
SunFounder 3in1 Kit

www.sunfounder.com

2024 年 05 月 21 日

目次

第 1 章	キットのコンポーネントについて学ぶ	3
1.1	Arduino Uno R4 Minima	4
1.2	ESP8266 モジュール	8
1.3	ブレッドボード	12
1.4	抵抗器	13
1.5	コンデンサー	16
1.6	ジャンパーワイヤー	18
1.7	74HC595	19
1.8	LED	21
1.9	RGB LED	23
1.10	7 セグメントディスプレイ	25
1.11	I2C LCD1602	28
1.12	ブザー	30
1.13	TT モーター	32
1.14	サーボ	34
1.15	遠心ポンプ	36
1.16	L9110 モータードライバモジュール	38
1.17	ボタン	41
1.18	リードスイッチ	43
1.19	ポテンショメーター	44
1.20	ジョイスティックモジュール	46
1.21	IR 受信機	48
1.22	フォトレジスタ	50
1.23	サーミスタ	52
1.24	DHT11 温湿度センサ	54
1.25	ライントラッキングモジュール	55
1.26	土壌湿度モジュール	57
1.27	障害物回避モジュール	60
1.28	超音波モジュール	61
第 2 章	Arduino のスタートガイド	65
2.1	Arduino とは何か?	65

2.2	Arduino で何ができるか?	66
2.3	Arduino プロジェクトの構築方法	66
第 3 章	コードをダウンロード	105
第 4 章	基本プロジェクト	107
4.1	1. デジタルライト	108
4.2	2. アナログライト	122
4.3	3. デジタルリード	131
4.4	4. アナログ読取り	148
4.5	5. さらなる構文	166
4.6	6. 面白いプロジェクト	249
第 5 章	カープロジェクト	283
5.1	車の組み立て	284
5.2	1. 移動	296
5.3	2. コードによる移動	302
5.4	3. 速度の上昇	306
5.5	4. ライン追従	308
5.6	5. 障害物回避モジュールを使って遊ぼう	313
5.7	6. 超音波モジュールで遊ぼう	318
5.8	7. 手を追う車	321
5.9	8. 自動運転車	325
5.10	9. リモートコントロール	329
5.11	10. ワンタッチスタート	334
5.12	11. 速度のキャリブレーション	336
第 6 章	IoT プロジェクト	341
6.1	1. Blynk でのスタート	343
6.2	2. Blynk からデータを取得する	360
6.3	3. Blynk へのデータ送信	372
6.4	4. クラウドミュージックプレイヤー	380
6.5	5. 家の環境モニタリング	388
6.6	6. 植物モニター	397
6.7	7. 数量制限ゲート	407
6.8	8. IoT カー	415
第 7 章	よくある質問	425
7.1	モバイルデバイスでの Blynk の使用方法	425
7.2	ESP8266 モジュールのファームウェアの再書き込み方法は?	429
第 8 章	ありがとうございます	441

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なレビュー：新製品の発表や先行レビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

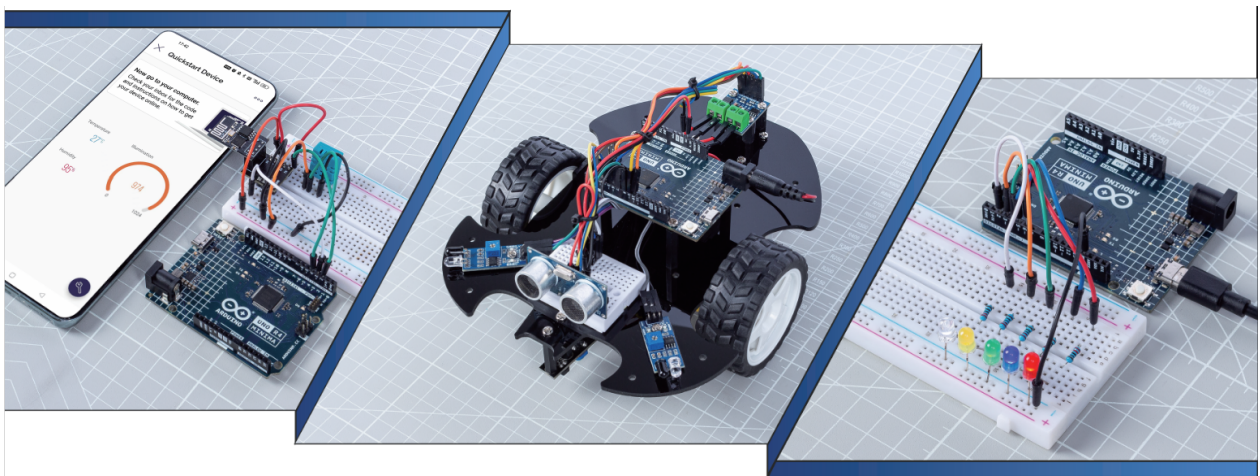
私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

当ててくれてありがとう。私たちの 3-in-1 スターターキットを選んでいただきました。

注釈: このドキュメントは以下の言語で利用可能です。

-
-
-

ご希望の言語でドキュメントにアクセスするために、それぞれのリンクをクリックしてください。



オンラインで学習キットを購入したことはありますか？ 簡単な PDF や冊子しか付属しておらず、プロジェクトを構築する手順だけが記載されていたか？

または、独自のスマートカーを作成したいけど、オンラインで見つかるものは高価で複雑ですか？

他の人が作った有用で興味深い IoT プロジェクトを見たことはありますか？でも、どこから始めるかわからないのですか？

これらの問題は、私たちの 3 in 1 スターターキットで解決できます。

3-in-1 スターターキットには、初心者が Arduino を学ぶのを助ける完全な Arduino コースが含まれており、他の学習キットでは提供されていないスマートカーのプロジェクトや IoT のプロジェクトなど、さまざまな興味深いプロジェクトもあります。キットのコースをステップバイステップでフォローするだけで、コードをコピーしてペーストするのではなく、独自のコードを書いて Arduino プロジェクトを好きなように実装できます。

さあ、ゼロからヒーローになるまで、Arduino プログラミングを自信を持って始めましょう！

質問や他の興味深いアイデアがあれば、お気軽に service@sunfounder.com までメールを送ってください。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

第 1 章

キットのコンポーネントについて学ぶ

パッケージを開けた後、部品の数量が製品の説明と一致しているか、またすべての部品が良好な状態にあるかを確認してください。

Packing List

Resistor(100Ω) 10 pcs	Resistor(100Ω) 10 pcs	Resistor(220Ω) 30 pcs	Resistor(330Ω) 10 pcs	Resistor(1KΩ) 10 pcs	Resistor(2KΩ) 10 pcs	Resistor(5.1KΩ) 10 pcs	Resistor(10KΩ) 10 pcs	Resistor(100KΩ) 10 pcs
Resistor(1MΩ) 10 pcs	Reed Switch 2 pcs	Button 5 pcs	Active/Passive Buzzer 1+1 pcs	Capacitor 104 pF 5 pcs	Potentiometer 1 pc	74HC595 1 pc	7-segment Display 1 pc	IR Receiver 1 pc
Green LED 5 pcs	Red LED 5 pcs	White LED 5 pcs	Blue LED 5 pcs	Yellow LED 5 pcs	RGB LED 1 pc	Photoresistor/ Thermistor 1+1 pcs	M2.5x6 Screw 10 pcs	M3 Nut 8 pcs
M3x10 Screw 4 pcs	M3x6 Screw 30 pcs	M3x30 Screw 6 pcs	M2.5x11 Standoff 6 pcs	M3x24 Standoff 7 pcs	M3x10 Standoff 10 pcs	Screwdriver 1 pc	DHT11 1 pc	ESP8266 Module 1 pc
Soil Moisture Module 1 pc	Joystick Module 1 pc	Ultrasonic Module 1 pc	IR Obstacle Avoidance Module 2 pcs	I2C LCD 1602 1 pc	Arduino UNO R4 Minima 1 pc	Line Tracking Module 1 pc	L9110 Module 1 pc	ESP8266 Adapter Module 1 pc
USB Cable 1 pc	TT Wheel 2 pcs	TT Motor 2 pcs	1" Universal Wheel 1 pc	Remote Control 1 pc	Pump 1 pc	Acrylic Plate 1 pc	9G Servo 1 pc	Mini Breadboard 1 pc
Breadboard 1 pc	Velcro 4 pcs	Tube 1 pc	Jump Wire F/M 20 pcs	F/F DuPont Wire 20 pcs	9V Battery Cable 1 pc	Jump Wire M/M 65 pcs	9V Battery 1 pc	Online Tutorials: https://3in1-kit-r4.rtfid.io

以下は、各部品の紹介です。これには部品の動作原理と対応するプロジェクトが含まれています。

制御ボード

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

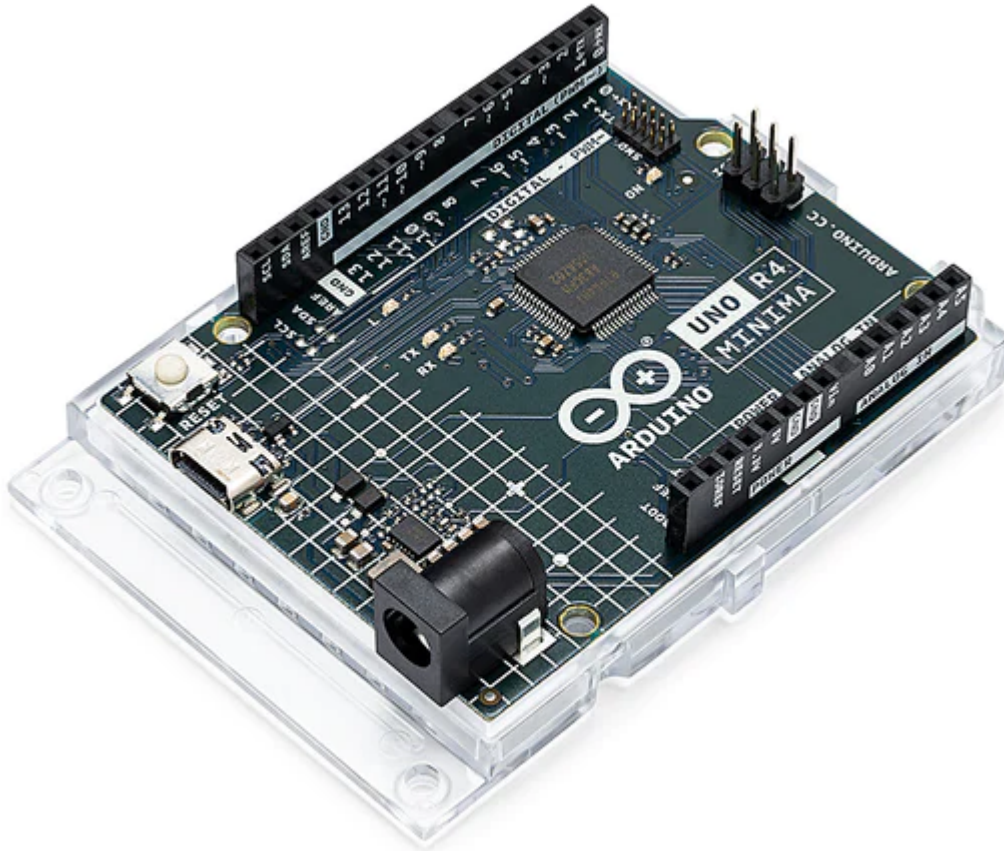
私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.1 Arduino Uno R4 Minima

概要

強化され、向上した Arduino UNO R4 Minima は、Renesas の優れた 32 ビットマイクロコントローラーを搭載しています。増強された処理能力、拡大したメモリ、そして新たなオンボード周辺機器のレベルを体験してください。最も良い点は？既存のシールドやアクセサリとの互換性はそのまま保たれ、標準のフォームファクターや 5 V の動作電圧を変更する必要はありません。

Arduino のエコシステムに参加する UNO R4 は、初心者から熟練した電子工学の愛好家まで適している信頼性の高い追加品です。プロジェクトの境界を押し広げたいのか、それともちょうど始めたばかりのかに関わらず、この頑丈なボードは毎回信頼性の高いパフォーマンスを提供します。



UNO R4 Minima が提供する特長：

- UNO 形状とのハードウェア互換性：UNO R4 Minima は、前身である UNO R3 と同じフォームファクター、ピン配置、および 5 V の動作電圧を保持しています。これにより、Arduino UNO の既存のシールドやプロジェクトのシームレスな移行が保証され、すでに確立された Arduino UNO の広範囲で独特なエコシステムを利用できます。
- 拡張メモリと高速クロック：より正確な計算と複雑なプロジェクトの容易な取り扱いのための準備をしてくださいます。UNO R4 Minima は、メモリを増やし、クロック速度を速めることで、要求の高いタスクにも努力なく対応できます。
- 追加のオンボード周辺機器：UNO R4 Minima には、12 ビット DAC、CAN BUS、OP AMP を含む一連のオンボード周辺機器が導入されています。これらの追加コンポーネントにより、デザインにおいて拡張された機能と柔軟性を手に入れることができます。
- 24 V の拡張許容範囲：UNO R4 Minima は、最大 24 V までの電源供給を許容するようになりました。これにより、モーターや LED ストリップ、その他のアクチュエータとのシームレスな統合が可能になり、単一の電源を利用することでプロジェクトの単純化が実現します。
- SWD コネクタ：デバッグは、あらゆるプロジェクトの重要な側面です。外部のデバッガを UNO R4 Minima に接続するだけで、システムの内部動作を努力なく監視できます。コントロールを保ち、貴重な洞察を得ることができます。

- HID サポート : UNO R4 Minima は、組み込みの HID (Human Interface Device) サポートを備えており、USB ケーブルを介してコンピュータに接続すると、マウスやキーボードをシミュレートすることができます。この便利な機能により、キーストロークやマウスの動きをコンピュータに送信するのが簡単になり、使用性と機能が向上します。

技術仕様

ボード:

- 名称: Arduino® UNO R4 Minima
- SKU: ABX00080
- マイクロコントローラ: Renesas RA4M1 (Arm® Cortex®-M4)
- USB: USB-C® プログラミングポート

ピン:

- デジタル I/O ピン: 14
- アナログ入力ピン: 6
- DAC: 1
- PWM ピン: 6

通信:

- UART: あり, 1x
- I2C: あり, 1x
- SPI: あり, 1x
- CAN: あり 1 CAN Bus

電源:

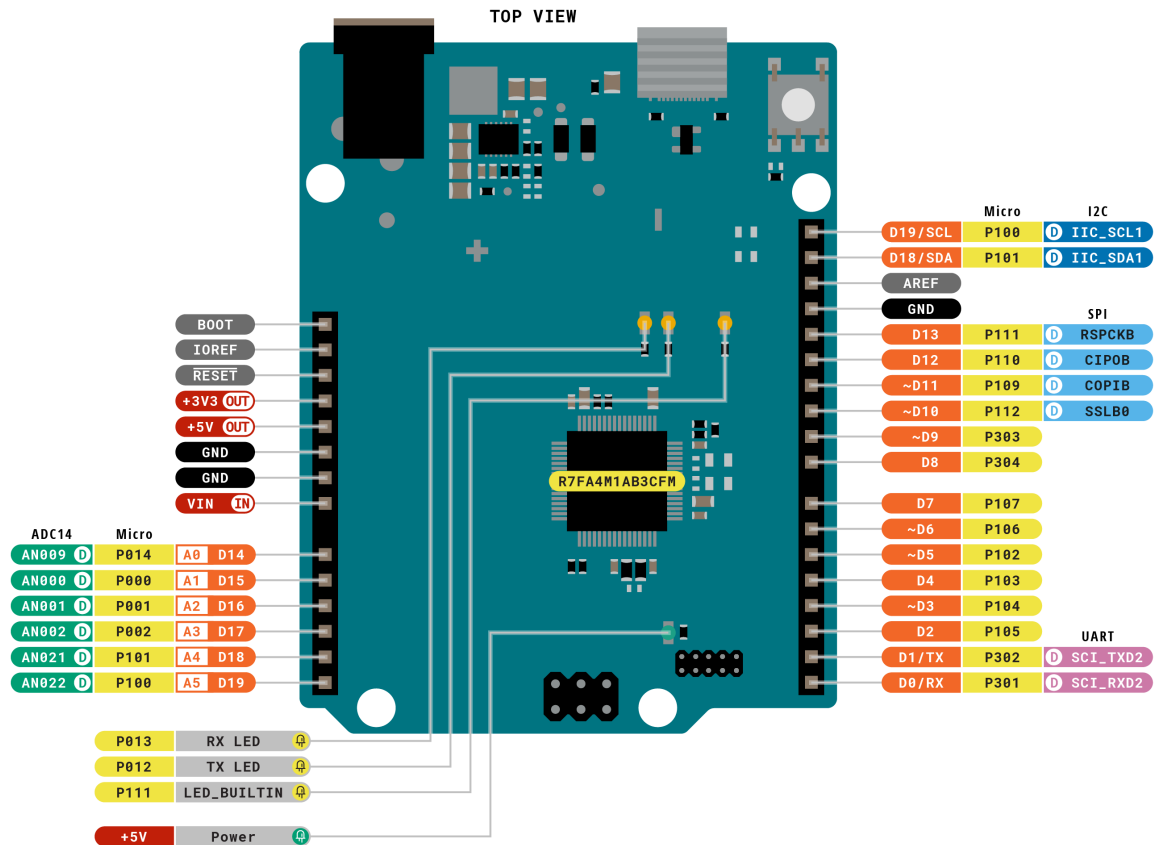
- 回路動作電圧: 5 V
- 入力電圧 (VIN): 6-24 V
- DC 電流/ I/O ピンあたり: 8 mA
- クロック速度 メインコア 48 MHz
- メモリ RA4M1 256 kB Flash, 32 kB RAM

サイズ:

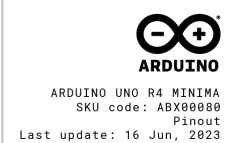
- 幅: 68.85 mm

– 長さ: 53.34 mm

ピン配置



Legend:	Digital	I2C	Other SERIAL
Power	Analog	SPI	Analog
Ground	Main Part	UART/USART	PWM/Timer



-
-
-

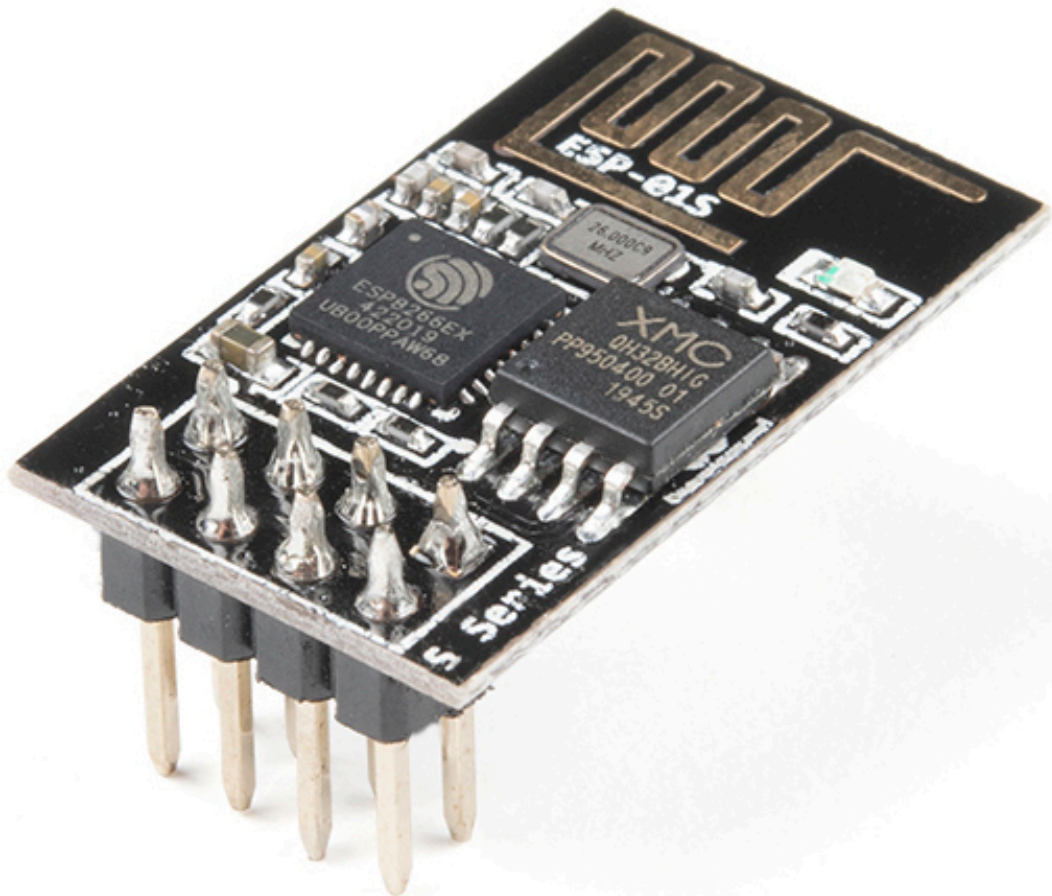
注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.2 ESP8266 モジュール



ESP8266 は、低コストの Wi-Fi マイクロチップであり、組み込みの TCP/IP ネットワーキングソフトウェア、およびマイクロコントローラ機能を備えており、中国の上海にある Espressif Systems で製造されています。

このチップは、2014 年 8 月に Ai-Thinker というサードパーティのメーカーが製造した ESP-01 モジュールで西洋のメーカーの注目を浴びました。この小さなモジュールを使用すると、マイクロコントローラは Wi-Fi ネットワー

クに接続し、Hayes スタイルのコマンドを使用してシンプルな TCP/IP 接続を行うことができます。ただし、最初はこのチップとそれが受け付けるコマンドに関する英語の文書はほとんどありませんでした。非常に低い価格と、モジュール上の外部部品が非常に少なかったことから、大量になれば非常に安価になる可能性があることが、多くのハッカーを引き付けました。その結果、モジュール、チップ、およびそのソフトウェアを調査し、中国語の文書を翻訳する動きが始まりました。

ESP8266 のピンとその機能:

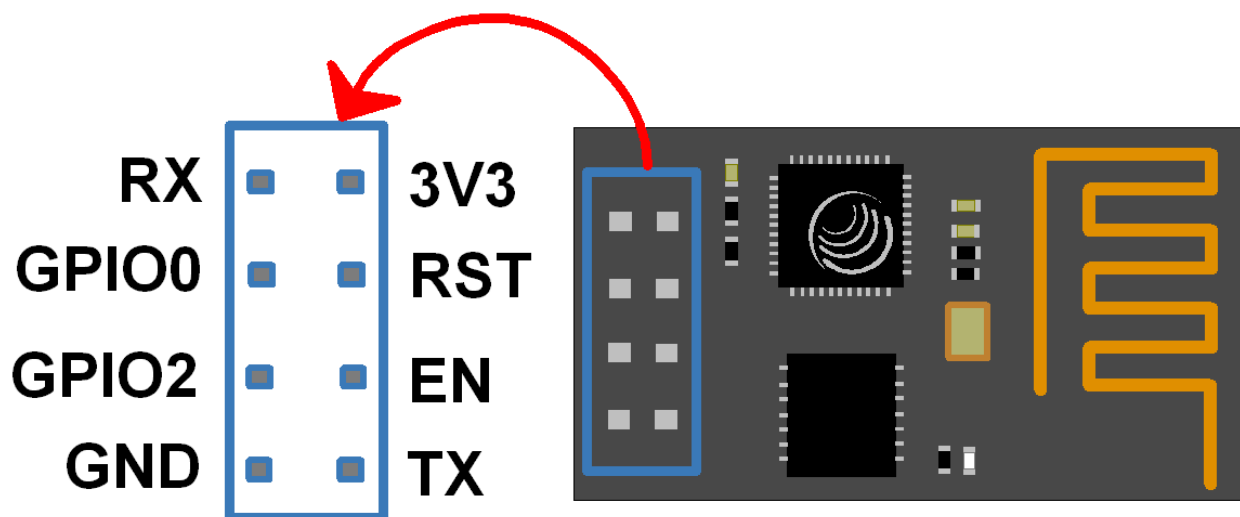
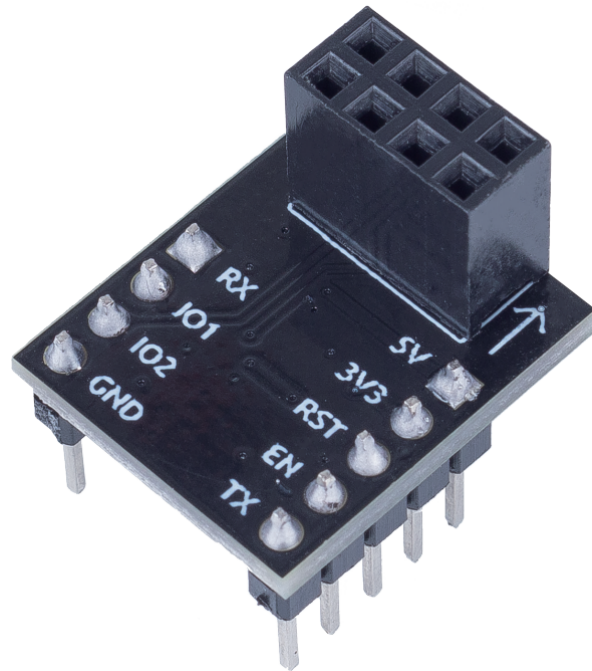


表 1 ESP8266-01 ピン

ピン	名前	説明
1	TXD	UART_TXD、送信; 一般的な入出力: GPIO1; スタートアップ時のプルダウンは許可されていない。
2	GND	GND
3	CU_PD	ハイレベルで動作; 低レベルが供給されると電源がオフ。
4	GPIO2	電源投入時にはハイレベルである必要があり、ハードウェアプルダウンは許可されていない; デフォルトでプルアップ;
5	RST	外部リセット信号、低レベルが供給されるとリセット; ハイレベルが供給されると動作 (デフォルトではハイレベル);
6	GPIO0	WiFi ステータスインジケータ; 操作モード選択: プルアップ: フラッシュブート、操作モード; プルダウン: UART ダウンロード、ダウンロードモード
7	VCC	電源供給 (3.3V)
8	RXD	UART_RXD、受信; 一般的な入出力: GPIO3;

- [ESP8266 - Espressif](#)
- [ESP8266 AT 指令セット](#)

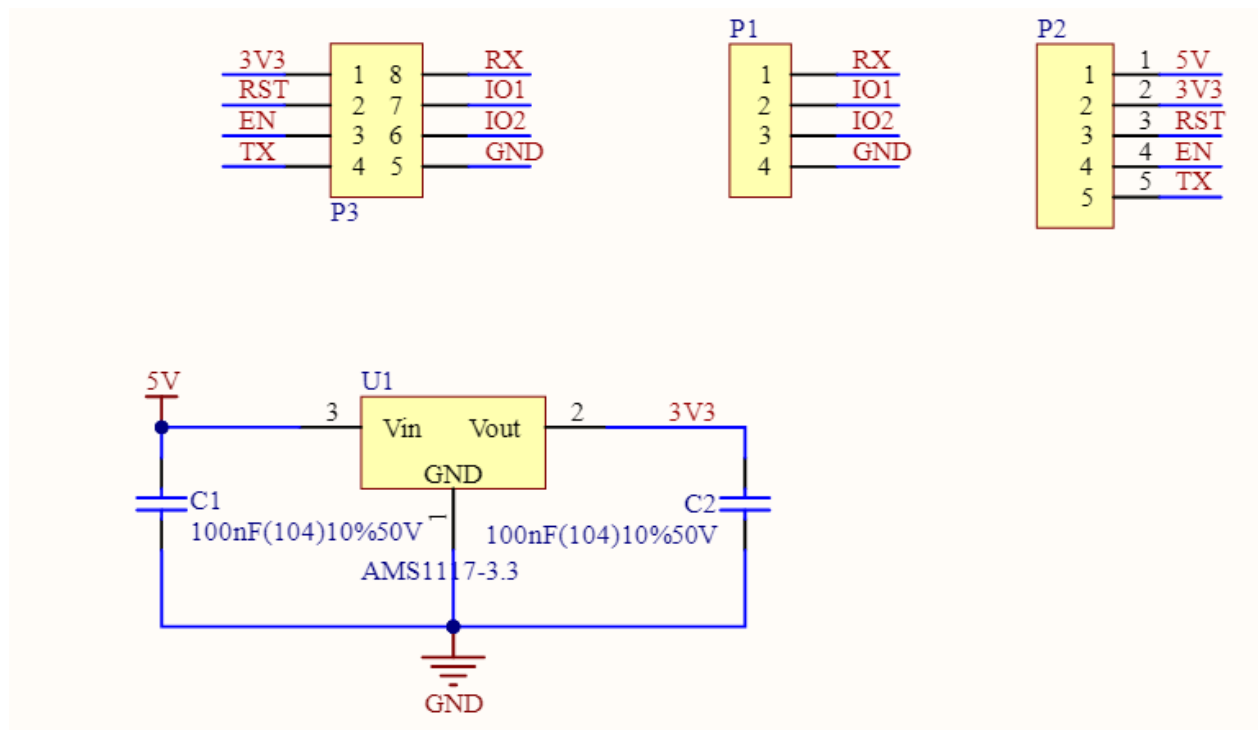
1.2.1 ESP8266 アダプタ



ESP8266 アダプタは、ESP8266 モジュールをブレッドボードでできるようにする拡張ボードです。

それは完全に ESP8266 自体のピンと一致し、Arduino ボードからの電圧を受け取る 5V ピンも追加します。統合された AMS1117 チップは、電圧を 3.3V に落としてから ESP8266 モジュールを駆動するために使用されます。

回路図は以下の通りです:



例

- IoT プロジェクト (IoT プロジェクト)

ベーシック

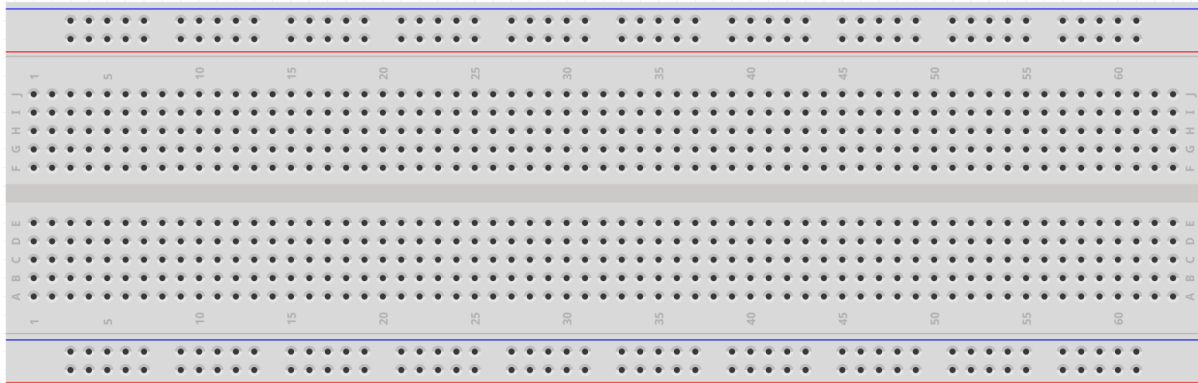
注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

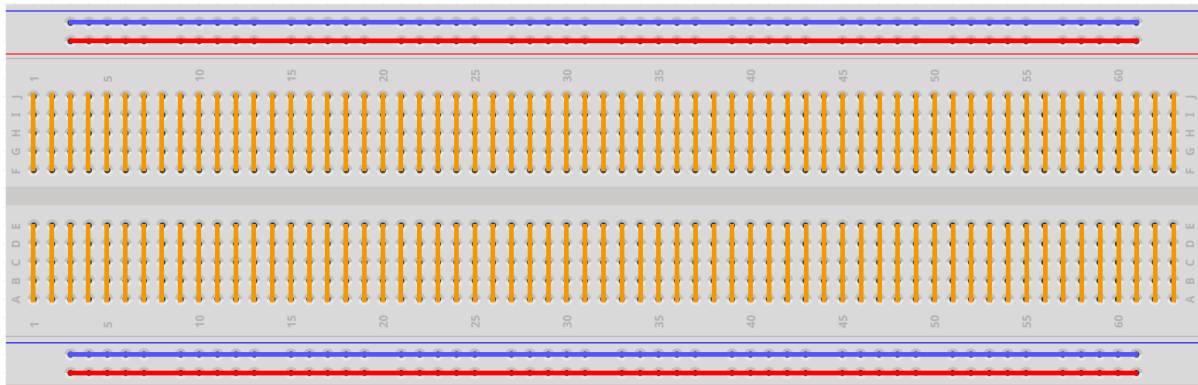
1.3 ブレッドボード



ブレッドボードは、電子回路のプロトタイピング用の構築ベースです。もともこの言葉は、文字通りのパンの板、すなわちパンを切るための磨かれた木片を指していました [1]。1970 年代にはんだ付け不要のブレッドボード（別名プラグボード、ターミナルアレイボード）が登場し、現在では「ブレッドボード」という言葉は主にこれらを指すようになりました。

ブレッドボードは、回路設計を完成させる前に迅速に回路を構築し、テストするために使用されます。そして、上記のようなコンポーネントを IC や抵抗器、ジャンパーワイヤーのように挿入できる多くの穴があります。ブレッドボードを使用すると、コンポーネントを簡単に差し込んだり、取り外したりすることができます。

この画像はブレッドボードの内部構造を示しています。ブレッドボード上のこれらの穴はお互いに独立しているように見えますが、実際には内部で金属のストリップを介して互いに接続されています。



ブレッドボードの詳細について知りたい場合は、こちらを参照してください：[ブレッドボードの使い方 - Science Buddies](#)

注釈： こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.4 抵抗器



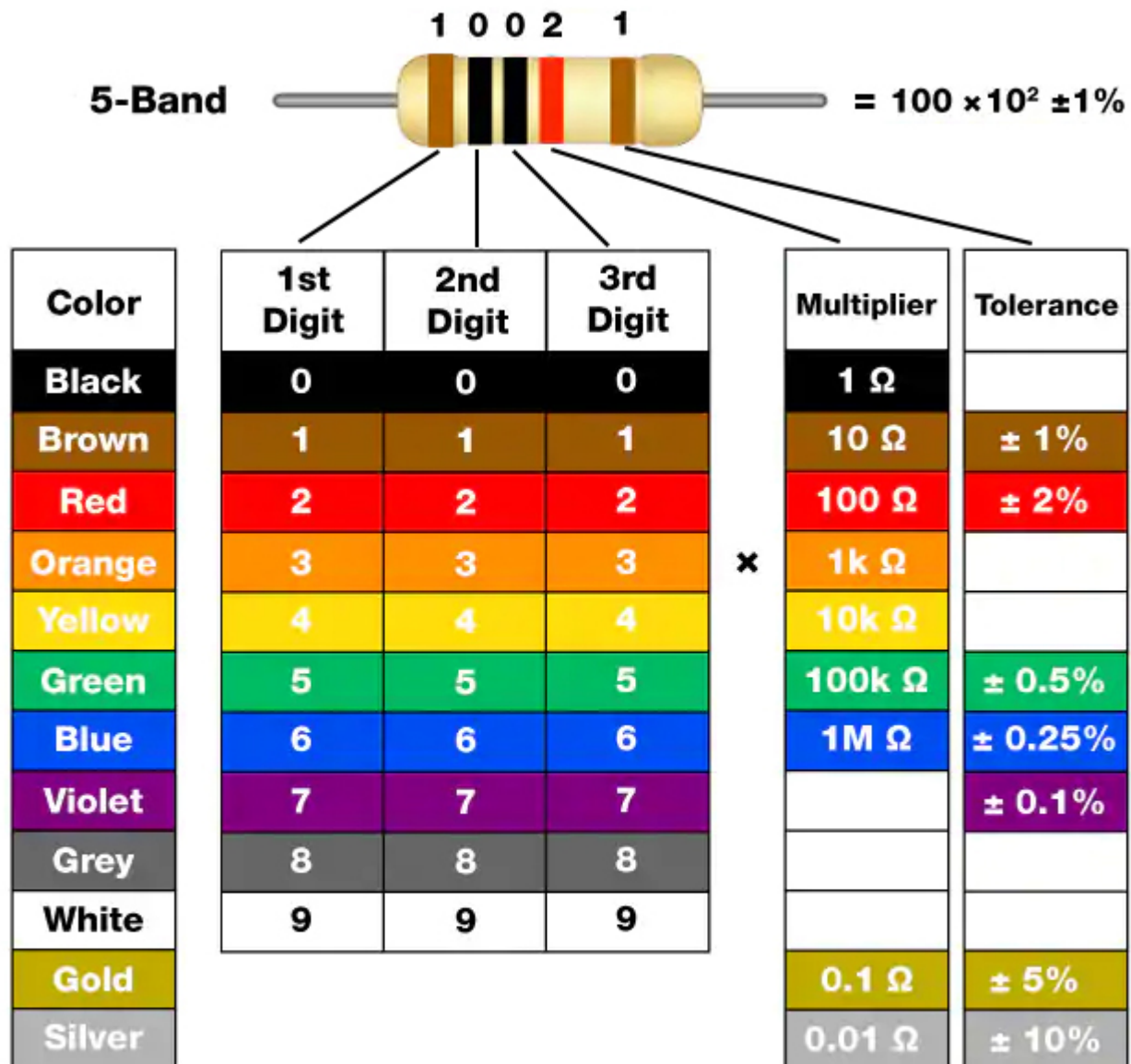
抵抗器は、ブランチ電流を制限する電子部品です。固定抵抗器は、抵抗値を変更できない抵抗器で、一方、ポテンショメーターや可変抵抗器は調整可能です。

抵抗器の一般的な回路シンボルが2つあります。通常、これらのシンボルには抵抗値が記載されています。従って、これらのシンボルが回路内にある場合、それは抵抗器を示しています。



は抵抗の単位で、大きな単位としては K 、M などがあります。これらの関係は次のように示されます：1 M = 1000 K 、1 K = 1000 。通常、抵抗器にはその値が記載されています。

抵抗器を使用する場合、まずその抵抗値を知る必要があります。2つの方法があります：抵抗器の色帯を観察するか、マルチメーターを使用して抵抗を測定します。前者の方法を使用することをおすすめします、より便利で迅速だからです。



このカードに示されているように、各色は数字を示しています。

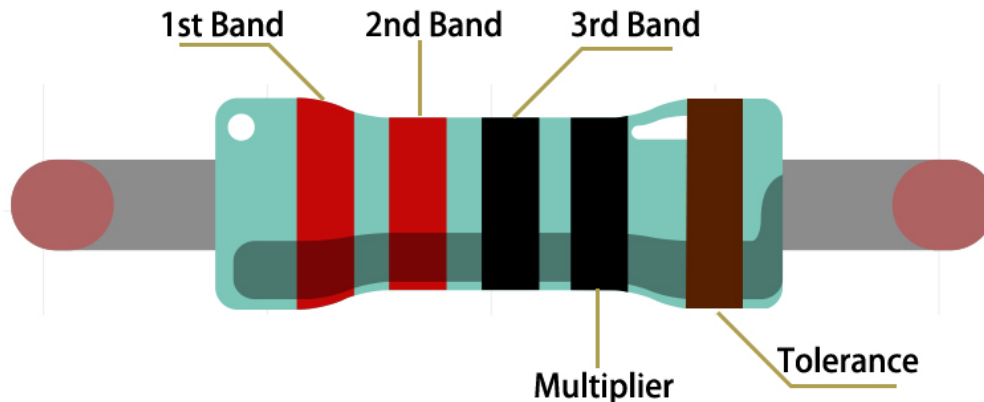
黒	茶	赤	橙	黄	緑	青	紫	灰	白	金	銀
0	1	2	3	4	5	6	7	8	9	0.1	0.01

4 帯と 5 帯の抵抗器はよく使用されており、それぞれには 4 つと 5 つの色帯があります。

抵抗器を手に入れると、どの端から色を読み取るかを決定するのが難しい場合があります。コツは、4 番目と 5 番目の帯の間の間隔が比較的大きいことです。

したがって、抵抗器の一端の 2 つの色帯の間隔を観察します。他の帯の間隔よりも大きい場合、反対側から読み取ることができます。

以下に示す 5 帯の抵抗器の抵抗値の読み取り方を見てみましょう。



この抵抗器の場合、抵抗は左から右に読み取る必要があります。値は次のフォーマットである必要があります：1 番帯 2 番帯 3 番帯 x $10^{\text{乗数}}$ () そして許容誤差は \pm 許容 %。この抵抗器の抵抗値は 2(赤) 2(赤) 0(黒) x 10^0 (黒) = 220 で、許容誤差は $\pm 1\%$ (茶) です。

Wikipedia で抵抗器についてさらに詳しく知ることができます：[抵抗器 - Wikipedia](#)。

注釈：こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.5 コンデンサー





コンデンサーは、与えられた電位差下での電荷蓄積量を指し、C として表され、国際単位はファラッド (F) です。一般に、電場中で電荷は力の下で移動します。導体間に媒体が存在すると、電荷の移動が妨げられ、電荷が導体上に蓄積され、電荷が蓄積される結果となります。

この蓄積される電荷の量を容量と言います。コンデンサーは電子機器において最も広く使用される電子部品の一つであり、直流絶縁、結合、バイパス、フィルタリング、チューニンググループ、エネルギー変換、制御回路などの用途で広く利用されています。コンデンサーは、電解コンデンサー、固体コンデンサーなどに分けられます。

材料の特性により、コンデンサーは以下のように分類されます：アルミニウム電解コンデンサー、フィルムコンデンサー、タンタルコンデンサー、セラミックコンデンサー、スーパーコンデンサーなど。

このキットでは、セラミックコンデンサーと電解コンデンサーが使用されています。

- [セラミックコンデンサー - Wikipedia](#)
- [電解コンデンサー - Wikipedia](#)

セラミックコンデンサーには 103 または 104 のラベルがあり、これは容量値を表し、 $103=10 \times 10^3 \text{pF}$ 、 $104=10 \times 10^4 \text{pF}$ となります。

単位換算

$$1\text{F}=10^3\text{mF}=10^6\mu\text{F}=10^9\text{nF}=10^{12}\text{pF}$$

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.6 ジャンパーワイヤー

2つの端子を接続するワイヤーをジャンパーワイヤーと言います。ジャンパーワイヤーにはさまざまな種類がありますが、ここではブレッドボードで使用されるものに焦点を当てます。特に、ブレッドボード上の任意の位置からマイクロコントローラの入出力ピンに電気信号を伝送するために使用されます。

ジャンプワイヤーは、その「エンドコネクタ」をブレッドボードに提供されるスロットに挿入することで取り付けられます。ブレッドボードの表面の下には、行または列のグループでスロットを接続するいくつかの並列プレートがあり、エリアに応じて変わります。「エンドコネクタ」は、特定のプロトタイプで接続する必要がある特定のスロットに、はんだ付けせずにブレッドボードに挿入されます。

ジャンパーワイヤーには3つのタイプがあります：メス-メス、オス-オス、オス-メス。オス-メスと呼ぶのは、片方の端が突出した先端を持ち、もう片方の端が凹んだメス端子を持っているためです。オス-オスは両方の端がオスで、メス-メスは両端がメスであることを意味します。



プロジェクトには、これらのうちの複数のタイプが使用される場合があります。ジャンプワイヤーの色は異なりますが、それがそれぞれの機能が異なることを意味するわけではありません。それは、各回路間の接続をより明確に識別するためのデザインです。

チップ

注釈： こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

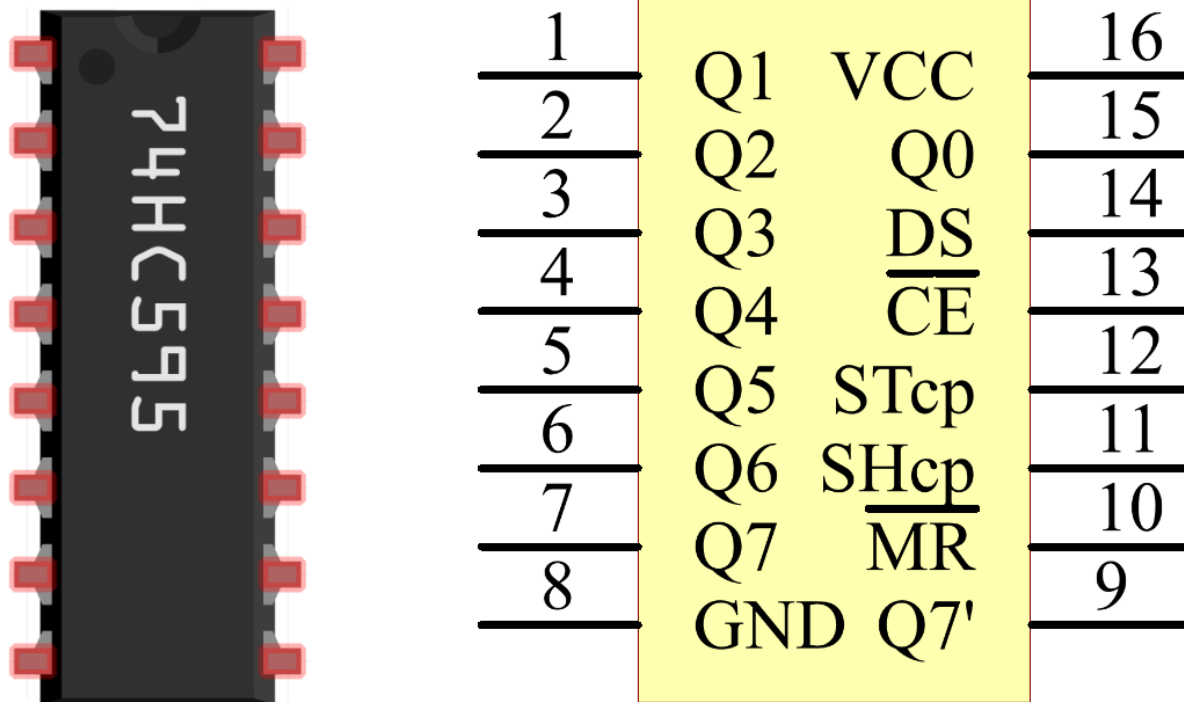
私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.7 74HC595



74HC595 は、8 ビットのシフトレジスタと、三状態の並列出力を持つストレージレジスタで構成されています。シリアル入力を並列出力に変換することで、MCU の IO ポートの使用を節約できます。MR (pin10) が高レベルで、OE (pin13) が低レベルの時、SHcp の立ち上がりエッジでデータが入力され、SHcp の立ち上がりエッジを通じてメモリレジスタに移動します。2 つのクロックが接続されている場合、シフトレジスタは常にメモリレジスタよりも 1 パルス早いです。メモリレジスタには、シリアルシフト入力ピン (Ds)、シリアル出力ピン (Q)、非同期リセットボタン (低レベル) があります。メモリレジスタは、3 つの状態ですべて 8 ビットのバスを出力します。OE が有効化 (低レベル) の場合、メモリレジスタのデータがバスに出力されます。

- [74HC595 データシート](#)



74HC595 のピンとその機能:

- **Q0-Q7**: 8 ビットの並列データ出力ピン。直接 8 個の LED または 7 セグメント表示の 8 ピンを制御することができます。
- **Q7'**: シリーズ出力ピン。他の 74HC595 の DS に接続して、複数の 74HC595 を直列に接続します。
- **MR**: 低レベルでアクティブなリセットピン。
- **SHcp**: シフトレジスタのタイムシーケンス入力。立ち上がりエッジで、シフトレジスタ内のデータが逐次 1 ビット移動します。例えば、Q1 のデータは Q2 に移動します。立ち下がりエッジでは、シフトレジスタ内のデータは変わらずに残ります。
- **STcp**: ストレージレジスタのタイムシーケンス入力。立ち上がりエッジで、シフトレジスタのデータがメモリレジスタに移動します。
- **CE**: 低レベルでアクティブな出力有効ピン。
- **DS**: シリアルデータ入力ピン。
- **VCC**: 正の供給電圧。
- **GND**: アース。

例

- [5.9 ShiftOut\(LED\)](#) (基本プロジェクト)
- [5.10 シフトアウト \(セグメントディスプレイ\)](#) (基本プロジェクト)

- 7. 数量制限ゲート (IoT プロジェクト)

ディスプレイ

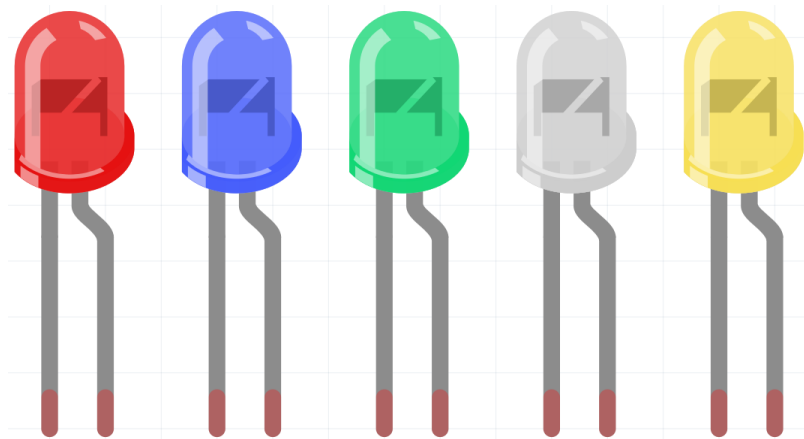
注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

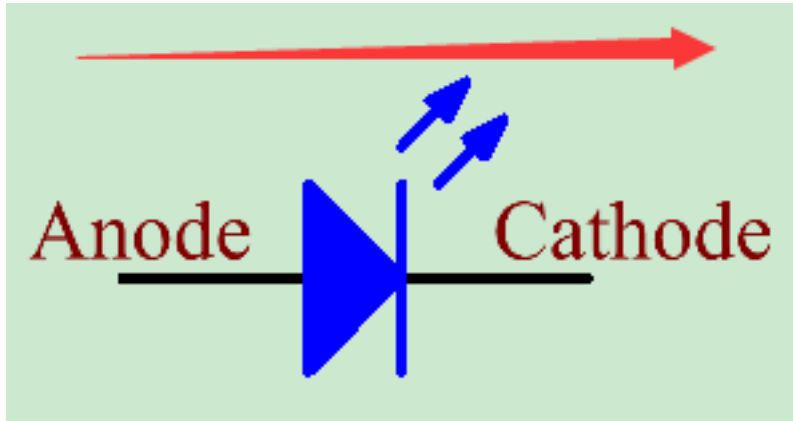
私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.8 LED



半導体発光ダイオードは、PN 接合を通じて電気エネルギーを光エネルギーに変換する部品の一つです。波長によって、レーザーダイオード、赤外線発光ダイオード、そして通常 LED として知られる可視光発光ダイオードに分類できます。

ダイオードは一方方向の導電性を持っているため、図の回路記号に示される矢印の方向に電流が流れます。陽極には正の電力を、カソードには負を供給する必要があります。そうすると LED は点灯します。



LED には 2 つのピンがあります。長い方が陽極、短い方がカソードです。逆に接続しないように注意してください。LED には一定の順方向電圧降下が存在するため、直接回路に接続することはできません。供給電圧がこの降下を超えて LED が焼ける原因となるからです。赤、黄、緑の LED の順方向電圧は 1.8 V で、白色のものは 2.6 V です。ほとんどの LED は最大 20 mA の電流に耐えることができるので、直列に電流制限抵抗を接続する必要があります。

抵抗値の計算式は以下の通りです:

$$R = (V_{\text{supply}} - V_D) / I$$

ここで、**R** は電流制限抵抗の抵抗値、**Vsupply** は供給電圧、**VD** は電圧降下、**I** は LED の動作電流を示します。

LED の詳細な紹介はこちら: [LED - Wikipedia](#)。

例

- [1.1 こんにちは、LED！](#) (基本プロジェクト)
- [2.1 フェーディング](#) (基本プロジェクト)
- [2. Blynk からデータを取得する](#) (IoT プロジェクト)

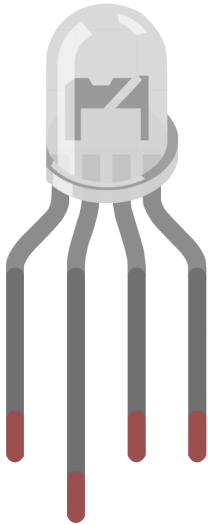
注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

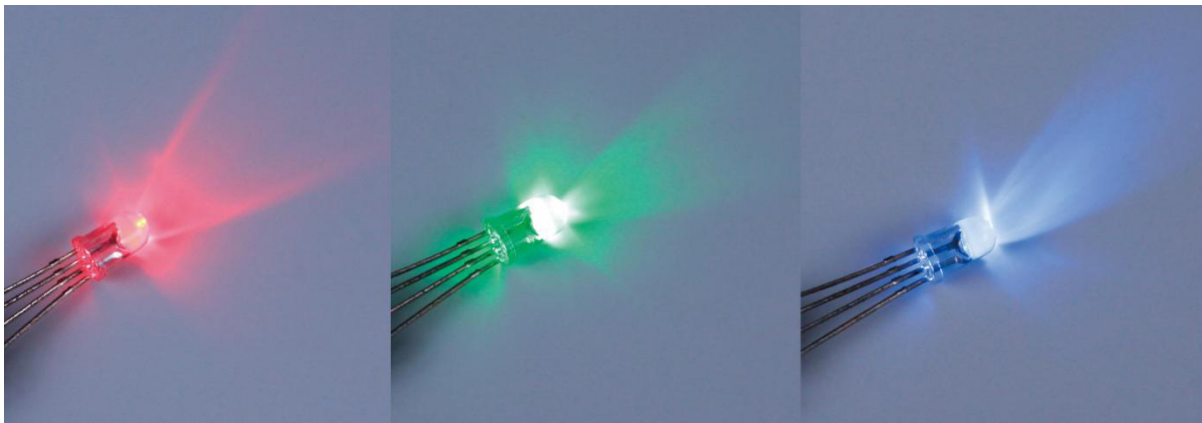
- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.9 RGB LED

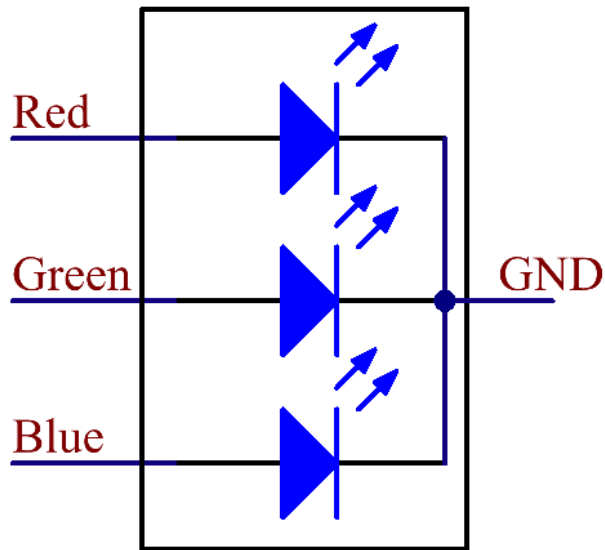


RGB LED はさまざまな色の光を放射します。RGB LED は赤、緑、青の 3 つの LED を透明または半透明のプラスチックケースにパッケージングしています。3 つのピンの入力電圧を変更してそれらを重ね合わせることで、統計によれば 16,777,216 色の異なる色を表示することができます。

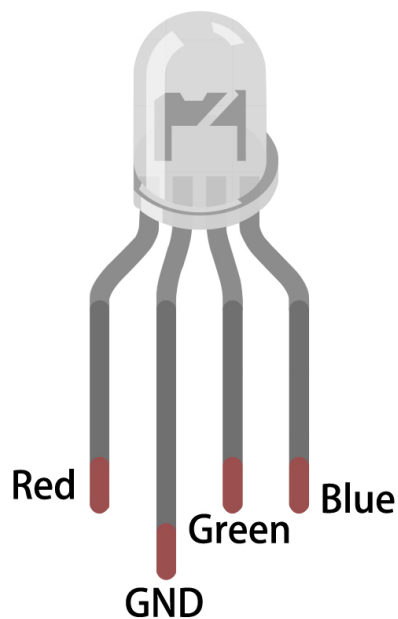


RGB LED は共通のアノードと共通のカソードに分類できます。このキットでは、後者が使用されています。共通カソード、または CC、は 3 つの LED のカソードを接続することを意味します。それを GND に接続し、3 つのピンにプラグを差し込むと、LED は対応する色で点滅します。

その回路記号は以下の図として示されています。



RGB LED には 4 つのピンがあります：最も長いものは GND で、他の 3 つは Red、Green、Blue です。そのプラスチックケースに触れると切れ込みが感じられます。この切れ込みに最も近いピンが最初のピンであり、赤としてマークされ、次に GND、緑、青の順番です。



例

- [2.2 カラフルな光](#) (基本プロジェクト)
- [5.2 閾値](#) (基本プロジェクト)

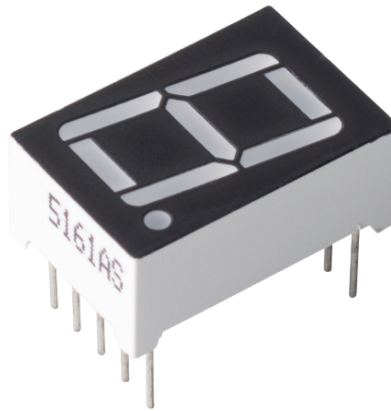
注釈： こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

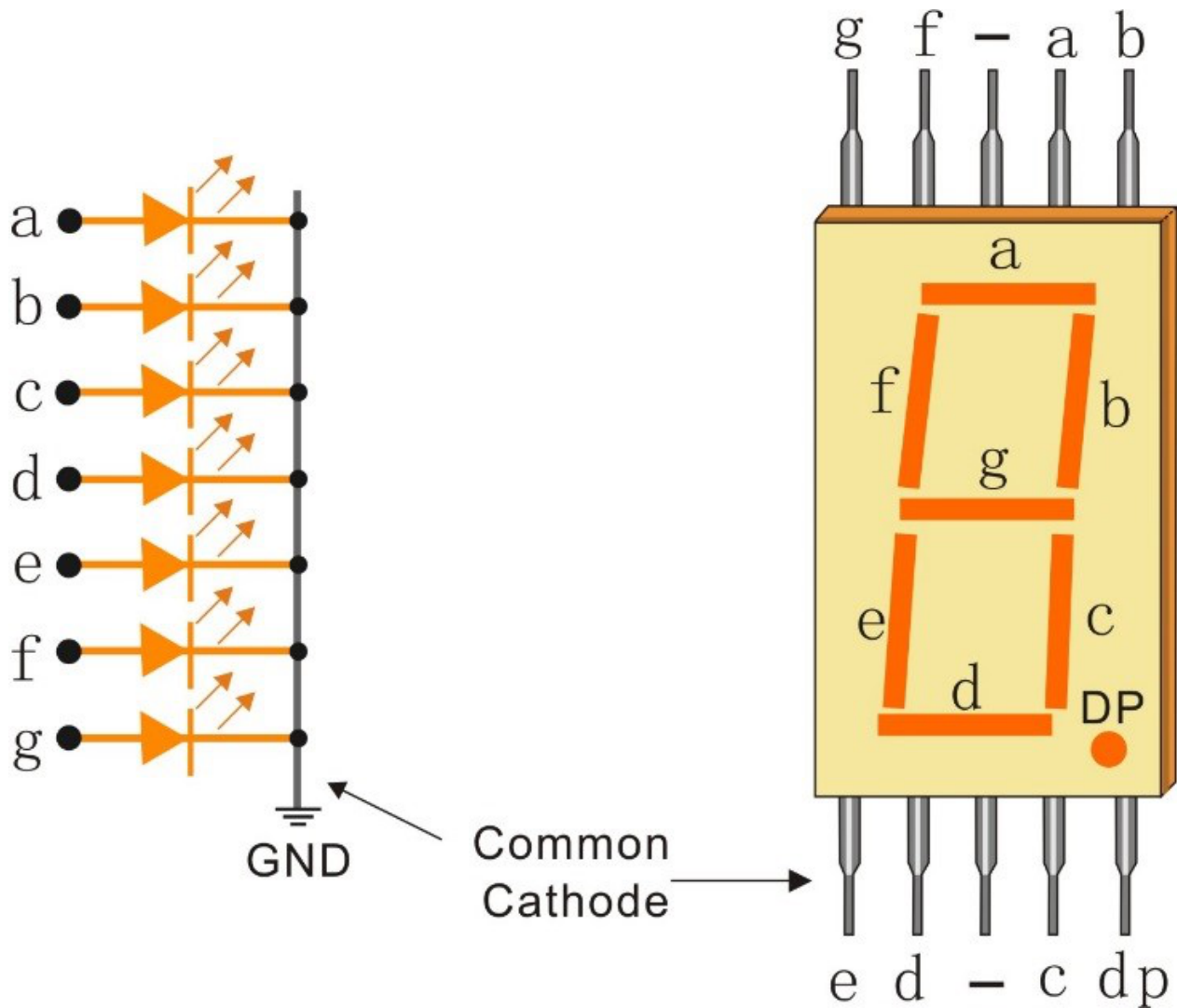
1.10 7 セグメントディスプレイ



7 セグメントディスプレイは、7 つの LED をパッケージングした 8 の形の部品です。各 LED はセグメントと呼ばれ、電力を供給すると、表示する数字の一部としてのセグメントが形成されます。

ピンの接続には 2 種類あります：共通カソード（CC）と共通アノード（CA）。名前が示す通り、CC ディスプレイは 7 つの LED のカソードがすべて接続されており、CA ディスプレイは 7 つのセグメントのアノードがすべて接続されています。

このキットでは、共通カソードの 7 セグメントディスプレイを使用しています。以下はその電子記号です。



ディスプレイの各 LED には、四角形のプラスチックパッケージから導出される接続ピンのうちの 1 つが与えられる位置的なセグメントが与えられます。これらの LED ピンは「a」から「g」までの各個別の LED を表すようにラベルが付けられています。他の LED ピンは、共通のピンを形成するために接続されます。したがって、LED セグメントの適切なピンに順方向にバイアスをかけると、一部のセグメントが明るくなり、他のセグメントが暗くなるため、ディスプレイ上に対応する文字が表示されます。

ディスプレイコード

7 セグメントディスプレイ（共通カソード）が数字をどのように表示するかを理解するのを助けるために、以下の表を描きました。Numbers は、7 セグメントディスプレイに表示される数字 0-F; (DP) GFEDCBA は、対応する LED が 0 または 1 に設定されていることを示しています。例えば、00111111 は DP と G が 0 に設定されているのに対し、他のすべてが 1 に設定されていることを意味します。したがって、数字 0 が 7 セグメントディスプレイに表示され、HEX Code は 16 進数に対応します。

Numbers	Common Cathode		Numbers	Common Cathode	
	(DP)GFEDCBA	Hex Code		(DP)GFEDCBA A	Hex Code
0	00111111	0x3f	A	01110111	0x77
1	00000110	0x06	B	01111100	0x7c
2	01011011	0x5b	C	00111001	0x39
3	01001111	0x4f	D	01011110	0x5e
4	01100110	0x66	E	01111001	0x79
5	01101101	0x6d	F	01110001	0x71
6	01111101	0x7d			
7	00000111	0x07			
8	01111111	0x7f			
9	01101111	0x6f			

例

- [5.15 EEPROM](#) (基本プロジェクト)
- [7. 数量制限ゲート](#) (IoT プロジェクト)

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

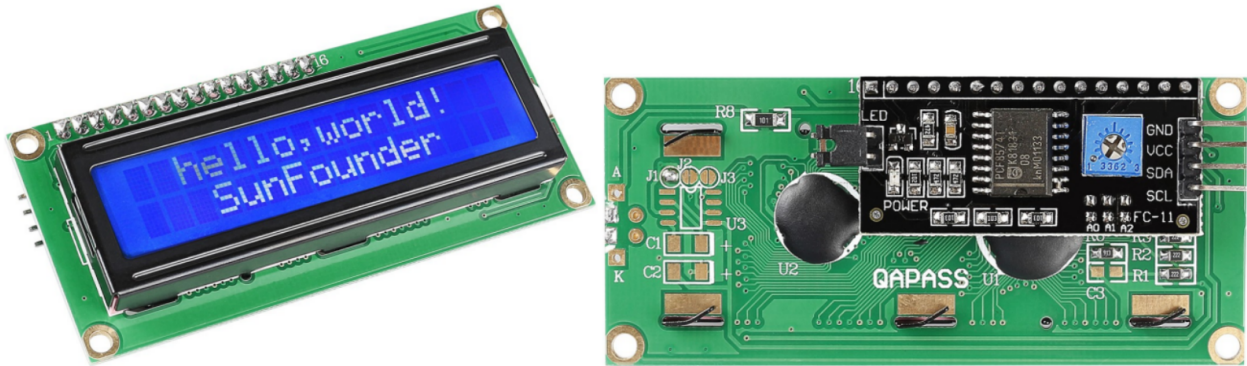
参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。

- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.11 I2C LCD1602



- **GND:** アース
- **VCC:** 電源供給、5V。
- **SDA:** シリアルデータライン。プルアップ抵抗を介して VCC に接続します。
- **SCL:** シリアルクロックライン。プルアップ抵抗を介して VCC に接続します。

皆さんがご存知のように、LCD やその他のディスプレイは人とマシンの相互作用を大いに豊かにしていますが、共通の弱点があります。コントローラに接続されると、多くの IO がコントローラの外部ポートを占有し、その他のコントローラの機能も制限されます。

そのため、この問題を解決するために I2C モジュールを備えた LCD1602 が開発されました。I2C モジュールには、I2C シリアルデータを LCD ディスプレイの並列データに変換する内蔵の PCF8574 I2C チップがあります。

- [PCF8574 データシート](#)

I2C アドレス

デフォルトのアドレスは基本的に 0x27 で、いくつかのケースでは 0x3F になることがあります。

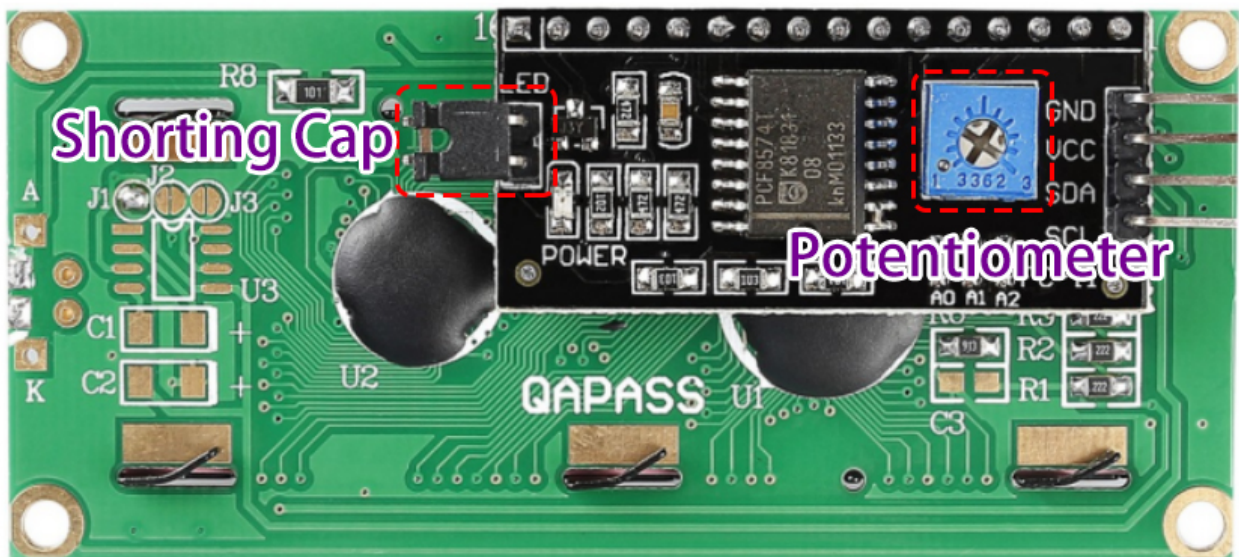
デフォルトのアドレス 0x27 を例にとると、A0/A1/A2 パッドをショートすることでデバイスアドレスを変更することができます。デフォルトの状態では、A0/A1/A2 は 1 であり、パッドがショートすると A0/A1/A2 は 0 になります。

Slave Address

0	0	1	0	0	A2	A1	A0	
0	0	1	0	0	1	1	1	0x27
0	0	1	0	0	1	1	0	0x26
0	0	1	0	0	1	0	1	0x25
0	0	1	0	0	0	1	1	0x23
.....								
0	0	1	0	0	0	0	0	0x20

バックライト/コントラスト

ジャンパーキャップでバックライトを有効にでき、ジャンパーキャップを外すとバックライトを無効にできます。背面の青いポテンショメータは、最も明るい白と最も暗い黒の明るさの比率を調整するために使用されます。



- ショートキャップ (Shorting Cap) : このキャップでバックライトを有効にでき、キャップを外すとバック

ライトを無効にできます。

- ポテンショメータ (**Potentiometer**) : 表示テキストの明瞭度を調整するために使用されます。時計回りに回すと増加し、反時計回りに回すと減少します。

例

- [5.11.1 液晶ディスプレイ](#) (基本プロジェクト)
- [5.12 シリアル読み取り](#) (基本プロジェクト)

サウンド

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.12 ブザー



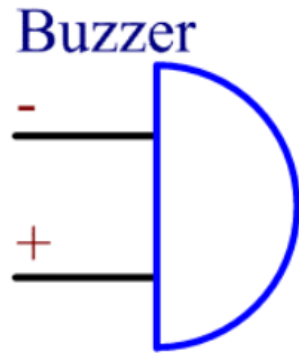
統合構造を持つ電子ブザーの一種として、DC 電源で駆動されるブザーは、コンピュータ、プリンタ、複写機、アラーム、電子玩具、自動車電子デバイス、電話、タイマーなどの電子製品や音声デバイスで広く使用されています。

ブザーはアクティブとパッシブの二つに分類されます（以下の画像参照）。ブザーのピンが上を向いているように回転させると、緑の基板が付いているブザーはパッシブブザーで、黒テープで囲まれているものはアクティブブザーです。

アクティブブザーとパッシブブザーの違い：

アクティブブザーは内蔵の振動源を持っているため、通電すると音が出ます。しかし、パッシブブザーはそのような源を持っていないため、DC 信号を使用してもピープ音が出ません。代わりに、2K から 5K の間の周波数の正方形波を使用して駆動する必要があります。複数の振動回路が内蔵されているため、アクティブブザーはパッシブブザーよりも高価です。

以下はブザーの電気的なシンボルです。正極と負極の 2 つのピンを持っています。表面に + が表示されているものが陽極、もう一方が陰極です。



ブザーのピンを確認することができます。長い方が陽極で、短い方が陰極です。接続する際に間違えないようにしてください。そうでないと、ブザーは音を出しません。

[ブザー - ウィキペディア](#)

例

- [1.2 ビープ音](#) (基本プロジェクト)
- [5.7 Tone\(\)](#) または [noTone\(\)](#) (基本プロジェクト)
- [4. クラウドミュージックプレイヤー](#) (IoT プロジェクト)

ドライバー

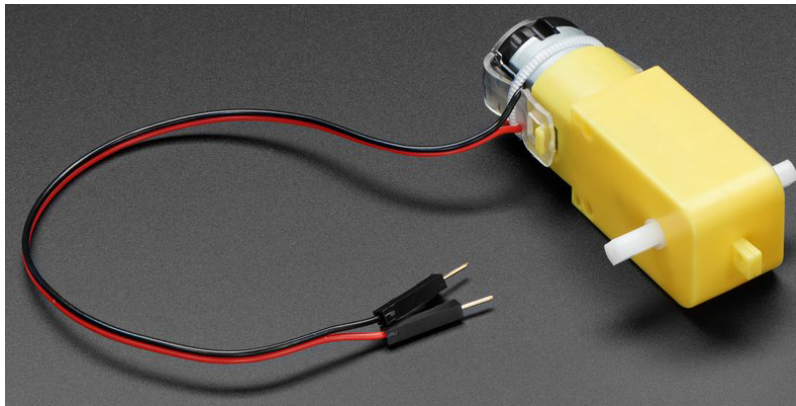
注釈： こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.13 TT モーター



これはギア比 1:48 の TT DC ギアボックスモーターで、ブレッドボードに適合する 0.1"の男性コネクタ付きの 2 x 200mm のワイヤーが付属しています。ブレッドボードや端子ブロックに接続するのに最適です。

これらのモーターは 3~6VDC で駆動することができますが、もちろん、電圧が高いほど少し速く動きます。

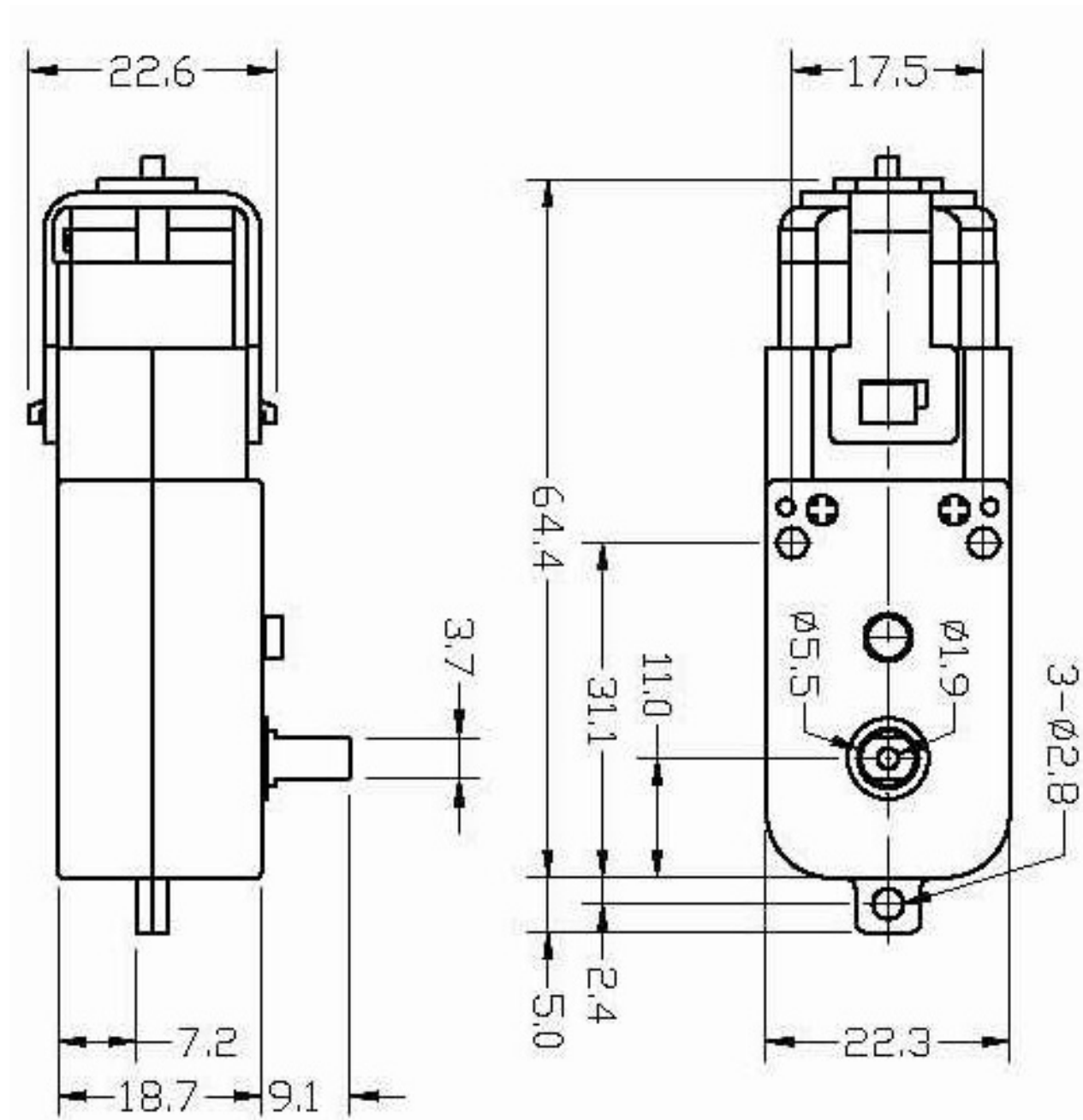
これらは非常に基本的なモーターであり、内蔵エンコーダ、速度制御、位置フィードバックはありません。電圧が入力され、回転が出力されます。モーターごとに変動があるため、正確な動きが必要な場合は、別のフィードバックシステムが必要です。

技術詳細

- 定格電圧: 3~6V
- 無負荷連続電流: 150mA +/- 10%
- 最低運転速度 (3V): 90+/- 10% RPM
- 最低運転速度 (6V): 200+/- 10% RPM
- ストールトルク (3V): 0.4kg.cm

- ストールトルク (6V): 0.8kg.cm
- ギア比: 1:48
- 本体寸法: 70 x 22 x 18mm
- ワイヤー長: 200mm & 28 AWG
- 重量: 30.6g

寸法図



例

- [1.3 車輪を回す](#) (基本プロジェクト)
- [1. 移動](#) (車のプロジェクト)
- [3. 速度の上昇](#) (車のプロジェクト)
- [8. IoT カー](#) (IoT プロジェクト)

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

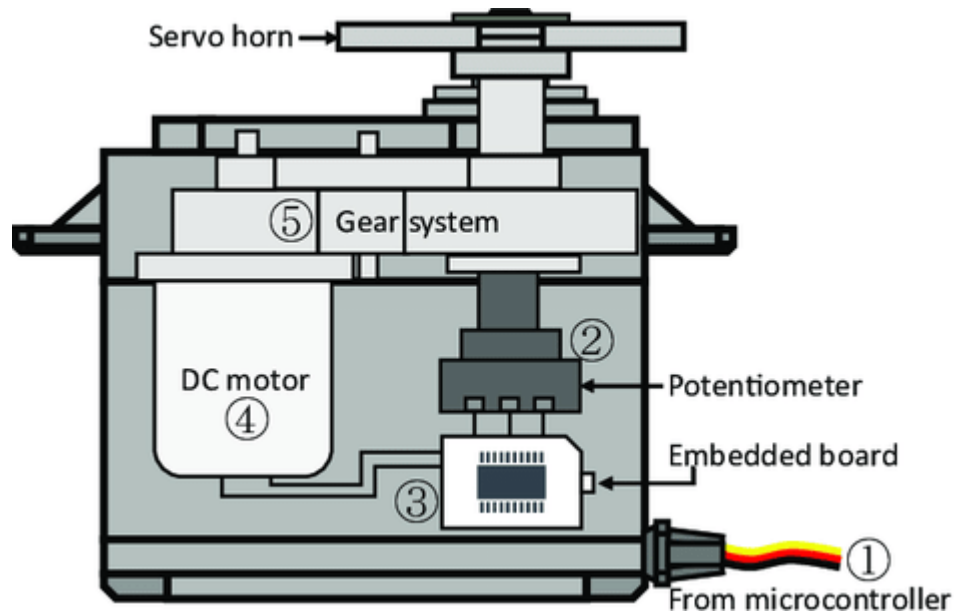
私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.14 サーボ

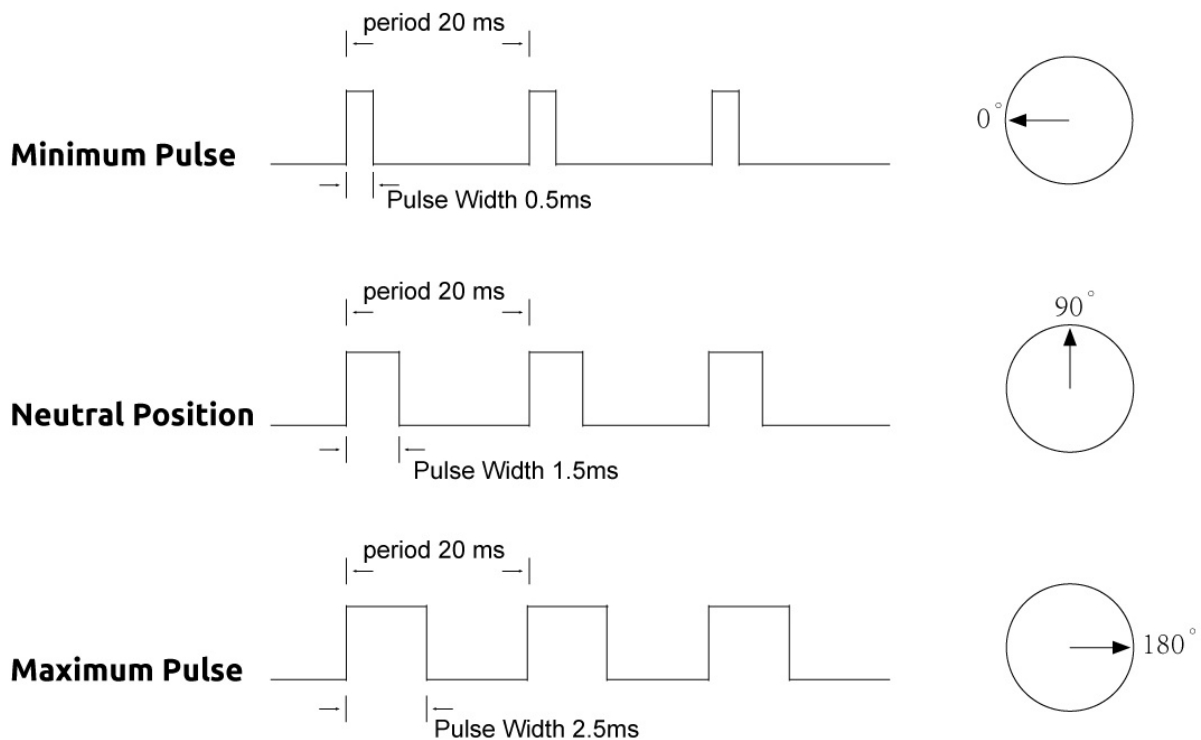


サーボは一般的に、ケース、シャフト、ギアシステム、ポテンショメータ、DC モーター、および組み込みボードで構成されています。

動作原理は以下の通りです。マイクロコントローラが PWM 信号をサーボに送信すると、サーボ内の組み込みボードが信号ピンを介して信号を受信し、内部のモーターを制御して回転させます。その結果、モーターはギアシステムを駆動し、減速後にシャフトを動かします。サーボのシャフトとポテンショメータは連結されています。シャフトが回転すると、ポテンショメータも駆動し、ポテンショメータから電圧信号が組み込みボードに出力されます。次に、ボードは現在の位置に基づいて回転の方向と速度を決定し、定義された正確な位置で正確に停止し、その位置を保持します。



角度は、制御ワイヤに適用されるパルスの持続時間によって決定されます。これをパルス幅変調と言います。サーボは 20ms ごとにパルスを受け取ることを期待しています。パルスの長さによって、モーターがどれだけ回転するかが決定されます。例えば、1.5ms のパルスは、モーターを 90 度の位置（中立位置）に回転させます。1.5ms 未満のパルスがサーボに送られると、サーボはある位置に回転し、その出力シャフトを中立点から反時計回りに何度か保持します。パルスが 1.5ms よりも広い場合、逆の現象が発生します。サーボが有効な位置に回転する命令の最小幅と最大幅は、各サーボの関数です。一般的に、最小パルスは約 0.5ms、最大パルスは 2.5ms の幅になります。



例

- 5.5 内蔵ライブラリの使用 (基本プロジェクト)
- 7. 数量制限ゲート (IoT プロジェクト)

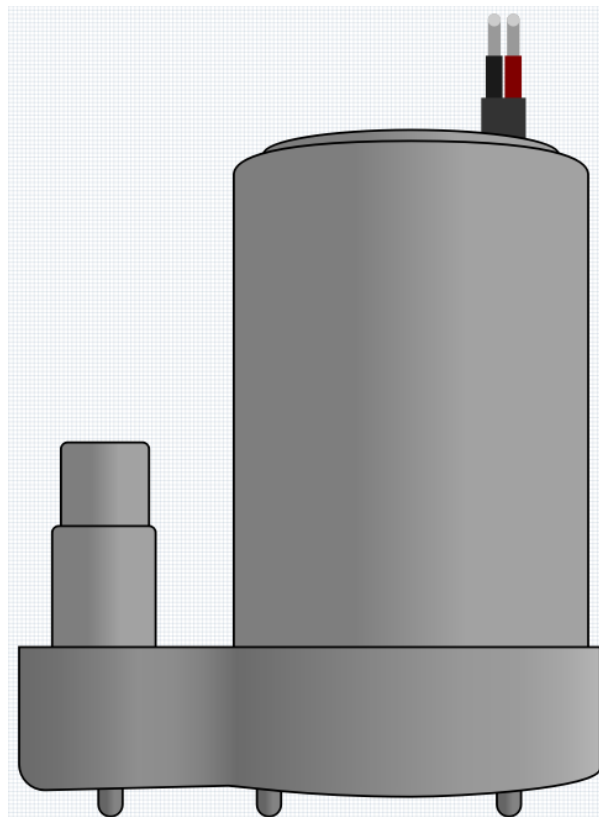
注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なレビュー：新製品の発表や先行レビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.15 遠心ポンプ



遠心ポンプは、回転運動のエネルギーを水流エネルギーに変換して流体を輸送します。この回転エネルギーは電動モータから供給されます。流体は、回転軸に沿ってまたは近くのインペラに入り、インペラによって加速され、放射状に外向きにディフューザまたはボルト室に流れ、そこから流出します。

遠心ポンプの一般的な用途には、水、下水、農業、石油、石油化学のポンピングが含まれます。

- [遠心ポンプ - Wikipedia](#)

特徴

- 電圧範囲: DC 3 ~ 4.5V
- 動作電流: 120 ~ 180mA
- 消費電力: 0.36 ~ 0.91W
- 最大揚水高: 0.35 ~ 0.55M
- 最大流量: 80 ~ 100 L/H
- 連続稼働寿命: 100 時間
- 防水等級: IP68
- 駆動モード: DC, 磁気駆動
- 材質: エンジニアリングプラスチック
- 排水口外径: 7.8 mm
- 排水口内径: 6.5 mm
- これは水中ポンプであり、そのように使用するべきです。水中でない状態で動かすと、過度に加熱してオーバーヒートのリスクがあります。

例

- [1.4 ポンピング \(基本プロジェクト\)](#)
- [6. 植物モニター \(IoT プロジェクト\)](#)

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び & 共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。

- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.16 L9110 モータードライバーモジュール

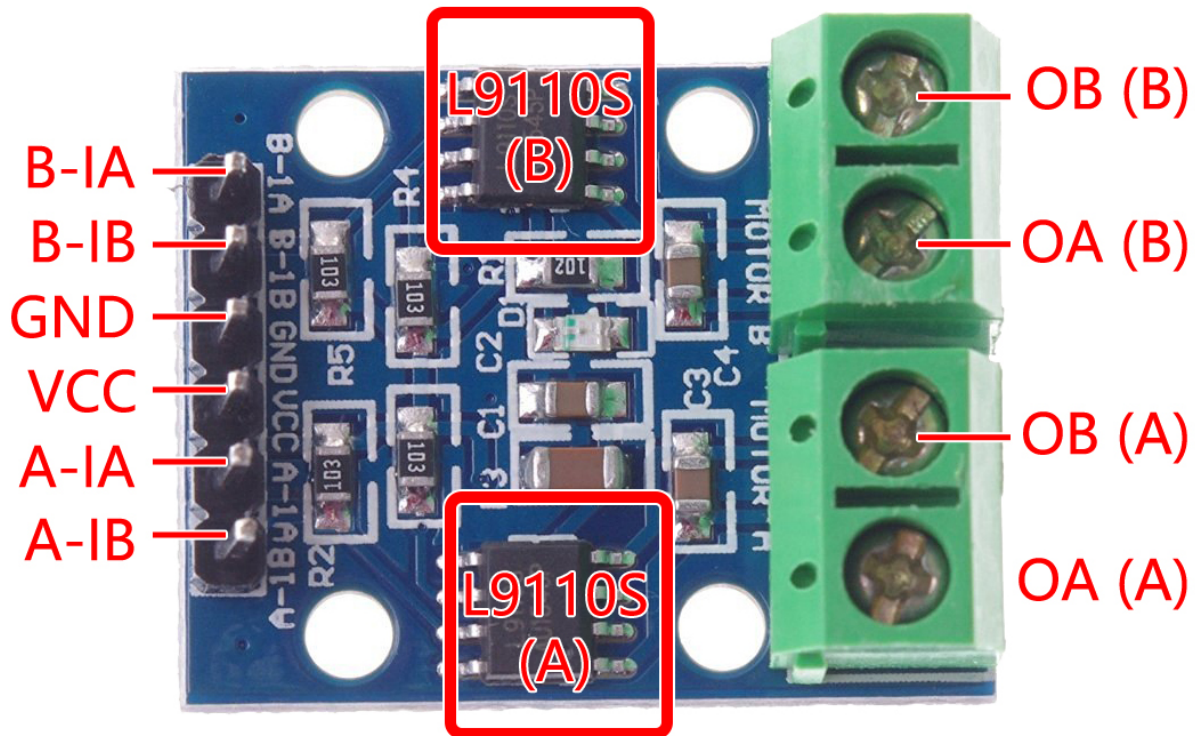
L9110 モータードライバーモジュールは、2 つのモーターを同時に駆動するのに適しています。このモジュールには 2 つの独立した L9110S ドライバーチップが搭載されており、各チャンネルで最大 800mA の安定した電流出力を誇ります。

2.5V から 12V の電圧範囲を持ち、3.3V および 5V のマイクロコントローラーともに適合します。

L9110 モータードライバーモジュールは、幅広いアプリケーションでのモーター制御を容易にするシンプルなソリューションとして機能します。デュアルチャンネルのアーキテクチャのおかげで、二つのモーターの独立した制御が可能になり、2 つのモーター操作が重要なプロジェクトに理想的です。

持続的な強力な電流出力を持つこのモジュールは、小型から中型のモーターを確実に駆動し、ロボティクス、オートメーション、モーター中心の取り組みに多様性をもたらします。広い電圧範囲はさらに適応性を注入し、さまざまな電源設定と整合します。

ユーザーフレンドリーに設計されたこのモジュールは、マイクロコントローラーや類似の制御デバイスへの接続を簡単にする直感的な入力および出力端子を提供します。さらに、安全面を疎かにしていません。統合された過電流および過熱保護機能が、モーター操作の信頼性と安全性を強化します。



- B-1A & B-1B(B-2A): モーター B の回転方向を制御するための入力ピン。
- A-1A & A-1B: モーター A の回転方向を制御するための入力ピン。
- OA & OB(A): モーター A の出力ピン。
- OA & OB(B): モーター B の出力ピン。
- VCC: 電源入力ピン (2.5V-12V)。
- GND: グランドピン。

特徴

- ボード上の 2 つの L9110S モーター制御チップ
- デュアルチャネルモーター制御。
- 独立したモーターの回転方向制御。
- 高電流出力 (チャネルあたり 800mA)。
- 広い電圧範囲 (2.5V-12V)。
- コンパクトな設計。
- 便利な入力および出力端子。
- 組み込みの保護機能。

- 多目的な応用。
- PCB サイズ: 29.2mm x 23mm
- 動作温度: -20 ° C ~ 80 ° C
- 電源オン LED インジケータ

動作原理

こちらはモーター B の真理値表です:

この真理値表は、入力ピン B-1A および B-1B(B-2A) の値に基づくモーター B のさまざまな状態を示しています。これにより、モーター B の回転方向 (時計回りまたは反時計回り)、ブレーキ、または停止を示します。

B-1A	B-1B(B-2A)	モーター B の状態
1	0	時計回りに回転
0	1	反時計回りに回転
0	0	ブレーキ
1	1	停止

こちらはモーター A の真理値表です:

この真理値表は、入力ピン A-1A および A-1B の値に基づくモーター A のさまざまな状態を示しています。これにより、モーター A の回転方向 (時計回りまたは反時計回り)、ブレーキ、または停止を示します。

A-1A	A-1B	モーター A の状態
1	0	時計回りに回転
0	1	反時計回りに回転
0	0	ブレーキ
1	1	停止

- [1.3 車輪を回す](#) (基本プロジェクト)
- [1. 移動](#) (カープロジェクト)
- [3. 速度の上昇](#) (カープロジェクト)
- [8. IoT カー](#) (IoT プロジェクト)

コントローラー

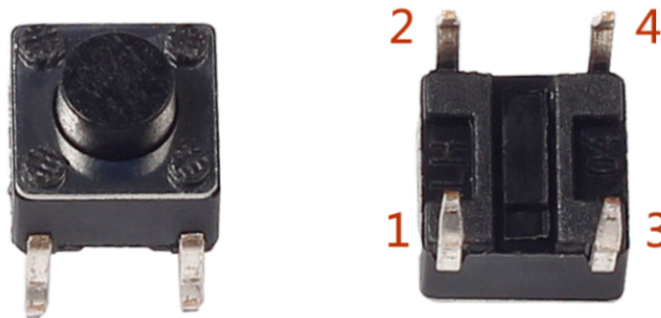
注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ! Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なレビュー：新製品の発表や先行レビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

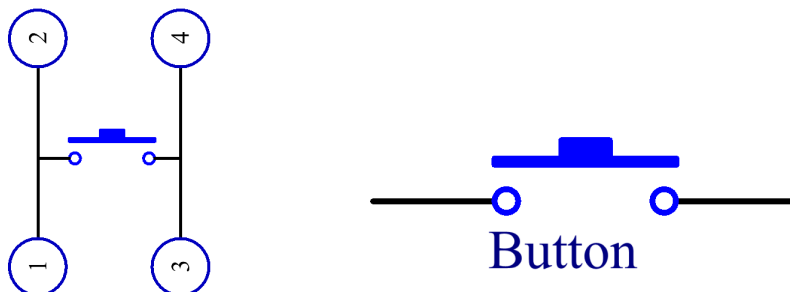
私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.17 ボタン

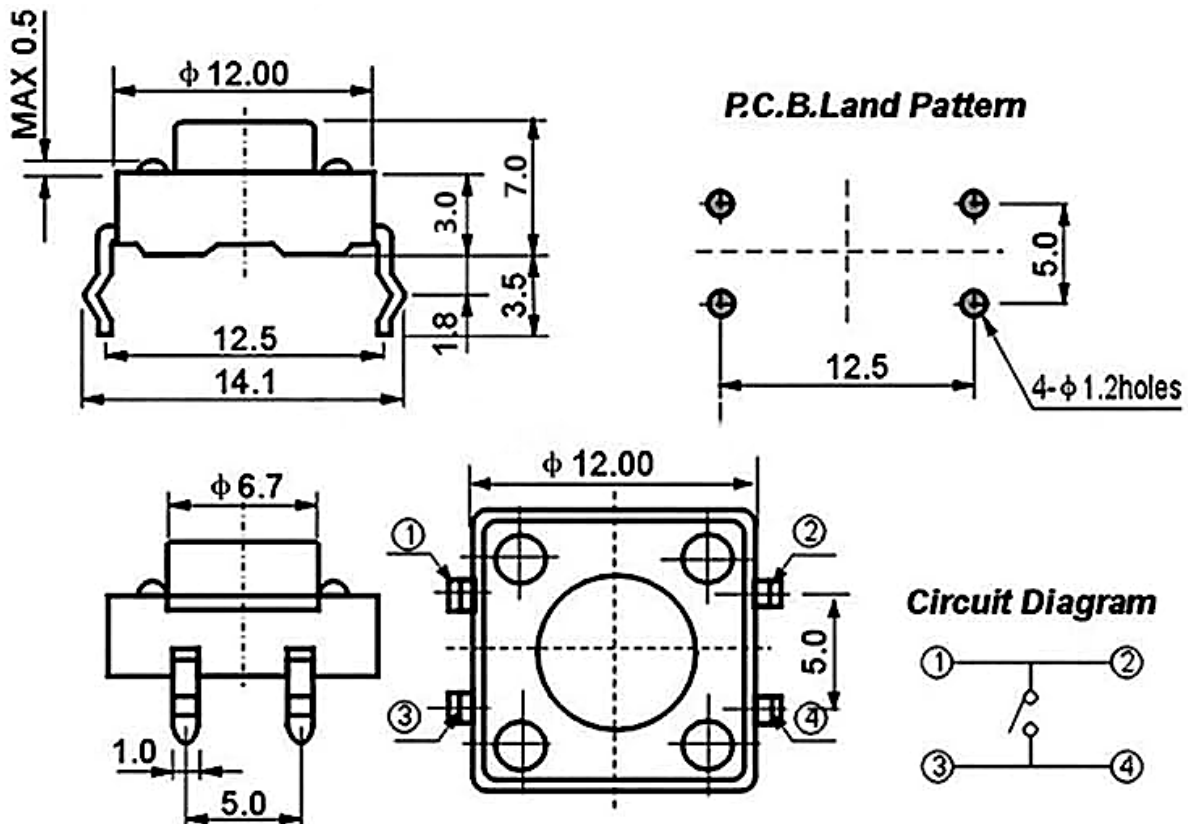


ボタンは電子機器を制御するための一般的なコンポーネントです。主にスイッチとして使用され、回路を接続または切断するために使用されます。ボタンにはさまざまなサイズや形状がありますが、ここで使用するのは次の画像に示す 6mm のミニボタンです。ピン 1 はピン 2 に、ピン 3 はピン 4 に接続されています。したがって、ピン 1 またはピン 2 のいずれかをピン 3 またはピン 4 に接続するだけです。

以下はボタンの内部構造です。下の右側の記号は、回路内でボタンを表すために通常使用されるものです。



ピン 1 がピン 2 に、ピン 3 がピン 4 に接続されているため、ボタンが押されると、4 つのピンが接続され、回路が閉じます。



例

- 3.1 ボタンの値の読取り (基本プロジェクト)

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.18 リードスイッチ

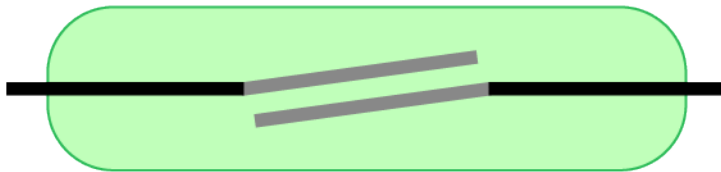


リードスイッチは、磁場を適用することで動作する電気スイッチです。これは、1936 年に Bell Telephone Laboratories の Walter B. Ellwood によって発明され、1940 年 6 月 27 日にアメリカ合衆国特許第 2264746 号として特許を取得しました。

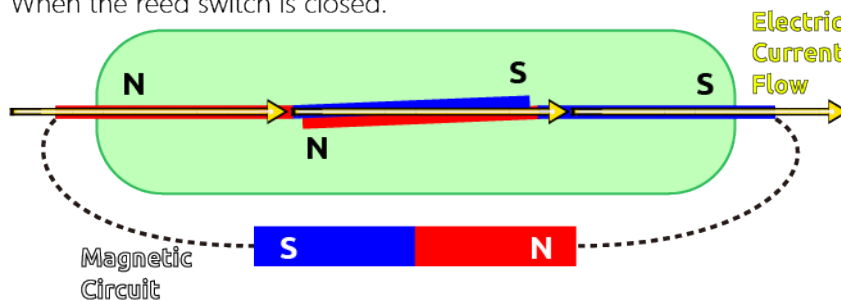
リードスイッチの動作原理は非常に単純です。2つのリード（通常は鉄とニッケル、2つの金属で作られる）が、端部で重なるようにガラス管の中に封入されています。2つのリードは、わずかに数マイクロンの小さいギャップを持って重なって分離されています。ガラス管は高純度の不活性ガス（例：窒素）で満たされており、一部のリードスイッチは、高電圧性能を高めるために内部に真空が作られています。

リードは磁束の伝導体として動作します。2つのリードは動作していない状態では接触していませんが、永久磁石や電磁コイルによって生成される磁場を通過すると、適用された磁場によって2つのリードの端部近くで異なる極性が生じ、磁気力がリード自体のばね力を超えると、2つのリードは引き合い、回路を伝導します。磁場が弱くなるか消失すると、リードは自らの弾性のために放出され、接触面は回路を開くために分離します。

When the reed switch is open.



When the reed switch is closed.



- [リードスイッチ - Wikipedia](#)

例

- [3.2 磁気を感じる \(基本プロジェクト\)](#)
- [7. 数量制限ゲート \(IoT プロジェクト\)](#)

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook

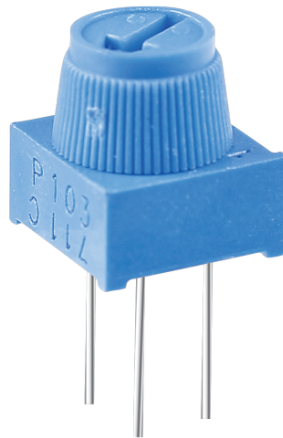
上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び & 共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.19 ポテンショメーター

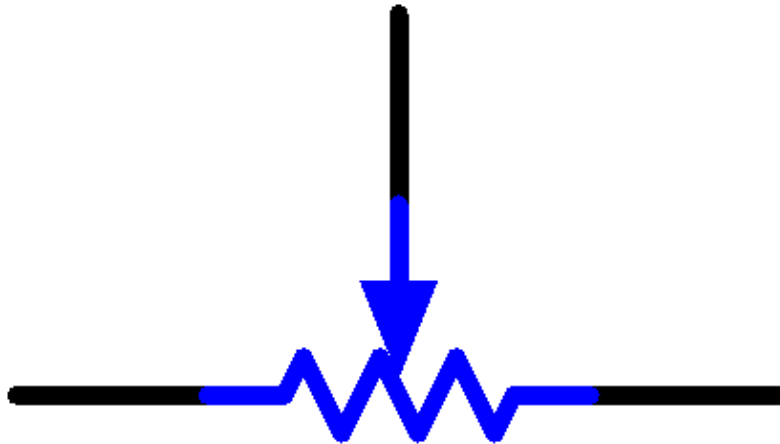


ポテンショメーターは3つの端子を持つ抵抗成分でもあり、その抵抗値は一定の変動に応じて調整することができます。

ポテンショメーターにはさまざまな形、サイズ、値がありますが、以下の共通点があります。

- 3つの端子（または接続点）を持っています。
- 抵抗を変化させるためのノブ、ネジ、またはスライダーがあります。これは中央の端子と外側の端子の一方との間の抵抗を変化させるために動かすことができます。
- ノブ、ネジ、またはスライダーが動かされると、中央の端子と外側の端子の一方との間の抵抗は0 からポテンショメーターの最大抵抗まで変わります。

以下は、ポテンショメーターの回路記号です。



回路におけるポテンショメーターの機能は以下の通りです：

1. 電圧分配器としての役割

ポテンショメーターは、連続的に調整可能な抵抗器です。ポテンショメーターのシャフトやスライドハンドルを調整すると、可動接点が抵抗体上をスライドします。このとき、ポテンショメーターに印加される電圧と、可動アームが回転した角度や移動した距離に応じて電圧が出力されます。

2. レオスタットとしての役割

ポテンショメーターをレオスタットとして使用する場合、中央のピンと他の2つのピンのいずれかを回路に接続します。こうすることで、移動接触の移動範囲内でスムーズに変化する抵抗値を取得できます。

3. 電流コントローラーとしての役割

ポテンショメーターが電流コントローラーとして動作する場合、スライド接触端子を出力端子の一方として接続する必要があります。

ポテンショメーターについての詳細はこちらを参照してください：[ポテンショメーター - Wikipedia](#)

例

- [4.1 ノブを回す](#) (基本プロジェクト)

注釈： こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

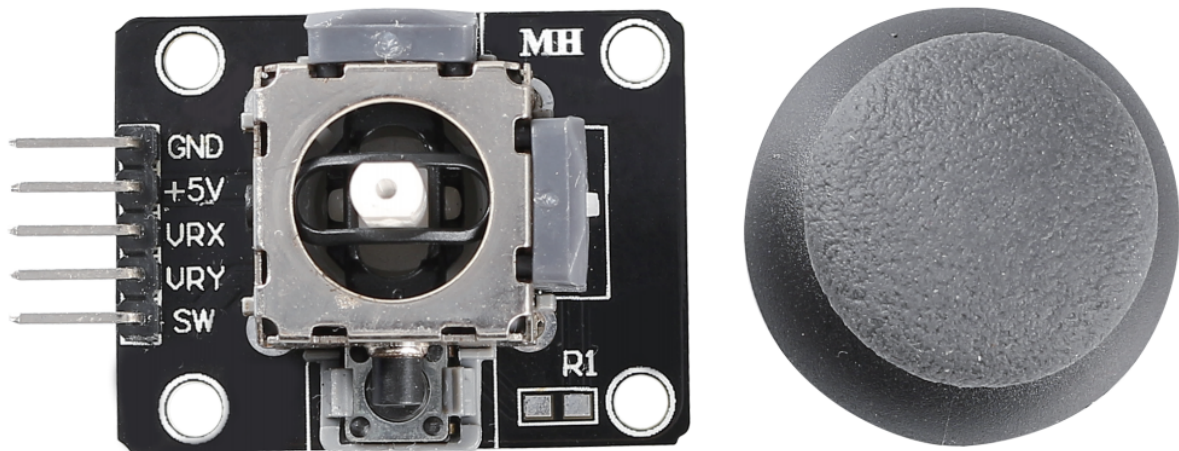
参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び & 共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。

- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.20 ジョイスティックモジュール

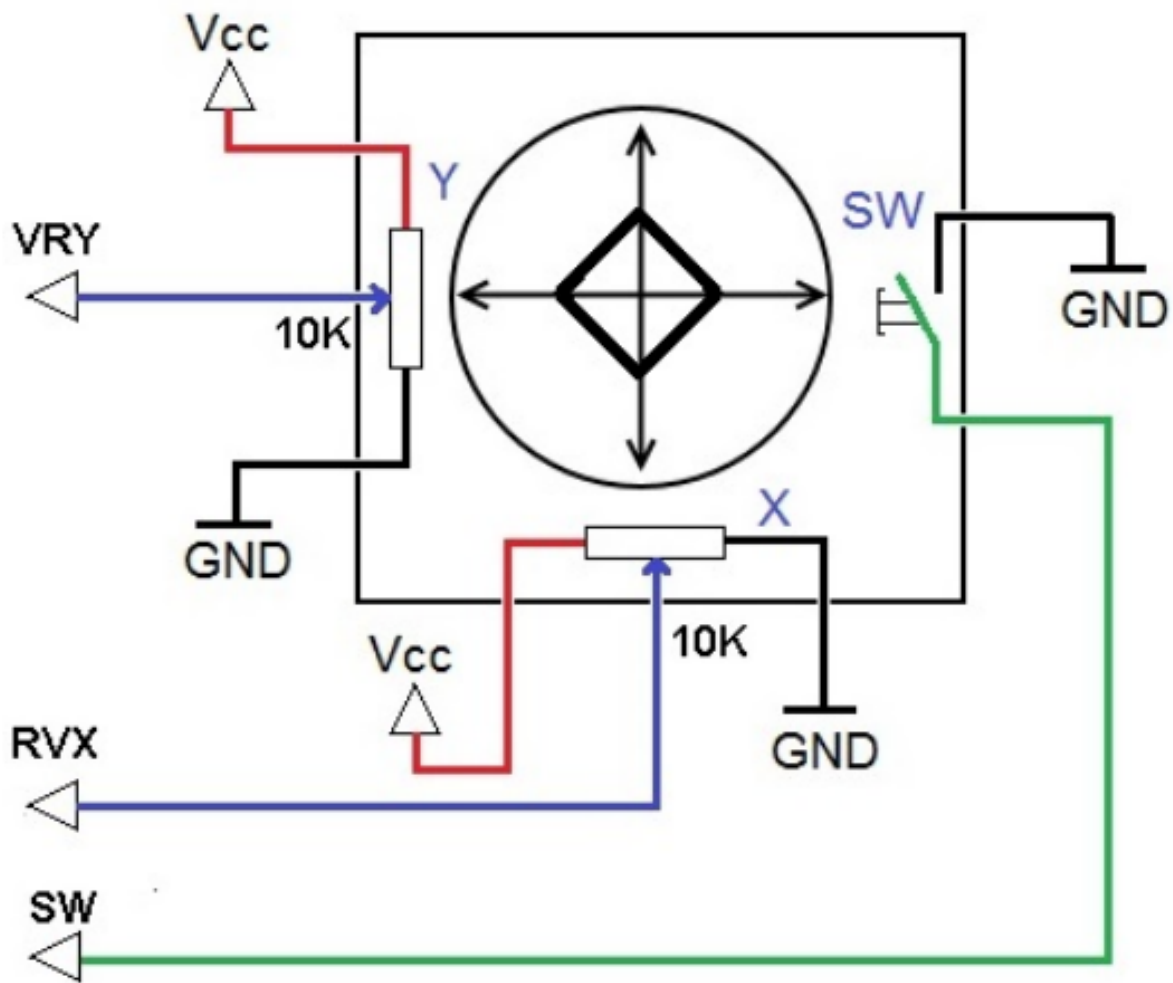


ジョイスティックの基本的な考え方は、スティックの動きをコンピュータが処理できる電子情報に変換することです。

コンピュータに完全な動きの範囲を伝えるために、ジョイスティックはスティックの位置を2つの軸、X軸（左から右）とY軸（上から下）で測定する必要があります。基本的な幾何学と同じように、X-Y座標はスティックの位置を正確に特定します。

スティックの位置を特定するために、ジョイスティックの制御システムは各シャフトの位置を単に監視します。従来のアナログジョイスティックの設計は、二つのポテンショメータ、または可変抵抗で行います。

ジョイスティックには、ジョイスティックが押されたときに作動するデジタル入力もあります。



例

- 4.3 ジョイスティックの操作 (基本プロジェクト)

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

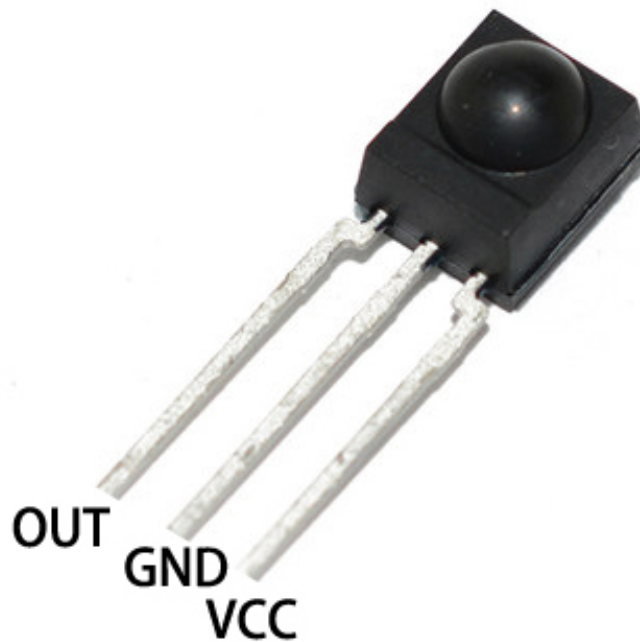
参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [] をクリックして今すぐ参加しましょう！

1.21 IR 受信機

IR 受信機



- OUT: 信号出力
- GND: GND
- VCC: 電源供給, 3.3v~5V

赤外線受信機は、赤外線の信号を受け取り、独立して赤外線を受信して TTL レベルと互換性のある信号を出力する部品です。サイズは一般的なプラスチック製トランジスタと同じで、あらゆる種類の赤外線リモコンや赤外線伝送に適しています。

赤外線、または IR 通信は、低コストで使いやすいワイヤレス通信技術として人気があります。赤外線は可視光よりもわずかに波長が長いので、人間の目には感知できない - ワイヤレス通信に理想的です。赤外線通信の一般的な変調方式は 38KHz 変調です。

- 採用されている [HS0038B](#) IR 受信センサー、高感度
- リモコンとして使用可能
- 電源供給: 5V
- インターフェース: デジタル

- 変調周波数: 38Khz
- ピン定義: (1) 出力 (2) Vcc (3) GND
- サイズ: 23.5mm x 21.5mm

リモコン



これは、21 の機能ボタンと最大 8 メートルの送信距離を持つミニ薄型の赤外線ワイヤレスリモコンで、子供の部屋の様々なデバイスを操作するのに適しています。

- サイズ: 85x39x6mm
- リモコン範囲: 8-10m
- バッテリー: 3V ボタン型リチウムマンガン電池
- 赤外線キャリア周波数: 38KHz
- 表面貼り材料: 0.125mm PET
- 有効寿命: 20,000 回以上

例

- [5.11.2 IR 受信機](#) (基本プロジェクト)
- [9. リモートコントロール](#) (車のプロジェクト)
- [10. ワンタッチスタート](#) (車のプロジェクト)

センサー

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

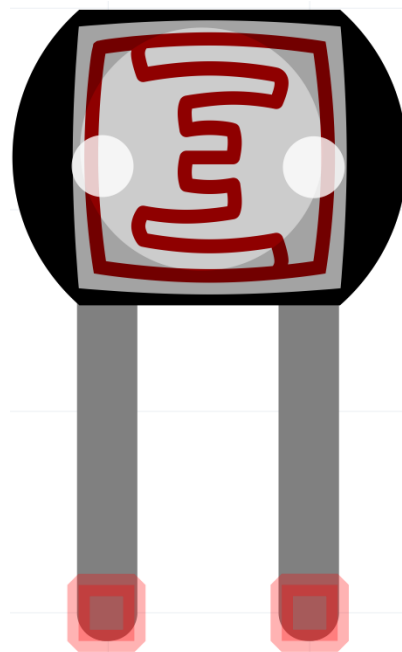
参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。

- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

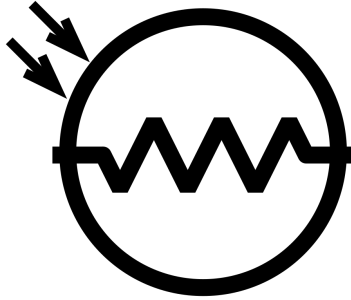
1.22 フォトレジスタ



フォトレジスタ、またはフォトセルは、光に応じて変わる抵抗値を持つ変数抵抗器です。フォトレジスタの抵抗は、光の強度が増えると減少します。言い換えれば、光電導性を示します。

フォトレジスタは、光感知検出回路や光活性・暗活性のスイッチ回路に適用できる抵抗半導体として動作します。暗い場所では、フォトレジスタの抵抗は数メガオーム（M Ω ）に達することができますが、明るい場所では、数百オームまで低下することがあります。

以下はフォトレジスタの電子シンボルです。



- [フォトレジスタ - Wikipedia](#)

例

- [4.2 光を感じる \(基本プロジェクト\)](#)
- [5. 家の環境モニタリング \(IoT プロジェクト\)](#)
- [6. 植物モニター \(IoT プロジェクト\)](#)

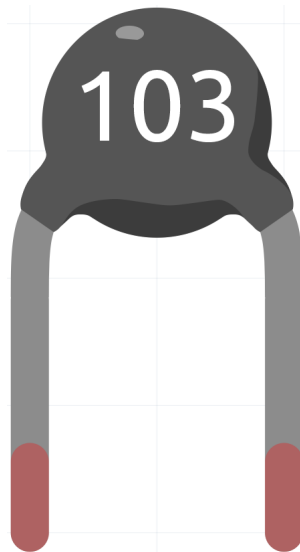
注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

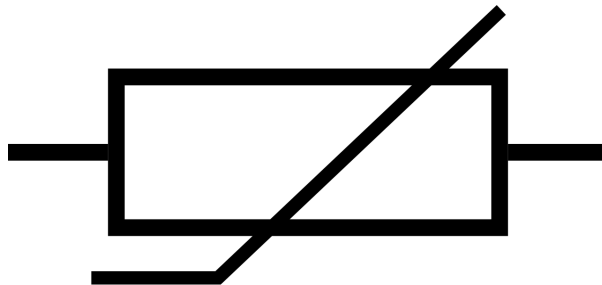
1.23 サーミスタ



サーミスタは、標準の抵抗器よりも、温度に強く依存する抵抗値を持つタイプの抵抗器です。この言葉は thermal（熱的な）と resistor（抵抗器）の組み合わせから成り立っています。サーミスタは、起動電流制限器、温度センサー（通常は NTC 型）、自己リセット型の過電流保護装置、自己調整型加熱要素（通常は PTC 型）として広く使用されています。

- [サーミスタ - Wikipedia](#)

以下はサーミスタの電子記号です。



サーミスタには、基本的に 2 つの対照的なタイプがあります：

- NTC サーミスタでは、温度が上昇すると、価電子帯からの熱振動により、導電電子が増加するため、抵抗が減少します。NTC は、温度センサーとして、または回路の直列としての起動電流制限器として一般的に使用されます。
- PTC サーミスタでは、温度が上昇すると、特に不純物や欠陥の熱格子振動が増加するため、抵抗が増加します。PTC サーミスタは、回路と直列に取り付けられ、過電流状態から保護するリセッタブルヒューズとして一般的に使用されます。

このキットでは、NTC 型を使用しています。各サーミスタには通常の抵抗があります。ここでは、25 度セルシウ

スで測定される 10k オームです。

抵抗と温度の関係は以下のとおりです：

$$RT = RN * \exp(B(1/TK - 1/TN))$$

- **RT** は、温度が TK のときの NTC サーミスタの抵抗。
- **RN** は、定格温度 TN 下での NTC サーミスタの抵抗。ここでは、RN の数値は 10k です。
- **TK** は、ケルビン温度で、単位は K。ここでは、TK の数値は 273.15 + 度数セルシウス。
- **TN** は、定格ケルビン温度で、単位も K。ここでは、TN の数値は 273.15+25 です。
- **B(beta)** は、NTC サーミスタの材料定数であり、熱感受性指数とも呼ばれ、数値は 3950 です。
- **exp** は指数関数の略で、ベース数 e は自然数で、約 2.7 に等しい。

この式 $TK=1/(\ln(RT/RN)/B+1/TN)$ を変換して、ケルビン温度から 273.15 を減算して度数セルシウスを得ます。

この関係式は、実験式です。温度と抵抗が有効範囲内のときのみ正確です。

例

- [6.3 高温アラーム](#) (基本プロジェクト)
- [4.5 温度計](#) (基本プロジェクト)

注釈： こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

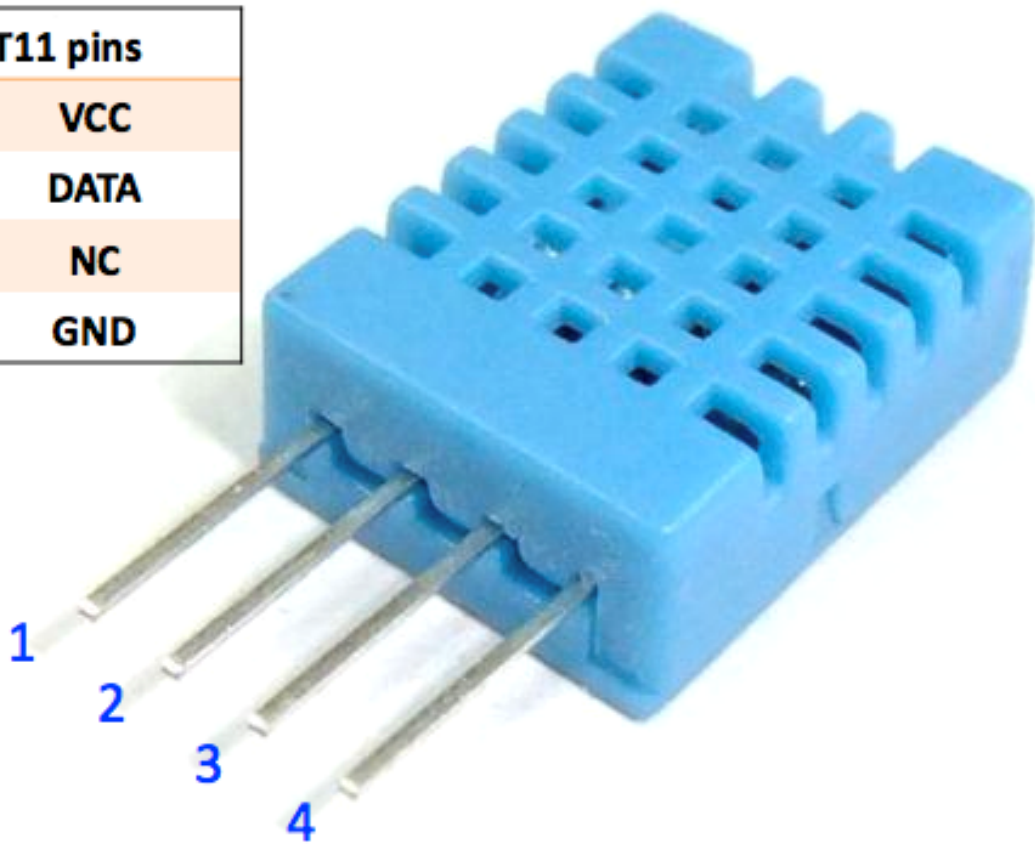
1.24 DHT11 温湿度センサ

デジタル温湿度センサ DHT11 は、温度と湿度の校正されたデジタル信号出力を持つ複合センサです。専用のデジタルモジュールの収集技術と温湿度センシング技術が組み合わせられており、製品の高い信頼性と優れた長期安定性を保証しています。

このセンサには、抵抗湿度感知要素と NTC 温度計測装置が組み込まれており、高性能の 8 ビットマイクロコントローラーと接続されています。

利用可能なピンは 3 つだけです：VCC、GND、DATA。通信プロセスは、DATA ラインが DHT11 に開始信号を送信することから始まり、DHT11 はこれを受信し、応答信号を返します。次に、ホストが応答信号を受信し、40 ビットの温湿度データ（8 ビット湿度整数 + 8 ビット湿度小数 + 8 ビット温度整数 + 8 ビット温度小数 + 8 ビットチェックサム）の受信を開始します。

DHT11 pins	
1	VCC
2	DATA
3	NC
4	GND



特徴

1. 湿度測定範囲: 20 - 90%RH
2. 温度測定範囲: 0 - 60
3. 温度と湿度を示すデジタル信号の出力

4. 動作電圧: DC 5V; PCB サイズ: 2.0 x 2.0 cm

5. 湿度測定精度: $\pm 5\%$ RH

6. 温度測定精度: ± 2

- [DHT11 データシート](#)

例

- [5.11.3 温度 - 湿度 \(基本プロジェクト\)](#)
- [5. 家の環境モニタリング \(IoT プロジェクト\)](#)
- [6. 植物モニター \(IoT プロジェクト\)](#)

