



Elegoo Super Starter Kit for UNO

V1.0.18.7.2

当社

序文

Elegoo は 2011 年に設立され、オープンソースのハードウェアの研究開発、生産、マーケティングに特化し、繁栄したテクノロジー企業です。深セン、中国のシリコンバレーに位置し、我々は 10,763 + 平方フィートの工場で、従業員は 150 人以上です。

当社の製品ラインは、デュポンのワイヤー、UNO R3 ボードから Arduino の知識を学ぶためのあらゆるレベルのお客様向けに設計されたスターターキットを完成させました。また、2.8 インチ TFT タッチと STM32 のようなラズベリーパイアクセサリーの製品も販売しています。将来的には、3D プリンタ製品などに多くのエネルギーと投資を投入します。当社の製品はすべて国際的な品質基準に準拠しており、世界中のさまざまな市場で高く評価されています。

公式サイト: <http://www.elegoo.com>

US Amazon storefront: <http://www.amazon.com/shops/A2WWHQ25ENKVJ1>

CA Amazon storefront: <http://www.amazon.ca/shops/A2WWHQ25ENKVJ1>

UK Amazon storefront: <http://www.amazon.co.uk/shops/AZF7WYXU5ZANW>

DE Amazon storefront: <http://www.amazon.de/shops/AZF7WYXU5ZANW>

FR Amazon storefront: <http://www.amazon.fr/shops/AZF7WYXU5ZANW>

ES Amazon storefront: <http://www.amazon.es/shops/AZF7WYXU5ZANW>

IT Amazon storefront: <http://www.amazon.it/shops/AZF7WYXU5ZANW>

JP Amazon storefront: <http://www.amazon.co.jp/shops/A21X7DQBM2LL85>

この教材

この教材は初心者のためのものです。Arduino コントローラボード、センサー、およびコンポーネントの使用方法に関する基本情報をすべて学びます。Arduino をさらに深く勉強したい場合は、Michael Margolis が書いた Arduino Cookbook を読むことをお勧めします。このチュートリアルの一部のコードは Simon Monk によって編集されています。Simon Monk はオープンソースハードウェアに関する数多くの書籍の著者です。

Amazon で利用可能です: Arduino のプログラミング、Evil Genius のための 30 の Arduino プロジェクト、Raspberry Pi のプログラミング。

顧客サービス

絶えず急速に成長しているテクノロジー企業として、私たちはお客様の期待に応える優れた製品とた質の高いサービスをご提供しており、service@elegoo.com または EUservice@elegoo.com までご連絡ください。皆様からのご意見をお待ちしております。批判的なご意見やご提案は、私たちにとって大変貴重なものとなります。

また、製品に関する問題やご質問は、経験豊かなエンジニアが 12 時間以内に (24 時間休暇中) 迅速に返信します。

Packing list

 www.elegoo.com

Stepper Motor
1PC



Servo Motor
(SG90)
1PC



IR Receiver
Module
1PC



5V Relay
1PC



Uno R3
Controller
Board
1PC



LCD 1602 Module
(with pin header)
1PC



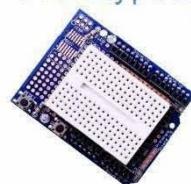
ULN2003 Stepper
Motor Driver
Module
1PC



Power Supply
Module
1PC



Prototype Expansion
Module
1PC





Fan Blade and
3-6V DC
Motor
(with wire)
each
1PC



Ultrasonic Sensor
1PC



Button (Small)
5PCS



Potentiometer
1PC



Passive Buzzer
1PC



9V Battery with
Snap-on Connector
Clip 1PC



Active Buzzer
1PC



Remote
1PC



USB Cable
1PC



Female-to-male
Dupont wire
10PCS



Breadboard Jumper
Wire 65PCS



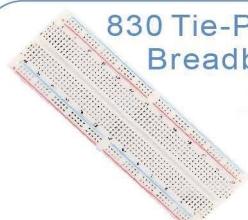
74HC595 IC
1PC



L293D
1PC



830 Tie-Points
Breadboard
1PC



Contact us : service@elegoo.com

Thermistor
1PC



Diode Rectifier
2PCS



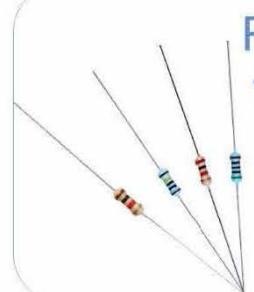
Photoresistor
(Photocell)
2PCS



LED
25PCS



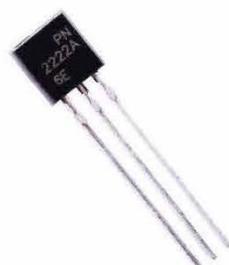
Resistor
120PCS



RGB
2PCS



NPN Transistor
PN2222
2PCS



Content

ディレクトリ

Lesson 0 IDE のインストール.....	15
Lesson 1 ライブラリを追加してシリアルモニタを開く	23
Lesson 2 点滅.....	32
Lesson 3 LED	43
Lesson 4 RGB LED	50
Lesson 5 デジタル入力.....	59
Lesson 6 アクティブプザー	64
Lesson 7 受動プザー	68
Lesson 8 傾斜ボールスイッチ.....	72
Lesson 9 サーボ.....	76
Lesson 10 超音波センサモジュール	80
Lesson 11 DHT11 温度湿度センサー.....	85
Lesson 12 アナログジョイスティックモジュール.....	91
Lesson 13 赤外線受信モジュール	102
Lesson 14 LCD Display	108
Lesson 15 Thermometer	113
Lesson 16 Eight LED with 74HC595	162
Lesson 17 シリアルモニタ	168
Lesson 18 光電池.....	175
Lesson 19 74HC595 と 7セグメント表示	180

Lesson 20 4つのデジタル7セグメント表示.....	186
Lesson 21 直流モータ	191
Lesson 22 リレー	200
Lesson 23 ステッパーモーター.....	209
Lesson 24 リモートでステッピングモータを制御する	214

Lesson 0 IDE のインストール

前書き

Arduino 統合開発環境 (IDE) は、Arduino プラットフォームのソフトウェア側です。

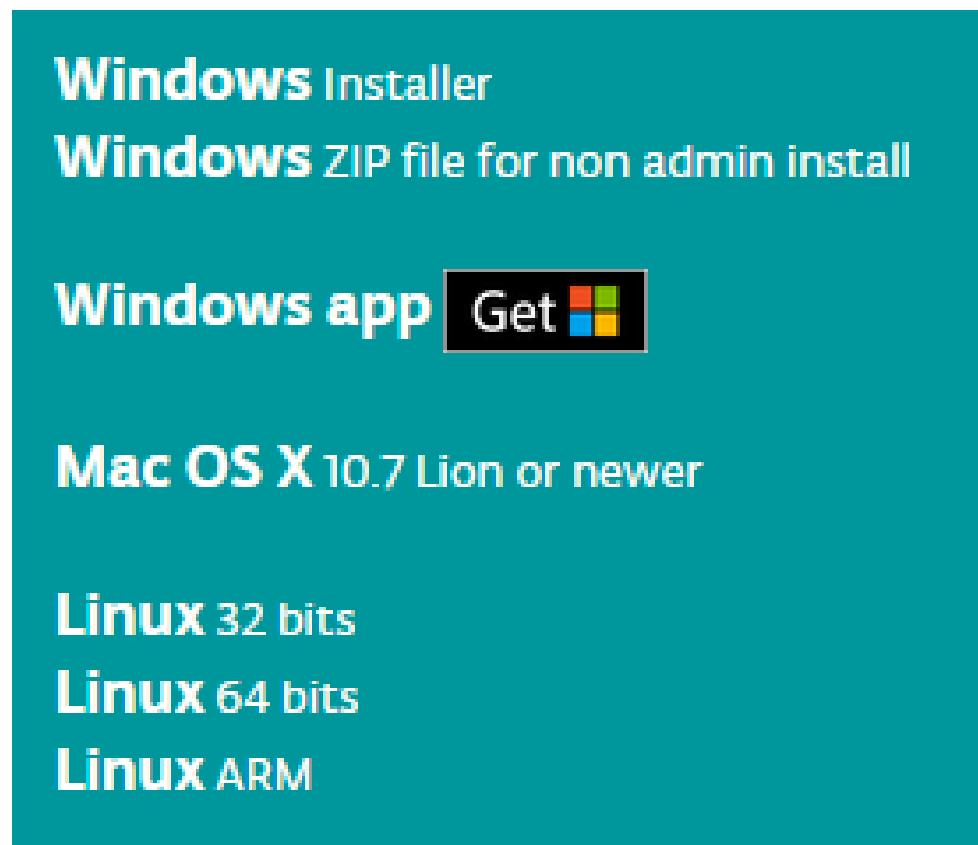
このレッスンでは、Arduino を使用するようにコンピュータを設定する方法と、それに続くレッスンについて設定する方法を学習します。

Arduino をプログラミングするために使用する Arduino ソフトウェアは、Windows、Mac、Linux で使用できます。 インストールプロセスは 3 つのプラットフォームすべてで異なります。 残念ながら、ソフトウェアをインストールするには一定の作業が必要です。

STEP 1: Go to <https://www.arduino.cc/en/Main/Software> and find below page.

このウェブサイトで利用可能なバージョンは通常最新バージョンであり、実際のバージョンは画像のバージョンよりも新しい場合があります。

STEP2: コンピュータのオペレーティングシステムと互換性のある開発ソフトウェアをダウンロードします。[ここで Windows を例に取る。](#)



Click *Windows Installer*.

Support the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.



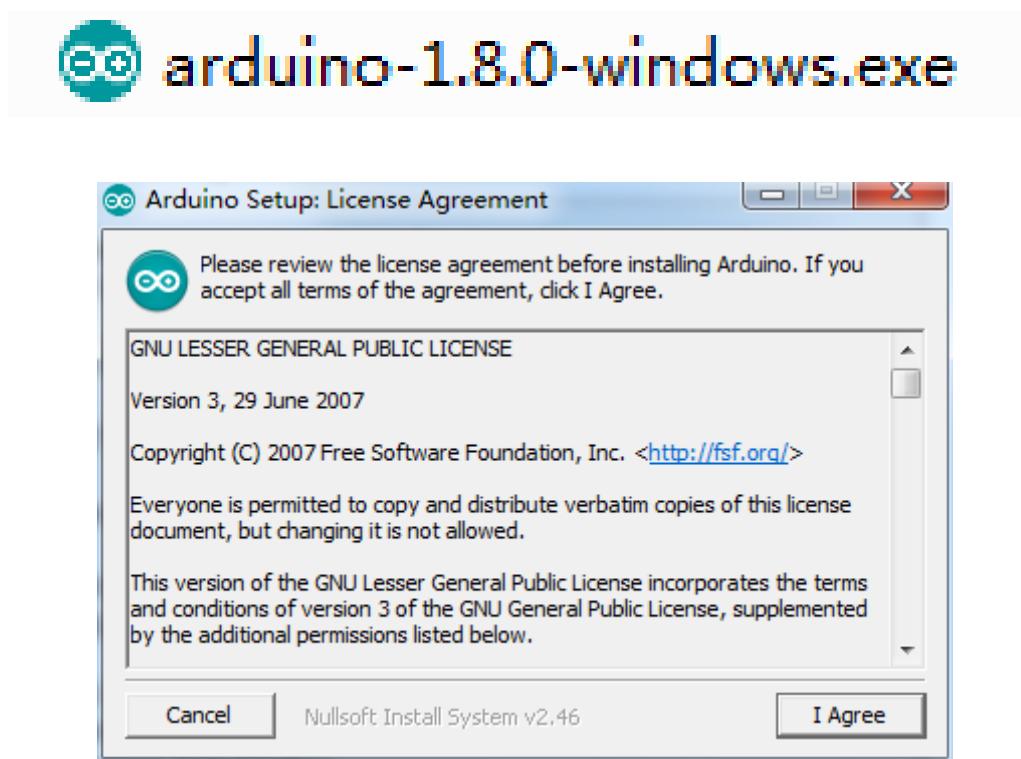
Click *JUST DOWNLOAD*.

私たちが提供した資料でもバージョン 1.8.0 が利用可能で、この資料のバージョンはこのコースが作成されたときの最新バージョンです。

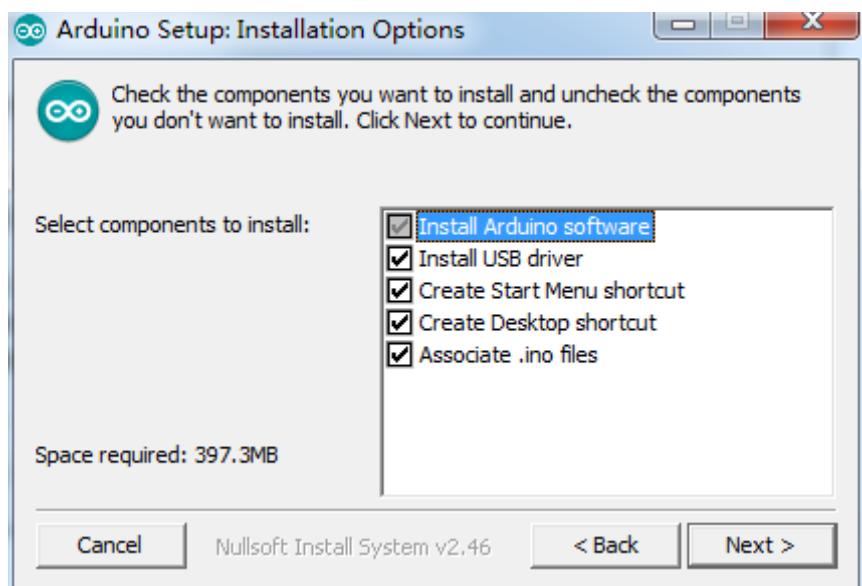
-  arduino-1.8.0-linux32.tar.xz
-  arduino-1.8.0-linux64.tar.xz
-  arduino-1.8.0-macosx.zip
-  arduino-1.8.0-windows.exe
-  arduino-1.8.0-windows.zip

Installing Arduino (Windows)

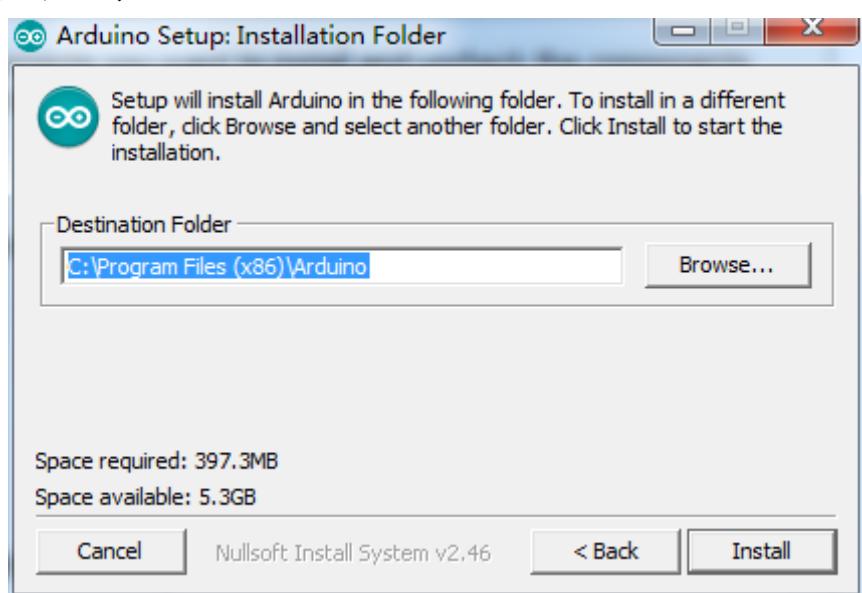
xe で Arduino をインストールします。 インストールパッケージ。



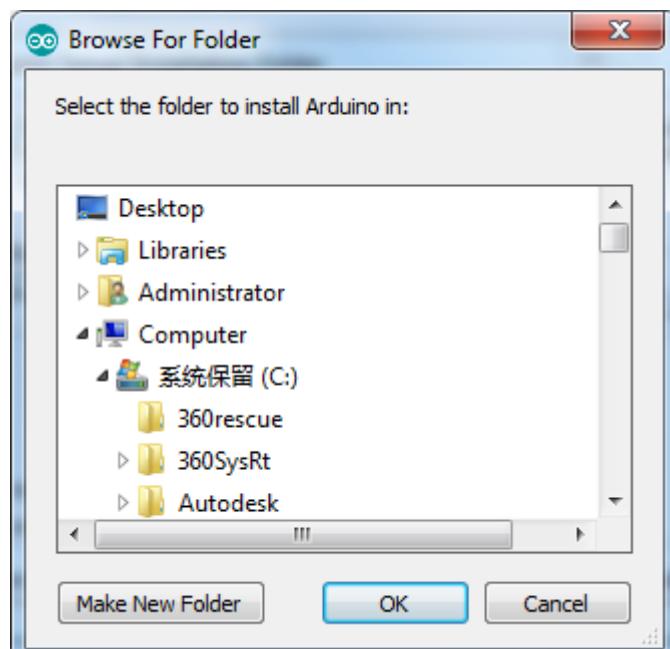
「同意する」をクリックすると、次のインターフェースが表示されます。



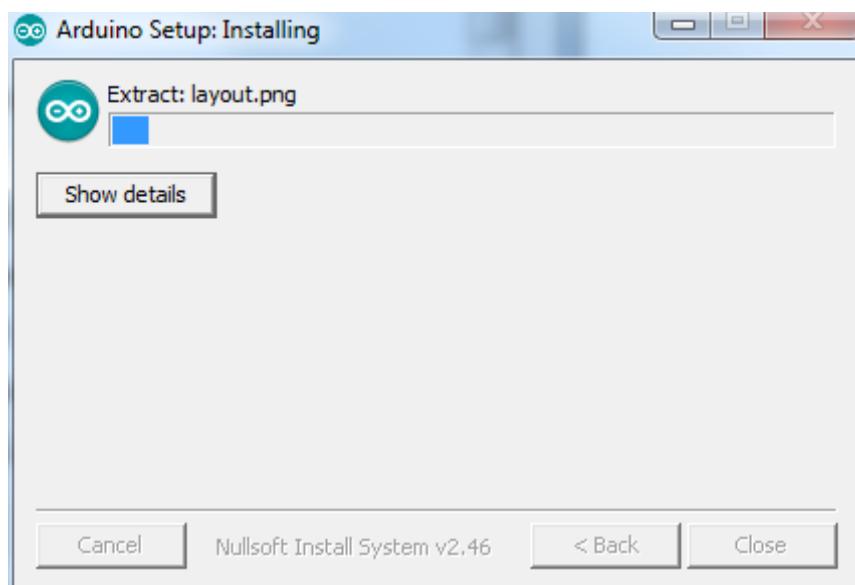
次へをクリックします



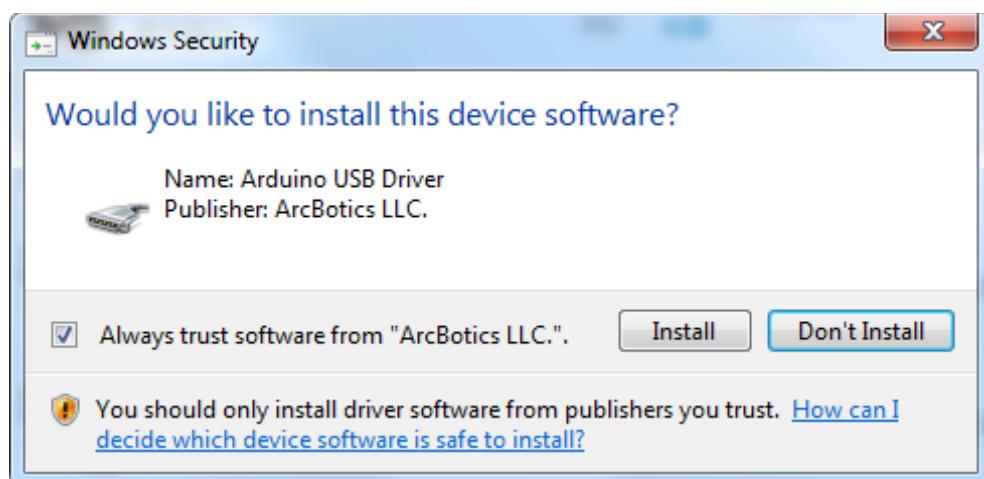
[参照]を押してインストールパスを選択するか、目的のディレクトリに直接入力することができます。



[インストール]をクリックしてインストールを開始します



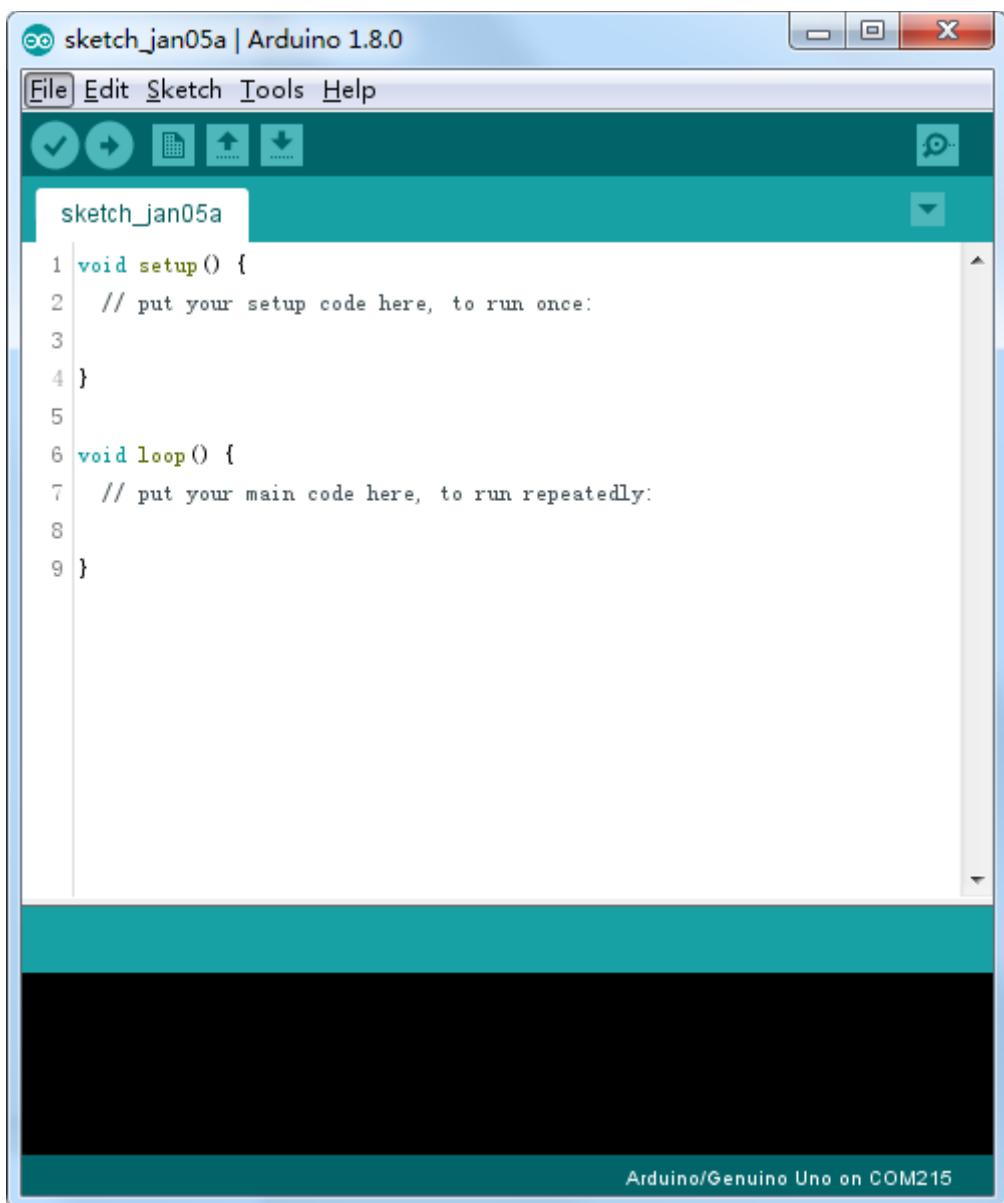
最後に、次のインターフェイスが表示されたら、[インストール]をクリックしてインストールを完了します。



次に、デスクトップに次のアイコンが表示されます。



ダブルクリックして希望の開発環境に入ります



インストール用のインストールパッケージを直接選択して下の内容をスキップし、次のセクションにジャンプすることができます。しかし、インストールパッケージ以外のいくつかの方法を学びたい場合は、引き続きこのセクションをお読みください。

ダウンロードした zip ファイルを解凍し、ダブルクリックしてプログラムを開き、希望する開発環境に入る。



Name	Date modified	Type	Size
drivers	2017/1/5 14:29	File folder	
examples	2017/1/5 14:29	File folder	
hardware	2017/1/5 14:29	File folder	
java	2017/1/5 14:29	File folder	
lib	2017/1/5 14:29	File folder	
libraries	2017/1/5 14:29	File folder	
reference	2017/1/5 14:29	File folder	
tools	2017/1/5 14:29	File folder	
tools-builder	2017/1/5 14:29	File folder	
arduino.exe	2016/12/21 17:16	Application	395 KB
arduino.l4j.ini	2016/12/21 17:16	Configuration sett...	1 KB
arduino_debug.exe	2016/12/21 17:16	Application	392 KB
arduino_debug.l4j.ini	2016/12/21 17:16	Configuration sett...	1 KB
arduino-builder.exe	2016/12/21 17:15	Application	3,192 KB
libusb0.dll	2016/12/21 17:14	Application extens...	43 KB
msvcp100.dll	2016/12/21 17:15	Application extens...	412 KB
msvcr100.dll	2016/12/21 17:15	Application extens...	753 KB
revisions.txt	2016/12/21 17:14	Text Document	80 KB
wrapper-manifest.xml	2016/12/21 17:16	XML Document	1 KB

19 items



sketch_jan05a | Arduino 1.8.0

File Edit Sketch Tools Help

sketch_jan05a

```

1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }

```

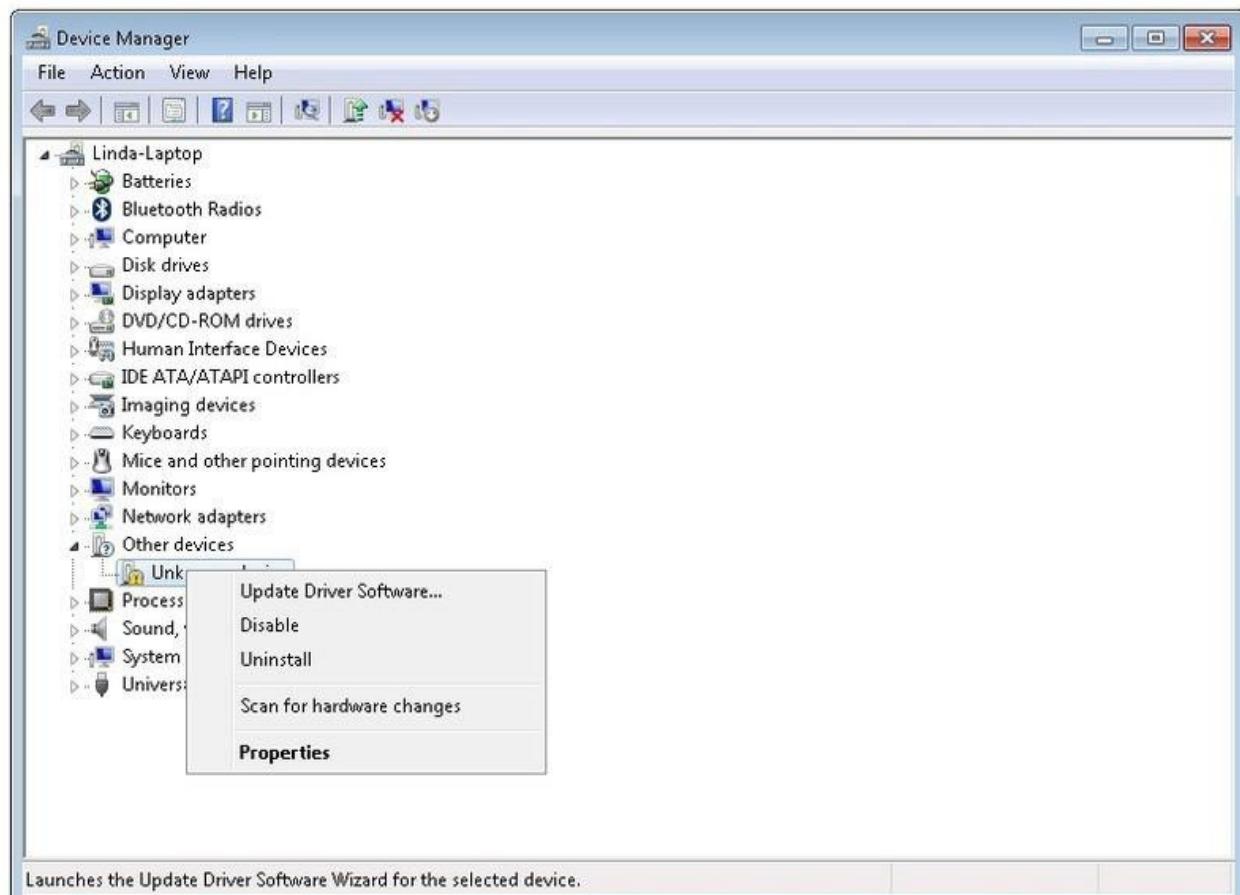
Arduino Nano, ATmega328 on COM239

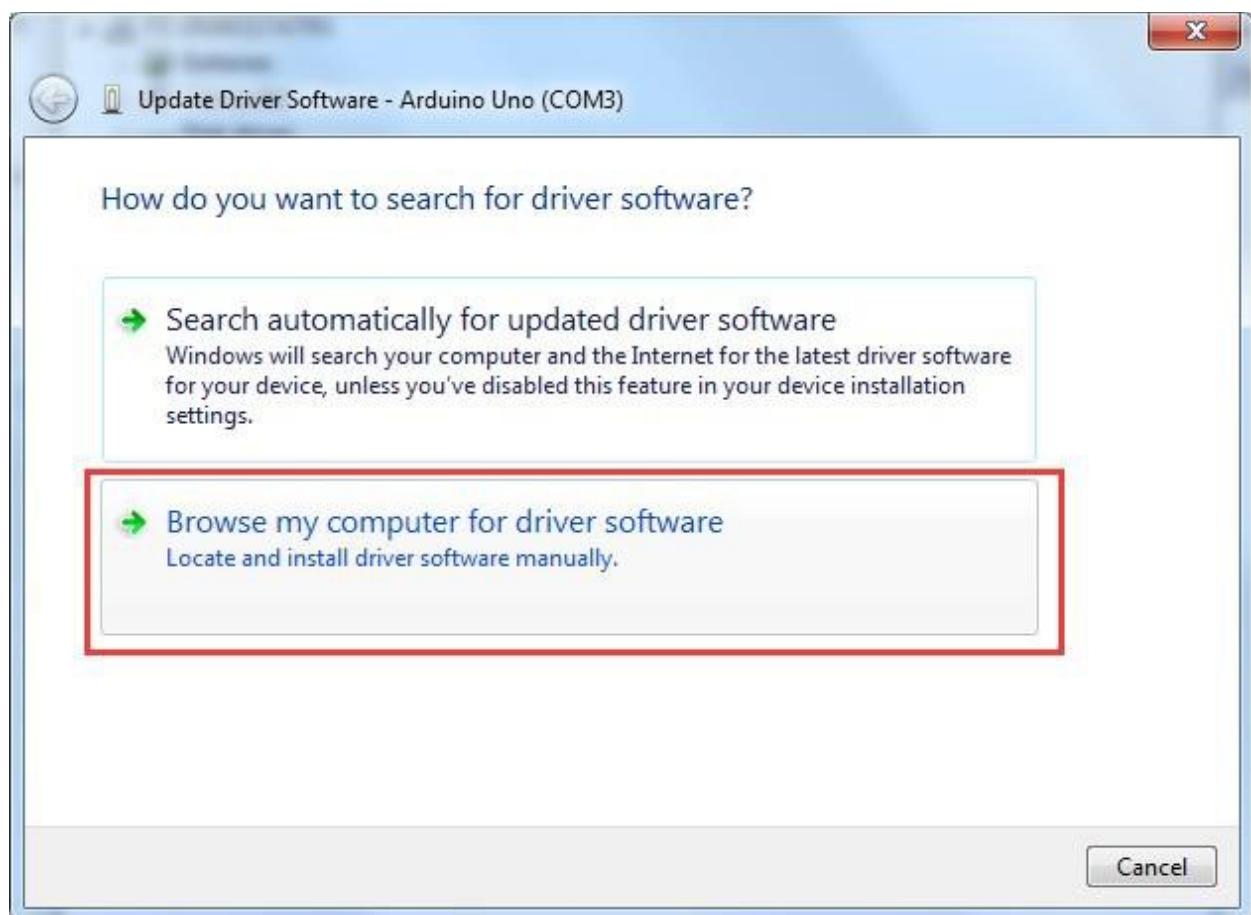
ただし、このインストール方法ではドライバを別途インストールする必要があります。

Arduino フォルダには、Arduino プログラム自体と、Arduino をコンピュータに USB ケーブルで接続するためのドライバが含まれています。 Arduino ソフトウェアを起動する前に、USB ドライバをインストールします。

USB ケーブルの一方の端を Arduino に差し込み、もう一方をコンピュータの USB ソケットに差し込みます。 LED の電源ランプが点灯し、Windows から「新しいハードウェアの発見」メッセージが表示される場合があります。 このメッセージを無視し、Windows がドライバを自動的にインストールしようとする試行をすべてキャンセルします。 USB ドライバをインストールする最も信頼性の高い方法は、デバイスマネージャを使用することです。 これは、Windows のバージョンによって異なる方法でアクセスされます。 Windows 7 では、最初にコントロールパネルを開き、アイコンを表示するオプションを選択する必要があります。リストでデバイスマネージャを見つけてください。

「その他のデバイス」の下に、「未知のデバイス」のアイコンが表示され、その隣に少し黄色の警告三角が表示されます。 これはあなたの Arduino です。





デバイスを右クリックし、トップメニュー「オプション」(Update Driver Software ...) を選択します。

「更新されたドライバソフトウェアを自動的に検索する」または「コンピュータでドライバソフトウェアを参照する」のいずれかのプロンプトが表示されます。 X \ arduino1.8.0 \ drivers をブラウズしてナビゲートするオプションを選択します。



[次へ]をクリックすると、セキュリティ警告が表示されます。インストールされている場合は、ソフトウェアのインストールが許可されます。ソフトウェアがインストールされると、確認メッセージが表示されます。



Windows ユーザーは、Mac および Linux システムのインストール手順をスキップして、レッスン 1 にジャンプすることができます。Mac および Linux のユーザーは引き続きこのセクションを読むことができます。

Installing Arduino (Mac OS X)

Zip ファイルをダウンロードして解凍し、Arduino.app をダブルクリックして Arduino IDE に入ります。コンピュータに Java ランタイムライブラリがインストールされていない場合は、Java ランタイムライブラリをインストールするように求められます。インストールが完了したら、Arduino IDE を実行できます。



Installing Arduino (Linux)

make install コマンドを使用する必要があります。 Ubuntu システムを使用している場合は、Ubuntu のソフトウェアセンターから Arduino IDE をインストールすることをお勧めします。

 [arduino-1.8.0-linux32.tar.xz](#)

 [arduino-1.8.0-linux64.tar.xz](#)

ヒント： ドライバのインストールに問題がある場合は、[UNO R3, MEGA, NANO DRIVER FAQ](#) を参照してください。



[UNO R3, MEGA, NANO DRIVER FAQ](#)

Lesson 1 ライブラリを追加してシリアルモニタを開く

追加 Arduino ライブラリのインストール

Arduino ソフトウェアに慣れていて、組み込み関数を使用すると、Arduino の機能を追加のライブラリで拡張することができます。

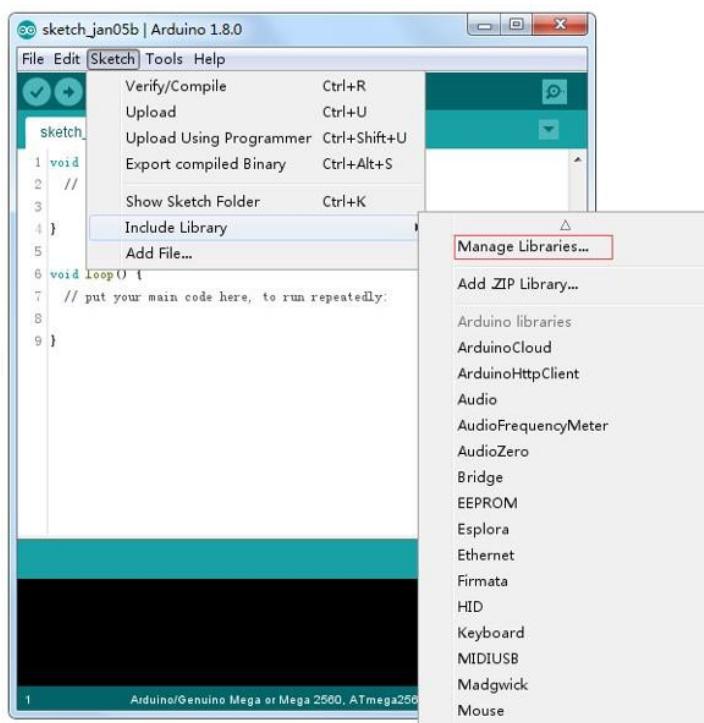
What are Libraries?

ライブラリは、センサ、ディスプレイ、モジュールなどに簡単に接続できるコードの集まりです。たとえば、内蔵の LiquidCrystal ライブラリを使用すると、キャラクタの LCD ディスプレイと簡単に会話することができます。ダウンロードのためにインターネット上で利用可能な数百の追加ライブラリがあります。組み込みライブラリとこれらの追加ライブラリの一部は、参考文献にリストされています。追加のライブラリを使用するには、それらをインストールする必要があります。

How to Install a Library

ライブラリマネージャの使用

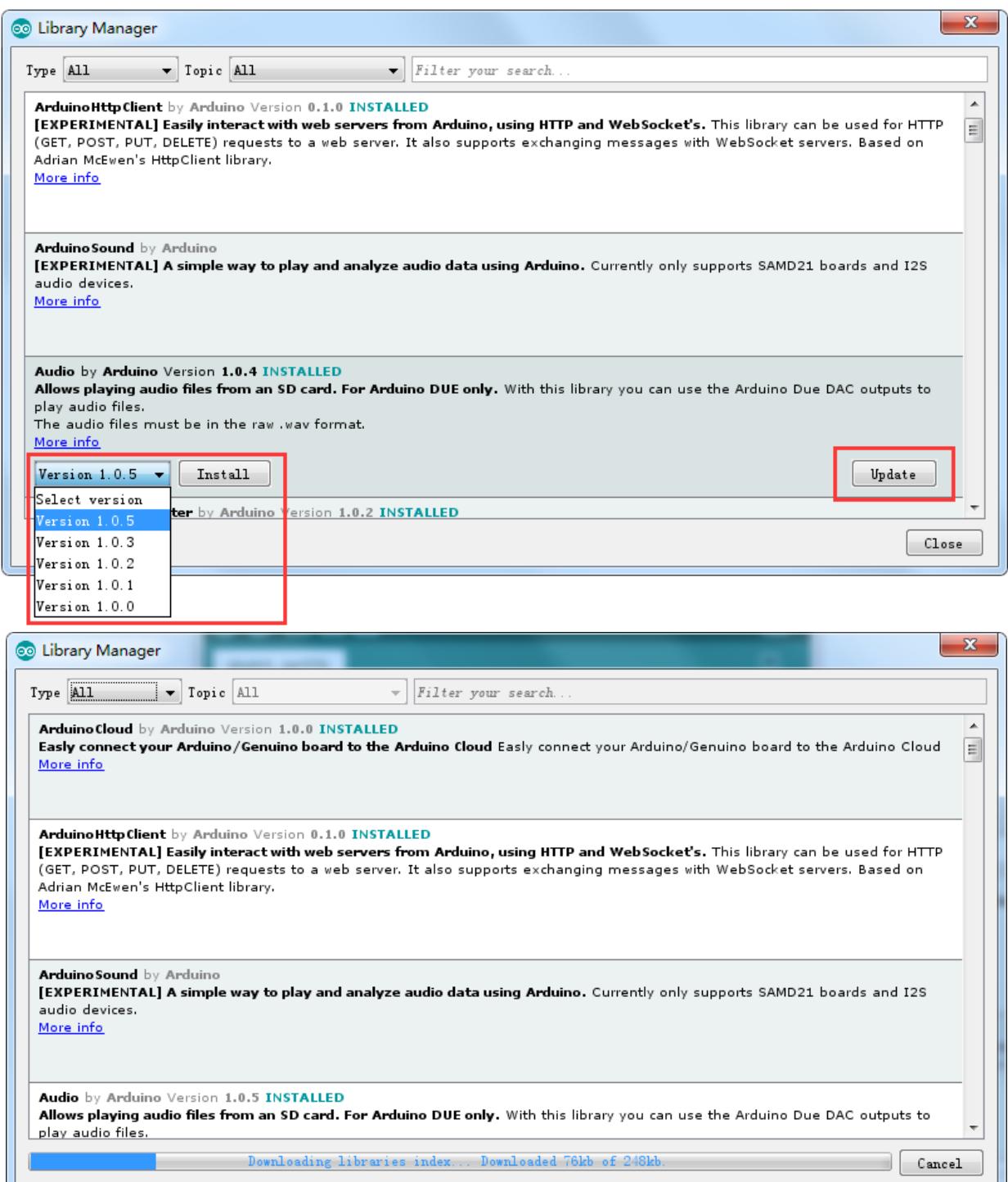
Arduino IDE に新しいライブラリをインストールするには、ライブラリマネージャ（IDE バージョン 1.8.0 から入手可能）を使用できます。IDE を開き、[スケッチ]メニューをクリックし、[ライブラリを含む]> [ライブラリを管理]をクリックします。



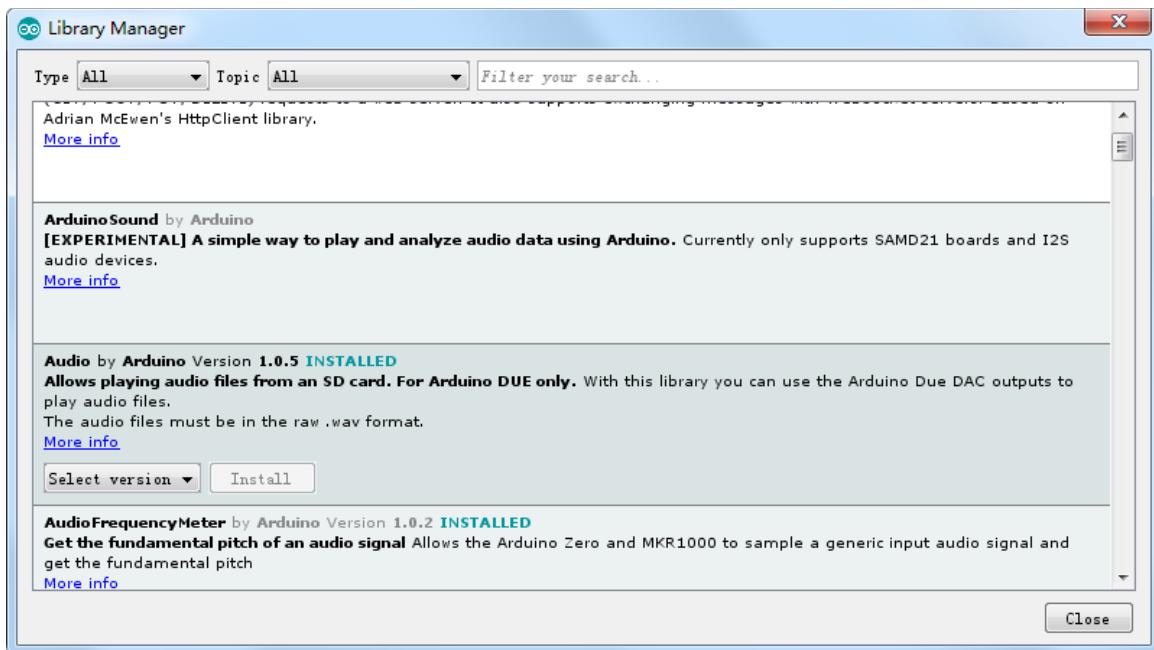
次に、ライブラリマネージャが開き、既にインストールされているか、インストールの準備が整っているライブラリのリストが表示されます。この例では、Bridge ライブラリをインストールします。

リストをスクロールして検索し、インストールするライブラリのバージョンを選択します。ライブラリの 1 つのバージョンしか利用できないことがあります。バージョン選択メニューが表示されない場合は、これは正常です。

忍耐しなければならない時があり、図に示されるように。それをリフレッシュしてそして待ってください。



最後に、install をクリックし、IDE が新しいライブラリをインストールするのを待ちます。ダウンロードには接続速度に応じて時間がかかることがあります。インストールが完了したら、Installed タグが Bridge ライブラリの横に表示されます。ライブラリマネージャを閉じることができます。

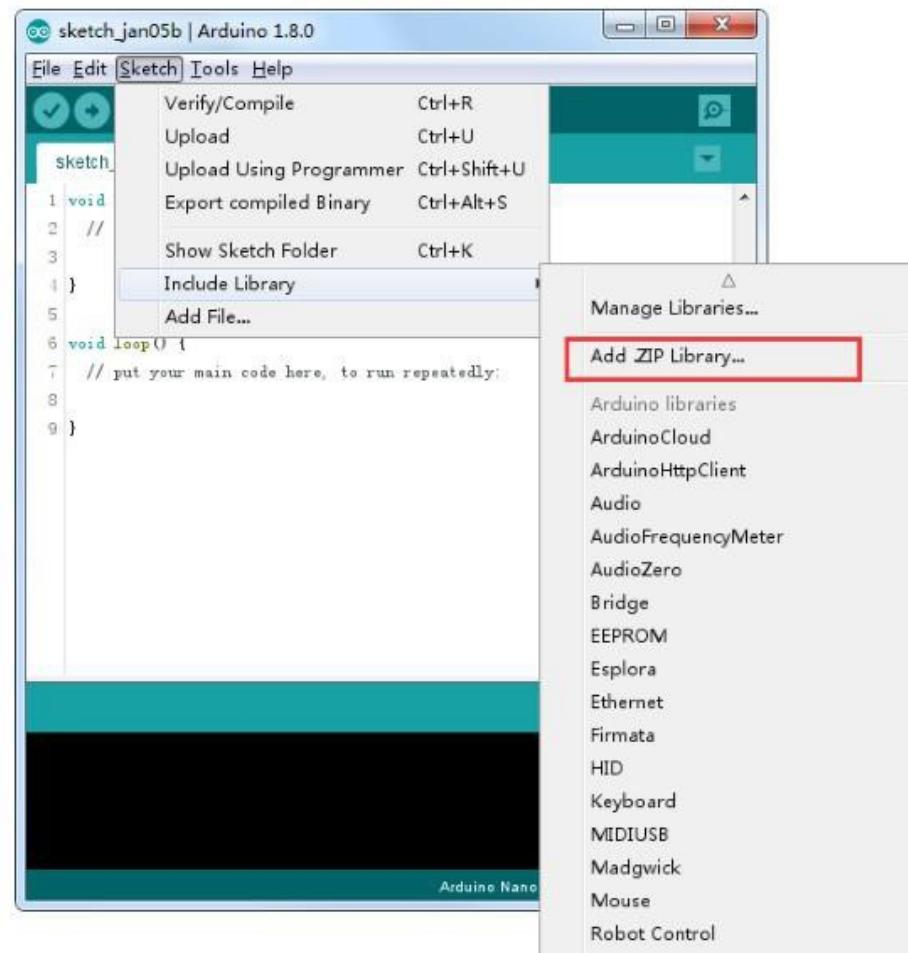


新しいライブラリは[ライブラリを含む]メニューで使用できるようになりました。独自のライブラリを追加したい場合は、Github で新しい問題を開きます。

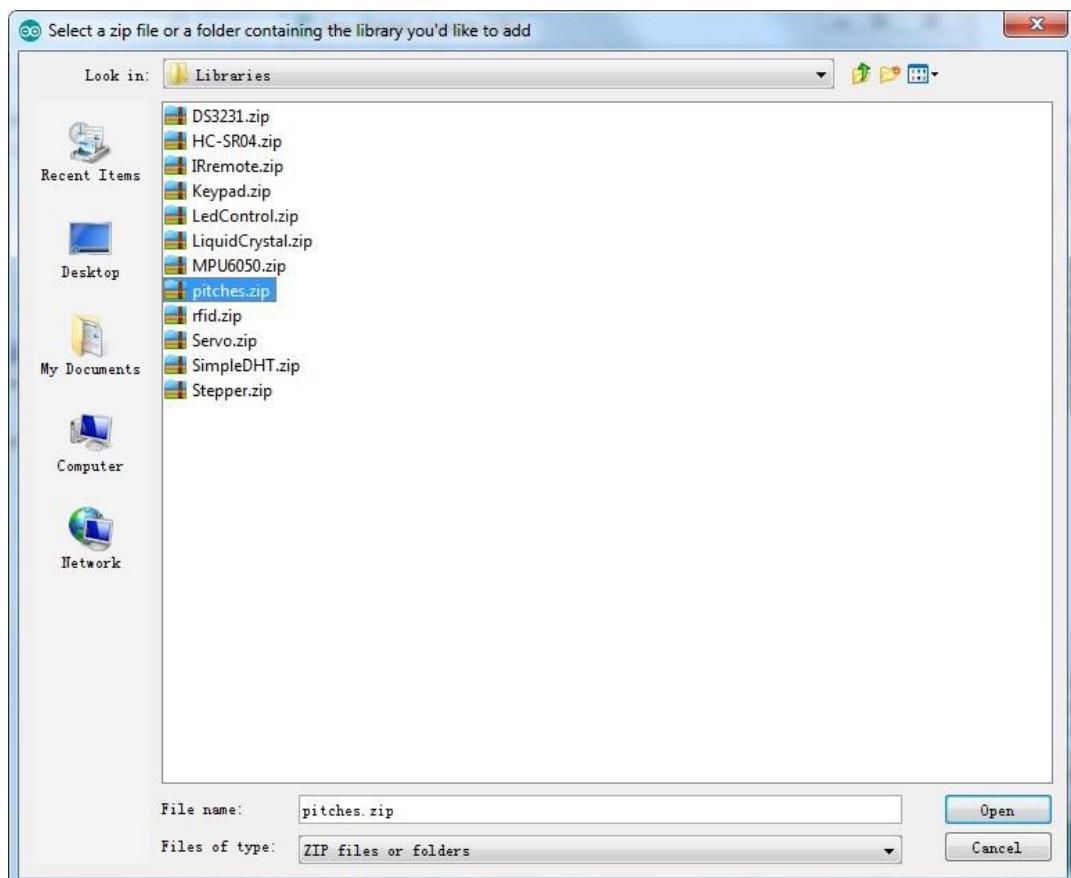
.zip ライブラリのインポート

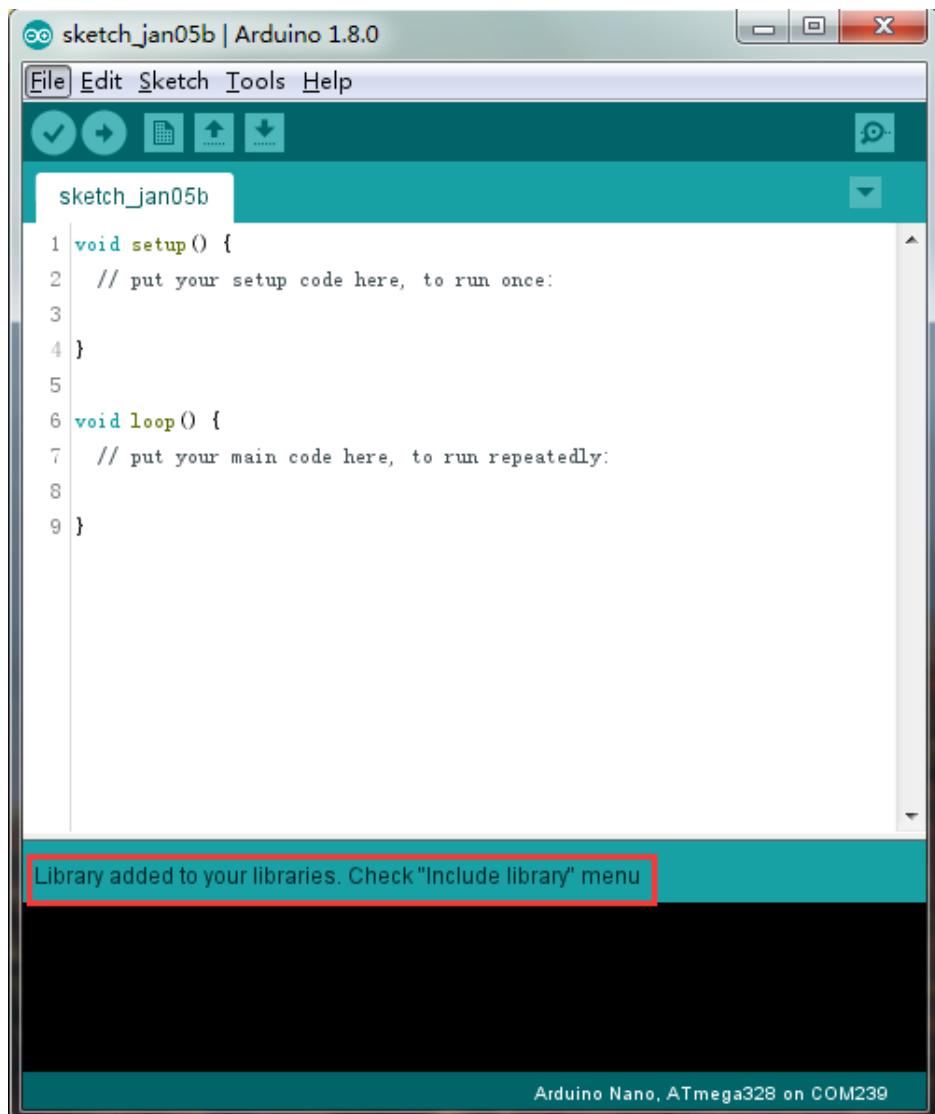
ライブラリは、多くの場合、ZIP ファイルまたはフォルダとして配布されます。フォルダの名前はライブラリの名前です。フォルダ内には、.cpp ファイル、.h ファイル、および keywords.txt ファイル、examples フォルダ、およびライブラリに必要なその他のファイルが含まれます。バージョン 1.0.5 から、サードパーティのライブラリを IDE にインストールすることができます。ダウンロードしたライブラリを解凍しないでそのままにしてください。

Arduino IDE で、[スケッチ]> [ライブラリを含める]に移動します。ドロップダウンリストの一番上にある[ZIP ライブラリを追加]オプションを選択します。



追加したいライブラリを選択し、.zip ファイルの場所に移動して開きます。





スケッチ] > [ライブラリの読み込み]メニューに戻ります。 ドロップダウンメニューの一番下にライブラリが表示されます。 あなたのスケッチで使用する準備が整いました。 zip ファイルは Arduino スケッチディレクトリの libraries フォルダに展開されます。

注: ライブラリはスケッチで使用できるようになりますが、IDE の再起動後までライブラリの例は File > Examples で公開されません。

これらの 2 つが最も一般的なアプローチです。 MAC および Linux システムも同様に処理できます。 下記で紹介するマニュアルインストールはほとんど使用されないかもしれませんし、必要のないユーザーはそれをスキップするかもしれません。

手動インストール

ライブラリをインストールするには、まず Arduino アプリケーションを終了します。 その後、ライブラリを含む ZIP ファイルを解凍します。 たとえば、「ArduinoParty」というライブラリをインストールする場合は、ArduinoParty.zip を解凍します。

ArduinoParty.cpp や ArduinoParty.h のようなファイルが入った ArduinoParty という名前のフォルダがあります。 (.cpp ファイルと.h ファイルがフォルダ内にない場合は、ファイルを作成する必要があります。この場合、 "ArduinoParty" というフォルダを作成し、ZIP 内にあるすべてのファイルに移動します ArduinoParty.cpp や ArduinoParty.h のようなファイルです。)

ArduinoParty フォルダをこのフォルダ (ライブラリフォルダ) にドラッグします。 Windows では、おそらく "My Documents \ Arduino \ libraries" です。

Mac ユーザーにとっては、おそらく "Documents / Arduino / libraries" です。 Linux では、スケッチブックの "libraries" フォルダになります。

あなたの Arduino ライブラリのフォルダは次のようにになります (Windows) :

My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.cpp
My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.h My
Documents\Arduino\libraries\ArduinoParty\examples

(Mac と Linux):

Documents/Arduino/libraries/ArduinoParty/ArduinoParty.cpp
Documents/Arduino/libraries/ArduinoParty/ArduinoParty.h
Documents/Arduino/libraries/ArduinoParty/examples

....

cpp ファイルと.h ファイルだけでなく、もっと多くのファイルがあるかもしれません。すべてがそこにあることを確認してください。

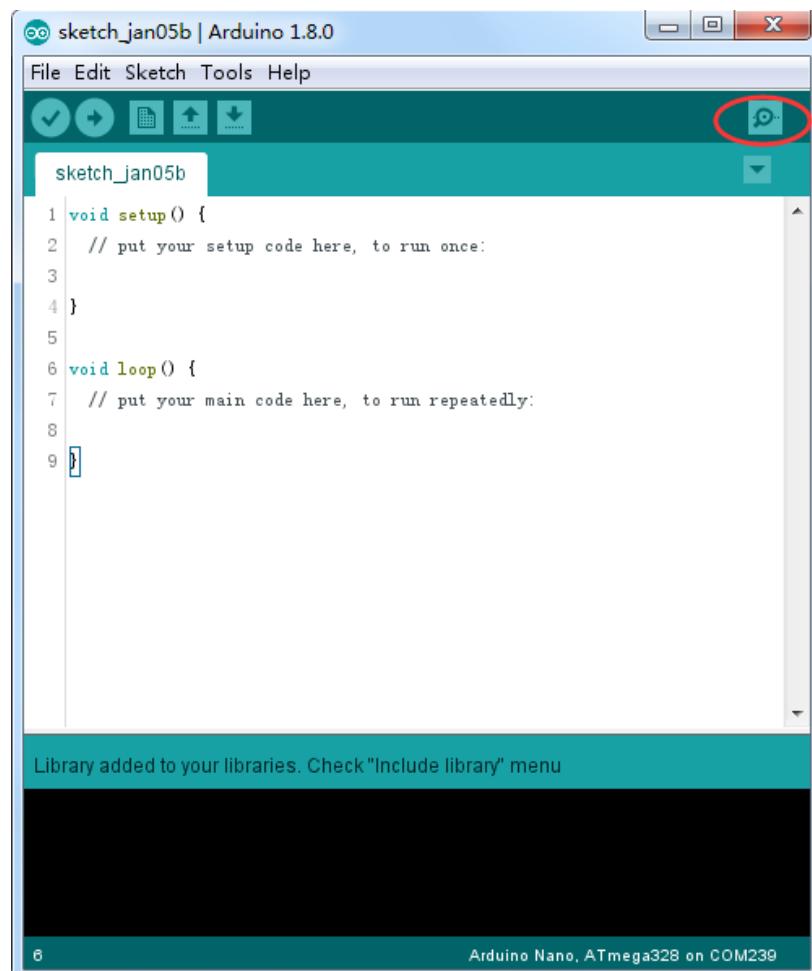
(.cpp ファイルと.h ファイルをライブラリフォルダに直接置くか、余分なフォルダに入れた場合、ライブラリは機能しません。例えば： Documents¥Arduino¥libraries¥ArduinoParty.cpp and Documents¥Arduino¥libraries¥ArduinoParty¥ArduinoParty¥ArduinoParty.cpp won't work.)
Arduino アプリケーションを再起動します。新しいライブラリが Sketch-> [ライブラリのインポート] メニュー項目を選択します。それでおしまい！ ライブラリをインストールしました！

Arduino Serial Monitor (Windows, Mac, Linux)

Arduino 統合開発環境 (IDE) は、Arduino プラットフォームのソフトウェア側です。そして、端末を使用することは、Arduinos と他のマイクロコントローラ、彼らはソフトウェアとシリアル端子を含めることにしました。 Arduino 環境では、これはシリアルモニタと呼ばれます。

