传感器信号检测及数字处理接口电路PS203

器件概述

PS203传感器信号检测及数字处理接口电路是专门为低噪声、微信号处理的数字接口电路。该电路集成了高灵敏度、高阻抗开关电容放大器作为传感器的微信号输入，同时内置14位分辨率的模数转换及一个数字式高性能二阶巴特沃斯低通滤波器，与PS202相比，特别添加一个高精度温度检测单元，主要用于传感器环境温度数据处理使用，直接向微处理器提供单线输出的脉冲接口。另外该电路内置稳压电源及内部时钟，能在同一时间内采集多个传感器信号。

功能部件

数字信号处理 (DSP)
可双传感器输入
单线串行接口 (DOCI)
0．05K的温度检测单元
低电压工作
低功耗
电源抑制比高
甚高输入阻抗，>10G
输入动态范围广
直接连接到微程序控制器(MCU)

应用领域

低噪声微信号数字处理
通用传感器接口
工业现场数据采集
红外线传感器
多路数字探测器
数字安防设备
USB传感器
局域网监控器
私人警报器
超级汽车防盗系统

引脚排列

插头编号.	名称	说明
1	V_{ss}	电源负极
2	IN1	信号输入1
3	IN2	信号输入2
4	Vdd	电源正极
5	TEST	保留的测试方式，正常时连接到Vss
6	DOCI	串行数字输出
7	N	悬空
8	N	悬空

最大绝对额定值

参量	符号	最小值	最大值	单位	备注
电源电压	V_{DD}	−0.3	3.6	V	
引脚电流极限		−100	100	mA	单针次
操作温度	T_{ST}	−20	70	°C	
存储温度	T_{ST}	−40	125	°C	

超过上述表中数值可能会导致对装置的永久性损坏。长期使用在最大绝对额定值可能会影响器件的可靠性。

工作条件 (T=25°C, Vdd=5V)

参量	符号	最小值	典型值	最大值	单位	备注
电源电压	V_{DD}	2．7	3	5.5	V	直流电
工作电流	I_{DD}		10	15	μA	V_{DD}=3.3V
数据输出接口	DOCI					
输出低电平	V_{IL}			20	%V_{DD}	
输出高电平	V_{IH}	80			%V_{DD}	
灌电流			200		μA	输入 V_{SS} / V_{DD}
输入电容	C_I		5		pF	
数据建立时间*	t_s	2				1/系统频率周期
低电平数据维持时间*	t_l	200			ns	
高电平数据维持时间	t_H	200			ns	
数据位稳定时间	t_{bit}	1			μs	CLOAD = 10pF
串行接口更新时间	t_{rep}		256		°	1/FCLK
模拟输入1/输入2,	IN1，IN2					
输入偏置电流 *	V_{IN} = -10mV .. +10mV		−1		1	fA
电压输入范围	单端模式/差分模式		−50 −100		50 100	mV mV
数字转换器	ADC					
模数转换器的分辨率				14	Bits	最大值 = 2^{14}
模数转换器的灵敏度		6	6.5	7	μV/count	
模数转换器的温度系数	Tc	-300		300	ppm/K	
输入噪声（有效值）	@ 0.5Hz @ 1Hz @ 2Hz @ 5Hz		2.5 1.5 0.5 0.4		μV μV μV μV	
模数转换器的残留误差		7000	8192	9200		数值
振荡器和滤波器						
低通滤波器截止频率*			10		Hz	

模拟量到数字量转换时间	T_{REP}		256		$1/F_{CLK}$	
内部时钟频率	F_{CLK}	58	64	70	kHz	
温度检测						
增益			80			-20~+90° C
测量范围		-20		+90	° C	
线性度		-5		+5	%	-20~+90° C
环境记数值		5700	6700	7700		@25° C

*)由设计担保，不在产品之内测试.

**)在评价或赋予资格的时候检测及表示其特性，不在产品之内测试.

除非另有专项说明，否则所有的电压参照工作条件.

除非另有专项说明，否则温度定为25°C.

除非另有专项说明，否则参数在运行条件以内取值才能得到保证.

