



STARTER KIT OTRONIC ARDUINO UNO

Starter kit Arduino UNO: tutorial e 16 progetti

Passo dopo passo: che cos'è un Arduino, che cosa c'è nel tuo kit,
come funziona Codey e 16 progetti da costruire da solo.

codey.online/it/starter-kits/uno/tutorial



Indice

Benvenuto! Lo costruirai tu! 🚀	3
Che cos'è un Arduino UNO?	4
Che cosa c'è nel tuo starter kit?	5
Breadboard	6
Cavi jumper	7
LED (lampadina)	8
Resistenza 220 Ω	9
Pulsante	10
Buzzer attivo	11
Sensore di luce (LDR)	12
Sensore di temperatura DHT11	13
Sensore di distanza HC-SR04	14
Sensore di movimento (PIR)	15
Piccolo schermo OLED 0,91"	16
Servomotore SG90	17
Anello LED RGB (8 LED)	18
Relè 5V	19
Clip per batteria 9V	20
Che cos'è il vibe coding?	21
Crea il tuo account gratis	22
Così funziona Codey	23
Collegare e caricare	24
Iniziamo! 🎉	25
Il tuo primo circuito su breadboard	26
Luce lampeggiante	27
Semaforo	28
Pulsante luce	29
Campanello	30
Anello arcobaleno	31
Mini stazione meteo	32
Sensore di parcheggio	33
Lucina notturna	34
Allarme di movimento	35
Sbarra	36
Gioco dei riflessi	37
Dado elettronico	38
Sensore di parcheggio deluxe	39
Misuratore di distanza con barra luminosa	40
Ventilatore intelligente	41
Sbarra automatica	42

STARTER KIT OTRONIC ARDUINO UNO

Benvenuto! Lo costruirai tu! 🚀

Con questo starter kit costruisci i tuoi piccoli dispositivi intelligenti. Una stazione meteo, un sensore di parcheggio, un allarme... E la cosa più bella: non devi saper programmare. Codey ti aiuta.

Dai una rapida occhiata a questo breve tutorial. Così sai che cosa c'è nel tuo kit e che cosa puoi farci. Poi inizi subito con il tuo primo progetto.

Funziona così:

- 2 **Che cos'è un Arduino e che cosa puoi farci?**
- 3 **Tutti i componenti del tuo kit, uno per uno**
- 4 **Che cos'è il vibe coding? Programmare con l'AI**
- 5 **Creare il tuo account gratuito**
- 6 **Ecco come funziona Codey**
- 7 **Collegare il tuo Arduino e caricare il tuo codice**
- 8 **Iniziare con 16 progetti**

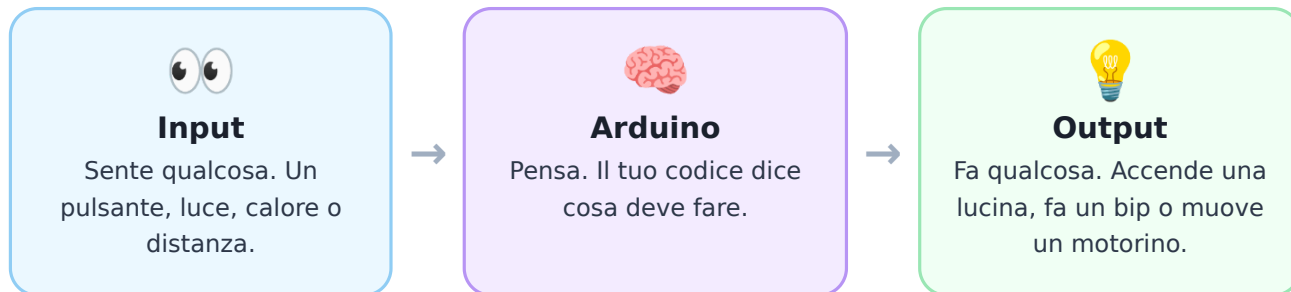


🕒 Ci vogliono circa 15 minuti. Puoi sempre saltare e tornare più tardi.

Che cos'è un Arduino UNO?

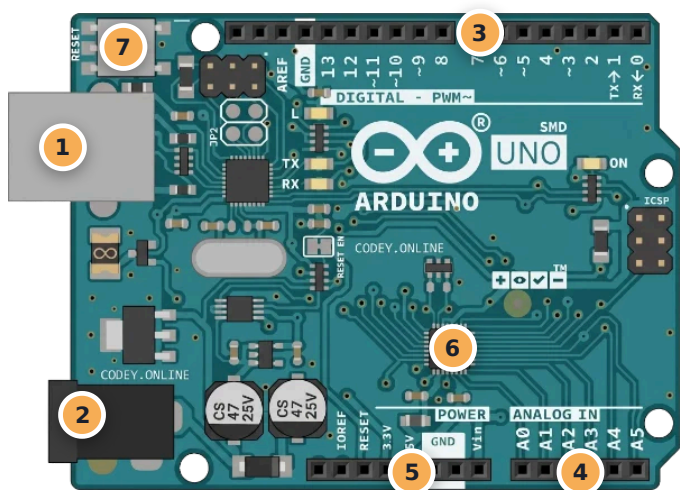
Arduino è un piccolo computer. Tu gli dici cosa deve fare, e lui lo fa. Sempre di nuovo, senza stancarsi.

Così pensa un Arduino



Esempio: il sensore di distanza vede che ti avvicini (input). Arduino pensa: "troppo vicino!" E il buzzer inizia a suonare (output). Così funziona un sensore di parcheggio in un'auto.

Che cosa c'è sulla scheda?



- 1 Porta USB: qui va il cavo verso il tuo computer. Da lì riceve corrente e codice.
- 2 Connettore di alimentazione: per una batteria o un alimentatore (7-12V), se non usi il computer.
- 3 Pin digitali da D0 a D13: acceso o spento. Per lucine, pulsanti e buzzer.
- 4 Pin analogici da A0 a A5: misurano quanto. Per esempio quanta luce c'è.
- 5 Pin di alimentazione: 5V (più) e GND (meno) danno corrente ai tuoi componenti.
- 6 Il chip: il cervello. Qui viene messo il tuo codice.
- 7 Pulsante reset: fa ripartire il tuo codice.

Due parole che vedrai spesso:

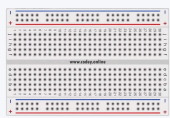
- **5V** è il più. Da qui arriva la corrente (5 volt, come un caricatore del telefono).
- **GND** (ground) è il meno. Da qui la corrente torna indietro. Ogni componente ha bisogno di un filo verso GND.

I **pin** sono i forellini sui lati. Lì inserisci i fili. Un pin può essere un **ingresso** (ascolta) oppure un **uscita** (dà corrente). Lo decidi tu nel codice.

Che cosa c'è nel tuo starter kit?

Il tuo kit è pieno di componenti. Alcuni sentono qualcosa (sensori), altri fanno qualcosa (lucine, cicalino, motorino). Clicca su Avanti, così li conosci uno per uno.

Base: con questi costruisci tutto



Breadboard

La basetta di montaggio: costruire senza saldare

Nel tuo kit: 1x



Cavi jumper

Cavi Dupont: i fili che collegano tutto

Nel tuo kit: 40x



Resistenza 220 Ω

Rallenta la corrente

Nel tuo kit: 10x

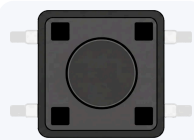


Clip per batteria 9V

Far funzionare il tuo progetto senza computer

Nel tuo kit: 1x

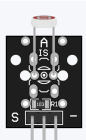
Input: questi componenti sentono o misurano qualcosa



Pulsante

Un piccolo tasto da premere

Nel tuo kit: 5x



Sensore di luce (LDR)

Misura quanta luce o quanto buio c'è

Nel tuo kit: 1x



Sensore di temperatura DHT11

Misura la temperatura e l'umidità dell'aria

Nel tuo kit: 1x



Sensore di distanza HC-SR04

Misura la distanza con il suono, proprio come un pipistrello

Nel tuo kit: 1x



Sensore di movimento (PIR)

Vede se qualcuno si muove

Nel tuo kit: 1x

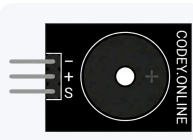
Output: questi componenti fanno qualcosa



LED (lampadina)

Una piccola lampadina in 5 colori

Nel tuo kit: 10x



Buzzer attivo

Fa dei bip

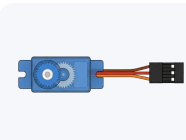
Nel tuo kit: 1x



Piccolo schermo OLED 0,91"

Un piccolo schermo per testo e numeri

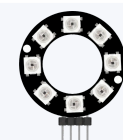
Nel tuo kit: 1x



Servomotore SG90

Un motorino che gira fino a una posizione

Nel tuo kit: 1x



Anello LED RGB (8 LED)

8 lucine intelligenti di ogni colore

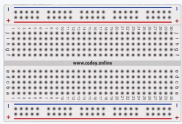
Nel tuo kit: 1x



Relè 5V

Un interruttore che il tuo Arduino può azionare

Nel tuo kit: 1x



Breadboard

La basetta di montaggio: costruire senza saldare

Nel tuo kit: 1x

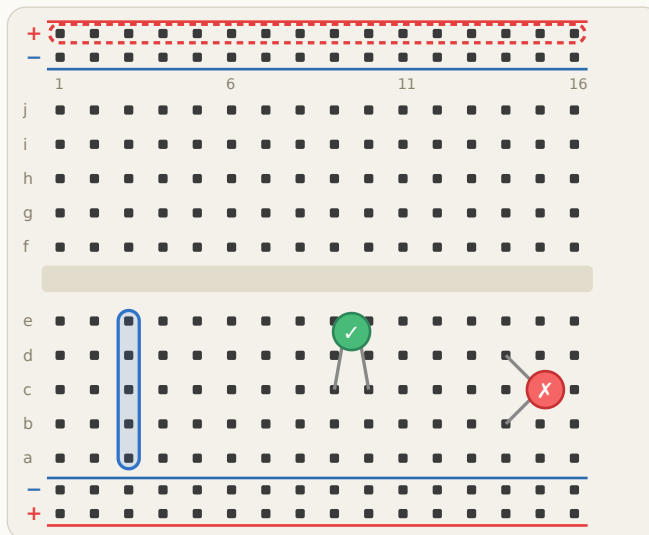
Che cos'è?

Una breadboard è una piccola basetta piena di fori. Ci infili componenti e fili. Sotto i fori ci sono strisce di metallo che collegano alcuni fori tra loro. Così costruisci un circuito senza saldare. Hai sbagliato? Lo tiri fuori e ricominci.

Così lo colleghi

Ecco come funziona:

- **File da 5:** nella parte centrale ci sono sempre 5 fori vicini collegati tra loro (a-e e f-j). Tutto quello che infili nella stessa fila è collegato insieme.
- **La fessura al centro** separa i due lati. Sinistra (a-e) e destra (f-j) quindi *non* sono collegate.
- **Strisce di alimentazione (+ e -)** sui lati lunghi sono collegate per tutta la lunghezza. Collega il + a 5V e il - a GND del tuo Arduino. Ora hai corrente dappertutto.



- Striscia di alimentazione: tutta la lunghezza è collegata
- 5 fori in una fila sono collegati
- ✗ Sbagliato: entrambi i piedini nella stessa fila

- Fessura: sinistra e destra non sono collegate
- ✓ Giusto: i piedini sono in file diverse

Che cosa puoi farci?

- Provare circuiti senza saldare
- Dare corrente a più componenti insieme tramite le strisce + e -
- Mettere bene in serie un LED con una resistenza

Consiglio

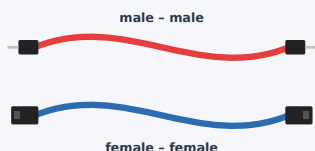
Non mettere mai i due piedini di un componente nella stessa fila da 5. Così sono collegati tra loro e la corrente passa semplicemente di lì. È meglio mettere un componente sopra la fessura.

🔧 **Provalo subito!** Nell'esercizio "Il tuo primo circuito su breadboard" costruisci un LED con una resistenza sulla breadboard e lo fai accendere e lampeggiare. →

Progetti con questo componente

★ Il tuo primo circuito su breadboard

Parte 2 di 15



Cavi jumper

Cavi Dupont: i fili che collegano tutto

Nel tuo kit: 40x

Che cos'è?

Con i cavi jumper colleghi il tuo Arduino ai componenti. Si chiamano anche **cavi Dupont**, per i connettori neri alle estremità. Nel tuo kit ci sono due tipi:

- **Male-male** (pin da entrambi i lati): da inserire in Arduino e nella breadboard.
- **Female-female** (foro da entrambi i lati): da infilare sui pin di un modulo.

Così lo colleghi

Inserisci bene il cavo nel foro o sopra il pin. Se unisci un cavo male e uno female, ottieni un cavo male-female: comodo per collegare un modulo direttamente al tuo Arduino.

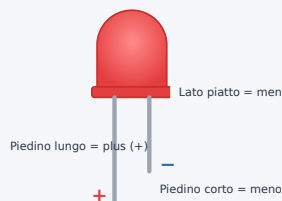
Che cosa puoi farci?

- Collegare il tuo Arduino alla breadboard
- Collegare moduli sensore
- Portare corrente (5V) e GND ai tuoi componenti

Consiglio

Usa colori fissi: rosso per 5V e nero per GND. Così vedi subito se la corrente è collegata bene.

Parte 3 di 15



LED (lampadina)

Una piccola lampadina in 5 colori

Nel tuo kit: 10x

Che cos'è?

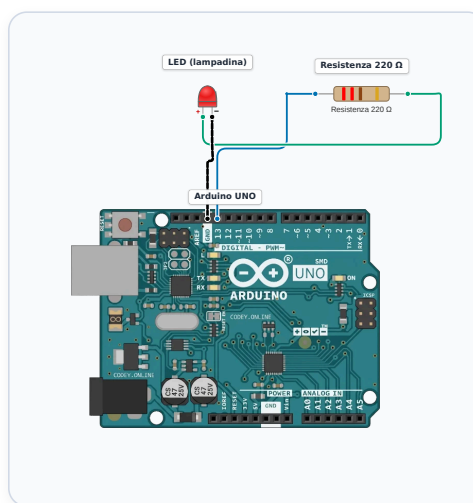
Un LED è una piccola lampadina che usa pochissima corrente. Il tuo kit ha 10 LED in 5 colori. Un LED funziona solo in un verso, quindi devi metterlo nel modo giusto.