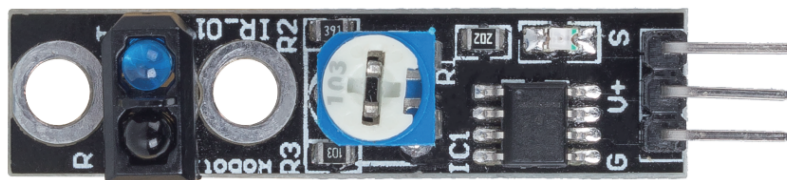
注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.25 ライントラッキングモジュール



- S: 通常は低レベル、黒いラインが検出されたときには高レベル。
- V+: 電源供給、3.3v~5V
- G: グ라운드

このモジュールは 1 チャンネルのライントラッキングモジュールで、名前が示す通り、白い背景上の黒いラインや黒い背景上の白いラインを追跡します。



モジュールには TCRT5000 赤外線センサが使用されており、赤外線 LED（青色）と感光トリプレット（黒色）で構成されています。

- 青色の赤外線 LED は、電源が入ると人間の目には見えない赤外線を放射します。
- 赤外線を受け取るために使用される黒いフォトリジスタは、内部に抵抗を持っており、この抵抗は受け取った赤外線の量によって変化します。赤外線を多く受け取れば抵抗は低下し、その逆も同様です。

モジュールには LM393 コンパレータがあり、これはフォトリジスタの電圧とセット電圧（ポテンショメータで調整）を比較するために使用されます。それがセット電圧よりも大きい場合、出力は 1 となり、それ以外の場合は 0 となります。

したがって、赤外線放射管が黒い表面に照射されると、黒は光を吸収するため、感光トリプレットが受け取る赤外線は減少し、その抵抗は増加します（電圧上昇）。LM393 コンパレータを通過後、高レベルを出力します。

同様に、白い表面に照射すると、反射光は増加し、感光トリプレットの抵抗は減少します（電圧低下）；したがって、コンパレータは低レベルを出力し、指示 LED が点灯します。

- **TCRT5000**

特徴

- 赤外線放射センサ TCRT5000 を使用
- 検出距離: 1-8mm、焦点距離 2.5mm
- コンパレータ出力信号がクリアで、波形が良く、駆動能力が 15mA 以上
- 感度調整のためのポテンショメータを使用
- 動作電圧: 3.3V-5V
- デジタル出力: 0（白）と 1（黒）
- 幅広い電圧の LM393 コンパレータを使用。

- サイズ: 42mmx10mm

例

- [3.4 ラインを検出する](#) (基本プロジェクト)
- [4. ライン追従](#) (車のプロジェクト)

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.26 土壌湿度モジュール

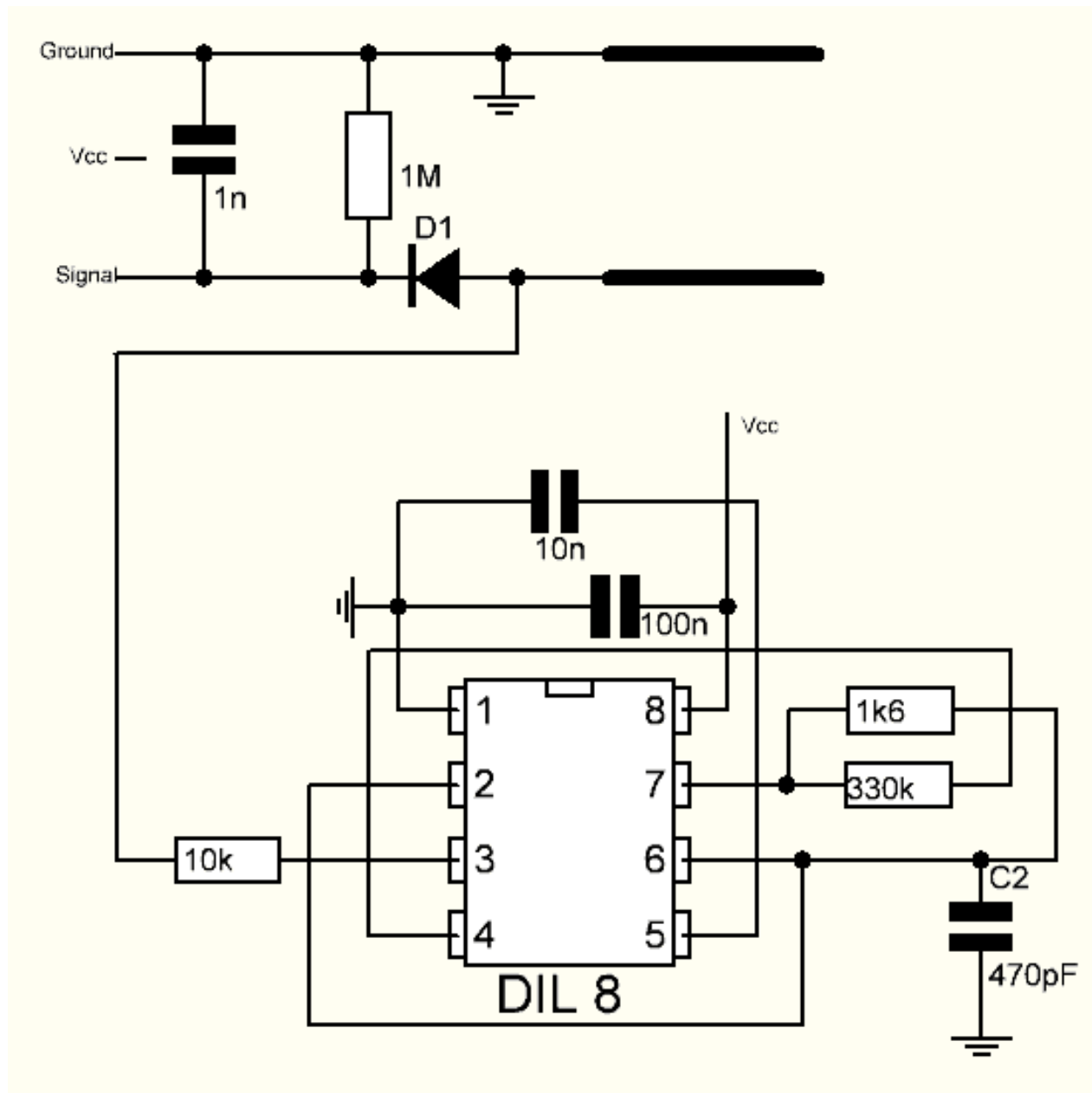


- GND: グラウンド
- VCC: 電源供給、3.3v~5V
- AOUT: 土壌の湿度値を出力し、土が湿っているほど、その値は小さくなります。

この容量性土壌湿度センサは、市場に出回っている抵抗性センサとは異なり、容量性誘導の原理を使用して土壌の湿度を検出します。これにより、抵抗性センサが腐食に非常に敏感である問題を回避し、その作業寿命を大幅に延ばしています。

耐腐食性の材料で作られており、優れた耐用年数があります。植物の周りの土に挿入して、リアルタイムで土壌の湿度データを監視します。モジュールには、3.3 V ~ 5.5 V の電圧範囲で動作するためのオンボード電圧レギュレータが含まれています。3.3 V および 5 V の電源を持つ低電圧のマイクロコントローラに最適です。

容量性土壌湿度センサのハードウェア回路図は以下の通りです。



固定周波数のオシレータがあり、これは 555 タイマー IC で構築されています。生成された矩形波は、キャパシタのようなセンサに供給されます。しかし、矩形波信号の場合、キャパシタは特定のリアクタンスを持っているか、あるいは議論のために、純粋なオーム抵抗器（ピン 3 の 10k 抵抗器）を持つ抵抗器として、電圧分配器を形成します。

土壌の湿度が高いほど、センサの容量が高くなります。その結果、矩形波はリアクタンスが少なく、信号線上の電

圧が低下し、マイクロコントローラを通じてのアナログ入力の値も小さくなります。

仕様

- 動作電圧: 3.3 ~ 5.5 VDC
- 出力電圧: 0 ~ 3.0VDC
- 動作電流: 5mA
- インターフェース: PH2.0-3P
- 寸法: 3.86 x 0.905 インチ (L x W)
- 重量: 15g

例

- [4.4 土壌の湿度を測定する \(基本プロジェクト\)](#)
- [6. 植物モニター \(IoT プロジェクト\)](#)

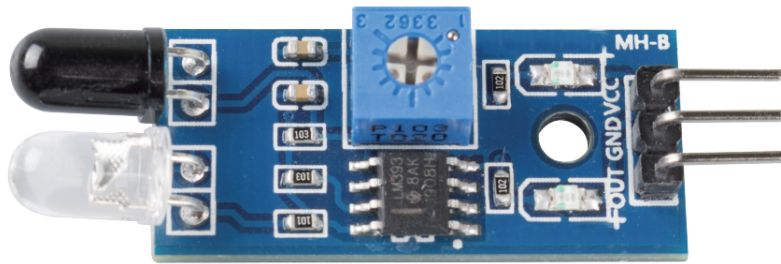
注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.27 障害物回避モジュール

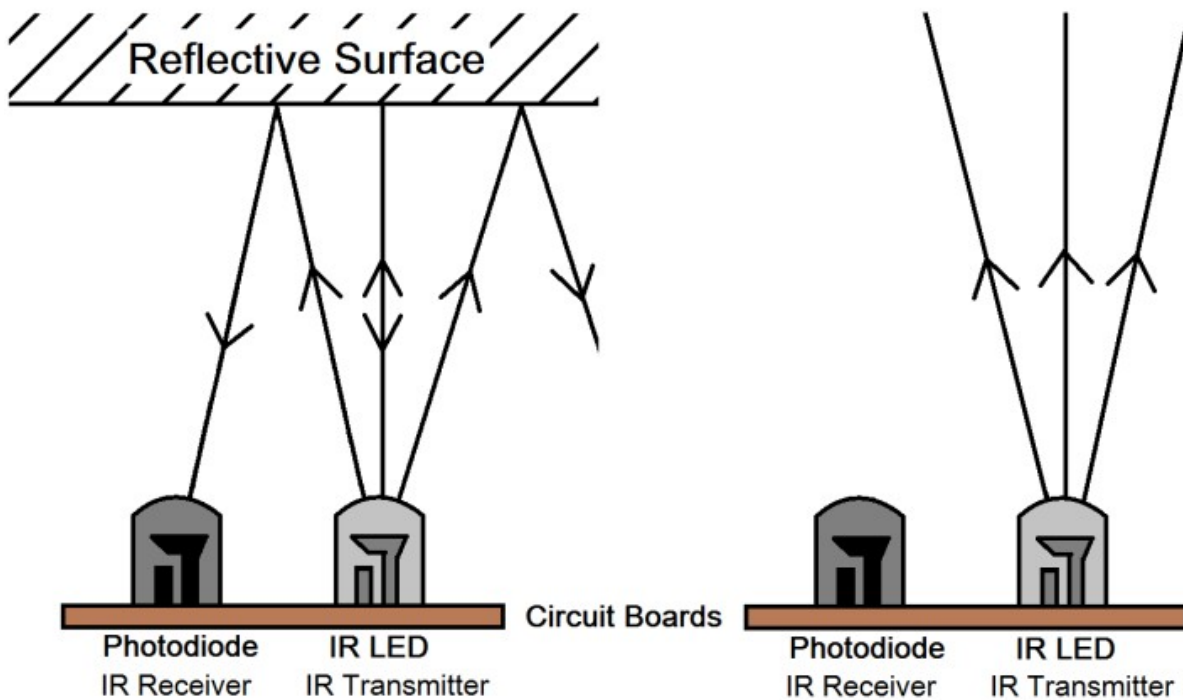


- **VCC:** 電源供給、3.3 ~ 5V DC。
- **GND:** グラウンド
- **OUT:** 信号ピン、通常は高レベルで、障害物を検出すると低レベルになります。

IR 障害物回避モジュールは、環境光に対する適応性が高く、赤外線を送受信管のペアを持っています。

送信管は赤外線の周波数を放出し、検出方向に障害物が存在すると、赤外線の放射が受信管によって受け取られ、コンパレータ回路の処理を経て、インジケータが点灯し、低レベルの信号を出力します。

検出距離はポテンショメータで調整可能で、有効距離の範囲は 2-30cm です。



例

- 3.3 障害物を検出する (基本プロジェクト)
- 5. 障害物回避モジュールを使って遊ぼう (車プロジェクト)

- 8. 自動運転車 (車プロジェクト)
- 7. 数量制限ゲート (IoT プロジェクト)

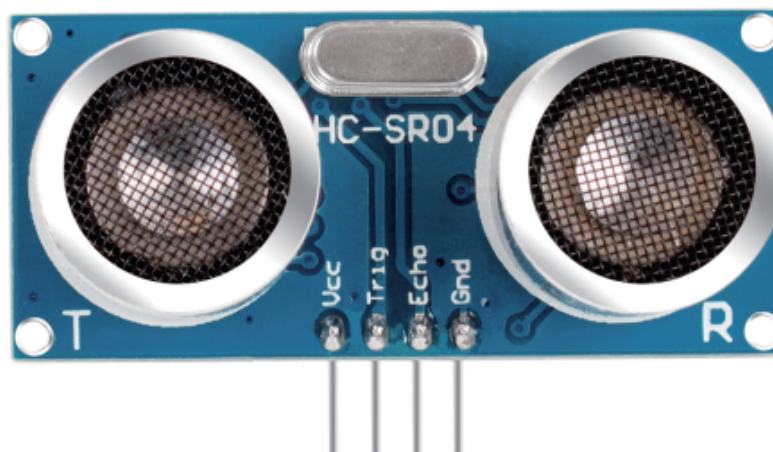
注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

1.28 超音波モジュール



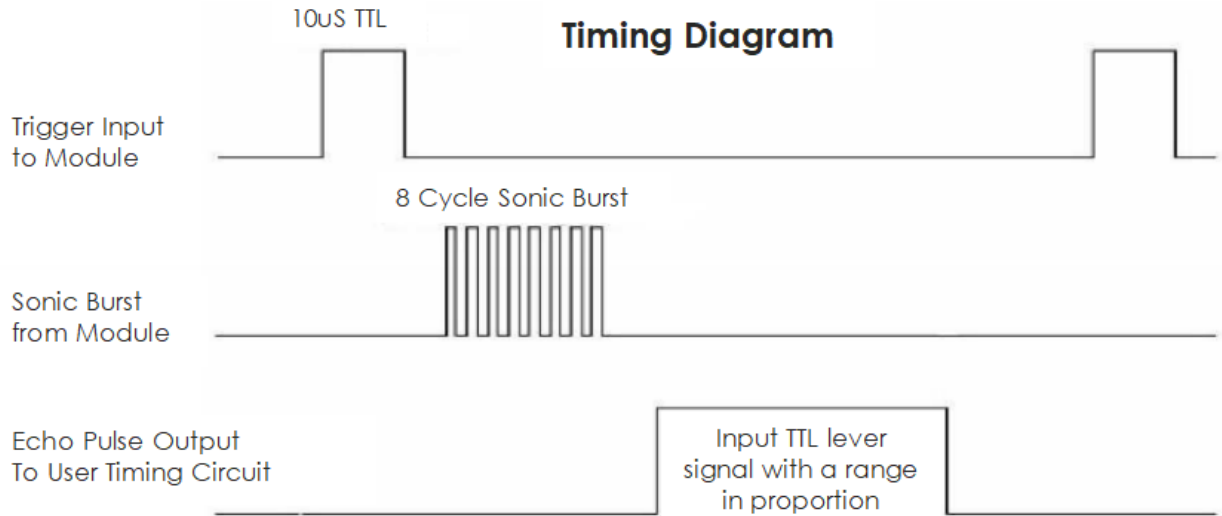
超音波測距モジュールは、2cm - 400cm の非接触測定機能を提供し、測距の精度は 3mm に達することができます。5m 以内では信号が安定しており、5m を超えると徐々に信号が弱まり、7m の位置で消失します。

このモジュールには、超音波の送信機、受信機、および制御回路が含まれています。基本的な原理は以下のとおりです：

1. IO トリガで 10us 以上の高レベル信号を処理します。
2. モジュールは自動的に 8 つの 40kHz の波を送信し、パルス信号の返信があるかどうかを検出します。

3. 信号が返ってきた場合、高レベルを通過し、高出力 IO の持続時間は、超音波の送信から返信までの時間です。ここで、測定距離 = (高い時間 x 音速 (340 m / s) / 2)。

以下は、タイミングダイアグラムです。



測距を開始するには、トリガ入力に短い 10us のパルスを供給するだけで、その後、モジュールは 40kHz での超音波の 8 サイクルバーストを送信し、エコーを上げます。トリガ信号の送信とエコー信号の受信の時間間隔を通じて、範囲を計算することができます。

式: $us / 58 = \text{センチメートル}$ または $us / 148 = \text{インチ}$; または: $\text{範囲} = \text{高レベル時間} * \text{速度} (340M/S) / 2$; トリガ信号とエコー信号の信号衝突を防ぐために、60ms 以上の測定周期を使用することをおすすめします。

例

- [5.8 ユーザー定義関数 \(基本プロジェクト\)](#)
- [7. 手を追う車 \(車プロジェクト\)](#)
- [6. 超音波モジュールで遊ぼう \(車プロジェクト\)](#)

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。

- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

第 2 章

Arduino のスタートガイド

Arduino について知らない方も多いでしょう。以下の言葉を知っていただきたいです：電子工学、デザイン、プログラミング、さらにはメーカー。これらの言葉は私たちから遠いように思えますが、実際にはそうではありません。なぜなら、Arduino は私たちをプログラミングの世界に導き、メーカーとしての夢を実現させてくれるからです。このセッションでは以下を学びます：

- Arduino とは何か？
- Arduino で何ができるか？
- Arduino プロジェクトをどのように作成するか？

2.1 Arduino とは何か？

まず、Arduino について簡単に紹介します。

Arduino は便利で柔軟性があり、使いやすいオープンソースの電子プロトタイピングプラットフォームで、さまざまなモデルのハードウェア Arduino ボードとソフトウェア Arduino IDE が含まれています。エンジニアのための迅速なプロトタイピングだけでなく、アーティスト、デザイナー、趣味人向けであり、現代のメーカーにとっては必須のツールです。

Arduino は非常に大きなシステムです。ソフトウェア、ハードウェア、共通の趣味を持つことで協力できる大規模なオンラインコミュニティがあります。Arduino ファミリーの各メンバーは、知恵を使って、手で作成し、次々と素晴らしい発明を共有しています。そして、あなたもその一部となることができます。

2.2 Arduino で何ができるか？

これについて、Arduino が実際に何を達成できるのか疑問に思うかもしれません。端的に言うと、Arduino はあなたの問題をすべて解決します。

技術的に言うと、Arduino はプログラム可能なロジックコントローラーです。リモコンの車、ロボットアーム、バイオニックロボット、スマートホームなど、多くのエキサイティングで創造的な電子制作を作成するために使用できる開発ボードです。

Arduino ボードは直感的でシンプルでありながら強力で、学生、メーカー、さらには専門のプログラマーにも適しています。

今日までに、世界中の電子愛好者たちは Arduino 開発ボードをベースに創造的な電子制作を続けて開発しています。

2.3 Arduino プロジェクトの構築方法

以下の手順に従って、ゼロから Arduino を使用方法を学びましょう！

注釈： こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

2.3.1 Arduino IDE 2.0 のダウンロードとインストール

Arduino IDE (Arduino 統合開発環境としても知られています) は、Arduino プロジェクトを完成させるために必要なソフトウェアサポートを全て提供します。これは Arduino のために特別に設計されたプログラミングソフトウェアで、Arduino チームによって提供されており、プログラムを書き、それを Arduino ボードにアップロードすることができます。

Arduino IDE 2.0 はオープンソースプロジェクトです。これは頑丈な前身である Arduino IDE 1.x からの大きなステップであり、一新された UI、改良されたボード&ライブラリマネージャ、デバッガ、オートコンプリート機能などが搭載されています。


このチュートリアルでは、Windows、Mac、Linux コンピュータに Arduino IDE 2.0 をダウンロードしてインストールする方法を紹介します。

必要システム条件

- Windows - Win 10 以降、64 ビット
- Linux - 64 ビット
- Mac OS X - バージョン 10.14 "Mojave"以降、64 ビット

Arduino IDE 2.0 のダウンロード

1. にアクセスしてください。
2. お使いの OS バージョンの IDE をダウンロードしてください。



 **Arduino IDE 2.0.0**

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

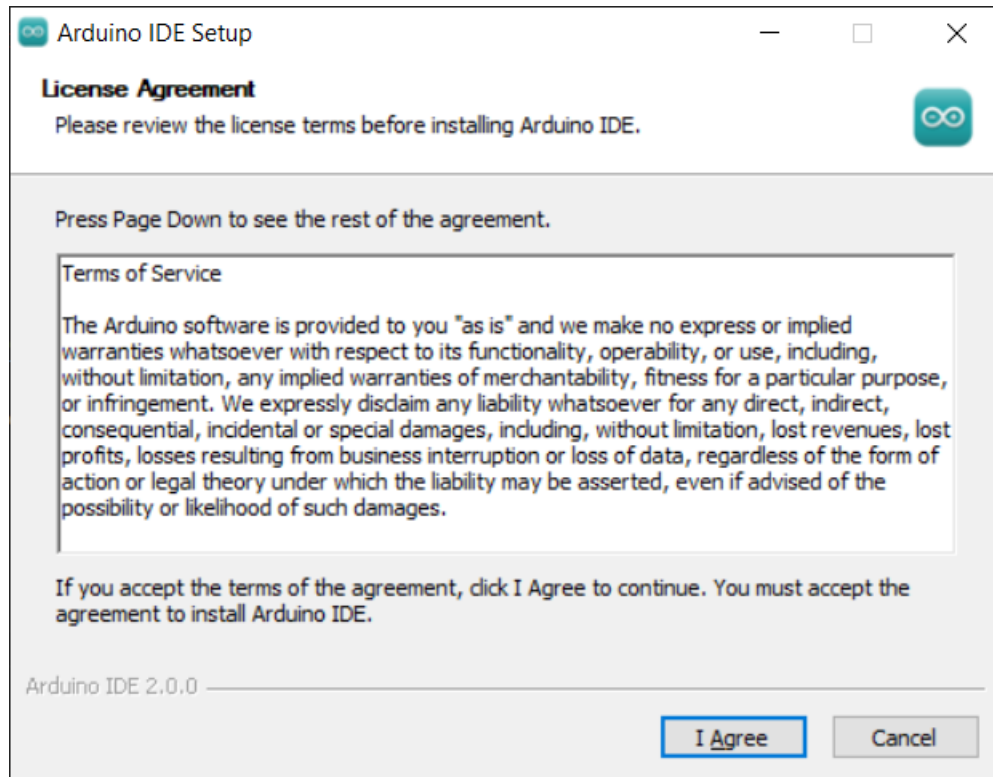
DOWNLOAD OPTIONS

- Windows** Win 10 and newer, 64 bits
- Windows** MSI installer
- Windows** ZIP file
- Linux** AppImage 64 bits (X86-64)
- Linux** ZIP file 64 bits (X86-64)
- macOS** 10.14: "Mojave" or newer, 64 bits

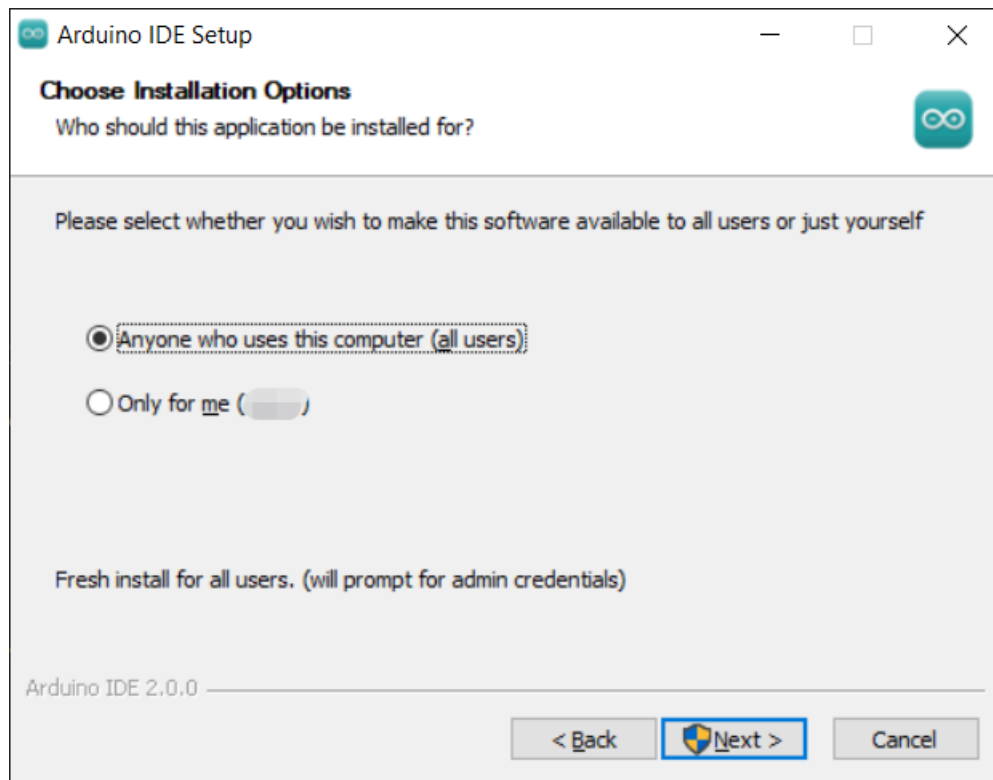
インストール方法

Windows

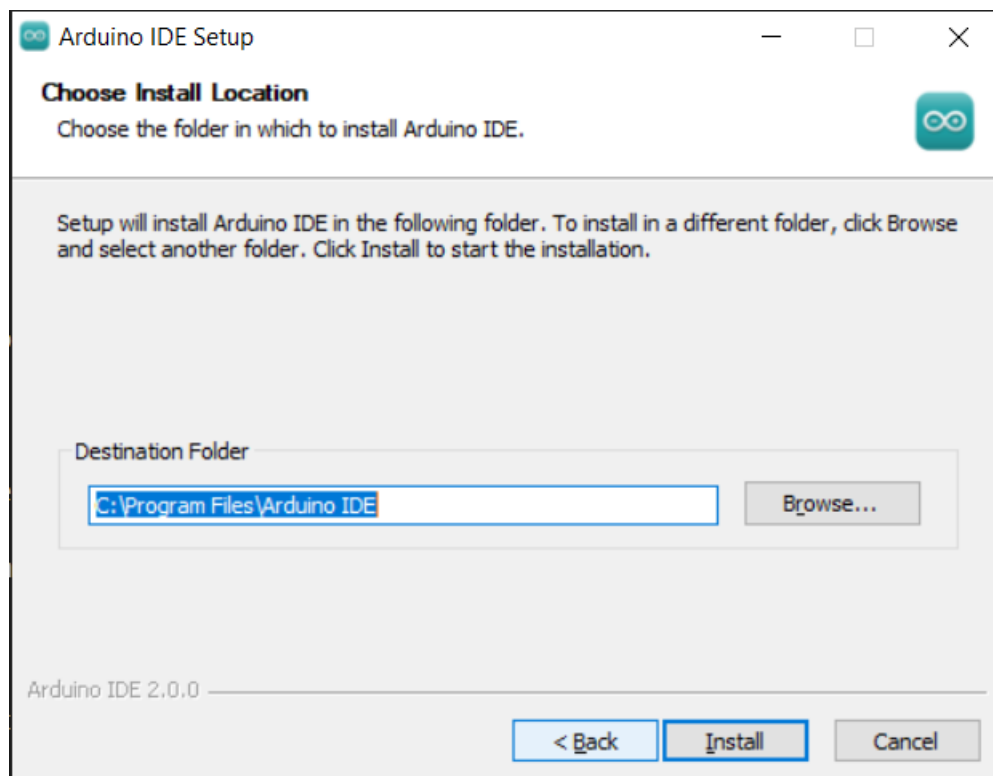
1. ダウンロードした arduino-ide_xxxx.exe ファイルをダブルクリックします。
2. ライセンス契約を読み、同意してください。



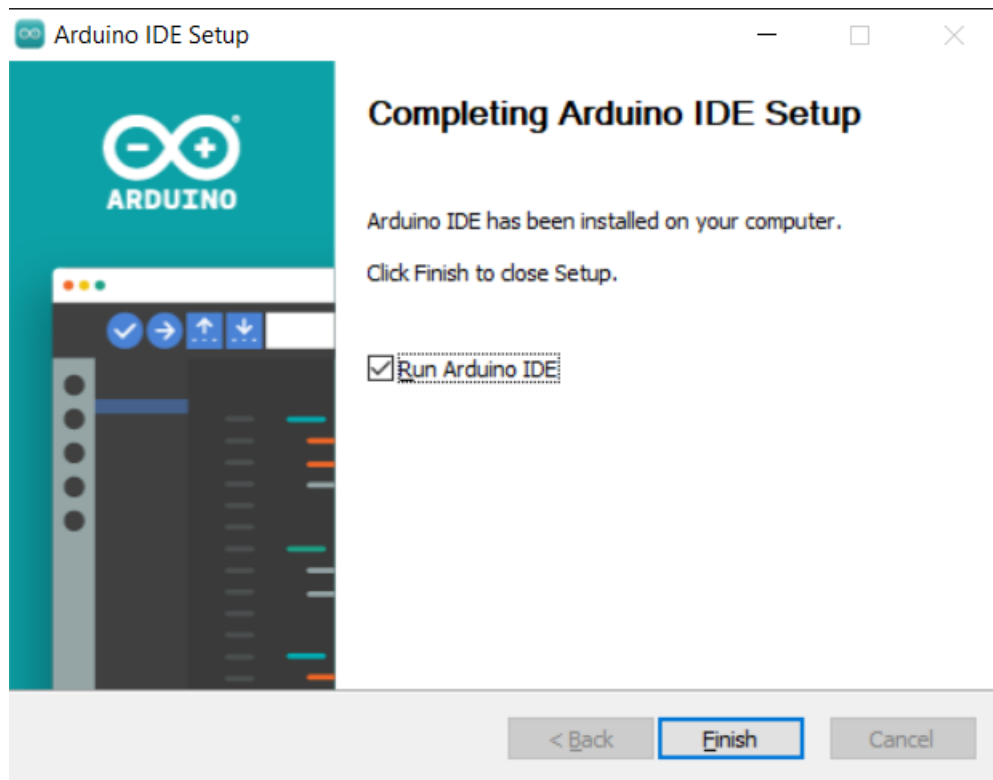
3. インストールオプションを選択します。



4. インストール場所を選択します。ソフトウェアはシステムドライブ以外のドライブにインストールすることを推奨します。

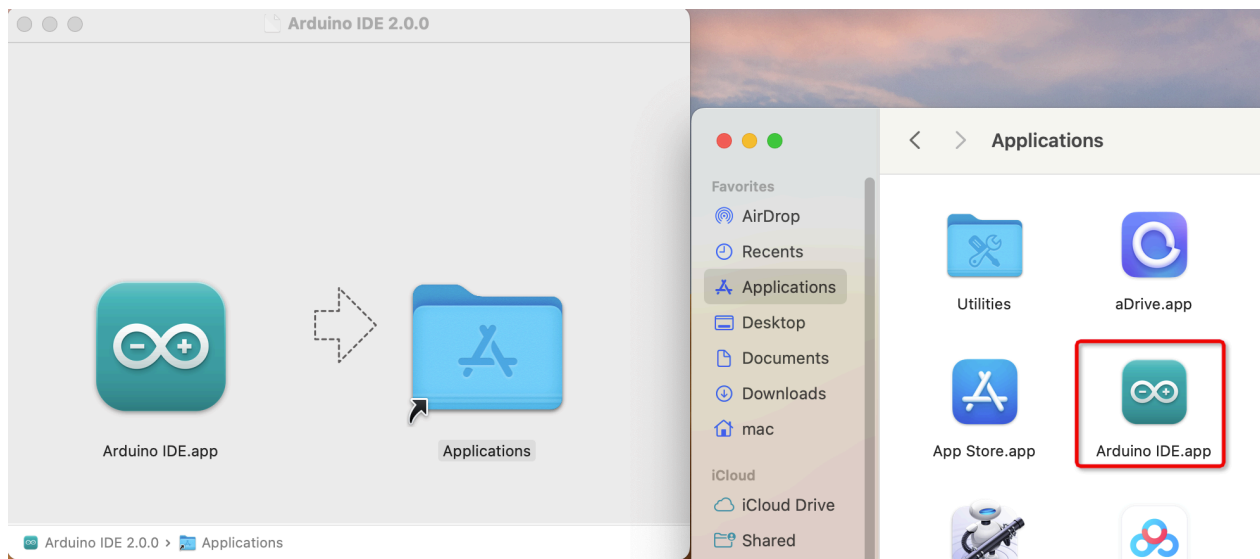


5. 完了をクリックします。



macOS

ダウンロードした `arduino_ide_xxxx.dmg` ファイルをダブルクリックし、指示に従って **Arduino IDE.app** を **Applications** フォルダにコピーします。数秒後、Arduino IDE が正常にインストールされたことが確認できます。

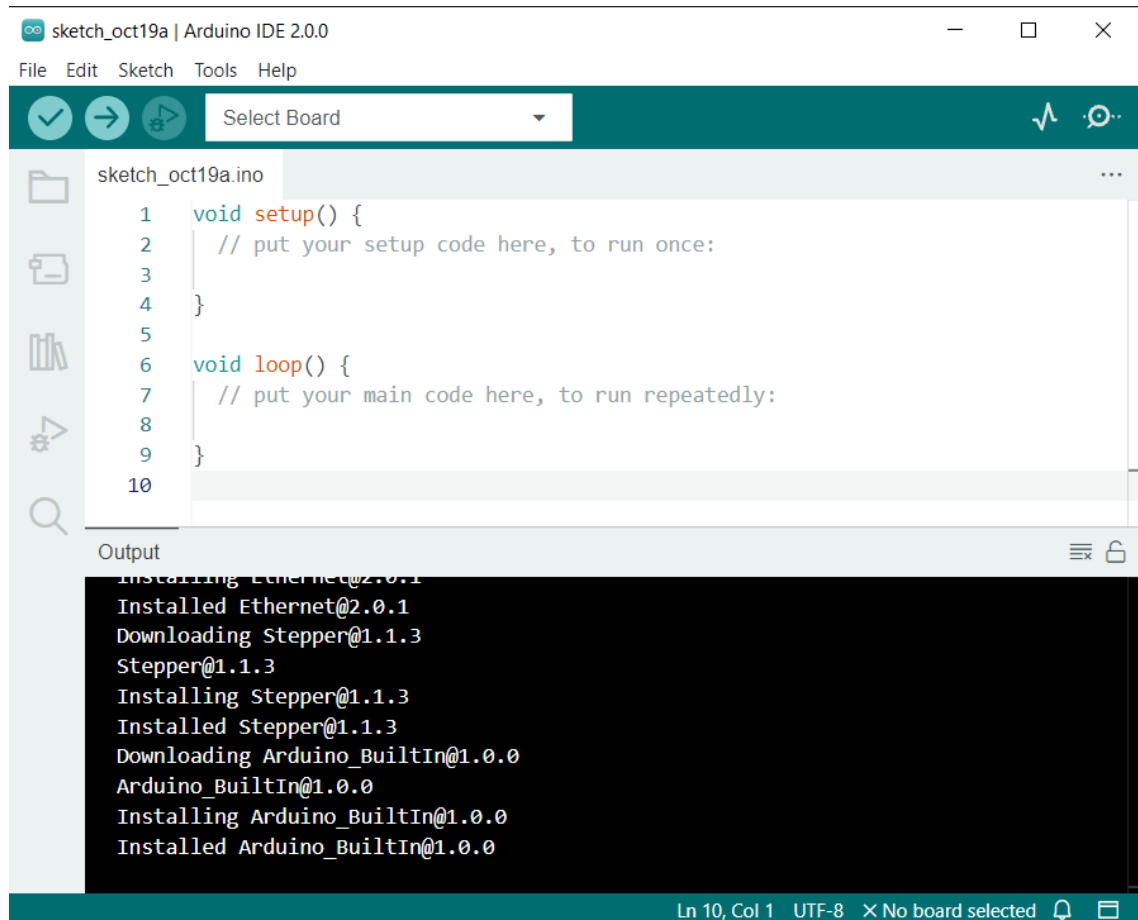


Linux

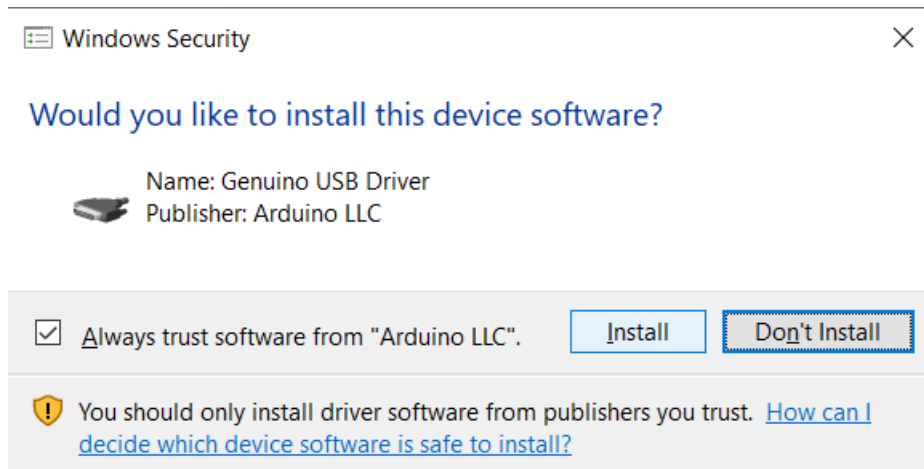
Linux システムでの Arduino IDE 2.0 のインストールチュートリアルについては、以下のリンクを参照してください：<https://docs.arduino.cc/software/ide-v2/tutorials/getting-started/ide-v2-downloading-and-installing#linux>

IDE の起動

1. Arduino IDE 2.0 を初めて開くと、Arduino AVR ボード、ビルトインライブラリ、その他の必要なファイルが自動的にインストールされます。



2. また、ファイアウォールやセキュリティセンターがデバイスドライバのインストールを求めるポップアップを表示することがあります。すべてのドライバをインストールしてください。



3. これで、Arduino IDE の準備が完了しました！

注釈： ネットワークの問題やその他の理由で一部のインストールが完了しなかった場合、Arduino IDE を再度開いて残りのインストールを完了することができます。すべてのインストールが完了した後、Verify または Upload をクリックしない限り、Output ウィンドウは自動的に開きません。

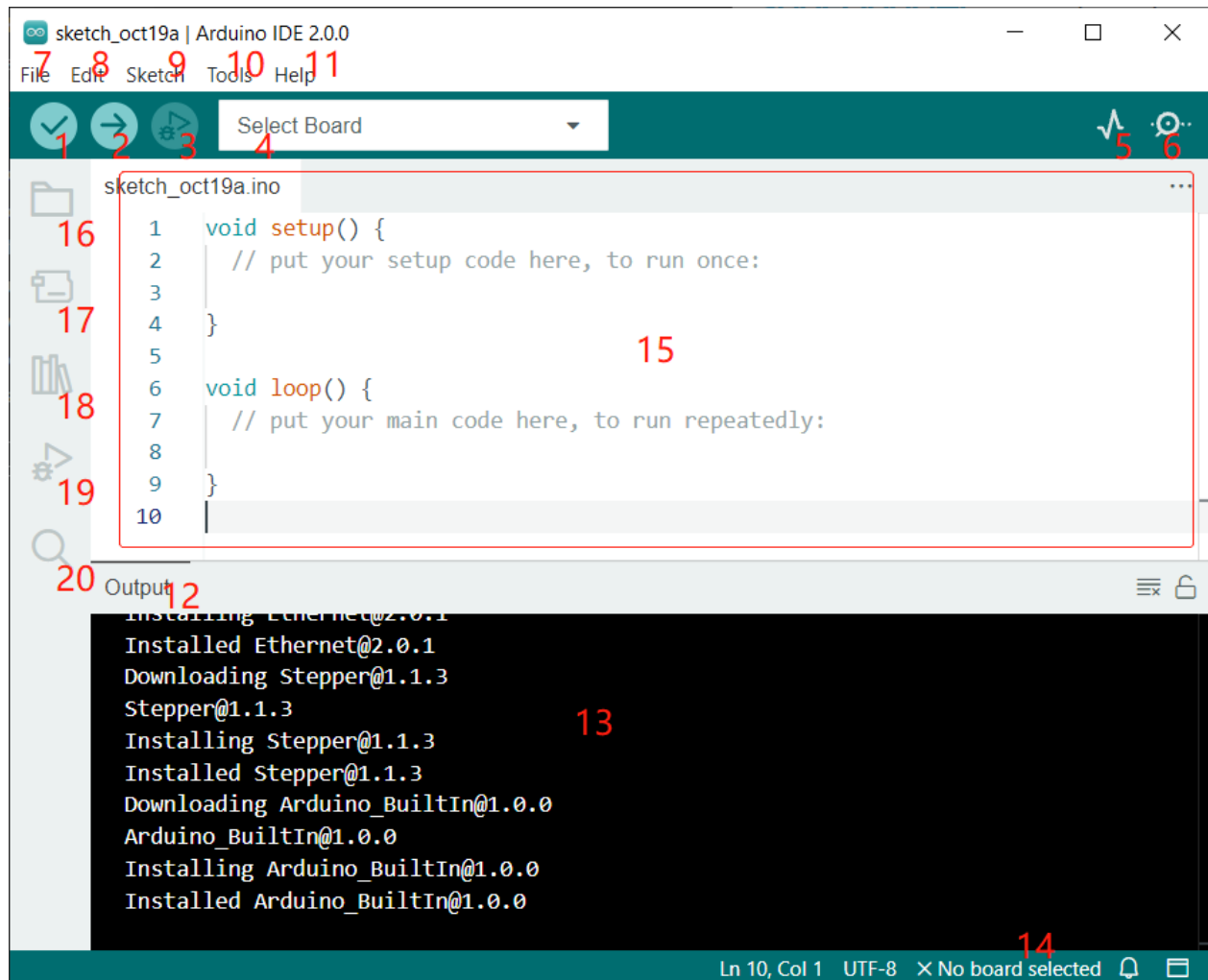
注釈： こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [] をクリックして今すぐ参加しましょう！

2.3.2 Arduino IDE の紹介



1. 検証 (Verify) : コードをコンパイルします。構文の問題があれば、エラーとして提示されます。
2. アップロード (Upload) : ボードにコードをアップロードします。このボタンをクリックすると、ボードの RX と TX LED が高速で点滅し、アップロードが完了するまで点滅が止まりません。
3. デバッグ (Debug) : 行ごとのエラーチェック用。
4. ボードの選択 (Select Board) : 素早くボードとポートを設定します。
5. シリアルプロッタ (Serial Plotter) : 読取値の変化を確認します。
6. シリアルモニター (Serial Monitor) : このボタンをクリックすると、ウィンドウが表示されます。制御ボードから送信されたデータを受信します。デバッグに非常に役立ちます。
7. ファイル (File) : メニューをクリックすると、ドロップダウンリストが表示されます。ファイルの作成、開く、保存、閉じる、一部のパラメータ設定などが含まれます。

8. 編集 (**Edit**): メニューをクリックします。ドロップダウンリストには、切り取り (**Cut**), コピー (**Copy**), 貼り付け (**Paste**), 検索 (**Find**) などの編集操作と、それに対応するショートカットが表示されます。

9. スケッチ (**Sketch**): 検証 (**Verify**), アップロード (**Upload**), ファイルの追加 (**Add**) などの操作が含まれます。より重要な機能はライブラリを含む (**Include Library**) – ライブラリを追加できます。

10. ツール (**Tool**): いくつかのツールが含まれます。最も頻繁に使用されるのはボード (使用するボード) とポート (ボードが接続されているポート) です。コードをアップロードしたいたびに、それらを選択またはチェックする必要があります。

11. ヘルプ (**Help**): 初心者の場合、メニューのオプションをチェックして、必要なヘルプを取得できます。IDE の操作、紹介情報、トラブルシューティング、コードの説明などが含まれます。

12. 出力バー (**Output Bar**): ここで出力タブを切り替えます。

13. 出力ウィンドウ (**Output Window**): 情報を表示します。

14. ボードとポート (**Board and Port**): ここでコードのアップロードのために選択されたボードとポートをプレビューできます。何か誤っていれば、ツール (**Tools**) -> ボード (**Board**) / ポート (**Port**) で再選択できます。

15. IDE の編集エリアです。ここでコードを書くことができます。

16. スケッチブック (**Sketchbook**): スケッチファイルを管理するためのもの。

17. ボードマネージャ (**Board Manager**): ボードドライバを管理するためのもの。

18. ライブラリマネージャ (**Library Manager**): ライブラリファイルを管理するためのもの。

19. デバッグ (**Debug**): コードのデバッグを支援します。

20. 検索 (**Search**): スケッチからコードを検索します。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ! Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

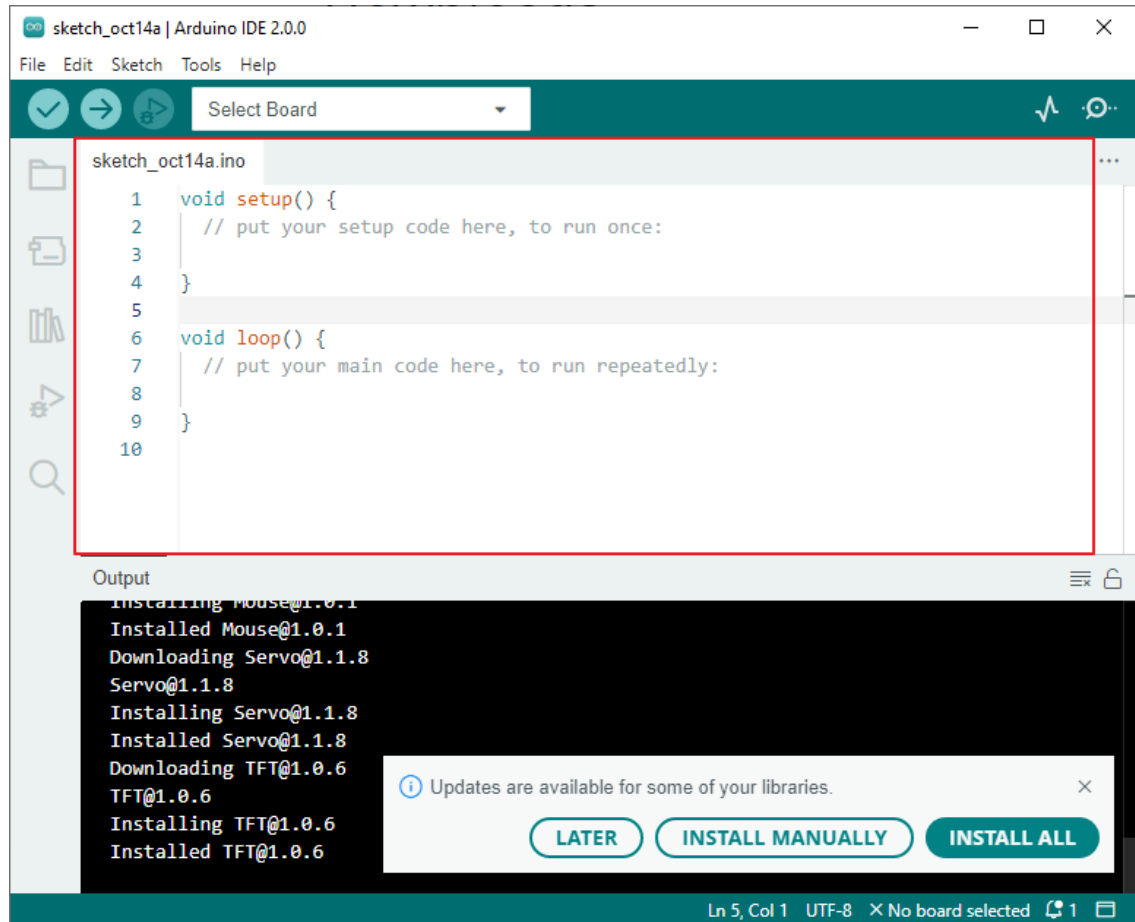
参加する理由は?

- エキスパートサポート: コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有: ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー: 新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引: 最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト: ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか? [] をクリックして今すぐ参加しましょう!

2.3.3 Arduino のスケッチの作成、開く、保存方法は？

1. Arduino IDE を初めて開くか、新しいスケッチを作成すると、このようなページが表示されます。Arduino IDE は新しいファイル、いわゆる"スケッチ"を自動的に作成します。



これらのスケッチファイルには一時的な名前がつけられており、その名前からファイルが作成された日付がわかります。sketch_oct14a.ino は 10 月 14 日の最初のスケッチを意味し、.ino はこのスケッチのファイル形式です。

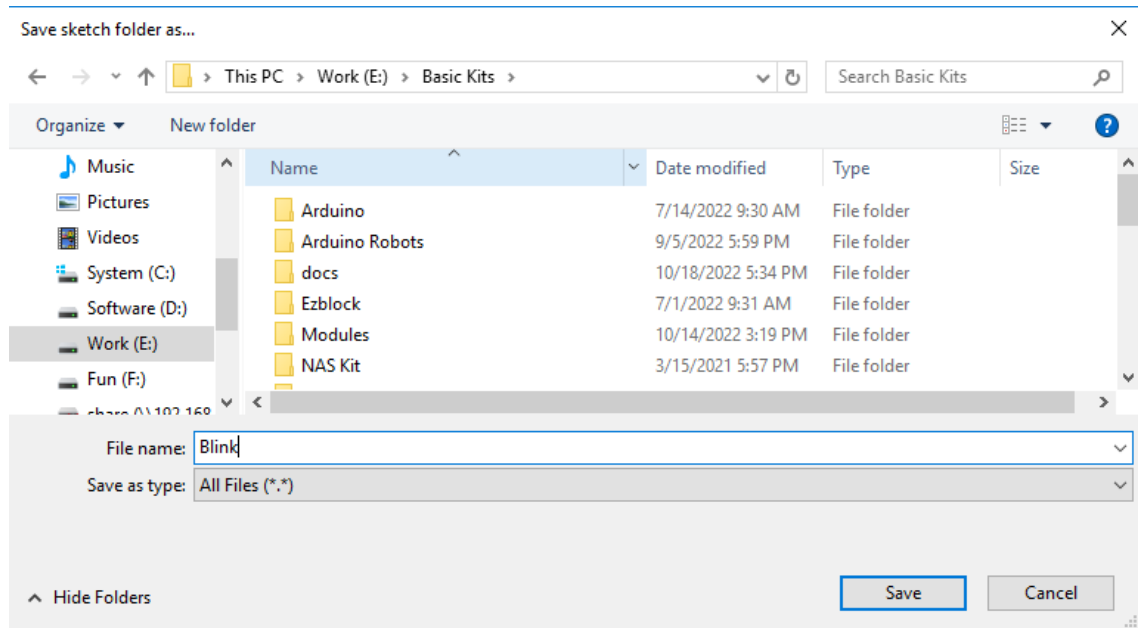
2. 次に新しいスケッチを作成してみましょう。以下のコードを Arduino IDE にコピーして、元のコードを置き換えます。



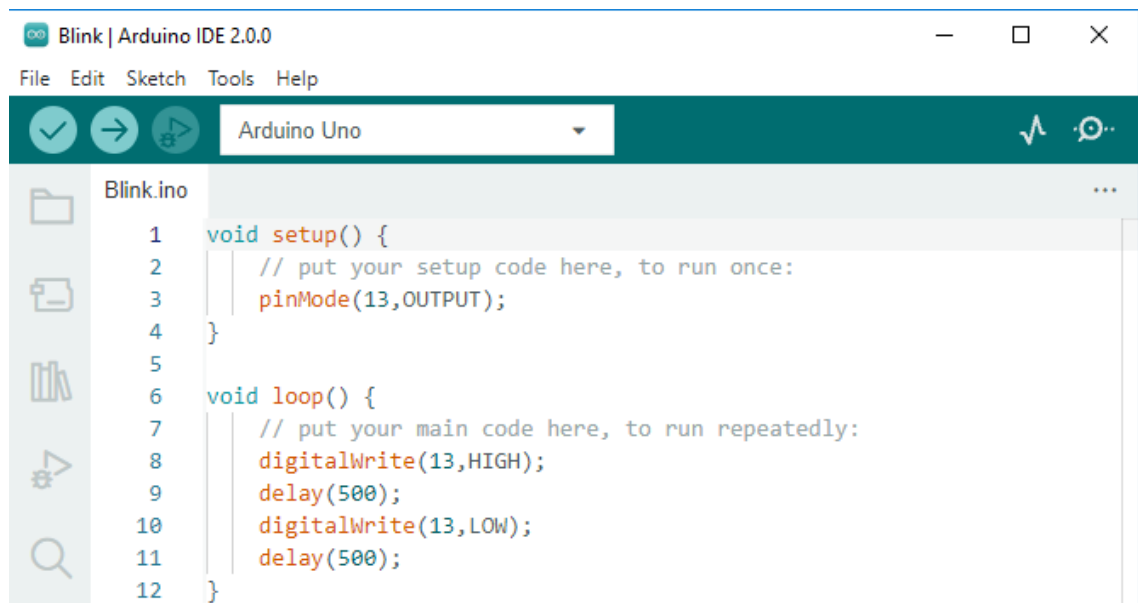
```
void setup() {
    // ここにセットアップコードを書いてください。一度だけ実行されます：
    pinMode(13,OUTPUT);
}

void loop() {
    // ここに主要なコードを書いてください。繰り返し実行されます：
    digitalWrite(13,HIGH);
    delay(500);
    digitalWrite(13,LOW);
    delay(500);
}
```

3. Ctrl+S を押すか、ファイル (File) -> 保存 (Save) をクリックします。デフォルトではスケッチは C:\Users\{your_user}\Documents\Arduino に保存されますが、名前を変更したり、新しいパスに保存することもできます。



4. 保存が成功すると、Arduino IDE 内の名前が更新されたことがわかります。



次のセクションに進み、この作成したスケッチを Arduino ボードにアップロードする方法を学びましょう。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。

- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [] をクリックして今すぐ参加しましょう！

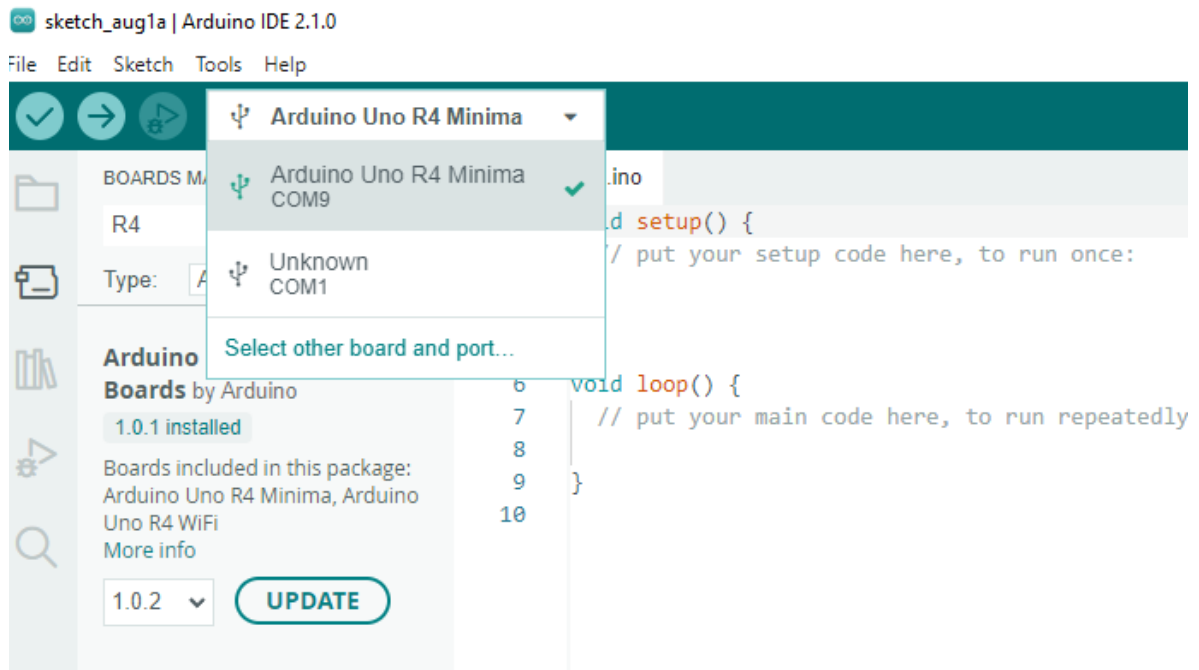
2.3.4 ボードにスケッチをアップロードする方法

このセクションでは、先ほど作成したスケッチを Arduino ボードにアップロードする方法と、考慮すべき点について学びます。

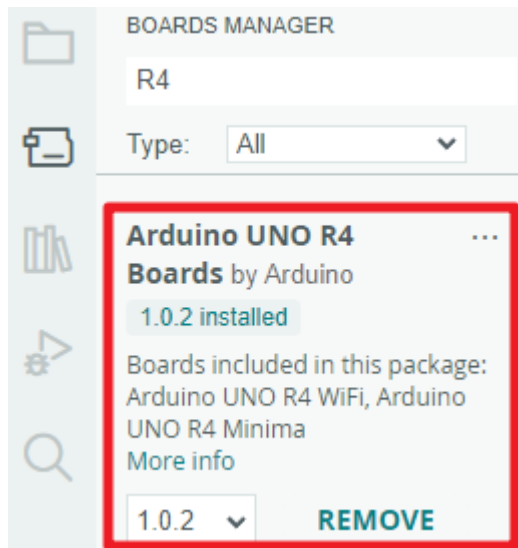
1. ボードとポートを選択

Arduino の開発ボードは通常、USB ケーブルが付属しています。これを使用してボードをコンピュータに接続します。

Arduino IDE で正しい ボード (Board) と ポート (Port) を選択します。通常、Arduino のボードはコンピュータに自動的に認識され、ポートが割り当てられるので、ここで選択できます。



もしボードがすでに接続されているのに認識されない場合、**Boards Manager** の **Arduino UNO R4 Boards** セクションに **INSTALLED** のロゴが表示されているか確認してください。表示されていないければ、少し下にスクロールして **INSTALL** をクリックしてください。



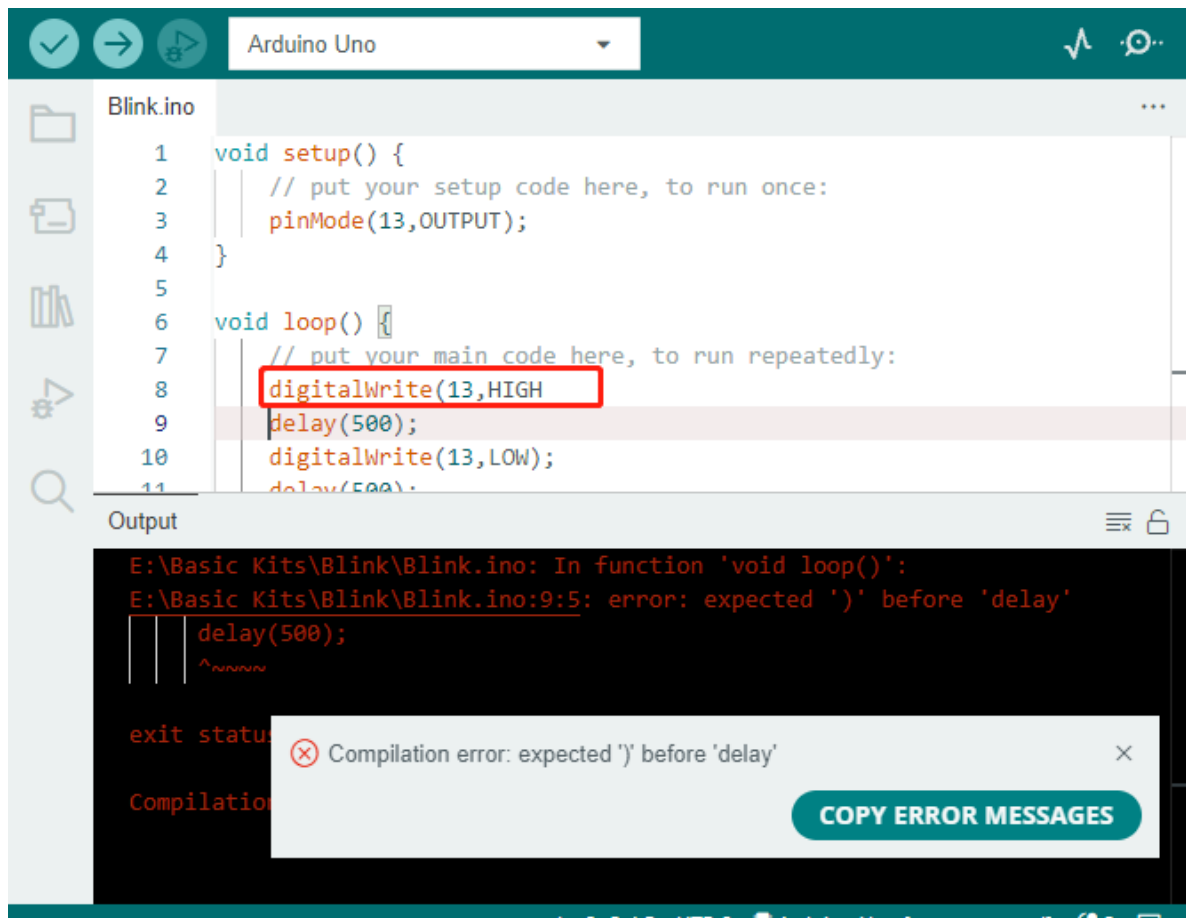
Arduino IDE を再起動し、Arduino のボードを再接続すると、ほとんどの問題が解決します。また、ツール (Tools) -> ボード (Board) や ポート (Port) をクリックして選択することもできます。

2. スケッチの確認

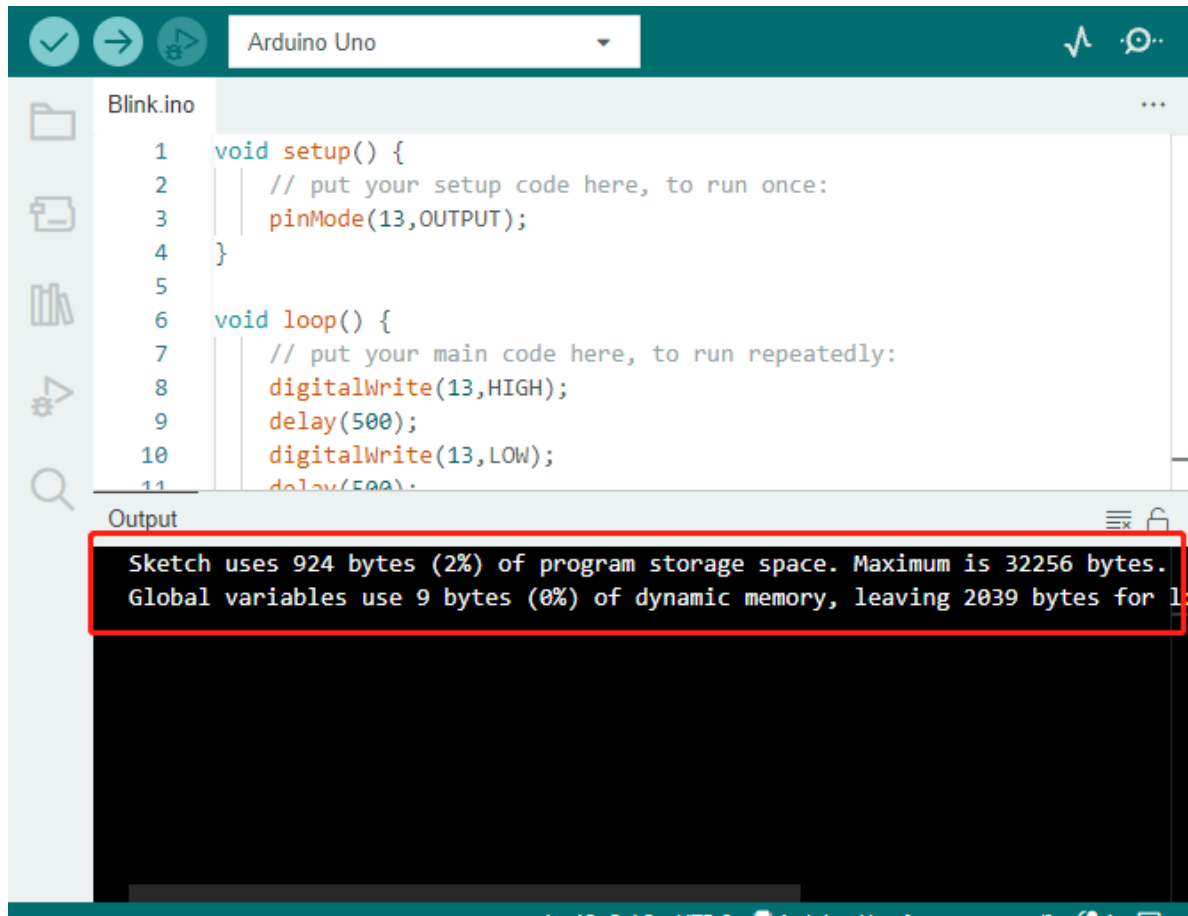
Verify ボタンをクリックすると、スケッチがコンパイルされ、エラーがないか確認されます。



何らかの文字を削除したり、間違って数文字入力したりした場合、この機能を使用してミスを見つけることができます。メッセージバーから、どの場所で何種類のエラーが発生したかを確認できます。

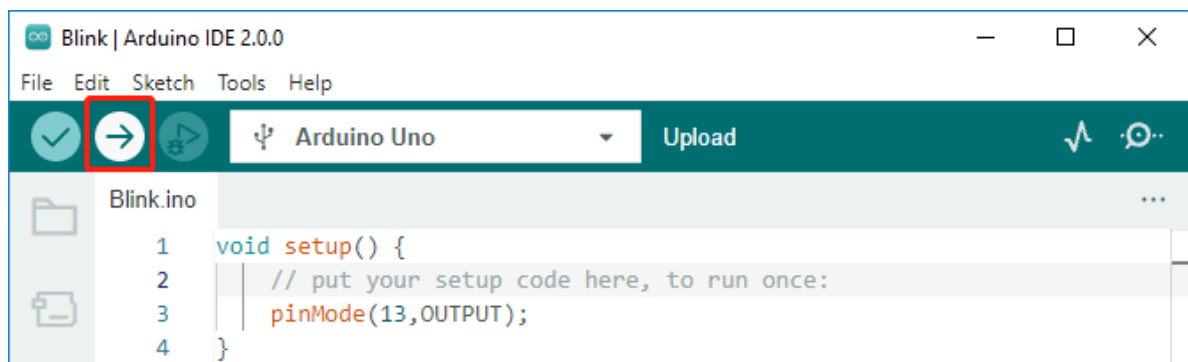


エラーがなければ、以下のようなメッセージが表示されます。



3. スケッチのアップロード

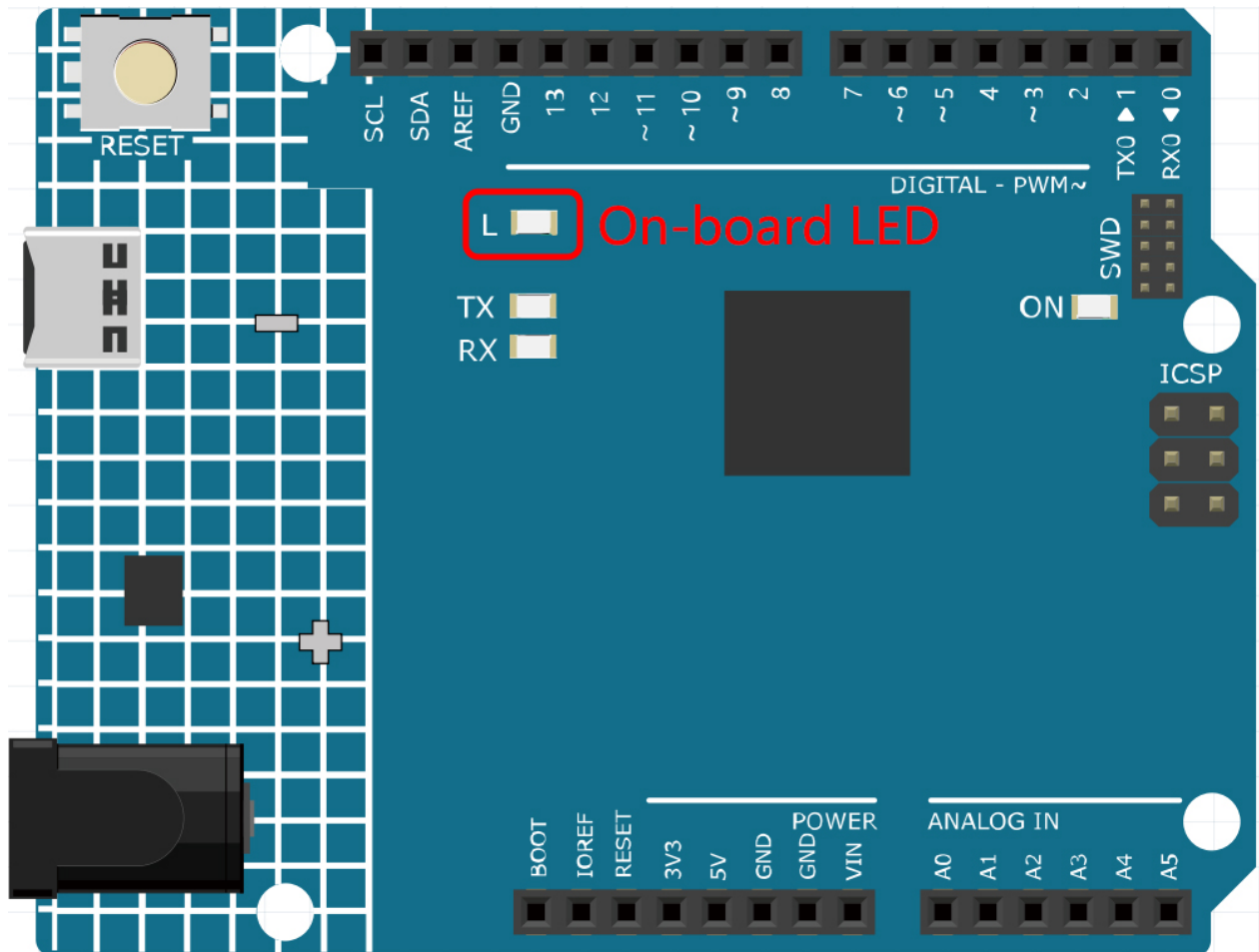
上記のステップを完了した後、アップロード (Upload) ボタンをクリックして、このスケッチをボードにアップロードします。



成功すると、以下のプロンプトが表示されます。



同時に、ボード上の LED が点滅します。



スケッチをアップロードした後、電源を入れると Arduino ボードは自動的にスケッチを実行します。実行中のプログラムは、新しいスケッチをアップロードすることで上書きすることができます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

2.3.5 Arduino のプログラム構造

新しいスケッチファイルを見てみましょう。コード自体は数行しかありませんが、実際には「空の」スケッチです。このスケッチを開発ボードにアップロードしても何も起こりません。

```
void setup() {  
  // ここにセットアップコードを記述します。一度だけ実行されます：  
  
}  
  
void loop() {  
  // ここにメインコードを記述します。何度も実行されます：  
  
}
```

もし `setup()` と `loop()` を削除し、スケッチを本当の blank ファイルにすると、検証に合格しないことがわかります。これらは人間の骨格に相当し、欠かせないものです。

スケッチを書く際、`setup()` は最初に行われ、ボードの電源が入れたりリセットされた後、及び一度だけその中のコード（`{ }` 内）が実行されます。`loop()` はメインの機能を記述するために使用され、`setup()` が実行された後、その中のコードはループで実行されます。

`setup()` と `loop()` の理解を深めるために、以下の 4 つのスケッチを使用します。その目的は、Arduino のオンボード LED を点滅させることです。各実験を順番に実行し、その特定の効果を記録してください。

- スケッチ 1: オンボード LED を連続して点滅させる。

```
void setup() {  
  // ここにセットアップコードを記述します。一度だけ実行されます：  
  pinMode(13,OUTPUT);  
}
```

(次のページに続く)

(前のページからの続き)

```
void loop() {  
    // ここにメインコードを記述します。何度も実行されます :  
    digitalWrite(13,HIGH);  
    delay(500);  
    digitalWrite(13,LOW);  
    delay(500);  
}
```

- スケッチ 2: オンボード LED を 1 回だけ点滅させる。

```
void setup() {  
    // ここにセットアップコードを記述します。一度だけ実行されます :  
    pinMode(13,OUTPUT);  
    digitalWrite(13,HIGH);  
    delay(500);  
    digitalWrite(13,LOW);  
    delay(500);  
}  
  
void loop() {  
    // ここにメインコードを記述します。何度も実行されます :  
}
```

- スケッチ 3: オンボード LED をゆっくり 1 回点滅させた後、速く点滅させる。

```
void setup() {  
    // ここにセットアップコードを記述します。一度だけ実行されます :  
    pinMode(13,OUTPUT);  
    digitalWrite(13,HIGH);  
    delay(1000);  
    digitalWrite(13,LOW);  
    delay(1000);  
}  
  
void loop() {  
    // ここにメインコードを記述します。何度も実行されます :  
    digitalWrite(13,HIGH);  
    delay(200);  
}
```

(次のページに続く)

(前のページからの続き)

```
digitalWrite(13,LOW);  
delay(200);  
}
```

- スケッチ 4: エラーを報告。

```
void setup() {  
    // ここにセットアップコードを記述します。一度だけ実行されます :  
    pinMode(13,OUTPUT);  
}  
  
digitalWrite(13,HIGH);  
delay(1000);  
digitalWrite(13,LOW);  
delay(1000);  
  
void loop() {  
    // ここにメインコードを記述します。何度も実行されます :  
}
```

これらのスケッチを用いて、setup-loop のいくつかの特性をまとめることができます。

- loop() は、ボードに電源が供給された後、繰り返し実行されます。
- setup() は、ボードに電源が供給された後、1 回だけ実行されます。
- ボードに電源が供給されると、まず setup() が実行され、次に loop() が実行されます。
- コードは setup() や loop() の {} のスコープ内に記述する必要がある、フレームワークの外部ではエラーになります。

注釈: digitalWrite(13,HIGH) のようなステートメントは、オンボード LED の制御に使用されます。これらの使用法については、後の章で詳しく説明します。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。

- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

2.3.6 スケッチの書き方のルール

友達に照明をつけてもらう場合、「照明をつけてください」とか「ライトをつけて、兄弟！」など、好きな口調で言うことができます。

しかし、Arduino ボードに何かをしてもらいたい場合、Arduino プログラムの書き方のルールに従ってコマンドを入力する必要があります。

この章では、Arduino 言語の基本的なルールを紹介し、自然言語をコードに翻訳する方法を理解するのに役立ちます。

もちろん、これは慣れるまで時間がかかるプロセスであり、初心者にとっては最もエラーが発生しやすい部分でもあるので、頻繁に間違えることがあっても大丈夫です。何度も試してみてください。

セミコロン；

手紙を書く時、各文の最後にピリオドを書くように、Arduino 言語では、コマンドの終了をボードに伝えるために；を使用する必要があります。

「オンボード LED の点滅」の例を取り上げてみましょう。正常なスケッチは以下のようになります。

例:

```
void setup() {  
    // put your setup code here, to run once:  
    pinMode(13,OUTPUT);  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
    digitalWrite(13,HIGH);  
    delay(500);  
    digitalWrite(13,LOW);  
}
```

(次のページに続く)

(前のページからの続き)

```
    delay(500);  
}
```

次に、以下の 2 つのスケッチを見て、実行する前に Arduino で正しく認識できるかを推測してみましょう。

スケッチ A:

```
void setup() {  
    // put your setup code here, to run once:  
    pinMode(13,OUTPUT);  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
    digitalWrite(13,HIGH)  
    delay(500)  
    digitalWrite(13,LOW)  
    delay(500)  
}
```

スケッチ B:

```
void setup() {  
    // put your setup code here, to run once:  
    pinMode(13,OUTPUT);  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
    digitalWrite(13,  
HIGH); delay  
    (500  
    );  
    digitalWrite(13,  
  
    LOW);  
        delay(500)  
    ;  
}
```

結果としては、スケッチ A はエラーを報告し、スケッチ B は実行されます。

- スケッチ A のエラーは、`;` が欠落していることで、見た目は正常に見えますが、Arduino は読み取ることができません。
- スケッチ B は、人間にとっては反対に見えますが、実際には、インデント、改行、ステートメント内のスペースなど、Arduino のプログラムには存在しないものですので、Arduino のコンパイラにとっては、例と同じに見えます。

しかし、スケッチ B のようにコードを書かないでください。コードを書き、表示するのは通常、人間ですので、自分自身を困らせないでください。

中括弧 `{}`

`{}` は Arduino プログラミング言語の主要な構成要素で、ペアで表示される必要があります。良いプログラムの慣習として、左の中括弧を入力した直後に右の中括弧を入力し、中括弧の間にカーソルを移動してステートメントを挿入するのが良いでしょう。

コメント `//`

コメントはコンパイラが無視するスケッチの部分です。プログラムの動作方法を他者に伝えるために通常使用されます。

コードの行に 2 つの連続したスラッシュを書くと、コンパイラはその行の終わりまでの内容を全て無視します。

新しいスケッチを作成すると、2 つのコメントが付属してきます。これら 2 つのコメントを削除しても、スケッチには何の影響もありません。

```
void setup() {  
    // put your setup code here, to run once:  
  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
  
}
```

プログラミングにおいてコメントは非常に便利で、以下にいくつかの一般的な使用例を示します。

- 使用法 A：このコードのセクションが何をするのかを自分自身や他者に伝える。

```
void setup() {  
    pinMode(13,OUTPUT); //ピン 13 を出力モードに設定、オンボード LED を制御  
}
```

(次のページに続く)

(前のページからの続き)

```
void loop() {
    digitalWrite(13,HIGH); // ピン 13 を HIGH にしてオンボード LED を点灯
    delay(500); // 500ms の状態維持
    digitalWrite(13,LOW); // オンボード LED を消灯
    delay(500); // 500ms の状態維持
}
```

- 使用法 B：一時的にいくつかの文を無効にし（削除せずに）、使用するとき必要があるときにコメントを外す。これはコードのデバッグやプログラムエラーの特定を試みる際に非常に便利です。

```
void setup() {
    pinMode(13,OUTPUT);
    // digitalWrite(13,HIGH);
    // delay(1000);
    // digitalWrite(13,LOW);
    // delay(1000);
}

void loop() {
    digitalWrite(13,HIGH);
    delay(200);
    digitalWrite(13,LOW);
    delay(200);
}
```

注釈：コードを迅速にコメントする、またはコメントを解除するのに、ショートカット Ctrl+/ を使用してください。

コメント **/**/**

// によるコメントと同様です。このタイプのコメントは複数行にわたることができ、コンパイラが /* を読み取ると、*/ に遭遇するまでの続く内容をすべて無視します。

例 1:

```
/* Blink */
```

(次のページに続く)

(前のページからの続き)

```
void setup() {
    pinMode(13,OUTPUT);
}

void loop() {
    /*
    以下のコードはオンボード LED を点滅させます
    delay() 内の数字を変更することで点滅の頻度を変えることができます
    */
    digitalWrite(13,HIGH);
    delay(500);
    digitalWrite(13,LOW);
    delay(500);
}
```

#define

これは役立つ C++ のツールです。

```
#define identifier token-string
```

コンパイラは identifier を読み取ったときに自動的に token-string に置き換えます。これは通常、定数の定義に使用されます。

例として、define を使用するスケッチを以下に示します。これによりコードの可読性が向上します。

```
#define ONBOARD_LED 13
#define DELAY_TIME 500

void setup() {
    pinMode(ONBOARD_LED,OUTPUT);
}

void loop() {
    digitalWrite(ONBOARD_LED,HIGH);
    delay(DELAY_TIME);
    digitalWrite(ONBOARD_LED,LOW);
    delay(DELAY_TIME);
}
```

コンパイラにとって、実際には次のように見えます。

```
void setup() {  
    pinMode(13,OUTPUT);  
}  
  
void loop() {  
    digitalWrite(13,HIGH);  
    delay(500);  
    digitalWrite(13,LOW);  
    delay(500);  
}
```

identifier が置き換えられ、プログラム内には存在しないことがわかります。そのため、使用する際にいくつか注意点があります。

1. token-string は手動でのみ変更でき、プログラム内の算術によって他の値に変換することはできません。
2. ; のような記号の使用を避けてください。例えば、以下のように。

```
#define ONBOARD_LED 13;  
  
void setup() {  
    pinMode(ONBOARD_LED,OUTPUT);  
}  
  
void loop() {  
    digitalWrite(ONBOARD_LED,HIGH);  
}
```

コンパイラはそれを次のように認識します。これがエラーとして報告されるものです。

```
void setup() {  
    pinMode(13;,OUTPUT);  
}  
  
void loop() {  
    digitalWrite(13;,HIGH);  
}
```

注釈: #define の命名規則として、変数との混同を避けるために identifier を大文字にすることが推奨されて

います。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なレビュー：新製品の発表や先行レビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

2.3.7 変数

変数は、プログラムにおける最も強力な重要なツールの一つです。プログラム内でデータを保存し、呼び出すのに役立ちます。

以下のスケッチファイルは変数を使用しています。オンボード LED のピン番号を変数 `ledPin` に、数字"500"を変数 `delayTime` に保存しています。

```
int ledPin = 13;
int delayTime = 500;

void setup() {
    pinMode(ledPin,OUTPUT);
}

void loop() {
    digitalWrite(ledPin,HIGH);
    delay(delayTime);
    digitalWrite(ledPin,LOW);
    delay(delayTime);
}
```

ちょっと待って、これは `#define` が行うことの複製ですか？答えは NO です。

- `#define` の役割はテキストを単純かつ直接に置き換えることであり、プログラムの一部としてコンパイラには認識されません。
- 一方、`variable` はプログラム内に存在し、値を保存し呼び出すために使用されます。変数はプログラム内でその値を変更することもできますが、`define` ではできません。

以下のスケッチファイルは、変数に自己追加を行い、各点滅の後でオンボード LED の点滅時間が長くなります。

```
int ledPin = 13;
int delayTime = 500;

void setup() {
    pinMode(ledPin,OUTPUT);
}

void loop() {
    digitalWrite(ledPin,HIGH);
    delay(delayTime);
    digitalWrite(ledPin,LOW);
    delay(delayTime);
    delayTime = delayTime+200; //各実行で値が 200 増加
}
```

変数の宣言

変数を宣言するということは、変数を作成することを意味します。

変数を宣言するには、データ型と変数名の 2 つが必要です。データ型は変数との間でスペースで区切られ、変数の宣言は ; で終了する必要があります。

この変数を例に使用してみましょう。

```
int delayTime;
```

データ型

ここでの `int` は、整数型と呼ばれるデータ型で、-32768 から 32766 までの整数を格納するために使用できます。小数点を格納することはできません。

変数は整数以外のさまざまなデータも格納できます。Arduino 言語 (C++ であることを忘れないでください) は、その中で最も頻繁に使用され、有用なもののいくつか (以下にリストされているもののみ) をサポートしています:

- `float`: 小数点を含む数値、例えば 3.1415926。
- `byte`: 0 から 255 までの数字を保持。

- `boolean` : `True` または `False` の 2 つの可能な値のみを保持し、メモリ内で 1 バイトを占有。
- `char` : -127 から 127 までの数字を保持。 `char` としてマークされているため、コンパイラはそれを の文字と一致させようとします。
- `string` : 文字列を格納できます、例えば `Halloween`。

変数名

変数名は `i`、`apple`、`Bruce`、`R2D2`、`Sectumsempra` など、任意の名前に設定できますが、基本的なルールに従う必要があります。

1. それが何に使われるのかを説明します。ここでは、変数の名前を `delayTime` としていますので、それが何をするのが簡単に分かります。変数の名前を `barryAllen` とするのも良いですが、コードを見ている人を混乱させるでしょう。
2. 一般的な命名規則を使用します。私のように `CamelCase` を使用することもできますし、変数が二つの単語で構成されていることが分かりやすいように、`delayTime` の `T` のように最初の文字を大文字にすることもできます。また、`UnderScoreCase` を使用して変数を `delay_time` として書くこともできます。これはプログラムの実行に影響しませんが、好みの命名規則を使用すると、プログラマがコードを読むのが楽になります。
3. キーワードを使用しない。"`int`"を入力すると、Arduino IDE はそれを色付けして、それが特別な目的を持つ単語であり、変数名として使用できないことをリマインドします。変数の名前が色付けされている場合は、名前を変更してください。
4. 特別なシンボルは許可されていません。例えば、スペース、`#`、`$`、`/`、`+`、`%` などです。英字（大文字/小文字を区別）、アンダースコア、数字の組み合わせ（ただし、数字は変数名の最初の文字として使用できません）は十分にリッチです。

変数に値を代入する

変数を宣言したら、データを保存する時がきました。代入演算子（すなわち `=`）を使用して、変数に値を入れます。

変数を宣言すると同時にその値を代入することができます。

```
int delayTime = 500;
```

それをある時点で新しい値に代入することも可能です。

```
int delayTime; // 値なし
delayTime = 500; // 値は 500
delayTime = delayTime + 200; // 値は 700
```

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

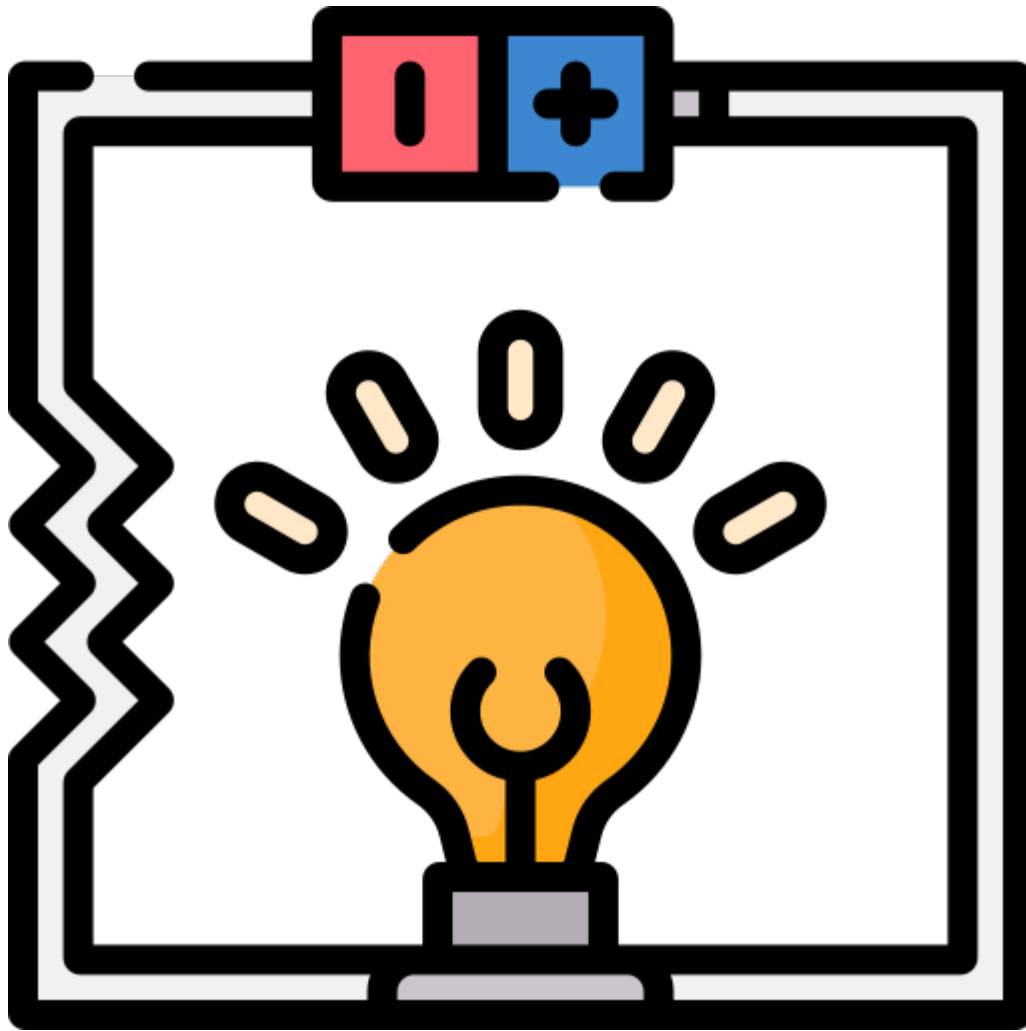
- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

2.3.8 回路の作り方

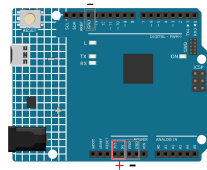
私たちが日常で使用する多くの物は電気で動いています。例えば、家の照明やこの文章を読んでいるコンピューターなどです。

電気を利用するためには、電気回路を作成する必要があります。基本的に、回路とは電気が流れる道のことであり、電子回路とも呼ばれます。回路は特定の方法で接続された電気デバイスやコンポーネント（機器）で構成されています。例えば、抵抗器、コンデンサー、電源、スイッチなどです。



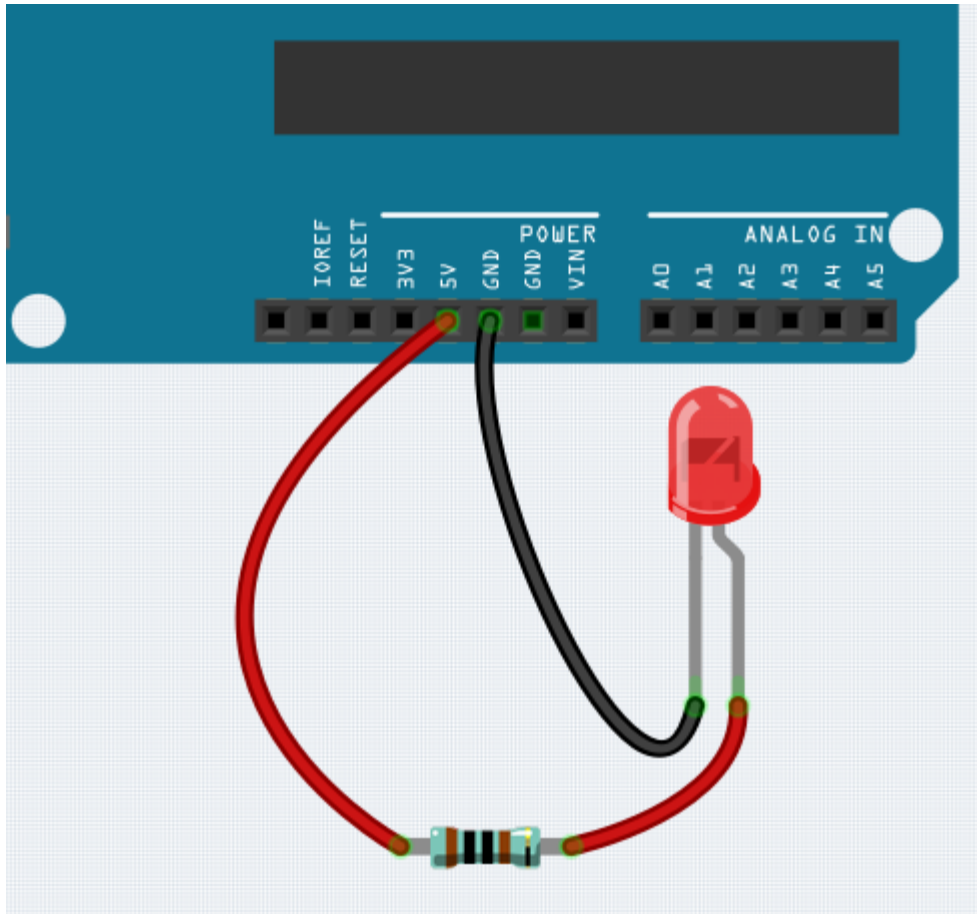
回路は電流を作成するために電子が移動する閉じた経路です。電流を流すためには、電源の正極と負極との間に導電経路が存在しなければならず、これを閉回路と言います（途切れている場合は、開回路と呼ばれます）。

Arduino ボードには電源出力ピン（正極）と接地ピン（負極）があります。これらのピンを使用して、電源の正負側としてボードに電源を接続することができます。



電気を使えば、光や音、動きのある作品を作成することができます。LED の長いピンを正極に、短いピンを負極に接続することで LED を点灯させることができます。この状態で使用すると、LED はすぐに壊れてしまうので、回路内に 1K の抵抗器を追加して保護する必要があります。

以下にその回路の形状を示します。



この時点で疑問が湧くかもしれません。この回路をどのように組み立てればよいのでしょうか。ワイヤーを手で持っているだけで、ピンとワイヤーをテープで固定するのでしょうか。

このような場合、はんだを使わないブレッドボードが大変役立ちます。

ブレッドボード、こんにちは！

ブレッドボードは小さな穴がたくさん開いている長方形のプラスチック製の板です。これらの穴には、電子部品を簡単に挿入して、電子回路を組むことができます。ブレッドボードは電子部品を恒久的に固定しないので、何か問題が発生した場合、簡単に修理や再開ができます。

注釈：ブレッドボードを使用する際に特別なツールは必要ありません。ただし、多くの電子部品は非常に小さいので、ピンセットを使用して小さな部品をより効果的に取り扱うことができます。

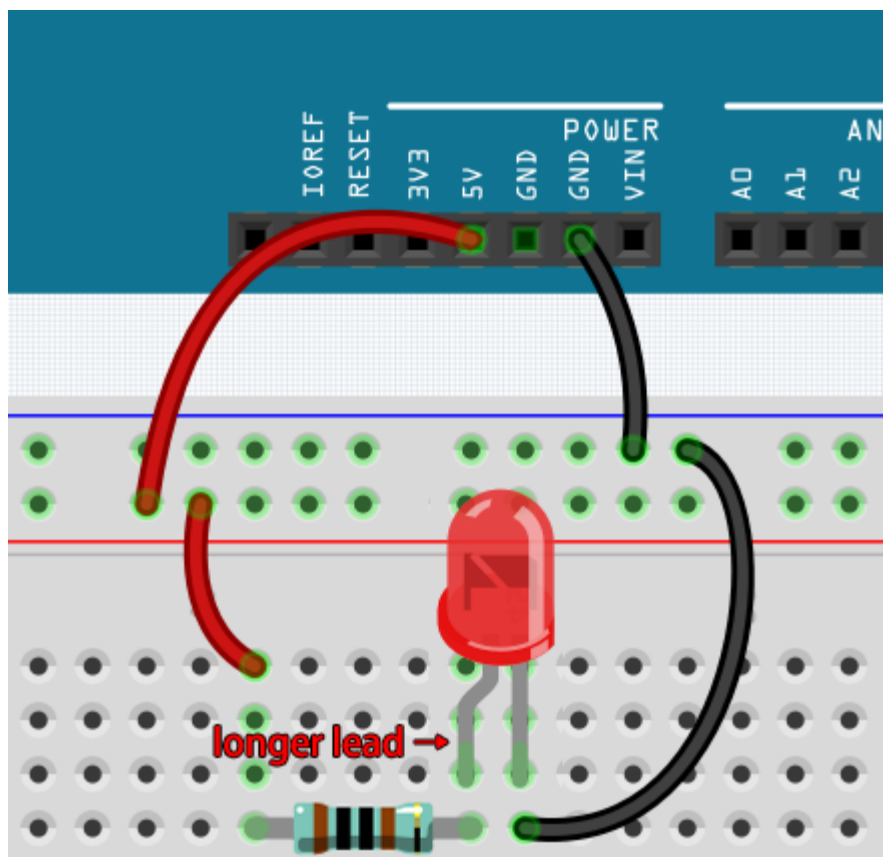
インターネット上では、ブレッドボードに関する多くの情報が見つかります。

- [ブレッドボードの使い方 - Science Buddies](#)
- [ブレッドボードとは何ですか? - Makezine](#)

ブレッドボードについて知っておくべきことは以下の通りです。

1. 各ハーフロウグループ（例：1 行目の列 A-E や 3 行目の列 F-J）は接続されています。そのため、A1 から電気信号が流れ込むと、B1, C1, D1, E1 から流れ出すことができますが、F1 や A2 からは流れ出すことはできません。
2. ほとんどの場合、ブレッドボードの両側は電源バスとして使用され、各列の穴（約 50 の穴）は互いに接続されています。一般的に、正の電源は赤いワイヤーの近くの穴に、負の電源は青いワイヤーの近くの穴に接続されます。
3. 回路内では、負極から正極へと電流が流れます。この場合、ショート回路が発生する可能性があります。

電流の流れる方向に従って、回路を組み立ててみましょう！



1. この回路では、ボードの 5V ピンを使用して LED に電力を供給します。M2M ジャンプワイヤーを使用して、それを赤い電源バスに接続します。
2. LED を保護するためには、1K オームの抵抗を経由して電流を流す必要があります。抵抗の一方の端子（どちらの端子でも良い）を赤い電源バスに、もう一方の端子をブレッドボードの空いている行に接続します。

注釈： 1k 抵抗器のカラーコードは茶、黒、黒、茶、茶です。

3. LED を取り上げると、片方のリードが他方よりも長いことがわかります。抵抗と同じ行に長いリードを接続し、短いリードを別の行に接続します。

注釈: 長いリードはアノード（正極）を示し、短いリードはカソード（負極）を示します。

アノードは抵抗を経由して GPIO ピンに接続する必要があり、カソードは GND ピンに接続する必要があります。

4. M2M ジャンパワイヤーを使用して、LED の短いピンをブレッドボードの負の電源バスに接続します。
5. ジャンパを使用して、ボードの GND ピンを負の電源バスに接続します。

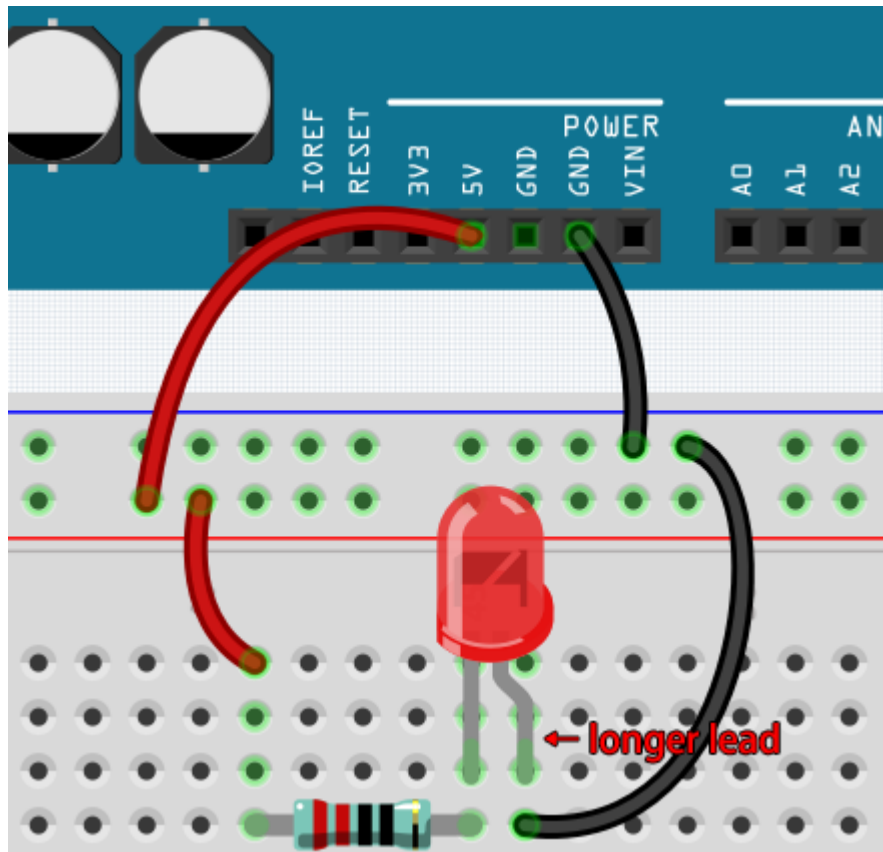
ショート回路に注意

ショート回路は、接続されるべきでない 2 つのコンポーネントが「偶然」接続されたときに発生します。このキットには、抵抗器、トランジスタ、コンデンサ、LED など、お互いに接触してショートを引き起こす可能性のある長い金属のピンを持つ部品が含まれています。ショートが発生すると、一部の回路は正常に動作しなくなるだけでなく、特に電源と接地バスの間で、回路が非常に高温になり、ブレッドボードのプラスチックが溶け、部品が焼損することもあります。

そのため、ブレッドボード上のすべての電子部品のピンが互いに接触していないことを常に確認してください。

回路の方向性

回路には方向性があります。この方向性は、特定の電子部品において非常に重要な役割を果たします。正極と負極に基づいて正しく接続する必要がある極性を持つデバイスがあります。方向性を間違えて構築された回路は正常に動作しません。



以前作成したこのシンプルな回路で LED を逆にすると、LED が動作しなくなることがわかります。

逆に、この回路の抵抗器のように方向性を持たないデバイスもありますので、それらを逆にしても LED の正常な動作に影響はありません。

"+"、"-","GND"、"VCC"などのラベルが付いている、または異なる長さのピンを持つほとんどの部品やモジュールは、特定の方法で回路に接続する必要があります。

回路の保護

電流は、完全な電気回路のある点を過ぎる電子の流れの速さです。最も基本的には、電流 = 流れです。アンペアは、電流を測定するための国際的な単位です。これは、特定の時間内に回路のある点を過ぎる電子（「電気充電」とも呼ばれる）の数量を表します。

電流の流れの背後にある原動力は電圧と呼ばれ、ボルト（V）で測定されます。

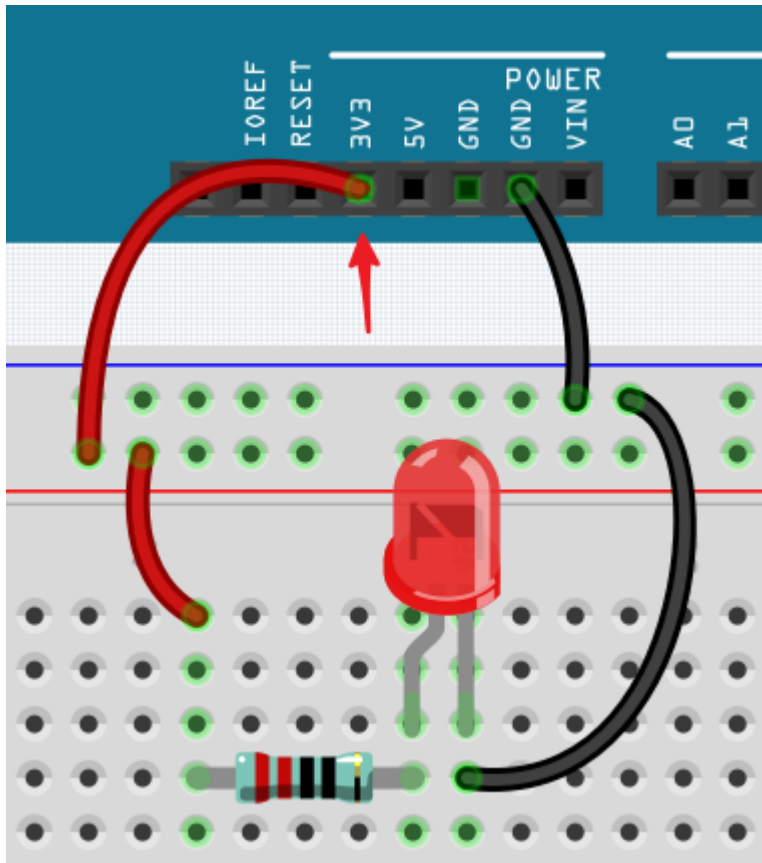
抵抗（R）は、電流の流れを制限する材料の性質であり、オーム（ Ω ）で測定されます。

オームの法則によれば（温度が一定の場合）、電流、電圧、および抵抗は比例します。回路の電流はその電圧に比例し、その抵抗に反比例します。

したがって、電流（I）= 電圧（V）/ 抵抗（R）です。

- オームの法則 - Wikipedia

オームの法則については、簡単な実験を行うことができます。



5V から 3.3V に接続するワイヤーを変更すると、LED の光が暗くなります。抵抗を 1kohm から 2kohm に変更すると、LED が以前よりも暗くなることに気付くでしょう。抵抗が大きいほど、LED は暗くなります。

注釈：抵抗についての紹介や抵抗値の計算方法については、[抵抗器](#) を参照してください。

ほとんどのパッケージ化されたモジュールは、超音波モジュールのように適切な電圧（通常 3.3V または 5V）へのアクセスのみが必要です。

ただし、自作の回路では、電気デバイスの供給電圧と抵抗の使用に注意する必要があります。

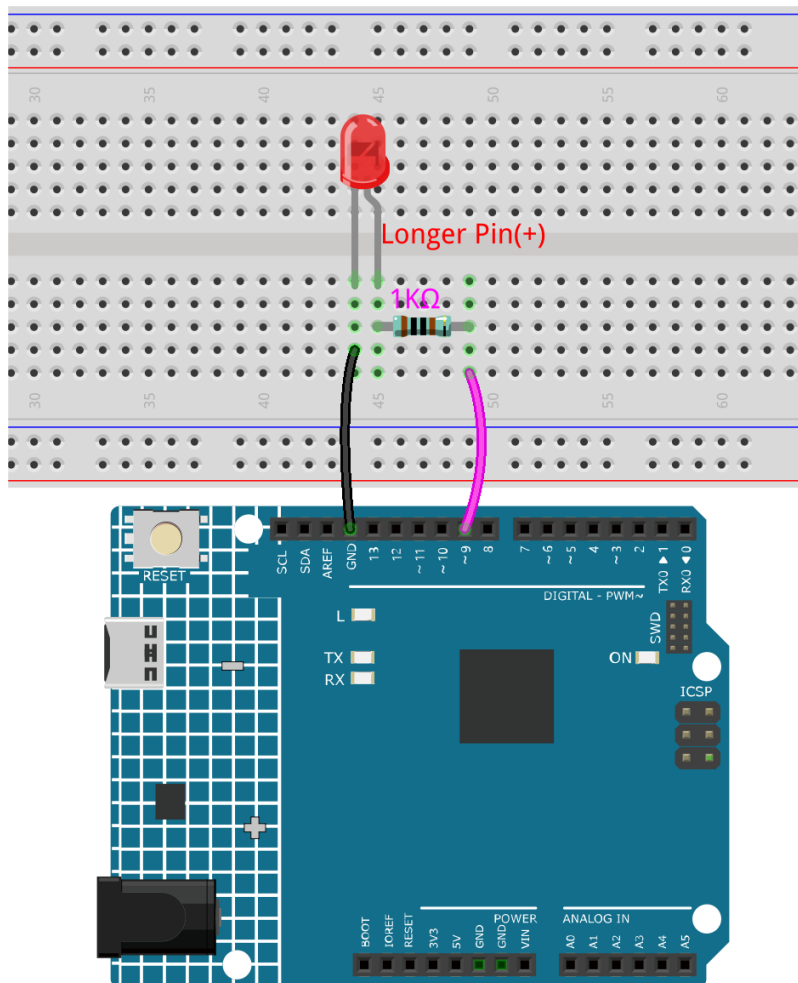
例として、LED は通常、20mA の電流を消費し、その電圧降下は約 1.8V です。オームの法則に従えば、5V の電源を使用する場合、LED を焼き切らないように、最低 160ohm の抵抗（ $(5-1.8)/20\text{mA}$ ）を接続する必要があります。

Arduino での回路制御

Arduino のプログラミングと電子回路の基本的な理解が得られたので、最も重要な質問に取り組む時が来ました：Arduino で回路をどのように制御するか。

簡単に言えば、Arduino が回路を制御する方法は、ボード上のピンのレベルを変更することです。例えば、オンボード LED を制御するとき、それはピン 13 に高いまたは低いレベルの信号を書き込むことです。

さて、ブレッドボード上の点滅する LED を制御するための Arduino ボードをコード化してみましょう。LED がピン 9 に接続されているように回路を構築します。



次に、このスケッチを Arduino 開発ボードにアップロードします。

```
int ledPin = 9;
int delayTime = 500;

void setup() {
  pinMode(ledPin, OUTPUT);
}
```

(次のページに続く)

(前のページからの続き)

```
}  
  
void loop() {  
    digitalWrite(ledPin,HIGH);  
    delay(delayTime);  
    digitalWrite(ledPin,LOW);  
    delay(delayTime);  
}
```

このスケッチは、オンボード LED の点滅を制御するために使用したものと非常に似ていますが、ledPin の値が 9 に変更されています。これは、今回はピン 9 のレベルを制御しようとしているためです。

これで、ブレッドボード上の LED が点滅しているのが見えるでしょう。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

第 3 章

コードをダウンロード

以下のリンクから関連するコードをダウンロードしてください。

- [SunFounder 3 in 1 Kit for Arduino](#)
- また、[SunFounder 3 in 1 Kit for Arduino - GitHub](#) でコードを確認できます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

第 4 章

基本プロジェクト

この章では、Arduino を使用して電子回路を制御する方法を学びます。

コンポーネントに応じて、Arduino の基本的な制御方法は 4 つのタイプに分けられます：

- **1. デジタルライト**: ピンの出力電圧を高または低に設定し、これを使用して光をオンまたはオフにすることができます。
- **2. アナログライト**: アナログ値 (PWM 波) をピンに書き込み、これを使用して光の明るさを調整することができます。
- **3. デジタルリード**: デジタルピンのレベル信号を読み取り、これを使用してスイッチの動作状態を読み取ることができます。
- **4. アナログ読み取り**: アナログピンの電圧を読み取り、これを使用してノブの動作状態を読み取ることができます。

また、使用するには追加のライブラリが必要なコンポーネントもあり、これらはセクション **5.11 外部ライブラリのインストール** の下にまとめられています。

最後に、キットにはいくつかの **6. 面白いプロジェクト** も提供されており、これには多くのシンプルで有用な操作が含まれています。このセクションのコードを試してみれば、ほとんどのシンプルなプロジェクトがどのように動作するかを理解することができます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。

- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.1 1. デジタルライト

デジタルライト は、デジタル信号をデジタルピンに出力または書き込むことです。デジタル信号には 2 つの状態しかなく、0 または 1、0V または 5V であるため、LED やブザーのようないくつかのコンポーネントをオンまたはオフにすることができます。

Arduino R4 ボードには、0 から 13 までの 14 のデジタル I/O ピンがあります。これらのデジタルピンに高または低レベルを書き込むために `pinMode()` と `digitalWrite()` 関数を使用します。

- `pinMode(pin, mode)`: 特定のピンを INPUT または OUTPUT として設定します。ここでは OUTPUT として設定する必要があります。

構文

```
pinMode(pin, mode)
```

パラメータ

- `pin`: モードを設定する Arduino のピン番号。
 - `mode`: INPUT, OUTPUT, または INPUT_PULLUP。
- `digitalWrite(pin, value)`: デジタルピンに高レベル (5V) または低レベル (0V) を書き込み、コンポーネントの動作状態を変更します。もしピンが `pinMode()` で OUTPUT として設定されていれば、その電圧は対応する値に設定されます: HIGH の場合は 5V (または 3.3V ボードの場合は 3.3V)、LOW の場合は 0V (グラウンド)。

構文

```
digitalWrite(pin, value)
```

パラメータ

- `pin`: Arduino のピン番号。
- `value`: HIGH または LOW。

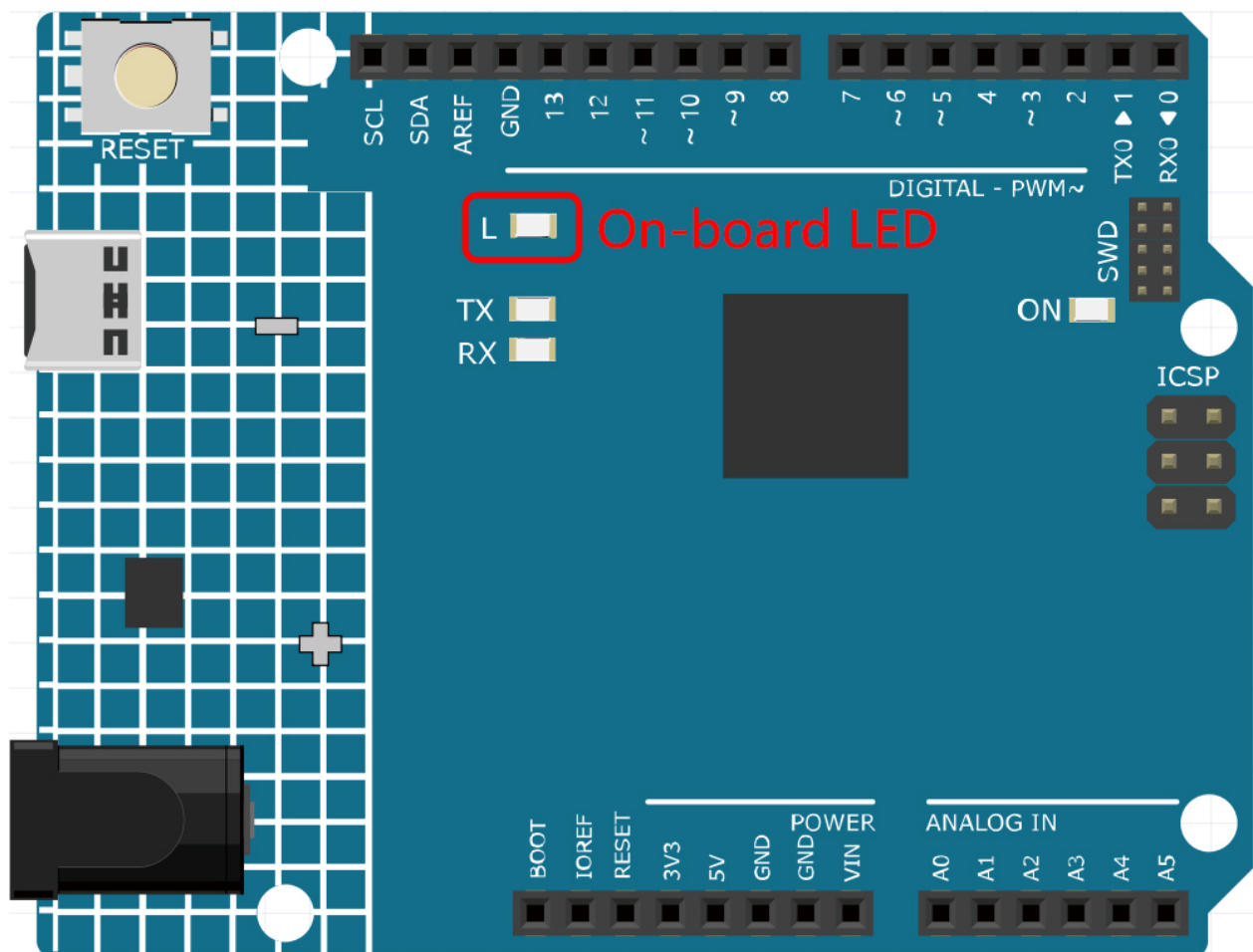
デジタルライトの例：

```
const int pin = 13;
```

(次のページに続く)

(前のページからの続き)

```
void setup() {  
  pinMode(pin, OUTPUT);    // デジタルピンを出力として設定  
}  
  
void loop() {  
  digitalWrite(pin, HIGH); // デジタルピンをオンに設定  
  delay(1000);             // 1秒待つ  
  digitalWrite(pin, LOW);  // デジタルピンをオフに設定  
  delay(1000);             // 1秒待つ  
}
```



注意事項と警告

- ピン 0~13 はすべてデジタルピンです。
- ピン 0 と 1 はコンピュータとの通信に使用されるので使用しないでください。これらのピンに何かを接続すると、通信が妨げられ、ボードのアップロードに失敗する原因となります。

- デジタルピンが使い果たされた場合、アナログピン（A0-A5）もデジタルピンとして使用できます。

関連するコンポーネント

以下は関連するコンポーネントであり、クリックして使用方法を学ぶことができます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.1.1 1.1 こんにちは、LED！

“Hello, world!” を表示することがプログラミング学習の第一歩のように、プログラムを使用して LED を駆動することは、物理的なプログラミング学習の伝統的な導入です。

必要な部品

このプロジェクトでは、以下のコンポーネントが必要です。

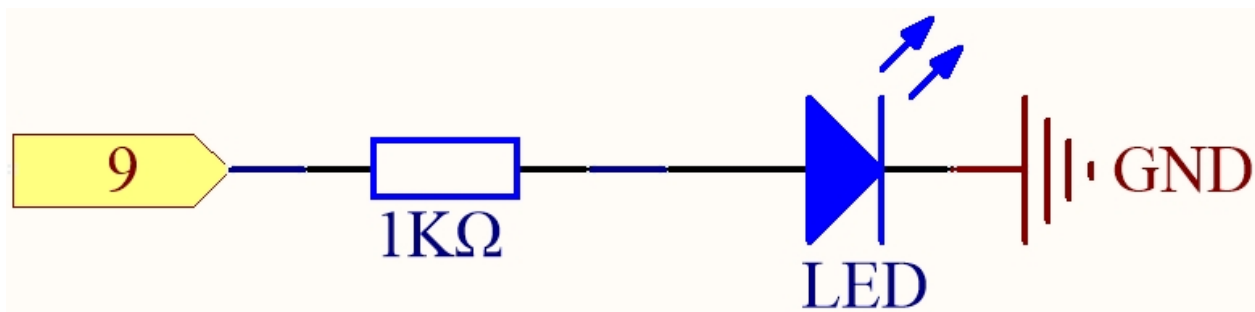
一式を購入するのが便利です、リンクはこちら：

名称	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから個別に購入することもできます。

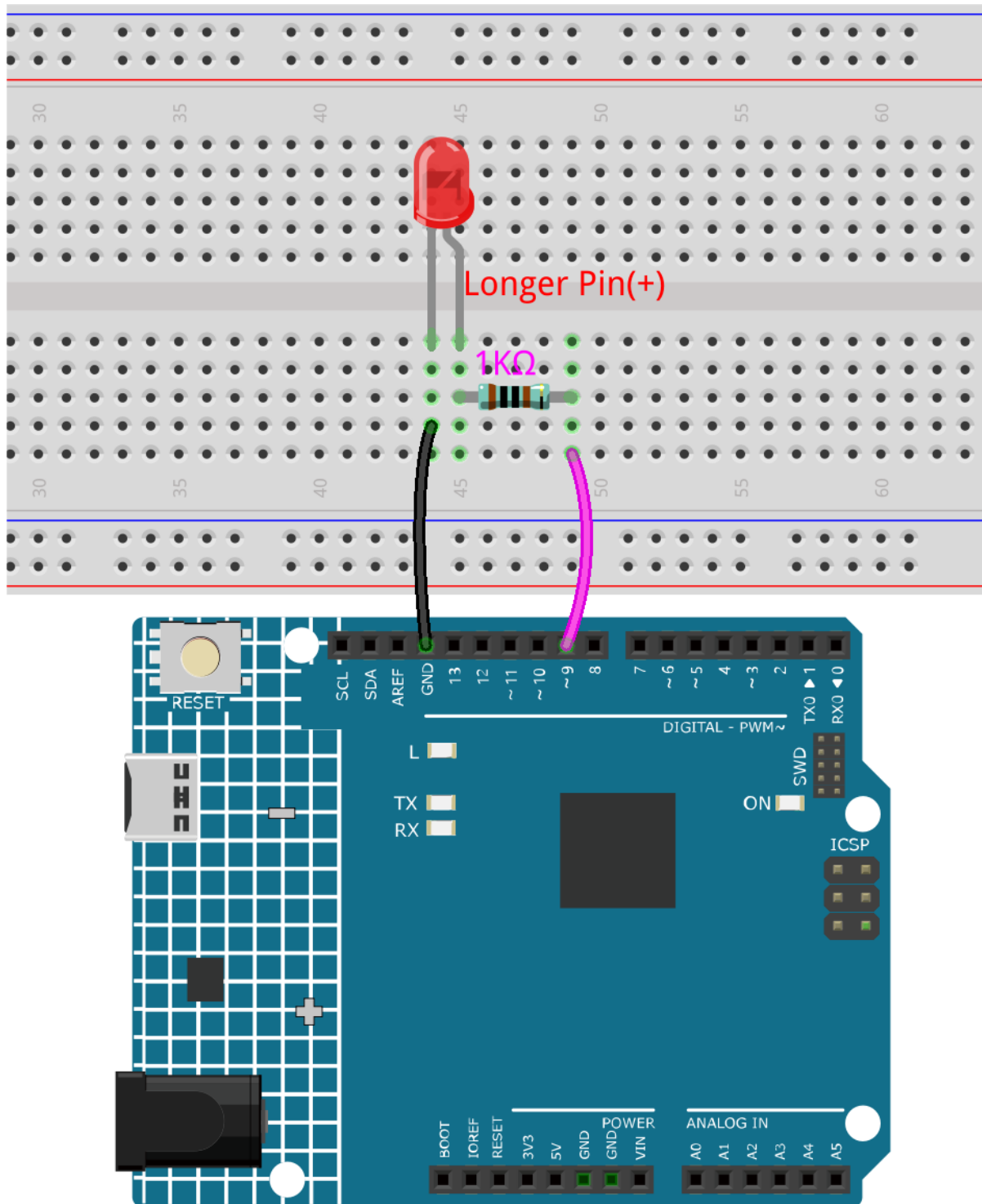
コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
<i>LED</i>	

回路図



この回路の原理は単純で、電流の方向は図に示されています。ピン 9 がハイレベル (5V) を出力すると、1k の電流制限抵抗を通して LED が点灯します。ピン 9 がローレベル (0V) を出力すると、LED は消灯します。

配線図



コード

注釈:

- ファイル 1.1.hello_led.ino をパス 3in1-kit\learning_project\1.1.hello_led で開くことができ

ます。

- または、このコードを **Arduino IDE** にコピーしてください。

コードが正常にアップロードされると、LED が点滅するのが見えるでしょう。

どのように動作するのか？

ここでは、LED をデジタルピン 9 に接続しているため、プログラムの始めに int 変数 ledpin を宣言し、9 の値を割り当てする必要があります。

```
const int ledPin = 9;
```

次に、setup() 関数内でピンを初期化します。ここでピンを OUTPUT モードに初期化する必要があります。

```
void setup() {  
    pinMode(ledPin, OUTPUT);  
}
```

loop() の中で、digitalWrite() を使用して ledpin に 5V の高レベル信号を供給し、LED ピン間に電圧差を生じさせて LED を点灯させます。

```
digitalWrite(ledPin, HIGH);
```