功能说明

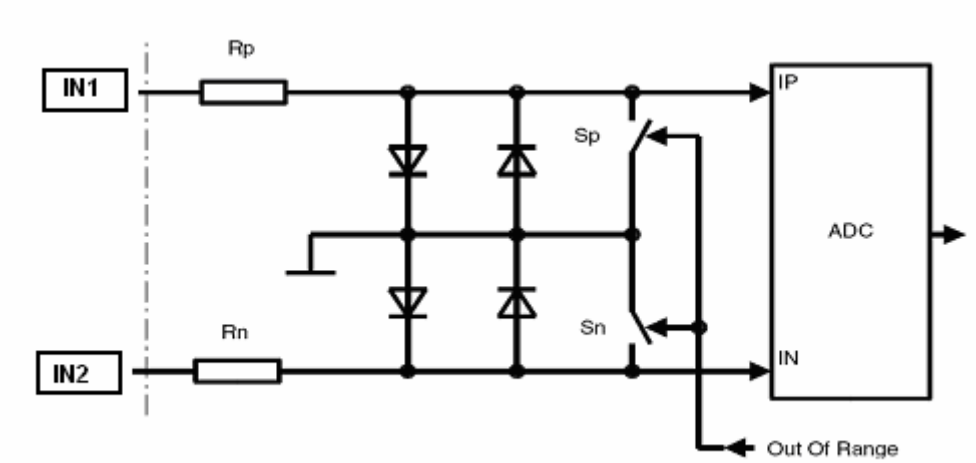

Fig .1 模数转换器输入级

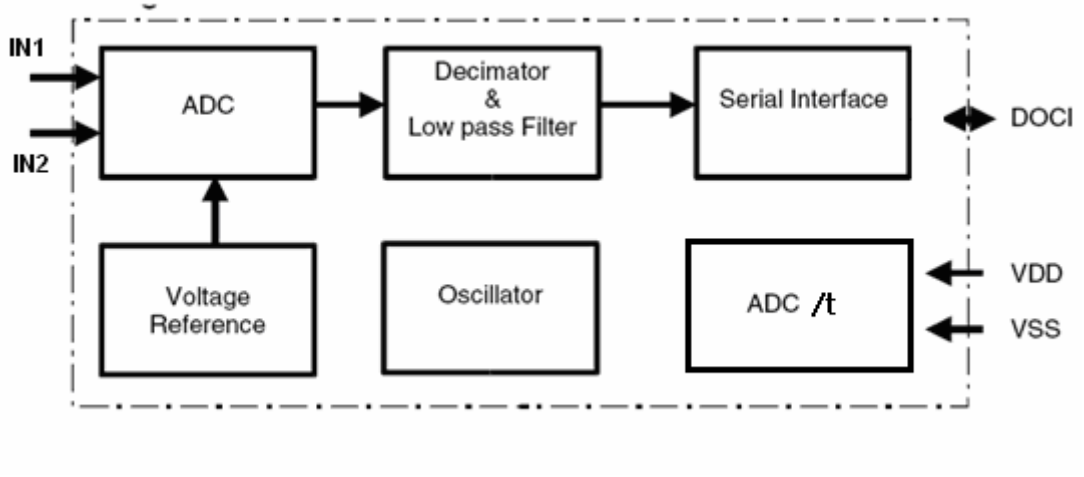

Fig. 2 PS203 内部功能

1,振荡器

这个集成电路包含一个低功耗振荡器器，频率是64 kHz。所有有关的时间信号及数字滤波器的中止频率都是和振荡器的频率紧密相关的。

2,模数转换器

14 Bit模数转换器(ADC) 会产生一个来自于传感器敏感元模拟信号转换成数字电平。

3,窄带和低通滤波器

一个数字式二阶巴特沃斯低通滤波器。模数转换器的范围是从0到16385。

4,温度测量

芯片上的温度测量ADC/t值是通过内置温度传感器电压转化而来，其解析度大于0.1K的基准数字。温度ADC/t为14位数据。

5, 串行接口及编程帮助

- PS203 每隔 512 个系统时钟（14ms）产生一个中断, PS203 滤波器数值产生后会把 DOCI 拉高。
- PS203 数据锁存器在不被 MCU 读出的时候，数据将从滤波器传输到数据锁存器。注意如果 MCU 读入寄存器的速度比滤波器的更新速度快，那么数据将显示全部为 0。
- 如果用 MCU 中断方式读取数据，MCU 将 DOCI 拉高, 高电平持续时间 75us,后读取 14 位数据。最后一个瞬间数据被读完以后，MCU 必须将 DOCI 强行拉低，释放 DOCI。当 DOCI 被 MCU 强行拉高的时候，PS203 锁存器数据将不再更新。如果读取包括温度的数据，应读取 28 个位数据，即 MCU 先读出传感器一个 14 位数据，再读取一个 14 位温度数据。
- 如果 DOCI 处于低电平时，MCU 读取过程超过一个系统时钟，数据锁存器将被更新。
