

## Váhový senzor 3 Kg YZC-131



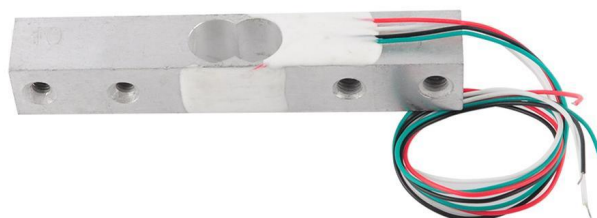
### POPIS

Senzor YZC-131 může být použit pro sestavení kuchyňské váhy nebo siloměru.

V kombinaci s vhodným AD převodníkem (např. HX711)

Základní charakteristika snímače:

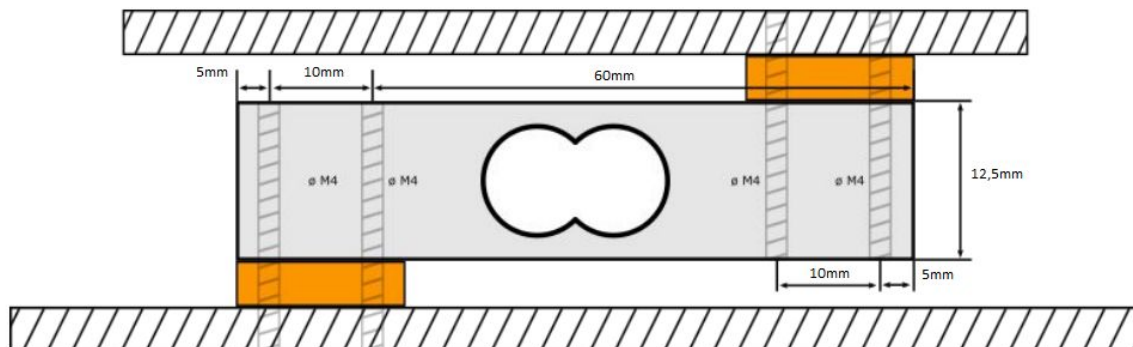
- Max. povolené zatížení 30 N
- Konfigurace: čtveřice tenzometrů zapojených do Wheatstoneova můstku
- Impedance 1 000  $\Omega$
- Snadná montáž díky otvorům se závity



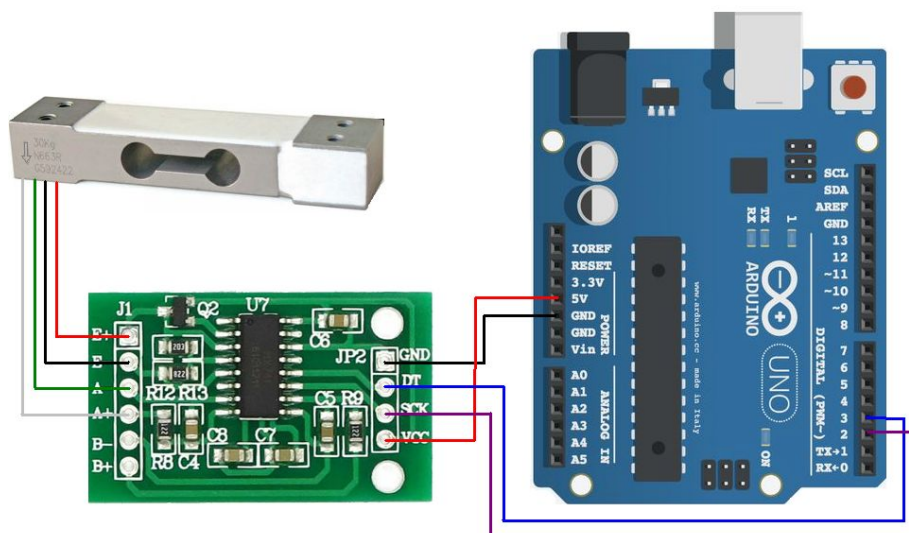
### SPECIFIKACE

|                                 |                  |                                               |                  |
|---------------------------------|------------------|-----------------------------------------------|------------------|
| Měřicí rozsah (N)               | 0–30             | Chyba linearity (% FS)                        | $\pm 0,03$       |
| Povolené přetížení (%)          | 150              | Opakovatelnost (% FS)                         | 0,03             |
| Excitační napětí (V DC)         | 5–10             | Hystereze (% FS)                              | 0,03             |
| Vstupní impedance ( $\Omega$ )  | 1 066 $\pm$ 10 % | Tečení (% FS, 5 min)                          | 0,05             |
| Výstupní impedance ( $\Omega$ ) | 1 000 $\pm$ 10 % | Stupeň krytí                                  | IP65             |
| Izolační odpor (M $\Omega$ )    | $\geq 2 000$     | Rozsah provoz. teploty ( $^{\circ}\text{C}$ ) | –20 až 40        |
| Výstupní napětí (mV/V FS)       | 1 $\pm$ 0,1      | Materiál                                      | hliník           |
| Vyvážení nuly (mV/V)            | $\pm 0,1$        | Rozměry (mm)                                  | 75 x 12,7 x 12,7 |

## Rozměry a montáž



## ZAPOJENÍ



## UKÁZKA PROGRAMU

```
// Arduino AD 24-bit převodník s 2 kanály

// piny pro připojení SCK a DT z modulu
int pSCK = 2;
int pDT = 3;

// definování různých nastavení kanálů a jejich zesílení
#define kanal_A_zesil_128 1
#define kanal_B_zesil_32 2
#define kanal_A_zesil_64 3
```

```

void setup ()
{
  // nastavení pinů modulu jako výstup a vstup
  pinMode (pSCK, OUTPUT);
  pinMode (pDT, INPUT);
  // komunikace přes sériovou linku rychlostí 9600 baud
  Serial.begin (9600);
  // probuzení modulu z power-down módu
  digitalWrite (pSCK, LOW);
  // spuštění prvního měření pro nastavení měřícího kanálu
  spusteniMereni (kanal_A_zesil_64);
}

void loop ()
{
  // výpis měření a jeho výsledku
  Serial.print ("Vysledek mereni: ");
  Serial.print (spusteniMereni (kanal_A_zesil_64));
  Serial.println (" uV");
  // pauza 0.5 s pro přehledné čtení
  delay (500);
}

// vytvoření funkce pro měření z nastaveného kanálu
long spusteniMereni (byte mericiMod)
{
  byte index;
  long vysledekMereni = 0L;
  // načtení 24-bit dat z modulu
  while (digitalRead (pDT));
  for (index = 0; index < 24; index++)
  {
    digitalWrite (pSCK, HIGH);
    vysledekMereni = (vysledekMereni << 1) | digitalRead (pDT);
    digitalWrite (pSCK, LOW);
  }
  // nastavení měřícího módu
  for (index = 0; index < mericiMod; index++)
  {
    digitalWrite (pSCK, HIGH);
    digitalWrite (pSCK, LOW);
  }
  // konverze z 24-bit dvojdoplňkového čísla
  // na 32-bit znaménkové číslo
  if (vysledekMereni >= 0x800000)
    vysledekMereni = vysledekMereni | 0xFF000000L;
  // přepočítání výsledku na mikrovolty podle zvoleného
  // kanálu a zesílení
  switch (mericiMod) {
    case 1: vysledekMereni = vysledekMereni / 128 / 2; break;
    case 2: vysledekMereni = vysledekMereni / 32 / 2; break;
    case 3: vysledekMereni = vysledekMereni / 64 / 2; break;
  }
  return vysledekMereni ;
}

```