











EMBEDDED DESIGN/ MAKERS BOARDS

 **elektor** ARTICLES INSIDE
design > share > sell

IN QUESTO NUMERO:

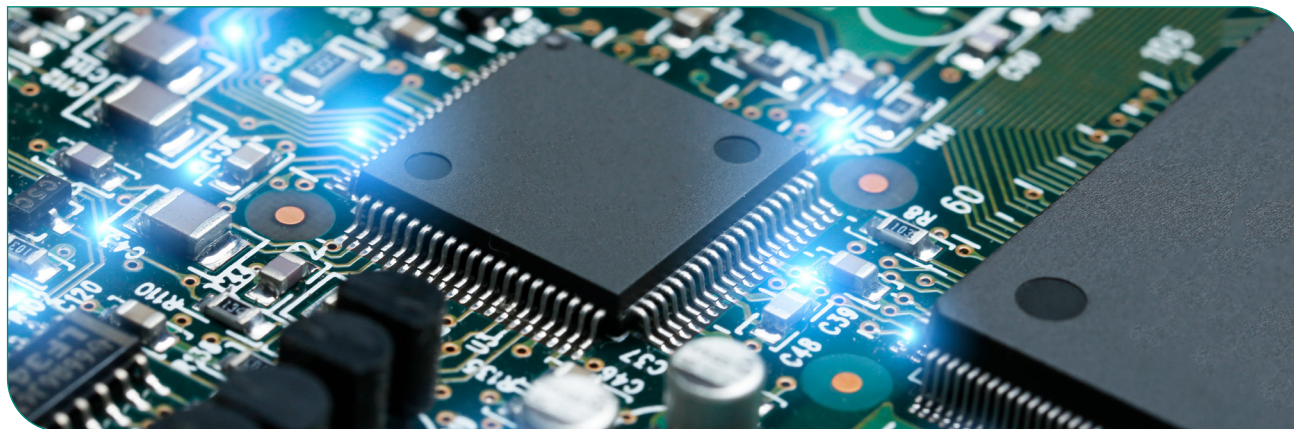
- IMPLEMENTAZIONE DEGLI ALGORITMI SHA-3 NEI SISTEMI EMBEDDED
- COME SFRUTTARE MICROPYTHON PER LA REALIZZAZIONE DI APPLICAZIONI EMBEDDED CON XBEE3
- LE SOLUZIONI ARDUINO PER IL SETTORE EDUCATIONAL DI TME ELECTRONIC COMPONENTS
- SMALL CIRCUITS REVIVAL - RELÈ AD ALTA EFFICIENZA ENERGETICA
- PROGETTIAMO I NOSTRI FILTRI CON FILTERPRO DELLA TEXAS INSTRUMENTS
- E MOLTO ALTRO!

FOCUS DEL MESE

IOT	IOT	GEN/FEB	 1 Febbraio
CAR HACKING - OBD	AUTOMOTIVE	MARZO	 1 Marzo
BLOCKCHAIN	AI	APRILE	 1 Aprile
AUDIO/VIDEO	WIRELESS/RF	MAGGIO	 1 Maggio
MAKERS SENSOR BOARDS	SENSORS	GIUGNO	 1 Giugno
ROBOTICS	POWER/MOTOR	LUGLIO	 1 Luglio
SMART PROJECTS	INDUSTRY 4.0	AGO/SET	 1 Settembre
MAKERS LABORATORY	TEST & MEASUREMENTS	OTTOBRE	 1 Ottobre
WEARABLE	LED LIGHTING OPTOELECTRONICS	NOVEMBRE	 1 Novembre
MAKERS BOARDS	EMBEDDED DESIGN	DICEMBRE	 1 Dicembre
QUESTO MESE:	MAKERS ZONE		

COSA LEGGERAI NEL 2020?

EMBEDDED DESIGN/ MAKERS BOARDS



Founder&Editor

Emanuele Bonanni

CFO

Lidia Balica

Editorial Assistant

Maria Pisani

Maker in Chief

Giordana Francesca Brescia

Advertising & Marketing

Cristian Balica

cristian@contangosl.com

Graphic Designer

Marilde Mirra

Circulation

Users - 137.863

Social Network - 126.031

© Copyright

Tutti i diritti di riproduzione o di traduzione degli articoli pubblicati sono riservati. Manoscritti e disegni sono di proprietà di Contango SL.

E' vietata la riproduzione anche parziale degli articoli salvo espressa autorizzazione scritta dell'editore. I contenuti pubblicitari sono riportati senza responsabilità, a puro titolo informativo.

EDITORIALE

EMBEDDED SYSTEMS:
IL CUORE
DELL'INDUSTRIA
ELETTRONICA

3

IMPLEMENTAZIONE
DEGLI ALGORITMI SHA-
3 NEI SISTEMI EMBEDDED

5

COME SFRUTTARE
MICROPYTHON PER
LA REALIZZAZIONE
DI APPLICAZIONI
EMBEDDED CON XBEE3

12

IL NUOVO
MICROCONTROLORE
ESP32-S2

21

IL NUOVO COMPUTER
DESKTOP RASPBERRY
PI 400

26

JUKEBOXINO: SCEGLI
IL TUO BRANO CON
ARDUINO

28

GIOCHIAMO CON
ARDUINO: STRUMENTO
MUSICALE AD
ULTRASUONI "NO TOUCH"

35

ACQUISIZIONE DATI DI
SCARICA PER CELLE
LI-POLI CON ARDUINO
E DISPLAY VFD

43

LE SOLUZIONI
ARDUINO PER
IL SETTORE
EDUCATIONAL DI TME
ELECTRONIC COMPONENTS

51

SCHEDA DI
CONTROLLO PER
RELE A 9 CANALI

57

SMALL CIRCUITS
REVIVAL - RELÈ AD
ALTA EFFICIENZA
ENERGETICA

61

IL LINGUAGGIO D:
FONDAMENTI
(PARTE 1)

63

IL LINGUAGGIO D:
TANTE POTENZIALITÀ
IN PIÙ (PARTE 2)

71

UNO SGUARDO ALLE
TECNOLOGIE DEL
DOMANI:
IL GPGPU E CUDA

81

FAST FOURIER
TRANSFORMATION:
RICONOSCIAMO UN
DTMF

86

PROGETTIAMO I
NOSTRI FILTRI CON
FILTERPRO DELLA
TEXAS INSTRUMENTS

91

OS WRITING [3]: PER
NON ESSERE FUORI
DAL MONDO

100

OS WRITING [4]:
REGOLARE IL
TRAFFICO

106



GLI ARTICOLI



Elettronica Open Source



elektor

design > share > sell

**IN UN SOLO
MAGAZINE!**

**OGNI MESE
SU FIRMWARE 2.0**

**2 ARTICOLI
DI ELEKTOR**



IMPLEMENTAZIONE DEGLI ALGORITMI SHA-3 NEI SISTEMI EMBEDDED

di Stefano Lovati

SHA-3, la versione più recente del Secure Hash Algorithm le cui specifiche sono state ufficialmente rilasciate nel 2015, si differenzia in modo significativo rispetto alle precedenti versioni di algoritmi della stessa famiglia (SHA-1 e SHA-2). Scopriamo in questo articolo quali sono i concetti basilari dello standard SHA-3 (e della famiglia di algoritmi Keccak da cui discende), analizzando alcune implementazioni a livello hardware dell'algoritmo.

INTRODUZIONE

Il 2 ottobre 2012 l'ente governativo americano **NIST** (National Institute of Standards and Technology) selezionò (al termine di una competizione ufficiale durata anni) l'**algoritmo Keccak** come migliore implementazione del nuovo e futuro standard per la crittografia **SHA-3**. Il processo che ha portato alla scelta di un nuovo standard per le funzioni di hash ebbe inizio nel lontano 2007. In quel periodo, infatti, furono registrati un certo numero di attacchi e individuati punti di debolezza nei predecessori dello standard **SHA-2** (più precisamente nell'**MD5** e nell'**SHA-1**). Si temeva allora che le implementazioni maggiormente utilizzate dello standard SHA-2 (**SHA-256**

e **SHA-512**) prima o poi sarebbero diventate vulnerabili o si sarebbe giunti alla "rottura" del codice crittografico, come avvenuto ad esempio per l'**MD5**. Poiché la scelta e la definizione di una primitiva per la crittografia richiede parecchio tempo, il NIST avviò il processo di creazione dell'**SHA-3** con il chiaro intento di trovare un valido sostituto per l'**SHA-2**. In realtà le cose andarono diversamente e lo standard SHA-2 si dimostrò più robusto di quanto inizialmente ipotizzato. Con l'obiettivo dichiarato di anteporre a ogni cosa la robustezza, altri aspetti come prestazioni, velocità di esecuzione e facilità di implementazione a livello software passarono in secondo piano. Alla fine, la scelta del NIST è ricaduta sull'algoritmo Keccak, che

