

- Elettronica Open Source - <https://it.emcelettronica.com> -

INA219: il sensore di corrente/tensione per monitorare un motore al lavoro

Posted By *Adriano Gandolfo* On 14 ottobre 2014 @ 5:00 In Arduino, Makers & Progetti Fai Da Te | [26 Comments](#)



Il sensore e il display qui utilizzati erano già stati presentati nell'articolo "L'Evaluation Kit XMC 2GO di Infineon come scheda sensori per i robot ROBI [1]".

Il sensore tipo INA219 prodotto dalla Texas Instruments, permette di misurare la tensione e la corrente fornite da una fonte di energia, nel nostro caso una batteria, comunicando i dati mediante un'interfaccia I2C.

L'integrato controlla sia la caduta shunt sia la tensione di alimentazione, con tempi di conversione e filtraggio programmabili. I dati vengono mostrati su un piccolo display di tipo OLED, anch'esso connesso tramite l'interfaccia I2C.

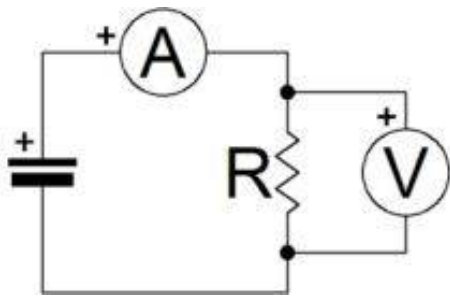
In quest'articolo, ne vedremo l'utilizzo per controllare il funzionamento di un piccolo motore elettrico. Per la sua gestione, si è utilizzato un modulo Arduino, rendendo il dispositivo facile da programmare usufruendo delle librerie già disponibili.

Un po' di teoria

Per ricavare la tensione della batteria e la corrente assorbita dal carico, occorre far riferimento alla legge di Ohm (dal nome del fisico tedesco George Simon Alfred Ohm): è una legge fisica che mette in relazione la corrente con la tensione in un resistore. Si avrà quindi una relazione di semplice proporzionalità che può essere scritta nel seguente modo:

$$V = R \cdot I$$

dove V è la tensione ai capi del resistore (misurata tramite un multimetro impostato come Voltmetro), I è la corrente che lo attraversa (misurata tramite un multimetro impostato come Amperometro), mentre R è il valore ohmico di resistenza.



Possiamo quindi calcolare il valore della tensione V ai capi del resistore.

In modo analogo, potremmo ricavare altre incognite, per esempio conoscendo la corrente e la tensione, R risulta da

$$R = V/I$$

Oppure conoscendo la tensione e la resistenza possiamo calcolare la corrente:

$$I = V/R$$

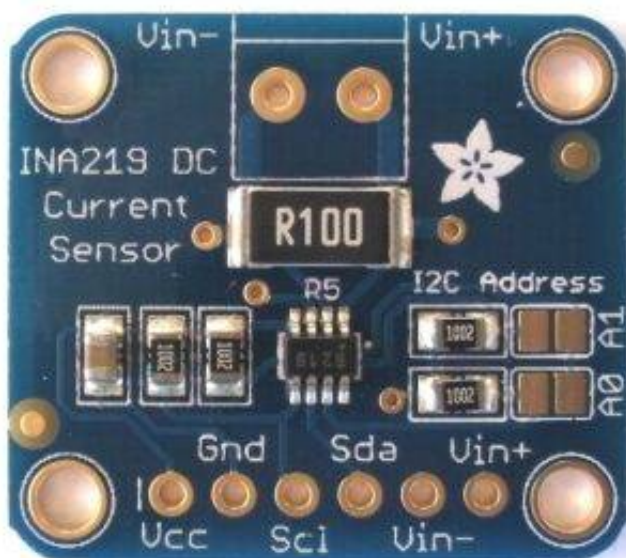
La potenza P si misura in watt (W) ed è il prodotto della tensione V espressa in volt (V) per la corrente I espressa in ampere (A):

$$P = V \cdot I$$

Il Sensore INA219B

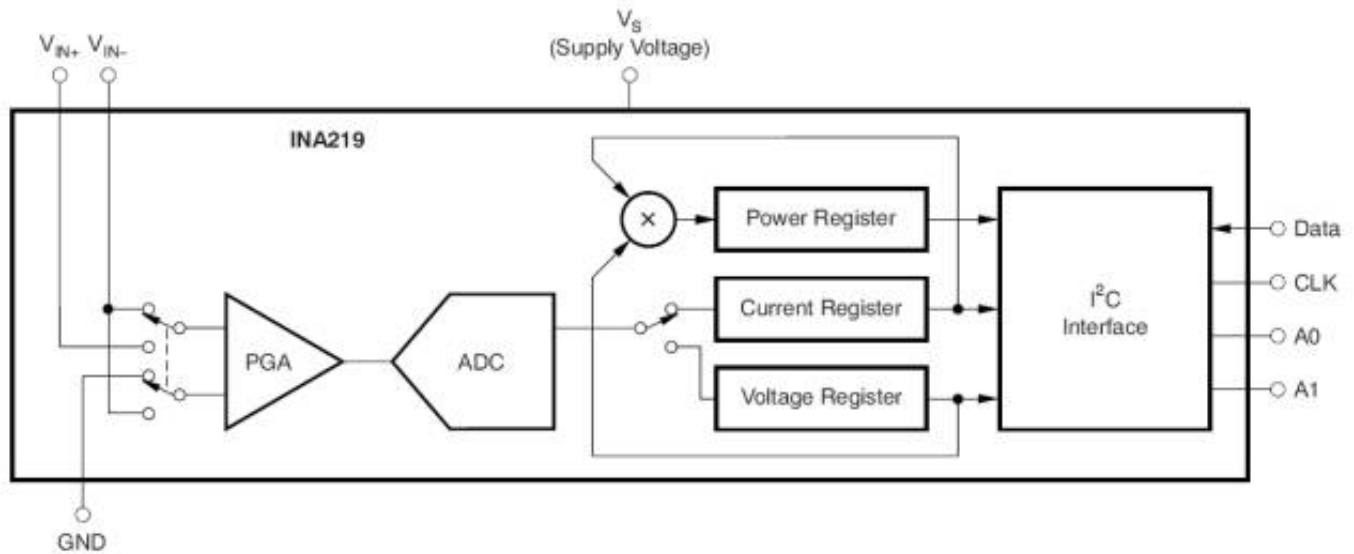
Come abbiamo visto, per ricavare i dati richiesti, occorrerebbero due multimetri, ma utilizzando la breakout board ^[2] prodotta dalla Adafruit che misura solamente 23x20 mm, si risolveranno tutti i problemi.

