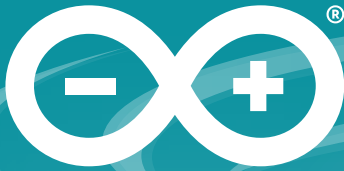


Edizione speciale
a cura di



ARDUINO

Edizione Bonus
Declassificata

Home Automation

Connettività Semplicata

p. 4 Retro-Gaming
con Arduino

p. 14 Un Controller
per Spotify

Dal Prototipo alla Produzione

Articoli per
professionisti,
maker e studenti!



Corso Introduttivo al
Mondo di Arduino

p. 10



Portare l'Elemento Umano
Nella Produzione

p. 24



Arte Floreale dai Fili a
Memoria di Forma

p. 32

From Arduino and Elektor with Engineering Love

OUT NOW

A unique Elektor Magazine edition curated by guest editor Arduino!

DIY electronics projects, engineering insights, and more from Arduino and Elektor engineers

Packed with projects and tutorials

Dive into hot topics such as MicroPython, TinyML, and home automation with Arduino

Get to know Arduino: Insights from Fabio, Massimo and David

Get Started with the Portenta x8

Get it now! From your favorite newsstand or buy in the Elektor web stores!

Direct links in articles give you easy access to Arduino products and solutions



More information

www.elektor.com/arduino-magazine



Elektor Magazine, Edizione Italiana

Supplemento Bonus all'edizione di Elektor Mag di Dicembre 2022 curata da Arduino.

www.elektor.com
www.elektormagazine.com

Elektor International Media

Sede Principale:

Elektor International Media b.v.
PO Box 11
6114 JG Susteren
Paesi Bassi
Telefono: (+31) 46 4389444

Abbonamenti:

E-mail: service@elektor.com
www.elektor.com/memberships

Pubblicità & Sponsorizzazioni:

Raoul Morreau
Telefono: +31 (0)6 4403 9907
E-mail: raoul.morreau@elektor.com

www.elektor.com/advertising

Tariffe pubblicitarie e termini di servizio disponibili a richiesta.

Nelle sue pubblicazioni Elektor utilizza solo contenuti originali (testi e immagini) o con l'autorizzazione dell'autore. Prima della pubblicazione, i contenuti forniti da terzi vengono sempre controllati per verificarne il copyright. Se il titolare dei diritti d'autore non è noto, facciamo il possibile per rintracciarlo e compensarlo a prezzi di mercato. Sfortunatamente, non sempre è possibile individuarlo. Se vi trovate in questa situazione e siete "l'ignoto detentore del copyright", oppure lo conoscete, vi preghiamo di contattare editor@elektor.com.

Avviso di Copyright

I circuiti descritti in questa rivista sono destinati esclusivamente all'uso domestico e didattico. Tutti i disegni, le fotografie, i layout dei circuiti stampati, i circuiti integrati programmati, i dischi, i CD-ROM, i DVD, i supporti software e i testi degli articoli pubblicati nei nostri libri e nelle nostre riviste (ad eccezione delle pubblicità di terzi) sono protetti da copyright di Elektor International Media b.v. e non possono essere riprodotti o trasmessi in nessuna forma o con nessun mezzo, comprese le fotocopie, le scansioni e le registrazioni, in tutto o in parte, senza la preventiva autorizzazione scritta dell'editore. Tale autorizzazione scritta deve essere ottenuta anche prima che qualsiasi parte di questa pubblicazione venga memorizzata in un sistema di recupero di qualsiasi natura. I circuiti, i dispositivi, i componenti ecc. descritti in questa rivista possono essere protetti da brevetti. L'Editore non si assume alcuna responsabilità per la mancata identificazione di tali brevetti o altre protezioni. L'editore declina ogni responsabilità per il funzionamento sicuro e corretto dei progetti assemblati dai lettori e basati su schemi, descrizioni o informazioni pubblicate nella rivista Elektor o in relazione ad essa.

Edizione Bonus Declassificata

La collaborazione creativa tra Elektor e Arduino non si è conclusa con il numero di Elektor Mag pubblicato all'inizio di Dicembre 2022. Abbiamo altri progetti, approfondimenti tecnici e articoli informativi che vi ispireranno per i prossimi mesi. Nel giro di quattro settimane, declassificheremo i contenuti di questa edizione per mettervi a disposizione, all'inizio di gennaio 2023, il supplemento bonus completo della rivista. Un modo fantastico per inaugurare l'anno nuovo! Sia che siate ingegneri professionisti che lavorano a un nuovo prodotto industriale, sia che siate amanti del fai-da-te alla ricerca di un progetto divertente basato su Arduino

per il vostro fine settimana, troverete questo supplemento di Elektor Mag ricco di informazioni e di stimoli. Vi proponiamo articoli su un'ampia gamma di argomenti e progetti legati ad Arduino, tra cui il retro gaming con Arduino, una training board Arduino di Elektor e un controller portatile per Spotify basato su Arduino.

Leggendo i progetti e gli articoli di questa rivista, sentitevi liberi di condividere i vostri pensieri con noi su elektormagazine.com, Arduino.cc e sui social media. Saremo lieti di ricevere il vostro feedback. Buon divertimento!

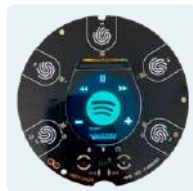
C. J. Abate (Content Director, Elektor)

IN QUESTO NUMERO



- ◀ **4 Girare Doom su un Portenta**
Retro-Gaming con Arduino

- 10 Corso Introduttivo al Mondo di Arduino**
Scheda di Sviluppo per Arduino Nano



- ◀ **14 Un Controller per Spotify**
L'Oplà IoT Kit è (Quasi) Tutto Ciò Che Vi Serve

- 20 Realizzare, Distribuire e Mantenere Applicazioni Scalabili e Sicure**
Con Arduino Portenta X8 Dotato di Processore Applicativo i.MX 8M Mini di NXP e di Secure Element EdgeLock® SE050



- ◀ **24 Portare l'Elemento Umano Nella Produzione**
Incontro con Daria Baradel, Responsabile di Produzione di Arduino

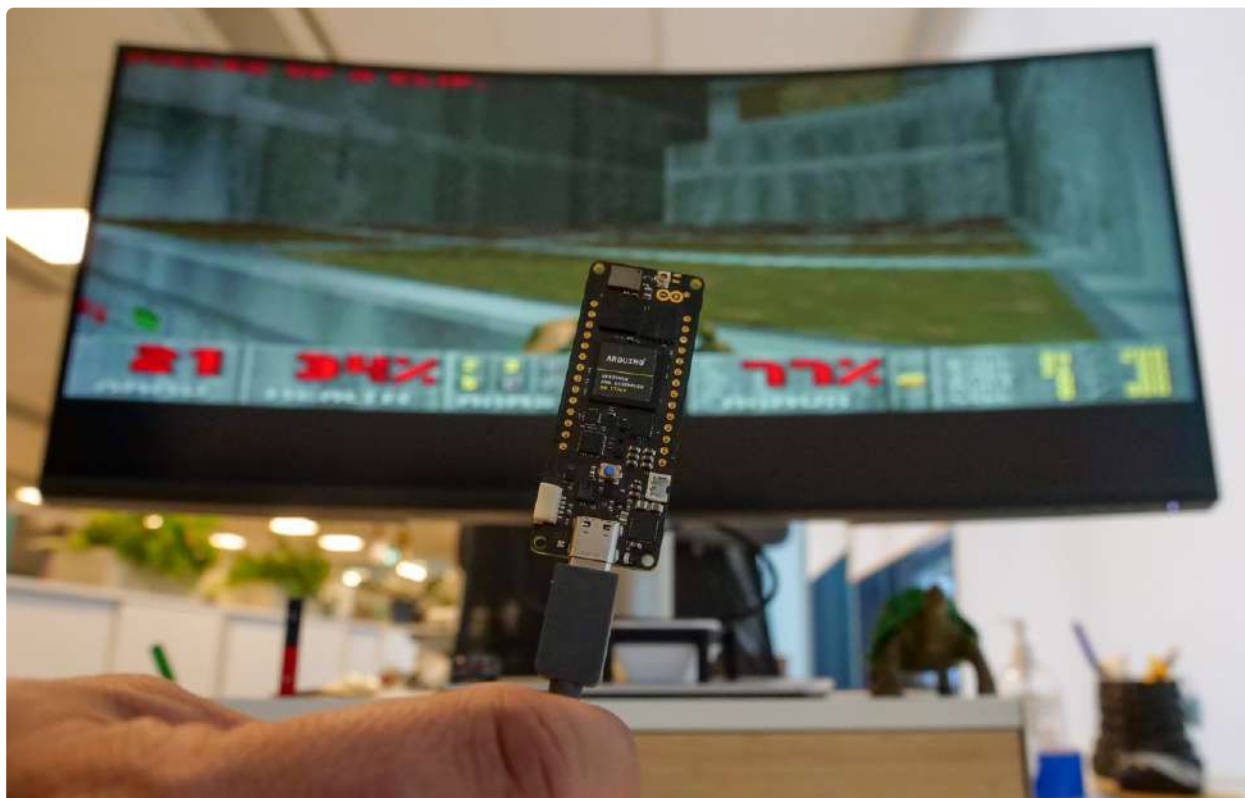


- 28 Schede di Sviluppo**
Passato, Presente e Futuro

- ◀ **32 Arte Floreale dai Fili a Memoria di Forma**
Sculpture Cinetiche che Comunicano col Suono



- ◀ **38 Catalogo dei Prodotti**
Afferra il Nuovo Hardware Arduino



Potete far girare Doom su un Portenta!

Girare Doom su un Portenta

Retro Gaming con Arduino

Di David Cuartielles (Arduino)

Potete eseguire Doom su un Arduino Portenta H7. Siete curiosi di sapere come fare? Volete sapere perché gli ingegneri di Arduino ci hanno fatto girare il gioco? Risponde Martino Facchin, responsabile del team firmware di Arduino.

Doom è probabilmente il gioco più venduto della storia, con oltre 3,5 milioni di copie distribuite. Con un prezzo al dettaglio di 50 dollari (USD), gli sviluppatori di Doom, che formavano una piccola società chiamata id Software, sono diventati milionari da un giorno all'altro. Quando Doom fu pubblicato nel 1993, id Software era già uscita con un altro gioco molto conosciuto da circa un anno, Wolfenstein 3D. Doom è uno sparatutto in prima persona, in cui il giocatore deve navigare in uno spazio tridimensionale e affrontare diversi nemici

utilizzando armi e munizioni diverse, che possono essere trovate in tutto il campo di battaglia del gioco. Doom è stato portato su tutti i sistemi operativi, ma gira anche in bare metal su più sistemi. Il codice sorgente è aperto, attualmente con licenza GPL. Anche se non è banale compilarlo, abbiamo visto versioni di Doom girare su computer molto piccoli e anche all'interno di altri programmi. A un certo punto, Excel 95 di Microsoft Office ha reso un omaggio a Doom sotto forma di un easter egg giocabile che si trova nella pagina dei crediti del software.

Il gioco è diventato un modo per verificare le prestazioni dei piccoli computer e per mostrare la propria abilità di hacker. Qualche mese fa, in occasione del DEF CON 22 di Las Vegas, l'hacker noto come @sickcodes su Twitter e GitHub ha presentato una demo di Doom su un trattore John Deere modificato per includere una grafica legata all'agricoltura. Arduino non fa eccezione. E quando è uscito il primo prototipo della Portenta H7 - la nostra scheda più potente fino a quel momento nel 2018 - abbiamo usato Doom per testare le capacità tecniche della scheda. Di recente ho invitato Martino Facchin, responsabile del team firmware di Arduino, a raccontarci di questa storia e di come è stata realizzata.



David Cuartielles: Parliamo dell'Arduino Portenta H7 che esegue Doom. Ho preparato un piccolo riassunto della storia di Doom, di come i ragazzi della id Software hanno realizzato il gioco, di come ha venduto alla grande e di come i realizzatori sono diventati milionari.

Martino Facchin: E poi hanno pubblicato il codice sorgente, la cosa più importante.

Cuartielles: Esattamente! Hanno pubblicato il codice sorgente, ma con quale licenza lo hanno pubblicato?

Facchin: Devo controllare, ma credo che sia una licenza compatibile con la GPL, la Doom Source Code Licence. La cosa importante è che solo il motore è open source. Le risorse sono chiuse e, di fatto, è possibile giocare solo alla versione shareware di Doom, quindi non è possibile giocare al gioco completo a meno che non lo si acquisti.

Una Nota da David

Sono andato a controllare perché volevo esserne sicuro, e nel 1997 id Software ha pubblicato Doom sotto la suddetta licenza, che apriva il sorgente a scopo didattico. Dopo un incidente occorso ai manutentori di glDoom, che a causa della clausola di non distribuzione della licenza di Doom lasciò il mondo senza una copia del relativo port OpenGL, id Software accettò di modificare la licenza in GPL.

Cuartielles: La cosa bella di questo gioco è che la gente creava i propri mod. Ricordo una versione di Wolfenstein 3D con i personaggi di Star Wars.

Facchin: Non me lo ricordo.

Cuartielles: Sono così vecchio. Me lo ricordo per entrambi.

Facchin: Beh, ricordo di aver giocato a Wolfenstein 3D quando ero bambino. Ma non avevamo una connessione a Internet, quindi non potevamo ottenere tutte queste chicche dalla comunità dei modder.

Cuartielles: Prima di continuare, permettimi di rivolgerti una domanda. Potresti presentarti?

Facchin: [ride] Martino Facchin, ingegnere firmware di Arduino.

Cuartielles: Qual è il tuo ruolo in Arduino?

Facchin: Sono il responsabile del firmware, l'uomo principale quando si ha bisogno di aiuto per il firmware. Adesso ho un meraviglioso team di colleghi che mi aiutano in questo, perché ho iniziato questo team da solo. Il gruppo continua a crescere. Cerchiamo di far crescere anche la comunità con noi, rendendo tutti più consapevoli di ciò che stiamo facendo.

Cuartielles: Oltre a far funzionare Doom su Portenta (di cui parleremo più avanti), qual è la cosa che hai realizzato con Arduino che ti rende più orgoglioso?

Facchin: Direi che è il framework PluggableUSB. Quando sono arrivato ad Arduino, dopo sei mesi, avevamo un grosso problema di persone che volevano aggiungere molteplici funzionalità alla porta USB di Arduino Due e dell'ormai imminente Arduino Zero. Ogni volta che si collegava un Arduino Leonardo a un computer, venivano visualizzati i driver della tastiera, del mouse e così via, anche se non li si utilizzava. Abbiamo dovuto costruire al volo un descrittore USB che aiutasse gli utenti a vedere solo le cose che volevano usare in quel momento. Allo stesso tempo, abbiamo anche permesso che si verificassero altre cose, come il MIDI USB e simili. Per me questo è stato un grande sviluppo. All'epoca ero piuttosto giovane e ho dovuto interagire con la comunità con l'aiuto di Matthijs Kooijman (per saperne di più sul suo lavoro su www.stdder.nl) e Paul Stoffregen (creatore di Teensy) per trovare la migliore strategia possibile. E ha funzionato. Di tanto in tanto, le persone vengono a dire: "Ho usato la libreria MIDI per questo o quello", oppure c'è persino uno sviluppatore che ora sta realizzando tutti i layout per tutte le tastiere internazionali costruite sulle fondamenta di quel codice. Ne sono molto orgoglioso.

Cuartielles: Hai detto che all'epoca eri "piuttosto giovane". Da quanto tempo lavori per Arduino?

Facchin: Da sei anni e mezzo. Lavoro nella sede di Torino.

Cuartielles: È un bel po' di tempo. E quante persone sono coinvolte nel team Firmware?

Facchin: Sei persone. Potrebbero sembrare molte, ma nel team Firmware ci occupiamo della manutenzione di tutti i prodotti, svolgendo anche altre attività come la certificazione. D'altra parte, abbiamo separato il Firmware dal Tooling, che è gestito da un team diverso, dedicato all'Arduino CLI e ad altre parti del software di fascia alta.

Cuartielles: E tutti gli sviluppatori lavorano su Github, giusto?

Facchin: Sì, alla fine del processo di sviluppo tutto è open source.



La scelta più ovvia è stata quella di fare il porting di Doom e cercare di strumentare tutte le caratteristiche necessarie per farlo girare.



Cuartielles: Ottimo. Torniamo a parlare di Doom. Sappiamo che è stato un gioco di grande successo che ha venduto oltre 3,5 milioni di unità, rendendo gli sviluppatori ricchissimi. È stato il primo best-seller tra gli sparatutto in prima persona (FPS). Il codice era open source ed è stato portato su tutti i tipi di dispositivi.

Facchin: Telefoni, calcolatrici, ogni sistema operativo.

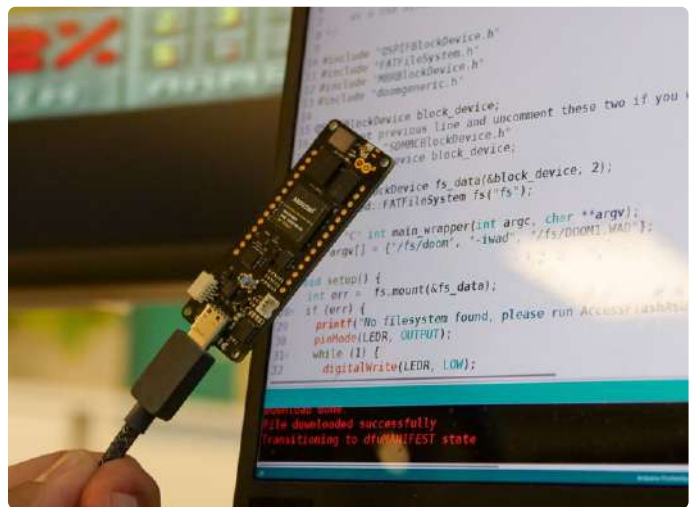
Cuartielles: And Arduino's Portenta H7, a dual core processor board aimed at industrial environments. Who had the idea of running Doom on the Portenta H7?