接続

シリアルモニタには、Arduino IDE のすべてのバージョンが付属しています。それを開くには、シリアルモニタアイコンをクリックするだけです。



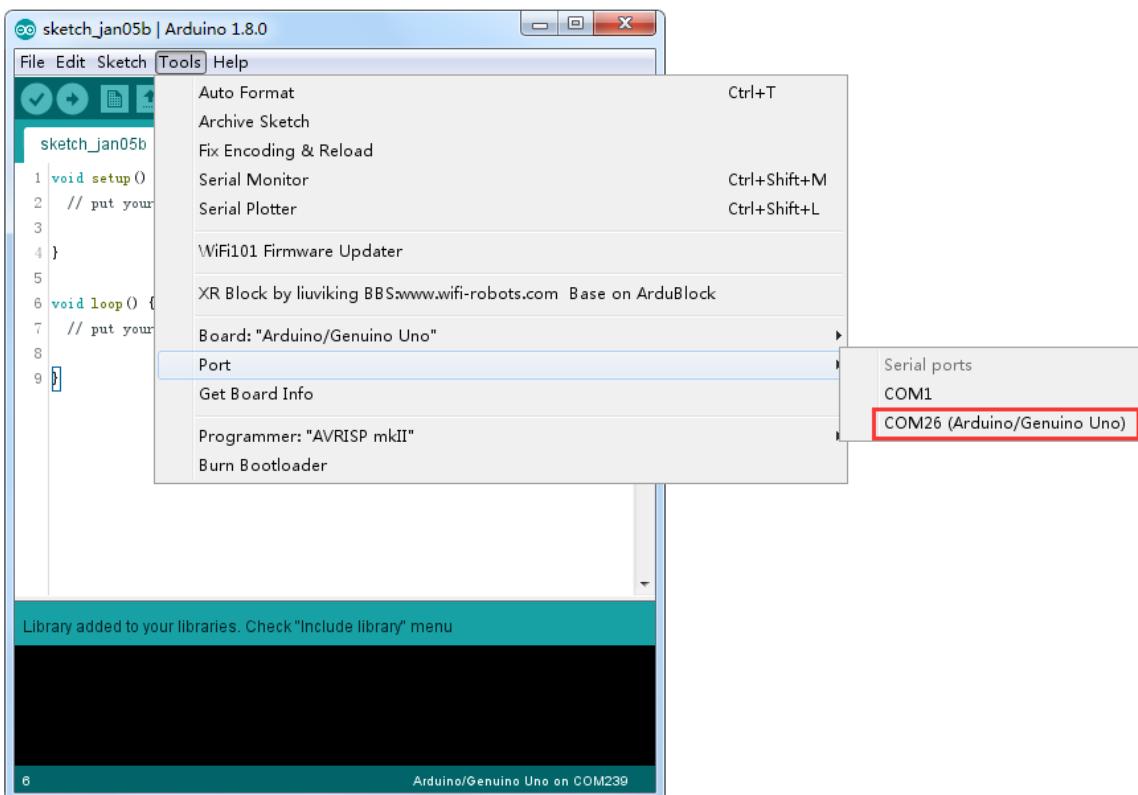
```
1 void setup() {  
2     // put your setup code here, to run once:  
3  
4 }  
  
5  
6 void loop() {  
7     // put your main code here, to run repeatedly:  
8  
9 }
```

Library added to your libraries. Check "Include library" menu

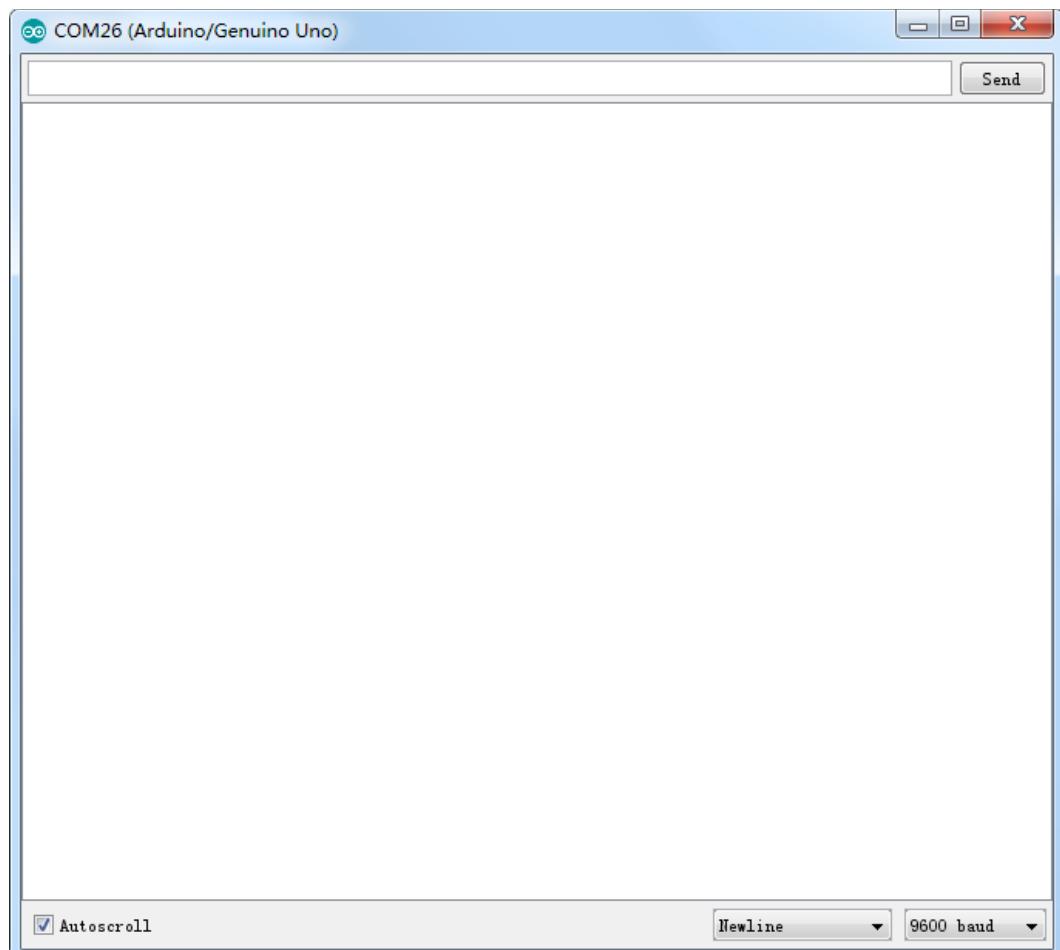
Arduino Nano, ATmega328 on COM239

シリアルモニタで開くポートを選択することは、Arduino コードをアップロードするポートを選択することと同じです。 「ツール」 -> 「シリアルポート」に移動し、正しいポートを選択します。

ヒント：デバイスマネージャーと同じ COM ポートを選択してください。

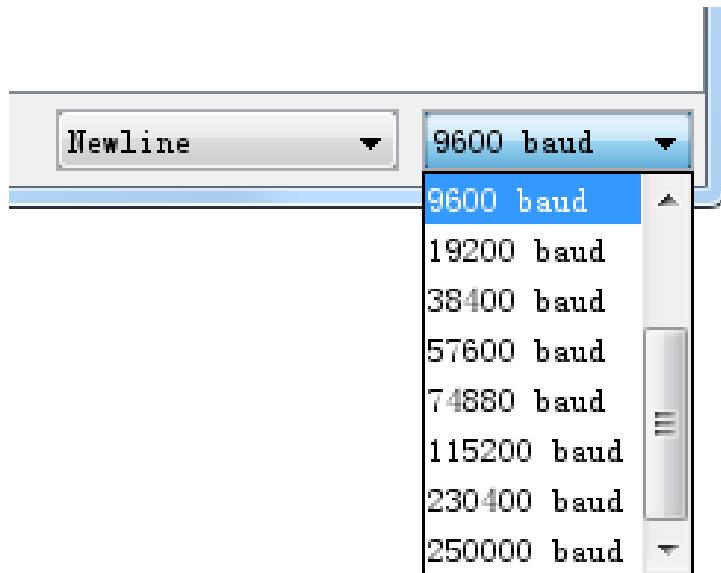


開いたら、次のように表示されます:

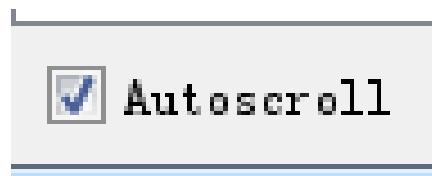


設定

アルモニタには設定が限られていますが、ほとんどのシリアル通信のニーズに対応できます。変更可能な最初の設定はボーレートです。ボーレートドロップダウンメニューをクリックして正しいボーレートを選択します。 (9600 ポー)



最後に、左下隅のチェックボックスをオンにすることで、端末を自動スクロールするかどうかを設定できます。



長所

シリアルモニタは Arduino とのシリアル接続を確立するのに素早く簡単な方法です。既に Arduino IDE で作業している場合は、別の端末を開いてデータを表示する必要はありません。

短所

シリアルモニタでは設定の欠如が望ましく、高度なシリアル通信の場合は、このトリックはできません。

Lesson 2 点滅

概要

このレッスンでは、Arduino の内蔵 LED を点滅させられる UNO R3 コントロール方法と、プログラムをダウンロードする手順を学習します。。

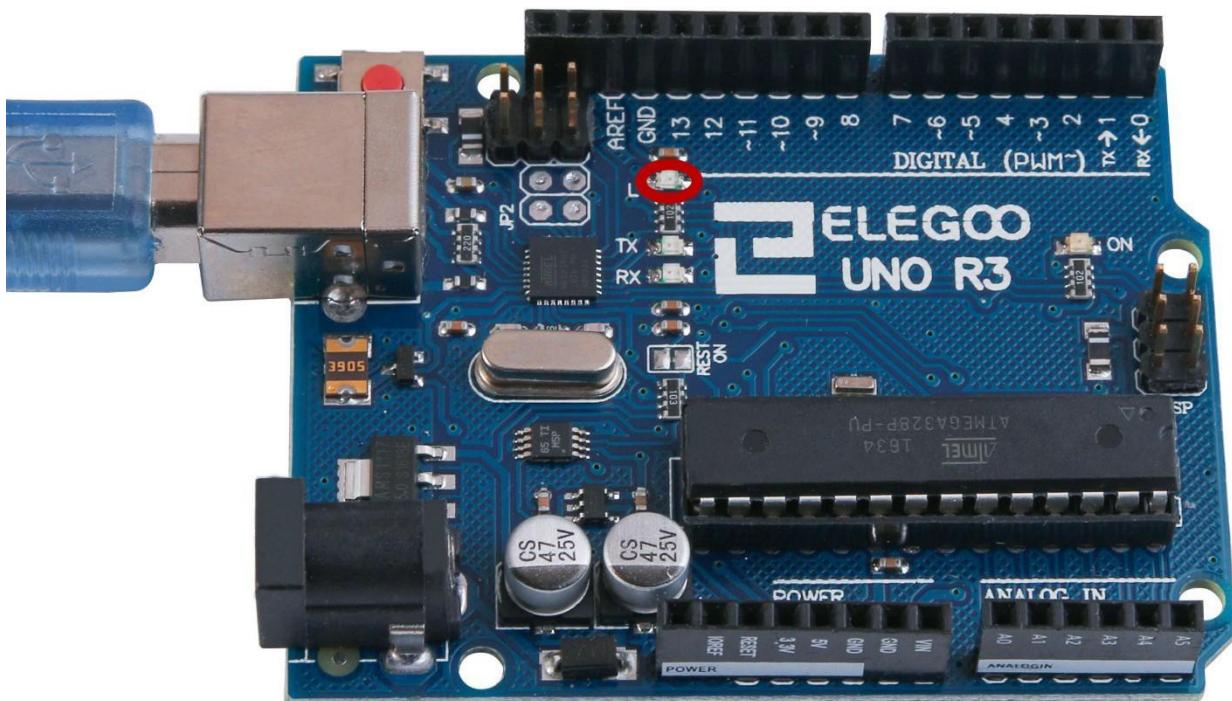
必要な構成部品:

- (1) x Elegoo Uno R3

原理

UNO R3 ボードには、複数の電子デバイスとその機能を拡張するプラグインのシールドに接続するために使用される、両側に沿ったコネクタ列があります。

また、スケッチから制御できる単一の LED もあります。この LED は UNO R3 ボードに内蔵されており、ボード上にラベル付けされているので、しばしば「L」LED と呼ばれます。



UNO R3 ボードの「L」LED は、USB プラグに接続すると既に点滅しています。これは、ボードには一般に「Blink」スケッチがプリインストールされて出荷されるためです。

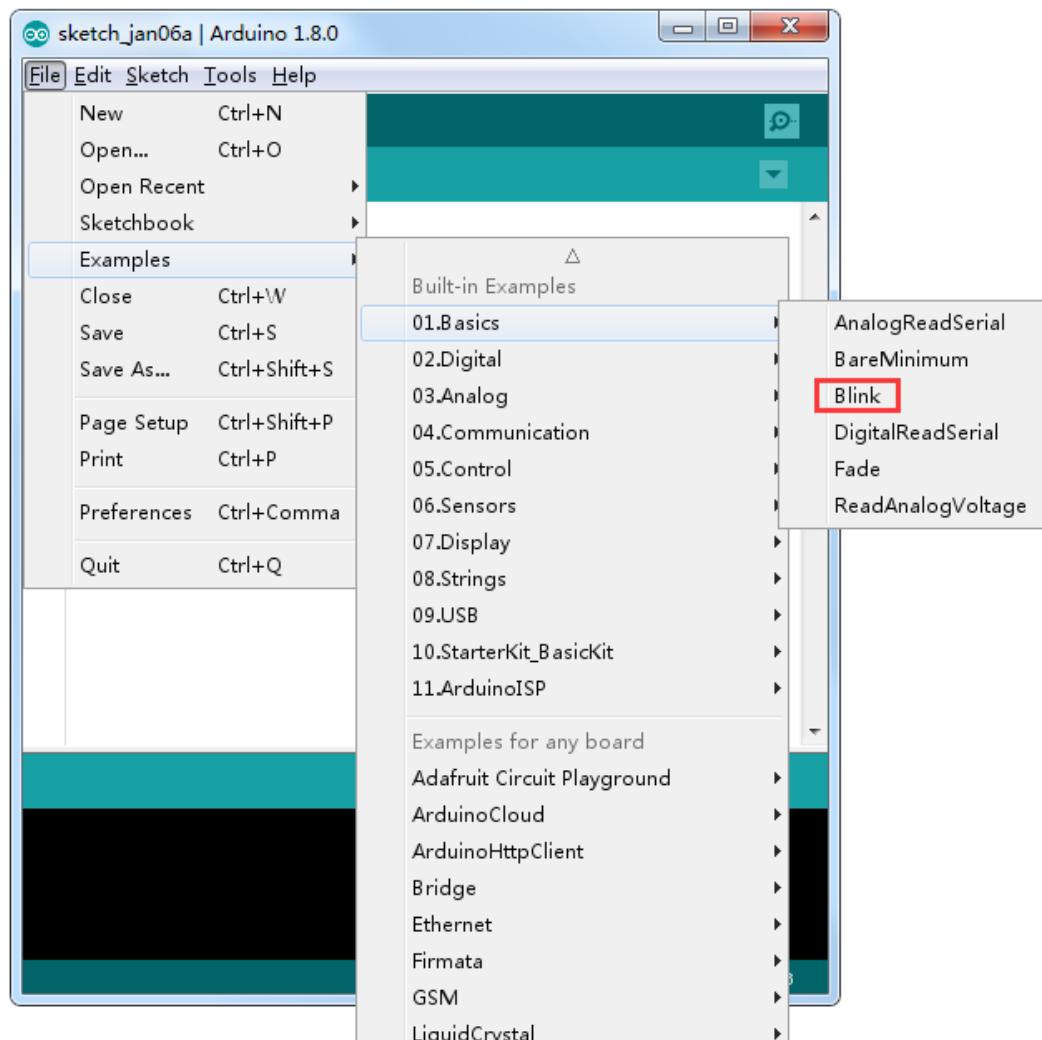
このレッスンでは、UNO R3 ボードを独自の Blink スケッチで再プログラムし、点滅するレートを変更します。

レッスン 0 では、Arduino IDE をセットアップし、UNO R3 ボードに接続するための正しいシリアルポートを見つけることができるることを確認しました。今、あなたの UNO R3 ボードをテストしてプログラミングできます。

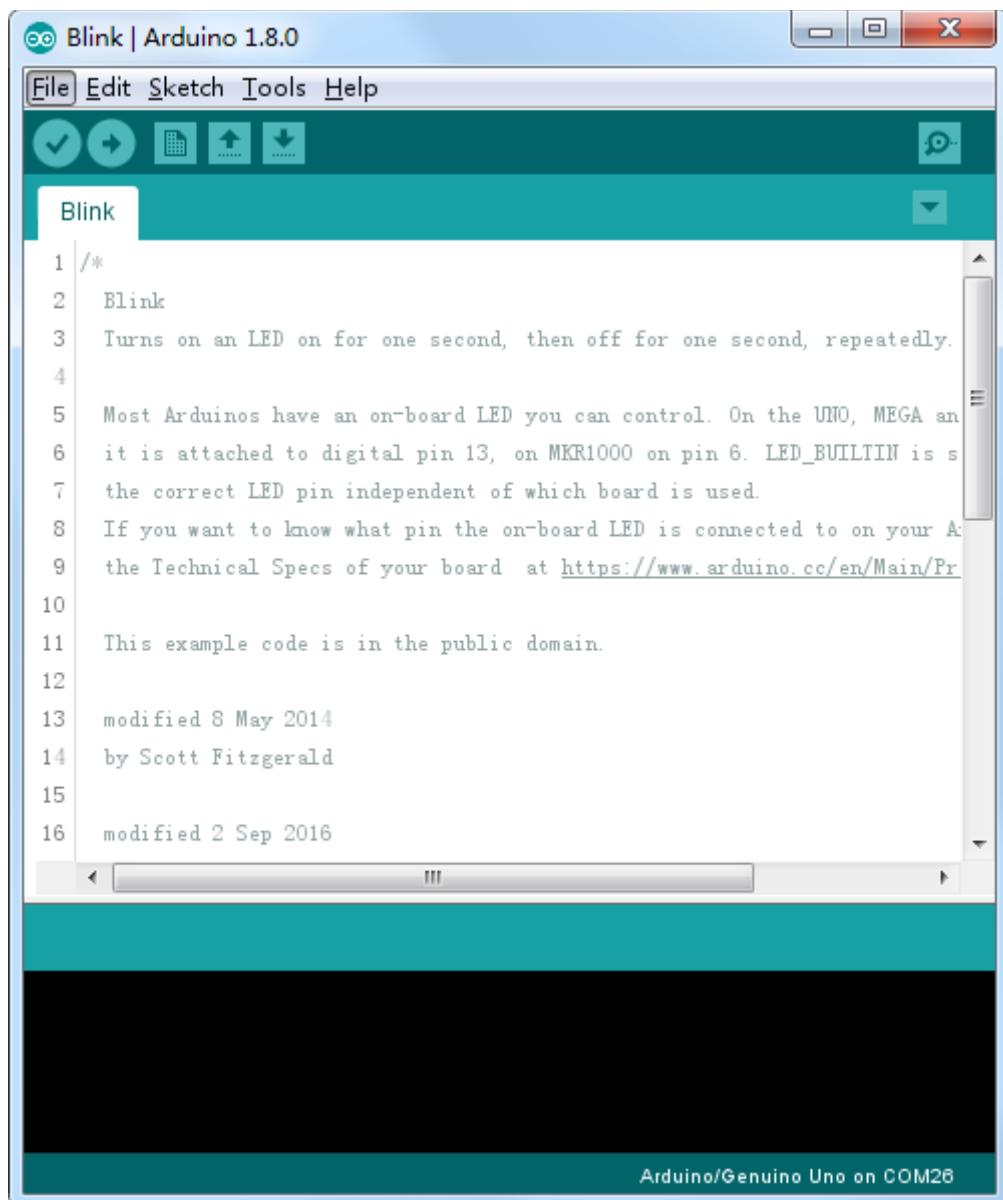
Arduino IDE には、ロードして使用できるサンプルスケッチの大きなコレクションが含まれています。

これには、「L」LED を点滅させるためのスケッチ例が含まれています。

IDE のメニュー系統で[ファイル]>[サンプル]>[01.Basics]の[Blink]スケッチをロードします。



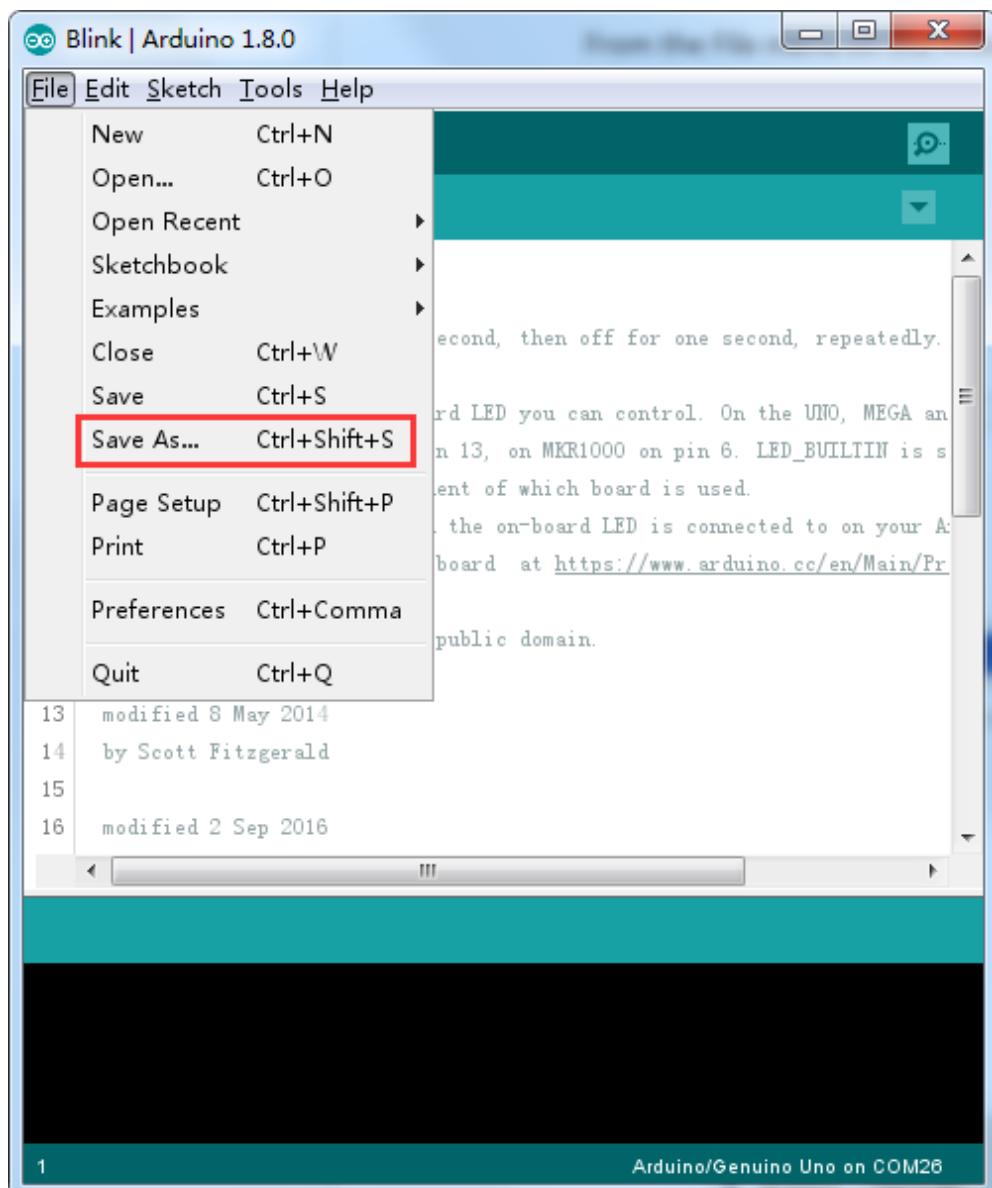
スケッチウィンドウが開いたら、拡大してスケッチ全体をウィンドウで見ることができます。

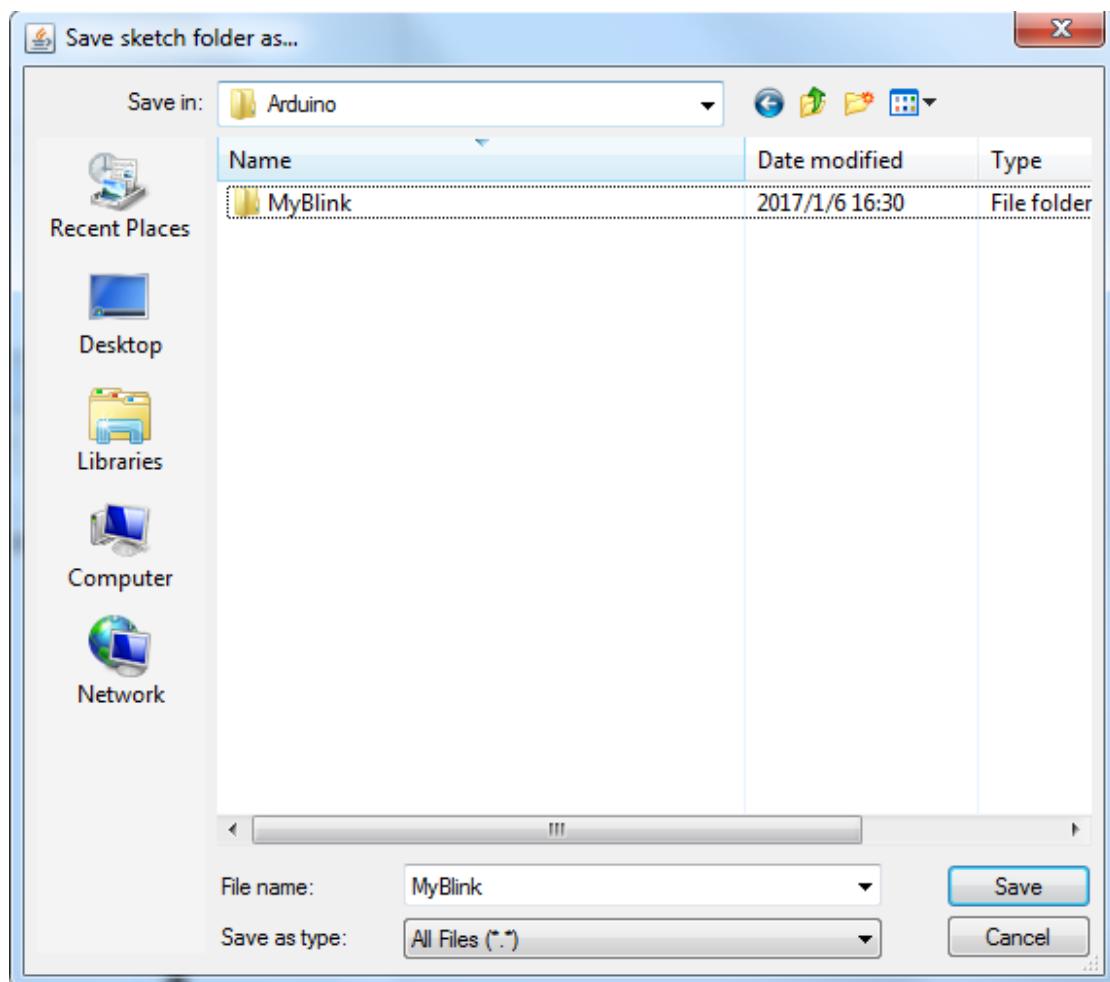


Arduino IDEに含まれているスケッチの例は「読み取り専用」です。つまり、それらを UNO R3 ボードにアップロードできますが、変更した場合は同じファイルとして保存することはできません。

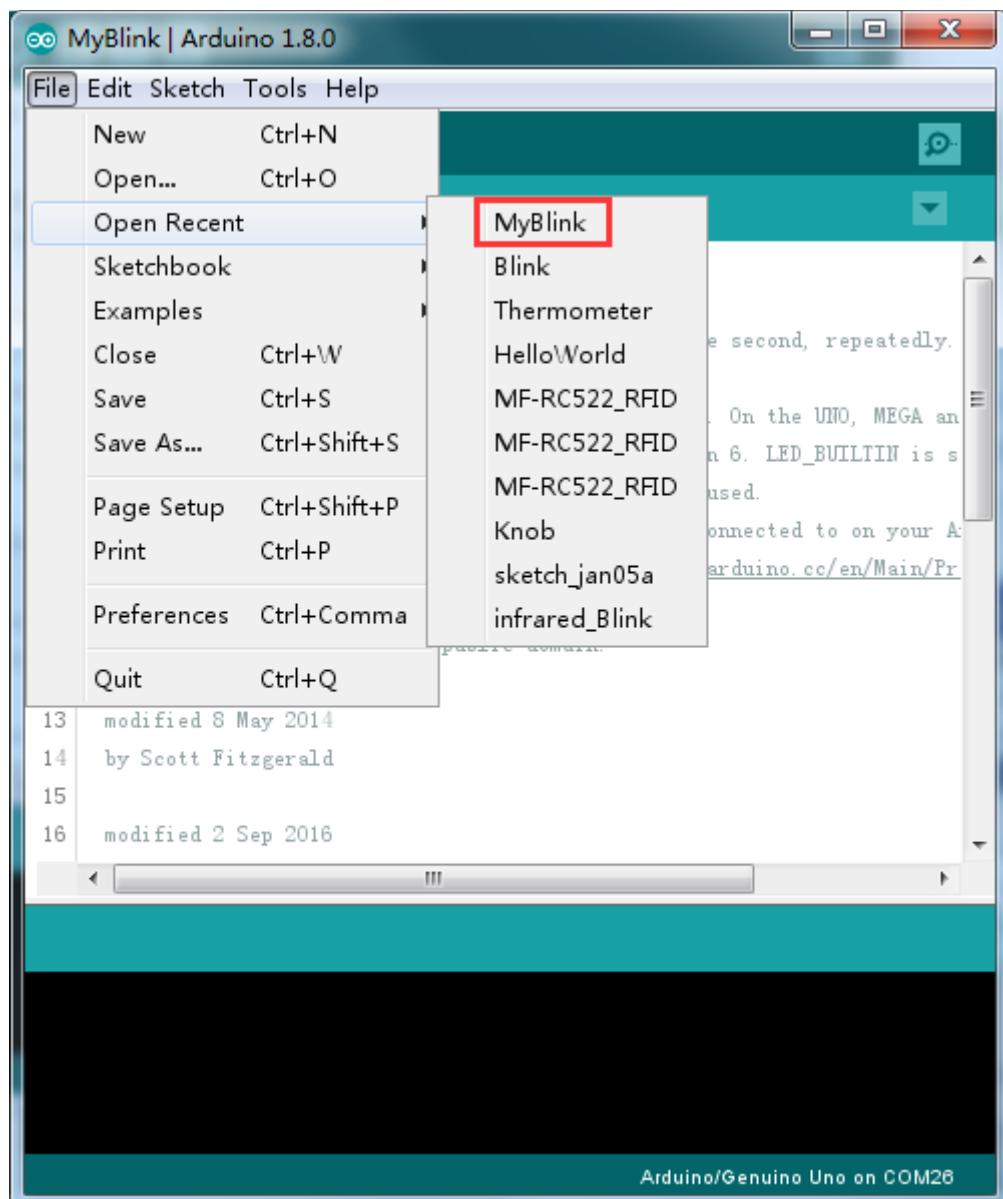
このスケッチを変更する予定なので、最初に行う必要があるのは、自分のコピーを保存することです。

Arduino IDE のファイルメニューから「名前を付けて保存」を選択し、スケッチを「MyBlink」という名前で保存します。

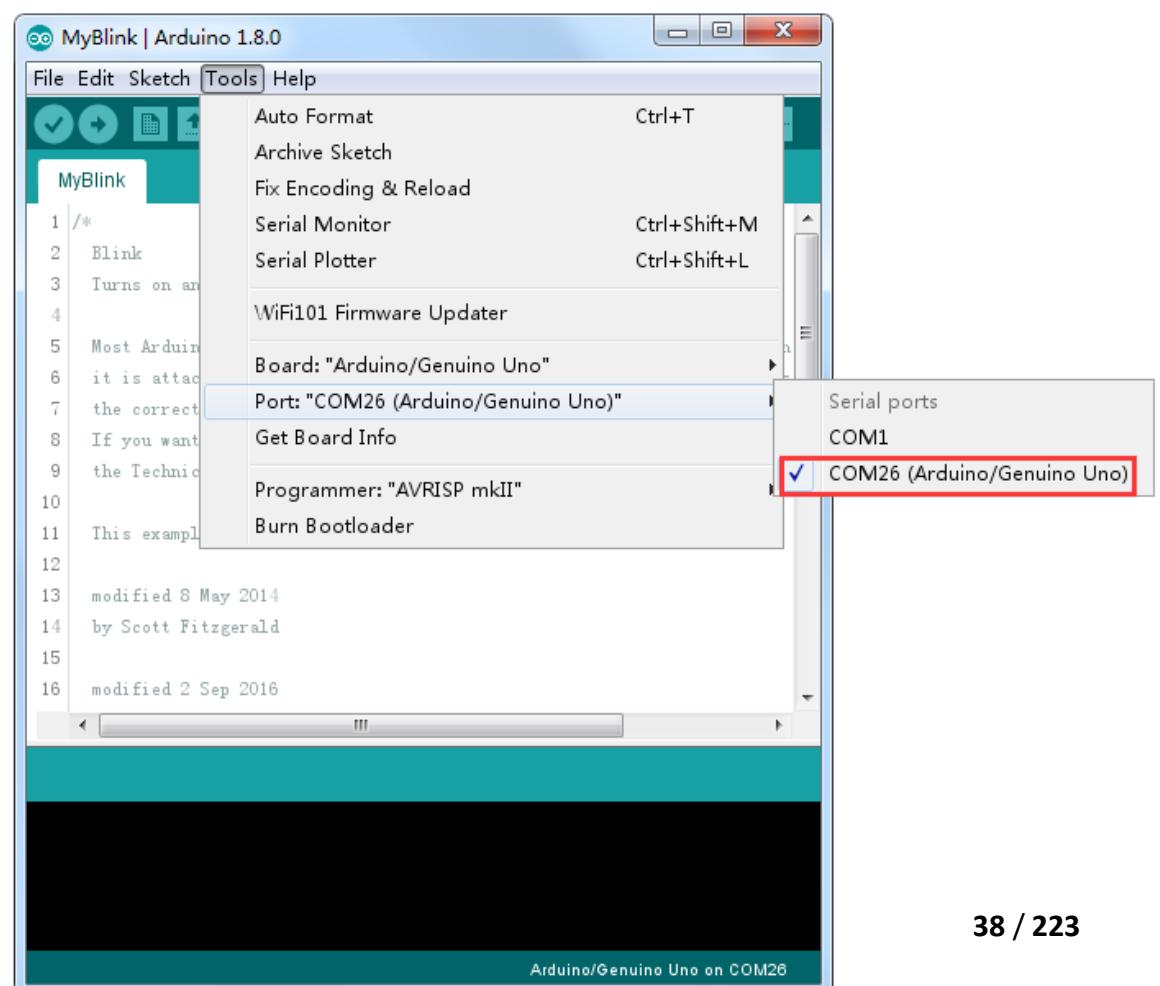
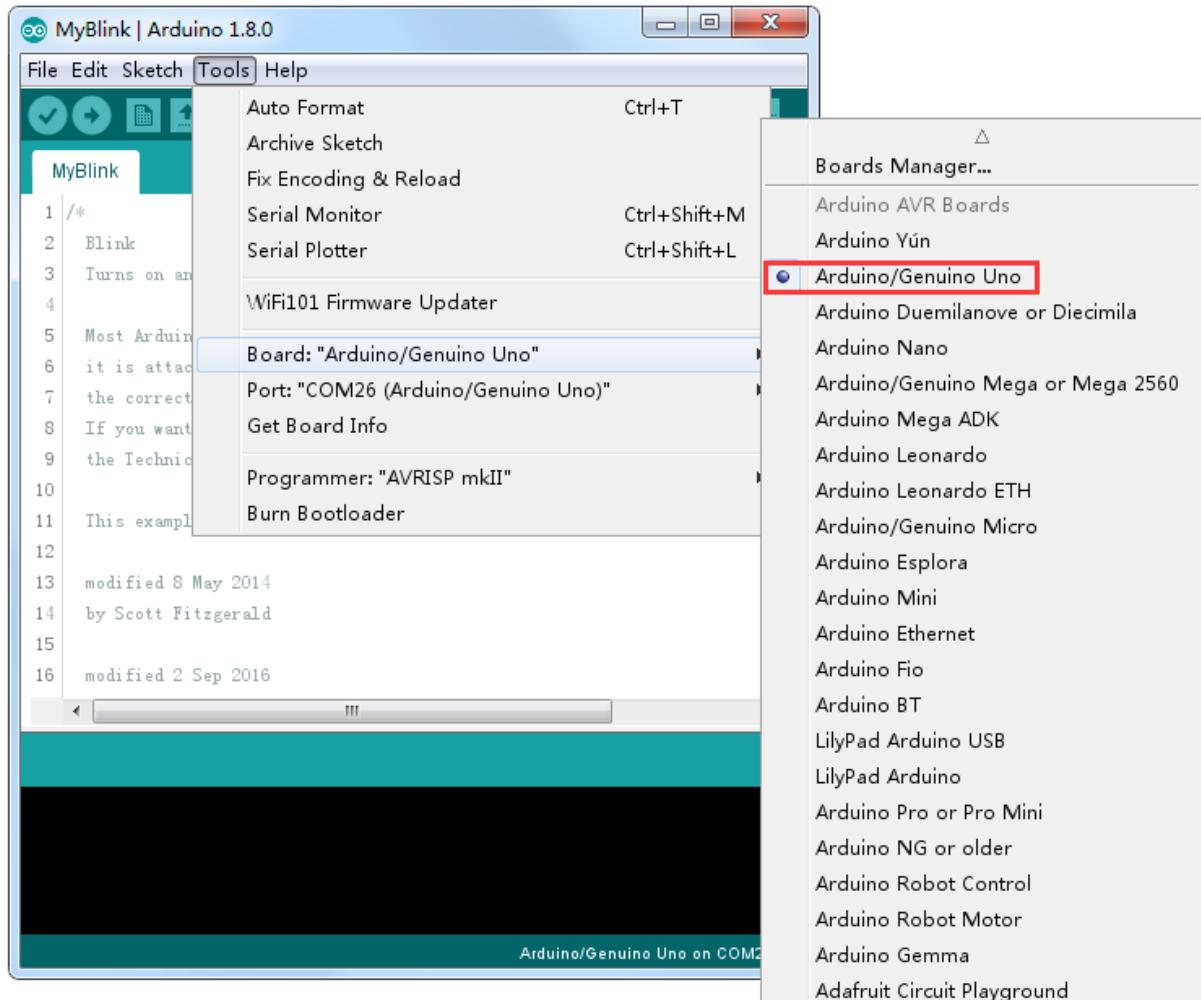




スケッチブックにあなたのコピーを保存しました。つまり、これを再度検索したい場合は、「ファイル」>「スケッチブック」メニューのオプションを使用して開くことができます。

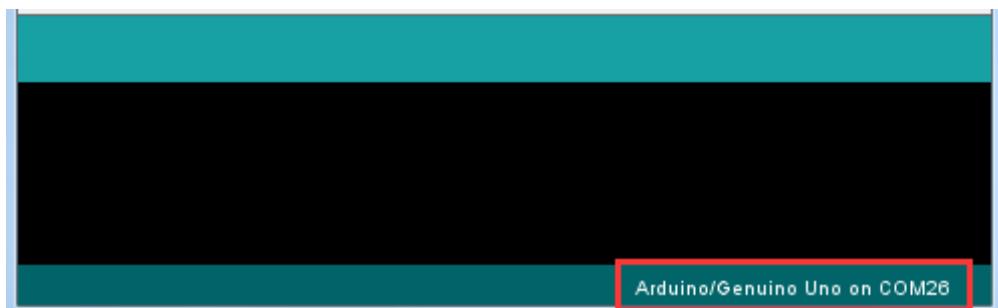


コンピュータに Arduino ボードを USB ケーブルで接続し、'Board Type'と 'Serial Port'が正しく設定されていることを確認します。



注: ボードタイプとシリアルポートは、ここに示したものと必ずしも同じではありません。2560を使用している場合は、Board TypeとしてMega 2560を選択する必要があります。他の選択肢も同じ方法で行うことができます。そして、すべての人に表示されるシリアルポートは、ここで選択されたCOM 26にもかかわらず、あなたのコンピュータ上のCOM3またはCOM4である可能性があります。正しいCOMポートは、認証基準であるCOMX(arduino XXX)であるはずです。

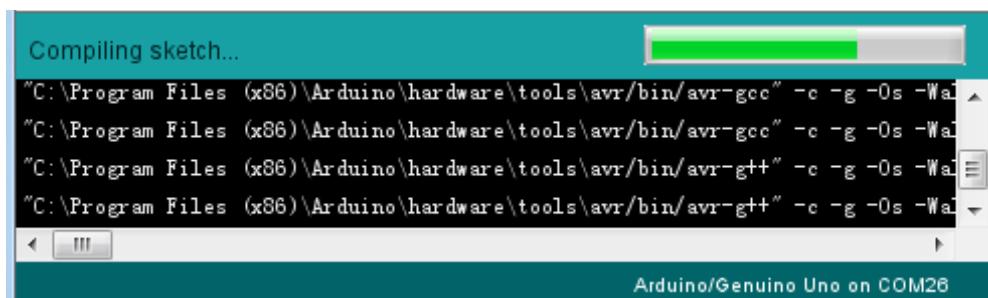
Arduino IDEは、ウィンドウの下部にあるボードの現在の設定を表示します。



[アップロード]ボタンをクリックします。ツールバーの左から2番目のボタン。



IDEのステータス領域を見ると、進行状況バーと一緒にメッセージが表示されます。最初は、「スケッチの編集...」と表示されます。これは、スケッチをボードにアップロードするのに適したフォーマットに変換します。



次に、ステータスが「アップロード中」に変わります。この時点で、ArduinoのLEDは、スケッチが転送されるときにちらつき始めるはずです。

```
Uploading...
C:\Program Files (x86)\Arduino\hardware\tools\avr/bin/avr-objcopy -O ihex -I
Sketch uses 928 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for l
Arduino/Genuino Uno on COM26
```

最後にステータスが「完了」に変わります。

```
Done uploading.
C:\Program Files (x86)\Arduino\hardware\tools\avr/bin/avr-objcopy -O ihex -I
Sketch uses 928 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for l
Arduino/Genuino Uno on COM26
```

他のメッセージは、スケッチが利用可能な 32,256 バイトの 928 バイトを使用していることを示しています。「スケッチをコンパイルしています」段階の後、次のエラーメッセージが表示されます:

```
Problem uploading to board. See http://www.arduino.cc/en/Troubleshooting/Uploa
Copy error messages
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 10 of 10: not in sync: resp=0x22
Problem uploading to board. See http://www.arduino.cc/en/Guide/Troubleshooting#U
Arduino/Genuino Uno on COM1
```

ボードがまったく接続されていないか、ドライバがインストールされていないか（必要な場合）、または間違ったシリアルポートが選択されている可能性があります。

これに遭遇した場合は、レッスン 0 に戻ってインストールを確認してください。

アップロードが完了したら、ボードを再起動して点滅を開始する必要があります。コードを開く

このスケッチの大部分はコメントで構成されていることに注意してください。これらは実際のプログラム命令ではありません。むしろ、プログラムの仕組みを説明するだけです。彼らはあなたの利益のためにそこにいます。

スケッチの最上部にある/*と*/の間のすべてがブロックコメントです。それはスケッチのためのものを説明します。

一行のコメントは、//で始まり、その行の終わりがコメントとみなされるまですべて終わります。

コードの最初の行は次のとおりです：

```
int led = 13;
```

上のコメントが説明するように、これは LED が付いているピンの名前を与えています。 UNO や Leonardo を含む、ほとんどの Arduinos では 13 です。

次に、「設定」機能があります。再度、コメントが示すように、リセットボタンが押されたときに実行されます。電源投入時やスケッチがアップロードされた後など、何らかの理由でボードがリセットされたときにも実行されます。

```
void setup() {  
    // initialize the digital pin as an output.  
    pinMode(led, OUTPUT);  
}
```

すべての Arduino スケッチには 'setup' 機能が必要です。自分で指示を追加したい場所は{と}の間にあります。

この場合、そこにコマンドが 1 つしかありません。コメント状態が Arduino ボードに LED ピンを出力として使用するように指示しています。

スケッチが 'ループ' 機能を持つことも必須です。一度しか動作しない「セットアップ」機能とは異なり、リセット後、「ループ」機能はコマンドの実行が終了した後すぐに再び開始されます。

```
void loop() {  
    digitalWrite(led, HIGH);      // turn the LED on (HIGH is the voltage level)  
    delay(1000);                // wait for a second  
    digitalWrite(led, LOW);       // turn the LED off by making the voltage LOW  
    delay(1000);                // wait for a second  
}
```

ループ機能の中で、コマンドはまず LED ピンを (HIGH) にしてから、1000 ミリ秒 (1 秒) だけ遅延させてから、LED ピンをオフにしてもう 1 秒間ポーズします。

あなたは今あなたの LED を速く点滅させるつもりです。あなたが推測したように、これの鍵は、'delay' コマンドのための () のパラメータを変更することにあります。

```
30 // the loop function runs over and over again forever
31 void loop() {
32   digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the volt
33   delay(500)                      // wait for a second
34   digitalWrite(LED_BUILTIN, LOW);   // turn the LED off by making the vo
35   delay(500)                      // wait for a second
36 }
```

この遅延時間はミリ秒単位であるため、LED を 2 倍速く点滅させたい場合は、値を 1000 から 500 に変更します。その後、1 秒間ではなく、0.5 秒ごとに一時停止します。

スケッチをもう一度アップロードすると、LED がすぐに点滅し始めます。

Lesson 3 LED

概要

このレッスンでは、異なる値の抵抗を使用して LED の輝度を変更する方法を学習します。

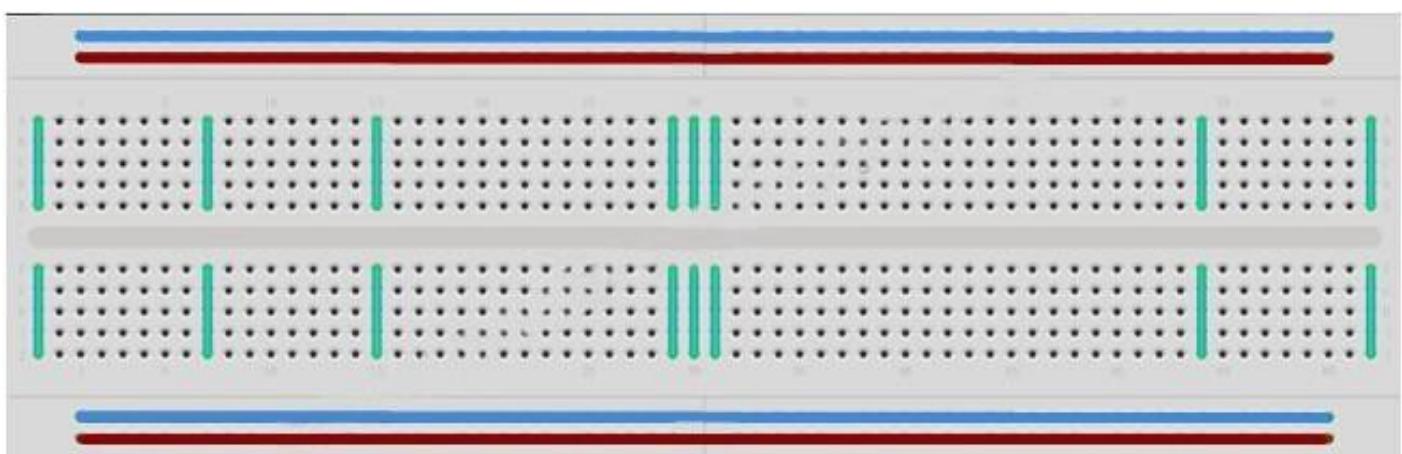
必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x 5mm red LED
- (1) x 220 ohm resistor
- (1) x 1k ohm resistor
- (1) x 10k ohm resistor
- (2) x M-M wires (Male to Male jumper wires)

部品の紹介

BREADBOARD MB-102:

ブレッドボードを使用すると、接続をハンダ付けすることなく、素早く回路を試作できます。以下は例です。



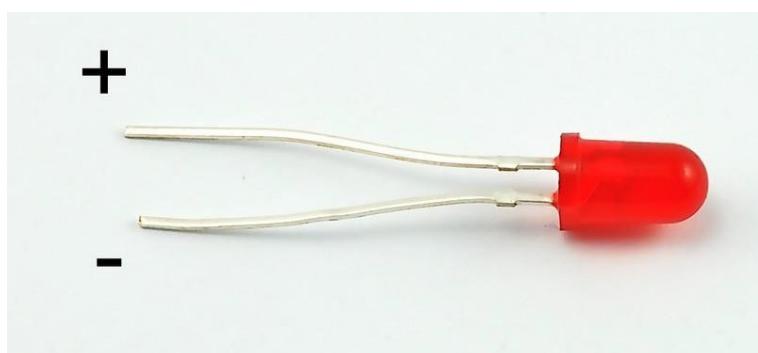
パンフレットには様々なサイズと構成があります。最も単純な種類は、プラスチックブロックの穴のグリッドです。内部には短い列の穴の間を電気的に接続するための金属ストリップがあります。2つ異なるコンポーネントの脚部を同じ列に押し込むと、それらを電気的に結合します。中央の深いチャンネルは、そこに接続が途切れていることを示します。つまり、チャンネルの両側にある脚をチップと一緒に接続せずに差し込むことができます。一部のブレッドボードには、ボードの長辺に沿ってメイングリッドから離れた2つのストリップの穴があります。これらのストリップは内部のボードを通り、共通の電圧を接続する手段を提供します。これらは通常+5ボルトとグランドのペアになります。これらのストリップはレールと呼ばれ、ボード内の多くのコンポーネントやポイントに電源を接続することができます。

ブレッドボードはプロトタイピングには最適ですが、いくつかの制限があります。接続はプッシュフィットで一時的なので、はんだ付けされた接続ほど信頼性がありません。断続的な回路に問題がある場合は、ブレッドボードの接続不良の可能性があります。

LED:

LED が優れたインジケータライトを作ります。彼らは電気をほとんど使わず、長寿命です。

このレッスンでは、最も一般的なすべての LED (5mm 赤色 LED) を使用します。5mm は LED の直径を意味します。他の一般的なサイズは 3mm と 10mm です。LED をバッテリまたは電圧源に直接接続することはできません。その理由は、1) LED は正と負のリード線を持ち、誤った方向に置かれた場合は点灯しません。2) LED は抵抗を使って電流を制限するか「チョーク」する必要があります。さもなければ、それは燃え尽きるでしょう！



LED を付属の抵抗器を使用しないと、あまりにも多くの電流が流れ、加熱され、光が発生する「接合部」を破壊するため、ほとんど直ちに破壊される可能性があります。

どちらが LED の正のリードであり、負のリードであるかを知るには 2 つの方法があります。

まず、陽性リードは長くなります。

第 2 に、負のリードが LED の本体に入る場合、LED のケースに対して平坦な縁がある。

長いリードの横に平坦な側面を持つ LED がある場合は、長いリードが陽性であると認識する必要があります。

抵抗器:

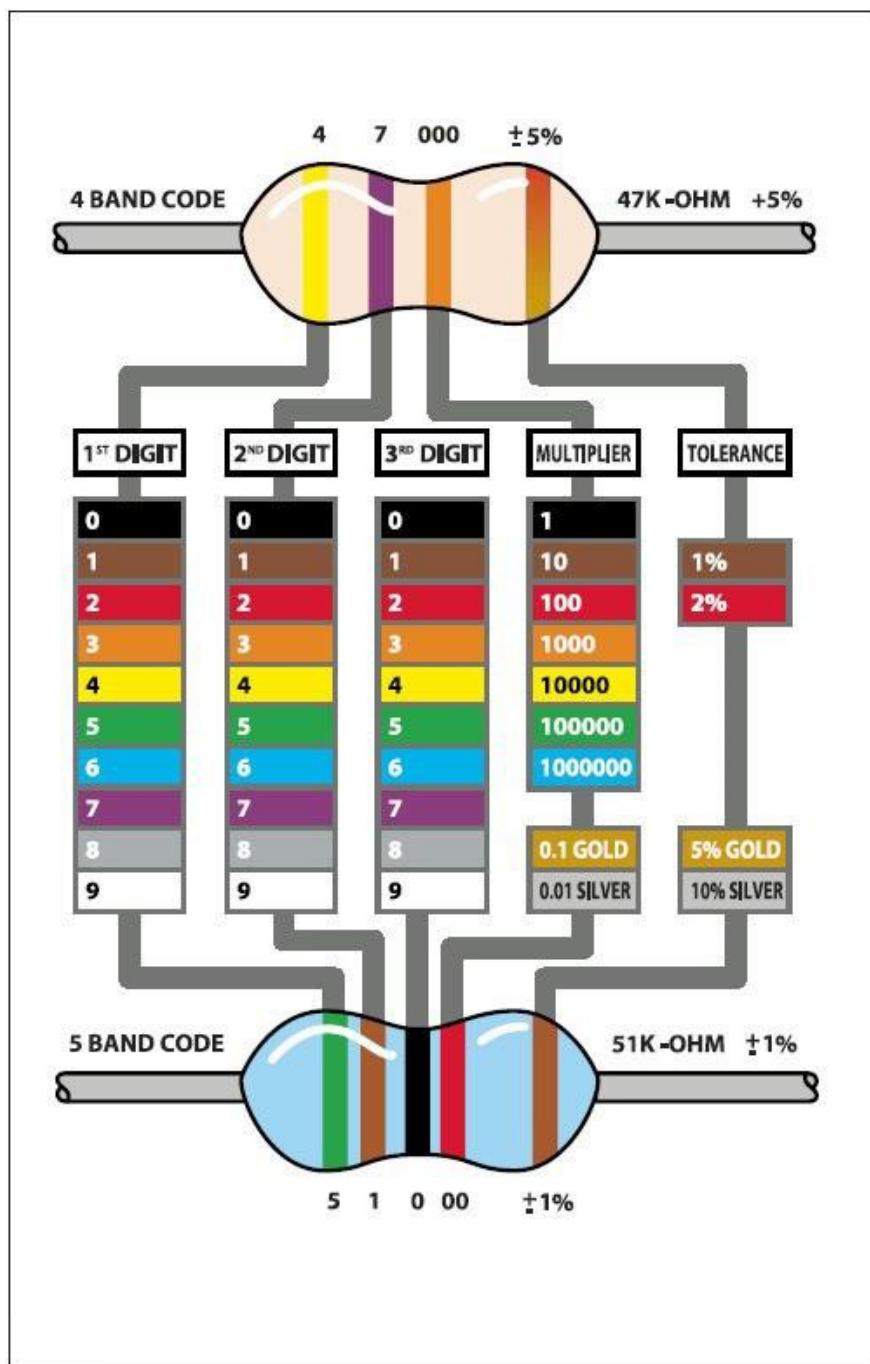
名前が示すように、抵抗器は電気の流れに抵抗します。 抵抗の値が高いほど、抵抗が大きくなり、流れる電流は少なくなります。 LED をどれだけの電流が流れているかを制御するために、これを使用して、それがどれだけ明るく輝いているかを制御します。



しかし、まず、さらに抵抗について...

