 | TP7 | TP6 | TP5 | TP4 | TP3 | TP2 | TP1 | TP0 |
| 3 | TP15 | TP14 | TP13 | TP12 | TP11 | TP10 | TP9 | TP8 |

**TP0~TP15:睡眠扫描通道设定(Define:00F0H)** (max: FFFH ,min: 001H)

**0000H=不使用。**

**Group 3**

| Data byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1 | IICM=1 | CT=1 | Item=1 | | 4 | | | |
| 2 | TPSLP Threshold M | | | | TPSLP Threshold L | | | |
| 3 | | | | | TPSLP Threshold H | | | |

TPSLP Threshold : 省电模式唤醒阀值。(Define : 002H) (max: FFFH ,min: 001H)

| Data byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1 | IICM=1 | CT=1 | Item=1 | | 5 | | | |
| 2 | TP7 | TP6 | TP5 | TP4 | TP3 | TP2 | TP1 | TP0 |
| 3 | TP15 | TP14 | TP13 | TP12 | TP11 | TP10 | TP9 | TP8 |

**TP0~TP15:睡眠扫描通道设定(Define:0F00H)**

**0000H=不使用。**

**Group 4**

| Data byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1 | IICM=1 | CT=1 | Item=1 | | 6 | | | |
| 2 | TPSLP Threshold M | | | | TPSLP Threshold L | | | |
| 3 | | | | | TPSLP Threshold H | | | |

TPSLP Threshold : 省电模式唤醒阀值。(Define : 002H) (max: FFFH ,min: 001H)

| Data byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1 | IICM=1 | CT=1 | Item=1 | | 7 | | | |
| 2 | TP7 | TP6 | TP5 | TP4 | TP3 | TP2 | TP1 | KEY1 |
| 3 | - | - | - | TP12 | TP11 | TP10 | TP9 | TP8 |

**TP0~TP15:睡眠扫描通道设定(Define:0000H)**

**0000H=不使用。**

Reference Adjust Speed :環境值校準速度

| Data byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1 | IICM=1 | CT=1 | Item=2 | | 0 | | | |
| 2 | Up adjust speed | | | | | | | |
| 3 | Down adjust speed | | | | | | | |

Up adjust speed (Define:80H@8ms) (max: FFH ,min: 01H)

Down adjust speed (Define:80H@8ms) (max: FFH ,min: 01H)

**Read Data**

| Data byte | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1 | C | WSET | | | | | | |
| 2 | Key 8 | Key 7 | Key 6 | Key 5 | Key 4 | Key 3 | Key 2 | Key 1 |
| 3 | Key 16 | Key 15 | Key 14 | Key 13 | Key 12 | Key 11 | Key 10 | Key 9 |

**C**

系统校正标志，当值为 0 时，表示系统校正中，键值读取无效。当值为 1 时，键值有效。

| **C** | **Calibrate** |
|---|---|
| 0 | Calibrating |
| 1 | Calibrate Finish |

**WSET**

系统写入标志，上电为 1，写入设定后该标志设置为 0。

| **WSET** | **Have write setting** |
|---|---|
| 0 | Have write setting |
| 1 | No write setting |

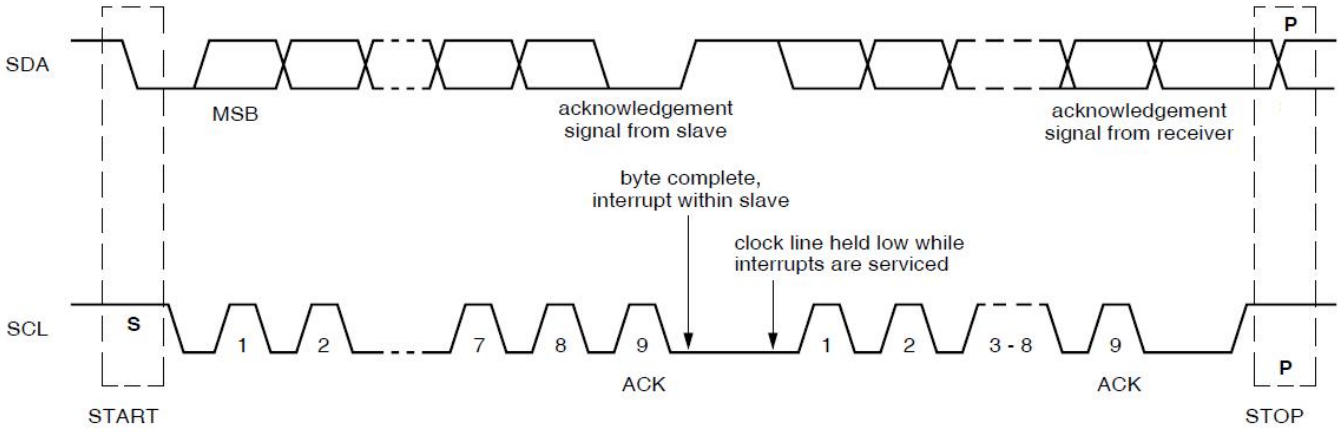
**K1…K16**

触摸按键标志，无按键为 0，有按键为 1。

| **K1…K16** | **Key1…Key16** |
|---|---|
| 0 | No touch |
| 1 | Touch |

# 特别说明:

1. TTY7952B的I2C接口有硬件的支持SCL可支持100KHz，但是译码为软件处理，处理约需20~100us视处理的情况而定，所以当Master发送完一组数据封包，应增加大于200us的延时，再发送下一组数据封包。




所以Master写程序时，需注意SCL拉Low的动作，若由硬件控制大多会支持此标准，若由程序控制IO脚，请增加对SCL输出High时要读回确认为High，才可让程序继续进行，若为Low应等待SCL为High后才可继续进行。C的程序如下：

```
SCL=1;
While(SCL!=1) { };
```

2. 若需要连续读取键值，建议读取完后暂停10ms以上，再读取下一次键值。否则会影响按键的反应速度。
3. 若开启睡眠模式，则禁止连续读取键值，因为每次读取键值时，都会清除进入睡眠的计时。，会导致系统无法睡眠。
4. 在系统进入睡眠模式时，会将IIC功能关闭。此时重新下IIC 指令可以唤醒系统，但是会收到no ACK的响应，需要等待系统唤醒后再重新下读写命令。
5. 按键阀值调整的步骤:

