

Completamente aggiornata
per Raspberry Pi 5

Come utilizzare il tuo nuovo computer



Gareth Halfacree

La guida introduttiva ufficiale Raspberry Pi, 5^a edizione

La guida introduttiva ufficiale Raspberry Pi

di Gareth Halfacree

ISBN: 978-1-912047-39-0

Copyright © 2024 Gareth Halfacree

Stampata nel Regno Unito

Pubblicata da Raspberry Pi, Ltd., 194 Science Park, Cambridge, CB4 0AB

Editori: Brian Jepson, Liz Upton, Francesco Marchetti, Phil King

Traduttore: Alpha CRC

Interior Designer: Sara Parodi

Produzione: Nellie McKesson

Fotografia: Brian O'Halloran

Illustrazioni: Sam Alder

Editor della grafica: Natalie Turner

Direttore editoriale: Brian Jepson

Responsabile del design: Jack Willis

Amministratore delegato: Eben Upton

Maggio 2024: Quinta edizione

Novembre 2020: Quarta edizione

Novembre 2019: Terza edizione

Giugno 2019: Seconda edizione

Dicembre 2018: Prima edizione

L'editore e i collaboratori non si assumono alcuna responsabilità per eventuali omissioni o errori relativi a beni, prodotti o servizi a cui si fa riferimento o che sono pubblicizzati in questa guida. Salvo diversa indicazione, il contenuto di questa guida è concesso in licenza Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0)

Indice

v	Benvenuto
vii	Informazioni sull'autore

Capitolo 1

1 **Cos'è Raspberry Pi**

Il tuo nuovo computer dalle dimensioni di una carta di credito. Fai un tour guidato del Raspberry Pi, scopri come funziona e conosci alcune delle incredibili cose che puoi fare.

Capitolo 2

15 **Introduzione a Raspberry Pi**

Scopri quali sono i componenti esterni essenziali per il tuo Raspberry Pi e come collegarli per farlo funzionare in modo ottimale.

Capitolo 3

31 **Come utilizzare Raspberry Pi**

Scopri tutto sul sistema operativo Raspberry Pi.

Capitolo 4

53 **Programmare con Scratch 3**

Scopri come iniziare a programmare con Scratch, il linguaggio di programmazione a blocchi

Capitolo 5

89 **Programmare con Python**

Ora che hai imparato a usare Scratch, affronteremo la programmazione basata su testo con Python.

Capitolo 6

121 **Physical computing con Scratch e Python**

Programmare non vuol dire solo eseguire azioni su uno schermo, ma anche controllare i componenti elettronici collegati ai pin GPIO del tuo Raspberry Pi

Capitolo 7

155 **Physical computing con Sense HAT**

Sense HAT è una scheda aggiuntiva multifunzione Raspberry Pi dotata di sensori e display a matrice LED, la stessa usata nella Stazione spaziale internazionale.

Capitolo 8

201 **Moduli fotocamera di Raspberry Pi**

Con il Camera Module o la High Quality Camera potrai scattare foto e registrare video in alta risoluzione e creare fantastici progetti di visione artificiale direttamente sul tuo Raspberry Pi.

Capitolo 9

219 **Raspberry Pi Pico e Pico W**

Raspberry Pi Pico e Pico W danno una dimensione completamente nuova ai progetti di physical computing

Appendice

241 **Appendice A**

Come installare un sistema operativo su una scheda microSD

249 **Appendice B**

Come installare e disinstallare il software

257 **Appendice C**

Interfaccia a riga di comando

267 **Appendice D**

Ulteriori letture

277 **Appendice E**

Strumento Configurazione di Raspberry Pi

287 **Appendice F**

Specifiche tecniche di Raspberry Pi

Benvenuto

Siamo sicuri che Raspberry Pi ti piacerà. Qualunque sia il tuo modello: una scheda Raspberry Pi standard, la compatta Raspberry Pi Zero 2 W o il Raspberry Pi 400 con tastiera integrata, questo computer economico può essere utilizzato per imparare la programmazione, costruire robot e creare ogni tipo di progetto particolare e meraviglioso.

Con Raspberry Pi puoi fare tutto quello che faresti con un computer normale: navigare in Internet, giocare, guardare film e ascoltare musica. Ma Raspberry Pi è molto più di un moderno computer.

È il vero e proprio cuore di un computer. Imparerai a configurare il tuo sistema operativo e a collegare fili e circuiti direttamente ai pin GPIO. È stato progettato per insegnare ai più giovani come programmare utilizzando linguaggi come Scratch e Python, due dei principali linguaggi di programmazione inclusi nel sistema operativo ufficiale. Con Raspberry Pi Pico puoi creare progetti discreti e a bassa potenza in grado di interagire con il mondo fisico.

Mai come oggi il mondo ha bisogno di programmatori e Raspberry Pi è riuscito a suscitare nelle nuove generazioni l'interesse per l'informatica e la tecnologia.

Persone di tutte le età usano Raspberry Pi per creare progetti straordinari: dalle console per il retrogaming alle stazioni meteorologiche collegate a Internet.

Quindi, se anche tu vuoi progettare giochi, costruire robot o realizzare tanti progetti sorprendenti, sfoglia questa guida e preparati a iniziare.

Potrai trovare esempi e altre informazioni relative a questa guida, comprese le correzioni, nel relativo repository GitHub all'indirizzo **rptl.io/bg-resources**. Se hai riscontrato un errore nella guida, ti invitiamo a comunicarcelo utilizzando il nostro apposito modulo all'indirizzo **rptl.io/bg-errata**.

Informazioni sull'autore

Gareth Halfacree è un giornalista freelance esperto in tecnologia, scrittore ed ex amministratore di sistema nel settore dell'istruzione con la passione del software e dell'hardware open-source. È stato uno dei primi ad adottare la piattaforma Raspberry Pi e a descriverne capacità e flessibilità in diverse pubblicazioni. Puoi trovarlo su Mastodon come @ghalfacree@mastodon.social o sul suo sito Web freelance.halfacree.co.uk.

Colophon

Raspberry Pi è un sistema economico per fare qualcosa di utile o di divertente.

La principale motivazione alla base del progetto Raspberry Pi è la democratizzazione della tecnologia fornendo l'accesso agli strumenti necessari. Abbassando il costo del general-purpose computing al di sotto dei 5 dollari, abbiamo aperto a chiunque la possibilità di utilizzare computer in progetti che prima richiedevano capitali proibitivi. Oggi, grazie all'abbattimento delle barriere di accesso, i computer Raspberry Pi vengono utilizzati ovunque, dalle mostre interattive nei musei alle scuole, dagli uffici postali ai call center governativi. Le aziende di tutto il mondo sono state in grado di scalare e conseguire risultati in un modo che non era possibile in un contesto in cui integrare la tecnologia significava spendere grandi somme in computer portatili e PC.

Raspberry Pi elimina l'elevato costo d'ingresso all'informatica per le persone di tutte le fasce: come i bambini possono beneficiare di un'educazione informatica che prima non era loro accessibile, così anche molti adulti, prima esclusi dall'uso dei computer per l'impresa, l'intrattenimento e la creatività a causa dei costi ingenti, possono finalmente usufruire degli strumenti necessari. Raspberry Pi infatti elimina tali barriere.

Raspberry Pi Press

store.rpipress.cc

Raspberry Pi Press è la tua libreria essenziale per l'informatica, il gioco e la creazione pratica. Siamo il marchio editoriale di Raspberry Pi Ltd, parte della Raspberry Pi Foundation. Grazie alla nostra vasta gamma di libri e riviste potrai scoprire le tue passioni, apprendere nuove abilità e creare progetti fantastici, dalla realizzazione di un PC a quella di un armadio.

The MagPi

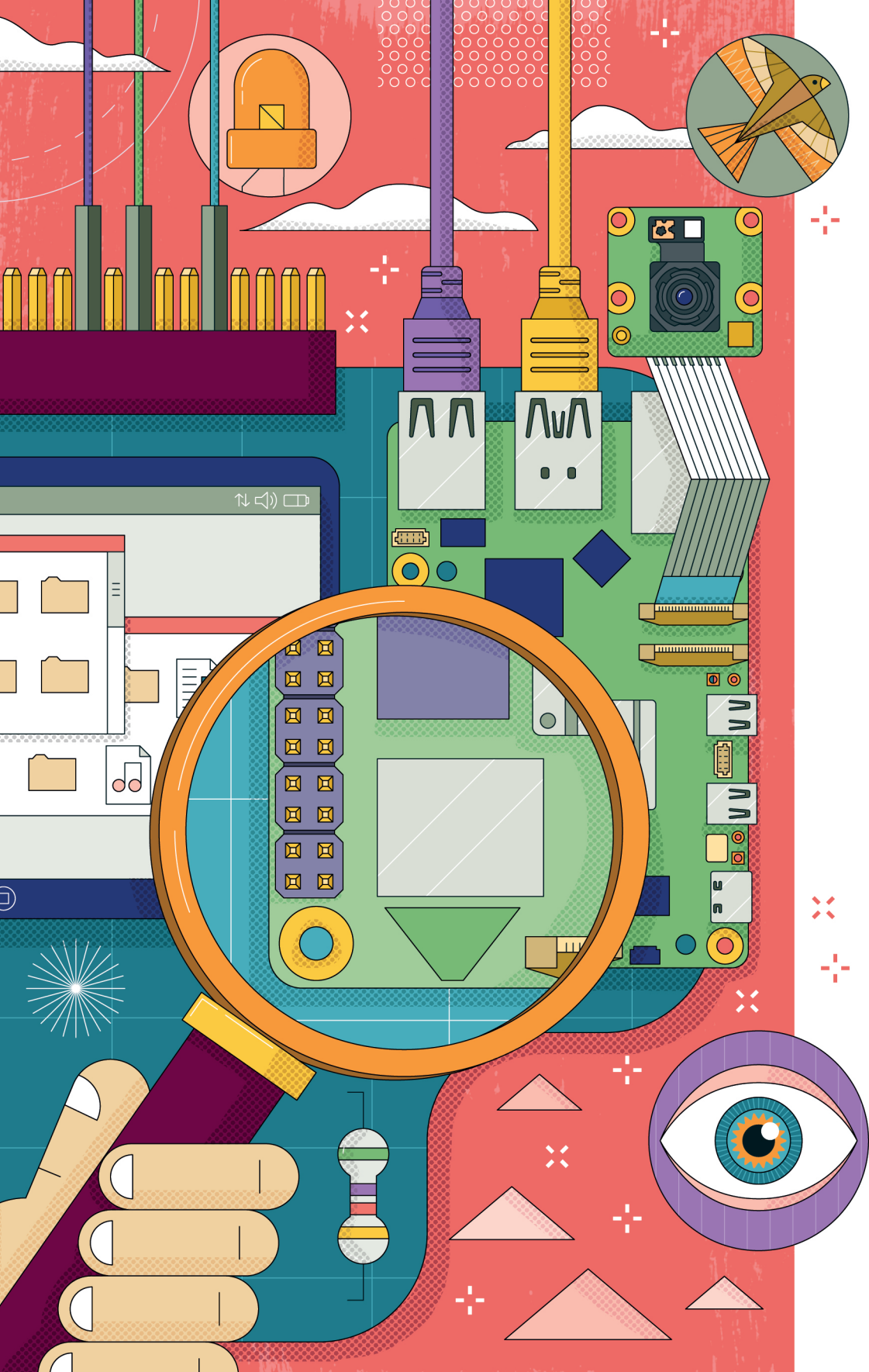
magpi.raspberrypi.com

The MagPi è la rivista ufficiale di Raspberry Pi. Scritta per la comunità Raspberry Pi, è ricca di progetti a tema Pi, tutorial sull'informatica e l'elettronica, guide pratiche e le ultime notizie ed eventi della comunità.

HackSpace

hackspace.raspberrypi.com

La rivista *HackSpace* presenta numerosi progetti pensati per vari livelli e dedicati a chi ama riparare e cimentarsi con progetti. Ti insegneremo nuove tecniche e ti aggiorneremo su quelle già conosciute, dalla stampa 3D, al taglio laser, alla lavorazione del legno, all'elettronica e all'Internet delle cose. *HackSpace* ti ispirerà a sognare più in grande e a costruire meglio.



Capitolo 1

Cos'è Raspberry Pi

Il tuo nuovo computer dalle dimensioni di una carta di credito. Fai un tour guidato del Raspberry Pi, scopri come funziona e conosci alcune delle incredibili cose che puoi fare.

Raspberry Pi è un dispositivo straordinario: un computer perfettamente funzionante dal design compatto ed economico. Che tu stia cercando un computer da utilizzare per navigare sul Web o per giocare, per imparare a scrivere programmi o per creare circuiti e dispositivi progettati da te, Raspberry Pi e la sua incredibile community ti supporteranno in ogni fase.

Raspberry Pi è noto come *computer a scheda singola*, che significa esattamente quello che sembra: è un computer a tutti gli effetti, proprio come un PC desktop, un laptop o uno smartphone, ma costituito da un'unica *scheda a circuito stampato*. Come la maggior parte dei computer a scheda singola, Raspberry Pi è grande più o meno come una carta di credito, ma ciò non significa che non sia potente: un Raspberry Pi può fare esattamente le stesse cose che fa un computer più grande, come ad esempio navigare sul Web, rilassarsi con un videogame oppure gestire altri dispositivi.

La gamma di prodotti Raspberry Pi nasce dal desiderio di promuovere un'educazione informatica più pratica e accessibile in tutto il mondo. I suoi creatori, che si sono uniti per formare l'organizzazione senza scopo di lucro Raspberry Pi Foundation, non avevano idea che sarebbe diventata così popolare. Dalle poche migliaia di unità costruite nel 2012 per sondare il terreno e andate immediatamente esaurite, sono passati a spedirne più di cinquanta milioni in tutto il mondo. Queste schede si sono rivelate la soluzione ideale per abitazioni, aule, uffici, data centre, fabbriche e persino barche a pilotaggio automatico e satelliti.

Dopo il Model B originario, si sono susseguiti diversi modelli, ciascuno con specifiche migliorate o caratteristiche dedicate a un uso specifico. La gamma Raspberry Pi Zero, ad esempio, è la versione ridotta dell'originale, che però manca di alcune caratteristiche, come ad esempio le varie porte USB e la porta Ethernet, a favore di un design significativamente più piccolo e un ridotto fabbisogno di potenza.

Tutti i modelli Raspberry Pi hanno una cosa in comune: sono *compatibili* fra loro, nel senso che la maggior parte del software scritto per un particolare modello funzionerà anche sugli altri. È persino possibile prendere l'ultima versione del sistema operativo di Raspberry Pi ed eseguirla sul prototipo del Model B originario. Funzionerà più lentamente, ma funzionerà.

Attraverso questa guida conoscerai Raspberry Pi 4 Model B, Raspberry Pi 5, Raspberry Pi 400 e Raspberry Pi Zero 2 W: le versioni più recenti e potenti di Raspberry Pi. Tutto ciò che apprenderai può essere facilmente applicato anche agli altri modelli della gamma Raspberry Pi, quindi non preoccuparti se stai utilizzando un modello o una versione differenti.



RASPBERRY PI 400

Se disponi di un Raspberry Pi 400, il circuito stampato è integrato nella custodia della tastiera. Continua a leggere per conoscere tutti i componenti che fanno funzionare un Raspberry Pi oppure consulta «Raspberry Pi 400» a pagina 10 per una panoramica delle caratteristiche del tuo dispositivo desktop.



RASPBERRY PI ZERO 2 W

Se possiedi un Raspberry Pi Zero 2 W, alcune porte e componenti appaiono diversi rispetto a quelli di un Raspberry Pi 4 Model B. Continua a leggere per scoprire le funzioni di ciascun componente oppure consulta «Raspberry Pi Zero 2 W» a pagina 12 per saperne di più sul tuo dispositivo.

Introduzione al Raspberry Pi

A differenza di un computer tradizionale, che nasconde tutte le sue parti all'interno di un case, un Raspberry Pi standard ha tutti i suoi componenti, porte e funzionalità in mostra — sebbene sia possibile acquistare un case per fornire una protezione aggiuntiva, se lo si preferisce. Questo lo rende un ottimo strumento per imparare cosa fanno le diverse parti di un computer e rende anche facile capire cosa-va-dove quando arriva il momento di collegare i vari altri pezzi di hardware, noti come *periferiche*, di cui avrai bisogno per iniziare.

Figura 1-1 mostra un Raspberry Pi 5 dall'alto. Durante la lettura di questa guida, cerca di tenere sempre il tuo Raspberry Pi orientato come nelle foto, per evitare di confonderti quando dovrai maneggiare cose come il connettore GPIO (maggiori dettagli nel Capitolo 6, *Physical computing con Scratch e Python*).

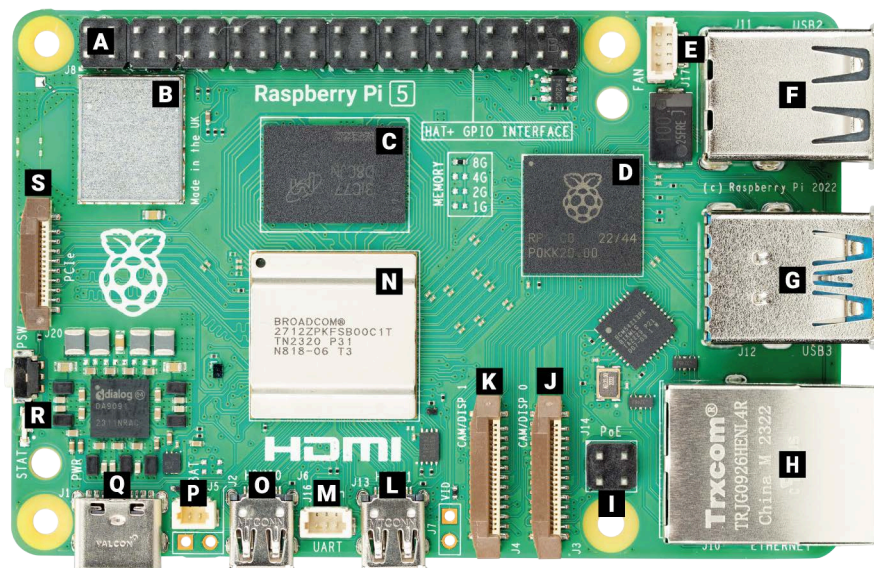


Figura 1-1 Raspberry Pi 5

- | | |
|---|--|
| A Connettore GPIO | K CSI/DSI 1 (fotocamera/display) |
| B Wireless | L Micro HDMI 1 |
| C RAM | M Connettore per porta seriale UART |
| D Chip controller I/O RP1 | N System-on-chip |
| E Connettore per la ventola | O Micro HDMI 0 |
| F USB 2.0 | P Connettore batteria RTC |
| G USB 3.0 | Q Alimentazione USB di tipo C |
| H Porta Ethernet | R Pulsante di accensione |
| I Pin Power-over-Ethernet (PoE) | S Connettore per PCI Express (PCIe) |
| J CSI/DSI 0 (fotocamera/display) | |

Anche se sembra che la scheda contenga troppi elementi, il Raspberry Pi è in realtà un dispositivo piuttosto semplice da capire, a partire dai suoi *componenti*, i meccanismi interni che lo fanno funzionare.

I componenti del Raspberry Pi

Come ogni computer, Raspberry Pi è costituito da molti componenti, ognuno dei quali svolge un ruolo ben preciso. Il primo, e probabilmente il più importante, si trova appena a sinistra del punto centrale della scheda (**Figura 1-2**), ed è coperto da una placca in metallo: è il *system-on-chip* (SoC).

Il nome "system-on-chip" è un ottimo indizio di ciò che troveresti rimuovendo la parte in metallo: un chip di silicio, meglio conosciuto come *circuito integrato*, che contiene la maggior parte del sistema di un Raspberry Pi. Questo circuito integrato è composto infatti da un'*unità di elaborazione centrale* (CPU), comunemente considerata come il "cervello" di un computer, e dall'*unità di elaborazione grafica* (GPU), che gestisce tutta la parte grafica e di rendering.

Un cervello senza memoria, però, non funziona bene ed è per questo che a fianco del SoC troverai un altro chip che sembra un rettangolino nero di plastica (**Figura 1-3**). Si tratta della *memoria ad accesso casuale* (RAM) del Raspberry Pi. Quando stai lavorando su un Raspberry Pi, è la RAM che conserva le informazioni di quello che stai facendo. Salvando il tuo lavoro, questi dati vengono spostati nella memoria di archiviazione permanente della microSD card. Insieme, questi componenti costituiscono la memoria *volatile* e *non volatile* del Raspberry Pi: la RAM volatile perde i suoi contenuti ogni volta che il Raspberry Pi viene spento, mentre la memoria non volatile nella microSD card mantiene i suoi contenuti anche dopo.



Figura 1-2
Il system-on-chip di Raspberry Pi (SoC)

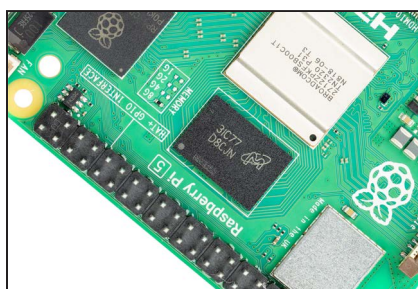


Figura 1-3
Memoria ad accesso casuale (RAM) di Raspberry Pi

Nella parte in alto a sinistra della scheda c'è un'altra copertura in metallo (**Figura 1-4**) sotto la quale si trova il *modulo radio*, il componente che consente al Raspberry Pi di comunicare con i dispositivi in modalità wireless. Tale mo-

dulo svolge la funzione di due altri componenti comuni: il *Wi-Fi*, che si collega alle reti informatiche, e il *Bluetooth*, che si connette a periferiche come i mouse e invia o riceve dati da dispositivi intelligenti nelle vicinanze come sensori o smartphone.

Sul lato destro della scheda, accanto alle porte USB (**Figura 1-5**), si trova un altro chip con copertura in plastica nera, marchiato con il logo di Raspberry Pi. Si tratta di *RP1*, un chip controller I/O personalizzato che comunica con le quattro porte USB, la porta Ethernet e la maggior parte delle interfacce a bassa velocità di altri hardware.

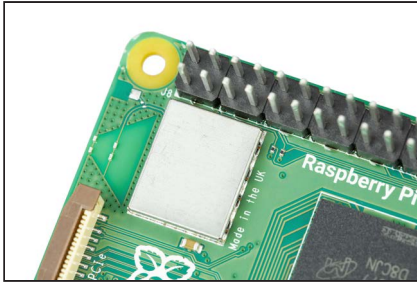


Figura 1-4
Il modulo radio di Raspberry Pi

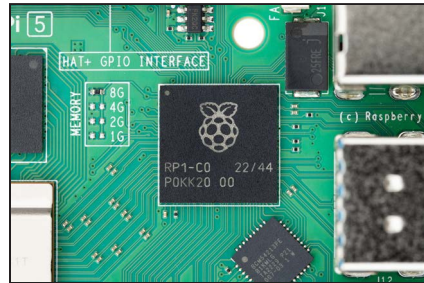


Figura 1-5
Il chip controller RP1 di Raspberry Pi

Un altro chip nero, più piccolo degli altri, si trova un po' sopra il connettore di alimentazione USB C in basso a sinistra della scheda (**Figura 1-6**). Si tratta del *circuito integrato di gestione dell'alimentazione (PMIC)*: prende l'alimentazione che proviene dalla porta USB C e la trasforma nell'energia di cui il Raspberry Pi ha bisogno per funzionare.

L'ultimo chip nero, situato sotto RP1 e posizionato in diagonale, supporta l'RP1 nella gestione della porta Ethernet del Raspberry Pi. Fornisce il cosiddetto *Ethernet PHY*, ovvero l'interfaccia *fisica* che si trova tra la porta Ethernet stessa e il controller Ethernet nel chip RP1.

Se ti sembra di doverti ricordare troppe cose, non preoccuparti: per poter utilizzare Raspberry Pi non è necessario conoscere ogni componente o sapere dove si trova sulla scheda.

Le porte del Raspberry Pi

Il Raspberry Pi ha diverse porte, a cominciare dalle quattro *porte USB (Universal Serial Bus)* (**Figura 1-7**) al centro e nella parte superiore del bordo destro. Queste porte consentono di collegare al tuo Raspberry Pi qualsiasi periferica compatibile con USB, come tastiere, mouse, fotocamere digitali e memorie

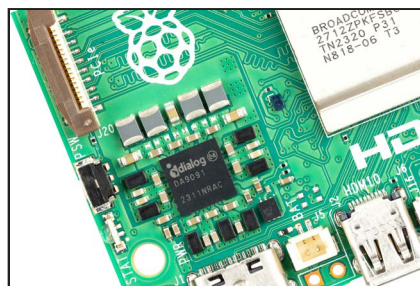


Figura 1-6
Circuito integrato di gestione dell'alimentazione
(PMIC) di Raspberry Pi

flash. Tecnicamente parlando, ci sono due tipi di porte USB sul Raspberry Pi, ognuno relativo ad un diverso standard Universal Serial Bus: quelle con parti in plastica nera all'interno sono porte USB 2.0, mentre quelle con parti in plastica blu sono porte USB 3.0, più veloci, basate sulla più recente versione 3.

Accanto alle porte USB c'è una *porta Ethernet*, nota anche come *porta di rete* (**Figura 1-8**), che serve a collegare il Raspberry Pi a una rete di computer cablata utilizzando un cavo con il cosiddetto connettore RJ45. Se guardi attentamente nella parte inferiore della porta Ethernet, noterai due diodi luminosi (LED) che fungono da indicatori di stato segnalando, tramite la luce accesa o lampeggiante, il funzionamento della connessione.



Figura 1-7
Le porte USB di Raspberry Pi



Figura 1-8
La porta Ethernet di Raspberry Pi

Appena a sinistra della porta Ethernet, sul bordo inferiore del Raspberry Pi, si trova il *connettore PoE (Power-over-Ethernet)* (**Figura 1-9**). Questo connettore, se abbinato a Raspberry Pi 5 PoE+ *HAT (Hardware Attached on Top)*, una speciale scheda aggiuntiva progettata per Raspberry Pi, e a un adeguato switch di rete con capacità PoE, consente di alimentare Raspberry Pi dalla sua porta Ethernet senza dover collegare nulla al connettore USB C. Lo stesso connet-

tore è disponibile anche su Raspberry Pi 4, anche se in una posizione diversa; Raspberry Pi 4 e Raspberry Pi 5 utilizzano HAT diversi per il supporto PoE.

Subito a sinistra del connettore PoE sono presenti un paio di connettori dall'aspetto un po' strano, dotati di alette in plastica che possono essere sollevate; si tratta dei *connettori di fotocamera e display*, noti anche come porte *Camera Serial Interface (CSI)* e *Display Serial Interface (DSI)* (**Figura 1-10**).



Figura 1-9
Il connettore Power-over-Ethernet di Raspberry Pi

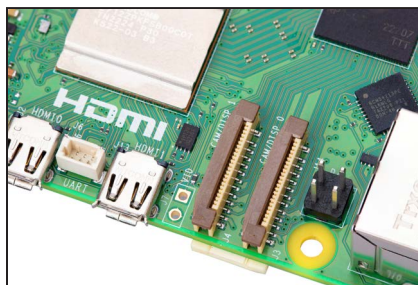


Figura 1-10
I connettori di fotocamera e display di Raspberry Pi

Puoi utilizzare questi connettori per collegare un display compatibile con DSI, come il display touchscreen di Raspberry Pi o la gamma di moduli fotocamera di Raspberry Pi appositamente progettati (vedi **Figura 1-11**). Puoi scoprire di più sui moduli fotocamera nel Capitolo 8, *Moduli fotocamera di Raspberry Pi*. Entrambe le porte possono fungere da ingresso per la fotocamera o da uscita per il display, per cui puoi avere due fotocamere CSI, due display DSI o una fotocamera CSI e un display DSI in esecuzione su Raspberry Pi 5.

A sinistra dei connettori di fotocamera e display, sempre sul bordo inferiore della scheda, sono presenti le *porte micro HDMI* (*micro High Definition Multimedia Interface*), che sono una versione più piccola dei connettori che puoi trovare su console di gioco, decoder o televisori (**Figura 1-12**). La "M" di Multimedia indica che la porta è in grado di inviare sia segnali audio che video, mentre "High Definition" indica che le prestazioni sono di alta qualità, per entrambi i segnali. Potrai utilizzare queste porte per collegare Raspberry Pi a uno o due dispositivi di visualizzazione, come un monitor per computer, un televisore o un proiettore.

Tra le due porte micro HDMI è presente un piccolo connettore denominato "UART", che consente di accedere a una *porta seriale UART* (*Universal Asynchronous Receiver-Transmitter*). Non parleremo di questa porta nella guida, ma ti potrebbe servire in futuro per comunicare con progetti più complessi o per investigarne eventuali problemi.

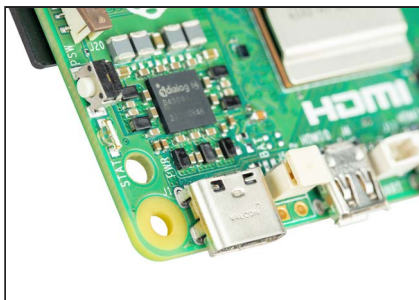


Figura 1-13

La porta di alimentazione USB di tipo C di Raspberry Pi

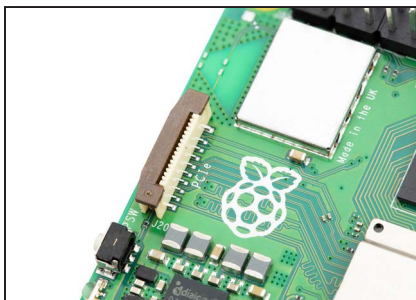


Figura 1-14

Il connettore di Raspberry Pi per PCI Express.

Sul bordo superiore della scheda sono presenti 40 pin in metallo, suddivisi in due file di 20 pin ciascuna (**Figura 1-15**). Si tratta del *connettore GPIO* (*general-purpose input/output*), un componente importante del Raspberry Pi, utilizzato per comunicare con l'hardware aggiuntivo, da LED e pulsanti fino a sensori di temperatura, joystick e monitor della frequenza cardiaca. Scoprirai di più sul connettore GPIO nel Capitolo 6, *Physical computing con Scratch e Python*.

C'è un'ultima porta sul Raspberry Pi, ma la vedrai solo capovolgendo la scheda. Qui, sul retro, è presente un *connettore per schede microSD* posizionato quasi esattamente sotto il connettore del lato frontale contrassegnato come "PCIe" (**Figura 1-16**). Questo connettore è destinato al dispositivo di archiviazione del Raspberry Pi: la microSD card inserita qui contiene tutti i file che salvi, tutto il software che installi e il sistema operativo che fa funzionare il tuo Raspberry Pi.

Puoi anche utilizzare il tuo Raspberry Pi senza una scheda microSD, caricando il software attraverso la rete, da una memoria USB o da un'unità SSD M.2. In questa guida manterremo le cose semplici concentrandoci sull'utilizzo di una scheda microSD come dispositivo di archiviazione principale.

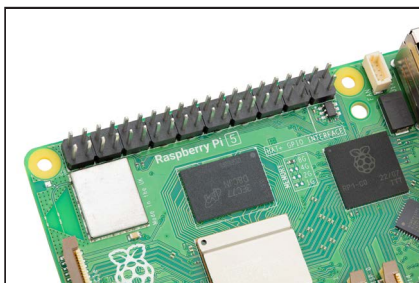


Figura 1-15

Il terminale GPIO di Raspberry Pi

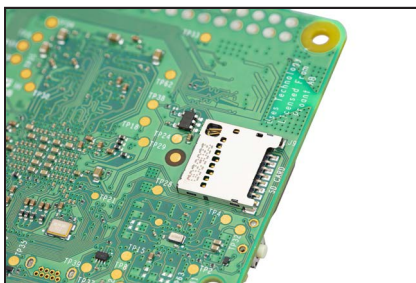


Figura 1-16

Il connettore per scheda microSD di Raspberry Pi

Raspberry Pi 400

Il Raspberry Pi 400 presenta gli stessi componenti del Raspberry Pi 4, compresi il system-on-chip e la memoria, posizionati all'interno di un comodo alloggiamento per tastiera. Oltre a proteggere i componenti elettronici, l'alloggiamento per tastiera occupa meno spazio e ti consente di tenere in ordine i cavi.

Sebbene non sia possibile vedere facilmente i componenti interni, è possibile vedere bene le parti esterne, a partire dalla tastiera stessa (**Figura 1-17**). Nell'angolo superiore destro ci sono tre diodi luminosi (LED): il primo si accende quando si preme il tasto **Num Lock**, il quale imposta alcuni dei tasti per agire come il tastierino numerico su una tastiera standard; il secondo si accende quando si preme **Caps Lock**, che consente di usare lettere maiuscole anziché minuscole, mentre l'ultimo si illumina quando il Raspberry Pi 400 è acceso.



Figura 1-17 Raspberry Pi 400 ha una tastiera integrata

Sulla parte posteriore del Raspberry Pi 400 (**Figura 1-18**) si trovano le porte. La porta più a sinistra è il connettore GPIO (general-purpose input/output). Questo è lo stesso connettore mostrato in **Figura 1-15**, ma capovolto: il primo pin, il pin 1, è in alto a destra, mentre l'ultimo pin, il pin 40, è in basso a sinistra. Per ulteriori informazioni sul connettore GPIO, consulta il Capitolo 6, *Physical computing con Scratch e Python*.

Vicino al connettore GPIO si trova l'alloggiamento per la scheda microSD. Allo stesso modo dello slot sul lato inferiore del Raspberry Pi 5, questo contiene la scheda microSD che funge da memoria per il sistema operativo, le applicazioni e gli altri dati del Raspberry Pi 400. Il Raspberry Pi 400 Personal Computer Kit viene fornito con una scheda microSD preinstallata. Una volta inserita, per rimuovere la microSD, premi delicatamente su di essa finché non senti un clic e la vedi fuoriuscire leggermente. A questo punto la puoi estrarre completamente. Quando la rimetti in posizione, assicurati che i contatti metallici siano rivolti verso il basso e premila delicatamente fino a sentire un altro clic, ad indicare che è bloccata nella giusta posizione.



Figura 1-18 Le porte sul retro di Raspberry Pi 400

Le due porte successive sono le porte micro HDMI, utilizzate per collegare monitor, televisori o altri dispositivi di visualizzazione. Come Raspberry Pi 4 e Raspberry Pi 5, anche Raspberry Pi 400 supporta fino a due display HDMI. Accanto alle porte micro HDMI si trova la porta di alimentazione USB di tipo C, utilizzata per collegare l'alimentatore originale di Raspberry Pi o qualsiasi altro alimentatore USB C compatibile.

Le due porte blu sono porte USB 3.0, che forniscono una connessione ad alta velocità a dispositivi come unità di memoria a stato solido (SSD), memorie USB, stampanti e altro ancora. La porta bianca a destra di queste è una porta USB 2.0 a bassa velocità, che può essere utilizzata per collegare il mouse Raspberry Pi fornito insieme al Raspberry Pi 400 Personal Computer Kit.

L'ultima è una porta di rete Gigabit Ethernet, utilizzata per collegare il Raspberry Pi 400 alla rete utilizzando un cavo RJ45 in alternativa alla scheda Wi-Fi integrata nel dispositivo. Per saperne di più su come collegare il Raspberry Pi 400 ad una rete, consulta il Capitolo 2, *Introduzione a Raspberry Pi*.

Raspberry Pi Zero 2 W

Il Raspberry Pi Zero 2 W (**Figura 1-19**) è stato progettato per offrire molte delle funzionalità presenti negli altri modelli della gamma Raspberry Pi, ma in un design molto più compatto. È più economico e consuma meno energia, ma manca di alcune porte presenti sui modelli più grandi.

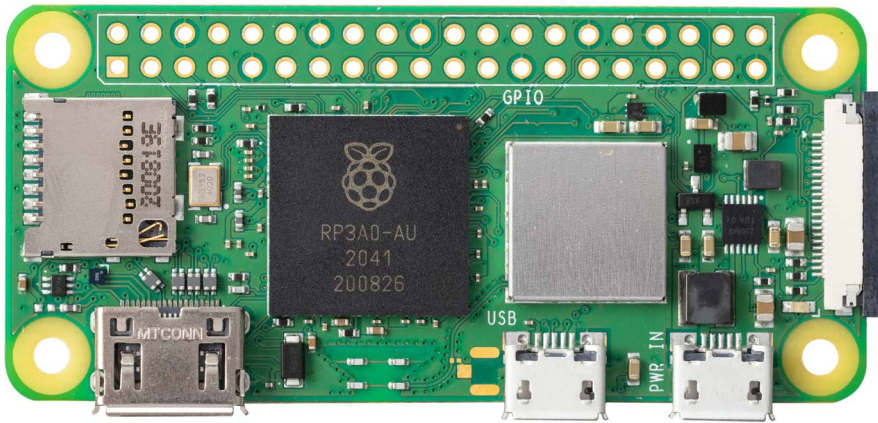


Figura 1-19 Raspberry Pi Zero 2 W

A differenza del Raspberry Pi 5 e del Raspberry Pi 400, il Raspberry Pi Zero 2 W non dispone di una porta Ethernet cablata. Puoi comunque collegarlo a una rete, ma solo utilizzando una connessione Wi-Fi. Per saperne di più su come connettere Raspberry Pi Zero 2 W a una rete, consulta il Capitolo 2, *Introduzione a Raspberry Pi*.

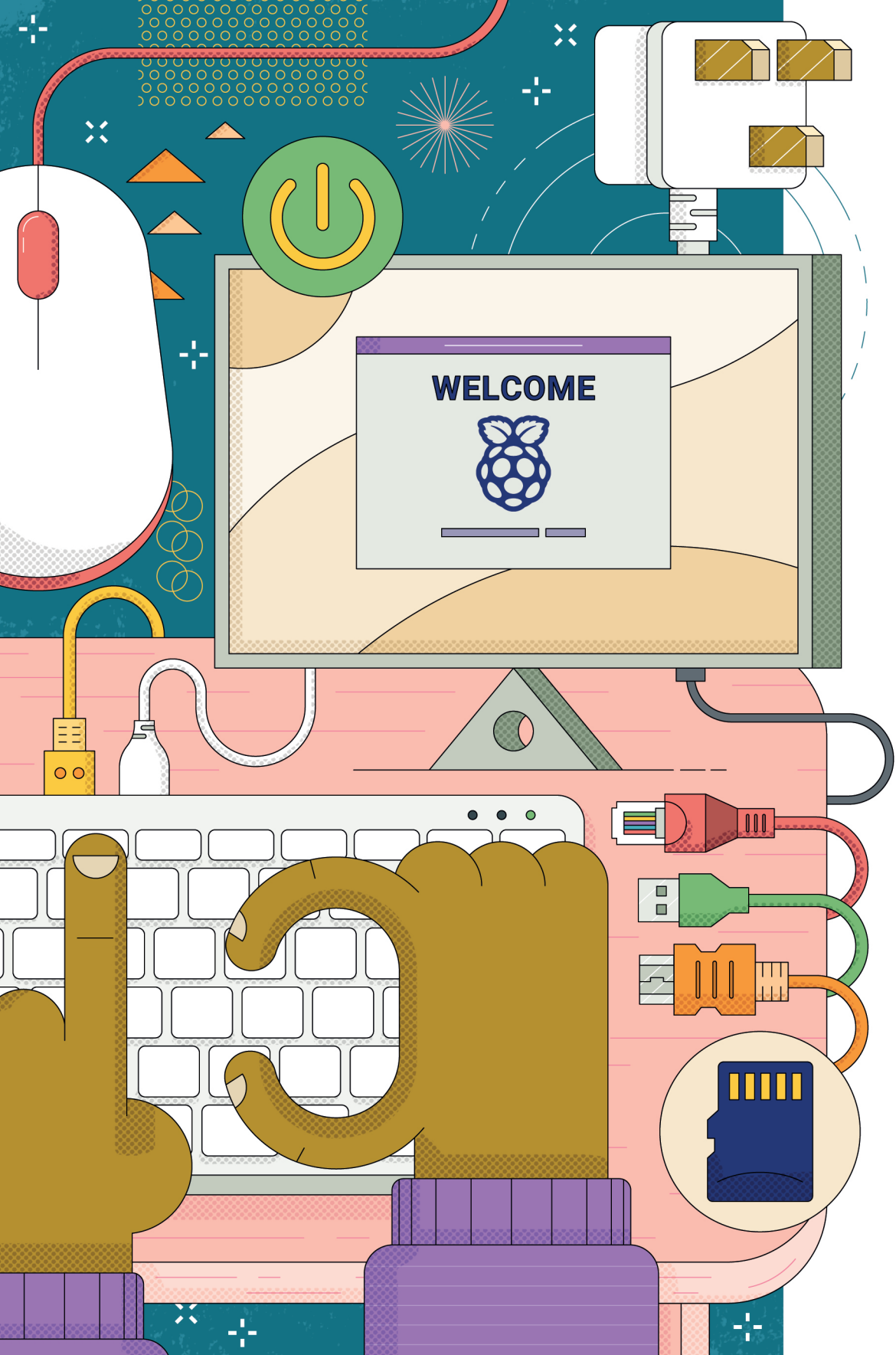
Puoi anche notare una differenza nel system-on-chip: è nero anziché argento e non è visibile alcun chip RAM separato. Questo perché i due componenti, il SoC e la RAM, sono uniti in un unico chip, contrassegnato dal logo Raspberry Pi inciso e collocato all'incirca al centro della scheda.

Sul bordo sinistro della scheda è presente il consueto slot per la scheda di archiviazione microSD, mentre al di sotto si trova una singola porta mini HDMI per video e audio. A differenza di Raspberry Pi 5 e Raspberry Pi 400, Raspberry Pi Zero 2 W supporta un solo display.

A destra troviamo due porte micro-USB: la porta di sinistra, contrassegnata dalla scritta "USB", è una porta USB On-The-Go (OTG) compatibile con gli adattatori OTG che consente di collegare tastiere, mouse, hub USB o altre periferiche; la porta di destra, contrassegnata dalla scritta "PWR IN", è il connettore per l'alimentazione. Non è possibile utilizzare un alimentatore progettato per Raspberry Pi 4 o Raspberry Pi 400 con Raspberry Pi Zero 2 W, poiché i connettori sono diversi.

Sul bordo destro della scheda è presente una Camera Serial Interface (CSI) che può essere utilizzata per collegare un modulo fotocamera Raspberry Pi. Puoi saperne di più consultando il Capitolo 8, *Moduli fotocamera di Raspberry Pi*.

Infine, il Raspberry Pi Zero 2 W presenta lo stesso connettore GPIO a 40 pin dei modelli più grandi, ma viene fornito *non popolato*. Ciò significa che il connettore non è assemblato. Se vuoi utilizzare il terminale GPIO, dovrai saldare un *connettore header 2×20* con passo 2,54 mm oppure acquistare un Raspberry Pi Zero 2 WH, che è dotato di un connettore già saldato.



Capitolo 2

Introduzione a Raspberry Pi

Scopri quali sono i componenti esterni essenziali per il tuo Raspberry Pi e come collegarli per farlo funzionare in modo ottimale.

Raspberry Pi è progettato per garantire la massima semplicità e velocità in termini di configurazione e utilizzo, ma come ogni computer si basa su una serie di componenti esterni, chiamati *periferiche*. È facile lasciarsi impressionare dal circuito stampato di un Raspberry Pi e temere di non riuscire a capirne il funzionamento a causa del suo aspetto così diverso da quello a cui siamo abituati con i classici computer con case, ma non devi preoccuparti. Seguendo le procedure dettagliate di questa guida, imparerai a utilizzarlo in meno di dieci minuti.

Se questa guida ti è stata fornita con un Kit Desktop di Raspberry Pi o con un Raspberry Pi 400, hai già quasi tutto quello che ti serve per iniziare. Procurati un monitor o un televisore con presa HDMI, lo stesso tipo di presa utilizzata da set-top box, lettori Blu-ray e console, in modo da vedere quali operazioni sta eseguendo il tuo Raspberry Pi.

Se invece hai acquistato un Raspberry Pi senza accessori, allora avrai bisogno anche di:

- **Alimentatore USB:** un alimentatore da 5 V con corrente nominale pari a 5 ampere (5 A) e connettore USB di tipo C per Raspberry Pi 5, un alimentatore da 5 V con corrente nominale pari a 3 ampere (3 A) e con un connettore USB di tipo C per Raspberry Pi 4 Model B o Raspberry Pi 400, o un alimentatore da 5 V con corrente nominale pari a 2,5 ampere (2,5 A) e con un connettore micro USB per Raspberry Pi Zero 2 W. Ti consigliamo di utilizzare gli alimentatori originali Raspberry Pi, poiché sono progettati per far fronte alle richieste di potenza che

variano velocemente del Raspberry Pi. Gli alimentatori di terze parti potrebbero non essere in grado di gestire in modo ottimale la corrente, causando problemi con l'alimentazione del Raspberry Pi.

- **Scheda microSD:** la scheda microSD è il sistema di archiviazione permanente di Raspberry Pi. Tutti i file creati e tutti i software installati, insieme al sistema operativo stesso, vengono memorizzati sulla scheda. Per iniziare, puoi usare una scheda da 8 GB e successivamente passare a quella da 16 GB, per avere più spazio. Il Kit Desktop di Raspberry Pi include una scheda microSD con Raspberry Pi OS già installato; consulta l'Appendice A, *Come installare un sistema operativo su una scheda microSD* per imparare a installare un sistema operativo (OS) su una scheda vuota.



Figura 2-1
Alimentatore USB

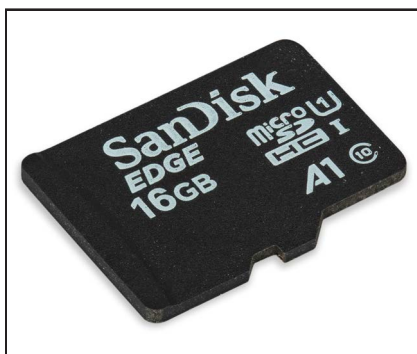


Figura 2-2
Scheda microSD

- **Tastiera e mouse USB:** la tastiera e il mouse consentono di controllare il tuo Raspberry Pi. Quasi tutte le tastiere e i mouse cablati o wireless con connettore USB sono compatibili con Raspberry Pi, anche se alcune tastiere per gaming con luci colorate assorbono troppa energia per poter essere utilizzate in modo affidabile. Raspberry Pi Zero 2 W richiede l'utilizzo di un adattatore micro USB OTG e, se vuoi collegare più di un dispositivo USB alla volta, avrai bisogno di un hub USB alimentato.
- **Cavo HDMI:** questo cavo è fondamentale per inviare l'audio e le immagini dal Raspberry Pi al tuo televisore o monitor. Per Raspberry Pi 4, Raspberry Pi 5 e Raspberry Pi 400 è necessario un cavo dotato di connettore micro HDMI su un'estremità, mentre per Raspberry Pi Zero 2 W è necessario un cavo con connettore mini HDMI; l'altra estremità deve essere dotata di un connettore HDMI full-size per il monitor. Puoi anche utilizzare un adattatore micro o mini HDMI-HDMI insieme a un cavo HDMI full-size standard. Se utilizzi un monitor senza presa HDMI,

puoi acquistare adattatori per la conversione in connettori DVI-D, DisplayPort o VGA.



Figura 2-3
Tastiera USB



Figura 2-4
Cavo HDMI

Raspberry Pi può essere utilizzato in tutta sicurezza anche senza case, a condizione che non lo si posizioni su una superficie metallica che potrebbe condurre elettricità e causare un cortocircuito. Tuttavia, l'utilizzo di un case garantisce una maggiore protezione. Il Kit Desktop include il case originale di Raspberry Pi, mentre i case di terze parti sono disponibili presso tutti i migliori rivenditori.

Se hai intenzione di utilizzare Raspberry Pi 4, Raspberry Pi 5 o Raspberry Pi 400 su rete cablata anziché su rete Wi-Fi, avrai bisogno anche di un cavo di rete, che dovrai collegare allo switch o al router della rete che vuoi utilizzare. Se invece stai pensando di utilizzare il modulo wireless integrato del Raspberry Pi, non avrai bisogno di un cavo, ma dovrai comunque sapere il nome e la chiave o la passphrase della tua rete wireless.

CONFIGURAZIONE DEL RASPBERRY PI 400

Le seguenti istruzioni consentono di configurare Raspberry Pi 5 o un altro dispositivo della famiglia Raspberry Pi. Per Raspberry Pi 400, consulta «Configurazione del Raspberry Pi 400» a pagina 27.



Come impostare l'hardware

Estrai il tuo Raspberry Pi dalla confezione. Come vedi, si tratta di un componente hardware robusto, ma ciò non significa che sia indistruttibile. Cerca di abituarti a tenere la scheda dai bordi piuttosto che dai lati e fai molta attenzione ai pin metallici rialzati. Se dovessero piegarsi, infatti, potrebbe essere più difficile utilizzare schede aggiuntive o altri componenti hardware e, nel peggiore dei casi, potrebbe verificarsi un cortocircuito che danneggerebbe il tuo Raspberry Pi.

Se non l'hai già fatto, dai un'occhiata al Capitolo 1, *Cos'è Raspberry Pi* per sapere dove si trovano esattamente le varie porte e la loro funzione.

Come montare il case di Raspberry Pi

Per installare Raspberry Pi 5 in un case, la prima cosa da fare è montarlo. Se utilizzi il case originale di Raspberry Pi, separa i tre singoli componenti: la base rossa, il gruppo ventola con il rispettivo telaio e il coperchio bianco.

Prendi la base e tienila in modo che l'estremità più alta sia a sinistra e quella più bassa a destra.

Tieni il tuo Raspberry Pi 5, senza scheda microSD inserita, dalla parte in cui si trovano le porte USB ed Ethernet, leggermente inclinato. Abbassa delicatamente l'altro lato per inserirlo nella base, come mostrato in **Figura 2-5**. Dovresti sentire uno scatto quando lo appoggi alla base.



INSTALLAZIONE DEL GRUPPO VENTOLA

All'estrazione dalla confezione, la ventola dovrebbe già essere inserita nel gruppo e quest'ultimo dovrebbe già essere inserito nel relativo telaio. In caso contrario, puoi semplicemente assemblare il tutto facendo scattare i vari componenti in sede (vedi **Figura 2-7**).

A questo punto, collega il connettore JST bianco della ventola alla presa della ventola su Raspberry Pi 5, come mostrato in **Figura 2-6**. Si inserisce solo in un senso, quindi non dovrai preoccuparti di collegarlo al contrario.



Figura 2-5
Raspberry Pi 5 posizionato nel suo case

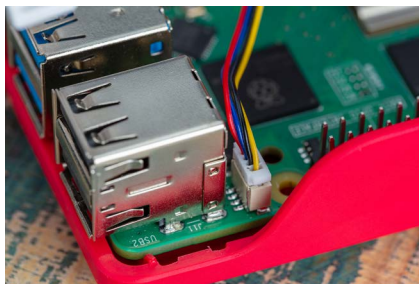


Figura 2-6
Collegamento del connettore della ventola

Fai scattare in posizione il gruppo ventola e il telaio come mostrato in **Figura 2-7** e spingi delicatamente verso il basso fino a sentire un clic.

Se vuoi coprire tutto con il case, prendi il coperchio bianco e posizionalo in modo che il logo Raspberry Pi si trovi sopra i connettori USB ed Ethernet del Raspberry Pi 5, come mostrato in **Figura 2-8**. Per fissarlo in posizione, spingi delicatamente verso il basso la parte centrale del coperchio fino a sentire uno scatto.



Figura 2-7
Montaggio del gruppo ventola e telaio



Figura 2-8
Posizionamento del coperchio sopra il case

HAT E COPERCHI

Puoi montare un HAT (Hardware Attached on Top) direttamente sopra Raspberry Pi 5 rimuovendo il gruppo ventola, oppure puoi impilarlo sopra il gruppo ventola e il telaio utilizzando distanziatori alti 14 mm e un estensore GPIO da 19 mm. Questi componenti sono venduti separatamente presso i rivenditori autorizzati.



Come montare il case del Raspberry Pi Zero

Per installare Raspberry Pi Zero 2 W in un case, la prima cosa da fare è montarlo. Se utilizzi il case originale del Raspberry Pi Zero, estrailo dalla confezione. Dovresti trovare quattro componenti: una base rossa e tre coperchi bianchi.

Se utilizzi Raspberry Pi Zero 2, dovrai utilizzare il coperchio solido. Se intendi utilizzare il connettore GPIO, di cui parliamo nel dettaglio nel Capitolo 6, *Physical computing con Scratch e Python*, installa il coperchio che presenta il foro rettangolare lungo. Se hai un Camera Module 1 o 2, installa il coperchio con il foro rotondo.

Poiché il Camera Module 3 e l'High Quality (HQ) Camera Module non sono compatibili con il coperchio fotocamera del case del Raspberry Pi Zero, dovrai utilizzarli al di fuori del case; all'estremità del case del Raspberry Pi Zero è presente un'apertura per il cavo della fotocamera.

Prendi la base e posizionala in piano sul tavolo, con le aperture delle porte rivolte verso di te, come mostrato in **Figura 2-9**.

Tenendo il Raspberry Pi Zero (con la scheda microSD inserita) dai bordi della scheda, allinealo in modo che i montanti circolari piccoli negli angoli della base si inseriscano nei fori di montaggio degli angoli del circuito stampato del Raspberry Pi Zero 2 W. Una volta allineati (**Figura 2-10**), spingi delicatamente il Raspberry Pi Zero 2 W verso il basso finché non senti un clic e le porte sono allineate con le aperture della base.



Figura 2-9
Il case di Raspberry Pi Zero



Figura 2-10
Posizionamento di Zero nel suo case

Prendi il coperchio bianco che hai scelto e posizionalo sopra la base del case del Raspberry Pi Zero, come mostrato in **Figura 2-11**. Se utilizzi il coperchio per il Camera Module, accertati che il cavo non rimanga intrappolato dentro. Quando il coperchio è in posizione, spingilo delicatamente verso il basso fino a farlo scattare in sede.

CAMERA MODULE E CASE DELLO ZERO

Se utilizzi un Camera Module Raspberry Pi, scegli il coperchio con il foro rotondo. Allinea i fori di montaggio del Camera Module con i montanti a croce del coperchio, in modo che il connettore della fotocamera sia rivolto verso il logo sul coperchio. Fallo scattare in sede. Allontana delicatamente la barra del connettore della fotocamera dal Raspberry Pi, quindi spingi l'estremità più sottile del cavo a nastro della fotocamera in dotazione nel connettore prima di spingere la barra in posizione. Collega l'estremità più larga del cavo al Camera Module nello stesso modo. Per ulteriori informazioni su come installare il Camera Module, consulta il Capitolo 8, *Moduli fotocamera di Raspberry Pi*.



A questo punto, puoi attaccare i piedini in gomma in dotazione sul fondo del case (vedi **Figura 2-12**): capovolgilo, rimuovi i piedini dal foglio di supporto e incollali nelle rientranze circolari sulla base per garantire una migliore presa.



Figura 2-11
Montaggio del coperchio



Figura 2-12
Montaggio dei piedini

Come installare la scheda microSD

Per installare la scheda microSD, la *memoria* del Raspberry Pi, capovolgi il Raspberry Pi (nel case, se utilizzato) e inserisci la scheda nello slot per microSD con l'etichetta rivolta verso l'alto. Si inserisce solo in un verso e dovrebbe scorrere in modo abbastanza fluido (consulta **Figura 2-13**).

La scheda microSD si inserirà nel connettore e si bloccherà senza fare clic.

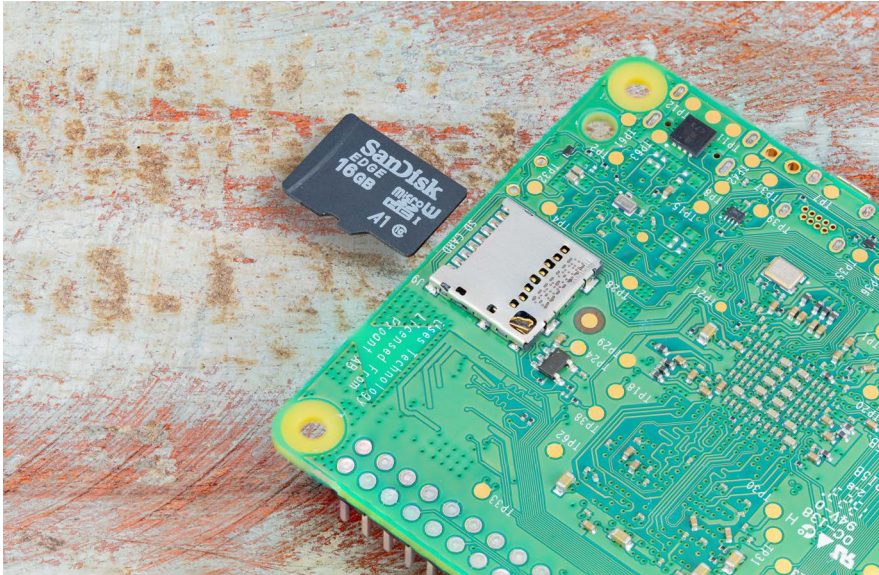


Figura 2-13 Inserimento della scheda microSD

Per il Raspberry Pi Zero 2 W, lo slot della scheda microSD si trova in alto sul lato sinistro. Inserisci la scheda con l'etichetta rivolta dalla parte opposta rispetto al Raspberry Pi.

Se in futuro vorrai rimuoverla, ti basterà estrarla delicatamente prendendola per l'estremità. Se usi un modello vecchio di Raspberry Pi, dovrai prima esercitare una leggera pressione sulla scheda per sbloccarla (nei Raspberry Pi 3, 4, 5 e qualsiasi modello di Raspberry Pi Zero non è necessario).

Come collegare una tastiera e un mouse

Collega il cavo USB della tastiera a una delle quattro porte USB (porta nera 2.0 o porta blu 3.0) del Raspberry Pi come mostrato in **Figura 2-14**. Se utilizzi la tastiera originale di Raspberry Pi, sul retro è presente una porta USB per il

mouse. Altrimenti, è sufficiente collegare il cavo USB del mouse a un'altra porta USB del Raspberry Pi.



Figura 2-14 Collegamento del cavo USB al Raspberry Pi 5

Per il Raspberry Pi Zero 2 W, dovrai utilizzare un cavo adattatore micro USB OTG. Inseriscilo nella porta micro USB di sinistra, quindi collega il cavo USB della tastiera all'adattatore USB OTG.

Se utilizzi una tastiera con mouse separato, piuttosto che una con touchpad integrato, dovrai utilizzare anche un hub USB alimentato. Collega il cavo adattatore micro USB OTG come indicato sopra, quindi collega il cavo USB dell'hub all'adattatore USB OTG prima di collegare la tastiera e il mouse all'hub USB. Infine, collega l'alimentatore dell'hub e accendilo.

I connettori USB per la tastiera e il mouse dovrebbero inserirsi facilmente; se devi forzare il connettore, allora qualcosa non va. Controlla che il connettore USB sia rivolto verso l'alto.

TASTIERA E MOUSE

La tastiera e il mouse sono il mezzo principale per dire al Raspberry Pi cosa fare. In ambito informatico, vengono chiamati *dispositivi di input*, al contrario del monitor che è un *dispositivo di output*.



Come collegare un monitor

Per Raspberry Pi 4 e Raspberry Pi 5, prendi il cavo micro HDMI e collega l'estremità più piccola alla porta micro HDMI più vicina alla porta USB di tipo C del Raspberry Pi come mostrato in **Figura 2-15**. Collega l'altra estremità al display.

Per Raspberry Pi Zero 2 W (**Figura 2-16**), prendi il cavo mini HDMI e collega l'estremità più piccola alla porta mini HDMI sul lato sinistro del Raspberry Pi, sotto lo slot della scheda microSD. L'altra estremità va collegata al monitor.



Figura 2-15
Collegamento del cavo HDMI al Raspberry Pi 5



Figura 2-16
Collegamento del cavo HDMI al Raspberry Pi Zero

Se il monitor ha più di una porta HDMI, cerca un numero di porta accanto al connettore: per vedere il monitor del Raspberry Pi dovrai commutare il televisore su questo ingresso. Se non trovi il numero di porta, non preoccuparti: prova ogni ingresso finché non trovi quello del Raspberry Pi.



COLLEGAMENTO TV

Se il tuo televisore o monitor non ha un connettore HDMI, non significa che tu non possa usare Raspberry Pi. I cavi adattatori, disponibili presso qualsiasi rivenditore di elettronica, consentono di convertire la porta micro o mini HDMI del Raspberry Pi in DVI-D, DisplayPort o VGA per l'uso con monitor di altri computer.

Come collegare un cavo di rete (opzionale)

Per collegare il Raspberry Pi a una rete cablata, prendi un cavo Ethernet e inseriscilo nella porta Ethernet del Raspberry Pi, con la linguetta in plastica rivolta verso il basso, fino a sentire un clic (vedi **Figura 2-17**). Per rimuovere il cavo, premi la linguetta verso l'interno in direzione della presa ed estrailo delicatamente.

L'altra estremità del cavo di rete deve essere collegata a una porta libera dell'hub di rete, dello switch o del router nello stesso modo.

Come collegare l'alimentazione

Collegare il Raspberry Pi all'alimentatore è la fase finale del processo di configurazione dell'hardware. È l'ultima cosa da fare prima di iniziare a configurare il software. Il Raspberry Pi si accende non appena viene collegato a un'alimentazione elettrica.

Per Raspberry Pi 4 e Raspberry Pi 5, collegare l'estremità USB di tipo C del cavo di alimentazione al connettore di alimentazione USB di tipo C del Raspberry Pi, come mostrato in **Figura 2-18**. Può essere inserito in entrambi i sensi e dovrebbe entrare facilmente. Se il tuo alimentatore ha un cavo scollegabile, assicurati che l'altra estremità sia inserita nel corpo dell'alimentatore.

ATTENZIONE

Raspberry Pi 5 ha bisogno di un alimentatore da 5 V in grado di erogare una corrente di 5 A e di un cavo USB di tipo C adeguato contrassegnato E-Mark. Se colleghi un alimentatore a bassa corrente, compreso l'alimentatore originale del Raspberry Pi 4, le porte USB del Raspberry Pi 5 saranno limitate ai soli dispositivi a bassa potenza.



Figura 2-17
Collegamento del Raspberry Pi 5 all'Ethernet

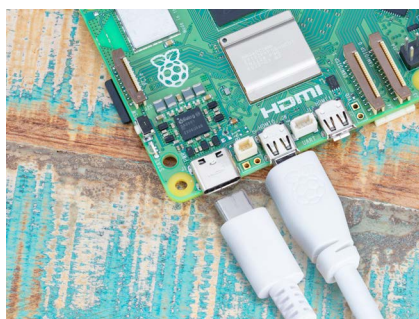


Figura 2-18
Accensione del Raspberry Pi 5

Per Raspberry Pi Zero 2 W, collega l'estremità micro USB del cavo di alimentazione alla porta micro USB destra del Raspberry Pi. Può essere inserita solo in un senso, quindi assicurati di verificarne l'orientamento prima di inserire il cavo.

Complimenti, hai montato correttamente il tuo Raspberry Pi!

Infine, collega l'alimentatore a una presa di corrente e accendila: il Raspberry Pi inizierà a funzionare immediatamente.

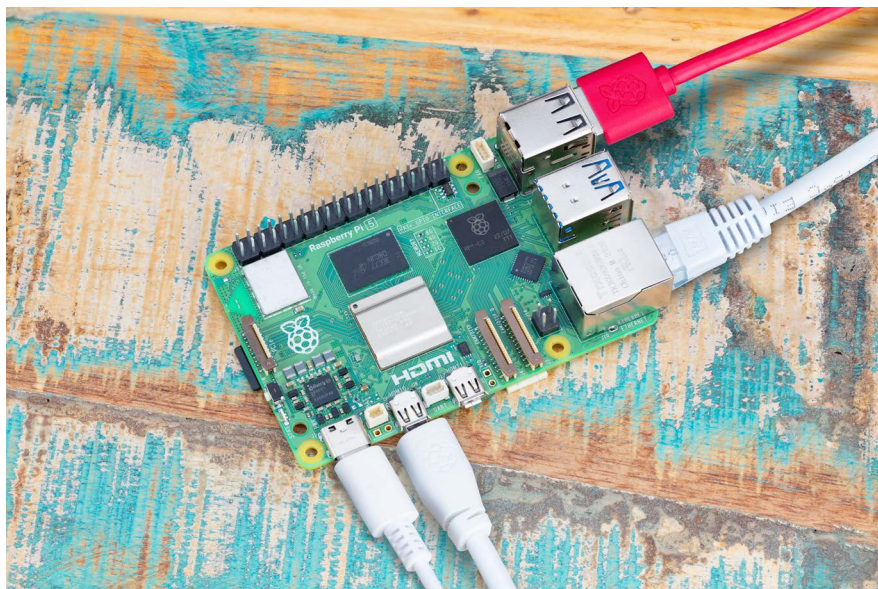


Figura 2-19 Raspberry Pi è pronto per essere utilizzato

Vedrai brevemente un cubo color arcobaleno seguito da una schermata informativa con il logo Raspberry Pi. È possibile che venga visualizzata una schermata blu mentre il sistema operativo si ridimensiona per sfruttare tutto il potenziale della scheda microSD. Se vedi una schermata nera, aspetta qualche minuto: al primo avvio, il Raspberry Pi eseguirà un po' di pulizie in background e questa operazione potrebbe richiedere del tempo.



Figura 2-20 Procedura guidata di benvenuto di Raspberry Pi OS

Dopo un po' verrà visualizzata la procedura guidata di benvenuto di Raspberry Pi OS, come mostrato in **Figura 2-20**. Il sistema operativo è ora pronto per essere configurato, operazione illustrata nel Capitolo 3, *Come utilizzare Raspberry Pi*.

Configurazione del Raspberry Pi 400

A differenza del Raspberry Pi 4, il Raspberry Pi 400 è dotato di una tastiera integrata e della scheda microSD già installata. Dovrai comunque collegare alcuni cavi all'inizio, ma ci vorranno solo pochi minuti.

Come collegare un mouse

La tastiera del Raspberry Pi 400 è già collegata, quindi ti resta da collegare solo il mouse. Prendi il cavo USB all'estremità del mouse e inseriscilo in una delle tre porte USB (2.0 o 3.0) sul retro del Raspberry Pi 400. Se desideri lasciare libere le due porte USB 3.0 ad alta velocità per altri accessori, utilizza la porta bianca USB 2.0.

Il connettore USB dovrebbe inserirsi facilmente (vedi **Figura 2-21**). Se devi forzarlo, allora qualcosa non va. Controlla che il connettore USB sia rivolto verso l'alto.

Come collegare un monitor

Prendi il cavo micro HDMI e collega l'estremità più piccola alla porta micro HDMI più vicina allo slot della scheda microSD del Raspberry Pi 400 e l'altra estremità al tuo monitor, come mostrato in **Figura 2-22**. Se il monitor ha più di una porta HDMI, cerca un numero di porta accanto al connettore: per vedere il monitor del Raspberry Pi dovrai commutare il televisore o il monitor su questo ingresso. Se non trovi il numero di porta, non preoccuparti: prova ogni ingresso finché non trovi quello del Raspberry Pi.



Figura 2-21
Collegamento di un cavo USB al Raspberry Pi 400



Figura 2-22
Collegamento del cavo HDMI al Raspberry Pi 400

Come collegare un cavo di rete (opzionale)

Per collegare il Raspberry Pi 400 a una rete cablata, prendi un cavo Ethernet e inseriscilo nella porta corrispondente del Raspberry Pi 400, con la linguetta in plastica rivolta verso l'alto, fino a sentire un clic (**Figura 2-23**). Per rimuovere il cavo, premi la linguetta verso l'interno in direzione della presa ed estrailo delicatamente.



Figura 2-23 Collegamento del Raspberry Pi 400 all'Ethernet

L'altra estremità del cavo di rete deve essere collegata a una porta libera dell'hub di rete, dello switch o del router nello stesso modo.

Come collegare l'alimentazione

Collegare il Raspberry Pi 400 a un alimentatore è l'ultimo passo del processo di configurazione dell'hardware e dovresti farlo solo quando sei pronto a configurare il software. Il Raspberry Pi 400 non ha un interruttore e si accende non appena viene collegato a un alimentatore sotto tensione.

Per prima cosa, collega l'estremità USB di tipo C del cavo di alimentazione al connettore di alimentazione USB di tipo C del Raspberry Pi. Può essere inserito in entrambi i sensi e dovrebbe entrare facilmente. Se il tuo alimentatore ha un cavo scollegabile, assicurati che l'altra estremità sia inserita nel corpo dell'alimentatore.

Infine, collega l'alimentatore a una presa di corrente e accendila: il Raspberry Pi 400 inizierà a funzionare immediatamente. Complimenti, hai montato correttamente il tuo Raspberry Pi 400 (**Figura 2-24**)!



Figura 2-24 Il tuo Raspberry Pi 400 è montato correttamente

Vedrai brevemente un cubo color arcobaleno seguito da una schermata informativa con il logo Raspberry Pi. È possibile che venga visualizzata una schermata blu mentre il sistema operativo si ridimensiona per sfruttare tutto il potenziale della scheda microSD. Se vedi una schermata nera, aspetta qualche minuto: al primo avvio, il Raspberry Pi eseguirà un po' di pulizie in background e questa operazione potrebbe richiedere del tempo.

Dopo un po' verrà visualizzata la procedura guidata di benvenuto di Raspberry Pi OS, come mostrato prima in **Figura 2-20**. Il sistema operativo è ora pronto per essere configurato, operazione illustrata nel Capitolo 3, *Come utilizzare Raspberry Pi*.



Capitolo 3

Come utilizzare Raspberry Pi

Scopri tutto sul sistema operativo Raspberry Pi.

Raspberry Pi consente di utilizzare un'ampia gamma di software e diversi sistemi operativi, ossia il software principale che fa funzionare un computer. Il sistema operativo più rinomato, nonché ufficiale della Raspberry Pi, è Raspberry Pi OS. Basato su Debian Linux, è pensato appositamente per Raspberry Pi e viene fornito con una serie di software extra preinstallati e pronti all'uso.

Se finora hai utilizzato solo Microsoft Windows o Apple macOS, non preoccuparti: Raspberry Pi OS è basato sugli stessi elementi intuitivi di finestre, icone, menu e puntatore (WIMP), così potrai acquisire rapidamente familiarità.

Continua a leggere per saperne di più su alcuni dei software in dotazione.

Welcome Wizard

La prima volta che esegui Raspberry Pi OS, viene avviata una procedura guidata chiamata Welcome Wizard (**Figura 3-1**). Questo strumento ti guiderà attraverso la modifica di alcune impostazioni in Raspberry Pi OS, per *configurare* il Raspberry Pi in modo tale che sia adatto all'utilizzo che ne vuoi fare.



Figura 3-1 Welcome Wizard

Fai clic sul pulsante **Next**, quindi scegli il paese, la lingua e il fuso orario selezionandoli dagli elenchi a discesa (**Figura 3-2**). Se utilizzi una tastiera con layout statunitense, fai clic sulla casella per assicurarti che il layout utilizzato in Raspberry Pi OS sia quello corretto. Se vuoi che il desktop e i programmi vengano visualizzati in inglese, indipendentemente dalla lingua del tuo paese, fai clic per spuntare la casella **Use English language**. Una volta terminato, fai clic su **Next**.



Figura 3-2 Selezione della lingua e altre opzioni

La schermata successiva chiede di scegliere un nome e una password per l'account utente (**Figura 3-3**). Scegli un nome: può essere qualsiasi cosa, ma deve iniziare con una lettera e può contenere solo lettere minuscole, cifre e trattini. Poi dovrai creare una password che puoi ricordare facilmente. Ti verrà chiesto di digitare la password due volte per assicurarti di non aver commesso errori che potrebbero bloccare l'accesso al nuovo account. Quando sei soddisfatto delle scelte effettuate, fai clic su **Next**.

Creazione utente

E' necessario creare un account utente per entrare nel Raspberry Pi.

Il nome utente può contenere solo lettere minuscole, numeri e trattini; inoltre deve iniziare con una lettera.

Inserire il nome utente: gareth

Inserire la password:

Confermare la password:

☒ Nascondi caratteri

Premere «Avanti» per creare l'account.

Indietro Avanti

Figura 3-3 Impostazione di una nuova password

Nella schermata seguente potrai scegliere la rete Wi-Fi da un elenco (**Figura 3-4**).

Selezione della rete WiFi

Selezionare la rete WiFi dalla lista.

Beam 2 27A0E		
DIRECT-F8-HP DeskJet 2600 series		
PLUSNET-5TCR		
PLUSNET-5TCR_EXT		
TNCAPBA133B		
Wifi_SVVEXQ8EW9		

Premere «Avanti» per collegarsi oppure «Salta» per continuare senza connessione.

Indietro Salta Avanti

Figura 3-4 Selezione della rete wireless

RETE WIRELESS

La rete wireless integrata è disponibile solo nei modelli Raspberry Pi 3, Raspberry Pi 4, Raspberry Pi 5 e nelle famiglie Raspberry Pi Zero W e Zero 2 W. Se utilizzi un altro modello di Raspberry Pi con una rete wireless, dovrai utilizzare un adattatore USB Wi-Fi.



Scorri l'elenco delle reti con il mouse o la tastiera finché non trovi la tua, quindi selezionala e fai clic su **Next**. Se la tua rete wireless è sicura, come dovrebbe essere, ti verrà chiesta la password (nota anche come chiave predefinita). Se non utilizzi una password personalizzata, quella predefinita è in genere scritta su una tessera fornita insieme al router, oppure sul fondo o sul retro del rou-

ter stesso. Fai clic su **Next** per collegarti alla rete. Se non desideri connetterti a una rete wireless, fai clic su **Skip**.

Successivamente ti verrà chiesto di scegliere il *browser web* predefinito tra i due preinstallati in Raspberry Pi OS: Chromium di Google, il predefinito, e Firefox di Mozilla (**Figura 3-5**). Per ora, puoi lasciare Chromium selezionato come predefinito, in modo da poter attenerti alle istruzioni riportate in questa guida; puoi sempre passare a Firefox in un secondo momento, se lo preferisci. Se cambi il browser predefinito, puoi anche scegliere di disinstallare il browser non predefinito per liberare spazio sulla scheda microSD. Ti basterà selezionare la casella quando trovi l'opzione e fare clic sul pulsante **Next**.



Figura 3-5 Selezione di un browser

Nella schermata successiva potrai controllare e installare gli aggiornamenti per Raspberry Pi OS e gli altri software presenti su Raspberry Pi (**Figura 3-6**). Raspberry Pi OS viene regolarmente aggiornato per correggere bug, aggiungere nuove funzionalità e migliorare le prestazioni. Per installare questi aggiornamenti, fai clic su **Next**. Altrimenti, fai clic su **Skip**. Il download degli aggiornamenti può richiedere diversi minuti, quindi ti chiediamo di avere un po' di pazienza.

Al termine dell'installazione, visualizzerai una finestra che indica l'avvenuto aggiornamento, quindi fai clic sul pulsante **OK**.

L'ultima schermata di Welcome Wizard (**Figura 3-7**) fornisce un'ultima informazione utile: alcune modifiche apportate verranno applicate solo al riavvio del Raspberry Pi. Fai clic sul pulsante **Restart** e il Raspberry Pi si riavvierà. Da questo momento in poi, non visualizzerai più Welcome Wizard e potrai iniziare a utilizzare il tuo Raspberry Pi.

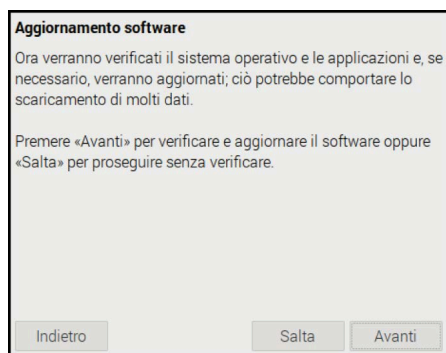


Figura 3-6 Controllo degli aggiornamenti



Figura 3-7 Riavvio del Raspberry Pi

ATTENZIONE

Se, dopo l'avvio del Raspberry Pi 5, viene visualizzato un messaggio in alto a destra che dice "this power supply is not capable of supplying 5A", significa che l'alimentatore che non è in grado di fornire i 5 V a 5 A necessari. Dovrai sostituire l'alimentatore con uno che supporti il funzionamento del Raspberry Pi 5, come il suo alimentatore ufficiale. Puoi ignorare l'avviso, ma alcuni dispositivi USB ad alta potenza non funzioneranno.