Così lo colleghi

- Il **piedino lungo** è il plus (+). Va a un pin del tuo Arduino tramite una resistenza da 220 Ω.
- Il **piedino corto** è il meno (-). Va a GND. Da quel lato anche il bordo della lampadina è piatto.

Usa sempre una resistenza! Senza resistenza il LED riceve troppa corrente e si rompe.

Schema di collegamento



- Arduino UNO · D13 → Resistenza 220 Ω · 1
- Resistenza 220 Ω · 2 → LED (lampadina) · ANODE
- LED (lampadina) · CATHODE → Arduino UNO · GND

Che cosa puoi farci?

- Una luce lampeggiante
- Un semaforo
- Una lampadina che mostra se qualcosa è acceso o spento

Consiglio

Il tuo LED non fa niente? Giralo. Funziona solo in un verso.

Progetti con questo componente

★ Il tuo primo circuito su breadboard

★ Luce lampeggiante

★ Semaforo

★ Pulsante luce

★★ Allarme di movimento

★★ Gioco dei riflessi

Parte 4 di 15

Rosso, rosso, marrone = 220 Ω

Resistenza 220 Ω

Rallenta la corrente

Nel tuo kit: 10x

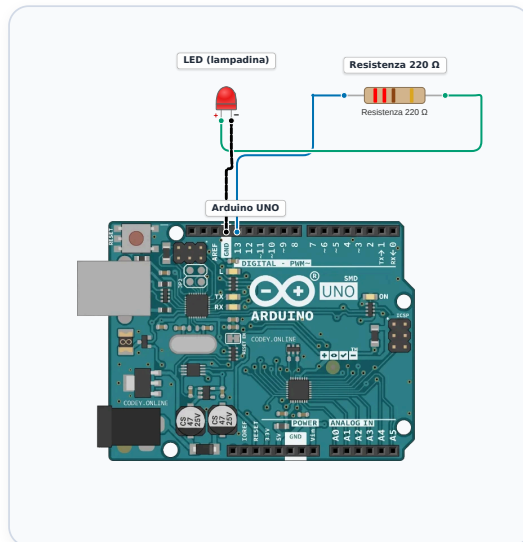
Che cos'è?

Una resistenza rallenta un pochino la corrente. Proprio come un rubinetto che chiudi a metà. Così un LED riceve la corrente giusta e non si rompe. Le resistenze nel tuo kit sono da 220 ohm (Ω). Lo vedi dagli anellini colorati: rosso, rosso, marrone.

Così lo colleghi

Una resistenza non ha più o meno: puoi metterla in entrambi i versi. Mettila tra il pin di Arduino e il piedino lungo del LED. Un lato va al pin, l'altro lato va al LED.

Schema di collegamento



- Arduino UNO · D13 → Resistenza 220 Ω · 1
- Resistenza 220 Ω · 2 → LED (lampadina) · ANODE
- LED (lampadina) · CATHODE → Arduino UNO · GND

Che cosa puoi farci?

- Proteggere ogni LED
- Limitare la corrente verso un componente

Consiglio

Una resistenza per ogni LED. Usi tre LED? Allora ti servono anche tre resistenze.

Progetti con questo componente

★ Il tuo primo circuito su breadboard

★ Luce lampeggiante

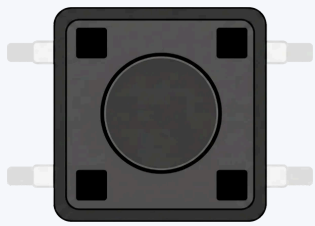
★ Semaforo

★ Pulsante luce

★★ Allarme di movimento

★★★ Gioco dei riflessi

Parte 5 di 15



Pulsante

Un piccolo tasto da premere

Nel tuo kit: 5x

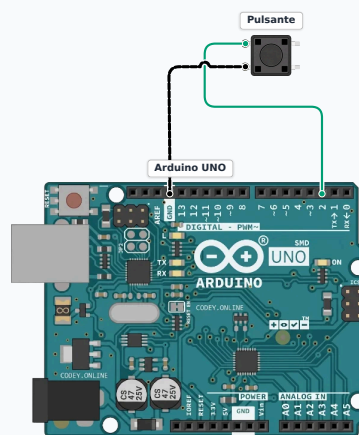
Che cos'è?

Un pulsante fa contatto mentre lo premi. Se lo lasci, il contatto sparisce di nuovo. Il tuo Arduino può sentire se il pulsante è premuto. Nel tuo kit ci sono 5 pulsanti.

Così lo colleghi

Il pulsante ha 4 piedini. Te ne servono solo 2: uno va a un pin (per esempio D2) e uno va a GND. Nel codice imposti il pin su `INPUT_PULLUP`. Così il pin è HIGH quando non premi e LOW quando premi.

Schema di collegamento



● Arduino UNO · D2 → Pulsante · 1

● Arduino UNO · GND → Pulsante · 2

Che cosa puoi farci?

- Accendere e spegnere una lucina
- Un campanello
- Un giochino: chi preme più veloce?

Consiglio

Non sei sicuro di quali piedini usare? Prendi due piedini che sono in diagonale, uno di fronte all'altro. Così va sempre bene. Metti il pulsante sopra la fessura della breadboard.

Progetti con questo componente

★ Pulsante luce

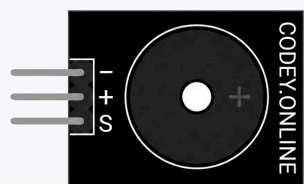
★ Campanello

★★ Sbarra

★★ Gioco dei riflessi

★★ Dado elettronico

Parte 6 di 15



Buzzer attivo

Fa dei bip

Nel tuo kit: 1x

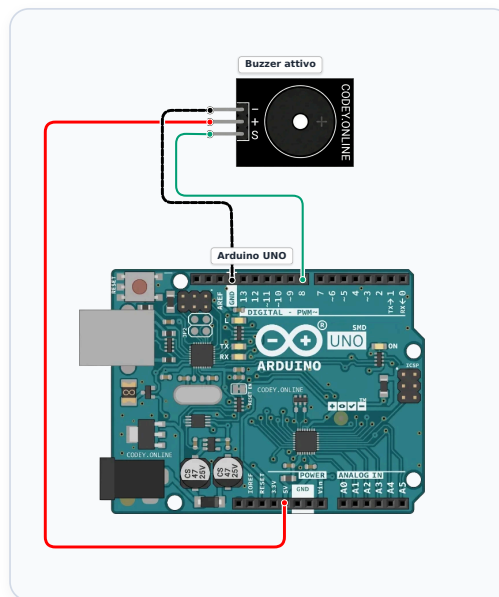
Che cos'è?

Il buzzer fa un suono: bip! È un buzzer **attivo**: suona da solo appena riceve corrente. Devi solo accenderlo e spegnerlo, proprio come una lucina.

Così lo colleghi

- **S** (segnale) a un pin, per esempio D8
- **+** a 5V
- **-** a GND

Schema di collegamento



● Arduino UNO · GND → Buzzer attivo · -

● Arduino UNO · 5V → Buzzer attivo · +

● Arduino UNO · D8 → Buzzer attivo · S

🌟 Che cosa puoi farci?

- Un allarme
- Un sensore di parcheggio che fa bip più veloci quando ti avvicini
- Un campanello o un timer

💡 Consiglio

Vuoi un bip corto? Accendi il buzzer, aspetta 100 millisecondi e spegnilo di nuovo.

Progetti con questo componente

★ Campanello

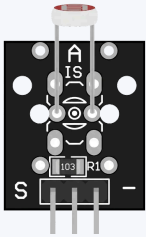
★★ Sensore di parcheggio

★★ Allarme di movimento

★★★★ Sensore di parcheggio deluxe

★★★★ Sbarra automatica

Parte 7 di 15



Sensore di luce (LDR)

Misura quanta luce o quanto buio c'è

Nel tuo kit: 1x

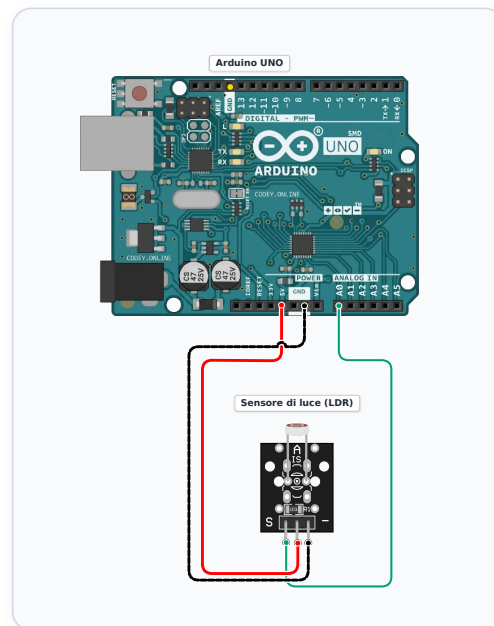
Che cos'è?

Questo sensore misura quanta luce c'è. Il tuo Arduino legge un numero tra 0 e 1023. Con tanta luce ottieni un numero diverso rispetto al buio. Così il tuo Arduino sa se è giorno o notte.

Così lo colleghi

- S a un pin analogico, per esempio A0
- + a 5V
- - a GND

Schema di collegamento



● Arduino UNO · A0 → Sensore di luce (LDR) · S

● Arduino UNO · 5V → Sensore di luce (LDR) · +

● Arduino UNO · GND → Sensore di luce (LDR) · -

✨ Che cosa puoi farci?

- Una lucina notturna che si accende da sola
- Un misuratore di luce
- Un allarme che suona se qualcuno apre il cassetto

💡 Consiglio

Guarda nel monitor seriale quale numero vedi. Poi tieni la mano sopra il sensore. Così vedi quale numero significa "buio".

Progetti con questo componente

★★ Lucina notturna

Parte 8 di 15



Sensore di temperatura DHT11

Misura la temperatura e l'umidità dell'aria

Nel tuo kit: 1x

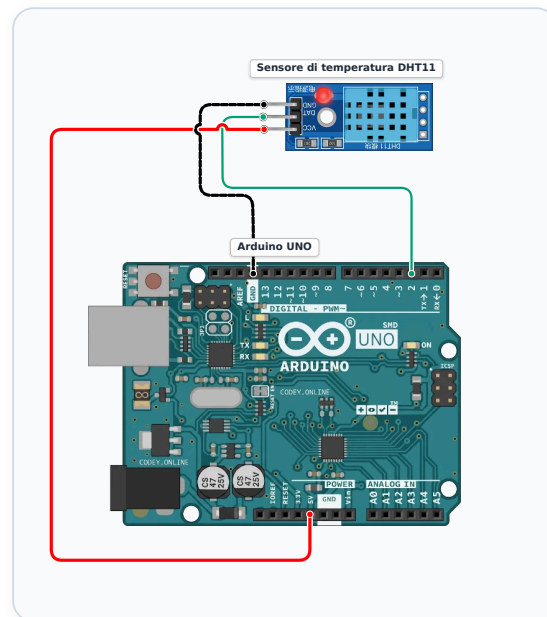
Che cos'è?

Il DHT11 misura quanto fa caldo (in gradi Celsius) e quanto è umida l'aria (in percentuale). Dà una nuova misura ogni 2 secondi.

Così lo colleghi

- **DAT** (data) a un pin, per esempio D2
- **VCC** a 5V
- **GND** a GND

Schema di collegamento



- Arduino UNO · 5V → Sensore di temperatura DHT11 · VCC
- Arduino UNO · GND → Sensore di temperatura DHT11 · GND
- Arduino UNO · D2 → Sensore di temperatura DHT11 · DAT

✨ Che cosa puoi farci?

- Una mini stazione meteo
- Una ventola che si accende quando fa troppo caldo
- Un avviso se c'è troppa umidità in bagno

💡 Consiglio

Prova a soffiare sul sensore. Vedrai subito salire l'umidità dell'aria!

Progetti con questo componente

★★ Mini stazione meteo

★★★★ Ventilatore intelligente

Parte 9 di 15



Sensore di distanza HC-SR04

Misura la distanza con il suono, proprio come un pipistrello

Nel tuo kit: 1x

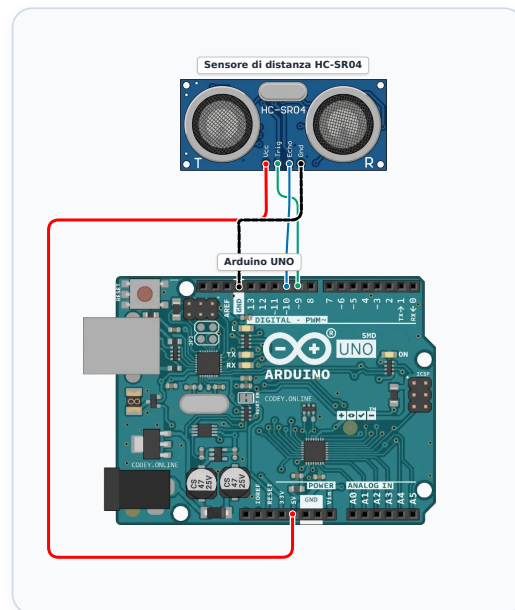
Che cos'è?

Questo sensore manda un piccolo suono che tu non puoi sentire. Il suono rimbalza contro qualcosa e torna indietro. Il sensore misura quanto tempo ci mette. Così Arduino sa quanto è lontano qualcosa: da 2 cm fino a circa 4 metri.

Così lo colleghi

- **Trig** (inviare) verso D9
- **Echo** (ricevere) verso D10
- **Vcc** verso 5V
- **Gnd** verso GND

Schema di collegamento



- Arduino UNO · 5V → Sensore di distanza HC-SR04 · Vcc
- Arduino UNO · GND → Sensore di distanza HC-SR04 · Gnd
- Arduino UNO · D9 → Sensore di distanza HC-SR04 · Trig
- Arduino UNO · D10 → Sensore di distanza HC-SR04 · Echo

🌟 Che cosa puoi farci?

