

# Beginner's Lab Kit をお選びいただきありがとうございます。

ビギナーズラボキットへようこそ。このキットは、電子工作とプログラミングの世界に初めて触れる方々 のために設計された、包括的なスターターパックです。LED、抵抗器、ブザー、ポテンシオメータ、フォト レジスタ、サーミスタ、プッシュボタン、デジタルチューブ、超音波モジュールなど、基本的な部品が多数 含まれています。このキットの目玉の一つは、電流、電圧、抵抗を測定できるマルチメーターの追加です。 このツールは、各部品の機能をより深く理解するのに非常に役立ちます。

このキットに付属するコースは、Arduino のプログラミング構文に基づいて構成されており、論理的かつ教 育的な進行を保証します。この構造により、回路を一步步構築し、それを制御するプログラムを書く方法 を学ぶことができます。コース全体を通じて、トラブルシューティングの課題に直面することで、教材の理 解を深めることができます。

お問い合わせやサポートが必要な場合は、[service@sunfounder.com](mailto:service@sunfounder.com) までご連絡ください。ビギナーズラボ キットを使って学びの旅に出発し、電子工作のエキサイティングな世界を探索しましょう！

# 目次

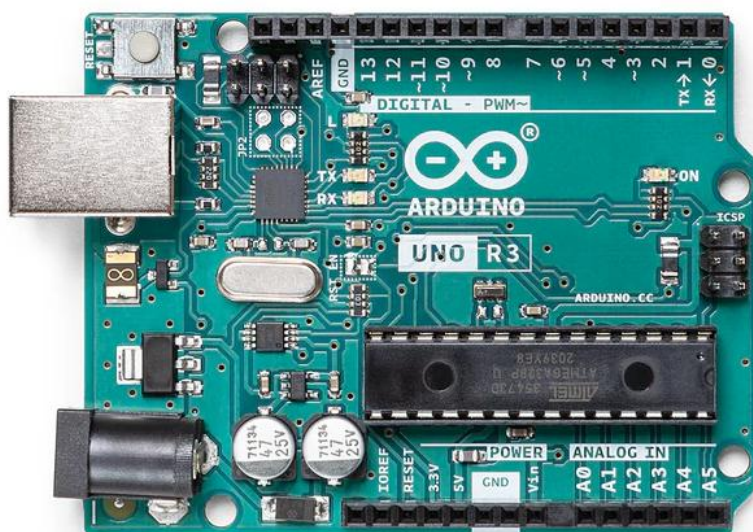
キットに含まれているもの .....	1
レッスン 2: 最初の回路 .....	8
レッスン 3: マルチメータの使い方を学ぶ .....	9
レッスン 4: オームの法則 .....	10
レッスン 5: 直列回路と並列回路 .....	11
レッスン 6: LED の点滅 .....	15
レッスン 7: 交通信号を作ろう! .....	16
レッスン 8: 歩行者ボタン付き信号機 .....	17
レッスン 9: 調光可能なデスクランプ .....	19
レッスン 10: ON/OFF デスクランプ .....	20
レッスン 11: ポテンショメーターで LED アレイを制御 .....	21
レッスン 12: 虹の色 .....	22
レッスン 13: 視覚のスペクトル .....	23
レッスン 14: ランダムな色 .....	24
レッスン 15: クールまたはウォームな色 .....	25
レッスン 16: 温度アラーム .....	26
レッスン 17: モールス信号 .....	27
レッスン 18: 光アラーム .....	28
レッスン 19: リバースパーキングアラームシステム .....	29
レッスン 20: ポモドーロタイマー .....	30
レッスン 21: サイレンサウンド .....	32
レッスン 22: 「きらきら星」を演奏する .....	34
レッスン 23: サイバーダイス .....	35
レッスン 24: 74HC595 を使った流れる光 .....	37
レッスン 25: 番号の表示 .....	38

## キットに含まれているもの

このコースで回路を作るために使用するさまざまな部品やパーツがキットに含まれています。以下はその内容の簡単なガイドです。

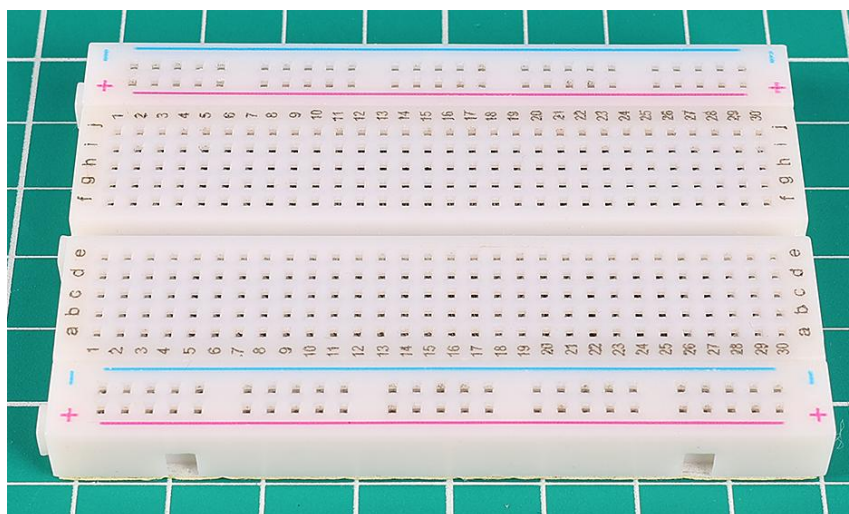
### 1 x オリジナル Arduino Uno R3

回路の頭脳であるマイクロコントローラーボードです。マイクロコントローラーをサポートするために必要なすべてが揃っており、USB ケーブルでコンピューターに接続するか、AC-DC アダプターやバッテリーで電源を供給するだけで使用を開始できます。



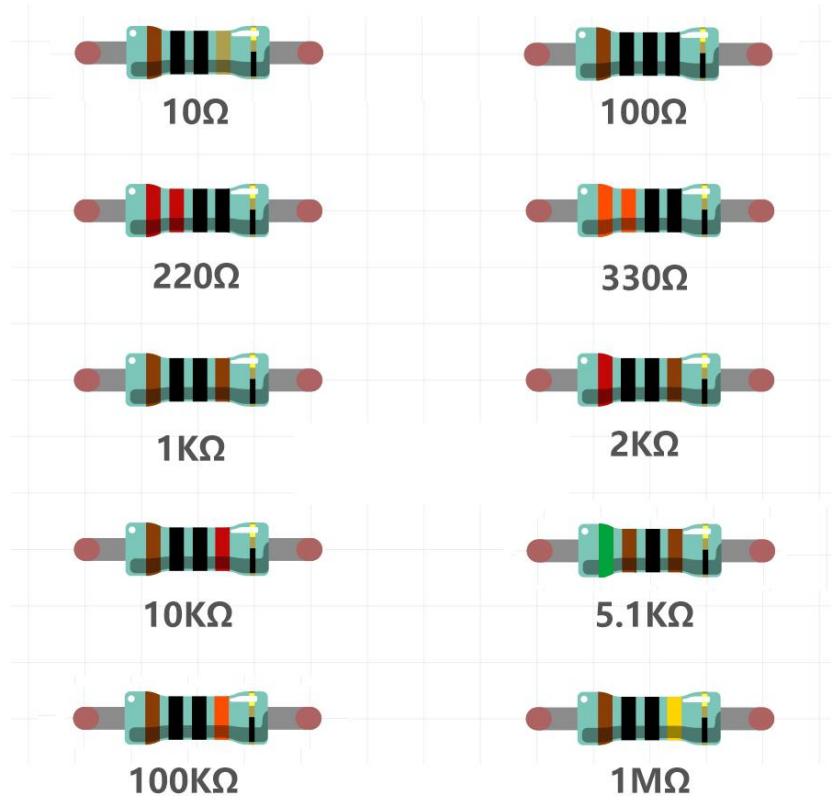
### 1 x 400 穴ブレッドボード

簡単に電子回路を構築できるはんだ不要のボードです。ワイヤーやコンポーネントを接続するための穴が並んでいます。



## 120 x 抵抗器 (各 10 個、220Ω抵抗器は 30 個)

抵抗器は電力の流れを妨げるコンポーネントで、回路内の電圧と電流を変える役割を果たします。抵抗器の値はオームで測定され、ギリシャ文字のオメガ ( $\Omega$ ) で表されます。抵抗器のカラーコードは、その抵抗値と許容誤差を示します。



## 25 x LED (各色 5 個)

このカラフルな LED セットには、赤、緑、青、黄、白の 5 色が含まれており、さまざまな照明や信号のニーズに対応できます。単純な状態表示から複雑な装飾照明プロジェクトまで、これらの LED はどんな電子プロジェクトでも視覚的な魅力を高める豊かな色の選択肢を提供します。



