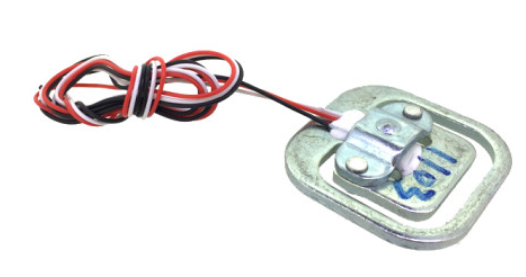
로드셀 자료

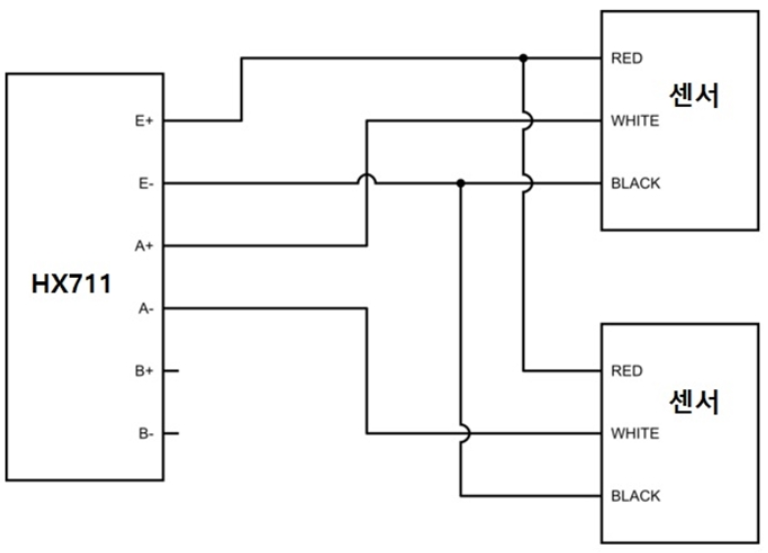
1. 3선 로드셀 스펙 + WIRING



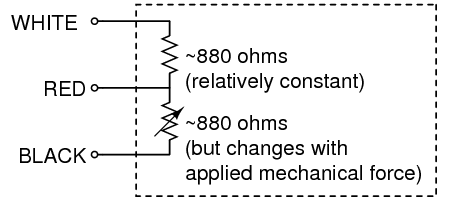
작동전압 5-10vdc

출력전압이 mv단위로 매우 작다. Hx711등 amp를 사용해야한다.

감지범위 0~50kg 출력감도 0.02(%FS) 비선형성0.02(%FS) 반복성0.02(%FS)



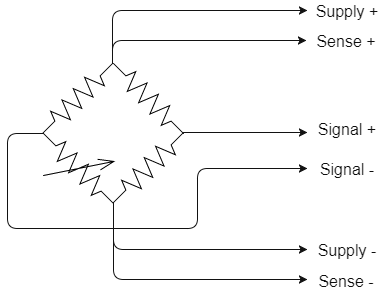
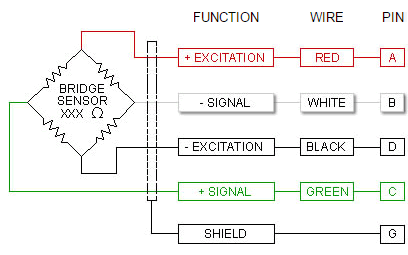
Example of a single strain gauge, or load sensor. Here RED is the center tap

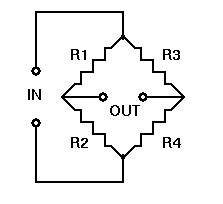


For load sensors, there isn't a set color coded standard. Comparing the scale pictured above with the load sensor schematic, while the black wires matched, the red and white wires were swapped.

To determine how to hook up your single strain gauge load cells to the combinator, measure the resistance between the three wires. You should find a larger resistance (close to double) between a pair. In our example the resistance between red and black was 1.6 kΩ, and the resistance between white/green and red was 800 Ω. Therefore, the center tap to the strain gauge is the white/green wire.

2. 4선셀





Vout ( V24 ) = Vin \*(( r2/r1+r2) – (r4/r4+r3)) (r2측을 +로 가정함.)

3.3V 입력할 때, 1Kg의 무게(최대값)를 주면, 전압출력값 Vout = 3.3\*0.001 V = 0.0033 V 이고, gain128곱해져서,

0V ~ 0.4244V

예상) 5V 입력일 때, 최대값은 128GAIN에서 20mV 이니까, 3.3V일때는 13.2mV 가 최대값이다.