   Step1. 选择初始测试用的 CS 电容(见建议线路图):
   先确定设计中是否使用滑条功能，若使用滑条功能，则建议 33nF 作为 CS 充放电电容，若仅做一般按键使用，则建议使用 10nF 作为初始测试电容。

   Step2. 每个按键做按压测试:
   以正常速度轻触按键或使用金属棒做测试条件，若在触摸到按键之前有按键输出，表示灵敏度太高，需要调高阀值，若触摸按键没有按键输出，或是要重压才有按键输出，表示灵敏度太低，需要降低阀值。

滑条按键因为锯齿状设计，在不同位置灵敏度也会不同，故建议做灵敏度测试时，以两个锯齿按键中间位置做灵敏度测试，避免滑动效果不佳。

Step3. 测试按键反应速度:

在判断按键灵敏度的时候，若觉得按键”不够灵敏”，需要进一步判断是按键响应速度不够快，还是按键灵敏度不够。判断方法是触摸停留一段时间(约 1 秒)，并检查是否有按键输出。若没有按键输出，则是按键不够灵敏，重新进行 Step2 调整，若有按键输出，则是按键响应速度不构快，则进行下一步。

Step4. 调整按键反应速度:

按键消抖时间(KAT)预设为 4，若按键反应速度不构快，可以下修值到 3。

若下修到 3 反应速度仍然不够，则建议将 CS 电容减小。

选择好适当的 CS 电容后需要回到 Step2 重新调整灵敏度。

需要注意的是选择较小的 CS 电容，同时会降低滑条按键的精细度。

# 示范程序:

```
/*
    项目名称:主控端对 TTY7952B 透过 IIC 控制的范例程序
    项目目的:1.透过软件模拟 IIC 主控端对 TTY7952B 写入设定参数
             2.透过软件模拟 IIC 主控端对 TTY7952B 读取按键状态
    主控 MCU:AT89C51
    Date & Version:    2016/01/06 v1.0
*/
//-------------------------------------------------------------------------------------------
#include<reg51.h>
#include<intrins.h>
#define uint unsigned int
#define uchar unsigned char
#define address_W 0xa6 //从机的地址和写入标志
#define address_R 0xa7 //从机的地址和读取标志

sbit SINT=P0^0; //主控端与从机的 IIC 接口
sbit SDA=P0^1; //主控端与从机的 IIC 接口
sbit SCL=P0^2; //主控端与从机的 IIC 接口
uchar Write_Buffer[3]; //主控端的写入资料缓存
uchar Read_Buffer[3]; //主控端的读取资料缓存
//-------------------------------------------------------------------------------------------
//函数名称: void delay(uint x)
//函数功能: 程序延时
//函数输入: x
//函数输出: 无
//中间变量: i, j
//-------------------------------------------------------------------------------------------
void delay(uint x)
{
    uint i,j;
    for(i=x;i>0;i--)
        for(j=0x40;j>0;j--);
}
//-------------------------------------------------------------------------------------------
//函数名称: void sendStart()
```

```
//函数功能: IIC 的起始位
//函数输入: 无
//函数输出: 无
//中间变量: 无
//-----------------------------------------------------------------------------------------
void sendStart() //开始位
{
    SDA=1; /*发送起始条件的数据信号*/
    SCL=1;
    while(SCL!=1) { };
    SDA=0; /*发送起始信号*/
    _nop_();
    SCL=0; /*此位置只需要将 SCL 输出为 0 之后等待 4US 即可*/
}
//-----------------------------------------------------------------------------------------
//函数名称: void sendStop()
//函数功能: IIC 的结束位
//函数输入: 无
//函数输出: 无
//中间变量: 无
//-----------------------------------------------------------------------------------------
void sendStop() //停止位
{
    SCL=0;
    SDA=0; /*发送结束条件的数据信号*/
    _nop_();
    SCL=1;
    while(SCL!=1) { };
    _nop_();
    SDA=1;
}
//-----------------------------------------------------------------------------------------
//函数名称: bit readACK()
//函数功能: 读取 IIC 的 acknowledge 标志位
//函数输入: 无
//函数输出: IIC 的 ACK 信号  返回 1 表示无 acknowledge，0 表示有 acknowledge
//中间变量: 无
//-----------------------------------------------------------------------------------------
bit readACK() //读取应答信号
```

```
{
    SCL=0;
    SDA=1; /*此处为释放 SDA 总线，由从从机发出低电平应答*/
    _nop_();
    SCL=1;
    _nop_();
    if(SDA)
        return 1; //no ACK
    else
        return 0; //ACK
}
//-----------------------------------------------------------------------------------------
//函数名称: void sendACK()
//函数功能: 主控端送出应答信号
//函数输入: 无
//函数输出: 无
//中间变量: 无
//-----------------------------------------------------------------------------------------
void sendACK() //输出应答信号
{
    SCL=0;
    SDA=0;
    _nop_();
    SCL=1;
}
//-----------------------------------------------------------------------------------------
//函数名称: void sendNOACK()
//函数功能: 主控端送出无应答信号
//函数输入: 无
//函数输出: 无
//中间变量: 无
//-----------------------------------------------------------------------------------------
void sendNOACK() //输出无应答信号
{
    SCL=0;
    SDA=1;
    _nop_();
    SCL=1;
}
```

```
//-------------------------------------------------------------------------------------------------
//函数名称: void sendByte(uchar dat)
//函数功能: 主控端写一个字节到从机
//函数输入: dat = 发送的字节
//函数输出: 无
//中间变量: i
//-------------------------------------------------------------------------------------------------
void sendByte(uchar dat) //写一个字节
{
    uchar i;
    for(i=0;i<8;i++)
    {
        SCL=0; /*钳住 I2C 总线，准备发送数据 */
        if(dat&0x80)
            SDA=1;
        else
            SDA=0;
        _nop_(); /*如果需要在 SDA,SCL,INT 上串接电阻，根据电阻大小不同，电阻越大建议将该
时间适当加长，100KHZ 以内即可； */
