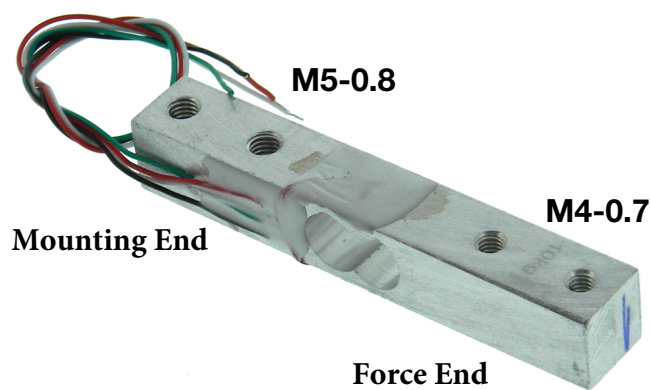# 34723 MI

Load Cell Principle and Use

1. Parallel beam load cell

Beam style strain gauge load cell features high precision, easy signal processing, compact size , high natural frequency. Typical structure & Specs. shown in Figure 1.

M5-0.8

M4-0.7

Mounting End

Force End

Material: Aluminum
Rated Load: 10kg
Overload: 150% F.S.
Sensitivity: 1mV/V
"0" Output: +-0.1mV/V
"0" Drift: .05%F.S./min
Resistance: 1K ohms +-50 ohms
Temperature Range: -10~+50deg. C
Temperature Drift: 0.2%F.S./10deg. C
Mounting Threads Fixed End: M5-0.8
Mounting Threads Force End: M4-0.7
L: 81mm **Sq:** 13mm **WT:** .06
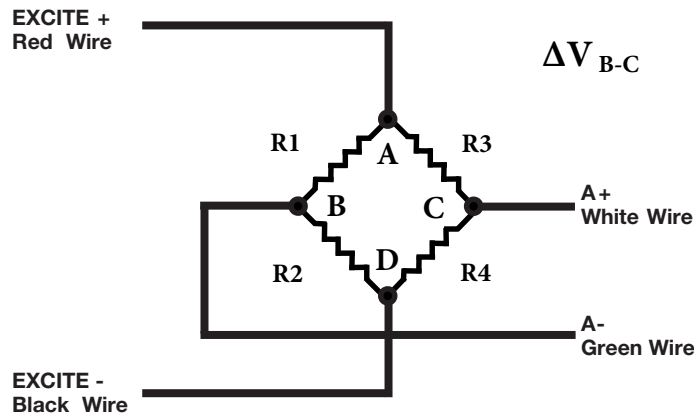
Figure1: Cantilever Parallel Beam Load Cell

2. Working Principal:
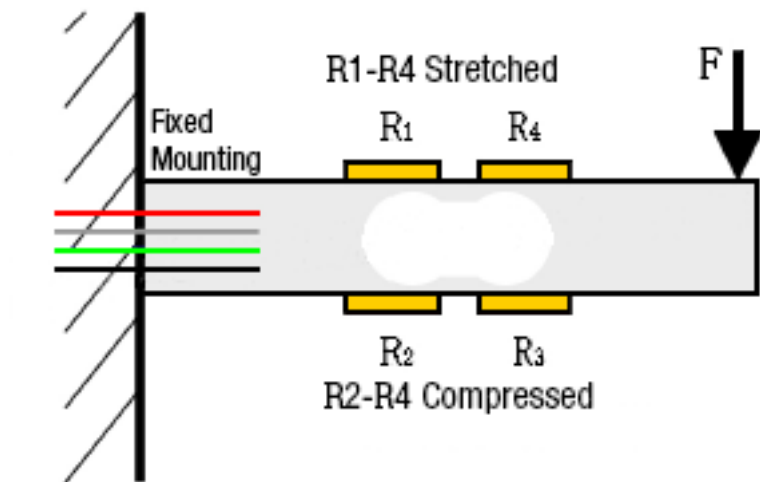
Figure 2: Typical Mounted Configuation

The strain gage resistance elements are attached to the aluminum bar whose 1 end is fixed (Figure 2). When a force flexes the Bar it causes a resistance changes in the bridge circuit that can used to calculate the weight of the measured object.

