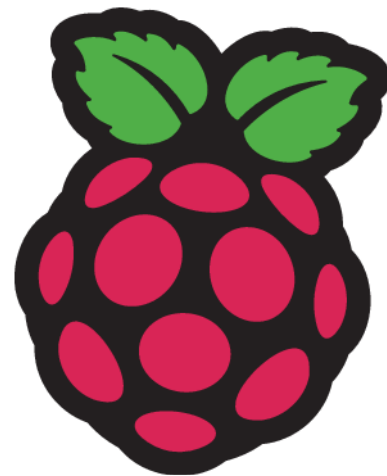


The MagPi



Numero 121 | Ottobre

2022

magpi.cc
raspberrypi.org

La rivista ufficiale Raspberry Pi
tradotta in italiano per RaspberryItaly 

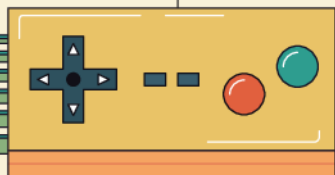
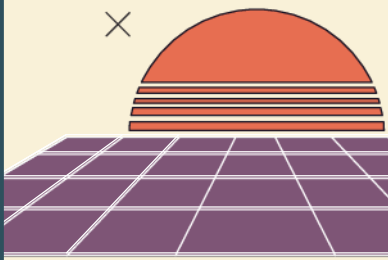
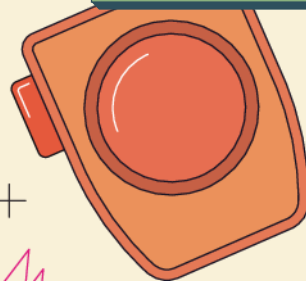
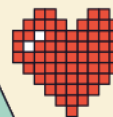
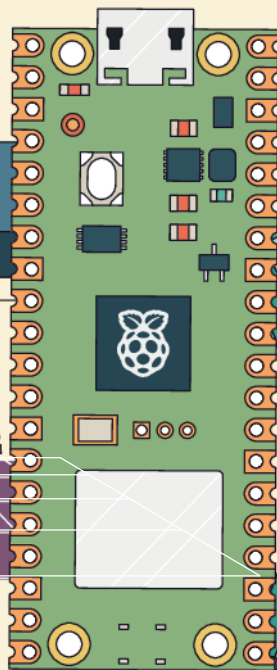
RETRO GAMING CON PICO W

- > Classici Palmari
- > Giochi Arcade
- > Computer Anni '80



FAI UN
MINI
MAGIC
MIRROR

VETRINA
PICO
POCKET
GAMER



Pico Pocket Gamer

Una console portatile che sia semplice da costruire anche per un principiante. Questo è il progetto di **Nicola King's**



MAKER

Grgo Mariani

Grgo è uno sviluppatore software che lavora per IBM a Böblingen, in Germania, ed è un grande fan di Raspberry Pi. magpi.cc/picopocket

Ricordando un gioco elettronico retrò chiamato **Lights Out** (un puzzle con una griglia di luci), Grgo Mariani si rese conto che poteva farne una versione aggiornata con un Raspberry Pi Pico inutilizzato.

"Ricordo come ci giocavo mentre visitavo i parenti e pensavo ad elaborare una strategia perfetta per risolvere ogni nuovo enigma. Da quando recentemente sono diventato padre, volevo che anche mio figlio avesse questa esperienza, ma sfortunatamente, non avrei trovato un negozio dove acquistare questo prodotto", racconta. Questa è stata la spinta di cui aveva bisogno per iniziare a creare, e il risultato è una piccola macchina da gioco piuttosto divertente che è semplice a sufficienza per un principiante.

Benvenuti nuovi arrivati

Poiché l'hardware comprende solo un Pico, un display touchscreen 320×240 e cinque pulsanti meccanici, Grgo vede questa costruzione come molto realizzabile. Tutto ciò che serve è collegare diversi pin del Pico al display e ai pulsanti meccanici. "Niente resistenze, niente condensatori, niente batteria, niente alimentatori esterni... lo chiamerei livello principiante",

conferma. Sebbene abbia progettato una PCB personalizzata per ottimizzare la sua mini console, dice che è facoltativa.

Per quanto riguarda il software, ha creato un semplice motore per gestire i pulsanti e l'input del touch, aggiornare lo stato del gioco e disegnare nel frame buffer in un ciclo continuo. "Ogni gioco condivide un frame buffer 320×240 per il disegno e un buffer di texture 64×64. C'è anche uno spazio di 16kB utilizzato per la memoria dello stato del gioco", spiega.

Finora ha creato 14 giochi per il dispositivo, comprese le sue versioni di Snake, Pong e, naturalmente, Lights Out, e ha adottato un approccio flessibile allo sviluppo dei giochi. "Il progetto finale è diventato molto più di quel singolo gioco, dato che ho davvero adottato un approccio agile durante la costruzione.

Ha scritto lui stesso la maggior parte del codice open source e adattato alcune parti di altri progetti – in particolare per la gestione della comunicazione con lo schermo. "Ho anche adattato l'algoritmo del ray casting [per] il mio gioco Labyrinth. Sottolinea che, dal punto di vista del codice, non sta usando nessuna funzionalità avanzata di Pico.

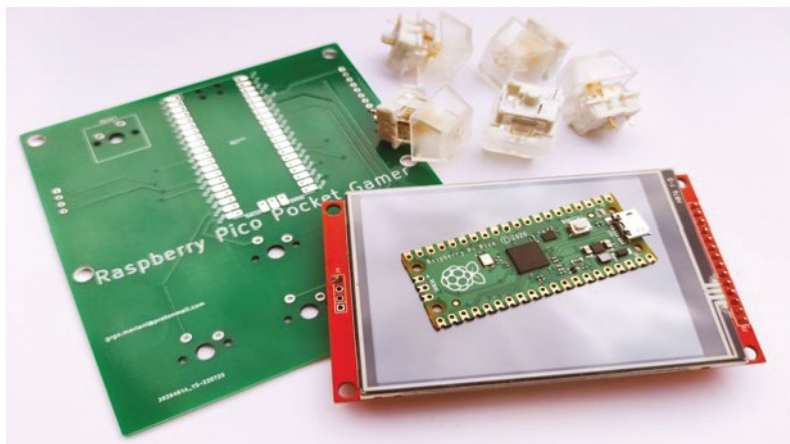
Oltre a trovare il tempo, come nuovo genitore, da spendere per il suo progetto, Grgo ha scoperto che l'elemento più difficile è stata la scelta del display giusto e ri-cablare costantemente la breadboard per adattarla ai nuovi pinout. "Mi ci sono voluti diversi mesi per trovare il display giusto... anche lo sviluppo del PCB è qualcosa che non avevo mai fatto prima, quindi ho dovuto imparare da zero."