Su di essa sono già presenti tutti i componenti necessari per il funzionamento dell'integrato INA219B ^[3] prodotto dalla Texas Instruments.



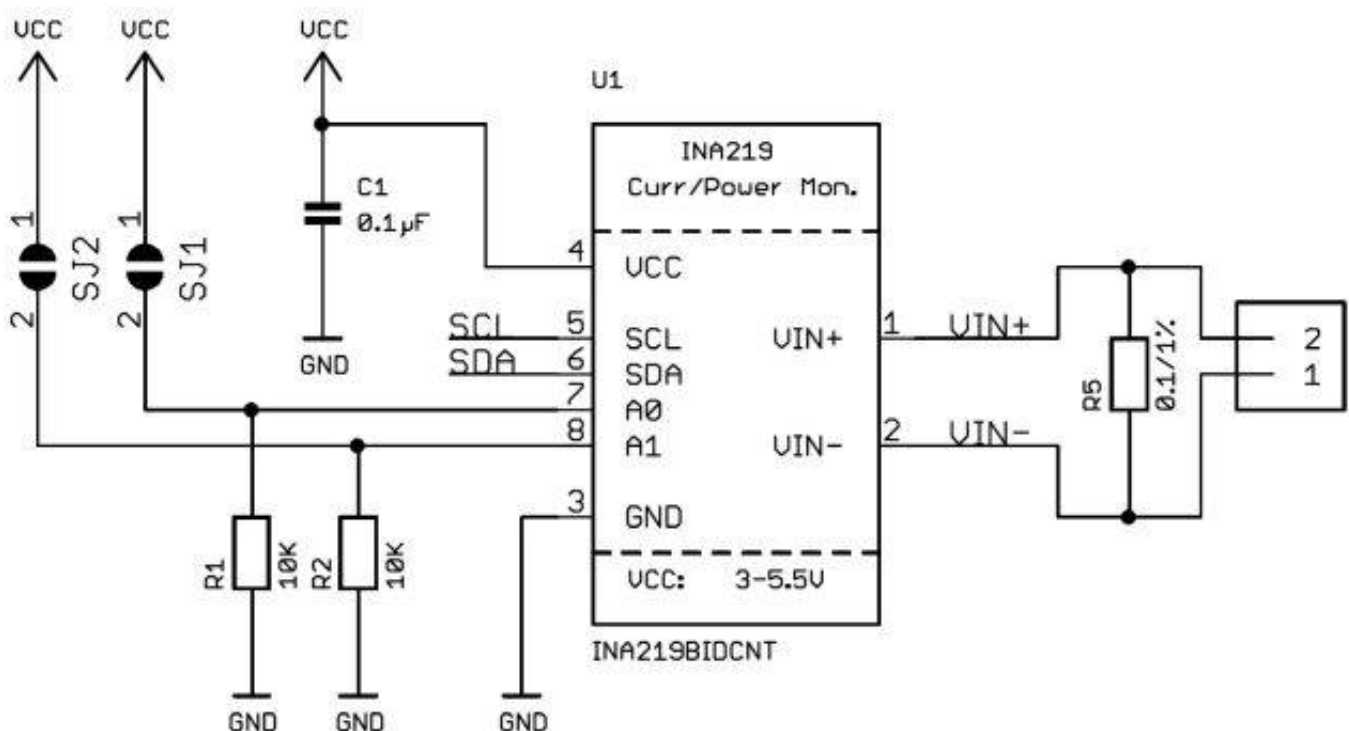
L'integrato si presenta in package SOT23-8 e può essere alimentato con una tensione V_S

compresa tra 3 e 5V permettendo di misurare la tensione DC, con un valore massimo di +26V con 1 % di precisione trasmettendo i dati al processore tramite il protocollo I2C [4]; è un'interfaccia digitale molto diffusa, utilizzata per la comunicazione a corto raggio tra differenti tipi di circuiti integrati, basata su un bus composto di due soli fili.



Nello schema a blocchi è possibile vedere un amplificatore di precisione che misura la tensione ai capi della resistenza shunt di rilevamento, che nella nostra scheda ha un valore di 0,1 ohm 1 %.

Poiché la differenza massima d'ingresso dell'amplificatore è $\pm 320\text{mV}$ significa che può misurare fino a ± 3.2 Ampere.



Con un convertitore ADC interno a 12 bit, la risoluzione è di $\pm 0.8\text{mA}$, nella gamma $\pm 3.2\text{A}$, mentre con il guadagno interno impostato al minimo, la corrente massima misurabile sarà di $\pm 400\text{mA}$ con la risoluzione di 0,1 mA.

