

Firmware

it.emcelettronica.com

Applicazioni audio high-quality



- Convertitore buck-boost
- Un basso consumo è fondamentale per l'IoT
- DesignCon: Where the chip meets the board
- Un sistema per sviluppare su micro da 16 o 32 bit
- OpenBCI e le onde cerebrali
- STM32 Discovery: Introduzione ed IDE



Uno strumento di lavoro!

Erano i primi mesi del 2006, mi ricordo che stavo progettando FTPmicro (il primo webserver di 8 cm) e mentre leggevo una rivista di elettronica, una di quelle presenti allora in edicola, pensavo: “non è possibile, ancora lo stesso articolo, non ne posso più di tutorial sull’RS232!!”.

Quella frustrazione nasceva proprio dal fatto che sulle riviste dell’epoca si leggevano sempre gli stessi articoli, ma non solo, in un contesto storico dove il web iniziava a prendere piede, questi articoli era facile trovarli liberamente online. Premetto che a quei tempi avevo tutte, dico tutte le riviste di elettronica disponibili, arretrati compresi. Mi ricordo che dalla mia assistente feci realizzare un foglio excel con tutti i numeri, titoli ed integrati utilizzati, in modo da poter risalire subito all’articolo relativo al progetto che stavo realizzando.

“ Ho da sempre considerato le riviste di elettronica uno strumento di lavoro!

Purtroppo però le mie aspettative furono deluse. Decisi allora di fare qualcosa, di realizzare io stesso degli articoli di elettronica in base alle mie esperienze e di coinvolgere amici progettisti elettronici nella condivisione della loro esperienza, dei loro progetti. Con il motto “share for life” nacque Your Electronics Open Source, quindi Elettronica Open Source.

Nello stesso periodo, qualcun altro forse ebbe un’idea simile dando vita a Firmware! Rivista che scoprii subito e da subito divenne la mia rivista di elettronica preferita e dopo poco, l’unica rivista italiana di elettronica che ho continuato a leggere.

Perché? Semplicemente perché è veramente

“ uno strumento di lavoro!

Con l’annessione di Firmware ad Elettronica Open Source, EOS-Book diventerà sempre più uno strumento di approfondimento a 360 gradi, con letture interessanti, sia per apprendere di nuove

l'Editoriale

tecnologie, sia per realizzare progetti. Insomma una rivista per gli amanti della tecnologia, hobbisti e makers.

Il nostro impegno nel tempo sarà quindi di soddisfare le richieste e far accrescere il know-how sia dei makers che dei professionisti.

Il primo numero della nuova gestione di Firmware sarà disponibile gratuitamente per tutti gli abbonati Platinum!

“ Avrete quindi a disposizione sia uno strumento di lavoro che uno strumento di divertimento che vi permetteranno di accrescere il vostro sapere in modo professionale e divertente allo stesso tempo!

P.S. Di RS232 ne abbiamo parlato anche su EOS ma in modo diverso

Emanuele Bonanni



SETTEMBRE 2015

Firmware

TOOLS

- 5 • Un sistema per sviluppare su micro da 16 o 32 bit**
- 16 • OpenBCI e le onde cerebrali**
- 42 • STM32 Discovery: Introduzione ed IDE**

ANALOG

- 11 • Un basso consumo è fondamentale per l'IoT**
- 25 • Convertitore buck-boost**
- 35 • Applicazioni audio high-quality**

TIPS'N TRICKS

- 53 • DesignCon: Where the chip meets the board**

ILLUMINOTRONICA

HUMAN CENTRIC LIGHTING TECHNOLOGIES

PADOVA 8/10 OTTOBRE 2015

www.illuminotronica.it



LED 
LIGHTING

soluzioni & design

per chi progetta, produce
installa corpi e strutture illuminanti
outdoor e indoor


SMART HUT

***domotica
della
visione & percezione***

prodotti e tecnologie hw / sw
per generare e controllare
illuminazione e sicurezza


elettro & micro

***up to date
elettronica industriale***

il riferimento di Assodel
per tecnici e imprese
***unico ora in Italia**



assodel



**TECNO
IMPRESE**



Un sistema modulare per lo sviluppo su architetture a microcontrollore da 16 o 32 bit? Eccolo qui!



di **Piero Boccadoro**
17 Settembre 2015

Tutti gli sviluppatori ed i programmatori, anche i più esperti, una volta scelto l'hardware di riferimento, hanno bisogno di sistemi adatti allo sviluppo, versatili e performanti, che siano in grado di offrire loro le giuste opportunità e gli strumenti più idonei per realizzare e testare la loro applicazione. La ricerca di una piattaforma adeguata è parte integrante del progetto ed il giusto consiglio può rendere questa fase sicuramente più semplice e meno problematica. Tutti coloro che hanno già esperienza con MPLAB saranno sicuramente felici di scoprire Explorer 16, una scheda che viene strutturata come demo board per mostrare in maniera semplice ed efficace le incredibili potenzialità che derivano dal completo supporto di dispositivi appartenenti a diverse famiglie, come ad esempio PIC24, dsPIC e PIC32. Vediamo qui insieme di che si tratta.

MPLAB

Prima di iniziare a parlare della scheda per capire meglio caratteristiche e punti di forza, è fondamentale spiegare come funziona questo ambiente di programmazione perché il suo ruolo, di supporto e verifica, è fondamentale e centrale per il corretto sviluppo e la progettazione. MPLAB® X IDE è un software dedicato che funziona su PC (Windows®, Mac OS®, Li-

nux®) e serve a realizzare applicazioni per microcontrollori Microchip e DSP più in generale. Ne esistono diverse versioni, in verità, riferite prevalentemente al tipo di microcontrollore al quale si rivolgono.

MPLAB 8.X, in particolare, è la release basata su Microsoft Visual C++ e supporta la gestione dei progetti, l'editing, il debug e la programmazione di microcontrollori PIC Microchip da 8, 16



e 32-bit. Questa release funziona soltanto con Microsoft Windows ed è ancora disponibile negli archivi Microchip ma non è consigliabile utilizzarla, dal momento che esiste una versione più nuova, aggiornata e versatile, come vedremo tra un attimo. Ciò nondimeno, è importante specificare che già questo IDE prevedeva supporti per i seguenti compilatori:

- **MPLAB MPASM Assembler**
- **MPLAB ASM30 Assembler**
- **MPLAB C Compiler for PIC18**
- **MPLAB C Compiler for PIC24 and dsPIC DSCs**
- **MPLAB C Compiler for PIC32**
- **HI-TECH C**

La versione X di MPLAB ha introdotto una serie di modifiche nella toolchain cui i programmatori e gli sviluppatori erano abituati utilizzando microcontrollori ad 8 bit. Prima di tutto, la modifica più importante è la possibilità di girare su tutte le piattaforme: Windows, MAC Os X e Linux, il che rende la soluzione estremamente più versatile ed adatta a tutti i progetti ed i progettisti.

MPLAB X si basa su una tecnologia open-source che prende il nome di NetBeans di Oracle e supporta i seguenti compilatori:

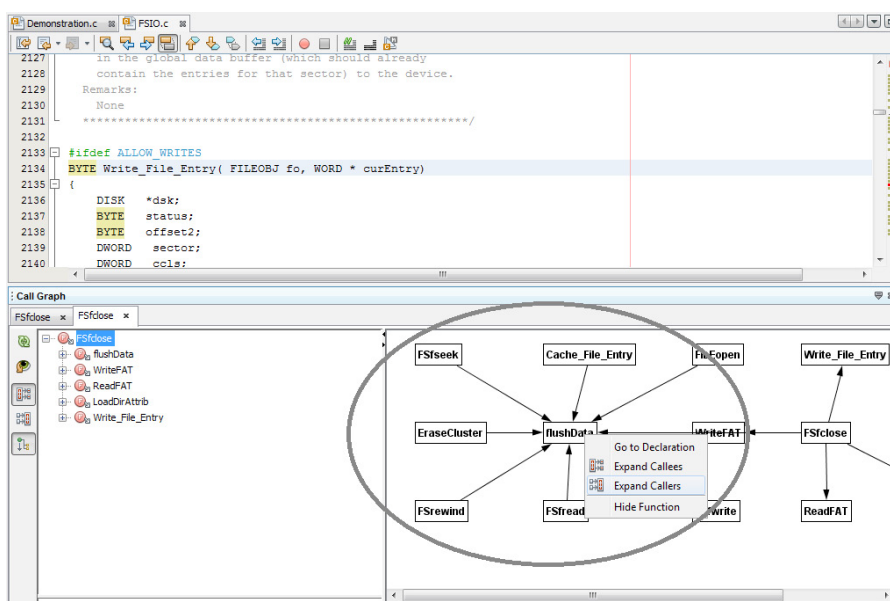
- **MPLAB XC8 - C compiler for 8-bit PIC devices**
- **MPLAB XC16 - C compiler for 16-bit PIC devices**
- **MPLAB XC32 - C++ compiler for 32-bit**

PIC devices

- **HI-TECH C - C compiler for 8-bit PIC devices**
- **SDCC - open-source C compiler**

Tra le sue features, troviamo:

- **Call Graph, una nuova funzione ottima per esplorare codice complesso;**
- **il supporto per diverse configurazioni all'interno dei progetti;**
- **il supporto per più versioni dello stesso compilatore;**
- **il supporto per diversi tool di debug;**
- **Live Parsing.**

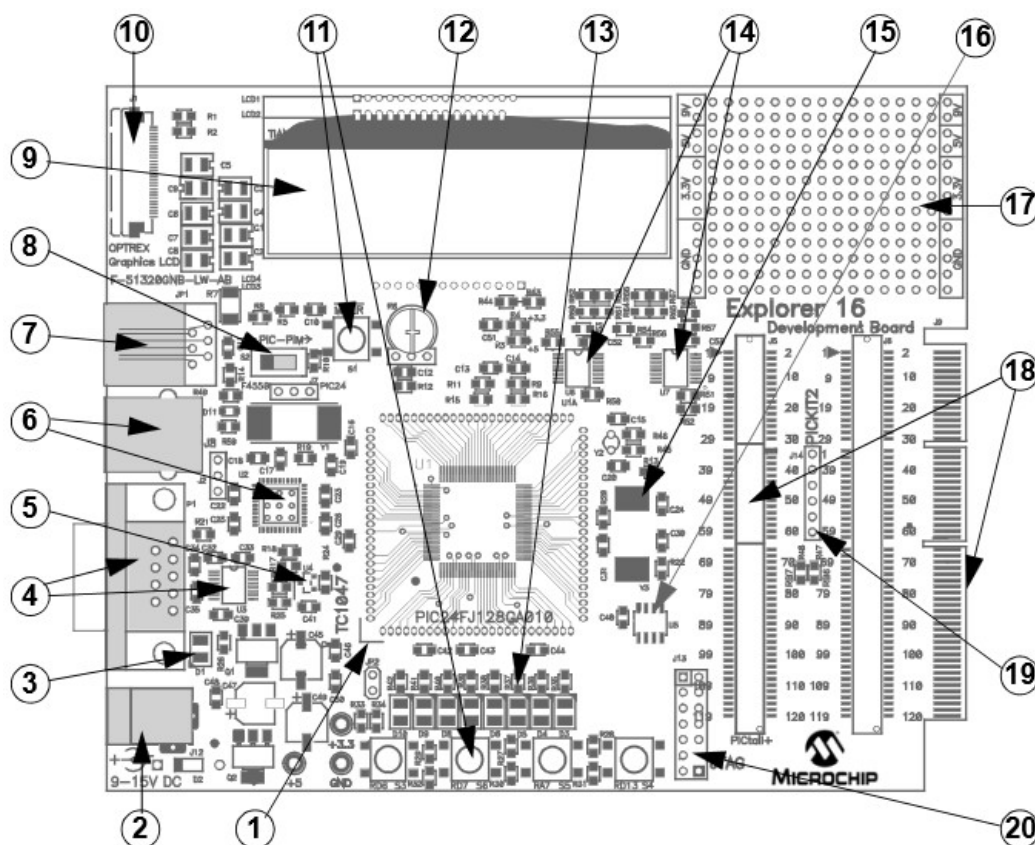


Grazie a questa nuova versione è anche possibile importare i progetti realizzati in MPLAB® 8 IDE, in maniera tale da non perdere il lavoro già realizzato. L'approccio proposto da questa nuova versione e gli interventi di aggiornamento già realizzati sono il risultato di un feedback diretto recepito dall'azienda sulle funzioni più utili e più richieste. Il programma, inoltre, si dimostra facilmente aggiornabile soprattutto in termini di



architetture supportate. MPLAB è stato progettato per lavorare con dispositivi MPLAB-certified come **MPLAB ICD 3** e **MPLAB REAL ICE**. L'ambiente di cui stiamo parlando consente, quindi, un'emulazione completa ma anche una serie di possibilità, in termini di debug, di caratura professionale. Senza contare che è possibile gestire interfacce a 3V e farle funzionare con dispositivi, e periferiche, da 5V.

che segue e poi spiegate per esteso:



VEDIAMO LA SCHEDA

Explorer 16 è davvero un'ottima opportunità, sia didattica sia di sviluppo per chi ha già esperienza. La scheda include un connettore PICtail Plus che consente l'aggiunta di schede di espansione e di interfaccia tra cui USB, CAN ed Ethernet ma anche Wireless, grafiche e molto altro.

Se utilizzata insieme ad MPLAB ICD 3 In Circuit Debugger oppure MPLAB REAL ICE, è possibile realizzare un'emulazione real-time, il che rende lo sviluppo estremamente più semplificato e veloce.

La scheda è organizzata in maniera estremamente essenziale e propone una serie di features che vediamo dapprima illustrate nella figura

Il diagramma richiama features numerate spiegate qui di seguito:

1. **100-pin PIM, compatibile con PIM version di alcune MCU Microchip PIC e dsPIC DSC;**
2. **Power input 9 VDC che fornisce +3.3V e +5V;**
3. **Power LED;**
4. **porta RS-232;**
5. **sensore di temperatura onboard;**
6. **[Non supportata] connettività USB;**
7. **Standard, 6-wire In-Circuit Debugger (ICD) per moduli MPLAB ICDprogrammer/debugger;**
8. **PIM Hardware selection o microcontrollore saldato a bordo;**



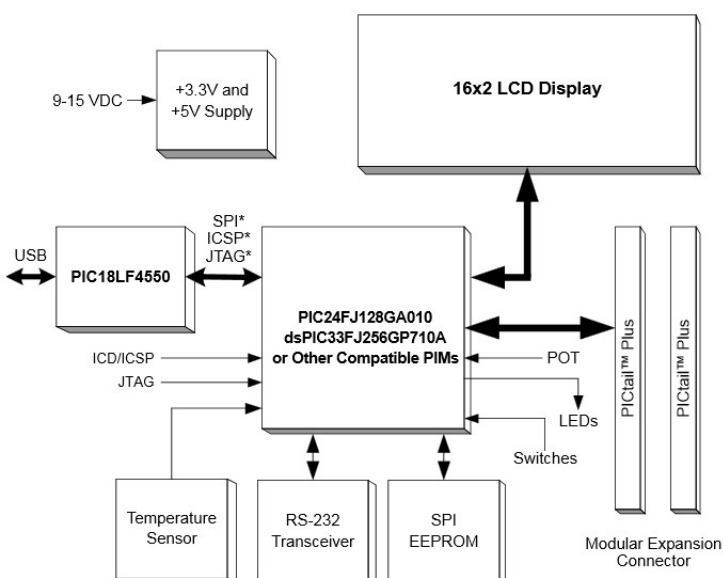
- 9. LCD 16x2;
- 10. Provisioning on PCB per l'aggiunta di un LCD;
- 11. Push buttons per reset o input user-defined;
- 12. Potenzziometro per input analogici;
- 13.8 LED;
- 14. multiplexer 74HCT4053;
- 15.8 Mhz e 32.768 kHz dedicati rispettivamente per il micro e l'RTC;
- 16. Serial EEPROM
- 17. Prototype area;
- 18. Socket and edge connector per PICtail Plus
- 19.6-pin interface for the PICkit Programmer
- 20. JTAG connector

L'utilizzo di alcune di queste caratteristiche funzionali risulta estremamente semplice ed immediato ma probabilmente avrete bisogno di un riferimento per cui date uno sguardo alla [guida dell'utente](#) che funziona da manuale di riferimento.

Delle features che abbiamo elencato, alcune risultano particolarmente interessanti soprattutto perché verrebbero utilizzate in ogni caso, anche soltanto per iniziare con un buon livello prototipale, aggiungendole all'occorrenza. L'utilizzo del display, infatti, rappresenta il modo più immediato per ottenere messaggi di errore, controllare lo stato del sistema e altrettanto va detto per l'utilizzo di sensori elementari come quello di temperatura, per esempio.

“ Questi sono, dunque, alcuni dei motivi per i quali riteniamo la dotazione estremamente completa e funzionale.

Per mostrare con più completezza il funzionamento e la logica della scheda, vi riportiamo qui di seguito il diagramma a blocchi della stessa:



FEATURES EXPLAINED

Non sempre e non tutto può essere chiaro della documentazione di una scheda, anche al programmatore più esperto. Per questo adesso vediamo insieme alcune delle caratteristiche funzionali più interessanti, come funzionano e come possono essere utilizzate. Abbiamo introdotto il supporto per i microcontrollori ed in effetti una delle caratteristiche più importanti di Explorer 16 è proprio la possibilità di utilizzarne diversi. PIMs sono indicati semplicemente ed in maniera immediata così da facilitare l'installazione, soprattutto basandosi sull'indicazione



dell'orientamento del micro. La scheda non viene, peraltro, fornita con un microcontrollore non rimovibile.

Parlando di **alimentazione**, esistono due modi per alimentare la scheda; il primo è con un'alimentazione DC da 9 a 15 V (dei quali il primo valore è quello consigliato) al connettore J12. Le funzionalità di base vengono raggiunte con un valore di corrente pari a 250 mA. Dal momento che la scheda può essere utilizzata come una piattaforma di sviluppo modulare da connettere anche ad altre schede di espansione, sono previsti Q1 e Q2, due regolatori di tensione con un valore di corrente massima pari ad 800 mA. Potrebbe, per alcune applicazioni, essere richiesto un valore di corrente superiore, addirittura fino a 1.6A; ciò nondimeno nella documentazione si dice chiaramente che questa possibilità è da escludersi, se si intende utilizzare il sistema sul lungo termine.

L'altra possibilità riguarda l'utilizzo di un **alimentatore DC**, sempre esterno, che sia in grado di fornire 5 V come anche 3.3V all'ingresso S3. La scheda è dotata di un LED verde (D1) il cui scopo è quello di segnalare se c'è l'alimentazione. Quando è acceso, il LED indica, infatti, la presenza della tensione 3.3V.

Abbiamo parlato di periferiche e connettività ed in particolare l'**interfaccia seriale RS-232** non manca. U3, un level shifter, è stato implementato con tutto l'hardware di supporto per la connessione al DB9. La porta è configurata come dispositivo DCE ed i dispositivi PIC e dsPIC sono connessi a RX2 (DIN2) e TX2 (DOUT2) del MAX3232.

Il Microchip **TC1074A**, nello schema indicato come U4, invece, è proprio il sensore di temperatura analogico cui abbiamo accennato in precedenza, già connesso all'ADC e pronto all'uso. Non è l'unico, in effetti, dal momento che è previsto anche un **potenziometro**, che provvede ad un input analogico, da 10 k. Esso risulta connesso ad un resistore in serie ad AN5 e può essere controllato facendo variare il valore tra VDD e GND.

Altro aspetto davvero interessante è la presenza del **display**, un alfanumerico da 16 caratteri per due righe. Esso può essere pilotato del modulo PMP, se supportato, oppure più facilmente dagli I/O. Una configurazione alternativa, infatti, coinvolge R60/62/64/66 il che prevede che si creino dei corti tra R61/63/65/67.

La dotazione è ricca anche di **indicatori LED** perché, oltre quello di cui abbiamo parlato, ci sono 8 diodi, da D3 a D10, disponibili. Essi possono essere disabilitati tutti utilizzando il jumper JP2 o controllati singolarmente pilotando gli ingressi. Anche questi sono disponibili per la programmazione a vario titolo e pertanto è possibile scegliere la loro funzione in ragione delle specifiche esigenze.

La dotazione continua con una **EEPROM 25LC256 256K (32K x 8)**, nello schema indicata con U5, che viene utilizzata per la memorizzazione del firmware. Come dicevamo è disponibile anche il connettore per il programmatore PICkitTM, J14, e il connettore JTAG al J13.

Vi segnaliamo che l'azienda produttrice ha lavorato in maniera estremamente scrupolosa sulla scheda e pertanto, soprattutto per quanto



riguarda le revisioni 4 e 5, esistono delle osservazioni da fare sul funzionamento dell'interfaccia RS-232 e sull'alimentazione, in particolare per quanto concerne le connessioni.

Uno dei problemi rilevati è operativo: in una particolare configurazione il microcontrollore opera al di fuori delle tolleranze specificate. Questo è evidente proprio nell'utilizzo del modulo display, che potrebbe restituire i valori convertiti dall'ADC che sono, però, più alti di quanto ci si sarebbe aspettato. Questo vuol dire che i comparatori stanno funzionando al di fuori dei loro valori di riferimento e pertanto c'è da studiare il metodo attraverso cui risolvere questo problema. Si tratterebbe, ad un occhio più attento, di un problema che sarebbe interessante risolvere per il progettista o per il programmatore che si trovino a lavorare su scheda. Il primo passo per la diagnosi di questo problema è l'utilizzo del voltmetro su tutti i pin dei display per verificare il valore corretto. Tale problema è certificato essersi verificato soltanto su modulo con LCD da 5 V. Consigliamo comunque a tutti coloro che abbiano a che fare con schede di sviluppo e piattaforme integrate di effettuare quest'operazione di verifica di tutti i valori in ogni caso: conoscere la vostra scheda, anche soltanto per un piccolo difetto, vi consentirà di evitare problemi di qualunque natura.

I pin 15 e 16 dell'LCD dovrebbero avere dei resistori da 0 Ohm. Riuscite a verificarlo? Il problema si manifesta rispetto al possibile valore

intermedio tra 0 e 5 V che sia presente sul pin. In questo caso la lettura è dubbia ed ovviamente il valore non preciso. **Come si può risolvere questo problema?** Molto semplice: installando un resistore di valore compreso tra 1 e 30 kOhm, a seconda di quanto viene fuori dalla lettura, che funga da pull-down. La soluzione, però, può anche non essere necessariamente hardware ma software: basta disabilitare la funzionalità analogica e pilotare attivamente ciascuno dei pin non 5V-tolerant con valori di tensione alti o bassi. Una conseguenza di tutto ciò è la valutazione di come le prestazioni della scheda sono cambiate e di quanto questi interventi abbiano modificato la resa effettiva.

CONCLUSIONI

Tra la dotazione hardware, la possibilità di programmare in diversi linguaggi e sfruttando diversi compilatori, la compatibilità con eventuali codici o progetti già sviluppati in precedenza, i vantaggi nell'utilizzare questa scheda di sviluppo sono veramente tantissimi. Senza contare la grande compatibilità con tante famiglie diverse di microcontrollori. Ed il fatto che si spazi da soluzioni poco performanti fino a micro a 32 bit fa capire chiaramente che questa scheda sarà il miglior banco di prova per chi comincia e contemporaneamente un ottimo strumento per chi invece ha già tanta esperienza.

L'autore è a disposizione nei commenti per eventuali approfondimenti sul tema dell'Articolo. Di seguito il link per accedere direttamente all'articolo sul Blog e partecipare alla discussione: <http://it.emcelettronica.com/un-sistema-modulare-per-lo-sviluppo-su-architetture-a-microcontrollore-da-16-o-32-bit-eccolo-qui>

Ti occorre Potenza Digitale con funzionalità di prossima generazione?