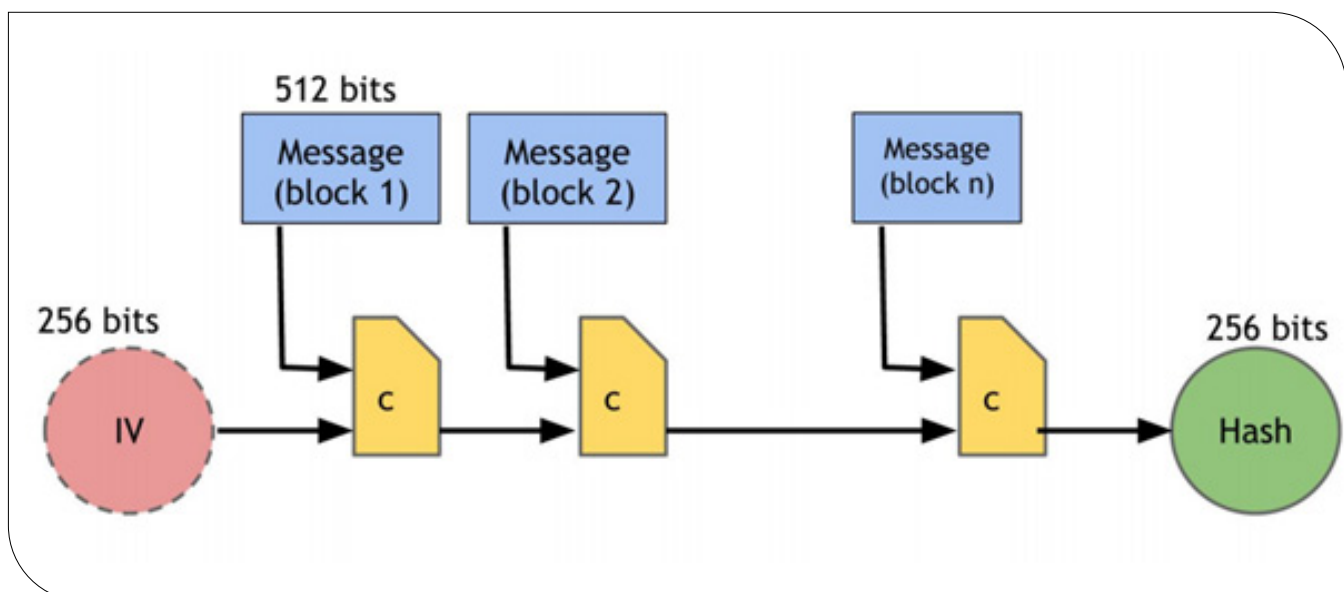


Figura 1: rappresentazione schematica di una singola iterazione della funzione SHA-256

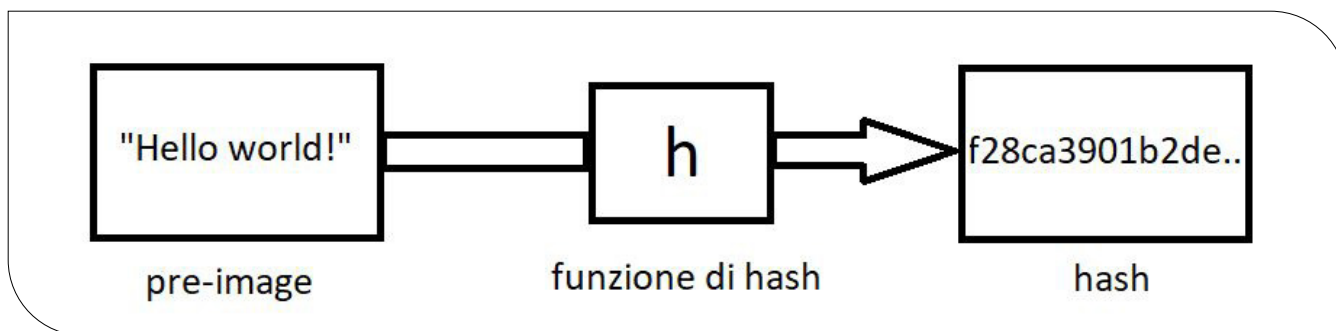


Figura 2: schema a blocchi di una generica funzione di hash

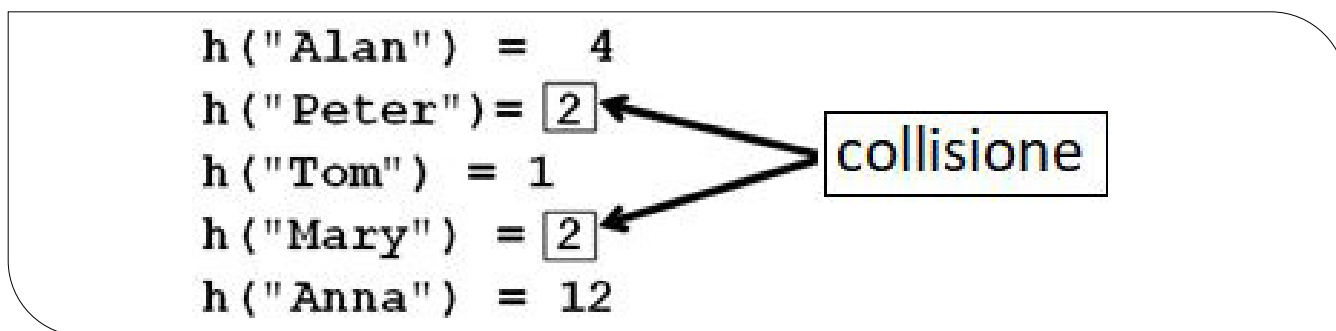


Figura 3: un esempio di collisione: due valori di ingresso producono lo stesso hash in uscita

presentava profonde differenze sia rispetto all'SHA-2 che all'AES. In questo modo si riteneva di ridurre i rischi di una "rottura" simultanea di più algoritmi. Keccak è un algoritmo sviluppato da Guido Bertoni, Joan Daemen (uno dei creatori di AES), Michaël Peeters e Gilles Van Assche. Dobbiamo a questo punto ritenere l'SHA-2 come non più sicuro? Assolutamente no, SHA-2 è tuttora valido e molto sicuro. Ci interessa, tuttavia, sapere che esiste una valida alternativa (con un maggiore grado di sicurezza) all'SHA-2 e questa si chiama SHA-3. In **Figura 1** possiamo osservare una rappresentazione schematica del **processo di hashing** utilizzato per la validazione di una transazione nell'ambito delle **criptovalute** (ad esempio nel caso del Bitcoin). L'algoritmo utilizzato in questo caso è l'SHA-256, una delle varianti dell'SHA-2. I dati in ingresso sono suddivisi in blocchi di 512 bit, mentre la funzione di compressione (evidenziata dai blocchi gialli) viene eseguita una volta per ogni blocco di ingresso, inclusi i risultati dell'iterazione precedente.

Che importanza rivestono gli algoritmi di hash e dove vengono utilizzati? Dobbiamo anzitutto fare un richiamo alle **funzioni di hash**, perché è da qui che nasce tutto.

FUNZIONI DI HASH

Molti tra noi sanno che le funzioni di hash rivestono un'importanza notevole e strategica nel mondo delle criptova-

lute, basti pensare ad esempi come **Bitcoin** ed **Ethereum**. In realtà, le funzioni di hash sono ampiamente utilizzate in tutti i sistemi crittografici (servono ad esempio alla creazione della *digital fingerprint*) e costituiscono la base della crittografia moderna. Una funzione di hash può essere definita come una funzione in grado di mappare una quantità di dati di dimensione arbitraria su una quantità di dati di dimensione fissa. I dati in ingresso alla **funzione di hash** prendono il nome di *preimage*, mentre i dati in uscita sono chiamati semplicemente *hash*. Il processo svolto da una generica funzione di hash è schematizzato in **Figura 2**, dove la funzione è contraddistinta semplicemente con la lettera "h". La dimensione del dato in uscita ("hash" in **Figura 2**) dipende dal particolare algoritmo utilizzato (nel caso dell'SHA-256, ad esempio, è pari a 256 bit).

