



**THE MOST COMPLETE STARTER KIT
TUTORIAL FOR UNO**

V1.0.17.7.10

序文

当社

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もので、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高品質のサービスをご提供しており、service@elegoo.com または EUserice@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

**Servo Motor
(SG90)
1PC**



**ULN2003 Stepper
Motor Driver
Module
1PC**



**Power Supply
Module
1PC**



**GY-521
Module
1PC**



**Elegoo UNO R3
Controller
Board
1PC**



**LCD 1602 Module
(with pin header)
1PC**



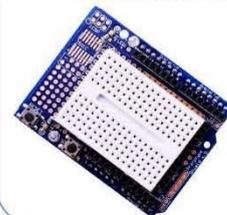
**Rc522
RFID Module
1PC**



**Stepper Motor
1PC**

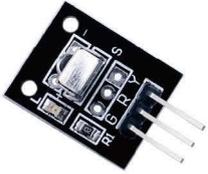


**Prototype Expansion
Module
1PC**



Contact us : service@elegoo.com

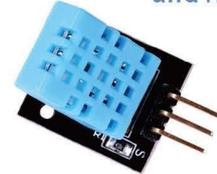
IR Receiver Module
1PC



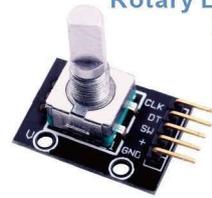
Joystick Module
1PC



DHT11 Temperature and Humidity Module
1PC



Rotary Encoder Module
1PC



Ultrasonic Sensor
1PC



DS1307 RTC Module
1PC



HC-SR501 PIR Motion Sensor Module
1PC



Sound Sensor Module
1PC

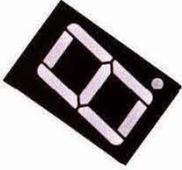


Water Level Detection Sensor Module
1PC



Contact us : service@elegoo.com

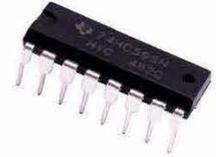
1 Digit 7-Segment Display
1PC



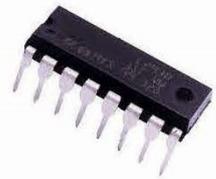
4 Digit 7-Segment Display
1PC



74HC595 IC
1PC



L293D
1PC



Potentiometer
(10K)
2PCS



Remote
1PC



MAX7219
Module
1PC

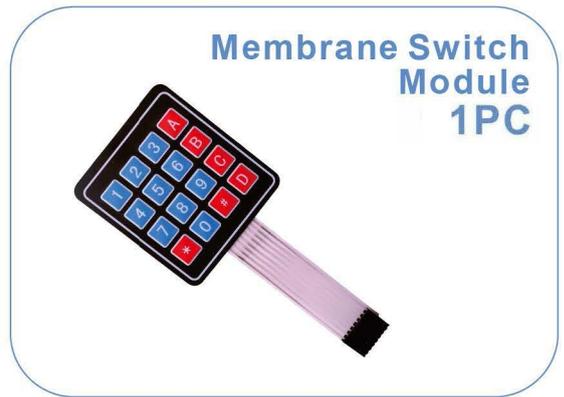


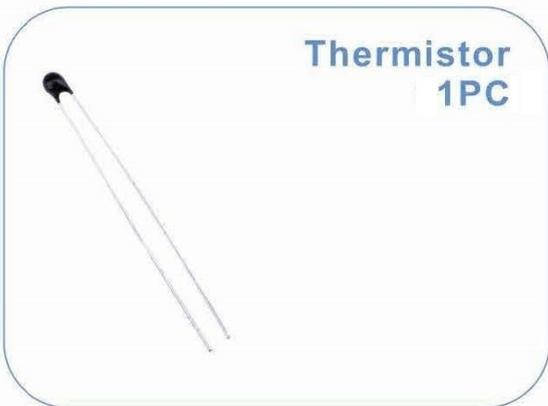
Active Buzzer
1PC

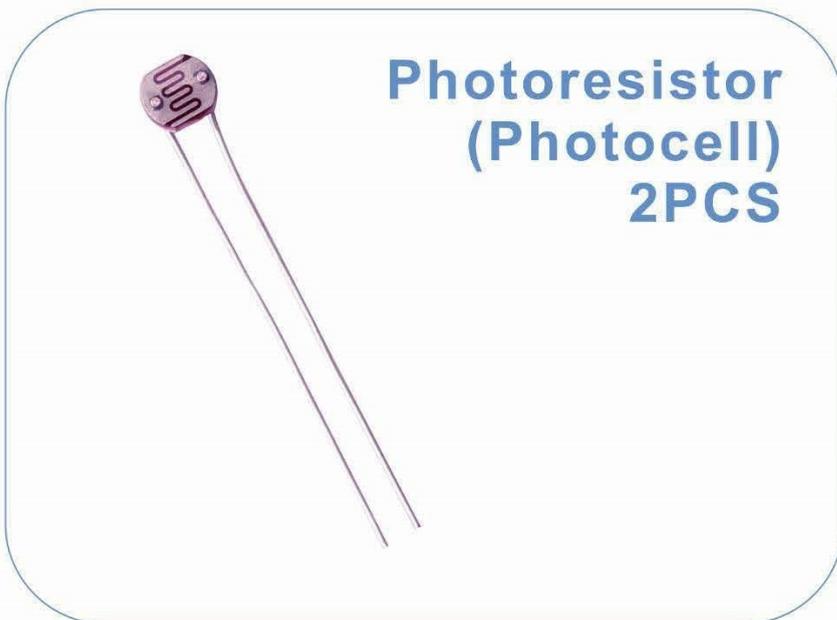


Passive Buzzer
1PC









Content

Lesson 0 IDE のインストール	12
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 アクティブブザー.....	64
Lesson 7 受動ブザー	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 膜スイッチモジュール.....	85
Lesson 12 DHT11 温度湿度センサー	91
Lesson 13 アナログジョイスティックモジュール	97
Lesson 14 赤外線受信モジュール.....	102
Lesson 15 MAX7219 LED ドット行列モジュール	108
Lesson 16 GY-521 モジュール.....	112
Lesson 17 HC-SR501 PIR 検知器	121
Lesson 18 水位検出検知器.....	131
Lesson 19 実時間モジュール	136
Lesson 20 音セ検知器.....	141
Lesson 21 RC522 RFID モジュール.....	147

Lesson 22 LCD ディスプレイ	152
Lesson 23 温度計	157
Lesson 24 8 個の LED と 74HC595.....	162
Lesson 25 シリアルモニタ.....	169
Lesson 26 光電池	175
Lesson 27 74HC595 と 7セグメント表示	180
Lesson 28 4 つのデジタル 7 セグメント表示	186
Lesson 29 直流モータ	191
Lesson 30 リレー	201
Lesson 31 ステッパーモーター	206
Lesson 32 リモートでステッピングモータを制御する	214
Lesson 33 ロータリーエンコーダ付きステッピングモータの制御.....	218

Lesson 0 IDE のインストール

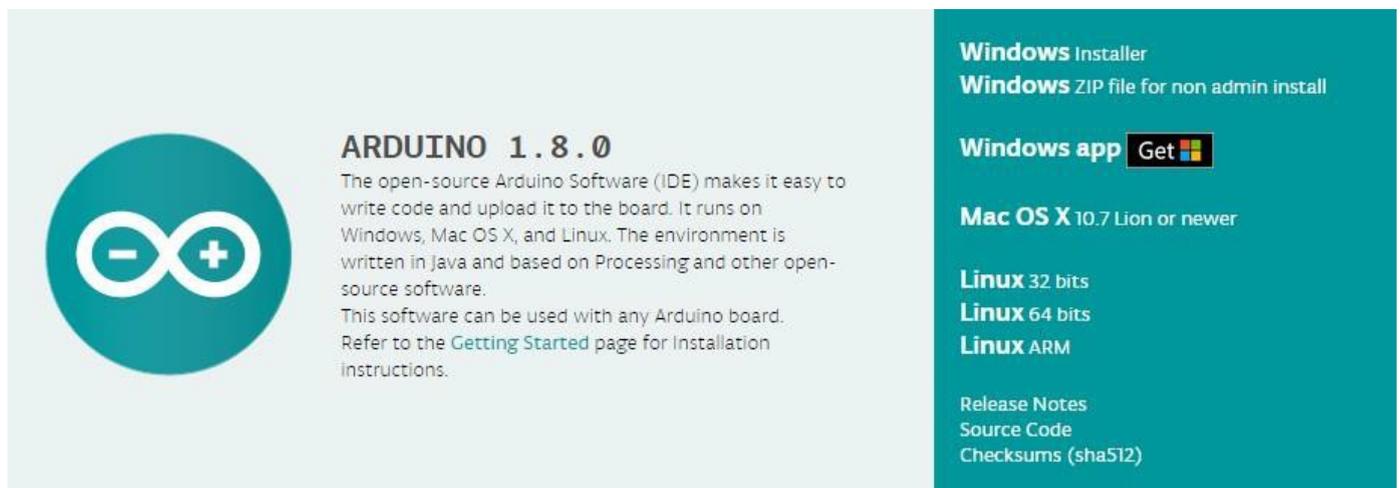
前書き

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.



ARDUINO 1.8.0

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer
Windows ZIP file for non admin install

Windows app [Get](#)

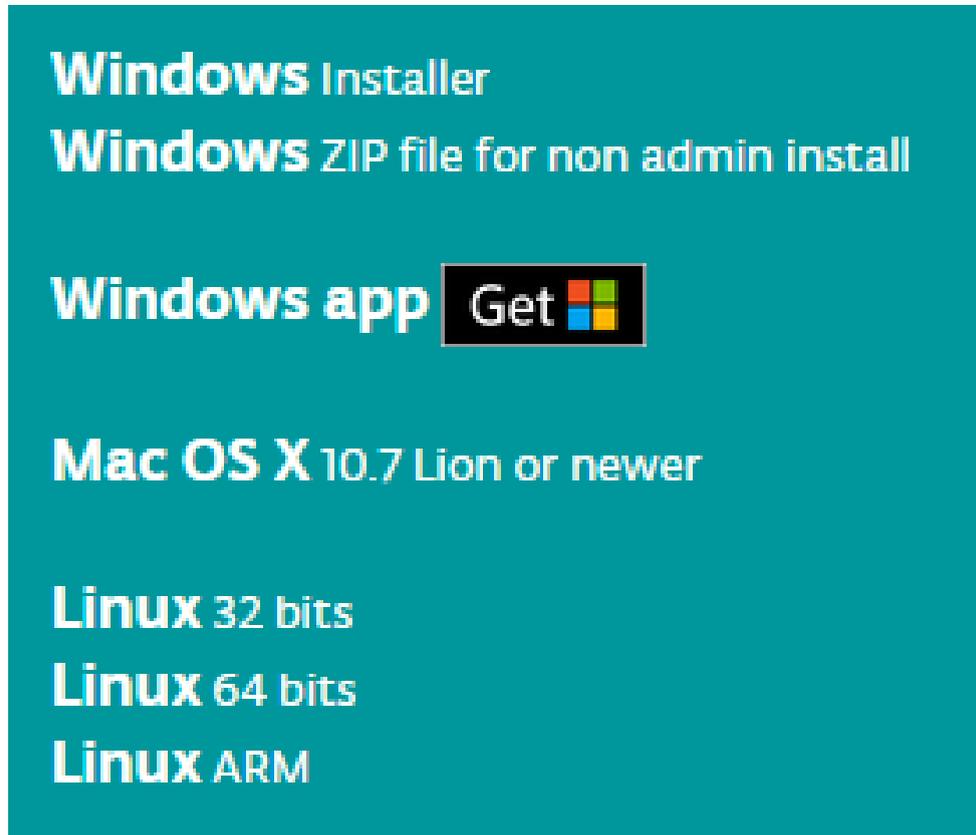
Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

このウェブサイトを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。



Windows Installer
Windows ZIP file for non admin install
Windows app [Get](#)
Mac OS X 10.7 Lion or newer
Linux 32 bits
Linux 64 bits
Linux ARM

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.



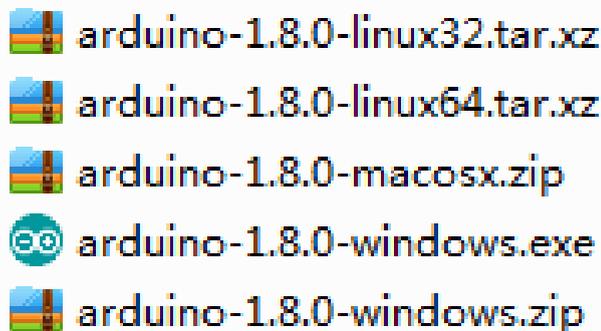
SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **8,808,272** TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

\$3 \$5 \$10 \$25 \$50 OTHER

JUST DOWNLOAD [CONTRIBUTE & DOWNLOAD](#)

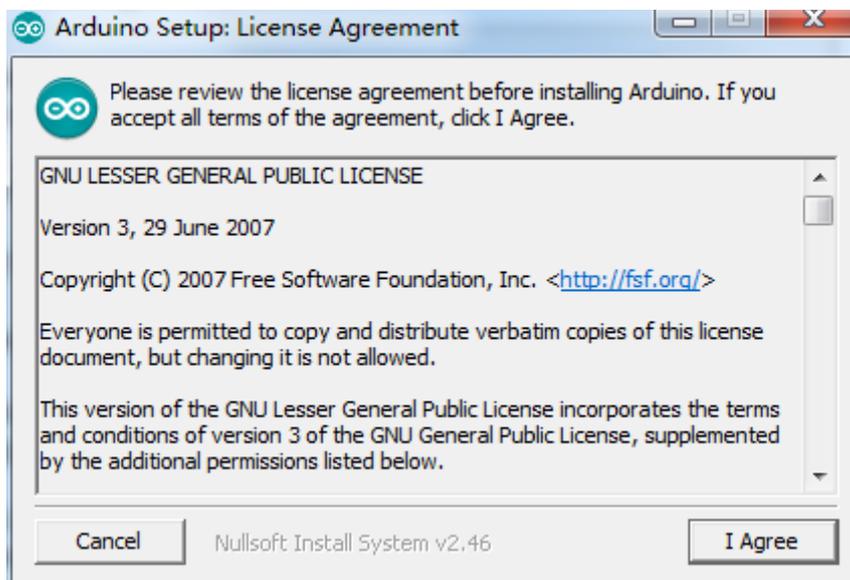
Click *JUSTDOWNLOAD*.

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。

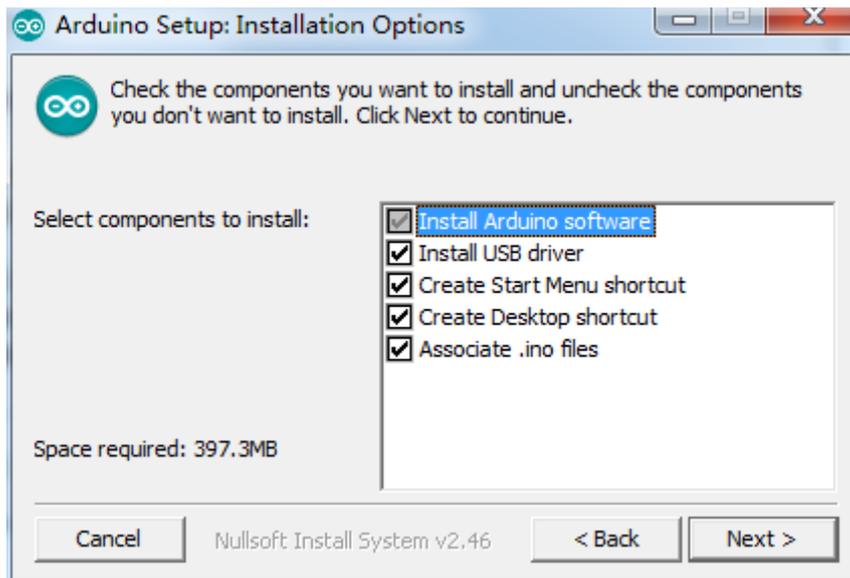


Installing Arduino (Windows)

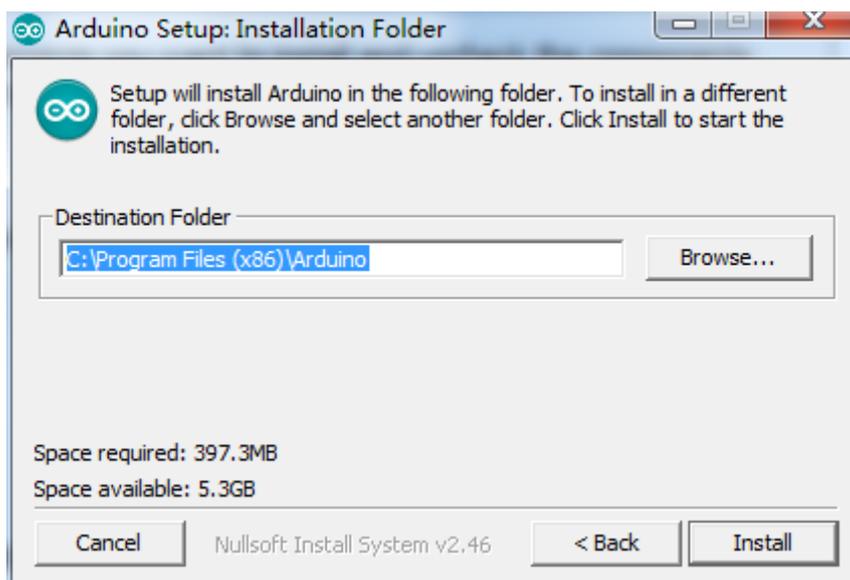
exe で Arduino をインストールします。インストールパッケージ。



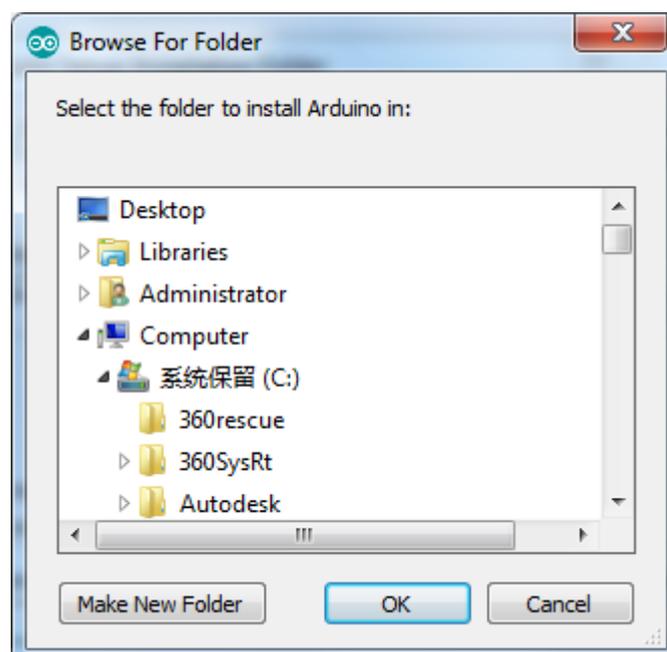
「同意する」をクリックすると、次のインターフェースが表示されます。



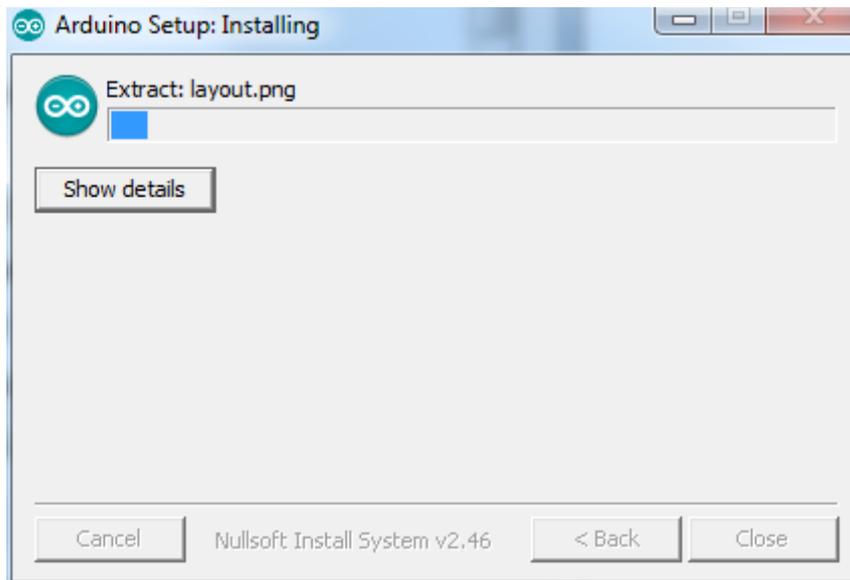
次へをクリックします



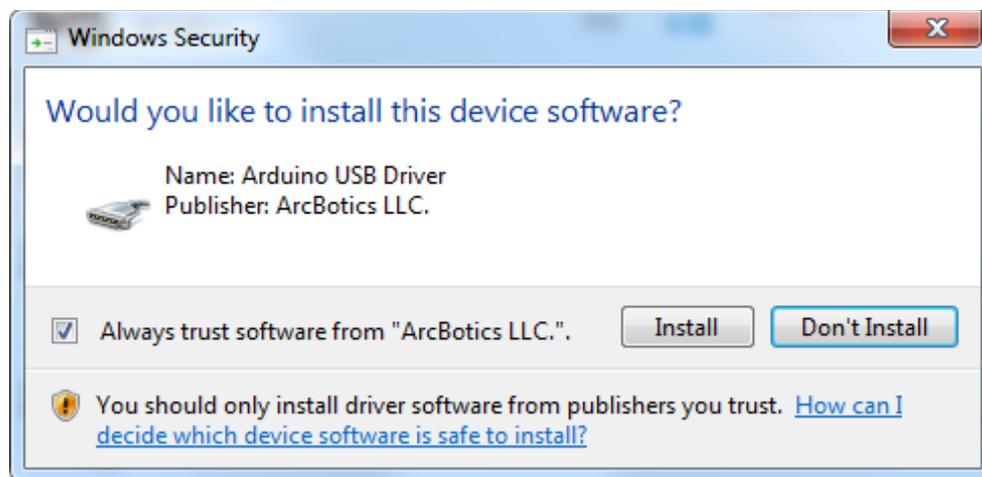
[参照]を押してインストールパスを選択するか、目的のディレクトリに直接入力することができます。



[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



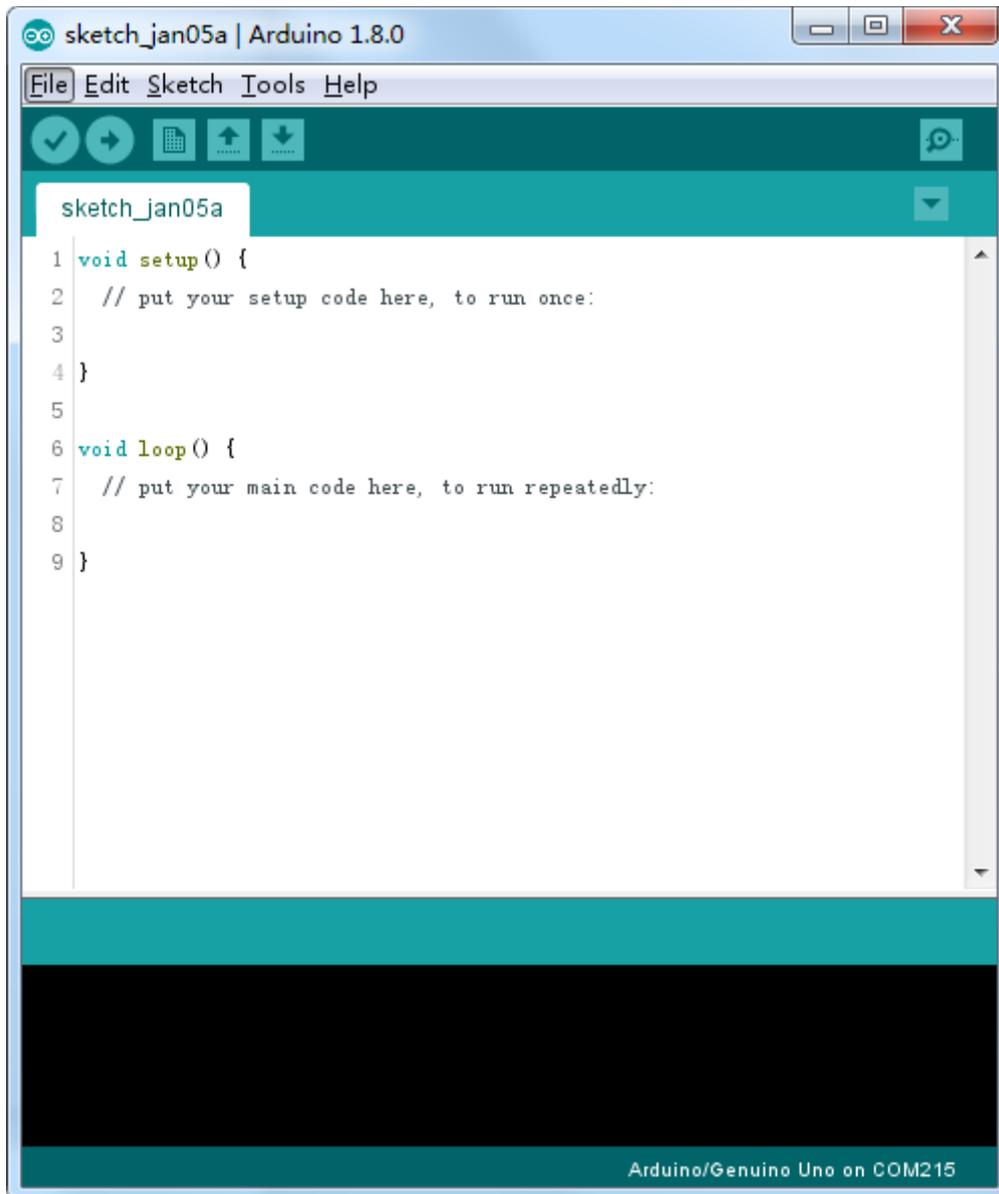
最後に、次のインターフェイスが表示されたら、[インストール]をクリックしてインストールを完了します。



次に、デスクトップに次のアイコンが表示されます。



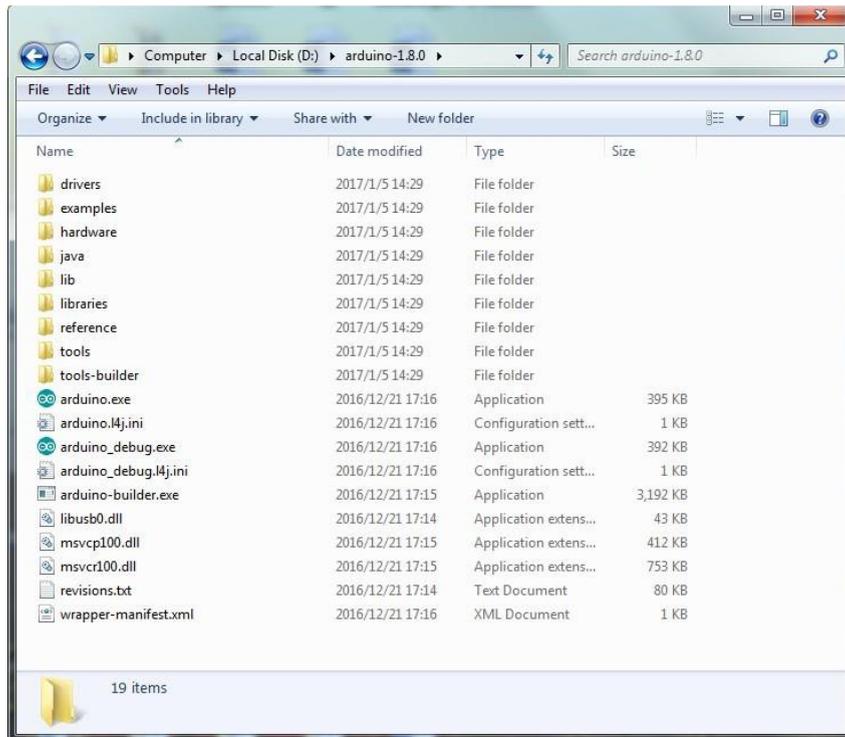
ダブルクリックして希望の開発環境に入ります



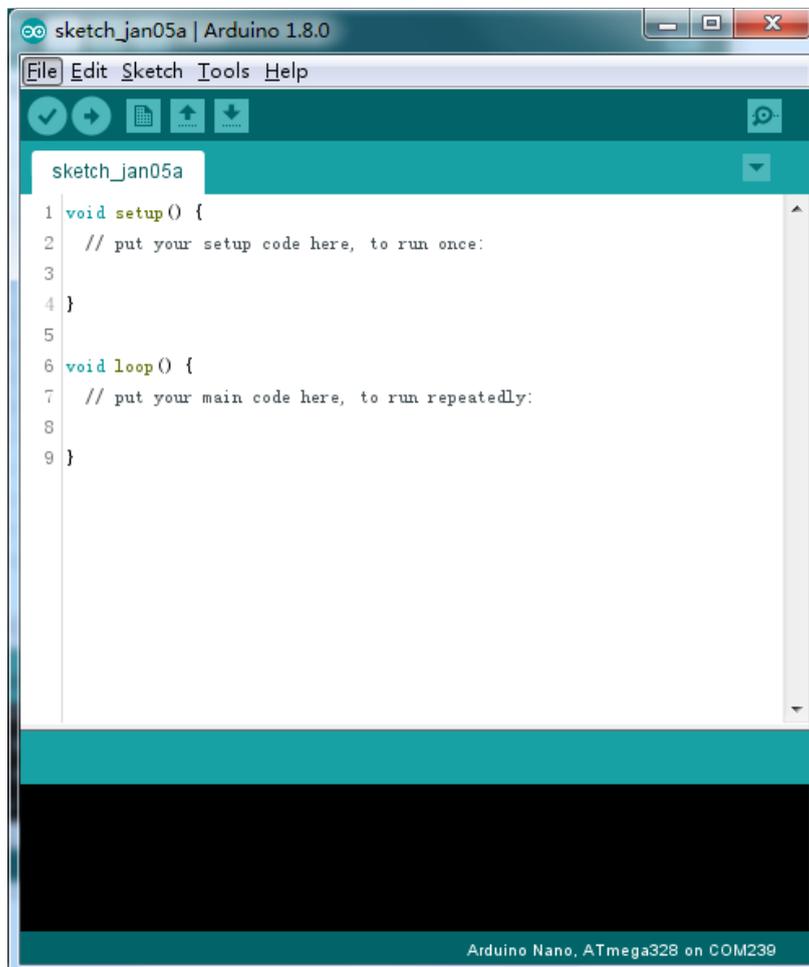
インストール用のインストールパッケージを直接選択して下の内容をスキップし、次のセクションにジャンプすることができます。しかし、インストールパッケージ以外のいくつかの方法を学びたい場合は、引き続きこのセクションをお読みください。

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。

 [arduino-1.8.0-windows.zip](#)



arduino.exe

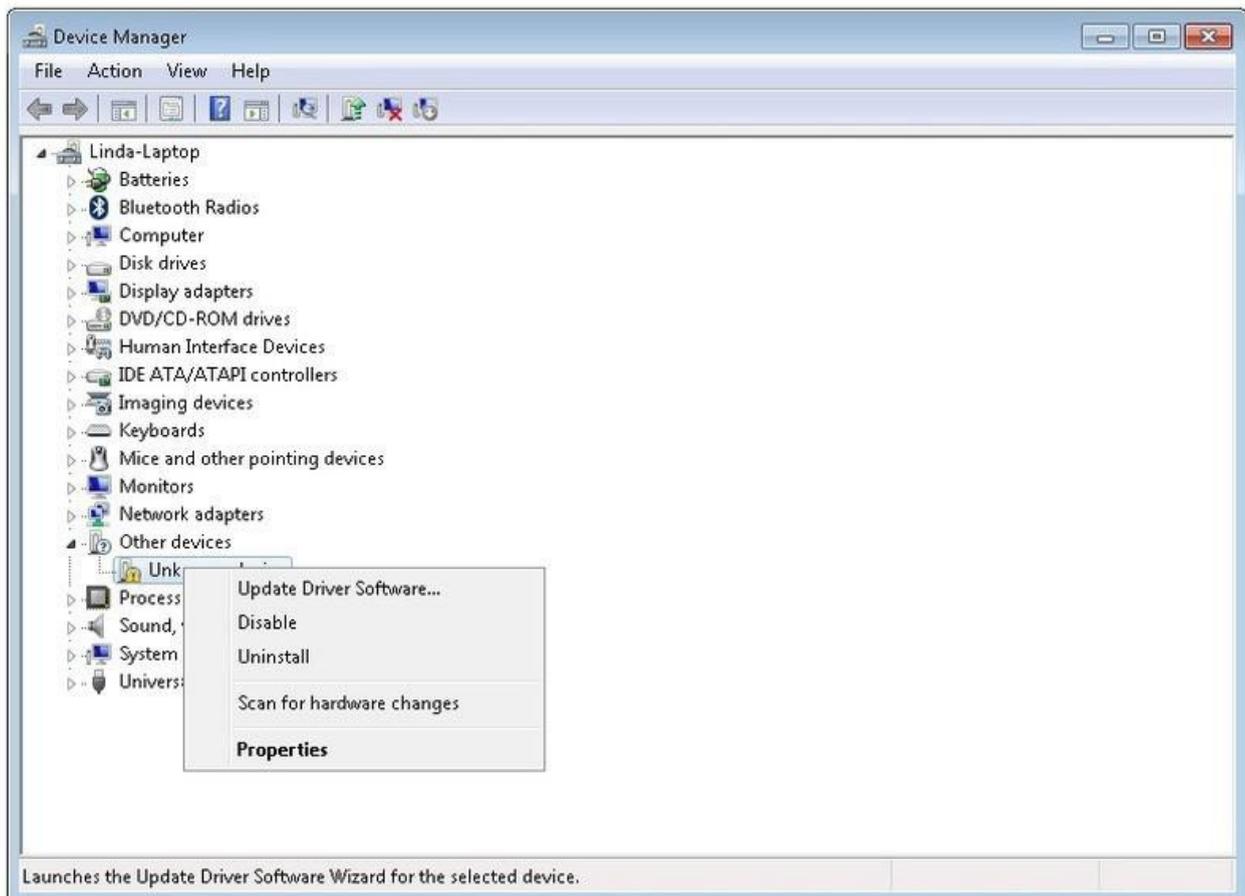


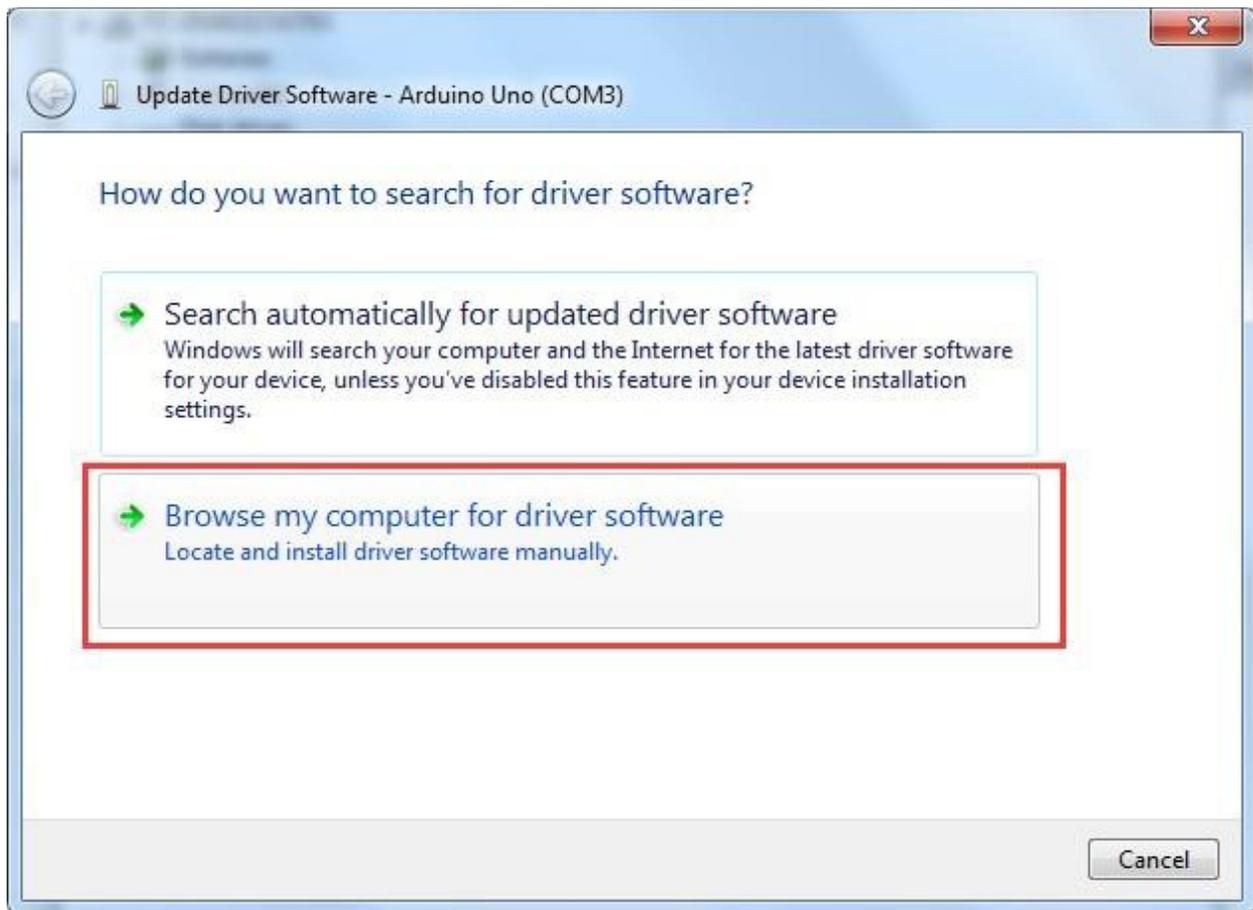
ただし、このインストール方法ではドライバを別途インストールする必要があります。

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あります。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 OSX)

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。

 [arduino-1.8.0-macosx.zip](#)

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 の 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。

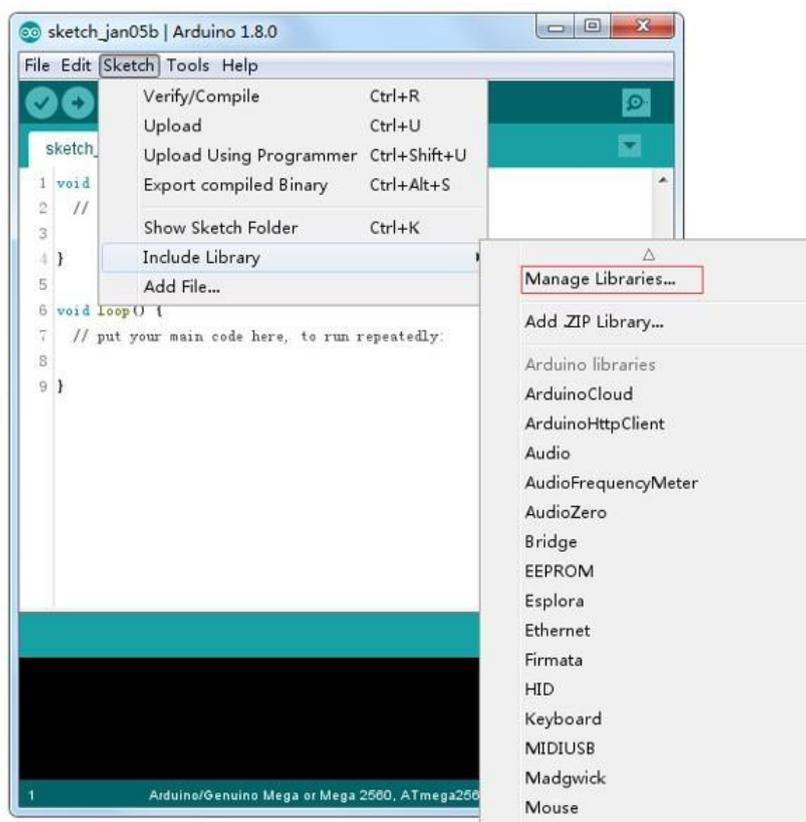
What are Libraries?