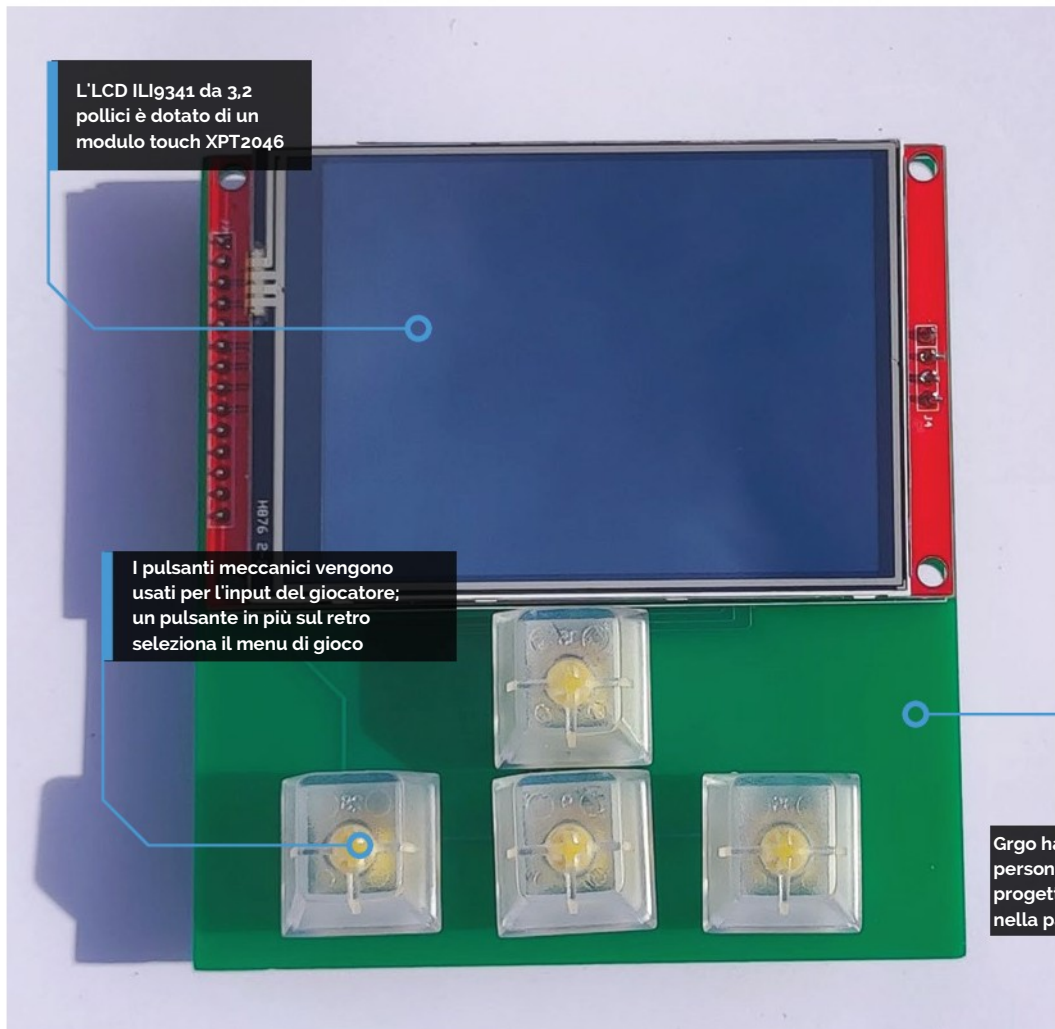
Dai un'occhiata al suo video su YouTube ([magpi.cc/picopocketyt](https://www.youtube.com/watch?v=9j8k8k8k8k)), che mostra come ha costruito il dispositivo.

Feedback favorevole

Il feedback ricevuto dal Pico Pocket Gamer è stato super positivo e Grgo è stato travolto dalla reazione di alcuni membri della comunità Reddit. "Alcune persone già mi scrivono in privato per chiedermi come possono replicare il progetto. Mi dà gioia pensare che potrei aver acceso una

▼ Tutti i componenti del progetto, pre-assemblaggio, abbastanza semplici da mettere insieme



**In BREVE**

- > Ci sono volute circa due settimane a Grgo
- > ...per scrivere il codice principale e i primi sei giochi
- > Stima il costo della costruzione a circa 40€
- > Ha ricevuto richieste dagli amici per costruirne loro uno!
- > Trovi tutti i dettagli, tra cui la progettazione del PCB, su: **magpi.cc/picopocket**

scintilla in qualcuno per usare il proprio Pico con qualche nuovo progetto che ha in mente, per una prova.

Per quanto riguarda i progetti futuri, Grgo non ha nessun piano concreto ancora, ma conferma: "farò qualcosa con il Pico W, questo è sicuro.

“ Il progetto finale è diventato molto più di un gioco singolo ”

" Una cosa che gli piacerebbe davvero fare è implementare alcuni degli algoritmi da *Graphics Programming Black Book* di Michael Abrash Black e usarli per creare un semplice gioco 3D. Ha in programma di aggiungere più giochi a questa console, ed è sostenuto nello slancio dalla

creazione di ogni gioco: "A ogni gioco finito, c'era questa spinta a costruirne solo uno in più". E sicuramente questo è un sentimento con il quale molti maker possono entrare in empatia.



▲ Esecuzione di un gioco su Pico Pocket Gamer: può persino gestire la grafica 3D



RETRO GAMING

con Raspberry Pi Pico e Pico W

Che videogiochi puoi far girare su un microcontrollore da 6€?



AUTORE

K.G. Orphanides

KG. è una sviluppatrice e collezionista di software con un amore costante per i computer vintage.

magpi.cc/MightyOwlbear

Da Dizzy a Doom, Raspberry Pi Pico e Pico W stanno tenendo testa a sistemi di emulazione molto più costosi e sono

pronti per giocare non appena accesi. Con il lancio, a luglio 2022, del Raspberry Pi Pico W, equipaggiato con funzionalità di rete wireless, c'è qualche altra funzionalità da sfruttare.

Sono stati portati su Pico numerosi emulatori maturi e altamente capaci, consentendo di eseguire enormi blocchi di codice dell'era dei computer a 8 bit e delle console, con kit hardware e schede di espansione che facilitano il collegamento di monitor, controller, memoria su scheda SD e audio di alta qualità.

A differenza degli emulatori di computer più pesanti, progettati per computer e in grado di far funzionare quasi tutto, se dedichi abbastanza tempo frugando nei menu, gli emulatori RP2040 diventano sistemi embedded, monouso, reattivi e ad avvio rapido, proprio come l'hardware originale che stanno emulando.

Emulare è cosa buona e giusta, ma puoi anche usare il Pico per portare i nuovissimi giochi homebrew su piattaforme hardware retrò, dando nuova vita alle amate classiche console. Abbiamo mostrato due diverse flash cart, per Nintendo Game Boy e N64, che utilizzano Pico con PCB personalizzati, che ti permettono di scrivere le cartucce console retrò a un costo molto inferiore rispetto a quelle dedicate commerciali.



▲ Riproduci gemme indie dei moderni sviluppatori ed editor C64, come Honey Bee di Psytronik



▲ Rivivi i cruenti giorni di gloria di Doom, amato da una generazione intera di giocatori

Se preferisci lavorare con hardware nuovo, c'è il Pico "GameBoy" di YouMakeTech, una console portatile a colori che porta stampata in fronte la sua ispirazione ma che, in effetti, è progettata per eseguire e ispirare lo sviluppo di nuovi giochi in MicroPython o C.

Questo piccolo microcontrollore può persino giocare a Doom, grazie all'impegnativo porting e al lavoro di ottimizzazione di Graham Sanderson, responsabile interno a Raspberry Pi dell'SDK Pico e appassionato di giochi classici, che è anche responsabile del port del BBC Micro emulator (b-em) su Pico.

Poiché quest'anno ricorre il 50° anniversario della prima versione arcade di Pong Atari del 29 novembre 1972, abbiamo incluso questo minimalista classico ping pong, illustrando un progetto che ti farà creare una colorata incarnazione portatile del gioco, e dimostrando come hostare un gioco di Pong sulla tua rete

Locale, utilizzando Raspberry Pi Pico come un piccolo server web. Siamo certi che il futuro di Pico W comprenda giochi in rete ancora più ambiziosi.

“ Siamo certi che il futuro di Pico W comprenda giochi in rete ancora più ambiziosi ”

Unisciti a noi in un tour vorticoso di alcuni dei migliori progetti di gioco su Pico. Che tu sia interessato all'hacking hardware, allo sviluppo software, o semplicemente a unire un'esperienza di gioco completamente funzionante all'hardware più conveniente in circolazione, c'è qualcosa qui di tuo interesse per Pico.



Gioca a Doom sul Pico

Parliamo con Graham Sanderson, responsabile dell'SDK del Pi Pico, del porting accurato di Doom in tutto il suo splendore su RP2040

Graham Sanderson, responsabile dell' SDK del Pico a Raspberry Pi e performance architect di giorno, ha accettato la sfida di rendere Doom eseguibile sul microcontrollore nel 2021. Il gioco sarebbe stato rilasciato nel marzo 2022 dopo sei mesi di sviluppo, ma la prima apparizione pubblica del gioco è avvenuta un anno prima, quando Sanderson ha twittato: "RP2040 Doom deve esistere, ma se lo faccio, deve eseguire tutta la demo WAD correttamente."

Sanderson è cresciuto con il Sinclair ZX81 e il BBC Micro. Dice che la parte più divertente del

suo lavoro sull'SDK RP2040 forse è stata "prendere più di 30 anni di sviluppo e applicarli all'ambiente vincolato del microcontrollore, che ha una serie di vincoli simili agli home computer che avevo da bambino".

Ha basato RP2040 Doom sul Chocolate Doom (chocolate-doom.org) port dell'originale codice sorgente, a sua volta progettato per rimuovere alcune delle limitazioni legate alla versione DOS originale.

La sfida più grande, dice, è stata adattare l'intera demo da 4 MB, i, insieme al gioco stesso, alla memoria flash da 2 MB di Pico. "Il file di dati del gioco Doom originale (DOOM1.WAD) è grande 4 MB e ho incluso tutto: tutta la grafica, livelli, musica, suono, schermate iniziali, networking multiplayer – tutto il pacchetto."