        _nop_();
        SCL=1; /*此处由于 51 单片机的特性不需要做输入输出设置，
但如果是其他单片机需要先将其 IO 口改为输入上拉的设置，读到高之后，SCL 转为输出为高。
在读写完 ACK 后的第一个 clock 下降缘从机会钳住 SCL 脚做资料处理，
所以将 SCL 脚置为输入上拉，并等待 SCL 被释放。*/
        while(SCL!=1) { };
        dat<<=1;
    }
}
//-------------------------------------------------------------------------------------------------
//函数名称: uchar readByte()
//函数功能: 主控端对从机读取一个字节
//函数输入: 无
//函数输出: 读取完成的字节
//中间变量: i, dat
//-------------------------------------------------------------------------------------------------
uchar readByte() //读一个字节
{
    uchar i, dat=0;
    for(i=0;i<8;i++)
```

```
    {
        SCL=0;
        SDA=1;
        _nop_(); /*如果需要在 SDA,SCL,INT 上串接电阻，根据电阻大小不同，电阻越大建议将该
时间适当加长，100KHZ 以内即可；*/
        dat<<=1;
        SCL=1; /*此处由于 51 单片机的特性不需要做输入输出设置，
但如果是其他单片机需要先将其 IO 口改为输入上拉的设置，读到高之后，SCL 转为输出为高。
在读写完 ACK 后的第一个 clock 下降缘从机会钳住 SCL 脚做资料处理，
所以将 SCL 脚置为输入上拉，并等待 SCL 被释放。*/
        while(SCL!=1) { };
        if(SDA==1)
        dat|=0x01;
    }
    return dat;
}
//-------------------------------------------------------------------------------------------------
//函数名称: bit writeIIC(uchar addrW, uchar *writeData, uchar length)
//函数功能: 主控端对从机数据写入
//函数输入: addrW = 从机地址及写入旗帜
//          *writeData = 预备写入数据的首个地址
//          length = 写入数据的长度(字节数)
//函数输出: 返回 IIC 通讯的 acknowledge 状态，若为 1，则停止并返回。若为 0，则完成通讯后返回
//中间变量: i, ACK
//-------------------------------------------------------------------------------------------------
bit writeIIC(uchar addrW, uchar *writeData, uchar length)
{
    uchar i;
    bit ACK;
    sendStart();
    sendByte(addrW); //传送地址与写入标记
    ACK = readACK();
    if (ACK)
    {
        sendStop(); //地址不正确或装置未连接，送出停止信号
        return ACK;
    }

    for(i = 0; i<length; i++)
```

```
    {
        sendByte(writeData[i]);
        ACK = readACK();
        if (ACK)
        {
            sendStop(); //未接收到 ACK，送出停止信号
            return ACK;
        }
    }
    sendStop(); //资料写入完成，送出停止信号
    return ACK;
}
//-----------------------------------------------------------------------------------------------
//函数名称: bit readIIC(uchar addrR, uchar *readData, uchar length)
//函数功能: 主控端对从机数据读取
//函数输入: addrR = 从机地址及读取旗帜
//          *readData = 预备读取后存放数据的首个地址
//          length = 读取数据的长度(字节数)
//函数输出: 返回 IIC 通讯的 acknowledge 状态，若为 1，则停止并返回。若为 0，则完成通讯后返回
//中间变量: i, ACK
//-----------------------------------------------------------------------------------------------
bit readIIC(uchar addrR, uchar *readData, uchar length)
{
    uchar i;
    bit ACK;
    sendStart();
    sendByte(addrR); //传送地址与读取标记
    ACK = readACK();
    if (ACK)
    {
        sendStop(); //地址不正确或装置未连接，送出停止信号
        return ACK;
    }

    for(i = 0; i<length; i++)
    {
        readData[i] = readByte();
        if(i<length-1)
            sendACK();
```

```
            else
                sendNOACK(); //读取最后一笔资料，送出 No ACK
        }
        sendStop(); //资料读取完成，送出停止信号
        return ACK;
}
//-----------------------------------------------------------------------------------------------
//函数名称: void setWrite_Buffer_3(uchar byte1, uchar byte2, uchar byte3)
//函数功能: 写入 3 个字节到写入缓存寄存器
//函数输入: byte1
//                byte2
//                byte3
//函数输出: 无
//中间变量: 无
//-----------------------------------------------------------------------------------------------
void setWrite_Buffer_3(uchar byte1, uchar byte2, uchar byte3)
{
        Write_Buffer[0] = byte1;
        Write_Buffer[1] = byte2;
        Write_Buffer[2] = byte3;
}
void main()
{
        bit ACK;
        SINT = 1;
        setWrite_Buffer_3(0xB1, 0xB9, 0x00);
        ACK = writeIIC(address_W, &Write_Buffer, 3); //MCU Setting
        delay(1); //delay 1ms
        setWrite_Buffer_3(0xC0, 0x10, 0x00);
        ACK = writeIIC(address_W, &Write_Buffer, 3); //TP0 Threshold
        delay(1); //delay 1ms
        setWrite_Buffer_3(0xC1, 0x10, 0x00);
        ACK = writeIIC(address_W, &Write_Buffer, 3); //TP1 Threshold
        delay(1); //delay 1ms
        setWrite_Buffer_3(0xC2, 0x10, 0x00);
        ACK = writeIIC(address_W, &Write_Buffer, 3); //TP2 Threshold
        delay(1); //delay 1ms
        setWrite_Buffer_3(0xC3, 0x10, 0x00);
        ACK = writeIIC(address_W, &Write_Buffer, 3); //TP3 Threshold
```