- A：在单个 Sensor 模式下：

1、 发出转换命令状态:主控器发出一个宽度不小于 2us 的低电平，然后释放总线转换过程
2、 接受到转换命令后，即开始 AD 转换过程，周期 14ms,这期间总线保持低电平。
3、 转换完成：内部 AD 转换完成后会将总线拉高，告诉 MCU 转换已经结束
4、 等待数据稳定：主控发现总线升高后，延时不低于 1us。这期间器件完成数字滤波及将数据推移至唯一寄存器
5、 启动接收：主控器发出一个低到高的上跳沿脉冲（L/H 脉冲宽度均不低于 200ns）,器件收到此时钟后将一位数据送上总线。主控器在规定时间内（Tbit）完成数据采样
6、 重复接收状态：反复动作 5,共 14 次
7、 回到状态 1 开始下一转换过程

B：MCU 定时中断读取多传感器模式：

1、 定时中断：MCU 内部定时 3—12ms（要比所有 Sensor 中产生中断的时间要快）的中断，用于读取传感器的数据。
2、 拉高总线：中断产生后，MCU 将 DOCI 强制拉高
3、 延时：DOCI 高电平持续时间不低于 75us
4、 启动接受：开始读取 14 位 AD 结果（读取过程与单 Sensor 相同）
5、 读完之后需拉低并且释放 DOCI 总线

MCU 定时中断用于多传感器，且定时间隔必须小于产生中断时间间隔最小的探测器。

6,越界检测

ADC 的动态范围是不确定的。+/-50Mv。为了避免饱和，该芯片包括一个超范围的检测逻辑电路，它可以发现高于 15872（97%的范围）和低于 511（3%的范围）的数值。如果数值在这个范围以外，开关 Sp/Sn,存储器和滤波网络将会关闭持续时间为 512 系统时间。这样才能确保在干扰后能够快速设定。