Se viene visualizzato il messaggio "low voltage warning", accompagnato dal simbolo di un fulmine, è necessario interrompere l'utilizzo del Raspberry Pi fino alla sostituzione dell'alimentatore: i cali di tensione degli alimentatori di bassa qualità possono causare il blocco del Raspberry Pi.



Il desktop

La versione di Raspberry Pi OS installata sulla maggior parte delle schede Raspberry Pi è conosciuta come "Raspberry Pi OS con desktop" relativamente all'interfaccia grafica (**Figura 3-8**). La maggior parte del desktop è occupata da una foto, ossia uno sfondo (**A** in **Figura 3-8**), sulla quale visualizzerai i programmi che esegui.

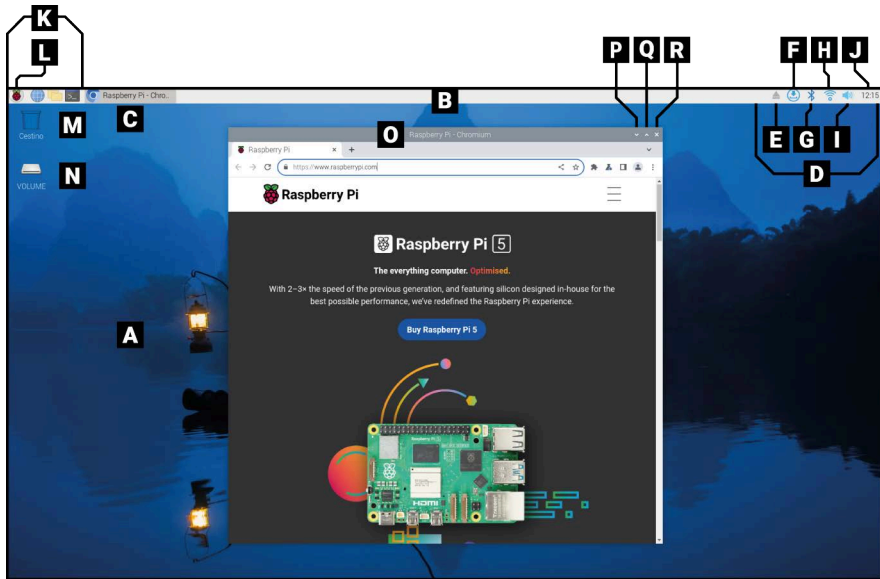


Figura 3-8 Il desktop di Raspberry Pi OS

- | | |
|---------------------------------------|--|
| A Sfondo | J Orologio |
| B Barra delle applicazioni | K Launcher |
| C Attività | L Menu (o icona Raspberry Pi) |
| D Area di notifica | M Icona cestino |
| E Espulsione media | N Icona unità rimovibile |
| F Icona aggiornamento software | O Barra del titolo della finestra |
| G Icona Bluetooth | P Riduci a icona |
| H Icona di rete | Q Ingrandisci |
| I Icona volume | R Chiudi |

Nella parte superiore del desktop è presente una barra delle applicazioni (**B**), che consente di avviare i programmi installati. Questi sono indicati come attività (**C**) nella barra delle applicazioni. Il lato destro della barra dei menu ospita l'*area di notifica* (**D**). Quando colleghi un *dispositivo di archiviazione rimovibile* (come ad esempio una chiavetta USB) al Raspberry Pi, vedrai un simbolo di espulsione (**E**): fai clic su questo simbolo per espellere e rimuovere il dispositivo in modo sicuro. L'icona di aggiornamento del software (**F**) viene visualizzata solo quando ci sono aggiornamenti di Raspberry Pi OS e delle sue applicazioni. L'orologio si trova a destra (**J**), facendovi clic sopra potrai visualizzare un calendario digitale (**Figura 3-9**).

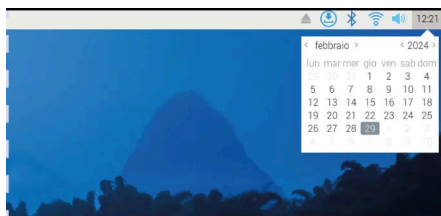


Figura 3-9 Calendario digitale

Accanto a questa icona c'è quella dell'altoparlante (**I**). Fai clic con il tasto sinistro del mouse per regolare il volume di Raspberry Pi oppure con il tasto destro per scegliere l'uscita audio. Accanto c'è un'icona di rete (**H**): se sei collegato a una rete wireless visualizzerai la potenza del segnale come una serie di barre, mentre se sei collegato a una rete cablata vedrai solo due frecce. Facendo clic sull'icona di rete visualizzerai un elenco di reti wireless presenti nelle vicinanze (**Figura 3-10**), mentre facendo clic sull'icona Bluetooth (**G**) accanto potrai connetterti a un dispositivo Bluetooth nelle vicinanze.

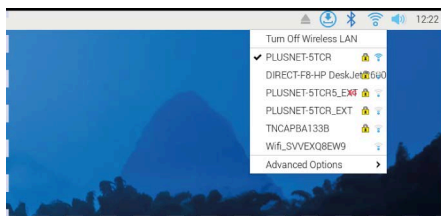


Figura 3-10 Elenco delle reti wireless vicine

Il menu *launcher* (**K**) si trova nella parte sinistra della barra dei menu, qui troverai i programmi installati con Raspberry Pi OS. Alcuni di questi sono visibili come icone di collegamento, mentre altri si trovano nel menu, che potrai richiamare facendo clic sull'icona di Raspberry Pi (**L**) a sinistra (**Figura 3-11**).

I programmi nel menu sono suddivisi in categorie. Il nome di ciascuna categoria indica cosa ospita: la categoria **Programmazione** contiene il software

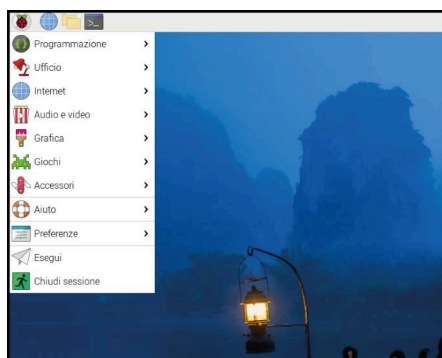


Figura 3-11 Il menu di Raspberry Pi

progettato per aiutarti a scrivere i tuoi programmi, come spiegato nel Capitolo 4, *Programmare con Scratch 3*, mentre i Giochi ti faranno passare il tempo.

In questa guida non verranno illustrati tutti i programmi: ti invitiamo a esplorarli e a sperimentare per conoscerli meglio. Sul desktop si trova il Cestino (**M**) e i dispositivi di archiviazione esterni (**N**) collegati a Raspberry Pi.

Il browser Chromium

Per fare pratica con il tuo Raspberry Pi, inizia aprendo il browser Web Chromium: fai clic sull'icona di Raspberry Pi in alto a sinistra per aprire il menu, sposta il puntatore del mouse per selezionare la categoria Internet e fai clic su **Browser web Chromium** per aprirlo.

Se hai già usato il browser Chrome di Google su un altro computer, Chromium ti sarà immediatamente familiare. Come browser Web, Chromium ti consente di visitare siti Web, guardare video, giocare e persino di comunicare con persone in tutto il mondo su forum e siti di chat.

Inizia a usare Chromium ingrandendo la finestra in modo che occupi più spazio sullo schermo: cerca le tre icone in alto a destra nella barra del titolo della finestra Chromium (**O**) e fai clic sull'icona della freccia in alto al centro (**Q**). Questo è il pulsante di *ingrandimento*. A sinistra del pulsante per ingrandire è presente quello per *ridurre a icona* (**P**), con cui potrai nascondere la finestra fino a che non farai clic su di essa nella barra delle applicazioni nella parte superiore dello schermo. La crocetta a destra del pulsante per ingrandire serve per *chiudere* (**R**) e fa esattamente quello che ci si aspetta: chiude la finestra.

CHIUDI E SALVA

Chiudere una finestra prima di aver salvato il lavoro svolto non è una buona idea, infatti non tutti i programmi avvertono di salvare quando fai clic sul pulsante di chiusura.



La prima volta che esegui il browser Web Chromium, il sito Web di Raspberry Pi dovrebbe aprirsi automaticamente, come mostrato in **Figura 3-12**. Se non succede (o se desideri visitare altri siti Web), fai clic nella barra degli indirizzi nella parte superiore della finestra di Chromium (la grande barra bianca con una lente d'ingrandimento sul lato sinistro) e digita **raspberrypi.com** o l'indirizzo del sito Web che vuoi visitare, quindi premi il tasto **INVIO** sulla tastiera. Verrà caricato il sito Web di Raspberry Pi.

Nella barra puoi digitare anche termini di ricerca: prova a cercare "Raspberry Pi", "Raspberry Pi OS" o "Retro gaming".

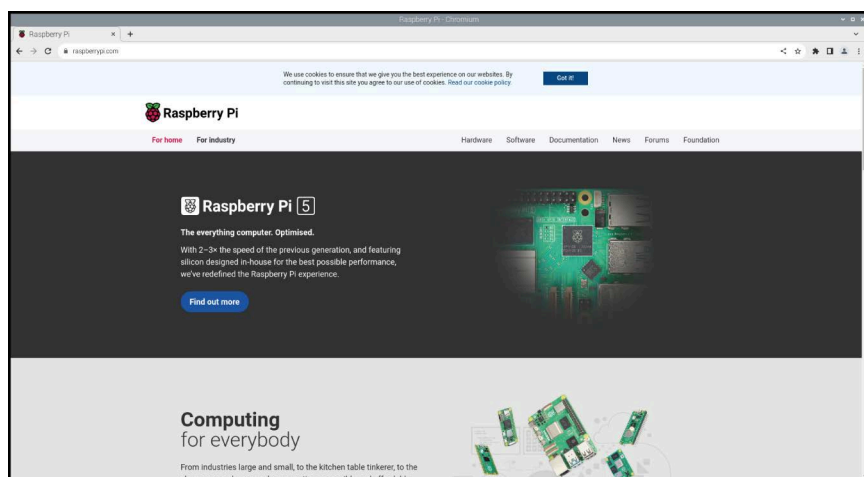


Figura 3-12 Il sito Web di Raspberry Pi in Chromium

Al primo avvio di Chromium potrebbero caricarsi alcune *schede* nella parte superiore della finestra. Per passare a un'altra scheda, fai clic su di essa; per chiudere una scheda senza chiudere Chromium, fai clic sulla crocetta in alto a destra nella scheda che vuoi chiudere.

Per aprire più siti Web senza dover passare da una finestra di Chromium all'altra, fai clic sul pulsante della scheda a destra dell'ultima scheda aperta oppure tieni premuto il tasto **CTRL** sulla tastiera, quindi premi il tasto **T** prima di rilasciare **CTRL**.

Quando hai finito di usare Chromium, fai clic sul pulsante di chiusura nell'angolo in alto a destra della finestra.

Il File Manager

I file che salvi, ad esempio programmi, video, immagini, vengono memorizzati nella *home directory*. Per visualizzare la home directory, fai clic sull'icona di Raspberry Pi per aprire il menu, sposta il puntatore del mouse per selezionare **Accessori**, quindi fai clic su **Gestore file PCManFM** per aprirlo (Figura 3-13).

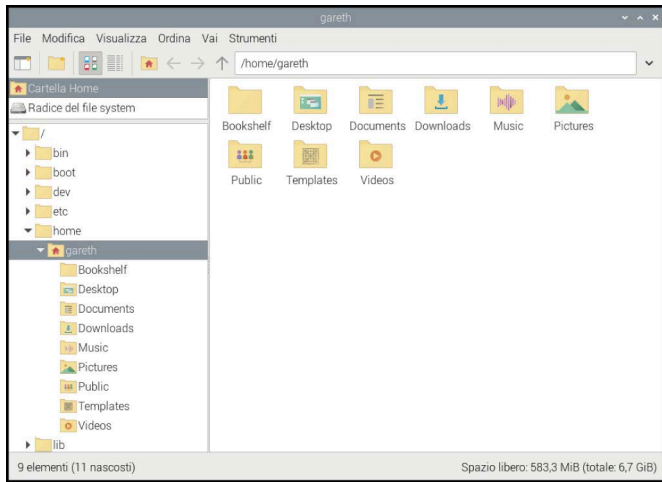


Figura 3-13 Il File Manager

Con il File Manager potrai sfogliare i file e le cartelle (*directory*) presenti nella scheda microSD di Raspberry Pi, nonché in qualsiasi dispositivo di archiviazione rimovibile (come chiavette USB) collegato alle porte USB di Raspberry Pi. Alla prima apertura, accede automaticamente alla home directory. Qui troverai una serie di cartelle, o *sottodirectory*, organizzate in categorie, proprio come il menu. Le sottodirectory principali sono:

- ▶ **Bookshelf:** contiene le copie digitali di libri e riviste della Raspberry Pi Press. Puoi leggerli e scaricarne altri con l'applicazione Bookshelf nella sezione Guida del menu.
- ▶ **Desktop:** questa cartella contiene ciò che vedi quando carichi Raspberry Pi OS per la prima volta. Se salvi un file qui, lo vedrai sul desktop e sarà facile trovarlo e aprirlo.
- ▶ **Documents:** la cartella in cui memorizzerai la maggior parte dei file di testo che creerai, dai racconti alle ricette.

- ▶ **Downloads:** quando scarichi un file da Internet con il browser Web Chromium, viene automaticamente salvato in Download.
- ▶ **Music:** qualsiasi file musicale che crei o scarichi può essere salvato qui.
- ▶ **Pictures:** questa cartella è dedicata alle immagini o, in termini tecnici, *file di immagini*.
- ▶ **Public:** mentre gli altri file sono privati, tutto ciò che salverai in Public sarà disponibile per gli altri utenti del tuo Raspberry Pi, anche se hanno il loro nome utente e password.
- ▶ **Templates:** questa cartella contiene tutti i modelli e i documenti vuoti con un layout o una struttura di base predefiniti creati da te o installati dalle tue applicazioni.
- ▶ **Videos:** una cartella per i video, nonché la prima cartella in cui cercheranno i programmi di riproduzione video.

La finestra File Manager è divisa in due riquadri principali: quello di sinistra mostra le directory presenti sul tuo Raspberry Pi, mentre quello di destra i file e le sottodirectory della directory selezionata a sinistra.

Quando colleghi un dispositivo di archiviazione rimovibile alla porta USB del Raspberry Pi, verrà visualizzata una finestra pop-up che chiede se desideri aprirlo in File Manager (**Figura 3-14**). Fai clic su **OK** per visualizzarne file e directory.

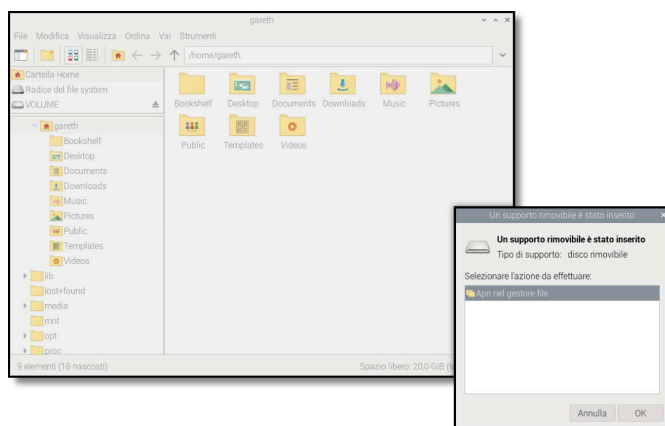


Figura 3-14 Inserimento di un dispositivo di archiviazione rimovibile

Puoi facilmente *trascinare e rilasciare* i file tra la scheda microSD del Raspberry Pi e un dispositivo rimovibile. Con l'home directory e il dispositivo rimovibile aperti in finestre File Manager separate, sposta il puntatore del mouse sul file

che vuoi copiare, fai clic e tieni premuto il tasto sinistro del mouse, quindi trascina il puntatore sull'altra finestra e rilascia il tasto del mouse (Figura 3-15).

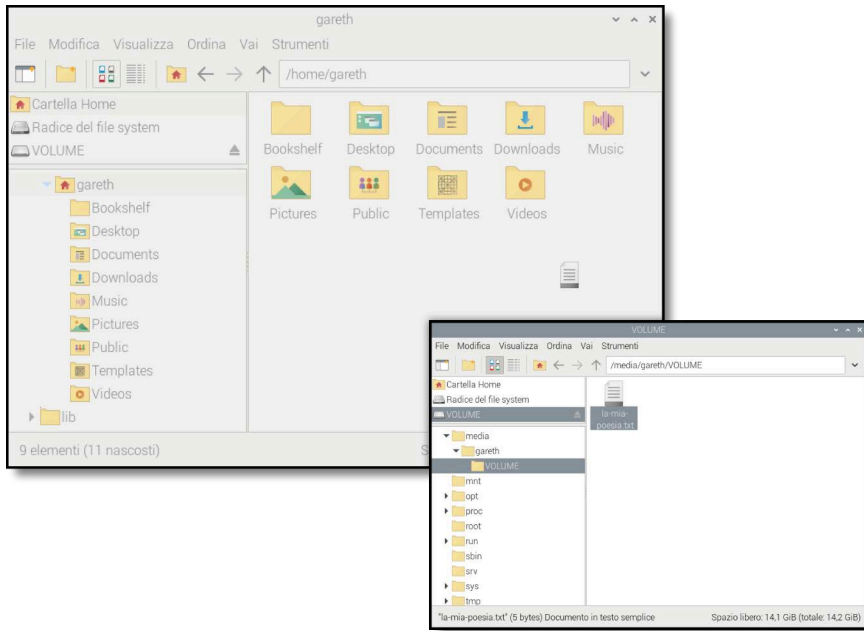


Figura 3-15 Trascinamento di un file

Un modo semplice per copiare un file è fare clic una volta sul file, fare clic sul menu **Modifica**, fare clic su **Copia**, fare clic sull'altra finestra, quindi fare clic sul menu **Modifica** e su **Incolla**.

L'opzione Taglia, disponibile anche nel menu Modifica, è simile, tranne per il fatto che cancella il file dalla sua posizione originale dopo averlo copiato. Entrambe le opzioni possono essere utilizzate anche attraverso le seguenti combinazioni di tasti di scelta rapida: **CTRL+C** per copiare o **CTRL+X** per tagliare e **CTRL+V** per incollare.



TASTI DI SCELTA RAPIDA

Quando vedi un tasto di scelta rapida come **CTRL+C**, significa che devi tenere premuto il primo tasto sulla tastiera (**CTRL**), quindi premere il secondo (**C**) e rilasciarli.

Quando hai finito di sperimentare, chiudi il File Manager facendo clic sul pulsante di chiusura della finestra, in alto a destra. Se hai più di una finestra aperta, chiudile tutte. Se hai collegato un dispositivo di archiviazione rimovibile a Raspberry Pi, espellilo facendo clic sul pulsante di espulsione in alto

a destra dello schermo, quindi cerca il dispositivo nell'elenco e selezionalo prima di scollegarlo.

ESPULSIONE DEI DISPOSITIVI

Utilizza sempre il pulsante di espulsione prima di scollegare un dispositivo di archiviazione esterno. Se non lo fai, i file in esso contenuti potrebbero danneggiarsi e diventare inutilizzabili.



Lo strumento Recommended Software

Raspberry Pi OS viene fornito con un'ampia gamma di software già installati, ma il tuo Raspberry Pi è compatibile con ancora più software. Nello strumento Recommended Software è disponibile una selezione dei migliori programmi.

Per utilizzarlo è necessaria una connessione a Internet. Se il tuo Raspberry Pi è collegato, fai clic sull'icona di Raspberry Pi, sposta il puntatore del mouse su **Preferenze** e fai clic su **Software consigliato**. Lo strumento verrà caricato e inizierà a scaricare le informazioni sul software disponibile.

Dopo qualche secondo visualizzerai un elenco di pacchetti software compatibili (**Figura 3-16**). Sono organizzati in varie categorie, proprio come il software nel menu di Raspberry Pi. Fai clic su una categoria nel riquadro a sinistra per visualizzare i software di quella categoria oppure fai clic su **Tutti i programmi** per visualizzare tutto.

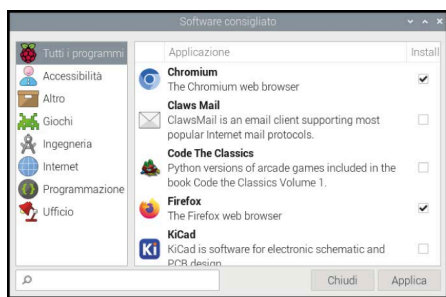


Figura 3-16 Lo strumento Recommended Software

Se accanto a un software è presente un segno di spunta, significa che è già installato sul Raspberry Pi. Altrimenti fai clic sulla casella di controllo accanto a esso per aggiungere un segno di spunta e contrassegnarlo per l'installazione. Puoi contrassegnare tutti i software che desideri prima di installarli in una volta sola, tuttavia se utilizzi una scheda microSD con meno memoria

rispetto a quanto consigliato, potrebbe non esserci abbastanza spazio per installare tutto.



APPLICAZIONI PREINSTALLATE

Alcune versioni di Raspberry Pi OS sono dotate di più software installati rispetto ad altre. Se lo strumento Recommended Software dice che Code the Classics è già installato (se la casella di controllo è già spuntata), puoi scegliere qualcos'altro dall'elenco da installare al suo posto.

Sono disponibili software per Raspberry Pi OS per eseguire un'ampia gamma di attività, tra cui una selezione di giochi scritti nel libro *Code the Classics, Volume 1*, che racconta la storia del gaming e ti insegna come creare i tuoi giochi in Python, disponibile su store.rpipress.cc.

Per installare i giochi di *Code the Classics*, fai clic sulla casella di controllo accanto a **Code the Classics** per selezionarla; potresti dover scorrere l'elenco delle applicazioni per individuarla. Il testo (**sarà installato**) verrà visualizzato a destra dell'applicazione selezionata, come mostrato in **Figura 3-17**.

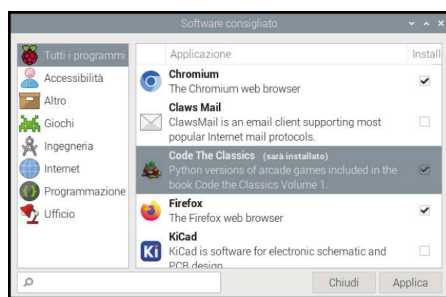


Figura 3-17 Selezione di Code the Classics per l'installazione

Fai clic su **Applica** per installare il software; ti verrà chiesto di inserire la tua password. L'installazione richiederà fino a un minuto, a seconda della velocità della connessione Internet (**Figura 3-18**). Al termine del processo, verrà visualizzato un messaggio che indica il completamento dell'installazione. Fai clic su **OK** per chiudere la finestra di dialogo, quindi fai clic sul pulsante **Chiudi** per chiudere lo strumento Recommended Software.

Se non vuoi più un software che avevi installato in precedenza, puoi liberare spazio disinstallandolo. Ti basterà caricare nuovamente lo strumento Recommended Software, individuare il software nell'elenco e fare clic sulla casella di controllo per rimuovere il segno di spunta. Facendo clic su **Applica**, il soft-

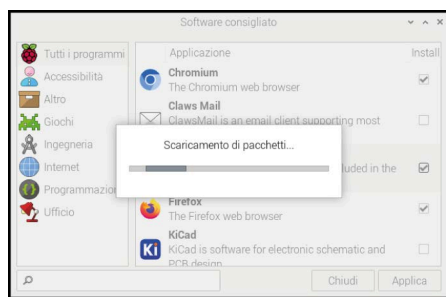


Figura 3-18 Installazione di Code the Classics

ware verrà rimosso, ma i file creati con esso e salvati nella cartella Documents rimarranno salvati.

Nella categoria Preferenze del menu Raspberry Pi potrai trovare lo strumento di aggiunta/rimozione software, uno strumento alternativo per installare o disinstallare software. Questo offre una selezione più ampia di software oltre all'elenco dei software consigliati. Impara a utilizzare lo strumento di aggiunta/rimozione software nell'Appendice B, *Come installare e disinstallare il software*.

Suite di produttività LibreOffice

Per avere un altro assaggio di ciò che Raspberry Pi può fare, fai clic sull'icona Raspberry Pi, sposta il puntatore del mouse su **Ufficio** e fai clic su **LibreOffice Writer**. Questa operazione consente di caricare la parte di elaborazione testi di LibreOffice (**Figura 3-19**), una suite di produttività open-source piuttosto popolare.

NON HAI LIBREOFFICE?

Se nel menu Raspberry Pi non è presente alcuna categoria Office o se non riesci a trovare LibreOffice Writer, potrebbe non essere installato. Accedi allo strumento Recommended Software e installa il software prima di procedere.



Con un elaboratore di testi potrai scrivere documenti e formattarli modificandone carattere, colore, dimensioni, aggiungendo effetti e persino inserendo immagini, grafici, tabelle e altri contenuti. Potrai anche controllare il tuo lavoro per verificare la presenza di errori: l'elaboratore di testi, infatti, ri-

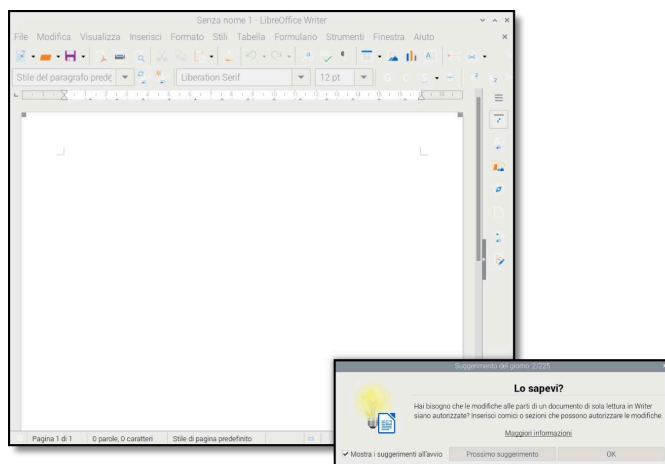


Figura 3-19 Programma LibreOffice Writer

leva gli errori ortografici e grammaticali rispettivamente in rosso e verde mentre digiti.

Per sperimentare la formattazione, potresti iniziare a scrivere un paragrafo su ciò che hai imparato finora sul Raspberry Pi e sul suo software. Esplora le diverse icone nella parte superiore della finestra per vedere cosa fanno: prova a ingrandire i caratteri e cambiarne il colore. Se non sei sicuro di come fare, sposta il puntatore del mouse su ogni icona, vedrai una descrizione che ti indica le sue funzioni. Quando hai finito di sperimentare, fai clic sul menu **File** e sull'opzione **Salva** per salvare ciò che hai fatto (**Figura 3-20**). Assegna un nome e fai clic sul pulsante **Salva**.



SALVA IL TUO LAVORO

Prendi l'abitudine di salvare il tuo lavoro, anche se non l'hai ancora finito, così avrai meno problemi qualora si verifici un'interruzione di corrente nel bel mezzo di un lavoro.

LibreOffice Writer è solo una parte della suite di produttività LibreOffice. Nella categoria del menu Ufficio, oltre a LibreOffice Writer troverai:

- ▶ **LibreOffice Base:** un database, ossia uno strumento per memorizzare le informazioni, cercarle velocemente e analizzarle.
- ▶ **LibreOffice Calc:** un foglio di calcolo, ovvero uno strumento per la gestione dei numeri e la creazione di grafici e diagrammi.

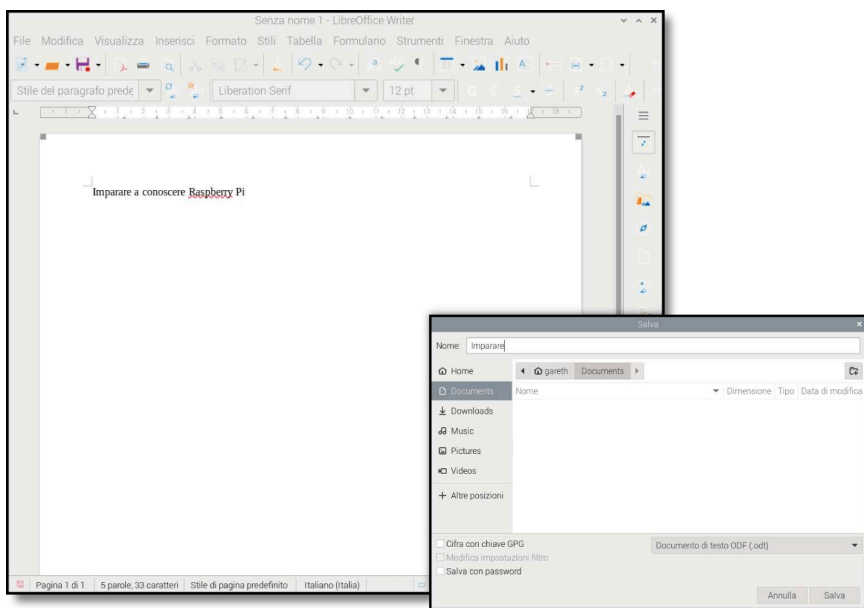


Figura 3-20 Salvataggio di un documento

- ▶ **LibreOffice Draw:** un programma di illustrazione, ossia uno strumento per la creazione di immagini e diagrammi.
- ▶ **LibreOffice Impress:** un programma di presentazione per la creazione di slide e presentazioni.
- ▶ **LibreOffice Math:** un editor di formule, ovvero uno strumento per creare formule matematiche formattate da utilizzare in altri documenti.

LibreOffice è disponibile anche per altri computer e sistemi operativi. Se ti piace usarlo su Raspberry Pi, puoi scaricarlo gratuitamente da libreoffice.org e installarlo su qualsiasi computer Microsoft Windows, Apple macOS o Linux. Altrimenti chiudi LibreOffice Writer, facendo clic sul pulsante di chiusura nell'angolo in alto a destra della finestra.

ISTRUZIONI PER DUBBI E PROBLEMI

La maggior parte dei programmi include un menu di guida con tutte le informazioni necessarie: dalle spiegazioni sul programma alle guide su come usarlo. Se durante l'uso di un programma ti senti disorientato o riscontri dei problemi, apri il menu Guida per riorientarti.



Strumento Configurazione di Raspberry Pi

L'ultimo programma che prenderemo in considerazione in questo capitolo è lo strumento Configurazione di Raspberry Pi. Questo strumento è simile alla procedura guidata Welcome Wizard utilizzata all'inizio, infatti ti consente di modificare alcune impostazioni in Raspberry Pi OS. Fai clic sull'icona di Raspberry Pi, sposta il puntatore del mouse per selezionare la categoria **Preferenze**, quindi fai clic su **Configurazione di Raspberry Pi** per aprire (Figura 3-21).



Figura 3-21 Strumento Configurazione di Raspberry Pi

Lo strumento è suddiviso in cinque schede. La prima è **Sistema**: ti consente di modificare la password del tuo account, impostare un nome host (ossia il nome usato da Raspberry Pi nella rete locale cablata o wireless) e modificare altre impostazioni, inclusa la scelta del browser Web predefinito. Tuttavia non dovrebbero essere necessarie modifiche per la maggior parte delle impostazioni in questa scheda. Fai clic sulla scheda **Schermo** per passare alla categoria successiva. Qui, se necessario, potrai modificare le impostazioni di visualizzazione dello schermo per adattarle al televisore o al monitor.



ALTRI DETTAGLI

Questa breve panoramica ti consente di abituarti allo strumento. Per maggiori informazioni sulle configurazioni, consulta l'Appendice E, *Strumento Configurazione di Raspberry Pi*.

Nella scheda **Interfacce** sono presenti diverse impostazioni, tutte disattivate (ad eccezione di **Console seriale** e **Serial Port**). Modificali solo se aggiungi un nuovo hardware e solamente se indicato dal produttore dell'hardware. Fanno eccezione: **SSH**, che attiva un "Secure Shell" e ti consente di collegarti al Raspberry Pi da un altro computer presente sulla tua rete tramite un client SSH; **VNC**, che abilita un "Virtual Network Computer" per consentirti di visualizzare e controllare il desktop di Raspberry Pi OS da un altro computer.

tramite un client VNC e **GPIO remoti**, che ti consente di utilizzare i pin del GPIO Raspberry Pi (che imparerai a conoscere nel Capitolo 6, *Physical computing con Scratch e Python* da un altro computer nella tua rete.

Fai clic sulla scheda **Prestazioni** per visualizzare la quarta categoria. Qui puoi configurare **Protezione file system**, che è un modo per utilizzare il Raspberry Pi senza scrivere modifiche sulla scheda microSD. Nella maggior parte dei casi non è necessario, quindi la maggior parte degli utenti può lasciare questa sezione invariata.

Infine, fai clic sulla scheda **Localizzazione** per visualizzare l'ultima categoria. Qui potrai modificare le impostazioni locali, che controllano, fra le varie impostazioni, la lingua usata in Raspberry Pi OS, la modalità di visualizzazione dei numeri, il fuso orario, il layout della tastiera e le impostazioni del Paese per il Wi-Fi. Per ora ti basta fare clic su **Annulla** per chiudere lo strumento senza effettuare modifiche.

ATTENZIONE

Le frequenze radio Wi-Fi consentite possono variare a seconda del Paese. Se nello strumento Configurazione di Raspberry Pi imposti un Paese diverso da quello impostato per il Wi-Fi, potresti riscontrare problemi nella connessione alle tue reti e potrebbe anche essere illegale in base alle leggi di licenza delle frequenze radio, quindi seleziona il tuo Paese.



Aggiornamenti software

Raspberry Pi OS riceve aggiornamenti di frequente, che aggiungono nuove funzionalità o correggono bug. Se il Raspberry Pi è collegato a una rete tramite cavo Ethernet o Wi-Fi, controllerà automaticamente la presenza di aggiornamenti e ti comunicherà se sono pronti per essere installati con una piccola icona nell'area di notifica (assomiglia a una freccia che punta verso il basso in una barra, circondata da un cerchio).

Se vedi questa icona in alto a destra sul desktop, significa che ci sono aggiornamenti pronti per essere installati. Fai clic sull'icona e su **Installa gli aggiornamenti** per scaricarli e installarli. Se preferisci vedere prima quali sono gli aggiornamenti, fai clic su **Mostra gli aggiornamenti** per visualizzare un elenco (Figura 3-22).

Il tempo necessario per installare gli aggiornamenti varia a seconda del numero degli aggiornamenti presenti e della velocità della connessione a Internet, ma dovrebbe richiedere solo pochi minuti. Dopo l'installazione degli aggiornamenti, l'icona scomparirà dall'area di notifica finché non ci saranno altri aggiornamenti da installare.

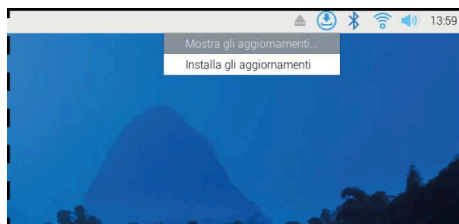


Figura 3-22 Utilizzo dello strumento di aggiornamento software

Alcuni aggiornamenti sono progettati per migliorare la sicurezza di Raspberry Pi OS. È importante utilizzare lo strumento di aggiornamento software per mantenere il sistema operativo aggiornato.

Spegnimento

Ora che hai scoperto alcune funzioni del desktop di Raspberry Pi OS è arrivato il momento di apprendere una funzione importante: come spegnere correttamente il tuo Raspberry Pi. Come in ogni computer, il Raspberry Pi mantiene i file su cui stai lavorando nella *memoria volatile*, ossia una memoria che viene svuotata una volta spento il sistema. Salva i documenti che stai creando nella scheda microSD, così non perderai nulla, perché verranno spostati dalla memoria volatile a quella *non volatile*.

I documenti su cui stai lavorando non sono gli unici file aperti. Il sistema operativo Raspberry Pi OS stesso mantiene una serie di file aperti mentre è in esecuzione: staccando il cavo di alimentazione dal tuo Raspberry Pi mentre questi sono ancora aperti potrebbe causarne la corruzione e la necessità di reinstallare il sistema operativo.

Per evitare che questo accada, assicurati che Raspberry Pi OS salvi tutti i suoi file e sia pronto per essere spento, un processo noto come *spegnimento* del sistema operativo.

Fai clic sull'icona di Raspberry Pi nell'angolo in alto a sinistra del desktop, quindi su **Arresta**. Visualizzerai una finestra con tre opzioni (**Figura 3-23**): **Arresta**, **Riavvia** e **Termina sessione**. **Arresta** sarà l'opzione che userai di più: facendovi clic sopra Raspberry Pi OS chiuderà tutti i software e file aperti, quindi spegnerà il sistema. Quando il display diventa nero, attendi qualche secondo lo spegnimento della luce verde lampeggiante su Raspberry Pi, successivamente potrai scollegare l'alimentazione.

Se premi il pulsante una volta, visualizzerai la stessa finestra che viene visualizzata facendo clic sull'icona di Raspberry Pi e su **Arresta**; premendo nuovamente il pulsante di accensione quando la finestra è visibile, il Raspberry Pi si spegnerà in modo sicuro.

Se tieni premuto il pulsante di accensione più a lungo, verrà eseguito uno *spegnimento forzato*. Esegui questa operazione solo se il Raspberry Pi non risponde alle istruzioni e non riesci a spegnerlo in nessun altro modo, poiché vi è il rischio di corrompere i file o il sistema operativo.

Per riaccendere il Raspberry Pi, scollega e ricollega il cavo di alimentazione o attiva la corrente della presa a muro.



Figura 3-23
Spegnimento di Raspberry Pi

Il riavvio è simile allo **Arresta**, infatti chiude tutto, ma invece di spegnere il Raspberry Pi lo riavvia, è quasi come se avessi scelto **Arresta**, scollegato e ricollegato il cavo di alimentazione. Dovrai selezionare **Riavvia** quando apporti alcune modifiche che richiedono un riavvio del sistema operativo (come l'installazione di alcuni aggiornamenti del software principale) o se si è verificato qualche problema con un software (*crash*) che ha reso inutilizzabile Raspberry Pi OS.

Termina sessione è utile solo se hai più di un account utente sul tuo Raspberry Pi: chiude tutti i programmi aperti e ti porta a una schermata di accesso in cui ti vengono richiesti nome utente e password. Se hai premuto accidentalmente Termina sessione e vuoi riaccedere, digita il nome utente e la password scelta nella procedura guidata Welcome Wizard illustrata all'inizio del capitolo.

ATTENZIONE

Non rimuovere mai il cavo di alimentazione da un Raspberry Pi o scollegare l'alimentatore a parete senza prima aver spento il dispositivo. Il sistema operativo potrebbe corrompersi e potresti perdere i file creati o scaricati.





Capitolo 4

Programmare con Scratch 3

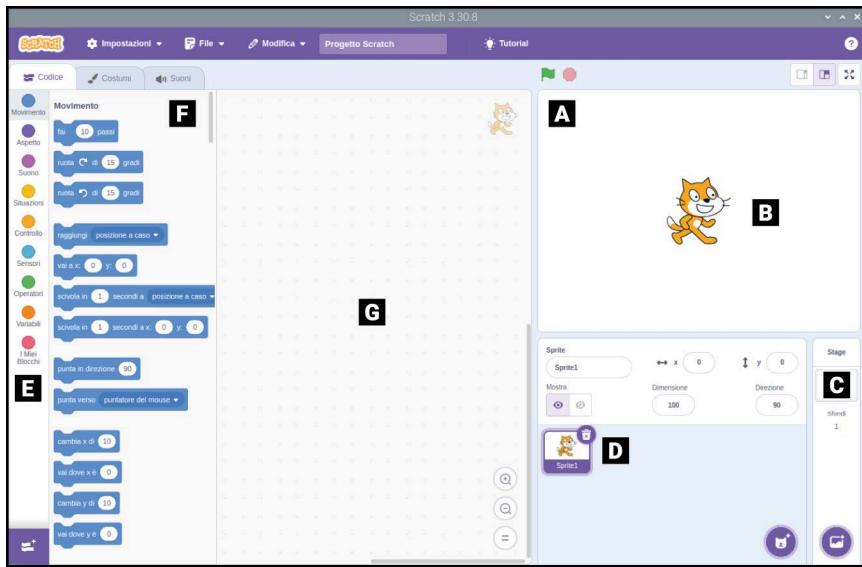
Scopri come iniziare a programmare con Scratch, il linguaggio di programmazione a blocchi

Con Raspberry Pi non solo potrai utilizzare i software creati da altre persone, ma potrai realizzare il tuo software seguendo la tua immaginazione. Raspberry Pi è la piattaforma perfetta per creare e sperimentare, sia che tu abbia già esperienza nella programmazione sia che ti avvicini per la prima volta a questa disciplina.

Scratch, un linguaggio di programmazione visuale sviluppato dal Massachusetts Institute of Technology (MIT), è ciò che rende la programmazione su Raspberry Pi così intuitiva. Mentre con i linguaggi di programmazione tradizionali è necessario scrivere istruzioni testuali che dovranno essere eseguite dal computer, all'incirca come scrivere la ricetta da seguire per preparare una torta, con Scratch potrai costruire il tuo programma passo dopo passo mediante blocchi: parti di codice già scritte mascherate da pezzi di puzzle raggruppati per colore.

Scratch è un ottimo primo linguaggio per aspiranti programmatori di qualunque età. Non farti ingannare dalla grafica: è un ambiente di programmazione potente e pienamente funzionale che ti consentirà di creare qualsiasi cosa, da semplici giochi e animazioni fino a complessi progetti di robotica interattiva.

L'interfaccia Scratch 3



- A** Area Stage
- B** Sprite
- C** Controlli Stage
- D** Elenco degli sprite
- E** Palette dei blocchi
- F** Blocchi
- G** Area codice

Come attori in uno spettacolo, i tuoi personaggi si muovono sul palcoscenico (stage, **A**) sotto il controllo del programma Scratch. Questi personaggi sono chiamati sprite (**B**). Per modificare lo stage, ad esempio per aggiungere uno sfondo personalizzato, utilizza i controlli dello stage (**C**). Tutti gli sprite creati o caricati sono presenti nell'elenco degli sprite (**D**).

Tutti i blocchi disponibili vengono visualizzati nella palette dei blocchi (**E**) suddivisi in categorie contraddistinte per colore. I blocchi (**F**) sono parti di codice già scritte. Potrai costruire il programma nell'area codice (**G**) trascinando e rilasciando i blocchi dalla relativa palette per formare gli script.

VERSIONI DI SCRATCH

Sono disponibili due versioni di Scratch per Raspberry Pi OS: Scratch e Scratch 3. Per questa guida abbiamo preso in considerazione Scratch 3, che è compatibile solo con Raspberry Pi 4, Raspberry Pi 5 e Raspberry Pi 400.



INSTALLAZIONE DI SCRATCH

Se non trovi Scratch 3 nel menu **Programmazione**, è possibile che non sia installato nella versione di Raspberry Pi OS che hai a disposizione. Vai a «Lo strumento Recommended Software» a pagina 43 e segui le istruzioni per installare Scratch 3 dalla categoria **Programmazione**, quindi, una volta installato, torna a questa sezione.




Il tuo primo programma Scratch: Ciao mondo!

Scratch 3 si apre come qualsiasi altro programma su Raspberry Pi: fai clic sull'icona a Raspberry Pi per aprire il menu, sposta il cursore sulla sezione **Programmazione**, quindi fai clic su Scratch 3. L'interfaccia utente di Scratch 3 si aprirà dopo pochi secondi. Potresti visualizzare un messaggio sulla raccolta dei dati: fai clic su **Condividi le mie statistiche di uso con lo Scratch Team** se vuoi inviare i dati di utilizzo al team di Scratch, altrimenti fai clic su **Non condividere le mie statistiche di uso con lo Scratch Team**. Scratch terminerà il caricamento una volta selezionata la scelta.

Scratch è diverso dalla maggior parte dei programmi di programmazione, infatti non dovrai dire al computer cosa fare tramite istruzioni scritte. Inizia facendo clic sulla categoria **Aspetto** nella palette dei blocchi presente nella parte sinistra della finestra Scratch. Vedrai i blocchi viola della categoria. Individua il blocco **dire Ciao!**, fai clic su di esso e tieni premuto il pulsante sinistro del mouse, quindi trascinalo nell'area codice al centro della finestra Scratch prima di rilasciarlo (Figura 4-1).

Osserva la forma del blocco che hai appena trascinato: presenta un incavo nella parte superiore e una sporgenza in quella inferiore. Questo significa che sopra e sotto al blocco che hai trascinato andranno collocati altri blocchi, proprio come in un puzzle. In questo programma, i blocchi da collocare sopra rientrano nella categoria dei *trigger*.

Fai clic sulla categoria **Situazioni** della palette dei blocchi, color oro, quindi fai clic e trascina il blocco **quando si clicca su** , o blocco *cappello*, nell'area codice. Posizionalo in modo che la parte sporgente combaci con l'incavo nella parte superiore del blocco **dire Ciao!** fino a quando non vedi un contorno bianco, poi rilascia il pulsante del mouse. È sufficiente che i blocchi siano ab-

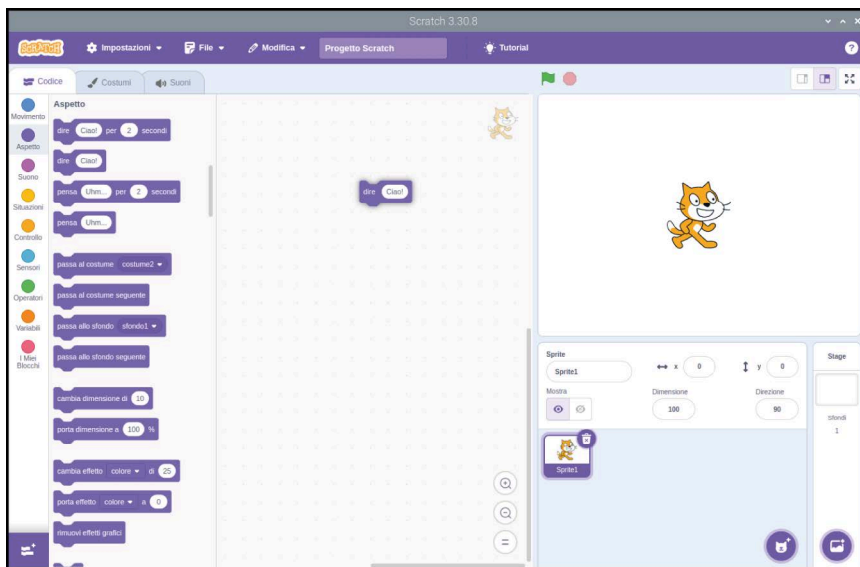
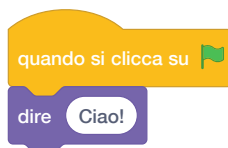


Figura 4-1 Trascinamento del blocco nell'area codice

bastanza vicini e si incastreranno perfettamente. Qualora non si incastrino, fai clic e tieni premuto per modificare la posizione finché non si incastrano.



Il tuo programma è completo. Per farlo funzionare o *eseguirlo*, fai clic sulla bandiera verde in alto a sinistra nell'area stage. Se hai programmato correttamente, lo sprite gatto nell'area stage ti saluterà dicendoti "**Ciao!**" (Figura 4-2). Complimenti per la tua prima animazione!

Prima di andare avanti, assegna un nome al progetto e salvalo. Fai clic sul menu **File**, poi su **Salva sul tuo computer**. Digita un nome e fai clic sul pulsante **Salva** (Figura 4-3).



COS'ALTRO PUÒ DIRE?

Alcuni blocchi in Scratch possono essere modificati. Prova a fare clic sulla parola "**Ciao!**", poi scrivi qualcos'altro e fai di nuovo clic sulla bandiera verde. Cosa succede sullo stage?

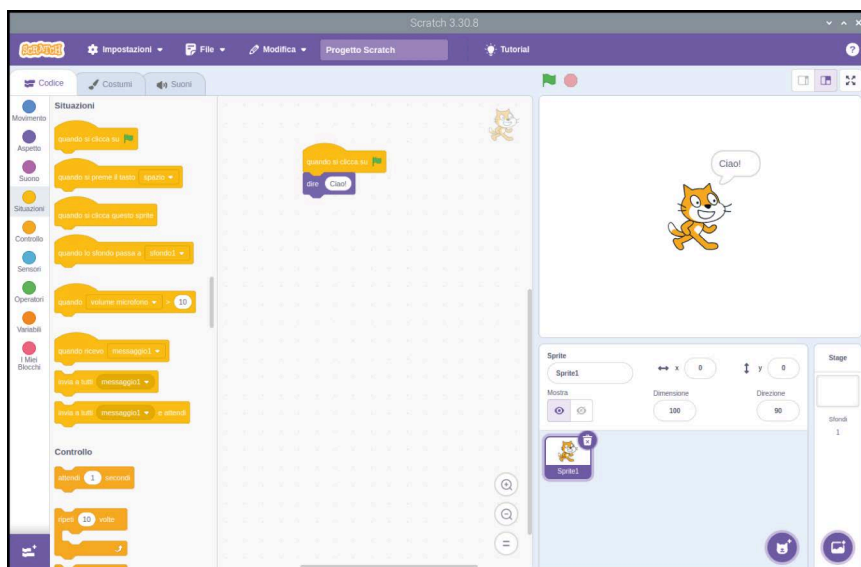


Figura 4-2 Fai clic sulla bandiera verde sullo stage e il gatto ti dirà "Ciao!"

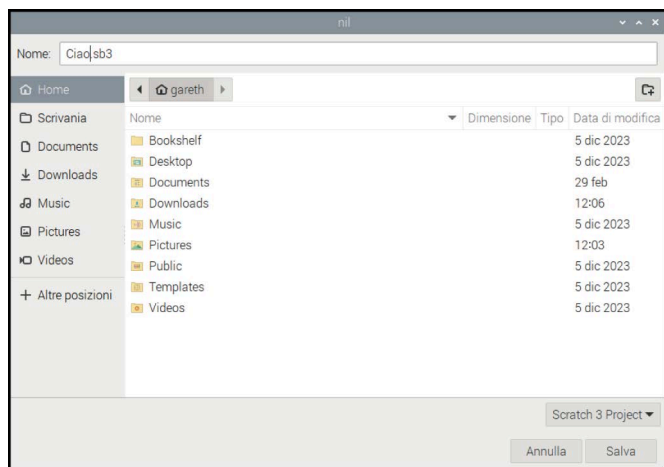


Figura 4-3 Salva il tuo progetto con un nome inconfondibile

Passaggi successivi: creazione di sequenze

Il tuo progetto ha due blocchi, ma comprende una sola istruzione, ossia dire "Ciao!" ogni volta che il programma viene eseguito. Per fare di più, bisogna approfondire la *creazione di sequenze*. I programmi per computer, nella loro semplicità, sono un elenco di istruzioni, proprio come una ricetta. Ogni istruzione segue la precedente in una progressione logica nota come *sequenza lineare*.

Inizia facendo clic e trascinando il blocco **dire Ciao!** dall'area codice alla palette dei blocchi (**Figura 4-4**). In questo modo il blocco verrà rimosso dal programma e rimarrà solo il blocco trigger: **quando si clicca su**.

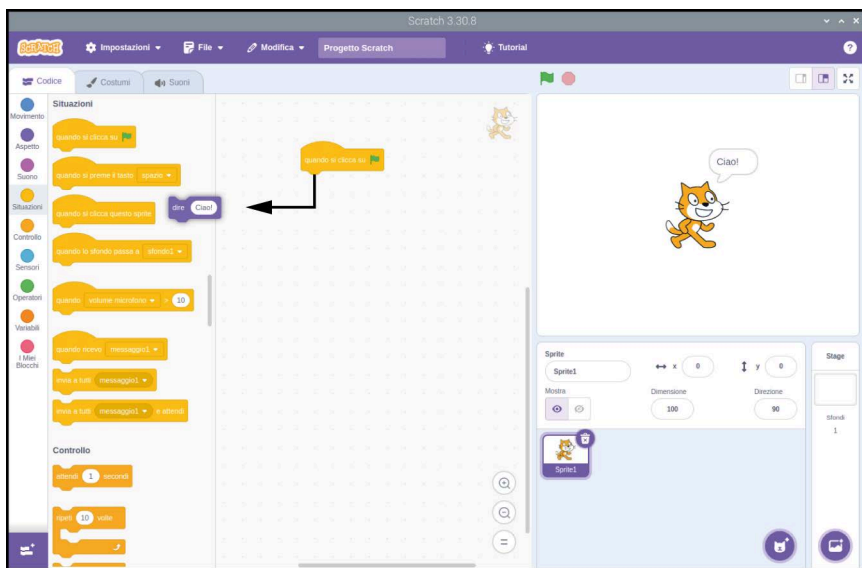
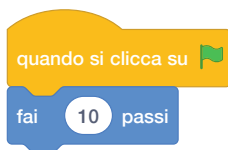


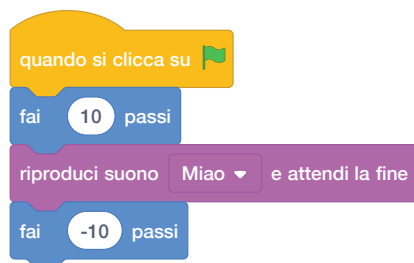
Figura 4-4 Per eliminare un blocco trascinalo fuori dall'area codice

Fai clic sulla categoria **Movimento** nella palette dei blocchi, quindi fai clic e trascina il blocco **fai 10 passi** in modo che si incastri sotto il blocco trigger nell'area codice.

Come suggerisce il nome, fa muovere lo sprite (il gatto) di un certo numero di passi nella direzione verso cui è rivolto.

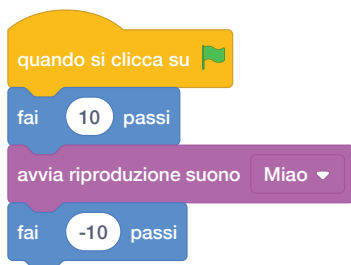


Successivamente aggiungi altre istruzioni al programma per creare una *sequenza*: fai clic sulla categoria **Suono**, di colore rosa, quindi fai clic e trascina il blocco **riproduci suono Miao e attendi la fine** in modo che si incastri sotto il blocco **fai 10 passi**. Ora continua la sequenza: fai di nuovo clic sulla categoria **Movimento** e trascina un altro blocco **fai 10 passi** sotto il blocco **Suono**, ma questa volta cambia **10** con **-10** per creare un blocco **fai -10 passi**.



Fai clic sulla bandiera verde sopra allo stage per eseguire il programma. Vedrai il gatto muoversi verso destra, fare un miagolio (per sentirlo assicurati di aver collegato gli altoparlanti o le cuffie) e tornare all'inizio. Ogni volta che fai clic sulla bandiera verde, il gatto ripeterà queste azioni.

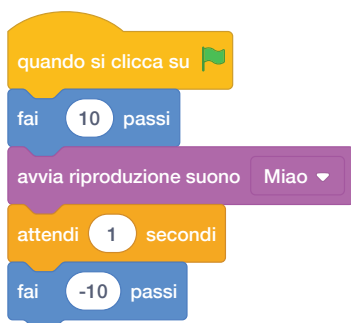
Complimenti! Hai creato una sequenza di istruzioni che viene eseguita da Scratch dall'inizio alla fine. Scratch eseguirà un'istruzione della sequenza alla volta, tuttavia lo farà molto velocemente: prova a cancellare il blocco **riproduci suono Miao e attendi la fine** trascinando il blocco successivo **fai -10 passi** per staccarlo, trascina quindi il blocco **riproduci suono Miao e attendi la fine** nella palette dei blocchi poi sostituiscilo con **avvia riproduzione suono Miao** prima di trascinare il blocco **fai -10 passi** in fondo al programma.



Fai clic sulla bandiera verde per eseguire di nuovo il programma. Questa volta lo sprite gatto non sembra muoversi. In realtà si sta muovendo, ma torna alla posizione di partenza talmente velocemente che sembra fermo. Questo perché con il blocco **avvia riproduzione suono Miao** il programma non attende che l'audio finisca di essere riprodotto prima del passo successivo. Poiché Raspberry Pi

"pensa" così velocemente, l'istruzione successiva viene eseguita prima che tu possa vedere lo sprite gatto muoversi.

C'è un altro modo per risolvere questo problema, oltre all'utilizzo del blocco **riproduci suono Miao e attendi la fine**: fai clic sulla categoria arancione chiaro **Controllo** della palette dei blocchi, quindi fai clic su un blocco **attendi 1 secondi** e trascinalo tra il blocco **avvia riproduzione suono Miao** e il blocco inferiore **fai -10 passi**.



Fai clic sulla bandiera verde per eseguire il programma un'ultima volta e vedrai che il gatto attenderà un secondo dopo essersi spostato a destra e prima di tornare. Questo viene chiamato *ritardo* ed è importante per controllare la durata della sequenza di istruzioni.



SFIDA: AGGIUNGI ALTRI PASSAGGI

Prova ad aggiungere altri passaggi nella sequenza e a modificare i valori in quelli già presenti. Cosa succede quando il numero di passaggi in un blocco **fai passi** non corrisponde a quello di un altro? Cosa succede se si prova a riprodurre un suono mentre un altro suono è ancora in esecuzione?

Ripetere la sequenza

La sequenza che hai creato finora viene eseguita solo una volta. Quando fai clic sulla bandiera verde, lo sprite gatto si muove e miagola e il programma si ferma finché non fai di nuovo clic sulla bandiera verde. Tuttavia non deve necessariamente fermarsi, infatti Scratch presenta anche un blocco **Controllo** *ripeti*.

Fai clic sulla categoria **Controllo** nella palette blocchi e cerca il blocco **per sempre**. Fai clic e trascina questo blocco nell'area codice, quindi rilascialo sotto al blocco **quando si clicca su** e sopra al primo blocco **fai 10 passi**.



Il blocco **per sempre** è a forma di C e si ingrandisce automaticamente per racchiudere gli altri blocchi della sequenza. Fai clic sulla bandiera verde e vedrai subito che cosa fa il blocco **per sempre**: il programma, anziché essere eseguito una sola volta, si ripeterà all'infinito. In programmazione, questo è noto come *loop infinito*, ossia un ciclo che non finisce mai.

Per fermare il suono del miagolio fai clic sull'ottagono rosso accanto alla bandiera verde sopra l'area stage, in questo modo il programma si arresterà. Per cambiare il tipo di loop, trascina il primo blocco **fai 10 passi** e i blocchi sottostanti fuori dal blocco **per sempre**, quindi rilasciali sotto il blocco **quando si clicca su**. Fai clic e trascina il blocco **per sempre** nella palette di blocchi per eliminarlo, quindi fai clic e trascina il blocco **ripeti 10 volte** sotto al blocco **quando si clicca su** in modo che circonda gli altri blocchi.



Fai clic sulla bandiera verde per eseguire il nuovo programma. All'inizio sembrerà uguale alla versione originale, che ripeteva in continuazione la sequenza di istruzioni. Questa volta invece, anziché continuare per sempre, il loop cesserà dopo dieci ripetizioni. Viene definito *loop definito*: sarai tu a decidere

quando finisce. I loop sono uno strumento essenziale e la maggior parte dei programmi, specialmente giochi e programmi di rilevamento, usa spesso loop sia infiniti sia definiti.



COSA SUCCEDDE ORA?

Cosa succede se cambi il numero nel blocco loop con uno superiore? Cosa succede se è inferiore? Cosa succede se metti il numero 0 nel blocco loop?

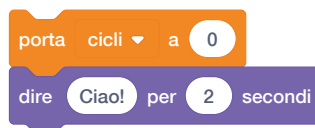
Variabili e condizionali

Gli ultimi concetti da comprendere prima di iniziare a programmare con Scratch sono strettamente legati tra loro: *variabili* e *condizionali*. Una variabile è, come suggerisce il nome, un valore che può variare o cambiare nel tempo e sotto il controllo del programma. Le variabili hanno due proprietà principali: il nome e il valore che memorizzano. Questo valore non deve essere necessariamente un numero. Può esser costituito da numeri, testo, vero o falso (i cosiddetti *valori booleani*) oppure può essere completamente vuoto, il cosiddetto *valore nullo*.

Le variabili sono strumenti essenziali. Pensa alle cose che devi tenere sotto controllo in un gioco: la salute di un personaggio, la velocità di un oggetto in movimento, il livello a cui stai giocando e il punteggio. Tutti questi aspetti vengono monitorati come variabili.

Per prima cosa, fai clic sul menu **File** e salva il programma esistente facendo clic su **Salva sul tuo computer**. Se hai già salvato il programma ti verrà chiesto se vuoi sovrascriverlo, sostituendo così la vecchia copia salvata con la nuova versione aggiornata. Successivamente fai clic su **File**, quindi su **Nuovo** per aprire un nuovo progetto vuoto (fai clic su **OK** quando viene chiesto se vuoi sostituire il contenuto del progetto corrente). Fai clic sulla categoria **Variabili** di colore arancione scuro nella palette dei blocchi, quindi sul pulsante **Crea una Variabile**. Digita **cicli** come nome della variabile (**Figura 4-5**), poi fai clic su **OK** per visualizzare una serie di blocchi nella palette dei blocchi.

Fai clic sul blocco **porta cicli a 0** e trascinalo nell'area codice. Indicherà al programma di *inizializzare* la variabile con valore 0. Successivamente, fai clic sulla categoria **Aspetto** della palette dei blocchi e trascina il blocco **dire Ciao! per 2 secondi** sotto il blocco **porta cicli a 0**.



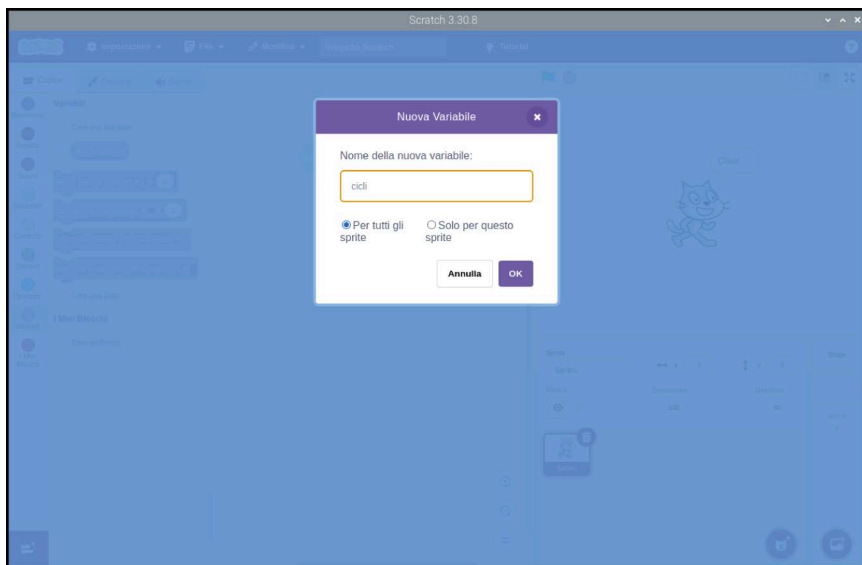


Figura 4-5 Assegna un nome alla nuova variabile

Come abbiamo visto prima, il blocco **dire Ciao!** fa dire allo sprite gatto qualsiasi cosa vi sia scritto dentro. Anziché scrivere il messaggio nel blocco puoi utilizzare una variabile. Fai nuovamente clic sulla categoria **Variabili** nella palette dei blocchi, quindi fai clic e trascina il blocco arrotondato **cicli** (detto anche *blocco reporter*, posizionato all'inizio della lista e con una casella di spunta accanto) sopra la parola **Ciao!** nel blocco **dire Ciao! per 2 secondi**. In questo modo si crea un nuovo blocco combinato: **dire cicli per 2 secondi**.



Fai clic sulla categoria **Situazioni** della palette dei blocchi, fai clic e trascina il blocco **quando si clicca su**, quindi posizionalo all'inizio della sequenza di blocchi. Fai clic sulla bandiera verde sopra l'area stage e lo sprite gatto dirà "0" (**Figura 4-6**), ossia il valore che hai assegnato alla variabile **cicli**.

Le variabili non sono immutabili. Fai clic sulla categoria **Variabili** della palette dei blocchi, fai clic e trascina il blocco **cambia cicli di 1** in fondo alla sequenza.

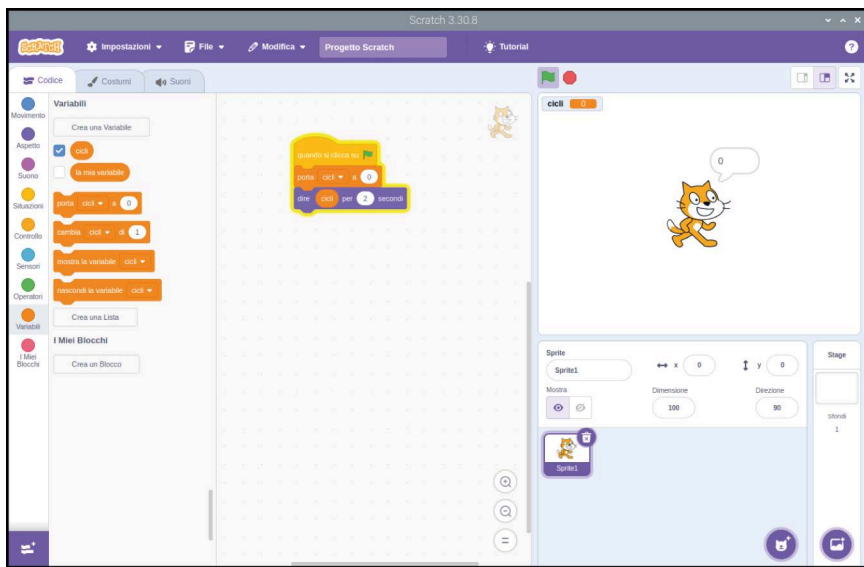


Figura 4-6 Questa volta il gatto dirà il valore della variabile

Successivamente fai clic sulla categoria **Controllo**, quindi fai clic e trascina un blocco **ripeti 10 volte** in modo che inizi direttamente sotto al blocco **porta cicli a 0** e sia attorno ai blocchi rimanenti nella tua sequenza.



Fai di nuovo clic sulla bandiera verde. Questa volta vedrai il gatto contare da 0 a 9. Questo succede perché il programma sta cambiando o *modificando* la variabile stessa: ogni volta che il loop viene eseguito, il programma aggiunge uno al valore della variabile **cicli** (Figura 4-7).

CONTARE DA ZERO

Anche se il loop che hai creato si ripete dieci volte, il gatto sprite conta solo fino a nove. Questo perché partiamo da zero per la nostra variabile. Includendo lo zero ci sono dieci numeri tra lo zero e il nove, così il programma si ferma prima che il gatto dica "10". Per modificarlo puoi impostare il valore iniziale della variabile a 1 anziché 0.

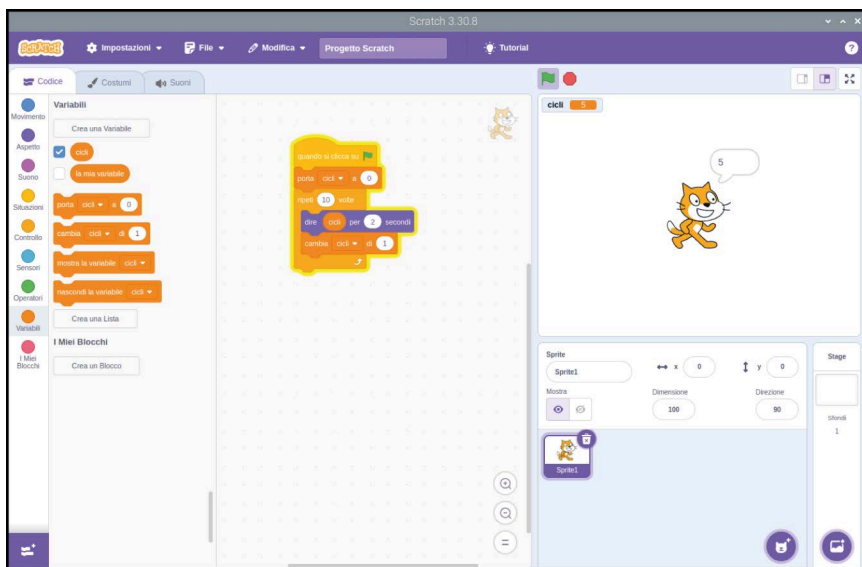
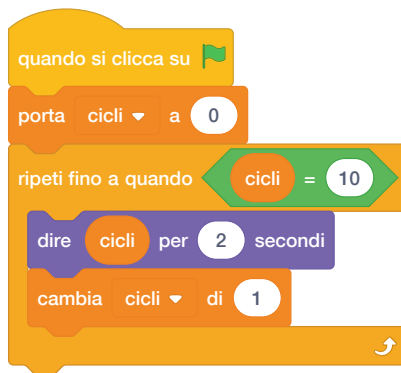


Figura 4-7 Grazie al loop il gatto conta

Puoi fare tantissime cose con le variabili. Fai clic sul blocco **dire cicli per 2 secondi** e trascinalo per staccarlo dal blocco **ripeti 10 volte** e rilasciarlo sotto il blocco **ripeti 10 volte**. Fai clic sul blocco **ripeti 10 volte** e trascinalo nella palette dei blocchi per eliminarlo, quindi sostituisilo con un blocco **ripeti fino a quando**, assicurandoti che il blocco sia collegato alla parte inferiore del blocco **porta cicli a 0**. Dovrebbe circondare entrambi gli altri blocchi della sequenza. Fai clic sulla categoria **Operatori**, di colore verde, nella palette dei blocchi, quindi fai clic e trascina il blocco a forma di diamante **=** e rilascialo sullo spazio a forma di diamante presente nel blocco **ripeti fino a quando**.

Il blocco **Operatori** ti consente di confrontare due valori, comprese le variabili. Fai clic sulla categoria **Variabili**, trascina il blocco reporter **cicli** nello spazio vuoto del blocco **Operatori** **=**, poi fai clic sullo spazio con **50** e digita il numero **10**.



Fai clic sulla bandiera verde sopra l'area stage e vedrai che il programma funzionerà come prima: lo sprite gatto conta da 0 a 9 (**Figura 4-8**) e il programma si ferma. Questo perché il blocco **ripeti fino a quando** funziona esattamente come il blocco **ripeti 10 volte**, ma invece di contare il numero di loop confronta il valore della variabile **cicli** con quello digitato a destra del blocco. Quando la variabile **cicli** raggiunge il valore 10 il programma si ferma.

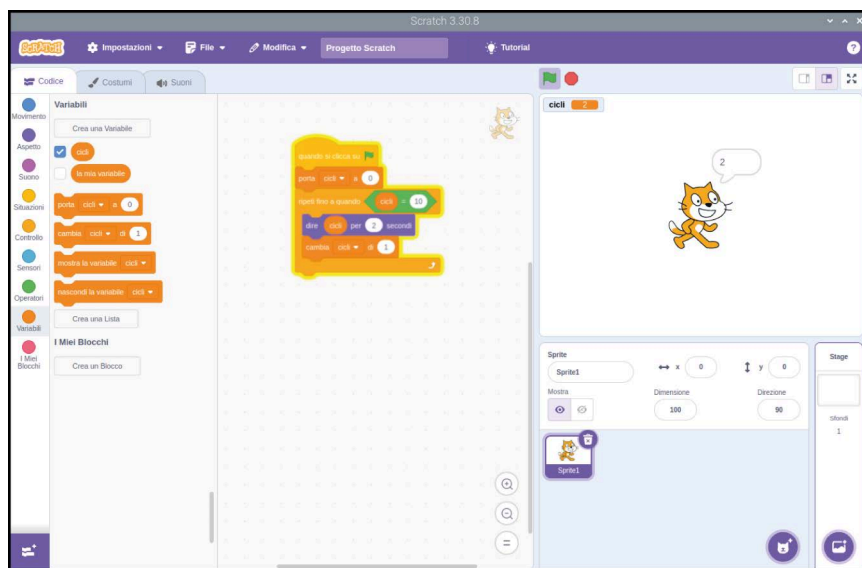


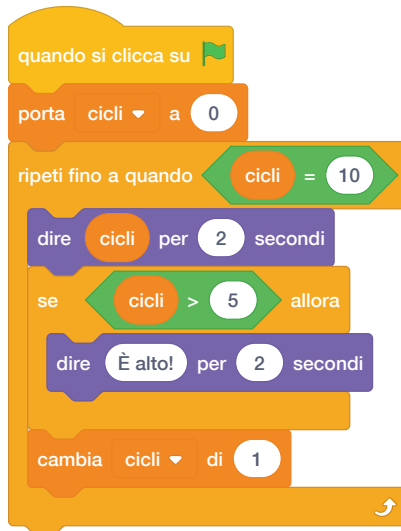
Figura 4-8 Utilizzo di un blocco repeat until con un operatore comparativo

Questo si chiama *operatore comparativo* perché confronta due valori. Fai clic sulla categoria **Operatori** della palette dei blocchi e cerca gli altri due blocchi a forma di diamante sopra e sotto quelli con il simbolo $=$. Sono operatori comparativi anche $<$, che confronta due valori e si attiva quando il valore di sinistra

è inferiore a quello di destra, e **>**, che si attiva quando il valore di sinistra è superiore a quello di destra.

Fai clic sulla categoria **Controllo** della palette dei blocchi, trova il blocco **se allora**, quindi fai clic e trascinalo nell'area del codice prima di collocarlo direttamente sotto al blocco **dire cicli per 2 secondi**. Circonderà automaticamente il blocco **cambia cicli di 1**, quindi fai clic e trascinalo per spostarlo in modo che si colleghi alla parte inferiore del blocco **se allora**. Fai clic sulla categoria **Aspetto** della palette dei blocchi, quindi fai clic e trascina un blocco **dire Ciao! per 2 secondi** e rilascialo all'interno del blocco **se allora**. Fai clic sulla categoria **Operatori** della palette dei blocchi, quindi fai clic e trascina il blocco **>** nello spazio a forma di diamante nel blocco **se allora**.

Il blocco **se allora** è un blocco **condizionale**, il che significa che i blocchi al suo interno funzioneranno solo se viene soddisfatta una certa condizione. Fai clic sulla categoria **Variabili** nella palette dei blocchi, trascina il blocco reporter **cicli** nello spazio vuoto del blocco **>**, poi fai clic sullo spazio con **50** e digita il numero **5**. Infine, fai clic sulla parola **Ciao!** nel blocco **dire Ciao! per 2 secondi** e digita **È alto!**.



Fai clic sulla bandiera verde per eseguire il programma. In un primo momento funzionerà come prima, con lo sprite gatto che conta da 0 in poi. Quando raggiungerà sei, il primo numero maggiore di cinque, il blocco **se allora** verrà attivato e il gatto commenterà quanto sono alti i numeri (**Figura 4-9**). Complimenti, ora sei in grado di lavorare con variabili e condizionali!

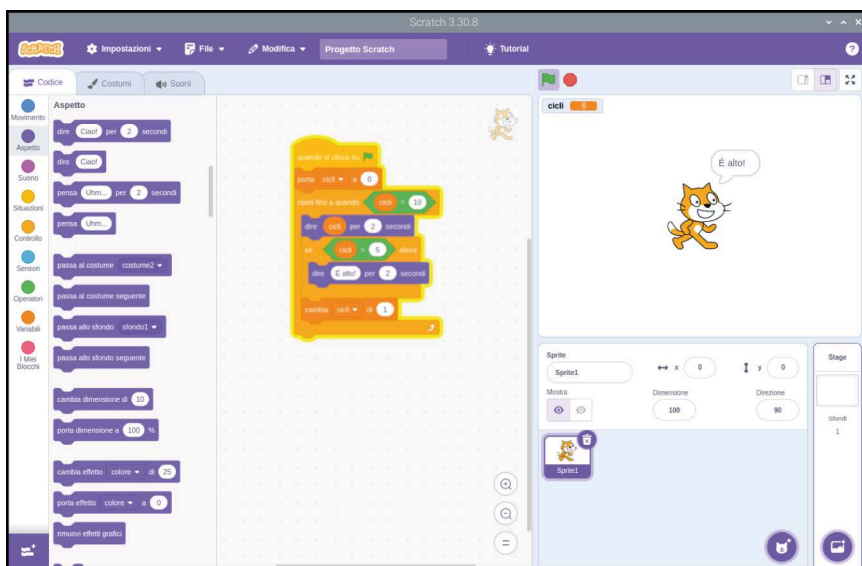


Figura 4-9 Il gatto fa un commento quando il numero raggiunge il sei



SFIDA: ALTO E BASSO

Come si può cambiare il programma in modo che lo sprite gatto commenti quanto sono bassi i numeri sotto il 5? Puoi cambiarlo in modo che il gatto commenti sia i numeri alti che quelli bassi? Sperimenta con il blocco **se allora altrimenti** per rendere tutto più semplice!