抵抗の単位はオームと呼ばれ、通常はオーム語のギリシャ文字 Ω に短縮されます。 オームは抵抗値が低いため（小さい抵抗です）、 $k\Omega$ ($1,000\Omega$) と $M\Omega$ ($1,000,000\Omega$) も抵抗値を示します。 これらはキロオームとメガオームと呼ばれます。

このレッスンでは、 220Ω 、 $1k\Omega$ 、 $10k\Omega$ の 3 種類の抵抗器を使用します。 これらの抵抗器はすべて、同じ色をしていますが、異なる色の縞模様があります。 これらのストライプは抵抗の値を示します。 レジスタのカラーコードには 3 つの色のストライプがあり、一方の端に金色のストライプがあります。

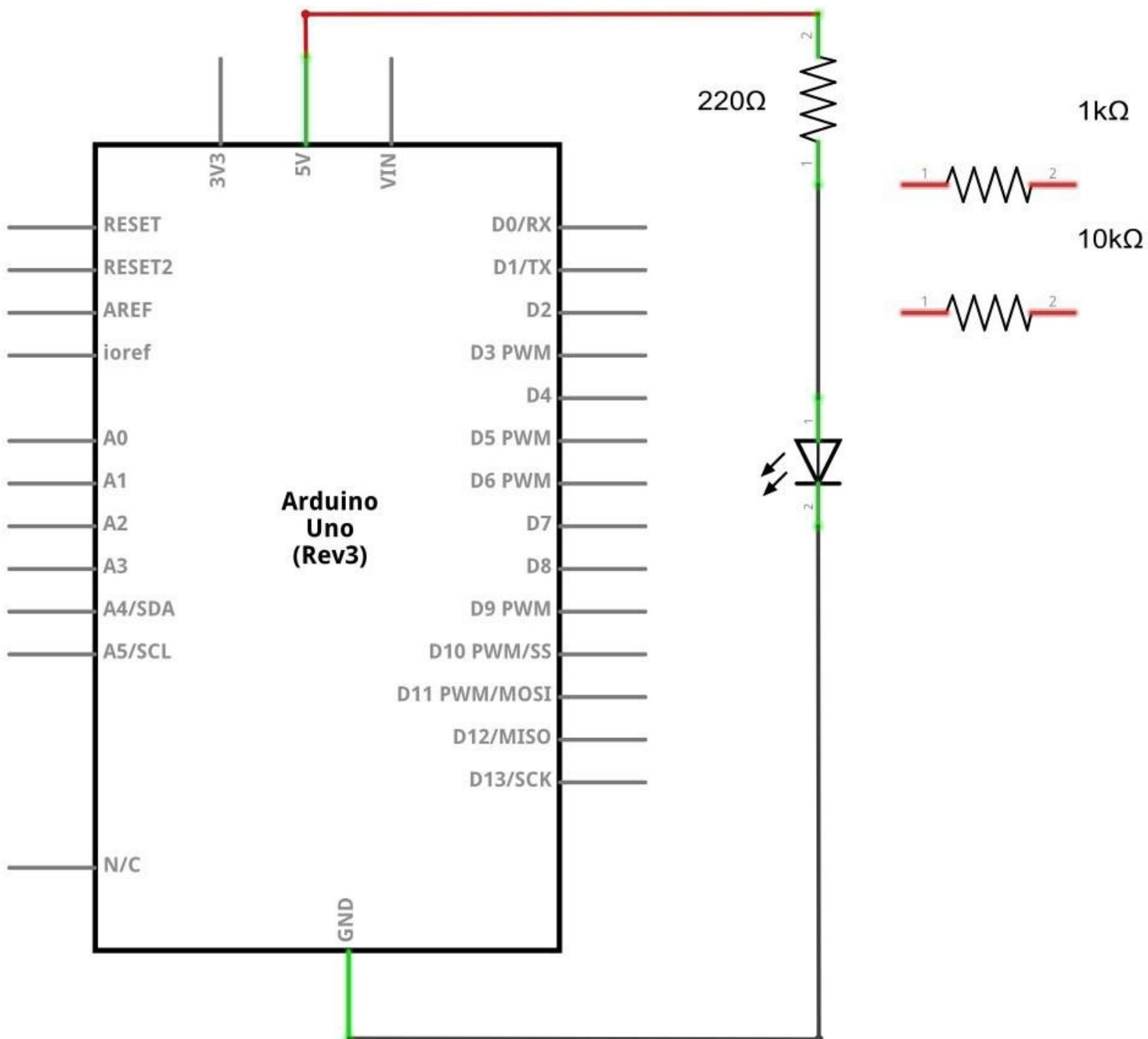


LED とは異なり、抵抗には正と負のリードがありません。それらはいずれかの方法で接続することができます。

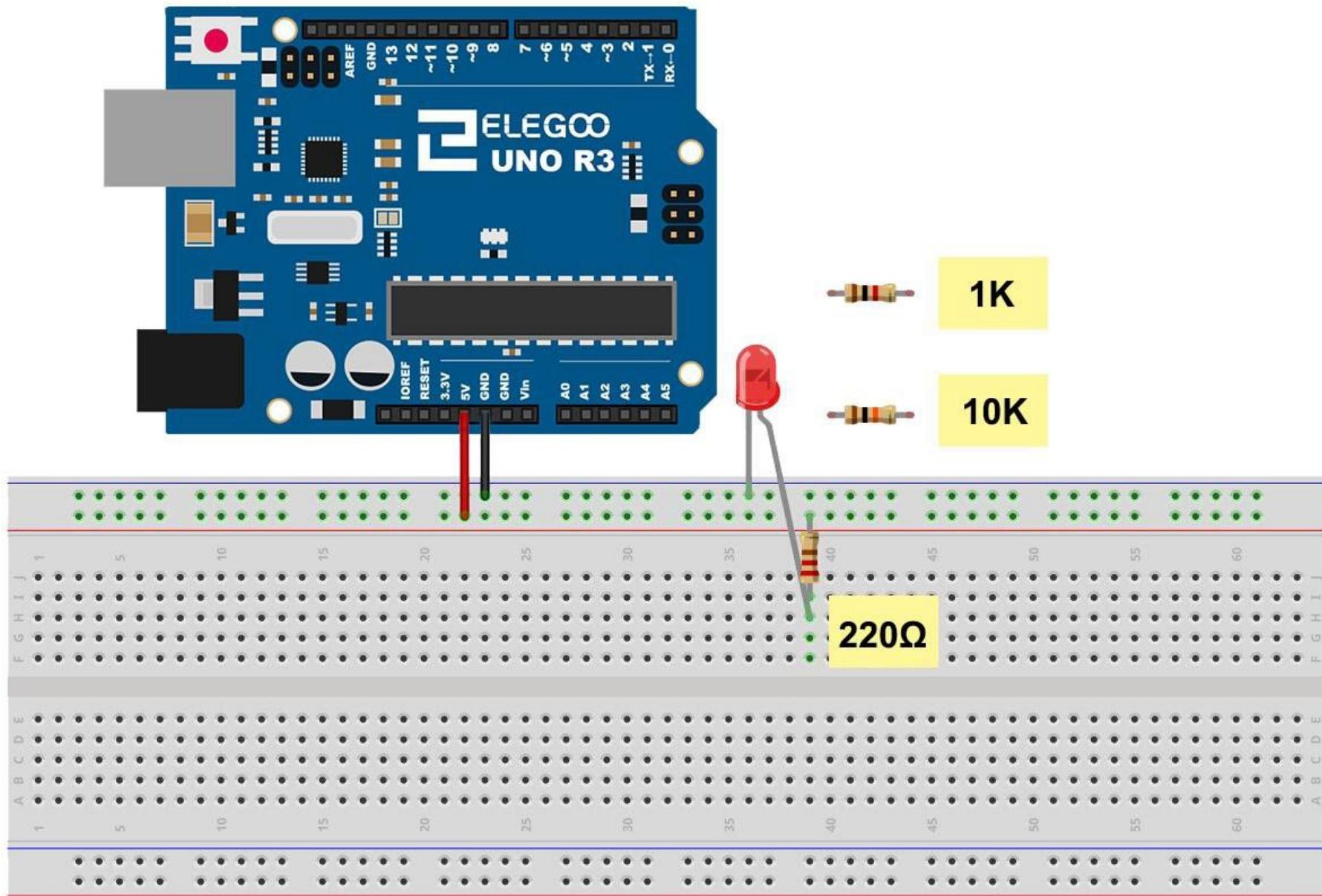
このアプローチ方法が複雑すぎるとわかったら、私たちの抵抗のカラーリングフラグを直接読んでその抵抗値を決定することができます。または、代わりにデジタルマルチメータを使用することもできます。

Connection

Schematic



Wiring diagram



UNO は、LED と抵抗に電力を供給するために使用する 5 ボルトの便利な電源です。 UNO を USB ケーブルに差し込む以外は何もする必要はありません。

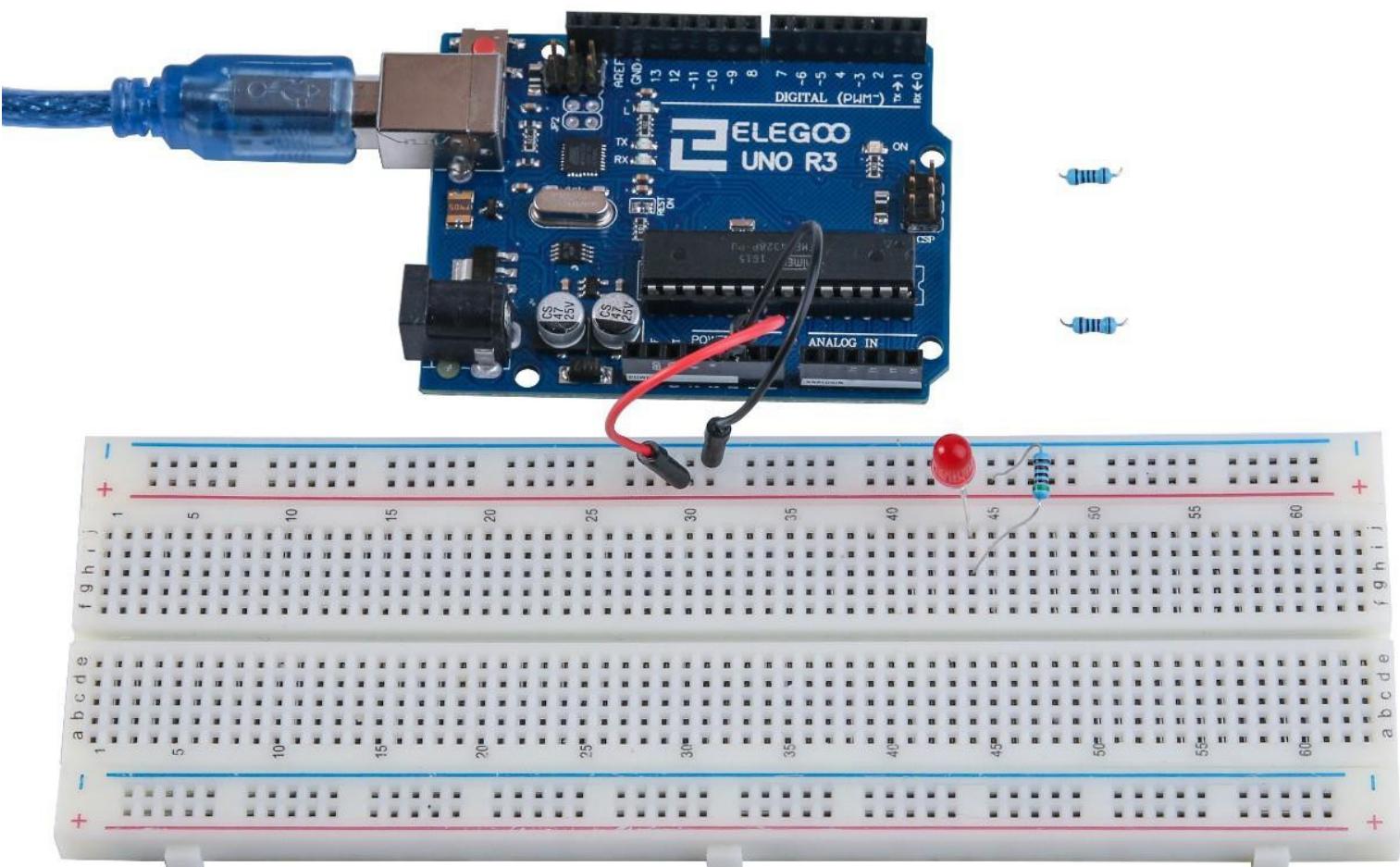
220Ωの抵抗を取り付けた場合、LED は非常に明るいはずです。 1kΩ抵抗の 220Ω抵抗を交換すると、LED が少し減光します。 最後に、10kΩの抵抗を配置すると、LED が目に見えます。 赤いジャンパリードをブレッドボードから引き出し、穴の中に触れて取り除き、スイッチのように動作させます。 あなたは違いを気付くことができます。

現時点では、抵抗器の一方の脚に 5V が流れ、抵抗器のもう一方の脚は LED のプラス側に、LED のもう一方の側は GND に向かって流れます。 ただし、LED を点灯させるように抵抗を移動すると、以下のように LED が点灯します。

おそらく 220Ωの抵抗を元の場所に戻したいと思うでしょう。

どこかにある限り、LED のどちら側に抵抗を置いても問題ありません。

Example picture



Lesson 4 RGB LED

概要

RGB LED は、プロジェクトに色を追加する楽しく簡単な方法です。 それらは 1 つは 3 つの LED で、それらを使用して接続する方法はそれほど違いはありません。

それらは主に 2 つのバージョンで提供されます: Common Anode または Common Cathode。

共通アノードは共通ピンに 5V を使用し、共通カソードはグランドに接続します。

他の LED と同様に、いくつかの抵抗をインライン (3 つの合計) に接続する必要があります。

私たちのスケッチでは、赤色の状態の LED で始まり、次に緑色にフェードアウトし、次に青色にフェードアウトし、最後に赤色に戻ります。これを行うことで、達成可能な色のほとんどを循環させます。

必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x 830 Tie Points Breadboard
- (4) x M-M wires (Male to Male jumperwires)
- (1) x RGB LED
- (3) x 220 ohm resistors

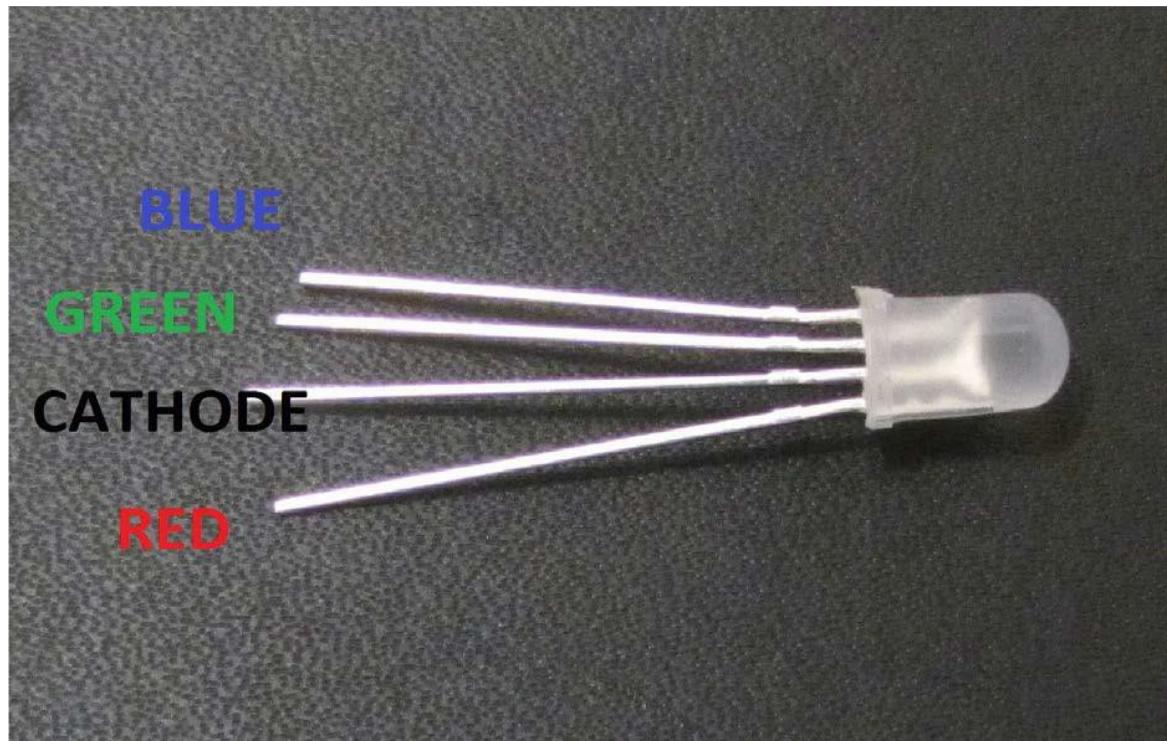
部品の紹介

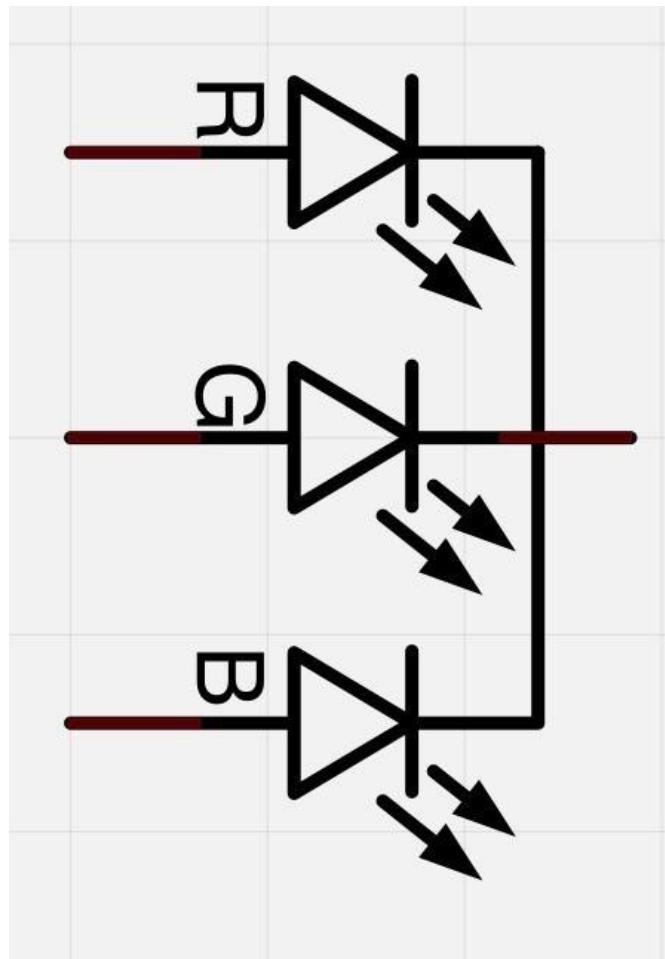
RGB:

一見すると、RGB（赤、緑、青）のLEDは通常のLEDのように見えます。しかし、通常のLEDパッケージの内部には、実際には赤色、緑色、はい、青色の3つのLEDがあります。個々のLEDの明るさを制御することで、あなたが望むどんな色でもかなり混在させることができます。

3つのLEDのそれぞれの明るさを調整することで、パレットにペイントを混ぜるのと同じ方法で色をミックスします。これを行う難しい方法は、レッスン2で行ったように異なる値の抵抗（または可変抵抗）を使用することですが、それは多くの作業です！UNO R3ボードにはanalogWrite機能があり、~でマークされたピンで使用して、適切なLEDにさまざまな電力を出力できます。

RGB LEDには4本のリード線があります。パッケージ内の単一のLEDのそれぞれの正の接続に向かう1つのリードと、LEDの3つの負の側面のすべてに接続された単一のリードがあります。





ここで写真には 4 つの電極 LED が見えます。 緑色、青色、赤色の各ピンはアノードと呼ばれます。

あなたは常にそれに "+"を接続します。 陰極は "-" (地面) に行きます。 それ以外の方法で接続すると、LED は点灯しません。

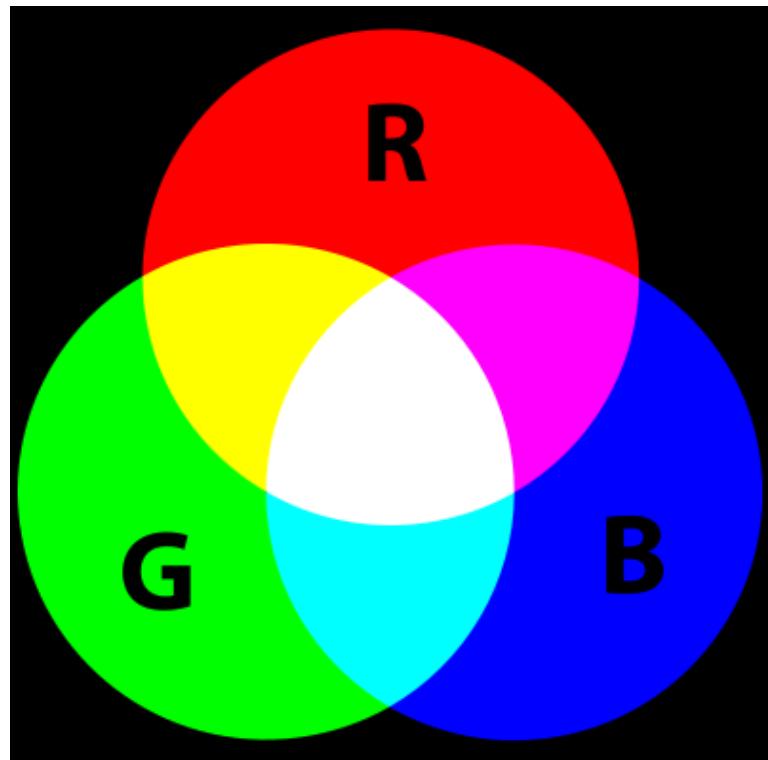
LED パッケージの共通の負の接続は、平らな側から 2 番目のピンです。 それはまた、4 つのリード線のうち最長であり、接地されます。

パッケージ内の各 LED は、あまりにも多くの電流が流れないように、独自の 220Ω 抵抗を必要とします。 LED の 3 つの正のリード線 (赤色、緑色、青色) は、これらの抵抗を使用して UNO 出力ピンに接続されています。

COLOR:

赤、緑、青の光の量を変えることで好きな色を混ぜることができるのは、目には 3 種類の光セプター（赤、緑、青）があるからです。あなたの目と脳は、赤、緑、青の量を処理し、それをスペクトルの色に変換します。

ある意味では、3 つの LED を使用することで、私たちは目は錯覚します。この同じ考え方は、LCD に赤、緑、青の色のドットが並び、各ピクセルを構成しているテレビで使用されています。



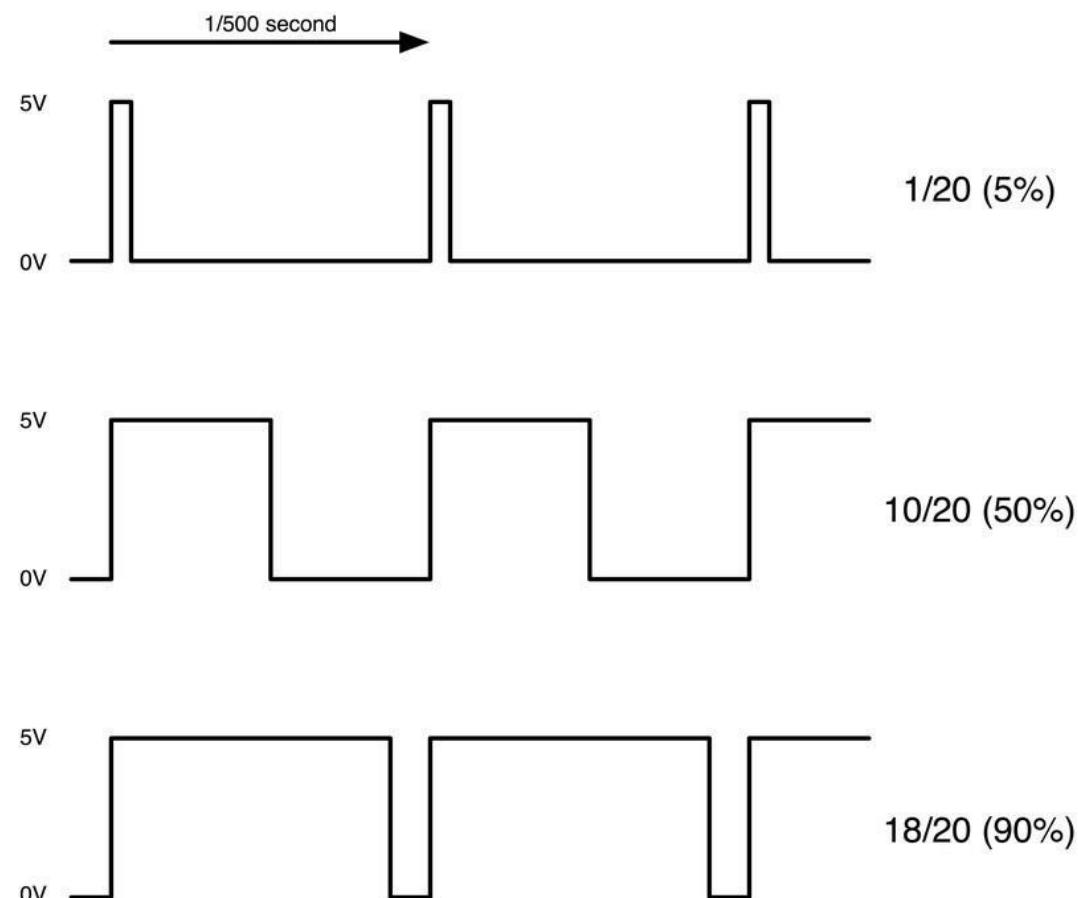
3つすべての LED の輝度を同じに設定すると、光の全体的な色は白色になります。赤と緑の LED だけが同じ明るさになるように青の LED を消すと、ライトは黄色に見えます。

LED の赤、緑、青の各部分の明るさを個別に制御できるので、好きな色を混在させることができます。黒はそれほど光のない色ではありません。したがって、我々の LED で最も近いのは 3 色すべてを消すことです。

理論 (PWM)

パルス幅変調 (PWM) は電力を制御する技術です。また、各 LED の輝度を制御するためにここで使用します。

下の図は、UNO の PWM ピンの 1 つからの信号を示しています。



およそ 1/500 秒ごとに、PWM 出力はパルスを生成します。

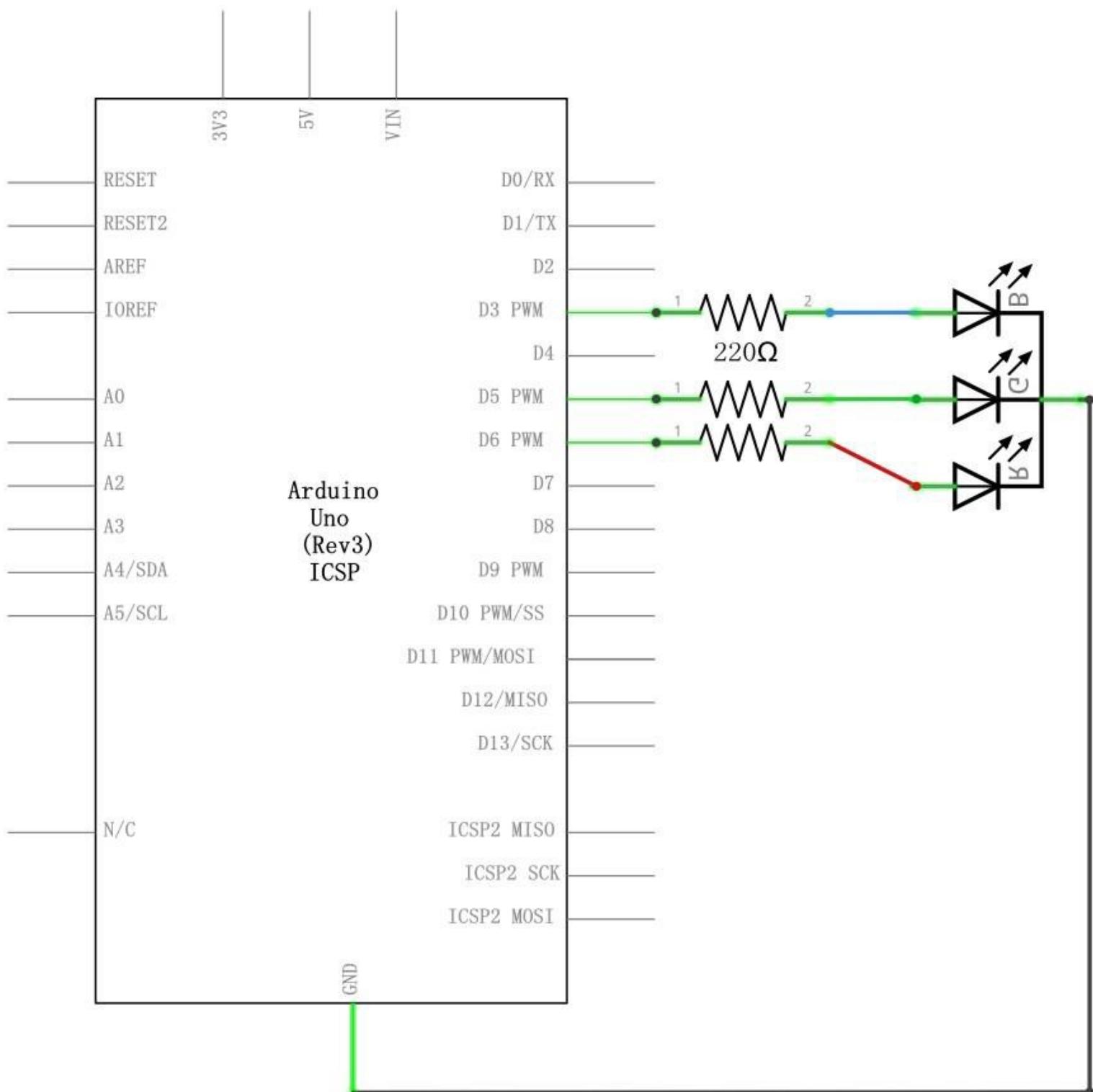
このパルスの長さは、'analogWrite'関数によって制御されます。したがって、'analogWrite (0)'はパルスをまったく生成せず、'analogWrite (255)'は次のパルスが到着するまですべての時間持続するパルスを生成します。その結果、実際には常に出力されます。

'analogWrite'に 0 から 255 の間の値を指定すると、パルスが生成されます。出力パルスが時間の 5%だけ高い場合、駆動しているものは全出力の 5%しか受け取りません。

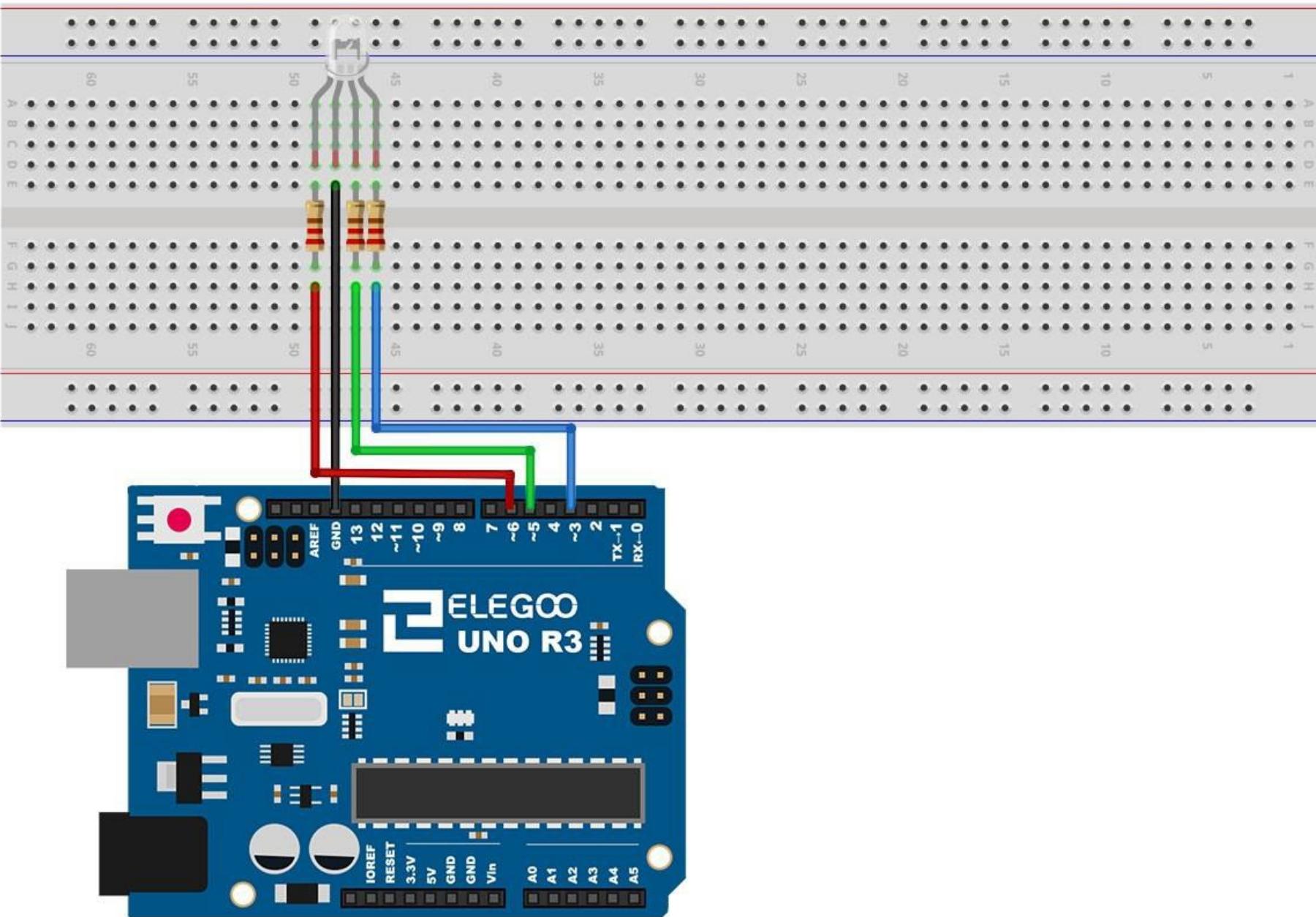
しかし、出力が時間の 90%の間 5V であれば、負荷はそれに供給される電力の 90%を得ます。そのスピードで LED の点滅が見えないので、明るさが変わっているように見えます。

Connection

Schematic

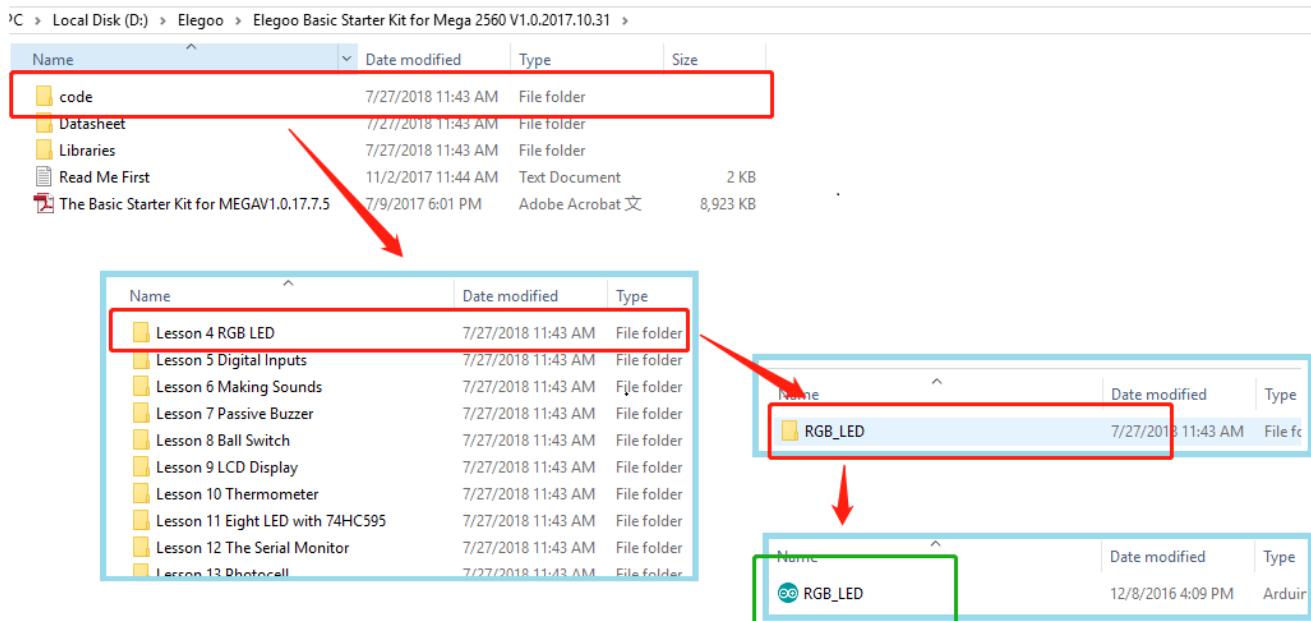


Wiring diagram



Code

After wiring, please open the Sketch in folder path: Tutorial > English > code > Lesson 4 RGB LED > RGB_LED, and click UPLOAD to upload the program.



スケッチは、各色にどのピンを使用するかを指定することから始まります:

```
// Define Pins  
#define BLUE 3  
#define GREEN 5  
#define RED 6
```

次のステップは 'setup' 関数を書くことです。以前のレッスンで学んだように、セットアップ機能は Arduino がリセットされた後に 1 回だけ実行されます。この場合、出力として使用している 3 つのピンを定義するだけです。

```
void setup()  
{  
    pinMode(RED, OUTPUT);  
    pinMode(GREEN, OUTPUT);  
    pinMode(BLUE, OUTPUT);  
    digitalWrite(RED, HIGH);  
    digitalWrite(GREEN, LOW);  
    digitalWrite(BLUE, LOW);  
}
```

'loop' 関数を見る前に、スケッチの最後の関数を見てみましょう。

定義変数

```
redValue = 255; // choose a value between 1 and 255 to change the color.  
greenValue = 0;
```

```
blueValue = 0;
```

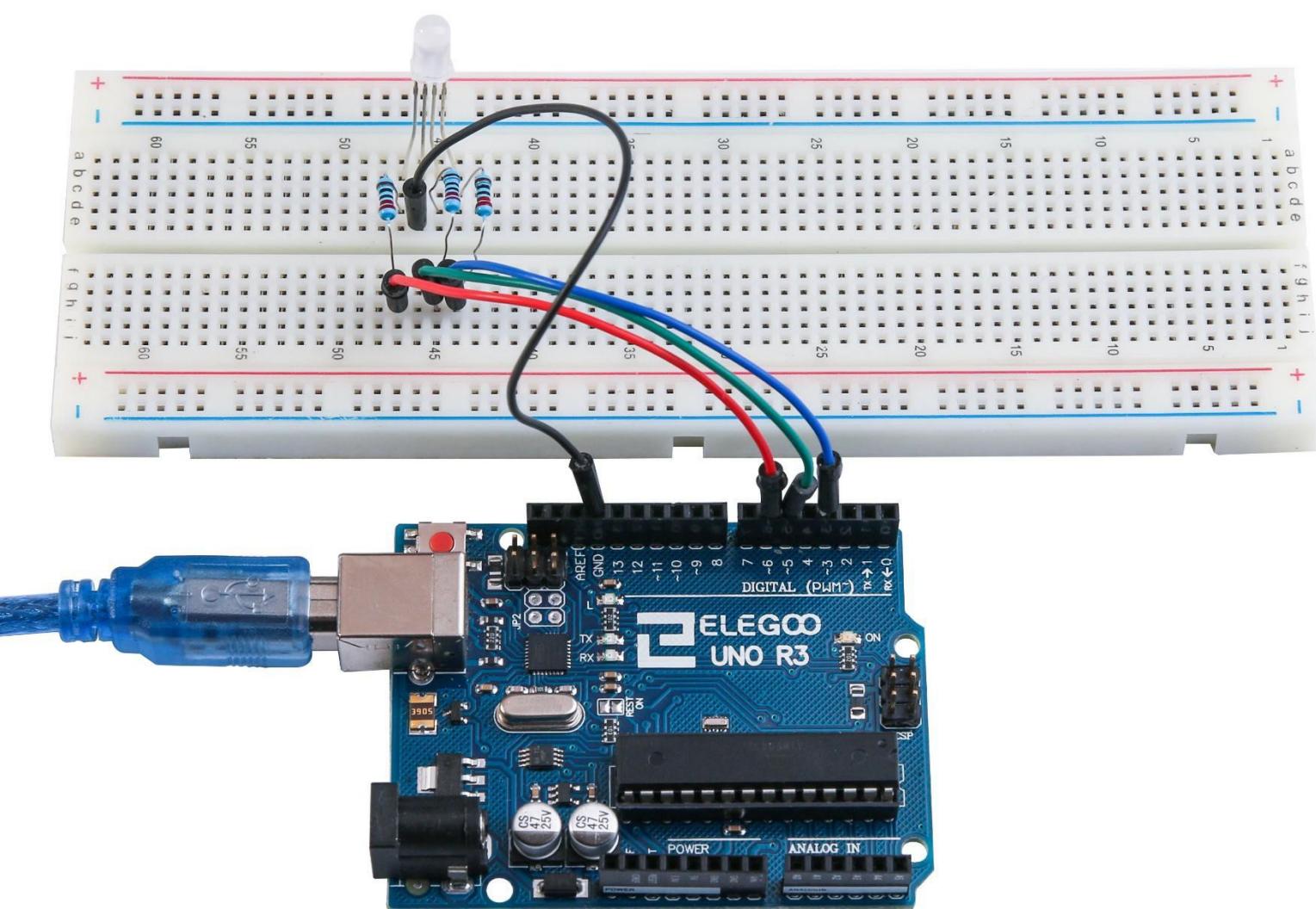
この関数は 3 つの引数を取ります。1 つは赤、緑、青の LED の明るさです。 いずれの場合も、数値は 0~255 の範囲内にあり、0 はオフ、255 は最大輝度を意味します。 次に、この関数は 'analogWrite' を呼び出して各 LED の輝度を設定します。

'ループ'機能を見ると、赤色、緑色、青色の光量を表示してから 1 秒間一時停止してから次の色に移動することがわかります。

```
#define delayTime 10 // fading time between colors  
Delay(delayTime);
```

スケッチにあなた自身のいくつかの色を追加し、あなたの LED の効果を見てみてください。

Example picture



Lesson 5 デジタル入力

概要

このレッスンでは、LED をオン/オフするためにデジタル入力付きのプッシュボタンを使用する方法を学習します。

ボタンを押すと LED が点灯します。もう一方のボタンを押すと LED が消灯します。

必要な構成部品:

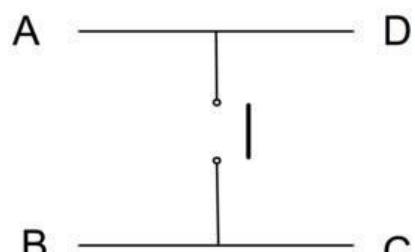
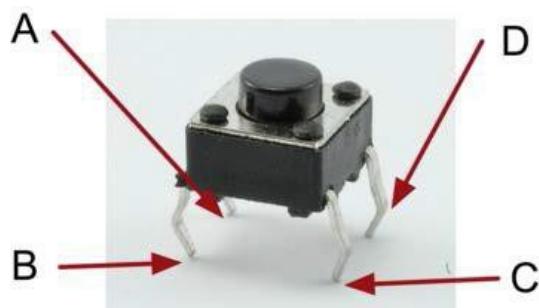
- (1) x Elegoo Uno R3
- (1) x 830 Tie-points Breadboard
- (1) x 5mm red LED
- (1) x 220 ohm resistor
- (2) x push switches
- (7) x M-M wires (Male to Male jumper wires)

部品の紹介

PUSH SWITCHES:

スイッチは本当にシンプルなコンポーネントです。ボタンを押すかレバーを回すと、2つの接点が電気的に接続されます。

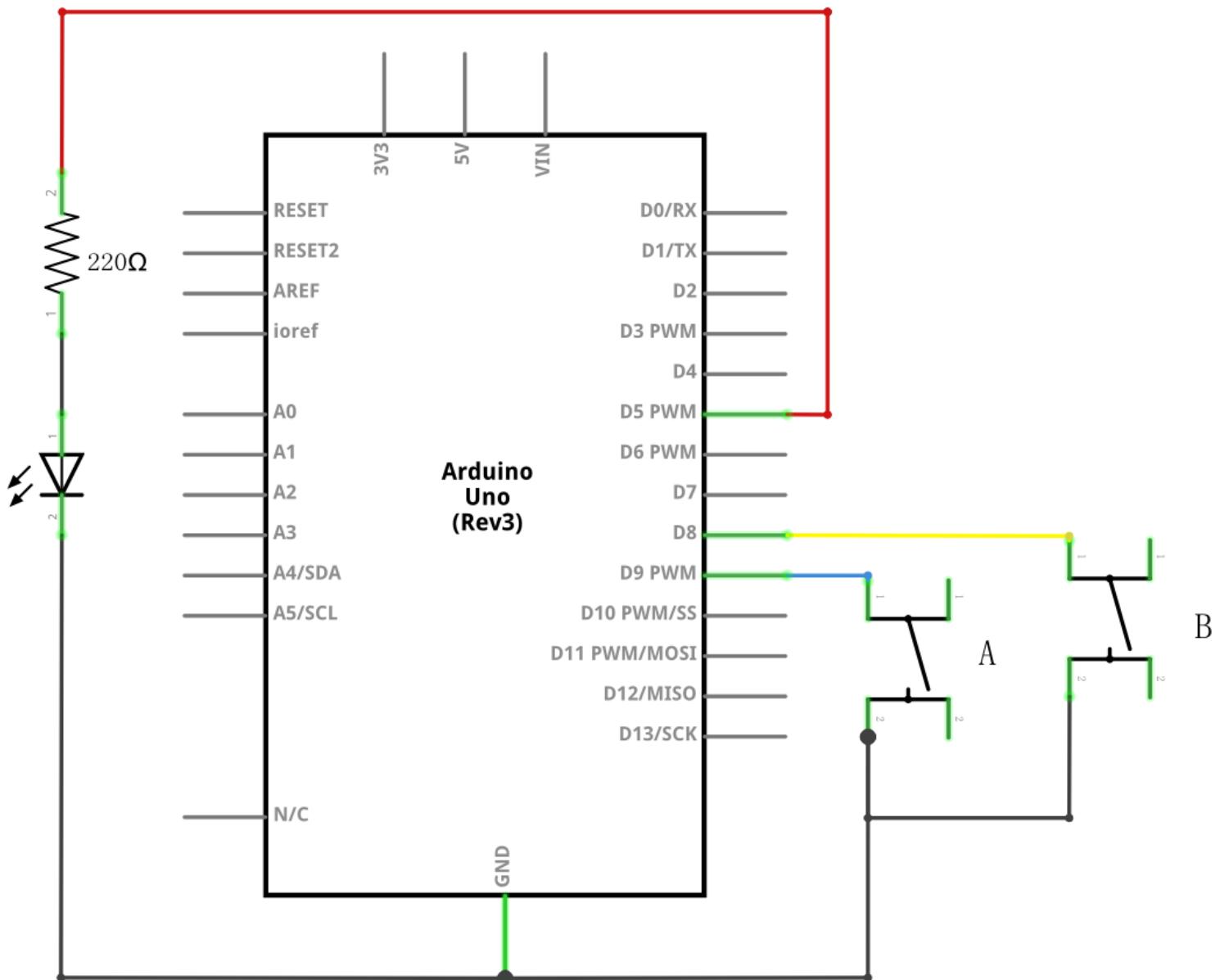
このレッスンで使用されているソフトタッチスイッチには4つの足があり、少し紛らわしいものです。



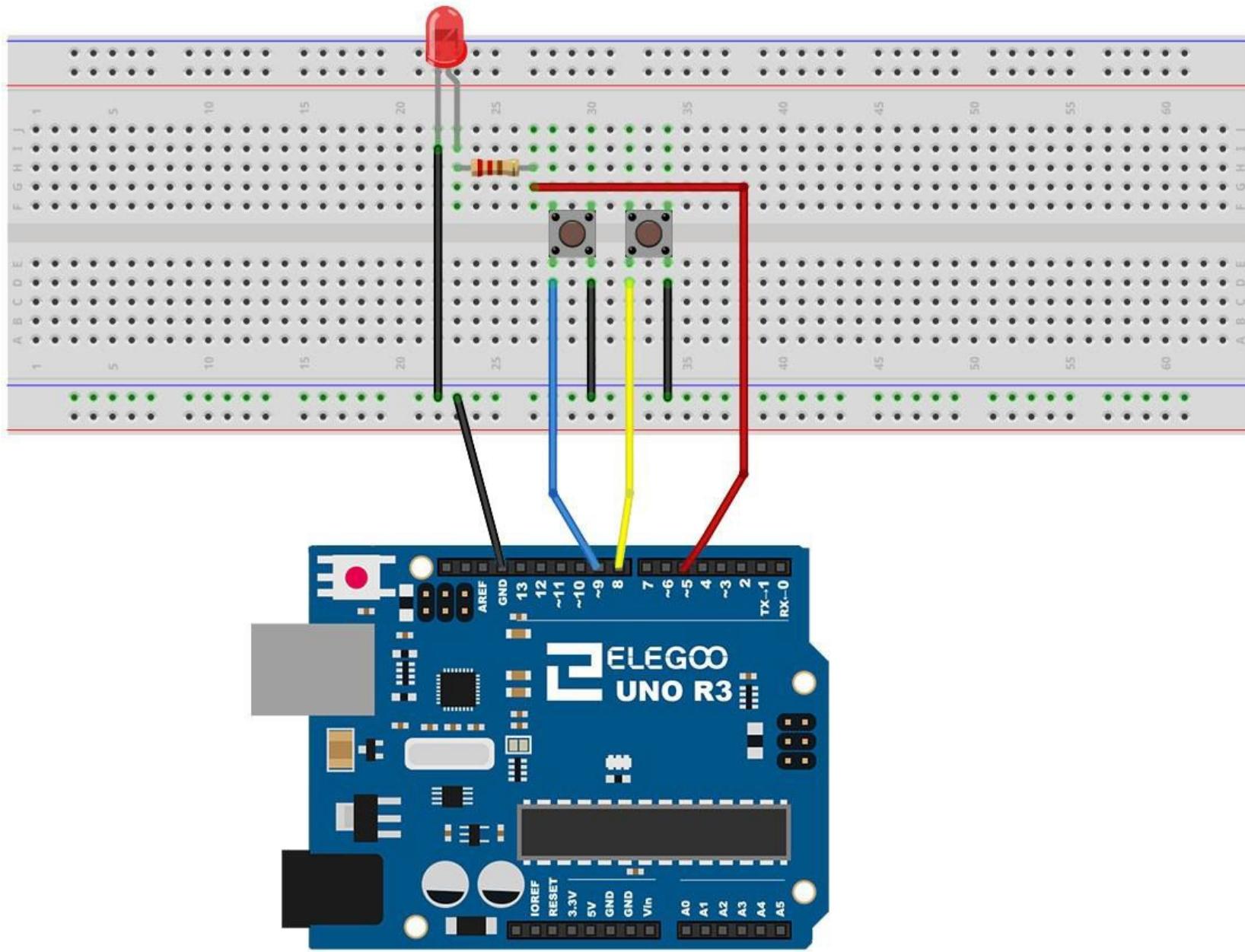
実際には、実際には電気接続が2つしかありません。スイッチパッケージ内では、AとDのようにピンBとCと一緒に接続されています。

Connection

Schematic



Wiring diagram



スイッチの本体は正方形であるが、ピンはスイッチの両側から突出している。これは、ピンがブレッドボード上に正しく配置されているときにピンが離れていることを意味します。
LEDは、左に短い負のリード線を持っていなければならないことに注意してください。

Code

配線後、コードフォルダのレッスン5デジタル入力でプログラムを開き、UPLOADを押してプログラムをアップロードしてください。エラーが表示される場合は、プログラムのアップロードに関するチュートリアルの詳細については、レッスン2を参照してください。

あなたのUNOボードにスケッチをロードします。左ボタンを押すとLEDが点灯し、右ボタンを押すと消灯します。

スケッチの最初の部分は、使用される3つのピンの3つの変数を定義します。'ledPin'は出力ピンで、'buttonApin'はブレッドボードの上部に近いスイッチを指し、「buttonBpin」は他のスイッチを指します。

'setup'関数はledPinをOUTPUTとして通常のように定義しますが、ここでは2つの入力を処理します。この場合、pinModeを次のように'INPUT_PULLUP'に設定します：

```
pinMode(buttonApin, INPUT_PULLUP);
pinMode(buttonBpin, INPUT_PULLUP);
```

INPUT_PULLUPのピン・モードは、ピンを入力として使用することを意味しますが、他のものが入力に接続されていない場合は、「プルアップ」してHIGHにする必要があります。言い換えると、入力のデフォルト値は、ボタンを押す動作によってLOWに引かれない限り、HIGHです。

このため、スイッチはGNDに接続されています。スイッチを押すと、入力ピンがGNDに接続されるため、ハイになりません。

入力が通常HIGHで、ボタンを押したときに入力がLOWになるので、ロジックは少し上下逆です。

これを'loop'関数で扱います。

```
void loop()
{
    if (digitalRead(buttonApin) == LOW)
    {
        digitalWrite(ledPin, HIGH);
    }
    if (digitalRead(buttonBpin) == LOW)
```

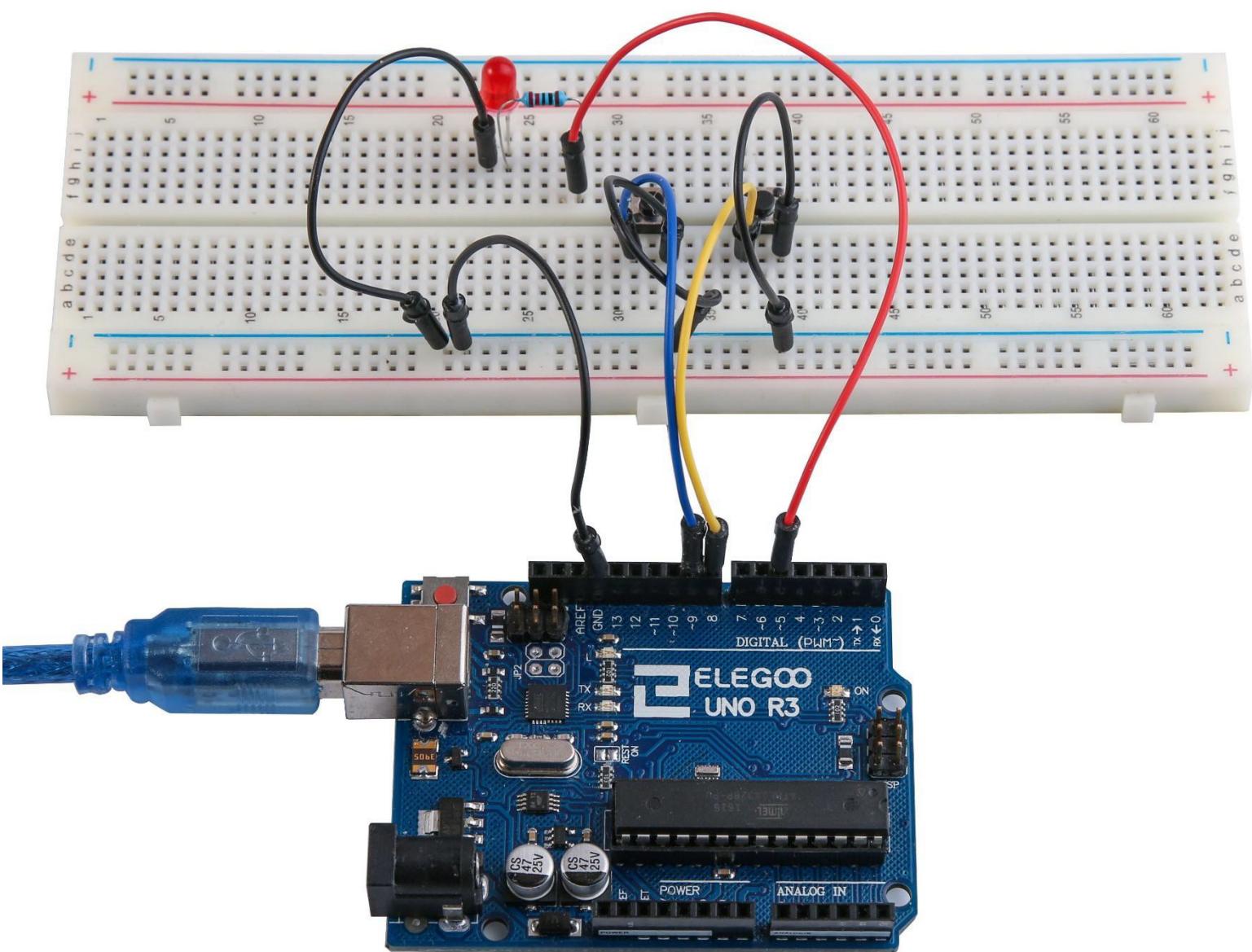
```
{  
    digitalWrite(ledPin, LOW);  
}  
}  
}
```

'loop'関数には2つの'if'ステートメントがあります。各ボタンに1つ。それぞれが適切な入力で「digitalRead」を行います。

ボタンが押されると、対応する入力はLOWになり、ボタンAが低ければ、ledPinの'digitalWrite'がオンになることを覚えておいてください。

同様に、ボタンBを押すと、LOWがledPinに書き込まれます。

Example picture



Lesson 6 アクティブブザー

概要

このレッスンでは、アクティブなブザーでサウンドを生成する方法を学習します。

必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x Active buzzer
- (2) x F-M wires (Female to Male DuPont wires)

部品の紹介

BUZZER:

電子ブザーは DC 電源であり、集積回路を備えています。 それらは、コンピュータ、プリンタ、複写機、アラーム、電子玩具、車載電子機器、電話機、タイマ、および音声デバイス用の他の電子製品に広く使用されている。 ブザーはアクティブなものとパッシブなものに分類できます。 2 つのブザーのピンを上に向けます。 緑色の回路基板を持つものは受動的なブザーですが、黒色のテープで囲まれたものはアクティブなブザーです。

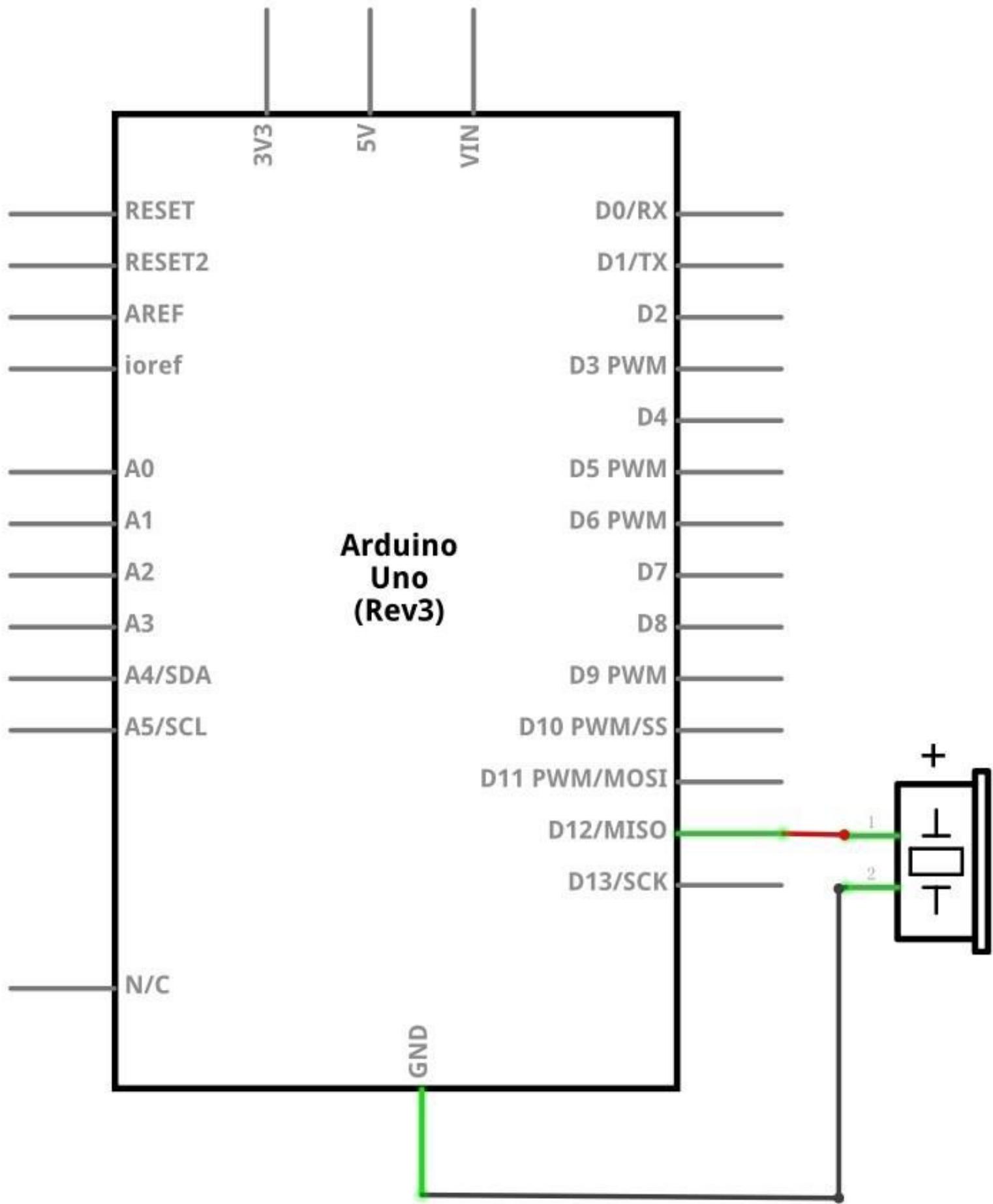
この 2 つの違いは、アクティブなブザーに内蔵振動があることです

音源があるので、電化すると音が鳴ります。 受動的なブザーはそのような音源を持っていないので、DC 信号が使われていればつぶれません。 代わりに、それを駆動するために周波数が 2K~5K の方形波を使用する必要があります。 アクティブなブザーは、複数の内蔵発振器のためにパッシブブザーよりしばしば高価です。

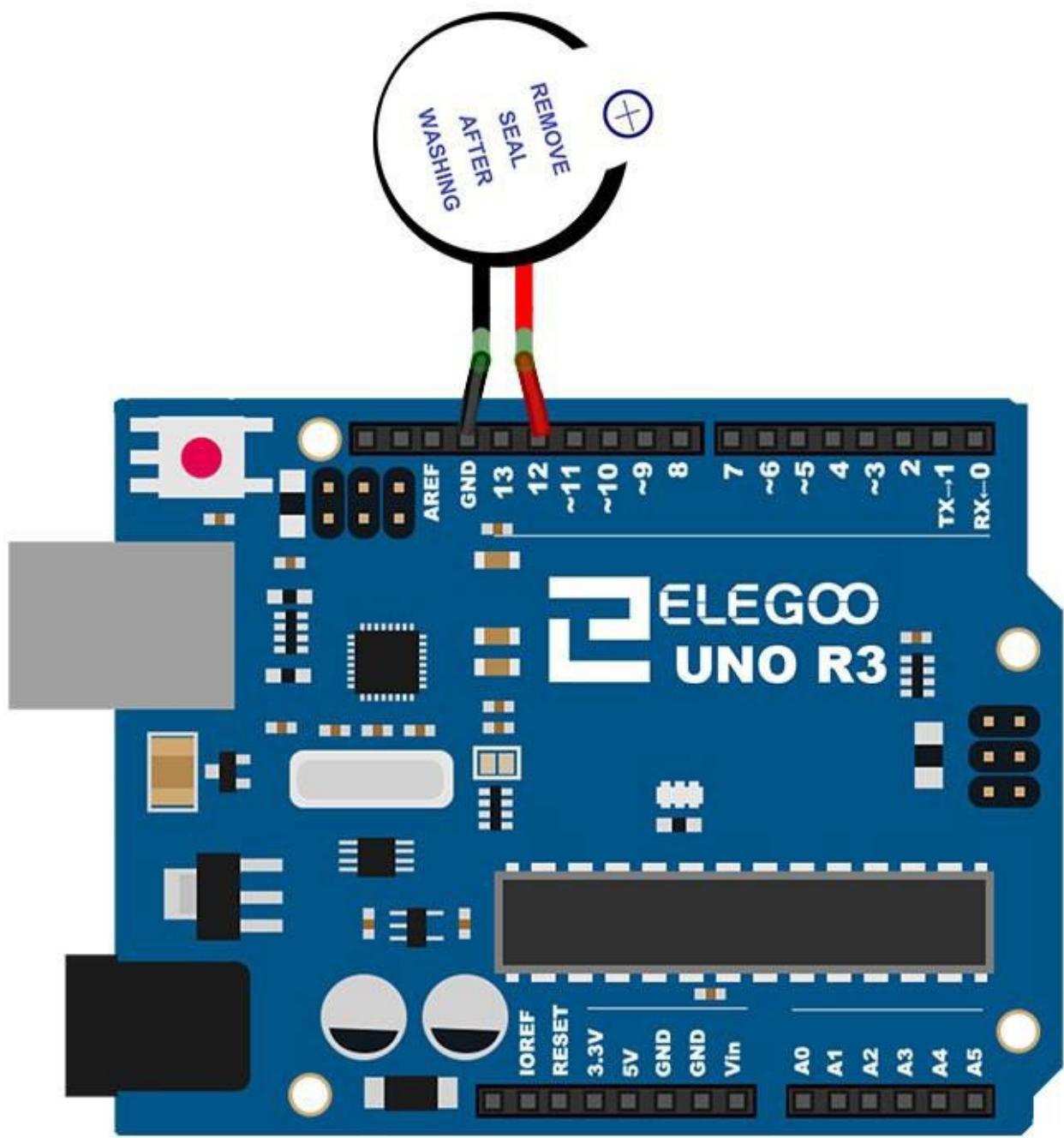


Connection

Schematic



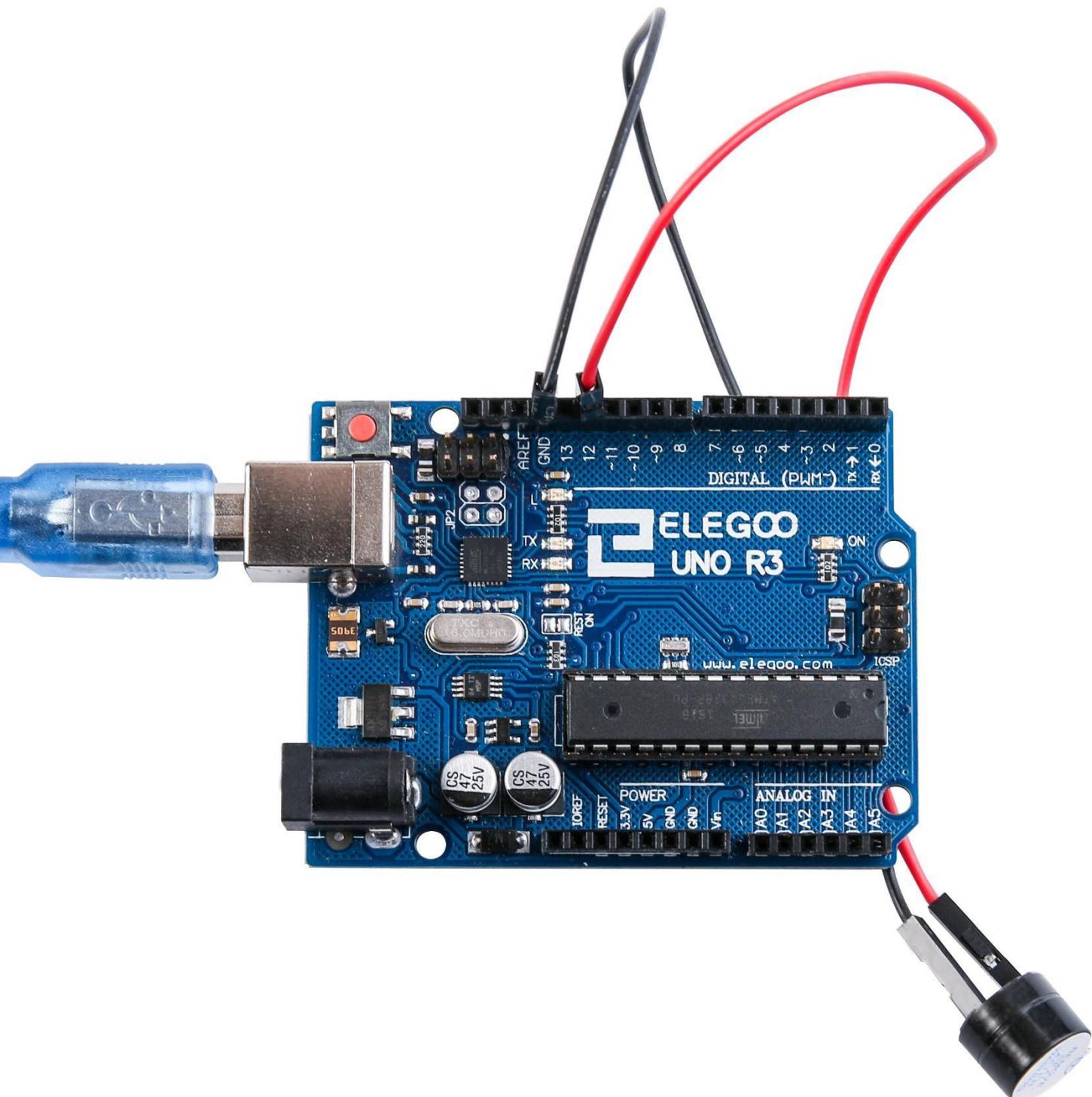
Wiring diagram



Code

配線後、コードフォルダのレッスン 6 のサウンドを作成し、UPLOAD をクリックしてプログラムをアップロードしてください。エラーがある場合のプログラムアップロードの詳細については、レッスン 2 を参照してください。