- Un sensore di parcheggio
- Un misuratore di distanza
- Una sbarra che si apre da sola

💡 Consiglio

I due "occhi" sono l'altoparlante e il microfono. Puntali dritti verso ciò che vuoi misurare.

Progetti con questo componente

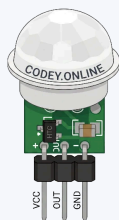
★★ Sensore di parcheggio

★★★★ Sensore di parcheggio deluxe

★★★★ Misuratore di distanza con barra luminosa

★★★★ Sbarra automatica

Parte 10 di 15



Sensore di movimento (PIR)

Vede se qualcuno si muove

Nel tuo kit: 1x

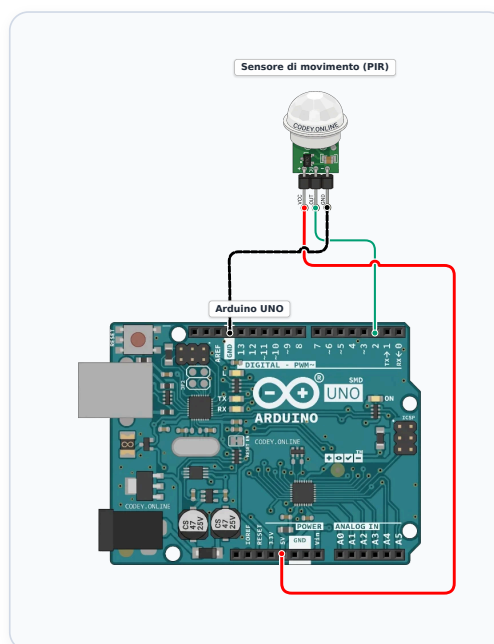
Che cos'è?

Un sensore PIR sente il calore di persone e animali. Qualcosa di caldo si muove davanti al sensore? Allora la sua uscita diventa HIGH. Così funzionano anche le luci che si accendono da sole quando ci passi davanti.

Così lo colleghi

- **OUT** verso un pin, per esempio D2
- **VCC** verso 5V
- **GND** verso GND

Schema di collegamento



● Arduino UNO · 5V → Sensore di movimento (PIR) · VCC

● Arduino UNO · GND → Sensore di movimento (PIR) · GND

● Arduino UNO · D2 → Sensore di movimento (PIR) · OUT

✨ Che cosa puoi farci?

- Un allarme antifurto
- Una lampada che si accende quando entri
- Contare quante volte qualcuno passa davanti

💡 Consiglio

Dopo l'avvio, il sensore ha bisogno di qualche secondo per abituarsi alla stanza. Resta fermo un attimo.

Progetti con questo componente

★★ Allarme di movimento

Parte 11 di 15



Piccolo schermo OLED 0,91"

Un piccolo schermo per testo e numeri

Nel tuo kit: 1x

Che cos'è?

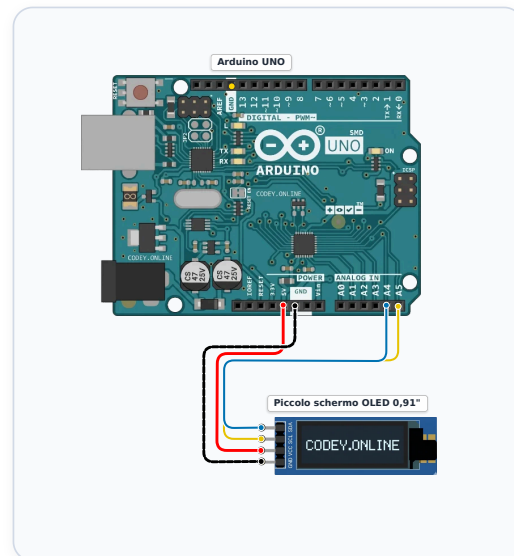
Un piccolo schermo nitido da 128 per 32 pixel. Puoi mostrarci testo, numeri e disegni semplici. Per esempio la temperatura o una distanza.

Così lo colleghi

Lo schermo parla tramite **I2C**: solo due fili dati.

- **SDA** verso A4
- **SCL** verso A5
- **VCC** verso 5V
- **GND** verso GND

Schema di collegamento



- Arduino UNO · A4 → Piccolo schermo OLED 0,91" · SDA
- Arduino UNO · A5 → Piccolo schermo OLED 0,91" · SCL
- Arduino UNO · 5V → Piccolo schermo OLED 0,91" · VCC
- Arduino UNO · GND → Piccolo schermo OLED 0,91" · GND

🌟 Che cosa puoi farci?

- Mostrare la temperatura
- Mostrare la distanza in centimetri
- Mostrare un punteggio o un tempo di reazione

💡 Consiglio

Lo schermo resta nero? Controlla che SDA e SCL non siano scambiati.

Progetti con questo componente

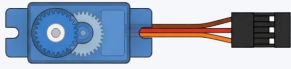
★★ Mini stazione meteo

★★ Gioco dei riflessi

★★★★ Sensore di parcheggio deluxe

★★★★ Ventilatore intelligente

Parte 12 di 15



Servomotore SG90

Un motorino che gira fino a una posizione

Nel tuo kit: 1x

Che cos'è?

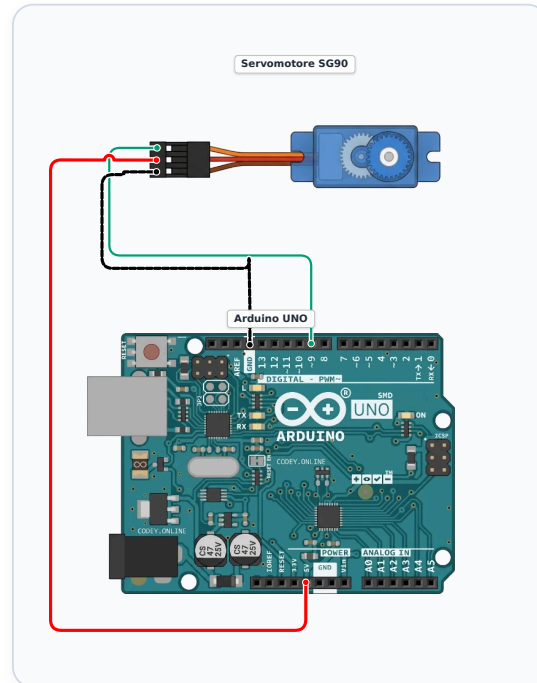
Un servo è un motorino che puoi far girare fino a una posizione precisa, da 0 a 180 gradi. Per esempio gli dici "vai a 90 gradi" e lui va proprio lì. Sul perno metti un braccetto.

Così lo colleghi

Il servo ha tre fili:

- **Arancione o giallo** (segnale) verso un pin, per esempio D9
- **Rosso** verso 5V
- **Marrone** verso GND

Schema di collegamento



● Arduino UNO · GND → Servomotore SG90 · GND

● Arduino UNO · 5V → Servomotore SG90 · VCC

● Arduino UNO · D9 → Servomotore SG90 · PWM

✨ Che cosa puoi farci?

- Una sbarra
- Una lancetta che indica la temperatura
- Un piccolo braccio robotico che saluta

Consiglio

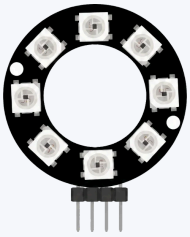
Non girare il braccetto con la mano. Altrimenti gli ingranaggi dentro possono rompersi.

Progetti con questo componente

★★ Sbarra

★★★★ Sbarra automatica

Parte 13 di 15



Anello LED RGB (8 LED)

8 lucine intelligenti di ogni colore

Nel tuo kit: 1x

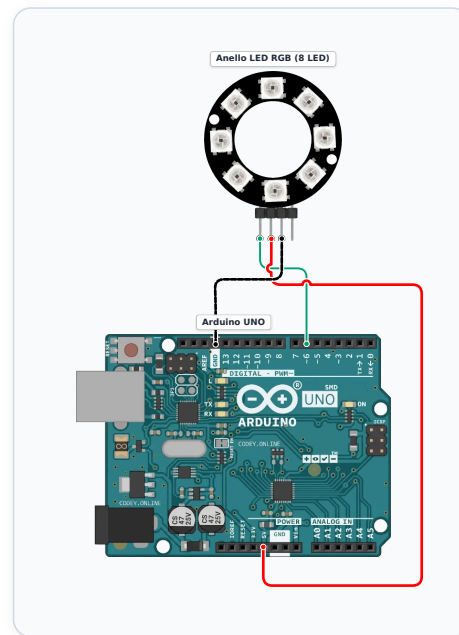
Che cos'è?

Un anello con 8 lucine intelligenti. Ogni lucina può diventare di qualsiasi colore, e le controlli tutte una per una. Con un solo filo dati! Super divertente per effetti di luce.

Così lo colleghi

- **DI** (data in) a un pin, per esempio D6
- **5V** a 5V
- **GND** a GND
- **DO** lo lasci libero (lì puoi collegare un altro anello dopo)

Schema di collegamento



- Arduino UNO · 5V → Anello LED RGB (8 LED) · 5V
- Arduino UNO · GND → Anello LED RGB (8 LED) · GND
- Arduino UNO · D6 → Anello LED RGB (8 LED) · DI

⚡ Che cosa puoi farci?

- Una lampada arcobaleno
- Una barra luminosa che mostra quanto è vicino qualcosa
- Un dado fatto di luce

💡 Consiglio

Non mettere la luminosità troppo alta nel codice. È meglio per i tuoi occhi e per il tuo Arduino.

Progetti con questo componente

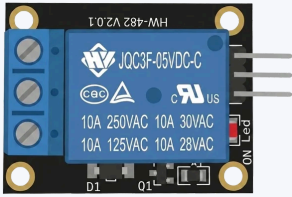
★ Anello arcobaleno

★★ Lucina notturna

★★ Dado elettronico

★★★★ Misuratore di distanza con barra luminosa

Parte 14 di 15



Relè 5V

Un interruttore che il tuo Arduino può azionare

Nel tuo kit: 1x

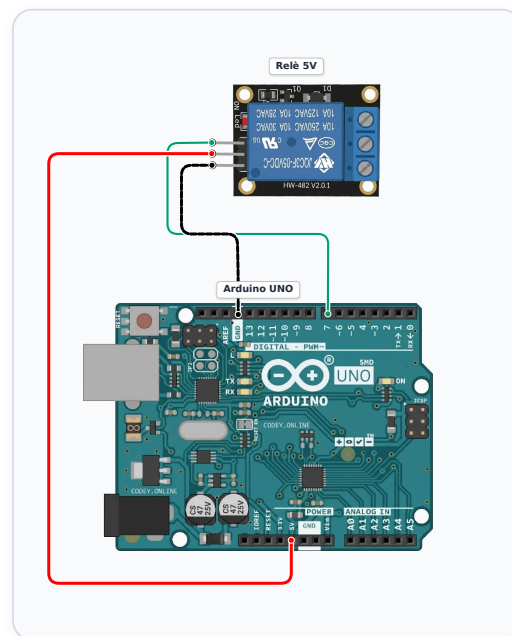
Che cos'è?

Un relè è un interruttore che il tuo Arduino può accendere e spegnere. Lo senti fare "clac". Con un relè il tuo Arduino può accendere e spegnere un altro apparecchio, per esempio una piccola ventola o una lampada a batteria.

Così lo colleghi

- **S** a un pin, per esempio D7
- **+** a 5V
- **-** a GND
- L'apparecchio che vuoi comandare lo colleghi a **COM** e **NO**. NO significa: di solito spento, acceso quando il relè fa clic.

Schema di collegamento



● Arduino UNO · GND → Relè 5V · -

● Arduino UNO · 5V → Relè 5V · +

● Arduino UNO · D7 → Relè 5V · S

🌟 Che cosa puoi farci?

- Una ventola che si accende quando fa caldo
- Una lampada con un timer
- Una piccola pompa per le piante

💡 Consiglio

Prima la sicurezza: usa il relè solo con batterie o bassa tensione (fino a 12V). Mai con la presa di corrente (230V)!

Progetti con questo componente

★★★★ Ventilatore intelligente

Parte 15 di 15



Clip per batteria 9V

Far funzionare il tuo progetto senza computer

Nel tuo kit: 1x

Che cos'è?

Con questa clip colleghi una batteria 9V al tuo Arduino. Così il tuo progetto funziona anche senza computer. Comodo se vuoi mostrarlo o metterlo da qualche parte.

Così lo colleghi

Aggancia la batteria alla clip e inserisci lo spinotto rotondo nella presa di alimentazione nera del tuo Arduino. Il tuo codice è ancora su Arduino, quindi parte subito.

Che cosa puoi farci?

- Appendere il tuo sensore di parcheggio in garage
- Mettere la tua stazione meteo sul davanzale
- Mostrare il tuo progetto agli amici

Consiglio

Togli la batteria quando hai finito. Così dura molto più a lungo.

Che cos'è il vibe coding?

Una volta dovevi prima imparare a programmare per mesi. Ora dici semplicemente con le tue parole cosa vuoi costruire. L'AI scrive il codice per te.

Questo si chiama **vibe coding**: tu hai l'idea, l'AI fa il lavoro di scrittura. Per esempio dici: *"Crea una lucina notturna che si accende quando fa buio."* In pochi secondi il codice è pronto, con anche uno schema dei collegamenti.

L'AI di oggi è molto brava a programmare. Spesso è persino più brava e più veloce di una persona. Conosce tutti i componenti, fa meno errori di battitura e spiega tutto con calma.

Costruire un progetto: prima e adesso

Programmare da soli

Cercare, scrivere, trovare errori, riprovare...

± 10 ore

Con Codey

Dire cosa vuoi, caricare, fatto.

± 5 minuti ✂

Ecco come funziona



Tu dici cosa vuoi costruire.



Codey scrive il codice e disegna come collegare tutto.



Tu lo costruisci uguale e carichi il codice.



Vuoi qualcosa di diverso? Dillo, e Codey lo modifica.

E intanto impari anche! Codey spiega cosa fa il codice. Così capisci sempre meglio come funziona.