レベル信号を LOW に変更すると、ledPin の信号が 0 V に戻り、LED が消灯します。

```
digitalWrite(ledPin, LOW);
```

オンとオフの間隔が必要です。変化を見るために人々に時間を与えるため、delay(1000) コードを使用して、コントローラーが 1000 ms 何もしないようにします。

```
delay(1000);
```

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。

- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [] をクリックして今すぐ参加しましょう！

4.1.2 1.2 ビープ音

アクティブブザーは、LED を点灯させるのと同じくらい使いやすい典型的なデジタル出力デバイスです！

キットには 2 種類のブザーが含まれています。アクティブブザーを使用する必要があります。裏返してみると、封印された背面（露出した PCB でない方）が私たちの求めているものです。



必要な部品

このプロジェクトでは、以下のコンポーネントが必要です。

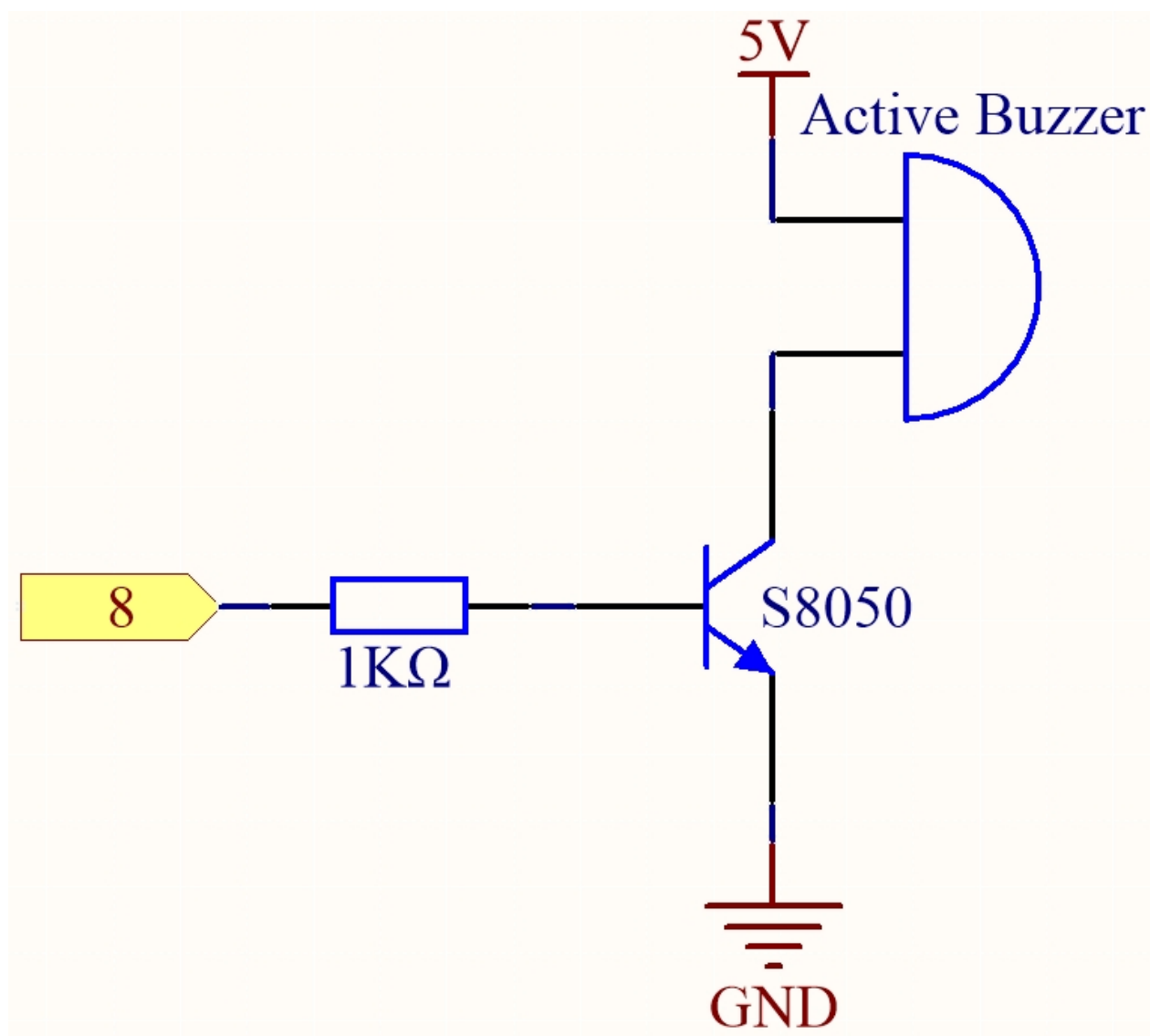
一式を購入するのが便利です、リンクはこちら：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

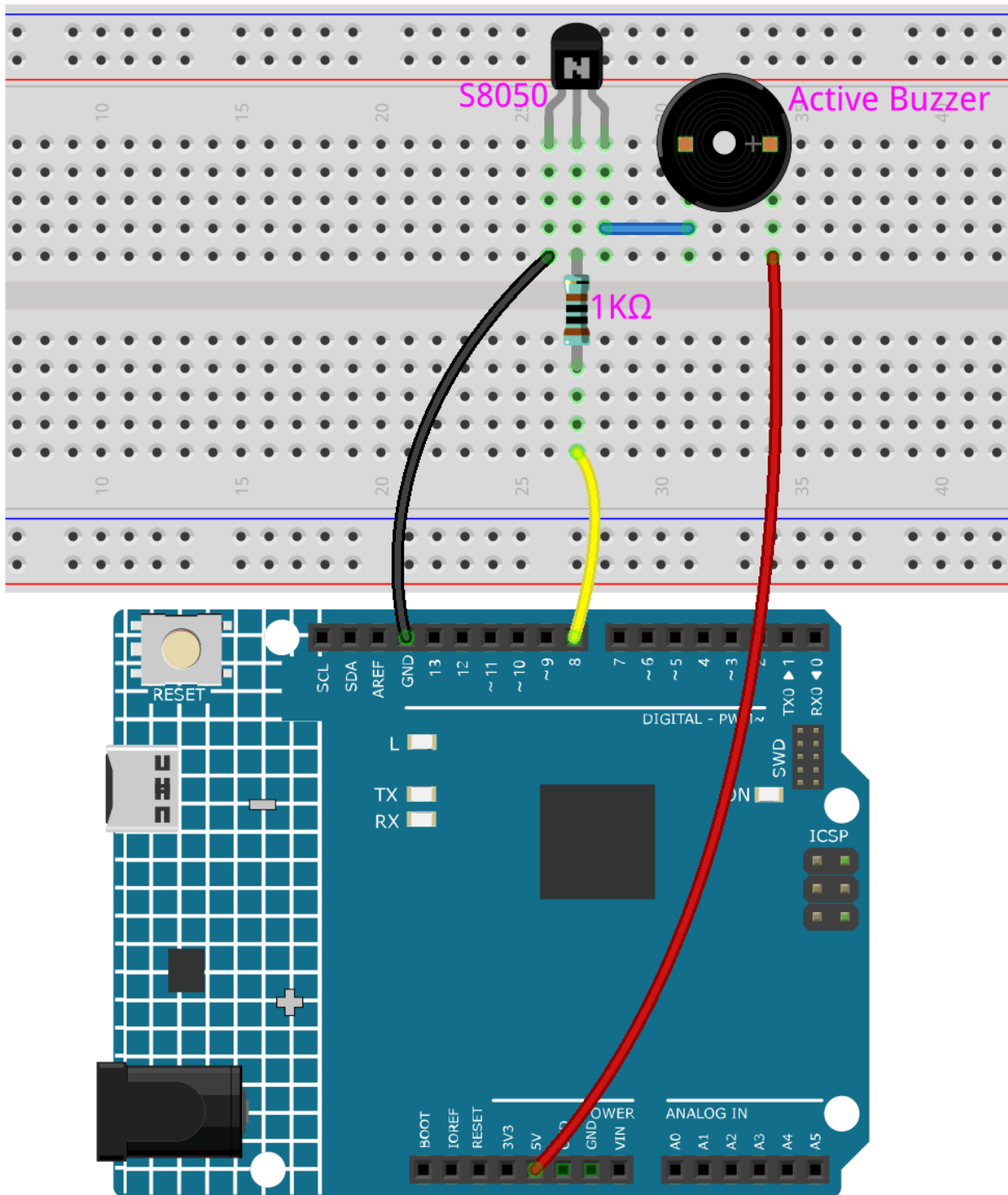
以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
Arduino Uno R4 Minima	-
ブレッドボード	
ジャンパーワイヤー	
ブザー	-

回路図



配線図



コード

注釈:

- ファイル 1.2.beep.ino をパス 3in1-kit\learning_project\1.2.beep で開くことができます。

- または、このコードを **Arduino IDE** にコピーしてください。

コードが正常にアップロードされた後、1 秒ごとにピープ音が聞こえるでしょう。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.1.3 1.3 車輪を回す

モーターは典型的なデジタル出力デバイスであり、LED と同じ方法で使用されます。ただし、モーターは大電流で駆動する必要があり、その大電流は R4 ボードなどのメインコントロールボードを損傷させる可能性があります。したがって、このような場合にはモータードライバモジュールが使用され、これは R4 ボードがモーターを安全に制御するのに役立ちます。

必要な部品

このプロジェクトでは、以下のコンポーネントが必要です。

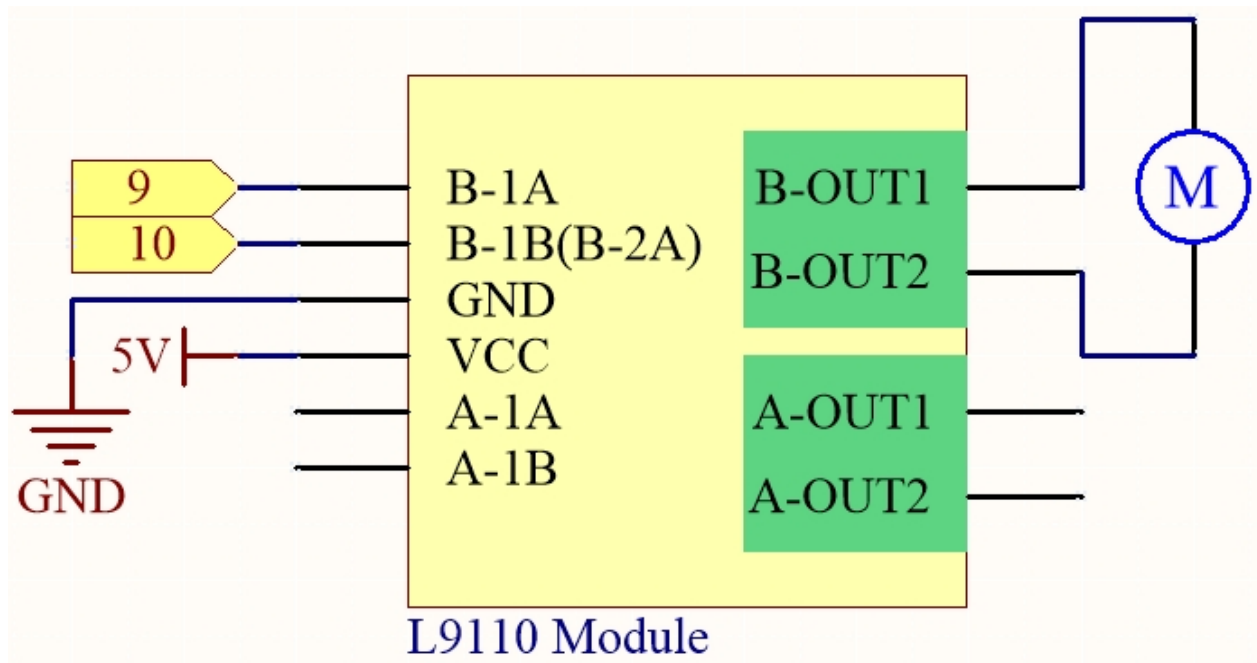
一式を購入するのが便利です、リンクはこちら：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから個別に購入することもできます。

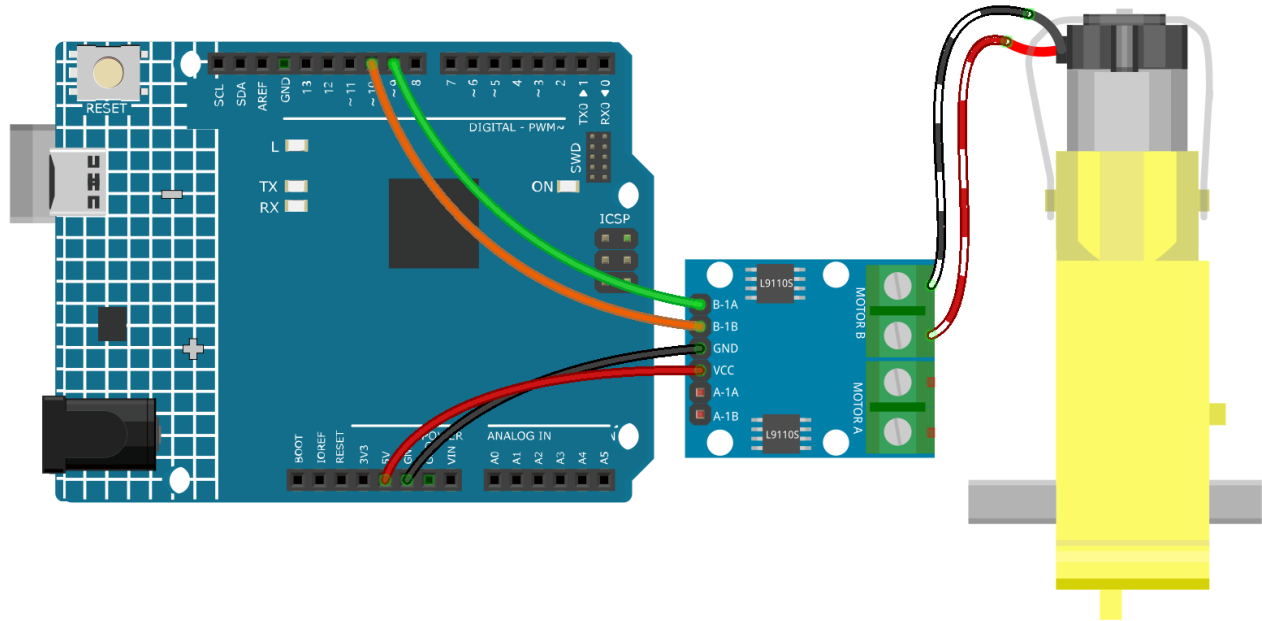
コンポーネントの紹介	購入リンク
Arduino Uno R4 Minima	-
ジャンパーワイヤー	
TT モーター	-
L9110 モータードライバーモジュール	-

回路図



配線図

L9110S	R4 ボード	モーター
VCC	5V	
GND	GND	
B-1B(B-2A)	9	
B-1A	10	
OA		モーターのワイヤー
OB		モーターのワイヤー



コード

注釈:

- ファイル 1.3.turn_the_wheel.ino をパス 3in1-kit\learning_project\1.3.turn_the_wheel で開くことができます。
- または、このコードを **Arduino IDE** にコピーしてください。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.1.4 1.4 ポンピング

水ポンプもモーターの一種であり、特別な構造を通じてモーターや他の外部エネルギーの機械的エネルギーを変換し、液体を輸送するものです。

必要な部品

このプロジェクトでは、以下のコンポーネントが必要です。

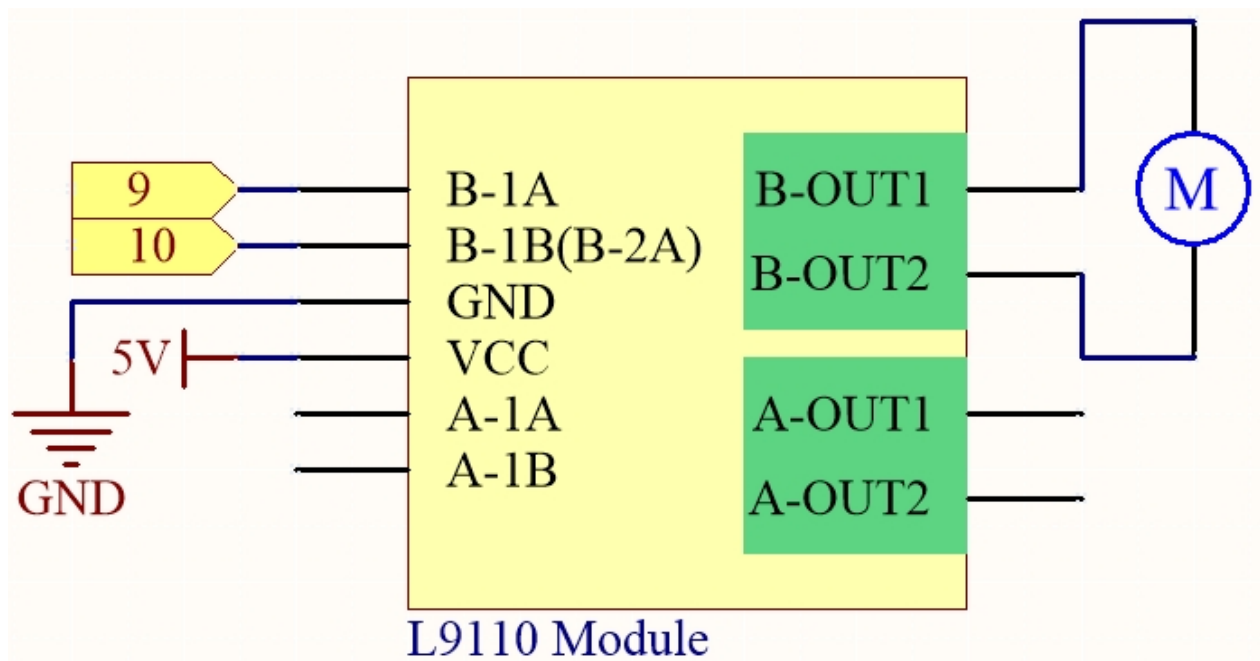
キット全体を購入すると確かに便利です、リンクはこちら：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから個別に購入することもできます。

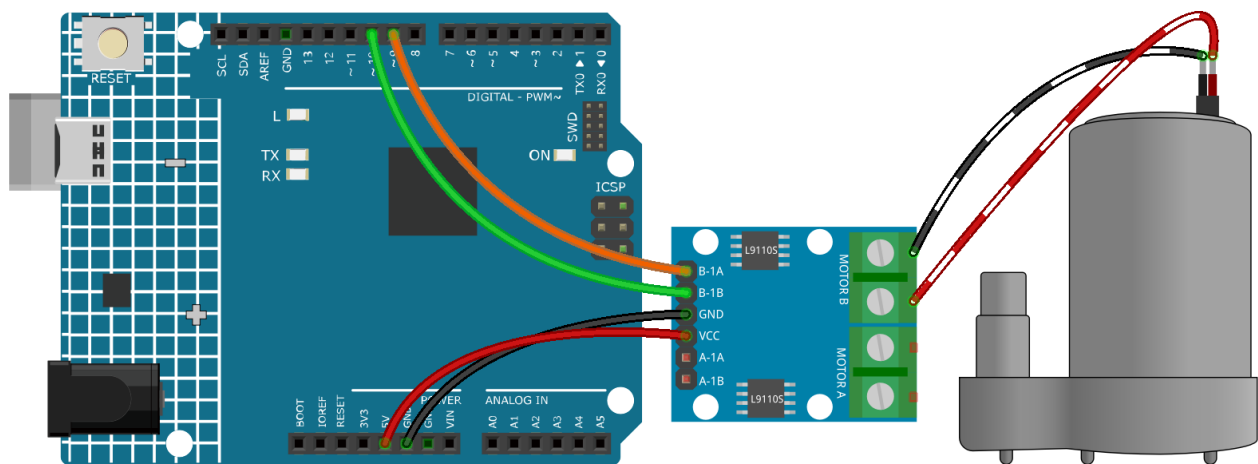
コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ジャンパーワイヤー	
<i>L9110 モータードライバーモジュール</i>	-
遠心ポンプ	-

回路図



配線図

L9110S	R4 ボード	モーター
VCC	5V	
GND	GND	
B-1B(B-2A)	9	
B-1A	10	
OA		モーターのワイヤー
OB		モーターのワイヤー



コード

注釈:

- ファイル 1.4.pumping.ino を 3in1-kit\learning_project\1.4.pumping のパスの下で開くことができます。
- または、このコードを **Arduino IDE** にコピーしてください。

ポンプにチューブを追加し、ベースンに置きます。コードが正常にアップロードされた後、しばらくするとベースンの水が排出されることが確認できます。この実験を行うときは、回路が水から離れているようにして、短絡を避けてください！

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。

- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [] をクリックして今すぐ参加しましょう！

4.2 2. アナログライト

Arduino の 14 のデジタルピンのうち 6 つは PWM 出力機能も備えています。したがって、これらの 6 つのピンにデジタル信号を書き込むだけでなく、アナログ信号（PWM 波信号）も書き込むことができます。これにより、LED を異なる明るさで点灯させたり、モーターを異なる速度で回転させることができます。

パルス幅変調、あるいは PWM は、デジタル手段でアナログ結果を得るための技術です。文字通りの意味を理解するのが難しいかもしれませんが、LED の強度を制御する例を以下に示します。

高レベルと低レベルで構成されるデジタル信号はパルスと呼ばれます。これらのピンのパルス幅は、ON/OFF の速度を変えることで調整できます。簡単に言えば、LED を点灯、消灯、再び点灯させる短い期間（例えば 20ms、ほとんどの人の視覚的停留時間）では、LED が消灯したことには気づかないかもしれませんが、光の明るさはわずかに弱くなります。この期間中、LED が点灯している時間が長いほど、LED の明るさは増します。つまり、一定の期間内で、パルスが広いほど、マイクロコントローラから出力される「電気信号の強さ」が大きくなります。

これが PWM 波を書き込むための関数です。

- `analogWrite(pin, value)`

ピンにアナログ値（PWM 波）を書き込む。指定されたパルス信号を生成することで、異なる出力電圧（0-5V）をシミュレートすることができます。この信号は、新しい read または write ステートメントで呼び出されるまでピンが保持します。

構文

```
analogWrite(pin, value)
```

パラメータ

- `pin`: 書き込む Arduino のピン。許容されるデータ型: `int`。
- `value`: デューティサイクル: 0（常にオフ）から 255（常にオン）の間。許容されるデータ型: `int`。

アナログライトの例

```
int pin = 9;      //pwm ピンに接続

void setup() {
  pinMode(pin, OUTPUT); // ピンを出力として設定
}

void loop() {
  for (int i = 0 ;i<255 ; i++){
    analogWrite(pin, i); //analogWrite の値は 0 から 255
    delay(30);
  }
}
```

注意と警告

- R4 ボードをよく見ると、"~"記号でマークされたピンはアナログ出力機能があります。
- ピン 5 と 6 で生成される PWM 出力は、予想よりもデューティサイクルが高くなる場合があります。これは、millis() 関数や delay() 関数との相互作用によるもので、これらの PWM 出力を生成するために使用される内部タイマと同じものを共有しているためです。これは主に低デューティサイクルの設定（例：0-10）で注意され、ピン 5 と 6 の出力を完全にオフにする値 0 がオフにならない場合があります。

関連するコンポーネント

以下は関連するコンポーネントです。クリックすると、その使用方法を学ぶことができます。

注釈： こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なレビュー：新製品の発表や先行レビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.2.1 2.1 フェーディング

このプロジェクトは [1.1 こんにちは、LED!](#) に似ていますが、違いは信号の種類です。前者はデジタル信号 (0&1) を出力して LED を点灯または消灯させるのに対し、このプロジェクトはアナログ信号を出力して LED の明るさを制御します。

必要な部品

このプロジェクトでは、以下のコンポーネントが必要です。

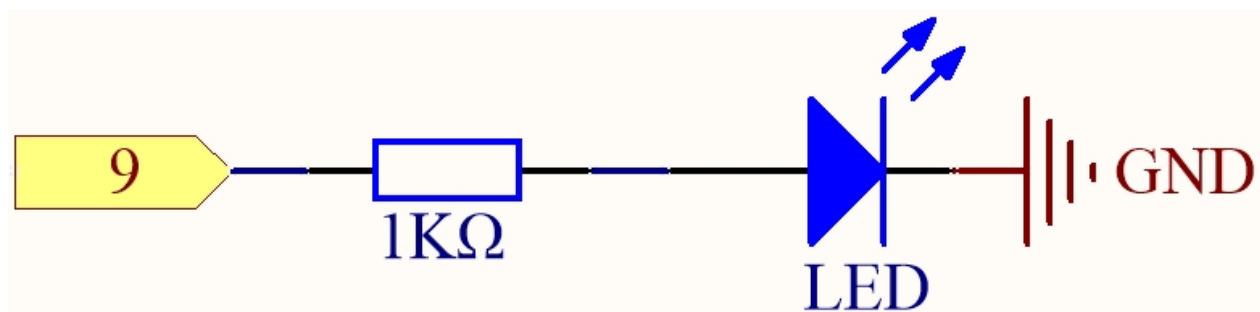
一式を購入するのが便利です。こちらのリンクから購入できます：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

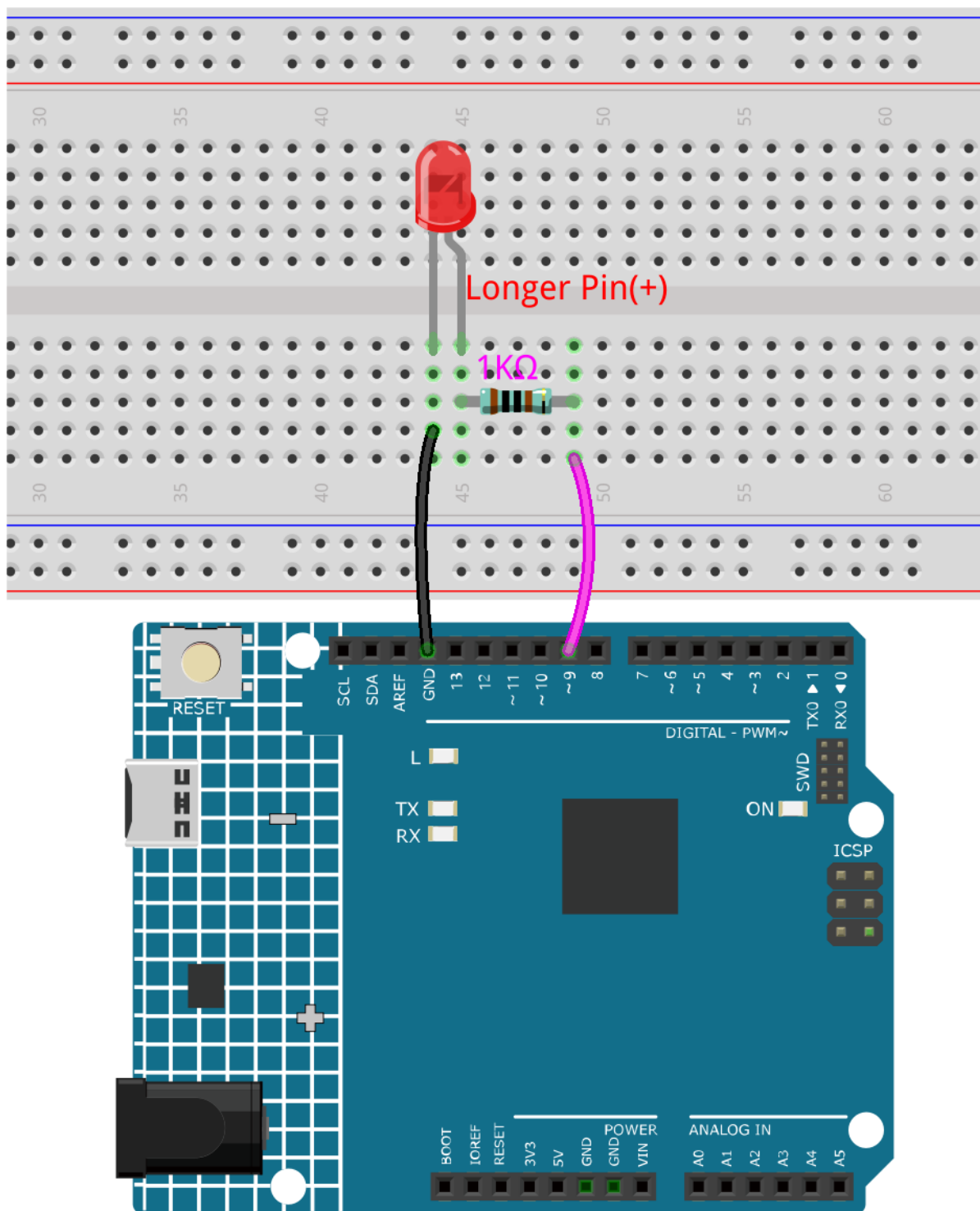
以下のリンクから、個別に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
<i>LED</i>	

回路図



配線図



コード

注釈:

- 3in1-kit\learning_project\2.analogWrite\2.1.fading のパスの下で 2.1.fading.ino ファイルを

開くことができます。

- または、このコードを **Arduino IDE** にコピーしてください。
-

コードが正常にアップロードされた後、LED が呼吸するように点滅するのが見えます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び & 共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.2.2 2.2 カラフルな光

私たちは知っているように、光は重ね合わせることができます。例えば、青色の光と緑色の光を混ぜるとシアン色の光になり、赤色の光と緑色の光を混ぜると黄色の光になります。これを"色の加算混合法"と呼びます。

- [加算色 - ウィキペディア](#)

この方法を基に、三原色を使って、異なる比重に応じて任意の色の可視光を混合することができます。例えば、赤色を多く、緑色を少なくするとオレンジ色が生成されます。

この章では、RGB LED を使用して、加算色混合の神秘を探索します！

RGB LED は、赤、緑、青の LED を 1 つのランプキャップの下に封入するものと同等であり、3 つの LED は共通のカソードピンを共有しています。各アノードピンに電気信号が供給されると、対応する色の光が表示されます。各アノードの電気信号の強度を変えることで、さまざまな色を生み出すことができます。

必要な部品

このプロジェクトには、以下のコンポーネントが必要です。

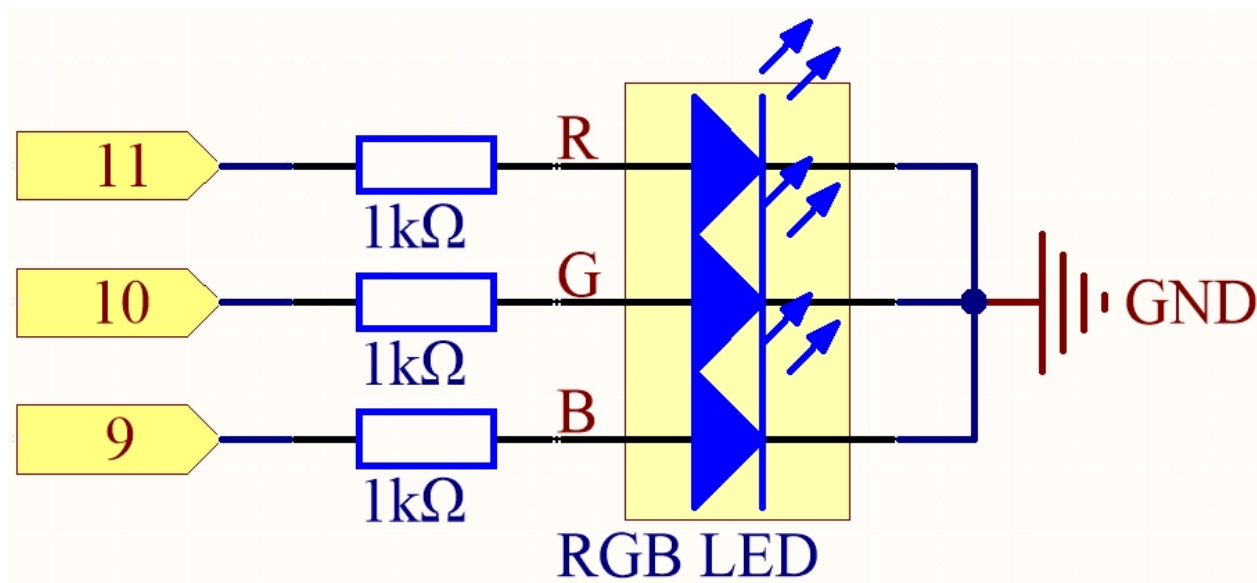
一式を購入するのが便利です。こちらのリンクから購入できます：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから、個別に購入することもできます。

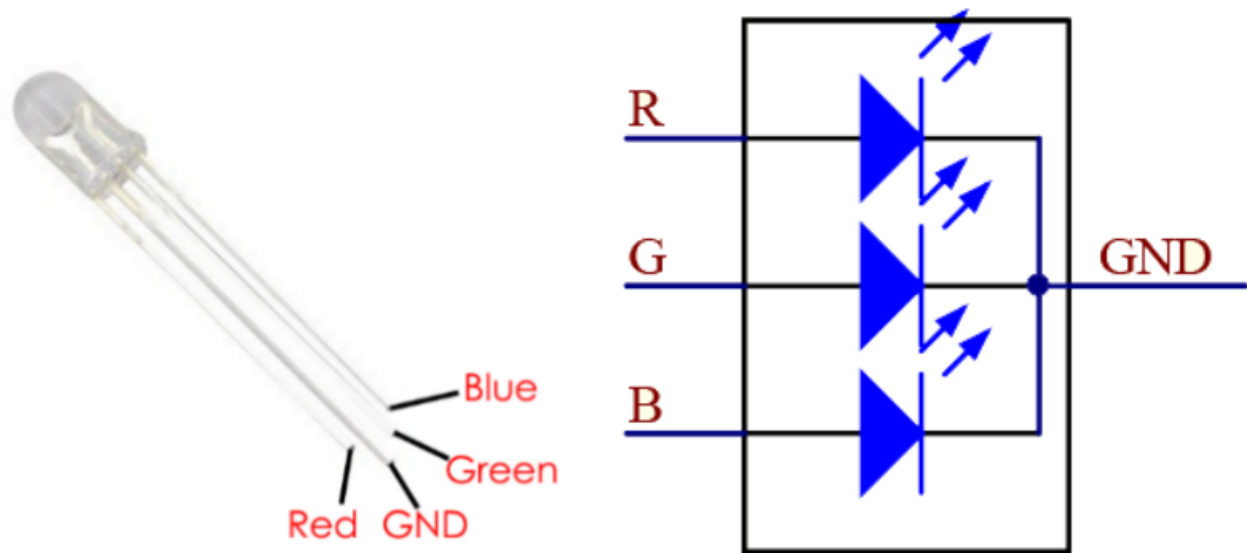
コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
<i>RGB LED</i>	

回路図

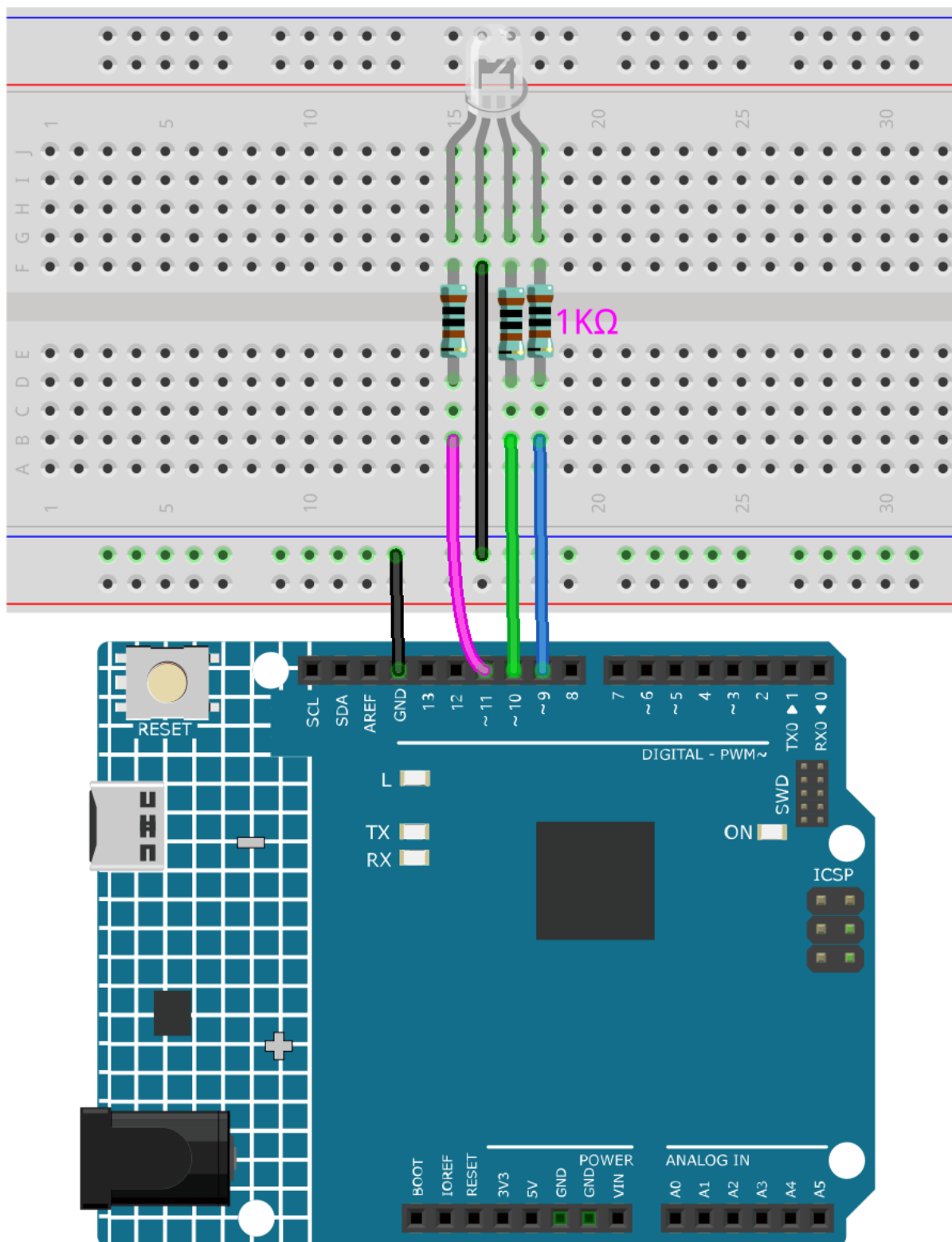


PWM ピンの 11、10、9 は、それぞれ RGB LED の赤、緑、青のピンを制御し、共通のカソードピンを GND に接続します。これにより、RGB LED は、これらのピンに異なる PWM 値で光を重ね合わせることで、特定の色を表示することができます。

配線図



RGB LED には 4 つのピンがあります：最も長いピンは共通のカソードピンで、通常 GND に接続され、最も長いピンの隣の左のピンは赤色で、右側の 2 つのピンは緑色と青色です。

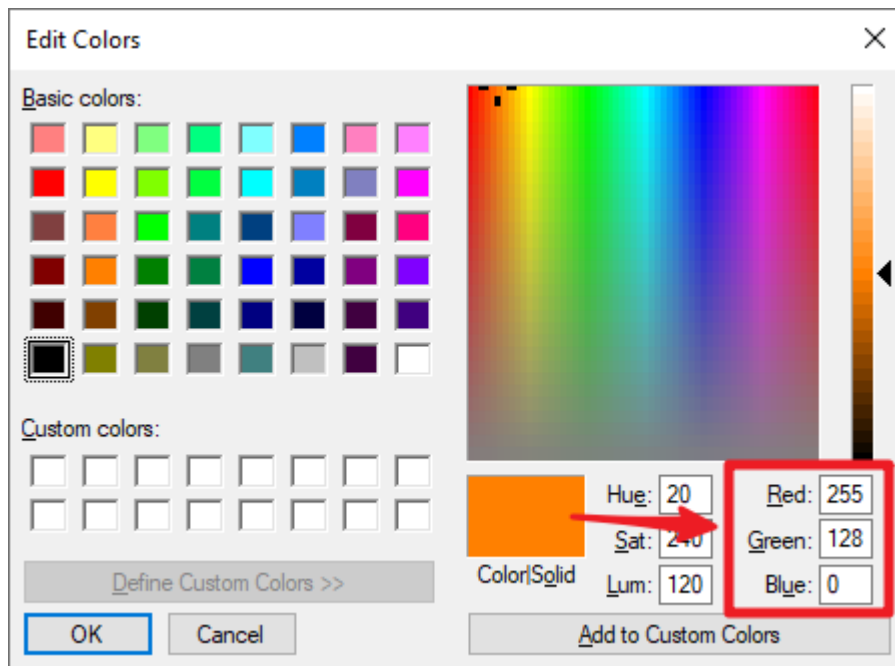


コード

ここで、お気に入りの色を描画ソフトウェア（ペイントなど）で選び、RGB LED で表示することができます。

注釈:

- 3in1-kit\learning_project\2.analogWrite\2.2.colorful_light のパスの下で 2.2.colorful_light.ino ファイルを開くことができます。
- または、このコードを **Arduino IDE** にコピーしてください。



RGB 値を `color_set()` に書き込むと、希望する色で RGB が点灯します。

どのように動作するのか？

この例では、RGB の 3 つのピンに値を割り当てるために使用される関数は、独立したサブ関数 `color()` にパッケージされています。

```
void color (unsigned char red, unsigned char green, unsigned char blue)
{
    analogWrite(redPin, red);
    analogWrite(greenPin, green);
    analogWrite(bluePin, blue);
}
```

`loop()` 内では、RGB 値は入力引数として機能し、関数 `color()` を呼び出して RGB が異なる色を放出することを実現しています。

```

void loop()
{
    color(255, 0, 0); // red
    delay(1000);
    color(0, 255, 0); // green
    delay(1000);
    color(0, 0, 255); // blue
    delay(1000);
}

```

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なレビュー：新製品の発表や先行レビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.3 3. デジタルリード

センサーは実世界の情報をキャッチし、この情報はピン（デジタルやアナログのもの）を介してメインボードに伝達されます。これにより、コンピュータは現状を知ることができます。

したがって、Arduino ボードは、ボタンや IR 障害回避モジュールのようなデジタルピンの値を読み取ることで、デジタルセンサーの動作状態を知ることができます。

以下は、必要な関数です。

- `pinMode(pin, mode)`: 特定のピンを INPUT または OUTPUT として設定します。ここでは INPUT として設定する必要があります。

文法

```
pinMode(pin, mode)
```

パラメータ

- pin: モードを設定する Arduino のピン番号。
 - mode: INPUT, OUTPUT, または INPUT_PULLUP。
- digitalRead(pin): 指定されたデジタルピンからの値（レベル状態）を読み取ります。

文法

```
digitalRead(pin)
```

パラメータ

- pin: 読み取りたい Arduino のピン番号

返り値

HIGH または LOW

デジタルリードの例

```
int ledPin = 13; // LED はデジタルピン 13 に接続
int inPin = 7;   // プッシュボタンはデジタルピン 7 に接続
int val = 0;     // 読み取った値を保存する変数

void setup() {
  pinMode(ledPin, OUTPUT); // デジタルピン 13 を出力として設定
  pinMode(inPin, INPUT);   // デジタルピン 7 を入力として設定
}

void loop() {
  val = digitalRead(inPin); // 入力ピンを読む
  digitalWrite(ledPin, val); // LED をボタンの値に設定する
}
```

注意と警告

1. プルアップ&プルダウン。

digitalRead() はピンがレベル信号を取得していない場合、ランダムな値を生成する可能性があります。入力ピンを既知の状態に指示することで、プロジェクトの信頼性を高めることができます。ボタンなどの入力部品を使用する場合、デジタル入力ピンに並列にプルアップまたはプルダウン抵抗を接続することが多いです。

プルアップ抵抗を接続するのとは別に、コード内でピンモードを INPUT_PULLUP に設定することもできます。たとえば、pinMode(pin, INPUT_PULLUP)。この場合、ピンはソフトウェア経由で Atmega の内蔵プルアップ抵抗にアクセスし、プルアップ抵抗を接続するのと同じ効果が得られます。

2. ピン 13 について。

R4 ボード上のすべてのデジタルピン (1-13) は `digitalRead()` として使用できます。しかし、デジタルピン 13 をデジタル入力として使用するの、他のデジタルピンよりも難しいです。これは、LED と抵抗が接続されており、ほとんどのボードにはんだ付けされているからです。内蔵の 20k プルアップ抵抗を有効にすると、期待される 5V の代わりに 1.7V 前後になります。これは、オンボードの LED と直列抵抗が電圧レベルを低くするためです。ピン 13 をデジタル入力として使用する必要がある場合は、その `pinMode()` を INPUT に設定し、外部のプルダウン抵抗を使用してください。

3. アナログピン。

デジタルピンが足りない場合、アナログピン (A0-A5) もデジタルピンとして使用できます。
`pinMode(pin,mode)` で INPUT に設定する必要があります。

関連コンポーネント

以下は関連するコンポーネントです。クリックして使用方法を学ぶことができます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なレビュー：新製品の発表や先行レビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.3.1 3.0 シリアルモニター

Arduino IDE には、コンピュータから Arduino ボードへ (USB 経由で) メッセージを送信することができるシリアルモニターがあります。また、Arduino からのメッセージを受信することもできます。

このプロジェクトでは、Arduino ボードからデータを受信する方法を学びます。

注釈: Uno、Nano、Mini、Mega では、0 ピンと 1 ピンがコンピュータとの通信に使用されています。これらのピ

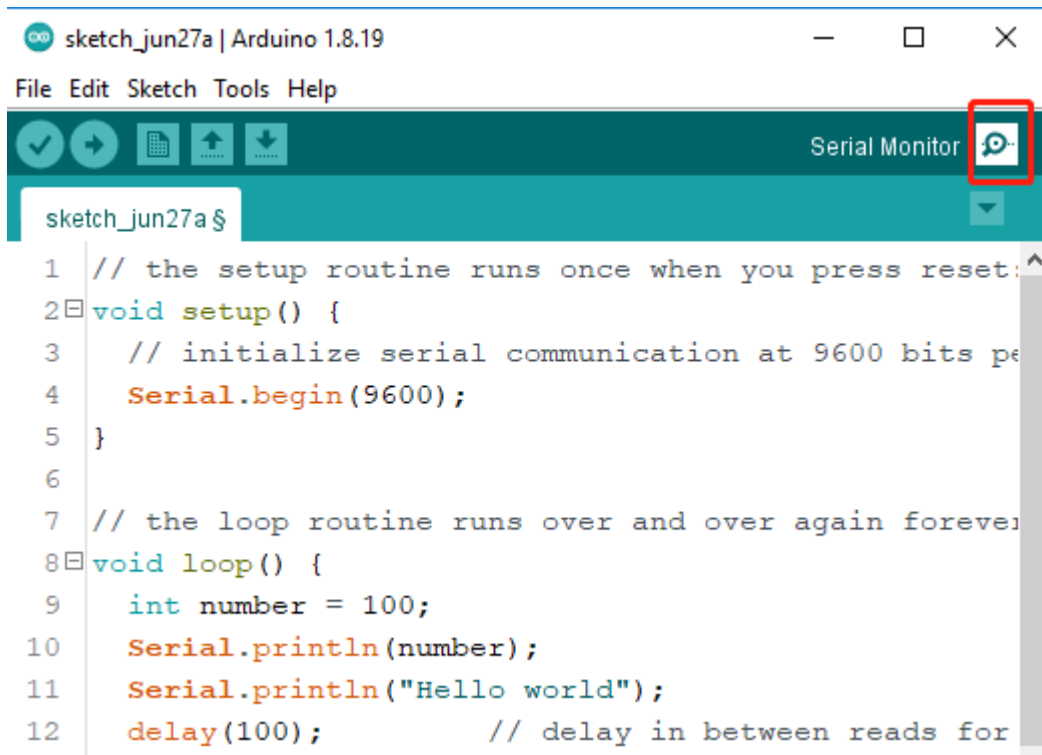
ンに何かを接続すると、ボードへのアップロードの失敗を含む通信の干渉が発生する可能性があります。

シリアルモニターの使用方法

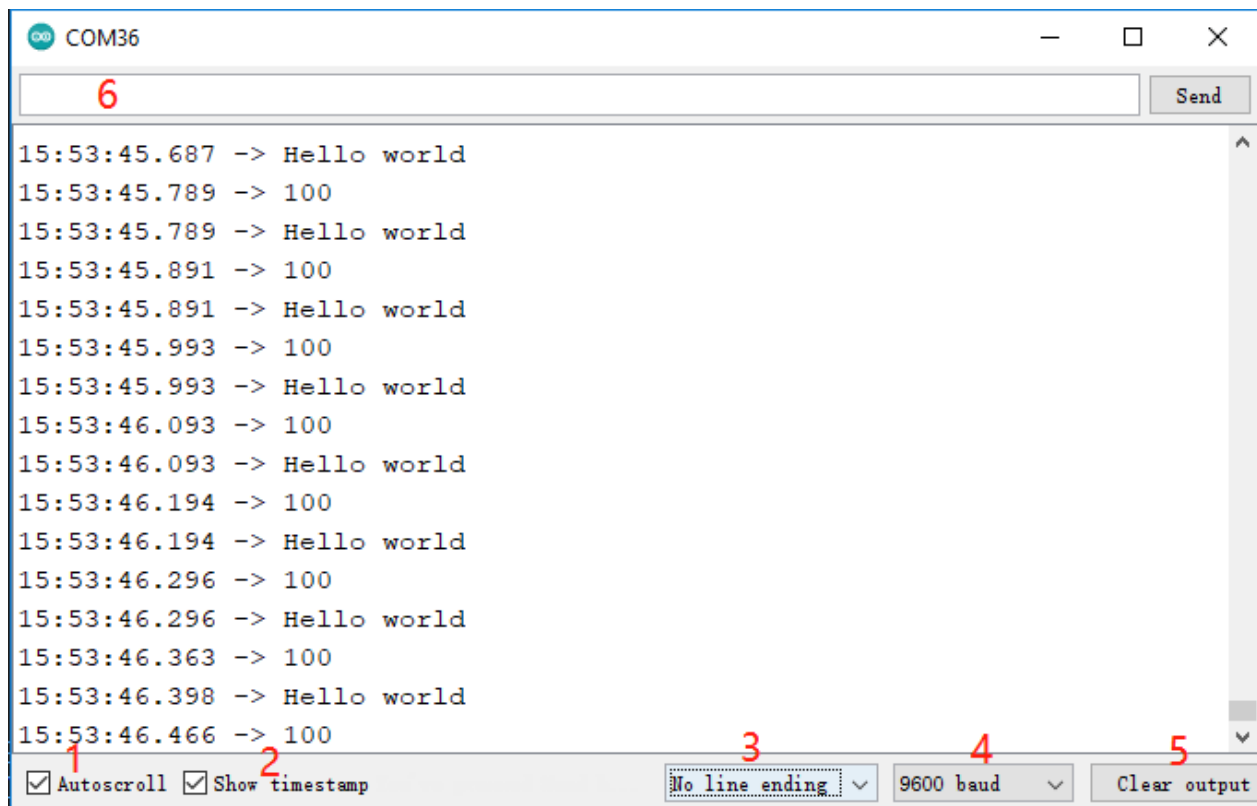
1. Arduino IDE を開き、以下のコードを貼り付けます。

```
// セットアップルーチンはリセットボタンを押すと一度だけ実行されます:  
void setup() {  
    // シリアル通信を 9600 ビット/秒で初期化:  
    Serial.begin(9600);  
}  
  
// ループルーチンは永遠に繰り返し実行されます:  
void loop() {  
    int number = 100;  
    Serial.println(number);  
    Serial.println("Hello world");  
    delay(100);          // 安定した読み取りのための間隔  
}
```

- `Serial.begin()`: シリアルデータ伝送のビットレート（ボーレート）を設定します。ここでは 9600 に設定。
 - `Serial.println()`: キャリッジリターン文字 (ASCII 13, 'r') および改行文字 (ASCII 10, 'n') に続く人間が読める ASCII テキストとしてシリアルポートにデータを出力します。このコマンドは `Serial.print()` と同じ形式を取ります。
2. コードをアップロードするための正しいボードとポートを選択します。
 3. ツールバーで虫眼鏡のアイコンをクリックして、シリアルモニターをオンにします。



4. これがシリアルモニターの画面です。



- 1: 自動的にスクロールするかどうかを選択するオプション。

- 2: シリアルモニタに表示されるデータの前にタイムスタンプを表示するオプション。
- 3: 終了選択、Arduino に送信されるデータに追加される終了文字を選択します。選択肢には以下が含まれます：
 - 行の終わりなし は入力したものだけを送信;
 - 改行 は \n で、入力した後に ASCII の改行コードを送信;
 - キャリッジリターン は \r で、入力した後に ASCII のキャリッジリターン文字を送信;
 - NL & CR の両方 は \r\n で、入力した後にキャリッジリターンと改行の両方の文字を送信。
- 4: Arduino ボードと PC との通信速度を選択します。この値は `Serial.begin()` で設定した値と同じでなければなりません。
- 5: 出力コンソール上の全テキストをクリア。
- 6: Arduino ボードに文字を送信するためのテキストボックス。チュートリアルは [5.12 シリアル読み取り](#) を参照してください。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び & 共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.3.2 3.1 ボタンの値の読取り

前回のプロジェクトでは出力関数を使用しましたが、この章では入力関数を使用してボタンの値を読み取ります。

必要な部品

このプロジェクトでは、以下のコンポーネントが必要です。

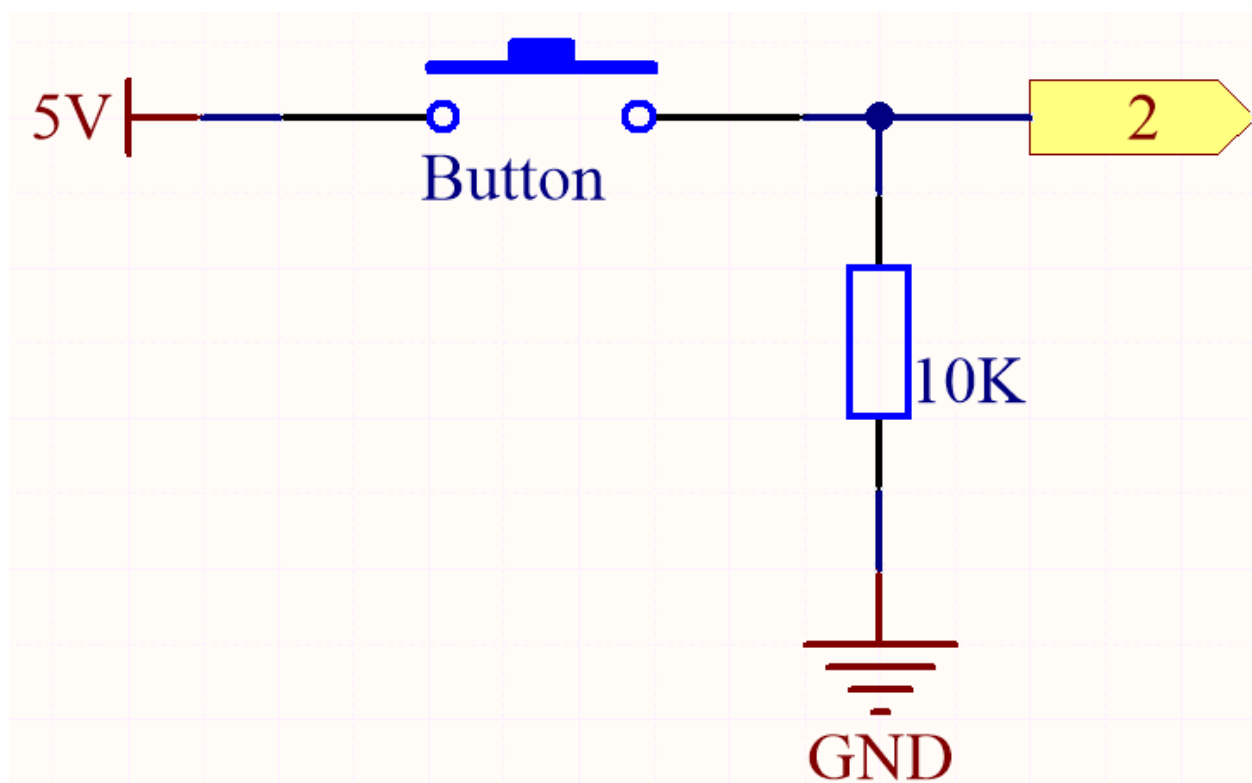
一式を購入すると非常に便利です。こちらがリンクです：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクからそれぞれ別々に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
ボタン	

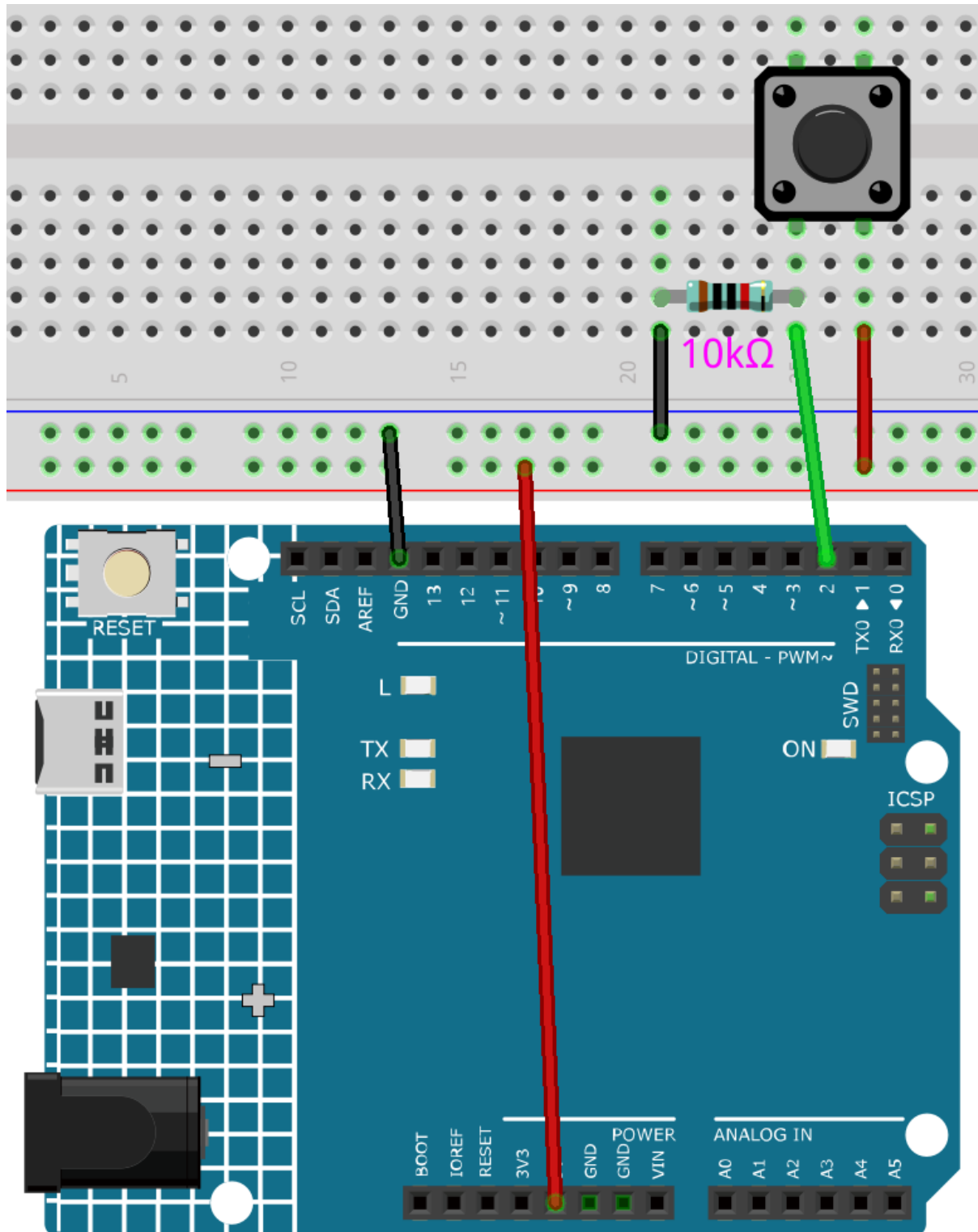
回路図



ボタンの片側のピンは5Vに接続され、もう一方のピンはピン2に接続されているため、ボタンが押されると、ピン2はハイになります。しかし、ボタンが押されていない場合、ピン2は浮遊状態であり、ハイまたはローのどちら

らになる可能性があります。ボタンが押されていないときに安定したローレベルを得るために、ピン 2 は 10K のプルダウン抵抗を通して GND に再接続する必要があります。

配線図

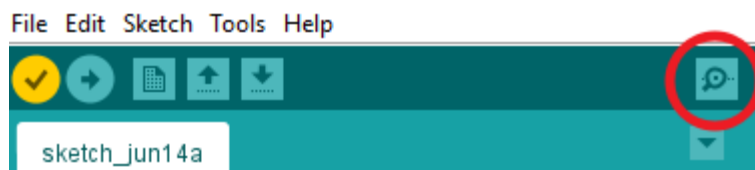


コード

注釈:

- 3in1-kit\learning_project\3.1.read_button_value のパスの下で 3.1.read_button_value.ino ファイルを開くことができます。
 - または、このコードを **Arduino IDE** にコピーしてください。
-

コードが正常にアップロードされた後、Arduino IDE の右上隅にある虫眼鏡のアイコン（シリアルモニタ）をクリックします。



ボタンを押すと、シリアルモニタに「1」と表示されます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [] をクリックして今すぐ参加しましょう！

4.3.3 3.2 磁気を感じる

最も一般的なリードスイッチには、スイッチが開いているときに小さなギャップによって端部が分離されている、磁気化可能な柔軟な金属のリードのペアが含まれています。

電磁石や永久磁石からの磁場は、リードが互いに引き付けるように作用し、電気回路を完成させます。リードのバネ力は、磁場が停止すると、それらを分離させて回路を開きます。

リードスイッチの一般的な例は、セキュリティアラームのためのドアや窓の開放を検出することです。

必要な部品

このプロジェクトには、以下の部品が必要です。

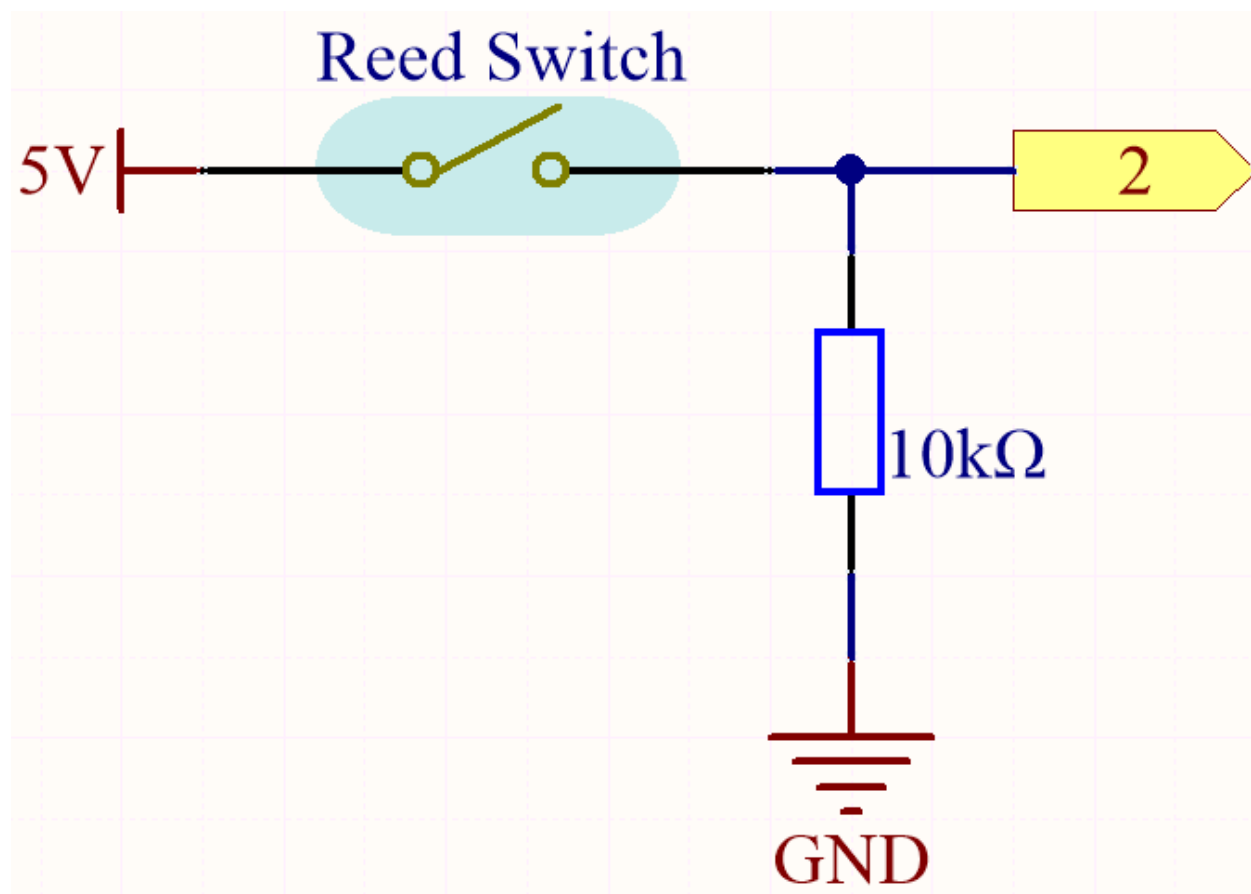
キット全体を購入することは確かに便利です。こちらがリンクです:

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
リードスイッチ	-

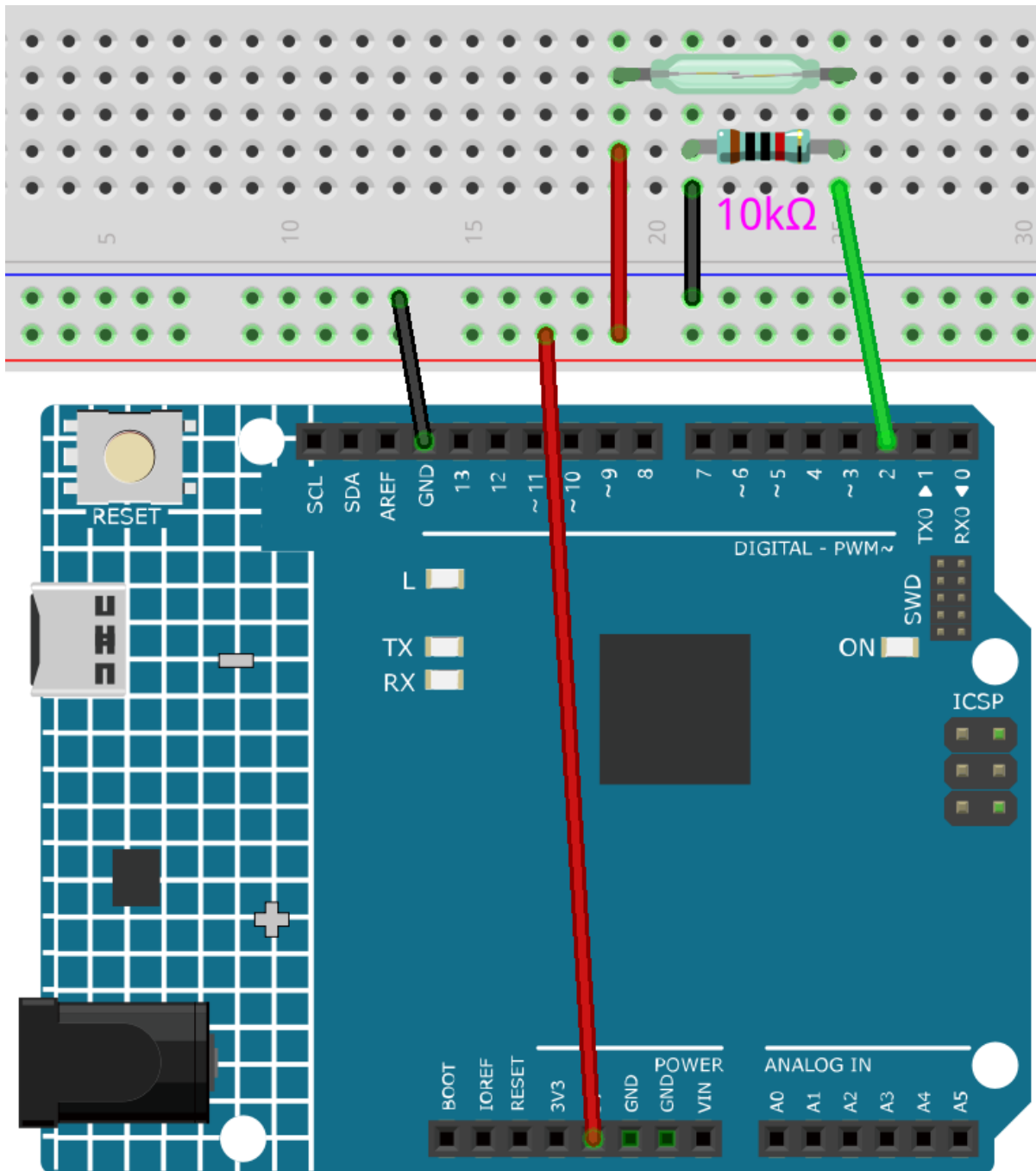
回路図



デフォルトでは、ピン 2 はローです。磁石がリードスイッチの近くにあると、ハイになります。

10K の抵抗の目的は、磁石が近くにいるときにピン 2 を安定したローレベルに保つことです。

配線図



コード

注釈:

- 3in1-kit\learning_project\3.2.feel_the_magnetism のパスの下にある 3.2.feel_the_magnetism.ino ファイルを開くことができます。
- または、このコードを **Arduino IDE** にコピーしてください。

コードが正常にアップロードされた後、磁石がリードスイッチの近くにあると、シリアルモニターは 1 を印刷します。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.3.4 3.3 障害物を検出する

このモジュールは、前方の障害物の存在を判断するために車やロボットに一般的に取り付けられています。また、携帯デバイスや水道の蛇口など、幅広い用途で使用されています。

必要な部品

このプロジェクトには、以下の部品が必要です。

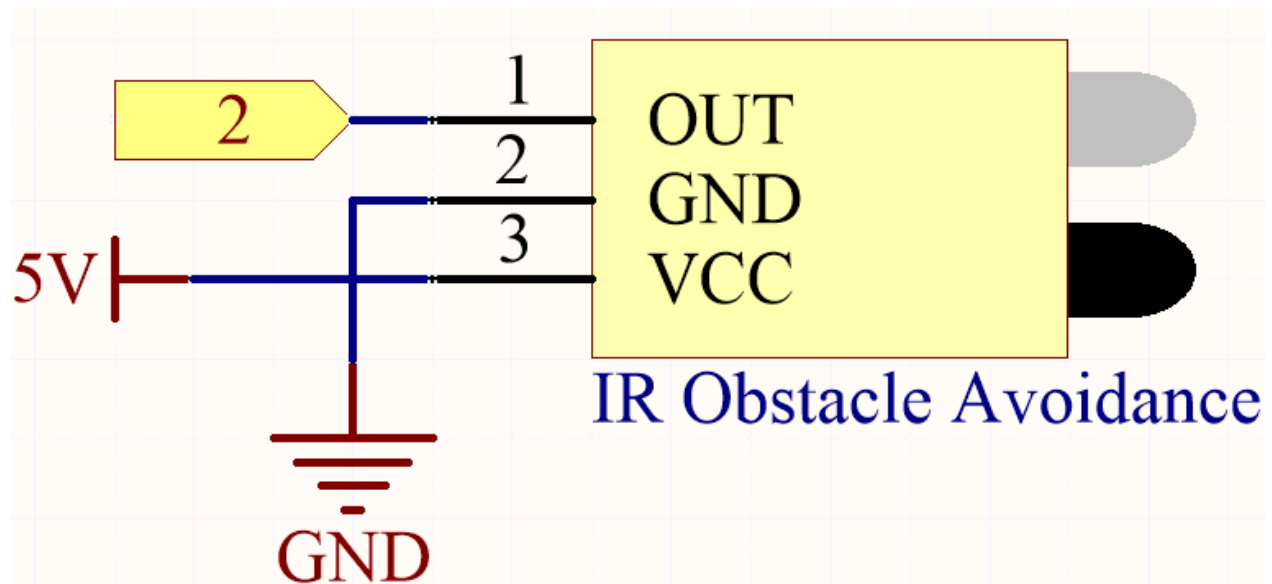
キット全体を購入するのは非常に便利です。こちらがリンクです:

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから個別に購入することもできます。

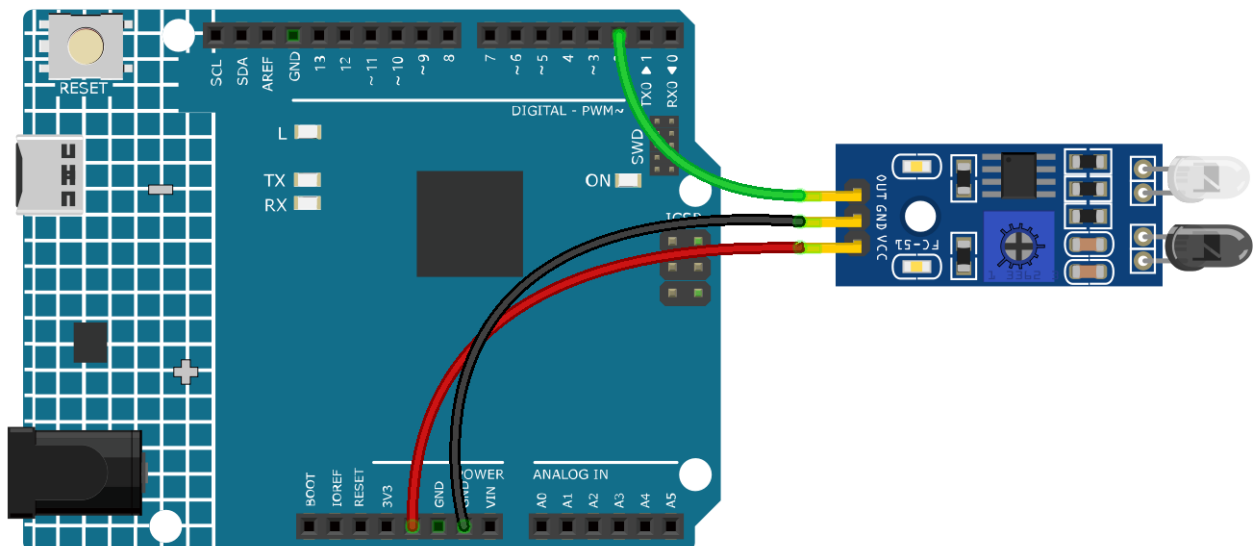
コンポーネントの紹介	購入リンク
Arduino Uno R4 Minima	-
ジャンパーワイヤー	
障害物回避モジュール	

回路図



デジタルピン 2 は IR 障害物回避モジュールの信号を読むために使用されます。IR センサーモジュールの VCC を 5V に、GND を GND に、OUT をデジタルピン 2 に接続します。

配線図



コード

注釈:

- 3in1-kit\learning_project\3.3.detect_the_obstacle のパスの下にある 3.3.detect_the_obstacle.ino ファイルを開くことができます。
 - または、このコードを **Arduino IDE** にコピーしてください。
-

IR 障害物回避モジュールが前方に何かをブロックしていることを検出すると、シリアルモニターに [0] が表示され、それ以外の場合は [1] が表示されます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び & 共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.3.5 3.4 ラインを検出する

ライン追跡モジュールは、電気テープで貼られた黒いラインなど、地面に黒いエリアがあるかどうかを検出するために使用されます。

モジュールの LED の一つが地面に適切な赤外線光を放射し、黒い表面は光を吸収する能力が比較的強く、反射能力が弱いです。白い表面はその逆です。反射光を検出した場合、現在の地面は白いことを意味します。検出されなかった場合、それは黒いことを意味します。

これが動作原理です。

必要な部品

このプロジェクトには、以下の部品が必要です。

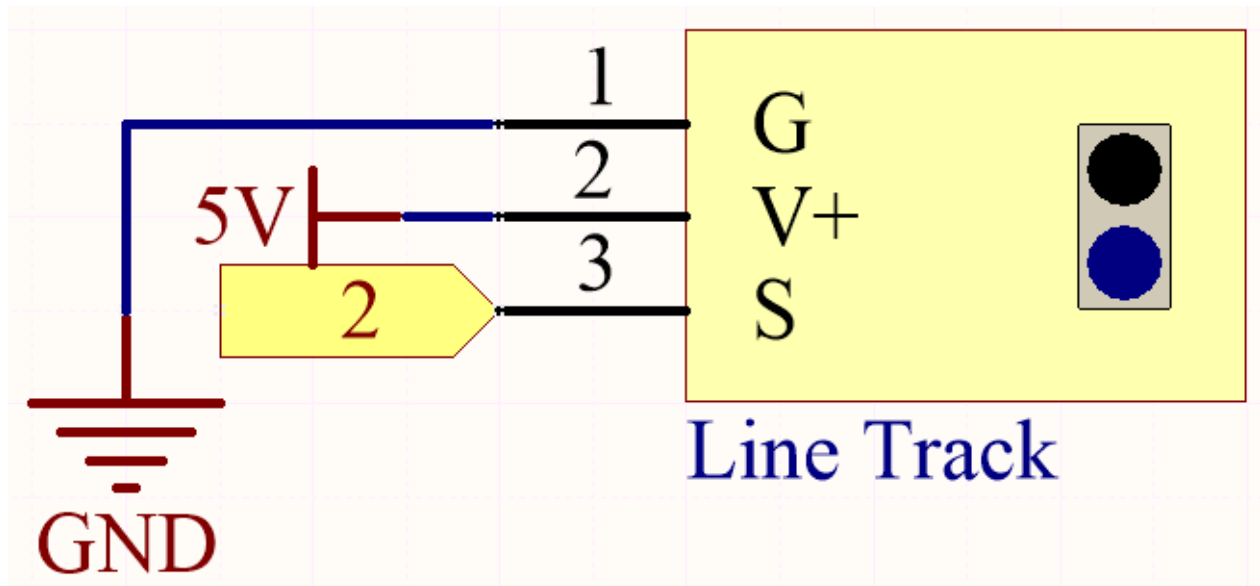
キット全体を購入するのは非常に便利です。こちらがリンクです:

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから個別に購入することもできます。

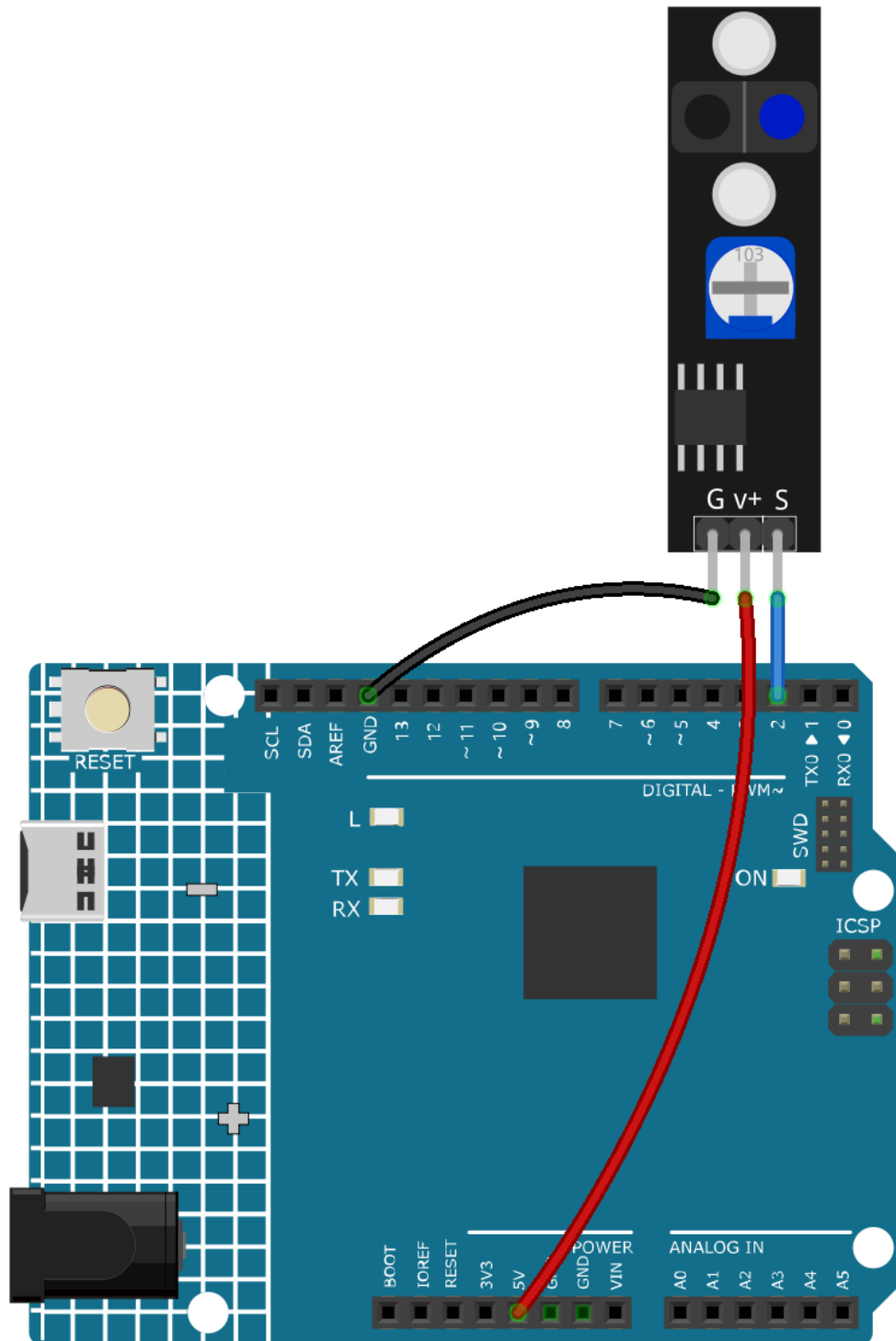
コンポーネントの紹介	購入リンク
Arduino Uno R4 Minima	-
ジャンパーワイヤー	
ライントラッキングモジュール	

回路図



デジタルピン 2 はライン追跡モジュールの信号を読むために使用されます。モジュールの VCC を 5V に接続し、GND を GND に、OUT をデジタルピン 2 に接続します。

配線図



コード

注釈:

- 3in1-kit\learning_project\3.4.detect_the_line のパスの下にある 3.4.detect_the_line.ino

ファイルを開くことができます。

- または、このコードを **Arduino IDE** にコピーしてください。
-

ライン追跡モジュールが黒いラインを検出すると、シリアルモニタに [1] が表示され、それ以外の場合は [0] が表示されます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なレビュー：新製品の発表や先行レビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.4 4. アナログ読み取り

Arduino は、アナログピンを介して接続されたアナログセンサを読み取ることができます。

R4 ボードには、多チャンネル、10 ビットのアナログ-デジタル変換器が含まれています。これは、0 から動作電圧（5V または 3.3V）の間の入力電圧を、0 から 1023 の間の整数値にマッピングすることを意味します。

アナログピンの値を読み取るには、`analogRead(pin)` 関数が必要です。

- `analogRead(pin)`: 指定したアナログピンからの値を読み取ります。

文法

`analogRead(pin)`

パラメータ

- `pin`: 値を読み取るアナログ入力ピンの名前（A0 から A5）。

戻り値

0-1023。データタイプ: `int`。

アナログ読み取りの例

```

int analogPin = A0; // アナログピン A0 にデバイスが接続されている
                    // 外部はグラウンドと +5V に接続
int val = 0; // 読み取った値を格納する変数

void setup() {
    Serial.begin(9600); // シリアルの設定アップ
}

void loop() {
    val = analogRead(analogPin); // 入力ピンを読み取る
    Serial.println(val); // 値をデバッグ
}

```

注意と警告

- アナログピンは A0-A5 です。
- アナログピンを呼び出す前に pinMode() を呼び出す必要はありませんが、ピンが以前に OUTPUT に設定されていた場合、analogRead() 関数は正しく動作しません。その場合、pinMode() を呼び出して INPUT に戻す必要があります。

関連するコンポーネント

以下は関連するコンポーネントで、クリックして使用方法を学ぶことができます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なレビュー：新製品の発表や先行レビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.4.1 4.1 ノブを回す

ポテンショメータは、3つの端子を持つ抵抗コンポーネントであり、その抵抗値は一定の変化に従って調整できます。

必要な部品

このプロジェクトでは、以下の部品が必要です。

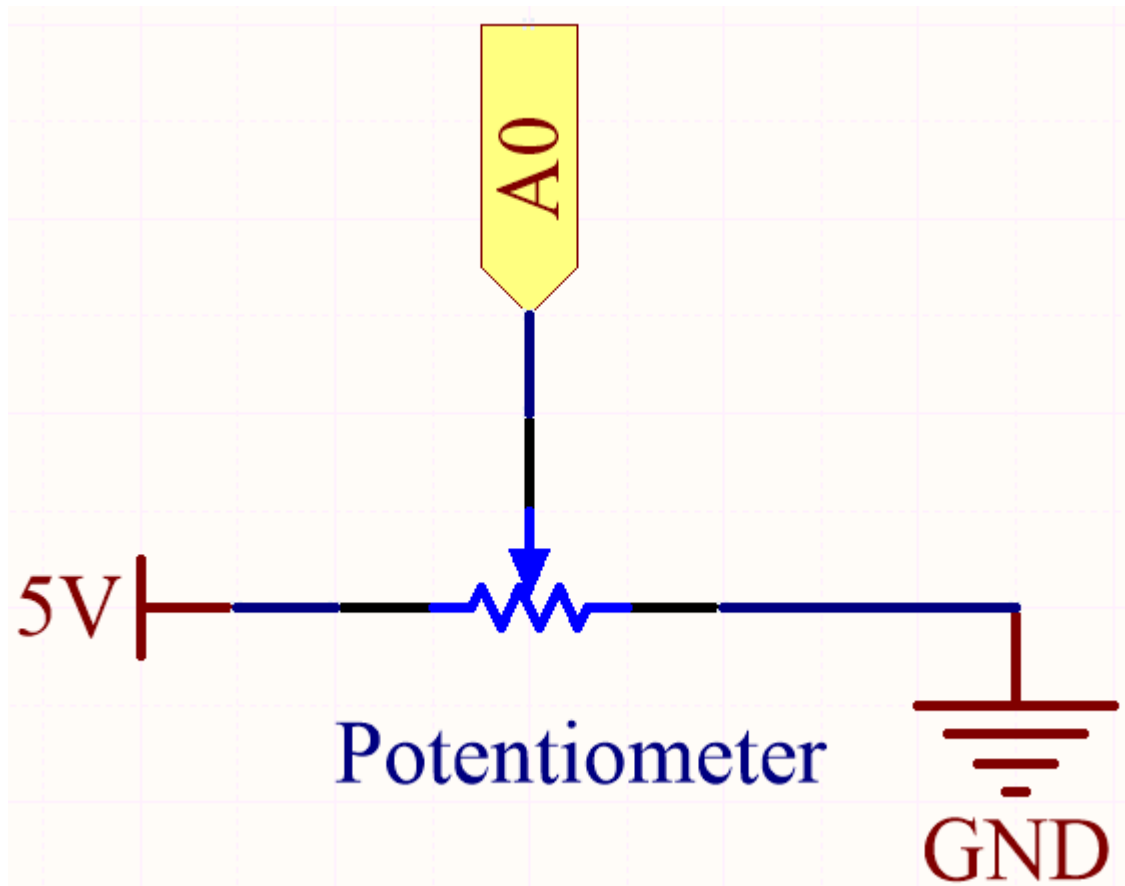
一式を購入すると確かに便利です。以下がリンクです：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから個別に購入することもできます。

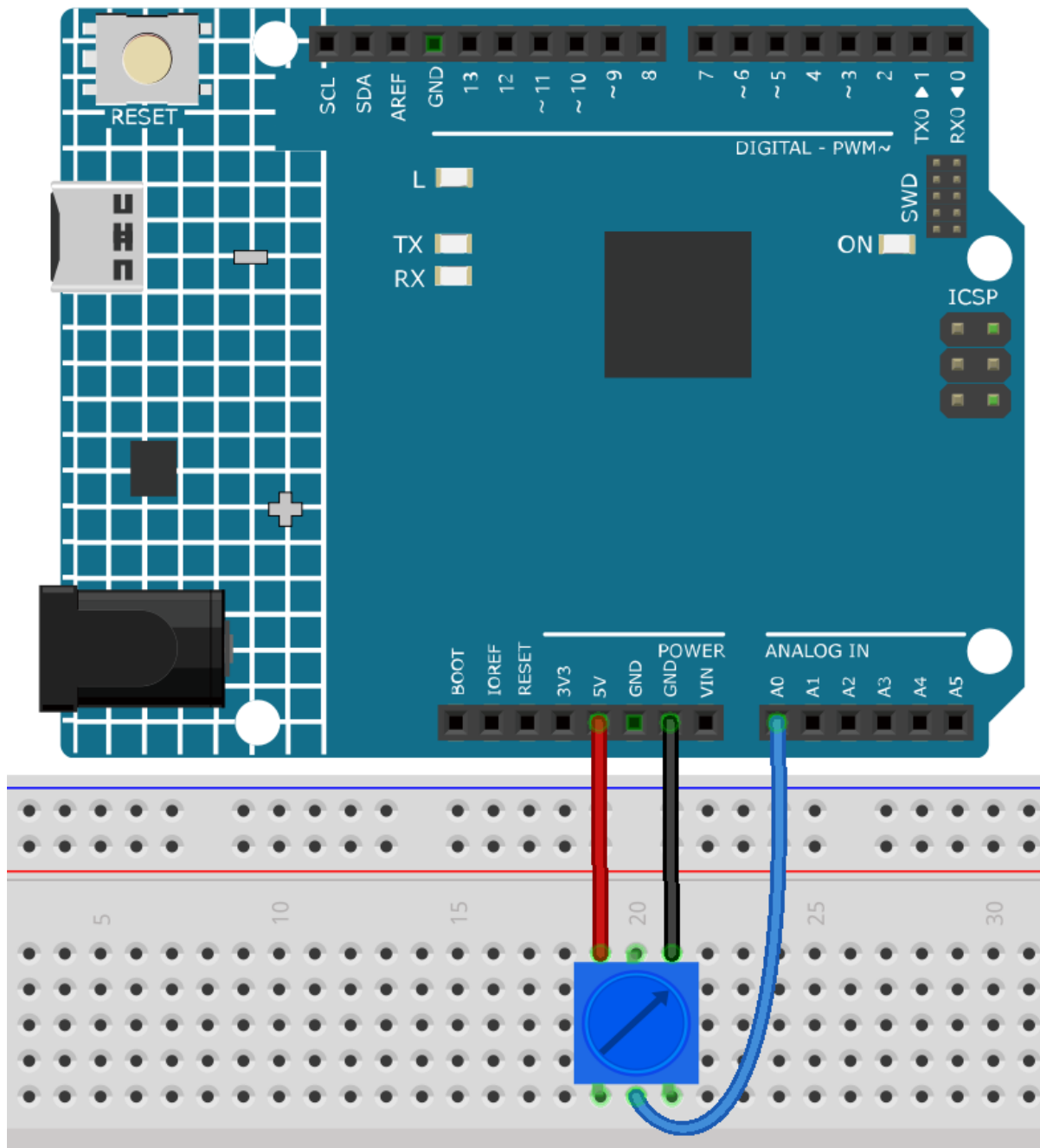
コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
ポテンショメーター	

回路図



この例では、アナログピン (A0) を使用して、ポテンショメータの値を読み取ります。ポテンショメータの軸を回転させることで、これらの 3 つのピンの間での抵抗の分布を変更し、中央のピンの電圧を変更できます。中央と 5V に接続された外部ピンとの間の抵抗がほぼゼロに近く (中央ともう一方の外部ピンとの間の抵抗がほぼ 10k に近い場合)、中央のピンの電圧は 5V に近くなります。逆の操作 (中央と 5V に接続された外部ピンとの間の抵抗がほぼ 10k に近い) は、中央のピンの電圧が 0V に近くなるでしょう。

配線図



コード

注釈:

- 3in1-kit\learning_project\4.1.turn_the_knob のパスの下にある 4.1.turn_the_knob.ino ファイルを開くことができます。
- または、このコードを **Arduino IDE** にコピーします。

ボードにコードをアップロードすると、シリアルモニタを開いてピンの読み取り値を確認できます。ポテンショメータの軸を回転させると、シリアルモニタは「0」～「1023」の値を表示します。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.4.2 4.2 光を感じる

光抵抗はアナログ入力のための代表的なデバイスで、ポテンショメータと非常に似た方法で使用されます。その抵抗値は光の強度に依存しており、照射される光が強いほど抵抗値は小さくなりますが、その逆の場合、抵抗値は増加します。

必要な部品

このプロジェクトでは、以下の部品が必要です。

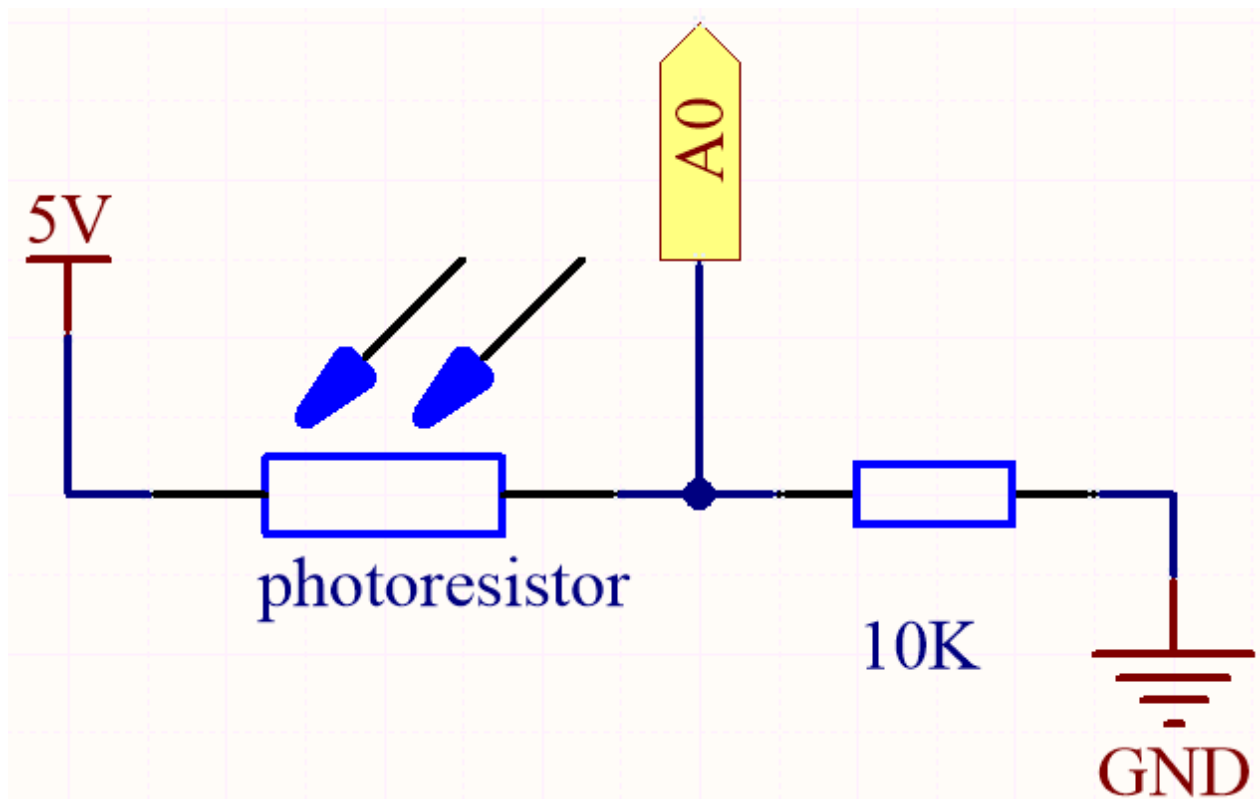
一式を購入すると確かに便利です。以下がリンクです：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
Arduino Uno R4 Minima	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
フォトレジスタ	

回路図

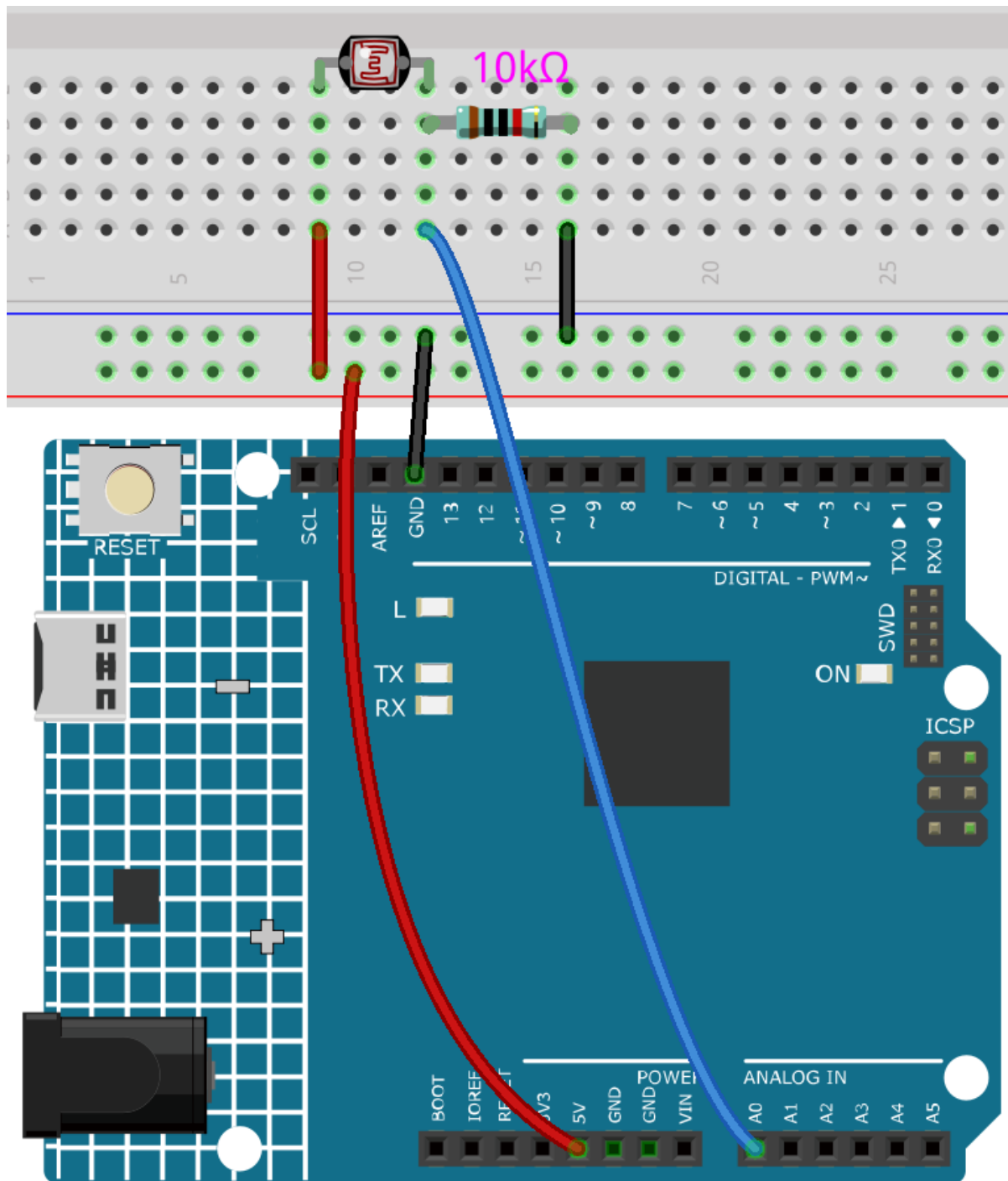


この回路では、10K の抵抗と光抵抗が直列に接続されており、両方を通る電流は同じです。10K の抵抗は保護として機能し、ピン A0 は光抵抗の電圧変換後の値を読み取ります。

光が強くなると、フォト・レジスタの抵抗が減少し、電圧が下がるので、ピン A0 からの値は増加する；光が十分に強ければ、フォト抵抗の抵抗値は 0 に近くなり、A0 ピンの値は 1023 に近くなる。この時、10K 抵抗が保護的な役割を果たし、5V と GND が接続され、短絡にならないようにしている。

光抵抗を暗い状況に置くと、ピン A0 の値は減少します。十分に暗い状況では、光抵抗の抵抗は無限大となり、その電圧は 5V (10K 抵抗は無視できる) に近く、ピン A0 の値は 0 に近くなります。

配線図



コード

注釈:

- 3in1-kit\learning_project\4.2.feel_the_light のパス下の 4.2.feel_the_light.ino ファイル

を開きます。

- または、このコードを **Arduino IDE** にコピーします。
-

コードが正常にアップロードされると、シリアルモニタは光抵抗の値を表示します。現在の周囲の明るさが強いほど、シリアルモニタに表示される値は大きくなります。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.4.3 4.3 ジョイスティックの操作

ジョイスティックは、定期的にビデオゲームを楽しむ人々にとっては非常に馴染み深いものです。通常、キャラクターの移動や画面の回転に使用されます。

我々の動きは、ジョイスティックによって読み取ることができ、これは非常に単純な原理で動作します。それは、互いに垂直な二つのポテンショメーターから成っています。これらのポテンショメーターはジョイスティックの垂直と水平の両方の方向でのアナログ値を測定し、平面直角座標系での値 (x,y) を生成します。

このキットには、デジタル入力を持つジョイスティックも含まれています。ジョイスティックが押されたときに作動します。

必要な部品

このプロジェクトでは、以下の部品が必要です。

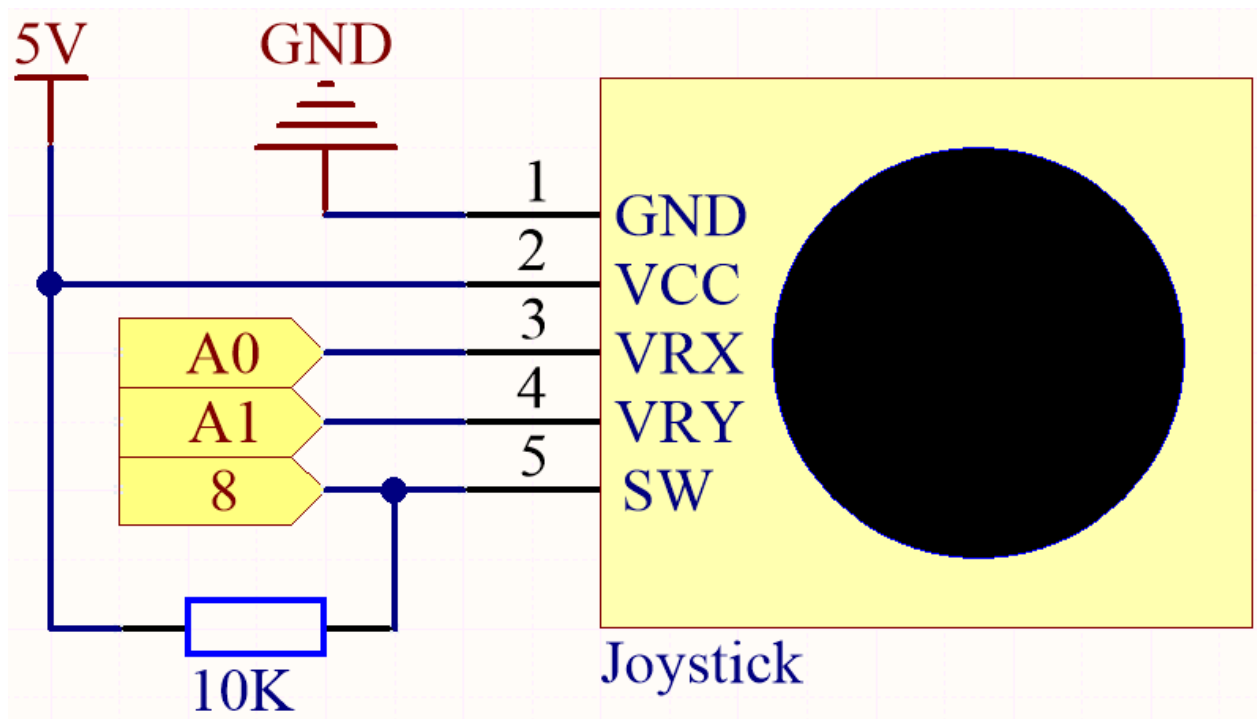
一式を購入するのは確かに便利です。リンクは以下のとおりです：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから別々に購入することも可能です。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
ジョイスティックモジュール	-

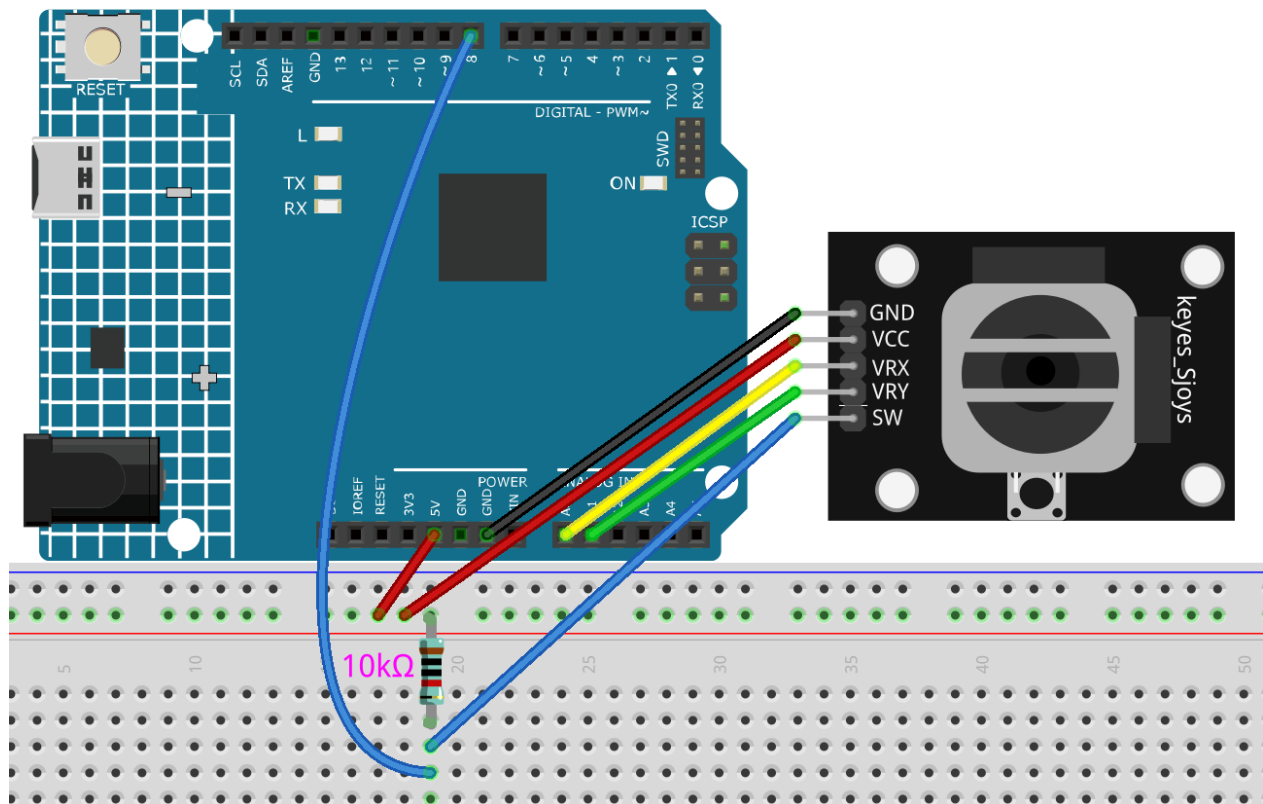
回路図



注釈: SW ピンは 10K のプルアップ抵抗に接続されています。これは、ジョイスティックが押されていないときに SW ピン (Z 軸) で安定した高レベルを得るためです。それ以外の場合、SW はサスペンド状態となり、出力値は

0/1 の間で変動する可能性があります。

配線図



コード

注釈:

- 3in1-kit\learning_project\4.3.toggle_the_joystick のパスの下に 4.3.toggle_the_joystick.ino ファイルを開きます。
- あるいは、このコードを **Arduino IDE** にコピーします。

コードが正常にアップロードされた後、シリアルモニターを開くと、ジョイスティックの x,y,z の値が表示されます。

- x 軸と y 軸の値は 0 から 1023 までのアナログ値です。
- Z 軸は 1 または 0 の状態のデジタル値です (押された場合は 0)。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.4.4 4.4 土壌の湿度を測定する

農業の世界では、作物自体が土壌の無機元素を直接取得することはできません。土壌中の水が、これらの無機元素を溶解する溶媒として作用します。

作物は根系を通じて土壌の湿度を吸収し、栄養を得て、成長を促進します。

作物の成長と発展の過程で、土壌の温度に対する要求も異なります。そのため、土壌湿度センサーが必要とされます。

必要な部品

このプロジェクトでは、以下の部品が必要です。

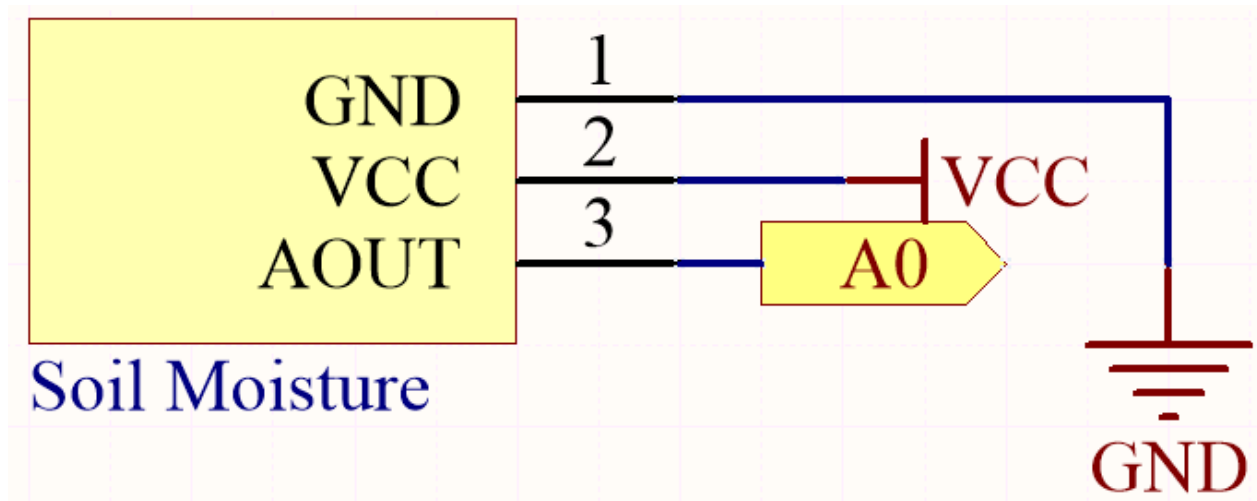
一式を購入するのは確かに便利です。リンクは以下のとおりです：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

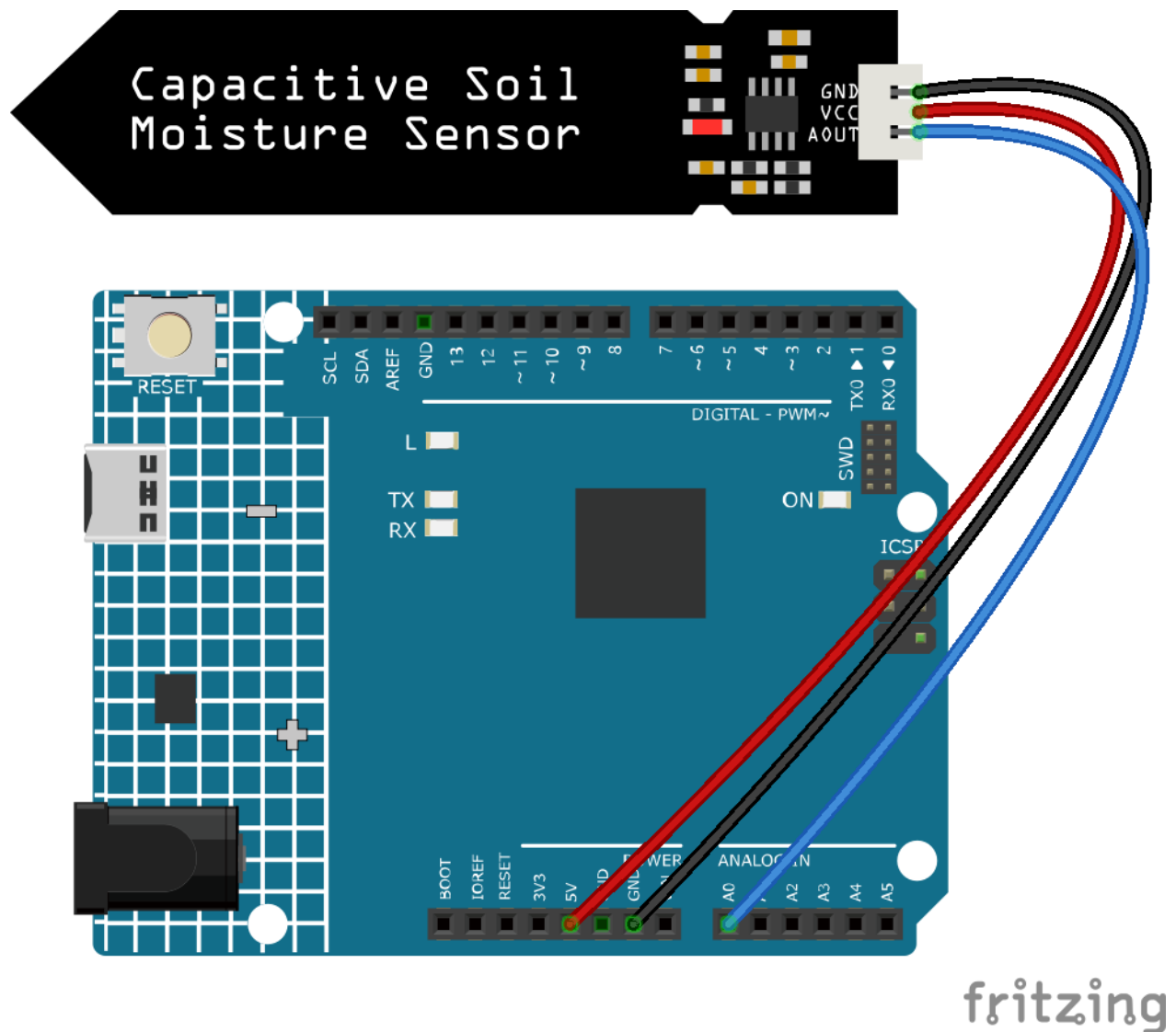
以下のリンクから別々に購入することも可能です。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ジャンパーワイヤー	
土壌湿度モジュール	

回路図



配線図



コード

注釈:

- 3in1-kit\learning_project\4.4.measure_soil_moisture のパスの下に 4.4.measure_soil_moisture.ino ファイルを開きます。
 - あるいは、このコードを **Arduino IDE** にコピーします。
-

コードが正常にアップロードされると、シリアルモニターに土壌の湿度値が表示されます。

モジュールを土壌に差し込み、水をやると、土壌湿度センサーの値は小さくなります。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.4.5 温度計

温度計は温度または温度勾配（物体の熱さまたは冷たさの度合い）を測定する装置です。温度計には2つの重要な要素があります：(1) 温度センサ（例えば、水銀ガラス温度計の球根や赤外線温度計の焼結センサ）で、これには温度の変化とともに何らかの変化が生じます；そして(2) この変化を数値として変換する手段（例えば、水銀ガラス温度計に表示されている目盛りや赤外線モデルのデジタル表示）。温度計は、工業や産業のプロセスを監視するため、気象学、医学、科学研究で広く使用されています。

サーミスターは、その抵抗が温度に強く依存するタイプの温度センサであり、2つのタイプがあります：負温度係数 (NTC) と正温度係数 (PTC)、これは NTC と PTC としても知られています。PTC サーミスターの抵抗は温度とともに増加し、NTC の条件は前者とは逆です。

この実験では、NTC サーミスター を使用して温度計を作成します。

SunFounder 3in1 Kit

必要な部品

このプロジェクトでは、以下の部品が必要です。

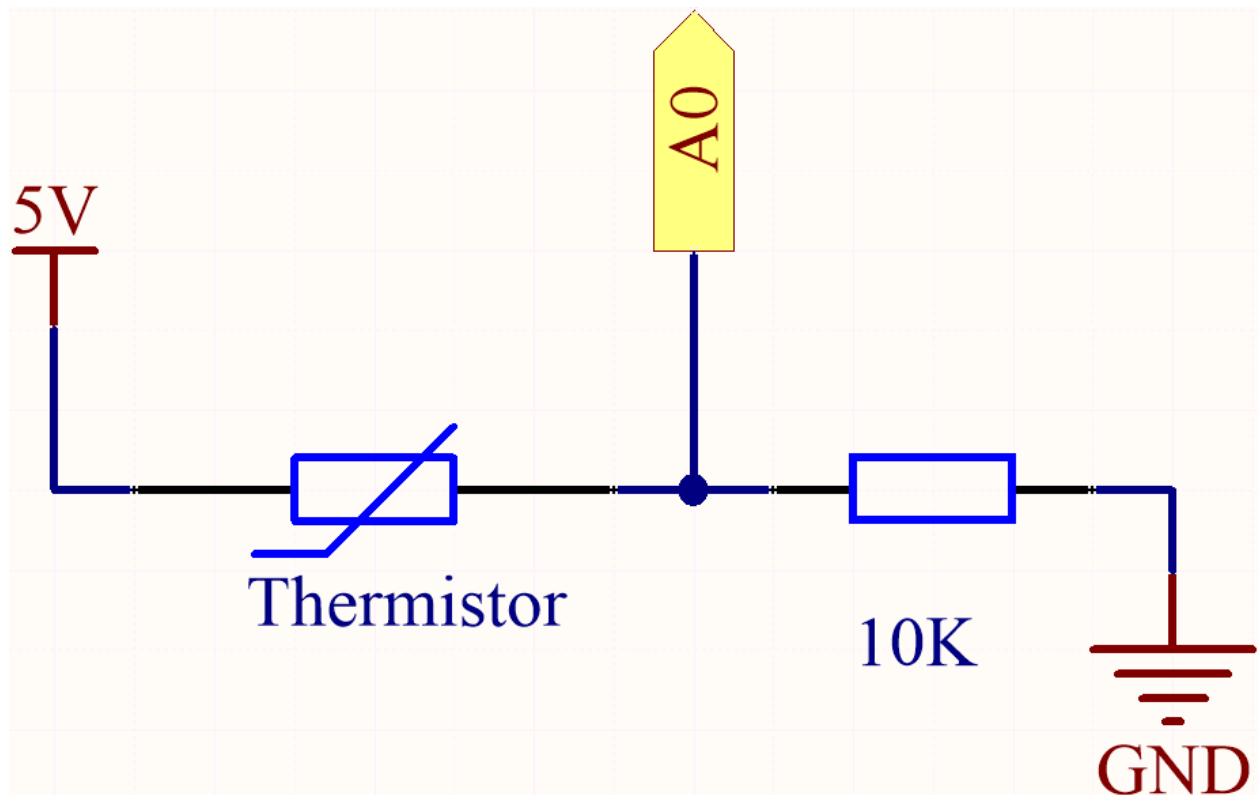
一式を購入するのは確かに便利です。リンクは以下のとおりです：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから別々に購入することも可能です。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
サーミスタ	

回路図



各サーミスターには通常の抵抗があります。ここでは、それは 10k オームで、これは 25 度セルシウスで測定されます。

温度が上昇すると、サーミスターの抵抗は減少します。その後、A/D アダプタによって電圧データがデジタル量に変換されます。

セルシウス度または華氏度の温度はプログラムによって出力されます。

抵抗と温度の関係は以下の通りです：

$$RT = RN \exp B(1/TK - 1/TN)$$

- **RT** は、温度が **TK** のときの NTC サーミスターの抵抗です。
- **RN** は、定格温度 **TN** での NTC サーミスターの抵抗です。ここで、**RN** の数値は 10k です。
- **TK** はケルビン温度で、単位は K です。ここでは、**TK** の数値は 273.15 + セルシウス度 です。
- **TN** は定格ケルビン温度で、単位も K です。ここでは、**TN** の数値は 273.15+25 です。
- **B**(ベータ)、NTC サーミスターの材料定数は、数値が 3950 の熱感受性指数としても知られています。
- **exp** は指数関数の略で、基数 **e** は自然数で、約 2.7 と等しい。

