

基于51单片机的HX711模块

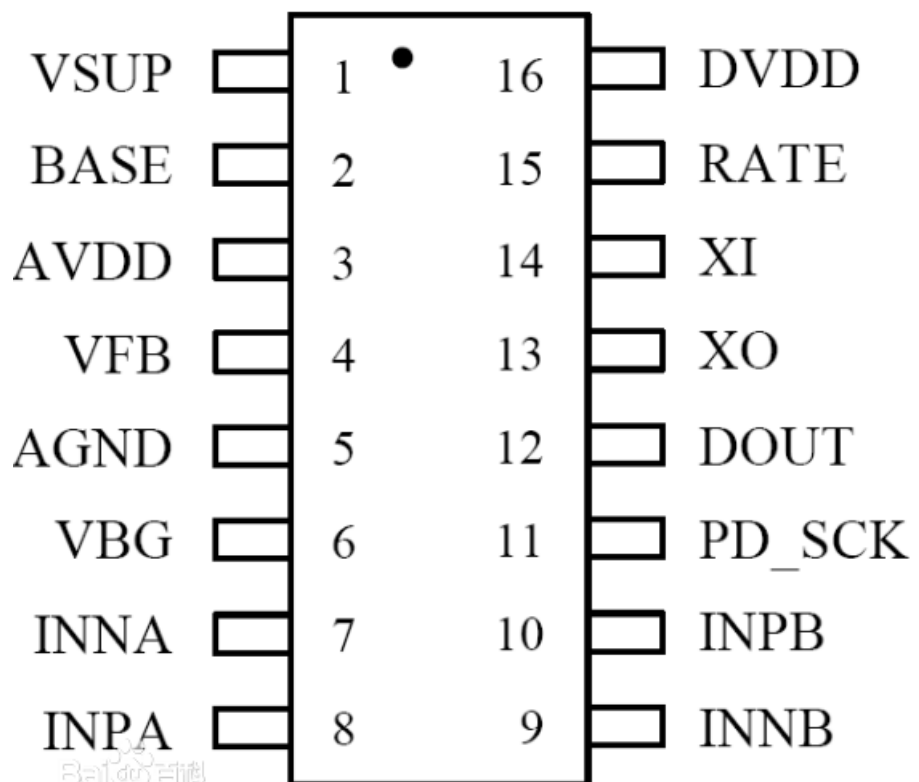
（一）项目介绍

基于51单片机的HX711称重计，最大称重量为5kg，并且在LCD1602上显示数值，可设置称重物品单价，称重精度可由按键key调节。误差可达0。传感器在某宝上买的，需要可以找我拿连接。

（二）HX711简介

HX711是一款专为高精度电子秤而设计的24位A/D转换器芯片。与同类型其它芯片相比，该芯片集成了包括稳压电源、片内时钟振荡器等其它同类型芯片所需要的外围电路，具有集成度高、响应速度快、抗干扰性强等优点。降低了电子秤的整机成本，提高了整机的性能和可靠性。该芯片与后端MCU 芯片的接口和编程非常简单，所有控制信号由管脚驱动，无需对芯片内部的寄存器编程。输入选择开关可任意选取通道A 或通道B，与其内部的低噪声可编程放大器相连。通道A 的可编程增益为128 或64，对应的满额度差分输入信号幅值分别为 $\pm 20\text{mV}$ 或 $\pm 40\text{mV}$ 。通道B 则为固定的32 增益，用于系统参数检测。芯片内提供的稳压电源可以直接向外部传感器和芯片内的A/D 转换器提供电源，系统板上无需另外的模拟电源。芯片内的时钟振荡器不需要任何外接器件。上电自动复位功能简化了开机的初始化过程。

