

TTY6760 (I2C 通讯) Slide Application

规格书 V1.3

目录

软件简介:.....	2
特色:.....	2
产品应用范围:	2
脚位定义:.....	3
电气特性:	3
1 最大绝对额定值.....	3
2 DC/AC 特性: (测试条件为室温=25℃)	4
功能描述:	4
IIC 协定:.....	5
特别说明:.....	7
5. 按键阈值调整的步骤:.....	8
建议参考线路图:	14
1. TTY6760 订购信息.....	17

软件简介:

此软件系提供给客户一个简易设定的滑条按键应用方案。客户只要使用 IIC 通讯格式，即可读取滑条按键及滑条地址的数据。

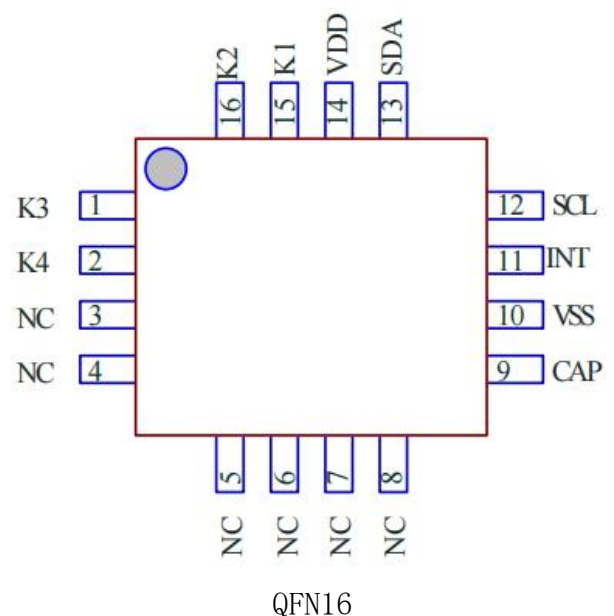
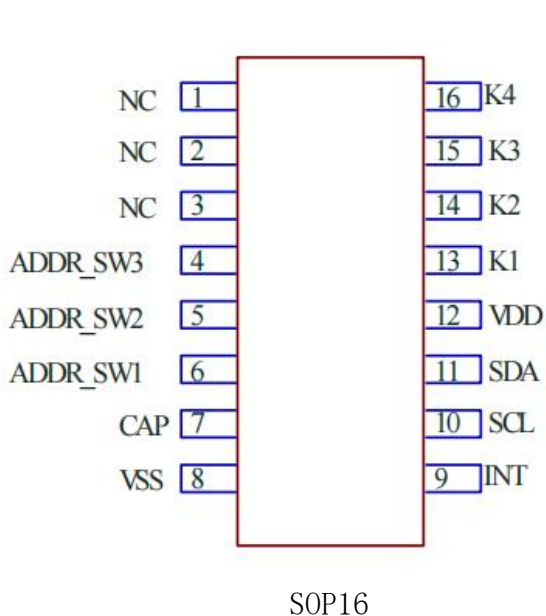
特色:

- 此应用使用 4 个 Touch Pad 组合为一组滑条按键。
- 设计有省电模式。

产品应用范围:

- 滑条触摸按键应用
- 门禁监控设备
- 消费类电子

● 封装脚位图与功能说明:



脚位定义:

SOP_16Pin	QFN_16Pin	Define	I/O	Pin Description
12	14	V _{DD}	Power	Positive power supply
8	10	V _{SS}	Power	Negative power supply, ground
7	9	CAP	I	Touch sensor input
9	11	IIC_INT	IO	IIC_INT pin, 应用时外部加上拉电阻。
10	12	IIC_SCL	IO	IIC_IIC clock pin, 应用时外部加上拉电阻。
11	13	IIC_SDA	IO	IIC_IIC data pin, 应用时外部加上拉电阻。
1	3	NC	/	无用脚
2	4	NC	/	无用脚
3	5	NC	/	无用脚
4	6	ADDR_SW3	IO/I	IIC slave address select
5	7	ADDR_SW2	IO/I	IIC slave address select
6	8	ADDR_SW1	IO/I	IIC slave address select
16	2	K4	IO/I	touch pad input
15	1	K3	IO/I	touch pad input
15	16	K2	IO/I	touch pad input
13	15	K1	IO/I	touch pad input