Crea il tuo account gratis

Per lavorare con Codey ti serve un account. È gratis e ci vuole solo un minutino. Con il tuo starter kit poi puoi usare Codey senza limiti.

1

Registrati

Clicca su "Crea account" e inserisci il tuo indirizzo e-mail e una password. Hai un account Google? Allora clicca su "Continua con Google" e hai già finito.

2

Conferma il tuo indirizzo e-mail

Riceverai una e-mail da Codey Online. Clicca sul link in quella e-mail. Non vedi la e-mail? Guarda anche nella posta indesiderata.

3

Accedi

Accedi con il tuo indirizzo e-mail e la password. Poi tornerai automaticamente in questo tutorial, al passo successivo.

Ho già un account: accedi →

👉 Vai su codey.online/uno per creare il tuo account gratis.



Sei ancora piccolo? Allora crea l'account insieme a tuo papà, tua mamma o un altro adulto.

Così funziona Codey

Codey è il tuo compagno di programmazione. È tutto in una sola schermata. Questi sono i pulsanti più importanti.

The screenshot shows the Codey Online web interface. On the left is a sidebar with a chat area, a project list, and navigation buttons. The main area is divided into three sections: a code editor on the left, a chat area in the middle, and a right-hand panel with instructions and a circuit diagram. Numbered callouts 1 through 6 highlight key features: 1. Chat, 2. Code editor, 3. Compile and upload button, 4. Circuit diagram, 5. Serial monitor, and 6. My kit button.

1 La chat

Qui parli con Codey. Scrivi cosa vuoi creare o cambiare, con parole semplici.

2 Il codice

Qui c'è il codice che Codey scrive per te. Puoi anche scriverti tu.

3 Compila e carica sulla scheda

Un solo pulsante per tutto: Codey controlla il tuo codice, lo prepara (compila) e lo manda al tuo Arduino con il cavo USB (caricamento). C'è un errore? Allora Codey spesso lo risolve da solo.

4 Schema di collegamento

Mostra quale filo va dove. Lo vedi nella chat; cliccaci sopra per vederlo grande.

5 Monitor seriale

Qui vedi i messaggi che il tuo Arduino invia, per esempio la temperatura o una distanza.

6 Il mio kit

Tutti i componenti del tuo starterkit. Codey usa solo cose che hai tu.

Buono a sapersi: con il tuo starterkit usi Codey senza limiti nelle chat del tuo kit. Puoi chiedere tutto a Codey, anche "cosa fa questa riga?" o "perché non funziona?".

Collegare e caricare

Il tuo codice è pronto? Allora adesso va nel tuo Arduino. Lo fai in tre passi.

Importante: usa **Google Chrome** o **Microsoft Edge** sul tuo computer. Solo questi browser possono parlare con il tuo Arduino via USB. Safari, Firefox e la maggior parte dei telefoni non possono farlo (ancora).

1 Collega il tuo Arduino

Inserisci il cavo USB nel tuo Arduino e nel tuo computer. Si accende una lucina verde (ON).

2 Clicca su Compila e carica sulla scheda

Codey controlla il tuo codice e lo prepara (compilazione). La prima volta a volte ci vuole un minutino.

3 Scegli il tuo Arduino nella finestra delle porte

Appare una finestra con le porte. Scegli il tuo Arduino e clicca su Connetti. Le lucine RX e TX lampeggiano un attimo: il tuo codice entra dentro (upload). Fatto!

Come si chiama il mio Arduino nella finestra delle porte?

Dipende dal tuo computer. Ne vedi più di una? Stacca un attimo il cavo e guarda quale sparisce. Quella è il tuo Arduino.

Windows

Una **porta COM** con un numero, per esempio `Arduino Uno (COM4)` o `USB-SERIAL CH340 (COM5)`. COM1 quasi mai è il tuo Arduino.

Mac

Qualcosa con `usbmodem` o `usbserial`, per esempio `/dev/cu.usbmodem14101`, `cu.usbserial-1420` o `cu.wchusbserial1420`. Scegli sempre la versione `cu.`.

Linux

`/dev/ttyACM0` o `/dev/ttyUSB0`. Non hai accesso? Aggiungi te stesso al gruppo `dialout` e fai di nuovo login.

Chromebook

Chrome mostra il nome del dispositivo, per esempio `Arduino Uno`, `USB2.0-Serial` o `USB Serial`.

Monitor seriale: alcuni progetti inviano messaggi al tuo computer, per esempio la temperatura. Li vedi se in Codey clicchi su Monitor seriale. Così vedi cosa "pensa" il tuo Arduino.

Non funziona? Niente panico!

▼ Non vedo il mio Arduino nella finestra delle porte.

Prova un altro cavo USB. Alcuni cavi possono solo caricare e non inviano dati. Prova anche un'altra porta USB sul tuo computer.

▼ Vedo CH340 o USB2.0-Serial, ma l'upload non funziona (Windows/Mac).

Allora il tuo Arduino usa un chip CH340. Installa una volta sola il driver CH340 gratuito (cerca "CH340 driver") e riavvia il computer.

▼ L'upload si blocca a metà.

Chiudi gli altri programmi che usano Arduino, come Arduino IDE o una seconda scheda di Codey. Stacca un attimo il cavo, ricollegalo e prova di nuovo.

▼ Funziona, ma il mio circuito non fa niente.

Controlla i fili con lo schema di collegamento. 5V e GND sono messi bene? Poi chiedi a Codey: "Il mio LED non si accende, cosa può essere?"

Iniziamo! 🎉

Scegli un progetto e clicca su CREA QUESTO. Codey prepara per te il codice e lo schema dei collegamenti. Inizia con un progetto con una stella ★ se non hai mai fatto nulla con Arduino.



★ ⌚ 10 minuti

Il tuo primo circuito su breadboard

Fai pratica con la breadboard: fai accendere un LED e poi fallo lampeggiare.



★ ⌚ 10 minuti

Luce lampeggiante

Il tuo primissimo progetto: far lampeggiare un LED.



★ ⌚ 15 minuti

Semaforo

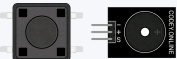
Tre LED che funzionano come un vero semaforo.



★ ⌚ 15 minuti

Pulsante luce

Premi il pulsante e la lampadina si accende. Ancora una volta: si spegne.



★ ⌚ 10 minuti

Campanello

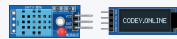
Premi il pulsante per un allegro din-don.



★ ⌚ 10 minuti

Anello arcobaleno

L'anello LED gira con tutti i colori dell'arcobaleno.



★★ ⌚ 20 minuti

Mini stazione meteo

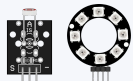
Temperatura e umidità dell'aria sul tuo schermo.



★★ ⌚ 20 minuti

Sensore di parcheggio

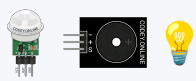
Più sei vicino, più suona veloce. Proprio come in auto.



★★ ⌚ 15 minuti

Lucina notturna

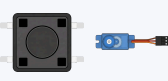
Diventa buio? Allora la lampada si accende da sola.



★★ ⌚ 20 minuti

Allarme di movimento

Passa qualcuno? Allora scatta l'allarme!



★★ ⌚ 20 minuti

Sbarra

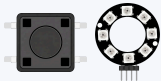
Premi il pulsante e la sbarra si apre.



★★ ⌚ 25 minuti

Gioco dei riflessi

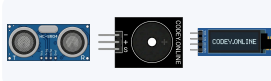
Quanto sei veloce? Misura il tuo tempo di reazione.



★★ ⌚ 15 minuti

Dado elettronico

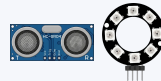
Premi il pulsante e lancia un dado di luce.



★★★ ⌚ 30 minuti

Sensore di parcheggio deluxe

Bip e distanza in centimetri su un piccolo display.



★★★ ⌚ 25 minuti

Misuratore di distanza con barra luminosa

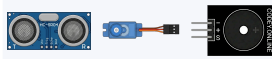
Più sei vicino, più lucine: verde, arancione, rosso.



★★★ ⌚ 30 minuti

Ventilatore intelligente

Fa troppo caldo? Allora il relé accende la ventola.



★★★ ⌚ 30 minuti

Sbarra automatica

Arriva un'auto? Allora la sbarra si apre da sola.

Hai un'idea tutta tua? Ancora meglio! Apri Codey e racconta cosa vuoi creare. Per esempio: "Crea una lucina che lampeggia più velocemente quando fa più caldo."

★ Principiante ⌚ 10 minuti

Il tuo primo circuito su breadboard

Un esercizio per scoprire come funziona una breadboard. Costruisci un circuito con un LED e una resistenza sulla breadboard. Prima lo fai accendere con la striscia 5V, senza codice. Poi lo fai lampeggiare con il pin 13.

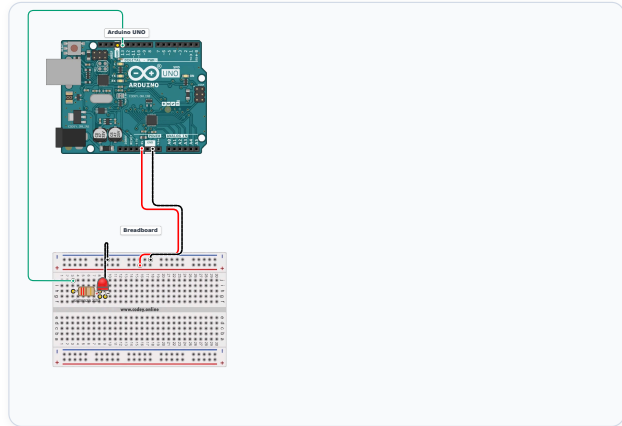
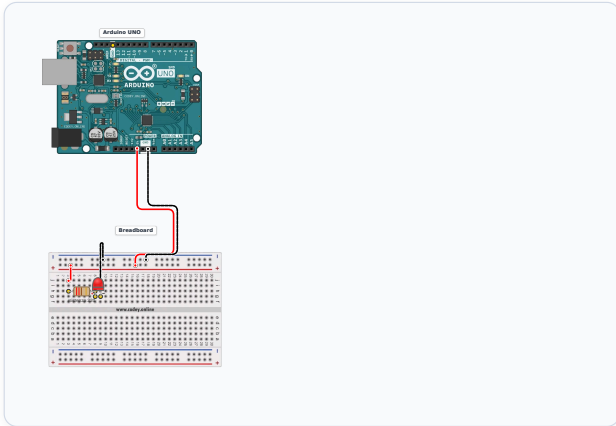
Schema di collegamento

Passo A: fai accendere il LED (senza codice)

Il filo rosso va dalla striscia + alla fila 3. La corrente fa così: 5V → striscia + → resistenza → LED → striscia - → GND. Il LED si accende? Allora hai usato bene la breadboard!

Passo B: fai lampeggiare il LED con il pin 13

Togli il filo rosso tra la striscia + e la fila 3. Al suo posto metti un filo dal pin 13 alla fila 3. Ora è Arduino a decidere quando il LED riceve corrente.



Che cosa ti serve?



1× Arduino UNO



1× Breadboard



1× LED (lampadina)



1× Resistenza 220 Ω



Cavo USB + Cavetti jumper

Così lo costruisci

1. Collega 5V di Arduino alla striscia + in alto sulla breadboard (filo rosso).
2. Collega GND di Arduino alla striscia - accanto (filo nero). Ora hai due strisce di corrente.
3. Metti la resistenza da 220 Ω dal foro H3 al foro H8. Così sta in due file diverse.
4. Metti la gambetta lunga del LED in G8 (la stessa fila della resistenza) e la gambetta corta in G9.
5. Collega il foro J9 alla striscia - (filo nero). Così la corrente può tornare a GND.
6. Passo A: collega la striscia + al foro J3 (filo rosso). Il LED si accende!
7. Passo B: togli quel filo rosso e collega il pin 13 al foro J3 (filo verde).
8. Fai creare il codice a Codey: clicca su CREA QUESTO, oppure scrivi tu nella chat: "Fai lampeggiare il LED sul pin 13 ogni secondo." Poi clicca su Compila e carica sulla scheda.

Il codice

```
// Il tuo primo circuito su breadboard
// Il LED sulla breadboard si accende e si spegne. Riceve corrente dal pin 13.

const int LED_PIN = 13; // Il filo verde va dal pin 13 alla breadboard

void setup() {
  pinMode(LED_PIN, OUTPUT); // Pin 13 è un'uscita: dà corrente
}

void loop() {
  digitalWrite(LED_PIN, HIGH); // LED acceso
  delay(1000);                // Aspetta 1 secondo
  digitalWrite(LED_PIN, LOW);  // LED spento
  delay(1000);                // Aspetta 1 secondo
}
```

Che cosa fa il codice?

- Il pin 13 è un'uscita. Con **HIGH** dà 5V, con **LOW** non dà nulla.
- In **loop()** accendiamo il LED, aspettiamo 1 secondo, lo spegniamo e aspettiamo di nuovo.
- La corrente passa nella breadboard: pin 13 → fila 3 → resistenza → fila 8 → LED → fila 9 → striscia - → GND.

Prova anche...

- Fai lampeggiare il LED due volte più veloce
- Aggiungi un secondo LED sulla breadboard che lampeggia a turno
- Fai lampeggiare il LED tre volte per poco e poi lascio spento un momento

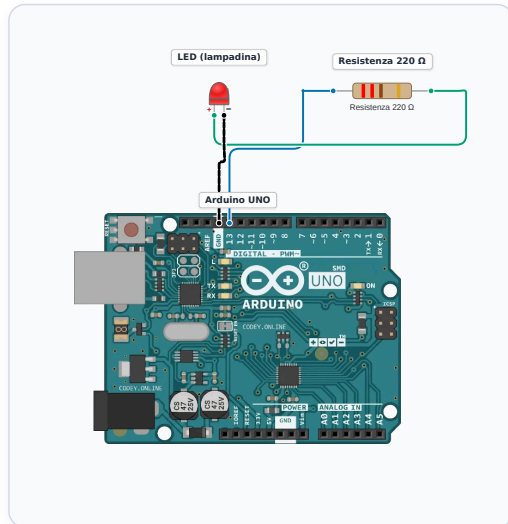
🔧 Costruisci questo progetto con Codey: apri codey.online/it/starter-kits/uno/projects/breadboard-led e clicca su FAI QUESTO. Così il codice e lo schema di collegamento sono subito pronti.