目前的 ADC 的输入阻抗几乎是无穷大的。

6, 数据输出时钟输入协议

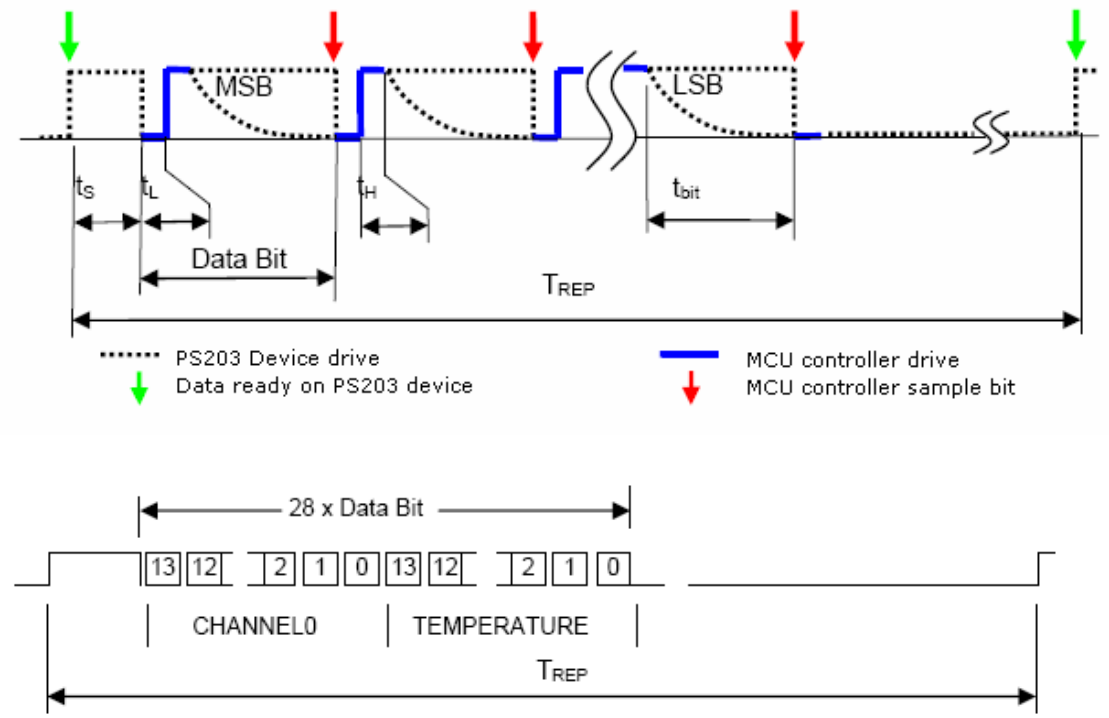

Fig. 3 PS203 DOCI 协议

典型应用

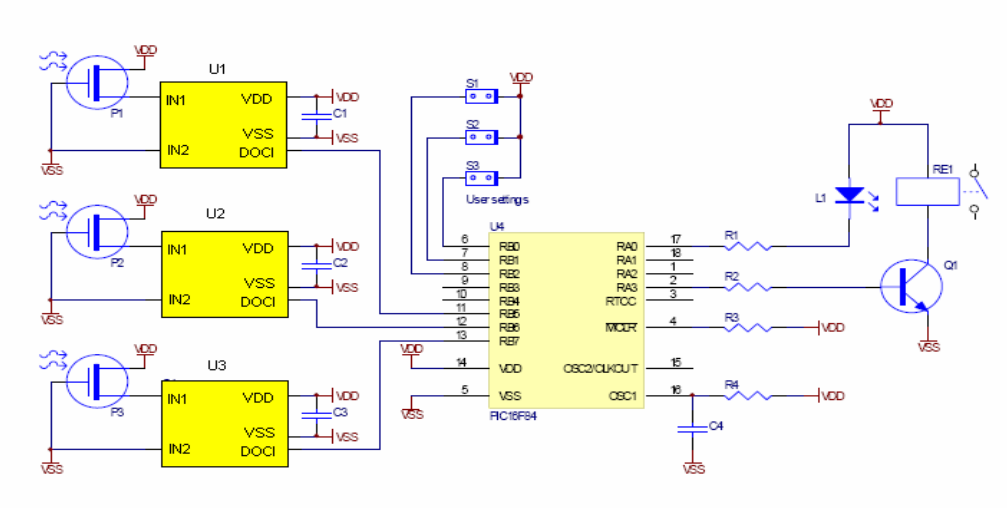

Fig. 4 PS203 直接连接MCU典型电路

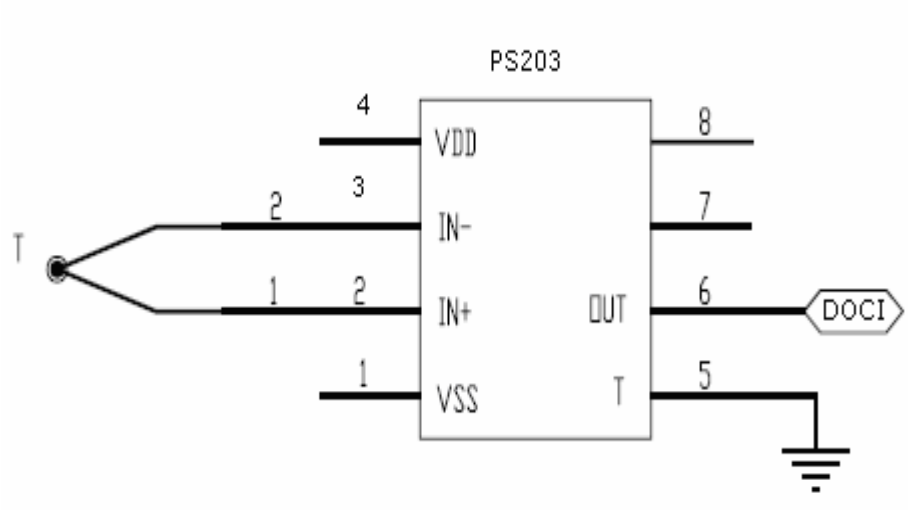

Fig. 5 PS203 直接连接热电偶（堆）典型电路

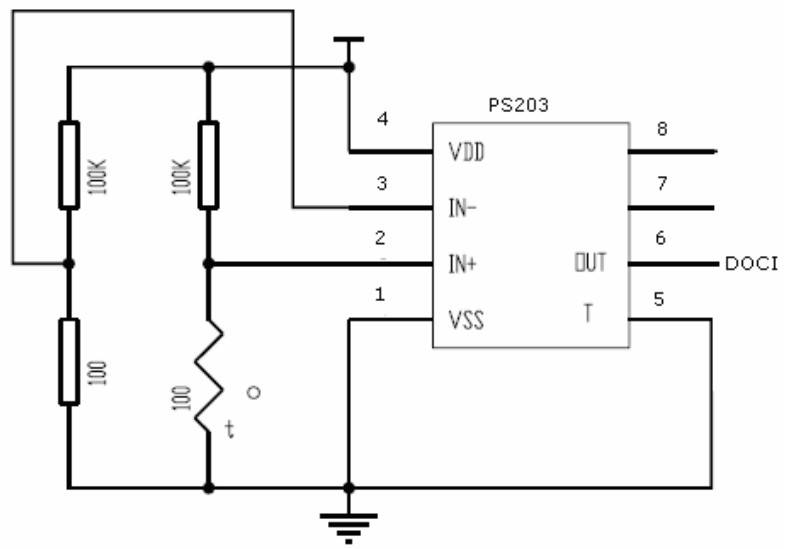

Fig. 6 PS203 直接连接热敏电阻典型电路

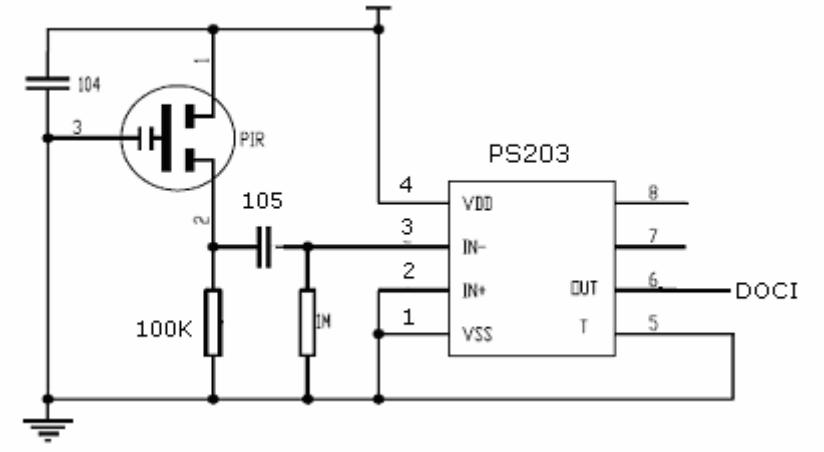

Fig. 7 PS203 直接连接普通PIR热释电传感器典型电路

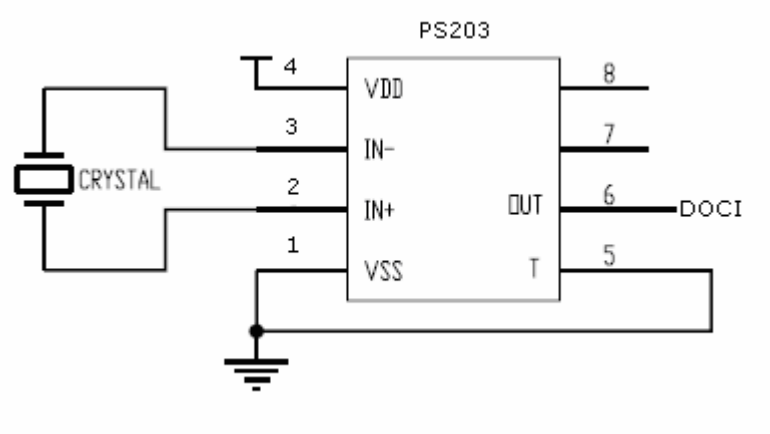

Fig. 8 PS203 直接连接压电晶体材料典型电路

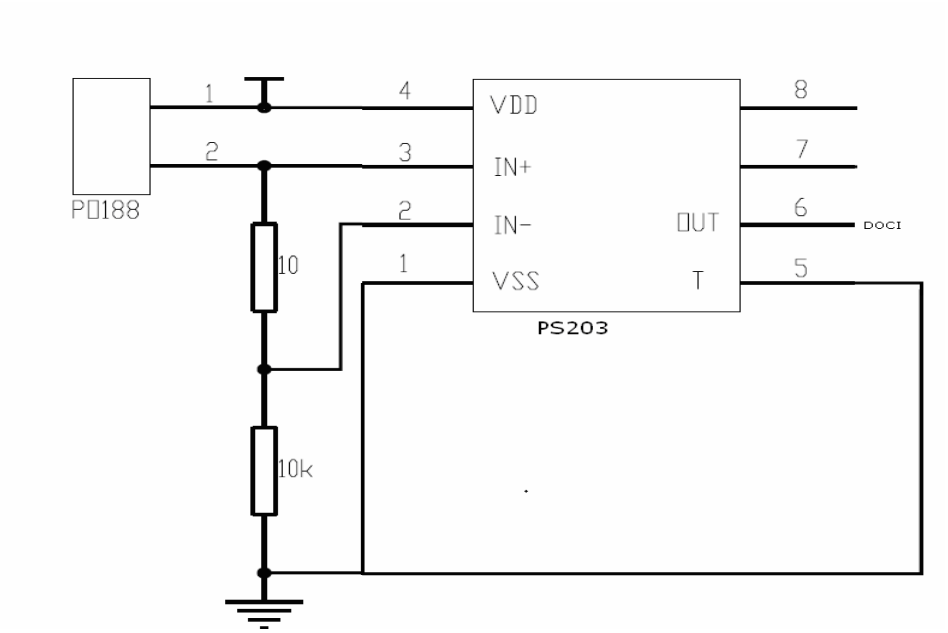

Fig. 9 PS203 直接连接Po188光电传感器典型电路

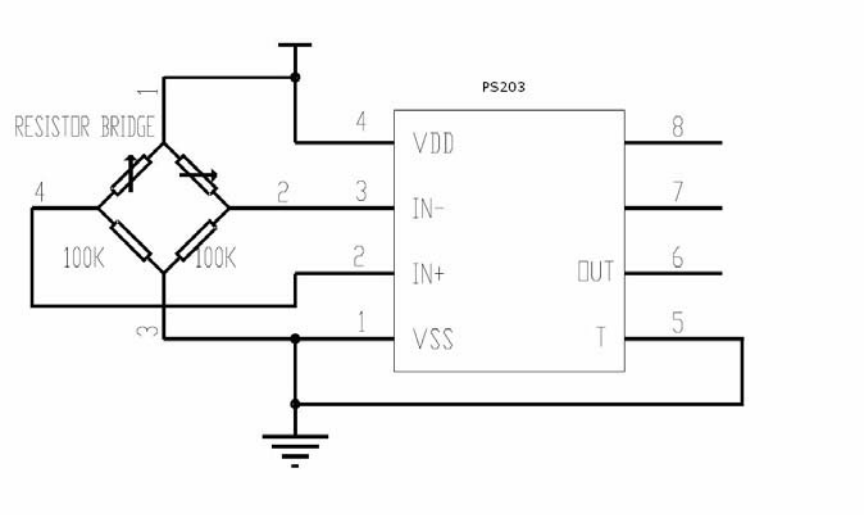