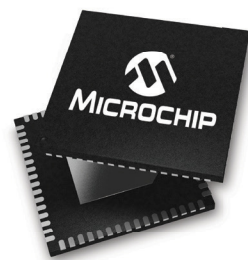
I nuovi DSC dsPIC® fissano nuovi benchmark per dimensioni, latenza e consumo di potenza



Grazie a sofisticati algoritmi di controllo funzionanti a frequenze switching superiori e a Live Update Flash, i Digital Signal Controller 16-bit dsPIC33EP "GS" di Microchip offrono prestazioni di digital-power di prossima generazione.

Questi DSC consumano fino all'80% di potenza in meno in qualsiasi applicazione e offrono una latenza pari a meno della metà di quella delle precedente generazione nell'utilizzo in un compensatore *three-pole three-zero*.

Oltre alle eccezionali performance grazie ad algoritmi di controllo predittivi ed adattivi non-lineari la famiglia DSPIC33EP "GS" offre maggiore integrazione e più funzionalità in package che includono il più piccolo *digital-power-optimised* DSC sul mercato, 4 x 4 mm UQFN.



microchip
DIRECT
www.microchipdirect.com

 **MICROCHIP**

www.microchip.com/get/eudspic33ep



Un basso consumo di potenza è fondamentale per l'Internet delle cose



di Emanuele
24 Settembre 2015

La proliferazione di sensori wireless a supporto dell'Internet delle cose (IoT) ha comportato un aumento della richiesta di convertitori di potenza compatti ed efficienti pensati per dispositivi mobili a consumo di potenza inferiore. Uno degli ultimi segmenti emergenti di questo settore, particolarmente interessante dal punto di vista dell'immagazzinamento di energia ("energy harvesting"), è quello dei dispositivi elettronici indossabili. Naturalmente, la tecnologia dei dispositivi indossabili non risponde solo alle esigenze delle persone; esistono molte applicazioni adatte anche per gli animali. Alcune delle più recenti includono speciali cerotti per ultrasuonoterapia, sistemi di ottimizzazione elettronica della sella di un cavallo, collari per altri animali che rispondono a scopi di monitoraggio, identificazione, diagnosi e così via. Ciò nonostante, indipendentemente dall'applicazione finale, la maggior parte di questi dispositivi richiede una batteria come fonte principale di alimentazione, anche se questa sarà completata da una fonte di energia ambientale – purché ve ne sia una disponibile.

Tuttavia, per quanto riguarda le applicazioni concepite per le persone, sembra che ben presto saranno disponibili tessuti indossabili in grado di generare energia elettrica utilizzando diverse forme di energia "dell'ambiente", libera, e che potrebbero richiedere solo una piccola batteria principale come fonte di riserva. Queste sorgenti di energia libera includono l'energia termica generata dal corpo umano, energia fotovoltaica come quella dell'illuminazione d'inter-

ni o anche la normalissima luce diurna nonché l'energia cinetica generata dai normali movimenti del corpo. Un termine adatto per indicarle potrebbe essere "abiti da alimentazione"! Una delle imprese all'avanguardia in questo campo di ricerca è Dephotex, un progetto fondato dall'Unione Europea che ha sviluppato metodi per fabbricare un materiale fotovoltaico leggero e flessibile quanto basta per essere indossato. Naturalmente, il materiale converte i fotoni in



energia elettrica, che a sua volta è utilizzabile per alimentare vari dispositivi elettronici indossati dall'utente o per caricarne le batterie principali o anche per una combinazione delle due funzioni.

Analogamente, al limite inferiore dello spettro di potenza esistono requisiti sulla conversione di nanopotenza per sistemi di energy harvesting (EH) come quelli frequentemente presenti in apparecchiature IoT (si pensi a Google Glass) e che devono impiegare circuiti integrati di conversione della potenza in grado di gestire livelli bassissimi di potenza e corrente, come decine di microwatt e nanoampere.

Le tecnologie energy harvesting all'avanguardia e immediatamente disponibili, ad esempio in celle fotovoltaiche indossabili o in locali chiusi e in sistemi di immagazzinamento dell'energia generata da vibrazioni, producono livelli di potenza dell'ordine dei milliwatt in condizioni di funzionamento tipiche. Sebbene questi livelli di potenza possano apparire limitativi, il funzionamento di elementi di energy harvesting nel corso di vari anni può far sì che le tecnologie siano in gran parte compatibili con batterie principali di lunga durata, sia per quanto riguarda la raccolta di energia che in termini del costo unitario di energia fornita. Inoltre, i sistemi che incorporano elementi EH normalmente saranno in grado di ricaricarsi una volta consumata l'intera energia iniziale, operazione impossibile per i sistemi alimentati solo da una batteria principale. Ciò nonostante, nella maggior parte delle implementazioni si impiega una sorgente di energia ambiente come generatore di ener-

gia principale, ma integrandola con una batteria che può essere inserita se tale sorgente cessa di essere disponibile.

SOLUZIONI

Naturalmente, l'energia fornita dalla sorgente di energy harvesting dipende dal periodo di tempo in cui quest'ultima è disponibile. Quindi, il parametro principale per confrontare le fonti di energia immagazzinata è la densità di potenza, non la densità di energia. Poiché l'immagazzinamento di energia dipende da livelli di potenza bassi, variabili e imprevedibili, si utilizza spesso una struttura ibrida che s'interfaccia con l'accumulatore di energia e una riserva secondaria di potenza. Quest'ultima potrebbe essere una batteria ricaricabile o un condensatore d'immagazzinamento dell'energia (forse anche supercondensatori). A causa della sua alimentazione illimitata e della carenza di potenza, l'accumulatore è la fonte energetica del sistema. La riserva di potenza secondaria – una batteria o un condensatore – fornisce una potenza di uscita maggiore ma accumula meno energia, fornendo potenza quando necessario, ma venendo regolarmente ricaricata dall'accumulatore. Quindi, in mancanza di energia ambientale occorre usare la riserva di potenza secondaria per alimentare i sistemi elettronici a valle.

Linear ha introdotto vari circuiti integrati per la conversione di potenza dotati delle necessarie funzionalità e caratteristiche prestazionali per consentire l'utilizzo di livelli così bassi di energia raccolta per applicazioni IoT.

L'**LTC3331** è una soluzione EH completa do-



tata di circuito di regolazione, che genera fino a 50mA di corrente continua per prolungare la durata della batteria quando è disponibile energia da immagazzinare. Non assorbe corrente di alimentazione dalla batteria quando trasferisce potenza regolata al carico utilizzando l'energia immagazzinata e solo 950nA quando è alimentato dalla batteria a vuoto. Integra un alimentatore EH ad alta tensione oltre a un convertitore CC/CC buck-boost sincrono alimentato da una batteria principale ricaricabile per creare una singola uscita di continuità per applicazioni di energy harvesting come quelle di dispositivi IoT o indossabili e dei nodi sensori wireless (WSN, wireless sensor node).

buck sincrono ad alta efficienza, immagazzina l'energia generata da sorgenti piezoelettriche (CA), solari (CC) o magnetiche (CA). Uno shunt da 10mA consente di caricare la batteria con l'energia immagazzinata, mentre una funzione di scollegamento della batteria quando questa è quasi scarica la protegge dagli effetti della scarica profonda. La batteria ricaricabile alimenta un convertitore buck-boost sincrono che funziona con tensione d'ingresso compresa tra 1,8V e 5,5V e s'inserisce quando l'energia immagazzinata non è disponibile, per regolare l'uscita indipendentemente dal livello a cui si trova l'ingresso: superiore, inferiore o uguale a quello di uscita. Il caricabatteria dell'LTC3331 è dota-

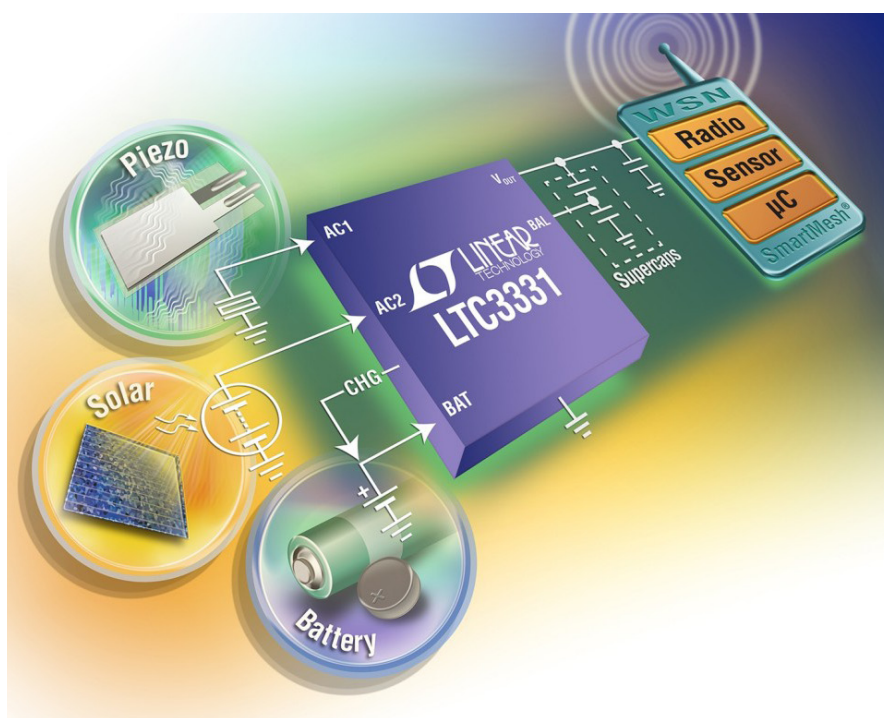


Figura 1. L'LTC3331 converte l'energia di più sorgenti e può utilizzare una batteria principale ricaricabile

L'alimentatore EH dell'LTC3331, consistente di un raddrizzatore a ponte a onda intera che accetta ingressi CA o CC e di un convertitore

to di una funzione di gestione della potenza molto importante, che non può essere trascurata quando si impiegano sorgenti di micropotenza. L'LTC3331 incorpora una funzione di controllo logico del circuito di carica della batteria tale che questa viene caricata solo quando l'alimentatore che immagazzina l'energia ne ha una quantità in eccesso. Senza questa funzione logica, all'avvio la sorgente di energia immagazzinata rimarrebbe 'bloccata' in qualche punto di funzionamento non ottimale e non sarebbe

in grado di alimentare il dispositivo previsto. L'LTC3331 inserisce automaticamente la batteria quando la sorgente per energy harvesting non è più disponibile. Si ottiene così l'ulteriore vantaggio di consentire ai WSN alimentati dal-



la batteria di prolungarne la durata operativa da 10 a oltre 20 anni se per almeno metà del tempo è disponibile una sorgente di alimentazione EH adatta; la durata sarà ancora maggiore se la sorgente EH è disponibile per un intervallo più lungo. È stato integrato anche un circuito di bilanciamento a supercondensatore, che consente di immagazzinare una quantità maggiore di energia all'uscita.

Poiché l'energia immagazzinabile dai dispositivi indossabili è molto bassa – le correnti vanno dai nanoampere ai milliampere – è importantissimo che ogni circuito di conversione CC/CC utilizzi quanto meno potenza possibile per assicurare il trasferimento ottimale dell'energia; quindi, per conseguire questo difficile obiettivo, lo stesso convertitore CC/CC deve consumare corrente nell'ordine dei nanoampere. È per rispondere a questa esigenza che Linear Technology ha introdotto l'LTC3335 – un convertitore CC/CC buck-boost a nanopotenza che incorpora un contatore di Coulomb pensato per prodotti IoT basati su WSN, dispositivi indossabili e applicazioni energy harvesting universali (vedere Figura 2).

L'LTC3335 è un convertitore a bassa corrente di riposo (680nA) e alta efficienza; integra un contatore di Coulomb che in applicazioni alimentate a batteria di lunga durata monitora la carica persa dalla batteria e ne accumula il valore in un registro interno accessibile tramite un'interfaccia I2C. Il convertitore buck-boost può funzionare a tensioni molto ridotte all'ingresso – fino a 1,8V – e permette di scegliere fra otto tensioni di uscita, selezionabili con pin, con corrente di uscita sino a 50mA. Al fine di impiegare il convertitore con un'ampia gamma di tipi e capacità della batteria, è possibile selezionare la corrente d'ingresso di picco da un minimo di 5mA a un massimo di 250mA, mentre il contatore di Coulomb a fondo scala ha un intervallo programmabile di 32.768:1.

Il contatore di Coulomb di precisione integrato monitora la carica accumulata che viene trasferita da una batteria ogni volta che il convertitore buck-boost eroga corrente al carico. Il conver-

tore funziona come un ponte a H per tutte le condizioni della batteria e della tensione di uscita quando non si trova nella modalità di sospensione (vedere Figura 3).

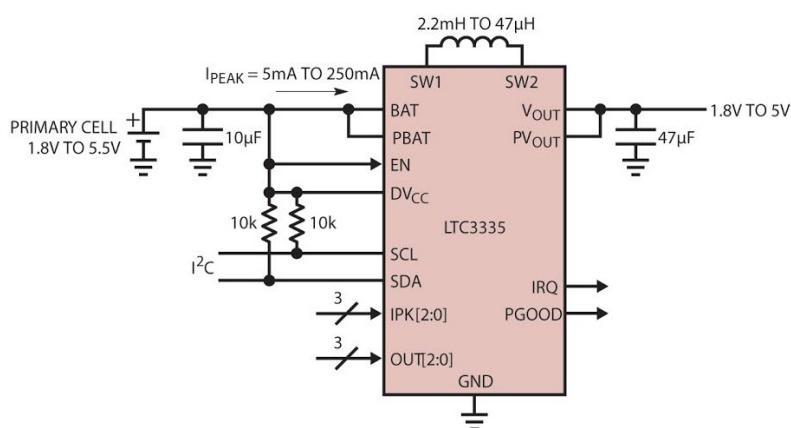


Figura 2. Tipico schema applicativo del convertitore buck-boost a nanopotenza LTC3335

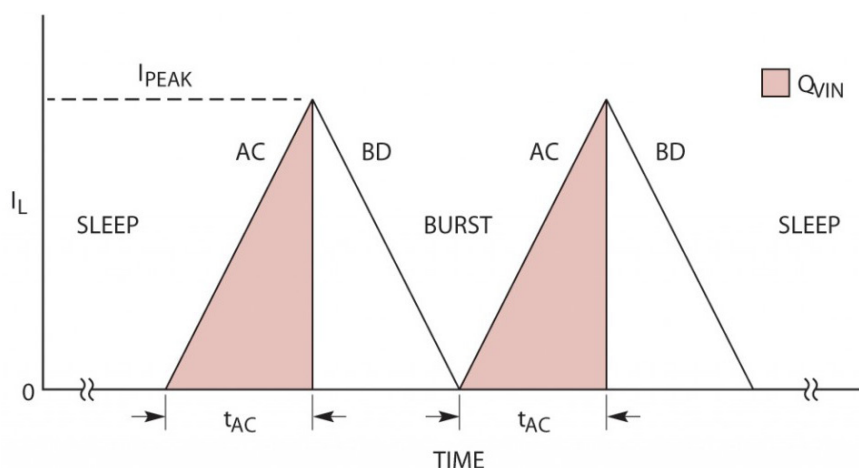


Figura 3. Diagramma di temporizzazione dell'LTC3335 funzionante come un ponte a H

Gli interruttori A e C si chiudono all'inizio di ciascun ciclo di burst. La corrente dell'induttore aumenta linearmente sino a I_{peak} ; a questo punto gli interruttori A e C si aprono, mentre gli interruttori B e D si chiudono finché la corrente dell'induttore non diminuisce linearmente sino a zero. Il ciclo si ripete finché V_{out} non raggiunge la soglia di sospensione. Se I_{peak} e il tempo (t_{AC}) in cui gli interruttori sono chiusi – AC(ON) – sono entrambi noti, è possibile calcolare la carica persa BAT (l'area ombreggiata nella Figura 3) contando il numero di cicli AC(ON) e moltiplicandoli per la carica corrispondente a AC(ON) secondo la seguente formula:

$$q_{AC(ON)} = (I_{peak} * t_{AC})/2$$

Quando il convertitore buck-boost è in funzione, l'LTC3335 misura il tempo AC(ON) effettivo re-

lativamente al tempo ON di fondo scala (t_{FS} , pari a circa $11,74\mu s$) che viene regolato internamente per compensare eventuali errori nell'effettivo valore I_{peak} selezionato dovuti alle variazioni dei parametri dell'alimentatore, della temperatura e del processo. Si ottiene così una "misura" molto precisa della carica trasferita dalla batteria durante ciascun ciclo AC(ON).

CONCLUSIONE

È evidente che saranno sviluppati molto WSN, dispositivi indossabili e prodotti IoT che avranno bisogno di conversione CC/CC a nanopotenza e conteggio della carica per assicurare durata e prestazioni ottimali. Tuttavia, solo recentemente tali prodotti hanno iniziato a essere disponibili sul mercato. Grazie a produttori come Linear Technology, i progettisti di dispositivi a nanopotenza avranno un'ampia scelta.

L'autore è a disposizione nei commenti per eventuali approfondimenti sul tema dell'Articolo. Di seguito il link per accedere direttamente all'articolo sul Blog e partecipare alla discussione: <http://it.emcelettronica.com/un-basso-consumo-di-potenza-e-fondamentale-per-linternet-delle-cose>

Rimini Beach Mini Maker Faire®

14 e 15 NOVEMBRE RIMINI FIERA

LA FIERA DEL FARE: un evento aperto a tutti,
dedicato all'innovazione e alla creatività, analogica e digitale.
Trasforma le tue visioni in realtà, il futuro è adesso!



MAK-ER
FABLAB



UAV
WORLD



VIDEOGAME



CRAFT



COS-MAKERS
& STEAMPUNK



FOTOMERCATO



3D PRINTER



MAKING
MUSIC



EXPO
ELETTRONICA



MINECRAFT
CHALLENGE



THE CALL IS OPEN!

Hai tempo fino al **30 settembre 2015**
per **iscriverti alla CALL4Makers** e
diventare protagonista con il tuo progetto!

makerfairerimini.it



ORGANIZZATO DA



PATROCINI



IN COLLABORAZIONE CON



MEDIA PARTNER



TECHNICAL PARTNER





OpenBCI: la piattaforma open source che utilizza le onde cerebrali



di [slovati](#)

24 Settembre 2015

OpenBCI, acronimo di Brain-Computer Interface, è un progetto open source che mira a utilizzare le onde cerebrali per interagire, tramite l'ausilio di un'opportuna piattaforma hardware e software, con il mondo circostante. Una prima applicazione, scontata, è sicuramente quella relativa al mondo del gaming, ma ne stanno nascendo altre, in grado di delineare un futuro non molto lontano da quanto descritto nei romanzi di fantascienza. Scopriamo insieme in questo articolo in cosa consiste questo affascinante progetto

Introduzione

OpenBCI è un progetto open source sviluppato da due ingegneri e artisti (come essi stessi amano definirsi) di New York: Conor Russomanno e Joel Murphy. Finanziato con successo attraverso una campagna di crowdfunding su Kickstarter, il progetto permette di misurare e registrare i segnali elettrici prodotti dal cervello (**EEG**, o elettroencefalografia), dai muscoli (**EMG**, o elettromiografia), e dal cuore (**EKG**, o elettrocardiogramma), ed è completamente compatibile con gli elettrodi standard utilizzati per l'elettroencefalografia.

OpenBCI può essere utilizzato con il proprio software di elaborazione e visualizzazione, ma può anche essere facilmente integrato con altre

applicazioni software open source nate per impieghi in ambito EEG.

Dal punto di vista **hardware**, le board che compongono OpenBCI utilizzano il circuito integrato **ADS1299** sviluppato da Texas Instruments. Si tratta di un convertitore analogico-digitale (**ADC**) sigma-delta a basso rumore, con risoluzione pari a 24 bit, espressamente progettato per i sistemi biomedicali, e più precisamente per applicazioni EEG quali l'elettroencefalogramma. Questo componente, che si occupa della conversione ed elaborazione del segnale di basso livello, si interfaccia con un **microcontrollore** il cui compito è quello di gestire le varie funzionalità svolte dalla board e l'interfacciamento con il sistema host. Esistono **due versioni** di schede: la prima si basa su un mi-



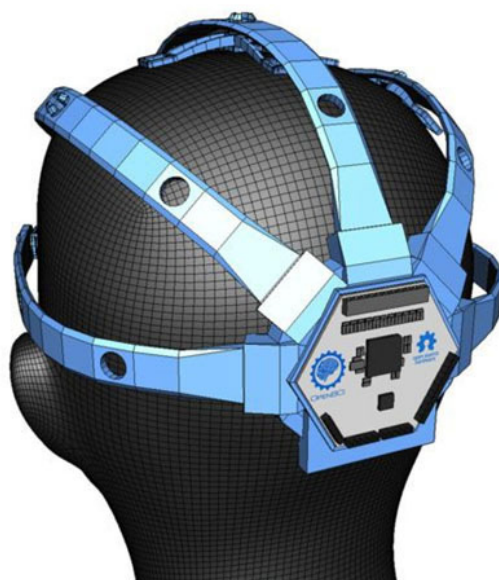
crocontrollore a **8 bit** Atmega328P, ed è equipaggiata con il bootloader Arduino, mentre la seconda versione impiega un microcontrollore a **32 bit** della famiglia PIC Microchip, ed include il bootloader Chipkit. Le board sono gemelle, e dispongono dello stesso numero di pin analogici e digitali delle schede della famiglia Arduino. Le misure elettriche vengono acquisite tramite un apposito caschetto corredato di sensori, e possono poi essere memorizzate su una normale **scheda SD**, oppure inviate a un dispositivo esterno tramite connessione **Bluetooth**.

Per quanta riguarda la componente **software**, OpenBCI può contare su un'applicazione open source sviluppata con il linguaggio **Processing**, oltre ad un opportuno software di elaborazione e visualizzazione sviluppato in **Python** e in **Node.js**.

La piattaforma è completata da un **apposito caschetto** completo di elettrodi (del quale sono disponibili anche i file per la stampa 3D) che, una volta indossato, permette al sistema hardware di acquisire e processare i segnali elettrici provenienti dall'encefalo.

Nell'immagine seguente possiamo osservare il "concept" originale che ha dato vita al progetto OpenBCI, lo stesso utilizzato per far partire la campagna su Kickstarter. si può notare facilmente la presenza, sui rami del caschetto, dei fori predisposti per l'alloggiamento degli elet-

trodi, e la board OpenBCI collocata nella parte posteriore dell'indossabile, in prossimità della nuca.



In quest'altra inquadratura possiamo invece vedere una versione prototipale (ma già riproducibile con una stampante 3D) del caschetto EEG OpenBCI indossata dallo stesso Conor Russomanno.



Nel 2014, durante una esibizione tenutasi al **Barbican Museum di Londra**, è stata mostrata al pubblico un'applicazione di OpenBCI nella quale la piattaforma veniva utilizzata per con-



trollare, tramite le onde cerebrali, i movimenti di un robot all'interno di un **videogioco**. E' facile immaginare come questa dimostrazione fu subito accolta con interesse ed entusiasmo da parte dei visitatori. Tutto qui? Assolutamente no. Tra le possibili applicazioni di questa piattaforma va sicuramente citata la **realtà aumentata**, dove la disponibilità di una nuova e sino ad ora inesplorata interfaccia uomo-macchina può aprire scenari un tempo impensabili. Non solo, esistono anche altre applicazioni ben più utili e, se vogliamo, nobili: si sta infatti pensando di utilizzarla per fornire un efficace mezzo di comunicazione ai pazienti affetti da **SLA** e da altre **patologie neurodegenerative**, che impediscono loro il movimento e/o la possibilità di parlare.