Cuartielles: E Portenta H7 di Arduino, una scheda con processore dual core destinata agli ambienti industriali. Chi ha avuto l'idea di far girare Doom su Portenta H7?

Facchin: È stato più che altro perché avevamo appena ricevuto il primo prototipo di Portenta H7, una bella scheda con molti chip, e tutto era ancora da fare. Avevamo questo chip di Analogix, che di solito si trova in altri dispositivi, che converte i segnali MIPI in DisplayPort, consentendo di collegare monitor esterni. Non avevamo alcuna esperienza sui sottosistemi del chip che avrebbero dovuto gestire questo aspetto, e anche gli esempi esistenti non ci aiutavano. All'inizio siamo riusciti a visualizzare alcune caselle gialle sullo schermo, poi il logo di Arduino con alcuni artefatti, ma non era affatto perfetto. Non riuscivamo a capire perché le cose non funzionassero come previsto, quindi ho cercato qualcosa con immagini in movimento che potessi riconoscere facilmente. La scelta più ovvia è stata quella di fare il porting di Doom e cercare di strumentare tutte le caratteristiche necessarie per farlo funzionare. C'è voluto davvero poco per farlo funzionare su qualcosa che non avesse un sistema operativo. Non è proprio il fork di Doom a cui vorreste giocare, ma è molto facile da portare. Basta cambiare

sei o sette funzioni per adattarlo all'hardware e il gioco è fatto. Per cominciare, non avevamo il codice per gestire la RAM interna o la memoria esterna. Prima ho dovuto farle funzionare, poi ho avviato il simulatore. Il video non ha funzionato fin dall'inizio, ho dovuto preparare un framebuffer, fare qualche trucco e ho ottenuto un'uscita corretta su USB-C.

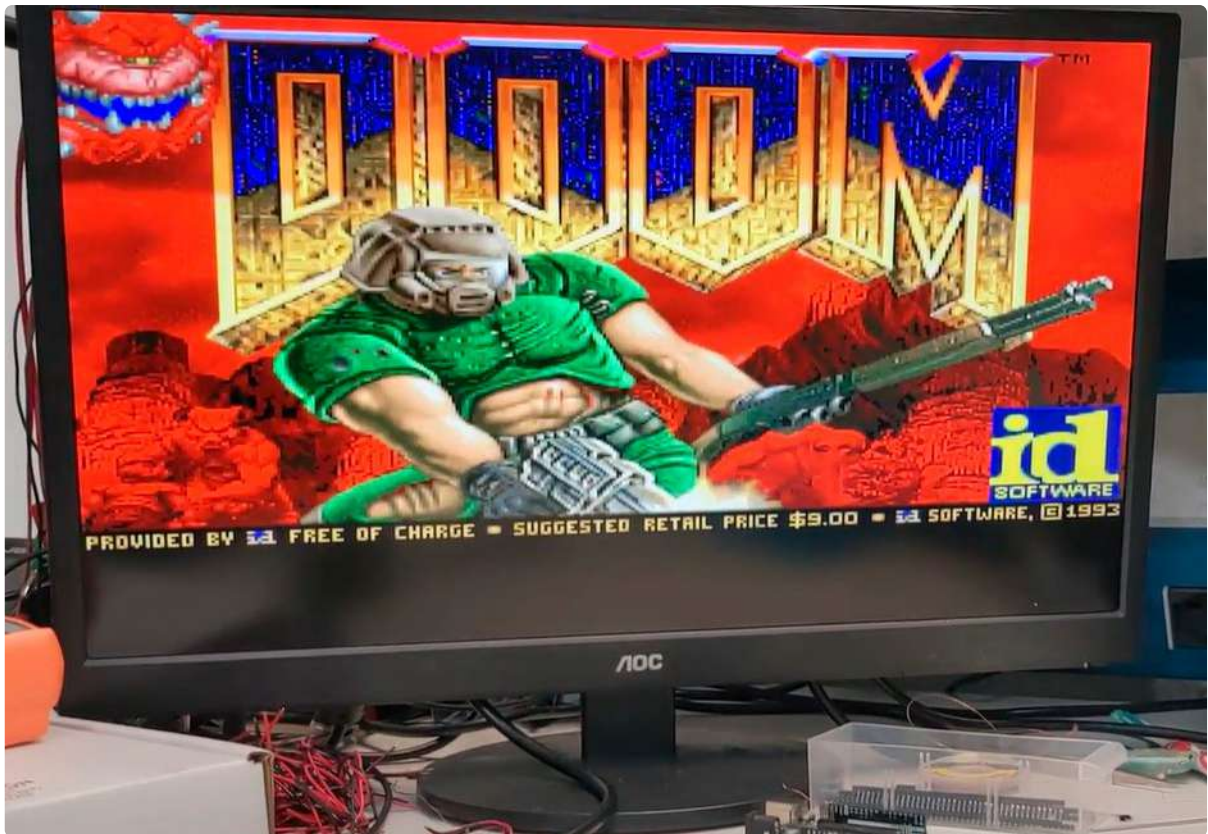
Cuartielles: Il Portenta H7 è un processore dual-core. Ha un Arm Cortex M4 e un M7 all'interno del processore. Qual era quello che eseguiva Doom?



Spetterà al lettore prendere in mano il progetto e aggiungere i controlli per rendere il gioco giocabile.

Come girare Doom sul vostro Portenta H7

1. Scaricate l'ultima versione dell'IDE Arduino. Si consiglia Arduino 2.0 o più recente.
2. Scaricate il core Portenta dal board manager.
3. Selezionate il core M7 per la vostra scheda Portenta H7. Tutti i passaggi seguenti devono essere eseguiti su questo core.
4. Assicuratevi che l'IDE abbia identificato la porta a cui è collegata la scheda.
5. In "Examples / Doom" è presente un esempio in cui è possibile vedere le istruzioni di base. Prima di installarlo, è necessario eseguire un paio di esempi sulla scheda.
6. [Facoltativo] Aggiornate il bootloader del Portenta H7 all'ultima versione con "Examples / STM32H747_System / STM32H747_manageBootloader".
7. Formattate la Flash esterna utilizzando l'esempio "Examples / STM32H747_System / QSPIFormat". Dopo l'installazione, dovrete aprire il terminale seriale e seguire le istruzioni in esso contenute.
8. Trasformate la scheda in un dispositivo di memoria di massa come se fosse un drive USB con "Examples / USB As Mass Storage / AccessFlashAsUSBDisk".
9. Aprite il monitor seriale e scegliete come procedere alla formattazione della scheda. Al termine, il computer dovrebbe registrare le due nuove unità esterne collegate.
10. Scaricate DOOM1.WAD dal sito DoomWiki all'indirizzo: <https://doomwiki.org/wiki/DOOM1.WAD> e copiatelo nella partizione più grande dell'unità virtuale Portenta.
11. Tornate all'esempio di Doom.ino visto al punto 5 e flashatelo sulla scheda, ricordando: sempre sul core M7. Se avete avuto problemi a vedere la porta di programmazione, fate doppio clic sul pulsante di reset del Portenta prima di caricarlo e assicuratevi che la porta seriale venga riconosciuta correttamente.
12. Scollegate il Portenta H7 dal computer e collegatelo a un hub USB-C come se fosse un computer portatile. L'hub dovrà essere collegato all'alimentazione esterna e a un cavo HDMI per inviare il segnale video al monitor del computer.



Doom è un classico!

Facchin: Oggi usiamo l'M7, ma all'epoca lo facevamo girare sull'M4, perché era molto più semplice dal punto di vista del programmatore embedded. È più simile a un tipico microcontrollore, senza alcuna caratteristica speciale. D'altra parte, l'M7 ha questa cache che bisogna controllare e invalidare al momento giusto quando si genera il video. La prima volta che ho provato a far girare qualcosa sull'M7 era molto veloce, ma anche completamente guasto e non riuscivo a vedere nulla sullo schermo. L'M4 era abbastanza veloce (25 fotogrammi al secondo) e i pixel erano perfetti.

Cuartielles: a 25 fps è molto più veloce del mio primo computer. Ma riassumiamo il tutto per il lettore: Doom funziona con il più lento dei due processori, su una scheda dotata di connettività Bluetooth e Wi-Fi. Il video viene inviato tramite USB-C. È possibile avere un hub, un mouse, una tastiera, qualsiasi cosa. Quali sono stati i controlli che avete implementato?

Facchin: Purtroppo il progetto per noi è morto lì, perché una volta visto che potevamo far girare il video, abbiamo iniziato a lavorare con LVGL, che è una libreria molto più utile per gli altri sviluppatori per costruire applicazioni su Portenta. LVGL è completamente integrata con l'hub USB, la tastiera e il mouse, in modo da poter costruire tutte le interfacce necessarie per il contesto professionale di cui molti utenti finali di Arduino hanno bisogno.

Cuartielles: Non sarebbe bello avere un PLC professionale basato su Portenta H7 in cui si possa giocare a Doom sul core Cortex M4 mentre si fa il lavoro serio sull'M7?

Facchin: Assolutamente sì!

Cuartielles: Grazie, Martino. È stato bello ascoltare la storia di Doom su Arduino Portenta H7. Condivideremo con la nostra comunità le istruzioni di base su come farlo funzionare. Spetterà al lettore prendere in mano il progetto e aggiungere i controlli per rendere il gioco giocabile. ◀

220542-05

Note sull'Autore

David Cuartielles è co-fondatore di Arduino. Detiene un PhD in Progettazione Interattiva e un MSc in Ingegneria delle Telecomunicazioni, insegna alla Malmö University.

Domande o Commenti?

Avete domande o commenti relativi a quest'articolo? Contattate il team Elektor a editor@elektor.com.



Prodotti Correlati

- ▶ **Arduino Portenta H7**
www.elektormagazine.com/arduino-portenta-h7

20%
di sconto
sul tuo primo anno
di abbonamento

Unisciti alla Elektor Community

Sottoscrivi una
membership!



- ✓ Accesso all'archivio web Elektor dal 1974!
- ✓ 8x Rivista Elektor (versione stampata)
- ✓ 8x Rivista Elektor (versione digitale)
- ✓ Sconto del 10% nel nostro web shop e offerte esclusive
- ✓ Accesso a più di 5000 file Gerber
- ✓ Spedizione gratuita in Europa



www.elektormagazine.com/gold-member

Usa il coupon::

ARDUINO20

Apriamo il Kit LCR METER di Elektor con David Cuartielles

Segnatevi la data: 26 gennaio 2023



Volete scoprire insieme a me il kit del misuratore LCR di Elektor? Guardate l'episodio del 26 gennaio 2023 (18:00 CET) di Elektor Lab Talk, in cui mi unirò agli ingegneri Elektor Mathias Claussen e Jens Nickel per discutere in merito a questo kit, oltre a rispondere alle vostre domande sulla tecnologia Arduino e su questa edizione di Elektor pubblicata a cura dei nostri ospiti. Non perdetevi il livestream. Fate le vostre domande!

220555-015

Elektor LabTalk

Guarda David dal vivo su Elektor Lab Talk il 26 gennaio 2023!



www.elektormagazine.com/labtalk-david

Corso Introduttivo al Mondo di Arduino

Scheda di Sviluppo per Arduino Nano

Di Wolfgang Trampert (Germania)

Elektor rimane fedele alla sua missione educativa: in questo articolo vi presentiamo una nuovissima scheda didattica con un Arduino Nano al cuore del sistema. Unitamente a un corso di formazione pratico e ben strutturato, rappresenta una piattaforma ideale per aggiornare le proprie competenze ed esplorare il mondo dei microcontrollori.



Nota dell'Editore

Al momento della pubblicazione, il libro associato a questo kit è disponibile solo in Tedesco. Nel prossimo futuro sono previste traduzioni. Una volta terminate, le versioni tradotte saranno disponibili nell'Elektor Store.

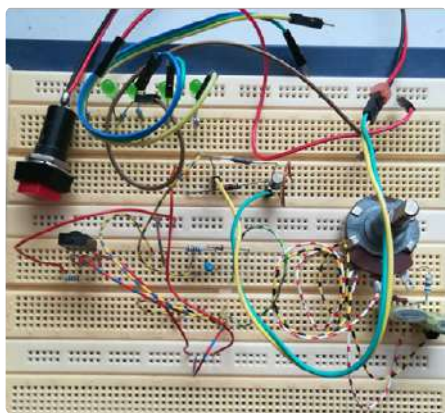


Figura 1: Tipico layout di circuito cablato per uno sketch Arduino.

Si potrebbe dire che la "filosofia Arduino" sia associata a un approccio di progettazione molto vicino all'hardware: nella maggior parte dei casi, tramite i GPIO del microcontrollore, il software o lo sketch Arduino accedono a componenti come interruttori, pulsanti, potenziometri, LED, display LC, segnalatori acustici piezoelettrici, transistor driver ecc. Altri tipi di dispositivi periferici o moduli elettronici come sensori, display o schede driver utilizzano varie interfacce seriali come SPI, I²C o bus a 1 filo per comunicare con il controllore. Per familiarizzare con il mondo dei microcontrollori e delle schede Arduino, è necessario costruire nuovi circuiti di prova e controllarli utilizzando una scheda Arduino.

Tuttavia, l'essenza di qualsiasi progetto basato sull'hardware Arduino è proprio nello

sviluppo del software (sketch). L'hardware è solo un mezzo per raggiungere un fine. Naturalmente è possibile collegare tutte le periferiche necessarie per un particolare progetto utilizzando una scheda di prototipazione (Figura 1). Questo approccio offre la massima libertà di installare le periferiche e collegare i segnali dove si vuole. Non sempre è una buona idea, soprattutto quando si è agli inizi. Le dimensioni del layout possono aumentare rapidamente e ci si ritrova con un groviglio di cavetti Dupont. Spesso si passa più tempo a fare il debug dell'hardware e a risolvere errori di cablaggio che a scrivere il codice. Questo non fa che aumentare il livello di frustrazione e la correzione di errori banali non necessariamente insegnerà qualcosa di utile.

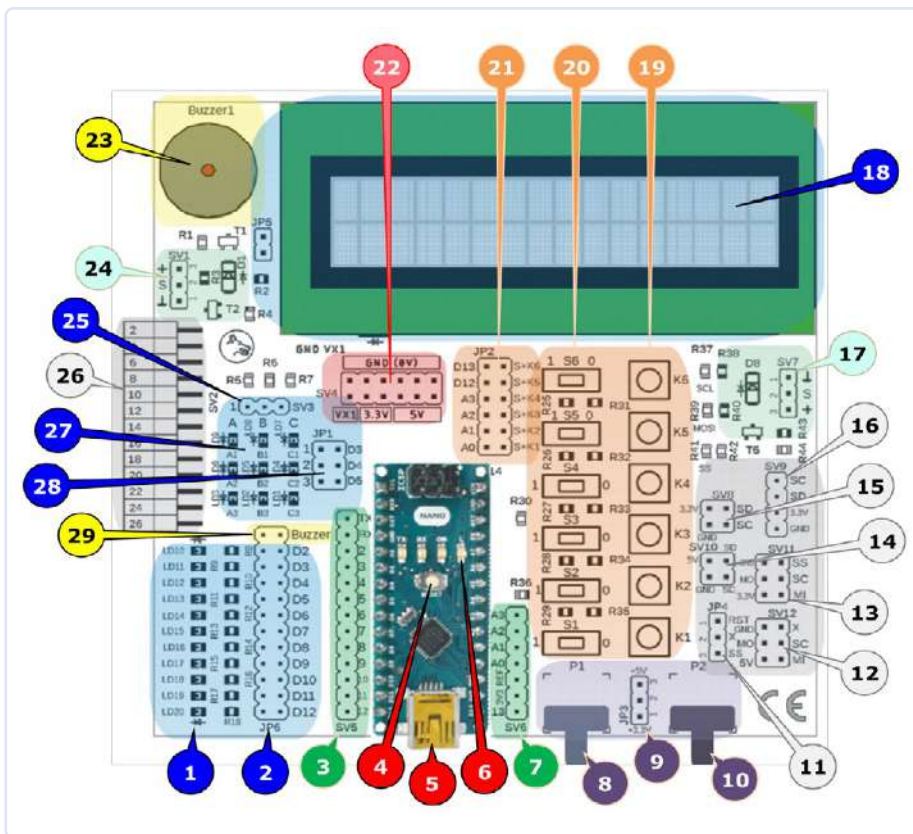


Figura 2: La training board MCCAB, Rev. 3.3.

La Training Board MCCAB

Per aggirare questi ostacoli abbiamo sviluppato la Elektor Arduino Training Board, nota anche come MCCAB Training Board (Figura 2), al cui centro troviamo collegata una scheda Arduino Nano. Accanto ad essa sono presenti molte delle periferiche di base di cui si ha generalmente bisogno per costruire un nuovo prototipo per innumerevoli applicazioni, come ad esempio una configurazione di laboratorio, circuiti di prova e sperimentali, progetti ed esercizi a supporto dei propri studi e della formazione e anche progetti hobbistici. I GPIO del microcontrollore sono tutti disponibili sui connettori a doppia fila di pin della Training Board MCCAB, il che le conferisce la massima flessibilità di implementazione. È possibile collegare periferiche aggiuntive o segnali esterni utilizzando i cavetti Dupont, a seconda delle necessità. Non dovrete preoccuparvi di un errato cablaggio delle periferiche sulla scheda e passerete meno tempo a rovistare nelle scatole dei ricambi per trovare quell'ultimo componente sfuggente che vi serve per completare un circuito. Anche i circuiti aggiuntivi costruiti su breadboard possono essere facilmente collegati, utilizzando i cavi Dupont, poiché tutti i GPIO del microcontrollore di Arduino Nano sono collegati ai due connettori SV5 e SV6 della

training board MCCAB (puntatori 3 e 7 nella Figura 2). Lungo il lato sinistro della scheda è inoltre visibile il connettore ad angolo retto a 26 pin a doppia fila SV2 (puntatore 26 nella Figura 2), dove è possibile collegare una scheda di espansione esterna. Questo connettore fornisce tutti i più importanti segnali GPIO dal microcontrollore. Le schede esterne che implementano funzioni come un tracciatore di curve di componenti elettronici, un alimentatore da laboratorio o un controllore semaforico possono essere collegate alla training board MCCAB ed essere controllate da essa. Le informazioni e i risultati di uno sketch in esecuzione possono essere scritti sull'LCD integrato da 2 x 16 caratteri (puntatore 18), che si interfaccia tramite il bus I²C della scheda. È presente anche una matrice di LED 3 x 3 (puntatore 27).