Una particolare categoria di funzioni di hash è rappresentata dalle **funzioni crittografiche di hash**, le quali presentano caratteristiche peculiari atte a garantire un elevato grado di sicurezza alle piattaforme su cui si basano le criptovalute. Una funzione crittografica di hash è una funzione di hash *monodirezionale* che mappa dati di dimensione arbitraria su una stringa di bit di lunghezza prefissata. La sua natura intrinseca *monodirezionale* garantisce che non sia fattibile, dal punto di vista computazionale, ricreare i dati in ingresso conoscendo solo i dati in uscita. L'unico modo per risalire all'informazione originaria

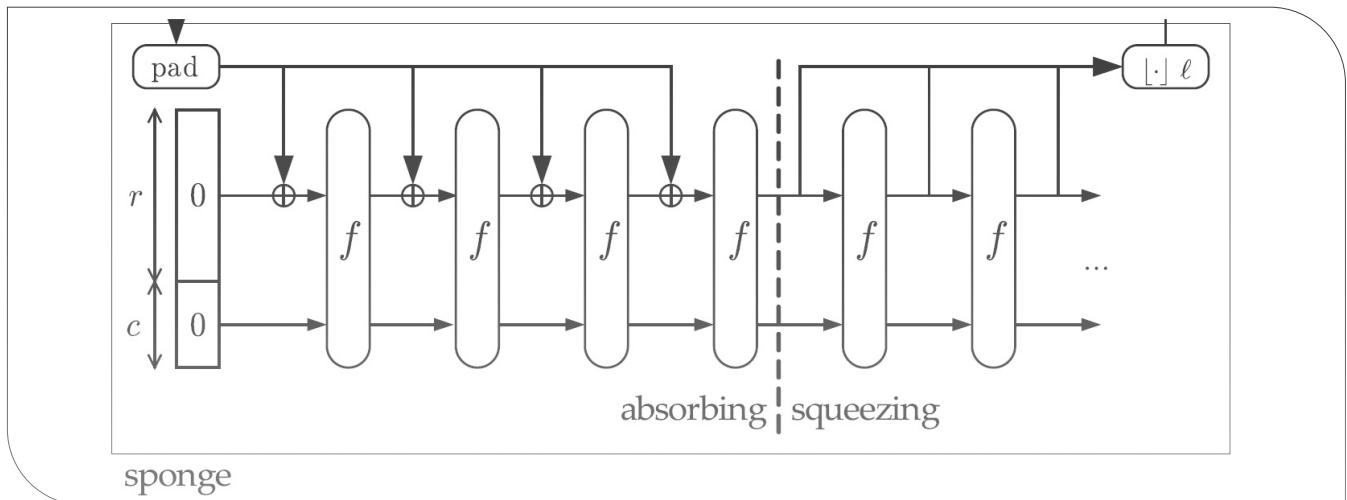


Figura 4: schema concettuale della sponge construction, il cuore di Keccak

è quello di applicare un attacco **forza bruta** (*brute force*): poiché il numero di possibili casi da testare è elevatissimo, è semplice comprendere come questo sistema non sia ragionevolmente fattibile a livello pratico. Non solo: anche se si trovasse una combinazione di dati in ingresso in grado di generare lo stesso valore di hash, non si avrebbe la certezza che si tratti dei dati in ingresso originali. A questo proposito si usa anche dire che le funzioni di hash sono funzioni **"molti a uno"**. Quando si riescono a individuare due insiemi di dati in ingresso in grado di generare la stessa uscita, si è creata una **"collisione di hash"**. Maggiore è la sicurezza di una funzione di hash, minore è la probabilità di generare delle collisioni (in criptovalute come Ethereum le collisioni sono praticamente impossibili). Con riferimento alla funzione di hash *h*, schematizzata

- **irreversibilità**: la possibilità di risalire al messaggio originale partendo dalla sola conoscenza dell'hash non deve essere fattibile a livello pratico, equivalente come detto all'applicazione del metodo *brute force* su tutti i possibili messaggi in ingresso;
- **protezione dalle collisioni**: non deve essere fattibile produrre due differenti messaggi in ingresso che generino lo stesso hash in uscita.

La resistenza alle collisioni è particolarmente importante per evitare la falsificazione delle firme digitali utilizzate nella blockchain. La combinazione delle cinque proprietà precedentemente illustrate rende le funzioni crittografiche di hash particolarmente adatte per molteplici applicazioni

QUELLO CHE HAI LETTO E' UN ESTRATTO, L'ARTICOLO COMPLETO E' RISERVATO AGLI ABBONATI AD ELETTRONICA OPEN SOURCE.

PERCHE' ABBONARSI A PLATINUM 2.0?

UN ANNO DI **FIRMWARE 2.0**
TUTTI GLI ARTICOLI TECNICI RISERVATI
CONTEST E PROMOZIONI RISERVATI



VOGLIO ABBONARMI!

COME SFRUTTARE MICROPYTHON PER LA REALIZZAZIONE DI APPLICAZIONI EMBEDDED CON XBEE3

di Nicolò Ferranti

Il connubio tra hardware a basso livello e linguaggio di programmazione ad alto livello potrebbe rivelarsi una delle mosse vincenti per lo sviluppo di sistemi embedded. In questo articolo vedremo come il linguaggio di programmazione orientato agli oggetti come quello di Python può affacciarsi, grazie alla sua implementazione MicroPython, a contesti di sviluppo di applicazioni embedded. Continueremo a lavorare con i nostri moduli XBee3 e finalmente scopriremo come scrivere un programma che può consentire al modulo di lavorare in maniera autonoma per la gestione di applicazioni custom.

INTRODUZIONE

Nello scorso appuntamento abbiamo iniziato ad apprezzare le potenzialità fornite da XCTU, il tool gratuito offerto dalla DIGI per la programmazione dei moduli XBee3. In particolare, dopo una rapida analisi delle caratteristiche dei nuovi moduli, abbiamo realizzato una nostra prima applicazione che sfrutta il **protocollo ZigBee**. L'applicazione in questione consentiva la creazione di una rete PAN con annessa configurazione di un **Coordinator** e di un **Router**, nodi fondamentali per il rispetto delle caratteristiche del protocollo. Conclusa la fase di configurazione ci siamo accertati del loro funzionamento verificando in un primo momento la costruzione della rete e successivamente ci siamo sincerati che i due moduli comunicano. Al fine di testare il corretto trasferimento dei messaggi ci siamo preoccupati di come eseguire una valida costruzione di un frame e abbiamo visto come la modalità API mode ci consente di ricavare diverse informazioni sulla qualità del messaggio inviato.

In questo articolo continueremo con la nostra pratica ed entreremo sempre più nel vivo delle potenzialità offerte dai nuovi moduli. Come già accennato nell'appuntamento "Alla scoperta degli Xbee3" una delle principali features che rende questi nuovi moduli degni di nota è l'introdu-

zione al **linguaggio di programmazione orientato agli oggetti** che consente di realizzare applicazioni ancora più custom e in particolare ci permette di svincolarci dalla necessità di microcontrollori esterni come accadeva nelle precedenti serie di XBee. Il focus di oggi sarà proprio **MicroPython** e la programmabilità dei moduli XBee3. Ma abbandoniamo le parole e iniziamo con la pratica!

MICROPYTHON

Ma cos'è MicroPython? E perché proprio lui? Come viene riportato nella documentazione ufficiale offerta da micropython.org: "MicroPython è un'implementazione snella ed efficace del linguaggio di programmazione Python3 che include un piccolo sottoinsieme delle librerie standard Python ed è ottimizzato per l'esecuzione su microcontrollori".

È proprio partendo dall'ottimizzazione per i microcontrollori che nel 2013 questo progetto ha avuto inizio, ponendosi l'obiettivo di consegnare agli sviluppatori un ambiente che gode delle potenzialità di un linguaggio di programmazione orientato agli oggetti come il **Python**, ma al tempo stesso risulta compatto e adatto per la realizzazione di applicazioni embedded con risorse limitate. La leggerezza di questa implementazione può essere apprezzata

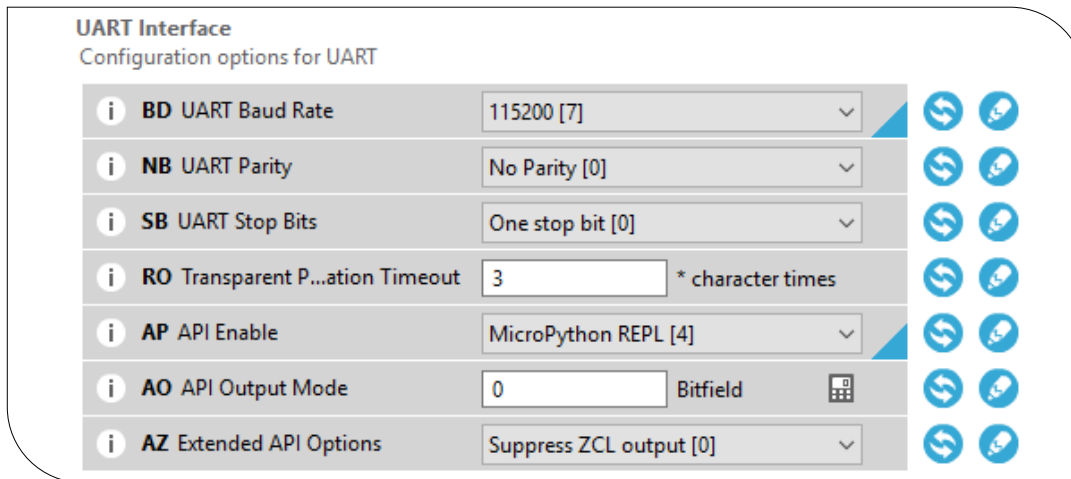


Figura 1: enable MicroPython da XCTU

osservando che il MicroPython è pensato per lavorare in condizioni limite come 16kB di RAM. Anticipiamo che nei moduli XBee3 RF è integrato il microcontrollore HCS08 con circa 1MB di memoria disponibile di cui è allocato uno spazio pari a 32kB per l'Heap di memoria destinata all'utilizzo di variabili, oggetti e moduli importati. Abbiamo detto che MicroPython è un'implementazione del linguaggio di programmazione Python3 ma allo stesso tempo, visto il suo scopo, presenta distinzioni con tale linguaggio. Le principali differenze che si hanno tra il MicroPython e CPython3, considerata l'implementazione di riferimento del linguaggio Python3, sono le seguenti:

- in MicroPython non è supportata l'intera libreria standard del linguaggio di programmazione Python;
- MicroPython utilizza come mezzo principale per la gestione della memoria il garbage collection;
- con MicroPython le metaclassi non sono supportate;
- usi avanzati del linguaggio come ereditarietà multipla potrebbero non funzionare.

Per avere una lista più dettagliata delle differenze potete dare un'occhiata [qui](#). Siccome lo scopo di questo articolo non è quello di entrare nel dettaglio del linguaggio di programmazione Python3, ci limiteremo a menzionare le sue strutture dati e statement solo al momento del loro utilizzo.

MICROPYTHON TERMINAL

Prima di analizzare l'ambiente di sviluppo dobbiamo abilitare le API di MicroPython per consentire il corretto funzionamento. Per fare questo dobbiamo tornare nell'area **Configuration Working** di XCTU e abilitare nella sezione

UART Interface la API Enable: **MicroPython REPL[4]**. Questo passaggio viene riportato in **Figura 1**.

Entriamo ora nel vivo della programmazione dei moduli XBee3, in particolare vediamo come programmarli grazie ad XCTU. Per fare questo, XCTU ci mette a disposizione in Tools un **MicroPython Terminal**, si veda **Figura 2**, con il quale sarà possibile interfacciarsi con il modulo e rendere interattiva la programmazione.

Una volta entrati in **MicroPython Terminal** e chiusa la connessione seriale possiamo eseguire subito un test per accertarci della comunicazione digitando la sequenza di caratteri **+++**. Nel caso di un esito positivo otterremo un messaggio di ritorno **'ok'** come quello riportato in **Figura 3**, in caso contrario accertatevi che sia garantita una buona comunicazione con il modulo.

Per accedere al prompt useremo quello che viene chiamato con l'acronimo **REPL (Read-Eval-Print-Loop)**, un linguaggio di shell che accetta un input dell'utente, lo valuta e restituisce un risultato come output. I principali comandi **REPL** che utilizzeremo sono:

- **CTRL+B**, consente di accedere all'ambiente di MicroPython;
- **CTRL+D**, effettua il reboot del REPL ripulendolo da ogni variabile e funzione dichiarati in precedenza;
- **CTRL+E**, permette l'accesso alla modalità **paste mode** dov'è consentito testare righe di codice senza scriverle nella memoria flash del dispositivo;
- **CTRL+F**, permette l'accesso alla modalità **flash mode** dov'è consentito compilare e salvare il nostro codice nella flash del device.

Una volta che ci siamo accertati della comunicazione pos-

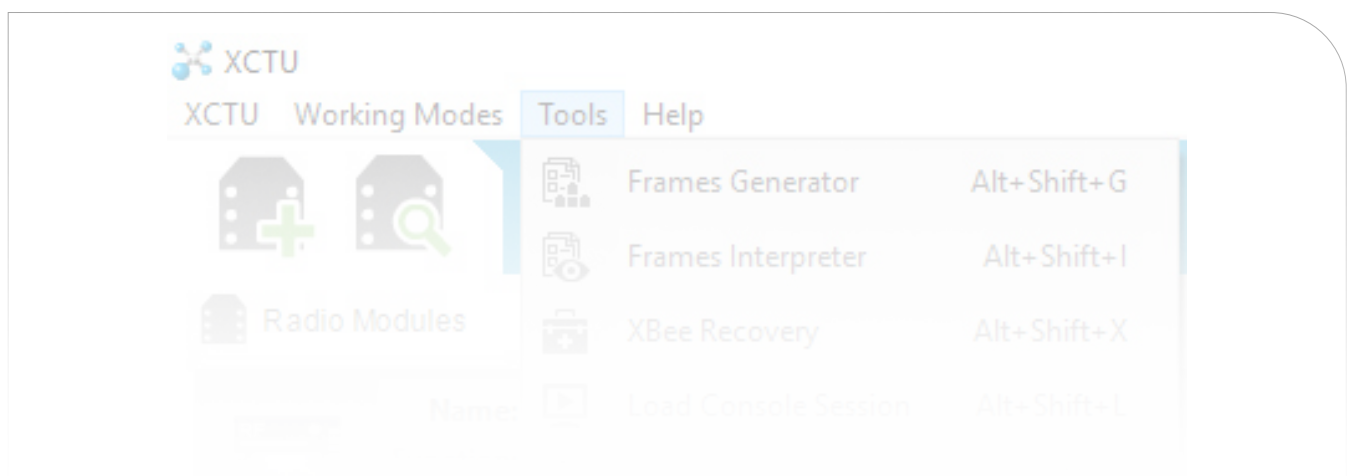
siamo entrare nell'ambiente di sviluppo di MicroPython utilizzando il comando **REPL CTRL+B** e se tutti i passaggi sono andati a buon fine saremo nella situazione riportata in **Figura 4**.

Siamo ora dentro l'ambiente! Per accertarci di quanto detto inizialmente sulle specifiche hardware e sui moduli software disponibili eseguiamo il comando:

```
>>help('modules')
```

Grazie a questo comando otterremo una lista dei moduli software resi disponibili da MicroPython. È lecito chiedersi che cosa sono per Python i moduli, pertanto prima di continuare diamo una rapida definizione a questo concetto. In Python i moduli sono semplicemente dei files usati per raggruppare costanti, funzioni e classi, nulla di più. In altri

linguaggi di programmazione vengono chiamati librerie. Al fine di utilizzare specifiche classi o funzioni di un modulo viene utilizzato lo statement import, vedremo in seguito come utilizzarlo. Inoltre, aggiungiamo a scopo informativo che nel momento in cui il progetto cresce ed insieme ad esso aumentano anche i moduli, è preferibile aggiungere un ulteriore livello di astrazione per evitare di mettere moduli dentro altri moduli. Per rendere concreto questo livello di astrazione ci viene in aiuto il **package**. Con questo termine ci riferiamo ad una collezione di moduli presenti in una cartella dove il nome della cartella corrisponde al nome del package. Non allarmatevi se non avete confidenza con questi termini, nei prossimi appuntamenti vedremo come usarli. Definito questo concetto possiamo tornare al nostro ambiente di MicroPython. In **Figura 5** è riportato l'elenco completo dei moduli disponibili ma i principali che useremo in questa prima fase di avvicina-



**QUELLO CHE HAI LETTO E' UN ESTRATTO, L'ARTICOLO
COMPLETO E' RISERVATO AGLI ABBONATI
AD ELETTRONICA OPEN SOURCE.**

PERCHE' ABBONARSI A PLATINUM 2.0?