L'EEG

Abbiamo accennato in precedenza come la piattaforma OpenBCI poggi le sue basi sui principi dell'elettroencefalografia, o **EEG**. Come noto, questa tecnica viene utilizzata ogni giorno in tutto il mondo per eseguire esami e test relativi al funzionamento dell'encefalo. In pratica viene rilevata l'attività elettrica del cervello mediante l'utilizzo di alcune placche di metallo (gli elettrodi) collocati in opportuni punti sullo scalpo. Le cellule cerebrali comunicano attraverso degli impulsi elettrici e sono sempre attive, anche durante il sonno. Questa attività viene registrata sotto forma di tracciati (linee curve) che danno così forma all'elettroencefalogramma. L'EEG presenta il vantaggio di essere una tecnica **minimamente invasiva**, necessitando solo di alcuni elettrodi posti in posizioni opportune e

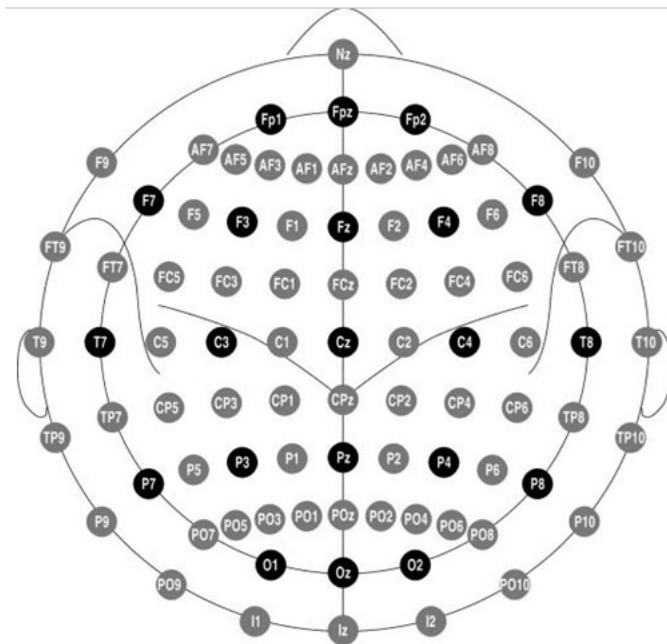
dell'applicazione di un'apposita pasta conduttiva (gel). Inoltre, i segnali EEG possono essere facilmente convertiti in una forma interpretabile da un computer: dopo tutto, i computer non sono altro che dispositivi il cui funzionamento si basa sull'andamento dei segnali elettrici, e la tecnica EEG non è altro che l'acquisizione dei segnali elettrici prodotti dall'encefalo. L'EEG è perciò la tecnica più semplice e ovvia per mettere in comunicazione l'encefalo con un computer.

L'immagine seguente mostra uno dei kit disponibili sul sito OpneBCI, comprendente un set completo di elettrodi e la pasta conduttiva da applicare agli stessi:

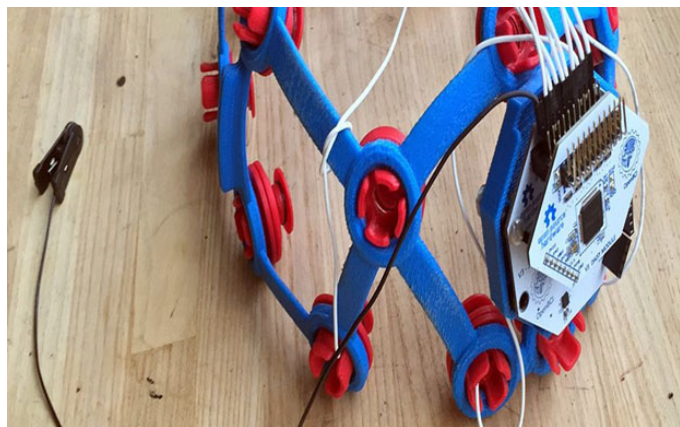


L'elettroencefalografia esiste da oltre cento anni, e viene utilizzata con successo nella diagnosi di malattie o disturbi legati al funzionamento dell'encefalo, tra le quali ricordiamo l'epilessia. A livello commerciale sono già stati compiuti dei tentativi di utilizzo di questa tecnica in ambiti al di fuori di quello strettamente medicale. OpenBCI, tuttavia, mira ad andare oltre. L'**obiettivo dichiarato** è infatti quello di creare uno strumento potente e personalizzabile che

dard). Inoltre, tutte le versioni stabili dei file che compongono il progetto (schemi hardware, software, e i file per la stampa 3D) sono disponibili per il download (nonchè un'eventuale personalizzazione) su **GitHub**.



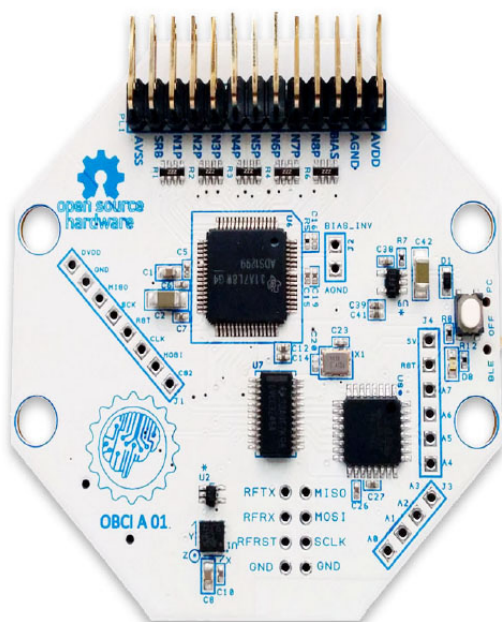
Tutto l'hardware che compone la piattaforma OpenBCI è stato prodotto con il tool **Design Spark**, un programma free per la creazione di PCB, modelli 3D, e non solo. Come abbiamo già accennato in precedenza, l'hardware si compone fondamentalmente di **due schede** a microcontrollore, una a 8 bit e la seconda a 32 bit, progettate seguendo una filosofia analoga a quella che ha guidato lo sviluppo di Arduino. In basso possiamo osservare il layout della parte superiore di una di queste schede (si noti la presenza dell'ADC di TI al centro della board, verso il bordo superiore)). In aggiunta a queste, vi sono poi altre schede o componenti hardware



22 - Firmware



aggiuntivi, e ovviamente il caschetto EEG. Vediamo di analizzare meglio questi componenti.



Scheda OpenBCI a 8 bit

Le principali caratteristiche tecniche di questa board sono le seguenti:

- alimentazione (esclusivamente) con batteria 6-12Vcc;
- microcontrollore ATmega328P con bootloader Arduino Uno;
- front end analogico basato sull'ADS1299;
- accelerometro a 3 assi di tipo LIS3DH;
- interfaccia Bluetooth basata sul modulo RFduino BLE;
- slot per schede Micro SD;
- regolatori di tensione a 5V, 3V, +2.5V, e -2.5V;
- traslatore di livelli SN74LVCC3245A;
- dimensioni della scheda pari a 2,41" x 2,41" (i lati dell'ottagono hanno lunghezza pari a 1");
- fori di fissaggio 1/16" ID, 0,8" x 2,166" posti al centro della board.

Sono inoltre presenti i **seguenti pin**:

pin ICSP per scaricare il bootloader sull'ATmega:

MISO, MOSI, SCLK, 5V, MCLR, GND

pin seriali per programmare il modulo RFduino: RFTX, RFRX, RFRST, GND

pin interfaccia SPI per l'espansione Daisy Module:

DVDD, GND, MISO, MOSI, SCK, CS, CLK, RST
pin ATmega non utilizzati:

A0, A1, A2, A3, A4, A5, A6, A7

I pin SPI includono il segnale CLK, collegato al pin CLK dell'ADS1299 per fornire temporizzazione al Daisy Module (una scheda di espansione per OpenBCI), e il segnale RST collegato al pin MCLR dell'ADS1299 per resettare l'ADC. I pin A0-A5 possono essere utilizzati come ingressi analogici oppure come uscite digitali, esattamente come su un normale Arduino UNO. I pin A6 e A7 sono utilizzabili soltanto come ingressi analogici.

Scheda OpenBCI a 32 bit

Le principali caratteristiche tecniche di questa board sono le seguenti:

- alimentazione (esclusivamente) con batteria 3-6 Vcc;
- microcontrollore PIC32MX250F128B con bootloader chipKIT UDB32-MX2-DIP;
- front end analogico basato sull'ADS1299;
- accelerometro a 3 assi di tipo LIS3DH;
- interfaccia Bluetooth basata sul modulo RFduino BLE;



ino BLE;
slot per schede Micro SD;
regolatori di tensione a 3,3V, +2,5V, e -2,5V;
dimensioni della scheda pari a 2,41" x 2,41" (i lati dell'ottagono hanno lunghezza pari a 1");
fori di fissaggio 1/16" ID, 0,8" x 2,166" posti al centro della board. Sono inoltre presenti i **seguenti pin**:

pin di programmazione per scaricare il bootloader sul PIC: PGC, PGD, VDD, MCLR, GND
pin seriali per programmare il modulo RFDuino: RFTX, RFRX, RFRST, GND

pin interfaccia SPI per l'espansione Daisy Module: DVDD, GND, MISO, MOSI, SCK, CS, CLK, RST

pin PIC non utilizzati:

D11, D12, D13, D17, D18

I pin SPI includono il segnale CLK, collegato al pin CLK dell'ADS1299 per fornire temporizzazione al Daisy Module, e il segnale RST collegato al pin MCLR dell'ADS1299 per resettare l'ADC. I pin D11, D12, e D13 possono essere utilizzati sia come pin analogici che digitali. Il pin D11 è anche PGD, e ha collegato il led blu in serie con una resistenza da 1K collegata a AGND. D12 corrisponde al segnale PGC, utilizzato per il bootloading. D17 e D18 sono pin di I/O digitale, con D17 collegato al pulsante PROG.

Nell'immagine qui sopra possiamo osservare la versione a 32 bit della board OpenBCI. Nella parte inferiore (a destra) si notano il modulo Bluetooth e il connettore per le schede Micro SD.

Il Dongle

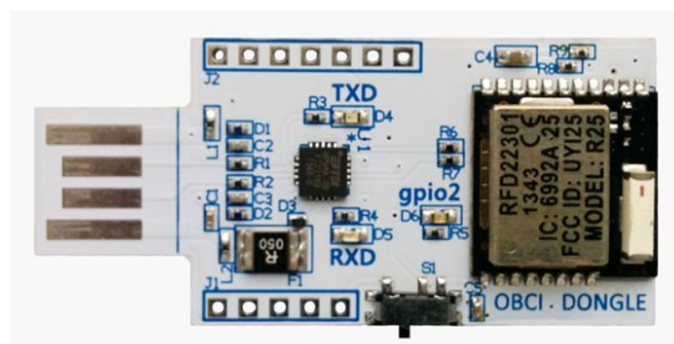
Il dongle (visibile nell'immagine qui sotto) consiste in una piccola chiavetta USB che, collegata a un normale PC equipaggiato con un'opportuna applicazione software, consente di comunicare via Bluetooth con la board OpenBCI e quindi con il resto del sistema. La sua realizzazione è relativamente semplice, ed include:

alimentazione tramite connettore USB;

modulo RFDuino BLE per la comunicazione via radio;

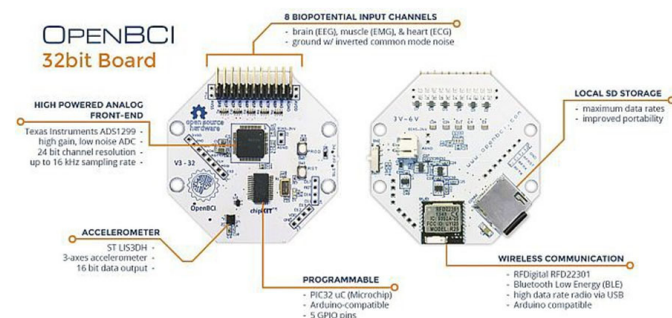
convertitore USB.seriale FTDI;

fusibile resettabile.



Il Daisy Module

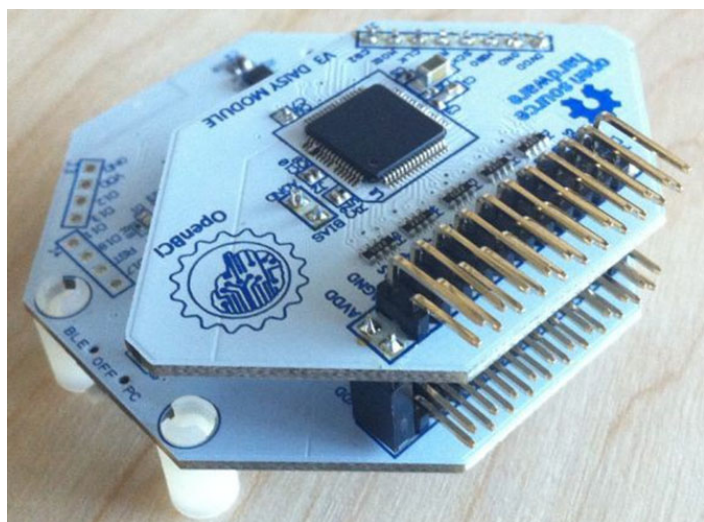
E' un modulo di espansione che si collega direttamente sulla board OpenBCI, e ha la funzione di fornire ulteriori 8 segnali di ingresso (canali), portando il numero totale di ingressi (collegabili ai sensori posti sul caschetto) a 16. Questo modulo di espansione è utilizzabile soltanto con la scheda OpenBCI a 32 bit, e deriva l'alimen-





tazione dalla scheda stessa. Il Daisy Module è anch'esso equipaggiato con un ADC ADS1299, che si interfaccia e interagisce con il gemello posto sulla main board (una caratteristica di questo componente è quella di rendere possibile e agevole la connessione di IC multipli dello stesso tipo).

In questa immagine possiamo vedere il modulo di espansione Daisy Modulo installato sulla board OpenBCI:



Il software

A livello software, il progetto OpenBCI è accompagnato da un Software Development Kit (SDK) che include il software installato su tutte le board del sistema. La scheda OpenBCI (sia nella versione a 8 che a 32 bit) viene programmata con un software che permette la comunicazione con il front end analogico (ADS1299), l'accelerometro (LIS3DH), la scheda micro SD (se installata), e il modulo RFduino. Quest'ultimo utilizza lo stack e la libreria Nordic Gazelle, e viene configurato come **DEVICE**. La comunicazione tra la scheda OpenBCI e l'applicazione

sul sistema host (tipicamente, ma non necessariamente, un PC) avviene tramite un link radio realizzato dal modulo Bluetooth, ed utilizza un protocollo di comunicazione proprietario basato su una serie di comandi ASCII.

L'utente (oppure l'applicazione), controlla la scheda inviando ad essa dei comandi composti da una sequenza di caratteri ASCII, trasmessi attraverso il Dongle USB. Quest'ultimo a sua volta contiene un modulo RFduino configurato come **HOST**, e si interfaccia con il computer tramite una Com Port Virtuale (realizzata dal componente FTDI).

Il set completo di comandi, consultabile sul sito di [OpenBCI](https://openbci.com), include comandi per:

abilitazione/disabilitazione del canale, test del canale, impostazione e configurazione del canale (guadagno, bias, ecc.), accesso alla scheda micro SD, controllo della trasmissione dati, ecc. A livello applicativo, quello che resta da fare è quindi realizzare su PC un programma che gestisca opportunamente l'invio di comandi alla scheda OpenBCI.

Particolare enfasi va posta sull'**interpretazione dei dati EEG**. Si può dire che questo rappresenta il punto chiave dell'applicazione software: occorre infatti decidere come interpretare i dati acquisiti assegnando ad essi un particolare significato.

I fattori da tenere in considerazione a questo proposito sono due:

- il rate di acquisizione dei dati ("sample rate");
- il fattore di scala ("scale factor").

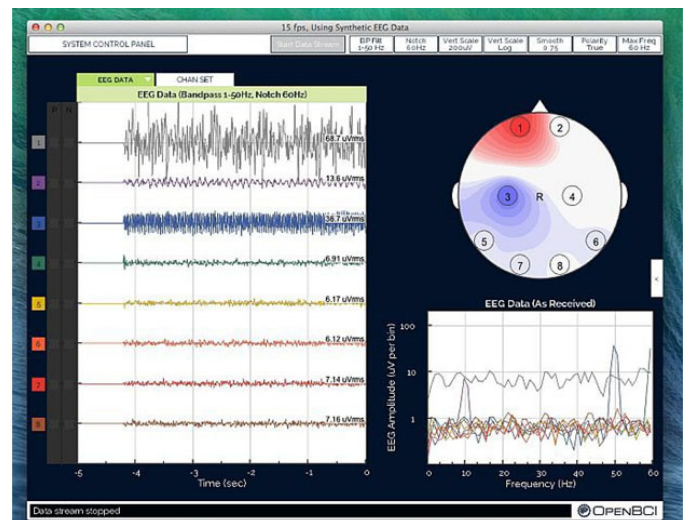


Il primo è stato impostato di default al valore pari a 250 Hz. L'ADS1299 è in grado di supportare rate anche molto superiori, ma il collo di bottiglia è rappresentato dal modulo Bluetooth che, acquisendo a velocità troppo elevate, potrebbe non essere in grado di stare al passo e perdere dei campioni durante la trasmissione verso il sistema host.

Per quanto riguarda il fattore di scala, esso rappresenta il fattore moltiplicativo da utilizzare per convertire le misure EEG da numero puro (fornito dall'ADC) in valore di tensione. Il software che gira sulla scheda OpenBCI imposta il massimo guadagno sull'ADS1299 (24x), il quale corrisponde a un fattore di scala di 0,02235 microVolts per unità di campionamento. Poiché il guadagno è configurabile dall'utente (valori ammessi: 24x, 12x, 8x, 6x, 4x, 2x, 1x), il fattore di scala può anch'esso variare di conseguenza. Abbiamo visto in precedenza come il **Daisy Module** permette di estendere il numero di canali di acquisizione **da 8 a 16**. Cosa succede in questo caso al sample rate? I progettisti di OpenBCI hanno deciso di mantenere come limite il rate di 250 campioni al secondo. Il workaround, se così si può dire, è quello di trasmettere **alternativamente** i campioni acquisiti dai due ADS1299 (una volta si trasmette un campione della main board, e la successiva un campione del modulo di espansione, e così via).

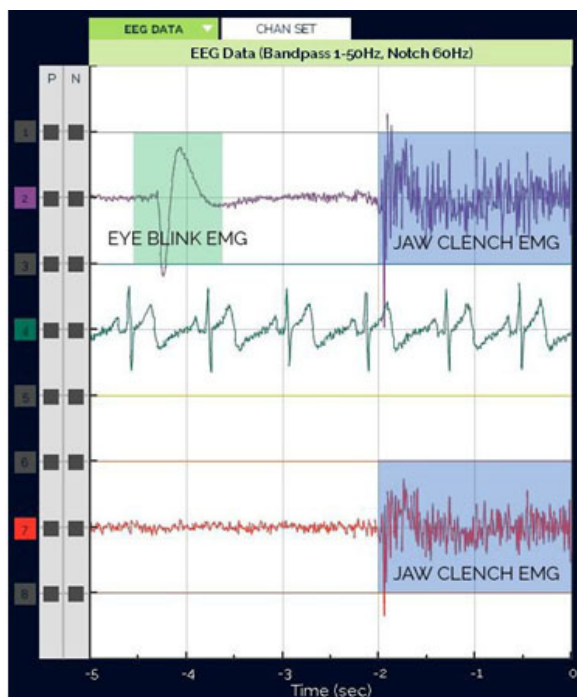
Lato sistema host (PC), OpenBCI fornisce un applicativo sviluppato nel linguaggio Processing, che andrà installato se non presente sul si-

stema. Questa applicazione, in pratica una GUI, è molto completa, e permette di configurare, calibrare, e controllare l'hardware OpenBCI. Sono disponibili numerose opzioni di visualizzazione grafica (un esempio è fornito dall'immagine seguente), attraverso le quali di possono analizzare i dati acquisiti dal sistema. E' ovviamente possibile testare il sistema su se stessi! Spiegazioni esaurienti e tutorial dettagliati sono disponibili al proposito sul sito OpenBCI.



In questa immagine possiamo osservare la rappresentazione grafica dei segnali EEG e il loro andamento nel tempo (in questo caso abbiamo un'acquisizione simultanea su 8 canali/elettrodi), collocati sul capo come visibile nel riquadro in alto a destra.

Quest'altra immagine si riferisce invece a un'applicazione particolare: è stato infatti chiesto al soggetto sotto esame di aprire e chiudere le palpebre. L'effetto è stata la generazione di un picco (spike) in corrispondenza del canale 2, cioè l'elettrodo posto proprio sopra l'occhio destro (vedi immagine precedente):



Ed infine, per chiudere in bellezza, ecco un'immagine relativa alle onde cerebrali. In questo caso è visibile l'andamento delle cosiddette onde cerebrali "alfa", cioè le onde cerebrali (con frequenza compresa tra 7,5 e 12,5 Hz) che hanno l'intensità EEG maggiore e vengono tipicamente emesse quando si chiudono gli occhi.

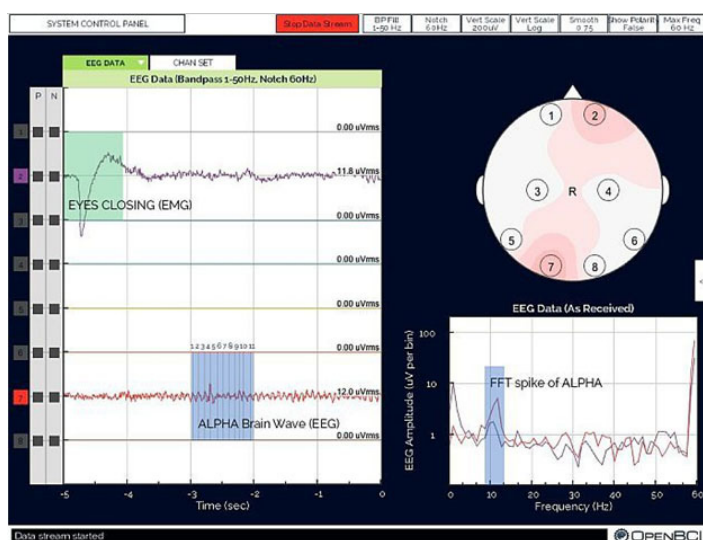
Dall'analisi di questa immagine possiamo evidenziare i seguenti aspetti:

all'atto della chiusura degli occhi, è stato generato uno spike EMG in corrispondenza del canale 2 (elettrodo posto sopra l'occhio destro). Lo spike è ovviamente dovuto al movimento dei muscoli che muovono la palpebra; successivamente, con gli occhi già chiusi, è stata rilevata un'attività cerebrale di tipo alfa con un picco evidenziato nel diagramma FFT (in basso a destra). Questo picco (intorno ai 10-12

Hz) corrisponde a una frequenza propria delle onde alfa; la maggiore intensità EEG delle onde alfa è stata rilevata dal canale 7, associato all'elettrodo occipitale O2, posto nella parte posteriore del capo. Quando si chiudono gli occhi, il lobo occipitale dell'encefalo (responsabile della vista) si porta infatti in uno stato in cui vengono generate le onde cerebrali alfa.

Conclusioni

Abbiamo visto in questo articolo in cosa consiste la piattaforma OpenBCI, un'affascinante progetto, interamente open source, rivolto ad applicazioni in un'area di confine tra il biomedicale e la fantascienza. Sul sito OpenBCI è presente una vastissima documentazione di progetto, schemi, tutorial, e un forum rivolto agli sviluppatori. Il progetto è in sé molto promettente e le possibili applicazioni future sono molto interessanti.



