

前置超低噪声放大器 24 位模/数 (A/D) 转换器芯片

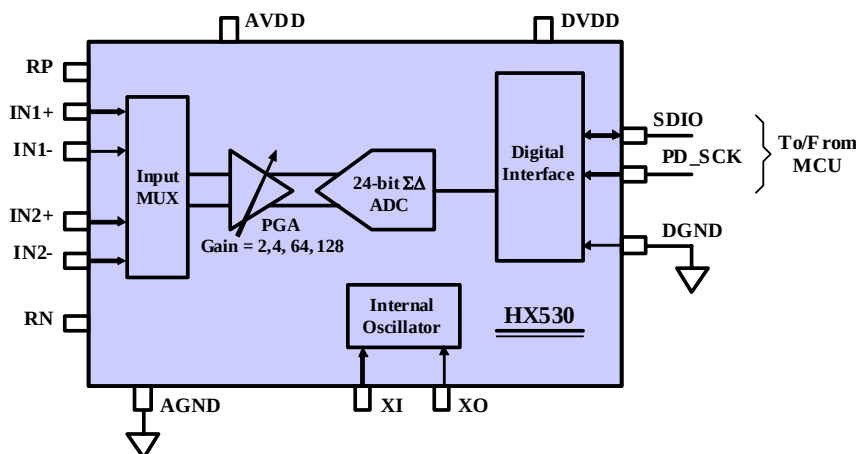
简介

HX530 采用了海芯科技集成电路专利技术，是一款专为高精度仪器仪表而设计的 24 位 A/D 转换器芯片。

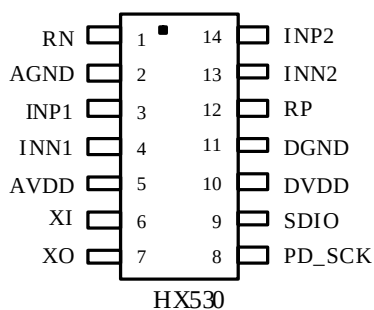
内置超低噪声 4~128 倍可编程增益斩波放大器，当增益为 128 倍时，在 0.1Hz 处的典型输入噪声为 10nV/√Hz。芯片提供了两个差分信号输入通道，通道 1 的可编程增益为 4、8、64 和 128，通道 2 的可编程增益为 4 和 8。当增益为 4 和 8 时，共模输入电压范围可达：0~AVDD。可编程 2.5~1280Hz 模/数转换输出数据速率。片内时钟振荡器无需任何外接器件，必要时也可使用外接晶振或时钟。模拟和数字电源 (AVDD 和 DVDD) 可以在 3~5V 之间灵活选择，例如：可以使用 AVDD=5V 来提高 A/D 的转换精度，而选择 DVDD=3.3V 来降低整机数字部分的功耗和增加 MCU 选型上的灵活性。MCU 只需要 2 个 I/O 口即可实现对 ADC 的所有控制，包括断电控制。

特点

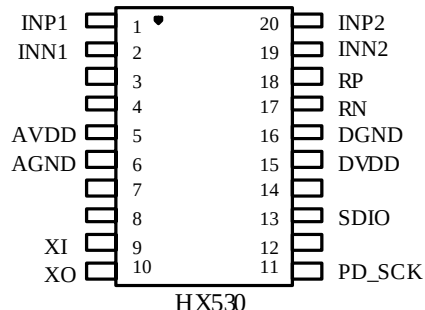
- 内置输入噪声 10nV/√Hz (在 0.1Hz 处) 的超低噪声 4~128 倍可编程增益斩波放大器
- 两个差分信号输入通道，可编程增益分别为 4/8/64/128 和 2/4
- 可编程 2.5~1280Hz 模/数转换输出数据速率
- 片内时钟振荡器无需任何外接器件，必要时也可使用外接晶振或时钟
- MCU 只需要 2 个 I/O 口即可实现对 ADC 的所有控制，包括断电控制
- 模拟和数字电源 (AVDD 和 DVDD) 可以在 3~5V 之间灵活选择
- 同步抑制 50Hz 和 60Hz 的电源干扰
- 典型工作电流：6mA
- 工作电压范围：3~5.5V
- 工作温度范围：-40~+85℃
- 14 管脚的 SOP-14 或 20 管脚的 SSOP-20 封装