Particolarmente impressionante è uno strumento chiamato whd_gen, parte del codice RP2040 Doom, che converte e comprime i file WAD in un formato personalizzato che riduce le dimensioni dei file dei livelli fino al 57% nel caso di DOOM1.WAD. WHD, che sta per "Where's Half the Data?" (Dove sono metà dei dati?), è particolarmente efficace nel ridurre le esigenze grafiche di Doom, dettagliate nell'articolo dietro-le-quinte di Sanderson sulle tecniche utilizzate per creare questo port.

Progetti di porting su Pico

Doom non è affatto la prima avventura di Sanderson nel porting di giochi classici su RP2040: "La prima cosa che ho portato è stata in realtà un emulatore ZX Spectrum. È nella mia lista di open source; è solo un po' in disordine. Questo è stato effettivamente sviluppato mentre stavamo ancora sviluppando l'RP2040 su FPGA e



▶ Le connessioni di ingresso e uscita sono ben gestite da Pico VGA Demo Base di Pimoroni



Attenzione! Copyright

I file dei videogiochi sono protetti dalla legge sul diritto d'autore. Assicurati di usare ROM che sono state rilasciate con il consenso del proprietario, o giochi homebrew moderni progettati per essere condivisi. Ci sono molte opzioni legali.

magpi.cc/legalroms



Quindi era limitato a un clock di sistema di 48 MHz. Inoltre, a un certo punto, girava con meno memoria”.

L'emulatore ZX Spectrum non aveva un ciclo accurato, ma questo non aveva molta importanza: le prestazioni della maggior parte dei giochi Spectrum erano accettabili. Ma questo non si può applicare a tutti i sistemi di emulazione retrò.

«Data la ritrovata libertà di un clock superiore,» – la versione in produzione di Raspberry Pi Pico ha una velocità di clock massima di 133MHz – “ho pensato di provare a portare un emulatore della BBC dove è fondamentale che tutto accada esattamente al momento giusto”.

Sanderson dice che lo fa per la sfida, per vedere se è possibile. Questo è un sentimento che sentirai da molti programmatori e hacker hardware con cui abbiamo parlato per questo speciale.

“Di solito inizio con un'altra base di codice, e ovviamente non sono mai progettati per funzionare su qualcosa di limitato. [La] prima cosa è generalmente separarli

Lo fa per la sfida

dall'idea che cercare in tabelle grandi molte centinaia di kB, sia una buona idea, e [io] generalmente devo riscrivere porzioni abbondanti con un nuovo metodo [o] approccio e rifare determinate parti in linguaggio assembly”.

RP2040 Doom mostra l'enorme potenziale di un piccolo microcontrollore, tanta pazienza e determinazione, e il valore di tre decenni di esperienza.

Ma non è necessario iniziare con quel livello di esperienza per lavorare ai propri progetti Raspberry Pi Pico. E se preferisci usare il lavoro di altre persone come piattaforma per il tuo, progetti di livello superiore, o come base per un bel progetto, ci sono molti kit e programmi per divertirti e con cui giocare.

Scarica il tuo Doom

È possibile scaricare i binari di RP2040 Doom su magpi.cc/rp2040doom, inserisci la modalità BOOTSEL e copia il file UF2 che desideri. Per eseguirlo, Pico dovrà essere connesso ad una scheda grafica RP2040 compatibile con vga-board. Il file da 4 MB è per altri dispositivi basati su RP2040, con più memoria flash.

È progettato per funzionare con un Pico VGA Demo Base di Pimoroni (magpi.cc/vgademobase),



che è funzionalmente identica alla “VGA, SD card, and audio demo board” descritta nel documento di Raspberry Pi “Hardware Design with RP2040” (magpi.cc/rp2040hardware). Sanderson osserva: “E’ molto più facile usare la VGA Demo Base, però!”

Anche se questo è il modo più semplice per collegare display, controlli e audio, non è l'unico modo. In alcuni video YouTube di Sanderson sul progetto, vedrai una versione breadboard prototipo, con i GPIO video collegati ad un connettore VGA tramite un DAC a resistenza e quelli audio a una scheda DAC PCM5102 I2S.

Puoi leggere un resoconto completo di Graham Sanderson sull'ottimizzazione su magpi.cc/makingrp2040doom.

▲ RP2040 Doom è stato progettato per essere fedele all'originale DOS in grafica, suono e reattività



Vetrina hardware: RetroVGA

Un vero e proprio computer retrò in tasca, con ingresso joystick DB9 e uscita VGA

Uno dei progetti in kit per Pico più impressionanti in giro è RetroVGA di Peter 'Bobricius' Misenko, un multi-retro computer basato su Raspberry Pi Pico che equipaggia il microcontrollore con uscita VGA, porta joystick DB9, tastiera QWERTY, uscita audio da 3,5 mm, altoparlante piezo integrato, porta USB per dispositivi di input sugli emulatori supportati e uno slot per scheda SD per l'archiviazione.

Il tutto misura solo 10 cm quadrati, abbastanza piccolo da stare in una tasca del cappotto se, per ogni evenienza, devi portare in giro un micro retro computer. Se usi un joystick DB9, avrai bisogno di ridurre la lunghezza del cavo a circa 30 cm, o registrerà false pressioni dei tasti.

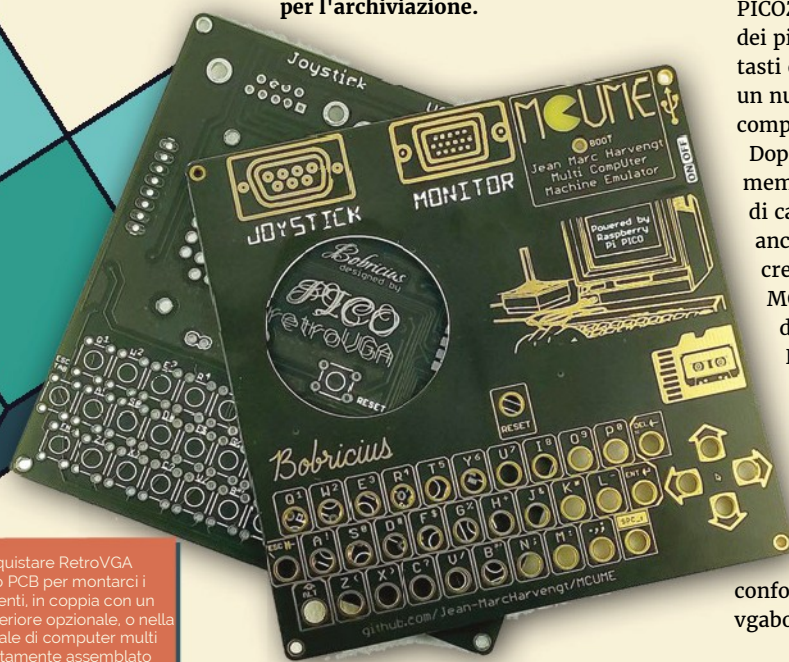
La tastiera non è del tutto completa, mancano ad esempio i tasti funzione, necessari in alcuni titoli C64. Il dispositivo gemello di RetroVGA, il PICOZX (magpi.cc/picozx128) cambia il numero dei pin GPIO dedicati all'uscita VGA aggiungendo tasti extra, potrebbe essere un problema, ma con un numero limitato di connessioni GPIO, qualche compromesso deve essere fatto.

Dopo aver selezionato il gioco dalla scheda di memoria SD di RetroVGA, la tastiera ti permette di caricarlo (LOAD) e eseguirlo (RUN), ci sono anche tasti per giocare. RetroVGA è stato creato per la prima volta per funzionare con MCUME, il Multi CompUter Machine Emulator di Jean-Marc Harvengt e la libreria display PicoVGA di Miroslav Nemecek. Adesso supporta anche l'emulatore pico-zxspectrum di Phil Scull, più ricco di funzionalità.

Standard grafico PicoVGA

Harvengt e Nemecek sono coinvolti da vicino nello sviluppo. Il cablaggio della uscita video della scheda RetroVGA è conforme allo standard PicoVGA e non allo standard vga-board predefinito di Pico SDK.

▲ Puoi acquistare RetroVGA come singolo PCB per montarci i tuoi componenti, in coppia con un pannello superiore opzionale, o nella sua forma finale di computer multi retrò completamente assemblato





▲ Con controlli, suono e spazio di archiviazione integrati, RetroVGA è un progetto compatto per gli ultimi giochi per il più vecchio dei computer

Utilizza diverse connessioni da GPIO a VGA, che sono documentati nel repository del progetto su magpi.cc/picovga.