L'autore è a disposizione nei commenti per eventuali approfondimenti sul tema dell'Articolo. Di seguito il link per accedere direttamente all'articolo sul Blog e partecipare alla discussione: <http://it.emcelettronica.com/openbci-la-piattaforma-open-source-che-utilizza-le-onde-cerebrali>



FEATURED EVENT

CONNECTED
AUTOMOBILES
2015

WWW.CONNECTEDAUTOMOBILES.EU

2015 WORLD EDITION

Smart*Mobility*World



AVL
AVM

LNG

SAFETY

APP

SMART MOBILITY
SMART PEOPLE
SMART CITY

28-30 OTTOBRE 2015
AUTODROMO DI MONZA

WWW.SMARTMOBILITYWORLD.EU

ORGANIZED BY



PROMOSSO DA



SHARING
MOBILITY

CONNECTED
CAR

E-TICKETING

GREEN
LOGISTICS

ITS

SMART
PARKING

E-MOBILITY

RFID



Convertitore buck-boost facile da collegare in parallelo per ridurre al minimo gli aumenti di temperatura



di Emanuele
26 Settembre 2015

L'LT3790 è un convertitore CC/CC buck-boost sincrono a 4 interruptori che regola sia la tensione costante che la corrente costante con efficienza sino al 98,5% utilizzando un solo induttore. Può applicare al carico centinaia di watt e funziona a ingresso e uscita nominali di 60V, per cui è perfetto come regolatore di tensione CC/CC e caricabatteria quando sono necessarie conversione sia in salita che in discesa. Un singolo convertitore LT3790 può applicare una potenza elevata al carico grazie alla sua topologia di commutazione sincrona, ma alla fine le perdite di commutazione e/o conduzione a potenza maggiore possono causare un riscaldamento eccessivo della scheda. Sebbene sia possibile smaltire il calore con dissipatori maggiorati, driver di gate esterni aggiuntivi e/o flusso d'aria forzata, potrebbe essere preferibile semplicemente collegare in parallelo due o più convertitori per distribuire il carico, un'operazione semplice da eseguire con il regolatore buck-boost LT3790.

REGOLATORE DI TENSIONE BUCK-BOOST CON USCITA A 5A, 24V, 120W

Il convertitore buck-boost illustrato nella Figura 1 regola 24V con un carico da 0A a 5A ed efficienza sino al 98,5% accettando tensioni d'ingresso comprese tra 8V e 56V. È protetto

da un sistema di blocco da sottotensione e sovratensione, regolabile; inoltre è protetto contro il cortocircuito all'uscita, indicato dall'apposito flag /SHORT. Funziona in modalità di conduzione discontinua (DCM) a carichi leggeri ai fini del minimo consumo di potenza e per offrire



protezione contro l'inversione della corrente. Il resistore di rilevazione ROUT determina il limite della corrente di uscita durante situazioni sia di cortocircuito che di sovraccarico, assicurando la robustezza di questa applicazione.

Il massimo aumento di temperatura sui componenti della scheda da 120W con ingresso di 12V è di soli 20°C (il componente che raggiunge la massima temperatura è un MOSFET a commutazione), come illustrato nella Figura 2a. Esiste ancora un margine per una potenza di uscita maggiore con ingresso di 12V o per lo stesso valore di 120W con VIN inferiore senza un aumento eccessivo della temperatura dei componenti; tenere presente che una potenza di uscita maggiore richiede un limite della corrente di uscita corrispondentemente maggiore. Quando il dispositivo funziona a tensione d'ingresso inferiore – sino a 8V – con uscita di 120W, i componenti della scheda di circuiti stampati a 4 strati standard dell'LT3790 rimangono a temperatura minore di 97°C (alla temperatura ambiente) senza bisogno di dissipatori né di flusso d'aria forzata. Per applicare una potenza notevolmente maggiore con lo stesso aumento limitato di temperatura e intervallo delle tensioni d'ingresso, si possono collegare facilmente in parallelo due o più LT3790.

CONVERTITORI IN PARALLELO, MASTER A TENSIONE COSTANTE, SLAVE A CORRENTE COSTANTE

In una situazione ideale, convertitori a commutazione collegati in parallelo condividono il carico in misura uguale nell'intero intervallo di uscita. Poiché l'LT3790 può funzionare sia a tensione costante che a corrente costante, un convertitore master può regolare la tensione di uscita mentre la sua uscita di monitoraggio della corrente (ISMON) indica a uno o più convertitori slave quanta corrente di uscita occorre regolare (ingresso CTRL) in modo che questa sia adattata al livello di uscita del convertitore. L'adattamento della corrente tra più convertitori è quasi perfetta utilizzando questa tecnica.

Il pin CLKOUT del master può essere collegato direttamente al pin d'ingresso SYNC dello slave per ottenere un interleaving di 180° delle fasi dei due convertitori in parallelo. Lo sfasamento di 180° tra i convertitori ne riduce il ripple di uscita complessivo anziché raddoppiarlo. Se più di due convertitori sono collegati in parallelo, è possibile sincronizzarli affinché funzionino in fase o fuori fase con un generatore di clock esterno o collegando in serie i pin CLKOUT.

La Figura 3 mostra un regolatore di tensione a 24V e 10A (o 25A in certe condizioni, vedere la figura) formato facendo funzionare due LT3790 in parallelo; con questa configurazione, il massimo aumento di temperatura sui componenti discreti è di soli 20°C con ingresso di 12V e di 50°C con ingresso di 9V – riscontrato sui MOSFET M3 e M7.

Il convertitore superiore (master) nella Figura 3



regola la tensione di uscita a 24V e comanda il livello di corrente che viene regolato dal convertitore inferiore (slave). L'uscita ISMON del master indica quanta corrente viene applicata dal master stesso e collegandola direttamente al pin d'ingresso CTRL dello slave, quest'ultimo è costretto a seguire il master. Il livello d'ingresso CTRL e il livello di uscita ISMON dell'LT3790 vengono mappati in modo identico per cui è possibile una connessione diretta dall'uno all'altro e ciò fa sì che la corrente di uscita totale sia condivisa in modo uguale fra i convertitori in parallelo, come illustrato nella Figura 4. Si noti che la tensione di uscita dello slave è impostata a un valore leggermente maggiore (28V) per cui l'anello di retroazione di tensione dello slave non è regolato e questo consente allo slave di seguire il master.

ANALISI DELL'ANELLO PER LA STABILITÀ

È possibile misurare la stabilità tramite sia l'analisi dell'anello con un analizzatore di rete sia con la risposta al transitorio. Una risposta al transitorio di una corrente compresa tra il 50% e il 100%, mostrata nella Figura 5, indica un convertitore compensato correttamente e una corrente di carico condivisa in egual misura. Un'ulteriore analisi con l'analizzatore di rete fornisce i dettagli dei convertitori separati. La misura e il punto di introduzione del rumore necessari per generare i diagrammi di Bode dell'anello di regolazione sono differenti per il regolatore ma-

ster a tensione costante e il regolatore slave a corrente costante. Separatamente, è possibile misurare la risposta di ciascun anello applicando appropriatamente il segnale di perturbazione, come illustrato nella Figura 6

È necessario aprire l'anello dello slave a corrente costante e applicare il segnale nel percorso di retroazione dell'anello di corrente anziché in quello tradizionale di tensione poiché è il primo a essere in uso durante il funzionamento in parallelo. Il diagramma di Bode del master nella Figura 7 dimostra la stabilità del sistema.

CONCLUSIONE

Il controller buck-boost sincrono LT3790 applica oltre 100W con efficienza fino al 98,5% a un'ampia gamma di carichi ed è facile collegare in parallelo più convertitori per ottenere all'uscita potenze ancora più alte. La possibilità di regolare la corrente o la tensione di uscita, in combinazione con l'adattamento tra i vari livelli ottenibile con l'amplificatore di uscita ISMON e l'amplificatore d'ingresso CTRL, semplifica la connessione del regolatore di tensione master e di uno o più regolatori di corrente slave. Si ottiene così un regolatore buck-boost da 60V ad alta potenza in grado di applicare centinaia di watt ad alta efficienza.

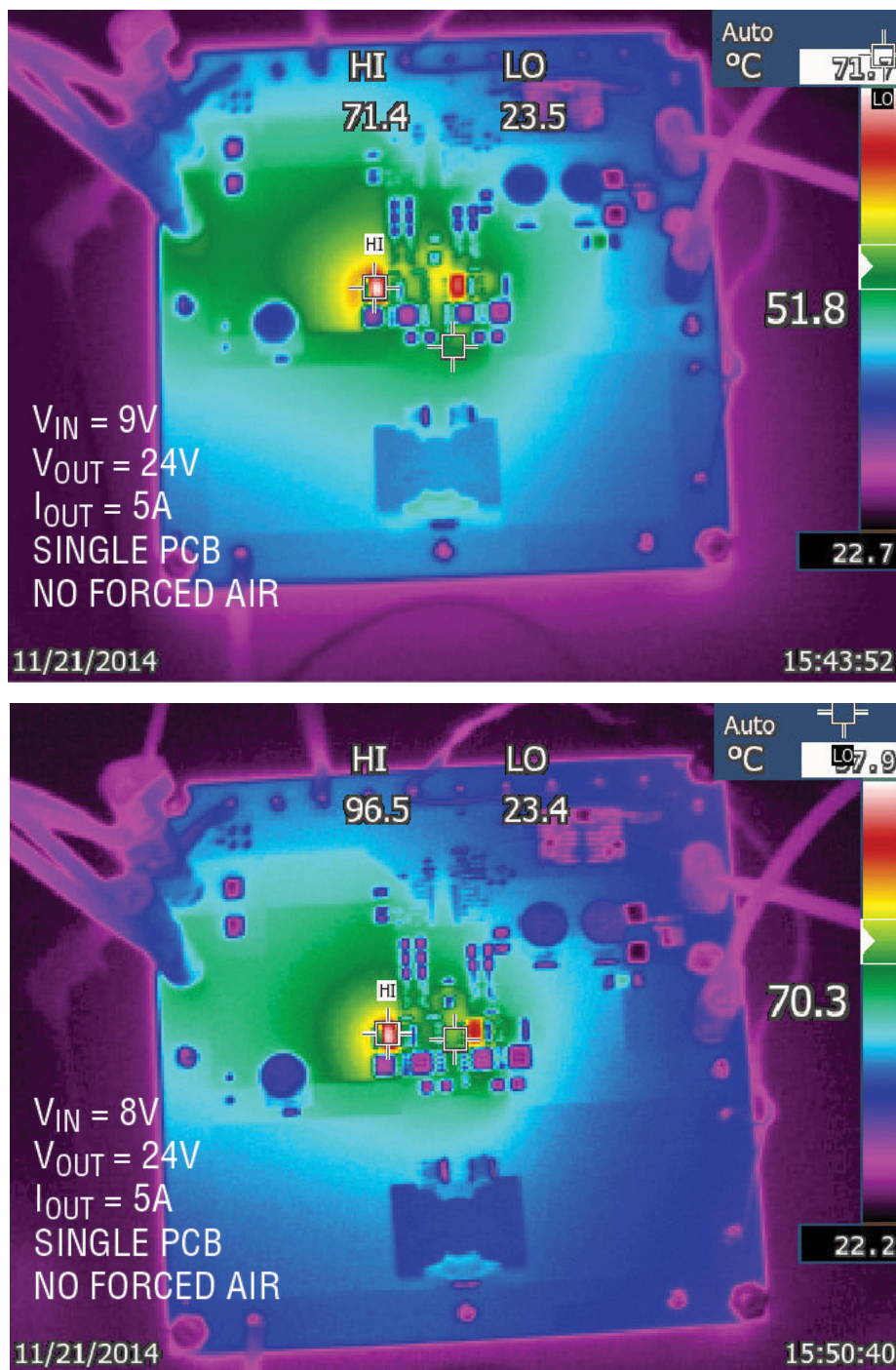
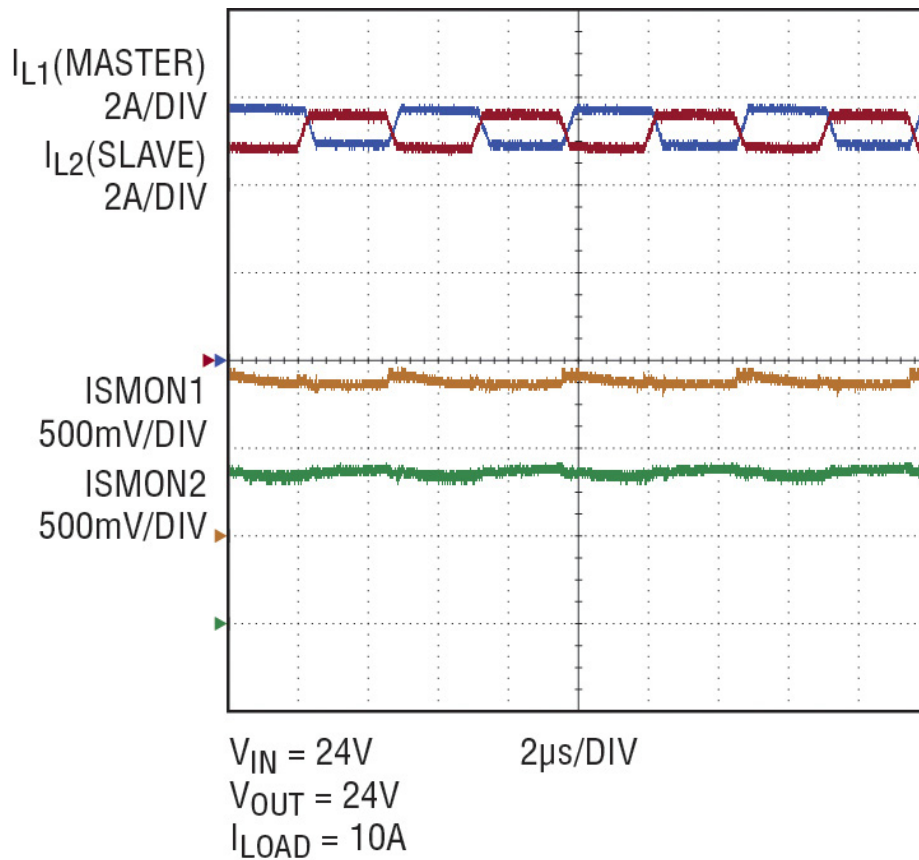
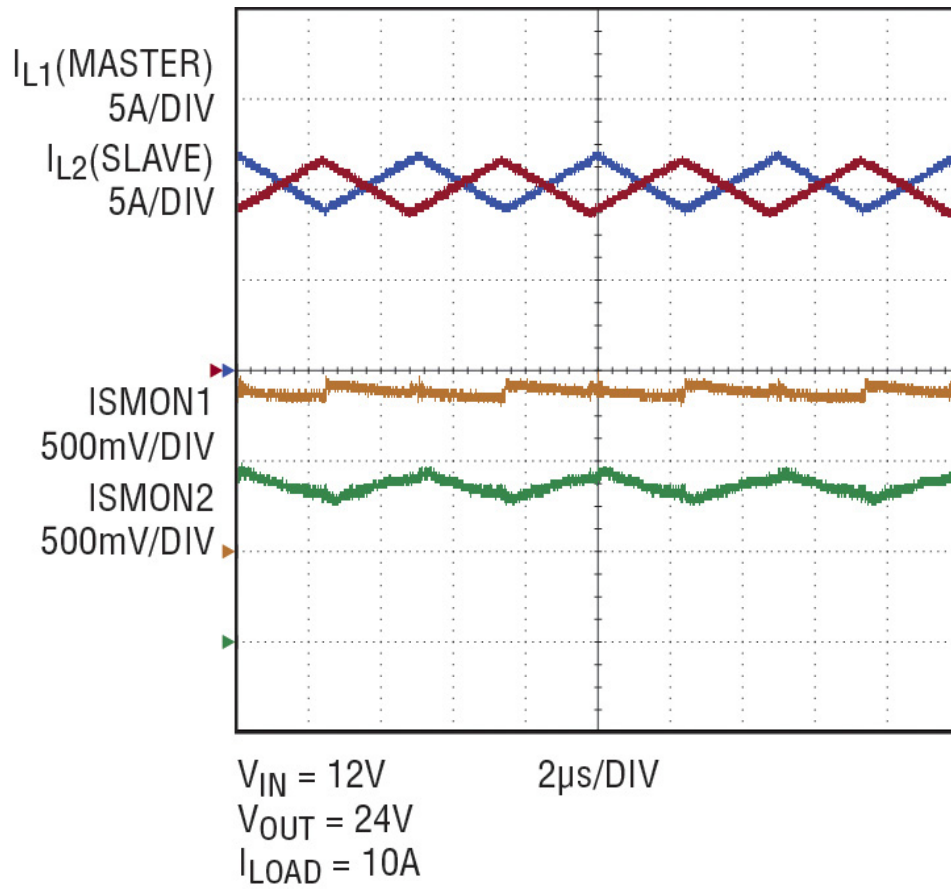


Figura 2. Il convertitore singolo da 24V, 5A illustrato nella Figura 1 presenta un aumento massimo di temperatura su qualsiasi componente pari a 20°C con ingresso a 12V (a) e a 50°C con ingresso a 9V (b). Anche con ingresso a 8V (c), il componente alla massima temperatura non supera i 96,5°C senza bisogno di dissipatori né di flusso d'aria forzata.



34 - Firmware



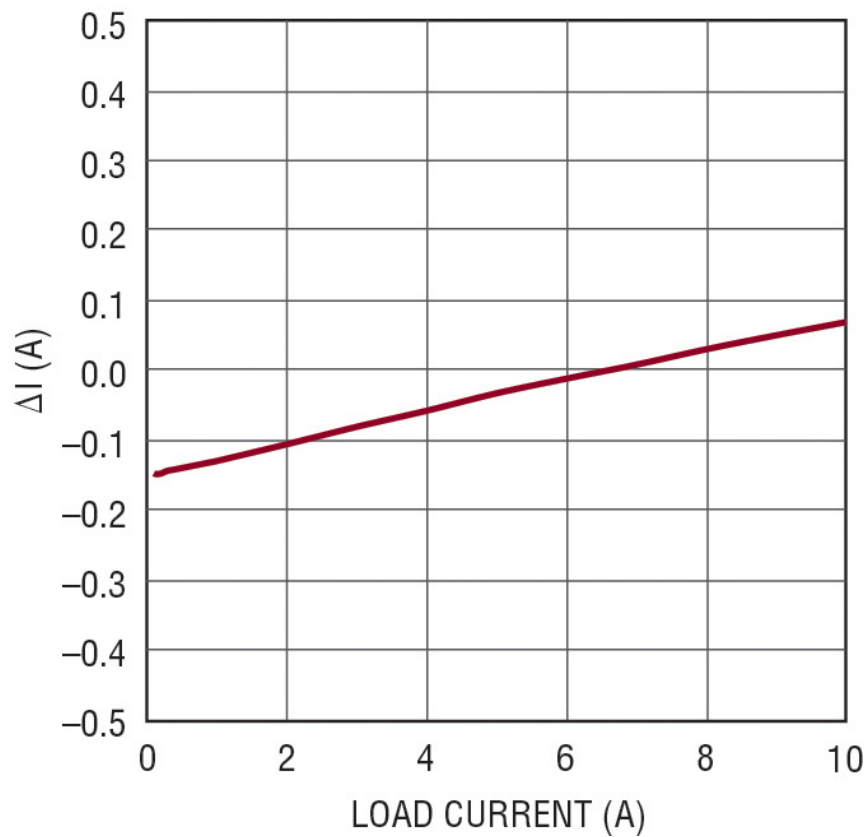
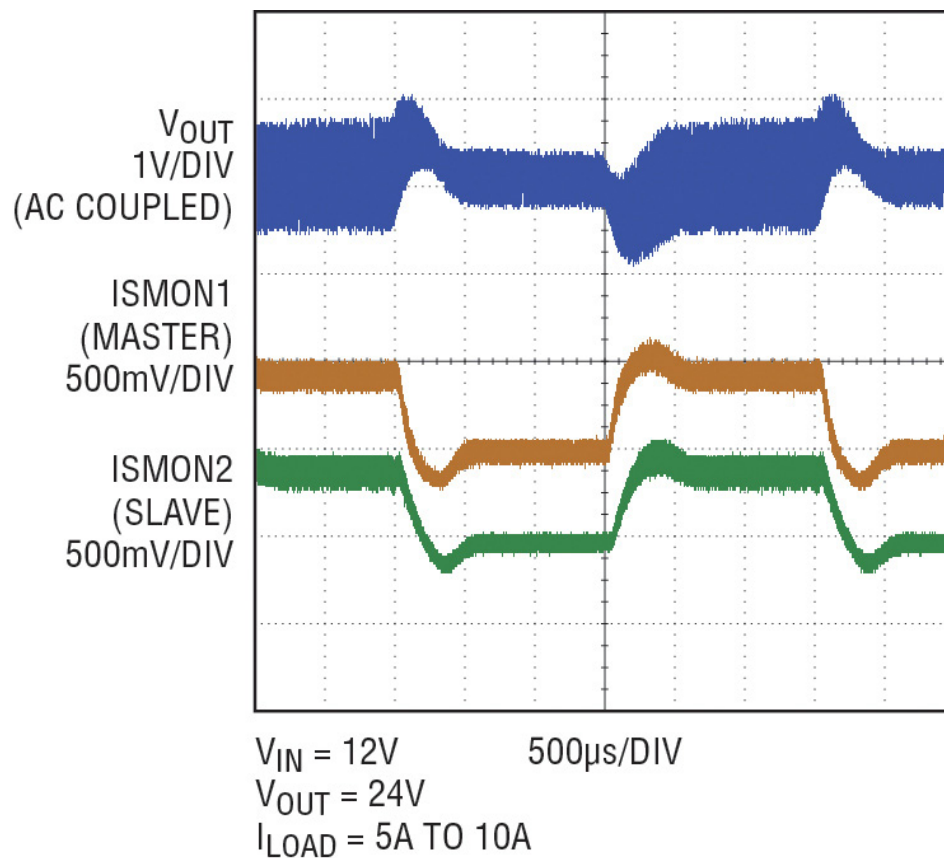


Figura 4. Induttore del convertitore in parallelo e adattamento tra le correnti di uscita



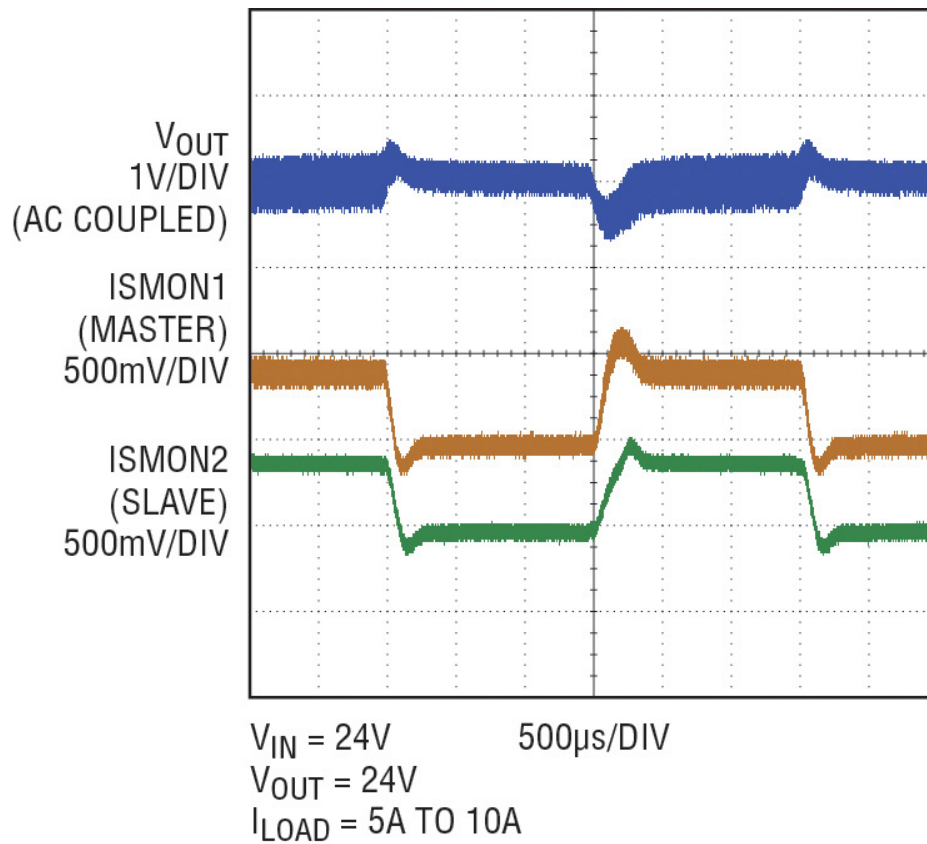


Figura 5. La risposta al transitorio del convertitore in parallelo mostra una condivisione uniforme della corrente