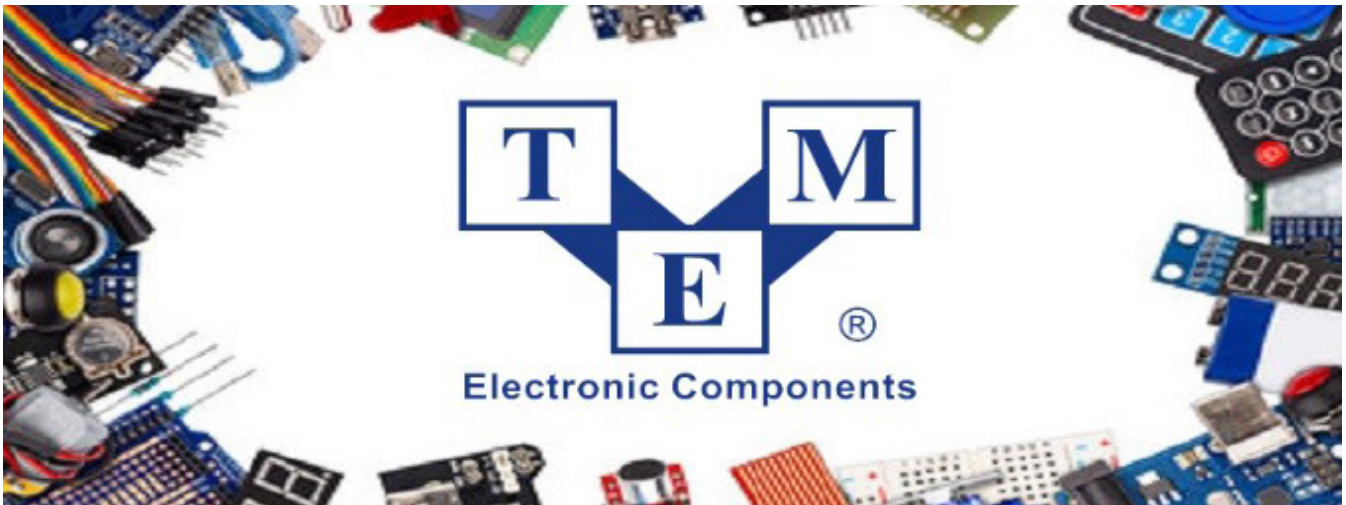
UN ANNO DI **FIRMWARE 2.0**
TUTTI GLI ARTICOLI TECNICI RISERVATI
CONTEST E PROMOZIONI RISERVATI



VOGLIO ABBONARMI!

LE SOLUZIONI ARDUINO PER IL SETTORE EDUCATIONAL DI TME ELECTRONIC COMPONENTS

di Giordana Francesca Brescia



La progettazione embedded è da sempre un processo complesso e articolato, che richiede competenze elettroniche trasversali, legate alla codifica, ai linguaggi di programmazione e all'hardware. I sistemi embedded si basano su microcontrollore e utilizzano sensori per l'acquisizione e la raccolta di dati, offrendo anche un buon livello di integrazione tra microelettronica e protocolli di comunicazione per la realizzazione di progetti in una vasta gamma di campi applicativi. La scelta della giusta scheda o del kit di sviluppo che rispecchierà le specifiche dell'applicazione scelta, è un tema di grande interesse sia tra i designer ed i professionisti embedded, sia tra i makers e gli hobbisti che, ad oggi, rappresentano la parte più creativa del mondo dell'elettronica. L'applicazione finale detterà inevitabilmente considerazioni imprescindibili sulle specifiche richieste, prestazioni, caratteristiche e dimensioni complessive del sistema. L'unica certezza in un processo di progettazione, per quanto complesso possa essere, è che non esiste una soluzione univoca e assoluta, dal momento che ciascun progetto va inteso nella sua unicità e risulta essere diverso da qualunque altro. La parola chiave che guiderà il flusso della progettazione sarà quindi "sperimentazione". Arduino introduce l'elettronica e la programmazione embedded attraverso progetti divertenti e pratici in ambito educational, che permettono di apprendere giocando con la tecnologia. A tal proposito TME Electronic Components offre un'ampia scelta di soluzioni specifiche per l'embedded design, l'IoT e la progettazione elettronica, con una vasta gamma di componenti, schede, moduli di comunicazione, display e kit di sviluppo adatti anche a chi è alle prime armi e desidera costruire un punto di partenza per progetti interattivi futuri.

INTRODUZIONE

L'elettronica e, in particolare, il settore embedded, sono caratterizzati da una costante tendenza alla miniaturizzazione e alla integrazione, implementando funzionalità specifiche della particolare applicazione, anche in architetture standardizzate. La scelta di una scheda di prototipazione rapida è uno step fondamentale nel flusso della progettazione elettronica. **Arduino** è la nota piattaforma elettronica di **contenuti open source, software e hardware**, che vanta una community mondiale di sviluppatori con più di 30 milioni di utenti attivi e più di 500.000 membri del forum Arduino, per la condivisione di idee progettuali innovative, prototipi sperimentali e progetti veri e propri. Ha ispirato migliaia di progettisti nel corso degli anni, dai progetti più semplici di uso quotidiano (domotica open source, dispositivi IoT, droni, controllo motori, etc.) ai più professionali strumenti scientifici di robotica e intelligenza artificiale, ma anche applicazioni nel campo medicale e degli azionamenti elettrici, insomma, un portfolio adatto a tutti i livelli di conoscenza elettronica e programmazione. Ciò è stato reso possibile soprattutto grazie a un'ampia gamma di schede elettroniche (note anche come "boards") idonee ai diversi protocolli di comunicazione, inclusi WiFi, Bluetooth, LoRa, Sigfox, GSM, e alla potente integrazione Arduino-Matlab. Il motivo principale del successo di Arduino risiede proprio nella possibilità di interfacciarsi con un ambiente di sviluppo, schemi elettrici, codici e librerie, completamente open source.

La mission di Arduino è "mettere una delle sue schede e dei suoi kit nelle mani di ogni studente ed insegnante" e venire così incontro alle diversificate esigenze di studenti e docenti durante il loro percorso educativo e formativo, per acquisire le competenze base relative alla programmazione dei microcontrollori ed ai componenti elettronici, attraverso l'uso di uno strumento user friendly sia dal punto di vista hardware che software. Grazie al suo ambiente di programmazione di facile utilizzo (IDE, **Integrated Development Environment**), è possibile sfruttare a pieno le potenzialità di Arduino.

Questo ha fatto di Arduino sia uno strumento molto popolare per lo sviluppo di progetti nell'ecosistema IoT, sia uno dei programmi di maggior successo per l'istruzione, una vera e propria risorsa preziosa per **istruire gli studenti al coding e alla robotica educativa e insegnare l'elettronica partendo da zero**, in un ampio ventaglio di possibili applicazioni, dall'acquisizione dati in tempo reale, alla realizzazione di dispositivi wearable e robot, sino al controllo di fenomeni fisici, utilizzando i classici strumenti (librerie, shields, sensori, attuatori, breadboard, LED, diodi, poten-

ziometri, resistenze, motori DC, servomotori, display LCD, e altra componentistica elettronica).

Designer, ingegneri, sviluppatori, produttori e studenti in tutto il mondo utilizzano Arduino per portare innovazione nel settore automobilistico, robotico, agricolo, dei dispositivi intelligenti, della domotica, della musica, del gaming, e innumerevoli altre applicazioni. Nelle scuole Arduino viene utilizzato come base per progetti ludici ed educativi che orientano lo studente al pensiero computazionale, al problem solving e alla codifica, mediante giocattoli integrati con la tecnologia Arduino, in grado di stimolare le capacità di costruzione e la logica, e si adatta anche a un pubblico di utilizzatori principianti, privo di uno specifico background di programmazione. Arduino è anche uno strumento per la didattica e l'istruzione nelle discipline **STEAM** (Scienze, Tecnologia, Ingegneria, Arte e Design, Matematica) per l'apprendimento interdisciplinare con modalità didattiche basate sulle nuove tecnologie, secondo l'approccio "*learning by doing*". Inoltre, la scheda Arduino è ampiamente adottata da startup e aziende per semplificare lo sviluppo dell'**Internet of Things (IoT)**, dal prototipo sperimentale alla produzione.

ARDUINO ENGINEERING KIT

TME Electronic Components offre un'ampia gamma di strumenti per la prototipazione di progetti elettronici, tra i quali [diverse schede di sviluppo per Arduino](#), kit di sviluppo ed altre soluzioni basate su schede e shields con esse compatibili. Per soddisfare le esigenze del settore educational e fornire così uno strumento di avviamento, **Arduino Engineering Kit** è la soluzione ideale per la prototipazione di progetti all'insegna dell'innovazione. [Arduino Engineering Kit](#) rappresenta la soluzione educational adatta a studenti universitari, fornendo un'implementazione pratica e all'avanguardia della tecnologia Arduino in un ambiente educativo. Il kit è rivolto principalmente a tre tipologie di utenti: studenti che studiano ingegneria presso un'università o una scuola secondaria, professori che insegnano ingegneria e desiderano interfacciarsi anche con risorse pratiche, makers e appassionati di progettazione elettronica con un interesse o un background nella progettazione, a livello professional o solo come hobby. L'Arduino Engineering Kit è un kit divertente per chiunque, utile per stimolare la creatività e le abilità progettuali. Il kit consente la realizzazione di progetti in modo semplice e rapido, anche a chi non possiede conoscenze approfondite di elettronica e programmazione.

In particolare, il kit include tre progetti all'avanguardia basati su Arduino in modo che gli studenti possano appren-



Figura 1. Arduino Engineering Kit - il pratico box che contiene i componenti per l'assemblaggio



- **Whiteboard Drawing Robot:** il robot può prendere un disegno che gli è stato dato e duplicarlo su una lavagna.