풀 레인지 13.2mV라고할때, 그 대비 3mV 는 3/13.2 = 0.227 비율이다. 0x800000~0x8fffff(1048575)

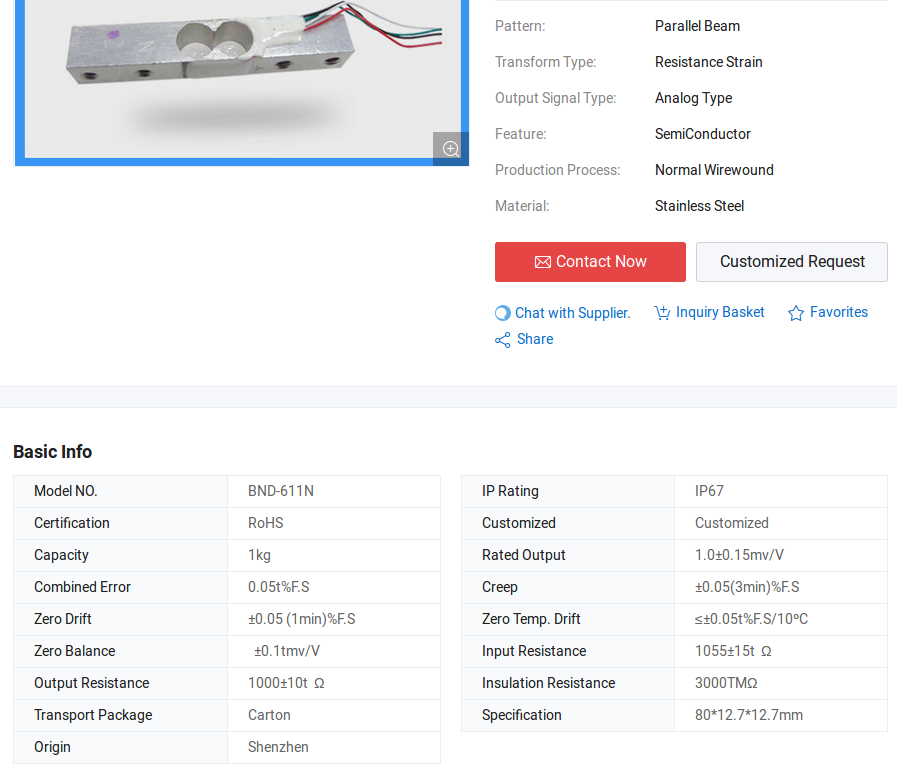
0.227\*1048575=238027

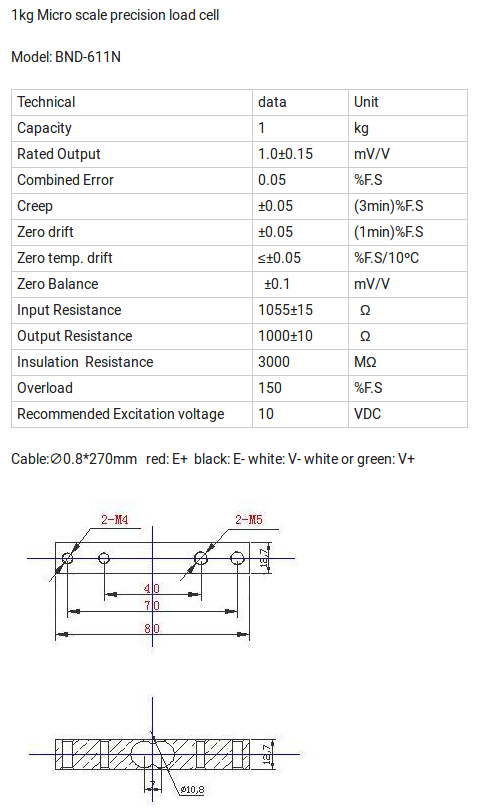
1kg 0.1A 611N 로드셀 정보

rated voltage : output voltage/ input voltage (at rated maximum weight)

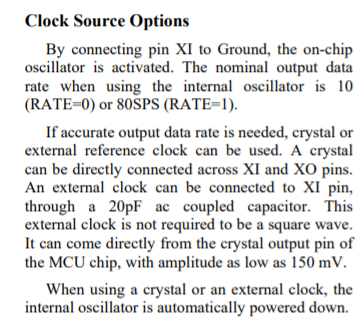
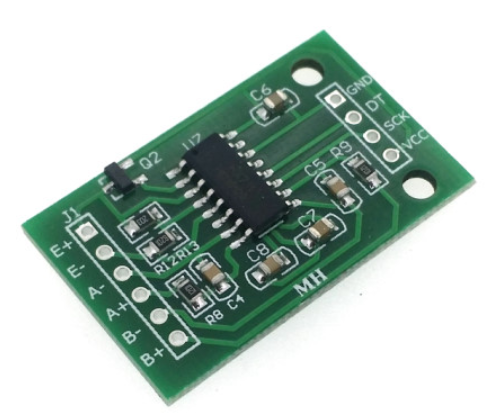
즉 1kg 을 올려놓고 , 3.3V전압을 인가할 때, 출력 전압은 3.3mV 가 될 것이다.