## 2 x RGB LED

赤、緑、青の LED を 1 つのケースに組み合わせたもので、入力電圧を調整することでさまざまな色を表示でき、数百万色を作り出すことができます。



## 1 x 光センサー

光センサーは、光の強度に応じて抵抗値が変わる光感受性コンポーネントで、電子プロジェクトで光を利用したコントロールやセンサーを作成するのに最適です。



## 1 x NTC サーミスタ

サーミスタは温度変化に敏感な抵抗器です。NTC サーミスタは温度が上昇すると抵抗が減少し、PTC サーミスタは温度が上昇すると抵抗が増加します。



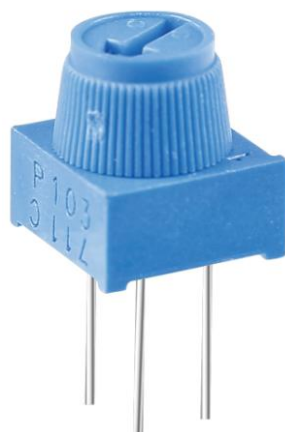
## 1 x アクティブブザー & 1 x パッシブブザー

ブザーは、電流が流れると音を発するオーディオ信号装置で、アクティブタイプとパッシブタイプがあります。アラーム、タイマー、通知システムによく使用されます。



## 1 x ポテンショメーター

ポテンショメーターは、3本のピンを持つ可変抵抗器です。2本のピンは抵抗器の両端に接続され、中央のピンは可動ワイパーに接続され、抵抗器を2つに分割します。ポテンショメーターは回路内の電圧を調整するために使用され、ラジオのボリュームノブのような役割を果たします。



## 10 x 小型ボタン

小型の押しボタンは、押すと物理的な反応を提供し、電子機器でアクションを開始したりコマンドを入力するために一般的に使用されます。





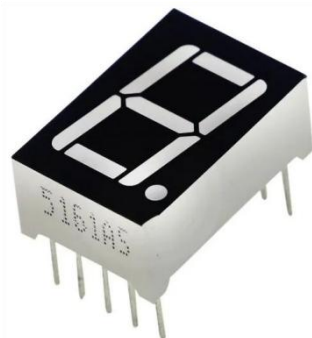
## 1 x 74HC595 チップ

74HC595 はシフトレジスタで、シリアル入力をパラレル出力に変換することでデジタル回路の入出力ポートを拡張し、接続ピンの数を減らします。このチップは、マイクロコントローラーのピンを多く占有することなく、7 セグメントディスプレイなどの多くの出力デバイスを制御するのに適しています。



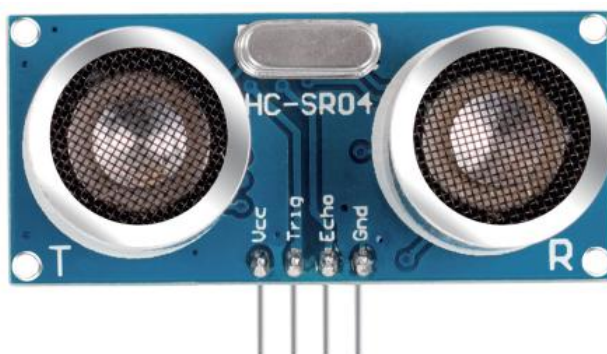
## 1 x 7 セグメントディスプレイ

7 セグメントディスプレイは、7 つの LED をパッケージ化した 8 の字型のコンポーネントです。各 LED はセグメントと呼ばれ、電源が供給されると、数字の一部を形成します。



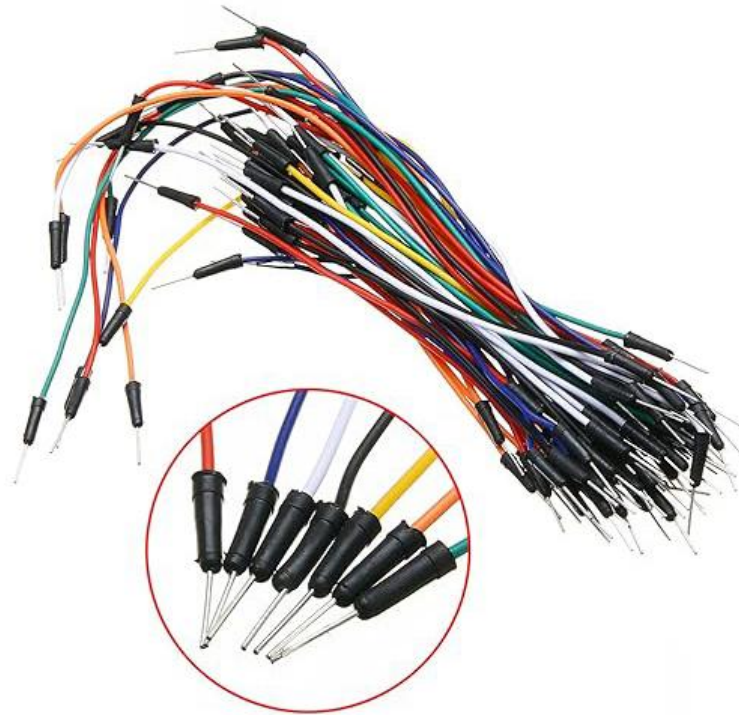
## 1 x 超音波モジュール

この超音波モジュールは、超音波を利用して距離を測定し、物体の位置や距離を正確に検出・測定します。ロボティクス、障害物回避システム、自動制御分野で広く使用され、環境認識と空間ナビゲーションの重要なコンポーネントです。



## 65 x ジャンパーワイヤー

ブレッドボード上のコンポーネントを互いに、また Arduino ボードに接続します。



## 10 x オス-メス DuPont ワイヤー

オス-メス DuPont ワイヤーは、超音波モジュールのようなオスピンヘッダーを持つモジュールをブレッドボードに接続するために特別に設計されています。これらのワイヤーは、電子プロジェクトでブレッドボード対応のオス-メス接続が必要な場合に不可欠です。





## 1 x USB ケーブル

Arduino ボードをコンピューターに接続します。プログラムの作成、コンパイル、転送を可能にし、ボードにも電源を供給します。



## 1 x 9V バッテリー

これは非充電式のアルカリ 9V バッテリーです。マルチメーターに取り付ける必要があります。



## 1 x マルチメーター (赤と黒のリード付き)

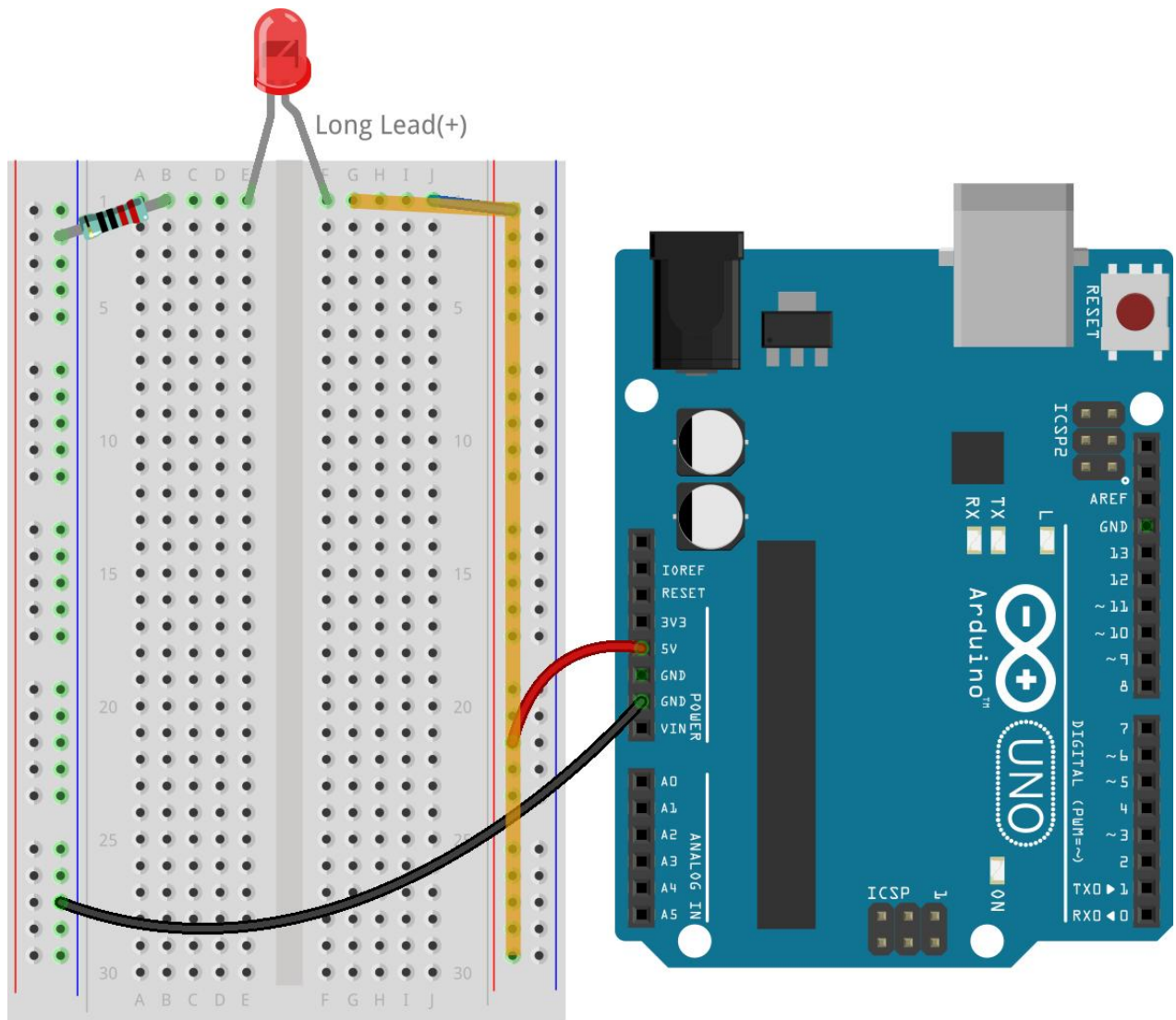
この多機能マルチメーターは、電圧、電流、抵抗を測定することができ、その他の電気テストも行えるため、電子機器や電気作業に欠かせないツールです。



