

Váhový senzor 50 Kg

1. POPIS

Tento váhový senzor je tvořen pružným členem s tenzometrem.

Může být použit pro konstrukci domácí či průmyslové váhy.

Základní charakteristika:

- Maximální hmotnost 50 kg
- Vstupní napětí 5 až 10 V
- Tři vodiče (napětí, GND a signál)
- Vnitřní odpor 2x 1000 Ω

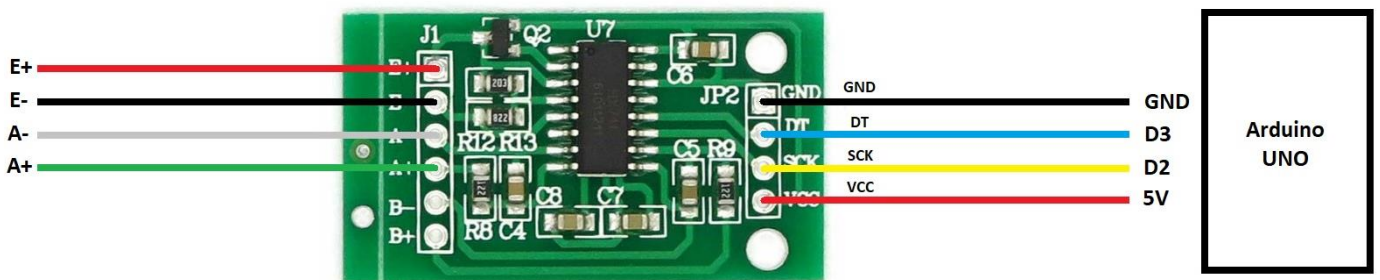
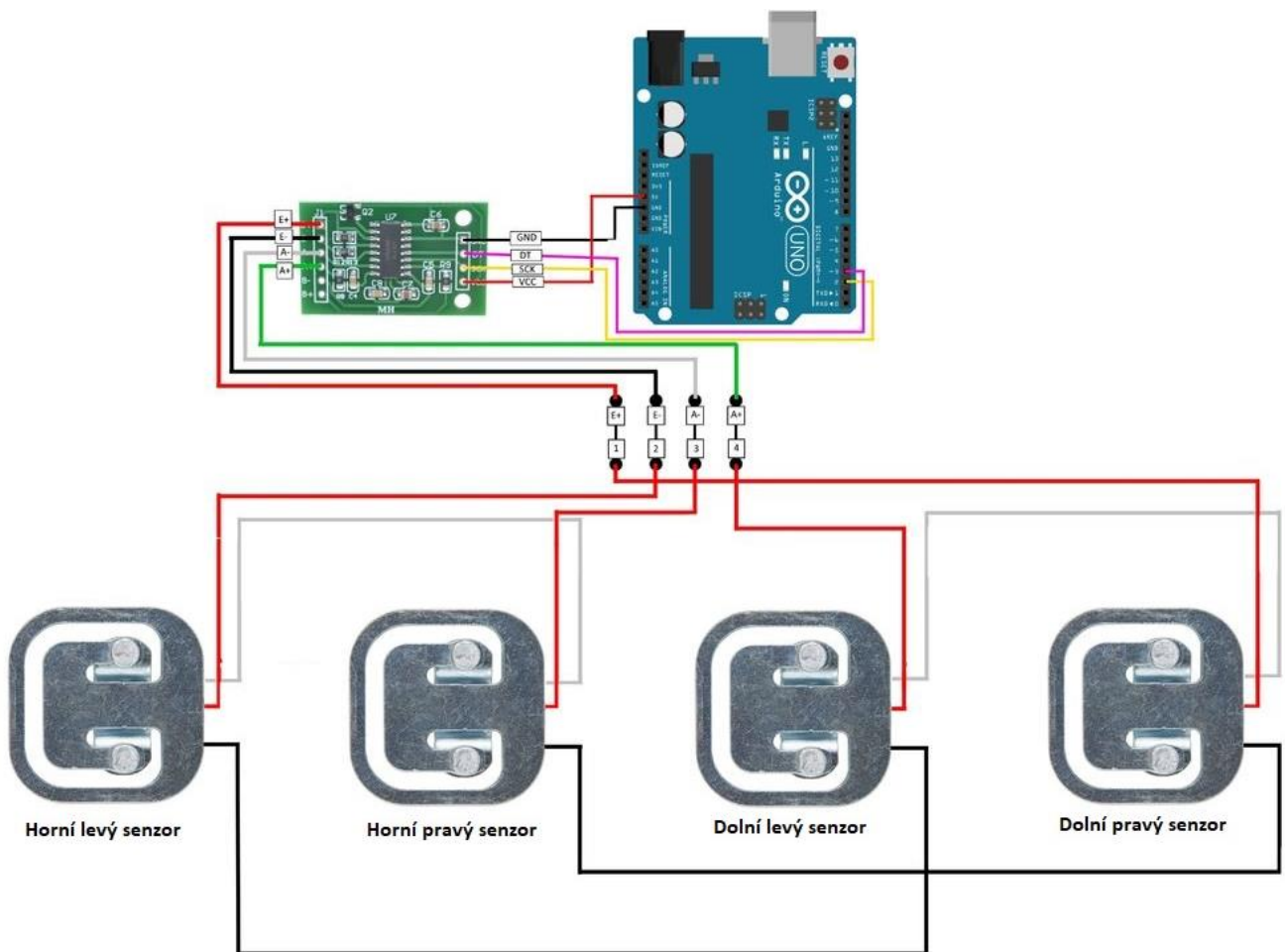