Example picture



Lesson 7 受動ブザー

概要

このレッスンでは、受動ブザーを使用する方法を学習します。

実験の目的は、アルト・ド (523Hz) 、 Re (587Hz) 、 Mi (659Hz) 、 Fa (698Hz) 、 So (784Hz) 、 La (880Hz) 、 Si (988Hz) から Treble Do (1047Hz) 。

必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x Passive buzzer
- (2) x F-M wires (Female to Male DuPont wires)

部品の紹介

Passive Buzzer:

受動ブザーの作動原理は、空気を振動させるために PWM 生成オーディオを使用することである。 振動周波数がある限り適切に変更され、異なる音を発生させることができます。 たとえば、523Hz のパルスを送信すると、Alto Do、587Hz のパルス、ミッドレンジの Re、659Hz のパルスを生成することができます。

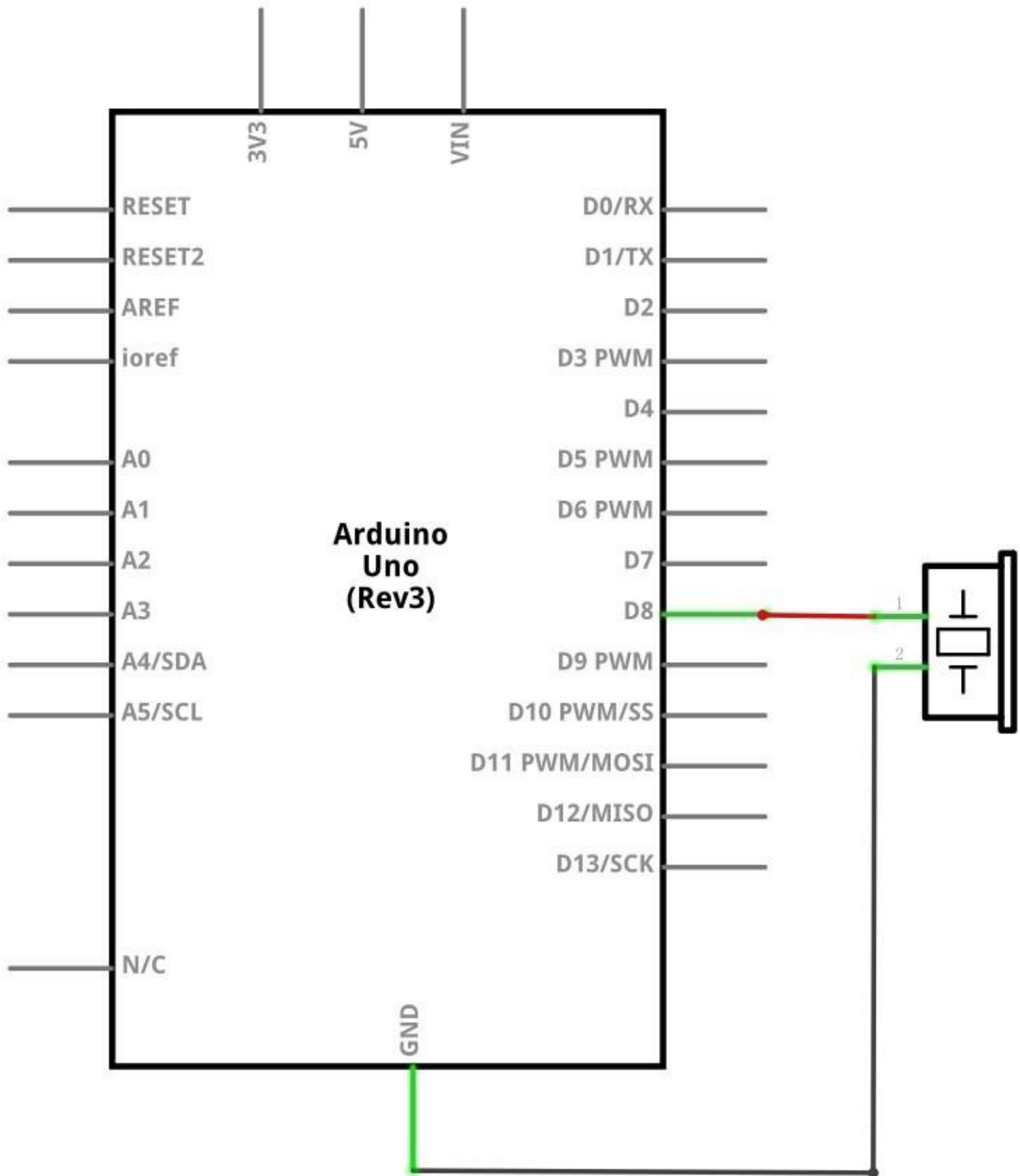
ミッドレンジミー。 ブザーで曲を演奏できます。

アナログ Write () のパルス出力が固定されているため、UNO R3 ボードのアナログ Write () 機能を使用してブザーにパルスを発生させないように注意してください
(500Hz) 。

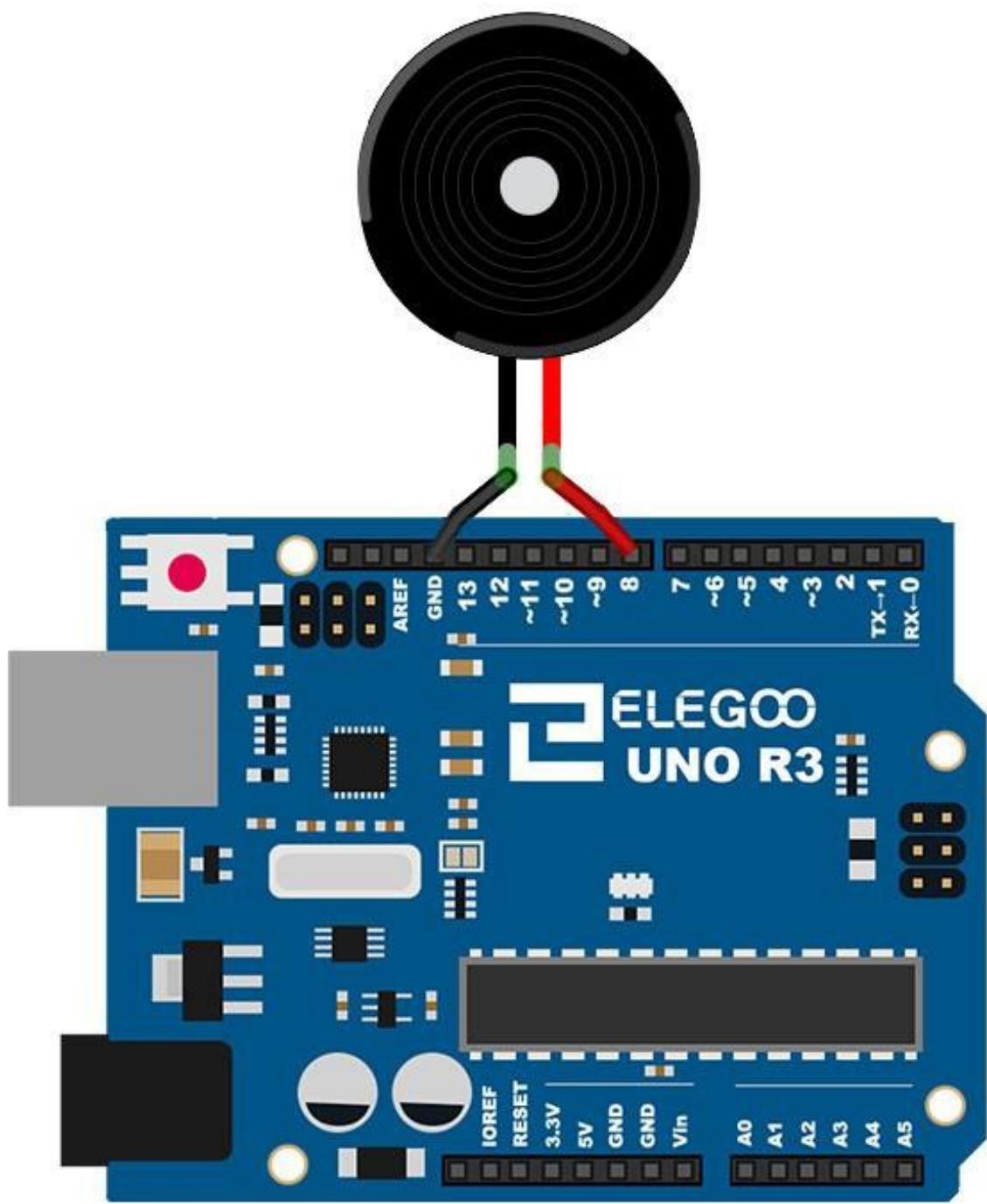


Connection

Schematic



Wiring diagram



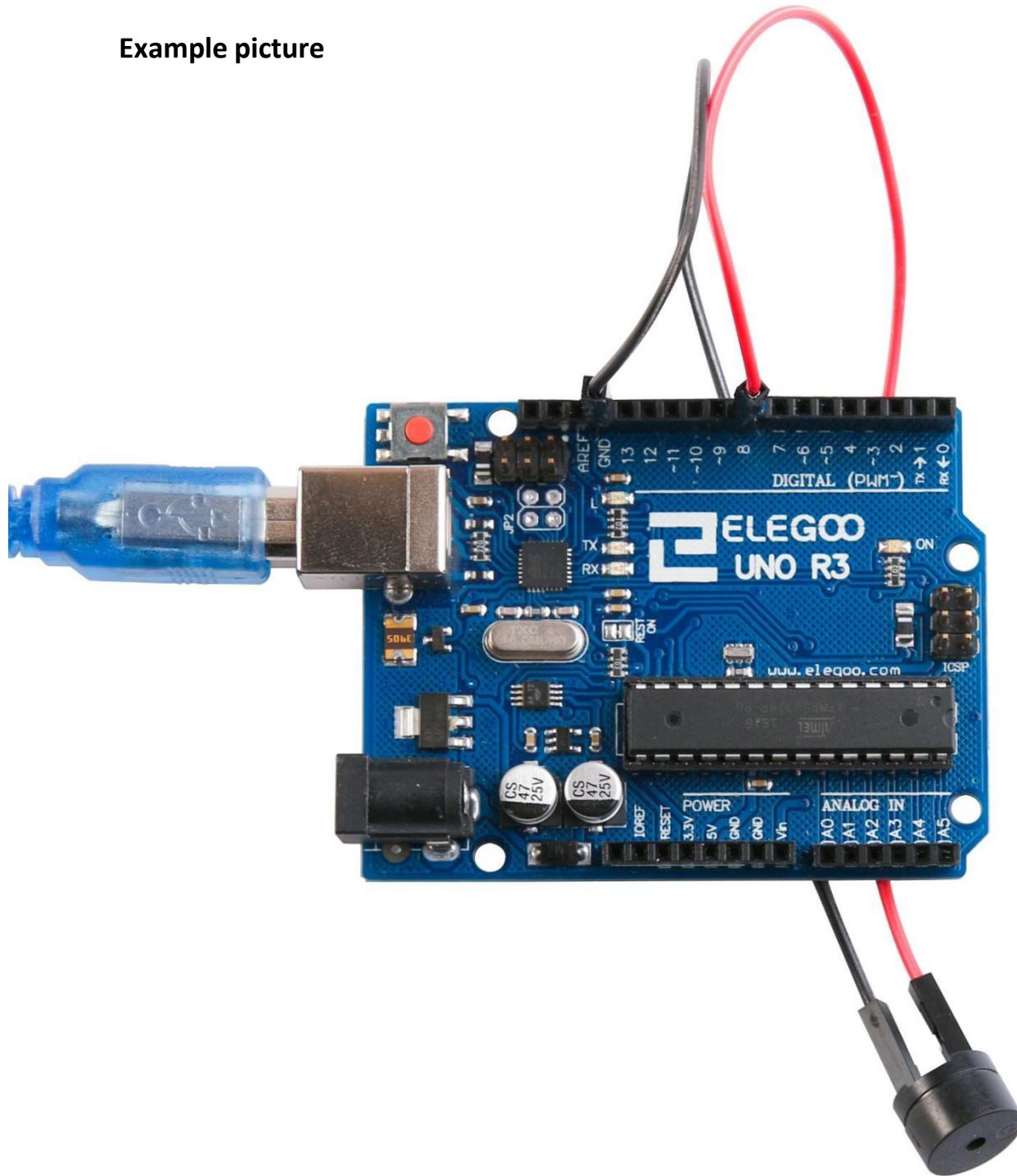
UNO R3 ボードに接続されているブザーを、赤色（正極性）のピン 8 に黒色線（負）を GND に配線します。

Code

配線後、コードフォルダ - レッスン 7 パッシブブザーでプログラムを開き、アップロードをクリックしてプログラムをアップロードしてください。 エラーがある場合のプログラムアップロードの詳細については、レッスン 2 を参照してください。

これを実行する前に、必要に応じて<pitchches>ライブラリがインストールされていることを確認するか、再インストールしてください。 そうしないと、コードが機能しません。
ライブラリファイルのロードの詳細については、レッスン 1 を参照してください。

Example picture



Lesson 8 傾斜ボールスイッチ

概要

このレッスンでは、小さな傾きを検出するためにチルトボールスイッチを使用する方法を学習します。

必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x Tilt Ball switch
- (2) x F-M wires (Female to Male DuPont wires)



部品の紹介

Tilt sensor:

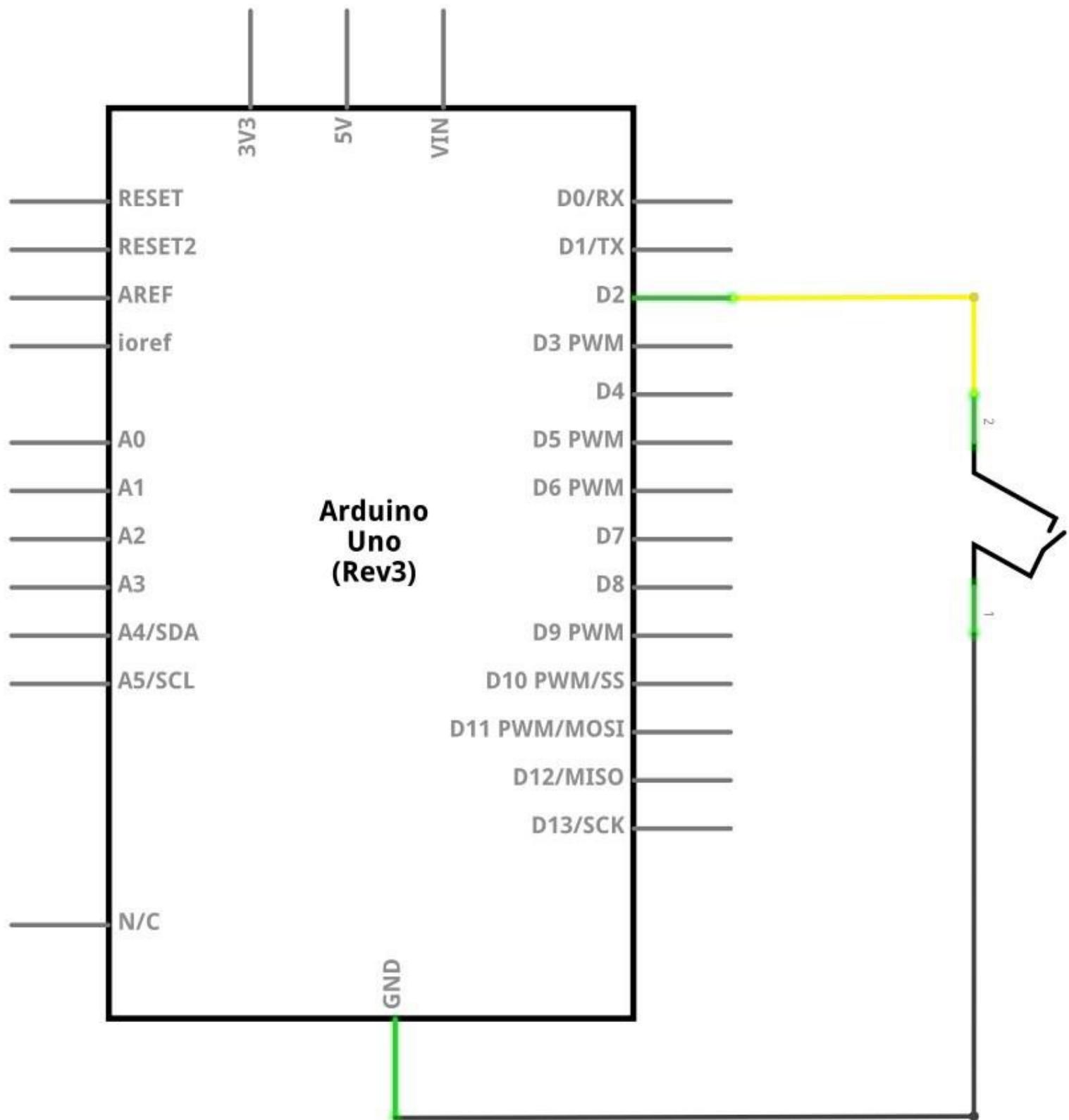
傾斜センサ（傾斜ボールスイッチ）を使用すると、向きや傾きを検出できます。それらは小さく、安価で、低電力で使いやすいものです。正しく使用すると、磨耗しません。彼らのシンプルさは、おもちゃ、ガジェット、家電製品で人気があります。明らかな理由から、「水銀スイッチ」、「傾斜スイッチ」または「ローリングボールセンサー」と呼ばれることがあります。

それらは通常、何らかの種類の空洞（円筒形が一般的ではありますが、必ずしも一般的ではありません）の中に、水銀や転がり球のような導電性の自由質量を内包しています。キャビティの一端は2つの導電性要素（極）を有する。センサがその端部が下方になるように配向されると、質量は極に転がり、それらを短絡させ、スイッチスローとして作用する。

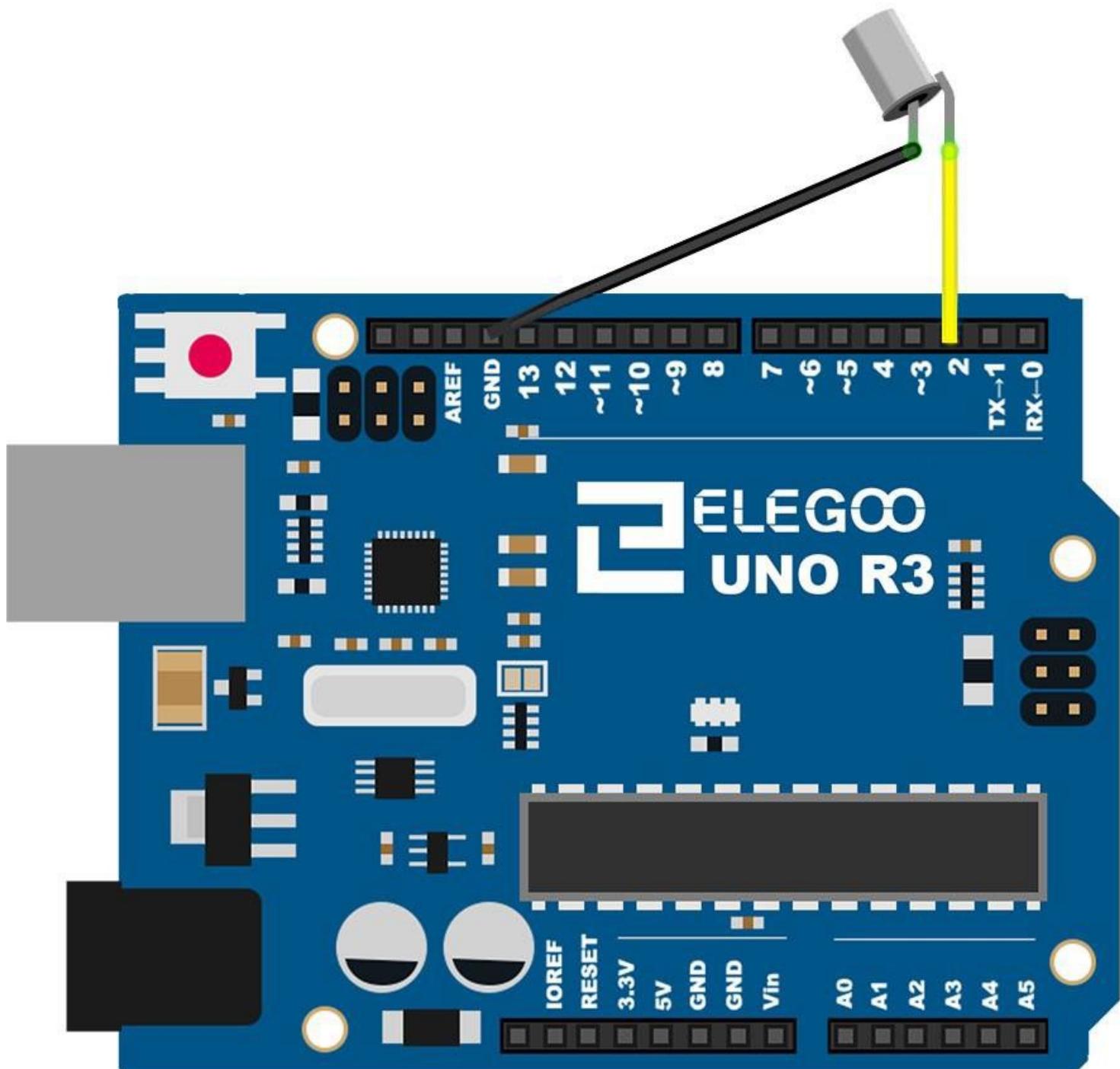
完全な加速度計と同じくらい正確で柔軟性はありませんが、傾斜スイッチは動きや方向を検出できます。もう1つの利点は、大きなものが自分の力を切り替えることができるということです。一方、加速度計は、デジタルまたはアナログ電圧を出力し、余分な回路を使用して解析する必要があります。

Connection

Schematic



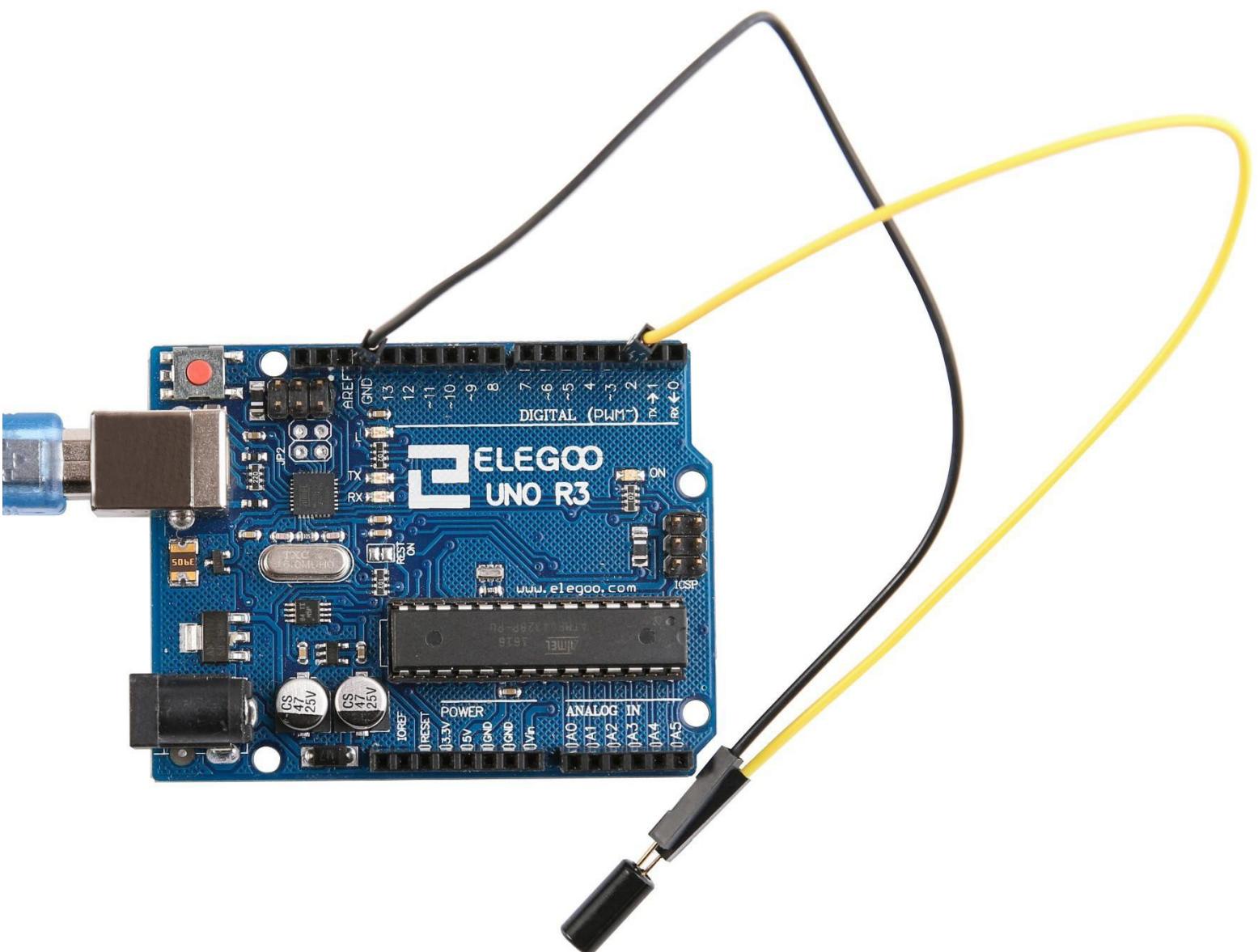
Wiring diagram



Code

配線後、コードフォルダのレッスン 8 ポールスイッチでプログラムを開き、UPLOAD をクリックしてプログラムをアップロードしてください。 エラーがある場合のプログラムアップロードの詳細については、レッスン 2 を参照してください。

Example picture



Lesson 9 サーボ

概要

サーボは 180 度回転することができるギヤードモータの一種です。 UNO R3 ボードから電気パルスを送信することによって制御されます。これらのパルスは、サーボがどの位置に移動すべきかをサーボに指示する。 Servo には 3 本のワイヤーがあり、茶色のワイヤーはグランドワイヤーで、UNO の GND ポートに接続する必要があります。赤色のワイヤーは電源ワイヤーで、5v ポートに接続し、オレンジ色のワイヤーは信号ワイヤー Dig #9 ポートに接続する必要があります。

必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x Servo (SG90)
- (3) x M-M wires (Male to Male jumperwires)

部品の紹介

SG90

- Universal for JR and FP connector
- Cable length : 25cm
- No load; Operating speed: 0.12 sec / 60 degree (4.8V), 0.10 sec / 60 degree (6.0V)
- Stall torque (4.8V): 1.6kg/cm
- Temperature : -30~60'C
- Dead band width: 5us
- Working voltage: 3.5~6V

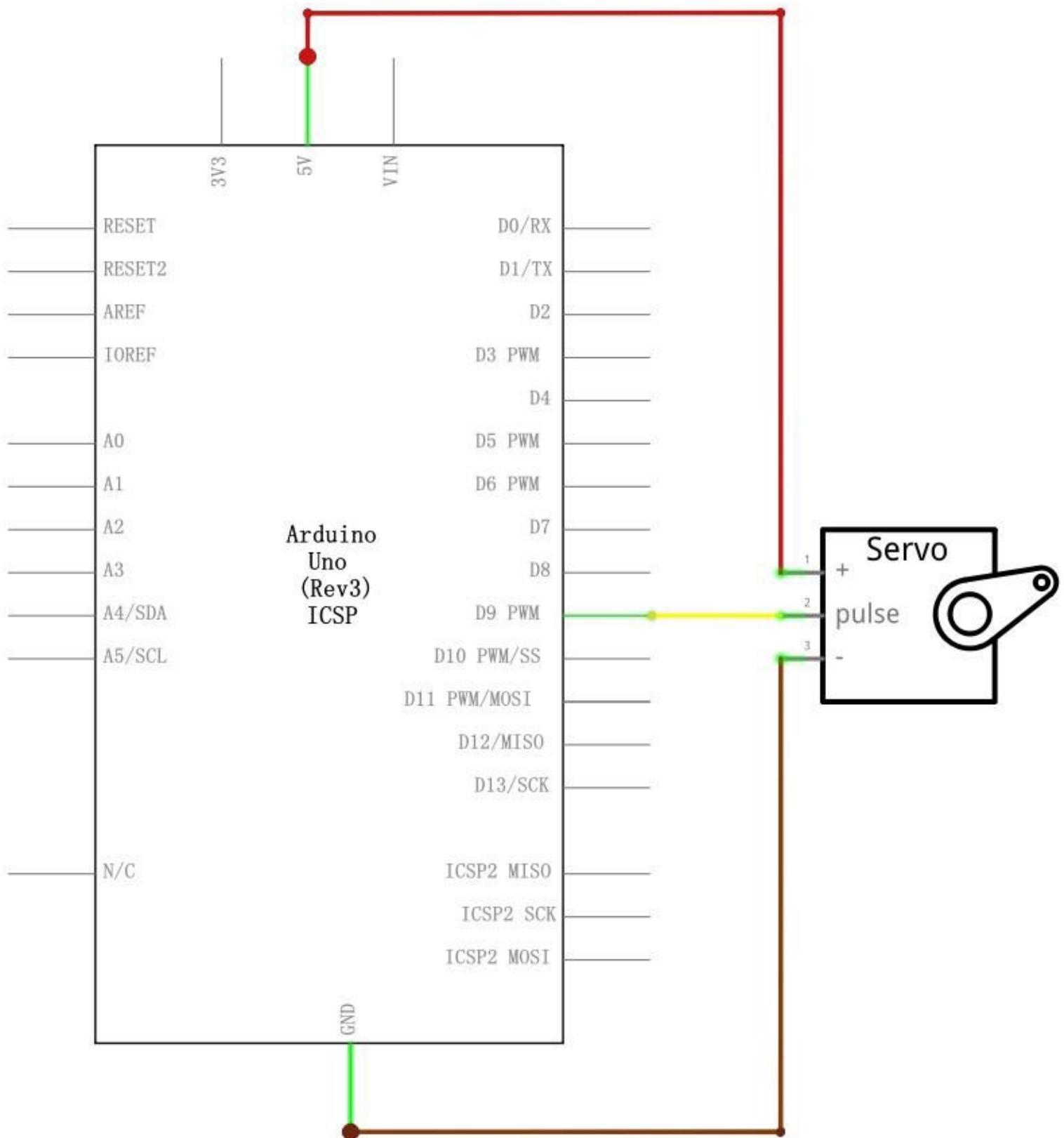
- Dimension : 1.26 in x 1.18 in x 0.47 in (3.2 cm x 3 cm x 1.2 cm)
- Weight : 4.73 oz (134 g)



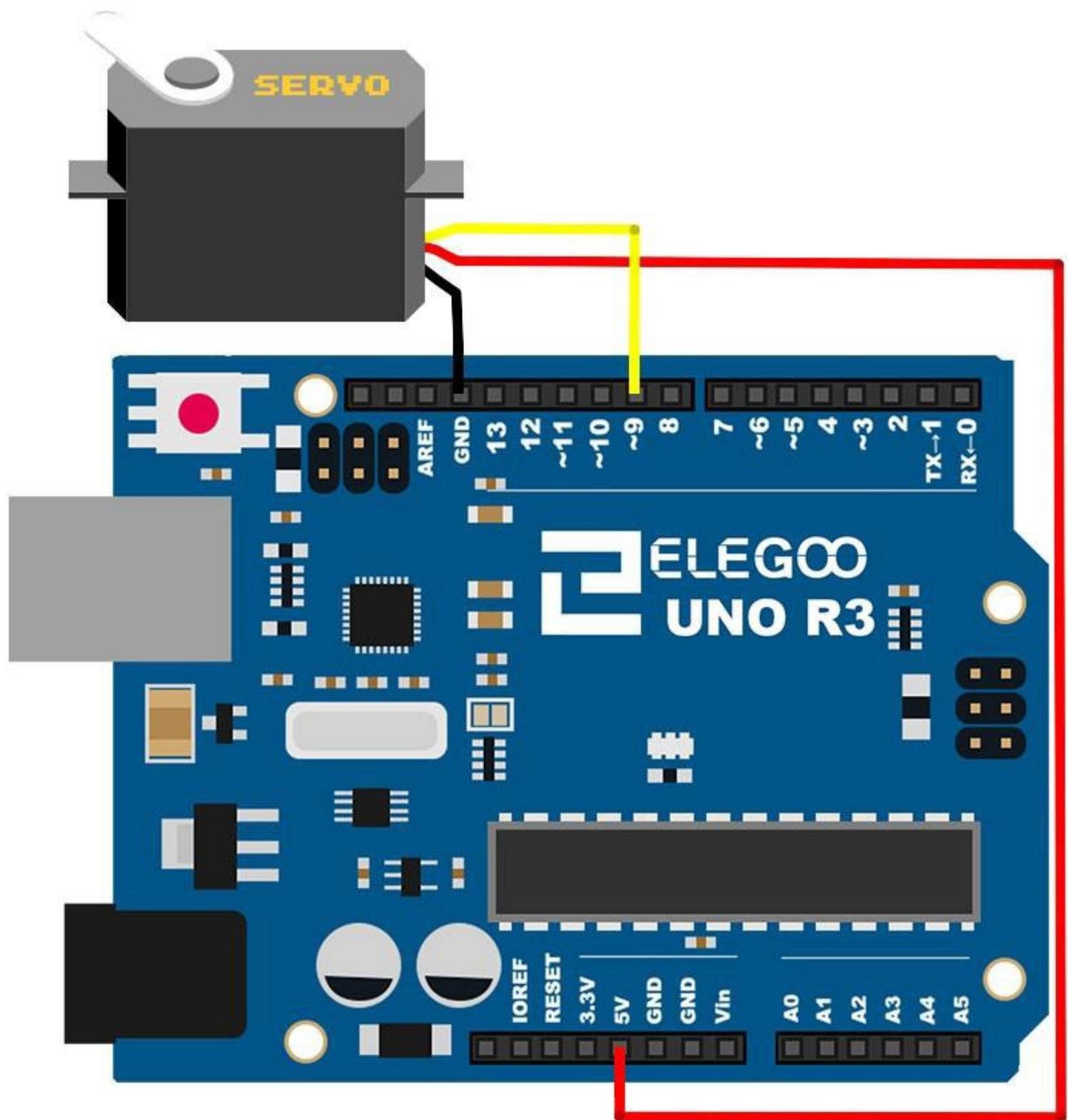
76 / 223

Connection

Schematic



Wiring diagram



Code

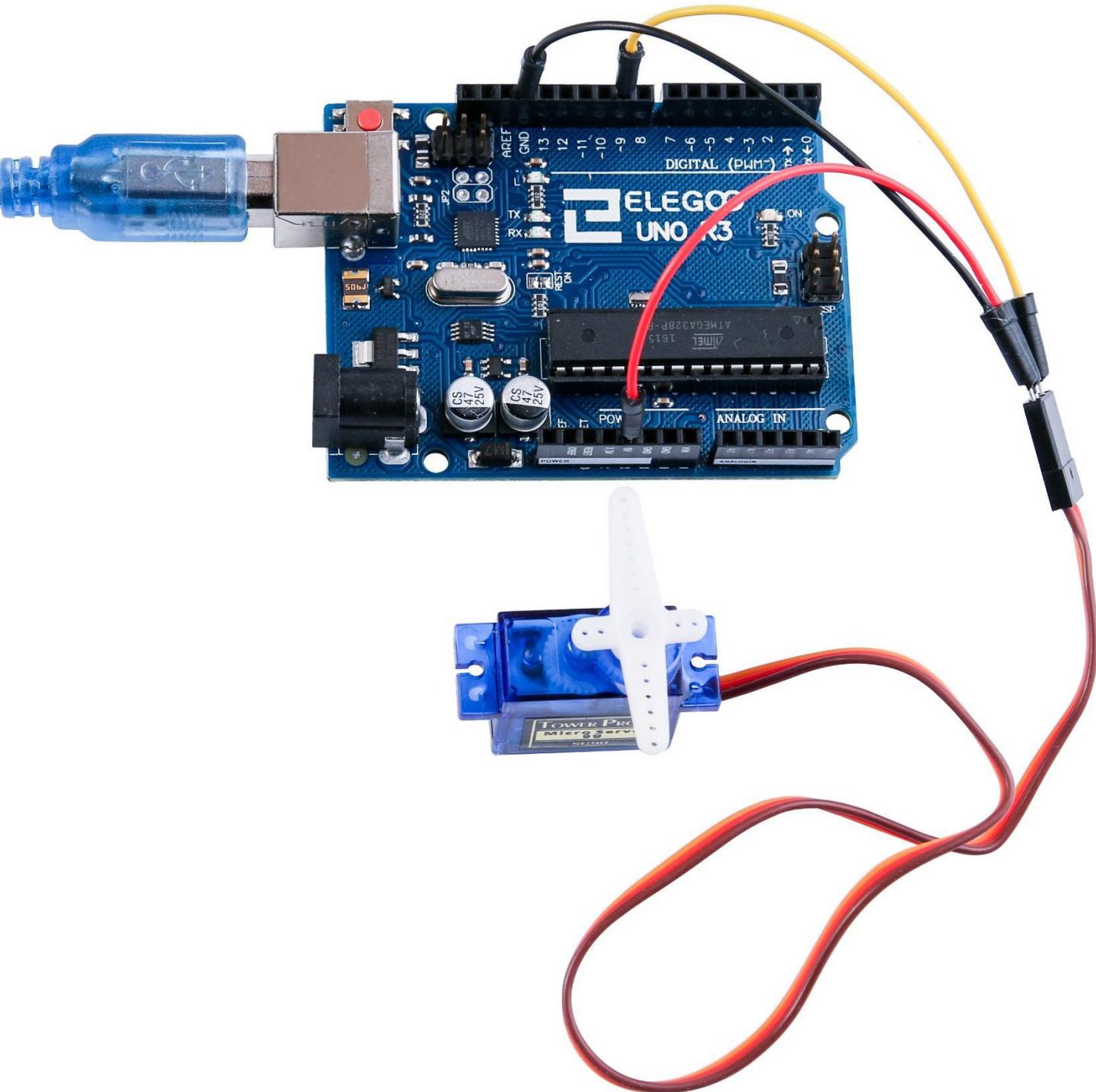
配線後、コードフォルダの Lesson 9 Servo でプログラムを開き、UPLOAD をクリックしてプログラムをアップロードしてください。エラーがある場合のプログラムアップロードの詳細については、レッスン 2 を参照してください。

これを実行する前に、<Servo>ライブラリがインストールされていることを確認するか、必要に応じて再インストールしてください。そうしないと、コードが機能しません。

ライブラリファイルのロードの詳細については、レッスン 1 を参照してください。

Example picture

写真では、サーボの茶色の線は、黒の M-M 線を介して接続され、赤の線は、赤の M-M 線を介して接続され、オレンジ色の線は、黄 M-M 線。

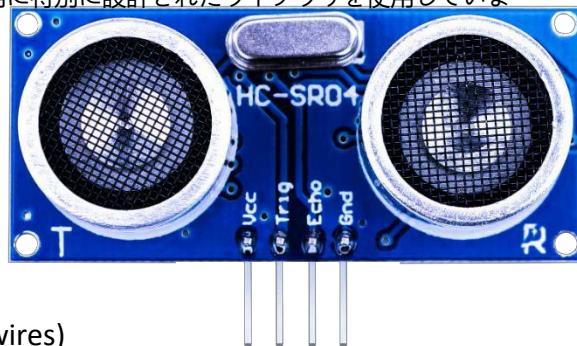


Lesson 10 超音波センサモジュール

概要

超音波センサーは、距離測定が必要なあらゆる種類のプロジェクトに適しており、例として障害物を避けることができる。

HC-SR04 は安価で使いやすく、これらのセンサ用に特別に設計されたライブラリを使用しています。



必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x Ultrasonic sensor module
- (4) x F-M wires (Female to Male DuPontwires)

部品の紹介

Ultrasonic sensor

超音波センサモジュール HC-SR04 は 2cm~400cm の非接触測定機能を提供し、測距精度は 3mm です。 モジュールには、超音波送信機、受信機および制御回路が含まれる。仕事の基本原則:

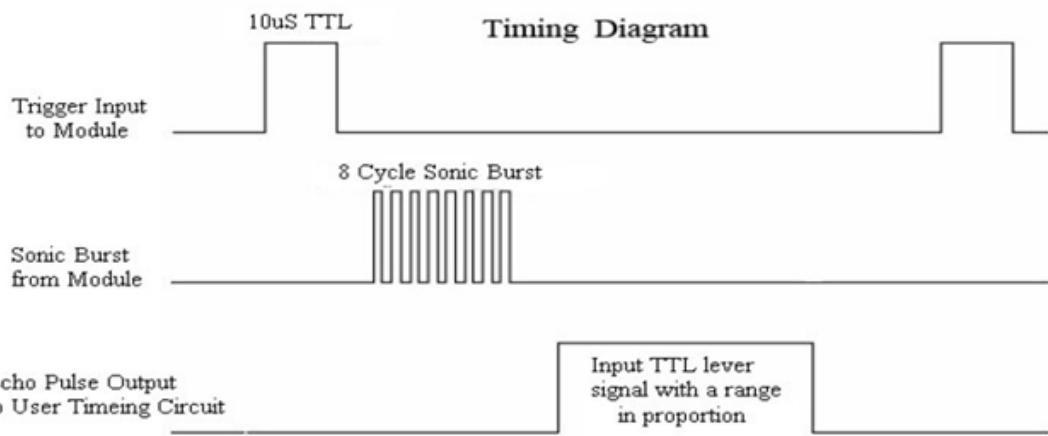
- (1) IO トリガを 10us 以上のハイレベル信号に使用すると、
- (2) モジュールは 40 kHz を自動的に 8 回送信し、パルス信号が戻ってきたかどうかを検出します。
- (3) 信号が戻ってくると、ハイレベルを通して、高出力 IO 持続時間は、超音波を送信してから戻るまでの時間である。

$$\text{試験距離} = (\text{高レベル時間} \times \text{音速 } (340\text{m / s}) / 2$$

以下にタイミング図を示します。 レンジングを開始するためにトリガ入力に短い 10us パルスを供給するだけで、モジュールは 40kHz で 8 サイクルの超音波バーストを送り、エコーを上げます。

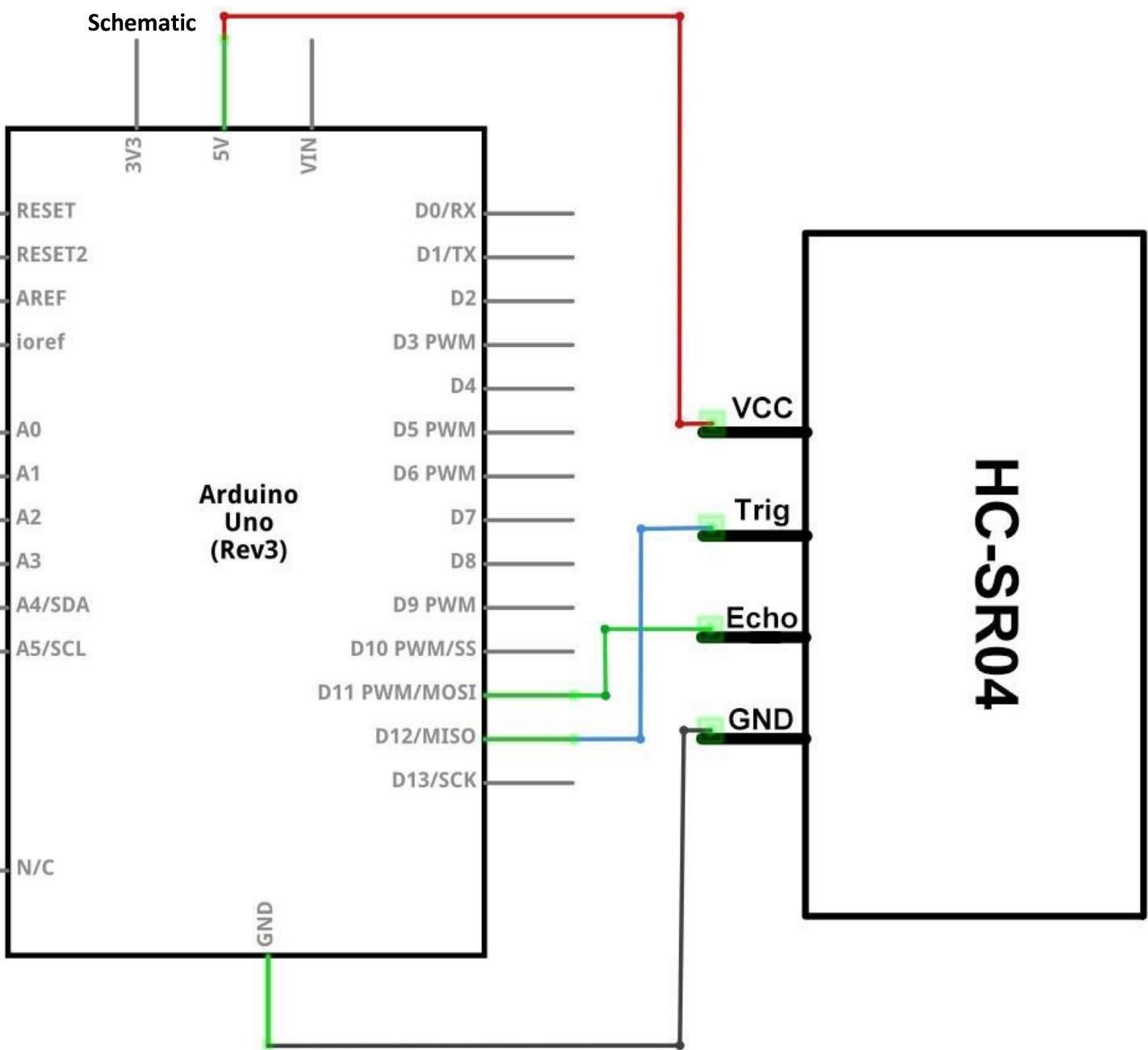
エコーは、パルス幅と距離範囲の距離オブジェクトです。 トリガー信号を送信してからエコー信号を受信するまでの時間間隔で範囲を計算できます。

数式: $\text{us} / 58 = \text{センチメートル}$ または $\text{us} / 148 = \text{インチ}$; または: 範囲 = 高レベル時間 * 速度 (340M / S) / 2; エコー信号へのトリガ信号を防止するために、60ms 以上の測定サイクルを使用することを推奨します。

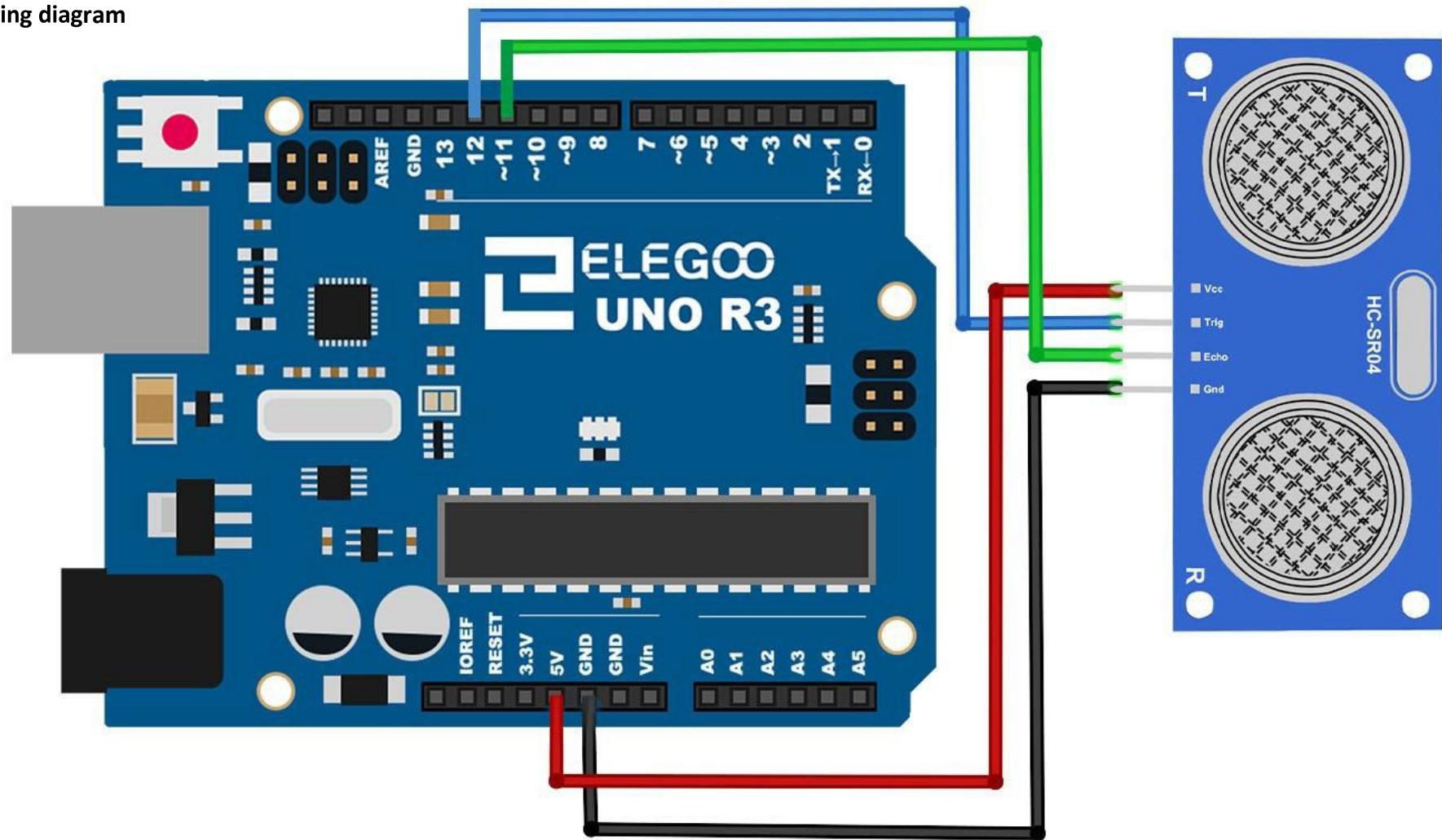


Connection

Schematic



Wiring diagram



Code

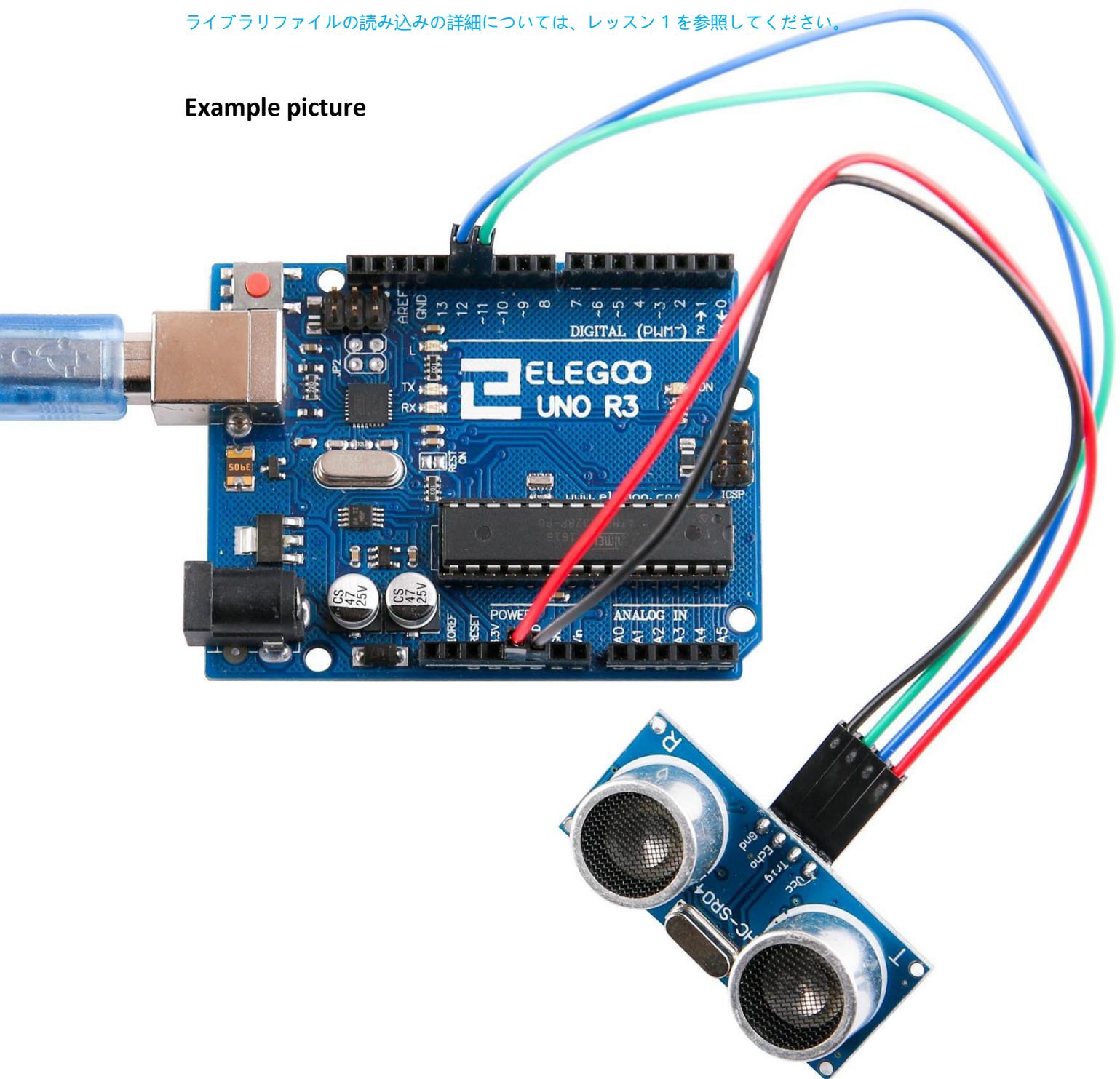
これらのセンサー用に設計されたライブラリを使用すると、コードが短く簡単になります。コードの始めにライブラリを組み込み、簡単なコマンドを使用してセンサの動作を制御することができます。

グラムをアップロードしてください。エラーがある場合のプログラムアップロードの詳細については、[レッスン 2](#) を参照してください。

これを実行する前に、<HC-SR04>ライブラリがインストールされていることを確認するか、必要に応じて再インストールしてください。そうしないと、コードが機能しません。

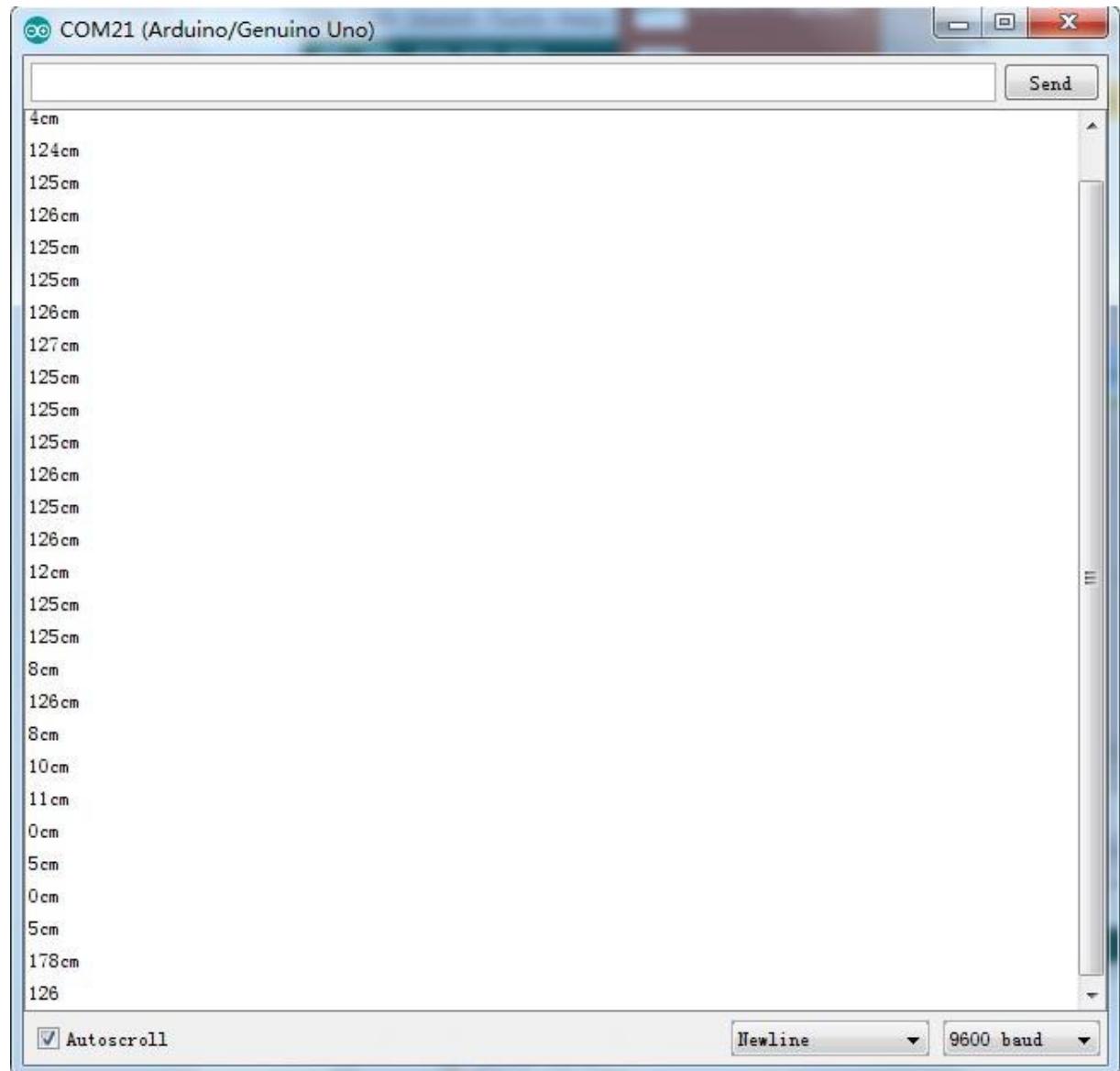
ライブラリファイルの読み込みの詳細については、[レッスン 1](#) を参照してください。