La training board MCCAB è alimentata da una tensione $V_{cc} = +5 V$. Di solito questa viene fornita tramite il cavo USB collegato al PC, necessario per creare e caricare lo sketch dell'esercizio sull'MCCAB. L'MCCAB può anche essere alimentata da un alimentatore esterno.

Nello schema della Training Board (Figura 2), tutti i componenti associati a una specifica funzione della scheda sono identificati con un colore di sfondo comune.

Controlli e indicatori della training board MCCAB

- 1 11 x LED (indicatori di stato degli ingressi/uscite da D2 a D12)
- 2 Connettore per il collegamento dei LED da LD10 a LD20 con i GPIO da D2 a D12
- 3 Ingressi e uscite del microcontrollore
- 4 Pulsante di *RESET*
- 5 Arduino NANO con connettore mini USB
- 6 LED *L*, collegato al GPIO D13
- 7 GPIO del microcontrollore
- 8 Potenziometro P1
- 9 Alimentazione per P1 e P2
- 10 Potenziometro P2
- 11 Segnale al pin X di SV12
- 12 SPI-Interface 5 V (The signal at pin X is selected by JP4)
- 13 Interfaccia SPI- 3,3 V
- 14 Interfaccia I²C- 5 V
- 15 Interfaccia I²C- 3,3 V
- 16 I²C-Interface 3.3 V
- 17 Switch output for external equipment
- 18 LCD 2 x 16 caratteri
- 19 6 x Pulsanti da K1 a K6
- 20 6 x Interruttori slider da S1 a S6
- 21 Pin di collegamento tra switch e GPIO del microcontrollore
- 22 Presa di alimentazione
- 23 Segnalatore acustico "Buzzer1"
- 24 Switch output for equipment
- 25 Connettore colonne matrice LED 3 x 3 LED
- 26 Connettore 2 x 13 pin per modulo esterno
- 27 Matrice LED 3 x 3 LED (rossa)
- 28 Connettore righe D3, D4 e D5 della matrice 3 x 3
- 29 Jumper di collegamento tra *Buzzer1* e GPIO D9

Table 1: Classi disponibili nella Libreria MCCAB_Lib.

Classe	Utilizzo
KeySwitch	Stato ("debounced") degli switch da S1 a S6 e pulsanti da K1 a K6
Matrix	Controllo della matrice LED 3 x 3 LED.
LED	Controllo On / off / lampeggio dei 12 LED da LD10 a LD20 e del LED
LedBlock	Uscita di un pattern di bit su tutti gli 11 LED (da LD10 a LD20)
Sound	Controllo di Buzzer1 e generatore di onde quadre.

La Libreria MCCAB_Lib per l'uso con le Training Board

Lo sviluppo del software prevede l'utilizzo dell'IDE Arduino per la scrittura del programma (o "Sketch", in gergo Arduino) che indica al microcontrollore come comportarsi. Successivamente lo sketch viene compilato e caricato sul microcontrollore di Arduino Nano sulla training board tramite un cavo mini USB. I GPIO del microcontrollore possono essere configurati come di consueto utilizzando la funzione Arduino `pinMode()` e il valore dei segnali da e verso i componenti della training board può essere letto o controllato utilizzando `digitalRead()`, `digitalWrite()`, `analogRead()` etc.

In ogni caso, è disponibile una libreria denominata *MCCAB_Lib* [1] che supporta lo sviluppatore fornendo comandi aggiuntivi per controllare le numerose periferiche hardware della training board MCCAB. La libreria può essere scaricata gratuitamente e integrata nel proprio sketch, rendendo molto più semplice la gestione delle periferiche presenti sulla scheda.

La libreria *MCCAB_Lib* contiene cinque classi per il controllo degli interruttori, dei LED e del segnalatore acustico della training board e può essere facilmente inclusa nello sketch dell'utente, a seconda delle necessità. La **Tabella 1** mostra un elenco delle classi disponibili.

Utilizzando questa libreria, l'utente non deve preoccuparsi di definire i periodi di tempo per il debouncing degli interruttori, di generare i segnali di controllo in multiplex per la matrice di LED 3 x 3 e di far lampeggiare i LED da LD10 a LD20 o L, né di generare i toni del segnalatore acustico. Le funzioni di libreria eseguono queste operazioni automaticamente in background nel flusso del programma, senza che l'utente se ne accorga.

Il **Listato 1** è un piccolo esempio di sketch per dimostrare l'uso della libreria *MCCAB_Lib*. Nella riga 15 dello sketch, viene dichiarata la variabile oggetto `Led` della Classe `LED` della libreria *MCCAB_Lib*. Il parametro `LED_PIN` passato nella dichiarazione della variabile oggetto `Led` è definito come una costante nella riga 13 che indica il pin a cui è collegato il LED. Questo pin viene automaticamente configurato come uscita durante la sua istanza. La variabile oggetto `Key` della classe `KeySwitch` della libreria *MCCAB_Lib* (dichiarata alla riga 22) durante l'esecuzione controlla (in background) lo stato dello switch in ingresso sul pin SK4, che le viene passato



Listato 1.

```
/*
 * Sketch which uses pushbutton K4 to toggle LED LD10 on and off using object variables in the
 * "KeySwitch" and "LED" classes in the MCCAB_Lib library.
 * To read the status of pushbutton K4 its necessary to insert a jumper to link position S+K4 (the switch
 * connection) with A3 (GPIO A3 of the microcontroller) on double header strip JP2 of the MCCAB.
 * Insert another jumper (in position D2 of the double header strip J6) to link LED LD10 with the
 * microcontroller GPIO of the MCCAB.
 */

11 #include <MCCAB_Lib.h> // bind the MCCAB_Lib Library to the Sketch
12
13 #define LED_PIN 2 // the LED is connected to pin D2
14
15 LED Led(LED_PIN); // Object-Variable
16
17 //function called by the object-variable "Key" when the switch is closed.
18 void switchTurnedOn() {
19     Led.toggle(); // toggle or flip the state of the LED
20 }
21
22 KeySwitch Key(SK4, ACTIVE_HIGH, switchTurnedOn, nullptr); // Object-Variable
23
24 void setup() { } // nothing to do here...
26 void loop() { } // or here
```

come parametro in base alla sua dichiarazione. Esegue il debouncing dell'interruttore quando viene premuto o rilasciato il pulsante K4 e chiama la funzione `switchTurnedOn()` quando viene premuto. Il metodo `toggle()` della classe `LED` della libreria `MCCAB_Lib` viene attivato nella funzione `switchTurnedOn()` alla riga 19 per invertire lo stato corrente del diodo luminoso LD10.

Poiché i pin di connessione per l'interruttore e il LED sono configurati automaticamente come ingresso e uscita quando gli oggetti vengono dichiarati, nella funzione `setup()` alla riga 24 di questo sketch non è necessario fare altro.

Anche la funzione `loop()` alla riga 26 non contiene istruzioni perché l'unica azione da eseguire in questo sketch è la commutazione dello stato del LED quando viene premuto il pulsante K4. Questa azione è condizionata dagli eventi della classe `KeySwitch` chiamando la funzione `switchTurnedOn()`. Utilizzando queste Classi della libreria `MCCAB_Lib` in sketch più articolati, le due funzioni `setup()` e `loop()` del modello software standard di Arduino non avrebbero bisogno di interrogare continuamente lo stato dei componenti periferici, lasciandoli così liberi per compiti più importanti.

12 Sketch di Progetto e 46 Esercizi

Per la training board MCCAB è disponibile un manuale di istruzioni dettagliato, che può essere scaricato dal sito web [1]. La training board MCCAB e la libreria `MCCAB_Lib` saranno descritte in dettaglio anche in un libro di prossima pubblicazione (vedi nota dell'editore).

Il libro spiega in dettaglio le basi hardware e software di un sistema a microcontrollore e introduce il linguaggio di programmazione C, utilizzato per scrivere gli sketch Arduino. Principalmente il libro è incentrato su esercizi pratici, per cui il concetto chiave utilizzato per acquisire le competenze necessarie per realizzare i propri progetti è "imparare facendo". La sezione pratica comprende 12 sketch di progetto e 46 esercizi, in modo che le conoscenze si sviluppino man mano che si lavora sui numerosi esempi. Gli esercizi sono strutturati in modo tale da dare al lettore un compito da risolvere con la training board MCCAB, utilizzando le conoscenze acquisite nella sezione teorica del libro. Per ogni esercizio è presente una spiegazione dettagliata e un esempio di soluzione ben commentato che aiuta a risolvere i problemi. ◀

220450-05

Domande o Commenti?

Se avete domande tecniche, potete scrivere al team editoriale di Elektor a editor@elektor.com.

Note sull'Autore

Wolfgang Trampert sviluppa e programma sistemi a microcontrollore da quando ha terminato gli studi di elettronica. La sua attività di ingegnere ha sviluppato soluzioni basate su microcontrollori per soddisfare le esigenze dei clienti. È autore di numerosi libri e articoli specialistici e tiene corsi di formazione sul tema dei microcontrollori.



Prodotti correlati

> **MCCAB Training Board (SKU 20295)**
www.elektor.com/20295

LINK WEB

[1] The MCCAB_Lib Library:
<http://www.elektor.de/20295>



Arduino & Co – Measure, Control, and Hack

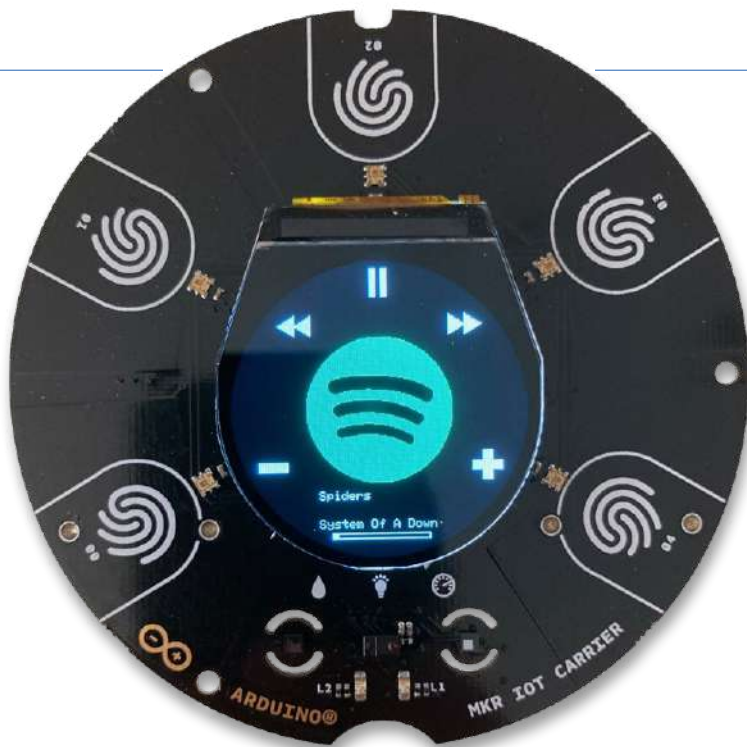
With a simple Pro Mini board and a few other components, projects that 20 or 30 years ago were unthinkable (or would have cost a small fortune) are realized easily and affordably in this book: From simple LED effects to a full battery charging and testing station that will put a rechargeable through its paces, there's something for everyone.

www.elektor.com/20243



Un Controller Per Spotify

L'Oplà IoT Kit è (Quasi) Tutto Ciò Che Vi Serve



Di Altuğ Bakan (Turchia)

Il kit Arduino Oplà IoT contiene la Maker Board MKR WiFi 1010 e una scheda carrier che integra relè, un display OLED di forma circolare, pulsanti touch capacitivi e alcuni sensori. In questo articolo descriviamo come costruire un controller portatile per il popolare lettore musicale Spotify. Ovviamente, serve un certo livello di sicurezza.

Grazie alle sue capacità Wi-Fi, la Maker Board MKR WiFi 1010 di Arduino è un cervello perfetto per il vostro prossimo progetto IoT. Con il kit Arduino Oplà IoT, che contiene questa scheda Maker e una scheda carrier (Figura 1), avrete tutto ciò che vi serve. Quest'ultima integra relè, un display OLED di forma circolare e pulsanti

touch capacitivi. Il kit comprende anche un sensore di umidità e un sensore PIR (Figura 2). Progetti quali allarmi per la sicurezza domestica e per l'irrigazione automatica delle piante saranno quindi facili da realizzare.

La funzione Wi-Fi consente inoltre di controllare i programmi in esecuzione sul PC, se dotati di interfaccia di rete. I pulsanti al tocco, il display, lo zoccolo della batteria e il contenitore consentono di progettare facilmente un controller portatile per diversi tipi di software per PC, in aggiunta a mouse e tastiera (Figura 3).

Sono un fan del lettore Spotify e così ho usato il kit Oplà per costruire il mio controller wireless per la riproduzione musicale. È possibile toccare i pulsanti per passare al brano successivo e a quello precedente, per riprodurre/mettere in pausa un brano e per aumentare e diminuire il volume. Per fare tutto ciò, è ovvio che il lettore Spotify debba essere in esecuzione sul PC o sullo smartphone.

Comunicazione Sicura

Spotify è dotato di un'interfaccia di programmazione facile da usare per il controllo del lettore tramite la rete; tuttavia, per questo è necessaria la licenza Spotify Plus. Ovviamente, serve una certa sicurezza. Per utilizzare l'API web di Spotify, che si basa su REST, è necessario prima autenticarsi presso il server degli account Spotify con il nome utente e la password di accesso. Una volta autenticati, il software deve inviare un Client ID e un Client Secret. Il server Spotify restituirà un token di accesso, che dovrà essere inviato a ogni chiamata dell'API Web per controllare il lettore. Questo flusso di autenticazione in due fasi si basa sul famoso processo OAuth2 (vedi Figura 4). Come ottenete i vostri Client ID e Secret? Basta lo

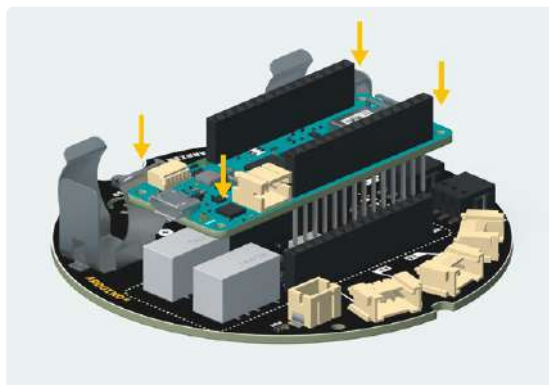


Figura 1: La Maker Board MKR WiFi 1010 viene collocata sulla carrier, che integra relè e altre utili periferiche.



Figura 2: Il kit Arduino Oplà IoT.



Figura 3: Il portabatteria rende portatile il kit Oplà IoT.

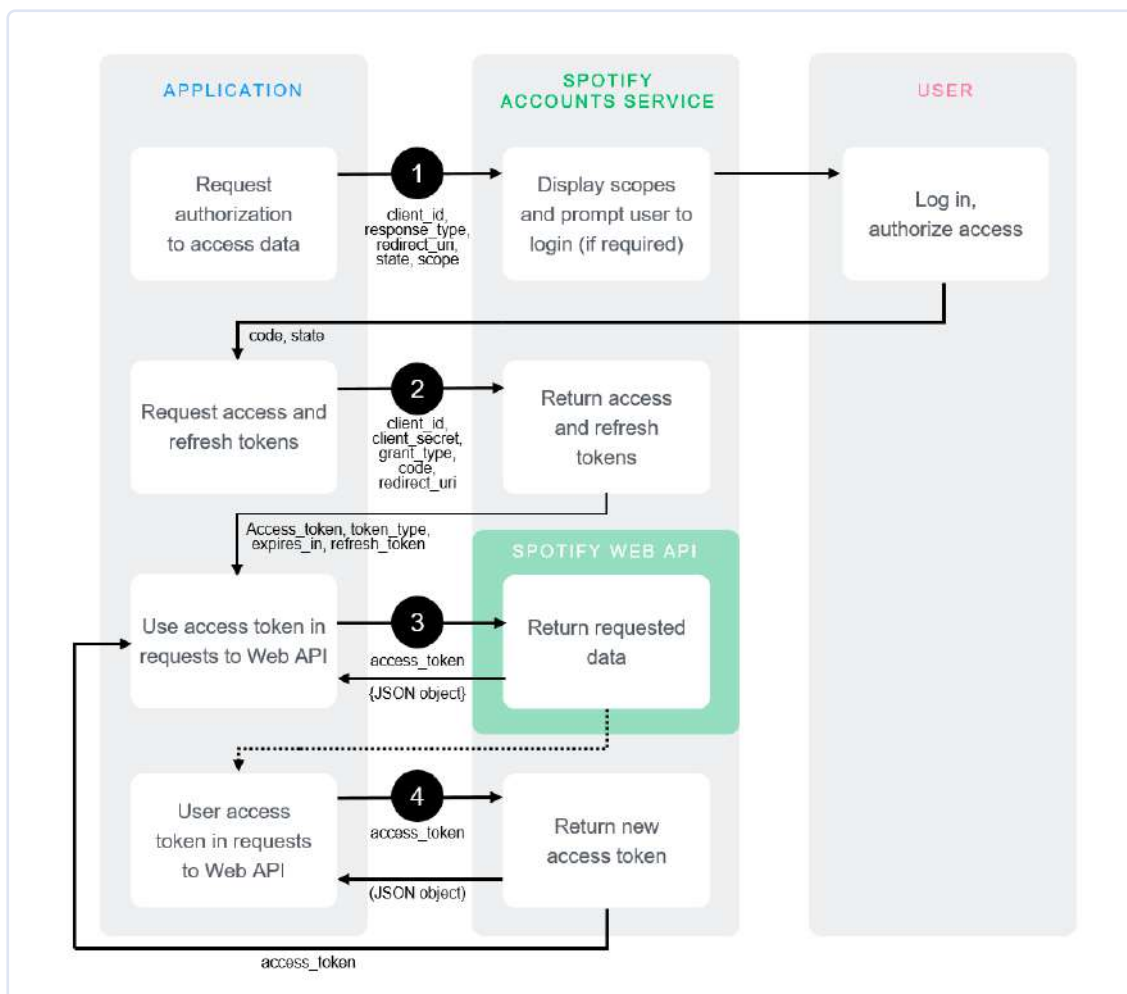


Figura 4: Il flusso di autenticazione a più fasi si basa sul famoso processo OAuth 2.0.