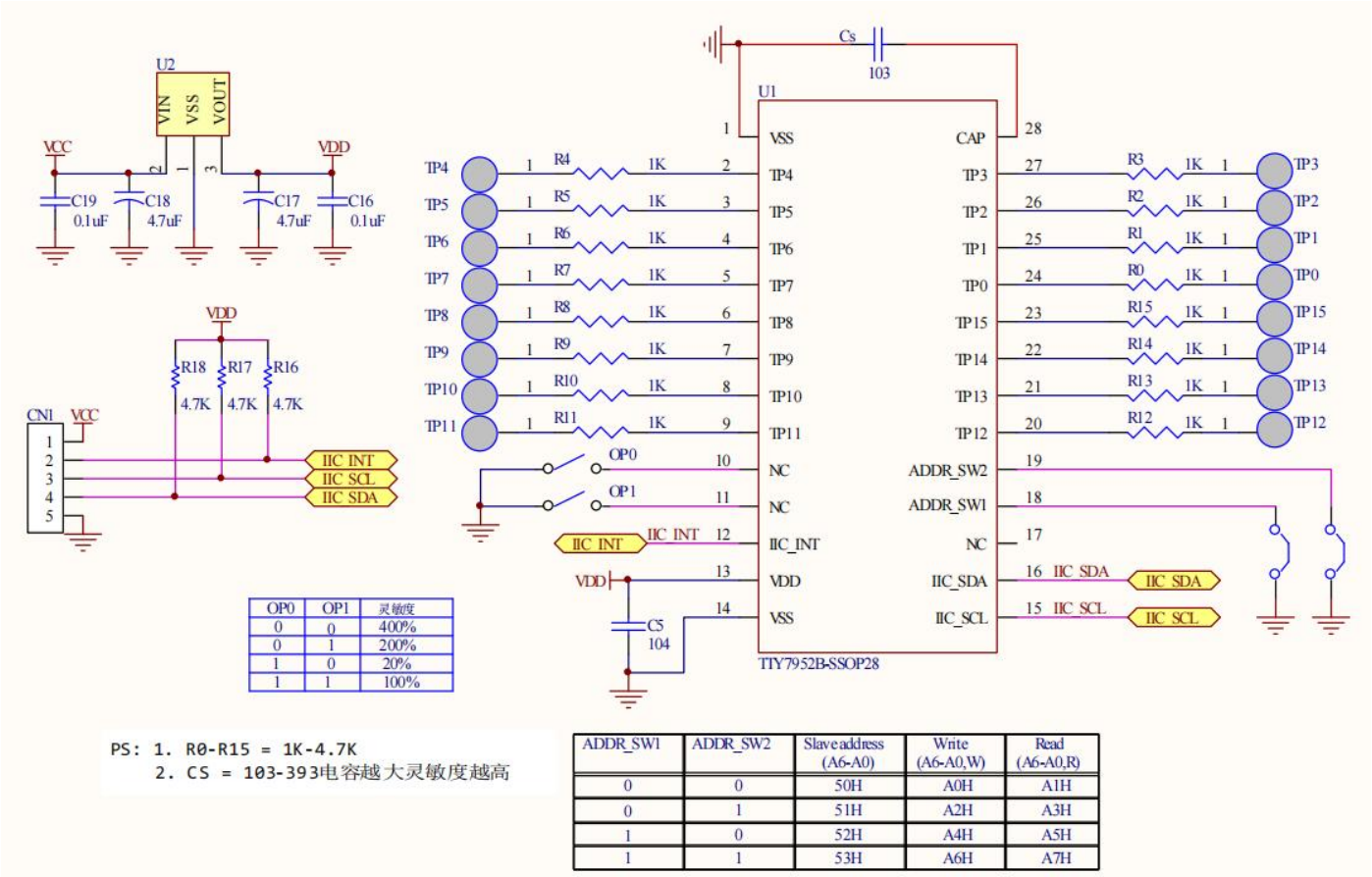
```
delay(1); //delay 1ms
setWrite_Buffer_3(0xC4, 0x10, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //TP4 Threshold
delay(1); //delay 1ms
setWrite_Buffer_3(0xC5, 0x10, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //TP5 Threshold
delay(1); //delay 1ms
setWrite_Buffer_3(0xC6, 0x10, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //TP6 Threshold
delay(1); //delay 1ms
setWrite_Buffer_3(0xC7, 0x10, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //TP7 Threshold
delay(1); //delay 1ms
setWrite_Buffer_3(0xC8, 0x10, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //TP8 Threshold
delay(1); //delay 1ms
setWrite_Buffer_3(0xC9, 0x10, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //TP9 Threshold
delay(1); //delay 1ms
setWrite_Buffer_3(0xCA, 0x10, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //TP10 Threshold
delay(1); //delay 1ms
setWrite_Buffer_3(0xCB, 0x10, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //TP11 Threshold
delay(1); //delay 1ms
setWrite_Buffer_3(0xD0, 0x02, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //Sleep Group 1 Threshold
delay(1); //delay 1ms
setWrite_Buffer_3(0xD1, 0x0F, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //Sleep Group 1
delay(1); //delay 1ms
setWrite_Buffer_3(0xD2, 0x02, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //Sleep Group 2 Threshold
delay(1); //delay 1ms
setWrite_Buffer_3(0xD3, 0xF0, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //Sleep Group 2
delay(1); //delay 1ms
setWrite_Buffer_3(0xD4, 0x02, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //Sleep Group 3 Threshold
```

```
    delay(1); //delay 1ms
    setWrite_Buffer_3(0xD5, 0x00, 0x0F);
    ACK = writeIIC(address_W, &Write_Buffer, 3); //Sleep Group 3
    delay(1); //delay 1ms
    setWrite_Buffer_3(0xD6, 0x02, 0x00);
    ACK = writeIIC(address_W, &Write_Buffer, 3); //Sleep Group 4 Threshold
    delay(1); //delay 1ms
    setWrite_Buffer_3(0xD7, 0x00, 0x00);
    ACK = writeIIC(address_W, &Write_Buffer, 3); //Sleep Group 4
    delay(50);

    while(1)
    {
        if(!SINT) /*等待读取请求，若关闭省电模式可以不用读取 SINT，但是每次读取按键建议间格 30ms*/
            ACK = readIIC(address_R, &Read_Buffer, 3); //读取按键状态
    }
}
```