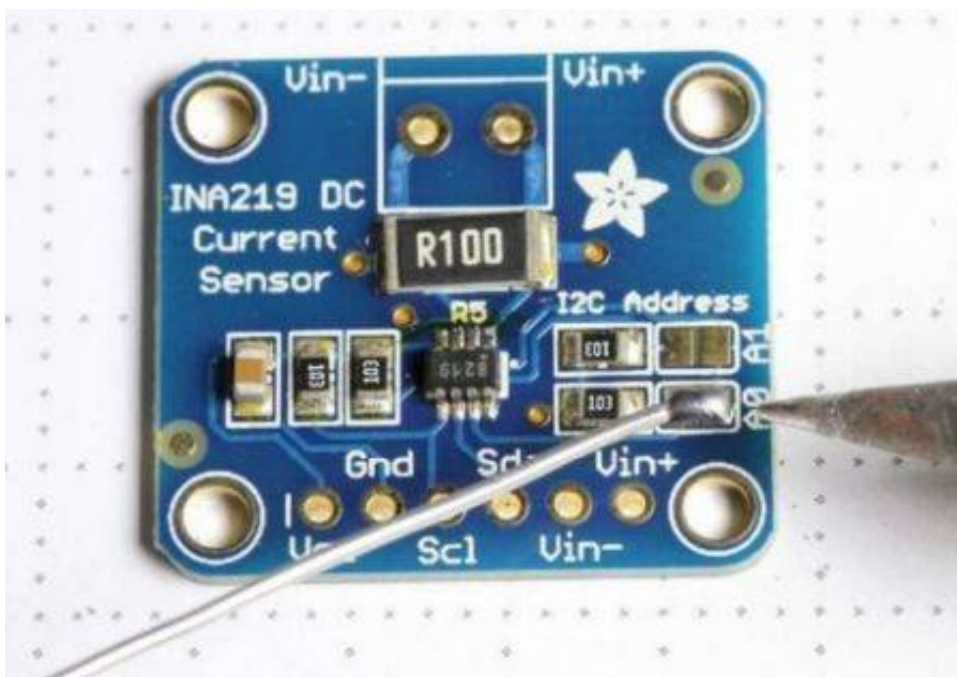
Il modulo ha un pin head di sei pin (in modo da poter collegare facilmente questo sensore a una basetta), così come una morsettiera a vite con un passo di 3,5 mm in modo da poter collegare il carico.

Con riferimento allo schema elettrico, la parte principale del circuito è quella che ruota intorno al sensore **INA219**, al pin 1 e 2, VIN+ e VIN-, a cui è collegata la resistenza di misura dal valore di $0,1\Omega$ 1%.

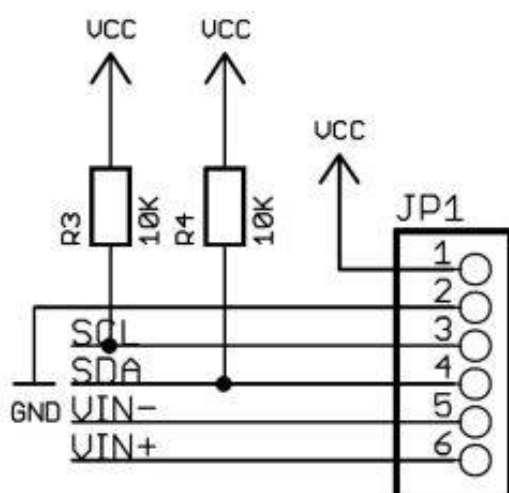
L'alimentazione, che può variare da un minimo di 3 a un massimo di 5V, è fornita al pin 4 (terminale positivo) e al pin 3 (terminale negativo), dove è presente il condensatore di disaccoppiamento da 100nF.

Tramite il livello logico fornito ai pin 7 e 8 (A0 e A1), viene scelto l'indirizzo della scheda che può essere: 0x40, 0x41, 0x44, 0x45.

Per modificarlo, occorre eventualmente, depositare una goccia di stagno sui ponticelli **SJ1** o **SJ2**; nel caso non sia selezionato, il valore di default sarà 0x40.



I pin dell'interfaccia I2C (acronimo di Inter Integrated Circuit) sono rappresentati dal pin 5 (**SCL** - **S**erial **C**lock **L**ine) per il clock e 6 (**SDA** - **S**erial **D**ata) per i dati.



Sul connettore d'uscita JP1, sono riportati tutti i segnali per l'interfacciamento, sono inoltre visibili i due resistori di pull-up da 10kΩ della linea I2C.

Descrizione del display OLED

Per mostrare i dati letti dal sensore, si è utilizzato un piccolo display monocromo 128x32 I2C OLED ^[5](dimensioni del modulo 20 x 35mm) prodotto anch'esso dalla Adafruit, che presenta una diagonale di 1", ma è molto leggibile dato l'alto contrasto di un display OLED.



Il display presenta 128x32 singoli pixel bianchi OLED, ciascuno è acceso o spento dal chip controller, non è richiesta alcuna retroilluminazione riducendo l'assorbimento di potenza necessaria.

L'integrato pilota è il modello SSD1306 che comunica tramite il protocollo I2C.

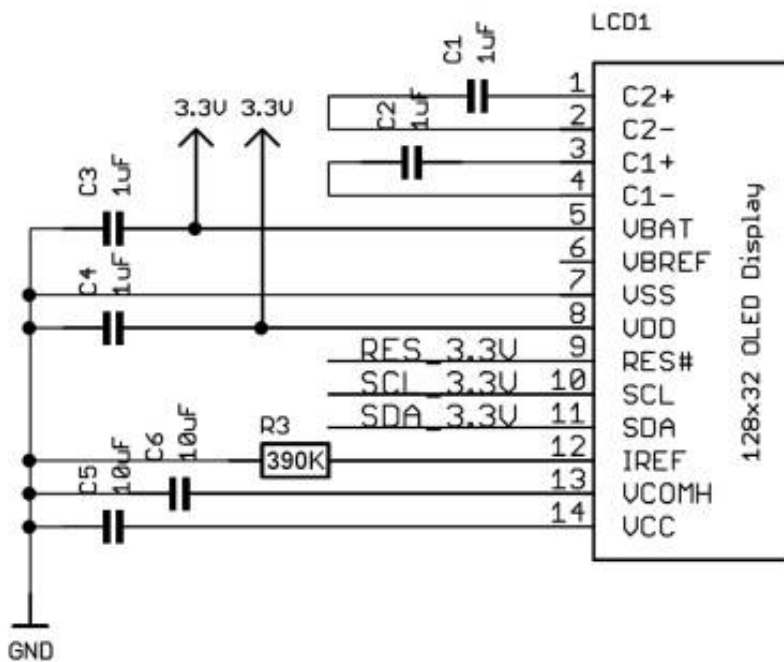
Sono necessari solo 3 pin per comunicare con il chip sul display OLED, due dei quali sono i pin I2C dati / clock.