Figura 5: È necessario creare una "App" per ottenere ...

Figura 6: ... i vostri Client ID e Client Secret.

Spotify App Builder [1], che potete usare per progettare il vostro software per PC o l'applicazione mobile per controllare Spotify (**Figura 5**). Tuttavia, in questa sede non lo facciamo; vogliamo solo le credenziali (vedi **Figura 6**). Client ID cliente e Secret devono essere memorizzati sulla nostra scheda Arduino MKR. Naturalmente, è possibile farlo in maniera hardcoded nello sketch di Arduino, ma esiste un modo più comodo e sicuro. L'Arduino Web Editor [2] offre un tab Secrets, in cui è possibile impostare variabili ambientali da utilizzare successivamente nel codice (**Figura 7**). È sufficiente inserire Client ID e Secret del client Spotify, nonché il nome e la password della rete Wi-Fi nei campi della scheda. Se si compila e si carica il software sul controller, verranno caricati anche i valori segreti individuali per essere utilizzati dal codice del progetto. Nel vostro sketch dovete sostituire le stringhe contenenti i dati sensibili scrivendo un'espressione `SECRET_xxx` - quindi, ad esempio: `SECRET_SPOTIFY_CLIENT`.

Autenticazione

Per avviare il flusso OAuth2, è necessario autenticarsi a Spotify. Quando si avvia il controller Spotify descritto in questo articolo, esso si collega alla rete Wi-Fi specificata (quella di casa) e mostra sul display OLED l'indirizzo IP ottenuto dal router. Volevo dare all'utente la possibilità di autenticarsi facilmente a Spotify, quindi ho creato questo approccio. Il controller Arduino genera una piccola

Figura 7: Prima di compilare e caricare il codice, inserite tutti i valori privati nel Tab Secrets dell'Arduino Web Editor.

Figura 8: Pagina Web offerta dal controller per accedere a Spotify.

pagina web che viene visualizzata in un browser, quando si inserisce l'indirizzo IP del controller (**Figura 8**). Questa piccola pagina web contiene un link. (Fate riferimento al **Listato 1** per vedere come viene generata la pagina web nel codice Arduino). Se cliccato, il browser va alla pagina di autenticazione di Spotify, dove è possibile effettuare facilmente il login. A questo punto viene chiesto se si dà al controller il permesso di controllare Spotify (**Figura 9**). Nota bene: per far funzionare tutto questo, è necessario

inserire l'indirizzo IP del controller come "Redirect URI" nell'editor dell'app Spotify (**Figura 10**). D'ora in poi il controller Arduino può ottenere il token di accesso all'API inviando a Spotify il Client ID e il Secret (**Listato 2**). Il token di accesso deve essere aggiornato regolarmente durante il funzionamento. Anche questo viene fatto da una funzione nello sketch (**Listato 3**), che viene richiamata ogni 3000 secondi.



Listato 1: Pagina Web offerta dal controller per autenticarsi a Spotify.

```
String webpage = "<!DOCTYPE html>\n";
webpage += "<html><body>";
webpage += getStyle();
webpage += "<a href=\"https://accounts.spotify.com/authorize?client_id=";
webpage += SPOTIFY_CLIENT;
webpage += "&response_type=code&redirect_uri=http://";
webpage += ip_address;
webpage += "/redirect/&scope=user-read-playback-state user-modify-playback-state\">Authenticate Spotify</a>\n";
webpage += "</body></html>";
wifiClient.print(webpage);
```

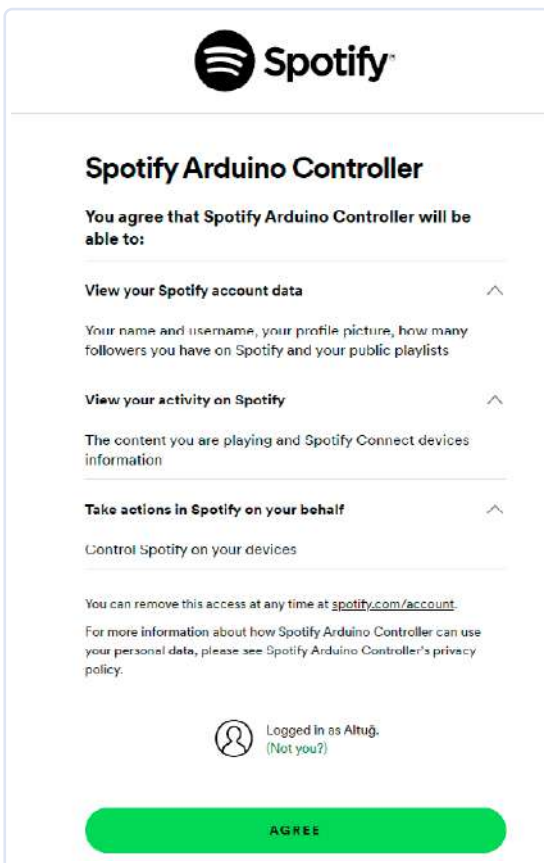
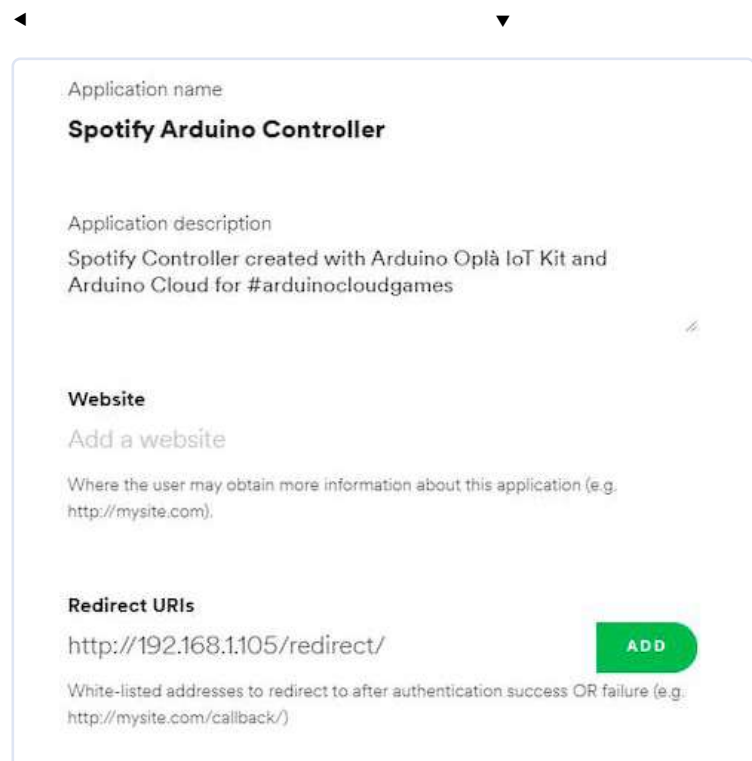


Figura 9: Se avete effettuato l'accesso a Spotify, dovete dare al controller il permesso di agire per vostro conto.

Figura 10: URI di reindirizzamento: L'indirizzo del controller nella vostra rete domestica.





Listato 2: Funzione per ottenere il token da Spotify per ulteriore uso dell'API.

```
// Get the user authorization token
bool getAccessToken(String userCode) {
    String postData = "grant_type=authorization_code&code=" + userCode + "&redirect_uri="
        "http://" + ip_address + "/redirect/";

    authClient.beginRequest();
    authClient.post("/api/token");
    authClient.setHeader("Content-Type", "application/x-www-form-urlencoded");
    authClient.setHeader("Content-Length", postData.length());
    authClient.sendBasicAuth(SPOTIFY_CLIENT, SPOTIFY_SECRET);
    // send the client id and secret for authentication
    authClient.beginBody();
    authClient.print(postData);
    authClient.endRequest();

    // If successful
    if (authClient.responseStatusCode() == 200) {
        lastTokenTime = millis();
        DynamicJsonDocument json(512);
        deserializeJson(json, authClient.responseBody());
        accessToken = json["access_token"].as<String>();
        refreshToken = json["refresh_token"].as<String>();
        return true;
    }
    return false;
}
```



Listato 3: Funzione per il refresh del token.

```
// Refresh the user authentication token
void refreshAccessToken() {
    String postData = "grant_type=refresh_token&refresh_token=" + refreshToken;
    authClient.beginRequest();
    authClient.post("/api/token");
    authClient.setHeader("Content-Type", "application/x-www-form-urlencoded");
    authClient.setHeader("Content-Length", postData.length());
    authClient.sendBasicAuth(SPOTIFY_CLIENT, SPOTIFY_SECRET);
    // send the client id and secret for authentication
    authClient.beginBody();
    authClient.print(postData);
    authClient.endRequest();

    // If successful
    if (authClient.responseStatusCode() == 200) {
        lastTokenTime = millis();
        DynamicJsonDocument json(256);
        deserializeJson(json, authClient.responseBody());
        accessToken = json["access_token"].as<String>();
    }
}
```



Figura 11: Le funzioni dei pulsanti sono visualizzate sul display.



Listato 4: Esempio d'uso dell'API (brano successivo e precedente).

```
// Skip a song towards a given direction
void skipSong(String direction) {
  apiClient.beginRequest();
  apiClient.post("/v1/me/player/" + direction);
  apiClient.setHeader("Content-Length", 0);
  apiClient.setHeader("Authorization", "Bearer " + accessToken);
  apiClient.endRequest();
}
```

Funzionamento

Il resto del codice è meno complesso. Il dispositivo mostra il logo Spotify e la funzione dei pulsanti sul display OLED (Figura 11). Se l'utente tocca un pulsante, viene richiamata la funzione API corrispondente. Fate riferimento al **Listato 4** per vedere come viene eseguita questa operazione per saltare un brano a quello precedente o successivo. Nel codice è presente anche una funzione che richiede lo stato del lettore all'API di Spotify. La risposta è una stringa JSON. Per elaborare più facilmente queste stringhe, utilizzo la libreria *ArduinoJson.h* e alcune mie funzioni. Per ottenere lo stato dei pulsanti, controllare i LED e mostrare la grafica sull'OLED, utilizzo la libreria *Arduino_MKRIoTCarrier.h*. Se volete, potete leggere il mio codice e trarre ispirazione per i vostri progetti che potete realizzare con il kit Oplà. Il mio software può essere scaricato all'indirizzo [3].

Connessione al cloud

Ho anche impostato una connessione al Cloud di Arduino e ho creato un pannello di controllo che mostra il brano corrente e il nome dell'artista accanto al volume del dispositivo (Figura 12). Naturalmente è possibile creare il proprio pannello personalizzato, con i dati che si desidera. Il mio progetto si è aggiudicato il terzo posto all'Arduino Cloud Games 2022! ◀

220407-05

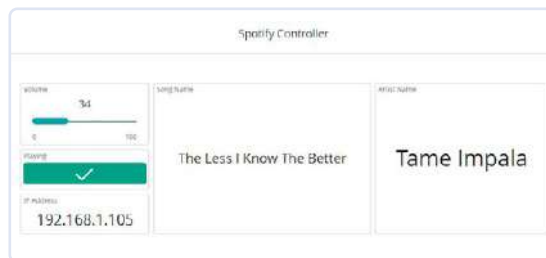


Figura 12: Il brano corrente e il nome dell'artista vengono inviati ad Arduino Cloud, dove sono visibili in un pannello personale.

Domande o Commenti?

Se avete domande tecniche scrivete una e-mail all'autore a mail@alt.ug oppure al Team editoriale di Elektor a editor@elektor.com.

Note sull'Autore

Altug Bakan lavora come ingegnere elettronico, principalmente con sistemi embedded. Nel suo lavoro ama utilizzare Arduino per la prototipazione rapida e la facilità d'uso. Le sue materie elettroniche preferite sono la programmazione embedded bare-metal e l'Internet delle cose (IoT).



Prodotti Correlati

- > **Arduino Oplà IoT Kit**
www.elektormagazine.com/arduino-opla-iot-kit

LINK WEB:

- [1] Spotify App Builder: <https://developer.spotify.com/dashboard/>
- [2] Arduino Web Editor: <https://create.arduino.cc/editor>
- [3] This Project on create.arduino.cc: <https://create.arduino.cc/projecthub/Altug/opla-spotify-controller-6e7bc4>

Realizzare, Distribuire e Mantenere Applicazioni Scalabili e Sicure

Con Arduino Portenta X8 Dotato di Processore Applicativo i.MX 8M Mini di NXP e di Secure Element EdgeLock® SE050

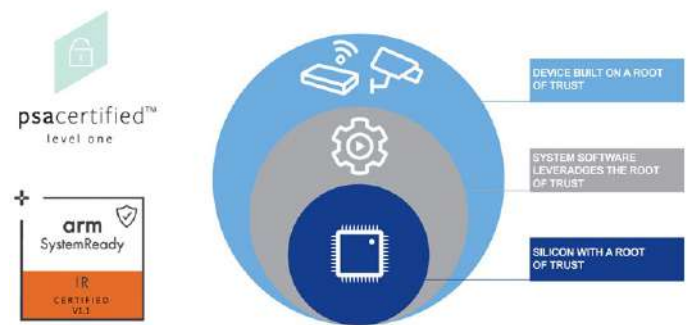
Contributo di NXP Semiconductors

L'introduzione di un dispositivo IoT sul mercato comporta un notevole sforzo di progettazione e sviluppo, con problemi di scalabilità, sfide di sicurezza e vincoli nascosti dietro ogni angolo. L'aggiunta di intelligenza rende il tutto ancora più complicato. Per questo la scelta dell'hardware e del software di sviluppo giusti è fondamentale per portare sul mercato più velocemente prodotti edge sicuri. Questo articolo presenta la piattaforma Arduino Portenta X8, un SOM sicuro di livello industriale, basato sul processore per applicazioni i.MX 8M Mini di NXP e su un elemento di sicurezza hardware EdgeLock® SE050 integrato. Questa piattaforma certificata PSA è anche Arm® SystemReady IR per garantire la sicurezza.

Arduino Portenta X8 è un potente sistema-su-modulo (SOM) di livello industriale con sistema operativo Linux® precaricato, in grado di eseguire software indipendente dal dispositivo grazie alla sua architettura modulare a container. Offre due approcci: flessibilità d'uso di Linux combinata con applicazioni in tempo reale attraverso l'ambiente Arduino. La connettività Wi-Fi/Bluetooth® Low Energy a bordo permette di aggiornare il sistema operativo e le applicazioni da remoto, mantenendo sempre l'ambiente kernel Linux ai massimi livelli di prestazioni.

Sicurezza allo Stato dell'Arte

Il sistema basato su container integra diversi livelli di sicurezza, a partire dal layer hardware che include il Secure Element di NXP. Utilizza la piattaforma DevOps basata su cloud di Foundries.io [1] per reinventare il modo in cui le soluzioni Linux embedded vengono costruite, testate, distribuite e mantenute. Il Portenta X8 include il sistema operativo open-source personalizzabile Linux microPlatform, realizzato secondo le migliori pratiche industriali per la sicurezza end-to-end, gli aggiornamenti OTA incrementali e la gestione del parco macchine. Il layer di virtualizzazione consente agli utenti di distribuire software indipendente dal dispositivo che gira all'interno di un ambiente controllato. È possibile creare i propri container utilizzando Docker e scaricare immagini predefinite dall' Hub Docker o da altri registri pubblici disponibili, per creare un'applicazione su misura. Se lo sviluppatore vuole entrare nel mondo embedded, può farlo facilmente costruendo la propria applicazione, eseguendola su un container, mettendola sulla scheda e testandola immediatamente. Questo offre un'ampia gamma di opportunità, combinando le capacità di Linux con l'esperienza dello standard Arduino.



Container e sicurezza nel Portenta X8.

Portenta X8 ha ottenuto la certificazione PSA e il Secure Element hardware EdgeLock SE050 di NXP fornisce la generazione della chiave, le operazioni di crittografia accelerata e l'archiviazione sicura. Inoltre, X8 ha ottenuto la certificazione Arm® SystemReady [2] e ha integrato i servizi Parsec, diventando così uno dei primi prodotti Cassini o dispositivi Cloud Native Edge disponibili sul mercato per gli sviluppatori. Esegue senza problemi Fedora IoT, Fedora Server, Debian e Linux microPlatform. Consentendo la migrazione all'edge dei carichi di lavoro cloud-nativi, Portenta X8 contribuisce a creare un'esperienza cloud-nativa per gli sviluppatori nell'ecosistema IoT sicuro e diversificato di Arm.

EdgeLock SE050 - Un'Ancora di Fiducia per l'IoT

EdgeLock SE050 [3] di NXP è un hardware di sicurezza discreto e resistente alle manomissioni per la protezione dell'identità di un dispositivo, che include chiavi crittografiche e certificati. Si tratta di un elemento di sicurezza incorporato indipendente che viene collegato al processore principale tramite l'interfaccia I²C. EdgeLock SE050 è certificato Common Criteria EAL 6+ per l'hardware e il sistema operativo. Questo elemento sicuro pronto all'uso per i dispositivi IoT fornisce una root-of-trust a livello di IC e offre una reale sicurezza end-to-end, dall'edge al cloud, senza la necessità di implementare codice di sicurezza né di gestire chiavi e credenziali critiche.