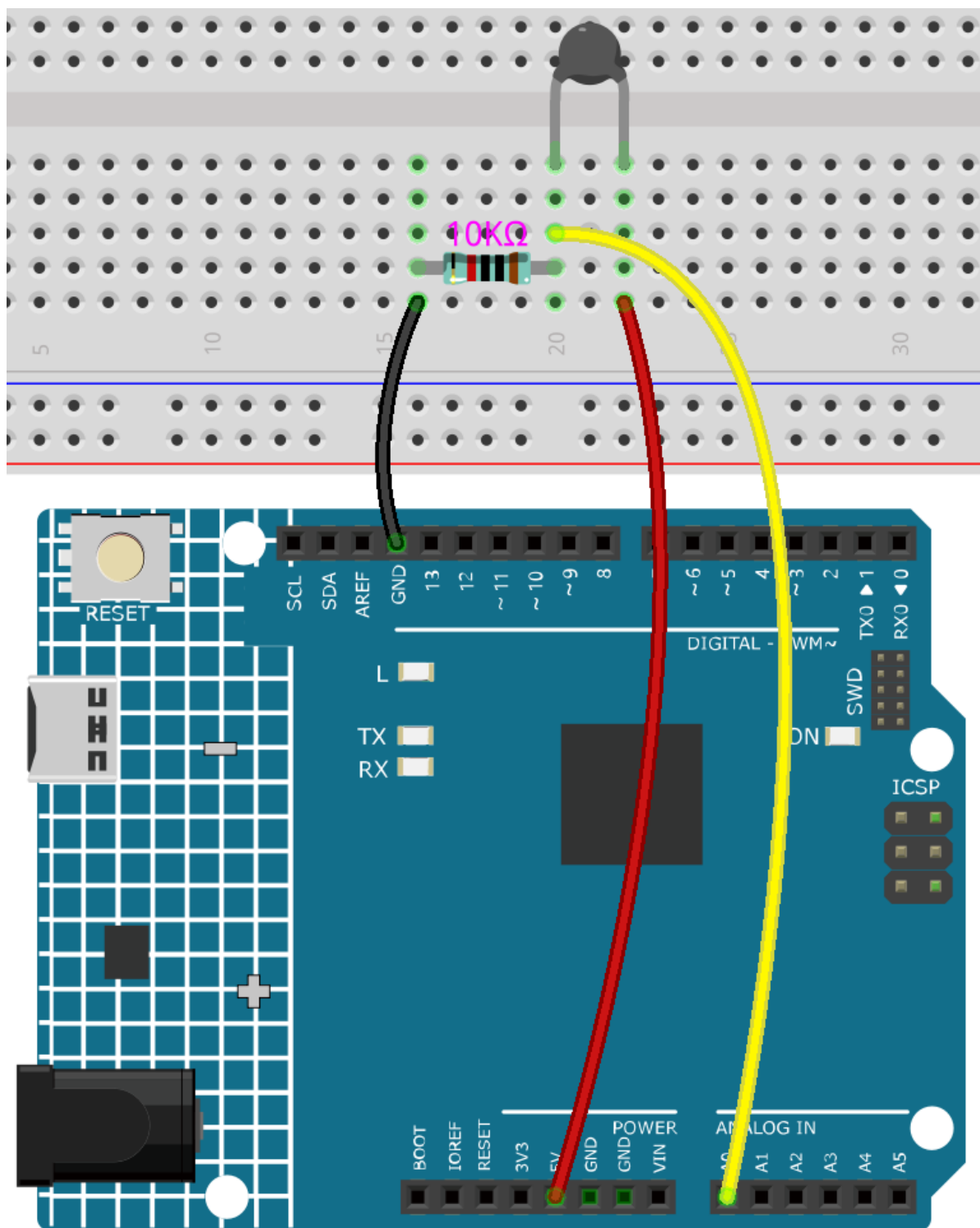
この式 $TK = 1 / ((\ln(RT/RN) / B) + 1/TN)$ を変換して、ケルビン温度から 273.15 を引くとセルシウス度となります。

この関係は経験的な式です。温度と抵抗が有効範囲内にある場合にのみ正確です。

配線図

注釈:

- サーミスターは黒色または緑色で、103 とマークされています。
-



コード

注釈:

- ファイル `4.5_thermometer.ino` を `euler-kit/arduino/4.5_thermometer` のパスの下で開いてください。
- または、このコードを **Arduino IDE** にコピーしてください。

アップロードボタンをクリックする前に、Raspberry Pi Pico ボードと正しいポートを選択してください。

コードが正常にアップロードされると、シリアルモニターはセルシウス度と華氏度の温度を表示します。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なレビュー：新製品の発表や先行レビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.5 5. さらなる構文

この章では、プログラムが現実の世界とどのように交流するかの基本ロジックを示す例をいくつか紹介します。これにより、Arduino のプログラミングに慣れるのを助けます。あなたが創造的なアイデアを持っていれば、プログラミングはもはや難しくありません。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
 - 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
 - 独占的なレビュー：新製品の発表や先行レビューに早期アクセスしましょう。
 - 特別割引：最新製品の独占割引をお楽しみください。
-

- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.5.1 5.1 If else

通常、私たちは条件判断を使用して、最も基本的な現実との対話プロジェクトを完了させます。ここでは、リードスイッチとLEDを使用してこのロジックを示すドア検出システムを構築します。

ドアの一方にマグネットを取り付け、もう一方のドア側にリードスイッチ（回路付き）を取り付けます。ドアが閉じている場合、マグネットはリードスイッチに近く、これによりスイッチがオンになります。

必要な部品

このプロジェクトでは、以下のコンポーネントが必要です。

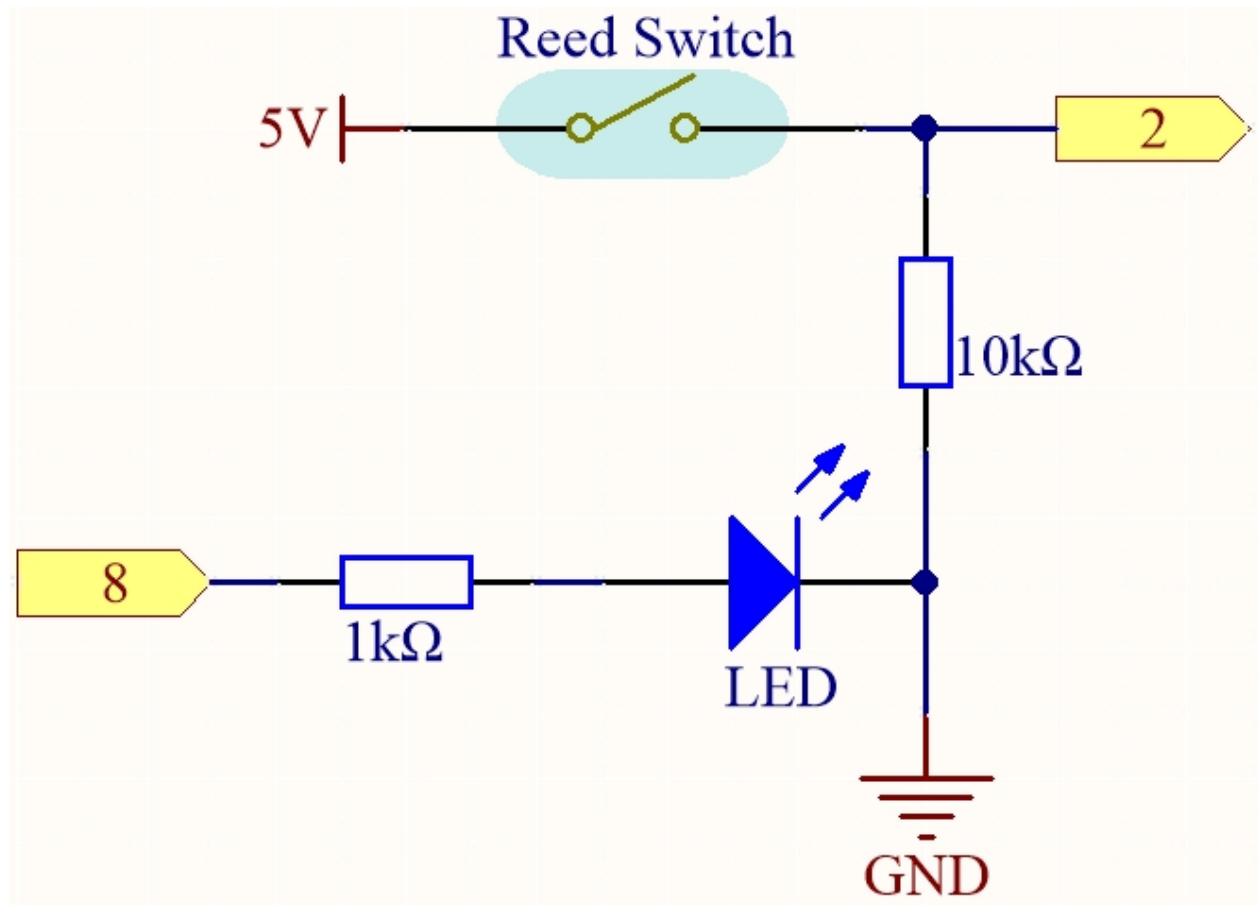
全体のキットを購入すると便利です、リンクはこちら：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

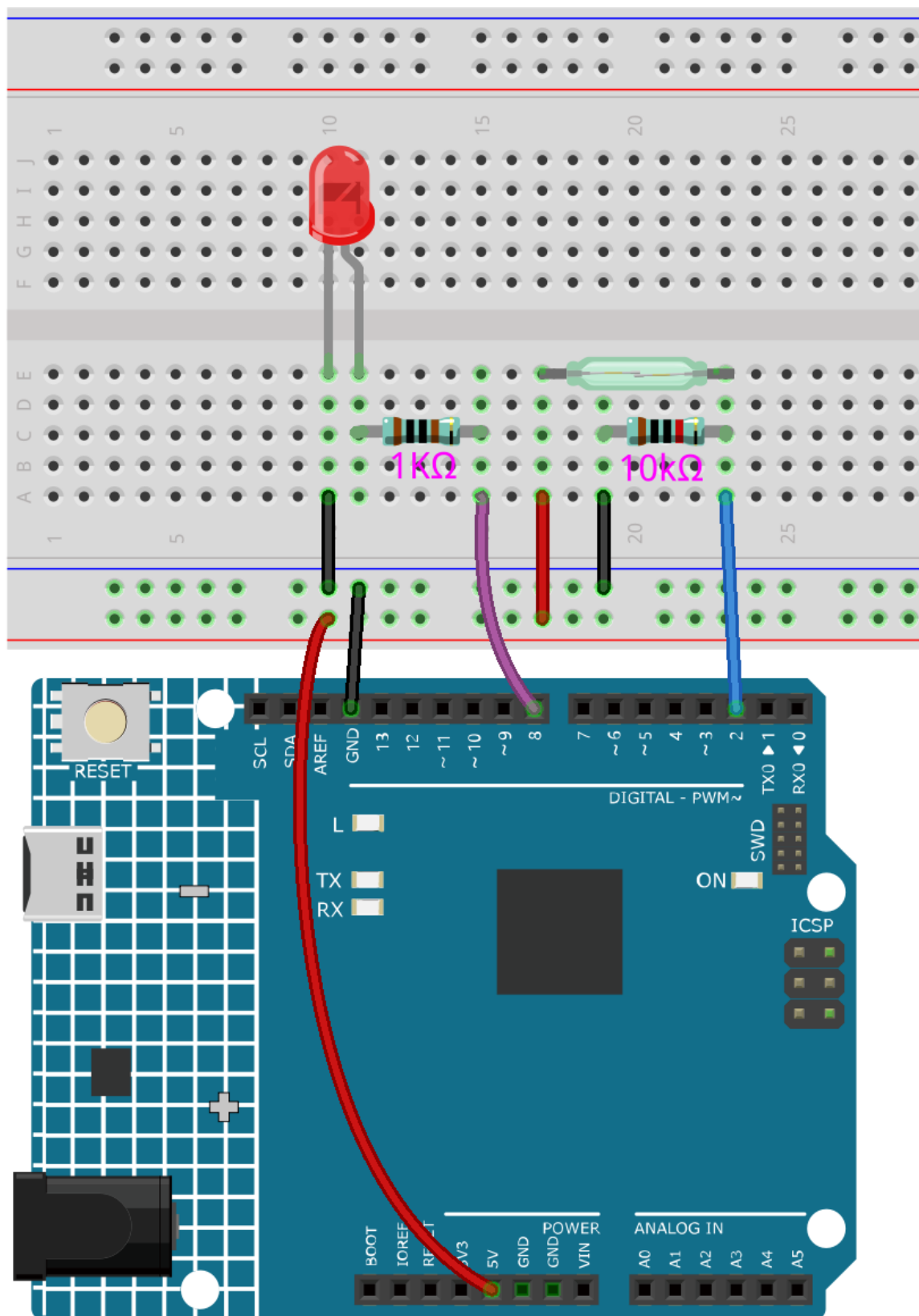
以下のリンクから別々に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
<i>LED</i>	
リードスイッチ	-

回路図



配線図



コード

注釈:

- 3in1-kit\learning_project\5.1.if_else のパスの下で 5.1.if_else.ino ファイルを開きます。
 - または、このコードを **Arduino IDE** にコピーします。
-

コードが正常にアップロードされた後、ドアを閉めない場合、LED が点灯して、ドアを閉めるように促します。

ちなみに、ドアが閉じているときに LED を点灯させる反対の効果が必要な場合、if の条件を変更するだけで済みます。

- if else

if else は、基本的な if 文よりもコードの流れをより制御することができ、複数のテストをグループ化することができます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.5.2 5.2 閾値

多くのプロジェクトで、以下のようなニーズに遭遇するでしょう。「xxx があるレベルに達したら、その後...」

例えば、スマートホームでは、光の強度が 50Lux 以下の場合、ライトを点灯する。もう一つの例として、コンピュータのマザーボードでは、CPU の動作温度が 65 度以上になるとファンを起動する、など。

これらの要件では、「閾値」というキーワードが反映されています。

閾値の値を調整することで、回路の動作を個々のニーズに合わせて調整することができます。例えば、私がもっと明るい生活環境を好む場合、スマートホームの自動照明の閾値を 80Lux に上げることができます。また、私のス

タジオの換気環境があまり良くなく、放熱要求が高い場合、自動ファン開始の閾値を 50 度に調整することができます。

ここでは、土の湿度センサーと 2 つの LED を使用して、鉢のモニターを作成します。土が乾燥していると赤い LED が点灯し、土が十分に湿っていると緑の LED が点灯します。土の乾燥と湿潤を判定する閾値は手動で調整する必要があります。

必要な部品

このプロジェクトには、以下のコンポーネントが必要です。

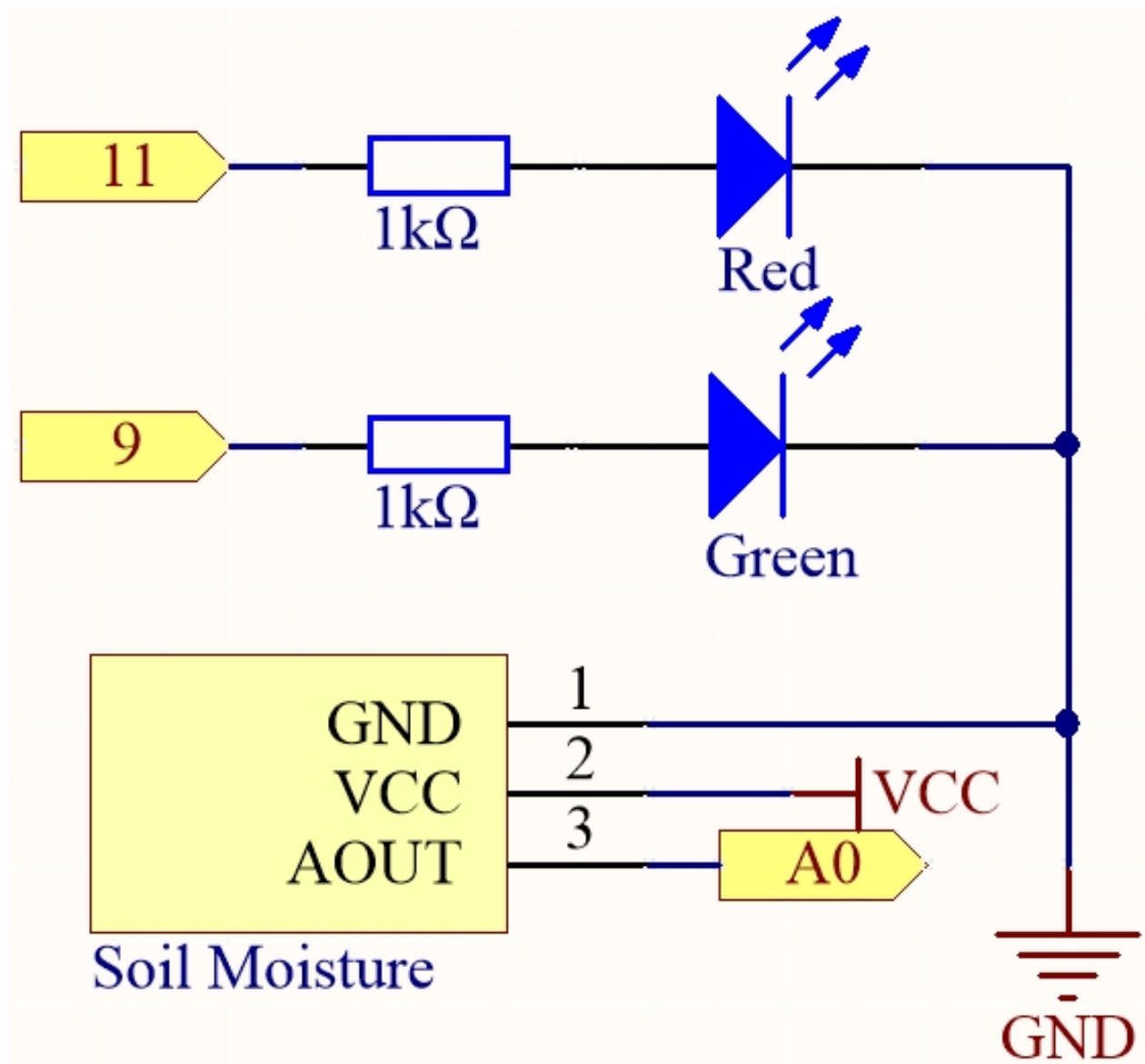
一式を購入するのが便利です、リンクはこちら：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

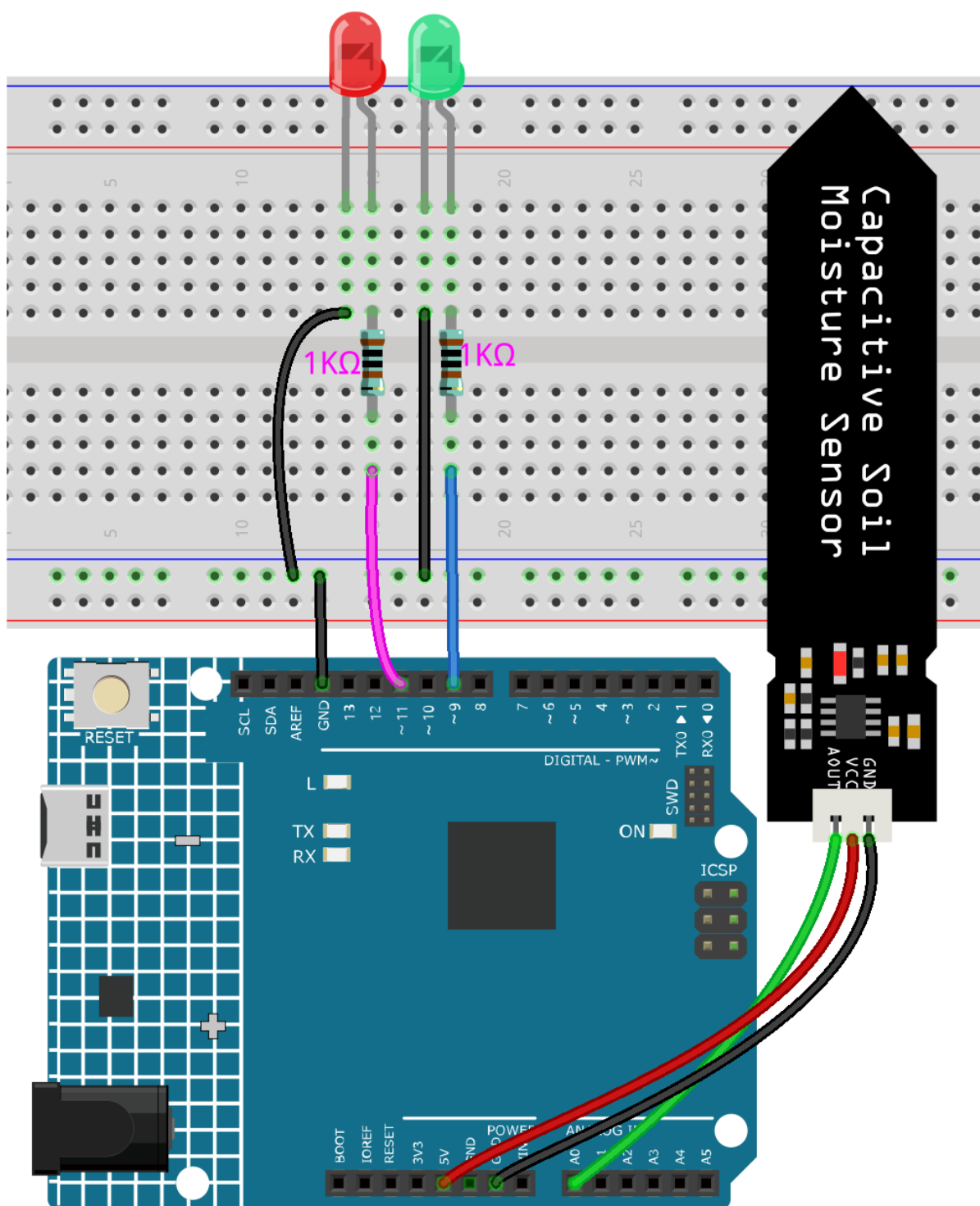
以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
<i>LED</i>	
土壌湿度モジュール	

回路図



配線図



コード

注釈:

- 3in1-kit\learning_project\5.2.threshold のパスの下で 5.2.threshold.ino ファイルを開きます。

- または、このコードを **Arduino IDE** にコピーします。
-

コードが正常にアップロードされた後、閾値が正しく設定されていれば、土が乾燥していると赤い LED が点灯し、水をやる必要があることを知らせてくれます。水をやった後、緑の LED が点灯します。

動作原理

```
...  
  
void loop() {  
    int sensorValue = analogRead(soilMoisture);  
    Serial.println(sensorValue);  
    if (sensorValue > threshold) {  
        digitalWrite(redPin, HIGH); // 赤い LED を点灯  
        digitalWrite(greenPin, LOW); // 緑  
    } else {  
        digitalWrite(greenPin, HIGH); // 緑の LED を点灯  
        digitalWrite(redPin, LOW); // 赤  
    }  
}  
...
```

まず、threshold の値を設定し、次に土の湿度モジュールの値を読み取ります。湿度レベルが上がると、その値は減少します。現在読み取った値が設定した threshold よりも大きい場合、赤い LED を点灯させます。それ以外の場合は、緑の LED を点灯させます。

この threshold の値は、実際の状況に応じて調整する必要があります。まずコードをアップロードして、シリアルモニターを開いて値を確認します。湿潤と乾燥の状態での値を記録し、その中間の値を threshold の値として選びます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.5.3 5.3 状態変化の検出

ボタンが他のデバイスを制御する場合、それは押されているときにのみ動作するだけでなく、リリースされたときに停止することもできます。ボタンが押されるたびに動作状態を切り替えることも可能です。

この効果を達成するために、ボタンが押されたときに動作状態をオフとオンの間で切り替える方法を知る必要があります。それは「状態変化の検出」です。

このプロジェクトでは、モーターを制御するためのボタンを使用します。

必要な部品

このプロジェクトには、以下の部品が必要です。

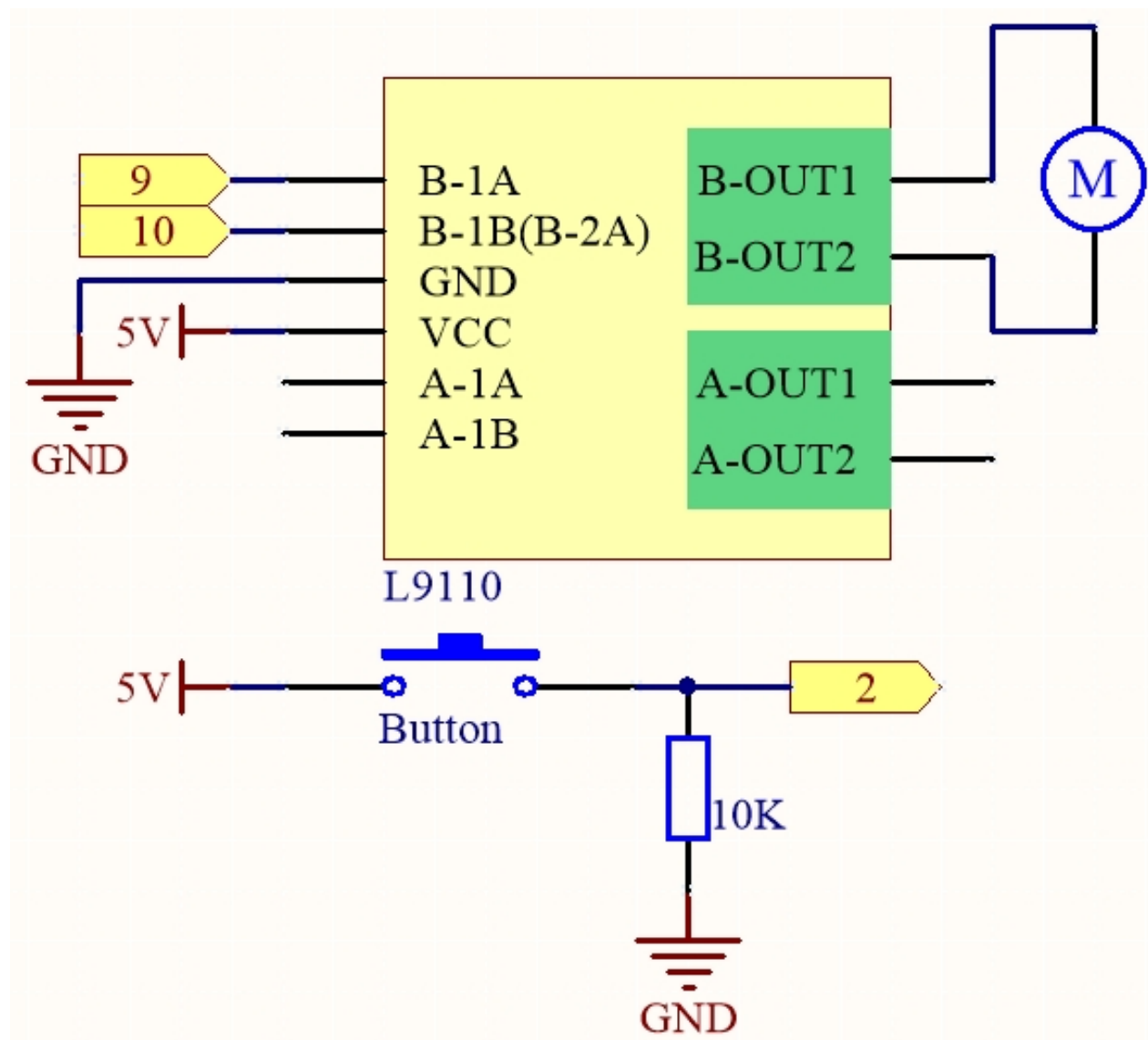
全体のキットを購入すると非常に便利です。こちらがリンクです：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

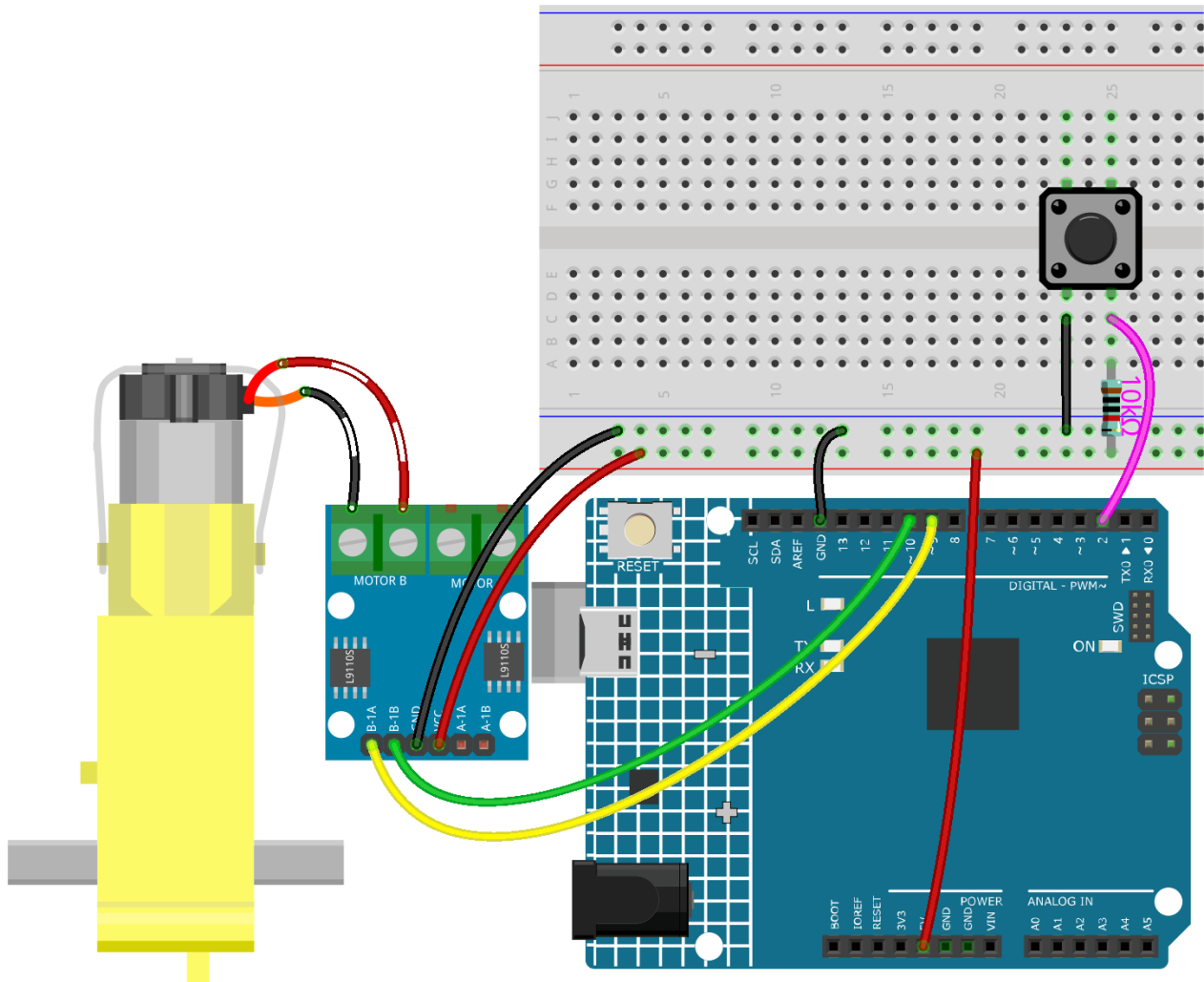
下記のリンクからも個別に購入することができます。

コンポーネントの紹介	購入リンク
Arduino Uno R4 Minima	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
ボタン	
TT モーター	-
L9110 モータードライバーモジュール	-

回路図



配線図



コード

注釈:

- 3in1-kit\learning_project\5.3.state_change_detection のパスの下で 5.3.state_change_detection.ino ファイルを開きます。
- または、このコードを **Arduino IDE** にコピーします。

コードが正常にアップロードされた後、ボタンを押すとモーターが動き、再びボタンを押すとモーターが停止します。

どのように動作するのか？

1. モーターとボタンのための変数を作成し、ピンを定義します。

```
...
int detectionState = 0;
int buttonState = 0;
int lastButtonState = 0;
```

- detectionState はボタンが押されるたびに値が変わるフラグです。例：この時は 0、次は 1、と交互に変わります。
- buttonState と lastButtonState は、この回と前回のボタンの状態を記録し、ボタンが押されたか、放されたかを比較するために使われます。

2. 各ピンを初期化し、シリアルモニターのボーレートを設定します。

```
void setup() {
  pinMode(buttonPin, INPUT);
  Serial.begin(9600);
  pinMode(A_1A, OUTPUT);
  pinMode(A_1B, OUTPUT);
}
```

3. まず、ボタンの状態を読み取ります。ボタンが押された場合、変数 detectionState の値は 0 から 1、または 1 から 0 に切り替わります。detectionState が 1 の場合、モーターが回転します。ボタンが押されるたびに、モーターが交互に動作し、停止する効果があります。

```
void loop() {
  // ボタンが押されるたびに detectionState を切り替える
  buttonState = digitalRead(buttonPin);
  if (buttonState != lastButtonState) {
    if (buttonState == HIGH) {
      detectionState=(detectionState+1)%2;
      Serial.print("The detection state is: ");
      Serial.println(detectionState);
    }
    delay(50);
  }
  lastButtonState = buttonState;

  // detectionState に従って、モーターを起動する
  if(detectionState==1){
    digitalWrite(A_1A,HIGH);
    digitalWrite(A_1B,LOW);
  }
}
```

(次のページに続く)

(前のページからの続き)

```

    }else{
        digitalWrite(A_1A,LOW);
        digitalWrite(A_1B,LOW);
    }
}

```

すべてのワークフローは以下の通りです。

- ボタンの値を読み取る。

```
buttonState = digitalRead(buttonPin);
```

- buttonState と lastButtonState が等しくない場合、ボタンの状態が変わったことを意味します。次の判断を続け、この時のボタンの状態を変数 lastButtonState に格納します。delay(50) はジッタを排除するために使用されます。

```

if (buttonState != lastButtonState) {
    ...
    delay(50);
}
lastButtonState = buttonState;

```

- ボタンが押されたとき、その値は HIGH です。ここでは、ボタンが押されたときに変数 detectionState の値が変更されます。例：操作後に 0 から 1 へ。

```

if (buttonState == HIGH) {
    detectionState=(detectionState+1)%2;
    Serial.print("The detection state is: ");
    Serial.println(detectionState);
}

```

- 変数 detectionState が 1 の場合、モーターを回転させます。それ以外の場合は停止します。

```

if(detectionState==1){
    digitalWrite(A_1A,HIGH);
    digitalWrite(A_1B,LOW);
}else{
    digitalWrite(A_1A,LOW);
    digitalWrite(A_1B,LOW);
}

```

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なレビュー：新製品の発表や先行レビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.5.4 5.4 インターバル

時には、二つのことを同時に行いたい場合があります。例えば、LED を点滅させながらボタンの押下を読み取りたい場合などです。この場合、`delay()` は使用できません。なぜなら、Arduino は `delay()` の間プログラムを一時停止させるからです。もしボタンが Arduino が `delay()` の待ち時間中に押された場合、プログラムはボタンの押下を検出できません。

この状況を例えるなら、電子レンジでピザを温めながら、重要なメールを待つシチュエーションです。ピザを電子レンジに入れ、10 分設定します。`delay()` の使用を例えると、電子レンジの前でタイマーが 0 になるのを見つめることになります。この間に重要なメールが届いた場合、それを見逃してしまいます。

実際には、ピザのタイマーをセットし、メールを確認し、他の短い作業を行い、時折電子レンジのタイマーをチェックすることでしょう。

このスケッチでは、`delay()` を使わずにブザーを鳴らす方法を示しています。ブザーをオンにして、現在の時間を記録します。そして、`loop()` のたびに、指定されたインターバル時間が経過したかどうかをチェックします。経過していれば、ブザーを鳴らし、新しい時間を記録します。この方法で、ブザーは連続して鳴らされ、スケッチの実行は一つの命令で遅れることはありません。

この条件に基づき、ブザーが音楽を奏でるのを邪魔されることなく、LED を制御するボタンのコードを追加することができます。

必要な部品

このプロジェクトでは、以下のコンポーネントが必要です。

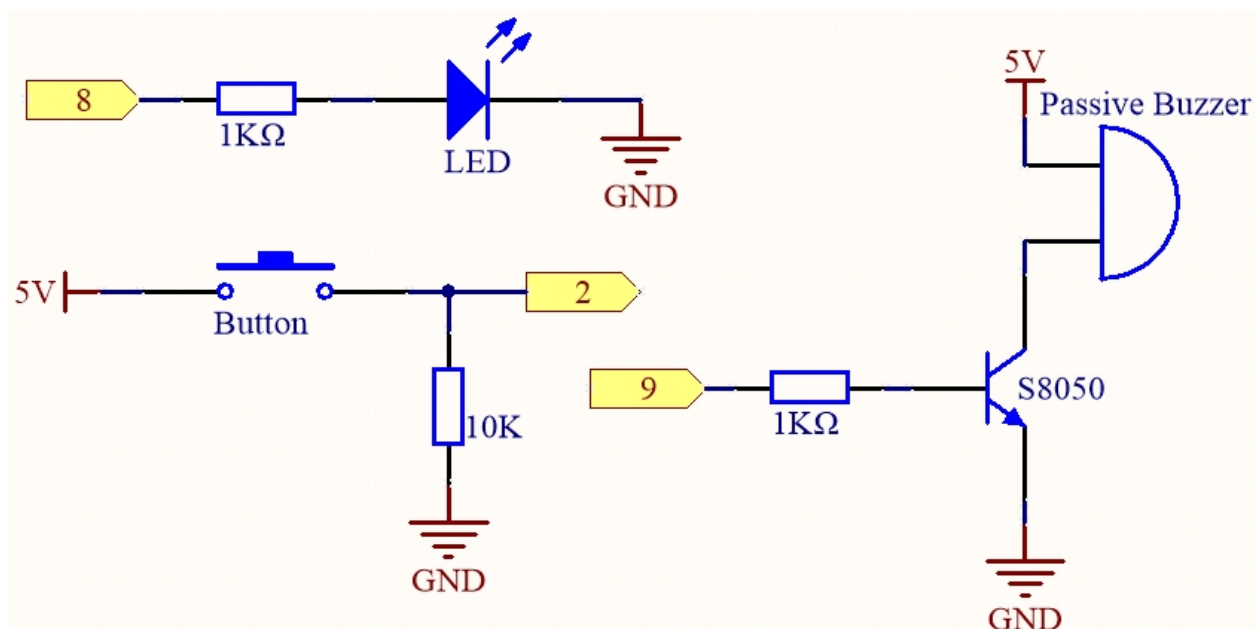
一式を購入するのが便利です。購入リンクはこちら：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

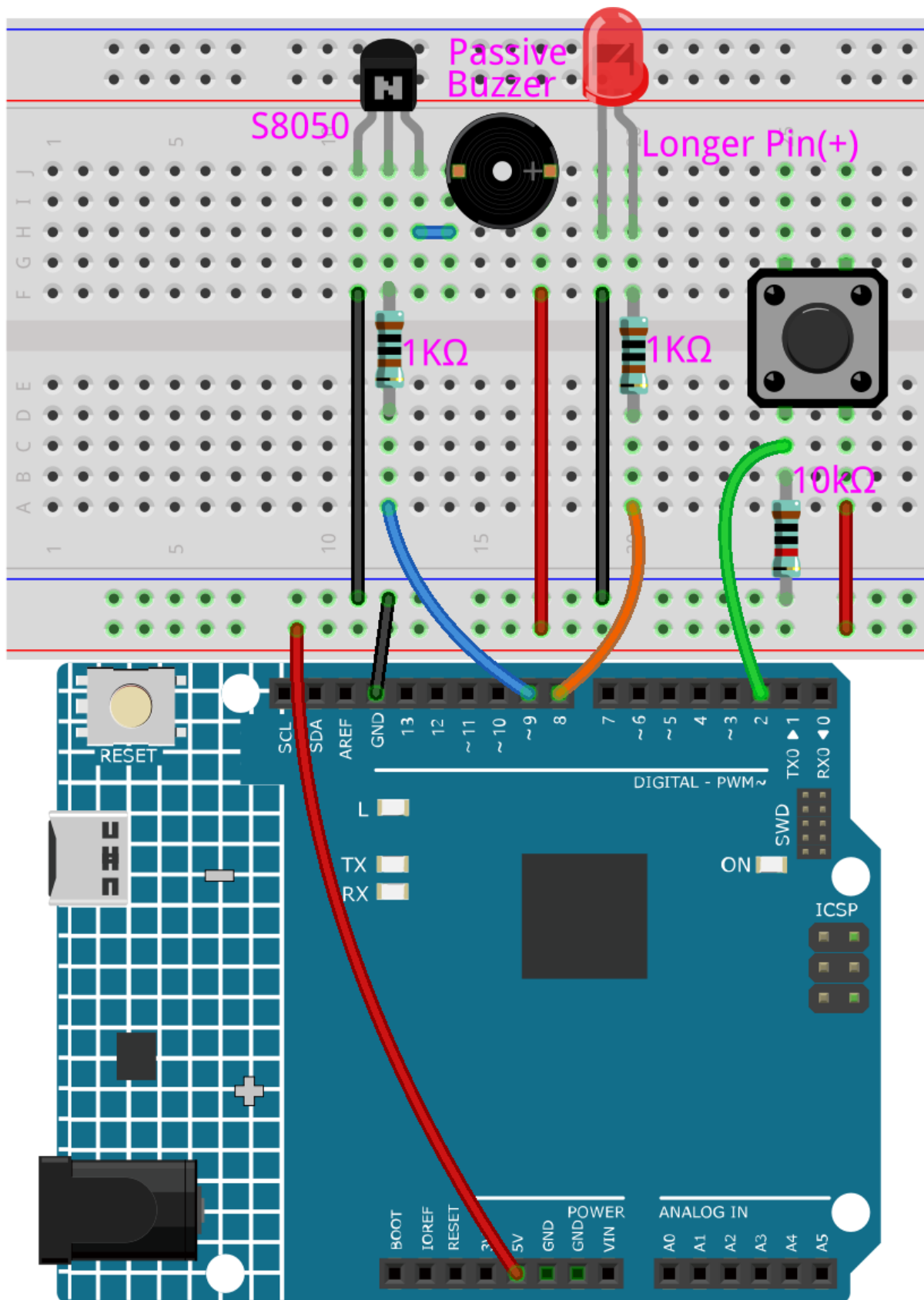
以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
LED	
ボタン	
ブザー	

回路図



配線図



注釈:

- 3in1-kit\learning_project\5.4.interval のパスの下の 5.4.interval.ino ファイルを開きます。
- または、このコードを **Arduino IDE** にコピーします。

コードが正常にアップロードされると、ブザーが音楽を再生します。ボタンを押すたびに、LED が点灯します。LED とブザーの動作は互いに干渉しません。

どのように動作するのか？

マイコンの前の動作時間を格納するための変数 `previousMillis` を初期化します。

```
unsigned long previousMillis = 0;
```

どの音符が再生されているかをマークします。

```
int thisNote=0;
```

各音符の間隔時間。

```
long interval = 1000;
```

`loop()` 内で、現在の時間を格納するための `currentMillis` を宣言します。

```
unsigned long currentMillis = millis();
```

現在の動作時間と最後の更新時間の間隔が 1000ms より大きい場合、特定の関数がトリガされます。その後、次のトリガが 1 秒後に発生するため、`previousMillis` を現在の時間に更新します。

```
if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis; // save the last time of the last tone
    //...
}
```

メロディの音符を順番に再生します。

```
tone(buzzerPin, melody[thisNote], 100);
interval=1000/noteDurations[thisNote]; // interval at which to tone
thisNote=(thisNote+1)%(sizeof(melody)/2); //iterate over the notes of the melody
```

ボタンは LED を制御します。

```
// play button & led
digitalWrite(ledPin,digitalRead(buttonPin));
```

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.5.5 5.5 内蔵ライブラリの使用

Arduino IDE では、対応する .h ファイルを直接コードに追加することで、多くの組み込みライブラリを使用することができます。

このプロジェクトでは Servo ライブラリを使用してサーボを駆動し、0° から 180° の間で回転させることができます。

必要な部品

このプロジェクトでは、以下の部品が必要です。

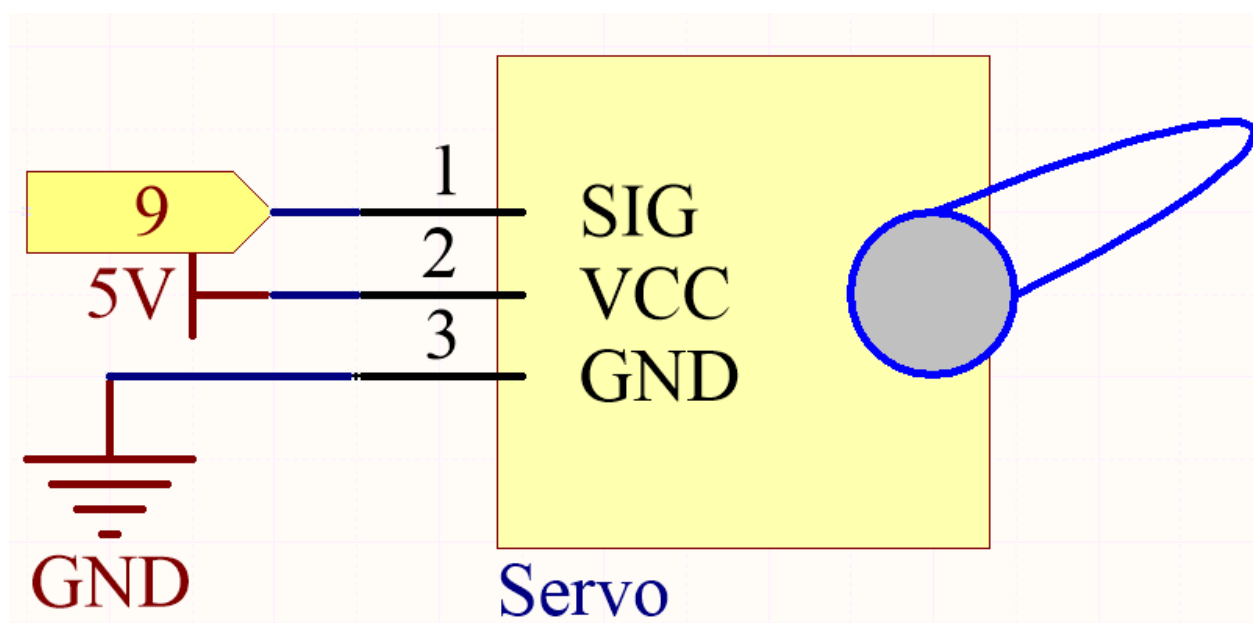
全体のキットを購入するのは確かに便利です。こちらがリンクです：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから、個別に購入することもできます。

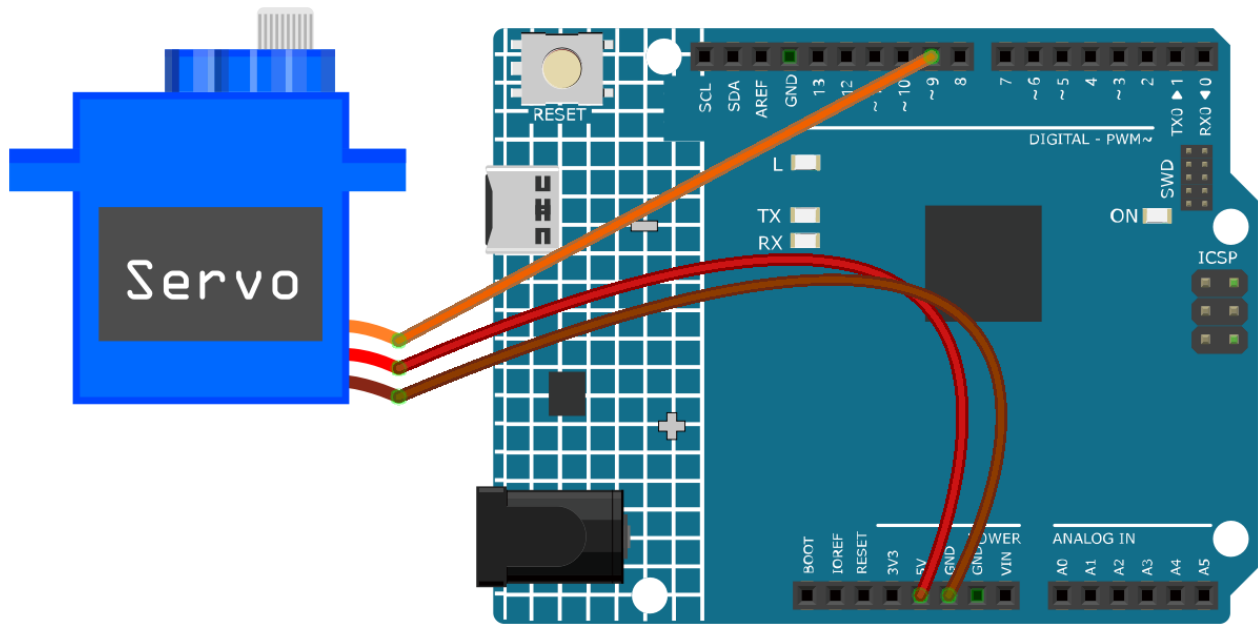
コンポーネントの紹介	購入リンク
Arduino Uno R4 Minima	-
ジャンパーワイヤー	
サーボ	

回路図



このプロジェクトでは、PWM ピン 9 を使用してサーボを駆動し、サーボのオレンジ色のワイヤを PWM ピン 9 に、赤色のワイヤを 5V に、茶色のワイヤを GND に接続します。

配線図



コード

注釈:

- 3in1-kit\learning_project\5.5.use_internal_library のパス下の 5.5.use_internal_library.ino ファイルを開く。
- または、このコードを **Arduino IDE** にコピーする。

コードを R4 ボードにアップロードすると、サーボアームが $0^{\circ} \sim 180^{\circ}$ の範囲で回転するのを確認できます。

どのように動作するのか？

Servo.h ライブラリを呼び出すことで、サーボを簡単に駆動することができます。

```
#include <Servo.h>
```

ライブラリ関数：

```
Servo
```

サーボを制御するための Servo オブジェクトを作成。

```
uint8_t attach(int pin);
```

pinMode() を呼び出して、ピンをサーボドライバーに変換し、失敗した場合は 0 を返す。

```
void detach();
```

サーボ駆動からピンを解放。

```
void write(int value);
```

サーボの角度を度数で設定、0 から 180 まで。

```
int read();
```

最後の write() で設定された値を返す。

```
bool attached();
```

サーボが現在アタッチされている場合は 1 を返す。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.5.6 5.6 マップ

もし注意深く観察すると、多くの値がプログラミング内で異なる範囲を持つことに気付くでしょう。例えば、アナログ入力の値の範囲は (0~1023) です。アナログ出力の値の範囲は (0~255) です。サーボの出力角度は (0~180) です。

これは、ポテンショメータを使用して LED の明るさやサーボの角度を制御したい場合、マッピング操作を行う必要があるということを意味しています。

それでは、どのようにこれを実現するか見てみましょう。

必要な部品

このプロジェクトに必要な部品は以下の通りです。

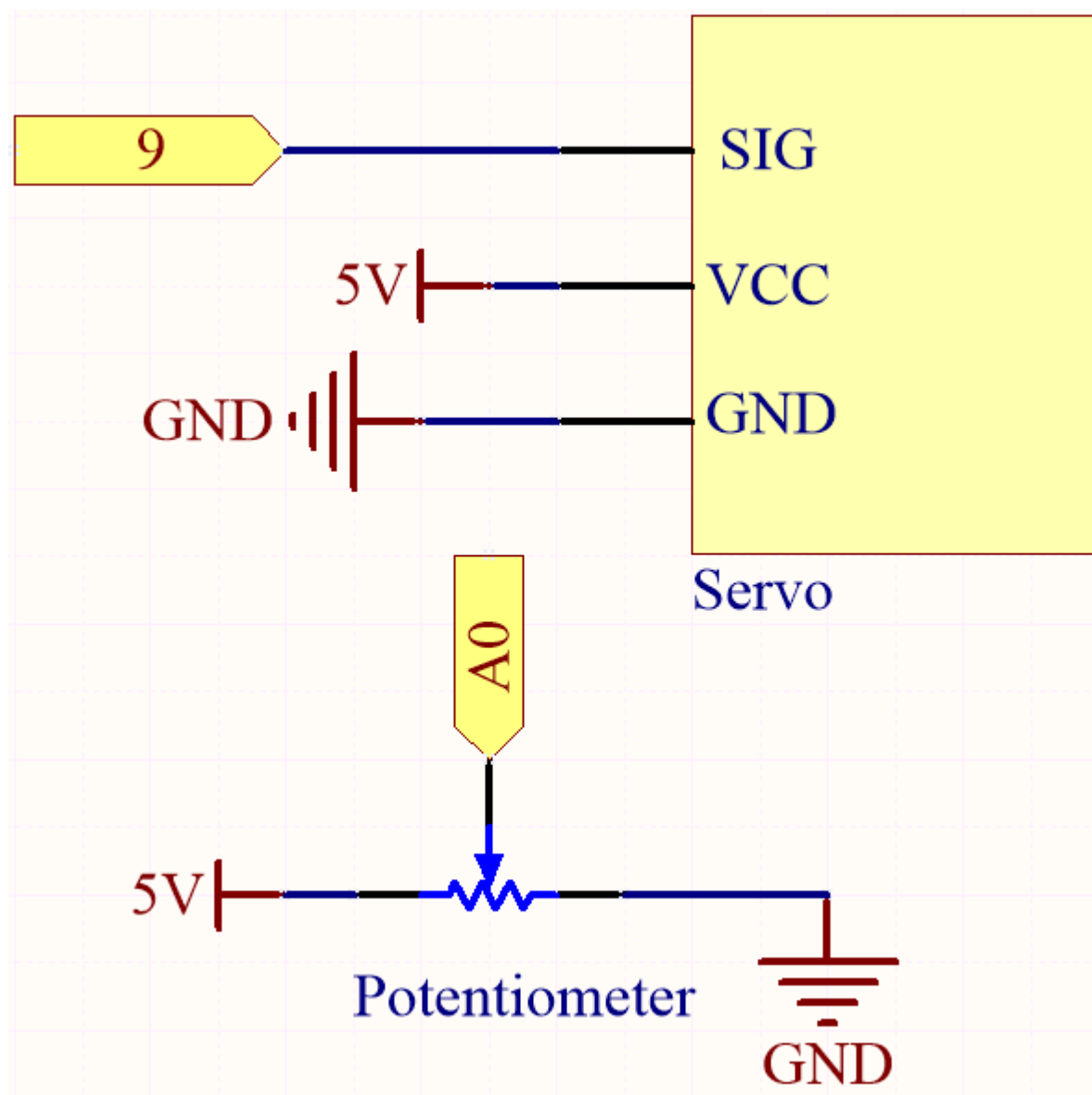
全体のキットを購入すると非常に便利です。以下はリンクです：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

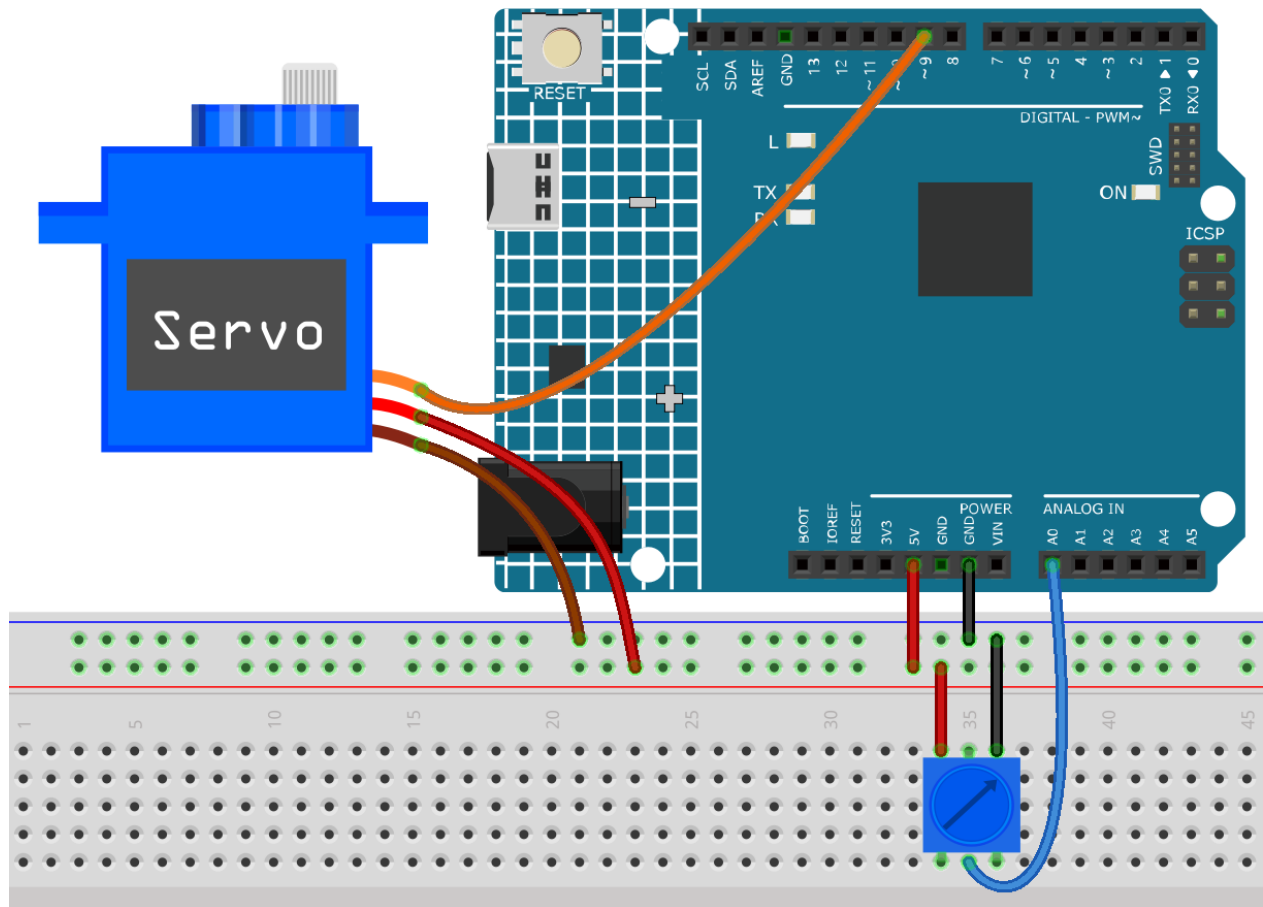
以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
サーボ	
ポテンシオメーター	

回路図



配線図



コード

注釈:

- 3in1-kit\learning_project\5.6.map のパスの下の 5.6.map.ino ファイルを開きます。
- または、このコードを **Arduino IDE** にコピーします。

コードが正常にアップロードされた後、ポテンショメータを前後に回転させると、サーボの出力軸も前後に回転します。

どのように動作するのか？

`map(value, fromLow, fromHigh, toLow, toHigh)`: ある範囲の数字を別の範囲にマッピングします。すなわち、`fromLow` の値は `toLow` に、`fromHigh` の値は `toHigh` にマッピングされます。

構文

```
map(value, fromLow, fromHigh, toLow, toHigh)
```

パラメータ

- value: マッピングする数字。
- fromLow: 値の現在の範囲の下限。
- fromHigh: 値の現在の範囲の上限。
- toLow: 値の目的範囲の下限。
- toHigh: 値の目的範囲の上限。

ポテンショメータで LED を制御する場合、マップを使用してタスクを完了することもできます。

```
int x = analogRead(knob);
int y = map(x,0,1023,0,255);
analogWrite(led,y);
```

注意事項と警告

- 両方の範囲の"下限"は"上限"より大きくまたは小さくなる場合があります、これは map() 関数を使用して数字の範囲を逆転させることができることを意味します。

```
y = map(x,0,180,180,0);
```

- マッピングは負の数に対しても適切に動作します。

```
y = map(x,0,1023,-90,90);
```

- マッピングは整数を使用し、浮動小数点の小数部は破棄されます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.5.7 5.7 Tone() または noTone()

Tone() は指定された周波数の正方形の波（および 50 % のデューティサイクル）をピンで生成するために使用されます。継続時間を指定することもできますが、noTone() が呼び出されるまで波は続きます。

このプロジェクトでは、この二つの関数を使用して、受動ブザーを振動させて音を鳴らします。アクティブブザーと同様に、受動ブザーも電磁誘導の現象を利用して動作します。違いは、受動ブザーには振動源がないため、直流信号が使用されるとピープ音がしないことです。しかし、これにより受動ブザーは自分の振動周波数を調整でき、"ド、レ、ミ、ファ、ソ、ラ、シ"などの異なる音符を発することができます。

必要な部品

このプロジェクトには、以下のコンポーネントが必要です。

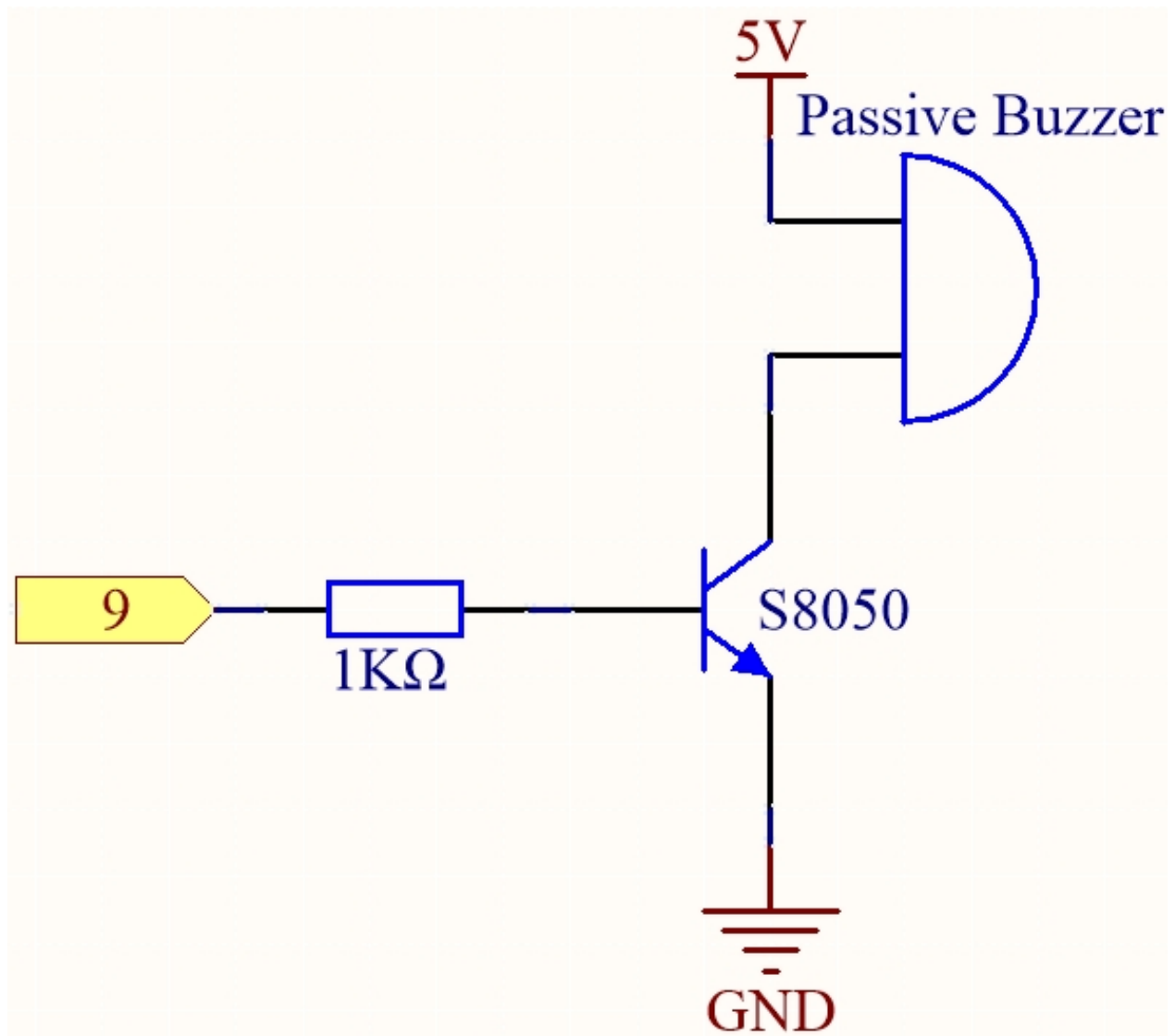
全体のキットを購入するのは確かに便利です、リンクはこちら：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから別々に購入することもできます。

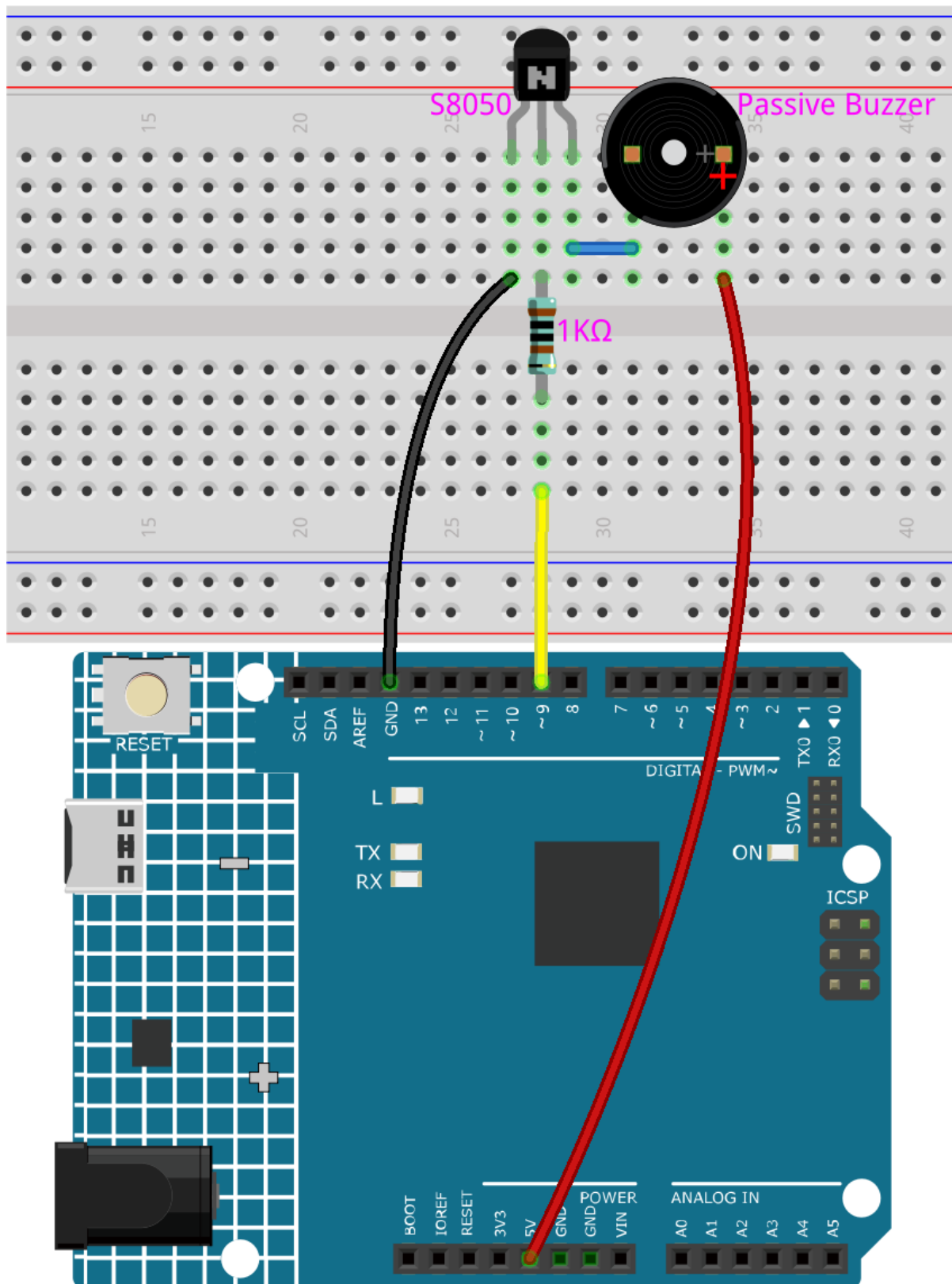
コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
ブザー	

回路図



ブザーのカソードを GND に、アノードをデジタルピン 9 に接続します。

配線図



コード

注釈:

- 3in1-kit\learning_project\5.7.tone_notone のパスの下で 5.7.tone_notone.ino ファイルを開きます。
 - または、このコードを **Arduino IDE** にコピーします。
-

コードを R4 ボードにアップロードしたとき、7 つの音符を含むメロディーが聞こえます。

どのように動作するのか？

以下、2 つの点に注意が必要です：

1. `tone()` & `noTone()`: この関数は、受動ブザーの音を直接制御するために使用され、そのプロトタイプは以下のとおりです：

文法

```
void tone(int pin, unsigned int frequency)
```

```
void tone(int pin, unsigned int frequency, unsigned long duration)
```

パラメータ

- `pin`: トーンを生成する Arduino のピン。
- `frequency`: ヘルツ単位のトーンの周波数。
- `duration`: ミリ秒単位のトーンの持続時間（オプションル）

指定された周波数の正方形の波（および 50 % のデューティサイクル）をピンで生成します（これにより受動ブザーが振動して音を発する）。持続時間を指定でき、指定しない場合は `noTone()` が呼び出されるまで波は続きます。ピンは、ピエゾブザーまたは他のスピーカーに接続して、音を再生することができます。

同時に 1 つのトーンのみを生成することができます。異なるピンですでにトーンが再生されている場合、`tone()` の呼び出しは効果がありません。同じピンでトーンが再生されている場合、呼び出しはその周波数を設定します。

`tone()` 関数の使用は、ピン 3 および 11 の PWM 出力に干渉します。

31Hz より低いトーンを生成することはできません。

文法

```
void noTone(int pin)
```

パラメータ

`pin`: トーンを生成する Arduino のピン。

`tone()` によってトリガされる正方形の波の生成を停止します。トーンが生成されていない場合、効果はありません。

これら 2 つの関数を知ったら、コードの理解が深まります。配列 `melody[]` と配列 `noteDurations[]` のインストールは、後続の複数回の `tone()` 関数の呼び出しや、音楽の再生の効果を高めるためのループ内でのトーンと持続時間の変更の準備です。

2. `pitches.h`: このコードは追加のファイル、`pitches.h` を使用しています。このファイルには、典型的なノートのためのすべてのピッチ値が含まれています。たとえば、`NOTE_C4` は中央の C です。`NOTE_FS4` は F シャープです、などです。このノートテーブルは、`tone()` コマンドがベースになっている Brett Hagman によって元々書かれました。音楽のノートを作りたいときに役立つでしょう。

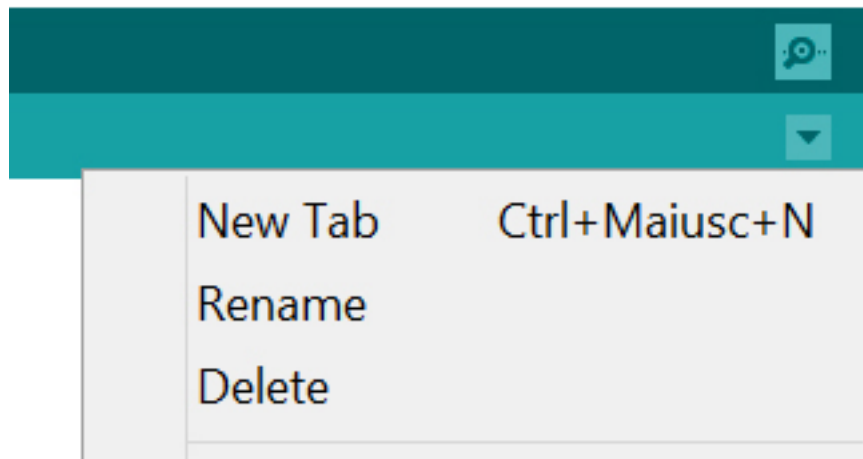
```
#include "pitches.h"
```

注釈: このサンプルプログラムには既に `pitches.h` ファイルがあります。メインコードと同じフォルダにまとめると、`pitches.h` のインストールの続きのステップは省略できます。



コードファイルを開いた後、`pitches.h` コードを開くことができない場合は、手動で 1 つ作成することができます。手順は以下の通りです：

`pitches.h` ファイルを作成するには、シリアルモニターアイコンのすぐ下のボタンをクリックして **New Tab** を選択するか、**Ctrl+Shift+N** を使用します。



次に、以下のコードをペーストして、それを `pitches.h` として保存します：


```
/******  
公開定数  
*****/  
  
#define NOTE_B0  31  
#define NOTE_C1  33  
#define NOTE_CS1 35  
#define NOTE_D1  37  
#define NOTE_DS1 39  
#define NOTE_E1  41  
#define NOTE_F1  44  
#define NOTE_FS1 46  
#define NOTE_G1  49  
#define NOTE_GS1 52  
#define NOTE_A1  55  
#define NOTE_AS1 58  
#define NOTE_B1  62  
#define NOTE_C2  65  
#define NOTE_CS2 69  
#define NOTE_D2  73  
#define NOTE_DS2 78  
#define NOTE_E2  82  
#define NOTE_F2  87  
#define NOTE_FS2 93  
#define NOTE_G2  98  
#define NOTE_GS2 104  
#define NOTE_A2  110  
#define NOTE_AS2 117  
#define NOTE_B2  123  
#define NOTE_C3  131  
#define NOTE_CS3 139  
#define NOTE_D3  147  
#define NOTE_DS3 156  
#define NOTE_E3  165  
#define NOTE_F3  175  
#define NOTE_FS3 185  
#define NOTE_G3  196  
#define NOTE_GS3 208  
#define NOTE_A3  220  
#define NOTE_AS3 233
```

(次のページに続く)

(前のページからの続き)

```
#define NOTE_B3  247
#define NOTE_C4  262
#define NOTE_CS4 277
#define NOTE_D4  294
#define NOTE_DS4 311
#define NOTE_E4  330
#define NOTE_F4  349
#define NOTE_FS4 370
#define NOTE_G4  392
#define NOTE_GS4 415
#define NOTE_A4  440
#define NOTE_AS4 466
#define NOTE_B4  494
#define NOTE_C5  523
#define NOTE_CS5 554
#define NOTE_D5  587
#define NOTE_DS5 622
#define NOTE_E5  659
#define NOTE_F5  698
#define NOTE_FS5 740
#define NOTE_G5  784
#define NOTE_GS5 831
#define NOTE_A5  880
#define NOTE_AS5 932
#define NOTE_B5  988
#define NOTE_C6 1047
#define NOTE_CS6 1109
#define NOTE_D6 1175
#define NOTE_DS6 1245
#define NOTE_E6 1319
#define NOTE_F6 1397
#define NOTE_FS6 1480
#define NOTE_G6 1568
#define NOTE_GS6 1661
#define NOTE_A6 1760
#define NOTE_AS6 1865
#define NOTE_B6 1976
#define NOTE_C7 2093
#define NOTE_CS7 2217
```

(次のページに続く)

(前のページからの続き)

```
#define NOTE_D7 2349
#define NOTE_DS7 2489
#define NOTE_E7 2637
#define NOTE_F7 2794
#define NOTE_FS7 2960
#define NOTE_G7 3136
#define NOTE_GS7 3322
#define NOTE_A7 3520
#define NOTE_AS7 3729
#define NOTE_B7 3951
#define NOTE_C8 4186
#define NOTE_CS8 4435
#define NOTE_D8 4699
#define NOTE_DS8 49
```

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.5.8 5.8 ユーザー定義関数

C 言語では、関数として知られる基本的な構造ブロックに大きなプログラムを分割することができます。関数には {} で囲まれた一連のプログラム文が含まれています。関数は、C プログラムに再利用性とモジュール性を提供するために、複数回呼び出すことができます。言い換えれば、関数の集合がプログラムを作成すると言えます。関数は、他のプログラミング言語では手続きやサブルーチンとしても知られています。

関数には以下のような利点があります。

- 関数を使用することで、プログラム内で同じロジック/コードを何度も書き直すことを避けることができます。
- C の関数は、プログラム内の任意の場所から何度でも呼び出すことができます。
- 複数の関数に分割された大きな C プログラムは、追跡しやすくなります。
- 再利用性は、C 関数の主な達成です。
- ただし、関数の呼び出しは、C プログラムにおいて常にオーバーヘッドとなります。

C プログラミングには 2 種類の関数があります：

- ライブラリ関数：C のヘッダーファイルで宣言されている関数。
- ユーザー定義関数：C プログラムによって作成され、何度でも使用できるようにされた関数。これにより、大きなプログラムの複雑さが減少し、コードが最適化されます。

このプロジェクトでは、超音波モジュールの値を読み取る関数を定義します。

必要な部品

このプロジェクトでは、以下のコンポーネントが必要です。

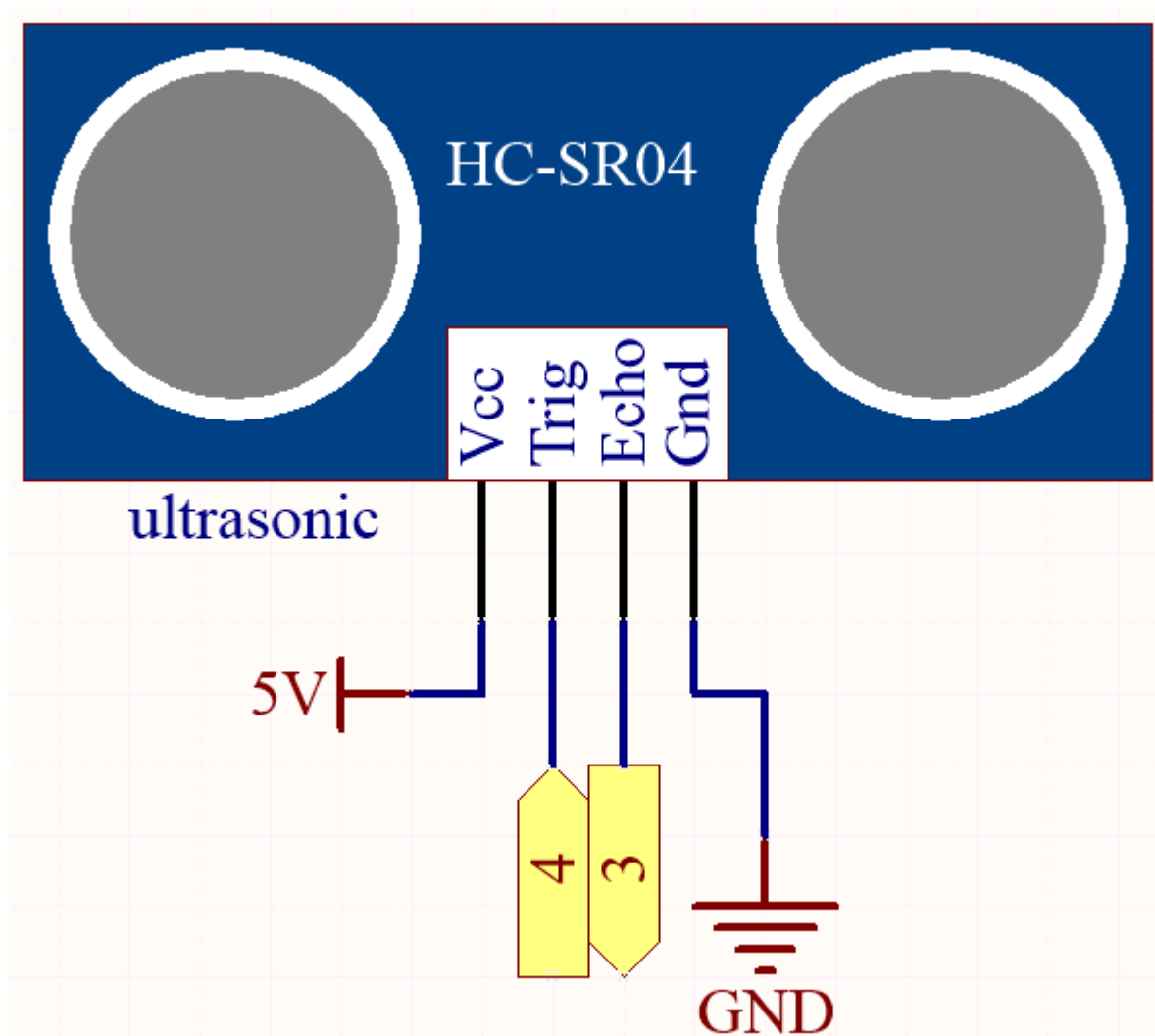
キット全体を購入するのは非常に便利です、こちらがリンクです：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

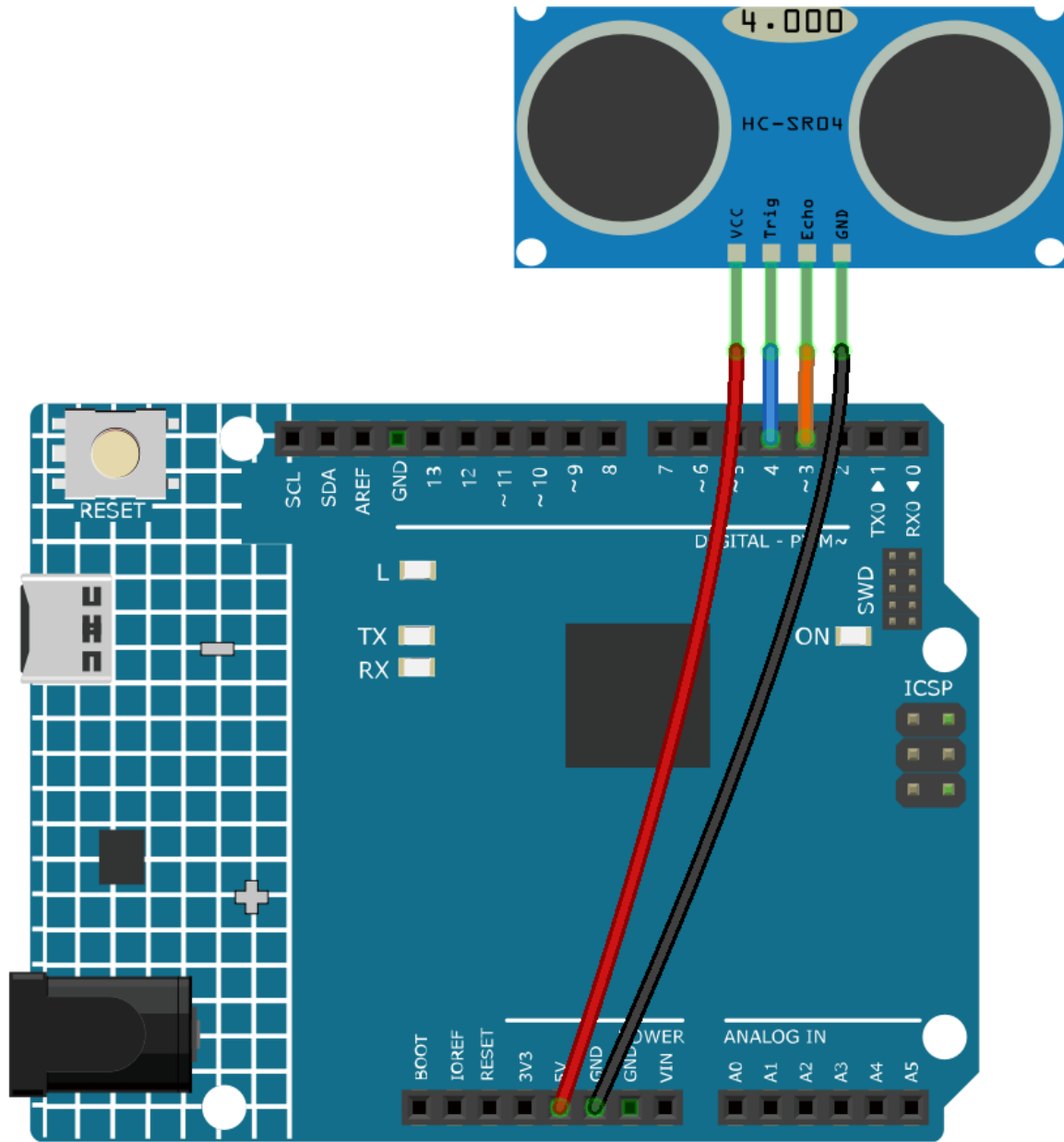
以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ジャンパーワイヤー	
超音波モジュール	

回路図



配線図



コード

注釈:

- 3in1-kit\learning_project\5.8.user_function のパスの下にある 5.8.user_function.ino ファイルを開いてください。
- または、このコードを **Arduino IDE** にコピーしてください。

コードが正常にアップロードされた後、シリアルモニターは超音波センサと前方の障害物との距離を表示します。

どのように動作するのか？

超音波センサの利用については、サブファンクションを直接確認できます。

```
float readSensorData(){// ...}
```

超音波モジュールの trigPin は、2us ごとに 10us の正方波信号を送信します。

```
digitalWrite(trigPin, LOW);  
delayMicroseconds(2);  
digitalWrite(trigPin, HIGH);  
delayMicroseconds(10);  
digitalWrite(trigPin, LOW);
```

範囲内に障害物がある場合、echoPin は高レベルの信号を受信し、pulseIn() 関数を使用して送信から受信までの時間を記録します。

```
microsecond=pulseIn(echoPin, HIGH);
```

音の速度は、340 m/s または 1cm あたり 29 マイクロ秒です。

これにより、正方波が移動した距離、外向きと帰還、が得られるので、障害物の距離を取得するために 2 で割ります。

```
float distance = microsecond / 29.00 / 2;
```

超音波センサは動作しているときにプログラムを一時停止するため、複雑なプロジェクトを書いているときにラグが発生することがあります。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.5.9 5.9 ShiftOut(LED)

shiftOut() は 74HC595 に 8 つのデジタル信号を出力させることができます。バイナリ番号の最後のビットを Q0 に、最初のビットを Q7 に出力します。言い換えれば、バイナリ番号「00000001」を書き込むと、Q0 は高レベルを出力し、Q1 ~ Q7 は低レベルを出力します。

このプロジェクトでは、74HC595 の使用方法を学びます。74HC595 は、三状態並列出力を持つ 8 ビットシフトレジスタとストレージレジスタで構成されています。これにより、シリアル入力を並列出力に変換し、MCU の IO ポートを節約できます。

具体的には、74hc595 は 8 ビットのバイナリ数を書き込むことで、デジタル信号出力のための 8 つのピンを代替できます。

- [バイナリ数 - Wikipedia](#)

必要な部品

このプロジェクトでは、以下の部品が必要です。

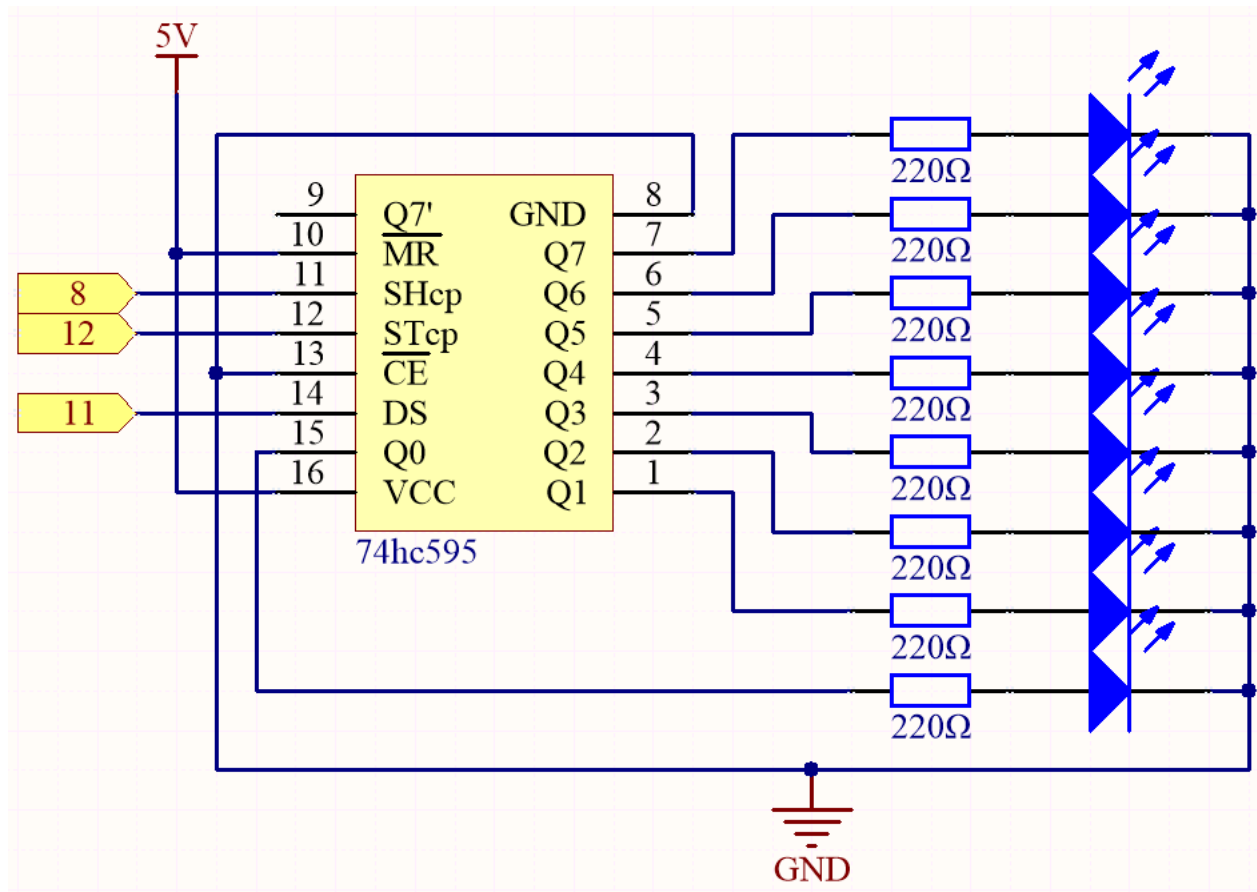
全体のキットを購入するのが非常に便利です。リンクはこちら：

名前	このキットの内容	リンク
3 in 1 スターターキット	380+	

以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
<i>LED</i>	
<i>74HC595</i>	

回路図



- MR (ピン 10) が高レベルで、OE (ピン 13) が低レベルのとき、SHcp の立ち上がりエッジでデータが入力され、SHcp の立ち上がりエッジを経てメモリレジスタに移動します。
- 二つのクロックが一緒に接続されている場合、シフトレジスタはメモリレジスタより常に一つのパルスが早いです。
- メモリレジスタには、シリアルシフト入力ピン (Ds)、シリアル出力ピン (Q)、非同期リセットボタン (低レベル) があります。
- メモリレジスタは、3 つの状態での並列 8 ビットのバスを出力します。
- OE が有効 (低レベル) のとき、メモリレジスタのデータがバス (Q0 ~ Q7) に出力されます。

配線図

最初に STcp を低レベルに設定し、次に高レベルに設定します。これにより、STcp の立ち上がりエッジのパルスが生成されます。

```
digitalWrite(STcp,LOW);
```

shiftOut() は、一度に 1 ビットのデータをシフトアウトするために使用されます。つまり、dataArray[num] のデータの 1 バイトを DS ピンでシフトレジスタにシフトします。MSBFIRST は高ビットから移動することを意味します。

```
shiftOut(DS,SHcp,MSBFIRST,dataArray[num]);
```

digitalWrite(STcp,HIGH) が実行されると、STcp は立ち上がりエッジになります。この時、シフトレジスタのデータがメモリレジスタに移動します。

```
digitalWrite(STcp,HIGH);
```

8 回後、1 バイトのデータがメモリレジスタに転送されます。その後、メモリレジスタのデータがバス (Q0-Q7) に出力されます。例えば、B00000001 をシフトアウトすると、Q0 で制御される LED が点灯し、Q1 ~ Q7 で制御される LED が消灯します。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

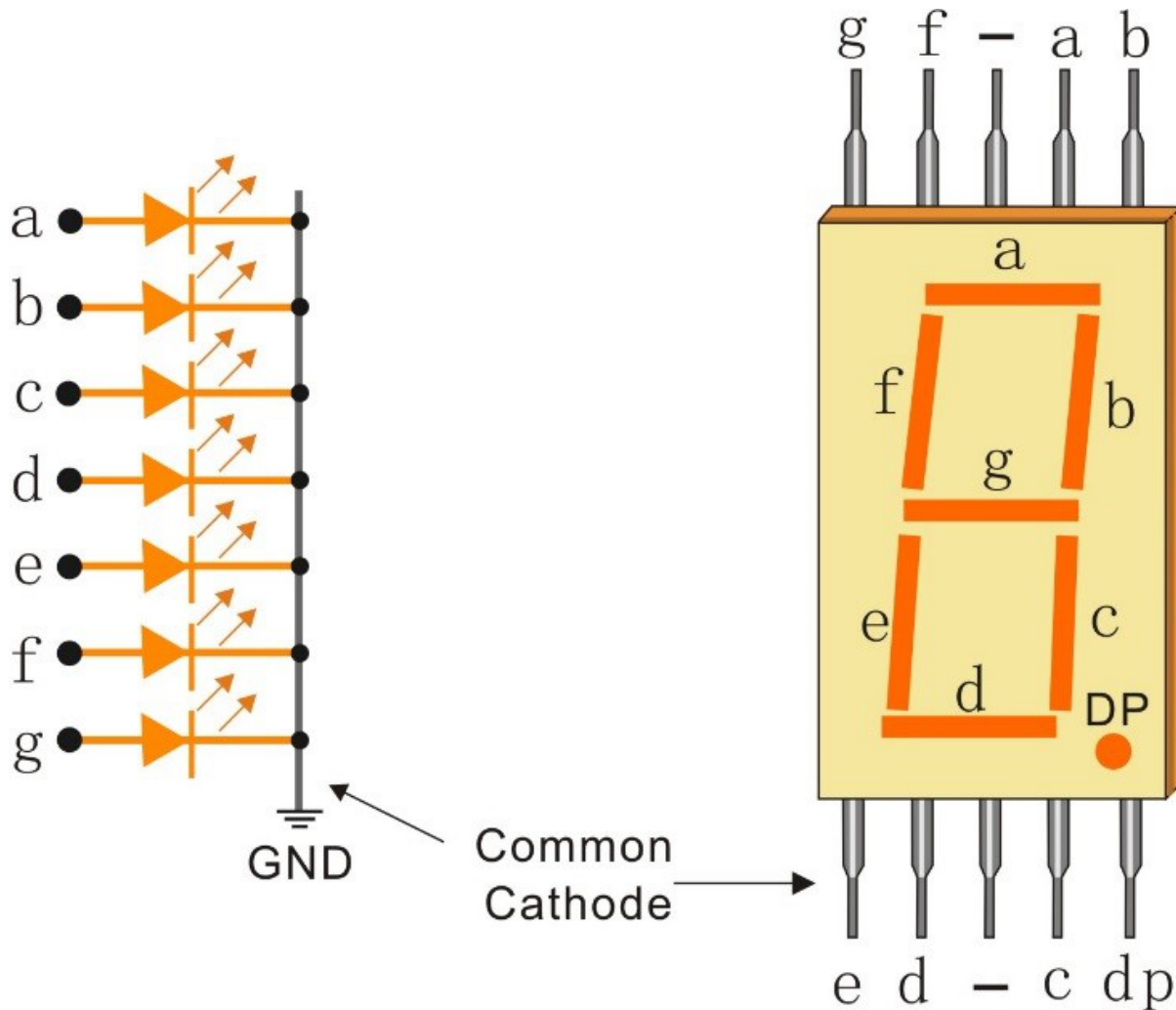
- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び & 共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.5.10 5.10 シフトアウト (セグメントディスプレイ)

以前、私たちは `shiftout()` 関数を使用して 8 つの LED を点灯させましたが、このセクションでは 7 セグメントディスプレイで 0-9 を表示させる方法を学びます。

7 セグメントディスプレイは、実質的に 8 つの LED で構成されており、そのうちの 7 つのストリップ形状の LED が「8」の形を作り、小さめのドットの LED が小数点として機能します。これらの LED は、a、b、c、d、e、f、g、および dp としてマークされています。それぞれが独自のアノードピンを持ち、カソードを共有しています。ピンの位置関係は下図に示されています。



必要な部品

このプロジェクトには、以下の部品が必要です。

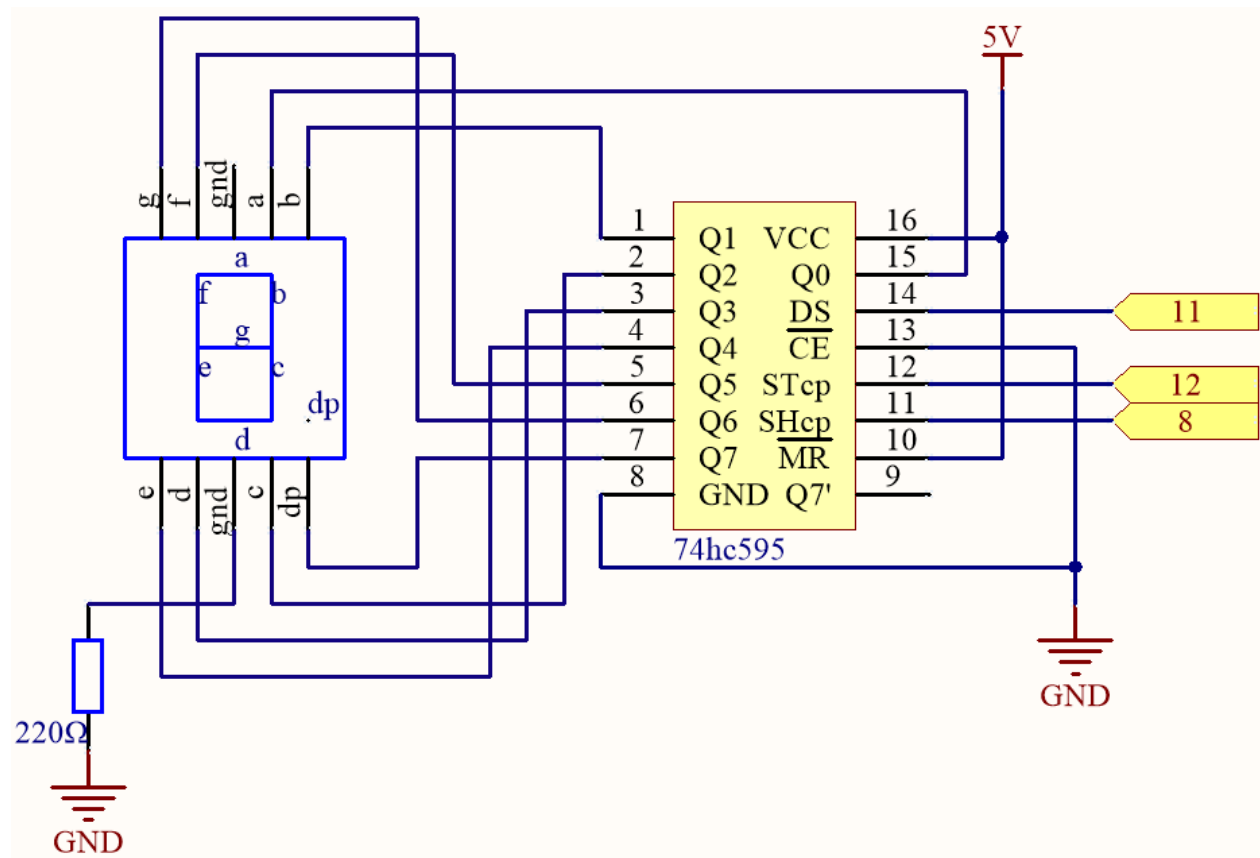
一式をまとめて購入するのがおすすめです。リンクは以下になります：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

下記のリンクから個別に購入することも可能です。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
7 セグメントディスプレイ	
<i>74HC595</i>	

回路図



配線図

表 1 配線

74HC595	LED セグメントディスプレイ
Q0	a
Q1	b
Q2	c
Q3	d
Q4	e
Q5	f
Q6	g
Q7	dp

コード

注釈:

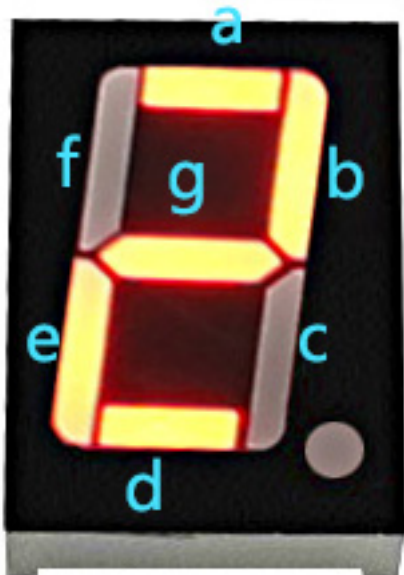
- 3in1-kit\learning_project\5.10.shiftout_segment のパスの下にある 5.10.shiftout_segment.ino ファイルを開きます。
 - または、このコードを **Arduino IDE** にコピーします。
-

コードのアップロードが成功した後、LED セグメントディスプレイで 0~9 が順番に表示されることが確認できます。

どのように動作するのか？

shiftOut() は 74HC595 に 8 つのデジタル信号を出力させます。最後のビットのバイナリ数を Q0 に、最初のビットの出力を Q7 に出力します。つまり、バイナリ数「00000001」を書くと、Q0 はハイレベルを出力し、Q1~Q7 はローレベルを出力します。

7 セグメントディスプレイで数字「2」を表示すると仮定します。a, b, d, e, g にハイレベルを書き、c, f, dp にローレベルを書く必要があります。つまり、バイナリ数「01011011」を書く必要があります。可読性のため、16 進数の表記「0x5b」を使用します。



- 16 進数
- [BinaryHex 変換ツール](#)

同様に、同じ方法で 7 セグメントディスプレイに他の数字を表示させることもできます。以下のテーブルは、これらの数字に対応するコードを示しています。

表2 グリフコード

数字	バイナリコード	16 進数コード
0	00111111	0x3f
1	00000110	0x06
2	01011011	0x5b
3	01001111	0x4f
4	01100110	0x66
5	01101101	0x6d
6	01111101	0x7d
7	00000111	0x07
8	01111111	0x7f
9	01101111	0x6f

これらのコードを `shiftOut()` に入力すると、LED セグメントディスプレイに対応する数字が表示されます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び & 共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.5.11 5.11 外部ライブラリのインストール

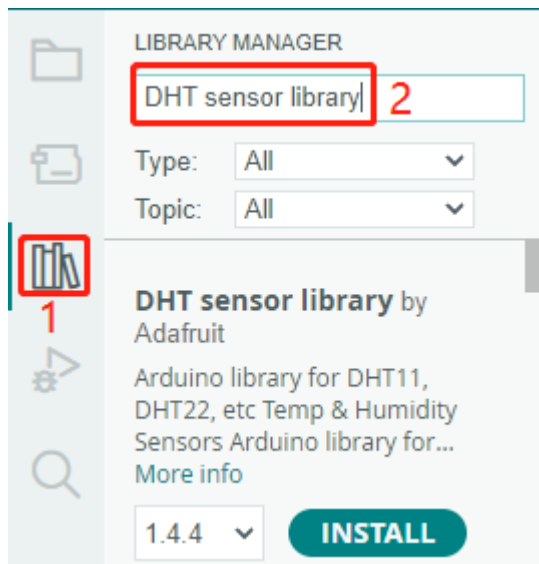
ライブラリは、Arduino IDE の機能を拡張する前に書かれたコードや関数の集合です。ライブラリは様々な機能のための使用準備が整ったコードを提供し、複雑な機能のコーディングにかかる時間と労力を節約できます。

ライブラリをインストールする主な方法は2つあります：

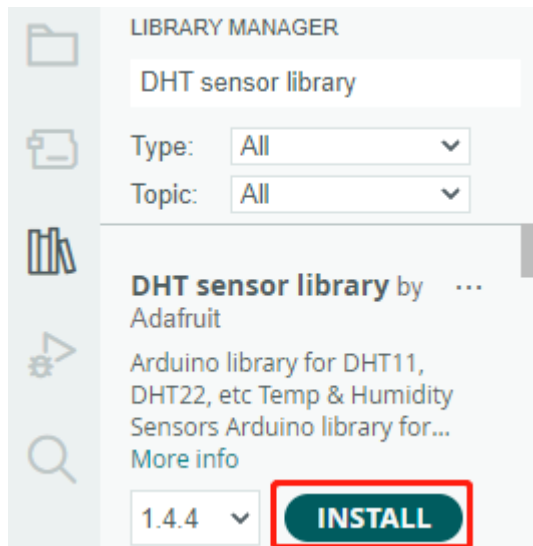
多くのライブラリは Arduino の **Library Manager** を通じて直接利用できます。以下の手順で **Library Manager** にアクセスできます：

1. **Library Manager** で、名前で目的のライブラリを検索したり、異なるカテゴリをブラウズしたりできます。

注釈: ライブラリのインストールが必要なプロジェクトでは、インストールするライブラリに関するプロンプトが表示されます。例えば、「DHT sensor library をこちらで使用しています。 **Library Manager** からインストールできます」といった指示に従ってください。推奨されるライブラリをプロンプトに従ってインストールするだけです。



2. インストールしたいライブラリを見つけたら、それをクリックし、次に **Install** ボタンをクリックします。



3. Arduino IDE は自動的にライブラリをダウンロードしてインストールします。

関連コンポーネント

以下は関連するコンポーネントであり、それらの使用方法を学ぶためにクリックできます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なレビュー：新製品の発表や先行レビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

5.11.1 液晶ディスプレイ

I2C 接続の LCD1602 は、LCD1602 本体と I2C モジュールで構成されています。LCD1602 は文字や数字などを表示するために使用できますが、多くの主制御のピンを使用する必要があります。I2C モジュールを構成することで、この LCD1602 を駆動するのに必要な I/O ピンは 2 つだけになります。

さて、この I2C 接続の LCD1602 をどのように動作させるか見てみましょう。

必要な部品

このプロジェクトには、以下の部品が必要です。

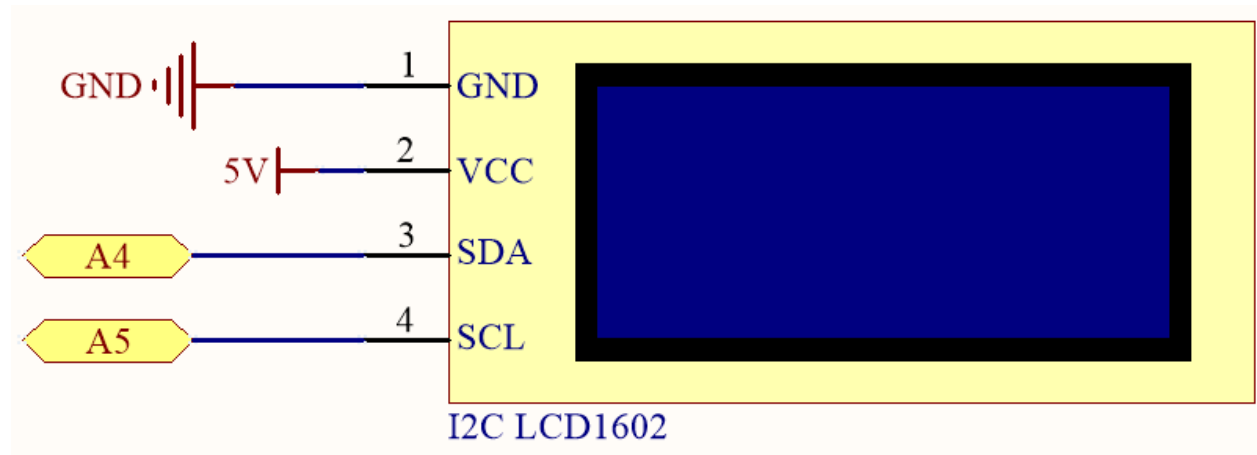
一式のキットとして購入すると非常に便利です。購入リンクは以下の通りです。

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

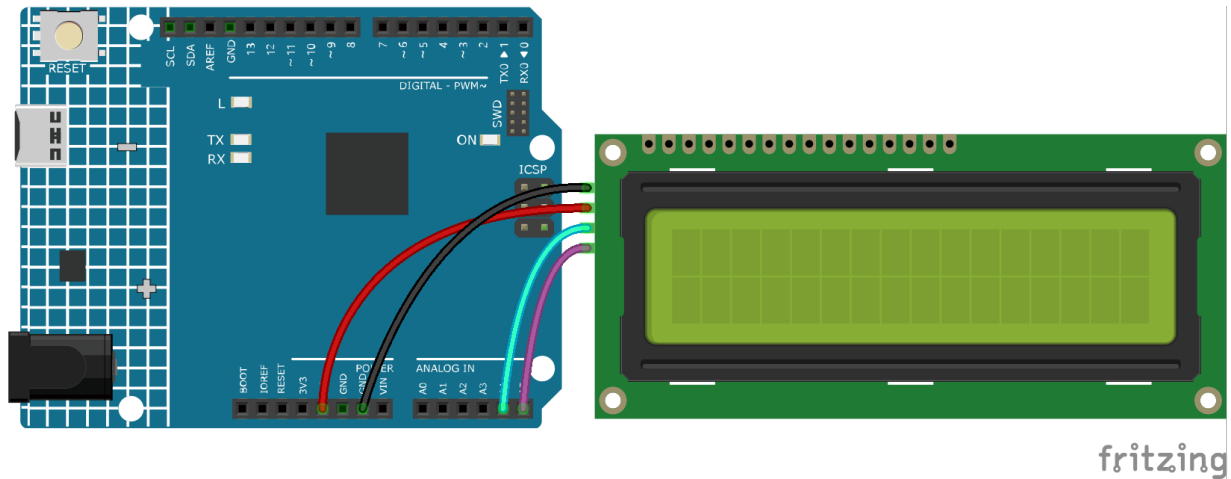
以下のリンクから個別に購入することも可能です。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ジャンパーワイヤー	
<i>I2C LCD1602</i>	

回路図



配線図

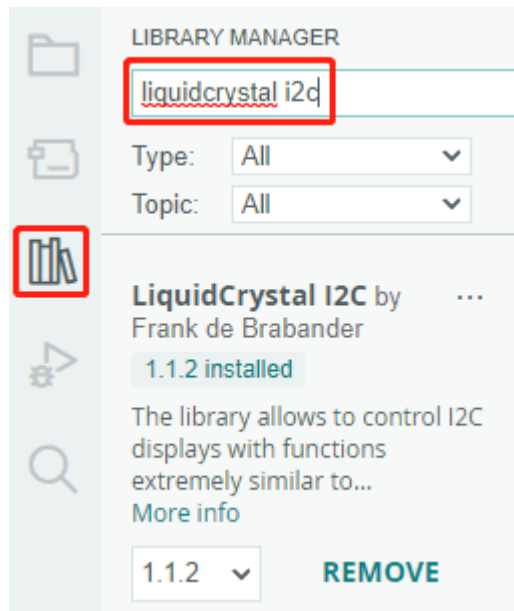


注釈: R4 ボードの SDA と SCL は、A4 ピンと A5 ピンです。

コード

注釈:

- 3in1-kit\learning_project\5.11.liquid_crystal_display のパス下の 5.11.liquid_crystal_display.ino ファイルを開きます。
- または、このコードを **Arduino IDE** にコピーしてください。
- ここでは LiquidCrystal I2C ライブラリが使用されています。 **Library Manager** からインストールできます。



コードが正常にアップロードされると、I2C 接続の LCD1602 に「SunFounder」と「Hello World」が表示されます。

注釈: コードと配線が問題ない場合でも、LCD に内容が表示されない場合は、背面のポテンショメータを回してみてください。

どのように動作するのか？

LiquidCrystal_I2C.h ライブラリを呼び出すことで、LCD を簡単に駆動できます。

```
#include "LiquidCrystal_I2C.h"
```

ライブラリ関数:

```
LiquidCrystal_I2C(uint8_t lcd_Addr, uint8_t lcd_cols, uint8_t lcd_rows)
```

Arduino ボードに接続された特定の LCD を表す LiquidCrystal_I2C クラスの新しいインスタンスを作成します。

- lcd_Addr: LCD のアドレスはデフォルトで 0x27 です。
- lcd_cols: LCD1602 は 16 列を持っています。
- lcd_rows: LCD1602 は 2 行を持っています。

```
void init()
```

LCD を初期化します。

```
void backlight()
```

(オプションの) バックライトをオンにします。

```
void nobacklight()
```

(オプションの) バックライトをオフにします。

```
void display()
```

LCD ディスプレイをオンにします。

```
void nodisplay()
```

LCD ディスプレイをすばやくオフにします。

```
void clear()
```

ディスプレイをクリアし、カーソル位置をゼロに設定します。

```
void setCursor(uint8_t col, uint8_t row)
```

カーソル位置を col,row に設定します。

```
void print(data, BASE)
```

テキストを LCD に出力します。

- data: 出力するデータ (char、byte、int、long、または string)
- BASE (オプション): 数値を出力するベース : BIN はバイナリ (基数 2)、DEC は 10 進数 (基数 10)、OCT は 8 進数 (基数 8)、HEX は 16 進数 (基数 16)。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。

- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

5.11.2 IR 受信機

このプロジェクトでは、IR 受信機の使用方法を学びます。

赤外線受信機は、赤外線信号を受信するコンポーネントで、独立して赤外線を受信し、TTL レベルと互換性のある信号を出力できます。通常のプラスチックパッケージのトランジスタと同じサイズで、あらゆる種類の赤外線リモコンや赤外線送信に適しています。

必要な部品

このプロジェクトに必要なコンポーネントは以下の通りです。

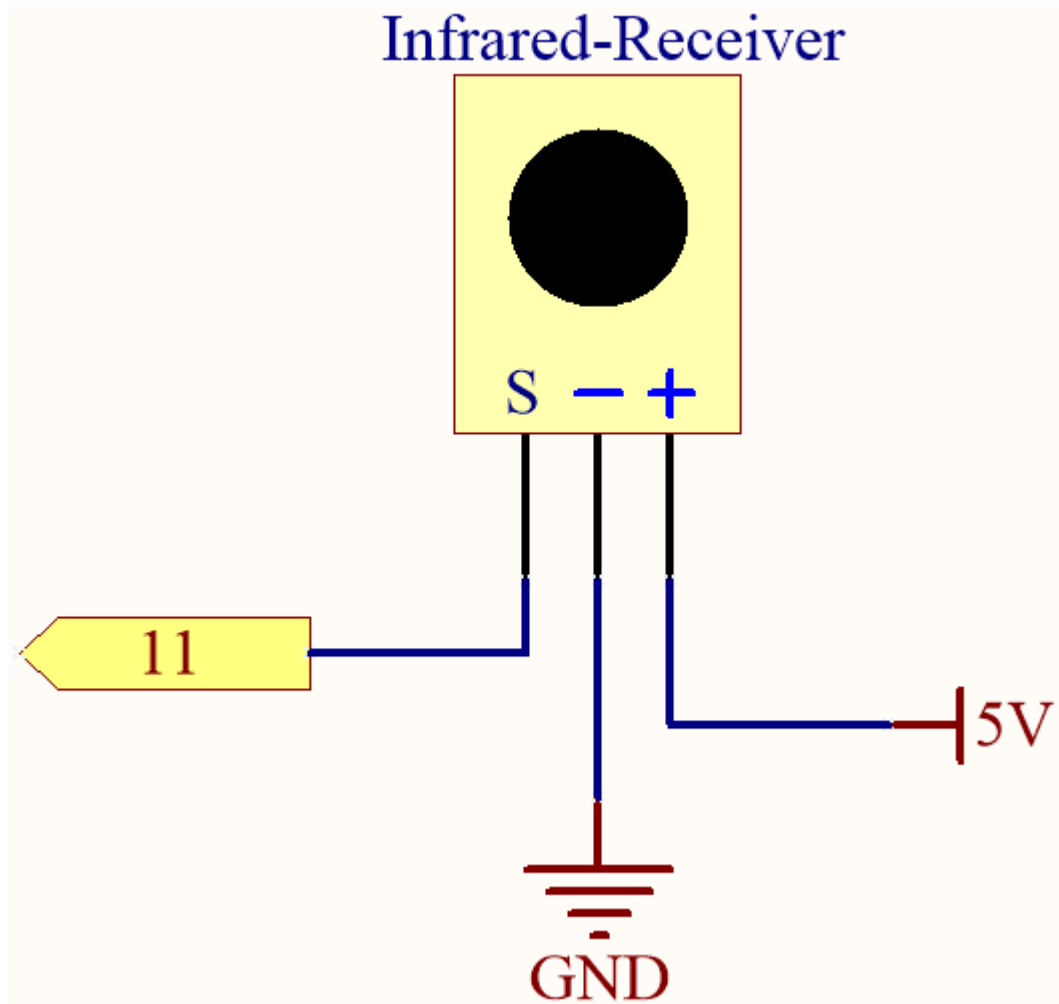
一式を購入するのは非常に便利です。以下にリンクを示します。

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから個別に購入することもできます。

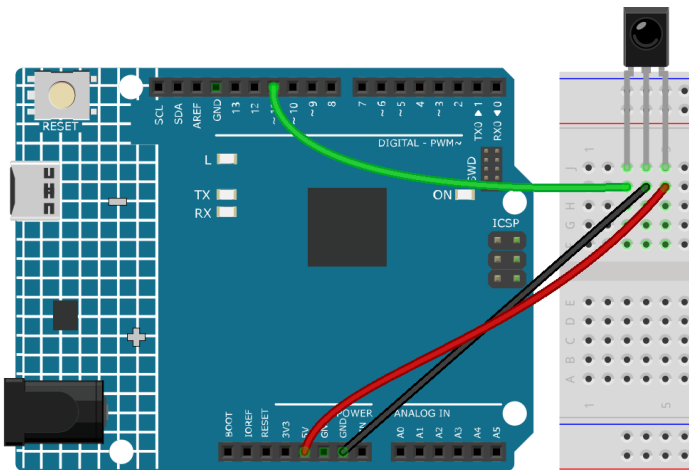
コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
IR 受信機	-

回路図



配線図

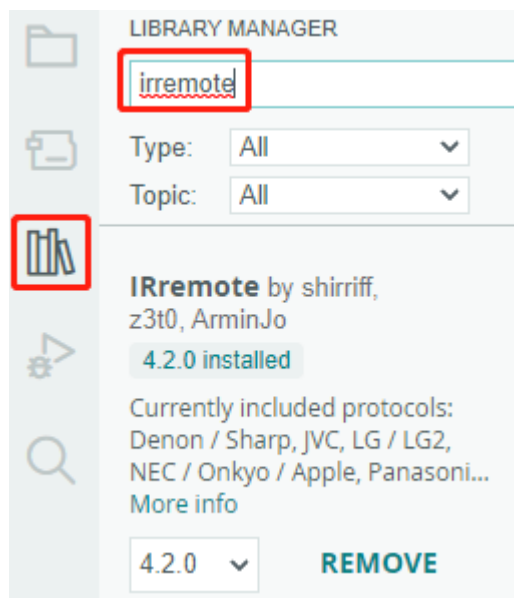
この例では、IR 受信機の左ピンをピン 11 に、中央ピンを GND に、右ピンを 5V に接続します。



コード

注釈:

- 3in1-kit\learning_project\5.11.ir_receiver のパスで 5.11.ir_receiver.ino ファイルを開きます。
- または、このコードを **Arduino IDE** にコピーします。
- ここでは IRremote ライブラリを使用しています。 **Library Manager** からインストールできます。



R4 ボードにコードをアップロードすると、IR リモートコントローラの押されたボタンの現在の値がシリアルモニタに表示されます。

どのように動作するのか？

このコードは、IRremote ライブラリを使用して赤外線 (IR) リモートコントロールと連携して動作するように設計されています。以下に詳細を示します：

1. ライブラリのインクルード: IRremote ライブラリをインクルードして、IR リモートコントロールと連携する関数を提供します。

```
#include <IRremote.h>
```

2. IR センサのシグナルピンが接続されている Arduino のピンを定義し。

```
const int IR_RECEIVE_PIN = 11; // IR センサのピン番号を定義
```

3. ボーレート 9600 でシリアル通信を初期化します。指定されたピン (IR_RECEIVE_PIN) で IR 受信機を初期化し、LED フィードバックを有効にします (該当する場合)。

```
void setup() {  
    Serial.begin(9600); // ボーレート 9600 でシ  
リアル通信を開始  
    IrReceiver.begin(IR_RECEIVE_PIN, ENABLE_LED_FEEDBACK); // IR 受信機を開始  
}
```

4. ループは、入力される IR リモート信号を継続的に処理するために継続的に実行されます。

```
void loop() {  
    if (IrReceiver.decode()) {  
        String decodedValue = decodeKeyValue(IrReceiver.decodedIRData.command);  
        if (decodedValue != "ERROR") {  
            Serial.println(decodedValue);  
            delay(100);  
        }  
        IrReceiver.resume(); // Enable receiving of the next value  
    }  
}
```

- IR 信号が受信され、正常にデコードされたかどうかを確認します。
- decodeKeyValue() 関数を使用して IR コマンドをデコードし、decodedValue に保存します。
- デコードされた値がエラーでないか確認します
- デコードされた IR の値をシリアルモニタに出力します。
- 次の信号のための IR 信号受信を再開します。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なレビュー：新製品の発表や先行レビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

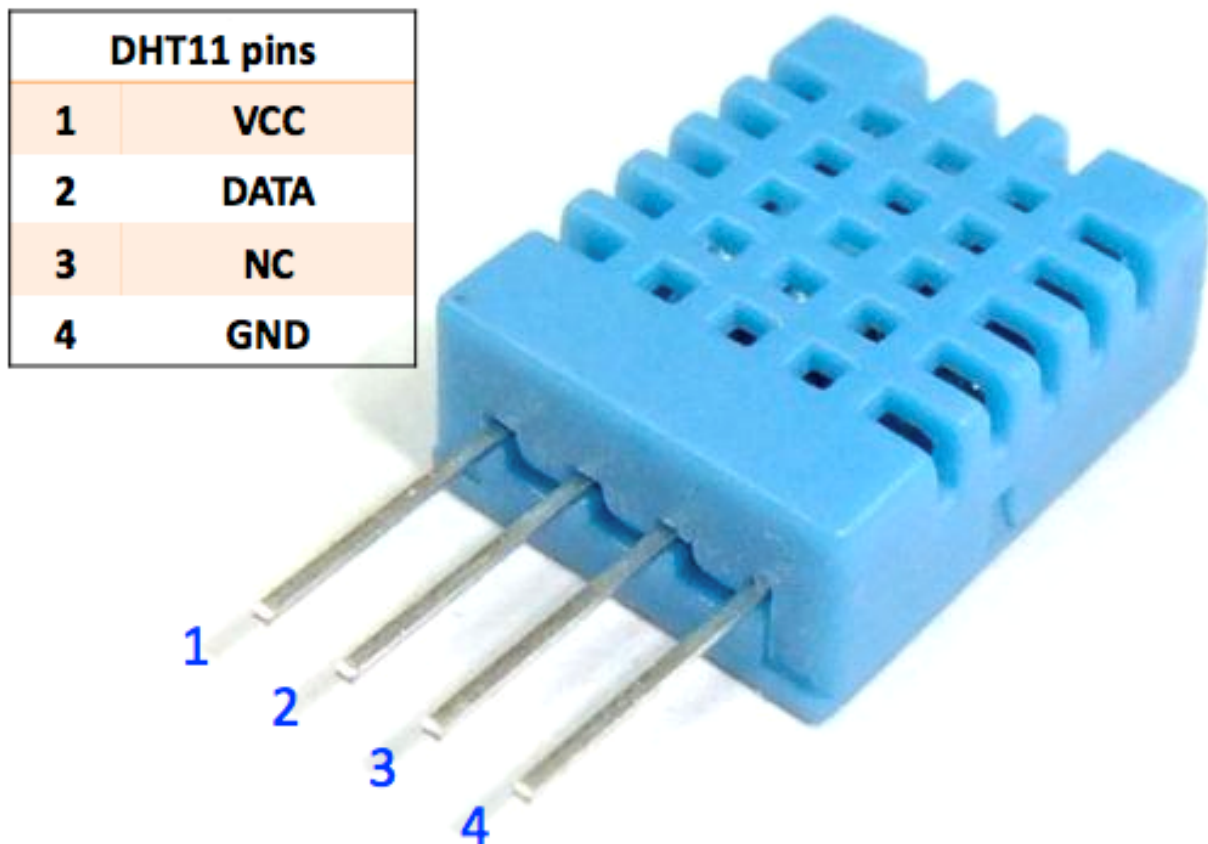
私たちと一緒に探索し、創造する準備はできていますか？ [] をクリックして今すぐ参加しましょう！

