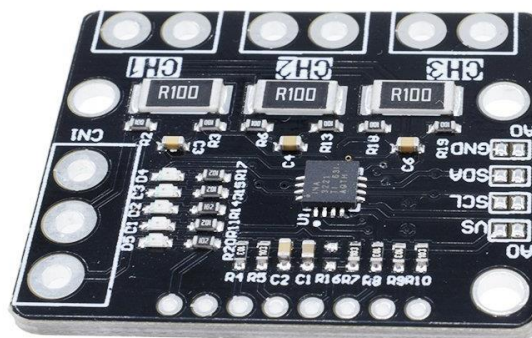


Senzor napětí a proudu s I2C sběrnici

1. POPIS

Díky tomuto modulu lze číst data reprezentující hodnoty napětí a proudu na výstupech jednotlivých kanálů. Kanály jsou celkem tři. Modul komunikuje s Arduinem pomocí rozhraní I2C. Toto zařízení je vhodné zejména pro softwarový monitoring elektrických obvodů nebo celkových sestav atd.



Základní charakteristika:

- tři výstupní měřené kanály
- až 26 VDC vstupní kanál pro zdroj
- sběrnice I2C
- indikační diody aktivity
- tři montážní otvory
- malé rozměry

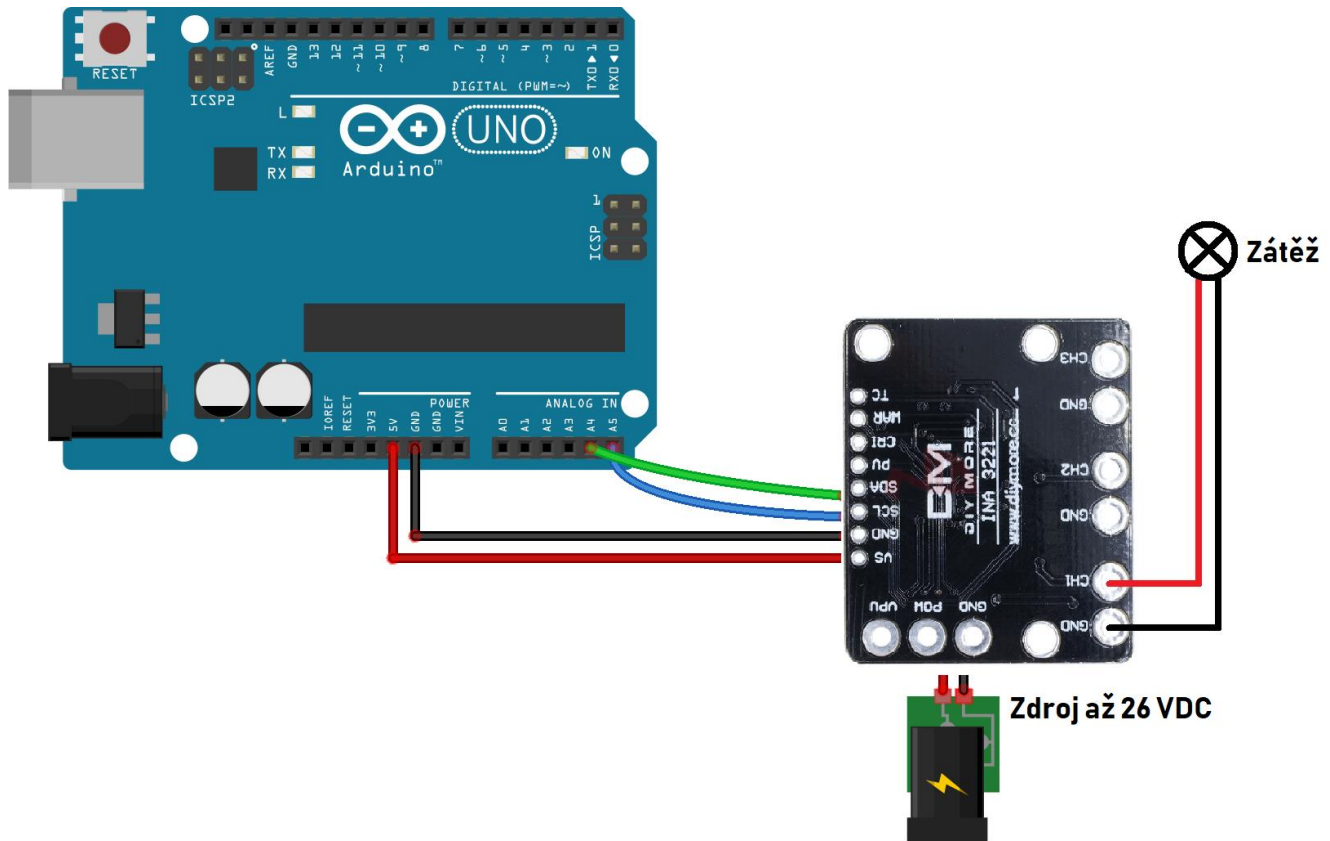
2. SPECIFIKACE

Hlavní čip	INA3221	Sběrnice	I2C
Napájecí napětí	2,7 až 5,5 V	Průměr mont. otvorů	2,5 mm
Proudový odběr	350 μ A	Rozteč mont. otvorů (mm)	32 x 18
Vstupní napětí zdroje	až 26 V	Rozměry (mm)	39 x 32 x 2,5

Více informací můžete nalézt v dokumentaci obvodu INA3221 od výrobce Texas Instruments.



