Introduzione

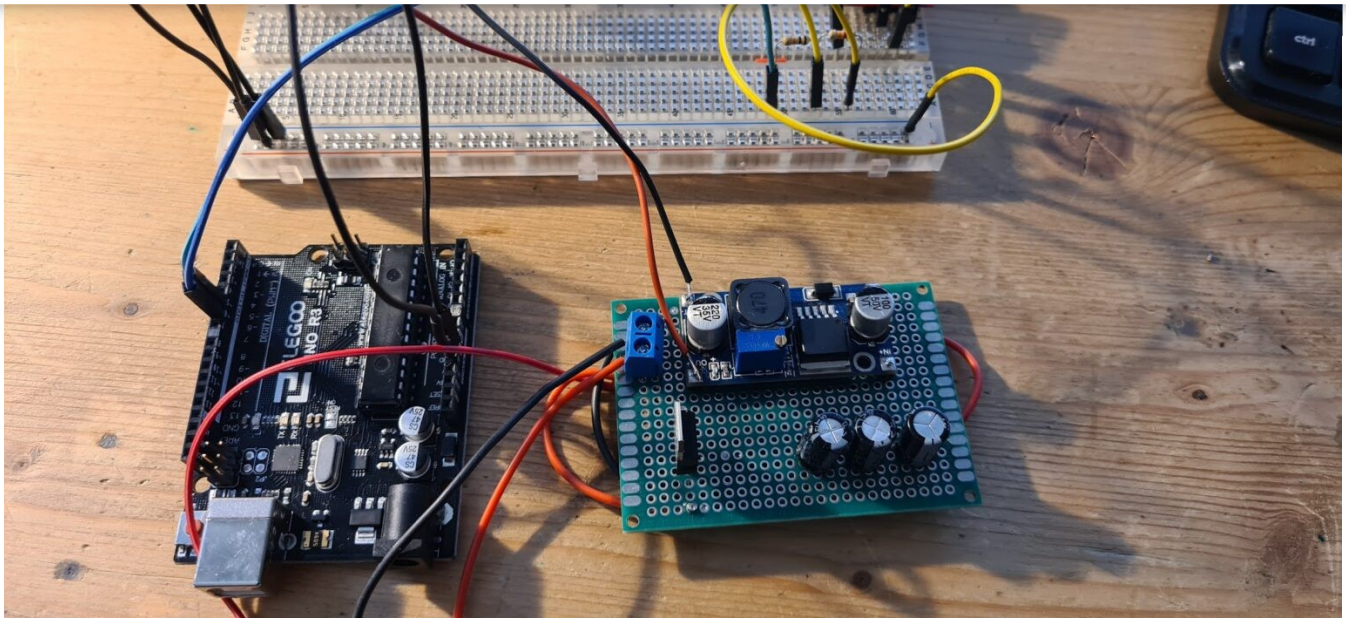
Confesso che non è la prima volta che progetto un dispositivo che integra questo modulo di comunicazione GSM. Tuttavia, in passato, mi sono sempre arenato senza riuscire a farlo funzionare.

Stavolta ho passato molte ore a studiare e a fare prove fino a quando non ho finalmente ottenuto il risultato sperato. Dalla lettura dei forum ho notato che non poche persone hanno avuto i miei stessi problemi. Ecco perché ho deciso di condividere l'esito delle mie tante ricerche e dei vari esperimenti (molti dei quali falliti).

Spero che questa guida aiuti molti maker appassionati di elettronica a risolvere gli stessi problemi e a realizzare i loro progetti.

Cliccando sul tag **SIM800L** potrete scoprire **tutti gli articoli** e i progetti in cui abbiamo utilizzato questo componente.

Componenti



Prototipo con breadbord per SIM800L e UNO R3

Iniziamo come sempre nei nostri articoli con il materiale utilizzato nel progetto:

- .Una CPU Arduino UNO R3 (versione cinese con interfaccia CH340) per gestire l'interfaccia con il PC: **guarda l'articolo su amazon**
- Uno step-down DC-DC con integrato LM2596: **guarda l'articolo su amazon**
- Un modulo GSM SIM800L: **guarda l'articolo su amazon**
- Una resistenza da 10K e una da 22K;
- Condensatori vari messi in parallelo.