（三）芯片管脚图以及功能



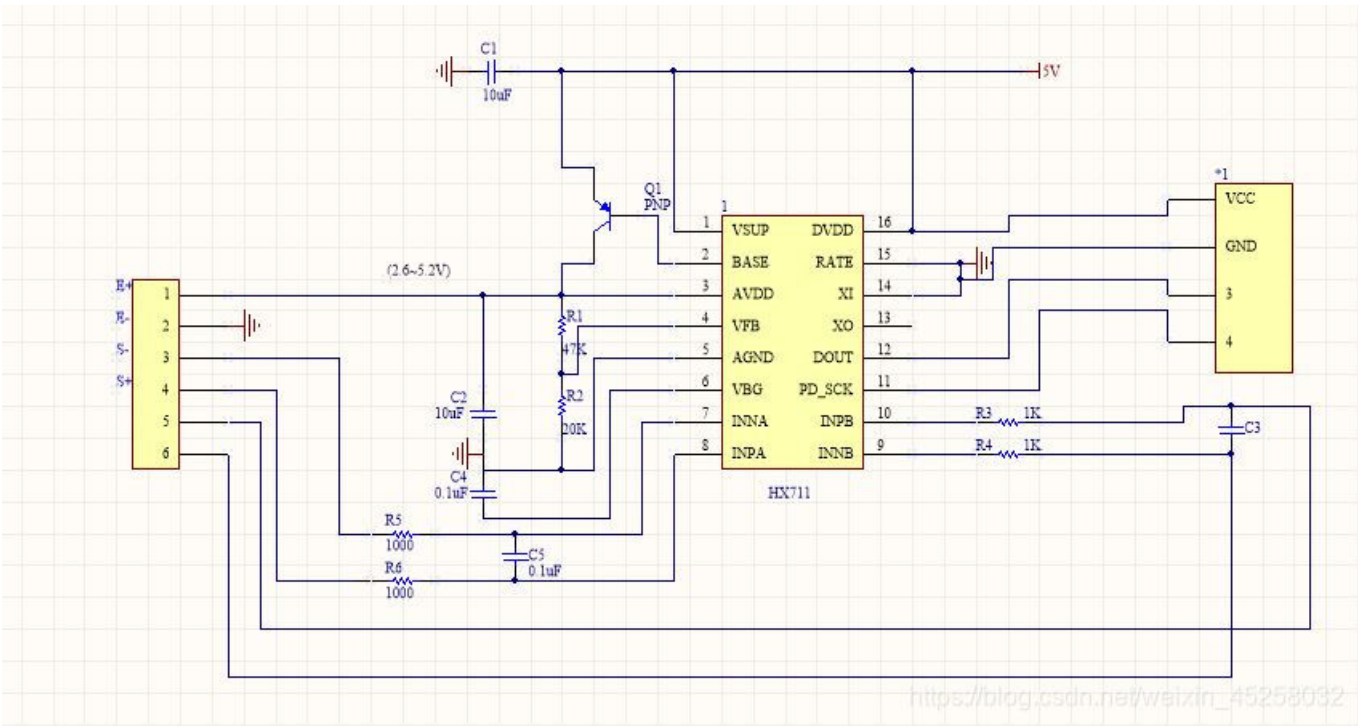
管脚功能：

- 1 VSUP 电源 稳压电路供电电源:2.6~5.5V（不用稳压电路时应接AVDD）
- 2 BASE 模拟输出 稳压电路控制输出（不用稳压电路时可为无连接）
- 3 AVDD 电源 模拟电源:2.6~5.5V
- 4 VFB 模拟输入 稳压电路控制输入（不用稳压电路时应接地）
- 5 AGND 地 模拟地

- 6 VBG 模拟输出 参考电源输出
- 7 INA- 模拟输入 通道 A 负输入端
- 8 INA+ 模拟输入 通道 A 正输入端
- 9 INB- 模拟输入 通道 B 负输入端
- 10 INB+ 模拟输入 通道 B 正输入端
- 11 PD_SCK 数字输入 断电控制（高电平有效）和串口时钟输入
- 12 DOUT 数字输出 串口数据输出
- 13 XO 数字输入输出 晶振输入（不用晶振时为无连接）
- 14 XI 数字输入 外部时钟或晶振输入，0: 使用片内振荡器
- 15 RATE 数字输入 输出数据速率控制，0: 10Hz 1: 80Hz
- 16 DVDD 电源 数字电源: 2.6 ~ 5.5V

（四）HX711原理图

该图为HX711相关部分的PCB原理图



（五）HX711配置代码

记得配置DOUT和SCK端口。不要复用了就OK。最后把HX711的VCC和GND连接到单片机，配置端口相应连接上就可以了。

```
#include "HX711.h"
//*****
//延时
//*****
sbit HX711_DOUT=P1^4;
sbit HX711_SCK=P1^5;
void Delay_hx711_us(void)
{
    _nop_();
    _nop_();
}
//*****
//读取HX711
//*****
unsigned long HX711_Read(void) //增益128
{
    unsigned long count;
    unsigned char i;
    HX711_DOUT=1;
    Delay_hx711_us();
    HX711_SCK=0;
    count=0;
    while(HX711_DOUT);
    for(i=0;i<24;i++)
```

```

{
    HX711_SCK=1;
    count=count<<1;
    HX711_SCK=0;
    if (HX711_DOUT)
        count++;
}
HX711_SCK=1;
count=count^0x800000; //第25个脉冲下降沿时，转换数据
Delay__hx711_us();
HX711_SCK=0;
return (count);
}