## レッスン 2: 最初の回路

レッスンを終えた後、以下の質問に教えてください。

1. ブレッドボードから赤いワイヤーを取り外し、異なる穴に挿入して実験してください。LED の変化を観察し、LED が点灯する穴の位置をスケッチしてください。



2. LED のピンを逆にした場合、どうなりますか？ 点灯しますか？ なぜですか、またはなぜではありませんか？

LED は点灯しません。これは LED が一方向性の導電性を持っているためで、正しく機能するためにはアノードからカソードへ電流が流れる必要があります。

## レッスン 3: マルチメータの使い方を学ぶ

「マルチメータについてもっと知る!」を終えた後にこの質問に教えてください。

1. マルチメータの使用方法について詳しく理解した今、以下の電氣的値を測定するためにどのマルチメータ設定を使用しますか?

測定対象	マルチメータ設定
9V 直流電圧	20V
1K オーム	2k $\Omega$
40 ミリアンペア	200mA
110V 交流電圧	200V~

「マルチメータで測定する」中でこの表を記入してください

種類	単位	測定結果	ノート
電圧	ボルト	$\approx 5.13$ ボルトボルト	
電流	ミリアンペア	$\approx 13.54$ ミリアンペア	
抵抗	オーム	$\approx 378.88$ オーム	

## レッスン 4: オームの法則

「実験を通じてオームの法則を探る」中で以下の表を記入してください

1. 220 オームの抵抗器と他の異なる値の抵抗器を交換してください。各交換で LED の明るさの変化を記録し、抵抗が電流にどのように影響し、それが結果として光出力にどのように影響するかを観察してください。

抵抗器	観察結果
100Ω	明るい
1KΩ	明るい t
10KΩ	薄暗い
1MΩ	ほぼ消灯

220 オームの抵抗器よりも 100Ωの抵抗器で LED が明るいことがわかります。抵抗が高くなると、LED の明るさが減少し、1MΩでは完全に消灯します。これはなぜでしょうか？

オームの法則 ( $I = V/R$ ) によると、電圧が一定のままで抵抗が増加すると、LED を通る電流が減少し、LED が暗くなります。1MΩでは、電流が小さすぎて LED が点灯しません。

2. 抵抗を 220 オームに保ったまま、回路の電圧供給を 5V から 3.3V に変更します。LED の明るさの変化を記録してください。

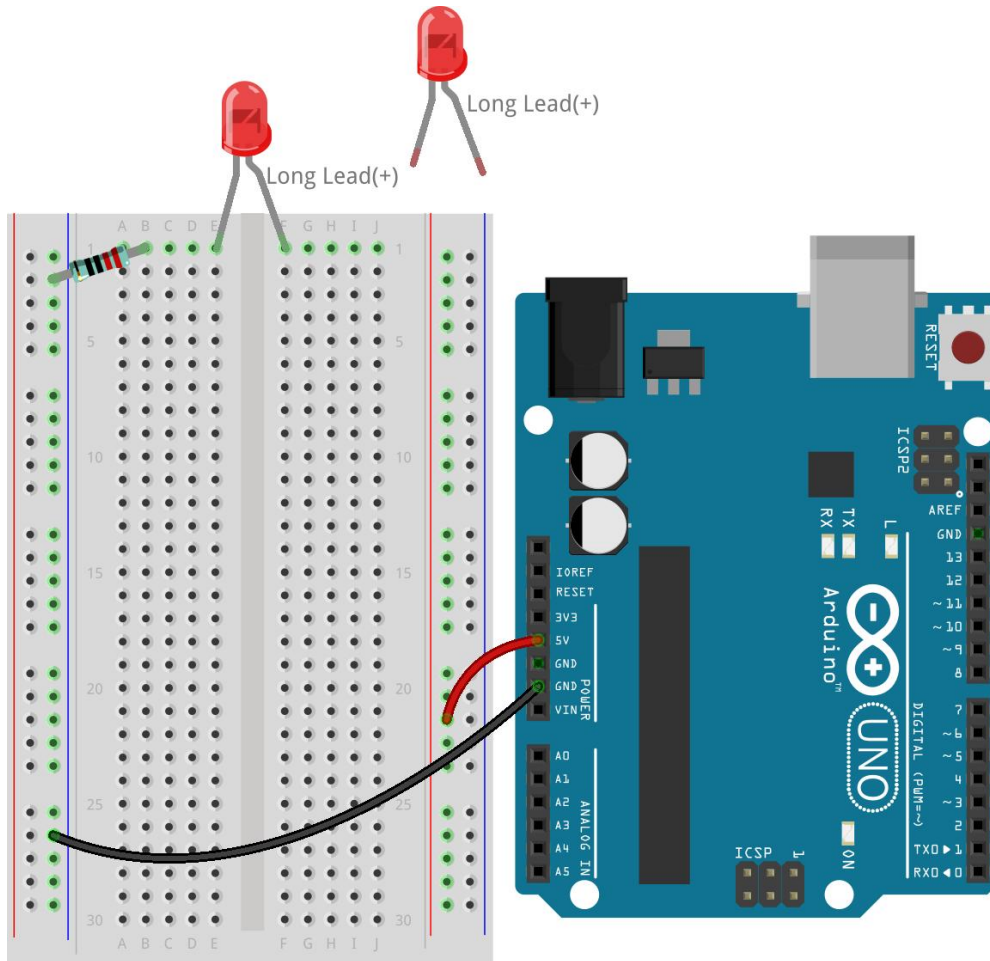
3.3V では 5V のときよりも LED が若干暗くなることがわかります。これはなぜでしょうか？

オームの法則を用いると、抵抗と新しい電圧がわかるため、電流は  $I = V/R$  となります。電圧が減少すると抵抗は同じままなので、電流が減少し、LED が暗くなります。

## レッスン 5: 直列回路と並列回路

「直列回路に飛び込む」中で以下の質問を完了してください

1. LED を 1 つ取り外すとどうなりますか？ これはなぜ発生しますか？



直列回路では、LED を 1 つ取り外すと、他の LED は点灯しません。これは直列回路では電流が経路のすべてのコンポーネントを通過しなければならないためです。LED を 1 つ取り外すと回路が切断され、残りの LED を通る電流が流れなくなります。

