

Grazie per aver scelto il Beginner's Lab Kit

Benvenuto nel Kit di Laboratorio per Principianti, un pacchetto completo progettato specificamente per i nuovi arrivati nel mondo dell'elettronica e della programmazione. Questo kit include una serie di componenti essenziali come LED, resistori, un buzzer, potenziometri, fotoresistori, termistori, pulsanti e tubi digitali, oltre a un modulo ultrasonico. Una delle caratteristiche distintive di questo kit è l'inclusione di un multimetro, uno strumento prezioso che ti consente di misurare corrente, tensione e resistenza nei tuoi circuiti. Questa aggiunta è particolarmente utile per approfondire la tua comprensione del funzionamento di ciascun componente.

La sequenza del corso fornita con questo kit è strutturata attorno alla sintassi di programmazione di Arduino, garantendo una progressione logica ed educativa. Questa struttura ti permette di costruire circuiti passo dopo passo mentre impari a scrivere i programmi che li controllano. Durante il corso, ti troverai ad affrontare sfide di risoluzione dei problemi che arricchiranno la tua comprensione del materiale.

Per qualsiasi domanda o supporto, contattaci a service@sunfounder.com. Inizia il tuo percorso di apprendimento con il Kit di Laboratorio per Principianti e inizia a costruire, programmare ed esplorare il mondo emozionante dell'elettronica!

Contenuti

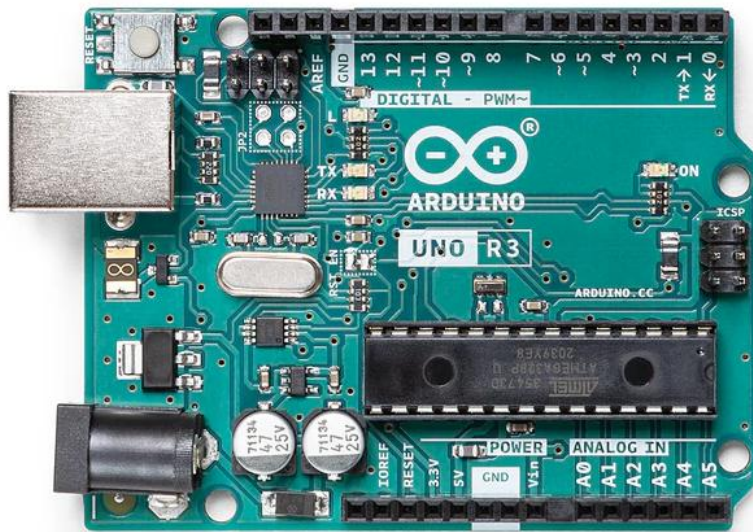
Cosa è incluso nel tuo kit.....	1
Lezione 2: Il Tuo Primo Circuito	8
Lezione 3: Misurare con il Multimetro	9
Lezione 4: La Legge di Ohm	10
Lezione 5: Circuito in Serie vs. Circuito in Parallelo	11
Lezione 6: LED lampeggiante	15
Lezione 7: Costruiamo i Semafori!	16
Lezione 8: Semaforo con Pulsante per Pedoni	17
Lezione 9: Lampada da Scrivania Dimmerabile	19
Lezione 10: Lampada da Scrivania ON/OFF	20
Lezione 11: Controllo delle Matrici di LED con Potenzimetro	21
Lezione 12: I Colori dell'Arcobaleno	22
Lezione 13: Lo Spettro della Visione	23
Lezione 14 : Colori Casuali	24
Lezione 15 : Colori Caldi o Freddi	25
Lezione 16 : Allarme Temperatura	26
Lezione 17 : Codice Morse	27
Lezione 18: Allarme Luminoso	28
Lezione 19: Sistema di Allarme per il Parcheggio in Retromarcia	29
Lezione 20: Il Timer Pomodoro	30
Lezione 21: Suono della Sirena	32
Lezione 22: Suonare "Twinkle, Twinkle, Little Star"	34
Lezione 23: Cyber Dice	35
Lezione 24: Luce Scorrevole con 74HC595	37
Lezione 25: Visualizzare un Numero	38

Cosa è incluso nel tuo kit

All'interno del nostro kit troverai una varietà di componenti e pezzi che utilizzerai durante questo corso per costruire circuiti. Ecco una guida rapida a ciò che è incluso.

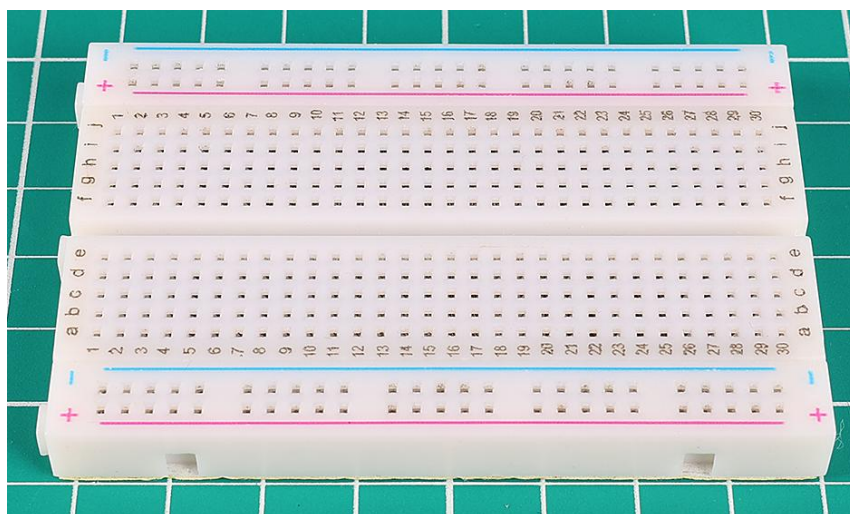
1 x Original Arduino Uno R3

Una scheda microcontrollore che funge da cervello per i tuoi circuiti. È dotata di tutto il necessario per supportare il microcontrollore; basta collegarla al computer tramite un cavo USB o alimentarla con un adattatore AC-DC o una batteria per iniziare.



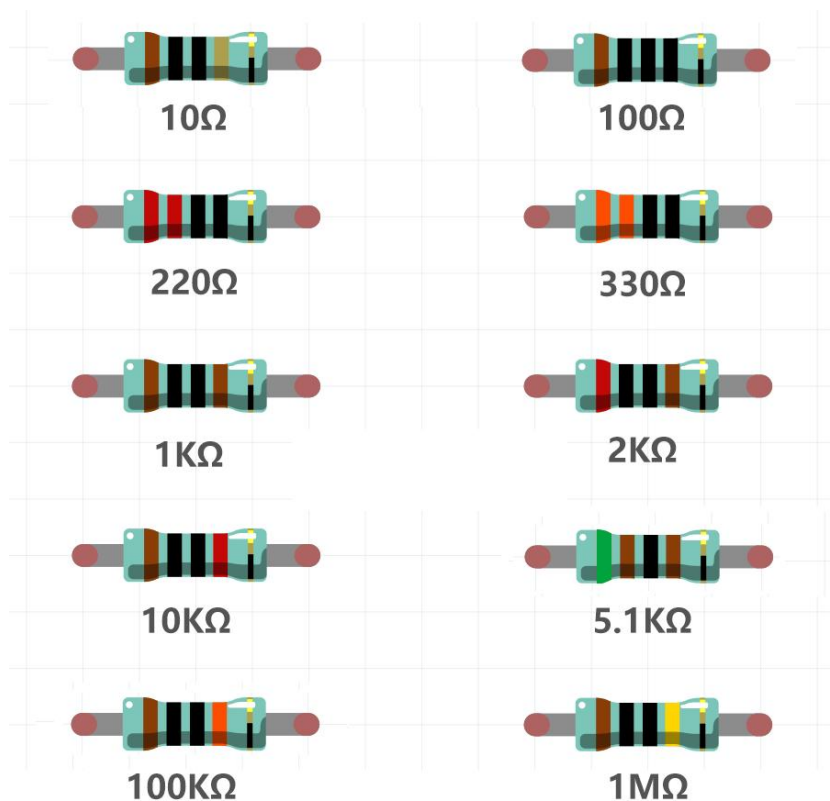
1 x Breadboard da 400 fori

Una scheda senza saldature che ti consente di costruire facilmente circuiti elettronici. È dotata di file di fori per collegare fili e componenti.



120 x Resistenze (10 di ciascuna, 30 resistenze da 220Ω)

Una resistenza è un componente che ostacola il flusso di corrente elettrica, alterando così la tensione e la corrente all'interno di un circuito. Il valore di una resistenza è misurato in ohm, simboleggiato dalla lettera greca omega (Ω). Le strisce colorate su una resistenza indicano il suo valore di resistenza e la tolleranza.



25 x LED (5 di ciascun colore)

Questa selezione di LED colorati comprende cinque colori: rosso, verde, blu, giallo e bianco, soddisfacendo diverse esigenze di illuminazione e segnalazione. Adatti per applicazioni che vanno da semplici indicatori di stato a complessi progetti di illuminazione decorativa, questi LED offrono una ricca gamma di colori per migliorare l'appeal visivo di qualsiasi progetto elettronico.



2 x RGB LEDs

Combinano LED rossi, verdi e blu in un'unica custodia. Possono visualizzare vari colori regolando la tensione di ingresso, creando milioni di colori.



1 x Photoresistor

Un photoresistore è un componente sensibile alla luce che modifica la sua resistenza in base all'intensità della luce a cui è esposto, ideale per creare controlli e sensori attivati dalla luce in progetti elettronici.