The resistance elements are configured as a balanced Bridge where all resistances are the same (Figure 3). The change in equilibrium causes an unbalanced resistance in the bridge circuit that will change the output voltage. By measuring the value of the output voltage, the weight of the measured object.

EXCITE +
Red Wire

ΔV_{B-C}

R1 A **R3**

B **C** A+
White Wire

R2 D **R4**

A-
Green Wire

EXCITE -
Black Wire

Figure 3: Strain Gage Resistance Bridge

The following formula can be deduced when the material and the resistance of the 4 arms of the bridge are the same:

$$\Delta U_{BC} = \frac{EK}{4}(\varepsilon_1 - \varepsilon_2 + \varepsilon_3 - \varepsilon_4) = \frac{EK}{4}4\varepsilon_1$$

3. Use

The parallel beam type sensor should be installed using cantilever mounting (Base mount on one End)

Sensor deformation is very small, you should pay special attention to not overload or shock (Drop item on bar or measurment surface). If the original form can not be restored after the removal of the external force, the sensor is damaged.
The sensor has four wires connecting the outer circuit,
Red lead is the power Positive (+)input, Black lead for power supply Negative (-) input,
White Lead is Signal Output 1,
Blue (or Green) lead is Signal Output 2.
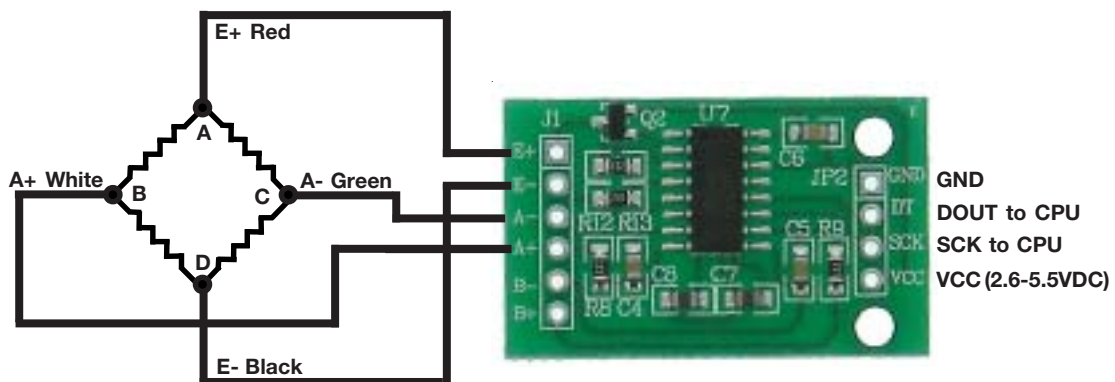For Best Accuracy, do not adjust lead length.