管脚说明



SOP-14L 封装



SSOP-20 封装

名称	性能	描述
INP1	模拟输入	通道 1 差分信号正输入端
INN1	模拟输入	通道 1 差分信号负输入端
INP2	模拟输入	通道 2 差分信号正输入端
INN2	模拟输入	通道 2 差分信号负输入端
RP	模拟输入	A/D 转换参考正输入端：1.8 ~ 5.5V
RN	模拟输入	A/D 转换参考负输入端：0 ~ 1.8V
XI	数字输入	晶振或外部时钟输入。接地：使用片内振荡器
XO	数字输出	晶振输入（不用晶振时为无连接）
AVDD	模拟电源	模拟电源输入（3.0~5.5V，应等于或高于 DVDD）
AGND	模拟地	模拟地输入
DVDD	数字电源	数字电源输入（3.0~5.5V，应等于或低于 AVDD）
DGND	数字地	数字地输入
SDIO	数字输入和输出	串口数据输入和输出（高电平=DVDD）
PD_SCK	数字输入	断电控制（低电平有效）和串口时钟输入

表一 管脚描述

主要电气参数

(无特殊说明时: 信号输入=通道 1, AVDD=5V, DVDD=5V, VREF=RP-RN=5V, GAIN=64/128, 输出速率=10Hz)

参数	条件及说明	最小值	典型值	最大值	单位
满量程差分输入范围 (FSR)	$V(\text{inp})-V(\text{inn})$	$\pm 0.5(V_{\text{REF}}/\text{GAIN})$			V
输入共模电压范围		0.9		AVDD-1.3	V
VREF 输入电压范围	$V_{\text{REF}} = \text{RP} - \text{RN}$	1.8		AVDD	V
无噪声位数 (Noise-Free Bits) ⁽¹⁾	Gain=64, Rate=5Hz		19.6		Bits
	Gain=64, Rate=10Hz		19.2		Bits
	Gain=128, Rate=5Hz		19.1		Bits
	Gain=128, Rate=10Hz		18.7		Bits
输出数据速率 (f_o) ⁽²⁾		2.5/5/10/20/40/80/160/320/640/1280			Hz
输出数据编码	二进制补码	800000		7FFFFFFF	HEX
输出稳定时间 ⁽³⁾		$4 \times 1/f_o$			ms
晶体或外部时钟频率 (f_x)	当 $f_x=11.0592\text{MHz}$ 时, 输出数据速率与使用内部振荡器相同	5	11.0592	20	MHz
非线性误差 (INL)	相比满量程增益	± 0.001			%of FS
输入零点漂移 (Input Offset)		0.01			mV
输入噪声	在 0.1Hz 处	10			nV/ $\sqrt{\text{Hz}}$
温度系数 (Temperature Drift)	零点漂移 (offset drift)	± 15			nV/ $^{\circ}\text{C}$
	增益漂移 (gain drift)	± 3			ppm/ $^{\circ}\text{C}$
模拟和数字电源电压	$\text{DVDD} \leq \text{AVDD}$	3.0	5	5.5	V
电源信号抑制比		100			dB
共模信号抑制比	At DC, $\Delta V_{\text{IN}}=10\text{mV}$	100			dB
电源电流	正常工作 (AVDD)	5.8			mA
	正常工作 (DVDD)	0.2			mA
	断电	0.5			μA

(1) 无噪声位数 (Noise-Free Bits) = $\ln(\text{FSR}/\text{Peak-to-Peak Noise})/\ln(2)$ 。