Table1. TTY6760 pin description

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	T _{OP}	- — —	-40~+85	℃
存放温度	T _{STG}	— — —	-50~+125	℃
电源电压	V _{DD}	Ta=25℃	V _{SS} -0.3~V _{SS} +5.5	V
输入电压	V _{IN}	Ta=25℃	V _{SS} -0.3~V _{DD} +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		3.5	—	5.5	V
系统震荡频率	F	VDD=5V	—	8M	—	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。

● 飞梭(滑条)介绍：

飞梭(滑条)按键的原理是利用在 PCB LAYOUT 上测得触摸的按压深度，来解析按压位置的一种方法。优点在可利用最少的按键解析出最多的按键地址。滑条图形主要分为环型跟直条两种应用，如下图：

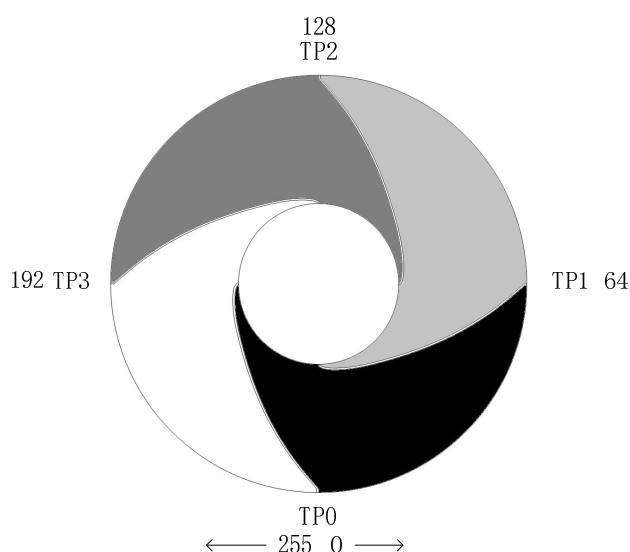


Figure1. 环形设计

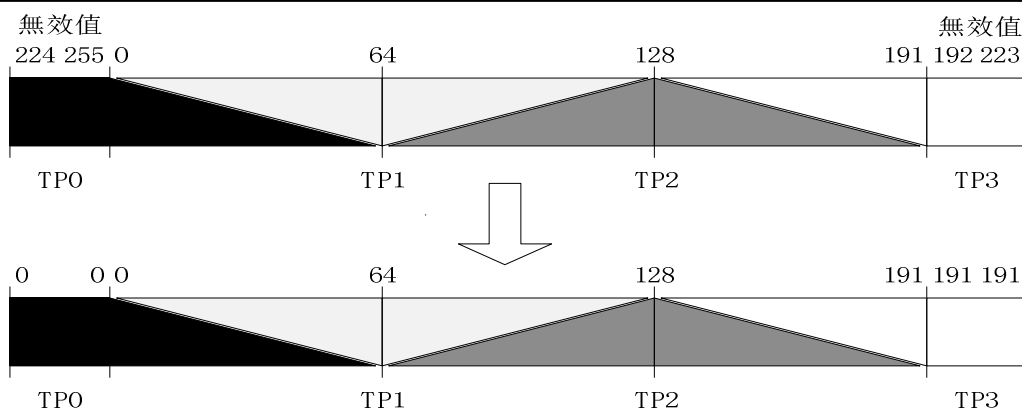
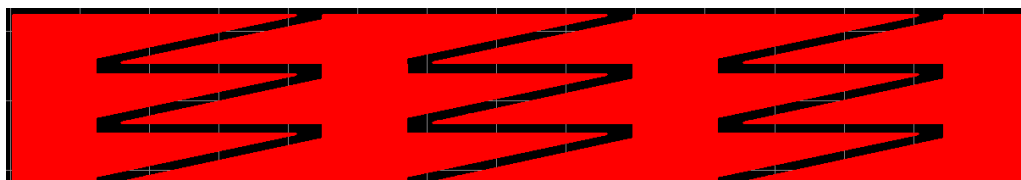