Example picture



モニタを開くと、データが壊れているのがわかります:

「シリアルモニタ」ボタンをクリックしてシリアルモニタをオンにします。 シリアルモニタの基本については、レッスン 1 で詳しく説明しています。



Lesson 11 DHT11 温度湿度センサー

概要

このチュートリアルでは、DHT11 温度湿度センサーの使用方法を学習します。 湿度と温度の読み取り値を把握する必要があるほとんどのプロジェクトでは十分正確な計測ができます。

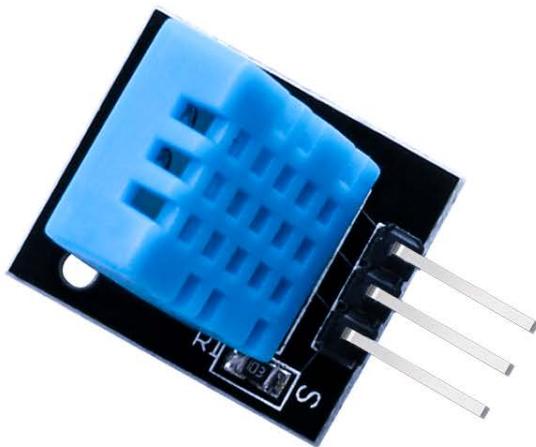
繰り返しますが、これらのセンサー専用に設計されたライブラリを使用して、コードを短く簡単に作成します。

必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x DHT11 Temperature and Humidity module
- (3) x F-M wires (Female to Male DuPontwires)

部品の紹介

Temp and humidity sensor:



DHT11 デジタル温湿度センサーは、温度と湿度の較正されたデジタル信号出力を含む複合センサーです。専用のデジタルモジュール収集技術と温湿度感知技術を適用し、高い信頼性と優れた長期安定性を保証します。

このセンサには、湿性成分と NTC 温度測定デバイスの抵抗感知機能が含まれ、高性能 8 ビットマイクロコントローラと接続されています。

アプリケーション: HVAC、除湿器、テストおよび検査装置、消費財、自動車、自動制御、データロガー、気象ステーション、家電、湿度レギュレーター、医療およびその他の湿度測定および制御

製品パラメータ

Relative humidity:

Resolution: 16Bit

Repeatability: $\pm 1\%$ RH

Accuracy: At 25°C $\pm 5\%$ RH

Interchangeability: fully interchangeable

Response time: 1 / e (63%) of 25°C 6s

1m / s air 6s

Hysteresis: $<\pm 0.3\%$ RH

Long-term stability: $<\pm 0.5\%$ RH / yr in

Temperature:

Resolution: 16Bit

Repeatability: $\pm 0.2^{\circ}\text{C}$

Range: At 25°C $\pm 2^{\circ}\text{C}$

Response time: 1 / e (63%) 10s

Electrical Characteristics

Power supply: DC3.5 ~ 5.5V

Supply Current: measurement 0.3mA standby 60μA

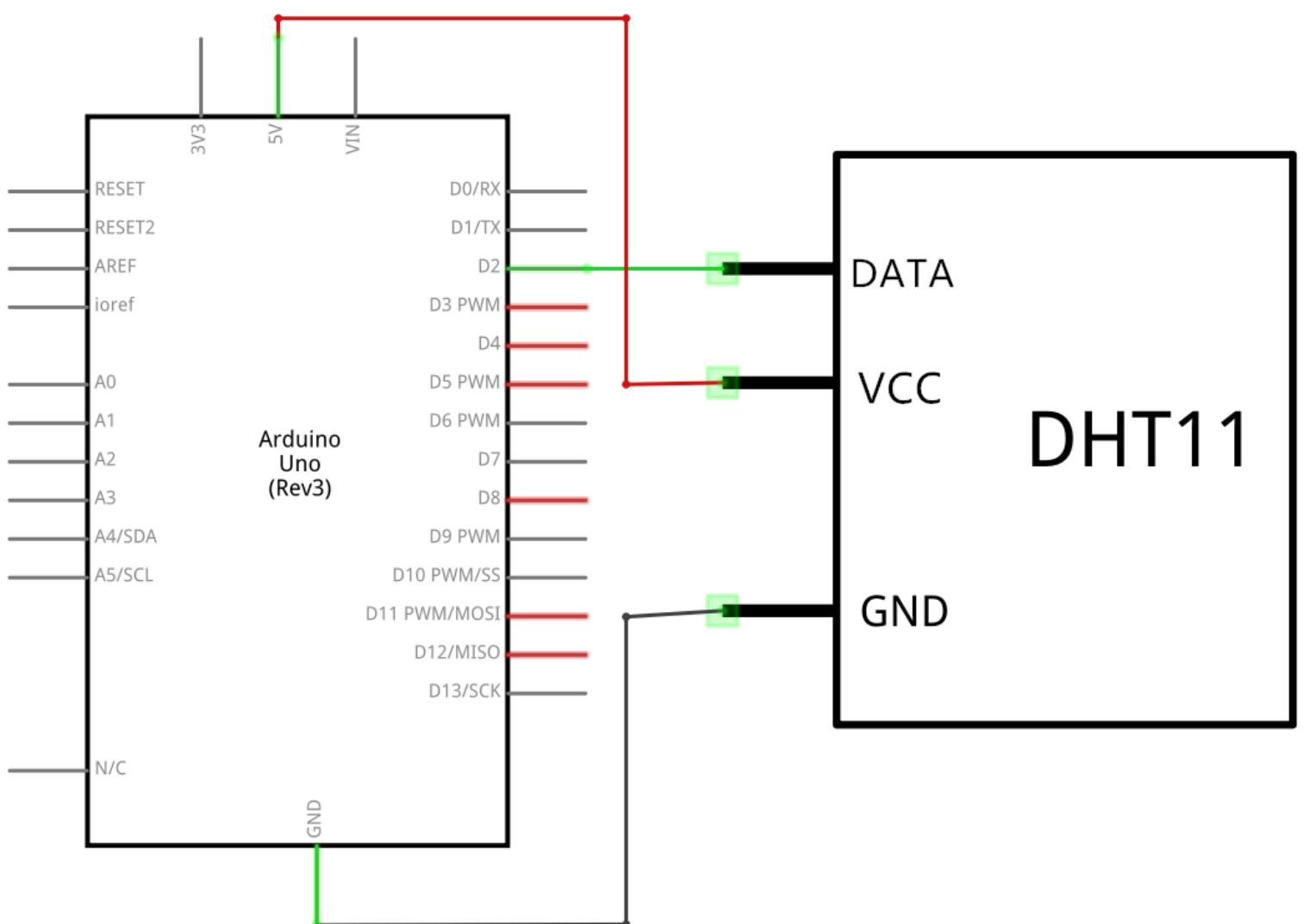
Sampling period: more than 2 seconds

Pin Description:

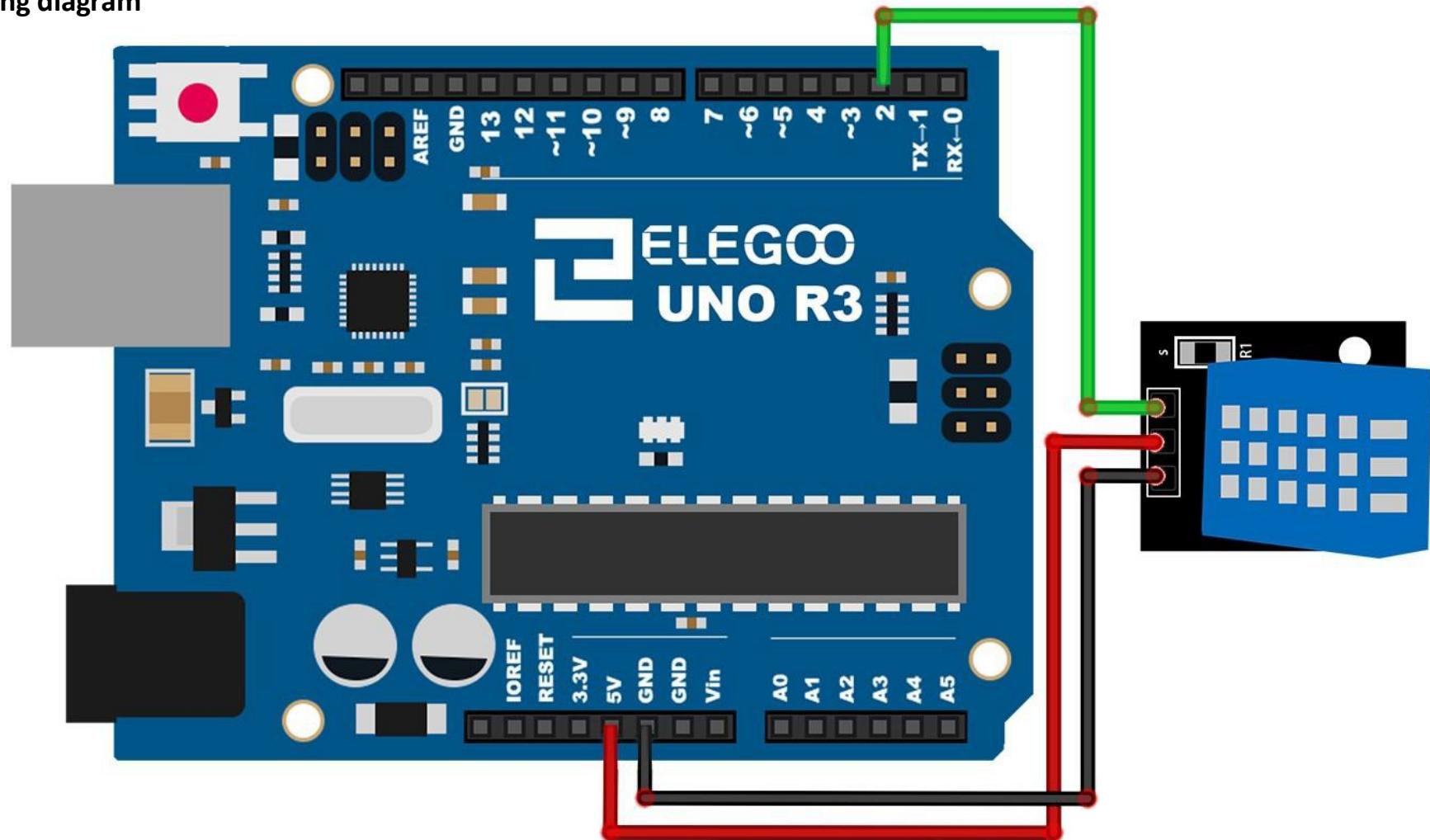
1. the VDD power supply 3.5 ~ 5.5V DC
2. DATA serial data, a single bus
3. NC, empty pin
4. GND ground, the negative power

Connection

Schematic



Wiring diagram



見てのよう、センサーの1つだけが使用されていないので、センサーへの接続が3つ必要です。

接続は、UNOの任意のピンに接続できる電圧、グランド、信号です。

Code

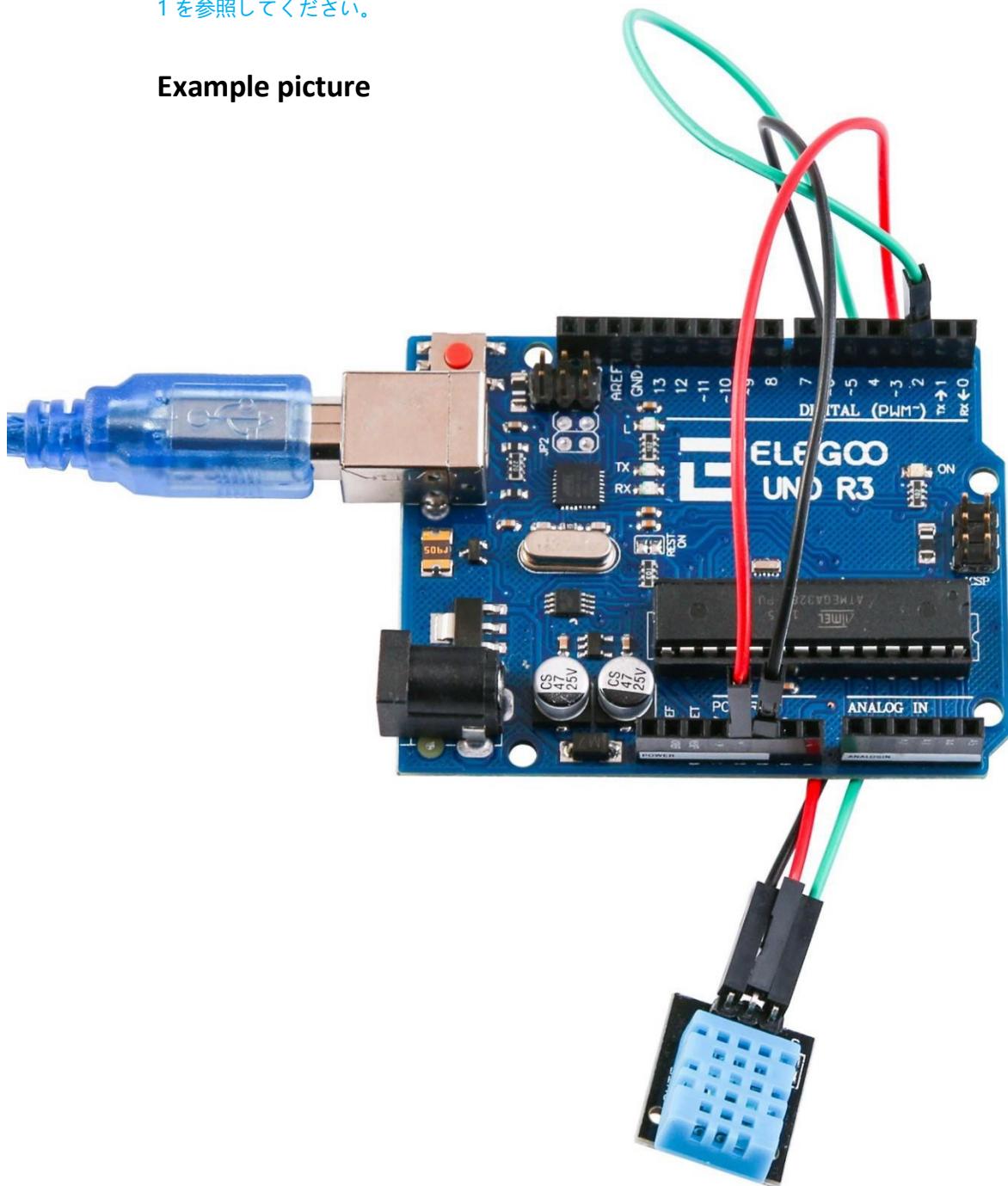
配線後、コードフォルダのレッスン12 DHT11 温度湿度センサーでプログラムを開き、UPLOADをクリックしてプログラムをアップロードしてください。エラーがある場合のプログラムアップロードの詳細については、レッスン2を参照してください。

これを実行する前に、<SimpleDHT>ライブラリがインストールされていることを確認してください

必要に応じて再インストールしてください。そうしないと、コードが機能しません。

ライブラリファイルの読み込みに関するチュートリアルの詳細については、レッスン1を参照してください。

Example picture



プログラムをアップしてモニタを開くと、以下のようなデータが表示されます:(環境の温度が示されて、22 度であることがわかります)
[シリルモニター]ボタンをクリックしてシリルモニターをオンにします。シリルモニターの基本については、レッスン 1 で詳しく説明しています。

```
Sample DHT11...
Sample RAW Bits: 0011 0011 0000 0000 0001 0110 0000 0000 0100 1001
Sample OK: 22 *C, 51 %
=====
Sample DHT11...
Sample RAW Bits: 0011 0011 0000 0000 0001 0110 0000 0000 0100 1001
Sample OK: 22 *C, 51 %
=====
Sample DHT11...
Sample RAW Bits: 0011 0011 0000 0000 0001 0110 0000 0000 0100 1001
Sample OK: 22 *C, 51 %
=====
Sample DHT11...
Sample RAW Bits: 0011 0010 0000 0000 0001 0111 0000 0000 0100 1001
Sample OK: 23 *C, 50 %
=====
Sample DHT11...
Sample RAW Bits: 0011 0011 0000 0000 0001 0110 0000 0000 0100 1001
Sample OK: 22 *C, 51 %
=====
Sample DHT11...
Sample RAW Bits: 0011 0011 0000 0000 0001 0110 0000 0000 0100 1001
Sample OK: 22 *C, 51 %
=====
```

Autoscroll Newline 9600 baud

Lesson 12 アナログジョイスティックモジュール

概要

アナログジョイスティックは、プロジェクトにいくつかのコントロールを追加するのに最適な方法です。このチュートリアルでは、アナログジョイスティックモジュールの使用方法を学習します。

必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x Joystick module
- (5) x F-M wires (Female to Male DuPontwires)

部品の紹介

Joystick

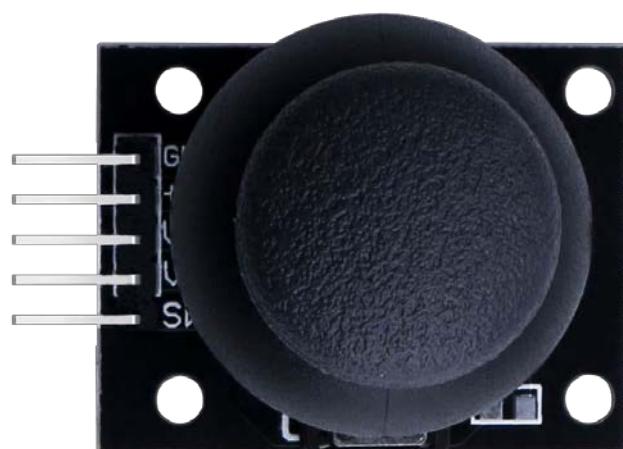
モジュールにはVCC、グランド、X、Y、キーの5つのピンがあります。モジュールを入手した場所によって、あなたのラベルが若干異なる場合があります。親指の棒はアナログであり、単純な「指向的な」ジョイスティックよりも正確な読み取りを提供する必要があります。さらに、ジョイスティックを押し下げるときにプッシュボタンをアクティブにすることができます。

X / Yピンからデータを読み出すにはArduinoのpinを使用し、ボタンを読むにはデジタルpinを使用する必要があります。Key pinは、ジョイスティックが押されるとグランドに接続され、そうでなければフローティングになります。

キー/セレクト・pinから安定した読み出しを得るには、プルアップ抵抗を介してVCCに接続する必要があります。

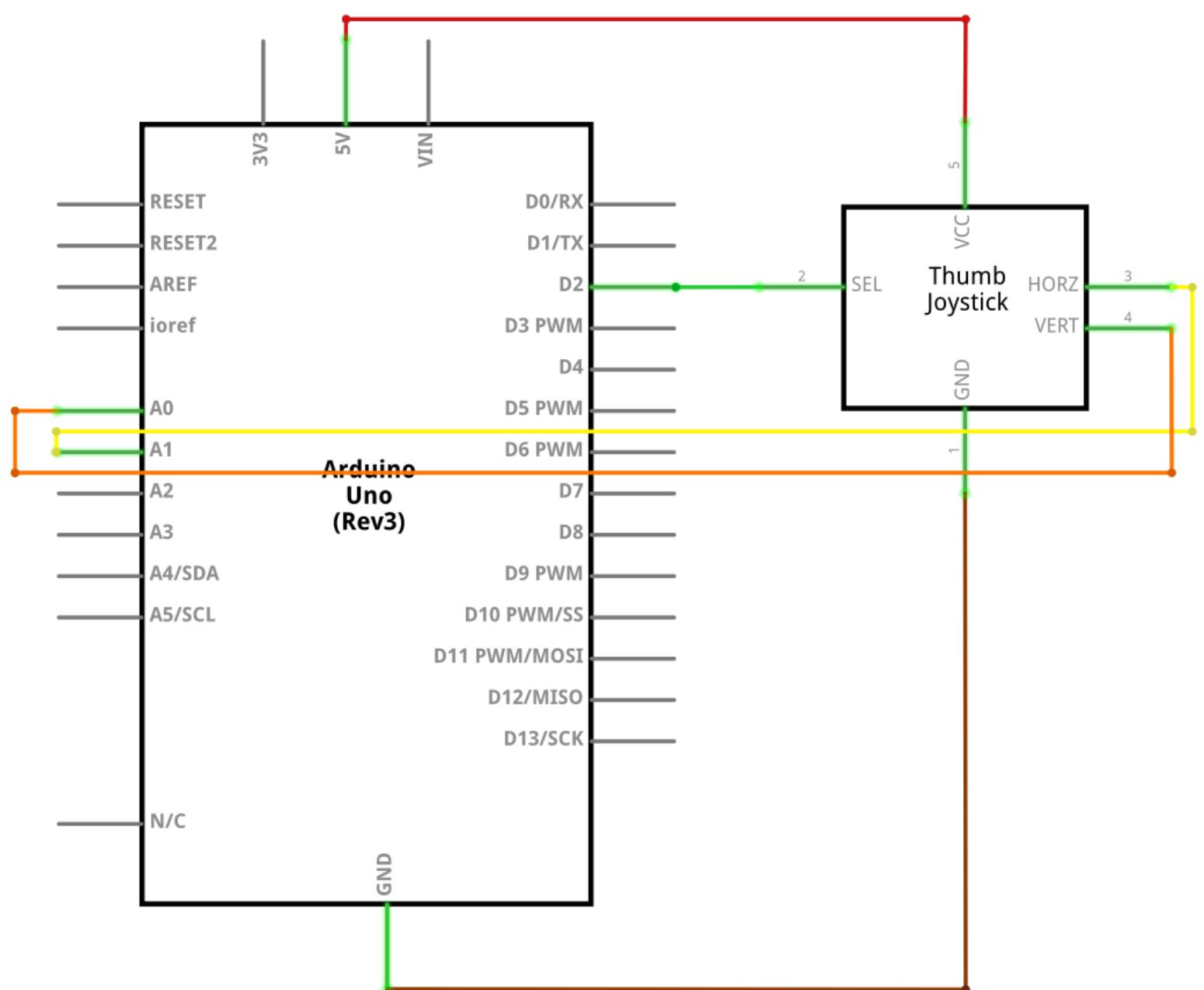
Arduinoデジタルpinに内蔵された抵抗を使用することができます。

Arduino pinのプルアップ抵抗をアクティブにする方法に関する学習

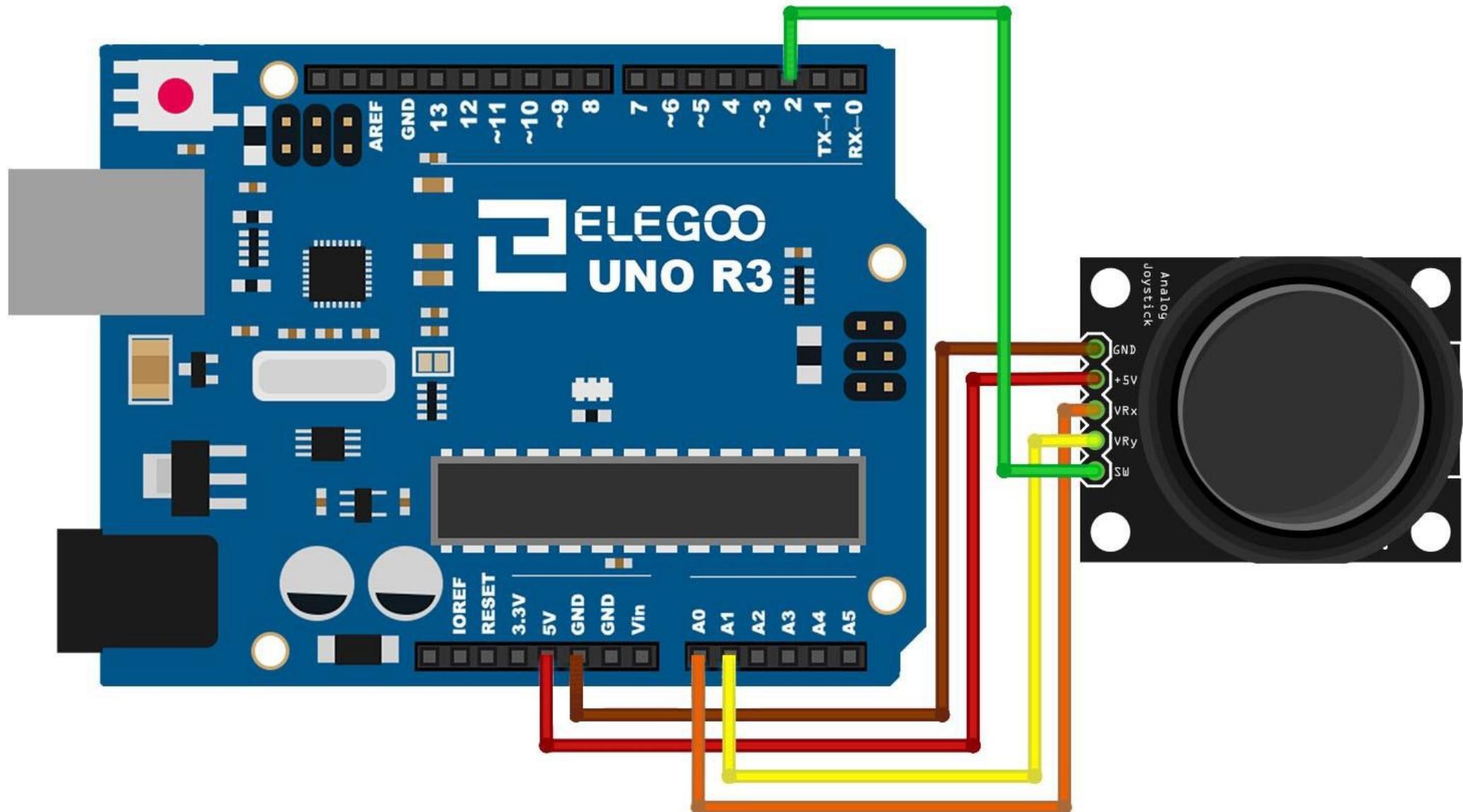


Connection

Schematic



Wiring diagram



ジョイスティックに 5 つの接続が必要です。

接続は、Key、Y、X、Voltage、Ground です。

"Y と X" はアナログ、"キー" はデジタルです。スイッチが不要な場合は、4 ピンしか使用できません。

Code

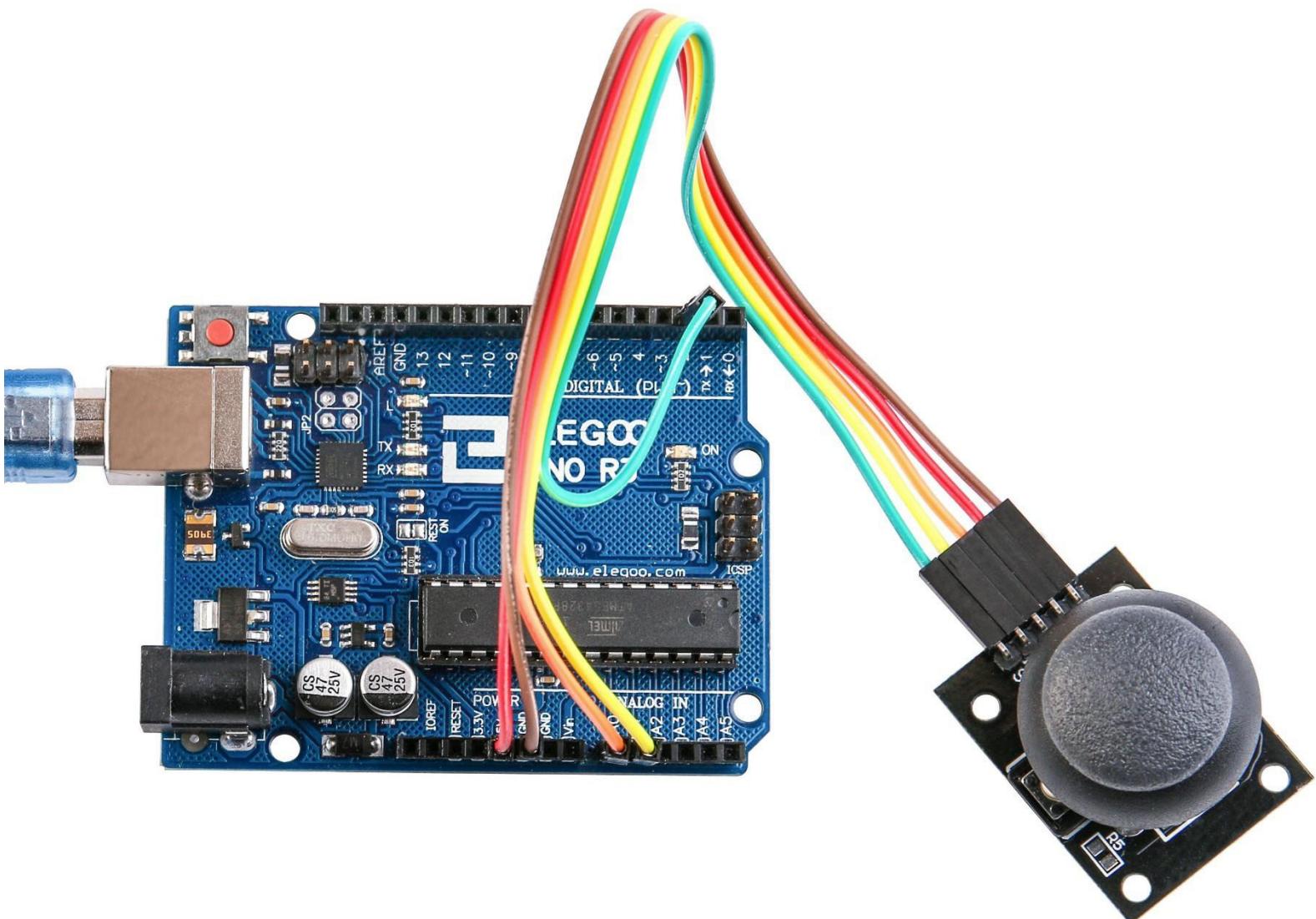
配線後、プログラムをコードフォルダ - レッサン 13 のアナログジョイスティックモジュールで開き、UPLOAD をクリックしてプログラムをアップロードしてください。エラーがある場合、プログラムのアップロードの詳細については、レッサン 2 を参照してください。

アナログジョイスティックは基本的にポテンショメータなので、アナログ値を返します。

ジョイスティックが静止位置または中立位置にあるとき、約 512 の値を戻すはずです。

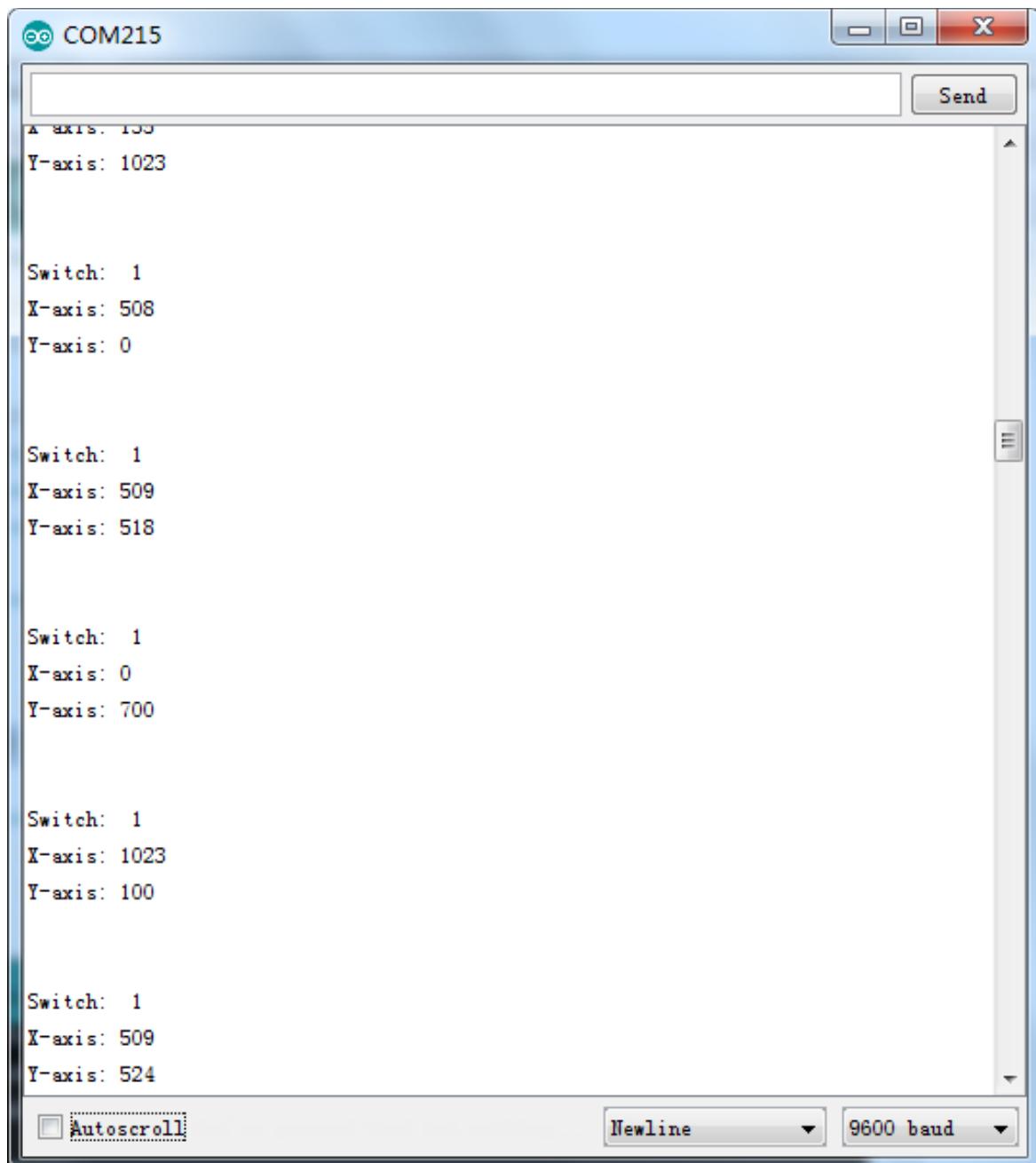
値の範囲は 0 から 1024 になります。

Example picture



モニタを開くと、データが壊れているのがわかります:

[シリアルモニタ]ボタンをクリックしてシリアルモニタをオンにします。 シリアルモニタの基本について
は、レッスン 1 で詳しく説明しています。



Lesson 13 赤外線受信モジュール

概要

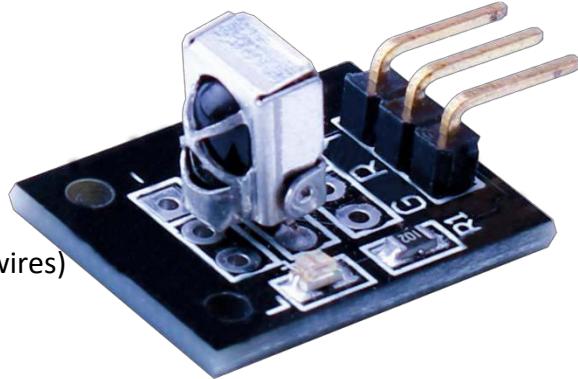
IR リモートを使用すると、プロジェクトをワイヤレスで制御することができます。

赤外線リモコンはシンプルで使いやすいです。このチュートリアルでは、IR 受信機を UNO に接続し、この特定のセンサー用に設計されたライブラリを使用します。

私たちのスケッチでは、このリモコンで利用できるすべての IR 16 進コードを取得し、コードが認識されたかどうかを検出し、キーを押している場合も検出します。

必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x IR receiver module
- (1) x IR remote
- (3) x F-M wires (Female to Male DuPontwires)



部品の紹介

IR RECEIVER SENSOR:

IR 検出器は、赤外線を受信するように調整されたフォトセルを備えたマイクロチップです。

ほとんどの場合、リモコンの検出に使用されます。テレビや DVD プレーヤーの前面には、クリッカーカーからの IR 信号を受信するためのものがあります。リモコンの内部には一致する IR LED があり、赤外線パルスを発してテレビの電源をオン、オフ、または変更するようにテレビに指示します。赤外光は人間の目には見えません。つまり、セットアップをテストするにはもう少し作業が必要です。

これらの間にはいくつかの違いがあり、CdS Photocells と言えます：

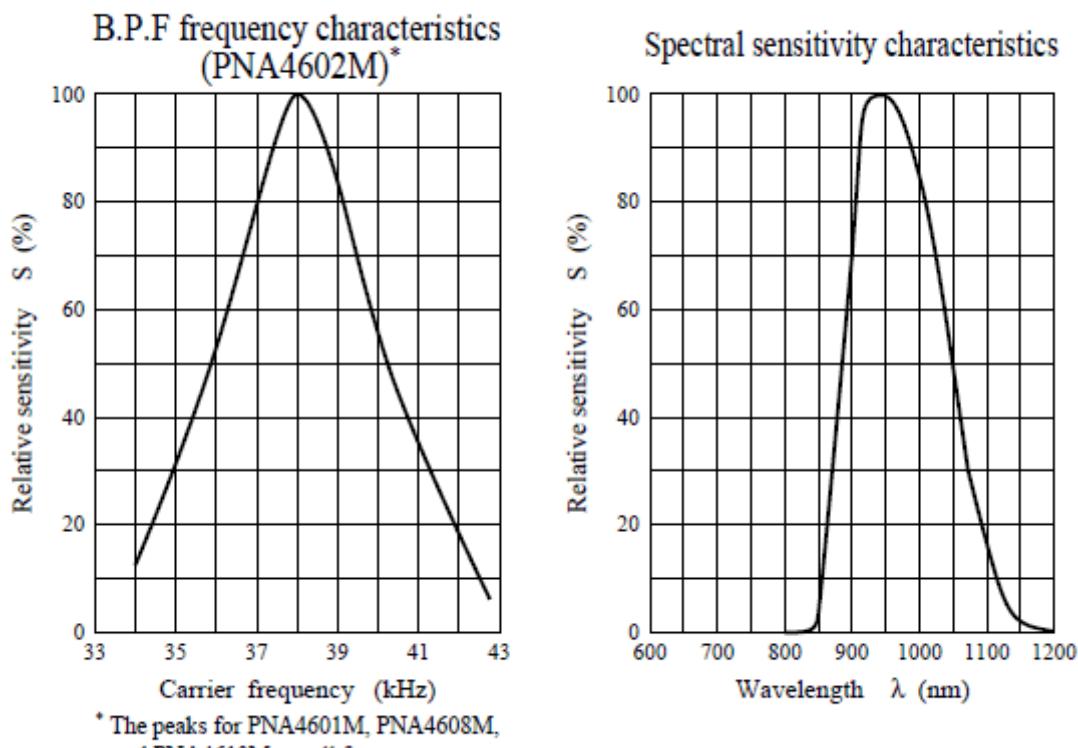
IR 検出器は IR 光のために特別にフィルタリングされ、可視光を検出するのには向きである。一方、光電池は、黄緑色の可視光を検出するのに優れており、IR 光は良好ではない。

IR 検出器の内部には、38KHz で変調された赤外線を探す復調器があります。IR LED を照らすだけでは、38KHz で PWM 点滅する必要があります。光電池は復調器を一切持たず、光電池の応答速度（約

1KHz) 内の任意の周波数 (DC を含む) を検出することができ、

IR 検出器はデジタル出力です。これらは 38KHz IR 信号を検出してロー (0V) を出力するか、いずれも検出せず、ハイ (5V) を出力します。光電池は抵抗器のように振る舞い、どれだけの光が当たるかによって抵抗が変化します。

What You Can Measure

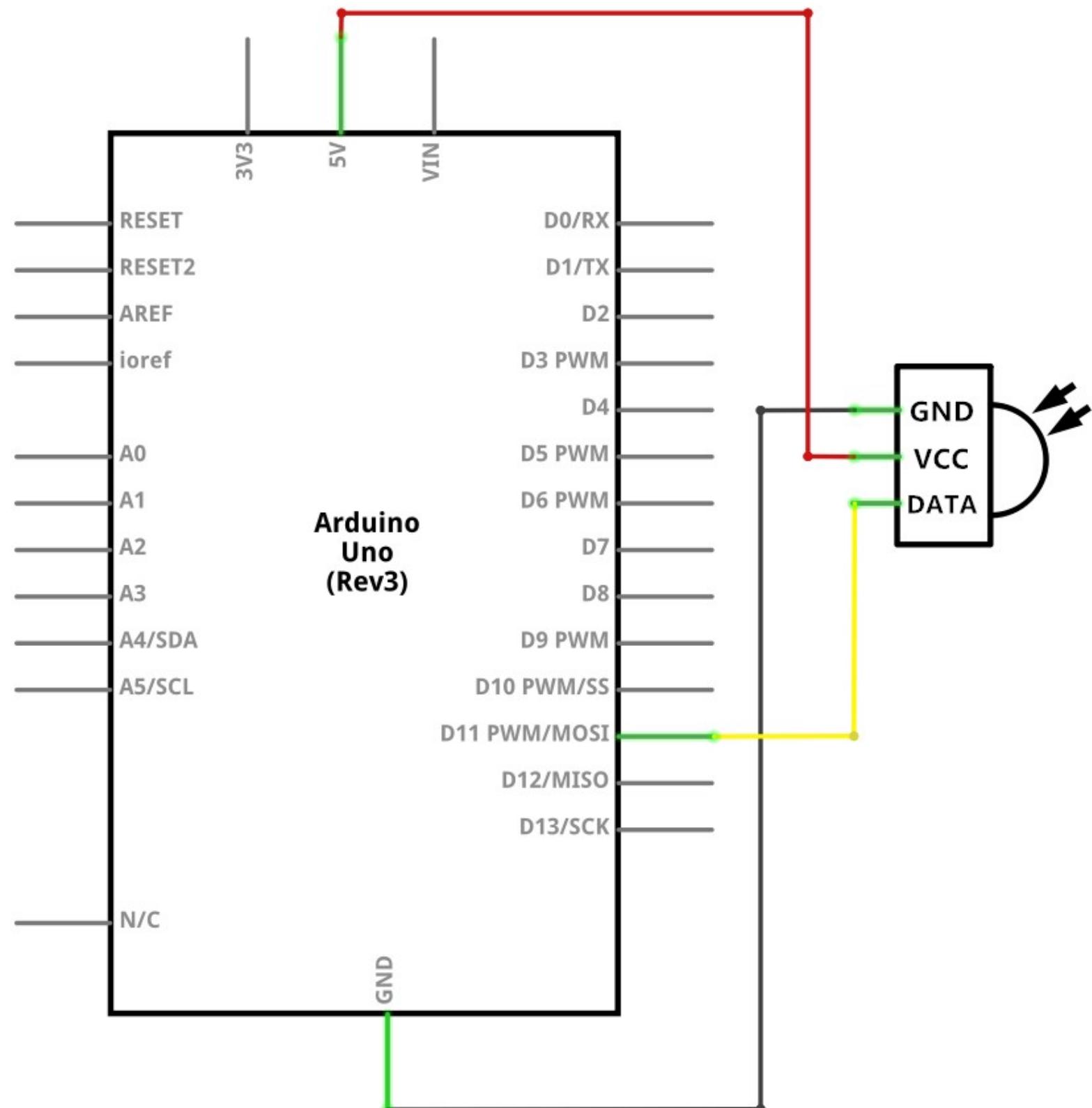


これらのデータシートのグラフからわかるように、ピーク周波数の検出は 38 KHz で、ピーク LED の色は 940 nm です。約 35KHz から 41KHz まで使用できますが、感度は低下し、遠くからも検出されません。同様に、850~1100 nm の LED を使用することもできますが、900~1000 nm と同様に動作しませんので、必ず一致する LED を取得してください! IR LED のデータシートで波長を確認してください。

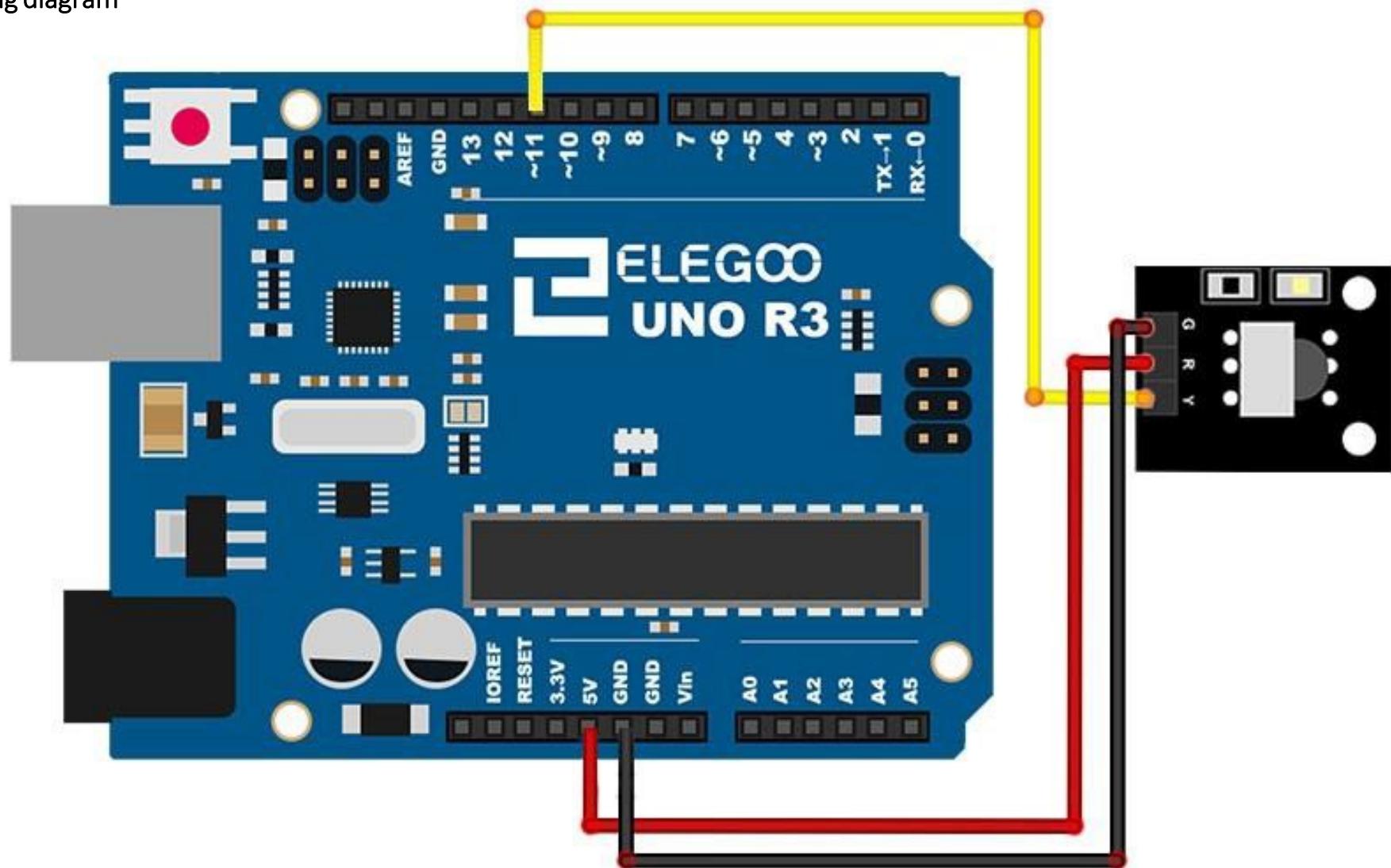
940nm を取得しようとする - 940nm は可視光ではないことを覚えておいてください。

Connection

Schematic



Wiring diagram



IR レシーバには 3 つの接続があります。

接続は信号、電圧、グラウンドです。

" - "はグラウンド、"S"は信号、中間ピンは電圧 5V です。

Code

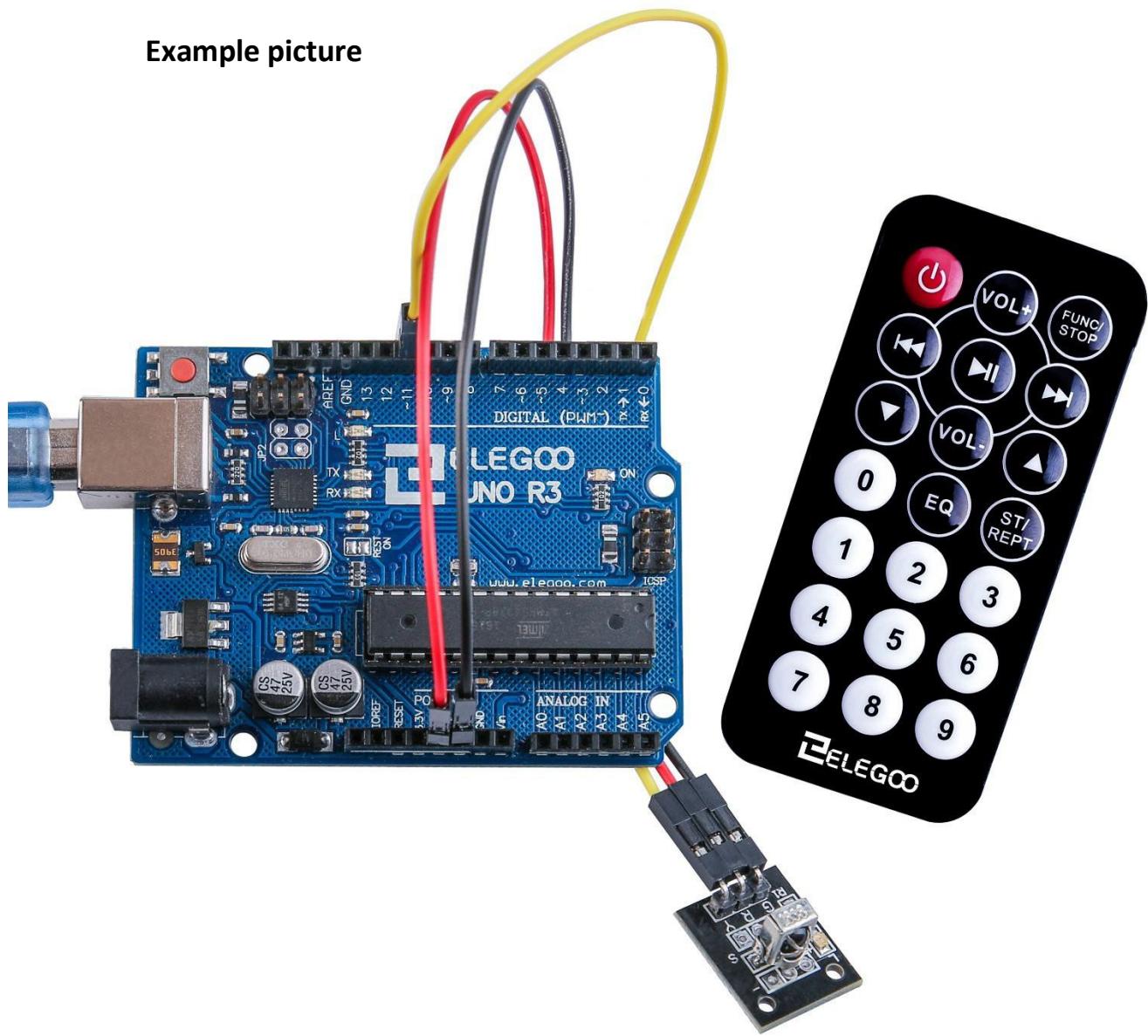
配線後、プログラムをコードフォルダ - レッスン 14 赤外線受信モジュールで開き、UPLOAD をクリックしてプログラムをアップロードしてください。エラーがある場合のプログラムアップロードの詳細については、レッスン 2 を参照してください。

これを実行する前に、<IRremote>ライブラリがインストールされていることを確認するか、必要に応じて再インストールしてください。そうしないと、コードが機能しません。

ライブラリファイルのロードの詳細については、レッスン 1 を参照してください。

次に、<RobotIRremote>を Library フォルダから移動します。これは、ライブラリが使用するライブラリと競合するためです。マイクロコントローラのプログラミングが完了したら、ライブラリフォルダの中にドラッグして戻すことができます。

Example picture



モニタを開くと、データが壊れているのがわかります:

[シリアルモニタ]ボタンをクリックしてシリアルモニタをオンにします。 シリアルモニタの基本については、レッスン 1 で詳しく説明しています。

```
IR Receiver Button Decode
POWER
VOL+
FUNC/STOP
FAST BACK
PAUSE
FAST FORWARD
DOWN
VOL-
UP
0
EQ
ST/REPT
1
2
3
4
5
6
7
8
9
```

Autoscroll Newline 9600 baud

Lesson 14 LCD Display

概要

このレッスンでは、英数字の LCD ディスプレイを配線して使用する方法を学習します。ディスプレイには LED バックライトがあり、各行に最大 16 文字の 2 行を表示できます。ディスプレイ上の各文字の長方形と各文字を構成するピクセルを見ることができます。ディスプレイは青色で白いだけで、テキストを表示するためのものです。

このレッスンでは、LCD ライブライ用の Arduino サンプルプログラムを実行しますが、次のレッスンではセンサーを使用して温度を表示するようにします。

必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x LCD1602 module
- (1) x Potentiometer (10k)
- (1) x 830 tie-points Breadboard
- (16) x M-M wires (Male to Male jumperwires)



部品の紹介

LCD1602

LCD1602 のピンの紹介:

VSS: グランドに接続するピン

VDD: + 5V 電源に接続するピン

VO: LCD1602 のコントラストを調整するピン。

RS: LCD のメモリ内のどこにデータを書き込むかを制御するレジスタ選択ピン。画面に表示されている内容を保持するデータレジスタ、または命令のレジスタ (LCD のコントローラが次に何をするかの指示を参照する) を選択できます。

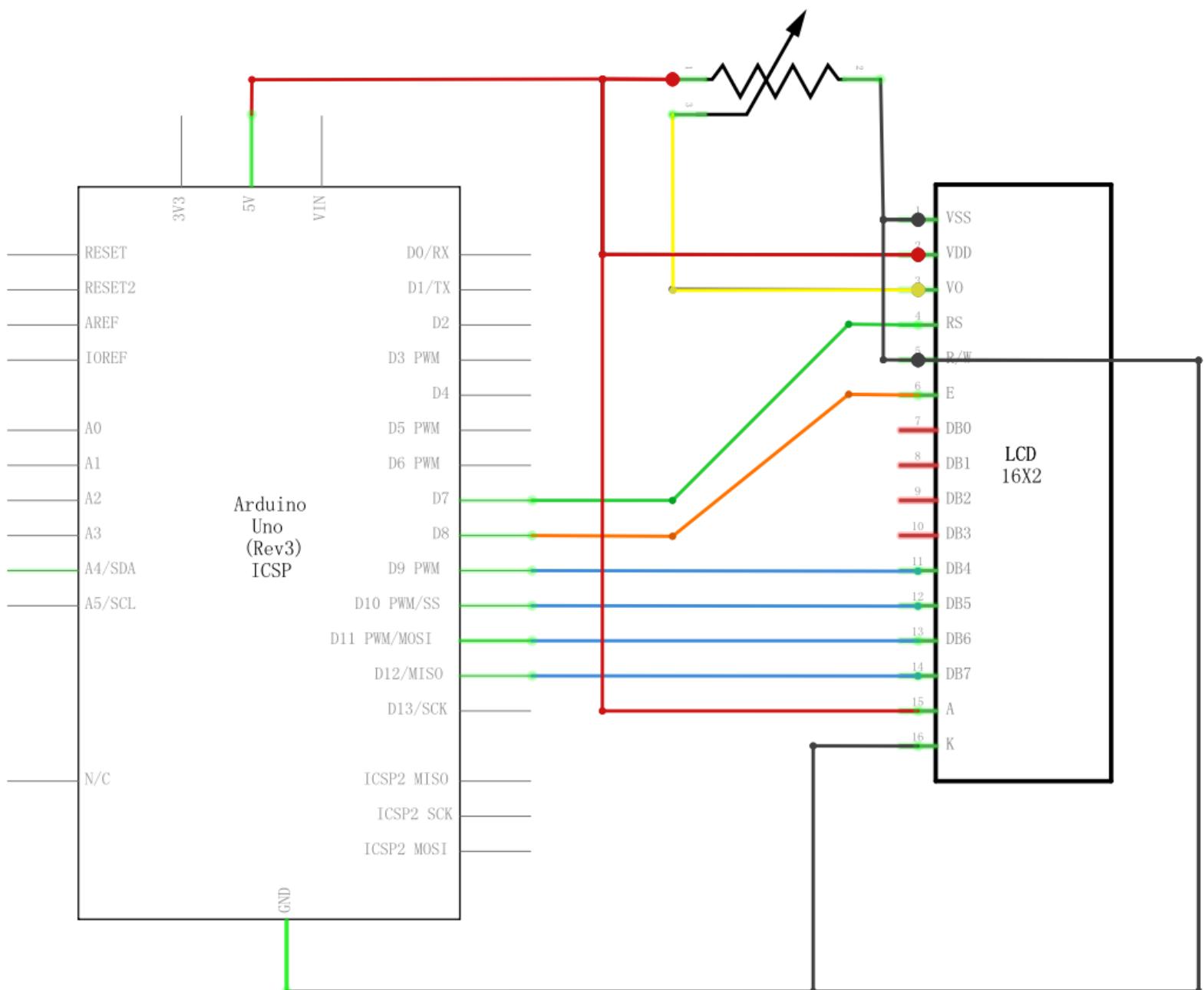
R/W: 読み出しモードまたは書き込みモードを選択する読み出し/書き込みピン

E: ローレベルのエネルギーが供給されると、LCD モジュールに関連する命令を実行させるイネーブルピン。

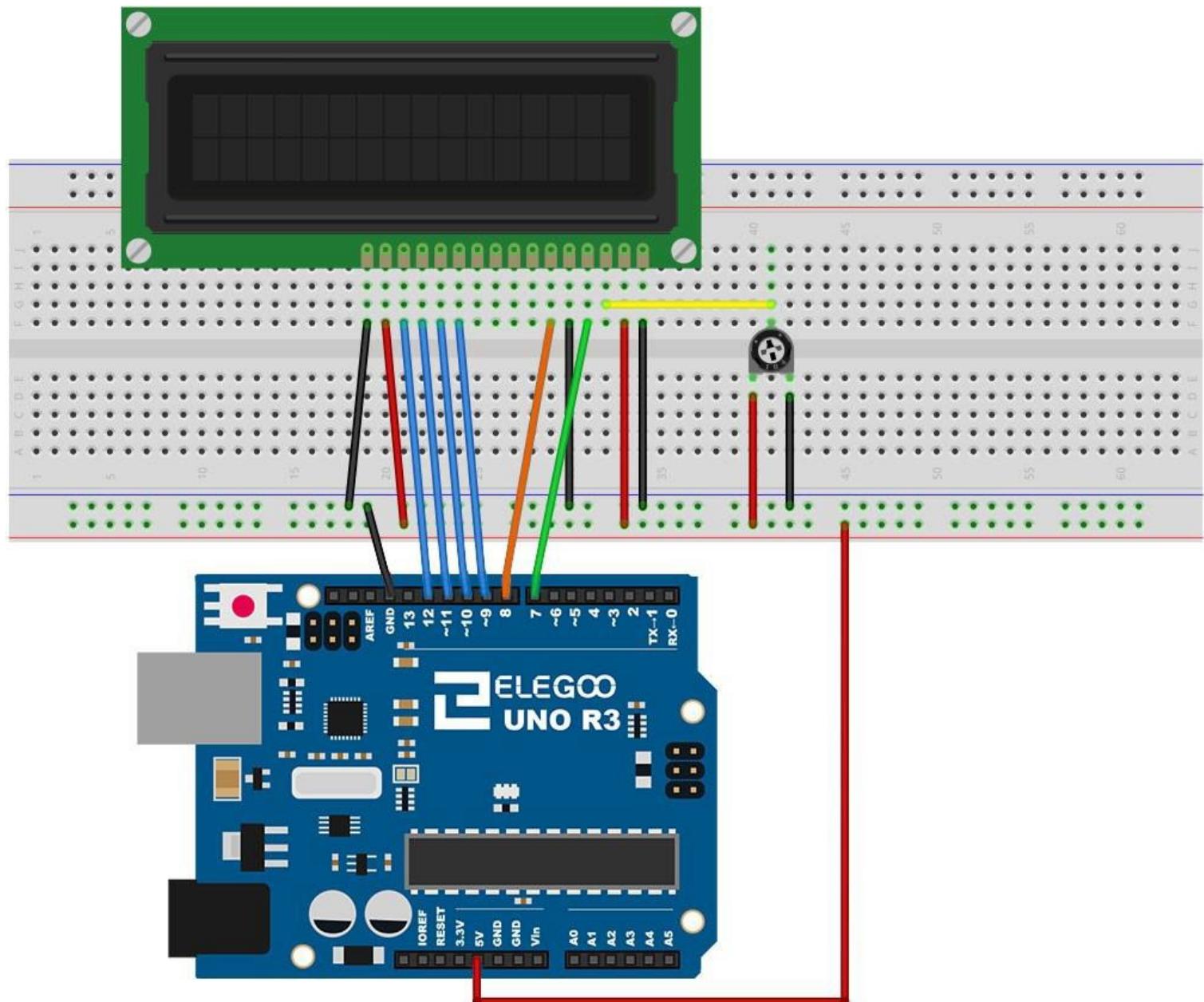
D0-D7: データを読み書きするためのピン

A and K: LED バックライトを制御するピン

Connection Schematic



Wiring diagram



LCD ディスプレイには 6 つの Arduino ピンが必要です。すべてがデジタル出力に設定されています。

また、5V と GND の接続も必要です。

いくつかの接続を行う必要があります。 ブレッドボードの上部にディスプレイを並べると、あまりカウントせずにピンを識別するのに役立ちます。特に、ブレッドボードの行の 1 行目がボードの一番上の行になっている場合は特にそうです。 ポットのスライダーをディスプレイのピン 3 につなぐ長い黄色のリード線を忘れないでください。 「ポット」は、ディスプレイのコントラストを制御するために使用されます。