Il kit, con una tensione di alimentazione 5÷12 VDC, viene venduto in un contenitore di plastica rigida, che ricorda la classica "cassetta degli attrezzi", impilabile per la conservazione e per il suo riutilizzo, e dal peso di appena 1760 g (**Figura 1** e **Figura 2**). All'interno della confezione è presente una scheda **Arduino MKR1000** di facile utilizzo, diverse parti personalizzate e un set completo di componenti elettrici e meccanici necessari per assemblare tutti e tre i progetti menzionati sopra, quali alimentatore, batteria, componenti meccanici, motori elettrici DC, Arduino MKR motor shield, web-cam, caricabatteria, cavo USB per la comunicazione con il computer, ruote, modulo a ultrasuoni, sensori hall.

Il contenuto del kit è strutturato in sei capitoli, con una breve introduzione, una guida introduttiva per gli strumenti che verranno utilizzati, una sezione con i concetti più importanti e infine i progetti.

L'elenco dettagliato dei componenti (**Figura 3**) del kit è il seguente:

- Arduino MKR 1000;
- Arduino MKR Motor Shield;
- Arduino MKR Imu Shield;
- motore DC;
- web-cam;
- batteria ai polimeri di litio;
- caricabatterie;
- cavo USB.

QUELLO CHE HAI LETTO E' UN ESTRATTO, L'ARTICOLO COMPLETO E' RISERVATO AGLI ABBONATI AD ELETTRONICA OPEN SOURCE.

PERCHE' ABBONARSI A PLATINUM 2.0?

UN ANNO DI **FIRMWARE 2.0**
TUTTI GLI ARTICOLI TECNICI RISERVATI
CONTEST E PROMOZIONI RISERVATI



VOGLIO ABBONARMI!

SMALL CIRCUITS REVIVAL – RELÈ AD ALTA EFFICIENZA ENERGETICA



I circuiti relè discussi in questo articolo possono essere collegati in serie con un carico (lampada) e una serie di pulsanti normalmente chiusi. Quando uno di questi interruttori viene premuto, la spia si accende per un certo periodo di tempo. Trascorso questo tempo, la lampada si spegnerà automaticamente.

INTRODUZIONE

Questo somiglia un pò al familiare circuito delle luci delle scale, in cui è possibile accendere l'illuminazione in una tromba delle scale o in un corridoio con la semplice pressione di un pulsante e spegnerla di nuovo con la semplice pressione di un altro pulsante. Tuttavia, **i circuiti qui descritti si spengono automaticamente e sono in linea di principio molto più rispettosi del consumo energetico.**

VARIANTE 1

La prima variante (Figura 1) non è difficile da comprende-

re. Il circuito funziona con una fonte di 12 VDC (batteria, adattatore di rete) e può quindi essere ricostruito in sicurezza anche da un principiante. Immediatamente dopo aver collegato il circuito, il condensatore C1 verrà caricato dalla lampada LA1 e dal diodo D1 tramite i pulsanti normalmente chiusi S1...Sn (ne sono stati presi solo tre) collegati in serie. La lampada si accenderà per un pò. Il LED1 indica che il circuito è armato. Per la cronaca: la corrente che scorre attraverso la lampada a riposo è troppo bassa per farla accendere.

Non appena uno dei pulsanti viene azionato, non può più fluire corrente da C1 a D1. La base del transistor T1, che

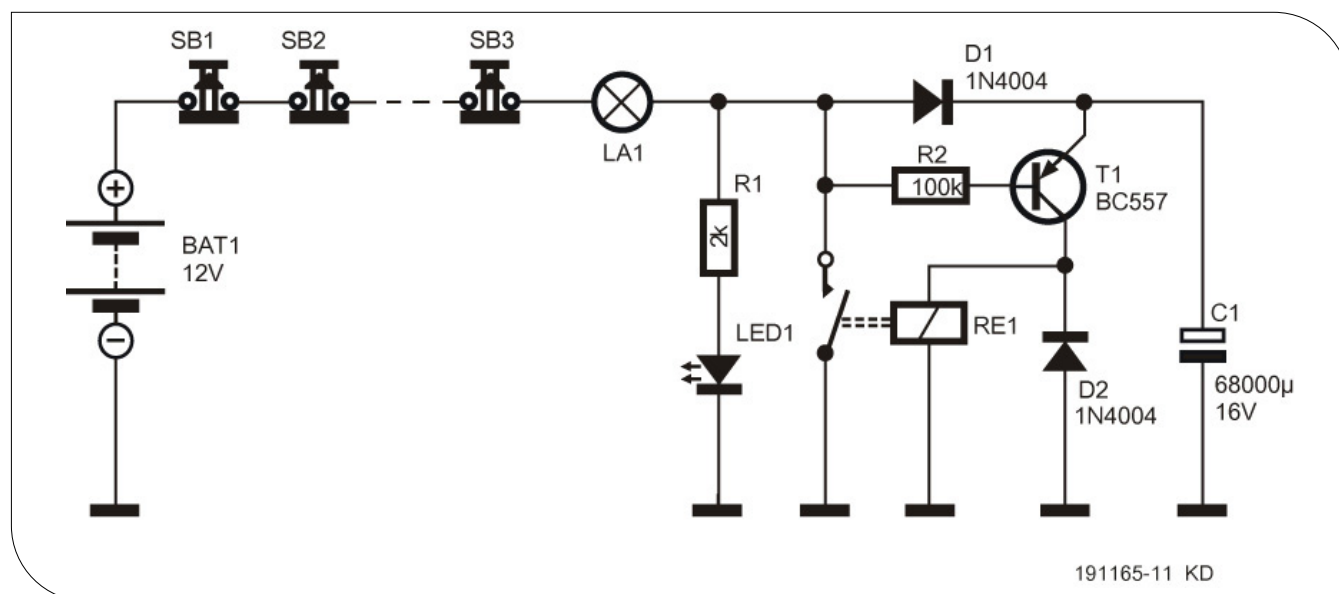


Figura 1. Schema di funzionamento

tramite la lampada è tenuta al potenziale + dell'alimentatore, viene ora portata a massa tramite la resistenza R1 e il LED1. Risultato: il transistor si accende e il condensatore C1 si scarica a massa tramite la bobina del relè. Il relè è eccitato e tramite il contatto del relè c'è ora abbastanza corrente per mantenere la lampada accesa. Inoltre, la corrente di carica in C1 viene interrotta. Dopo un pò di tempo, il condensatore si è scaricato così tanto che il relè non può più rimanere attivato. Il contatto del relè si apre e lo stato iniziale viene ripristinato (per cui si presume che il pulsante non sia tenuto premuto). Per tutto il tempo in cui la lampada rimane accesa, si applica la seguente relazione approssimativa:

$$t = 0.67 \times R_{Relay} \times C1 \quad [\text{secondi, ohm, farad}]$$

Il tipo EMR menzionato nel diagramma ha una resistenza della bobina di 1050 Ω; per il tipo G6DS, è pari a 1200 Ω. Con il primo relè e un condensatore da 68.000 µF si arriva a circa 40 s. Il valore del resistore R1 deve essere scelto in modo che la corrente attraverso il LED1 sia abbastanza grande da farlo accendere debolmente. Il valore menzionato nel diagramma sarà generalmente sufficiente, ma c'è spazio per qualche sperimentazione. **Attenzione: questo circuito NON deve essere utilizzato per lampade collegate alla linea di alimentazione AC (rete)!** La seconda variante sarà discussa nel prossimo articolo.

WEB LINK

ARTICOLO ORIGINALE IN INGLESE AL LINK: [SMALL CIRCUITS REVIVAL – RELAY](#)

THE BIGGEST EMBEDDED COMMUNITY IN ITALY

CATEGORIES

COMPANIES/CONSULTANTS
53 %

ACADEMICS/STUDENTS
25 %

MAKERS/HOBBYISTS
22 %

SOCIAL CONNECTIONS

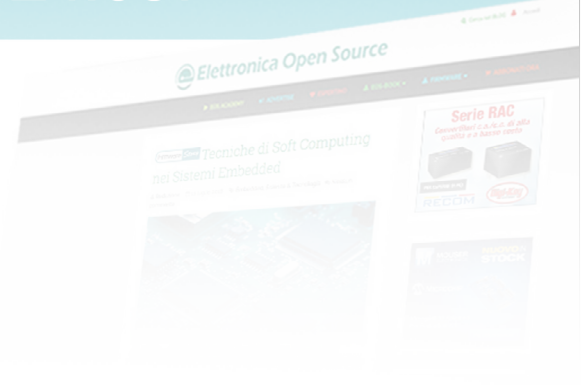
f + 83.000

in + 23.000

+ 130.000
REGISTERED USERS

6.138 AVERAGE DAILY PAGEVIEWS (DEC2019)

824.057 2019 ANNUAL VISITORS



QUELLO CHE HAI LETTO E' UN ESTRATTO, L'ARTICOLO COMPLETO E' RISERVATO AGLI ABBONATI AD ELETTRONICA OPEN SOURCE.