(2) CKSEL=0, 采用内部时钟或外部 11.0592MHz 时钟时, $f_o = 5/10/20/40/80/320/640/1280$ Hz。

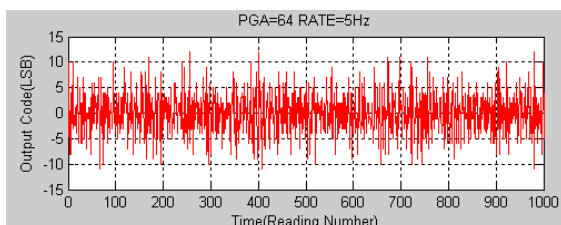
CKSEL=1, 采用内部时钟或外部 11.0592MHz 时钟时, $f_o = 2.5/5/10/20/40/160/320/640$ Hz。

(3) 输出稳定时间指从上电、复位或寄存器值改变到有效的稳定输出数据的时间。

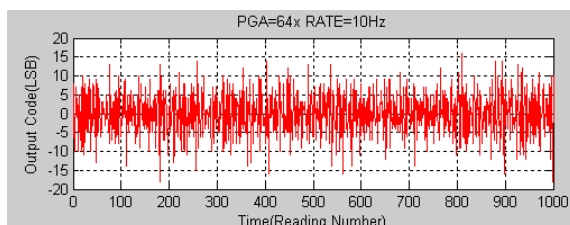
表二 主要电气参数表

典型噪声特性

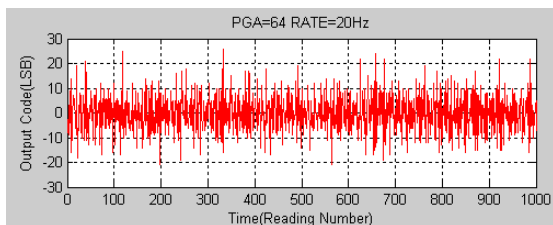
测试条件：温度=25℃；AVDD=DVDD=RP=5V，RN=AGND；使用片内振荡器。