5.11.3 温度 - 湿度

湿度と温度は、物理量自体から実際の人々の生活まで、密接に関連しています。人間の生活環境の温度と湿度は、人体の体温調節機能と熱伝達効果に直接影響を与えます。これはさらに思考活動と精神状態に影響を与え、私たちの学習と仕事の効率に影響を与えます。

温度は、国際単位系における 7 つの基本物理量の一つで、物体の熱さや冷たさを測定するために使用されます。摂氏は、世界で広く使用されている温度尺度の一つで、" ° " という記号で表されます。

湿度は、空気中に存在する水蒸気の濃度を示します。相対湿度は日常生活でよく使用され、%RH で表されます。相対湿度は温度と密接に関連しています。一定の容積の密閉ガスに対して、温度が高ければ相対湿度は低く、温度が低ければ相対湿度は高くなります。



このキットには、デジタル温湿度センサである dht11 が付属しています。周囲の空気を測定するための静電容量式湿度センサとサーミスタを使用し、データピンにデジタル信号を出力します。

必要な部品

SunFounder 3in1 Kit

このプロジェクトでは、以下のコンポーネントが必要です。

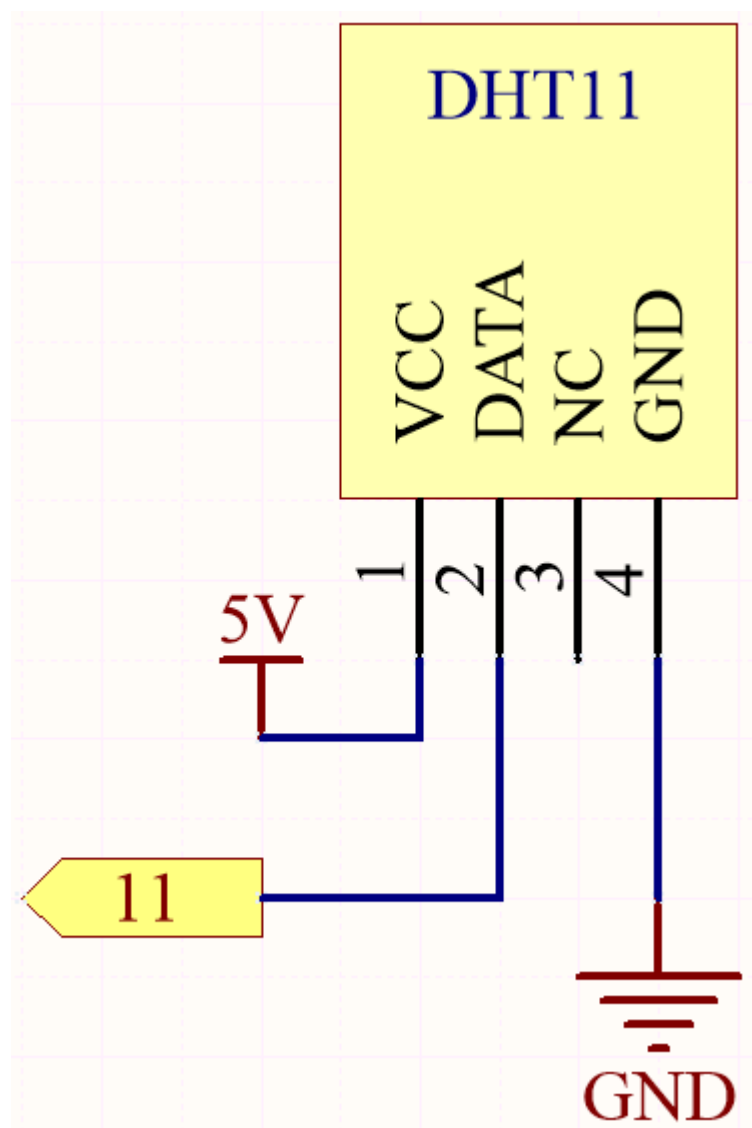
キットを一式で購入するのは非常に便利です、リンクは以下の通りです：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

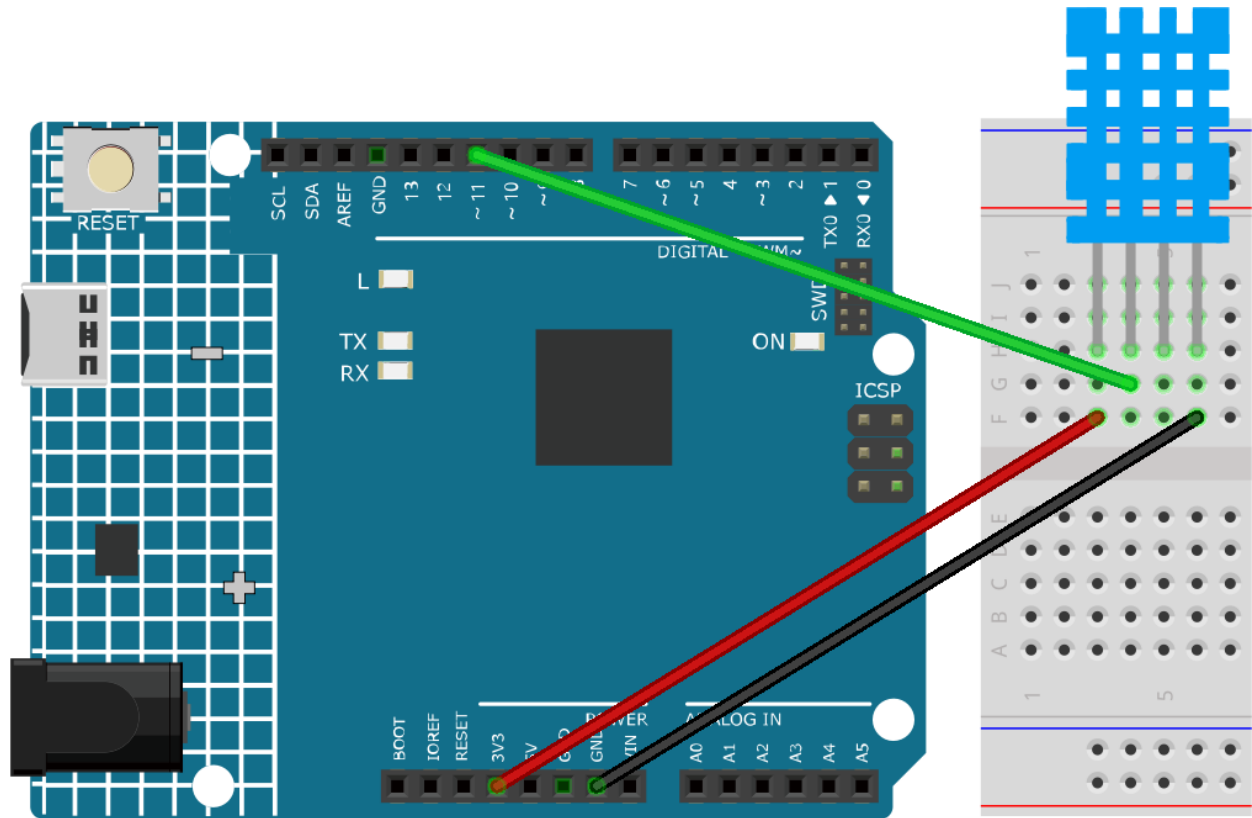
以下のリンクから別々に購入することもできます。

コンポーネントの紹介	購入リンク
Arduino Uno R4 Minima	-
ブレッドボード	
ジャンパーワイヤー	
DHT11 温湿度センサ	-

回路図



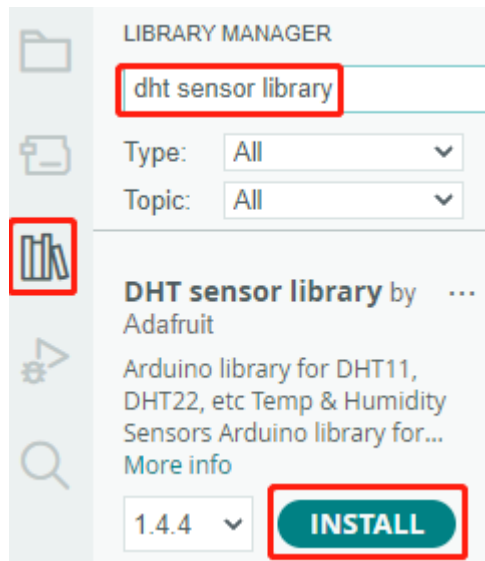
配線図



コード

注釈:

- 3in1-kit\learning_project\5.11.temperature_humidity のパスの下で 5.11.temperature_humidity.ino ファイルを開きます。
- または、このコードを **Arduino IDE** にコピーします。
- ここでは DHT sensor library を使用しています。 **Library Manager** からインストールできます。



コードが正常にアップロードされた後、シリアルモニターが連続して温度と湿度を表示するのを見ることができます。プログラムが安定して実行されると、これらの2つの値はますます正確になります。

どのように動作するのか？

1. DHT.h ライブラリを含めることで、DHT センサと対話するための関数を提供します。次に、DHT センサのピンとタイプを設定します。

```
#include "DHT.h"

#define DHTPIN 11 // DHT11 データピンに接続されたピンを設定
#define DHTTYPE DHT11 // DHT 11

DHT dht(DHTPIN, DHTTYPE);
```

2. 115200 のボーレートでシリアル通信を初期化し、DHT センサを初期化します。

```
void setup() {
  Serial.begin(115200);
  Serial.println("DHT11 test!");
  dht.begin();
}
```

3. loop() 関数で、DHT11 センサから温度と湿度の値を読み取り、それらをシリアルモニターに表示します。

```
void loop() {
  // 測定の間に数秒待機します。
```

(次のページに続く)

(前のページからの続き)

```
delay(2000);

// 温度または湿度の読み取りには約 250 ミリ秒かかります！
// センサーの読み取りは最大 2 秒「古い」ものになることがあります（非常に遅いセンサーで
す）
float humidity = dht.readHumidity();
// 温度を摂氏で読み取る（デフォルト）
float temperture = dht.readTemperature();

// 読み取りが失敗したかどうかを確認して、早期に終了します（再試行するため）。
if (isnan(humidity) || isnan(temperture)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
}
// 湿度と温度を表示
Serial.print("Humidity: ");
Serial.print(humidity);
Serial.print(" %\t");
Serial.print("Temperature: ");
Serial.print(temperture);
Serial.println(" *C");
}
```

- dht.readHumidity() 関数は、DHT センサから湿度値を読み取るために呼び出されます。
- dht.readTemperature() 関数は、DHT センサから温度値を読み取るために呼び出されます。
- isnan() 関数は、読み取りが有効であるかどうかを確認するために使用されます。湿度または温度の値が NaN (数値でない) の場合、センサーからの読み取りに失敗したことを示し、エラーメッセージが表示されます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。

- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.5.12 5.12 シリアル読み取り

`Serial.print()` 関数を使用するときにこれに気付くかもしれません。表示機能があるのなら、読み取り機能もあるのでは？シリアルモニタのテキストボックスは何のために使われるのでしょうか？正解です、シリアルモニタのテキストボックスを通じて情報を入力することで、プログラムや回路を制御する方法があります。

このプロジェクトでは、シリアルモニタで入力されたテキストを I2C LCD1602 に表示することで、`Serial.read()` の使用法を体験します。

必要な部品

このプロジェクトには以下の部品が必要です。

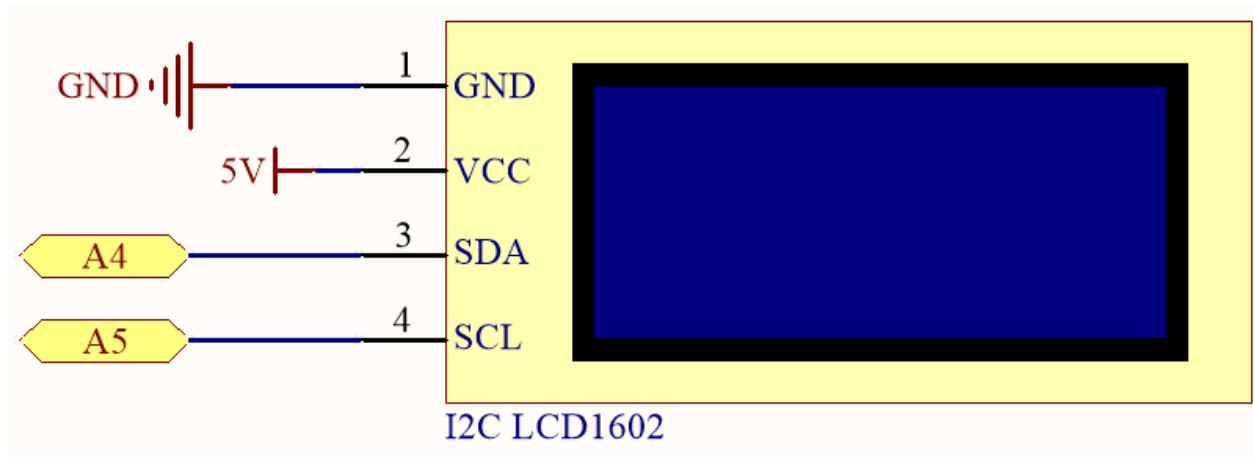
一式を購入するのが便利です、リンクはこちらです:

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

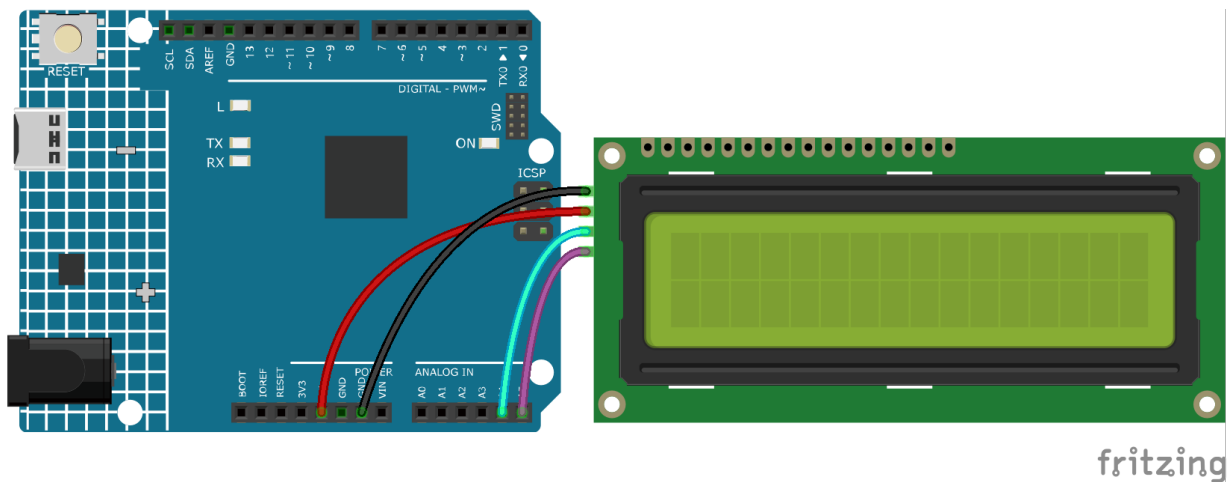
以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ジャンパーワイヤー	
<i>I2C LCD1602</i>	

回路図



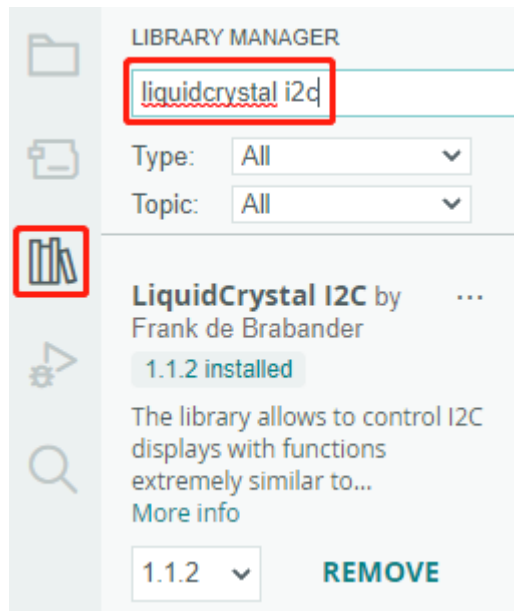
配線図



コード

注釈:

- 3in1-kit\learning_project\5.12.serial_read のパス下の 5.12.serial_read.ino ファイルを開きます。
- または、このコードを **Arduino IDE** にコピーします。
- ここで LiquidCrystal I2C ライブラリを使用しています。 **Library Manager** からインストールできます。



コードが正常にアップロードされた後、シリアルモニタのテキストボックスにテキストを入力すると、LCD に情報が表示されます。

どのように動作するのか？

```
void loop()
{
  // シリアルポートから文字が到着すると...
  if (Serial.available()) {
    // メッセージが全体として到着するのを少し待つ
    delay(100);
    // 画面をクリアする
    lcd.clear();
    // 利用可能な文字すべてを読む
    while (Serial.available() > 0) {
      // 各文字を LCD に表示
      lcd.write(Serial.read());
    }
  }
}
```

- `Serial.available()` は、テキストボックスから何かを入力するときに、入力ストリームに利用可能な文字の数を取得できます。入力には 2 つの終端記号があるため、A を入力すると実際には 3 つの文字があり、AB を入力すると 4 つの文字があります。
- `Serial.read()` は、入力ストリームから最初の文字を取得します。例えば、AB を入力した場合、`Serial.`

`read()` を一度だけ呼び出すと、文字 A を取得します。2 回目の呼び出しで B を取得し、3 回目と 4 回目の呼び出しで 2 つの終端記号を取得します。入力ストリームに文字が利用できない場合、この関数を呼び出すとエラーになります。

要するに、上記の 2 つを組み合わせて、毎回入力されたすべての文字を読み取るために `while` ループを使用するのが一般的です。

```
while (Serial.available() > 0) {  
    Serial.print(Serial.read());  
}
```

ちなみに、入力ストリームから文字を取得するために `Serial.read()` を使用しない場合、入力ストリームからの文字は一緒に積み上げられます。例えば、A の後に AB を入力すると、入力ストリームには 7 文字が蓄積されます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び & 共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.5.13 5.13 インタラプト

センサーを使用するプロジェクトで `delay()` 関数を使用すると、これらのセンサーをトリガーするとプログラムが効果を発揮しないことがあります。これは、`delay` 文によりプログラムが一時停止し、センサーから主制御ボードへの信号を取得できないためです。

この場合、インタラプトを使用できます。インタラプトにより、プログラムがパルスを逃さないようにすることができます。

この章では、アクティブブザーとボタンを使用して、インタラプトの使用プロセスを体験します。

`loop()` 関数内で、`delay(1000)` は秒をカウントするために使用されます。ボタンを ISR 内のブザーの制御に配置することで、遅延による干渉を受けずにタスクをスムーズに完了させることができます。

注釈: ISRs は、他の関数にはないユニークな制限を持つ特別な関数です。ISR にはパラメータを持たせることはできず、何も返さないようにする必要があります。一般的に、ISR はできるだけ短く、高速にする必要があります。複数の ISR を使用するスケッチでは、一度に 1 つしか実行できず、現在の ISR が終了した後に他のインタラプトが優先順位に応じて実行されます。

必要な部品

このプロジェクトでは、以下のコンポーネントが必要です。

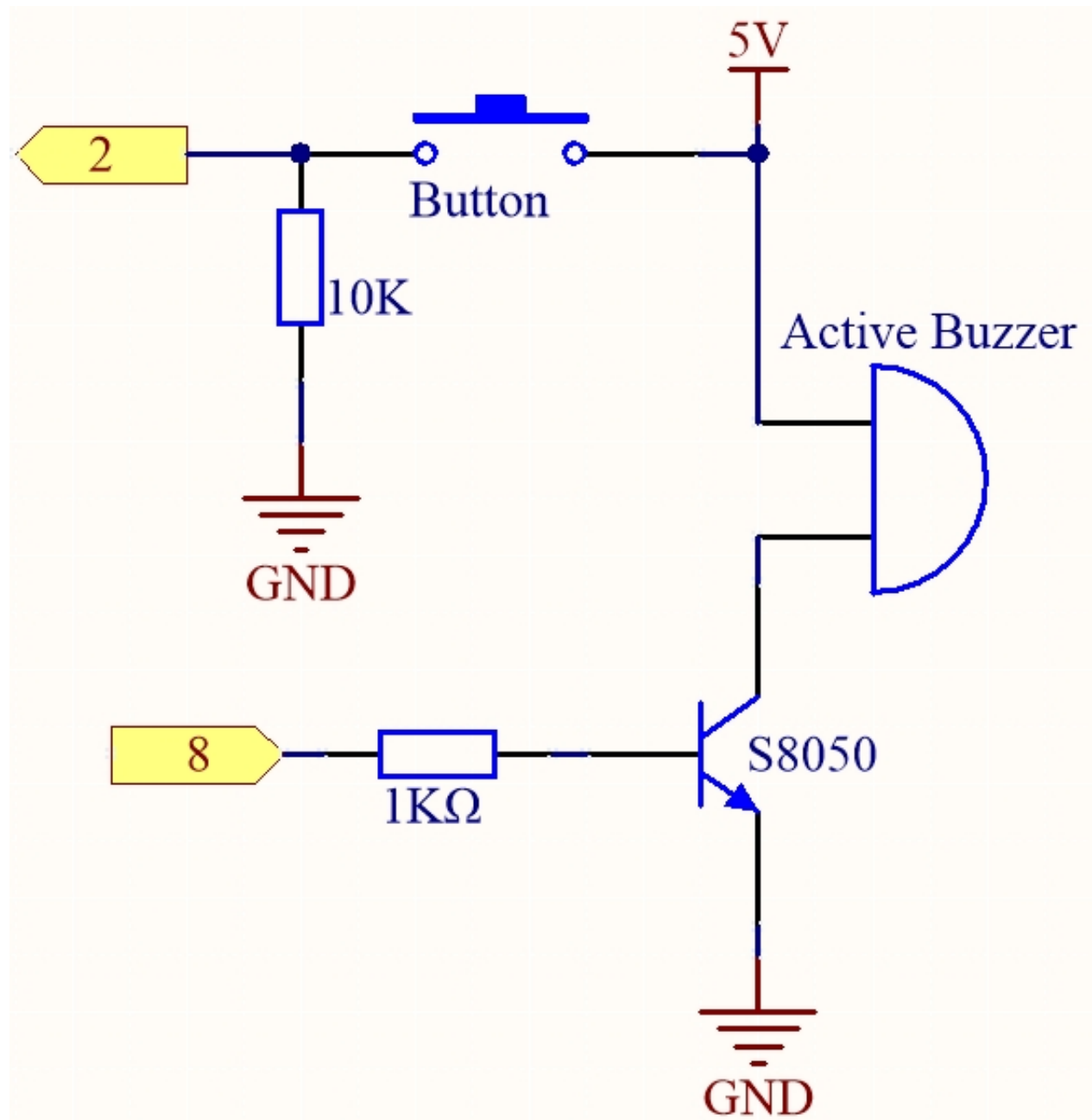
全体のキットを購入すると確かに便利です。リンクはこちら：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

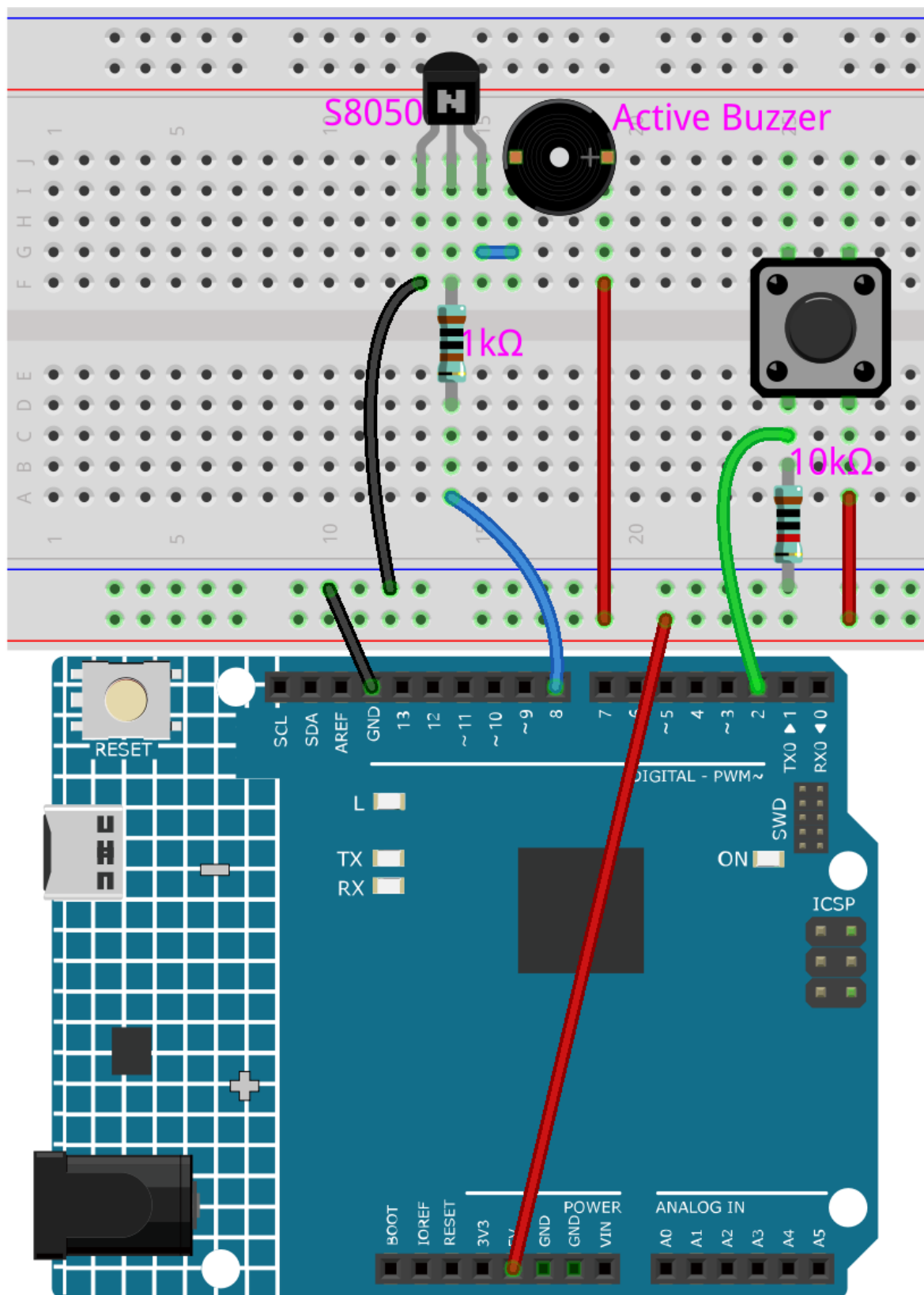
以下のリンクから別々に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
ボタン	
ブザー	-

回路図



配線図



コード

注釈:

- 3in1-kit\learning_project\5.13.interrupt のパスの下の 5.13.interrupt.ino ファイルを開きます。
 - または、このコードを **Arduino IDE** にコピーします。
-

コードが正常にアップロードされると、シリアルモニターをオンにして、毎秒自動で増加する数字が表示されます。ボタンを押すと、ブザーが鳴ります。ボタンで制御されるブザーの機能とタイミングの機能は互いに競合しません。

どのように動作するのか？

- `attachInterrupt(digitalPinToInterrupt(pin), ISR, mode)`: インタラプトを追加します。

構文

```
attachInterrupt(digitalPinToInterrupt(pin), ISR, mode)
```

パラメータ

- `pin`: Arduino のピン番号。 `digitalPinToInterrupt(pin)` を使用して、実際のデジタルピンを特定のインタラプト番号に変換する必要があります。例えば、ピン 3 に接続する場合、第 1 のパラメータとして `digitalPinToInterrupt(3)` を使用します。
- `ISR`: インタラプトが発生したときに呼び出す ISR。この関数はパラメータを持たないこと、何も返さないことが必要です。この関数は、インタラプトサービスルーチンとしても参照されることがあります。
- `mode`: インタラプトをトリガーするタイミングを定義します。4 つの定数が有効な値として事前に定義されています:
 - * `LOW` ピンが低いときにインタラプトをトリガーします。
 - * `CHANGE` ピンの値が変わるたびにインタラプトをトリガーします。
 - * `RISING` ピンが低から高に変わるときにトリガーします。
 - * `FALLING` ピンが高から低に変わるときにトリガーします。

注釈: 異なる主制御ボードでは、インタラプトピンの使用方法が異なる場合があります。あなたのボードでは、ピン 2 とピン 3 だけがインタラプトを使用できます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.5.14 5.14 キャリブレーション

アナログ入力コンポーネント（例：フォトレジスタ、土壌湿度センサなど）を使用する際、読み取り範囲が 0 から 1023 でなく、0 から 800 や 600 から 1000 のような範囲になることがあります。これは、通常の使用ではこれらのデバイスの限界に達することができないためです。

このような場合、センサ入力をキャリブレーションする技術が役立ちます。起動時に、制御ボードがセンサの読み取りを 5 秒間行い、最高値と最低値を記録します。この 5 秒間の読み取りが、サイクル中の読み取りの最小値と最大値を定義します。

このプロジェクトでは、上記のキャリブレーション技術を使用して、フォトレジスタとパッシブブザーを使用し、**テレミン** のようなゲームを実装します。

注釈：**テレミン** は、物理的な接触を必要としない電子楽器です。プレイヤーの手の位置を感知して、異なる音を生成します。

必要な部品

このプロジェクトには以下のコンポーネントが必要です。

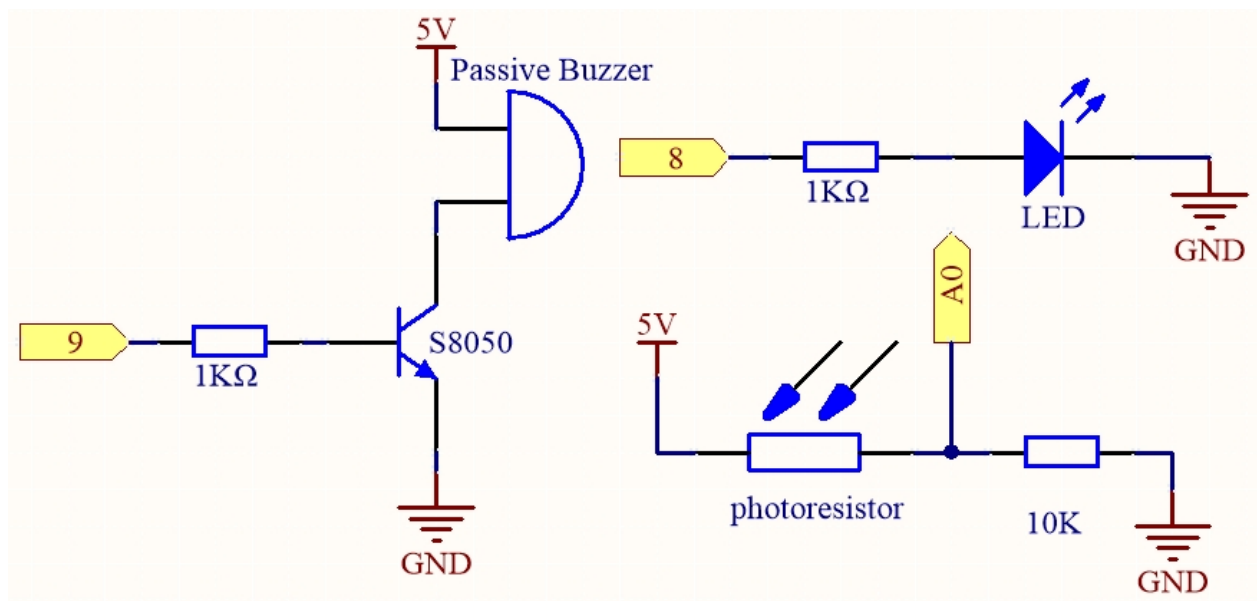
一式のキットを購入すると非常に便利です。リンクはこちら：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

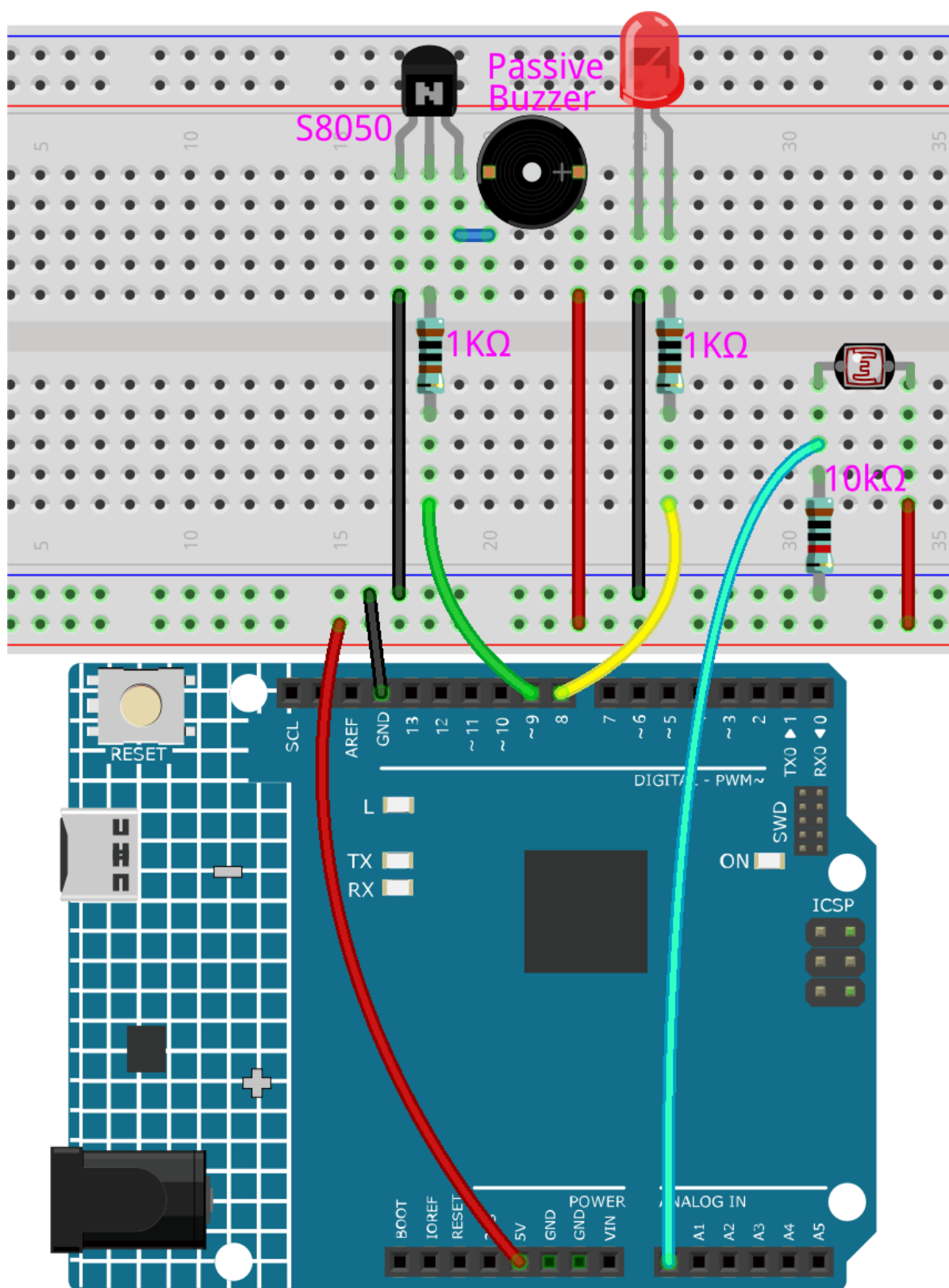
以下のリンクから、個別に購入することも可能です。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
ブザー	
<i>LED</i>	
フォトレジスタ	

回路図



配線図



コード

注釈:

- 3in1-kit\learning_project\5.14.calibration のパスの下で 5.14.calibration.ino ファイルを開きます。
 - または、このコードを **Arduino IDE** にコピーします。
-

コードが正常にアップロードされると、LED が点灯し、フォトレジスタの検出範囲をキャリブレーションするための 5 秒間が与えられます。これは、毎回使用する際に異なる光環境になる可能性があるためです（例：正午と夕方では光の強度が異なる）。

このとき、フォトレジスタの上で手を上下に振る必要があります。手の動きの範囲がこの楽器の演奏範囲にキャリブレーションされます。

5 秒後、LED が消灯し、フォトレジスタの上で手を振ることで演奏することができます。

どのように動作するのか？

1. すべてのコンポーネントの初期値とピンを設定する。

```
const int buzzerPin = 9;
const int ledPin = 8;
const int photocellPin = A0; //フォトレジスタは A2 に接続

int lightLow = 1023;
int lightHigh = 0;

int sensorValue = 0; // センサからの読み取り値
int pitch = 0; // センサ値を LED 'bars' に変換

unsigned long previousMillis = 0;
const long interval = 5000;
```

2. setup() 内でキャリブレーションプロセスを設定する。

```
void setup()
{
    pinMode(buzzerPin, OUTPUT); // ブザーを出力として設定
    pinMode(ledPin, OUTPUT); // LED ピンを出力として設定

    /* フォトレジスタの最大値 & 最小値をキャリブレートする */
    previousMillis = millis();
```

(次のページに続く)

(前のページからの続き)

```

digitalWrite(ledPin, HIGH);
while (millis() - previousMillis <= interval) {
    sensorValue = analogRead(photocellPin);
    if (sensorValue > lightHigh) {
        lightHigh = sensorValue;
    }
    if (sensorValue < lightLow) {
        lightLow = sensorValue;
    }
}
digitalWrite(ledPin, LOW);
}

```

ワークフローは以下の通りです。

- 5000ms の間隔で millis() を使ってタイミングを取る。

```

previousMillis = millis();
...
while (millis() - previousMillis <= interval) {
    ...
}

```

- この 5 秒間、フォトレジスタの周りで手を振ると、検出された光の最大値と最小値が記録され、それぞれ lightHigh と lightLow に割り当てられる。

```

sensorValue = analogRead(photocellPin);
if (sensorValue > lightHigh) {
    lightHigh = sensorValue;
}
if (sensorValue < lightLow) {
    lightLow = sensorValue;
}

```

3. これで、このテルミンを演奏することができます。フォトレジスタの値を sensorValue に読み取り、それを小さい範囲から大きい範囲にマップして、ブザーの周波数として使用します。

```

void loop()
{
    /* 演奏 */
}

```

(次のページに続く)

(前のページからの続き)

```
sensorValue = analogRead(photocellPin); //A0 の値を読み取る
pitch = map(sensorValue, lightLow, lightHigh, 50, 6000); // ブザーの周波数にマップ
する
if (pitch > 50) {
    tone(buzzerPin, pitch, 20);
}
delay(10);
}
```

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.5.15 5.15 EEPROM

EEPROM はメモリであり、メインコントロールボードをオフにしてもその中のデータは消去されません。これを使用して、いくつかのデータを記録し、次回電源を入れたときにそれを読み取ることができます。

例として、毎日のジャンプロープの回数を記録するスポーツカウンタを作成することができます。

また、1つのプログラムでデータを書き込み、別のプログラムでそれを読み取ることもできます。例えば、車のプロジェクトに取り組んでいるとき、2つのモータの速度が一致しない場合、モータ速度の補正値を記録するキャリブレーションプログラムを書くことができます。

必要な部品

このプロジェクトでは、以下のコンポーネントが必要です。

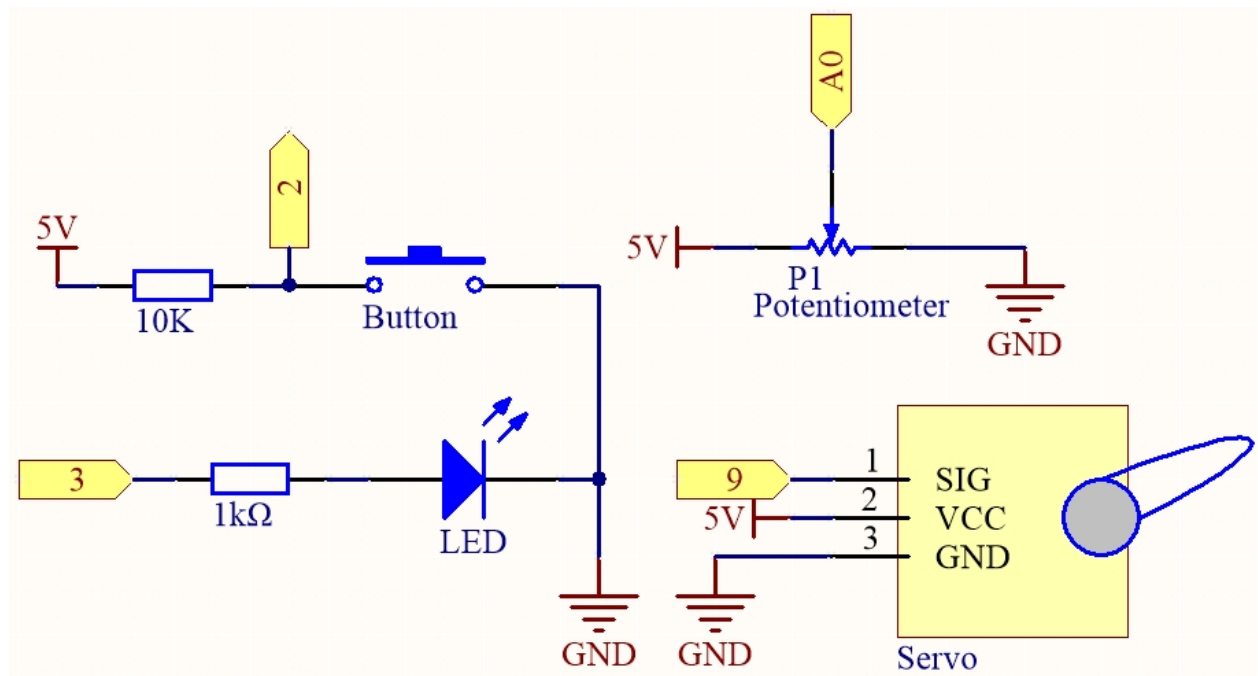
キット全体を購入するのは確かに便利です、リンクはこちら：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

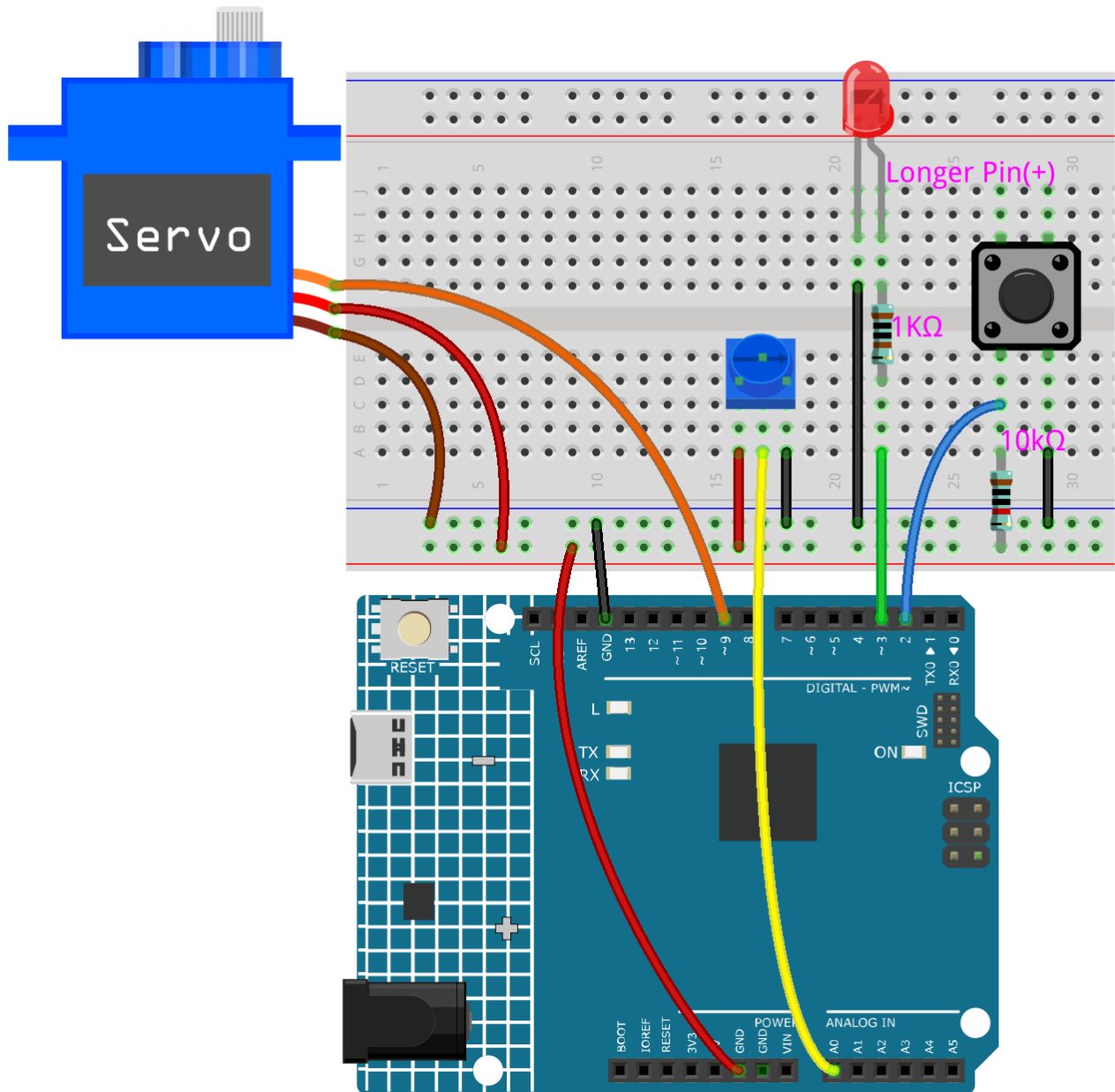
以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
<i>LED</i>	
サーボ	
ボタン	
ポテンシオメーター	

回路図



配線図



コード

注釈:

- 3in1-kit\learning_project\5.15.eeproom のパスの下にある 5.15.eeproom.ino ファイルを開きます。
- または、このコードを **Arduino IDE** にコピーします。

この回路を使用するには、ボタンを押して記録を開始し、ポテンショメータを通じて所望の情報を入力します。これで、ボードはあなたのアクションを無限に繰り返します（そして、それぞれの繰り返しで LED が点滅します）

新しいアクションを記録するために再びボタンを押すまで。resolution と recordTime の値を変更することで、記録される時間の長さを変更することもできます。

どのように動作するのか？

1. EEPROM.h ライブラリをインポートし、EEPROM メモリを初期化します。

```
...
#include <EEPROM.h> //記録された値を保存するために使用

...
float resolution = 1000; //EEPROM.length() より小さくする必要があります
float recordTime = 5; //遅延時間
bool recording = false;
...
```

/EEPROM.length() より小さくする必要があります の点に注意してください。setup() でボードの EEPROM のメモリを印刷します。あなたのボードの場合は 1024 であるはずです。異なるボードを使用している場合、変数 resolution の値を変更できます。

2. あなたのボードの EEPROM メモリを印刷します。

```
void setup() {
    ...
    Serial.begin(9600);
    //Serial.println(EEPROM.length());
}
```

ボードの EEPROM メモリのサイズを見つけるために、Serial.println(EEPROM.read(i)) の行のコメントを外します。これにより、シリアルモニタに EEPROM のサイズが印刷され、変数 resolution の値をそれに応じて変更できます。

3. ボタンの押下が検出されると、録音が始まり、必要な情報がポテンショメータを介して入力されます。ボタンを再度押すまで、ボードはあなたのアクションを無限に繰り返します（そして、LED がそれぞれの繰り返して点滅します）。

```
void loop() {
    if (recording == true) { //record
        for (int i = 1; i <= resolution; i++) {
            digitalWrite(ledPin, HIGH); //ライトステータス LED
            int val = map(analogRead(A0), 0, 1023, 0, 180);
            EEPROM.write(i, val);
            //Serial.println(EEPROM.read(i));
        }
    }
}
```

(次のページに続く)

(前のページからの続き)

```

        myServo.write(val);
        delay(recordTime);
    }
    digitalWrite(ledPin, LOW); // ステータス LED をオフにする
    delay(1000); // 人のために時間を与える
    recording = false;
}
else {
    for (int i = 1; i <= resolution; i++) { // playback
        if (digitalRead(buttonPin) == 0) { // 再生を停止し、新しい値を記録します
            recording = true;
            break;
        }
        int readval = EEPROM.read(i);
        myServo.write(readval);
        //Serial.println(readval);
        delay(recordTime);
    }
    digitalWrite(ledPin, HIGH); // 新しいリピートを表示する
    delay(100);
    digitalWrite(ledPin, LOW);
}
}
}

```

- ボタンが押されると、変数 recording を true にします。
- 変数 recording が true の場合、メモリ範囲でのアクションの録音を開始します。
- ポテンショメータの値を読み取り、それを 0-180 にマッピングして EEPROM に保存し、サーボの回転を制御します。
- 録音の開始時に LED が点灯し、終了時に消灯します。
- LED の短い点滅で記録されたアクションを繰り返します。

4. EEPROM ライブラリについて。

以下は、その機能の一部です。

- write(address,value): EEPROM にバイトを書き込みます。
 - address: 0 から始まる書き込む場所 (int)
 - value: 0 から 255 までの書き込む値 (byte)

- EEPROM の書き込みには 3.3 ms かかります。EEPROM メモリの指定された寿命は 100,000 回の書き込み/消去サイクルなので、頻繁に書き込む際は注意が必要です。
- `Read(address)`: EEPROM からバイトを読み取ります。書き込まれていない場所の値は 255 です。
- `update(address, value)`: EEPROM にバイトを書き込みます。書き込まれる値は、同じアドレスに既に保存されている値と異なる場合のみです。
 - EEPROM の書き込みには 3.3 ms かかります。データが頻繁に変更されない場合、この関数を `write()` の代わりに使用することで、サイクルを節約できます。
- `EEPROM.put(address, data)`: EEPROM に任意のデータ型やオブジェクトを書き込みます。
 - `address`: 0 から始まる読み取る場所 (int)
 - `data`: 読み取るデータ、プリミティブ型 (例: float) やカスタム構造体が可能。
 - この関数は `EEPROM.update()` を使用して書き込みを行うので、値が変わらない場合は書き込みを行いません。
- `EEPROM.get(address, data)`: EEPROM から任意のデータ型やオブジェクトを読み取ります。
 - `address`: 0 から始まる読み取る場所 (int)
 - `data`: 読み取るデータ、プリミティブ型 (例: float) やカスタム構造体が可能。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.6 6. 面白いプロジェクト

この章では、ほとんどのプログラムが現実とどのように相互作用するかの基本的なロジックを示す例をいくつか紹介します。これにより、Arduino のプログラミングに慣れる手助けとなります。創造的なアイデアを思いついたとき、プログラミングはもうあなたにとって難しいものではありません。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.6.1 6.1 光感知配列

フォトレジスタまたはフォトセルは、光によって変動する抵抗器です。フォトレジスタの抵抗は、入射光の強度が増加するにつれて減少します。言い換えれば、フォトコンダクティビティを示します。フォトレジスタは、光感知検出回路や、光と闇で動作するスイッチング回路に適用することができます。

フォトレジスタの抵抗は、入射光の強度によって変わります。光の強度が高くなると、抵抗は減少します。光の強度が低くなると、抵抗は増加します。この実験では、光の強度を示すために 8 つの LED を使用します。光の強度が高いほど、多くの LED が点灯します。光の強度が十分に高い場合、すべての LED が点灯します。光がない場合、すべての LED は消灯します。

必要な部品

このプロジェクトでは、以下の部品が必要です。

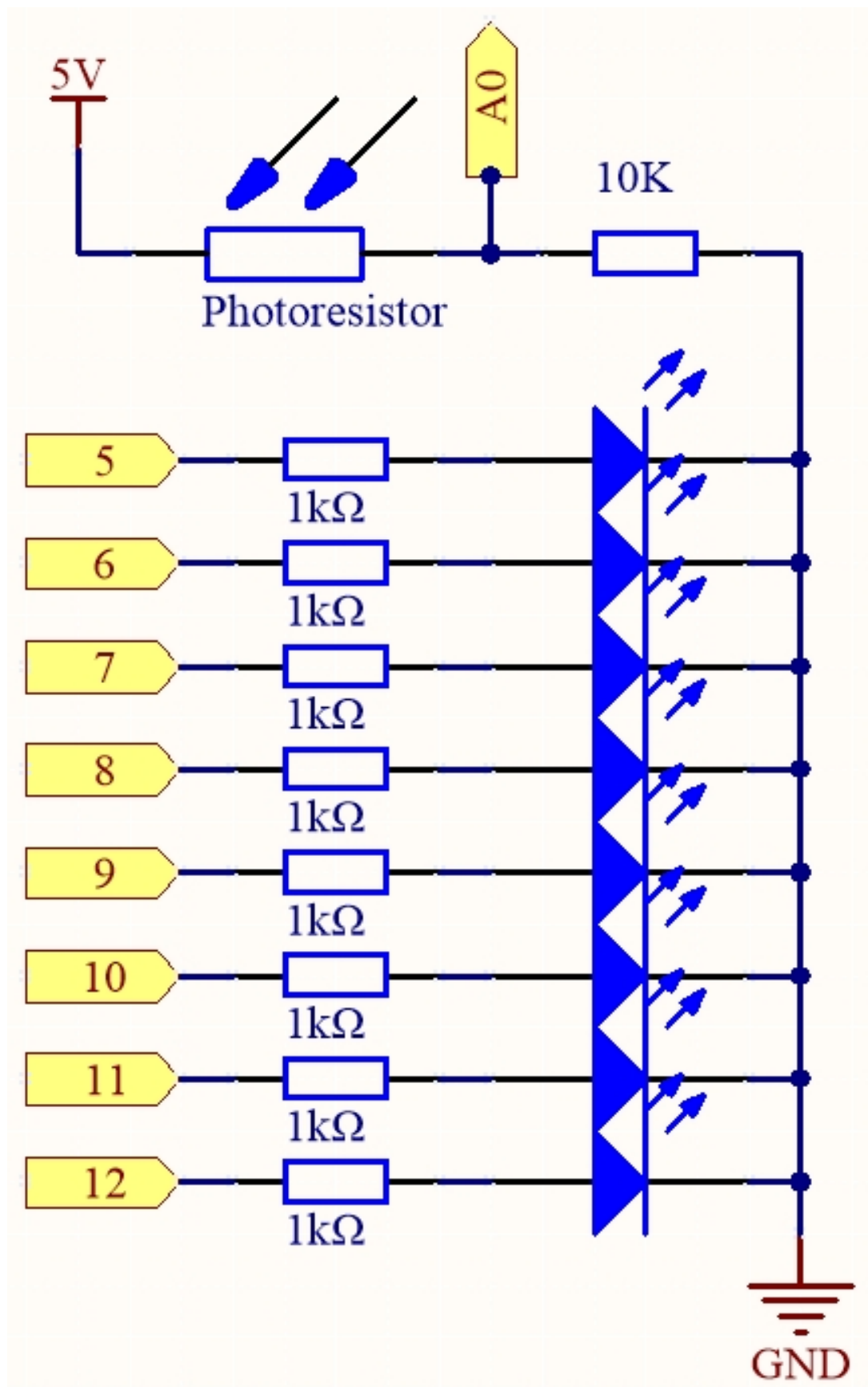
一式を購入するのは非常に便利です、リンクはこちらです:

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

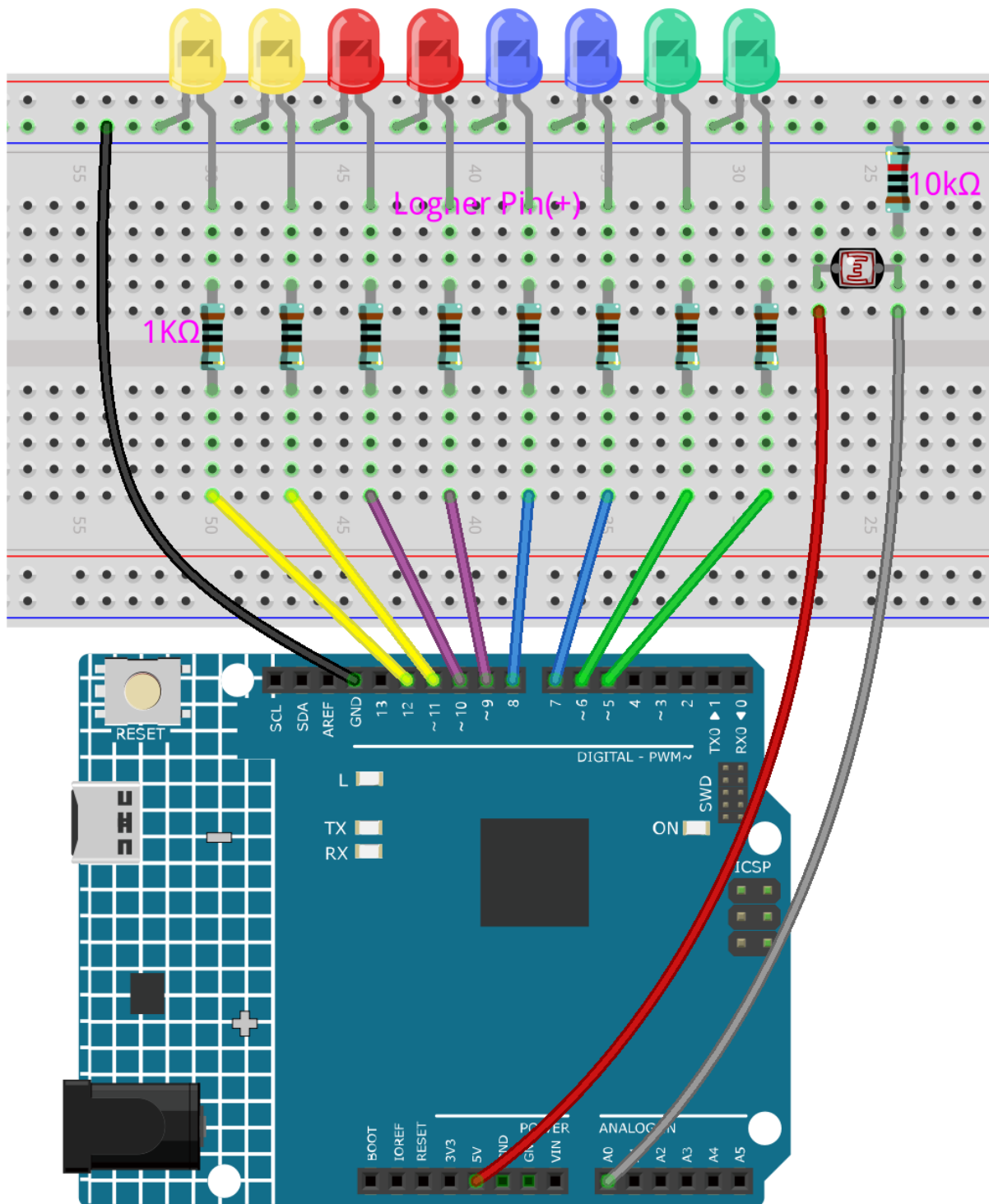
以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
<i>LED</i>	
フォトレジスタ	

回路図



配線図



コード

注釈:

- 3in1-kit\learning_project\6.1.light_control_led のパスの下に 6.1.light_control_led.ino ファイルを開きます。
- または、このコードを **Arduino IDE** にコピーしてください。

フォトレジスタに光を当てると、いくつかの LED が点灯します。さらに光を当てると、さらに多くの LED が点灯します。暗い環境に置くと、すべての LED が消えます。

どのように動作するのか？

```
void loop()
{
    sensorValue = analogRead(photocellPin); // A0 の値を読み取る
    ledLevel = map(sensorValue, 300, 1023, 0, NbrLEDs); // LED の数にマッピングする
    for (int led = 0; led < NbrLEDs; led++)//
    {
        if (led < ledLevel ) // led が ledLevel よりも小さい場合、以下のコードを実行します。
        {
            digitalWrite(ledPins[led], HIGH); // レベル以下のピンをオンにする
        }
        else
        {
            digitalWrite(ledPins[led], LOW); // レベル以上のピンをオフにする
        }
    }
}
```

map() 関数を使用して、フォトレジスタの値を 8 つの LED にマッピングすることができます。例えば、sensorValue が 560 の場合、ledLevel は 4 となります。この時点で、ledPins[0] から ledPins[4] が点灯し、ledPins[5] から ledPins[7] が消灯することになります。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。

- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.6.2 6.2 デジタルサイコロ

このプロジェクトでは、ボタン、7 セグメント表示、および 74hc595 を使用して電子サイコロを作成します。ボタンを押すたびに、1 から 6 までのランダムな数字が生成され、7 セグメント表示に表示されます。

必要な部品

このプロジェクトに必要な部品は以下のとおりです。

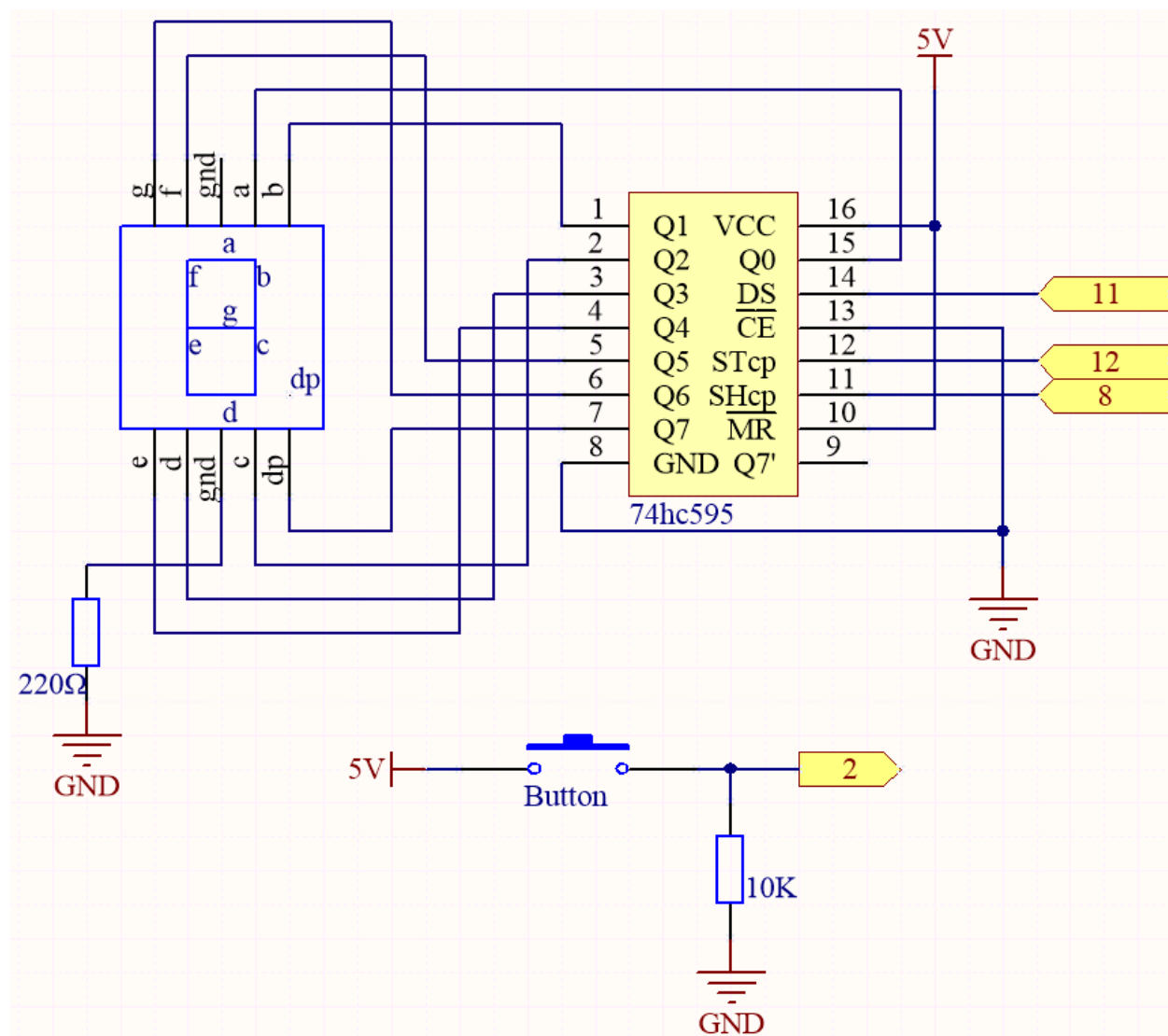
一式を購入するのは確かに便利です。リンクはこちら：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
ボタン	
<i>74HC595</i>	
7 セグメントディスプレイ	

回路図



配線図

注釈:

- 256

- または、このコードを **Arduino IDE** にコピーしてください。

コードが正常にアップロードされると、7 セグメント表示は 0-7 を高速でスクロールして表示しますが、ボタンを押すとランダムな数字を表示し、スクロールが停止します。再びボタンを押すと、スクロール表示が再開されます。

どのように動作するのか？

このプロジェクトは、[5.10 シフトアウト \(セグメントディスプレイ\)](#) をベースに、ボタンを使用して 7 セグメント表示のスクロール表示を開始/一時停止します。

1. 各ピンを初期化し、ボタンの値を読み取ります。

```
void setup ()
{
    ...
    attachInterrupt(digitalPinToInterrupt(buttonPin), rollDice, FALLING);
}
```

- ここでは割り込みを使用してボタンの状態を読み取ります。buttonPin のデフォルト値は低いですが、ボタンを押すと低から高に変わります。
 - rollDice は割り込みがトリガーされたときに呼び出される関数を表し、変数 state の値を切り替えるために使用されます。
 - FALLING は、buttonPin が低から高になったときに割り込みがトリガーされることを意味します。
2. 変数 state が 0 の場合、関数 showNumber() が呼び出されて、7 セグメント表示が 1 から 7 の間でランダムに数字を表示するようになります。

```
void loop()
{
    if (state == 0) {
        showNumber((int)random(1, 7));
        delay(50);
    }
}
```

3. rollDice() 関数について。

```
void rollDice() {
    state = !state;
}
```

この関数が呼び出されると、state の値が切り替わります。前回は 1 で、今回は 0 の場合など。

4. showNumber() 関数について。

```
void showNumber(int num) {  
    digitalWrite(STcp, LOW); //ST_CP を GND に接続し、送信中は常に低く保持  
    shiftOut(DS, SHcp, MSBFIRST, dataArray[num]);  
    // ラッチ ピンを High に戻してチップに信号を送ります  
    // 情報を聞く必要がなくなりました  
    digitalWrite(STcp, HIGH); //ST_CPST_CP を上げてデータを保存  
}
```

これは、プロジェクト 5.10 シフトアウト (セグメントディスプレイ) の loop() 内のコードを関数 showNumber() に入れたものです。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.6.3 6.3 高温アラーム

次に、サーミスター、プッシュボタン、ポテンショメータ、LCD を使用して、高温アラームデバイスを作成します。LCD1602 は、サーミスターで検出される温度と、ポテンショメータを使用して調整できる高温閾値を表示します。閾値は同時に EEPROM に保存されるため、現在の温度が閾値を超えると、ブザーが鳴ります。

必要な部品

このプロジェクトには、以下の部品が必要です。

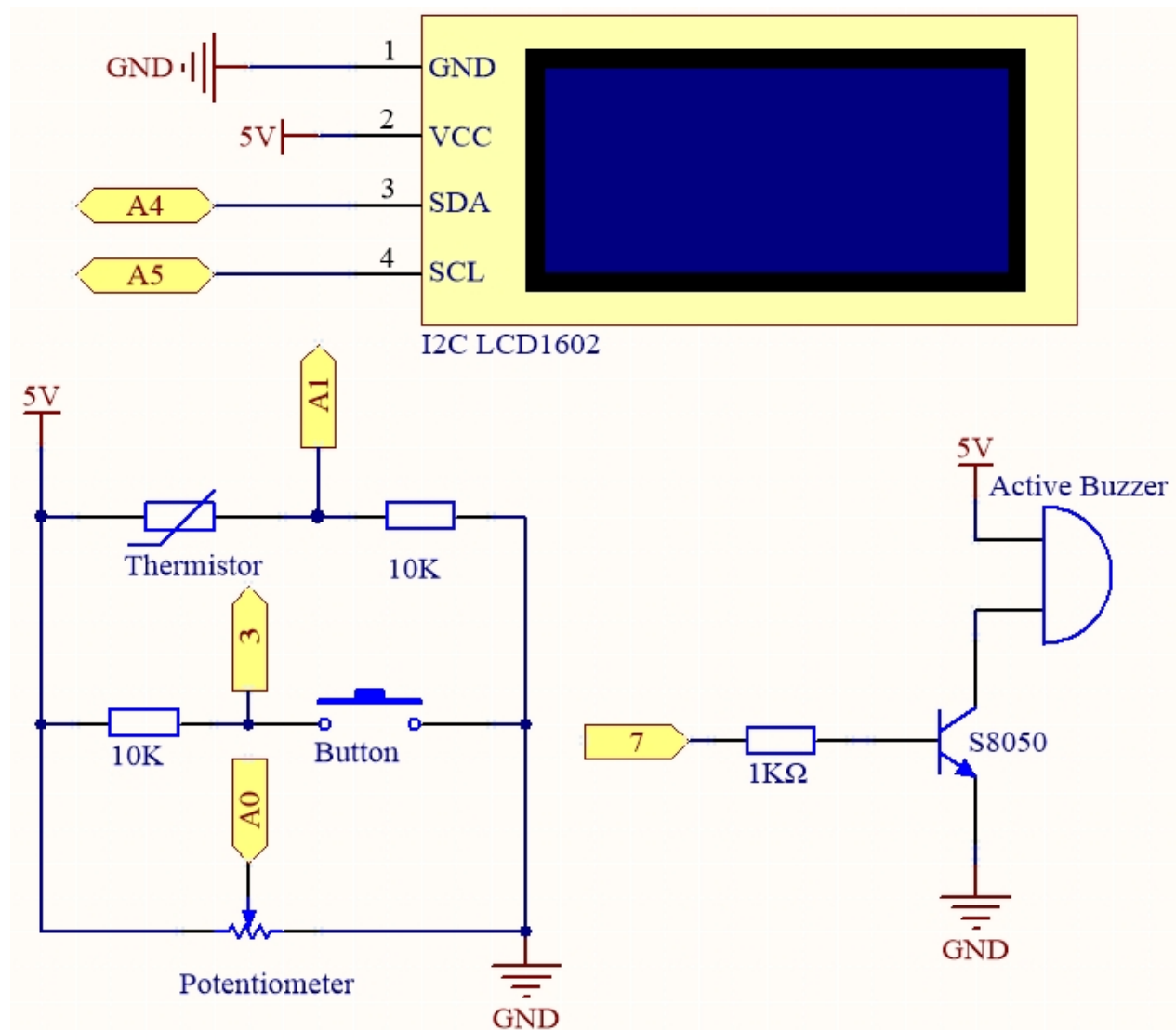
一式を購入するのは確かに便利です。こちらがリンクです:

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

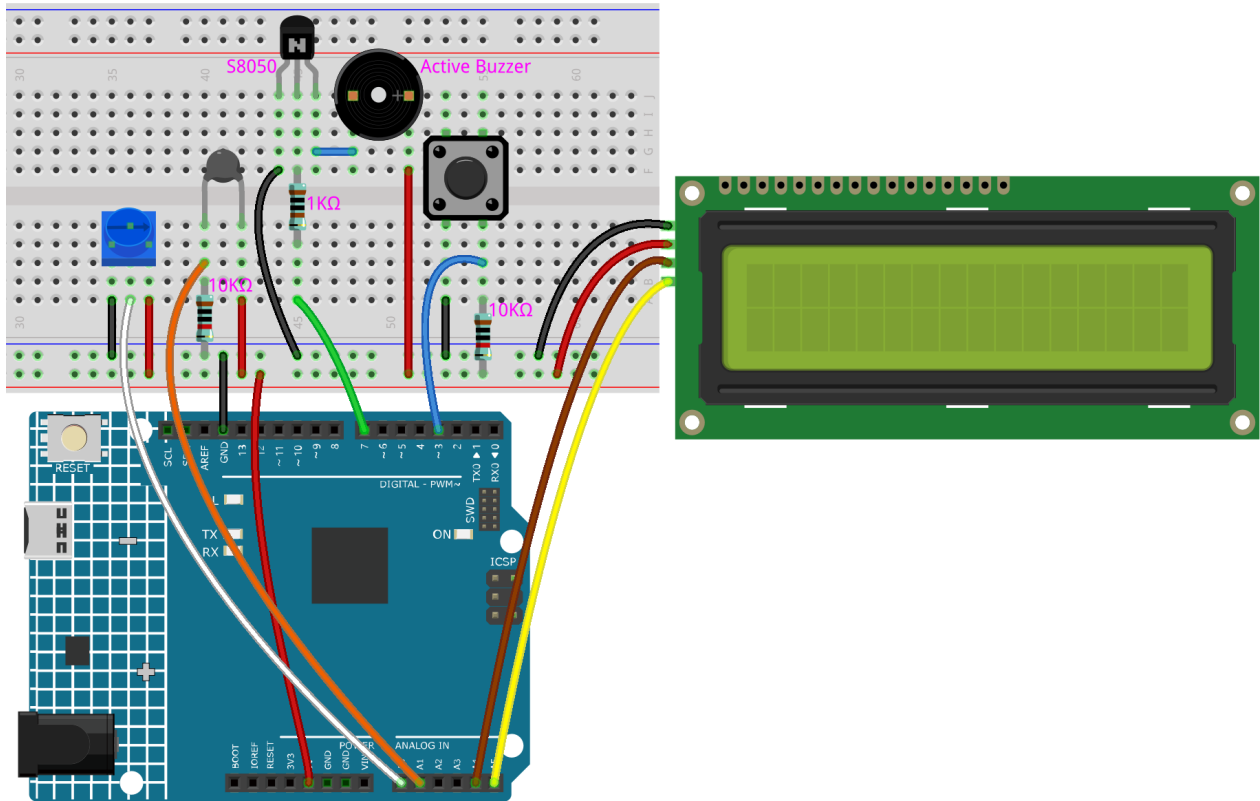
以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
ブザー	-
ボタン	
<i>I2C LCD1602</i>	
サーミスタ	
ポテンショメーター	

回路図



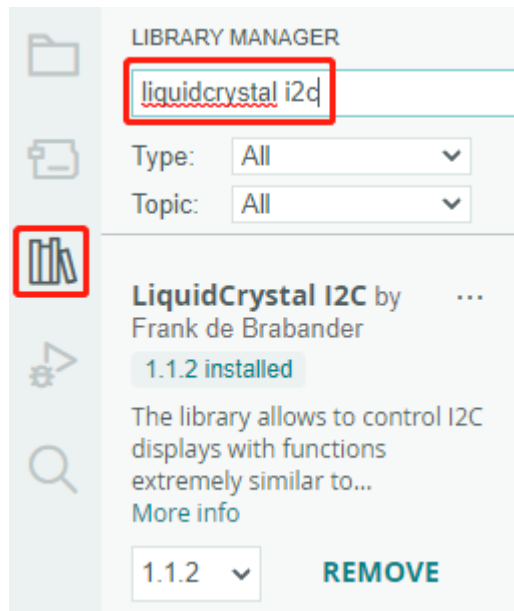
配線図



コード

注釈:

- ファイル 6.3.high_tem_alarm.ino は、3in1-kit\learning_project\6.3.high_tem_alarm のパスの下で直接開くことができます。
- または、このコードを Arduino IDE にコピーしてください。
- ここでは LiquidCrystal I2C ライブラリが使用されています。 **Library Manager** からインストールできます。



コードが正常にアップロードされた後、LCD1602 はサーミスターで検出される温度と高温閾値を表示します。この閾値はポテンショメータを使用して調整することができます。閾値は同時に EEPROM に保存されるため、現在の温度が閾値を超えると、ブザーが鳴ります。

注釈: コードと配線が問題ないのに、LCD が内容を表示しない場合は、背面のポテンショメータを回して調整してください。

どのように動作するのか？

1. ボタン、ブザー、I2C LCD1602 の初期化を行い、EEPROM の値を読み取ります。また、ボタンの状態を読み取るために、ここでは割り込みも使用されています。

```
void setup()
{
    pinMode(buzzerPin, OUTPUT);
    pinMode(buttonPin, INPUT);
    lcd.init();
    lcd.backlight();
    upperTem = EEPROM.read(0);
    delay(1000);
    attachInterrupt(digitalPinToInterrupt(buttonPin), buttonState, FALLING);
}
```

- この割り込みはボタンの状態を読み取るために使用されます。ボタンが押されると、buttonPin

が低から高に変わります。

- 割り込みが発生すると、buttonState 関数が呼び出され、変数 state の値が切り替えられます。
- FALLING は、buttonPin が低から高になるときに割り込みが発生することを意味します。

2. 高温のしきい値を設定するため、メインプログラム内で state が 1 の場合（ボタンが押されると 0 と 1 の間で切り替わる）に upperTemSetting() 関数が呼び出され、そうでなければ monitoringTemp() が呼び出され、現在の温度と設定されたしきい値が表示されます。

```
void loop()
{
    if (state == 1)
    {
        upperTemSetting();
    }
    else {
        monitoringTemp();
    }
}
```

3. upperTemSetting() 関数について。

```
void upperTemSetting()
{
    int setTem = 0;

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Adjusting...");
    lcd.setCursor(0, 1);
    lcd.print("Upper Tem: ");

    while (1) {
        lcd.setCursor(11, 1);
        setTem = map(analogRead(potPin), 0, 1023, 0, 100);
        lcd.print(setTem);
        if (state == 0)
        {
            EEPROM.write(0, setTem);
            upperTem = setTem;
            lcd.clear();
        }
    }
}
```

(次のページに続く)

(前のページからの続き)

```

        return;
    }
}
}

```

- この関数を使うとしきい値を設定できます。この関数に入ると、LCD1602 に現在のしきい値が表示され、これはポテンショメータを使って変更することができます。このしきい値は EEPROM に保存され、ボタンが再び押されると終了します。

4. monitoringTemp() 関数について。

```

void monitoringTemp()
{
    long a = analogRead(temPin);
    float tempC = beta / (log((1025.0 * 10 / a - 10) / 10) + beta / 298.0) - 273.
    ↪0;
    float tempF = 1.8 * tempC + 32.0;
    lcd.setCursor(0, 0);
    lcd.print("Temp: ");
    lcd.print(tempC);
    lcd.print(char(223));
    lcd.print("C  ");
    lcd.setCursor(0, 1);
    lcd.print("Upper: ");
    lcd.print(upperTem);
    lcd.print(char(223));
    lcd.print("C  ");
    delay(300);
    if (tempC >= upperTem)
    {
        digitalWrite(buzzerPin, HIGH);
        delay(50);
        digitalWrite(buzzerPin, LOW);
        delay(10);
    }
    else
    {
        digitalWrite(buzzerPin, LOW);
    }
}

```

- この関数を使用すると、温度を表示し、アラームを設定することができます。
- サーミスタの値が読み取られ、それからセ氏温度に変換され、LCD1602 に表示されます。
- 設定されたしきい値も LCD に表示されます。
- 現在の温度がしきい値よりも高い場合、ブザーがアラームを鳴らします。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.6.4 6.4 駐車補助

科学技術の発展に伴い、多くの高度な技術製品が車に取り付けられています。その中で、バックアップ補助システムは一つです。この章では、超音波モジュール、LCD、LED、ブザーを使用して、シンプルな超音波駐車補助システムを作成します。

必要な部品

このプロジェクトでは、以下の部品が必要です。

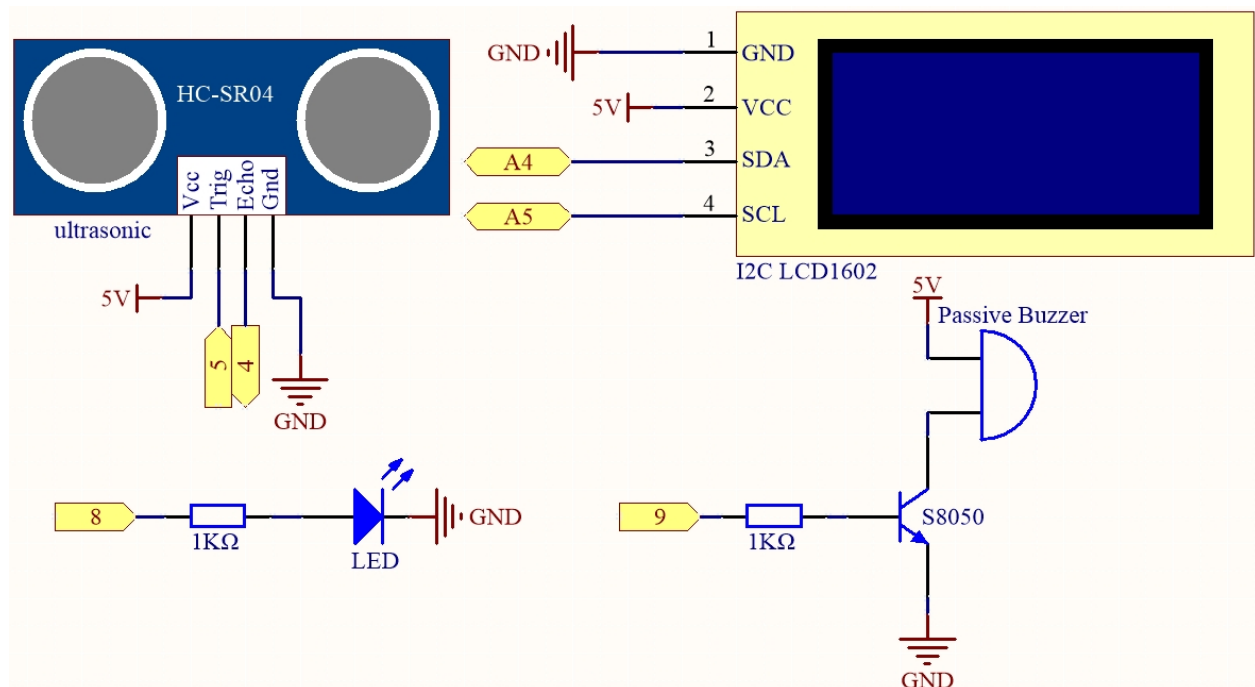
全セットを購入すると非常に便利です。以下がリンクです:

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

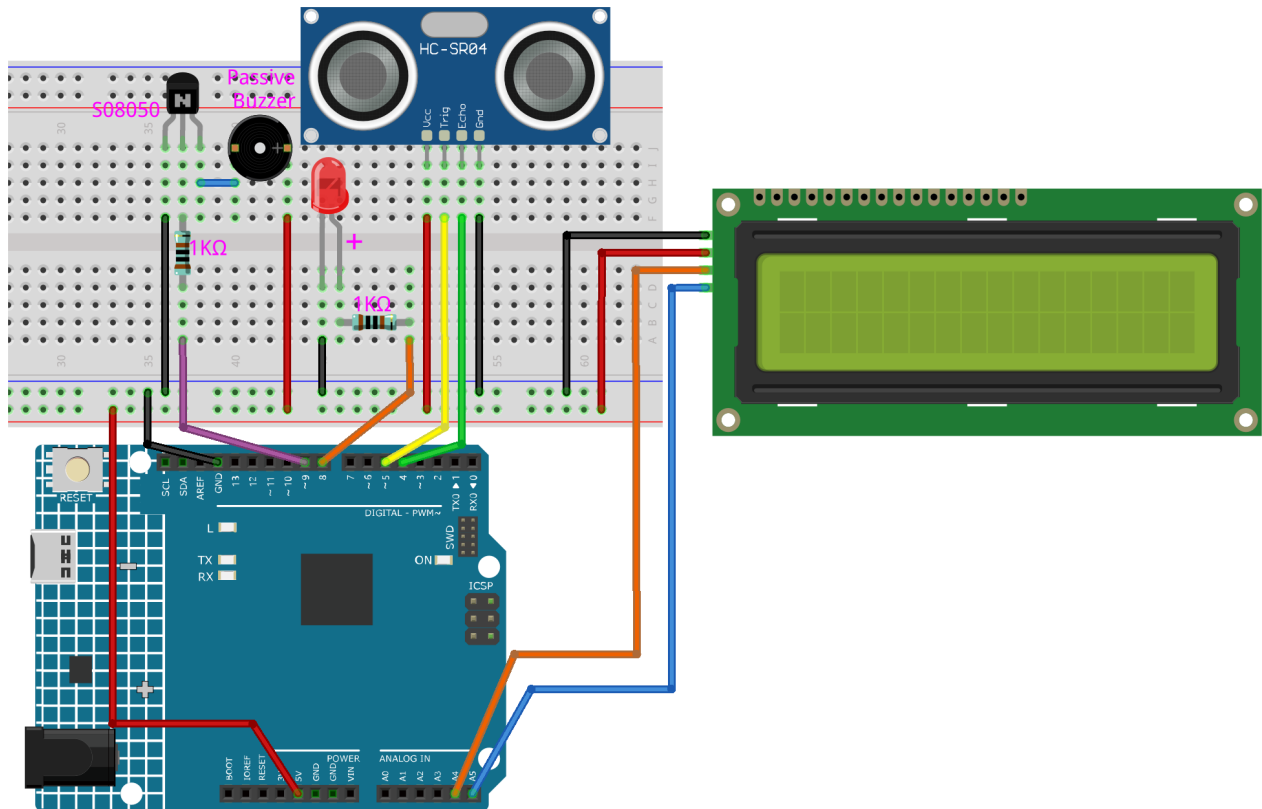
下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
<i>LED</i>	
ブザー	
<i>I2C LCD1602</i>	
超音波モジュール	

回路図



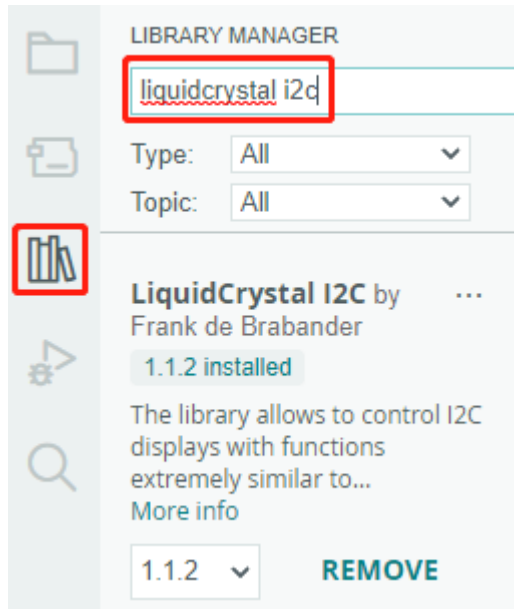
配線図



コード

注釈:

- ファイル 6.4_reversingAid.ino を 3in1-kit\learning_project\6.4_reversingAid のパスで直接開くことができます。
- または、このコードを Arduino IDE にコピーしてください。
- ここでは LiquidCrystal I2C ライブラリを使用しています。 **Library Manager** からインストールできます。



コードが正常にアップロードされると、LCD に検出された現在の距離が表示されます。そして、距離に応じてブザーの音の頻度が変わります。

注釈: コードと配線が正しいのに、LCD に内容が表示されない場合は、裏面のポテンショメータを回して調整してください。

どのように動作するのか？

このコードは、物体との距離を測定し、LCD ディスプレイとブザーを通じてフィードバックを提供するシンプルな距離測定デバイスを作成するのに役立ちます。

loop() 関数はプログラムの主要なロジックを含み、継続的に実行されます。loop() 関数を詳しく見てみましょう。

1. 距離を読み取り、パラメータを更新するループ

loop 内のコードは、超音波モジュールで測定された距離を最初に読み取り、距離に基づいて間隔のパラメータを更新します。

```
// 距離を更新
distance = readDistance();

// 距離に基づいて間隔を更新
if (distance <= 10) {
    intervals = 300;
```

(次のページに続く)

(前のページからの続き)

```

} else if (distance <= 20) {
    intervals = 500;
} else if (distance <= 50) {
    intervals = 1000;
} else {
    intervals = 2000;
}

```

2. ビープ音の時間かどうかを確認する

コードは、現在の時間と前回のビープ音の時間との差を計算し、その差が間隔の時間以上の場合、ブザーをトリガーして、前回のビープ音の時間を更新します。

```

unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= intervals) {
    Serial.println("Beeping!");
    beep();
    previousMillis = currentMillis;
}

```

3. LCD ディスプレイを更新

コードは、LCD ディスプレイをクリアし、次に "Dis:" と現在の距離（センチメートル）を 1 行目に表示します。

```

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Dis: ");
lcd.print(distance);
lcd.print(" cm");

delay(100);

```

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。

- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.6.5 6.5 反応ゲーム

私たちの身体には様々な反応時間があります。例として、聴覚の反応時間、視覚の反応時間、触感の反応時間などがあります。

反応時間は日常生活において多くの影響を及ぼします。例えば、運転中に反応時間が普段より遅いと、重大な結果を招く可能性があります。

このプロジェクトでは、3つのボタンと2つのLEDを使用して、私たちの視覚的な反応時間を計測します。

Arduinoのシリアルモニターは「waiting...」というメッセージを表示します。Ready ボタンを押すと、ランダムな時間が経過した後に2つのLEDのうちの1つがランダムに点灯します。被験者は対応するボタンをできるだけ早く押すことが重要です。ArduinoはLEDが点灯するタイミングと、人が対応するボタンを押すタイミングとの時間差を記録し、Arduinoのシリアルモニターに計測した反応時間を出力します。

必要な部品

このプロジェクトには以下の部品が必要です。

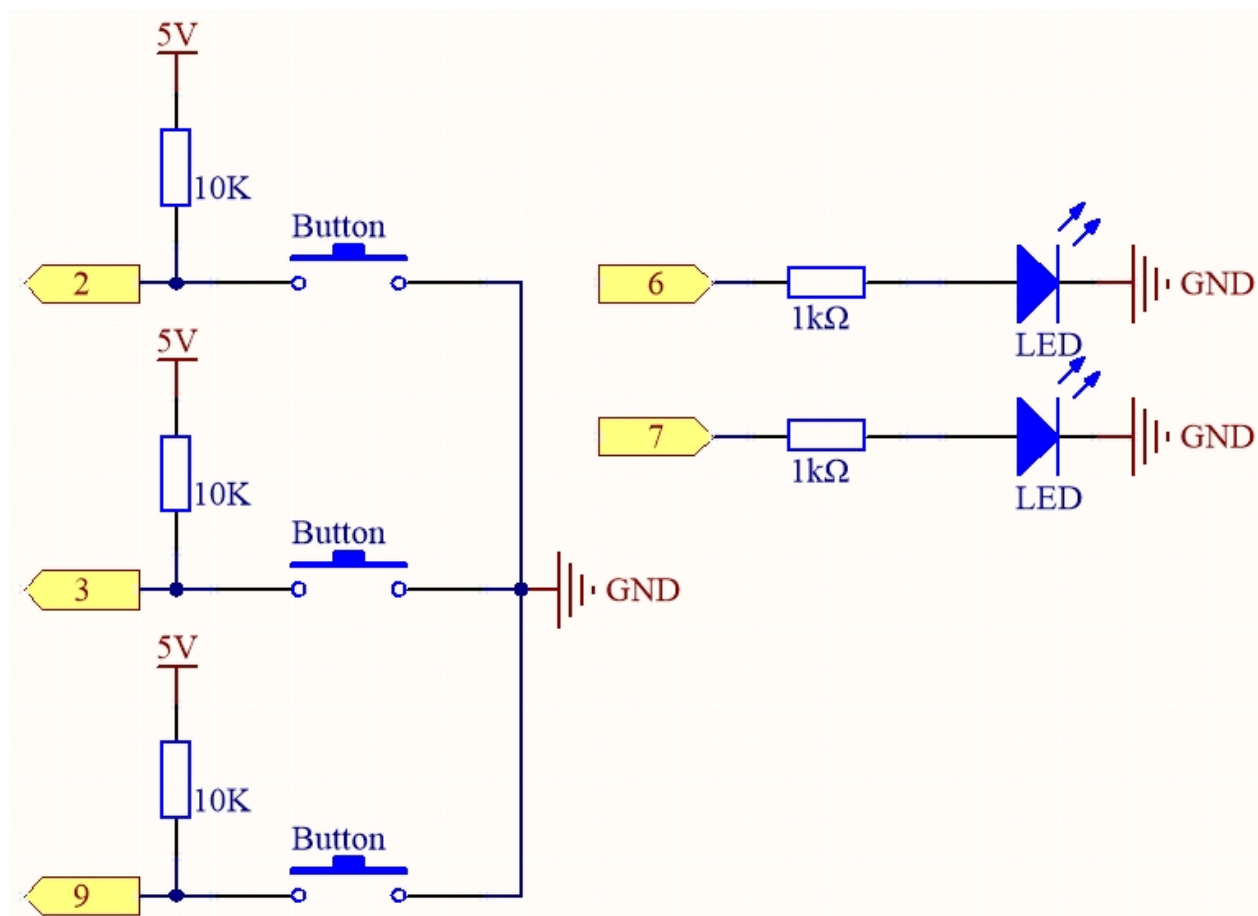
一式をまとめて購入すると便利です。以下のリンクを参照してください：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

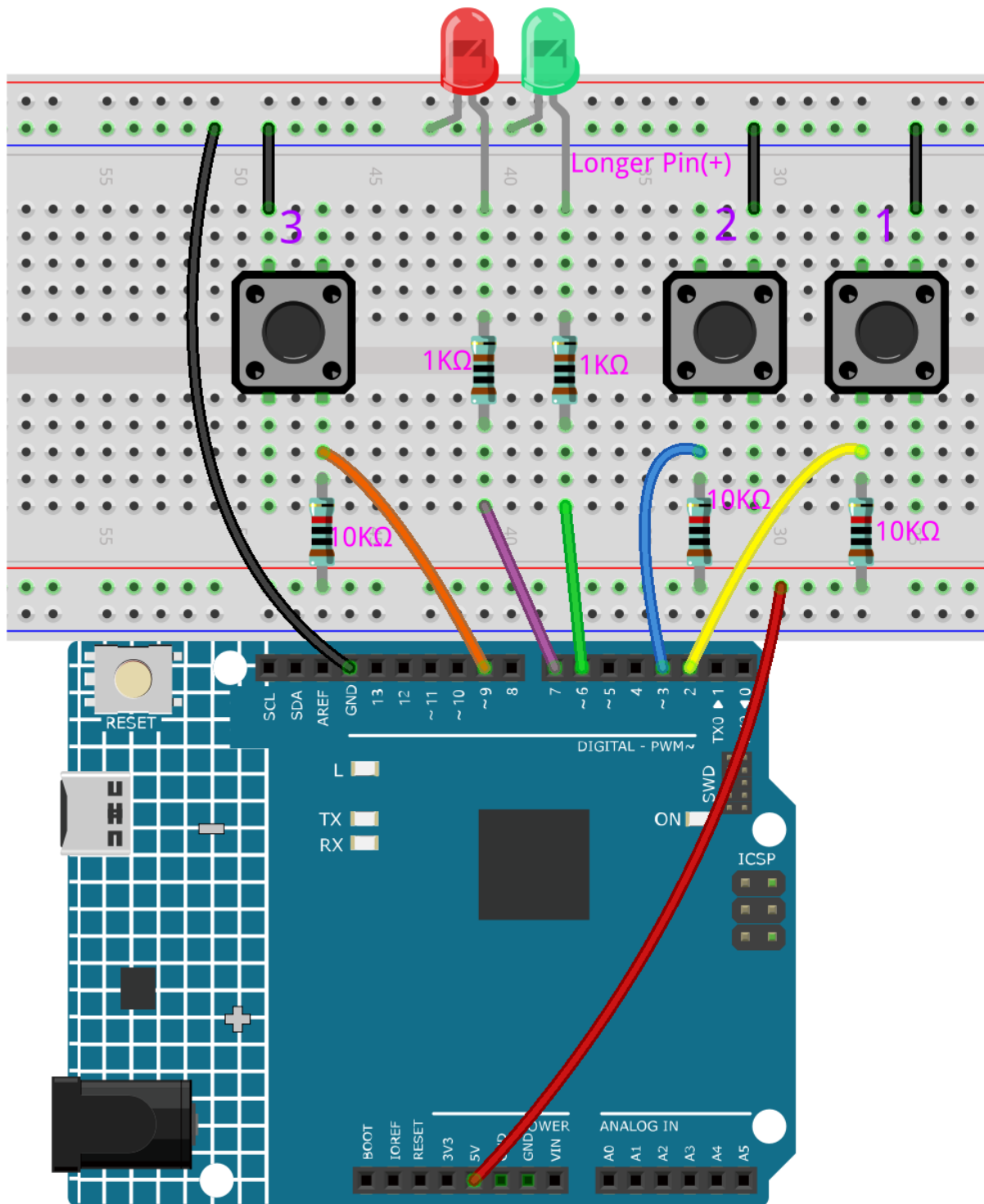
以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
Arduino Uno R4 Minima	-
ブレッドボード	
ジャンパーワイヤー	
抵抗器	
LED	
ボタン	

回路図



配線図



コード

注釈:

- ファイル 6.5_reaction_time.ino を 3in1-kit\learning_project\6.5_reversingAid のパスから直

接開くことができます。

- または、このコードを Arduino IDE にコピーします。
- LiquidCrystal_I2C ライブラリを追加していることを確認してください。詳しいチュートリアルは [5.11 外部ライブラリのインストール](#) を参照してください。

どのように動作するのか？

1. ボタンと LED を初期化し、2 つの割り込みを使用してボタンのステータスを読み取ります。

```
void setup()
{
    ...
    attachInterrupt(digitalPinToInterrupt(buttonPin1), pressed1, FALLING);
    attachInterrupt(digitalPinToInterrupt(buttonPin2), pressed2, FALLING);
    ...
}
```

2. rstBtn ボタンが押されると、ゲームが再開します。2ms から 5ms の間のランダムな時間で、LED のうちの 1 つを点灯させます。

```
void loop()
{
    if (flag == -1 && digitalRead(rstBtn) == LOW) {
        digitalWrite(ledPin1, LOW);
        digitalWrite(ledPin2, LOW);
        Serial.println("Waiting...");
        int randomTime = random(2000, 5000);
        delay(randomTime);

        timer = millis();
        flag = randomTime % 2;
        Serial.println("Light!");

        if (flag == 0) {
            digitalWrite(ledPin1, HIGH);
        } else if (flag == 1) {
            digitalWrite(ledPin2, HIGH);
        }
    }
}
```

(次のページに続く)

(前のページからの続き)

```
    delay(200);  
}
```

- flag が-1 で rstBtn ボタンが押された場合、 random() 関数を使用して 2-5s のランダムな時間を生成します。
- この時間は LED の点灯を制御するために使用されます。
- 2 つの LED の点灯も randomTime % 2 でランダムに生成されます。flag が 0 なら LED1 が点灯、1 なら LED2 が点灯します。

3. pressed1() 関数について

```
void pressed1() {  
    if (flag == -1) {  
        return;  
    }  
    if (flag == 0) {  
        int currentTime = millis();  
        Serial.print("Correct! You reaction time is :");  
        Serial.print(currentTime - timer);  
        Serial.println(" ms");  
    } else if (flag == 1) {  
        Serial.println("Wrong Click!");  
    }  
    flag = -1;  
}
```

これはボタン 1 が押されたときにトリガーされる関数です。ボタン 1 が押され、その時の flag が 0 であれば、反応時間が出力されます。それ以外の場合、クリックエラーが表示されます。

4. pressed2() 関数について

```
void pressed2() {  
    if (flag == -1) {  
        return;  
    }  
    if (flag == 1) {  
        int currentTime = millis();  
        Serial.print("Correct! You reaction time is : ");  
        Serial.print(currentTime - timer);  
    }  
}
```

(次のページに続く)

(前のページからの続き)

```
Serial.println(" ms");
} else if (flag == 0) {
    Serial.println("Wrong Click!");
}
flag = -1;
}
```

これはボタン 2 が押されたときにトリガーされる関数です。ボタン 2 が押され、その時の flag が 1 であれば、反応時間が出力されます。それ以外の場合、クリックエラーが表示されます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なレビュー：新製品の発表や先行レビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

4.6.6 6.6 数字を当てるゲーム

数字を当てるゲームは、あなたと友人が交互に数字（0～99）を入力する楽しいパーティーゲームです。数字を入力することで範囲が狭くなり、正しい答えを出したプレイヤーが見つかったら、そのプレイヤーは敗北し、罰を受けることとなります。例えば、プレイヤーが見ることのできないラッキーナンバーが 51 で、プレイヤー 1 が 50 を入力すると、数字の範囲のヒントが 50～99 に変わります。プレイヤー 2 が 70 を入力すると、数字の範囲は 50～70 になります。プレイヤー 3 が 51 を入力すると、彼または彼女が不運なプレイヤーとなります。このゲームでは、IR リモートコントローラーで数字を入力し、LCD で結果を表示します。

必要な部品

このプロジェクトには、以下の部品が必要です。

一式をまとめて購入するのが便利です。以下のリンクから購入できます：

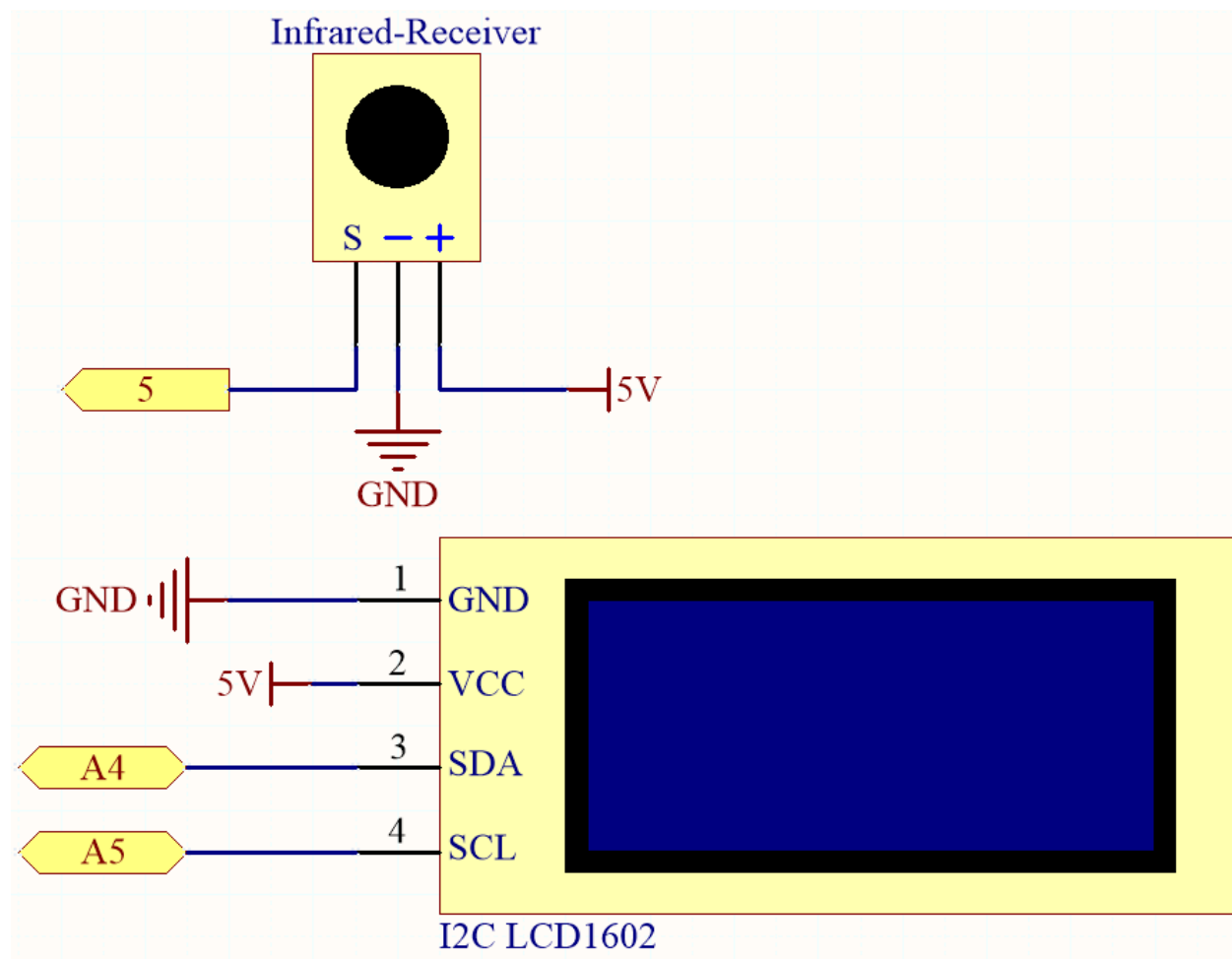
SunFounder 3in1 Kit

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから部品を個別に購入することも可能です。

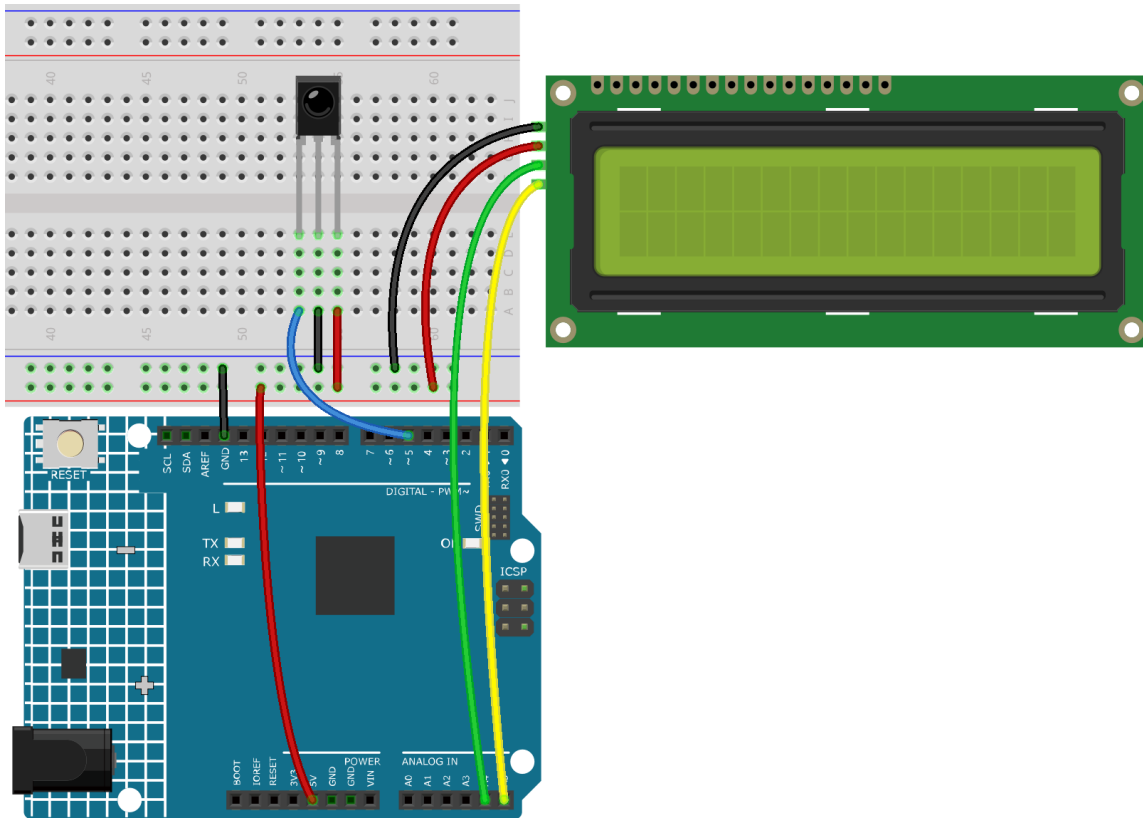
コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ジャンパーワイヤー	
<i>I2C LCD1602</i>	
<i>IR 受信機</i>	-

回路図



配線図

この例では、LCD1602 と赤外線受信モジュールの配線は以下のとおりです。



コード

注釈:

- ファイル 6.6.guess_number.ino を 3in1-kit\learning_project\6.6.guess_number のパスで直接開くことができます。
- または、このコードを Arduino IDE にコピーします。
- ここでは LiquidCrystal I2C および IRremote libraries が使用されています。 **Library Manager** からインストールすることができます。

コードのアップロードが成功すると、LCD1602 に歓迎の文字が表示されます。画面の範囲のヒントに従って番号を押すと、表示が次第に小さくなって、幸運な番号を推測するまで続きます。

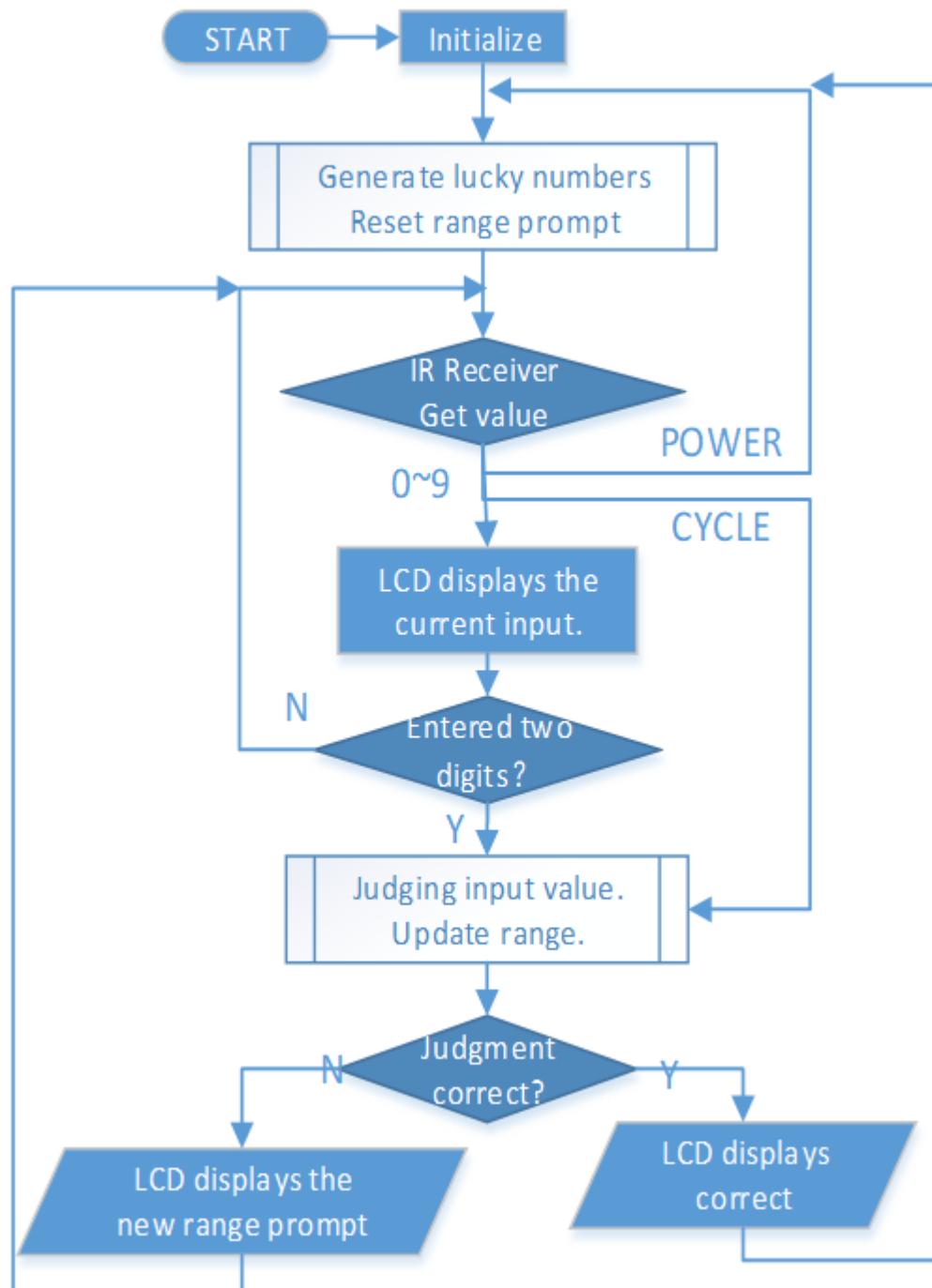
注釈: コードと配線が正しい場合でも、LCD が内容を表示しない場合は、裏面のポテンショメータを回してコントラストを上げることができます。

どのように動作するのか？

数字当てゲームを生き生きと面白くするために、以下の機能を実現する必要があります：

1. ゲームを開始し、リセットするときに、幸運な番号が表示され、数字の範囲のヒントは0～99 にリセットされます。
2. LCD は、入力されている番号と番号の範囲のヒントを表示します。
3. 2桁の数字を入力した後、結果の判定が自動的に表示されます。
4. 1桁の数字を入力する場合、結果の判定を開始するために CYCLE キー（コントローラの中央のキー）を押すことができます。
5. 答えが当てられない場合、新しい数字の範囲のヒントが表示されます（幸運な番号が51で、50を入力すると、数字の範囲のヒントは50～99 に変わります）。
6. 幸運な番号が推測されると、ゲームは自動的にリセットされ、プレイヤーは新しいラウンドをプレイすることができます。
7. ゲームは、POWER ボタン（左上角のボタン）を直接押すことでリセットできます。

結論として、プロジェクトのワークフローはフローチャートで示されています。



注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。

- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

第 5 章

カープロジェクト

多くの異なるスマートロボットカーを見たことがあると思いますが、その基本的な機能は似ています。基本的な動き、障害物回避、ラインフォロワー、追尾、リモートコントロールによる制御などです。

ここでは、最もシンプルな構造を使用してスマートロボットカーを組み立てますが、上記のすべての機能も実現できます。さらに、携帯電話で制御することもできます。チュートリアルについては、[8. IoT カー](#)を参照してください。

組み立て手順

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

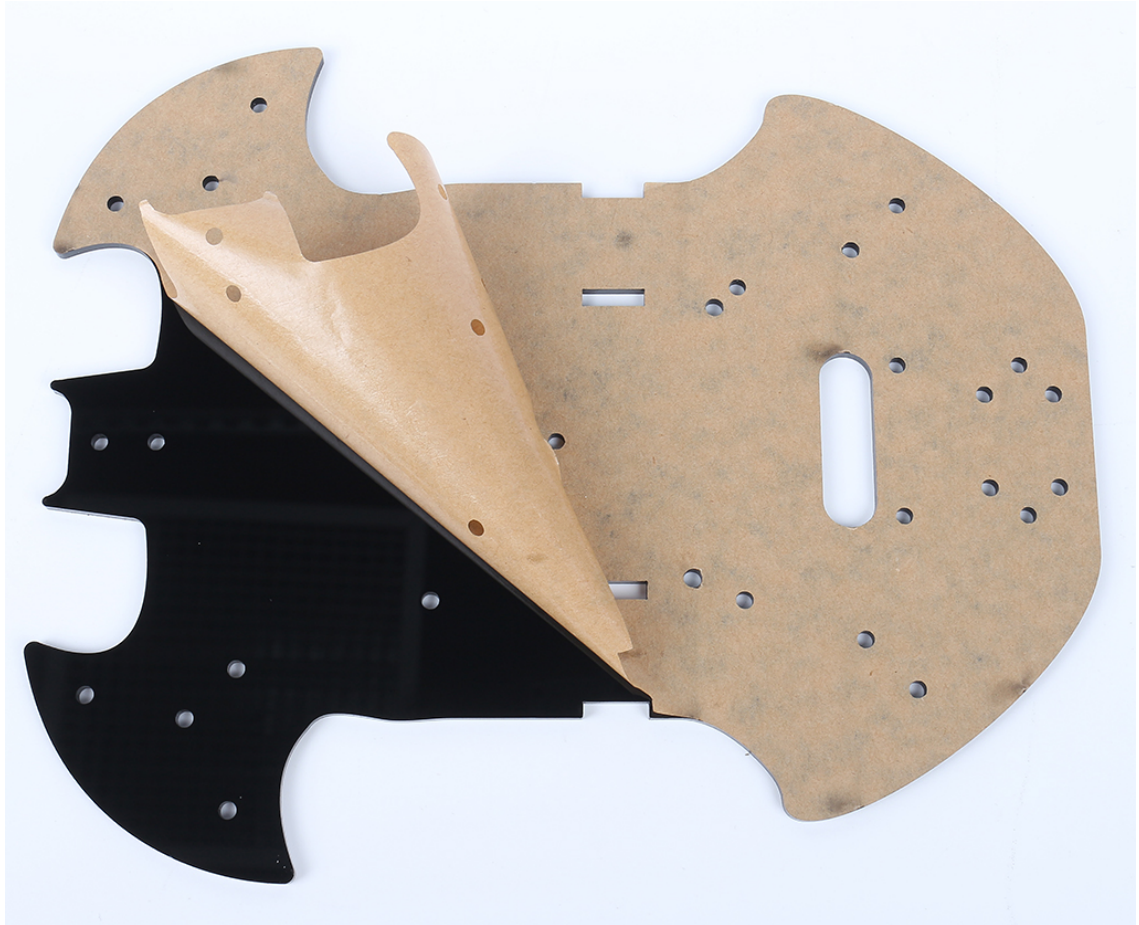
- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