(In qualità di Affiliato Amazon io ricevo un guadagno dagli acquisti idonei)

Alimentazione

Iniziamo con una tabella che riporta l'assorbimento (in mA e A) del modulo per ogni specifica modalità di funzionamento:

Modes	Frequency	Current Consumption
Power down		60 uA
Sleep mode		1 mA
Stand by		18 mA

Call	DCS1800	146 mA
Call	PCS1900	131 mA
GPRS		453 mA
Transmission burst		2 A

Tabella degli assorbimenti per modalità di funzionamento

L'intervallo di tensione per l'alimentazione di questo modulo va dai 3.4V ai 4.4V. Per questo motivo, nel mio precedente articolo sull'allarme GSM per il livello di carica della batteria della moto (**https://www.officinasottocasa.it/allarme-gsm-per-batteria-scarica-progetto-hardware/**), avevo pensato di alimentare il modulo con un semplice partitore di tensione in uscita dall'integrato LM7805.

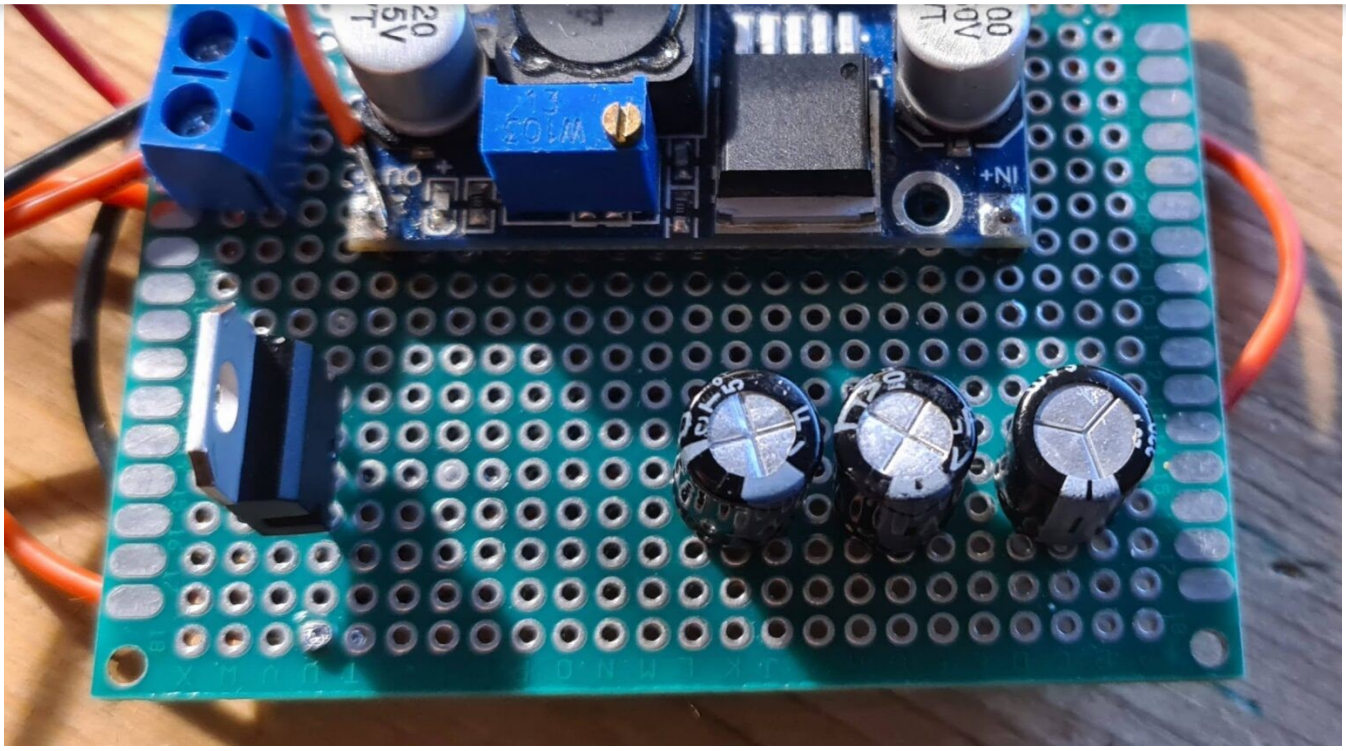
Guardando la tabella della corrente necessaria per il funzionamento delle varie modalità appare evidente che, contrariamente a quanto pensassi (e avessi visto in qualche articolo su internet), non è assolutamente possibile alimentare il modulo in questo modo. Tanto meno utilizzando direttamente la porta 5V di Arduino.