Progetto 1: misuratore della velocità di reazione con astronauta

Ora che hai appreso come funziona Scratch, è il momento di creare qualcosa di più interattivo: un misuratore della velocità di reazione per onorare l'astronauta britannico dell'ESA Tim Peake e il suo tempo a bordo della Stazione Spaziale Internazionale.

Salva il programma appena creato, se desideri tenerlo, quindi apri un nuovo progetto facendo clic su **File** e **Nuovo**. Prima di iniziare, assegnagli un nome facendo clic su **File** e **Salva sul tuo computer** e chiamalo "Misuratore della velocità di reazione con astronauta".

Questo progetto si basa su due immagini non incluse nelle risorse integrate di Scratch: lo sfondo dello stage e lo sprite. Per scaricarle, fai clic sull'icona di Raspberry Pi per caricare il menu Raspberry Pi, sposta il puntatore del mouse su **Internet** e fai clic su **Browser web Chromium**. Una volta aperto il browser, digita **rptl.io/astro-bg** nella barra degli indirizzi, quindi premi il tasto **INVIO**. Fai clic con il tasto destro del mouse sull'immagine dello spazio, fai clic su **Salva immagine con nome**, quindi fai clic sul pulsante **Salva** (Figura 4-10). Fai nuovamente clic sulla barra degli indirizzi e digita **rptl.io/astro-sprite** e premi il tasto **INVIO** per scaricare il file audio.

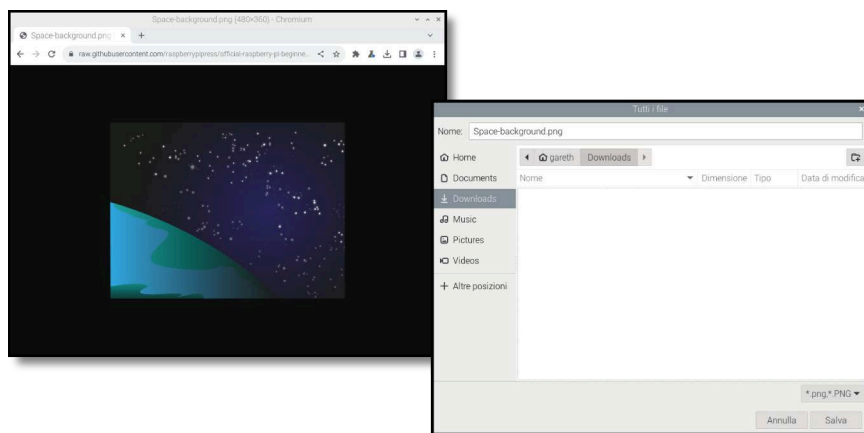




Figura 4-10 Salvataggio dell'immagine di sfondo

Fai clic con il tasto destro del mouse sull'immagine di Tim Peake, fai clic su **Salva immagine con nome** quindi seleziona la cartella **Downloads** e fai clic sul pulsante **Salva**. Una volta salvate le immagini, potrai chiudere Chromium o lasciarlo aperto e utilizzare la barra delle applicazioni per tornare a Scratch 3.

INTERFACCIA UTENTE

Se hai seguito questo capitolo fin dall'inizio, dovresti avere familiarità con l'interfaccia utente di Scratch 3. Le seguenti istruzioni presuppongono che tu conosca già la struttura del programma, se dimentichi dove trovare qualcosa, guarda di nuovo l'immagine dell'interfaccia utente all'inizio del capitolo.



Fai clic con il tasto destro del mouse sullo sprite gatto nell'elenco e fai clic su **cancella**. Passa il puntatore del mouse sull'icona **Scegli uno Sfondo** . Successivamente fai clic sull'icona **Importa Sfondo**  dall'elenco visualizzato.

Cerca il file **Space-background.png** nella cartella **Downloads**, fai clic per selezionarlo, quindi fai clic su **OK**. Lo stage avrà quindi come sfondo l'immagine

dello spazio e l'area codice sarà sostituita dall'area dello sfondo (**Figura 4-11**). Qui puoi disegnare sullo sfondo, ma per ora fai clic sulla scheda **Codice** nella parte superiore della finestra di Scratch 3.

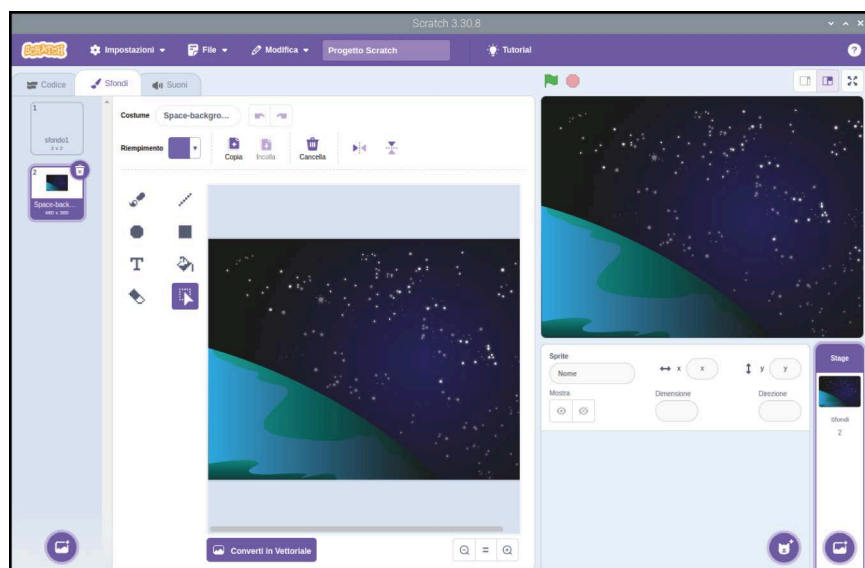






Figura 4-11 Sfondo nell'area stage

Carica il tuo nuovo sprite passando il puntatore del mouse sull'icona **Scegli uno Sprite** . Successivamente fai clic sull'icona **Importa Sprite**  nella parte superiore dell'elenco visualizzato. Cerca il file **Astronaut-Tim.png** nella cartella Downloads, fai clic per selezionare, quindi fai clic su **OK**. Lo sprite verrà aperto immediatamente nello stage, ma potrebbe non essere al centro: fai clic e trascinalo con il mouse e rilascialo in modo che sia vicino al centro inferiore (**Figura 4-12**).

Una volta aperti sfondo e sprite puoi iniziare a creare il tuo programma, quindi fai clic sulla scheda **Codice**. Inizia creando una nuova variabile chiamata **tempo**, accertandoti di aver selezionato **Per tutti gli sprite** prima di fare clic su **OK**. Fai clic sul tuo sprite (sullo stage o nel pannello sprite) per selezionarlo, quindi aggiungi un blocco **quando si clicca su**  dalla categoria **Situazioni** nell'area del codice. Quindi aggiungi un blocco **dire Ciao! per 2 secondi**  dalla categoria **Aspetto** e fai clic su di esso per modificare in **Ciao! Qui l'astronauta dell'ESA britannico Tim Peake. Sei pronto?**

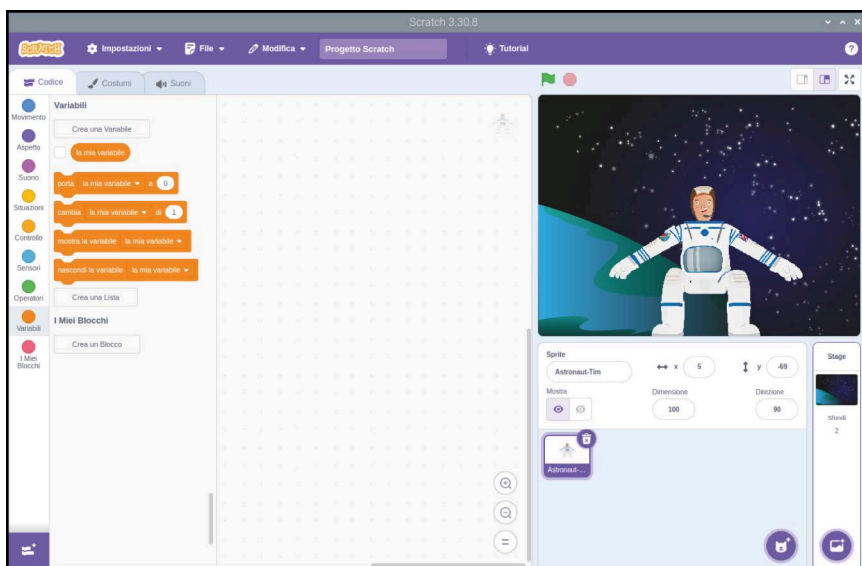


Figura 4-12 Trascinamento dello sprite astronauta nella parte inferiore centrale dello stage

quando si clicca su

dire Ciao! Qui l'astronauta dell'ESA britannico Tim Peake. Sei pronto? per 2 secondi

Aggiungi un blocco **attendi 1 secondi** dalla categoria **Controllo**, poi un blocco **dire Ciao!**. Cambia questo blocco in **"Premi la barra spaziatrice!"**, quindi aggiungi un blocco **azzerà cronometro** dalla categoria **Sensori**. Questo blocco controlla una variabile speciale di Scratch che ti consente di cronometrare alcune azioni e servirà per capire quanto velocemente reagisci nel gioco.

quando si clicca su

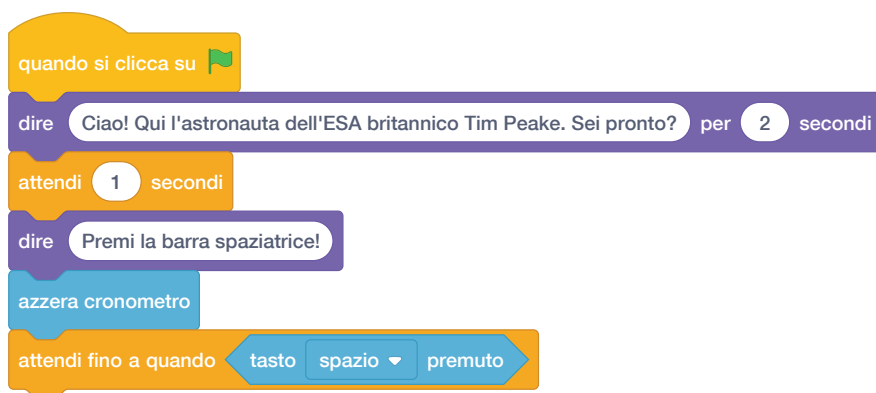
dire Ciao! Qui l'astronauta dell'ESA britannico Tim Peake. Sei pronto? per 2 secondi

attendi 1 secondi

dire Premi la barra spaziatrice!

azzerà cronometro

Aggiungi un blocco **Controllo** *attendi fino a quando*, quindi trascina un blocco **Sensori** *tasto spazio premuto* nel suo spazio bianco. Il programma rimarrà in pausa finché non premi il tasto **SPAZIO** sulla tastiera, ma il cronometro continuerà ad andare avanti e cronometrerà il tempo intercorso tra il messaggio che richiede di "**Premi la barra spaziatrice!**" e il momento in cui hai premuto **SPAZIO**.



Ora hai bisogno che Tim ti dica quanto tempo hai impiegato a premere il tasto **SPAZIO** in un modo facile da leggere. Per farlo dovrai utilizzare un blocco **Operatori** *unione*. Questo prenderà due valori, comprese le variabili, e li unirà uno dopo l'altro, azione nota come *concatenazione*.

Inizia con un blocco **dire Ciao!**, poi trascina un blocco **Operatori** *unione* sulla parola **Ciao!**. Fai clic su **mela** e digita **Il tuo tempo di reazione è stato**, accertandoti di aggiungere uno spazio alla fine, quindi trascina un altro blocco **unione di** sulla **banana** nella seconda casella. Trascina un blocco **Reporter** *cronometro* dalla categoria **Sensori** nella casella centrale e digita **secondi** nell'ultima casella, accertandoti di includere uno spazio all'inizio.

Infine, trascina un blocco **Variabili** *porta la mia variabile a 0* alla fine della sequenza. Fai clic sulla freccia verso il basso accanto a "**la mia variabile**", quindi fai clic su "**tempo**" dall'elenco, infine sostituisci lo **0** con un blocco **Reporter** *cronometro* dalla categoria **Sensori**. Puoi testare il tuo gioco facendo clic sulla bandiera verde sopra allo stage. Preparati e, non appena vedi il messaggio "**Premi la barra spaziatrice!**", premi il tasto **SPAZIO** il più velocemente possibile (Figura 4-13).

Cerca di battere il nostro punteggio più alto!

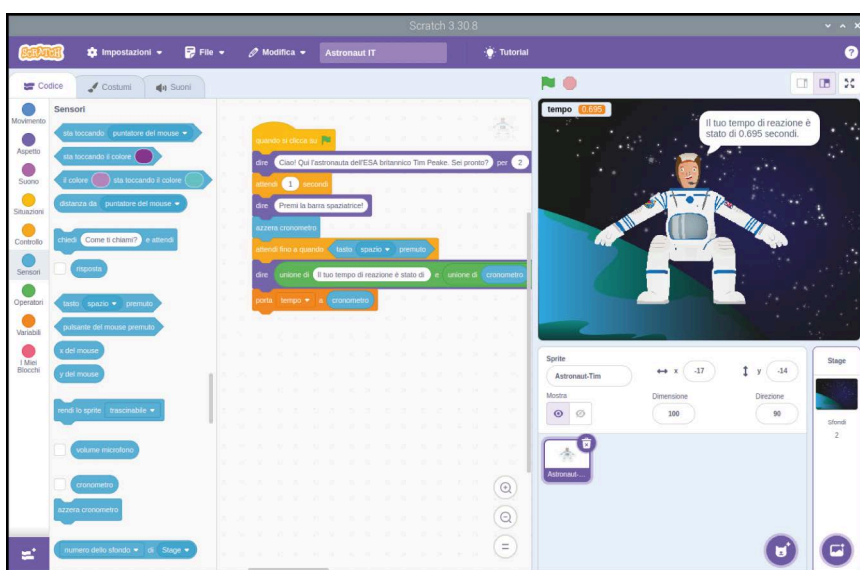
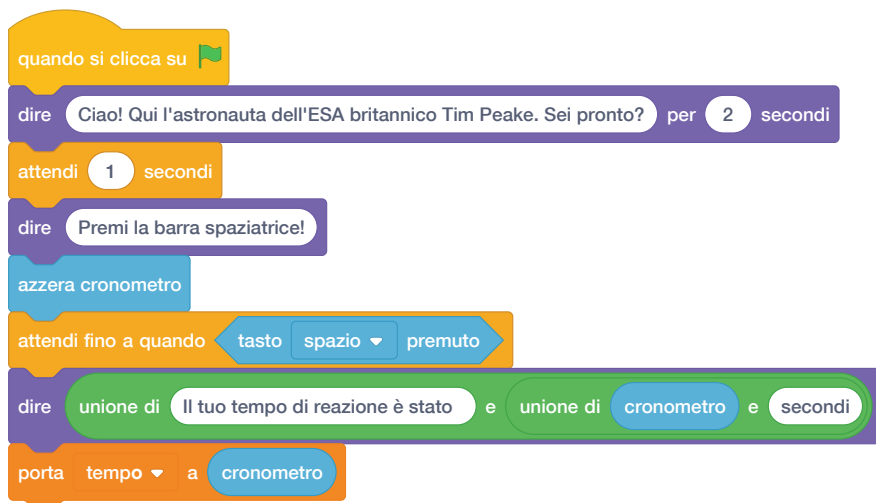
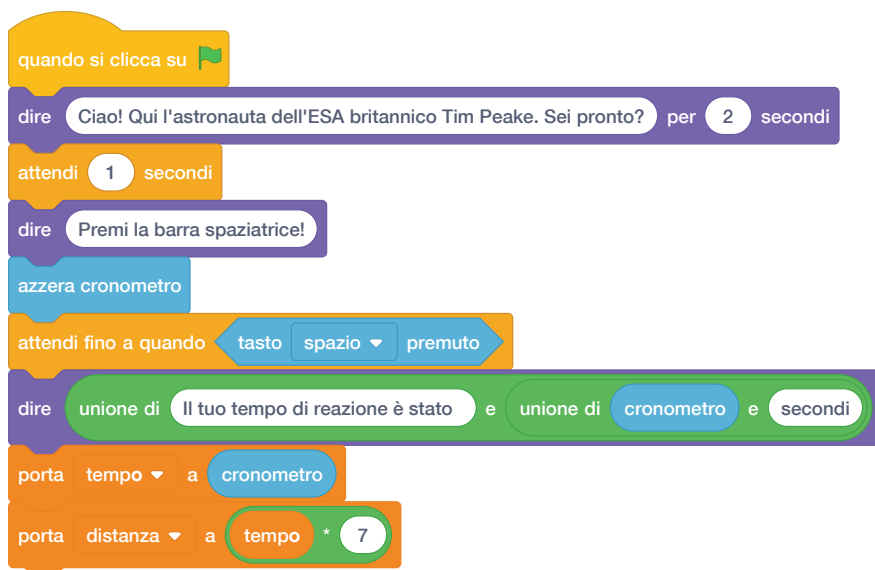


Figura 4-13 È arrivato il momento di giocare!

Puoi estendere ulteriormente questo progetto facendo calcolare la distanza approssimativa percorsa dalla Stazione Spaziale Internazionale nel tempo che ti è occorso per premere il tasto **SPAZIO** in base alla velocità di sette chilometri al secondo pubblicata dalla stazione. Per prima cosa, crea una nuova variabile chiamata **distanza**. Osserva come i blocchi nella categoria **Variabili** cambiano automaticamente per mostrare la nuova variabile, ma i blocchi di variabili **tempo** già presenti nel programma rimangono invariati.

Aggiungi un blocco **porta distanza a 0**, poi trascina un blocco **Operatori** $\circ * \circ$ (che indica la moltiplicazione) sopra il blocco **0**. Trascina un blocco **Reporter** **tempo** sul primo spazio bianco, quindi digita il numero **7** nel secondo spazio. Quando avrai finito, il blocco combinato indicherà **porta distanza a tempo * 7**. Il tempo che ti è occorso per premere il tasto **SPAZIO** verrà quindi moltiplicato per sette per ottenere la distanza percorsa in chilometri dalla stazione spaziale.



Aggiungi un blocco **attendi 1 secondi** e cambialo in **4**. Infine, trascina un altro blocco **dire Ciao!** alla fine della sequenza e aggiungi due blocchi **unione** proprio come hai fatto prima. Nel primo spazio, sopra **mela**, digita **In quel lasso di tempo la ISS percorre circa**, ricordandoti di aggiungere uno spazio alla fine; nello spazio **banana**, digita **chilometri.**, ricordandoti di aggiungere uno spazio all'inizio.

Infine, trascina un blocco **Operatori** **unione** nello spazio vuoto centrale, quindi trascina un blocco **Reporter** **distanza** nel nuovo spazio vuoto creato. Il blocco **unione** arrotonda i numeri per eccesso o per difetto al numero intero più vicino, così si ottiene un numero di facile lettura dei chilometri, anziché un numero preciso ma difficile da leggere.

Fai clic sulla bandiera verde per eseguire il programma e vedere quanto si sposta la stazione spaziale nel tempo che ti è occorso a premere il tasto **SPAZIO** (Figura 4-14). Ricordati di salvare il programma quando hai finito in modo da poterlo caricare facilmente in futuro senza dover ricominciare da capo.

SFIDA: GRAFICA PERSONALIZZATA



Fai clic su uno sprite o su uno sfondo, quindi sulla scheda **Costumi** o **Sfondi** per visualizzare un editor con gli strumenti di disegno. Riesci a disegnare i personaggi e lo sfondo e modificare il codice per far dire al tuo personaggio qualcosa di diverso?

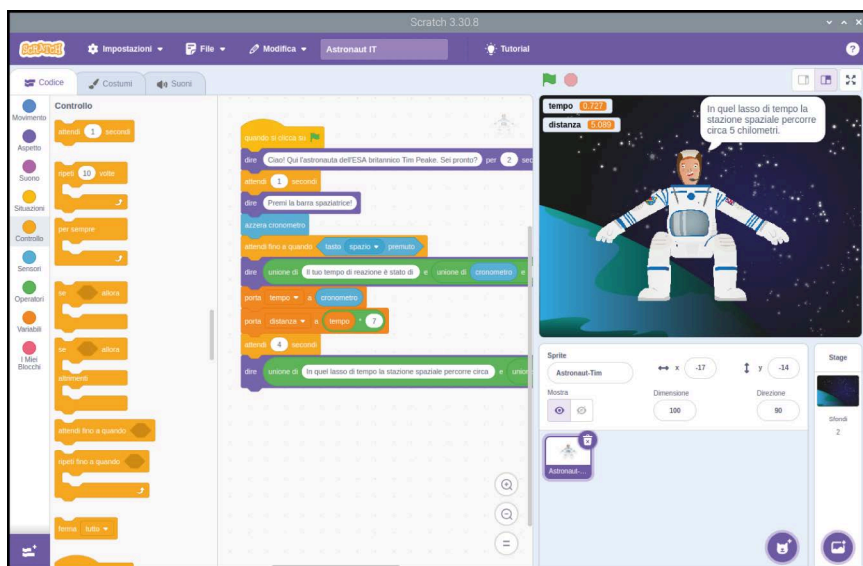
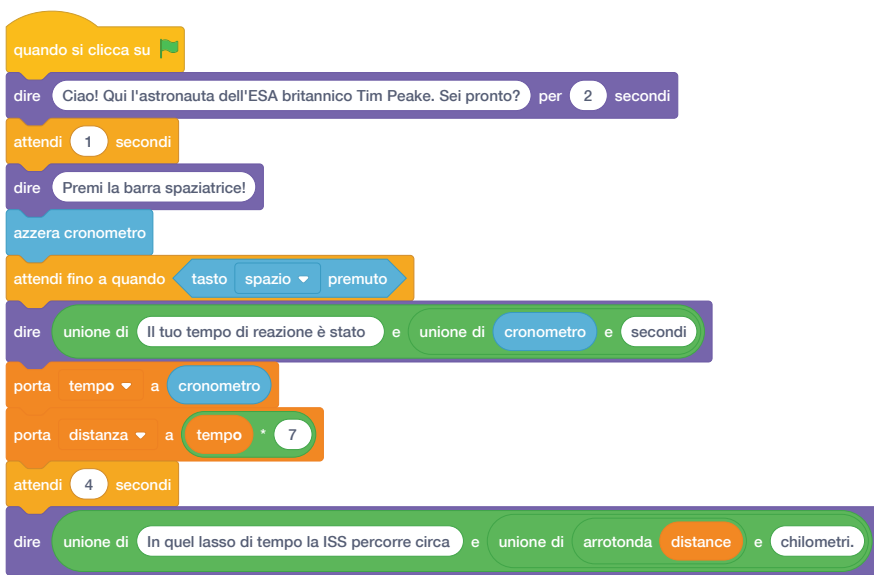


Figura 4-14 Tim ti dice quanta distanza ha percorso la stazione spaziale

Progetto 2: nuoto sincronizzato

La maggior parte dei giochi utilizza più di un tasto. Questo progetto lo dimostra, offrendo un controllo a due pulsanti utilizzando i tasti freccia destra e sinistra della tastiera.

Crea un nuovo progetto e salvalo come "Nuoto sincronizzato". Fai clic su **Stage** nella sezione dei controlli Stage, quindi fai clic sulla scheda **Sfondi** in alto a sinistra. Fai clic sul pulsante **Converti in Bitmap** sotto lo sfondo. Scegli un colore blu simile all'acqua dalla tavolozza **Riempimento** e fai clic sull'icona **Riempimento**.

Successivamente, fai clic sullo sfondo a scacchiera per colorarlo di blu (vedi Figura 4-15).

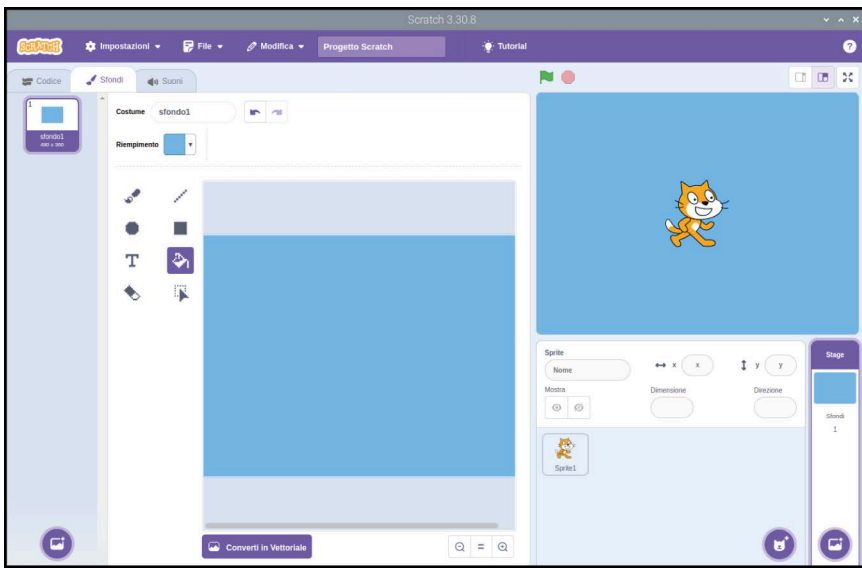


Figura 4-15 Riempi lo sfondo con un colore blu

Fai clic con il tasto destro del mouse sullo sprite gatto nell'elenco e fai clic su **cancella**. Fai clic sull'icona **Scegli uno Sprite** per aprire una lista di sprite predefiniti. Fai clic sulla categoria **Animali**, poi su **"Cat Flying"** (Figura 4-16), quindi su **OK**. Questo sprite è molto adatto anche ai progetti di nuoto.

Fai clic sul nuovo sprite, quindi trascina due blocchi **Situazioni** quando si preme il tasto spazio nell'area del codice. Fai clic sulla piccola freccia verso il basso accanto alla parola "spazio" nel primo blocco e scegli **freccia sinistra** dall'elenco delle opzioni possibili. Trascina un blocco **Movimento** ruota in senso antiorario di 15 gradi sotto il blocco

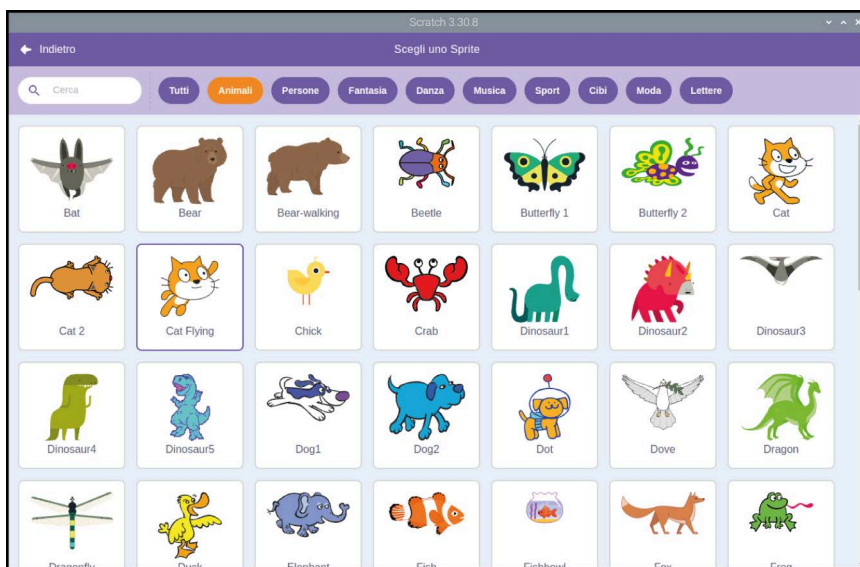


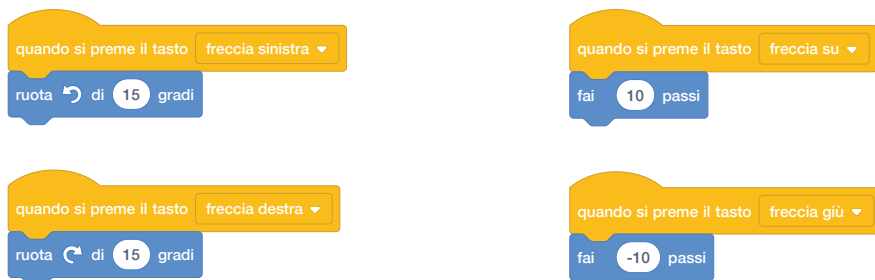
Figura 4-16 Scegli uno sprite dalla libreria

quando si preme il tasto freccia sinistra , poi fai lo stesso con il secondo blocco **Situazioni**, ma scegli **freccia destra** dall'elenco e utilizza un blocco **Movimento** ruota in senso orario di 15 gradi .



Premi il tasto freccia sinistra o destra per testare il tuo programma. Vedrai lo sprite gatto girare in corrispondenza della direzione scelta sulla tastiera. Hai notato che questa volta non è stato necessario fare clic sulla bandiera verde? Questo perché i blocchi trigger **Situazioni** che hai utilizzato sono sempre attivi, anche quando il programma non è in esecuzione nel senso consueto del termine.

Esegui di nuovo la stessa procedura per due volte, ma questa volta scegli **freccia su** e **freccia giù** per i blocchi trigger **Situazioni**, poi **fai 10 passi** e **fai -10 passi** per i blocchi **Movimento**. Premi i tasti freccia ora e vedrai che il gatto può girarsi e anche nuotare in avanti e indietro!



Per rendere più realistico il movimento dello sprite gatto, è possibile modificare il suo aspetto utilizzando i cosiddetti *costumi*. Fai clic sullo sprite gatto, quindi sulla scheda **Costumi** sopra la palette dei blocchi. Fai clic sul costume "cat flying-a" e sull'icona del cestino nell'angolo in alto a destra per cancellarlo. Successivamente fai clic sul costume "cat flying-b" e usa la casella del nome in alto per rinominarlo "destra" (Figura 4-17).

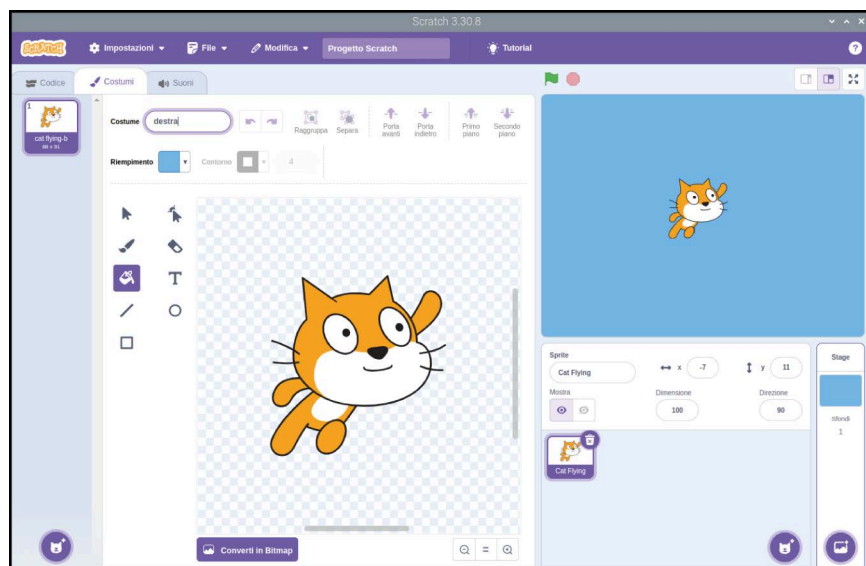


Figura 4-17 Rinominare il costume come "destra"

Fai clic con il tasto destro del mouse sul costume appena rinominato "destra" e fai clic su **duplica** per crearne una copia. Fai clic su questa copia per sele-

zionarla, quindi fai clic sull'icona **Seleziona**  . Successivamente fai clic sull'icona **Rifletti in Orizzontale**  .

Infine, rinomina il costume duplicato in "sinistra" (vedi **Figura 4-18**). Avrai due "costumi" per il tuo sprite, che sono immagini speculari esatte: uno chiamato "destra" con il gatto rivolto a destra, e uno chiamato "sinistra" con il gatto rivolto a sinistra.

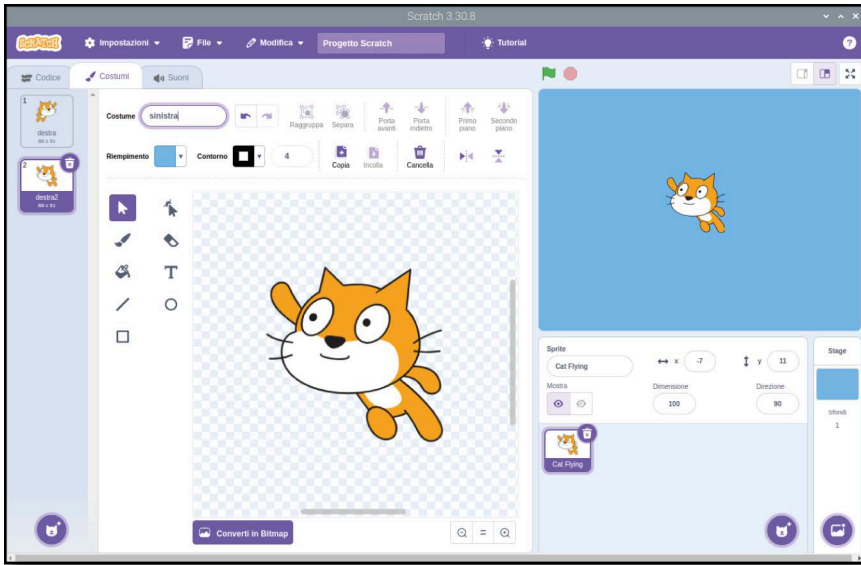


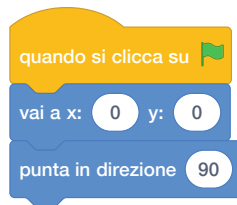
Figura 4-18 Duplica il costume, capovolgilo e chiamalo "sinistra"

Fai clic sulla scheda **Codice** sopra l'area del costume, quindi trascina due blocchi **Aspetto** **passa al costume sinistra** sotto i blocchi **Situazioni** freccia sinistra e destra, cambiando quello sotto il blocco freccia destra perché indichi **passa al costume destra** . Prova di nuovo i tasti freccia, il gatto ora gira nella direzione in cui sta nuotando.





Per il nuoto sincronizzato in stile olimpico, però, abbiamo bisogno di più nuotatori, quindi di un modo per ripristinare la posizione dello sprite gatto. Aggiungi un blocco **Situazioni** `quando si clicca su` (bandiera verde), poi aggiungi un blocco **Movimento** `vai a x: 0 y: 0` (modificando i valori se necessario) e un blocco **Movimento** `punta in direzione 90`. Ora, quando fai clic sulla bandiera verde il gatto si sposterà al centro dello stage e sarà rivolto a destra.



Per creare più nuotatori, aggiungi un blocco `ripeti 6 volte`, cambiando il valore predefinito di "10", e aggiungi un blocco **Controllo** `crea clone di me stesso` al suo interno. Per fare in modo che i nuotatori non nuotino tutti nella stessa direzione, aggiungi un blocco `ruota in senso orario di 60 gradi` sopra il blocco `crea clone di`, ma all'interno del blocco `ripeti 6 volte`. Fai clic sulla bandiera verde e prova i tasti freccia per vedere i tuoi nuotatori muoversi!



Per un'atmosfera più olimpica, aggiungi un po' di musica. Fai clic sulla scheda **Suoni** sopra la palette dei blocchi, quindi sull'icona **Scegli un Suono** (🔊). Fai clic sulla categoria **Loop**, poi sfoglia l'elenco (Figura 4-19) fino a trovare la

musica che ti piace, noi abbiamo scelto **"Dance Around"**. Fai clic sul pulsante **OK** per scegliere la musica, quindi fai clic sulla scheda **Codice** per aprire nuovamente l'area del codice.

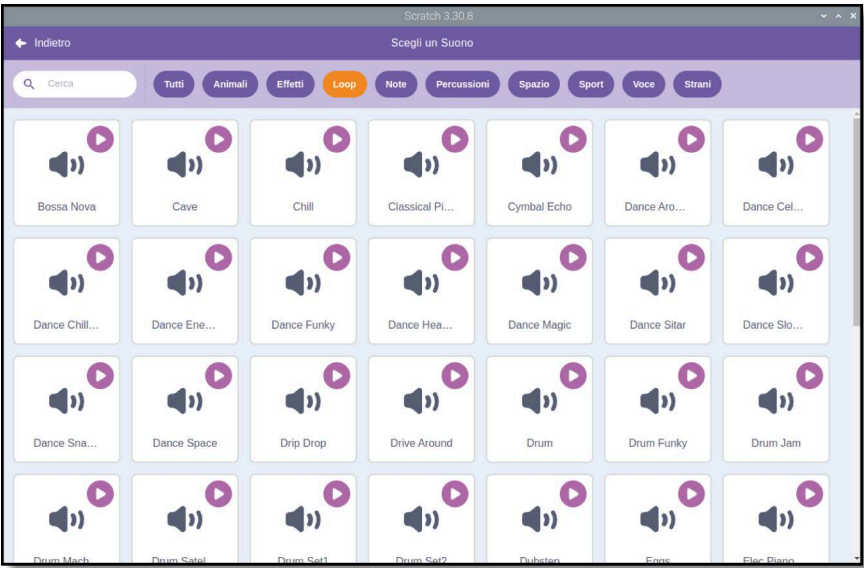
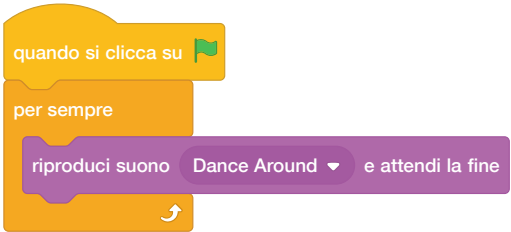


Figura 4-19 Seleziona un loop musicale dalla libreria di suoni

Aggiungi un altro blocco **Situazioni** **quando si clicca su** alla tua area codice, quindi aggiungi un blocco **Controllo** **per sempre**. Nel blocco **Controllo**, aggiungi un blocco **riproduci suono dance around e attendi la fine**, ricordandoti di cercare il nome del brano musicale scelto, quindi fai clic sulla bandiera verde per testare il nuovo programma. Se vuoi fermare la musica, fai clic sul pulsante Ferma tutto (ottagono rosso) per fermare il programma e la musica.



Infine, è possibile simulare una routine di ballo completa aggiungendo un nuovo trigger di eventi al programma. Aggiungi un blocco **Situazioni** **quando si preme il tasto spazio**, poi un blocco **passa al costume destra**. Sotto di esso, aggiungi un blocco **ripeti 36 volte** (ricordati di modificare il valore pre-

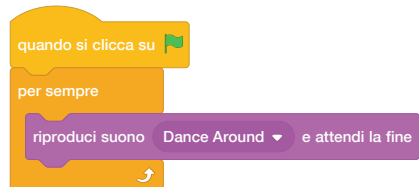
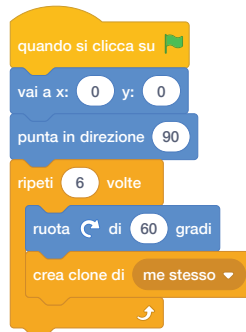
definito) e all'interno di questo un blocco **ruota in senso orario di 10 gradi** e un blocco **fai 10 passi**.

Fai clic sulla bandiera verde per avviare il programma, quindi premi il tasto **SPAZIO** per provare la nuova routine (**Figura 4-20**), e salva il tuo programma quando hai finito.



SFIDA: ROUTINE PERSONALIZZATA

Puoi creare la tua routine di nuoto sincronizzato utilizzando loop? Cosa dovresti cambiare per includere più o meno nuotatori? Puoi aggiungere più routine di nuoto attivabili con diversi tasti della tastiera?



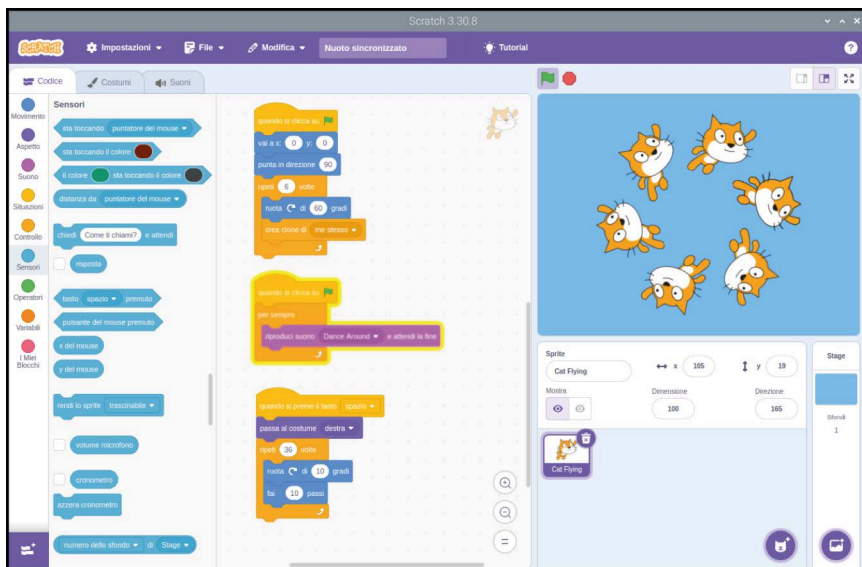


Figura 4-20 Nuoto sincronizzato completo

Progetto 3: gioco di tiro con l'arco

Ora che stai diventando un po' esperto di Scratch, è il momento di lavorare su qualcosa di un po' più impegnativo: un gioco di tiro con l'arco, in cui il giocatore deve colpire un bersaglio con un arco e una freccia che oscillano casualmente.

Inizia aprendo il browser Web Chromium e digitando **rptl.io/archery** quindi premendo il tasto **INVIO**. Le risorse del gioco sono contenute in un file zip, quindi dovrai estrarle. Per farlo, fai clic con il tasto destro del mouse sul file e seleziona **Estrai qui**. Torna a Scratch 3 e fai clic sul menu **File** seguito da **Carica dal computer**. Fai clic su **ArcheryResources.sb3** e poi su **Seleziona**. Ti verrà chiesto se vuoi sostituire il contenuto del tuo progetto attuale. Se non hai salvato le tue modifiche, fai clic su **Annulla** e salvale, quindi fai clic su **OK**.

Il progetto appena caricato contiene uno sfondo e uno sprite (Figura 4-21), ma dovrai aggiungere tu il codice del gioco. Inizia aggiungendo un blocco **quando si clicca su**, poi un blocco **invia a tutti messaggio1**. Fai clic sulla freccia in basso alla fine del blocco, poi **Nuovo Messaggio...** e digita **nuova freccia** prima di fare clic sul pulsante **OK**. Il tuo blocco ora sarà **invia a tutti nuova freccia**.

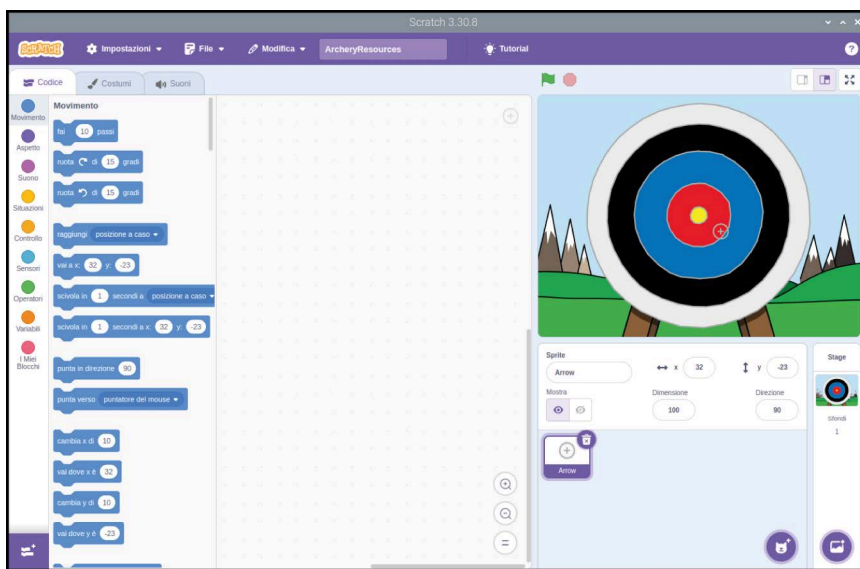
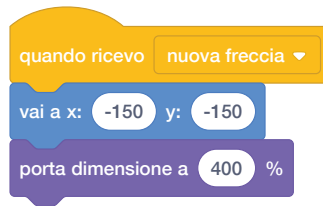


Figura 4-21 Risorse del progetto caricate per il gioco del tiro con l'arco

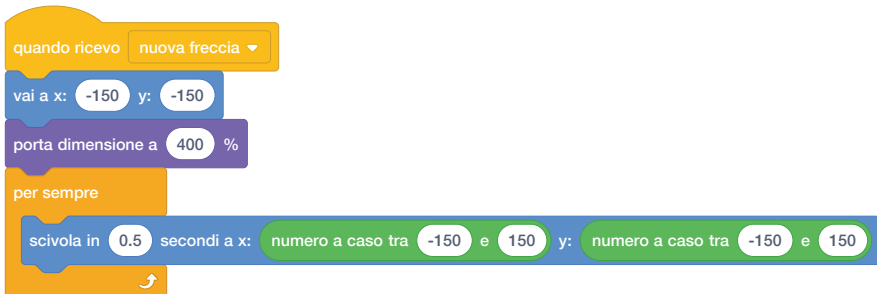


Un messaggio inviato a tutti (o broadcast) è un messaggio di una parte del programma che può essere ricevuto da qualsiasi altra parte del programma. Per fare in modo che faccia effettivamente qualcosa, aggiungi un blocco **quando ricevo messaggio1** e cambialo in **quando ricevo nuova freccia**. Questa volta dovrai solo fare clic sulla freccia verso il basso e scegliere **"nuova freccia"** dall'elenco, non è necessario creare di nuovo il messaggio.

Sotto il blocco **quando ricevo nuova freccia**, aggiungi un blocco **vai a x: -150 y: -150** e un blocco **porta dimensione a 400 %**. Ricorda che questi non sono i valori predefiniti per quei blocchi, quindi dovrai cambiarli una volta trascinati nell'area codice. Fai clic sulla bandiera verde per vedere quanto fatto finora: lo sprite freccia, che il giocatore usa per mirare al bersaglio, si sposterà nell'angolo inferiore sinistro dello stage e avrà dimensioni quadruplicate.

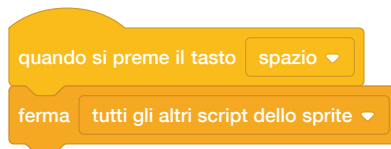


Per includere una sfida per il giocatore, aggiungi un movimento che simuli l'oscillazione dell'arco quando viene tirato indietro e l'arciere prende la mira. Trascina un blocco **per sempre**, seguito da un blocco **scivola in 1 secondi a x: -150 y: -150**. Modifica la prima casella bianca in **0.5** anziché **1**, poi colloca un blocco **Operatori** **numero a caso tra -150 e 150** in ognuna delle altre due caselle bianche. Questo significa che la freccia si sposterà sullo stage in una direzione casuale per una distanza casuale, rendendo così più difficile colpire il bersaglio.



Fai clic di nuovo sulla bandiera verde e vedrai cosa fa quel blocco: il tuo sprite freccia ora si sta spostando nello stage, coprendo diverse parti del bersaglio. Al momento, però, non c'è modo di scagliare la freccia verso il bersaglio.

Trascina un blocco **quando si preme il tasto spazio** nell'area del codice, seguito da un blocco **Controllo** **ferma tutto**. Fai clic sulla freccia verso il basso alla fine del blocco e modificala con un blocco **ferma tutti gli altri script dello sprite**.



Se hai fermato il tuo programma per aggiungere i nuovi blocchi, fai clic sulla bandiera verde per farlo ripartire e premi il tasto **SPAZIO**: vedrai lo sprite freccia smettere di muoversi. È un buon inizio, ma devi fare in modo che la freccia sembri volare verso il bersaglio. Aggiungi un blocco **ripeti 50 volte** seguito da un blocco **cambia dimensione di -10**, quindi fai clic sulla bandiera verde per testare di nuovo il gioco. Questa volta la freccia sembra volare via verso il bersaglio.



Per rendere il gioco più divertente, dovresti aggiungere un modo per tenere il punteggio. Nello stesso insieme di blocchi, aggiungi un blocco **se allora**, assicurandoti che sia al di sotto del blocco **ripeti 50 volte** e non all'interno, con un blocco **Sensori** **sta toccando il colore** nello spazio a forma di diamante. Per scegliere il colore corretto, fai clic sulla casella colorata alla fine del blocco **Sensori** e poi sull'icona **contagocce**. Successivamente fai clic sul centro del bersaglio giallo nello stage



Aggiungi un blocco **avvia riproduzione suono cheer** e un blocco **dire 200 punti per 2 secondi** all'interno del blocco **se allora** in modo che il giocatore sappia di aver segnato. Infine, aggiungi un blocco **invia a tutti nuova freccia** in fondo alla serie di blocchi, sotto e fuori dal blocco **se allora**, per dare al giocatore un'altra freccia ogni volta che ne scaglia una. Fai clic sulla bandiera verde per avviare la partita e cerca di colpire il centro giallo: quando lo farai, sarai ricompensato con un applauso dalla folla e un punteggio di 200 punti!



A questo punto il gioco funziona, ma potrebbe essere un po' troppo impegnativo. Usa quello che hai imparato in questo capitolo per aggiungere punteggi per i vari cerchi del bersaglio oltre al centro: 100 punti per il rosso, 50 punti per il blu e così via.


Per altri progetti Scratch da provare, consulta Appendice D, *Ulteriori letture*.

quando si clicca su 








invia a tutti  nuova freccia 



quando ricevo  nuova freccia 



vai a x:  -150 y:  -150

porta dimensione a  400 %

per sempre

scivola in  0.5 secondi a x:  numero a caso tra  -150 e  150 y:  numero a caso tra  -150 e  150

quando si preme il tasto  spazio 

ferma  tutti gli altri script dello sprite 

ripeti  50 volte

cambia dimensione di  -10

se  sta toccando il colore  allora

avvia riproduzione suono  cheer 

dire  200 punti per  2 secondi

invia a tutti  nuova freccia 

SFIDA: PUOI MIGLIORARLO?

Come puoi renderlo più semplice? Come puoi renderlo più difficile? Puoi usare le variabili per far aumentare il punteggio del giocatore quando lancia più frecce? Puoi aggiungere un timer per il conto alla rovescia per mettere più pressione al giocatore?



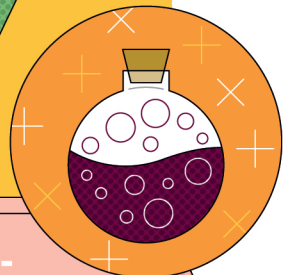
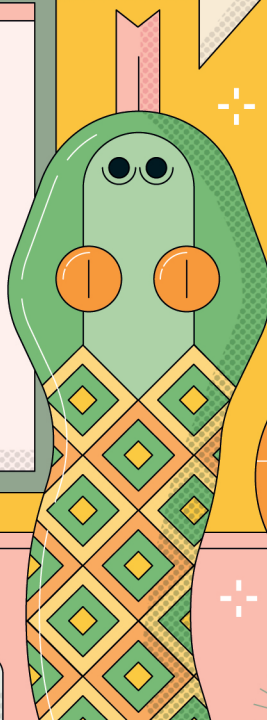
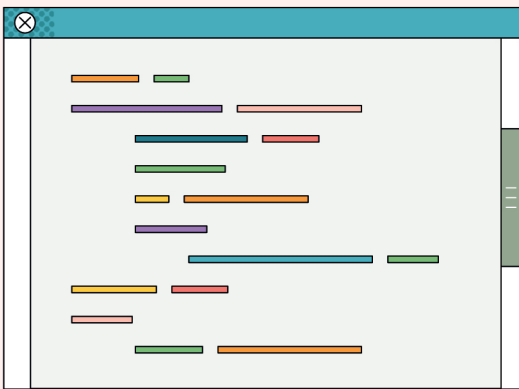


Hello,

world!

def

#



Capitolo 5

Programmare con Python

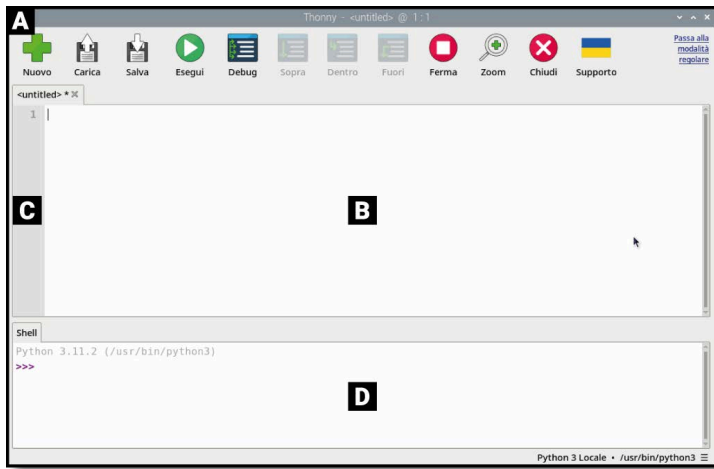
Ora che hai imparato a usare Scratch, affronteremo la programmazione basata su testo con Python.

Python, che prende il nome dal gruppo comico Monty Python, dal 1991 ad oggi è passato dall'essere un progetto creato per hobby da Guido van Rossum a un linguaggio di programmazione molto apprezzato alla base di numerosi progetti. A differenza dell'ambiente visivo di Scratch, Python si basa su testo: si scrivono istruzioni in un linguaggio semplificato e con un formato specifico, che poi verranno eseguite dal computer.

Grazie alla flessibilità e a un ambiente di programmazione più tradizionale, è un ottimo programma per fare un passo in più rispetto a Scratch. Non è difficile da apprendere: tutti, con un po' di pratica, possono imparare a scrivere programmi Python per realizzare i progetti più disparati, da semplici calcoli a giochi più complessi.

In questo capitolo svilupperemo ulteriormente i termini e i concetti introdotti nel Capitolo 4, *Programmare con Scratch 3*. Ti consigliamo quindi di provare le esercitazioni introdotte nel capitolo precedente, così ti sarà più facile seguire questo capitolo.

Panoramica di Thonny Python IDE



A Barra degli strumenti

C Numeri di riga

B Area script

D Python shell

L'interfaccia semplice di Thonny utilizza come menu una barra di icone intuitive (A) che permettono di creare, salvare, caricare ed eseguire i programmi Python, oltre che testarli in vari modi. L'area script (B) è il luogo in cui vengono scritti i programmi Python, ed è suddivisa in un'area principale per il programma e un piccolo margine laterale che mostra i numeri di riga (C). Python Shell (D) consente di digitare singole istruzioni che vengono poi eseguite non appena si preme il tasto **INVIO** e fornisce anche informazioni sull'esecuzione dei programmi.



INTERFACCE DI THONNY

Thonny ha due versioni principali della sua interfaccia: "Regular Mode" e "Simple Mode", che è quella consigliata ai principianti. In questo capitolo viene utilizzata la Simple Mode, che viene caricata di default quando si apre Thonny dalla sezione **Programmazione** del menu di Raspberry Pi.

Per modificare la lingua di Thonny, fai clic su **Python 3 Locale** in basso a destra della finestra di Thonny e poi su **Configura l'interprete....** Successivamente fai clic sulla scheda **Generale**, scegli la tua lingua e fai clic su **OK**.

Il tuo primo programma Python: ciao mondo!

Thonny si apre come qualsiasi altro programma su Raspberry Pi: fai clic sull'icona di Raspberry Pi per aprire il menu, sposta il cursore sulla sezione **Programmazione** e fai clic su **Thonny**. L'interfaccia utente di Thonny (Simple Mode di default) si aprirà dopo pochi secondi.

Thonny è un pacchetto che rientra negli *ambienti di sviluppo integrato* (*integrated development environment, IDE*), un nome dal suono complicato con una semplice spiegazione: riunisce, o *integra*, tutti i vari strumenti necessari per scrivere, ossia *sviluppare*, software in un'unica interfaccia utente, detta anche *ambiente*. Ci sono numerosi IDE, alcuni di questi supportano molti linguaggi di programmazione diversi, mentre altri, come Thonny, si concentrano su un singolo linguaggio.

A differenza di Scratch, che consente di creare programmi partendo da blocchi visivi, Python è un linguaggio di programmazione più tradizionale dove è necessario scrivere il codice. Crea il tuo primo programma facendo clic sull'area Python shell in fondo alla finestra Thonny, quindi digita le seguenti istruzioni prima di premere il tasto **INVIO**:

```
print("Ciao mondo!")
```

Quando premi **INVIO**, vedrai che il programma viene immediatamente eseguito: Python risponderà nella stessa area shell con il messaggio "Ciao a tutti!" (**Figura 5-1**), proprio come hai chiesto. La shell infatti è collegata direttamente all'*interprete* Python, che ha il compito di esaminare le istruzioni scritte e *interpretarne* il significato. Questa modalità è detta *interattiva* e possiamo paragonarla a una conversazione faccia a faccia con qualcuno: quando hai finito di parlare, l'altra persona risponderà, poi aspetterà qualsiasi cosa tu dica dopo.

ERRORE DI SINTASSI

Se il programma non viene eseguito, ma stampa invece un messaggio di errore di sintassi nell'area shell, significa che c'è un errore da qualche parte in quello che hai scritto. In Python è necessario scrivere le istruzioni in un modo molto specifico: se mancano una parentesi o una virgoletta, si scrive "print" in modo sbagliato o con una P maiuscola oppure si aggiungono altri simboli da qualche parte nelle istruzioni, il programma non funzionerà. Prova a digitare di nuovo le istruzioni e assicurati che corrispondano a quanto indicato in questa guida prima di premere il tasto **INVIO**.








Figura 5-1 Python stampa il messaggio "Ciao a tutti!" nell'area shell

Non è necessario utilizzare sempre Python in modalità interattiva. Fai clic sull'area script al centro della finestra Thonny, quindi digita di nuovo l'istruzione:

```
print("Ciao mondo!")
```

Una volta premuto il tasto **INVIO** questa volta non succederà nulla, ma passerai a una riga vuota nell'area dello script. Per far funzionare questa versione del programma, dovrai fare clic sull'icona **Esegui**  nella barra degli strumenti Thonny. Prima di farlo, però, è necessario fare clic sull'icona **Salva** . Assegna al tuo programma un nome descrittivo, come **Ciao mondo.py** e fai clic sul pulsante **OK**. Una volta salvato il programma, fai clic sull'icona **Esegui**  e verranno visualizzati due messaggi nell'area Python shell (**Figura 5-2**):

```
>>> %Run 'Ciao.py'  
Ciao mondo!
```

La prima di queste righe è un'istruzione di Thonny che dice all'interprete Python di eseguire il programma appena salvato, mentre la seconda è l'output del programma, ossia il messaggio che hai detto di stampare a Python. Complimenti, hai scritto ed eseguito il tuo primo programma Python sia in modalità interattiva, sia script!

SFIDA: NUOVO MESSAGGIO

È possibile modificare il messaggio che il programma Python stampa come output? Useresti la modalità interattiva o la modalità script per aggiungere altri messaggi? Cosa succede se rimuovi le parentesi o le virgolette dal programma e poi cerchi di eseguirlo?

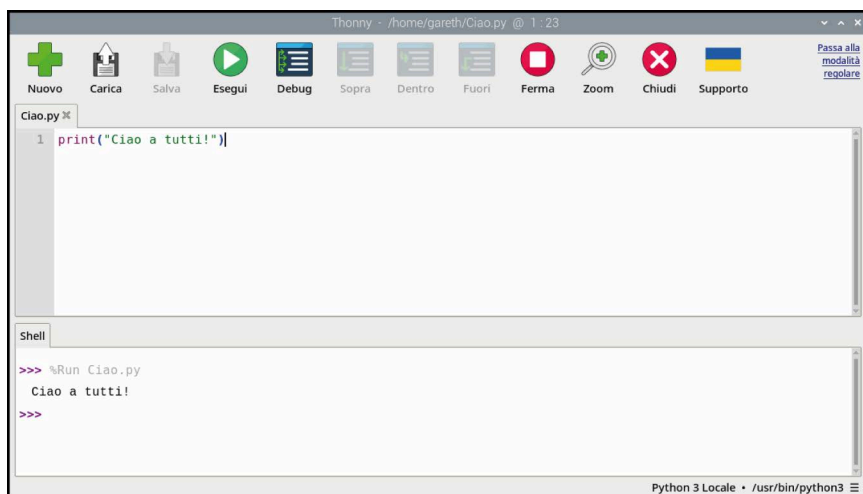



Figura 5-2 Eseguire un semplice programma

Prossimi passaggi: loop e indentazione del codice

Per controllare il collegamento tra le varie parti del programma Scratch si serve di blocchi simili a puzzle, mentre Python controlla la sequenza in cui vengono eseguiti i programmi tramite l'*indentazione*, ossia la rientranza. Crea un programma facendo clic sull'icona **Nuovo**  nella barra degli strumenti Thonny. Non perderai il programma che hai già scritto, Thonny infatti creerà una nuova scheda sopra l'area dello script. Inizia digitando quanto segue:

```
print("Il loop sta iniziando!")
for i in range(10):
```

La prima riga stampa un semplice messaggio in shell, proprio come il programma Ciao mondo. La seconda avvia un loop *definito* che funziona come in Scratch: al loop viene assegnato un contatore, **i**, e una serie di numeri. Si tratta dell'istruzione **range**, che indica al programma di iniziare dal numero 0 e di

procedere in ordine crescente, senza mai raggiungere il numero 10. Il simbolo dei due punti (:) indica a Python di includere l'istruzione successiva nel loop.

In Scratch le istruzioni da includere nel loop sono incluse nel blocco a forma di C. Python utilizza un approccio diverso, ossia l'indentazione del codice. La riga successiva inizia con quattro spazi vuoti aggiunti da Thonny una volta premuto **INVIO** dopo la riga 2:

```
print("Numero di loop", i)
```



La riga risulterà quindi rientrante rispetto alle altre. Con questa indentazione Python distinguerà le istruzioni fuori dal loop da quelle all'interno, il codice indentato è *annidato*.

Noterai che quando hai premuto **INVIO** alla fine della terza riga, Thonny ha indentato automaticamente la linea successiva, assumendo che facesse parte del loop. Per rimuovere la rientranza, premi semplicemente il tasto **BACKSPACE** prima di digitare la quarta riga:

```
print("Loop ha finito!!")
```

Il tuo programma è completo. La prima riga si trova al di fuori del loop e verrà eseguita una sola volta. La seconda riga imposta il loop; la terza si trova all'interno del loop e verrà eseguita una volta per ogni ciclo del loop. La quarta riga si trova ancora una volta al di fuori del loop.

```
print("Il loop sta iniziando!")
for i in range(10):
    print("Numero di loop", i)
print("Loop ha finito!!")
```

Fai clic sull'icona **Salva** , salva il programma come **Indentazione.py**, quindi fai clic sull'icona **Esegui**  e guarda l'area shell per vedere il risultato (vedi Figura 5-3):

```
Il loop sta iniziando!
Numero di loop 0
Numero di loop 1
Numero di loop 2
Numero di loop 3
Numero di loop 4
Numero di loop 5
Numero di loop 6
Numero di loop 7
Numero di loop 8
```

Numero di loop 9
Loop ha finito!!

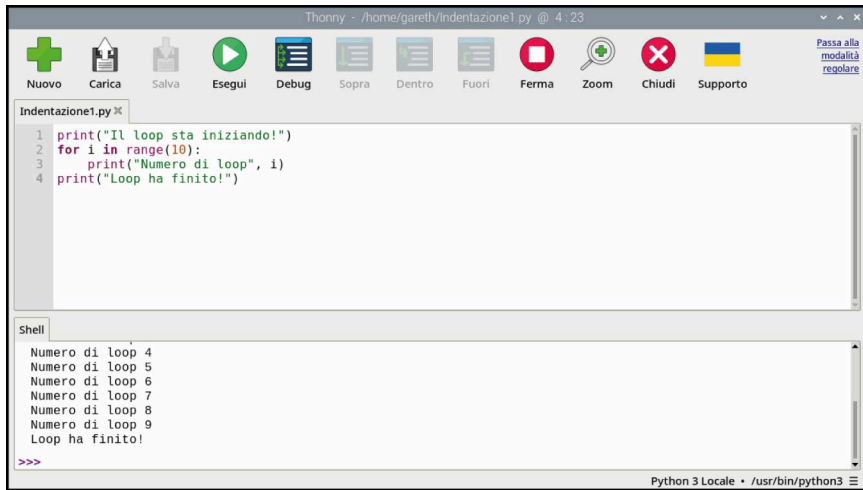


Figura 5-3 Esecuzione di un loop

CONTARE DA ZERO

Python è un linguaggio a indice zero, il che significa che inizia a contare da 0 e non da 1, ed è per questo che il programma stampa i numeri da 0 a 9 anziché da 1 a 10. Puoi modificare questo comportamento, se lo desideri, passando dall'istruzione **range(10)** a **range(1, 11)** o a qualsiasi altro numero.

L'indentazione è una parte fondamentale e integrante di Python ed è uno dei motivi più comuni per cui un programma non funziona come ci si aspetta. Quando si cercano problemi in un programma, un processo noto come *debugging*, controlla sempre le indentazioni, soprattutto quando inizi ad annidare i loop in altri loop.


Python supporta anche loop *infiniti*. Per passare da un ciclo definito a uno infinito, modifica la riga 2 affinché diventi:

while True:

Facendo clic sull'icona **Esegui** in questo momento, visualizzerai un errore: **name 'i' is not defined**. Perché hai eliminato la riga che creava e assegnava un valore alla variabile **i**.

Per risolvere questo problema, modifica la riga 3 in modo che non utilizzi più la variabile:

```
print("Loop è in funzione!")
```

Fai clic sull'icona **Esegui**  e, se sei veloce, vedrai il messaggio **"Il loop sta iniziando!"** seguito da una stringa infinita di messaggi **"Loop è in funzione!"** (Figura 5-4). Il messaggio **"Loop ha finito!!"** non verrà mai stampato perché il ciclo non ha fine: ogni volta che Python termina di stampare il messaggio **"Loop è in funzione!"** tornerà all'inizio della ripetizione e stamperà nuovamente.

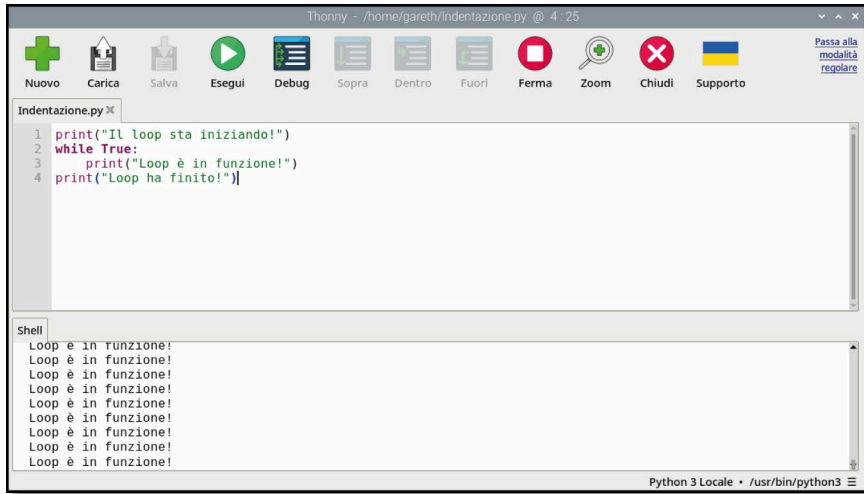



Figura 5-4 Un loop infinito continua fino a quando non si interrompe il programma


Fai clic sull'icona **Ferma**  sulla barra degli strumenti Thonny per dire al programma di interrompere ciò che sta facendo (*interruzione* del programma). Vedrai un messaggio nell'area shell Python e il programma si arresterà senza raggiungere la riga 4.





SFIDA: LOOPING DEL LOOP

È possibile cambiare il loop e renderlo definito? Puoi aggiungere un secondo loop definito al programma? Come aggiungeresti un loop all'interno di un loop e quale comportamento ti aspetteresti?

Condizionali e variabili

Le variabili in Python, come in tutti i linguaggi di programmazione, non servono solamente a controllare le ripetizioni. Crea un programma facendo clic sull'icona **Nuovo**  nel menu Thonny, quindi nell'area script digita quanto segue:

```
userName = input("Come ti chiami? ")
```

Fai clic sull'icona **Salva** , salva il programma come **Test nome.py**, fai clic su **Esegui**  e guarda cosa succede nell'area shell. Dovresti visualizzare un prompt che chiede di inserire il tuo nome. Digita il tuo nome nell'area shell, seguito da **INVIO**. Poiché questa è l'unica istruzione del programma, non accadrà nient'altro (**Figura 5-5**). Se vuoi fare qualcosa con i dati inseriti nella variabile, avrai bisogno di più righe nel programma.

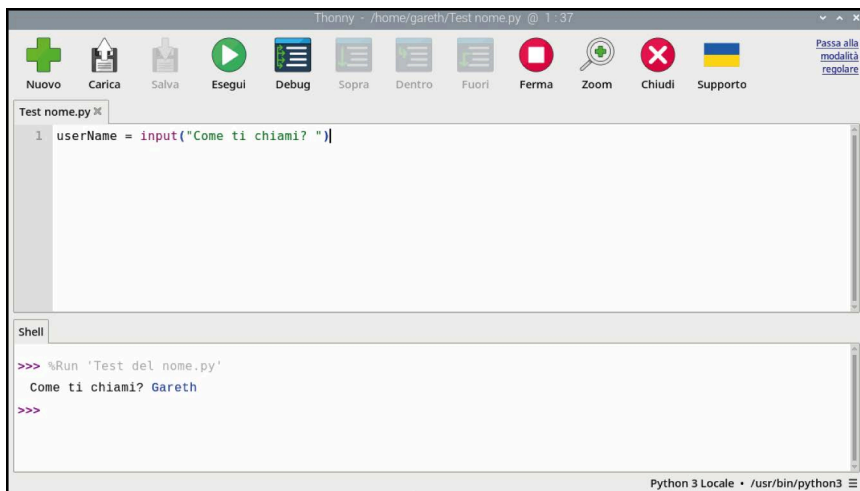




Figura 5-5 La funzione `input` ti consente di chiedere a un utente di inserire del testo

Per far sì che il programma faccia qualcosa con il nome, aggiungi un'istruzione condizionale digitando quanto segue:

```
if userName == "Clark Kent":  
    print("Sei Superman!")  
else:  
    print("Non sei Superman!")
```

Ricorda che quando Thonny riconosce che il codice deve essere indentato, lo farà automaticamente. Tuttavia, non sa quando il codice non deve più essere rientrante, quindi dovrai cancellare tu gli spazi prima di digitare **else:**.

Fai clic su **Esegui**  e digita il tuo nome nell'area shell. A meno che il tuo nome non sia Clark Kent, visualizzerai il messaggio "Non sei Superman!". Fai di nuovo clic su **Esegui**  e questa volta digita il nome "Clark Kent" assicurandoti di scriverlo esattamente come nel programma, con C e K maiuscole (Figura 5-6).

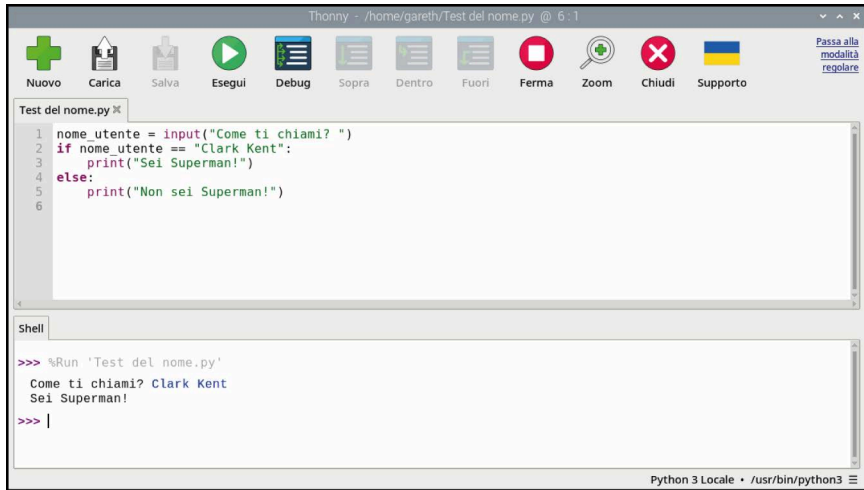


Figura 5-6 Non dovresti essere in giro a salvare il mondo?

I simboli **==** indicano a Python di fare un confronto diretto e quindi individuare se la variabile **userName** corrisponde al testo (*stringa*) nel programma. Se lavori con i numeri puoi fare altri confronti: **>** per vedere se un numero è maggiore di un altro numero, **<** per vedere invece se è minore, **=>** per vedere se è uguale o maggiore e **=<** per vedere se è uguale o minore. C'è anche **!=**, che significa diverso da, ossia l'esatto opposto di **==**. Questi simboli sono noti come *operatori di confronto*.





UTILIZZO DI = E ==

La chiave per utilizzare le variabili è conoscere la differenza tra **=** e **==**. Ricorda: **=** significa "rendi questa variabile uguale a questo valore", mentre **==** significa "controlla se la variabile è uguale a questo valore". Se vengono confuse, il programma non funzionerà come ti aspetti.

Puoi utilizzare gli operatori di confronto anche in un loop. Cancella le righe dalla 2 alla 5, quindi digita quanto segue:

```
while userName != "Clark Kent":
    print("Non sei Superman, prova di nuovo!")
    userName = input("Come ti chiami? ")
print("Sei Superman!")
```

Fai nuovamente clic sull'icona  **Esegui**. Questa volta, invece di smettere, il programma continuerà a chiederti il nome fino a quando non avrà la conferma che sei Superman (**Figura 5-7**), una sorta di password molto semplice. Per interrompere il loop, digita "Clark Kent" o fai clic sull'icona **Ferma**  sulla barra degli strumenti di Thonny. Complimenti, ora sai come usare condizionali e variabili!

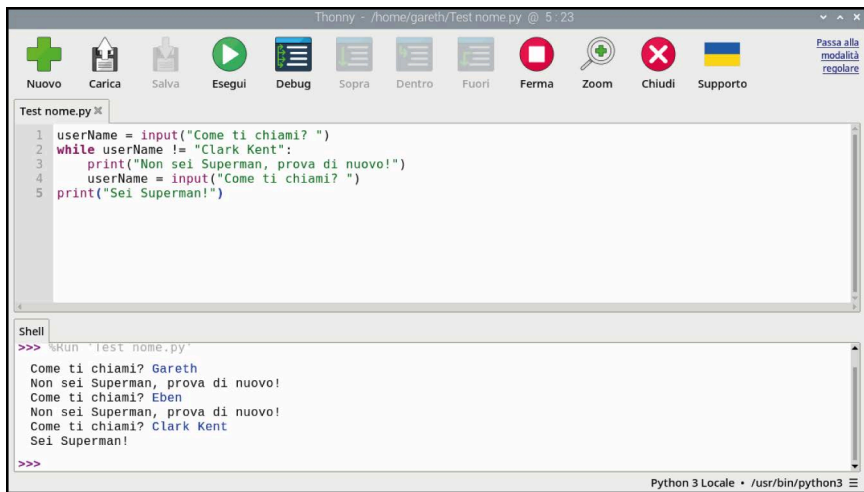


Figura 5-7 Continuerà a chiedere il tuo nome finché non dirai che è "Clark Kent"

SFIDA: AGGIUNGI ALTRE DOMANDE

Puoi modificare il programma per porre più di una domanda, memorizzando le risposte in diverse variabili? Puoi realizzare un programma che utilizza condizionali e operatori di confronto per stampare se un numero digitato dall'utente è superiore o inferiore a 5, come il programma creato nel Capitolo 4, *Programmare con Scratch 3?*



Progetto 1: fiocchi di neve con turtle

Ora che hai capito come funziona Python, è il momento di giocare con la grafica e creare un fiocco di neve utilizzando uno strumento noto come *turtle* (tartaruga).