1 x NTC Thermistor

Un termistore è un resistore sensibile alle variazioni di temperatura. I termistori NTC (Negative Temperature Coefficient) riducono la resistenza al crescere della temperatura, mentre i termistori PTC (Positive Temperature Coefficient) aumentano la resistenza con l'aumento della temperatura.



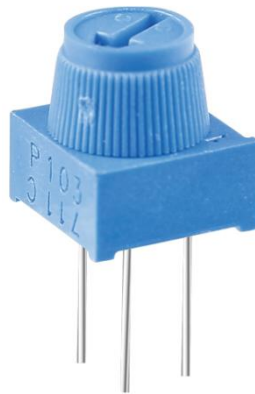
1 x Cicalino Attivo e 1 x Cicalino Passivo

Un cicalino, disponibile in versioni attive e passive, è un dispositivo di segnalazione audio che emette suoni quando viene applicata una corrente elettrica. È comunemente utilizzato in allarmi, timer e sistemi di notifica.



1 x Potenzimetro

Un potenziometro è un resistore variabile con tre terminali. Due terminali si collegano alle estremità di un resistore, mentre il terminale centrale è collegato a un cursore mobile, che divide il resistore in due parti. I potenziometri, spesso utilizzati per regolare la tensione nei circuiti, funzionano come le manopole del volume delle radio.



10 x Pulsanti Piccoli

Un pulsante piccolo è utilizzato per fornire una risposta fisica quando viene premuto, ed è comunemente impiegato nei dispositivi elettronici per avviare azioni o inserire comandi.



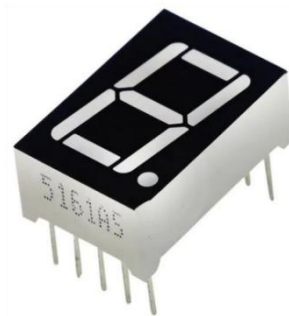
1 x 74HC595 Chip

Il 74HC595 è un registro di scorrimento utilizzato per espandere le porte di ingresso/uscita dei circuiti digitali, convertendo l'input seriale in output parallelo, riducendo così il numero di pin di connessione necessari. Questo chip è adatto per controllare un gran numero di dispositivi di output, come i display a 7 segmenti, senza occupare troppi pin del microcontrollore.



1 x Display a 7 segmenti

Un display a 7 segmenti è un componente a forma di otto che contiene 7 LED. Ogni LED è chiamato segmento: quando è attivato, un segmento contribuisce alla visualizzazione di un numero.



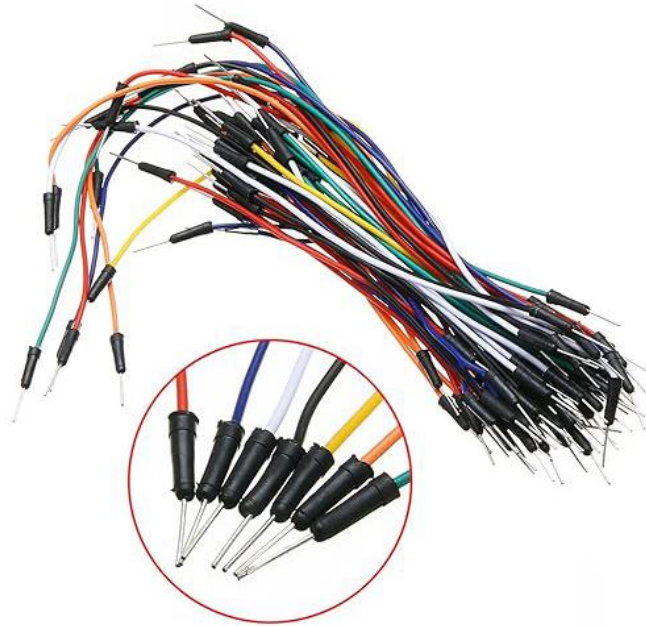
1 x Ultrasonic Module

Questo è un modulo ultrasonico che utilizza onde ultrasoniche per misurare distanze, rilevando e misurando con precisione la posizione e la distanza degli oggetti. È ampiamente utilizzato nella robotica, nei sistemi di evitamento ostacoli e nei campi del controllo automatico.



65 x Cavi Jumper

Collegano i componenti sulla breadboard tra loro e alla scheda Arduino.



10 x Cavi DuPont Maschio-Femmina

I cavi DuPont maschio-femmina sono progettati specificamente per collegare moduli con connettori a pin maschio, come il modulo ultrasuoni, alla breadboard. Questi cavi sono essenziali per interfacciare diversi componenti nei progetti elettronici.



1 x Cavo USB

Collega la scheda Arduino a un computer. Consente di scrivere, compilare e trasferire programmi sulla scheda Arduino. Alimenta anche la scheda.



1 x Batteria da 9V

Questa è una batteria alcalina da 9V non ricaricabile. È necessario installarla sul multimetro.



1 x Multimetro con Cavi Rossi e Neri

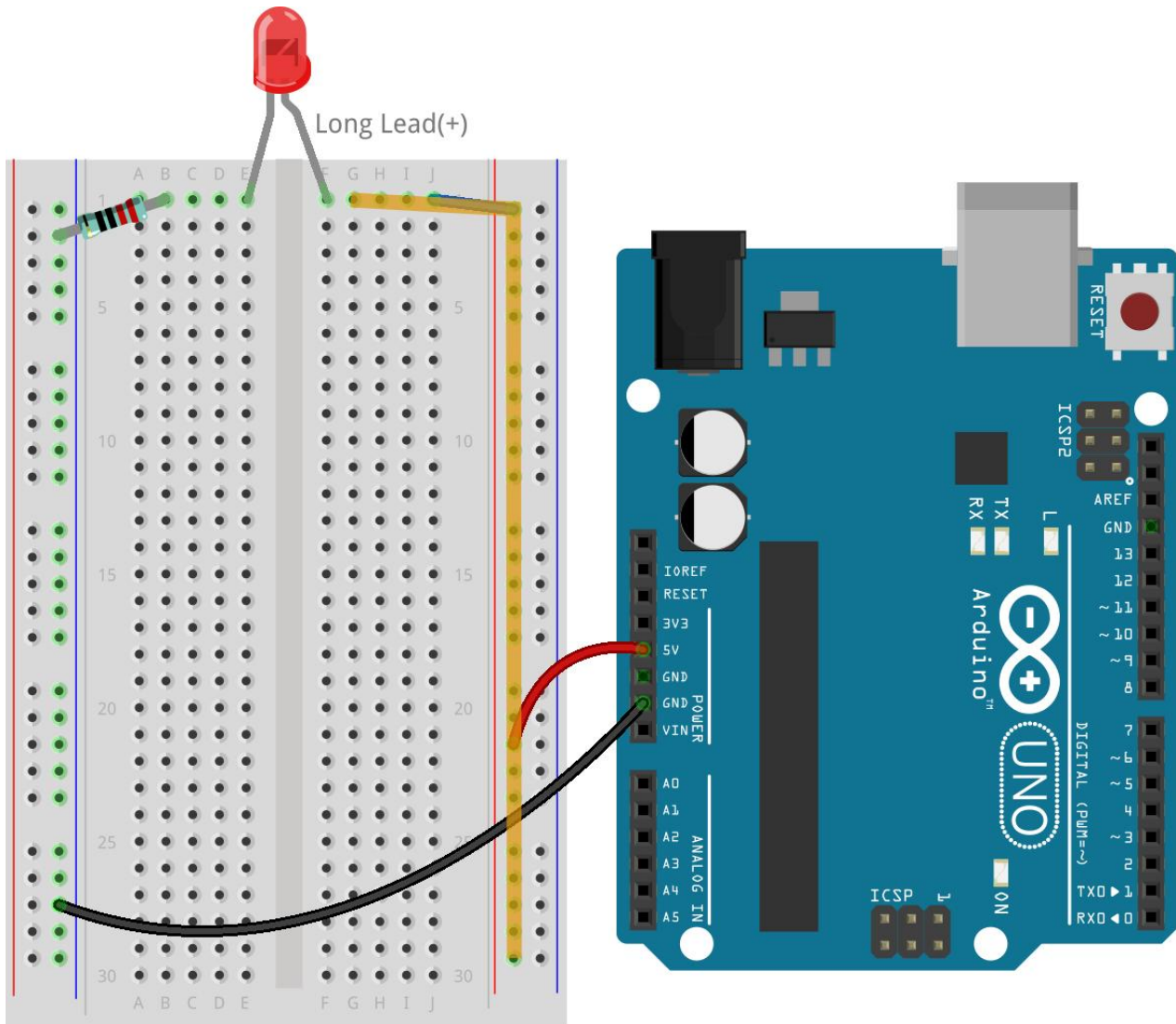
Questo è un multimetro versatile in grado di misurare tensione, corrente e resistenza, oltre a eseguire altri test elettrici, rendendolo uno strumento indispensabile per lavori di elettronica e elettricità.



Lezione 2: Il Tuo Primo Circuito

Dopo aver completato la lezione, rispondi alle seguenti domande.

1. Rimuovi il filo rosso dalla breadboard e sperimenta posizionandolo in diverse aperture della breadboard. Osserva eventuali cambiamenti nell'LED. Schizza le posizioni delle aperture che permettono all'LED di accendersi.



1. Cosa succede se si invertono i pin dell'LED? Si accenderà? Perché o perché no?

L'LED non si accende perché ha una conducibilità unidirezionale; la corrente deve fluire dall'anodo al catodo per funzionare.