Figure2. 直条设计

注:直条设计中因为首尾不相连(非环形),所以有效地址为 0~191,若读取到值 192~223 或是 224~255 则需要在主控端分别判断为是 191 或是 0。

Figure1. 环形设计



其原理是利用按压 Touch Pad 时取得的数值变化,再使用内差法来计算其相对地址。因此需要最少 3 个按键,用以取得按压最深的按键与左右两边按键的差值来进行运算。

设计上建议按键与按键中心距离需小于 30mm。齿与齿间的距离则约为 0.4mm(如下图),一般以 3~4 齿的设计为佳。

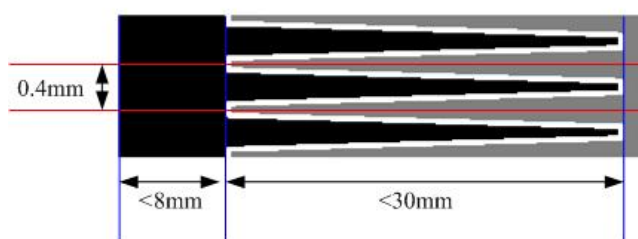


Figure3. Layout 设计要点

Table2. Wheel pad and key pad define

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。

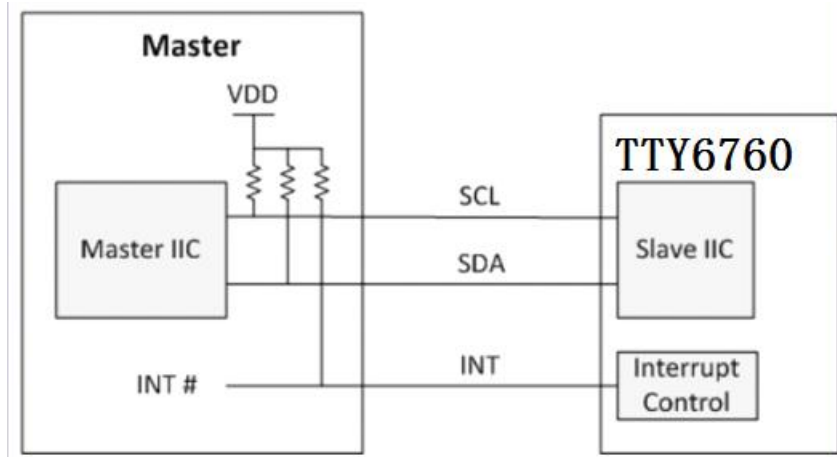


Figure6. IIC connect for master and TTY6760

INT 在无按键状态变化时为 High，当有按键时，INT 脚位会拉 Low，无按键为 High。