I turtle sono robot a forma di tartaruga, progettati per muoversi in linea retta, girare e sollevare e abbassare una penna. In parole semplici, un turtle inizierà o interromperà il tracciamento di una linea mentre si muove. A differenza di altri linguaggi, come Logo e le sue numerose varianti, Python non ha uno strumento turtle incorporato, ma viene fornito con una *libreria* di codice aggiuntivo che consente di operare similmente. Le librerie sono insiemi di codice che aggiungono nuove istruzioni per espandere le capacità di Python e puoi importarle nei tuoi programmi utilizzando un comando **import**.

Crea un nuovo programma facendo clic sull'icona **Nuovo**  e digita quanto segue:



```
import turtle
```


Quando si utilizzano le istruzioni incluse in una libreria è necessario utilizzare il nome della libreria seguito da un punto, quindi il nome dell'istruzione. Può essere fastidioso digitarlo ogni volta, quindi puoi assegnare un'istruzione a una variabile con un nome breve. Potrebbe essere composto da una sola lettera, ma abbiamo pensato che sarebbe carino se fosse anche un nomignolo per il turtle. Digita quanto segue:

```
pat = turtle.Turtle()
```

Per testare il programma, dovrai assegnare un compito al turtle. Digita:

```
pat.forward(100)
```

Fai clic sull'icona **Salva** , salva il programma come **Fiocchi di neve tartaruga.py**, quindi fai clic sull'icona  **Esegui**; viene visualizzata una nuova finestra chiamata "Turtle Graphics" che mostra il risultato del tuo programma, il tuo turtle, Pat, si muoverà in avanti di 100 unità, disegnando una linea retta (**Figura 5-8**).

Torna alla finestra principale Thonny e fai clic sul pulsante Stop per chiudere la finestra Turtle Graphics, qualora questa nasconda la finestra principale, fai clic sul pulsante minimizza nella finestra Turtle Graphics o fai clic sulla voce Thonny nella barra delle applicazioni nella parte superiore dello schermo. Dopo aver portato la finestra di Thonny in primo piano, fai clic su **Ferma**  per chiudere la finestra Turtle Graphics.

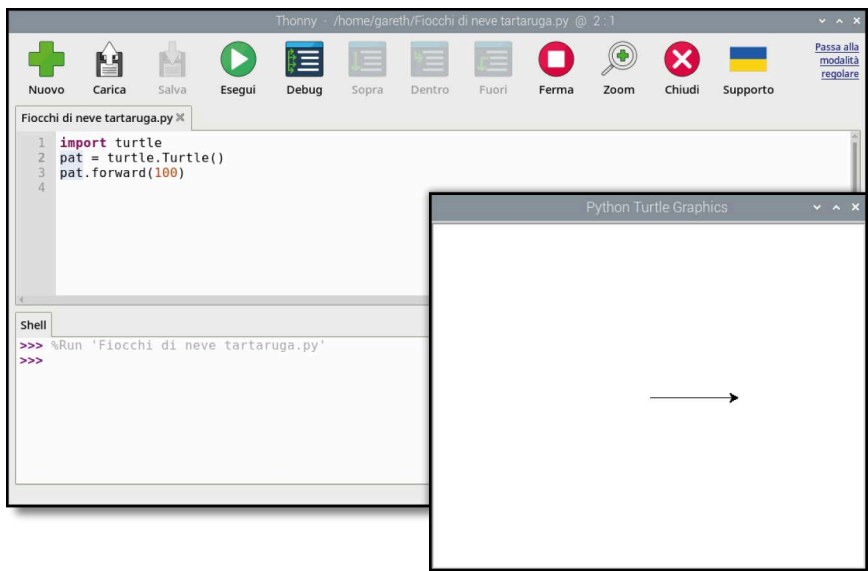



Figura 5-8 Il turtle si sposta in avanti per tracciare una linea retta

Per non dover digitare ogni singola istruzione di movimento per disegnare a mano qualcosa di più complesso, cancella la riga 3 e crea un ciclo per creare le forme:

```
for i in range(2):
    pat.forward(100)
    pat.right(60)
    pat.forward(100)
    pat.right(120)
```

Esegui il programma e Pat disegnerà un parallelogramma (**Figura 5-9**).

Per creare una forma simile a un fiocco di neve, fai clic su **Ferma**  nella finestra principale di Thonny e aggiungi un ciclo esterno utilizzando la seguente istruzione alla riga 3:

```
for i in range(10):
```

... e quanto segue alla fine del programma:

```
    pat.right(36)
```

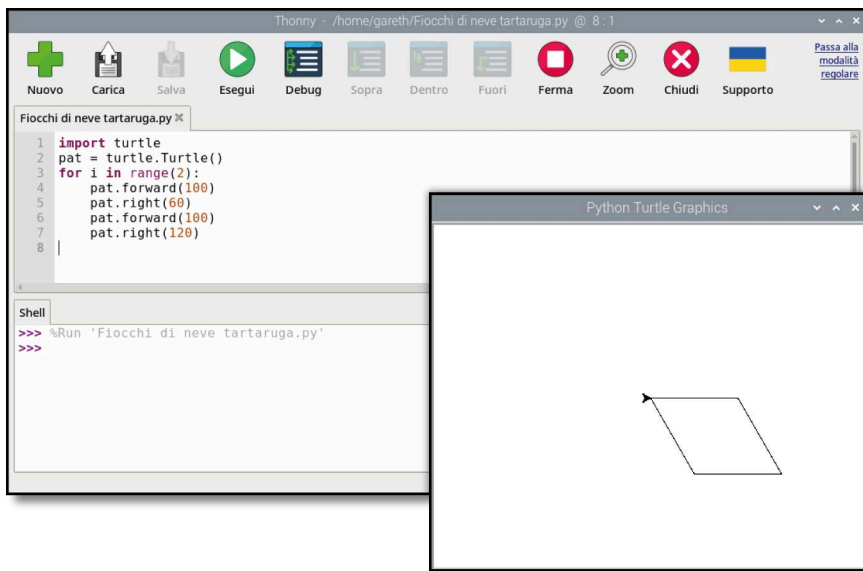



Figura 5-9 Combinando curve e movimenti, puoi disegnare forme

Il programma non funzionerà così com'è scritto, perché il ciclo esistente non presenta un'indentazione corretta. Per risolvere il problema, fai clic sull'inizio di ogni linea della ripetizione (linee da 4 a 8) e premi il tasto **SPAZIO** quattro volte per correggere l'indentazione. Il programma dovrebbe avere questa struttura:

```
import turtle
pat = turtle.Turtle()
for i in range(10):
    for i in range(2):
        pat.forward(100)
        pat.right(60)
        pat.forward(100)
        pat.right(120)
    pat.right(36)
```

Fai clic sull'icona **Esegui** , e guarda il turtle: disegnerà un parallelogramma, come prima, ma quando avrà finito girerà di 36 gradi e ne disegnerà un altro, poi un altro ancora e così via fino a quando ci saranno dieci parallelogrammi sovrapposti sullo schermo, uno schema per realizzare il fiocco di neve (vedi **Figura 5-10**).

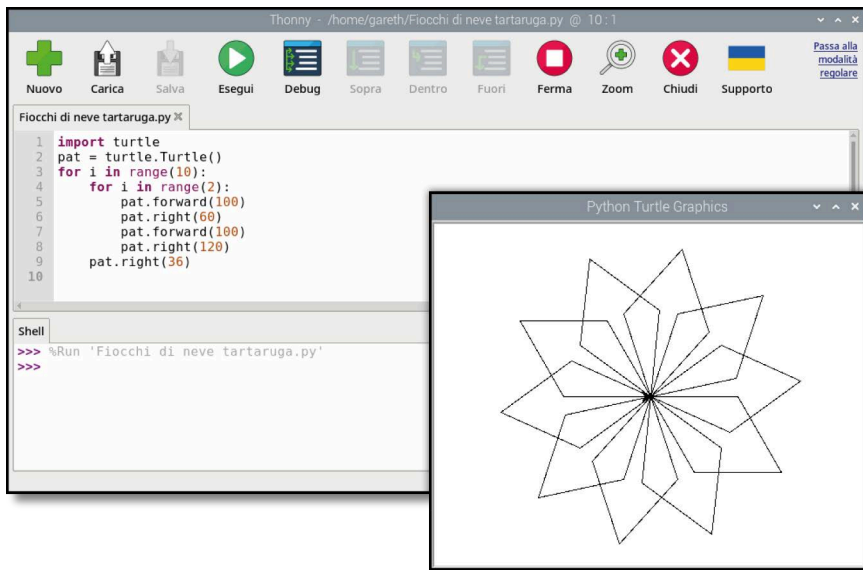


Figura 5-10 Ripetere la forma per realizzarne una più complessa

Mentre una tartaruga robot disegna in un unico colore su un grande pezzo di carta, la tartaruga simulata di Python può utilizzare vari colori. Aggiungi quanto segue come nuove linee 3 e 4, spostando in basso le linee esistenti:

```
turtle.Screen().bgcolor("blue")
pat.color("cyan")
```

Esegui di nuovo il programma e vedrai l'effetto del nuovo codice: il colore di sfondo della finestra Turtle Graphics è diventato blu, mentre il fiocco di neve ora è ciano (**Figura 5-11**).

Puoi anche scegliere i colori in modo casuale utilizzando la libreria **random**. Torna all'inizio del programma e inserisci quanto segue alla riga 2:

```
import random
```

Cambia il colore di sfondo in quella che ora è la riga 4 da blu a grigio, quindi crea una nuova variabile chiamata **colours** inserendo una nuova riga 5:

```
colours = ["cyan", "purple", "white", "blue"]
```

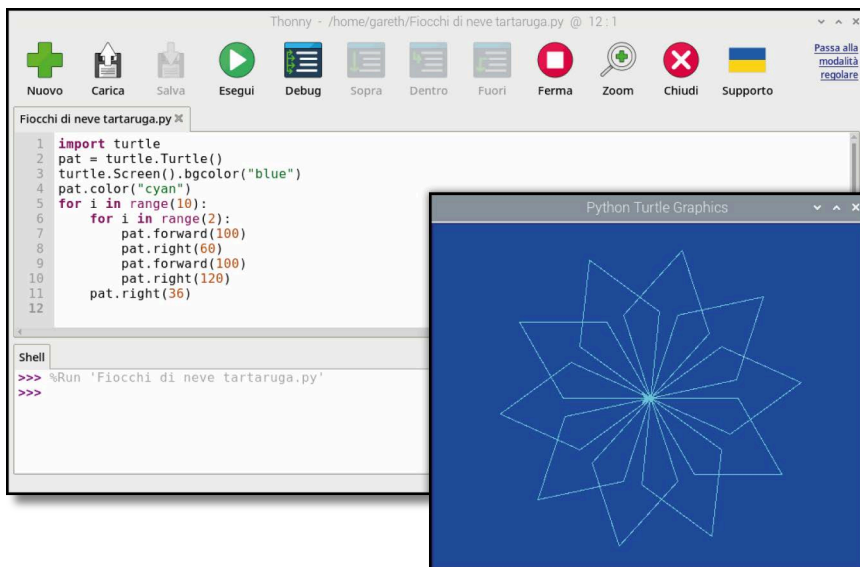


Figura 5-11 Modifica del colore di sfondo e del fiocco di neve


Questo tipo di variabile è chiamato *lista* ed è contrassegnato da parentesi quadre. In questo caso, la lista è piena di possibili colori per i segmenti del fiocco di neve, ma è comunque necessario indicare a Python di sceglierne uno ogni volta che il ciclo si ripete. Alla fine del programma inserisci quanto segue, assicurandoti che ogni riga abbia quattro spazi di indentazione in modo che faccia parte del ciclo esterno, proprio come la linea sopra:

```
pat.color(random.choice(colours))
```



INGLESE STATUNITENSE

Molti linguaggi di programmazione usano l'inglese americano e Python non fa eccezione: il comando per cambiare il colore della penna turtle si scrive **color** e se lo si scrive con l'ortografia inglese britannica, ossia **colour**, non funzionerà. Tuttavia, per le variabili, puoi usare l'ortografia che preferisci, ecco perché puoi chiamare la nuova variabile **colours** e fare in modo che Python la riconosca.

Fai di nuovo clic sull'icona **Esegui**  e verrà ancora disegnato il fiocco di neve. Questa volta però Python sceglierà casualmente un colore dalla lista mentre disegna ogni petalo, dando al fiocco di neve un aspetto multicolore, come mostrato in Figura 5-12.

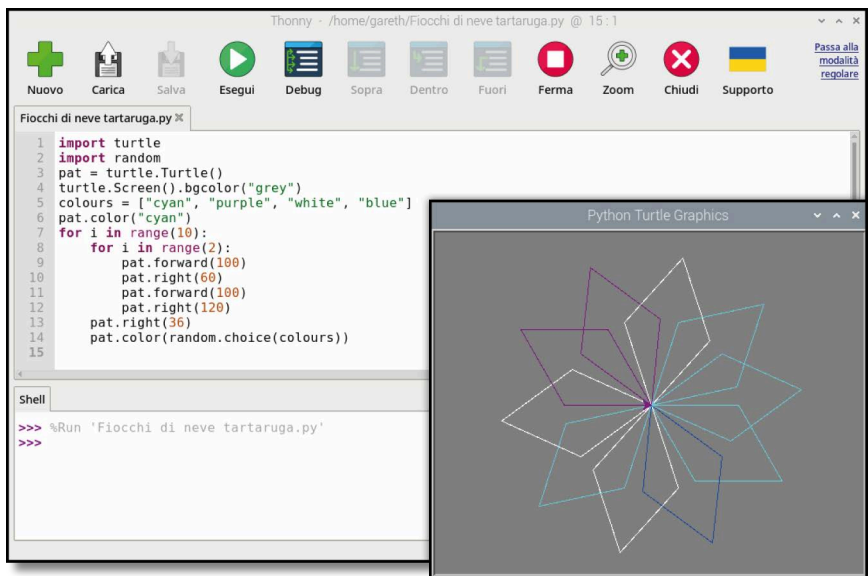


Figura 5-12 Utilizzo di colori casuali per le sezioni

Per far sembrare il fiocco di neve meno simile a una stella ninja e più simile a un vero e proprio fiocco di neve, aggiungi una nuova riga 6, direttamente sotto alla lista `colours` e digita quanto segue:

```
pat.penup()
pat.forward(90)
pat.left(45)
pat.pendown()
```

Le istruzioni `penup` e `pendown` sposterebbero una penna fisica su e giù dalla carta se utilizzate con un robot turtle, ma nel mondo virtuale servono semplicemente a indicare al tuo turtle quando fermarsi e quando iniziare a disegnare linee. Questa volta, invece di usare un ciclo, creerai una *funzione*, ossia un segmento di codice chiamabile in qualsiasi momento, creando, così, la tua personale istruzione Python.

Inizia cancellando il codice per disegnare i fiocchi di neve basati su parallelogrammi, ossia tutto ciò che è compreso tra l'istruzione `pat.color("cyan")` sulla linea 10 fino a `pat.right(36)` sulla linea 17 incluse. Lascia l'istruzione `pat.color(random.choice(colours))`, ma aggiungi il simbolo cancelletto (`#`) all'inizio della linea. Tale simbolo serve a *commentare fuori* una istruzione, facendo in modo che Python la ignori durante l'esecuzione. Puoi usare i

commenti per aggiungere spiegazioni al codice, così sarà molto più facile da capire quando tornerai a lavorarci dopo un po' di tempo o lo invierai a qualcun altro.

Crea la funzione, che chiameremo **branch**, digitando la seguente istruzione sulla riga 10 sotto a `pat.pendown()`:

```
def branch():
```

Questo *definisce* la funzione dandole un nome, **branch**. Quando premi il tasto INVIO Thonny aggiungerà automaticamente l'indentazione per le istruzioni della funzione. Digita quanto segue facendo attenzione alla rientranza, perché a un certo punto dovrai annidare del codice aggiungendo tre spazi di indentazione.

```
    for i in range(3):
        for i in range(3):
            pat.forward(30)
            pat.backward(30)
            pat.right(45)
        pat.left(90)
        pat.backward(30)
        pat.left(45)
    pat.right(90)
    pat.forward(90)
```

Infine, crea un nuovo ciclo in fondo al programma, ma sopra la linea commentata **color**, per eseguire, o, *chiamare* la tua nuova funzione:

```
for i in range(8):
    branch()
    pat.left(45)
```

Il programma finito dovrebbe avere questa struttura:

```
import turtle
import random

pat = turtle.Turtle()
turtle.Screen().bgcolor("grey")
colours = ["cyan", "purple", "white", "blue"]

pat.penup()
pat.forward(90)
pat.left(45)
```

```

pat.pendown()

def branch():
    for i in range(3):
        for i in range(3):
            pat.forward(30)
            pat.backward(30)
            pat.right(45)
        pat.left(90)
        pat.backward(30)
        pat.left(45)
    pat.right(90)
    pat.forward(90)

for i in range(8):
    branch()
    pat.left(45)
# pat.color(random.choice(colours))

```

Fai clic su **Esegui** e guarda la finestra grafica mentre Pat crea il disegno seguendo le tue istruzioni. Complimenti, adesso sembra un vero un fiocco di neve (Figura 5-13)!

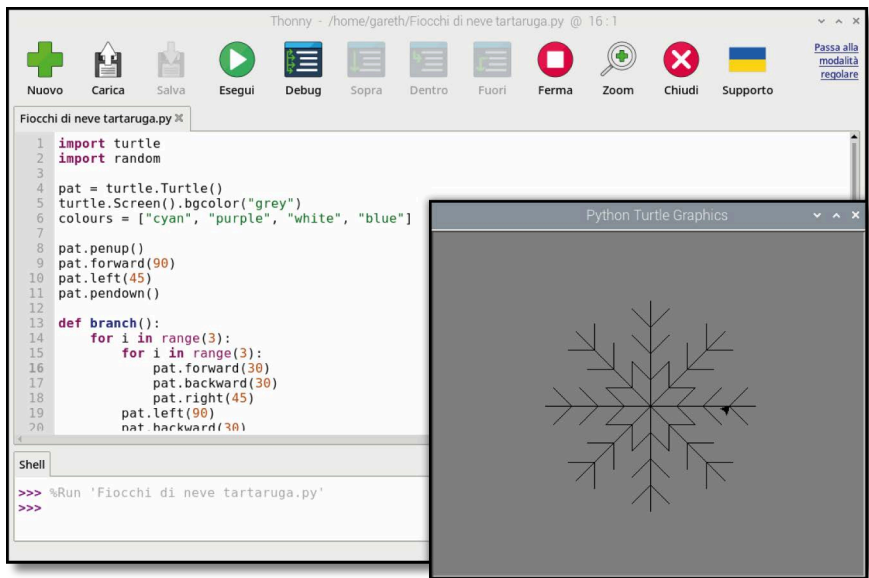


Figura 5-13 Le linee fanno sembrare la figura un fiocco di neve



SFIDA: CHE SI FA ADESSO


Riesci a utilizzare le istruzioni commentate per far disegnare le linee del fiocco di neve in diversi colori? Sei in grado di creare una funzione "fiocco di neve" e di utilizzarla per disegnare molti fiocchi di neve sullo schermo? Riesci a far cambiare al programma la dimensione e il colore dei fiocchi di neve in modo casuale?

Progetto 2: uno spaventoso "Trova le differenze"

Python può gestire immagini e suoni così come grafica basata su turtle, e questi possono essere utilizzati con grande effetto per fare uno scherzo ai tuoi amici: un gioco in cui bisogna trovare le differenze con uno spaventoso segreto al suo interno, perfetto per Halloween!

Per questo progetto sono necessari un file audio e due immagini: una per trovare le differenze e una "spaventosa" immagine a sorpresa. Fai clic sull'icona di Raspberry Pi per accedere al menu Raspberry Pi, scegli la categoria **Internet** e fai clic su **Browser web Chromium**. Una volta aperto, digita **rptl.io/spot-pic** nella barra degli indirizzi e premi il tasto **INVIO**. Fai clic con il tasto destro del mouse sull'immagine, fai clic su **Salva immagine con nome**, seleziona la cartella **Home** dall'elenco a sinistra, quindi fai clic su **Salva**. Fai clic nuovamente sulla barra degli indirizzi di Chromium, quindi digita **rptl.io/scary-pic** e premi il tasto **INVIO**. Come prima, fai clic con il tasto destro del mouse sull'immagine, fai clic su **Salva immagine con nome**, scegli la cartella **Home** e fai clic su **Salva**.

Per scaricare il file audio, fai nuovamente clic sulla barra degli indirizzi, digita **rptl.io/scream** e premi il tasto **INVIO**. Questo file contiene il suono di un urlo, per spaventare il lettore, e verrà scaricato automaticamente. Per poterlo utilizzare, però, dovrai spostarlo nella cartella **Home**. Apri una nuova finestra di Gestore file, quindi vai alla cartella **Downloads** e cerca il file appena scaricato. Nella finestra Gestore file visualizzata, rinominare il file in **scream.wav**. Fai clic con il tasto destro del mouse sul file **scream.wav**, quindi fai clic su **Taglia**. Infine, fai clic su **Cartella Home** in alto a sinistra in Gestore file, fai clic con il tasto destro del mouse su qualsiasi spazio vuoto nella finestra di visualizzazione dei file di grandi dimensioni sulla destra e fai clic su **Incolla**. Ora puoi chiudere la finestra di Chromium e quella di Gestore file.

Fai clic sull'icona **Nuovo**  nella barra degli strumenti di Thonny per iniziare un nuovo progetto. Come già fatto in precedenza, dovrai utilizzare una libreria per ampliare le funzionalità di Python. Questa volta si tratta della libreria **pygame** che, come suggerisce il nome, è stata creata pensando ai giochi. Digita quanto segue:

```
import pygame
```

Necessiterai anche di parti di altre librerie e di una sottosezione della libreria Pygame. Importale digitando quanto segue:



```
from pygame.locals import *
from time import sleep
from random import randrange
```

L'istruzione **from** è diversa da **import**, permettendoti di importare solo le parti di libreria che ti occorrono, anziché una libreria intera. Successivamente dovrai configurare Pygame, questo processo è noto come *inizializzazione*. Pygame deve conoscere la larghezza e l'altezza del monitor o del televisore del giocatore, ossia la *risoluzione*. Digita quanto segue:

```
pygame.init()
width = pygame.display.Info().current_w
height = pygame.display.Info().current_h
```

L'ultimo passaggio per configurare Pygame è quello di creare la sua finestra, ossia lo "screen" di Pygame. Digita quanto segue:

```
screen = pygame.display.set_mode((width, height))
pygame.display.update()
# Scrivi il tuo programma qui
pygame.quit()
```

Il tuo programma andrà nella riga con commento al centro. Per ora, però, fai clic sull'icona **Salva** , salva il tuo programma con il nome **Trova le differenze.py** e fai clic sull'icona **Esegui**  e guarda. Pygame creerà una finestra, la riempirà con uno sfondo nero e la chiuderà quasi immediatamente non appena riceverà l'istruzione di uscire. A parte un breve messaggio nell'area shell (Figura 5-14), il programma non fa molto fino ad ora.

Per visualizzare l'immagine con le differenze, cancella il commento sopra **pygame.quit()** e digita quanto segue nello spazio:

```
difference = pygame.image.load('spot_the_diff.png')
```

Per assicurarti che l'immagine riempi lo schermo, dovrai ridimensionarla in modo che corrisponda alla risoluzione del monitor o della TV. Digita quanto segue:

```
difference = pygame.transform.scale(difference, (width, height))
```

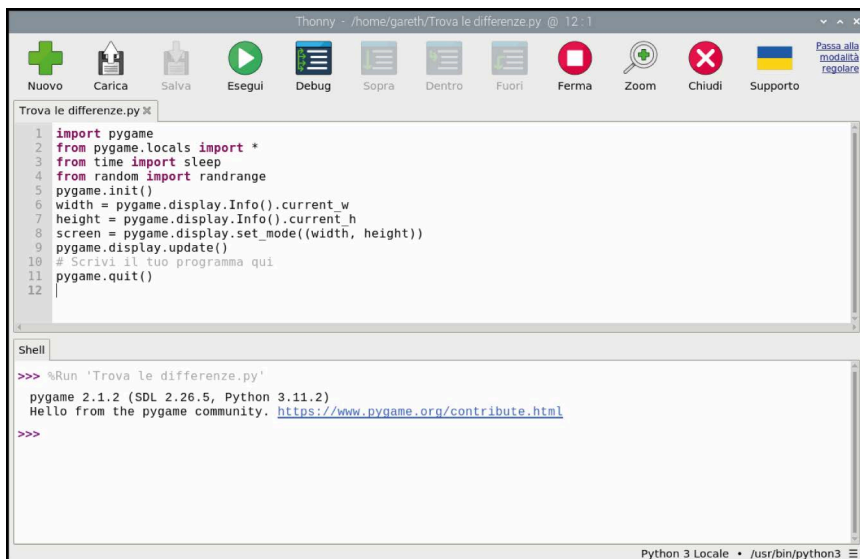



Figura 5-14 Il programma è funzionale, ma non fa ancora molto

Ora che l'immagine è stata memorizzata, è necessario dire a Pygame di mostrarla sullo schermo, un processo noto come *blitting* o *trasferimento a blocchi di bit*. Digita quanto segue:

```
screen.blit(difference, (0, 0))
pygame.display.update()
```

La prima di queste righe copia l'immagine sullo schermo, partendo dall'angolo in alto a sinistra, mentre la seconda indica a Pygame di ridisegnare lo schermo. Senza questa seconda riga, l'immagine sarà memorizzata correttamente, ma non la vedrai.


Fai clic sull'icona **Esegui**  e l'immagine, mostrata in **Figura 5-15**, comparirà brevemente sullo schermo.

Perché l'immagine rimanga visualizzata per un periodo di tempo più lungo, aggiungi la seguente riga appena sopra a **pygame.quit()**:

```
sleep(3)
```



Figura 5-15 La tua immagine per il gioco "Trova le differenze"

Fai di nuovo clic sull'icona **Esegui**  e l'immagine rimarrà sullo schermo più a lungo. Aggiungi l'immagine a sorpresa digitando quanto segue appena sotto la riga `pygame.display.update()`:


```
zombie = pygame.image.load('scary_face.png')
zombie = pygame.transform.scale(zombie, (width, height))
```

Aggiungi un ritardo, in modo che l'immagine dello zombie non appaia subito:

```
sleep(3)
```

Esegui il blitting dell'immagine sullo schermo e aggiornala in modo da mostrarla al giocatore:

```
screen.blit(zombie, (0,0))
pygame.display.update()
```

Fai clic sull'icona **Esegui**  e guarda cosa succede: Pygame caricherà l'immagine per il gioco trova le differenze, ma dopo tre secondi sarà sostituita da uno spaventoso zombie (Figura 5-16)!

Il ritardo di tre secondi rende il tutto abbastanza prevedibile. Modifica la riga `sleep(3)` sopra a `screen.blit(zombie, (0,0))` in:

```
sleep(randrange(5, 15))
```



Figura 5-16 Uno scherzetto spaventoso

Verrà scelto un numero casuale tra 5 e 15, che corrisponderà al tempo di ritardo. Successivamente aggiungi la seguente riga appena sopra all'istruzione `sleep` per caricare il file del suono dell'urlo:


```
scream = pygame.mixer.Sound('scream.wav')
```

Digita quanto segue in una nuova riga dopo l'istruzione `sleep` per avviare la riproduzione del suono. Dovrebbe partire poco prima che l'immagine spaventosa venga effettivamente mostrata al giocatore:

```
scream.play()
```

Infine indica a Pygame di smettere di riprodurre il suono digitando la seguente riga appena sopra a `pygame.quit()`:

```
scream.stop()
```

Fai clic sull'icona **Esegui**  e ammira il tuo lavoro: dopo qualche secondo di innocente divertimento con il gioco per trovare le differenze, il tuo terribile zombie apparirà insieme a un urlo agghiacciante che sicuramente spaventerà i tuoi amici. Se l'immagine dello zombie appare prima che il suono venga riprodotto, puoi aggiungere un piccolo ritardo subito dopo l'istruzione `scream.play()` e prima di `screen.blit`:

```
sleep(0.4)
```

Il programma finito dovrebbe avere questa struttura:

```
import pygame
from pygame.locals import *
from time import sleep
from random import randrange

pygame.init()
width = pygame.display.Info().current_w
height = pygame.display.Info().current_h
screen = pygame.display.set_mode((width, height))
pygame.display.update()

difference = pygame.image.load('spot_the_diff.png')
difference = pygame.transform.scale(difference, (width, height))
screen.blit(difference, (0, 0))
pygame.display.update()

zombie = pygame.image.load('scary_face.png')
zombie = pygame.transform.scale (zombie, (width, height))
scream = pygame.mixer.Sound('scream.wav')
sleep(randrange(5, 15))
scream.play()
screen.blit(zombie, (0,0))
pygame.display.update()

sleep(3)
scream.stop()
pygame.quit()
```

Ora non ti resta che invitare i tuoi amici a giocare a questo gioco e ovviamente assicurarti che il volume degli altoparlanti sia attivo.

SFIDA: CAMBIA L'ASPETTO


Riesci a cambiare le immagini per rendere lo scherzo più appropriato per altri eventi, come il Natale? Sei in grado di disegnare le immagini per il gioco trova le differenze e quelle con cui spaventare i tuoi amici usando un software grafico come GIMP? Riesci a tenere traccia delle azioni dell'utente quando fa clic su una differenza, per rendere il tutto più convincente?




Progetto 3: Avventura testuale


Ora che stai prendendo la mano con Python, è il momento di usare Pygame per realizzare qualcosa di un po' più complicato: un gioco labirinto basato su testo, completamente funzionante, ispirato ai classici giochi di avventura testuale. Noti come avventure testuali o narrativa interattiva, questi giochi risalgono a quando i computer non erano in grado di gestire la grafica e, ancora oggi, hanno appassionati che sostengono che nessuna grafica sarà mai vivida quanto l'immaginazione.

Questo programma è un po' più complesso degli altri illustrati in questo capitolo. Per semplificare le cose, inizierai con una versione parzialmente scritta. Apri il browser Web Chromium e accedi a <http://rptl.io/text-adventure-it>.

Chromium caricherà il codice del programma nel browser. Fai clic on il tasto destro del mouse sulla pagina del browser, seleziona **Salva con nome** e salva il file come **text-adventure.py** nella tua cartella Download. Potresti ricevere un messaggio che ti avverte che questo tipo di file (un programma Python) potrebbe danneggiare il tuo computer. Poiché il file è stato scaricato da una fonte affidabile, puoi fare tranquillamente clic sul pulsante **Conserva** se visualizzi questo messaggio di avviso nella parte inferiore dello schermo. Torna in su Thonny e fai clic sull'icona **Carica** . Individua il file, **text-adventure.py**, nella cartella **Downloads** e fai clic sul pulsante **Carica**.

Inizia facendo clic sull'icona **Esegui**  per prendere familiarità con il funzionamento di un'avventura testuale. L'output del gioco viene visualizzato nell'area shell nella parte inferiore della finestra di Thonny. Se necessario, puoi ingrandire la finestra di Thonny facendo clic sul pulsante di ingrandimento per facilitare la lettura.

Il gioco, così com'è ora, è molto semplice: ci sono due stanze e nessun oggetto. Il giocatore inizia nell'**Ingresso**, la prima delle due stanze. Per entrare in **Cucina**, digita semplicemente **"vai sud"** e premi il tasto **INVIO** (Figura 5-17). Quando sei in **Cucina**, puoi digitare **"vai nord"** per tornare nell'**Ingresso**. Puoi anche provare a digitare **"vai ovest"** e **"vai est"**, ma dato che non ci sono stanze in quelle direzioni il gioco ti mostrerà un messaggio di errore.

Premi l'icona **Ferma**  per interrompere il programma, poi scorri fino alla riga 30 del programma nell'area script per trovare una variabile chiamata **rooms**. Questo tipo di variabile è noto come *dizionario* e indica al gioco le stanze, le loro uscite e a quale stanza porta una determinata uscita.

Per rendere il gioco più interessante, aggiungiamo un'altra stanza: la **Sala da pranzo** a est dell'**Ingresso**.



Figura 5-17 Finora ci sono solo due stanze

Trova la variabile `rooms` nell'area degli script ed estendila aggiungendo un simbolo di virgola (,) dopo (,) sulla riga 38, digitando poi quanto segue:

```

'Sala da pranzo' : {
    'ovest' : 'Ingresso'
}


```


Avrai anche bisogno di una nuova uscita nell'`Ingresso`, in quanto non viene creata automaticamente. Vai alla fine della riga 33, aggiungi una virgola, quindi aggiungi la riga seguente:

```

'est' : 'Sala da pranzo'

```

Fai clic sull'icona **Esegui**  e visita la nuova stanza: digita "`vai est`" mentre ti trovi nell'`Ingresso` per entrare nella `Sala da pranzo` (Figura 5-18) e digita "`vai ovest`" mentre ti trovi nella `Sala da pranzo` e vuoi entrare di nuovo nell'`Ingresso`. Complimenti, hai creato la tua stanza!

Le stanze vuote non sono molto divertenti. Per aggiungere un elemento a una stanza, dovrai modificare il dizionario della stanza. Fai clic sull'icona **Ferma**  per interrompere il programma. Trova il dizionario `Ingresso` nell'area degli

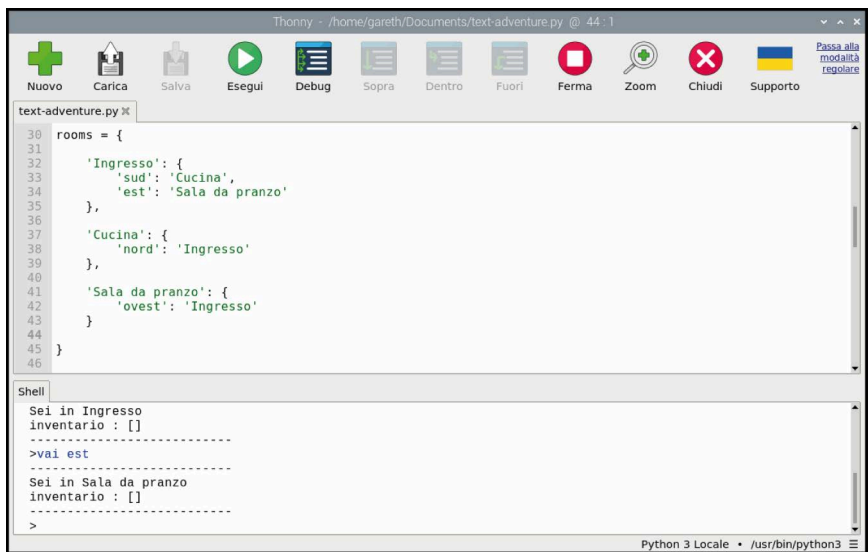


Figura 5-18 Hai aggiunto un'altra stanza

script, aggiungi una virgola alla fine della riga `'east' : 'Sala da pranzo'`, premi INVIO e digita questa riga:

```
'oggetto' : 'chiave'
```

Fai nuovamente clic sull'icona **Esegui**. Questa volta il gioco ti dirà che puoi vedere il nuovo oggetto: una chiave. Digita `"prendi chiave"` (Figura 5-19) per raccogliarlo e aggiungerlo all'elenco di oggetti che hai raccolto, chiamato anche *inventario*. L'inventario rimane con te mentre passi da una stanza all'altra.

Fai clic sull'icona **Ferma** e rendi il gioco più interessante aggiungendo un mostro da evitare. Trova la voce del dizionario `Cucina` e aggiungi l'oggetto `"mostro"` nello stesso modo in cui hai aggiunto l'oggetto `"chiave"`, ricordandoti di aggiungere una virgola alla fine della riga precedente:

```
'oggetto' : 'mostro'
```

Devi aggiungere una logica per consentire al mostro di attaccare il giocatore. Scorri fino alla fine del programma nell'area dello script e aggiungi le seguenti righe, compreso il commento, contrassegnato da un cancelletto, che ti aiuterà

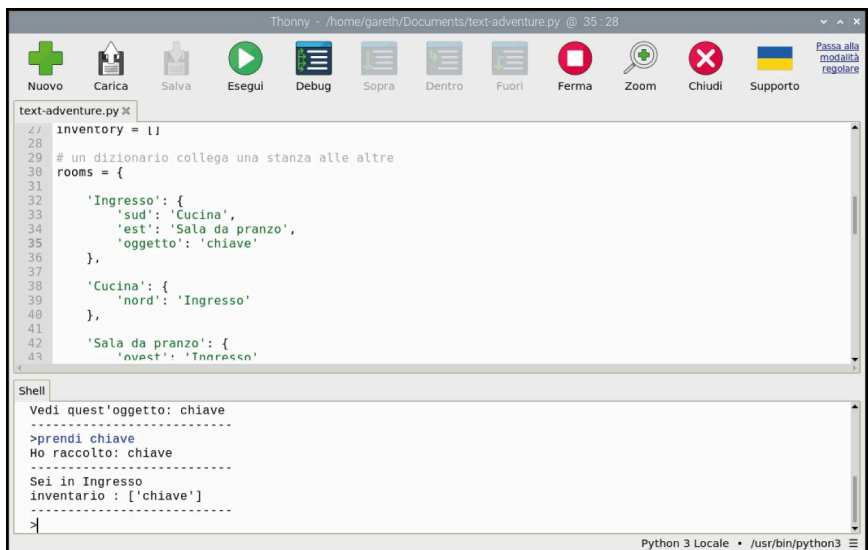



Figura 5-19 La chiave raccolta viene aggiunta all'inventario

a capire il programma se ci lavorerai un altro giorno. Assicurati di far rientrare le righe e di digitare tutto tra **if** e i due punti (:) su una riga:

```
# Il giocatore perde se c'è un mostro qui
if 'oggetto' in rooms[currentRoom]
    and 'mostro' in rooms[currentRoom]['oggetto']:
    print('Une creatura mostruosa ti ha catturato... HAI PERSO!')
    break
```

Fai clic sull'icona **Esegui**  e prova ad andare in cucina (**Figura 5-20**), al mostro non sembrerà interessare molto.

Per trasformare quest'avventura in un vero e proprio gioco avrai bisogno di più oggetti, di un'altra stanza e della possibilità di vincere lasciando la casa con tutti gli oggetti al sicuro nel tuo inventario. Inizia aggiungendo un'altra stanza come hai fatto prima con la **Sala da pranzo**, ma questa volta si tratta di un **Giardino**. Aggiungi un'uscita dal dizionario **Sala da pranzo**, ricordandoti di aggiungere una virgola alla fine della riga precedente:

```
'sud' : 'Giardino'
```

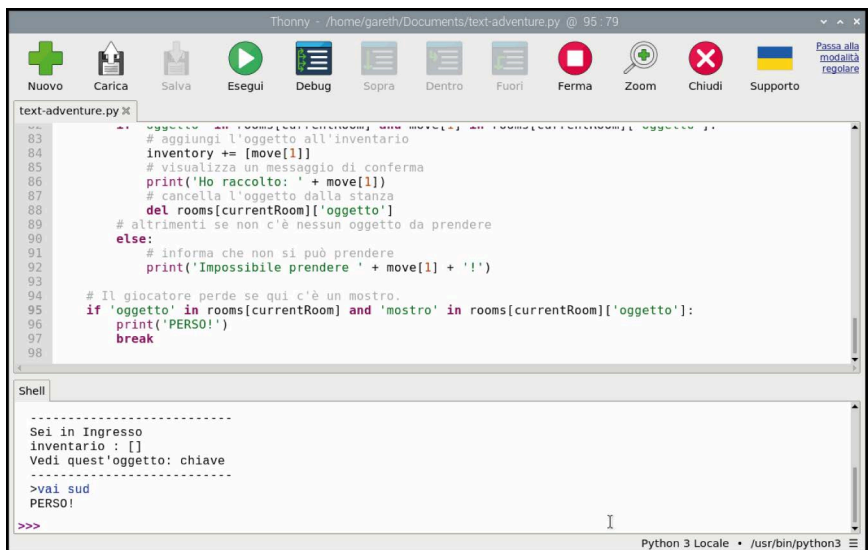


Figura 5-20 Che paura fa un topo quando c'è un mostro in cucina?!

Quindi aggiungi la tua nuova stanza al dizionario `rooms` principale, ricordandoti sempre di aggiungere una virgola dopo `}` sulla riga precedente:

```

'Giardino' : {
    'nord' : 'Sala da pranzo'
}

```

Aggiungi un oggetto pozione al dizionario `Sala da pranzo`, ricordandoti di aggiungere una virgola nella riga precedente:

```

'oggetto' : 'pozione'


```

Infine, scorri fino in fondo al programma e aggiungi la logica necessaria per controllare se il giocatore ha tutti gli oggetti e, in caso affermativo, digli che ha vinto la partita (assicurati di far rientrare le righe e di digitare tutto tra `if` e i due punti `:` su una riga):

```

# Scappa con la chiave e la pozione per vincere
if currentRoom == 'Giardino' and 'chiave' in inventory
    and 'pozione' in inventory:
    print('Sei scappato dalla casa... HAI VINTO!')
    break

```

Fai clic sull'icona **Esegui**  e cerca di finire il gioco raccogliendo la chiave e la pozione prima di andare in giardino. Ricordati di non entrare in **Cucina**, perché è lì che si trova il mostro!

Come ultima modifica per il gioco, aggiungi alcune istruzioni che spiegano al giocatore come completare il gioco. Scorri fino all'inizio del programma, dove è definita la funzione **showInstructions()**, e aggiungi quanto segue:

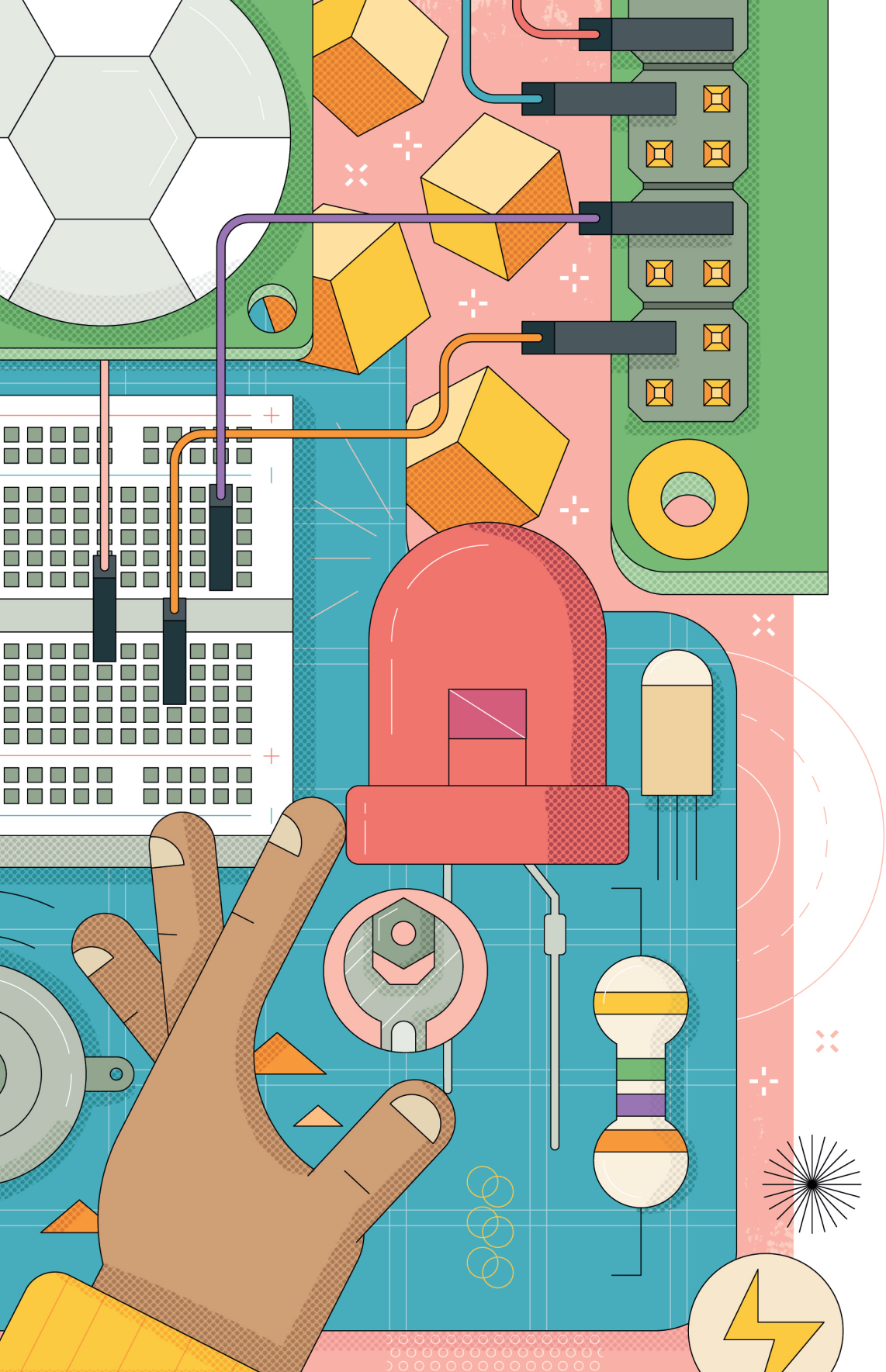
Arriva al giardino con una chiave e una pozione
Evita i mostri!

Esegui il gioco un'ultima volta e visualizzerai le nuove istruzioni all'inizio. Complimenti, hai realizzato un gioco labirinto interattivo basato su testo!

SFIDA: ESPANDI IL GIOCO

Riesci ad aggiungere altre stanze per far durare il gioco più a lungo? Sei in grado di aggiungere un oggetto per proteggerti dal mostro? Come aggiungeresti un'arma per uccidere il mostro? Riesci ad aggiungere stanze sopra e sotto le camere esistenti a cui accedere tramite scale?





Capitolo 6

Physical computing con Scratch e Python

Programmare non vuol dire solo eseguire azioni su uno schermo, ma anche controllare i componenti elettronici collegati ai pin GPIO del tuo Raspberry Pi

Quando si parla di "programmazione" o "scrittura di codice" generalmente si pensa al software. Tuttavia la programmazione va oltre il software: essa può essere usata anche nel mondo reale per gestire hardware. In questo caso viene chiamata *physical computing*.

Come suggerisce il nome, il physical computing (letteralmente: computazione fisica), riguarda il controllo di cose nel mondo reale tramite i propri programmi, utilizzando l'hardware insieme al software. Quando imposti il programma della tua lavatrice, cambi la temperatura del termostato o premi un pulsante al semaforo per attraversare la strada in sicurezza, stai utilizzando il physical computing.

Raspberry Pi è un ottimo strumento per imparare a conoscere il physical computing grazie a una funzione chiave: il connettore *general-purpose input/output* (GPIO).

Il connettore GPIO

Sul bordo superiore del circuito stampato del Raspberry Pi, o sul retro del Raspberry Pi 400, troverai due file di pin metallici. Si tratta del connettore GPIO (general-purpose input/output), posizionato lì per permetterti di collegare hardware come LED e interruttori al Raspberry Pi e controllarli con i programmi che crei. I pin possono essere utilizzati sia in ingresso che in uscita.

Il connettore GPIO del Raspberry Pi presenta 40 pin maschi, come mostrato nella **Figura 6-1**. Alcuni pin possono essere usati nei progetti di physical computing, altri forniscono l'alimentazione, mentre altri ancora sono dedicati alla comunicazione con l'hardware aggiuntivo come Sense HAT (vedi Capitolo 7, *Physical computing con Sense HAT*).

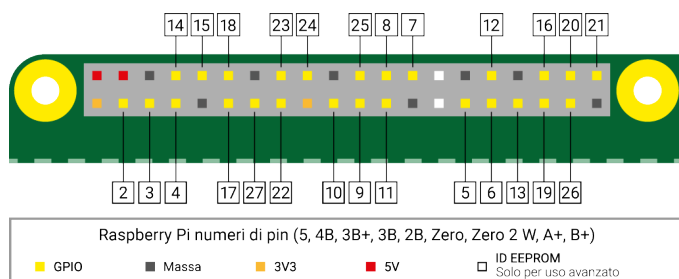


Figura 6-1 Piedinatura GPIO di Raspberry Pi

Il Raspberry Pi 400 presenta lo stesso connettore GPIO e gli stessi pin, ma è capovolto rispetto agli altri modelli di Raspberry Pi. Nella **Figura 6-2** si presuppone che tu stia guardando il connettore GPIO dalla parte posteriore del Raspberry Pi 400. Controlla sempre il cablaggio quando colleghi qualcosa al connettore GPIO del Raspberry Pi 400: è facile dimenticarsi, nonostante le etichette "Pin 40" e "Pin 1" sul case.

Anche il Raspberry Pi Zero 2 W è predisposto per un connettore GPIO, ma quest'ultimo non è saldato. Se vuoi utilizzare un Raspberry Pi Zero 2 W, o un altro modello della gamma Raspberry Pi Zero, per il physical computing dovrai *saldare* i pin mediante un saldatore. Se ti sembra un po' rischioso, rivolgiti a un rivenditore autorizzato di Raspberry Pi per trovare un Raspberry Pi Zero 2 W con i pin terminale già saldati in posizione.



ESTENSIONI GPIO

È possibile utilizzare il terminale GPIO del Raspberry Pi 400 così com'è, ma potrebbe essere più facile usare un'estensione. Con un'estensione, i pin vengono portati sul lato del Raspberry Pi 400, il che significa che è possibile controllare e modificare il cablaggio senza dover andare sulla parte posteriore.

Le estensioni compatibili includono la gamma Black HAT Hack3r di **pimoroni.com** e Pi T-Cobbler Plus di **adafruit.com**.

Se acquisti un'estensione, controlla sempre come è cablata. Alcune, come Pi T-Cobbler Plus, modificano la disposizione dei pin GPIO. In caso di dubbi, fai sempre riferimento alle istruzioni del produttore piuttosto che agli schemi dei pin riportati in questa guida.

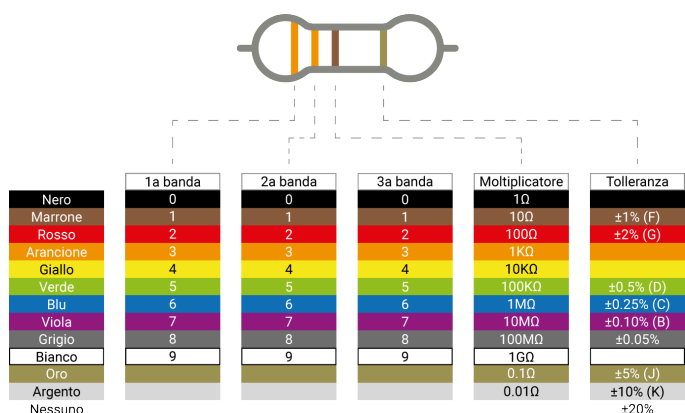


Figura 6-2 Piedinatura GPIO del Raspberry Pi 400

Esistono diverse tipologie di pin, ognuna delle quali ha una funzione particolare:

3V3	Tensione 3,3 volt	Una fonte di alimentazione permanentemente accesa da 3,3 V, la stessa tensione interna di Raspberry Pi
5V5	Tensione 5 volt	Una fonte di alimentazione permanentemente accesa da 5 V, la stessa tensione di Raspberry Pi a livello del connettore USB C
Massa (GND)	Massa a 0 volt	Un collegamento a massa, utilizzato per completare un circuito collegato alla fonte di alimentazione
GPIO XX	Pin di ingresso/uscita per uso generale numero "XX"	I pin GPIO disponibili per i programmi, identificati con un numero da 2 a 27
ID EEPROM	Pin riservati per usi speciali	Pin riservati all'uso con Hardware Attached on Top (HAT) e altri accessori



ATTENZIONE

Il connettore GPIO di Raspberry Pi è un modo divertente e sicuro per sperimentare con il physical computing, ma deve essere trattato con cura. Fai attenzione a non piegare i pin quando colleghi e scollegi l'hardware. Non collegare mai due pin direttamente tra loro, accidentalmente o deliberatamente, a meno che non sia espressamente indicato nelle istruzioni di un progetto. Se lo fai, creerai un cortocircuito e, a seconda dei pin, potresti danneggiare in modo permanente il tuo Raspberry Pi.

Componenti elettronici

Il connettore GPIO è solo una parte di ciò che ti serve per iniziare a lavorare con il physical computing. Avrai anche bisogno di alcuni componenti elettrici, i dispositivi che controllerai dal connettore GPIO. Sono migliaia i componenti disponibili, ma la maggior parte dei progetti GPIO viene realizzata con quelli indicati di seguito.

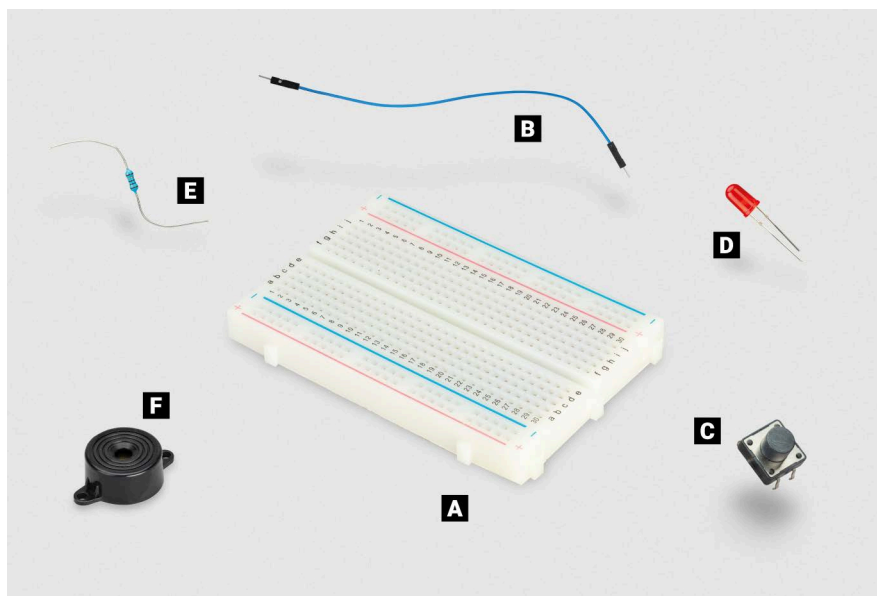


Figura 6-3 Componenti elettronici più comuni

- | | |
|----------------------------------|--|
| A Breadboard | D Diodo a emissione di luce (LED) |
| B Cavo jumper | E Resistore |
| C Interruttore momentaneo | F Cicalino piezoelettrico |

Una *breadboard* (**A**), nota anche come *breadboard senza saldature*, può rendere i progetti di physical computing molto più semplici. Aniché avere un gruppo di componenti separati da collegare con fili, una breadboard consente di inserire i componenti e collegarli attraverso le guide metalliche nascoste sotto la sua superficie. Molte breadboard includono anche sezioni per distribuire l'alimentazione, rendendo più facile realizzare circuiti. Non è necessario avere una breadboard per iniziare a lavorare con il physical computing, ma sicuramente aiuta.

I *cavi jumper* (**B**) collegano i componenti al Raspberry Pi o tra loro, se non usi una breadboard. Sono disponibili in tre versioni: maschio-femmina (M2F), per collegare una breadboard ai pin GPIO, femmina-femmina (F2F), che può essere utilizzata per collegare insieme i singoli componenti se non si utilizza una breadboard, e maschio-maschio (M2M) per effettuare collegamenti da una parte all'altra della breadboard. A seconda del progetto, potresti aver bisogno di tutti e tre i tipi di cavi jumper, mentre se stai usando una breadboard, possono bastare i cavi jumper M2F e M2M.

Un *interruttore momentaneo* (**C**) è un tipo di interruttore che potresti trovare sui controller di una console di gioco. Generalmente disponibile con due o quattro connettori (entrambi compatibili con Raspberry Pi), il pulsante è un dispositivo di input: puoi indicare al programma di attendere che venga premuto per eseguire un compito. Un altro tipo di interruttore comune è l'*interruttore a scatto*: mentre il pulsante è attivo solo quando lo si tiene premuto, l'interruttore a scatto (come ad esempio l'interruttore della luce) si attiva quando lo si aziona e rimane attivo fino a quando non lo si aziona di nuovo.

Un *diodo a emissione di luce* (LED, **D**) è un *dispositivo di output* che puoi controllare direttamente dal programma. Un LED si illumina quando è alimentato e puoi trovarne in tutta la casa: da quelli piccoli che ti fanno sapere quando hai lasciato la lavatrice accesa, a quelli più grandi che illuminano le stanze. I LED possono avere varie forme, colori e dimensioni, ma non tutti possono essere usati con Raspberry Pi: non utilizzare quelli indicati per alimentazione a 5 V o 12 V.

I *resistori* (**E**) sono componenti che controllano il flusso di *corrente elettrica* e possono presentare diversi valori misurati utilizzando un'unità chiamata *ohm* (Ω). Più elevato è il numero di ohm, maggiore è la resistenza. Nei progetti di physical computing con Raspberry Pi il loro uso più comune è quello di impedire ai LED di assorbire troppa corrente e danneggiarsi o danneggiare il Raspberry Pi: per questo ti consigliamo resistori da circa 330 Ω , anche se molti fornitori vendono pratiche confezioni contenenti resistori con diversi valori di resistenza per dare maggiore flessibilità.

Il *cicalino piezoelettrico* (**F**), di solito chiamato semplicemente cicalino o buzzer, è un altro dispositivo di output. Mentre un LED produce luce, un cicalino produce un rumore, un ronzio. L'alloggiamento in plastica del cicalino contiene

un paio di piastre metalliche. Quando sono attive, queste piastre vibrano l'una contro l'altra producendo un ronzio. Ci sono due tipi di cicalino: *attivo* e *passivo*. Accertati di procurarti un cicalino attivo, in quanto è più semplice da usare.

Altri componenti elettrici comuni sono i motori, che necessitano di una speciale scheda di controllo prima di poter essere collegati a un Raspberry Pi, i sensori a infrarossi, che rilevano il movimento, i sensori di temperatura e umidità, che possono essere utilizzati per prevedere le condizioni atmosferiche e infine i fotoresistori (LDR), ossia dispositivi di ingresso che funzionano al contrario dei LED rilevando la luce.

Rivenditori di tutto il mondo forniscono componenti per il physical computing con Raspberry Pi, sia come parti singole che in kit con tutto il necessario per iniziare. Per trovare i rivenditori, visita il sito rptl.io/products, fai clic su **Raspberry Pi 5** e quindi sul pulsante **Buy now** per vedere l'elenco dei negozi online partner di Raspberry Pi e dei rivenditori approvati per il tuo Paese o la tua regione.

Per completare i progetti di questo capitolo, dovresti disporre di almeno:

- ▶ 3 LED: rosso, verde, giallo o ambra
- ▶ 2 interruttori a pulsante
- ▶ 1 cicalino attivo
- ▶ Cavi jumper maschio-femmina (M2F) e femmina-femmina (F2F)
- ▶ Facoltativamente, una breadboard e cavi jumper maschio-maschio (M2M)

Lettura dei codici colore del resistore

I resistori sono disponibili in un'ampia gamma di valori: dalle versioni a resistenza zero, che sono sostanzialmente pezzi di filo, alle versioni ad alta resistenza dalle dimensioni di una gamba. Sono pochi però i resistori a presentare i valori numerici stampati: viene invece utilizzato un codice speciale (**Figura 6-4**) stampato come strisce o bande colorate attorno al corpo del resistore.

Per leggere il valore di un resistore, posizionalo in modo che il gruppo di bande sia a sinistra e la banda isolata a destra. Partendo dalla prima banda, cerca il colore nella colonna "1^a/2^a banda" della tabella per ottenere la prima e la seconda cifra. In questo esempio sono presenti due bande arancioni, che significano entrambe un valore di "3" per un totale di "33". Se il resistore presenta quattro bande raggruppate invece di tre, annota anche il valore della terza banda (per i resistori a cinque/sei bande, vedi rptl.io/5-6-band).

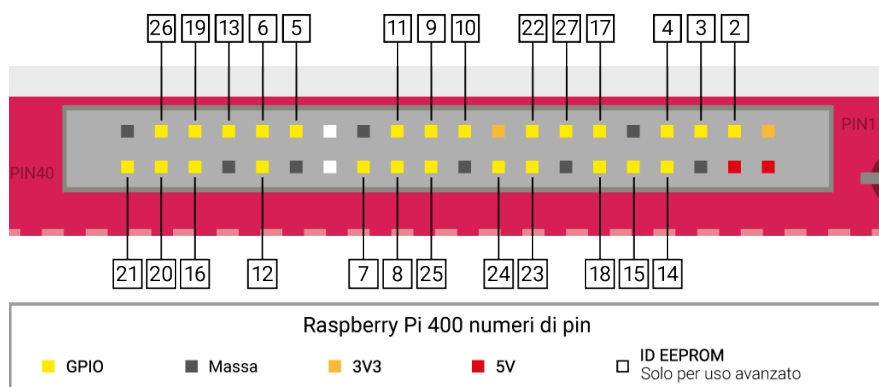


Figura 6-4 Codici colore dei resistori

Passando all'ultima banda nel gruppo (la terza o la quarta) cerca il colore nella colonna "Moltiplicatore". Indica per quale numero devi moltiplicare per ottenere il valore effettivo del resistore. In questo esempio è presente una banda marrone, che significa " $\times 10^1$ ". Può sembrare complesso, ma non è altro che la *notazione scientifica*: " $\times 10^1$ " significa semplicemente "aggiungere uno zero alla fine del numero". Se fosse blu, per " $\times 10^6$ ", significherebbe "aggiungere sei zeri alla fine del numero".

Prendiamo quindi il 33 dalle bande arancioni, più lo zero aggiunto dalla banda marrone e avremo 330, che è il valore del resistore misurato in ohm. L'ultima banda sulla destra è la *tolleranza* del resistore, ossia quanto si avvicina al valore nominale. I resistori più economici possono avere una banda argentata, che indica una tolleranza superiore o inferiore del 10% rispetto al valore nominale, oppure, se privi di banda, hanno una tolleranza superiore o inferiore del 20%. I resistori più costosi hanno una banda grigia, che indica una tolleranza entro lo 0,05% del valore nominale. Per i progetti amatoriali, la precisione non è così importante: generalmente va bene qualsiasi tolleranza.

Se il valore del resistore supera i 1000 ohm (1000 Ω), è solitamente valutato in kilohm (k Ω), se invece supera un milione di ohm, si tratta di megohm (M Ω). Un resistore da 2200 Ω verrebbe indicato come 2,2 k Ω , mentre da 2.200.000 Ω come 2,2 M Ω .

RIESCI A RISOLVERLO?

Quali bande colorate dovrebbe avere un resistore da 100 Ω ? Quali bande colorate dovrebbe avere un resistore da 2,2 M Ω ? Per trovare i resistori più economici che bande di tolleranza devi individuare?



Il tuo primo programma di physical computing: Ciao LED!

Proprio come la stampa di "Ciao mondo!" sullo schermo è un primo passo nell'apprendimento di un linguaggio di programmazione, riuscire a illuminare un LED è la tradizionale introduzione all'apprendimento del physical computing. Per questo progetto avrai bisogno di un LED e di un resistore da 330 ohm (330 Ω), o un valore simile, oltre che di cavi jumper femmina-femmina (F2F).



LA RESISTENZA È ESSENZIALE

Il resistore è un componente essenziale in questo circuito: protegge il Raspberry Pi e il LED limitando la quantità di corrente elettrica che il LED può assorbire. Senza di esso, il LED potrebbe assorbire troppa corrente e bruciarsi (o danneggiare il Raspberry Pi). Quando viene usato in questo modo, il resistore è noto come *resistenza di limitazione della corrente*. Il valore esatto del resistore necessario dipende dal LED che stai usando, tuttavia generalmente 330 Ω è adatto per la maggior parte dei LED più comuni. Più elevato è il valore, più tenue sarà la luce del LED, più è basso il valore, più luminoso sarà il LED.

Non collegare mai un LED a un Raspberry Pi senza una resistenza di limitazione di corrente, a meno che il LED non abbia un resistore incorporato di valore appropriato.

Inizia controllando che il LED funzioni. Ruota il Raspberry Pi in modo che il terminale GPIO sia posizionato come due strisce verticali sul lato destro. Collega un'estremità del resistore da 330 Ω al primo pin da 3,3 V (contrassegnato con 3V3 nella **Figura 6-5**) utilizzando un cavo jumper femmina-femmina, quindi collega l'altra estremità al terminale lungo (positivo o anodo) del LED con un altro cavo jumper femmina-femmina. Prendi un ultimo cavo jumper femmina-femmina e collega il terminale corto (negativo o catodo) del LED al primo pin massa (contrassegnato come GND nella **Figura 6-5**).

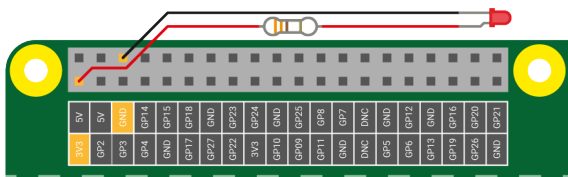


Figura 6-5 Collega il LED a questi pin senza dimenticare il resistore!

Se il Raspberry Pi è acceso, il LED dovrebbe accendersi. In caso contrario, controlla il circuito: assicurati di non aver usato una resistenza troppo elevata, che tutti i fili siano collegati correttamente e di aver scelto i pin GPIO giusti. Controlla anche i terminali del LED, in quanto funzionano in un solo modo: con il terminale più lungo collegato al lato positivo del circuito e quello più corto al negativo.

Una volta che il LED è funzionante è il momento di programmarlo. Scollega il cavo jumper dal pin 3,3 V (indicato come 3V3 nella **Figura 6-6**) e collegalo al pin 25 del GPIO (indicato come GP25 nella **Figura 6-6**). Il LED si spegne, ma non preoccuparti, è normale.

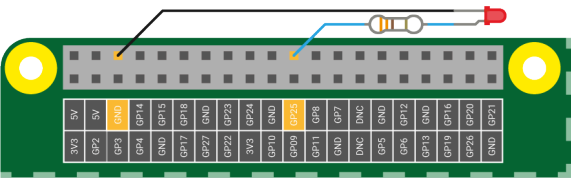


Figura 6-6 Scollega il cavo da 3V3 e collegalo al pin 25 del GPIO

Ora sei pronto per creare un programma Scratch o Python per accendere e spegnere il LED.


CONOSCENZE DI PROGRAMMAZIONE

Per eseguire i progetti di questo capitolo è necessario che tu sia a tuo agio con l'utilizzo di Scratch 3 e l'ambiente di sviluppo integrato Thonny Python (IDE). Se non l'hai ancora fatto, consulta il Capitolo 4, *Programmare con Scratch 3*, e il Capitolo 5, *Programmare con Python*, e lavora prima su quei progetti.

Se non hai ancora installato Scratch 3, segui le istruzioni di «Lo strumento Recommended Software» a pagina 43 per installarlo.



Controllo LED in Scratch

Carica Scratch 3 e fai clic sull'icona **Aggiungi un'Estensione** . Scorri verso il basso per trovare l'estensione **Raspberry Pi GPIO** (**Figura 6-7**), quindi fai clic su di essa. Si caricheranno i blocchi necessari per controllare il terminale GPIO di Raspberry Pi da Scratch 3. Visualizzerai nuovi blocchi nella categoria Raspberry Pi GPIO della palette dei blocchi.

ATTENZIONE

Al momento della stesura di questo articolo, Scratch 3 non dispone ancora di un aggiornamento per supportare Raspberry Pi 5. Se riscontri problemi, verifica la presenza di un aggiornamento (vedi «Aggiornamenti software» a pagina 49). Se non è disponibile alcun aggiornamento, controlla il repository GitHub di questa guida all'indirizzo rptl.io/bg-resources per informazioni sullo stato degli aggiornamenti futuri.



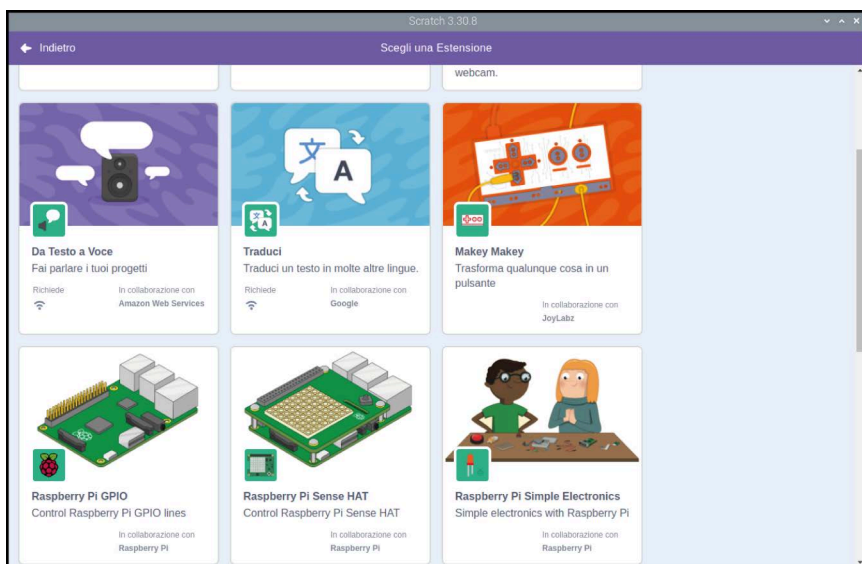
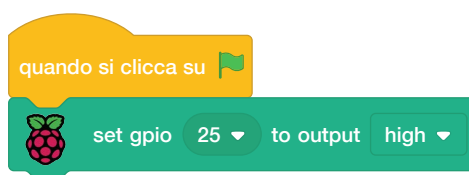
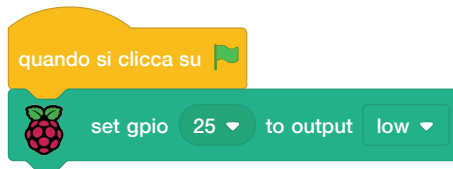


Figura 6-7 Aggiunta dell'estensione Raspberry Pi GPIO a Scratch 3

Inizia trascinando un blocco **Situazioni** **quando si clicca su** nell'area codice, quindi trascina il blocco verde **set gpio to output high** direttamente sotto di esso. Dovrai scegliere il numero del pin che stai usando: fai clic sulla freccia piccola per aprire la selezione a tendina e fai clic su **25** per dire a Scratch che stai controllando il pin GPIO 25.



Fai clic sulla bandiera verde per eseguire il programma. Vedrai il tuo LED illuminarsi. Congratulazioni: hai programmato il tuo primo progetto di physical computing! Fai clic sull'ottagono rosso per fermare il programma: vedrai che il LED rimane acceso. Questo perché il programma ha indicato a Raspberry Pi solamente di accendere il LED: questo è il significato della parte **output high** del blocco **set gpio 25 to output high**. Per spegnerlo, fai clic sulla freccia in giù alla fine del blocco, quindi seleziona **"low"** dall'elenco.



Fai di nuovo clic sulla bandiera verde, questa volta il programma spegnerà il LED. Per rendere le cose più interessanti, aggiungi un blocco di **Controllo** **per sempre** arancione e un paio di blocchi arancioni **attendi 1 secondi** per creare un programma per accendere e spegnere il LED ogni secondo.



Fai clic sulla bandiera verde e osserva il LED: si accenderà per un secondo, si spegnerà per un secondo, si accenderà di nuovo per un secondo e continuerà così fino a quando non farai clic sull'ottagono rosso per fermarlo. Guarda cosa succede quando fai clic sull'ottagono mentre il LED è in stato di accensione o spegnimento.

SFIDA: RIESCI A MODIFICARLO?

Come cambieresti il programma per far sì che il LED rimanga acceso più a lungo? Oppure spento più a lungo? Qual è il ritardo più piccolo che puoi usare per vedere comunque il LED accendersi e spegnersi?



Controllo LED in Python

Carica Thonny dalla sezione **Programmazione** del menu di Raspberry Pi, quindi fai clic sul pulsante **Nuovo** per avviare un nuovo progetto e **Salva** per salvarlo come **Ciao LED.py**. Per utilizzare i pin GPIO di Python, è necessaria una libreria chiamata GPIO Zero. Per questo progetto occorre solo la parte della libreria per lavorare con i LED. Importa solo questa sezione della libreria digitando quanto segue nell'area shell di Python:

```
from gpiozero import LED
```

Successivamente, è necessario indicare a GPIO Zero a quale pin GPIO è collegato il LED. Digita quanto segue:

```
led = LED(25)
```

Insieme, queste due righe consentono a Python di controllare i LED collegati ai pin GPIO del Raspberry Pi e indicano quale o quali pin (se è presente più di un LED nel circuito) controllare. Per controllare il LED e accenderlo, digita quanto segue:

```
led.on()
```

Per spegnerlo di nuovo, digita:

```
led.off()
```

Congratulazioni, ora stai controllando i pin GPIO del tuo Raspberry Pi in Python! Prova a digitare nuovamente le due istruzioni. Se il LED è già spento, digitando `led.off()` non succederà nulla, lo stesso vale se il LED è già acceso e si digita `led.on()`.

Per scrivere il tuo programma, digita quanto segue nell'area dello script:

```
from gpiozero import LED
from time import sleep
led = LED(25)
while True:
    led.on()
    sleep(1)
    led.off()
    sleep(1)
```

Questo programma importa la funzione **LED** dalla libreria gpiozero (GPIO Zero) e **sleep** dalla libreria "time", quindi realizza un ciclo infinito per accendere il LED per un secondo, spegnerlo per un secondo e ripetere. Fai clic sul pulsante **Esegui** per vederlo in azione: il LED inizierà a lampeggiare.

Come per il programma Scratch, prendi nota del comportamento quando fai clic sul pulsante **Ferma** mentre il LED è acceso rispetto a quando il LED è spento.

SFIDA: ILLUMINAZIONE PIÙ LUNGA

Come cambieresti il programma per far sì che il LED rimanga acceso più a lungo? Oppure spento più a lungo? Qual è il ritardo più piccolo che puoi usare per vedere il LED accendersi e spegnersi?



Utilizzo di una breadboard

I prossimi progetti di questo capitolo saranno molto più facili da completare se hai a disposizione una breadboard (**Figura 6-8**) per mantenere i componenti in posizione e realizzare i collegamenti elettrici.

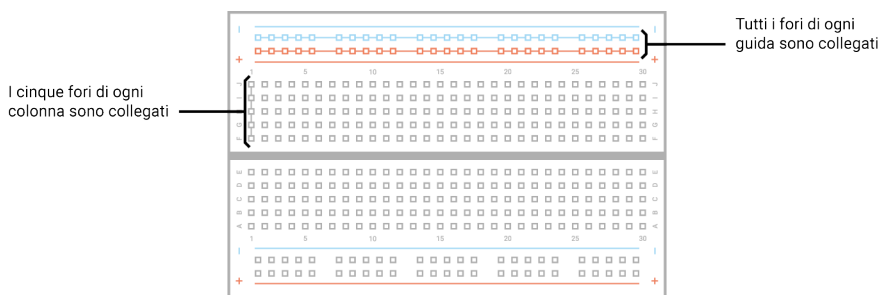


Figura 6-8 Una breadboard senza saldature

Una breadboard presenta fori distanziati di 2,54 mm tra loro per adattarsi alla maggior parte dei componenti. Sotto a questi fori sono presenti strisce in metallo (terminali) che si comportano come i cavi jumper usati finora. Queste sono disposte in colonna in tutta la breadboard. La maggior parte delle breadboard presenta uno spazio al centro che le divide a metà. Molte breadboard hanno anche lettere sul lato sinistro e numeri in alto e in basso per trovare più facilmente determinati fori: A1 è l'angolo in basso a sinistra, B1 è il foro appena sopra, mentre B2 è un foro a destra. A1 è collegato a B1 tramite le strisce metalliche nascoste, tuttavia nessun foro numerato è mai collegato a un foro con un numero differente a meno che tu non aggiunga personalmente un cavetto di collegamento.