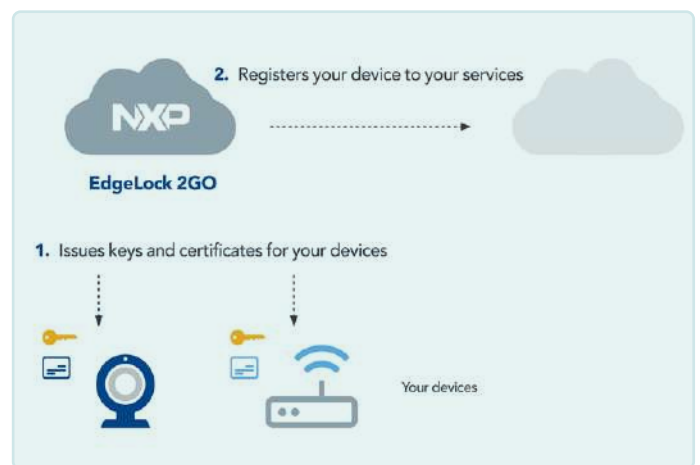


Root-of-Trust basata-su-silicio: il Secure Element EdgeLock® SE050.

Fornito come soluzione pronta all'uso, EdgeLock SE050 è dotato di molteplici algoritmi e protocolli crittografici pre-implementati e di un pacchetto completo di supporto al prodotto che semplifica la progettazione e riduce il time-to-market. Oltre alle librerie per diverse MCU e MPU, il pacchetto di supporto offre anche l'integrazione con i più comuni sistemi operativi, tra cui Linux, RTOS e Android.

Quando implementano le operazioni di integrazione dei dispositivi nel cloud, i progettisti IoT devono affrontare due sfide principali: la fornitura delle identità dei dispositivi e la gestione delle stesse, una volta rilasciati sul campo. La fase di messa in servizio del dispositivo si riferisce all'installazione di chiavi e certificati. La gestione delle identità si riferisce all'aggiornamento, all'aggiunta o alla revoca di chiavi e certificati durante il ciclo di vita del dispositivo.

Per aiutare i progettisti a risolvere queste sfide, NXP offre il servizio gestito EdgeLock 2GO [4]. La piattaforma è una combinazione di hardware e servizi appositamente realizzata che stabilisce una root-of-trust basata-su-silicio. EdgeLock 2GO emette le identità necessarie per i dispositivi IoT e installa le credenziali in modo sicuro nell'hardware EdgeLock SE050. Inoltre, registra automaticamente il dispositivo IoT direttamente al servizio cloud.



NXP gestisce le credenziali del dispositivo.

Questo servizio flessibile supporta vari tipi di credenziali e applica diverse configurazioni, a seconda del progetto. Le credenziali possono essere rinnovate oppure aggiunte ai dispositivi rilasciati sul campo. Con la messa in servizio di EdgeLock SE050 e EdgeLock 2GO, gli utenti ottengono una soluzione end-to-end semplice, sicura e flessibile. Con la continua espansione dell'IoT, aumentano anche i rischi. La combinazione EdgeLock di NXP, con la sua sicurezza basata-sull'hardware e il servizio di gestione delle credenziali, offre ai produttori di dispositivi un modo più sicuro di operare. Con NXP EdgeLock a supporto della distribuzione di un dispositivo, si riduce il time-to-market e i costi quotidiani di gestione di una distribuzione IoT, con la certezza che i dispositivi saranno protetti da una sicurezza di alto livello.

Liberare la Potenza: Offrire una Maggior Velocità e un'Efficienza Migliorata

Il SoC i.MX 8M Mini [5] è il primo processore multicore per applicazioni embedded di NXP, realizzato con l'avanzata tecnologia di processo 14LPC FinFET, che garantisce maggiore velocità ed efficienza energetica migliorata. La famiglia di processori applicativi i.MX 8M Mini combina prestazioni di calcolo elevate, efficienza energetica e sicurezza embedded necessarie per gestire le applicazioni di edge node computing, streaming multimediale e machine learning in rapida crescita.



Il SoC i.MX 8M Mini è disponibile nelle varianti single, dual e quadcore con Arm® Cortex®-A53 che funziona fino a 1,8 gigahertz per core. Realizzato con un processo avanzato a basso consumo, il complesso di core è ottimizzato per un funzionamento senza ventole, bassa dissipazione termica del sistema e una lunga durata della batteria. I core Cortex-A possono essere spenti mentre il sottosistema Cortex-M4 esegue il monitoraggio low-power del sistema in tempo reale. Il controller DRAM supporta memorie LPDDR4, DDR4 e DDR3L a 32 bit/16 bit, garantendo una grande flessibilità nella progettazione del sistema stesso.

Le opzioni di core i.MX 8M Mini sono ottimizzate per applicazioni ultra-low-power, anche sub-Watt in casi specifici, ma offrono tutta la potenza di elaborazione necessaria per applicazioni consumer, audio, industriali, di apprendimento e inferenza del Machine Learning su una vasta gamma di cloud provider. Il SoC i.MX 8M Mini integra anche l'accelerazione hardware a 1080p per consentire applicazioni video bidirezionali, grafica 2D e 3D per offrire una ricca esperienza visiva HMI e funzionalità avanzate per applicazioni audio di qualità elevata. Un'ampia selezione di interfacce ad alta velocità consente una maggiore connettività del sistema e punta a qualifiche di livello industriale.

Esempi applicativi includono:

> Automazione Industriale

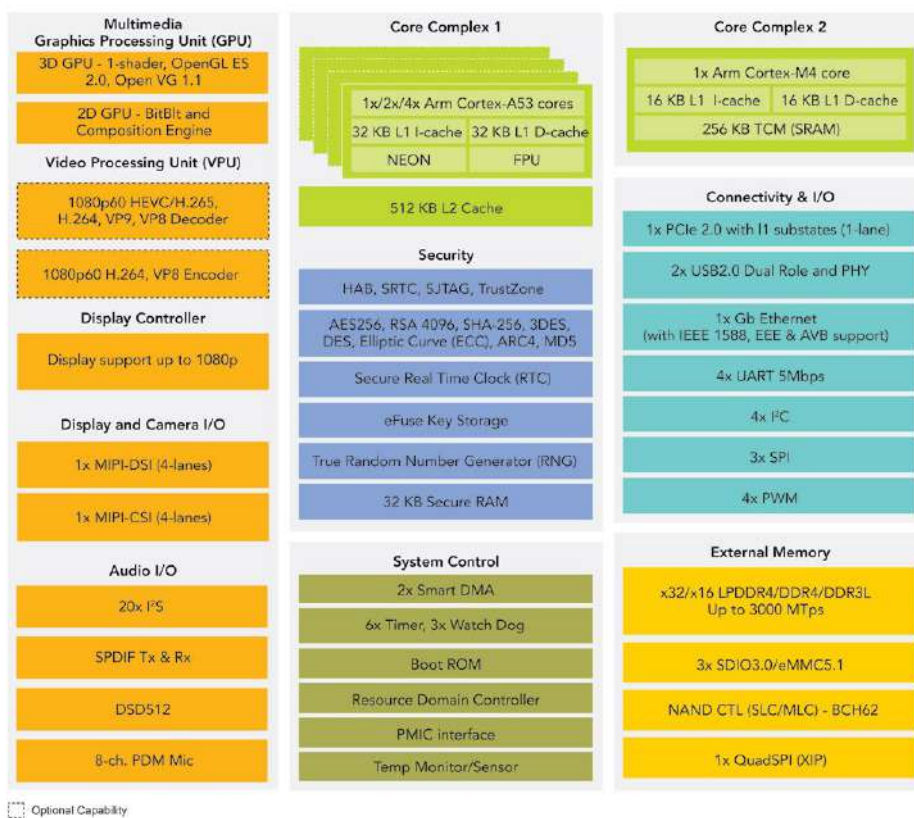
- Il Portenta X8 può quindi fungere da gateway multiprotocollo, inviando i dati al cloud o al sistema ERP tramite Wi-Fi, LoRa, NB/IoT, LTE Cat.M1.
- La disponibilità di container Linux come ROS all'interno dell'ambiente Arduino rende il Portenta X8 ideale per i veicoli a guida autonoma.

> Automazione degli Edifici

- Interagendo con i sensori intelligenti dell'ambiente, Portenta X8 consente di implementare l'elaborazione ML e delle immagini in tempo reale all'edge di rete.
- I chioschi intelligenti di solito sfruttano diversi componenti (ad esempio lettori di schede, telecamere, microfoni), richiedendo una selezione diversificata di I/O. Se abbinato a una Max Carrier, il Portenta X8 garantisce la connettività Wi-Fi e consente agli amministratori di monitorare in remoto l'utilizzo della macchina.
- Il Portenta X8 può controllare simultaneamente i sistemi HVAC, accendere/spengere elettrodomestici intelligenti, regolare autonomamente l'illuminazione e controllare gli accessi all'edge di rete.

Iniziate a sviluppare oggi stesso con il SOM Portenta X8 [6], sicuro e di livello industriale, con una densità di calcolo eccezionale. ▶

220576-05



Schema a blocchi del processore applicativo i.MX 8M Mini.

LINK WEB:

- [1] Foundries.io: <https://foundries.io/>
- [2] Arm SystemReady: <https://www.arm.com/architecture/system-architectures/systemready-certification-program>
- [3] EdgeLock SE050: <https://bit.ly/EdgeLockSE050>
- [4] EdgeLock 2GO: <https://bit.ly/EdgeLock2GO>
- [5] i.MX 8M Mini: <https://bit.ly/iMX8MMini>
- [6] Portenta X8 SOM: <https://www.arduino.cc/pro/hardware/product/portenta-x8>

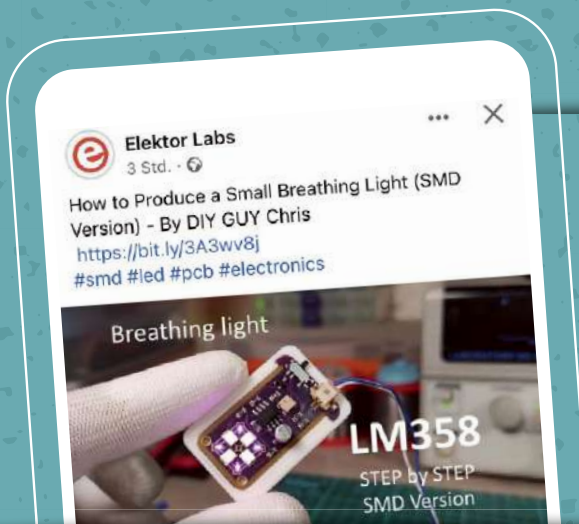
Nuovo Progetto di *Elettronica,* Anche con *Arduino?*

Condividilo con la nostra comunità!

Seguici su:



www.elektor.com/TW



www.elektor.com/Intsta



www.elektor.com/FB





Portare l'Elemento Umano nella Produzione

▲
Daria Baradel (in primo piano) con i colleghi di Arduino.

Incontro con Daria Baradel, Responsabile di Produzione di Arduino

Di Keith Jackson (Arduino)

Come ha fatto uno dei produttori di elettronica più famosi al mondo a superare con successo le sfide della supply chain degli ultimi tre anni? Con una passione incrollabile per l'innovazione, una stretta collaborazione con i partner locali e un tocco umano.

Keith Jackson: Ciao, Daria. Con tutto quello che sentiamo dai notiziari sulla carenza globale di componenti elettronici che bloccano la produzione nelle fabbriche di tutto il mondo, deve essere stato un anno difficile per Arduino?

Daria Baradel: Sì. Sebbene tutti abbiano sentito parlare di carenze nel settore dell'elettronica che interessano qualsiasi cosa, dalle auto nuove alle console di gioco, non è solo la mancanza di scorte a causare le maggiori difficoltà, ma anche i ritardi impre-

vedibili e i costanti aumenti dei costi che rendono praticamente impossibile mantenere il controllo. L'intero team ha dovuto adattare il proprio approccio e diventare più flessibile. Mantenere il controllo richiede prevedibilità e gestione del rischio, ma quando la prevedibilità viene meno, mantenere la produzione è una sfida quotidiana. In un giorno qualsiasi, una consegna prevista può essere annullata, causando blocchi di produzione in fabbrica. Tuttavia, non sono sempre le cancellazioni a causare cambiamenti dell'ultimo minuto nella programmazione della produzione. Ci sono casi esattamente opposti in cui, improvvisamente, si presenta la possibilità di ricevere casualmente una consegna di componenti che non era prevista fino al 2023. Nei casi in cui questa era l'unica riga in sospeso nel BOM, ci offre in modo inaspettato l'opportunità di rimettere in produzione un modello, per cui adattiamo costantemente i piani a ciò che è disponibile e a ciò che chiedono i nostri clienti. Abbiamo dovuto rivedere tutte le procedure standard e adattare a questa situazione instabile e imprevedibile, rendendole meno standard e più flessibili per essere pronti a gestire ogni singolo problema che si sarebbe potuto presentare.

Keith: Puoi condividere alcuni esempi di come tu e il team avete adattato il vostro approccio per affrontare queste sfide?

Daria: All'inizio cercavo di prevedere cosa sarebbe successo, ma questo si è rivelato impossibile. Perciò ho dovuto cambiare mentalità e adottare procedure più proattive. Come team abbiamo convenuto che era necessario ottenere un maggiore controllo sulle consegne; pertanto, ogni venerdì verificavamo con tutti i nostri fornitori le consegne previste per la settimana successiva. Per i componenti principali, poi, seguiamo la situazione su base giornaliera. Non si tratta solo di lavorare con i nostri fornitori; è diventato assolutamente necessario lavorare a stretto contatto con il team Hardware di Arduino, responsabile della progettazione e del collaudo dei prodotti. Prima di finalizzare il BOM di un nuovo prodotto o anche di confermare quali componenti montare su una scheda durante la prototipazione, entrambi i team si incontrano per valutare la disponibilità e l'affidabilità del fornitore, come parte del processo di progettazione. La disponibilità di un componente, così come la sua capacità, ormai è un elemento da considerare nel progetto finale. Ma non è un aspetto del tutto negativo, perché adesso il team ha una comprensione molto più approfondita delle schede e dei componenti, essendo

stato coinvolto molto prima nel processo di progettazione. Quindi tutti noi viviamo e respiriamo ogni nuovo prodotto fin dal concetto iniziale.

Nel 2022, il team della catena di fornitura e quello dell'hardware hanno gestito la convalida di oltre 100 componenti alternativi e rielaborato il progetto di 16 prodotti per adeguare il BOM e poter procedere alla produzione.

Keith: Com'è stato il rapporto con i fornitori, visto che Arduino è solo uno dei tanti clienti che chiedono componenti?

Daria: Ironia della sorte, negli ultimi anni ho investito molto tempo nell'automatizzare il più possibile le procedure di acquisto dei componenti. Quest'anno, però, l'interazione umana è tornata in primo piano. È essenziale comunicare e lavorare con altri esseri umani. Solo attraverso continue telefonate siamo in grado di tenere il passo con gli ultimi cambiamenti e le conseguenze che ne derivano. Probabilmente, con i continui messaggi che gli invio quotidianamente potrei essere considerata una stalker!

Molti dei nostri fornitori si interessano all'intero concetto di Arduino, e molti di essi sono anche degli appassionati. Pertanto, è evidente che da alcuni fornitori abbiamo ricevuto un grande supporto, mentre da altri ne abbiamo ottenuto uno scarso; ma nel complesso, quelli da cui abbiamo ricevuto un ottimo supporto adesso sono fornitori di Arduino realmente affiatati, che lavorano a stretto contatto con noi come parte di un'unica grande comunità.

Keith: Come sta andando la produzione e quali sono le tue prospettive per il futuro?

Daria: Nonostante tutto quello che è successo quest'anno, i volumi di produzione sono comunque aumentati di oltre il 20% rispetto alla stessa fase del 2021. Questo risultato è stato ottenuto grazie a uno sforzo trasversale di tutto il gruppo. Il team Hardware ha contribuito immensamente controllando e approvando letteralmente centinaia di componenti alternativi, in molti casi montandoli e testandoli per l'approvazione il giorno stesso per assicurarci di poter reperire le scorte disponibili. Quindi, anche se in termini di carenza di componenti non si vede luce alla fine del tunnel, con molti tempi di consegna che si estendono fino alla fine del 2023, sono ancora convinta che possiamo continuare a crescere perché, dopo tutto, le aziende sono fatte da esseri umani. I cambiamenti di mentalità e di procedure che abbiamo dovuto attuare quest'anno non potranno che renderci più forti ed efficaci per il futuro.





Keith: Hai parlato di collaudo e approvazione di componenti alternativi nello stesso giorno. Come si può fare?

Daria: Avere i nostri produttori vicini agli uffici di ricerca e sviluppo di Arduino è stato un enorme vantaggio durante la pandemia COVID e in questi tempi di carestie. Arduino si è sempre vantata di produrre in Italia, con tutte le schede prodotte in due stabilimenti della regione Piemonte. In pratica, quando si sono verificati dei problemi, gli ingegneri sono saliti in auto e si sono recati immediatamente nella vicina fabbrica di Torino per controllare e cambiare i componenti, se necessario. Quando ci viene comunicata la disponibilità di un componente alternativo, è necessario confermare l'ordine il giorno stesso, altrimenti lo stock viene preso da qualcun altro. Bisogna essere molto rapidi. Quindi, gli ingegneri si recano in fabbrica, testano il prodotto alternativo e ci danno il via libera, consentendo di effettuare l'ordine in giornata. Se la produzione fosse stata all'estero, questo non sarebbe stato possibile.

Keith: Arduino deve essere una delle poche aziende di elettronica che utilizza ancora produttori locali per produrre le sue schede?