# 应用参考线路:

## TTY7952B（SSOP28）

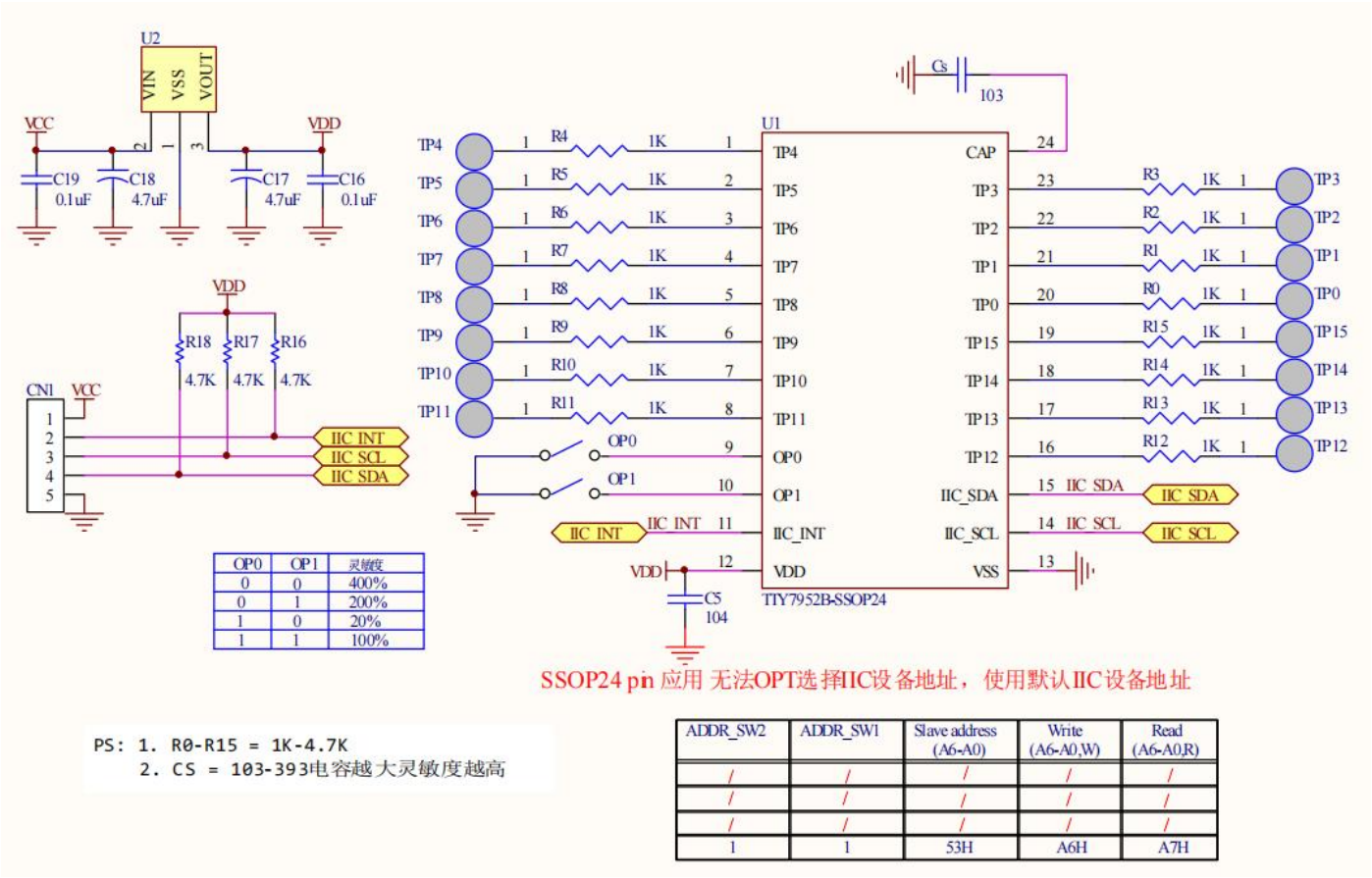



| OP0 | OP1 | 灵敏度 |
|---|---|---|
| 0 | 0 | 400% |
| 0 | 1 | 200% |
| 1 | 0 | 20% |
| 1 | 1 | 100% |

PS: 1. R0-R15 = 1K-4.7K
2. CS = 103-393电容越大灵敏度越高

| ADDR_SW1 | ADDR_SW2 | Slave address (A6-A0) | Write (A6-A0,W) | Read (A6-A0,R) |
|---|---|---|---|---|
| 0 | 0 | 50H | A0H | A1H |
| 0 | 1 | 51H | A2H | A3H |
| 1 | 0 | 52H | A4H | A5H |
| 1 | 1 | 53H | A6H | A7H |