Il display OLED e il driver, richiedono un'alimentazione di 3,3 V e pari valore hanno anche i livelli logici per la comunicazione.

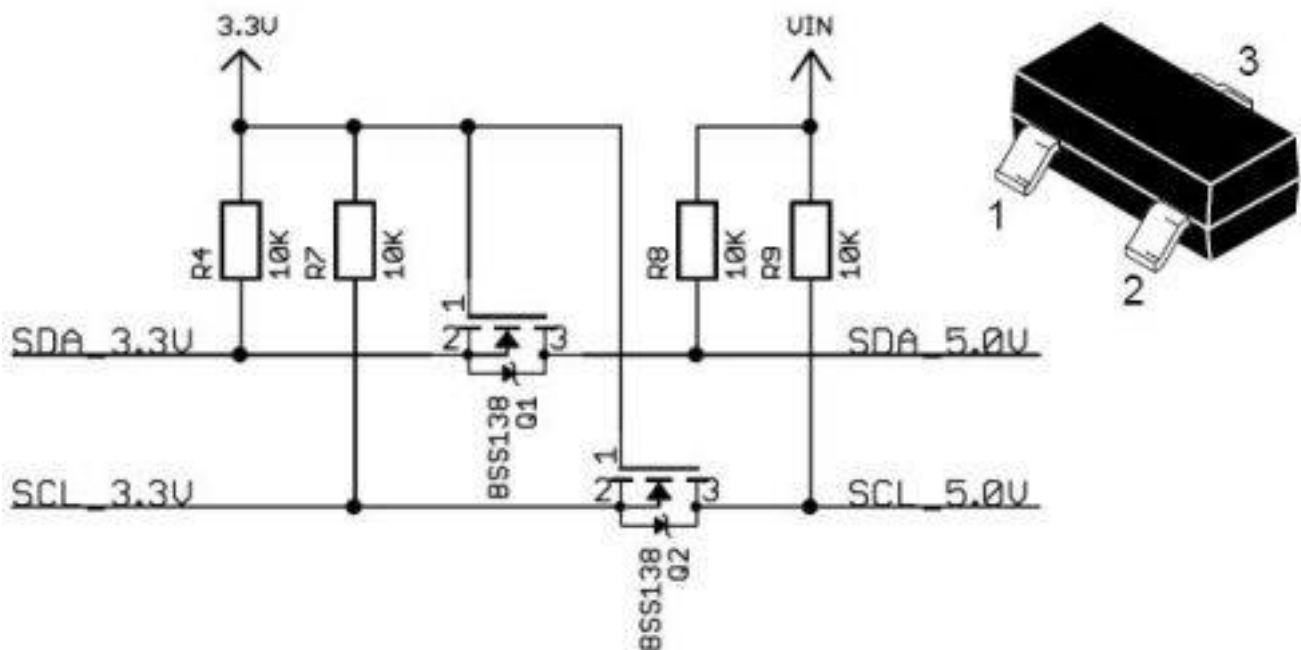
I requisiti di alimentazione dipendono da quanto del display è acceso, ma in media il display assorbe circa 20mA dalla rete di alimentazione 3.3V.

Per l'utilizzo, è necessario un microcontrollore con più di 512 byte di RAM, in quanto la memoria del display deve essere bufferizzata.

Esaminando i vari blocchi che formano il circuito, abbiamo la sezione del display/processore, basata sull'integrato **SSD1306** ^[6] che è un single-chip CMOS driver per display tipo OLED / PLED (Organic Light Emitting Diode, Polymer light-emitting diodes).



La sezione del circuito permette di traslare i livelli logici tra i valori di tensione di 3.3 e i 5V.



In questo caso, è stata utilizzata una tipologia di circuito che usa dei MOSFET e che presenta i seguenti vantaggi:

è economico;

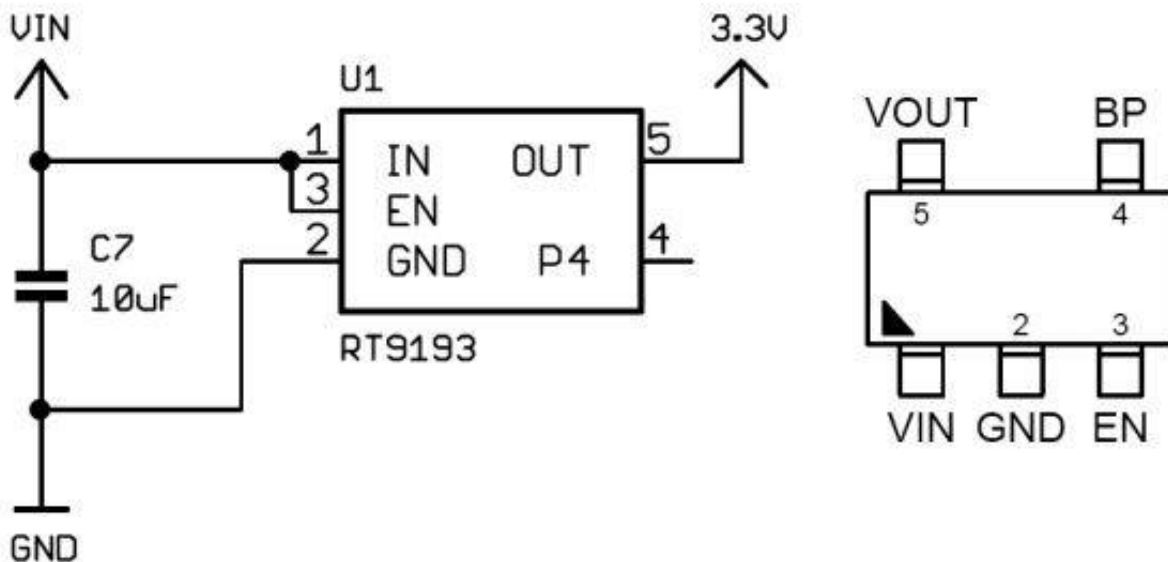
sono utilizzati componenti generici, in questo caso dei MOSFET tipo **BSS138** [7];