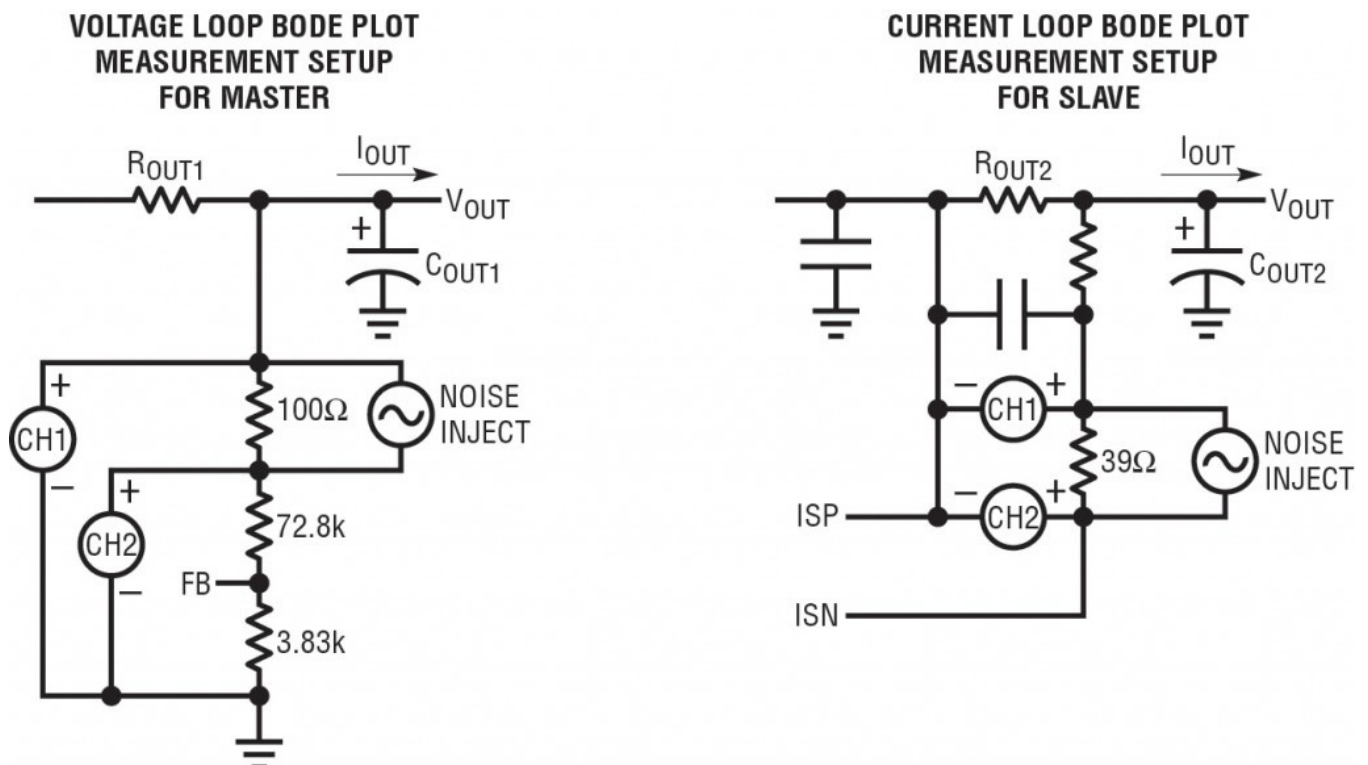


Figura 6. Misura della risposta dell'anello di convertitori in parallelo

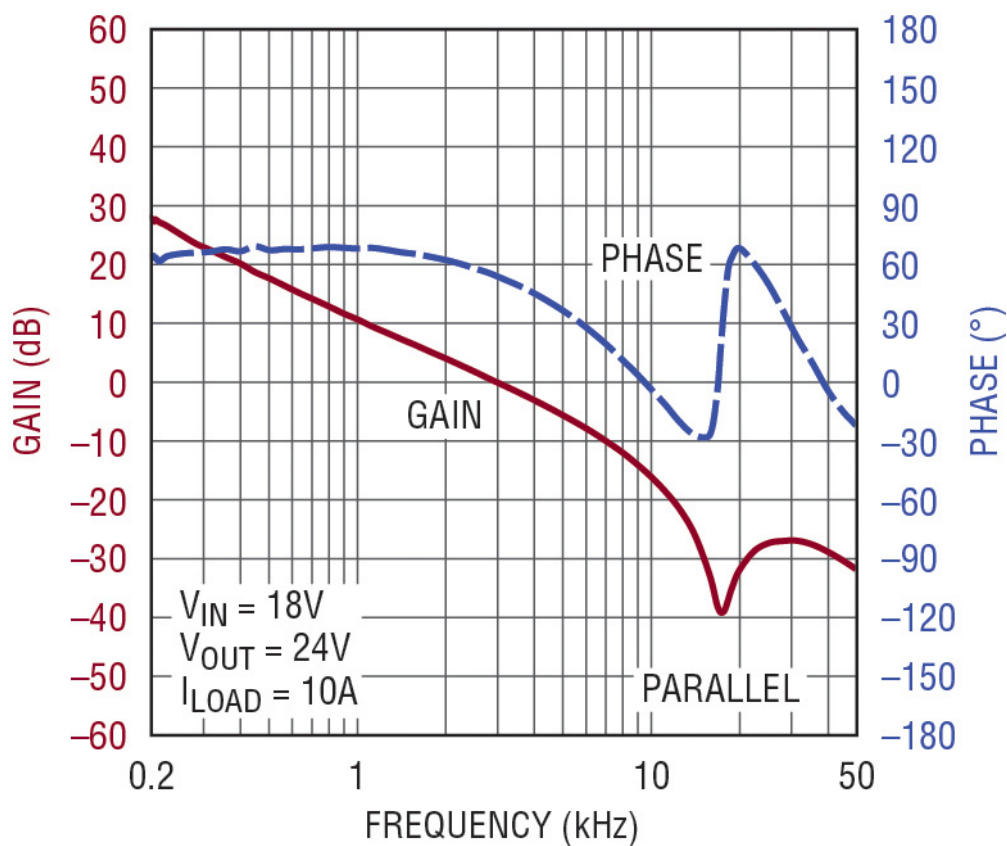


Figura 7. Il diagramma di Bode mostra i risultati della misura per un sistema in parallelo.

L'autore è a disposizione nei commenti per eventuali approfondimenti sul tema dell'Articolo. Di seguito il link per accedere direttamente all'articolo sul Blog e partecipare alla discussione:
<http://it.emcelettronica.com/convertitore-buck-boost-facile-da-collegare-in-parallelo-per-ridurre-al-minimo-gli-aumenti-di-temperatura>



Wireless & Positioning modules for the Internet of Things that really matter



locate, communicate, accelerate

TOBY-L1/L2
LTE modules
24.8 x 35.6 x 2.6 mm



SARA-U2/G3
UMTS/GSM modules
16.0 x 26.0 x 3.0 mm



ODIN-W2
Wi-Fi/Bluetooth modules
14.8 x 22.3 x 4.5 mm



MAX-M8
GPS/GNSS modules
9.7 x 10.1 x 2.5 mm



2G/3G/4G, Wi-Fi/Bluetooth and positioning modules



Connettiti il futuro è con noi

Tel +39 02 30465311 Fax +39 02 33200917

info@kevin.it
www.kevin.it



www.u-blox.com



Applicazioni audio high-quality con PIC32



di **Maurizio**
26 Settembre 2015

Nel progettare docking station e accessori audio per dispositivi digitali portatili, i progettisti sono vincolati dal costo nel tentativo di fornire una riproduzione audio di altissima qualità. Una delle sfide progettuali è quella di eseguire il cambiamento della frequenza di campionamento senza degradare la qualità audio, cercando a sua volta di contenere i costi. I Microcontrollori PIC32 dispongono di soluzioni per ottenere un audio di alta qualità, mantenendo nello stesso tempo un basso costo di progettazione. In questo articolo, analizzeremo alcune specifiche disponibili nei MCU PIC32, con particolare attenzione ai moduli maggiormente utilizzati.

1. INTRODUZIONE

In una docking station una sorgente audio digitale invia un flusso di dati tramite un'interfaccia USB. Il dock cattura il flusso di dati durante l'esecuzione di altri compiti importanti e lo indirizza ad un audio **codec** o **convertitore digitale - analogico (DAC)** con una frequenza di campionamento specifica, come mostrato in Figura 1.

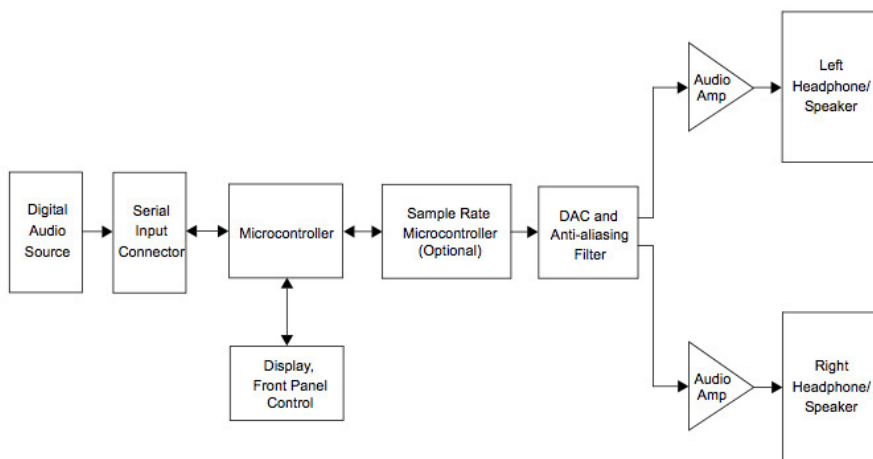


Figura 1: Tipica applicazione audio

Il flusso audio stereo catturato scorre, quindi, attraverso un'interfaccia seriale al codec o **DAC**. Poiché ci sono molte possibili fonti di audio digitalizzato, e non tutti usano la stessa frequenza di



campionamento, l'interfaccia seriale adatta tipicamente la frequenza di campionamento o converte il flusso di dati campionati in un data rate comune.

Pertanto, una delle sfide nella progettazione di sistemi docking è di eseguire il cambiamento della frequenza di campionamento senza degradare la qualità audio e ad un costo il più basso possibile. Per far fronte a queste sfide, i progettisti hanno in genere utilizzato un circuito di conversione apposito e / o un DAC audio high-end che comprende complessi Loops Phase-Locked (PLL) per garantire frequenze di campionamento flessibili per una comunicazione stabile del segnale audio campionato. L'interfaccia USB è una comoda interfaccia per il trasferimento di dati audio. Tuttavia, per soddisfare i requisiti di audio professionale, dovranno essere affrontati problemi di mismatching clock che determinano una perdita della qualità audio. In questo articolo, analizzeremo alcune specifiche audio disponibili nei MCU PIC32; in particolare, il modulo SPI e quello Reference Clock Output. Il modulo SPI supporta diversi modi di comunicazione audio standard e offre una risoluzione di bit elevata per applicazioni audio di alta qualità; il Reference Clock Output, invece, può essere utilizzato per fornire il clock master al front-end analogico per generare le diverse frequenze di campionamento.

2. IL DIGITALE

La tecnica digitale consiste nel campionare l'audio analogico ad una frequenza di almeno due volte la componente di frequenza più alta del segnale (**frequenza di Nyquist**). Pertanto, un segnale audio che si estende da 0 a 20 kHz può esse-

re campionato ad una velocità di trasferimento dati di 44,1 kHz, che in questo caso è il tasso di Nyquist affinché il segnale può essere ricostruito senza aliasing quando è riconvertito nel dominio analogico. Oltre alla frequenza di campionamento, la risoluzione dati può essere a 16-bit o 24-bit. Per compact disc (CD) di qualità, lo standard di risoluzione è a 16 bit con una frequenza di campionamento di 44,1 kHz. Tuttavia, ci sono opzioni che determinano prestazioni più elevate. Uno degli standard di codifica dati con risoluzione a 24 bit aumenta la frequenza di campionamento a 96 kHz. Per l'audio professionale, i file sono codificati con una risoluzione a 24 bit per campione. **La capacità di offrire audio di alta qualità, rende l'interfaccia USB una soluzione facile ed economica per gestire lo streaming dati audio ad alta risoluzione e ad alta frequenza di campionamento.** Le differenze nei tassi di campionamento e risoluzione in bit richiedono un hardware per il sistema di riproduzione, o dock, in grado di gestire le diverse velocità del flusso dati. Per fare questo il sistema deve usare DAC più complessi e costosi, o utilizzare un convertitore di frequenza di campionamento esterno (IC), oppure convertire tutti i campioni in una frequenza di campionamento e bit rate standard, utilizzando un algoritmo in esecuzione su un microcontroller con un semplice DAC a basso costo in grado di gestire le funzionalità.

3. PIC32

La maggior parte dei codec offrono una comunicazione seriale su un'interfaccia a 4 linee. La trasmissione e la **ricezione dei campioni audio**



stereo tra MCU e codec avvengono tramite questa interfaccia le cui caratteristiche possono essere descritte di seguito analizzando i principali segnali:

- Serial Data Output (SDO) per trasmettere dati stereo audio al codec
- Serial Data Input (SDI) per ricevere dati audio del codec
- Serial Bit Clock (SCK / BCLK) è il clock bit fornito dal Master
- Left / Right Clock (LRCK) è il phase clock fornito dal Master per i dati stereo

guenti:

- I2S Format: la modalità I2S (Figura 2) è quando il byte più significativo (MSB) è disponibile sul secondo fronte di salita del BCLK dopo una transizione LRCK.
- Format Left-Justified: la modalità Left-Justified è quando l'MSB è disponibile sul primo fronte di salita di BCLK in seguito ad una transizione LRCK (Figura 3).
- Format Right-Justified: la modalità Right-Justified è quando il byte meno significativo (LSB) è disponibile sul fronte di salita del BCLK precedente ad una transizione LRCK e il MSB è ancora trasmesso per primo (Figura 4).

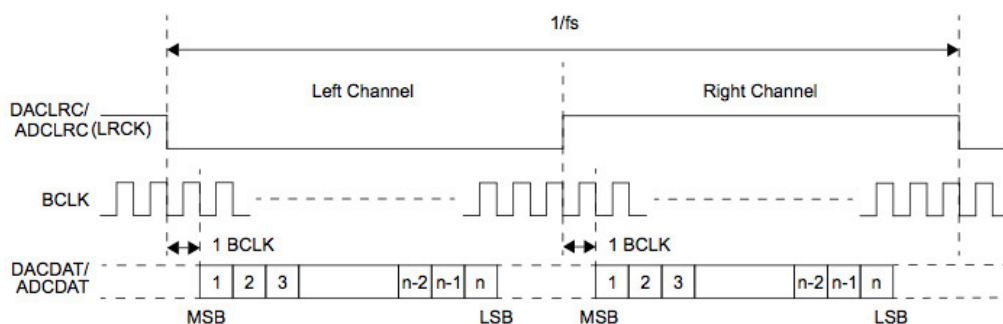


Figura 2: Format I2S

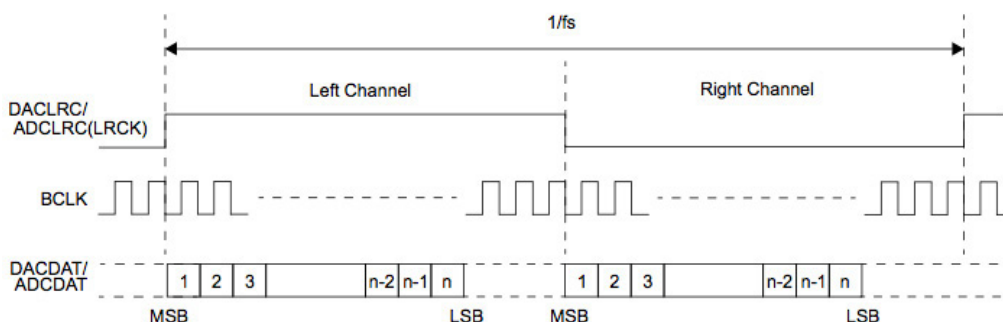


Figura 3: Formato Left-Justified

I dispositivi PIC32 hanno un modulo SPI con Audio mode che offre diversi formati di interfaccia, risoluzioni e **configurazioni Master / Slave**. Le modalità di comunicazione supportati sono i se-

guinti:
- I2S Format: la modalità I2S (Figura 2) è quando il byte più significativo (MSB) è disponibile sul secondo fronte di salita del BCLK dopo una transizione LRCK.

- Format DSP / PCM: l'MSB del canale sinistro è disponibile sul primo fronte di salita del BCLK seguito da un fronte di salita del LRC. A seconda della lunghezza del campione in termini di word, la frequenza BCLK e quella di campionamento, possono verificarsi cicli di BCLK inutilizzati tra il LSB dei dati del canale di destra e il campione successivo (figura 5).

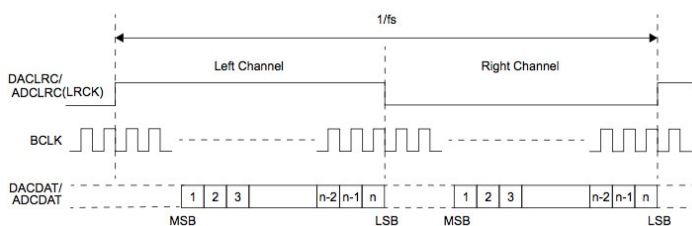


Figura 4: Formato Right-Justified

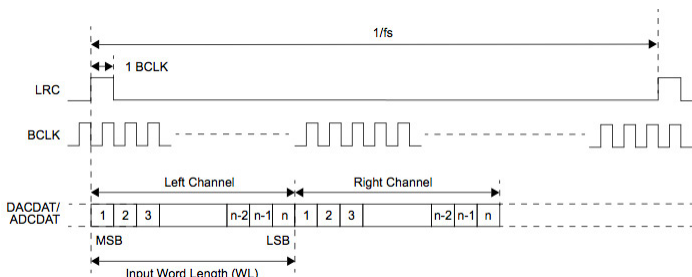


Figura 5: Formato DSP/PCM

La Modalità audio PIC32 supporta 16, 24, e 32-bit di dati stereo con la gestione avanzata degli errori (overflow, bit control). Inoltre, **i clock generati dal BCLK e LRC sono free running quando il PIC32 è un Serial Master**; questo è essenziale per il trasferimento di dati audio in applicazioni di streaming.

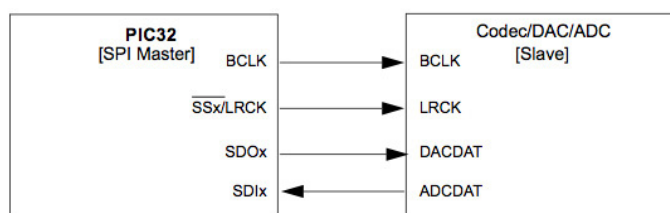


Figura 6: Configurazione PIC32 SPI-Codec con il microcontrollore come Master e il Codec come slave

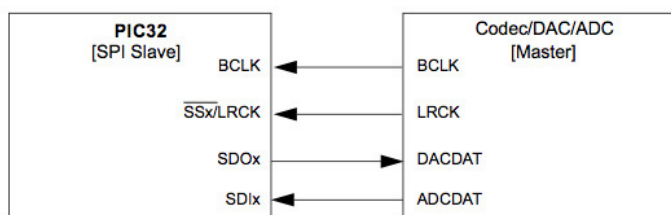


Figura 7: Configurazione PIC32 SPI-Codec con il microcontrollore come slave e il Codec come Master

I codec audio consentono una o entrambe le configurazioni Master/Slave. Nella configurazione SPI Master, il PIC32 fornisce il BCLK e LRCK al codec. Questa configurazione è illustrata nella Figura 6. Nella configurazione Slave, invece, il codec fornisce il clock di bit (BCLK) e lo Stereo phase clock (LRCK) al PIC32. Questa configurazione è illustrata nella Figura 7. **I dispositivi PIC32 forniscono, inoltre, un canale di interfaccia di controllo sul modulo periferico Inter-Integrated Circuit™ (I2CTM).** L'interfaccia di controllo consente di configurare il codec per una specifica modalità di funzionamento configurando i corrispondenti registri.

4. REFERENCE CLOCK OUTPUT

I dispositivi PIC32 hanno un modulo di clock reference (REFCLKO) che può essere utilizzato per generare il clock opportunamente frazionato e utilizzato dai codec audio / DAC per varie frequenze di campionamento. Il clock di riferimento può essere mappato a qualsiasi pin/porta del dispositivo in accordo con il Peripheral Pin Select (PPS). Inoltre, **REFCLKO può essere configurato come sorgente del modulo SPI sul PIC32 invece del clock bus periferico**. Questo garantisce la sincronizzazione del clock USB con quello di canale SPI per minimizzare gli effetti di jitter quando il clock PLL USB è selezionato come sorgente per il REFCLKO. La sorgente di clock per il modulo di uscita di riferimento ha varie opzioni: USB-PLL, primary PLL, POSC, FRC, SOSC, LPRC, PBCLK, clock SYSCLK. La Figura 8 mostra il modulo oscillatore del PIC32 con le diverse sorgenti di clock.