# TTY7952B（SSOP24）



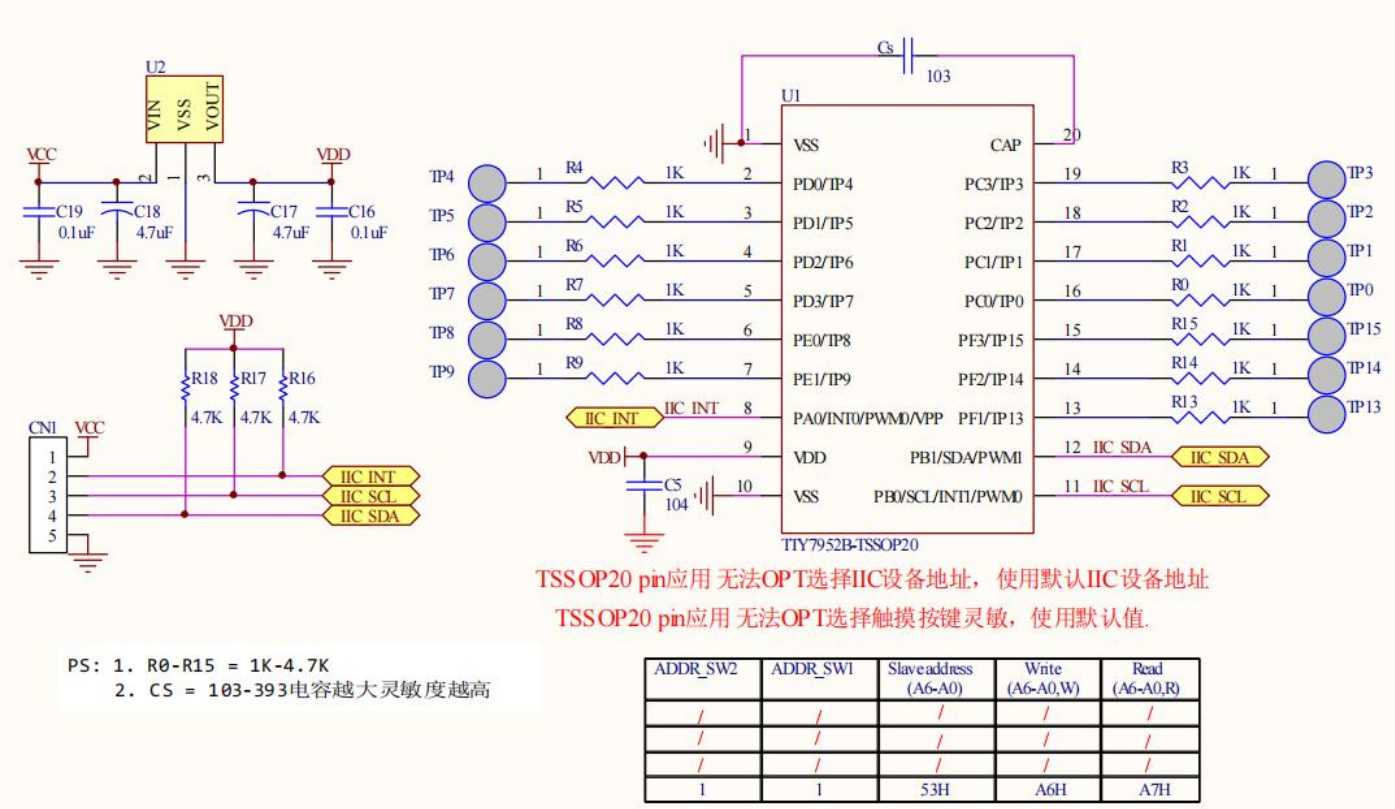

| OP0 | OP1 | 灵敏度 |
|---|---|---|
| 0 | 0 | 400% |
| 0 | 1 | 200% |
| 1 | 0 | 20% |
| 1 | 1 | 100% |

SSOP24 pin 应用 无法OPT选择IIC设备地址，使用默认IIC设备地址

PS: 1. R0-R15 = 1K-4.7K
2. CS = 103-393电容越大灵敏度越高

| ADDR_SW2 | ADDR_SW1 | Slave address (A6-A0) | Write (A6-A0,W) | Read (A6-A0,R) |
|---|---|---|---|---|
| / | / | / | / | / |
| / | / | / | / | / |
| / | / | / | / | / |
| 1 | 1 | 53H | A6H | A7H |

# TTY7952B（SSOP20）




| ADDR_SW2 | ADDR_SW1 | Slave address (A6-A0) | Write (A6-A0,W) | Read (A6-A0,R) |
|---|---|---|---|---|
| / | / | / | / | / |
| / | / | / | / | / |
| / | / | / | / | / |
| 1 | 1 | 53H | A6H | A7H |

触摸 PAD 到 IC pin 串 1Kohm-4Kohm 是可以增强抗手机和对讲机的干扰。

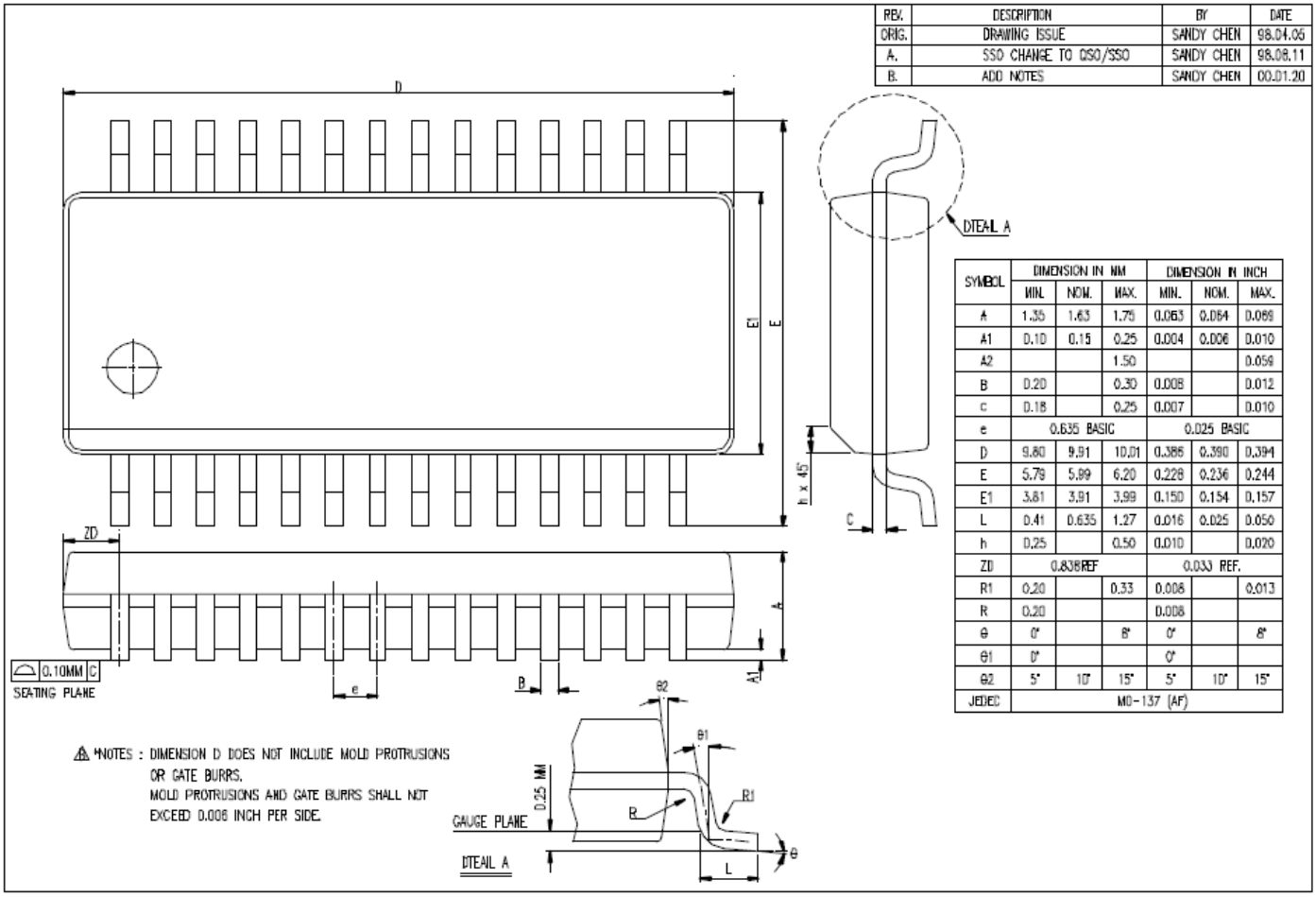
Cs 外接电容与压克力厚度关系：(预设阀值 010H)