PERCHE' ABBONARSI A PLATINUM 2.0?

UN ANNO DI **FIRMWARE 2.0**
TUTTI GLI ARTICOLI TECNICI RISERVATI
CONTEST E PROMOZIONI RISERVATI



VOGLIO ABBONARMI!

PROGETTIAMO I NOSTRI FILTRI CON FILTERPRO DELLA TEXAS INSTRUMENTS

di Giovanni Di Maria

Scopriamo assieme un software davvero notevole, prodotto dalla Texas Instruments. Si tratta di FilterPro, un programma semplice e potente al tempo stesso che consente di progettare filtri attivi di diversi tipi. Per il suo utilizzo è sufficiente inserire i parametri di risposta desiderata ed effettuare alcune scelte di configurazione per ottenere schemi personalizzati di filtri attivi. Dopo aver provato il programma, un progettista non ne può fare più a meno.

IL FILTRO ATTIVO

In tutti i libri di testo di elettronica e in qualsiasi tutorial che si rispetti, esiste sempre un capitolo dedicato ai filtri. Senza approfondire qui la teoria che sta alla loro base, diciamo solo che un filtro attivo è un **circuito a frequenza selettiva** che ha lo scopo di lasciar passare solo

i segnali di una banda di frequenza desiderata e bloccare quelli al di fuori di essa. Il filtro attivo utilizza componenti elettronici attivi, quali **amplificatori operazionali e transistor**. Esso, sicuramente, comporta maggiori vantaggi dei filtri passivi che, tuttavia, sono utilizzati nei progetti più semplici ed economici. La funzionalità di un filtro è come

quella di un vigile urbano: esso lascia passare (o bloccare) solo un determinato tipo di veicolo, secondo le esigenze del traffico (vedi **Figura 1**).

COME OTTENERE FILTERPRO

E' sufficiente accedere all'indirizzo <http://www.ti.com/filterpro> e cliccare sull'apposito link, come evidenziato in **Figura 2**. Occorre prima registrarsi gratuitamente al sito per iniziare la procedura di download che copierà, sul proprio hard disk, il file d'installazione. Dopo lo scaricamento e relativa installazione, un'icona sul desktop attesta la corretta operazione avvenuta e si può avviare il programma, mediante doppio click sulla stessa. Ricordiamo che esistono le seguenti tipologie di filtro, come mostrato anche in **Figura 3**:

- filtro passa basso;
- filtro passa alto;
- filtro passa banda;

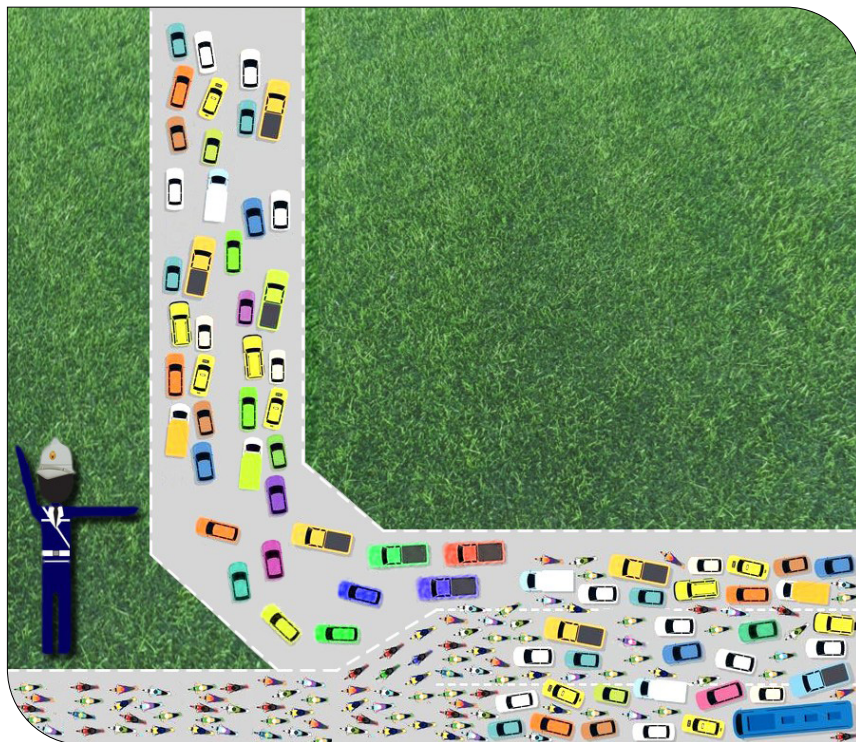


Figura 1: il filtro è come un vigile urbano: lascia passare (o bloccare) solo un determinato tipo di veicolo

The screenshot shows the Texas Instruments website interface. At the top, there is a search bar and navigation links. The main content area is titled "WEBENCH Active Filter Designer" and includes a "Key Document" section with a link to the "FilterPro™ MFB and Sallen-Key Low-Pass Filter Design Program User Guide (Rev. C)". Below this is a "Description" section with a "View FilterPro Design Wizard Screen" link. A green arrow points to the "Download FilterPro v3.1" link in the description. The "Features" section is partially visible at the bottom.