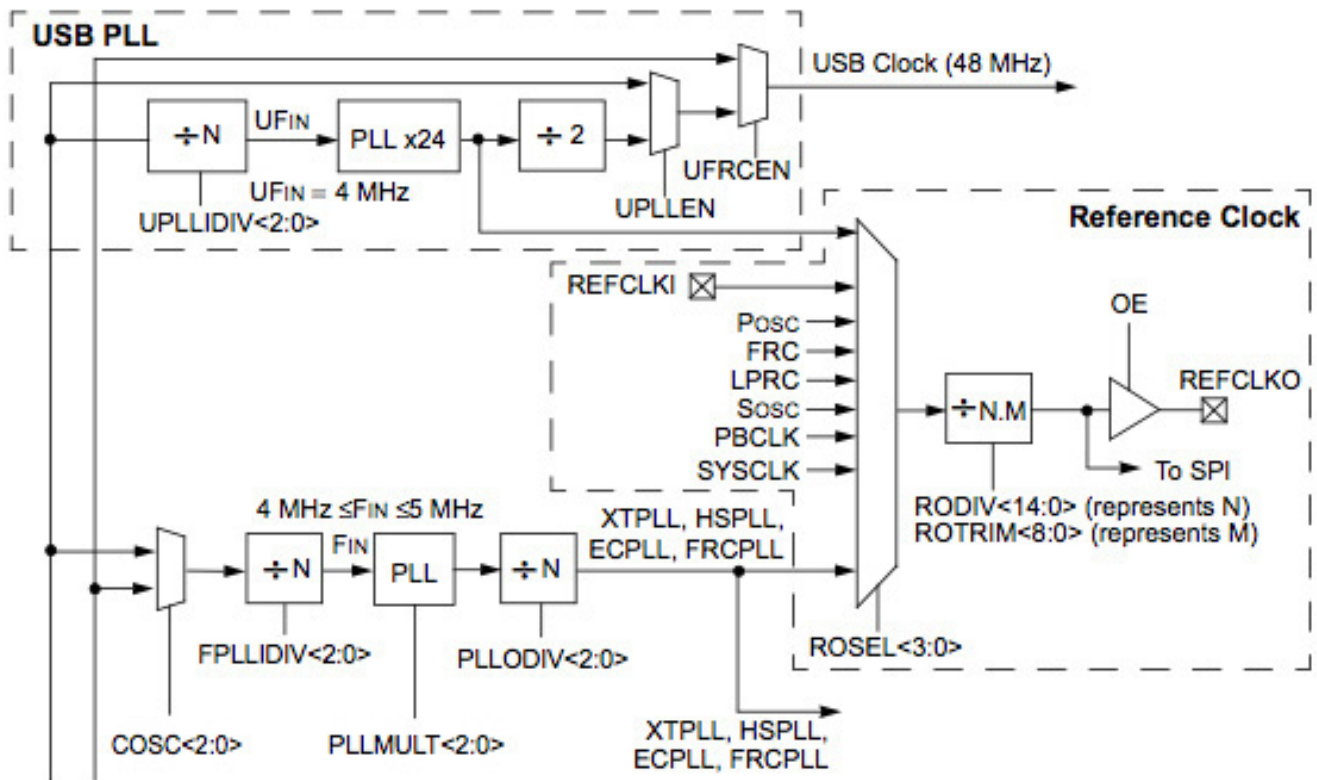


Figura 8: Schema a blocchi dell'oscillatore PIC32 con il modulo Reference Clock Output

Quando il PIC32 SPI è il Master, il BCLK e LRCK sono generati dal dispositivo. Con REFCLKO come sorgente di clock per il modulo SPI, il BCLK può essere configurato caricando i valori appropriati nel registro SPI Baud Rate Generator, SPIxBRG. Se necessario, REFCLKO può fornire l'ingresso di clock master al codec. **Quando il PIC32 SPI è lo Slave, invece, il BCLK e LRCK sono generati dal codec.** Il clock reference è in grado di fornire il clock master per il codec / DAC e può essere cambiato con una precisione molto fine. La risoluzione è compresa tra circa 0,02 a 0,05% in base alla sua frequenza. Questo assicura che la deviazione della frequenza di campionamento è ben

al di sotto dei limiti richiesti dal codec quando il clock di output di riferimento è utilizzato solo come sorgente di clock SPI e non come master al codec. Se il clock reference è usato come master per il codec, invece, la deviazione è limitata solo dalle specifiche del codec stesso. La tabella di figura 9 mostra un esempio di un clock master audio generato da un codec costoso con un PLL standard di 12MHz e un altro con il modulo di riferimento del PIC32. **I dati indicano che il clock generato è molto vicino a quello richiesto.** Questo eliminerebbe la necessità di un PLL sul codec o DAC utilizzato come front end analogico e la necessità di un cristallo esterno utilizzato come base temporale per il codec. Questo si tradurrà in risparmio



sui costi di progettazione audio fornendo al contempo la stessa o una migliore qualità rispetto ad un codec costoso (Figura 10).

Expected Master Clock (MHz)	Average Master Clock using External Crystal (MHz)	Average Master Clock using REFCLKO (MHz)
8.192	8.192	8.192
11.2896	11.2889025	11.28893
12.288	12.288	12.288

Figura 9: Confronto tra il clock master generato usando il codec PLL e il modulo reference clock output su un PIC 32

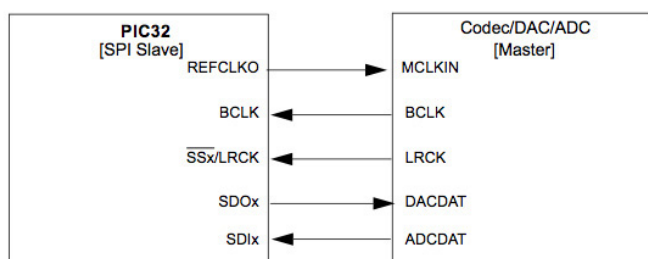


Figura 10: Configurazione codec e SPI

5. USB CLOCK MISMATCH

Le specifiche USB richiedono un budget di tolleranza al fine di ottenere l'immunità alle interferenze radio. Lo streaming in tempo reale deve garantire precisi intervalli di tempo regolari in modo che il DAC può convertire campioni digitali di un segnale analogico con il sample rate previsto. **Il clock DAC che aspetta e che riceve i campioni audio in una determinata frequenza di campionamento, non può mancare nemmeno un solo campione.** Un campione mancante è manifestato come una sporca qualità, in quanto il DAC non riesce a generare una rappresentazione accurata del segnale audio streaming. Su un microcontrollore o microprocessore con un modulo USB

incorporato, il clock proviene da un modulo indipendente come un PLL on-chip con un oscillatore a cristallo esterno di valore specifico. Dal momento che il clock non proviene dall'interfaccia USB, il mismatching introduce effetti di rumore (buffer overrun o underrun) come mostrato in Figura 11.

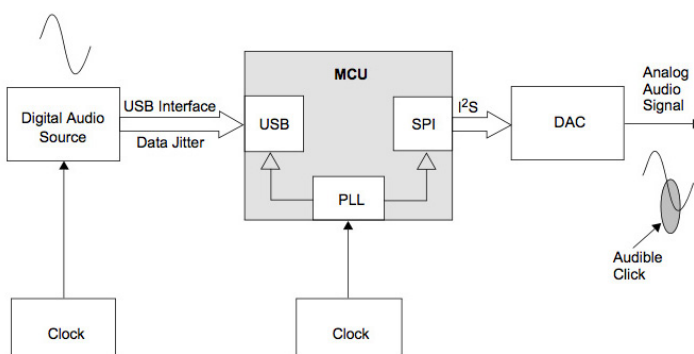


Figura 11: Clock Jitter DAC Underrun/Overrun

Un miglioramento è quello di utilizzare un buon Asynchronous Sample Rate Converter (ASRC) dove la frequenza di campionamento di ingresso è funzione del jitter, con i filtri interni sintonizzati dinamicamente per una nuova frequenza di campionamento. Tuttavia, **un buon ASRC è molto costoso e il sistema richiederebbe ancora un DAC per la conversione analogica.** Come soluzione a basso costo, può essere quello di utilizzare un meccanismo di feedback che monitora il buffer assicurando la stessa qualità di un costoso ASRC. Il modulo reference clock output con il clock USB-PLL come sua sorgente può essere utilizzato per generare e regolare il clock master. Questa funzionalità impedisce il buffer underrun e overrun mantenendo una frequenza di campionamento accettabile con una gamma di oscillazione dello 0,2%.



6. CONCLUSIONI

I moduli SPI con modalità audio e USB On The Go (OTG) in dispositivi **PIC32**, possono essere utilizzati per applicazioni audio di alta qualità. **La versatilità e flessibilità delle caratteristiche dei PIC32 possono fornire una soluzione ottimale pur mantenendo i costi e il consumo energetico molto basso.** Le caratteristiche principali dei dispositivi **PIC32 a sostegno di un audio high-quality** possono essere così riassunte: modulo SPI con vari formati dati tra cui I2S, capacità di gestire flussi di dati audio stereo di 16 e 24-bit, uscita di clock di riferimento flessibile che può essere utilizzato per generare i clock master audio standard, modulo **USBOTG** per lo streaming audio digitale.

L'autore è a disposizione nei commenti per eventuali approfondimenti sul tema dell'Articolo. Di seguito il link per accedere direttamente all'articolo sul Blog e partecipare alla discussione:
<http://it.emcelettronica.com/applicazioni-audio-high-quality-con-pic32#comments>

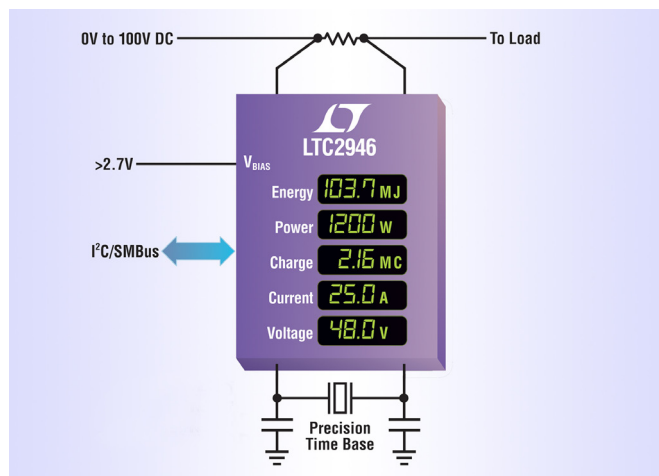
Sistema di monitoraggio dell'energia, operativo da 0V a 100V con precisione 1% e uscita a 12 bit

L'LT2946 è un dispositivo di monitoraggio di carica, potenza ed energia funzionante sia sul lato basso che sul lato alto dei rail di alimentazione DC nell'intervallo da 0V a 100V. Un convertitore analogico-digitale (ADC) integrato a 12 bit con precisione $\pm 0,4\%$ e un time base di precisione esterno (cristallo o clock) consente un'accuratezza di misura superiore a $\pm 0,6\%$ per corrente e carica e a $\pm 1\%$ per alimentazione ed energia. Un riferimento di tempo interno con precisione $\pm 5\%$ si sostituisce in assenza di uno esterno. Tutte le letture digitali, inclusi i valori minimo e massimo di tensione, corrente e potenza, sono memorizzate in registri accessibili tramite un'interfaccia I²C/SMBus. Un'uscita di allarme segnala quando le misure superano le soglie di allarme configurabili, evitando all'host il polling dei dati. L'LT2946 di Linear Technology offre accesso a tutti i parametri necessari per valutare con precisione e gestire il consumo di energia a livello di scheda. Inoltre, l'ampio range operativo rende questo dispositivo ideale per monitorare il consumo di energia della scheda nei server blade, nelle apparecchiature di telecomunicazione, solari e industriali e nelle schede AMC (advanced mezzanine card).

L'LT2946 può essere alimentato da 2,7V a 5V direttamente, da 4V a 100V tramite un regolatore lineare interno oppure oltre i 100 V attraverso un regolatore shunt interno. Due dei tre pin di ingresso/uscita per uso generico (GPIO) sono configurabili per l'abilitazione dell'accumulatore e l'uscita di allarme. Il convertitore analogico-digitale funziona in modalità di scansione continua o in modalità snapshot. In modalità shutdown, il consumo di corrente del dispositivo scende da 900 μ A a 15 μ A.

L'LT2946 è disponibile in due versioni: l'interfaccia I²C dell'LT2946 è dotata di pin di ingresso e uscita SDA separati per l'I²C standard o opto-isolata, mentre l'LT2946-1 è provvisto di un'uscita SDA per l'inversione delle configurazioni dell'opto-isolatore. Specifico per il funzionamento nei range di temperature commerciale, industriale, automotive e militare, l'LT2946 è offerto nei package MSOP a 16 pin e DFN di 4 x 3mm.

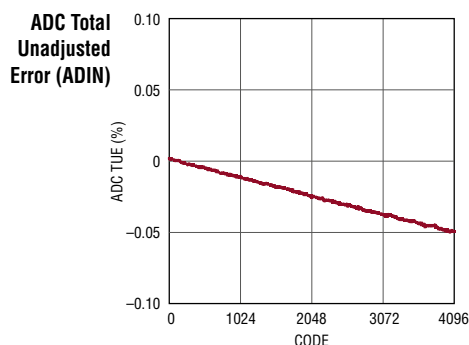
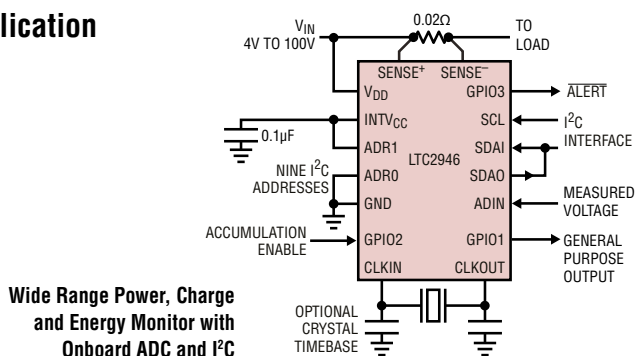
Per ulteriori info: www.linear.com/products/power_monitors



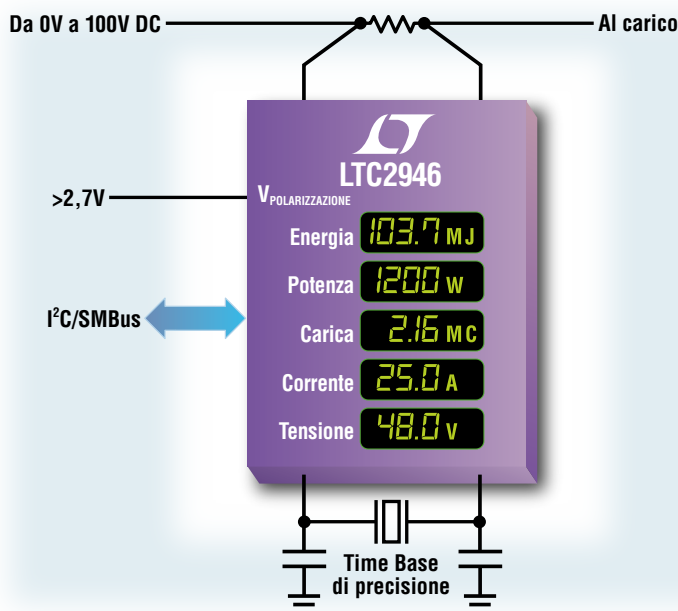
Riepilogo delle caratteristiche: LTC2946

- Controlla l'uso di tensione di alimentazione, corrente, carica, potenza ed energia
- Intervallo di monitoraggio da 0V a 100V, >100V con regolatore shunt interno
- Convertitore analogico-digitale (ADC) a 12 bit con modalità Scan e Snapshot
- Interfaccia digitale I²C/SMBus
- Precisione garantita:
 - $\pm 0,4\%$ per la tensione a 12 bit
 - $\pm 0,6\%$ per la corrente a 12 bit e la carica a 32 bit
 - $\pm 1\%$ per la potenza a 24 bit e l'energia a 32 bit
- $\pm 5\%$ interna, time base esterno o cristallo
- Registratore di valore minimo e massimo
- Range di alimentazione di polarizzazione: da 4V a 100V oppure da 2,7V a 5,9V
- Allarmi per il superamento delle soglie di allarme
- SDA diviso facilita l'opto-isolamento
- Modalità di spegnimento con $I_Q < 40\mu A$
- Package MSOP a 16 pin e DFN di 4 x 3mm

Typical Application



Monitoraggio dell'energia con una precisione pari all'1%



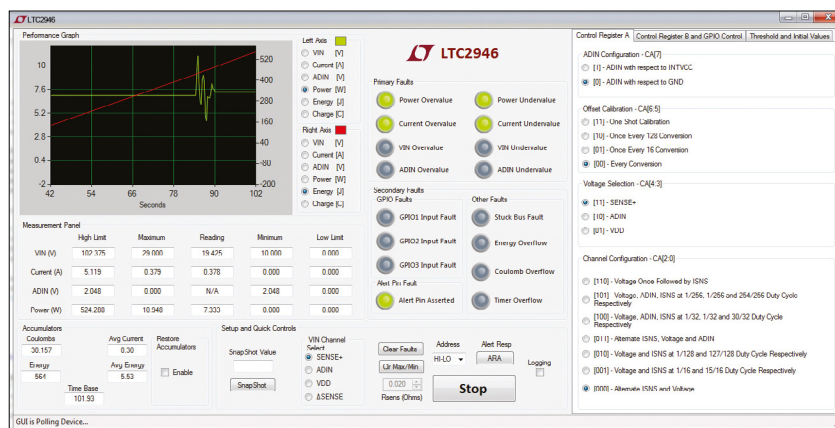
Misurazione semplice e accurata dell'energia dell'alimentazione DC a livello di scheda

L'LT[®]C2946 semplifica il monitoraggio accurato di tutti i parametri necessari per valutare e gestire il consumo energetico di un sistema. I rail di alimentazione DC nell'intervallo da 0V a 100V possono essere monitorati attraverso il rilevamento sia sul lato alto che su quello basso, indipendentemente dalla tensione di alimentazione. L'LT[®]C2946 memorizza in modo pratico le letture digitali, compresi i valori minimi e massimi, in registri accessibili tramite I²C/SMBus. Un allarme segnala quando le letture superano le soglie di allarme configurabili, evitando all'host il polling ripetuto dei dati.

Caratteristiche

- Monitora la tensione di alimentazione DC, la corrente, la carica, la potenza e il consumo di energia
- Intervallo di monitoraggio da 0V a 100V, >100V con regolatore shunt interno
- Convertitore analogico-digitale a 12 bit con modalità Scan/Snapshot
- Precisione garantita: $\pm 1\%$ per potenza ed energia
- Time base interno $\pm 5\%$ o esterno
- Intervallo di alimentazione di polarizzazione: da 4V a 100V o da 2,7V a 5,9V
- L'SDA diviso facilita l'opto-isolamento
- Package MSOP a 16 pin e DFN di 4 x 3mm

Schermata dell'interfaccia grafica utente di LTC2946



LT, LTC, LTM, Linear Technology e il logo Linear sono marchi registrati di Linear Technology Corporation. Tutti gli altri marchi sono di proprietà dei rispettivi titolari.

www.linear.com/product/LTC2946
Tel.: +39-039-596 50 80



STM32 Discovery: Introduzione ed IDE di Sviluppo



di **Gaspare Santaera**
29 Settembre 2015

Nate come alternativa/concorrente di Arduino, le schede Discovery della ST offrono in ridotte dimensioni (65x96mm), o meglio quasi alle stesse dimensioni di un Arduino Due, un hardware che non ha nulla da invidiare ai concorrenti, un prezzo in certi casi addirittura di gran lunga inferiore ed un paio di MEMS (accelerometro e microfono) come valore aggiunto. Tuttavia, osservando l'altra faccia della medaglia, l'utente medio rischia di trovarsi in mano una scheda potente ma di non riuscire a far lampeggiare un led, a causa della mancanza ad oggi per queste schede di un IDE di sviluppo, unico e gratuito, nonché di una community minimamente paragonabile a quella di Arduino.

Questo articolo è il primo di una serie di quattro articoli con i quali tenteremo di mettere un po' di ordine sulla questione programmazione, esplorando le IDE di sviluppo, (uno grafico per il settaggio dei pin della scheda e uno per la scrittura del codice) per poi di volta in volta soffermarci sulle periferiche presenti sulla scheda descrivendole ed utilizzandole in piccoli progetti pratici.

In questo primo capitolo dopo una veloce descrizione di una Discovery andremo alla scoperta delle IDE di sviluppo per poi concludere con un paio di semplici esempi con dei led. Nello specifico useremo la STM32F4 Discovery Board (Fig. 1) che monta un ARM Cortex-M4 a 32 bit,

ovvero un STM32F407VGT6 con una memoria flash da 1 Mb ed una ram da 192Kb, inoltre sulla scheda è presente un tool di debug ST-LINK/V2 utilizzabile tramite un apposito connettore SWD sia per il debug appunto che per la programmazione del microcontrollore a bordo.

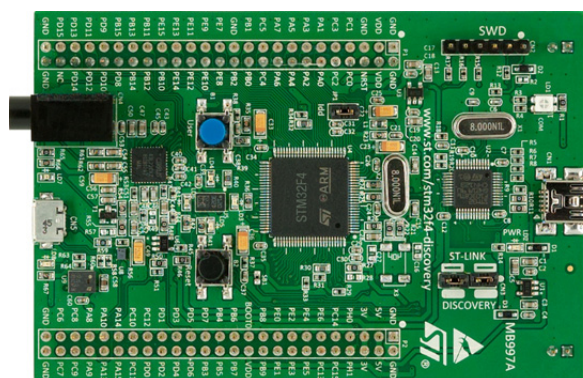


Figura 1: STM32F4 Discovery Board



Infine come accennato la scheda offre due valori aggiunti rappresentati da una accelerometro triassiale (LIS302DL) e un microfono digitale omnidirezionale (MP45DT02) della ST.

Le IDE di Sviluppo

Per la scrittura del codice useremo due differenti IDE, ovvero l'STM32CubeMX (scaricabile direttamente dal sito di [ST Microelectronics](https://www.st.com)) ed ColIDE (scaricabile dal sito di [CooCox](https://www.coo-cox.com)).

STM32CubeMX

L'STM32CubeMx offre la possibilità all'utente meno esperto di creare in modo facile e veloce tutti i file necessari al progetto.

In particolare tramite un interfaccia grafica molto intuitiva l'utente può scegliere quali periferiche utilizzare per il proprio progetto e i pin ad esse collegate, successivamente il programma provvederà in modo del tutto automatico alla creazione di tutti i file necessari nonché all'inizializzazione delle periferiche selezionate. L'utilizzo del programma come detto è semplice, all'avvio

(fig. 2) ci apparirà una schermata per la creazione o l'apertura di un progetto.

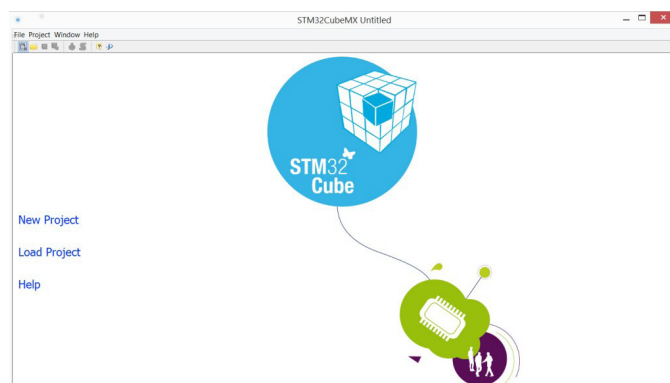


Figura 2: Schermata d'Avvio di STM32CubeMX

Selezioniamo "New Project" e proseguiamo scegliendo il microcontrollore, nello specifico la scheda che useremo monta a bordo un microcontrollore della serie STM32F407VGTx con un package LQFP (Leads Quad Flat Package) a 100 pin (fig. 3).

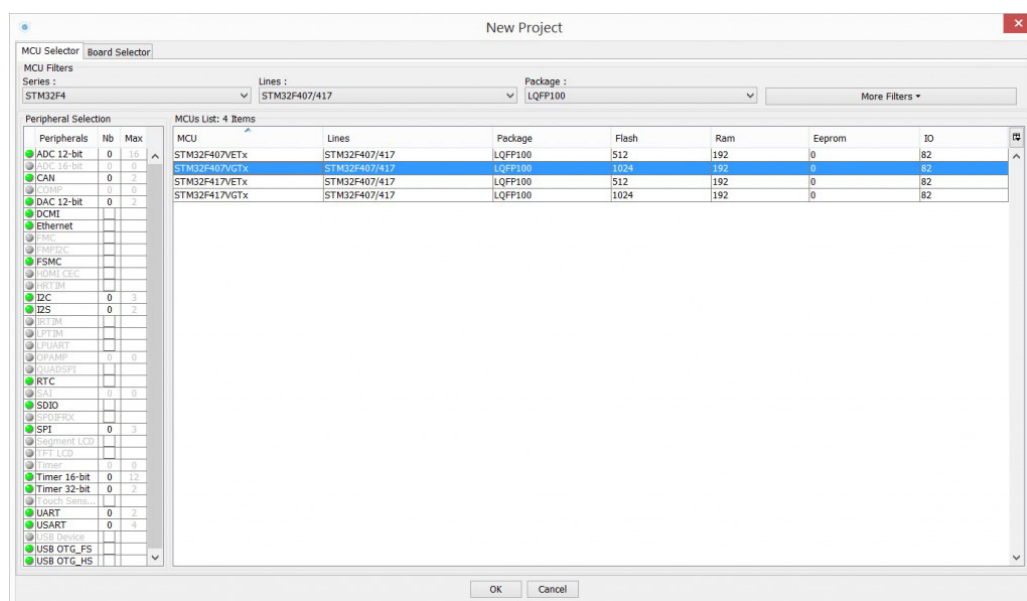


Figura 3: Schermata Selezione MCU



A questo punto ci apparirà una schermata con il microcontrollore selezionato, nel nostro primo esempio utilizzeremo i led integrati nella scheda, che sono connessi ai piedini; PD12 (led verde), PD13 (led arancio), PD14 (led rosso) e PD15 (led blu), selezioniamo quindi i suddetti pin ed impostiamoli come “*GPIO_Output*”, per il nostro primo esempio non serve altro, possiamo quindi passare alla creazione dei file (fig. 4).