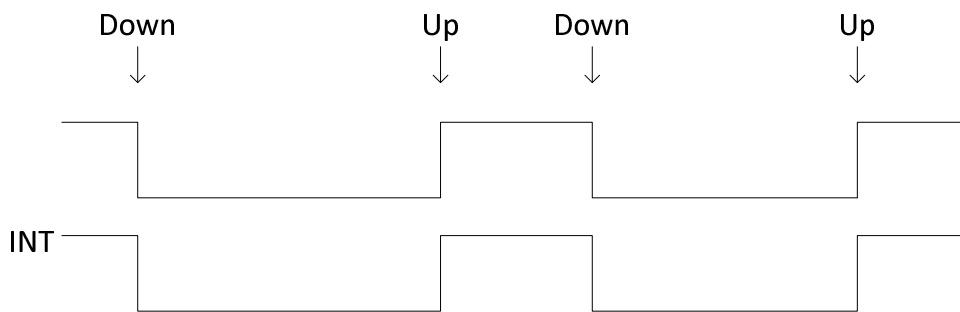
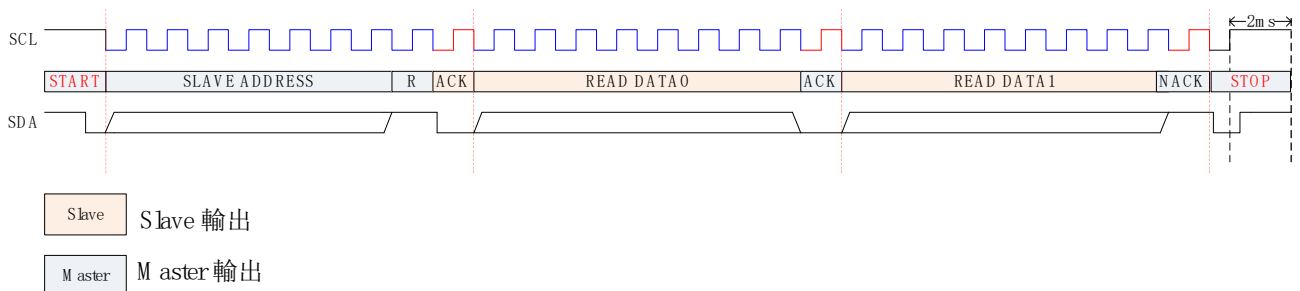


Figure7. INT pin describe

Package Stream



Slave Address

可透过 ADDR_SW1, ADDR_SW2, ADDR_SW3 脚位做切换，如下表

ADDR_SW1	ADDR_SW2	ADDR_SW3	Slave address (50-57)	Read (A6-A0,R)
0	0	0	50H	A1H
0	0	1	51H	A3H
0	1	0	52H	A5H

0	1	1	53H	A7H
1	0	0	54H	A9H
1	0	1	55H	ABH
1	1	0	56H	ADH
1	1	1	57H	AFH

Package Data

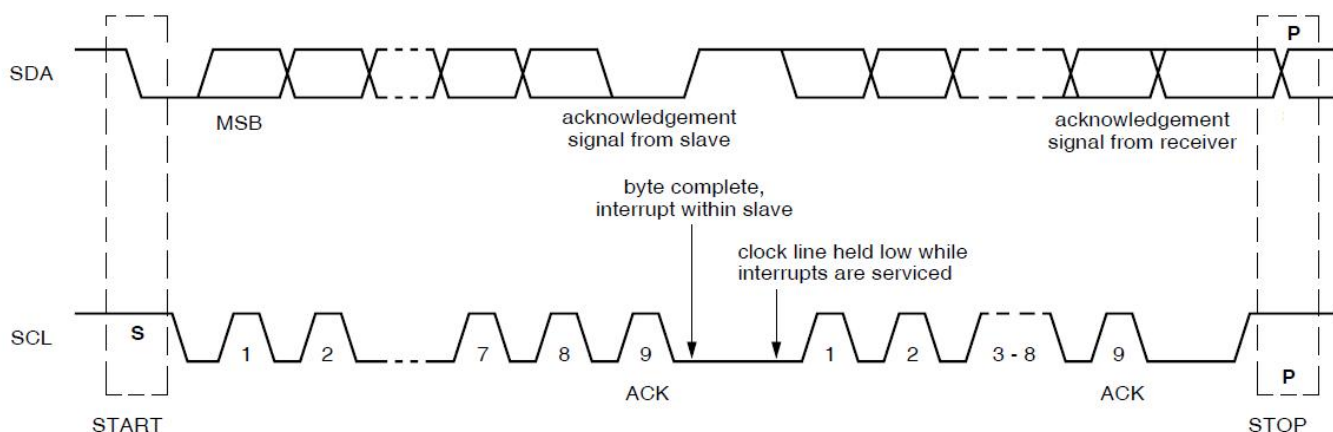
Read byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	C	0	0	WT	TP4	TP3	TP2	TP1
1	Wheel Position(0~255)							