Daria: Sì. Produrre localmente è stata una scelta deliberata, poiché Arduino ha sempre adottato un approccio positivo alla sostenibilità, all'impatto ambientale e alla comunità locale. I nostri obiettivi di sostenibilità sono più di un semplice foglio di carta: abbiamo sempre cercato di ridurre il più possibile le emissioni di CO₂ e di limitare gli sprechi. Per esempio, regaliamo le vecchie scorte di magazzino alle scuole locali e la produzione e il confezionamento dei prodotti avvengono nel raggio di 50 km dalla sede di Torino. Nella scelta dei nostri partner e nella produzione dei prodotti a livello locale si applica lo spirito comunitario. Il kitting e l'imballaggio sono realizzati da un'azienda locale che impiega attivamente persone con disabilità, offrendo loro l'opportunità di essere indipendenti.

Keith: Sembra che tu sia molto orgogliosa di lavorare per Arduino. Ma oltre a gestire la carenza quotidiana di componenti, cos'altro ispira Daria?

Daria: Di formazione sono un ingegnere, con due master in ingegneria della produzione e gestione ingegneristica. Nel 2014, quando sono entrata in azienda come project manager, producevamo tra le 3.000 e le 5.000 schede al mese. Oggi la produzione è cresciuta fino a oltre 50 volte, il che significa che ora ho un team completo di otto persone che si occupano della gestione.

Oltre al lavoro, però, mi piace viaggiare e praticare molti sport, soprattutto alpinismo, corsa, wind surf e pallavolo da spiaggia. Trovo che le attività sportive siano il modo migliore per scaricare lo stress della giornata. Ma se dovessi avere un sogno per il futuro, sarebbe quello di possedere una fattoria, poiché amo gli animali, provenendo da una famiglia che allevava mucche, pecore e capre. Sono sicura che anche allora userò Arduino dappertutto per renderla il più efficiente possibile. ◀

220426-05

Note sull'Autore

Keith Jackson lavora nel marketing di Arduino ed è un appassionato di tutto ciò che riguarda Arduino, dato che non si tratta solo di un'azienda o di un marchio, ma di un'intera comunità eterogenea.

Domande o Commenti?

Avete domande o commenti da fare su questo articolo? Contattate l'autore a: k.jackson@arduino.cc o il team di Elektor a: editor@elektor.com

MicroPython Enters the World of Arduino

with Stuart Cording & Sebastian Romero

MicroPython has made it to the world of Arduino, providing the first significant alternative to programming in C and C++. So, what's all the fuss, how easy is it to use, and who can benefit from programming in this, for microcontrollers, relatively new language? Stuart Cording will speak with Sebastian Romero (Head of Content, Arduino) during our live webinar to find out more.

Join for free

www.elektor.com/webinar-MicroPython



Schede di Sviluppo

Passato, Presente e Futuro

Di Mark Patrick (Mouser Electronics)

Negli ultimi anni, il significato del termine “scheda di sviluppo” si è perso nella miriade di termini usati per descrivere le schede hardware utilizzate a scopo di sviluppo, tra cui le schede dimostrative (demo), i kit di valutazione e i progetti di riferimento. In questo articolo spieghiamo il significato di questi vari termini e mostriamo come si differenziano dai loro parenti stretti, i computer a scheda singola (SBC). Tracciamo la loro evoluzione dal passato al presente e analizziamo alcune tendenze su come possano evolversi in futuro.

Che cos'è una Scheda di Sviluppo?

È necessario innanzitutto definire chiaramente cosa si intende per scheda di sviluppo e come si differenzia da un computer a scheda singola (SBC). Una scheda di sviluppo viene in genere creata dal produttore di un microcontrollore per metterne in evidenza le caratteristiche (sebbene il termine sia ora spesso applicato anche ad altri tipi di componenti). Un microcontrollore è un circuito integrato che contiene un processore, un po' di RAM, una memoria flash e funzionalità di I/O che gli consentono di interfacciarsi con il mondo reale. Funziona effettivamente come un computer in miniatura racchiuso in un unico package, il cui scopo è fornire agli sviluppa-

tori un modo conveniente per interfacciarsi con esso e controllare componenti esterni come luci, piccoli motori, ecc. Anche un SBC offre queste funzionalità, con la differenza che la CPU, la RAM e la memoria sono contenute in circuiti integrati separati sulla scheda e le interfacce consentono di collegarlo a una tastiera e/o a un display. Il microprocessore di un SBC richiede un sistema operativo, mentre un microcontrollore viene gestito tramite un ambiente di sviluppo integrato (IDE) fornito dal produttore. In molti casi, i produttori creano schede di sviluppo che includono un microcontrollore, ma il cui scopo principale non è quello di dimostrare le caratteristiche del microcontrollore stesso, bensì quelle dei sensori o di altri circuiti integrati a cui si interfaccia. Queste schede vengono chiamate “schede demo”, “kit di valutazione” o, se sono state assemblate per consentire all'insieme di parti di svolgere uno scopo tangibile, “progetti di riferimento”. Lo scopo di alcune schede non è principalmente lo sviluppo dell'hardware, ma l'accesso ai dati del mondo reale di cui gli sviluppatori di software hanno bisogno per creare e perfezionare gli algoritmi necessari per le applicazioni di intelligenza artificiale e apprendimento automatico. Anche se queste schede possono non essere conformi alla definizione e allo scopo originale di “scheda di sviluppo”, oggi si intende collettivamente qualsiasi pezzo di hardware che può essere utilizzato nello sviluppo di software e hardware di nuovi prodotti elettronici.

(Fonte: Shutterstock)

Il passato

La prima scheda di sviluppo per microcontrollori a catturare l'attenzione della comunità ingegneristica è stata rilasciata nel 2006. Questa piattaforma di prototipazione, che in seguito divenne nota come Arduino [1] (**Figura 1**), fu rapidamente adottata da una nuova categoria di progettisti elettronici che comprendeva appassionati, hobbisti e tecnici fai-da-te. Arduino ha gettato le basi per il successo commerciale delle successive piattaforme SBC e di quelle basate su microcontrollori ed è stato presto seguito, nel 2008, da BeagleBoard [2], che ha fornito agli ingegneri una piattaforma di sviluppo a basso costo e open-source supportata dalla comunità. Il 2012 ha visto il rilascio del Raspberry Pi [3], il primo computer moderno a scheda singola. Come BeagleBoard, è stato concepito come una piattaforma educativa destinata a fornire uno strumento a basso costo che gli studenti potessero utilizzare per imparare a scrivere codice. Il fascino di Raspberry Pi è andato ben oltre i soli studenti, ed è stato rapidamente adottato sia da hobbisti che da ingegneri professionisti.

Il presente

Oggi esistono due categorie principali di SBC: quelli proprietari e quelli open-source. Gli SBC proprietari sono tipicamente progettati per l'uso in applicazioni finali e sono stati sottoposti allo stesso tipo di test e garanzia di qualità degli altri prodotti finali. Sono integrati in apparecchiature elettroniche o installati in una configurazione a rack. Gli SBC open-source consentono agli utenti di accedere alla progettazione e al layout dell'hardware e a qualsiasi codice sorgente utilizzato. Ciò consente agli utenti di apprendere rapidamente e facilmente il funzionamento del software e dell'hardware e di adottare il progetto in base alle proprie esigenze.

Oggi le schede di sviluppo e gli SBC dispongono di un'ampia varietà di processori. Si va da quelli basati su processori x86 nell'ambito dei PC tradizionali (AMD e Intel) ai processori ARM utilizzati nelle applicazioni industriali e mobili. Linux e i suoi derivati (Ubuntu, Fedora, Debian, ecc.), Android e Windows CE sono i sistemi operativi più utilizzati sulle SBC. Le schede di sviluppo per microcontrollori non richiedono un sistema operativo e vengono programmate tramite un IDE fornito dal produttore. Le schede di sviluppo a microcontrollore e gli SBC si sono evoluti per includere la connettività wireless (Wi-Fi, Bluetooth) e le più recenti interfacce audio e video, il che significa che alcuni SBC hanno ora caratteristiche equivalenti a quelle presenti in molti PC e tablet.

Il futuro: Le schede di sviluppo diventeranno il prodotto finale

Tradizionalmente, i produttori creavano strumenti di sviluppo con l'intenzione di utilizzarli come ausilio al marketing per migliorare le probabilità di vendita dei loro microcontrollori ai potenziali clienti (spesso definiti

"Design-in" nel settore). Si sperava che, riducendo al minimo la quantità di lavoro necessaria agli ingegneri progettisti per mettere in funzione un componente in laboratorio e rendendo più facile l'accesso e l'approfondimento delle sue caratteristiche, si sarebbero potuti scegliere più facilmente i microcontrollori e le parti accessorie da utilizzare per la prototipazione iniziale del prodotto e, infine, si sarebbero potuti ottenere ordini di volumi più elevati se il componente fosse stato scelto per la produzione di massa. Per i prodotti in cui la differenza nelle specifiche tecniche tra i componenti dei diversi fornitori è trascurabile, si tratta di un approccio prudente. Tuttavia, per i produttori questo approccio è stato, per certi versi, vittima del suo stesso successo. I produttori si sono resi conto che dovevano continuare a ridurre la quantità di lavoro richiesto a un ingegnere per lavorare con i loro prodotti, quindi la scheda di sviluppo è diventata il fattore di differenziazione chiave, soprattutto per i prodotti ampiamente simili a quelli della concorrenza. Le aspettative degli ingegneri progettisti sono aumentate al punto che, anche per i componenti che presentano un vantaggio competitivo chiaramente identificabile (ad esempio, in termini di potenza o velocità), si aspettano ancora che le schede di sviluppo associate abbiano livelli di accessibilità plug-and-play.

I produttori hanno ulteriormente migliorato la loro proposta di valore offrendo progetti di riferimento costituiti da un microcontrollore e altri circuiti integrati (in genere sensori). Inizialmente, questi progetti erano destinati a fornire una guida su come i dispositivi potessero essere interconnessi per emulare le funzionalità elettriche di un prodotto finale, con scarsa attenzione al fattore di forma, alle dimensioni del progetto o alla facilità di fabbricazione. Tuttavia, alcuni produttori hanno portato i reference design a un livello superiore, creando prototipi di prodotti a tutti gli effetti e persino prodotti completamente realizzabili.

Figura 1: La scheda di sviluppo a microcontrollore Arduino. (Fonte: Shutterstock)

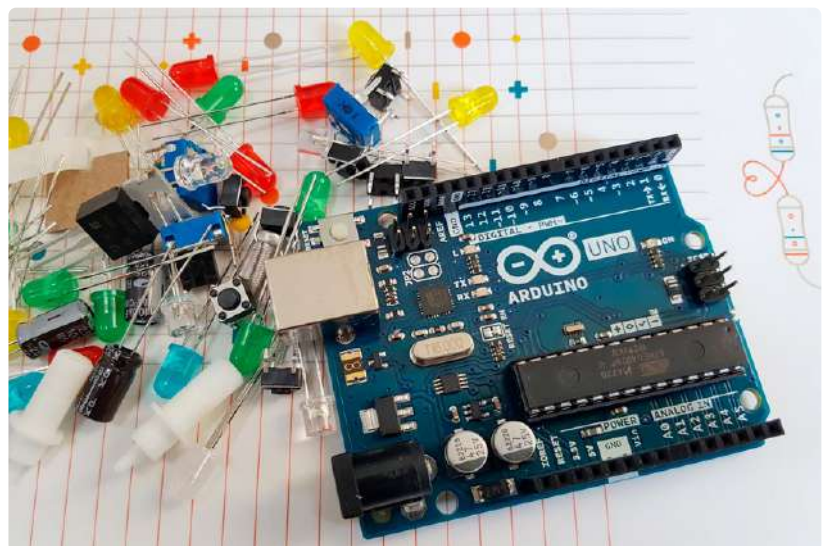




Figura 2: HSP3.0 di Maxim Integrated.

I progetti di riferimento per la piattaforma di sensori sanitari (HSP) [4] di Maxim Integrated (ora parte di Analog Devices) possono essere utilizzati come esempio per tracciare questa evoluzione. La versione iniziale di questi progetti di riferimento era una piccola scheda di sviluppo dotata di un assortimento di sensori (temperatura, pressione, accelerazione, biopotenziale, ecc.) adatti all'uso in applicazioni sanitarie e di fitness configurabili tramite un microcontrollore. I suoi successori, HSP2.0 e HSP3.0, avevano fattori di forma che consentivano di indossarli al polso e assomigliavano molto ad altri wearable disponibili sul mercato (Figura 2).

Ciò ha permesso agli sviluppatori di valutare la funzionalità dei loro sensori in scenari reali. Inoltre, questi progetti hanno consentito agli sviluppatori di software di accedere liberamente alle letture dei sensori (informazioni non facilmente accessibili da altri indossabili per la salute e il fitness). Lo scopo di questo approccio era quello di consentire lo sviluppo di algoritmi di apprendimento automatico e di intelligenza artificiale che avrebbero aggiunto valore all'applicazione.

Dimostrando come il proprio hardware faciliti l'accesso ai dati, Maxim sperava che gli sviluppatori di prodotti

Figura 3: Controllore Logico Programmabile. (Fonte: Shutterstock)



scegliessero alcuni (o tutti) i circuiti integrati della soluzione per sensori di Maxim per utilizzarli nei prodotti. Maxim ha esteso questo approccio fino a sviluppare il MAX HEALTH BAND [5] (da polso) e il MAX ECG MONITOR [6] (fascia toracica), entrambi completamente progettati e costruiti per essere dispositivi indossabili per la salute e il fitness. Anche se non erano destinati alla vendita diretta ai consumatori, le aziende potevano stipulare un accordo con Maxim per avere questi prodotti con il proprio marchio in cambio di un pagamento di royalty. L'offerta di un prodotto completamente funzionante in questo modo, dove tutto il lavoro di sviluppo è già stato fatto, ha il potenziale di attrarre una nuova e più ampia base di clienti commerciali non tecnici. Thingy:91 [7] di Nordic Semiconductor è un altro esempio di piattaforma di sviluppo in cui l'hardware è diventato quasi accessorio rispetto al compito di fornire agli sviluppatori l'accesso ai dati necessari per sviluppare il software e gli algoritmi che consentono di realizzare il valore intrinseco dell'hardware (ma, così facendo, lo rendono la scelta più ovvia per l'uso in nuovi progetti di prodotti che sfruttano questi algoritmi). È probabile che questo approccio venga adottato da un numero ancora maggiore di produttori in futuro.

Maggiore utilizzo delle schede di sviluppo nei prodotti industriali

L'adattamento di schede di sviluppo e SBC per l'uso in prodotti commerciali è diventato sempre più comune, ma un'altra tendenza emergente è l'uso di schede in volumi inferiori ma per applicazioni di valore più elevato, ad esempio in prodotti destinati all'industria come i controllori logici programmabili (Figura 3), che sono soggetti a standard più esigenti rispetto ai loro equivalenti commerciali.

Collaudo delle schede per applicazioni industriali

Molti degli attuali SBC sono diventati modelli completamente verificati perché le parti che contengono sono state originariamente sviluppate per l'uso in prodotti finali e sono state quindi testate e garantite dal punto di vista della qualità. Questo anche perché i progetti open-source sono costantemente rivisti da un esercito di progettisti e programmatori competenti che aggiornano e valutano le schede e il software che utilizzano.

I test delle schede SBC vengono ora eseguiti da aziende di progettazione e produzione di alta qualità e sono sottoposti allo stesso rigoroso livello di controllo della qualità di qualsiasi altro prodotto finale, consentendo loro di ottenere persino le certificazioni CE o FCC. Questo flusso di test può essere facilmente esteso per soddisfare i requisiti di prodotti industriali.

D'altro canto, le schede di sviluppo per microcontrollori fornite dai produttori o da terzi, pur essendo solitamente adatte all'uso in prodotti commerciali, non sono in genere sottoposte agli stessi rigorosi livelli di test richiesti per i

prodotti industriali. Ciò significa che attualmente i produttori non ne raccomandano l'uso immediato (nella loro veste attuale) in queste applicazioni.

Sebbene alcune schede includano componenti di tipo industriale, più spesso sono solo di tipo commerciale e sono progettate per funzionare a temperatura ambiente. I prototipi di schede di sviluppo sono in genere sottoposti a test a temperatura ambiente per diversi giorni o settimane, ma questo varia a seconda del produttore, poiché non esistono standard prestabiliti. Il requisito principale di qualità per i produttori è che le loro schede funzionino in modo affidabile a temperatura ambiente e quindi gli acquirenti devono essere consapevoli che è improbabile che queste schede siano state testate a temperature o umidità estreme. Inoltre, non vengono normalmente testate per sopravvivere alle sollecitazioni associate a vibrazioni o urti intensi.

Di conseguenza, l'obiettivo principale nel determinare la scheda di sviluppo da utilizzare in un'applicazione industriale è quello di ridurre i rischi: I componenti della scheda devono avere il grado di temperatura adeguato. È inoltre opportuno sottoporre a stress test diverse schede contemporaneamente ad alta temperatura per un periodo di giorni. Allo stesso modo, se si prevede di utilizzare una scheda di sviluppo in un prodotto che sarà soggetto a elevata umidità, le schede devono essere valutate in condizioni analoghe. Se una scheda è destinata all'uso in un'applicazione con vibrazioni elevate, deve essere montata in un sistema di prova e sottoposta a test di vibrazione.

Conclusione

Le schede SBC e le schede di sviluppo per microcontrollori offrono alle piccole aziende un modo conveniente per portare rapidamente i loro progetti sul mercato senza dover affrontare le spese di sviluppo di nuovo hardware. Le schede consentono di concentrarsi sull'innovazione del software e, sempre più spesso, sullo sviluppo di algoritmi di apprendimento automatico e intelligenza artificiale. Le SBC e le schede di sviluppo hanno ampliato il loro campo di applicazione ben oltre quello originariamente previsto e hanno avuto un impatto reale sulla storia recente dell'industria elettronica. Continuano a diventare sempre più potenti, intelligenti e reattive, pur rimanendo facilmente accessibili sia agli ingegneri professionisti che agli appassionati di elettronica. ◀

220597-05



Note sull'Autore

In qualità di Technical Marketing Manager di Mouser Electronics per l'area EMEA, Mark Patrick è responsabile della creazione e della diffusione di contenuti tecnici all'interno della regione, contenuti che sono fondamentali per la strategia di Mouser di supportare, informare e ispirare il suo pubblico di ingegneri.

