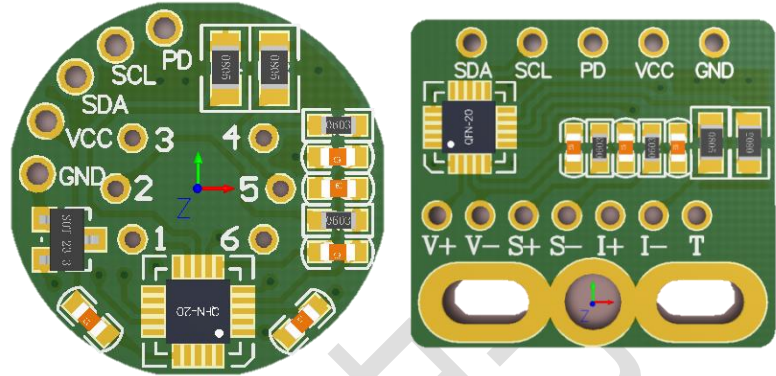


## 宝鸡市利诺德 IIC 板卡使用说明

版本：V0.1

## 一、特点

- IIC 接口输出接口
- 具有恒压、恒流激励源输出
- 24bit 高精度 ADC
- 具有多点非线性修正和温度补偿
- 具有低功耗控制引脚，适合低功耗运用场合
- 有批量校准系统



圆板尺寸：15mm

方板 21\*18.9

## 二、技术参数

供电电源	3.0~5.5V DC	正常模式工作电流	3mA(和传感器激励源有关)
输出方式	IIC	睡眠模式工作电流	100nA
ADC 分辨率	24bit	转换速率	10Hz~2.4KHz
激励源类型	恒流、恒压	唤醒时间	4ms
工作温度	-40~85℃	存储温度	-40~85℃

## 三、接线说明

GND	供电电源负
VCC	供电电源正
PD	休眠引脚，内置 68k 上拉电阻，低电平休眠，高电平唤醒
SCL	IIC 接口时钟引脚
SDA	IIC 接口数据引脚

## 四、IIC 接口说明

I2C 总线使用 SCL 和 SDA 作为信号线。这两根线都通过上拉电阻连接到 VCC，不通信时都保持为高电平。

I2C 设备地址如下：

A7	A6	A5	A4	A3	A2	A1	W/R
1	1	0	1	1	0	1	0/1

通讯引脚的电性特性

标示	参数	条件	最小值	最大值	单位
f <sub>scl</sub>	时钟频率			400	kHz
t <sub>LOW</sub>	时钟低脉冲维持时间		1.3		us
t <sub>HIGH</sub>	时钟高脉冲维持时间		0.6		us
t <sub>SUDAT</sub>	SDA 建立时间		0.1		us
t <sub>HDDAT</sub>	SDA 保持时间		0.0		Us
t <sub>SUSTA</sub>	每次开始时的建立时间		0.6		Us
t <sub>HDSTA</sub>	开始条件保持时间		0.6		Us
t <sub>SUSTO</sub>	停止时间建立时间		0.6		Us
t <sub>BUF</sub>	两次通讯之间间隔时间		1.3		Us

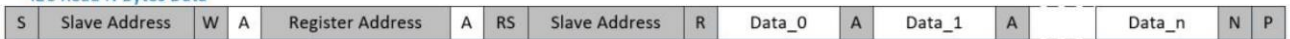
I2C 通讯协议有着特殊的开始(S)和终止 (P)条件。当 SCL 处于高电平同时，SDA 的下降沿标志数据传输开始。I2C 主设备依次发送从设备的地址（7 位）和读/写控制位。当从设备识别到这个地址后，产生一个应答信号并在第九个周期将 SDA 拉低。得到从设备应答后，主设备继续发送 8 位寄存器地址，得到应答后继续发送或读取数据。SCL 处于高电平，SDA 发生一个上升沿动作标志 I2C 通信结束。除了开始和结束标志之外，当 SCL 为高时 SDA 传输的数据必须保持稳定。当 SCL 为低时 SDA 传输的值可以改变。I2C 通信中的所有数据传输以 8 位为基本单位，每 8 位数据传输之后需要一位应答信号以保持继续传输。

具体 I2C 通讯格式如下图所示

I2C Read 1 Byte Data



I2C Read N Bytes Data



## 五、寄存器说明

地址	位地址	寄存器名	默认值	描述
0x06	7-0	PDATA<23:16>	0x00	主通道数据寄存器，只读，有符号数
0x07	7-0	PDATA<15:8>	0x00	
0x08	7-0	PDATA<7:0>	0x00	
0xC5	7-0	RANGE_L<15:8>	0x00	量程零位，只读，有符号整型
0xC6	7-0	RANGE_L<7:0>	0x00	
0xC7	7-0	RANGE_H<15:8>	0x00	量程满度，只读，有符号整型
0xC8	7-0	RANGE_H<7:0>	0x00	
0xCF	7-0	DOT<7:5> UNIT<4:0>	0x00	只读 高 3 位：量程小数位 低 5 位：单位 0: kPa 1: MPa 2: mH2O

				3: bar
				4: psi
				5: mbar
				6: kgf/cm2
				7: mmHg
				8: atm
				9: Pa
				10: inH2O
				12: inHg
				13: mmH2O
				14: mm
				15: m
				16: °C
				17: °F

测量值计算公式:

量程零位:  $\text{rangeL} = \text{RANGE\_L} / 10^{\text{DOT}}$

量程满度:  $\text{rangeH} = \text{RANGE\_L} / 10^{\text{DOT}}$

输出值: 如果 PDATA > 8388608, 则 pdata = PDATA | 0xFF000000

如果 PDATA ≤ 8388608, 则 pdata = PDATA

测量值 =  $(\text{rangeH} - \text{rangeL}) * ((\text{pdata} - 8388608 * 0.1) / (8388608 * 0.8))$

C 语言源码:

```
int32_t pdata;           //PATA
```

```
float rangeL;           //量程零位
```

```
float rangeH;           //量程满度
```

```
uint8_t dot;           //小数位数
```

```
float result;           //测量值
```

```
pdata = (((uint32_t)(Reg[0x06]) << 16) + ((uint32_t)(Reg[0x07]) << 8) + ((uint32_t)(Reg[0x08]) << 0) & 0xFFFFF);
```

```
if(Reg[0x06] & 0x80)
```

```
{
    pdata |= 0xFF000000;
}
```

```
dot = (Reg[0xCF] >> 5) & 0x07;
```

```
rangeL = (float)((int16_t)(((uint16_t)Reg[0xC5] << 8) + Reg[0xC6])) / pow(10, dot);
```

```
rangeH = (float)((int16_t)(((uint16_t)Reg[0xC7] << 8) + Reg[0xC8])) / pow(10, dot);
```

```
result = (rangeH - rangeL) * ((float)(adc - 8388608 * 0.1) / (8388608 * 0.8));
```