★ Principiante ⌚ 10 minuti

Luce lampeggiante

Il tuo primo progetto! Un LED si accende e si spegne, proprio come una luce lampeggiante. Così impari come Arduino accende e spegne un'uscita.

Schema di collegamento



Che cosa ti serve?



Così lo costruisci

1. Metti il LED nella breadboard, con i piedini in due file diverse.
2. Metti una resistenza da 220 Ω dalla fila del piedino lungo a un'altra fila.
3. Collega quell'altra fila con un cavetto al pin 13 del tuo Arduino.
4. Collega il piedino corto con un cavetto a GND.

Il codice

```
// Luce lampeggiante
// Un LED si accende e si spegne sempre, proprio come una luce lampeggiante.

const int LED_PIN = 13; // Il LED è sul pin 13 (tramite la resistenza da 220 ohm)
const int TEMPO_ATTESA = 500; // Per quanto tempo il LED resta acceso o spento (in millisecondi)

void setup() {
  // setup() viene eseguita una volta quando Arduino parte
  pinMode(LED_PIN, OUTPUT); // Il pin 13 è un'uscita: dà corrente
}

void loop() {
  // loop() gira sempre di nuovo, per sempre
  digitalWrite(LED_PIN, HIGH); // LED acceso
  delay(TEMPO_ATTESA); // Aspetta un po'
  digitalWrite(LED_PIN, LOW); // LED spento
  delay(TEMPO_ATTESA); // Aspetta un po'
}
```

Che cosa fa il codice?

- In `setup()` diciamo che il pin 13 è un'uscita.
- In `loop()` accendiamo il LED, aspettiamo un po', lo spegniamo e aspettiamo di nuovo.
- Poi `loop()` ricomincia da capo. Così il LED continua a lampeggiare.

Prova anche...

- Fai lampeggiare il LED due volte più in fretta
- Trasformalo in un segnale SOS: corto-corto-corto, lungo-lungo-lungo, corto-corto-corto
- Fai diventare il LED piano piano più luminoso e più debole, come se respirasse

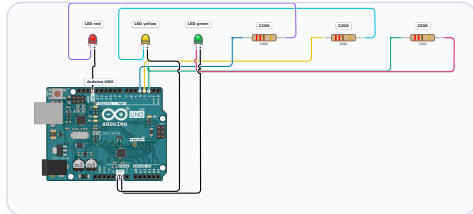
🚀 Costruisci questo progetto con Codey: apri codey.online/it/starter-kits/uno/projects/blink e clicca su FAI QUESTO. Così il codice e lo schema di collegamento sono subito pronti.

★ Principiante ⌚ 15 minuti

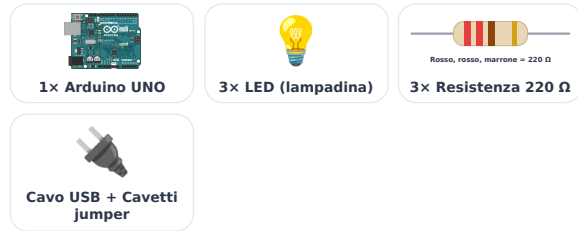
Semaforo

Costruisci un vero semaforo con un LED rosso, uno giallo e uno verde. Si accendono uno alla volta, proprio come in strada.

Schema di collegamento



Che cosa ti serve?



Così lo costruisci

1. Metti i tre LED uno accanto all'altro nella breadboard.
2. Dai a ogni LED la sua resistenza da 220 Ω verso il suo pin: rosso verso 4, giallo verso 3, verde verso 2.
3. Collega tutti i piedini corti alla striscia meno (-) della breadboard.
4. Collega la striscia meno a GND del tuo Arduino.

Il codice

```
// Semaforo
// Tre LED fanno proprio quello che fa un vero semaforo: verde, arancione, rosso.

const int ROSSO = 4;    // LED rosso sul pin 4
const int ARANCIONE = 3; // LED giallo/arancione sul pin 3
const int VERDE = 2;    // LED verde sul pin 2

void setup() {
  pinMode(ROSSO, OUTPUT);
  pinMode(ARANCIONE, OUTPUT);
  pinMode(VERDE, OUTPUT);
}

// Accendi una lucina e spegni le altre due
void accendiLuce(int acceso) {
  digitalWrite(ROSSO, acceso == ROSSO ? HIGH : LOW);
  digitalWrite(ARANCIONE, acceso == ARANCIONE ? HIGH : LOW);
  digitalWrite(VERDE, acceso == VERDE ? HIGH : LOW);
}

void loop() {
  accendiLuce(VERDE);    // Verde: puoi passare
  delay(5000);
  accendiLuce(ARANCIONE); // Arancione: attento, diventa rosso
  delay(2000);
  accendiLuce(ROSSO);    // Rosso: fermati!
  delay(5000);
}
```

Che cosa fa il codice?

- Ogni LED ha il suo pin: rosso su 4, giallo su 3 e verde su 2.
- La funzione `accendiLuce()` accende una lucina e spegne le altre due.
- In `loop()` passiamo da verde ad arancione a rosso, con una pausa in mezzo.

Prova anche...

- Fai lampeggiare la lucina arancione tre volte prima che diventi rossa
- Aggiungi un pulsante: quando lo premi, la luce passa più in fretta al rosso (un pulsante per pedoni)
- Crea una modalità notte in cui lampeggia solo la lucina arancione

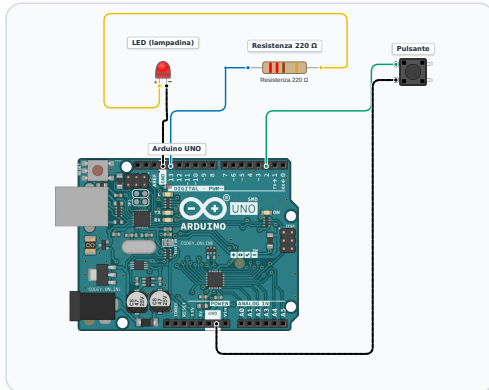
🚀 Costruisci questo progetto con Codey: apri codey.online/it/starter-kits/uno/projects/traffic-light e clicca su FAI QUESTO. Così il codice e lo schema di collegamento sono subito pronti.

★ Principiante ⌚ 15 minuti

Pulsante luce

Crei un pulsante luce. Premi una volta: il LED si accende. Premi ancora: il LED si spegne. Così impari come Arduino "sente" un pulsante.

Schema di collegamento



Che cosa ti serve?



Così lo costruisci

1. Metti il pulsante sopra la fessura della tua breadboard.
2. Collega un piedino al pin 2 e un piedino dall'altro lato a GND.
3. Metti il LED con una resistenza da 220 Ω sul pin 13, con il piedino corto verso GND.

Il codice

```
// Pulsante luce
// Premi una volta il pulsante: il LED si accende. Ancora una volta: il LED si spegne.

const int PIN_PULSANTE = 2; // Il pulsante è sul pin 2 (l'altro lato va a GND)
const int PIN_LED = 13; // Il LED è sul pin 13 (tramite la resistenza)

bool ledAcceso = false; // Ricorda se il LED ora è acceso o spento
bool pulsantePrecedente = HIGH; // Com'era il pulsante la volta prima

void setup() {
  pinMode(PIN_PULSANTE, INPUT_PULLUP); // Ingresso con resistenza integrata: libero = HIGH, premuto = LOW
  pinMode(PIN_LED, OUTPUT);
}

void loop() {
  bool pulsante = digitalRead(PIN_PULSANTE);

  // Reagire solo nel momento in cui premi il pulsante
  if (pulsantePrecedente == HIGH && pulsante == LOW) {
    ledAcceso = !ledAcceso; // Invertire: acceso diventa spento, spento diventa acceso
    digitalWrite(PIN_LED, ledAcceso ? HIGH : LOW);
    delay(50); // Piccola pausa contro il "rimbalzo" del pulsante
  }

  pulsantePrecedente = pulsante;
}
```

Che cosa fa il codice?

- Il pulsante è sul pin 2 con `INPUT_PULLUP`. Libero è HIGH, premuto è LOW.
- Ricordiamo se il LED è acceso o spento in `ledAcceso`.
- Solo nel momento in cui premi il pulsante, invertiamo `ledAcceso`.

Prova anche...

- Fai lampeggiare il LED invece di lasciarlo acceso
- Fai spegnere il LED da solo dopo 10 secondi
- Conta quante volte viene premuto il pulsante e mostralo nel monitor seriale

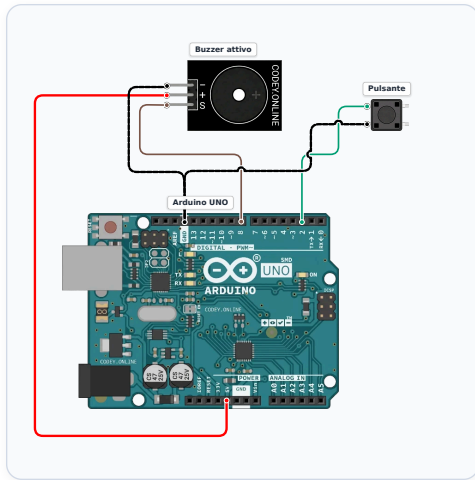
🚀 Costruisci questo progetto con Codey: apri codey.online/it/starter-kits/uno/projects/light-switch e clicca su FAI QUESTO. Così il codice e lo schema di collegamento sono subito pronti.

★ Principiante ⌚ 10 minuti

Campanello

Costruisci il tuo campanello. Premi il pulsante e il cicalino fa "din-don"!

Schema di collegamento



Che cosa ti serve?



Così lo costruisci

1. Metti il pulsante sopra la fessura della breadboard: un piedino va a pin 2, un piedino dall'altro lato va a GND.
2. Collega il cicalino: S a pin 8, + a 5V e - a GND.

Il codice

```
// Campanello
// Premi il pulsante e il cicalino fa un allegro "din-don".

const int PULSANTE_PIN = 2; // Pulsante su pin 2 (altro lato a GND)
const int CICALINO_PIN = 8; // Cicalino attivo: S su pin 8

void setup() {
  pinMode(PULSANTE_PIN, INPUT_PULLUP); // Libero = HIGH, premuto = LOW
  pinMode(CICALINO_PIN, OUTPUT);
}

// Fa suonare il cicalino per un po'
void bip(int quantoTempo) {
  digitalWrite(CICALINO_PIN, HIGH);
  delay(quantoTempo);
  digitalWrite(CICALINO_PIN, LOW);
}

void loop() {
  if (digitalRead(PULSANTE_PIN) == LOW) { // Qualcuno preme il campanello!
    bip(300); // "Din"
    delay(150);
    bip(600); // "Don"
    delay(1000); // Aspetta un attimo, così il campanello non continua a suonare
  }
}
```

Che cosa fa il codice?

- Il pulsante è su pin 2 e il cicalino su pin 8.
- La funzione `bip()` accende il cicalino per un attimo e poi lo spegne.
- Quando il pulsante è premuto, facciamo sentire un bip corto e uno lungo.

Prova anche...

- Suona una melodia allegra di cinque bip
- Fai lampeggiare anche un LED quando qualcuno suona
- Conta quante volte hanno suonato e mostralo nel monitor seriale

🚀 Costruisci questo progetto con Codey: apri codey.online/it/starter-kits/uno/projects/doorbell e clicca su FAI QUESTO. Così il codice e lo schema di collegamento sono subito pronti.

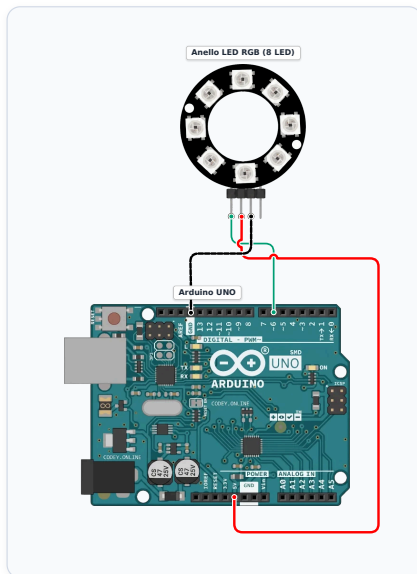


★ Principiante ⌚ 10 minuti

Anello arcobaleno

Con l'anello LED fai un arcobaleno che gira. Ogni lucina ha un colore diverso, e tutti i colori scorrono in cerchio.

Schema di collegamento



Che cosa ti serve?



Così lo costruisci

1. Collega l'anello LED: DI al pin 6, 5V a 5V e GND a GND.
2. È tutto! DO lo lasci libero.

Il codice

```
// Anello arcobaleno
// L'anello LED fa girare in cerchio tutti i colori dell'arcobaleno.

#include <Adafruit_NeoPixel.h>

const int PIN_ANELLO = 6; // DI dell'anello è sul pin 6
const int NUMERO_LED = 8; // L'anello ha 8 lucine
const int VELOCITA = 20; // Più piccolo è, più veloce gira

Adafruit_NeoPixel anello(NUMERO_LED, PIN_ANELLO, NEO_GRB + NEO_KHZ800);

void setup() {
  anello.begin();
  anello.setBrightness(40); // Non troppo forte: 0 = spento, 255 = super forte
  anello.show();
}

void loop() {
  // Sposta sempre un po' l'arcobaleno
  for (long colore = 0; colore < 65536; colore += 512) {
    for (int i = 0; i < NUMERO_LED; i++) {
      // Ogni lucina riceve un colore un po' diverso
      long coloreLed = colore + (i * 65536L / NUMERO_LED);
      anello.setPixelColor(i, anello.gamma32(anello.ColorHSV(coloreLed)));
    }
    anello.show();
    delay(VELOCITA);
  }
}
```

Che cosa fa il codice?

- Usiamo la libreria `Adafruit_NeoPixel` per comandare l'anello.
- `ColorHSV()` crea un colore dell'arcobaleno. Ogni lucina riceve un pezzetto più avanti nell'arcobaleno.
- Spostando sempre un po' il colore, sembra che l'arcobaleno giri in cerchio.

Prova anche...