è possibile isolare le sezioni del bus semplicemente togliendo l'alimentazione ai MOSFETs interessati;

viene fornita una protezione della sezione a basso livello contro eventuali spikes (con spike ci si riferisce ad una rapida variazione di tensione) nella sezione ad alto livello.

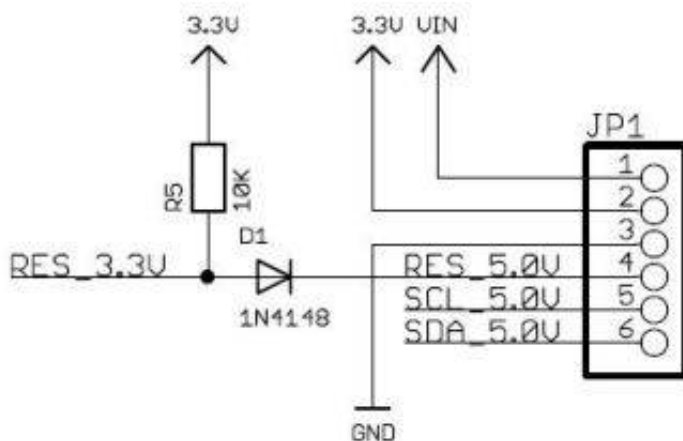
Per rendere più semplice il collegamento, sul piccolo modulo, è stata aggiunta una sezione alimentatrice formata dal regolatore tipo **RT9193** [8] di tipo LDO Ultra-Low Noise, Ultra-

Fast che fornisce in uscita 3,3V con 300mA di corrente.



Per ultimo abbiamo il connettore d'ingresso, dove sono presenti pin di alimentazione e d'interfaccia I2C rappresentati dal pin SCL, SDA più il pin RES.

Per quanto riguarda l'indirizzo I2C della scheda, questo è del tipo a 7-bit ed è fissato al valore di 0x3C.



Librerie di gestione

Gestire il modulo sensore e il modulo display sarebbe piuttosto complesso, fortunatamente la stessa Adafruit ha messo a disposizione delle apposite librerie che semplificano la scrittura del programma, possono essere scaricate dal sito del produttore, in particolare sono necessarie:

[La libreria di gestione del modulo INA 219](#) ^[9];

[La libreria di gestione del modulo OLED](#) ^[10];

[La librerie grafiche per il modulo OLED](#) ^[11].

Per il giusto funzionamento del display, occorrerà modificare all'interno del file

Adafruit_SSD1306.h della libreria, la linea che definisce la dimensione del display per cui si dovranno cercare le linee:

```
#define SSD1306_128_64
//#define SSD1306_128_32
```

E di dovrà modificarle in:

```
//#define SSD1306_128_64
#define SSD1306_128_32
```

Uso della libreria

Per utilizzare il sensore, occorre installare l'apposita libreria, espandendo il file zip nella cartella Libraries nella cartella Arduino, la si dovrà rinominare in **Adafruit_INA219**. Si dovranno chiudere tutte le istanze dell'IDE, per poi riaprire una, in modo che l'IDE riconosca la nuova libreria.

Funzioni di assegnazione e inizializzazione

```
Adafruit_INA219 (uint8_t addr = INA219_ADDRESS);
```

Questa funzione crea l'oggetto Adafruit_INA219 con relativo indirizzo, se questo non è specificato, è utilizzato quello di default (0x40). Se invece si utilizza più di un modulo INA219, occorrerà definire per ognuno, un proprio indirizzo tra quelli disponibili.

```
void begin (void);
```

Questa istruzione inizializza la comunicazione I2C con il dispositivo Adafruit_INA219 utilizzando i valori di configurazione predefiniti.

Funzioni di lettura del sensore

```
float getBusVoltage_V (void);
```

Legge la tensione tra GND e V. Questa è la tensione totale vista dal circuito in esame. (Tensione di alimentazione - tensione shunt). Il valore di ritorno è in Volt.

```
float getShuntVoltage_mV (void);
```