Prima di dirigere il team di marketing tecnico, Patrick ha fatto parte del team EMEA Supplier Marketing e ha svolto un ruolo fondamentale nell'instaurare e sviluppare relazioni con i principali partner di

produzione. Oltre a una serie di posizioni tecniche e di marketing, i precedenti ruoli di Patrick includono otto anni presso Texas Instruments nel supporto alle applicazioni e nelle vendite tecniche.

È un ingegnere "pratico", con una passione per i sintetizzatori e le motociclette d'epoca, e non ha problemi a effettuare riparazioni su entrambi. Patrick ha conseguito una Laurea con lode in Ingegneria Elettronica presso l'Università di Coventry.

LINK WEB:

- [1] Arduino Boards Distributor: <https://elektor.link/MouserArduino>
- [2] BeagleBoard Distributor: <https://elektor.link/MouserBeagleBoard>
- [3] Raspberry Pi Distributor: <https://elektor.link/MouserRaspberryPi>
- [4] Maxim Integrated Distributor: <https://elektor.link/MouserMaxim>
- [5] MAX HEALTH BAND: <https://elektor.link/MouserMaxHealthBand>
- [6] MAX-ECG-MONITOR : <https://elektor.link/MouserMaxECGMonitor>
- [7] Nordic Semiconductor Thingy:91™ Multisensor Prototyping Kit : <https://elektor.link/MouserThingy91>



Arte Floreale

dai Fili a Memoria di Forma

Sculture Cinetiche che Comunicano col Suono

Di Dave Vondle (IDEO)

Da quando sono state scoperte le loro strane proprietà, oltre 60 anni fa, i fili di Nitinol sono stati una soluzione in cerca di applicazione. In questo caso, l'applicazione ispirata è l'arte: fiori che rispondono al suono con luce, suono e movimento.

Quest'anno alcuni di noi in IDEO, un'azienda di progettazione, sono stati invitati a tenere un laboratorio all'Eyeo Festival [1]. Con le persone volevamo costruire qualcosa di fisico, ma che potesse essere controllato tramite codice e con cui sperimentare. La mia collega e collaboratrice, Jenna Fizel, ha fatto un sacco di cose interessanti scrivendo un software che supporta la creazione di oggetti 3D partendo dalla carta e da altri materiali

sottili. Di recente, ho sperimentato il filo di Nitinol, un materiale che può cambiare forma quando viene attraversato dalla corrente. Con il supporto aggiuntivo dei progettisti software di IDEO Derek Olson e YC Sun, abbiamo cercato di capire come realizzare un kit di componenti che permettesse ad altri di creare sculture di carta in movimento (Figura 1).

Nel laboratorio volevamo che le persone se ne andassero con qualcosa di unicamente loro, ma volevamo anche che il prodotto finale del lavoro fosse collaborativo. Questo è stato il primo seminario che abbiamo fatto dal vivo dopo la pandemia, e volevamo che ci fosse una sorta di valore unico per tutti noi nel costruire qualcosa insieme, di persona. Le sculture che abbiamo realizzato avrebbero potuto interagire l'una con l'altra? Come avrebbero potuto comunicare? Abbiamo pensato che se avessero potuto comunicare a livello sonoro, avremmo potuto sentirle suonare insieme. Il workshop avrebbe potuto creare una sinfonia generativa!





Figura 2: Infilatura del filo di Nitinol attraverso i fiori.

I partecipanti al gruppo di lavoro avevano un'ampia varietà di esperienze e conoscenze di elettronica. Costruendo questi fiori, siamo stati in grado di familiarizzare con un ampio spettro di concetti. Alcuni degli aspetti fondamentali sono stati "Iniziare con Arduino" e la legge di Ohm per capire la tensione necessaria a pilotare il Nitinol. Alcuni dei concetti più complessi che abbiamo potuto trattare sono stati la modulazione a larghezza di impulso, la trasformata veloce di Fourier ("FFT") per identificare le frequenze dei fiori vicini e le basi della sintesi audio. Ogni partecipante ha costruito un fiore dipinto a mano, di cui ha disegnato la forma dei petali (Figura 2). L'elettronica dei fiori controlla le luci LED sulla sommità, i suoni emessi dal fiore e il controllo indipendente della posizione dei petali.

Al link [2] è possibile vedere un video di come si è svolta la rappresentazione finale in occasione di un successivo workshop IDEO.

Entrando nei dettagli, illustrerò la progettazione dell'hardware, quindi tratterò del software. Infine, spiegherò come potrete realizzarne uno voi stessi.

Hardware

L'hardware è liberamente basato su una combinazione di Adafruit Trinket M0 [3] e della Sparkfun Electret Microphone Breakout Board [4]. Oltre alle funzionalità di queste schede, abbiamo aggiunto un altro amplificatore per un altoparlante, un circuito buck-boost a 5 V affinché potesse funzionare con una batteria, alcuni LED indirizzabili WS2812 e 8 MOSFET per controllare la corrente nel filo di Nitinol.

Lo schema completo è riportato nella Figura 3. La scheda finita ha l'aspetto mostrato nelle Figure 4a e 4b.

Per il nostro laboratorio, volevamo evitare la saldatura, poiché il Nitinol può essere estremamente difficile da saldare senza materiali e tecniche speciali, quindi

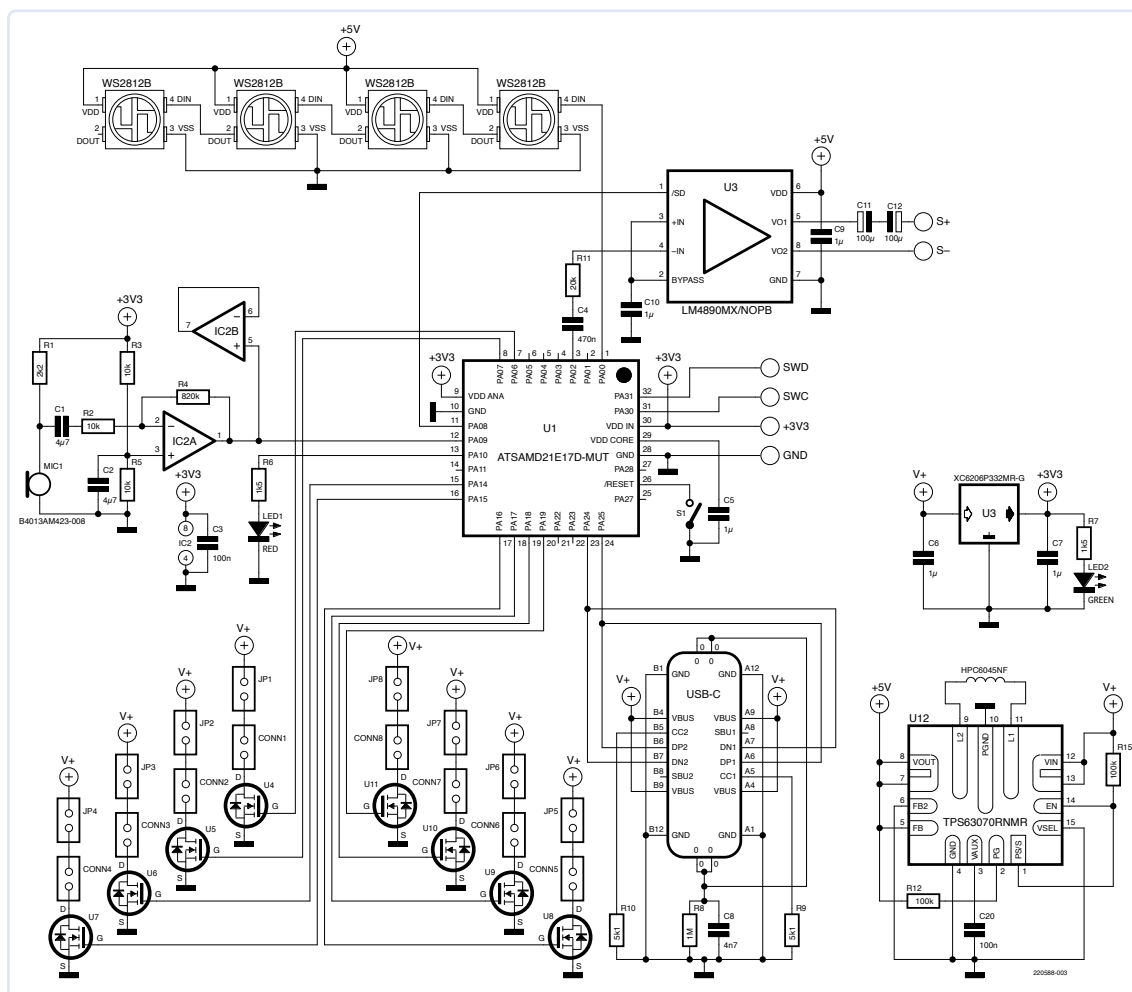


Figura 3: Schema elettrico del fiore Eyeo.

Figura 4a: Scheda finita, lato stelo.

Figura 4b: Scheda finita, lato petali.



abbiamo utilizzato un sistema di inserti filettati e viti per collegare meccanicamente il filo. Ciò ha consentito anche di controllare con precisione il pretensionamento del filo dei petali.

In base alla lunghezza del filo di Nitinol, sono stati scelti i valori delle resistenze montate su zoccolo per limitare la corrente massima. Nel progetto si utilizza un filo di Nitinol da 100 μm [5] con una resistenza di 126 Ω/m e una corrente nominale massima di 200 mA. In base a ciò, è possibile selezionare la resistenza adatta per evitare di danneggiare il filo. Nei test precedenti avevamo omesso queste resistenze, scegliendo di limitare la corrente impostando un rapporto PWM massimo via software. Ma abbiamo riscontrato che durante il processo di programmazione a volte un pin diventava alto, quindi abbiamo risolto il problema con le resistenze fisiche.

È stato predisposto un altoparlante con dei pin da test point saldati sul retro, in modo da poterlo montare sulla parte superiore della scheda. Sul fondo della scheda, al centro, è presente un connettore USB verticale, in modo che il cavo USB possa essere utilizzato come "gamba" del fiore.

Il Microcontrollore

Abbiamo scelto la famiglia di microcontrollori ATSAM21 per la sua caratteristica di essere compatibile con Arduino in un unico chip (la maggior parte delle altre schede Arduino utilizza un altro componente per la funzionalità di comunicazione seriale UART-USB). Tuttavia, la variante ATSAM21E18 utilizzata nel Trinket M0 di Adafruit non era disponibile a causa della carenza di chip. Siamo invece riusciti a trovare i chip ATSAM21E17D, che risultavano disponibili. La differenza principale tra i due chip è la riduzione della memoria flash, da 256 KB a 128 KB. Sebbene questa differenza sia minima, abbiamo dovuto modificare il bootloader di Arduino [6] (fork da `uf2-samdx1`) e creare una nuova scheda associata nell'IDE di Arduino per parlare con questo nuovo chip/bootloader.

Esempio di Codice

Lo sketch Arduino di esempio [7] fornisce una base per far "cantare" i fiori insieme, emettendo suoni udibili dall'altoparlante, ascoltando le note con il microfono e reagendo con il movimento dei petali e la luce dei LED. Vi illustrerò brevemente il funzionamento di tutto questo.

Riproduzione dei Suoni

L'ATSAMD21 dispone di un pin che può essere configurato come DAC (convertitore digitale-analogico). Questa caratteristica è ottima per poter ottenere un'ampia varietà di timbri musicali dal chip. Permette al fiore di essere molto più espressivo di quanto sarebbe stato utilizzando la libreria Arduino Tone [8], la cui funzione `tone()` permette di variare la tonalità di un'onda quadra, ma il DAC che utilizziamo ci permette di controllare la forma d'onda stessa. Utilizziamo la fantastica libreria Mozzi [9], che fornisce un framework per la definizione e la riproduzione di suoni da questo DAC.

Nel codice abbiamo impostato un paio di oscillatori coseno, uno per la frequenza portante principale e uno per il vibrato. Abbiamo anche impostato un inviluppo (altrimenti noto come ADSR - Attack, Decay, Sustain, Release), che ci permette di controllare l'ampiezza del suono nel tempo.

Mentre riproduciamo la forma d'onda audio, il microcontrollore esegue un lookup e calcola il livello di tensione successivo che deve essere inviato dal DAC. Questo avviene a una frequenza di 16,384 kHz, quindi, se abbiamo un codice bloccante o una grande quantità di elaborazioni che avvengono in quel lasso di tempo, possiamo compromettere la creazione di un'onda omogenea. Per questo motivo, il codice prima ascolta, reagisce con la luce e il movimento, poi riproduce il suono a sua volta quando non sono in corso altri processi.

Ascolto del suono

Come già detto nell'introduzione, il fiore ascolta un tono specifico e poi reagisce con un tono proprio. Il segnale del microfono viene inviato a un pin del microcontrollore (che può essere configurato come "ingresso analogico") tramite un amplificatore.

Per far sì che il fiore comprenda il tono o la frequenza dei suoni che sente, utilizziamo il già citato principio della trasformata veloce di Fourier. Si tratta di una trasformazione matematica che prende una serie di campioni nel dominio del tempo (tempo su ampiezza) e la trasforma in una serie di campioni nel dominio della frequenza (frequenza su ampiezza). Questo processo matematico può utilizzare molte risorse e richiede un compromesso tra la granularità della frequenza e la velocità/memoria. Per farlo





funzionare il più velocemente possibile, usiamo la libreria Zero DMA (Direct Memory Addressing) di Adafruit [10] per estrarre i campioni dal microfono in un array e la libreria Zero FFT di Adafruit [11] per eseguire la FFT. Una volta che i dati sono stati elaborati attraverso la FFT, possiamo guardare la frequenza primaria del campione per vedere se corrisponde alla nota che stiamo cercando.

Movimento dei Petali

È possibile vedere un esempio del prototipo cartaceo nella **Figura 5**, oppure una GIF animata che lo mostra in azione in [12]. Il filo è inserito nel petalo in modo asimmetrico. Nella **Figura 6** si può notare che c'è molto più filo sulla superficie superiore del petalo rispetto a quella inferiore. Quando il filo si contrae, la superficie superiore del petalo si restringe, provocando uno sforzo di taglio che lo incurva.

Prima di entrare nel merito del codice del movimento del petalo, soffermiamoci un attimo su come funziona il Nitinol.

La corrente elettrica viene utilizzata per riscaldare il filo di Nitinol oltre una soglia di temperatura (per il nostro filo è di 70 °C) in cui subisce un cambiamento di fase e si contrae. Affinché il filo torni alla sua lunghezza originale, deve essere nuovamente allungato fisicamente.

Il cambiamento di fase segue questo ciclo (vedi **Figura 7**): quando il filo è allungato, si trova nello stato martensitico deformato. La struttura cristallina della martensite è chiamata "cubica a corpo centrato". Quando viene riscaldato, il Nitinol passa a una struttura cristallina austenitica. Questa struttura è chiamata "cubica a facce centrate", cioè "a strati ravvicinati", il che significa che la struttura consente a un maggior numero di atomi di essere impacchettati più vicini l'uno all'altro. Quando il filo si riscalda, la struttura cristallina cambia e gli atomi si riorganizzano in una disposizione a strati ravvicinati, provocando una contrazione del filo.

Come abbiamo detto, il filo deve essere allungato fisicamente. Avevamo bisogno di un modo per far sì che questo avvenisse senza molle complicate o ingombranti. Invece di usare la carta per i petali, abbiamo scelto una "carta da acquerello" in polipropilene con finitura opaca, chiamata YUPO [13]. Utilizzando questo materiale, possiamo sfruttare l'elasticità intrinseca del materiale per agire come una molla e riportare i petali del fiore allo stato aperto dopo la piegatura.

Ogni petalo può essere controllato separatamente ed è collegato tramite un MOSFET a una linea con modulazione a larghezza di impulso (PWM). Nel codice, chiudiamo i petali e poi, poiché sono "spenti" quando li apriamo, liberiamo il nostro processore per riprodurre il suono (anche mentre i petali si stanno ancora aprendo).

Per calcolare le resistenze di protezione per il Nitinol, dobbiamo fare un po' di calcoli. Con la nostra forma a petalo, ci ritroviamo con circa 13 cm di filo che va da punto a punto. Il nostro filo ha una resistenza di 126 Ω/m e una corrente nominale massima di 200 mA. Quindi, la resistenza del nostro filo è $126 \text{ } \Omega/\text{m} \times 0,13 \text{ m} = 16,38 \text{ } \Omega$. Per calcolare la resistenza che vogliamo con una tensione massima di 5 V, usiamo la legge di Ohm per ottenere $V / I = R$, quindi $5 \text{ V} / 0,2 \text{ A} = 25 \text{ } \Omega$. Abbiamo scelto un resistore da 10 Ω in serie con il nostro filo da 16,38 Ω ($10 \text{ } \Omega + 16,38 \text{ } \Omega$ è più di 25 Ω) per assicurarci di non inviare al filo una corrente eccessiva.

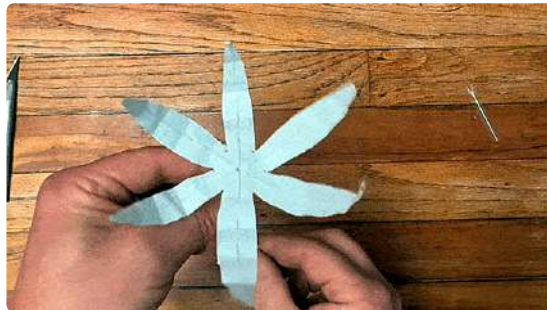


Figura 5: Il primo prototipo in carta realizzato con una cannuccia, uno spago e della carta per verificare l'idea prima di progettare l'hardware.



Figura 6: Primo piano di un petalo infilato.