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

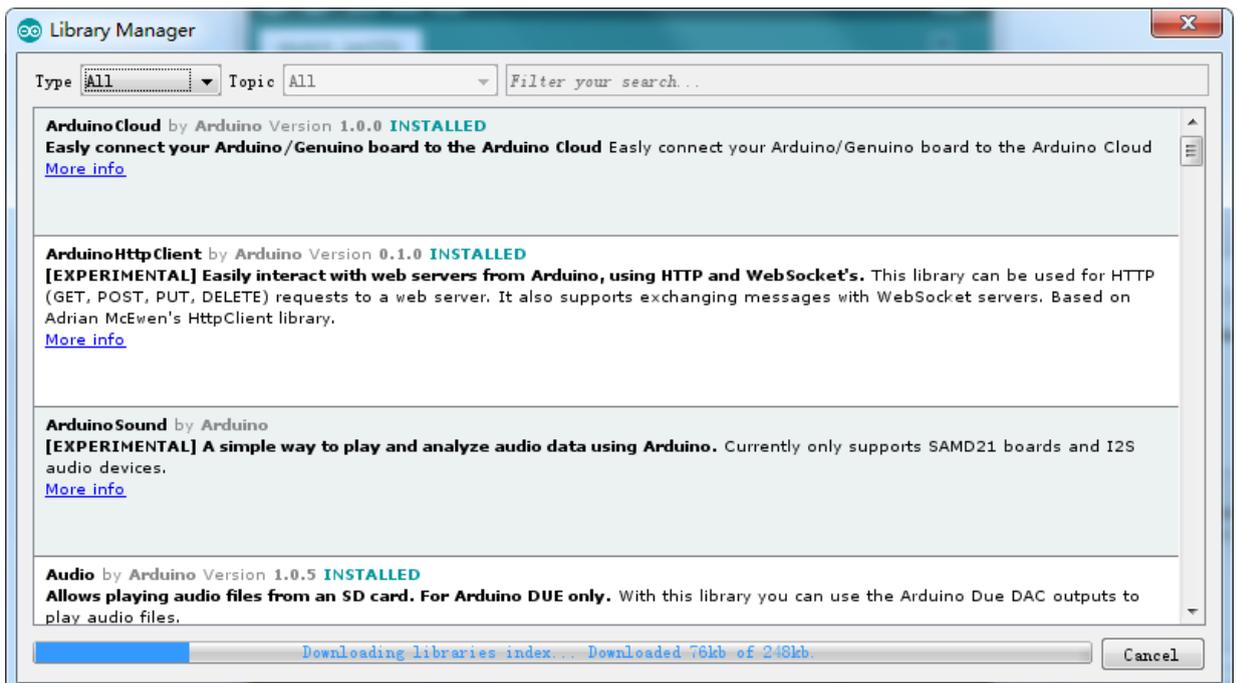
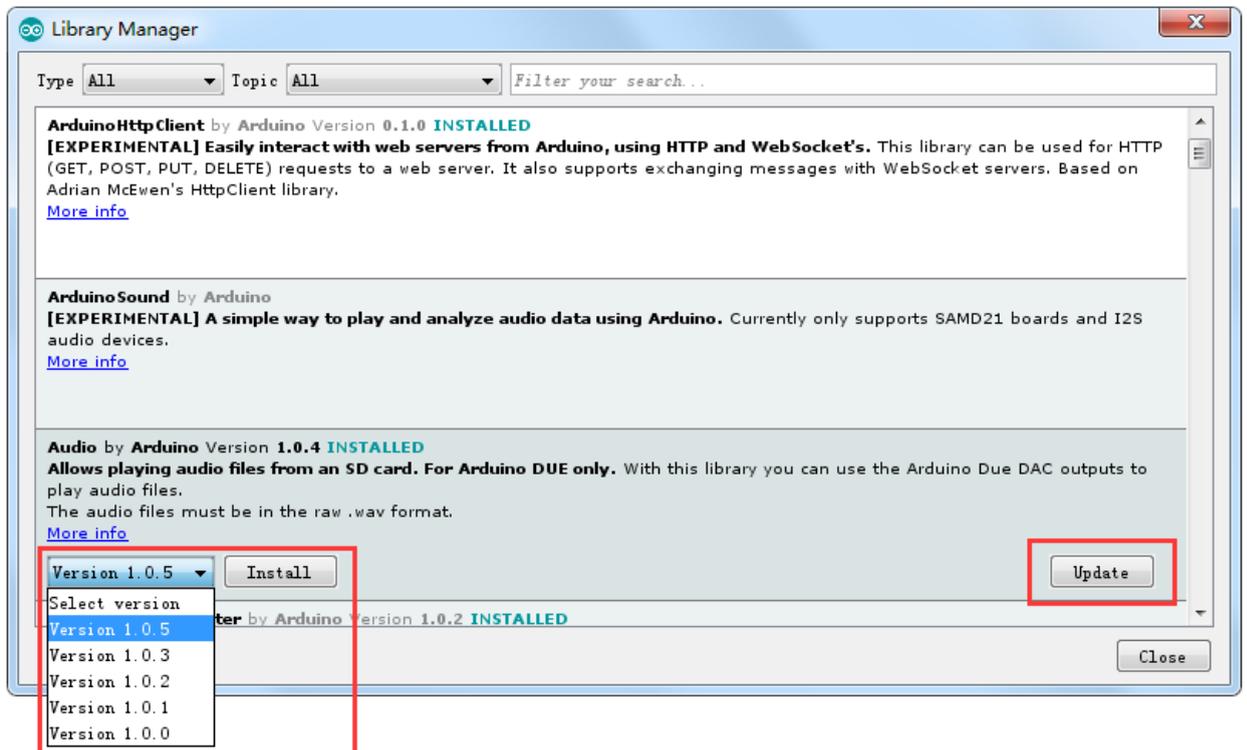
ライブラリマネージャの使用

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。

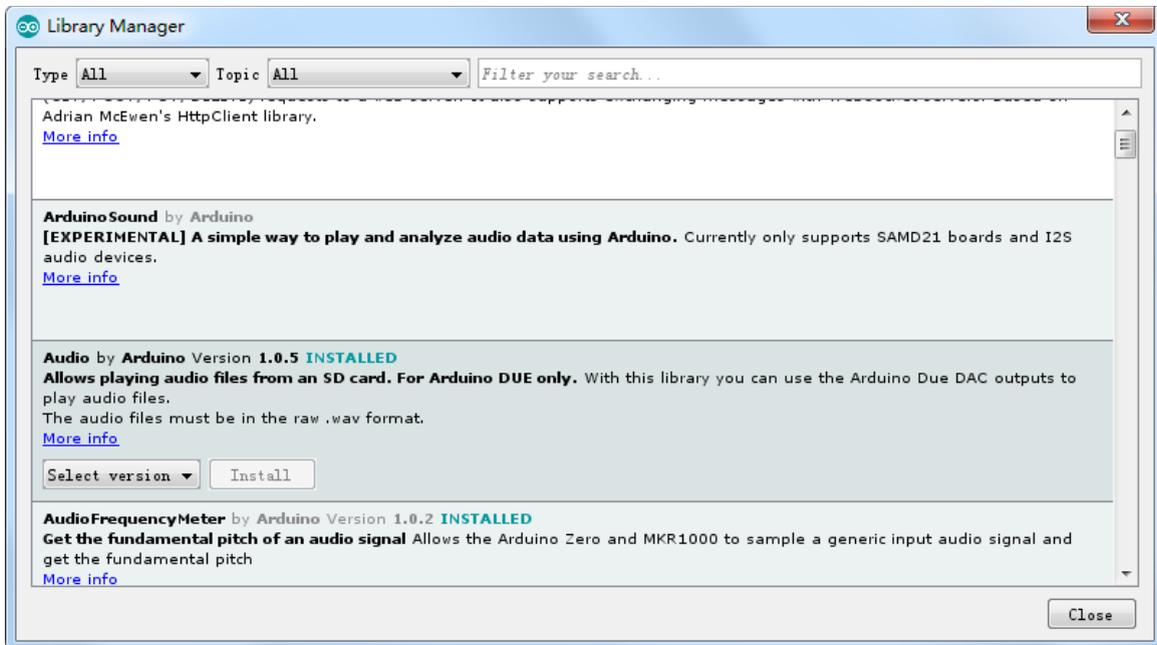


次に、ライブラリマネージャが開き、既にインストールされているか、インストールの準備が整っているライブラリのリストが表示されます。この例では、Bridge ライブラリをインストールします。リストをスクロールして検索し、インストールするライブラリのバージョンを選択します。ライブラリの 1 つのバージョンしか利用できないことがあります。バージョン選択メニューが表示されない場合は、これは正常です。

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。

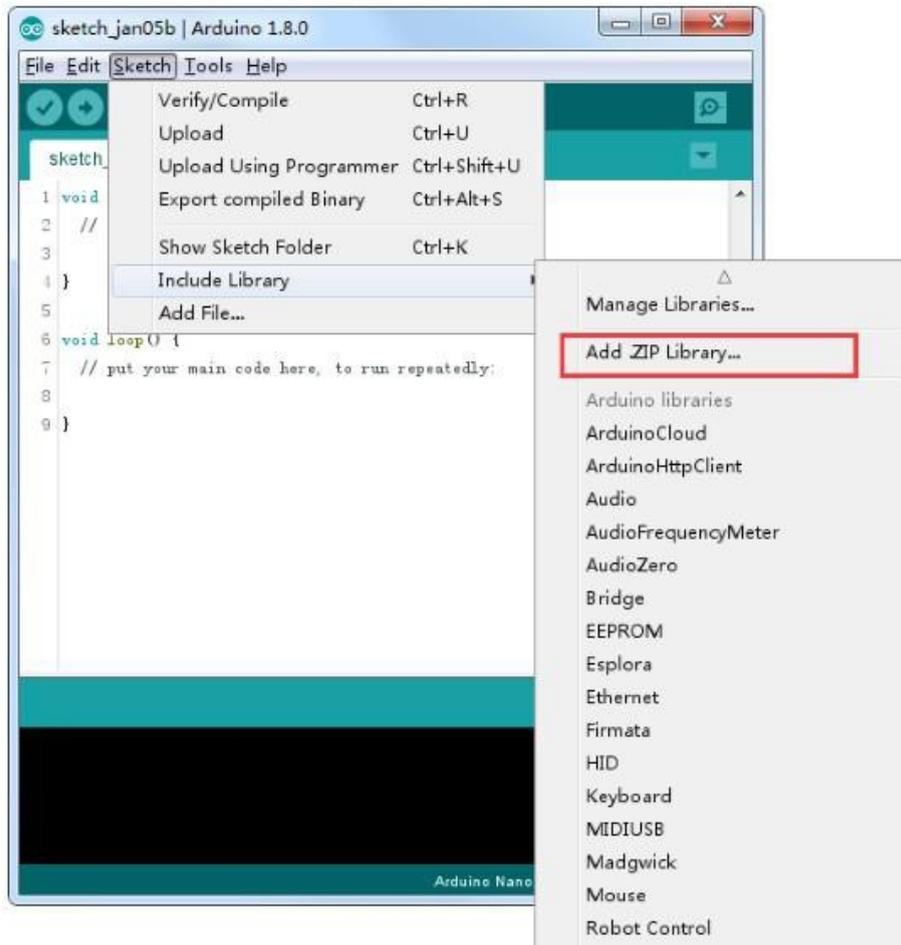


新しいライブラリは[ライブラリを含む]メニューで使用できるようになりました。独自のライブラリを追加したい場合は、[Github](#) で新しい問題を開きます。

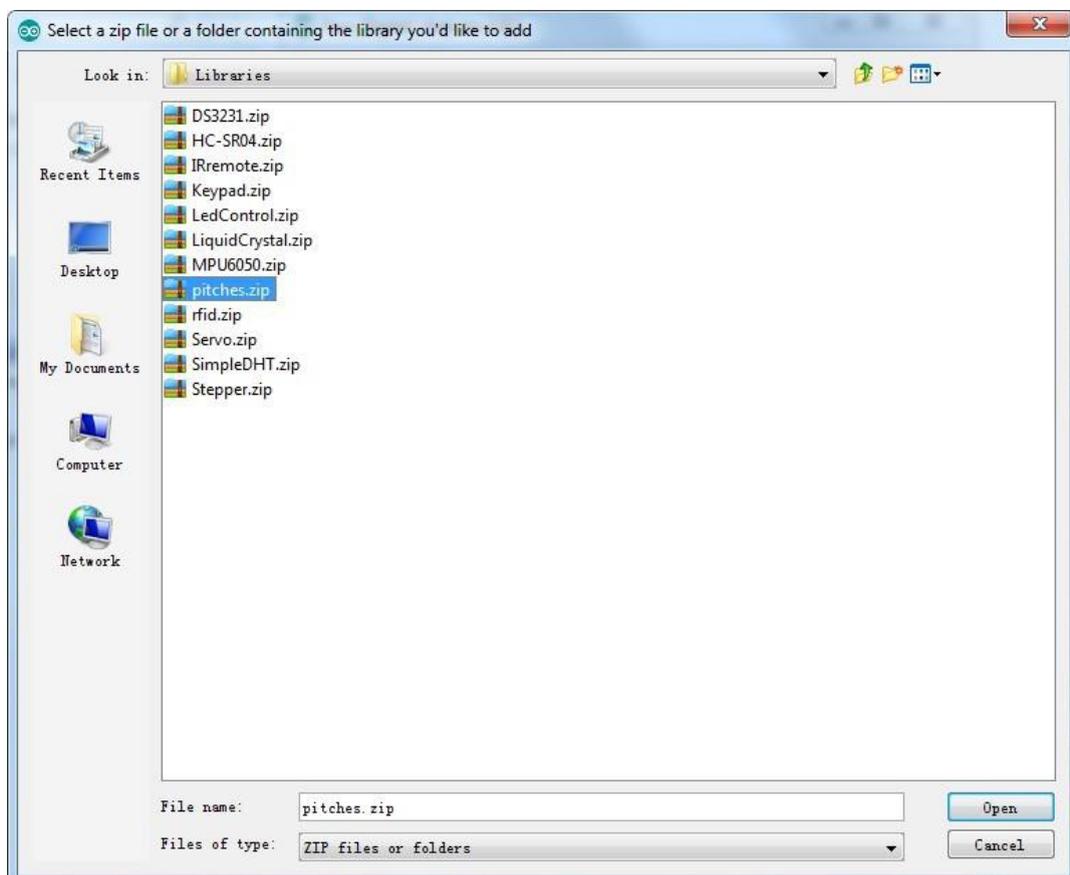
.zip ライブラリのインポート

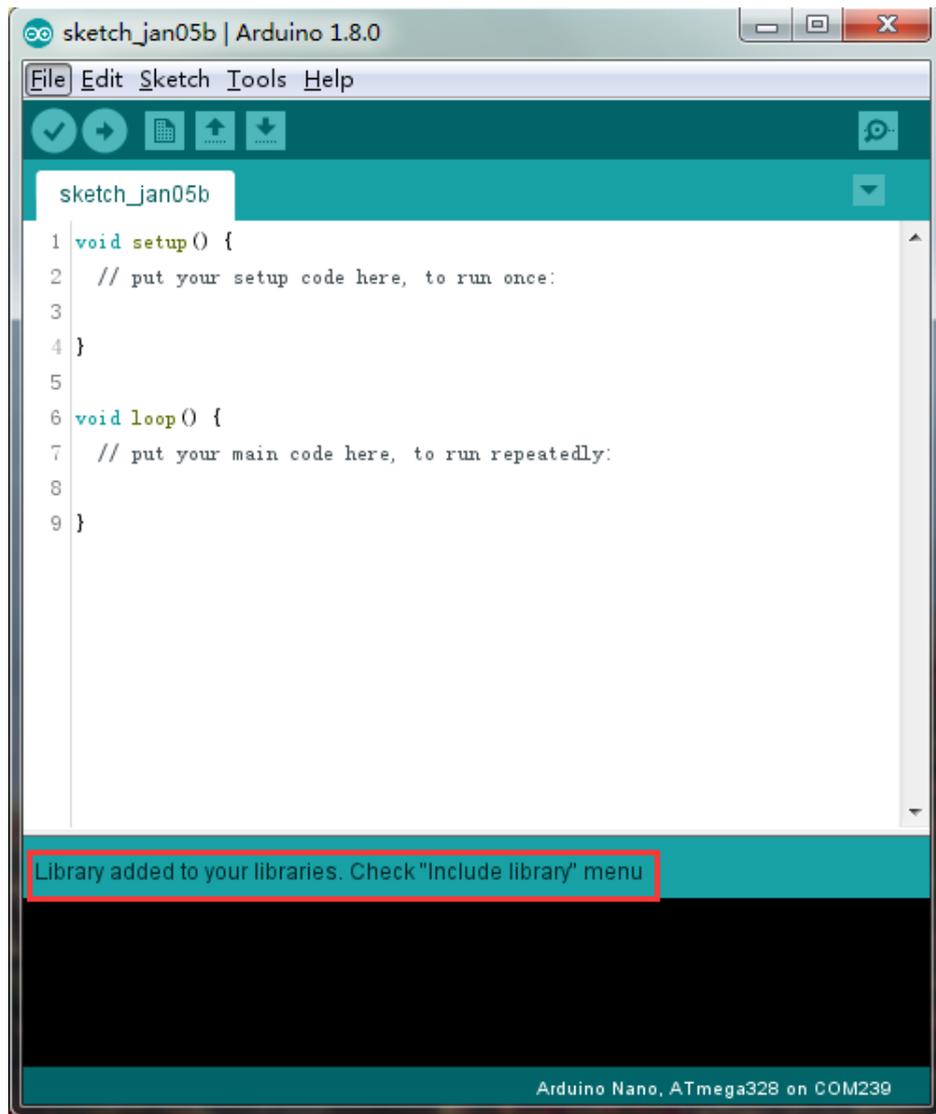
ライブラリは、多くの場合、ZIP ファイルまたはフォルダとして配布されます。フォルダの名前はライブラリの名前です。フォルダ内には、.cpp ファイル、.h ファイル、および keywords.txt ファイル、examples フォルダ、およびライブラリに必要なその他のファイルが含まれます。バージョン 1.0.5 から、サードパーティのライブラリを IDE にインストールすることができます。ダウンロードしたライブラリを解凍しないでそのままにしてください。

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が、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がない場合は、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-

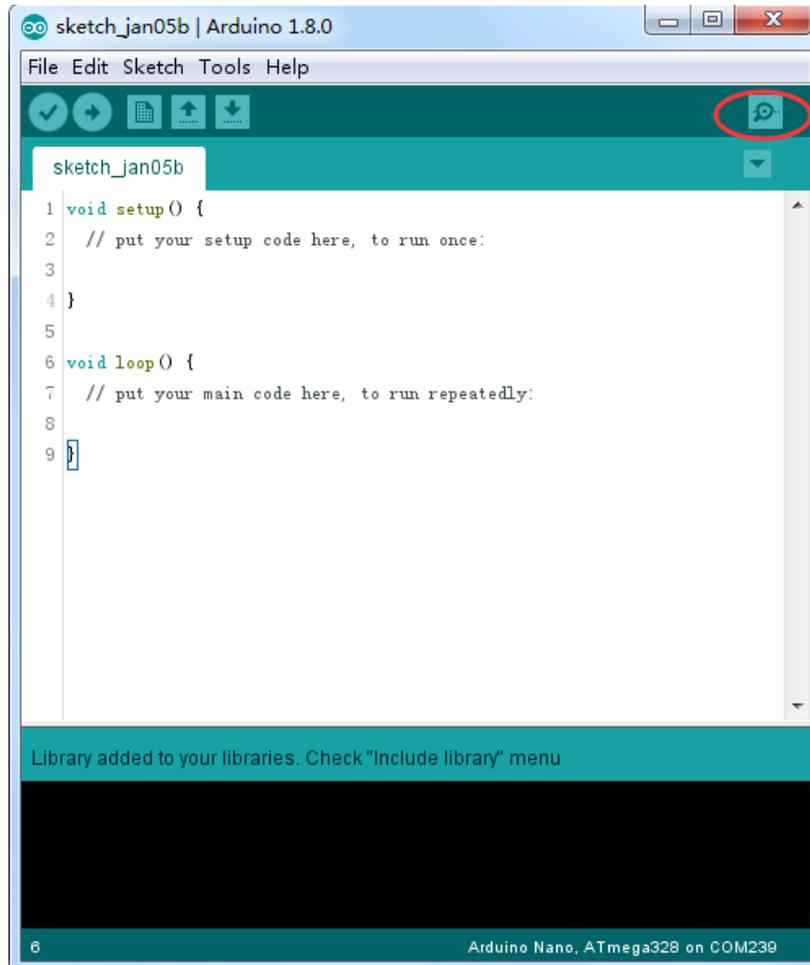
> [ライブラリのインポート]メニュー項目を選択します。それでおしまい! ライブラリをインストールしました!

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。

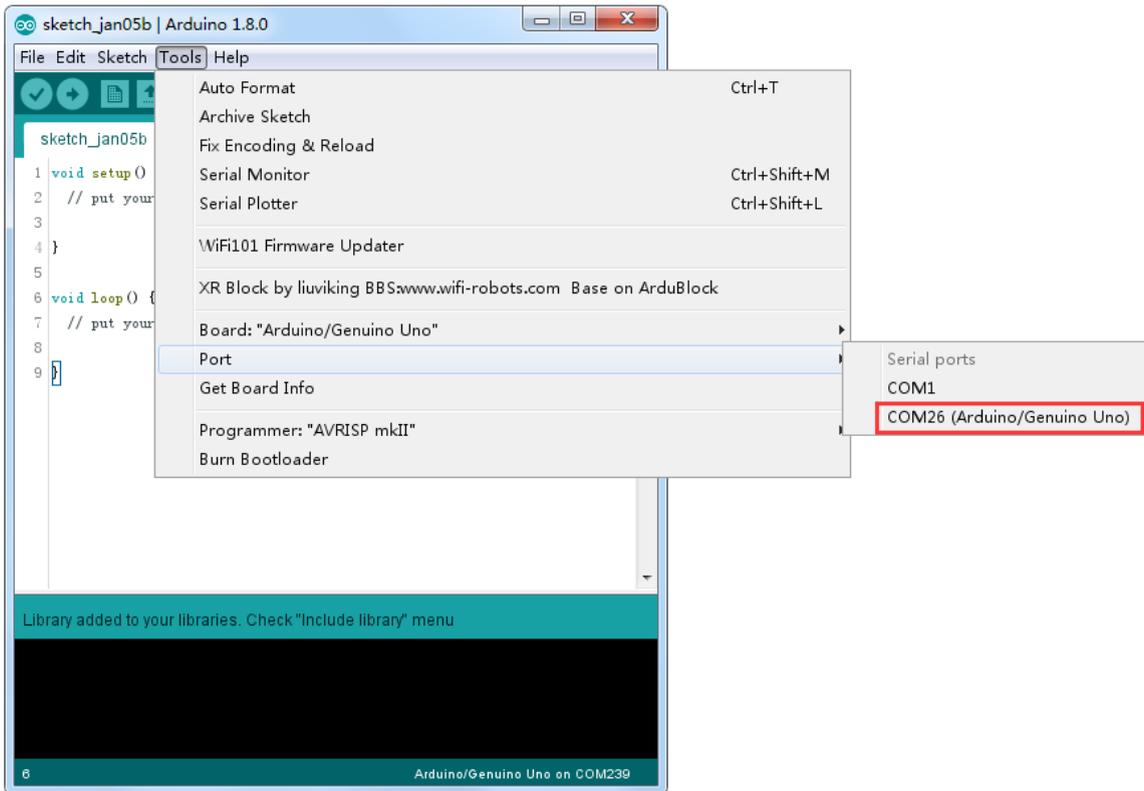
接続

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。

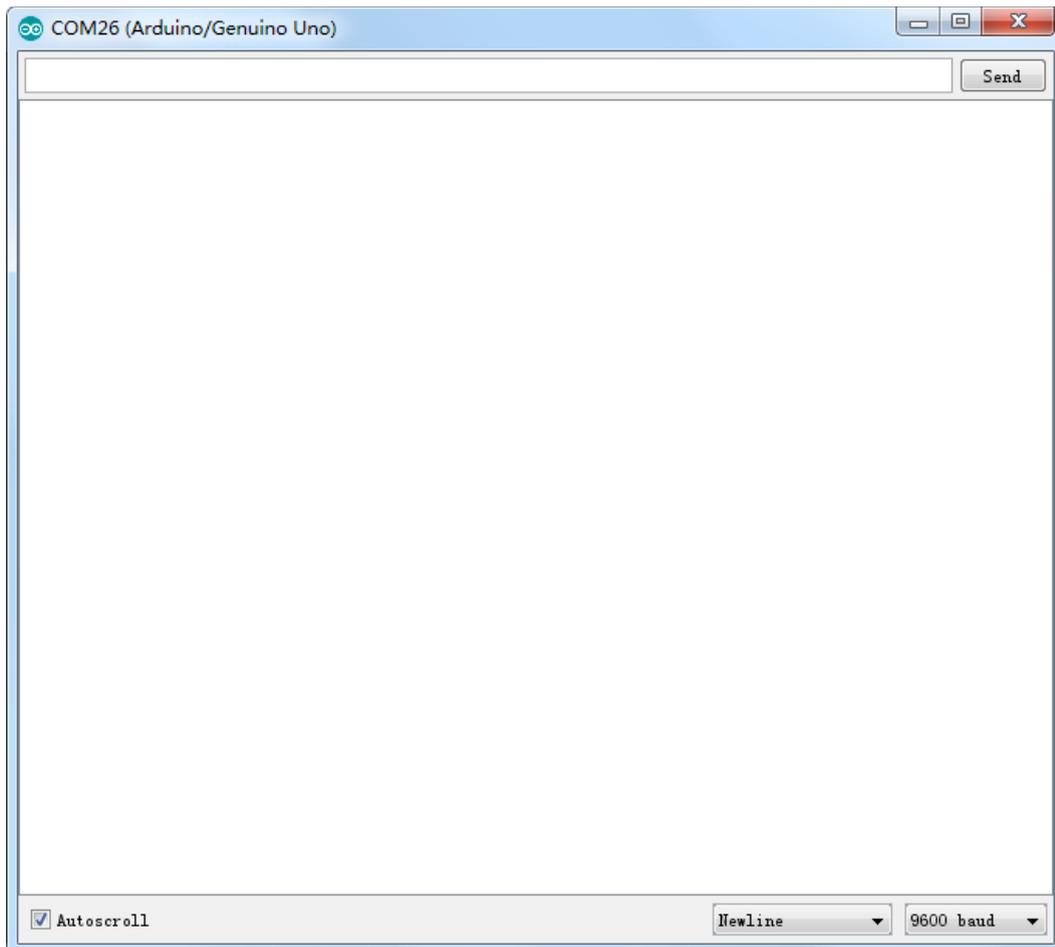


シリアルモニタで開くポートを選択することは、Arduino コードをアップロードするポートを選択することと同じです。「ツール」 -> 「シリアルポート」に移動し、正しいポートを選択します。

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。

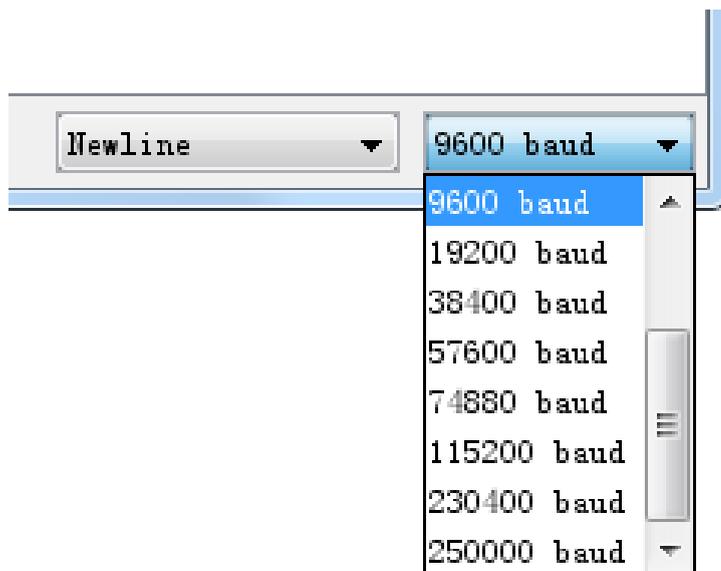


開いたら、次のように表示されます:

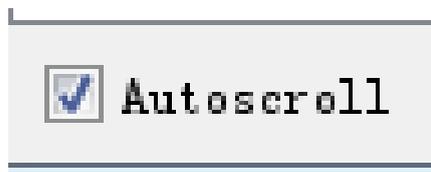


設定

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 ボー)



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。。

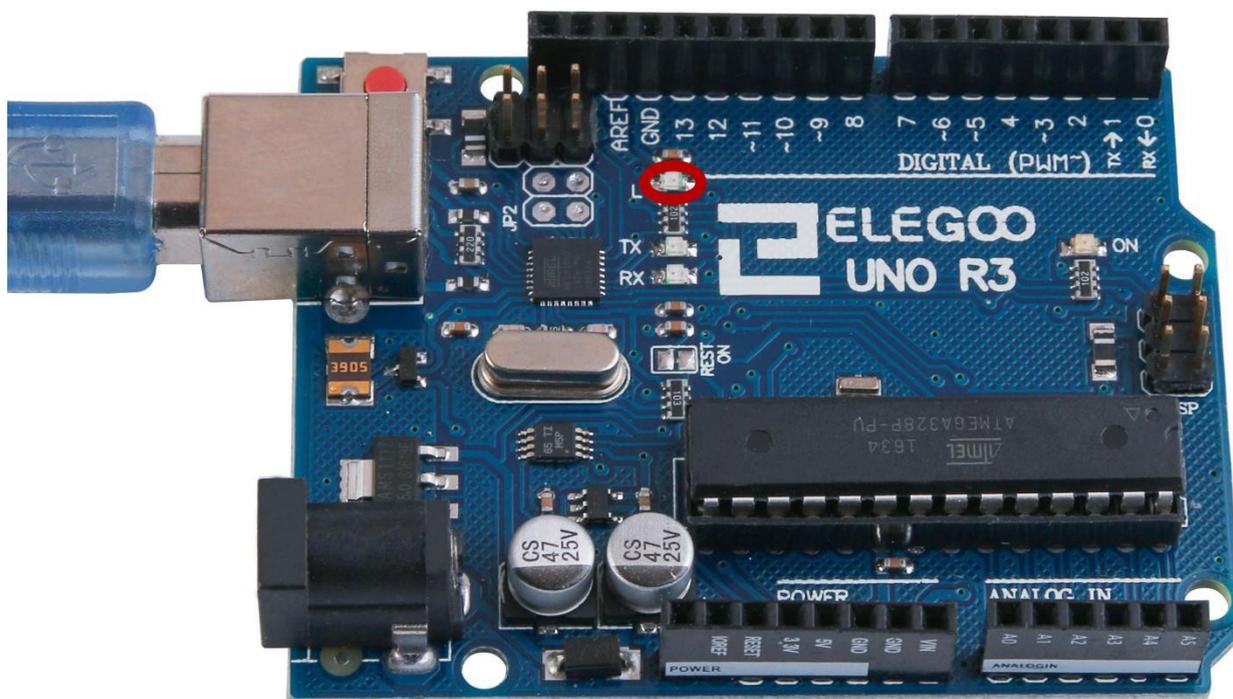
必要な構成部品:

(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。

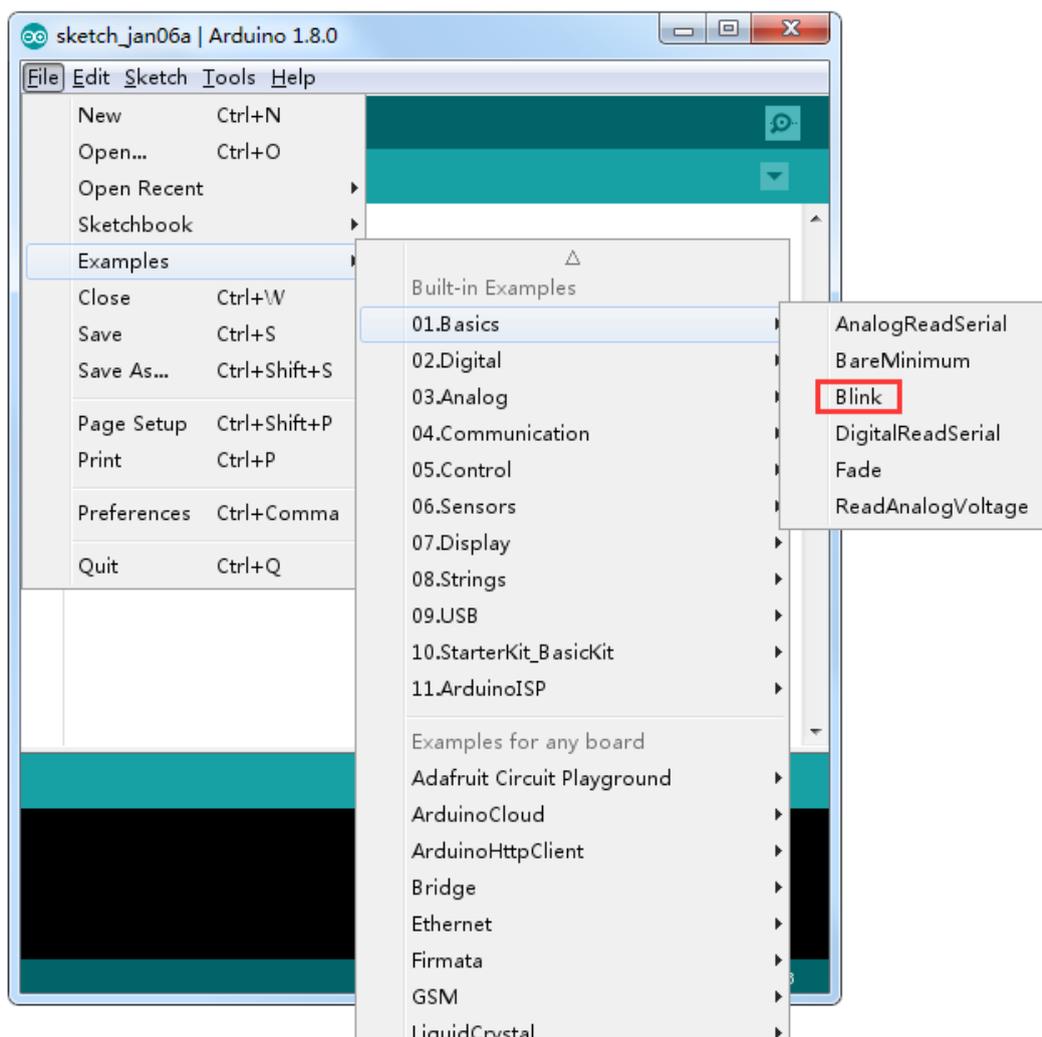
このレッスンでは、UNO R3 ボードを独自の Blink スケッチで再プログラムし、点滅するレートを変更します。

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。

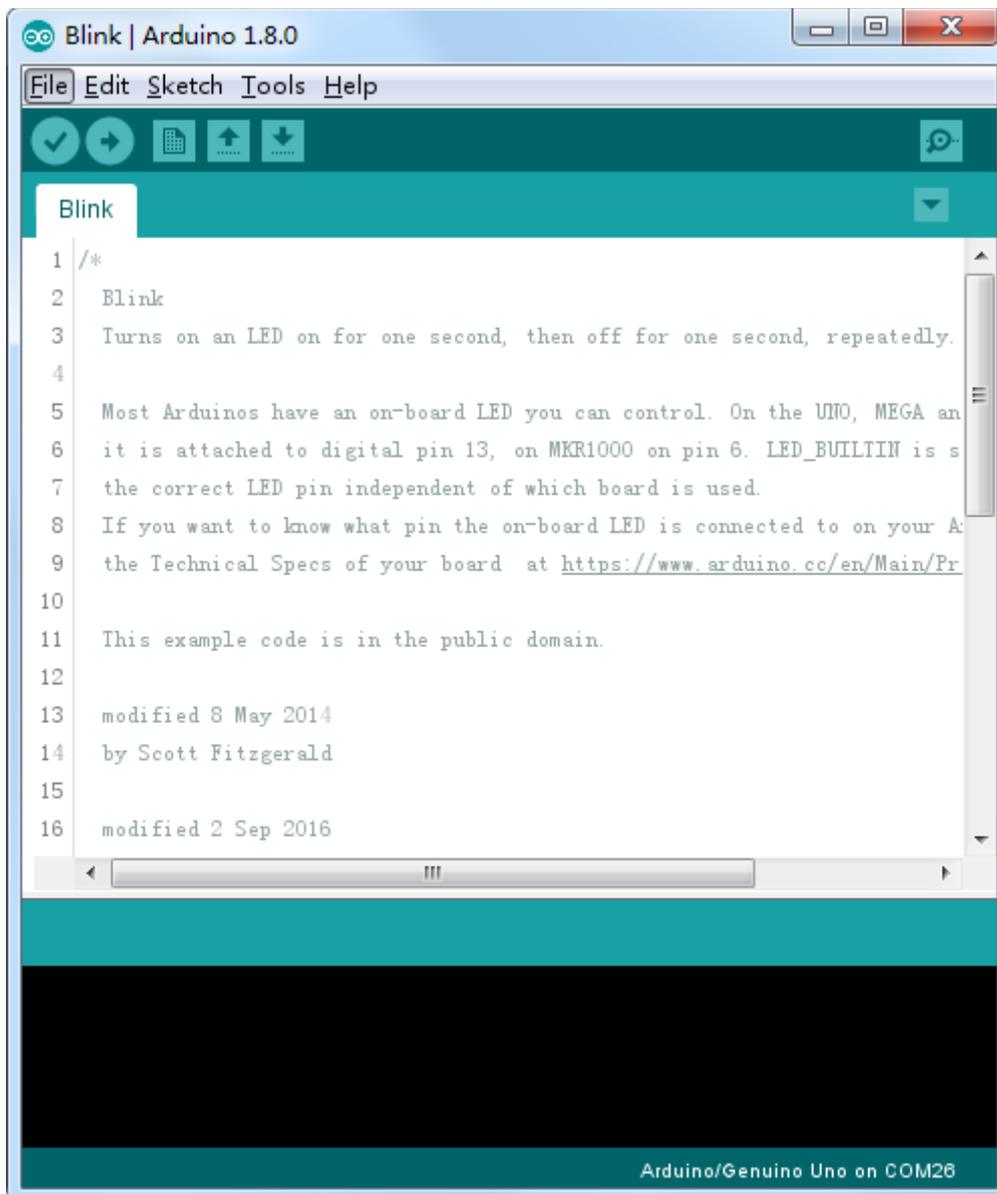
Arduino IDE には、ロードして使用できるサンプルスケッチの大きなコレクションが含まれています。

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 のメニューシステムで[ファイル]> [サンプル]> [01.Basics]の[Blink]スケッチをロードします。