C	Calibrate
0	Calibrating
1	Calibrate Finish (Define)

WT	Wheel Touch
0	No touch
1	Touch

特别说明:

- TTY6760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. 按键阈值调整的步骤：

Step1. 选择初始测试用的 CS 电容(见建议线路图)：

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，需要调高阈值，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，需要降低阈值。

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够快，则进行下一步。

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 TTY6760 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 TTY6760 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: 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 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();
```

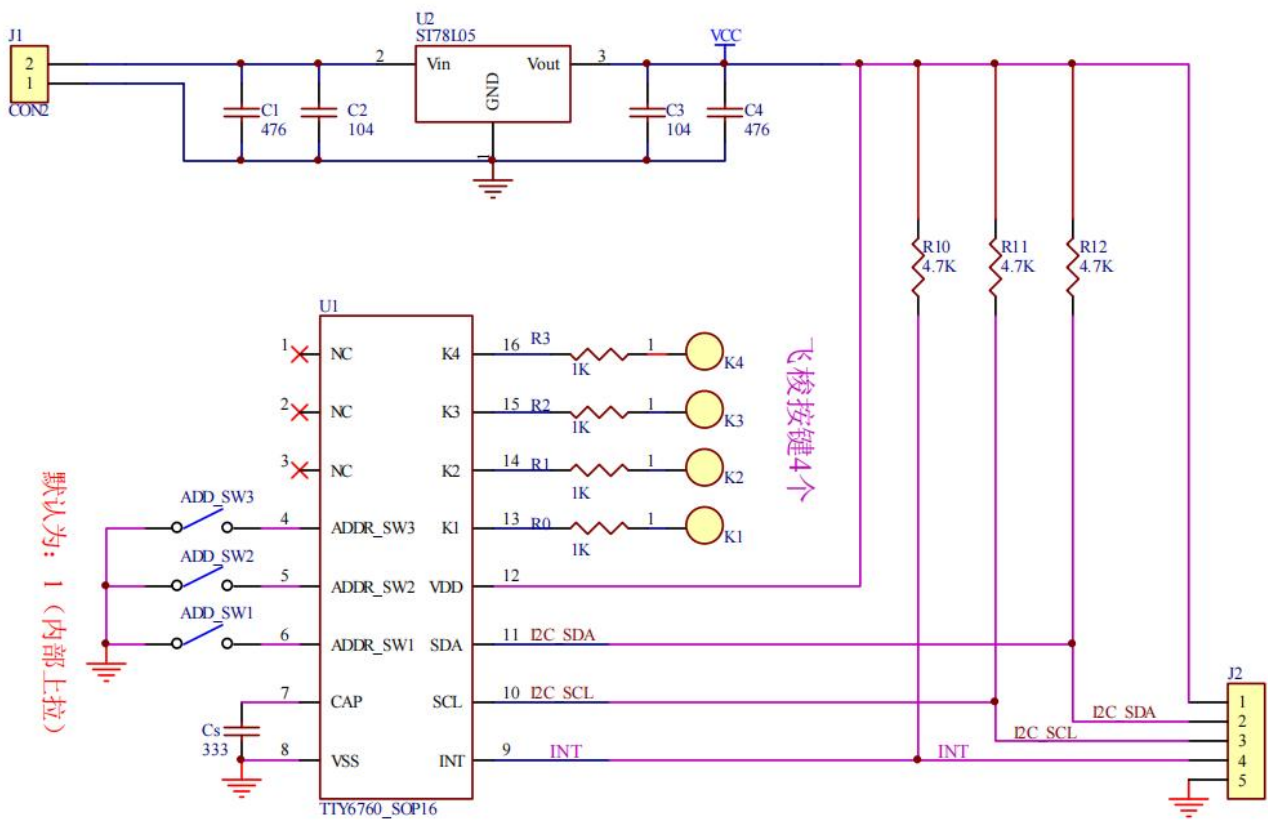
```
        if(i<length-1)
```