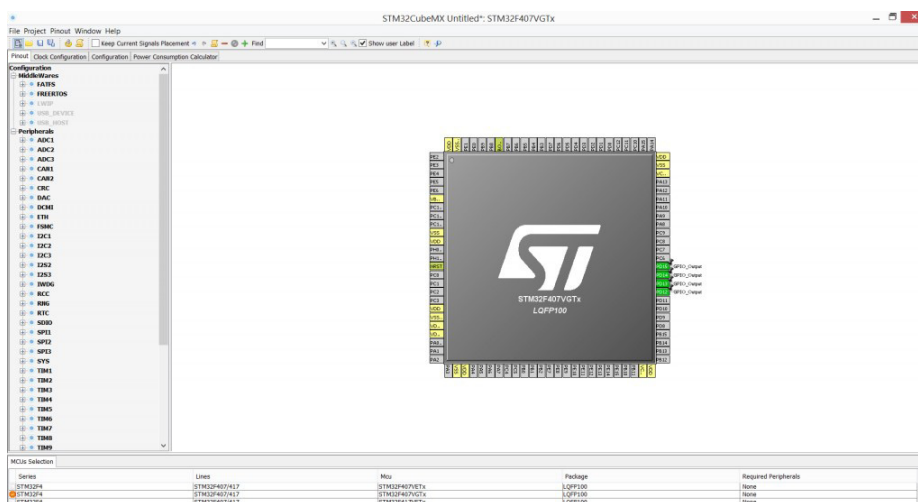


Figura 4: Chip con i Pin Selezionati

Clicchiamo su “*Project*” e quindi “*Generate Code*”, inseriamo tutti i dati relativi al progetto corrente (nome, destinazione, ect ...) e nella combo “*Toolchain IDE*” selezioniamo “*TrueStudio*”, il motivo di questa ultima scelta lo vedremo più avanti. Sul lato “*code generator*” selezioniamo “*Copy only the necessary libraries files*” e clicchiamo sul tasto “*OK*”.

Come già detto il programma provvederà alla generazione di tutti i file contenenti le librerie relative alle periferiche selezionate nonché all’inizializzazione di quest’ultime, alla prima esecu-

zione però il programma ci chiederà di installare dei pacchetti aggiuntivi necessari per i driver delle periferiche (fig. 5), clicchiamo su “*OK*” ed aspettiamo che il programma abbia finito.

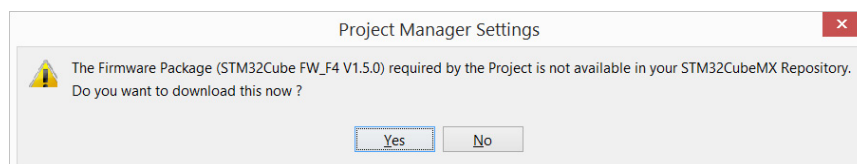


Figura 5: Richiesta di Installazione dei Pacchetti Aggiuntivi

Dopo la creazione dei file, il programma ci chiede se vogliamo aprire il progetto o vedere i file creati nella cartella, clicchiamo su “*Cancel*” e prima di passare all’altra IDE di sviluppo torniamo alla scelta fatta sul menu “*Toolchain IDE*”.

Il Toolchain IDE è l’IDE che l’utente utilizzerà per scrivere

il proprio codice, nel menù STM32CubeMX ci propone cinque differenti tipologie di IDE: EWARM che sta per Embedded Workbench ARM, MDK-ARM v4, MDK-ARM v5, TrueStudio ed SW4ST32, purtroppo alcune di queste IDE non offrono tutti gli strumenti necessari allo sviluppatore, come le due MDK-ARM, non sono gratuite come ad esempio EWARM o lo stesso TrueStudio o presentano una struttura non propriamente intuitiva come SW4ST32.

Nel nostro caso abbiamo scelto TrueStudio ma non lo useremo in quanto useremo un IDE che lavora e che offre le stesse funzionalità di True-



Studio ma totalmente gratuita, ovvero ColIDE.

ColIDE

Per scaricare ColIDE è necessario registrarsi al sito di CooCox. Installato il programma prima di iniziare a scrivere il nostro codice dobbiamo importare e modificare un po' i file generati da STM32CubeMX, infatti, quando si crea un nuovo progetto il ColIDE provvede alla creazione dei file necessari alla compilazione, alcuni di questi file però sono già stati creati da STM32CubeMX e quindi bisogna fare un po' di pulizia eliminando i file di troppo.

Apriamo ColIDE e creiamo un nuovo progetto, scegliamo la scheda che stiamo utilizzando, ovvero “STM32F407VG”, “New Project” scegliamo una cartella vuota come cartella di progetto e chiamiamo il progetto “led1”. Al passo successivo, attraverso il Repository dobbiamo scegliere quali periferiche verranno utilizzate, cerchiamo ed aggiungiamo “coocox-master/STM32F407xx_cmsisboot” cliccando sul relativo tasto “add”. A questo punto nella cartella “components”, creata tra le cartelle di progetto eliminiamo tutti i file lasciando soltanto il file di startup, in quanto gli altri file li prenderemo tra quelli creati da STM32CubeMX. Nella finestra sulla destra dovremmo vedere i file di progetto come nella figura 6.

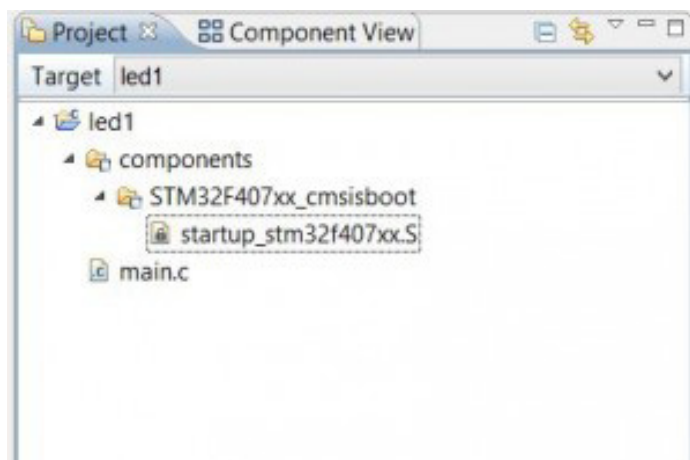


Figura 6: File Generati da ColIDE

Importiamo gli altri file creati da STM32CubeMX, per questi basta andare nella cartella precedentemente usata da STM32CubeMX e trascinare le cartelle “Drivers” ed “Inc” nonché tutti i file contenuti nella “Src” nel target led1, nel quale è contenuto il file main che andremo ad utilizzare;cancelliamo quindi l’attuale file main creato da ColIDE. A questo punto con i file abbiamo quasi finito dobbiamo soltanto cancellare il file “startup_stm32f407xx.S” contenuto nella cartella “ /Drivers/CMSIS/Device/ST/STM32F4xx/Source/Templates/gcc “ per evitare che vada in conflitto con quello contenuto nella cartella “STM32F407xx_cmsisboot”, nella figura 7 viene mostrato un riepilogo dei file del progetto.



Figura 7: File di Progetto

Inseriti i file passiamo al compilatore specificandogli quali sono i file aggiuntivi da includere nel progetto. Clicchiamo su “view” -> “configuration” e nella sezione “Defined Symbols” aggiungiamo “**STM32F047XX**” ed “**USE_HAL_DRIVER**”, come mostrato nella figura 8 (evidenziati in giallo).

Adesso possiamo finalmente compilare per la prima volta il nostro progetto, prima però assicuriamoci che il chip che stiamo usando sia definito tra i file di progetto. Per questo apriamo il file “stm32f4xx.h” contenuto tra i file di progetto nel percorso “Driver\CMSIS\Device\ST\STM32F4xx\Include”, alla riga 76 dovremmo trovare la definizione della scheda che stiamo usando, nel caso in cui la riga fosse ancora commentata provvediamo ad abilitarla “**#define STM32F407xx**”.

Per la compilazione del progetto clicchiamo su “project” -> “build”, alla prima compilazione ColIDE ci chiederà di impostare il compilatore da usare, lavorando con dei processori ARM useremo il “gcc-arm-none-eab” che è possibile scaricare dall’apposito sito (una volta installato, bisognerà indicare il path della cartella bin. Ad esempio “C:\Program Files (x86)\GNU Tools ARM Embedded\4.8 2014q3\bin”).

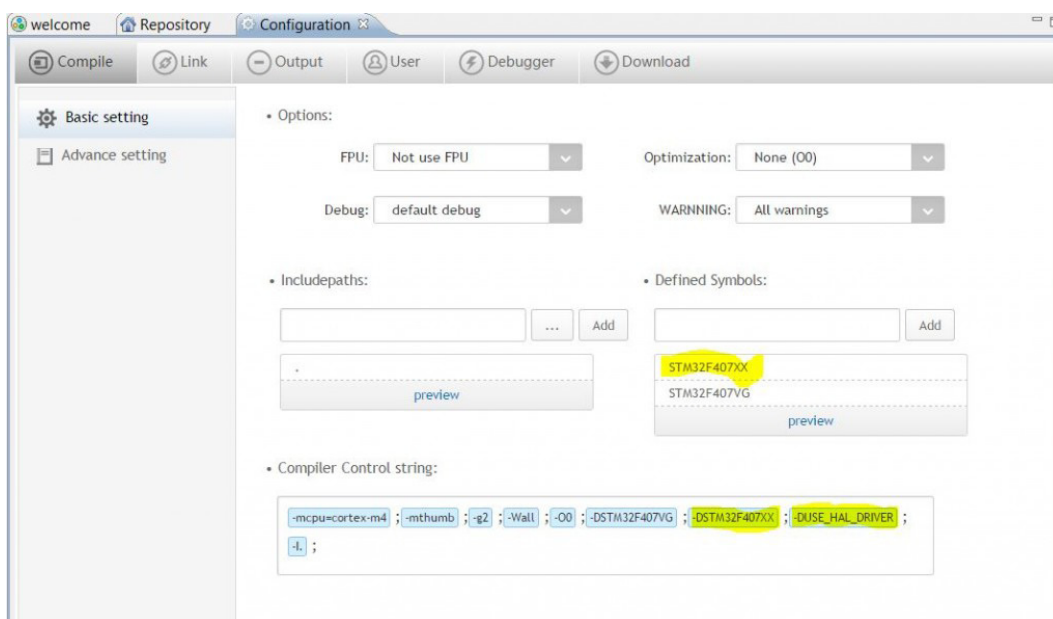


Figura 8: Settaggi

Se la compilazione è andata bene possiamo finalmente scrivere il codice. Nel nostro primo esempio (scaricabile [qui](#)) faremo lampeggiare i LED presenti sulla board in modo sequenziale. Prima vengono accesi tutti insieme, poi a gruppi di due (i due orizzontali e i due ver-

ticali) ed infine uno alla volta in senso orario. Tutte le modalità sono intervallate di circa 1 se-



condo. Bisogna notare che lo scopo di questo primo articolo non è implementare correttamente il comportamento periodico di un programma, useremo infatti una variabile “*counter*” (opportunamente dimensionata in modo da simulare un delay con granularità di 1 ms) che decremente-remo simulando il comportamento di un timer. Pertanto al main generato da STM32CubeMX, oltre alle variabili necessarie sono state aggiunte cinque funzioni, ovvero “Blinking1”, “Blinkin2”, “Blinking3”, “Delay_Init” e “Delay_Millis”.

La struttura del codice è molto semplice all’avvio il programma chiama le quattro funzioni generate automaticamente da STM32CubeMX, nell’ordine “*System_Init*”, “*HAL_Init*”, “*SystemClock_Config*” e “*MX_GPIO_Init*”.

Tralasciando le prime tre per le quali rimandiamo a testi più specialistici spendiamo due parole su “*MX_GPIO_Init*” questa si occupa di inizializzare la struttura “*GPIO_Init_Structure*” definita in precedenza come variabile globale, seguendo le impostazioni su pin e tipologia di periferica che avevamo specificato in STM32CubeMX.

Dopo queste quattro funzioni di default nel main viene chiamata la funzione “*Delay_Init*” che si occupa semplicemente di assegnare un valore alla variabile “*multiplier*”, questa dipende dalla frequenza di clock del microcontrollore e verrà successivamente utilizzata dalla funzione “*Delay_Millis*” al fine di stabilire una relazione tra il numero di colpi di clock del processore e i millisecondi.

Finita la prima parte di inizializzazione si entra infine in un loop infinito (`while(1)`) nel quale il microcontrollore dapprima spegne tutti i LED e

successivamente non fa altro che controllare la variabile “*mode*” chiamando tramite una struttura “*switch*” le tre diverse funzioni di blinking.

Infine la variabile “*mode*” viene prima incrementata di uno e poi a questa viene assegnato il valore dato dal modulo tre del proprio valore. Tale scelta ovviamente è motivata dal fatto che abbiamo tre diverse funzioni di blinking e vogliamo attivarle tutte in modo sequenziale, pertanto non facciamo altro che incrementare una variabile e vedere successivamente quanto vale il modulo tre di quest’ultima, in tal modo ci assicuriamo che questa assuma sempre un valore intero compreso tra 0 e 2.

I vari LED vengono gestiti in modo molto semplice dalla funzione “*HAL_GPIO_WritePin()*” alla quale vanno passati in ingresso la porta che si vuole controllare (A,B,C,D, ect ...), il pin specifico di quella porta (1,2,3,ect ...) e lo stato da assegnare al pin (0 = off, 1 = on).

Tornando alle funzioni di blinking si vede abbastanza bene che, una volta stabiliti i pin tramite STM32CubeMX, per controllare un led basta una semplice riga di codice.

A questo punto, dopo aver modificato il nostro file “*main.c*” non ci resta che ricompilare (“*project*” -> “*build*”) e caricare il codice sulla nostra Discovery tramite “*Flash*” -> “*Program Download*”. Durante la fase di caricamento vedremo il Led accanto alla porta USB di alimentazione e flash, lampeggiare verde e rosso. Quando il LED ritornerà rosso fisso vorrà dire che il nostro programma è stato caricato sulla Discovery.

Timer



Dopo aver fatto lampeggiare alcuni led con il primo semplice esempio ed aver imparato ad usare CubeMX e ColIDE, introduciamo una delle tante periferiche presenti generalmente sui microcontrollori, che ci tornerà utile nei prossimi appuntamenti, ovvero i timers.

Prima di introdurre i timers ed i contatori spendiamo due parole sul codice in generale, generalmente quando un utente medio scrive un codice procede in modo verticale scrivendo un codice sequenziale (se ad esempio si accoppia un attuatore ad un sensore; si legge il sensore, si confronta il valore letto con una soglia prestabilita e si dà il comando all'attuatore, il tutto all'interno di un loop infinito), le istruzioni vengono eseguite una dopo l'altra in modo sequenziale eseguendo una determinata istruzione solo dopo aver terminato la precedente. Questo modus operandi per molte applicazioni comuni è più che sufficiente, tuttavia in alcune applicazioni può essere fondamentale eseguire alcune operazioni in istanti prefissati di tempo indipendentemente da tutte le altre operazioni presenti nel programma. Ad esempio in un cronometro il microcontrollore deve occuparsi della gestione dei tasti, della gestione del display ma soprattutto del conteggio del tempo. In questi casi una logica sequenziale è fallace in quanto se ad esempio il cronometro ha una risoluzione di 1 ms e il microcontrollore impiega 6 ms per il controllo del display non sarà mai possibile scrivere sul display il tempo e successivamente incrementare quest'ultimo di un millisecondo, in quanto il tempo mostrato non sarebbe mai esatto.

In questi casi si ha la necessità di operare in modo orizzontale ovvero di eseguire delle operazioni in modo parallelo. Poiché un microcontrollore normalmente non dispone dell'hardware adatto ad eseguire operazioni parallele si utilizzano le periferiche di timing.

Queste periferiche vengono utilizzate dal microcontrollore per la generazione di interrupt ad istanti prefissati di tempo, quando un interrupt viene generato il microcontrollore congela le operazioni che stava facendo e corre ad eseguire quelle relative all'interrupt.

In questo modo si garantisce l'esecuzione delle istruzioni relative all'interrupt in istanti prefissati di tempo. Tornando all'esempio del cronometro il modo migliore di agire è quello di far gestire i tasti ed il display al microcontrollore dentro un loop infinito e di affidare l'incremento del contatore del tempo ad un interrupt con cadenza di un 1 ms, in questo modo ogni millisecondo il microcontrollore congelerà quello che stava facendo, eseguirà le istruzioni di interrupt, ovvero incrementerà il contatore e tornerà a quello che stava facendo fino al nuovo interrupt. Ovviamente una logica di questo tipo ha senso se è solo se le operazioni eseguite dentro l'interrupt richiedono un tempo di esecuzione inferiore al tempo stesso tra un interrupt e un altro.

Normalmente le periferiche di timing all'interno di un microcontrollore sono di due tipi: Counter e Timer.

I timers hanno un funzionamento simile ad degli orologi, infatti, ad ogni colpo di clock il timer incrementa un registro contatore, quando il contatore raggiunge un determinato valore, o



semplicemente va in overflow, il timer può generare un interrupt. E' quindi possibile generare interrupt ad istanti precisi di tempo conoscendo la frequenza di clock del microcontrollore e fissando il valore della soglia che il contatore deve superare per generare l'interrupt.

I parametri caratteristici dei timer sono il range e la risoluzione. Per range di un timer si intende la massima quantità di tempo che un timer può contare, ovvero il massimo tempo tra due interrupt successivi. Viceversa la risoluzione indica il minor tempo possibile tra due interrupt successivi.

I contatori hanno un funzionamento molto simile ai timers, anche per i contatori, infatti, viene incrementato un registro contatore in occasione di un determinato impulso o evento che può essere interno al microcontrollore

come lo è un colpo di clock per i timers o esterno a quest'ultimo come ad esempio un colpo di clock da un dispositivo esterno.

Nel caso del counter è possibile usare un interrupt generato da un timer per incrementare il proprio registro contatore il quale verrà poi usato per generare interrupt di range maggior rispetto a quelli generabili dal singolo timer. Per il settaggio dei timers spesso si utilizzano diversi registri quelli storici o più comuni sono: il prescaler e il postscaler. Il registro di prescaler è presente in quasi tutti i timers del microcontrollore e si usa per dividere la frequenza di clock di

quest'ultimo in modo da incrementare il registro contatore del timer con frequenze più basse, ovvero più lentamente. In questo modo si aumenta sia il range che la risoluzione del timer. Il postscaler invece si trova a valle dei timers e fissa il moltiplicatore per la generazione dell'interrupt, ovvero, quanti overflow del timer sono necessari a creare un interrupt

I microcontrollori a bordo delle schede Discovery montano svariate periferiche timers e nello specifico il microcontrollore che stiamo usando monta ben 12 timer (fig. 9).

Timer	Type	Resolution	Prescaler	Channels	MAX INTERFACE CLOCK	MAX TIMER CLOCK*	APB
TIM1, TIM8	Advanced	16bit	16bit	4	SysClk/2	SysClk	2
TIM2, TIM5	General purpose	32bit	16bit	4	SysClk/4	SysClk, SysClk/2	1
TIM3, TIM4	General purpose	16bit	16bit	4	SysClk/4	SysClk, SysClk/2	1
TIM9	General purpose	16bit	16bit	2	SysClk/2	SysClk	2
TIM10, TIM11	General purpose	16bit	16bit	1	SysClk/2	SysClk	2
TIM12	General purpose	16bit	16bit	2	SysClk/4	SysClk, SysClk/2	1
TIM13, TIM14	General purpose	16bit	16bit	1	SysClk/4	SysClk, SysClk/2	1
TIM6, TIM7	Basic	16bit	16bit	0	SysClk/4	SysClk, SysClk/2	1

Figura 9: Timer a Bordo del Microcontrollore

Blinking LEDs tramite interruzione Timer

Dopo aver introdotto il concetto di periferiche di timer vediamo ora con un semplice esempio come utilizzarli nelle nostre applicazioni. A tal proposito continuando a giocare con il led a bordo della scheda vediamo come far lampeggiare il singolo led ad istanti di tempo precisi indipendentemente dagli altri. Nel esempio quindi elimineremo quella che era la variabile "counter" introducendo un timer, tale strategia di control-



lo, oltre ad essere più corretta dal punto di vista implementativo, risulta estremamente più precisa rispetto alla precedente.

Per questo esempio useremo il timer TIM2, che viene gestito da un counter a 32 bits ed è collegato al APB1 bus.

Apriamo STMCubeMX e seguiamo i passaggi visti in precedenza per creare un nuovo progetto, come mostrato in figura 4, selezioniamo di nuovo i pin PD12, PD13, PD14, PD15 come “GPIO_Output” ed abilitiamo il timer cliccando su TIM2 nella colonna a sinistra, per quest’ultimo scegliamo come sorgente “Internal clock”, come mostrato in figura 10. A questo punto, aprendo il menù relativo alla configurazione della nostra

discovery, cliccando su “Configuration”, modifichiamo i parametri rela-

tivi al TIM2, in particolare dobbiamo impostare il “Counter Period” a 8000000 e selezionare nel “Trigger Event Selection” l’opzione “Update Event” (figura 11).

Con queste impostazioni abbiamo abilitato il timer TIM2 in modo da generare un’interruzione del tipo Update Event ogni 0.5 secondi, ovvero ad una frequenza di 2 Hz (in quanto come è possibile vedere dal menù “Clock Configuration” di STMCubeMX, i parametri di default per il clock sono di 16 MHz per il SysTick al quale è collegato il bus APB1, a cui a sua volta è colle-

gato il TIM2).

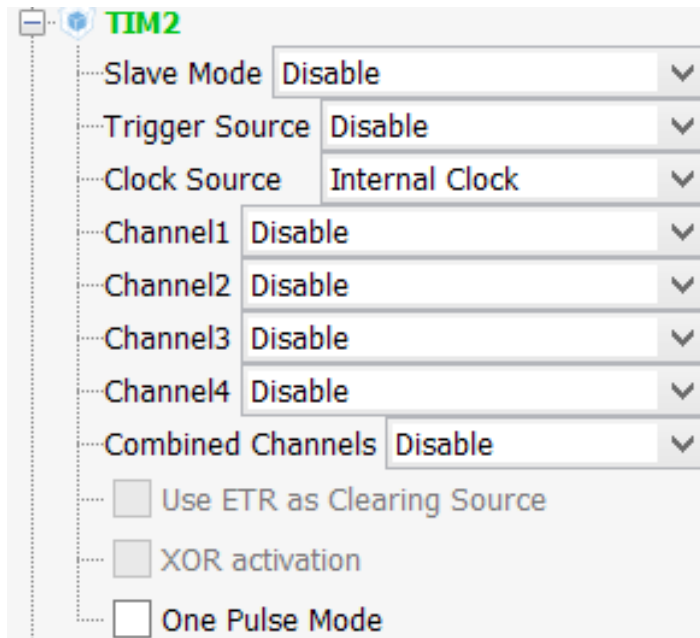


Figura 10: Configurazione Timer

Counter Settings	
Prescaler (PSC - 16 bits value)	0
Counter Mode	Up
Counter Period (AutoReload Register - 32 bits value)	8000000
Internal Clock Division (CKD)	No Division
Trigger Output (TRGO) Parameters	
Master/Slave Mode	Disable (no sync between this TIM (Master) and its Slaves
Trigger Event Selection	Update Event

Figura 11: Parametri del Timer

Infine provvediamo ad abilitare, dal menù “Configuration” il “Nested Vector Interrupt Controller (NVIC)” per il “TIM2 global interrupt” (figura 12), in questo modo CubeMX provvederà alla generazione della funzione “TIM2_IRQHandler(void)”.