La documentazione completa di RetroVGA è disponibile su magpi.cc/retrovgadocs per

capace, supportando gli stati di salvataggio rapido, così come tastiere e joystick USB.

Anche quando si utilizza un emulatore multi-dispositivo come MCUME, utilizzerai un emulatore alla volta. I binari di MCUME sono disponibili per il download all'indirizzo magpi.cc/picoretrovga e i file UF2 per pico-zxspectrum sono tutti disponibili su magpi.cc/zxuf2.

Mentre alcuni dei progetti in questa panoramica di giochi retrò basati su Pico sono ancora nelle fasi sperimentali, che richiedono assemblaggi su breadboard o di avere PCB su misura, puoi semplicemente acquistare un kit RetroVGA su magpi.cc/retrovga.

A partire da 10€ per il PCB principale nudo, 30€ per il PCB e pannello superiore con supporto tastiera oro, fino a 108€ per un'unità completamente assemblata e pronta per la programmazione con tutti i componenti, ci sono molte opzioni disponibili, a seconda delle tue esigenze e di quante saldature sei pronto a fare.

“Ci sono molte opzioni disponibili, a seconda delle tue esigenze e di quante saldature sei pronto a fare”

▼ Per i migliori risultati di input, dovrai ridurre la lunghezza del cavo del joystick DB9 a 30 cm

aiutarti ad assemblare RetroVGA e lo mantengono attivo e funzionante con una selezione di emulatori o la libreria PicoVGA, e ci sono abbastanza dettagli da mantenere gli sviluppatori belli svegli.

Attualmente puoi usare MCUME per emulare lo ZX81, ZX Spectrum, Atari 800, C64, VIC20, Atari 2600, Odyssey/Videopac, Colecovision, e Atari 5200. MCUME fornisce una esperienza di emulazione efficace, ma essenziale. Può leggere i file di gioco dalla scheda SD, dandoti accesso al ricco mondo homebrew e indie di C64 e Spectrum. L'emulatore pico-zxspectrum è ancora più



Hacking hardware

Quattro progetti hardware di retrogaming da testare

PICOCART64

Il PicoCart64 è una flash cart per il Nintendo N64: un dispositivo che può contenere il codice per essere letto dall'hardware della console originale, consentendo di caricare i giochi e il software su un vero N64.

Può avviare gli homebrew N64, il che lo rende un modo avvincente per accedere in tempo reale al ricco ecosistema di giochi post-mercato della console hardware, dato che le flash cart N64 più popolari sono sia costose che spesso difficili da trovare.

Il creatore Konrad Beckmann afferma di aver iniziato il progetto per vedere se fosse possibile utilizzare Raspberry Pi Pico invece degli FPGA (field-programmable gate array) utilizzati nella maggior parte delle flash cart disponibili in commercio, hardware già costoso che è diventato ancora più raro di fronte alla carenza globale di chip.

PicoCart64 Lite è il primo prototipo flash cart funzionale del progetto, in grado di caricare giochi N64, file di test e homebrew. Richiede un singolo Raspberry Pi Pico, più un MOSFET e una resistenza, e costa meno di 10\$.

Il repository GitHub del progetto include elenchi di componenti, schemi PCB e, utile per chi è nuovo al mondo dei PCB, un collegamento a un servizio relativamente poco costoso che farà proprio questo.

Troverai sia gli schemi hardware che il software nel ramo sviluppo del repository PicoCart e una discussione attiva tra collaboratori e hacker hardware che implementano la tecnologia sul canale Discord Dubious Technology.

Beckmann afferma che PicoCart64 sta "abbassando la barriera, dando la possibilità alle persone di creare i propri giochi ed eseguirli sull'hardware attuale." Trova i titoli homebrew N64 e gli SDK per aiutarti a creare il tuo personale su magpi.cc/n64homebrew.

magpi.cc/picocart64

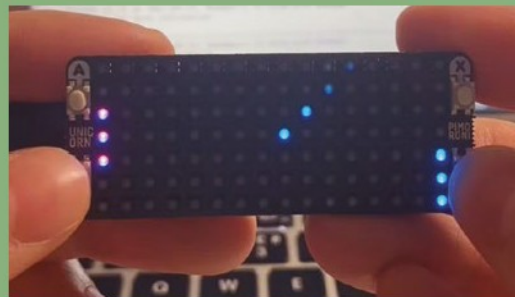


PICO PONG

L'ingegnere del software Pip Austin ha combinato Raspberry Pi Pico con la scheda Rainbow LED Unicorn di Pimoroni (magpi.cc/picounicorn), una matrice 16x7 di LED RGB multicolore con quattro pulsanti integrati, per creare un accattivante gioco luminoso di Pong. Ha scelto Pong perché, come dice lei, è "un gioco classico con regole semplici, ma è visivamente molto potente. Questo aspetto estetico è qualcosa dal quale il Pico Unicorn ha davvero preso vita. Le sue parti preferite del gioco sono visivamente accattivanti: la scia luminosa della palla e il testo scorrevole quando vinci. Rilasciato nel 2021, Pico Pong è stato la prima avventura di Austin nella programmazione Python e il suo tutorial auto-scritto (magpi.cc/picoponggame) fa un ottimo lavoro nel mostrare i suoi lavori e spiegare come e perché ha preso ogni decisione nel suo codice, rendendolo una stimolante risorsa per i programmatori nuovi al linguaggio.

Ma se vuoi solo mettere le tue mani su questa Interpretazione di Pong brillante e con le luci dell'arcobaleno, puoi trovare il suo programma MicroPython completo nel repository GitHub, pronto per essere copiato su Pico con un IDE come Thonny. Lei spera di fare altri giochi sul Pico in futuro: "Ho iniziato a lavorare su Rat On A Scooter - facendo uso della funzione di scorrimento!"

magpi.cc/ponggit



PICO-GB-CART

Se desideri incanalare lo spirito del classico palmare Nintendo più direttamente, puoi sempre trasformare il tuo Pico in una vera e propria flash cart Game Boy. John Green è stato ispirato dalla sua esperienza nella creazione di emulatori di Game Boy per vedere se poteva creare un emulatore hardware con competenze che descrive come "basiche".

Ha usato GB-BRK-CART di Gekkio (magpi.cc/gbbrkcart), una scheda breakout Game Boy, per interfacciarsi con il palmare, per scoprire quali segnali il Game Boy si aspettava e provare a vedere se il Pico era abbastanza veloce da inviargli segnali di I/O.

Quindi, ha progettato un altro prototipo di scheda che ha consentito di saldare il Pico direttamente al PBC. Ha funzionato, ma in entrambi i casi queste schede prototipo interfacciate con il Game Boy erano poco affidabili, dice: "Il Game Boy funziona a 5V e il Pico funziona 3.3V, ma dopo aver testato vari convertitori di livello di tensione, ecc., non trovo mai un modo per leggere/scrivere i dati dal Game Boy e dal Pico mantenendo i voltaggi "corretti".

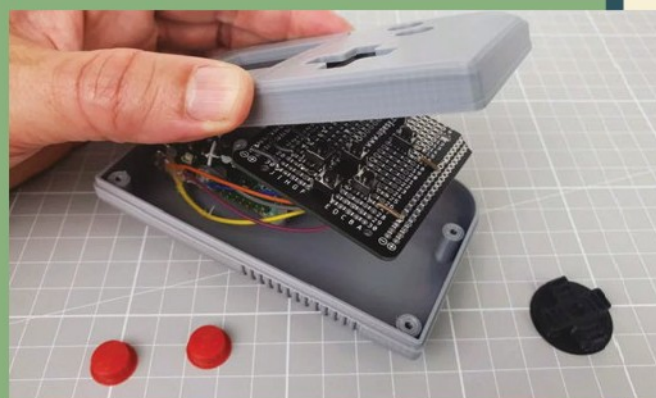
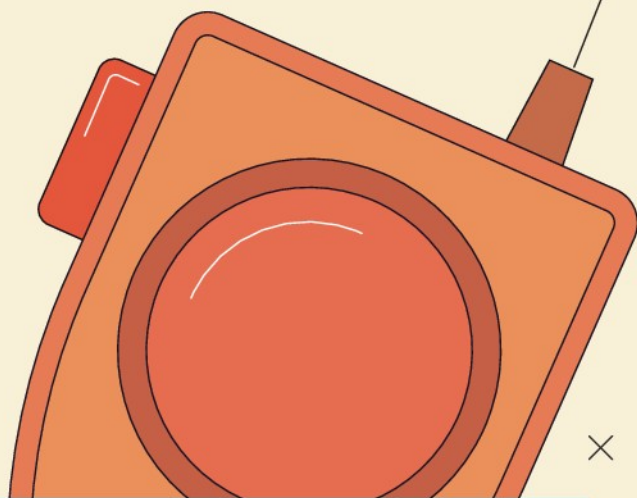
La costruzione è stata abbastanza impegnativa: "Penseresti, con il Game Boy che funziona solo a 4 MHz e il Pico è in grado di funzionare a 133 MHz di base, che [il timing] non dovrebbero essere un problema", dice Green, ma questo non tiene conto del tempo impegnato dal Pico per decodificare il segnale, ottenere la prossima istruzione CPU da inviare al Game Boy e impostare i corretti pin GPIO. alla fine, Green ha scoperto che 133MHz era troppo lento e il Pico doveva essere overcloccato a 360 MHz. Solo uno dei suoi dieci Pico è stato in grado di funzionare stabilmente a questa velocità.