3. ZAPOJENÍ



4. UKÁZKA PROGRAMU

Pro zkompilevání programu je nutné stáhnout knihovnu `SDL_Arduino_INA3221.h`, která je volně k dispozici na serveru `github.com`. Tento kód je příkladem z uvedené knihovny.

```
#include <Wire.h>
#include <SDL_Arduino_INA3221.h>

SDL_Arduino_INA3221 ina3221;
#define LIPO_BATTERY_CHANNEL 1
#define SOLAR_CELL_CHANNEL 2
#define OUTPUT_CHANNEL 3

void setup(void)
{
  Serial.begin(57600);
  Serial.println("SDA_Arduino_INA3221_Test");
  Serial.println("Measuring voltage and current with ina3221 ...");
  ina3221.begin();
}
```

```

void loop(void)
{
  Serial.println("-----");
  float shuntvoltage1 = 0;
  float busvoltage1 = 0;
  float current_mA1 = 0;
  float loadvoltage1 = 0;
  busvoltage1 = ina3221.getBusVoltage_V(LIPO_BATTERY_CHANNEL);
  shuntvoltage1 = ina3221.getShuntVoltage_mV(LIPO_BATTERY_CHANNEL);
  current_mA1 = -ina3221.getCurrent_mA(LIPO_BATTERY_CHANNEL); // minus is to get the "sense" right. - means the
  battery is charging, + that it is discharging
  loadvoltage1 = busvoltage1 + (shuntvoltage1 / 1000);

  Serial.print("LIPO_Battery Bus Voltage: "); Serial.print(busvoltage1); Serial.println(" V");
  Serial.print("LIPO_Battery Shunt Voltage: "); Serial.print(shuntvoltage1); Serial.println(" mV");
  Serial.print("LIPO_Battery Load Voltage: "); Serial.print(loadvoltage1); Serial.println(" V");
  Serial.print("LIPO_Battery Current 1:   "); Serial.print(current_mA1); Serial.println(" mA");
  Serial.println("");

  float shuntvoltage2 = 0;
  float busvoltage2 = 0;
  float current_mA2 = 0;
  float loadvoltage2 = 0;

  busvoltage2 = ina3221.getBusVoltage_V(SOLAR_CELL_CHANNEL);
  shuntvoltage2 = ina3221.getShuntVoltage_mV(SOLAR_CELL_CHANNEL);
  current_mA2 = -ina3221.getCurrent_mA(SOLAR_CELL_CHANNEL);
  loadvoltage2 = busvoltage2 + (shuntvoltage2 / 1000);

  Serial.print("Solar Cell Bus Voltage 2: "); Serial.print(busvoltage2); Serial.println(" V");
  Serial.print("Solar Cell Shunt Voltage 2: "); Serial.print(shuntvoltage2); Serial.println(" mV");
  Serial.print("Solar Cell Load Voltage 2: "); Serial.print(loadvoltage2); Serial.println(" V");
  Serial.print("Solar Cell Current 2:   "); Serial.print(current_mA2); Serial.println(" mA");
  Serial.println("");

  float shuntvoltage3 = 0;
  float busvoltage3 = 0;
  float current_mA3 = 0;
  float loadvoltage3 = 0;

  busvoltage3 = ina3221.getBusVoltage_V(OUTPUT_CHANNEL);
  shuntvoltage3 = ina3221.getShuntVoltage_mV(OUTPUT_CHANNEL);
  current_mA3 = ina3221.getCurrent_mA(OUTPUT_CHANNEL);
  loadvoltage3 = busvoltage3 + (shuntvoltage3 / 1000);

  Serial.print("Output Bus Voltage 3: "); Serial.print(busvoltage3); Serial.println(" V");
  Serial.print("Output Shunt Voltage 3: "); Serial.print(shuntvoltage3); Serial.println(" mV");
  Serial.print("Output Load Voltage 3: "); Serial.print(loadvoltage3); Serial.println(" V");
  Serial.print("Output Current 3:   "); Serial.print(current_mA3); Serial.println(" mA");
  Serial.println("");

  delay(2000);
}

```