2. 直列回路の各コンポーネントの電圧を測定します。

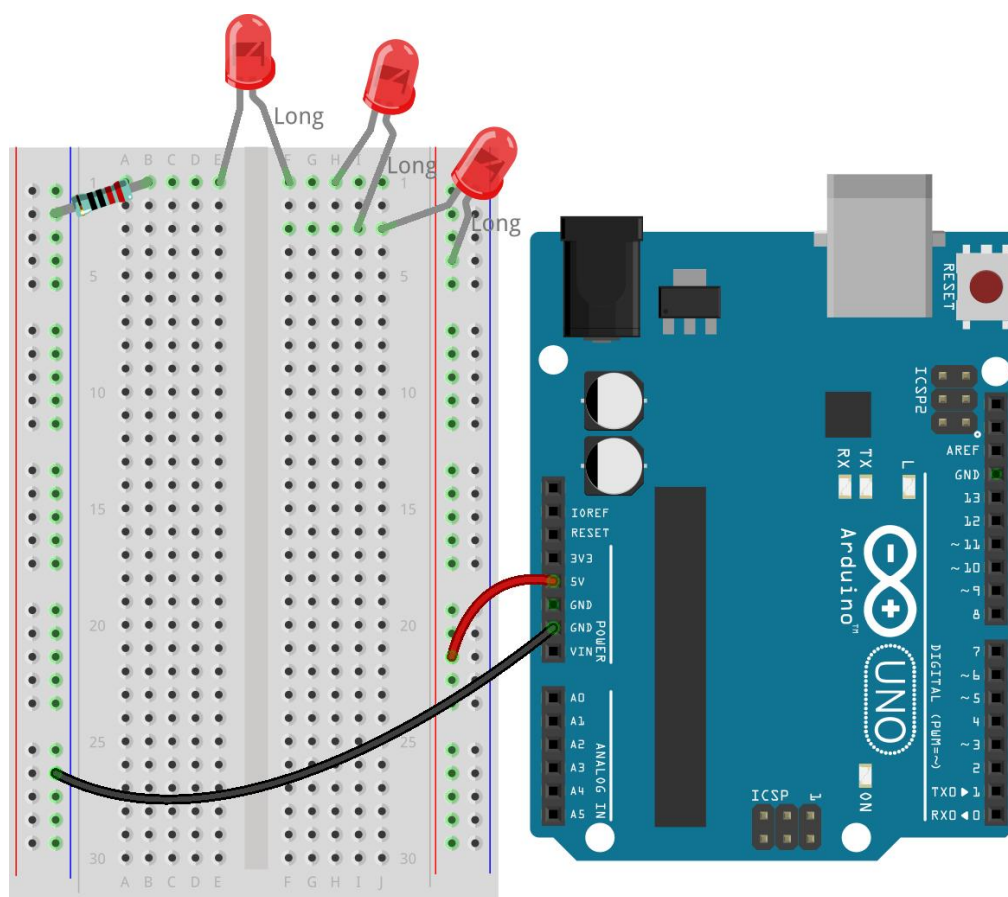
回路	抵抗器の電圧	LED1 の電圧	LED2 の電圧	合計電圧
2 つの LED	≈1.13 ボルト	≈1.92 ボルト	≈1.92 ボルト	≈4.97 ボルト



3. 直列回路の各コンポーネントの電流を測定します。.

回路	LED1 の電流	LED2 の電流
2 つの LED	$\approx 4.43$ ミリアンペア	$\approx 4.43$ ミリアンペア

4. この回路に別の LED を追加し、3 つの LED になった場合、LED の明るさはどのように変化しますか？ なぜですか？ 3 つの LED の電圧はどのように変化しますか？

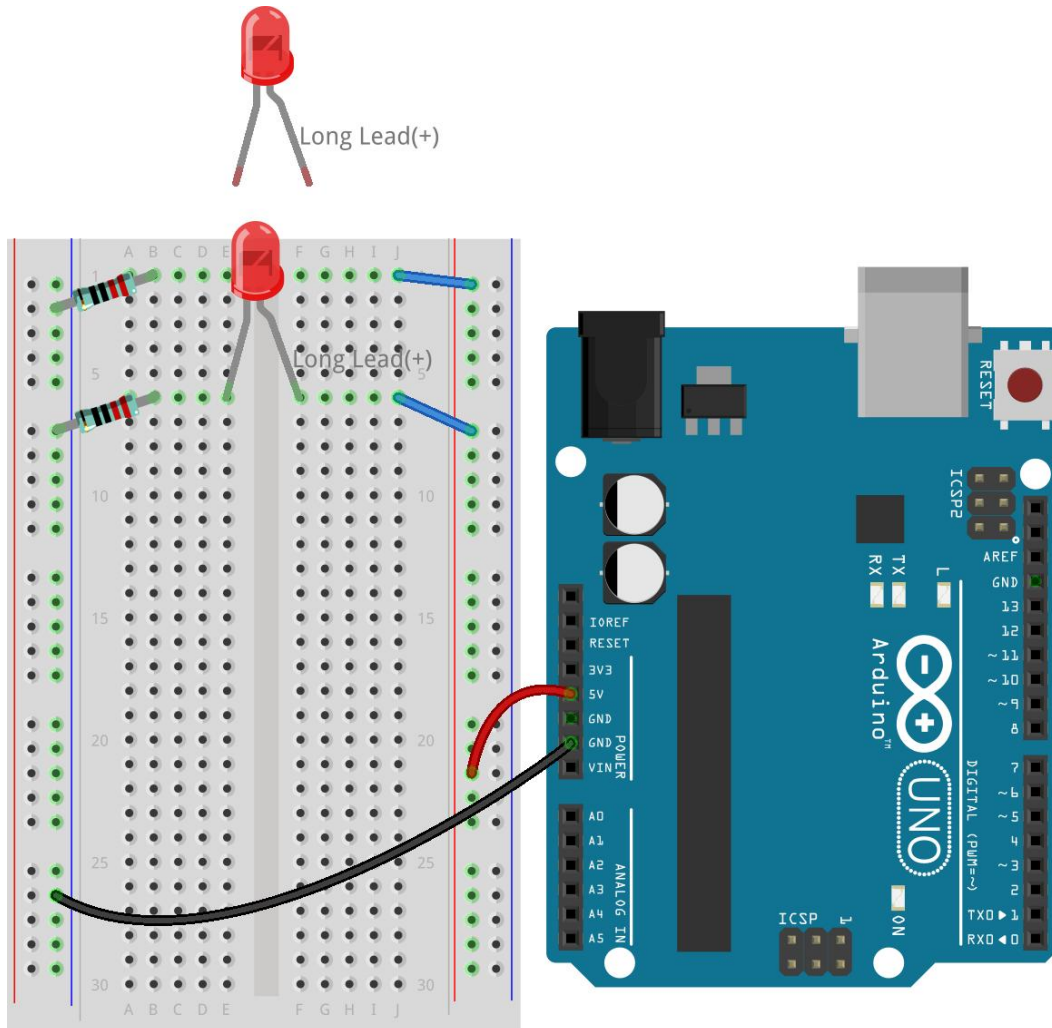


すでに 2 つの LED がある直列回路に別の LED を追加すると、各 LED の明るさは一般的に減少します。これは、電源の全電圧がより多くのコンポーネント間で分割されるため、以前の 2 つの LED のときよりも各 LED にかかる電圧が低下するためです。結果として、各 LED を通る電流が減少し、その明るさも減少します。

3 つの LED にかかる電圧については、各 LED が回路の全電圧の一部を受け取ります。電源の電圧が同じである場合、この電圧は 3 で割られます（すべての LED が同様の電気特性を持っていると仮定）。したがって、回路内の各 LED にかかる電圧は、電源から提供される全電圧の約 3 分の 1 になります。

「並列回路に飛び込む」中で以下の質問を完了してください。

1. この並列回路で LED を 1 つ取り外すとどうなりますか？ これはなぜ発生しますか？



並列回路では、LED を 1 つ取り外しても、回路の他の LED は点灯し続けます。これは、並列回路の各 LED が電源に対して独自の独立した経路を持っているためです。LED を 1 つ取り外しても、他の LED への電流の流れは中断されず、影響を受けずに正常に動作し続けます。この設定により、並列回路の各コンポーネントは他のコンポーネントから独立して動作できます。