Green non è tornato al codice di una base del progetto per un po', ma ha fatto piani per il futuro, inclusa l'aggiunta della possibilità di caricare immagini di gioco da una scheda SD.

Potete trovare istruzioni e software per il PICO-GB-CART nel suo repository GitHub, mentre schemi per il PCB creato su misura per il progetto di HDR puoi trovarli su magpi.cc/rp2040gbcart.

Gli altri progetti Pico di Green includono un controller hitbox di combattimento, e puoi trovare una panoramica dei progetti software, come raycaster per Game Boy, oltre ad uno sguardo al suo lavoro professionale nell'industria dei giochi, su oxen.github.io.

magpi.cc/picogbcart



YOUMAKE TECH PICO GAMEBOY

Uno dei progetti Raspberry Pi Pico più esteticamente gradevoli che abbiamo visto è questo GameBoy Raspberry Pico, una console giocabile completa con un case personalizzato stampato in 3D, creato da Vincent di YouMakeTech. Vincent fornisce i file STL necessari per creare il case, un elenco dettagliato delle parti, schemi di collegamento e una video guida al montaggio.

Anche se è progettato per assomigliare al classico palmare di Nintendo, anche se è tre quarti delle dimensioni dell'originale, il GameBoy Raspberry Pi Pico non è una semplice emulazione di console.

Dotato di uno schermo a colori da 1,54 pollici, 240x240, 65K, un D-pad e due pulsanti, esiste per incoraggiare i makers e aspiranti sviluppatori ad esplorare la programmazione di giochi a loro scelta in MicroPython o C++.

È una di una serie di console stampate in 3D basate su Pico che Vincent ha creato. Se stai cercando una console di pura emulazione, Pico-GB di YouMakeTech (magpi.cc/picogbemu) può effettivamente giocare a giochi DMG progettati per l'originale Game Boy (non il Game Boy Color) che utilizza l'emulatore RP2040-GB (magpi.cc/rp2040gb).

Ciò include i molti giochi homebrew moderni per Game Boy che puoi trovare su piattaforme come Itch.io (magpi.cc/gbgames).

magpi.cc/picogb



Realizzare Pico Pong

Hosta giochi HTML, incluso Pong, su Pico W e giocali sulla tua rete locale

Top Tip

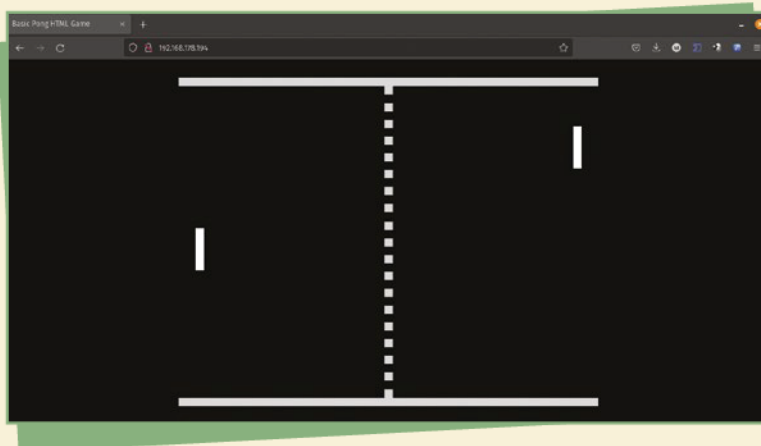
Ultimo indirizzo conosciuto

La maggior parte dei router DHCP riassegna al Pico l'indirizzo IP avuto l'ultima volta; per non avere problemi, assegnagli un indirizzo IP statico.

La maggior parte dei progetti che abbiamo esaminato in queste pagine funzioneranno perfettamente con un Raspberry Pi Pico standard, ma stavolta abbiamo fatto qualcosa che può trarre vantaggio della capacità unica di **networking di Pico W**. Possiamo ospitare alcuni giochi web. Ci sono limiti a ciò che puoi ospitare. I giochi complessi generano errori di memoria, ad esempio. Ma poiché siamo interessati ai giochi retrò, siamo fortunati. Per contribuire a celebrare il 50° anniversario di Pong, ecco una guida completa per principianti all'hosting della versione a due giocatori, HTML5 e JavaScript di Pong su Pico, dall'installazione di MicroPython.

01 Installare Thonny

Se stai usando qualsiasi computer Raspberry Pi per scrivere su Pico, l'IDE Python Thonny dovrebbe essere già installato. Lo troverai dentro i repository di altre



Nuke it from orbit

Se inizi a riscontrare errori di memoria, o è necessario eliminare i file dei progetti o esperimenti precedenti, puoi usare il comando `os.remove` per eliminare i file uno alla volta al prompt dell'interprete REPL. Ma se hai bisogno di cancellare tutto in modo completo, MicroPython incluso, c'è un'opzione più drastica.

Scarica **flash_nuke.uf2** da magpi.cc/flashnuke (download diretto del file). Scollega Pico (se connesso), tieni premuto il pulsante BOOTSEL e collegalo al PC.

Copia **flash_nuke.uf2** su di esso e si riavvierà. Non dimenticare di copiare una nuova build MicroPython nightly prima di intraprendere il tuo prossimo progetto.

distribuzioni Linux, mentre gli utenti Windows e macOS possono scaricarlo da thonny.org.

02 Installare MicroPython

Prendi una nuova build nightly di MicroPython da magpi.cc/rp2picow. Assicurati che Pico W sia scollegato, tieni premuto il pulsante BOOTSEL e contemporaneamente collegalo al tuo PC. Apparirà come dispositivo di archiviazione di massa. Copia il file UF2 scaricato su di esso. Pico W si riavvierà.

03 Preparare Thonny

Apri Thonny. In basso a destra della finestra, vedrai una riga di testo che indica

Thonny parte in modalità semplice, ma vorrai passare alla modalità normale per sbloccare tutte le sue capacità, come il menu Salva con nome

quale interprete che stai utilizzando. Se questo non è "MicroPython (Raspberry Pi Pico)", fai clic sulla riga di testo e seleziona quella opzione.

04 Scripting di un server

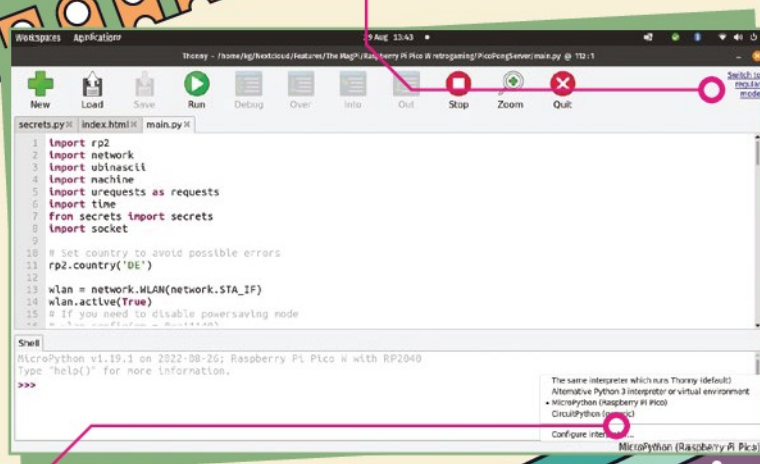
Useremo due script per mettere online il nostro server di gioco intranet Pico W: **main.py** è il codice del server e **secrets.py** contiene l'SSID e la password per la rete wireless, consentendo a Pico W di connettersi automaticamente e iniziare a servire il suo contenuto in prossimità del punto di accesso della rete, ogni volta che è alimentato. I nostri script sono basati su script progetto di server con licenza MIT (magpi.cc/picowledserver) creati da Nathan Bustler di Pi Cockpit. Scarica le nostre versioni di entrambi

Utilizzeremo due script per mettere online il server del nostro gioco intranet Pico W

gli script dalla nostra pagina del progetto all'indirizzo magpi.cc/picopongserver.