ディスプレイにヘッダピンが取り付けられていない状態でディスプレイが供給されていることがあります。 その場合は、次のセクションの指示に従ってください。

Code

配線後、コードフォルダのレッスン 22 LCD ディスプレイでプログラムを開き、UPLOAD をクリックしてプログラムをアップロードしてください。 エラーがある場合のプログラムアップロードの詳細については、レッスン 2 を参照してください。

これを実行する前に、<LiquidCrystal>ライブラリがインストールされていることを確認するか、必要に応じて再インストールしてください。 そうしないと、コードが機能しません。

ライブラリファイルのロードの詳細については、[レッスン 1](#) を参照してください。

Arduino ボードにコードをアップロードすると、「hello, world」というメッセージが表示され、その後にゼロからカウントアップする数字が表示されます。

スケッチの最初のことは、次の行です:

```
#include <LiquidCrystal.h>
```

これは Arduino に液晶ライブラリを使用したいことを伝えます。

次に、修正する必要がある行があります。 これは、Arduino のどのピンをディスプレイのどのピンに接続するかを定義します。

```
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

このコードをアップロードした後、バックライトが点灯していることを確認し、文字メッセージが表示されるまでポテンショメータを調整します。

'setup'機能には、次の 2 つのコマンドがあります:

```
lcd.begin(16, 2);
```

```
lcd.print("Hello, World!");
```

最初は、液晶ディスプレイに表示される列と行の数を液晶ライブラリに伝えます。 2 行目には、画面の最初の行に表示されるメッセージが表示されます。

'loop'関数には、次の 2 つのコマンドもあります:

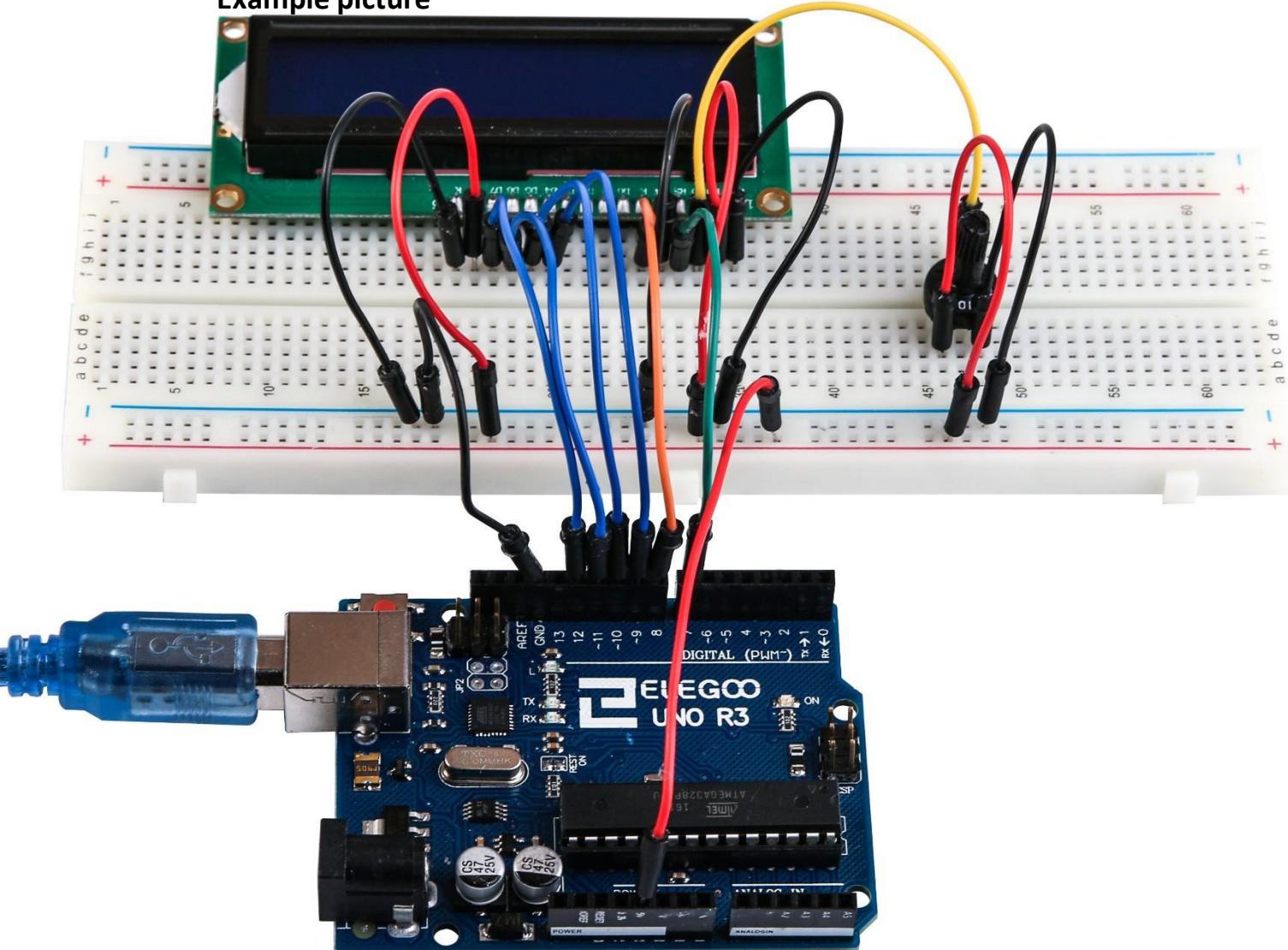
```
lcd.setCursor(0, 1);
```

```
lcd.print(millis()/1000);
```

最初の行は、カーソルの位置（次のテキストが表示される位置）を列 0 と行 1 に設定します。列番号と行番号は両方とも 0 ではなく 1 から始まります。

2 行目には、Arduino がリセットされてからのミリ秒数が表示されます。

Example picture



Lesson 15 Thermometer

概要

このレッスンでは、LCD ディスプレイを使用して温度を表示します。

必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x LCD1602 Module
- (1) x 10k ohm resistor
- (1) x Thermistor
- (1) x Potentiometer
- (1) x 830 tie-points Breadboard
- (18) x M-M wires (Male to Male jumperwires)

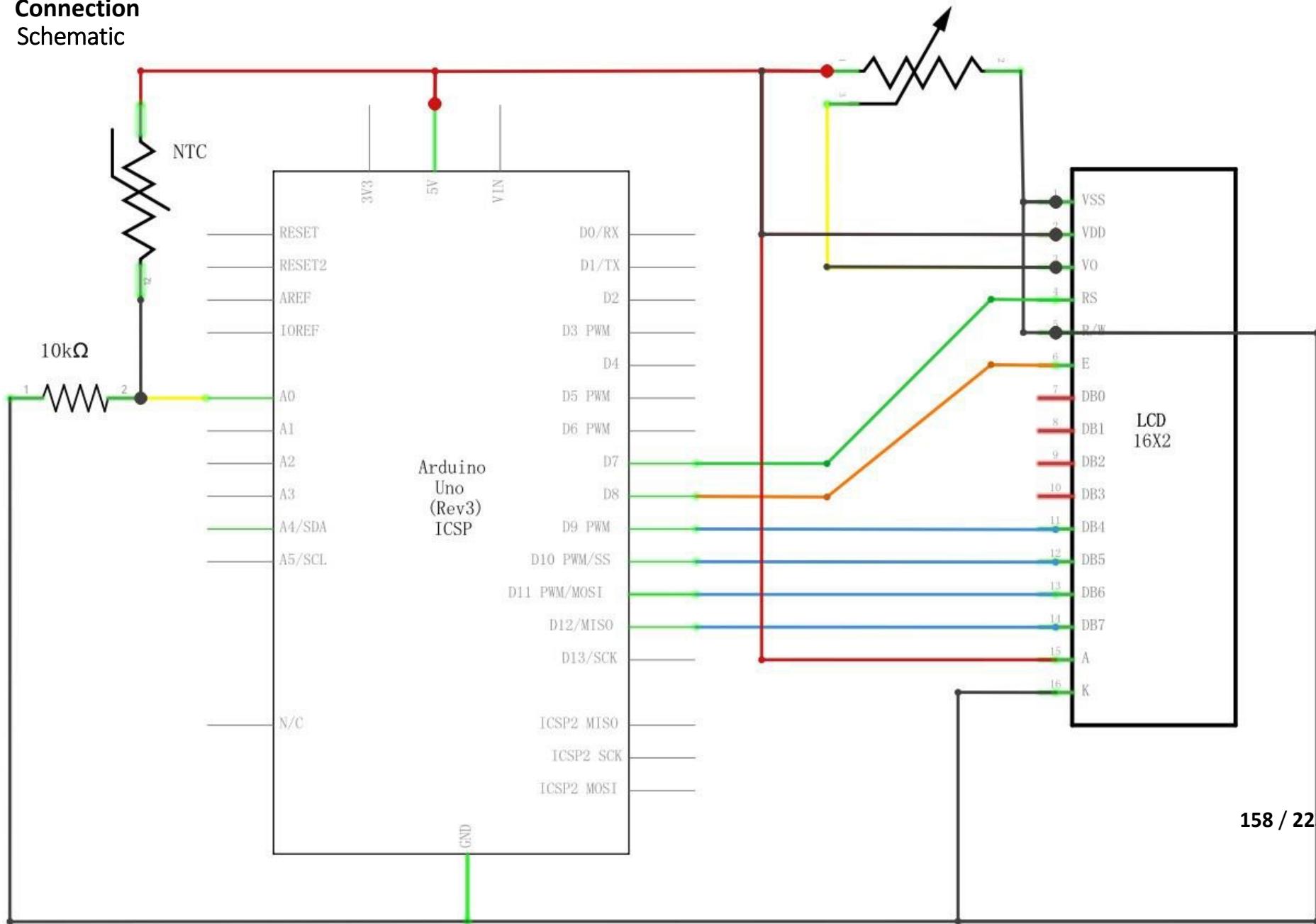
部品の紹介

Thermistor

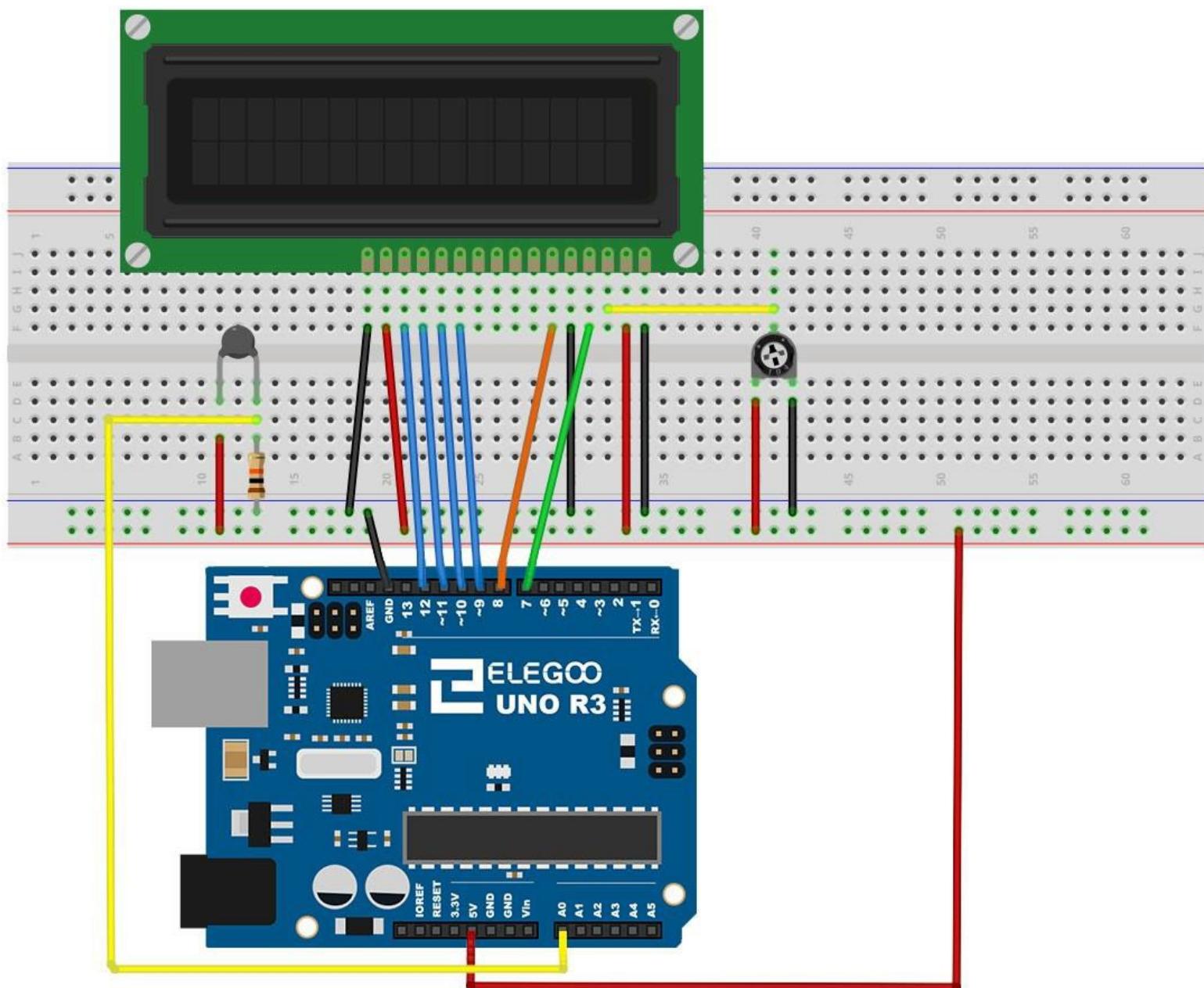
サーミスタはサーマル抵抗で、温度によって抵抗が変化する抵抗です。技術的には、すべての抵抗器はサーミスタです。抵抗は温度によってわずかに変化しますが、その変化は通常非常に小さく、測定するのが困難です。サーミスタは温度によって抵抗値が大きく変化するように作られており、1度あたり100オーム以上変化させることができます！

サーミスタにはNTC（負の温度係数）とPTC（正の温度係数）の2種類があります。一般に、温度測定にはNTCセンサーが使用されています。PTCはリセッタブルヒューズとしてよく使用されます。温度が上昇すると抵抗が増加します。つまり、電流が流れると発熱し、電流を「チョークバック」して回路を保護します。

Connection Schematic



Wiring diagram



ブレッドボードのレイアウトはレッスン 22 のレイアウトに基づいているため、ブレッドボードにこれをまだ残しておくと、簡単になります。

このレイアウトで少し動かされたポットの近くにいくつかのジャンパーウイヤーがあります。10kΩの抵抗とサーミスタはすべてボードに追加されています。

Code

配線後、プログラムをコードフォルダ - レッスン 23 温度計で開き、アップロードをクリックしてプログラムをアップロードしてください。 エラーがある場合のプログラムアップロードの詳細については、レッスン 2 を参照してください。

これを実行する前に、<LiquidCrystal>ライブラリがインストールされていることを確認するか、必要に応じて再インストールしてください。 そうしないと、コードが機能しません。

ライブラリファイルのロードの詳細については、[レッスン 1](#)を参照してください。

これについてのスケッチはレッスン 22 のスケッチに基づいています。それをあなたの Arduino にロードして、あなたの指を置くことによって温度センサーを暖めると、温度の上昇がわかります。

'lcd'コマンドの上にコメント行を置くと便利です。

```
// BS      E      D4 D5      D6 D7
```

```
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

これにより、使用するピンを変更する場合にはより簡単になります。

'ループ'機能では、2つの面白いことが起こっています。 まず、温度センサーからのアナログを実際の温度に変換しなければなりません。次に、それらを表示する方法を工夫しなければなりません。

まず、温度の計算を見てみましょう。

```
int tempReading = analogRead(tempPin);
double tempK = log(10000.0 * ((1024.0 / tempReading - 1)));
tempK = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * tempK * tempK ))
* tempK );
float tempC = tempK - 273.15;
float tempF = (tempC * 9.0) / 5.0 + 32.0;
```

液晶ディスプレイに変化する表示を表示するのは難しいことがあります。 主な問題は、読みが常に同じ桁数になるとは限りません。 したがって、温度が 101.50 から 99.00 に変更された場合、古い読み取り値からの余分な数字がディスプレイに残る危険があります。

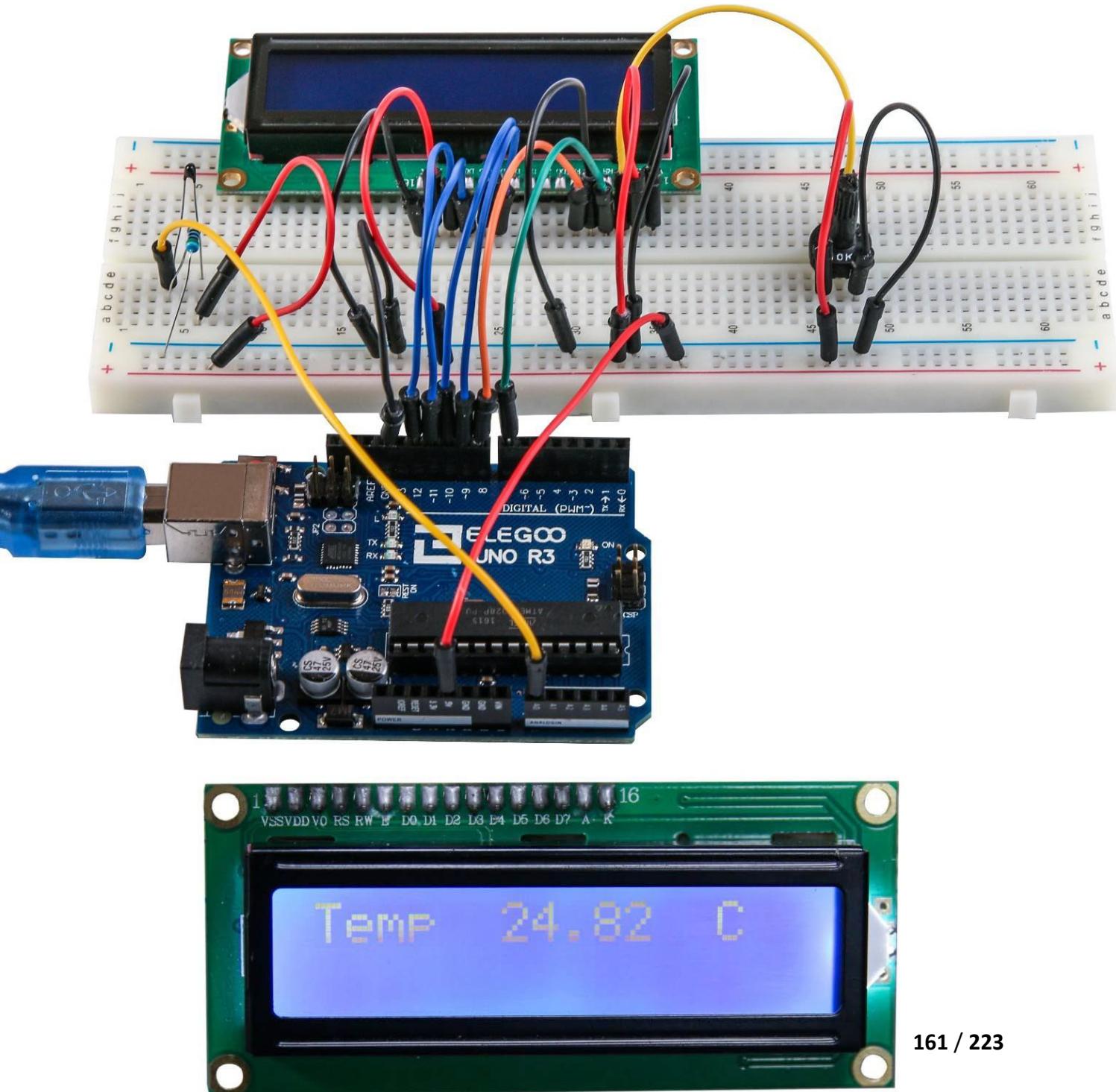
これを避けるには、ループの周りに LCD の行全体を書きます。

```
lcd.setCursor(0, 0);
lcd.print("Temp      C  ");
lcd.setCursor(6, 0);
lcd.print(tempF);
```

かなり奇妙なコメントは、ディスプレイの 16 の列を思い出させるのに役立ちます。 その長さの文字列を実際の読み取りが行われるスペースで印刷することができます。

ブランクを記入するには、読み取り位置のカーソル位置を設定し、それを印刷します。

Example picture



Lesson 16 Eight LED with 74HC595

概要

このレッスンでは、8つの出力ピンをあきらめることなく、UNOで8つの大きな赤いLEDを使用する方法を学習します。UNOピンに8個のLEDを接続して抵抗を接続することもできますが、すぐにUNOのピンが使い尽くされます。UNOにたくさんのが繋がっていない場合。そうすることはOKですが、ボタンやセンサー、サーボなどが必要な時がありますが、ピンが残っていないことを知る前によく覚えておいてください。したがって、これを行う代わりに、74HC595シリアル・パラレル・コンバータと呼ばれるチップを使用することになります。このチップには8つの出力(完璧)と3つの入力があり、一度に少しづつデータを入力します。

このチップは、LEDを駆動するのが少し遅くなります(1秒間に8,000,000回ではなく1秒間に約500,000回LEDを変えることができます)が、人間が検出できるよりも速く、速いです。

必要な構成部品:

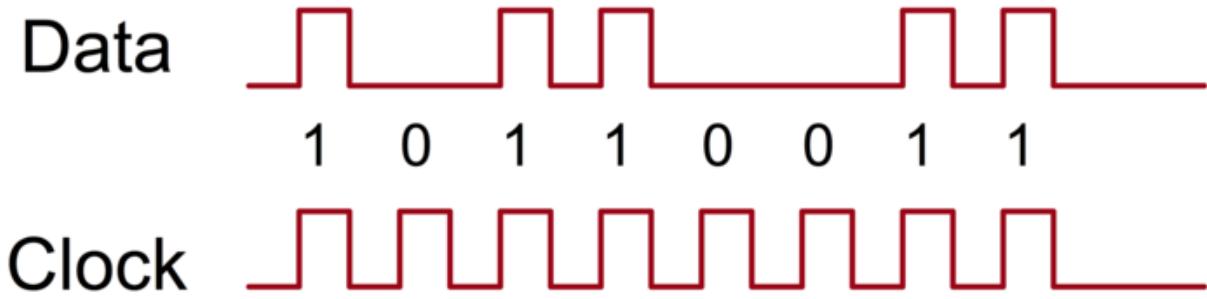
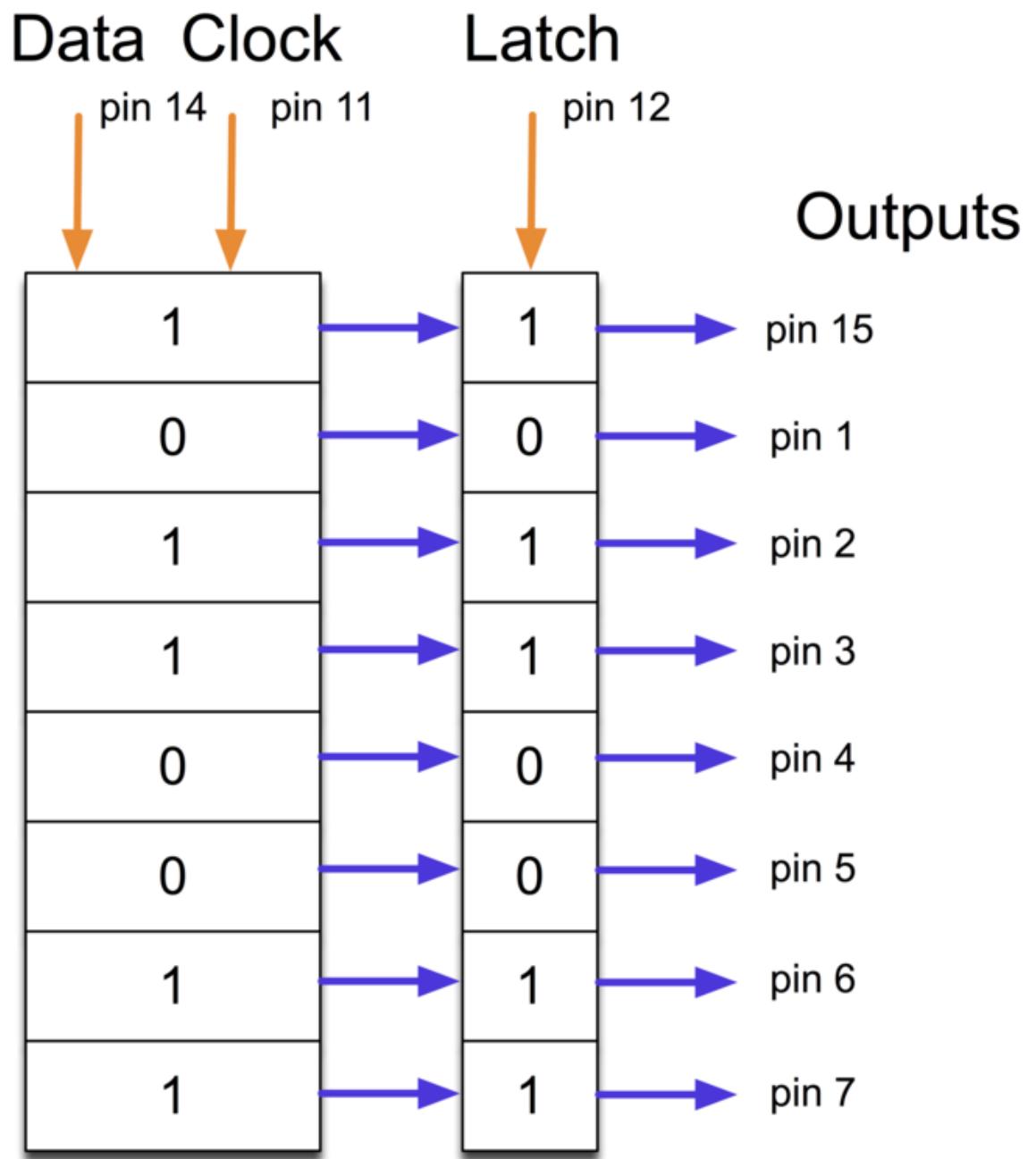
- (1) x Elegoo Uno R3
- (1) x 830 tie-points breadboard
- (8) x leds
- (8) x 220 ohm resistors
- (1) x 74hc595 IC
- (14) x M-M wires (Male to Male jumperwires)



部品の紹介

74HC595 Shift Register:

シフトレジスタは、8つのメモリ位置として考えられるものを保持する一種のチップであり、それが1または0のいずれかになります。これらの値をそれぞれオンまたはオフに設定するには、チップの「データ」ピンと「クロック」ピンです。

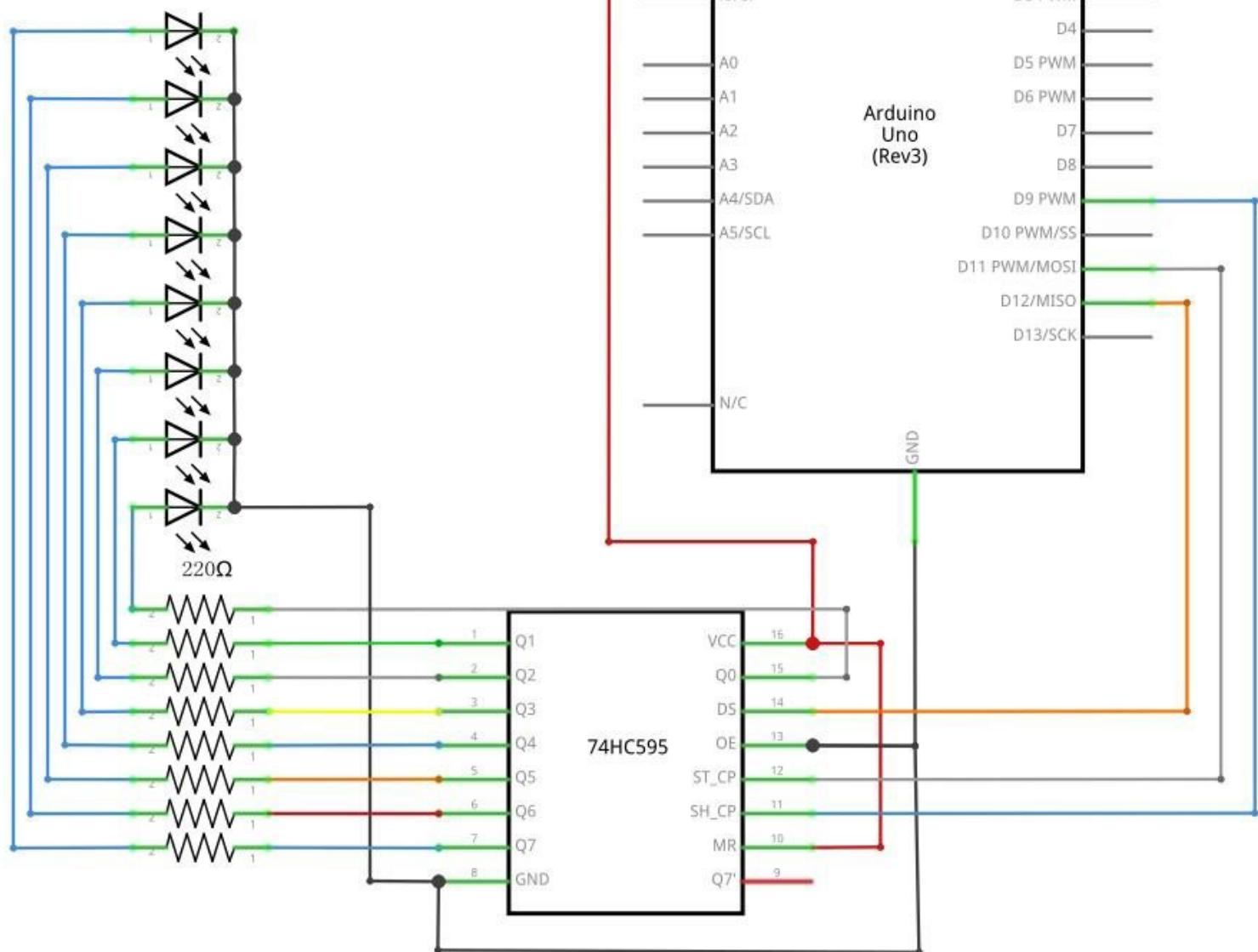


クロックピンは8個のパルスを受信する必要があります。各パルスで、データピンがハイの場合、1がシフトレジスタにプッシュされます。8個のパルスがすべて受信されると、「Latch」ピンを有効にすると8個の値がラッチ・レジスタにコピーされます。これは必要です。そうしないと、データがシフトレジスタにロードされている間、間違ったLEDが点滅します。

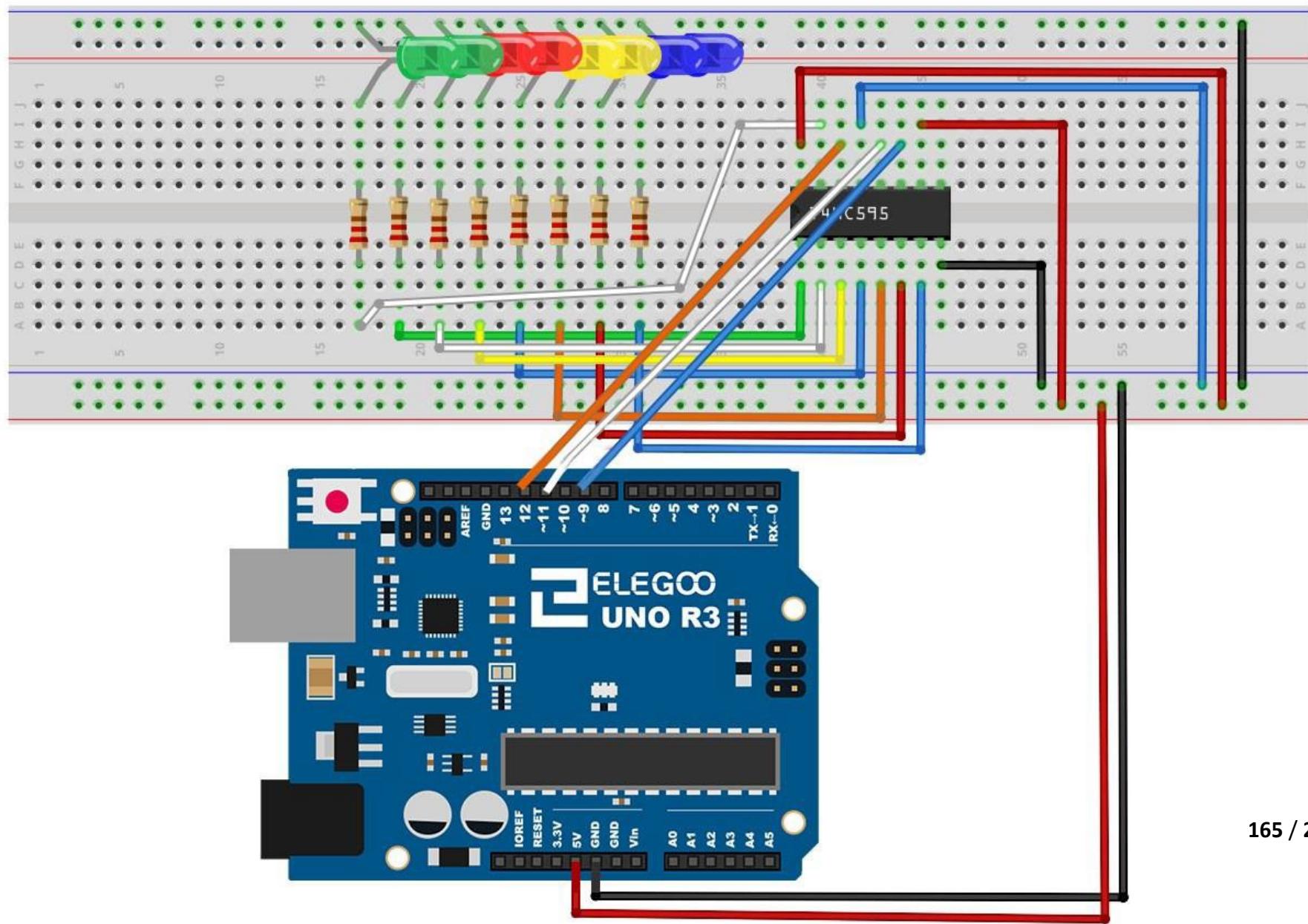
また、チップには出力イネーブル(OE)ピンがあり、これを使用して出力を一度にイネーブルまたはディセーブルします。これをPWM対応のUNOピンに接続し、「analogWrite」を使用してLEDの輝度を制御することができます。このピンはアクティブローであるため、GNDに接続します。

Connection

Schematic



Wiring diagram



8つのLEDと8つの抵抗を接続するので、実際にはかなりの数の接続が必要です。

他のすべてのものがそれに接続するので、おそらく74HC595チップを最初に置くのが最も簡単です。小さなU字型ノッチがブレッドボードの上部に向くようにしてください。チップのピン1はこのノッチの左側にあります。

UNOからのデジタル12がシフトレジスタの14番ピンに行きますUNOからのデジタル11がシフトレジスタの12番ピンに行きますデジタルUNOからデジタル9番がシフトレジスタのピン#11に行きます

ICの出力の1つを除いて、すべてがチップの左側にあります。したがって、接続を容易にするために、LEDも同様です。

チップの後、抵抗器を所定の位置に配置します。抵抗器のリードのどれも互いに接触していないことに注意する必要があります。電源をUNOに接続する前に、これを再度確認する必要があります。リードが接触していない状態で抵抗器を配置することが困難な場合は、リード線を短くしてブレッドボードの表面に近づけるようにしてください。

次に、ブレッドボードにLEDを配置します。より長い正のLED導線は、チップ上にあるブレッドボードのどちらの側にもなければなりません。

上記のようにジャンパリードを取り付けます。ICのピン8からブレッドボードのGNDカラムに行くものを忘れないでください。少し後でリストされたスケッチをロードし、それを試してみてください。すべてのLEDが点灯するまで各LEDが順番に点灯し、すべて消灯してサイクルが繰り返されます。

Code

配線後、コードフォルダにあるプログラムを開きます - レッスン24 8つのLEDと74HC595を開き、アップロードをクリックしてプログラムをアップロードします。エラーがある場合のプログラムアップロードの詳細については、レッスン2を参照してください。

最初に行うのは、使用する3つのピンを定義することです。これらはUNO

74HC595のラッチ、クロック、およびデータ・ピンに接続されるデジタル出力を備えています。

```
int latchPin = 11; int clockPin = 9; int dataPin  
= 12;
```

次に、「leds」と呼ばれる変数が定義されます。これは、LEDが現在オンまたはオフになっているパターンを保持するために使用されます。タイプ「バイト」のデータは、8ビットを使用する数字を表す。各ビットはオンまたはオフのどちらでもかまいませんので、これは私たちの8つのLEDのうちのどれがオンまたはオフになっていますか？

```
byte leds = 0;
```

「セットアップ」機能は、使用している3つのピンをデジタル出力に設定するだけです。

```
void setup()  
{  
    pinMode(latchPin, OUTPUT);    pinMode(dataPin,  
    OUTPUT); pinMode(clockPin, OUTPUT);  
}
```

'loop'関数は最初、変数'leds'に値0を渡して、すべてのLEDをオフにします。次に、「leds」パターンをシフトレジスタに送り、すべてのLEDがオフになる'updateShiftRegister'を呼び出します。後'updateShiftRegister'がどのように動作するかを扱います。

ループ関数は0.5秒間停止し、「for」ループと変数'i'を使用して0から7までカウントし始めます。毎回、Arduino関数'bitSet'を使用して、変数'leds'のLEDを制御するビットを設定します。次に、「updateShiftRegister」を呼び出して、'leds'が変数'leds'にあるものを反映するように更新します。次に、「i」がインクリメントされ、次のLEDが点灯するまでに半秒の遅延があります。

```
void loop()
```

```

{
  leds = 0; updateShiftRegister(); delay(500);
  for (int i = 0; i < 8; i++)
  {
    bitSet(leds, i); updateShiftRegister(); delay(500);
  }
}

```

関数 'updateShiftRegister'は、最初に latchPin をローに設定し、次に 'latchPin'を再びハイにする前に UNO 関数 'shiftOut'を呼び出します。 これは 4 つのパラメータをとり、最初の 2 つはデータとクロックにそれぞれ使用するピンです。

3 番目のパラメータは、データのどの部分を開始するかを指定します。 私たちは、最も重要なビットから始めるつもりです。 これは、「最下位ビット」です。 Bit' (LSB).

最後のパラメータは、シフトレジスタにシフトされる実際のデータです（この場合は 'leds'です）。

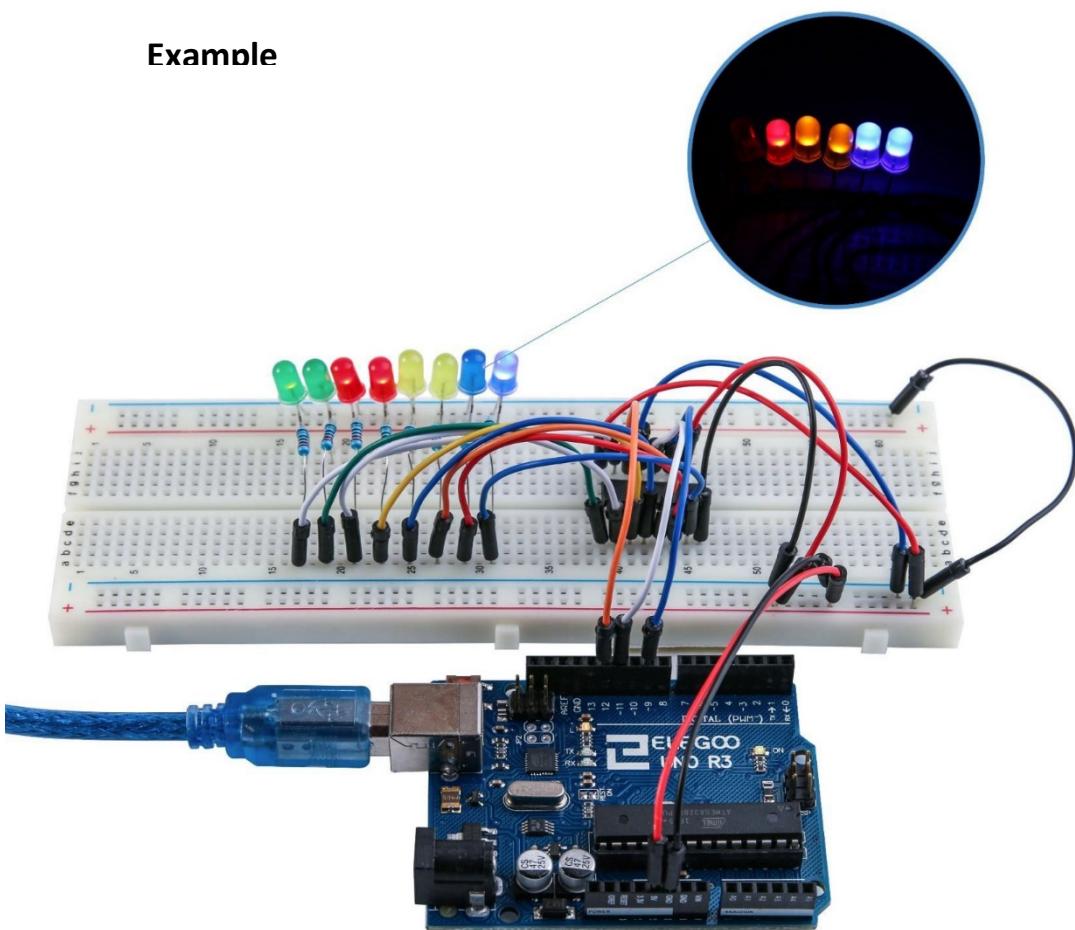
```

void updateShiftRegister()
{
  digitalWrite(latchPin, LOW); shiftOut(dataPin, clockPin, LSBFIRST, leds); digitalWrite(latchPin, HIGH);
}

```

LED の 1 つをオンにするのではなくオフにしたい場合は、同様の Arduino 関数 (bitClear) を 'leds'変数でコールします。 これにより、'leds'のビットが 0 に設定され、実際の LED を更新するために 'updateShiftRegister'を呼び出すだけでそれに続く必要があります。

Example



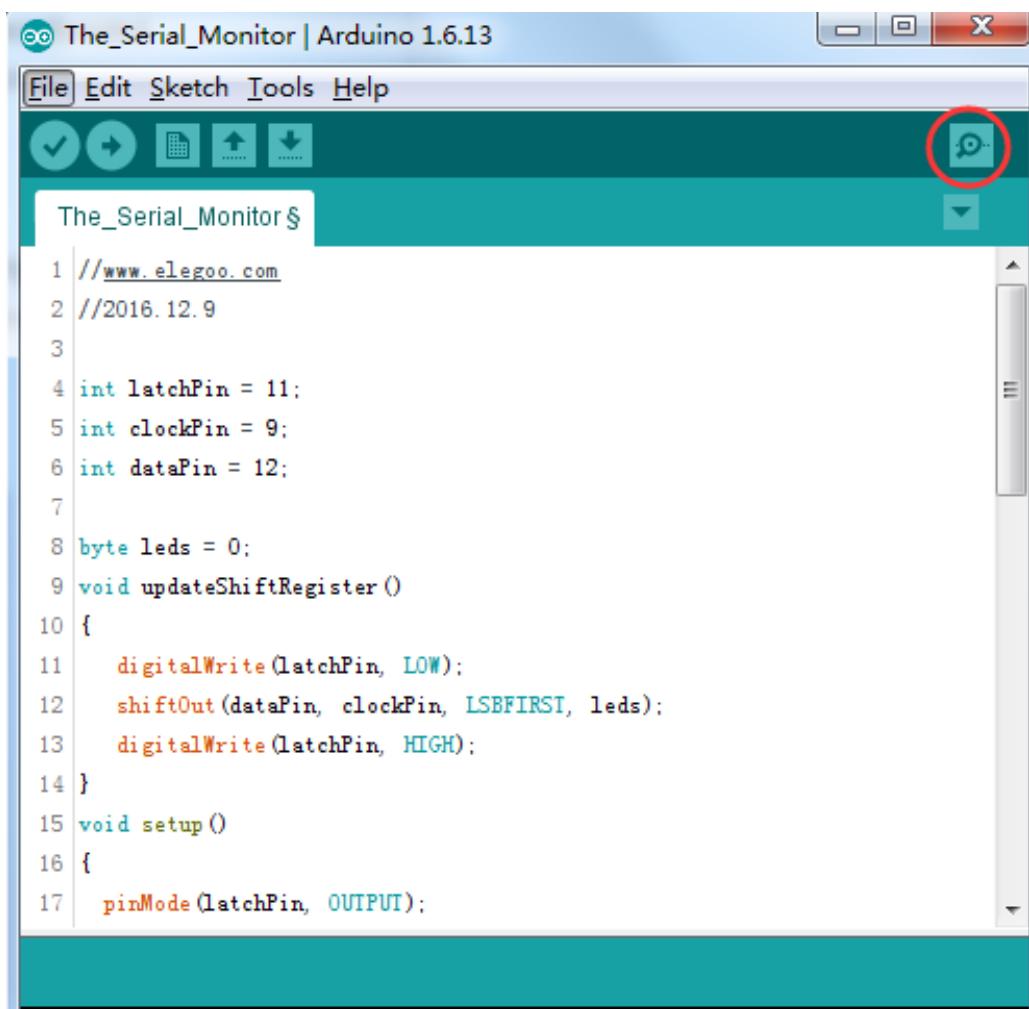
Lesson 17 シリアルモニタ

概要

このレッスンでは、レッスン 16 をベースに、Arduino シリアルモニターを使用してコンピュータから LED を制御する機能を追加します。シリアルモニタは、コンピュータと UNO との間の「つなぎ」です。それは、デバッグに便利で、キーボードから UNO を制御するために、テキストメッセージを送受信することができます！たとえば、コンピュータからコマンドを送信して LED を点灯させることができます。このレッスンでは、レッスン 16 とまったく同じ部品と同様のブレッドボードレイアウトを使用します。まだレッスン 16 に従っていない場合は、レッスン 16 に従ってください。

Steps taken

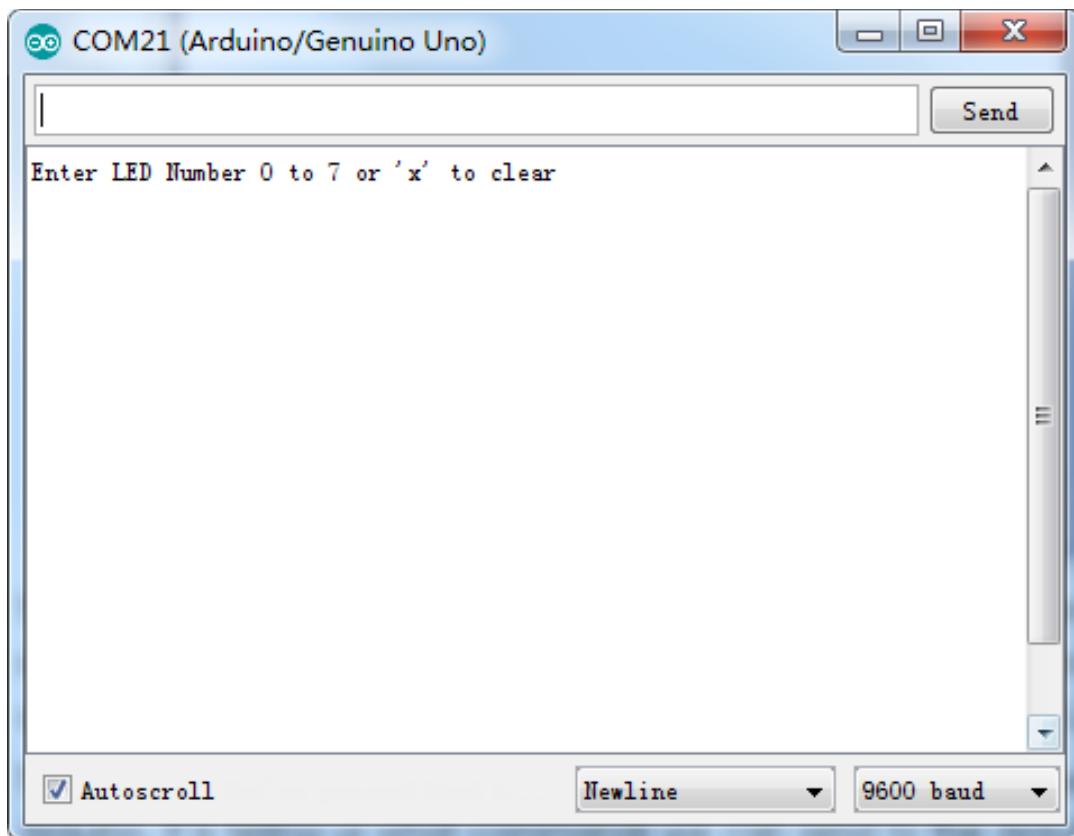
このスケッチをあなたの UNO にアップロードしたら、Arduino IDE のツールバーの一番右のボタンをクリックしてください。ボタンは下に丸で囲まれています。



```
//www.elegoo.com
//2016.12.9
int latchPin = 11;
int clockPin = 9;
int dataPin = 12;
byte leds = 0;
void updateShiftRegister()
{
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, LSBFIRST, leds);
    digitalWrite(latchPin, HIGH);
}
void setup()
{
    pinMode(latchPin, OUTPUT);
```

次のウィンドウが開きます。

[シリアルモニタ]ボタンをクリックしてシリアルモニタをオンにします。 シリアルモニタの基本については、レッスン 1 で詳しく説明しています。

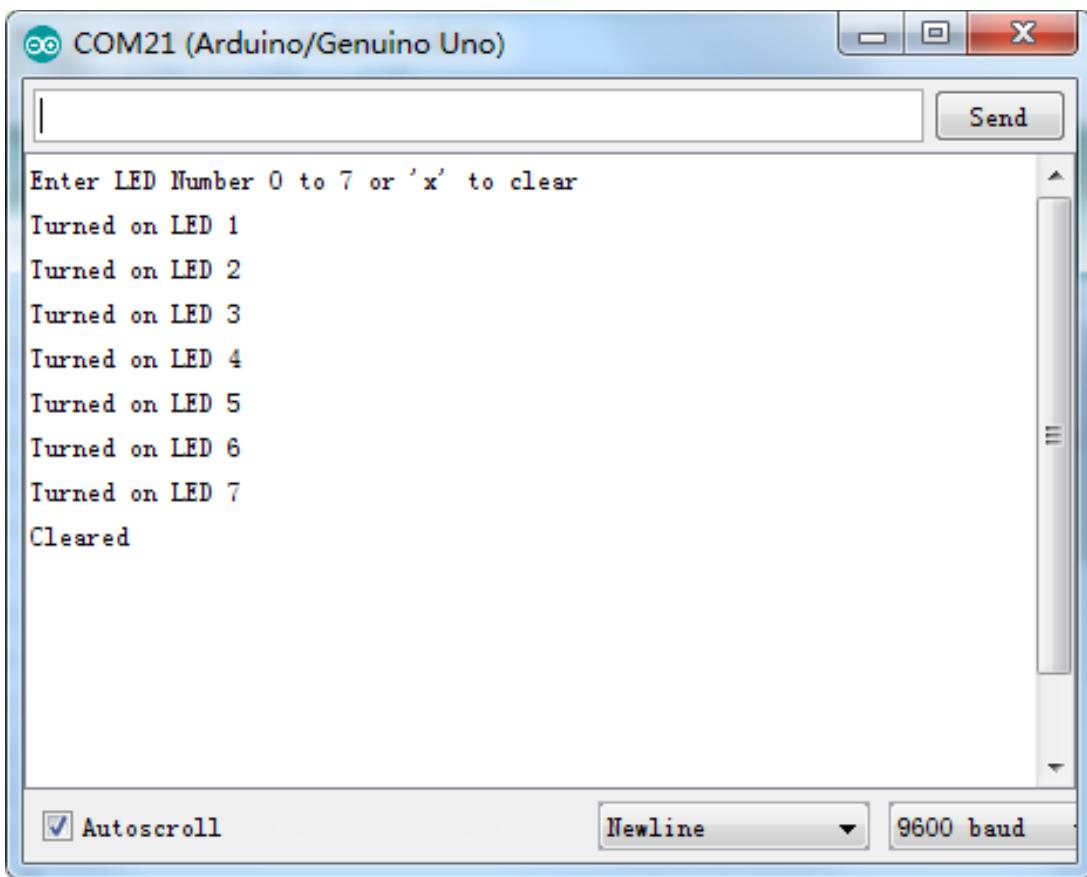


このウィンドウはシリアルモニタと呼ばれ、Arduino IDE ソフトウェアの一部です。 その仕事は、コンピュータから UNO ボード (USB 経由) へのメッセージの送信と、UNO からのメッセージの受信の両方を可能にすることです。

Arduino によって「LED 番号 0~7 を入力するか、x 'をクリアする」というメッセージが送信されました。 Arduino に送ることができるコマンドは、「x」（すべての LED をオフにする）またはオンにする LED の番号（0 は下の LED、1 は次の LED 1 つ、上の LED で 7 まで）。

シリアルモニタの上部に「送信」ボタンを押して、次のコマンドを入力してみてください。これらの文字をそれぞれ入力した後、「送信」を押します。x 0 3 5

LED がすでにすべて消灯している場合は、x を入力しても効果はありませんが、各番号を入力すると対応する LED が点灯し、UNO ボードから確認メッセージが表示されます。 シリアルモニタが以下のように表示されます。



もう一度 x と入力して[送信]を押すと、すべての LED が消灯します。

Code

配線後、コードフォルダにあるプログラムを開きます - レッスン 25 シリアルモニタを開き、アップロードをクリックしてプログラムをアップロードしてください。エラーがある場合のプログラムアップロードの詳細については、レッスン 2 を参照してください。

ご想像のとおり、スケッチはレッスン 24 で使用されているスケッチに基づいています。そこでここでは新しいビットについて説明します。 Arduino IDE の完全なスケッチを参照すると便利です。

'setup'機能には、最後に 3 つの新しい行があります:

```
void setup()
{
    pinMode(latchPin, OUTPUT);
    pinMode(dataPin, OUTPUT);
    pinMode(clockPin, OUTPUT);
    updateShiftRegister();
    Serial.begin(9600);
```

```
while (! Serial); // Wait until Serial is ready -Leonardo  
Serial.println("Enter LED Number 0 to 7 or 'x' to clear");  
}
```

まず、'Serial.begin (9600)'というコマンドがあります。これによりシリアル通信が開始され、UNOはUSB接続を介してコマンドを送信できます。値9600は接続の「ポーレート」と呼ばれます。これはデータの送信速度です。これをより高い値に変更できますが、Arduinoシリアルモニターも同じ値に変更する必要があります。これについては後で説明します。今のところ、9600のままにしておいてください。

'while'で始まる行は、Arduinoがメッセージの送信を開始する前に、USB接続のもう一方の端に何かがあることを保証します。そうしないと、メッセージは送信されても表示されないことがあります。このラインは、Arduino UNOがArduinoボードを自動的にリセットするので、Arduino Leonardoを使用している場合にのみ必要です。これは、シリアルモニターを開いたときに自動的にリセットされますか、これはLeonardoでは起こりません。

'setup'の最後の新しい行は、シリアルモニタの上部に表示されているメッセージを送信します。

「ループ」機能は、すべてのアクションが発生する場所です：

```
void loop()  
{  
    if (Serial.available())  
    {  
        char ch = Serial.read();  
        if (ch >= '0' && ch <= '7')  
        {  
            int led = ch - '0';  
            bitSet(leds, led);  
            updateShiftRegister();  
            Serial.print("Turned on LED ");  
            Serial.println(led);  
        }  
        if (ch == 'x')  
        {  
            leds = 0;  
            updateShiftRegister();  
        }  
    }  
}
```

```
    Serial.println("Cleared");
}
}
```

ループ内で発生するすべてのものは、'if'ステートメント内に含まれます。だから、組み込みの Arduino 関数 'Serial.available ()' の呼び出しが '真'でなければ、それ以外のことは起こりません。 'Serial.available ()' は、データが UNO に送信され、処理準備が整っていれば 'true'を返します。 受信メッセージはバッファと呼ばれるものに保持され、そのバッファが空でない場合は 'Serial.available ()' が true を返します。

メッセージが受信された場合は、次のコード行に表示されます:

```
char ch = Serial.read();
```

これにより、バッファから次の文字が読み込まれ、バッファから削除されます。また、変数 'ch'に割り当てます。変数 'ch'は 'char'を表す 'char'型であり、名前が示唆するように、単一の文字を保持します。

シリアルモニタの上部にあるプロンプトの指示に従っている場合、この文字は 0 から 7 の間の 1 枠の数字か「x」の文字のいずれかになります。

次の行の 'if'ステートメントは、「ch」が文字 '0'以上で、文字 '7'以下であることを確認して、1 枠であるかどうかをチェックします。この方法で文字を比較するのが少し奇妙に見えますが、完全に受け入れられます。

各文字は、その ASCII 値と呼ばれる固有の番号で表されます。これは、<=と> =を使用して文字を比較すると、実際には比較されていた ASCII 値であることを意味します。

テストに合格すると、次の行に進みます:

```
int led = ch - '0';
```

今は文字の算術演算を行っています! 入力された数字のいずれかから数字「0」を減算しています。したがって、「0」を入力すると '0' - '0'は 0 になります。「7」を入力すると '7' - '0'は実際に使用されている ASCII 値なので数字 7 と等しくなります 減算で LED をオンにしたいということを知っているので、そのビットを変数 'leds'にセットし、シフトレジスタを更新するだけです。

```
bitSet(leds, led);
updateShiftRegister();
```

次の 2 行は、確認メッセージをシリアルモニタに書き戻します。

```
Serial.print("Turned on LED ");
Serial.println(led);
```

最初の行は Serial.println ではなく Serial.print を使用しています。 2 つの違いは、Serial.print は、そのパラメータに何も印刷した後に新しい行を開始しないということです。最初の行でこれを使用します。これは、メッセージを 2 つの部分に出力するためです。 最初に、一般的なビット： 'Turned on LED' と LED の番号。

LED の数は、テキスト文字列ではなく、「int」変数に保持されます。 Serial.print は、二重引用符で囲まれたテキスト文字列、または 'int' かそれともかなりの種類の変数をとることができます。 ケースを処理する 'if' ステートメントの後に、1 つの数字が処理された場合、 'ch' が文字 'x' であるかどうかを確認する 2 番目の 'if' ステートメントがあります。

```
if (ch == 'x')
{
    leds = 0;
    updateShiftRegister();
    Serial.println("Cleared");
}
```

そうであれば、すべての LED をクリアし、確認メッセージを送信します。

Lesson 18 光電池

概要

このレッスンでは、アナログ入力を使用して光強度を測定する方法を学習します。 レッスン 16 で構築し、ライトのレベルを使用して点灯させる LED の数を制御します。

光電管は、鍋が上にあったブレッドボードの底にあります。

必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x 830 tie-points breadboard
- (8) x leds
- (8) x 220 ohm resistors
- (1) x 1k ohm resistor
- (1) x 74hc595 IC
- (1) x Photoresistor (Photocell)
- (16) x M-M wires (Male to Male jumperwires)