2. SPECIFIKACE

Měřicí rozsah (FS)	0 až 50 kg	Chyba linearity	0,2 % (FS)
Povolené přetížení	150 % (FS)	Opakovatelnost	0,1 % (FS)
Rozsah. vstup. napětí	5 až 10 V	Hystereze	0,2 % (FS)
Vstupní odpor	1000 Ω \pm 50 Ω	Tečení (3 min)	0,1 % (FS)
Výstupní odpor	1000 Ω \pm 50 Ω	Teplotní drift (nula)	<0,2 % (FS/10 °C)
Izolační odpor	\geq 2 000 M Ω	Teplotní vliv (rozsah)	<0,15 % (FS/10 °C)
Kombinovaná chyba	0,2 % (FS)	Změna vyvážení (1 min)	0,1 % (FS)
Výstupní napětí (FS)	1,0 \pm 0,1 mV/V	Rozsah provozní teploty	0 až 50 °C
Vyvážení nuly	\pm 0,3 mV/V	Rozměry	34 x 34 x 8 mm

3. ZAPOJENÍ

Toto zapojení zahrnuje 4x váhový senzor, AD převodník HX711 (1427807780) a Arduino Uno. Celek představuje standardní váhu, kdy je v každém rohu podstavy usazen jeden senzor.



00101
01001
00001

4. UKÁZKA PROGRAMU

Pro správnou funkci kódu je nutné stáhnout knihovnu HX711.h <https://github.com/bogde/HX711/archive/v0.1.zip>. Následující ukázky jsou převzaty z příkladů této knihovny.

Kalibrace

```
#include "HX711.h"

#define DOUT 3
#define CLK 2

HX711 scale(DOUT, CLK);

float calibration_factor = -7050; //-7050 worked for my 440lb max scale setup

void setup() {
  Serial.begin(9600);
  Serial.println("HX711 calibration sketch");
  Serial.println("Remove all weight from scale");
  Serial.println("After readings begin, place known weight on scale");
  Serial.println("Press + or a to increase calibration factor");
  Serial.println("Press - or z to decrease calibration factor");

  scale.set_scale();
  scale.tare(); //Reset the scale to 0

  long zero_factor = scale.read_average(); //Get a baseline reading
  Serial.print("Zero factor: "); //This can be used to remove the need to tare the scale. Useful in permanent scale projects.
  Serial.println(zero_factor);
}

void loop() {

  scale.set_scale(calibration_factor); //Adjust to this calibration factor

  Serial.print("Reading: ");
  Serial.print(scale.get_units(), 1);
  Serial.print(" lbs"); //Change this to kg and re-adjust the calibration factor if you follow SI units like a sane person
  Serial.print(" calibration_factor: ");
  Serial.print(calibration_factor);
  Serial.println();

  if(Serial.available())
  {
    char temp = Serial.read();
    if(temp == '+' || temp == 'a')
      calibration_factor += 10;
    else if(temp == '-' || temp == 'z')
      calibration_factor -= 10;
  }
}
```

Hlavní program

```
#include "HX711.h"

#define calibration_factor -7050.0 //This value is obtained using the SparkFun_HX711_Calibration sketch

#define DOUT 3
#define CLK 2

HX711 scale(DOUT, CLK);

void setup() {
  Serial.begin(9600);
  Serial.println("HX711 scale demo");

  scale.set_scale(calibration_factor); //This value is obtained by using the SparkFun_HX711_Calibration sketch
  scale.tare(); //Assuming there is no weight on the scale at start up, reset the scale to 0

  Serial.println("Readings:");
}

void loop() {
  Serial.print("Reading: ");
  Serial.print(scale.get_units(), 1); //scale.get_units() returns a float
  Serial.print(" lbs"); //You can change this to kg but you'll need to refactor the calibration_factor
  Serial.println();
}
```