In rete ho visto alcuni "colleghi" che hanno utilizzato una batteria LIPO da 3.7V. Effettivamente sarebbe stato possibile. Ma per l'applicazione in dispositivi permanentemente connessi avremmo dovuto sviluppare anche la gestione della carica della stessa batteria, magari con un BMS per batterie LIPO.

Ho preferito la soluzione dello step-down DC-DC che trova facile impiego sia per applicazione su banco prova che nel mio progetto originario da cui nasce l'esigenza del suo utilizzo.

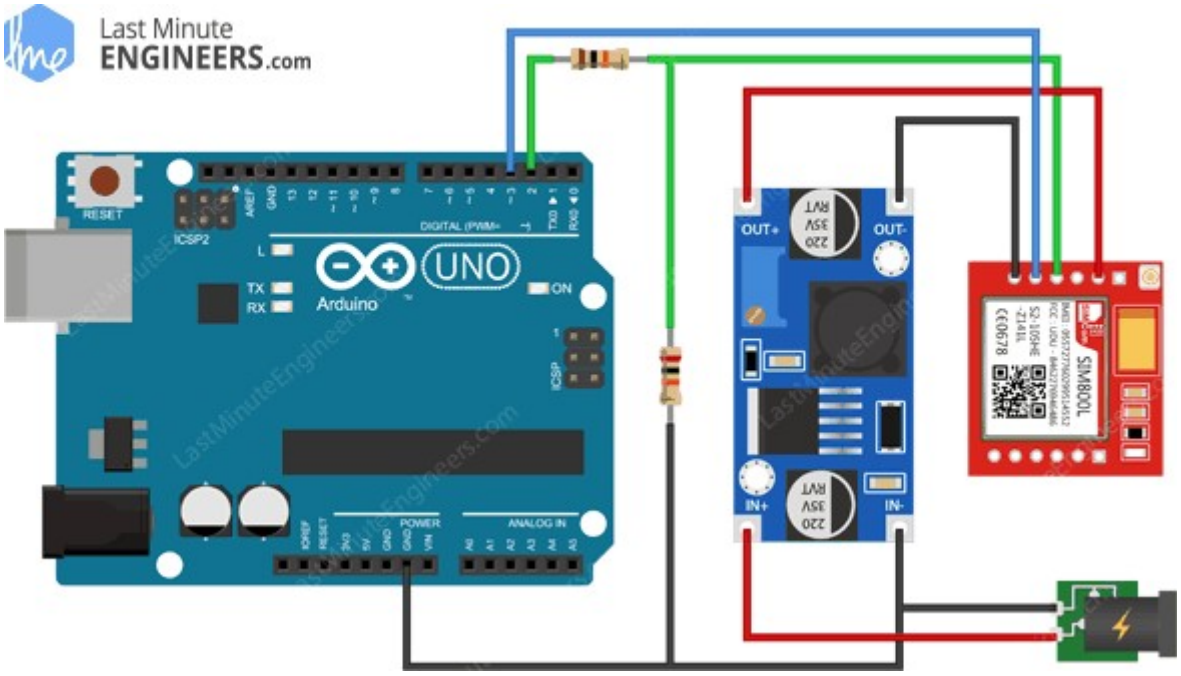
Visto che tornerà utile spesso, mi sono costruito (su una basetta mille-fori) un mini modulo di alimentazione permanente composto da un banale LM7805 (che, lo ricordiamo, è uno step-down DC-DC con uscita fissa a 5V) per alimentare qualunque scheda Arduino e uno step-down DC-DC con uscita variabile che utilizza un LM2596, in grado di supportare correnti molto più elevate.

Attenzione: i condensatori che vedete in foto non sono collegati al circuito.