Fig.10 PS203 直接连接压力传感器典型电路

关于PS203编程的核心提示

PS203 与主控制器之间使用 DOCI 数据总线进行数据交换

DOCI 的意思是：在一根线上，实现 PS203 的数据（Data）输出（Out），和主控制器的时钟（Clock）输入（Input）。

总线的工作是在主控制器的控制下进行，工作状态是：

1， 待机状态：主控制器（H）释放总线，PS203 (P) 输出口处于等待时钟状态，总线空闲，示波器测试为高电平。

2， 启动接收：H 发出一个 低——高的上跳沿时钟脉冲，告诉 P 准备接收数据。经过 Th 时间（约 200 纳秒）延迟后，P 将数据送上总线，（0 或者 1）。

3， 接收一位：H 在规定时间内（Tbit）进行数据采样（读取一位数据）。

4， 重复接收：反复进行 2、3 动作共 28 次，接收 28 位 2 进制数。

5， 释放总线：完成后 H 释放总线，回到待机状态。

说明：有效数值范围：0-16383，就是说 2 的 14 次幂。

核心编程：

数据采样子程序

```
        Data = 0;                                   //清数据区
        configure Pin as input;                     //设定主机的接收口线 Pin
    为输入
LOOP：  if ( DL! = High) goto LOOP                  //不为高，反复测试,等待
PS203ADC转换完成
        wait > 25μs                                 //为高，等待数据稳定
        for (j=0; j<28; j++)                        //采样28次，读取14位ADC
    转换结果
            {
            configure Pin as output;                //设置 Pin 为输出，准备发
        送时钟
            set DL low for >200 ns;                 //时钟上跳沿的低电平时间
        为200纳秒
            set DL high for >200 ns;                //时钟上跳沿的高电平时间
        为200纳秒

            configure Pin as input;                 //设置 Pin 为输入，准备读
        取数据
            Data <<=1; // shift Bits left           //数据区左移一位，给当前
        位留位置
            if (DL == high) Data++; // read 1       //如果是高电平，数据是
        ‘1’，则加1，否则不加（默认是0）
            }                                       //28位采样完成

        configure Pin as output;                    //设置 Pin 为输出，准备发
    送时钟
        set Pin low for >200ns;                     //延迟200纳秒
        configure Pin as input;                     //释放总线
        (next value in 7,3 ms)                      //下一个 ADC 转换周期间
```

隔时间，不小于 7。3 毫秒，否则转换内有完成，读出为 0

return ();