Lezione 3: Misurare con il Multimetro

Rispondi a questa domanda dopo aver completato "**Scopri di più sul Multimetro!**"

1. Ora che hai una comprensione dettagliata su come utilizzare un multimetro, considera quale impostazione del multimetro utilizzeresti per misurare i seguenti valori elettrici?

Oggetto di Misurazione	Impostazione del Multimetro
9V volts DC	<i>20V</i>
1K ohms	<i>2kΩ</i>
40 milliamps	<i>200mA</i>
110 volts AC	<i>200V~</i>

Compila questa tabella durante "**Misurare con un Multimetro**"

Tipo	Unità	Risultati di Misurazione	Notes
Tensione	Volts	<i>≈ 5.13 volts</i>	
Corrente	Milliamps	<i>≈ 13.54 milliamps</i>	
Resistenza	Ohms	<i>≈ 378.88 ohms</i>	

Lezione 4: La Legge di Ohm

Compila la seguente tabella durante "Esplorando la Legge di Ohm con Esperimenti Pratici"

1. Sostituisci il resistore da 220 ohm con altri resistori di valori diversi come indicato di seguito. Annota le variazioni di luminosità dell'LED con ogni sostituzione per osservare come la resistenza influisca sulla corrente e, di conseguenza, sull'emissione di luce.

Resistore	Osservazioni
100Ω	<i>Più luminoso</i>
1KΩ	<i>Luminoso</i>
10KΩ	<i>Meno luminoso</i>
1MΩ	<i>Quasi spento</i>

Noterai che solo con il resistore da 100Ω l'LED è più luminoso rispetto al precedente resistore da 220Ω. Con resistenze più elevate, la luminosità dell'LED diminuisce fino a spegnersi completamente a 1MΩ. Perché avviene questo?

Secondo la legge di Ohm ($I = V/R$), aumentando la resistenza mentre la tensione rimane costante, la corrente attraverso l'LED diminuisce, riducendo così la luminosità dell'LED. A 1MΩ, la corrente è troppo bassa per accendere l'LED.

2. Dopo aver osservato gli effetti della variazione della resistenza, mantieni il resistore a 220 ohm e modifica la tensione di alimentazione del circuito da 5V a 3.3V. Registra eventuali cambiamenti nella luminosità dell'LED.

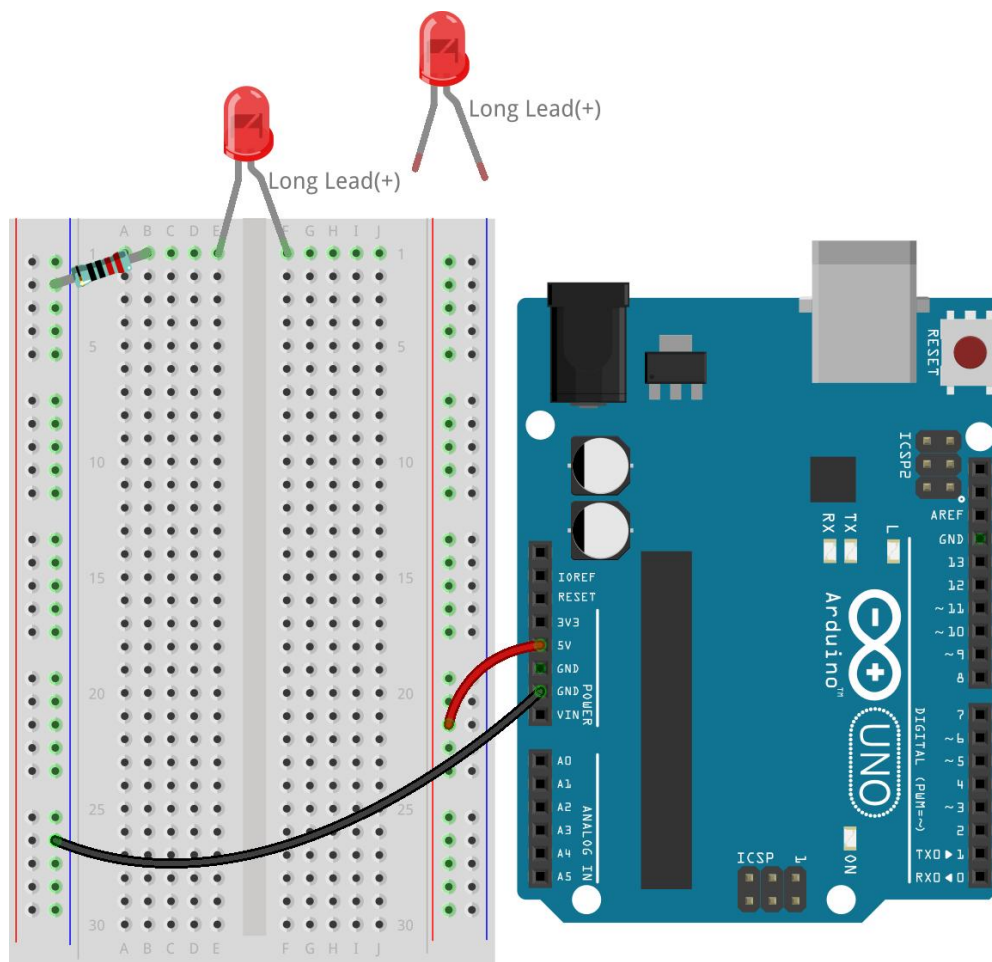
Scoprirai che l'LED è leggermente meno luminoso a 3.3V rispetto a 5V. Perché avviene questo?

Applicando la legge di Ohm, conoscendo la resistenza e la nuova tensione, la corrente dovrebbe essere $I = V/R$. Con una diminuzione della tensione mantenendo costante la resistenza, la corrente diminuisce, rendendo l'LED meno luminoso.

Lezione 5: Circuito in Serie vs. Circuito in Parallelo

Completa le seguenti domande durante "Approfondimento sui Circuiti in Serie"

1. Cosa succede se rimuovi un LED? Perché si verifica questo?



In un circuito in serie, se rimuovi un LED, l'altro LED non si accenderà. Questo avviene perché in un circuito in serie, la corrente deve passare attraverso ogni componente lungo il percorso. Rimuovere un LED interrompe il circuito, impedendo alla corrente di fluire attraverso l'LED rimanente.

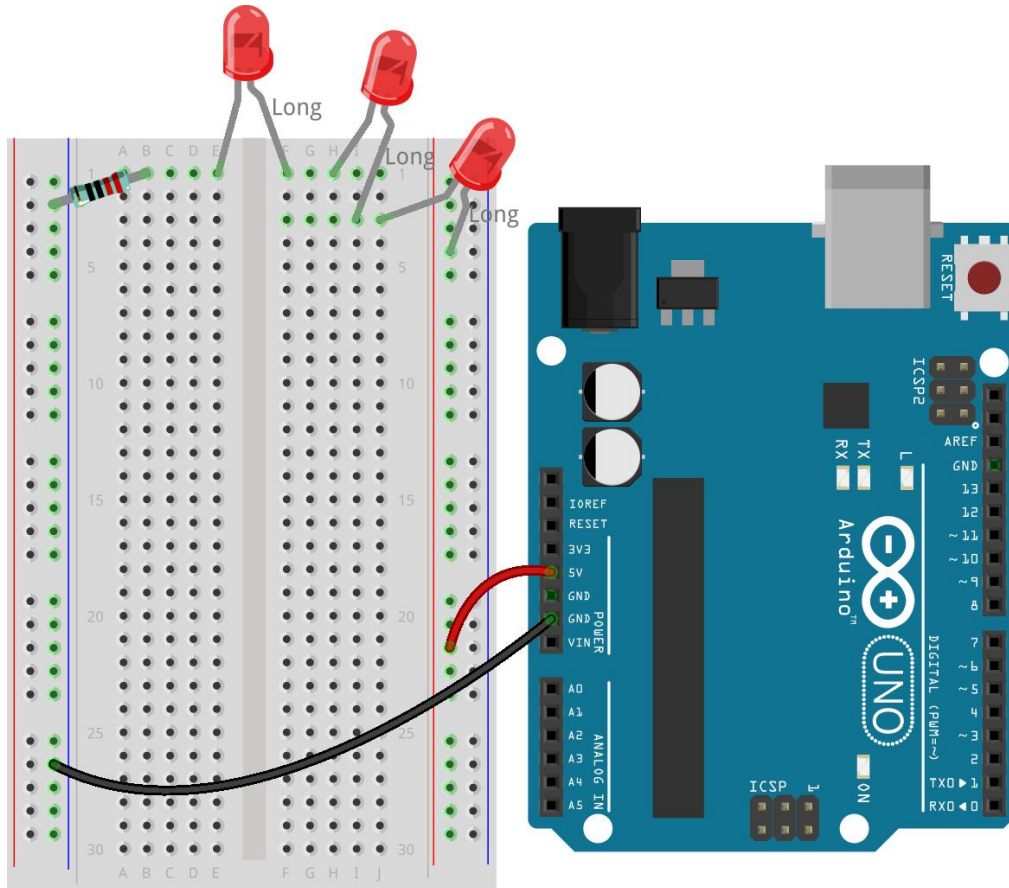
1. Misura la tensione di ciascun componente nel circuito in serie.

Circuito	Tensione Resistor	Tensione LED1	Tensione LED2	Tensione Totale
2 LEDs	$\approx 1.13 \text{ volts}$	$\approx 1.92 \text{ volts}$	$\approx 1.92 \text{ volts}$	$\approx 4.97 \text{ volts}$

2. Misura la corrente di ciascun componente nel circuito in serie.

Circuito	Corrente LED1	Corrente LED2
2 LEDs	≈ 4.43 milliamps	≈ 4.43 milliamps

3. Se viene aggiunto un altro LED a questo circuito, formando così un totale di tre LED, come cambia la luminosità degli LED? Perché? Come cambiano le tensioni sui tre LED?