以铁片弹簧键，圆型实心直径 12 MM 为例，压克力厚度与 CS 电容的关系如下:

| 压克力厚度(mm) | CS |
|---|---|
| 1 | 682 |
| 2 | 882 |
| 4 | 103 |
| 6 | 153 |
| 8 | 223 |
| 10 | 223 |
| | |

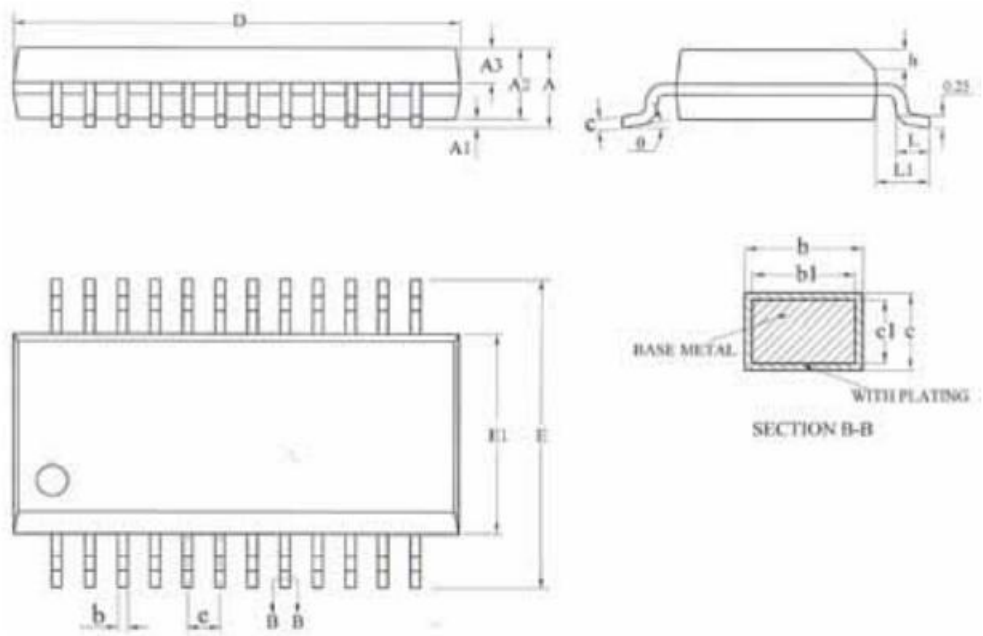
此表格仅供参考，不同的 PAD 大小，PCB layout 皆会影响。

# 封装说明：（28-SSOP）




| SYMBOL | DIMENSION IN MM | | | DIMENSION IN INCH | | |
|---|---|---|---|---|---|---|
| | MIN. | NOM. | MAX. | MIN. | NOM. | MAX. |
| A | 1.35 | 1.63 | 1.75 | 0.053 | 0.064 | 0.069 |
| A1 | 0.10 | 0.15 | 0.25 | 0.004 | 0.006 | 0.010 |
| A2 | | | 1.50 | | | 0.059 |
| B | 0.20 | | 0.30 | 0.008 | | 0.012 |
| c | 0.18 | | 0.25 | 0.007 | | 0.010 |
| e | 0.635 BASIC | | | 0.025 BASIC | | |
| D | 9.80 | 9.91 | 10.01 | 0.386 | 0.390 | 0.394 |
| E | 5.79 | 5.99 | 6.20 | 0.228 | 0.236 | 0.244 |
| E1 | 3.81 | 3.91 | 3.99 | 0.150 | 0.154 | 0.157 |
| L | 0.41 | 0.635 | 1.27 | 0.016 | 0.025 | 0.050 |
| h | 0.25 | | 0.50 | 0.010 | | 0.020 |
| ZD | 0.838REF | | | 0.033 REF. | | |
| R1 | 0.20 | | 0.33 | 0.008 | | 0.013 |
| R | 0.20 | | | 0.008 | | |
| θ | 0° | | 8° | 0° | | 8° |
| θ1 | 0° | | | 0° | | |
| θ2 | 5° | 10° | 15° | 5° | 10° | 15° |
| JEDEC | MO-137 (AF) | | | | | |

(SSOP24)