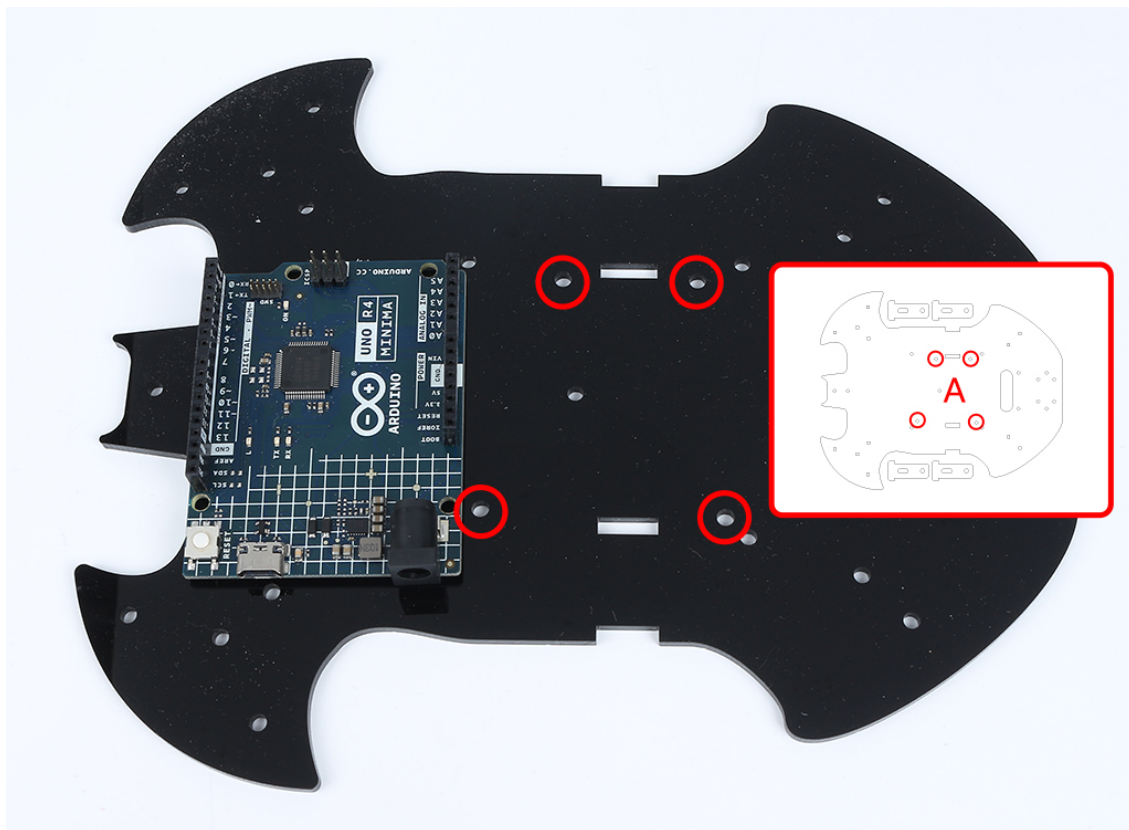
5.1 車の組み立て

以下の手順に従って、車の組み立てを完了してください。

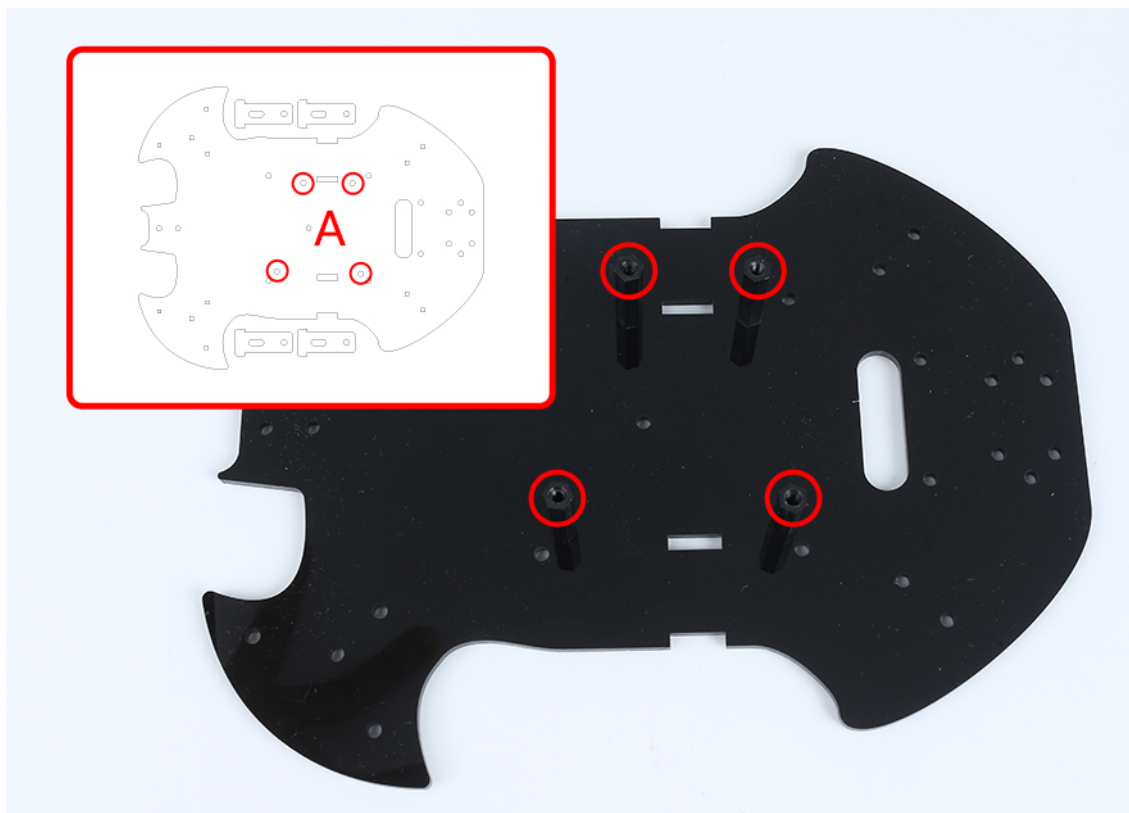
1. アクリルの保護フィルムを取り外します。



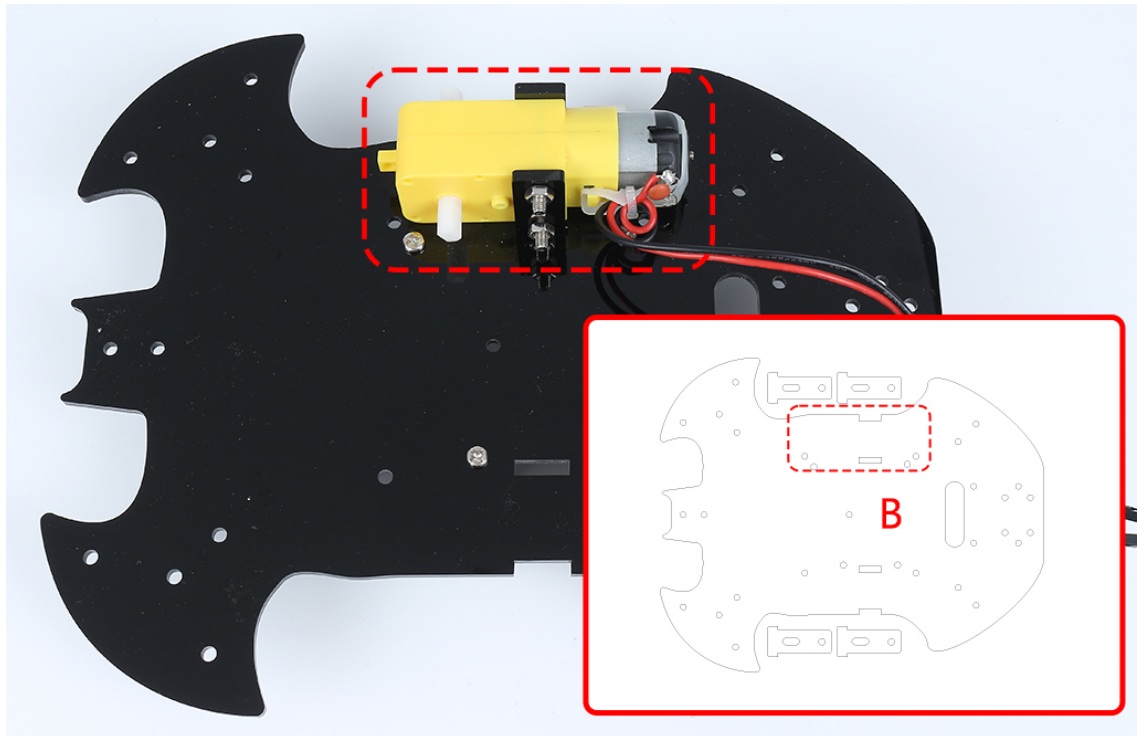
2. 画像に示されているようにボードをテーブルに置きます。R4 ボードと同じ穴の側面を A と呼び、裏面を B と呼びます。これにより、組み立て中の間違いを避けることができます。



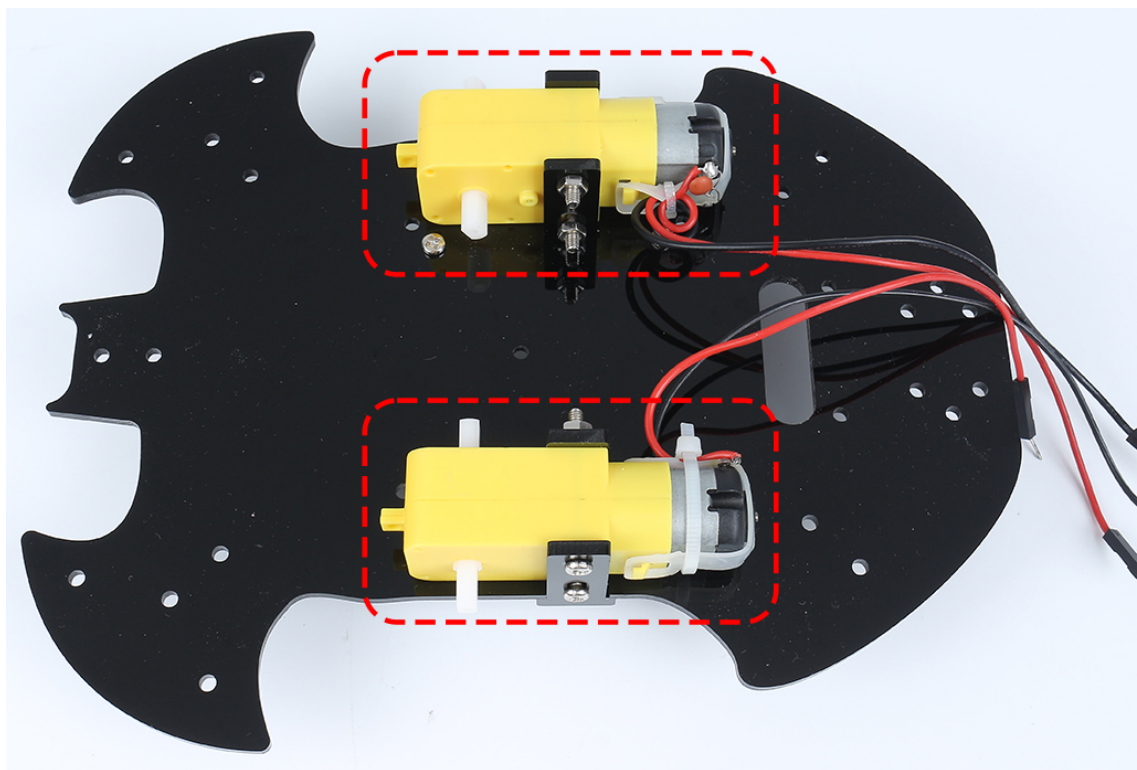
3. 下の画像に示す位置に、M3x24mm スタンドオフを M3x6mm ネジ で取り付けます。



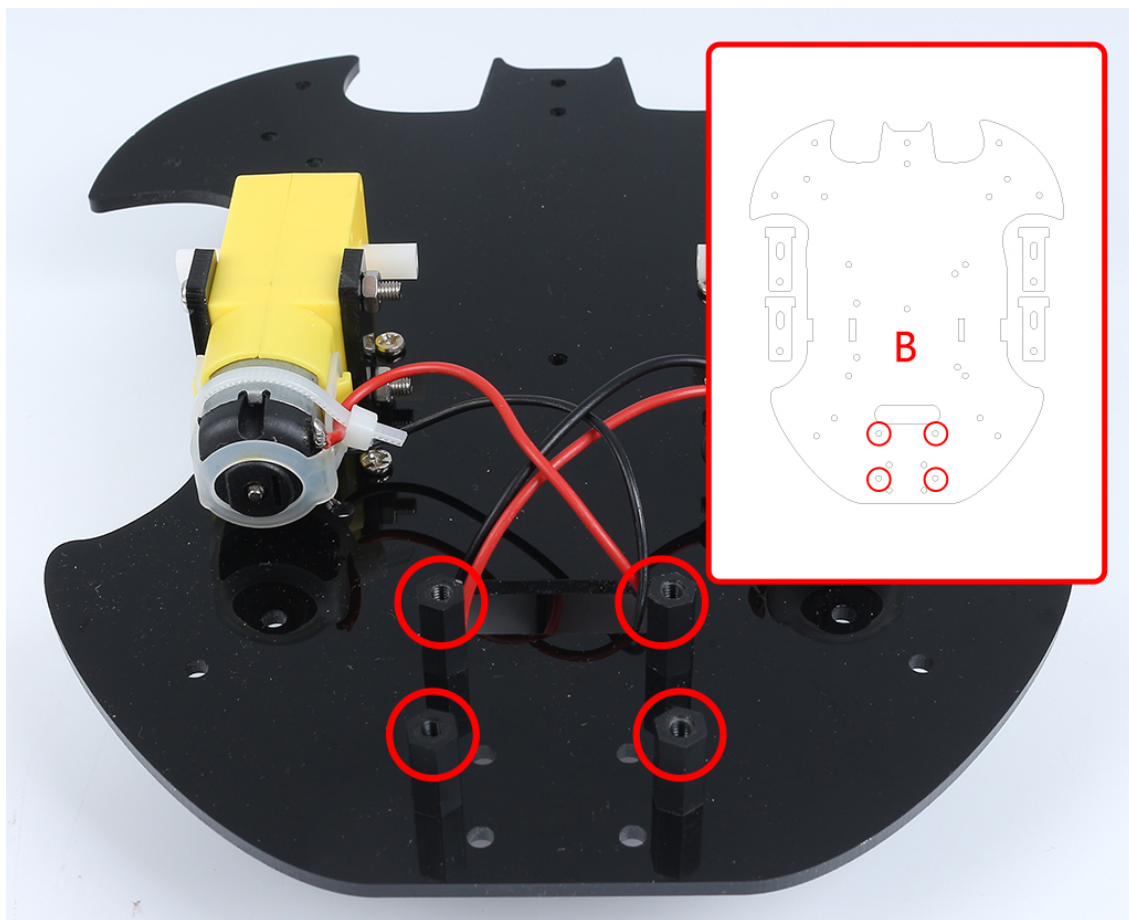
4. B 側に向かって、M3x30mm ネジ と M3 ナット を使用して TT モーターを取り付けます。2 つの詳細点：
1 - 出力軸はコウモリの形をした側に向かっています；2 - モーターケーブルは内側に向かっています。



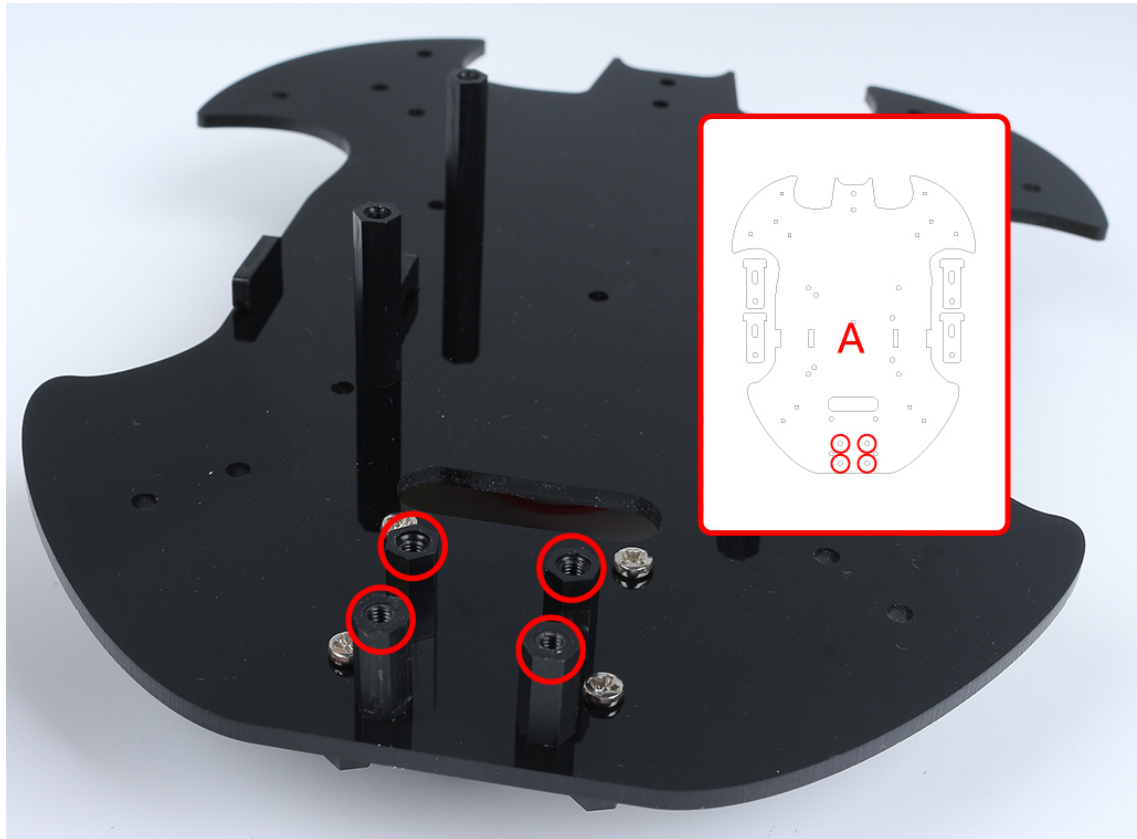
5. 別の TT モーターを取り付けます。出力軸の方向とケーブルの方向に注意が必要です。



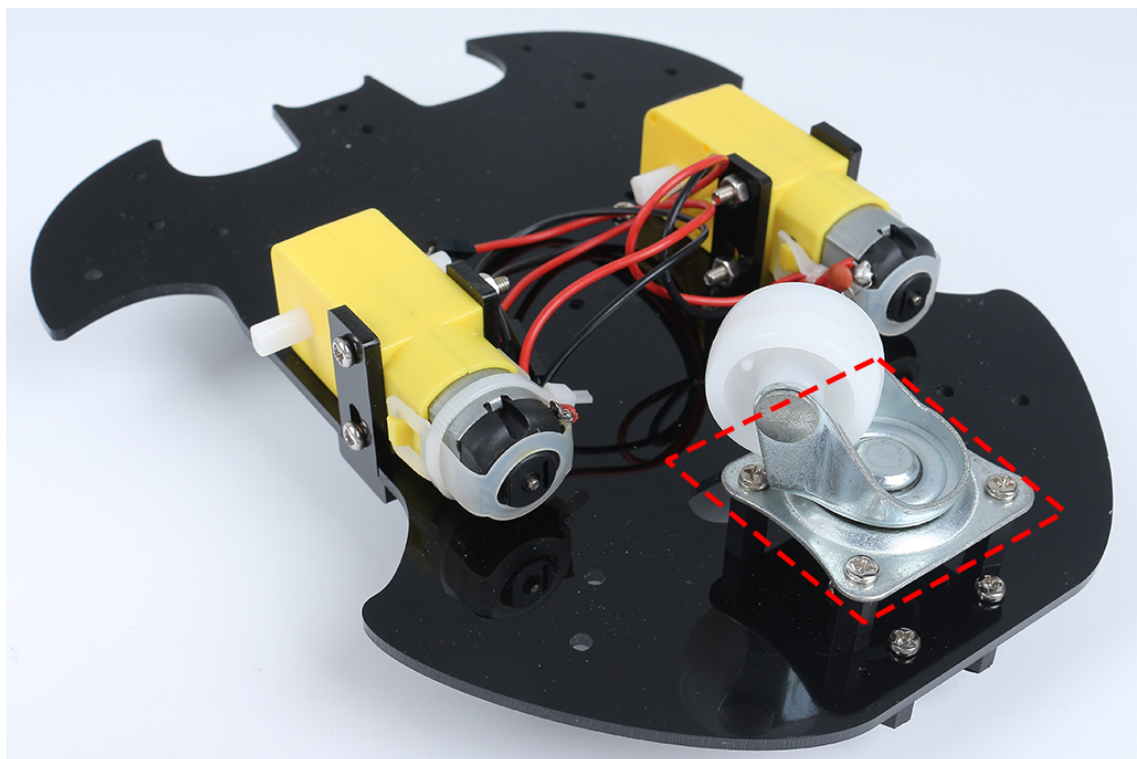
6. 下の画像に示す位置に、M3x6mm ネジ を使用して M3x10mm スタンドオフ を取り付けます。



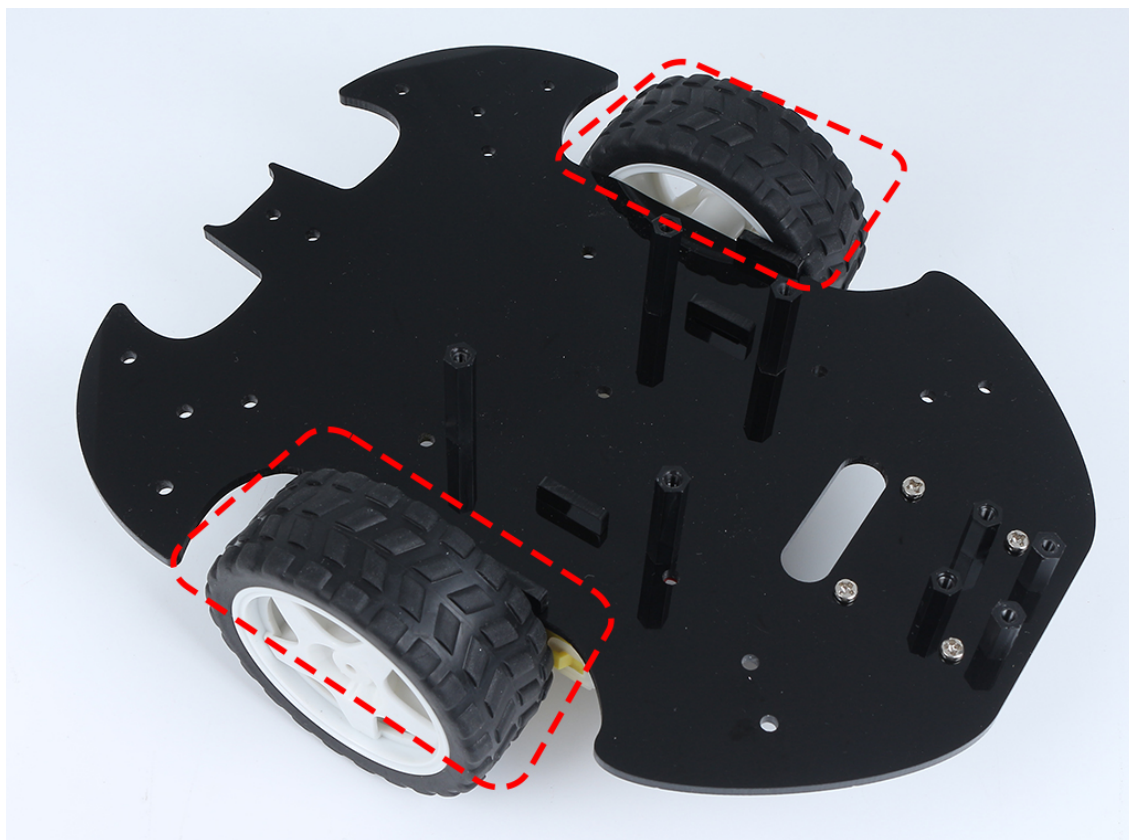
7. 車の後部に M2.5x11mm スタンドオフ を M2.5x6mm ネジ で取り付けます。



8. 汎用の車輪を M3x6mm ネジ で取り付けます。



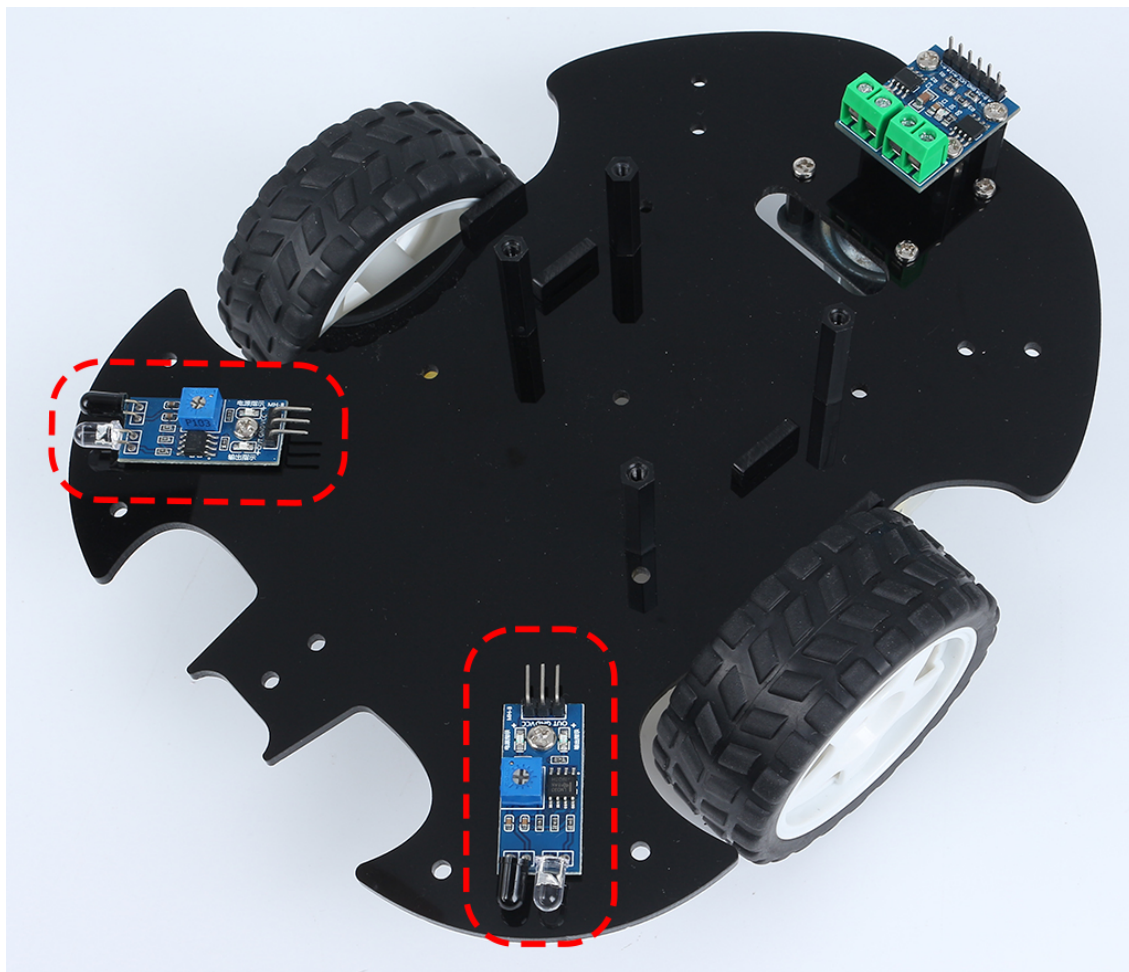
9. 2つの車輪を取り付け、車の基本構造が完成します。



10. L9110 モジュールを M2.5x6mm ネジ で取り付けます。



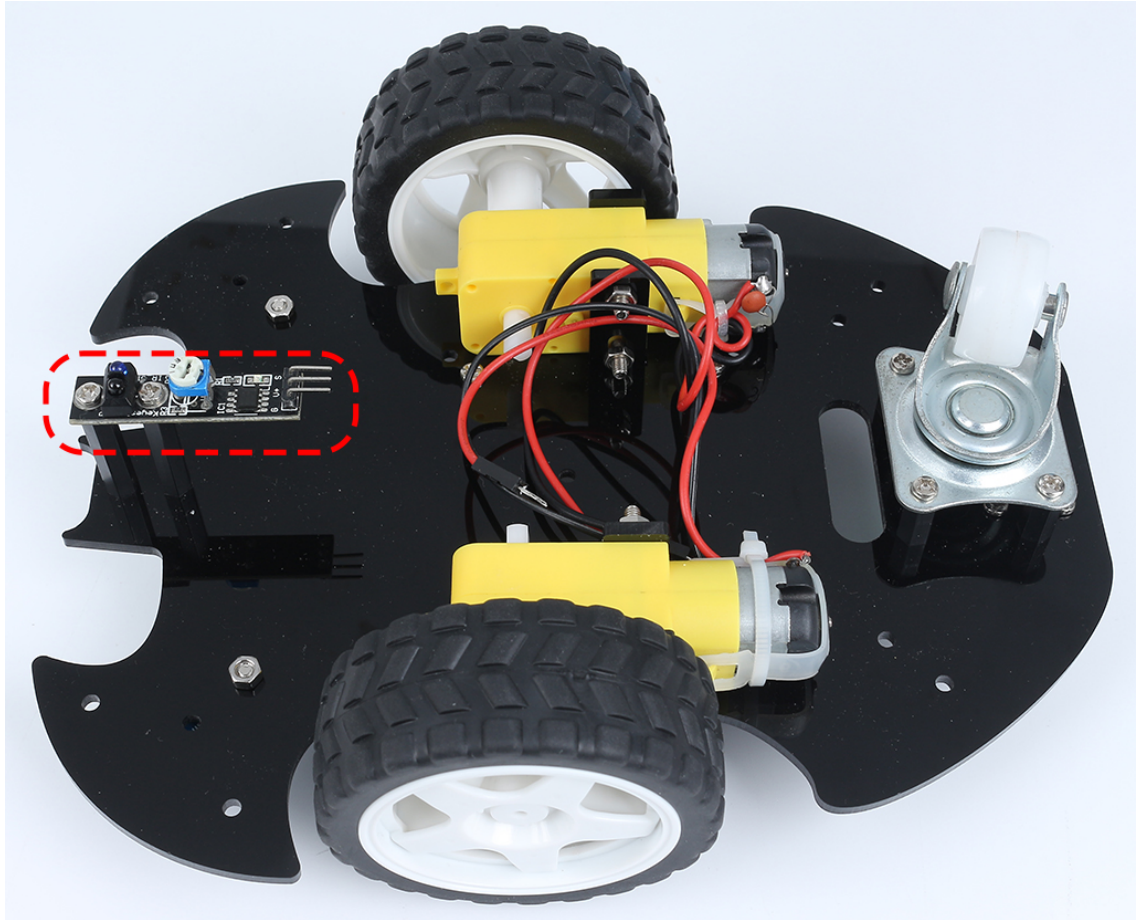
11. 2つのIR 障害モジュールを M3x10mm ネジ と M3 ナット で組み立てます。



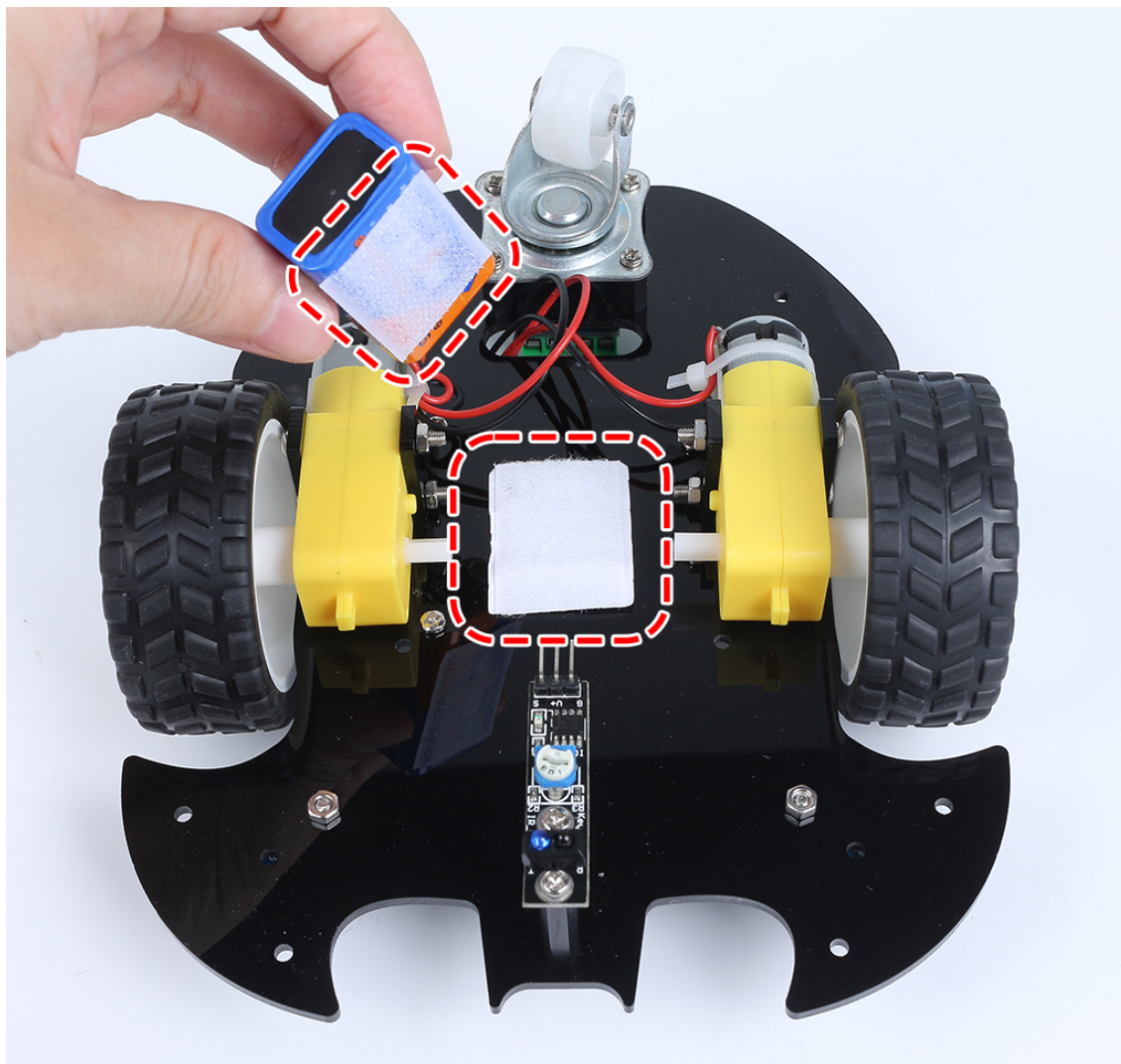
12. B 側に戻って、ライントラックモジュールを四つの M3x6mm ネジ と二つの M3x24mm スタンドオフ で取り付けます。

注釈: ライントラックモジュールに M3x24mm スタンドオフ を先に固定することをおすすめします。

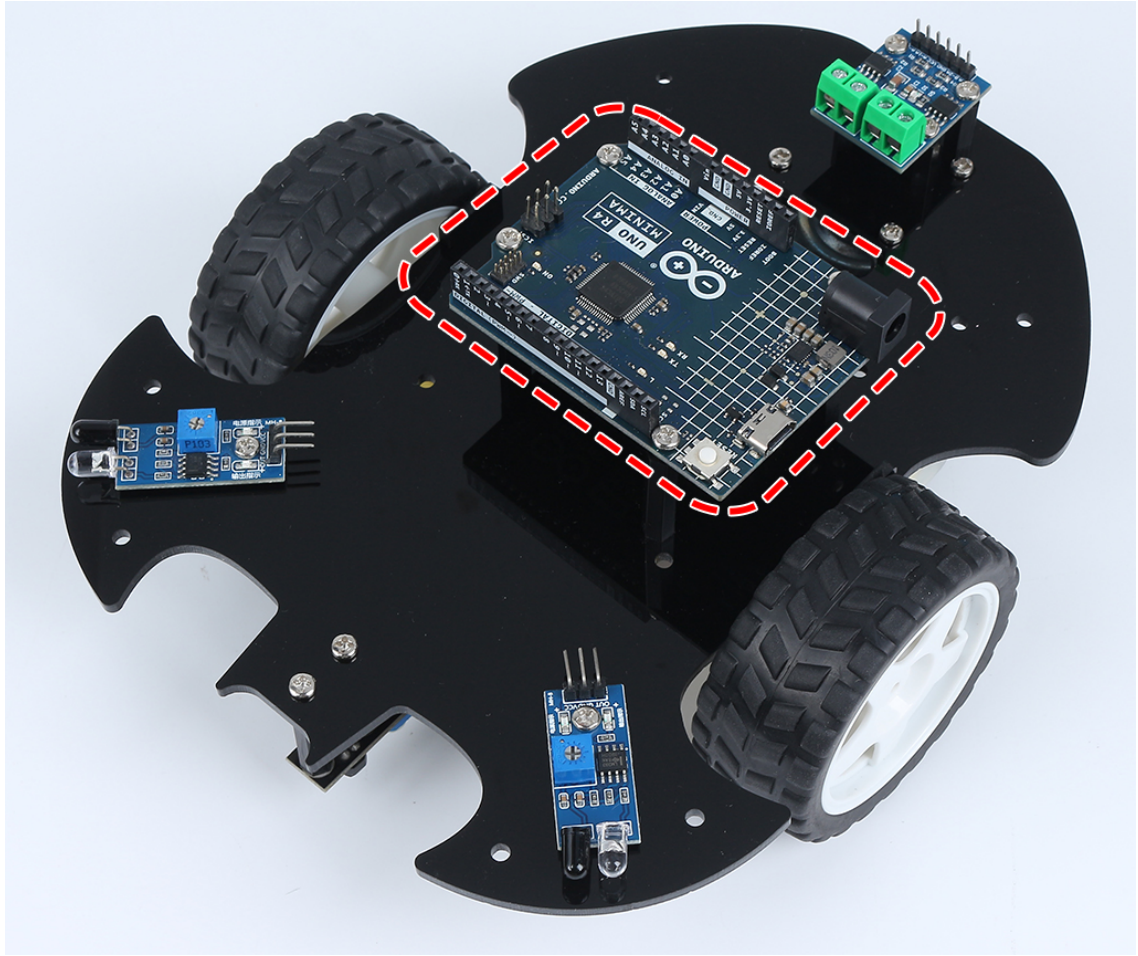
特に注意して頂きたいのは、ラインセンサーのピンは少し柔らかく、穴の方に少し突き出しています。M3x24mm スタンドオフ を取り付ける際、センサーピンを優しく押しのけるような感じでプレッシャーをかけてください。



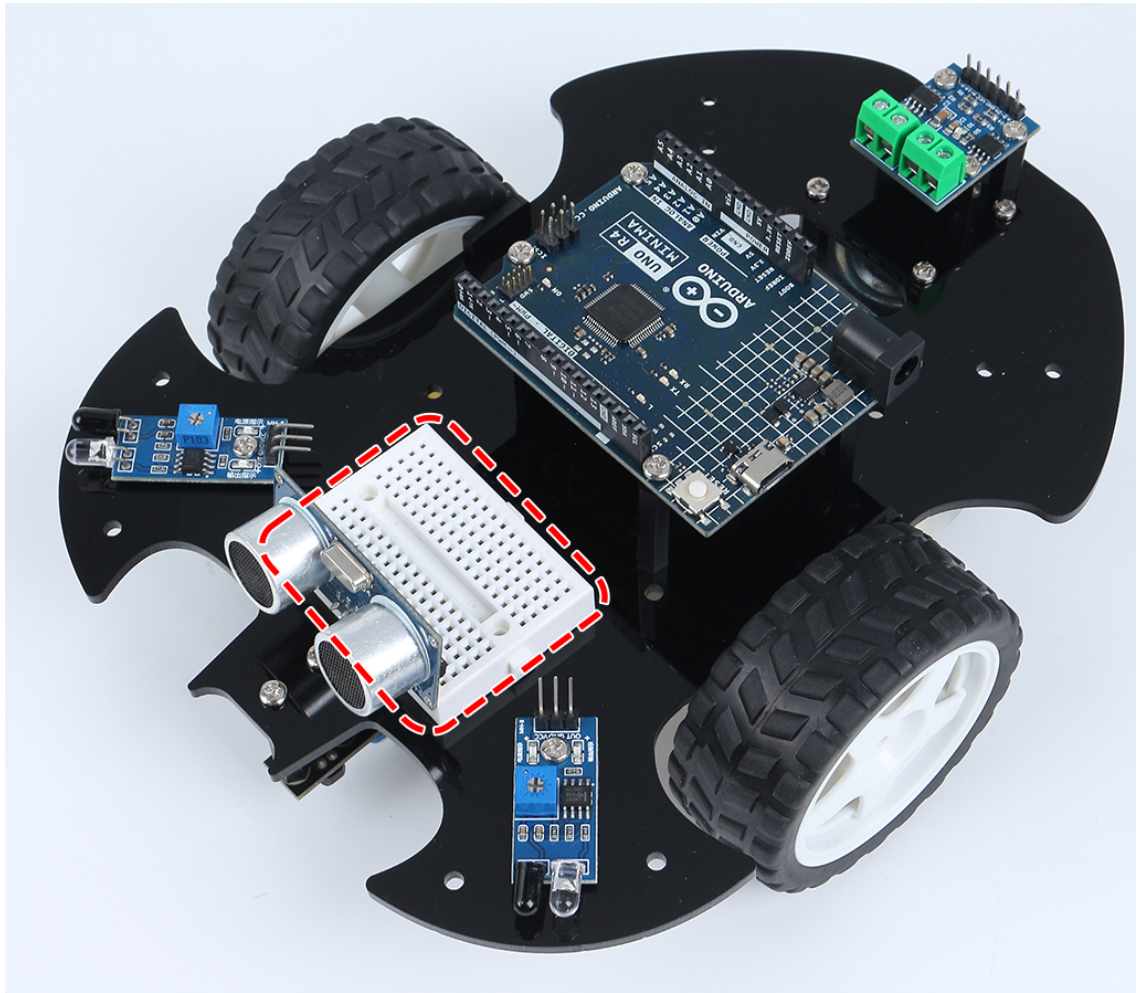
13. 9V バッテリーにベルクロを貼り付け、バッテリークリップを取り付けます。車にベルクロの他の部分を貼り付けて、バッテリーを固定します。



14. A 側に戻って、R4 ボードを M3x6mm ネジ で取り付けます。



15. 車の前面にブレッドボードを取り付けます。その後、プロジェクトに必要なに応じて、さまざまなコンポーネント（例：超音波モジュール）をブレッドボードに追加できます。



16. 車を走らせるためには、配線とコードの記述も必要です。これについては、次のセクションで詳しく説明します。

プロジェクト

以下は、Arduino IDE で C 言語でプログラムされた車のためのいくつかのプロジェクトです。Arduino に特に熟練していない場合、[Arduino のスタートガイド](#)を参照してください。

以下のプロジェクトは、プログラムの難易度の順に書かれています。これらの本を順番に読むことをおすすめします。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

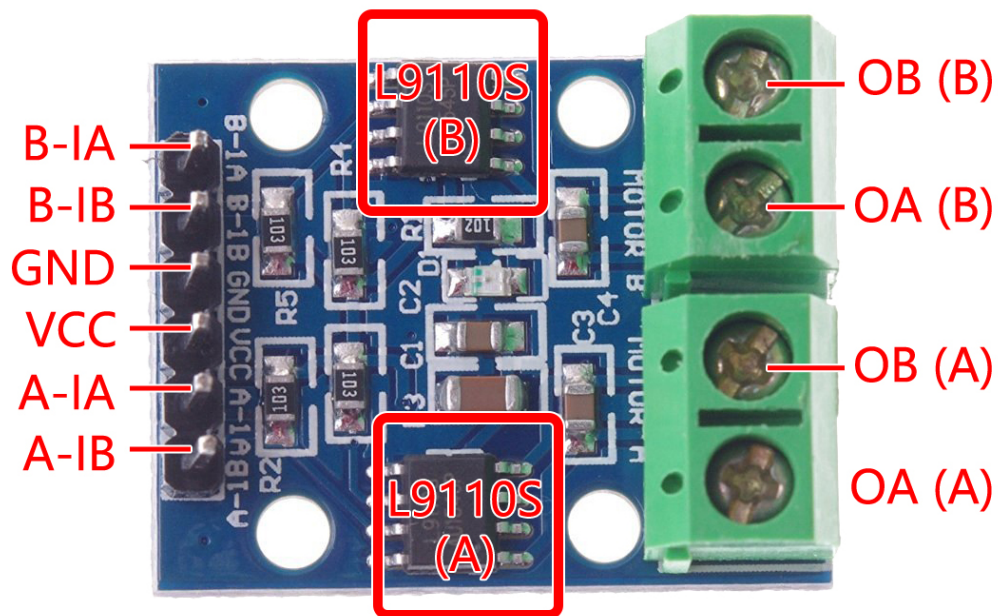
参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び & 共有：ヒントやチュートリアルを交換してスキルを向上させましょう。

- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [] をクリックして今すぐ参加しましょう！

5.2 1. 移動



プログラムを開始する前に、L9110 モジュールの動作原理を確認しましょう。

以下はモーター B の真理表です：

B-1A	B-1B(B-2A)	モーター B の状態
1	0	時計回りに回転
0	1	反時計回りに回転
0	0	ブレーキ
1	1	停止

以下はモーター A の真理表です：

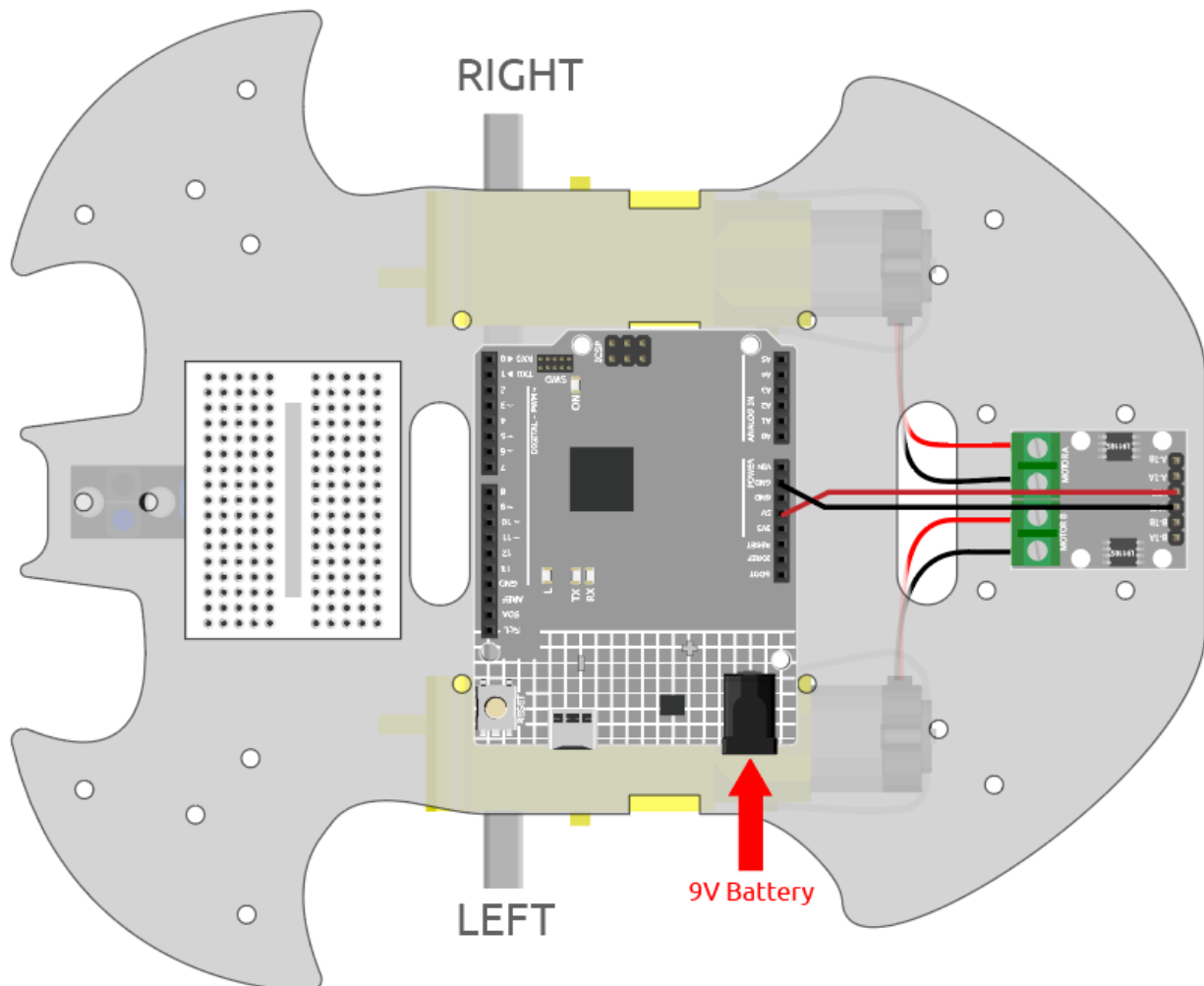
A-1A	A-1B	モーター A の状態
1	0	時計回りに回転
0	1	反時計回りに回転
0	0	ブレーキ
1	1	停止

• L9110 モータードライバーモジュール

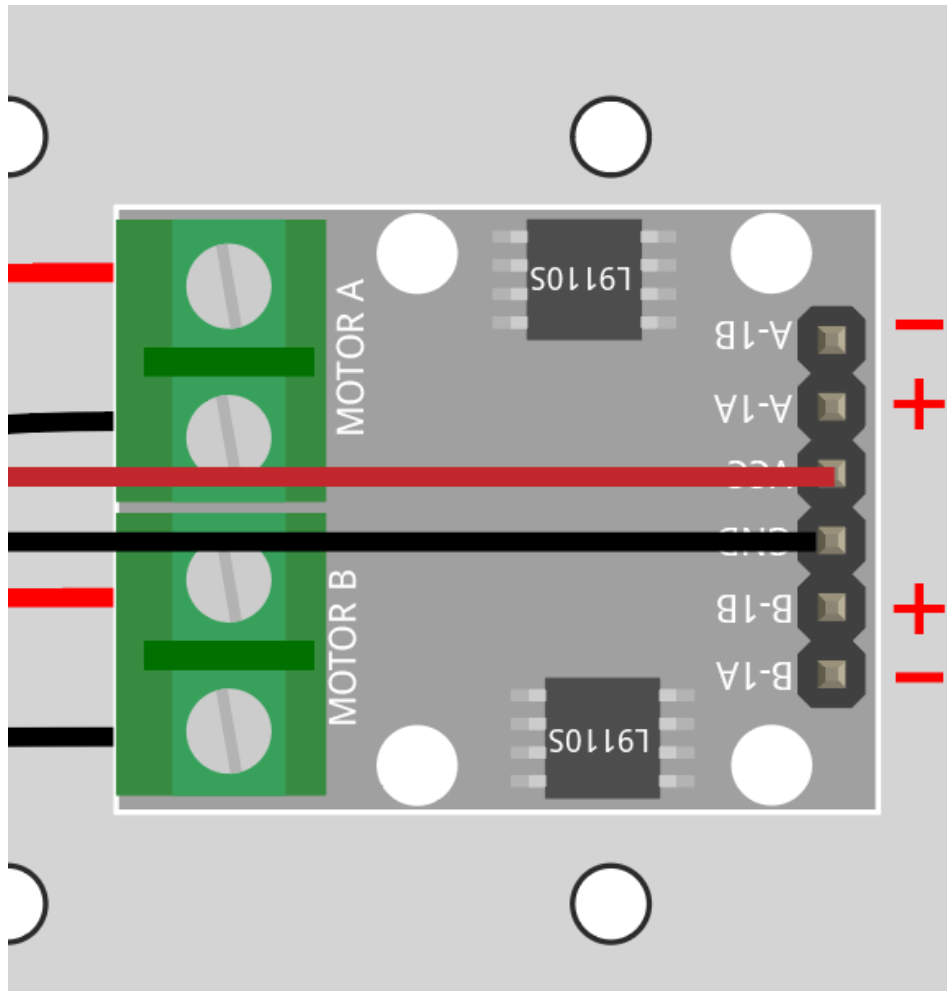
前進

車を動かすために、L9110 モジュールの入力をそれぞれ 12V と GND に直接接続してみましょう。

1. R4 ボード、L9110 モジュール、そして 2 つのモーターを接続します。



2. B-1B(B-2A) と A-1A を VCC に、B-1A と A-1B を GND に接続すると、車が前進するのが見えるはずです。

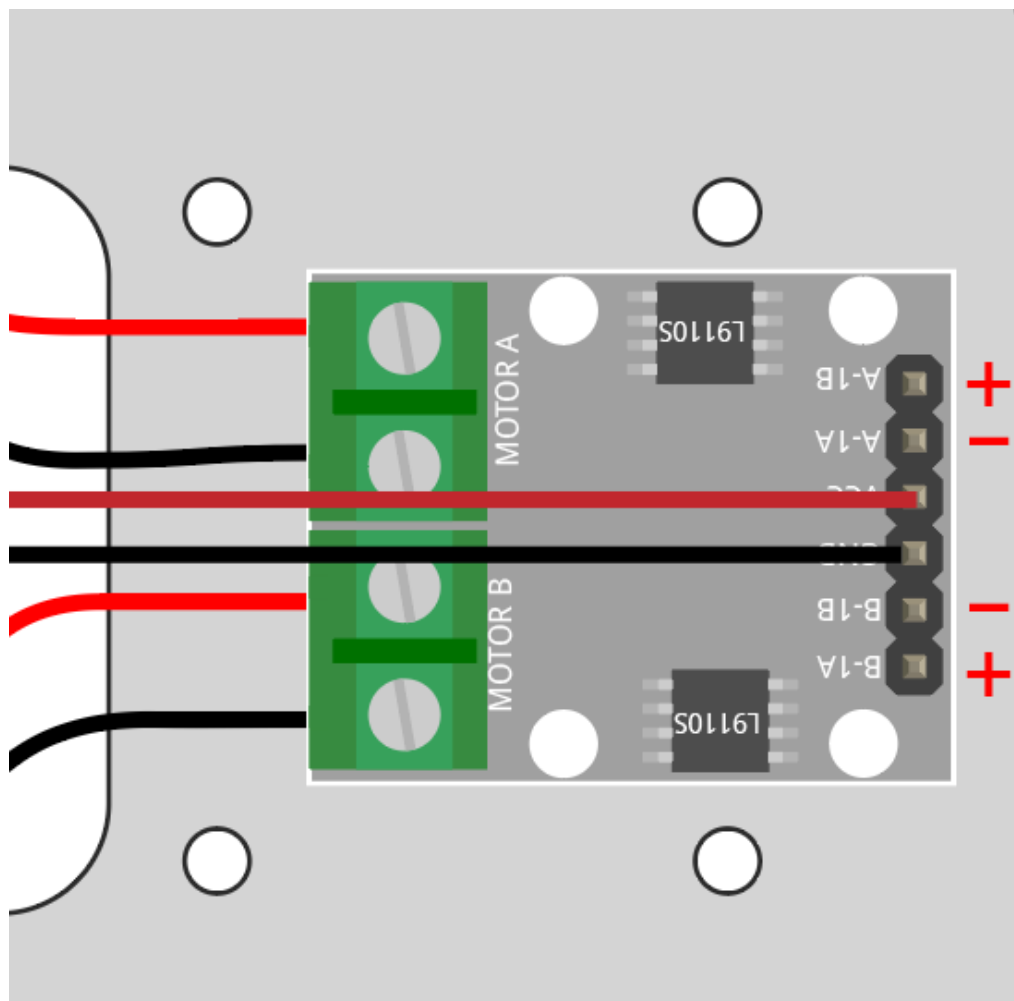


両方が前進しない場合や以下の状況が発生した場合は、2つのモーターの配線を再調整する必要があります。

- 両方のモーターが同時に後退する場合（左のモーターが時計回りに、右のモーターが反時計回りに回転）、左と右のモーターの配線を同時に入れ替え、OA(A)とOB(A)、OA(B)とOB(B)を交換します。
- 左のモーターが後退（時計回りの回転）の場合、左のモーターのOA(B)とOB(B)の配線を交換します。
- 右のモーターが後退（反時計回りの回転）の場合、右のモーターのOA(A)とOB(A)の配線を交換します。

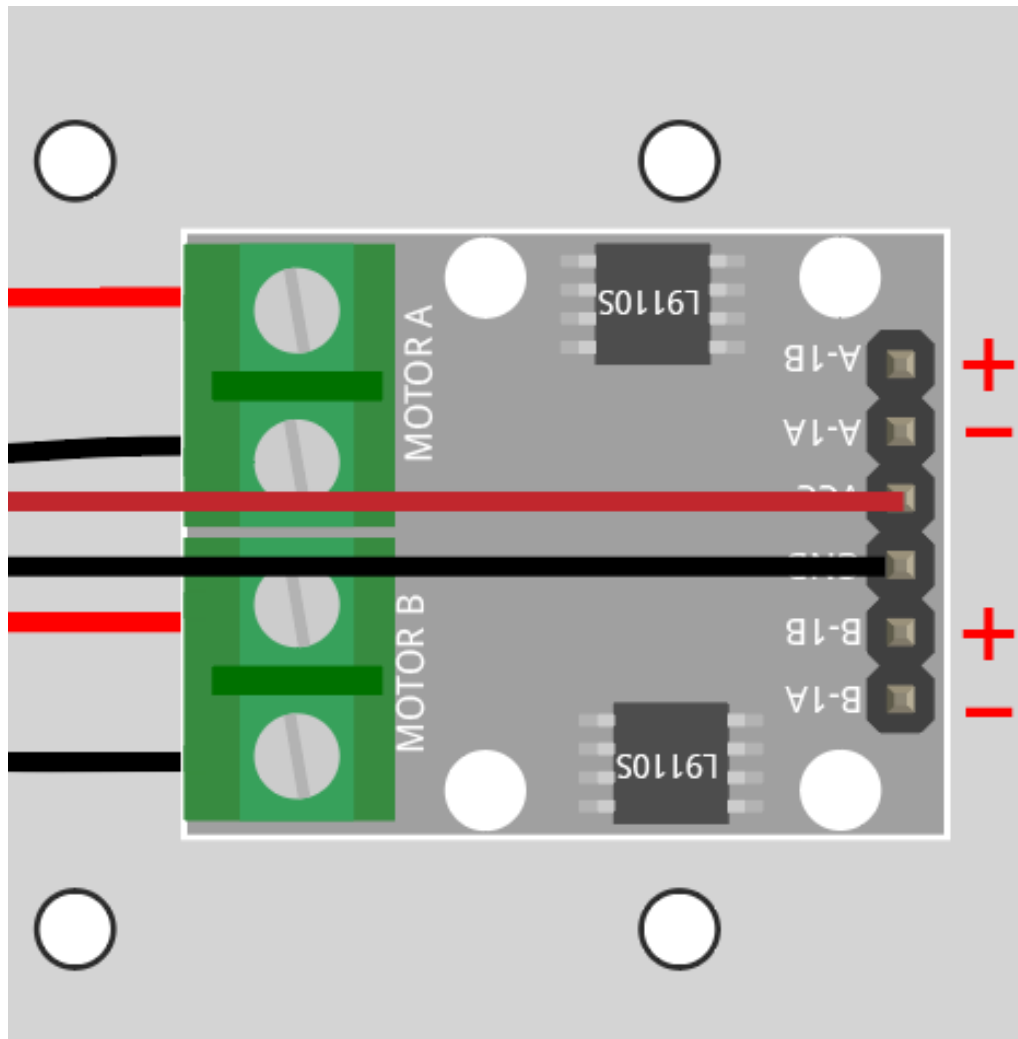
後退

B-1B(B-2A)とA-1AをGNDに、B-1AとA-1BをVCCに接続すると、車が後退するのが見えるはずです。



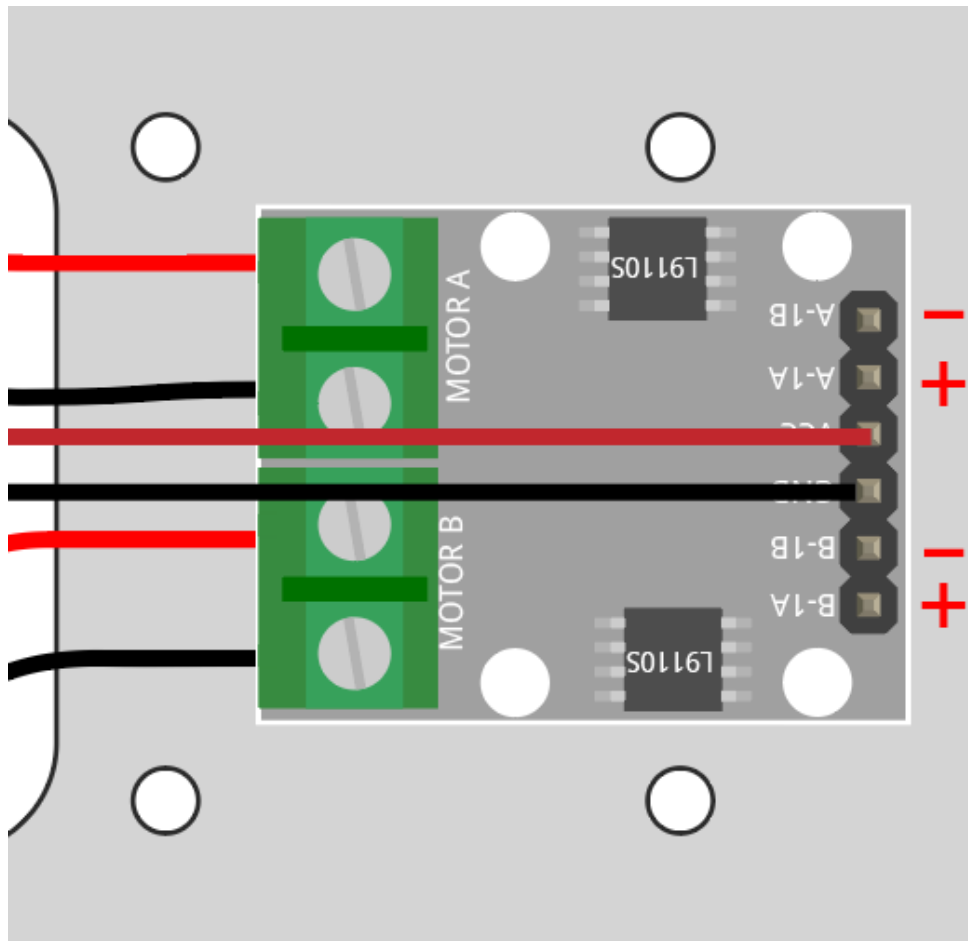
左折

車を左に曲げたい場合、つまり両方のモーターを時計回りに回転させたい場合は、B-1A と A-1A を GND に、B-1B(B-2A) と A-1B を VCC に接続します。



右折

逆に、車を右に曲げたい場合、つまり両方のモーターを反時計回りに回転させたい場合は、B-1A と A-1A を VCC に、B-1B(B-2A) と A-1B を GND に接続します。



停止

モーターを停止するには、同じ側の入力を同時に 12V または GND に接続します。例：B-1A と B-1B(B-2A) を同時に 12V または 5V に接続、同様に A-1A と A-1B も接続します。

もちろんこれは理論的なことであり、後でコードで制御する際に必要になります。ここでは車への電源を切ることです。

注釈： こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。

- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

5.3 2. コードによる移動

前のプロジェクトでは、L9110 モジュールの入力に異なるレベルの信号を使用してモーターの動作を制御しました。

プログラムを通じてレベルの信号を変更すれば、車の動きを柔軟に制御することができます。

必要な部品

このプロジェクトには、以下の部品が必要です。

一式で購入すると便利です、リンクは以下の通りです：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

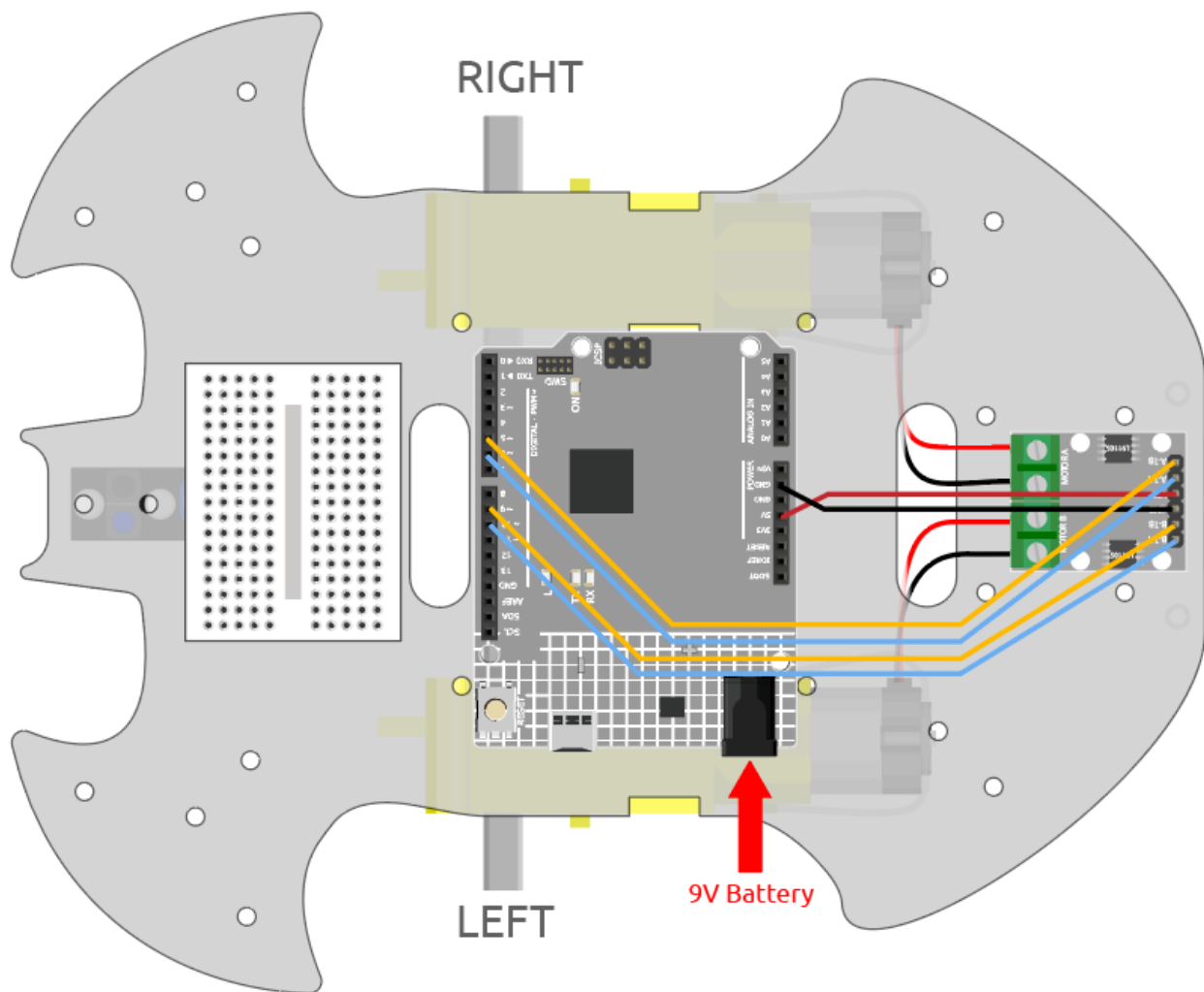
以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	
<i>L9110 モータードライバーモジュール</i>	-
<i>TT モーター</i>	-

配線図

以下の図に従って、L9110 モジュールと R4 ボードの間にワイヤを接続してください。

L9110 モジュール	R4 ボード	モータ
A-1B	5	
A-1A	6	
B-1B(B-2A)	9	
B-1A	10	
OB(B)		右モータの黒ワイヤ
OA(B)		右モータの赤ワイヤ
OB(A)		左モータの黒ワイヤ
OA(A)		左モータの赤ワイヤ



コード

注釈:

- 3in1-kit\car_project\2.move のパスの下の 2.move.ino ファイルを開きます。

- または、このコードを **Arduino IDE** にコピーします。
-

コードがアップロードされると、車はそれぞれ 2 秒間前進、後退、左折、右折をします。

どのように動作するのか？

このプロジェクトは基本的に前回のものと同じで、L9110 モジュールの入力ピンに異なる信号レベルを提供することで、車を前進、後進、左折、右折させるだけでなく、停止させるものです。

1. L9110 モジュールのピンを初期化する。

```
const int A_1B = 5;
const int A_1A = 6;
const int B_1B = 9;
const int B_1A = 10;

void setup() {
  pinMode(A_1B, OUTPUT);
  pinMode(A_1A, OUTPUT);
  pinMode(B_1B, OUTPUT);
  pinMode(B_1A, OUTPUT);
}
```

2. 左右のモーターの回転を制御するために入力ピンを異なる高さや低さのレベルに設定し、それを個別の関数にカプセル化する。

```
void moveForward() {
  digitalWrite(A_1B, LOW);
  digitalWrite(A_1A, HIGH);
  digitalWrite(B_1B, HIGH);
  digitalWrite(B_1A, LOW);
}

void moveBackward() {
  digitalWrite(A_1B, HIGH);
  digitalWrite(A_1A, LOW);
  digitalWrite(B_1B, LOW);
  digitalWrite(B_1A, HIGH);
}

...
```

3. これらの関数を `loop()` で呼び出す。

```
void loop() {  
    moveForward();  
    delay(2000);  
    stopMove();  
    delay(500);  
  
    moveBackward();  
    delay(2000);  
    stopMove();  
    delay(500);  
    ...  
}
```

- `digitalWrite(pin, value)`

- pin: Arduino のピン番号。
- value: HIGH または LOW。

デジタルピンに HIGH または LOW の値を書き込みます。ピンが `pinMode()` で OUTPUT として設定されている場合、その電圧は対応する値に設定されます：HIGH の場合は 5V (3.3V ボードでは 3.3V)、LOW の場合は 0V (グラウンド)。

- `pinMode(pin, mode)`

- pin: モードを設定する Arduino のピン番号。
- mode: INPUT、OUTPUT、または INPUT_PULLUP。

指定されたピンを入力または出力として動作するように設定します。

- `delay(ms)`

- ms: 一時停止するミリ秒数。許可されるデータタイプ: unsigned long。

パラメータとして指定された時間 (ミリ秒) の間、プログラムを一時停止します。(1 秒は 1000 ミリ秒です。)

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。

- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

5.4 3. 速度の上昇

デジタル信号（HIGH/LOW）に加えて、L9110 モジュールの入力は PWM 信号も受け取ることができ、出力の速度を制御します。

言い換えれば、`AnalogWrite()` を使用して車の移動速度を制御することができます。

このプロジェクトでは、車が前進速度を徐々に変えるようにしました。最初に加速し、その後減速します。

配線

このプロジェクトの配線は [2. コードによる移動](#) と同じです。

コード

注釈:

- `3in1-kit\car_project\3.speed_up` のパスの下に `3.speed_up.ino` ファイルを開く。
 - または、このコードを **Arduino IDE** にコピーする。
 - または、[Arduino Web Editor](#) を通じてコードをアップロードする。
-

プログラムが実行されると、車は徐々に加速し、次に徐々に減速します。

どのように動作するのか？

このプロジェクトの目的は、L9110 モジュールの入力ピンに異なる PWM 値を書き込むことで、車の前進速度を制御することです。

1. `for()` 文を使用して、0 から 255 までの値を 5 のステップで `speed` に与えることで、車の前進速度の変化を確認できます。

```
void loop() {  
    for(int i=0;i<=255;i+=5){  
        moveForward(i);  
    }  
}
```

(次のページに続く)

(前のページからの続き)

```

        delay(500);
    }
    for(int i=255;i>=0;i-=5){
        moveForward(i);
        delay(500);
    }
}

```

2. moveForward() 関数について。

2. コードによる移動 では L9110 モジュールの入力ピンに直接高/低レベルを与えていますが、ここでは高レベルを与える必要がある場所に speed パラメータを渡します。

```

void moveForward(int speed) {
    analogWrite(A_1B, 0);
    analogWrite(A_1A, speed);
    analogWrite(B_1B, speed);
    analogWrite(B_1A, 0);
}

```

• for

for 文は、波括弧で囲まれた文のブロックを繰り返し実行するために使用されます。インクリメントカウンタは通常、ループのインクリメントと終了のために使用されます。

```

for (initialization; condition; increment) {
    // statement(s);
}

```

- initialization: 最初に一度だけ実行されます。
- condition: ループを通じて毎回、condition がテストされる;それが真の場合、文のブロックとインクリメントが実行され、その後 condition が再びテストされる。condition が false になると、ループは終了する。
- increment: condition が true のときにループを通じて実行されます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。

- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [] をクリックして今すぐ参加しましょう！

5.5 4. ライン追従

この車はライントラックモジュールを搭載しており、車が黒い線を追従するようにすることができます。

ライン追従モジュールが黒い線を検出すると、右のモーターが回転し、左のモーターは回転しないため、車は左前方に一步進みます。車が動くと、ラインモジュールは線から外れ、次に左のモーターが回転し、右のモーターは回転しないため、車は右に一步進んで線に戻ります。上記の2つのステップを繰り返すことで、車は黒い線に沿って動くことができます。

プロジェクトを開始する前に、黒い線テープで曲線マップを作成する必要があります。推奨される線の幅は0.8-1.5cmで、曲がる角度は90度未満であってはいけません。

必要な部品

このプロジェクトでは、以下のコンポーネントが必要です。

キット全体を購入すると非常に便利です。以下がリンクです：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクからそれぞれ別々に購入することもできます。

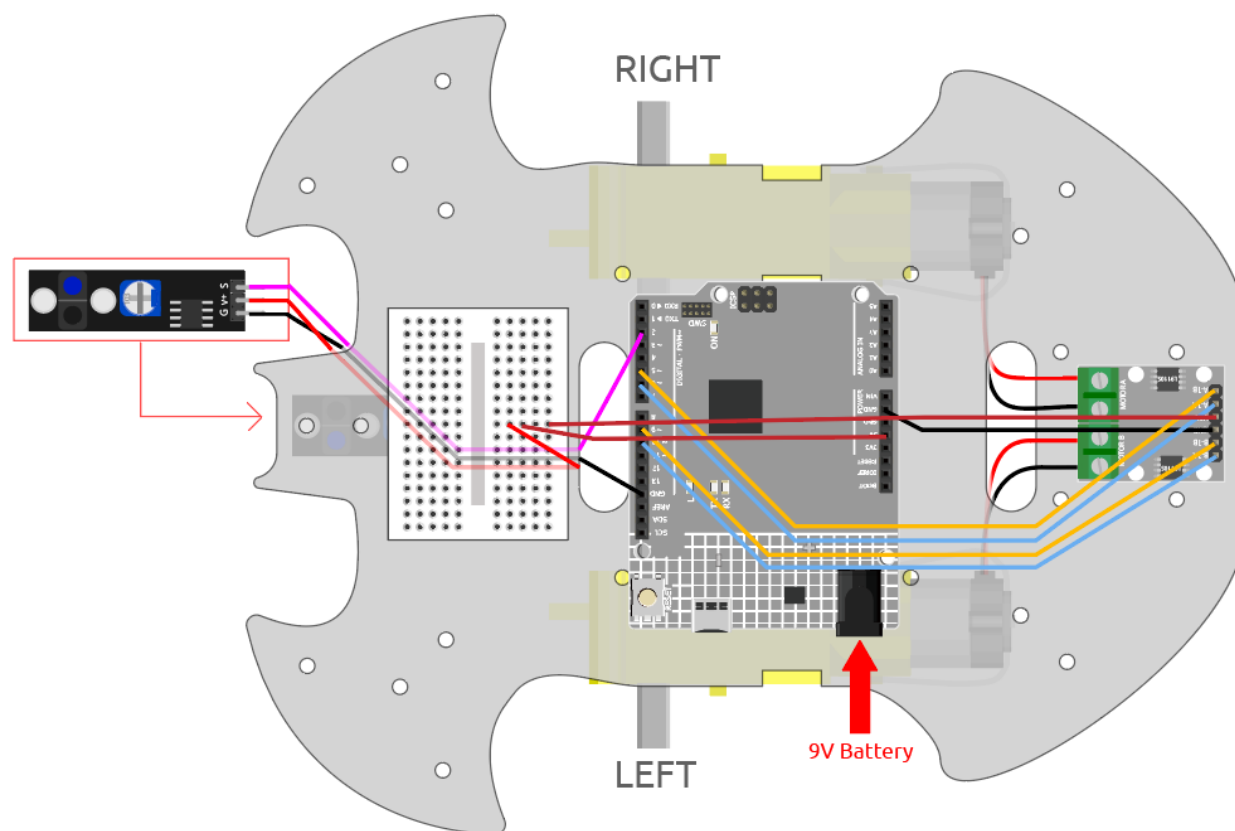
コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	
<i>L9110</i> モータードライバーモジュール	-
<i>TT</i> モーター	-
ライントラッキングモジュール	

配線図

これはデジタルライン追従モジュールです。黒い線が検出されると、出力は 1 となり、白い線が検出されると、出力は 0 となります。さらに、モジュール上のポテンショメータを調整することで、感知距離を調整することができます。

以下の図に従って回路を組み立ててください。

ライントラッキングモジュール	R4 ボード
S	2
V+	5V
G	GND



モジュールの調整

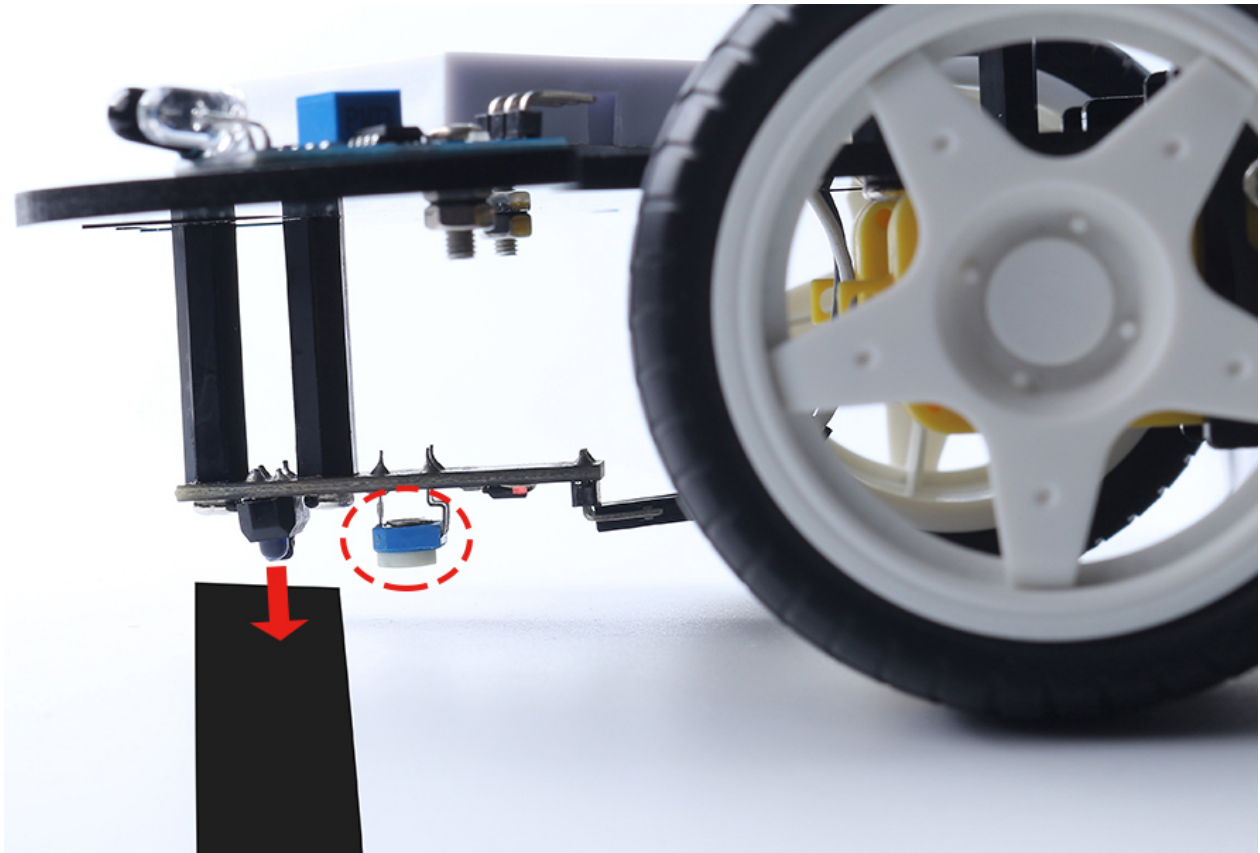
プロジェクトを開始する前に、モジュールの感度を調整する必要があります。

上記の図に従って配線し、R4 ボードに電源を供給します (USB ケーブルを直接挿入するか、9V の電池ボタンケーブルを使用)。コードをアップロードすることなく。

黒い電気テープをテーブルに貼り、カートをその上に置きます。

モジュール上の信号 LED を観察して、白いテープの上で点灯し、黒いテープの上で消灯することを確認してください。

そうでない場合は、モジュール上のポテンショメータを調整して、上記の効果を実現できるようにします。



コード

注釈:

- 3in1-kit\car_project\4.follow_the_line のパスの下で 4.follow_the_line.ino ファイルを開きます。
 - または、このコードを **Arduino IDE** にコピーします。
 - あるいは、[Arduino Web Editor](#) を通じてコードをアップロードします。
-

R4 ボードにコードをアップロードした後、車の下ライン追跡モジュールを黒線に合わせると、車が線に沿って動くのを見ることができます。

どのように動作するのか？

このコードでは、ライントラックモジュールの値に従って2つのモータを微調整することで、車が黒線に沿って動くのを見ることができます。

1. ライントラッキングモジュールのピン定義を追加します。ここでは INPUT に設定されています。また、シ

リアルモニターを初期化し、ボーレートを 9600bps に設定します。

```
...
const int lineTrack = 2;
Serial.begin(9600);
void setup() {
    ...
    pinMode(lineTrack, INPUT);
}
```

2. ライントラッキングモジュールの値を読み取ります。それが 1 の場合、車を左に進めます。そうでなければ、右に進めます。また、USB ケーブルを抜く前に、右上隅の虫眼鏡アイコンをクリックしてシリアルモニターを開き、黒と白の線上でのライントラッキングモジュール値の変化を見ることができます。

```
void loop() {

    int speed = 150;

    int lineColor = digitalRead(lineTrack); // 0:white    1:black
    Serial.println(lineColor);
    if (lineColor) {
        moveLeft(speed);
    } else {
        moveRight(speed);
    }
}
```

3. moveLeft() および moveRight() 関数について。

プロジェクト 2. コードによる移動の左右回転機能とは異なり、ここでは小さな左右の回転のみが必要です。したがって、毎回 A_1A または B_1B の値を調整するだけで十分です。例えば、左前方に移動する場合 (moveLeft())、A_1A の速度を設定し、他のすべてを 0 に設定するだけで、右のモータが時計回りに回転し、左のモータは動かないようになります。

```
void moveLeft(int speed) {
    analogWrite(A_1B, 0);
    analogWrite(A_1A, speed);
    analogWrite(B_1B, 0);
    analogWrite(B_1A, 0);
}
```

(次のページに続く)

(前のページからの続き)

```
void moveRight(int speed) {  
    analogWrite(A_1B, 0);  
    analogWrite(A_1A, 0);  
    analogWrite(B_1B, speed);  
    analogWrite(B_1A, 0);  
}
```

- Serial

Arduino ボードとコンピュータや他のデバイスとの通信に使用されます。

- Serial.begin(): シリアルデータの送信のためのビットレート (baud) を設定します。
- Serial.println(): データを人間が読める ASCII テキストとしてシリアルポートに出力し、キャリッジリターン文字 (ASCII 13 または'r') と改行文字 (ASCII 10 または'n') に続きます。

- if else

if else は基本的な if 文よりもコードの流れをより細かく制御することができるように、複数のテストをグループ化することを可能にします。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

5.6 5. 障害物回避モジュールを使って遊ぼう

車の前部には二つの赤外線障害物回避モジュールが取り付けられており、近くの障害物を検出することができます。

このプロジェクトでは、車は自由に前進することができ、障害物に遭遇した場合、それを回避して他の方向に移動を続けることができます。

必要な部品

このプロジェクトには以下の部品が必要です。

全体のキットを購入すると非常に便利です、リンクはこちら：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	
<i>L9110 モータードライバーモジュール</i>	-
<i>TT モーター</i>	-
障害物回避モジュール	

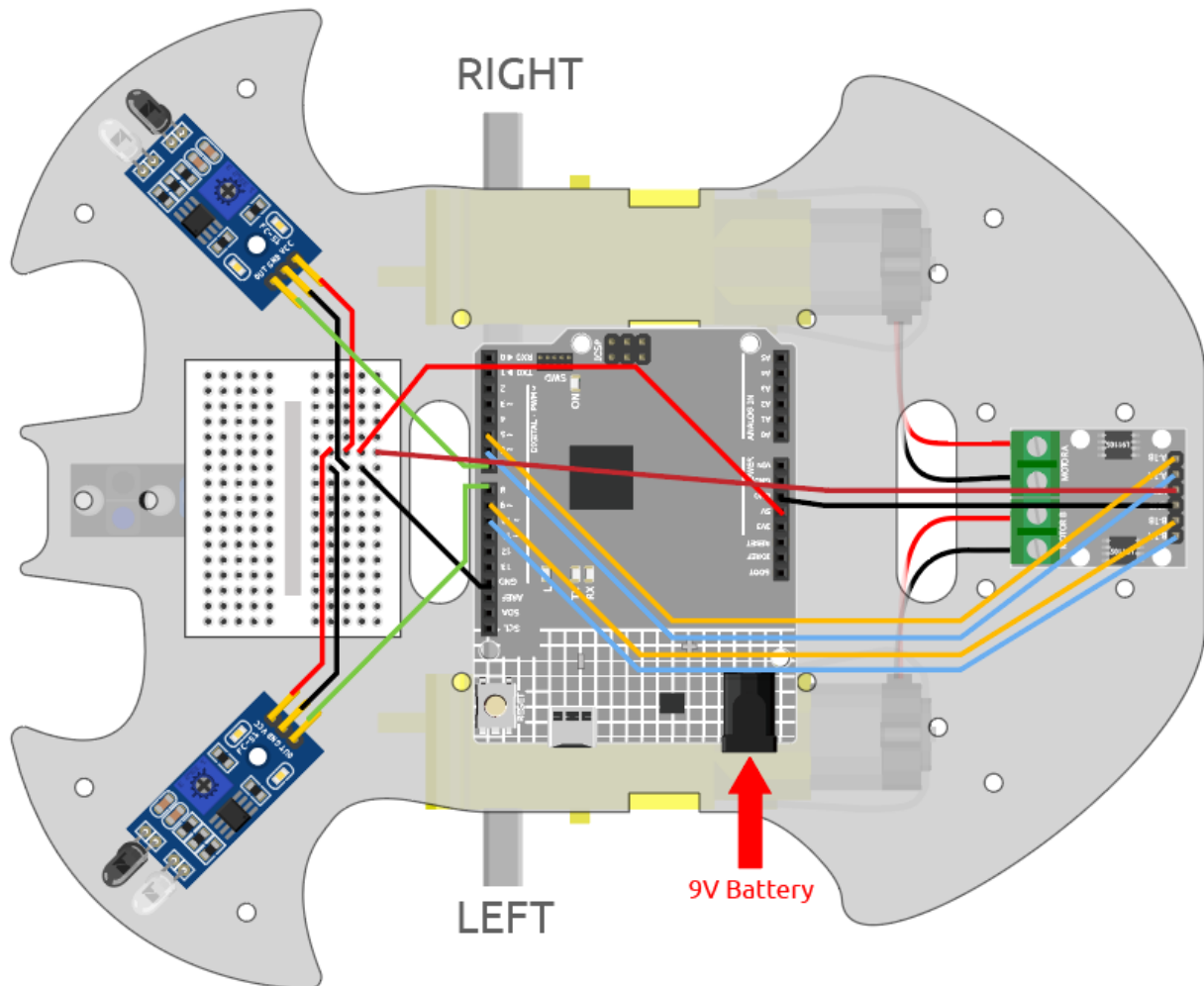
配線図

障害物回避モジュールは、障害物を検出したときに低く、通常は高い出力を持つ距離調整可能な赤外線近接センサーです。

下の図に従って回路を組み立ててください。

左 IR モジュール	R4 ボード
OUT	8
GND	GND
VCC	5V

右 IR モジュール	R4 ボード
OUT	7
GND	GND
VCC	5V



モジュールの調整

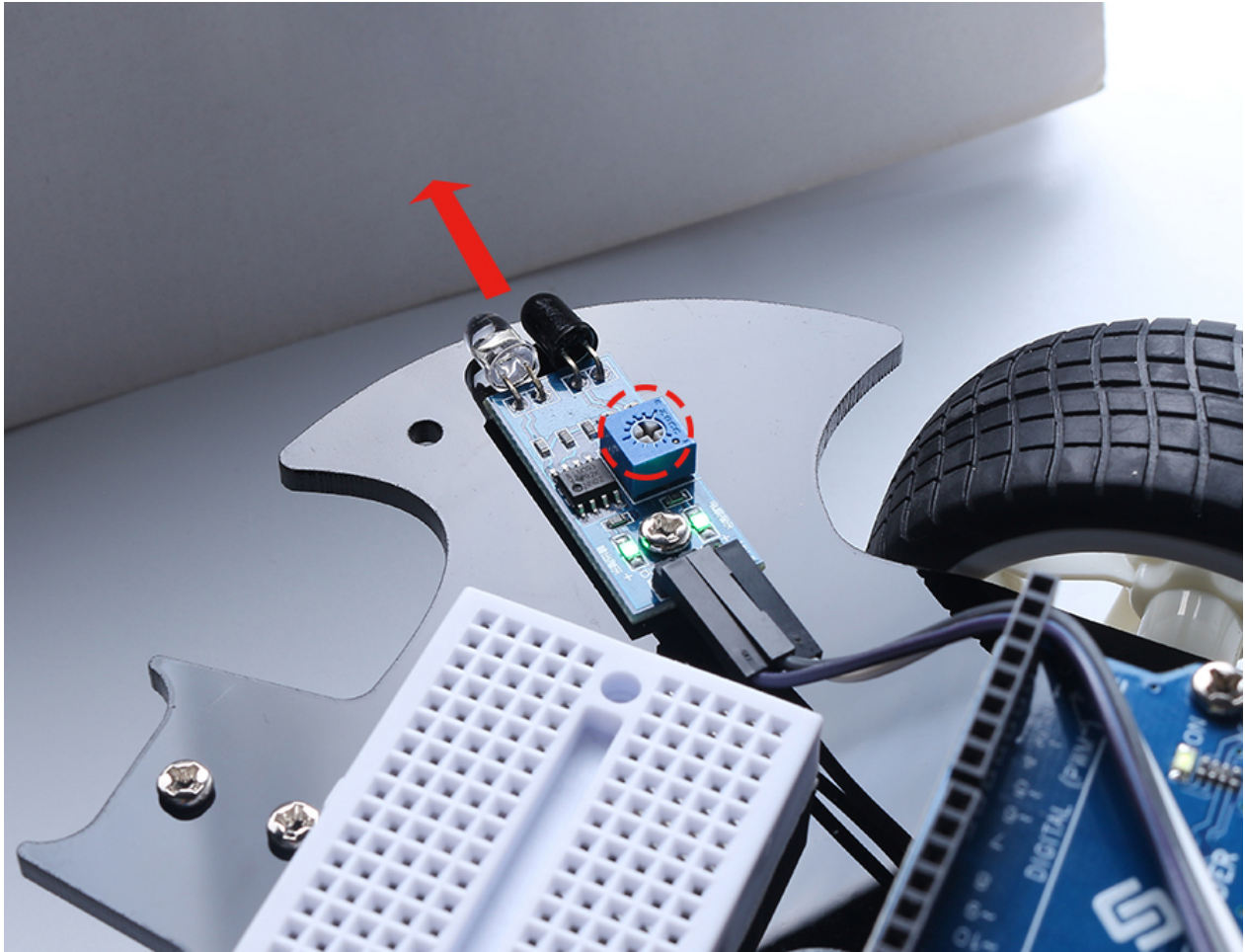
プロジェクトを開始する前に、モジュールの検出距離を調整する必要があります。

上の図に従って配線し、R4 ボードに電源を供給します（USB ケーブルを直接差し込むか、9V のバッテリーケーブルをスナップするかのいずれか）、コードはアップロードしません。

IR 障害物回避の前方約 5cm の位置にノートや他の平らな物を置きます。

その後、モジュールの信号インジケータがちょうど点灯するまで、モジュール上のポテンショメータをドライバーで回して、最大検出距離を 5cm に調整します。

同じ方法で別の赤外線モジュールも調整してください。



コード

注釈:

- 3in1-kit\car_project\5.obstacle_avoidance_module のパス下で 5.obstacle_avoidance_module.ino ファイルを開きます。
- または、このコードを **Arduino IDE** にコピーします。
- または、[Arduino Web Editor](#) を通じてコードをアップロードします。

コードが正常にアップロードされると、車は前進します。左の赤外線モジュールが障害物を検出すると、左側に後退します；右の赤外線モジュールが障害物を検出すると、右側に後退します；両方の赤外線モジュールが障害物を検出すると、直接後退します。

どのように動作しますか？

このプロジェクトは、左右の赤外線障害物回避モジュールの値に基づいて、車に適切な動作をさせるものです。

1. 2つの障害物回避モジュールのピン定義を追加します。ここでは INPUT に設定されています。

```
...  
const int rightIR = 7;  
const int leftIR = 8;  
  
void setup() {  
...  
  
//IR 障害物回避  
  pinMode(leftIR, INPUT);  
  pinMode(rightIR, INPUT);  
}
```

2. 左右の赤外線モジュールの値を読み取り、車に対応する動作をさせます。

```
void loop() {  
  
  int left = digitalRead(leftIR);  // 0: 障害物あり  1: 空き  
  int right = digitalRead(rightIR);  
  int speed = 150;  
  
  if (!left && right) {  
    backLeft(speed);  
  } else if (left && !right) {  
    backRight(speed);  
  } else if (!left && !right) {  
    moveBackward(speed);  
  } else {  
    moveForward(speed);  
  }  
}
```

- 左 IR モジュールが 0 (障害物を検出) で、右 IR モジュールが 1 の場合、車を左に後退させます。
- 右 IR モジュールが 0 (障害物を検出) の場合、車を右に後退させます。
- 2 つの IR モジュールが同時に障害物を検出すると、車は後退します。
- それ以外の場合、車は前進し続けます。

3. backLeft() 関数について。

右モーターが反時計回りに回転し、左モーターが回転しないと、車は左に後退します。


```
void backLeft(int speed) {  
    analogWrite(A_1B, speed);  
    analogWrite(A_1A, 0);  
    analogWrite(B_1B, 0);  
    analogWrite(B_1A, 0);  
}
```

4. backRight() 関数について。

左モーターが時計回りに回転し、右モーターが回転しないと、車は右に後退します。

```
void backRight(int speed) {  
    analogWrite(A_1B, 0);  
    analogWrite(A_1A, 0);  
    analogWrite(B_1B, 0);  
    analogWrite(B_1A, speed);  
}
```

- `&&`: 両方のオペランドが真の場合にのみ真となる論理 AND。
- `!`: オペランドが偽の場合に真となり、その逆もまた真となる論理 NOT。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なレビュー：新製品の発表や先行レビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

5.7 6. 超音波モジュールで遊ぼう

5. 障害物回避モジュールを使って遊ぼう のプロジェクトでは、2つの赤外線障害物回避モジュールを使用して障害物を避けていますが、赤外線障害物回避モジュールの検出距離は短いため、車が障害物を避けるのが遅くなる可能性があります。

このプロジェクトでは、車が遠くの障害物を感知して判断を下すために、超音波モジュールを使用して長距離の検出を行います。

必要な部品

このプロジェクトには、以下の部品が必要です。

一式を購入すると非常に便利です。リンクはこちら：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから個別に購入することもできます。

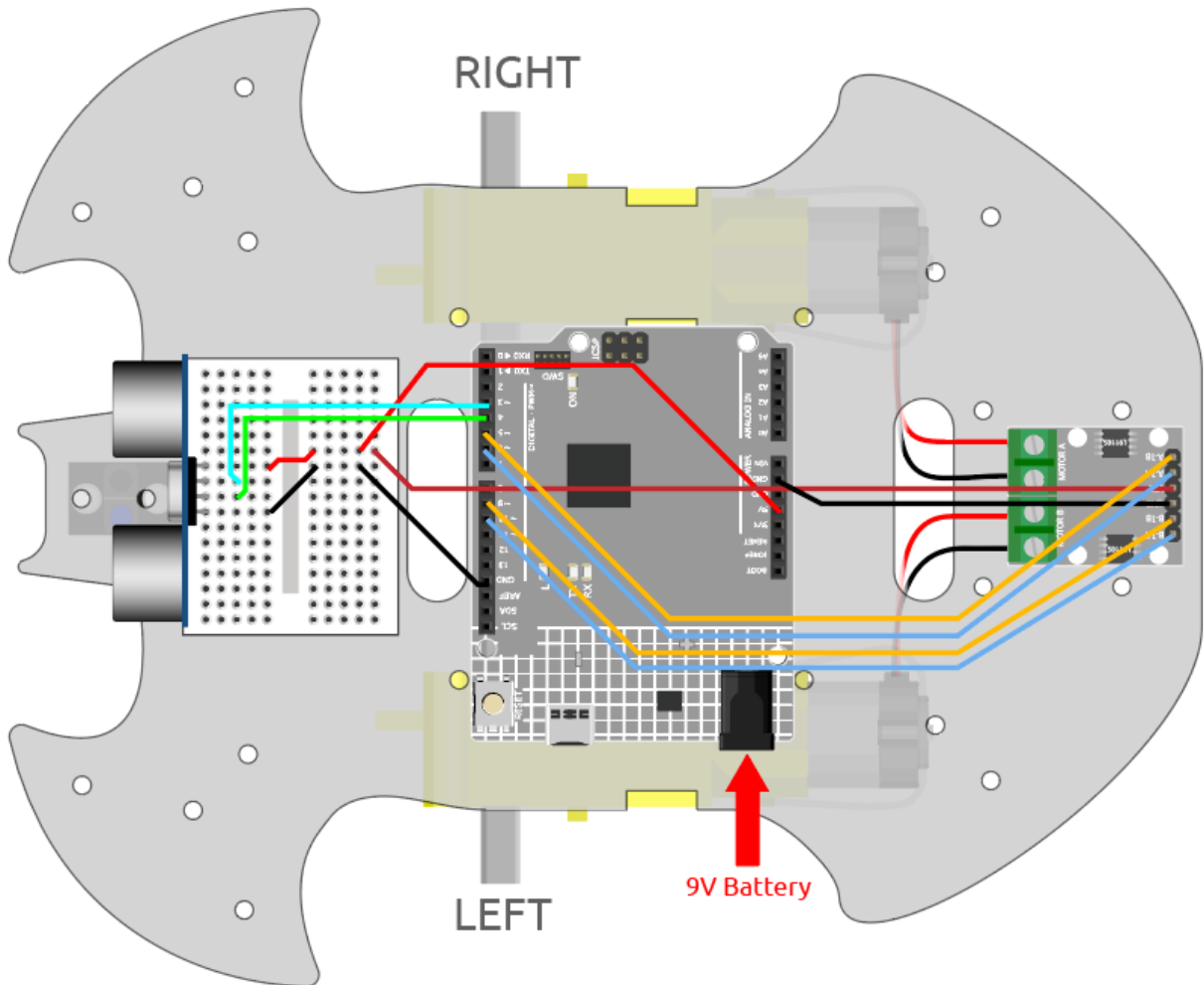
コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	
<i>L9110 モータードライバーモジュール</i>	-
<i>TT モーター</i>	-
超音波モジュール	

配線図

超音波センサーモジュールは、超音波を使用してオブジェクトまでの距離を測定する器具です。2つのプローブがあります。一つは超音波を送信するためのもの、もう一つは送受信の時間を距離に変換し、装置と障害物との距離を検出するためのものです。

以下の図に従って回路を組み立ててください。

超音波モジュール	R4 ボード
Vcc	5V
Trig	3
Echo	4
Gnd	GND



コード

注釈:

- 3in1-kit\car_project\6.ultrasonic_module のパス下で 6.ultrasonic_module.ino ファイルを開いてください。
- または、このコードを **Arduino IDE** にコピーしてください。
- あるいは、[Arduino Web Editor](#) を使用してコードをアップロードしてください。

コードが正常にアップロードされたら、車を壁の方向に向けてください。距離が遠すぎる場合は前進し、近すぎる場合は後退し、安全な距離であれば停止します。

どのように動作するのか？

このプロジェクトは、超音波モジュールから読み取った距離に基づいて車を動かすものです。

1. 超音波モジュールのピン定義を追加します。 trigPin は超音波を送信するために使用されるため、 OUTPUT に設定します。 echoPin は超音波を受信するために INPUT に設定されます。

```
...
const int trigPin = 3;
const int echoPin = 4;

void setup() {
...

//超音波
  pinMode(echoPin, INPUT);
  pinMode(trigPin, OUTPUT);
}
```

2. 超音波モジュールから得られた距離の値をまず読み取ります。距離が 25 より大きい場合、車を前進させます。距離が 2-10cm の間であれば、車を後退させます。それ以外（10~25 の間）の場合は、停止します。

```
void loop() {
  float distance = readSensorData();
  if (distance > 25) {
    moveForward(200);
  }
  else if (distance < 10 && distance > 2) {
    moveBackward(200);
  } else {
    stopMove();
  }
}
```

3. readSensorData() 関数について。

超音波モジュールの送信機は、2us ごとに 10us の矩形波信号を送信し、受信機は範囲内に障害物がある場合に高レベルの信号を受信します。 pulseIn() 関数を使用して送信から受信までの時間を記録し、音速 340m/s で割った後、2 で割ります。その結果、このモジュールと障害物との間の距離（単位：cm）が得られます。

```
float readSensorData() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
```

(次のページに続く)

(前のページからの続き)

```

    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    float distance = pulseIn(echoPin, HIGH) / 58.00; // (340m/s*1us)/2 と同等
    return distance;
}

```

- `pulseIn(pin, value)`

- pin: パルスを読み取りたい Arduino のピンの番号。許可されているデータ型 : int。
- value: 読み取りたいパルスのタイプ : HIGH または LOW。許可されているデータ型 : int。

ピン上のパルス (HIGH または LOW) を読み取ります。たとえば、value が HIGH の場合、`pulseIn()` はピンが LOW から HIGH に変わるのを待ち、タイミングを開始し、ピンが LOW になるのを待ってタイミングを停止します。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び & 共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なレビュー：新製品の発表や先行レビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

5.8 7. 手を追う車

この車をベットののように考えて、手を振ると車があなたの方に走ってきます。

必要な部品

このプロジェクトには、以下のコンポーネントが必要です。

キット全体を購入するのは確かに便利です。以下がリンクです：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	
<i>L9110 モータードライバーモジュール</i>	-
<i>TT モーター</i>	-
超音波モジュール	
障害物回避モジュール	

配線図

超音波モジュールと 2 つの IR 障害物回避モジュールを同時に接続します。

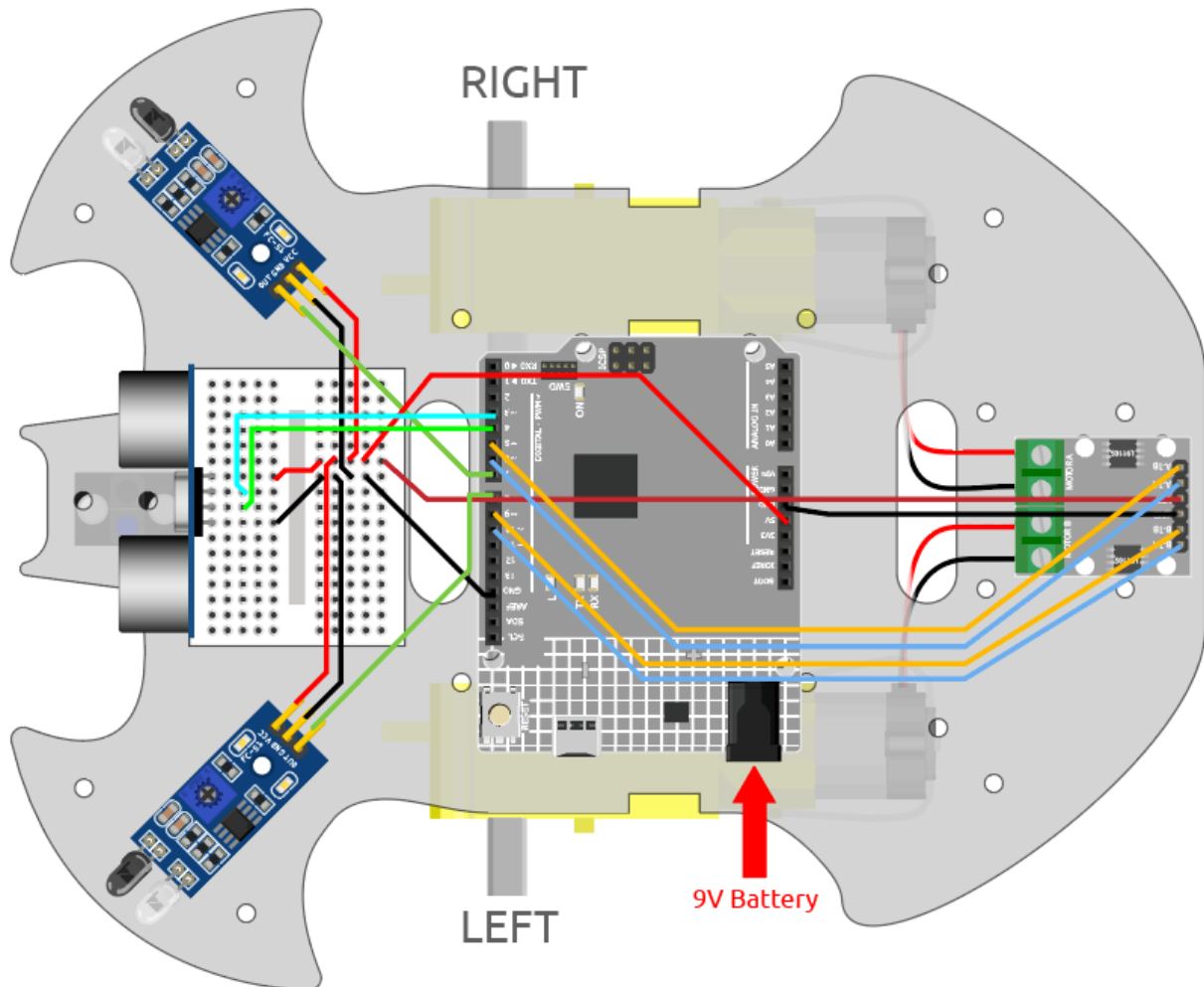
R4 ボードに超音波を次のように配線します。

超音波モジュール	R4 ボード
Vcc	5V
Trig	3
Echo	4
Gnd	GND

R4 ボードへの 2 つの IR 障害物回避モジュールの配線は次のとおりです。

左 IR モジュール	R4 ボード
OUT	8
GND	GND
VCC	5V

右 IR モジュール	R4 ボード
OUT	7
GND	GND
VCC	5V



コード

注釈:

- 3in1-kit\car_project\7.follow_your_hand のパスの下で 7.follow_your_hand.ino ファイルを開きます。
- または、このコードを **Arduino IDE** にコピーします。
- または、[Arduino ウェブエディター](#) を通じてコードをアップロードします。

コードが正常にアップロードされたら、車を地面に置きます。車の前で 5*10cm の距離で手を近づけると、車はあなたの手を前に追って進みます。IR 障害物モジュールの両側に手を近づけると、対応する方向に転向します。

動作原理

このプロジェクトは前の 2 つのプロジェクト、[6. 超音波モジュールで遊ぼう](#) と [5. 障害物回避モジュールを使って遊ぼう](#)、の組み合わせですが、実装された効果は異なります。前の 2 つのプロジェクトは障害物を後ろから検知

していましたが、このプロジェクトでは、あなたの手が前進または方向転換を追いかけるのを検知しています。このプロジェクトのワークフローは以下の通りです。

- 超音波モジュールによって検知された距離と、両方の赤外線モジュールの値を読み取ります。
- 距離が 5 ~ 10cm の場合、車をあなたの手と一緒に動かします。
- 左の IR モジュールがあなたの手を検知した場合、左に曲がります。
- 右の IR モジュールがあなたの手を検知した場合、右に曲がります。
- 赤外線モジュールも超音波モジュールもあなたの手を検知しない場合、車を停止させます。

```
void loop() {  
  
    float distance = readSensorData();  
  
    int left = digitalRead(leftIR);    // 0: 遮蔽物あり 1: 空  
    int right = digitalRead(rightIR);  
    int speed = 150;  
  
    if (distance>5 && distance<10){  
        moveForward(speed);  
    }  
    if(!left&&right){  
        turnLeft(speed);  
    }else if(left&&!right){  
        turnRight(speed);  
    }else{  
        stopMove();  
    }  
}
```

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。

- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

5.9 8. 自動運転車

このプロジェクトは、[6. 超音波モジュールで遊ぼう](#) および [5. 障害物回避モジュールを使って遊ぼう](#) の 2 つのプロジェクトの組み合わせです。2 つの赤外線障害物回避モジュールは短距離または端の検出を行い、超音波モジュールは長距離の検出を行い、自由に運転する過程で車が障害物にぶつからないことを確認します。

必要な部品

このプロジェクトには、以下のコンポーネントが必要です。

一式を購入するのは便利です。リンクはこちら：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
Arduino Uno R4 Minima	
L9110 モータードライバーモジュール	-
TT モーター	-
超音波モジュール	
障害物回避モジュール	

配線図

超音波モジュールと 2 つの IR 障害物回避モジュールを同時に接続します。

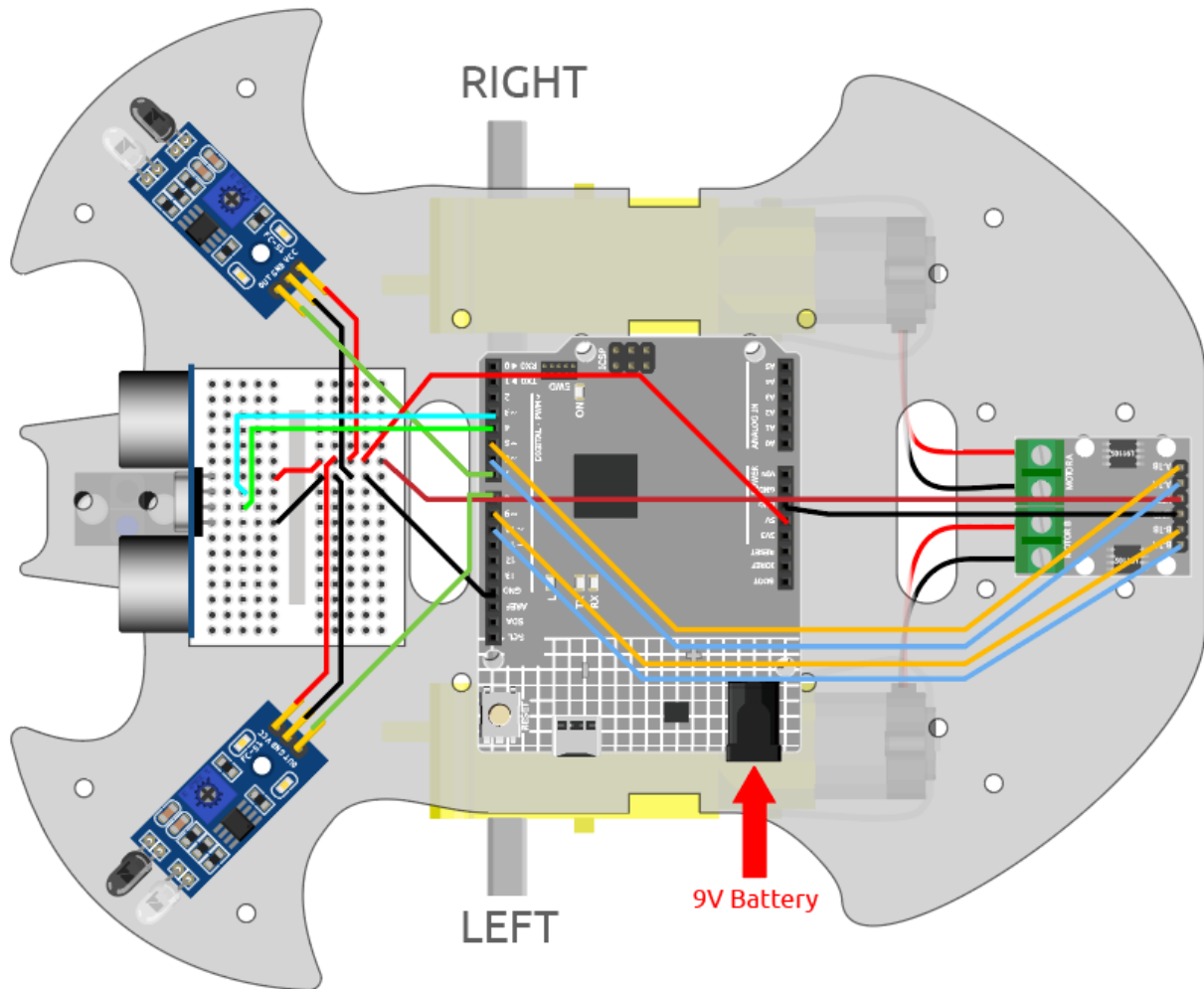
R4 ボードに超音波を以下のように接続します。

超音波モジュール	R4 ボード
Vcc	5V
Trig	3
Echo	4
Gnd	GND

R4 ボードへの 2 つの IR 障害物回避モジュールの接続方法は次の通りです。

左 IR モジュール	R4 ボード
OUT	8
GND	GND
VCC	5V

右 IR モジュール	R4 ボード
OUT	7
GND	GND
VCC	5V



コード

注釈:

- パス 3in1-kit\car_project\8.self_driving_car の下にある 8.self_driving_car.ino ファイルを開きます。
- または、このコードを **Arduino IDE** にコピーします。
- あるいは、[Arduino Web Editor](#) を通じてコードをアップロードします。

コードが正常にアップロードされると、車は自由に運転します。両側の IR 遮断モジュールが障害物を検出すると、緊急回避のために反対方向に移動します。車の正面 2 ~ 10cm 以内に障害物がある場合、車は左にバックアップして方向を調整し、前進します。

どのように動作するのか？

このプロジェクトのワークフローは以下の通りです。

- 左右の IR 障害物回避モジュールの値を優先的に読み取ります。
- 左の IR モジュールが 0 (障害物検出) で、右の IR モジュールが 1 の場合、車は左にバックアップします。
- 右の IR モジュールが 0 (障害物検出) の場合、車は右にバックアップします。
- 2 つの IR モジュールが同時に障害物を検出すると、車はバックアップします。
- それ以外の場合は、超音波モジュールによって検出された距離を読み取ります。
- 距離が 50cm 以上の場合、車を前進させます。
- 距離が 2-10cm の場合、車を旋回する前にバックアップします。
- 距離が 10-50cm の場合、車は低速で前進します。

```
void loop() {  
  
    int left = digitalRead(leftIR);    // 0: 遮蔽物あり 1: 空  
    int right = digitalRead(rightIR);  
  
    if (!left && right) {  
        backLeft(150);  
    } else if (left && !right) {  
        backRight(150);  
    } else if (!left && !right) {  
        moveBackward(150);  
    } else {  
        float distance = readSensorData();  
        Serial.println(distance);  
        if (distance > 50) { // 安全  
            moveForward(200);  
        } else if (distance < 10 && distance > 2) { // 注意  
            moveBackward(200);  
            delay(1000);  
            backLeft(150);  
            delay(500);  
        } else {  
            moveForward(150);  
        }  
    }  
}
```

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なレビュー：新製品の発表や先行レビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

5.10 9. リモートコントロール

このキットには IR 受信機が付属しており、IR リモートコントロールを使用して車の動きをコントロールすることができます。

必要な部品

このプロジェクトでは、以下のコンポーネントが必要です。

一式を購入するのが確実に便利です。リンクは以下の通りです。

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから個別に購入することもできます。

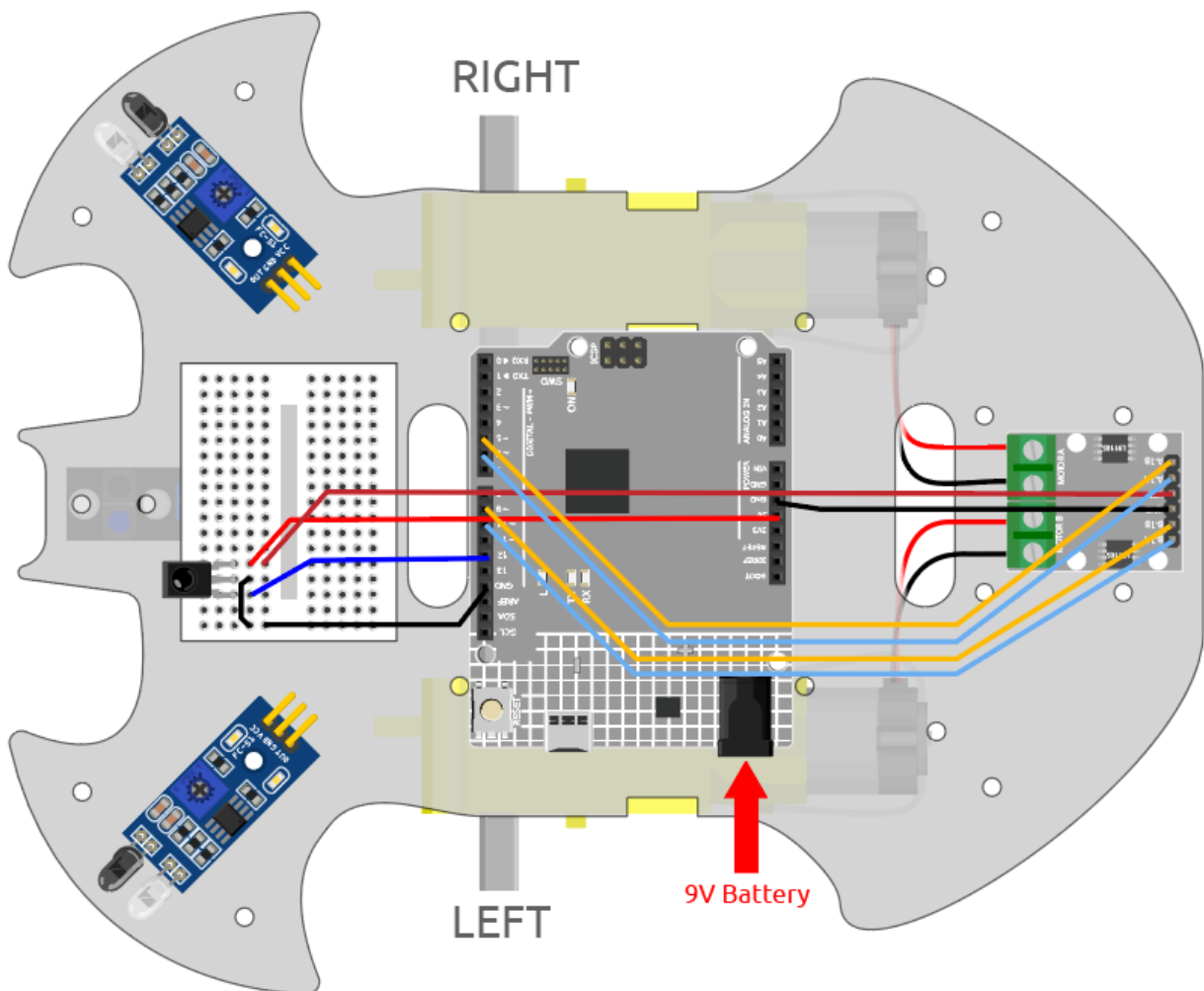
コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	
<i>L9110 モータードライバーモジュール</i>	-
<i>TT モーター</i>	-
<i>LED</i>	
<i>IR 受信機</i>	-

配線図

以下の図に従って回路を組み立ててください。

IR 受信機	R4 ボード
OUT	12
GND	GND
VCC	5V

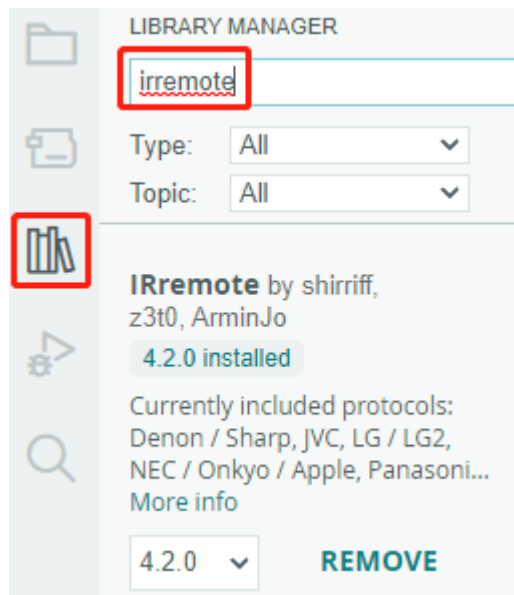
LED	R4 ボード
アノード (長いピン)	13
カソード	GND



コード

注釈:

- パス 3in1-kit\car_project\9.remote_control の下にある 9.remote_control.ino ファイルを開きます。
- または、このコードを **Arduino IDE** にコピーします。
- ここでは IRremote ライブラリが使用されています。 **Library Manager** からインストールできます。



コードが正常にアップロードされた後、リモートコントロールのボタンを押すと、LED が一度点滅して信号を受信したことを示し、あなたが押したボタンに応じて車が動きます。車をコントロールするための以下のキーを押すことができます。

- +: 加速
- -: 減速
- 1: 左へ前進
- 2: 前進
- 3: 右へ前進
- 4: 左折
- 6: 右折
- 7: 左へ後退
- 8: 後退

- 9: 右へ後退

どのように動作するのか？

このプロジェクトの効果は、IR リモートコントロールのキー値を読み取ることで車を動かすことです。さらに、IR 信号が正常に受信されたことを示すために LED が追加されています。

1. IRremote ライブラリをインポートし、**Library Manager** からインストールできます。

```
#include <IRremote.h>

const int IR_RECEIVE_PIN = 12;  // IR センサのピン番号を定義
```

2. ボーレート 9600 でシリアル通信を初期化します。指定されたピン (IR_RECEIVE_PIN) で IR 受信機を初期化し、LED フィードバックを有効にします (該当する場合)。

```
...

void setup() {

    ...
    //IR リモート
    IrReceiver.begin(IR_RECEIVE_PIN, ENABLE_LED_FEEDBACK);  // IR レシーバを開始
    Serial.println("REMOTE CONTROL START");
}

}
```

3. リモコンのキーを押すと、赤外線レシーバはどのキーが押されたかを知り、それに応じて車に対応するキー値に従って動きます。

```
void loop() {

    if (IrReceiver.decode()) {
        // Serial.println(results.value, HEX);
        String key = decodeKeyValue(IrReceiver.decodedIRData.command);
        if (key != "ERROR") {
            Serial.println(key);

            if (key == "+") {
                speed += 50;
            } else if (key == "-") {
                speed -= 50;
            }
        }
    }
}
```

(次のページに続く)

(前のページからの続き)

```
    } else if (key == "2") {  
        moveForward(speed);  
        delay(1000);  
        ...  
    }  
    IrReceiver.resume(); // Enable receiving of the next value  
}  
}
```

- IR 信号が受信され、正常にデコードされたかどうかを確認します。
- カスタムの decodeKeyValue() 関数を使用して IR コマンドをデコードし、key に保存します。
- デコードされた値がエラーでないか確認します。
- デコードされた IR 値をシリアルモニタに出力します。
- 次の信号の IR 信号受信を再開します。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

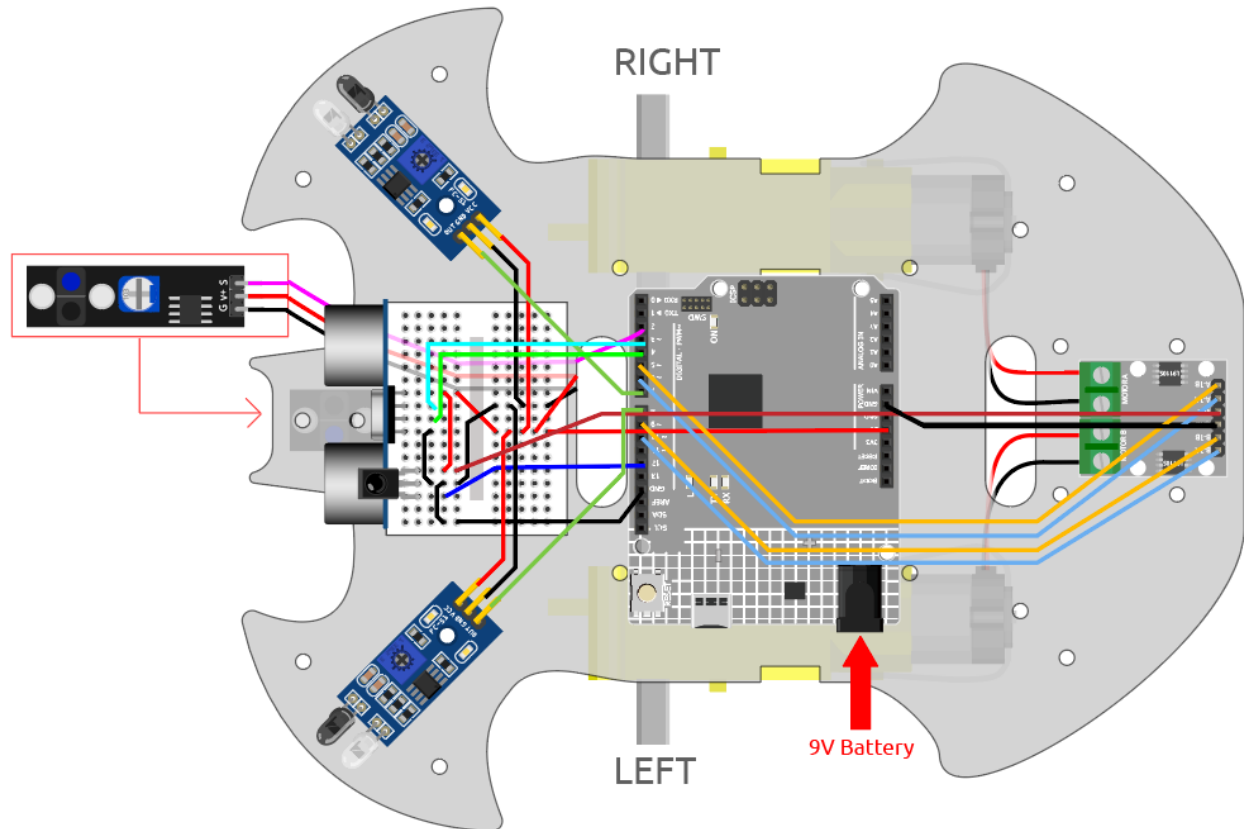
- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

5.11 10. ワンタッチスタート

このプロジェクトでは、これまでのプロジェクト - ライン追従、追跡、障害物回避、自動運転などを統合しました。リモコンのボタンで切り替えが可能で、車をスタートさせて一度にすべての機能を体験することができます。

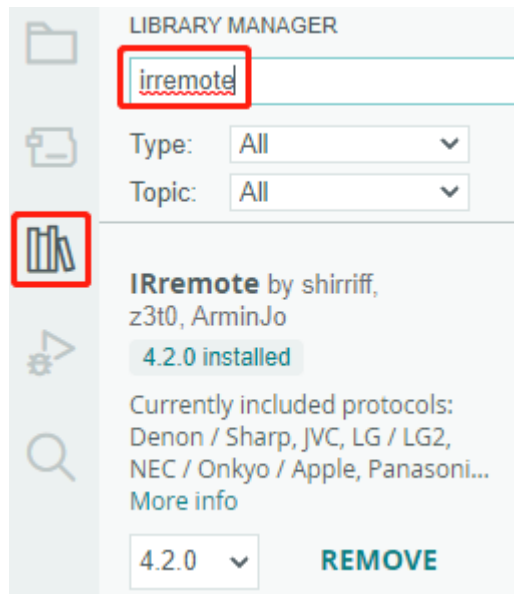
配線図



コード

注釈:

- パス 3in1-kit\car_project\10.one_touch_start の下の 10.one_touch_start.ino ファイルを開きます。
- または、このコードを **Arduino IDE** にコピーします。
- ここで IRremote ライブラリが使用されています。 **Library Manager** からインストールできます。



コードが正常にアップロードされた後、IR レシーバがリモコンからの信号を受信するたびに LED が 3 回高速で点滅します。以下のキーを押してカーンを操作できます。

- +: 加速
- -: 減速
- 1: 左前に移動
- 2: 前進
- 3: 右に移動
- 4: 左折
- 6: 右折
- 7: 左に後退
- 8: 後退。
- 9: 右に後退
- CYCLE: ラインを追跡
- U/SD: 自動運転
- |: 超音波モジュールを使用した障害物回避
- |: IR 障害物モジュールを使用した障害物回避
- EQ: あなたの手を追跡

- 0: 停止

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なレビュー：新製品の発表や先行レビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

5.12 11. 速度のキャリブレーション

車を前進させる際、車がまっすぐ進まないことがあります。これは、工場出荷時に 2 つのモーターの速度が同じでないためです。しかし、2 つのモーターにオフセットを書き込むことで、その回転速度を一致させることができます。

このプロジェクトでは、オフセットを **EEPROM** に保存する方法を学びます。このポイントは、キャリブレーションの後、すべてのプロジェクトが EEPROM から直接オフセット値を取得できるため、車がスムーズにまっすぐ進むことができます。

配線図

このプロジェクトの配線は、[2. コードによる移動](#) と同じです。

遊び方は？

1. パス 3in1-kit\car_project\11.speed_calibration の下の 11.speed_calibration.ino ファイルを開く。または、このコードを **Arduino IDE** にコピーします。
 2. コードが正常にアップロードされた後、車に 9V のバッテリーを接続し、地面に置いて前進させ、どちらの方向にオフセットされているかを確認します。
- 車が左前方に移動する場合、右のモーターの速度が早すぎることを意味し、減速する必要があります。

```
EEPROM.write(1, 100) // 1 は右モーターを意味し、100 は速度が 100% であることを意味します。  
実際の状況に応じて 90、95 などに設定できます。
```

- 車が右に移動する場合、左のモーターの速度が早すぎることを意味し、減速する必要があります。

`EEPROM.write(0, 100)` // 0は左モーターを意味し、100は速度が100%であることを意味します。実際の状況に応じて90、95などに設定できます。

3. コードを修正した後、R4 ボードにコードをアップロードして効果を確認します。車がほぼまっすぐになるまで、上記の手順を繰り返します。
4. このオフセットはEEPROMに記録されます。他のプロジェクトで使用する際には、このオフセットを読むだけでよく、例として5. 障害物回避モジュールを使って遊ぼうを参照してください。

```
#include <EEPROM.h>

float leftOffset = 1.0;
float rightOffset = 1.0;

const int A_1B = 5;
const int A_1A = 6;
const int B_1B = 9;
const int B_1A = 10;

const int rightIR = 7;
const int leftIR = 8;

void setup() {
  Serial.begin(9600);

  //motor
  pinMode(A_1B, OUTPUT);
  pinMode(A_1A, OUTPUT);
  pinMode(B_1B, OUTPUT);
  pinMode(B_1A, OUTPUT);

  //IR obstacle
  pinMode(leftIR, INPUT);
  pinMode(rightIR, INPUT);

  leftOffset = EEPROM.read(0) * 0.01; //右モーターのオフセットを読み取る
  rightOffset = EEPROM.read(1) * 0.01; //右モーターのオフセットを読み取る
}
```

(次のページに続く)

(前のページからの続き)

```
void loop() {

    int left = digitalRead(leftIR);  // 0: 遮蔽物あり 1: 空
    int right = digitalRead(rightIR);
    int speed = 150;

    if (!left && right) {
        backLeft(speed);
    } else if (left && !right) {
        backRight(speed);
    } else if (!left && !right) {
        moveBackward(speed);
    } else {
        moveForward(speed);
    }
}

void moveForward(int speed) {
    analogWrite(A_1B, 0);
    analogWrite(A_1A, int(speed * leftOffset));
    analogWrite(B_1B, int(speed * rightOffset));
    analogWrite(B_1A, 0);
}

void moveBackward(int speed) {
    analogWrite(A_1B, speed);
    analogWrite(A_1A, 0);
    analogWrite(B_1B, 0);
    analogWrite(B_1A, speed);
}

void backLeft(int speed) {
    analogWrite(A_1B, speed);
    analogWrite(A_1A, 0);
    analogWrite(B_1B, 0);
    analogWrite(B_1A, 0);
}

void backRight(int speed) {
```

(次のページに続く)

(前のページからの続き)

```
analogWrite(A_1B, 0);  
analogWrite(A_1A, 0);  
analogWrite(B_1B, 0);  
analogWrite(B_1A, speed);  
}
```

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

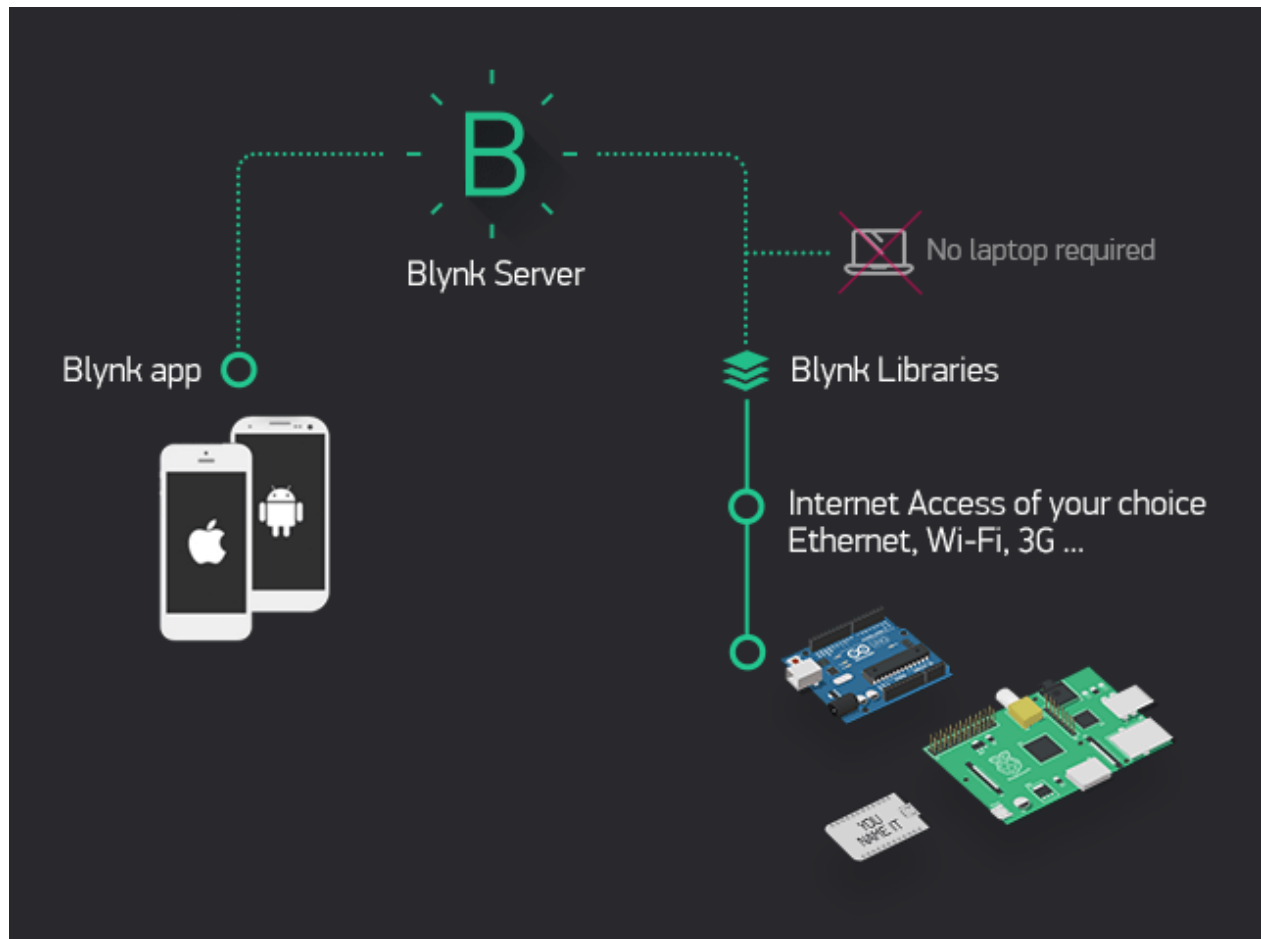
第 6 章

IoT プロジェクト

このキットには、Arduino が IoT 実験のためにインターネットに接続することを可能にする ESP8266 Wifi モジュールが含まれています。

ここでは、ESP8266 Wifi モジュールを使用して Arduino を [BLYNK](#) プラットフォームに接続する方法をガイドします。これにより、興味深い IoT プロジェクトを実現できます。また、携帯電話の Blynk APP を使用してスマートカーを操作することもできます。

Blynk は、個人の IoT プロジェクトから数百万の商用接続製品まで、任意の規模での接続された電子デバイスのプロトタイプ制作、デプロイ、遠隔管理に必要なソフトウェアのフルセットです。Blynk を使用すると、誰でもハードウェアをクラウドに接続し、コードなしで iOS、Android、およびウェブアプリケーションを構築して、デバイスからのリアルタイムおよび履歴データを分析し、世界中のどこからでも遠隔操作することができます。また、重要な通知を受け取ることができ、さらに多くのことができます...