05 Configura la tua rete

In Thonny, apri gli scripts **main.py** e **secrets.py**, aggiorna **secrets.py** con l'SSID e la password della rete WiFi, quindi salvalo sul Pico W connesso. Questo è più facile se la copia di Thonny è in modalità normale anziché la modalità semplice in cui inizia. Se sei in modalità semplice fai clic sul collegamento "Passa alla modalità normale" in



Cliccando sul testo in basso a destra si apre il selettore d'interprete. Assicurati che sia selezionato 'MicroPython (Raspberry Pi Pico)'

in alto a destra della barra delle icone di Thonny. In modalità normale, utilizzare l'opzione "Salva come..." del menu File.

06 Basta aggiungere Pong

Utilizzeremo la licenza CC di Straker del Basic Pong Game (magpi.cc/basicpong). Scarica il nostro mirror dell'HTML da magpi.cc/github, aprilo in Thonny e salva il file su Pico. Puoi personalizzarlo, ad esempio modificando la variabile **ballSpeed** per far muovere la pallina più lentamente.

07 Servi la palla!

In Thonny, seleziona il file **main.py** che hai salvato su Pico e premi il pulsante di riproduzione. Supponendo che la rete sia stata configurata correttamente, vedrai linee che mostrano il MAC e l'IP del dispositivo. Copia l'indirizzo IP nel browser di un computer connesso alla stessa rete locale. Congratulazioni, tu e un amico (o la tua mano destra e la tua mano sinistra) ora possono giocare un classico clone Pong nel tuo browser.

Top Tip

Ulteriori esperimenti

Prova altri giochi HTML5 leggeri e vedi quanto in là puoi spingere il Pico. È anche un ottimo modo per fornire oggetti digitali e materiali per i giochi fisici



index.html

> Linguaggio: HTML

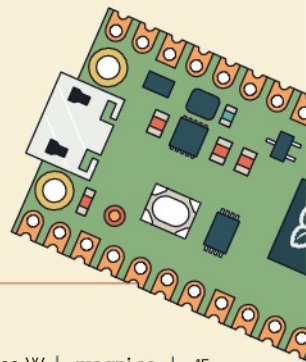
```

001. <!DOCTYPE html>
002. <html>
003. <head>
004.   <title>Basic Pong HTML Game</title>
005.   <meta charset="UTF-8">
006.   <style>
007.     html, body {
008.       height: 100%;
009.       margin: 0;
010.     }
011.
012.     body {
013.       background: black;
014.       display: flex;
015.       align-items: center;
016.       justify-content: center;
017.     }
018.   </style>
019. </head>
020. <body>
021.   <canvas width="750" height="585" id="game"></canvas>
022.   <script>
023.     const canvas = document.getElementById('game');
024.     const context = canvas.getContext('2d');
025.     const grid = 15;
026.     const paddleHeight = grid * 5; // 80
027.     const maxPaddleY = canvas.height - grid -
       paddleHeight;
028.
029.     var paddleSpeed = 6;
030.     var ballSpeed = 5;
031.
032.     const leftPaddle = {
033.       // inizia nel mezzo del gioco, sul lato
       sinistro
034.       x: grid * 2,
035.       y: canvas.height / 2 - paddleHeight / 2,
036.       width: grid,
037.       height: paddleHeight,
038.
039.       // velocita' del paddle
040.       dy: 0
041.     };
042.     const rightPaddle = {
043.       // inizia nel mezzo del gioco, sul lato
       destro
044.       x: canvas.width - grid * 3,
045.       y: canvas.height / 2 - paddleHeight / 2,
046.       width: grid,
047.       height: paddleHeight,
048.
049.       // velocita' del paddle
050.       dy: 0
051.     };
052.     const ball = {
053.       // inizia nel mezzo del gioco
054.       x: canvas.width / 2,
055.       y: canvas.height / 2,
056.       width: grid,
057.       height: grid,
058.
059.       // tieni traccia di quando è necessario
       ripristinare la posizione della palla
060.       resetting: false,
061.
062.       // velocità della palla (inizia andando
       nell'angolo in alto a destra)
063.       dx: ballSpeed,
064.       dy: -ballSpeed
065.     };
066.
067.     // controlla la collisione tra due oggetti usando
       Il riquadro di delimitazione allineato all'asse (AABB)
068.     // @vedi https://developer.mozilla.org/en-US/docs/
       Games/Techniques/2D_collision_detection
069.     function collides(obj1, obj2) {
070.       return obj1.x < obj2.x + obj2.width &&
071.         obj1.x + obj1.width > obj2.x &&
072.         obj1.y < obj2.y + obj2.height &&
073.         obj1.y + obj1.height > obj2.y;
074.     }
075.
076.     // loop di gioco
077.     function loop() {
078.       requestAnimationFrame(loop);
079.       context.clearRect(0,0,canvas.width,canvas.height);
080.
081.       // sposta i paddle in base alla loro velocita'
082.       leftPaddle.y += leftPaddle.dy;
083.       rightPaddle.y += rightPaddle.dy;
084.
085.       // preveni che i paddle passino attraverso i bordi
086.       if (leftPaddle.y < grid) {
087.         leftPaddle.y = grid;
088.       }
089.       else if (leftPaddle.y > maxPaddleY) {
090.         leftPaddle.y = maxPaddleY;
091.       }
092.
093.       if (rightPaddle.y < grid) {
094.         rightPaddle.y = grid;
095.       }
096.       else if (rightPaddle.y > maxPaddleY) {
097.         rightPaddle.y = maxPaddleY;
098.       }
099.
100.       // disegna i paddle
101.       context.fillStyle = 'white';
102.       context.fillRect(leftPaddle.x, leftPaddle.y,
       leftPaddle.width, leftPaddle.height);

```

**SCARICA IL
CODICE COMPLETO:****magpi.cc/github**

```
103. context.fillRect(rightPaddle.x, rightPaddle.y,
104.   rightPaddle.width, rightPaddle.height);
105. // muovi la palla in base alla sua velocita'
106. ball.x += ball.dx;
107. ball.y += ball.dy;
108.
109. // impedisce alla palla di attraversare i muri
    Cambiando la sua velocita'
110. if (ball.y < grid) {
111.   ball.y = grid;
112.   ball.dy *= -1;
113. }
114. else if (ball.y + grid > canvas.height - grid) {
115.   ball.y = canvas.height - grid * 2;
116.   ball.dy *= -1;
117. }
118.
119. // ripristina la palla se supera il paddle (ma solo
    se non l'abbiamo già fatto)
120. if ( (ball.x < 0 || ball.x > canvas.width) &&
    !ball.resetting) {
121.   ball.resetting = true;
122.
123.   // dai tempo al giocatore di riprendersi
    prima di lanciare ancora la palla
124.   setTimeout(() => {
125.     ball.resetting = false;
126.     ball.x = canvas.width / 2;
127.     ball.y = canvas.height / 2;
128.   }, 400);
129. }
130.
131. // controlla se la palla colpisce il paddle. Se
    Così' cambia velocita' x
132. if (collides(ball, leftPaddle)) {
133.   ball.dx *= -1;
134.
135.   // sposta la palla vicino al paddle altrimenti
    La collisione avviene ancora
136.   // nel prossimo frame
137.   ball.x = leftPaddle.x + leftPaddle.width;
138. }
139. else if (collides(ball, rightPaddle)) {
140.   ball.dx *= -1;
141.
142.   // sposta la palla vicino al paddle altrimenti
    La collisione avviene ancora
143.   // nel prossimo frame
144.   ball.x = rightPaddle.x - ball.width;
145. }
146.
147. // disegna la palla
148. context.fillRect(ball.x, ball.y, ball.width,
    ball.height);
149.
150. // disegna i muri
151. context.fillStyle = 'lightgrey';
152. context.fillRect(0, 0, canvas.width, grid);
153. context.fillRect(0, canvas.height - grid,
    canvas.width, canvas.height);
154.
155. // disegna una linea tratteggiata nel mezzo
156. for (let i = grid; i < canvas.height - grid; i +=
    grid * 2) {
157.   context.fillRect(canvas.width / 2 - grid / 2, i,
    grid, grid);
158. }
159. }
160.
161. // ascolta eventi di tastiera per muovere il paddle
162. document.addEventListener('keydown', function(e) {
163.
164.   // tasto freccia su
165.   if (e.which === 38) {
166.     rightPaddle.dy = -paddleSpeed;
167.   }
168.   // tasto freccia giù
169.   else if (e.which === 40) {
170.     rightPaddle.dy = paddleSpeed;
171.   }
172.
173.   // tasto w
174.   if (e.which === 87) {
175.     leftPaddle.dy = -paddleSpeed;
176.   }
177.   // tasto a
178.   else if (e.which === 83) {
179.     leftPaddle.dy = paddleSpeed;
180.   }
181. });
182.
183. // ascolta eventi di tastiera per fermare il paddle
    Se il tasto viene rilasciato
184. document.addEventListener('keyup', function(e) {
185.   if (e.which === 38 || e.which === 40) {
186.     rightPaddle.dy = 0;
187.   }
188.
189.   if (e.which === 83 || e.which === 87) {
190.     leftPaddle.dy = 0;
191.   }
192. });
193.
194. // lancia il gioco
195. requestAnimationFrame(loop);
196. </script>
197. </body>
198. </html>
```