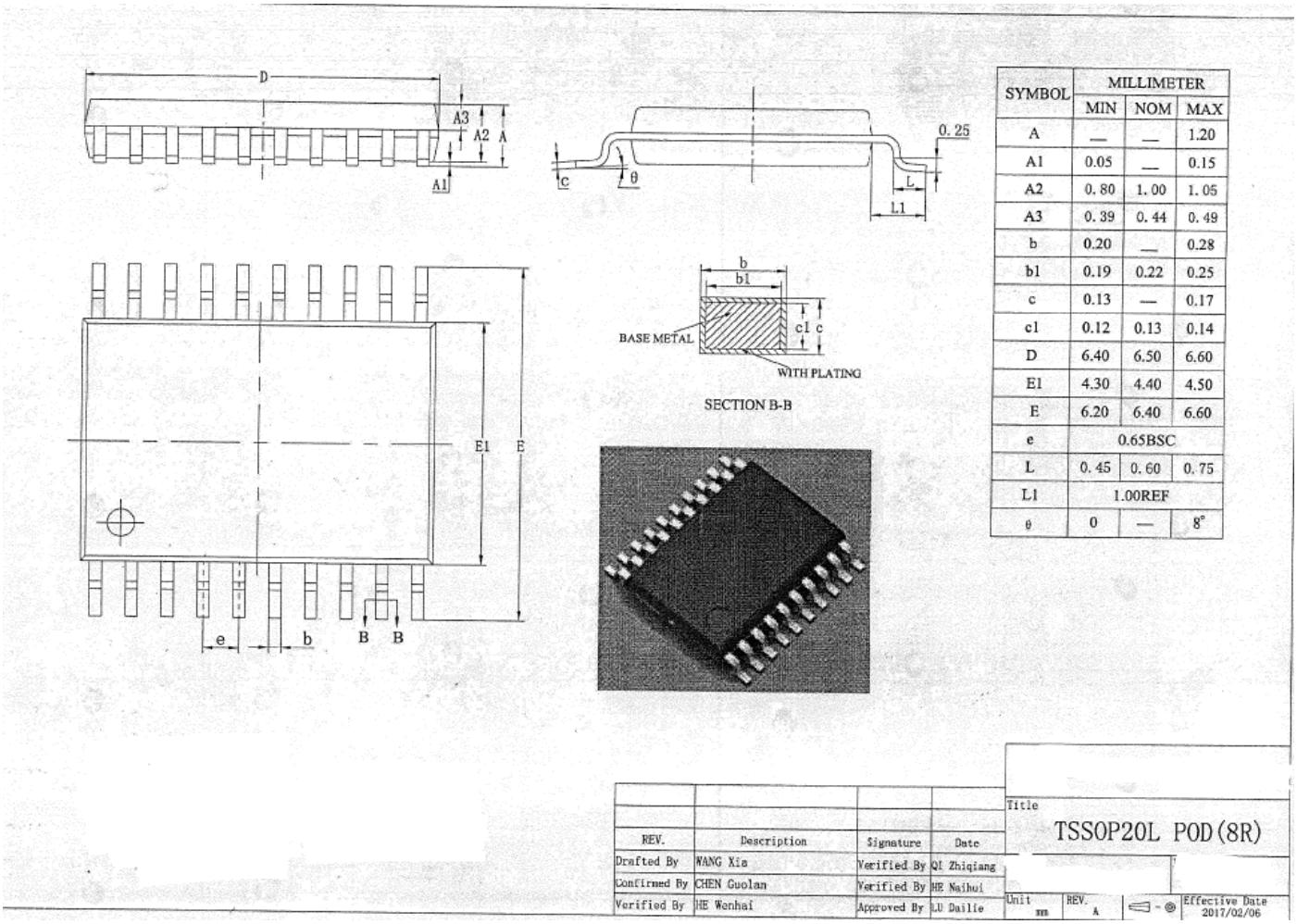



| Symbol Parameter (Unit : mm) | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | | | A1 | | | A2 | | | A3 | | | b | | |
| Min | Nom | Max | Min | Nom | Max | Min | Nom | Max | Min | Nom | Max | Min | Nom | Max |
| | | 1.75 | 0.10 | 0.15 | 0.25 | 1.30 | 1.40 | 1.50 | 0.60 | 0.65 | 0.70 | 0.23 | | 0.31 |

| Symbol Parameter (Unit : mm) | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| b1 | | | c | | | c1 | | | D | | | E | | |
| Min | Nom | Max | Min | Nom | Max | Min | Nom | Max | Min | Nom | Max | Min | Nom | Max |
| 0.22 | 0.25 | 0.28 | 0.20 | | 0.24 | 0.19 | 0.20 | 0.21 | 8.55 | 8.65 | 8.75 | 5.80 | 6.00 | 6.20 |

| Symbol Parameter (Unit : mm) | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E1 | | | e | | h | | | L | | | L1 | | θ | | |
| Min | Nom | Max | | Typ | | Min | Nom | Max | Min | Nom | Max | | Typ | | Min | Nom | Max |
| 3.80 | 3.90 | 4.00 | | 0.635 BSC | | 0.30 | | 0.50 | 0.50 | | 0.80 | | 1.05 REF | | 0 | | 8° |

（20-TSSOP）




| SYMBOL | MILLIMETER | | |
|---|---|---|---|
| | MIN | NOM | MAX |
| A | — | — | 1.20 |
| A1 | 0.05 | — | 0.15 |
| A2 | 0.80 | 1.00 | 1.05 |
| A3 | 0.39 | 0.44 | 0.49 |
| b | 0.20 | — | 0.28 |
| b1 | 0.19 | 0.22 | 0.25 |
| c | 0.13 | — | 0.17 |
| c1 | 0.12 | 0.13 | 0.14 |
| D | 6.40 | 6.50 | 6.60 |
| E1 | 4.30 | 4.40 | 4.50 |
| E | 6.20 | 6.40 | 6.60 |
| e | 0.65BSC | | |
| L | 0.45 | 0.60 | 0.75 |
| L1 | 1.00REF | | |
| θ | 0 | — | 8° |

| REV. | Description | Signature | Date |
|---|---|---|---|
| Drafted By | WANG Xia | Verified By | QI Zhiqiang |
| Confirmed By | CHEN Guolan | Verified By | HE Naihui |
| Verified By | HE Wenhai | Approved By | LU Dailie |

Title: TSSOP20L POD(8R)

Unit: mm | REV. A | Effective Date 2017/02/06

订购信息

TTY7952B

a. 封装型号 ：TTR051-ASFN（SSOP28）
b. 封装型号 ：TTR051-HSEN（SSOP24）
c. 封装型号 ：TTR051-DTDN（TSSOP20）

修改记录：

1. 2023/3/22 增加SSOP24pin封装线路图，包装尺寸图。
2. 2026/3/18 修正选项脚与表格数据不符