- Fai girare l'arcobaleno due volte più veloce
- Trasformalo in una luce della polizia che lampeggia rosso e blu a turno
- Fai girare una sola lucina in cerchio, come una lucina che corre

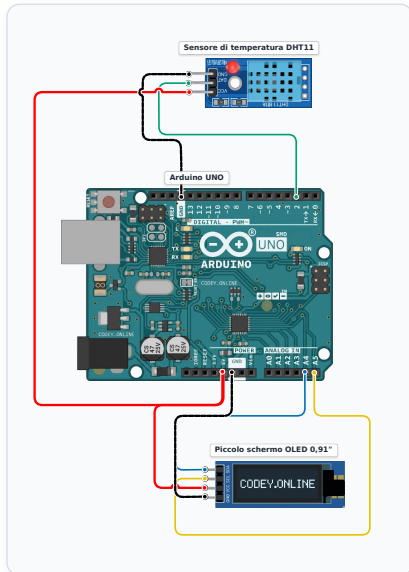
🔧 Costruisci questo progetto con Codey: apri codey.online/it/starter-kits/uno/projects/rainbow-ring e clicca su FAI QUESTO. Così il codice e lo schema di collegamento sono subito pronti.

★★ Principiante avanzato ⌚ 20 minuti

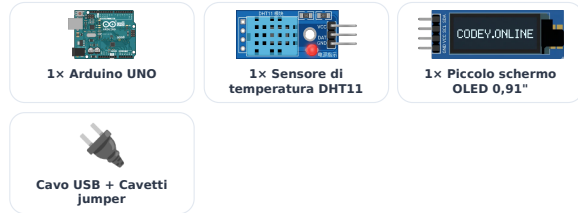
Mini stazione meteo

Costruisci una mini stazione meteo. Il DHT11 misura la temperatura e l'umidità dell'aria. Lo schermo OLED le mostra.

Schema di collegamento



Che cosa ti serve?



Così lo costruisci

1. Collega il DHT11: DAT al pin 2, VCC a 5V e GND a GND.
2. Collega lo schermo OLED: SDA a A4, SCL a A5, VCC a 5V e GND a GND.
3. Consiglio: due pezzi hanno bisogno di 5V e GND. Usa le strisce + e - della tua breadboard.

Il codice

```
// Mini stazione meteo
// Il DHT11 misura la temperatura e l'umidità dell'aria. Lo schermo OLED lo
// mostra.

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <DHT.h>

const int DHT_PIN = 2; // DAT del DHT11 è sul pin 2
DHT dht(DHT_PIN, DHT11);

Adafruit_SSD1306 schermo(128, 32, &Wire, -1); // OLED da 128 x 32 pixel (SDA =
A4, SCL = A5)

void setup() {
  Serial.begin(9600);
  dht.begin();
  if (!schermo.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // 0x3C è l'indirizzo dello
schermo
    Serial.println("Schermo non trovato. Controlla i fili di SDA e SCL.");
    while (true); // Stop: senza schermo non
serve
  }
  schermo.setTextColor(SSD1306_WHITE);
}

void loop() {

float temperatura = dht.readTemperature(); // In gradi Celsius
float umidita = dht.readHumidity(); // In percentuale

schermo.clearDisplay();
schermo.setTextSize(2);
schermo.setCursor(0, 0);

if (isnan(temperatura) || isnan(umidita)) { // La misura non riesce?
  schermo.setTextSize(1);
  schermo.print(F("DHT11 non trovato"));
} else {
  schermo.print(temperatura, 1); // Il simbolo dei gradi
  schermo.print((char)247);
  schermo.println(F("C"));
  schermo.print(umidita, 0);
  schermo.print(F("% umid"));

  Serial.print(F("Temperatura: "));
  Serial.print(temperatura);
  Serial.print(F(" C Umid: "));
  Serial.print(umidita);
  Serial.println(F(" %"));
}

schermo.display(); // Solo ora tutto appare davvero sullo schermo
delay(2000); // Il DHT11 può misurare solo ogni 2 secondi
}
```

Che cosa fa il codice?

- Il DHT11 è sul pin 2. Con `readTemperature()` e `readHumidity()` leggiamo i valori.
- Lo schermo OLED comunica via SDA (A4) e SCL (A5). Con `print()` mettiamo il testo sullo schermo.
- Ogni 2 secondi arriva una nuova misura. La misura non riesce? Allora vedi un messaggio.

Prova anche...

- Mostra anche la temperatura in Fahrenheit
- Fai accendere un LED se fa più caldo di 25 gradi
- Ricorda la temperatura più alta di oggi e mostra anche quella

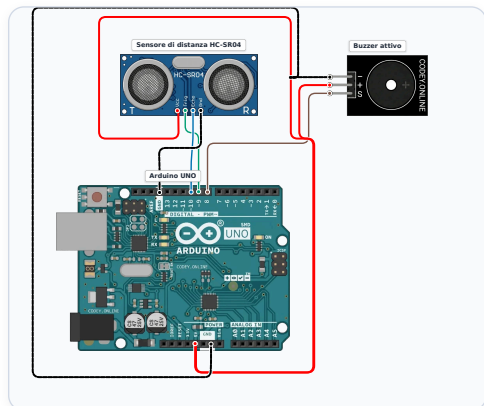
Costruisci questo progetto con Codey: apri codey.online/it/starter-kits/uno/projects/weather-station e clicca su FAI QUESTO. Così il codice e lo schema di collegamento sono subito pronti.

★ ★ Principiante avanzato ⌚ 20 minuti

Sensore di parcheggio

Costruisci un sensore di parcheggio, proprio come in auto. Più ti avvicini al sensore, più il cicalino suona veloce. Sei vicinissimo? Allora senti un tono lungo.

Schema di collegamento



Che cosa ti serve?



Così lo costruisci

1. Collega il HC-SR04: Trig al pin 9, Echo al pin 10, Vcc a 5V e Gnd a GND.
2. Collega il cicalino: S al pin 8, + a 5V e - a GND.
3. Usa le strisce + e - della tua breadboard per 5V e GND.

Il codice

```
// Sensore di parcheggio
// Più ti avvicini al sensore, più il cicalino suona veloce.
// Proprio come in auto!

const int TRIG_PIN = 9;    // Trig del HC-SR04 sul pin 9
const int ECHO_PIN = 10;   // Echo del HC-SR04 sul pin 10
const int CICALINO_PIN = 8; // S del cicalino sul pin 8

void setup() {
  Serial.begin(9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(CICALINO_PIN, OUTPUT);
}

// Misura la distanza in centimetri
long misuraDistanza() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);    // Manda un suono breve (che tu
  // non senti)
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  long tempo = pulseIn(ECHO_PIN, HIGH, 30000); // Quanto tempo
  // serve perché l'eco torni?
  if (tempo == 0) return 999;          // Nessuna eco:

  niente vicino
  return tempo / 58;                  // Converti il
  // tempo in centimetri
}

void loop() {
  long distanza = misuraDistanza();
  Serial.print(F("Distanza: "));
  Serial.print(distanza);
  Serial.println(F(" cm"));

  if (distanza < 5) {
    digitalWrite(CICALINO_PIN, HIGH); // Molto vicino: un tono
    // lungo
    delay(100);
  } else if (distanza < 50) {
    digitalWrite(CICALINO_PIN, HIGH); // Bip breve
    delay(50);
    digitalWrite(CICALINO_PIN, LOW);
    delay(distanza * 10);              // Più vicino = pausa più
    // corta = bip più rapidi
  } else {
    digitalWrite(CICALINO_PIN, LOW);  // Lontano: silenzio
    delay(100);
  }
}
```

Che cosa fa il codice?

- `misuraDistanza()` manda un suono e misura quanto tempo ci mette l'eco a tornare. Poi lo trasformiamo in centimetri.
- Oltre 50 cm: il cicalino sta zitto.
- Più vicino: la pausa tra i bip diventa più corta. Sotto 5 cm suona senza fermarsi.

Prova anche...

- Fai iniziare il sensore di parcheggio a suonare già da 1 metro
- Aggiungi un LED rosso che si accende quando è troppo vicino
- Mostra la distanza nel monitor seriale, con una barretta di asterischi

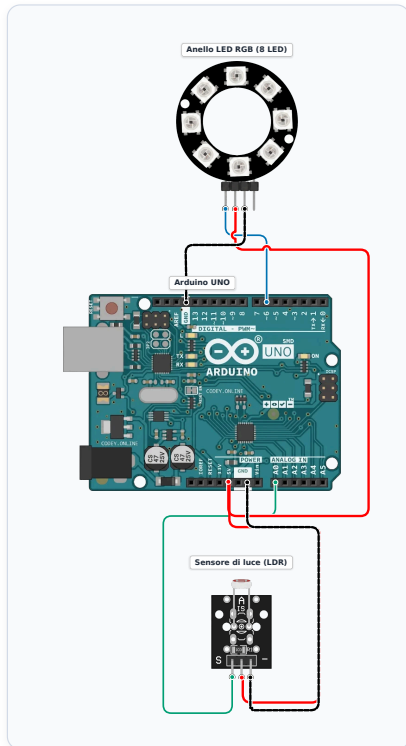
🚀 Costruisci questo progetto con Codey: apri codey.online/it/starter-kits/uno/projects/parking-sensor e clicca su FAI QUESTO. Così il codice e lo schema di collegamento sono subito pronti.

★★ Principiante avanzato ⏱ 15 minuti

Lucina notturna

Costruisci una lucina notturna intelligente. Quando diventa buio, l'anello LED si accende piano. Quando torna la luce, si spegne di nuovo.

Schema di collegamento



Che cosa ti serve?



Così lo costruisci

1. Collega il sensore di luce: S a A0, + a 5V e - a GND.
2. Collega l'anello LED: DI al pin 6, 5V a 5V e GND a GND.
3. Apri il monitor seriale e guarda che numero vedi con la luce e al buio. Poi adatta [SOGLIA_BUIO](#).

Il codice

```
// Lucina notturna
// Diventa buio? Allora l'anello LED si accende piano. Diventa chiaro? Allora si
// spegne.

#include <Adafruit_NeoPixel.h>

const int PIN_LUCE = A0; // S del sensore di luce su A0
const int PIN_ANELLO = 6; // DI dell'anello LED sul pin 6
const int NUMERO_LED = 8;

// A questo valore diciamo che e' "buio". Guarda nel monitor seriale quale
// numero vedi
// con la luce e al buio, e metti qui un numero a meta.
const int SOGLIA_BUIO = 500;

Adafruit_NeoPixel anello(NUMERO_LED, PIN_ANELLO, NEO_GRB + NEO_KHZ800);

void setup() {
  Serial.begin(9600);
  anello.begin();
  anello.setBrightness(60);
  anello.show();
}

void loop() {
  int luce = analogRead(PIN_LUCE); // Un numero da 0 a 1023
  Serial.print(F("Sensore luce: "));
  Serial.println(luce);

  // Con questo sensore il numero sale quando diventa piu buio.
  // Da te funziona al contrario? Allora cambia > in <.
  bool buio = luce > SOGLIA_BUIO;

  for (int i = 0; i < NUMERO_LED; i++) {
    if (buio) {
      anello.setPixelColor(i, anello.Color(255, 140, 40)); // Luce gialla calda
    } else {
      anello.setPixelColor(i, 0); // Spento
    }
  }
  anello.show();
  delay(200);
}
```

Che cosa fa il codice?

- Il sensore di luce è su A0. `analogRead()` dà un numero da 0 a 1023.
- Il numero è più alto di `SOGLIA_BUIO`? Allora per noi è buio.
- Al buio diamo a tutte le lucine un colore giallo caldo. Altrimenti si spengono.

Prova anche...

- Fai diventare la lucina sempre più forte man mano che diventa più buio
- Fai cambiare piano il colore, come una lampada rilassante
- Fai spegnere la lucina da sola dopo 1 minuto

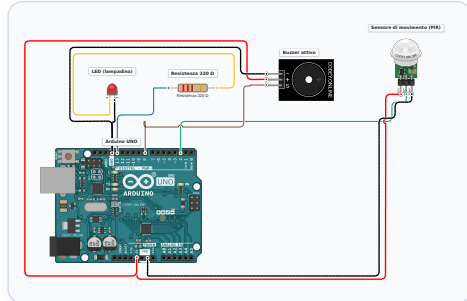
🔧 Costruisci questo progetto con Codey: apri codey.online/it/starter-kits/uno/projects/night-light e clicca su FAI QUESTO. Così il codice e lo schema di collegamento sono subito pronti.

★ ★ Principiante avanzato ⌚ 20 minuti

Allarme di movimento

Costruisci un allarme antifurto. Il sensore di movimento vede qualcuno? Allora il buzzer suona e il LED rosso lampeggia.

Schema di collegamento



Che cosa ti serve?



Così lo costruisci

1. Collega il sensore PIR: OUT al pin 2, VCC a 5V e GND a GND.
2. Collega il buzzer: S al pin 8, + a 5V e - a GND.
3. Metti il LED rosso con una resistenza da 220 Ω sul pin 13, la gambetta corta verso GND.

Il codice

```
// Allarme di movimento
// Passa qualcuno? Allora il LED rosso lampeggia e il buzzer suona.

const int PIR_PIN = 2; // OUT del sensore di movimento sul pin 2
const int BUZZER_PIN = 8; // S del buzzer sul pin 8
const int LED_PIN = 13; // LED rosso sul pin 13 (con la resistenza)

void setup() {
  Serial.begin(9600);
  pinMode(PIR_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(LED_PIN, OUTPUT);
  Serial.println(F("Il sensore si scalda... stai fermo!"));
  delay(10000); // Il sensore ha bisogno di un attimo per abituarsi alla stanza
  Serial.println(F("Allarme acceso."));
}

void loop() {
  if (digitalRead(PIR_PIN) == HIGH) { // Movimento visto!
    Serial.println(F("Movimento! ALLARME!"));
    for (int i = 0; i < 5; i++) { // 5 volte: suona e lampeggia
      digitalWrite(LED_PIN, HIGH);
      digitalWrite(BUZZER_PIN, HIGH);
      delay(150);
      digitalWrite(LED_PIN, LOW);
      digitalWrite(BUZZER_PIN, LOW);
      delay(150);
    }
  }
  delay(100);
}
```

Che cosa fa il codice?