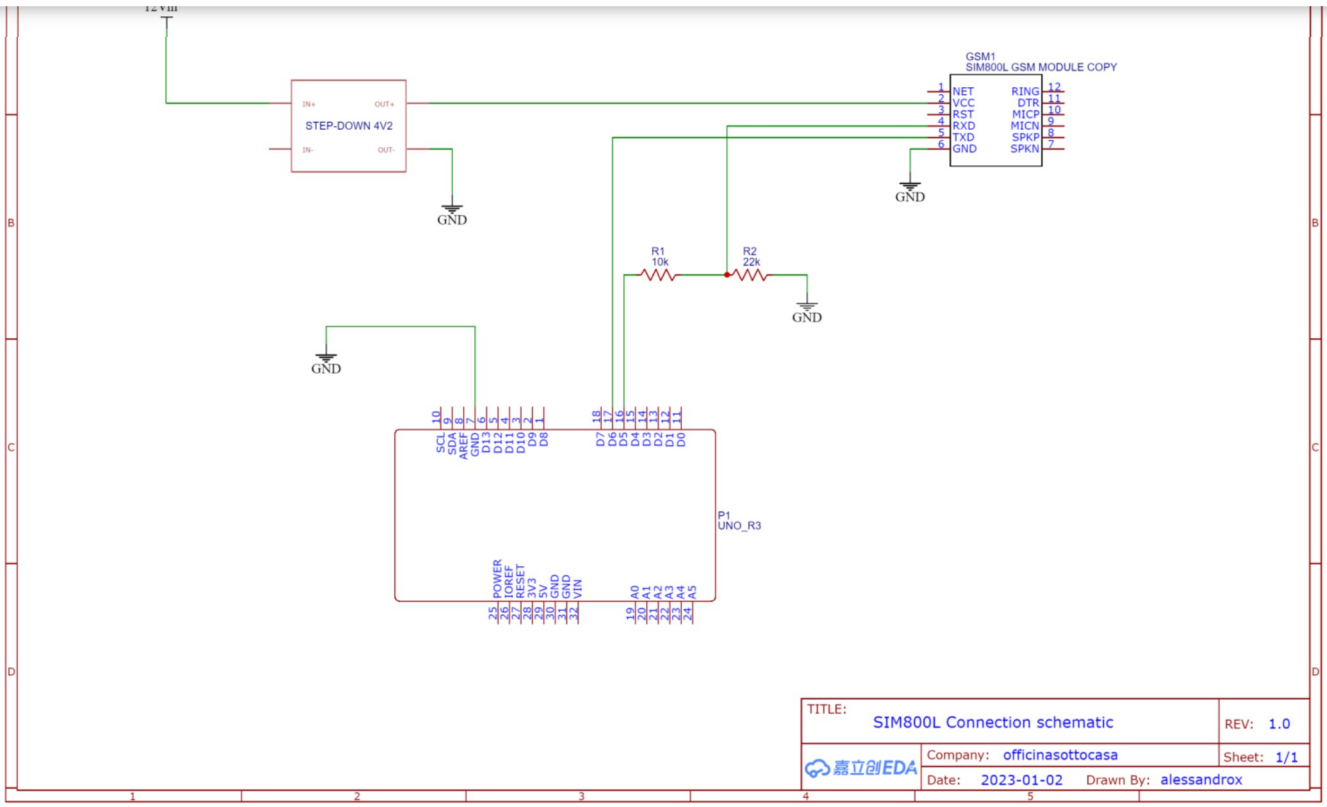


Scheda di alimentazione 5V con LM7805 e 4.2V con LM2596

Che poi, in buona sostanza, altro non è che la realizzazione di questo schema che ho trovato in questo articolo: **https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/**



Connessione seriale con Arduino



Schema di connessione SIM800L e Arduino UNO R3

Iniziamo con una piccola nota sulla scheda Elegoo UNO R3, clone della omologa Arduino UNO R3.

Ha lo stesso integrato dell'originale Arduino ATMEGA328P. Tuttavia, il convertitore USB – Seriale utilizza i chip CH340G e i driver per questo Chip non sono compresi nell'installazione dell'IDE di Arduino. Motivo per cui dovranno essere installati separatamente. In questo articolo **https://www.makerslab.it/installare-i-driver-per-il-modulo-ftdi-ch340g-per-i-cloni-di-arduino/** trovate le informazioni e i link per scaricare e installare i driver per il CH340G.

Convertitore USB-Seriale cloni Arduino

A questo punto dobbiamo aprire un inciso sui livelli logici che non sono gli stessi tra la sc

Arduino UNO utilizza il livello logico 5V mentre il modulo SIM800L utilizza il livello logico 3.3V.