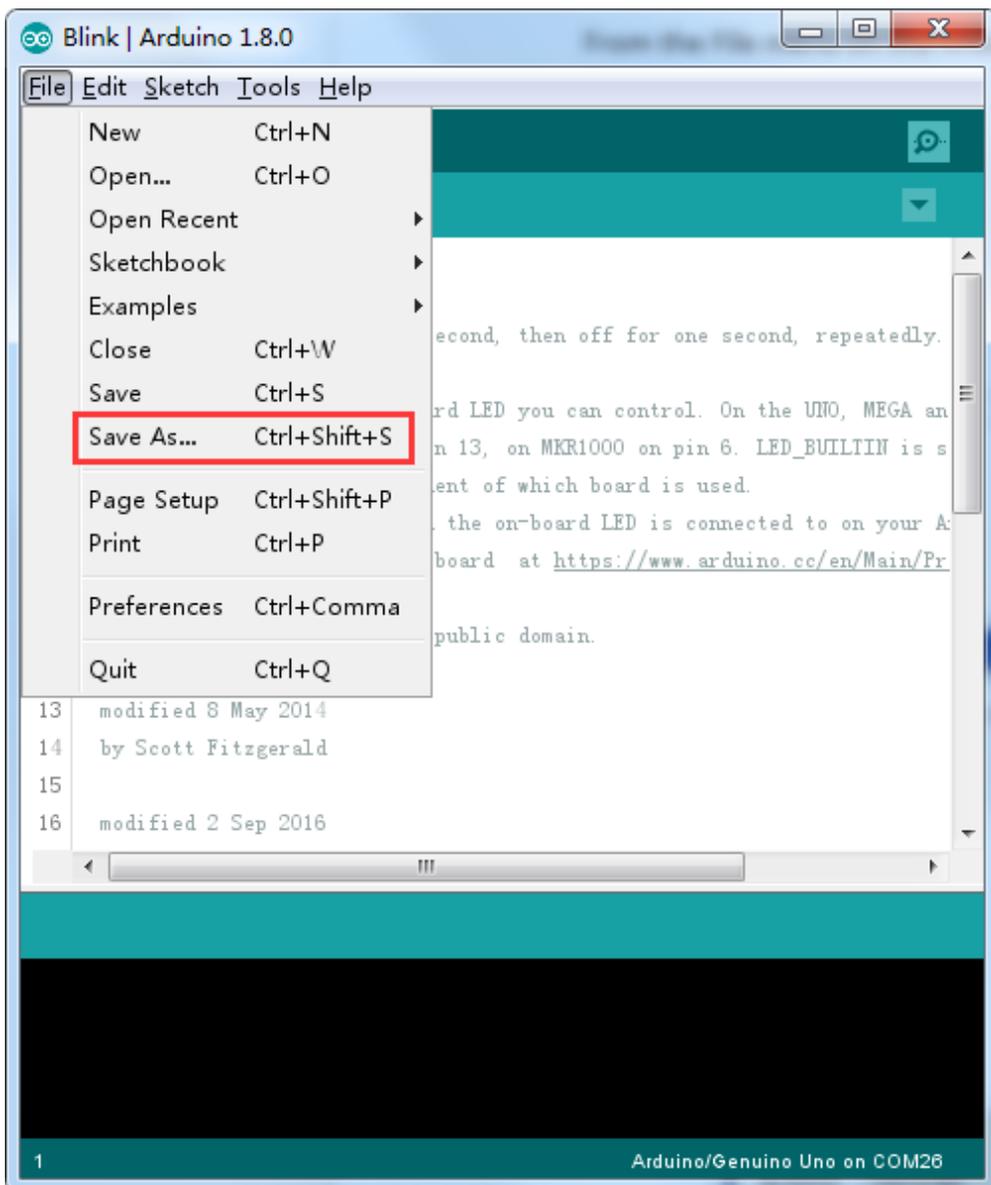


スケッチウィンドウが開いたら、拡大してスケッチ全体をウィンドウで見ることができます。



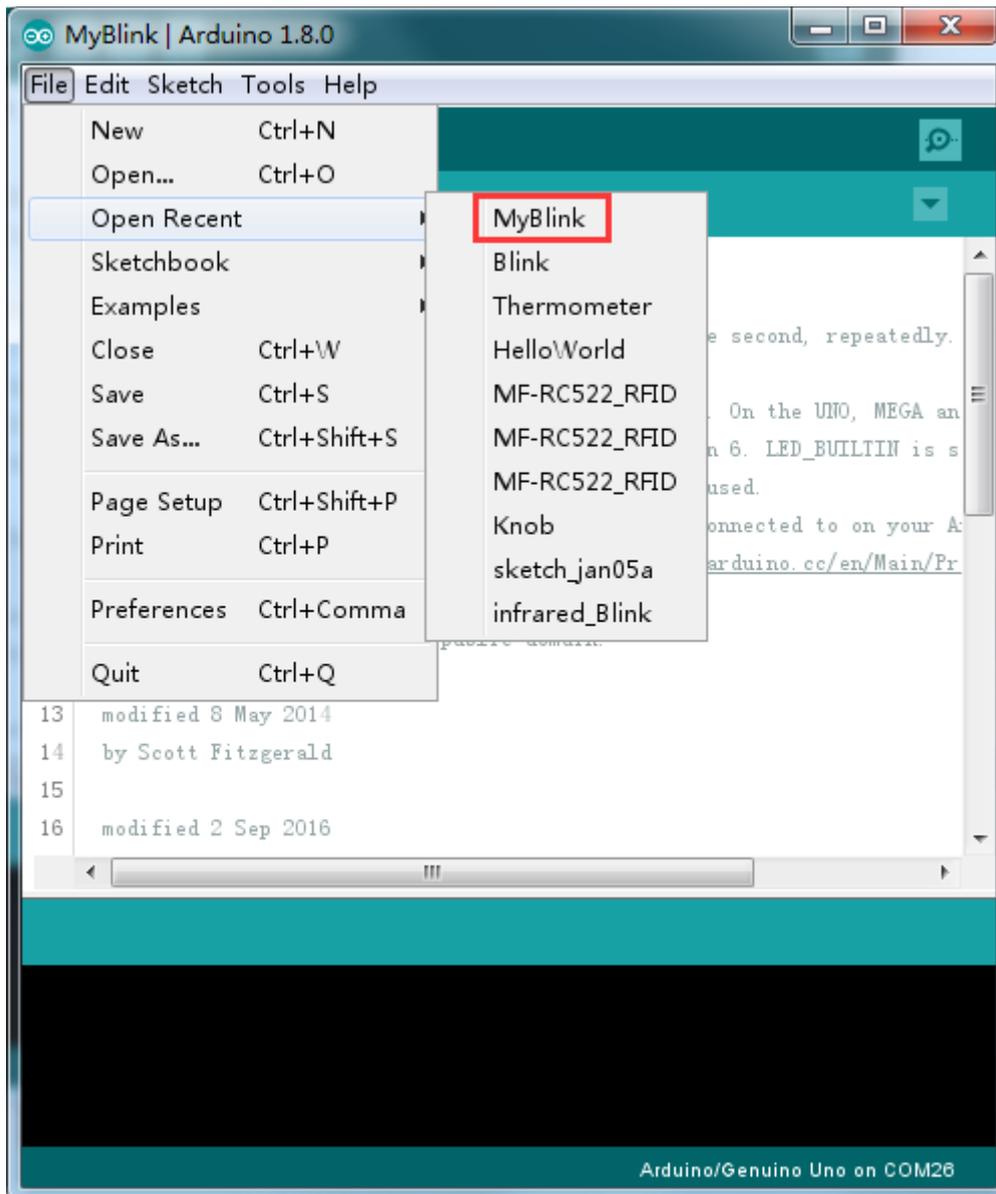
Arduino IDEに含まれているスケッチの例は「読み取り専用」です。つまり、それらを UNO R3 ボードにアップロードできますが、変更した場合は同じファイルとして保存することはできません。

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。

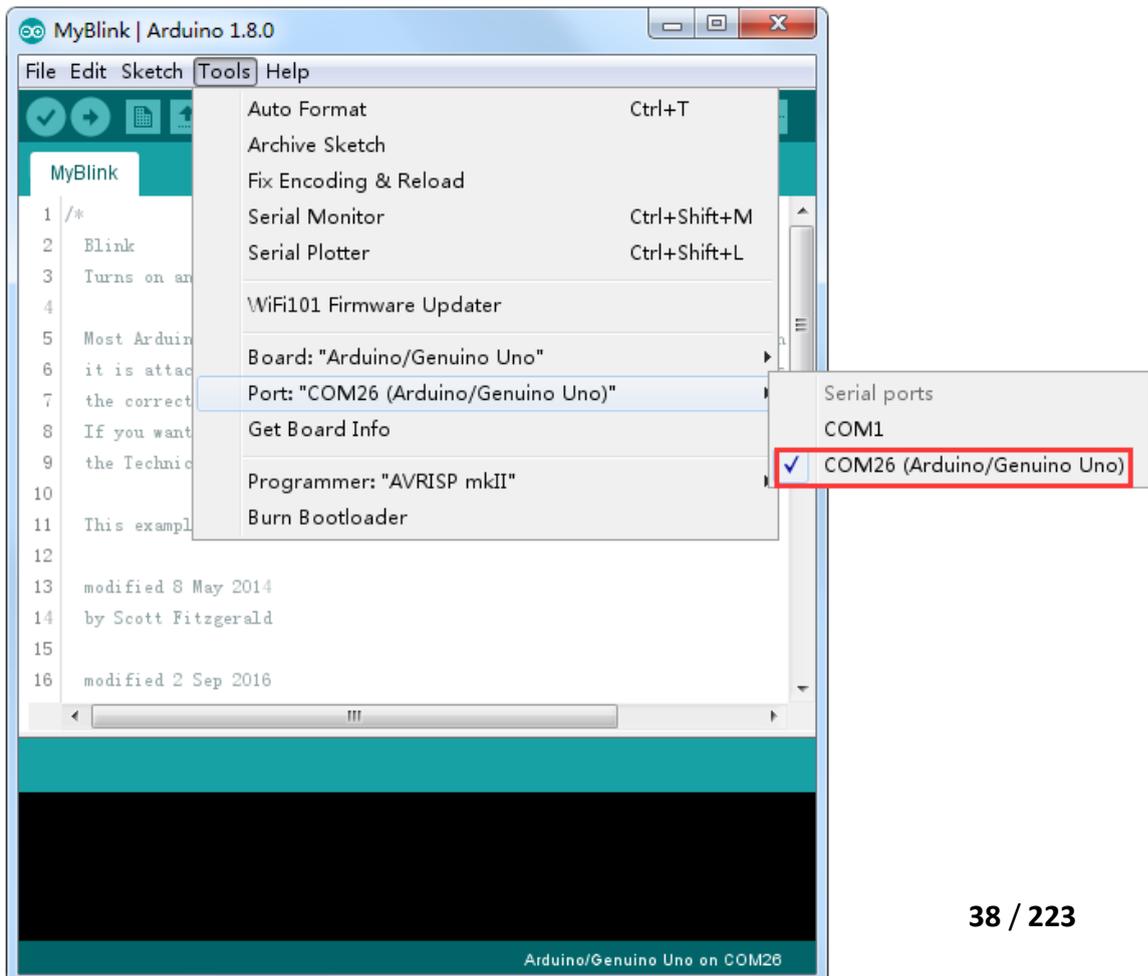
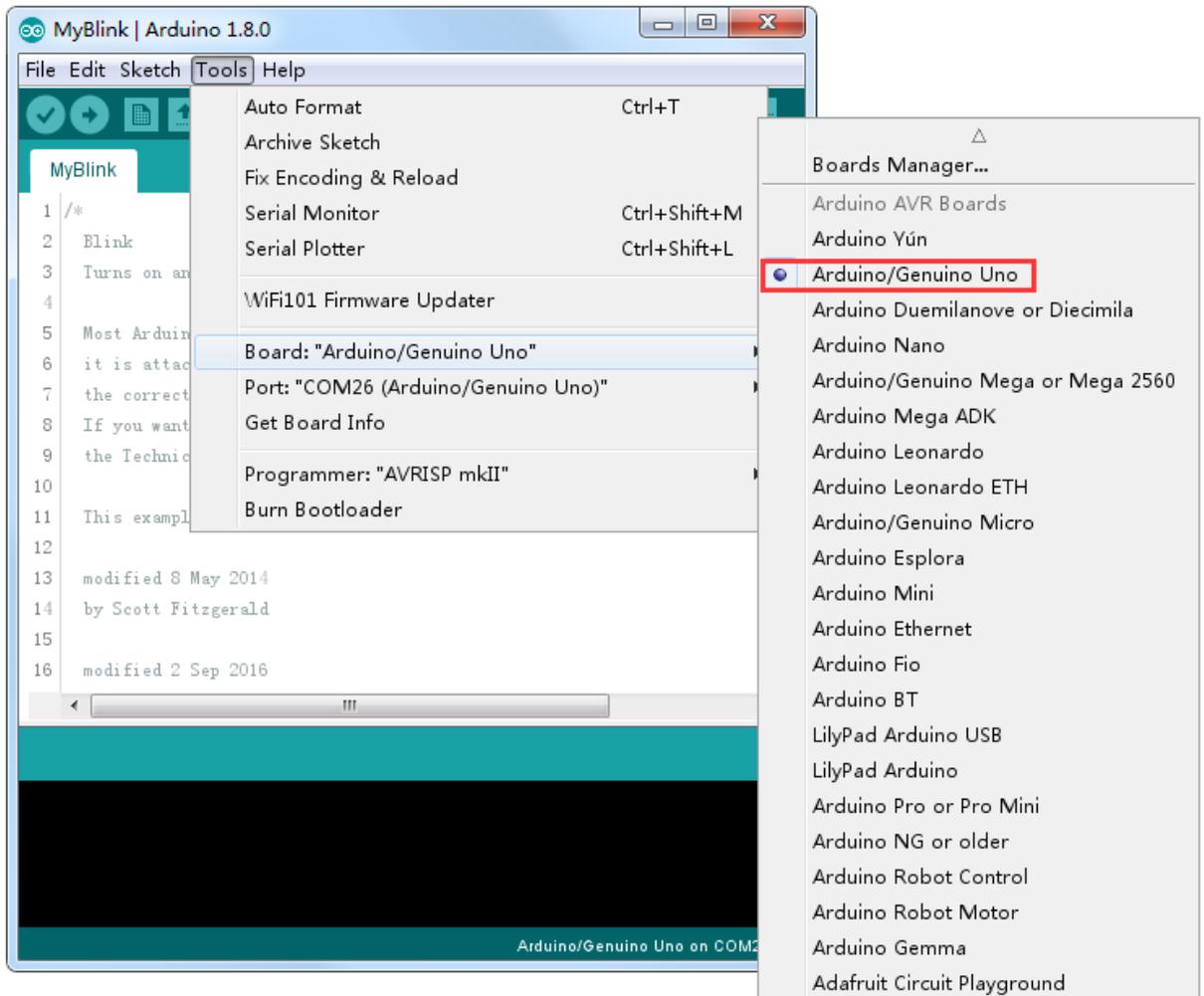




スケッチブックにあなたのコピーを保存しました。つまり、これを再度検索したい場合は、「ファイル」>「スケッチブック」メニューオプションを使用して開くことができます。

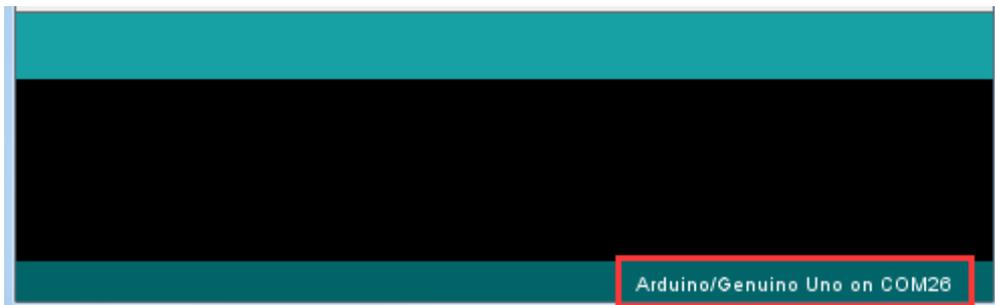


コンピュータに Arduino ボードを USB ケーブルで接続し、'Board Type'と 'Serial Port'が正しく設定されていることを確認します。



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) であるはずですが。

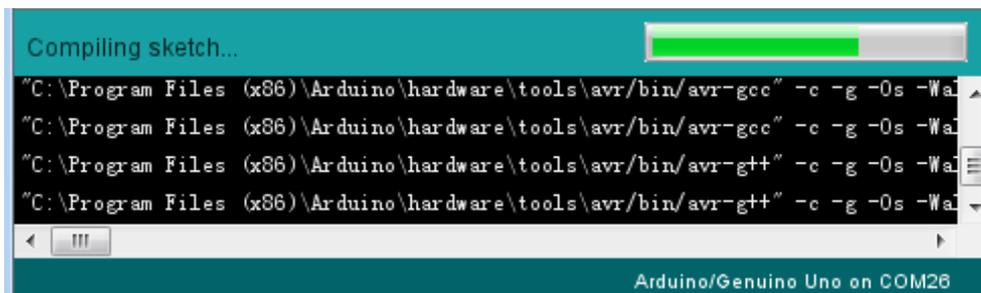
Arduino IDE は、ウィンドウの下部にあるボードの現在の設定を表示します。



[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、進行状況バーと一連の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が。

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{}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。

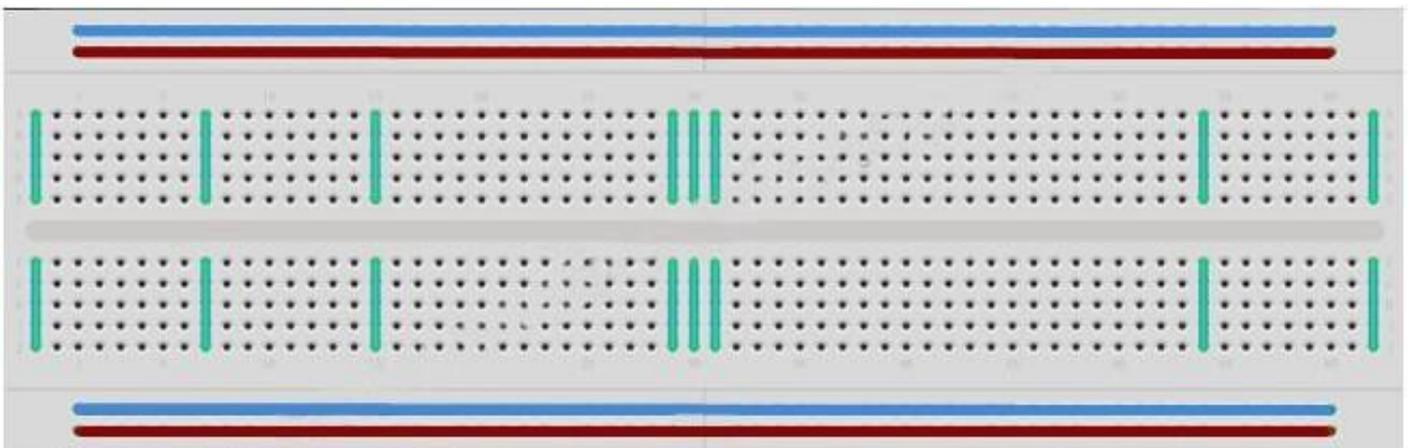
必要な構成部品:

- (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氣的に接続するための金属ストリップがあります。2つの異なるコンポーネントの脚部を同じ列に押し込むと、それらを電氣的に結合します。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ボルトとグラウンドのペアになっています。これらのストリップはルールと呼ばれ、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もあります。

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をバッテリー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。さもないと、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:

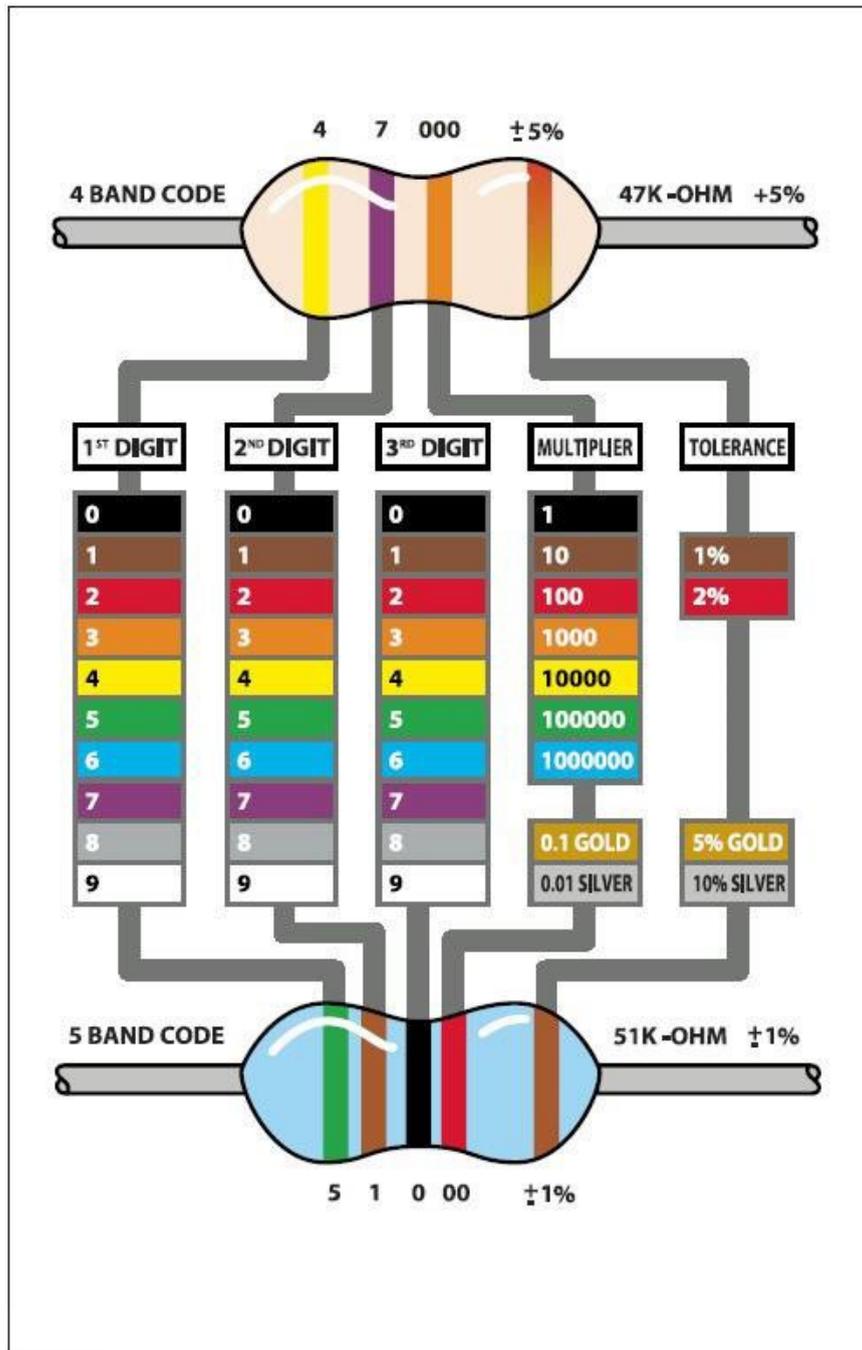
名前が示すように、抵抗器は電気の流れに抵抗します。抵抗の値が高いほど、抵抗が大きくなり、流れる電流は少なくなります。LED をどれだけ電流が流れているかを制御するために、これを使用して、それがどれだけ明るく輝いているかを制御します。



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場合 (小さい抵抗です)、 $k\Omega$ (1,000 Ω) と $M\Omega$ (1,000,000 Ω) 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、220 Ω 、1k Ω 、10k Ω 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、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。

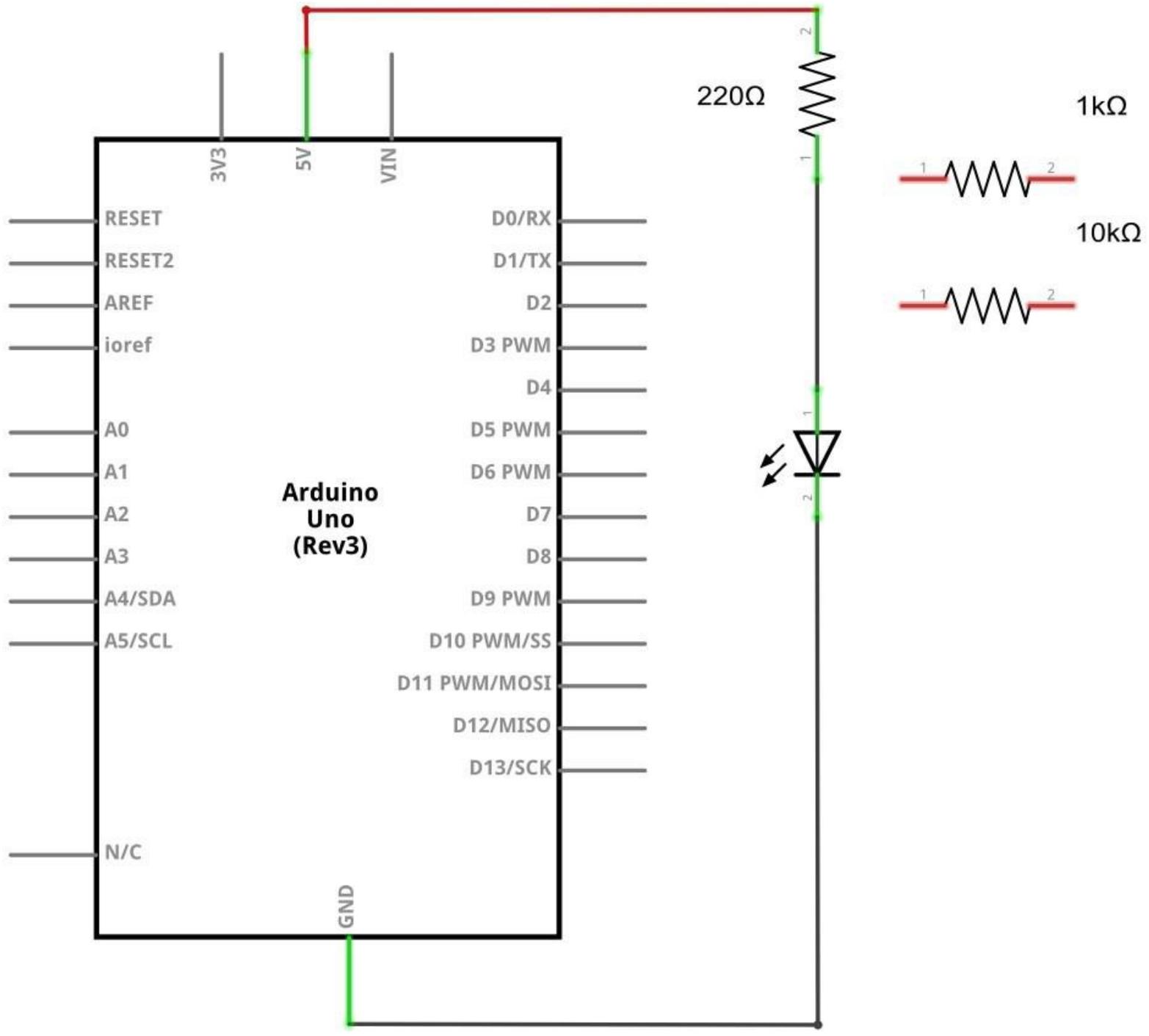


LED とは異なり、抵抗には正と負のリードがありません。それらはいずれかの方法で接続することができます。

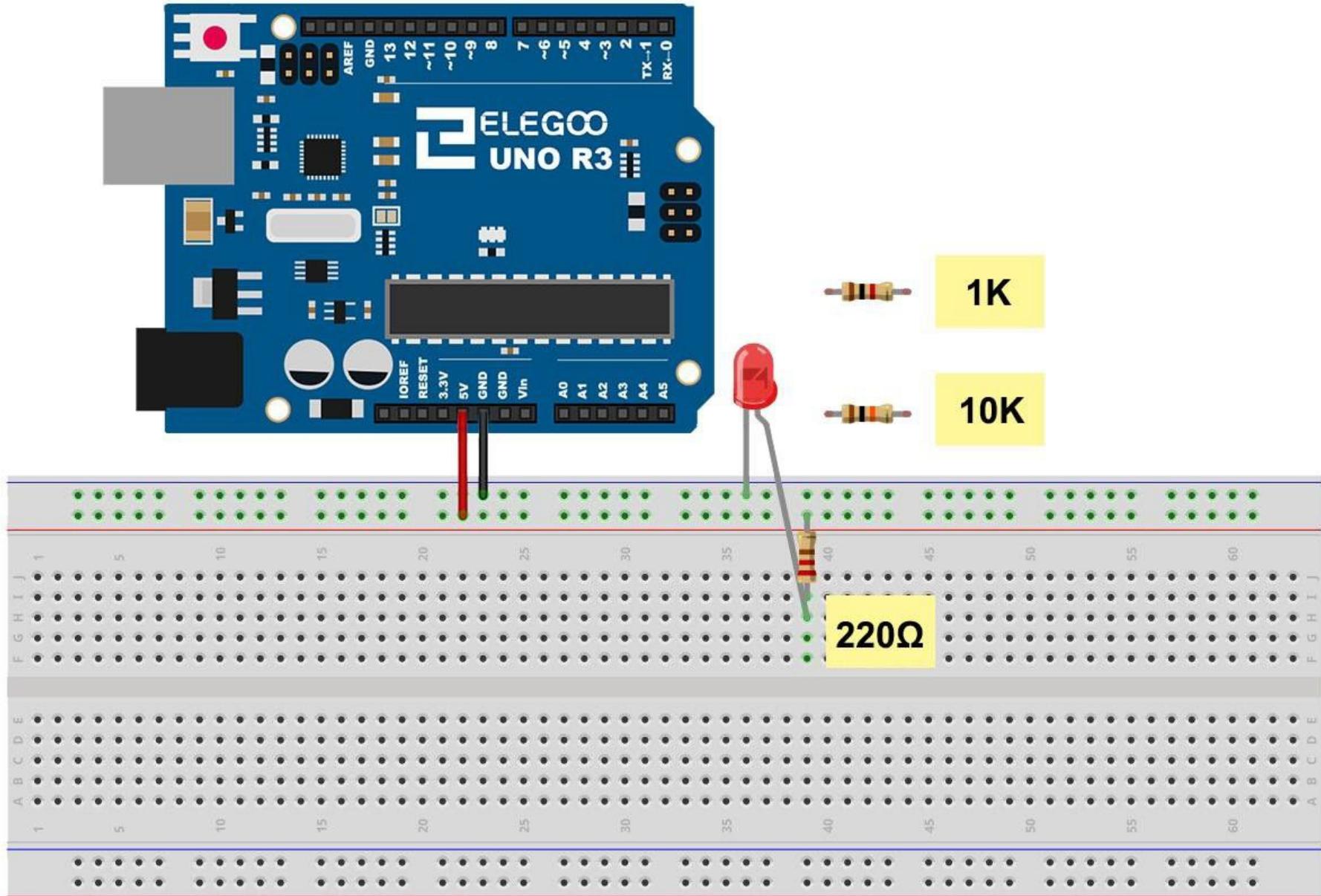
このアプローチ方法が複雑すぎるとわかったら、私たちの抵抗のカラーリングフラグを直接読んでその抵抗値を決定することができます。または、代わりにデジタルマルチメータを使用することもできます。

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。

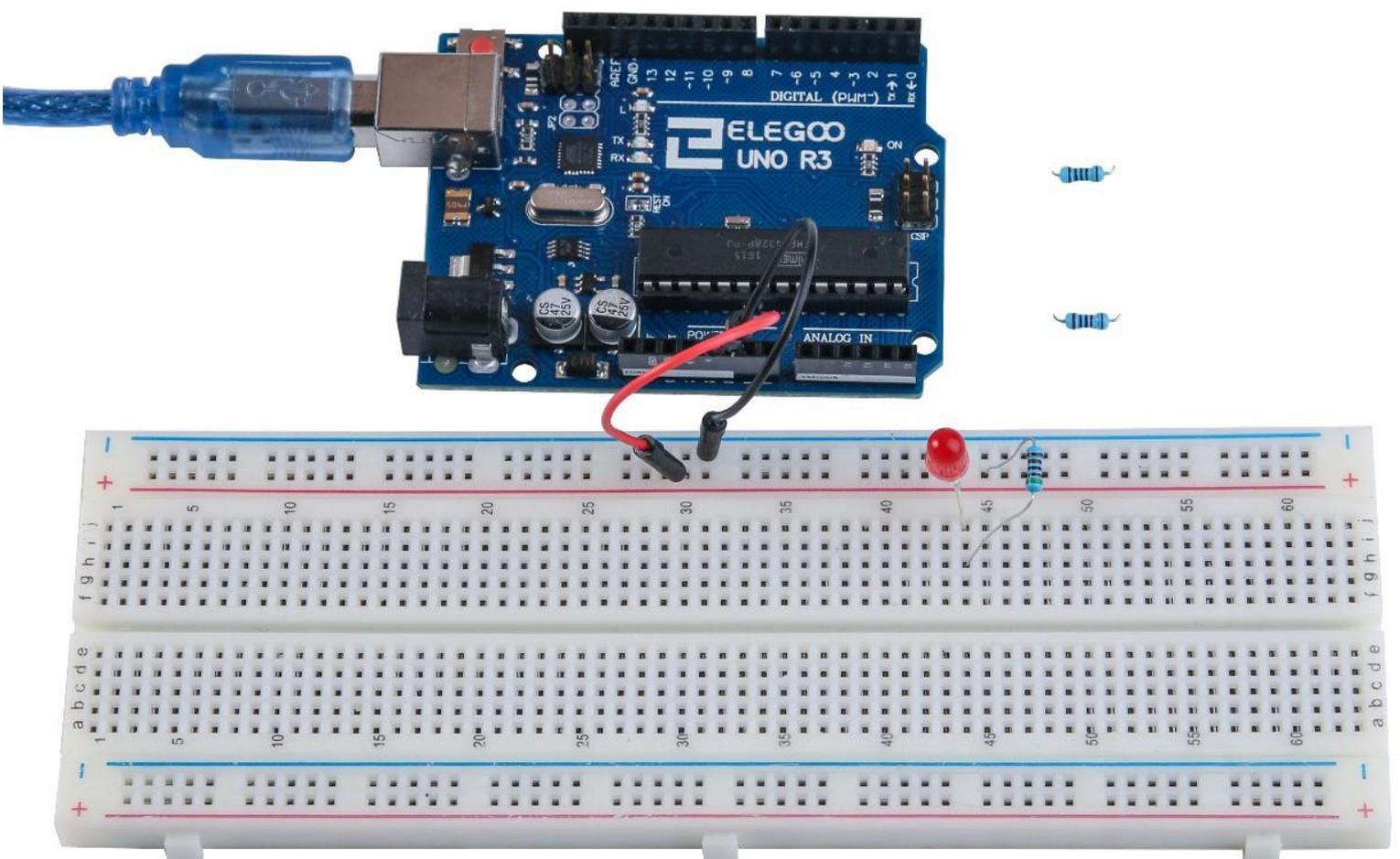
220Ωの抵抗を取り付けた場合、LED は非常に明るいはずですが、1kΩ抵抗の 220Ω抵抗を交換すると、LED が少し減光します。 最後に、10kΩの抵抗を配置すると、LED が目に見えます。 赤いジャンパーリードをブレッドボードから引き出し、穴の中に触れて取り除き、スイッチのように動作させます。 あなたは違いを気付くことができます。

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ウンドに接続します。

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 jumper wires)
- (1) x RGB LED
- (3) x 220 ohm resistors

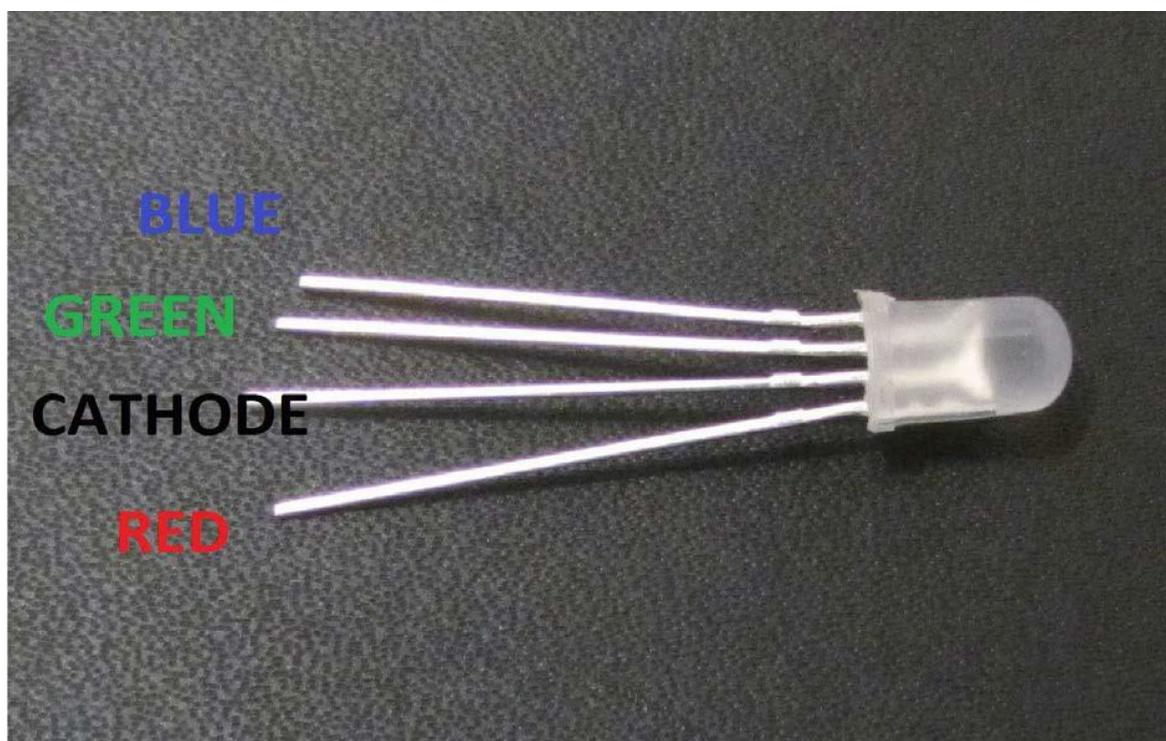
部品の紹介

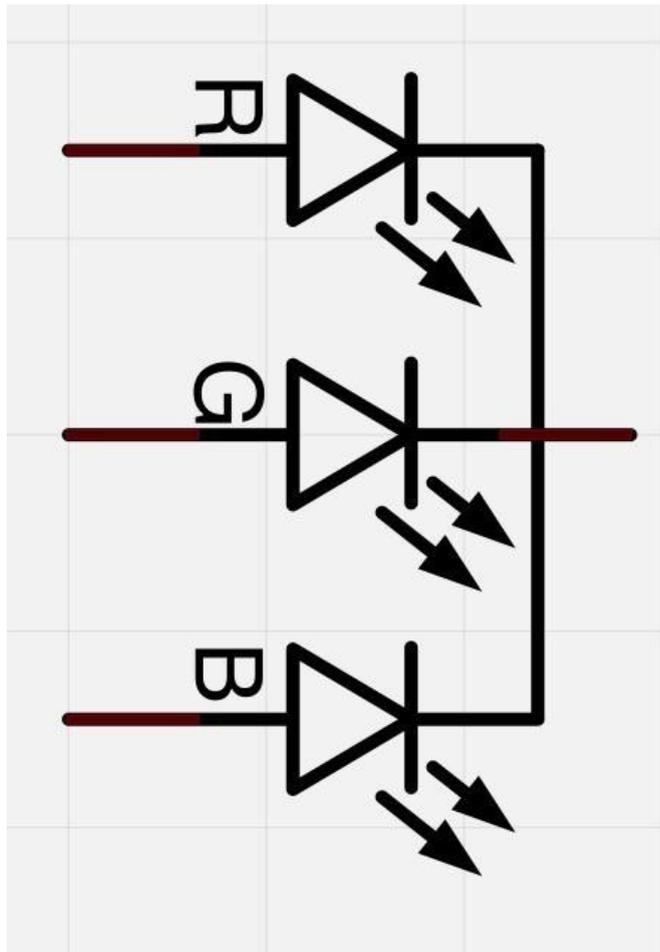
RGB:

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。

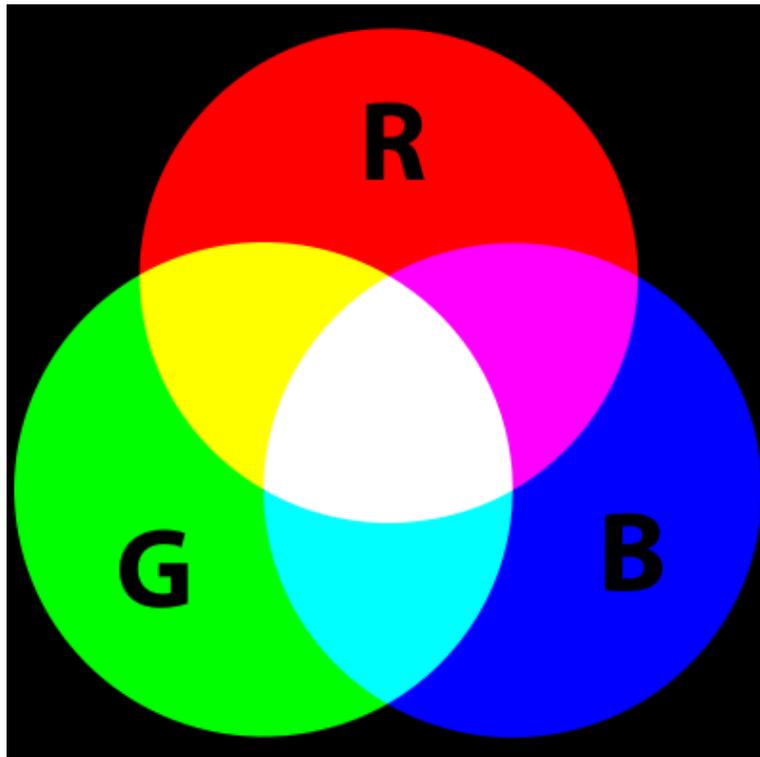
LED パッケージの共通の負の接続は、平らな側から 2 番目のピンです。 それはまた、4 つのリード線のうち最長であり、接地されます。