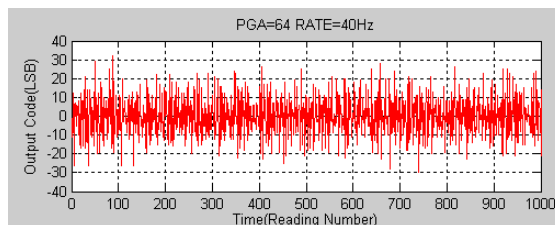
图二 输出噪声（增益=64，速率=5Hz）



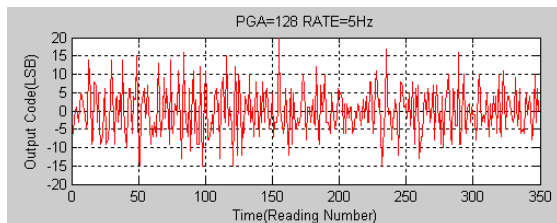
图三 输出噪声（增益=64，速率=10Hz）



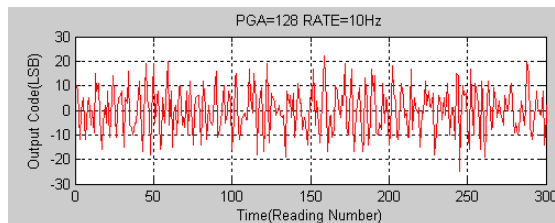
图四 输出噪声（增益=64，速率=20Hz）



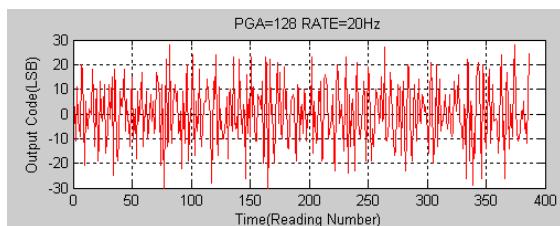
图五 输出噪声（增益=64，速率=40Hz）



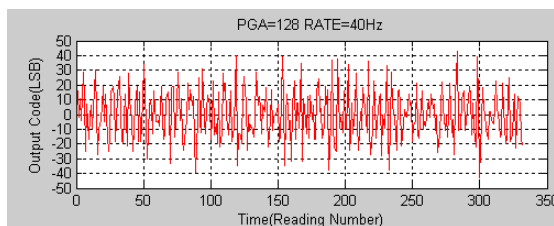
图六 输出噪声（增益=128，速率=5Hz）



图七 输出噪声（增益=128，速率=10Hz）



图八 输出噪声（增益=128，速率=20Hz）



图九 输出噪声（增益=128，速率=40Hz）

模拟信号输入

模拟差分输入可直接与桥式传感器的差分输出相接。当前置放大器的增益为 128，参考电压 $V_{REF}=R_P-R_N$ 为 5V 时，所对应的满量程差分输入电压为 $\pm 20\text{mV}$ 。

同时，应注意差分信号输入的共模电压范围要求。当增益为 64 或 128 时，差分输入信号的共模电压为 $0.9\sim(\text{AVDD}-1.3)\text{V}$ 。当增益为 4 或 8 时，共模电压为 $0\sim\text{AVDD}$ 。

供电电源和 A/D 转换参考电压输入

供电电源 AVDD 和 DVDD 的电压可以任意在 $3\sim 5.5\text{V}$ 之间选择，但是应满足 DVDD 的电压不高于 AVDD 电压的要求。在精度要求较高时，AVDD 应采用 5V 供电，DVDD 可以采用 3.3V，这样可以降低数字电路功耗，同时减少数字电路对模拟电路的干扰。

A/D 转换参考电压输入 (R_P 和 R_N) 应分别与传感器的供电电源和地直接相连。该电压可直接取用 AVDD，也可由 AVDD 经电阻与传感器分压后供给，以减少传感器的耗电量。但这样会造成传感器输出信号幅值减小，使整机精度下降。

时钟选择

如果将管脚 XI 接地，HX530 将自动选择使用内部时钟振荡器，并自动关闭外部时钟输入和晶振的相关电路。使用内部时钟振荡器工作时的输出数据速率，与使用 11.0592MHz 的外部时钟振荡器或晶体时相同。

如果需要准确的输出数据速率，或者改变在给定 RATE[2:0] 寄存器设置时的输出数据速率，可将外部输入时钟通过一个 20pF 的隔直电容连接到 XI 管脚上，或将晶振直接连接到 XI 和 X0 管脚上。这种情况下，芯片内的时钟振荡器电路会自动关闭，晶振时钟或外部输入时钟电路被采用。此时，若晶振频率为 11.0592MHz ，根据 CKSEL 寄存器和 RATE[2:0] 寄存器的设置，对应的输出数据速率为准确的 $2.5/5/10/20/40/80/160/320/640/1280\text{Hz}$ 。若改变晶振时钟或外部输入时钟频率，输出数据

速率与晶振频率以上述关系按比例增加或减少。

使用外部输入时钟时，外部时钟信号不一定需要为方波。可将 MCU 芯片的晶振输出管脚上的时钟信号通过 20pF 的隔直电容连接到 HX530 的 XI 管脚上，作为外部时钟输入。外部时钟输入信号的幅值可低至 150mV 。

寄存器设置和串口通讯

寄存器设置如下表所示。

寄存器名	功能	描述
CKSEL	时钟选择	0: 高速时钟 1: 低速时钟
CHSEL	通道选择	1: 通道 1 0: 通道 2
GAIN[1: 0]	增益选择	通道 1: [00~11]: 4/8/64/128 倍 通道 2: [x0~x1]: 4/8 倍
RATE[2: 0]	输出速率	CKSEL=0 时, [000~111]: 5/10/20/40/80/320/640/1280 CKSEL=1 时, [000~111]: 2.5/5/10/20/40/160/320/640
DATA[23: 0]	数据	24 位无失码数据