4. Interface Board:

Based on Avia Semiconductor's HX711 is a precision 24-bit analog-to-digital converter for weigh scales and industrial controls .applications to interface directly with bridge sensors and Microcontrollers
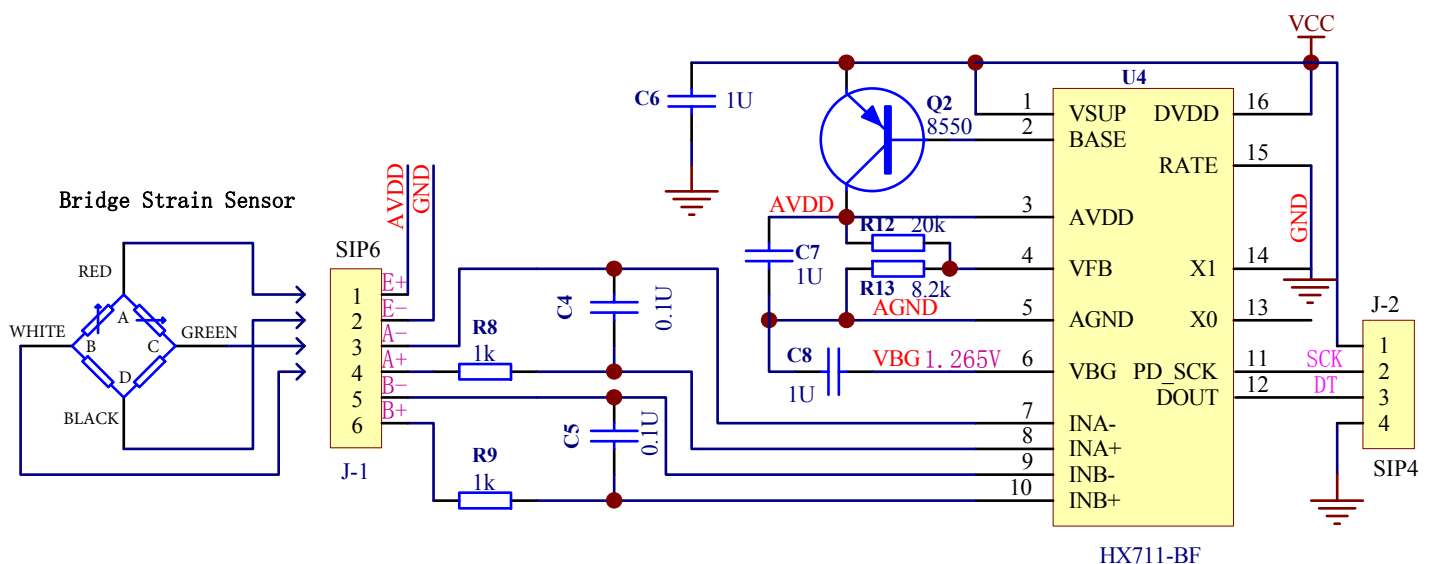Power: 2.6-5.5 VDC
Interface: DOUT & SCK
L: 35mm   W: 21mm   T: 3mm (+headers if used)

Figure 4: Connections

Figure 5: Schematic

Information including Drawings, Schematics, Links and Code (Software) Supplied or Referenced in this Document is supplied by MPJA inc. as a service to our customers and accuracy or usefulness is not guaranteed nor is it an Endorsement of any particular part, supplier or manufacturer. Use of information and suitability for any application is at users own discretion and user assumes all risk.

All rights are retained by the respective Owners/Author(s)

MARLIN P. JONES & ASSOC., INC.
P.O. Box 530400 Lake Park, Fl 33403
800-652-6733   FAX 561-844-8764
WWW.MPJA.COM

NOTE:
Driver Translated from Chinese. You should Review Before Use

```c
/*****************************************************************

        24bit HX711Reference driver (C)  Date 2010/11

Call in C: extern unsigned long ReadAD(void);
            .
            .
            unsigned long data;
            data=ReadAD();
            .
            .
-------------------------------------------------------------*/
sbit ADDO = P1^5;
sbit ADSK = P0^0;
unsigned long ReadCount(void)
{
    unsigned long Count;
    unsigned char i;
    ADSK=0;                 // enable AD (PD_SCK set low)
    Count=0;
    while(ADDO);        // AD conversion is not completed then wait,or start reading
    for (i=0;i<24;i++)
    {
        ADSK=1;         // PD_SCK high (send pulse)
        Count=Count<<1; // falling edge Count variable to the left by one,  the
        right zero ADSK = 0; / / PD_SCK set low
        if(ADDO) Count++;
    }
    ADSK=1;
    Count=Count^0x800000; // When the 25th pulse falls,  the conversion data is set low
     ADSK=0;
    return(Count);
}
```