Per abbassare il livello del segnale di Arduino da 5V a 3.3V possiamo utilizzare il nostro (ormai caro e inseparabile) partitore di tensione con una resistenza da 10K tra il pin Rx di SIM800L e il pin digitale di Arduino e una da 20K tra il pin Rx e GND di SIM800L. Esistono anche dei convertitori di livello logico 5V-3V3, vi ho messo l'articolo nella sezione componenti.

Sketch per utilizzo

A questo punto non ci resta che caricare sulla nostra scheda UNO lo sketch per permetterci di colloquiare con il SIM800L e fare i nostri esperimenti.

Io ho utilizzato un semplice sketch che invia al modulo GSM quello che noi scriviamo sul monitor seriale e visualizza sul monitor seriale quello che il modulo GSM invia alla scheda.

```
//Includiamo la libreria per la comunicazione seriale tra l'Arduino e il SIM800L
#include "SoftwareSerial.h";

//Definiamo il pin dell'Arduino a cui è collegato il pin TX del Sim800L, in questo caso D6
#define SIM_TX_PIN 6
//Definiamo il pin dellì'Arduino a cui è collegato il pin TX del Sim800L, in questo caso D
5 (occhio al partitore!)
#define SIM_RX_PIN 5
//Definiamo una seriale software sui TX e RX del SIM800L
SoftwareSerial serialSIM800(SIM_TX_PIN, SIM_RX_PIN);

void setup() {
//Inizializziamo la seriale tra Arduino e il PC
Serial.begin(57600);
//Inizializziamo la seriale virtuale lato sim800l
serialSIM800.begin(57600);
delay(500);
}

void loop() {

//Scrivi sulla seriale Arduino-PC quello che leggi dal pin TX del SIM800L
if(serialSIM800.available()){
Serial.write(serialSIM800.read());
}
//Invia al pin RX del SIM800L quello che leggi sulla seriale PC-Arduino
if(Serial.available()){
serialSIM800.write(Serial.read());
}
}
```

Ora che lo abbiamo connesso e alimentato possiamo inserire la scheda SIM e verificare il led di funzionamento:

- 1 lampeggio ogni 1 secondo: modulo attivo ma non ancora registrato sulla rete;
- 1 lampeggio ogni 3 secondi: modulo attivo e correttamente registrato sulla rete.

Per avere maggiori dettagli sul funzionamento dobbiamo inviare qualche comando AT e analizzare le risposte.

Comandi AT

Il manuale completo dei comandi è disponibile nella documentazione. Ne riporto qui solo alcuni, i principali, che ci permettono di testare se i collegamenti funzionano e se possiamo quindi utilizzare il nostro modulo per inviare messaggi.

Io li ho lanciati in sequenza per riepilogare i risultati e commentarli insieme.

```
10:40:20.795 -> AT
10:40:20.796 -> OK
```

Primo comando da lanciare per verificare il dialogo tra l'Arduino e il SIM800L. Se non ricevete in risposta la stringa "OK" c'è un problema nei collegamenti tra i pin RX e TX del SIM800L e i pin dell'Arduino.

```
10:40:25.735 -> AT+CSQ
10:40:35.980 -> +CSQ: 16,0
```

Serve per verificare la qualità del segnale. Il primo numero (16 nel mio caso) dovrebbe trovarsi tra 2 e 30 che equivale ad un segnale tra -110 e i -54 dbm.

```
10:40:35.983 -> AT+CCID
10:40:42.373 -> 8944502206206365147
```

Si tratta dell'identificativo della SIM. Lo uso per verificare che il modulo legga correttamente i dati della SIM.

```
10:40:46.745 -> AT+CREG
10:40:50.949 -> +CREG: 0,5
```

```
10:40:50.949 -> AT+COPS?
10:40:57.062 -> +COPS: 0,0,"Wind Telecom SpA"
10:40:57.065 ->
10:40:57.065 -> OK
```

Verifichiamo anche l'operatore con cui siamo connessi ed effettivamente troviamo conferma di quanto sopra.