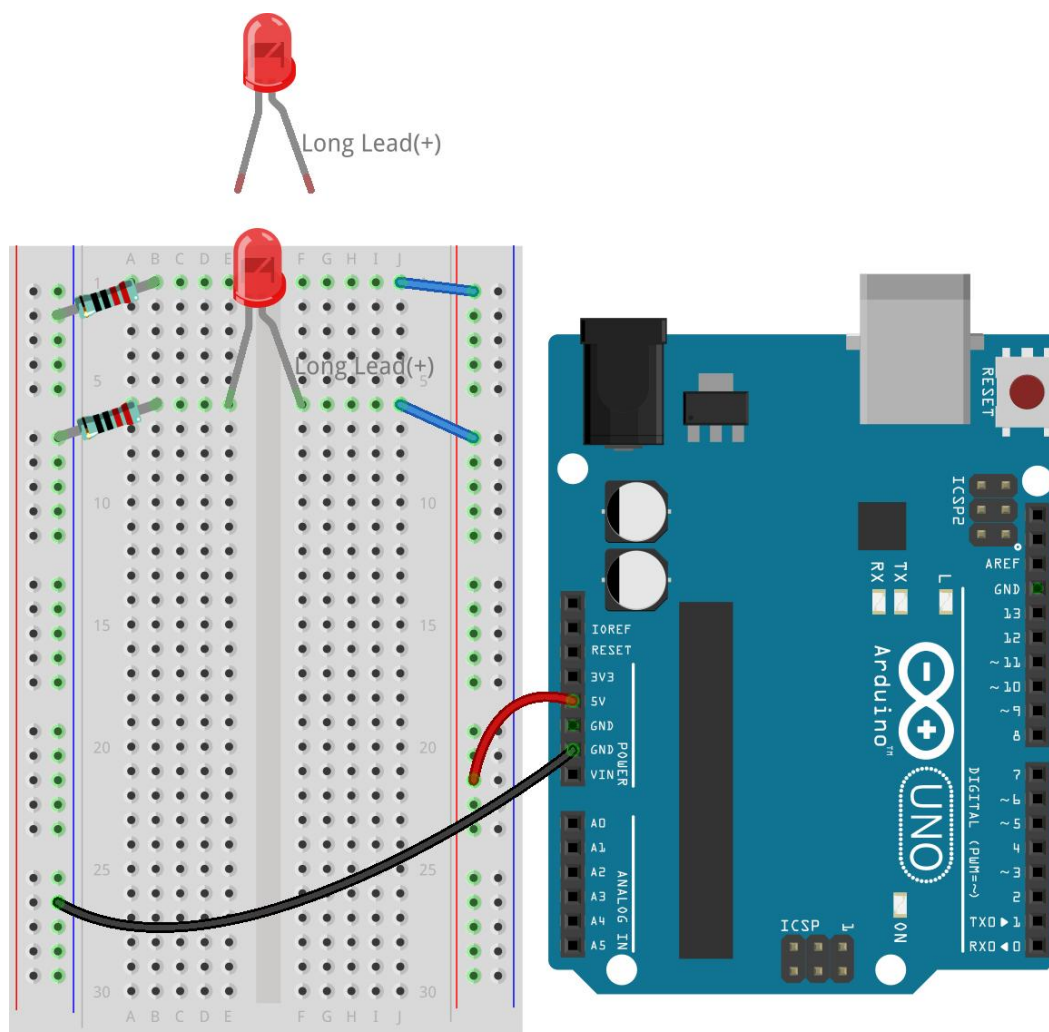


Aggiungere un altro LED a un circuito in serie che già ne contiene due porterà generalmente a una diminuzione della luminosità di ciascun LED. Questo accade perché la tensione totale della fonte di alimentazione viene suddivisa tra un numero maggiore di componenti, risultando in una caduta di tensione inferiore su ciascun LED rispetto a quando ce ne erano solo due. Di conseguenza, scorre meno corrente attraverso ciascun LED, riducendo la loro luminosità.

Per quanto riguarda le tensioni sui tre LED, ciascun LED avrà ora una porzione minore della tensione totale del circuito. Se la tensione della fonte di alimentazione rimane invariata, questa tensione sarà suddivisa per tre, assumendo che tutti i LED abbiano caratteristiche elettriche simili. Pertanto, la tensione su ciascun LED nel circuito sarà approssimativamente un terzo della tensione totale fornita dalla fonte di alimentazione.

Completa le seguenti domande durante "Esplorando i Circuiti Paralleli".

1. In questo circuito parallelo, cosa succede se si rimuove un LED? Perché avviene questo?



In un circuito parallelo, se si rimuove un LED, gli altri LED del circuito continueranno a illuminarsi. Questo avviene perché ogni LED in un circuito parallelo ha un percorso indipendente verso la fonte di alimentazione. Rimuovere un LED non interrompe il flusso di corrente verso gli altri LED, quindi questi rimangono inalterati e continuano a funzionare normalmente. Questa configurazione consente a ciascun componente in un circuito parallelo di operare indipendentemente dagli altri.

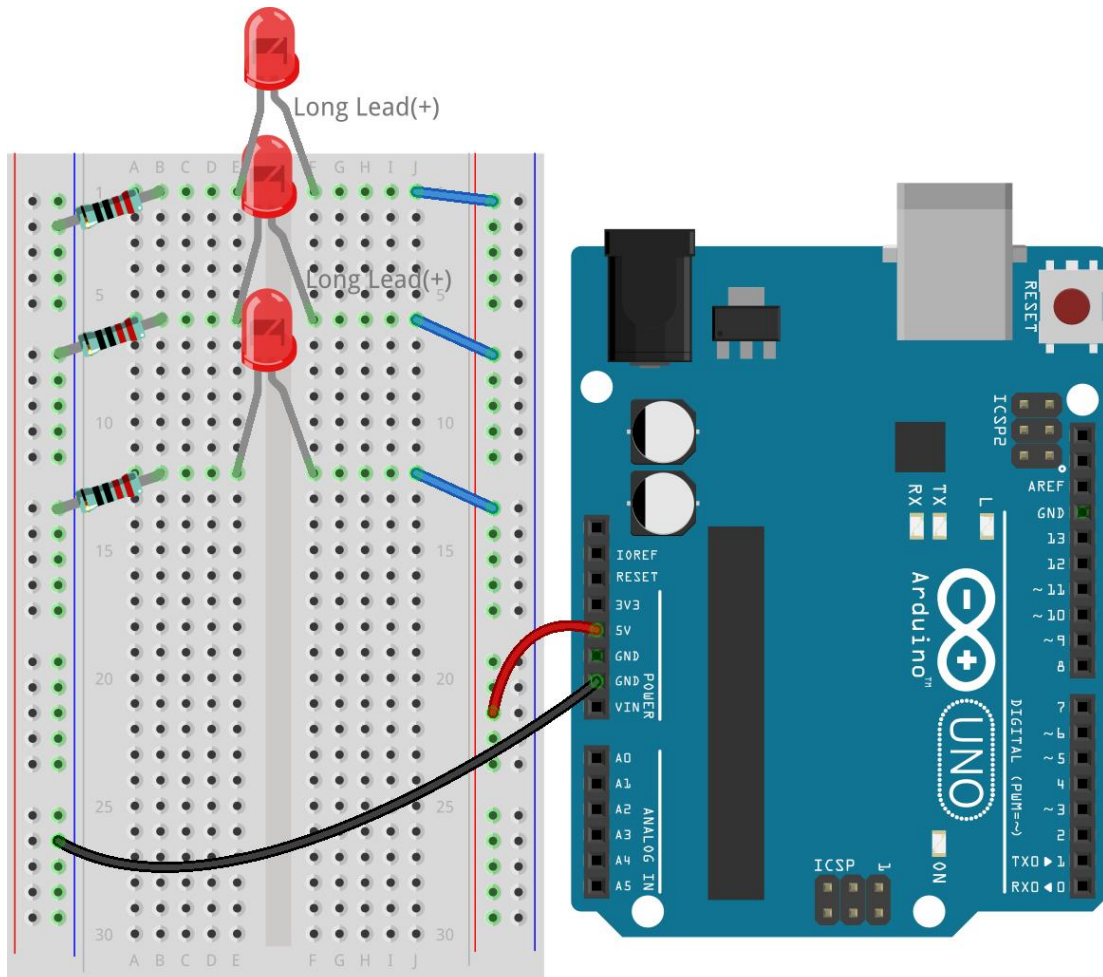
1. Registra la tensione misurata nella tabella.

Circuito	Tensione del Percorso 1	Tensione del Percorso 2
2 LEDs	≈ 5.00 volts	≈ 5.00 volts

2. Annota la corrente misurata nella tabella.

Circuito	Corrente LED1	Corrente LED2	Corrente Totale
2 LEDs	$\approx 12.6 \text{ milliamps}$	$\approx 12.6 \text{ milliamps}$	$\approx 25.3 \text{ milliamps}$

3. Se a un circuito viene aggiunto un altro LED, cosa succede alla luminosità degli LED? Perché? Registra la tua risposta nel tuo quaderno.