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種類の光レセプター（赤、緑、青）があるからです。あなたの目と脳は、赤、緑、青の量进行处理し、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。



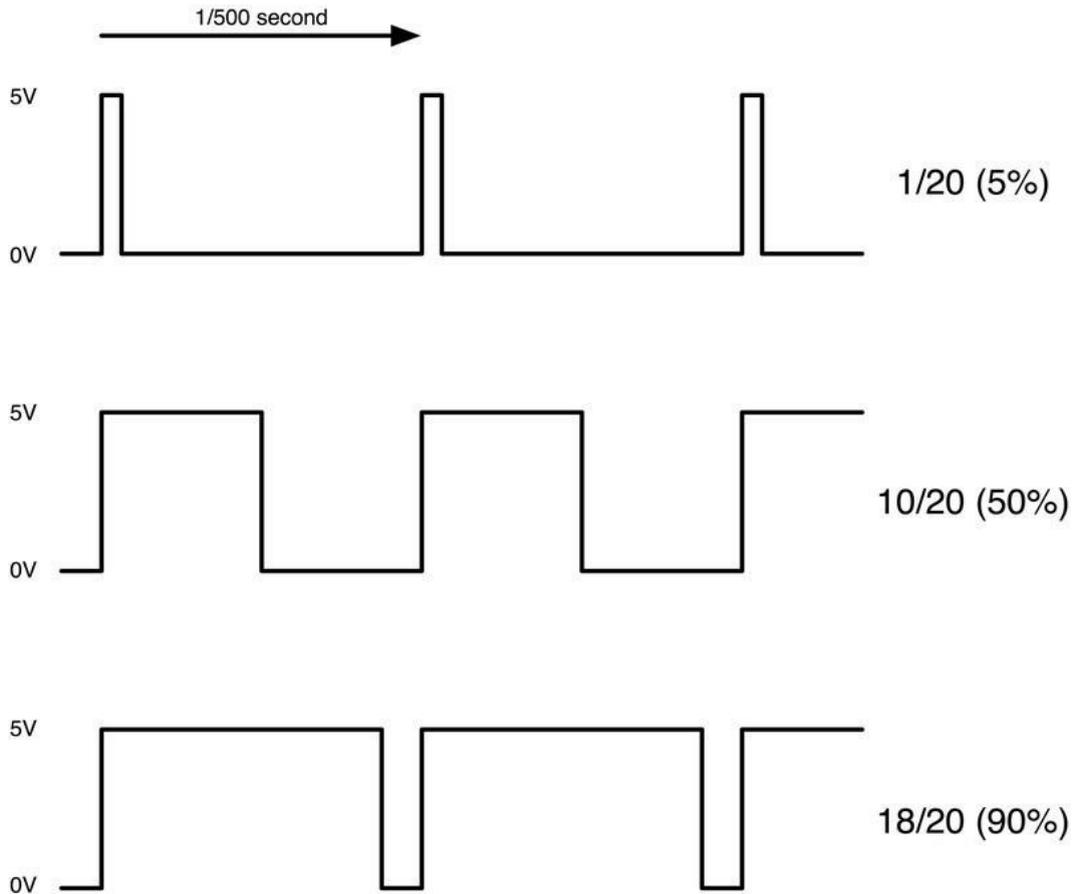
3つすべてのLEDの輝度を同じに設定すると、光の全体的な色は白色になります。赤と緑のLEDだけが同じ明るさになるように青のLEDを消すと、ライトは黄色に見えます。

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。

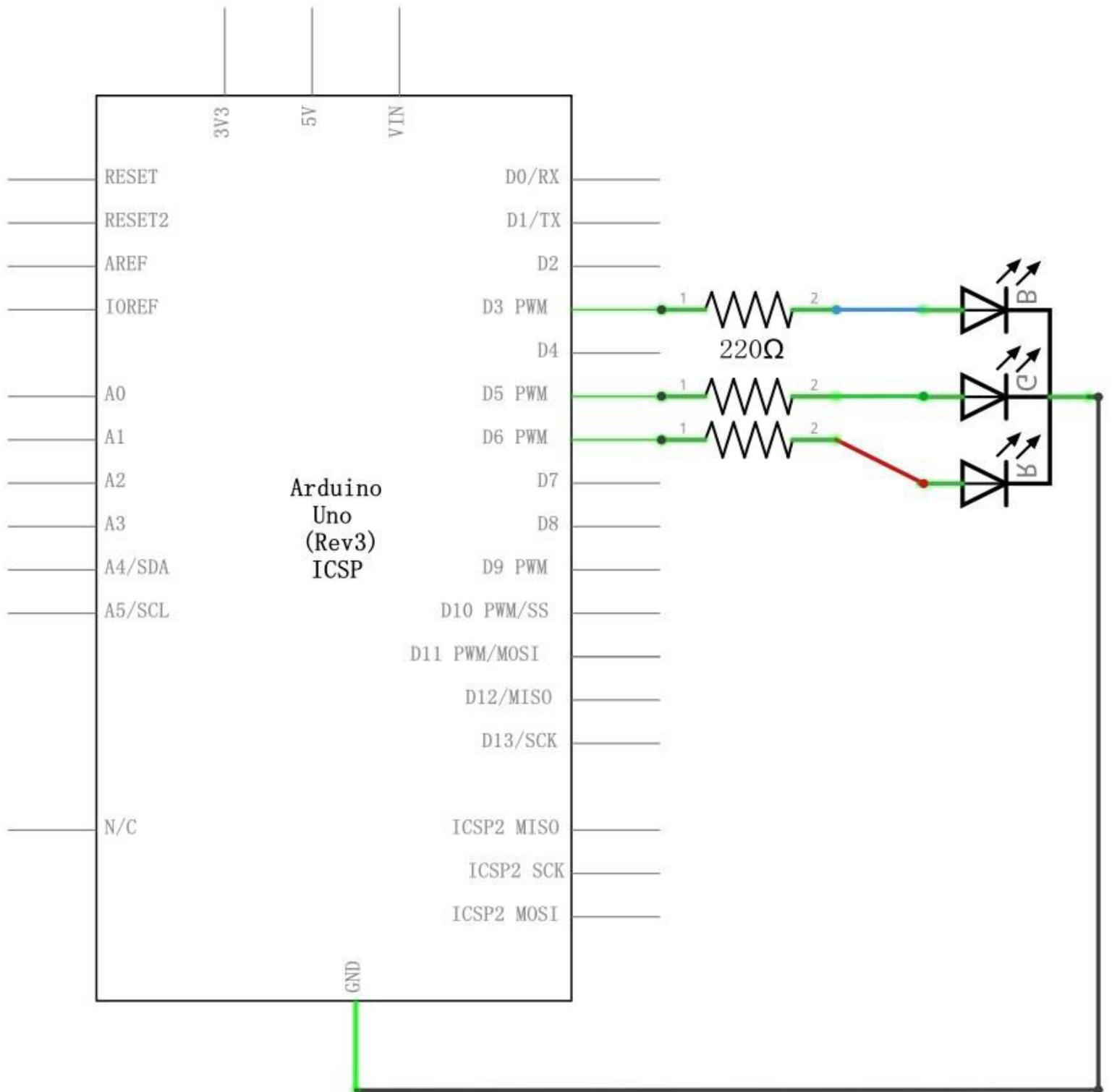
このパルスの長さは、'analogWrite'関数によって制御されます。したがって、'analogWrite (0)'はパルスをまったく生成せず、'analogWrite (255)'は次のパルスが到着するまですべての時間持続するパルスを生成します。その結果、実際には常に出力されます。

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。

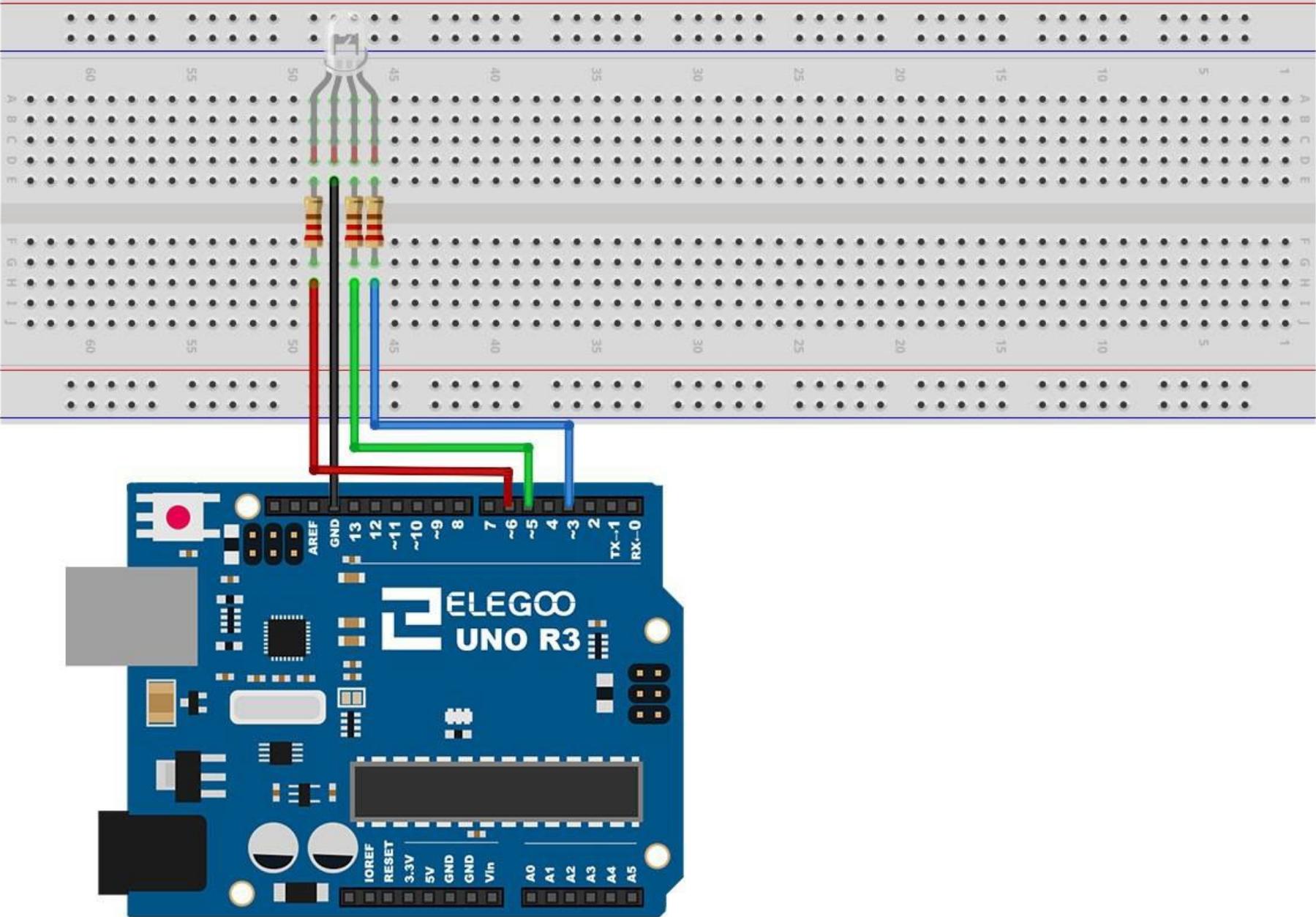
しかし、出力が時間の 90%の間 5V であれば、負荷はそれに供給される電力の 90%を得ます。そのスピードで LED の点滅が見えないので、明るさが変わっているように見えます。

Connection

Schematic

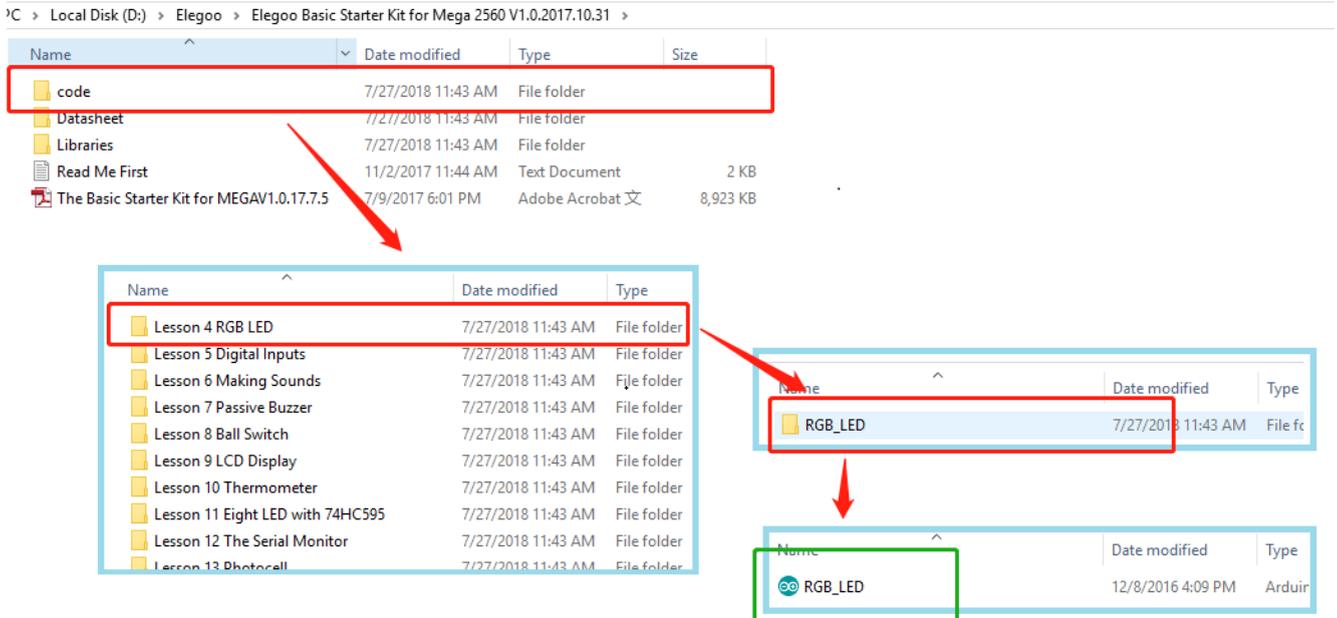


Wiring diagram



Code

配線後、[Tutorial]> [日本語]> [code]> [Lesson 4 RGB LED]> [RGB_LED]の[フォルダ内のスケッチ]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のアップロードについて詳しくは、レッスン2を参照してください。

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。

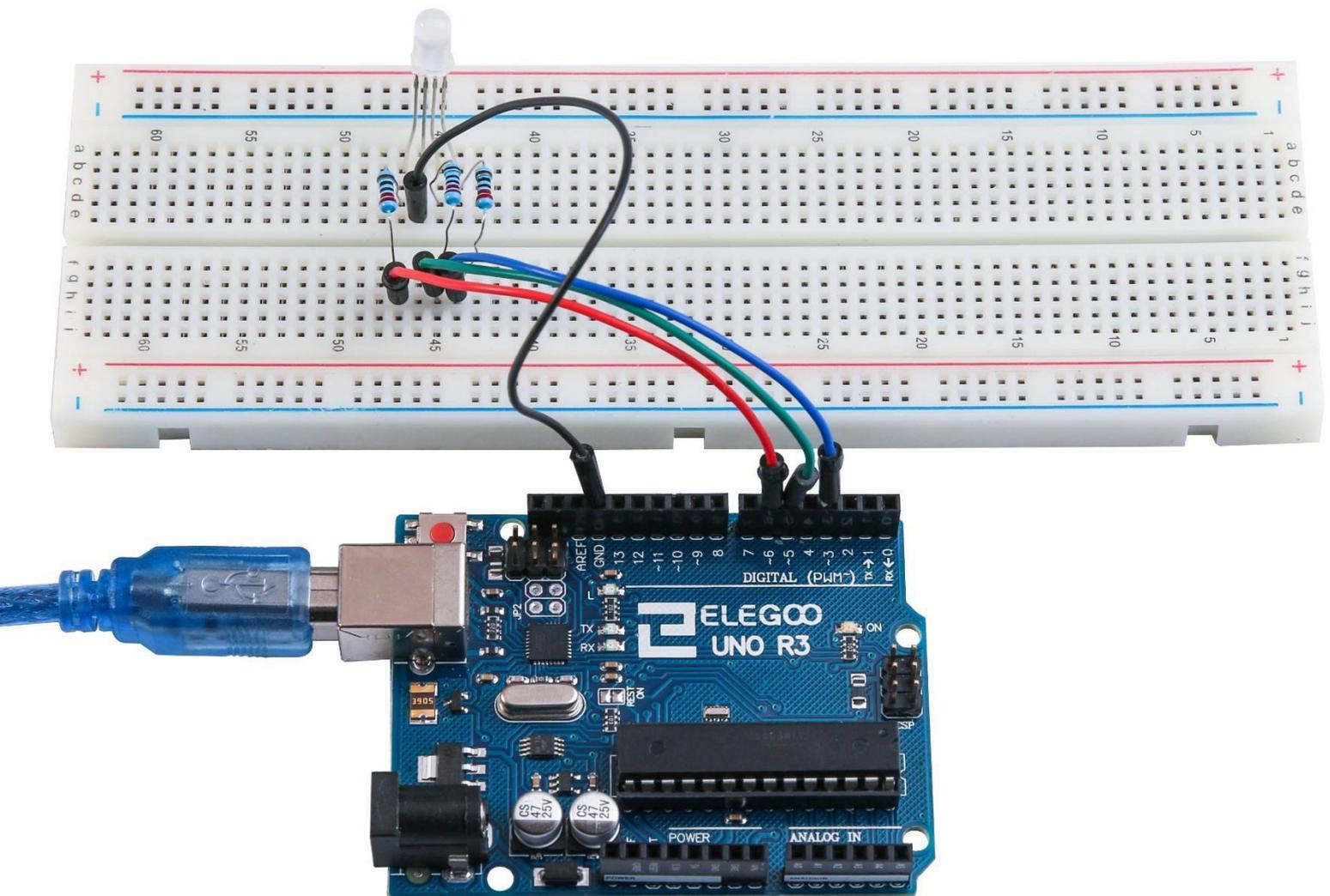
'ループ'機能を見ると、赤色、緑色、青色の光量を表示してから 1 秒間一時停止してから次の色に移動することがわかります。

```
#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:

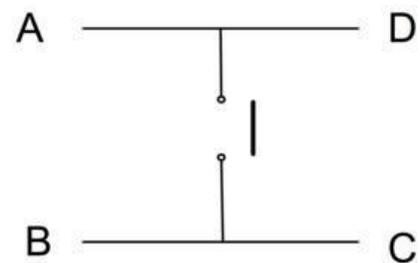
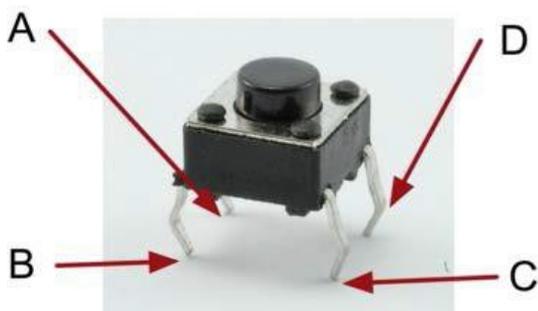
- (1) x Elegoo Uno R3
- (1) x 830 Tie-points Breadboard
- (1) x 5mm red LED
- (1) x 220 ohm resistor
- (2) x push switches
- (7) x M-M wires (Male to Male jumperwires)

部品の紹介

PUSH SWITCHES:

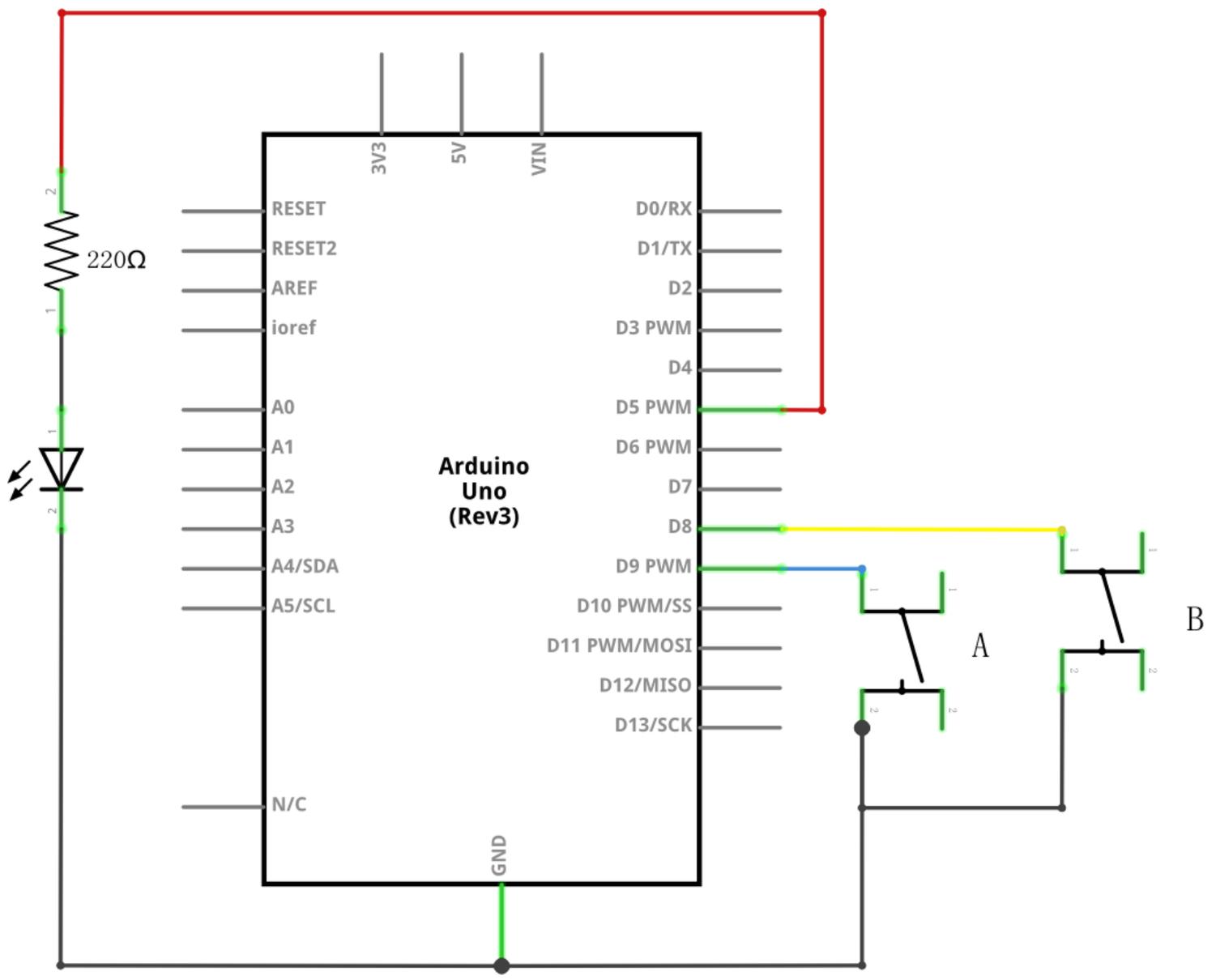
スイッチは本当にシンプルなコンポーネントです。ボタンを押すかレバーを回すと、2つの接点が電氣的に接続されます。

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。

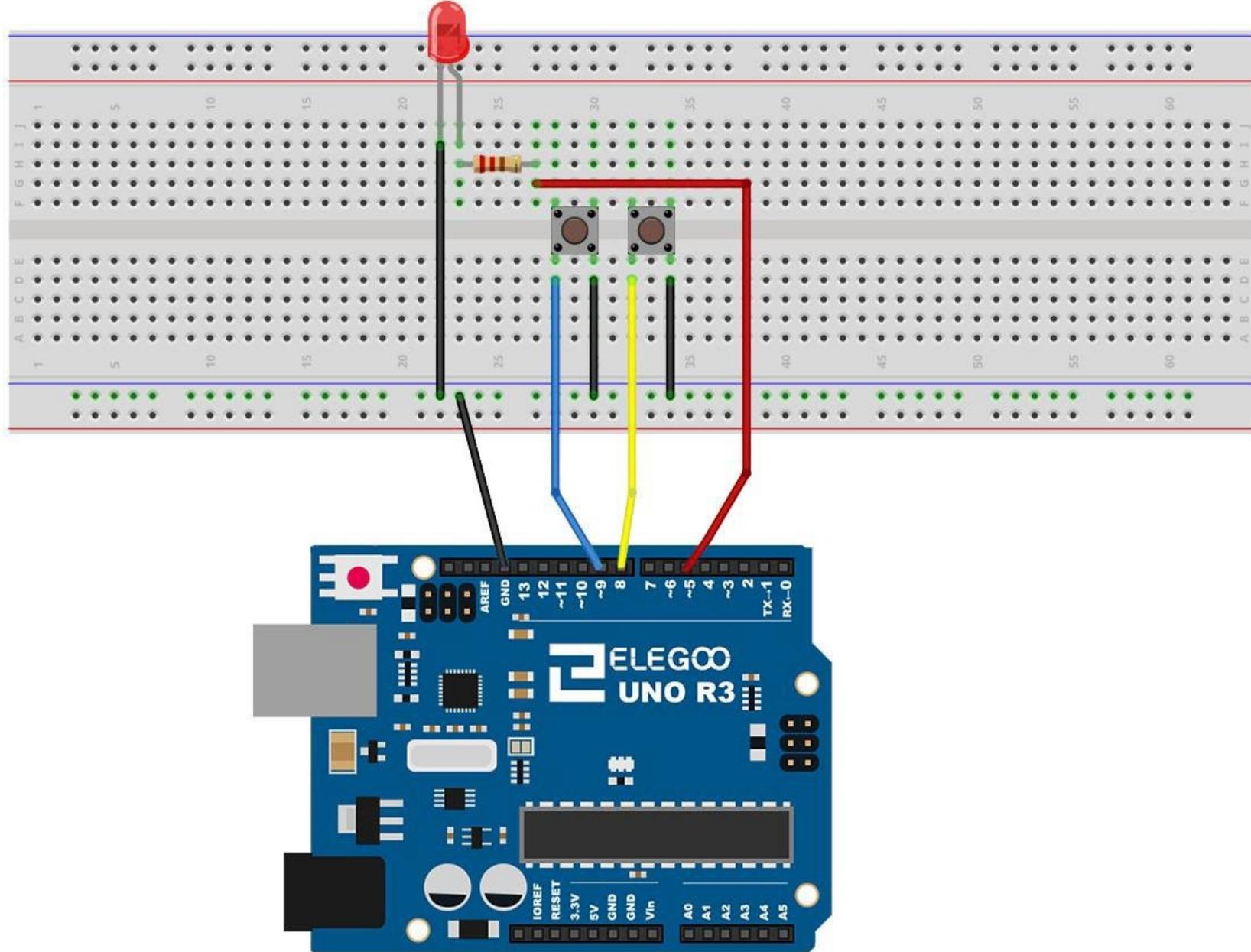


実際には、実際には電気接続が2つしかありません。スイッチパッケージ内では、AとDのようにピンBとCが一緒に接続されています。

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いか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。言い換えれば、入力のデフォルト値は、ボタンを押す動作によって LOW に引かれられない限り、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)
```

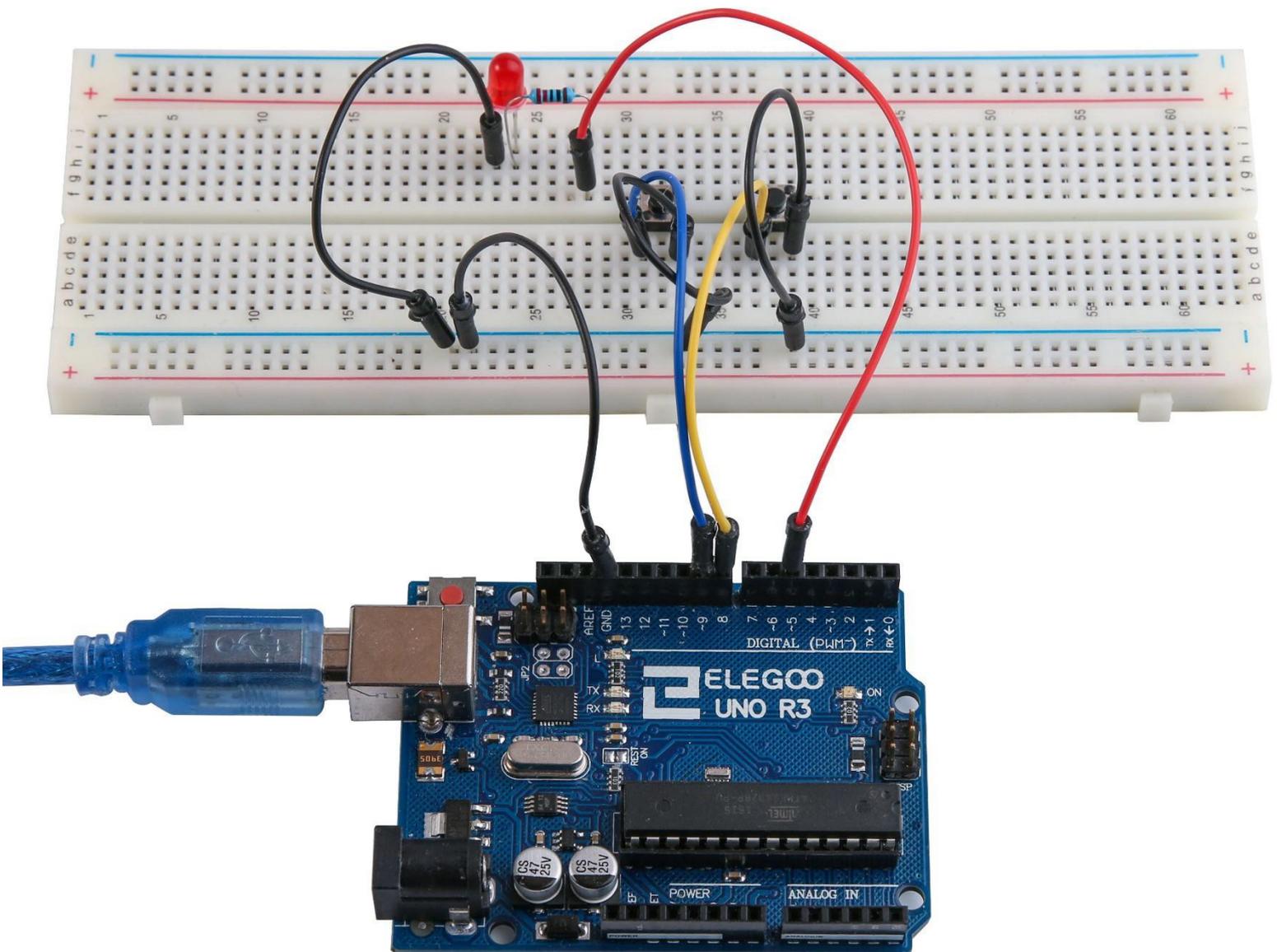
```
{  
  digitalWrite(ledPin, LOW);  
}  
}
```

'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:

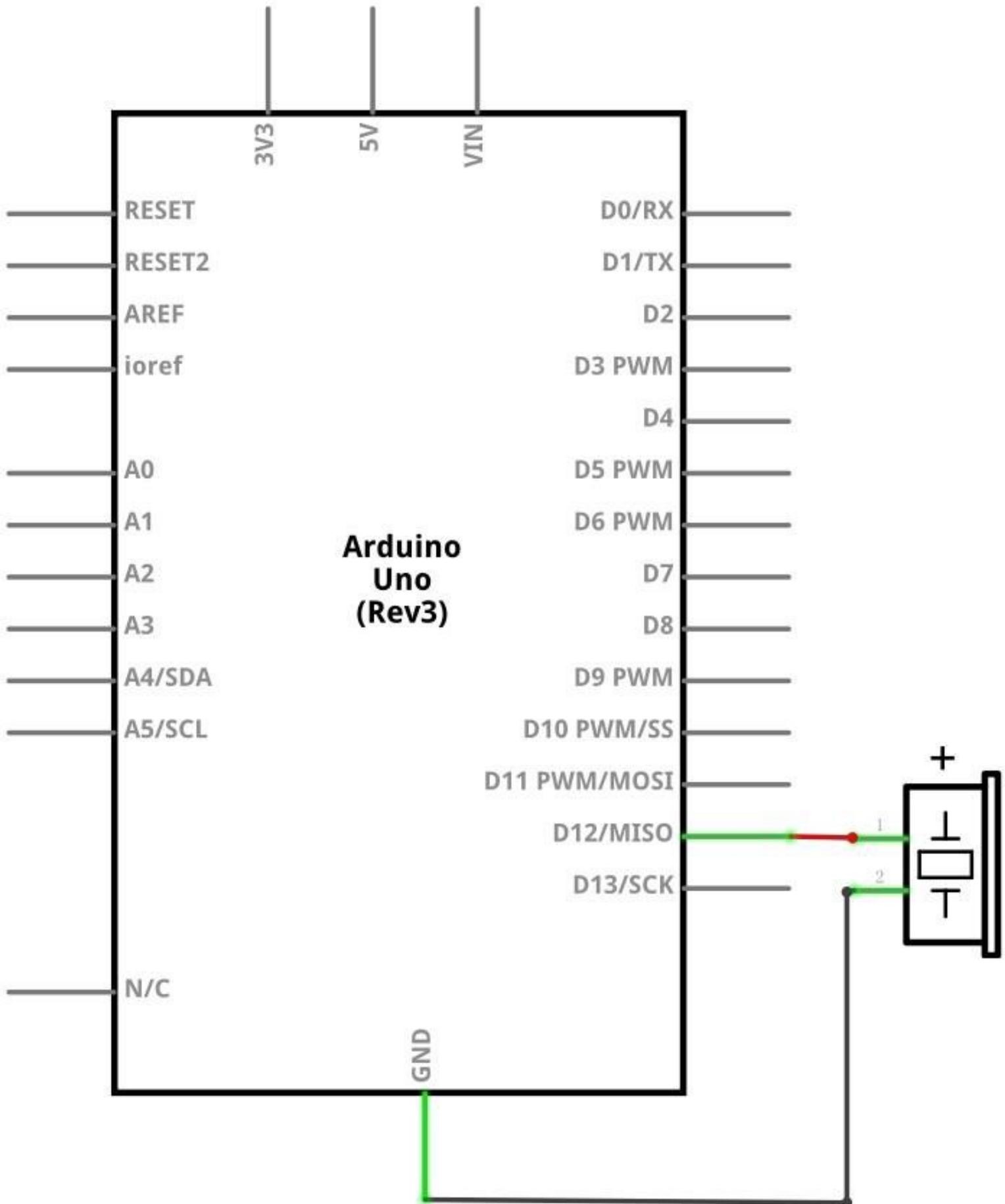
電子ブザーは DC 電源であり、集積回路を備えています。それらは、コンピュータ、プリンタ、複写機、アラーム、電子玩具、車載電子機器、電話機、タイマ、および音声デバイス用の他の電子製品に広く使用されている。ブザーはアクティブなものと同様に受動的なものに分類できます。2つのブザーのピンを上に向けます。緑色の回路基板を持つものは受動的なブザーですが、黒色のテープで囲まれたものはアクティブなブザーです。

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

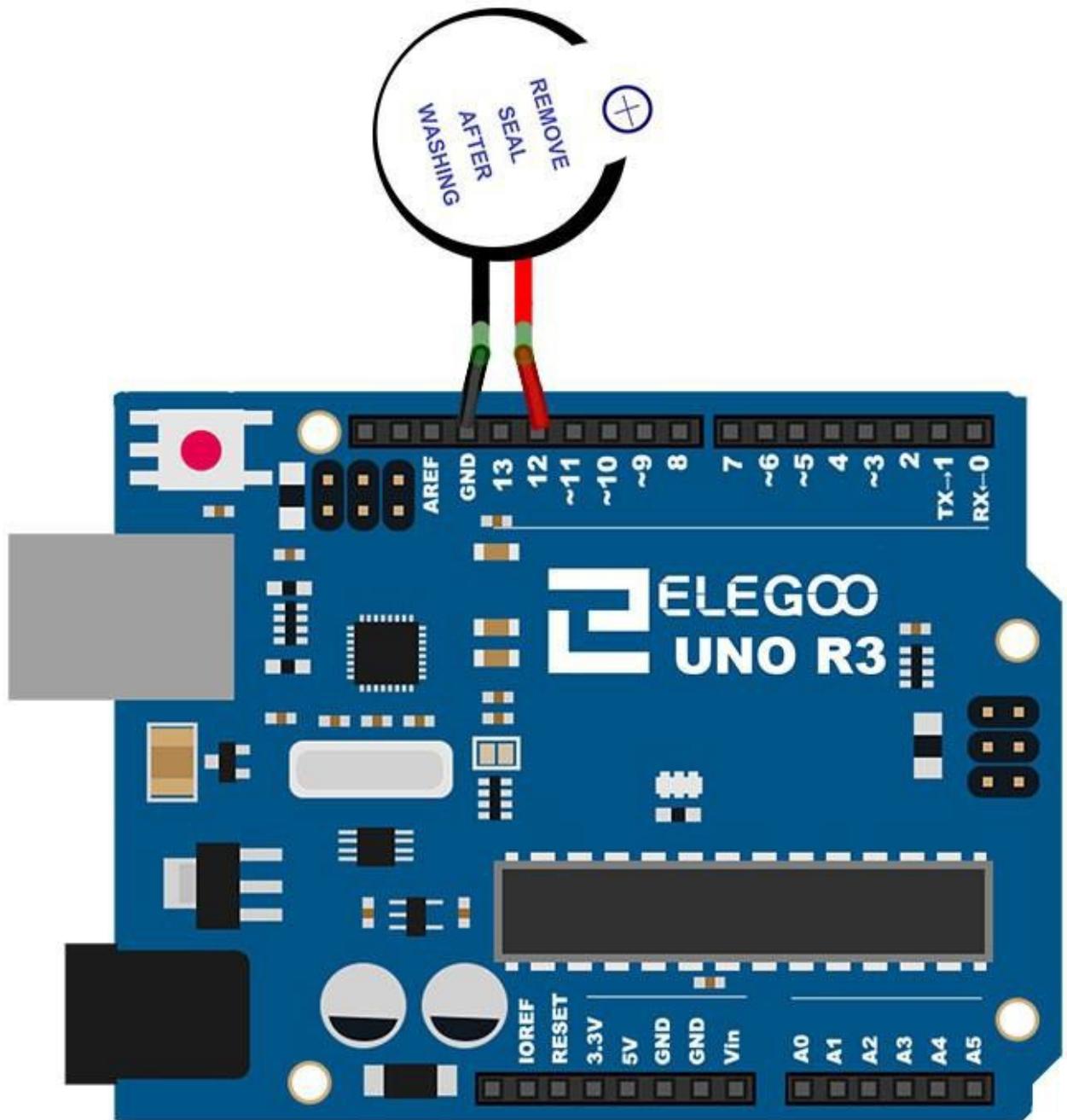
音源があるので、電化すると音が鳴ります。受動的なブザーはそのような音源を持っていないので、DC信号が使われていればつぶれません。代わりに、それを駆動するために周波数が2K~5Kの方形波を使用する必要があります。アクティブなブザーは、複数の内蔵発振器のためにパッシブブザーよりしばしば高価です。