- Il sensore PIR è sul pin 2. Se vede movimento, il pin diventa HIGH.
- All'avvio aspettiamo 10 secondi. Così il sensore può abituarsi alla stanza.
- Quando c'è movimento, facciamo suonare e lampeggiare 5 volte.

Prova anche...

- Aggiungi un pulsante per accendere e spegnere l'allarme
- Conta quante volte c'è stato movimento e mostralo nel monitor seriale
- Fai suonare l'allarme più a lungo e più forte al secondo movimento

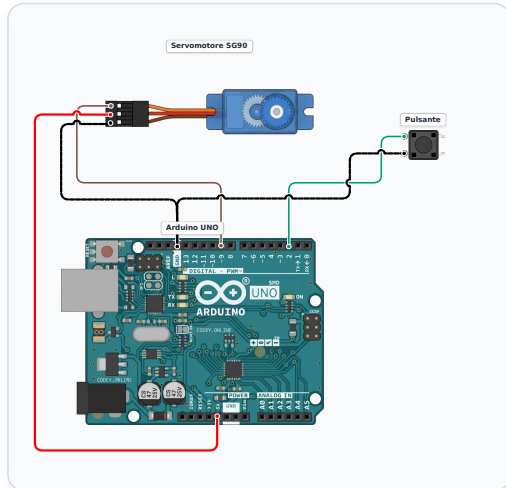
🚀 Costruisci questo progetto con Codey: apri codey.online/it/starter-kits/uno/projects/motion-alarm e clicca su FAI QUESTO. Così il codice e lo schema di collegamento sono subito pronti.

☆☆ Principiante avanzato ⌚ 20 minuti

Sbarra

Costruisci una sbarra, come in un parcheggio. Premi il pulsante: il servo alza la sbarra, aspetta un po' e poi si richiude.

Schema di collegamento



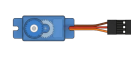
Che cosa ti serve?



1x Arduino UNO



1x Pulsante



1x Servomotore SG90



Cavo USB + Cavetti jumper

Così lo costruisci

1. Metti il pulsante sopra la fessura della breadboard: un piedino va al pin 2, un piedino dall'altro lato va a GND.
2. Collega il servo: arancione al pin 9, rosso a 5V e marrone a GND.
3. Incolla una cannuccia o un bastoncino da gelato sul braccetto del servo: quella è la tua sbarra!

Il codice

```
// Sbarra
// Premi il pulsante: la sbarra (servo) si alza, aspetta un po'
// e si richiude.

#include <Servo.h>

const int PULSANTE_PIN = 2;    // Pulsante su pin 2 (altro
                                // lato verso GND)
const int SERVO_PIN = 9;      // Filo arancione/giallo del servo
                                // su pin 9

const int CHIUSO = 0;         // Angolo del servo quando la
                                // sbarra è chiusa
const int APERTO = 90;        // Angolo del servo quando la
                                // sbarra è aperta

Servo sbarra;

void setup() {
  pinMode(PULSANTE_PIN, INPUT_PULLUP);
```

```
sbarra.attach(SERVO_PIN);
sbarra.write(CHIUSO);    // Inizia con la sbarra chiusa
}

void loop() {
  if (digitalRead(PULSANTE_PIN) == LOW) { // Pulsante premuto
    // Ruota piano in apertura, grado per grado
    for (int angolo = CHIUSO; angolo <= APERTO; angolo++) {
      sbarra.write(angolo);
      delay(15);
    }
    delay(3000); // 3 secondi aperta:
    // l'auto passa
    // Piano si richiude
    for (int angolo = APERTO; angolo >= CHIUSO; angolo--) {
      sbarra.write(angolo);
      delay(15);
    }
  }
}
```

Che cosa fa il codice?

- Usiamo la libreria `Servo`. Con `write()` dici al servo a quale angolo andare.
- Dopo aver premuto il pulsante, giriamo grado per grado da 0 a 90. Così va con calma.
- Dopo 3 secondi torniamo a 0: la sbarra è chiusa.

Prova anche...

- Fai aprire e chiudere la sbarra più in fretta
- Fai suonare un buzzer mentre la sbarra si muove
- Accendi un LED rosso quando la sbarra è chiusa e uno verde quando è aperta

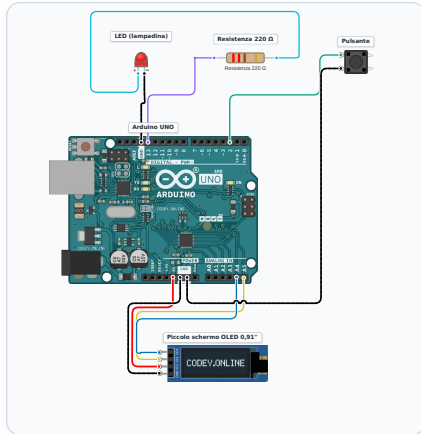
🚀 Costruisci questo progetto con Codey: apri codey.online/it/starter-kits/uno/projects/barrier e clicca su FAI QUESTO. Così il codice e lo schema di collegamento sono subito pronti.

★★ Principiante avanzato ⌚ 25 minuti

Gioco dei riflessi

Quanto sei veloce? Aspetta che il LED si accenda e poi premi il pulsante il più in fretta possibile. Lo schermo mostra il tuo tempo in millisecondi. Sfida i tuoi amici!

Schema di collegamento



Che cosa ti serve?



Così lo costruisci

1. Metti il pulsante sopra la fessura della breadboard: un piedino va al pin 2, un piedino dall'altro lato va a GND.
2. Metti il LED con una resistenza da 220 Ohm sul pin 13, con il piedino corto verso GND.
3. Collega lo schermo OLED: SDA ad A4, SCL ad A5, VCC a 5V e GND a GND.

Il codice

```
// Gioco dei riflessi
// Aspetta che il LED si accenda e poi premi il pulsante il più in fretta
// possibile.
// Lo schermo mostra quanto sei stato veloce!

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

const int PIN_PULSANTE = 2; // Pulsante sul pin 2 (altro lato a GND)
const int PIN_LED = 13; // LED sul pin 13 (con la resistenza)

Adafruit_SSD1306 schermo(128, 32, &Wire, -1); // OLED: SDA = A4, SCL = A5

unsigned long migliorTempo = 0; // Il tuo tempo più veloce finora

// Mostra un testo grande sullo schermo
void mostra(const __FlashStringHelper* sopra, unsigned long numero) {
  schermo.clearDisplay();
  schermo.setTextSize(1);
  schermo.setCursor(0, 0);
  schermo.print(sopra);
  if (numero > 0) {
    schermo.setTextSize(2);
    schermo.setCursor(0, 14);
    schermo.print(numero);
    schermo.print(F(" ms"));
  }
  schermo.display();
}

void setup() {
  pinMode(PIN_PULSANTE, INPUT_PULLUP);
  pinMode(PIN_LED, OUTPUT);
  randomSeed(analogRead(A0)); // Crea attese davvero casuali

  schermo.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  schermo.setTextColor(SSD1306_WHITE);
}

void loop() {
  mostra(F("Premi il tasto..."), 0);
  while (digitalRead(PIN_PULSANTE) == HIGH); // Aspetta finché premi il
  pulsante
  while (digitalRead(PIN_PULSANTE) == LOW); // ...e poi lo lasci
  mostra(F("Pronto? Occhio LED!"), 0);

  delay(random(2000, 5000)); // Un'attesa segreta tra 2 e 5
  secondi

  if (digitalRead(PIN_PULSANTE) == LOW) { // Premuto troppo presto?
    Furbetto!
    mostra(F("Presto! Riprova."), 0);
    delay(2000);
    return;
  }

  digitalWrite(PIN_LED, HIGH); // ORA!
  unsigned long inizio = millis();
  while (digitalRead(PIN_PULSANTE) == HIGH); // Aspetta la tua pressione
  unsigned long tempo = millis() - inizio;
  digitalWrite(PIN_LED, LOW);

  if (migliorTempo == 0 || tempo < migliorTempo) {
    migliorTempo = tempo;
    mostra(F("NUOVO RECORD!"), tempo);
  } else {
    mostra(F("Tuo tempo:"), tempo);
  }
  delay(3000);
}
```

Che cosa fa il codice?

- Dopo aver premuto il pulsante, aspettiamo un tempo segreto tra 2 e 5 secondi (`random()`).
- Poi il LED si accende e facciamo partire un cronometro con `millis()`.
- Quando premi, calcoliamo quanti millisecondi sono passati. Ricordiamo il tuo tempo migliore.

Prova anche...

- Trasformalo in un gioco per due giocatori, ognuno con il suo pulsante
- Fai suonare un buzzer quando c'è un nuovo record
- Gioca 5 turni e poi mostra il tuo tempo medio

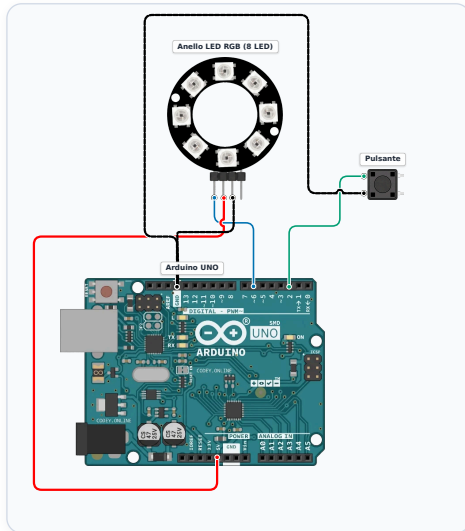
🔧 Costruisci questo progetto con Codey: apri codey.online/it/starter-kits/uno/projects/reaction-game e clicca su FAI QUESTO. Così il codice e lo schema di collegamento sono subito pronti.

☆☆ Principiante avanzato ⌚ 15 minuti

Dado elettronico

Costruisci un dado di luce. Premi il pulsante: le lucine girano un po', e poi vedi quanto hai lanciato. Da 1 a 6 lucine!

Schema di collegamento



Che cosa ti serve?



Così lo costruisci

1. Metti il pulsante sopra la fessura della breadboard: una zampetta va al pin 2, una zampetta dall'altro lato va a GND.
2. Collega l'anello LED: DI al pin 6, 5V a 5V e GND a GND.

Il codice

```
// Dado elettronico
// Premi il pulsante: l'anello LED gira un attimo e poi lascia accese
// da 1 a 6 lucine.

#include <Adafruit_NeoPixel.h>

const int PULSANTE_PIN = 2; // Pulsante sul pin 2 (altro lato
// verso GND)
const int ANELLO_PIN = 6; // DI dell'anello LED sul pin 6
const int NUMERO_LED = 8;

Adafruit_NeoPixel anello(NUMERO_LED, ANELLO_PIN, NEO_GRB +
NEO_KHZ800);

// Accende alcune lucine in un solo colore
void mostraNumero(int numero, uint32_t colore) {
  anello.clear();
  for (int i = 0; i < numero; i++) {
    anello.setPixelColor(i, colore);
  }
  anello.show();
}

void setup() {
  pinMode(PULSANTE_PIN, INPUT_PULLUP);
  randomSeed(analogRead(A0)); // Fa sì che il dado sia davvero
casuale
  anello.begin();
  anello.setBrightness(50);
  anello.show();
}

void loop() {
  if (digitalRead(PULSANTE_PIN) == LOW) {
    // Un po' di "giro": numeri che cambiano in fretta, sempre un po'
più piano
    for (int i = 0; i < 15; i++) {
      mostraNumero(random(1, 7), anello.Color(0, 80, 255));
      delay(40 + i * 10);
    }
    int lancio = random(1, 7); // Un numero da 1 a
6
    mostraNumero(lancio, anello.Color(0, 255, 60)); // Il numero
vero in verde
    delay(500);
  }
}
```

Che cosa fa il codice?

- `random(1, 7)` sceglie un numero a caso da 1 a 6.
- Prima mostriamo numeri che cambiano in fretta, in blu. Così sembra che il dado stia girando.
- Poi `mostraNumero()` mostra il numero vero in verde.

Prova anche...

- Dai a ogni numero il suo colore
- Fai fare un bip breve a un buzzer quando il dado si ferma
- Trasformalo in un dado da 1 a 8

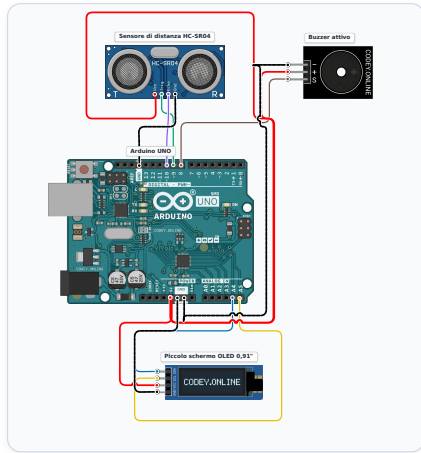
🚀 Costruisci questo progetto con Codey: apri codey.online/it/starter-kits/uno/projects/dice e clicca su FAI QUESTO. Così il codice e lo schema di collegamento sono subito pronti.

★★★ Sfida ⌚ 30 minuti

Sensore di parcheggio deluxe

La versione deluxe del sensore di parcheggio! Fa bip più in fretta quando ti avvicini, e il display OLED mostra la distanza in centimetri. Con una barretta che diventa più lunga quando ti avvicini.

Schema di collegamento



Che cosa ti serve?



Così lo costruisci

1. Collega il HC-SR04: Trig al pin 9, Echo al pin 10, Vcc a 5V e Gnd a GND.
2. Collega il cicalino: S al pin 8, + a 5V e - a GND.
3. Collega il display OLED: SDA a A4, SCL a A5, VCC a 5V e GND a GND.
4. Tre componenti hanno bisogno di 5V e GND. Usa le strisce + e - della tua breadboard.