芯片采用 2 线串口通讯方式，包含 2 根信号线：串行时钟输入口 PD_SCK，串行数据输入输出口 SDIO。

PD_SCK 不带内部上拉电阻，默认是高电平。用于提供串行时钟，芯片在其上升沿从 SDIO 口读取数据，在其下降沿从 SDIO 口输出数据。

SDIO 不带内部上拉电阻，默认是高电平。SDIO 用于串行数据输入和输出，高电平表示数据 1，低电平表示数据 0。串行数据输入或输出的顺序是高位在前，低位在后。

在 PD_SCK 为高电平期间发生的 SDIO 下降沿定义为串行接口的启动信号，在 PD_SCK 为高电平期间发生的 SDIO 上升沿定义为串行接口的停止信号。芯片与 MCU 的通讯过程分为 4 个步骤，按 MCU 的操作方向分成两种类型，一种是写操作，用于发送数据；一种是读操作，用于接收数据。

写操作包括以下 4 个步骤：MCU 输出启动信号、MCU 输出写信号“0”、MCU 输出 8 位数据

(CKSEL+1+CHSEL+GAIN[1:0]+RATE[2:0])、MCU 输出停止信号，如图十所示。

写操作时，PD_SCK 必须输入 9 个正脉冲，数据才能正确写入芯片内部的寄存器，少于或多于 9 个脉冲都无效。

当芯片的寄存器被修改时，A/D 转换器需要 4 个数据输出周期才能稳定，SDIO 在 4 个数据输出周期后才会从高电平变低电平，输出有效数据。

读操作包括以下 4 个步骤：芯片输出串口通信启动信号、MCU 输出读信号“1”、芯片输出 30 位数据

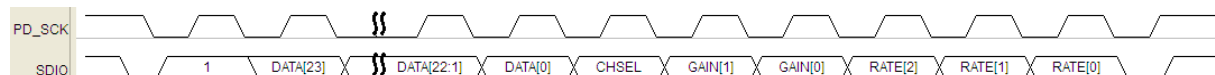
(DATA[23:0]+CHSEL+GAIN[1:0]+RATE[2:0])、芯片输出串口通信停止信号，如图十一所示。

读操作时，当 SDIO 为高电平时，表明 A/D 转换器还未准备好输出数据，此时 PD_SCK 应为高电平。当 SDIO 从高电平变低电平后，可以输入 32 个脉冲读取所有数据；也可以输入 25 个脉冲只读取 A/D 转换数据 DATA[23: 0]；也可以输入少于 25 个脉冲只读取高位 DATA，如输入 21 个脉冲，则读取 DATA[23: 4]。

在串口数据通讯期间，如果 PD_SCK 处于高电平的时间超过 140 μ S（采用内部时钟或外部 11.0592MHz 时钟，且 CKSEL=0），则芯片会自动复位该次数据传输，该次数据传输无效。



图十 MCU 写操作



图十一 MCU 读操作

说明	最小值	典型值	最大值	单位
SDIO 下降沿到 PD_SCK 下降沿	0.1			μ S
PD_SCK 下降沿到 SDIO 数据有效			0.1	μ S
PD_SCK 负脉冲电平时间	0.2		70	μ S
PD_SCK 正脉冲电平时间	0.2		140	μ S

复位和断电

当芯片上电时，芯片内的上电自动复位电路会使芯片自动复位，但各寄存器值处于不确定状态，上电后 MCU 必需要先通过写操作来配置芯片内部的各个寄存器。

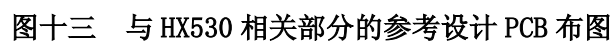
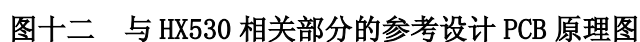
PD_SCK 脚可以用于控制 HX530 的断电。当 PD_SCK 为高电平时，芯片处于正常工作状态。当 PD_SCK 从高电平变低电平并保持在低电平超过 70 μ S（采用内部时钟或外部 11.0592MHz 时钟，且 CKSEL=0），芯片进入断电状态。当