Connection
Schematic



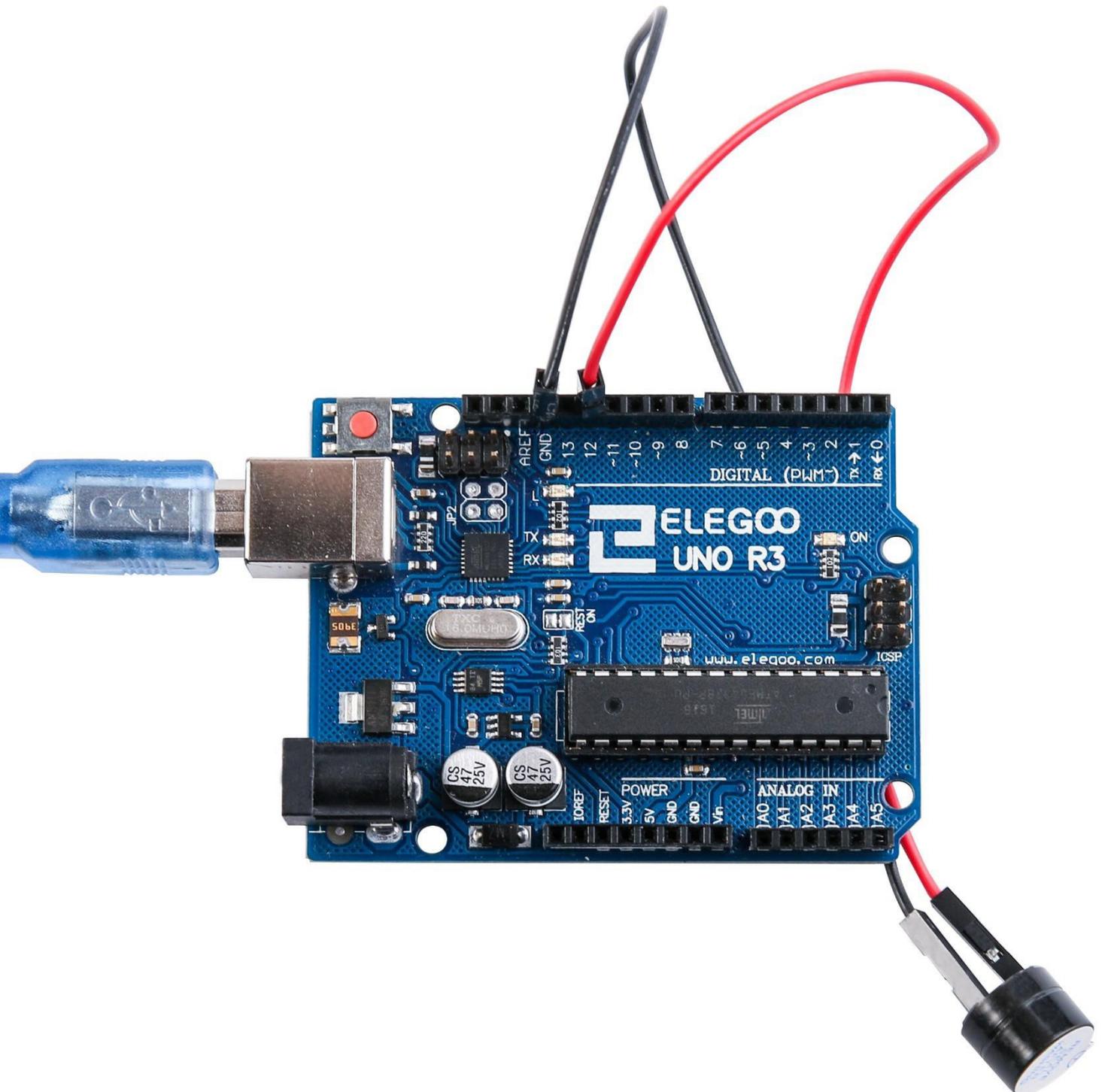
Wiring diagram



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。

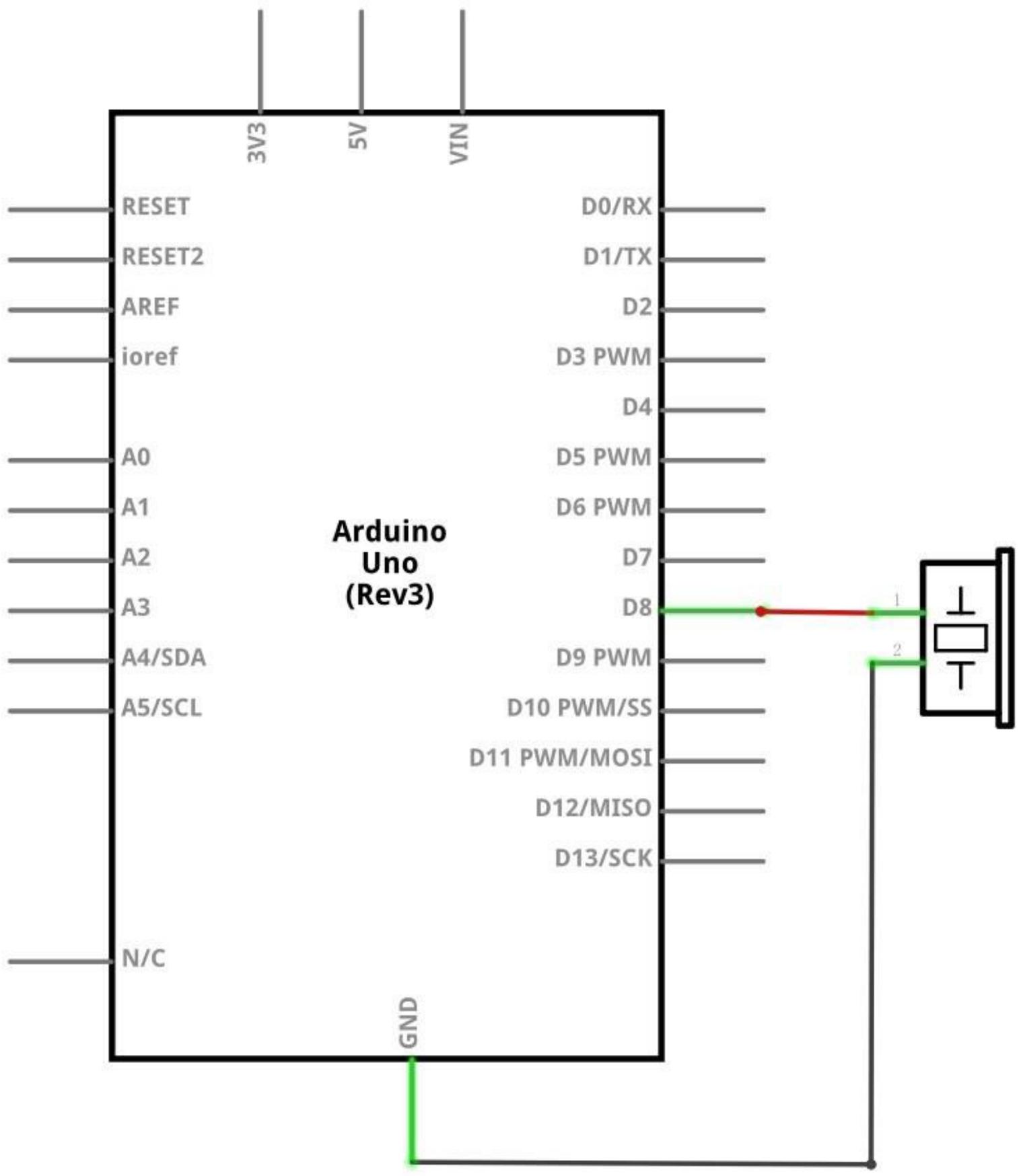
ミッドレンジミー。ブザーで曲を演奏できます。

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

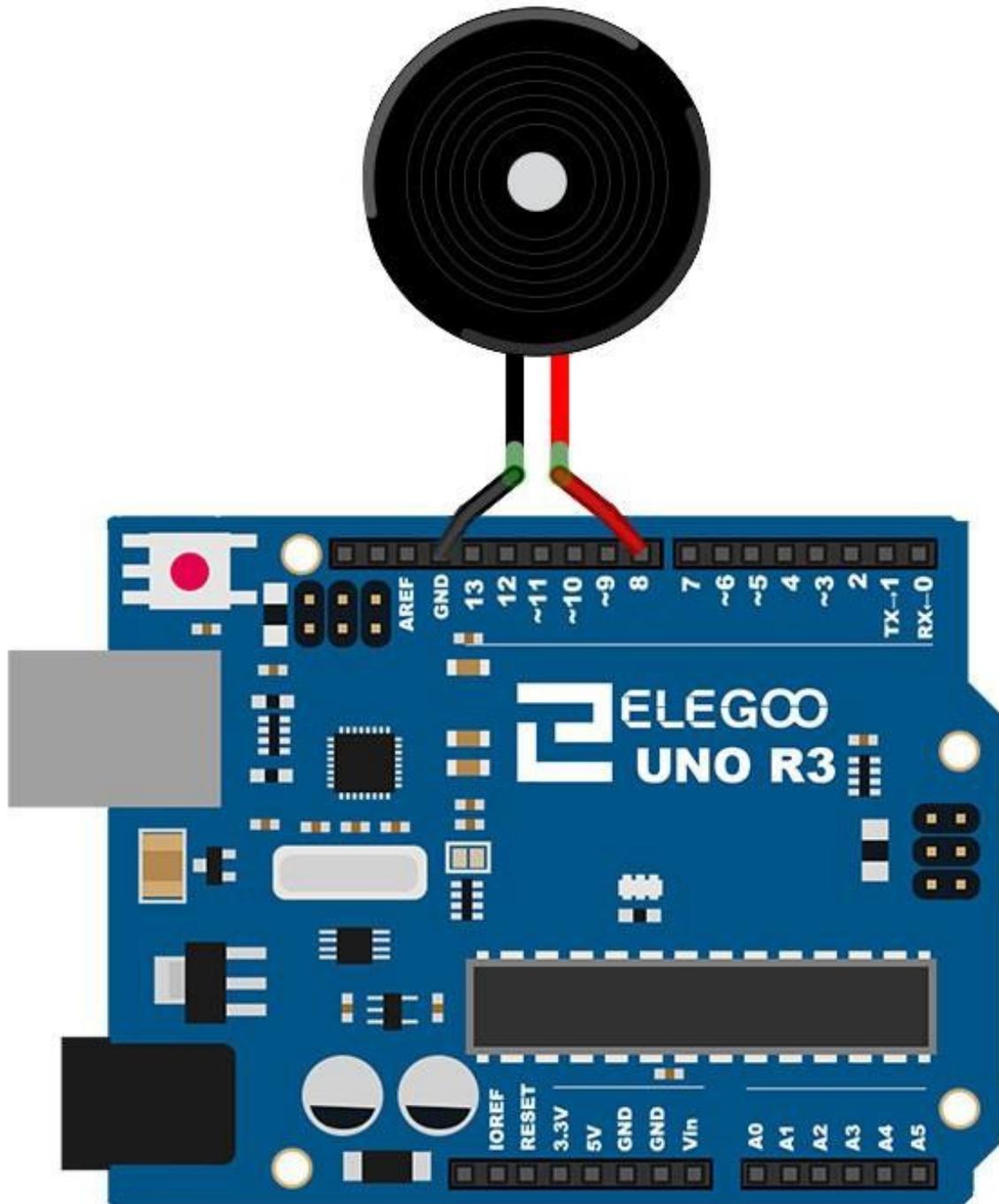
(500Hz) 。



**Connection
Schematic**



Wiring diagram



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

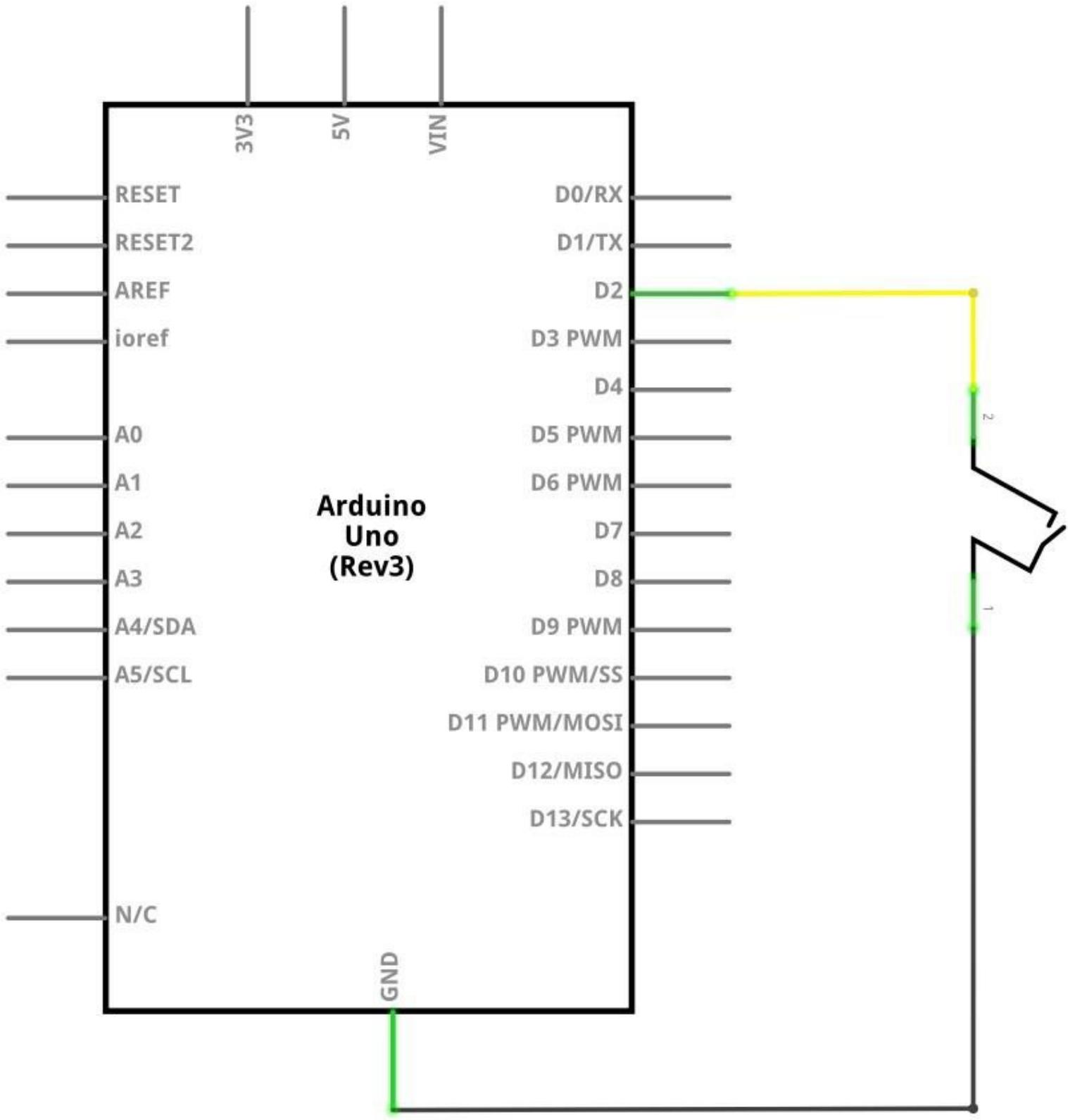
Tilt sensor:

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」と呼ばれることもあります。

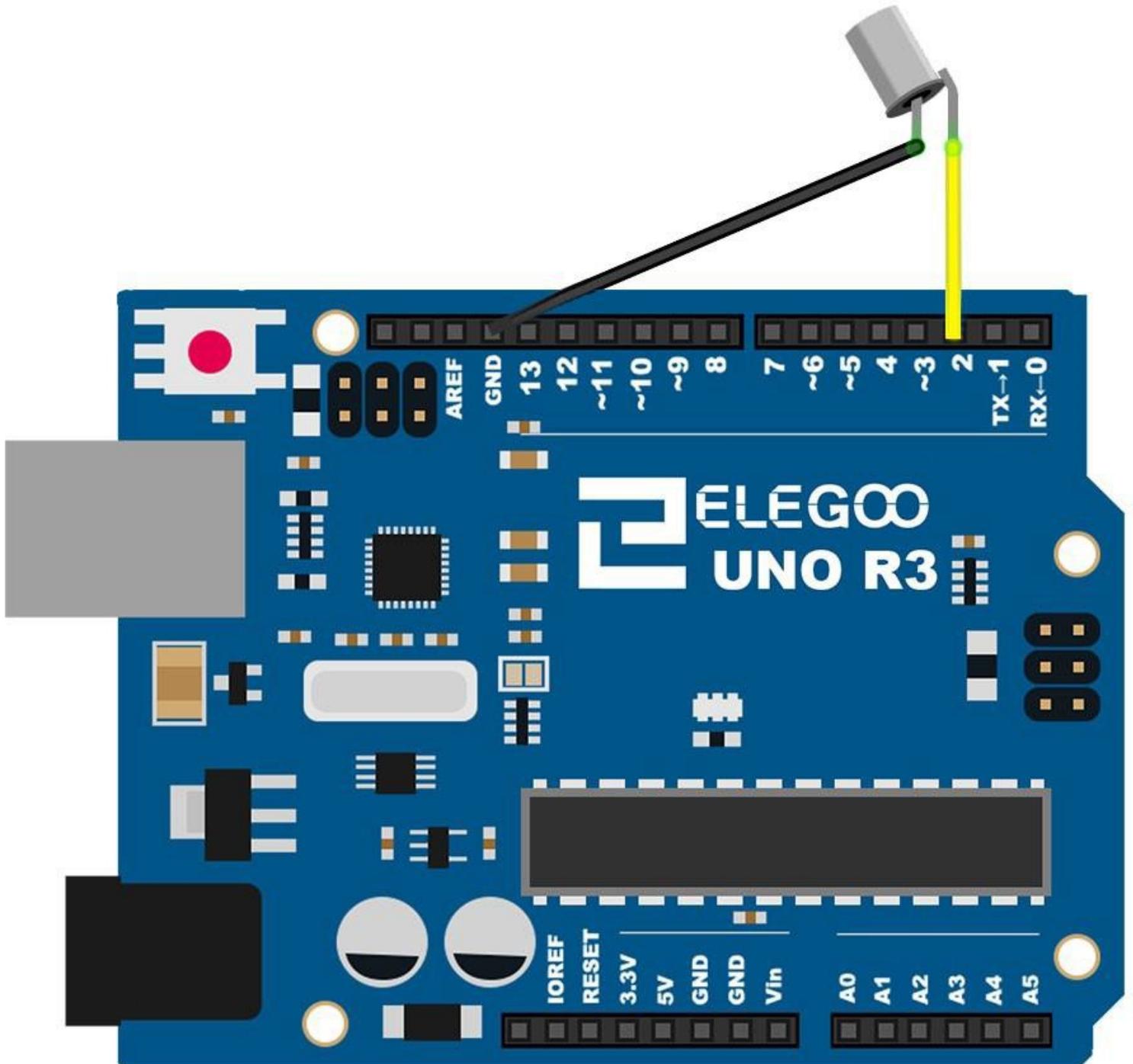
それらは通常、何らかの種類の空洞（円筒形が一般的ではありますが、必ずしも一般的ではありません）の中に、水銀や転がり球のような導電性の自由質量を内包しています。キャビティの一端は2つの導電性要素（極）を有する。センサがその端部が下方になるように配向されると、質量は極に転がり、それらを短絡させ、スイッチスローとして作用する。

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**



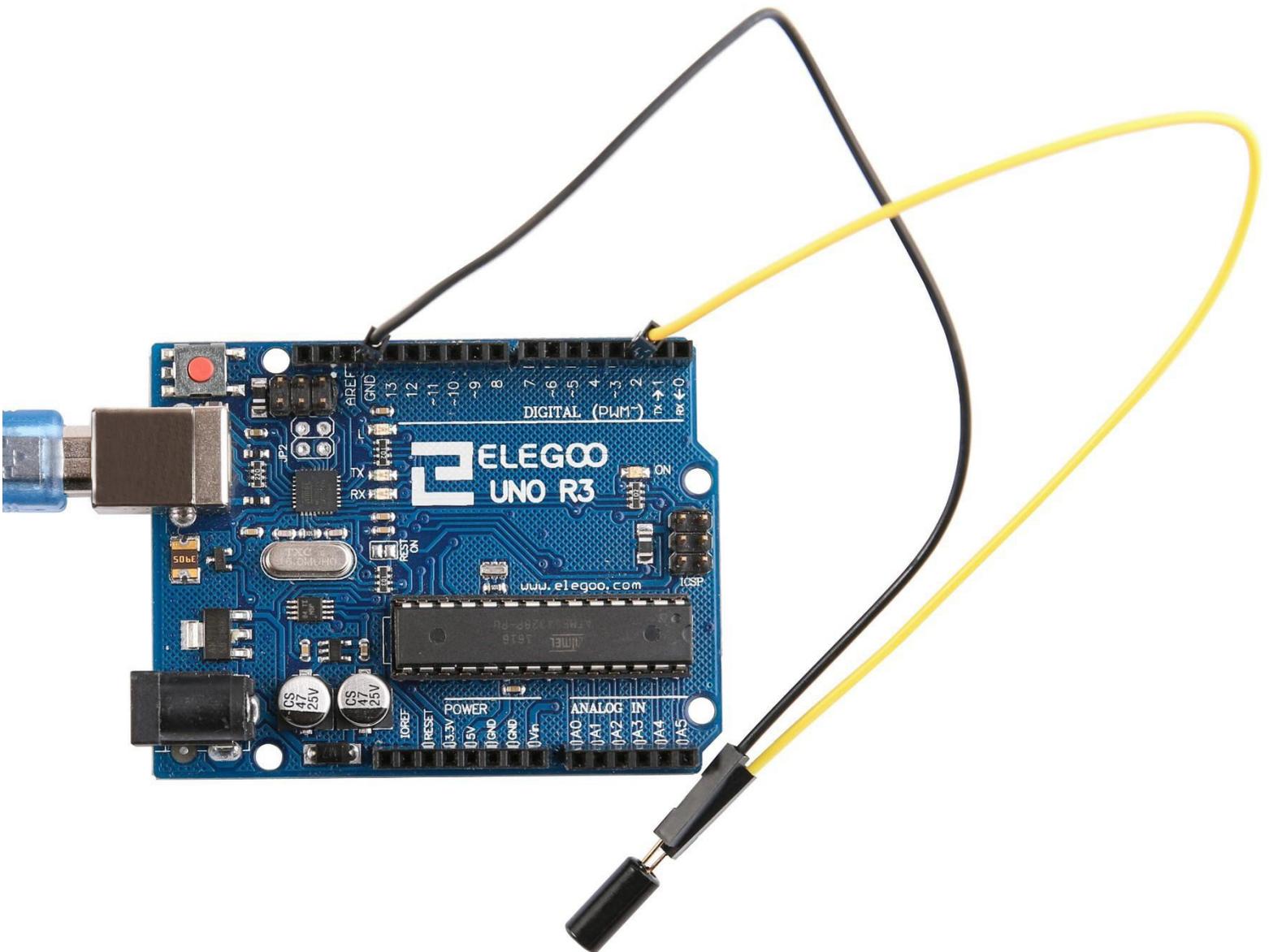
Wiring diagram



Code

配線後、コードフォルダのレッスン 8 ボ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る、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)

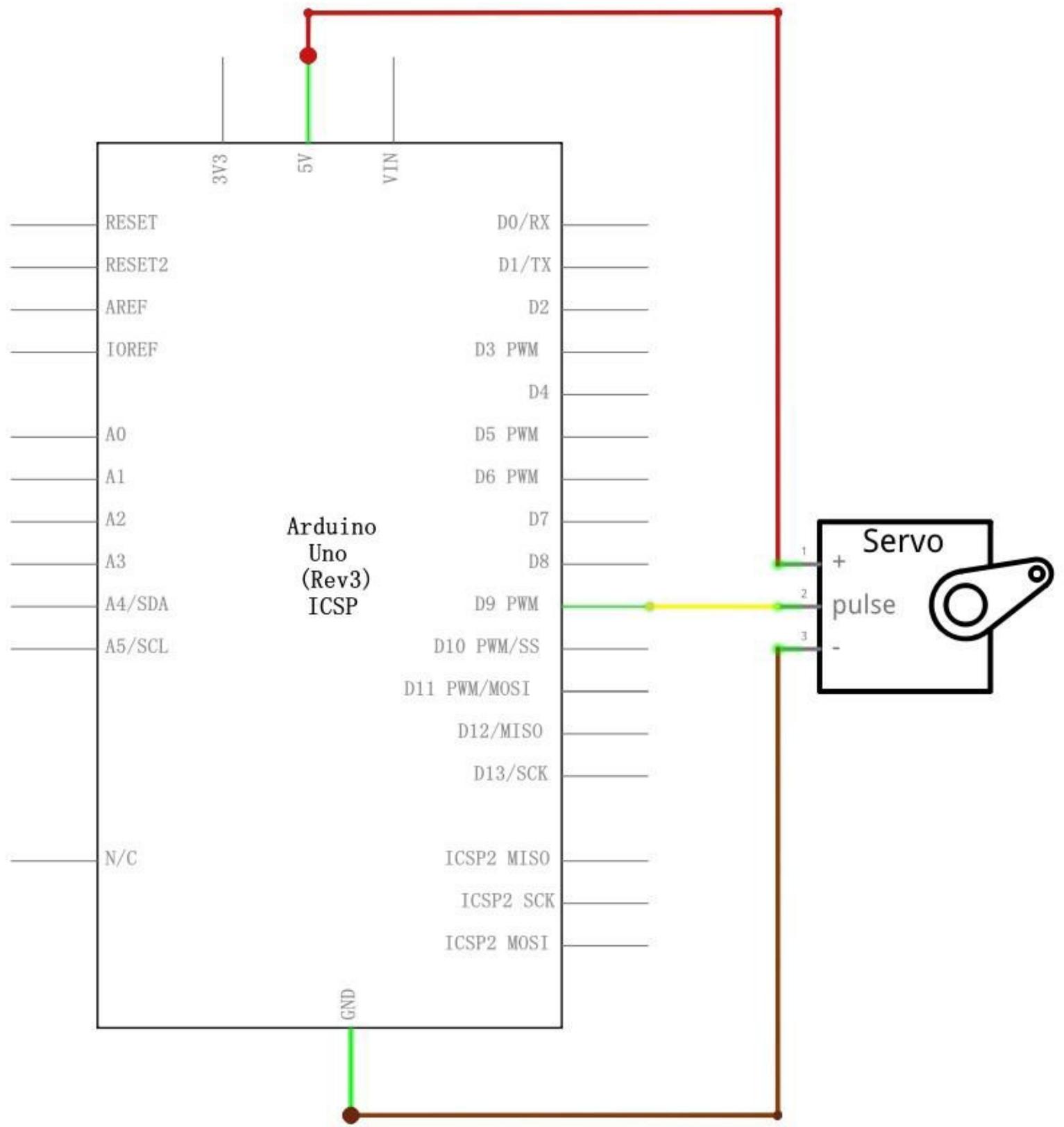
部品の紹介

SG90

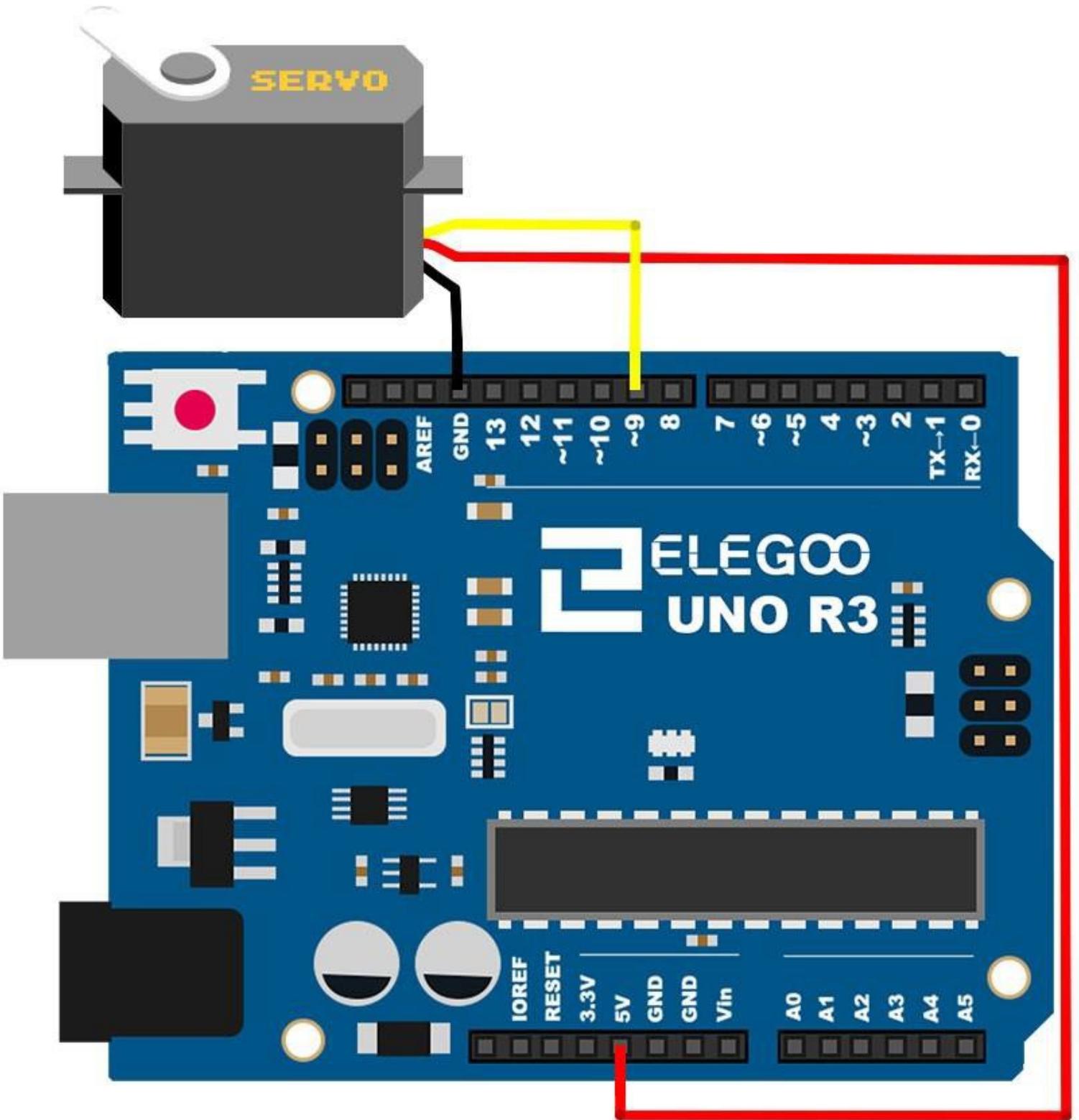
- Universal for JR and FP connector
- Cable length : 25cm
- No load; Operating speed: 0.12 sec / 60 degree (4.8V), 0.10 sec / 60 degree (6.0V)
- Stall torque (4.8V): 1.6kg/cm
- Temperature : -30~60'C
- Dead band width: 5us
- Working voltage: 3.5~6V
- Dimension : 1.26 in x 1.18 in x 0.47 in (3.2 cm x 3 cm x 1.2 cm)
- Weight : 4.73 oz (134 g)



Connection Schematic



Wiring diagram



Code

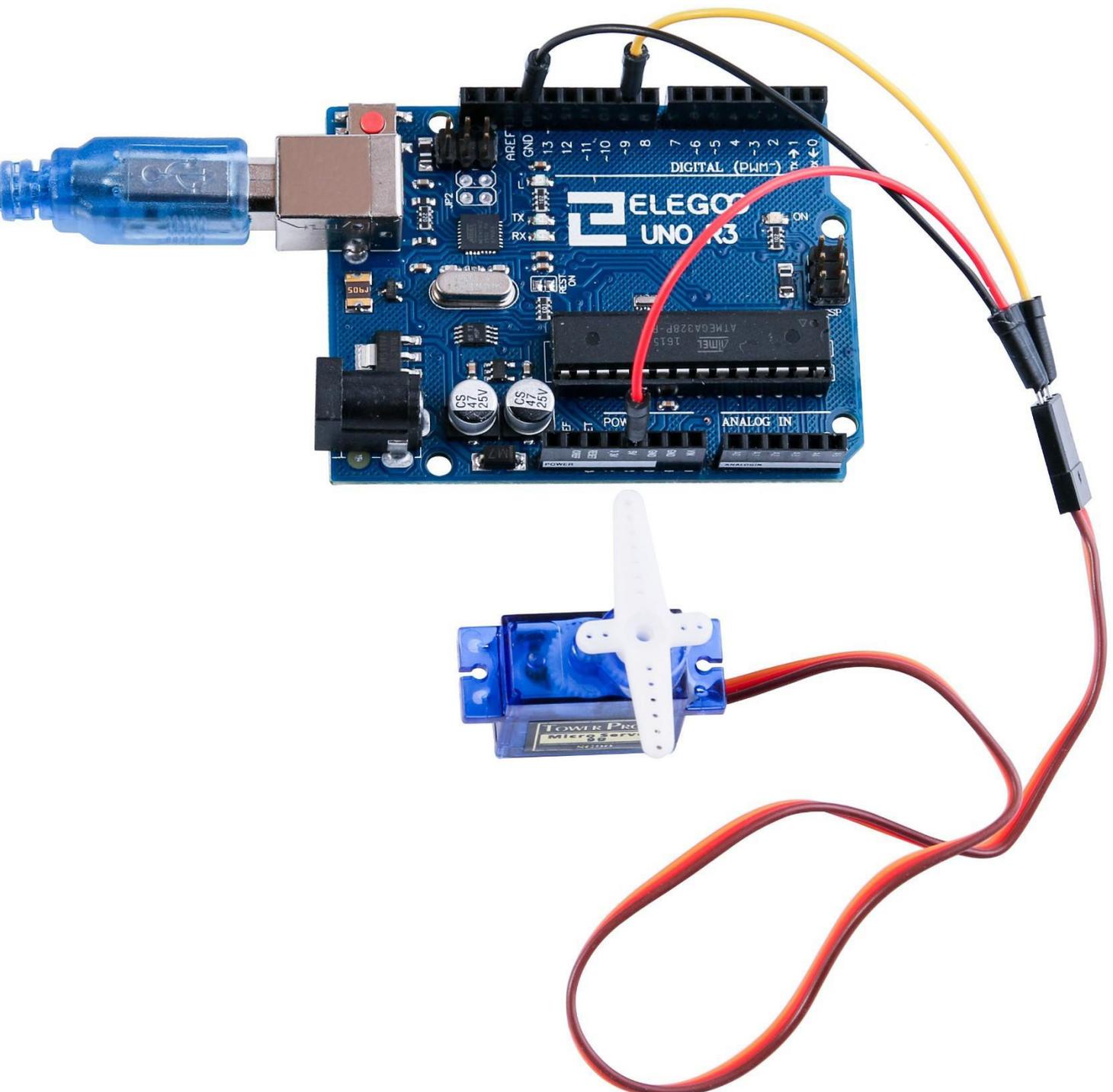
配線後、コードフォルダの Lesson 9 Servo でプログラムを開き、UPLOAD をクリックしてプログラムをアップロードしてください。エラーがある場合のプログラムアップロードの詳細については、レッスン 2 を参照してください。

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 線を介して接続され、



Lesson 10 超音波センサモジュール

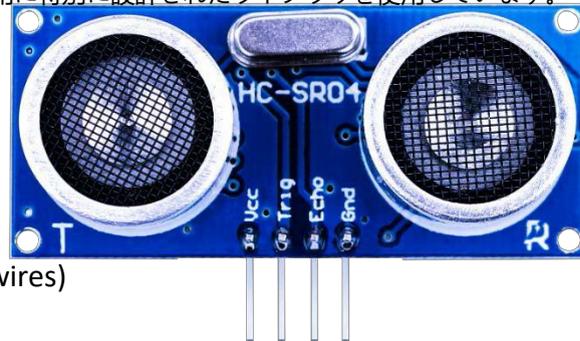
概要

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 DuPont wires)



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:

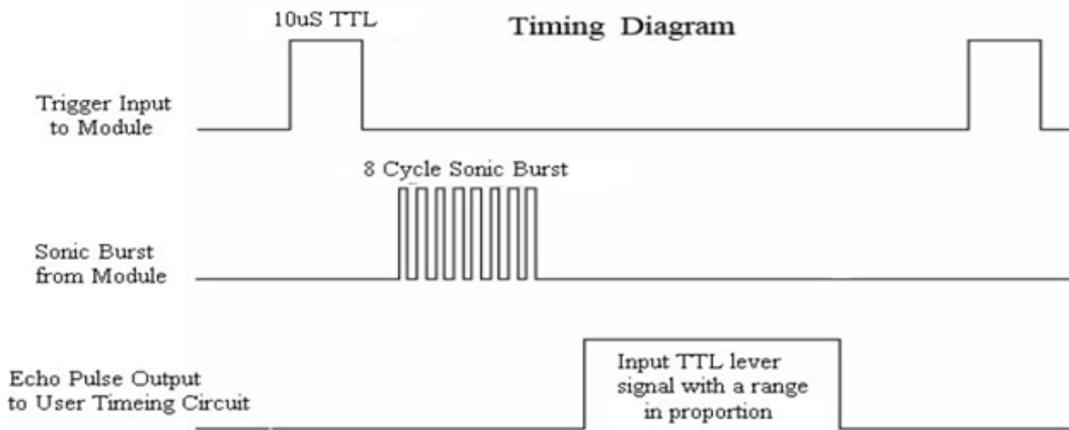
- (1) IO トリガを 10us 以上のハイレベル信号に使用すると、
- (2) モジュールは 40 kHz を自動的に 8 回送信し、パルス信号が戻ってきたかどうかを検出します。
- (3) 信号が戻ってくると、ハイレベルを通して、高出力 IO 持続時間は、超音波を送信してから戻るまでの時間である。

試験距離 = (高レベル時間 × 音速 (340m / 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 サイクルの超音波パーストを送り、エコーを上げます。

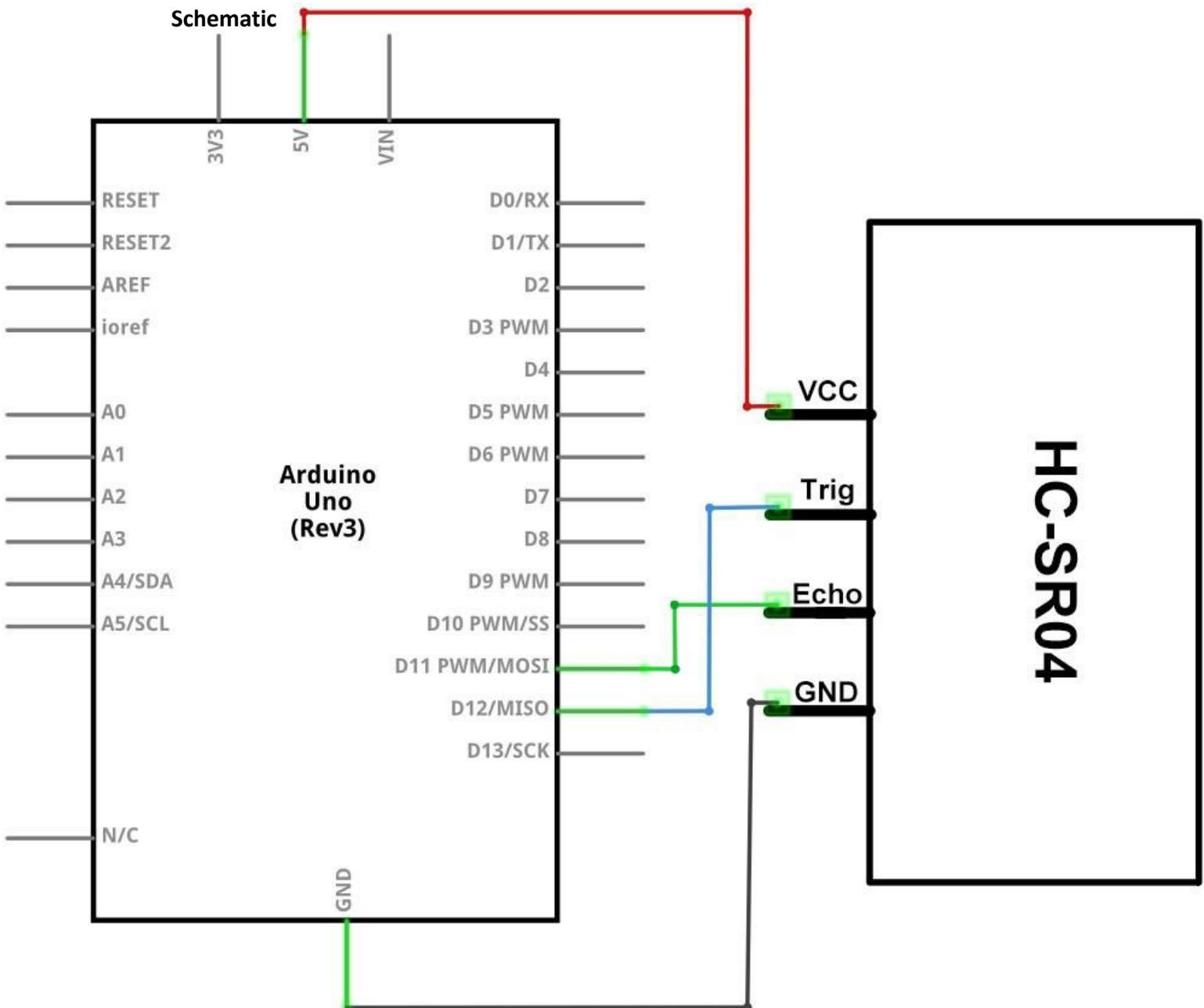
エコーは、パルス幅と距離範囲の距離オブジェクトです。トリガー信号を送信してからエコー信号を受信するまでの時間間隔で範囲を計算できます。