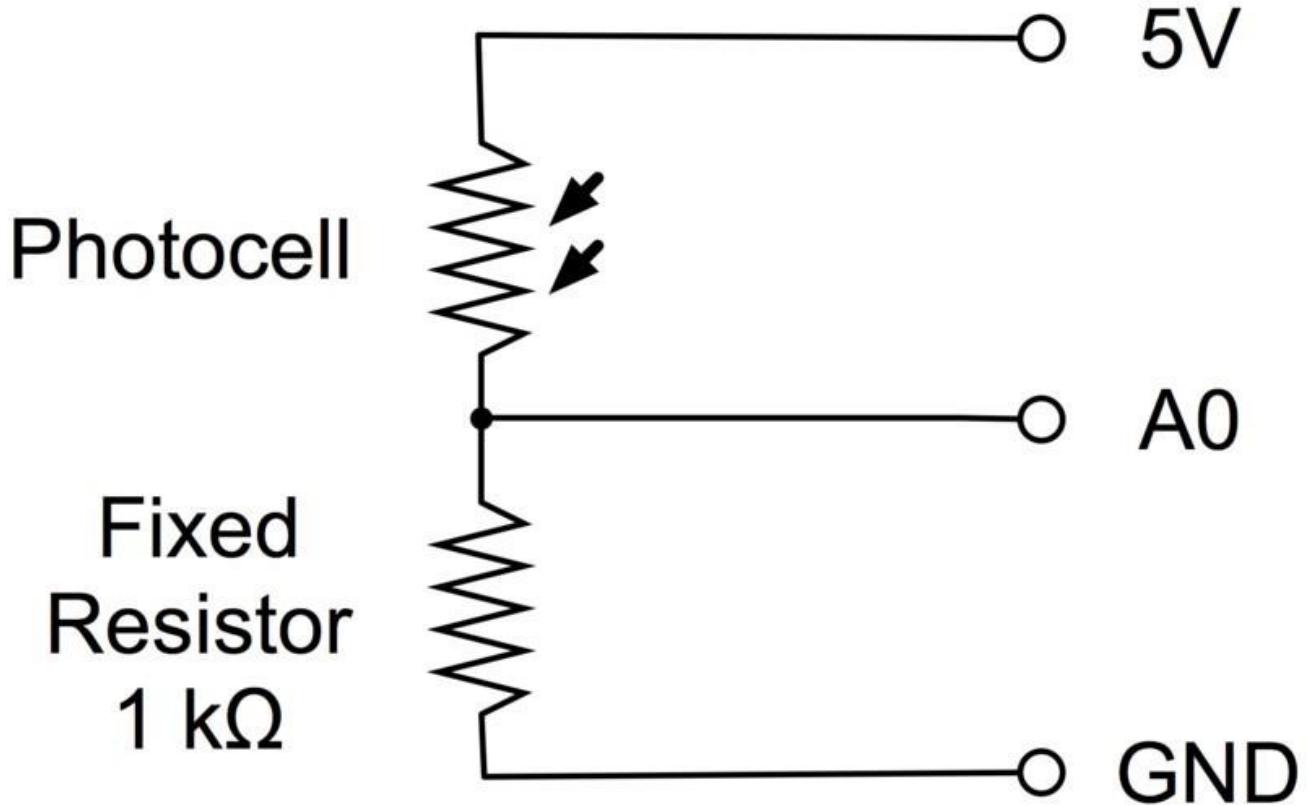
部品の紹介

PHOTOCELL:

使用される光電池は、光依存性抵抗器と呼ばれるタイプのもので、LDR と呼ばれることもあります。 名前が示すように、これらのコンポーネントは抵抗器のように機能しますが、どれだけの光がそれらに当たるかに応じて抵抗が変化します。

これは、暗闇では約 $50\text{k}\Omega$ 、明るい光では 500Ω の抵抗を持っています。 この抵抗値の変化を UNO R3 ボードのアナログ入力で測定できるものに変換するには、電圧に変換する必要があります。

これを行う最も簡単な方法は、固定抵抗と組み合わせることです。

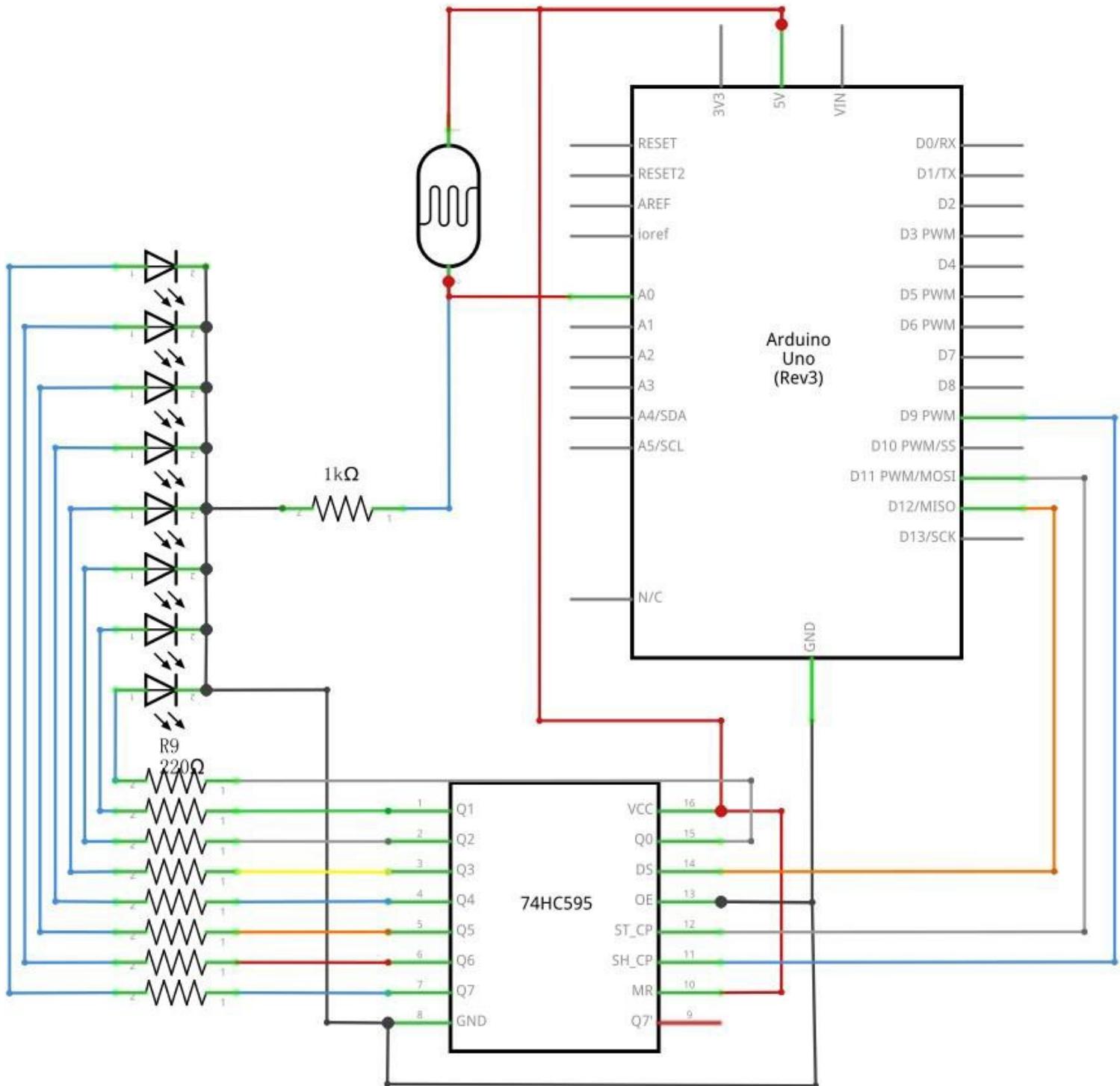


抵抗と光電池は一緒にポットのように振る舞います。光が非常に明るい場合、光電池の抵抗は固定値の抵抗と比較して非常に低く、ポットが最大になったかのようです。

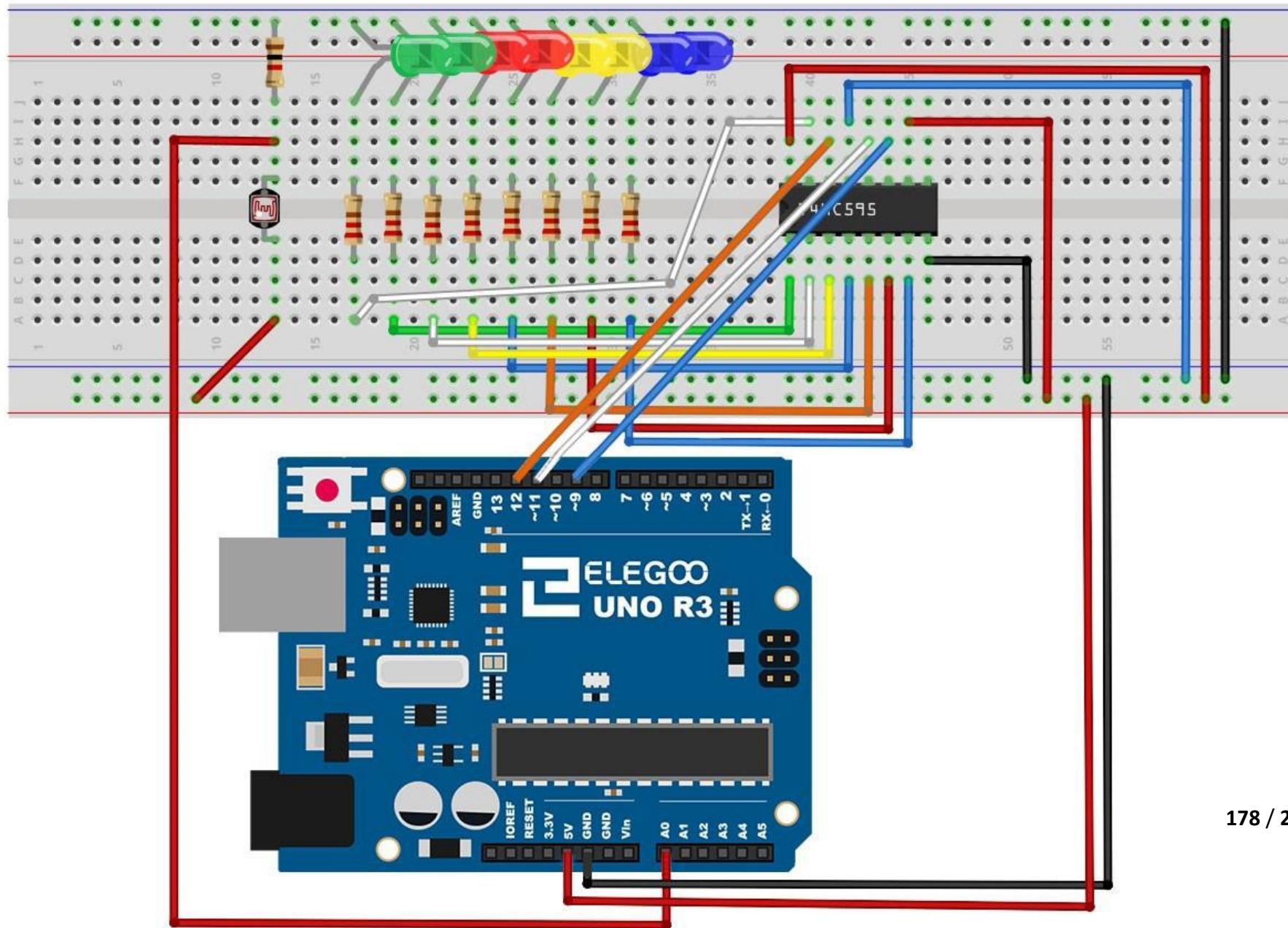
フォトセルが鈍い光の中にあるとき、抵抗は固定された $1\text{k}\Omega$ の抵抗よりも大きくなり、ポットが GND に向かって回転しているかのようになります。

次のセクションにあるスケッチをロードして、光電管を指で覆い、光源の近くに保持してみてください。

Connection Schematic



Wiring diagram



Code

配線後、プログラムをコードフォルダ - レッスン 26 フォトセルで開き、アップロードをクリックしてプログラムをアップロードしてください。エラーがある場合のプログラムアップロードの詳細については、[レッスン 2 を参照してください。](#)

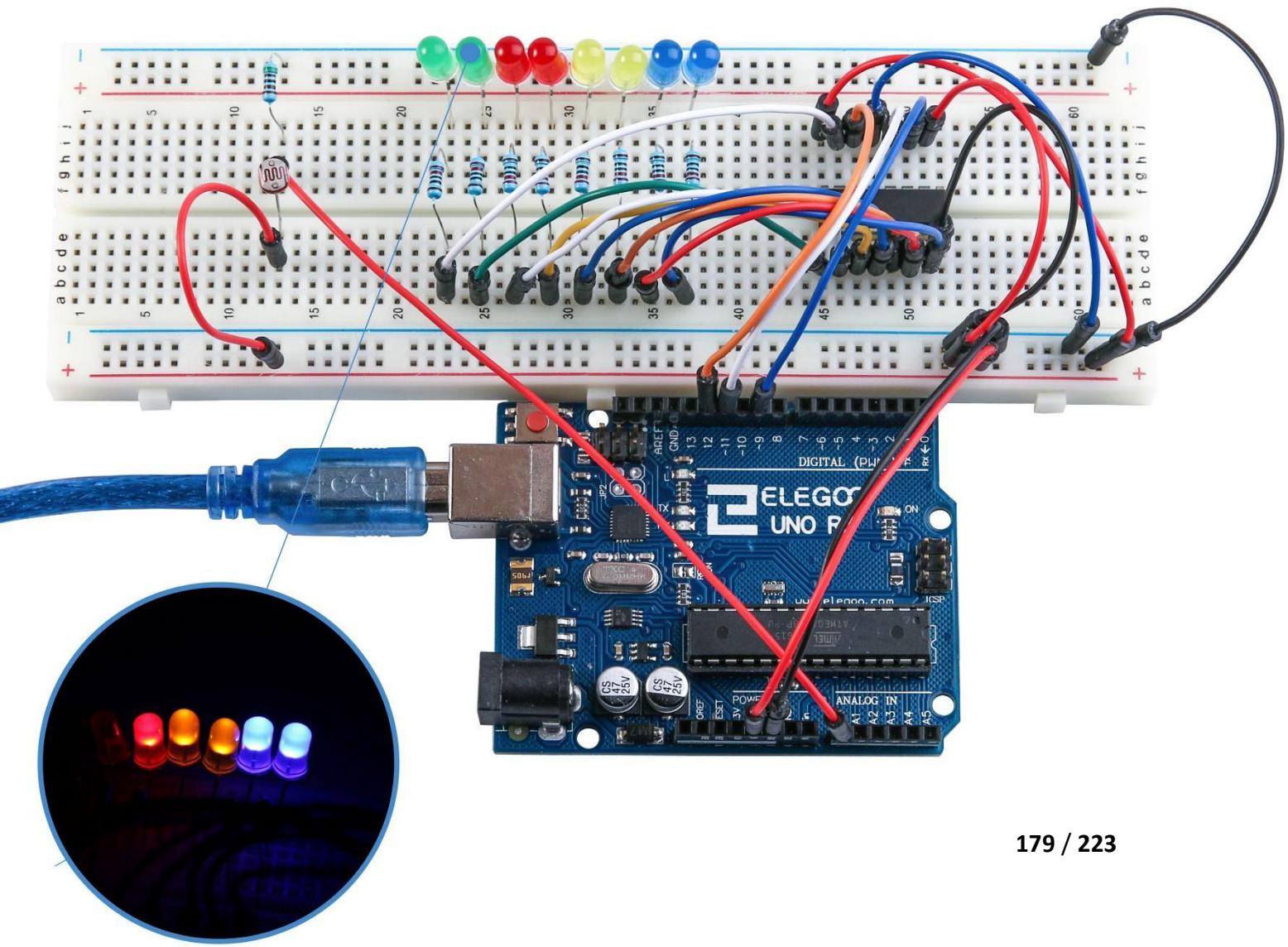
最初に留意すべき点は、アナログピンの名前を `potPin` ではなく `'lightPin'` に変更したことです。ポットが接続されていないためです。

スケッチに対する唯一の他の実質的な変更は、点灯する LED の数を計算する線です:

```
int numLEDSLit = reading / 57; // all LEDs lit at 1k
```

今回は生の読みを 114 ではなく 57 で割っています。換言すると、それをポットで半分に分けて LED を 9 つのゾーンに分割します。この余分な要因は固定 $1\text{k}\Omega$ の抵抗を考慮することです。これは、光電池の抵抗が $1\text{k}\Omega$ (固定抵抗と同じ) である場合、生の読み取り値は $1023/2 = 511$ になります。これは点灯しているすべての LED と同じになり、ビット (`numLEDSLit`) が 8。

Example picture



Lesson 19 74HC595 と 7 セグメント表示

概要

レッスン 16,17 と レッスン 18 を学習した後、74HC595 シフトレジスタを使用してセグメント表示を制御します。セグメント表示には 9~0 の数字が表示されます。

必要な構成部品:

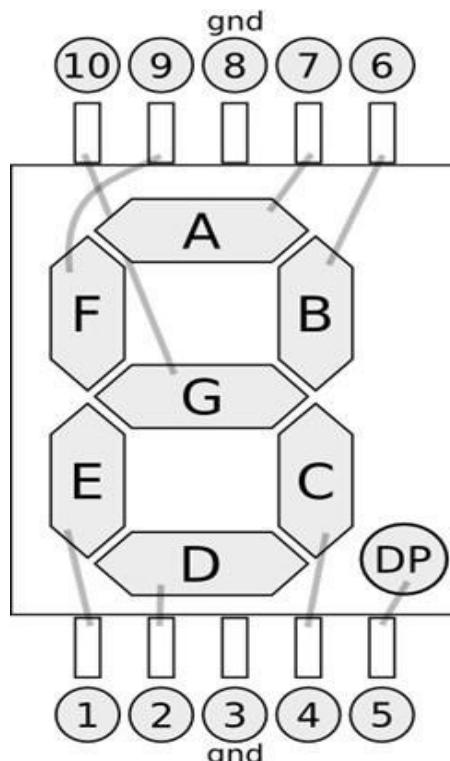
- (1) x Elegoo Uno R3
- (1) x 830 tie-points breadboard
- (1) x 74HC595 IC
- (1) x 1 Digit 7-Segment Display
- (8) x 220 ohm resistors
- (26) x M-M wires (Male to Male jumperwires)

部品の紹介

Seven segment display



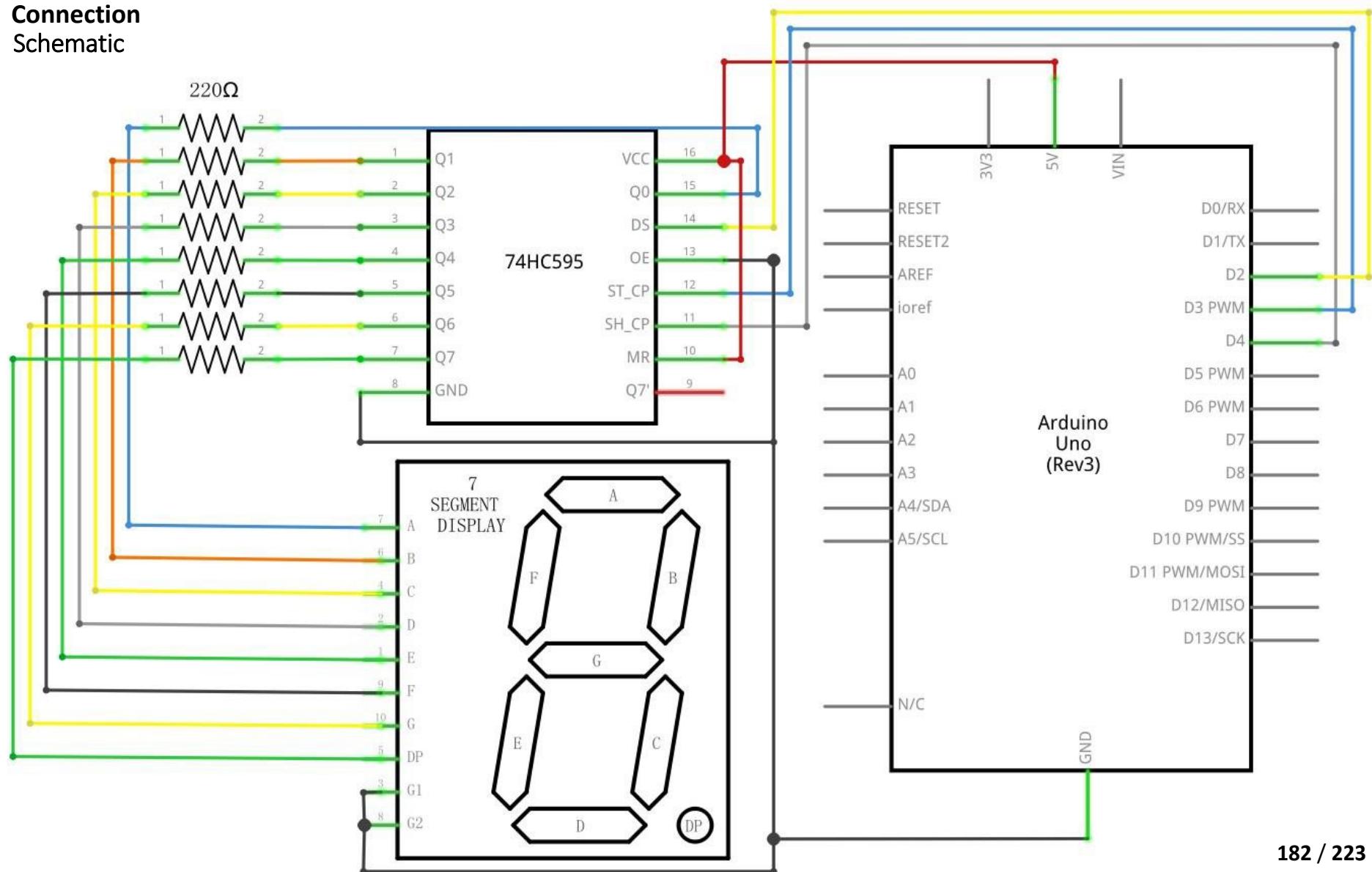
以下は 7 セグメントのピン図です。



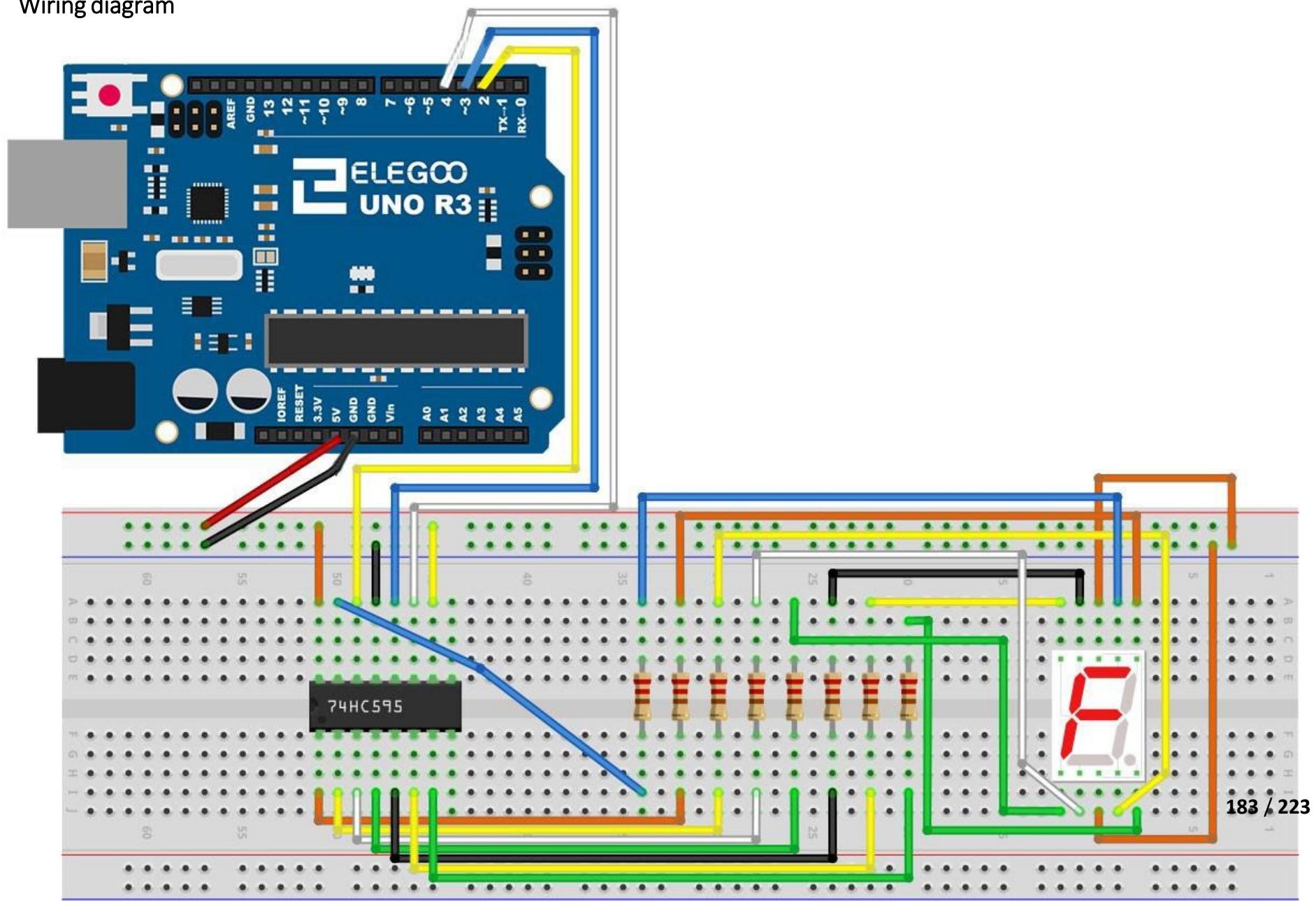
0-9 9桁の各セグメントに対応しています（次の表は、共通の陽極を使用している場合は、共通の陰極7セグメント表示デバイスを適用します。表はすべて100をすべて1に置き換える必要があります）：

Display digital	dp	a	b	c	d	e	f	g
0	0	1	1	1	1	1	1	0
1	0	0	1	1	0	0	0	0
2	0	1	1	0	1	1	0	1
3	0	1	1	1	1	0	0	1
4	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1
6	0	1	0	1	1	1	1	1
7	0	1	1	1	0	0	0	0
8	0	1	1	1	1	1	1	1
9	0	1	1	1	1	0	1	1

Connection Schematic



Wiring diagram



次の表は、7セグメントディスプレイ 74HC595 ピン対応表を示しています:

74HC595 pin	Seven shows remarkable control pin (stroke)
Q0	7 (A)
Q1	6 (B)
Q2	4 (C)
Q3	2 (D)
Q4	1 (E)
Q5	9 (F)
Q6	10 (G)
Q7	5 (DP)

ステップ 1: 74HC595 を接続する

最初に、配線は電源とグランドに接続されます:

VCC (ピン 16) と **MR** (ピン 10) は 5V に接続

GND (ピン 8) と **OE** (ピン 13) はグランドに接続

接続 **DS**、**ST_CP**、**SH_CP** ピン:

DS (ピン 14) は UNO R3 ボードのピン 2 (黄色の線の下の図) に接続

ST_CP (ピン 12、ラッチピン) は UNOR3 ボードのピン 3 (下の青のライン) に接続

SH_CP (ピン 11、クロックピン) は UNO R3 ボードのピン 4 (白い線の下の図) に接続

ステップ 2: 7セグメントディスプレイを接続する

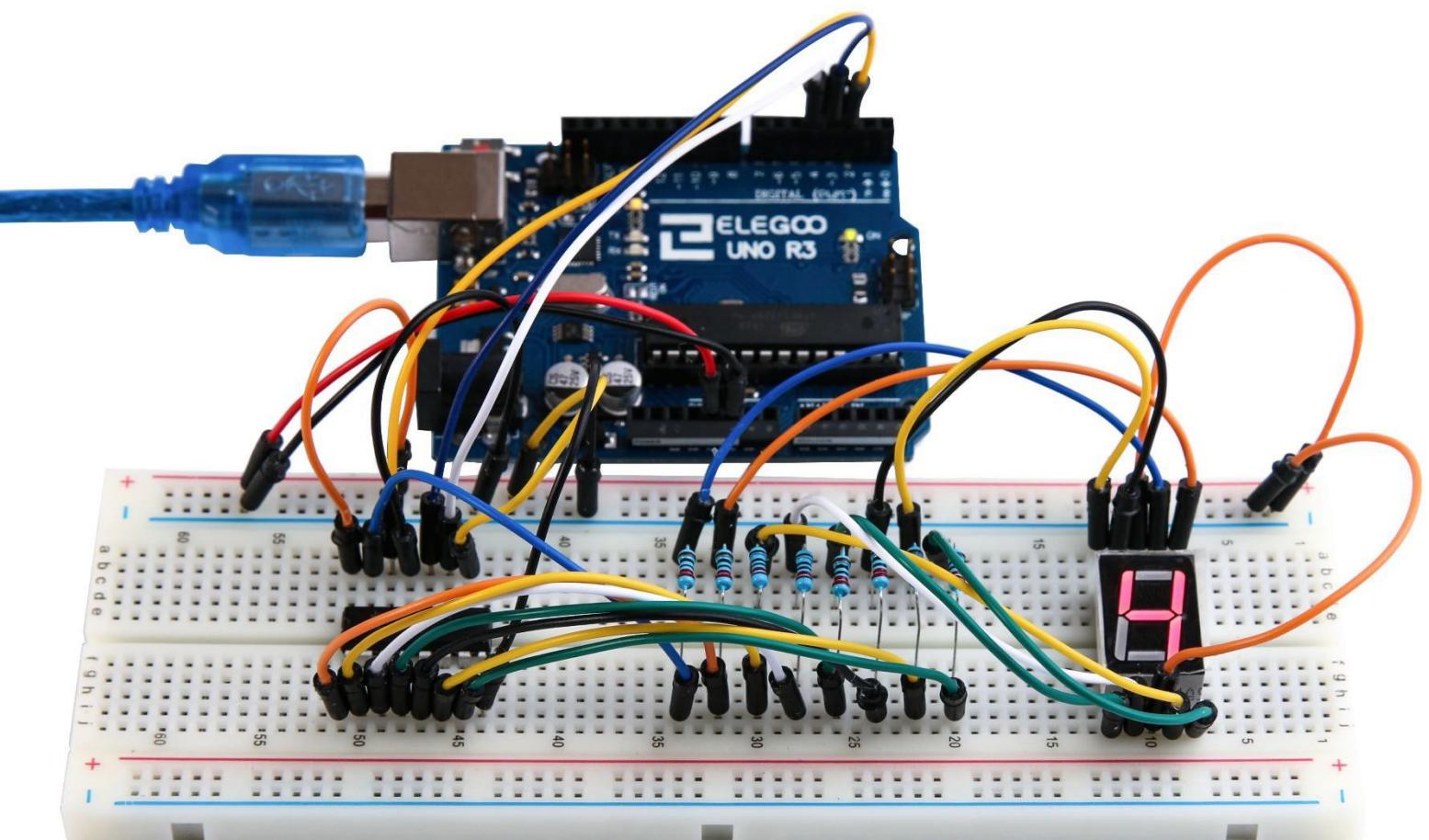
7セグメントディスプレイ 3,8 ピンから UNO R3 ボード GND (この例では共通のアノードを使用していますが、共通のアノードを使用する場合は 3,8 ピンを UNO R3 ボード + 5V に接続してください)

上記の表に従って、74HC595 の Q0～Q7 を 7セグメントディスプレイの対応するピン (A～G および DP) に接続し、次に各足を 220 オームの抵抗器に直列に接続します。

Code

配線後、コードフォルダのレッスン 27 の 74HC595 とセグメント表示でプログラムを開き、UPLOAD をクリックしてプログラムをアップロードしてください。エラーがある場合のプログラムアップロードの詳細については、レッスン 2 を参照してください。

Example picture



Lesson 20 4つのデジタル 7 セグメント表示

概要

このレッスンでは、4 枠の 7 セグメント表示を使用する方法を学習します。

1 枠の 7 セグメントディスプレイを使用する場合、共通のアノードであれば、共通のアノードピンが電源に接続されています。 それが共通陰極である場合、共通陰極ピンは GND に接続する。

4 枠の 7 セグメントディスプレイを使用する場合、共通のアノードまたは共通のカソードピンを使用して、表示する桁を制御します。 Persistence of Vision の原則は、1 枠の数字が働いているにもかかわらず、各走査速度が非常に速いために表示されるすべての数字を見ることができません。

必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x 830 tie-points breadboard
- (1) x 74HC595 IC
- (1) x 4 Digit 7-Segment Display
- (4) x 220 ohm resistors
- (23) x M-M wires (Male to Male jumper wires)

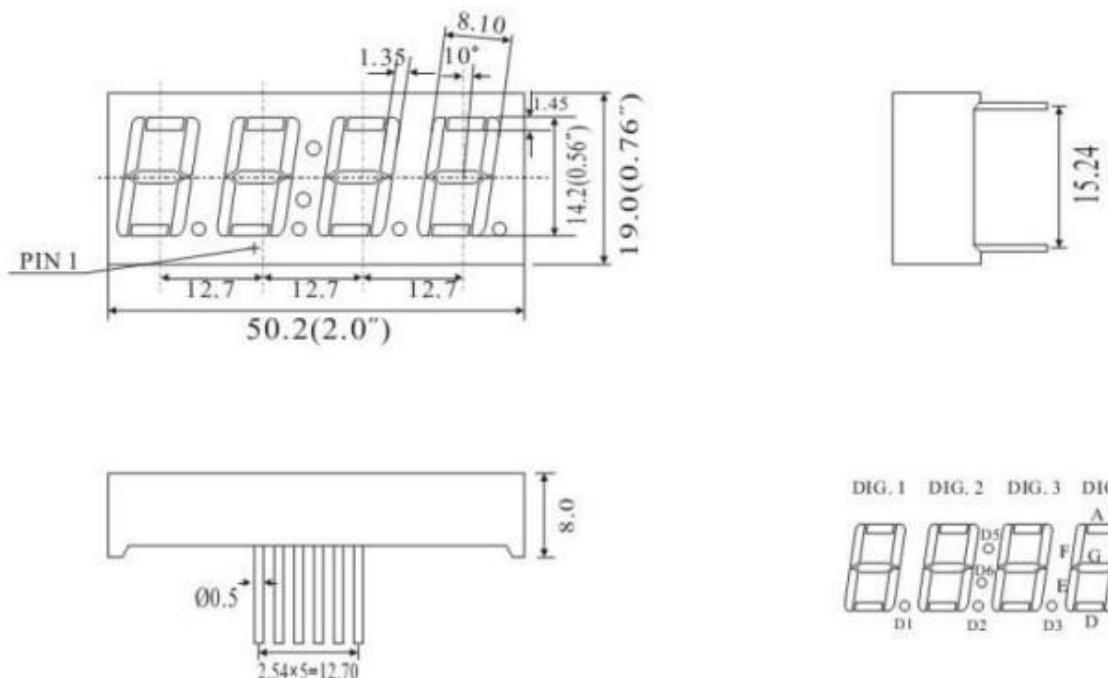


部品の紹介

Four Digital Seven segment display

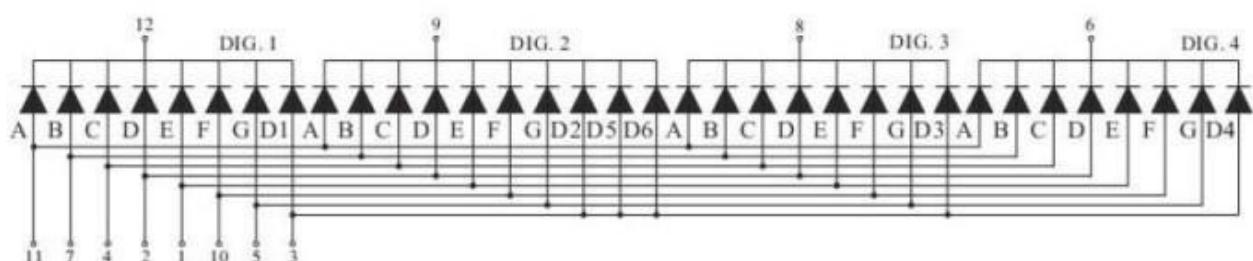
Package Dimensions

CPS05643AB

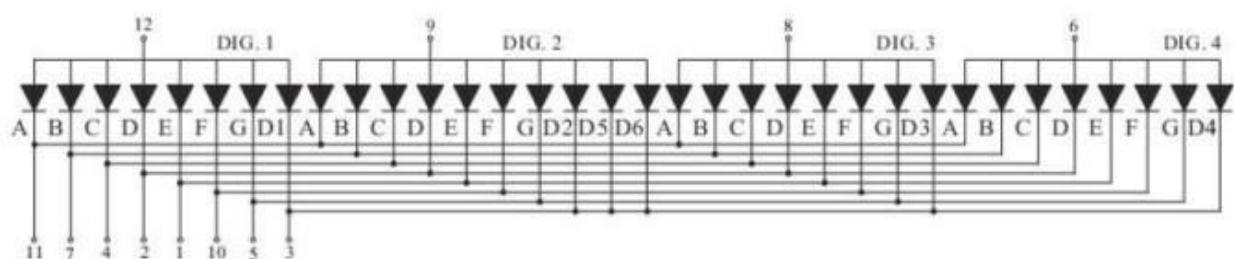


UNIT: MM(INCH) TOLERANCE: $\pm 0.25(0.01")$

Internal Circuit Diagram



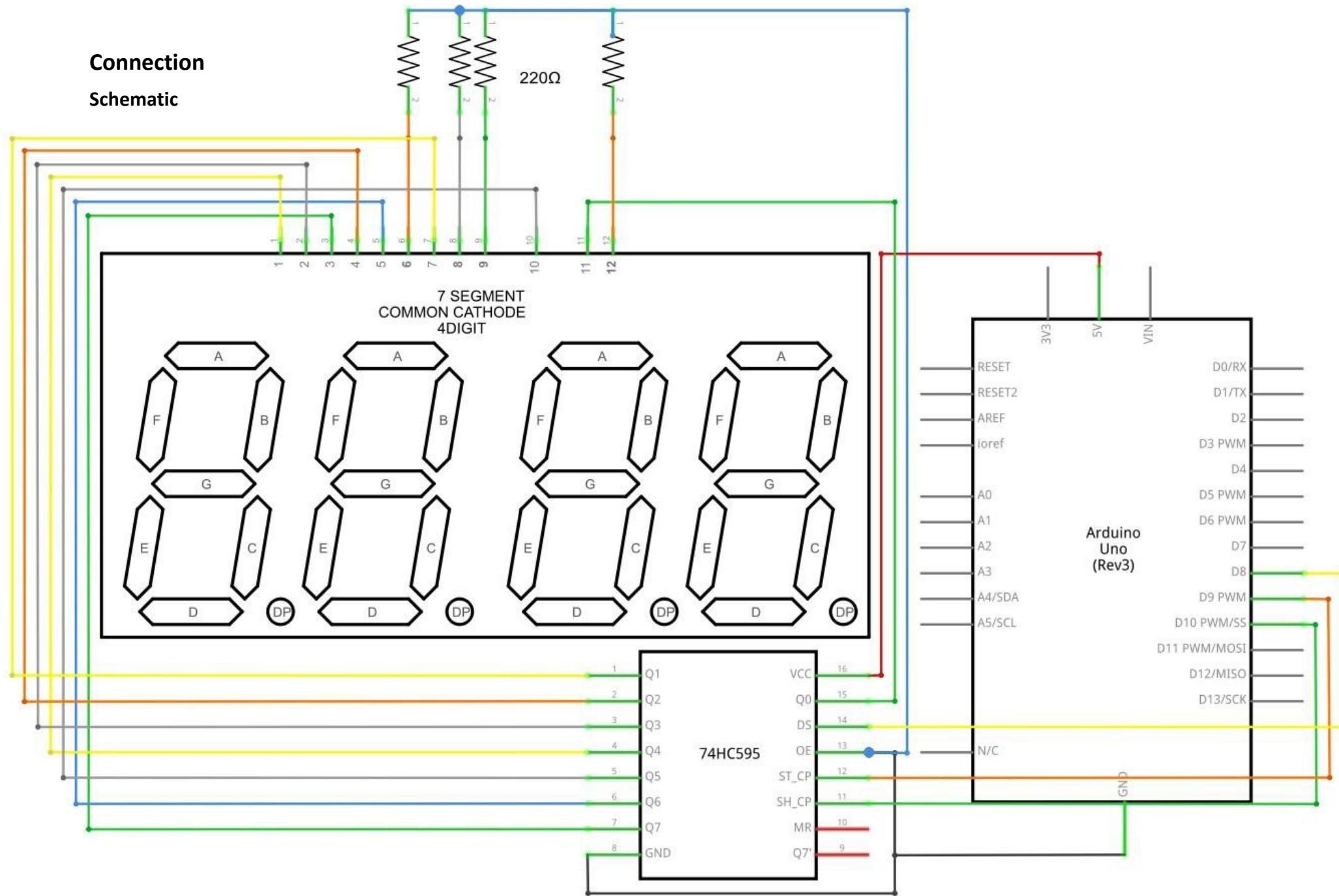
5643A



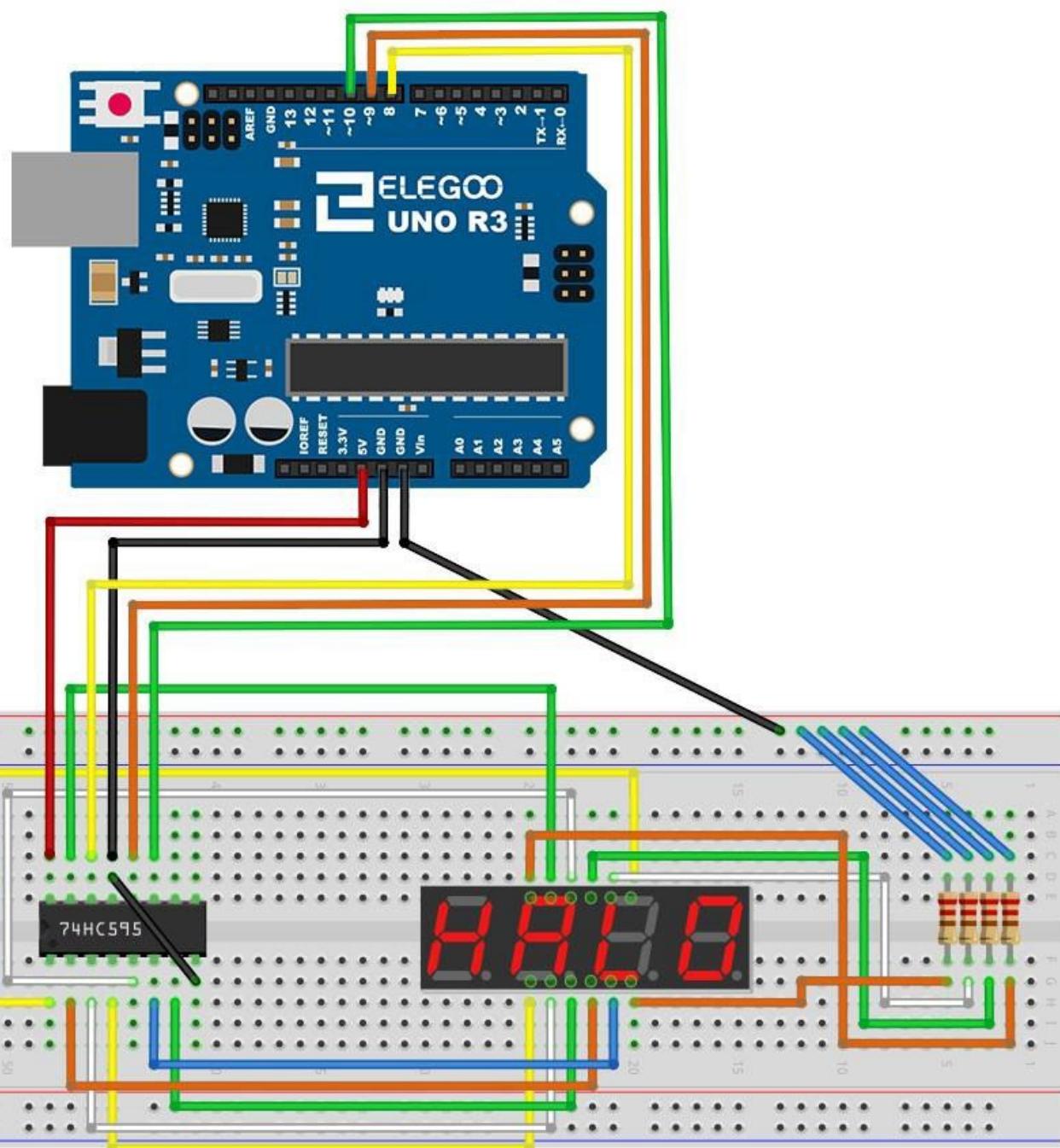
5643B

Four Digits Displays Series

Connection Schematic



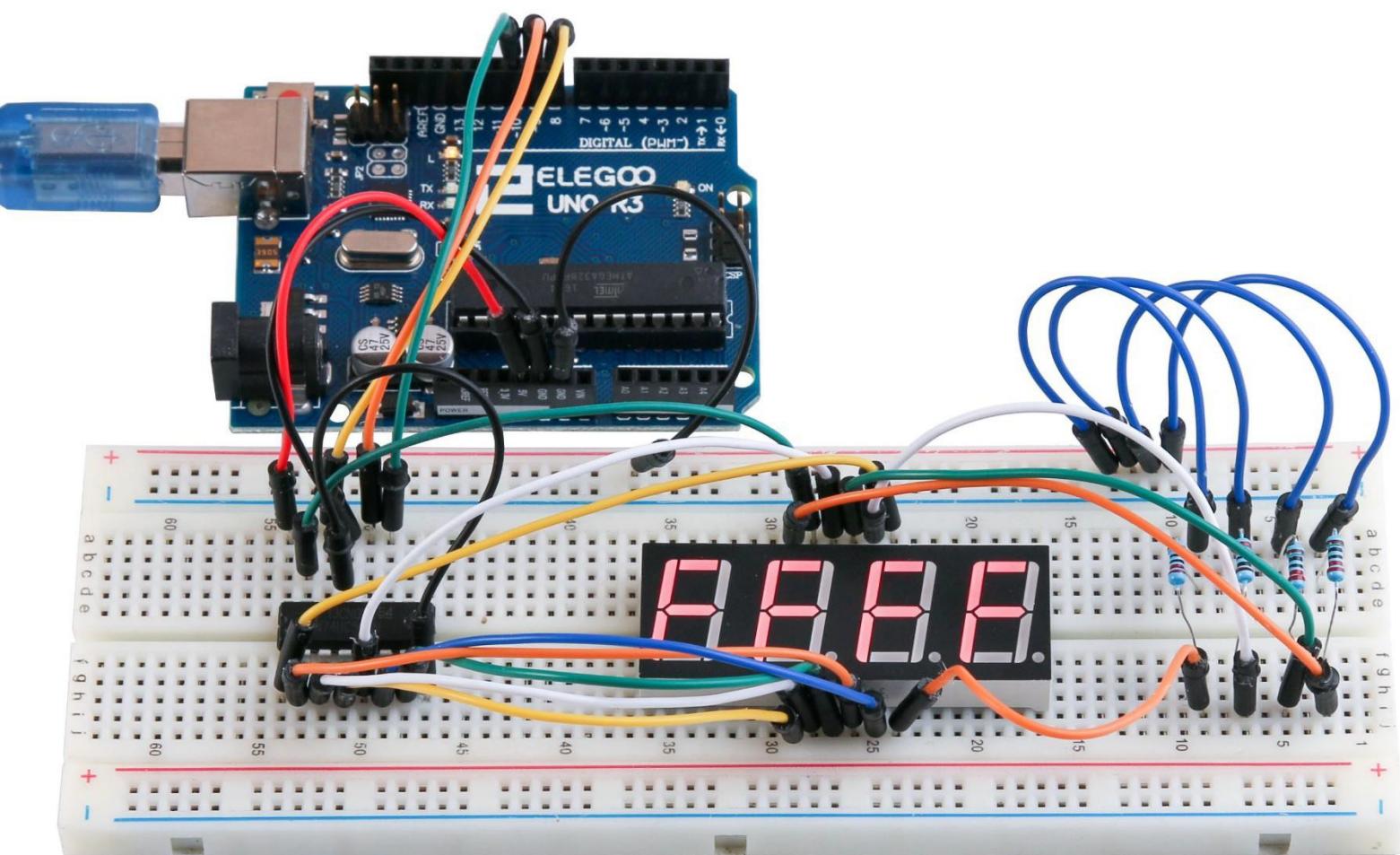
Wiring diagram



Code

配線後、コードフォルダのレッスン 28 の 4つのデジタルセブンセグメントディスプレイでプログラムを開き、UPLOAD をクリックしてプログラムをアップロードしてください。 エラーがある場合のプログラムアップロードの詳細については、レッスン 2 を参照してください。

Example picture



Lesson 21 直流モータ

概要

このレッスンでは、UNO R3 とトランジスタを使用して小型 DC モーターを制御する方法を学習します。

必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x 830 tie-points breadboard
- (1) x L293D IC
- (1) x Fan blade and 3-6v motor
- (5) x M-M wires (Male to Male jumperwires)
- (1) x Power Supply Module
- (1) x 9V1A adapter

部品の紹介

Breadboard Power Supply

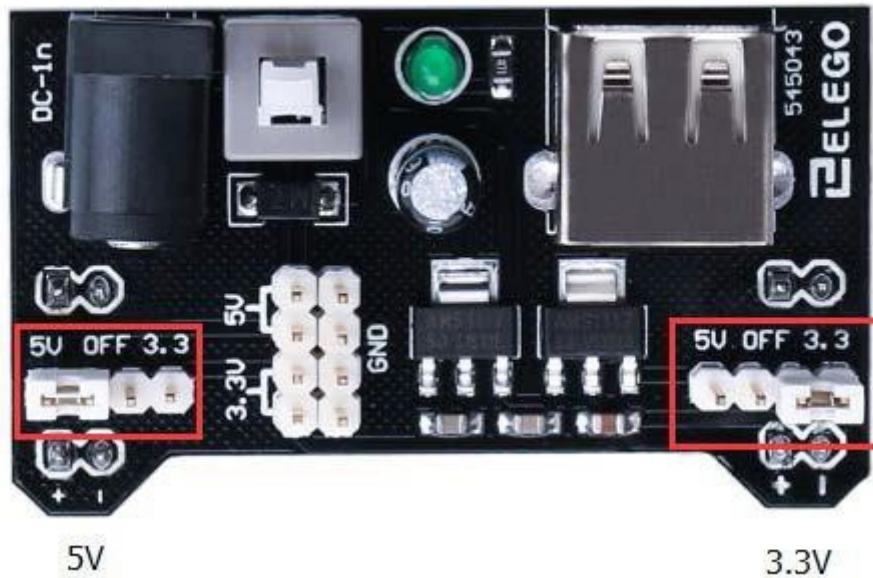
小型の DC モーターは、UNO R3 ボードのデジタル出力よりも多くの電力を直接使用できます。モーターを UNO R3 ボードのピンにまっすぐ接続しようとすると、UNO R3 ボードに損傷を与える可能性があります。 私たちは電源モジュールを使用して電源を供給します。



製品仕様:

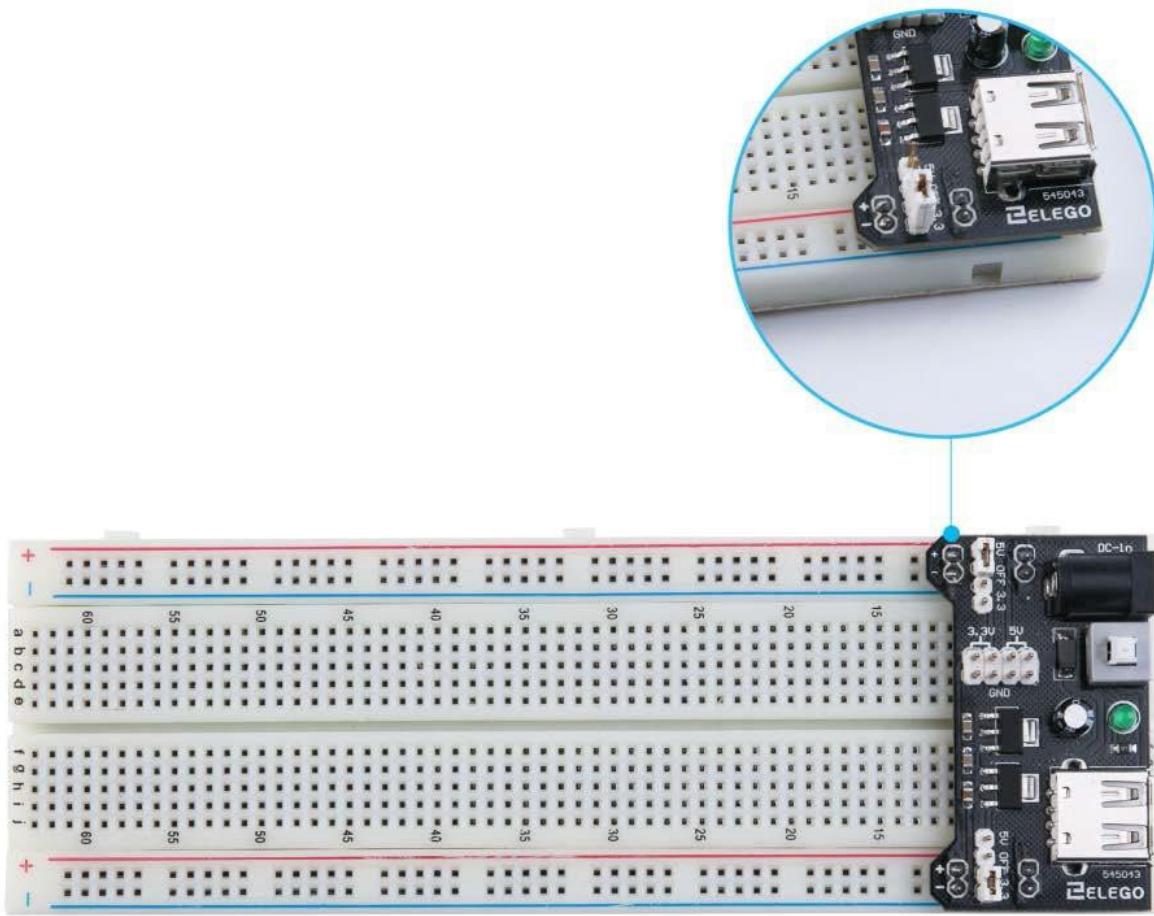
- Locking On/Off Switch
- LED Power Indicator
- Input voltage: 6.5-9v (DC) via 5.5mm x 2.1mm plug
- Output voltage: 3.3V/5v
- Maximum output current: 700mA
- 独立した制御レール出力 0v, 3.3v, 5v to breadboard
- 外部使用に便利な出力
- Size: 2.1 in x 1.4 in
- 外部デバイスに電源を供給するためのオンボード USB デバイスコネクタ

Setting up output voltage:



左右の電圧出力を独立して設定することができます。出力電圧を選択するには、対応するピンにジャンパーを接続します。

注: 両方のジャンパーが "OFF" の位置にある場合、電源インジケーターLED とブレッドボード電源レールの電源は入らない。



Important note:

ブレッドボードに正しくモジュールが挿していることを確認してください。モジュールの負のピン（-）は、ブレッドボードの青い線（-）と並んでおり、正のピン（+）が赤い線（+）で上に並んでいます。そうしないと、誤ってプロジェクトの電源を逆にする可能性があります

L293D

これは非常に便利なチップです。実際には2つのモータを独立して制御することができます。このレッスンではチップの半分だけを使用しています。チップの右側のピンの大部分は、第2のモーターを制御するためのものです。



製品仕様:

- テキサスインスツルメンツの Unitrode L293 と L293D 製品を特長とする
- 広い電源電圧範囲: 4.5 V to 36 V
- 入力ロジック電源を分離する
- 内部 ESD 保護
- 過熱遮断
- 高ノイズ耐性入力
- SGS L293 および SGS L293D と機能的に類似
- 出力電流 1 A Per Channel (600 mA for L293D)
- ピーク出力電流: 2 A / チャンネル(1.2 A for L293D)
- 誘導性過渡抑制用出力クランプダイオード(L293D)

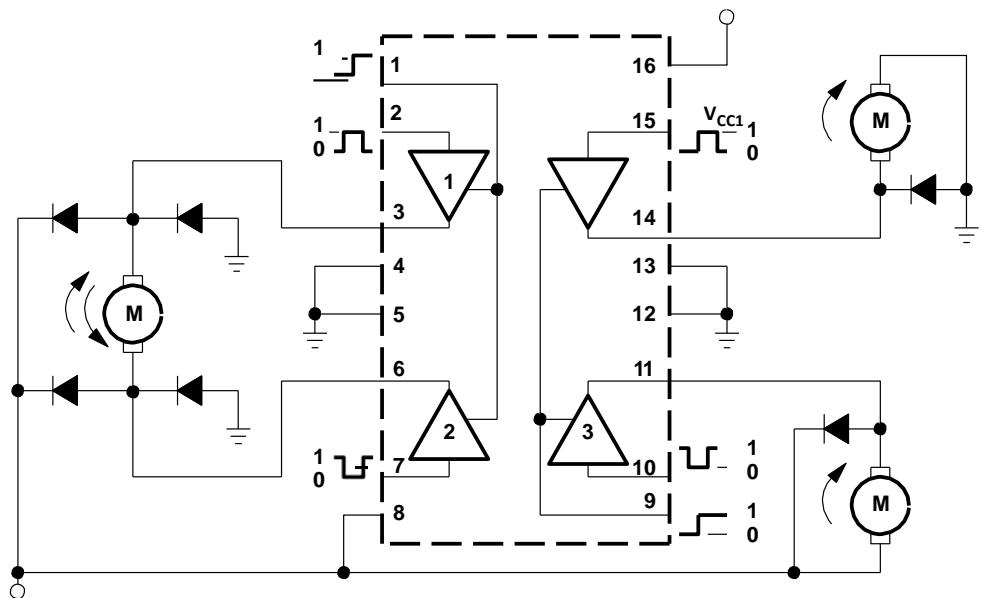


説明/発注情報

L293 と L293D は 4 倍の高電流のハーフ H ドライバです。 L293 は、4.5V~36V の電圧で最大 1A の双方向駆動電流を供給するように設計されています。L293D は、4.5V~36V の電圧で最大 600mA の双方向駆動電流を提供するように設計されています。リレー、ソレノイド、DC およびバイポーラステッピングモーターなどの誘導負荷、および正電源アプリケーションにおける他の高電流/高電圧負荷を駆動するように設計されています。

すべての入力は TTL 互換です。各出力は、完全なトーテムポール駆動回路であり、ダーリントントランジスタシングルと疑似ダーリントンソースを備えています。ドライバはペアで有効になり、1,2EN でドライバ 1 と 2 を有効にし、3,4EN でドライバ 3 と 4 を有効にします。イネーブル入力がハイのとき、関連ドライバはイネーブルされ、その出力はアクティブで入力と同相です。イネーブル入力がローのとき、これらのドライバはディセーブルされ、出力はオフになり、ハイ・インピーダンス状態になります。適切なデータ入力によって、各ドライバ対は、ソレノイドまたはモータ用途に適したフル H (またはブリッジ) 可逆ドライブを形成する。

Block diagram



V_{CC2}

私は、データシート内の解説可能なピン配置図に飽きてしまったので、もっと関連性の高い情報が得られると私自身が設計しました。

Arduino に接続されている 3 本のワイヤー、モーターに接続されている 2 本のワイヤー、バッテリーに接続されている 1 本のワイヤーがあります。

L293D

M1 PWM	1	16	Battery +ve
M1 direction 0/1	2	15	M2 direction 0/1
M1 +ve	3	14	M2 +ve
GND	4	13	GND
GND	5	12	GND
M1 -ve	6	11	M2 -ve
M1 direction 1/0	7	10	M2 direction 1/0
Battery +ve	8	9	M2 PWM

Motor 1

Motor 2

このピン配置の使用方法:

左側は第 1 のモータを処理し、右側は第 2 のモータを処理する。 はい、あなたは 1 台のモーターだけを接続して走らせることができます。

Arduino 接続

M1 PWM - これを Arduino の PWM ピンに接続します。それらは Uno でラベル付けされています、ピン 5 は例です。

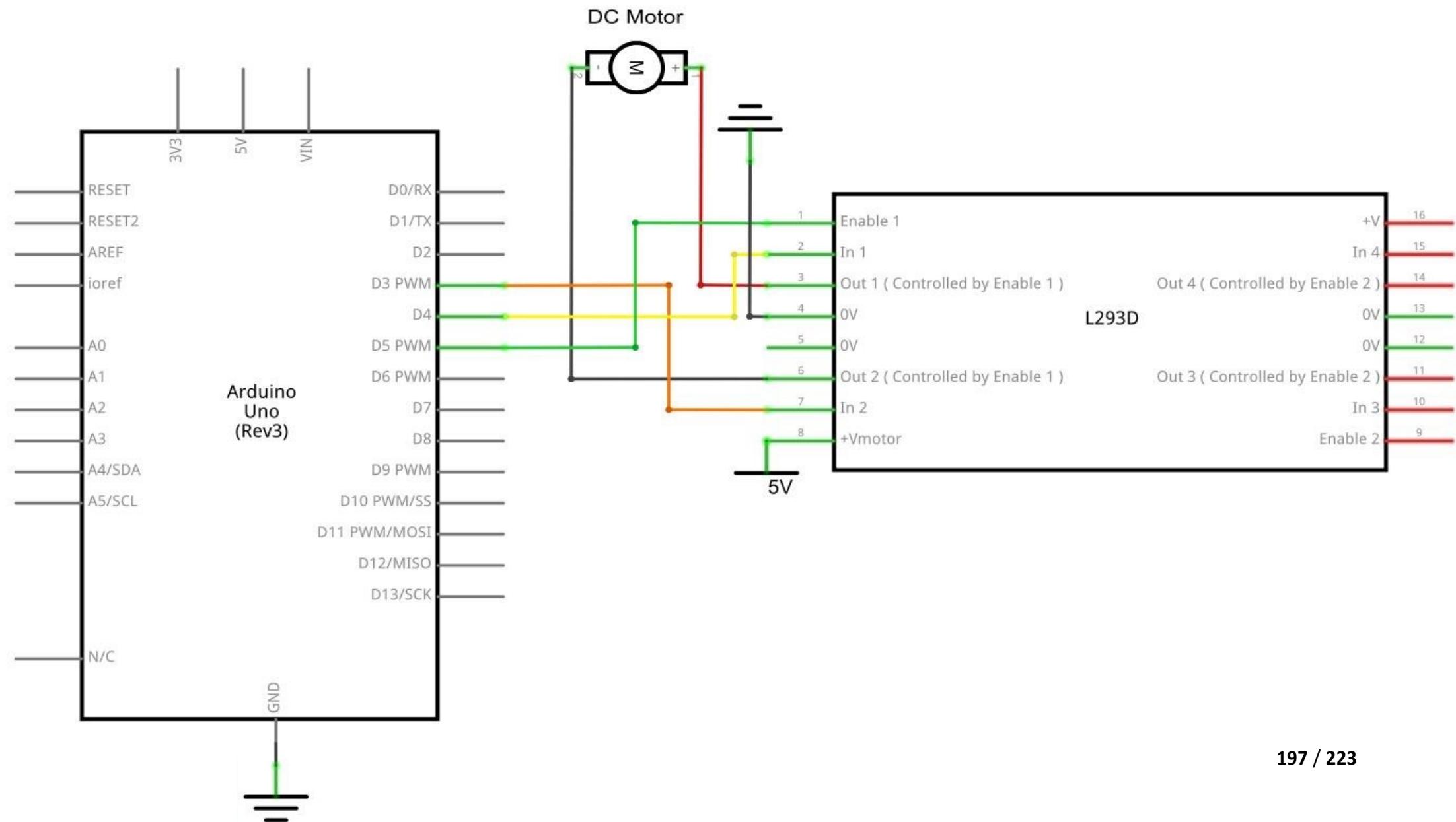
0 から 255 までの任意の整数を出力します。0 はオフ、128 は半速度、255 は マックス・スピード。

M1 方向 0/1 および M1 方向 1/0 - これらの 2 つを 2 つのデジタル Arduino ピンに接続します。 1 ピンを HIGH、他の

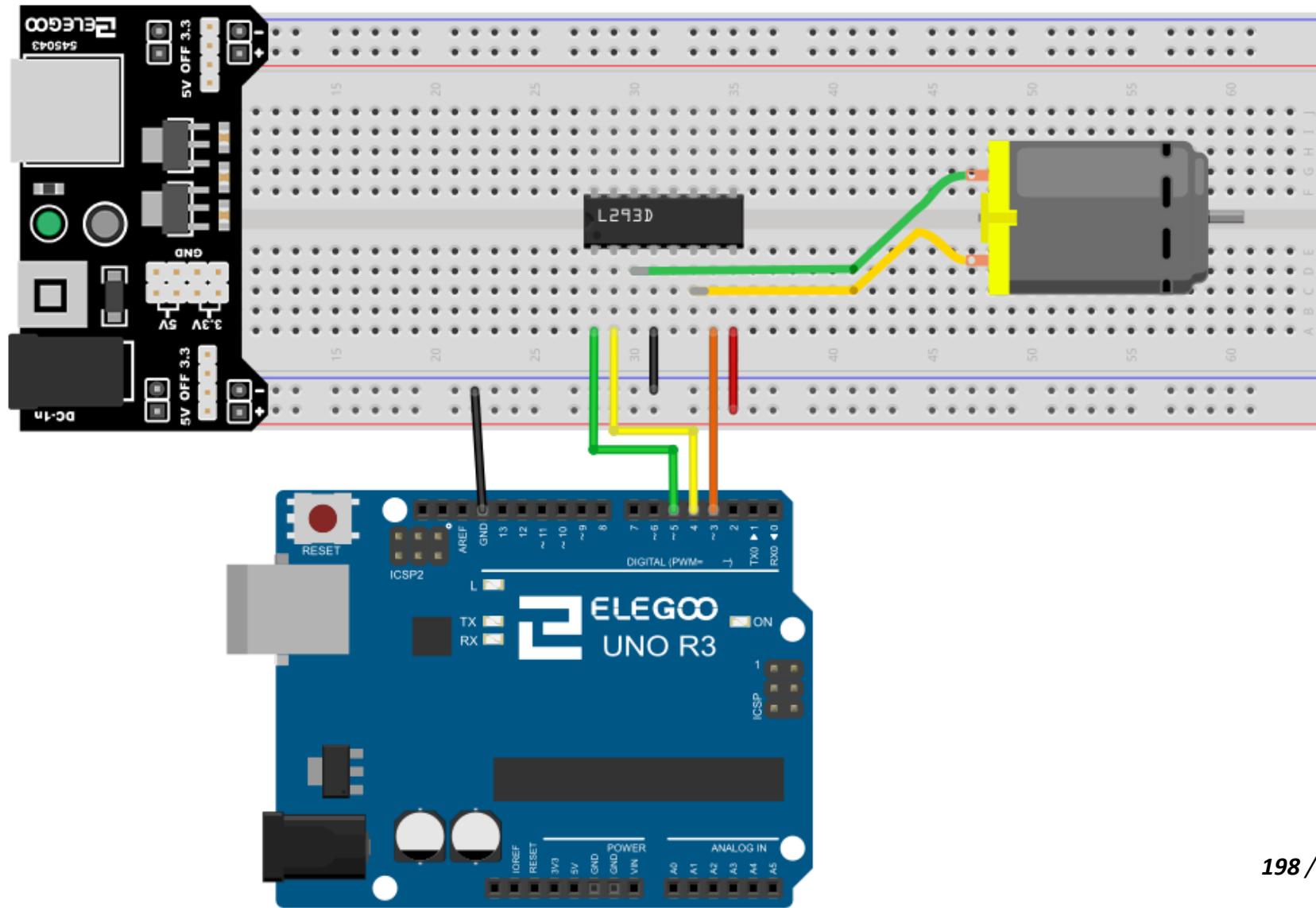
ピンを LOW として出力すると、モーターは一方向に回転します。

出力を LOW と HIGH に反転すると、モーターは反対方向に回転します。

Connection

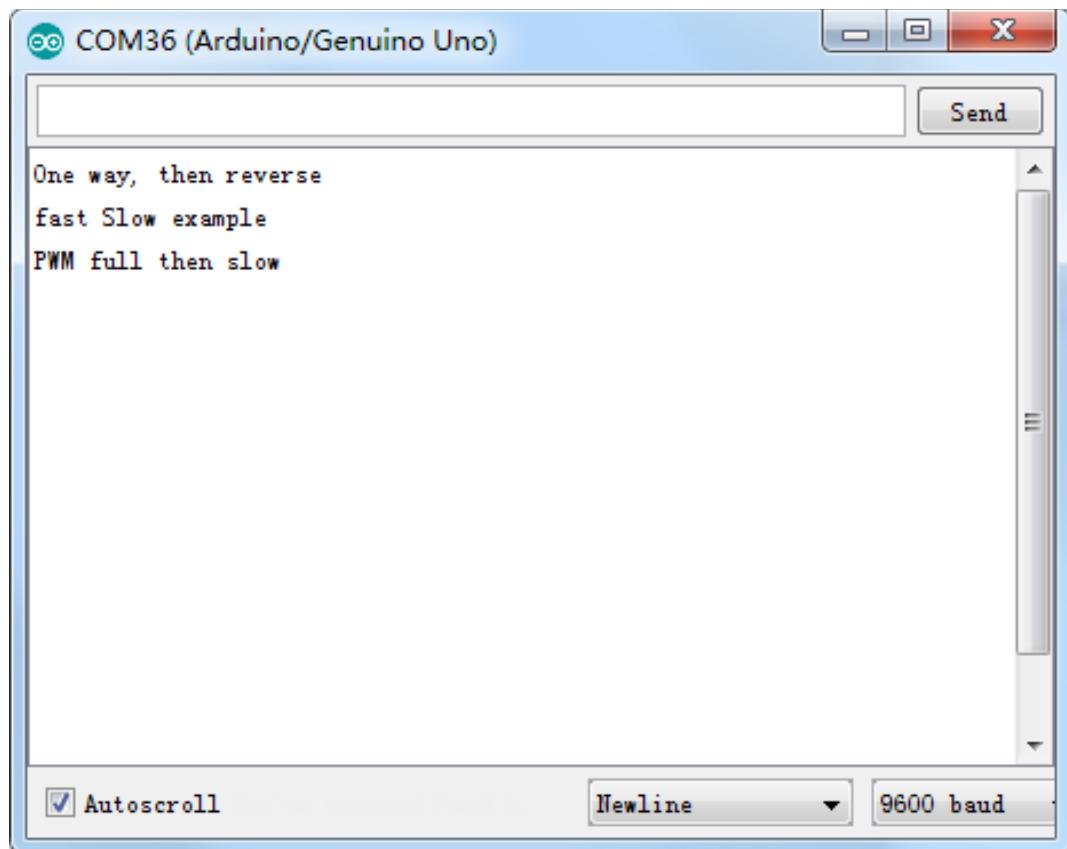


Wiring diagram



以下のコードでは、別の電源（バッテリ）を使用していませんが、代わりに Arduino の 5V 電源を使用しています。これは L293D がそれを制御しなければ危険であることに注意してください。

モーターを Arduino に直接接続する必要はありません。なぜなら、モーターを切って電気的フィードバックを得るからです。小さなモーターでは、これはあなたの Arduino を傷つけ、大きなモーターでは、興味深い炎と火花の効果を見ることができます。

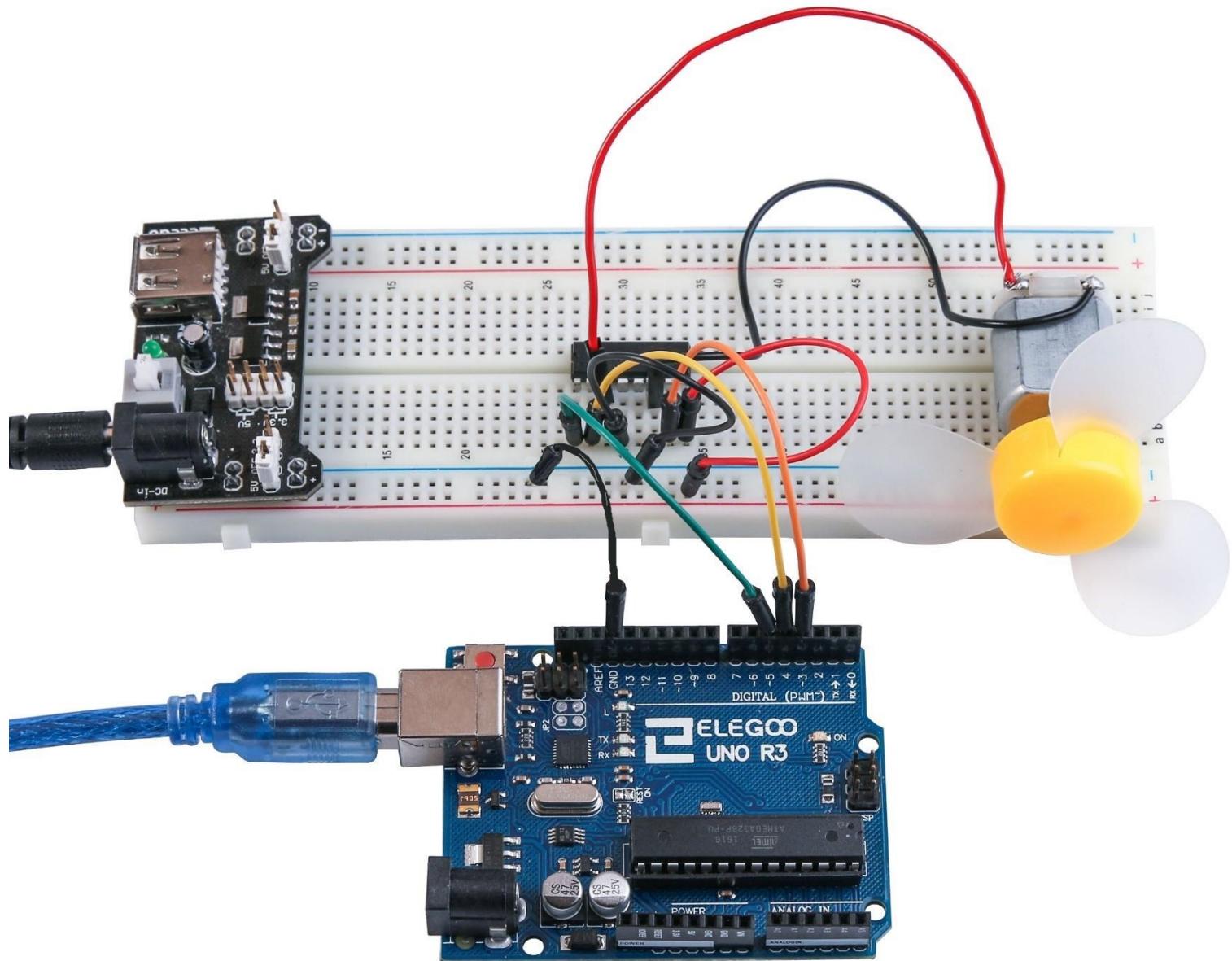


Code

配線後、プログラムをコードフォルダ - レッスン 29 DC モータで開き、アップロードをクリックしてプログラムをアップロードしてください。エラーがある場合のプログラムアップロードの詳細については、レッスン 2 を参照してください。

プログラムのロード後、すべての電源スイッチをオンにします。モーターは時計回りと反時計回りに 5 回少しほ回転します。その後、時計回りに劇的に回転し続けます。短い休止後、劇的に反時計回りに回転します。コントローラボードは PWM 信号を送信してモータを駆動し、モータはその最大 RPM を最小値までゆっくりと下げ、再び最大値まで上昇します。最後に、次のサイクルが始まるまで 10 秒間停止します。

Example picture



Lesson 22 リレー

概要

このレッスンでは、リレーの使用方法を学習します。

必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x 830 tie-points breadboard
- (1) x Fan blade and 3-6v dc motor
- (1) x L293D IC
- (1) x 5v Relay
- (1) x Power Supply Module
- (1) x 9V1A Adapter
- (8) x M-M wires (Male to Male jumper wires)



部品の紹介

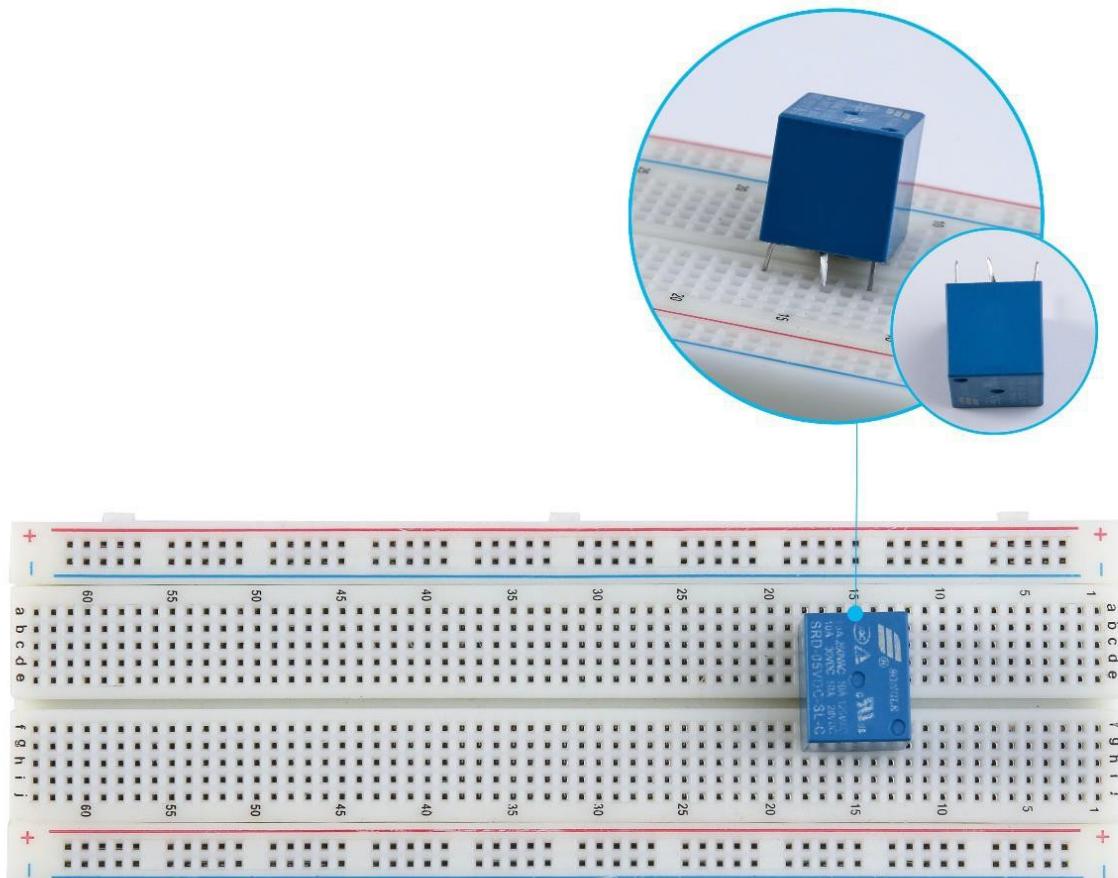
Relay:

リレーは電気的に操作されるスイッチである。多くのリレーは電磁石を使用してスイッチを機械的に動作させますが、他の動作原理もソリッドステートリレーのように使用されます。リレーは、低電力信号（制御と制御された回路の間の完全な電気的絶縁を使用）によって回路を制御する必要がある場合、または複数の回路を1つの信号で制御する必要がある場合に使用されます。第1のリレーは増幅器として長距離電信回路に用いられた。彼らは、ある回路から入ってきた信号を繰り返し、別の回路でそれを再送しました。リレーは、論理的な操作を行うために電話交換機や初期のコンピュータで広く使用されていました。

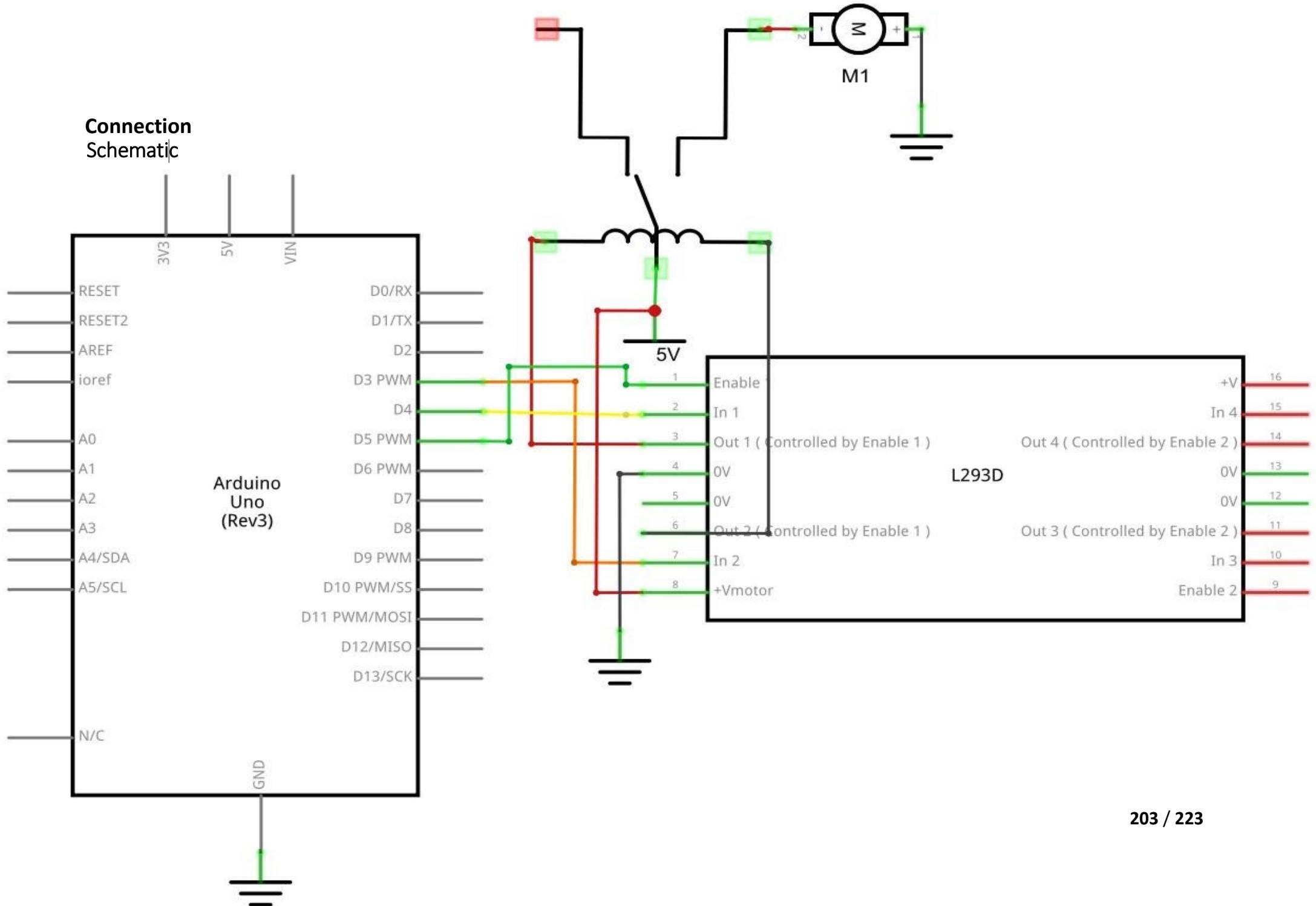
電気モータまたは他の負荷を直接制御するのに必要な高出力に対応できるタイプのリレーはコンタクタと呼ばれます。ソリッドステートリレーは、半導体デバイスを使用してスイッチングを実行する代わりに、可動部分のない電源回路を制御します。校正された動作特性を備えたリレーおよび場合によっては複数の動作コイルが、過負荷または過失から電気回路を保護するために使用されます。最新の電力システムでは、これらの機能は「保護リレー」と呼ばれるデジタル機器によって実行されます。

以下は Arduino でリレーを駆動する方法の模式図です。

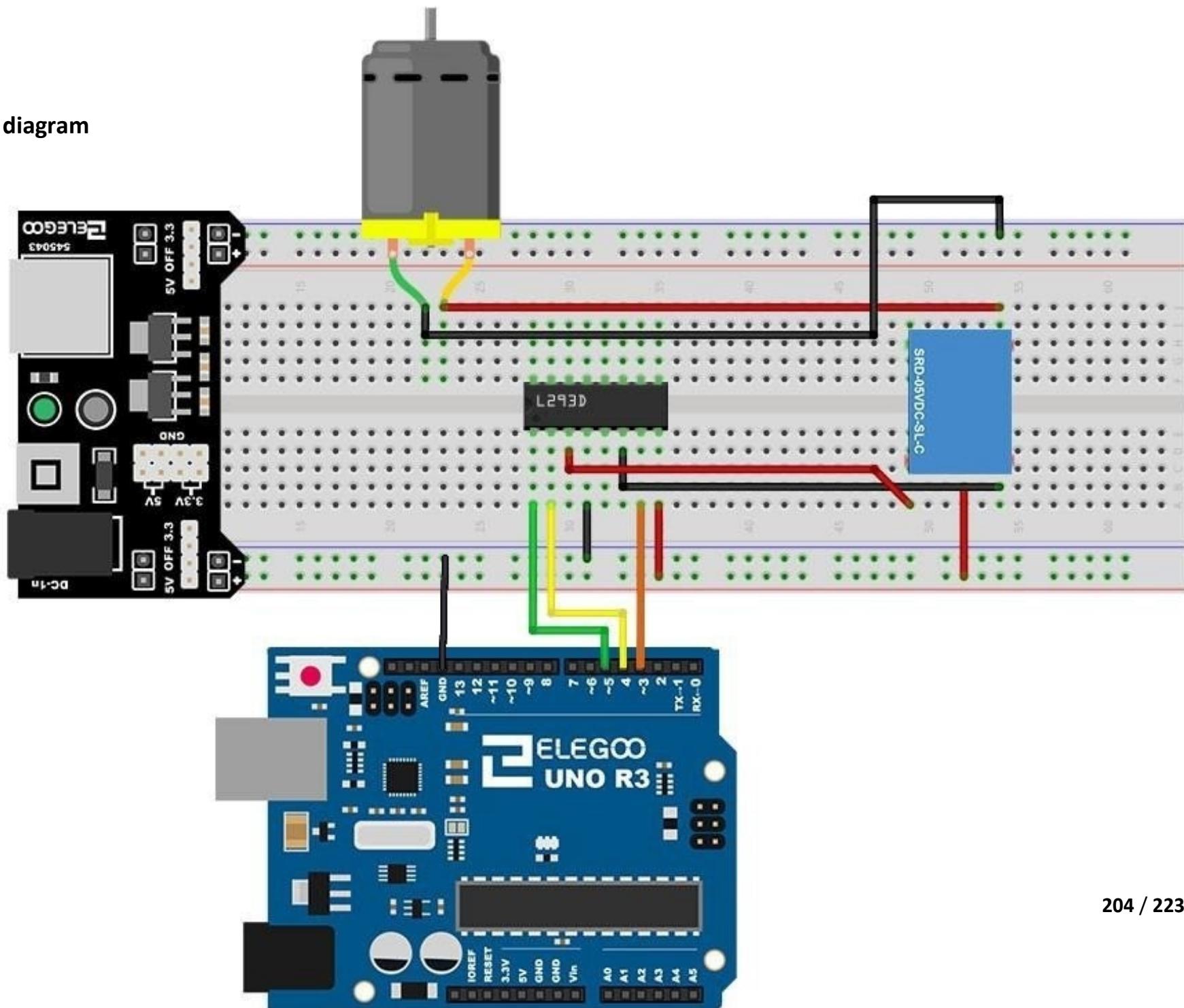
ブレッドボードにリレーを挿入する方法について混乱する可能性があります。下の図に示すように、リレーのピンの1つを少し曲げて、パンボードに挿入する必要があります。



Connection Schematic



Wiring diagram

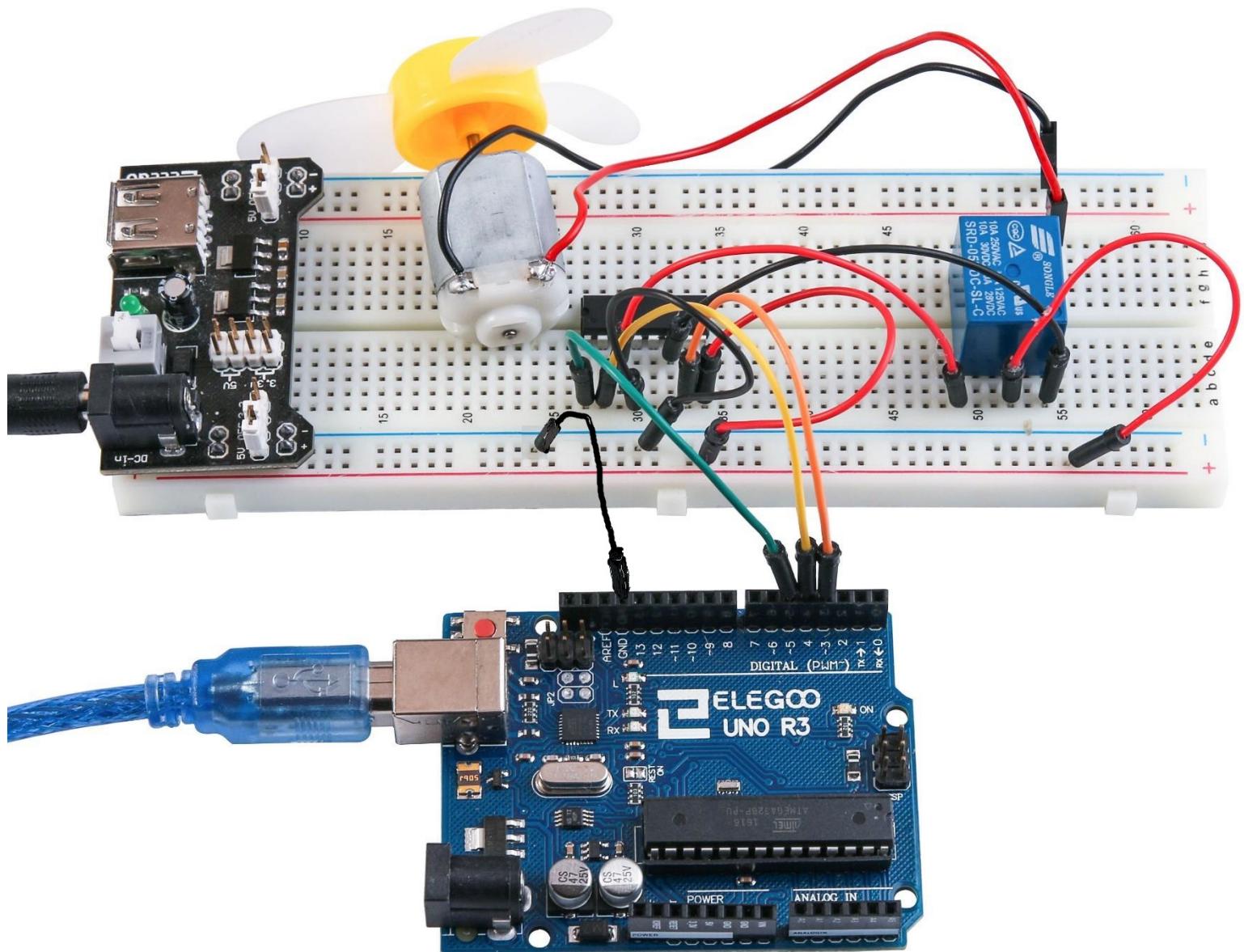


Code

配線後、コードフォルダのレッスン 30 リレーでプログラムを開き、UPLOAD をクリックしてプログラムをアップロードしてください。エラーがある場合のプログラムアップロードの詳細については、[レッスン 2 を参照してください。](#)

プログラムのロード後、すべての電源スイッチをオンにします。リレーは鳴る音でピックアップします。その後、モーターが回転します。一定時間後、リレーが解放され、モータが停止します。

Example picture



Lesson 23 ステッパーモーター

概要

このレッスンでは、ステッパーモーターを操作するための楽しく簡単な方法を学習します。

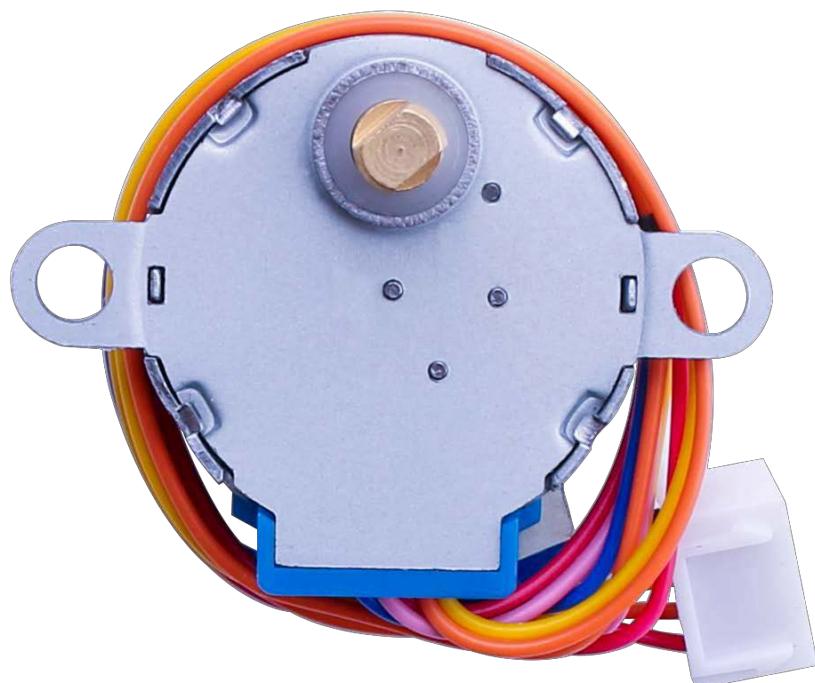
私たちが使用しているステッパーには、独自のドライバボードが付属しており、UNO に簡単に接続できます。

必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x 830 tie-points breadboard
- (1) x ULN2003 stepper motor driver module
- (1) x Stepper motor
- (1) x 9V1A Adapter
- (1) x Power supply module
- (6) x F-M wires (Female to Male DuPont wires)
- (1) x M-M wire (Male to Male jumper wire)

Stepper Motor

部品の紹介



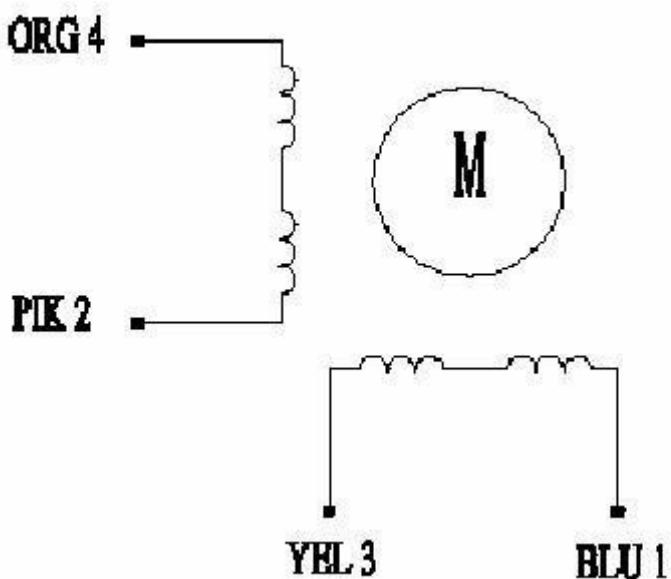
ステッパモータは、電気パルスを個別の機械的動作に変換する電気機械的装置である。ステッパモータのシャフトまたはスピンドルは、電気指令パルスが適切な順序で印加されると、離散的ステップ増分で回転する。モータの回転は、これらの印加された入力パルスに対していくつかの直接関係を持ちます。印加されるパルスのシーケンスは、モータシャフトの回転方向に直接関係します。モータシャフトの回転速度は、入力パルスの周波数に直接関係し、回転の長さは、印加される入力パルスの数に直接関係しています。ステッピングモータの最も重要な利点の1つは、開ループシステムで正確に制御できることです。開ループ制御は、位置に関するフィードバック情報が必要でないことを意味する。このタイプの制御は、光エンコーダなどの高価なセンシングおよびフィードバックデバイスの必要性を排除します。あなたの位置は、入力ステップのパルスを追跡するだけで知られています。

Stepper motor 28BYJ-48 Parameters

- Model: 28BYJ-48
- Rated voltage: 5VDC
- Number of Phase: 4
- Speed Variation Ratio: 1/64
- Stride Angle: $5.625^\circ / 64$
- Frequency: 100Hz
- DC resistance: $50\Omega \pm 7\%(25^\circ C)$
- Idle In-traction Frequency: > 600Hz
- Idle Out-traction Frequency: > 1000Hz
- In-traction Torque > 34.3mN.m(120Hz)
- Self-positioning Torque > 34.3mN.m
- Friction torque: 600-1200 gf.cm
- Pull in torque: 300 gf.cm
- Insulated resistance > $10M\Omega$ (500V)
- Insulated electricity power: 600VAC/1mA/1s
- Insulation grade: A
- Rise in Temperature < 40K(120Hz)
- Noise < 35dB(120Hz, No load, 10cm)

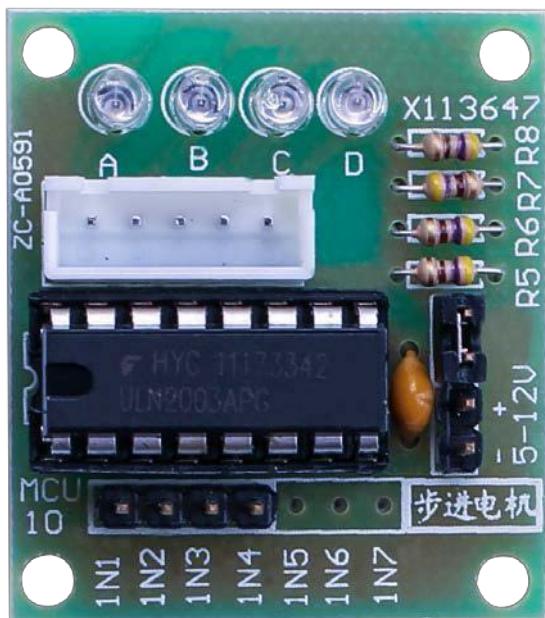
Interfacing circuits

WIRING DIAGRAM



バイポーラステッピングモーターは通常、そこから 4 本の電線が出ています。ユニポーラステッパーとは異なり、バイポーラステッパーは共通のセンター接続がありません。代わりに 2 つの独立したコイルセットがあります。ワイヤー間の抵抗を測定することで、ユニポーラステッパーと区別することができます。同じ抵抗を持つ 2 組のワイヤーが見つかるはずです。接続されていない（つまり、同じコイルに接続されていない）2 本のワイヤにメーターのリード線を接続している場合は、無限の抵抗（または無接続）が表示されます。

ULN2003 Driver Board



製品説明

- Size: 42mmx30mm
- Use ULN2003 driver chip, 500mA
- A. B. C. D LED4 相ステッパモータの動作状態を示します。
- ホワイトジャックは4相ステッピングモーター標準ジャックです。
- 電源ピンは分離されています。

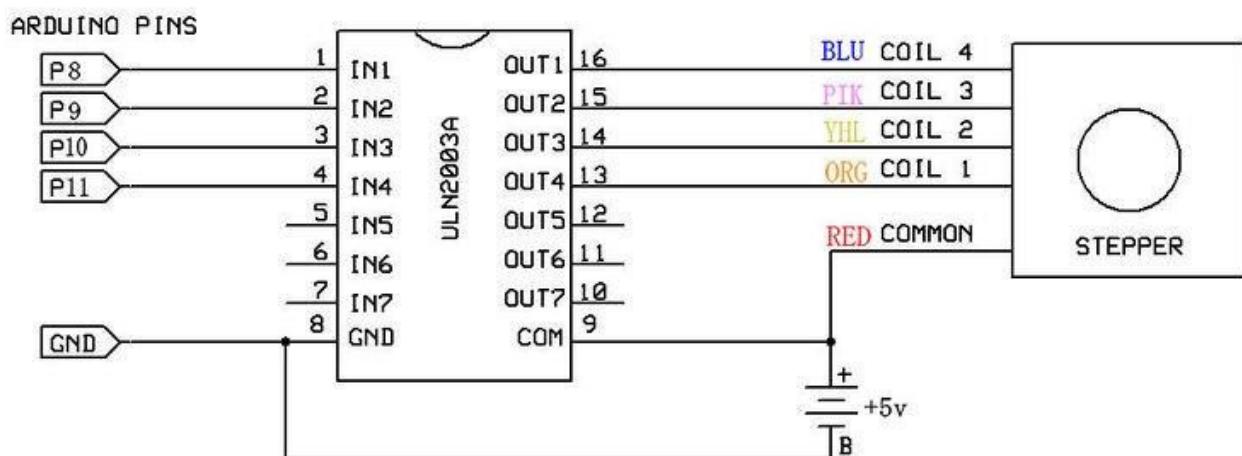
プロトタイプを作成するために、ULN2003 チップの残りのピンを保管しました。

ユニポーラステッパーを Arduino に接続する最も簡単な方法は、ULN2003A トランジスタアレイチップのブレークアウトを使用することです。ULN2003A には 7 つのダーリントントランジスタドライバが含まれており、7 つの TIP120 トランジスタを 1 つのパッケージにまとめているのと似ています。ULN2003A は、チャネルあたり最大 500mA を通過させることができ、オン時に内部電圧降下が約 1V になります。また、誘導負荷を駆動する際に電圧スパイクを散逸させるための内部クランプ・ダイオードも内蔵しています。ステッパを制御するには、特定の順序で各コイルに電圧を印加します。

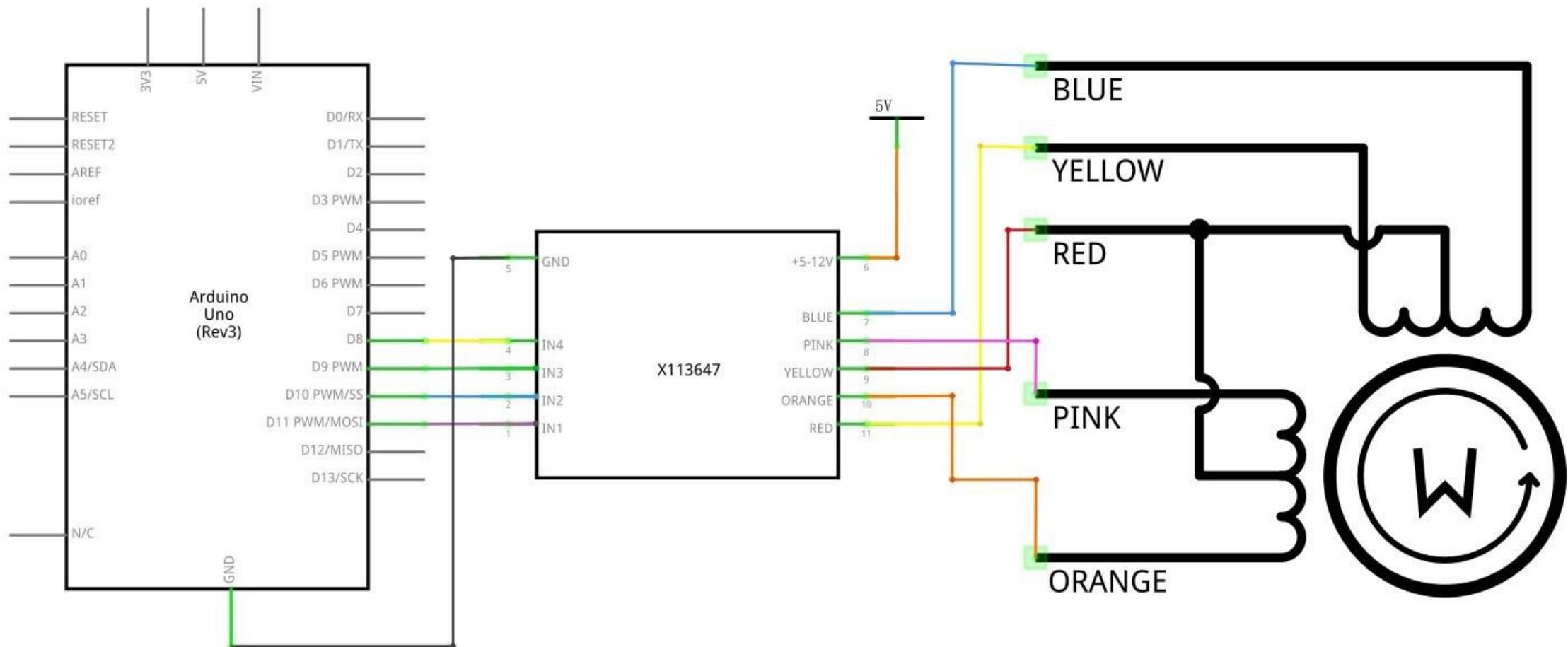
シーケンスは次のようにになります:

Lead Wire Color	---> CW Direction (1-2 Phase)							
	1	2	3	4	5	6	7	8
4 ORG	-	-						-
3 YEL		-	-	-				
2 PIK				-	-	-		
1 BLU						-	-	-

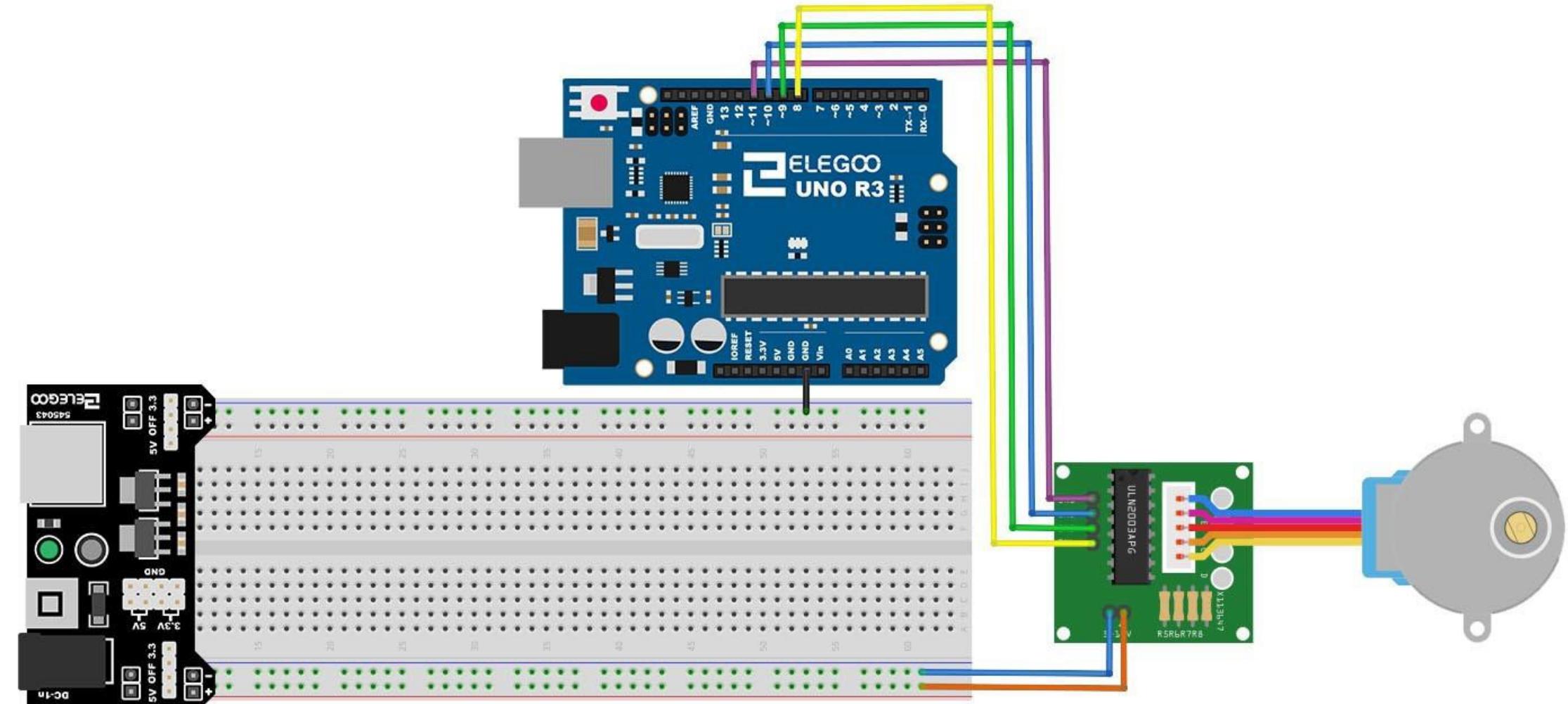
ULN2003A を使用してユニポーラ・ステッパー・モーターを 4 つのコントローラー・ピンにインターフェースする方法と 4 つの COM を使用してインターフェースする方法を示す概略図



Connection Schematic



Wiring diagram



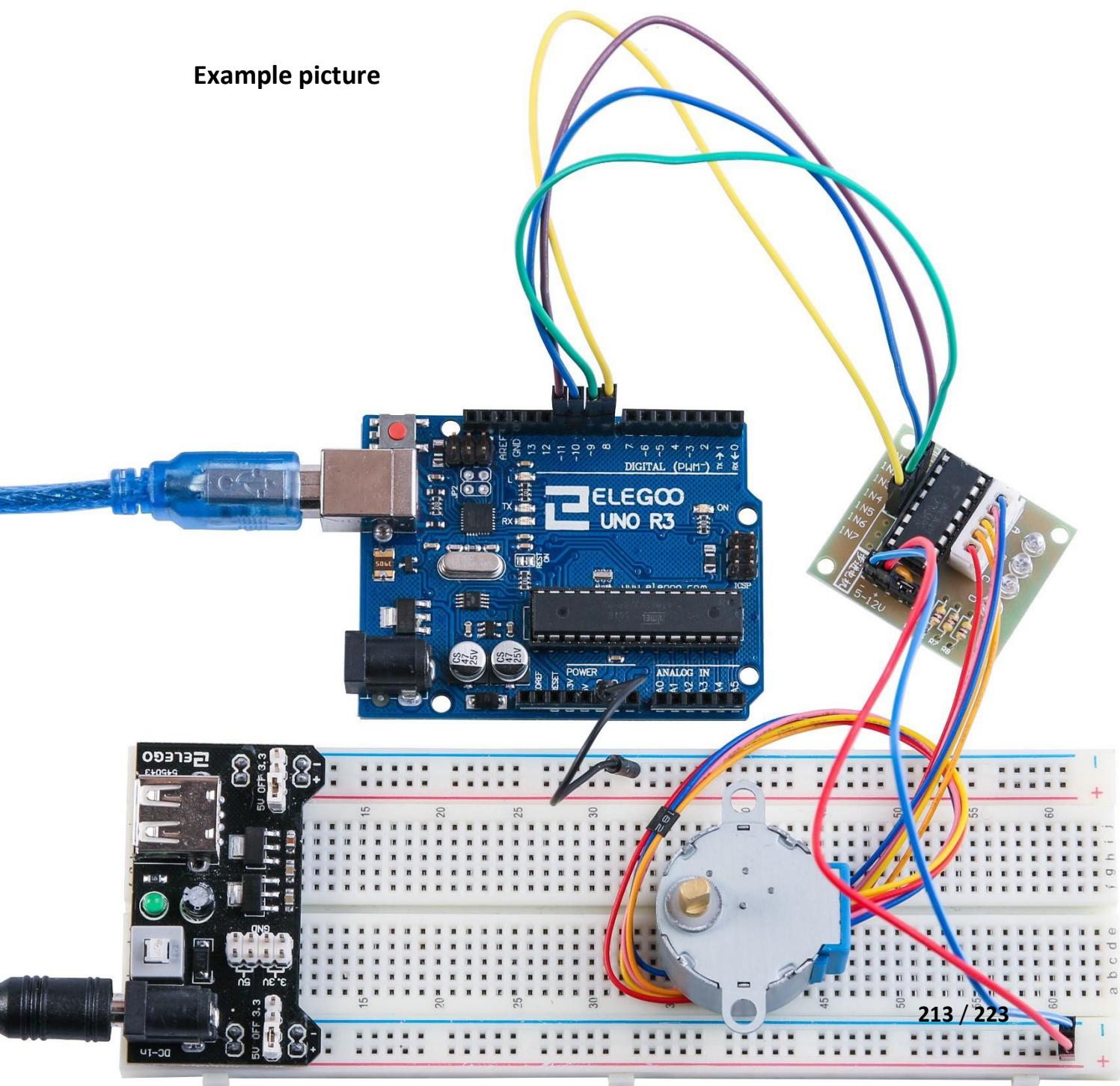
私たちはステッパーを制御するために4つのピンを使用しています。 ピン8-11はステッパーモーターを制御しています。 グランドをUNOからステッパーモーターに接続します。

Code

配線後、コードフォルダのレッスン31ステッパーでプログラムを開き、UPLOADをクリックしてプログラムをアップロードしてください。 エラーがある場合のプログラムアップロードの詳細については、レッスン2を参照してください。

これを実行する前に、<Stepper>ライブラリがインストールされていることを確認するか、必要に応じて再インストールしてください。 そうしないと、コードが機能しません。
ライブラリファイルのロードの詳細については、レッスン1を参照してください。

Example picture



Lesson 24 リモートでステッピングモータを制御する

概要

このレッスンでは、IR リモコンを使用してステッピングモーターを遠隔操作する、楽しく簡単な方法を学習します。

私たちが使用しているステッパーには、独自のドライバボードが付属しており、UNO に簡単に接続できます。

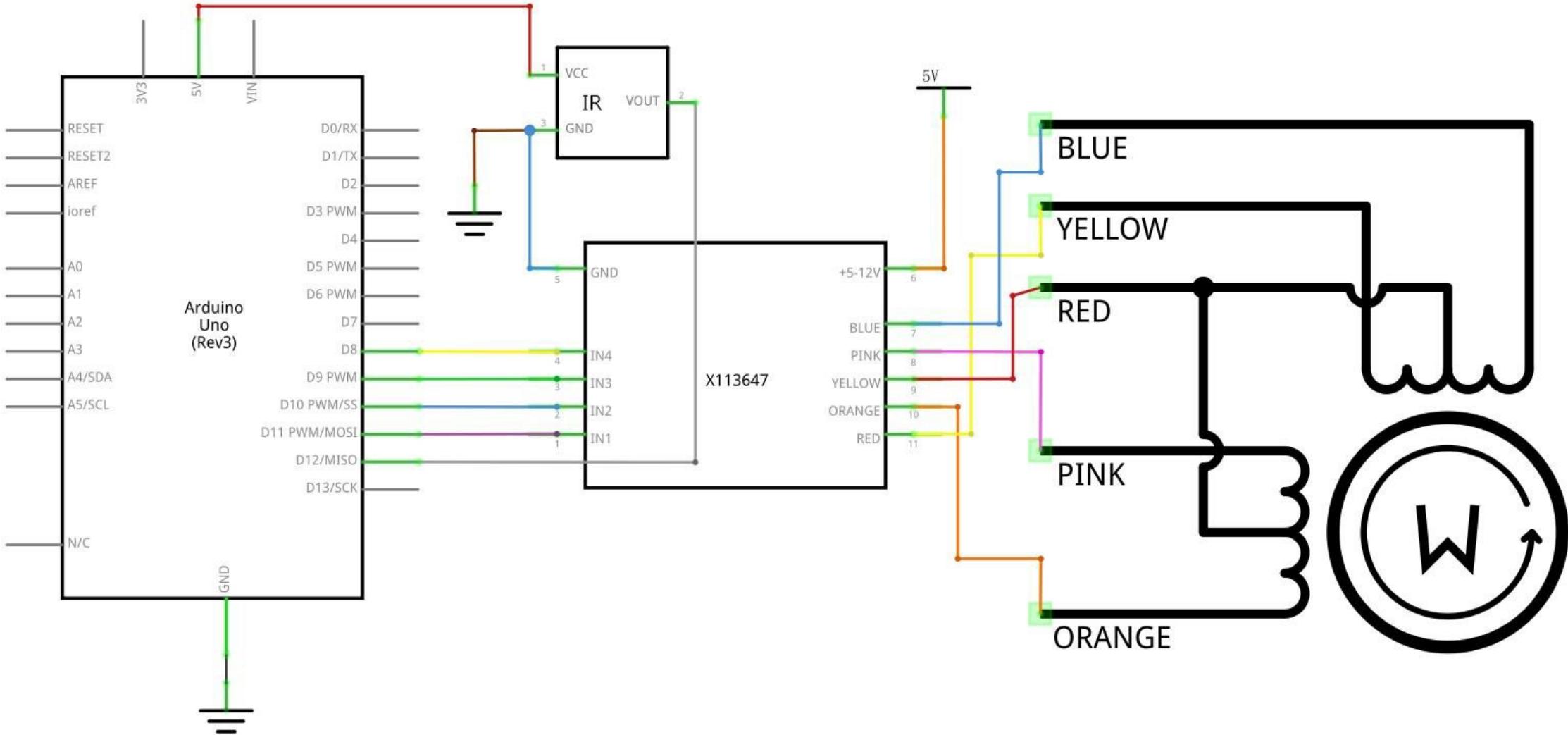
UNO から直接モータを駆動したくないので、ブレッドボードにプラグを差し込んで安価なブレッドボード電源を使用し、9V 1Amp 電源で電源を供給します。

IR センサは、ほとんど電力を消費しないため UNO に直接接続されています。

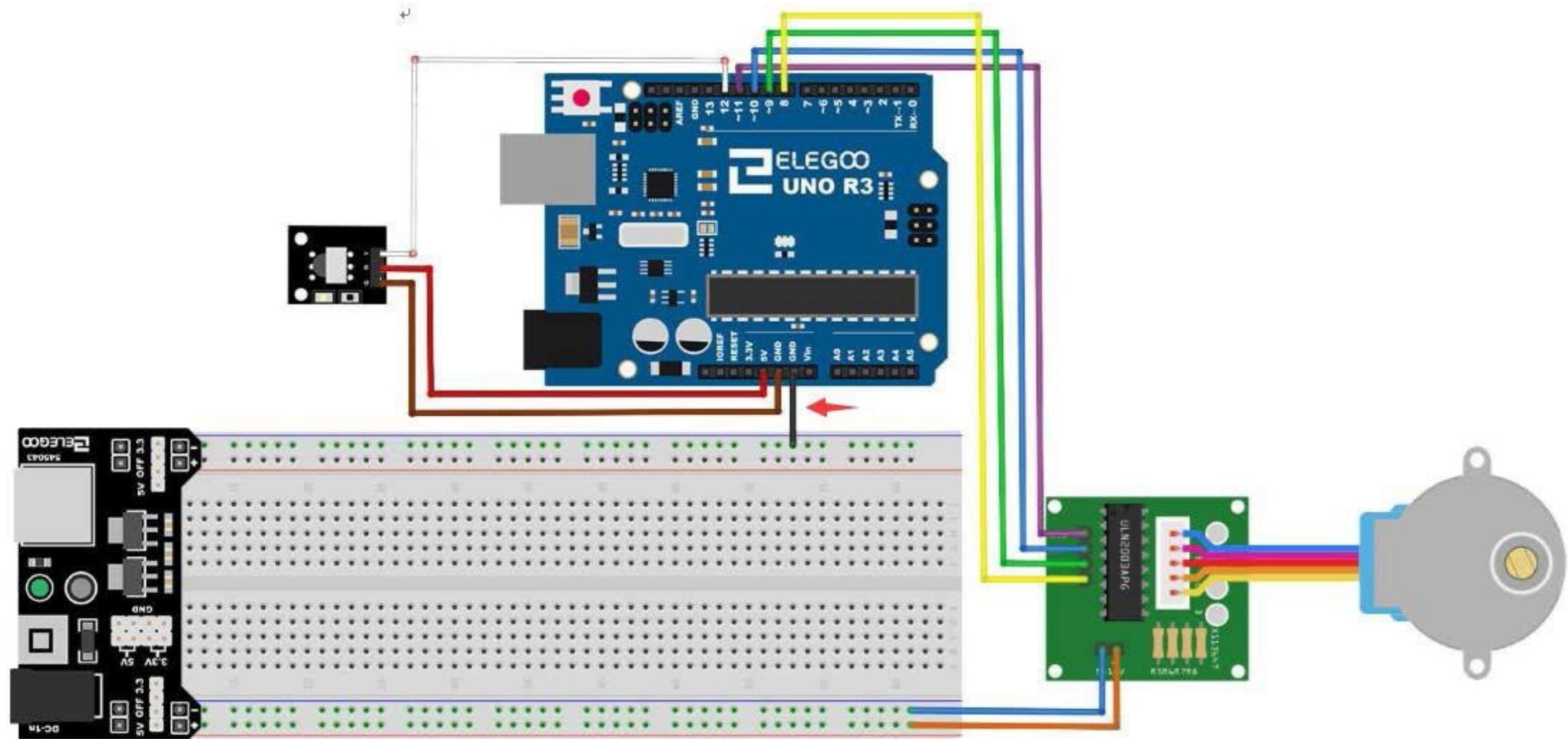
必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x 830 tie-points breadboard
- (1) x IR receiver module
- (1) x IR remote
- (1) x ULN2003 stepper motor driver module
- (1) x Stepper motor
- (1) x Power supply module
- (1) x 9V1A Adapter
- (9) x F-M wires (Female to Male DuPont wires)
- (1) x M-M wire (Male to Male jumper wire)

Connection Schematic



Wiring diagram



ステッパーを制御するために 4 本のピンを使用し、IR センサ用に 1 本のピンを使用しています。 ピン 8-11 はステッパモータを制御し、ピン 12 は IR 情報を受信している。

UNO からセンサーに 5V とグランドを接続します。 予防措置として、プレッドボード電源を使用してステッパーモーターに電力を供給します。これは、より多くの電力を使用できるため、UNO の電源を損傷したくないからです。

Code

配線後、コードフォルダのプログラムを開きます - レッスン 32 ステッピングモータをリモートで制御し、プログラムをアップロードするにはアップロードをクリックしてください。 エラーがある場合のプログラムアップロードの詳細については、レッスン 2 を参照してください。

これを実行する前に、[`<IRremote>`](#)

[`<Stepper>`](#)ライブラリをインストールするか、必要に応じて再インストールしてください。 そうしないと、コードが機能しません。 ライブラリファイルのロードの詳細については、レッスン 1 を参照してください。

このコードは、IR リモコンの 2 つの値 (VOL + と VOL-) のみを認識します。 リモートで VOL + を押すと、モーターは時計回りに完全に回転します。 VOL-は反時計回りに完全に回転します。