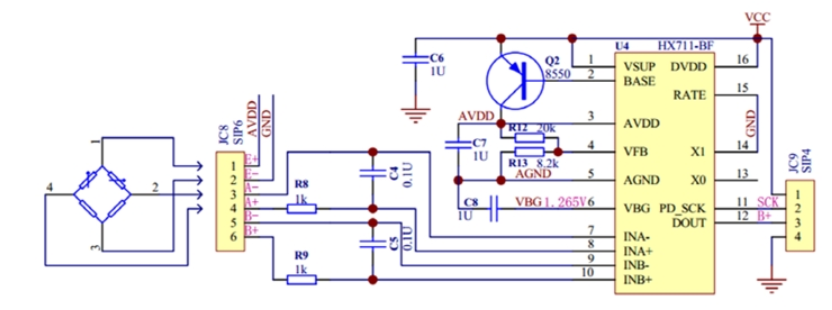
\* 실제 전압계로 측정해보니, 아무것도 무게 없을때 0.000V 1kg 정도 달았을때, 0.003V 찍히는 걸 확인했다.

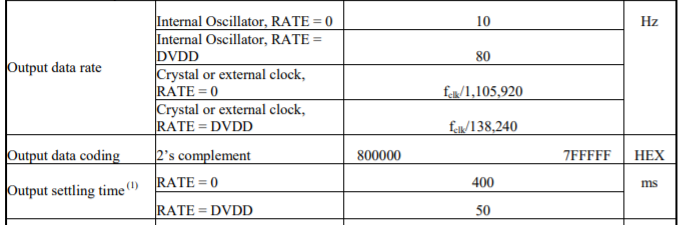




HX711 데이터시트



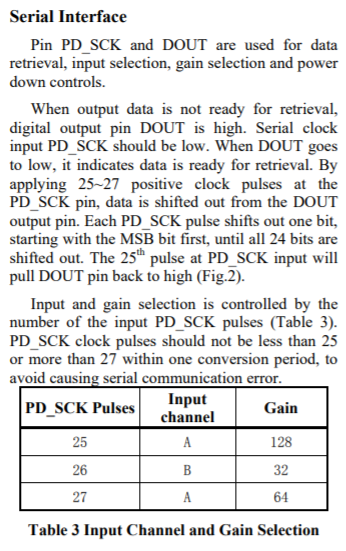
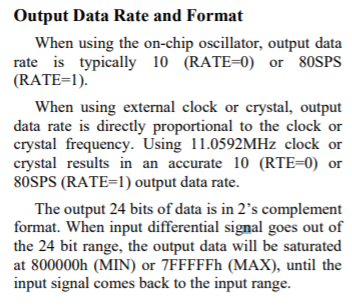


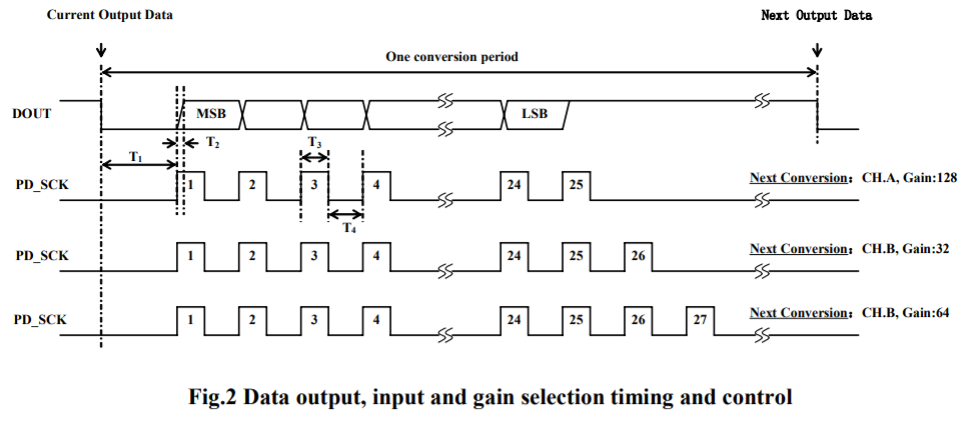


RATE : 0이니까(납땜gnd로되어있어서), Output data rate : 10Hz

PD\_SCK : Serial clock input

DOUT : data not ready(1) data ready(0)



나머지내용 : <https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf>

참고사항 : <https://learn.sparkfun.com/tutorials/getting-started-with-load-cells/all>

레퍼런스 드라이버 코드 까지 들어있다. Sck dat 2선 인터페이스

1. 적용 사진

<코드 1> // 무게 무거워지면 값 높아지나 테스트만해봄

적용사항 : TIMING도 MIN MAX보고 그에 맞게 시간만 조절해서 BIT BANGING 해줌. 음수처리는없음.