```
            sendACK();
```

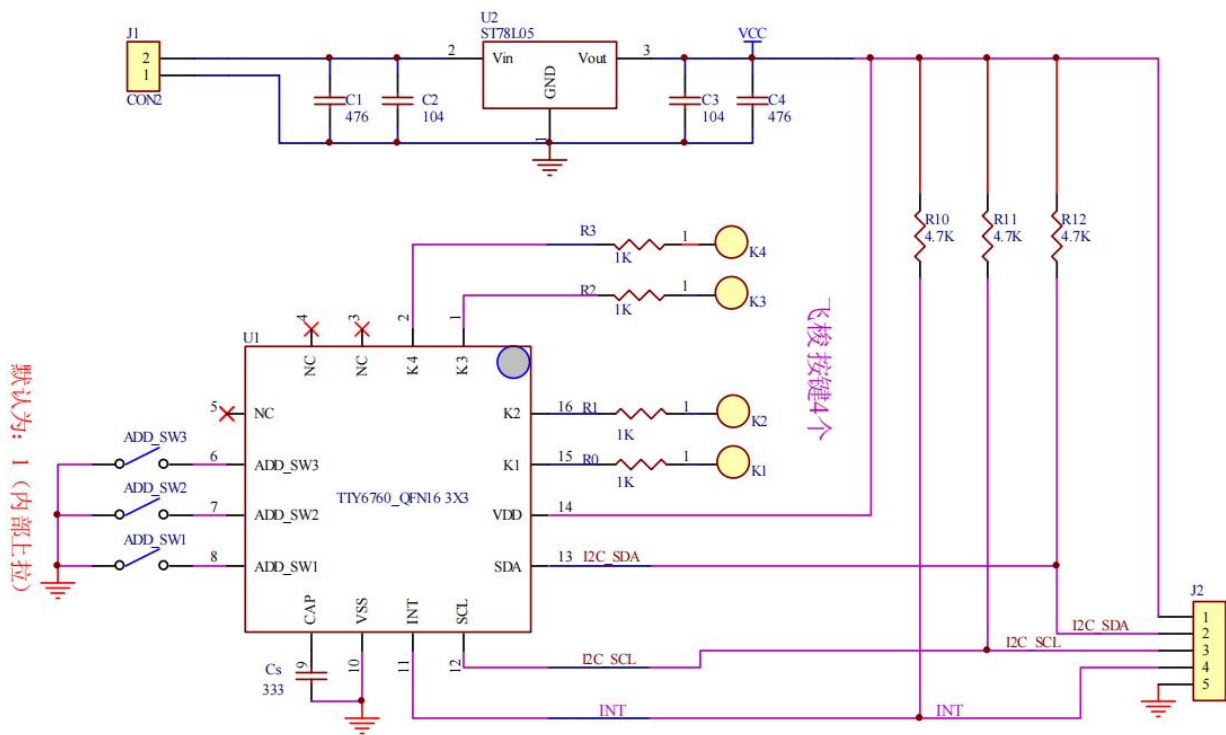
```
        else
            sendNOACK(); //读取最后一笔资料，送出 No ACK
    }
    sendStop(); //资料读取完成，送出停止信号
    return ACK;
}