Figura 2: per scaricare FilterPro è sufficiente cliccare sull'apposito link

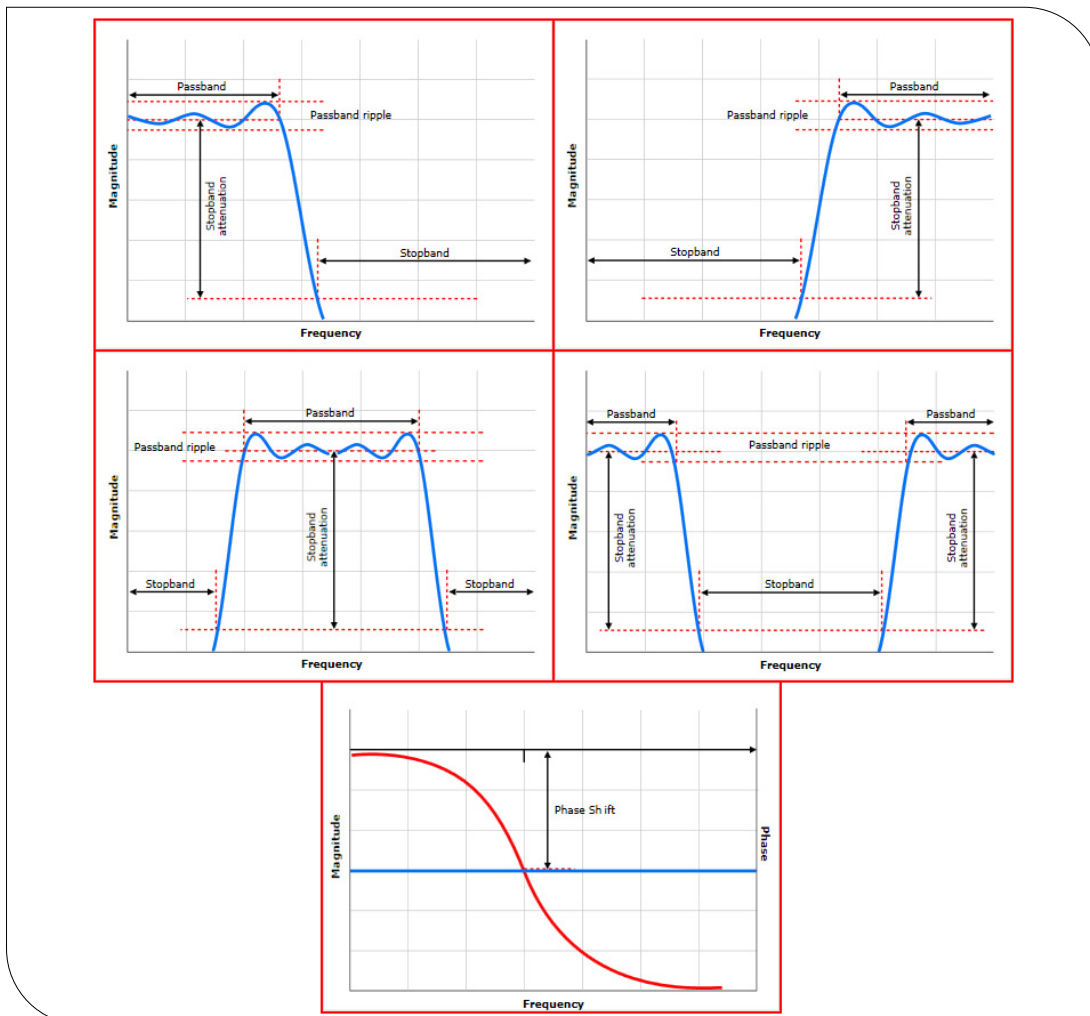


Figura 3: i vari tipi di filtri

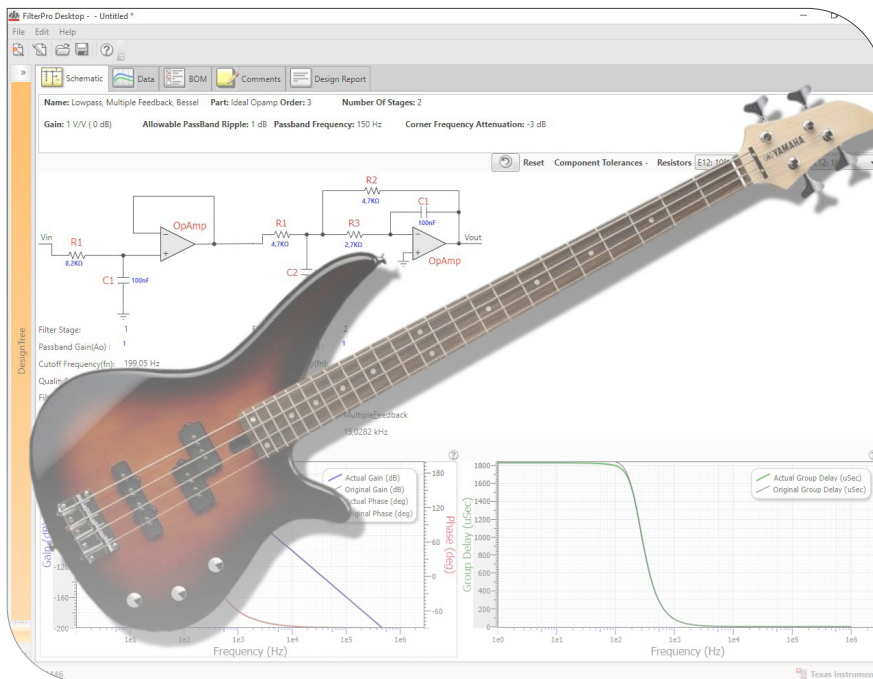
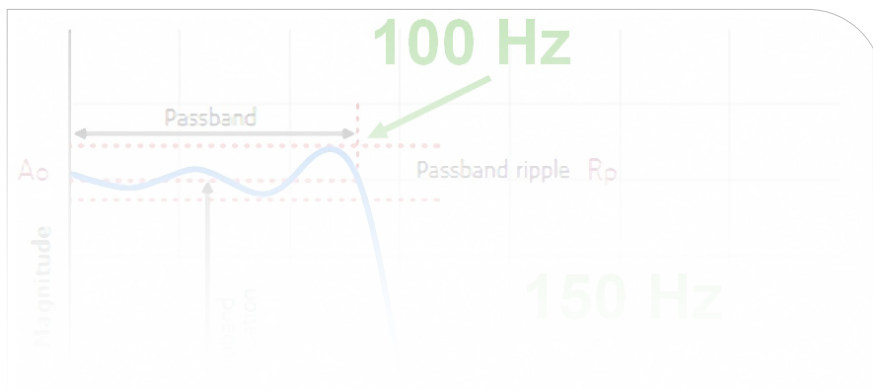


Figura 4: il progetto di un filtro attivo per basso elettrico



per questa tipologia di applicazioni. Ovviamente, i valori di taglio sono solo di esempio e sono stati scelti arbitrariamente, ai fini della prova. Il progettista può studiare e adottare i parametri secondo le sue necessità. Il filtro che andiamo a preparare deve possedere le seguenti caratteristiche:

- deve essere di tipo **passa basso**, per tagliare le frequenze superiori a una certa soglia;
- deve avere una frequenza di taglio di 100 Hz;
- deve essere caratterizzato da una frequenza di elimina banda di 150 Hz con relativa attenuazione di ben -45dB.

Tali parametri progettuali sono ben osservabili in Figura 5.

Il progetto è costantemente accompagnato da un semplice Wizard che chiede, passo per passo, tutte le informazioni necessarie e i parametri per la sua realizzazione. Dopo aver cliccato sul menù File->New->Design ha inizio la procedura guidata vera e propria.

PRIMA FASE: SCELTA DEL

QUELLO CHE HAI LETTO E' UN ESTRATTO, L'ARTICOLO COMPLETO E' RISERVATO AGLI ABBONATI AD ELETTRONICA OPEN SOURCE.

PERCHE' ABBONARSI A PLATINUM 2.0?

**UN ANNO DI FIRMWARE 2.0
TUTTI GLI ARTICOLI TECNICI RISERVATI
CONTEST E PROMOZIONI RISERVATI**



VOGLIO ABBONARMI!

+ 130.000

REGISTERED USERS

6.138 AVERAGE DAILY PAGEVIEWS (DEC2019)

824.057 2019 ANNUAL VISITORS

THE BIGGEST
EMBEDDED
COMMUNITY
IN ITALY

CATEGORIES

COMPANIES/CONSULTANTS

53 %

ACADEMICS/STUDENTS

25 %

MAKERS/HOBBYISTS

22 %

SOCIAL CONNECTIONS

f + 83.000

in + 23.000

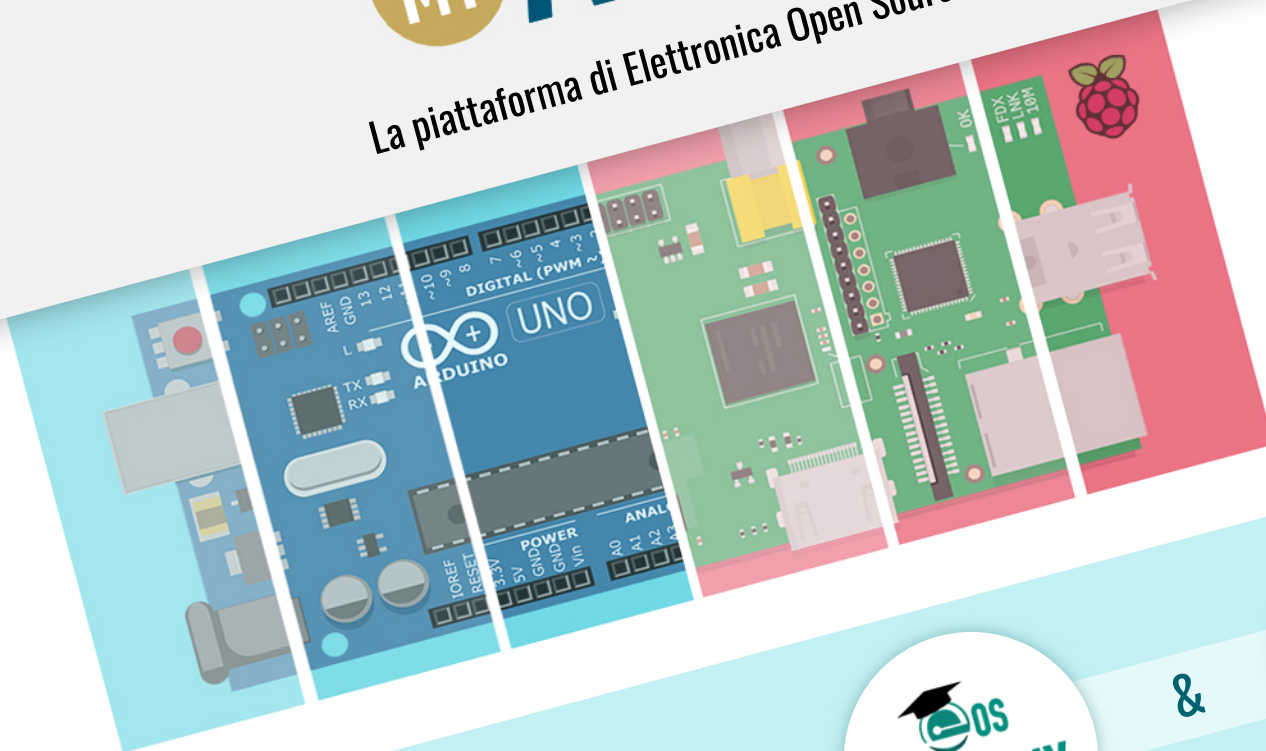


I NOSTRI CORSI DI ELETTRONICA
PER I PROFESSIONISTI
E I MAKERS



ACADEMY

La piattaforma di Elettronica Open Source dedicata ai corsi



&



PUOI AVERE TUTTI I CORSI DI

A PORTATA DI CLICK