PD_SCK 重新回到高电平时，芯片会重新进入断电前的工作状态。

芯片从复位或断电状态进入正常工作状态后，A/D 转换器需要 4 个数据输出周期才能稳定。SDIO 在 4 个数据输出周期后才会从高电平变低电平，输出有效数据。

参考 PCB 设计

图十二和图十三为与 HX530 芯片相关部分的参考 PCB 设计原理图和相应的单面板参考布图。



参考驱动程序（汇编）

MCU 写操作：

```
/*-----*/
Input: R6 = CKSEL+1+CHSEL+GAIN[1:0]+RATE[2:0]
/*-----*/

Send:
    SETB PD_SCK          // 空闲时PD_SCK为高电平
    ##0##                // 将SDIO切换成推挽输出或输出状态
    SETB SDIO
    CLR SDIO             // 步骤1: MCU发送启动信号，拉低SDIO
    CLR PD_SCK
    CLR SDIO             // 步骤2: MCU写操作第1位送0
    SETB PD_SCK
    MOV R4, #8           // 步骤3: MCU发送8位数据: CKSEL、1、CHSEL、GAIN[1:0]、RATE[2:0]
Lop1:
    CLR PD_SCK
    MOV A, R6
    JNB ACC. 7, Low
High:
    SETB SDIO
    SJMP Com
Low:
    CLR SDIO
Com:
    SETB PD_SCK
    XCH A, R6            // 数据移位
    RLC A
    XCH A, R6
    DJNZ R4, Lop1       // 判断是否发送完8位数据
    CLR PD_SCK          // 步骤4: MCU发送停止信号
    CLR SDIO
    SETB PD_SCK
    SETB SDIO
    ##I##               // 将SDIO切换成准双向或输入状态
    RET
    END
```

MCU读操作：

```
/*-----*/
Output: R5, R6, R7  R5=DATA[23:16], R6=DATA[15:8], R7=DATA[7:0]
/*-----*/

Read:
    SETB PD_SCK
    SETB SDIO           // 对于准双向口，通过写1开启IO口的上拉功能
    ##I##               // 将SDIO切换成准双向或输入状态
    JB SDIO, $          // 步骤1: HX530发送启动信号，拉低SDIO
    CLR PD_SCK
    ##0##               // 将SDIO切换成推挽输出或输出状态
    SETB SDIO           // 步骤2: MCU读数据第1位送1
```



```
SETB PD_SCK
##I## // 将SDIO切换成准双向或输入状态
MOV R4, #24 // 步骤3: MCU读取24位数据: DATA[23:0]
Lop2:
CLR PD_SCK
SETB PD_SCK
MOV C, SDIO
XCH A, R7 // 数据移位
RLC A
XCH A, R7
XCH A, R6
RLC A
XCH A, R6
XCH A, R5
RLC A
XCH A, R5
DJNZ R4, Lop2 // 判断是否读取完24位数据
CLR PD_SCK // 步骤4: HX530发送停止信号
SETB PD_SCK
RET
END
```

参考驱动程序（C）

MCU写操作:

```
void Senddata(unsigned int dat)
{
    unsigned char i;

    PD_SCK = 1; // 空闲时PD_SCK为高电平
    ##0##; // 将SDIO切换成推挽输出或输出状态
    SDIO = 1; // 空闲时SDIO为高电平
    SDIO = 0; // 步骤1: MCU发送启动信号, 拉低SDIO
    PD_SCK = 0;
    SDIO = 0; // 步骤2: MCU写操作第1位送0
    PD_SCK = 1;
    for (i=0; i<8; i++) // 步骤3: MCU发送8位数据: CKSEL、1、CHSEL、GAIN[1:0]、RATE[2:0]
    {
        PD_SCK = 0;
        if (dat&0x80) SDIO = 1;
        else SDIO = 0;
        PD_SCK = 1;
        dat = dat<<1;
    }
    PD_SCK = 0; // 步骤4: MCU发送停止位
    SDIO = 0;
    PD_SCK = 1;
    SDIO = 1;
    ##I##; // 将SDIO切换成准双向或输入状态
}
```