Le breadboard più grandi presentano anche linee di fori in alto e in basso, tipicamente contrassegnate con linee rosse e nere o rosse e blu. Queste linee sono le *linee di alimentazione* e sono progettate per facilitare il cablaggio: è possibile collegare un singolo cavo dal pin ground (massa) del Raspberry Pi a una delle linee di alimentazione (tipicamente contrassegnate con una striscia blu o nera e un simbolo meno) per fornire una *massa comune* per molti componenti sulla breadboard e si può fare lo stesso se il circuito necessita di alimentazione a 3,3 V o 5 V.

Aggiungere componenti elettronici a una breadboard è semplice: basta allineare i loro terminali (le parti metalliche sporgenti) con i fori e spingere delicatamente fino a quando il componente non è in posizione. Per le connessioni da effettuare oltre a quelle che la breadboard fa per te, puoi usare cavi jumper maschio-maschio (M2M), mentre per le connessioni dalla breadboard al Raspberry Pi, usa cavi jumper maschio-femmina (M2F).



ATTENZIONE

Non provare mai a inserire in un foro più di un connettore o cavo jumper del componente. Ricorda: i fori sono collegati tra loro in colonne, a parte nella suddivisione centrale, quindi un connettore del componente in A1 è collegato elettricamente a qualsiasi cosa tu aggiunga a B1, C1, D1 ed E1.

Passaggi successivi: lettura di un pulsante

Gli output come i LED sono una cosa, ma la parte "input/output" di "GPIO" significa che si possono usare i pin anche come ingressi. Per questo progetto avrai bisogno di una breadboard, un cavo jumper maschio-maschio (M2M) e maschio-femmina (M2F), nonché di un interruttore a pulsante. Se non disponi di una breadboard puoi utilizzare cavi jumper femmina-femmina (F2F), ma il pulsante sarà molto più difficile da premere senza rischiare di rompere il circuito.

Inizia aggiungendo il pulsante alla breadboard. Se il pulsante ha solo due connettori, assicurati che siano in fori diversamente numerate della breadboard, mentre se ne ha quattro, giralo in modo che i lati da cui escono i terminali siano allineati come mostrato nella **Figura 6-9**. Collega la massa della breadboard a un pin massa del Raspberry Pi (contrassegnato con GND) con un cavo jumper maschio-femmina, quindi collega un terminale del pulsante alla massa con un cavo jumper maschio-maschio. Infine, collega l'altro terminale (quello sullo stesso lato del terminale appena collegato se si utilizza un interruttore a quattro connettori) al pin GPIO 2 (contrassegnato con GP2) del Raspberry Pi con un cavo jumper maschio-femmina.

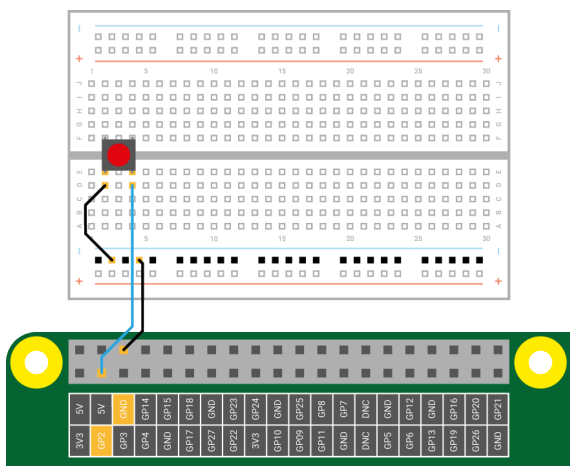
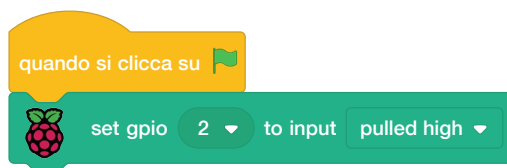


Figura 6-9 Cablaggio del pulsante ai pin GPIO

Letture di un pulsante in Scratch

Avvia un nuovo programma Scratch e trascina un blocco **quando si clicca su** arancione nell'area codice. Collega un blocco verde **set gpio to input pulled high** e seleziona il numero 2 dal menu a tendina in modo che corrisponda al pin GPIO utilizzato per il pulsante.



Se fai clic sulla bandiera verde non accadrà nulla. Questo perché hai detto a Scratch di usare il pin come input, ma non cosa fare con quell'input. Trascina un blocco **per sempre** alla fine della sequenza, quindi trascina un blocco **se allora altrimenti** al suo interno. Cerca il blocco **gpio is high?** verde, trascinalo nello spazio bianco a forma di diamante nella parte **se allora** del blocco arancione e usa il menu a tendina per selezionare il numero 2, che indica quale pin GPIO controllare. Trascina un blocco viola **dire Ciao! per 2 secondi** nella parte **altrimenti** del blocco e modificalo affinché ci sia scritto **"Pulsante premuto!"**. Lascia per ora vuoto lo spazio tra **se allora** e **altrimenti** nel blocco arancione.



C'è molto da fare qui. Inizia testando il programma: fai clic sulla bandiera verde, poi premi il pulsante sulla tua breadboard. Il tuo sprite dovrebbe dirti che il pulsante è stato premuto. Congratulazioni: hai letto con successo un input dal pin GPIO!

Poiché lo spazio tra **se allora** e **altrimenti** nel blocco arancione è vuoto per il momento, non succede nulla quando **gpio 2 is high?** valuta true. Il codice che viene eseguito quando il pulsante viene effettivamente premuto si trova nella parte **altrimenti** del blocco. Sembra contraddittorio: premendo il pulsante non andrà verso l'alto? In realtà, è il contrario: i pin GPIO di Raspberry Pi sono normalmente alzati, o accesi, quando sono impostati come input e premendo il pulsante vengono portati verso il basso.

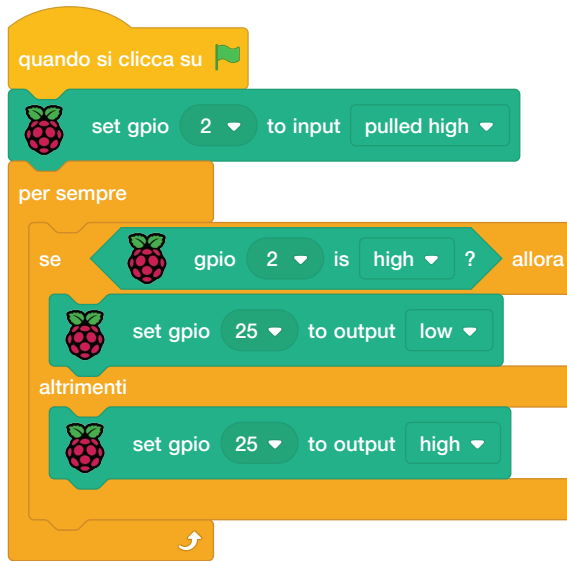
Guarda di nuovo il tuo circuito: osserva come il pulsante sia collegato al pin GPIO 2, che fornisce la parte positiva del circuito, e al pin di massa. Quando si preme il pulsante, la tensione sul pin GPIO viene abbassata tramite il polo di massa e il programma Scratch smette di eseguire il codice (se presente) nel blocco **if gpio 2 is high ? then** ed esegue invece il codice nella parte **altrimenti** del blocco.

Se tutto ciò può sembrare strano, ricorda che il pulsante sul pin GPIO Raspberry Pi viene considerato premuto quando il pin va verso il basso e non il contrario.

Per estendere ulteriormente il programma, aggiungi il LED e il resistore al circuito. Ricordati di collegare il resistore al pin 25 del GPIO e alla parte lunga del LED, mentre la parte più corta del LED al pin di massa della breadboard.

Trascina il blocco **dire Pulsante premuto! per 2 secondi** dall'area del codice alla palette dei blocchi per cancellarlo, quindi sostituiscilo con un blocco

set gpio 25 to output high verde, ricordandoti che dovrai cambiare il numero GPIO usando la freccia verso il basso. Aggiungi un blocco verde **set gpio 25 to output low**, ricordandoti di modificarne i valori, nella parte vuota **if gpio 2 is high ? then** del blocco.



Fai clic sulla bandiera verde e premi il pulsante. Il LED resterà acceso finché tieni premuto il pulsante; rilasciandolo, si spegnerà di nuovo. Complimenti: stai controllando un pin GPIO sulla base di un input di un altro!

SFIDA: MANTIENI ACCESO IL LED

Come cambieresti il programma per far rimanere acceso il LED per qualche secondo, anche dopo aver lasciato il pulsante? Cosa bisogna cambiare per avere il LED acceso mentre non premi il pulsante e spento quando lo premi?



Lettura di un pulsante in Python

Fai clic sul pulsante **Nuovo** in Thonny per avviare un nuovo progetto e sul pulsante **Salva** per salvarlo come **Input pulsante.py**. L'uso di un pin GPIO come ingresso per un pulsante è molto simile all'uso di un pin come uscita per un LED, ma è necessario importare una parte diversa della libreria GPIO Zero. Digita quanto segue nell'area dello script:

```
from gpiozero import Button
button = Button(2)
```

Per far eseguire il codice quando il pulsante viene premuto, GPIO Zero fornisce l'opzione `wait_for_press`. Digita quanto segue:

```
button.wait_for_press()
print("Mi hai premuto!")
```

Fai clic sul pulsante **Esegui**, quindi premi l'interruttore a pulsante. Il messaggio verrà stampato nella Python shell in fondo alla finestra di Thonny. Congratulazioni: hai letto con successo un input dal pin GPIO!

Se vuoi riprovare il programma, dovrai fare nuovamente clic sul pulsante **Esegui**. Poiché non c'è una ripetizione nel programma, questo si chiude non appena finisce di stampare il messaggio nell'area shell.

Per estendere ulteriormente il programma, aggiungi il LED e il resistore nel circuito se non l'hai già fatto: ricorda di collegare il resistore al pin GPIO 25 e al terminale lungo del LED, mentre quello corto del LED al pin di massa sulla breadboard.

Per controllare un LED, oltre a leggere un pulsante, è necessario importare sia l'opzione `Button` sia `LED` della libreria GPIO Zero. Avrai anche bisogno della funzione `sleep` della libreria `time`. Torna all'inizio del programma e inserisci quanto segue come prime due righe:

```
from gpiozero import LED
from time import sleep
```

Sotto la linea `button = Button(2)`, digita:

```
led = LED(25)
```

Elimina la riga `print("Mi hai premuto!")` e sostituiscila con:

```
led.on()
sleep(3)
led.off()
```

Il programma finito dovrebbe avere questa struttura:

```
from gpiozero import LED
from time import sleep
from gpiozero import Button

button = Button(2)
led = LED(25)
```

```
button.wait_for_press()
led.on()
sleep(3)
led.off()
```

Fai clic sul pulsante **Esegui**, quindi premi l'interruttore a pulsante, il LED si accenderà per tre secondi, quindi si spegnerà e il programma verrà chiuso. Complimenti: ora sai come controllare un LED utilizzando un input del pulsante in Python!

SFIDA: AGGIUNGERE UNA RIPETIZIONE

Come si fa ad aggiungere una ripetizione per far ripetere il programma invece di terminare dopo la pressione di un tasto? Cosa bisogna cambiare per avere il LED acceso mentre non premi il pulsante e spento quando lo premi?



Facciamo un po' di rumore: controllare un cicalino

I LED sono un ottimo dispositivo di output, ma non sono molto utili se non li si guarda. La soluzione sono i cicalini, che producono un rumore udibile in qualsiasi punto della stanza. Per questo progetto avrai bisogno di una breadboard, di cavi jumper maschio-femmina (M2F) e di un cicalino attivo. Se non disponi di una breadboard puoi collegare il cicalino utilizzando cavi jumper femmina-femmina (F2F).

Un cicalino attivo può essere trattato esattamente come un LED, in termini di circuiteria e programmazione. Ripeti la procedura del circuito che hai realizzato per il LED, ma sostituisci il LED con il cicalino attivo e ometti il resistore, poiché il cicalino avrà bisogno di più corrente per funzionare. Collega un terminale del cicalino al pin GPIO 15 (contrassegnato GP15 nella **Figura 6-10**) e l'altro al pin di massa (contrassegnato GND nello schema) utilizzando la breadboard e i cavi jumper maschio-femmina.

Se il cicalino ha tre terminali, assicurati che il terminale contrassegnato con il simbolo meno (-) sia collegato al pin di massa e quello contrassegnato con "S" o "SIGNAL" al pin 15, quindi collega il terminale rimanente, generalmente quello centrale, al pin da 3,3 V (contrassegnato con 3V3).

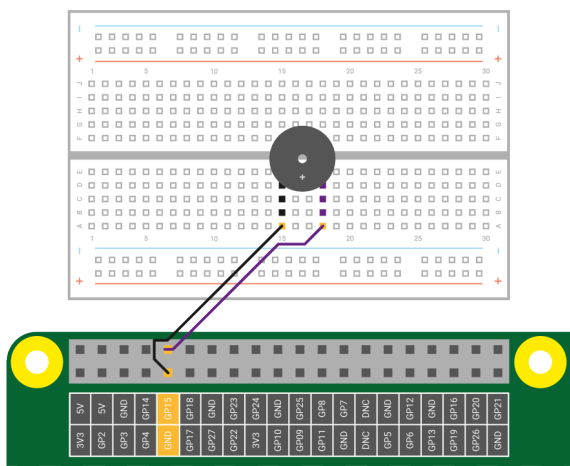
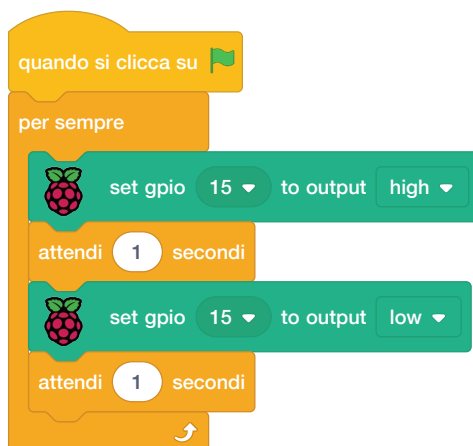


Figura 6-10 Collegamento di un cicalino ai pin GPIO

Controllo di un cicalino in Scratch

Ricrea il programma che hai usato per far lampeggiare il LED o caricarlo, se lo hai salvato in precedenza. Utilizza il menu a tendina nei blocchi verdi **set gpio to output high** per selezionare il numero **15**, in modo che Scratch controlli il pin GPIO corretto.



Fai clic sulla bandiera verde e il cicalino inizierà a produrre un ronzio: un secondo sì e uno no. Se senti il cicalino fare clic solo una volta al secondo, significa che è passivo, non attivo. Per far vibrare le piastre metalliche, il cicalino attivo genera un segnale che cambia rapidamente, noto come *oscillazione*, mentre il cicalino passivo deve ricevere un segnale di oscillazione esterno e non

lo produce da sé. Quando lo si accende con Scratch, le piastre si muovono una sola volta e si fermano, producendo il "clic" fino alla prossima accensione o allo spegnimento del pin da parte del programma.

Fai clic sull'ottagono rosso per fermare il cicalino, ma assicurati di farlo quando non emette alcun suono, altrimenti il cicalino continuerà a suonare fino a quando non eseguirai di nuovo il programma!

SFIDA: CAMBIARE IL SUONO

Come si può cambiare il programma per far suonare il cicalino per un tempo più breve? Si può costruire un circuito in modo che il cicalino sia controllato da un pulsante?



Controllo di un cicalino in Python

Il controllo di un cicalino attivo attraverso la libreria GPIO Zero è quasi identico al controllo di un LED, in quanto ha stati di accensione e spegnimento. Hai bisogno di una funzione diversa, però: **Buzzer**. Crea un nuovo progetto in Thonny e salvalo come **Cicalino.py**, quindi digita quanto segue:

```
from gpiozero import Buzzer
from time import sleep
```

Come per i LED, GPIO Zero ha bisogno di sapere a quale pin è collegato il cicalino per poterlo controllare. Digita quanto segue:

```
buzzer = Buzzer(15)
```

Da qui, il programma è quasi identico a quello che hai scritto per controllare il LED: l'unica differenza (a parte un diverso numero di pin GPIO) è che stai usando **buzzer** al posto di **led**. Digita quanto segue:

```
while True:
    buzzer.on()
    sleep(1)
    buzzer.off()
    sleep(1)
```

Fai clic sul pulsante **Esegui** e il cicalino inizierà a produrre un ronzio: un secondo sì e uno no. Se stai usando un cicalino passivo invece di uno attivo, sentirai solo un breve clic ogni secondo invece di un ronzio continuo.

Fai clic sul pulsante **Ferma** per uscire dal programma, ma assicurati che il cicalino non emetta alcun suono in quel momento, altrimenti continuerà a suonare fino a quando non eseguirai di nuovo il programma.

Progetto Scratch: Semafori

Ora che sai come usare pulsanti, cicalini e LED come input e output, sei pronto per costruire un esempio di computazione del mondo reale, ossia dei semafori, completi di pulsante da premere per attraversare la strada. Per questo progetto avrai bisogno di una breadboard, tre LED (rosso, giallo e verde), tre resistori da 330 Ω , un cicalino, un interruttore a pulsante e infine una selezione di cavi jumper maschio-maschio (M2M) e maschio-femmina (M2F).

Inizia a costruire il circuito (**Figura 6-11**) collegando il cicalino al pin 15 del GPIO (indicato come GP15 nella **Figura 6-11**), il LED rosso al pin 25 (indicato come GP25), il LED giallo al pin 8 (GP8), il LED verde al pin 7 (GP7) e l'interruttore al pin 2 (GP2). Ricordati di collegare i resistori da 330 Ω tra i pin GPIO e i terminali lunghi dei LED e di collegare i secondi terminali di tutti i componenti al conduttore di massa della breadboard. Infine, collega il conduttore di massa a un pin di massa (contrassegnato GND) sul Raspberry Pi per completare il circuito.

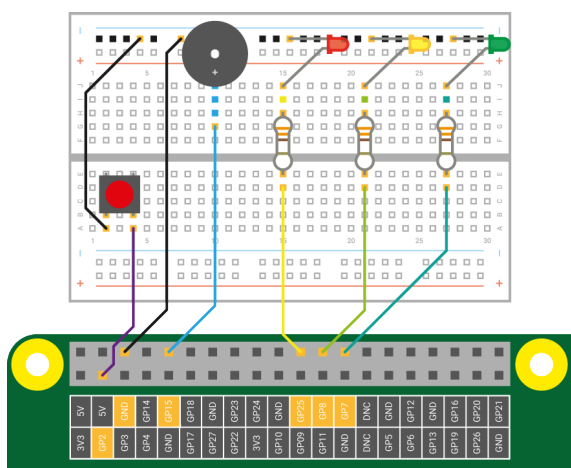
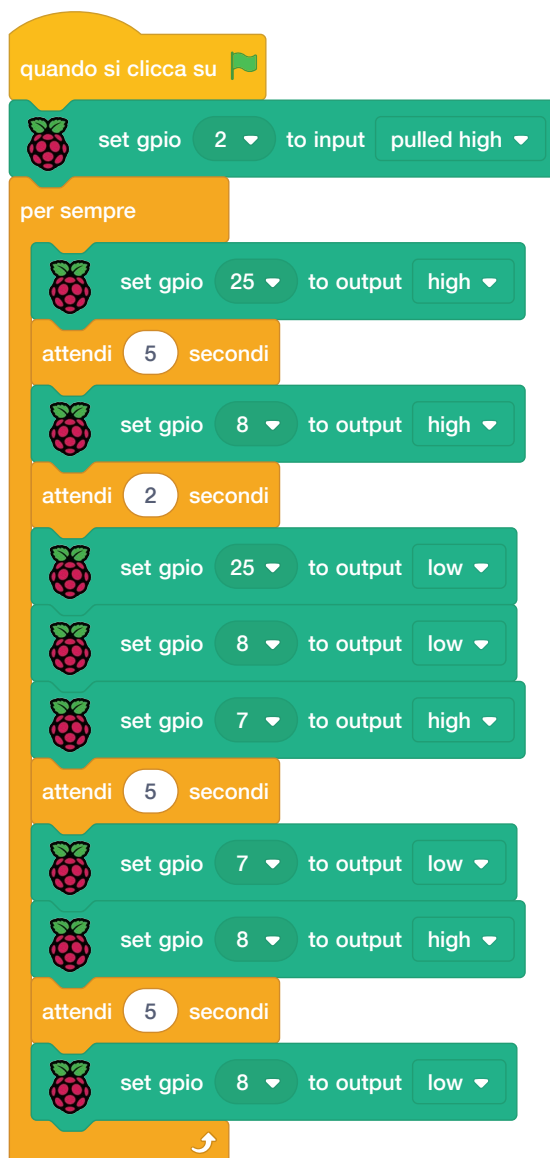


Figura 6-11 Schema di cablaggio per il progetto Semaforo

Crea un nuovo progetto Scratch 3 e trascina un blocco **quando si clicca su** nell'area codice. Successivamente, dovrai dire a Scratch che il pin 2 del GPIO, collegato all'interruttore a pulsante, è un input e non un output. Trascina un blocco **set gpio to input pulled high** verde dalla categoria **Raspberry Pi GPIO** della palette dei blocchi sotto il blocco **quando si clicca su**. Fai clic sulla freccia verso il basso accanto a **0** e seleziona il numero **2** dall'elenco a discesa.

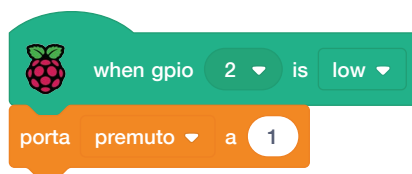
Successivamente, è necessario creare la sequenza del semaforo. Trascina un blocco arancione **per sempre** nel programma, quindi aggiungi blocchi per accendere e spegnere i LED del semaforo secondo uno schema. Ricordati quali pin GPIO hanno ciascun componente collegato: con il pin 25 si usa il LED rosso, con il pin 8 il LED giallo e con il 7 il LED verde.



Fai clic sulla bandiera verde e osserva i LED: prima si accenderà il rosso, poi sia il rosso sia il giallo, poi il verde, poi il giallo e infine la sequenza si ripete a partire dalla luce rossa. Questo schema corrisponde a quello utilizzato dai semafori nel Regno Unito: se lo desideri, è possibile modificare la sequenza in modo che corrisponda agli schemi di altri Paesi.

Per simulare un attraversamento pedonale, è necessario che il programma attenda che venga premuto il pulsante. Fai clic sull'ottagono rosso per fermare il programma se è in funzione. Trascina un blocco arancione **se allora altrimenti** nell'area script e collegalo in modo che si trovi direttamente sotto il blocco **per sempre**, con la sequenza del semaforo nella sezione **se allora**. Per ora lascia vuoto lo spazio a forma di diamante.

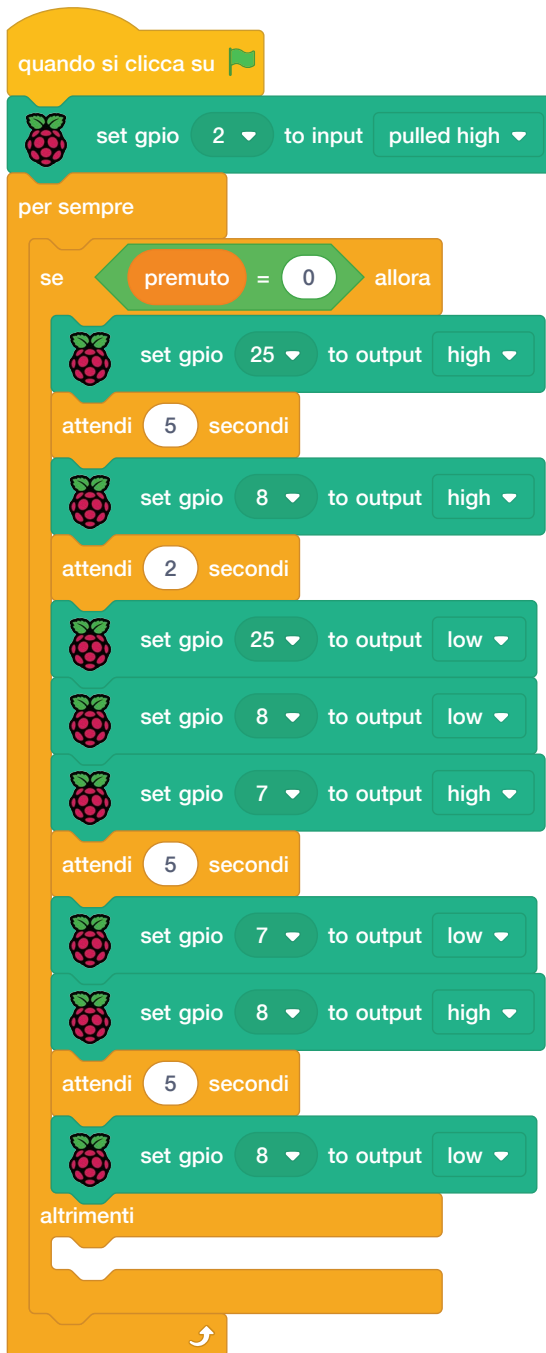
Un vero e proprio attraversamento pedonale non attiva il rosso non appena si preme il pulsante, ma attende il semaforo rosso successivo nella sequenza. Per inserirlo nel programma, trascina un blocco **when gpio is low** nell'area del codice e seleziona **2** dall'elenco a discesa. Crea una nuova variabile chiamata **premuto**. Poi trascina un blocco **porta premuto a 1** arancione sotto di esso.



Questi blocchi attendono che il pulsante venga premuto, quindi impostano la variabile **premuto** su 1. Impostando la variabile in questo modo è possibile memorizzare il fatto che il pulsante è stato premuto, anche se non lo utilizzerai subito.

Torna alla tua sequenza di blocchi originale e cerca il blocco **se allora**. Trascina un blocco Operatori verde a forma di diamante **=** nello spazio vuoto a forma di diamante del blocco **se allora**, quindi trascina un blocco reporter arancione scuro **pushed** nel primo spazio vuoto. Digita **0** sopra a **50** sul lato destro del blocco.

Fai clic sulla bandiera verde e guarda la sequenza di luci del semaforo. Quando sei pronto premi il pulsante: all'inizio sembrerà che non stia succedendo nulla, ma una volta che la sequenza è giunta al termine (con il solo LED giallo acceso) le luci del semaforo si spegneranno e rimarranno spente, grazie alla variabile **premuto**.



Non resta che fare in modo che il pulsante di attraversamento pedonale non spenga solo le luci. Nella sequenza di blocchi principale, cerca il blocco **altrimenti** e trascina al suo interno un blocco **set gpio 25 to output high**, ricordandoti di cambiare il numero di pin GPIO predefinito in modo che corrisponda al pin a cui è collegato il LED rosso.

Sotto di esso, sempre nel blocco **altrimenti**, crea uno schema per il cicalino: trascina un blocco arancione **ripeti 10 volte**, poi inserisci al suo interno i blocchi verde **set gpio 15 to output high**, arancione **attendi 0.2 secondi**, verde **set gpio 15 to output low** e un altro blocco arancione **attendi 0.2 secondi**, cambiando i valori del pin GPIO per farli corrispondere al pin del cicalino.

Infine, sotto la parte inferiore del blocco **ripeti 10 volte**, ma sempre nel blocco **altrimenti**, aggiungi un blocco **set gpio 25 to output low** verde e un blocco **porta premuto a 0** arancione scuro. L'ultimo blocco resetta la variabile che memorizza la pressione del pulsante, in modo che la sequenza del cicalino non si ripeta all'infinito.

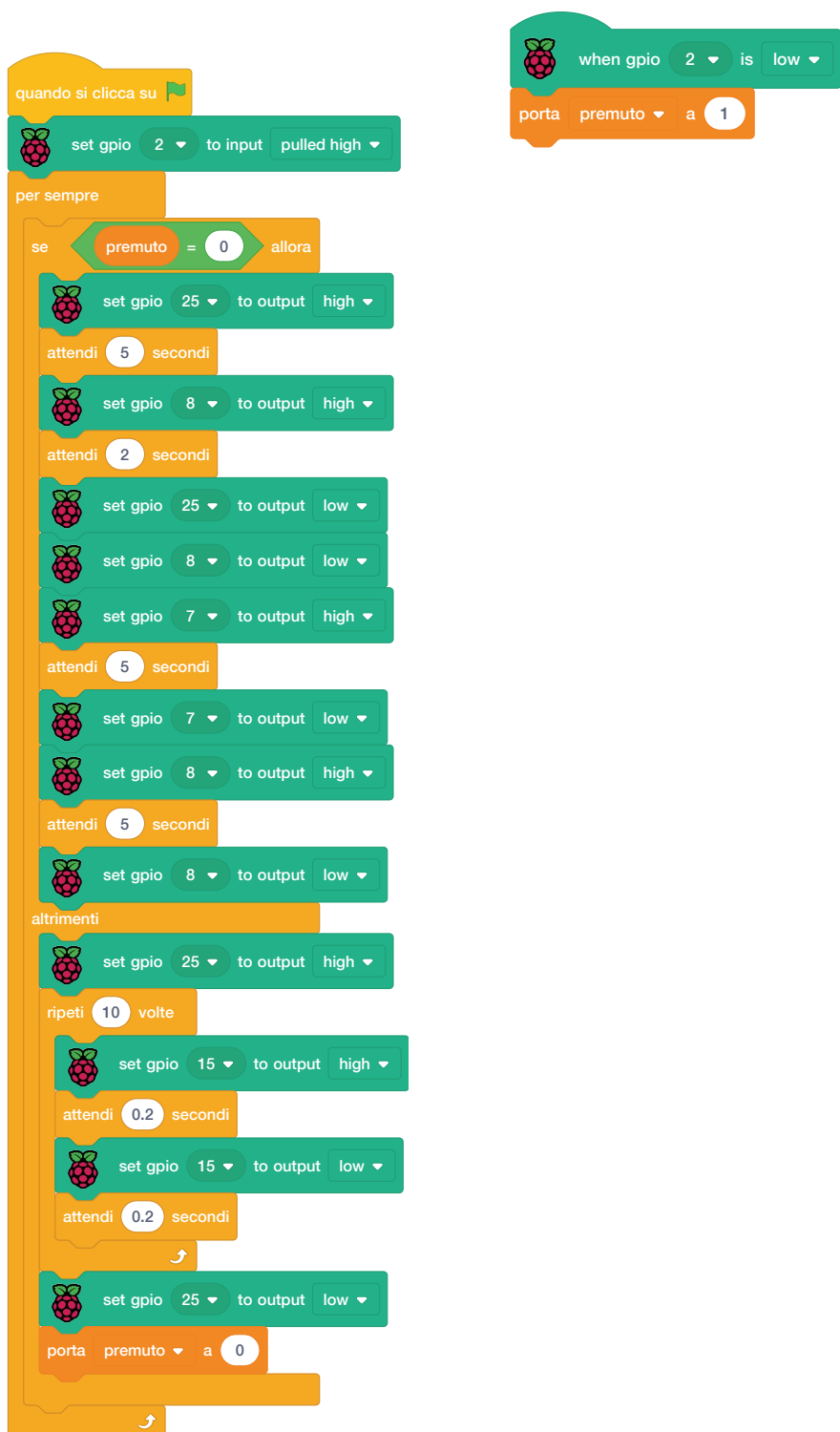
Fai clic sulla bandiera verde, poi premi l'interruttore sulla breadboard. Al termine della sequenza, vedrai la luce rossa accendersi e sentirai il cicalino suonare per far sapere ai pedoni che è sicuro attraversare. Dopo un paio di secondi il cicalino si ferma e la sequenza del semaforo ricomincia e continua fino alla successiva pressione del tasto.

Complimenti: hai programmato il tuo semaforo perfettamente funzionante, completo di attraversamento pedonale!



SFIDA: PUOI MIGLIORARLO?

Sei in grado di modificare il programma in modo che il pedone abbia più tempo per attraversare? Riesci a trovare informazioni sulle sequenze semaforiche di altri Paesi e riprogrammare le luci in modo che corrispondano? Come potresti rendere i LED meno luminosi?



Progetto Python: Gioco di reazione

Ora che sai come usare pulsanti e LED come input e output, sei pronto per costruire un esempio di computazione del mondo reale: un gioco di reazione per due giocatori, progettato per vedere chi ha il tempo di reazione più veloce! Per questo progetto, avrai bisogno di una breadboard, un LED, un resistore da 330 Ω , due interruttori a pulsante, alcuni cavi jumper maschio-femmina (M2F) e maschio-maschio (M2M).

Inizia costruendo il circuito (**Figura 6-12**): collega il primo interruttore sul lato sinistro della breadboard al pin 14 del GPIO (indicato come GP14 nella **Figura 6-12**). Il secondo interruttore sul lato destro della breadboard va al pin 15 (indicato come GP15); il terminale più lungo del LED si collega al resistore da 330 Ω , che poi si collega al pin 4 del GPIO (indicato come GP4). Il secondo terminale di tutti i componenti va collegato al conduttore di massa della breadboard. Infine, collega il conduttore di massa al pin di massa del Raspberry Pi (contrassegnato GND).

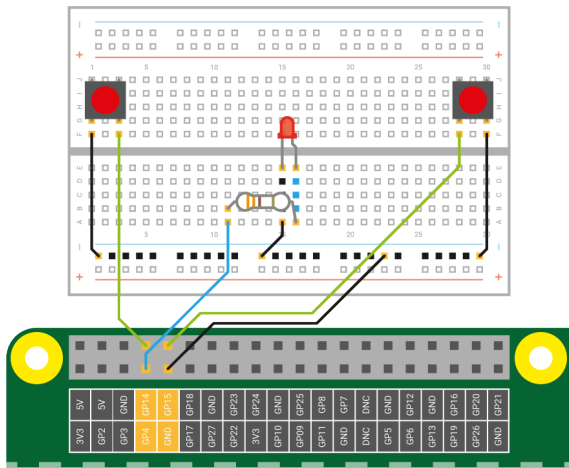


Figura 6-12 Schema di cablaggio per il gioco per la velocità di reazione

Crea un nuovo progetto in Thonny e salvalo come **Gioco per la velocità di reazione.py**. Utilizzerai le funzioni **LED** e **button** della libreria GPIO Zero e la funzione **sleep** dalla libreria time. Invece di importare le due funzioni GPIO Zero su due righe separate puoi risparmiarti tempo e importarle insieme usando il simbolo di una virgola (,) per separarle. Digita quanto segue nell'area dello script:

```
from gpiozero import LED, Button
```

```
from time import sleep
```

Come prima, dovrai indicare a GPIO Zero a quali pin sono collegati i due pulsanti e il LED. Digita quanto segue:

```
led = LED(4)
right_button = Button(15)
left_button = Button(14)
```

Ora aggiungi le istruzioni per accendere e spegnere il LED, in modo da poter controllare che funzioni correttamente:

```
led.on()
sleep(5)
led.off()
```

Fai clic sul pulsante **Esegui**. Il LED si accende per cinque secondi e poi si spegne. Poi il programma si chiuderà. Ai fini di un gioco di reazione, però, se il LED si spegne sempre dopo esattamente cinque secondi è un po' prevedibile. Aggiungi questo comando sotto la riga `from time import sleep`:

```
from random import uniform
```

La libreria `random`, come suggerisce il nome, consente di generare numeri casuali (qui con una distribuzione uniforme, vedi rptl.io/uniform-dist). Trova la riga `sleep(5)` e modificala come segue:

```
sleep(uniform(5, 10))
```

Fai nuovamente clic sul pulsante **Esegui**: questa volta il LED rimarrà acceso per un numero di secondi casuale compreso tra 5 e 10. Conta per vedere quanto tempo impiega il LED a spegnersi, poi fai clic qualche altra volta sul pulsante **Esegui**. Vedrai che il tempo è diverso per ogni esecuzione, rendendo il programma meno prevedibile.

Per trasformare i pulsanti in trigger per ogni giocatore, è necessario aggiungere una funzione. Al termine del programma digita quanto segue:

```
def pressed(button):
    print(str(button.pin.number) + " ha vinto la partita")
```

Ricorda che Python usa l'indentazione per comprendere quali righe fanno parte della funzione: Thonny farà rientrare automaticamente la seconda riga.

Infine aggiungi le seguenti due righe per individuare quando i giocatori premono i pulsanti, ricordati che non devono essere rientranti, altrimenti Python le tratterà come parte della funzione.

```
right_button.when_pressed = pressed
left_button.when_pressed = pressed
```

Esegui il programma, questa volta prova a premere uno dei due pulsanti non appena il LED si spegne. Vedrai un messaggio che indica quale pulsante è stato premuto prima, stampato nella shell Python in fondo alla finestra di Thonny. Ogni volta che viene premuto uno dei due pulsanti vedrai lo stesso messaggio con indicato il numero di pin anziché un nome più comprensibile per il pulsante.

Per risolvere il problema, chiedi ai giocatori i loro nomi. Sotto la riga `from random import uniform`, digita quanto segue:

```
left_name = input("Il nome del giocatore a sinistra è ")
right_name = input("Il nome del giocatore a destra è ")
```

Torna alla funzione e sostituisci la riga `print(str(button.pin.number) + " ha vinto la partita")` con:

```
if button.pin.number == 14:
    print(left_name + " ha vinto la partita")
else:
    print(right_name + " ha vinto la partita")
```

Fai clic sul pulsante **Esegui**, quindi digita i nomi di entrambi i giocatori nell'area shell di Python. Quando premerete il pulsante, ricordandovi di farlo il più velocemente possibile dopo lo spegnimento del LED, vedrete che al posto del numero del pin viene visualizzato il nome del giocatore.

Per fare in modo che non vengano segnalate come vincenti tutte le pressioni dei pulsanti, è necessario aggiungere una nuova funzione dalla libreria sys (abbreviazione di *sistema*): `exit`. Sotto l'ultima riga `import`, digita quanto segue:

```
from os import _exit
```

Quindi, alla fine della funzione, sotto la riga `print(right_name + " ha vinto la partita")`, digita quanto segue:

```
_exit(0)
```

In questo caso l'indentazione è importante: `_exit(0)` dovrebbe essere preceduto da quattro spazi, allineato con `else` due righe sopra di esso e con `if` altre due righe sopra. Questa istruzione indica a Python di fermare il programma dopo che è stato premuto il primo pulsante, il che significa che il giocatore che preme il pulsante troppo tardi non viene premiato.

Il programma finito dovrebbe avere questa struttura:

```
from gpiozero import LED, Button
from time import sleep
from random import uniform
from os import _exit

left_name = input("Il nome del giocatore a sinistra è ")
right_name = input("Il nome del giocatore a destra è ")
led = LED(4)
right_button = Button(15)
left_button = Button(14)

led.on()
sleep(uniform(5, 10))
led.off()

def pressed(button):
    if button.pin.number == 14:
        print(left_name + " ha vinto la partita")
    else:
        print(right_name + " ha vinto la partita")
    _exit(0)

right_button.when_pressed = pressed
left_button.when_pressed = pressed
```

Fai clic sul pulsante **Esegui**, inserisci i nomi dei giocatori, aspetta che il LED si spenga e vedrai il nome del giocatore vincente. Vedrai anche un messaggio da Python: **Process ended with exit code 0**. Questo significa che Python ha ricevuto il comando `_exit(0)` e ha interrotto il programma ed è pronto per le prossime istruzioni. Se vuoi giocare di nuovo, fai clic ancora una volta sul pulsante **Esegui**.

Complimenti, hai creato il tuo gioco fisico!

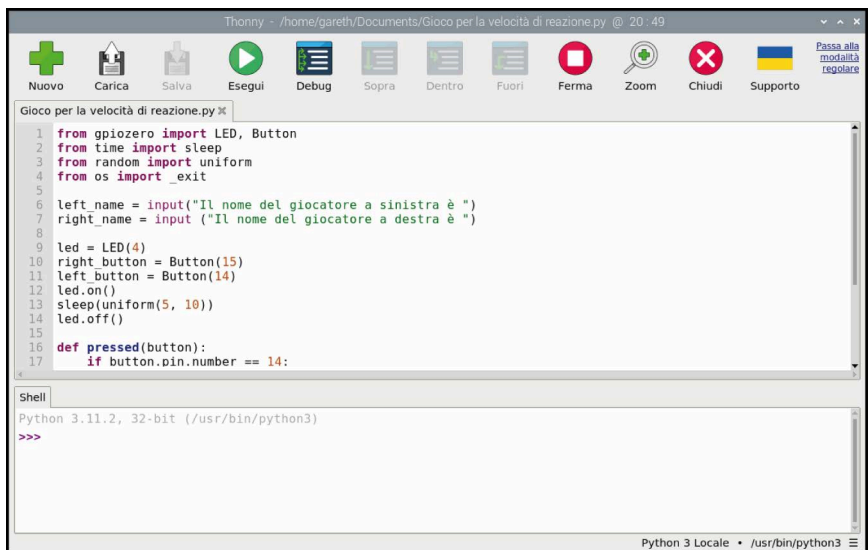
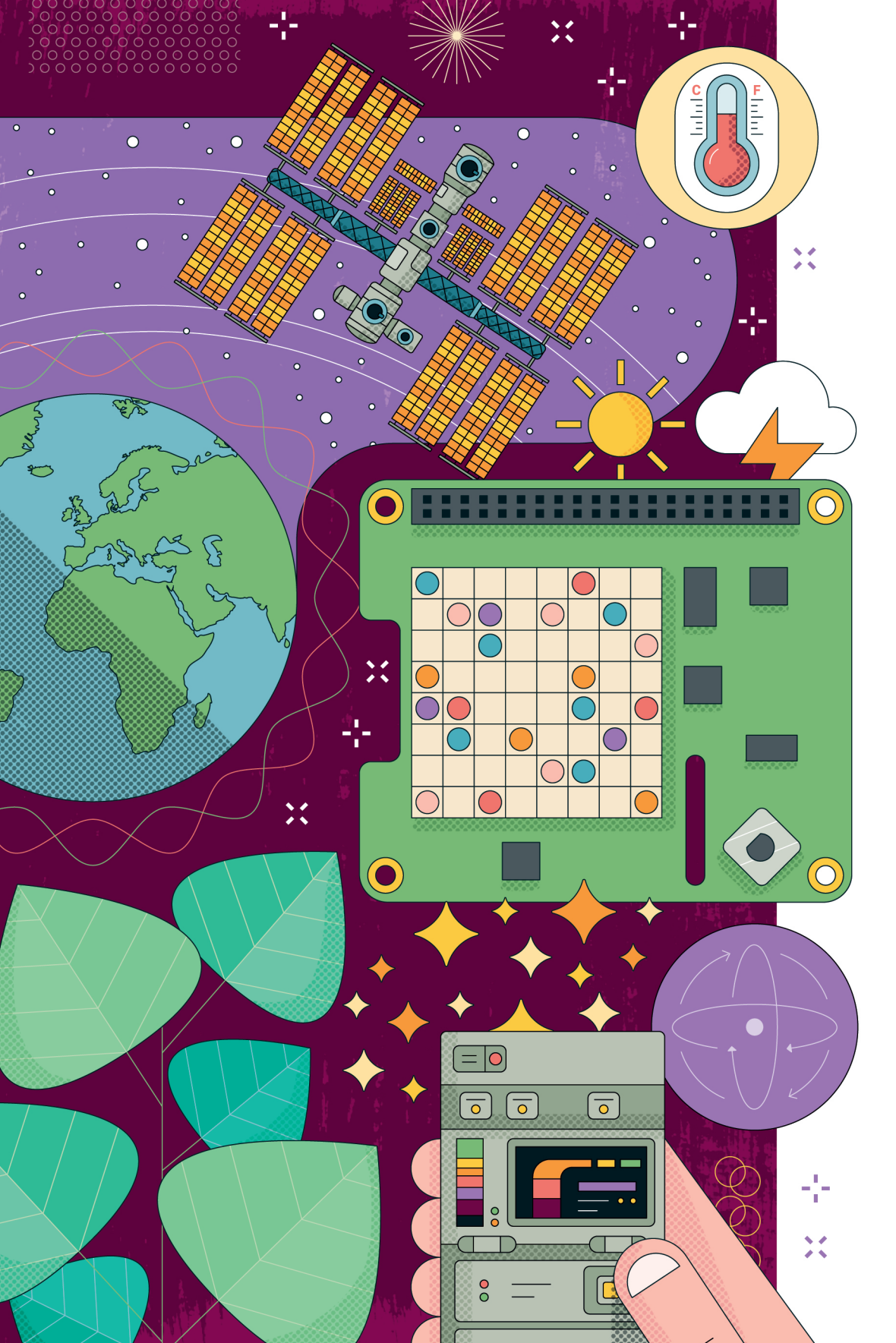


Figura 6-13 Il primo a premere il pulsante dopo che la luce si è spenta sarà dichiarato vincitore



SFIDA: MIGLIORA IL GIOCO

Riesci ad aggiungere una ripetizione, in modo che il gioco funzioni continuamente? Ricordati di rimuovere l'istruzione `_exit(0)` prima. Riesci ad aggiungere un conteggio per il punteggio in modo da poter vedere chi sta vincendo nei vari round? Che ne dici di un timer, in modo da vedere quanto tempo ci è voluto per reagire allo spegnimento della luce?



Capitolo 7

Physical computing con Sense HAT

Sense HAT è una scheda aggiuntiva multifunzione Raspberry Pi dotata di sensori e display a matrice LED, la stessa usata nella Stazione spaziale internazionale.

Raspberry Pi supporta un tipo speciale di scheda aggiuntiva chiamata *HAT* (*Hardware Attached on Top*). Gli HAT possono aggiungere al Raspberry Pi microfoni, luci, relè elettronici, schermi e molto altro ancora, ma un HAT in particolare è molto speciale: il Sense HAT.

Sense HAT è stato progettato appositamente per la missione spaziale Astro Pi. Con il progetto Astro Pi, nato dalla collaborazione tra Raspberry Pi Foundation, UK Space Agency e l'Agenzia spaziale europea (ESA), le schede, le fotocamere e i Sense HAT di Raspberry Pi sono stati portati dal veicolo spaziale Cygnus di Orbital Science sulla Stazione spaziale internazionale (ISS). Da quando sono stati portati in orbita, i dispositivi Raspberry Pi (soprannominati Ed e Izzy dagli astronauti) sono stati usati per eseguire codici ed esperimenti scientifici con il contributo di decine di migliaia di studenti da diversi paesi in tutta Europa. Un nuovo e aggiornato hardware Raspberry Pi (dei Raspberry Pi 4, soprannominati Flora, Fauna e Fungi) è stato inviato sulla ISS nel 2022. Se sei in Europa e hai meno di 19 anni, puoi scoprire come eseguire il tuo codice e i tuoi esperimenti nello spazio all'indirizzo **astro-pi.org**.

Puoi trovare lo stesso hardware Sense HAT impiegato nella ISS anche qui sulla Terra, presso tutti i rivenditori Raspberry Pi e, se in questo momento non vuoi acquistare un Sense HAT, puoi simularlo tramite software.



REALE O SIMULATO

Questo capitolo è più facile da comprendere se utilizzi un vero Sense HAT collegato al terminale GPIO di un Raspberry Pi, ma se non ne possiedi uno, puoi saltare «Installazione del Sense HAT» a pagina 158 e provare a realizzare i progetti nell'emulatore Sense HAT.

Cos'è Sense HAT

Il Sense HAT (**Figura 7-1**) è un potente componente aggiuntivo per Raspberry Pi, che racchiude diverse funzioni. Oltre a presentare una matrice 8×8 con 64 LED programmabili rossi, verdi e blu (RGB) in grado di produrre milioni di colori, il Sense HAT è dotato di un controller joystick a cinque posizioni e sei sensori (sette nei modelli più nuovi).

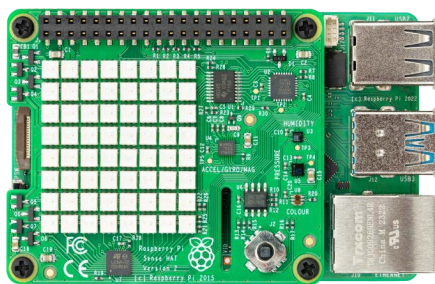


Figura 7-1 Sense HAT

- ▶ **Sensore giroscopico:** utilizzato per rilevare i cambiamenti di angolazione nel tempo, ossia la *velocità angolare*, il sensore giroscopico è in grado di rilevare la rotazione del Sense HAT su uno qualsiasi dei tre assi e la velocità con cui viene ruotato.
- ▶ **Accelerometro:** è simile al sensore giroscopico, ma anziché monitorare un angolo, misura la forza di accelerazione in più direzioni. Le letture (dati) combinate dell'accelerometro e del sensore giroscopico possono aiutarti a comprendere dove sta puntando il Sense HAT e come viene mosso.
- ▶ **Magnetometro:** misura la forza di un campo magnetico ed è un altro sensore che consente di tenere traccia dei movimenti del Sense HAT. Misurando il campo magnetico naturale della Terra il magnetometro individua la direzione del nord magnetico. Lo stesso sensore può essere usato anche per rilevare oggetti metallici e persino campi elettrici.

Questi tre sensori si trovano in un unico chip, denominato **ACCEL/GYRO/MAG** sul circuito del Sense HAT.

- ▶ **Sensore di umidità:** misura la quantità di vapore acqueo presente nell'aria, ossia l'*umidità relativa*. L'umidità relativa può variare dallo 0% (quando non c'è acqua) al 100% (quando l'aria è completamente satura). I dati sull'umidità possono anche aiutare a rilevare quando sta per piovere.
- ▶ **Sensore di pressione barometrica:** noto anche come *barometro*, misura la pressione dell'aria. La maggior parte delle persone ha familiarità con la pressione barometrica di cui si parla nelle previsioni del tempo, tuttavia il barometro ha un secondo uso segreto: consente di tenere traccia di quando si sale o si scende da una collina o una montagna, poiché più ci si allontana dal livello del mare più l'aria diventa rarefatta e la pressione si abbassa.
- ▶ **Sensore di temperatura:** misura quanto l'ambiente circostante sia caldo o freddo, tuttavia è influenzato dalla temperatura del Sense HAT, infatti se utilizzi una custodia, i valori potrebbero essere più elevati del previsto. Il Sense HAT non ha un sensore di temperatura separato, ma utilizza sensori di temperatura integrati nei sensori di umidità e di pressione barometrica. Un programma può utilizzare uno o entrambi i sensori, dipende da te.
- ▶ **Sensore di colore e luminosità:** disponibile solo su Sense HAT V2, il sensore di colore e luminosità rileva la luce circostante e ne segnala l'intensità, ideale per i progetti in cui si desidera attenuare e illuminare automaticamente i LED in base all'illuminazione della stanza. Il sensore può anche essere utilizzato per segnalare il colore della luce in ingresso. Quando progetti i tuoi esperimenti, considera che le letture vengono influenzate dalla luce proveniente dalla matrice LED del Sense HAT. Questo è l'unico sensore che non è possibile simulare utilizzando l'emulatore Sense HAT; per utilizzarlo è necessario un vero Sense HAT V2.

SENSE HAT SU RASPBERRY PI 400

Il Sense HAT è compatibile con il Raspberry Pi 400 e può essere inserito direttamente nel terminale GPIO dalla parte posteriore. Tuttavia in questo modo i LED non saranno rivolti verso di te e la scheda sarà rivolta verso il basso.

Per risolvere questo problema, è necessario un cavo di prolunga GPIO o una scheda. Tra le estensioni compatibili c'è Black HAT Hack3r range di **pimoroni.com**, puoi usare il Sense HAT con la scheda Black HAT Hack3r oppure utilizzare semplicemente il cavo a nastro a 40 pin come estensione. Controlla sempre le istruzioni del produttore per essere sicuro di collegare il cavo e il Sense HAT nel modo giusto.



Installazione del Sense HAT

Inizia rimuovendo il Sense HAT dalla confezione e assicurandoti di avere tutti i pezzi: dovrebbero essere presenti il Sense HAT stesso, quattro perni in metallo o plastica, ossia i *distanziatori*, e otto viti. Potrebbero essere presenti anche alcuni pin in metallo inseriti in una striscia di plastica nera, come i pin GPIO del Raspberry Pi; se sono presenti, spingi questa striscia con i pin rivolti verso l'alto in direzione della parte inferiore del Sense HAT finché non senti un clic.

I distanziatori sono progettati per impedire al Sense Hat di piegarsi quando usi il joystick. Il Sense HAT funzionerà anche se non li installi, tuttavia ti consentono di proteggere il Sense HAT, il Raspberry Pi e il connettore GPIO da eventuali danni.

Se utilizzi il Sense HAT con un Raspberry Pi Zero 2 W, non potrai utilizzare tutti e quattro i distanziatori. Dovrai inoltre aver saldato alcuni pin sul connettore GPIO o aver acquistato la scheda da un rivenditore che lo ha fatto per te.



ATTENZIONE

I moduli HAT (Hardware Attached on Top) devono essere collegati e rimossi dal terminale GPIO solo quando il Raspberry Pi è spento e scollegato dall'alimentazione. Mantieni sempre il modulo HAT orizzontalmente durante l'installazione e verifica che sia allineato con i pin del connettore GPIO prima di spingerlo verso il basso.

Installa i distanziatori spingendo quattro delle viti da sotto la parte inferiore del Raspberry Pi attraverso i quattro fori di montaggio in ogni angolo, quindi avvita i distanziatori sulle viti. Spingi il Sense HAT sul connettore GPIO del Raspberry Pi, assicurandoti di allinearli correttamente con i pin sottostanti e di mantenerlo in posizione orizzontale il più possibile.

Infine, avvita le rimanenti quattro viti attraverso i fori di montaggio sul Sense HAT e nei distanziatori installati in precedenza. Se installato correttamente, il Sense HAT dovrebbe risultare perfettamente orizzontale e non dovrebbe piegarsi o traballare quando si preme sul joystick.

Collega alla corrente il Raspberry Pi e vedrai i LED del Sense HAT accendersi come un arcobaleno (**Figura 7-2**), quindi spegnersi. Il tuo Sense HAT è installato.

Se vuoi rimuovere nuovamente il Sense HAT, allenta le viti superiori e sollevalo facendo attenzione a non piegare i perni sul connettore GPIO, in quanto è fissato saldamente (potrebbe essere necessario aiutarsi con un piccolo cacciavite), quindi rimuovi i distanziatori da Raspberry Pi.

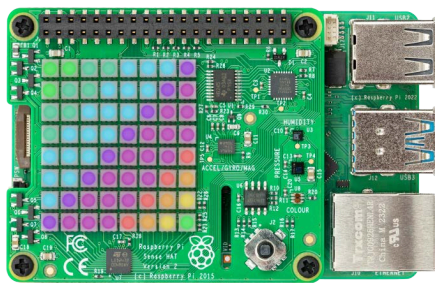


Figura 7-2 Alla prima accensione si illumina con una sequenza arcobaleno

Per programmare il Sense HAT avrai bisogno di un software che potrebbe non essere già installato. Se non riesci a trovare Scratch 3 e l'emulatore Sense HAT nella sezione **Programmazione** del menu Raspberry, leggi il Capitolo 3, *Come utilizzare Raspberry Pi* e segui le istruzioni nella sezione Recommended Software per installare Scratch 3. Segui le istruzioni nella Appendice B, *Come installare e disinstallare il software* per installare l'emulatore Sense HAT.

ESPERIENZA DI PROGRAMMAZIONE

Questo capitolo presuppone che tu sappia già utilizzare Scratch 3 o Python e l'ambiente di sviluppo integrato (IDE) Thonny. Se non l'hai ancora fatto, leggi il Capitolo 4, *Programmare con Scratch 3* o Capitolo 5, *Programmare con Python* e realizza prima i progetti illustrati in questi due capitoli.



Ciao Sense HAT!

Come in ogni progetto di programmazione che si rispetti, anche con Sense HAT è consigliato iniziare con un messaggio di benvenuto sul display LED. Se utilizzi l'emulatore Sense HAT, caricalo facendo clic sull'icona Raspberry Pi, selezionando la categoria **Programmazione** e facendo clic su **Sense HAT Emulator**.

Benvenuto da Scratch

Carica Scratch 3 dal menu Raspberry Pi. Fai clic sul pulsante **Aggiungi un'Estensione** in basso a sinistra della finestra Scratch. Fai clic sull'estensione **Raspberry Pi Sense HAT** (Figura 7-3). Si caricherà il blocco necessario per

controllare le varie funzioni di Sense HAT, incluso il display LED. Quando ne avrai bisogno, le troverai nella categoria **Raspberry Pi Sense HAT**.

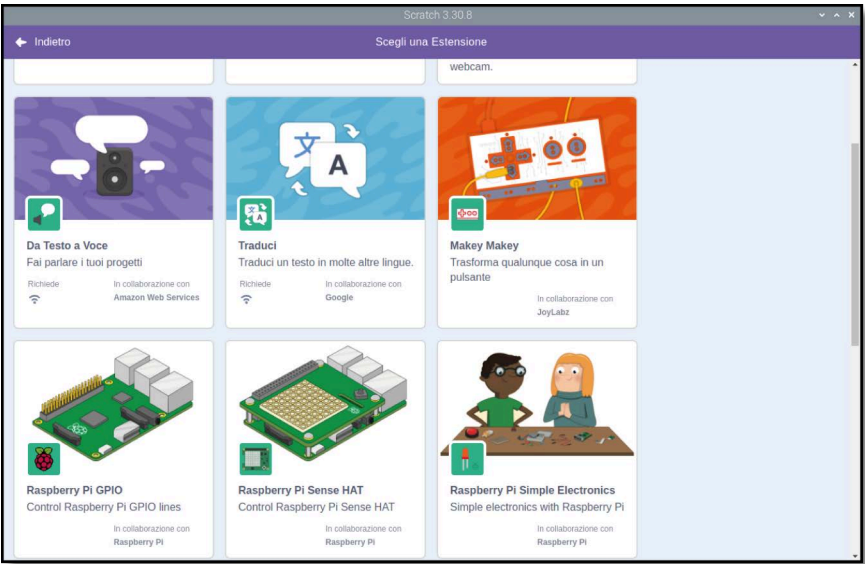

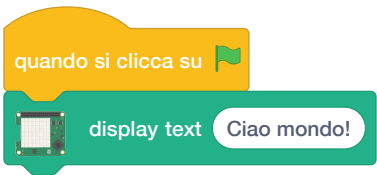



Figura 7-3 Aggiunta dell'estensione Raspberry Pi Sense HAT a Scratch 3

Inizia trascinando un blocco **quando si clicca su**  **Situazioni** nell'area script, quindi trascina un blocco **display text Ciao!** direttamente sotto di esso. Modifica il testo in modo che vi sia scritto **display text Ciao mondo!**.



Fai clic sulla bandiera verde nell'area stage e guarda il tuo Sense HAT o l'emulatore: vedrai scorrere lentamente il messaggio nella matrice LED Sense HAT, i pixel LED si accenderanno a turno per formare ciascuna lettera (**Figura 7-4**). Complimenti, il tuo programma è un successo!

Ora che sai come far scorrere un semplice messaggio, è il momento di controllare come viene visualizzato. Oltre a poter modificare il messaggio, puoi alterarne la rotazione, ossia la direzione di visualizzazione del messaggio. Trascina un blocco **set rotation to 0 degrees** dalla palette e inseriscilo sotto a **quando si clicca su**  e sopra a **display text Ciao mondo!**. Fai clic sulla freccia rivolta verso il basso accanto a **0** e porta il valore a **90**.

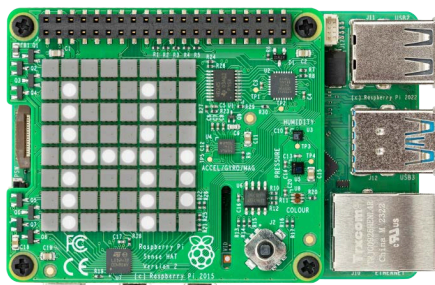


Figura 7-4 Il tuo messaggio scorre nella matrice LED

Fai clic sulla bandiera verde e vedrai lo stesso messaggio di prima, ma anziché scorrere da sinistra a destra, scorrerà dal basso verso l'alto (**Figura 7-5**): dovrai girare la testa o il Sense HAT per leggerlo.

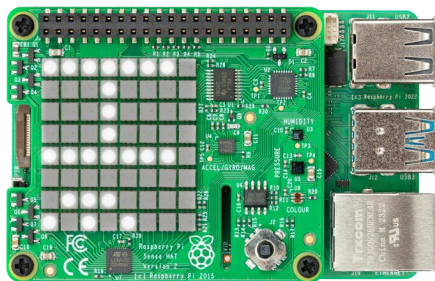


Figura 7-5 Questa volta il messaggio scorre verticalmente

Ora porta nuovamente la rotazione a 0, quindi trascina un blocco **set colour** tra **set rotation to 0 degrees** e **display text Ciao mondo!**. Fai clic sul colore alla fine del blocco per visualizzare il selettore di colori di Scratch e trovare un bel colore giallo brillante, poi fai clic sulla bandiera verde per vedere come è cambiato l'output del programma (**Figura 7-6**).

Infine, trascina un blocco **set background** tra **set colour** e **display text Ciao mondo!**, quindi fai clic sul colore per visualizzare di nuovo il selettore di colori. In questo caso la scelta del colore non influisce sui LED che compongono il messaggio, ma sugli altri, ossia quelli di sfondo. Trova un bel colore blu, poi fai di nuovo clic sulla bandiera verde: questa volta il tuo messaggio sarà giallo brillante su sfondo blu. Prova a cambiare i colori per trovare la tua combinazione preferita, non tutti i colori si abbinano bene tra loro.

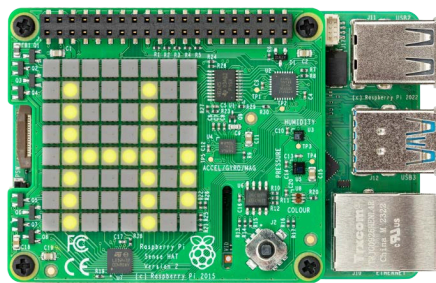


Figura 7-6 Modifica del colore del testo

Oltre a poter far scorrere interi messaggi, puoi mostrare singole lettere. Trascina il blocco `display text Ciao mondo!` fuori dall'area script per cancellarlo e sostituisilo trascinando un blocco `display character A` nell'area script.

Fai clic sulla bandiera verde e vedrai la differenza: questo blocco mostra solo una lettera alla volta e la lettera rimane sul Sense HAT senza scorrere o scomparire finché non inserisci altre istruzioni. A questo blocco si applicano gli stessi blocchi di controllo del colore del blocco `display text`: prova a cambiare il colore della lettera selezionando il rosso (Figura 7-7).

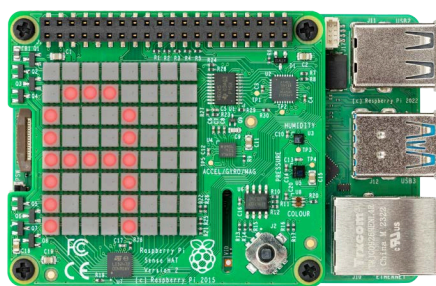


Figura 7-7 Visualizzazione di una sola lettera



SFIDA: RIPETERE IL MESSAGGIO

Riesci a usare le tue nozioni sulle ripetizioni per ottenere un messaggio che scorre e si ripete? Puoi creare un programma che scrive una parola lettera per lettera con diversi colori?

Benvenuto da Python

Carica Thonny facendo clic sull'icona Raspberry, selezionando **Programmazione** e facendo clic su **Thonny**. Se utilizzi l'emulatore Sense HAT e viene nascosto dalla finestra Thonny, fai clic e tieni premuto il pulsante del mouse sulla barra del titolo di una delle due finestre (in alto, blu) e trascina per spostarla sul desktop fino a visualizzare entrambe le finestre.

CAMBIO DI RIGA IN PYTHON

Il codice Python scritto per un Sense HAT fisico viene eseguito anche sull'emulatore Sense HAT e viceversa, è necessario apportare solo una modifica. Se utilizzi l'emulatore Sense HAT con Python dovrai modificare la riga **from sense_hat import SenseHat** in tutti i programmi di questo capitolo in **from sense_emu import SenseHat**. Se poi vuoi eseguirli su un Sense HAT fisico, ripristina la riga precedente.



Per utilizzare Sense HAT o l'emulatore in un programma Python dovrai importare la libreria Sense HAT. Digita quanto segue nell'area script, ricordandoti di utilizzare **sense_emu** (al posto di **sense_hat**) se utilizzi un emulatore Sense HAT:

```
from sense_hat import SenseHat
sense = SenseHat()
```

La libreria Sense HAT presenta una funzione semplice per scrivere un messaggio, formattarlo in modo che possa essere visualizzato sul display LED e farlo scorrere in modo fluido. Digita quanto segue:

```
sense.show_message("Ciao mondo!")
```

Salva il tuo programma come **Ciao Sense HAT.py** e fai clic sul pulsante **Esegui**. Vedrai scorrere lentamente il messaggio nella matrice LED Sense HAT, i pixel LED si accenderanno a turno per formare ciascuna lettera (**Figura 7-8**). Complimenti, il tuo programma è un successo!

La funzione **show_message()** consente di effettuare altre azioni. Torna al tuo programma e modifica l'ultima riga come segue:

```
sense.show_message("Ciao mondo!", text_colour=(255, 255, 0),
                    back_colour=(0, 0, 255), scroll_speed=(0.05))
```

Queste istruzioni aggiuntive, separate da virgole, sono note come *parametri* e controllano vari aspetti della funzione **show_message()**. Il più semplice è **scroll_speed=()**, che cambia la velocità con cui il messaggio scorre sullo

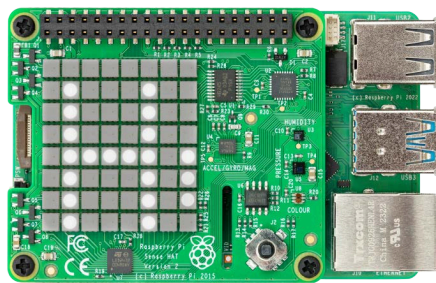


Figura 7-8 Far scorrere un messaggio nella matrice LED

schermo. Inserendo come valore 0.05 la velocità di scorrimento all'incirca raddoppia. Più elevato è il numero, minore è la velocità.

I parametri `text_colour=()` e `back_colour=()` (con ortografia inglese britannico, a differenza della maggior parte delle istruzioni Python) consentono di impostare rispettivamente il colore della scritta e dello sfondo. Tuttavia non accettano nomi di colori: dovrai quindi indicare il colore desiderato con un numero a tre cifre. Il primo numero rappresenta la quantità di rosso nel colore (dove 0 indica l'assenza di rosso e 255 il più rosso possibile), il secondo la quantità di verde, mentre il terzo la quantità di blu. Insieme, questi valori sono noti come *RGB*, che sta per red, green, blue (rosso, verde e blu).

Fai clic sull'icona **Esegui** e guarda il Sense HAT: questa volta, il messaggio scorrerà molto più velocemente e sarà giallo brillante su sfondo blu (**Figura 7-9**). Prova a modificare i parametri per trovare la perfetta combinazione di velocità e colore.

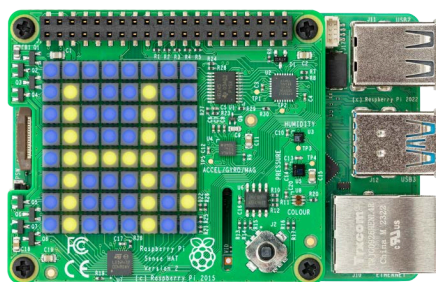


Figura 7-9 Modifica del colore del messaggio e dello sfondo

Se vuoi usare nomi più comprensibili al posto dei valori RGB per impostare i colori, dovrai creare delle variabili. Sopra alla riga `sense.show_message()` aggiungi quanto segue:

```
yellow = (255, 255, 0)
blue = (0, 0, 255)
```

Torna alla riga `sense.show_message()` e modificala come segue:

```
sense.show_message("Ciao mondo!", text_colour=(yellow),
                   back_colour=(blue), scroll_speed=(0.05))
```

Fai di nuovo clic sull'icona **Esegui** e vedrai che non è cambiato nulla: il messaggio è ancora in giallo su sfondo blu. Questa volta, però, avrai usato i nomi delle variabili per rendere il codice più leggibile: invece di una stringa di numeri, il codice indica il colore che sta impostando. Puoi definire tutti i colori che vuoi: prova ad aggiungere una variabile chiamata **red** con i valori 255, 0 e 0; una variabile chiamata **white** con i valori 255, 255, 255; infine una variabile chiamata **black** con i valori 0, 0 e 0.

Oltre a poter far scorrere interi messaggi, puoi mostrare singole lettere. Cancella la riga `sense.show_message()` digita al suo posto la seguente riga:

```
sense.show_letter("A")
```

Fai clic su **Esegui** e vedrai la lettera "A" apparire sul display del Sense HAT. Questa volta rimarrà ferma, infatti le singole lettere, a differenza dei messaggi, non scorrono automaticamente. Puoi controllare `sense.show_letter()` con gli stessi parametri di colore di `sense.show_message()`: prova a cambiare il colore della lettera passando al rosso (Figura 7-10).

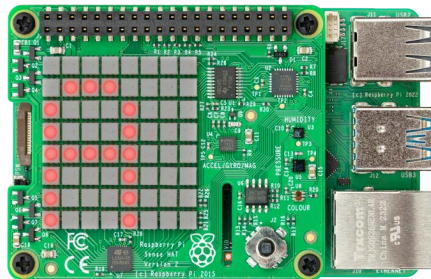


Figura 7-10 Visualizzazione di una sola lettera



SFIDA: RIPETERE IL MESSAGGIO

Riesci a usare le tue nozioni sulle ripetizioni per ottenere un messaggio che scorre e si ripete? Puoi creare un programma che scrive una parola lettera per lettera con diversi colori? Quanto velocemente si può far scorrere un messaggio?

Prossimi passaggi: disegnare con le luci

Il display LED del Sense HAT non è solo per i messaggi: puoi usarlo anche per rappresentare immagini. Ogni LED può essere trattato come un singolo pixel, abbreviazione che sta per *picture element*, ovvero *elemento dell'immagine*, in un'immagine a tua scelta, consentendoti di ravvivare i tuoi programmi con immagini e persino animazioni.

Per creare disegni dovrai cambiare i singoli LED. Per fare questo, dovrai apprendere come è disposta la matrice LED del Sense HAT per scrivere un programma che accenda o spenga i LED corretti.

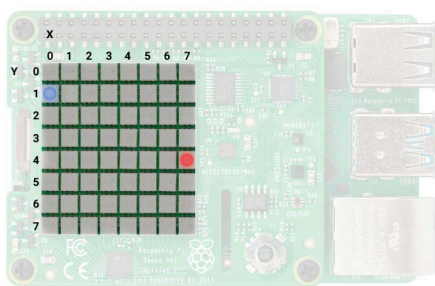


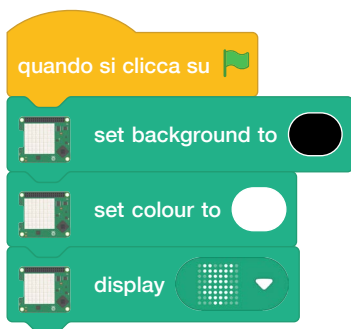
Figura 7-11 La matrice LED coordina il sistema

Ci sono otto LED in ogni riga del display e otto in ogni colonna (**Figura 7-11**). Tuttavia quando si contano i LED, come nella maggior parte dei linguaggi di programmazione, si dovrebbe iniziare da 0 e finire con 7. Il primo LED è in alto a sinistra, l'ultimo in basso a destra. Utilizzando i numeri delle righe e delle colonne, puoi trovare le *coordinate* di qualsiasi LED sulla matrice. Nella matrice raffigurata, il LED blu si trova in corrispondenza delle coordinate (0, 1), mentre il LED rosso alle coordinate (7, 4). La coordinata dell'asse X viene prima e aumenta andando verso destra, seguita dall'asse Y, che aumenta verso il basso.

Quando pianifichi le immagini da disegnare su Sense HAT, ti consigliamo di disegnarle a mano su carta millimetrata o a quadretti oppure in un foglio di calcolo come LibreOffice Calc.

Immagini in Scratch

Salva il programma appena creato, quindi crea un nuovo progetto su Scratch. Se hai lavorato sui progetti di questo capitolo, Scratch 3 manterrà caricata l'estensione Raspberry Pi Sense HAT; se invece hai chiuso e riaperto Scratch 3 dall'ultimo progetto, carica l'estensione usando il pulsante **Aggiungi un'Estensione**. Inizia trascinando un blocco **quando si clicca su** **Situazioni** nell'area codice, quindi trascina i blocchi **set background** e **set colour** direttamente sotto di esso. Modifica entrambi per impostare il colore di sfondo su nero e il colore su bianco: puoi ottenere il nero facendo scorrere i cursori **Luminosità** e **Saturazione** fino a raggiungere lo 0 e il bianco facendo scorrere il cursore **Luminosità** fino a 100 e **Saturazione** fino a 0. Dovrai eseguire quest'operazione all'avvio di ogni programma Sense HAT, altrimenti Scratch utilizzerà gli ultimi colori che hai scelto, anche se li hai selezionati in un programma diverso. Infine trascina un blocco **display raspberry** in fondo al tuo programma.



Fai clic sulla bandiera verde: vedrai i LED del Sense HAT accendersi formando un lampone (Figura 7-12).

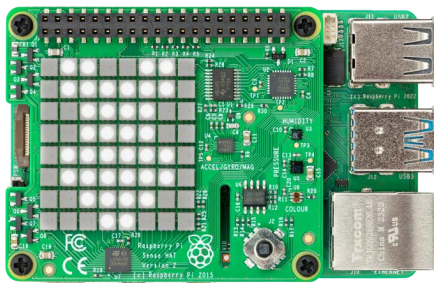


Figura 7-12 Visualizzazione della forma del lampone con Scratch



ATTENZIONE

Quando i LED sono di colore bianco intenso, come in questo esempio di codice, evita di guardarli direttamente: sono abbastanza luminosi da danneggiare gli occhi.

Potrai ottenere molte forme oltre a quella del lampone. Fai clic sulla freccia in basso accanto al lampone per attivare la modalità disegno: è possibile fare clic su qualsiasi LED per accenderlo o spegnerlo, mentre i due pulsanti in basso accendono o spengono tutti i LED. Prova a realizzare il tuo disegno, poi fai clic sulla freccia verde per vederlo sul Sense HAT. Prova anche a cambiare il colore del disegno e dello sfondo utilizzando i blocchi indicati in precedenza.

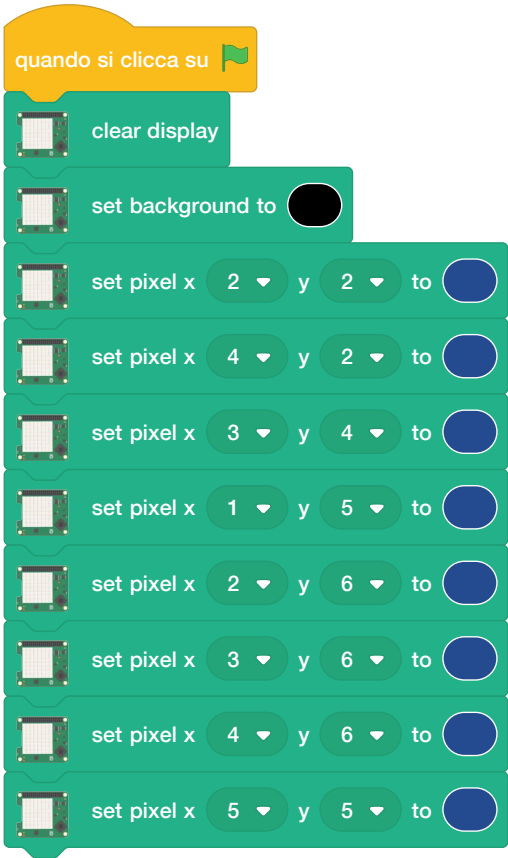
Quando hai finito, trascina i tre blocchi nella palette per cancellarli e metti un blocco `set clear display` sotto a `quando si clicca su`, quindi fai clic sulla bandiera verde e tutti i LED si spegneranno.

Per realizzare un'immagine devi essere in grado di controllare i singoli pixel e assegnare loro colori diversi. Puoi farlo concatenando i blocchi `display raspberry` modificati con blocchi `set colour` oppure puoi gestire ogni singolo pixel. Prova a creare la tua versione della matrice LED mostrata nell'esempio all'inizio di questa sezione. Due LED specificamente selezionati si accendono in rosso e in blu. Lascia il blocco `clear display` all'inizio del programma e trascina un blocco `set background` al di sotto. Modifica il colore del blocco `set background` su nero, poi trascina due blocchi `set pixel x 0 y 0` sotto di esso. Infine modifica questi blocchi come mostrato.