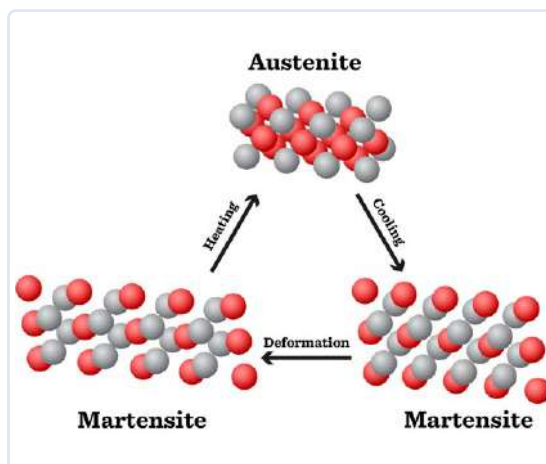
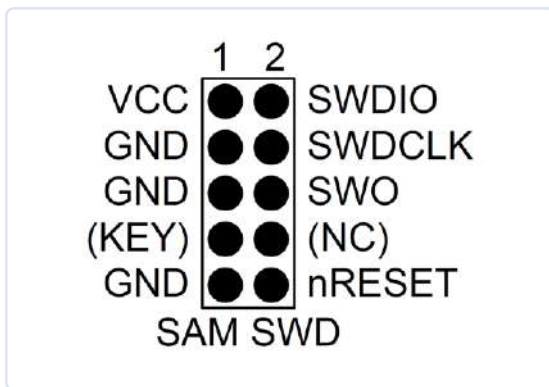


Figura 7: Modello del ciclo della struttura cristallina a memoria di forma del Nitinol.

Figura 8: Piedinatura consigliata del connettore ARM SWD/JTAG. Fonte: Datasheet Atmel.



Luce LED

Il bootloader uf2 che abbiamo modificato supporta i NeoPixel, che sono LED WS2812 a marchio Adafruit. Questo permette ai LED di essere diagnostici e di darci lo stato del bootloader e della connessione USB. Per controllare i LED quando il codice è in esecuzione, utilizziamo la libreria Adafruit NeoPixel [14]. I LED WS2812 sono progettati per essere collegati in daisy-chain. Quando si inviano dati ai LED, ogni LED riceve 8 bit per colore (24 bit in totale). Quando si inviano altri dati, questi vengono spostati al LED successivo, consentendo il controllo indipendente di numerosi LED da un singolo pin.

Come Creare i Propri Fiori

Abbiamo realizzato un sito web interattivo [15] che consente di abbozzare rapidamente la forma di un petalo di fiore da intagliare con una taglierina da tavolo come la Cricut.

I file sorgente Eagle, i file Gerber, CPL e il BOM per realizzare i PCB sono disponibili sulla pagina GitHub del progetto [16]. Con questi file, è possibile rivolgersi a un produttore di schede per ottenere le schede già assemblate. Noi abbiamo usato JLCPCB [17], quindi questi file sono già formattati per i servizi di assemblaggio e PCB di JLCPCB. Si può tentare di assemblarli da soli, ma ci sono diversi IC a passo fine e componenti passivi di dimensioni 0402 che possono rappresentare una sfida. Nel repository di GitHub ci sono anche i link per trovare la carta YUPO, i cavi USB a collo d'oca, il filo di Nitinol, le viti e gli inserti di brocciatura con interno filettato.

Per inserire il bootloader Arduino nel chip, abbiamo usato un programmatore SEGGER J-Link e abbiamo seguito le eccellenti istruzioni su come programmare i bootloader SAMD di Adafruit [18].

Per collegare la scheda al programmatore, abbiamo preso in considerazione la piedinatura SAM SWD dalla Guida all'uso dell'ICE di Atmel [19] (vedi **Figura 8**).

Quello che ci interessa è:

- Pin 1: VCC / Vref
- Pin 2: SWDIO (SWD)
- Pin 3: GND
- Pin 4: SWDCLK (SWC)

Questi devono essere collegati ai pin corrispondenti sulla scheda, indicati nella **Figura 9** come GND, SWC, +3V3 e SWD.

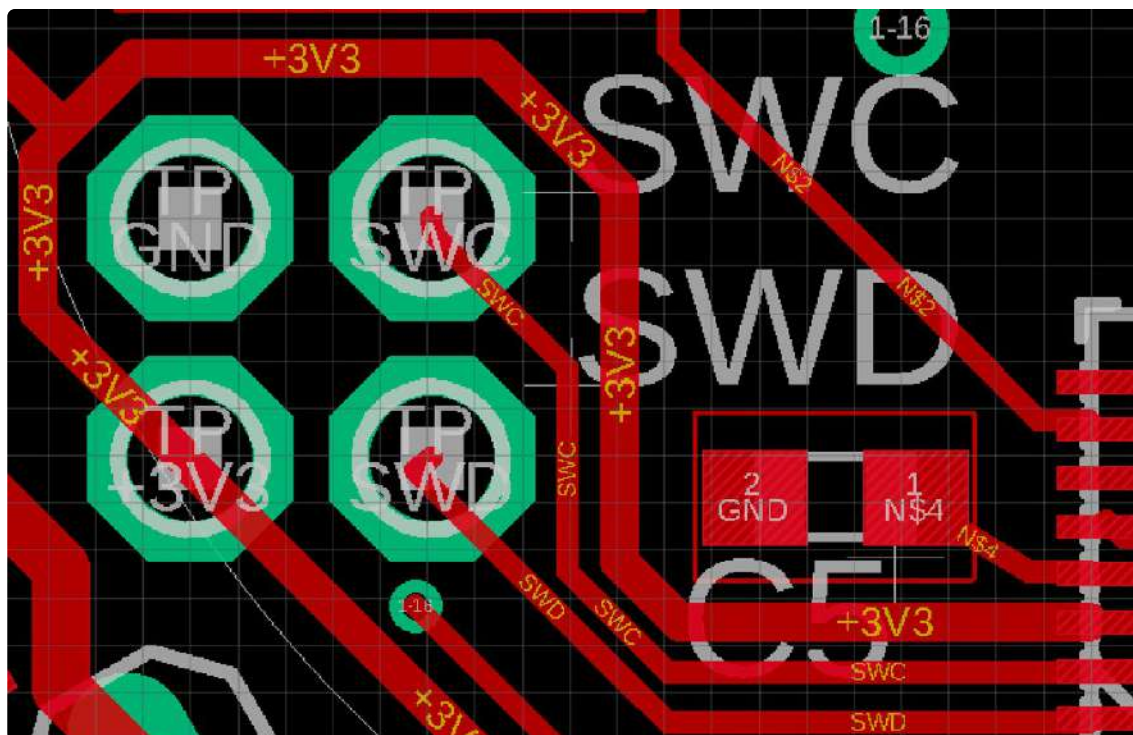


Figura 9: Posizione dei pin SWD sulla nostra scheda.



Poiché dovevamo programmare molte schede, abbiamo costruito un programmatore utilizzando dei pogo pin, per alternare facilmente le schede (**Figura 10**).

Dobbiamo anche inserire a pressione le "ghiere di brocciatura" (inserti filettati) nella nostra scheda in modo da poter avvitare il filo di Nitinol alla stessa. Per creare un attrezzo di montaggio personalizzato per pressare le ghiera di brocciatura sulla scheda abbiamo utilizzato una vecchia maniglia X-ACTO (**Figura 11**).

A seconda degli strumenti di cui si dispone, è probabile che esistano modi migliori per realizzare dispositivi di programmazione e ghiera di brocciatura. Fateci sapere se trovate un modo migliore!

Conclusione

Ci è piaciuto molto realizzare questi oggetti e condividerli con il mondo. Pensiamo che siano una piattaforma divertente per sperimentare una serie di principi diversi in un piccolo pacchetto. Se volete sviluppare questo progetto o volete discuterne con noi, fatecelo sapere! ◀

(220588-05)

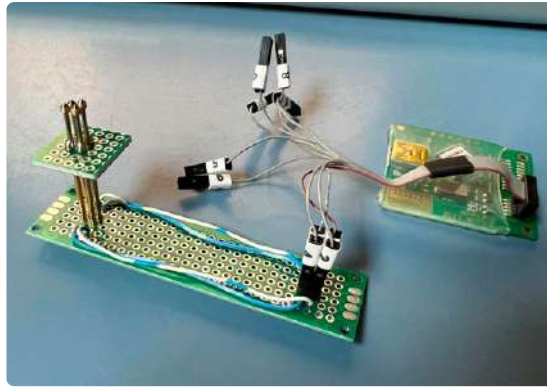


Figura 10: Attrezzo di programmazione personalizzato.



Figura 11: Personalizzazione per realizzare una piccola pressa ad arco.



Domande o Commenti?

Se volete sviluppare questo progetto o volete discuterne con noi, scrivete a @ideo su Instagram, inviatemi una email a dvondle@ideo.com o contattate il team editoriale di Elektor a editor@elektor.com.

Note sull'Autore



Dave Vondle è "Director of Experimentation and Publishing" presso IDEO. Lavora per creare prodotti ed esperienze ben realizzati, progettando e facilitando il loro sviluppo. Preferendo non limitarsi a un unico ruolo, Dave si muove tra

la guida di progetti, la codifica, la progettazione di interfacce, la costruzione di prototipi e la progettazione di circuiti. Prima di approdare a IDEO, ha conseguito una laurea in Ingegneria Elettrica presso la Brown University e ha frequentato corsi presso la Rhode Island School of Design per soddisfare i suoi interessi creativi. Nel frattempo, sta lavorando alla costruzione di una serie di oscilloscopi analogici X-Y a tubo catodico in formato Eurorack. Lo trovate su Instagram su @ideo e @vondle_synths.

LINK WEB:

- [1] Eyeo Festival: <https://eyeofestival.com>
- [2] Nitinol Flowers Demonstration: <https://youtu.be/MBdbXO-WHJ4>
- [3] Adafruit Trinket M0: <https://adafruit.com/product/3500>
- [4] Sparkfun Electret Microphone Breakout: <https://sparkfun.com/products/12758>
- [5] Muscle Wires Actuator Wire 100 µm: <https://elektor.link/musclewires>
- [6] Modified Arduino bootloader: <https://github.com/ideo/eyeo-flower/tree/main/Bootloader>
- [7] Eyeo Flower Example Sketch: github.com/ideo/ArduinoCore-samd/tree/master/libraries/Eyeo_Flower
- [8] Arduino Tone library: <https://arduino.cc/reference/en/libraries/tone>
- [9] Mozzi: <https://sensorium.github.io/Mozzi>
- [10] Adafruit Zero DMA library: <https://arduino.cc/reference/en/libraries/adafruit-zero-dma-library>
- [11] Adafruit Zero FFT library: <https://www.arduino.cc/reference/en/libraries/adafruit-zero-fft-library>
- [12] Nitinol Flower Paper Prototype (animated GIF): <https://elektor.link/gif/nitinol-flower-paper-prototype.gif>
- [13] YUPO synthetic paper: <https://yupousa.com/what-is-yupo>
- [14] Adafruit NeoPixel library: <https://arduino.cc/reference/en/libraries/adafruit-neopixel>
- [15] EYEO Flora interactive site: <https://observablehq.com/@jftesser/eyeo-flower>
- [16] EYEO Flower GitHub repository: <https://github.com/ideo/eyeo-flower>
- [17] JLCPCB circuit board manufacturer: <https://jlcpcb.com>
- [18] How to Program SAMD Bootloaders: <https://learn.adafruit.com/how-to-program-samd-bootloaders>
- [19] Atmel-ICE Debugger User Guide: <https://elektor.link/AtmelICEUserGuide>

Afferra il nuovo

Hardware Arduino!



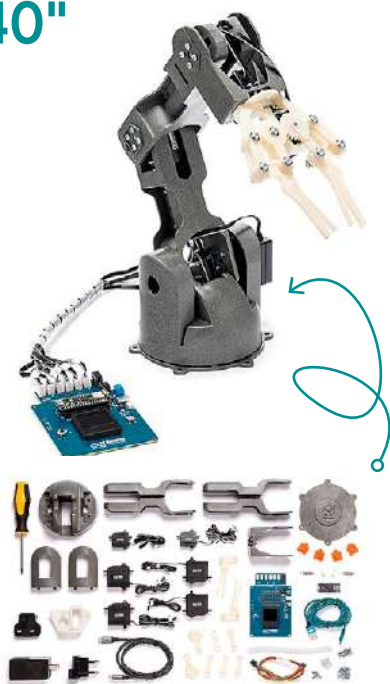
Non c'è niente che ci entusiasmi di più che mettere le mani su hardware nuovo, perciò questa collaborazione con Arduino è stata un vero piacere! Volete sperimentare voi stessi provando dal vivo? Elektor mette a vostra disposizione tutti i prodotti presentati in questo numero!



Braccio Robotico "Arduino Braccio++ RP2040"

La prossima evoluzione del robot Tinkerkit Braccio si chiama Arduino Braccio ++, un braccio robotico nuovo di zecca progettato per utenti avanzati. Arduino Braccio ++ può essere assemblato in diversi modi per svolgere molteplici compiti, come spostare oggetti, utilizzare una telecamera per seguire i vostri movimenti o collegare un pannello fotovoltaico per seguire il movimento del sole. Arduino Braccio ++ vi offre fin da subito una moltitudine di possibilità di espansione, tra cui una nuova Carrier Braccio con schermo LCD, nuovi servomotori RS-485 e un'esperienza totalmente migliorata.

www.elektor.com/20174



Arduino Pro Portenta H7



Portenta H7 vi permette di realizzare il vostro prossimo progetto smart. Avete mai desiderato una casa automatizzata? O un giardino intelligente? Ebbene, con le schede compatibili con Arduino IoT Cloud adesso è facilissimo. Ciò significa che potrete collegare i dispositivi, visualizzare i dati, controllare e condividere i vostri progetti da qualsiasi parte del mondo.

www.elektor.com/19351



PID-based Practical Digital Control with Raspberry Pi and Arduino Uno

www.elektor.com/20274

Arduino Pro Nicla Vision

Nicla Vision combina un potente processore STM32H747AI16 Dual ARM Cortex-M7/-M4 IC con una fotocamera a colori da 2 MP che supporta TinyML, oltre a un sensore di movimento intelligente a 6 assi, un microfono integrato e un sensore di distanza.

www.elektor.com/20152



Arduino Pro Portenta X8



Portenta X8 è un potente SOM di livello industriale con sistema operativo Linux precaricato, in grado di eseguire software indipendente dal dispositivo grazie alla sua architettura modulare a container. In pratica si tratta di due prodotti industriali in uno, che combinano la disponibilità di librerie/competenze di Arduino con la distribuzione Linux basata su container.

www.elektor.com/20270



Arduino Pro Nicla Sense ME

Un nuovo standard per soluzioni di rilevamento intelligente.

www.elektor.com/20327

Shield Portenta Vision (Ethernet)

La Shield Portenta Vision offre funzionalità di livello industriale al vostro Portenta. Visione computerizzata professionale, rilevamento audio direzionale, Ethernet e JTAG per Arduino Portenta.

www.elektor.com/19511

Shield Portenta Vision (LoRa®)

Questo add-on hardware di Portenta consente di eseguire applicazioni di computer vision embedded, collegandosi in modalità wireless tramite LoRa® al Cloud di Arduino o alla propria infrastruttura.

www.elektor.com/20332

Arduino Pro Portenta Breakout

La scheda Portenta Breakout è stata progettata per aiutare gli ingegneri hardware e i maker a prototipare e testare le connessioni e la capacità dei dispositivi nella famiglia di schede Portenta.

www.elektor.com/20341



Arduino Pro Portenta Max Carrier

Prototipate facilmente le vostre applicazioni Portenta. Implementazione a tempo zero. Max Carrier trasforma i moduli Portenta in computer a scheda singola o in reference design che abilitano l'AI all'edge di rete per applicazioni industriali, automazione degli edifici e robotica ad alte prestazioni.

www.elektor.com/20271

Arduino Uno Rev3

Il classico microcontrollore AVR ad alte prestazioni e basso consumo. La Uno è la scheda migliore per iniziare a lavorare con l'elettronica e la programmazione, semplicemente la scheda più robusta per iniziare a smanettare con la piattaforma Arduino.

www.elektor.com/15877



Kit Arduino Make-Your-Uno

Un nuovo kit che include un **UNO DIY** di tipo "through hole" con tutti i componenti per realizzare il tuo UNO e il synthesizer basato su di esso!

www.elektor.com/20330



Shield Arduino Ethernet 2

www.elektor.com/19941



Arduino Sensor Kit Base

www.elektor.com/19944

Arduino Nano

Arduino Nano è una scheda piccola, completa e con funzionalità di breadboard, basata sull'ATmega328 e racchiusa nel più piccolo fattore di forma disponibile di 18x45 mm!

www.elektor.com/17002

Arduino Nano RP2040 Connect

Arduino Nano RP2040 Connect è una scheda Arduino basata su RP2040 e dotata di Wi-Fi, Bluetooth, microfono e sensore di movimento intelligente a sei assi con funzionalità di AI.

www.elektor.com/19754

Arduino Nano 33 BLE Sense

Portate nelle vostre mani la potenza dell' AI grazie al più potente processore nRF52840 e a una serie di sensori integrati, con la possibilità di eseguire applicazioni di Edge Computing (AI).

www.elektor.com/19936

Supporto dei Rivenditori Arduino

Questa edizione dell'Elektor Magazine, a cura di Arduino, è stata resa possibile grazie al supporto di questi membri della comunità dei rivenditori Arduino.

Consultateli per qualsiasi vostra esigenza relativa ad Arduino!

GOTRON
AALST GENT HASSELT



www.gotron.be

HELLAS
digital



www.hellasdigital.gr

TINYTRONICS



www.tinytronics.nl

 **Paradisetronic.com**



www.paradisetronic.com

 **Techni Science.**



www.techniscience.com

 **WHADDA**
MADE BY VELLEMAN



www.whadda.com

 **KUBII**



www.kubii.fr

GO TRONIC
ROBOTIQUE ET COMPOSANTS ÉLECTRONIQUES



www.gotronic.fr

Consultate uno di questi rivenditori per qualsiasi vostra esigenza relativa ad Arduino!