数式: $us / 58 = \text{センチメートル}$ または $us / 148 = \text{インチ}$; または: $\text{範囲} = \text{高レベル時間} * \text{速度} (340M / S) / 2$; 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。

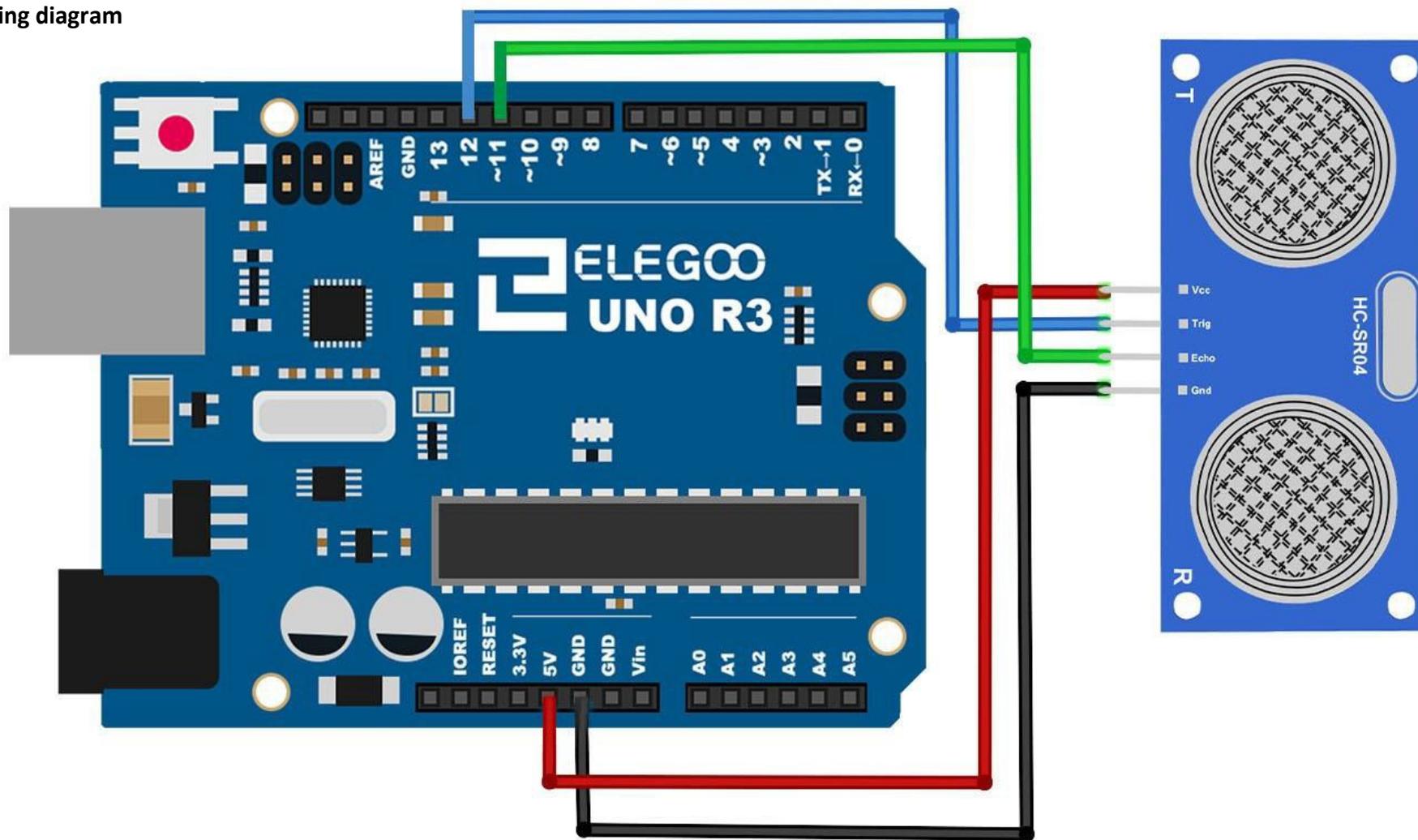


Connection

Schematic



Wiring diagram



Code

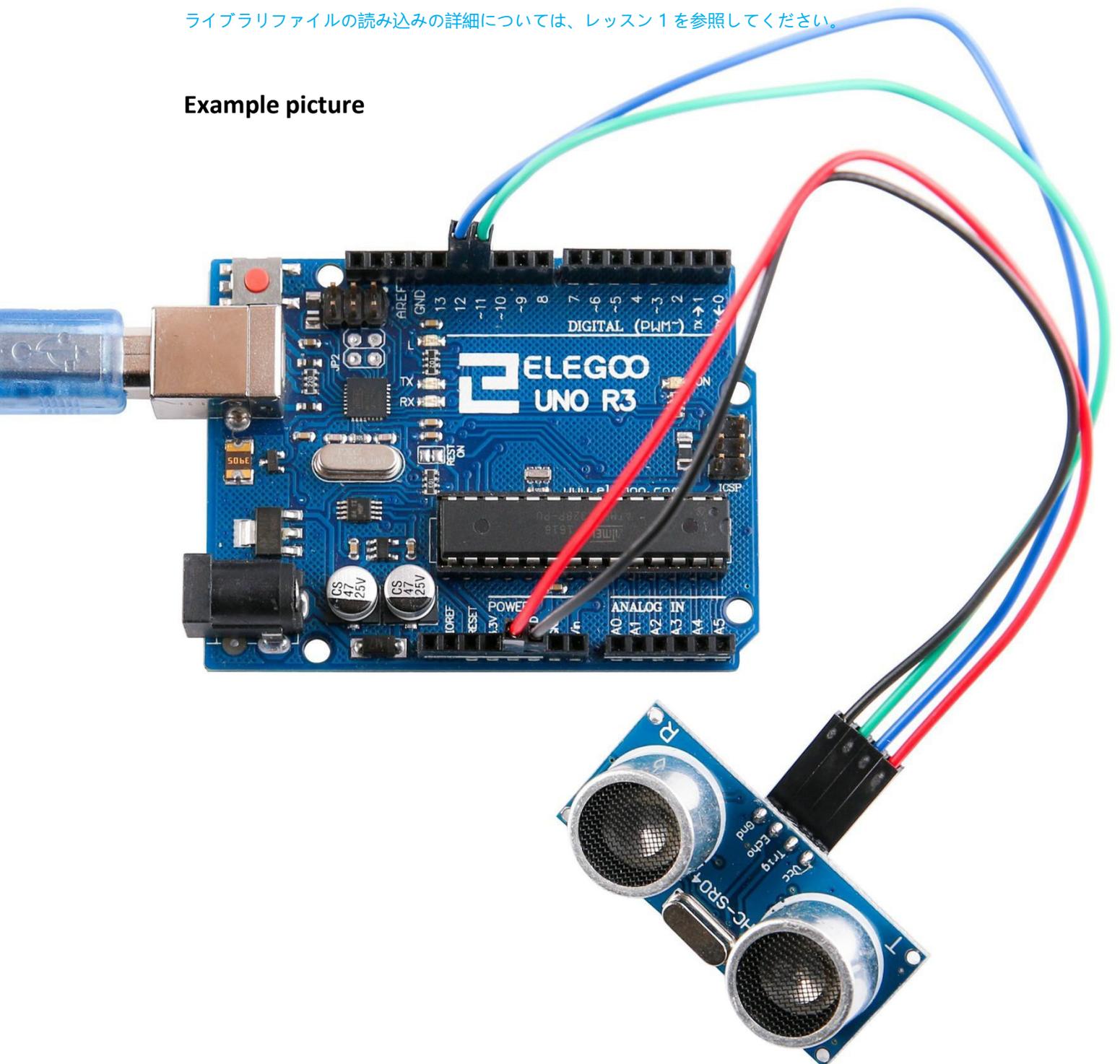
これらのセンサー用に設計されたライブラリを使用すると、コードが短く簡単になります。コードの始めにライブラリを組み込み、簡単なコマンドを使用してセンサの動作を制御することができます。

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、<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。

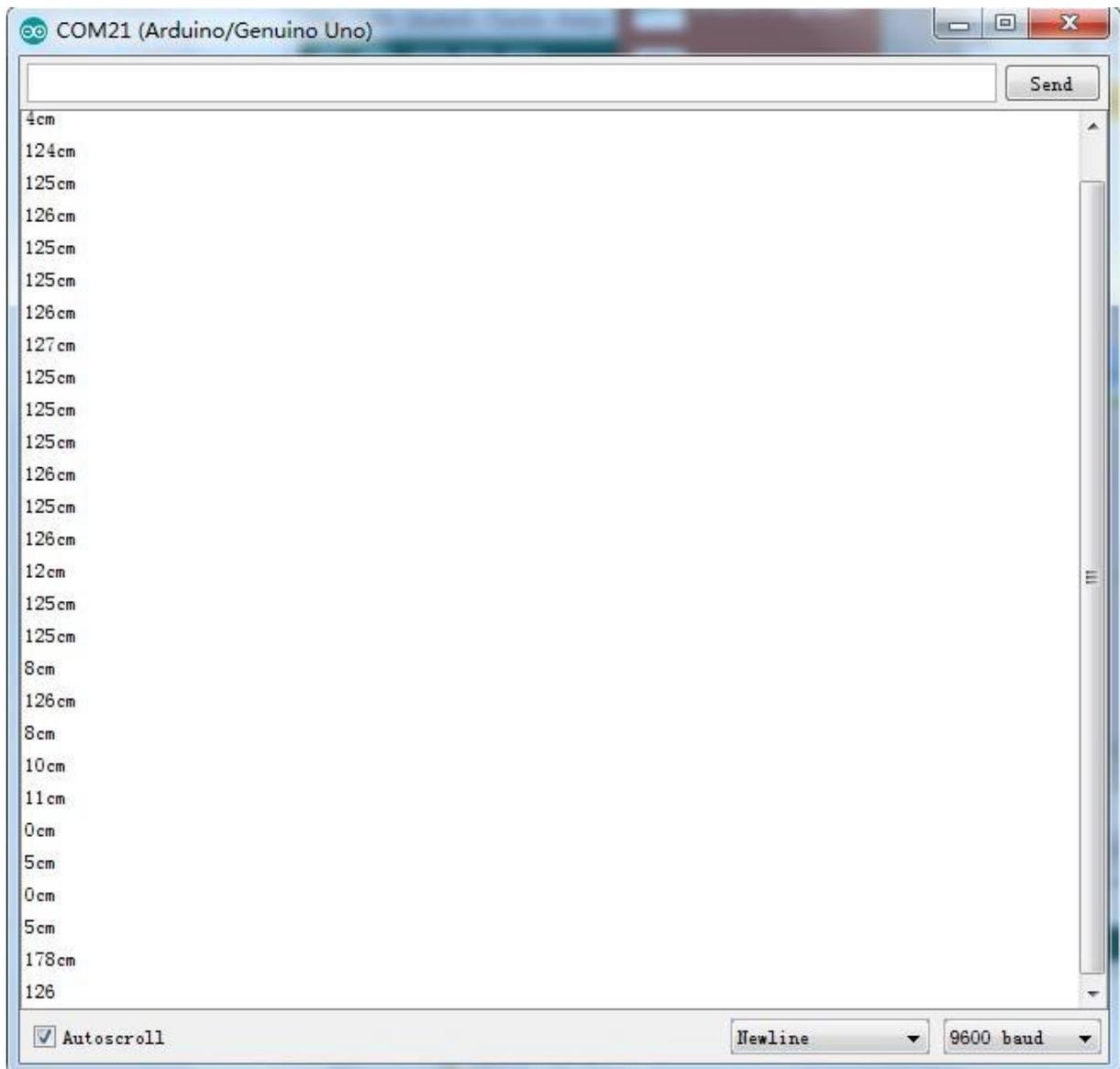
ライブラリファイルの読み込みの詳細については、[レッスン 1](#) を参照してください。

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 膜スイッチモジュール

概要

このプロジェクトでは、キーボードを UNO R3 ボードに統合し、UNO R3 がユーザーによって押されているキーを読み取る方法について説明します。

キーパッドは、携帯電話、ファクシミリ、電子レンジ、オープン、ドアロックなど、あらゆる種類のデバイスで使用されています。数多くの電子機器がそれらを使用してユーザ入力を行います。

キーパッドを UNO R3 ボードなどのマイクロコントローラに接続する方法を知っていることは、さまざまな種類の商用製品を構築する上で非常に貴重です。

最後にすべてが正しく接続され、プログラムされているときに、キーを押すと、コンピュータのシリアルモニタに表示されます。キーを押すたびに、シリアルモニタに表示されます。簡略化のために、単にコンピュータに押されたキーを表示するところから始める。

このプロジェクトでは、使用するキーパッドのタイプはマトリックスキーパッドです。これは、キーよりもずっと少ない出力ピンを持つことができる符号化方式に従うキーパッドです。たとえば、使用しているマトリックスキーパッドには 16 個のキー (0-9、A-D、*、#) がありますが、8 個の出力ピンしかありません。リニア・キーパッドの場合、動作させるために 17 本の出力ピン (各キーとグラウンド・ピンに 1 本) が必要です。

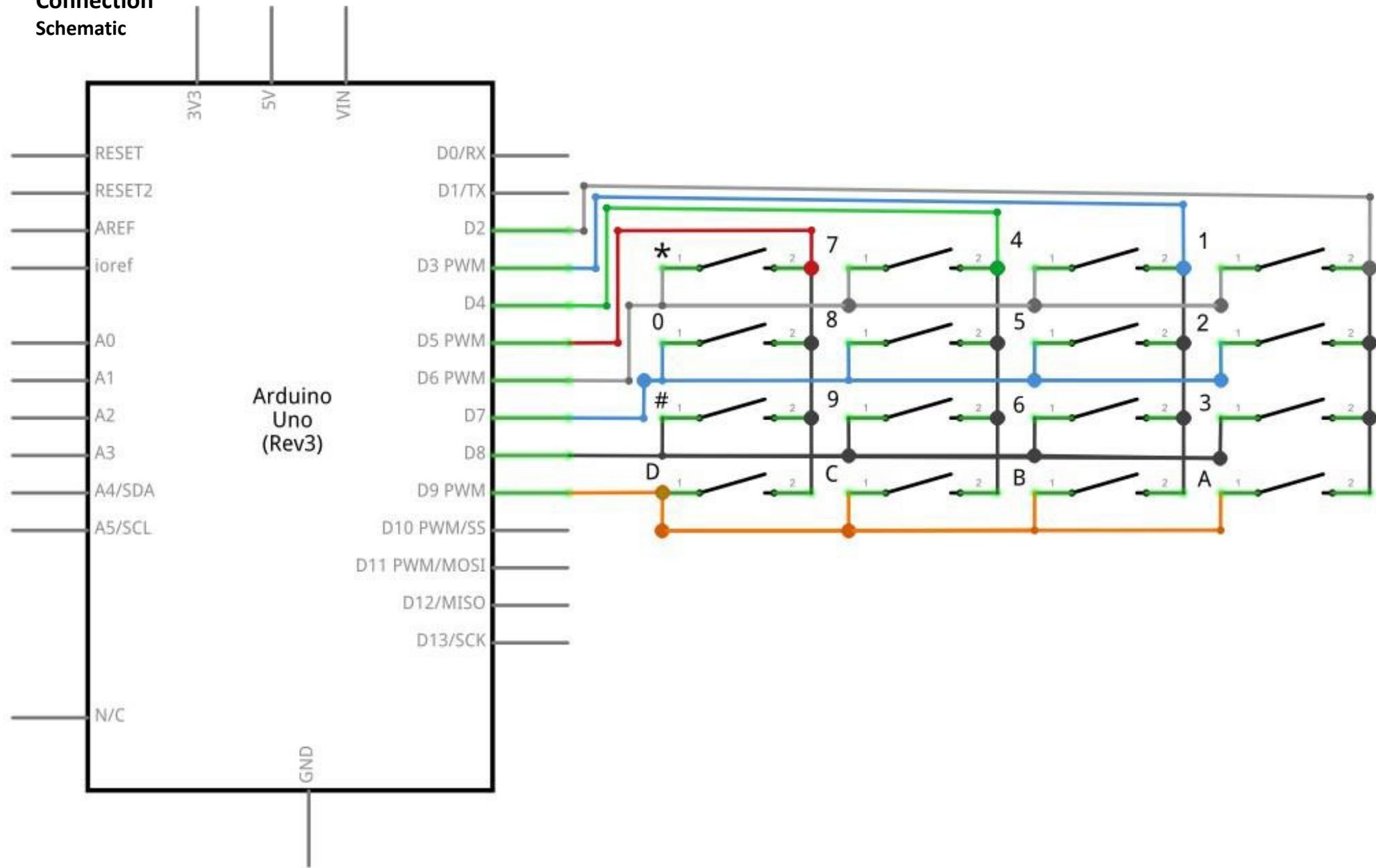
マトリクス符号化方式は、出力ピンを少なくすることができ、したがって、キーパッドが機能するために必要な接続を大幅に少なくすることができる。このようにして、線形キーパッドよりも効率的です。配線が少なくて済みます。

必要な構成部品:

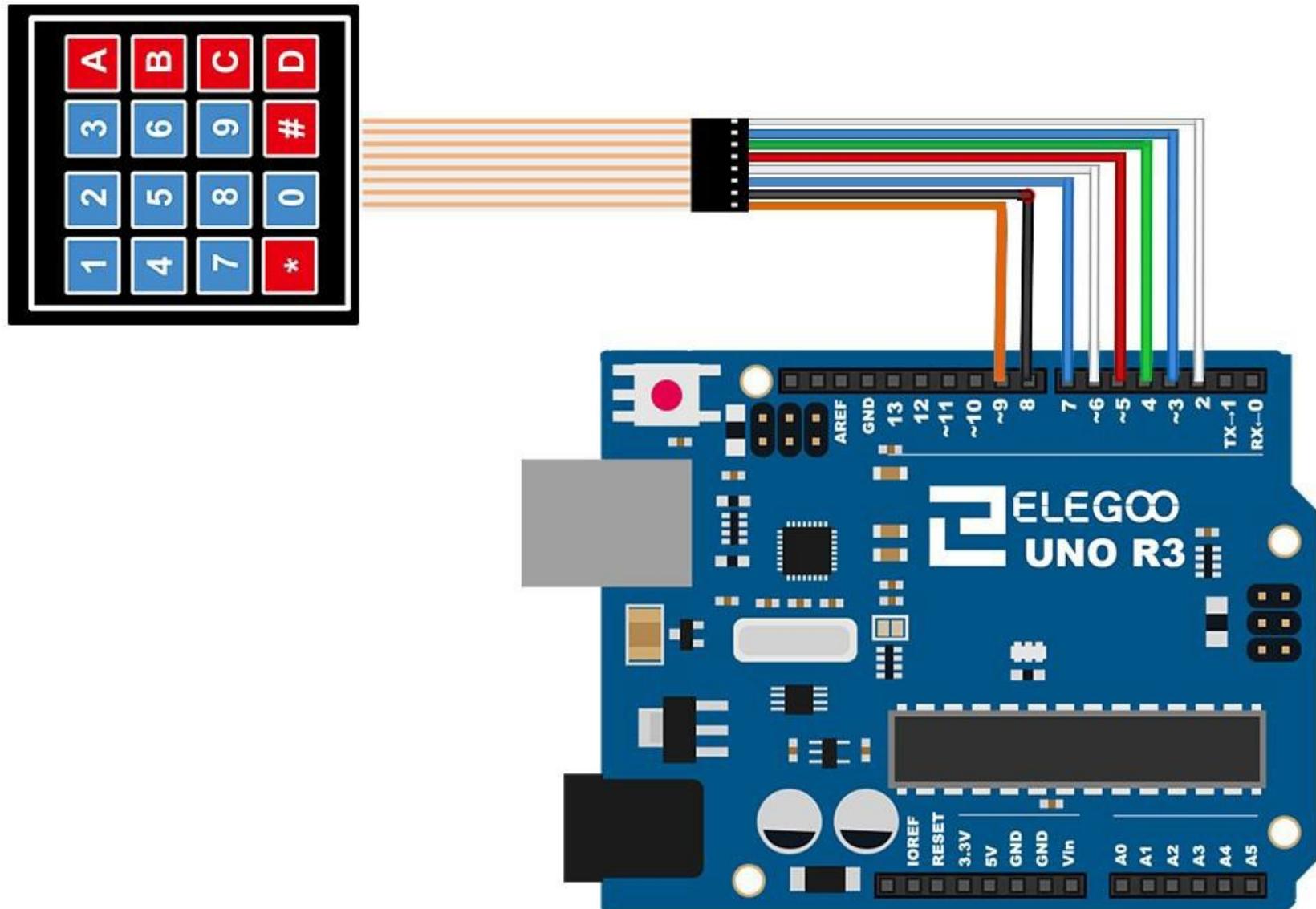
- (1) x Elegoo Uno R3
- (1) x Membrane switch module
- (8) x M-M wires (Male to Male jumper wires)



Connection Schematic



Wiring diagram



UNO R3 ボードにピンを接続するときは、それらをデジタル出力ピン D9-D2 に接続します。キーボードの最初のピンを D9 に、2 番目のピンを D8 に、3 番目のピンを D7 に、4 番目のピンを D6 に、5 番目のピンを D5 に、6 番目のピンを D4 に、7 番目のピンを D3 に、8 番目のピンを D2 に接続します。ピンから D2 へ。

これらはテーブル内の接続です:

Keypad Pin	Connects to Arduino Pin...
1	D9
2	D8
3	D7
4	D6
5	D5
6	D4
7	D3
8	D2

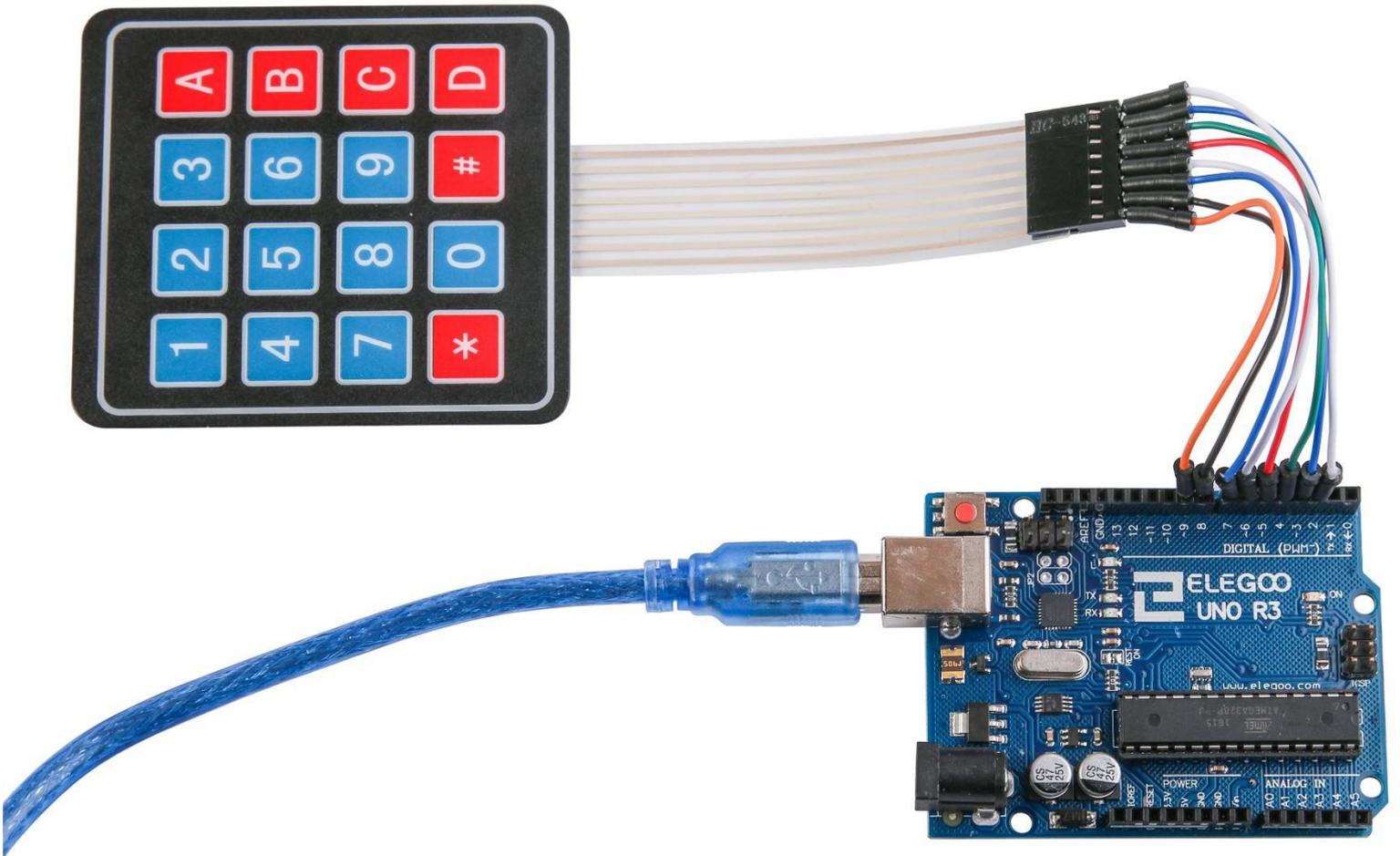
Code

結線後、プログラムをコードフォルダ - レッスン 11 メンブレンスイッチモジュールで開き、**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、<Keypad>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。

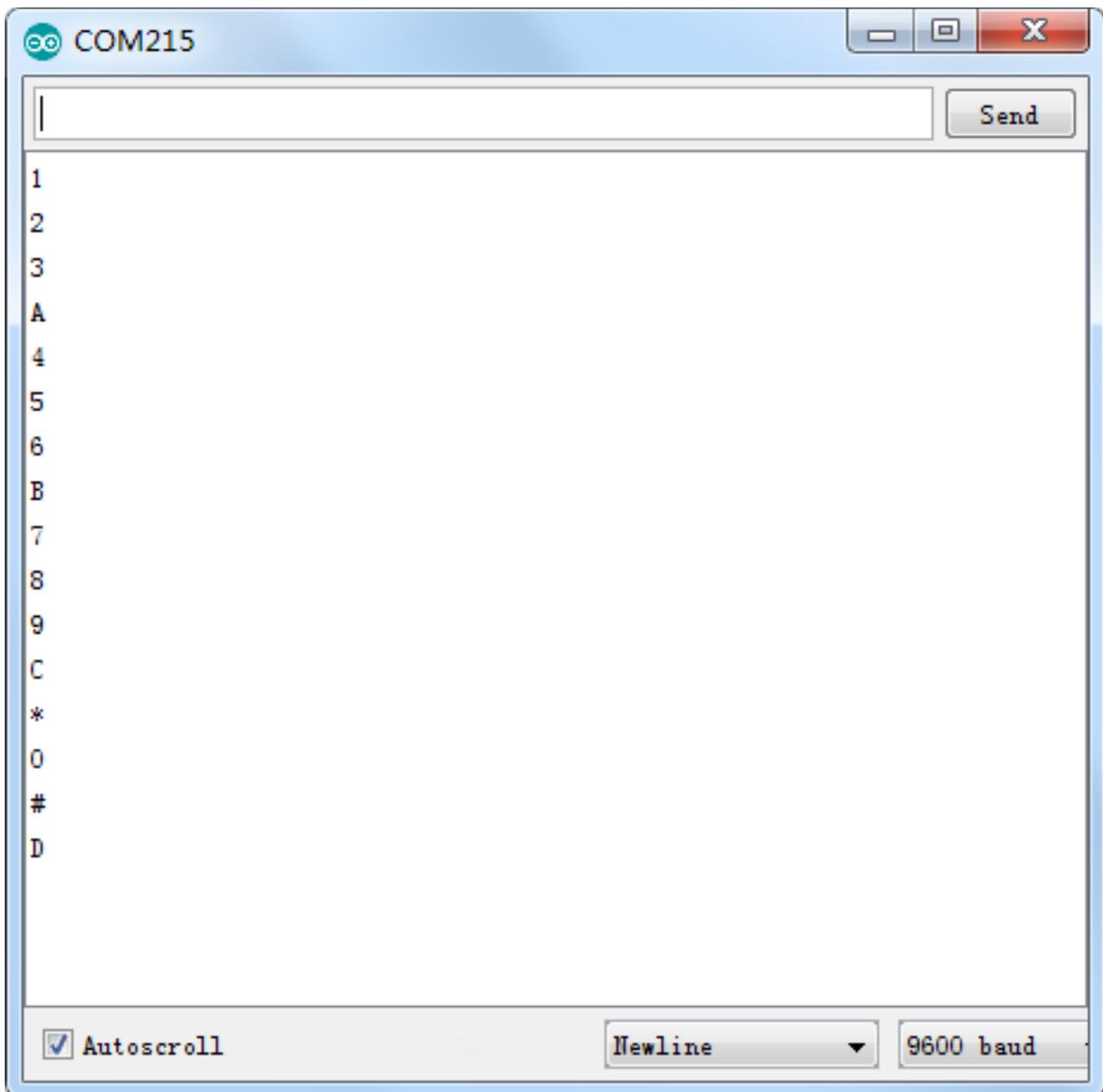
ライブラリファイルのロードの詳細については、[レッスン 1](#) を参照してください。

Example picture



このコードでは、キーパッドのキーを押すと、コードがコンパイルされて UNO R3 ボードにアップロードされると、Arduino ソフトウェアのシリアルモニタに表示されます。

[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 12 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。

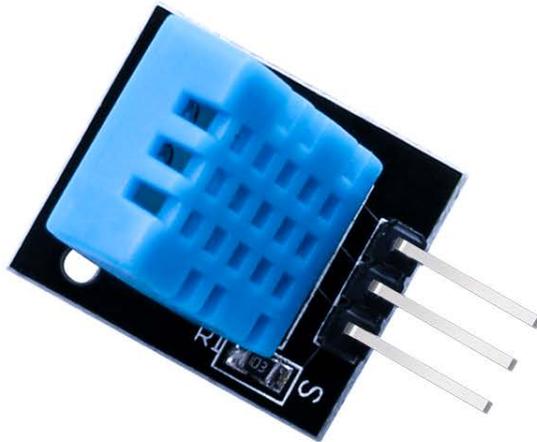
繰り返しますが、これらのセンサー専用設計されたライブラリを使用して、コードを短く簡単に作成します。

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 DuPont wires)

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: DC 3.5 ~ 5.5V

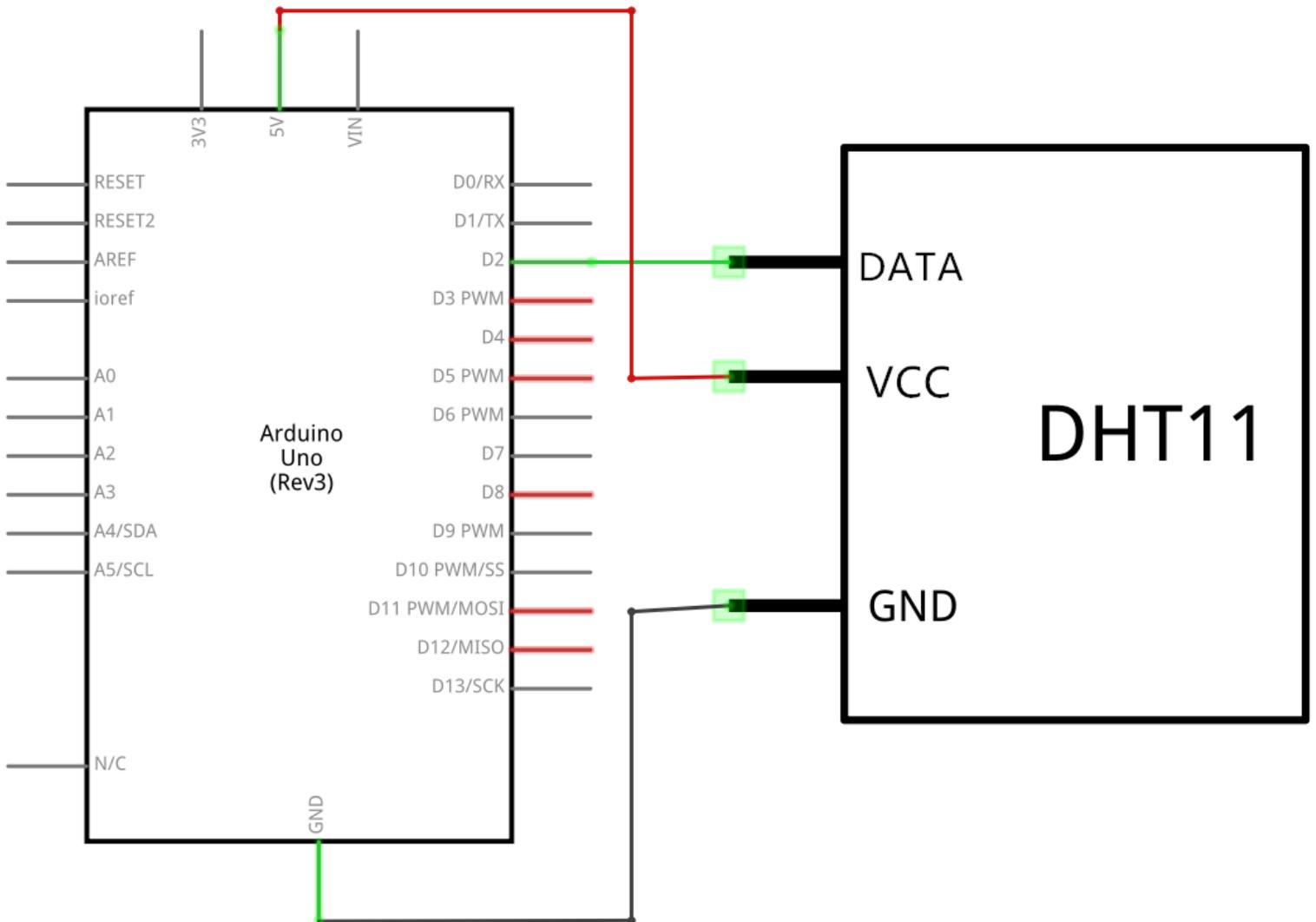
Supply Current: measurement 0.3mA standby 60 μ A

Sampling period: more than 2 seconds

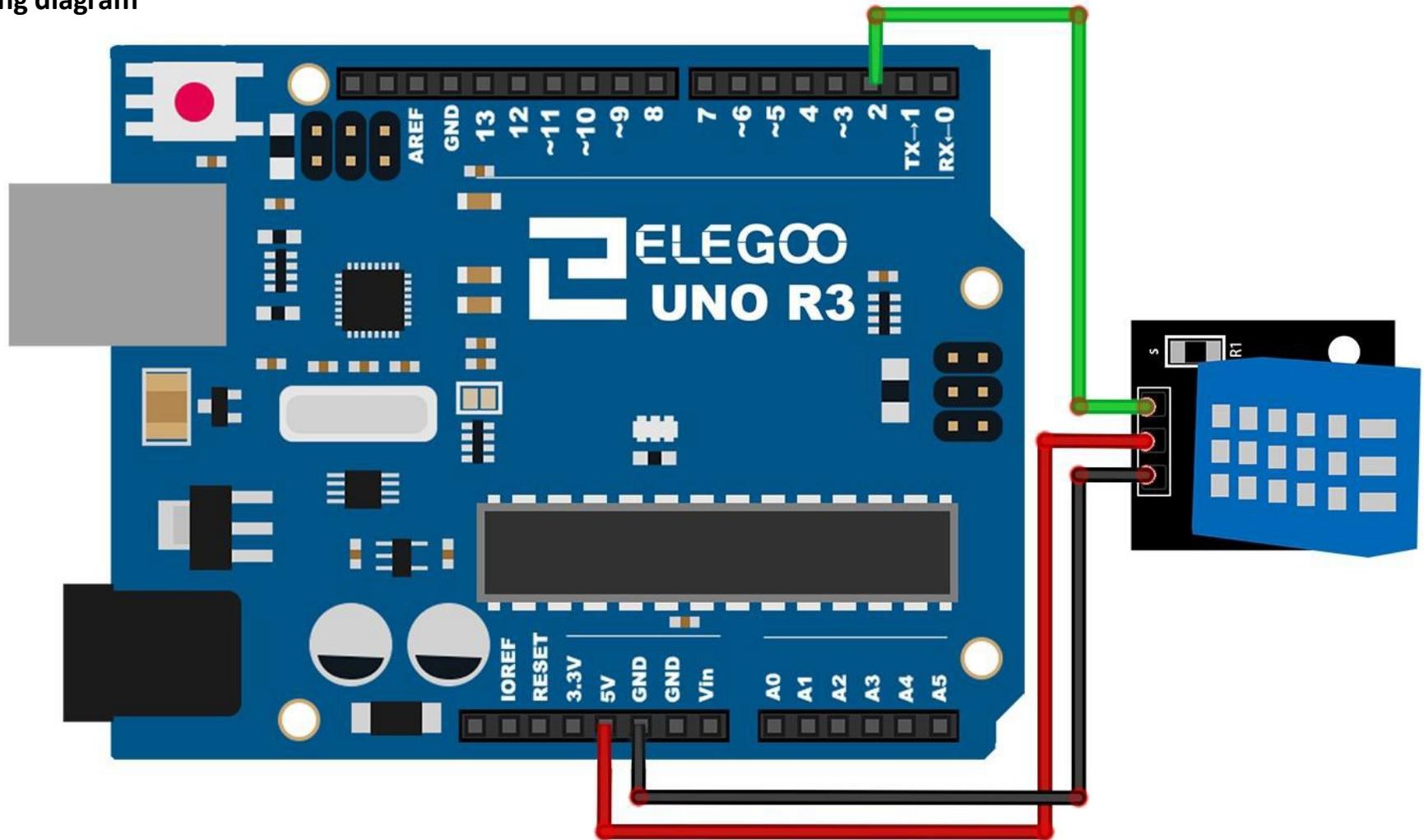
Pin Description:

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



見てのように、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。

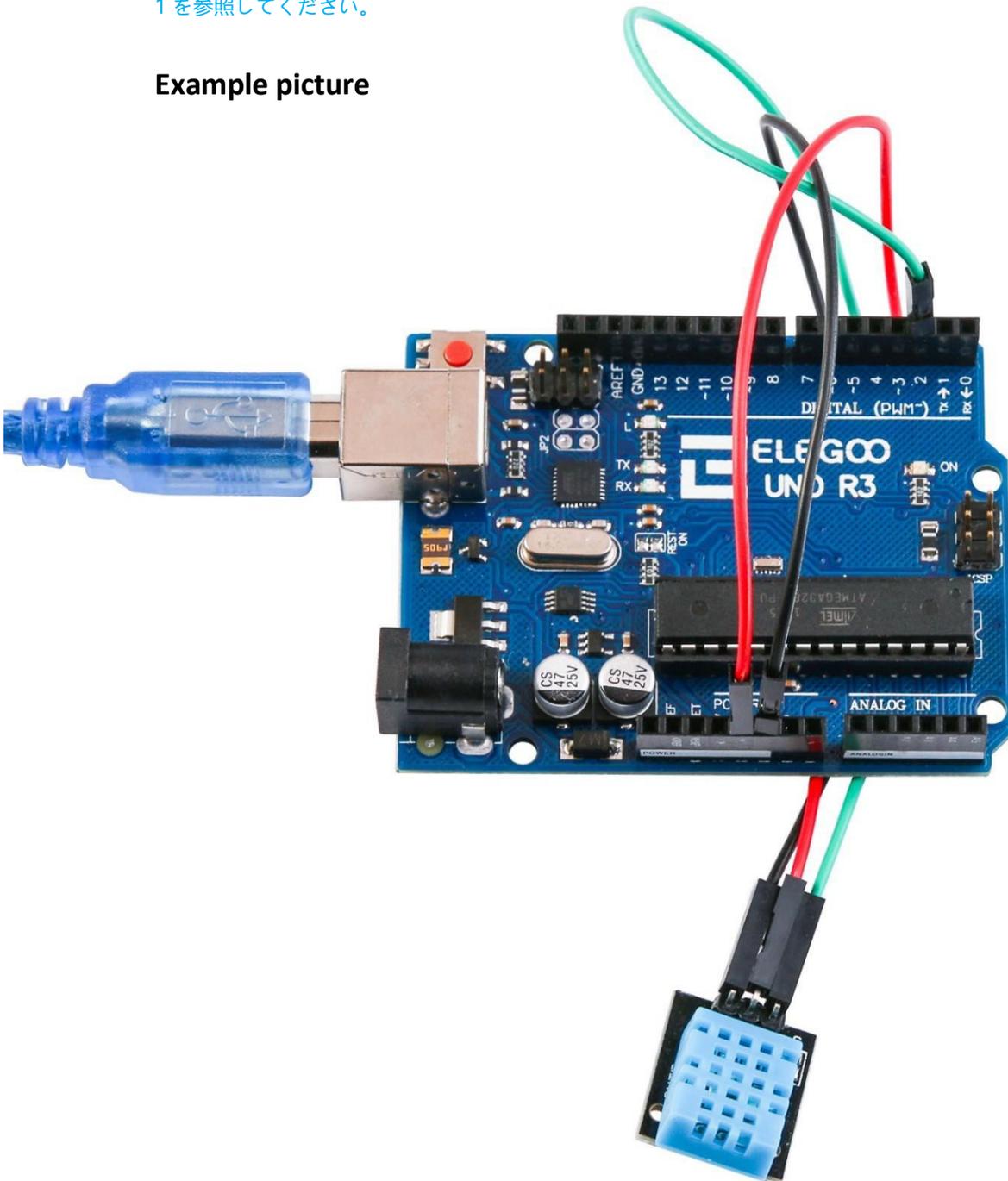
Code

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。

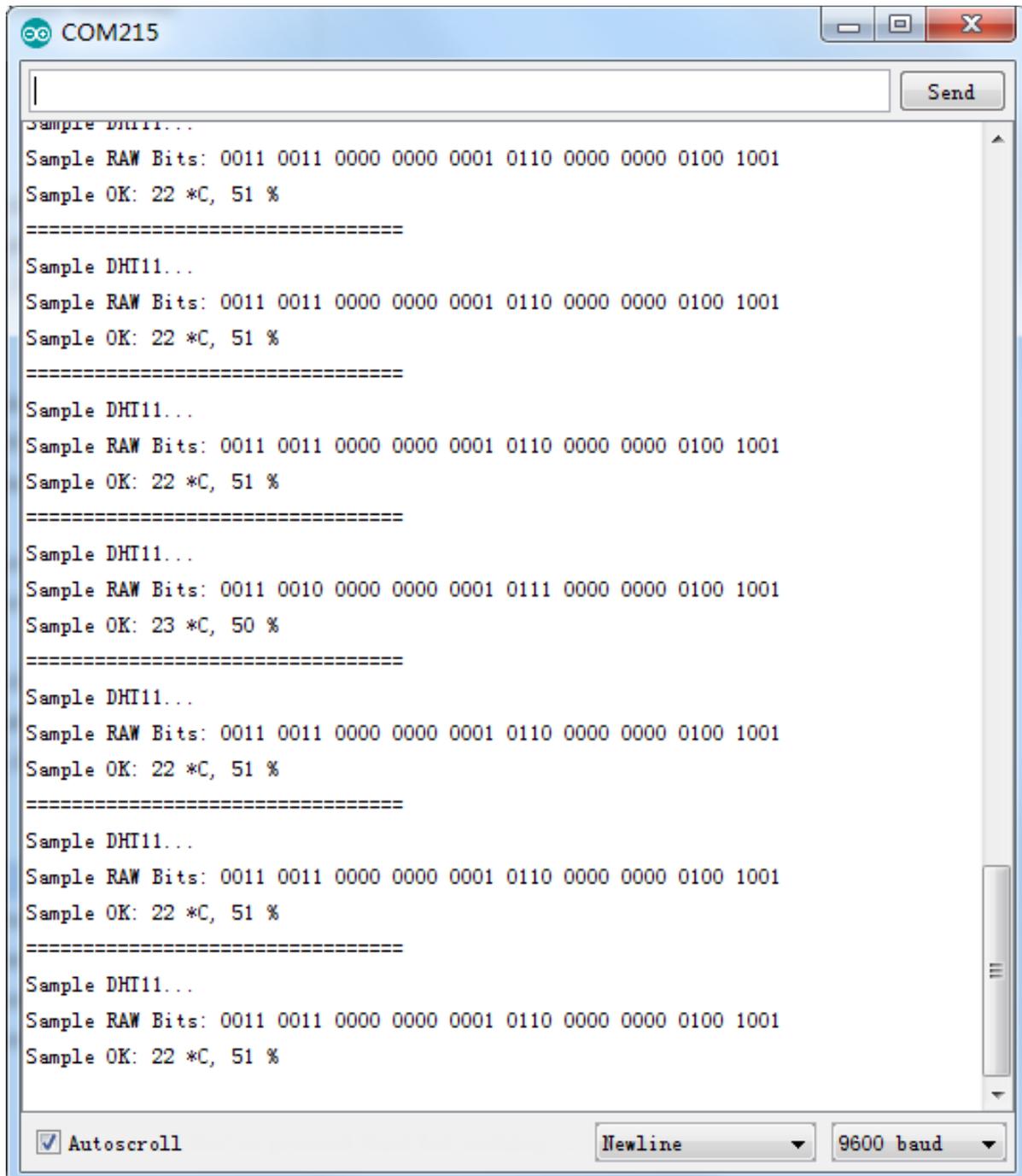
ライブラリファイルの読み込みに関するチュートリアルの詳細については、レッスン 1 を参照してください。

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)

[シリアルモニター]ボタンをクリックしてシリアルモニターをオンにします。シリア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。



The screenshot shows a window titled "COM215" with a "Send" button at the top right. The main area contains a stream of text representing sensor data. Each data point consists of a header, raw bits, and a status line, separated by horizontal lines. The status line indicates temperature and humidity. At the bottom, there are controls for "Autoscroll" (checked), "Newline" (dropdown), and "9600 baud" (dropdown).