Fai clic sulla bandiera verde, vedrai i LED accendersi in corrispondenza dell'immagine della matrice (**Figura 7-13**). Complimenti, ora puoi controllare i singoli LED!



Modifica i blocchi di pixel come nel prossimo esempio e trascinane altri sul fondo fino a ottenere il seguente programma.



Prima di fare clic sulla bandiera verde, guarda se riesci a indovinare quale immagine apparirà in base alle coordinate della matrice LED che hai usato, poi esegui il tuo programma e vedi se hai ragione!

SFIDA: NUOVE IMMAGINI

Puoi creare altre immagini? Procurati carta millimetrata per pianificare l'immagine che vuoi ottenere disegnandola a mano. Riesci a disegnare l'immagine sul tuo Sense HAT e a far cambiare i colori?



Immagini in Python

Avvia un nuovo programma e digita quanto segue su Thonny, quindi salva come disegno Sense HAT, ricordandoti di utilizzare `sense_emu` (al posto di `sense_hat`) se utilizzi un emulatore:

```
from sense_hat import SenseHat
sense = SenseHat()
```

Per utilizzare il Sense HAT devi inserire entrambe le righe nel tuo programma. Successivamente digita:

```
sense.clear(255, 255, 255)
```

Fai clic sull'icona **Esegui** senza guardare direttamente i LED del Sense HAT: dovresti comunque vederli diventare tutti di un bianco brillante (**Figura 7-13**), per questo non devi guardarli direttamente quando esegui il programma.

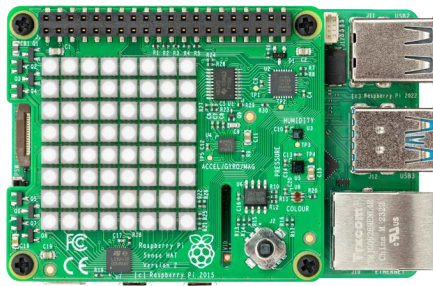


Figura 7-13 Accensione di tutti i LED



ATTENZIONE

Quando i LED sono di colore bianco intenso evita di guardarli direttamente: sono abbastanza luminosi da danneggiare gli occhi.

Il comando `sense.clear()` è progettato per spegnere i LED di qualsiasi programmazione precedente, ma accetta parametri di colore RGB, il che significa che si può ottenere qualsiasi colore. Modifica la riga così:

```
sense.clear(0, 255, 0)
```

Fai clic su **Esegui** e il Sense HAT diventerà verde brillante (**Figura 7-14**). Sperimenta con colori diversi oppure aggiungi le variabili dei nomi dei colori che

hai creato per il tuo programma Hello World (Ciao mondo) per agevolare la lettura.

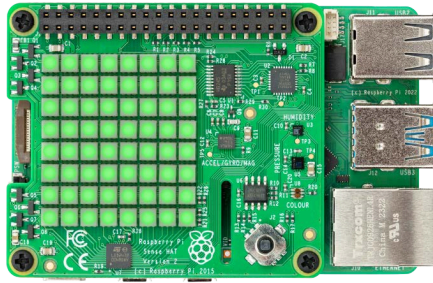


Figura 7-14 La matrice LED si illumina di verde brillante

Per cancellare i LED, è necessario utilizzare i valori RGB per il nero: 0 rosso, 0 blu e 0 verde. Ma esiste un modo ancora più semplice. Modifica la riga del programma così:

`sense.clear()`

Il Sense HAT diventerà nero, in quanto se non si inserisce nulla tra le parentesi della funzione **`sense.clear()`**, i LED diventano neri, ossia vengono spenti (**Figura 7-15**). Questa è la funzione da utilizzare quando è necessario spegnere completamente i LED nei programmi.

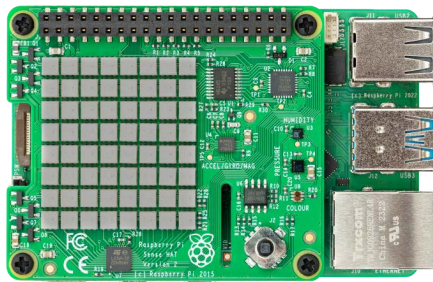


Figura 7-15 Utilizza la funzione **`sense.clear`** per spegnere tutti i LED

Per creare la tua versione della matrice LED mostrata nell'esempio precedente in questo capitolo, con due LED specifici illuminati in rosso e blu, aggiungi le seguenti righe al programma dopo **`sense.clear()`**:


```
sense.set_pixel(0, 1, (0, 0, 255))
sense.set_pixel(7, 4, (255, 0, 0))
```

I primi due numeri sono la posizione del pixel nella matrice, con l'asse X (trasversale) seguito dall'asse Y (discendente). La seconda serie di numeri, circondata da parentesi, sono i valori RGB del colore del pixel. Fai clic sul pulsante **Esegui** per vederne l'effetto: due LED sul Sense HAT si accenderanno, come mostrato in **Figura 7-11**.

Elimina queste due righe e digita quanto segue:

```
sense.set_pixel(2, 2, (0, 0, 255))
sense.set_pixel(4, 2, (0, 0, 255))
sense.set_pixel(3, 4, (100, 0, 0))
sense.set_pixel(1, 5, (255, 0, 0))
sense.set_pixel(2, 6, (255, 0, 0))
sense.set_pixel(3, 6, (255, 0, 0))
sense.set_pixel(4, 6, (255, 0, 0))
sense.set_pixel(5, 5, (255, 0, 0))
```

Prima di fare clic su **Esegui**, guarda le coordinate e confrontale con la matrice: riesci a indovinare l'immagine che otterrai? Fai clic su **Esegui** per scoprire se hai ragione!

Disegnare un'immagine dettagliata utilizzando la funzione `set_pixel()` per gestire ogni singolo pixel richiede molto tempo. Per lavorare con più velocità, modifica più pixel contemporaneamente. Cancella tutte le righe `set_pixel()` e digita quanto segue:

```
g = (0, 255, 0)
b = (0, 0, 0)
creeper_pixels = [
    g, g, g, g, g, g, g, g,
    g, g, g, g, g, g, g, g,
    g, b, b, g, g, b, b, g,
    g, b, b, g, g, b, b, g,
    g, g, g, b, b, g, g, g,
    g, g, b, b, b, b, g, g,
    g, g, b, b, b, b, g, g,
    g, g, b, g, g, b, g, g
]
sense.set_pixels(creeper_pixels)
```

C'è molto da fare, ma fai clic su **Esegui** per vedere se riconosci un piccolo creeper. Le prime due linee creano due variabili per i colori: verde e nero. Per

rendere il codice più facile da scrivere e leggere, le variabili sono lettere singole: "g" per il verde e "b" per il nero.

Il blocco di codice successivo crea una variabile che contiene i valori di colore per i 64 pixel della matrice LED, separati da virgole e racchiusi tra parentesi quadre. Anziché i numeri usa le variabili di colore create prima: guarda bene, tenendo a mente che "g" indica il verde e "b" il nero, e vedrai l'immagine apparire (Figura 7-16).

Infine `sense.set_pixels(creeper_pixels)` prende la variabile e usa la funzione `sense.set_pixels()` per disegnare su tutta la matrice. È molto più semplice che cercare di disegnare pixel per pixel.

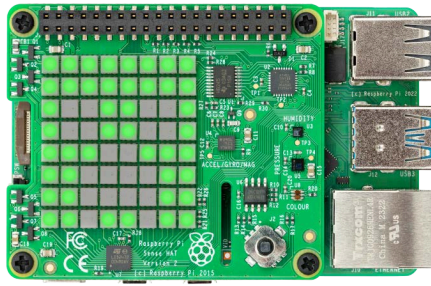


Figura 7-16 Visualizzazione di un'immagine sulla matrice

Puoi anche ruotare e capovolgere le immagini, sia per mostrarle nel modo giusto quando il Sense HAT è girato, sia per creare semplici animazioni da un'unica immagine asimmetrica.

Inizia modificando la variabile `creeper_pixels` per chiudere l'occhio sinistro sostituendo i quattro pixel "b" con "g", iniziando dai primi due sulla terza riga e poi passando ai primi due sulla quarta:

```
creeper_pixels = [  
    g, g, g, g, g, g, g, g,  
    g, g, g, g, g, g, g, g,  
    g, g, g, g, g, b, b, g,  
    g, g, g, g, g, b, b, g,  
    g, g, b, b, b, g, g, g,  
    g, g, b, b, b, b, g, g,  
    g, g, b, b, b, b, g, g,  
    g, g, b, g, g, b, g, g  
]
```

Fai clic su **Esegui** e vedrai l'occhio sinistro del creeper chiudersi (**Figura 7-17**). Per realizzare un'animazione torna all'inizio del programma e inserisci la riga:

```
from time import sleep
```

Quindi vai alla fine e digita:

```
while True:
    sleep(1)
    sense.flip_h()
```

Fai clic su **Esegui** per vedere il creeper che chiude e riapre gli occhi, uno alla volta.

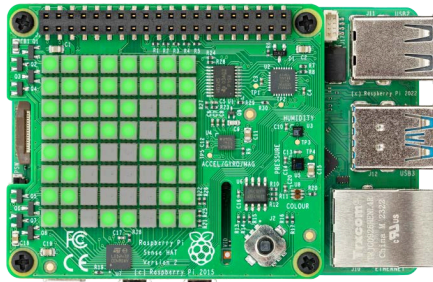


Figura 7-17 Una semplice animazione a due fotogrammi

La funzione `flip_h()` capovolge un'immagine sull'asse orizzontale. Se desideri capovolgere un'immagine sull'asse verticale, sostituisci `sense.flip_h()` con `sense.flip_v()`. Puoi anche ruotare un'immagine di 0, 90, 180 o 270 gradi utilizzando `sense.set_rotation(90)` e modificando il numero in base a quanti gradi desideri ruotare l'immagine. Prova questa funzione per vedere il creeper girare anziché chiudere e aprire gli occhi.



SFIDA: NUOVE IMMAGINI

Riesci a creare altre immagini e animazioni? Procurati carta millimetrata per pianificare l'immagine che vuoi ottenere disegnandola a mano per rendere più semplice la scrittura della variabile. Sei in grado di realizzare un'immagine e far cambiare i colori? Ricorda: puoi modificare le variabili dopo averle già utilizzate.

Percepire il mondo che ti circonda

La vera potenza del Sense HAT risiede nei suoi sensori. Questi consentono di rilevare valori che vanno dalla temperatura all'accelerazione e di utilizzare le informazioni che ti forniscono nei tuoi programmi.

EMULAZIONE DEI SENSORI


Se utilizzi l'emulatore Sense HAT dovrai abilitare la simulazione dei sensori inerziali e ambientali: nell'emulatore, fai clic su **Edit**, poi su **Preferences**, quindi spunta le opzioni se non sono già spuntate. Nello stesso menu scegli **180°..360°|0°..180°** sotto **Orientation Scale** per assicurarti che i numeri nell'emulatore corrispondano a quelli riportati da Scratch e Python, quindi fai clic sul pulsante di chiusura.

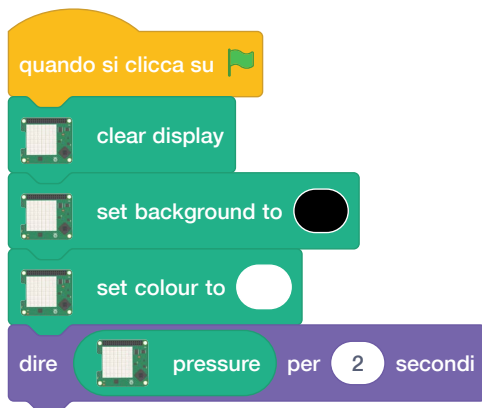


Rilevamento ambientale

I sensori di pressione barometrica, umidità e temperatura sono sensori ambientali che rilevano i valori dell'ambiente in cui si trova il Sense HAT.

Rilevamento ambientale in Scratch

Avvia un nuovo programma in Scratch, salvando il precedente se lo desideri, e aggiungi l'estensione **Raspberry Pi Sense HAT** se non è già presente. Inizia trascinando un blocco **quando si clicca su**  **Situazioni** nell'area codice, quindi trascina un blocco **clear display** sotto di esso e poi un blocco **set background to black** sotto quello precedente. Quindi aggiungi un blocco **set colour to white** e utilizza i cursori **Luminosità** e **Saturazione** per scegliere il colore corretto. Ti consigliamo di farlo all'inizio della programmazione, così sarai certo che il Sense HAT non mostri pattern di vecchi programmi e potrai essere sicuro dei colori scelti. Trascina un blocco **dire Ciao! per 2 secondi** **Aspetto** direttamente sotto i blocchi già presenti. Per effettuare una lettura dal sensore di pressione, cerca il blocco **pressione** nella categoria **Raspberry Pi Sense HAT** e trascinalo sopra la parola "**Ciao!**" nel tuo blocco **dire Ciao! per 2 secondi**.



Fai clic sulla bandiera verde e il gatto di Scratch ti dirà la pressione in *millibar* rilevata dal sensore. Il messaggio scomparirà dopo due secondi: prova a soffiare sul Sense HAT (o a spostare verso l'alto il cursore **Pressure** nell'emulatore) e a fare clic sulla bandiera verde per eseguire nuovamente il programma; questa volta dovresti vedere un valore più alto (Figura 7-18).

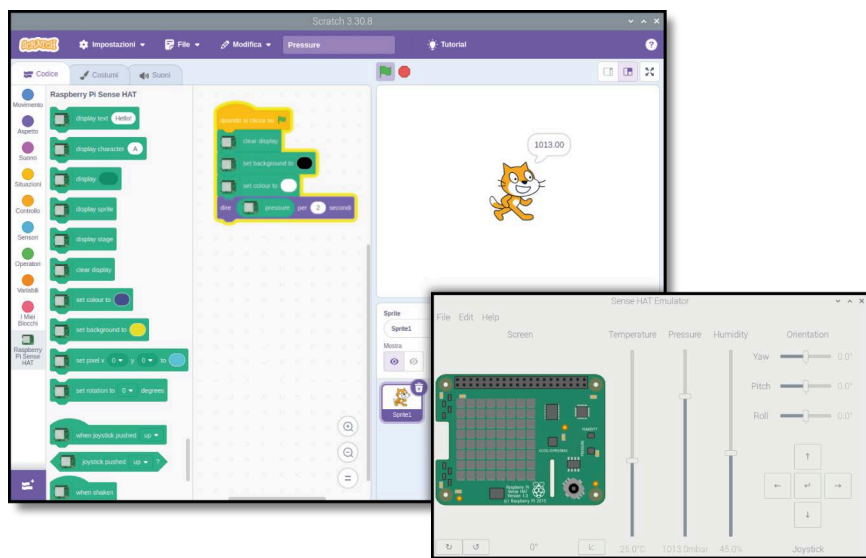


Figura 7-18 Visualizzazione della lettura del sensore di pressione



MODIFICA DEI VALORI

Se utilizzi l'emulatore Sense HAT puoi modificare i valori indicati da ciascun sensore mediante cursori e pulsanti. Prova a far scorrere il sensore di pressione verso il basso e fai clic nuovamente sulla bandiera verde.

Per passare al sensore di umidità, cancella il blocco **pressure** e sostituisilo con **humidity**. Esegui nuovamente il programma e vedrai l'umidità relativa attuale della stanza. Anche in questo caso puoi provare ad eseguirlo di nuovo mentre soffi sul Sense HAT (o spostando il cursore **Humidity** dell'emulatore verso l'alto) per modificare la lettura (Figura 7-19); il respiro è infatti particolarmente umido.

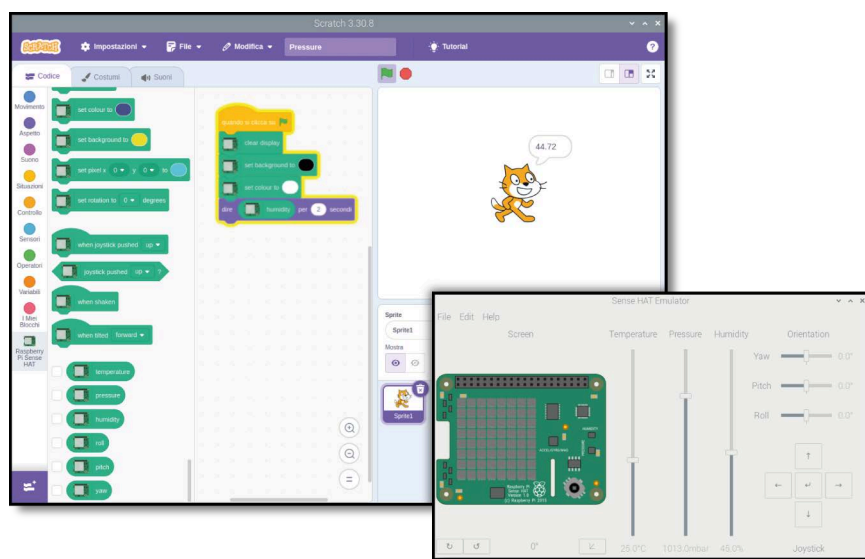


Figura 7-19 Visualizzazione della lettura del sensore di umidità

Per utilizzare il sensore di temperatura, dovrai solamente cancellare il blocco **humidity** e sostituirlo con **temperature**, quindi eseguire nuovamente il programma. Visualizzerai la temperatura in gradi Celsius (Figura 7-20). Tuttavia, questa potrebbe non essere la temperatura esatta della tua stanza: il Raspberry Pi genera calore mentre è in funzione e così anche il Sense HAT e i suoi sensori si riscaldano.

SFIDA: SCORRIMENTO E RIPETIZIONE

Riesci a modificare il programma per effettuare una lettura da ciascuno dei sensori a turno, quindi farli scorrere attraverso la matrice LED piuttosto che stamparli nell'area stage? Puoi far eseguire in modo ciclico il programma in modo che stampi continuamente le condizioni ambientali istantanee?



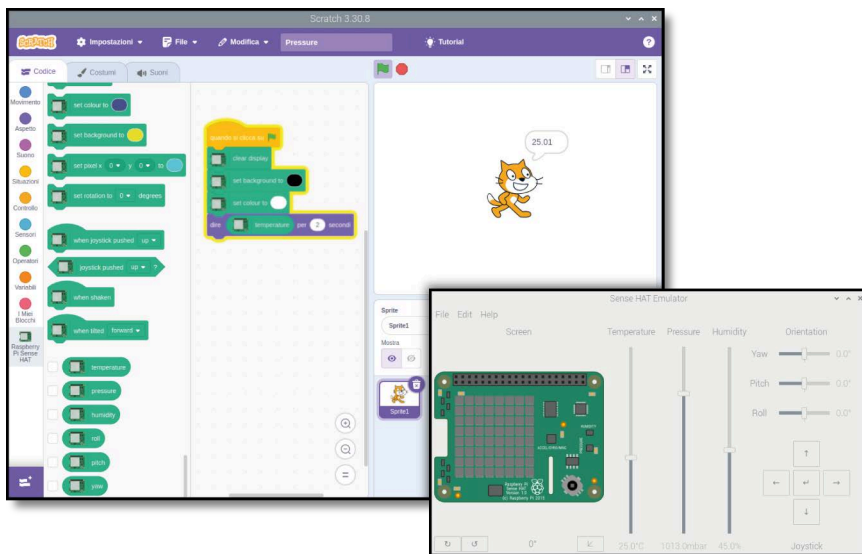


Figura 7-20 Visualizzazione della lettura del sensore di temperatura

Rilevamento ambientale in Python

Per iniziare a rilevare le letture dei sensori, crea un nuovo programma in Thonny e salvalo come **Sensori Sense HAT.py**. Digita quanto segue nell'area dello script (dovrai farlo ogni volta che usi il Sense HAT) e ricorda di usare **sense_emu** se stai usando l'emulatore:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()
```

È sempre una buona idea includere **sense.clear()** all'inizio dei programmi nel caso in cui il Sense HAT mostri ancora qualcosa dell'ultimo programma che ha eseguito.

Per effettuare una lettura dal sensore di pressione, digita:

```
pressure = sense.get_pressure()
print(pressure)
```

Fai clic su **Esegui** e vedrai un numero in Python shell in fondo alla finestra di Thonny. Questa è la lettura della pressione dell'aria in *millibar* rilevata dal sensore di pressione barometrica (**Figura 7-21**).

MODIFICA DEI VALORI



Se utilizzi l'emulatore Sense HAT puoi modificare i valori indicati da ciascun sensore mediante cursori e pulsanti. Prova a far scorrere il sensore di pressione verso il basso e fai clic nuovamente su **Esegui**: dovresti vedere un valore più alto.

Prova a soffiare sul Sense HAT (o a spostare verso l'alto il cursore **Pressure** nell'emulatore) e a fare nuovamente clic sull'icona **Esegui**: dovresti vedere un valore più alto.

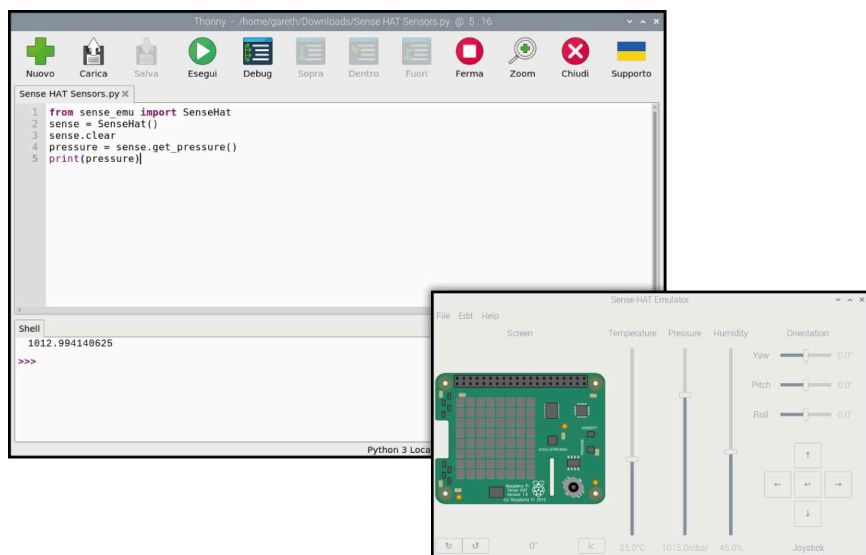


Figura 7-21 Stampa della lettura della pressione dal Sense HAT

Per passare al sensore di umidità, cancella le ultime due righe di codice e sostituiscile con:

```
humidity = sense.get_humidity()
print(humidity)
```

Fai clic su **Esegui** e vedrai un numero in Python shell: questa volta è l'umidità relativa della tua stanza indicata con una percentuale. Anche in questo caso puoi soffiare sul Sense HAT (o spostare il cursore **Humidity** dell'emulatore verso l'alto) per modificare la lettura (**Figura 7-22**); il respiro è infatti particolarmente umido.

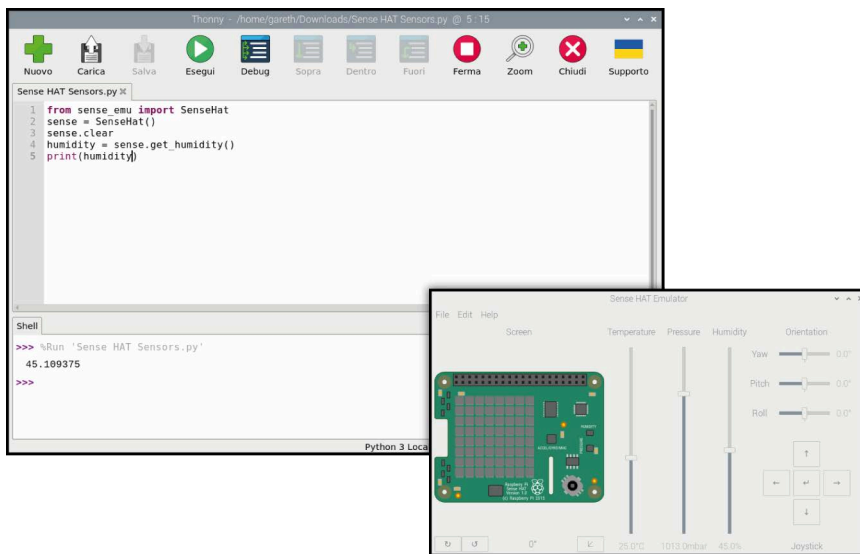


Figura 7-22 Visualizzazione della lettura del sensore di umidità

Per passare al sensore di temperatura, cancella le ultime due righe di codice e sostituiscile con:

```
temp = sense.get_temperature()
print(temp)
```

Fai clic di nuovo su **Esegui** e visualizzerai la temperatura in gradi Celsius (**Figura 7-23**). Tuttavia, questa potrebbe non essere la temperatura esatta della tua stanza: il Raspberry Pi genera calore mentre è in funzione e così anche il Sense HAT e i suoi sensori si riscaldano.

Normalmente il Sense HAT riporta la temperatura in base alla lettura del sensore di temperatura integrato nel sensore di umidità. Se invece desideri utilizzare la lettura del sensore di pressione, devi utilizzare `sense.get_temperature_from_pressure()`. È possibile unire le due letture per ottenere una media, che è generalmente più precisa rispetto all'utilizzo dei singoli sensori. Per farlo, cancella le ultime due righe del programma e sostituiscile con:

```
htemp = sense.get_temperature()
ptemp = sense.get_temperature_from_pressure()
temp = (htemp + ptemp) / 2
print(temp)
```

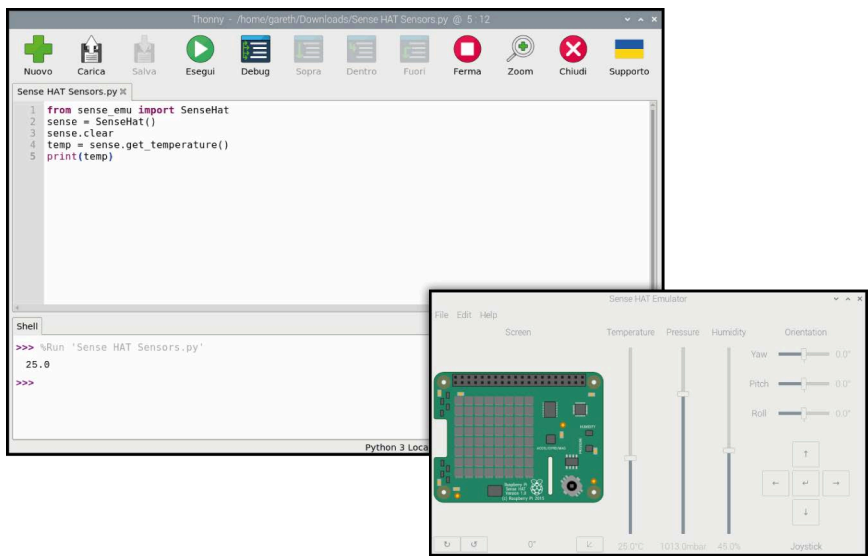


Figura 7-23 Visualizzazione della lettura della temperatura

Fai clic sull'icona **Esegui** e vedrai un numero nella console Python (**Figura 7-24**). In questo caso si basa sulle letture di entrambi i sensori sommate e divise per due (il numero di letture) per ottenere una media. Se utilizzi l'emulatore, tutti e tre i metodi (umidità, pressione e media) mostreranno lo stesso numero.

SFIDA: SCORRIMENTO E RIPETIZIONE

Riesci a modificare il programma per effettuare una lettura da ciascuno dei sensori a turno, quindi farli scorrere attraverso la matrice LED piuttosto che stamparli in shell? Puoi far eseguire in modo ciclico il programma in modo che stampi continuamente le condizioni ambientali istantanee?



Rilevamento inerziale

Il sensore giroscopico, l'accelerometro e il magnetometro insieme formano la cosiddetta *unità di misura inerziale (IMU)*. Tecnicamente parlando, questi sensori rilevano i valori dall'ambiente circostante proprio come i sensori ambientali (il magnetometro, per esempio, misura l'intensità del campo magnetico), tuttavia sono usati per raccogliere i dati sul movimento del Sense HAT stesso. L'IMU è la somma di più sensori. Mentre alcuni linguaggi di

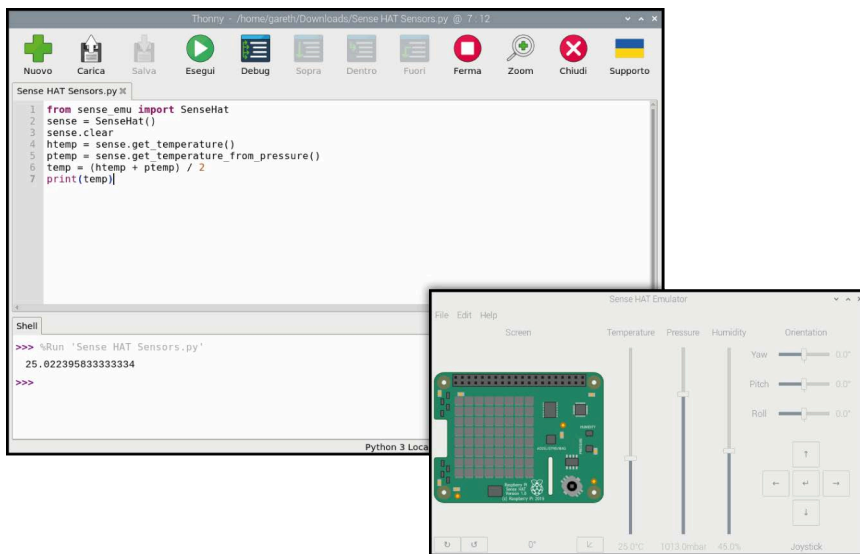



Figura 7-24 Temperatura basata sulle letture di entrambi i sensori

programmazione permettono di effettuare letture da ogni sensore in modo indipendente, altri forniscono solo una lettura combinata.

Per comprendere l'IMU è necessario apprendere come si muovono gli oggetti. Il Sense HAT e il Raspberry Pi a cui è collegato possono muoversi lungo tre assi spaziali: da un lato all'altro sull'asse X, in avanti e indietro sull'asse Y e su e giù sull'asse Z (**Figura 7-25**). Può anche ruotare lungo questi tre assi, ma i nomi cambiano: la rotazione sull'asse X è chiamata *rollio*, sull'asse Y *beccheggio* e sull'asse Z *imbardata*. Quando ruoti il Sense HAT sull'asse corta modifichi il suo beccheggio, sull'asse lunga il rollio e se lo ruoti mantenendolo in posizione orizzontale sul tavolo modifichi la sua imbardata. È come se fosse un aereo: quando sta decollando, aumenta il suo beccheggio per salire, quando ruota su se stesso, sta ruotando attorno all'asse di rollio, mentre quando usa il timone per virare senza rollio, è l'imbardata.

Rilevamento inerziale in Scratch

Avvia un nuovo programma in Scratch e carica l'estensione **Raspberry Pi Sense HAT**, se non è già stata caricata. Avvia il programma come prima: inizia trascinando un blocco **quando si clicca su**  **Situazioni** nell'area codice, trascina un blocco **clear display** sotto di esso, quindi trascina e modifica un blocco **set background to black** e **set colour to white**. Infine trascina un blocco **per sempre** in fondo al tuo programma e inserisci un blocco **dire Ciao!**. Per

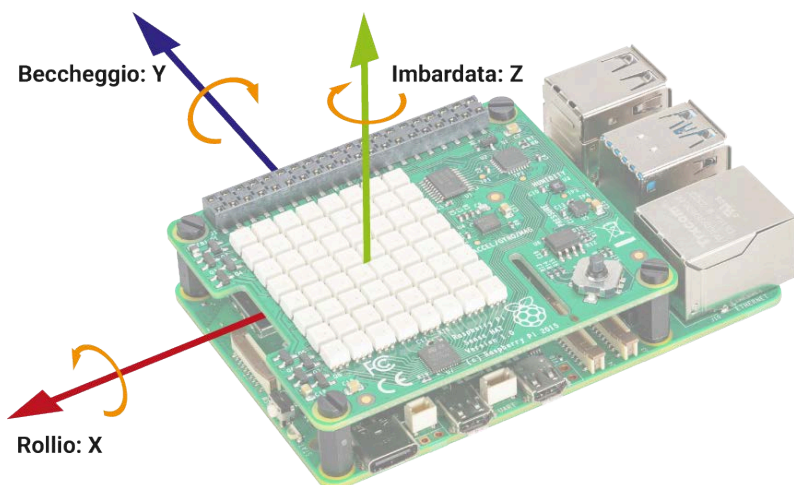


Figura 7-25 Gli assi spaziali dell'IMU di Sense HAT

mostrare una lettura per i tre assi IMU (beccheggio, rollio e imbardata), dovrai aggiungere dei blocchi **unione** Operatori oltre ai blocchi **Raspberry Pi Sense HAT** corrispondenti. Ricordati di includere spazi e virgole, in modo che l'output sia di facile lettura.



Fai clic sulla bandiera verde per eseguire il programma e provare a spostare il Sense HAT e il Raspberry facendo attenzione a non staccare i cavi. Mentre inclini il Sense HAT lungo i suoi tre assi, vedrai valori di beccheggio, rollio e imbardata cambiare (**Figura 7-26**).

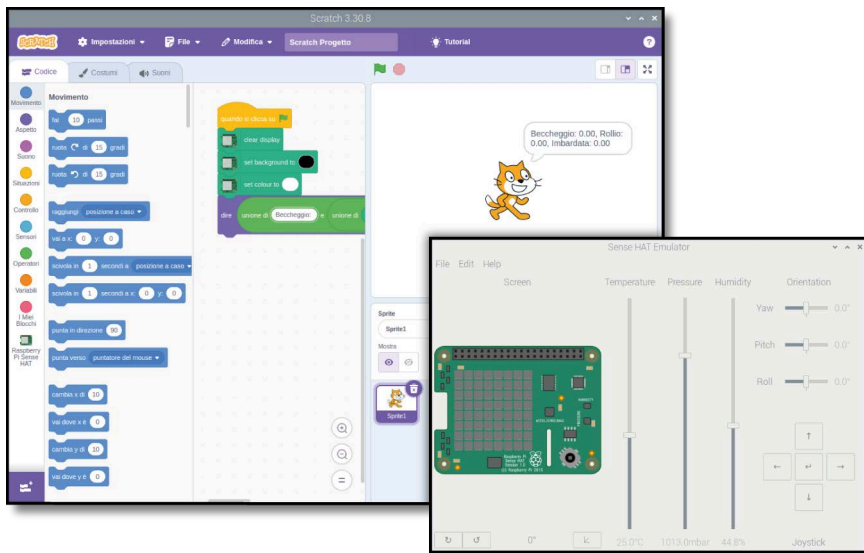


Figura 7-26 Visualizzazione dei valori di beccheggio, rollio e imbardata

Rilevamento inerziale in Python

Avvia un nuovo programma in Thonny e salvalo come **Movimento Sense HAT.py**. Digita le righe iniziali usate anche in precedenza, ricordandoti di utilizzare **sense_emu** se utilizzi un emulatore Sense HAT:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()
```

Per utilizzare le informazioni IMU per elaborare l'orientamento attuale del Sense HAT sui suoi tre assi, digita quanto segue:

```
orientation = sense.get_orientation()
pitch = orientation["pitch"]
roll = orientation["roll"]
yaw = orientation["yaw"]
print("beccheggio {0} rollio {1} imbardata {2}".format(
    pitch, roll, yaw))
```

Fai clic su **Esegui** e vedrai le letture relative all'orientamento del Sense HAT suddivise sui tre assi (**Figura 7-27**). Prova a ruotare il Sense HAT e a fare di nuovo clic su **Esegui**. I numeri dovrebbero cambiare per riflettere il nuovo orientamento.

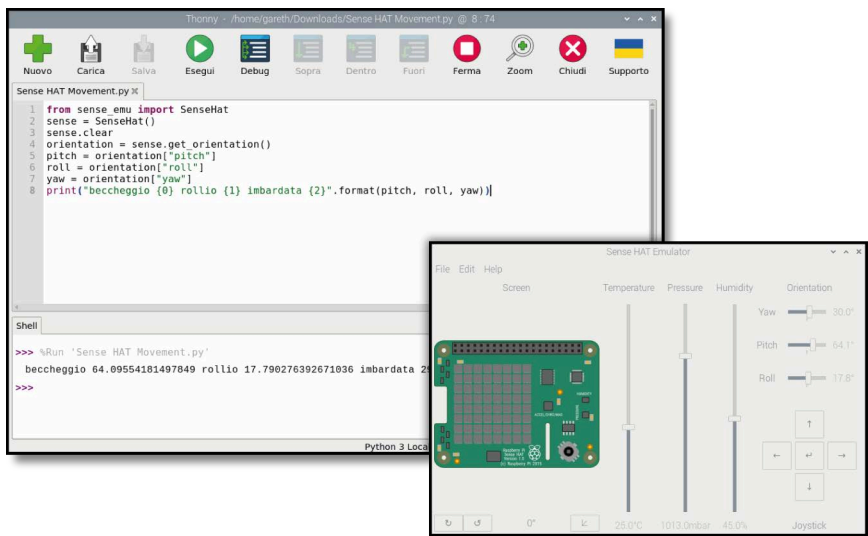


Figura 7-27 Visualizzazione dei valori di beccheggio, rollio e imbardata del Sense HAT

Oltre all'orientamento, con l'unità di misura inerziale è possibile rilevare anche il movimento. Per ottenere letture accurate del movimento, l'IMU deve essere letto frequentemente in una ripetizione. Una sola lettura non fornisce informazioni utili per il rilevamento del movimento. Elimina tutte le righe dopo **`sense.clear()`** e digita il seguente codice:

```
while True:
    acceleration = sense.get_accelerometer_raw()
    x = acceleration["x"]
    y = acceleration["y"]
    z = acceleration["z"]
```

Ora disponi delle variabili che contengono le letture dell'accelerometro per i tre assi spaziali: X (sinistra e destra), Y (avanti e indietro) e Z (su o giù). I numeri del sensore dell'accelerometro potrebbero essere difficili da leggere, quindi digita quanto segue per renderli più comprensibili arrotondandoli al numero intero più vicino:

```
x = round(x)
y = round(y)
z = round(z)
```

Infine, stampa i tre valori digitando quanto segue:

```
print("x={0}, y={1}, z={2}".format(x, y, z))
```

Fai clic su **Esegui** e visualizzerai i valori dell'accelerometro stampati nell'area Python shell (**Figura 7-28**). A differenza dei valori del programma precedente, questi verranno stampati in modo continuo. Per interrompere la stampa, fai clic sul pulsante **Ferma** rosso per arrestare il programma.

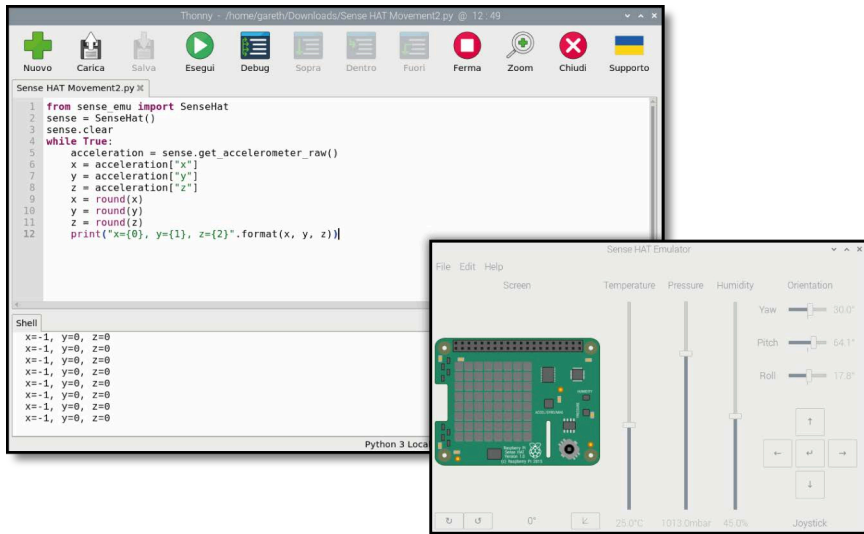


Figura 7-28 Letture dell'accelerometro arrotondate al numero intero più vicino

Avrai notato che l'accelerometro ti sta dicendo che uno degli assi (l'asse Z, se il Raspberry Pi è in orizzontale sul tavolo) ha un valore di accelerazione di gravità pari a 1.0 (1G) anche se il Sense HAT non si muove. Questo perché sta rilevando l'attrazione gravitazionale della Terra, ossia la forza che attrae il Sense HAT verso il centro della Terra e la ragione per cui, se lo fai cadere dalla tua scrivania, finirà sul pavimento.

Con il programma in esecuzione, prova a prendere con delicatezza il Sense HAT e il Raspberry Pi e a farli ruotare, assicurandoti di non staccare i cavi. Con le porte di rete e USB del Raspberry Pi rivolte verso il pavimento, potrai notare che i valori cambiano: l'asse Z mostra 0 G e l'asse X mostra ora 1 G. Ruota di nuovo il dispositivo in modo che le porte HDMI e di alimentazione siano rivolte verso il pavimento e vedrai che è l'asse Y a indicare 1 G. Se fai il contrario e orienti il Raspberry Pi in modo che la porta HDMI sia rivolta verso il soffitto, vedrai invece un -1 G sull'asse Y.

Usando le conoscenze che hai sulla gravità della Terra (all'incirca intorno a 1 G) e sugli assi spaziali, potrai usare le letture dell'accelerometro per capire quando il dispositivo è rivolto verso il basso e quando verso l'alto. Puoi anche usarlo per rilevare il movimento: prova a scuotere con cautela il Sense HAT e il Raspberry Pi e osserva i numeri: più forte viene scosso, maggiore è l'accelerazione.

Quando utilizzi `sense.get_accelerometer_raw()` stai dicendo a Sense HAT di spegnere gli altri due sensori dell'IMU (sensore giroscopico e magnetometro) e di indicare esclusivamente i dati dall'accelerometro. Puoi fare la stessa cosa anche con gli altri sensori.

Trova la riga `acceleration = sense.get_accelerometer_raw()` e modificala come segue:

```
orientation = sense.get_gyroscope_raw()
```

Sostituisci la parola `acceleration` con `orientation` su tutte e tre le righe seguenti. Fai clic su **Esegui** e vedrai l'orientamento del Sense HAT per tutti e tre gli assi, arrotondato al numero intero più vicino. A differenza dell'ultimo valore dell'orientamento, questa volta i dati provengono esclusivamente dal giroscopio ed escludono i valori di accelerometro o magnetometro. Questo può essere utile ad esempio per conoscere l'orientamento di un Sense HAT in movimento sul retro di un robot, senza che il movimento crei confusione, oppure se stai usando il Sense HAT vicino a un forte campo magnetico.

Arresta il programma facendo clic sul pulsante **Ferma** rosso. Per utilizzare il magnetometro, cancella tutte le righe del programma eccetto le prime quattro, quindi digita quanto segue sotto la riga `while True:`

```
north = sense.get_compass()
print(north)
```

Esegui il programma e vedrai la direzione del nord magnetico stampata ripetutamente nell'area Python shell. Ruota con attenzione il Raspberry Pi, e vedrai il cambiamento di direzione mentre l'orientamento del Sense HAT relativo al nord si sposta: hai costruito una bussola. Se hai un magnete (anche una semplice calamita da frigo) prova a spostarlo intorno al Sense HAT per vedere come incide sulle letture del magnetometro.

SFIDA: ROTAZIONE AUTOMATICA

Usando quello che hai imparato sulla matrice LED e sui sensori dell'unità di misura inerziale, riesci a scrivere un programma che fa ruotare un'immagine a seconda della posizione del Sense HAT?



Controllo del joystick

Il joystick del Sense HAT, nell'angolo in basso a destra, è piccolo ma sorprendentemente potente: oltre ad essere in grado di riconoscere gli input da quattro direzioni (su, giù, sinistra e destra) ha anche un quinto input al quale si accede premendo il joystick dall'alto come un interruttore a pulsante.



ATTENZIONE

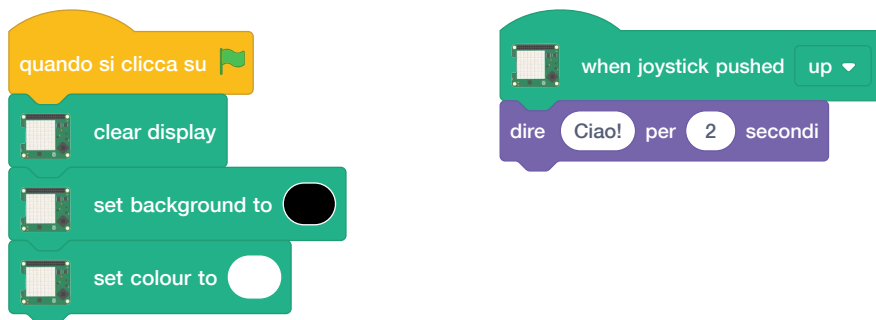
Il joystick del Sense HAT deve essere utilizzato solo se sono stati montati i distanziatori, come descritto all'inizio di questo capitolo. Senza i distanziatori, spingendo verso il basso il joystick, la scheda Sense HAT potrebbe flettersi e danneggiarsi, danneggiando anche il connettore GPIO del Raspberry Pi.

Controllo del joystick in Scratch

Avvia un nuovo programma in Scratch con l'estensione **Raspberry Pi Sense HAT** caricata. Come hai fatto prima, trascina un blocco **quando si clicca su** sotto di esso, quindi trascina e modifica un blocco verde **clear display** sotto al precedente. A questo punto, aggiungi un blocco **set background to black** e **set colour to white**.

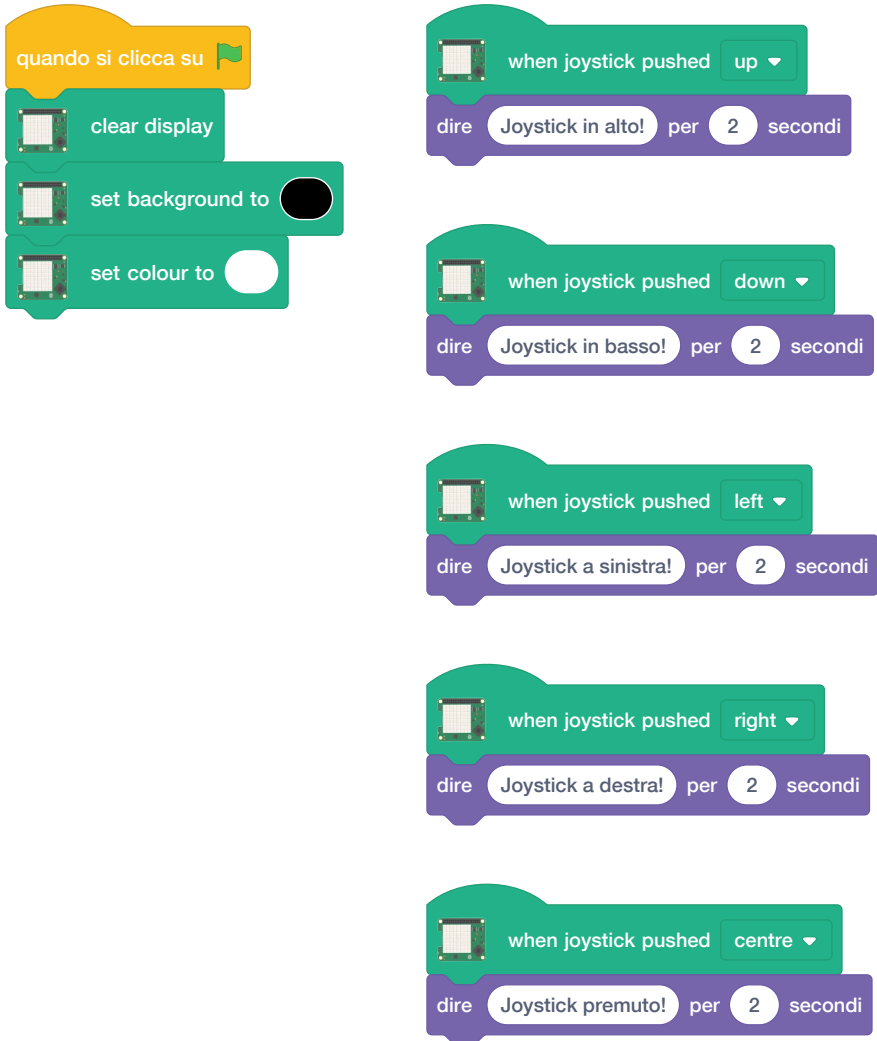
In Scratch, il joystick del Sense HAT corrisponde ai tasti del cursore sulla tastiera: se spingi il joystick verso l'alto è come se stessi premendo il tasto freccia su, mentre se lo spingi verso il basso equivale a premere il tasto freccia giù. Se lo spingi verso sinistra o destra è come se premessi i tasti freccia sinistra o destra. Se premi il joystick al centro, come se fosse un interruttore a pulsante, equivale a premere il tasto **INVIO**.

Trascina un blocco **when joystick pushed up** nell'area codice. Quindi trascina un blocco **dire Ciao! per 2 secondi** sotto di esso.



Porta il joystick verso l'alto e vedrai il gatto Scratch dire un allegro "Ciao!". Il controllo del joystick è disponibile solo sul Sense HAT fisico. Quando utilizzi l'emulatore Sense HAT, usa i tasti corrispondenti sulla tastiera per simulare le pressioni del joystick.

Quindi, modifica **dire Ciao!** in **dire Joystick in alto!** e aggiungi i blocchi **Situazioni** e **Aspetto** fino a quando non hai aggiunto azioni per tutte e cinque le direzioni in cui può essere mosso il joystick. Prova a muovere il joystick nelle varie direzioni e osserva come vengono visualizzati i messaggi!





SFIDA FINALE

Puoi usare il joystick del Sense HAT per controllare uno sprite Scratch nell'area stage? Puoi fare in modo che se lo sprite raccoglie un altro sprite che rappresenta un oggetto il display a matrice LED del Sense HAT mostri un messaggio?

Controllo del joystick in Python

Avvia un nuovo programma in Thonny e salvalo come Joystick Sense HAT. Inizia con le consuete tre righe che consentono di impostare Sense HAT e cancellare eventuali pattern dalla matrice LED, ricordando di utilizzare `sense_emu` (al posto di `sense_hat`) se stai utilizzando un emulatore:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()
```

Quindi imposta un loop infinito:

```
while True:
```

Indica a Python di rilevare gli input dal joystick del Sense HAT con la seguente riga, che verrà fatta rientrare automaticamente da Thonny:

```
    for event in sense.stick.get_events():
```

Infine, aggiungi la seguente riga (verrà fatta rientrare da Thonny) per indicare le azioni da eseguire quando viene rilevata una pressione del joystick:

```
        print(event.direction, event.action)
```

Fai clic su **Esegui** e prova a spostare il joystick nelle varie direzioni. Vedrai la direzione che hai scelto stampata nell'area Python shell: su, giù, sinistra, destra e al centro per quando hai premuto il joystick come un interruttore a pulsante.

Per ogni volta che premi il joystick vedrai due eventi: un evento `pressed` per quando si sposta la prima volta in una direzione e uno `released` per quando il joystick torna al centro.

Puoi usarlo nei programmi: pensa a un personaggio in un gioco, che potrebbe iniziare a muoversi quando il joystick viene premuto in una direzione per fermarsi non appena viene rilasciato.

Puoi anche utilizzare il joystick per attivare funzioni, anziché limitarti a utilizzarlo per un ciclo `for`. Elimina le righe sotto a `sense.clear()` e digita quanto segue:

```
def red():
    sense.clear(255, 0, 0)

def blue():
    sense.clear(0, 0, 255)

def green():
    sense.clear(0, 255, 0)

def yellow():
    sense.clear(255, 255, 0)
```

Queste funzioni modificano l'intera matrice LED del Sense HAT in un unico colore, rosso, blu, verde o giallo, così riuscirai a capire con facilità che il programma funziona. Per attivarle è necessario indicare a Python le funzioni correlate ai vari input del joystick. Digita quanto segue:

```
sense.stick.direction_up = red
sense.stick.direction_down = blue
sense.stick.direction_left = green
sense.stick.direction_right = yellow
sense.stick.direction_middle = sense.clear
```

Infine, il programma ha bisogno di un ciclo infinito (noto come *main loop*, ossia *ciclo principale*) per continuare a girare. Questo significa che dovrai continuare a controllare gli input dei joystick, anziché limitarti a eseguire una volta il codice che hai scritto e terminare. Digita quanto segue:

```
while True:
    pass
```

Il programma completato dovrebbe avere questa struttura:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()

def red():
    sense.clear(255, 0, 0)

def blue():
```

```

sense.clear(0, 0, 255)

def green():
    sense.clear(0, 255, 0)

def yellow():
    sense.clear(255, 255, 0)

sense.stick.direction_up = red
sense.stick.direction_down = blue
sense.stick.direction_left = green
sense.stick.direction_right = yellow
sense.stick.direction_middle = sense.clear

while True:
    pass

```

Fai clic su **Esegui**, prova a muovere il joystick e vedrai i LED illuminarsi del colore scelto. Per spegnere i LED, premi il joystick come se fosse un pulsante. La direzione **middle** è impostata per utilizzare la funzione **sense.clear()** per spegnerli tutti. Complimenti ora puoi acquisire input dal joystick.



SFIDA FINALE

Riesci ad usare quello che hai imparato per disegnare un'immagine sullo schermo e farla ruotare in qualsiasi direzione il joystick venga spostato? Sei in grado di usare l'input centrale per passare da un'immagine all'altra?

Progetto Scratch: luce scintillante con Sense HAT

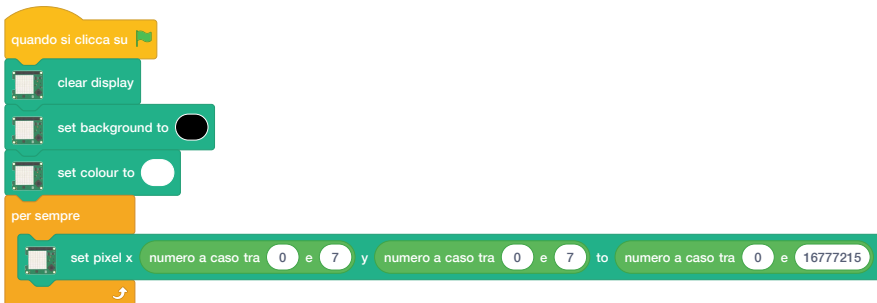
Ora che conosci il funzionamento del Sense HAT, è il momento di creare un programma che emetta una luce scintillante in base alla temperatura: con una pulsazione rapida quando fa freddo e gradualmente più lenta man mano che diventa più caldo.

Crea un nuovo progetto in Scratch e aggiungi l'estensione Sense HAT Raspberry Pi se non è già presente. Come sempre, inizia con quattro blocchi: aggiungi **quando si clicca su** con un blocco **clear display** al di sotto di esso. Poi aggiungi **set background to black** e **set colour to white**, ricordandoti che dovrai cambiare i colori predefiniti.

Inizia creando uno scintillio semplice, ma artistico. Inizia trascinando un blocco **per sempre** nell'area codice, quindi completa con un blocco **set pixel x 0 y 0 to colour**. Anziché utilizzare i numeri impostati, completa ciascuna delle sezioni x, y e colore di quel blocco con un blocco **numero a caso tra 1 e 10** **Operatori**.

I valori da 1 a 10 non sono molto utili in questo caso, quindi è necessario fare qualche modifica. I primi due numeri del blocco **set pixel** sono le coordinate X e Y del pixel sulla matrice LED, il che significa che dovrebbero essere numeri compresi tra 0 e 7, quindi modifica i primi due blocchi in **numero a caso tra 0 e 7**.

La sezione successiva è il colore con cui impostare il pixel. Quando utilizzi il selettore dei colori, il colore scelto viene mostrato direttamente nell'area dello script, internamente però i colori sono rappresentati da un numero, quindi puoi utilizzare quest'ultimo. Modifica l'ultimo blocco **numero a caso tra** per leggere **numero a caso tra 0 e 16777215**.



Fai clic sulla bandiera verde e vedrai i LED del Sense HAT cominciare a illuminarsi con colori casuali (**Figura 7-29**). Complimenti, hai realizzato una luce scintillante!

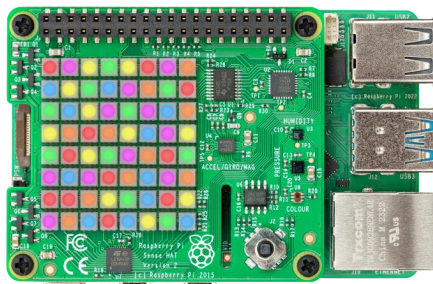
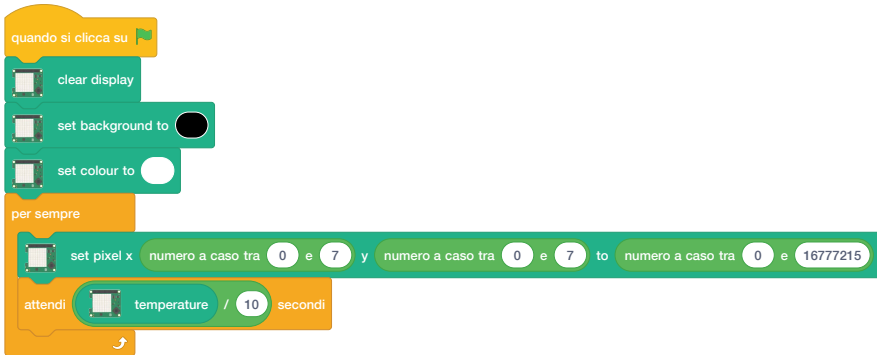


Figura 7-29 Accendere i pixel con colori casuali

Il programma ora non è molto interattivo. Per renderlo interattivo, trascina un blocco **attendi 1 secondi** in modo che sia sotto il blocco **set pixel**, ma all'interno del blocco **per sempre**. Trascina un blocco **Operatori** sopra l'1, quindi digita 10 nel secondo spazio. Infine, trascina un blocco **temperature** sul primo spazio del blocco **Operatori** di divisione.



Fai clic sulla bandiera verde e, a meno che tu non viva in un posto molto freddo, noterai che la scintilla è più lenta di prima. Questo perché hai creato un ritardo basato sulla temperatura: il programma ora aspetta *la temperatura attuale divisa per 10* secondi prima di ogni ciclo. Se la temperatura nella stanza è di 20 °C, il programma aspetterà 2 secondi prima di eseguire la ripetizione, mentre se la temperatura è di 10 °C, aspetterà 1 secondo e sotto i 10 °C meno di un secondo.

Se il tuo Sense HAT rileva una temperatura negativa (ossia al di sotto di 0 °C, il punto di congelamento dell'acqua) cercherà di aspettare meno di 0 secondi, ma siccome questo è impossibile senza viaggiare nel tempo, vedrai comunque lo stesso effetto, come se stesse aspettando 0 secondi. Congratulazioni, hai imparato a integrare le funzioni del Sense HAT nei tuoi programmi.

Progetto Python: Tricorder con Sense HAT

Ora che conosci il funzionamento del Sense HAT, è il momento di creare un tricorder, un dispositivo dotato di sensori che i fan di una certa saga fantascientifica riconosceranno subito. Il tricorder fittizio utilizza diversi sensori per conoscere l'ambiente circostante.

Avvia un nuovo progetto in Thonny e salvalo con il nome di **Tricorder.py**, quindi inizia con le righe da utilizzare ogni volta che avvii un programma Sense HAT in Python, ricordandoti di usare `sense_emu` se stai utilizzando l'emulatore Sense HAT:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()
```

Dovrai quindi avviare le funzioni per definire ciascun sensore del Sense HAT. Inizia con l'unità di misura inerziale digitando:

```
def orientation():
    orientation = sense.get_orientation()
    pitch = orientation["pitch"]
    roll = orientation["roll"]
    yaw = orientation["yaw"]
```

Poiché i risultati del sensore vengono fatti scorrere attraverso i LED, arrotonda i risultati in modo da non dover attendere per visualizzare decine di decimali. Invece di visualizzare solo numeri interi, arrotonda a un decimale digitando quanto segue:

```
pitch = round(pitch, 1)
roll = round(roll, 1)
yaw = round(yaw, 1)
```

Infine dovrai indicare a Python di mostrare i risultati nei LED, in modo che il tricorder funzioni come un dispositivo portatile senza bisogno di essere collegato a un monitor o un televisore:

```
sense.show_message("Beccheggio {0}, Rollio {1}, Imbardata {2}".
                    format(pitch, roll, yaw))
```

Ora che disponi di una funzione completa per la lettura e la visualizzazione dell'orientamento dall'IMU, crea funzioni simili per ciascuno degli altri sensori. Inizia con il sensore di temperatura:

```
def temperature():
    temp = sense.get_temperature()
    temp = round(temp, 1)
    sense.show_message("Temperatura: %s gradi Celsius" % temp)
```

Guarda attentamente la riga che stampa il risultato sui LED: `%s` è un *segnaposto* e viene sostituito con il contenuto della variabile `temp`. In questo modo è possibile formattare l'output in modo comprensibile con un'etichetta, "Temperatura:", e un'unità di misura, "gradi Celsius".

Quindi definisci una funzione per il sensore di umidità:

```
def humidity():
    humidity = sense.get_humidity()
    humidity = round(humidity, 1)
    sense.show_message("Umidità: %s percento" % humidity)
```

Poi il sensore di pressione:

```
def pressure():
    pressure = sense.get_pressure()
    pressure = round(pressure, 1)
    sense.show_message("Pressione: %s millibar" % pressure)
```

Infine, definisci una funzione per la lettura della bussola dal magnetometro:

```
def compass():
    for i in range(0, 10):
        north = sense.get_compass()
        north = round(north, 1)
    sense.show_message("Nord: %s gradi" % north)
```

Il ciclo breve **for** in questa funzione richiede dieci letture del magnetometro per avere dati sufficienti al fine di fornire un risultato accurato. Se ritieni che il valore riportato continui a cambiare, prova a estenderlo a 20, 30 o anche 100 ripetizioni per migliorare ulteriormente la precisione.

Il tuo programma ora ha cinque funzioni, ognuna delle quali riporta una lettura di uno dei sensori del Sense HAT facendola scorrere sui LED. È però necessario un modo per scegliere il sensore da usare e il joystick è perfetto a questo scopo.

Digita quanto segue:

```
sense.stick.direction_up = orientation
sense.stick.direction_right = temperature
sense.stick.direction_down = compass
sense.stick.direction_left = humidity
sense.stick.direction_middle = pressure
```

Queste righe assegnano un sensore a ciascuna delle cinque possibili direzioni sul joystick: **Up** legge dal sensore di orientamento, **down** dal magnetometro, **left** dal sensore di umidità, **right** dal sensore di temperatura e premendo **middle** dal sensore di pressione.

Infine, è necessario un ciclo principale in modo che il programma continui ad attendere le pressioni del joystick e non si fermi immediatamente. Al termine del programma digita quanto segue:

```
while True:
    pass
```

Il programma completato dovrebbe avere questa struttura:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()

def orientation():
    orientation = sense.get_orientation()
    pitch = orientation["pitch"]
    roll = orientation["roll"]
    yaw = orientation["yaw"]

    pitch = round(pitch, 1)
    roll = round(roll, 1)
    yaw = round(yaw, 1)

    sense.show_message("Beccheggio {0}, Rollio {1}, Imbardata {2}".
                       format(pitch, roll, yaw))

def temperature():
    temp = sense.get_temperature()
    temp = round(temp, 1)
    sense.show_message("Temperatura: %s gradi Celsius" % temp)

def humidity():
    humidity = sense.get_humidity()
    humidity = round(humidity, 1)
    sense.show_message("Umidità: %s percento" % humidity)

def pressure():
    pressure = sense.get_pressure()
    pressure = round(pressure, 1)
    sense.show_message("Pressione: %s millibar" % pressure)

def compass():
    for i in range(0, 10):
        north = sense.get_compass()
        north = round(north, 1)
        sense.show_message("Nord: %s gradi" % north)
```

```
sense.stick.direction_up = orientation
sense.stick.direction_right = temperature
sense.stick.direction_down = compass
sense.stick.direction_left = humidity
sense.stick.direction_middle = pressure
```

```
while True:
    pass
```

Fai clic su **Esegui** e prova a spostare il joystick per effettuare una lettura da uno dei sensori (**Figura 7-30**). Quando ha finito di scorrere il risultato, spostalo in un'altra direzione. Complimenti: hai costruito un tricorder portatile che renderebbe orgogliosa la Federazione dei Pianeti Uniti!

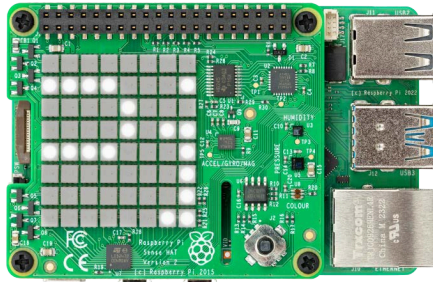
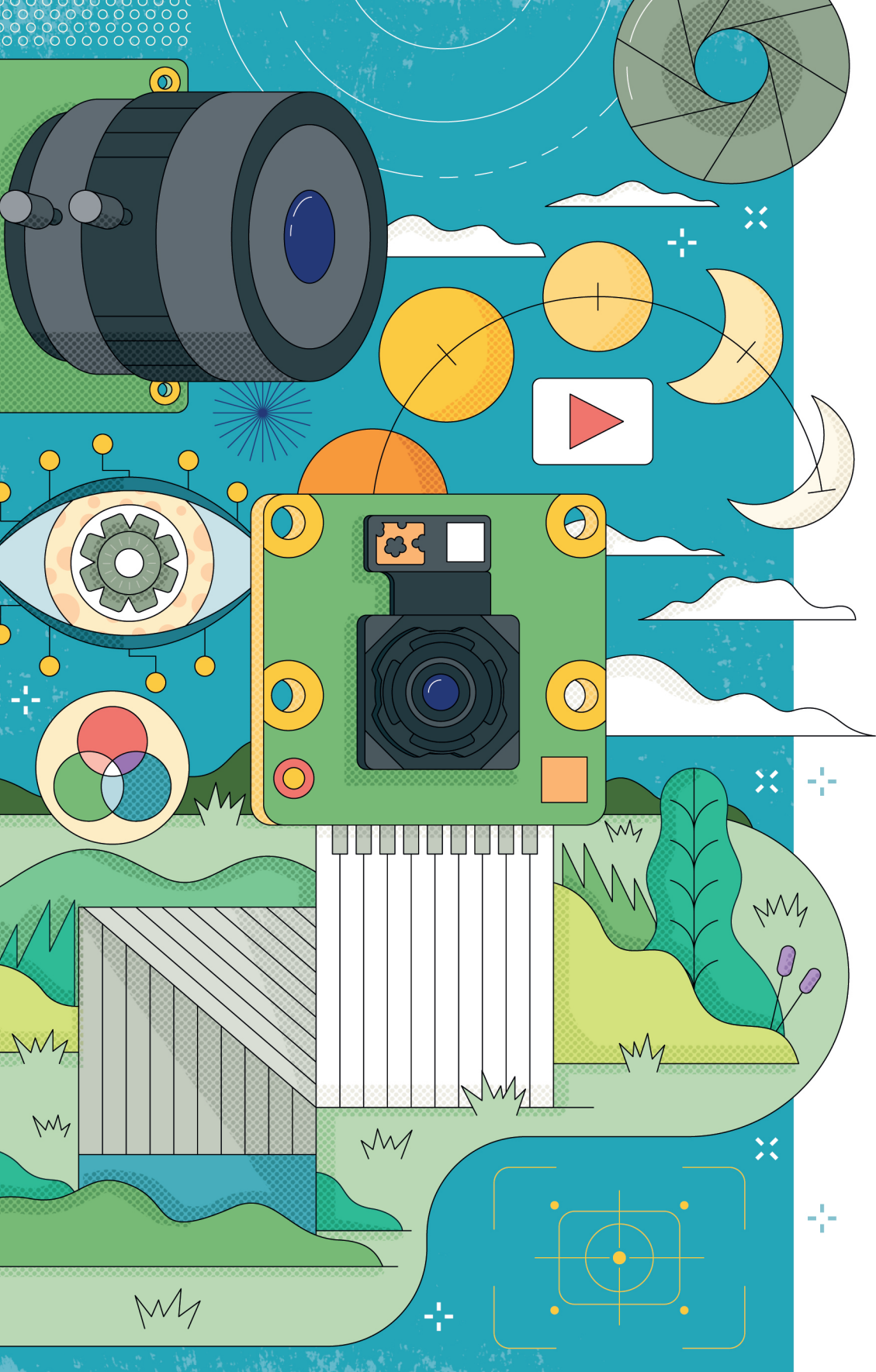


Figura 7-30 Ogni lettura scorre sul display

Per altri progetti Sense HAT, compreso un esempio di utilizzo del sensore di colore su Sense HAT V2, segui i link nell'Appendice D, *Ulteriori letture*.



Capitolo 8

Moduli fotocamera di Raspberry Pi

Con il Camera Module o la High Quality Camera potrai scattare foto e registrare video in alta risoluzione e creare fantastici progetti di visione artificiale direttamente sul tuo Raspberry Pi.

Se hai sempre desiderato costruire qualcosa che sia in grado di vedere da autonomamente, noto nel campo della robotica come *visione artificiale*, allora il Camera Module 3 (modulo fotocamera 3) (**Figura 8-1**), la High Quality Camera (fotocamera ad alta qualità) o la fotocamera Global Shutter di Raspberry Pi sono accessori opzionali indispensabili. I tre moduli fotocamera sono piccoli circuiti stampati di forma quadrata con un sottile cavo a nastro; collegali alla porta dell'interfaccia seriale della fotocamera (CSI) sul tuo Raspberry Pi per avere immagini fisse ad alta risoluzione e segnali video in movimento che possono essere utilizzati così come sono o integrati nei tuoi programmi.

RASPBERRY PI 400

I moduli fotocamera Raspberry Pi non sono compatibili con il computer desktop Raspberry Pi 400. Puoi utilizzare le webcam USB come alternativa, ma non potrai utilizzare gli strumenti software illustrati in questo capitolo con il Raspberry Pi 400.



Varianti della fotocamera

Esistono diversi tipi di moduli fotocamera Raspberry Pi e il modello perfetto per te dipende dalle immagini che vuoi acquisire: il Camera Module standard 3, la versione "NoIR", la High Quality (HQ) Camera e la fotocamera Global

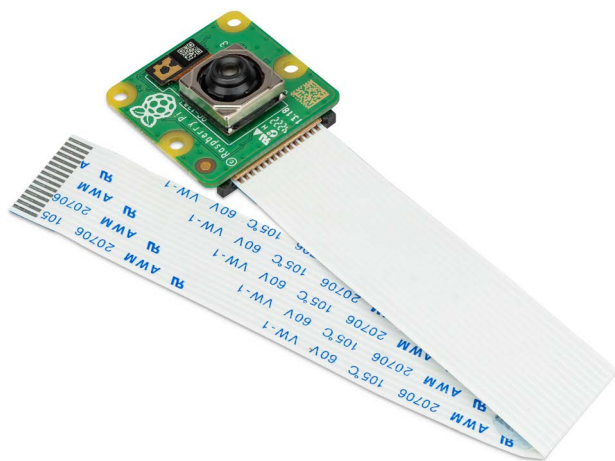


Figura 8-1 Modulo fotocamera 3 di Raspberry Pi

Shutter. Se desideri scattare foto e video normali in ambienti ben illuminati, ti consigliamo di utilizzare il Camera Module standard 3 o il modulo fotocamera 3 Wide per un campo visivo più ampio.

Se desideri poter scambiare gli obiettivi e vuoi ottenere immagini di qualità migliore, utilizza la HQ Camera. Il Camera Module 3 NoIR (chiamato così perché non ha un filtro a infrarossi o IR), è progettato per essere usato con sorgenti di luce infrarossa per scattare foto e video al buio, ed è disponibile anche nella versione grandangolare. Se vuoi realizzare una casetta per uccelli, una telecamera di sicurezza o altri progetti che prevedono la visione notturna, opta per la versione NoIR, ma ricordati di acquistare una sorgente di luce infrarossa. Infine, la fotocamera Global Shutter acquisisce l'immagine intera in una sola volta, anziché riga per riga, rendendola adatta alla fotografia ad alta velocità e alla visione artificiale.

Camera Module 3 di Raspberry Pi

Il Camera Module 3 di Raspberry Pi, nelle versioni standard e NoIR, è costruito attorno a un sensore di immagine Sony IMX708. Si tratta di un *sensore da 12 megapixel*, il che significa che acquisisce immagini con un massimo di 12 milioni di pixel. Il risultato è una dimensione massima dell'immagine di 4608 pixel di larghezza per 2592 pixel di altezza. Sono disponibili due obiettivi per il Camera Module 3 di Raspberry Pi: l'obiettivo standard, con un campo visivo di 75 gradi, e l'obiettivo grandangolare, che ha un campo visivo di 120 gradi.

Oltre alle fotografie, il Camera Module 3 di Raspberry Pi è in grado di registrare filmati con risoluzione Full HD (1080p) a una velocità di 50 fotogrammi al secondo (50 fps). Per un movimento più fluido o anche per creare un effetto a rallentatore, la fotocamera può essere impostata per acquisire a una frequenza di fotogrammi più elevata abbassando la risoluzione: avrai 100 fps a una risoluzione di 720p e 120 fps a una risoluzione di 480p (VGA). Il modulo ha un ultimo asso nella manica rispetto alle versioni precedenti: offre l'*autofocus*, che consente di regolare il punto di messa a fuoco dell'obiettivo per i soggetti vicini o lontani.

High Quality Camera di Raspberry Pi

La High Quality Camera utilizza un sensore Sony IMX477 da 12,3 megapixel, più grande di quello presente nei Camera Module standard e NoIR: ciò significa che è in grado di raccogliere una quantità maggiore di luce e quindi di ottenere immagini di qualità superiore. A differenza dei moduli fotocamera, la fotocamera HQ non include un obiettivo e senza di esso non potrai scattare foto o video. Puoi utilizzare qualsiasi obiettivo con un attacco C o CS oppure altri obiettivi dotati di un adattatore appropriato per tale attacco. È disponibile una versione alternativa della fotocamera HQ per l'utilizzo di obiettivi con attacco M12.

Fotocamera Global Shutter di Raspberry Pi

La fotocamera Global Shutter utilizza un sensore Sony IMX296 da 1,6 megapixel. Sebbene fornisca una risoluzione inferiore rispetto al modulo fotocamera standard di Raspberry Pi o alla fotocamera HQ, la sua capacità di catturare l'intera immagine in una sola volta significa che eccelle nell'acquisizione di soggetti in rapido movimento senza la distorsione che si può ottenere con una fotocamera Rolling Shutter. Come la fotocamera HQ, viene fornita senza obiettivo e supporta gli stessi obiettivi con attacco C e CS; a differenza della fotocamera HQ, attualmente non esiste una versione con attacco M12.

Camera Module 2 di Raspberry Pi

Il precedente Camera Module 2 di Raspberry Pi e la sua variante NoIR si basano su un sensore d'immagine Sony IMX219 da 8 megapixel, il che significa che può scattare foto con un massimo di 8 milioni di pixel (3280 pixel in larghezza e 2464 in altezza). Oltre alle immagini, il modulo fotocamera è in grado di acquisire video con risoluzione Full HD (1080p) a 30 fotogrammi al secondo (30 fps), con frequenze più elevate disponibili a risoluzioni inferiori: 60 fps per filmati a 720p e fino a 90 fps per filmati a 480p (VGA).



RASPBERRY PI ZERO E RASPBERRY PI 5

Tutti i modelli del Camera Module Raspberry Pi sono compatibili con Raspberry Pi Zero 2 W, con le versioni più recenti di Raspberry Pi Zero e Zero W originali e con Raspberry Pi 5. Se utilizzi un Raspberry Pi 5, è necessario un cavo a nastro diverso da quello utilizzato con Raspberry Pi 4 e i modelli precedenti.

Chiedi al tuo rivenditore autorizzato di fiducia un cavo adatto: l'estremità più larga va collegata alla fotocamera, mentre quella più stretta al Raspberry Pi.

Come installare la fotocamera

Come qualsiasi componente hardware, il Camera Module o la HQ Camera devono essere collegati o scollegati dal Raspberry Pi solo quando l'alimentazione è spenta e il cavo di alimentazione è scollegato. Se il tuo Raspberry Pi è acceso, seleziona **Arresta** dal menu Raspberry Pi, attendi lo spegnimento, quindi scollega l'alimentazione.