Sleep & wake-up

Ho creato due specifiche funzioni rispettivamente per mandare il modulo in sleep mode e per "svegliarlo". Il prerequisito è sempre che il pin DTR del SIM800L sia connesso ad un pin output dell'Arduino impostato al valore HIGH.

```
void sim800sleepmode(){
  serialSIM800.println("AT+CSCLK=2");

}

void sim800wakeup(){
  digitalWrite(DTR_PIN,LOW);
  delay(200);
  digitalWrite(DTR_PIN,HIGH);
  serialSIM800.println("AT");
  serialSIM800.println("AT+CSCLK=0");
}
```

Come vedete, per mandarlo in sleep mode è sufficiente un comando AT (AT+CSCLK=2) con pin DTR impostato ad HIGH. Il problema è che una volta entrato in sleep mode il modulo chiude le comunicazioni sulla seriale.

Per svegliarlo dalla modalità sleep è necessario mandare il pin DTR al valore LOW (io ho impostato 200 millisecondi) e a quel punto inviare un comando AT qualsiasi. Solo dopo queste operazioni si può inviare il comando AT+CSCLK=1 per ripristinare la normale funzionalità.

Inviare un SMS

Per inviare un sms possiamo usare lo sketch di interfaccia seriale scrivendo i comandi direttamente nel monitor seriale oppure scrivere lo sketch che invia un SMS.

vedremo in uno specifico articolo.

```
11:26:23.833 -> at+cmgf=1
11:26:30.783 -> OK
11:26:30.783 -> at+cmgs="+39xxxxxxxxx"
11:26:46.372 -> > testo di prova
11:26:53.794 -> >
11:26:58.901 -> >
11:27:12.546 -> +CMGS: 12
11:27:12.550 ->
11:27:12.550 -> OK
```

Documentazione

Fondamentale la consultazione della documentazione relativa al modulo e ai comandi AT che vi condivido qui:

- **Datasheet SIM800L** – File pdf con tutte le specifiche hardware;
- **Manuale comandi AT** – file pdf con tutti i comandi AT, linguaggio per poter colloquiare con il modulo.

Unisciti alla nostra *community* e aiutaci a crescere!

Ti è piaciuto questo articolo? Condividi la tua opinione lasciando un commento! Aiutaci a migliorare sempre di più i nostri contenuti.

Hai un blog o un sito web? Condividi il link nei commenti e visitiamolo insieme. Aiutiamoci a crescere a vicenda!

Hai un'idea in mente ma non sai da dove iniziare? Contattaci e raccontaci la tua storia.

Riferimenti

Per arrivare a far funzionare ho consultato numerosi siti e ho deciso di proporvi l'intera sitografia qui di seguito:

- **https://www.makerslab.it/installare-i-driver-per-il-modulo-ftdi-ch340g-per-i-cloni-di-arduino/** – driver per il ch340G;
- **https://www.treccarichi.net/2016/06/sim800l/** – nei commenti viene evidenziato

interessante;

- **https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/** – uno degli articoli più interessanti e forse fatti meglio;
- **https://iemma.it/blog/sim800l-modulo-gsm-gprs-per-arduino** – fondamentale per la questione del partitore per l'adattamento dei livelli logici tra arduino e sim800l;
- **https://pijaeducation.com/arduino/gsm/send-receive-messages-using-sim800l-with-arduino/**
- **https://andino.systems/extensions/2g-modem-sim800l/sms**
- **https://www.raviyp.com/sim900-sim800-sleep-mode-at-commands/** – articolo molto interessante er gestire lo sleep-mode e soprattutto la funzione di wake-up da software

Categorie: **ELETTRONICA**

Tags: Arduino sim800l

6 commenti

giuseppe · 18/11/2024 alle 19:27

ciao il modulo SIM800L funziona con tutte le sim

RISPONDI

Alessandro · 20/11/2024 alle 11:33

Ciao Giuseppe e grazie per il tuo contributo. Se sei interessato ad altri contenuti come questo faccelo sapere. Il vostro feedback è fondamentale per pubblicare contenuti utili e di qualità.

RISPONDI

GIUSEPPE · 18/11/2024 alle 19:24

Ciao mario , questo modulo sim800L funziona con le sim 4 e 5, grazie

RIS

Mario · 17/06/2024 alle 12:10

Margherita e Alessandro · 26/08/2024 alle 08:37

Grazie Mario. Cerchiamo sempre di condividere informazioni in modo fruibile ai nostri lettori. Non esitare a contattarci per qualsiasi informazione e/o approfondimento.