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

Lesson 13 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 DuPont wires)

部品の紹介

Joystick

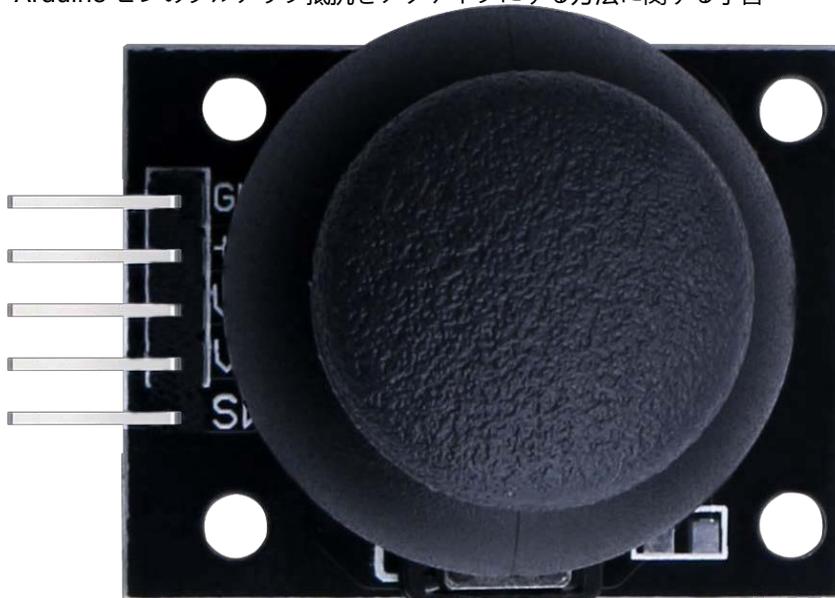
モジュールには VCC、グランド、X、Y、キーの 5 つのピンがあります。モジュールを入手した場所によって、あなたのラベルが若干異なる場合があります。親指の棒はアナログであり、単純な「指向的な」ジョイスティックよりも正確な読み取りを提供する必要があります。さらに、ジョイスティックを押下げてプッシュボタンをアクティブにすることができます。

X / Y ピンからデータを読み出すにはアナログの Arduino ピンを使用し、ボタンを読むにはデジタルピンを使用する必要があります。Key ピンは、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。

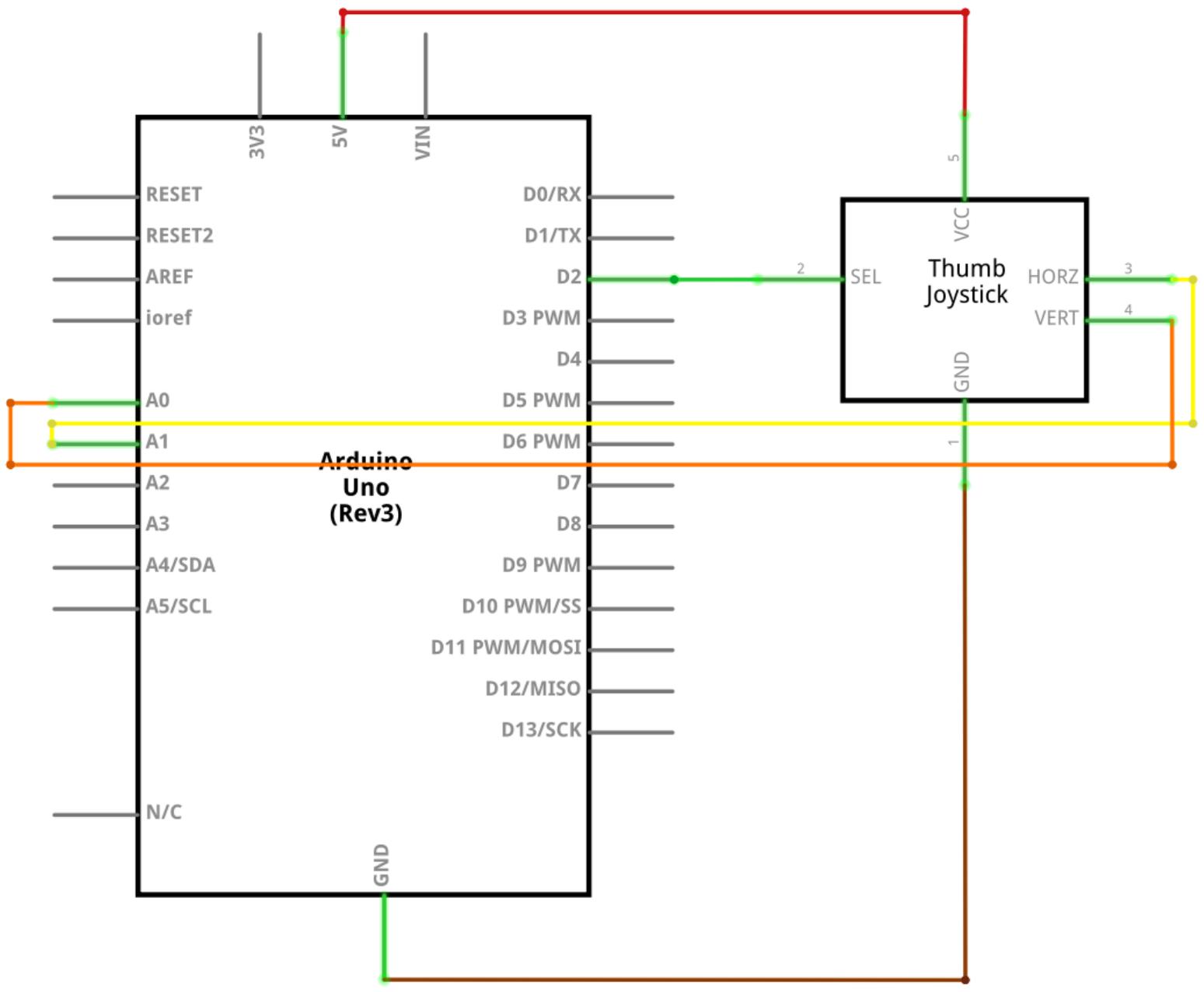
キー/セレクト・ピンから安定した読み出しを得るには、プルアップ抵抗を介して VCC に接続する必要があります。

Arduino デジタルピンに内蔵された抵抗を使用することができます。

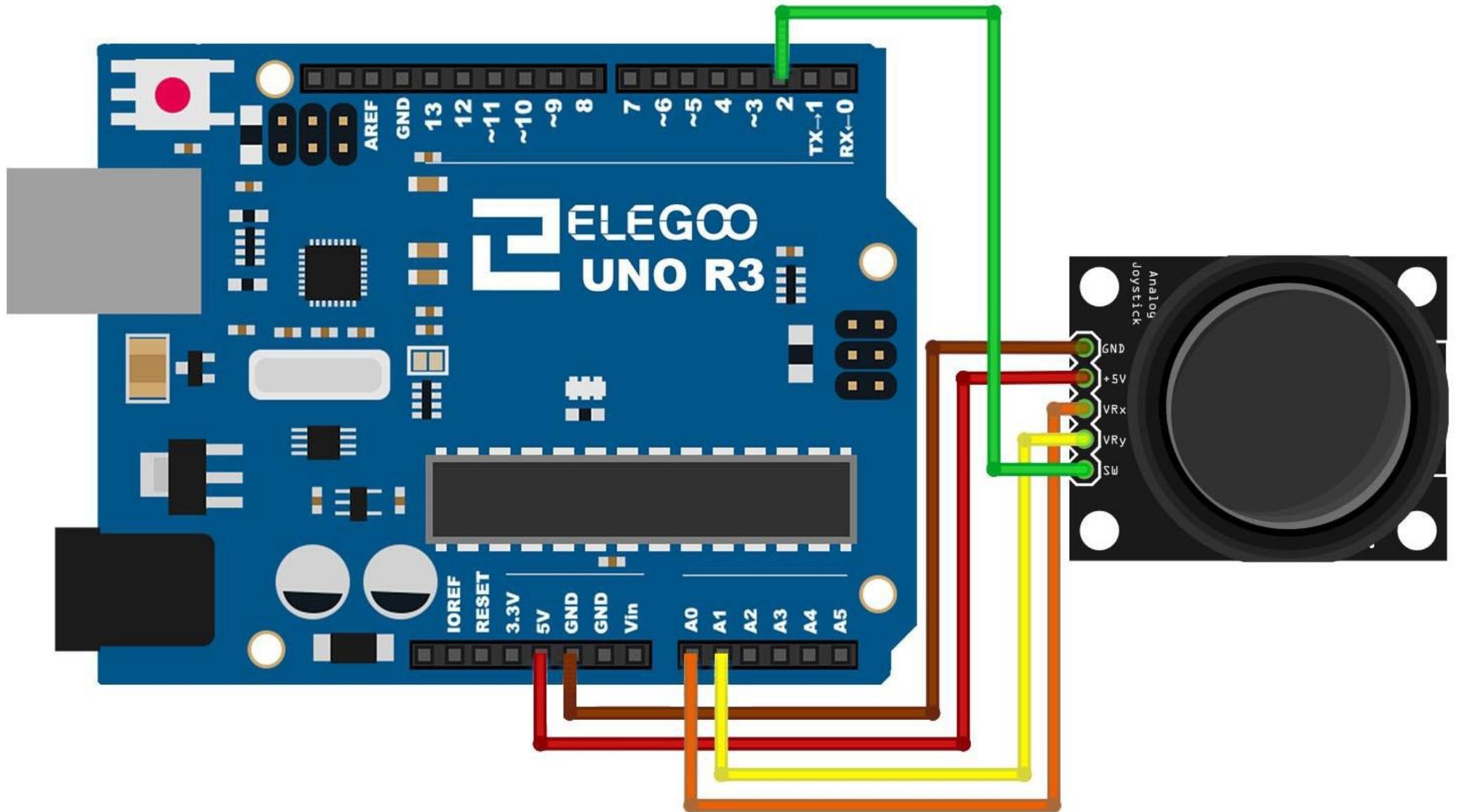
Arduino ピンのプルアップ抵抗をアクティブにする方法に関する学習



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

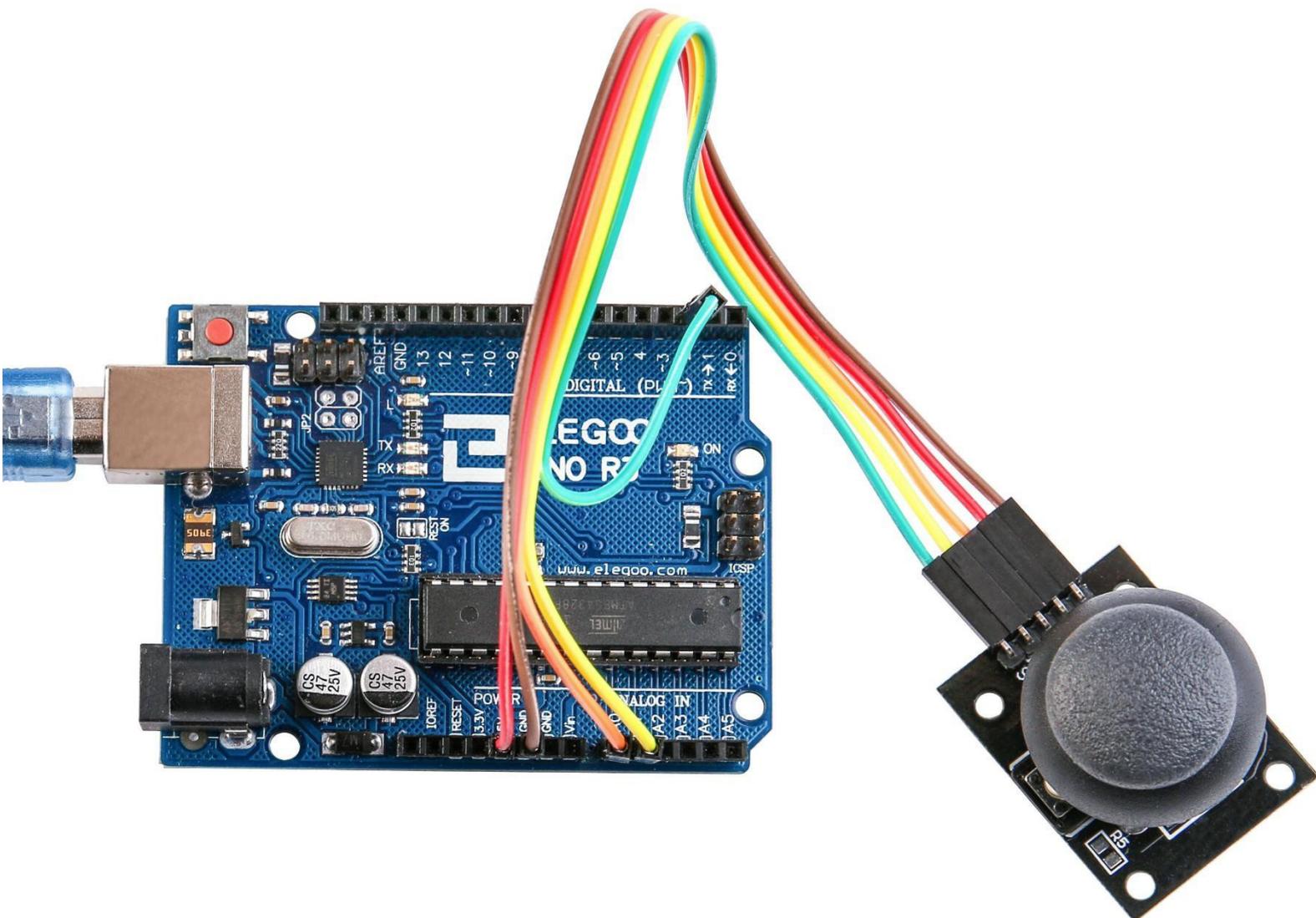
配線後、プログラムをコードフォルダ - レッスン 13 のアナログジョイスティックモジュールで開き、**UPLOAD** をクリックしてプログラムをアップロードしてください。エラーがある場合、プログラムのアップロードの詳細については、[レッスン 2](#) を参照してください。

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

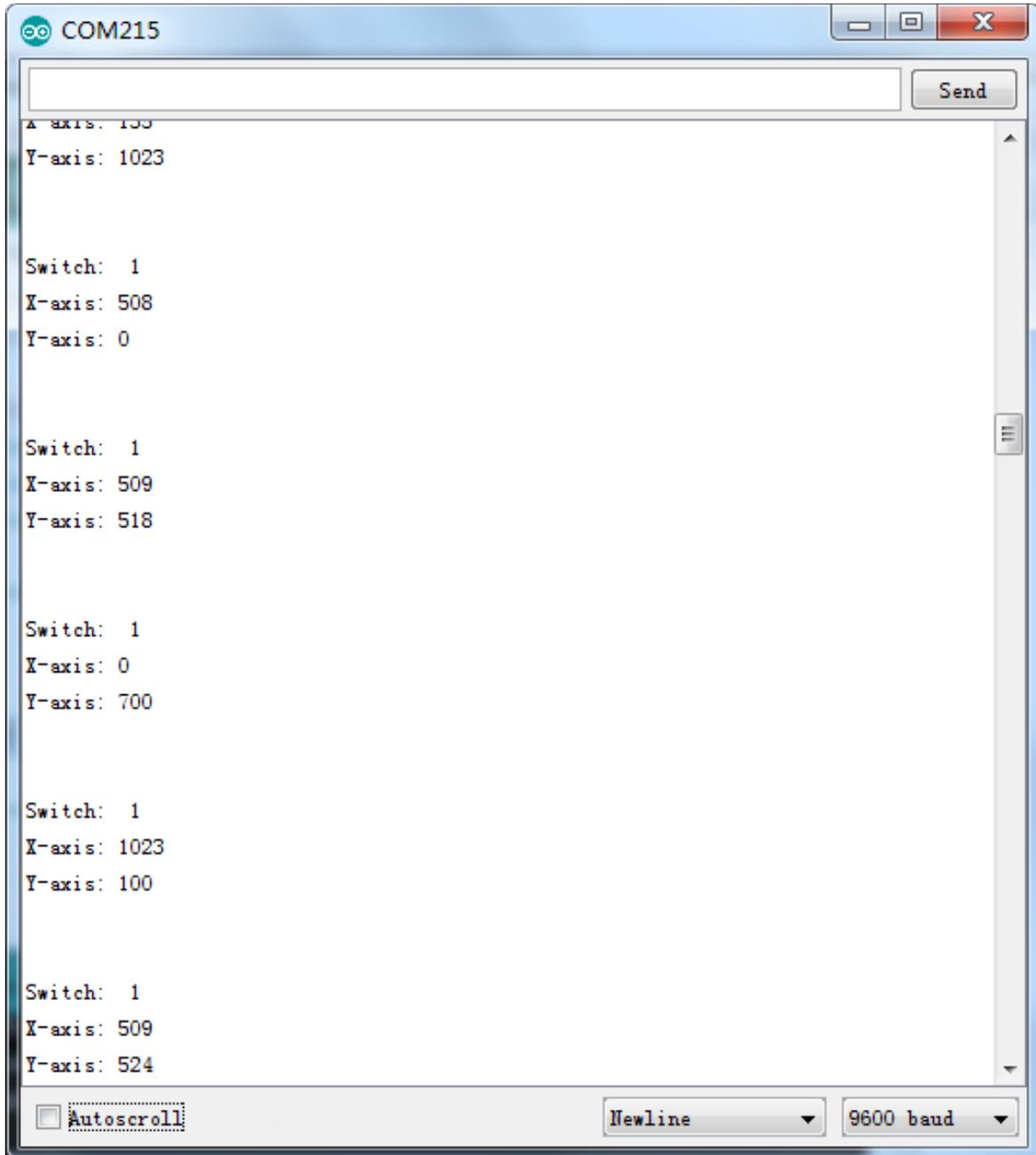
値の範囲は 0 から 1024 になります。

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 14 赤外線受信モジュール

概要

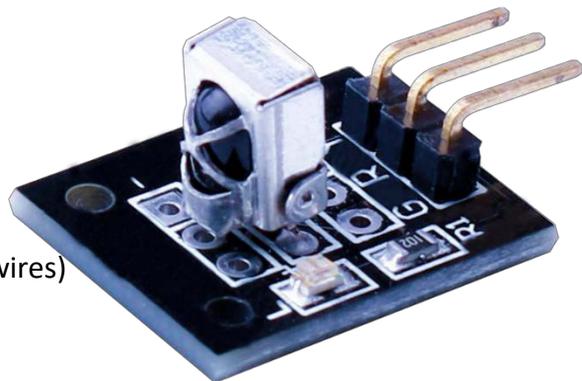
IR リモートを使用すると、プロジェクトをワイヤレスで制御することができます。

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 DuPont wires)



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、クリッカ一からの 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、赤外線パルスを送っ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。

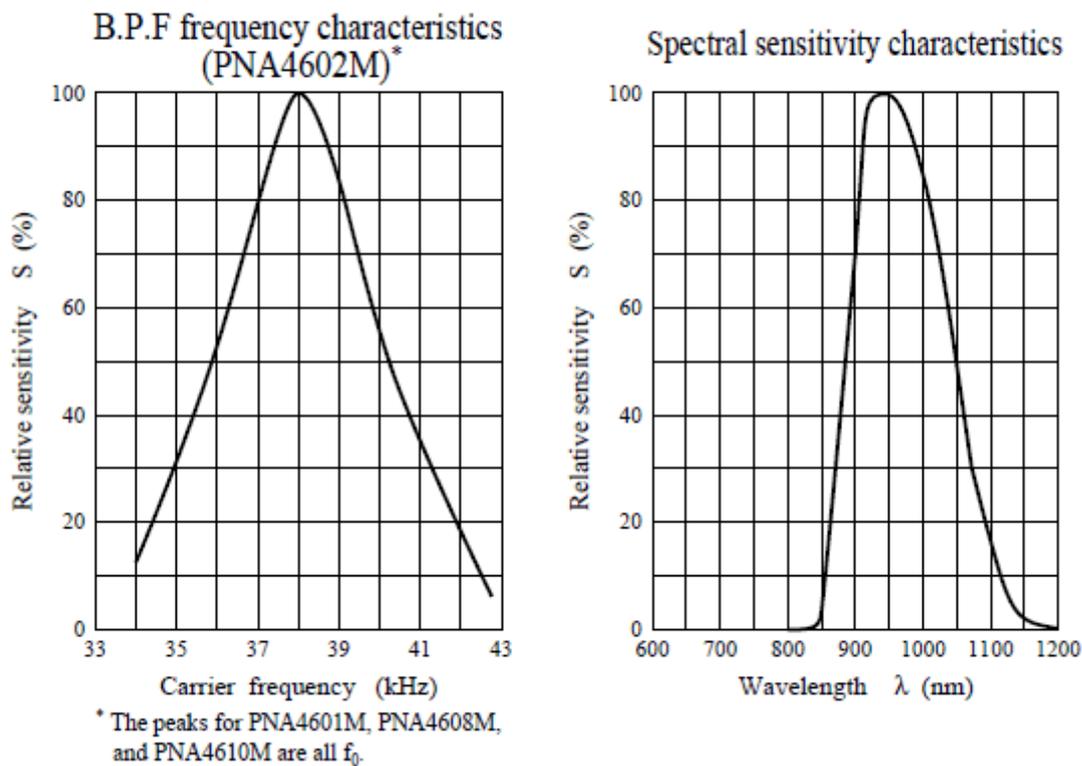
これらの間にはいくつかの違いがあり、CdS Photocells と言えます:

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不向きである。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ます。

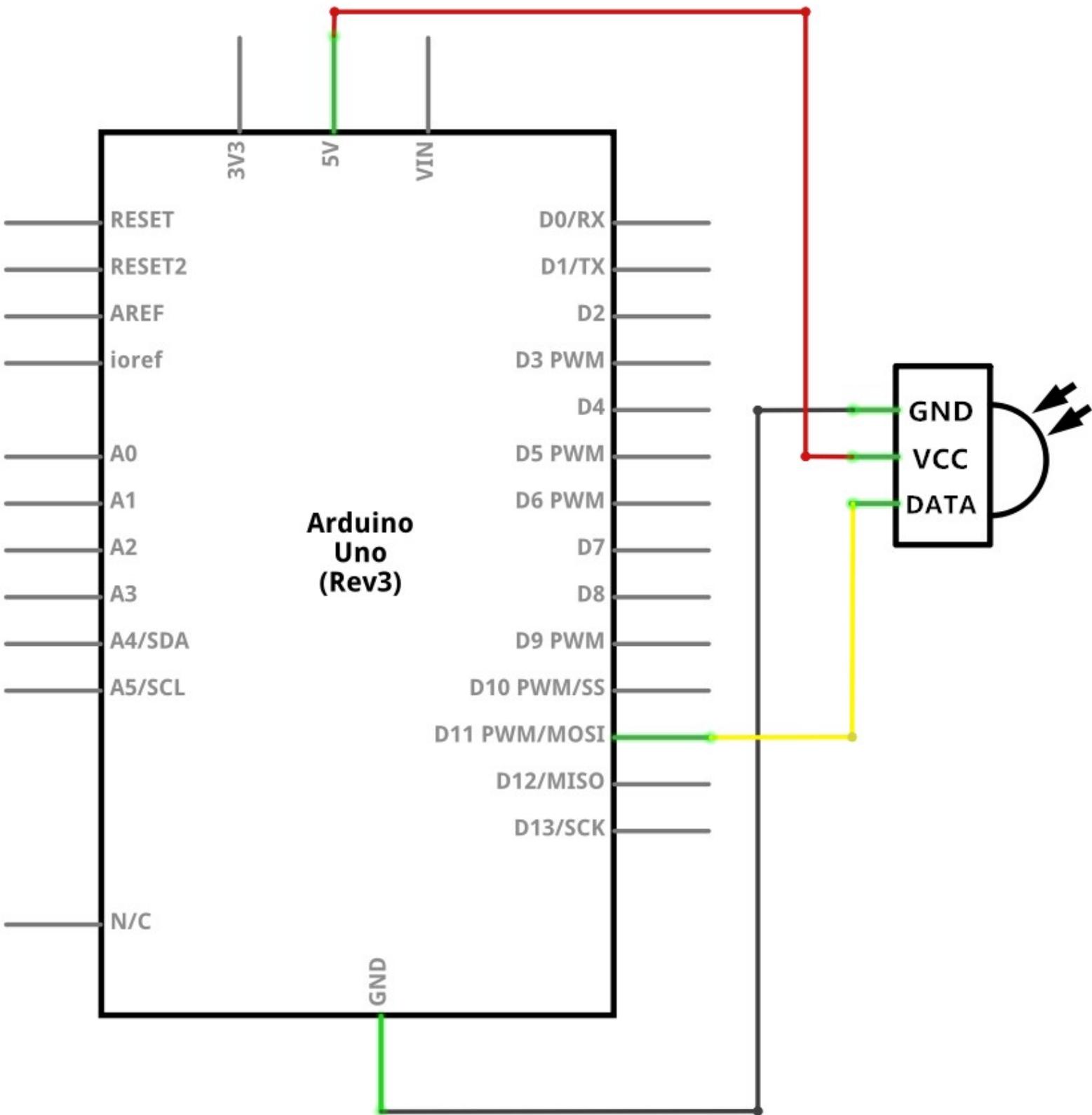
IR 検出器はデジタル出力です。これらは 38KHz IR 信号を検出してロー (0V) を出力するか、いずれも検出せず、ハイ (5V) を出力します。光電池は抵抗器のように振る舞い、どれだけの光が当たるかによって抵抗が変化します。

What You Can Measure

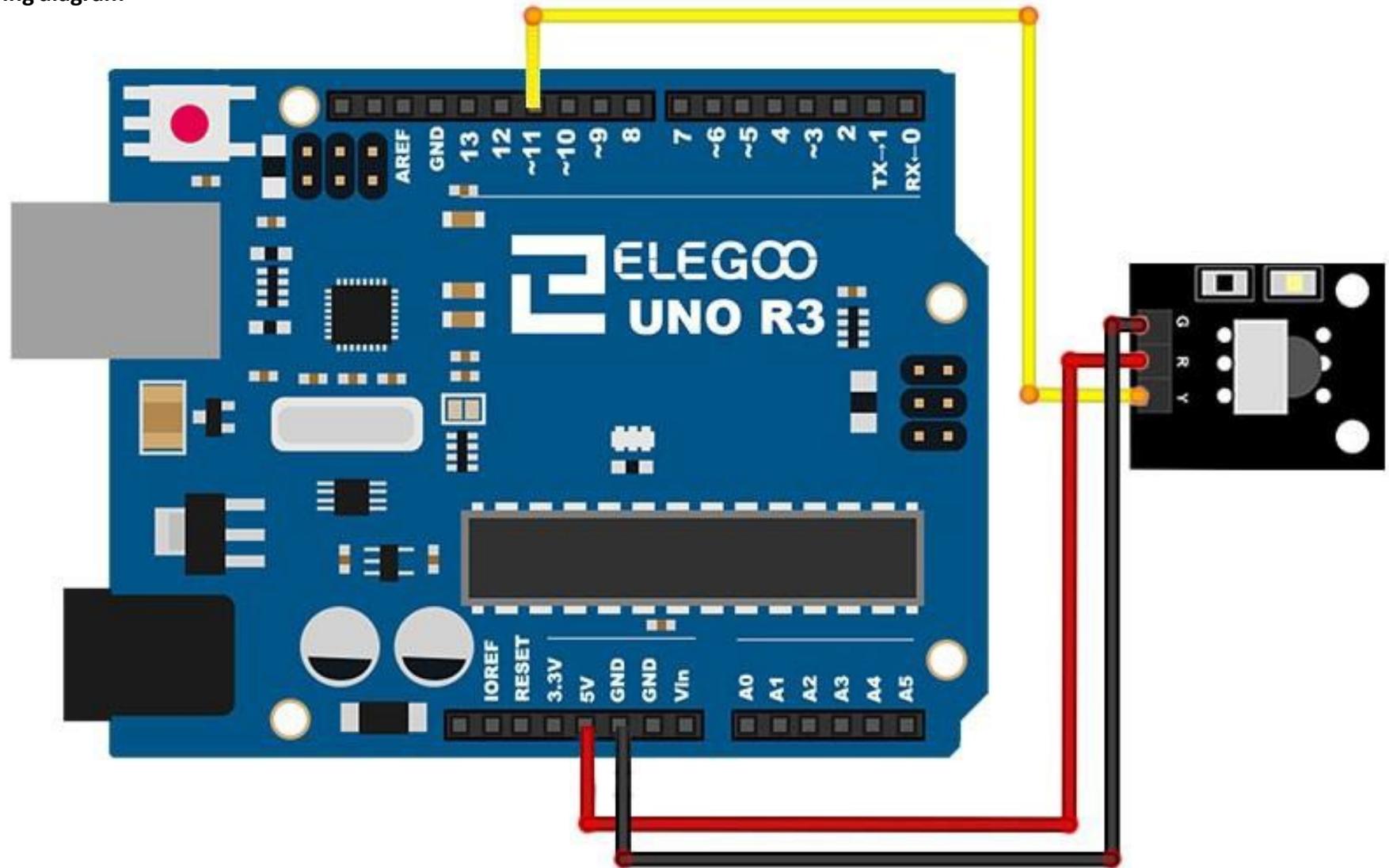


これらのデータシートのグラフからわかるように、ピーク周波数の検出は 38 KHz で、ピーク LED の色は 940 nm です。約 35KHz から 41KHz まで使用できますが、感度は低下し、遠くからも検出されません。同様に、850~1100 nm の LED を使用することもできますが、900~1000 nm と同様に動作しませんので、必ず一致する LED を取得してください! IR LED のデータシートで波長を確認してください。940nm を取得しようとする - 940nm は可視光ではないことを覚えておいてください。

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

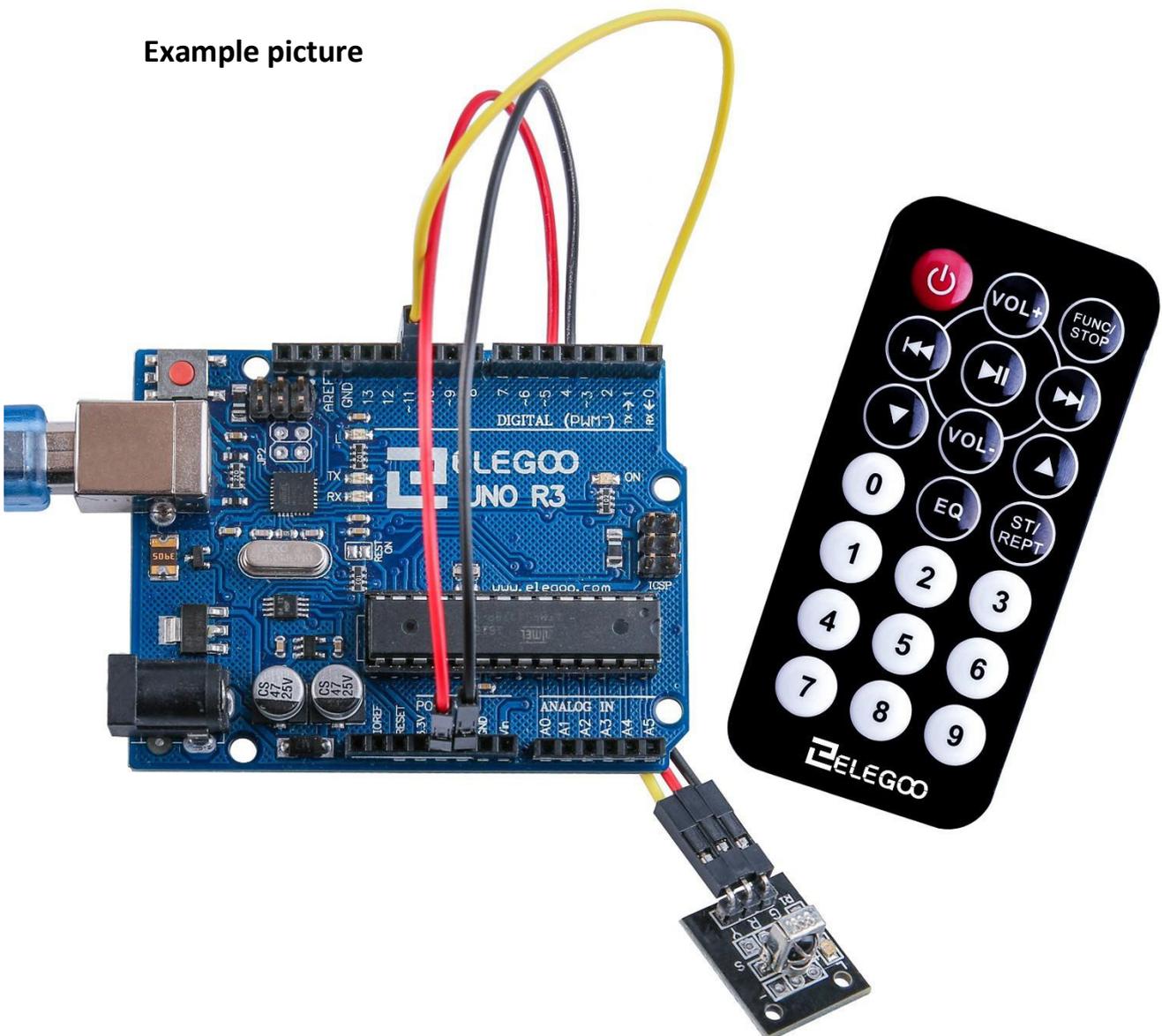
配線後、プログラムをコードフォルダ - レッスン 14 赤外線受信モジュールで開き、UPLOAD をクリックしてプログラムをアップロードしてください。エラーがある場合のプログラムアップロードの詳細については、レッスン 2 を参照してください。