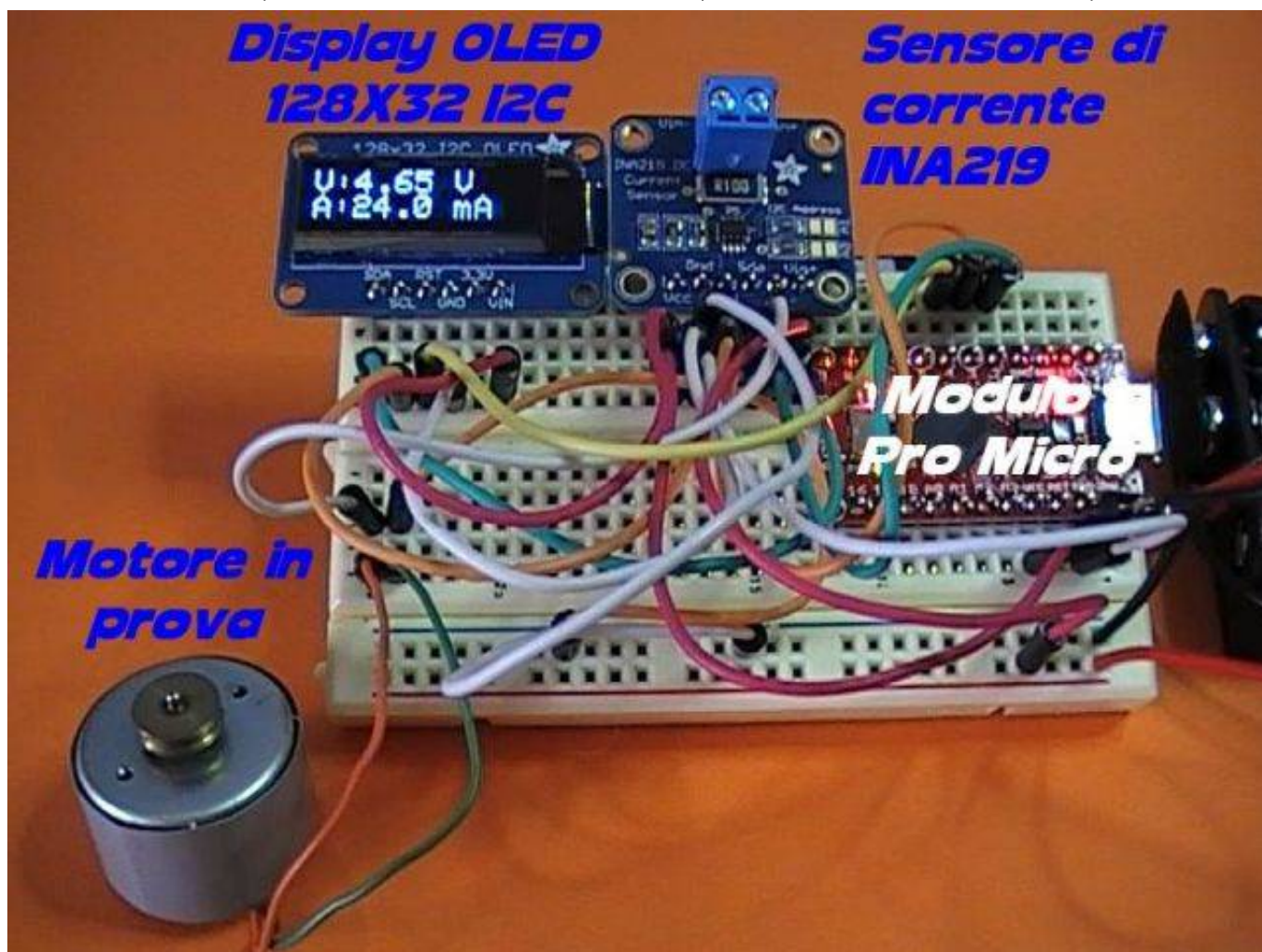
Legge la tensione tra V e V +. Questa è la caduta di tensione misurata ai capi del resistore shunt. Il valore di ritorno è in millivolt.

```
float getCurrent_mA (void);
```

Legge l'attuale corrente ricavata tramite la legge Ohm dalla tensione shunt misurata. Il valore di ritorno è in milliampere.

Il collegamento

Per testare il dispositivo, si è realizzato un semplice circuito, in cui il sensore è utilizzato per la lettura della tensione di alimentazione, la corrente e la potenza assorbita da un piccolo motore elettrico.



Nell'esempio riportato nella foto, si è utilizzato un **Micro Pro** [12] prodotto dalla SparkFun, che è una piccola scheda che monta un processore **ATmega32U4**, questo fornisce la connettività on-board, per cui non è necessario utilizzare un'interfaccia USB esterna.