Nella maggior parte dei casi, il cavo a nastro è già collegato al Camera Module o alla HQ Camera. In caso contrario, capovolgi la scheda della fotocamera in modo che il sensore si trovi sul fondo e cerca un connettore piatto in plastica. Afferra con cautela i bordi sporgenti con le unghie e tira verso l'esterno fino a estrarre parzialmente il connettore. Inserisci il cavo a nastro con i bordi argentati o dorati verso il basso e il rinforzo in plastica rivolto verso l'alto sotto la linguetta appena estratta, quindi spingi delicatamente la linguetta in posizione fino a sentire un clic (**Figura 8-2**). Se il cavo è installato correttamente, sarà dritto e non si scollegherà se viene tirato leggermente; in caso contrario, estrai la linguetta e riprova.

Installa l'altra estremità del cavo nello stesso modo. Individua la porta inferiore tra le due porte fotocamera/display, contrassegnata con "CAM/DISP 0" su Raspberry Pi 5 o la porta singola per fotocamera su Raspberry Pi 4, Raspberry Pi Zero 2 W e modelli precedenti, e solleva il piccolo coperchio in plastica. Se il Raspberry Pi è installato in un case, rimuovilo.

Posiziona il Raspberry Pi 5 in modo che il connettore GPIO sia a destra e le porte HDMI a sinistra. Inserisci il cavo a nastro in modo che i bordi argentati o dorati siano rivolti verso di te e il rinforzo in plastica sia sul lato opposto (**Figura 8-3**). Quindi spingi delicatamente la linguetta in posizione.

Per Raspberry Pi 4 e modelli precedenti, il cavo a nastro deve essere al contrario, con il rinforzo in plastica rivolto verso di te e i bordi argentati o dorati sul lato opposto. Se utilizzi un Raspberry Pi Zero 2 W o un Raspberry Pi Zero più vecchio, i bordi argentati o dorati devono essere rivolti verso il tavolo e il rin-

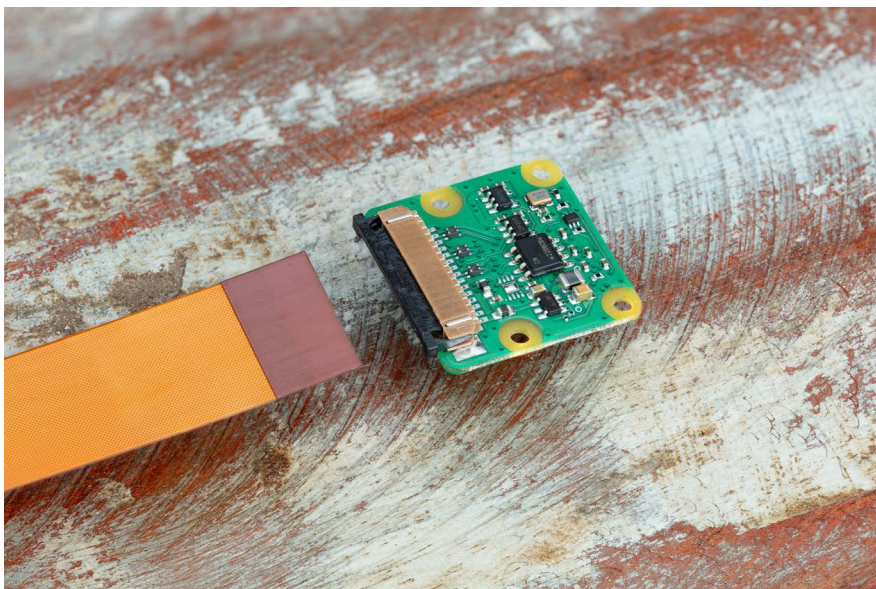


Figura 8-2 Collegamento del cavo a nastro al modulo fotocamera

forzo in plastica verso il soffitto. Se il cavo è installato correttamente, sarà dritto e non si scollegherà se viene tirato leggermente; in caso contrario, estrai la linguetta e riprova.

Il Camera Module presenta un piccolo pezzo di plastica blu che copre l'obiettivo per proteggerlo dai graffi durante la produzione, la spedizione e l'installazione. Individua la piccola linguetta di plastica e rimuovila delicatamente dall'obiettivo per iniziare a utilizzare la fotocamera.

Collega l'alimentatore al Raspberry Pi e fai caricare Raspberry Pi OS.

REGOLAZIONE DELLA MESSA A FUOCO

Tutte le versioni del Camera Module 3 di Raspberry Pi includono un sistema di messa a fuoco automatica motorizzato, in grado di regolare il punto di messa a fuoco dell'obiettivo tra oggetti vicini e lontani. Il Camera Module 2 di Raspberry Pi utilizza un obiettivo che prevede una regolazione manuale limitata della messa a fuoco. La confezione contiene un piccolo strumento per ruotare l'obiettivo e regolare la messa a fuoco.



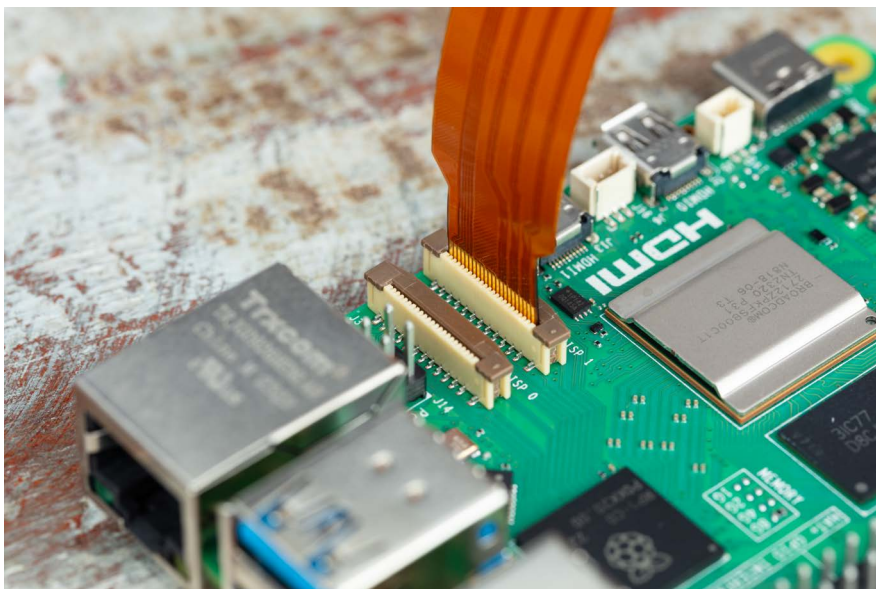


Figura 8-3 Collegamento del cavo a nastro alla porta fotocamera/CSI su Raspberry Pi

Come verificare il funzionamento della fotocamera

Per verificare che il Camera Module o la HQ Camera siano installati correttamente, è possibile utilizzare gli strumenti **rpicam**. Questi sono progettati per acquisire immagini dalla fotocamera mediante l'*interfaccia a riga di comando (CLI)* del Raspberry Pi.

A differenza dei programmi usati finora, non troverai gli strumenti **rpicam** nel menu. Fai clic sull'icona di Raspberry Pi per accedere al menu, scegli la categoria **Accessori** e fai clic su **LXTerminal**. Visualizzerai una finestra nera con una scritta verde e blu (**Figura 8-4**): questo è il *terminale*, che consente di accedere all'interfaccia a riga di comando.

Per acquisire un'immagine con la fotocamera, digita nel terminale quanto segue:

```
rpicam-still -o test.jpg
```

Non appena premi il tasto **INVIO**, vedrai apparire una finestra che mostra l'immagine di ciò che la fotocamera vede (**Figura 8-5**). Questa è una *live view* e, a meno che tu non fornisca indicazioni diverse a **rpicam-still**, durerà cinque secondi. Trascorsi i cinque secondi, la fotocamera scatterà una foto e la salverà

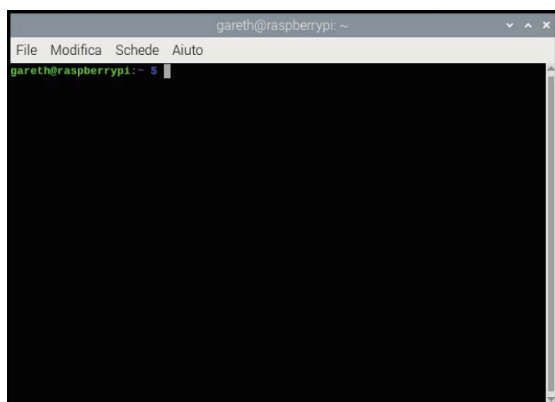


Figura 8-4 Apri una finestra Terminal per inserire i comandi

nella cartella Home con il nome **test.jpg**. Se vuoi acquisire un'altra immagine, digita nuovamente lo stesso comando, ma assicurati di cambiare il nome del file di output dopo **-o**, altrimenti sovrascriverai la prima foto.

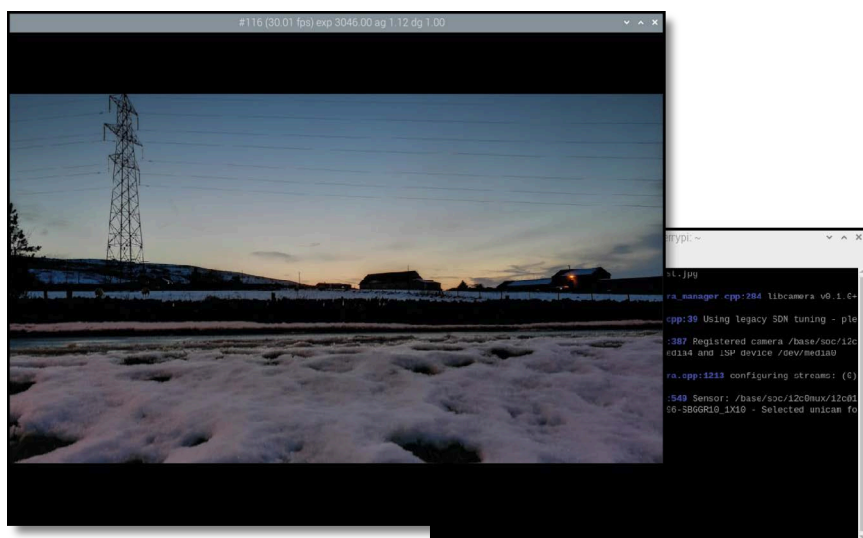


Figura 8-5 Live view della fotocamera

Se l'anteprima dal vivo era capovolta, dovrai indicare a `rpicam-still` che la fotocamera è ruotata. Il Camera Module è progettato perché il cavo a nastro esca dal bordo inferiore; se esce dai lati o dalla parte superiore, come con alcuni accessori di terze parti per l'attacco, è possibile ruotare l'immagine di 90,

180 o 270 gradi utilizzando l'opzione **--rotation**. Per una fotocamera montata con il cavo che esce dalla parte superiore è sufficiente utilizzare il seguente comando:

```
rpikam-still --rotation 180 -o test.jpg
```

Utilizza un valore di rotazione di 90 gradi se il cavo a nastro esce dal bordo destro e di 270 gradi se esce dal sinistro. Se l'immagine originale presentava un'angolazione sbagliata, prova a correggerla usando l'opzione **--rotation**.

Per visualizzare l'immagine, apri **Gestore file PCManFM** dalla categoria **Accessori** del menu Raspberry Pi: l'immagine che hai scattato, chiamata **test.jpg**, si troverà nella cartella **home/<username>**. Cerca nell'elenco dei file, quindi fai doppio clic per caricarla in un visualizzatore di immagini (**Figura 8-6**). Puoi anche allegare l'immagine alle e-mail, caricarla su siti Web tramite il browser o trascinarla in un dispositivo di archiviazione esterno.

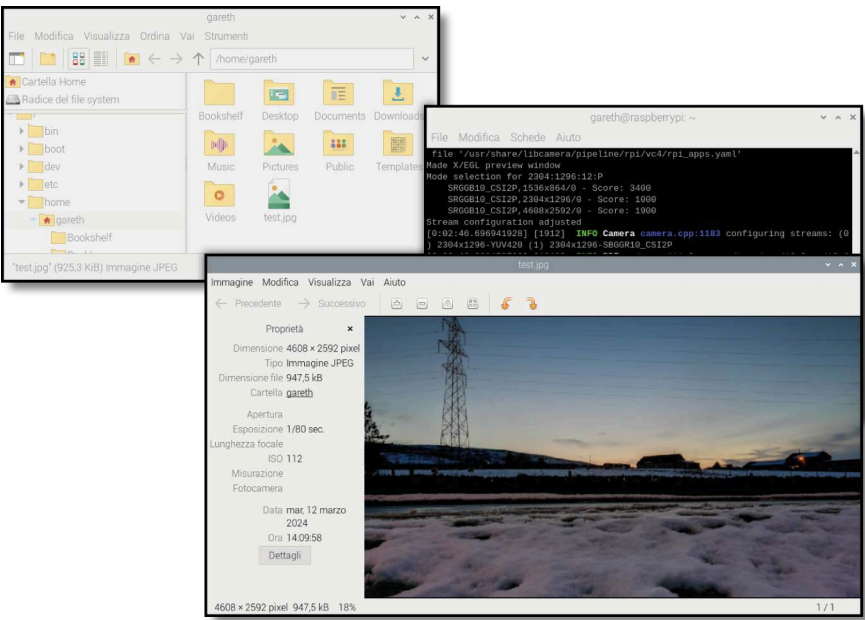


Figura 8-6 Apertura dell'immagine acquisita

Il Camera Module 3 di Raspberry Pi include la possibilità di regolare il punto di messa a fuoco dell'immagine utilizzando un sistema di messa a fuoco automatica motorizzato. Questa funzione è attivata per impostazione predefinita: quando scatti un'immagine, il modulo fotocamera regola automaticamente la messa a fuoco per rendere l'immagine il più nitida possibile, utilizzando la cosiddetta *messa a fuoco automatica continua*.

Proprio come suggerisce il nome, la messa a fuoco automatica continua regola costantemente il punto di messa a fuoco fino al momento in cui l'immagine viene acquisita. Se scatti più immagini o se stai registrando un video, il sistema continuerà a regolare la messa a fuoco. Se un oggetto in movimento passa tra la fotocamera e il soggetto, la fotocamera modifica automaticamente la messa a fuoco.

Se i risultati della messa a fuoco automatica continua non ti soddisfano, puoi scegliere altre modalità di messa a fuoco automatica, di cui si parla nel dettaglio nella sezione *Impostazioni avanzate della fotocamera* alla fine di questo capitolo.

Registrazione di video

Il Camera Module non si limita a scattare immagini fisse: può anche registrare video utilizzando uno strumento chiamato *rpicam-vid*.

UN PO' DI SPAZIO, PER FAVORE

La registrazione di video può richiedere molto spazio di archiviazione. Se hai intenzione di fare molti video, assicurati di avere una scheda microSD di grandi dimensioni. Potresti anche considerare di utilizzare una chiavetta USB o un altro dispositivo di archiviazione esterno.

Per impostazione predefinita, gli strumenti rpicam salvano i file nella cartella da cui vengono avviati. Assicurati quindi di impostare il dispositivo di archiviazione giusto quando scegli il percorso in cui salvi i file. Puoi trovare ulteriori informazioni su come modificare il percorso nel terminale nell'Appendice C, *Interfaccia a riga di comando*.



Per registrare un video breve, digita nel terminale quanto segue:

```
rpicam-vid -t 10000 -o test.h264
```

Come prima, visualizzerai la finestra di anteprima. Questa volta, però, invece di fare il conto alla rovescia e acquisire una singola immagine fissa, la fotocamera registrerà dieci secondi di video in un file. Al termine della registrazione, la finestra di anteprima si chiude automaticamente.

Per registrare un video più lungo, cambia il numero dopo **-t** con la durata di registrazione desiderata in millisecondi. Ad esempio, per effettuare una registrazione di dieci minuti, dovrai digitare:

```
rpicam-vid -t 600000 -o test2.h264
```


Per riprodurre il video, individualo in Gestore file e fai doppio clic sul file per caricarlo nel lettore video VLC (Figura 8-7). Il video si aprirà e la riproduzione verrà avviata, ma potrebbe non essere molto fluida. C'è una soluzione a questo problema: aggiungere alla registrazione delle informazioni sulla temporizzazione.

Il video registrato con `rpicas-vid` è in un formato chiamato *bitstream* e funziona in modo leggermente diverso da quello dei file video a cui siamo abituati. Di solito, i file contengono più parti: il video, l'eventuale audio acquisito insieme al video, le informazioni sul timecode che indicano quando ogni fotogramma deve essere visualizzato e informazioni aggiuntive note come *metadati*. Un *bitstream* è diverso. Non ha nulla di tutto ciò: è composto solo da dati video puri.

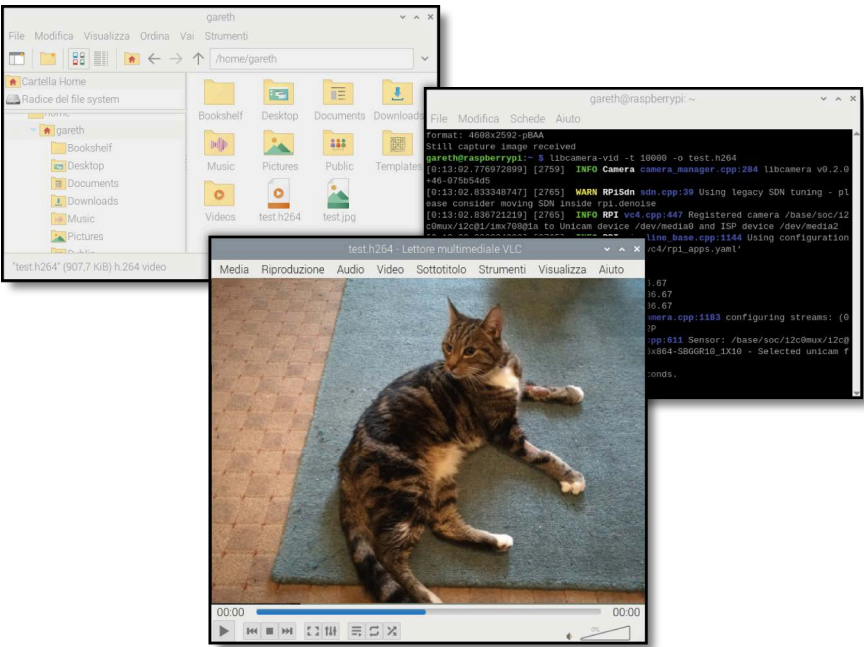


Figura 8-7 Apertura del video registrato

Per fare in modo che i file video vengano riprodotti sul maggior numero possibile di piattaforme software, compresi quelli installati su computer diversi da Raspberry Pi, è necessario elaborarli in un *contenitore*. A tal fine, sono necessarie alcune informazioni mancanti: la temporizzazione dei fotogrammi.

Nel terminale, registra un nuovo video, ma questa volta inserisci il comando che consente a `rpicas-vid` di registrare le informazioni di temporizzazione in un file chiamato **timestamps.txt**:

```
rpicam-vid -t 10000 --save-pts timestamps.txt -o test-time.h264
```

Quando apri la cartella video nel Gestore file, vedrai due file: il bitstream video, **test-time.h264**, e il file **timestamps.txt** (Figura 8-8).

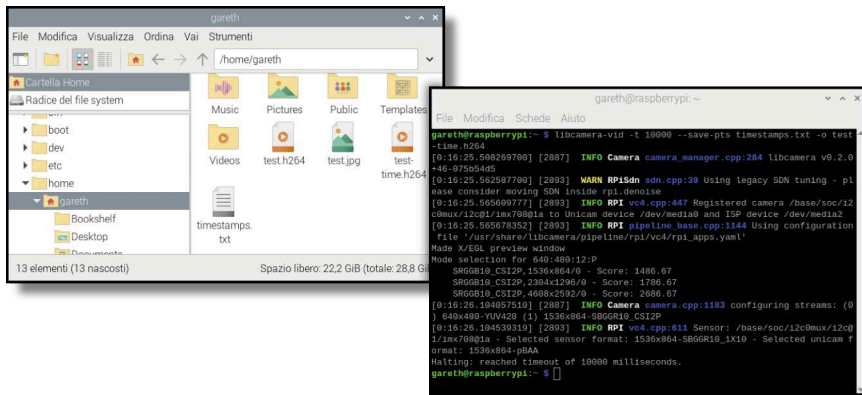


Figura 8-8 Un file video con un file timestamp separato

Per combinare questi due file in un unico contenitore adatto alla riproduzione su altri dispositivi, utilizza lo strumento **mkvmerge**, che prende il video, lo unisce con i timestamp e crea un file contenitore video noto come *file video Matroska* o **MKV**.

Nella riga di comando, digita (**** è un carattere speciale che consente di suddividere il comando su due righe):

```
mkvmerge --timecodes 0:timestamps.txt test-time.h264 \  
-o test-time.mkv
```

Ora avrai un terzo file, **test-time.mkv**. Fai doppio clic su questo file in Gestore file per caricarlo in VLC, dove potrai guardare in modo fluido il video che hai registrato. Se vuoi trasferire il video su un'unità rimovibile per la riproduzione su un altro computer, è necessario solo il file MKV e puoi eliminare i file H264 e TXT.

Ricorda sempre di salvare i timestamp insieme al tuo video se vuoi creare un file che venga riprodotto correttamente sul maggior numero di computer possibile. È più complicato tornare indietro e crearli una volta effettuata la registrazione.

Fotografia time-lapse

C'è un altro trucchetto che il modulo fotocamera ha in serbo per te: la *fotografia time-lapse*. Nella fotografia time-lapse, le immagini vengono scattate per un periodo di tempo a intervalli regolari, per riuscire a immortalare quei cambiamenti che avvengono troppo lentamente per essere osservati a occhio nudo. È un ottimo strumento per osservare come cambia il tempo in una giornata o come cresce e fiorisce un fiore nell'arco di alcuni mesi. Puoi anche utilizzare le tecniche del time-lapse per realizzare la tua animazione in stop-motion.

Avvia una sessione di fotografia time-lapse digitando nel terminale il seguente comando per creare una nuova directory e accedervi. In questo modo, puoi conservare tutti i file acquisiti in un unico posto:

```
mkdir timelapse
cd timelapse
```

Quindi inizia a fotografare digitando:

```
rpicam-still --width 1920 --height 1080 -t 100000 \
  --timelapse 10000 -o %05d.jpg
```

Il nome del file di output è un po' diverso questa volta: `%05d` indica a `rpicam-still` di usare i numeri, partendo da 00000 e proseguendo in ordine crescente, come suggerisce il nome del file. Senza questa funzione, il sistema sovrascriverebbe automaticamente le immagini meno recenti ogni volta che ne scatta una nuova, e tu avresti una sola immagine che testimonia tutto il tuo impegno.

Le opzioni `--width` e `--height` controllano la *risoluzione* delle immagini acquisite. In questo caso, impostiamo le immagini a una larghezza di 1920 pixel e un'altezza di 1080 pixel, la stessa risoluzione di un file video Full HD.

L'opzione `-t` agisce come già abbiamo visto in precedenza, impostando un timer per la durata del funzionamento della fotocamera. In questo caso, è impostata su 100.000 millisecondi (100 secondi).

Infine, l'opzione `--timelapse` indica a `rpicam-still` quanto tempo attendere tra un'immagine e l'altra. Qui è impostata su 10.000 millisecondi (dieci secondi). Poiché non verrà scattata alcuna foto prima che siano trascorsi i primi dieci secondi, otterrai un totale di nove foto.

Lascia `rpicam-still` in funzione per 100 secondi, poi apri la directory `timelapse` in Gestore file. Vedrai nove singole foto, ciascuna etichettata con un numero che inizia con `00000` (Figura 8-9).

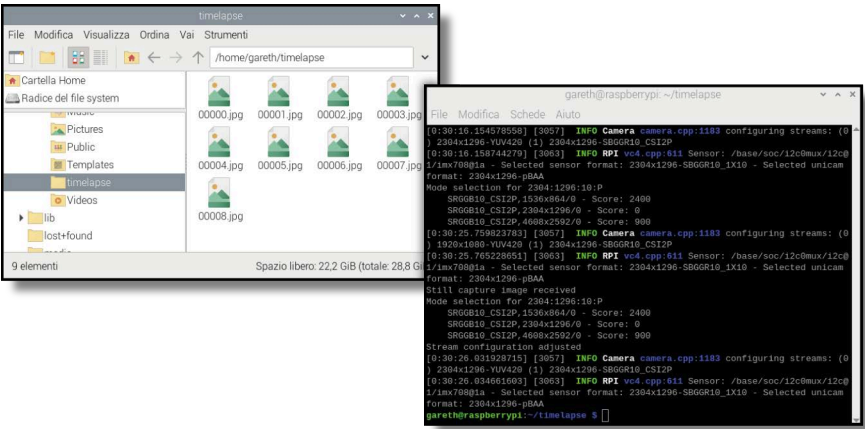


Figura 8-9 Foto scattate durante una sessione di time-lapse

Se desideri combinare queste immagini per creare un'animazione, utilizza lo strumento `ffmpeg`. Digita:

```
ffmpeg -r 0.5 -i %05d.jpg -r 15 animation.mp4
```

Questo indica a `ffmpeg` di interpretare le immagini acquisite come se fossero un video eseguito a 0,5 fotogrammi al secondo e di usarle per produrre un video animato che viene eseguito a 15 fotogrammi al secondo.

Fai doppio clic sul file `animation.mp4` per riprodurlo in VLC. Le foto scattate verranno visualizzate una dopo l'altra (Figura 8-10).

Per rendere l'animazione più veloce, prova a modificare la frequenza dei fotogrammi in ingresso da 0,5 fotogrammi al secondo a 1 o più; per renderla più lenta, prova a ridurla a 0,2 o meno.

Perché non provi a realizzare il tuo video in stop-motion? Posiziona dei giocattoli davanti alla fotocamera e inizia una sessione di time-lapse, quindi modificane la posizione subito dopo aver scattato ogni foto. Ricorda di togliere le mani dall'inquadratura prima che la foto venga scattata.

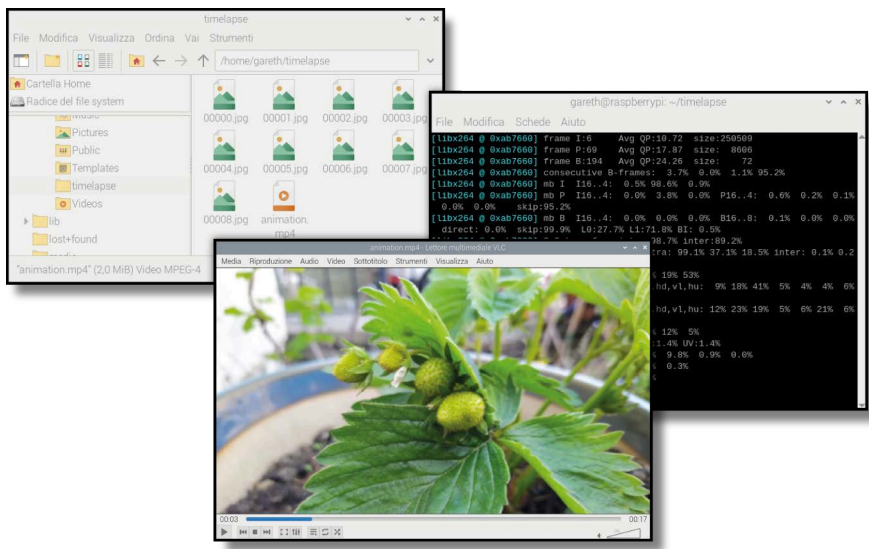


Figura 8-10 Riproduzione di un'animazione time-lapse

Impostazioni avanzate della fotocamera

Sia `rpicam-still` che `rpicam-vid` supportano una serie di parametri avanzati, che consentono un controllo più preciso su impostazioni come la risoluzione, ovvero la dimensione dell'immagine o del video acquisito. Le immagini e i video con una risoluzione più elevata sono di qualità superiore, ma occupano una quantità di spazio di archiviazione maggiore, quindi presta attenzione quando fai le prove.

`rpicam-still` e `rpicam-vid`

Le impostazioni riportate di seguito possono essere utilizzate sia con `rpicam-still` che con `rpicam-vid` aggiungendole al comando digitato nel terminale.

`--autofocus-mode`

Configura il sistema di messa a fuoco automatica sul Camera Module 3 di Raspberry Pi. Le opzioni disponibili sono: **`continuous`**, la modalità predefinita; **`manual`**, che disabilita completamente la messa a fuoco automatica; e **`auto`**, che esegue una singola operazione di messa a fuoco automatica al primo avvio della fotocamera. Questa impostazione non ha effetto su altre versioni del Camera Module.

--autofocus-range

Imposta l'intervallo per il sistema di messa a fuoco automatica del Camera Module 3 di Raspberry Pi. Se il sistema di messa a fuoco automatica fatica a inquadrare il soggetto, modificare la distanza può essere d'aiuto. Le opzioni possibili sono: **normal**, l'impostazione predefinita; **macro**, che privilegia gli oggetti ravvicinati; e **full**, che può mettere a fuoco sia gli oggetti estremamente vicini che quelli molto lontani.

--lens-position

Controlla manualmente il punto di messa a fuoco dell'obiettivo, da utilizzare con l'impostazione **--autofocus-mode manual**. Ciò consente di impostare il punto di messa a fuoco dell'obiettivo utilizzando un'unità di misura chiamata *diottria*, che è pari a uno diviso la distanza del punto di messa a fuoco in metri. Per impostare la messa a fuoco su 0,5 m (50 cm), ad esempio, utilizza **--lens-position 2**; per impostare la messa a fuoco su 10 m, utilizza **--lens-position 0.1**. Un valore di 0,0 rappresenta un punto di messa a fuoco all'infinito, la distanza massima che la fotocamera può mettere a fuoco.

--width --height

Imposta la risoluzione dell'immagine o del video. Per registrare un video Full HD (1920×1080), ad esempio, utilizza questi argomenti con `rpicam-vid`:

```
-t 10000 --width 1920 --height 1080 -o biggest.h264
```

--rotation

Ruota l'immagine da 0 gradi, valore predefinito, a 90, 180 e 270 gradi. Se la fotocamera è montata in modo che il cavo a nastro non fuoriesce dalla parte inferiore, questa impostazione consente di acquisire immagini e video nella parte giusta.

--hflip --vflip

Capovolge l'immagine o il video lungo l'asse orizzontale, come farebbe uno specchio, e/o l'asse verticale.

--sharpness

Consente di rendere più chiara l'immagine o il video applicando un filtro di nitidezza. I valori superiori a 1,0 aumentano la nitidezza rispetto a quella predefinita; i valori inferiori a 1,0 la diminuiscono.

--contrast

Aumenta o diminuisce il contrasto dell'immagine o del video acquisito. I valori superiori a 1,0 aumentano il contrasto rispetto a quello predefinito. I valori inferiori a 1,0 lo riducono.

--brightness

Aumenta o diminuisce la luminosità dell'immagine o del video. Se diminuisce il valore rispetto a quello predefinito di 0,0, l'immagine diventerà più scura fino a raggiungere il valore minimo di -1,0, e avrai un'immagine completamente nera. Aumentando il valore si schiarisce l'immagine fino a raggiungere il valore massimo di 1,0, e avrai un'immagine completamente bianca.

--saturation

Aumenta o diminuisce la saturazione del colore dell'immagine o del video. Diminuendo il valore da 1,0 predefinito, i colori si attenuano fino a raggiungere il valore minimo di 0,0, e avrai un'immagine completamente in scala di grigi senza alcun colore. I valori superiori a 1,0 rendono i colori più vivaci.

--ev

Imposta un valore di compensazione dell'esposizione, da -10 a 10, che controlla il funzionamento del comando del guadagno della fotocamera. Di solito, il valore predefinito di 0 offre i risultati migliori. Se la fotocamera acquisisce immagini troppo scure, puoi aumentare il valore; se sono troppo chiare, diminuisce il valore.

--metering

Imposta la modalità di misurazione per i controlli automatici dell'esposizione e del guadagno. Il valore predefinito, **centre**, in genere offre i risultati migliori; puoi sovrascrivere questo valore impostando la misurazione **spot** o **average**.

--exposure

Consente di passare dalla modalità di esposizione predefinita, **normal**, alla modalità di esposizione **sport** progettata per i soggetti in rapido movimento (e viceversa).

--awb

Consente di modificare l'algoritmo di bilanciamento automatico del bianco dalla modalità automatica predefinita a: **incandescent**, **tungsten**, **fluorescent**, **indoor**, **daylight** o **cloudy**.

rpicam-still

In rpicam-still sono disponibili le seguenti opzioni

-q

Imposta la qualità dell'immagine JPEG acquisita, da 0 a 100, dove 0 corrisponde alla qualità minima e alla dimensione minima del file e 100 alla qualità massima e alla dimensione massima del file. La qualità predefinita è 93.

--datetime

Utilizza la data e l'ora correnti (nel formato due cifre per il mese, due cifre per il giorno, minuti, ore, secondi) come nome del file di output. Da utilizzare al posto di **-o**.

--timestamp

Simile a **--datetime**, ma imposta il nome del file di output sul numero di secondi dall'inizio del 1970, noto come *epoca di UNIX*.

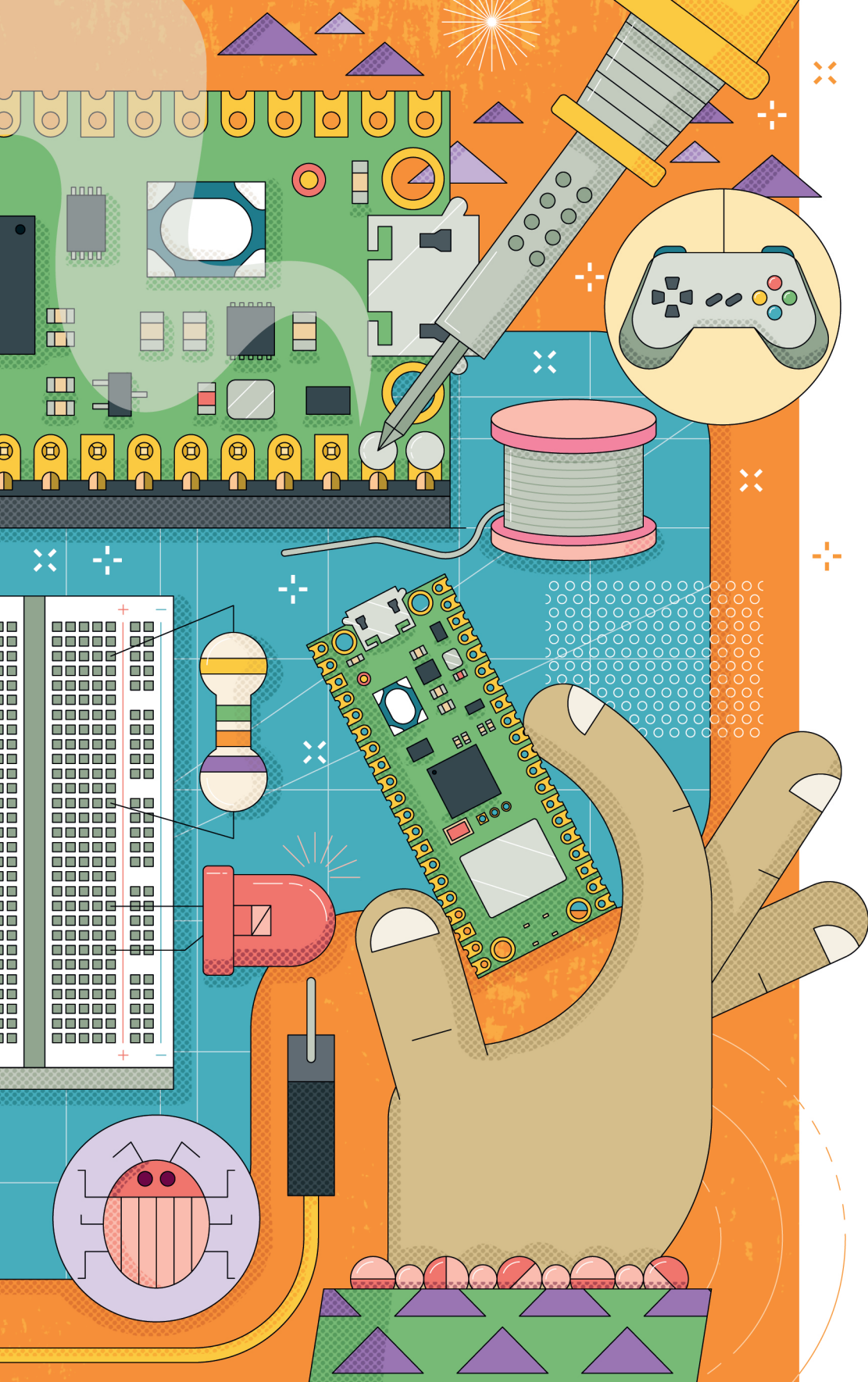
-k

Acquisisce un'immagine fissa quando premi il tasto Invio, invece di acquisirla automaticamente dopo un ritardo. Se desideri annullare un'acquisizione, digita **x** e premi **INVIO**. Funziona meglio con il timeout, **-t**, impostato a 0. rpicam-vid ha un'opzione **-k** simile, ma funziona in modo leggermente diverso e utilizza il tasto Invio per avviare o mettere in pausa la registrazione in modalità di registrazione. Al termine, digita **x** e premi **INVIO** per uscire.

INFORMAZIONI PIÙ DETTAGLIATE

Questo capitolo tratta le opzioni più comuni per le applicazioni rpicam, ma ce ne sono molte altre. Puoi trovare una descrizione tecnica completa di rpicam, con anche le differenze con le vecchie applicazioni raspivid e raspistill, all'indirizzo rptl.io/camera-software.





Capitolo 9

Raspberry Pi Pico e Pico W

Raspberry Pi Pico e Pico W danno una dimensione completamente nuova ai progetti di physical computing

Raspberry Pi Pico e Pico W sono *schede di sviluppo per microcontrollori*. Sono stati progettati per sperimentare con il physical computing mediante un tipo speciale di processore: un *microcontrollore*. Grandi come una gomma da masticare, Raspberry Pi Pico e Pico W sono sorprendentemente potenti grazie al chip al centro della scheda: un microcontrollore RP2040.

Raspberry Pi Pico e Pico W non sono pensati per sostituire il Raspberry Pi, che rientra in una categoria diversa di dispositivi, noti come computer a scheda singola. Puoi usare il Raspberry Pi per giocare, scrivere software o navigare sul Web, come hai visto in precedenza in questa guida. Raspberry Pi Pico è invece progettato per progetti di physical computing, dove viene utilizzato per controllare diversi tipi di elementi: da LED e pulsanti a sensori, motori e persino altri microcontrollori.

Grazie ai pin di ingresso/uscita generici (GPIO), puoi eseguire operazioni di physical computing anche con un Raspberry Pi, ma ci sono dei vantaggi nell'utilizzare una scheda di sviluppo per microcontrollori anziché un computer a scheda singola. Raspberry Pi Pico è più piccolo, più economico e offre alcune funzionalità specifiche per il physical computing, come timer ad alta precisione e sistemi di input/output programmabili.

Questo capitolo non vuole essere una guida esaustiva di ciò che puoi fare con Raspberry Pi Pico e Pico W, e non è necessario acquistare un Pico per ottenere il massimo da Raspberry Pi. Se possiedi già un Raspberry Pi Pico o Pico W, o semplicemente vuoi saperne di più, questo capitolo ti servirà come introduzione alle loro caratteristiche principali.

Per una panoramica completa delle caratteristiche di Raspberry Pi Pico e Pico W, ti consigliamo il libro *Get started with MicroPython on Raspberry Pi Pico*.

Presentazione introduttiva di Raspberry Pi Pico

Raspberry Pi Pico ("Pico" in breve) è molto più piccolo anche di un Raspberry Pi Zero, il più compatto della gamma di computer a scheda singola di Raspberry Pi. Nonostante questo, include tantissime funzioni, tutte accessibili usando i pin intorno al bordo della scheda. È disponibile in due versioni: Raspberry Pi Pico e Raspberry Pi Pico W, più avanti tratteremo le differenze tra le due.

La **Figura 9-1** mostra Raspberry Pi Pico visto dall'alto. Se osservi i bordi più lunghi, vedrai delle sezioni color oro con piccoli fori. Questi sono i pin che forniscono al microcontrollore RP2040 le connessioni al mondo esterno, note come ingresso/uscita (o input/output, I/O).

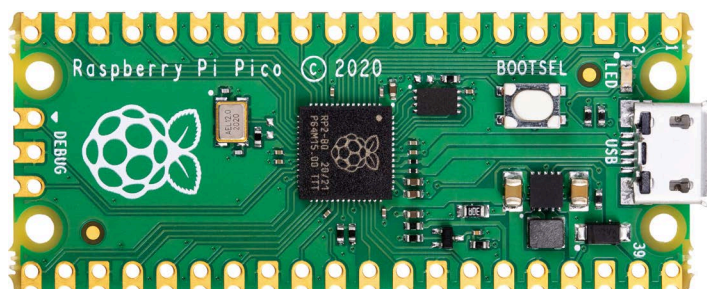


Figura 9-1 Parte superiore della scheda

I pin presenti su Pico sono molto simili a quelli che compongono il connettore general-purpose input/output (GPIO) su Raspberry Pi, ma, mentre la maggior parte dei computer a scheda singola Raspberry Pi viene fornita con i pin fisici di metallo già collegati, Raspberry Pi Pico e Pico W non ne sono dotati.

Se vuoi acquistare un Pico con terminali montati, prendi invece in considerazione Raspberry Pi Pico H e Pico WH. I modelli senza terminali hanno uno scopo preciso: guarda il bordo esterno del circuito stampato, noterai che ha delle rientranze, con piccoli tagli circolari (**Figura 9-2**).

Queste rientranze creano quello che viene chiamato un *circuito stampato a mezzi fori*, che può essere saldato sopra ad altri circuiti senza utilizzare alcun pin metallico. È molto utile nelle applicazioni in cui è necessario contenere l'altezza per ottenere un progetto finito più piccolo. Se acquisti un gadget basato su Raspberry Pi Pico o Pico W, quest'ultimo sarà quasi certamente installato mediante i mezzi fori.

I fori appena all'interno delle rientranze sono destinati ad accogliere i pin maschi da 2,54 mm. Riconoscerai che si tratta dello stesso tipo di pin utilizzati sul connettore GPIO del Raspberry Pi più grande. Saldandoli verso il basso, puoi inserire Pico in una *breadboard senza saldature* per rendere semplice il collegamento e lo scollegamento di nuovo hardware e fare quindi esperimenti, nonché creare prototipi, in tutta semplicità.

Il chip al centro di Pico (Figura 9-3) è un microcontrollore RP2040. Si tratta di un *circuito integrato personalizzato* (IC), progettato e costruito da Raspberry Pi per fungere da "mente" di Pico e di altri dispositivi basati su microcontrollore. Guardando da vicino potrai vedere il logo Raspberry Pi inciso nella parte superiore del chip insieme a una serie di lettere e numeri che permettono agli ingegneri di sapere quando e dove è stato prodotto il chip.

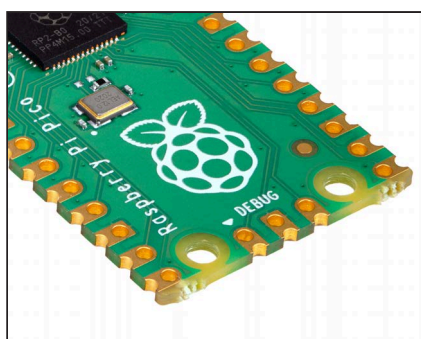


Figura 9-2
Mezzi fori

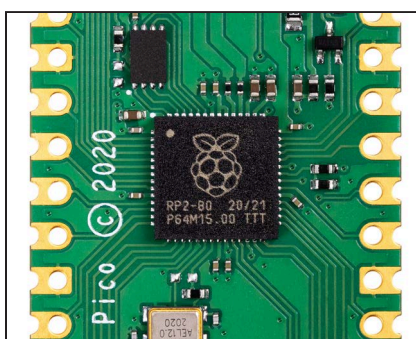


Figura 9-3
Chip RP2040

Nella parte superiore di Pico è presente una *porta micro USB* (Figura 9-4). Fornisce l'energia necessaria per far funzionare Pico e invia e riceve i dati che consentono a Pico di comunicare con Raspberry Pi o un altro computer tramite la porta USB. In questo modo potrai caricare i programmi su Pico.

Se guardi frontalmente la porta micro USB di Pico vedrai che è più stretta in basso e più larga in alto. Osserva un cavo micro USB, noterai che il suo connettore è sagomato allo stesso modo.

Il cavo micro USB entrerà nella porta micro USB del tuo Pico solo in una direzione. Quando lo colleghi, assicurati di allineare le parti strette e larghe nel verso giusto, altrimenti potresti danneggiare Pico cercando di inserire forzatamente il cavo micro USB nel modo errato.

Appena sotto la porta micro USB c'è un piccolo pulsante contrassegnato come "BOOTSEL" (Figura 9-5). "BOOTSEL" è l'abbreviazione di *boot selection* (selezione di avvio) e consente di passare tra le due modalità di avvio alla prima

accensione di Pico. Utilizzerai il pulsante di selezione di avvio in seguito, quando Pico sarà pronto per la programmazione.

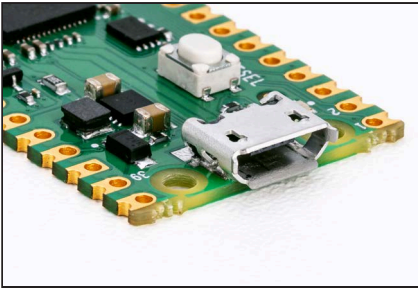


Figura 9-4
Porta micro USB

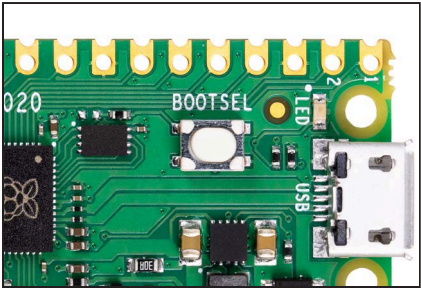


Figura 9-5
Interruttore di selezione avvio

Nella parte inferiore di Pico sono presenti tre piccole piazzole dorate sopra le quali è presente la scritta "DEBUG" (**Figura 9-6**). Sono progettate per eseguire il *debug*, ossia trovare errori, nei programmi in esecuzione su Pico mediante uno speciale strumento chiamato *debugger*. All'inizio non avrai bisogno di usare il terminale di debug, ma potresti trovarlo utile quando scriverai programmi più lunghi e complessi. In alcuni modelli di Raspberry Pi Pico, le piazzole di debug sono sostituite da un piccolo connettore a tre pin.

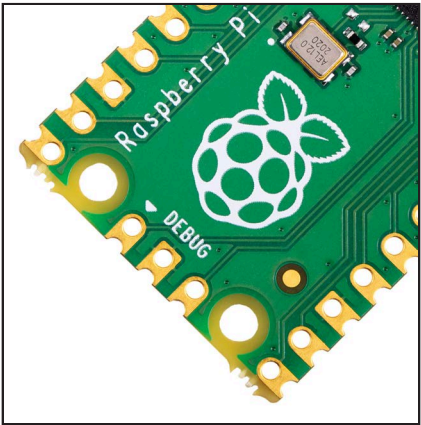


Figura 9-6
Piazzole di debug

Se capovolgi Pico vedrai che la parte inferiore presenta delle scritte (**Figura 9-7**). Il testo stampato è noto come *strato di serigrafia* ed etichetta ciascuno dei pin con la sua funzione principale. Vedrai diciture come "GP0" e "GP1", "GND", "RUN" e "3V3". Se dimentichi a cosa serve un pin, queste etichette te lo indicheranno, tuttavia siccome non potrai vederle quando Pico è inserito in

una breadboard, per semplificare la consultazione abbiamo inserito nella guida gli schemi di piedinatura.

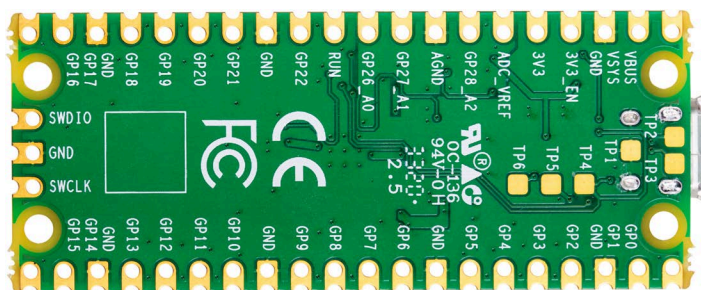


Figura 9-7 Etichettatura inferiore

Potresti aver notato che non tutte le etichette sono allineate con i rispettivi pin. I piccoli fori nella parte superiore e inferiore della scheda sono fori di montaggio: consentono di fissare Pico nei progetti in modo più permanente mediante viti o dadi e bulloni. Nei punti in cui i fori non consentono l'etichettatura, le etichette si trovano più in alto o in basso sulla scheda: guardando in alto a destra, "VBUS" è il primo pin a destra, "VSYS" il secondo e "GND" il terzo.

Vedrai anche alcune piazzole dorate piatte etichettate con "TP" e un numero. Si tratta di punti di prova progettati per consentire agli ingegneri di verificare rapidamente il funzionamento di Raspberry Pi Pico dopo l'assemblaggio in fabbrica, non dovrai necessariamente utilizzarli. A seconda della piazzola di test, l'ingegnere potrebbe utilizzare un multimetro o un oscilloscopio per verificare che Pico funzioni correttamente prima di essere imballato e spedito.

Se hai un Raspberry Pi Pico W o Pico WH, troverai un altro pezzo di hardware sulla scheda: un rettangolo di metallo argentato (**Figura 9-8**). Si tratta di una schermatura per il modulo wireless, come quello presente su Raspberry Pi 4 e Raspberry Pi 5, che serve a collegare Pico a una rete Wi-Fi o a dispositivi Bluetooth. È collegato a una piccola antenna che si trova nella parte inferiore della scheda, motivo per cui, su Raspberry Pi Pico W e Pico WH, il connettore o le piazzole di debug si trovano più vicini al centro della scheda.

Pin del terminale

Puoi notare che Raspberry Pi Pico o Pico W, sono completamente piatti. Non sono presenti pin metallici che sporgono dai lati, come quelli che si trovano sul connettore GPIO del Raspberry Pi o su Raspberry Pi Pico H e Pico WH. Puoi utilizzare i mezzi fori per collegare Pico a un altro circuito stampato o per saldare direttamente i fili in un progetto in cui Pico sarà fissato in modo permanente.

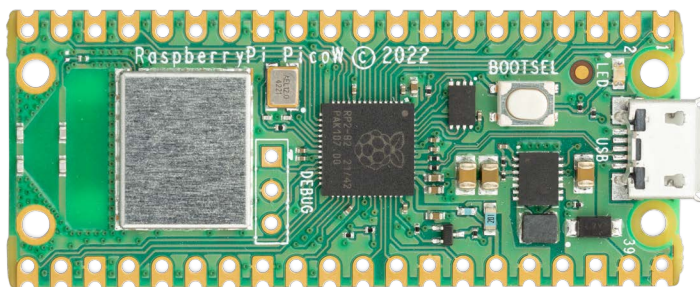


Figura 9-8 Il modulo wireless e l'antenna di Raspberry Pi Pico W

Il modo più semplice per utilizzare Pico, tuttavia, è collegarlo a una bread-board e per questo dovrai collegare i terminali dei pin. Per inserire i terminali dei pin su Raspberry Pi Pico è necessario un saldatore, che riscalda i pin e le piazzole in modo da poterli connettere utilizzando una *lega per saldatura*, ossia una lega metallica morbida.

Per i progetti introduttivi di questo capitolo, non dovrai collegare alcun pin a Pico. Se però vuoi realizzare progetti più complessi, puoi scoprire come saldare i pin in modo sicuro nel Capitolo 1 di *Get Started with MicroPython on Raspberry Pi Pico*. Puoi anche verificare se il tuo rivenditore Raspberry Pi di fiducia ha una versione di Raspberry Pi Pico con pin terminale già saldati. Si chiamano Raspberry Pi Pico H e Raspberry Pi Pico WH, rispettivamente per le versioni standard e Wi-Fi.

Installazione di MicroPython

Puoi programmare Raspberry Pi Pico in Python, proprio come Raspberry Pi. Tuttavia, poiché si tratta di un microcontrollore e non di un computer a scheda singola, necessita di una particolare versione, ossia *MicroPython*.

MicroPython funziona come la versione standard di Python e puoi usare lo stesso IDE Thonny che usi per programmare Raspberry Pi. In MicroPython, tuttavia, mancano alcune caratteristiche della versione standard di Python, mentre ne sono presenti altre, come le librerie speciali per i microcontrollori e le loro periferiche.

Prima di poter programmare Pico in MicroPython, è necessario scaricare e installare il *firmware*. Inizia collegando un cavo micro USB alla relativa porta di Pico, assicurandoti che sia rivolto nel senso corretto prima di inserirlo con cautela.

ATTENZIONE

Per installare MicroPython su Pico, dovrai scaricarlo da Internet. Dovrai farlo solo una volta: dopo aver installato MicroPython, questo rimarrà su Pico a meno che tu non decida di sostituirlo con qualcos'altro in futuro.



Tieni premuto il pulsante "**BOOTSEL**" sulla parte superiore di Pico. Tenendolo ancora premuto, collega l'altra estremità del cavo micro USB a una delle porte USB del computer. Conta fino a tre, quindi rilascia il pulsante.

NOTA

Su macOS, quando colleghi Pico al computer ti verrà chiesto se vuoi consentire all'accessorio di connettersi. Dovrai fare clic su **Sì** per autorizzare il collegamento. Dopo aver installato MicroPython su Pico, macOS potrebbe porre la domanda una seconda volta, perché ora sembra un dispositivo diverso.



Dopo qualche secondo dovresti visualizzare Pico come unità rimovibile, come se avessi collegato un'unità flash USB o un disco rigido esterno. Su Raspberry Pi vedrai un pop-up che ti chiede se desideri aprire l'unità nel Gestore file. Assicurati che **Apri nel gestore file** sia selezionato e fai clic su **OK**.

Nella finestra del Gestore file vedrai due file su Pico (**Figura 9-9**): **INDEX.HTM** e **INFO_UF2.TXT**. Il secondo file contiene informazioni su Pico, come la versione di bootloader in esecuzione. Il primo file, **INDEX.HTM**, è un link al sito Web di Raspberry Pi Pico. Fai doppio clic su questo file oppure apri il browser Web e digita **rptl.io/microcontroller-docs** nella barra degli indirizzi.

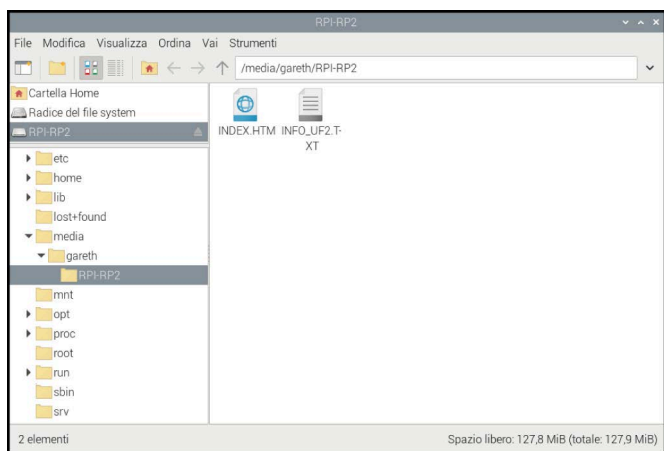


Figura 9-9 Vedrai due file su Raspberry Pi Pico

Quando si apre la pagina Web, vedrai informazioni sul microcontrollore Raspberry Pi e sulle schede di sviluppo, tra cui Raspberry Pi Pico e Pico W. Fai clic sulla casella MicroPython per andare alla pagina di download del firmware. Scorri verso il basso fino alla sezione **Drag-and-Drop MicroPython**, come mostrato nella **Figura 9-10**, e trova il link per la versione di MicroPython per la tua scheda. È presente una versione per Raspberry Pi Pico e Pico H e un'altra per Raspberry Pico W e Pico WH. Fai clic sul link per scaricare il file UF2 appropriato. Se per sbaglio scarichi il file sbagliato, non preoccuparti; puoi tornare alla pagina e scaricare il nuovo firmware sul tuo dispositivo seguendo la stessa procedura.

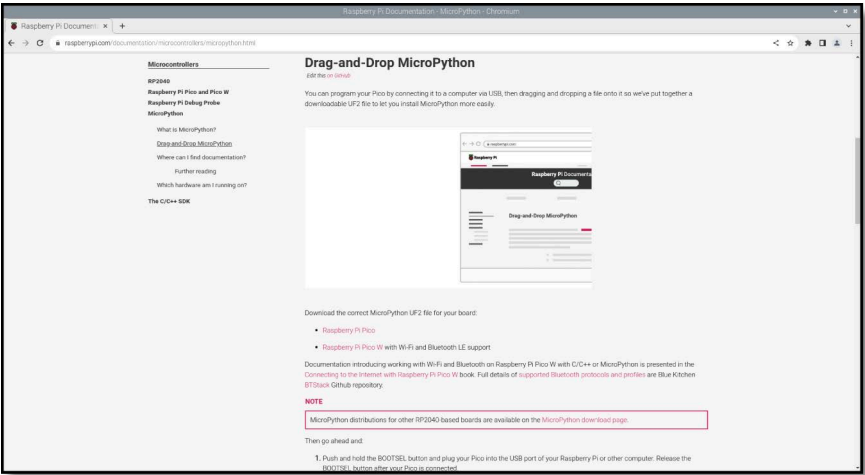


Figura 9-10 Fai clic sul link per scaricare il firmware MicroPython

Apri una nuova finestra di Gestore file, quindi vai alla cartella **Downloads** e cerca il file appena scaricato. Il nome sarà: "**rp2-pico**" o "**rp2-pico-w**" seguito da una data, da testo e da numeri, che servono a distinguere le diverse versioni del firmware, e dall'estensione "**uf2**".



NOTA

Per trovare la cartella Downloads in Raspberry Pi, fai clic sul menu Raspberry Pi, scegli **Accessori** e apri **Gestore file PCManFM**. Quindi, cerca **Downloads** nell'elenco delle cartelle a sinistra della finestra **Gestore file PCManFM**. A seconda di quante cartelle hai su Raspberry Pi, potrebbe essere necessario scorrere l'elenco per trovarla.

Fai clic e tieni premuto il pulsante del mouse sul file UF2, quindi trascinalo nell'altra finestra aperta sull'unità di archiviazione rimovibile di Pico. Passa il mouse sopra la finestra, quindi rilascia il pulsante per rilasciare il file su Pico come mostrato nella **Figura 9-11**.

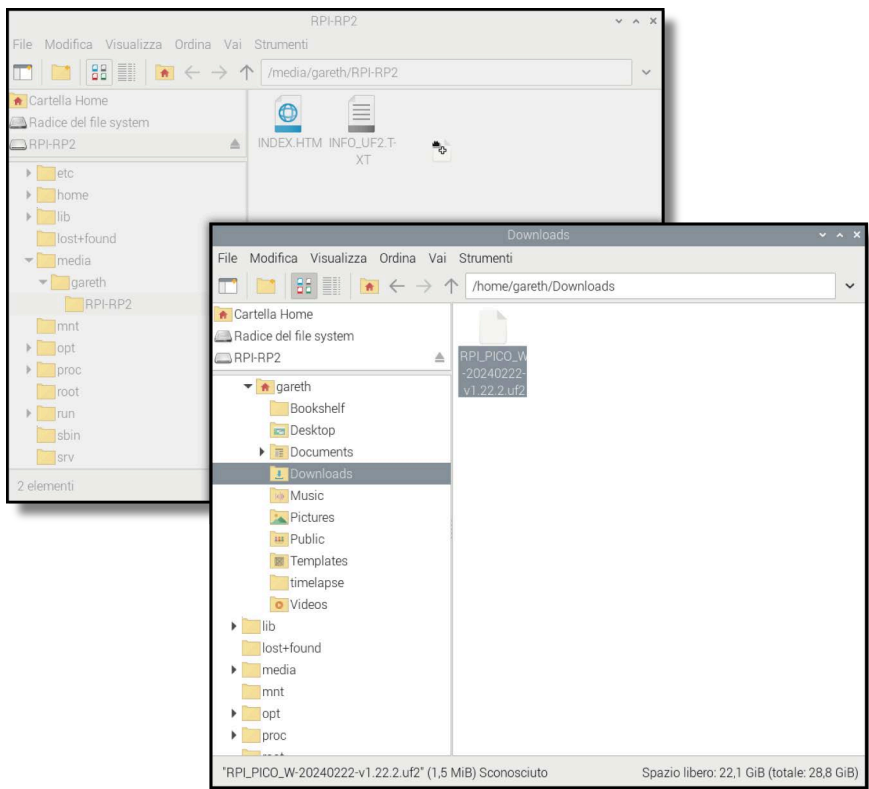


Figura 9-11 Trascina il file del firmware MicroPython su Raspberry Pi Pico

Dopo pochi secondi vedrai la finestra dell'unità Pico scomparire dal **Gestore file PCManFM, Explorer o Finder** e potresti anche vedere un avviso che indica che un'unità è stata rimossa senza essere espulsa. Non preoccuparti, è normale che accada! Quando hai trascinato il file del firmware MicroPython su Pico, gli hai indicato di eseguire il flashing del firmware sulla memoria interna. Per fare ciò Pico esce dalla modalità speciale impostata con il pulsante "BOOTSEL", esegue il flashing del nuovo firmware quindi lo carica, il che significa che Pico ora esegue MicroPython.

Complimenti: ora puoi iniziare a usare MicroPython su Raspberry Pi Pico!



ULTERIORI LETTURE

La pagina Web collegata da **INDEX.HTM** non solo consente di scaricare MicroPython, ma ospita numerose risorse aggiuntive. Fai clic sulle schede e scorri per accedere alle guide, ai progetti e al file *datobook*: una libreria di documentazione tecnica dettagliata e completa su argomenti che vanno dal funzionamento interno del microcontrollore RP2040 su cui si basa Pico alla programmazione in Python e C/C++.

I pin di Pico

Pico comunica con l'hardware attraverso una serie di pin posti lungo entrambi i bordi. La maggior parte dei pin funziona come ingresso/uscita programmabile (PIO), ossia può essere programmata per agire come ingresso o come uscita e non ha uno scopo predefinito fino a quando non gliene viene assegnato uno. Alcuni pin hanno funzionalità extra e modalità alternative per comunicare con hardware più complesso, altri hanno funzioni specifiche, fornendo connessioni ad esempio per l'alimentazione.

I 40 pin di Raspberry Pi Pico sono etichettati sul lato inferiore della scheda, tre di questi presentano etichette con i numeri anche sulla parte superiore della scheda: Pin 1, Pin 2 e Pin 39. Queste etichette ti aiutano a ricordare come funziona la numerazione: il pin 1 si trova in alto a sinistra guardando la scheda dall'alto, con la porta micro USB sul lato superiore. Il pin 20 è in basso a sinistra, il pin 21 in basso a destra e il pin 39 immediatamente sotto il pin in alto a destra con il pin 40 non etichettato sopra. L'etichettatura sul lato inferiore è più accurata, ma non sarà possibile vederla quando Pico è collegato a una breadboard.

Su Raspberry Pi Pico, i pin sono solitamente indicati con le loro funzioni (vedi **Figura 9-12**) piuttosto che con un numero. Esistono diverse tipologie di pin, ognuna delle quali ha una funzione particolare:

- ▶ **3V3** (*tensione 3,3 volt*): una fonte di alimentazione a 3,3 V generata dall'ingresso VSYS. Questo alimentatore può essere acceso e spento utilizzando il pin 3V3_EN sopra di esso, che spegne anche Pico.
- ▶ **VSYS** (*tensione ~2-5 volt*): un pin collegato direttamente all'alimentatore interno di Pico, che non può essere spento senza spegnere anche Pico.
- ▶ **VBUS** (*tensione 5 volt*): una fonte di alimentazione a 5 V dalla porta micro USB di Pico e utilizzata per alimentare hardware che necessita di più di 3,3 V.

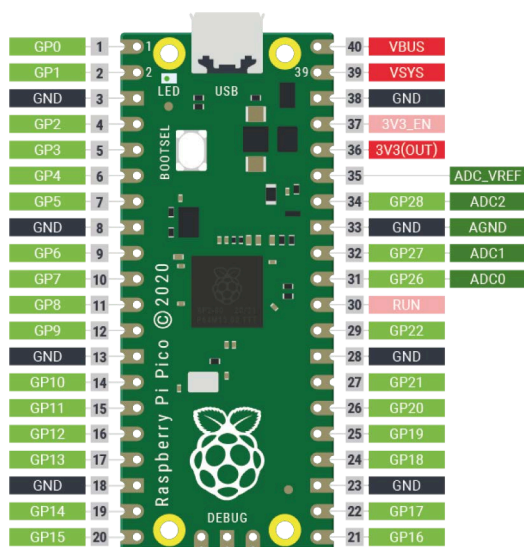


Figura 9-12 Pin di Raspberry Pi Pico, visti dalla parte superiore della scheda

- ▶ **GND** (*pin di massa a 0 volt*): un collegamento a massa, utilizzato per completare un circuito collegato alla fonte di alimentazione. Molti pin GND si trovano in vari punti intorno a Pico per facilitare il cablaggio.
- ▶ **GPxx** (*General-purpose input/output numero pin "xx"*): i pin GPIO disponibili per il programma, etichettati da "GP0" a "GP28".
- ▶ **GPxx_ADCx** (*General-purpose input/output numero pin "xx", con numero di input analogico "x"*): un pin GPIO che termina con "ADC" e un numero può essere utilizzato come ingresso analogico e come ingresso o uscita digitale, ma non contemporaneamente per entrambe le funzioni.
- ▶ **ADC_VREF** (*riferimento di tensione del convertitore analogico-digitale, ADC*): uno speciale pin di ingresso che imposta una tensione di riferimento per eventuali ingressi analogici.
- ▶ **AGND** (*pin di massa a 0 volt per il convertitore analogico/digitale, ADC*): una speciale connessione a massa da utilizzare con il pin ADC_VREF.
- ▶ **RUN** (*attiva o disattiva Pico*): il pin RUN consente di avviare e arrestare Pico da un altro microcontrollore.

Collegamento di Thonny a Pico

Inizia caricando Thonny: fai clic sul menu Raspberry Pi in alto a sinistra sullo schermo, sposta il mouse sulla sezione **Programmazione** e fai clic su **Thonny**.

Con Pico collegato a Raspberry Pi, fai clic sulla scritta **Python 3 Locale** in basso a destra della finestra di Thonny. Verrà mostrato l'interprete in uso, che è responsabile di recepire le istruzioni che digiti per poi trasformarle in un codice che il computer, o il microcontrollore, può capire ed eseguire. Generalmente l'interprete è la copia di Python eseguita su Raspberry Pi, ma deve essere cambiato per eseguire i programmi in MicroPython su Pico.

Nell'elenco visualizzato cerca "MicroPython (Raspberry Pi Pico)" (**Figura 9-13**) e fai clic su di esso. Se non riesci a vederlo nell'elenco, controlla che Pico sia collegato correttamente al cavo micro USB e che il cavo micro USB sia collegato correttamente a Raspberry Pi o al computer.

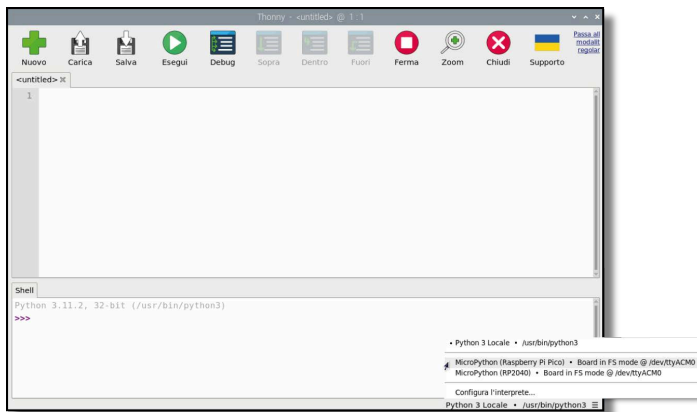


Figura 9-13 Scelta di un interprete Python



PROFESSIONISTI PYTHON

Questo capitolo presuppone che tu abbia familiarità con l'IDE Thonny e con la scrittura di semplici programmi Python. Se non l'hai ancora fatto, lavora sui progetti del Capitolo 5, *Programmare con Python* prima di continuare con questo capitolo.

SCELTA DELL'INTERPRETE

La scelta dell'interprete determina dove e come verrà eseguito il programma: se scegli **MicroPython (Raspberry Pi Pico)**, i programmi verranno eseguiti su Pico; mentre con **Python 3 Locale** i programmi verranno eseguiti su Raspberry Pi.

Se ti accorgi che i programmi non vengono eseguiti dove ti aspetteresti, controlla le impostazioni di Thonny per verificare quale interprete è impostato.



Il tuo primo programma MicroPython: Ciao a tutti!


Puoi verificare che tutto funzioni nello stesso modo in cui hai imparato a scrivere programmi Python su Raspberry Pi: scrivendo un semplice programma "Ciao a tutti!". Innanzitutto, fai clic sull'area Python Shell nella parte inferiore della finestra di Thonny, appena a destra dei simboli `>>>` in basso, e digita le seguenti istruzioni prima di premere il tasto **INVIO**.


```
print("Ciao mondo!")
```

Quando premi **INVIO**, vedrai che il programma viene immediatamente eseguito: Python risponderà nella stessa area shell con il messaggio "**Ciao mondo!**" (Figura 9-14), proprio come hai chiesto. La shell infatti è collegata direttamente all'interprete MicroPython in esecuzione su Pico, che ha il compito di esaminare le istruzioni scritte e interpretarne il significato. Questa modalità interattiva funziona come quando programmi Raspberry Pi: le istruzioni scritte nell'area della shell vengono eseguite immediatamente, senza alcun ritardo. L'unica differenza è che vengono inviate a Pico per essere eseguite e ogni risultato, in questo caso il messaggio "Ciao a tutti!", viene inviato a Raspberry Pi per essere visualizzato.

Non devi programmare Pico (o Raspberry Pi) in modalità interattiva. Fai clic sull'area script al centro della finestra Thonny, quindi digita di nuovo il programma:

```
print("Ciao mondo!")
```

Una volta premuto il tasto **INVIO** questa volta non succederà nulla, ma passerai a una riga vuota nell'area dello script. Per far funzionare questa versione del programma, dovrai fare clic sull'icona **Esegui**  nella barra degli strumenti di Thonny.

Anche se si tratta di un programma semplice, ti consigliamo di prendere l'abitudine di salvare il lavoro. Prima di eseguire il programma, fai clic sull'icona  **Salva**. Ti verrà chiesto se vuoi salvare il tuo programma su "Que-

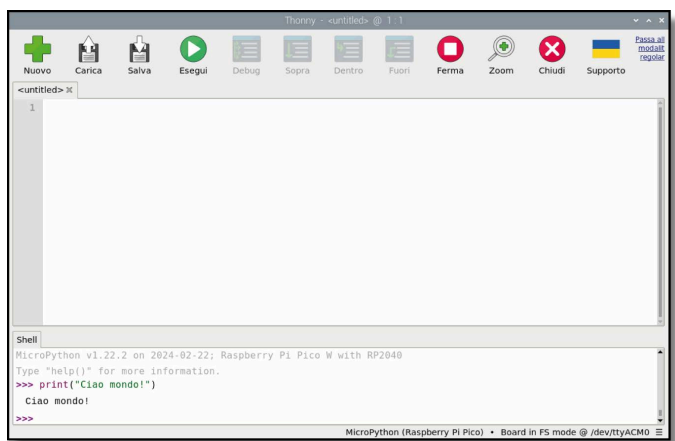


Figura 9-14 MicroPython stampa il messaggio "Ciao a tutti!" nell'area shell

sto computer", ossia su Raspberry Pi o su qualsiasi altro computer su cui stai eseguendo Thonny, oppure su **"Raspberry Pi Pico"** (Figura 9-15). Fai clic su **Raspberry Pi Pico**, poi digita un nome descrittivo come **Ciao mondo.py** e fai clic sul pulsante OK.

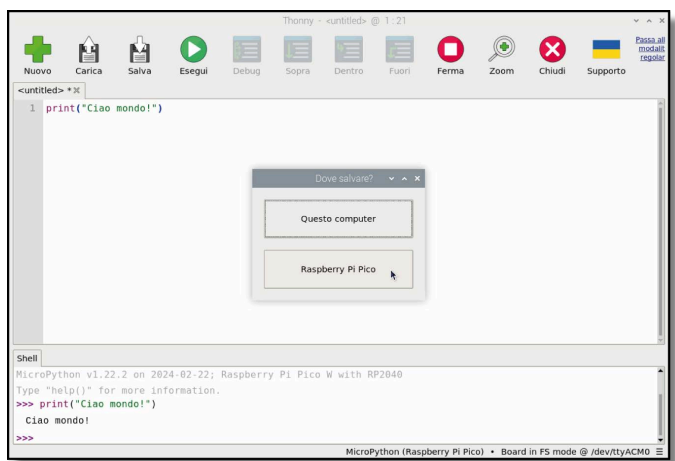




Figura 9-15 Salvataggio di un programma su Pico

Fai nuovamente clic sull'icona **Esegui** . Verrà eseguito automaticamente su Pico. Visualizzerai due messaggi nell'area shell nella parte inferiore della finestra di Thonny:

```
>>> %Run -c $EDITOR_CONTENT
```

Ciao mondo!

La prima di queste righe è un'istruzione di Thonny che indica all'interprete MicroPython su Pico di eseguire il codice dell'area dello script (EDITOR_CONTENT). Mentre la seconda è l'output del programma, ossia il messaggio che hai detto di stampare a MicroPython. Complimenti, ora hai scritto due programmi MicroPython in modalità interattiva e script e li hai eseguiti correttamente su Pico.

C'è solo un altro tassello del puzzle: caricare nuovamente il programma. Chiudi Thonny premendo la X in alto a destra della finestra su Windows o Linux (usa il pulsante di chiusura in alto a sinistra della finestra su macOS), quindi avvia nuovamente Thonny. Questa volta invece di scrivere un nuovo programma, fai clic sull'icona **Carica**  nella barra degli strumenti di Thonny. Ti verrà chiesto se vuoi salvarlo nuovamente su "Questo computer" o su "Raspberry Pi Pico". Fai clic su **Raspberry Pi Pico** e vedrai un elenco di tutti i programmi che hai salvato su Pico.

UN PICO PIENO DI PROGRAMMI

Quando dici a Thonny di salvare il tuo programma su Pico, significa che i programmi vengono memorizzati su Pico stesso. Se scolleghi Pico e lo colleghi a un altro computer, i programmi saranno ancora dove li hai salvati, ossia su Pico.



Cerca **Ciao_mondo.py** nell'elenco: se Pico è nuovo, sarà l'unico file presente. Fai clic per selezionarlo, quindi fai clic su OK. Il programma verrà caricato in Thonny, pronto per essere modificato o per essere eseguito nuovamente.

SFIDA: NUOVO MESSAGGIO

È possibile modificare il messaggio che il programma Python stampa come output? Useresti la modalità interattiva o la modalità script per aggiungere altri messaggi? Cosa succede se si rimuovono le parentesi o le virgolette dal programma e poi si cerca di eseguirlo di nuovo?



Il tuo primo programma di physical computing: Ciao LED!

Proprio come la stampa di "Ciao a tutti!" sullo schermo è un primo passo nell'apprendimento di un linguaggio di programmazione, riuscire a illuminare un LED è la tradizionale introduzione all'apprendimento del physical computing su una nuova piattaforma. Puoi iniziare anche senza componenti

aggiuntivi: Raspberry Pi Pico infatti dispone già di un piccolo LED, noto come *dispositivo a montaggio superficiale (o SMD) LED*.

Inizia trovando il LED: è il piccolo componente rettangolare a sinistra della porta micro USB nella parte superiore della scheda (**Figura 9-16**) e contrassegnato come "LED".