Quando si aggiunge un altro LED a un circuito parallelo, la luminosità dei LED già presenti di solito rimane invariata. Questo accade perché ogni LED in un circuito parallelo ha il proprio percorso diretto verso la fonte di alimentazione, quindi la tensione su ciascun LED rimane costante, indipendentemente dal numero di LED aggiunti. Ogni LED riceve la tensione necessaria per funzionare alla luminosità prevista. Pertanto, aggiungere ulteriori LED non influisce sulla luminosità dei LED già presenti, a condizione che l'alimentazione possa sostenere la domanda totale di corrente del circuito. Assicurati di registrare questo nel tuo quaderno per future consultazioni.

Lezione 6: LED lampeggiante

Compila la seguente tabella durante il "Rendere gli LED vivi"

1. Registra la tensione misurata nella tabella per il Pin 3.

State	Pin 3 Tensione
ALTO	<i>≈4.95 volts</i>
BASSO	<i>0.00 volts</i>

Dopo aver completato la lezione, rispondi alla seguente domanda.

1. Carica il codice sopra e vedrai il LED lampeggiare ripetutamente a intervalli di 3 secondi. Se desideri che si accenda e si spenga solo una volta, cosa dovresti fare?

Puoi spostare i comandi che accendono e spengono il LED dalla funzione `loop()` alla funzione `setup()`. La funzione `setup()` viene eseguita solo una volta all'avvio del programma, quindi questa modifica farà sì che il LED si accenda e si spenga un'unica volta. Ecco come puoi modificare il tuo codice:

```
void setup() {  
  // Setup code here, to run once:  
  pinMode(3, OUTPUT); // set pin 3 as output  
  
  digitalWrite(3, HIGH); // Light up the LED on pin 3  
  delay(3000);           // Wait for 3 seconds  
  digitalWrite(3, LOW);  // Switch off the LED on pin 3  
}  
  
void loop() {  
  // Main code here, to run repeatedly:  
}
```

Lezione 7: Costruiamo i Semafori!

Completa la seguente domanda durante "Scrittura del Pseudocodice per un Semaforo"

Pensa a cosa deve accadere affinché il tuo circuito funzioni come un semaforo. Nello spazio fornito nel tuo registro, annota il pseudocodice che descrive come funzionerà il tuo semaforo. Usa un linguaggio semplice.

Per simulare un semaforo utilizzando un Arduino, è necessario un sistema con tre LED (rosso, giallo e verde) e una sequenza che controlli l'illuminazione in modo da imitare i semafori reali. Ecco un semplice schema di pseudocodice che puoi annotare nel tuo registro per descrivere come potrebbe funzionare questo circuito del semaforo:

Setup:

- Define pins for the red, yellow, and green LEDs.
- Set all these pins as outputs.

Main Loop:

- Turn on the red LED for 5 seconds.
- Turn off the red LED.
- Turn on the yellow LED for 2 seconds.
- Turn off the yellow LED.
- Turn on the green LED for 5 seconds.
- Turn off the green LED.
- Repeat the cycle.

Completa la seguente domanda durante "Scrivere Pseudocodice per un Semaforo"

Dai un'occhiata agli incroci intorno a casa tua. Quanti semafori ci sono di solito? Come si coordinano tra loro?

Nelle aree urbane, gli incroci hanno spesso semafori per gestire in modo efficiente il flusso di veicoli e pedoni. Il numero di semafori in un incrocio può variare notevolmente a seconda delle sue dimensioni e complessità. Un semplice incrocio a quattro vie ha tipicamente almeno quattro semafori, uno per ogni direzione del traffico. Gli incroci più complessi possono avere luci aggiuntive per le corsie di svolta, i passaggi pedonali e altre esigenze di gestione del traffico.

Lezione 8: Semaforo con Pulsante per Pedoni

Dopo aver completato "**Costruire il Circuito**", rispondi alla seguente domanda

1. Il tuo semaforo è un mix di circuiti in serie e in parallelo. Discuti quali parti del tuo circuito sono in serie e perché. Poi, spiega quali parti sono in parallelo e perché.

Nel circuito, il pulsante e il suo resistore di pull-down da 10K sono collegati in serie. Questa configurazione garantisce che, quando il pulsante viene premuto, cambi correttamente lo stato del pin 8, collegandolo direttamente a massa quando non è premuto, evitando ingressi flottanti.

I tre LED collegati ai pin 3, 4 e 5 sono invece in parallelo tra loro. Ogni LED funziona in modo indipendente poiché sono collegati a pin di controllo separati e condividono un'alimentazione comune. Questa configurazione consente a ciascun LED di operare senza influenzare gli altri, il che è cruciale per un sistema di semafori.

Compila questa tabella durante "**Creazione del Codice**"

1. Compila la tabella con la tensione misurata sul pin 8 quando il pulsante è premuto e non premuto. Successivamente, indica gli stati corrispondenti di alto e basso livello.

Stato del Pulsante	Tensione Pin 8	Stato Pin 8
Rilasciato	<i>0.00 volts</i>	<i>BASSO</i>
Premuto	<i>≈4.97 volts</i>	<i>ALTO</i>

Completa la seguente domanda al termine di questa lezione

1. Durante il collaudo, potresti notare che il LED verde lampeggia solo mentre il pulsante pedonale è tenuto premuto, ma i pedoni non possono attraversare la strada mentre premendo continuamente il pulsante. Come puoi modificare il codice per garantire che, una volta premuto il pulsante pedonale, il LED verde si accenda a lungo abbastanza per un attraversamento sicuro senza richiedere una pressione continua? Si prega di scrivere la soluzione in pseudo-codice nel proprio quaderno.

Per garantire che il LED verde si accenda per i pedoni senza richiedere che il pulsante venga premuto continuamente, e per continuare il normale ciclo del semaforo successivamente, puoi modificare il tuo pseudocodice per controllare la pressione del pulsante e poi cambiare lo stato di funzionamento in base a quella pressione. Ecco una

versione ottimizzata e più chiara del pseudocodice che riflette queste modifiche:

Setup:

- Define pins for red, yellow, and green LEDs as output
- Define the button pin as input

Main Loop:

- Check if the button is pressed

- If button is pressed:

 - Turn off all LEDs

 - Turn on green LED for pedestrians

 - Delay 10 seconds

- Else:

 - Execute normal traffic light cycle:

 - Turn on green LED (for vehicles), turn off other LEDs

 - Delay 10 seconds

 - Turn on yellow LED, turn off other LEDs

 - Delay 3 seconds

 - Turn on red LED, turn off other LEDs

 - Delay 10 seconds

Lezione 9: Lampada da Scrivania Dimmerabile

Compila questa tabella durante "Costruisci il Circuito"

1. Ruota il potenziometro in senso orario dalla posizione 1 alla posizione 3 e misura la resistenza in ciascun punto, registrando i risultati nella tabella.

Punto di Misura	Resistenza (kilohm)
1	1.52
2	5.48
3	9.01

2. Come pensi che cambierebbe la tensione su A0 quando il potenziometro viene ruotato in senso orario e antiorario?

Puoi considerare il potenziometro come composto da due resistori collegati in serie all'interno del circuito. Secondo la misurazione delle resistenze, la resistenza tra A0 e GND aumenta quando il potenziometro viene ruotato in senso orario. Poiché la corrente rimane costante in un circuito in serie, secondo la Legge di Ohm ($\text{tensione} = \text{corrente} \times \text{resistenza}$), un aumento della resistenza porta a un incremento della tensione su A0. Pertanto, ruotando il potenziometro in senso orario si aumenta la tensione su A0, mentre ruotandolo in senso antiorario si riduce, poiché la resistenza diminuisce.

Completa la seguente domanda al termine di questa lezione.

1. Se colleghi il LED a un pin diverso, come il pin 8, e ruoti il potenziometro, la luminosità del LED cambierà ancora? Perché sì o perché no?

Se colleghi il LED al pin 8 di un Arduino UNO e ruoti il potenziometro, la luminosità del LED non cambierà. Questo perché il pin 8 non supporta il PWM (Modulazione di Larghezza di Impulso), che è necessario per regolare i livelli di luminosità utilizzando la funzione `analogWrite()`. Su un Arduino UNO, i pin che supportano il PWM e possono quindi essere utilizzati per controllare la luminosità di un LED tramite `analogWrite()` sono i pin 3, 5, 6, 9, 10 e 11.

Lezione 10: Lampada da Scrivania ON/OFF

Completa le seguenti domande al termine di questa lezione.

1. Cosa succederebbe se imposti il pin digitale 7 su INPUT solamente? Perché?

```
void setup() {  
  pinMode(9, OUTPUT); // Set pin 9 as output  
  pinMode(7, INPUT);  // Set pin 7 as input with an internal pull-up resistor  
  Serial.begin(9600); // Serial communication setup at 9600 baud  
}
```

Impostare il pin digitale 7 in modalità INPUT nel tuo sketch Arduino, anziché INPUT_PULLUP, può portare a una potenziale instabilità nel segnale letto dal pin. Quando un pin è configurato come INPUT e non è collegato né a un'alimentazione alta né a una bassa tramite circuiti esterni, diventa ciò che viene definito "floating" (fluttuante). Un pin fluttuante non si trova in uno stato stabile alto o basso; il suo stato può oscillare a causa di rumore elettrico o interferenze ambientali. Questa fluttuazione può portare a letture imprevedibili quando si tenta di leggere lo stato del pin tramite le funzioni di input digitale, risultando in dati errati o incoerenti ricevuti dal microcontrollore.

2. Se il pin 7 è impostato solo su INPUT, quali modifiche sarebbero necessarie al circuito?

Se il pin 7 della tua Arduino è impostato in modalità INPUT e vuoi garantire letture stabili e prevedibili, dovresti aggiungere una resistenza di pull-up esterna al circuito. Questo comporta il collegamento di una resistenza da 10kΩ tra il pin 7 e l'alimentazione di 5V dell'Arduino. La resistenza di pull-up assicura che il pin di ingresso si trovi in uno stato alto (livello logico 1) quando non è presente alcun altro segnale di ingresso.