RISPONDI

Come monitorare blackout della rete elettrica da remoto con Arduino e un SIM800L - officinasottocasa · 11/11/2024 alle 10:53

[…] In questo articolo usiamo il nostro caro circuito SIM800L. Non ci dilungheremo sul suo funzionamento perché abbiamo già detto tutto in questo articolo: Guida completa modulo GSM SIM800L […]

RISPONDI

Lascia un commento

Nome *

Email *

Sito web

A cosa stai pensando?

Salva il mio nome, email e sito web in questo browser per la prossima volta che commento.

INVIA COMMENTO

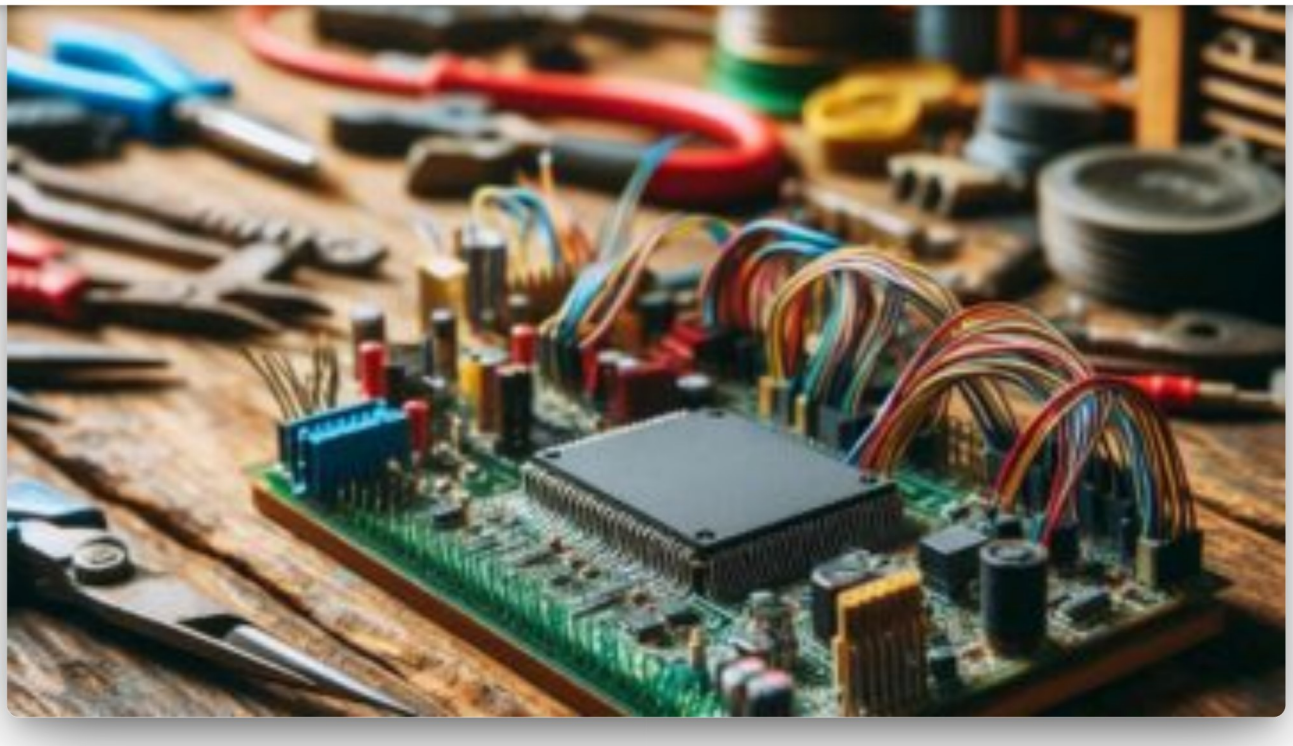
Categorie

- Arredo giardino (3)
  - Armadi da giardino (2)
  - Coffee table (1)
  - Tavoli da giardino (1)
- Complementi (2)
  - arredo soggiorno (2)
- Elettronica (6)
- Ferro (12)
- Lampade (2)
  - Lampade da terra (2)
- Legno (8)
  - Montessori (2)
- Melange (5)
- Mobili (5)
  - Carrelli TV (1)
  - Consolle (1)
  - HI-FI (2)
  - Librerie (1)
  - Mobili TV (1)
- Officina (12)
  - Creazioni (10)
  - Materiali (1)
  - Storie (2)