// DT : PORTA PIN0 SCK : PORTA PIN1

**#include** "HL\_system.h"

**#include** "HL\_sys\_common.h"

**#include** "HL\_gio.h"

**#include** "HL\_sci.h"

**#include** <stdio.h>

**#include** <string.h>

**#define** UART sciREG1

**char** buf[128]={0};

**void** **wait**(uint32 time){

**int** i=0;

**for**(i=0;i<time;i++)

;

}

**void** **sci\_display**(sciBASE\_t \*sci, uint8 \*text, uint32 len)

{

**while**(len--)

{

**while**((UART->FLR & 0x4)==4)

;

sciSendByte(UART, \*text++);

}

}

**unsigned** **long** **read\_count**(**void**)

{

**unsigned** **long** count=0;

**unsigned** **char** i;

gioSetBit(gioPORTA,0,1);

gioSetBit(gioPORTA,1,0);

**while**(gioGetBit(gioPORTA,0));

wait(3); // > 0.1us

**for**(i=0;i<24;i++){

gioSetBit(gioPORTA,1,1);

count=count<<1;

wait(28); //1us

gioSetBit(gioPORTA,1,0);

wait(28);

**if**(gioGetBit(gioPORTA,0)) count++;

}

gioSetBit(gioPORTA,1,1);

wait(28);

count=count^0x800000;

gioSetBit(gioPORTA,1,0);

wait(28);

**return**(count);

}

**int** **main**(**void**)

{

**unsigned** **long** data=0;

gioInit();

sciInit();

gioSetDirection(gioPORTA,0x02);

**unsigned** **int** buf\_len;

**for**(;;)

{

data=read\_count();

**sprintf**(buf,"hx711 adc result = %d\n\r\0",data);

buf\_len = **strlen**(buf);

sci\_display(UART, (uint8 \*)buf, buf\_len);

wait(2800000); //0.1s

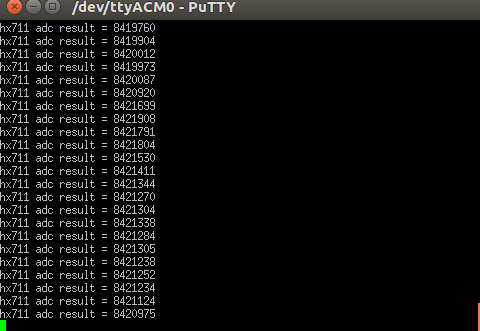
}

**return** 0;

}

/\* USER CODE BEGIN (4) \*/

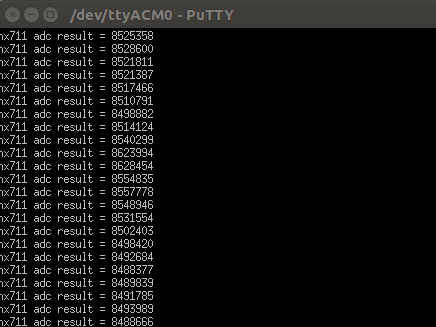
/\* USER CODE END \*/



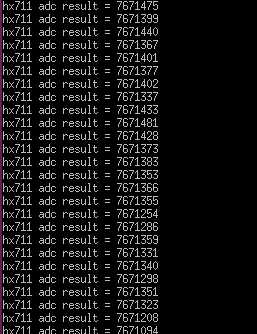
8419760 in hex = 0x8079B0 → 7079B0 → DEC → 7371184

8525358 in hex = 0x82162E → 72162E → DEC → 7476782

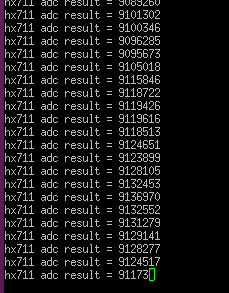
7FFFFF in dec = 8388607



<암것도없을때 >



7673000(DEC) → 7514A8(HEX) 또는 7675000(DEC) → 751C78(HEX)

<물통꽉찰때>

9115846(DEC)→ 8B18C6(HEX) 9128105(DEC)→ 8B48A9(HEX)

꽉찰때 - 비어있을때 = 162C31(HEX) 또는 1453105(DEC)

\*\*

700000 ----- 0x77FFFF 과 0x80000사이 (524287.5)----- 0xFFFFF(최대값) (1048575)