Lezione 11: Controllo delle Matrici di LED con Potenzenziometro

Rispondi alle seguenti domande prima di procedere con "Creazione del Codice"

1. Scrivi il tuo pseudocodice per l'array di LED.

Lo pseudocodice funge da schizzo del programma, scritto in linguaggio semplice per facilitarne la comprensione. Il tuo compito è creare uno pseudocodice per un array di LED che reagisce a un potenziometro. Man mano che il valore del potenziometro aumenta, si accenderanno più LED.

Ecco un pseudocodice semplificato che delinea il controllo di un array di LED in base all'input di un potenziometro:

```
Declare readValue variable.  
Setup.  
Declare 3 digital pin outputs.  
Main Loop.  
If the potentiometer's value is below 200, all LEDs should be off.  
If the value is between 200 and 600, the first LED should be on.  
If the value is between 600 and 1000, the first two LEDs should be on.  
If the value exceeds 1000, all LEDs should be on.  
Delay for a short time.
```

Dopo aver completato la lezione, rispondi alla seguente domanda.

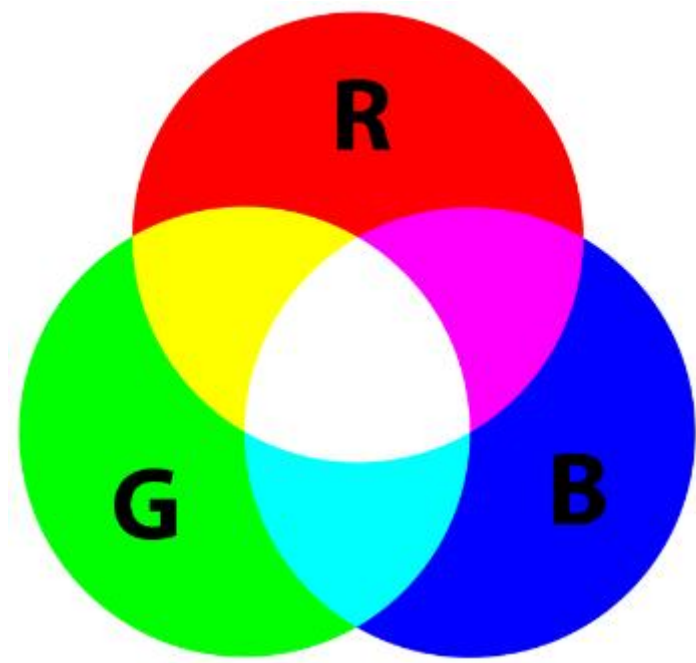
1. Nel codice precedente, determiniamo il numero di LED da accendere in base al valore del potenziometro. Come possiamo modificare il codice in modo che, mentre accendiamo i LED, la loro luminosità cambi in base al potenziometro?

Per modificare il tuo codice in modo che la luminosità dei LED cambi in base al valore del potenziometro, puoi utilizzare la funzione `analogWrite()` invece di `digitalWrite()`. La funzione `analogWrite()` ti consente di controllare la luminosità dei LED attraverso il PWM (Modulazione di Larghezza di Impulso).

Lezione 12: I Colori dell'Arcobaleno

Compila questa tabella durante "Creazione del Codice"

1.Se desideri altri colori, cosa dovresti fare? Fai riferimento al diagramma qui sotto e annota le tue idee nel tuo quaderno.



Colore	Pin Rosso	Pin Verde	Pin Blu
Rosso	<i>ALTO</i>	<i>BASSO</i>	<i>BASSO</i>
Verde	<i>BASSO</i>	<i>ALTO</i>	<i>BASSO</i>
Blu	<i>BASSO</i>	<i>BASSO</i>	<i>ALTO</i>
Giallo	<i>ALTO</i>	<i>ALTO</i>	<i>BASSO</i>
Rosa	<i>ALTO</i>	<i>BASSO</i>	<i>ALTO</i>
Ciano	<i>BASSO</i>	<i>ALTO</i>	<i>ALTO</i>
Bianco	<i>ALTO</i>	<i>ALTO</i>	<i>ALTO</i>

Lezione 13: Lo Spettro della Visione

Compila questa tabella durante "Creazione del Codice"

1. Ora puoi regolare i valori dei pin 9, 10 e 11 separatamente e registrare i colori osservati nel tuo quaderno.

Pin Rosso	Pin Verde	Pin Blu	Colore
0	128	128	<i>blu scuro</i>
128	0	255	<i>viola</i>
128	128	255	<i>azzurro</i>
255	128	0	<i>arancione</i>

2. Scegli alcuni dei tuoi colori preferiti e compila la tabella con i loro valori RGB.

Colore	Rosso	Verde	BIU

Lezione 14 : Colori Casuali

Dopo aver completato la lezione, rispondi alle seguenti domande.

1. Se cambi il codice da `randomSeed(analogRead(A0))` a `randomSeed(0)`, come cambieranno i colori del LED RGB e perché?

Se cambi il codice da `randomSeed(analogRead(A0))` a `randomSeed(0)`, la casualità dei colori del LED RGB sarà influenzata. La funzione `randomSeed(seed)` serve a inizializzare il generatore di numeri pseudo-casuali in Arduino, il che influisce sulla sequenza dei numeri casuali generati da funzioni come `random()`.

Utilizzare `analogRead(A0)` come valore di seme legge un valore casuale da A0, che è tipicamente influenzato dal rumore ambientale e da altri fattori, portando a valori di seme differenti ogni volta che il programma viene avviato. Ciò significa che la sequenza di numeri casuali (e quindi i colori) varierà ogni volta che l'Arduino viene ripristinato.

Al contrario, impostare il seme con un valore fisso come `randomSeed(0)` inizializza il generatore di numeri casuali con lo stesso punto di partenza ogni volta che il programma viene eseguito. Questo porta alla stessa sequenza di numeri casuali, e quindi il LED RGB mostrerà lo stesso schema di colori ogni volta che resettì o accendi l'Arduino. Questo elimina la casualità nei cambiamenti di colore del LED.

1. Quali sono alcune situazioni in cui si utilizza la casualità per risolvere problemi nella vita quotidiana, oltre a scegliere casualmente colori per la decorazione e numeri per la lotteria?

La casualità è utilizzata in vari contesti quotidiani per risolvere problemi, tra cui:

- **Giochi da tavolo:** Lanciare i dadi per determinare le mosse, garantendo che ogni partita sia diversa e equa.
- **Playlist musicali:** Mescolare le canzoni per mantenere l'esperienza di ascolto fresca e imprevedibile.
- **Scelte alimentari:** Scegliere casualmente un ristorante o un pasto può rendere la decisione più facile e divertente quando non si è sicuri di cosa mangiare.
- **Sistemazione dei posti:** Assegnare i posti a caso per eventi per mescolare gli ospiti e incoraggiare le interazioni sociali.
- **Serate di cinema:** Utilizzare un'estrazione casuale per scegliere un film quando tutti hanno preferenze diverse.

Lezione 15 : Colori Caldi o Freddi

Compila questa tabella durante "Creazione del Codice"

1. Apri Paint o un qualsiasi strumento di selezione dei colori, trova quelli che consideri i colori più caldi e più freddi, e annota i loro valori RGB nel tuo quaderno.

Tipo di Colore	Red	Green	Blue
Colore Caldo	246	52	8
Colore Freddo	100	150	255

Dopo aver completato la lezione, rispondi alla seguente domanda.

1. Si noti che i "limiti inferiori" di ciascun intervallo possono essere maggiori o minori dei "limiti superiori", quindi la funzione map() può essere utilizzata per invertire un intervallo di numeri, ad esempio:

```
y = map(x, 1, 50, 50, 1);
```

La funzione gestisce bene anche i numeri negativi, quindi questo esempio è valido e funziona correttamente.

```
y = map(x, 1, 50, 50, -100);
```

Per $y = \text{map}(x, 1, 50, 50, -100)$, se x è uguale a 20, quale dovrebbe essere y ? Fai riferimento alla seguente formula per calcolarlo.

$$\frac{\text{Value} - \text{From Low}}{\text{From High} - \text{From Low}} = \frac{\text{Y} - \text{To Low}}{\text{To High} - \text{To Low}}$$

$$\text{Y} = \frac{\text{Value} - \text{From Low}}{\text{From High} - \text{From Low}} \times (\text{To High} - \text{To Low}) + \text{To Low}$$

Per $x=20$ utilizzando la formula di mapping $y = \text{map}(x, 1, 50, 50, -100)$; il valore di y sarebbe approssimativamente -8.16 .

Lezione 16 : Allarme Temperatura

Compila questa tabella durante "Costruzione del Circuito"

1. Leggi il valore della resistenza a diverse temperature e registralo nella tabella sottostante.

Ambiente	Resistenza (kilohm)
Temperatura attuale	9.37
Temperatura elevata	6.10
Temperatura ridotta	12.49

Dopo aver completato la lezione, rispondi alla seguente domanda.

1. Nel codice, vengono calcolate le temperature in Kelvin e Celsius. Se desideri conoscere anche la temperatura in Fahrenheit, cosa dovresti fare?

Questo è il metodo standard per convertire i gradi Celsius in Fahrenheit e ti fornirà la temperatura in Fahrenheit basata sul valore Celsius che hai già ottenuto dai tuoi calcoli.

$$F = C * 1.8 + 32$$

2. Puoi pensare ad altre situazioni o luoghi in cui un sistema di monitoraggio della temperatura come quello che abbiamo costruito oggi potrebbe essere utile?