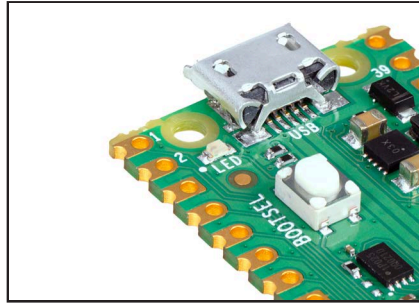



Figura 9-16
Il LED integrato si trova a sinistra del connettore micro USB

Il LED integrato è collegato a uno dei pin di ingresso/uscita per uso generico dell'RP2040, GP25. Si tratta di uno dei pin GPIO "mancanti" forniti dal microcontrollore RP2040, ma che non sono stati esposti su un pin fisico sul bordo di Pico. Sebbene non sia possibile collegare alcun hardware al pin (a parte il LED integrato), può essere trattato come qualsiasi altro pin GPIO all'interno dei programmi. Questo rappresenta un ottimo modo per aggiungere un output ai programmi senza bisogno di componenti aggiuntivi.

Fai clic sull'icona **Nuovo**  in Thonny e inizia a programmare con la seguente riga:

```
import machine
```

Questa breve riga di codice è fondamentale per lavorare con MicroPython su Pico. Carica, o *importa*, una raccolta di codice MicroPython nota come *libreria*, in questo caso, la libreria `machine`. La libreria `machine` contiene tutte le istruzioni di cui MicroPython ha bisogno per comunicare con Pico e altri dispositivi compatibili con MicroPython, estendendo il linguaggio per il physical computing. Senza questa riga non sarai in grado di controllare i pin GPIO di Pico né accendere il LED presente sulla scheda.

La libreria `machine` espone l'*interfaccia di programmazione dell'applicazione (API)*. Il nome sembra complicato, ma descrive esattamente ciò che fa: consente al tuo programma, o *applicazione*, di comunicare con Pico tramite un'*interfaccia*.

La riga successiva del programma fornisce un esempio dell'API della libreria **machine**:

```
led_onboard = machine.Pin("LED", machine.Pin.OUT)
```

Questa riga definisce un oggetto chiamato **led_onboard**, che offre un nome descrittivo che è possibile utilizzare per fare riferimento al LED integrato più avanti nel programma. È tecnicamente possibile utilizzare qualsiasi nome, tuttavia, per rendere il programma più facile da leggere e capire, è meglio attenersi a nomi che descrivono lo scopo della variabile.

La seconda parte della riga chiama la funzione **Pin** nella libreria della macchina. Questa funzione, come suggerisce il nome, è progettata per gestire i pin GPIO di Pico. In questa fase nessuno dei pin GPIO (incluso GP25, il pin collegato al LED di bordo) conosce la propria funzione. Il primo argomento, **LED**, è una *macro* speciale assegnata al LED di bordo, che puoi utilizzare invece di ricordare il numero del pin. Il secondo, **machine.Pin.OUT**, indica a Pico che il pin deve essere utilizzato come *uscita* piuttosto che come *ingresso*.

Quella riga è sufficiente per impostare il pin, ma non accenderà il LED. Per farlo, devi dire a Pico di attivare effettivamente il pin. Digita il codice seguente nella riga successiva:

```
led_onboard.value(1)
```

Anche questa riga utilizza l'API della libreria machine. La riga precedente ha creato l'oggetto **led_onboard** come uscita sul pin GP25 mediante la macro **LED**, questa riga prende l'oggetto e ne imposta il *valore* a 1 per "acceso". Potrebbe anche impostare il valore a 0, cioè "spento".

NUMERI PIN

Quando si parla dei pin GPIO su Pico, generalmente ci si riferisce ai nomi completi: ad esempio GP25 per il pin collegato al LED sulla scheda. In MicroPython tuttavia le lettere G e P vengono eliminate, quindi, se utilizzi il numero di pin, anziché la macro **LED**, assicurati di scrivere "25" invece di "GP25" nel tuo programma, altrimenti non funzionerà.



Fai clic sul pulsante **Esegui** e salva il programma su Pico come **Blink.py**. Vedrai il LED accendersi. Complimenti, hai scritto il tuo primo programma di physical computing!

Noterai però che il LED rimane acceso. Questo perché il programma dice a Pico di accenderlo, ma non di spegnerlo. Puoi aggiungere un'altra riga alla fine del programma:

```
led_onboard.value(0)
```

Tuttavia, questa volta quando esegui il programma il LED sembra non accendersi mai. Questo perché Pico funziona molto più rapidamente di quanto si possa vedere a occhio nudo. Il LED si accende, ma per così poco tempo che sembra rimanere sempre spento. Per risolvere questo problema, è necessario rallentare il programma introducendo un ritardo.

Torna all'inizio del programma: fai clic per spostare il cursore alla fine della prima riga e premi **INVIO** per inserire una seconda riga. Su questa riga, digita:

```
import time
```

Come per **import machine**, questa riga importa una nuova libreria in Micro-Python: **time**. Questa libreria gestisce tutto ciò che ha a che fare con il tempo, dalla misurazione all'inserimento di ritardi nei tuoi programmi.

Vai in fondo al programma e fai clic sulla fine della riga **led_onboard.value(1)**, quindi premi **INVIO** per inserire una nuova riga. Digita:

```
time.sleep(5)
```

Viene richiamata la funzione **sleep** dalla libreria **time**, che mette in pausa il programma per il numero di secondi digitato, in questo caso cinque secondi.



Fai clic sul pulsante **Esegui**. Questa volta vedrai il LED di Pico accendersi, rimanere acceso per cinque secondi (prova a contare) e spegnersi di nuovo.

Adesso è arrivato il momento di far lampeggiare il LED. Per farlo, dovrai creare un ciclo. Riscrivi il tuo programma in modo che corrisponda a quello qui sotto:

```
import machine  
import time
```

```
led_onboard = machine.Pin(LED, machine.Pin.OUT)
```

```
while True:  
    led_onboard.value(1)  
    time.sleep(5)  
    led_onboard.value(0)  
    time.sleep(5)
```

Ricorda che le linee all'interno del ciclo devono avere un rientro di quattro spazi, in questo modo MicroPython saprà che formano una ripetizione. Fai di nuovo clic sull'icona **Esegui**  e vedrai il LED accendersi per cinque secondi, spegnersi per cinque secondi e riaccendersi, ripetendo un ciclo infinito. Il LED continuerà a lampeggiare finché non si fa clic sull'icona **Ferma**  per annullare il programma e ripristinare Pico.

C'è anche un altro modo per gestire lo stesso lavoro: usare *toggle* anziché impostare l'uscita del LED su 0 o 1. Elimina le ultime quattro righe del tuo programma e sostituiscile così:

```
import machine
import time

led_onboard = machine.Pin(LED, machine.Pin.OUT)

while True:
    led_onboard.toggle()
    time.sleep(5)
```

Esegui di nuovo il programma. Vedrai la stessa attività di prima: il LED si accenderà per cinque secondi, poi si spegnerà per cinque secondi, quindi si accenderà di nuovo in un ciclo infinito. Questa volta, però, il tuo programma è più breve di due righe: l'hai *ottimizzato*. Disponibile su tutti i pin di uscita digitale, `toggle()` passa semplicemente da acceso a spento: se il pin è acceso, `toggle()` lo spegne, mentre se è spento, `toggle()` lo accende.

SFIDA: ILLUMINAZIONE PIÙ LUNGA

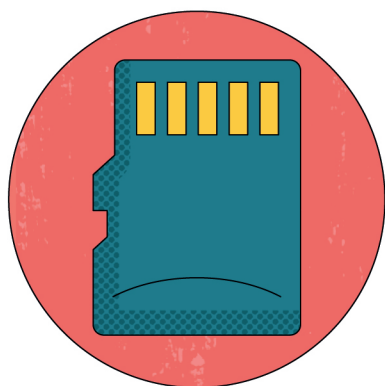
Come cambieresti il tuo programma per far sì che il LED rimanga acceso più a lungo? Oppure spento più a lungo? Qual è il ritardo più piccolo che puoi utilizzare pur riuscendo a vedere il LED accendersi e spegnersi?



Complimenti! Hai appreso cos'è un microcontrollore, come collegare Raspberry Pi Pico al tuo Raspberry Pi, come scrivere programmi MicroPython e come attivare un LED controllando un pin di Pico.

C'è molto altro da imparare su Raspberry Pi Pico, come: usarlo con una breadboard, collegare hardware aggiuntivo come LED, pulsanti, sensori di movimento o schermi, e persino utilizzare funzioni avanzate come i *convertitori analogico-digitali* (ADC) e le capacità di *ingresso/uscita programmabile* (PIO). Inoltre potresti anche collegarlo alla tua rete per iniziare a sperimentare l'*Internet delle cose* (IoT).

Per saperne di più, acquista una copia di *Get Started with MicroPython on Raspberry Pi Pico*. È disponibile in tutte le migliori librerie, online e in versione cartacea.



Appendice A

Come installare un sistema operativo su una scheda microSD

Puoi acquistare schede microSD con Raspberry Pi OS preinstallato presso tutti i migliori rivenditori di Raspberry Pi, in modo da poter iniziare a utilizzare Raspberry Pi in modo semplice e veloce. Le schede microSD precaricate vengono fornite anche con Raspberry Pi Desktop Kit e Raspberry Pi 400.

Puoi comunque installare facilmente tu stesso il sistema operativo su una scheda microSD vuota mediante Raspberry Pi Imager. Se utilizzi Raspberry Pi 4, Raspberry Pi 400 o Raspberry Pi 5, puoi anche scaricare e installare il sistema operativo tramite la rete direttamente sul dispositivo.

ATTENZIONE

Se hai acquistato una scheda microSD con Raspberry Pi OS già preinstallato, non dovrai fare altro che collegarla a Raspberry Pi. Queste istruzioni servono per installare Raspberry Pi OS su schede microSD vuote o su schede che hai già utilizzato e che vuoi riutilizzare. Se la scheda microSD dovesse già contenere altri file, questi verranno eliminati, quindi assicurati di aver fatto prima un backup.



Come Scaricare Raspberry Pi Imager

Basato su Debian, Raspberry Pi OS è il sistema operativo ufficiale di Raspberry Pi. Il modo più semplice per installare il sistema operativo Raspberry Pi su una scheda microSD è quello di utilizzare lo strumento Raspberry Pi Imager, scaricabile da rptl.io/imager.

L'applicazione Raspberry Pi Imager è disponibile per computer Windows, macOS e Ubuntu Linux, quindi scegli la versione più adatta al tuo sistema. Se l'unico computer a cui hai accesso è il tuo Raspberry Pi, passa alla sezione "Esecuzione di Raspberry Pi Imager tramite la rete" per vedere se è possibile eseguire l'applicazione direttamente su Raspberry Pi. In caso contrario, dovrai acquistare una scheda microSD con il sistema operativo già installato presso un rivenditore Raspberry Pi o chiedere a un amico di installarlo sulla scheda microSD per te.

Su macOS, fai doppio clic sul file **DMG** che hai scaricato. Per consentire l'esecuzione delle applicazioni scaricate da App Store e sviluppatori identificati, potresti dover modificare le impostazioni di sicurezza e privacy. Puoi quindi trascinare l'icona di **Raspberry Pi Imager** nella cartella Applicazioni.

Su un PC Windows, fai doppio clic sul file **EXE** che hai scaricato. Quando richiesto, seleziona il pulsante **Sì** per autorizzarne l'esecuzione. Quindi, fai clic su **Install** per avviare l'installazione.

Su Ubuntu Linux, fai doppio clic sul file **DEB** scaricato per aprire il Software Centre con il pacchetto selezionato, quindi segui le istruzioni indicate sullo schermo per installare Raspberry Pi Imager.

Ora puoi collegare la scheda microSD al computer. Avrai bisogno di un adattatore USB, a meno che il tuo computer non sia dotato di un lettore di schede integrato, che è presente in molti laptop, ma non in molti computer desktop. La scheda microSD non necessita di essere preformattata.

Esegui l'applicazione Raspberry Pi Imager, quindi passa a «Come scrivere il sistema operativo sulla scheda microSD» a pagina 244.

Come eseguire Raspberry Pi Imager tramite la rete

Raspberry Pi 4 e Raspberry Pi 400 possono eseguire autonomamente Raspberry Pi Imager, caricandolo tramite la rete, senza la necessità di utilizzare un computer desktop o portatile separato.

ATTENZIONE

Al momento della stesura di questa guida, l'installazione tramite rete non è supportata su Raspberry Pi 5, ma sarà disponibile con un futuro aggiornamento del firmware.



Per eseguire direttamente Raspberry Pi Imager, ti occorreranno: un Raspberry Pi, una scheda microSD vuota, una tastiera (se non usi la tastiera integrata del Raspberry Pi 400), una TV o un monitor e un cavo Ethernet collegato al modem o router. L'installazione tramite connessione Wi-Fi non è supportata.

Inserisci la scheda microSD vuota nello slot microSD del tuo Raspberry Pi e collega la tastiera, il cavo Ethernet e l'alimentatore USB. Se stai riutilizzando una vecchia scheda microSD, tieni premuto il tasto **Maiuscolo** sulla tastiera mentre il Raspberry Pi si avvia per caricare il programma di installazione tramite rete. Se la scheda microSD è vuota, il programma di installazione si caricherà automaticamente.

Quando viene visualizzata la schermata del programma di installazione tramite rete, tieni premuto il tasto **Maiuscolo** per avviare il processo di installazione. Il programma di installazione scaricherà automaticamente una versione speciale di Raspberry Pi Imager e la caricherà sul Raspberry Pi come mostrato nella **Figura A-1**. Una volta scaricata, vedrai una schermata esattamente come quella della versione autonoma di Raspberry Pi Imager, completa di opzioni per scegliere un sistema operativo e un dispositivo di archiviazione per l'installazione.

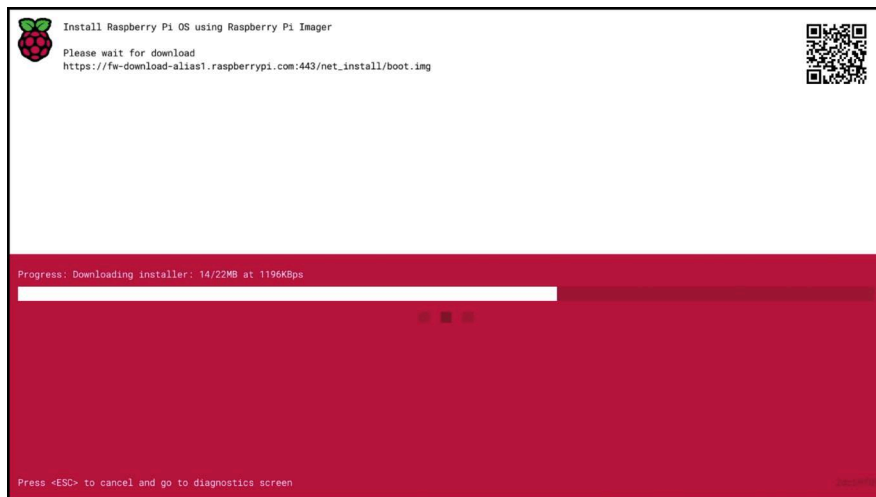


Figura A-1 Installazione del sistema operativo di Raspberry Pi attraverso la rete

Come scrivere il sistema operativo sulla scheda microSD

Fai clic sul pulsante **Scegli Dispositivo** per selezionare il modello di Raspberry Pi in tuo possesso e vedrai la schermata mostrata nella **Figura A-2**. Trova il tuo Raspberry Pi nell'elenco e fai clic su di esso. Quindi fai clic su **Scegli S.O.** per selezionare il sistema operativo che desideri installare e verrà visualizzata la schermata mostrata nella **Figura A-3**.

Ti consigliamo di selezionare il sistema operativo standard Raspberry Pi OS (con desktop) ma, se preferisci la versione più leggera (Lite) o quella completa (ossia Raspberry Pi OS con tutti i software consigliati preinstallati), seleziona **Raspberry Pi OS (other)**.

Puoi anche scorrere l'elenco per vedere una serie di sistemi operativi di terze parti compatibili con Raspberry Pi. A seconda del modello di Raspberry Pi che hai a disposizione, questi possono andare da sistemi operativi di uso generale come Ubuntu Linux e RISC OS Pi a sistemi operativi pensati appositamente per intrattenimento, giochi, emulazione, stampa 3D, segnaletica digitale e altro ancora.

In fondo all'elenco troverai l'opzione **Cancella**, che elimina tutti i dati presenti sulla scheda microSD.

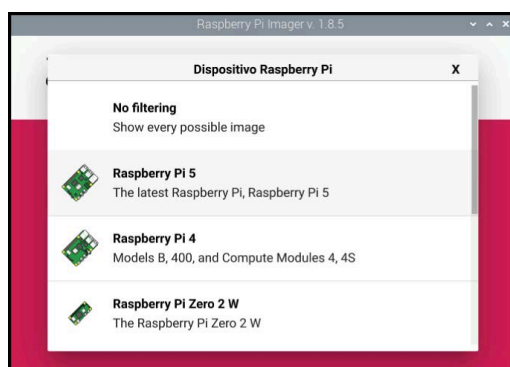


Figura A-2 Scelta del modello di Raspberry Pi



32-BIT O 64-BIT?

Dopo aver selezionato un modello di Raspberry Pi, ti verranno proposte solo le immagini compatibili con esso. Se tra le opzioni c'è Raspberry Pi OS (64 bit), come nel caso di Raspberry Pi 4 o Raspberry Pi 5, scegli l'opzione a 64 bit a meno che tu non debba installare una versione a 32 bit del sistema operativo per qualche motivo specifico.

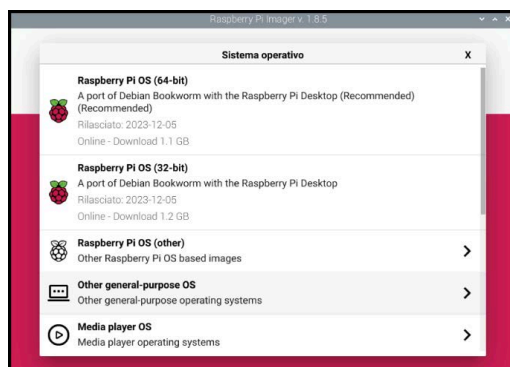


Figura A-3 Scelta di un sistema operativo

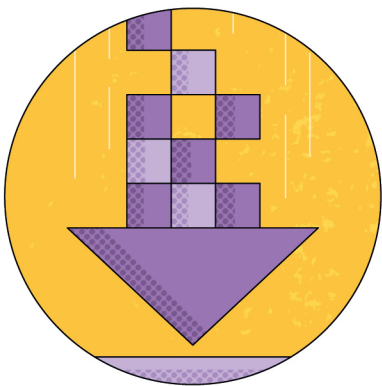
Se vuoi provare un sistema operativo non presente nell'elenco, puoi comunque installarlo utilizzando Raspberry Pi Imager. Dovrai solo andare sul sito Web del sistema operativo, scaricare l'immagine e scegliere l'opzione **Use custom** in fondo all'elenco Choose OS.

Una volta selezionato il sistema operativo, fai clic sul pulsante **Scegli Scheda SD** e seleziona la scheda microSD. Generalmente è l'unico dispositivo di archiviazione presente nell'elenco. Se vedi più di un dispositivo di archiviazione (questo può succedere se hai un'altra scheda microSD o una chiavetta USB collegata al computer) accertati di selezionare il dispositivo giusto, altrimenti potresti cancellare tutti i dati presenti sull'unità. In caso di dubbi, chiudi Raspberry Pi Imager, scollega tutte le unità rimovibili tranne la scheda microSD di destinazione e riapri Raspberry Pi Imager.

Infine, fai clic sul pulsante **Avanti** e ti verrà chiesto se vuoi personalizzare il sistema operativo. Se stai utilizzando la versione Lite, dovrai eseguire questo passaggio perché ti permette di configurare il nome utente, la password, la connessione alla rete wireless e altro ancora senza dover collegare tastiera, mouse e monitor.

Successivamente, Raspberry Pi Imager ti chiederà di confermare se vuoi sovrascrivere il contenuto della scheda SD. Facendo clic su **Sì**, verrà avviato il processo di scrittura. Attendi mentre l'applicazione scrive il sistema operativo selezionato sulla scheda e lo verifica. Quando il sistema operativo è stato scritto sulla scheda, puoi rimuovere la scheda microSD dal computer desktop o portatile e inserirla nel Raspberry Pi per avviare il nuovo sistema operativo. Se hai scritto il nuovo sistema operativo sul tuo Raspberry Pi utilizzando la funzione di avvio da rete, spegni e riaccendi il Raspberry Pi per caricare il nuovo sistema operativo.

Assicurati sempre che il processo di scrittura sia terminato prima di rimuovere la scheda microSD o spegnere il Raspberry Pi. Se il processo si interrompe quando non è ancora stato completato, il nuovo sistema operativo non funzionerà correttamente. Se ciò accade, è sufficiente avviare nuovamente il processo di scrittura per sovrascrivere il sistema operativo danneggiato e sostituirlo con una copia funzionante.



Appendice B

Come installare e disinstallare il software

Raspberry Pi OS viene fornito con una selezione di rinomati pacchetti software accuratamente scelti dal team di Raspberry Pi, ma questi non sono gli unici pacchetti funzionanti su un Raspberry Pi. Attenendoti alle seguenti istruzioni, potrai cercare altri tipi di software e installarli e disinstallarli a piacere, ampliando le capacità del tuo Raspberry Pi.

Le istruzioni contenute in questa appendice sono complementari a quelle indicate nel Capitolo 3, *Come utilizzare Raspberry Pi*, che spiega come utilizzare lo strumento **Software consigliato**.

Come cercare i software disponibili

Per visualizzare e cercare i pacchetti software disponibili per Raspberry Pi OS utilizzando i cosiddetti *repository di software*, fai clic sull'icona di Raspberry Pi per caricare il menu, seleziona la categoria Preferenze, quindi fai clic su **Aggiungi / Rimuovi software**. Dopo alcuni secondi, viene visualizzata la finestra dello strumento, come mostrato nella **Figura B-1**.

La parte sinistra della finestra **Aggiungi / Rimuovi software** contiene un elenco di categorie. Sono le stesse categorie presenti nel menu principale quando si fa clic sull'icona Raspberry Pi.

Facendo clic su una di esse, potrai visualizzare l'elenco dei relativi software disponibili. Nella casella in alto a sinistra della finestra, puoi anche inserire un termine di ricerca, come "editor di testo" o "gioco", e visualizzare l'elenco dei pacchetti software corrispondenti per qualsiasi categoria. Facendo clic su

uno qualsiasi dei pacchetti, verranno visualizzate ulteriori informazioni a riguardo nello spazio in basso nella finestra come mostrato nella **Figura B-2**.

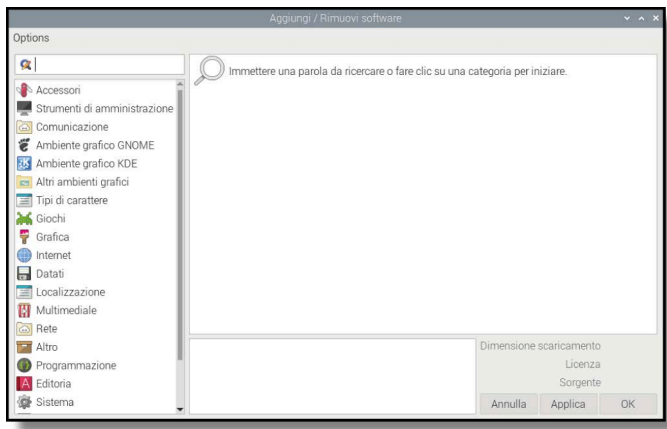


Figura B-1 Finestra **Aggiungi / Rimuovi software**

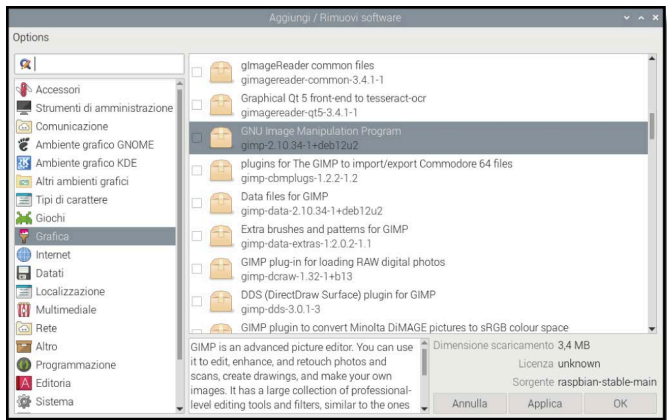


Figura B-2 Informazioni aggiuntive sul pacchetto

Se per la categoria scelta sono disponibili molti pacchetti software, potrebbe essere necessario un po' di tempo prima che lo strumento di aggiunta o rimozione del software finisca di analizzare l'elenco.

Come installare il software

Per selezionare un pacchetto per l'installazione, spunta la casella accanto a esso. Puoi installare più di un pacchetto alla volta semplicemente selezionando quelli che ti interessano. L'icona accanto al pacchetto diventerà una scatola aperta con il simbolo "+", come mostrato nella **Figura B-3**, per confermare che verrà installato.

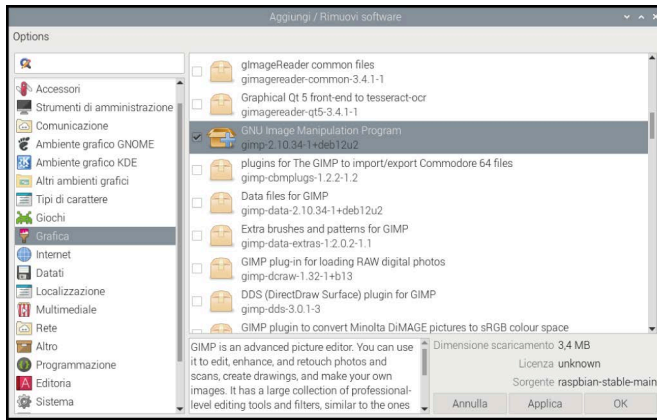


Figura B-3 Selezione di un pacchetto per l'installazione

Quando hai scelto, fai clic sul pulsante **OK** o **Applica**. L'unica differenza è che se selezioni **OK** lo strumento di aggiunta o rimozione del software verrà chiuso una volta completata l'installazione, mentre se selezioni **Applica** rimarrà aperto. Per evitare che il software venga aggiunto o rimosso da altri utenti, ti verrà chiesto di inserire la password personale (**Figura B-4**) per confermare la tua identità.

Quando si installa un singolo pacchetto, potrebbero essere installati anche altri pacchetti insieme ad esso. Si tratta di *dipendenze*, ossia pacchetti che servono al software scelto per funzionare, come bundle di effetti sonori per un gioco o database da collegare a un server Web.

Una volta installato il software, dovresti trovarlo facendo clic sull'icona di Raspberry Pi per caricare il menu e trovare la categoria del pacchetto software (vedi **Figura B-5**). Tieni presente che la categoria del menu non sempre corrisponde alla categoria dello strumento di aggiunta/rimozione software e che alcuni software non hanno alcuna voce nel menu. Questo software è noto come *software a riga di comando* e deve essere eseguito dal terminale. Per ulteriori informazioni sulla riga di comando e sul terminale, consulta il *Appendice C, Interfaccia a riga di comando*.

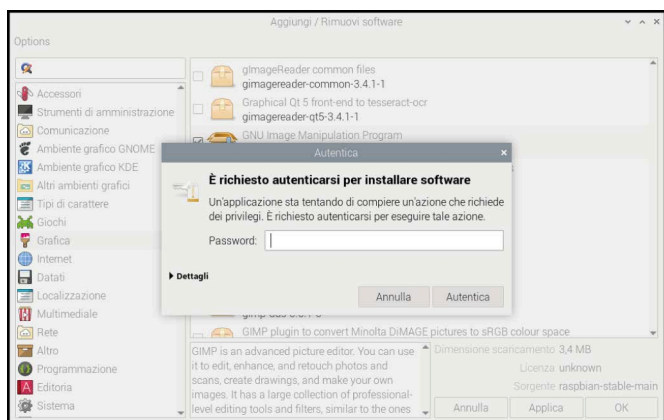


Figura B-4 Conferma della propria identità

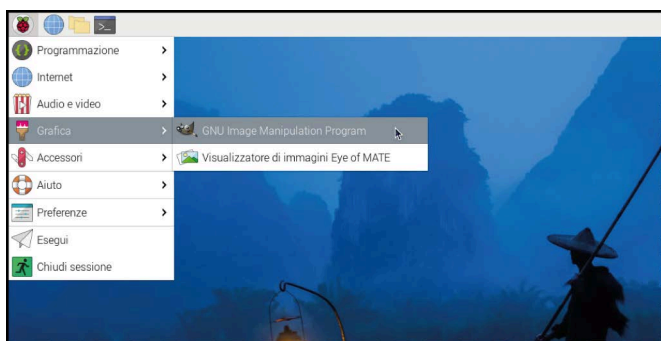


Figura B-5 Trovare il software appena installato

Come disinstallare il software

Per selezionare un pacchetto da rimuovere o *disinstallare*, cercalo nell'elenco dei pacchetti (in questo caso la funzione di ricerca torna utile) e toglì la spunta dalla casella corrispondente al pacchetto che vuoi rimuovere. Puoi disinstallare più di un pacchetto alla volta semplicemente deselegzionando quelli che vuoi rimuovere. L'icona accanto al pacchetto diventerà una scatola aperta vicino a un cestino, per confermare che verrà disinstallato (vedi **Figura B-6**).

Come prima, puoi scegliere se fare clic su **OK** o su **Applica** per iniziare a disinstallare i pacchetti software selezionati. Ti verrà chiesto di confermare la tua password, a meno che tu non lo abbia fatto negli ultimi minuti, e potresti

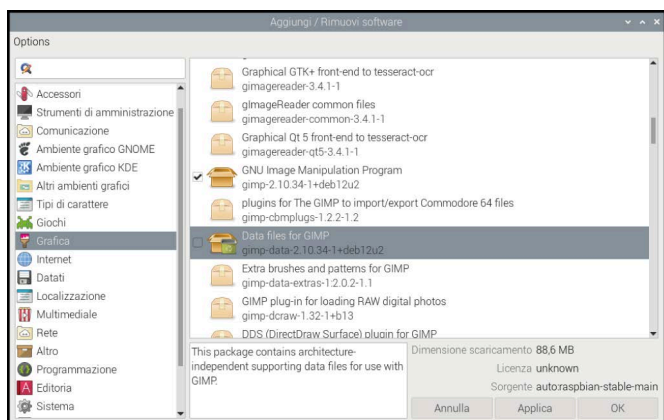


Figura B-6 Selezione di un pacchetto da rimuovere

anche ricevere una richiesta di conferma per rimuovere eventuali dipendenze relative al tuo pacchetto software (vedi **Figura B-7**). Quando la disinstallazione sarà terminata, il software scomparirà dal menu di Raspberry Pi, ma i file creati con il software, come le immagini per un pacchetto grafico o i salvataggi di un gioco, non verranno rimossi.

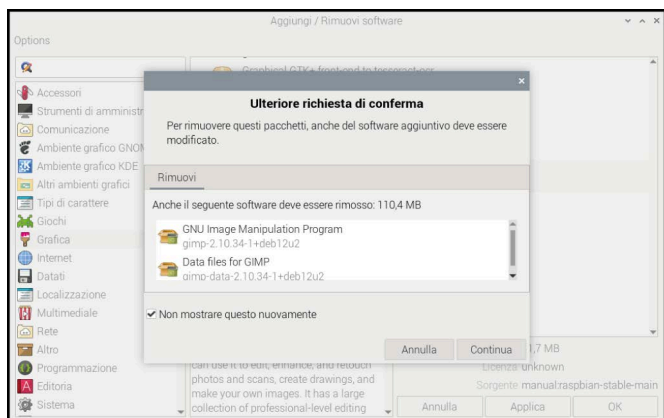
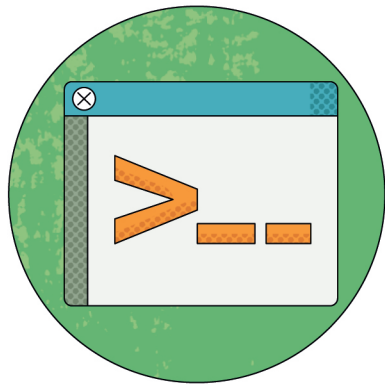


Figura B-7 Conferma della rimozione delle dipendenze



ATTENZIONE

Tutti i software installati nel sistema operativo Raspberry Pi OS vengono visualizzati nello strumento di aggiunta o rimozione del software, incluso quello necessario per l'esecuzione di Raspberry Pi. È possibile rimuovere i pacchetti che il desktop deve caricare, ma ti consigliamo di non disinstallare nulla se non sei sicuro di quelli che potrebbero ancora servirti. È possibile reinstallare Raspberry Pi OS seguendo le istruzioni riportate nel Appendice A, *Come installare un sistema operativo su una scheda microSD*.



Appendice C

Interfaccia a riga di comando

La maggior parte dei software su Raspberry Pi è gestibile dal desktop, ma alcuni sono accessibili solo utilizzando una modalità testuale nota come *interfaccia a riga di comando (CLI)* in un'applicazione denominata LXTerminal.

La maggior parte degli utenti non avrà mai bisogno di utilizzare la CLI, ma per coloro che vogliono saperne di più, questa appendice offre una piccola introduzione.

Come caricare l'LXTerminal

La CLI è accessibile tramite LXTerminal, un pacchetto software che carica un componente tecnicamente noto come *telescrivente virtuale (VTY)*, nome che risale ai primi tempi dei computer, quando gli utenti lanciavano i comandi tramite una grande macchina da scrivere elettromeccanica invece che da tastiera e monitor.

Per caricare il pacchetto LXTerminal, fai clic sull'icona a Raspberry Pi per aprire il menu, scegli la categoria **Accessori**, quindi fai clic su **LXTerminal**. Visualizzerai la finestra LXTerminal, come mostrato nella **Figura C-1**.

Puoi trascinare la finestra di LXTerminal sul desktop, ridimensionarla, ingrandirla o ridurla a icona come qualsiasi altra finestra. Puoi ingrandire la dimensione dei caratteri per leggere meglio, oppure ridurla per visualizzare più testo nella finestra: fai clic sul menu **Modifica** e scegli rispettivamente **Aumenta Zoom** o **Diminuisci Zoom**, oppure tieni premuto il tasto **CTRL+SHIFT** sulla tastiera insieme al **+** o al **-**.

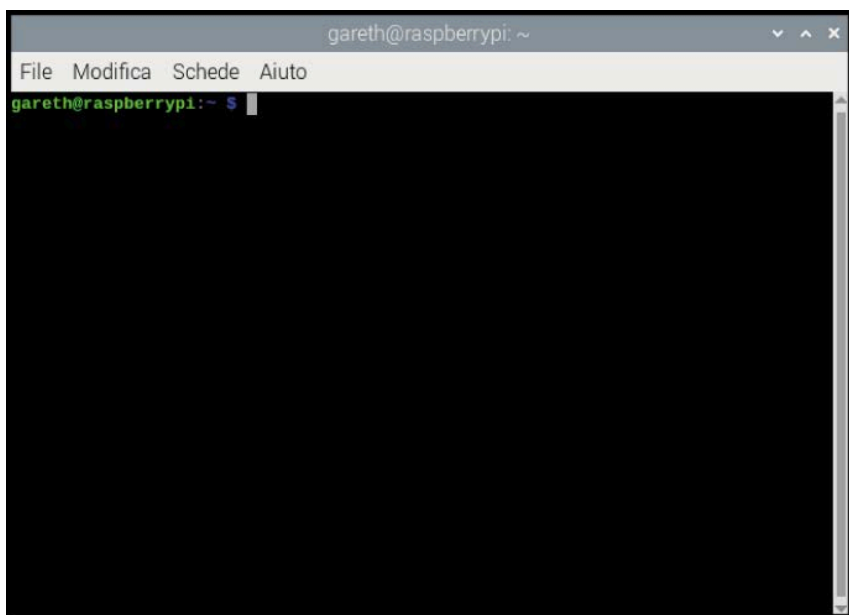


Figura C-1 La finestra LXTerminal

Il prompt

La prima cosa che viene visualizzata nel terminal è il *prompt* in attesa che tu inserisca le istruzioni. Il prompt su un Raspberry Pi che esegue Raspberry Pi OS si presenta così:

username@raspberrypi:~ \$

- ▶ La prima parte del prompt, **username**, è costituita dal nome utente.
- ▶ La seconda parte, dopo **@**, è il nome host del computer che stai utilizzando, che per impostazione predefinita è **raspberrypi**.
- ▶ Dopo **:"** c'è una tilde, **~**, un'abbreviazione che si riferisce alla home directory e rappresenta la tua *directory di lavoro corrente* (CWD).
- ▶ Infine, il simbolo **\$** indica che il tuo utente è un *utente senza privilegi*, il che significa che dovrai elevare i tuoi permessi prima di eseguire alcune operazioni come l'aggiunta o la rimozione di software.

Come muoversi

Prova a digitare quanto segue, quindi premi il tasto **INVIO**:

```
cd Desktop
```

Vedrai il prompt diventare:

```
pi@raspberrypi:~/Desktop $
```

Come vedi, la directory di lavoro corrente è cambiata: prima eri nella home directory, indicata con la `~`, mentre adesso sei nella sottodirectory **Desktop** della directory principale. Per farlo, hai usato il comando `cd` - *cambia directory*.

DISTINZIONE TRA MAIUSCOLE E MINUSCOLE

L'interfaccia a riga di comando del sistema operativo Raspberry Pi OS necessita che per i comandi o i nomi venga rispettata la distinzione tra caratteri maiuscoli e minuscoli. Se quando provi a cambiare directory visualizzi il messaggio "no such file or directory", verifica che la D di **Desktop** sia maiuscola.



Ci sono quattro modi per tornare alla home directory: provali tutti, tornando ogni volta nella sottodirectory **Desktop**. Il primo modo è:

```
cd ..
```

Il simbolo `..` è un'altra scorciatoia che porta alla "directory sopra questa", nota anche come *directory padre*. Poiché la directory sopra **Desktop** è la tua home directory, questa scorciatoia ti riporta lì.

Torna alla sottodirectory **Desktop** e prova il secondo metodo:

```
cd ~
```

In questo caso, viene utilizzato il simbolo `~`, che significa letteralmente "cambia nella mia home directory". A differenza di `cd ..`, che ti porta alla directory padre di qualsiasi directory in cui ti trovi, questo comando funziona da qualsiasi posizione.

C'è però un modo ancora più semplice:

```
cd
```


Senza il nome di una directory, **cd** punta direttamente alla home directory per impostazione predefinita.

C'è un altro modo per tornare alla tua home directory (sostituisci **username** con il tuo nome utente):

```
cd /home/username
```

Questo comando utilizza quello che viene chiamato *percorso assoluto*, che funzionerà indipendentemente dalla directory di lavoro corrente. Quindi, come **cd** da solo o **cd ~**, questo ti riporterà alla tua home directory indipendentemente da dove ti trovi. A differenza degli altri metodi, però, è necessario che tu conosca il tuo nome utente.

Come gestire i file

Per esercitarti a lavorare con i file, vai (**cd**) alla directory **Desktop** e digita quanto segue:

```
touch Test
```

Sul desktop verrà visualizzato un file chiamato **Test**. Il comando **touch** viene normalmente utilizzato per aggiornare le informazioni di data e ora su un file, ma se, come in questo caso, il file non esiste, lo crea.

Prova a digitare:

```
cp Test Test2
```

Sul desktop verrà visualizzato un altro file chiamato **Test2**, che è una *copia* identica del file originale. Eliminalo digitando:

```
rm Test2
```

Questo comando *rimuove* il file.

ATTENZIONE

Quando elimini dei file utilizzando il File Manager grafico, questo li sposta nel Cestino per consentirti di recuperarli in seguito, qualora fosse necessario. I file eliminati con **rm** sono invece eliminati per sempre e non vengono spostati nel cestino. Assicurati di scrivere i comandi correttamente!



Ora prova:

mv Test Test2

Questo comando *sposta* il file e consente di sostituire il file **Test** originale, che scomparirà, con il file **Test2**. Il comando di spostamento, **mv**, può essere utilizzato per rinominare i file.

Quando non ti trovi sul desktop, però, dovrai comunque essere in grado di visualizzare i file contenuti in una directory. Digita:

ls

Questo comando *elenca* il contenuto della directory corrente, o di qualsiasi altra directory indicata. Per ulteriori dettagli, tra cui l'elenco di tutti i file nascosti e le dimensioni dei file, prova ad aggiungere altri commutatori:

ls -larth

Questi parametri controllano il comando **ls**:

- ▶ **l** trasforma il suo output in un lungo elenco verticale
- ▶ **a** indica di mostrare tutti i file e directory, anche quelli nascosti.
- ▶ L'opzione **r** inverte il normale ordinamento
- ▶ **t** ordina per ora di modifica e, insieme a **r**, consente di visualizzare i file più vecchi all'inizio della lista e i più nuovi alla fine.
- ▶ **h** utilizza dimensioni dei file più leggibili, rendendo l'elenco più facile da capire.

Come eseguire i programmi

Alcuni programmi possono essere eseguiti solo da riga di comando, mentre altri hanno interfacce sia grafiche che a riga di comando come, ad esempio, Raspberry Pi Software Configuration Tool, che normalmente si carica dal menu con l'icona a lampone.

Per sperimentare l'uso di Software Configuration Tool da riga di comando, digita:

raspi-config

Ti verrà mostrato un messaggio di errore che ti dirà che il software può essere eseguito solo come *root*, ossia l'account superuser di Raspberry Pi, poiché il tuo account utente non dispone dei privilegi necessari. Ti indicherà anche come eseguire il software come root, digitando:

sudo raspi-config

La parte **sudo** del comando significa *cambia utente* e dice a Raspberry Pi OS di eseguire il comando come utente root. Il Software Configuration tool di Raspberry Pi apparirà come mostrato nella **Figura C-2**.

Dovrai usare **sudo** solo quando un programma ha bisogno di *privilegi* avanzati, ad esempio quando si tratta di installare o disinstallare un software o di settare le impostazioni di sistema. Un gioco, per esempio, non dovrebbe mai essere eseguito utilizzando **sudo**.

Premi due volte il tasto **TAB** per selezionare Finish e **INVIO** per uscire da Raspberry Pi Software Configuration Tool e tornare all'interfaccia a riga di comando. Infine, digita:

exit

Questo comando termina la sessione dell'interfaccia a riga di comando e chiude l'applicazione LXTerminal.

Come utilizzare i terminali TTY

L'applicazione LXTerminal non è l'unico modo per utilizzare l'interfaccia a riga di comando: è possibile anche passare a uno dei numerosi terminali già in funzione, noti come *telescriventi* o *TTY*. Tieni premuti i tasti **CTRL** e **ALT** sulla tastiera mentre premi **F2** per passare a "tty2" (vedi **Figura C-3**).

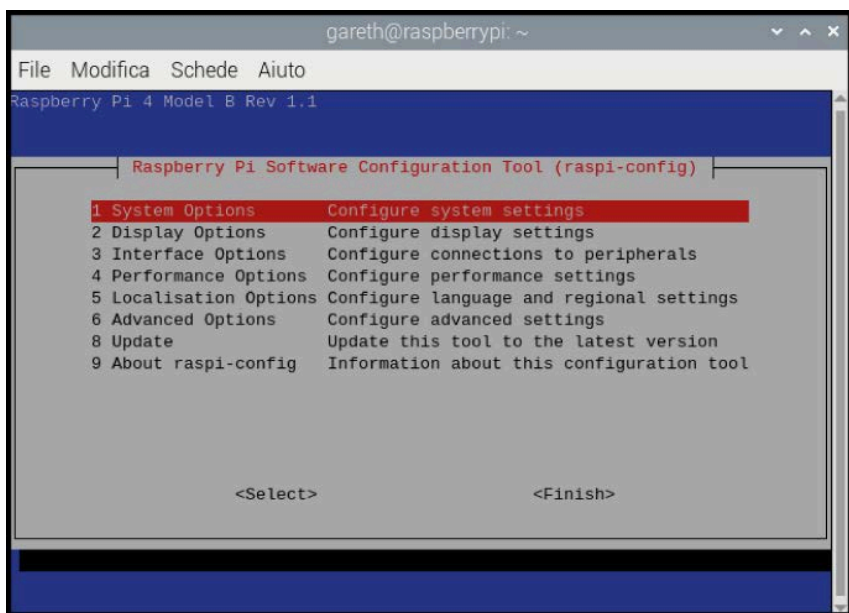


Figura C-2 Il Software Configuration tool di Raspberry Pi

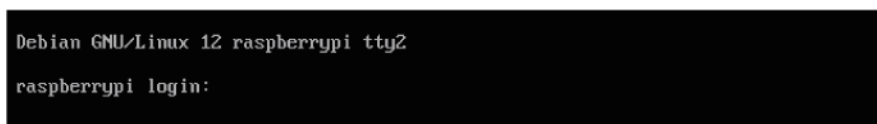


Figura C-3 Uno dei TTY

Dovrai effettuare nuovamente l'accesso con nome utente e password, dopodiché potrai utilizzare l'interfaccia a riga di comando proprio come nell'applicazione LXTerminal. L'utilizzo di questi TTY è utile quando, per qualsiasi motivo, l'interfaccia principale del desktop non funziona.

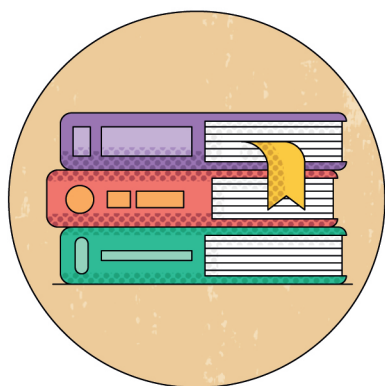
Per uscire dai TTY, tieni premuto **CTRL+ALT**, quindi premi **F7** e il desktop verrà nuovamente visualizzato. Premi nuovamente **CTRL+ALT+F2** per tornare a "tty2" e ritrovare tutto ciò che era in esecuzione.

Prima di cambiare di nuovo, digita:

exit

Quindi premi **CTRL+ALT+F7** per tornare al desktop. Prima di uscire dai TTY è importante eseguire il comando di uscita poiché chiunque abbia accesso alla tastiera potrebbe passare a un TTY senza dovere inserire la password.

Complimenti, hai mosso i tuoi primi passi nell'interfaccia a riga di comando del sistema operativo Raspberry Pi OS.



Appendice D

Ulteriori letture

La *Guida introduttiva ufficiale Raspberry Pi* ti aiuta a muovere i primi passi con il tuo Raspberry Pi ma non approfondisce tutti gli aspetti e le funzionalità a tua disposizione. Raspberry Pi vanta una community vasta, composta da persone che lo utilizzano per tutte le attività possibili, dai giochi alle applicazioni di rilevamento, dalla robotica all'intelligenza artificiale. Troverai numerose fonti di ispirazione.

In questa appendice troverai alcuni spunti per elaborare idee progettuali, piani di lezione e altro materiale per ampliare le conoscenze che hai acquisito con la *Guida introduttiva*.

Bookshelf

icona Raspberry Pi > Aiuto > Biblioteca

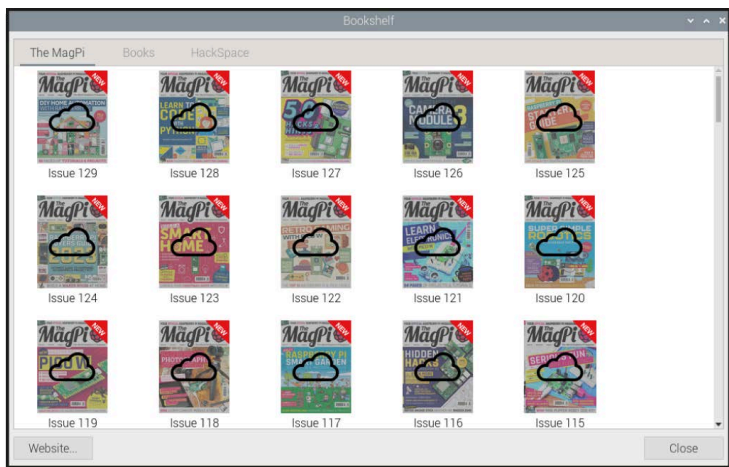


Figura D-1 L'applicazione Bookshelf

Bookshelf (mostrata nella **Figura D-1**) è un'applicazione inclusa in Raspberry Pi OS che ti permette di cercare, scaricare e leggere le versioni digitali delle pubblicazioni di Raspberry Pi Press. Per aprirla, fai clic sull'icona di Raspberry, seleziona Help e fai clic su **Biblioteca**, potrai cercare tra numerose riviste e libri, che potrai scaricare gratuitamente e leggere comodamente.

Novità su Raspberry Pi

raspberrypi.com/news

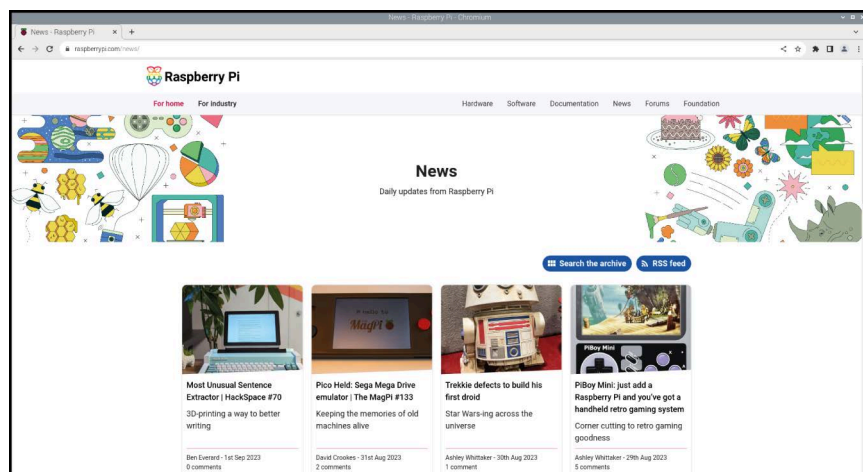


Figura D-2 Novità su Raspberry Pi

Ogni giorno infrasettimanale troverai un nuovo articolo, con annunci su nuovi computer e accessori Raspberry Pi, gli ultimi aggiornamenti software e i progetti della community, oltre agli aggiornamenti riguardo alle pubblicazioni di Raspberry Pi Press, tra cui The MagPi e HackSpace Magazine (**Figura D-2**).

Raspberry Pi Projects

rpf.io/projects

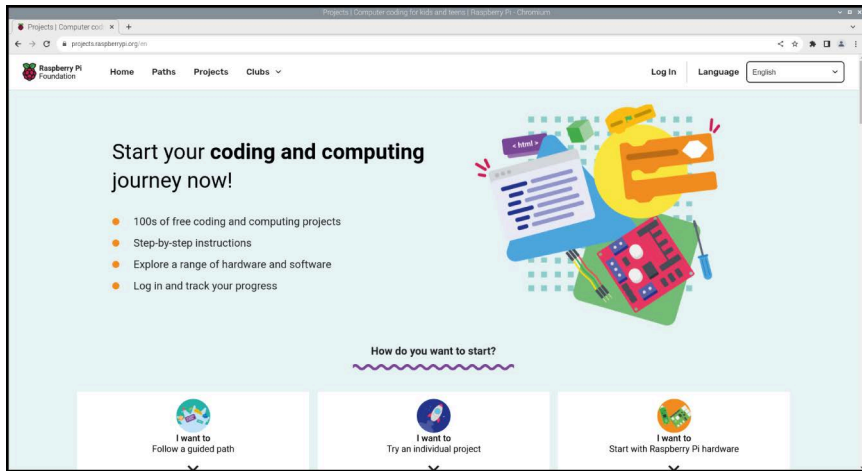


Figura D-3 Raspberry Pi Projects

Il sito ufficiale Raspberry Pi Projects della Raspberry Pi Foundation (« **Raspberry Pi Projects**» a **pagina 270**) offre tutorial dettagliati per la realizzazione di diverse tipologie di progetti, dalla creazione di giochi e musica allo sviluppo del proprio sito Web o alla creazione di robot con scheda Raspberry Pi. La maggior parte dei progetti è disponibile in diverse lingue e livelli di difficoltà adatti a tutti, dai principianti assoluti agli utenti più esperti.

Raspberry Pi Education

rpf.io/education

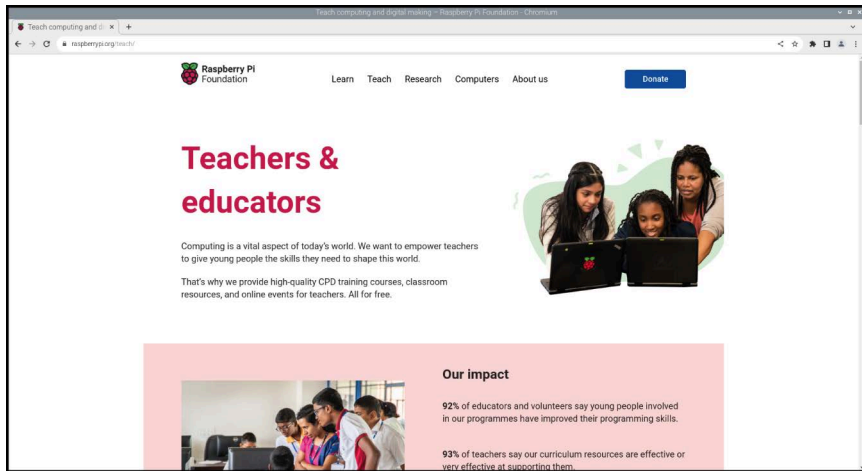


Figura D-4 Il sito di Raspberry Pi Education

Il sito ufficiale di Raspberry Pi Education (**Figura D-4**) offre newsletter, didattica online e progetti a tutti coloro che si occupano di formazione. Il sito è collegato anche ad altre risorse, tra cui i programmi di formazione gratuiti, i programmi Code Club e CoderDojo dedicati alla programmazione e gestiti da volontari e tanto altro ancora.

I forum di Raspberry Pi

rptl.io/forums

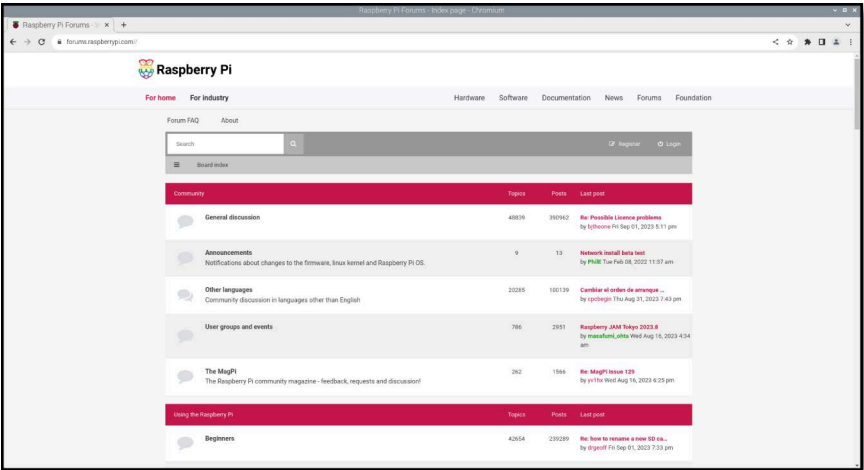


Figura D-5 I forum di Raspberry Pi

Nei forum di Raspberry Pi, mostrati nella **Figura D-5**, gli appassionati di questo prodotto possono incontrarsi e parlare di tutto, dai problemi dei principianti alle questioni più tecniche, e c'è anche un'area "off-topic" per discutere di argomenti più generici.

Il magazine The MagPi

magpi.cc

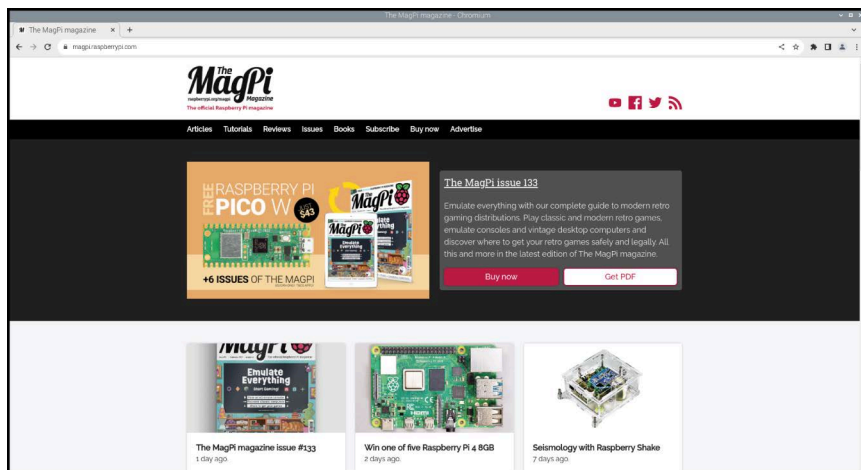


Figura D-6 Il magazine The MagPi

The MagPi, la rivista ufficiale di Raspberry Pi supportata in gran parte dalla community internazionale di Raspberry Pi (**Figura D-6**), è un mensile che contiene tutto ciò di cui hai bisogno, dai tutorial e guide alle recensioni, fino alle ultime novità.

Le copie sono disponibili in tutte le migliori edicole e supermercati e possono anche essere scaricate gratuitamente in formato digitale con licenza Creative Commons. *The MagPi* pubblica anche libri e riviste su diversi argomenti, che possono essere acquistati in formato cartaceo o scaricati gratuitamente.

Il magazine HackSpace

hsmag.cc

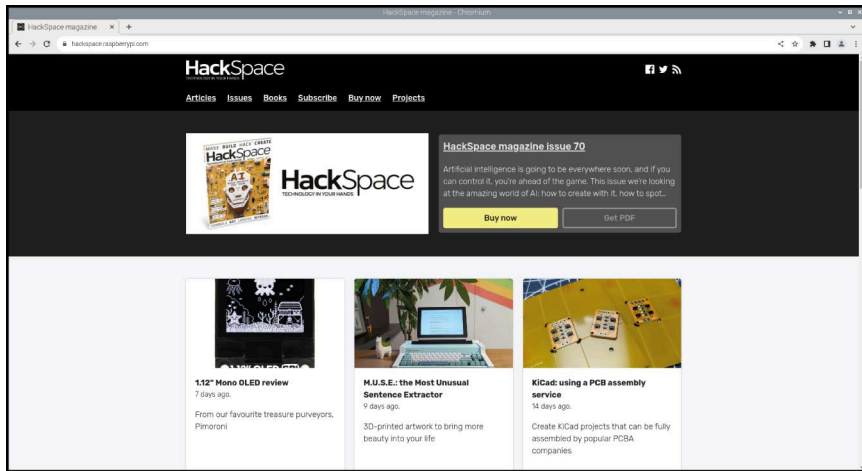
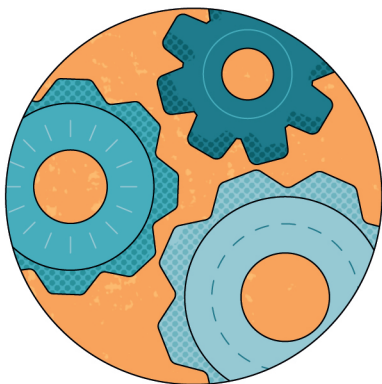


Figura D-7 Il magazine HackSpace

Il magazine *HackSpace* è incentrato sulla community degli sviluppatori, con recensioni di hardware e software, tutorial e interviste (**Figura D-7**). Se sei interessato a espandere i tuoi orizzonti oltre Raspberry Pi, *HackSpace* è un ottimo punto di partenza. Lo puoi trovare in edicola e nei supermercati in versione cartacea o scaricarlo gratuitamente in formato digitale.



Appendice E

Strumento Configurazione di Raspberry Pi

Lo strumento Configurazione di Raspberry Pi è un potente pacchetto che ti consente di regolare tutta una serie di impostazioni sul tuo Raspberry Pi, dalle interfacce ai programmi fino alle modalità per controllarlo in rete. Poiché all'inizio potresti demoralizzarti un po', questa appendice ti illustrerà nel dettaglio ciascuna impostazione, spiegandone lo scopo.

ATTENZIONE

A meno che tu non abbia bisogno di modificare una specifica impostazione, è meglio non utilizzare lo strumento Configurazione di Raspberry Pi. Se stai aggiungendo un nuovo componente hardware al tuo Raspberry Pi, ad esempio un HAT audio, segui le istruzioni per sapere quale impostazione modificare. In caso contrario, non modificare le impostazioni predefinite.



Puoi avviare lo strumento Configurazione di Raspberry Pi dal menu Raspberry Pi, sotto la categoria **Preferenze**. Oppure puoi eseguirlo dall'interfaccia a riga di comando o dall'**LXTerminal** usando il comando **raspi-config**. I layout della versione a riga di comando e della versione grafica differiscono tra loro, con opzioni che vengono visualizzate in categorie diverse a seconda della versione utilizzata. Questa appendice si riferisce alla versione grafica.

Scheda Sistema

La scheda Sistema (**Figura E-1**) contiene le opzioni che controllano le varie impostazioni del sistema operativo Raspberry Pi OS.

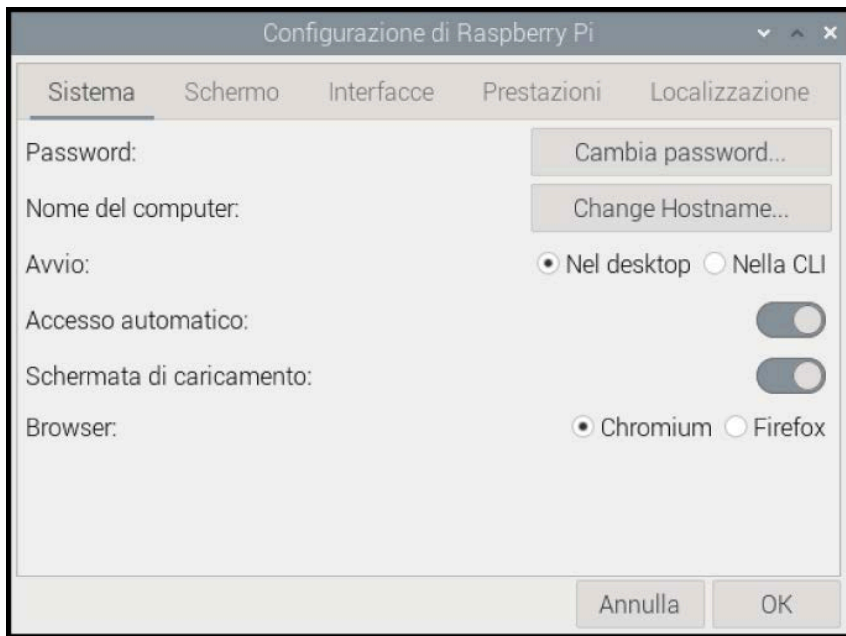


Figura E-1 La scheda Sistema

- ▶ **Password:** fai clic sul pulsante "**Cambia password**" per impostare una nuova password per l'account utente in uso.
- ▶ **Nome del computer:** il nome con cui un Raspberry Pi si identifica sulle reti. Se hai più di un Raspberry Pi sulla stessa rete, ogni dispositivo deve avere un nome univoco. Fai clic sul pulsante **Change Hostname** per sceglierne uno nuovo.
- ▶ **Avvio:** impostando l'avvio su "**Nel desktop**" (impostazione predefinita) verrà caricato il desktop di Raspberry Pi OS che già conosci. Se invece lo imposti su "**Nella CLI**" si caricherà l'interfaccia a riga di comando come descritto nel *Appendice C, Interfaccia a riga di comando*.
- ▶ **Accesso automatico:** quando attivo (impostazione predefinita), Raspberry Pi OS carica il desktop senza bisogno di inserire nome utente e password.

- ▶ **Schermata di caricamento:** quando attiva (impostazione predefinita), i messaggi di avvio di Raspberry Pi OS saranno nascosti dietro una schermata grafica di avvio.
- ▶ **Browser:** ti permette di passare da Chromium di Google (predefinito) a Firefox di Mozilla come browser Web predefinito.

Scheda Schermo

La scheda Schermo (**Figura E-2**) contiene le impostazioni che controllano la visualizzazione dello schermo.



Figura E-2 Scheda Schermo

- ▶ **Spegnimento schermo:** questa opzione ti permette di attivare e disattivare lo spegnimento dello schermo. Se abilitata, Raspberry Pi farà diventare nero il display se non lo usi per alcuni minuti al fine di proteggerlo da eventuali danni che potrebbero essere provocati dal mostrare un'immagine statica per periodi di tempo prolungati.
- ▶ **Risoluzione senza monitor:** questa opzione controlla la risoluzione del desktop virtuale quando utilizzi Raspberry Pi senza un monitor o una TV collegati, un'operazione nota come *headless*.

Scheda Interfacce

La scheda Interfacce (**Figura E-3**) contiene le impostazioni che controllano le interfacce hardware di Raspberry Pi.



Figura E-3 Scheda Interfacce

- ▶ **SSH:** abilita o disabilita l'interfaccia Secure Shell (SSH). Consente di aprire un'interfaccia a riga di comando su Raspberry Pi da un altro computer della rete utilizzando un client SSH.
- ▶ **VNC:** abilita o disabilita l'interfaccia Virtual Network Computing (VNC). Consente di visualizzare il desktop di Raspberry Pi da un altro computer della rete utilizzando un client VNC.
- ▶ **SPI:** abilita o disabilita l'interfaccia seriale periferica (SPI), utilizzata per controllare alcuni componenti che si collegano ai pin GPIO di Raspberry Pi.
- ▶ **I2C:** abilita o disabilita l'interfaccia Inter-Integrated Circuit (I²C), utilizzata per controllare alcuni componenti hardware aggiuntivi che si collegano ai pin GPIO.
- ▶ **Porta seriale:** abilita o disabilita la porta seriale di Raspberry Pi, disponibile sui pin GPIO.

- ▶ **Console seriale:** abilita o disabilita la console seriale, un'interfaccia a riga di comando disponibile sulla porta seriale. Questa opzione è disponibile solo se l'impostazione della porta seriale sopra citata è impostata su "Abilita".
- ▶ **1-Wire:** abilita o disabilita l'interfaccia 1-Wire, utilizzata per controllare alcuni componenti hardware aggiuntivi che si collegano ai pin GPIO di Raspberry Pi.
- ▶ **GPIO remoti:** abilita o disabilita un servizio di rete che consente di controllare i pin GPIO di Raspberry Pi da un altro computer in rete utilizzando la libreria GPIO Zero. Maggiori informazioni sui GPIO remoti sono consultabili su gpiozero.readthedocs.io.

Scheda Prestazioni

La scheda Prestazioni (**Figura E-4**) mostra le impostazioni che controllano le prestazioni del tuo Raspberry Pi.

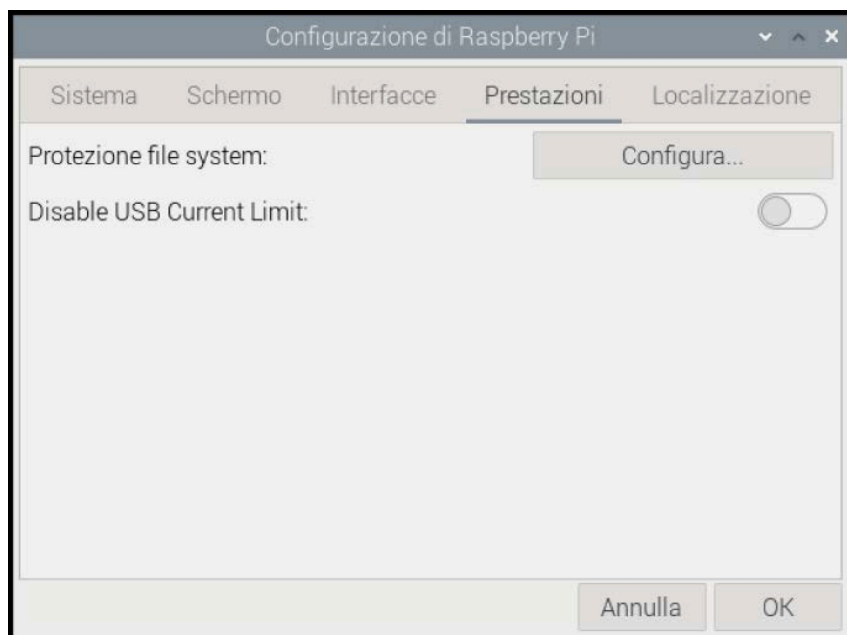


Figura E-4 Scheda Prestazioni

- ▶ **Protezione file system:** permette di bloccare il file system di Raspberry Pi affinché le modifiche vengano effettuate solo su un disco virtuale

mantenuto in memoria piuttosto che essere scritte sulla scheda microSD, in modo che sia possibile riportare il sistema a uno stato pulito a ogni riavvio, senza modifiche.

Anche i modelli di Raspberry Pi precedenti a Raspberry Pi 5 dispongono delle seguenti opzioni:

- ▶ **Ventola:** permette di attivare o disattivare una ventola di raffreddamento opzionale collegata al terminale GPIO di Raspberry Pi, progettata per mantenere il processore fresco in ambienti più caldi o sotto carico estremo. Una ventola compatibile con il case ufficiale di Raspberry Pi 4 è disponibile su rptl.io/casefan.
- ▶ **GPIO della ventola:** la ventola di raffreddamento è normalmente collegata al pin 14 del GPIO. Se a questo pin è collegato qualcos'altro, puoi scegliere un altro pin GPIO.
- ▶ **Temperatura della ventola:** la temperatura minima, in gradi Celsius, alla quale la ventola deve girare. Finché il processore di Raspberry Pi non raggiunge questa temperatura, la ventola rimarrà spenta per minimizzare il rumore.

Scheda Localizzazione

La scheda Localizzazione (**Figura E-5**) contiene le impostazioni che controllano per quale area geografica è stato progettato il tuo Raspberry Pi, comprese le impostazioni di layout della tastiera.



Figura E-5 Scheda Localizzazione

- ▶ **Lingua:** un'impostazione di sistema che consente di scegliere la lingua, il Paese e il set di caratteri. Cambiando la lingua da qui, cambierà solo la lingua visualizzata nelle applicazioni per le quali è disponibile una traduzione, non verranno modificati i documenti creati o scaricati.
- ▶ **Fuso orario:** consente di impostare il fuso orario selezionando l'area geografica desiderata seguita dalla città più vicina. Se Raspberry Pi è collegato alla rete ma l'orologio mostra l'ora sbagliata, è probabile che il fuso orario non sia stato impostato correttamente.
- ▶ **Tastiera:** permette di scegliere il tipo di tastiera, la lingua e il layout. Se la tua tastiera digita le lettere o i simboli sbagliati, puoi impostarla correttamente da qui.
- ▶ **Nazione WiFi:** consente di impostare il proprio Paese ai fini del regolamento sulle radiocomunicazioni. Assicurati di selezionare il Paese in cui il tuo Raspberry Pi viene utilizzato: selezionando un altro Paese,

potresti non riuscire a connetterti ai punti di accesso LAN wireless nelle vicinanze e commettere una violazione della legge sulle trasmissioni radiotelevisive. Prima di poter utilizzare il WiFi è necessario impostare un Paese.



Appendice F

Specifiche tecniche di Raspberry Pi

I vari componenti e le caratteristiche di un computer sono noti come *specifiche tecniche* e forniscono le informazioni necessarie per mettere a confronto computer di tipologie diverse. Queste specifiche possono confondere. Non è necessario conoscerle o capirle per utilizzare un Raspberry Pi, ma le includiamo per i lettori più curiosi.

Raspberry Pi 5

Il sistema su chip (system-on-chip, SoC) di Raspberry Pi 5 è un Broadcom BCM2712, una sigla che vedrai scritta sulla placca metallica, se osservi attentamente. Questo sistema è dotato di quattro core di unità di elaborazione centrale (CPU) Arm Cortex-A76 a 64 bit, ognuno dei quali funziona a 2,4 GHz, e di un'unità di elaborazione grafica (GPU) Broadcom VideoCore VII che funziona a 800 MHz per video e rendering 3D come i giochi.

Il sistema su chip è dotato di RAM (memoria ad accesso casuale) a 4 GB o 8 GB LPDDR4X (Low-Power Double-Data-Rate 4) che funziona a 4.267 MHz. Questa memoria è condivisa tra processore centrale e processore grafico. Lo slot per schede microSD supporta fino a 512 GB di memoria.

La porta Ethernet supporta connessioni fino a 1 gigabit (1000 Mbps, 1000-Base-T), mentre la radio supporta le reti Wi-Fi 802.11ac che operano sulle bande di frequenza 2,4 GHz e 5 GHz, Bluetooth 5.0 e connessioni Bluetooth Low Energy (BLE).

Raspberry Pi 5 dispone di due porte USB 2.0 e due porte USB 3.0 per le periferiche, oltre che di un connettore per una singola linea PCI Express (PCIe) 2.0 ad alta velocità. Grazie a un accessorio HAT opzionale, questo connettore può essere utilizzato per aggiungere unità di archiviazione M.2 Solid State Drive (SSD) ad alta velocità, acceleratori per il machine learning (ML) e la visione artificiale (CV) e altro hardware.

Raspberry Pi 4 e 400

- ▶ **CPU:** Arm Cortex-A72 64 bit quad-core (Broadcom BCM2711) da 1,5 GHz o 1,8 GHz (Raspberry Pi 400)
- ▶ **GPU:** VideoCore VI da 500 MHz
- ▶ **RAM:** LPDDR4 da 1 GB, 2 GB, 4 GB (Raspberry Pi 400), o 8 GB
- ▶ **Rete :** 1 Gigabit Ethernet, dual-band 802.11ac, Bluetooth 5.0, BLE
- ▶ **Uscite audio/video:** 1 jack AV analogico da 3,5 mm (solo Raspberry Pi 4), 2 porte micro-HDMI 2.0
- ▶ **Connettività periferiche:** 2 porte USB 2.0, 2 porte USB 3.0, 1 CSI (solo Raspberry Pi 4), 1 DSI (solo Raspberry Pi 4)
- ▶ **Memoria:** 1 microSD fino a 512 GB (16 GB nel kit Raspberry Pi 400)
- ▶ **Alimentazione:** 5 V a 3 A tramite USB C, PoE (con HAT aggiuntivo, solo Raspberry Pi 4)
- ▶ **Extra:** terminale GPIO a 40 pin

Raspberry Pi Zero 2 W

- ▶ **CPU:** Arm Cortex-A53 quad-core a 64 bit da 1 GHz (Broadcom BCM2710)
- ▶ **GPU:** VideoCore IV da 400MHz
- ▶ **RAM:** LPDDR2 da 512 MB
- ▶ **Rete:** 802.11b/g/n a banda singola, Bluetooth 4.2, BLE
- ▶ **Uscite audio/video:** 1 mini-HDMI
- ▶ **Connettività periferiche:** 1 porta Micro USB OTG 2.0, 1 CSI
- ▶ **Memoria:** 1 microSD fino a 512 GB
- ▶ **Alimentazione:** 5 volt a 2,5 ampere tramite micro USB
- ▶ **Extra:** terminale GPIO a 40 pin (non popolato)

Raspberry Pi è un piccolo, intelligente computer costruito nel Regno Unito e pieno di potenzialità. Realizzato utilizzando un processore di classe desktop ad alta efficienza energetica, Raspberry Pi è progettato per aiutarti a imparare a programmare, scoprire come funzionano i computer e costruire le tue incredibili invenzioni. Questo libro è stato scritto per mostrarti quanto sia facile iniziare.

Impara come:

- Configurare il tuo Raspberry Pi, installare il suo sistema operativo e iniziare a utilizzare questo computer completamente funzionale.
- Iniziare progetti di programmazione con guide passo dopo passo, utilizzando i linguaggi di programmazione Scratch 3, Python e MicroPython.
- Sperimentare collegando componenti elettronici e divertirsi creando progetti sorprendenti.

Novità nella 5ª edizione:

- Aggiornato per gli ultimi computer Raspberry Pi: Raspberry Pi 5 e Raspberry Pi Zero 2 W.
- Copre l'ultimo sistema operativo Raspberry Pi.
- Include un nuovo capitolo su Raspberry Pi Pico!



raspberrypi.com

ISBN 978-1-912047-39-0

UK £19.99

US \$24.99



9 781912 047390