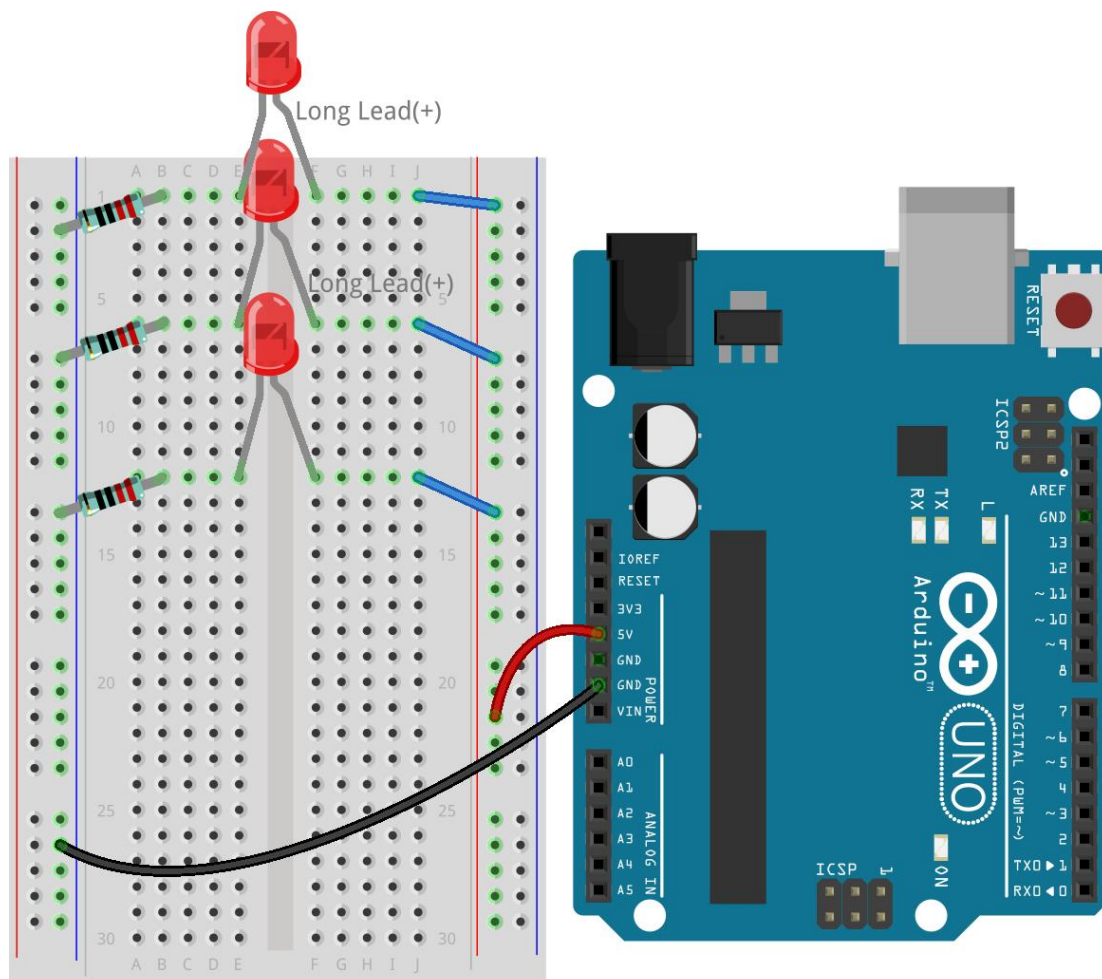
2. 測定した電圧を表に記録します。

回路	Path1 の電圧	Path2 の電圧
2 LEDs	≈5.00 ボルト	≈5.00 ボルト

3. 表に測定した電流を記録します。

回路	LED1 の電流	LED2 の電流	合計の電流
2 LEDs	≈12.6 ミリアンペア	≈12.6 ミリアンペア	≈25.3 ミリアンペア

4. この回路に別の LED を追加した場合、LED の明るさはどうなりますか？ なぜですか？ その答えを手帳に記録してください。



別の LED を並列回路に追加した場合、既存の LED の明るさは通常変わりません。これは、並列回路の各 LED が電源に直接つながる自身の経路を持っているためで、追加される LED の数に関わらず各 LED にかかる電圧は一定です。各 LED は、意図した明るさで動作するために必要な全電圧を受け取ります。したがって、電源が回路の合計電流需要を支えられる場合、より多くの LED を追加しても既にある LED の明るさには影響しません。将来の参照のためにこの情報を手帳に記録しておいてください。

## レッスン 6: LED の点滅

「LED を生かす」中で以下の表を記入してください。

1. Pin 3 の測定された電圧を表に記録します。

State 状態	Pin 3 の電圧
HIGH	≈4.95 ボルト
LOW	0.00 ボルト

レッスンを終えた後、以下の質問に答えてください。

1. 上記のコードをアップロードすると、LED が 3 秒間隔で繰り返し点滅します。LED を一度だけオン・オフしたい場合、どうすればよいですか？

LED をオン・オフするコマンドを `loop()`関数から `setup()`関数に移動します。`setup()`関数はプログラムが開始されるときに一度だけ実行されるので、この変更により LED が一度だけ点灯し、消灯します。以下のようにコードを調整できます：

```
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:  
}
```

## レッスン 7: 交通信号を作ろう!

「交通信号の疑似コードを書く」中で以下の質問を完了してください。

自分の回路が交通信号のように機能するために必要なことを考えてください。ログの提供されたスペースに、交通信号がどのように機能するかを説明する疑似コードを平易な英語で書き留めてください。

回路が信号機のように機能するために必要なことを考えてみましょう。ログのスペースに、信号機がどのように動作するかを説明する疑似コードを書きましょう。

Arduino を使って交通信号をシミュレートするには、赤、黄、緑の 3 つの LED と、実際の交通信号を模倣する制御シーケンスが必要です。この交通信号回路がどのように機能するかを説明する簡単な疑似コードのアウトラインをログに記述できます:

セットアップ:

- 赤、黄、緑の LED のピンを定義します。
- これらのピンを出力として設定します。

メインループ:

- 赤 LED を 5 秒間点灯させます。
- 赤 LED を消灯します。
- 黄 LED を 2 秒間点灯させます。
- 黄 LED を消灯します。
- 緑 LED を 5 秒間点灯させます。
- 緑 LED を消灯します。
- このサイクルを繰り返します。

「交通信号の疑似コードを書く」中で以下の質問を完了してください。

自宅周辺の交差点を見てみてください。通常、どのくらいの交通信号がありますか？ それらはどのように調整されていますか？

自宅周辺の交差点を見てみてください。通常、どのくらいの交通信号がありますか？ それらはどのように調整されていますか？

都市部の交差点には、通常、車両と歩行者の流れを効率的に管理するために交通信号が設置されています。交差点のサイズや複雑さに応じて、交差点にある交通信号の数は大きく異なることがあります。単純な四方交差点では、通常、各方向の交通に向けて少なくとも四つの交通信号が設置されています。より複雑な交差点では、右折レーン、歩行者横断、その他の交通管理のための追加の信号が必要になることがあります。



## レッスン 8: 歩行者ボタン付き信号機

「回路の構築」を完了した後、以下の質問に教えてください。

- あなたの信号機は直列と並列の回路が混在しています。どの部分が直列で、どの部分が並列になっているか説明してください。

回路では、ボタンとその 10K プルダウン抵抗が直列に接続されています。この設定により、ボタンが押されるとピン 8 の状態が正しく変更され、押されていないときは直接グランドに接続され、浮遊入力を防ぎます。

ピン 3、4、5 に接続された 3 つの LED は互いに並列に接続されています。各 LED は独立して動作します。これは、それぞれが別々の制御ピンに接続され、共通の電源を共有しているためです。この設定により、各 LED が他の LED に影響を与えることなく機能し、これは信号機システムにとって重要です。

「コード作成」中にこの表を記入してください。

- ボタンが押されたときと押されていないときのピン 8 の測定された電圧で表を埋め、対応する高レベルと低レベルの状態を記入してください。

ボタンの状態	Pin 8 の電圧	Pin 8 State
離す	0.00 ボルト	LOW
押す	≈4.97 ボルト	HIGH

このレッスンを終えた後、次の質問を完了してください。

- テスト中、歩行者ボタンを押し続けると緑の LED が点滅することに気づくかもしれませんが、歩行者はボタンを続けて押しながら道を渡ることはできません。歩行者ボタンを押した後、緑の LED が安全な横断のために十分な時間点灯するようにコードをどのように変更できますか？その疑似コードの解決策を手帳に書き留めてください。

歩行者用の緑の LED が連続してボタンを押す必要なく点灯し、その後通常の信号機サイクルを続けるように、ボタンの押下をチェックし、その押下に基づいて動作状態を変更する疑似コードを調整できます。ここに、これらの変更を反映した最適化されたより明確な疑似コードのバージョンがあります：

セットアップ:

- 赤、黄、緑の LED 用のピンを出力として定義

- ボタンピンを入力として定義

メインループ:

- ボタンが押されているか確認

- ボタンが押された場合:

  - すべての LED を消灯

  - 歩行者用に緑の LED を点灯

  - 10 秒間待つ

それ以外の場合:

- 通常の信号機サイクルを実行:

  - 緑の LED (車両用) を点灯し、他の LED を消灯

  - 10 秒間待つ

  - 黄色の LED を点灯し、他の LED を消灯

  - 3 秒間待つ

  - 赤の LED を点灯し、他の LED を消灯

  - 10 秒間待つ

## レッスン 9: 調光可能なデスクランプ

「回路の構築」中にこの表を記入してください。

1. ポテンショメータを時計回りに位置 1 から 3 まで回し、各点での抵抗を測定し、結果を表に記録してください。

測定点	抵抗値 (キロオーム)
1	1.52
2	5.48
3	9.01

2. ポテンショメータを時計回りおよび反時計回りに回したとき、A0 の電圧はどのように変化するとおもいますか？

ポテンショメータは、回路内で直列に接続された 2 つの抵抗器から成ると考えることができます。抵抗の測定によると、ポテンショメータを時計回りに回すと、A0 と GND の間の抵抗が増加します。直列回路では電流が一定であるため、オームの法則 (電圧 = 電流 × 抵抗) により、抵抗の増加は A0 の電圧の増加につながります。したがって、ポテンショメータを時計回りに回すと A0 の電圧が増加し、反時計回りに回すと抵抗が減少するため電圧が減少します。

このレッスンを完了した後、次の質問を完了してください。

1. LED を異なるピン、例えばピン 8 に接続し、ポテンショメータを回した場合、LED の明るさは変わりますか？ なぜですか、またはなぜ変わりませんか？

Arduino UNO のピン 8 に LED を接続し、ポテンショメータを回しても、LED の明るさは変わりません。これは、ピン 8 が PWM (パルス幅変調) をサポートしていないためです。明るさレベルを調整するために analogWrite() 関数を使用するには、PWM が必要です。Arduino UNO では、PWM をサポートし、analogWrite() を通じて LED の明るさを制御できるピンはピン 3、5、6、9、10、および 11 です。

## レッスン 10: ON/OFF デスクランプ

このレッスンを完了した後、以下の質問を完了してください。

---

1. デジタルピン 7 を INPUT のみに設定した場合、どうなりますか？ なぜですか？

```
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  
}
```

Arduino スケッチでデジタルピン 7 を INPUT モードに設定すると、INPUT\_PULLUP とは異なり、ピンから読み取った信号に潜在的な不安定さが生じる可能性があります。ピンが INPUT のみで設定され、外部回路を通じて明確な高または低の電圧に接続されていない場合、そのピンは「浮遊」と呼ばれる状態になります。浮遊ピンは安定した高または低の状態ではなく、その状態は環境の電気ノイズや干渉によって変動する可能性があります。この変動は、デジタル入力関数を介してピンの状態を読み取るときに予測不能な読み取りを引き起こし、マイクロコントローラーに誤ったまたは一貫性のないデータが受信される原因となる可能性があります。

2. ピン 7 が INPUT のみに設定されている場合、回路にどのような調整が必要ですか？

Arduino のピン 7 を INPUT モードに設定し、安定して予測可能な読み取りを得るためには、回路に外部プルアップ抵抗を追加する必要があります。これには、ピン 7 と Arduino の 5V 電源の間に 10kΩ の抵抗を接続することを含みます。プルアップ抵抗は、他の入力信号が存在しない場合に入力ピンが高状態（ロジックレベル 1）になることを保証します。

# レッスン 11: ポテンショメーターで LED アレイを制御

「コード作成」に進む前に、以下の質問に答えてください。

---

1. LED アレイの疑似コードを書いてください。

疑似コードはプログラムのスケッチであり、理解を容易にするために平易な言葉で書かれます。ポテンショメーターに反応する LED アレイの疑似コードを作成するのがあなたの課題です。ポテンショメーターの値が増加すると、より多くの LED が点灯します。

こちらはポテンショメーターの入力に基づいて LED アレイを制御するための簡略化された疑似コードです：

変数 `readValue` を宣言する。

セットアップ。

3 つのデジタルピン出力を宣言する。

メインループ。

ポテンショメーターの値が 200 未満の場合、すべての LED をオフにする。

値が 200 から 600 の間の場合、最初の LED をオンにする。

値が 600 から 1000 の間の場合、最初の 2 つの LED をオンにする。

値が 1000 を超える場合、すべての LED をオンにする。

短時間遅延する。

レッスンを完了した後、次の質問に答えてください。

---

1. 最後のコードでは、ポテンショメーターの値に基づいて点灯する LED の数を決定します。ポテンショメーターに応じて、LED の明るさが変化するようにコードをどのように変更できますか？

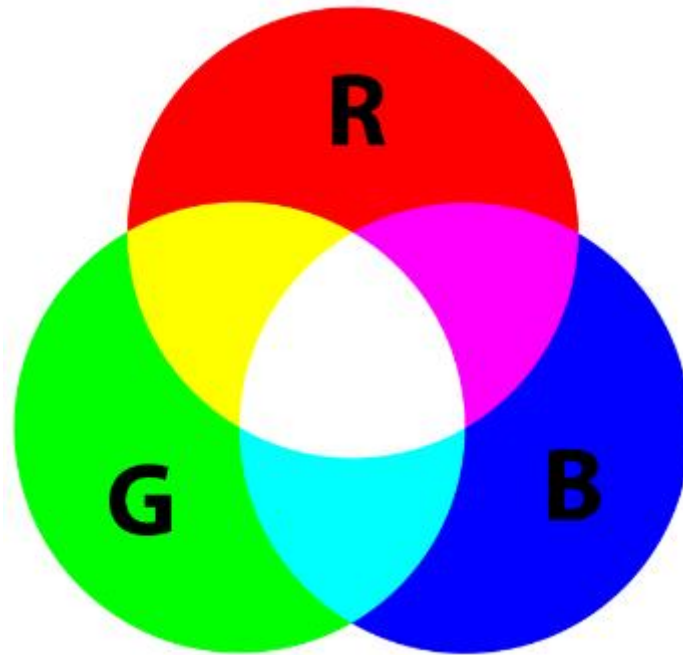
LED の明るさをポテンショメーターの値に応じて変更するために、`digitalWrite()` の代わりに `analogWrite()` 関数を使用することができます。`analogWrite()` 関数を使用すると、PWM (パルス幅変調) を通じて LED の明るさを制御できます。



## レッスン 12: 虹の色

「コード作成」中にこの表を記入してください。

1. 他の色が欲しい場合はどうすればいいですか？ 以下の図を参照して、あなたのアイデアを手帳に記入してください。



色	赤ピン	緑ピン	青ピン
赤	<i>HIGH</i>	<i>LOW</i>	<i>LOW</i>
緑	<i>LOW</i>	<i>HIGH</i>	<i>LOW</i>
青	<i>LOW</i>	<i>LOW</i>	<i>HIGH</i>
黄	<i>HIGH</i>	<i>HIGH</i>	<i>LOW</i>
ピンク	<i>HIGH</i>	<i>LOW</i>	<i>HIGH</i>
シアン	<i>LOW</i>	<i>HIGH</i>	<i>HIGH</i>
白	<i>HIGH</i>	<i>HIGH</i>	<i>HIGH</i>

## レッスン 13: 視覚のスペクトル

「コード作成」中にこの表を記入してください。

1. これでピン 9、10、11 の値を個別に調整でき、観察された色を手帳に記録できます。

赤ピン	緑ピン	青ピン	色
0	128	128	濃い青
128	0	255	紫
128	128	255	明るい青
255	128	0	オレンジ

2. お気に入りの色をいくつか選び、その RGB 値で表を記入してください。

色	赤	緑	青

## レッスン 14: ランダムな色

レッスンを終えた後、次の質問に教えてください。

---

1. コードを `randomSeed(analogRead(A0))` から `randomSeed(0)` に変更すると、RGB LED の色はどのように変化し、その理由は何ですか？

`randomSeed(analogRead(A0))` から `randomSeed(0)` にコードを変更すると、RGB LED の色のランダム性に影響が出ます。`randomSeed(seed)` 関数は、Arduino で疑似ランダム数生成器を初期化するために使用され、`random()` のような関数によって生成されるランダム数の系列に影響を与えます。

シード値として `analogRead(A0)` を使用すると、環境ノイズなどの要因により多少ランダムな値がアナログピン A0 から読み取られ、プログラムが開始されるたびに異なるシード値が生成されます。これは、Arduino がリセットされるたびにランダム数（そして色）の系列が変わることを意味します。

一方で、`randomSeed(0)` のような固定値でシードを設定すると、プログラムが実行されるたびにランダム数生成器が同じ開始点で初期化されます。これにより、ランダム数の同じ系列が生成され、その結果、RGB LED はリセットまたは電源を入れるたびに同じ色パターンを示します。これにより、LED の色変更のランダム性がなくなります。

2. デコレーションの色をランダムに選んだり、宝くじの番号を選んだりする以外で、日常生活でランダム性が問題解決に使用される状況は何ですか？

ランダム性は、日常のさまざまな文脈で問題を解決するために使用されます。これには以下のようなものがあります：

- ボードゲーム：サイコロを振って動きを決め、各ゲームが異なり公平であることを保証します。
- 音楽プレイリスト：曲をシャッフルして、聴く経験を新鮮で予測不可能に保ちます。
- 食事選択：何を食べるかわからないときに、レストランや食事をランダムに選ぶことで、意思決定を容易で楽しいものにします。
- 席次の配置：イベントで席をランダムに抽選し、ゲストを混ぜて社交を促進します。
- 映画の夜：それぞれが異なる好みを持つとき、ランダムな抽選で映画を選びます。