I sistemi di monitoraggio della temperatura trovano ampio utilizzo in situazioni quotidiane e in vari ambienti. Ecco alcuni esempi semplificati:

- **Comfort domestico:** Regola automaticamente il riscaldamento o il raffreddamento della tua casa in base alle letture della temperatura in tempo reale per mantenere il tuo spazio abitativo confortevole.
- **Giardinaggio:** Monitora le temperature delle serre per garantire che le piante crescano in condizioni ottimali. Aggiungi sistemi automatizzati per regolare le temperature quando necessario.
- **Sicurezza alimentare:** Tieni traccia delle temperature di frigorifero e congelatore per garantire che il cibo rimanga sicuro da mangiare, specialmente nei ristoranti o durante il trasporto alimentare.
- **Sanità:** Monitora e registra le temperature nelle aree di stoccaggio per medicinali e vaccini sensibili alla temperatura, per garantire che rimangano efficaci.

Lezione 17 : Codice Morse

Rispondi alla seguente domanda durante "Costruzione del Circuito"

1. Cosa succederà se colleghi il catodo di un buzzer attivo direttamente a GND e l'anodo a 5V? Perché?

Se colleghi il catodo di un buzzer attivo direttamente a GND e l'anodo a 5V, il buzzer emetterà un suono continuo. Questo avviene perché l'oscillatore interno del buzzer viene attivato dalla tensione di 5V, causando la generazione di suono fino a quando il circuito non viene disconnesso.

Dopo aver completato la lezione, rispondi alla seguente domanda.

1. Utilizzando la tabella del codice Morse fornita, scrivi un codice per inviare il messaggio "Ciao".

In codice Morse, "Ciao" verrebbe codificato come segue in base ai caratteri:

- H:
- E: .
- L: .-..
- L: .-..
- O: ---

Riassumendo, il codice Morse per "Ciao" è:

```
.... .-.. .-.. ---
```

Nella comunicazione pratica, di solito c'è una pausa più lunga tra le parole per differenziarle chiaramente, ma poiché "Ciao" è una sola parola, il codice è continuo con spazi che separano solo le singole lettere.

Lezione 18: Allarme Luminoso

Compila questa tabella durante "Costruzione del Circuito"

1. Leggi il valore della resistenza in base alla luce ambientale attuale e registralo nella tabella sottostante.

Ambiente	Resistenza (kilohm)
Luce Normale	≈ 5.48
Luce Intensa	≈ 0.16
Buio	≈ 1954

Dopo aver completato la lezione, rispondi alla seguente domanda.

1. Ladri astuti potrebbero scegliere di rubare di notte e, se un quadro scompare, il fotorilevatore potrebbe non rilevare alcuna variazione di luce, non riuscendo così ad attivare l'allarme. Cosa si può fare per migliorare questa lacuna?

Una soluzione consiste nell'installare una fonte di luce di fronte al dipinto. Questo non solo garantisce che il dipinto sia ben visibile e illuminato, ma assicura anche che qualsiasi disturbo o rimozione del dipinto attivi immediatamente l'allarme del fotorilevatore, modificando il livello di luce rilevato.

Lezione 19: Sistema di Allarme per il Parcheggio in Retromarcia

Dopo aver completato la lezione, rispondi alla seguente domanda

1. Se desideri che la distanza rilevata da questo dispositivo sia più precisa a livello decimale, come dovresti modificare il codice?

Per rendere la misurazione della distanza più precisa a livello decimale, puoi modificare il calcolo nella funzione `measureDistance` utilizzando l'aritmetica a virgola mobile invece di quella intera. In questo modo, il risultato includerà valori decimali, fornendo misurazioni più accurate.

```
float distance = duration * 0.034 / 2.0;
```

La variabile `distance` nella funzione `measureDistance` è stata convertita in `float` per supportare valori decimali.

Il calcolo della `distance` utilizza l'aritmetica a virgola mobile dividendo per `2.0` anziché per `2`.

Lezione 20: Il Timer Pomodoro

Rispondi alla seguente domanda durante la lezione "Creazione di Codice - millis()"

1. Se `*delay(100)*` viene modificato in `*delay(1000)*`, cosa succederà al programma? Perché?

```
if (currentMillis - previousMillis >= interval) {  
  previousMillis = currentMillis; // Save the last time the buzzer beeped  
  digitalWrite(buzzerPin, HIGH); // Make a voice  
  delay(100);  
  digitalWrite(buzzerPin, LOW); // silence  
}
```

Nel codice originale, il cicalino emette un segnale acustico per circa 100 millisecondi ogni 1000 millisecondi (1 secondo, come impostato dalla variabile `*interval*`), seguito da un silenzio di 900 millisecondi. Dopo la modifica, il cicalino emetterà un segnale per 1000 millisecondi ogni 1000 millisecondi e poi riprenderà quasi immediatamente, poiché l'intervallo successivo inizia subito. Pertanto, cambiando il `*delay*` da 100 a 1000 millisecondi, il cicalino passa da brevi segnali acustici a un suono continuo, diventando più fastidioso e inadatto all'intento originale.

Modificare `*delay(100)*` in `*delay(1000)*` nel tuo codice farà sì che il cicalino emetta un suono per un secondo intero invece di un breve segnale acustico, poiché aumenta il tempo di pausa quando il cicalino è attivo. Questo comporta suoni più lunghi e cicli del programma meno frequenti, rendendo potenzialmente il programma meno reattivo ad altri compiti durante questi intervalli.

Dopo aver completato la lezione, rispondi alla seguente domanda

Pensa ad altri contesti della tua vita in cui puoi "sentire" il tempo. Elenca alcuni esempi e scrivilli nel tuo manuale!

Sentire il tempo è un concetto interessante e ci sono diversi scenari quotidiani in cui possiamo sperimentarlo. Ecco alcuni esempi che potresti annotare:

- **Orologi e Sveglie:** Il ticchettio degli orologi analogici o i segnali acustici specifici degli orologi digitali che indicano il passare di ogni secondo o minuto.
- **Timer da Cucina:** Il ticchettio e l'allarme finale di un timer da cucina meccanico o

digitale che scandisce il tempo di cottura o di preparazione.

- **Campanelli Scolastici:** Il suono delle campanelle nelle scuole che segnano l'inizio e la fine delle lezioni o delle pause.
- **Annunci nei Trasporti Pubblici:** I segnali acustici o i jingle che precedono gli annunci nelle stazioni ferroviarie o sugli autobus, indicando l'imminente partenza o arrivo.
- **Forno a Microonde:** Il segnale acustico alla fine del timer, che avvisa che il processo di riscaldamento è terminato.
- **Tracker di Fitness o Orologi Sportivi:** I segnali acustici o gli allarmi che indicano il completamento di un intervallo di tempo durante gli allenamenti o le sessioni di esercizio.

Lezione 21: Suono della Sirena

Rispondi alla seguente domanda durante la fase **"Costruisci il Circuito"**

1. Cosa succede se colleghi il catodo di un cicalino passivo direttamente a GND e l'anodo a 5V? Perché?

Se colleghi il catodo di un cicalino passivo direttamente a GND e l'anodo a 5V, a differenza di un cicalino attivo, il cicalino passivo non emetterà alcun suono perché non dispone di un oscillatore interno. Un cicalino passivo richiede un segnale esterno per generare suoni. Solitamente, è necessario pilotarlo con un'onda quadra (tensione oscillante) alla frequenza desiderata per produrre suoni udibili.

Rispondi alle seguenti domande durante la fase **"Creazione del Codice - Far Suonare il Cicalino Passivo"**

1. Se cambi i pin nel codice e nel circuito a 7 o 8, che non sono pin PWM, il cicalino emetterà comunque un suono? Puoi fare un test e poi scrivere la tua risposta nel manuale.

Anche se il pin 8 non è un pin PWM, la funzione `*tone()` può comunque generare un'onda quadra precisa su di esso, permettendo di pilotare efficacemente un cicalino passivo per produrre suoni. Questa flessibilità consente di utilizzare qualsiasi pin digitale per l'uscita audio senza essere limitati ai pin compatibili con PWM. Quando si chiama la funzione `*tone(pin, frequency)*`, Arduino configura un timer per alternare lo stato del pin (da HIGH a LOW e di nuovo a HIGH) alla frequenza specificata, creando un'onda quadra. Questa onda quadra pilota il cicalino passivo, facendolo emettere un suono alla frequenza dell'onda generata.

1. Per esplorare come la frequenza e la durata nella funzione `*tone(pin, frequency, duration)*` influenzino il suono del cicalino, modifica il codice in base a due condizioni e annota i fenomeni osservati nel tuo manuale:

- Mantenendo la frequenza a 1000, aumenta gradualmente la durata da 100, 500 a 1000. Come cambia il suono del cicalino e perché?
 - Durata di 100 ms: Il suono è un breve bip.
 - Durata di 500 ms: Il suono è un bip più lungo, chiaramente udibile e della durata di mezzo secondo.
 - Durata di 1000 ms: Il suono è ancora più lungo, della durata di un secondo intero.

Man mano che aumenti la durata, il suono emesso dal cicalino persiste più a lungo. Il tono o la frequenza del suono rimane costante (essendo impostata a 1000 Hz), il che significa che la "nota" del tono non cambia, ma aumenta il tempo in cui lo si percepisce. Questo è utile per segnalare diverse durate di allarmi, dove l'urgenza o il tipo di allarme possono essere distinti in base alla lunghezza del tono.