注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

6.1 1. Blynk でのスタート

R4 ボードを Blynk と通信させるためには、Blynk を初めて使用する際にいくつかの設定が必要です。

以下の手順に従ってください。章をスキップせず、順番に実行することが必要です。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

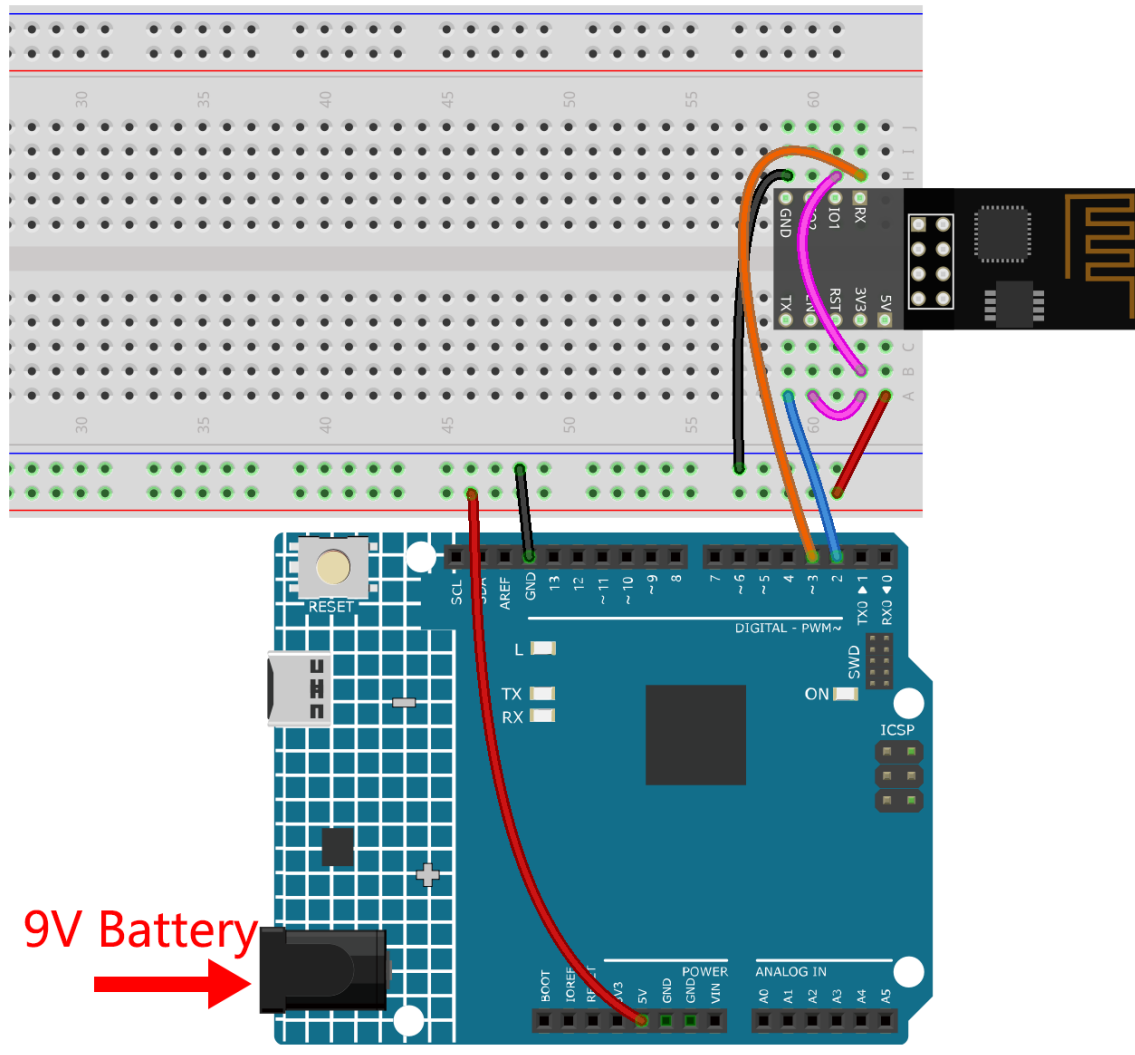
- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

6.1.1 1.1 ESP8266 の設定

キットに含まれる ESP8266 モジュールはすでに AT ファームウェアで事前書き込まれていますが、以下の手順に従ってその設定を変更する必要があります。

1. 回路を組む。



2. 3in1-kit\iot_project\1.set_software_serial のパスの下に 1.set_software_serial.ino ファイルを開く。または、このコードを **Arduino IDE** にコピーします。

```
#include <SoftwareSerial.h>
SoftwareSerial espSerial(2, 3); //Rx,Tx

void setup() {
  // セットアップ コードをここに配置して、1 回実行します。
  Serial.begin(115200);
  espSerial.begin(115200);
}

void loop() {
  if (espSerial.available()) {
    Serial.write(espSerial.read());
  }
}
```

(次のページに続く)

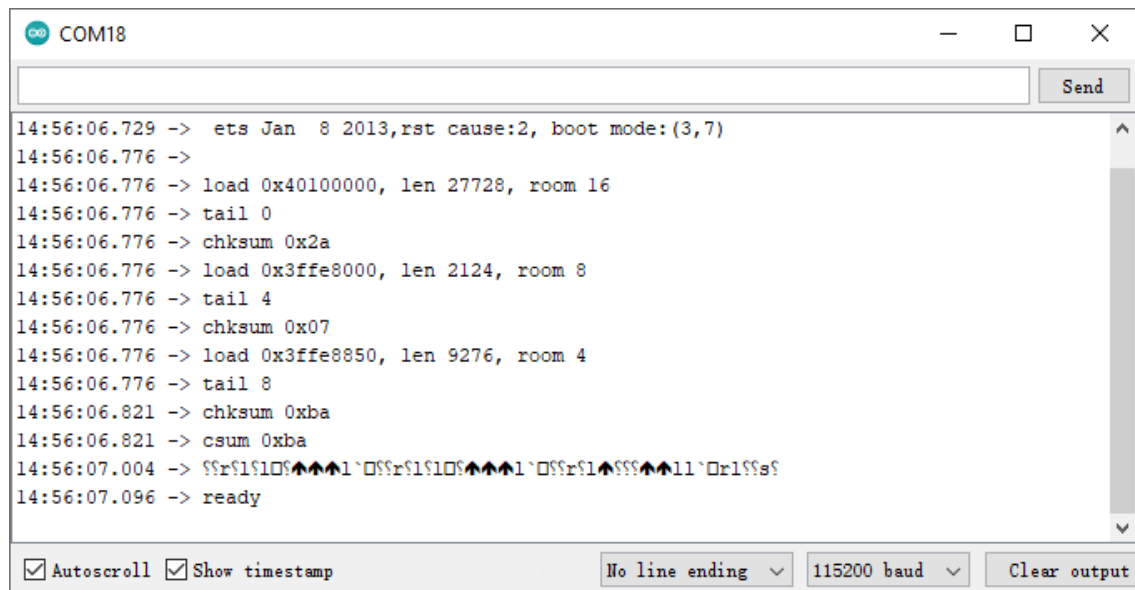
(前のページからの続き)

```

    }
    if (Serial.available()) {
        espSerial.write(Serial.read());
    }
}

```

3. 右上隅の虫眼鏡のアイコン (シリアルモニター) をクリックして、ボーレートを **115200** に設定します。(私のように何か情報が表示されるかもしれませんが、表示されないかもしれませんが、問題ありません。次の手順に進んでください。)

**警告:**

- ready が表示されない場合、ESP8266 モジュールをリセットして (RST を GND に接続) シリアルモニターを再度開くことができます。
- さらに、結果が OK の場合、ファームウェアを再度書き込む必要があるかもしれません。詳細については [ESP8266 モジュールのファームウェアの再書き込み方法は?](#) を参照してください。それでも解決できない場合は、シリアルモニターのスクリーンショットを sevice@sunfounder.com に送信してください。できるだけ早く問題を解決するお手伝いをいたします。

4. **NEWLINE DROPDOWN BOX** をクリックして、ドロップダウンオプションで both NL & CR を選択し、AT を入力します。OK と返されると、ESP8266 がボードとの接続に成功したことを意味します。

```

COM18
AT2
14:56:06.776 -> load 0x40100000, len 27728, room 16
14:56:06.776 -> tail 0
14:56:06.776 -> chksum 0x2a
14:56:06.776 -> load 0x3ffe8000, len 2124, room 8
14:56:06.776 -> tail 4
14:56:06.776 -> chksum 0x07
14:56:06.776 -> load 0x3ffe8850, len 9276, room 4
14:56:06.776 -> tail 8
14:56:06.821 -> chksum 0xba
14:56:06.821 -> csum 0xba
14:56:07.004 -> ?r?l?l? ????l`?r?l?l? ????l`?r?l?l? ????l`?r?l?l?
14:56:07.096 -> ready
15:00:05.119 -> AT
15:00:05.119 ->
15:00:05.119 -> OK

```

Autoscroll Show timestamp Both NL & CR 115200 baud Clear output

5. AT+CWMODE=3 を入力すると、管理モードが **Station and AP** の共存に変更されます。

```

COM18
AT+CWMODE=3
14:56:06.776 -> tail 4
14:56:06.776 -> chksum 0x07
14:56:06.776 -> load 0x3ffe8850, len 9276, room 4
14:56:06.776 -> tail 8
14:56:06.821 -> chksum 0xba
14:56:06.821 -> csum 0xba
14:56:07.004 -> ?r?l?l? ????l`?r?l?l? ????l`?r?l?l? ????l`?r?l?l?
14:56:07.096 -> ready
15:00:05.119 -> AT
15:00:05.119 ->
15:00:05.119 -> OK
15:01:08.226 -> AT+CWMODE=3
15:01:08.319 ->
15:01:08.319 -> OK

```

Autoscroll Show timestamp Both NL & CR 115200 baud Clear output

6. 後でソフトウェアシリアルを使用するために、AT+UART=9600,8,1,0,0 を入力して、ESP8266 のボーレートに 9600 に変更する必要があります。

```

COM18
AT+UART=9600,8,1,0,0
14:56:06.776 -> tail 8
14:56:06.821 -> chksum 0xba
14:56:06.821 -> csum 0xba
14:56:07.004 -> $$$r$1$10$####1`0$$$r$1$10$####1`0$$$r$1$10$####11`0r1$$$?
14:56:07.096 -> ready
15:00:05.119 -> AT
15:00:05.119 ->
15:00:05.119 -> OK
15:01:08.226 -> AT+CWMODE=3
15:01:08.319 ->
15:01:08.319 -> OK
15:03:06.853 -> AT+UART=9600,8,1,0,0
15:03:06.853 ->
15:03:06.853 -> OK

```

☒ Autoscroll ☒ Show timestamp Both NL & CR 115200 baud Clear output

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

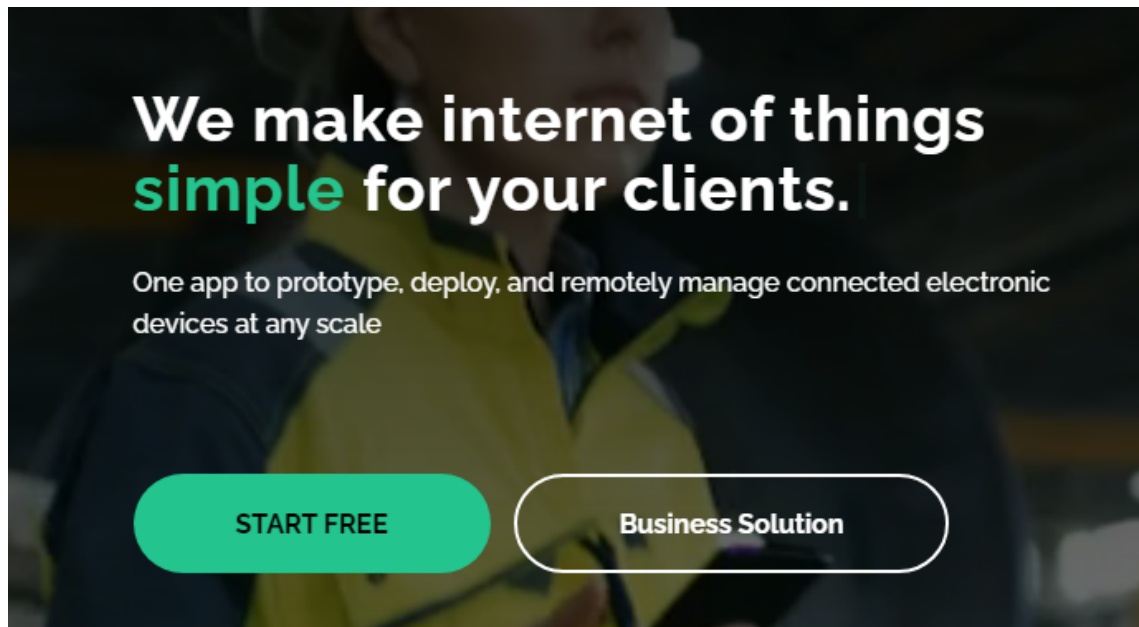
参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

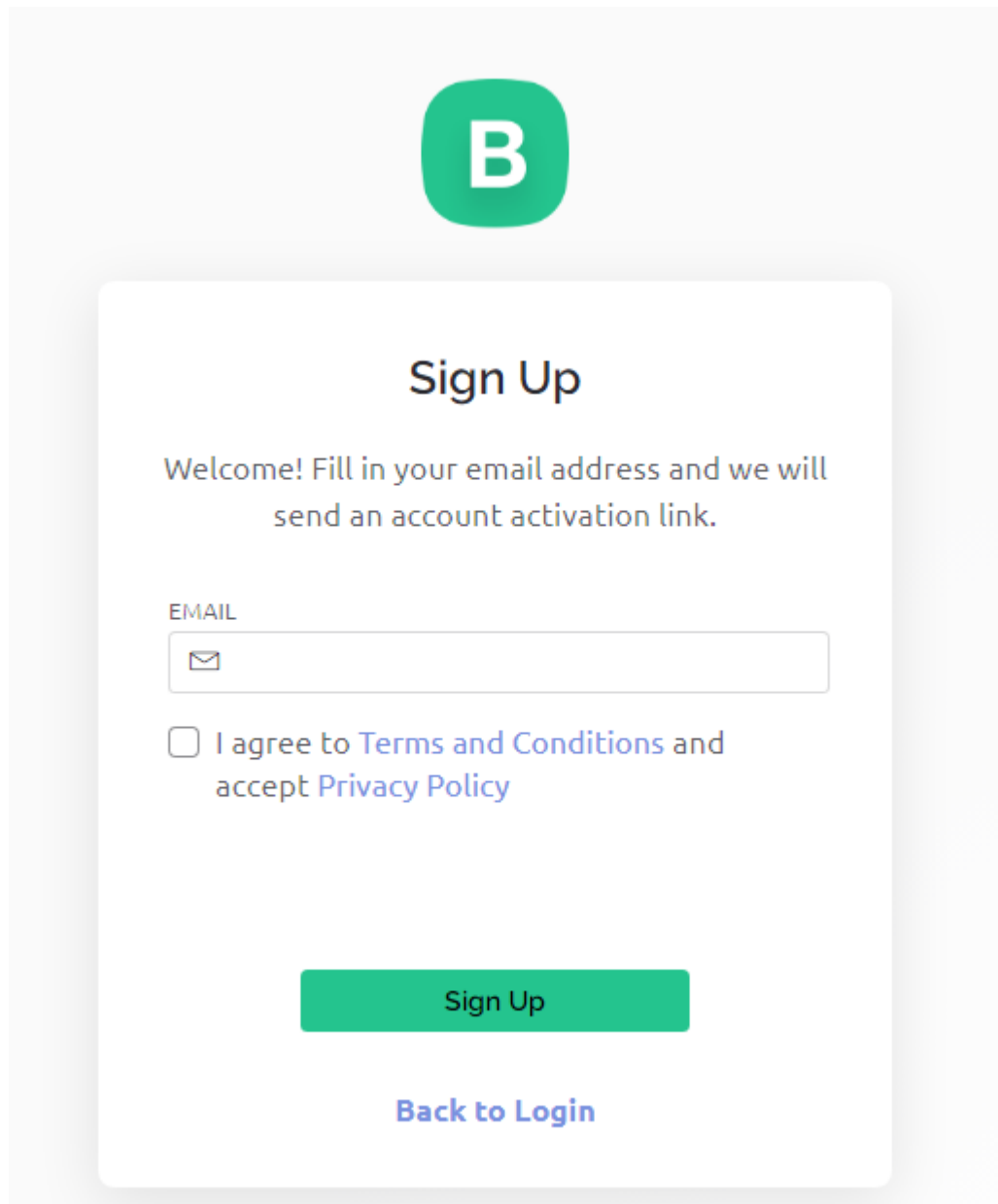
私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

6.1.2 1.2 Blynk の設定手順

1. [BLYNK](#) サイトを開き、**START FREE** ボタンを押下します。



2. メールアドレスを入力し、アカウントを作成します。



The image shows a 'Sign Up' form for Blynk. At the top is a green rounded square logo with a white letter 'B'. Below the logo, the title 'Sign Up' is centered. A welcome message follows: 'Welcome! Fill in your email address and we will send an account activation link.' There is an email input field with a placeholder icon of an envelope. Below the input field is a checkbox with the text 'I agree to Terms and Conditions and accept Privacy Policy'. At the bottom of the form is a green 'Sign Up' button and a blue 'Back to Login' link.

B

Sign Up

Welcome! Fill in your email address and we will send an account activation link.

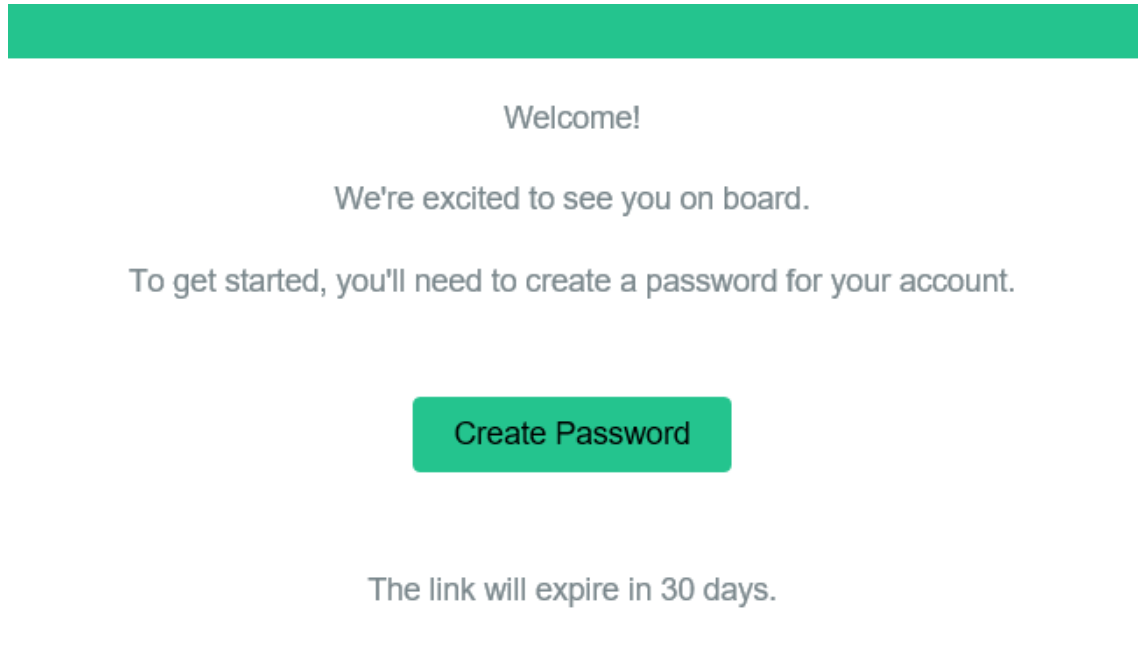
EMAIL

☐ I agree to [Terms and Conditions](#) and accept [Privacy Policy](#)

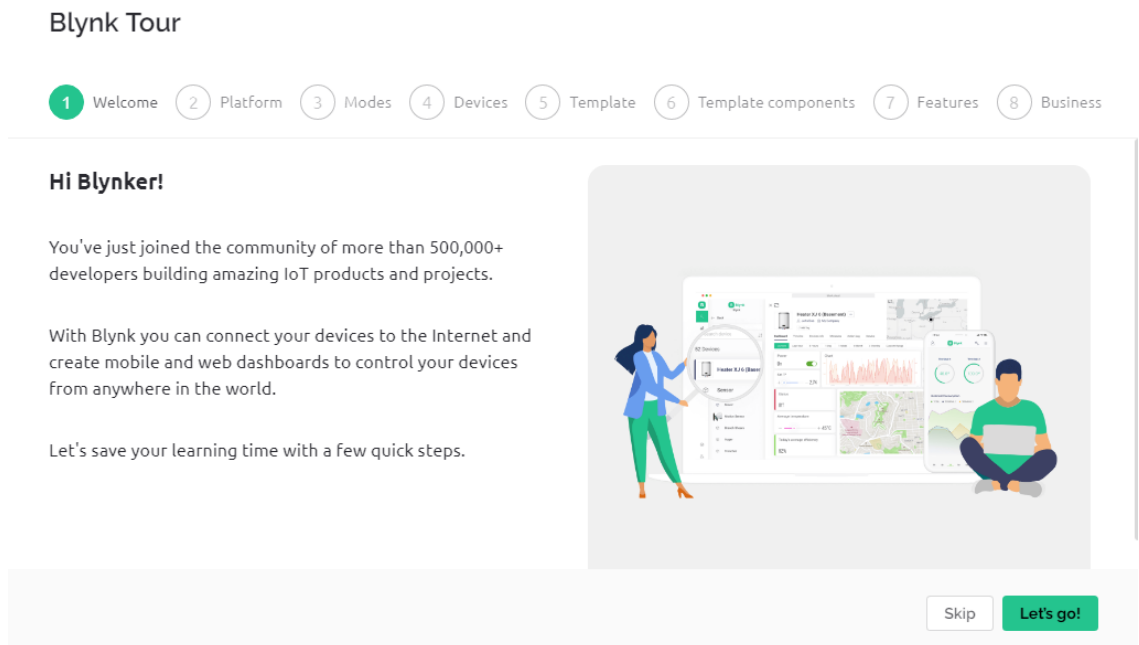
Sign Up

[Back to Login](#)

3. 登録したメールアドレスを開き、案内に従ってアカウントの確認を行います。



4. アカウント確認後、**Blynk Tour** というガイドが表示されます。これを読むことで Blynk の基本情報を得ることができます。



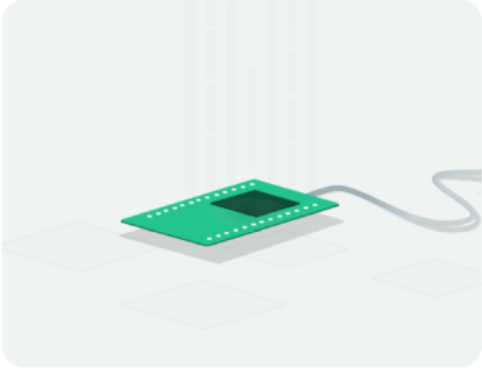
5. 次に、**Quick Start** ガイドを利用してテンプレートとデバイスを作成します。そのため、**Let's go** ボタンを押下します。

Quickstart

This is a step by step guide to get your first device online and start controlling it from anywhere in the world in **less than 5 minutes**

What you will need:

- Supported hardware. Check the full list of supported hardware [here](#).
- IDE. You can use Arduino IDE or PlatformIO or any other editor.
- Blynk Library
- It will be beneficial if you already know how to upload code to your hardware.



Cancel **Let's go!**

6. 用意したハードウェアと接続タイプを選択します。

Quickstart

1 Hardware — 2 IDE — 3 Blynk Library — 4 Code — 5 Device activation

Which hardware are you using?

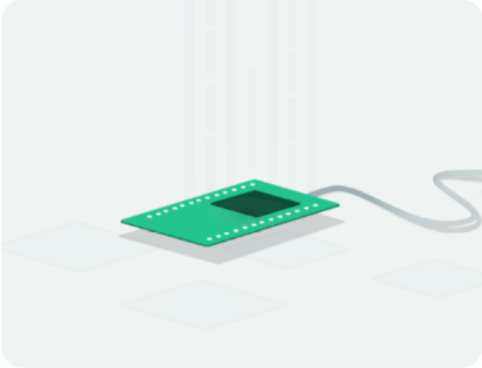
We will help you prepare the code for you board

ESP8266

What is your device connectivity type

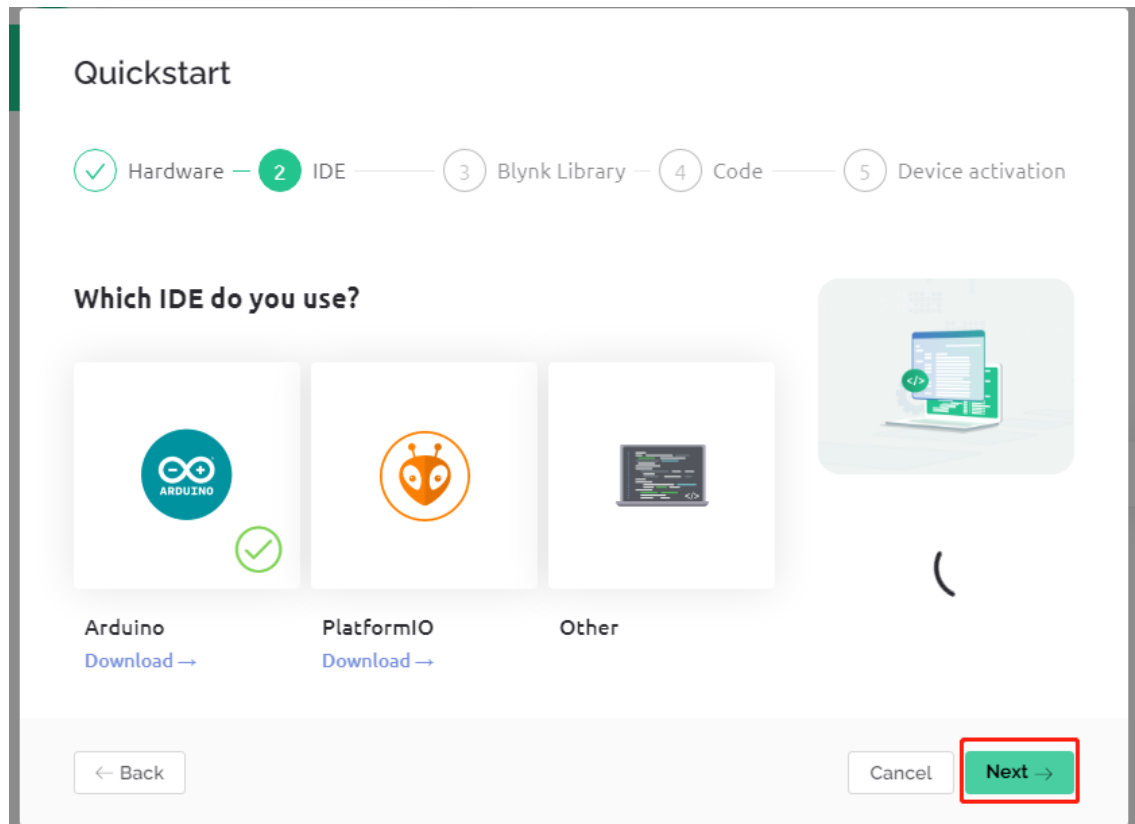
Blynk supports various connection types (BLE is not supported yet).

WiFi



Cancel **Next →**

7. 必要な開発環境（IDE）についての案内が表示されますが、私たちの推奨は **Arduino IDE** です。



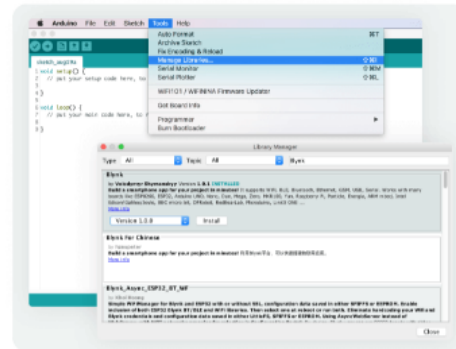
8. 必要とされるライブラリ一覧が表示されます。しかし、こちらの推奨ライブラリには課題があるため、後述する方法で別のライブラリを手動で追加する必要があります。こちらの画面では **Next** ボタンを押下して、新たなテンプレートとデバイスの作成を進めます。

Quickstart



Install Blynk Library for Arduino

1. Go to Arduino -Tools - Manage Libraries...
2. Search for Blynk there.
3. Choose the latest version and press Install.



← Back

Cancel

Next →

9. 次のステップでは、関連するコードをアップロードし、Blynk にボードを接続する作業が行われます。しかし、先述したライブラリの課題のため、再度ライブラリの追加が必要です。そのため、この **Quick Start** ガイドを一時停止するため **Cancel** ボタンを押下します。

Quickstart

☒ Hardware — ☒ IDE — ☒ Blynk Library — **4** Code — ☐ 5 Device activation

Here is a code for your device

1. Enter your Wi-Fi network SSID (name) and password to connect your device.

* We never store or send this information anywhere. It's only used to generate the firmware code. You can leave these fields empty and manually add WiFi credentials in your sketch.

Wi-Fi Network Name (SSID)

Password

```

/***** Copy code Download As File *****/

This is a simple demo of sending and receiving some data.
Be sure to check out other examples!
*****/

// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID      "TMPLDbHgRCnL"
#define BLYNK_DEVICE_NAME      "Quickstart Device"
#define BLYNK_AUTH_TOKEN       "zsWQM55SheRPSoTjCgmPz0Zhpj9Mr4ivv"

// Comment this out to disable prints and save space
// #define BLYNK_PRINT true
  
```

10. **Search** ボタンを押下すると、先程作成したデバイスの情報が表示されます。

B

My organization - 3901HS

DEVICES

My devices 1

All 1

LOCATIONS

My locations 0

All 0

My devices

+ New Device

1 Device

Device name

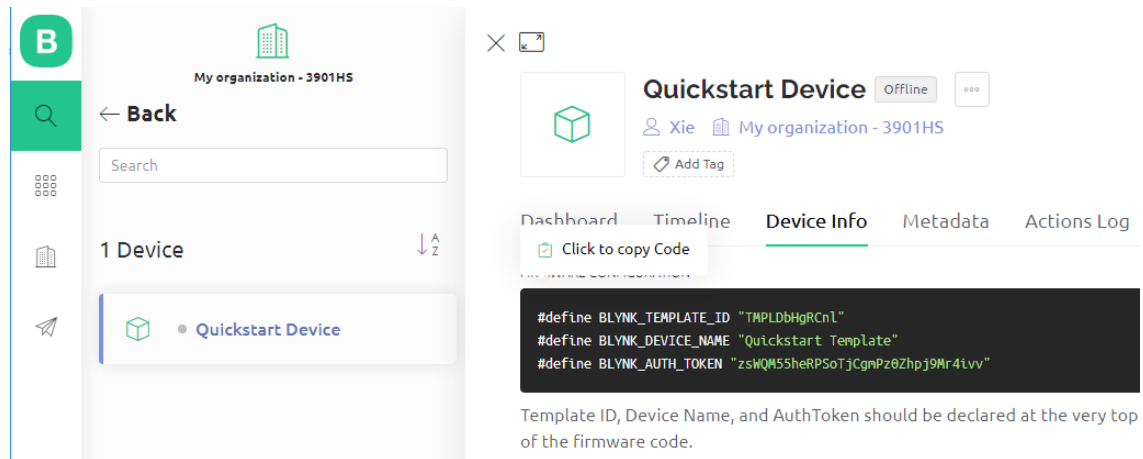
Device

Actions

Quickstart Device

Xie

11. 表示された **Quickstart Device** ページに移動し、**Device info** 欄に **TEMPLATE_ID**、**DEVICE_NAME**、及び **AUTH_TOKEN** の情報が記載されています。これらの情報は後ほどコピーが必要となりますので、控えておきます。



注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。




私たちと一緒に探索し、創造する準備はできていますか？ [] をクリックして今すぐ参加しましょう！

6.1.3 1.3 必要なライブラリの追加

Blynk を Arduino IDE で使用するために、正しいライブラリを追加する必要があります。

1. [HERE](#) をクリックして、ページの最下部にスクロールして、最初の .zip ファイルをダウンロードします。

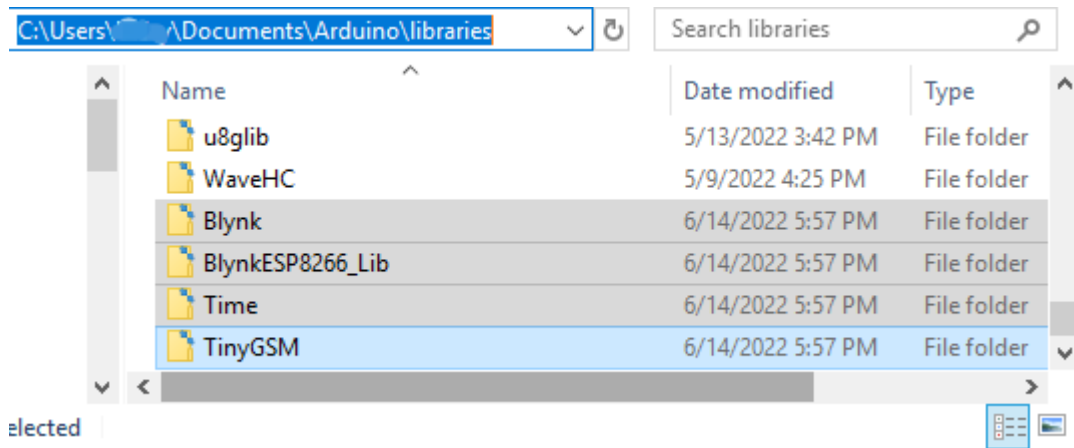
▼ Assets 3

 Blynk_Release_v1.1.0.zip	779 KB	22 days ago
 Source code (zip)		22 days ago
 Source code (tar.gz)		22 days ago

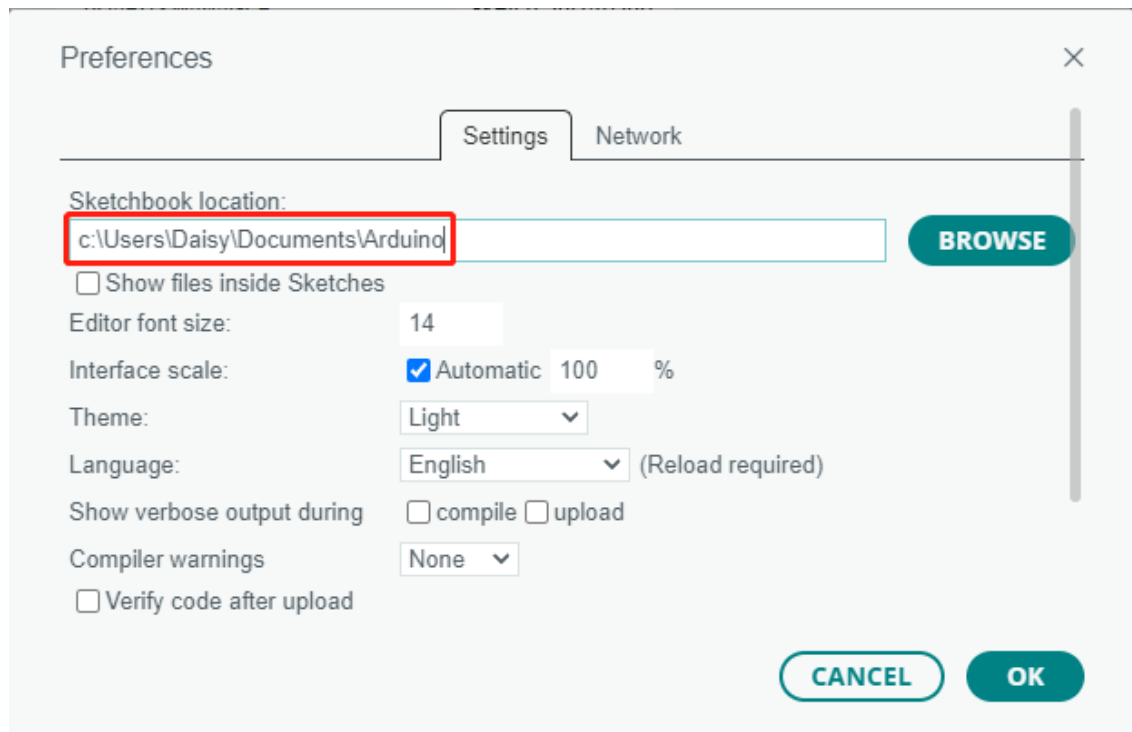
2. このファイルを解凍すると、以下のフォルダが表示されます。

..			File folder	
Blynk	1,060,482	459,412	File folder	7/16/2021 1:13 ...
BlynkESP8266_Lib	57,090	11,208	File folder	5/25/2021 5:12 ...
Time	63,774	25,512	File folder	5/25/2021 5:12 ...
TinyGSM	602,241	172,967	File folder	5/25/2021 5:12 ...

3. これらすべてをコピーし、通常 C:\Users\xxx\Documents\Arduino\libraries にある Arduino IDE のデフォルトのライブラリディレクトリに貼り付けます。



4. ライブラリのディレクトリが異なる場合は、**File -> Preferences** に移動して確認することができます。



注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook

上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

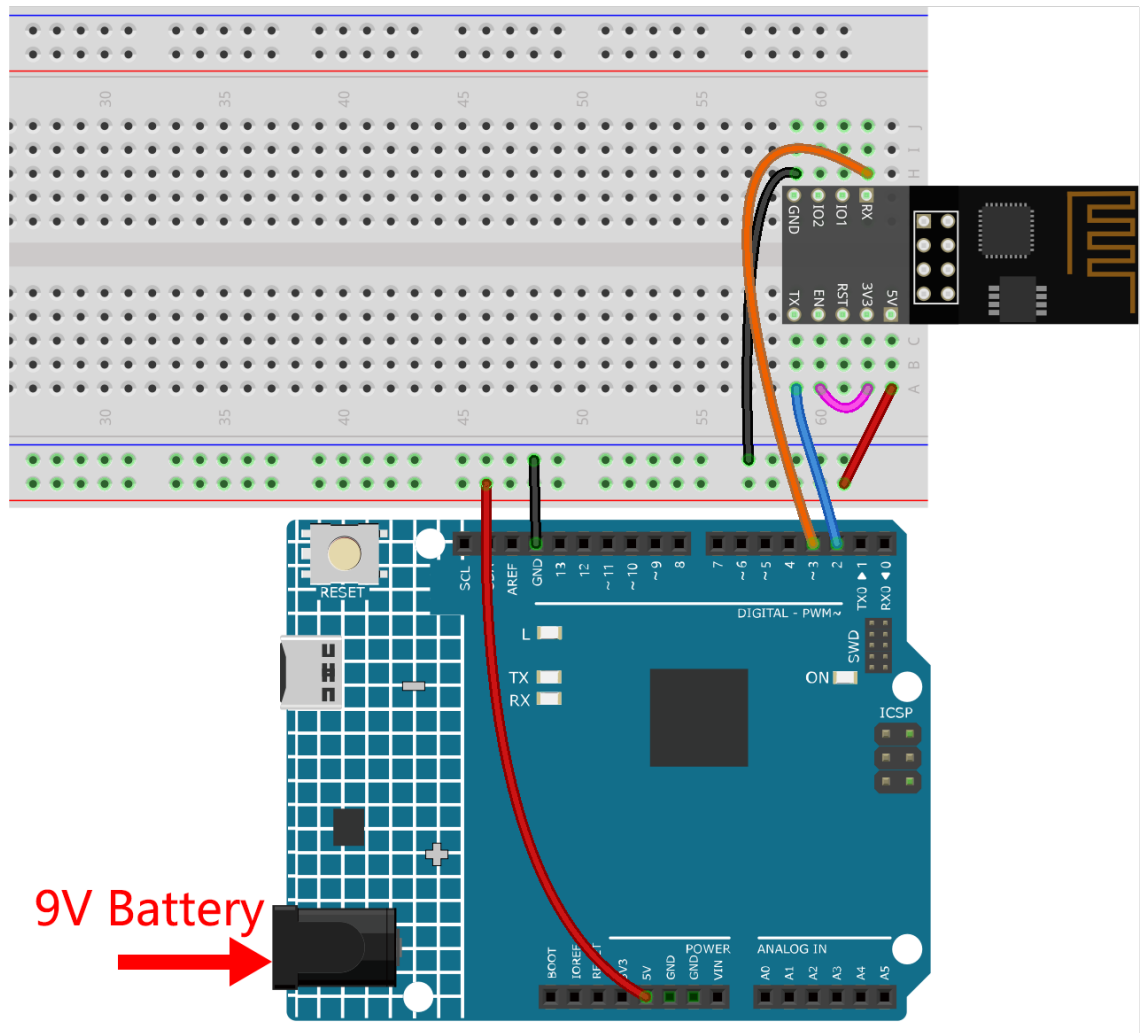
- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

6.1.4 1.4 R4 ボードを Blynk に接続する

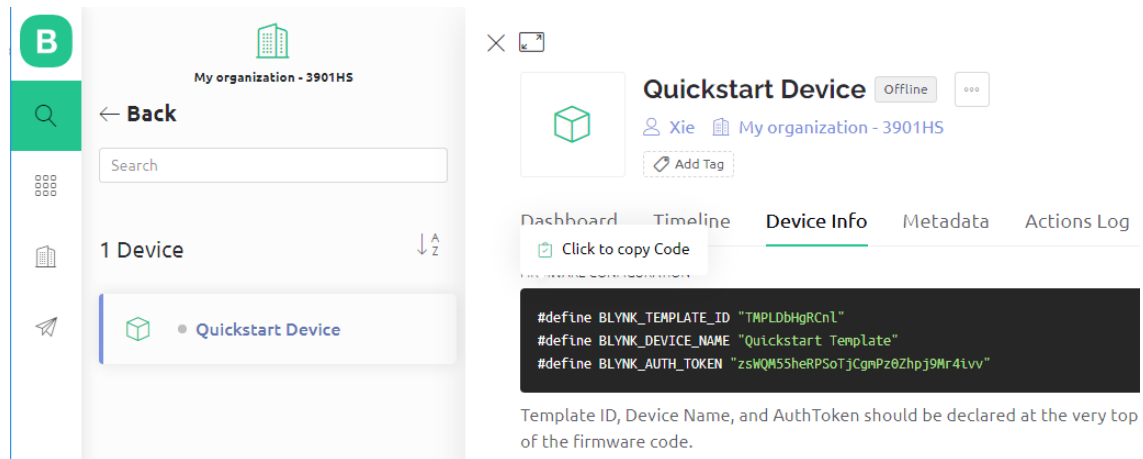
1. ESP8266 モジュールと R4 ボードを再接続します。ここではソフトウェアシリアルを使用しているため、TX と RX はそれぞれ R4 ボードの 2 番ピンと 3 番ピンに接続されます。

注釈：ESP8266 モジュールは安定した動作環境を提供するために高電流が必要ですので、9V の電池が接続されていることを確認してください。



2. 3in1-kit\iot_project\1.connect のパスの下にある 1.connect.ino ファイルを開くか、このコードを **Arduino IDE** にコピーします。
3. あなたのアカウントの **Device info** ページからコピーできる以下の 3 行のコードを置き換えます。これら 3 行のコードにより、R4 ボードがあなたの blynk アカウントを見つけることができます。

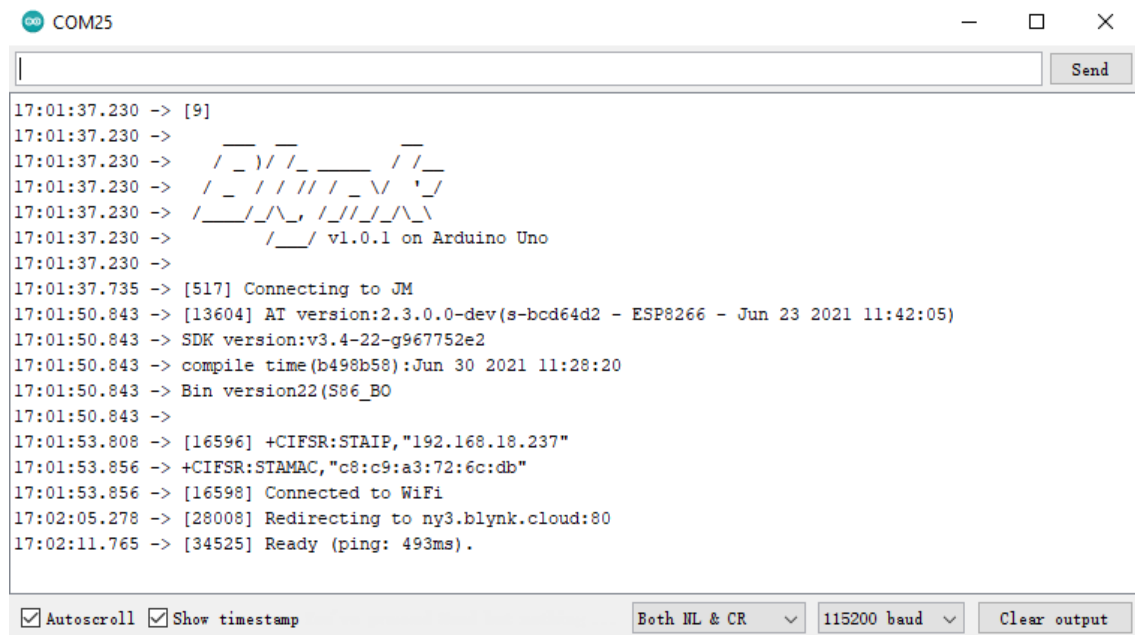
```
#define BLYNK_TEMPLATE_ID "TMPLxxxxxx"
#define BLYNK_DEVICE_NAME "Device"
#define BLYNK_AUTH_TOKEN "YourAuthToken"
```



4. 使用中の WiFi の ssid および password を入力します。

```
char ssid[] = "ssid";
char pass[] = "password";
```

5. コードを R4 ボードにアップロードし、シリアルモニタを開いてボーレートを 115200 に設定します。R4 ボードが Blynk と正常に通信すると、シリアルモニタに ready 文字が表示されます。



注釈: 接続時に ESP is not responding というメッセージが表示される場合は、以下の手順に従ってください。

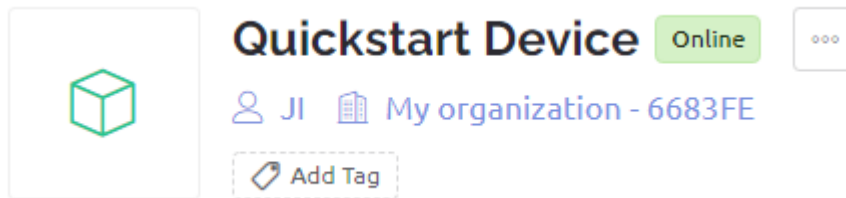
- 9V の電池が接続されていることを確認します。
- ESP8266 モジュールの RST ピンを 1 秒間 GND に接続してリセットし、その後、それを取り外し

ます。

- R4 ボードのリセットボタンを押します。

これらの操作を 3-5 回繰り返す必要がある場合があります。ご了承ください。

6. Blynk のステータスが **offline** から **online** に変わります。



注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

6.2 2. Blynk からデータを取得する

この章では、Blynk を使用して回路を制御する方法を学びます。インターネットを介して LED を点灯させましょう！

必要な部品

このプロジェクトでは、以下のコンポーネントが必要です。

キット全体を購入すると非常に便利です。こちらがリンクです：

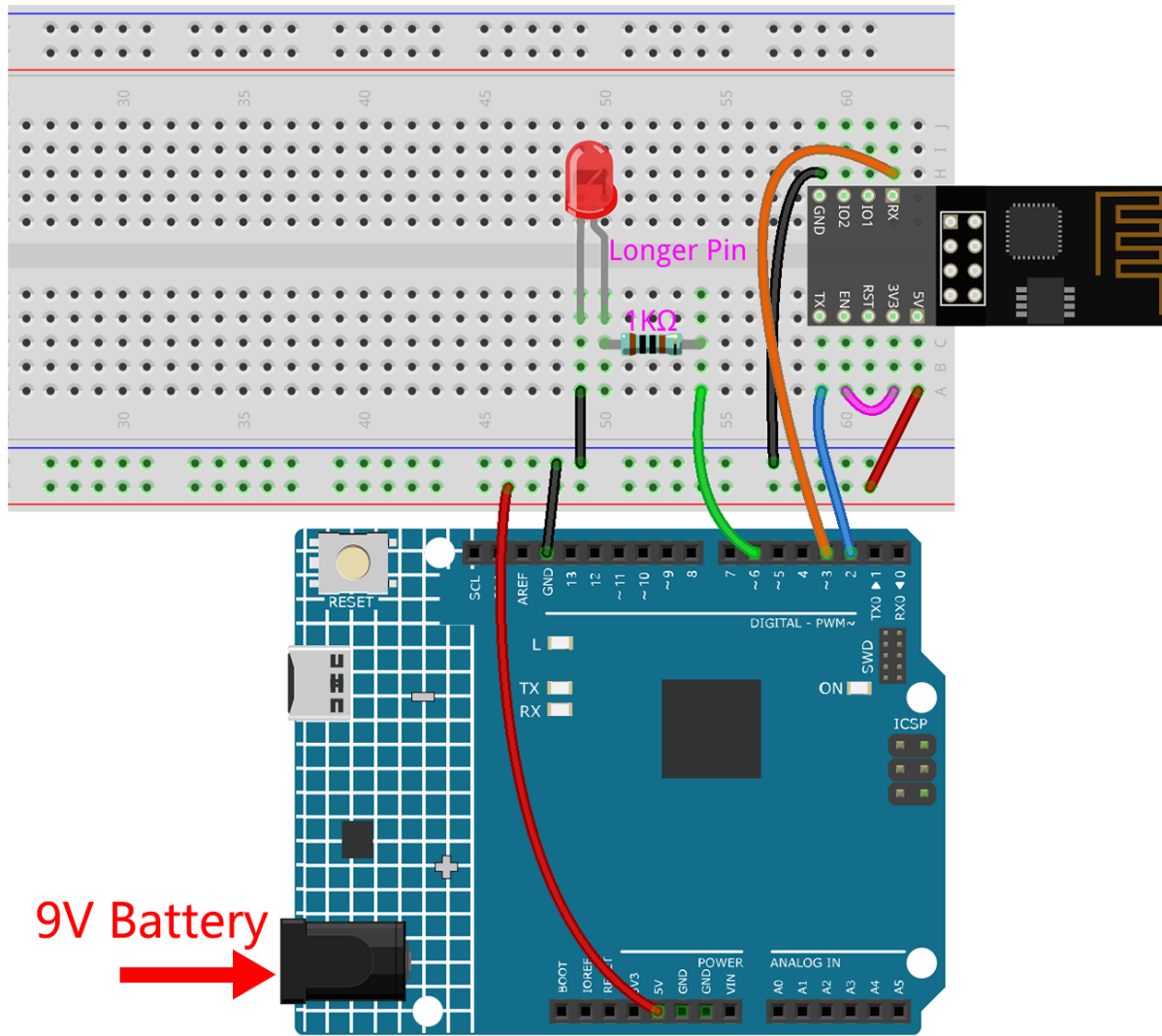
名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクからそれぞれ個別に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ESP8266 モジュール	
ジャンパーワイヤー	
抵抗器	
LED	

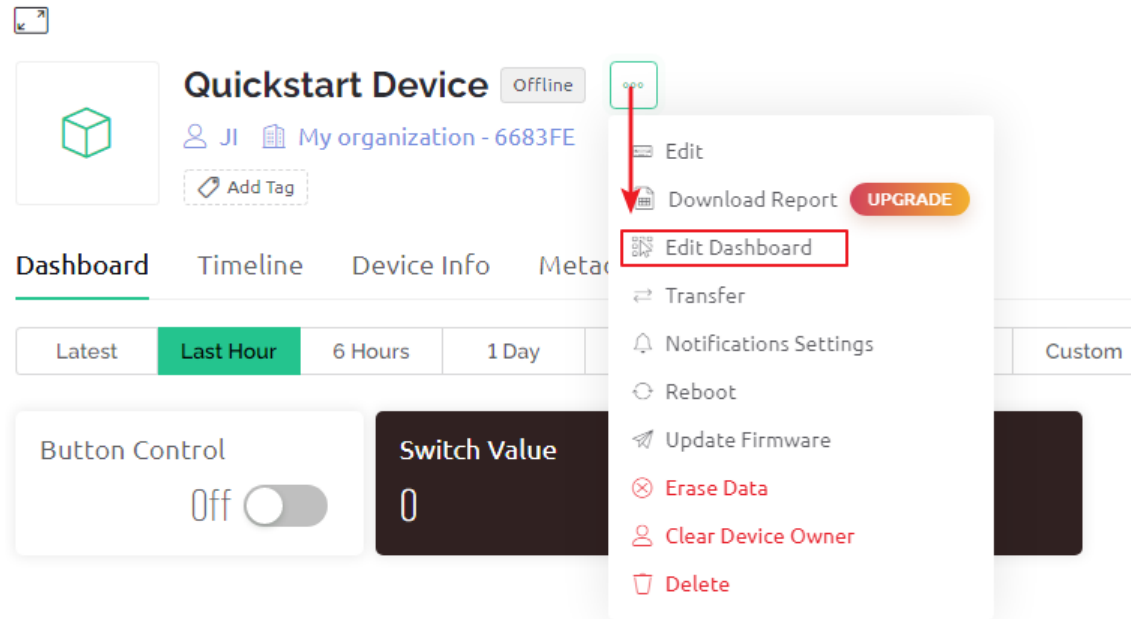
1. 回路を組む

注釈: ESP8266 モジュールは安定した動作環境を提供するために高電流が必要ですので、9V の電池が接続されていることを確認してください。

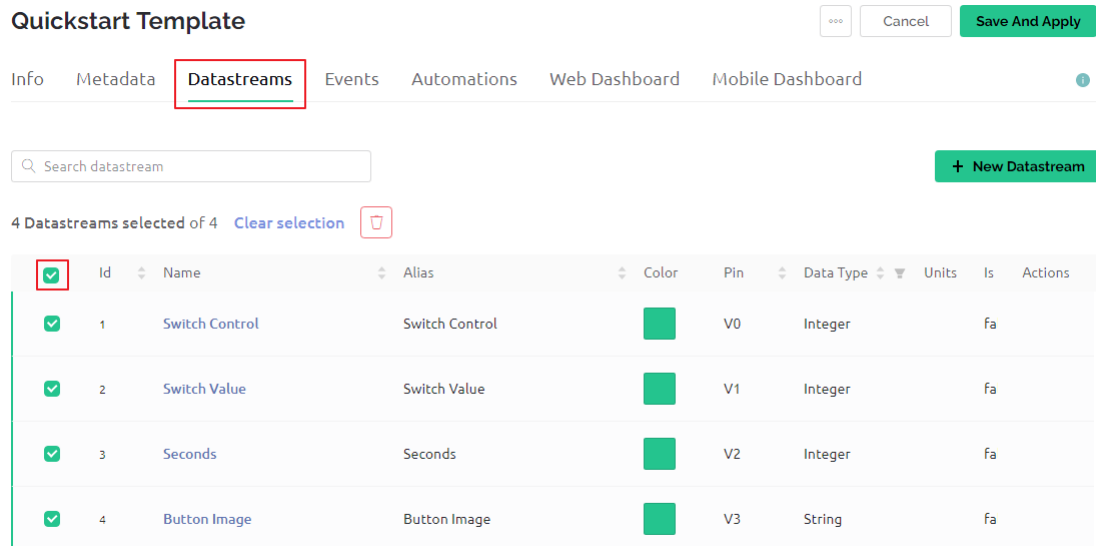


2. ダッシュボードを編集

1. 以前に作成した **Quickstart Device** に移動し、右上のメニューアイコンをクリックして **edit dashboard** を選択します。




2. Datastreams は、Blynk のウィジェットとボード上のコードがお互いを認識することを可能にします。完全な設定プロセスを体験するために、Datastreams ページからすべての Datastreams を削除します。



3. Datastreams を削除する前に、警告を慎重に読み、それが正しいことを確認してください。

DANGER ZONE

 Deleting Datastream(s) can lead to breaking changes.

- All widgets using this Datastream(s) will stop working
- All Automation scenarios using this Datastream(s) will stop working
- All active rules in Rules Engine using this Datastream(s) will stop working

This can not be undone.

Type DELETE in the field below to proceed with the deletion.

☒ I fully understand that this action is critical and will lead to data loss

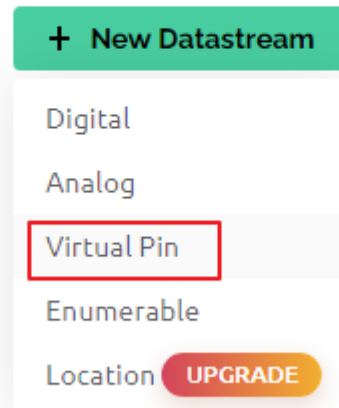
Cancel

Delete

4. Blynk のスイッチを使用して LED を制御するために使用される **Virtual Pin** タイプの Datastream を作成します。

Datastreams

Datastreams is a way to structure data that regularly flows in and out from device. Use it for sensor data, any telemetry, or actuators.



5. **Virtual Pin** を設定します。ボタンと LED は ON と OFF のみ必要なので、DATA TYPE を Integer に設定し、MIN と MAX を 0 と 1 に設定します。

Virtual Pin Datastream

NAME: Button Control

ALIAS: Button Control

PIN: V0

DATA TYPE: Integer

UNITS: None

MIN: 0 MAX: 1 DEFAULT VALUE: 0

+ ADVANCED SETTINGS

Cancel Save

Region: ny3 Privacy Policy

6. **Web Dashboard** ページに移動し、既存のウィジェットを削除します。

Quickstart Template

Info Metadata Datastreams Events Automations **Web Dashboard** Mobile Dashboard

Widget Box
3 of 30 widgets

CONTROL

Switch

Slider

Number Input

Device name
Online
Show map
UPGRADE

Device Owner
Company Name

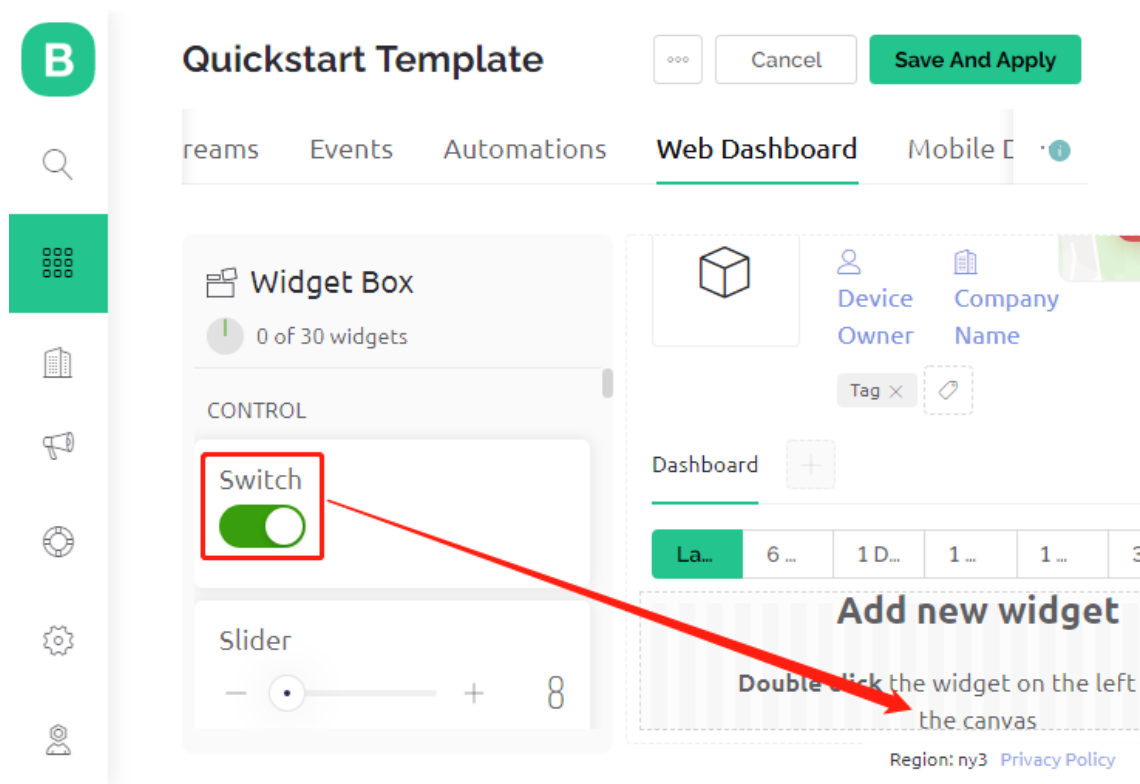
Tag

Dashboard

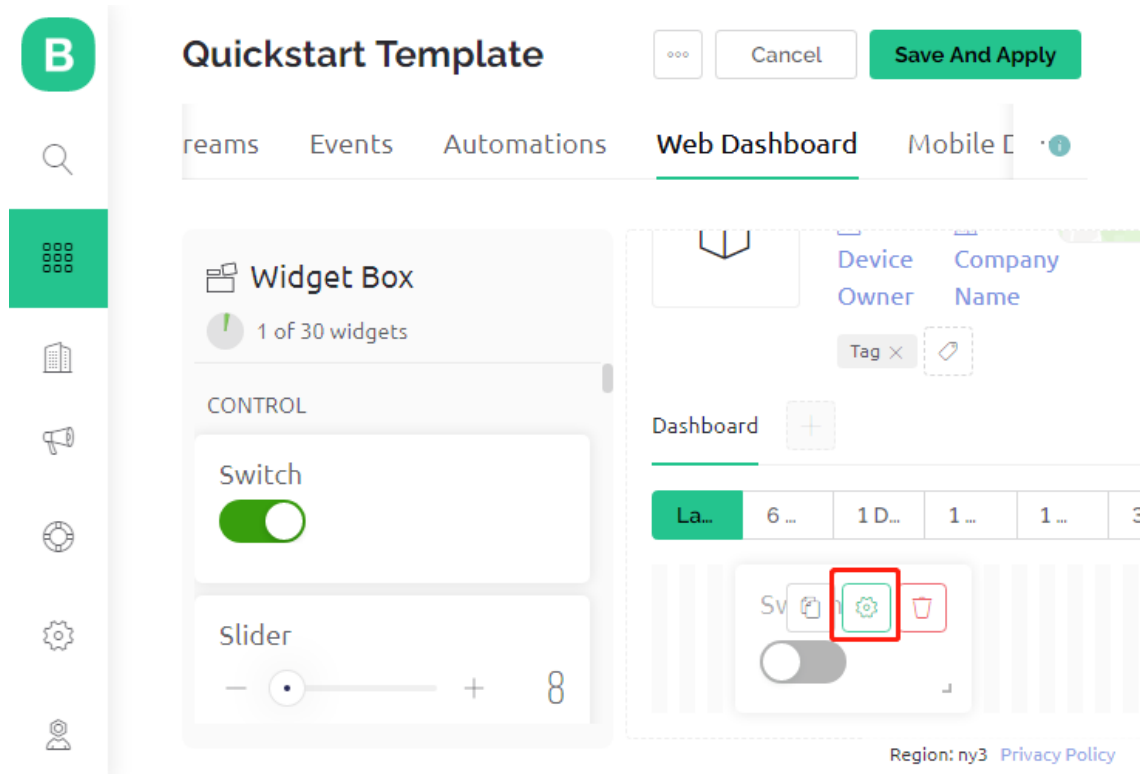
Last H... 6 Hours 1 Day 1 Week 1 Month 3 Mont... Custom

Switc... Uptime

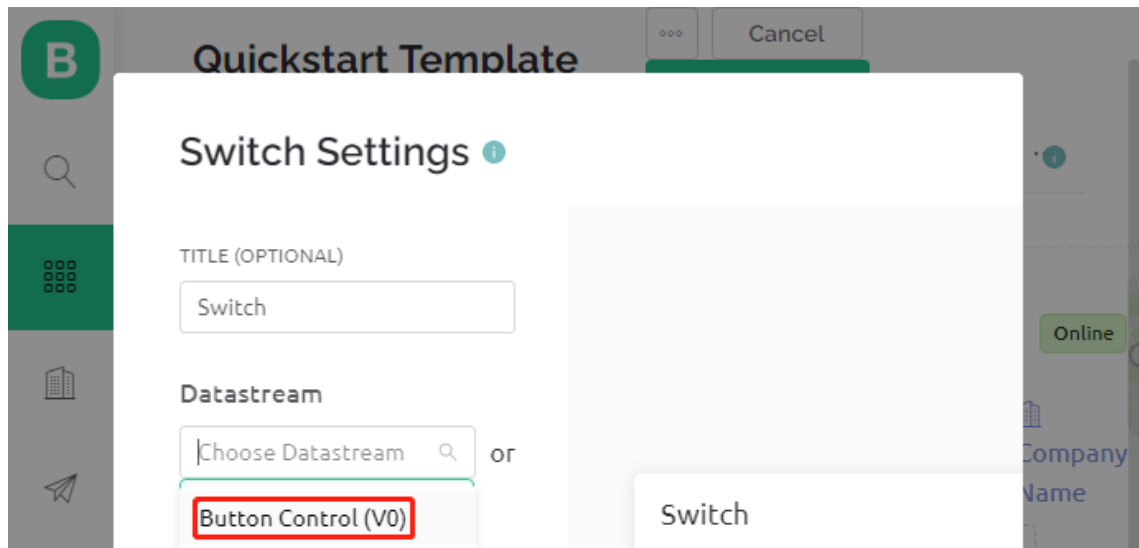
7. 左の **Widget Box** から **switch** ウィジェットをドラッグアンドドロップします。



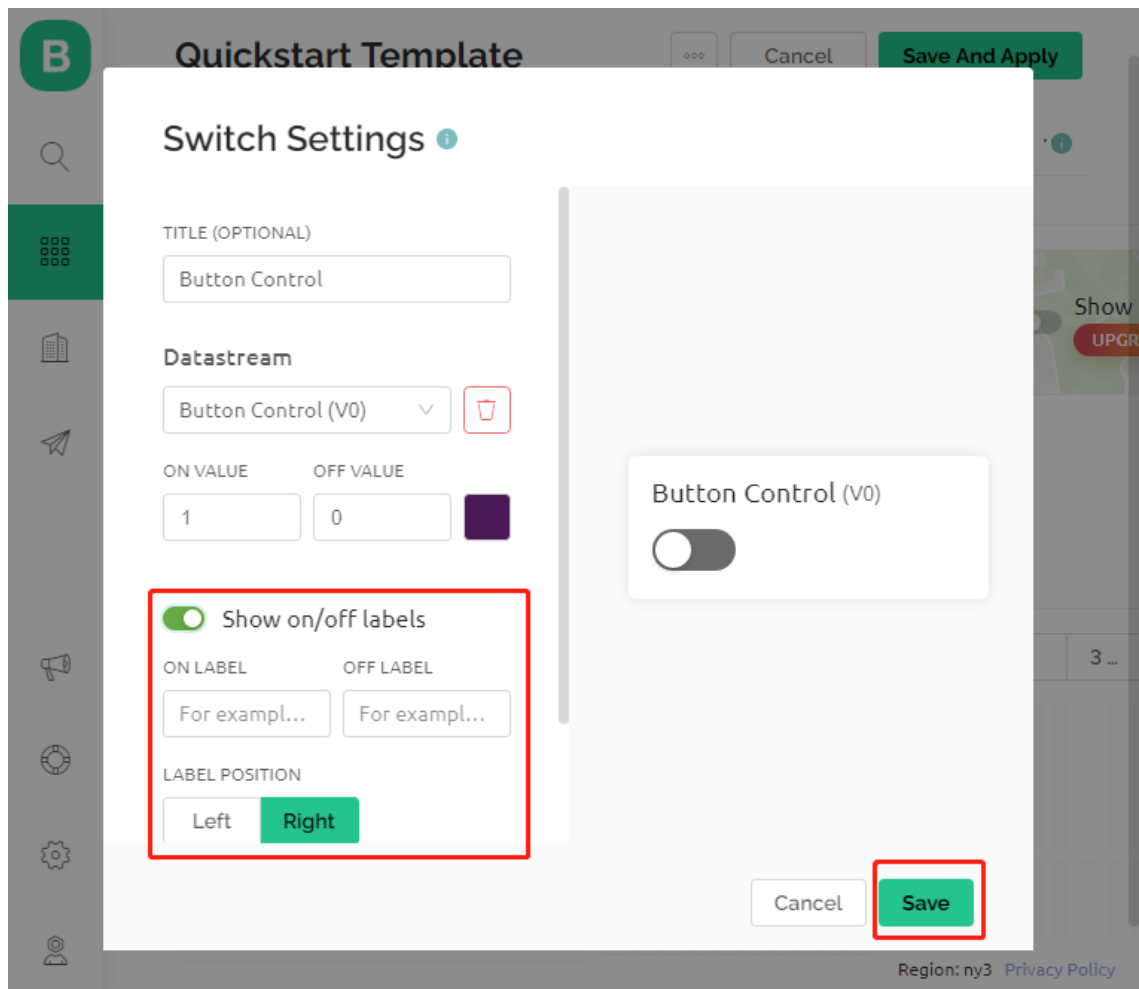
8. 次に、それを設定します。



9. **Datastream** を以前に設定したものとして選択します。



10. **Datastream** を選択すると、いくつかのカスタム設定が表示されます。それを確認したら、保存をクリックします。

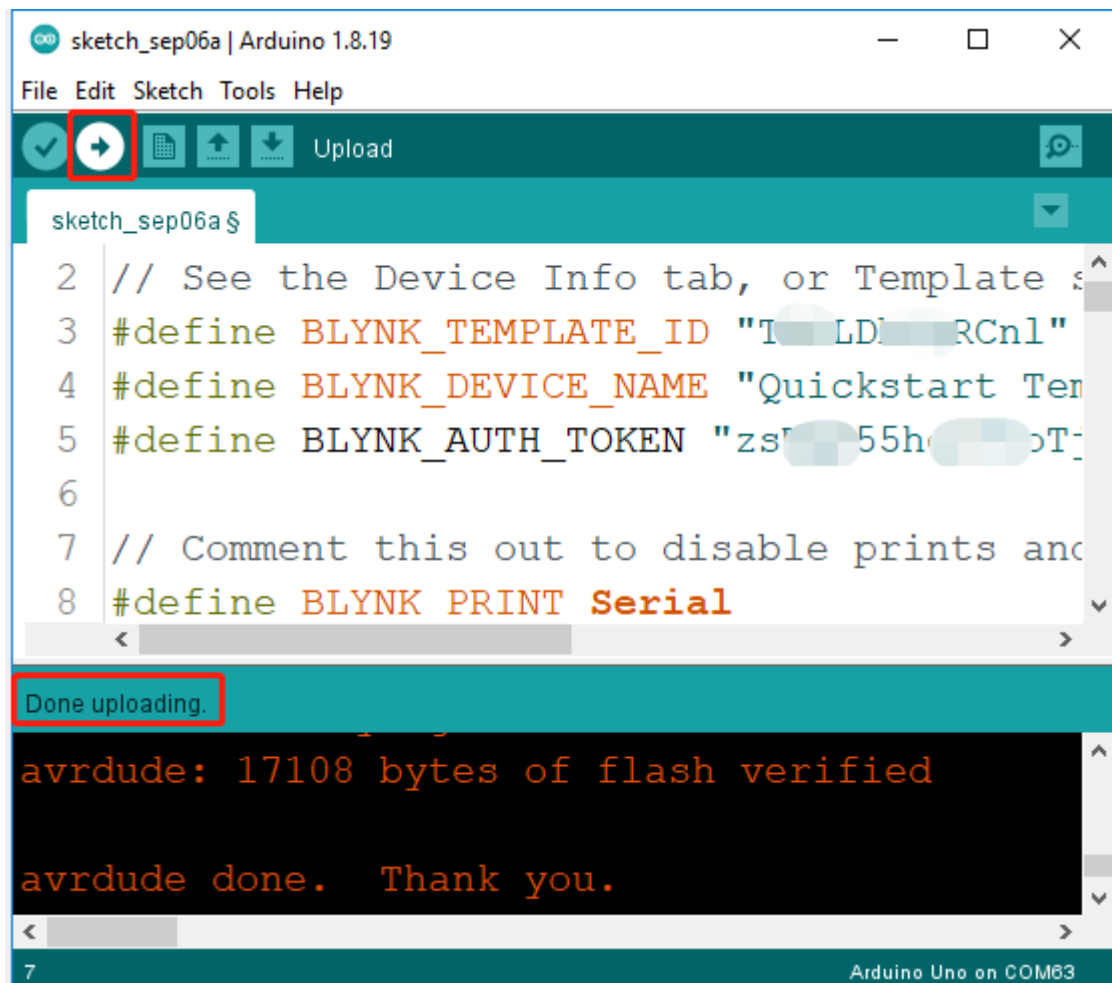


11. 最後に、**Save And Apply** をクリックします。



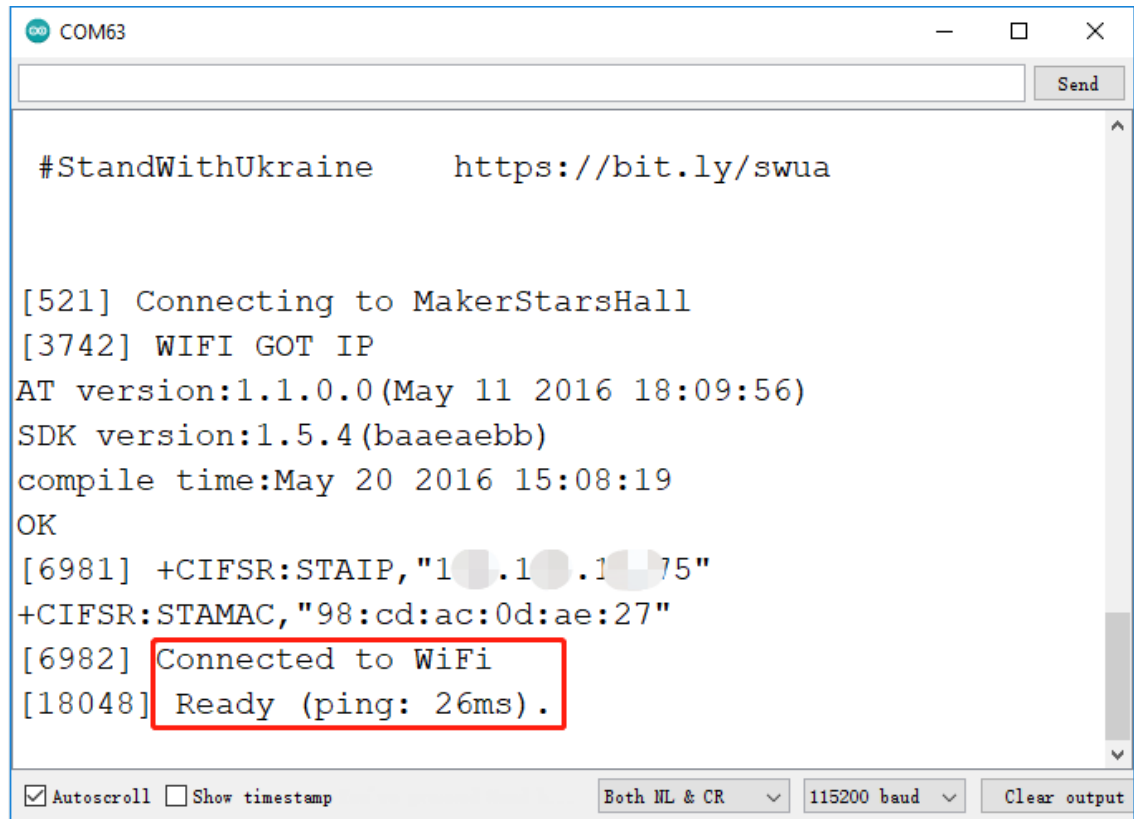
3. コードの実行

1. 3in1-kit\iot_project\2.get_data_from_blynk のパスの下での 2.get_data_from_blynk.ino ファイルを開くか、このコードを **Arduino IDE** にコピーしてください。
2. Template ID、Device Name、および Auth Token を自分のものに置き換えてください。使用している WiFi の ssid と password も入力する必要があります。詳しいチュートリアルについては、[1.4 R4 ボードを Blynk に接続する](#) を参照してください。
3. 正しいボードとポートを選択した後、**Upload** ボタンをクリックします。



4. シリアルモニタを開く（ボーレートを 115200 に設定）し、成功した接続のようなプロンプトが表示される

のを待ちます。



```
COM63

#StandWithUkraine https://bit.ly/swua

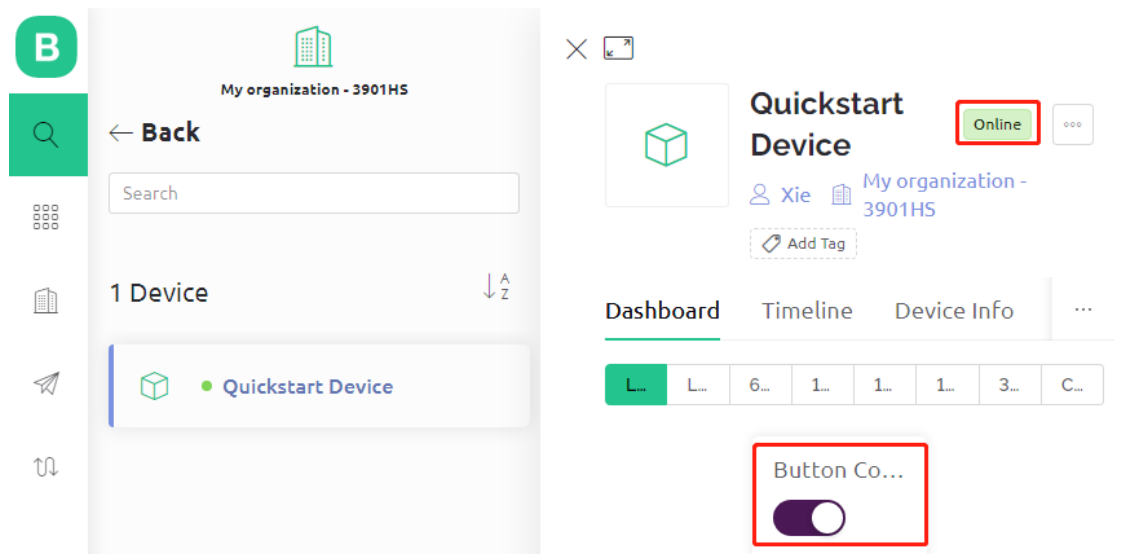
[521] Connecting to MakerStarsHall
[3742] WIFI GOT IP
AT version:1.1.0.0(May 11 2016 18:09:56)
SDK version:1.5.4(baaeaebb)
compile time:May 20 2016 15:08:19
OK
[6981] +CIFSR:STAIP,"1.1.1.175"
+CIFSR:STAMAC,"98:cd:ac:0d:ae:27"
[6982] Connected to WiFi
[18048] Ready (ping: 26ms).
```

注釈: 接続時に ESP is not responding というメッセージが表示された場合は、以下の手順に従ってください。

- 9V の電池が接続されていることを確認してください。
- RST ピンを 1 秒間 GND に接続して ESP8266 モジュールをリセットし、それを抜きます。
- ボードのリセットボタンを押します。

こうした操作を 3~5 回繰り返す必要があることもありますので、お待ちください。

5. Blynk に戻ると、ステータスがオンラインに変わり、blynk のスイッチウィジェットで R4 ボードに接続された LED を制御できるようになります。



6. Blynk をモバイルデバイスで使用したい場合は、[モバイルデバイスでの Blynk の使用方法](#) を参照してください。

どのように動作するのか？

このプロジェクトのコードと前章の *1.4 R4* ボードを Blynk に接続する のコードの違いは以下の行です。

```
const int ledPin=6;

BLYNK_WRITE(V0)
{
    int pinValue = param.asInt(); // ピン V0 からの受信値を変数に代入する
    // 以下も使用できます。
    // 文字列 i = param.asStr();
    // ダブル d = param.asDouble();
    digitalWrite(ledPin,pinValue);
}

void setup()
{
    pinMode(ledPin,OUTPUT);
}
```

ledPin の pinMode および digitalWrite に関しては、既によく知っていると思いますので、再度説明しません。注目すべきは BLYNK_WRITE(V0) 関数です。

これにより、Blynk の V0 の値が変わると、Blynk.Cloud はデバイスに「私は **Virtual Pin** V0 に書き込んでいます」と伝え、デバイスはこの情報を取得すると何かを実行できるようになります。

前の手順で V0 データストリームを作成し、それをスイッチ ウィジェットに適用しました。これは、Switch Widget を操作するたびに BLYNK_WRITE(V0) がトリガーされることを意味します。

この関数には 2 つの指示を書きました。

```
int pinValue = param.asInt();
```

V0 の値を取得し、変数 pinValue に割り当てます。

```
digitalWrite(ledPin,pinValue);
```

Blynk のスイッチウィジェットが LED を制御できるように、取得した V0 の値を ledPin に書き込みます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

6.3 3. Blynk へのデータ送信

本章では、Blynk へのデータ送信方法をご紹介します。

ここでは、ドアおよび窓の検出デバイスを作成します。リードスイッチを使用した回路は、ドアや窓の隣に配置され、磁石はドアや窓の端に取り付けられます。ドアや窓が閉じられると、リードスイッチは磁力でオンになり、R4 ボード上の対応するピンの値が変化します。Blynk.cloud はこの値を受け取るので、家を離れていても家のドアや窓が閉じているかどうかを確認できます。

現在、Blynk の LED ウィジェットを使用して、窓とドアが閉じているかどうか（すなわち、リードスイッチがオンまたはオフか）を示すことになります。

必要な部品

このプロジェクトでは、以下の部品が必要です。

キットをまとめて購入すると確実に便利です、以下がそのリンクです：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
Arduino Uno R4 Minima	-
ブレッドボード	
ESP8266 モジュール	
ジャンパーワイヤー	
抵抗器	
リードスイッチ	-

1. 回路を組む

注釈：ESP8266 モジュールは安定した動作環境を提供するために高い電流を必要とするため、9V のバッテリーが接続されていることを確認してください。



Quickstart Template

Cancel Save And Apply

Virtual Pin Datastream

NAME: Reed

ALIAS: Reed

PIN: V1

DATA TYPE: Integer

UNITS: None

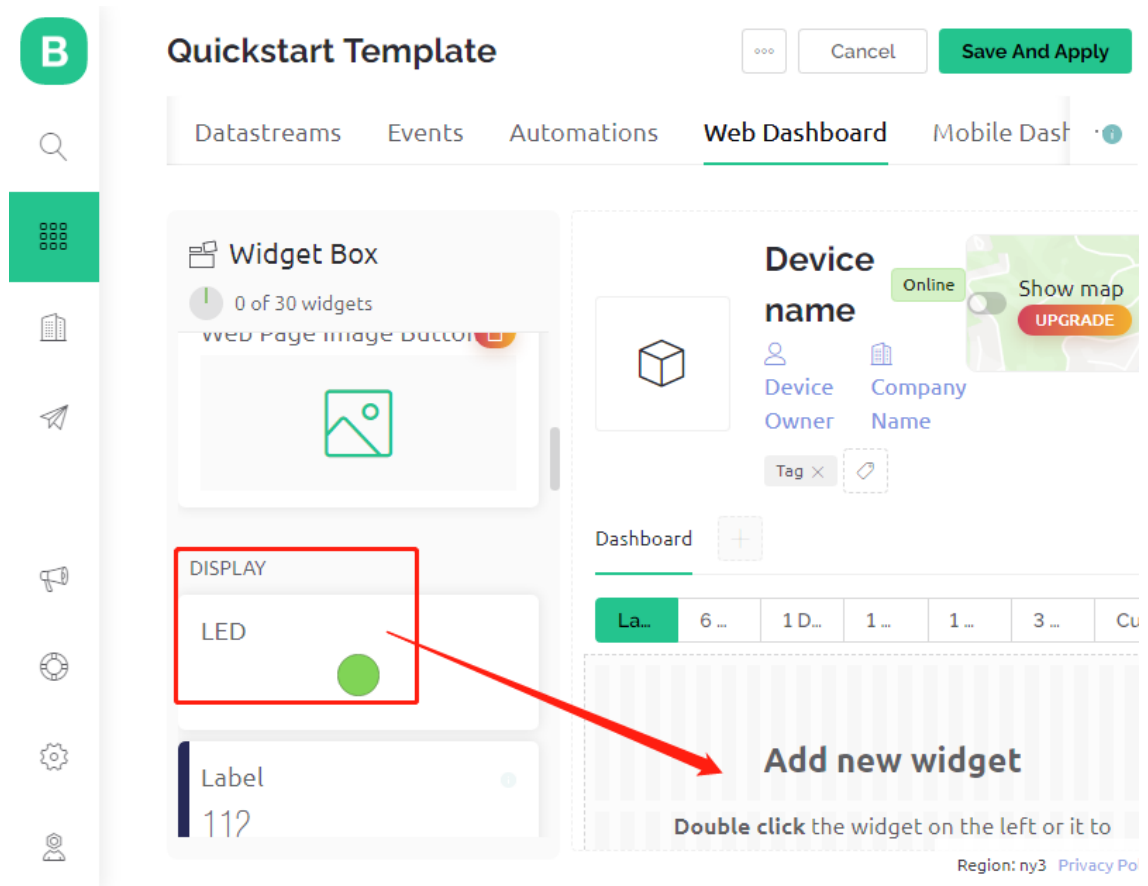
MIN: 0 MAX: 1 DEFAULT VALUE: 0

+ ADVANCED SETTINGS

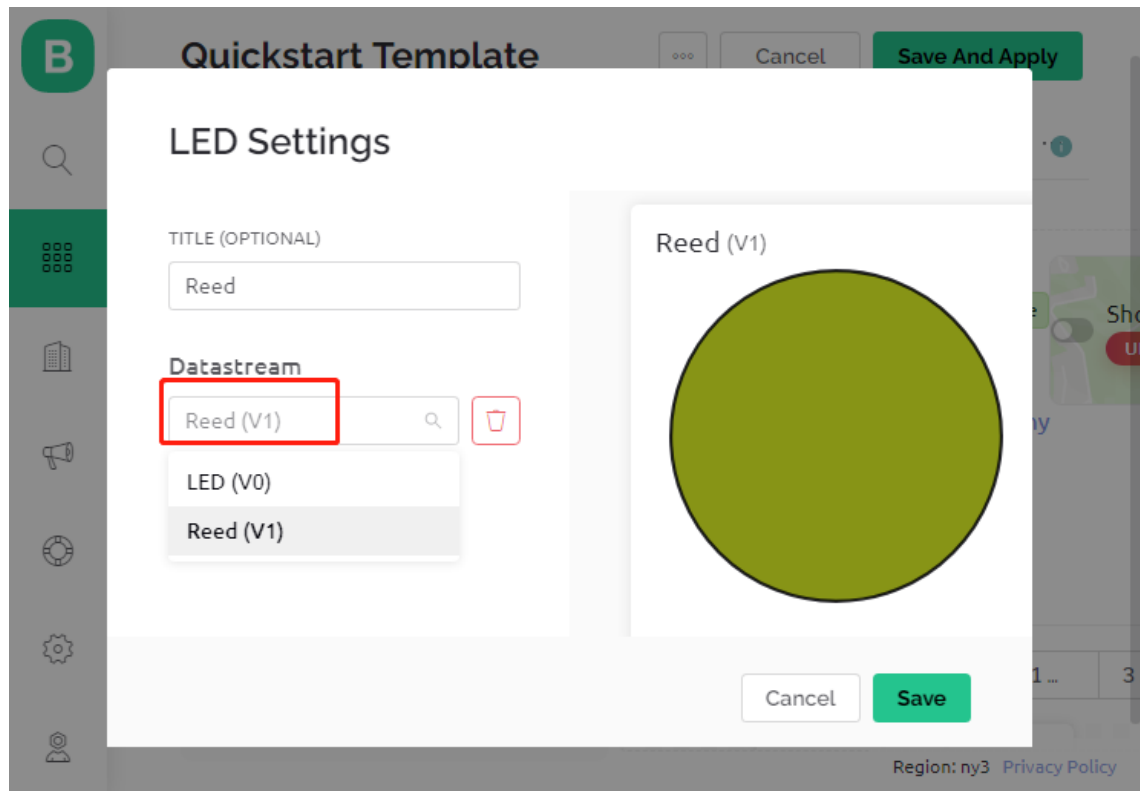
Cancel Save

Region: ny3 Privacy Policy

2. **Web Dashboard** ページで **LED widget** をドラッグアンドドロップし、値が 1 の場合、それは（色と共に）点灯し、それ以外の場合は白くなります。



3. LED widget の設定ページで、Datastream を Reed(V1) として選択し、保存します。



3. コードの実行

1. 3in1-kit\iot_project\3.push_data_to_blynk のパスの下に 3.push_data_to_blynk.ino ファイルを開く、またはこのコードを **Arduino IDE** にコピーします。
2. Template ID、Device Name、および Auth Token を自分のものに置き換えます。また、使用している WiFi の ssid および password を入力する必要があります。詳しいチュートリアルは、[1.4 R4 ボードを Blynk に接続する](#) を参照してください。
3. 正しいボードとポートを選択した後、**Upload** ボタンをクリックします。
4. シリアルモニターを開いて（ボーレートを 115200 に設定し）、接続が成功したようなプロンプトが表示されるのを待ちます。

```

COM63

#StandWithUkraine https://bit.ly/swua

[521] Connecting to MakerStarsHall
[3742] WIFI GOT IP
AT version:1.1.0.0(May 11 2016 18:09:56)
SDK version:1.5.4(baaeaebb)
compile time:May 20 2016 15:08:19
OK
[6981] +CIFSR:STAIP,"1.1.1.175"
+CIFSR:STAMAC,"98:cd:ac:0d:ae:27"
[6982] Connected to WiFi
[18048] Ready (ping: 26ms).

☒ Autoscroll ☐ Show timestamp
Both NL & CR 115200 baud Clear output

```

注釈: 接続するときに ESP is not responding というメッセージが表示される場合は、次の手順に従ってください。

- 9V のバッテリーが接続されていることを確認してください。
- ピン RST を 1 秒間 GND に接続して、ESP8266 モジュールをリセットし、その後、それを抜きます。
- R4 ボードのリセットボタンを押します。

まれに、上記の操作を 3-5 回繰り返す必要がある場合があります。しばらくお待ちください。

5. これで、Blynk はあなたのドアや窓の状態を表示します。ドアや窓が閉じている場合、LED ウィジェットは緑色になり、そうでない場合は灰色になります。
6. Blynk をモバイルデバイスで使いたい場合は、[モバイルデバイスでの Blynk の使用方法](#) を参照してください。

どのように動作するのか？

この例では、次の行に注目してください。"Blynk Cloud の V1 Datastream に每秒データを書き込む"は、これらの行で定義されています。

```

BlynkTimer timer;

void myTimerEvent()
{
    Blynk.virtualWrite(V1, pinValue);
}

void setup()
{
    timer.setInterval(1000L, myTimerEvent);
}

void loop()
{
    timer.run(); // BlynkTimer を開始
}

```

Blynk ライブラリは組み込みのタイマーを提供しています。まず、タイマーオブジェクトを作成します。

```
BlynkTimer timer;
```

setup() でタイマー間隔を設定します。ここでは、myTimerEvent() 関数を毎秒 1000ms で実行するように設定します。

```
timer.setInterval(1000L, myTimerEvent);
```

loop() で BlynkTimer を実行します。

```
timer.run();
```

カスタム関数 myTimerEvent() を編集します。コード Blynk.virtualWrite(V1, pinValue) は、V1 のデータ pinValue を書き込むために使用されます。

```

void myTimerEvent()
{
    Blynk.virtualWrite(V1, pinValue);
}

```

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ! Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

6.4 4. クラウドミュージックプレイヤー

このプロジェクトの目的は、Blynk を使用して音楽プレイヤーを作成することです。音楽は、[5.7 Tone\(\)](#) または [noTone\(\)](#) と同じ方法で再生され、曲をプログラムに書き込んでパッシブブザーで再生します。しかし、この例では、スイッチをクリックして再生/一時停止を切り替えたり、スライダーをスライドして再生の進行状況を変更することができます。

必要な部品

このプロジェクトでは、以下のコンポーネントが必要です。

キット全体を購入すると確かに便利です。以下にリンクを示します：

名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
Arduino Uno R4 Minima	-
ブレッドボード	
ESP8266 モジュール	
ジャンパーワイヤー	
ブザー	

Quickstart Template...

Virtual Pin Datastream

NAME: Slider ALIAS: Slider

PIN: V2 DATA TYPE: Integer

UNITS: None

MIN: 0 MAX: 30 DEFAULT VALUE: 0

+ ADVANCED SETTINGS

Cancel Create

Region: ny3 Privacy Policy

2. 音楽の名前を表示するための **Virtual Pin** タイプの別の **Datastream** も作成し、DATA TYPE を String に設定します。

Virtual Pin Datastream

NAME: ALIAS:

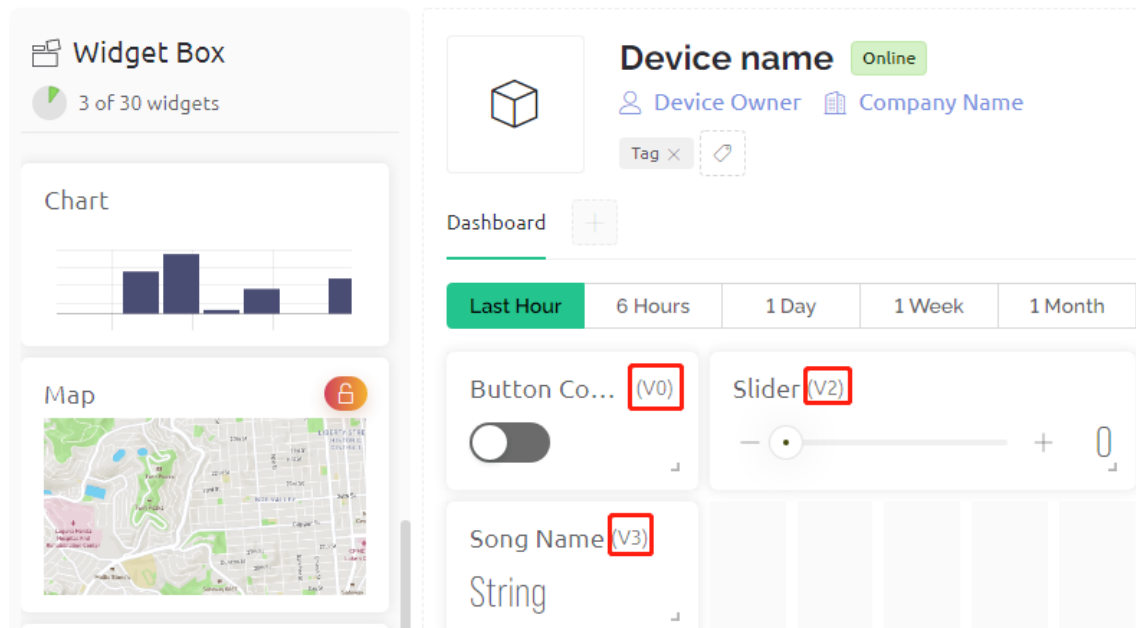
PIN: DATA TYPE:

DEFAULT VALUE:

[+ ADVANCED SETTINGS](#)

Region: ny3 [Privacy Policy](#)

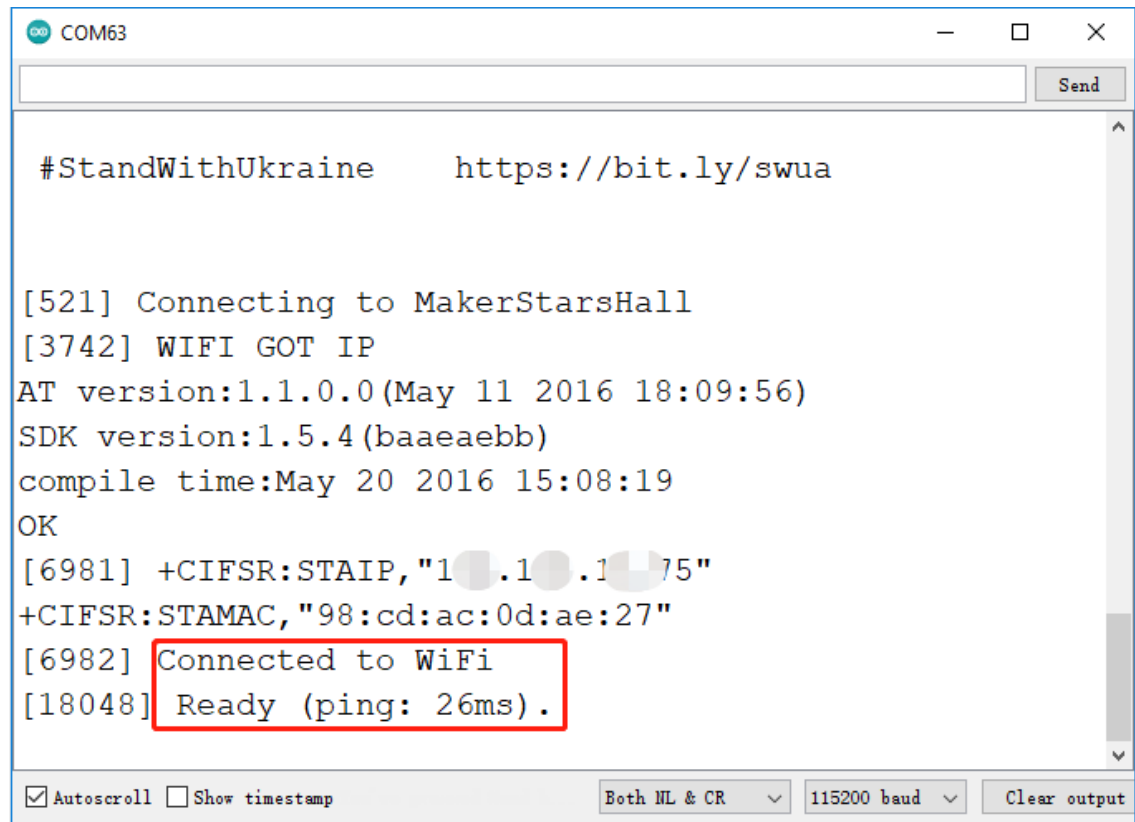
3. **Web Dashboard** ページに移動し、**Switch** ウィジェットをドラッグし、**Datastream** を V0 に設定します (V0 はすでに 2. *Blynk* からデータを取得する で設定されています); **Label** ウィジェットをドラッグして V3 に設定します; **Slider** ウィジェットをドラッグして V2 に設定します。



注釈: あなたの仮想ピンは私のものと異なる場合があります。あなたのものが優先されますが、コード内の対応するピン番号を変更する必要があります。

3. コードを実行する

1. 3in1-kit\iot_project\4.cloud_music_player のパスの下に 4.cloud_music_player.ino ファイルを開きます。
2. Template ID、Device Name、および Auth Token を自分のものに置き換えます。また、使用している WiFi の ssid および password を入力する必要があります。詳しいチュートリアルは、[1.4 R4 ボードを Blynk に接続する](#) を参照してください。
3. 正しいボードとポートを選択したら、**Upload** ボタンをクリックします。
4. シリアルモニターを開き (baudrate を 115200 に設定)、接続が成功したというプロンプトが表示されるのを待ちます。



```
COM63

#StandWithUkraine https://bit.ly/swua

[521] Connecting to MakerStarsHall
[3742] WIFI GOT IP
AT version:1.1.0.0(May 11 2016 18:09:56)
SDK version:1.5.4(baaeaebb)
compile time:May 20 2016 15:08:19
OK
[6981] +CIFSR:STAIP,"192.168.1.75"
+CIFSR:STAMAC,"98:cd:ac:0d:ae:27"
[6982] Connected to WiFi
[18048] Ready (ping: 26ms).
```

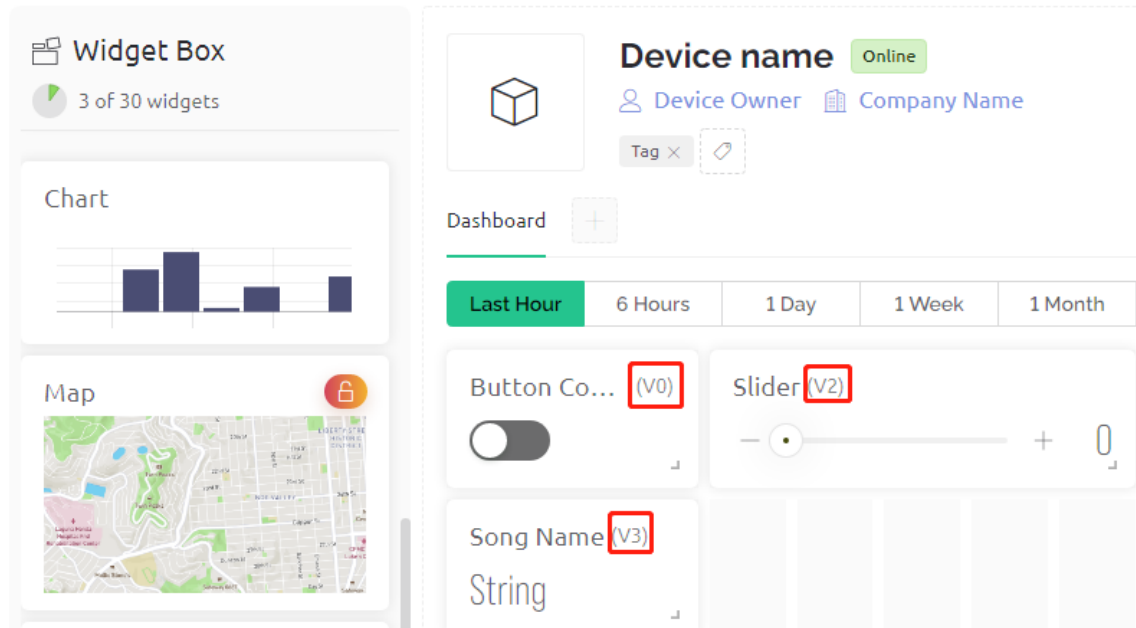
☒ Autoscroll ☐ Show timestamp Both NL & CR 115200 baud Clear output

注釈: 接続時に ESP is not responding というメッセージが表示された場合は、次の手順に従ってください。

- 9V のバッテリーが接続されていることを確認します。
- ピン RST を 1 秒間 GND に接続して ESP8266 モジュールをリセットし、その後、それを取り外します。
- R4 ボードのリセットボタンを押します。

ときどき、上記の操作を 3~5 回繰り返す必要があることがありますので、お待ちください。

5. これで、Blynk のボタン制御ウィジェットを使用して音楽の再生/一時停止を切り替えることができ、スライダーで再生の進行状況を調整することができます。また、音楽の名前も表示されます。



6. Blynk をモバイルデバイスで使用する場合は、[モバイルデバイスでの Blynk の使用方法](#) を参照してください。

どのように動作するのか？

データストリーム V0 は、Switch ウィジェットのステータスを取得し、それを変数 **musicPlayFlag** に割り当てるために使用されます。これは、音楽の一時停止と再生を制御します。

```
int musicPlayFlag=0;

BLYNK_WRITE(V0)
{
    musicPlayFlag = param.asInt(); // 音楽の開始/一時停止
}
```

データストリーム V2 は、スライダーウィジェットの値を取得し、スライダーが移動されたときにそれを変数 **scrubBar** に割り当てるために使用されます。

```
int scrubBar=0;

BLYNK_WRITE(V2)
{
    scrubBar=param.asInt();
}
```

デバイスが Blynk Cloud に接続されているとき、V3 データストリームの音楽名を書き込み、それを **Label** ウィジェットで表示します。

```

BLYNK_CONNECTED() {
    String songName = "Ode to Joy";
    Blynk.virtualWrite(V3, songName);
}

```

Blynk Timer は毎秒実行されます。 **musicPlayFlag** が 0 でない場合、つまり、 **Switch** ウィジェットが ON の場合、音楽が再生されます。2 つのノートが再生されると、プログレスバー変数 **scrubBar** が 2 増加し、その値は次に **Blynk Cloud** に書き込まれ、 **Slider** ウィジェットの値と同期されます。

```

void myTimerEvent()
{
    if(musicPlayFlag!=0)
    {
        tone(buzzerPin,melody[scrubBar],250);
        scrubBar=(scrubBar+1)%(sizeof(melody)/sizeof(int));
        delay(500);
        tone(buzzerPin,melody[scrubBar],250);
        scrubBar=(scrubBar+1)%(sizeof(melody)/sizeof(int));
        Serial.println(scrubBar);
        Blynk.virtualWrite(V2, scrubBar);
    }
}

```

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

6.5 5. 家の環境モニタリング

この章では、Blynk を使用して家の環境モニタを作成します。DHT11 とフォトレジスタを使用することで、部屋の温度、湿度、および光の強度を測定できます。これらのデータを Blynk に送信することで、インターネットを通じてあなたの家の環境を把握することができます。

必要な部品

このプロジェクトには、以下の部品が必要です。

一式として購入するのがおすすめです。以下はそのリンクです：

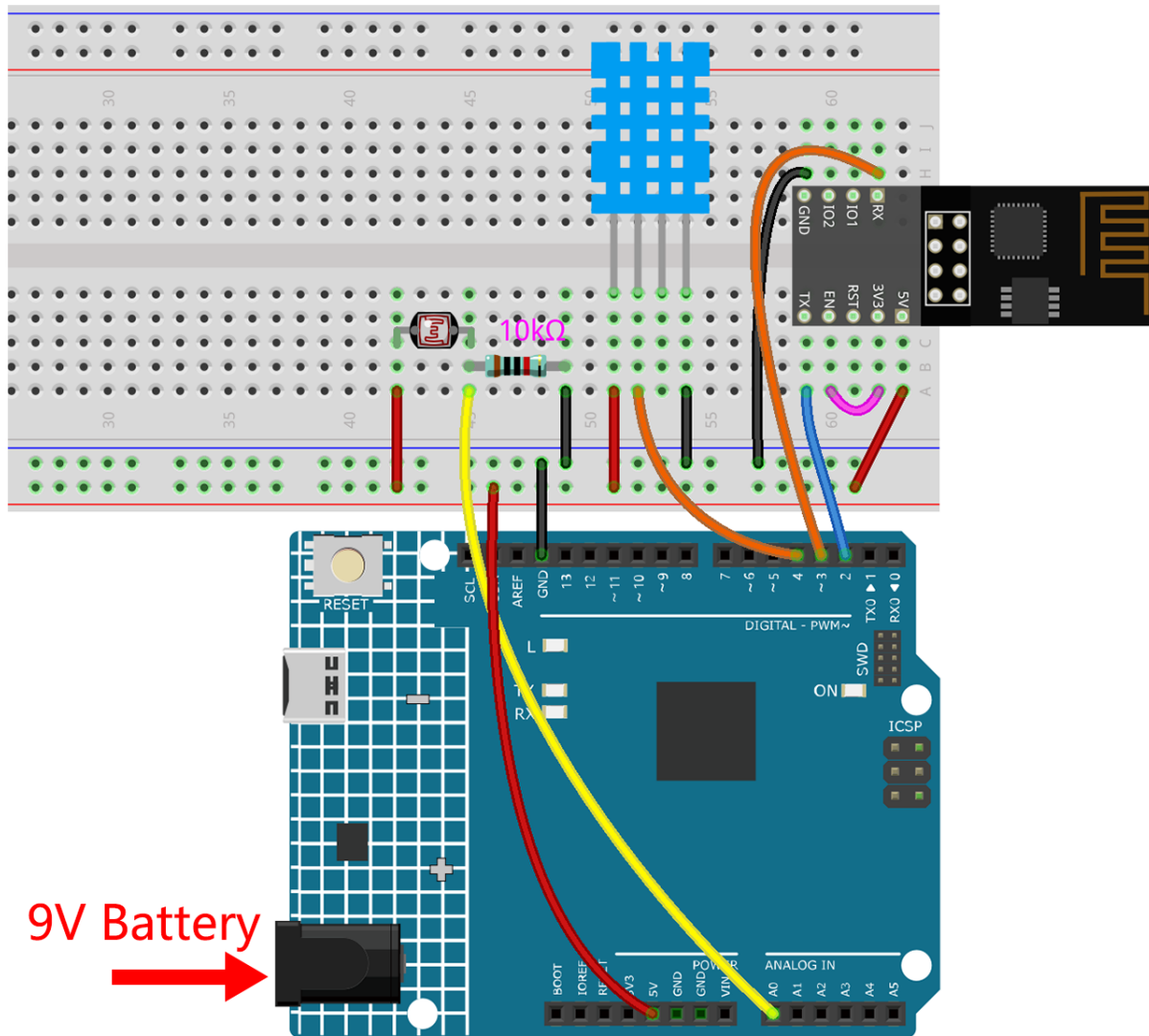
名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

また、以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
<i>ESP8266</i> モジュール	
ジャンパーワイヤー	
抵抗器	
フォトレジスタ	
<i>DHT11</i> 温湿度センサ	-

1. 回路を組む

注釈：ESP8266 モジュールは安定した動作のために高い電流が必要なので、9V のバッテリーが接続されていることを確認してください。



2. ダッシュボードを編集

1. 湿度のデータを記録するため、**Datastream** ページで **Virtual Pin** タイプの **Datastream** を作成します。
DATA TYPE を **Double** に設定し、MIN と MAX を **0** と **100** にします。また、単位は **Percentage, %** に設定します。

Quickstart Template

Virtual Pin Datastream

NAME: Humidity ALIAS: Humidity

PIN: V4 DATA TYPE: Double

UNITS: Percentage, %

MIN: 0 MAX: 100 DECIMALS: ### DEFAULT VALUE: Default Value

+ ADVANCED SETTINGS

Cancel Save

Region: ny3 Privacy Policy

2. 温度のデータを記録するための Virtual Pin タイプの Datastream を作成します。DATA TYPE を Double に、MIN と MAX を -30 と 50 に設定し、単位を Celsius, ° C にします。

Virtual Pin Datastream

NAME: ALIAS:

PIN: DATA TYPE:

UNITS:

MIN: MAX: DECIMALS: DEFAULT VALUE:

ADVANCED SETTINGS

Region: ny3 [Privacy Policy](#)

3. 光の強度のデータを記録するための **Virtual Pin** タイプの **Datastream** を作成します。デフォルトのデータタイプ、すなわち **Integer** を使用し、MIN と MAX を 0 と 1024 に設定します。

Quickstart Templa...