## レッスン 15: クールまたはウォームな色

「コード作成」中にこの表を記入してください。

1. Paint や他の色選択ツールを開き、最も暖かく感じる色と最もクールな色を見つけ、その RGB 値を手帳に記録してください。

色のタイプ	赤	緑	青
ウォームな色	246	52	8
クールな色	100	150	255

レッスンを終えた後、次の質問に答えてください。

1. 「下限」と「上限」の範囲が逆転することがあるため、map() 関数を使用して数値の範囲を逆にすることがあります。例えば、

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

この関数は負の数にもうまく対応しており、この例も有効でうまく機能します。

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

x が 20 の場合、y はどうなるべきですか？ 以下の式を参照して計算してください。

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

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

x=20 を使用したマッピング式  $y = \text{map}(x, 1, 50, 50, -100)$  によると、y の値は約 -8.16 になります。

## レッスン 16: 温度アラーム

「回路の構築」中にこの表を記入してください。

1. 異なる温度下での抵抗値を読み取り、以下の表に記録してください。

環境	抵抗値 (キロオーム)
現在の温度	9.37
高温	6.10
低温	12.49

レッスンを終えた後、次の質問に答えてください。

1. コードでケルビンとセルシウスの温度が計算されています。華氏温度も知りたい場合、どうすればよいですか？

これはセルシウスから華氏への変換の標準的な方法であり、計算から得られたセルシウス値を基に華氏温度を求めることができます。

$$F = C * 1.8 + 32$$

2. Can you think of other situations or places where a temperature monitoring system like the one we built today could be useful? 今日構築した温度監視システムが役立つ他の状況や場所について考えてみてください。

温度監視システムは、日常的な状況やさまざまな環境で広く適用されます。以下はいくつかの簡略化された例です：

- ホームコンフォート：リアルタイムの温度読み取りに基づいて自宅の暖房や冷房を自動調整し、居住空間を快適に保ちます。
- ガーデニング：温室の温度を監視し、植物が最適な条件で成長していることを確認します。必要に応じて温度を調整する自動システムを追加します。
- 食品安全：特にレストランや食品輸送中に、冷蔵庫や冷凍庫の温度を監視して、食品が安全に食べられる状態に保ちます。
- ヘルスケア：温度に敏感な薬やワクチンの保管エリアの温度を監視し、記録し、効果が維持されるようにします。



## レッスン 17: モールス信号

「回路の構築」中に次の質問に教えてください。

---

1. アクティブブザーのカソードを GND に直接接続し、アノードを 5V に接続した場合、どうなりますか？ その理由は？

アクティブブザーのカソードを GND に直接接続し、アノードを 5V に接続すると、ブザーは連続音を発します。これは、ブザー内の内部オシレータが 5V 電源によって活性化され、回路が切断されるまで音を生成するためです。

レッスンを終えた後、次の質問に教えてください。

---

1. 提供されたモールス信号表を使用して、「Hello」というメッセージを送信するコードを書いてください。

モールス信号では、「Hello」は以下の文字に基づいて次のように符号化されます：

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

組み合わせると、「Hello」のモールスコードは次の通りです：

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

実際の通信では、通常、単語間を明確に区別するために単語間により長い休止がありますが、「Hello」は単一の単語なので、個々の文字を区切るスペースのみでコードが連続します。

## レッスン 18: 光アラーム

「回路の構築」中にこの表を記入してください

1. 現在の周囲光の下での抵抗値を読み取り、以下の表に記録してください。

環境	抵抗値 (キロオーム)
普通の光	≈5.48
明るい光	≈0.16
暗闇	≈1954

レッスンを終えた後、次の質問に答えてください。

1. 狡猾な泥棒が夜に盗みを行うことを選ぶかもしれませんが、絵画がなくなると、光抵抗器が光の変化を検出できず、アラームが作動しない可能性があります。この欠点を改善するにはどうすればよいでしょうか？

一つの解決策は、絵画の前に光源を設置することです。これにより、絵画がはっきりと見え、照らされるだけでなく、絵画の攪乱や取り外しによって即座に光抵抗器のアラームが光レベルの変化を検出して作動するようになります。

## レッスン 19: リバースパークینگアラームシステム

レッスンを終えた後、次の質問に教えてください。

---

1. このデバイスによって検出された距離を小数点以下までより正確にしたい場合、コードをどのように変更すべきですか？

距離測定を小数点以下まで正確にするために、measureDistance 関数の計算を整数算術から浮動小数点算術に変更できます。これにより、結果に小数点以下の値が含まれ、より正確な測定が可能になります。

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

measureDistance 関数の distance 変数を float に変更して小数点以下の値をサポートします。

距離の計算には浮動小数点算術を使用し、2 ではなく 2.0 で割ります。

## レッスン 20: ポモドーロタイマー

「コード作成 - millis()」中に次の質問に教えてください。

1. delay(100)を delay(1000)に変更すると、プログラムにどのような影響がありますか？ その理由は？

```
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
}
```

元のコードでは、ブザーが 1000 ミリ秒（1 秒間隔で設定された間隔変数）ごとに約 100 ミリ秒鳴り、その後 900 ミリ秒の沈黙があります。修正後、ブザーは 1000 ミリ秒毎に 1000 ミリ秒鳴り、次の間隔がほぼすぐに始まるため、ほぼ直ちに再び鳴ります。したがって、delay を 100 ミリ秒から 1000 ミリ秒に変更すると、ブザーが短いビーブ音から連続音に変わり、元の目的には不適切でうるさいものになります。

コードで delay(100)を delay(1000)に変更すると、ブザーが短いビーブではなく 1 秒間鳴るようになり、ブザーがオンのときの一時停止時間が長くなります。これにより、ブザーの音が長くなり、プログラムサイクルが少なくなり、これらの間隔中に他のタスクに対するプログラムの応答が潜在的に低下する可能性があります。

レッスンを終えた後、次の質問に教えてください。

あなたの生活の中で「時間を聞く」他の場所を考えてみてください。いくつかの例をリストアップし、手帳に書き留めてください！

- 時間を聞くというのは興味深い概念であり、日常のいくつかのシナリオでこれを経験することができます。以下はメモする価値があるかもしれないいくつかの例です：
- 時計と腕時計：アナログ時計のカチカチ音や、デジタル時計からの特定のビーブ音が、経過する秒数や分数を示します。
- キッチンタイマー：料理や焼き時間をカウントダウンする機械式またはデジタルキッチンタイマーのカチカチ音と最終アラーム。
- 学校のベル：授業や休憩の開始と終了を知らせる学校のベルの鳴り響き。

- 公共交通機関のアナウンス：列車駅やバス内でアナウンスに先立って鳴るビープ音やチャイムが、まもなく発車または到着を示します。
- 電子レンジ：タイマーが終了するときのビープ音が、加熱プロセスが完了したことを知らせます。
- フィットネストラッカーやスポーツウォッチ：ワークアウト中または間隔で設定された時間の完了を示すビープ音やアラーム。

## レッスン 21: サイレンサウンド

「回路の構築」中に次の質問に教えてください。

---

1. 受動ブザーのカソードを GND に直接接続し、アノードを 5V に接続した場合、何が起こりますか？ その理由は？

受動ブザーのカソードを GND に直接接続し、アノードを 5V に接続すると、アクティブブザーとは異なり、受動ブザーは自身では音を出しません。これは内蔵の発振器がないためです。

受動ブザーは外部の信号が必要です。通常、聞こえる音を作り出すために、所望の周波数で四角波（振動す電圧）で駆動する必要があります。

「コード作成 - 受動ブザーを鳴らす」中に次の質問に教えてください。

---

1. コードと回路のピンを PWM ピンではない 7 または 8 に変更した場合、ブザーはまだ音を出しますか？ テストしてから手帳に回答を書いてください。

ピン 8 は PWM ピンではありませんが、`tone()`関数はそれに正確な四角波を生成することができます。この柔軟性により、PWM 対応ピンに限定されることなく、任意のデジタルピンを音声出力に使用できます。`tone(pin, frequency)`関数を呼び出すと、Arduino は指定された周波数でピン状態をトグル（HIGH から LOW へ、そして HIGH へ戻る）するタイマーを設定し、四角波を作り出します。この四角波が受動ブザーを駆動し、生成された波の周波数で音を出します。