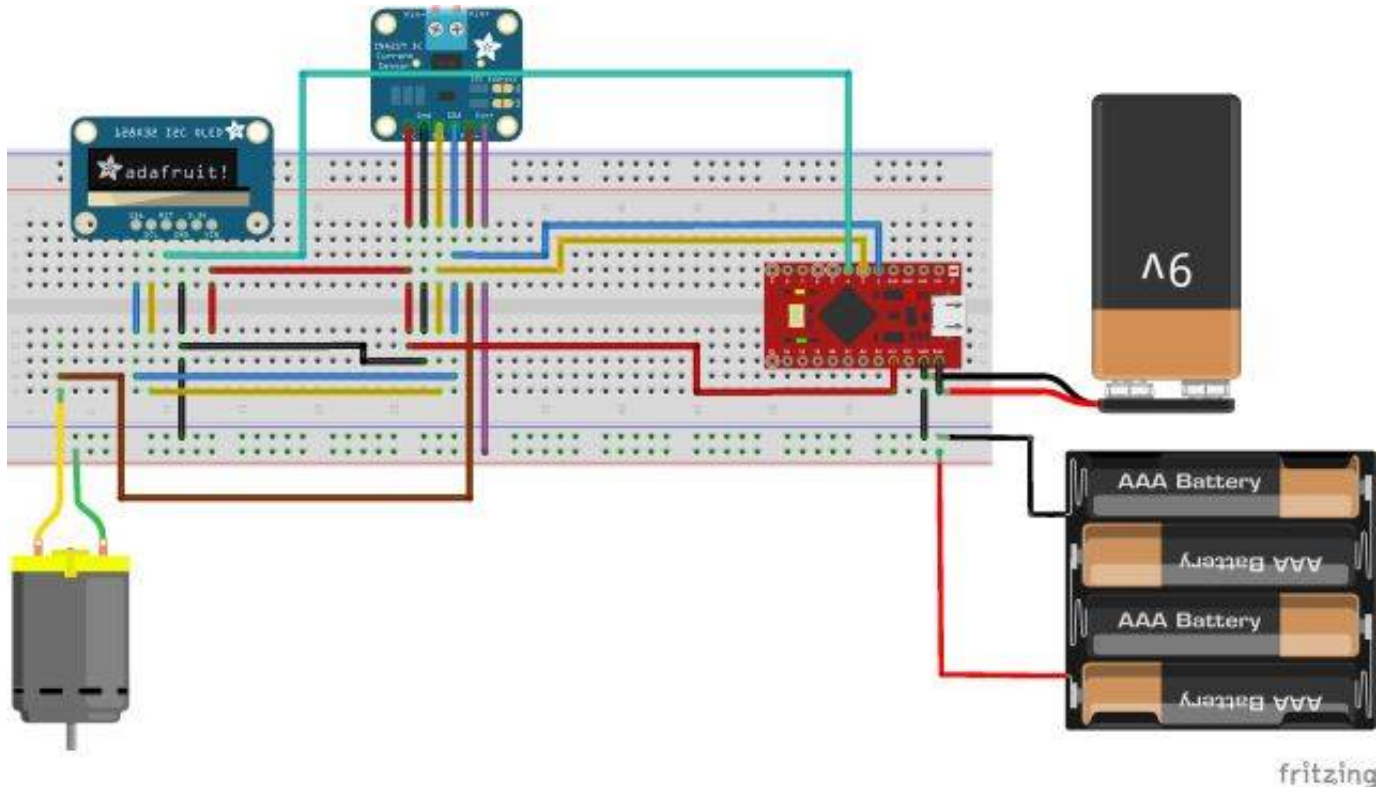
In alternativa, sarà possibile utilizzare una qualunque **scheda Arduino**, tipo la Arduino UNO.

Come abbiamo visto, i due moduli hanno un diverso indirizzo, che vale **0x40** per il sensore

e **0x3C** per il display.

Se si utilizza la scheda Arduino UNO, i pin dell'interfaccia I2C sono rappresentati dal pin A4 (SDA) e A5 (SCL).

Abbiamo ancora bisogno di connettere il pin RES del display ad un pin IN/OUT digitale, nel nostro caso al PIN 4, anche se potrà essere diverso, variando il valore indicato nel programma.



Nel caso si utilizzasse una scheda Arduino UNO R3, i pin I2C sono stati replicati a lato dei pin digitali come riportato in figura.



Come abbiamo visto le caratteristiche di alimentazione, sia del modulo sensore sia del

display, sono simili e possono variare tra i 3 e i 5 V, permettendo di utilizzarli con una grande varietà di schede di controllo.

Per questo, oltre che a schede tipo Arduino UNO, Arduino MEGA2560 e Arduino Leonardo, potranno essere collegati alla scheda Arduino DUE che lavora con logica a 3,3 V.

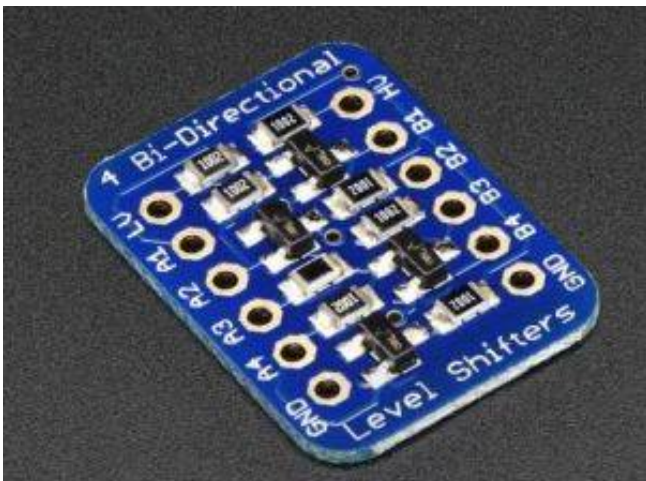
Utilizzando entrambi i dispositivi, su bus I2C (che usa solo due pin), sono lasciate libere molte porte digitali utilizzabili per altri scopi.

Sarà altresì, possibile inserire altri dispositivi dotati di bus I2C, curando solo che abbiano tutti un indirizzo diverso [13].

Va verificato inoltre, che i vari dispositivi lavorino tutti con la medesima tensione; in caso contrario, occorrerà inserire un traslatore di livello.

Uno di questi potrebbe essere il convertitore di livello logico a 4 canali [14] bidirezionale della Adafruit.

In questo modo, sarà possibile avere moduli I2C con livelli logici (3,3-5V) diversi, presenti sulla stessa linea del bus.



Il programma di gestione

Il programma di gestione è presente nell'allegato [15] ed è molto semplice, il listato è sufficientemente commentato per comprenderne il funzionamento.