main.py

> Linguaggio: Python

```

001. import rp2
002. import network
003. import ubinascii
004. import machine
005. import urequests as requests
006. import time
007. from secrets import secrets
008. import socket
009.
010. # Imposta il paese per evitare possibili errori
011. rp2.country('IT')
012.
013. wlan = network.WLAN(network.STA_IF)
014. wlan.active(True)
015. # Se devi disabilitare la modalita'powersaving
016. # wlan.config(pm = 0xa11140)
017.
018. # Vedi il MAC address nel chip wireless OTP
019. mac = ubinascii.hexlify(
    network.WLAN().config('mac'),'').decode()
020. print('mac = ' + mac)
021.
022. # Altre cose da interrogare
023. # print(wlan.config('channel'))
024. # print(wlan.config('ssid'))
025. # print(wlan.config('txpower'))
026.
027. # Carica i dati di login da un file differente
    Per questioni di sicurezza
028. ssid = secrets['ssid']
029. pw = secrets['pw']
030.
031. wlan.connect(ssid, pw)
032.
033. # Attendi 10 secondi per la connessione
034. timeout = 10
035. while timeout > 0:
036.     if wlan.status() < 0 or wlan.status() >= 3:
037.         break
038.     timeout -= 1
039.     print('Waiting for connection...')
040.     time.sleep(1)
041.
042. # Definisci una funzione blinking per il LED di bordo
    Per indicare codici di errore
043. def blink_onboard_led(num_blinks):
044.     led = machine.Pin('LED', machine.Pin.OUT)
045.     for i in range(num_blinks):
046.         led.on()
047.         time.sleep(.2)
048.         led.off()
049.         time.sleep(.2)
050.
051. # Gestione errori di connessione
052. # Significato degli errori
053. # 0 Link Down
054. # 1 Link Join
055. # 2 Link NoIp
056. # 3 Link Up
057. # -1 Link Fail
058. # -2 Link NoNet
059. # -3 Link BadAuth
060.
061. wlan_status = wlan.status()
062. blink_onboard_led(wlan_status)
063.
064. if wlan_status != 3:
065.     raise RuntimeError('Wi-Fi connection failed')
066. else:
067.     print('Connected')
068.     status = wlan.ifconfig()
069.     print('ip = ' + status[0])
070.
071. # Funzione da caricare nella pagina html
072. def get_html(html_name):
073.     with open(html_name, 'r') as file:

```

SCARICA IL
CODICE COMPLETO:

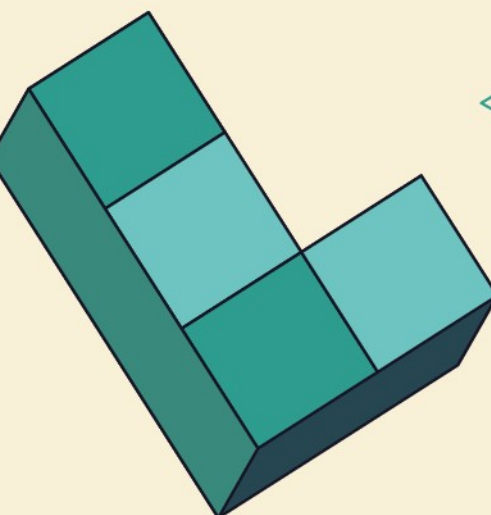
 magpi.cc/github

```
074.         html = file.read()
075.
076.         return html
077.
078. # Server HTTP con socket
079. addr = socket.getaddrinfo('0.0.0.0', 80)[0][-1]
080.
081. s = socket.socket()
082. s.bind(addr)
083. s.listen(1)
084.
085. print('Listening on', addr)
086. led = machine.Pin('LED', machine.Pin.OUT)
087.
088. # In attesa delle connessioni
089. while True:
090.     try:
091.         cl, addr = s.accept()
092.         print('Client connected from', addr)
093.         r = cl.recv(1024)
094.         # print(r)
095.
096.
097.         response = get_html('index.html')
098.         cl.send('HTTP/1.0 200 OK\r\nContent-type:
text/html\r\n\r\n')
099.         cl.send(response)
100.         cl.close()
101.
102.     except OSError as e:
103.         cl.close()
104.         print('Connection closed')
105.
106. # Fare richiesta GET
107. #request = requests.get('http://www.google.com')
108. #print(request.content)
109. #request.close()
```

secrets.py

> Linguaggio: **Python**

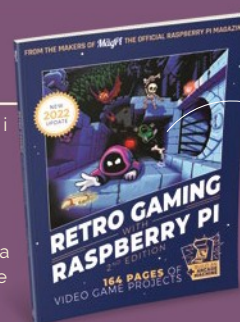
```
001. secrets = {
002.     'ssid': 'Enter_your_SSID_here',
003.     'pw': 'Enter_your_Wi-Fi_password_here',
004. }
```



Retro Gaming with Raspberry Pi

Scopri come configurare Raspberry Pi per i giochi classici nella nuovissima versione della nostra guida al retro gaming. In questo Libro di 164 pagine, imparerai a costruire una macchina da gioco portatile, assemblare un cabinet arcade a grandezza naturale e ad emulare computer e console classici.

magpi.cc/retrogaming



Realizza un Mini Magic Mirror

Riduci uno smart mirror Raspberry Pi
per adattarsi in spazi più piccoli

I magic mirror (specchio magico) è uno di quei grandiosi progetti Raspberry Pi che sembrano fantastici, non sono troppo difficili da fare, e in realtà sono anche un po' pratici. Il software si prende praticamente cura di tutto, ma è molto personalizzabile per coloro che hanno il know-how per andare oltre con i loro progetti.

I magic mirror sono però troppo utili per limitarti a un solo grande specchio nella tua casa, quindi abbiamo raccolto alcune versioni più piccole che puoi provare a mettere sopra il lavandino o anche sulla scrivania. Vedere per credere.



Le basi del Magic Mirror

Come costruire qualsiasi tipo di smart mirror con Raspberry Pi

01 Costruisci una cornice

Prima di ogni altra cosa, ti servirà un TV o monitor vecchio / di riserva per il tuo magic mirror. Dovrai costruire una cornice della dimensione del display, quindi è inutile iniziare senza averne uno. Prendendo le misure, puoi prendere del legno dal negozio di fai-da-te locale, o prendere una cornice adatta da IKEA o su internet per adattarla. Avrai anche bisogno di un foglio di acrilico o pellicola specchio a due vie, per coprire il display per renderlo davvero riflettente.

02 Programmalo

Molti progetti di magic mirror usano il codice software MagicMirror preso da **magicmirror.builders**. È facile da installare e personalizzare e, da un nuovo Raspberry Pi OS installato su Raspberry Pi o Raspberry Pi Zero, puoi andare sul Terminale e scaricare il software con:

```
git clone https://github.com/MichMich/MagicMirror
```

Vai alla documentazione di MagicMirror per vedere tutte le istruzioni di installazione e le opzioni di personalizzazione: **magpi.cc/mirrorstart**.

03 Appendilo

Dovrai trovare un posto con facile accesso a una presa di alimentazione per appendere il tuo specchio e farlo funzionare. Usa le viti invece dei chiodi, ove possibile, poiché l'intera realizzazione sarà pesante, ed è meglio per il progetto e per qualsiasi altra cosa lì vicino che non cada dal muro.

Puoi personalizzarlo con moduli aggiuntivi per diverse funzioni, collegarlo al tuo calendario e molto altro per una reale esperienza di vivere il futuro.