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隔 30ms*/
        ACK = readIIC(address_R, &Read_Buffer, 3); //读取按键状态
    }
}
```

SOP16 建议线路:



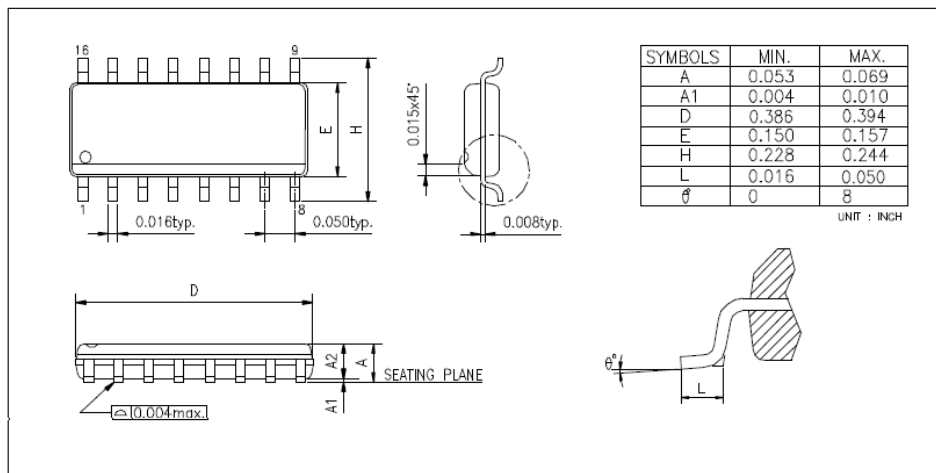
QFN16 (3*3mm) 建议线路:



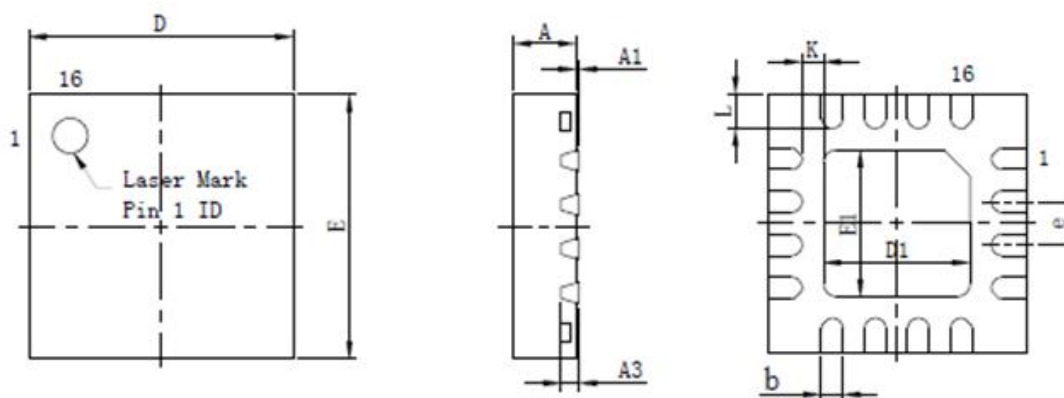
R0 - R3 接 1K-4.7Kohm 可增强 RF 干扰!

● 封装尺寸图:

(16_SOP)



(16/QFN 3*3mm)



Symbol Parameter (Unit : mm)													
A			A1			A3		b			D		
Min	Nom	Max	Min	Nom	Max		Typ	Min	Nom	Max	Min	Nom	Max
0.70	0.75	0.80	0.00		0.05		0.203 REF	0.20	0.25	0.30	2.90	3.00	3.10

Symbol Parameter (Unit : mm)												
E			D1			E1			e		K	
Min	Nom	Max	Min	Nom	Max	Min	Nom	Max		Typ	Min	
2.90	3.00	3.10	1.60	1.70	1.80	1.60	1.70	1.80		0.50	0.20	

Symbol Parameter (Unit : mm)			
L			
Min	Nom	Max	
0.30	0.40	0.50	

1. TTY6760 订购信息

修订记录

1. 2024/3/22 – Version: 1.0
2. 2024/3/25 – Version: 1.1
3. 2024/5/25 – Version: 1.2
4. 2025/2/10 – Version: 1.3

增加QFN16pin封装