2. tone(pin, frequency, duration)の周波数と持続時間がブザーの音にどのように影響するかを  
探るために、2つの条件下でコードを変更し、観察した現象を手帳に記入してください:

- 周波数を 1000 のままで、持続時間を 100、500、1000 に徐々に増やします。ブザーの音はどのように変化し、その理由は何ですか？

- 100 ms 持続時間: 音は短いビーブ音です。
- 500 ms 持続時間: 音は長く、はっきりと聞こえ、半秒続きます。
- 1000 ms 持続時間: 音はさらに長く、1 秒間続きます。

持続時間を増やすと、ブザーから出る音の長さが長くなります。音のピッチまたは周波数は一定です (1000 Hz に設定されているため)、音の「ノート」は変わりませんが、聞こえる時間の長さが増えます。これは、緊急性またはアラートの種類を音の長さで区別できる異なる持続時間のアラートを知らせるのに役立ちます。

- 持続時間を 100 のままで、周波数を 1000、2000、5000 に徐々に増やします。ブザーの音はどのように変化し、その理由は何ですか？

- 1000 Hz 周波数: 音は中音域のビーブ音です。
- 2000 Hz 周波数: 1000 Hz に比べて高いピッチの音です。
- 5000 Hz 周波数: 音はさらに高いピッチで、近くで聞くと鋭く、おそらく不快に感じられる可能性があります。

持続時間を一定に保ちながら周波数を増やすと、音のピッチが変わります。高い周波数は高いピッチの音を生じます。この原理は、緊急性または重要度に基づいて異なるタイプの通知や信号を区別するのに役立ち、より緊急なアラートにはしばしば高いピッチが使用されます。

## レッスン 22: 「きらきら星」を演奏する

「コード作成 - 配列」中に次の質問に教えてください。

---

1. 配列の要素に対して、`Serial.println(melody[i] * 1.3)`に変更するとどうなりますか？ どのようなデータが得られるか、その理由は？

1.3 は浮動小数点数です。int 型のメロディ配列の整数が 1.3 で乗算されると、演算結果は自動的に浮動小数点数 (float) に昇格します。

この配列の各ノートの周波数に 1.3 を乗算して結果を出力すると、以下のようになります：

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

レッスンを終えた後、次の質問に教えてください。

---

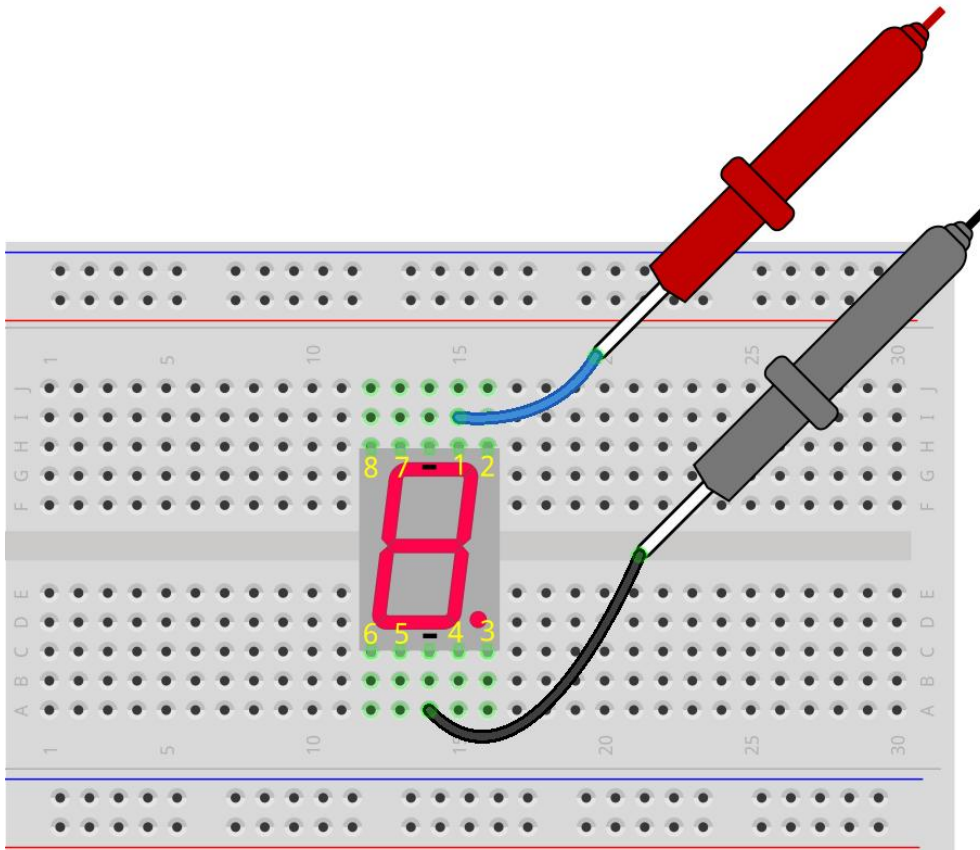
1. 回路で受動ブザーをアクティブブザーに置き換えると、「きらきら星」を正確に演奏できるでしょうか？ その理由は？

「きらきら星」を演奏するために受動ブザーをアクティブブザーに置き換えると、意図した通りには機能しません。アクティブブザーは内蔵オシレーターを持っているため、一つの音しか出せません。したがって、メロディーを正確に演奏するためのピッチを制御することはできず、曲のリズムに合わせて繰り返しビープ音がするだけで、実際の音符は聞こえません。

## レッスン 23: サイバーダイス

「7 セグメントディスプレイの理解」中に次の質問に教えてください。

1. セグメントが点灯する場合、この図を参照してセグメントのピン番号とおおよその位置をハンドブックの表に記録してください。



ピン	セグメント番号	位置
1	A	上部
2	B	右上
3	C	右下
4	D	下部
5	E	左下
6	F	左上
7	G	中央
8	小数点	ドット

2. 上記のテストから、キットのディスプレイがコモンカソードであることがわかります。これは、コモンピンを GND に接続し、他のピンに高電圧を供給するだけで対応するセグメントを点灯させる必要があることを意味します。ディスプレイに数字の 2 を表示させたい場合、どのピンに高電圧を供給すべきですか？ その理由は？



数字の 2 を表示するためには、セグメント a、b、d、e、および g をアクティブ化（高電圧を設定）する必要があります。これらはディスプレイ上で数字 2 を形成するセグメントです。f、c、および dp（小数点がある場合）のセグメントはオフのまま（低電圧）でなければなりません。これらは数字 2 のディスプレイの一部ではありません。

したがって、数字 2 を正しく表示するために高電圧を供給すべきピンは、a、b、d、e、および g セグメントに接続されているピンです。

## レッスン 24: 74HC595 を使った流れる光

「コード作成 - LED の点灯」中に次の質問に教えてください。

---

1. `shiftOut(DS, SHcp, MSBFIRST, B11101110)`で `MSBFIRST` を `LSBFIRST` に変更した場合はどうなりますか？ その理由は？

`MSBFIRST` を `LSBFIRST` に変更すると、ビットの順序が逆転し、バイトが最下位ビット（最も右側）からシフトアウトされるようになります。シフトレジスタを使用して LED を制御している場合、ビットの順序を変更すると、LED が点灯する順序が逆転します。もともとプログラムされた順序で点灯する代わりに、逆の順序で点灯します。

レッスンを終えた後、次の質問に教えてください。

---

1. 一度に 3 つの LED を点灯させ、それらが「流れる」ように見える場合、`dataArray[]`配列の要素はどのように変更すべきですか？

最初に 1、2、3 の LED を点灯させ、その後の各パターンで右に 1 つの LED を移動させ、最後に 6、7、8 の LED が点灯するまで続けます。ここで、これらのパターンをバイナリで定義する方法です：

B11100000: 1、2、3 の LED が点灯；他はオフ。

B01110000: 2、3、4 の LED が点灯；他はオフ。

B00111000: 3、4、5 の LED が点灯；他はオフ。

B00011100: 4、5、6 の LED が点灯；他はオフ。

B00001110: 5、6、7 の LED が点灯；他はオフ。

B00000111: 6、7、8 の LED が点灯；他はオフ。

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

## レッスン 25: 番号の表示

「0 から 9 までの数字のバイナリ数値」中にこの表を記入してください。

1. 数字 0 と 2 のバイナリ表現がわかったので、以下の表に残りの数字のバイナリ数を記入してください。

数字	バイナリ
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>

「バイナリ変換」中にこの表を記入してください。

1. 数字 0 から 9 を表すバイナリ数を電卓を使用して 10 進数と 16 進数に変換し、表に記入してください。これにより、基数変換のためのクイックリファレンスガイドが得られます。

数字	バイナリ	10 進数	16 進数
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>