```

(六) main函数部分主要代码

1.配置定时器

```

//定时器T0初始化
void Timer0_Init()
{
    ET0 = 1;           //允许定时器0中断
    TMOD = 1;          //定时器工作方式选择
    TL0 = 0xb0;
    TH0 = 0x3c;         //定时器赋予初始值
    TR0 = 1;           //启动定时器
}

//定时器0中断
void Timer0_ISR (void) interrupt 1 using 0
{
    uchar Counter;
    TL0 = 0xb0;
    TH0 = 0x3c;         //定时器赋予初值

    //每0.5秒刷新重量
    Counter ++;
    if (Counter >= 10)
    {
        FlagTest = 1;
        Counter = 0;
    }
}

```

2.按键去皮以及精度加减操作

```

void KeyPress()
{
    if (ROW1==0)       //去皮键
    {
        Delay_ms(5);
        if (ROW1==0)
        {
            // Get_Maopi();
            if (qupi==0)
                qupi=Weight_Shiwu;
            else
                qupi=0;
            Buzzer=0;
            Delay_ms(50);
            Buzzer=1;
            while (ROW1==0);
        }
    }
    if (ROW2==0)       //加
    {
        Delay_ms(5);
        if (ROW2==0)
        {
            while (!ROW2)
            {
                key_press_num++;
                if (key_press_num>=100)
                {
                    key_press_num=0;
                    while (!ROW2)
                    {
                        if (GapValue<10000)
                            GapValue+=10;
                        Buzzer=0;

```

```

    Delay_ms(10);
    Buzzer=1;
    Delay_ms(10);
    Get_Weight();
}
}
Delay_ms(10);
}
if(key_press_num!=0)
{
    key_press_num=0;
    if(GapValue<10000)
    GapValue++;
    Buzzer=0;
    Delay_ms(50);
    Buzzer=1;
}
write_eeprom();
}
}
if(ROW3==0)  //减
{
    Delay_ms(5);
    if(ROW3==0)
    {
        while(!ROW3)
        {
            key_press_num++;
            if(key_press_num>=100)
            {
                key_press_num=0;
                while(!ROW3)
                {
                    if(GapValue>1)
                    GapValue-=10;
                    Buzzer=0;
                    Delay_ms(10);
                    Buzzer=1;
                    Delay_ms(10);
                    Get_Weight();
                }
            }
            Delay_ms(10);
        }
        if(key_press_num!=0)
        {
            key_press_num=0;
            if(GapValue>1)
            GapValue--;
            Buzzer=0;
            Delay_ms(50);
            Buzzer=1;
        }
        write_eeprom();    //保存数
    }
}
}
}

```

3.称重模块

```

void Get_Weight()
{
    Weight_Shifu = HX711_Read();
    Weight_Shifu = Weight_Shifu - Weight_Maopi;  //获取净重

    Weight_Shifu = (unsigned int)((float)(Weight_Shifu*10)/GapValue)-qupi;  //计算实物的实际重量
    if(Weight_Shifu >= 5000)  //超重报警
    {
        Buzzer = !Buzzer;
        LED=!LED;
        LCD1602_write_com(0x80+0x40+8);
        LCD1602_write_word("--.---");
    }
    else
    {
        if(Weight_Shifu==0)
        LED=0;
        else if(Weight_Shifu>0)
        LED=1;
        Buzzer = 1;
        Display_Weight();
    }
}

```

```

}
}

```

4.主函数

```

void main()
{
    init_eeprom(); //开始初始化保存的数据
    Init_LCD1602(); //初始化LCD1602
    EA = 0;
    Timer0_Init();
    //初始化完成, 开启中断
    EA = 1;

    // Get_Maopi();
    LCD1602_write_com(0x80); //LCD上显示位置设置
    LCD1602_write_word(" welcome to use "); //
    LCD1602_write_com(0x80+0x40); //LCD上显示位置设置
    LCD1602_write_word("Electronic Scale");
    Delay_ms(2000);
    Get_Maopi();
    LCD1602_write_com(0x80); //LCD上显示位置设置
    LCD1602_write_word("The Weight: ");
    LCD1602_write_com(0x80+0x40); //LCD上显示位置设置
    LCD1602_write_word(" 0.000kg");
    // Get_Maopi(); //称毛皮重量

    while(1)
    {
        //每0.5秒称重一次
        if (FlagTest==1)
        {
            Get_Weight();
            FlagTest = 0;
        }
        KeyPress();
    }
}

```

LCD1602的配置参考我另一篇文章
望各位大佬指出不足之处, 感激不尽!