用于单只/四只红外传感器和温度采样的编程参考：

Read One Detector PIR and Temperature

```
//=====================================================================
void clear_DOCI_interrupt(void)
{
//=====================================================================
// This leaves the bus floating that the sensor can generate an interrupt
//=====================================================================
//Release the DOCI Line leave in zero state
#use fast_io(B)
output_low(DOCI_PIN); // Clear DOCI pin
doci_dir&=~DOCI_PIN_DIR; // Clear direction bit
set_tris_b(doci_dir); // DOCI pin output
delay_cycles(75); //time depends on track capacitance
doci_dir|=DOCI_PIN_DIR; // Set direction bit
set_tris_b(doci_dir); // DOCI pin output
#use standard_io(B)
}
//=====================================================================
void read_digi_pir(void)
{
//=====================================================================
// Read filter from On
// DOCI is a IO pin
// DOCI_BID is a bit pattern that set or clear the direction
// of the IO pad on the processor
//=====================================================================
int n;
#use fast_io(B)
pir_data=0;
for(n=0;n<14;n++) // There are 14 bits of data from the sensors
{
output_low(DOCI_PIN); // Clear DOCI pin
doci_dir&=~DOCI_PIN_DIR; // Clear direction bit
set_tris_b(doci_dir); // DOCI pin output
delay_cycles(2); // Wait a bit
output_high(DOCI_PIN); // Set DOCI pin
doci_dir|=DOCI_PIN_DIR; // Set direction bit
set_tris_b(doci_dir); // DOCI pin input
delay_cycles(2); // Wait a bit
pir_data<<=1; // Make room
```

```
if(input(DOCI_PIN)==1)
pir_data++;
}
temp_data=0;
for(n=0;n<14;n++) // There are 14 bits of data from the sensors
}
output_low(DOCI_PIN); // Clear DOCI pin
doci_dir&=~DOCI_PIN_DIR; // Clear direction bit
set_tris_b(doci_dir); // DOCI pin output
delay_cycles(2); // Wait a bit
output_high(DOCI_PIN); // Set DOCI pin
doci_dir|=DOCI_PIN_DIR; // Set direction bit
set_tris_b(doci_dir); // DOCIwqw pin input
delay_cycles(2); // Wait a bit
temp_data<<=1; // Make room
if(input(DOCI_PIN)==1)
temp_data++;
}
clear_DOCI_interrupt();
#use standard_io(B)
}
```

Read four Detectors simultaneously

```
//=============================================================================
void clear_all_DOCI_interrupt(void)
{
//Drive a Zero onto the DOCI pins
#use fast_io(B)
output_b(CLEAR_ALL_DOCI); // Clear all DOCI pins
doci_dir&=~DOCI_PINS_DIR; // Clear direction bit
set_tris_b(doci_dir); // DOCI pin output
delay_cycles(175); //time depends on track capacitance
doci_dir|=DOCI_PINS_DIR; // Set direction bit
set_tris_b(doci_dir); // DOCI pin output
#use standard_io(B)
}
//=============================================================================
void read_all_digi_pyro(void)
{
char n;
char pb;
#use fast_io(B) // need to set the TRIS bits manually
output_b(SET_ALL_DOCI); // Set all DOCI pins
doci_dir&=~DOCI_PINS_DIR; // Clear direction bit
```

```
set_tris_b(doci_dir); // DOCI pin input
delay_us(75);
pir_data[0]=0; // 0
pir_data[1]=0; // 1
pir_data[2]=0; // 2
pir_data[3]=0; // 3
for(n=0;n<14;n++) // There are 14 bits of data from the sensors

{
output_b(CLEAR_ALL_DOCI); // Clear all DOCI pins
doci_dir&=~DOCI_PINS_DIR; // Clear direction bit
set_tris_b(doci_dir); // DOCI pin output
delay_cycles(2); // Wait a bit
output_b(SET_ALL_DOCI); // Set all DOCI pins
doci_dir|=DOCI_PINS_DIR; // Set direction bit
set_tris_b(doci_dir); // DOCI pin input
pir_data[0]<<=1; // 0
pir_data[1]<<=1; // 1
pir_data[2]<<=1; // 2
pir_data[3]<<=1; // 3
pb=input_b();
if((pb&ZERO)==ZERO)
pir_data[0]++;
if((pb&ONE)==ONE)
pir_data[1]++;
if((pb&TWO)==TWO)
pir_data[2]++;
if((pb&THREE)==THREE)
pir_data[3]++;
}
clear_all_DOCI_interrupt();
#use standard_io(B)
}
```

封装规格

标准的SOP 8 封装 （商业级），SMD 8金属及陶瓷（工业级）

使用告知

1.所有类型的采样干扰都会与来自于环境有关，从而造成误报，因此带来的经济损失，此电路生产商不负责任。

2.这个电路不能使用于非法场所以及秘密场所。当采用此电路的探测器被经常性地使用在这些场所时，探测器制造商将不负任何法律责任。

3.所有的原材料都是与 RoHS要求一致的。

4.如果电路的规格将来改变或性能提高了，原谅我们未必通知您。