- Scrivanie (1)
- Tavoli per esterni (1)
- Tavoli per interni (1)
- Tavolini quadrati (1)

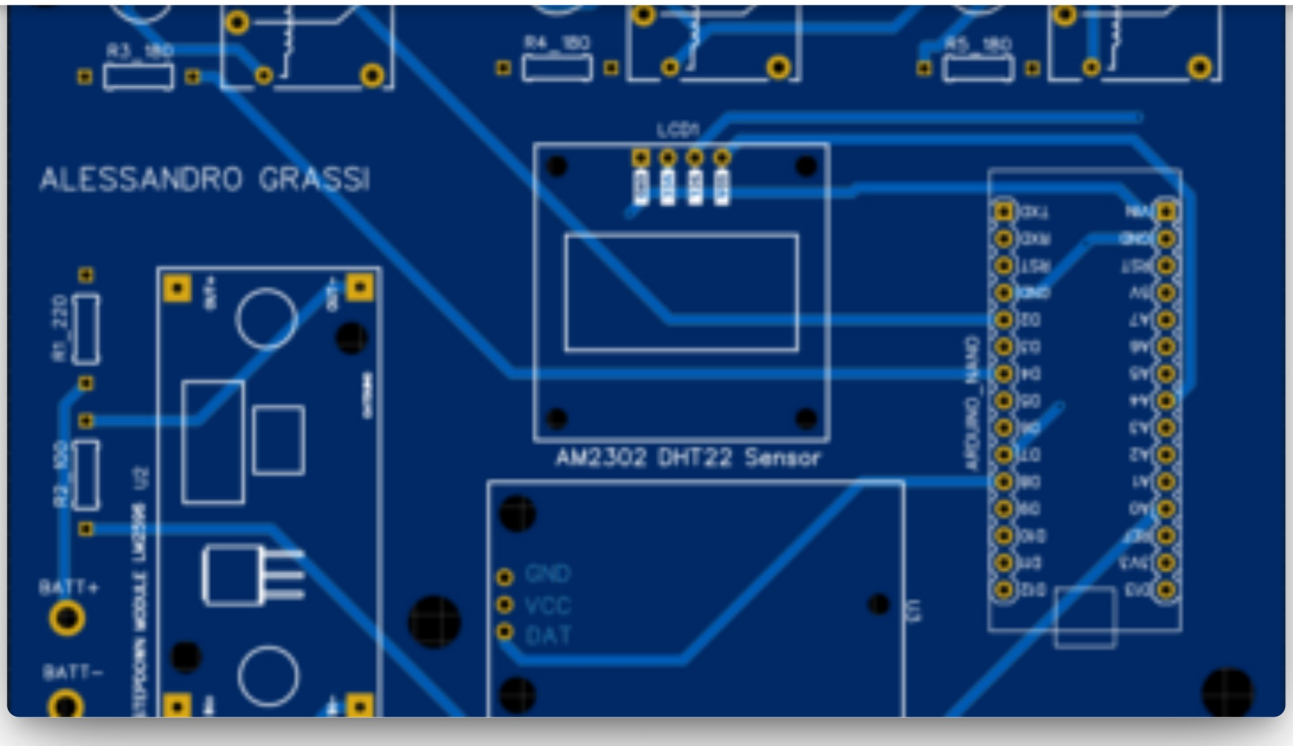
Vetro (1)

Articoli correlati



Come monitorare blackout della rete elettrica da remoto con Arduino e un SIM800L

Come monitorare blackout della rete elettrica da remoto con Arduino e un SIM800L



Upgrade del sistema di estrazione umidità: ottimizzazione con Arduino Nano e DHT22-AM2302 per un controllo avanzato della gestione energetica

Un nuovo progetto di officinasottocasa.it con Arduino Nano e DHT22-AM2302 per il controllo avanzato dell'umidità. Cablaggio facile e codice sorgente incluso. Mantieni il tuo ambiente sempre perfetto con questo tutorial dettagliato!

Allarme GSM per batteria scarica – Parte 2: Software

La puntata conclusiva del sistema di allarme per il livello di carica della batteria della moto

officinasottocasa.it

About

Chi siamo

Privacy policy

Follow us

Hestia | Sviluppato da ThemeIsle