Interrupt Table	Enabled	Preemption Priority	Sub Priority
Non Maskable Interrupt	☐	0	0
Memory management fault	☐	0	0
Pre-fetch fault, memory access fault	☐	0	0
Undefined instruction or illegal state	☐	0	0
Debug Monitor	☐	0	0
System tick timer	☒	0	0
RCC global interrupt	☐	0	0
TIM2 global interrupt	☒	1	0

Figura 12: Nester Vector Interrupt Controller

Generato il codice da CubeMX, creiamo un nuovo progetto in ColIDE e rifacendo i passaggi già visti importiamo le cartelle “Driver”, “Inc” e “Src” e sostituiamo il file “main.c” con quello relativo a questo nuovo esempio, scaricabile da qui [Timer_BlinkingLED.c](#).

Come accennato, l’abilitazione del NVIC, porta alla generazione della funzione “TIM2_IRQHandler(void)” che all’interno richiama l’handler dell’interruzione associato al timer TIM2 “HAL_TIM_IRQHandler(&htim2)”, quest’ultima controllata la causa dell’interruzione, manda in esecuzione una funzione di callback.

In questo secondo esempio, utilizzeremo il timer per far lampeggiare i LEDs ad una precisa frequenza. Dalle impostazioni sappiamo che ogni 500 ms (2 Hz), viene generata un’interruzione e richiamata la callback che esegue il codice relativo all’interruzione.

Nel nostro caso specifico avendo scelto tra le opzioni “Update Event” andrà in esecuzione la callback “HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)”.

Per evitare equivoci, dovuti al passaggio di codice tra CubeMX e ColIDE ed assicurarci che il microcontrollore esegua la callback desiderata è opportuno in questi casi eliminare le altre possibili callback che portano lo stesso nome,

pertanto commenteremo la “__weak void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)” che si trova alla riga 4226 ca nel file “Drivers/STM32F4xx_HAL_Driver/Src/stm32f4xx_hal_tim.c”.

Spiegati tutti i passaggi un po più tecnici non ci resta che spiegare in modo più semplice cosa fa il codice di questo secondo esempio. A questo punto non ci resta che spendere due parole per spiegarne il funzionamento.

Ogni volta che il timer genera un’interruzione viene chiamata la funzione “void TIM2_IRQHandler(void)” questa a sua volta contiene tutte le funzioni da richiamare in funzione della causa che ha generato l’interrupt, nel nostro caso è stato un “Update Event”, a generare l’interruzione, pertanto viene chiamata la funzione associata a questo tipo di interruzione, ovvero “HAL_TIM_IRQHandler(&htim2)” alla quale è associata la callback “HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)”.

In questo nuovo esempio all’interno della callback, abbiamo dichiarato 4 variabili statiche che utilizziamo per controllare l’accensione e lo spegnimento dei LEDs.

In particolare ad ogni variabile è associato un led, se la variabile è uguale a 0 il relativo led viene acceso (“HAL_GPIO_WritePin(.., .., 1)”) altrimenti il led viene spento (“HAL_GPIO_WritePin(.., .., 0)”).

Nel nostro semplice esempio facciamo lampeggiare i 4 leds a frequenze diverse, ovvero, il led



connesso al pin “PD12” lampeggia con una frequenza di 2 Hz (500ms), quello connesso al pin “PD13” con una frequenza di 1 Hz (1s), quello connesso al pin “PD14” con una frequenza di 0.5 Hz (2s) ed infine quello connesso al pin “PD15” lampeggia con una frequenza di 0.25 Hz (4s) secondi.

Per ottenere questo risultato è sufficiente incrementare in maniera opportuna le variabili statiche associate ai relativi led, ad esempio la variabile associata al led connesso al pin “PD12”, che lampeggia con una frequenza di 2 Hz cambia stato ad ogni interruzione, pertanto per farla passare dal valore 0 a 1 e viceversa basta incrementarla ad ogni esecuzione della callback e poi assegnarli se stessa in modulo 2. Stesso discorso per la variabile associata al pin “PD13”, questa che lampeggia con una frequenza dimezzata rispetto alla precedente e viene pertanto incrementata ogni due esecuzioni della callback, salvo poi riassegnarle se stessa in modulo 2. Analoga logica per le altre due variabili che vengono incrementare rispettivamente ogni quattro ed otto esecuzioni della callback. Questo esempio semplicissimo dimostra come sia possibile eseguire porzioni di codice in modo preciso ad istanti determinati di tempo indipendentemente dal resto del codice che viene normalmente eseguito all'interno del main o del while infinito dentro il quale quasi sempre si trovano i firmware per i microcontrollori.

Il nostro primo appuntamento con le STM Discovery board finisce qui, abbiamo visto le IDE di sviluppo per questo programma e un semplice esempio sulla gestione dei LED ed introdotto

RUTRONIK SMART

Committed to excellence

Sensors
 Wireless
 Microcontrollers
 Power Management

Discover the Internet of Things

Design SMART Products

RUTRONIK **SMART** offers you a new range of bundled hardware, software and services. RUTRONIK **SMART** brings together entire solutions to build applications for:

■ Healthcare

■ Comfort

■ Lifestyle

■ Safety

Learn more: www.rutronik.com/smart

Get in touch:
smart@rutronik.com or call +49 (0) 7231 801-1320

BOSCH
 DIODES
 intersil
 Kionix
 MICROCHIP

NORDIC
 OSRAM
 Panasonic
 REDPINE SIGNALS
 RENESAS

ROHM
 LAPIS Semiconductor
 ST
 Telit
 VISHAY
 YAGEO

RUTRONIK
ELECTRONICS WORLDWIDE

www.rutronik.com



la prima periferica un po' più complessa, ovvero il timer.

Qualcuno potrebbe obiettare che rispetto ad un Arduino, il quale offre un IDE semplice con la scrittura sequenziale del codice, le Discovery o quanto meno i tool di sviluppo che abbiamo visto siano molto meno intuitivi. Questo sicuramente è vero nel caso in cui vogliamo controllare dei led o fare qualcosa di molto semplice, ma nel caso in cui vogliamo realizzare opere più complesse, come ad esempio far lampeggiare i led a frequenze fisse di tempo, indipendentemente dal carico di lavoro sul controllore, o se necessitiamo di un hardware più potente e un numero maggiore di risorse (come pin, ADC, timer, ...) alla lunga la potenza di calcolo delle schede STM e il feedback visuale offerto da STM32CubeMX vincono.

Nel prossimo episodio vedremo gli output analogici tramite i segnali PWM e gli input sia digitali che analogici, ovvero l'ADC, chiudendo con paio di progetti un po' più pratici, ovvero un crepuscolare, un led che varia l'intensità luminosa al variare della luce ambientale e un semplice controllo di velocità per un motore in dc.

Questa serie di articoli è scritta a 4 mani con Salvo Armentia ([linkedin](#)) che colgo l'occasione di ringraziare per la dedizione e la passione da sempre mostrata.

L'autore è a disposizione nei commenti per eventuali approfondimenti sul tema dell'Articolo. Di seguito il link per accedere direttamente all'articolo sul Blog e partecipare alla discussione:
<http://it.emcelettronica.com/stm32-discovery-introduzione-ed-ide-di-sviluppo>



DesignCon: where the chip meets the board



di **Mario Rotigni**
29 Settembre 2015

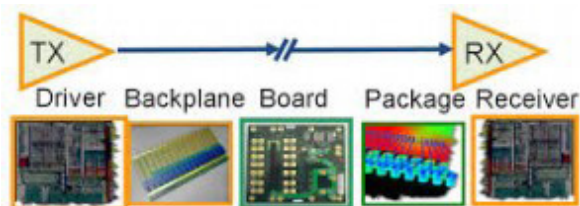
In questo articolo parleremo di un'attività intensa e poco conosciuta, non per questo meno importante, svolta da molti tecnici e ricercatori dietro le quinte, parallelamente alle attività quotidiane di sviluppo dei prodotti. Si tratta sostanzialmente di un aspetto della ricerca applicata ai vari settori della tecnologia, ricerca svolta all'interno dei laboratori di sviluppo di aziende grandi e piccole, da consulenti specializzati in aree di nicchia, scuole ed università, produttori di attrezzature di collaudo e software di progettazione (CAD/CAE).

I know how che ne deriva è evidentemente uno degli ingredienti della competizione tra aziende e quindi naturalmente soggetto a restrizioni di divulgazione per ragioni di confidenzialità. Facendo salva proprietà intellettuale e confidenzialità, vi sono oggi molte occasioni per condividere informazioni, ricerche, risultati in varie forme, tra gli esperti di ogni settore.

Come ben sappiamo, oggi i principali strumenti di condivisione sono riviste specializzate, siti Web, blog, conferenze tematiche. In particolare ci occuperemo qui di una di queste conferenze periodiche particolarmente interessante, **Design Conference** meglio nota come DesignCon, che si tiene in California all'inizio di ogni anno da oltre venti anni.

L'intrigante sottotitolo del sito Web dedicato alla

conferenza ne definisce l'ambizioso obiettivo. **"Dove i chip incontrano le schede", riassume lo spirito che nei 25 anni di vita ha contrassegnato DesignCon.** Essa è stata tra gli enti principali che hanno introdotto e definito la disciplina che ora va sotto il nome di Signal Integrity (vedi rif. 1) e si è poi espansa a coprire Power Integrity (vedi rif. 2), con incursioni sempre più frequenti nel mondo della compatibilità elettromagnetica su scheda e su chip, fino ad occuparsi recentemente delle nuove tecnologie di interconnessione quali la fotonica.



What could possibly go wrong?



GENESI DI UNA CONFERENZA DIVENTA PUNTO DI RIFERIMENTO

Venticinque anni fa Signal e Power Integrity iniziavano a presentarsi come problematiche ma erano lungi dall'essere definite ed articolate come ai giorni nostri. DesignCon nacque sostanzialmente come evoluzione di un simposio organizzato ogni anno da Hewlett Packard. Il simposio aveva due argomenti principali, progettazione di sistemi digitali ad alta velocità e progettazione di sistemi di comunicazione digitale.

Tra i titoli delle presentazioni svolte nel 1995 troviamo ad esempio uno studio sulla fattibilità del raddoppio della frequenza di clock di un sistema di memoria per processori 486, da 50 a 100 MHz. Troviamo anche un articolo di progettisti Apollo (vedi riferimento 3) circa glitches rumore e malfunzionamenti intermittenti, sottotitolato "the art of noise budgeting".

Un'altra presentazione si occupava di tecniche di progettazione per il controllo delle emissioni elettromagnetiche su circuito stampato, sottolineando le differenze tra Differential Mode e Common Mode, analizzando le differenti performance di stripline e microstrip.

Possiamo notare come negli eventi di vent'anni fa siano già presenti argomenti di assoluta attualità, quali i guasti intermittenti croce di ogni application e validation engineer o il controllo di EMC, trattati a frequenze operative che oggi potremmo trovare commoventi. In realtà, pur nella differenza dell'ordine di grandezza dei parametri, possiamo riconoscere la spinta a spingere i prodotti esistenti al limite massimo delle loro

performance, criterio familiare tutti noi.

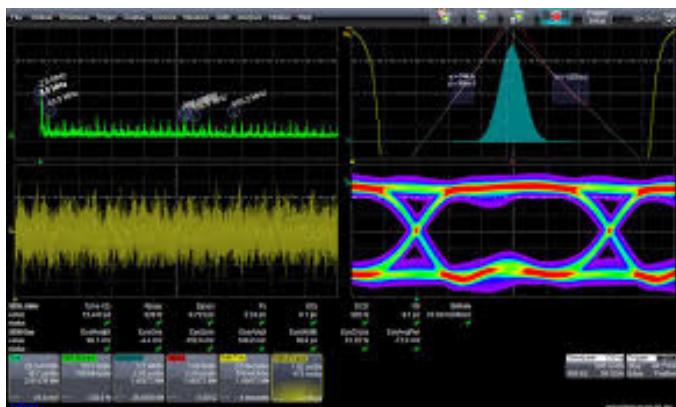
Caratteristica fondamentale della conferenza sin dal suo inizio, è stato un bilanciamento tra tutorial, presentazione di soluzioni adottate, tecniche di simulazione, tecniche sperimentali per la verifica del laboratorio di quanto previsto dalla teoria e dalle simulazioni. Questo interessante mix caratterizza ancora oggi e costituisce uno degli aspetti più interessanti di DesignCon.

Hewlett Packard si è divisa anni fa in due compagnie indipendenti, una delle quali si è concentrata sullo sviluppo e commercializzazione di computer, la seconda denominata Agilent, ha invece ereditato e sviluppato l'originale missione di sviluppo e produzione di strumentazione cui si è aggiunto nel tempo il software di simulazione. Molto recentemente Agilent si è ulteriormente divisa per creare due entità economiche indipendenti, una concentrata sulla strumentazione biomedica ed ambientale, che ha conservato il nome. La seconda azienda continua invece la tradizione di Hewlett e Packard nella strumentazione elettronica, prendendo il nome di Keysight.

È interessante notare che mentre DesignCon si è sviluppata autonomamente, da qualche anno è organizzata e gestita da UBM (vedi rif. 4), l'evento madre, progettazione e caratterizzazione di circuiti e sistemi ad alta velocità, esiste tuttora, organizzato ogni anno in varie parti del mondo da Keysight. Naturalmente, una parte significativa dell'evento è dedicata alla presentazione delle prestazioni e capacità della strumentazione e del software Keysight, ma possiamo ritrovare lo stesso spirito teoria - simulazione - mi-



sura della conferenza figlia, rendendo l'evento sempre interessante.



COME NASCE UN'EDIZIONE DELLA CONFERENZA

Torniamo ad occuparci di DesignCon, vedendo come la conferenza viene preparata e organizzata e dando un'occhiata più dettagliata al suo contenuto. La conferenza vede la presentazione di un vasto numero di interventi su argomenti specifici o tutorial mirati.

Ogni presentazione è basata su una serie di lucidi (slides) ed accompagnata da un articolo (paper) che sviluppa l'argomento in modo più ampio di quanto sia possibile trattare durante i 40 minuti assegnati a ciascun oratore. Sia le slides che i papers vengono resi disponibili ai partecipanti sotto forma di Conference Proceedings (atti della conferenza, normalmente su CD-ROM o DVD).

Parecchi mesi prima dell'inizio della conferenza la società organizzatrice, UBM in questo caso, emette una cosiddetta Call for Abstract. Chi intende proporre una presentazione è invitato a fornire un esaustivo e dettagliato estratto di una lunghezza massima predefinita, sul quale si ot-

terrà o meno una accettazione, in funzione della pertinenza, originalità, qualità ed interesse della proposta. La selezione è fatta da un comitato tecnico appositamente costituito.

La richiesta di proposte per DesignCon 2016 si è chiusa il 30 giugno di quest'anno, il processo di selezione si concluderà entro fine settembre. La conferenza 2016 inizierà il 19 gennaio, a Santa Clara, California e si concluderà il 21. Varie settimane prima dell'evento gli autori devono fornire le versioni definitive sia delle slides che dei papers, in modo da poter preparare il materiale per i partecipanti.

Le procedure di selezione dei contenuti per altre conferenze, ad esempio organizzate dai comitati della associazione internazionale degli ingegneri elettrotecnici ed elettronici (IEEE) o dai vari comitati degli enti normatori (ANSI, IEC ...) sono leggermente diverse. Queste prevedono infatti che la selezione venga fatta sulla versione preliminare dei papers e non sugli abstract. In ogni caso, tipicamente il documento da rivedere è inviato a più esperti del settore, ciascuno dei quali in modo indipendente formula il suo giudizio lasciando al comitato tecnico il compito di tirare le somme e proporre o bocciare la presentazione.

Questo meccanismo di revisione introduce un efficace filtro sul materiale delle conferenze, contribuendo alla qualità e rilevanza tecnico scientifica dell'informazione scambiata e della documentazione prodotta.

DesignCon, come altre conferenze simili, si compone di una serie di sessioni di presentazioni organizzate per argomenti omogenei. Le



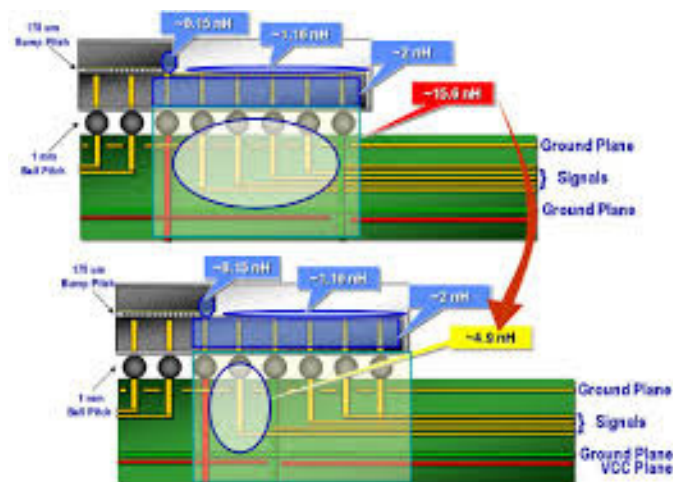
sessioni sono in parte contemporanee, in modo da condensare in modo ragionevole la durata complessiva della conferenza stessa e mettere ai partecipanti di focalizzarsi su quanto loro interessa maggiormente.

Oltre alle presentazioni orali vi sono normalmente tutorials che possono durare varie ore. La conferenza inoltre è accompagnata da una mostra esibizione in cui produttori di strumentazione o software CAD possono presentare le loro soluzioni.

La conferenza ha un forte taglio didattico, mirando a scambiare informazioni sulle tecnologie di punta, ma anche su aspetti fondamentali, incoraggiando il dialogo ed il contatto diretto tra tecnici direttamente coinvolti nella progettazione di circuiti sistemi ad alta velocità.

Tutti sappiamo per esperienza diretta che l'interazione diretta tra gli esseri umani è malgrado tutti gli strumenti informatici e social networks, il modo più efficace e veloce per condividere, sviluppare, acquisire idee nel proprio settore di attività specialistica.

I partecipanti a DesignCon, come altre conferenze simili, provengono dall'Università, dai laboratori e centri di ricerca di grandi aziende, ma anche da piccole e medie realtà, molte delle quali colgono l'occasione per farsi conoscere e presentare i servizi che sono in grado di offrire. È questo il caso di varie aziende specializzate nella fornitura di servizi di simulazione e verifica interfacce o sistemi.



LA PROSSIMA EDIZIONE

DesignCon 2016 è organizzato in 16 sessioni separate, le più interessanti tra le quali sono brevemente descritte nel seguito. La prima sessione non può che essere dedicata ai chips, raccogliendo tutto quanto relativo a signal e power integrity, interconnessioni, interfacce, tecniche di riduzione di rumore e jitter a livello di circuito integrato. La distribuzione del clock in System on chip e le strategie di reset, progettazione per bassa potenza assorbita, alimentazione multipla, trovano casa in questa sessione.

Accanto a problematiche storiche, quali Simultaneous Switching Noise, trovano posto anche tecnologie avanzate come le interconnessioni tra diversi chip, realizzati con o senza interconnessioni di silicio, utilizzate per creare System in Package (SiP), combinazione di chip prodotti ciascuno con la tecnologia più efficace, assemblati in un solo contenitore saldabile sulla scheda applicativa.

Le problematiche tipiche delle simulazioni miste analogico-digitali, sia circuitali che comporta-



mentali, sono oggetto di una sessione dedicata, così come la co-simulazione di chip package e board, oggi necessaria per garantire performance a livello di sistema.

L'accurata caratterizzazione dei materiali utilizzati per la produzione dei PCB e le tecnologie di produzione sono oggetti di un'altra sessione. L'importanza dell'argomento si può ben comprendere considerando l'exasperazione delle performance richiesta per trasmettere segnali su backplane e tra daughter card a velocità dell'ordine dei 40 Gbps, come richiesto dalle più avanzate interconnessioni in sviluppo. La spinta all'utilizzo di materiali standard e a basso costo, quali il classico dielettrico FR4, richiede un'accurata valutazione dei parametri parassiti introdotti da un isolante che non può più essere considerato ideale, combinata con la garanzia di una buona riproducibilità nella produzione in serie.

Nelle passate edizioni di DesignCon sono state introdotte metodologie di misura dell'impedenza delle reti di distribuzione dell'alimentazione su circuito stampato ora assunte come riferimento nell'industria (vedi rif.5). La sessione metodologie di collaudo e misura continua questa tradizione, includendo in ogni edizione informazioni sulle misure e l'uso dei parametri S ed il trattamento delle interconnessioni nelle misure ad alta frequenza (de-embedding delle misure).

Non mancano poi sessioni dedicate a tecnologie emergenti, rappresentate nella conferenza del prossimo anno dalla fotonica. L'uso della luce per la trasmissione di dati ad alta velocità tra circuiti integrati e sistemi, con tutte le proble-

matiche associate, è la proposta per superare i limiti in velocità dei collegamenti fisici attuali.

I Conference Proceedings sono disponibili qualche tempo dopo la conferenza anche per l'acquisto da parte di chi non abbia partecipato all'evento (vedi rif. 6). Si tratta di una vera miniera di informazioni, destinata a mantenere validità per lungo tempo. Alcuni documenti sono inoltre messi a disposizione gratuitamente ogni anno.

Tra i documenti disponibili per l'edizione della conferenza 2015 troviamo l'introduzione ai transistori FET all'arseniuro di gallio ed i loro ulteriori sviluppi recenti in applicazioni ad alta frequenza di commutazione.

Cadence ha reso disponibile una relazione sull'estensione delle regole di verifica di un circuito stampato in modo da tener conto delle esigenze delle simulazioni signal e power integrity, aggiunte alle classiche verifiche delle regole elettriche e meccaniche (isolamenti eccetera).

Keysight ha invece messo a disposizione la presentazione sull'analisi del protocollo PCIe, applicato ai dischi allo stato solido (SSD).

Certamente di questi tempi un viaggio intercontinentale è cosa impegnativa, ma se vi trovate in Silicon Valley nella seconda metà di Gennaio, non dimenticate di considerare la partecipazione a DesignCon.

PER APPROFONDIRE:

Riferimento 1: Signal Integrity, cos'è e quando preoccuparsene, Firmware n. 62, Marzo 2011

TIPS'N TRICKS



Riferimento 2: Power Integrity, Firmware nn 57 e 58 Ottobre, Novembre 2010

Riferimento 3: Apollo fu uno dei primi produttori di computer dedicati ad attività CAD (workstation). Fu acquistata nel 1989 Hewlett Packard.

Riferimento 4: UBM sito web: <http://tech.ubm.com/>

Riferimento 5: vedi ad esempio, "Accuracy Improvements of PDN Impedance Measurements in the Low to Middle Frequency Range", Desi-

gnCon 2010, disponibile qui <http://www.electrical-integrity.com/>

Riferimento 6: Acquisto proceeding DesignCon al sito web: <https://designcon.tech.ubm.com/all/Proceedings>

Riferimento 7: sito web dedicato a DesignCon: www.designcon.com

L'autore è a disposizione nei commenti per eventuali approfondimenti sul tema dell'Articolo.
Di seguito il link per accedere direttamente all'articolo sul Blog e partecipare alla discussione:
<http://it.emcelettronica.com/designcon-where-the-chip-meets-the-board>

Distributore Autorizzato



MOUSER
ELECTRONICS

NUOVO
INSTOCK

OSRAM
Opto Semiconductors



DURIS S 8 LED

Più informazioni