Virtual Pin Datastream

NAME: Illumination ALIAS: Illumination

PIN: V6 DATA TYPE: Integer

UNITS: None

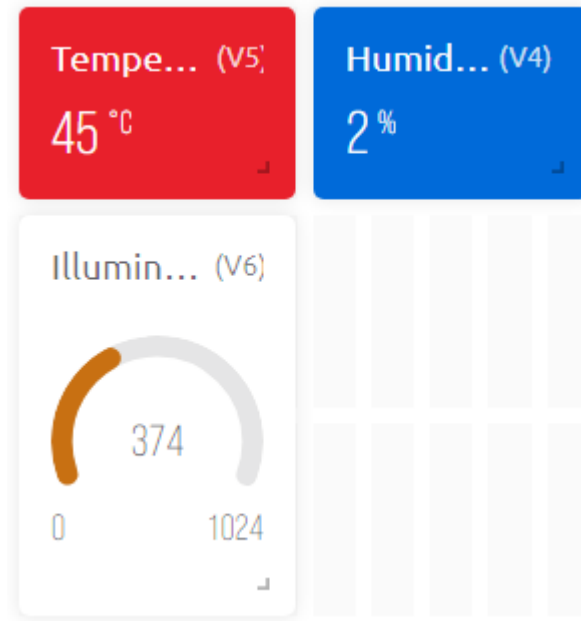
MIN: 0 MAX: 1024 DEFAULT VALUE: 0

+ ADVANCED SETTINGS

Cancel Create

Region: ny3 Privacy Policy

4. **Wed Dashboard** ページに移動し、V4 および V5 にデータストリームを設定した 2 つの **Label** ウィジェットを配置し、V6 にデータストリームを設定した **Gauge** ウィジェットを配置します。ウィジェットの設定で、値に応じて色を変更する オプションを有効にし、見やすく直感的な表示のために適切な色を選択できます。

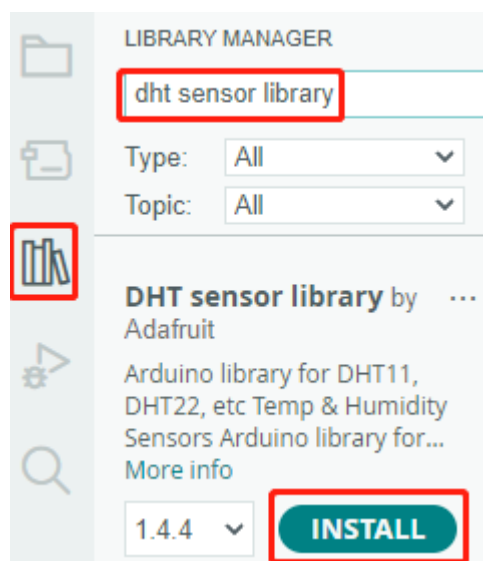


3. コードの実行

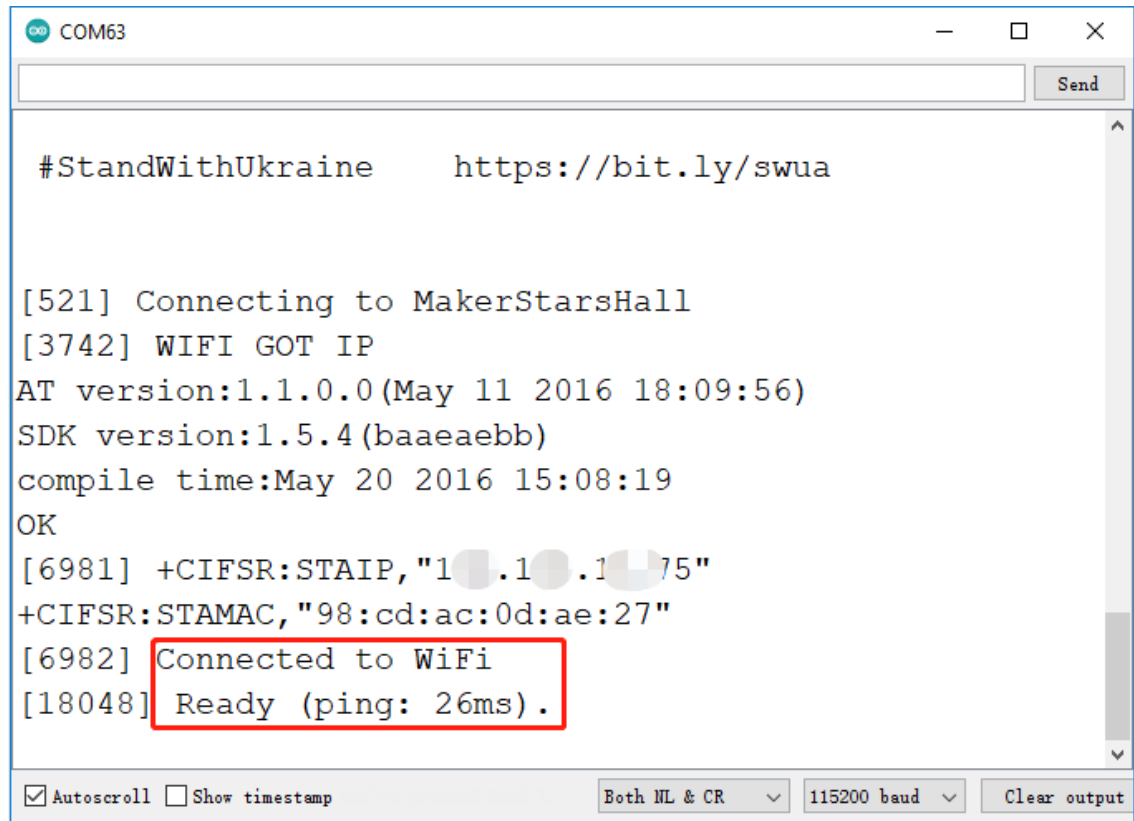
1. 3in1-kit\iot_project\5.home_environment_monitoring のパスの下で 5. home_environment_monitoring.ino ファイルを開く、またはこのコードを **Arduino IDE** にコピーします。

注釈:

- ここでは DHT sensor library が使用されています。 **Library Manager** からインストールできます。



2. Template ID、Device Name、および Auth Token を自分のものに置き換えます。使用している WiFi の ssid と password も入力する必要があります。詳細なチュートリアルは、[1.4 R4 ボードを Blynk に接続する](#) を参照してください。
3. 正しいボードとポートを選択した後、**Upload** ボタンをクリックします。
4. シリアルモニターを開いて（ボーレートを 115200 に設定）成功した接続などのプロンプトが表示されるのを待ちます。



```
COM63

#StandWithUkraine      https://bit.ly/swua

[521] Connecting to MakerStarsHall
[3742] WIFI GOT IP
AT version:1.1.0.0(May 11 2016 18:09:56)
SDK version:1.5.4(baaeaebb)
compile time:May 20 2016 15:08:19
OK
[6981] +CIFSR:STAIP,"192.168.1.75"
+CIFSR:STAMAC,"98:cd:ac:0d:ae:27"
[6982] Connected to WiFi
[18048] Ready (ping: 26ms).
```

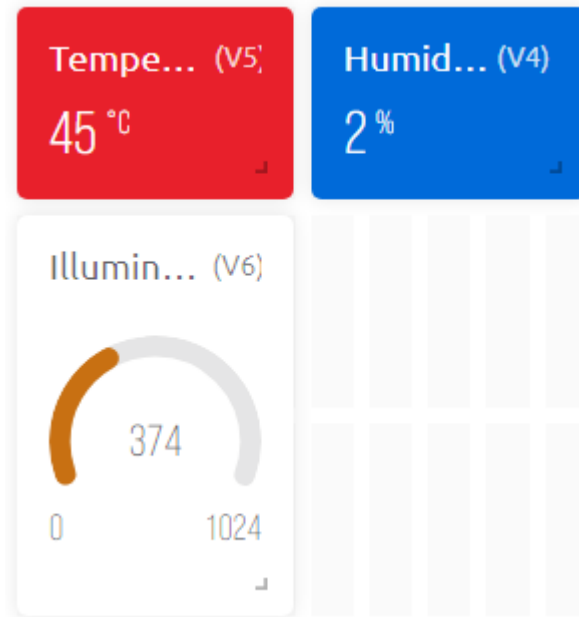
☒ Autoscroll ☐ Show timestamp Both NL & CR 115200 baud Clear output

注釈: 接続時に ESP is not responding というメッセージが表示された場合は、以下の手順を実行してください。

- 9V のバッテリーが接続されていることを確認します。
- RST ピンを 1 秒間 GND に接続して ESP8266 モジュールをリセットし、その後、抜きます。
- R4 ボードのリセットボタンを押します。

こうした操作を 3~5 回繰り返す必要があることもありますので、お待ちください。

5. これで、Blynk 上で現在の周囲の温度、湿度、光の強度を見ることができます。



6. Blynk をモバイルデバイスで使用したい場合は、[モバイルデバイスでの Blynk の使用方法](#) を参照してください。



どのように動作するのか？

これらの二つの関数は、部屋の温度、湿度、光の強度を取得するために使用されます。

```
int readLight(){
    return analogRead(lightPin);
}
```

(次のページに続く)

```
bool readDHT() {  
  
    // 温度または湿度の読み取りには約 250 ミリ秒かかります！  
    // センサーの読み取りは、最大 2 秒「古い」場合があります（非常に遅いセンサーです）  
    humidity = dht.readHumidity();  
    // 温度を摂氏で読み取る（デフォルト）  
    temperature = dht.readTemperature();  
  
    // いずれかの読み取りに失敗した場合は確認し、早期に終了します（再試行のため）  
    if (isnan(humidity) || isnan(temperature)) {  
        Serial.println("Failed to read from DHT sensor!");  
        return false;  
    }  
    return true;  
}
```

Blynk の Timer を使用して、周囲の温度、湿度、光の強度は毎秒取得され、Blynk Cloud 上のデータストリームに送信されます。そして、ウィジェットがデータを表示します。

```
void myTimerEvent()  
{  
    bool chk = readDHT();  
    int light = readLight();  
    if(chk){  
        Blynk.virtualWrite(V4,humidity);  
        Blynk.virtualWrite(V5,temperature);  
    }  
    Blynk.virtualWrite(V6,light);  
}
```

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。

- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

6.6 6. 植物モニター

このプロジェクトの目的は、現在の温度、湿度、光の強度、土壌の湿度を検出し、それらを Blynk に表示するスマートな水やりシステムを作成することです。

Blynk Cloud でスイッチをオンにすると、ポンプが動作を開始し、植物に水分が供給されます。

必要な部品

このプロジェクトでは、以下の部品が必要です。

全体のキットを購入するのは非常に便利です。リンクは以下のとおりです：

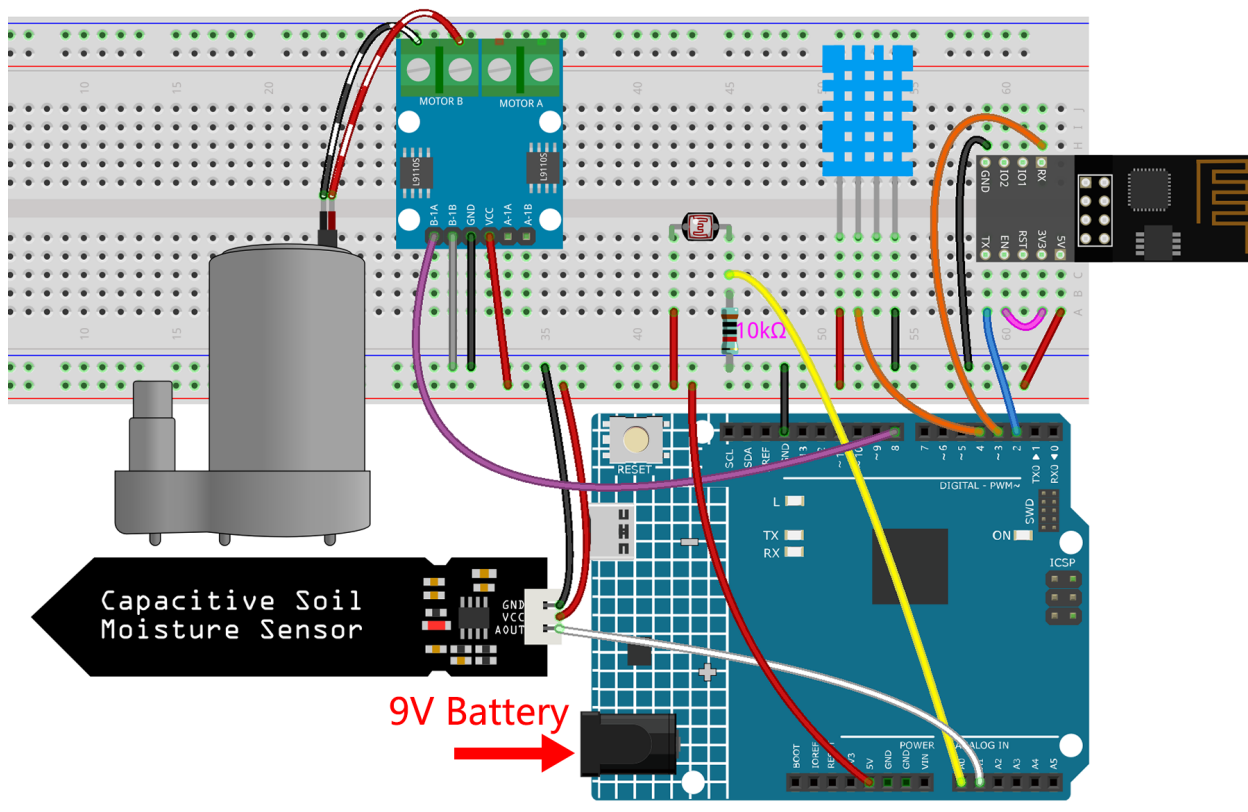
名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ESP8266 モジュール	
ジャンパーワイヤー	
抵抗器	
フォトレジスタ	
<i>DHT11</i> 温湿度センサ	-
土壌湿度モジュール	
<i>L9110</i> モータードライバモジュール	-
遠心ポンプ	-

1. 回路を組む

注釈: ESP8266 モジュールは、安定した動作環境を提供するために高電流が必要ですので、9V のバッテリーが接続されていることを確認してください。



2. ダッシュボードを編集

1. 前のプロジェクトで作成されたデータストリームは保存される必要があり、このプロジェクトでも使用されます。
2. 土壌の湿度を記録するために、**Datastream** ページで **Virtual Pin** タイプの別の **Datastream** を作成します。
DATA TYPE を Integer に設定し、MIN と MAX をそれぞれ 0 と 1024 に設定します。

Quickstart Templa...

Virtual Pin Datastream

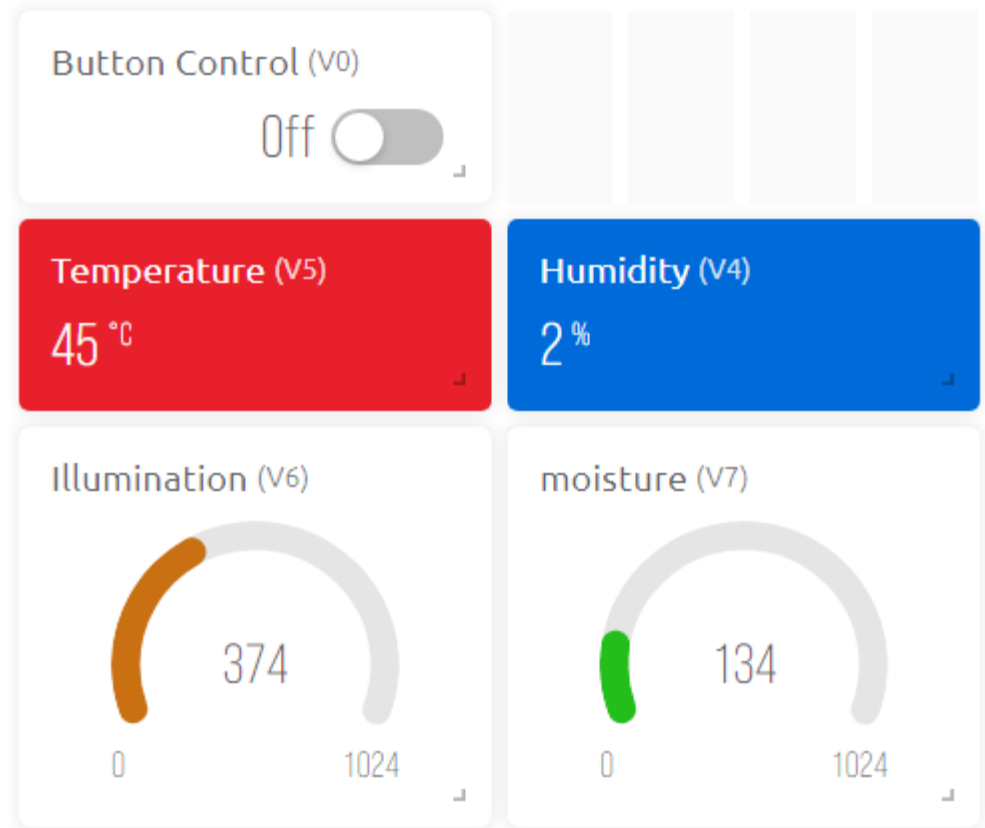
NAME	Moisture	ALIAS	Moisture
PIN	V7	DATA TYPE	Integer
UNITS	None		
MIN	0	MAX	1024
		DEFAULT VALUE	0

+ ADVANCED SETTINGS

Cancel Create

Region: ny3 Privacy Policy

3. 次に **Wed Dashboard** ページに移動し、2つの **Label** ウィジェットをドラッグし、それらのデータストリームをそれぞれ **V4** と **V5** に設定します。2つの **Gauge** ウィジェットをドラッグし、そのデータストリームをそれぞれ **V6** と **V7** に設定します。最後に、**Switch** ウィジェットをドラッグし、そのデータストリームを **V0** に設定します。

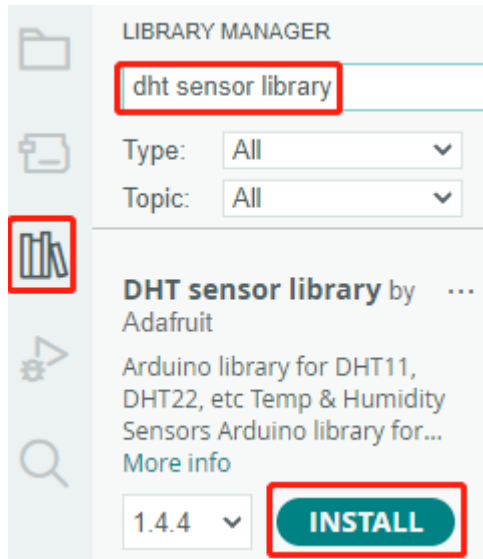


3. コードの実行

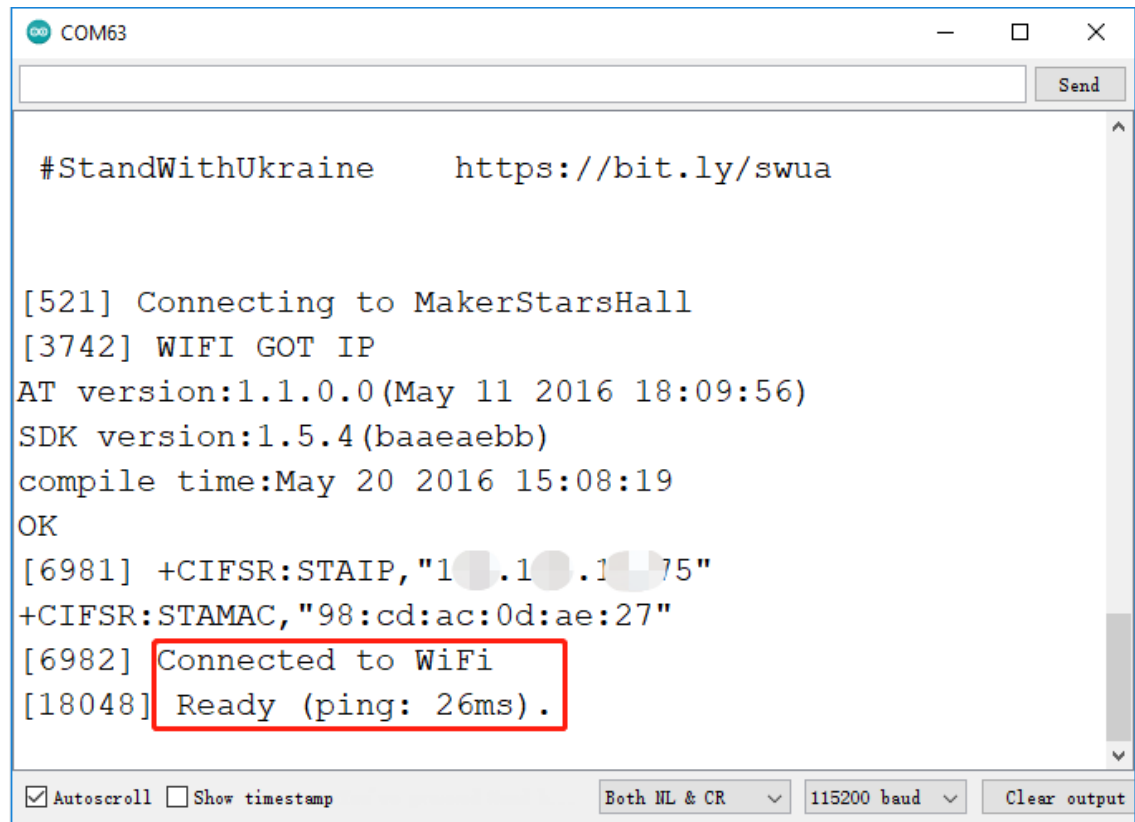
3in1-kit\iot_project\6.plant_monitoring のパスの下で 6.plant_monitoring.ino ファイルを開く、もしくはこのコードを **Arduino IDE** にコピーします。

注釈:

- ここでは DHT sensor library が使用されています。 **Library Manager** からインストールすることができます。



1. Template ID、Device Name、および Auth Token を自分のものに置き換えます。使用している WiFi の ssid および password も入力する必要があります。詳しいチュートリアルは [1.4 R4 ボードを Blynk に接続する](#) を参照してください。
2. 適切なボードとポートを選択した後、**Upload** ボタンをクリックします。
3. シリアルモニター（ボーレートを 115200 に設定）を開き、成功した接続などのプロンプトが表示されるのを待ちます。



```
COM63

#StandWithUkraine    https://bit.ly/swua

[521] Connecting to MakerStarsHall
[3742] WIFI GOT IP
AT version:1.1.0.0(May 11 2016 18:09:56)
SDK version:1.5.4(baaeaebb)
compile time:May 20 2016 15:08:19
OK
[6981] +CIFSR:STAIP,"192.168.1.75"
+CIFSR:STAMAC,"98:cd:ac:0d:ae:27"
[6982] Connected to WiFi
[18048] Ready (ping: 26ms).
```

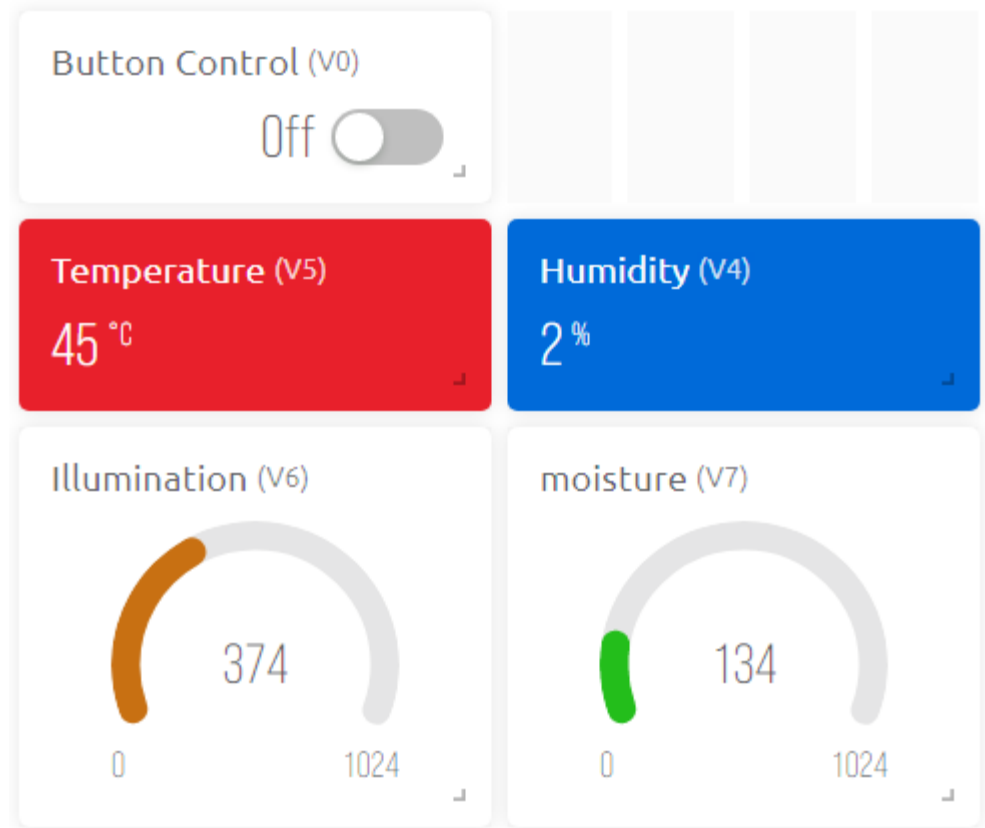
☒ Autoscroll ☐ Show timestamp Both NL & CR 115200 baud Clear output

注釈: 接続時に ESP is not responding というメッセージが表示される場合、以下の手順に従ってください。

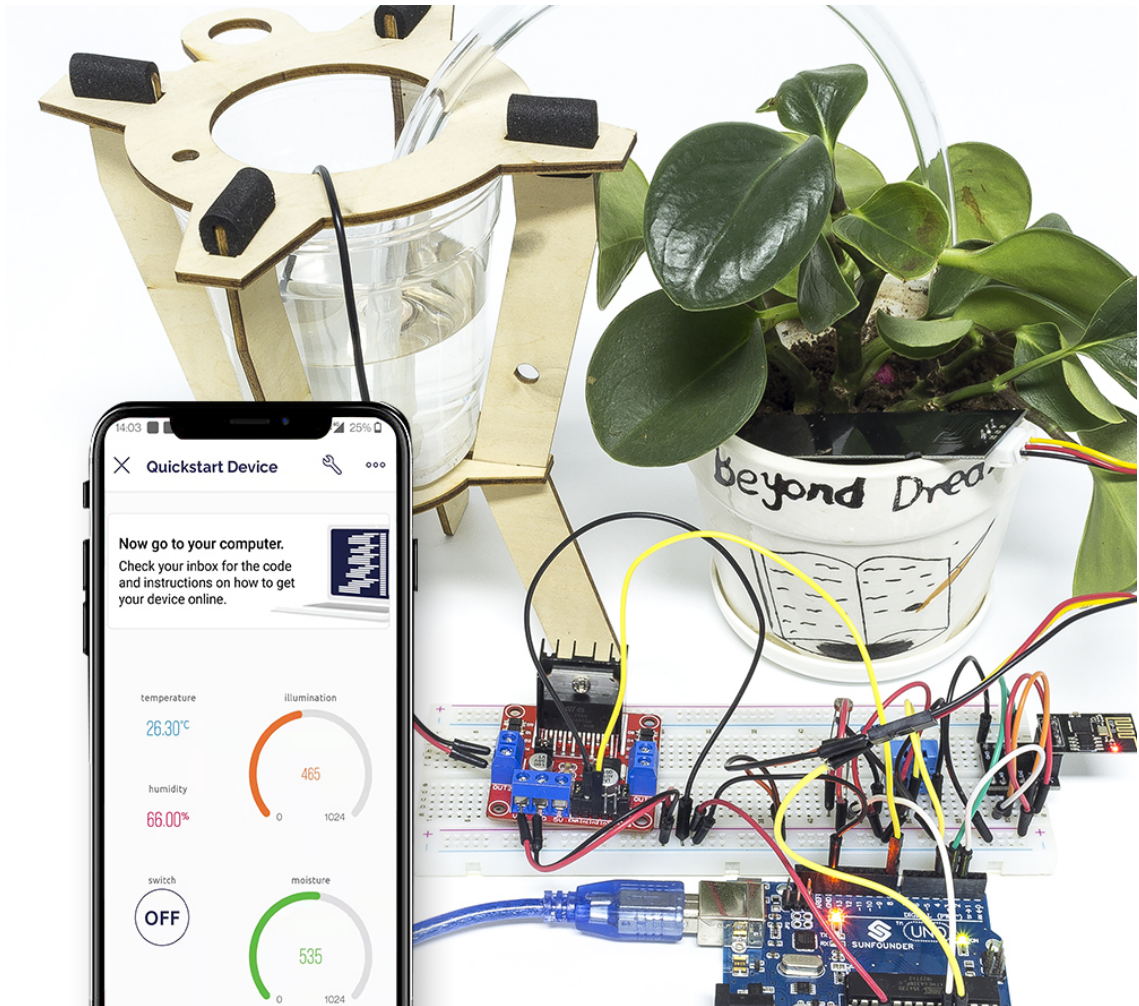
- 9V のバッテリーが接続されていることを確認します。
- ピン RST を 1 秒間 GND に接続して、ESP8266 モジュールをリセットします。その後、それを取り外します。
- R4 ボード上のリセットボタンを押します。

この操作を 3~5 回繰り返す必要があることもあります。しばらくお待ちください。

4. Blynk に戻ると、現在の温度、湿度、光の強度、土壌の湿度が表示されます。必要に応じて、ボタンコントロールウィジェットをクリックして植物に水をやることができます。



5. モバイルデバイスで Blynk を使用したい場合は、[モバイルデバイスでの Blynk の使用方法](#) を参照してください。



どのように動作するのか？

この BLYNK_WRITE は、Blynk の **Switch** ウィジェットが ON のときにポンプを起動し、OFF のときに停止するようにします。

```
BLYNK_WRITE(V0)
{
  if(param.asInt()==1){
    digitalWrite(pumpA,HIGH);
  }else{
    digitalWrite(pumpA,LOW);
  }
}
```

これらの三つの関数は、現在の環境温度、湿度、光の強度、および土壌の湿度を取得するために使用されます。

```
int readMoisture(){
    return analogRead(moisturePin);
}

int readLight(){
    return analogRead(lightPin);
}

bool readDHT() {

    // 温度または湿度の読み取りには約 250 ミリ秒かかります。
    // センサーの読み取り値は最大 2 秒ほど「古い」場合もあります（非常に遅いセンサーです）
    humidity = dht.readHumidity();
    // 温度を摂氏として読み取ります（デフォルト）
    temperature = dht.readTemperature();

    // 失敗した読み取りがあるかどうかを確認し、早めに終了します（再試行するため）。
    if (isnan(humidity) || isnan(temperature)) {
        Serial.println("Failed to read from DHT sensor!");
        return false;
    }
    return true;
}
```

Blynk の Timer を使用すると、周囲の温度、湿度、光の強度、土壌の湿度が毎秒取得され、それらが Blynk Cloud 上のデータストリームに送信されます。その結果、ウィジェットがデータを表示します。

```
void myTimerEvent()
{
    bool chk = readDHT();
    int light = readLight();
    int moisture = readMoisture();
    if(chk){
        Blynk.virtualWrite(V4,humidity);
        Blynk.virtualWrite(V5,temperature);
    }
    Blynk.virtualWrite(V6,light);
    Blynk.virtualWrite(V7,moisture);
}
```

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なレビュー：新製品の発表や先行レビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

6.7 7. 数量制限ゲート

駐車場などの状況では、数量の管理が必要とされます。

ここでは、スマートゲートを作成します。サーボはゲートとして使用され、その前に IR 障害物検知器が配置されます。オブジェクト（車など）が検出されると、ゲートが開き、数字が 1 増加します。カウントは 7 セグメントディスプレイで表示され、Blynk Cloud にもアップロードされ、遠隔地での閲覧が可能です。最後に、Blynk にはこのスマートゲートシステムを有効/無効にする Switch ウィジェットがあります。

必要な部品

このプロジェクトでは、以下の部品が必要です。

全体のキットを購入するのが確かに便利です、以下がリンクです：

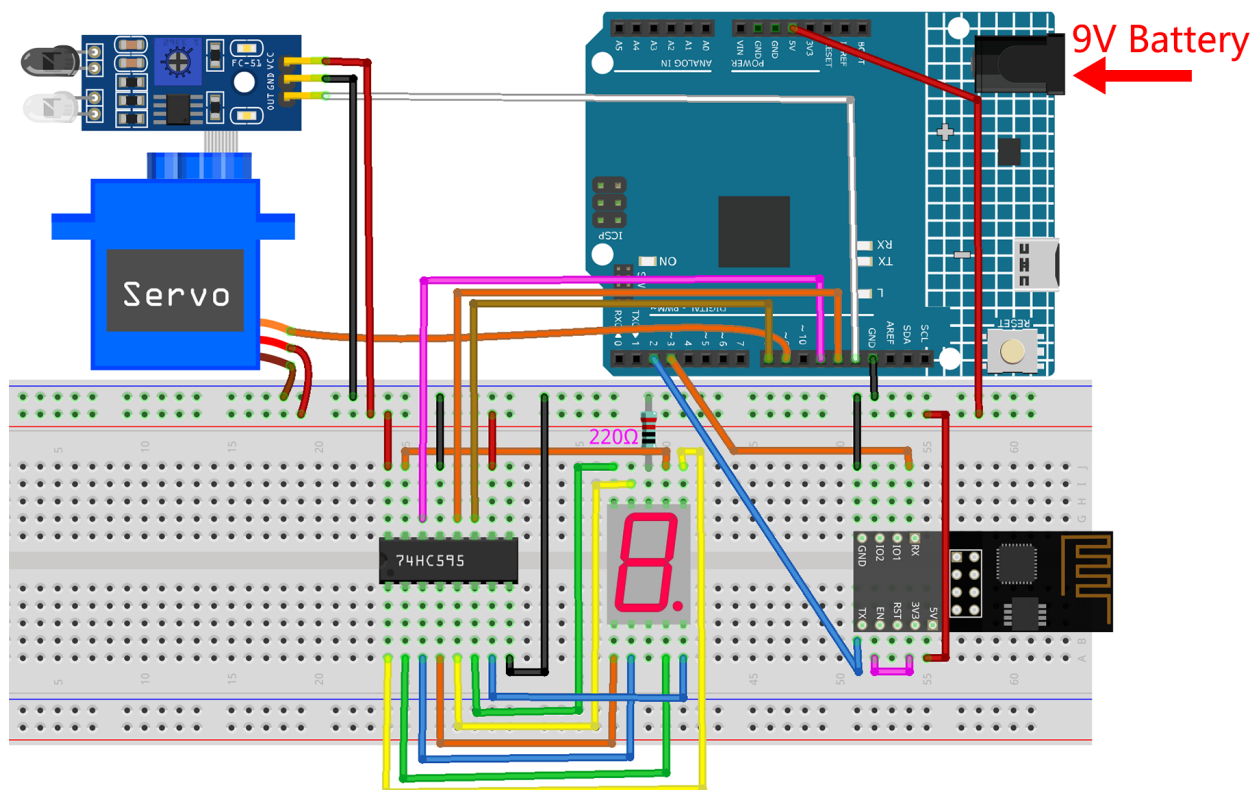
名前	このキットのアイテム	リンク
3 in 1 Starter Kit	380+	

以下のリンクから個別に購入することもできます。

コンポーネントの紹介	購入リンク
<i>Arduino Uno R4 Minima</i>	-
ブレッドボード	
ESP8266 モジュール	
ジャンパーワイヤー	
抵抗器	
サーボ	
障害物回避モジュール	
7 セグメントディスプレイ	
<i>74HC595</i>	

1. 回路を組む

注釈: ESP8266 モジュールは、安定した動作環境を提供するために高電流が必要ですので、9V のバッテリーが接続されていることを確認してください。



2. ダッシュボードを編集

1. 数字を記録するには、**DataStream** ページで **Virtual Pin** タイプの **DataStream** を作成します。DATA TYPE を Integer に設定し、MIN と MAX を 0 と 10 に設定します。

Virtual Pin Datastream

NAME: ALIAS:

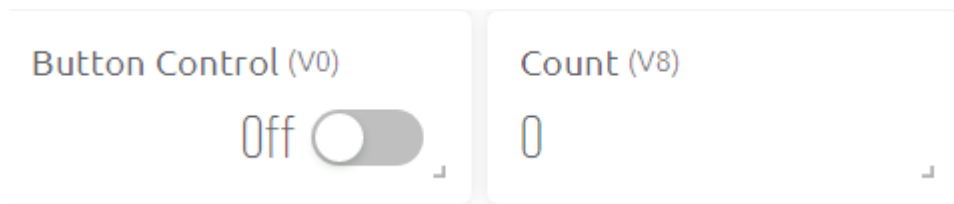
PIN: DATA TYPE:

UNITS:

MIN: MAX: DEFAULT VALUE:

Region: ny3 [Privacy Policy](#)

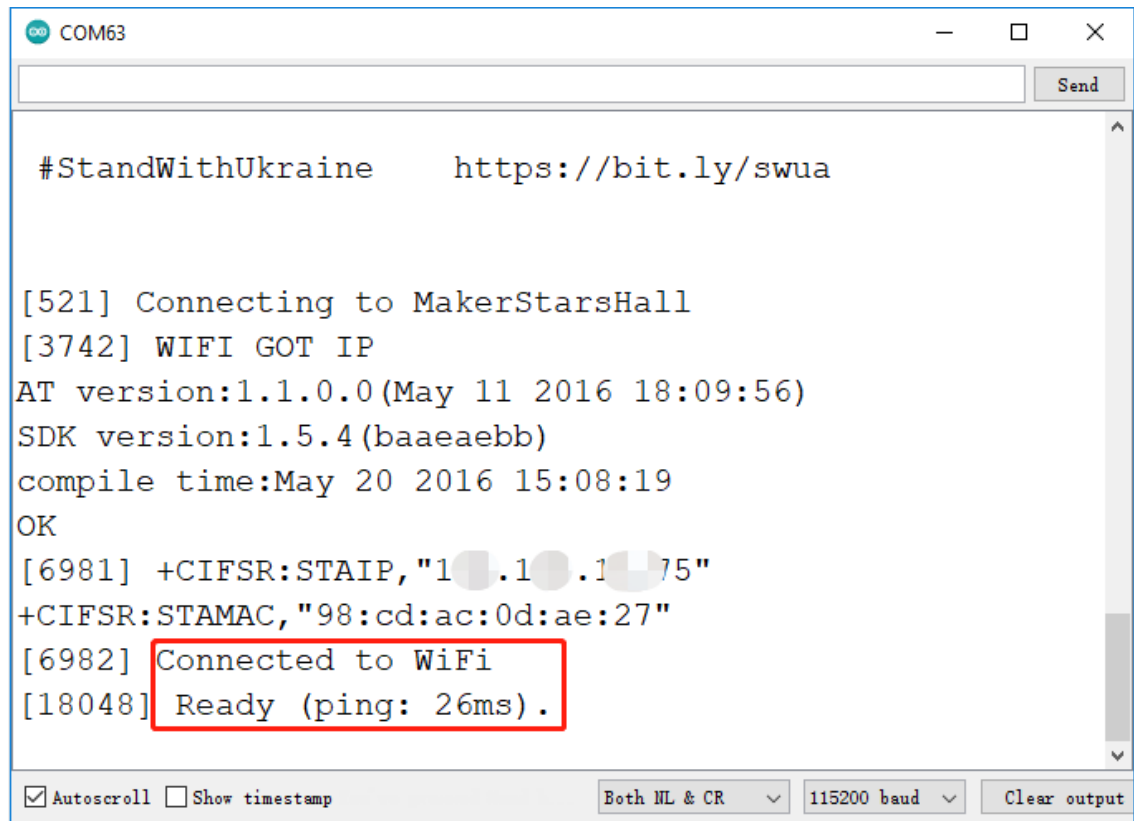
2. 今度は **Web Dashboard** ページに移動し、**Switch** ウィジェットをドラッグしてデータストリームを **V0** に、**Label** ウィジェットをドラッグしてデータストリームを **V8** に設定します。



3. コードの実行

1. 3in1-kit\iot_project\7.current_limiting_gate のパスの下で 7.current_limiting_gate.ino ファイルを開くか、このコードを **Arduino IDE** にコピーします。
2. Template ID、Device Name、および Auth Token を自分のものに置き換えます。使用している WiFi の ssid および password も入力する必要があります。詳しいチュートリアルは、[1.4 R4 ボードを Blynk に接続する](#) を参照してください。

- 適切なボードとポートを選択した後、**Upload** ボタンをクリックします。
- シリアルモニター（ボーレートを 115200 に設定）を開き、成功した接続などのプロンプトが表示されるのを待ちます。



```
#StandWithUkraine https://bit.ly/swua

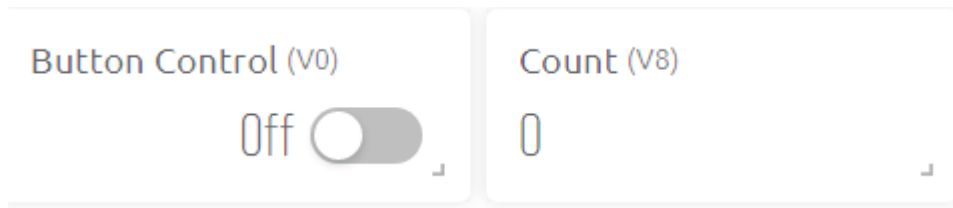
[521] Connecting to MakerStarsHall
[3742] WIFI GOT IP
AT version:1.1.0.0(May 11 2016 18:09:56)
SDK version:1.5.4(baaeaebb)
compile time:May 20 2016 15:08:19
OK
[6981] +CIFSR:STAIP,"1.1.1.175"
+CIFSR:STAMAC,"98:cd:ac:0d:ae:27"
[6982] Connected to WiFi
[18048] Ready (ping: 26ms).
```

注釈: 接続時に ESP is not responding というメッセージが表示された場合は、以下の手順に従ってください。

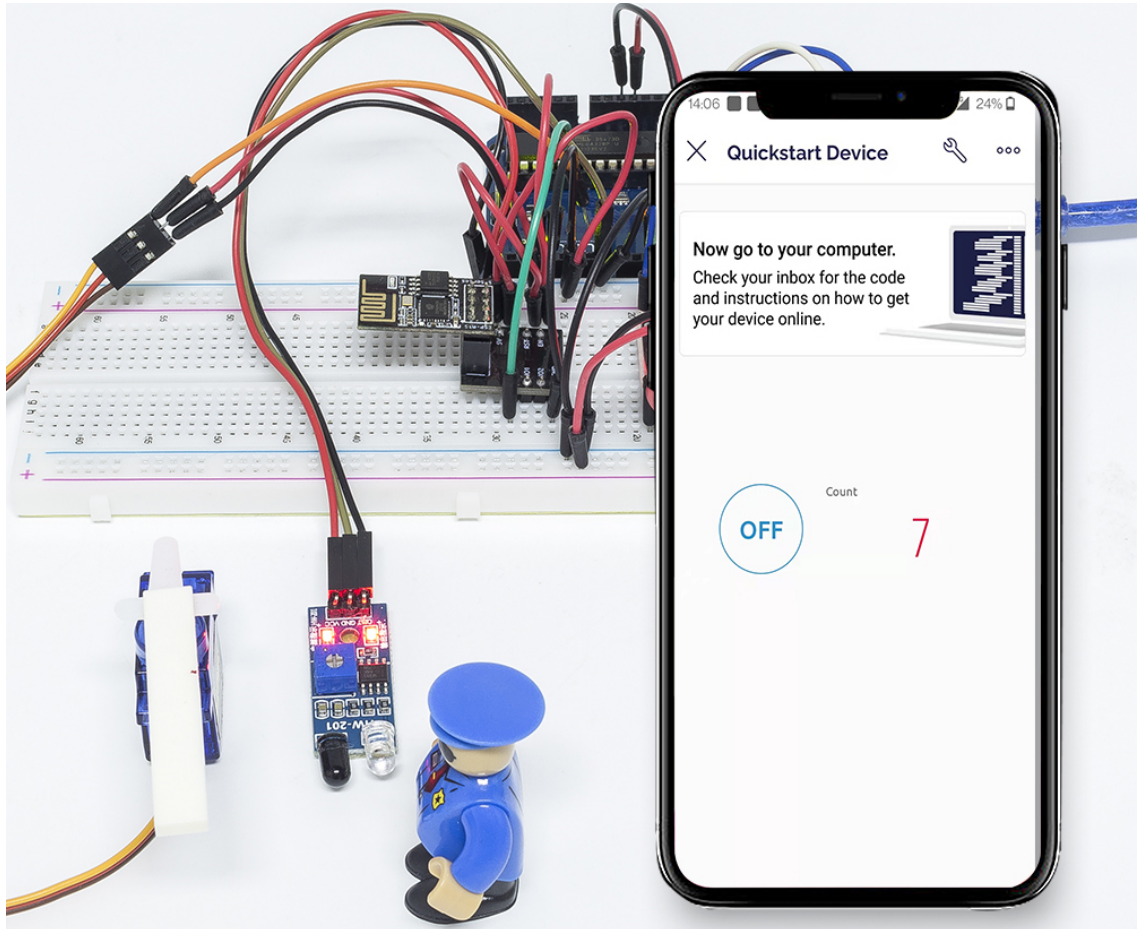
- 9V のバッテリーが接続されていることを確認してください。
- RST ピンを 1 秒間 GND に接続して ESP8266 モジュールをリセットし、その後、それを取り外します。
- R4 ボード上のリセットボタンを押します。

これらの操作を 3~5 回繰り返す必要があることがあります。忍耐強く行ってください。

- Blynk 上の Button Control ウィジェットをクリックしてスマートドアシステムを有効にします。IR 障害物回避モジュールが障害物を検出すると、ゲートが開き、7 セグメントディスプレイと Blynk 上の Count ウィジェットに 1 が加算されます。



6. モバイルデバイスで Blynk を使用したい場合は、[モバイルデバイスでの Blynk の使用方法](#) を参照してください。



どのように動作するのか？

関数 `BLYNK_WRITE(V0)` は **Switch** ウィジェットの状態を取得し、それを変数 `doorFlag` に割り当てます。これはスマートゲートシステムが有効かどうかを判断するために使用されます。

```
BLYNK_WRITE(V0)
{
    doorFlag = param.asInt(); // ゲートを有効にする
}
```

Blynk タイマーでは、`doorFlag` は毎秒判断され、有効な場合、ゲートの主要な関数が実行されます。


```

void myTimerEvent()
{
    if (doorFlag)
    {
        channelEntrance();
    }
}

```

ゲートの主要な関数は channelEntrance() です。オブジェクトがゲートに近づくと（センサーが障害物があることを検出すると）、count は 1 増加します。count を Blynk Cloud のデータストリーム V8 と回路上の 7 セグメントディスプレイに書き込み、ドアを開きます。オブジェクトが現在から欠如する場合、つまりオブジェクトがドアに入った場合、ドアを閉じます。

```

void channelEntrance()
{
    int currentState = digitalRead(irPin); // 0: 障害物 1: 障害物なし
    if (currentState == 0 && lastState == 1) {
        count=(count+1)%10;
        Blynk.virtualWrite(V8, count);
        showNumber(count);
        operateGate(true);
    } else if ((currentState == 1 && lastState == 0)) {
        operateGate(false);
    }
    lastState = currentState;
}

```

関数 showNumber(int num) は、7 セグメントディスプレイに値を表示するために使用されます。

```

void showNumber(int num)
{
    digitalWrite(STcp, LOW); //ST_CPをグラウンドにし、データを送信している間、ローに保持します
    shiftOut(DS, SHcp, MSBFIRST, dataArray[num]);
    digitalWrite(STcp, HIGH); //データを保存するために ST_CP をプルアップします
}

```

関数 operateGate(bool openGate) は、参照が True の場合、ゆっくりとドアを開き、参照が False の場合、ゆっくりとドアを閉じます。

```

void operateGate(bool openGate) {

```

(次のページに続く)

(前のページからの続き)

```
if (openGate == true)
{
    // open gate
    while (angle <= 90) {
        angle++;
        myservo.write(angle);
        delay(5);
    }
} else {
    // close gate
    while (angle >= 0){
        angle--;
        myservo.write(angle);
        delay(5);
    }
}
}
```

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

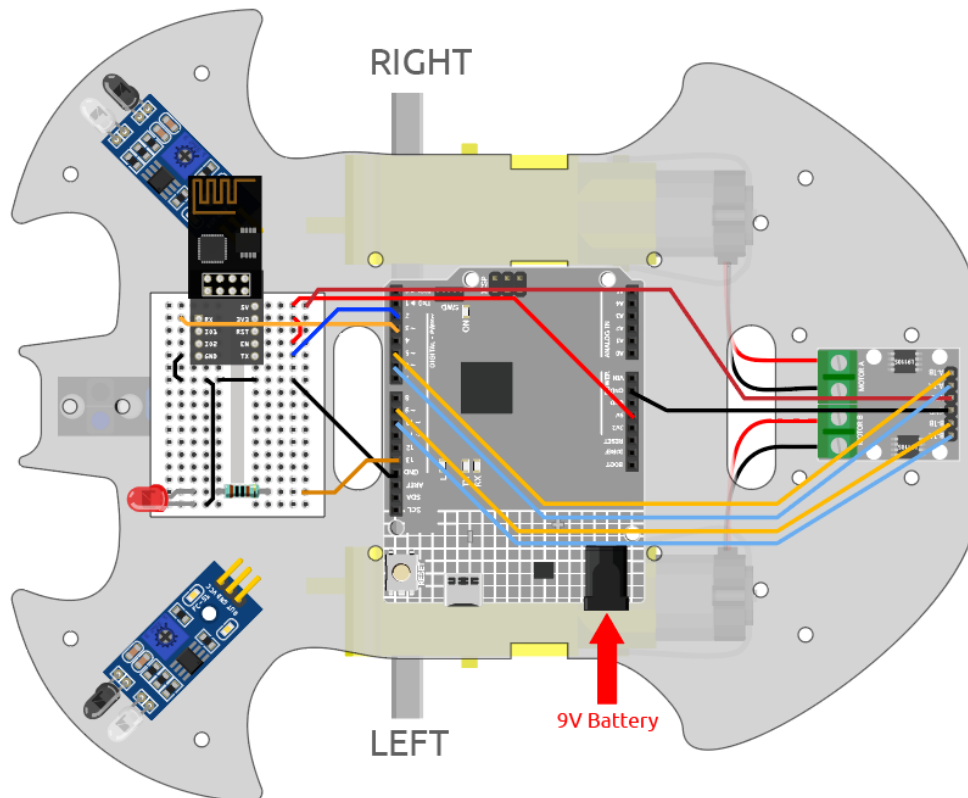
- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

6.8 8. IoT カー

このプロジェクトでは、携帯電話の Blynk アプリを使用して車を制御しました。しかし、車の組み立てや基本的な理解については、[カープロジェクト](#) を参照してください。5G ネットワークの普及時代に、このモードは多くの産業で主要な生産方法の一つとなるかもしれませんので、先取りしてこのプレイを体験してみましょう。

1. 回路を組む



2. ダッシュボードを編集

モバイルの Blynk では Datastream を編集することができないので、これらのステップはウェブサイドで行う必要があります。

1. **Datastream** ページで、ジョイスティックの X 軸の値を記録するための **Virtual Pin** タイプの **Datastream** を作成します。NAME を Xvalue、DATA TYPE を Integer、MIN と MAX を -10 と 10 に設定します。

Quickstart Templ...

Virtual Pin Datastream

NAME: Xvalue ALIAS: Xvalue

PIN: V9 DATA TYPE: Integer

UNITS: None

MIN: -10 MAX: 10 DEFAULT VALUE: 0

+ ADVANCED SETTINGS Cancel Create

Region: ny3 Privacy Policy

2. ジョイスティックの Y 軸の値を記録するための **Virtual Pin** タイプの **Datastream** を作成します。NAME を Yvalue、DATA TYPE を Integer、MIN と MAX を -10 と 10 に設定します。

Virtual Pin Datastream

NAME: Yvalue ALIAS: Yvalue

PIN: V10 DATA TYPE: Integer

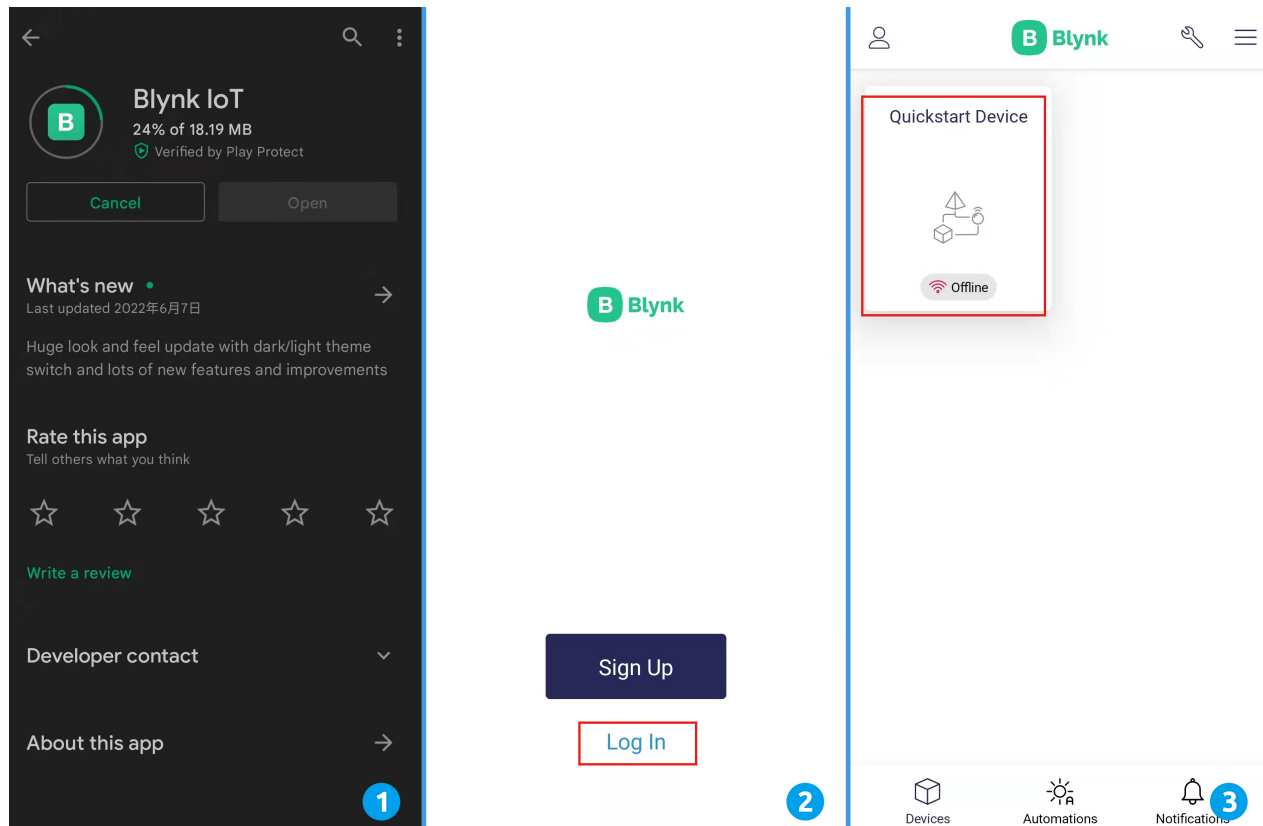
UNITS: None

MIN: -10 MAX: 10 DEFAULT VALUE: 0

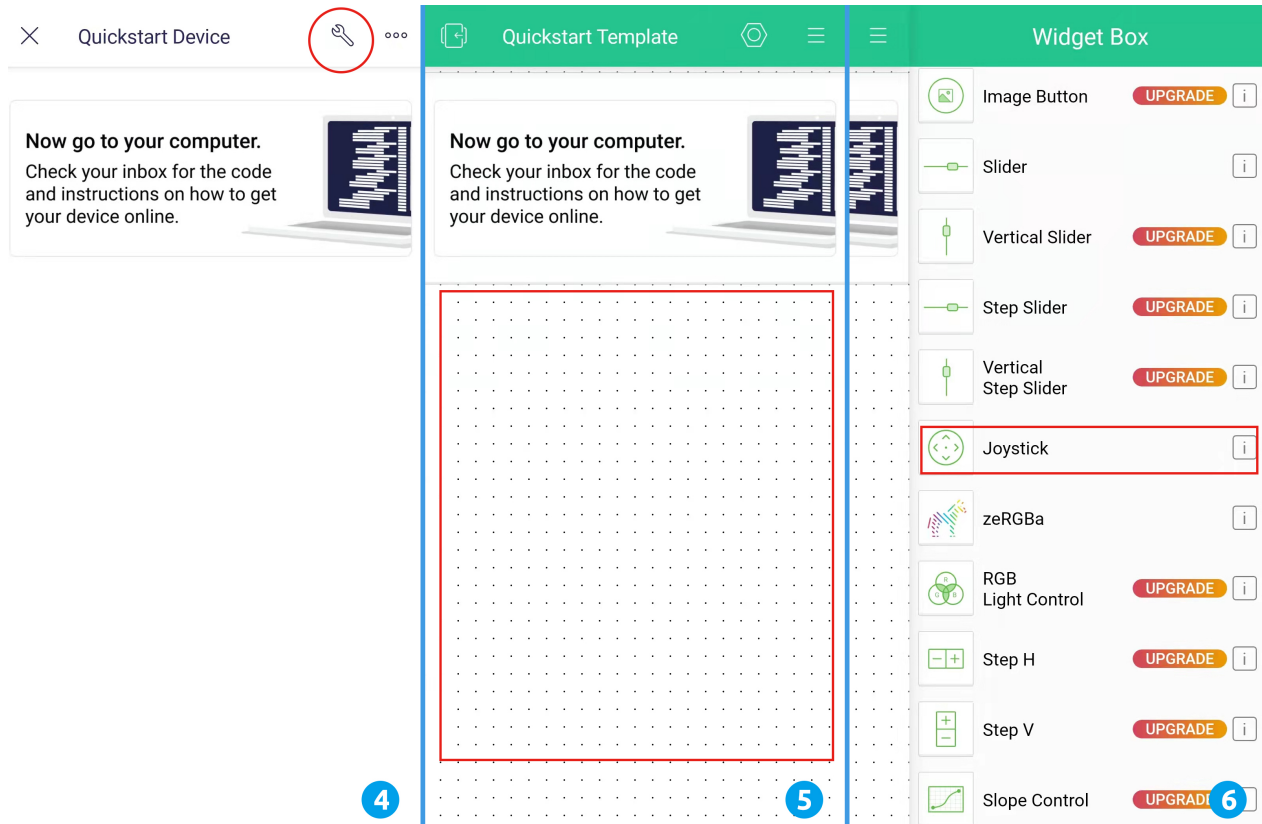
+ ADVANCED SETTINGS Cancel Create

次に、以下の手順を携帯電話で行ってください。

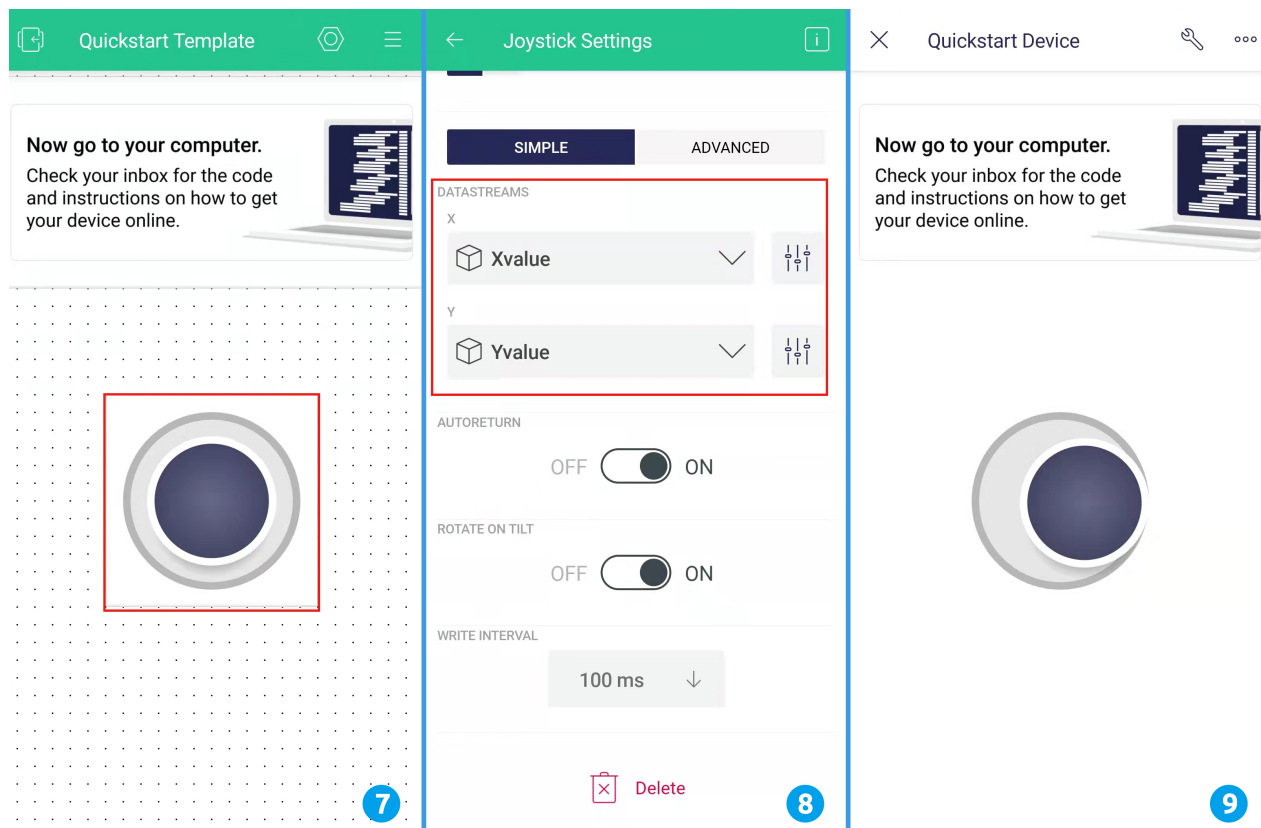
1. GOOGLE Play または APP Store で "Blynk IoT" (Blynk(legacy) ではありません) を検索してダウンロードします。
2. アプリを開いてログインします。このアカウントは、ウェブクライアントで使ったアカウントと同じである必要があります。
3. 次にダッシュボードに移動します（持っていない場合は作成します）。モバイルとウェブのダッシュボードは互いに独立していることがわかります。



4. 編集アイコンをクリックします。
5. 空白のエリアをクリックします。
6. ジョイスティックウィジェットを選択します。

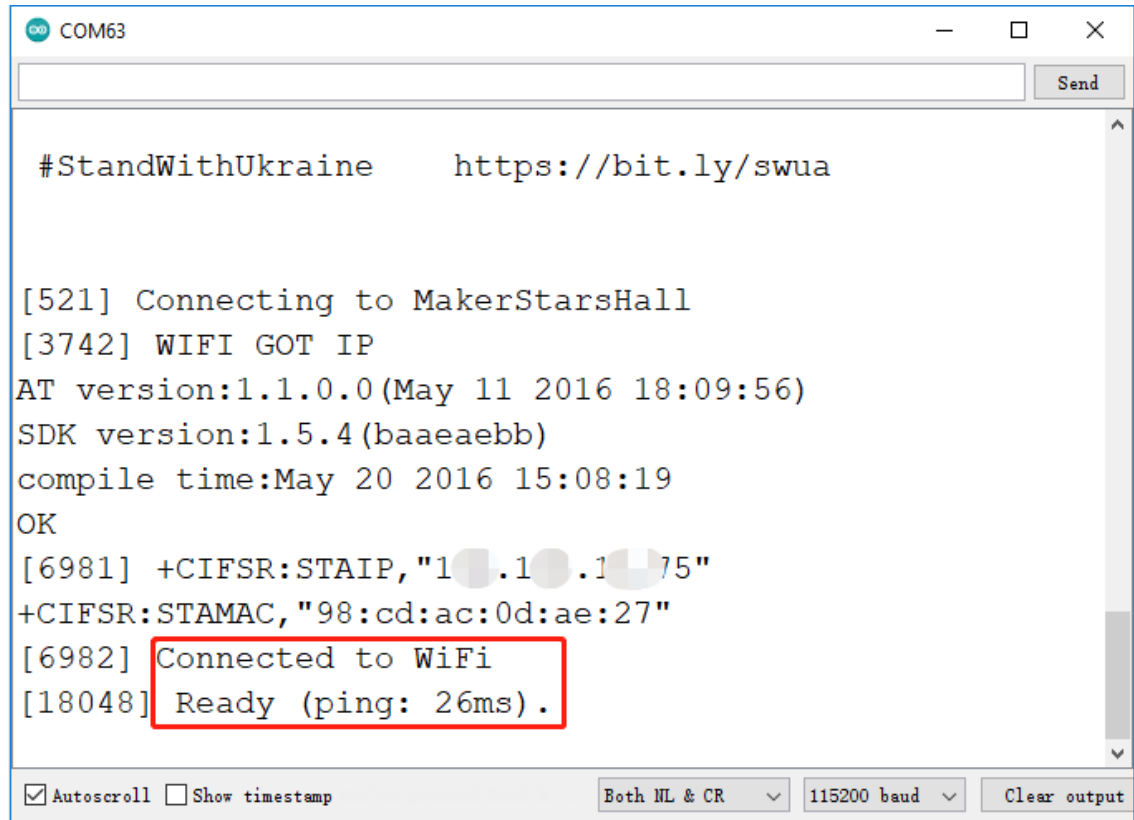


7. 空白のエリアにジョイスティックウィジェットが表示されますので、それをクリックします。
8. ジョイスティックの設定が表示されますので、先ほど datastreams で設定した Xvalue と Yvalue を選択します。
9. ダッシュボードページに戻って、ジョイスティックを操作したいときに操作します。



3. コードの実行

1. 3in1-kit\iot_project\8.iot_car のパス下で 8.iot_car.ino ファイルを開くか、このコードを **Arduino IDE** にコピーします。
2. 自分の Template ID、Device Name、および Auth Token に置き換えます。使用している WiFi の ssid および password も入力する必要があります。詳しいチュートリアルは、[1.4 R4 ボードを Blynk に接続する](#) を参照してください。
3. 適切なボードとポートを選択した後、**Upload** ボタンをクリックします。
4. シリアルモニター（ボーレートを 115200 に設定）を開き、成功した接続などのプロンプトが表示されるのを待ちます。



```
COM63

#StandWithUkraine      https://bit.ly/swua

[521] Connecting to MakerStarsHall
[3742] WIFI GOT IP
AT version:1.1.0.0(May 11 2016 18:09:56)
SDK version:1.5.4(baaeaebb)
compile time:May 20 2016 15:08:19
OK
[6981] +CIFSR:STAIP,"1.1.1.175"
+CIFSR:STAMAC,"98:cd:ac:0d:ae:27"
[6982] Connected to WiFi
[18048] Ready (ping: 26ms).
```

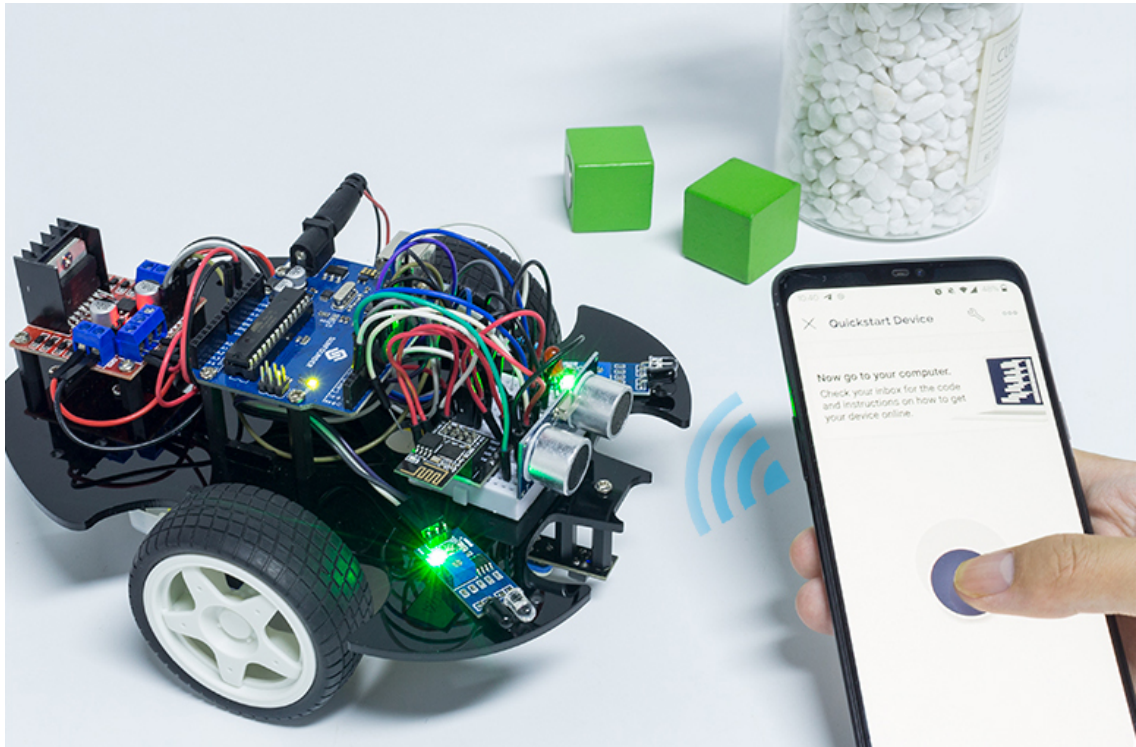
☒ Autoscroll ☐ Show timestamp Both NL & CR 115200 baud Clear output

注釈: 接続時に ESP is not responding というメッセージが表示された場合、以下の手順に従ってください。

- 9V のバッテリーが接続されていることを確認してください。
- RST ピンを 1 秒間 GND に接続して ESP8266 モジュールをリセットし、その後、それを取り外します。
- R4 ボード上のリセットボタンを押します。

これらの操作を 3~5 回繰り返す必要があることがあります。忍耐強く行ってください。

5. 今、USB ケーブルを抜いて、9V のバッテリーだけでカーを起動します。LED が点灯するのを待ちます。これは、車が Blynk に接続されていることを示しています。
6. 携帯電話で Blynk を開き、ジョイスティックウィジェットを使用して車の動きを制御できます。



どのように動作するのか？

これらの関数は、車の動きを制御するために使用されます。

```
void moveForward(int speed) {...}
void moveBackward(int speed) {...}
void turnRight(int speed) {...}
void turnLeft(int speed) {...}
void stopMove() {...}
```

IoT セクションは、ジョイスティックウィジェットの値を読み取り、それらを変数 Xvalue および Yvalue に割り当てます。

```
int Xvalue = 0;
int Yvalue = 0;

BLYNK_WRITE(V9)
{
    Xvalue = param.asInt();
}

BLYNK_WRITE(V10)
{
```

(次のページに続く)

(前のページからの続き)

```
Yvalue = param.asInt();
}
```

loop() で、Xvalue および Yvalue に基づいて車にさまざまなアクションを実行させます。

```
if (Yvalue >= 5) {
    moveForward(255);
} else if (Yvalue <= -5) {
    moveBackward(255);
} else if (Xvalue >= 5) {
    turnRight(150);
} else if (Xvalue <= -5) {
    turnLeft(150);
} else {
    stopMove();
}
```

また、Blynk Cloud に接続されている場合に LED を点灯させるために、loop() にネットワークステータスの判定を追加します。

```
if (!Blynk.connected()) {
    digitalWrite(ledPin, LOW);
    Serial.print("offline!");
    bool result = Blynk.connect();
    Serial.println(result);
} else {
    digitalWrite(ledPin, HIGH);
}
```

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。

- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

第 7 章

よくある質問

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

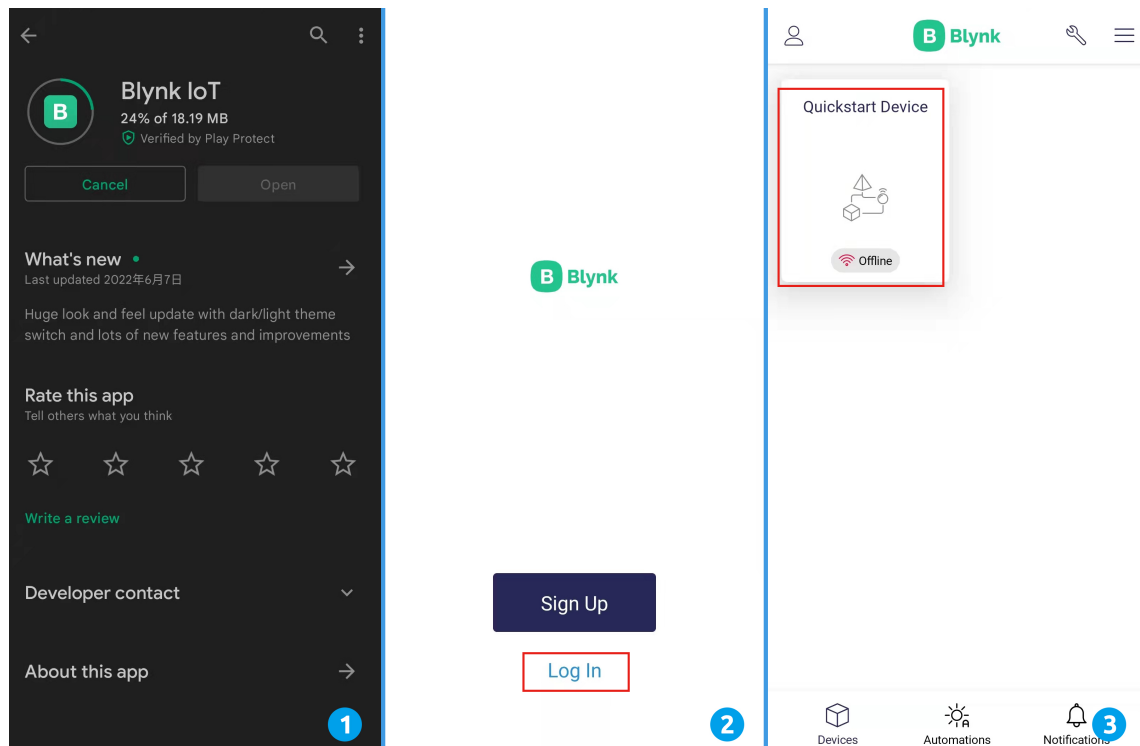
私たちと一緒に探索し、創造する準備はできていますか？ [\[\]](#) をクリックして今すぐ参加しましょう！

7.1 モバイルデバイスでの Blynk の使用方法

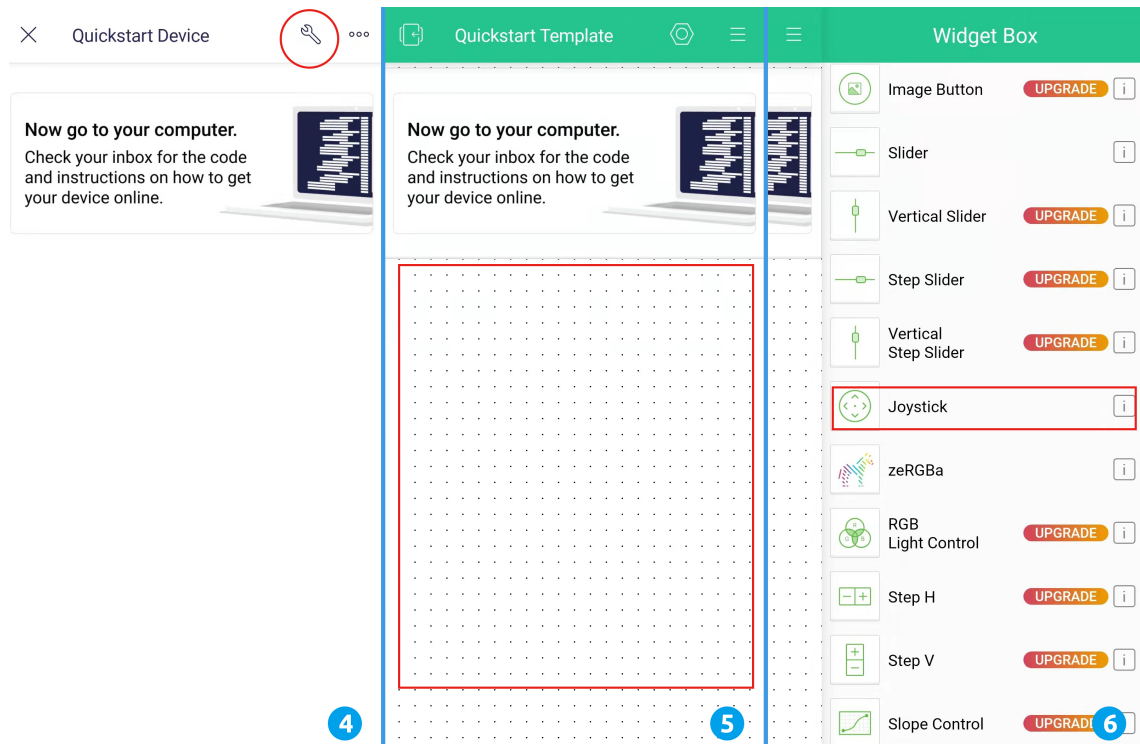
注釈: データストリームは Blynk のウェブでのみ作成可能なため、ウェブでデータストリームを作成するための異なるプロジェクトを参照し、以下のチュートリアルに従ってモバイルデバイスの Blynk でウィジェットを作成する必要があります。

1. お使いのモバイルデバイスで Google Play または APP Store を開き、"Blynk IoT" (Blynk(legacy) ではありません) を検索してダウンロードします。
2. APP を開いた後、ログインします。このアカウントはウェブクライアントで使用されるものと同じでなければなりません。

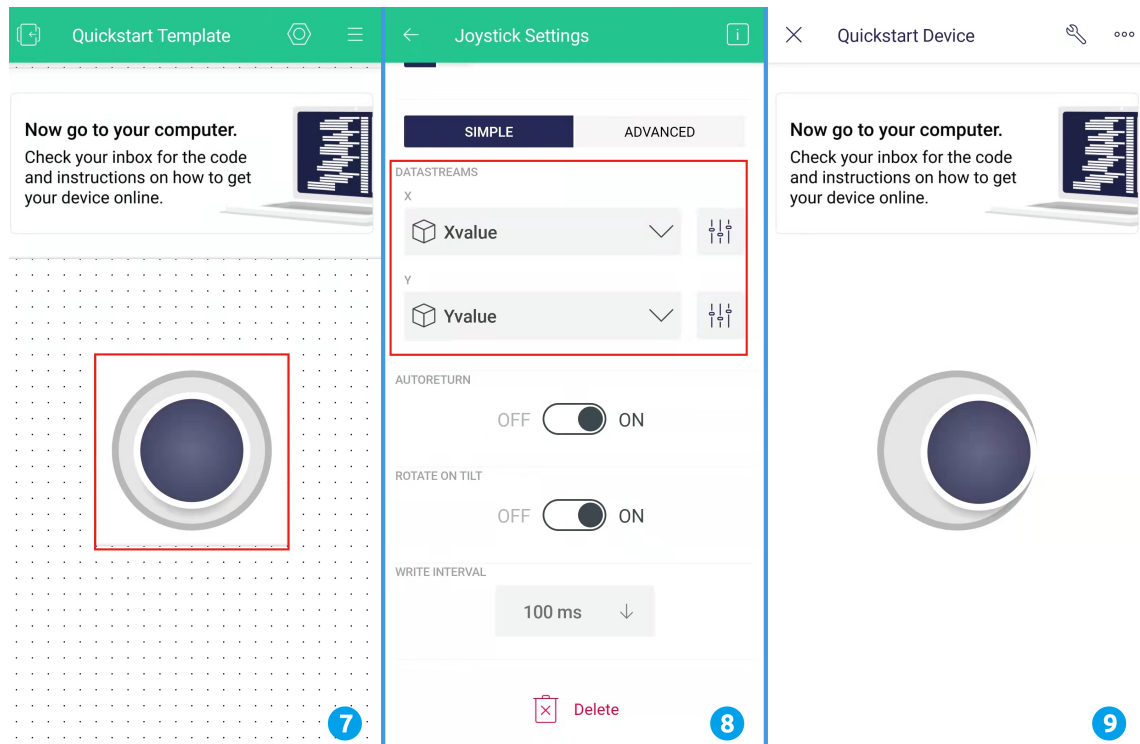
3. 次に、**Dashboard** に移動します（持っていない場合は作成してください）。モバイルとウェブの **Dashboard** は互いに独立していることがわかります。



4. **Edit** アイコンをクリックします。
5. 空白の領域をクリックします。
6. ウェブページと同じウィジェットを選択します。たとえば、**Joystick** ウィジェットを選択します。



7. これで空白の領域に **Joystick** ウィジェットが表示されますので、クリックします。
8. **Joystick** の設定が表示されるので、先ほどのウェブページで設定した **Xvalue** と **Yvalue** のデータストリームを選択します。各ウィジェットは各プロジェクトで異なるデータストリームに対応していることに注意してください。
9. **Joystick** ページに戻り、必要に応じて **Dashboard** を操作できます。



注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

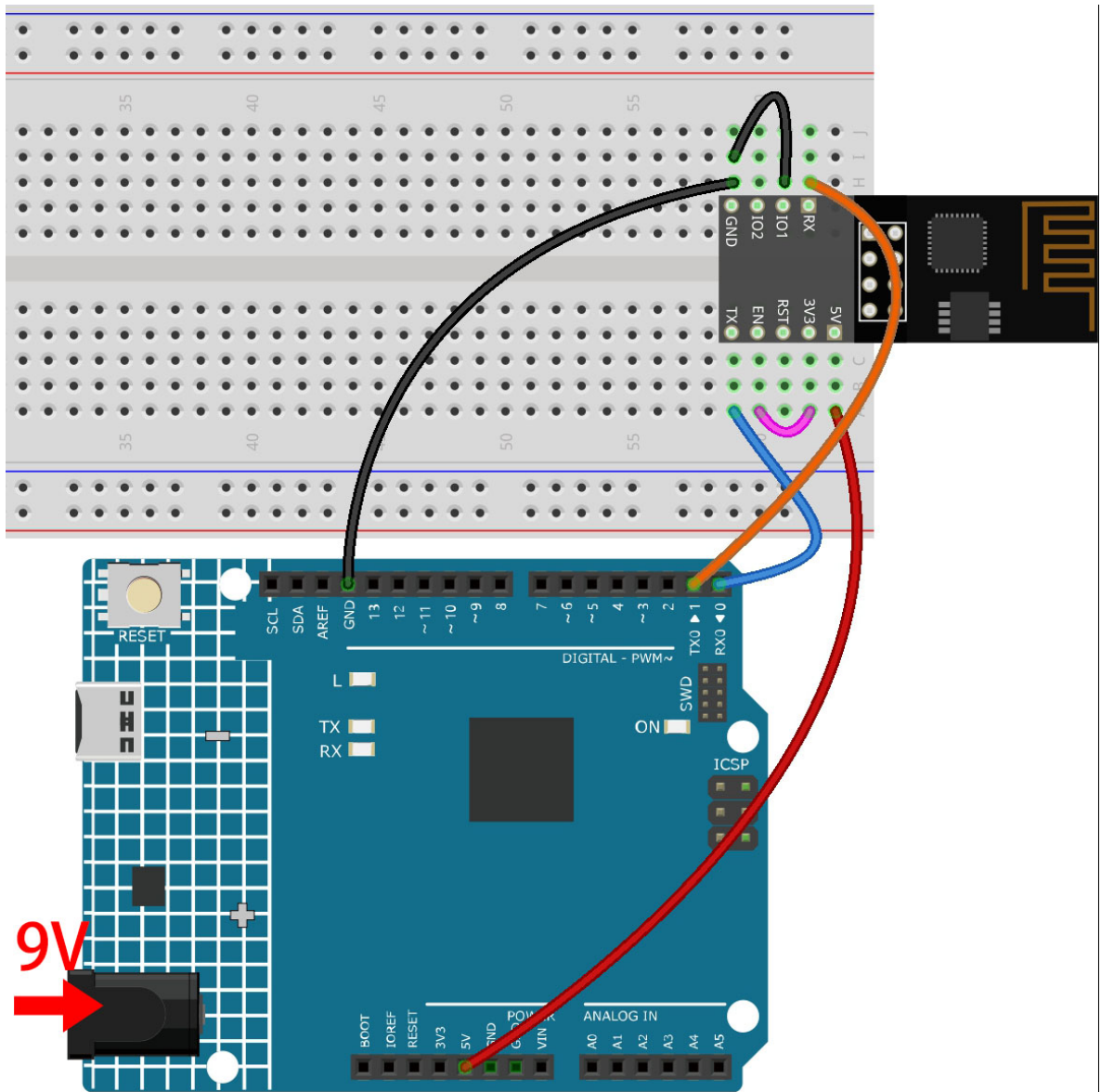
私たちと一緒に探索し、創造する準備はできていますか？ [] をクリックして今すぐ参加しましょう！

7.2 ESP8266 モジュールのファームウェアの再書き込み方法は？

7.2.1 R4 でファームウェアを再書き込む

1. 回路を組む

ESP8266 と Arduino UNO R4 ボードを接続します。



2. R4 に次のコードをアップロード

```
void setup() {
  Serial.begin(115200);
  Serial1.begin(115200);
}
```

(次のページに続く)

(前のページからの続き)

```
}

void loop() {
    if (Serial.available()) {          // シリアル (USB) に何かが入った場合、
        Serial1.write(Serial.read()); // それを読み取って送信します Serial1 (ピン 0 と 1)
    }
    if (Serial1.available()) {          // Serial1 (ピン 0 と 1) に何かが入った場合
        Serial.write(Serial1.read()); // 読み取って送信 シリアル (USB)
    }
}
```

3. ファームウェアの書き込み

- Windows を使用している場合、以下の手順でファームウェアを書き込んでください。

1. ファームウェアと書き込みツールをダウンロードする。

- ESP8266 ファームウェア

2. 解凍すると、4 つのファイルが表示されます。

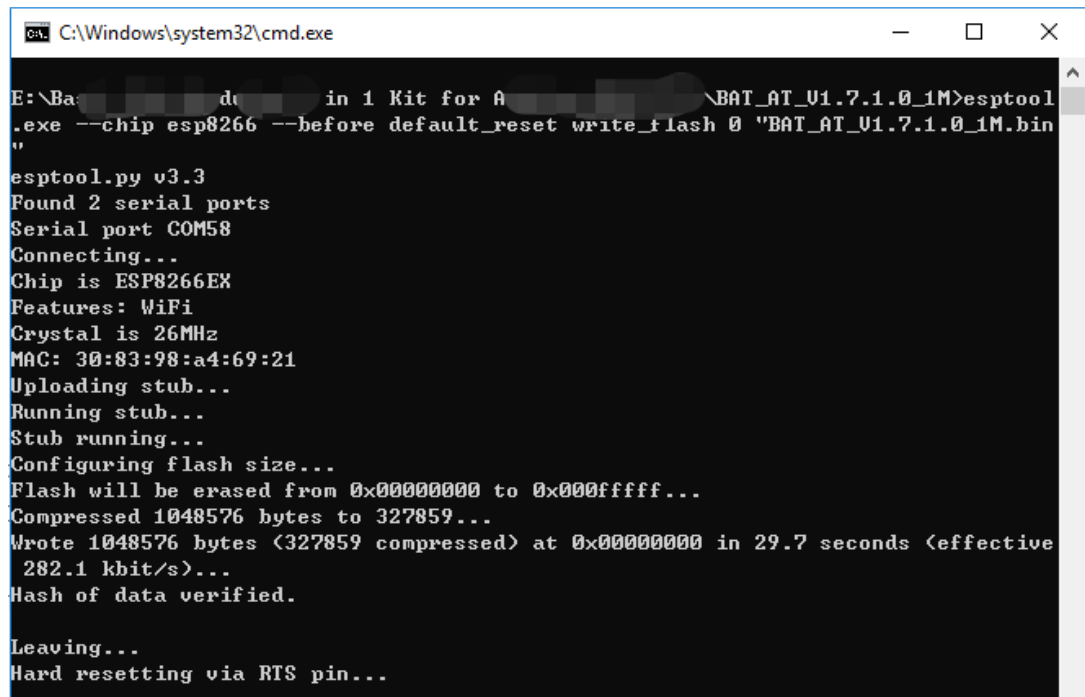
- BAT_AT_V1.7.1.0_1M.bin: ESP8266 モジュールに書き込むファームウェア。

- esptool.exe: Windows 用のコマンドラインユーティリティです。

- install_r3.bat: Windows システム用のコマンドパッケージ、このファイルをダブルクリックすると、ファイル内のすべてのコマンドが実行されます。

- install_r4.bat: install_r3.bat と同様ですが、UNO R4 ボード専用です。

3. install_r4.bat をダブルクリックして、ファームウェアの書き込みを開始します。以下のプロンプトが表示される場合、ファームウェアは正常にインストールされました。



```

C:\Windows\system32\cmd.exe

E:\Ba... du... in 1 Kit for A... \BAT_AT_V1.7.1.0_1M>esptool
.exe --chip esp8266 --before default_reset write_flash 0 "BAT_AT_V1.7.1.0_1M.bin
"
esptool.py v3.3
Found 2 serial ports
Serial port COM58
Connecting...
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 30:83:98:a4:69:21
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Flash will be erased from 0x00000000 to 0x000fffff...
Compressed 1048576 bytes to 327859...
Wrote 1048576 bytes (327859 compressed) at 0x00000000 in 29.7 seconds (effective
 282.1 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

```

注釈: 書き込みに失敗した場合、以下の点を確認してください。

- ESP8266 モジュールをリセットします (ESP8266 アダプタの RST を GND に差し込み、それを抜く)
- 配線が正しいか確認してください。
- コンピュータがボードを正しく認識しているか、ポートが占有されていないことを確認してください。
- install.bat ファイルを再度開きます。

• Mac OS を使用している場合、以下の手順でファームウェアを書き込んでください。

1. Esptool をインストールするための以下のコマンドを使用します。Esptool は Python ベースで、オープンソースの、Espressif チップ内の ROM ブートローダと通信するためのプラットフォーム非依存のユーティリティです。

```
python3 -m pip install --upgrade pip
python3 -m pip install esptool
```

2. esptool が正しくインストールされていれば、python3 -m esptool を実行すると [usage: esptool] というメッセージが出力されます。
3. ファームウェアをダウンロードします。

- ESP8266 ファームウェア
4. 解凍すると、4 つのファイルが表示されます。
- BAT_AT_V1.7.1.0_1M.bin: ESP8266 モジュールに書き込むファームウェア。
 - esptool.exe: Windows 用のコマンドラインユーティリティです。
 - install_r3.bat: Windows システム用のコマンドパッケージ。
 - install_r4.bat: install_r3.bat と同様ですが、UNO R4 ボード専用です。
5. ターミナルを開き、cd コマンドを使ってダウンロードしたファームウェア・フォルダーに入り、以下のコマンドを実行して既存のファームウェアを消去し、新しいファームウェアを焼き直す。

```
python3 -m esptool --chip esp8266 --before no_reset_no_sync erase_flash
python3 -m esptool --chip esp8266 --before no_reset_no_sync write_flash 0
↪ "BAT_AT_V1.7.1.0_1M.bin"
```

6. 以下のプロンプトが表示される場合、ファームウェアは正常にインストールされました。

```
MacdeMacBook-Pro:BAT_AT_V1.7.1.0_1M mac$ python3 -m esptool --chip esp8266 --before default_reset w
rite_flash 0 "BAT_AT_V1.7.1.0_1M.bin"
esptool.py v4.3
[Found 3 serial ports
Serial port /dev/cu.usbmodem143401
Connecting...
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 30:83:98:a4:69:21
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Flash will be erased from 0x00000000 to 0x000fffff...
Compressed 1048576 bytes to 327859...
Wrote 1048576 bytes (327859 compressed) at 0x00000000 in 29.6 seconds (effective 283.0 kbit/s)...
Hash of data verified.

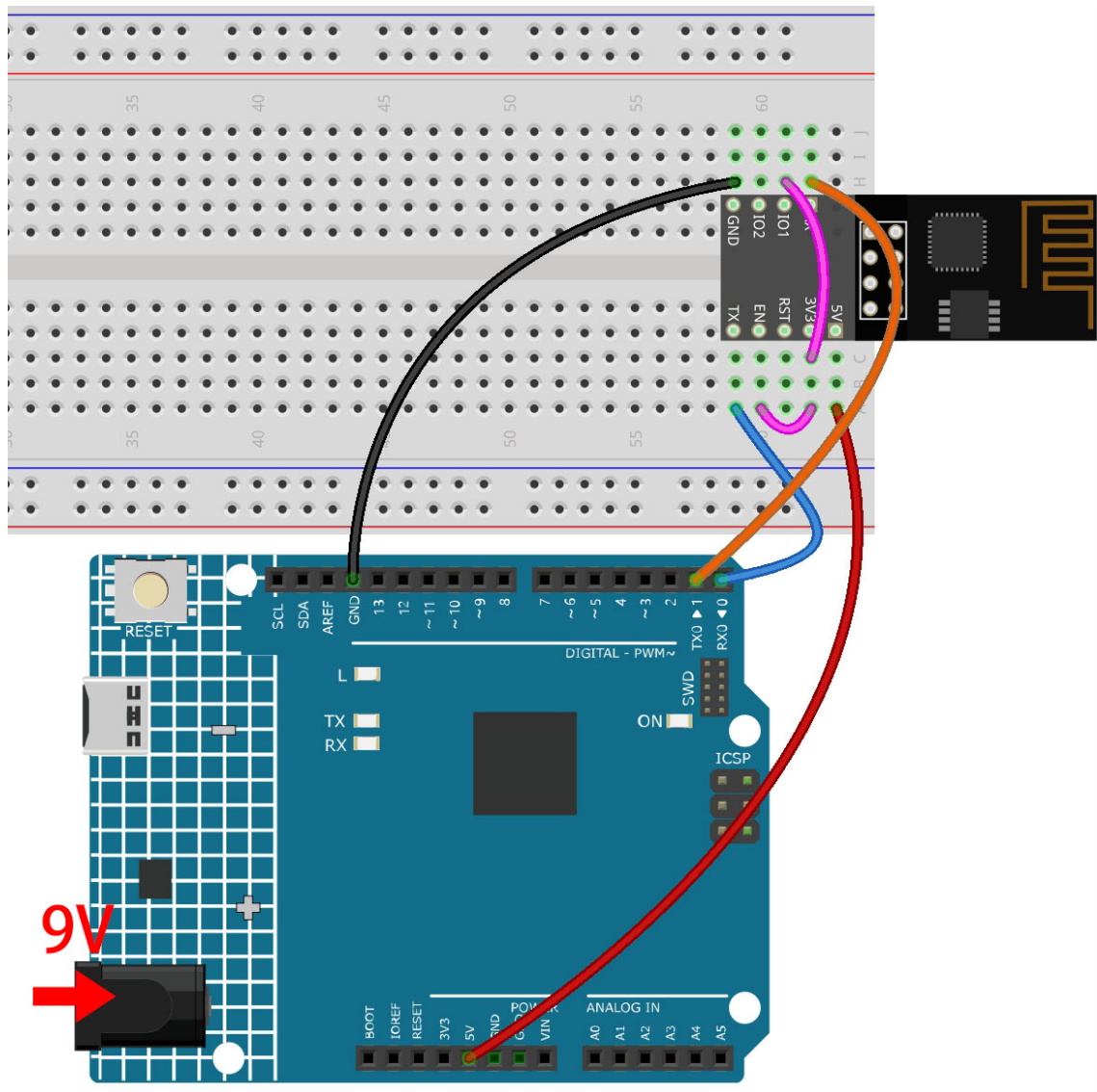
Leaving...
Hard resetting via RTS pin...
MacdeMacBook-Pro:BAT_AT_V1.7.1.0_1M mac$
```

注釈: 書き込みに失敗した場合、以下の点を確認してください。

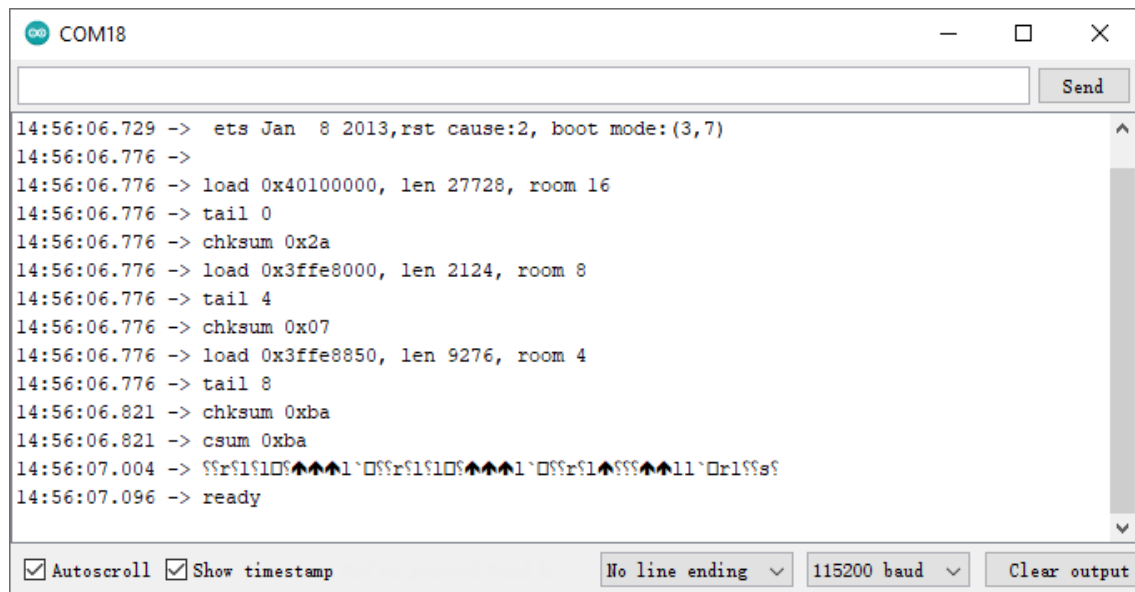
- ESP8266 モジュールをリセットします (ESP8266 アダプタの RST を GND に差し込み、それを抜く)。
- 配線が正しいか確認してください。
- コンピュータがボードを正しく認識しているか、ポートが占有されていないことを確認してください。
- install.bat ファイルを再度開きます。

4. テスト

1. 元の配線のまま、IO1 を 3V3 に接続します。



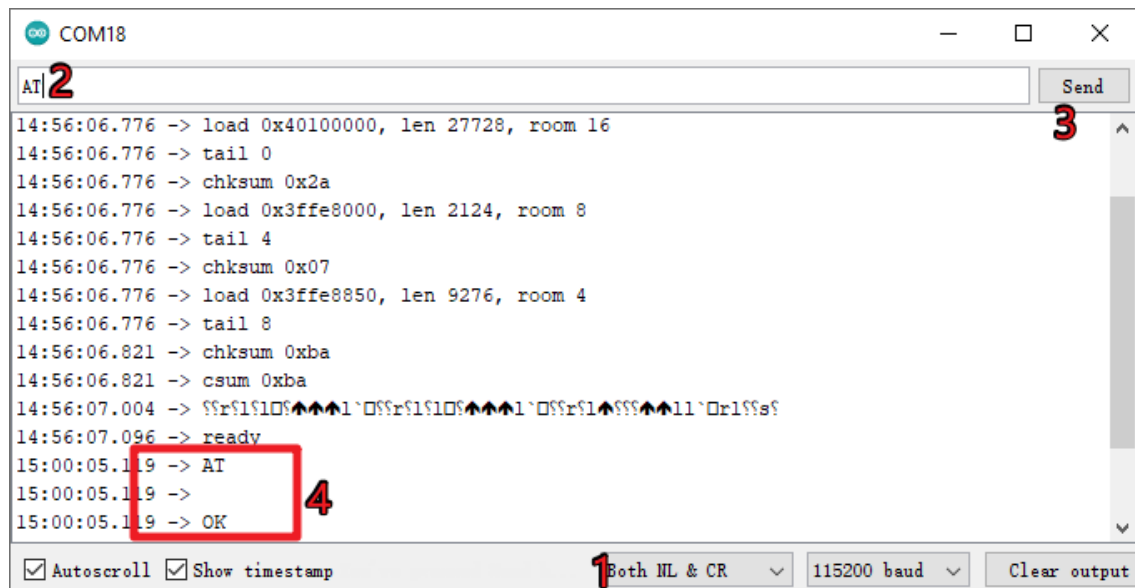
2. 右上の虫眼鏡アイコン（シリアルモニタ）をクリックし、ボーレートを **115200** に設定すると、ESP8266 モジュールの情報が表示されます。



注釈:

- ready が表示されない場合、ESP8266 モジュールをリセットして (RST を GND に接続) シリアルモニタを再度開くと良いでしょう。

3. **NEWLINE DROPDOWN BOX** をクリックし、ドロップダウンオプションで both NL & CR を選択し、AT を入力します。OK が返される場合、ESP8266 がボードと正常に接続されたことを意味します。

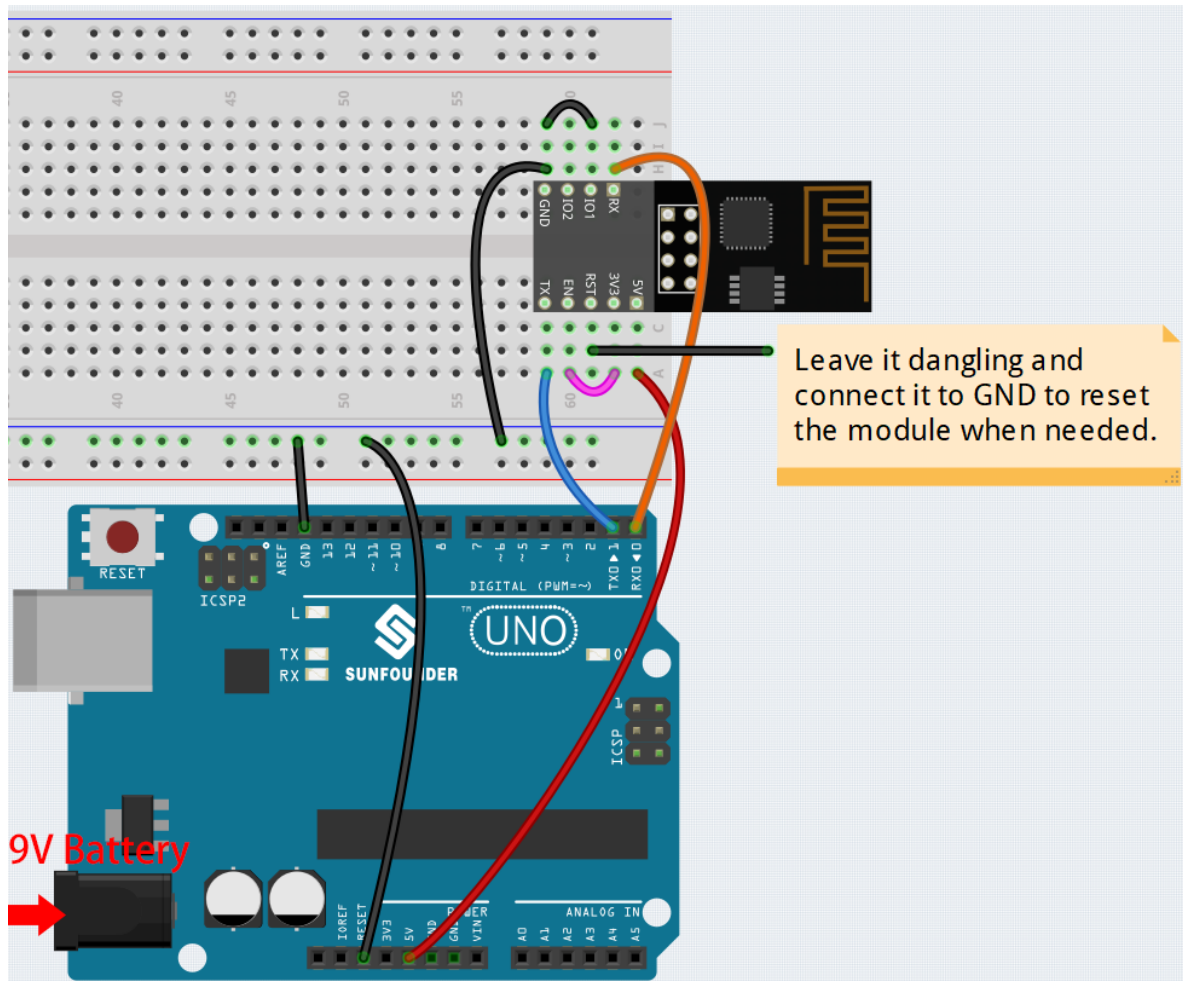


これで、1.1 ESP8266 の設定 に従って、ESP8266 モジュールの動作モードやボーレートを設定することができます。

7.2.2 R3 でファームウェアを再書き込む

1. 回路を作成する

ESP8266 と SunFounder R3 ボードを接続します。



2. ファームウェアの書き込み

- Windows を使用している場合、以下の手順に従ってファームウェアを書き込んでください。

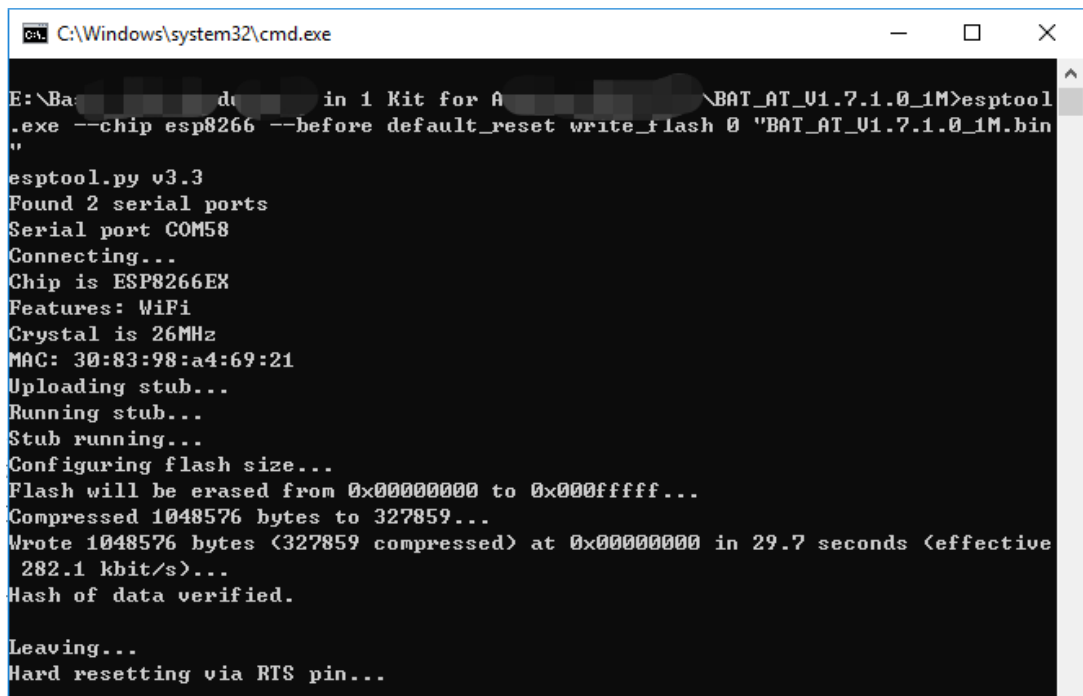
1. ファームウェアと書き込みツールをダウンロードします。

– ESP8266 Firmware

2. 解凍すると、4 つのファイルが表示されます。

BAT_AT_V1.7.1.0_1M.bin
esptool.exe
install.bat

- BAT_AT_V1.7.1.0_1M.bin: ESP8266 モジュールに書き込むファームウェアです。
 - esptool.exe: Windows 用のコマンドラインユーティリティです。
 - install_r3.bat: Windows システム用のコマンドパッケージで、このファイルをダブルクリックすると、ファイル内のすべてのコマンドが実行されます。
 - install_r4.bat: install_r3.bat と同じですが、UNO R4 ボード専用です。
3. install_r3.bat をダブルクリックしてファームウェアの書き込みを開始します。次のプロンプトが表示されれば、ファームウェアが正常にインストールされました。



```
C:\Windows\system32\cmd.exe
E:\Ba... du... in 1 Kit for A... \BAT_AT_V1.7.1.0_1M>esptool
.exe --chip esp8266 --before default_reset write_flash 0 "BAT_AT_V1.7.1.0_1M.bin
"
esptool.py v3.3
Found 2 serial ports
Serial port COM58
Connecting...
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 30:83:98:a4:69:21
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Flash will be erased from 0x00000000 to 0x000fffff...
Compressed 1048576 bytes to 327859...
Wrote 1048576 bytes (327859 compressed) at 0x00000000 in 29.7 seconds (effective
282.1 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

注釈: 書き込みが失敗した場合は、以下の点を確認してください。

- ESP8266 アダプタの RST を GND に挿入し、それを抜くことで ESP8266 モジュールをリセットします。
 - 配線が正しいか確認します。
 - コンピュータがボードを正しく認識しているか、そしてポートが占有されていないことを確認します。
 - install.bat ファイルを再度開きます。
- Mac OS システムを使用している場合、以下の手順に従ってファームウェアを書き込んでください。

1. Esptool をインストールするための次のコマンドを使用します。Esptool は、Espressif チップの ROM

ブートローダーと通信するための Python ベースのオープンソースのプラットフォームに依存しないユーティリティです。




```
python3 -m pip install --upgrade pip
python3 -m pip install esptool
```

2. esptool が正しくインストールされていれば、`python3 -m esptool` を実行すると `[usage: esptool]` というメッセージが出力されます。

3. ファームウェアをダウンロードします。

– ESP8266 Firmware

4. 解凍すると、3 つのファイルが表示されます。

 BAT_AT_V1.7.1.0_1M.bin
 esptool.exe
 install.bat

– BAT_AT_V1.7.1.0_1M.bin: ESP8266 モジュールに書き込むファームウェアです。

– esptool.exe: Windows 用のコマンドラインユーティリティです。

– install_r3.bat: Windows システム用のコマンドパッケージです。

– install_r4.bat: install_r3.bat と同じですが、UNO R4 ボード専用です。

5. ターミナルを開き、先ほどダウンロードしたファームウェアのフォルダに `cd` コマンドを使用して移動します。次に、既存のファームウェアを消去し、新しいファームウェアを再書き込むための次のコマンドを実行します。

```
python3 -m esptool --chip esp8266 --before default_reset erase_flash
python3 -m esptool --chip esp8266 --before default_reset write_flash 0 "BAT_
AT_V1.7.1.0_1M.bin"
```

6. 次のプロンプトが表示されれば、ファームウェアが正常にインストールされました。

```
BAT_AT_V1.7.1.0_1M -- -bash -- 99x22
MacdeMacBook-Pro:BAT_AT_V1.7.1.0_1M mac$ python3 -m esptool --chip esp8266 --before default_reset write_flash 0 "BAT_AT_V1.7.1.0_1M.bin"
esptool.py v4.3
[Found 3 serial ports
Serial port /dev/cu.usbmodem143401
Connecting...
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 30:83:98:a4:69:21
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Flash will be erased from 0x00000000 to 0x000fffff...
Compressed 1048576 bytes to 327859...
Wrote 1048576 bytes (327859 compressed) at 0x00000000 in 29.6 seconds (effective 283.0 kbit/s)...
Hash of data verified.

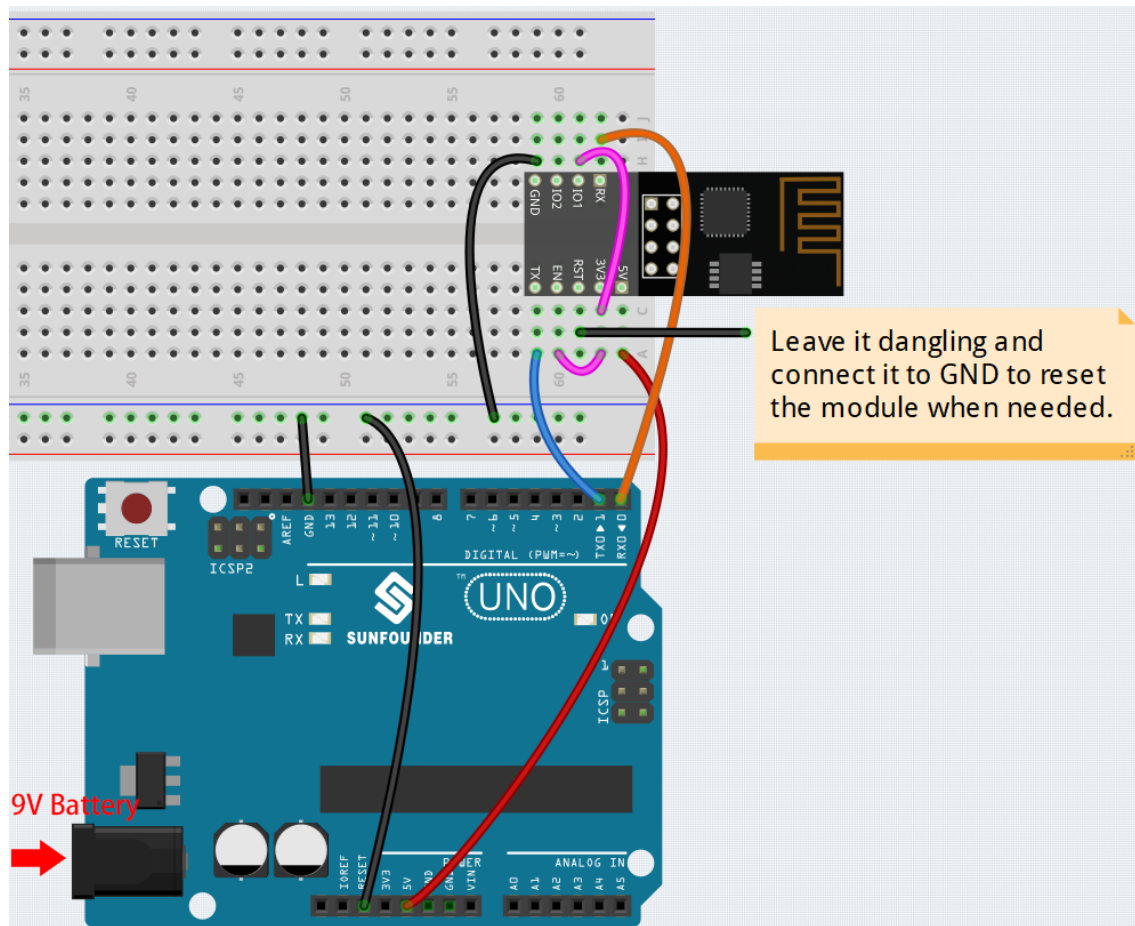
Leaving...
Hard resetting via RTS pin...
MacdeMacBook-Pro:BAT_AT_V1.7.1.0_1M mac$
```

注釈: 書き込みが失敗した場合は、以下の点を確認してください。

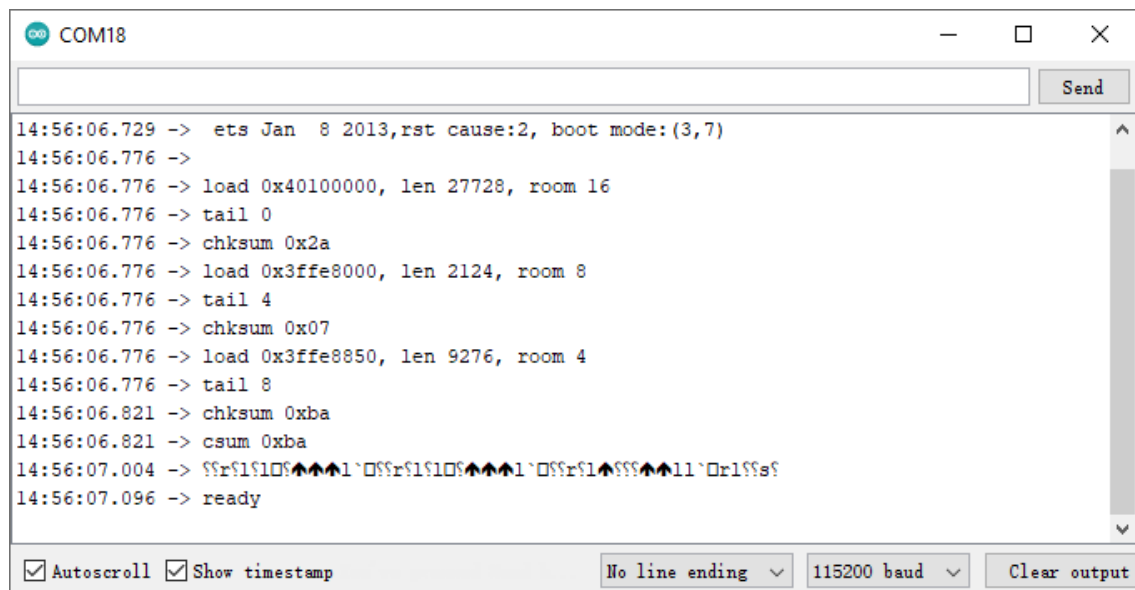
- ESP8266 アダプタの RST を GND に挿入し、それを抜くことで ESP8266 モジュールをリセットします。
- 配線が正しいか確認します。
- コンピュータがボードを正しく認識しているか、そしてポートが占有されていないことを確認します。
- install.bat ファイルを再度開きます。

3. テスト

1. 元の配線の基礎の上で、IO1 を 3V3 に接続します。



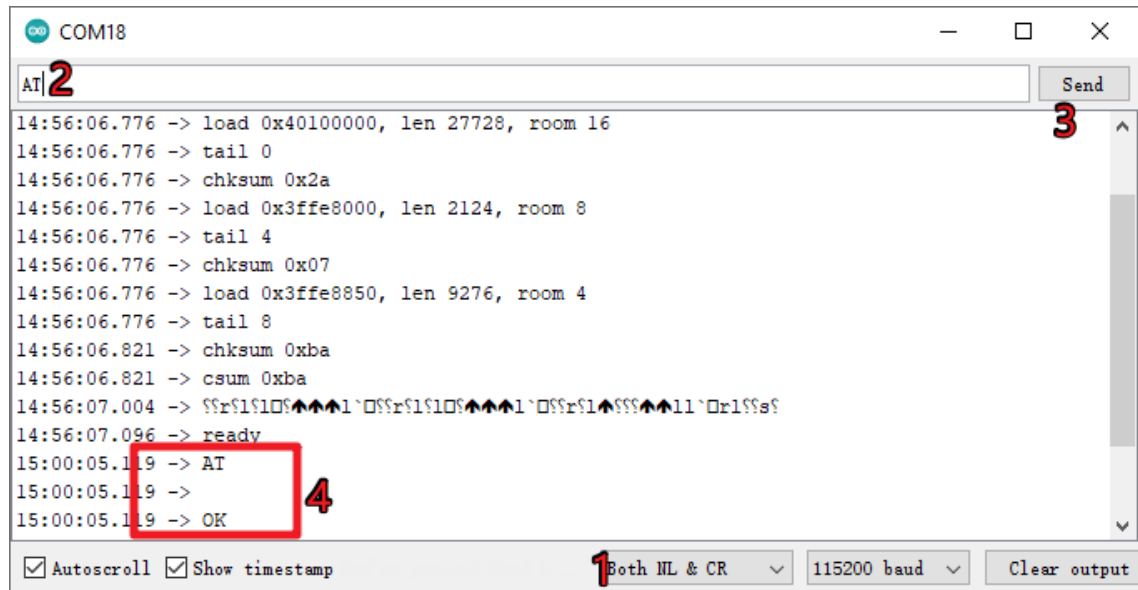
2. 右上隅の虫眼鏡アイコン（シリアルモニタ）をクリックし、ボーレートを **115200** に設定すると、ESP8266 モジュールに関する情報が表示されます。



注釈:

- ready が表示されない場合は、ESP8266 モジュールをリセット（RST を GND に接続）して、シリアルモニタを再度開きます。

3. **NEWLINE DROPDOWN BOX** をクリックし、ドロップダウンオプションで both NL & CR を選択し、AT と入力します。OK が返されれば、ESP8266 が R3 ボードと正常に接続されていることを意味します。



次に、[1.1 ESP8266 の設定](#) に従って、ESP8266 モジュールの動作モードとボーレートを設定できます。

注釈: こんにちは、SunFounder の Raspberry Pi & Arduino & ESP32 愛好家コミュニティへようこそ！ Facebook 上で Raspberry Pi、Arduino、ESP32 についてもっと深く掘り下げ、他の愛好家と交流しましょう。

参加する理由は？

- エキスパートサポート：コミュニティやチームの助けを借りて、販売後の問題や技術的な課題を解決します。
- 学び&共有：ヒントやチュートリアルを交換してスキルを向上させましょう。
- 独占的なプレビュー：新製品の発表や先行プレビューに早期アクセスしましょう。
- 特別割引：最新製品の独占割引をお楽しみください。
- 祭りのプロモーションとギフト：ギフトや祝日のプロモーションに参加しましょう。

私たちと一緒に探索し、創造する準備はできていますか？ [] をクリックして今すぐ参加しましょう！

第 8 章

ありがとうございます

製品を評価してくれた評価者、チュートリアルを提案してくれたベテラン、そして私たちをフォローしサポートしてくれるユーザーの皆様、ありがとうございます。皆様からの貴重な提案は、私たちがより良い製品を提供するための動機となっています。

特別な感謝

- レン・デイビスソン
- カレン・ダニエル
- フアン・デラコスタ

少しだけお時間を割いて、このアンケートに回答していただけますか？

注釈: アンケートを提出した後、結果を表示するためにページの先頭に戻ってください。

第 9 章

著作権通知

このマニュアルに含まれるテキスト、画像、コードなどのすべての内容は SunFounder Company が所有しています。著者と関連する権利者の法的権利を侵害せずに、関連する規制や著作権法の下で、個人的な学習、調査、楽しみ、または他の非営利または非営利目的のためにのみ使用する必要があります。許可なしにこれらを営利目的で使用する個人または組織に対して、会社は法的措置を取る権利を留保しています。