Progetti Mini Mirror

Rendi più piccolo il tuo specchio smart



Magic Mirror con Raspberry Pi

MAKER: SparkFun | magpi.cc/smallmirror

Questa versione dello specchio magico usa un monitor da computer più piccolo e una bacheca a cornice (deep shadow box) per creare uno specchio orizzontale si adatti facilmente a una scrivania o un tavolino. È stato ispirato dalla vita personale frenetica dei suoi creatori, e da come

loro e i loro partner potrebbero usare una sorta di "stazione di battaglia" per tenere traccia dei propri piani.

La bacheca che usa non è proprio delle stesse dimensioni del monitor, e quindi, un po' di spazio è stato riempito di legno in modo che appaia un po' più naturale. Questa è una cosa che dovrai affrontare in qualsiasi progetto di magic mirror che utilizza cornici già pronte ma, applicata la pellicola riflettente, è molto meno evidente di quanto sarebbe stato altrimenti.

Infine, questo tutorial ha alcuni ottimi suggerimenti per personalizzare il layout e i moduli del tuo specchio, adattandoli alle tue esigenze personali.



◀ Il legno nella parte inferiore aiuta a riempire lo spazio e a sollevare il monitor e i connettori ad angolo retto aiutano a mantenere l'intero sistema piatto contro il muro



Smart Mirror usando un Raspberry Pi Zero

MAKER: Brad Morton for LinuxScrew | magpi.cc/smallmirrorzero

Puoi sicuramente fare qualcosa di piccolo con un Raspberry Pi standard, ma usare un Raspberry Pi Zero ti permette davvero di portare le dimensioni di uno smart mirror ai minimi. Anche questo è a orientamento orizzontale, utilizza un piccolissimo schermo da 5 pollici, il tutto inserito in una cornice IKEA. Il display è montato sul retro di un sottile pezzo di acrilico a specchio che occupa tutto lo spazio all'interno della cornice.

Potresti praticamente collegare un Raspberry Pi Zero e fermarti qui, ma questo progetto ti mostra come aggiungere un pulsante che spegnerà e riattiverà lo schermo – risparmiando energia e permettendoti anche di utilizzare lo specchio magico come specchio puro. Ci vuole un po' di hacking extra e l'aggiunta di uno script Python, ma non è poi così difficile da fare.

▼ Con la giusta finitura in legno, questo mini specchio può adattarsi facilmente a qualsiasi arredamento



Progetti schermo informativo

Vuoi i dati ma non lo specchio? Ecco alcune alternative



Sveglia MagicMirror

Anche se non è proprio uno specchio, questa sveglia usa il software MagicMirror per realizzare una sveglia informativa facilmente personalizzabile.

magpi.cc/magicalarm

Schermo informativo digitale

Utilizzando un software chiamato Yodeck, il maker Syed Khairi è stato in grado di creare un utile schermo informativo per il suo dipartimento di scienze.

magpi.cc/discscreen

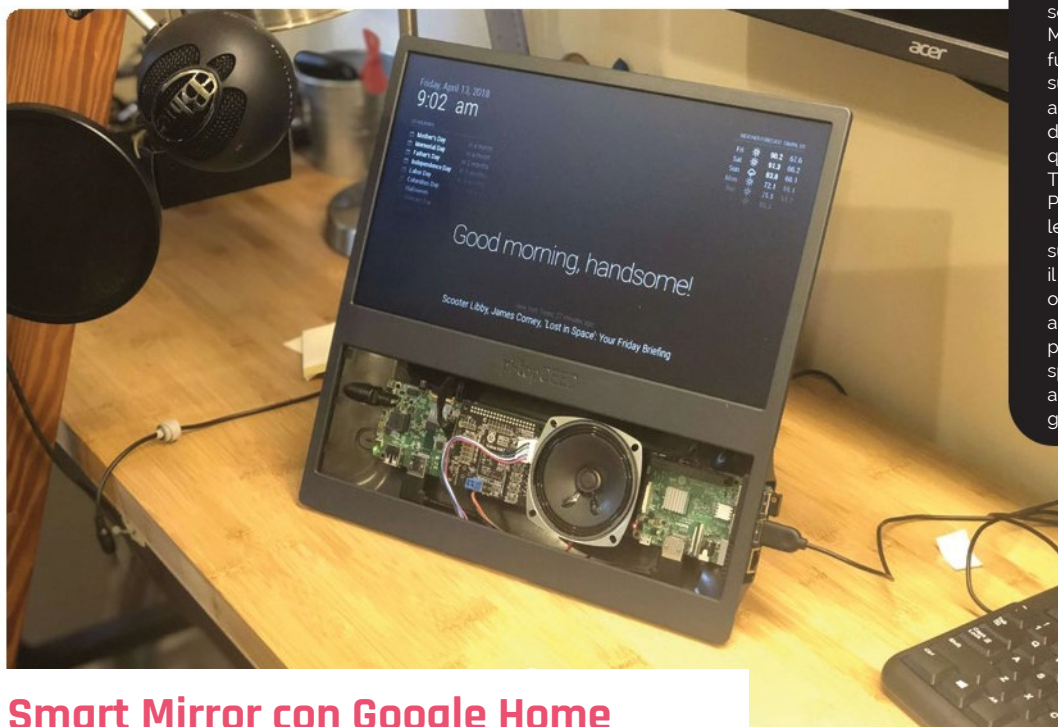


Chromium Kiosk

La modalità chiosco di Chromium consente di creare una selezione di pannelli e funzioni che limitano l'utente a determinate funzioni – come guardare il loro calendario.

magpi.cc/chromiumkiosk





Schermi Raspberry Pi Pico

Sfortunatamente, il software principale MagicMirror non funziona al momento su Pico – è abbastanza diverso da un Raspberry Pi, quindi è logico. Tuttavia, puoi usare Pico per visualizzare le informazioni su uno schermo, con il codice giusto: occorre solo aggiungere una pellicola o acrilico specchio a due vie e avrai uno smart mirror gestito da Pico.

Smart Mirror con Google Home

MAKER: Zach | magpi.cc/voicemirror

Aggiungere i comandi vocali a uno specchio magico è molto figo, specialmente quando usa la tecnologia che una volta è stato regalata da *The MagPi*! L'AIY Voice Kit è un modo semplice per aggiungere un controllo vocale tipo Google Home a un progetto Raspberry Pi e a noi piace così tanto da riutilizzare anche un vecchio pi-topCEED. Ti permette di usare lo spazio in modo efficiente, poiché è già progettato per essere abbastanza sottile ma capace di contenere un Raspberry Pi.

Si può fare anche in un pomeriggio! Zach calcola che gli ci sono volute circa due ore, e possiamo vedere perché: tutto quel che doveva fare era aggiungere il Voce Kit, il software MagicMirror e lo specchio a due vie – che è abilmente fissato con dei magneti per renderlo

rimovibile per una facile manutenzione. Non c'è nemmeno una saldatura: tutto a incastro.

La parte più difficile è far funzionare il kit Voice AIY con il software, ma Zach ha una ottima guida alla risoluzione dei problemi per te, se iniziassi a averene.



► A causa della forma del pi-top, lo specchio finale è quadrato (13" per 13"), ma ha un aspetto piuttosto unico



► Ci piace l'idea di portarlo nel bagno e usarlo per radersi

Mini Magic Mirror

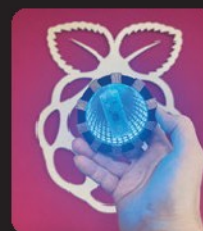
MAKER: **GeekToolkit** | magpi.cc/minimirror

Questo progetto di specchio magico è davvero mini – utilizza uno schermo da 3,5 pollici e uno specchio a due vie quadrato da 115 mm. Non preoccuparti della miscela tra pollici e sistema metrico, soprattutto quando l'intero progetto ti costerà da 60€/ 50£ per realizzarlo!

Il supporto a piramide per lo schermo è fantastico per molteplici ragioni. In primo luogo, è molto stabile; mentre l'intera realizzazione non è molto pesante, comunque deve essere robusta. In secondo luogo, è vuoto, offrendo spazio più che sufficiente per un Raspberry Pi Zero. Il supporto è composto da due parti stampate in 3D che si collegano perfettamente insieme e offre



anche dello spazio per collegare lo specchio alla presa di corrente – o a una batteria mobile se vuoi portarlo in giro in casa. È un progetto molto interessante, molto poco faticoso, che ha molti usi.



Specchio infinito

Non molto smart, ma molto bello, gli specchi infiniti usano specchi e luci per farli sembrare proseguire per sempre. Toby Roberts, di Raspberry Pi, ha costruito questo e altri con dentro un Raspberry Pi Pico e alimentando tutte le luci.