MCU读操作:

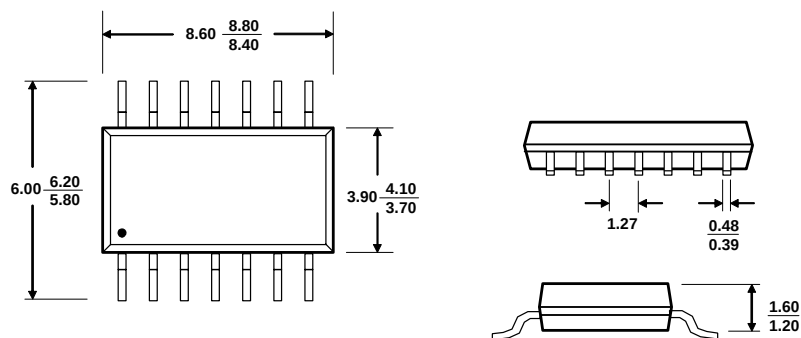
```
void Receivedata()
{
    unsigned char i;
    unsigned int dat;

    dat = 0;
    PD_SCK = 1;
    SDIO = 1;           // 对于准双向口, 通过写1开启IO口的上拉功能
    ##I##;              // 将SDIO切换成准双向或输入状态
    while (SDIO==1);    // 步骤1: HX530发送启动信号, 拉低SDIO
    PD_SCK = 0;
    ##0##;              // 将SDIO切换成推挽输出或输出状态
    SDIO = 1;           // 步骤2: MCU读数据第1位送1
    PD_SCK = 1;
    ##I##;              // 将SDIO切换成准双向或输入状态
    for (i=0;i<24;i++)  // 步骤3: MCU读取24位数据: DATA[23:0]
    {
        PD_SCK = 0;
        PD_SCK = 1;
        dat = dat<<1;
        if (SDIO==1) dat++;
    }
    PD_SCK = 0;         // 步骤4: HX530发送停止信号
    PD_SCK = 1;
}
```

注意事项

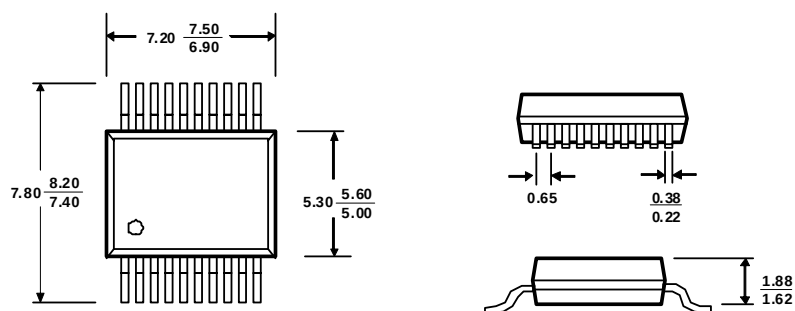
1. 输入管脚, 包括 SDIO 和 XI 管脚, 芯片内没有内置的拉高或拉低电阻。这些管脚在使用时不应悬空。
2. 建议使用通道 1 与传感器相连, 作为小信号输入通道; 通道 2 用于系统参数检测, 如电池电压检测。如果其中的一个通道没有使用, 其输入端可以接地或者悬空 (无连接)。
3. 传感器和 A/D 转换器应使用同一模拟供电电源, 尤其是传感器电源和地端口应该直接连接到 RP 和 RN 管脚。
4. 与 SDIO 相连的 MCU 接口在正常使用时应设置为输入口, 并且不接任何拉高或拉低电阻, 以减少 MCU 与 ADC 之间的电流交换 (干扰)。

封装尺寸



典型值 $\frac{\text{最大值}}{\text{最小值}}$ 单位: 毫米

SOP-14L 封装



典型值 $\frac{\text{最大值}}{\text{最小值}}$ 单位: 毫米

SSOP-20 封装