- Mantenendo la durata a 100, aumenta gradualmente la frequenza da 1000, 2000 fino a 5000. Come cambia il suono del cicalino e perché?
 - Frequenza di 1000 Hz: il suono è un bip a tonalità media.
 - Frequenza di 2000 Hz: il suono ha una tonalità più alta rispetto a 1000 Hz.
 - Frequenza di 5000 Hz: il suono è molto più acuto, probabilmente percepito come più stridulo e potenzialmente fastidioso a distanza ravvicinata.

Aumentare la frequenza mantenendo costante la durata comporta una variazione dell'altezza del suono. Frequenze più alte generano suoni con tonalità più acute. Questo principio è utile per distinguere diversi tipi di notifiche o segnali in base alla loro urgenza o importanza, poiché le tonalità più alte sono spesso utilizzate per avvisi più urgenti.

Lezione 22: Suonare "Twinkle, Twinkle, Little Star"

Rispondi alla seguente domanda durante la fase di "Creazione del Codice - Array".

1. Puoi anche eseguire operazioni sugli elementi dell'array, come modificare in `Serial.println(melody[i] * 1.3)`, Quali dati otterrai e perché?

Il numero 1,3 è un numero a virgola mobile. Quando un intero dell'array `melody` (di tipo `int`) viene moltiplicato per 1,3, il risultato dell'operazione viene automaticamente convertito in un numero a virgola mobile (`float`).

Per ogni frequenza di nota in questo array, moltiplicando per 1,3 e poi stampando il risultato si otterrà:

```
340.6  
340.6  
509.6  
509.6  
572.0  
572.0  
509.6  
...
```

Dopo aver completato la lezione, rispondi alla seguente domanda.

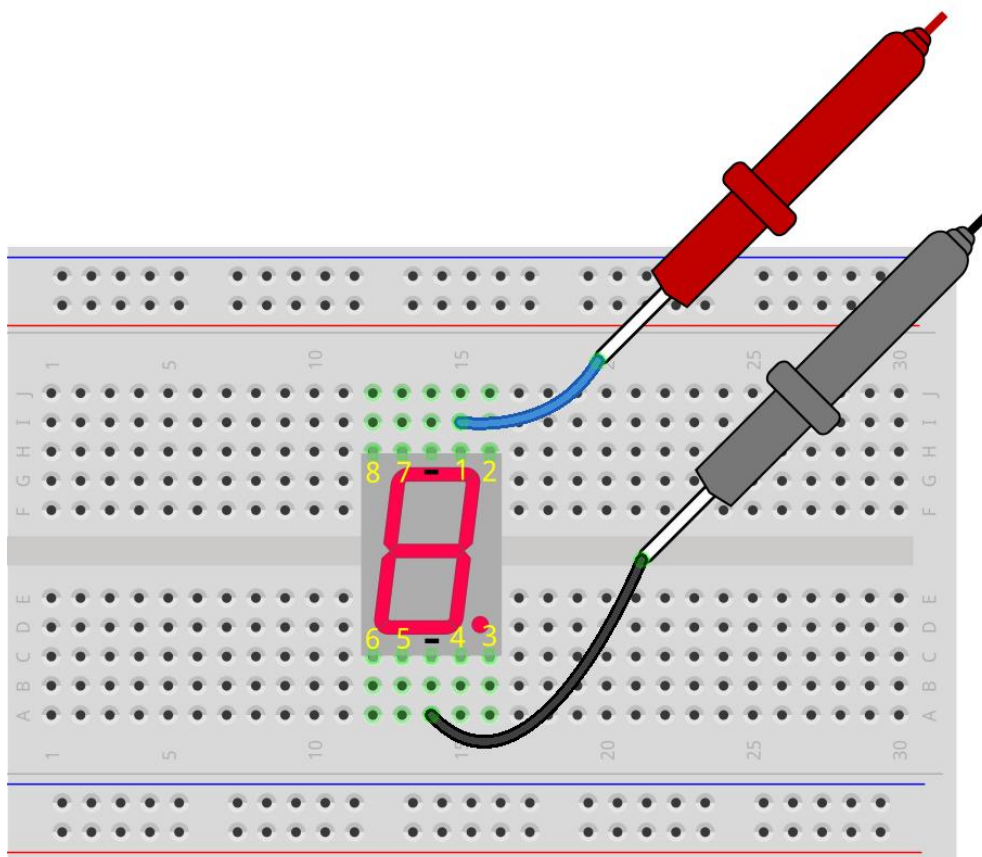
1. Se sostituisci il cicalino passivo nel circuito con un cicalino attivo, riesci a riprodurre correttamente "Twinkle Twinkle Little Star"? Perché?

Se sostituisci il cicalino passivo con un cicalino attivo per suonare "Twinkle Twinkle Little Star", il risultato non sarà quello desiderato. I cicalini attivi possono produrre solo un singolo tono perché hanno un oscillatore integrato. Di conseguenza, non è possibile controllare l'altezza delle note per riprodurre la melodia in modo accurato; si sentirebbe solo un bip ripetitivo con il ritmo della canzone, ma non le note reali.

Lezione 23: Cyber Dice

Rispondi alle seguenti domande durante la fase di "Comprensione del Display a 7 Segmenti".

1. Se un segmento si accende, fai riferimento a questo diagramma per annotare il numero di pin del segmento e la sua posizione approssimativa nella tabella del Manuale.



Pin	Numero Segmento	Posizione
1	A	La parte superiore
2	B	La parte superiore destra
3	C	La parte inferiore destra
4	D	La parte inferiore
5	E	La parte inferiore sinistra
6	F	La parte superiore sinistra
7	G	La parte centrale
8	Punto decimale	Il punto

2. Dai test sopra descritti, si sa che il display nel kit è a catodo comune, il che significa che è sufficiente collegare il pin comune a GND e fornire un'alta tensione agli altri pin per accendere i segmenti corrispondenti. Se desideri che il display mostri il numero 2, a quali pin dovresti fornire un'alta tensione? Perché?



Per il numero 2, i segmenti a, b, d, e e g devono essere attivati (impostati su alta tensione) perché sono i segmenti che formano il numero 2 sul display. I segmenti f, c e dp (punto decimale, se presente) devono rimanere spenti (bassa tensione) poiché non fanno parte della visualizzazione del numero 2.

Quindi, i pin a cui deve essere fornita un'alta tensione sono quelli collegati ai segmenti a, b, d, e e g per visualizzare correttamente il numero 2.

Lezione 24: Luce Scorrevole con 74HC595

Rispondi alla seguente domanda durante la fase di "Creazione del Codice - Accensione dei LED".

1. Cosa succede se cambiamo `MSBFIRST` con `LSBFIRST` in `shiftOut(DS, SHcp, MSBFIRST, B11101110);`? Perché?

Se cambi `MSBFIRST` con `LSBFIRST`, l'ordine dei bit viene invertito e il byte viene trasmesso a partire dal bit meno significativo (quello più a destra). Se utilizzi il registro a scorrimento per controllare i LED, cambiare l'ordine dei bit invertirà la sequenza di accensione dei LED. Invece di accendersi nella sequenza programmata inizialmente, i LED si accenderanno nell'ordine inverso.

Dopo aver completato la lezione, rispondi alla seguente domanda.

1. Se vogliamo che tre LED siano accesi contemporaneamente e che diano l'impressione di "scorrere", come dovrebbero essere modificati gli elementi dell'array `dataArray[]`?

Dovresti iniziare con i primi tre LED accesi e poi spostare un LED verso destra in ogni schema successivo fino a quando non saranno accesi gli ultimi tre LED. Ecco come potresti definire questi schemi in formato binario:

```
B11100000: LEDs 1, 2, 3 are on; others are off.  
B01110000: LEDs 2, 3, 4 are on; others are off.  
B00111000: LEDs 3, 4, 5 are on; others are off.  
B00011100: LEDs 4, 5, 6 are on; others are off.  
B00001110: LEDs 5, 6, 7 are on; others are off.  
B00000111: LEDs 6, 7, 8 are on; others are off.
```

```
byte dataArray[] = { B11100000, B01110000, B00111000, B00011100, B00001110,  
B00000111 };
```


Lezione 25: Visualizzare un Numero

Compila questa tabella durante la lezione **"Numeri Binari per le Cifre da 0 a 9"**.

1. Ora che conosciamo le rappresentazioni binarie per le cifre 0 e 2, compila i numeri binari per le cifre rimanenti nella tabella qui sotto.

Numero	Binario
0	<i>B00111111</i>
1	<i>B00000110</i>
2	<i>B01011011</i>
3	<i>B01001111</i>
4	<i>B01100110</i>
5	<i>B01101101</i>
6	<i>B01111101</i>
7	<i>B00000111</i>
8	<i>B01111111</i>
9	<i>B01101111</i>

Compila questa tabella durante la lezione **"Conversione Binaria"**.

1. Converti i numeri binari che rappresentano le cifre da 0 a 9 in numeri decimali ed esadecimali utilizzando una calcolatrice e completa la tabella. Questo ti fornirà una guida di riferimento rapido per le conversioni tra basi.

Numero	Binario	Decimale	Esadecimale
0	B00111111	63	0x3F
1	B00000110	6	0x06
2	B01011011	91	0x5B
3	B01001111	79	0x4F
4	B01100110	102	0x66

5	B01101101	<i>109</i>	<i>0x6D</i>
6	B01111101	<i>125</i>	<i>0x7D</i>
7	B00000111	<i>7</i>	<i>0x07</i>
8	B01111111	<i>127</i>	<i>0x7F</i>
9	B01101111	<i>111</i>	<i>0x6F</i>