Dopo il caricamento delle varie librerie, l'inizializzazione del sensore e del display, il programma legge ciclicamente i dati rilevati dal sensore, mostrandoli sul display OLED.

```
//Caricamento delle librerie di comunicazione
#include <SPI.h>
#include <Wire.h>
//Caricamento delle librerie gestione display
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define OLED_RESET 4 //Definizione pin reset display
Adafruit_SSD1306 display(OLED_RESET);
//Caricamento delle librerie gestione sensore
```

```
#include <Adafruit_INA219.h>
//Costruisce un'istanza della Adafruit_INA219
//Se non viene specificato alcun indirizzo, viene utilizzato
//quello di default (0x40). Se si collegano più modulo INA219
//occorre indicare il suo indirizzo per esempio
// Adafruit_INA219 ina219_B(0x41)
Adafruit_INA219 ina219;

void setup() {
  //Impostazione modulo INA219
  uint32_t currentFrequency;
  //Inizializza la comunicazione I2C con il dispositivo Adafruit_INA219
  // utilizzando i valori di configurazione predefiniti.
  ina219.begin();
  // Impostazione modulo display con indirizzo I2C addr 0x3C
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  // Cancellazione schermo e buffer
  display.clearDisplay();
  // Messaggio iniziale
  display.setTextSize(1); //Definizione della grandezza del carattere
  display.setTextColor(WHITE); //Impostazione del colore del testo
  display.setCursor(0,0); //Posizionamento del cursore
  display.println("www.adrirobot.it");
  display.setTextColor(BLACK, WHITE); // Inverte il colore del testo
  display.println("Voltmetro digitale");
  display.display(); // Stampa sul display
  delay(2000); //Pausa per lettura dei valori
}

void loop()
{
  //Definizione variabili
  float shuntvoltage = 0;
  float busvoltage = 0;
  float current_mA = 0;
  float loadvoltage = 0;
  //Acquisizione dei valori dal sensore
  shuntvoltage = ina219.getShuntVoltage_mV();
  //Legge la tensione tra V e V +. Questa è la caduta di tensione
  //misurata ai capi del resistore shunt. Il valore è espresso in mV
  busvoltage = ina219.getBusVoltage_V();
  //Legge la tensione tra GND e V. Questa è la tensione totale vista
```

```
//dal circuito in esame. (Tensione di alimentazione - tensione shunt).
//Il valore è in Volt.
current_mA = ina219.getCurrent_mA();
//Legge la corrente, ricavata tramite la legge di Ohm
//dalla tensione shunt misurata. Il valore è in mA.
loadvoltage = busvoltage + (shuntvoltage / 1000);
//Calcola la tensione sul carico
//Stampa dei valori misurati sul display
display.clearDisplay(); // cancellazione dello schermo
display.setTextSize(2); //Definizione della grandezza del testo
display.setTextColor(WHITE); //Impostazione del colore del testo
display.setCursor(0,0); //Posizionamento del cursore
// Definizione dei valori con relative unità di misura
display.print("V:");
display.print(loadvoltage, 2);
display.println(" V");
display.print("A:");
display.print(current_mA,1);
display.println(" mA");
display.display(); // Stampa dei valori sul display
delay(2000); //Pausa per lettura dei valori
display.clearDisplay(); // cancellazione dello schermo
display.setCursor(0,0); //Posizionamento del cursore
// Mostra sul display il valore della potenza assorbita
// ottenuta moltiplicando la corrente per la tensione sul carico
display.println("Potenza");
display.print((current_mA*loadvoltage)/1000);
display.print(" W");
display.display(); // Stampa dei valori sul display
delay(2000); //Pausa per lettura dei valori
}
```

Immagini del display



Immagine iniziale



Lettura tensione e corrente



Letture potenza assorbita

Filmato illustrativo

Nel filmato, potrete vedere il sensore e il display in funzione; il test è stato realizzato con un Micro Pro, ma può essere utilizzato un qualunque modello di Arduino.

Conclusioni

Quanto presentato in quest'articolo è un tipico utilizzo del sensore INA 219, che potrà essere inserito in un sistema di controllo più complesso e completo.

I valori letti, potranno essere utilizzati per monitorare eventualmente i consumi di corrente e produrre allarmi o avvisi in caso questi superino valori preimpostati, oppure far sì che, in via automatica, sia disconnesso il carico, per evitare guasti.

Si potrà implementare la funzione che avverta, nel caso di alimentazione a batteria, che il valore fornito sia troppo basso.

Abbiamo anche visto l'utilizzo del display OLED, che potrà essere utilizzato in altri progetti in sostituzione dei normali display LCD 16x2.

Starà solo all'inventiva del progettista utilizzare al meglio il dispositivo.

Article printed from Elettronica Open Source: <https://it.emcelettronica.com>

URL to article: <https://it.emcelettronica.com/ina219-sensore-di-correntetensione-monitorare-motore-al-lavoro>

URLs in this post:

[1] L'Evaluation Kit XMC 2GO di Infineon come scheda sensori per i robot ROBI: <https://it.emcelettronica.com/levaluation-kit-xmc-2go-di-infineon-come-scheda-sensori-i-robot-robi>

[2] breakout board: <http://www.adafruit.com/products/904>

- [3] **INA219B** : <http://www.ti.com/lit/ds/symlink/ina219.pdf>
- [4] il protocollo I2C: <https://it.emcelettronica.com/i2c-viaggio-alla-riscoperta-del-protocollo-tra-basi-funzioni-avanzate>
- [5] display monocromo 128x32 I2C OLED : <http://www.adafruit.com/products/931>
- [6] **SSD1306**: <https://www.adafruit.com/datasheets/SSD1306.pdf>
- [7] **BSS138**: <https://www.fairchildsemi.com/datasheets/BS/BSS138.pdf>
- [8] **RT9193**: http://www.richtek.com/download_ds.jsp?p=RT9193
- [9] La libreria di gestione del modulo INA 219:
https://github.com/adafruit/Adafruit_INA219
- [10] La libreria di gestione del modulo OLED:
https://github.com/adafruit/Adafruit_SSD1306
- [11] La librerie grafiche per il modulo OLED: <https://github.com/adafruit/Adafruit-GFX-Library>
- [12] Micro Pro: <http://www.sparkfun.com/products/12640>
- [13] un indirizzo diverso: <https://it.emcelettronica.com/i2c-risoluzione-delle-problematiche-comuni>
- [14] convertitore di livello logico a 4 canali: <http://www.adafruit.com/products/757>
- [15] presente nell'allegato:
https://it.emcelettronica.com/files/lettura_dati_ina219.zip

Copyright © 2017 Elettronica Open Source. All rights reserved.