Il codice

```
// Sensore di parcheggio deluxe
// Come il sensore di parcheggio normale, ma ora vedi la distanza anche sul display OLED.

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

const int TRIG_PIN = 9;    // Trig del HC-SR04 sul pin 9
const int ECHO_PIN = 10;   // Echo del HC-SR04 sul pin 10
const int CICALINO_PIN = 8; // S del cicalino sul pin 8

Adafruit_SSD1306 display(128, 32, &Wire, -1); // OLED: SDA = A4, SCL = A5

// Misura la distanza in centimetri
long misuraDistanza() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long tempo = pulseIn(ECHO_PIN, HIGH, 30000);
  if (tempo == 0) return 999; // Nessun eco: niente vicino
  return tempo / 58;
}

// Mostra la distanza grande sul display, con una barretta sotto
void mostraDistanza(long distanza) {
  display.clearDisplay();
  display.setTextSize(2);
  display.setCursor(0, 0);
  if (distanza >= 999) {
    display.print(F("..."));
  } else {
    display.print(distanza);
    display.print(F(" cm"));
  }
}

// Barretta: più vicino, più lunga
int lunghezza = map(constrain(distanza, 0, 100), 0, 100, 128, 0);
display.fillRect(0, 24, lunghezza, 8, SSD1306_WHITE);
display.display();
}

void setup() {
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(CICALINO_PIN, OUTPUT);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.setTextColor(SSD1306_WHITE);
}

void loop() {
  long distanza = misuraDistanza();
  mostraDistanza(distanza);

  if (distanza < 5) {
    digitalWrite(CICALINO_PIN, HIGH); // Molto vicino: un tono lungo
    delay(100);
  } else if (distanza < 50) {
    digitalWrite(CICALINO_PIN, HIGH); // Bip breve
    delay(50);
    digitalWrite(CICALINO_PIN, LOW);
    delay(distanza * 10); // Più vicino = bip più veloci
  } else {
    digitalWrite(CICALINO_PIN, LOW);
    delay(100);
  }
}
```

Che cosa fa il codice?

- `misuraDistanza()` misura la distanza in centimetri con il HC-SR04.
- `mostraDistanza()` mette il numero grande sul display e disegna una barretta con `fillRect()`.
- Il cicalino fa bip più in fretta quando la distanza diventa più piccola.

Prova anche...

- Mostra "STOP!" sul display se sei a meno di 5 cm
- Mostra la distanza in metri se qualcosa è a più di 1 metro
- Aggiungi l'anello LED che cambia colore da verde a rosso

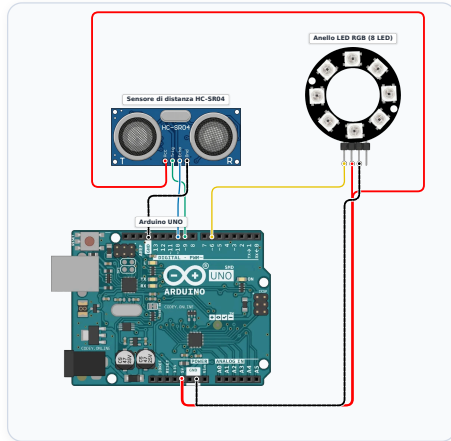
Costruisci questo progetto con Codey: codey.online/it/starter-kits/uno/projects/parking-sensor-deluxe e clicca su FAI QUESTO. Così il codice e lo schema di collegamento sono subito pronti.

★★★ Sfida ⌚ 25 minuti

Misuratore di distanza con barra luminosa

Costruisci una barra luminosa che mostra quanto è vicina una cosa. Lontano: nessuna lucina.
Più è vicina, più lucine si accendono. Prima verde, poi arancione e molto vicino rosso.

Schema di collegamento



Che cosa ti serve?



Così lo costruisci

1. Collega il HC-SR04: Trig al pin 9, Echo al pin 10, Vcc a 5V e Gnd a GND.
2. Collega l'anello LED: DI al pin 6, 5V a 5V e GND a GND.

Il codice

```
// Misuratore di distanza con barra luminosa
// Più ti avvicini, più lucine si accendono: verde, poi arancione, poi rosso.

#include <Adafruit_NeoPixel.h>

const int TRIG_PIN = 9;    // Trig del HC-SR04 sul pin 9
const int ECHO_PIN = 10;   // Echo del HC-SR04 sul pin 10
const int RING_PIN = 6;    // DI dell'anello LED sul pin 6
const int NUMERO_LED = 8;
const int DISTANZA_MAX = 80; // Oltre questo (in cm): tutte le lucine spente

Adafruit_NeoPixel anello(NUMERO_LED, RING_PIN, NEO_GRB + NEO_KHZ800);

long misuraDistanza() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long tempo = pulseIn(ECHO_PIN, HIGH, 30000);
  if (tempo == 0) return 999;
  return tempo / 58;
}

void setup() {
  Serial.begin(9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  anello.begin();
}
```

```
anello.setBrightness(40);
anello.show();
}

void loop() {
  long distanza = misuraDistanza();
  Serial.print(F("Distanza: "));
  Serial.print(distanza);
  Serial.println(F(" cm"));

  // Quante lucine devono accendersi? Vicino = 8, lontano = 0
  int numero = map(constrain(distanza, 0, DISTANZA_MAX), 0, DISTANZA_MAX, NUMERO_LED, 0);

  for (int i = 0; i < NUMERO_LED; i++) {
    if (i < numero) {
      if (i < 3) anello.setPixelColor(i, anello.Color(0, 255, 0));
      // Verde
      else if (i < 6) anello.setPixelColor(i, anello.Color(255, 120, 0));
      // Arancione
      else anello.setPixelColor(i, anello.Color(255, 0, 0));
      // Rosso
    } else {
      anello.setPixelColor(i, 0);
      // Spento
    }
  }
  anello.show();
  delay(100);
}
```

Che cosa fa il codice?

- `misuraDistanza()` misura la distanza con il HC-SR04.
- `map()` trasforma la distanza in un numero di lucine: da 0 a 8.
- Le prime 3 lucine sono verdi, le 3 dopo arancioni e le ultime 2 rosse.

Prova anche...

- Fai lampeggiare tutte le lucine in rosso se sei più vicino di 10 cm
- Aggiungi il buzzer che fa beep quando è rosso
- Rendi il misuratore più sensibile: tutte le lucine accese a 40 cm invece di 80 cm

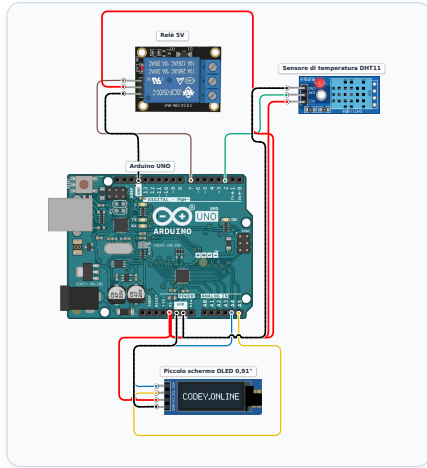
🚀 Costruisci questo progetto con Codey: apri codey.online/it/starter-kits/uno/projects/distance-light-bar e clicca su FAI QUESTO. Così il codice e lo schema di collegamento sono subito pronti.

★★★ Sfida ⌚ 30 minuti

Ventilatore intelligente

Costruisci un termostato. Se fa più caldo di 25 gradi, il relè fa clic e si accende. Così il tuo Arduino può accendere, per esempio, una piccola ventola con una batteria. Sullo schermo vedi la temperatura e se la ventola è accesa.

Schema di collegamento



Che cosa ti serve?



Così lo costruisci

1. Collega il DHT11: DAT verso pin 2, VCC verso 5V e GND verso GND.
2. Collega il relè: S verso pin 7, + verso 5V e - verso GND.
3. Collega lo schermo OLED: SDA verso A4, SCL verso A5, VCC verso 5V e GND verso GND.
4. Vuoi davvero accendere qualcosa? Usa solo una batteria o bassa tensione su COM e NO, mai la presa di corrente.

Il codice

```
// Ventilatore intelligente (termostato)
// Fa troppo caldo? Allora il relè si attiva, per esempio per una ventola.
// Sullo schermo vedi la temperatura e se la ventola è accesa.

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <DHT.h>

const int DHT_PIN = 2;      // DAT del DHT11 sul pin 2
const int RELE_PIN = 7;     // S del relè sul pin 7
const float ACCESO_SOPRA = 25.0; // Sopra questa temperatura la ventola si accende
const float SPENTO_SOTTO = 24.0; // Sotto questa temperatura si spegne di nuovo

DHT dht(DHT_PIN, DHT11);
Adafruit_SSD1306 schermo(128, 32, &Wire, -1); // OLED: SDA = A4, SCL = A5

bool ventilatoreAcceso = false;

void setup() {
  Serial.begin(9600);
  pinMode(RELE_PIN, OUTPUT);
  digitalWrite(RELE_PIN, LOW); // Inizia con il relè spento
  dht.begin();
  schermo.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  schermo.setTextColor(SSD1306_WHITE);
}

void loop() {
  float temperatura = dht.readTemperature();
```

```
if (!isnan(temperatura)) {
  // Due limiti, così il relè non fa clic di continuo
  if (temperatura > ACCESO_SOPRA) ventilatoreAcceso = true;
  if (temperatura < SPENTO_SOTTO) ventilatoreAcceso = false;
  digitalWrite(RELE_PIN, ventilatoreAcceso ? HIGH : LOW);
}

schermo.clearDisplay();
schermo.setTextSize(2);
schermo.setCursor(0, 0);
if (isnan(temperatura)) {
  schermo.print(F("DHT11 ?"));
} else {
  schermo.print(temperatura, 1);
  schermo.print((char)247);
  schermo.print(F("C"));
}
schermo.setTextSize(1);
schermo.setCursor(0, 22);
schermo.print(ventilatoreAcceso ? F("Ventola: accesa") : F("Ventola:
spenta"));
schermo.display();

Serial.print(F("Temperatura: "));
Serial.print(temperatura);
Serial.println(ventilatoreAcceso ? F(" C -> accesa") : F(" C ->
spenta"));

delay(2000);
}
```

Che cosa fa il codice?

- Il DHT11 misura la temperatura. Sopra i 25 gradi accendiamo il relè.
- Solo sotto i 24 gradi il relè si spegne di nuovo. Con questi due limiti non continua ad accendersi e spegnersi.
- Lo schermo OLED mostra la temperatura e "Ventola: accesa" o "spenta".

Prova anche...

- Aggiungi due pulsanti per alzare e abbassare la temperatura limite
- Fai accendere un LED finché la ventola è accesa
- Fai partire la ventola anche quando l'umidità dell'aria supera il 70%

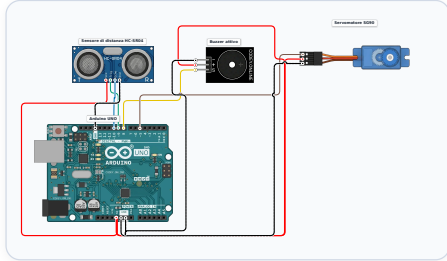
🚀 Costruisci questo progetto con Codey: apri codey.online/it/starter-kits/uno/projects/thermostat e clicca su FAI QUESTO. Così il codice e lo schema di collegamento sono subito pronti.

★★★★ Sfida ⌚ 30 minuti

Sbarra automatica

La versione furba della sbarra! Il sensore di distanza vede che arriva un'auto. Poi il cicalino fa beep e la sbarra si apre da sola. L'auto è andata via? Allora si richiude.

Schema di collegamento



Che cosa ti serve?



Così lo costruisci

1. Collega il HC-SR04: Trig al pin 9, Echo al pin 10, Vcc a 5V e Gnd a GND.
2. Collega il servo: arancione al pin 5, rosso a 5V e marrone a GND.
3. Collega il cicalino: S al pin 8, + a 5V e - a GND.
4. Incolla una cannuccia o uno stecchino del gelato sul servo come sbarra e metti il sensore davanti.

Il codice

```
// Sbarra automatica
// Arriva un'auto (qualcosa a meno di 15 cm)? Allora il cicalino fa beep
// e la sbarra si apre.

#include <Servo.h>

const int TRIG_PIN = 9;    // Trig del HC-SR04 sul pin 9
const int ECHO_PIN = 10;   // Echo del HC-SR04 sul pin 10
const int SERVO_PIN = 5;   // Filo arancione/giallo del servo sul pin 5
const int CICALINO_PIN = 8; // S del cicalino sul pin 8
const int VICINO = 15;     // Da questa distanza (cm) la sbarra si apre

const int CHIUSO = 0;      // Angolo quando la sbarra è chiusa
const int APERTO = 90;     // Angolo quando la sbarra è aperta

Servo sbarra;

long misuraDistanza() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long tempo = pulseIn(ECHO_PIN, HIGH, 30000);
  if (tempo == 0) return 999;
  return tempo / 58;
}

void setup() {
  Serial.begin(9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(CICALINO_PIN, OUTPUT);
  sbarra.attach(SERVO_PIN);
  sbarra.write(CHIUSO);
}

void loop() {
  long distanza = misuraDistanza();

  if (distanza < VICINO) {
    Serial.println(F("Auto vista! Sbarra aperta."));
    digitalWrite(CICALINO_PIN, HIGH); // Beep breve
    delay(100);
    digitalWrite(CICALINO_PIN, LOW);

    sbarra.write(APERTO);
    delay(1000);
    // Resta aperta finché l'auto è ancora lì
    while (misuraDistanza() < VICINO) {
      delay(200);
    }
    delay(2000); // Aspetta ancora un po' finché
    // l'auto è davvero via
    sbarra.write(CHIUSO);
    Serial.println(F("Sbarra chiusa."));
  }
  delay(100);
}
```

Che cosa fa il codice?

- Il HC-SR04 misura sempre la distanza. Meno di 15 cm? Allora c'è un'auto.
- Il cicalino fa un breve beep e il servo gira a 90 gradi: aperto.
- Finché l'auto è lì, la sbarra resta aperta. Poi aspettiamo 2 secondi e si chiude.

Prova anche...

- Conta quante auto sono passate e mostralo nel monitor seriale
- Aggiungi un LED rosso e uno verde come semaforo vicino alla sbarra
- Fai aprire la sbarra solo se viene premuto anche un pulsante

Costruisci questo progetto con Codey: [apri codey.online/it/starter-kits/uno/projects/automatic-barrier](https://codey.online/it/starter-kits/uno/projects/automatic-barrier) e clicca su FAI QUESTO. Così il codice e lo schema di collegamento sono subito pronti.