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。

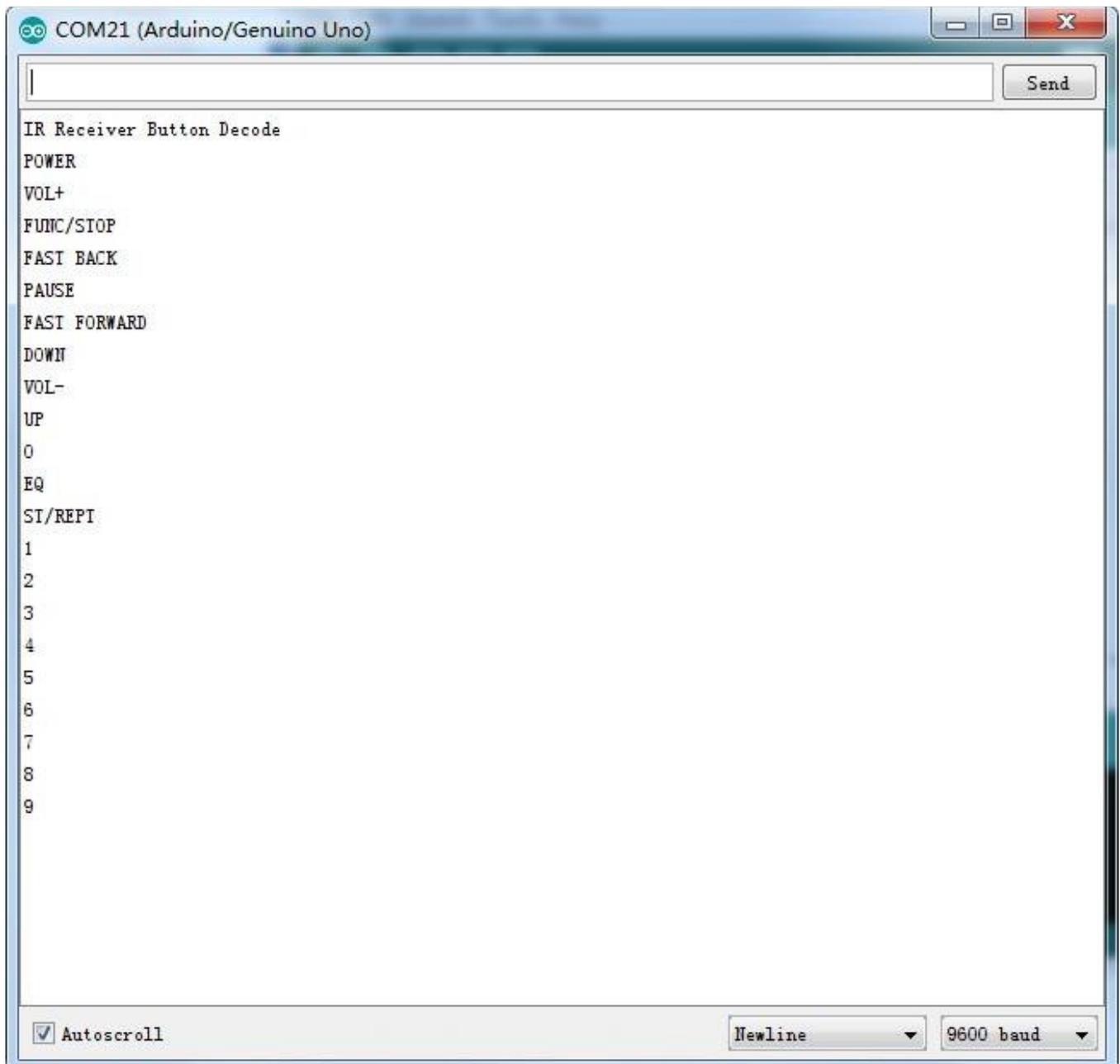
次に、<RobotIRremote>を Library フォルダから移動します。これは、ライブラリが使用するライブラリと競合するためです。 マイクロコントローラのプログラミングが完了したら、ライブラリフォルダの中にドラッグして戻すことができます。

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。



Lesson 15 MAX7219 LED ドット行列モジュール

概要

このチュートリアルでは、MAX7219 を接続し、テキストをスクロールします。

これらのモジュールは MAX7219 LED ドライバチップを使用しているため、UNO の 3 ピンのみを使用して、各モジュールの 64 個の LED をオン/オフすることができます。

必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x Max7219 module
- (5) x F-M wires (Female to Male DuPont wires)

部品の紹介

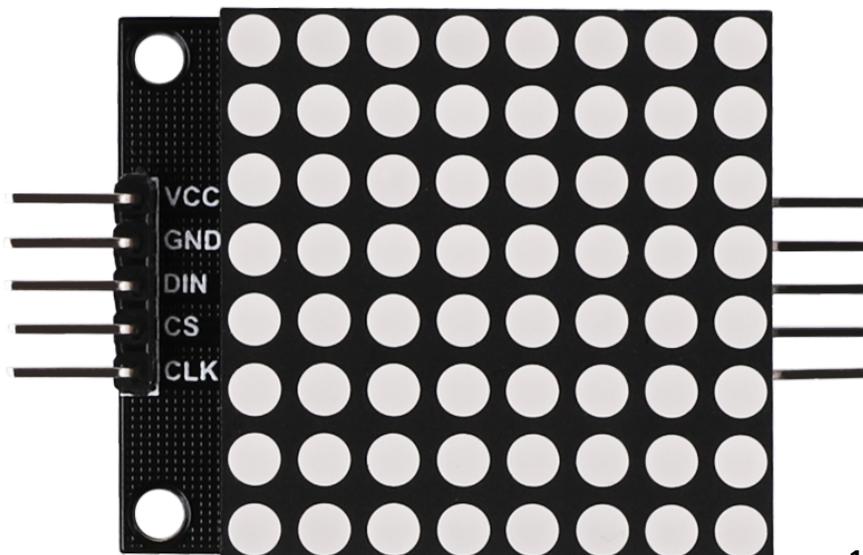
MAX7219 LED Dot Matrix Module

私たちのプロジェクトは、実際には直列接続 MAX7219 を備えた Arduino です。8X8 LED マトリクスを動作させます。MAX7219 IC は、シリアル入力/出力共通陰極ディスプレイドライバで、マイクロプロセッサを最大 8 桁の 7 セグメント数値 LED ディスプレイ、グラフ表示、または 64 個の個別 LED があります。便宜上、プリワイヤードモジュールとして利用可能な MAX7219 IC セットアップと統合された 8×8 LED マトリクスが使用されています。

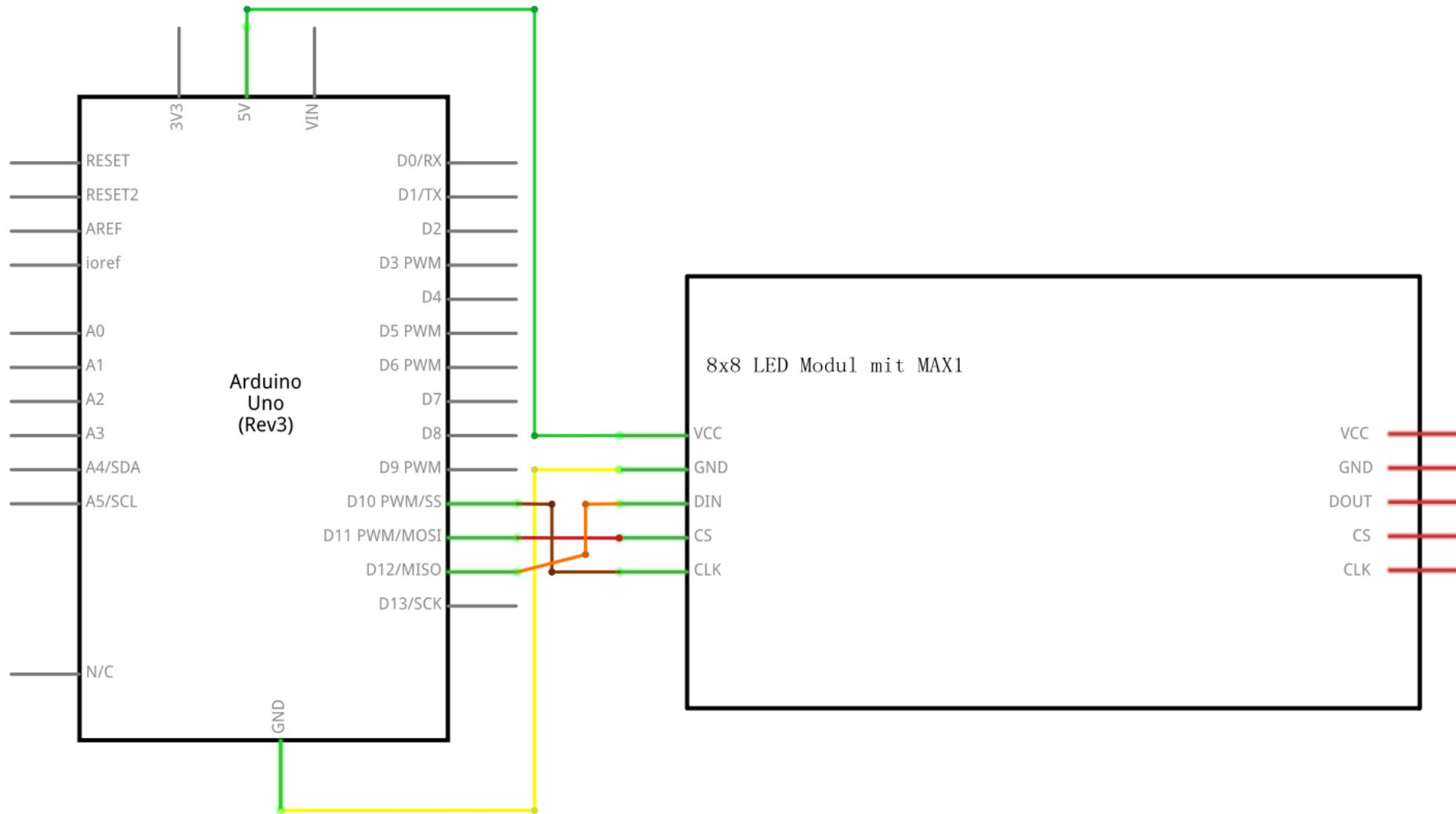
この LED マトリクスモジュールの標準仕様を以下に示します:

動作電圧: DC 4.7V – 5.3V ; 代表的な電圧:5V;

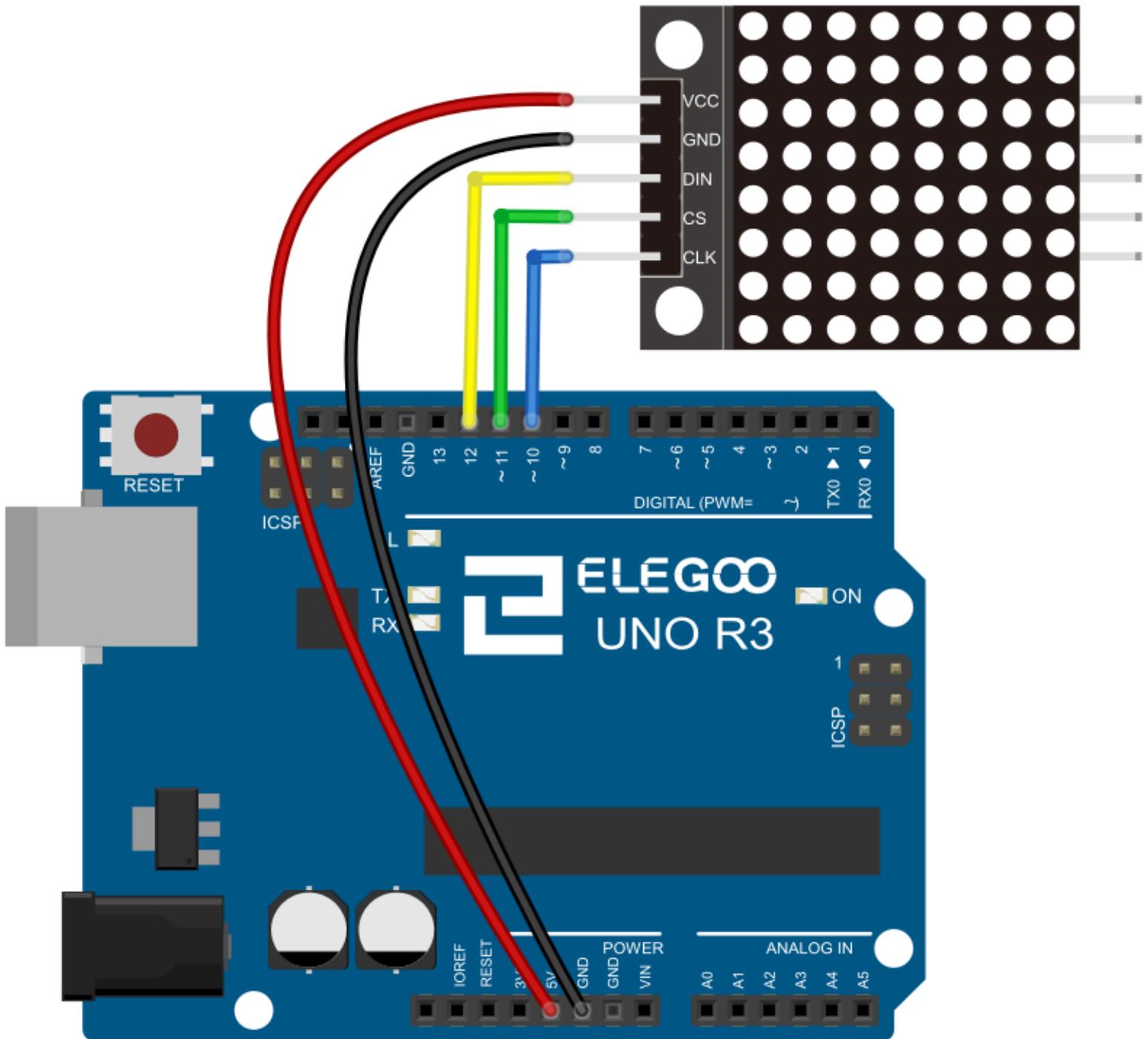
動作電流: 320mA ; 最大動作電流: 2A



Connection Schematic



Wiring diagram



VCCとグラウンドは Arduino に接続されています。

ピン 12 は DIN に接続され、ピン 11 は CS に接続され、ピン 10 は CLK に接続されます。

Code

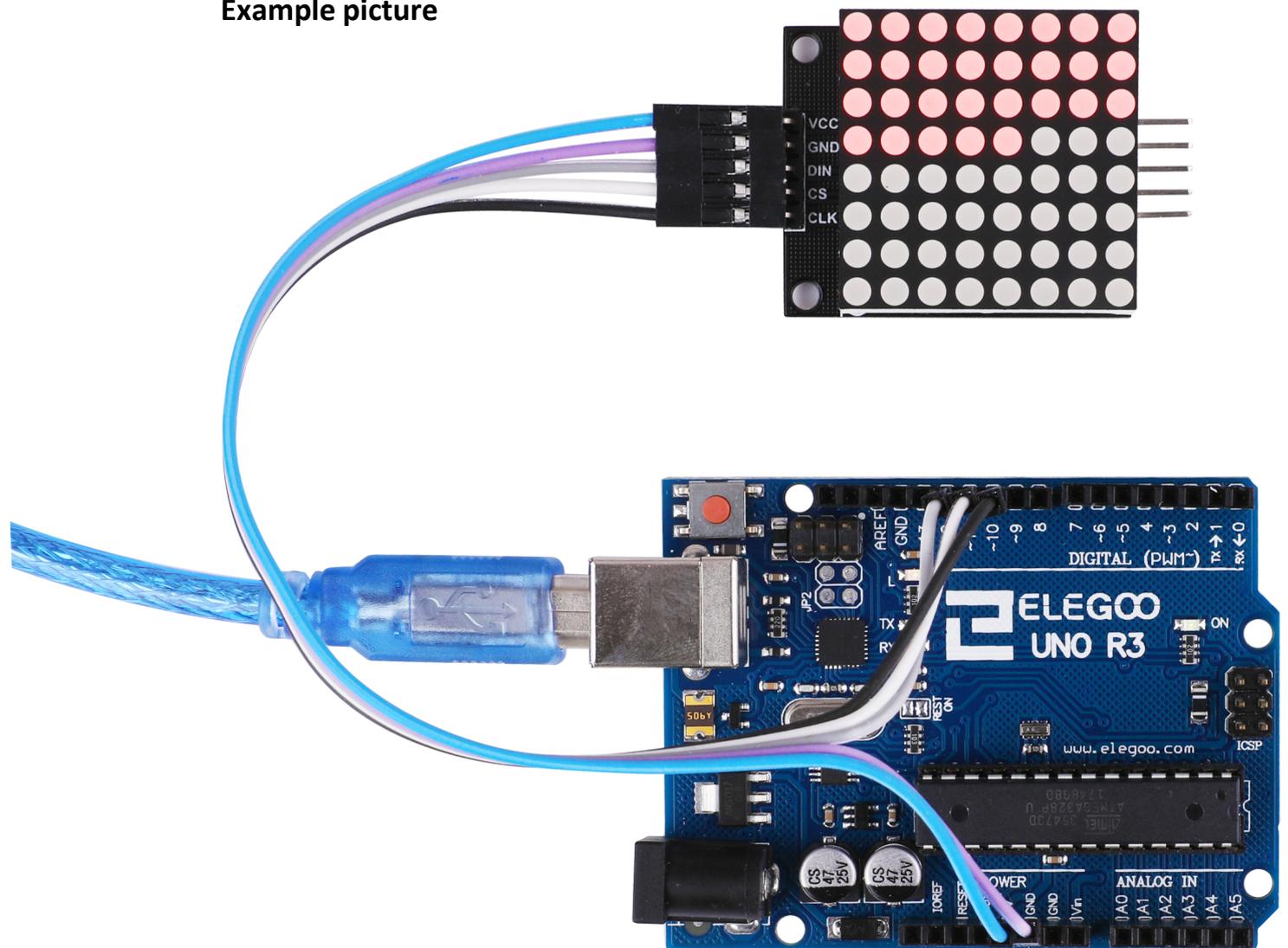
当社のスケッチは MAX7219 モジュールと通信するために「Maxmatrix」ライブラリを利用します。

配線後、プログラムをコードフォルダ - レッスン 15 MAX7219 LED ドットマトリクスモジュールで開き、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、レッスン 2 を参照してください。

これを実行する前に、<LedControl>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



Lesson 16 GY-521 モジュール

概要

このレッスンでは、Arduino と互換性がある最高の IMU（慣性計測ユニット）センサーの 1 つである GY-521 モジュールの使用方法を学習します。GY-521 のような IMU センサは、セルフバランスングロボット、UAV、スマートフォンなどに使用されます。

必要な構成部品:

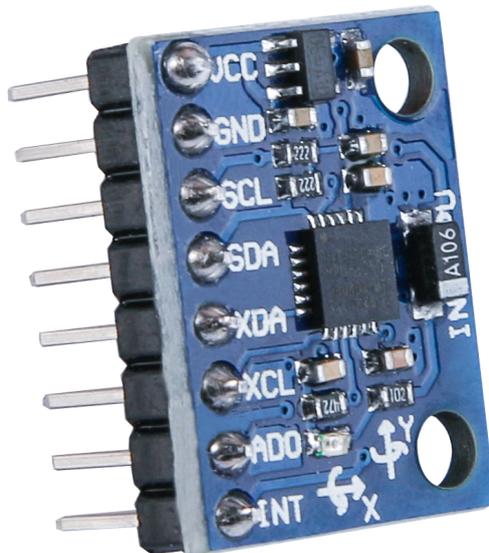
- (1) x Elegoo Uno R3
- (1) x GY-521 module
- (4) x F-M wires

部品の紹介

GY-521 SENSOR

InvenSense GY-521 センサは、MEMS 加速度計と MEMS ジャイロを 1 つのチップに内蔵しています。これは、各チャンネルの 16 ビットアナログ/デジタル変換ハードウェアが含まれているため、非常に正確です。したがって、x、y、z チャンネルを同時にキャプチャします。センサーは I2C バスを使用して Arduino とインターフェースします。

GY-521 は高価ではありません。特に、加速度計とジャイロの両方を組み合わせているためです。

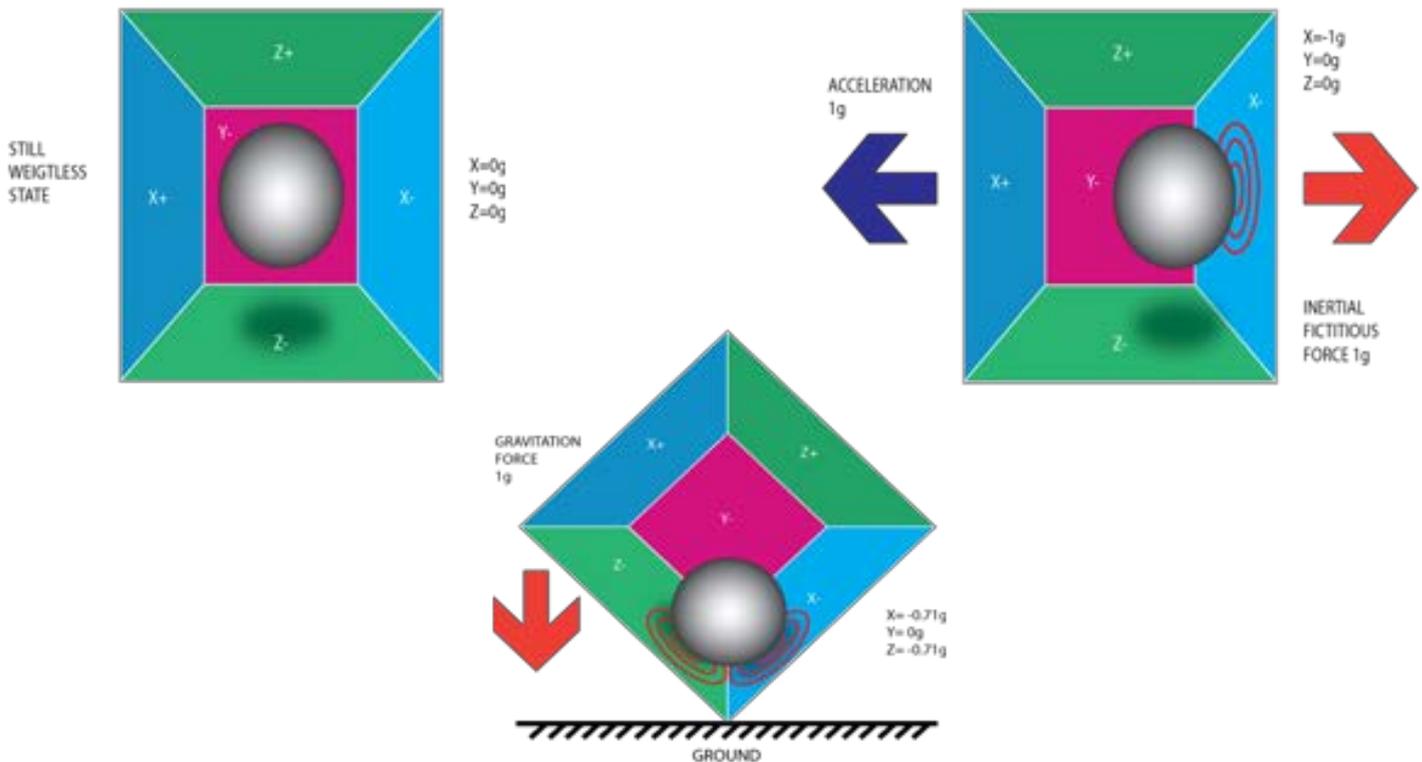


IMU センサは、あらゆる種類の電子器具で現在使用されている最も良好なタイプのセンサの 1 つです。スマートフォン、ウェアラブル、ゲームコントローラなどに見られます。IMU センサーは、3次元空間でセンサーに取り付けられた物体の姿勢を得るのに役立ちます。これらの値は通常角度で表され、その姿勢を決定するのに役立ちます。したがって、スマートフォンの向きを検出するためにスマートフォンで使用されています。また、Nike の燃料バンドやフィットビットのようなウェアラブルガジェットでは、IMU センサーを使って動きを追跡します。

How does it work?

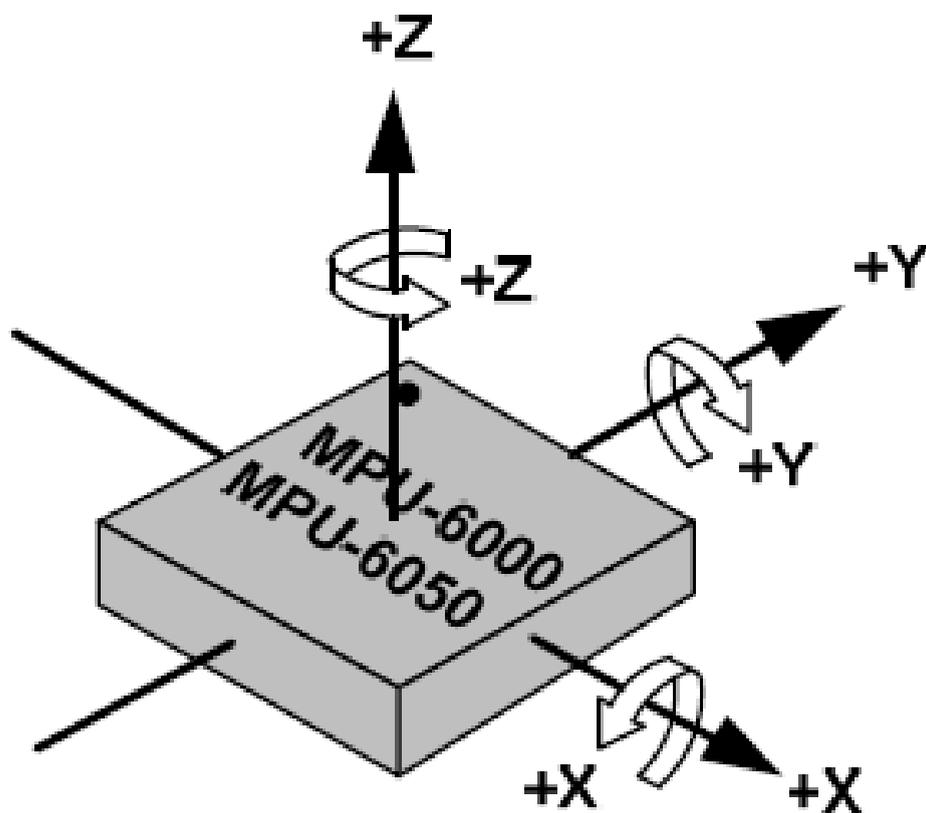
IMU センサーは、通常、2つ以上の部品で構成されています。加速度計、ジャイロスコープ、磁力計、高度計を優先的にリストします。GY-521 は 6 自由度 (Degrees of Freedom) または 6 軸 IMU センサーで、出力として 6 つの値が得られます。加速度計からの 3 つの値とジャイロスコープからの 3 つの値。GY-521 は、MEMS (Micro Electro Mechanical Systems) 技術に基づくセンサーです。加速度計とジャイロスコープは両方とも 1 つのチップに内蔵されています。このチップは、I2C (Inter Integrated Circuit) プロトコルを通信に使用します。

How does an accelerometer work?

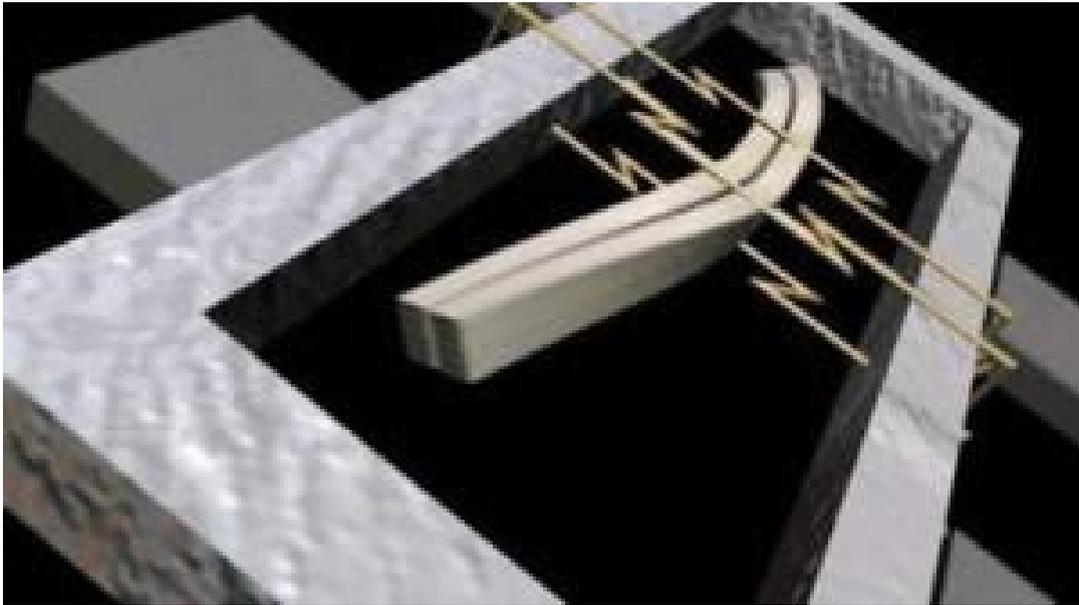


加速度計はピエゾ電気効果の原理に基づいて動作する。ここでは、上の写真のように、その中に小さなボールがある立方体のボックスを想像してください。この箱の壁はピエゾ電気結晶で作られています。箱を傾けるたびに、ボールは重力のために傾きの方向に動かされます。ボールが衝突する壁は、小さなピエゾ電流を作り出します。直方体には正反対の壁が全部 3 組あります。各ペアは 3D 空間の軸に対応します: X 軸、Y 軸、Z 軸

ピエゾ電気壁から発生する電流に依存して、傾斜の方向とその大きさを決定することができる。詳細はこちらをご覧ください。



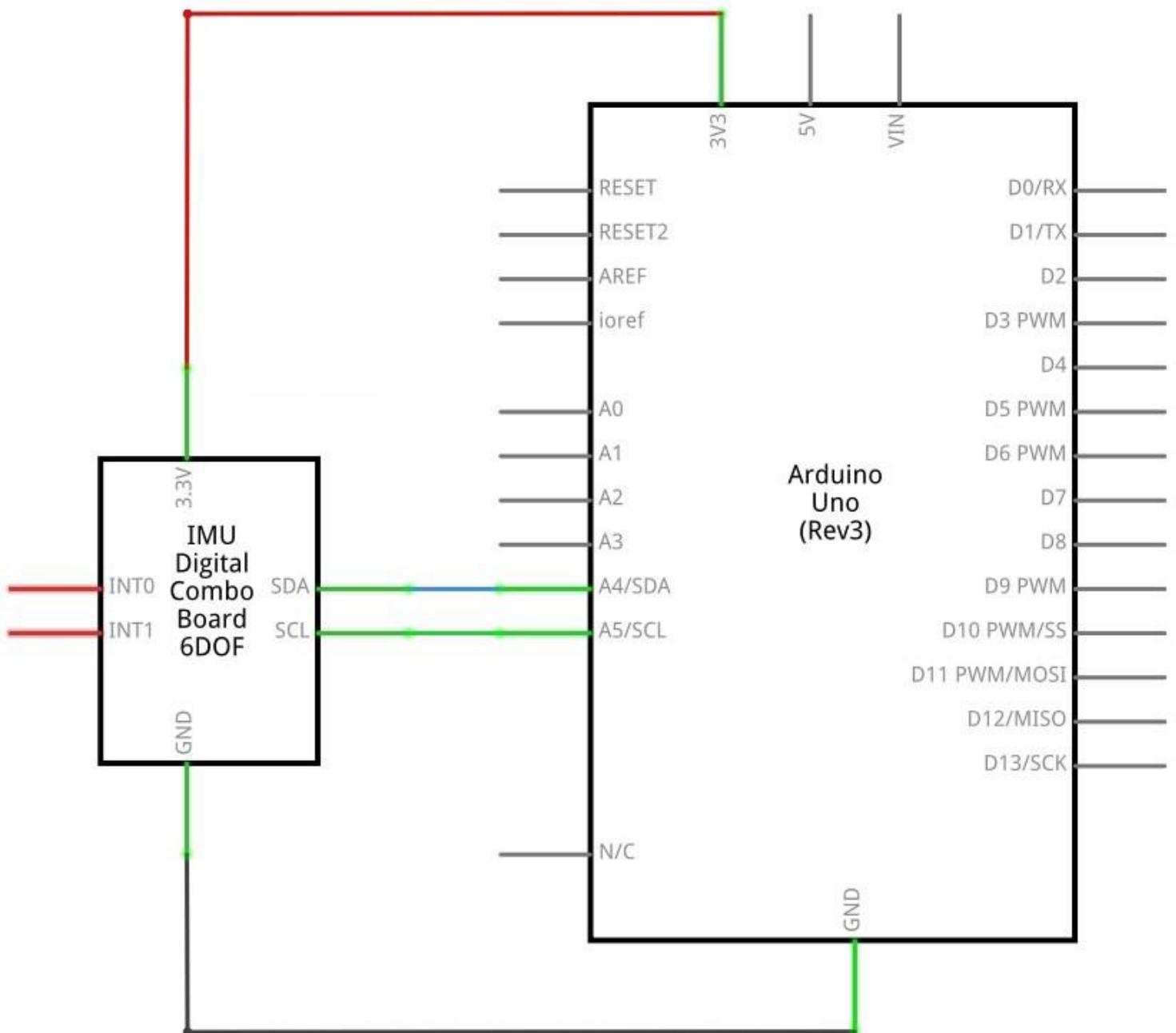
How does a gyroscope work?



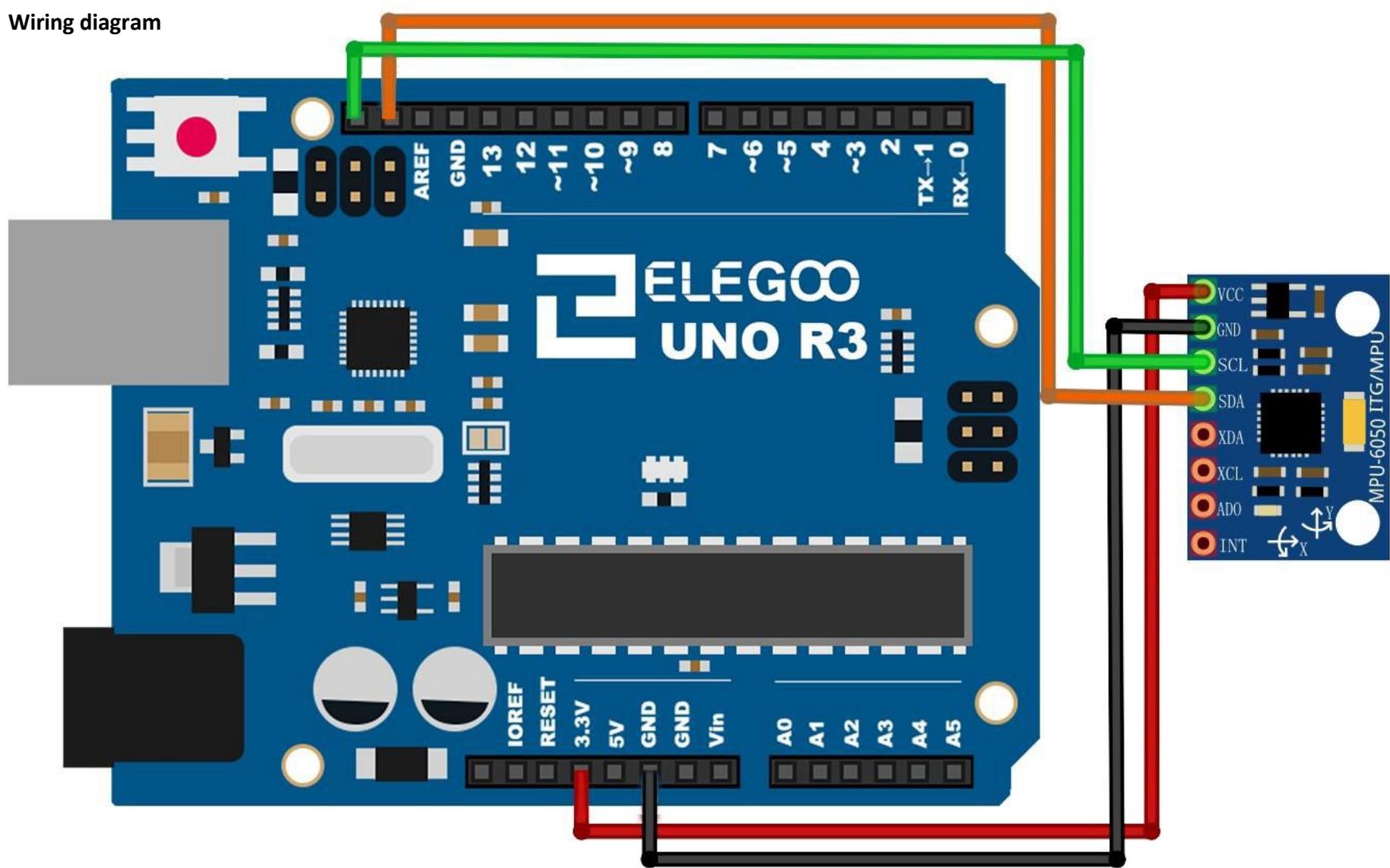
ジャイロスコープはコリオリ加速の原理に基づいて動作します。フォークのような構造があり、それは一定の前後運動であると想像してください。これは、圧電結晶を用いて所定位置に保持される。この配置を傾けようとする、結晶は傾きの方向に力を受けます。これは、移動フォークの慣性の結果として生じる。従って結晶は、ピエゾ電気効果とのコンセンサスで電流を生成し、この電流が増幅される。その後、ホストマイクロコントローラによって値が洗練されます。

Connection

Schematic



Wiring diagram



次に、I2C 回線を設定する必要があります。このために、GY-521 の SDA と書かれたピンを Arduino のアナログピン 4 (SDA) に接続します。そして、GY-521 の SCL とラベルされたピンは Arduino のアナログピン 5 (SCL) に接続されています。Arduino GY-521 の配線が完了しました。

Libraries needed

MPU-6050

The Code

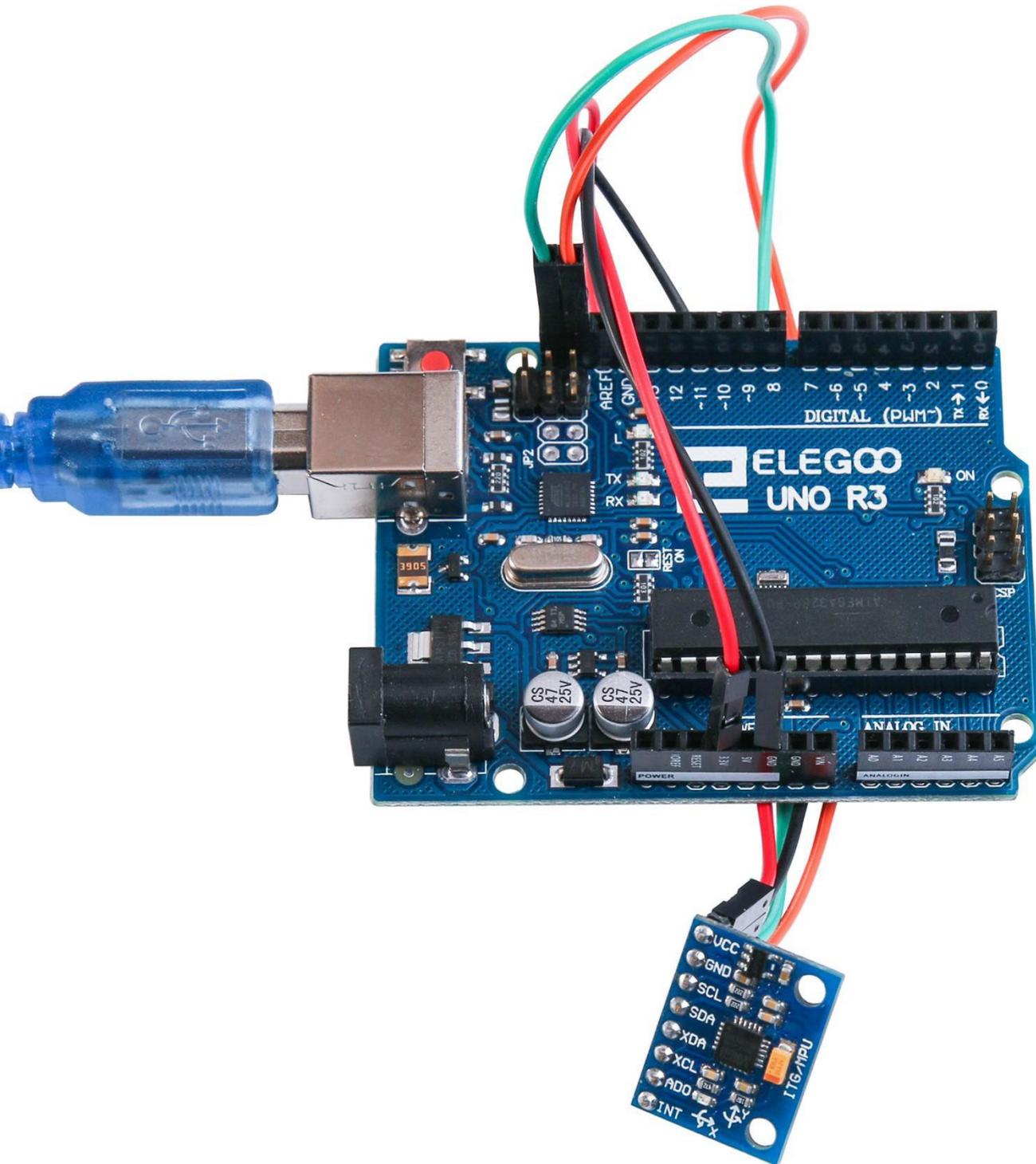
短い例題のスケッチは非常に短いスケッチで、すべての生の値（加速度計、ジャイロ、温度）を表示します。Arduino Uno、Nano、Leonardo、Due にも対応しています。

配線後、プログラムをコードフォルダ - レッスン 16 GY-521 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、<GY-521>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。

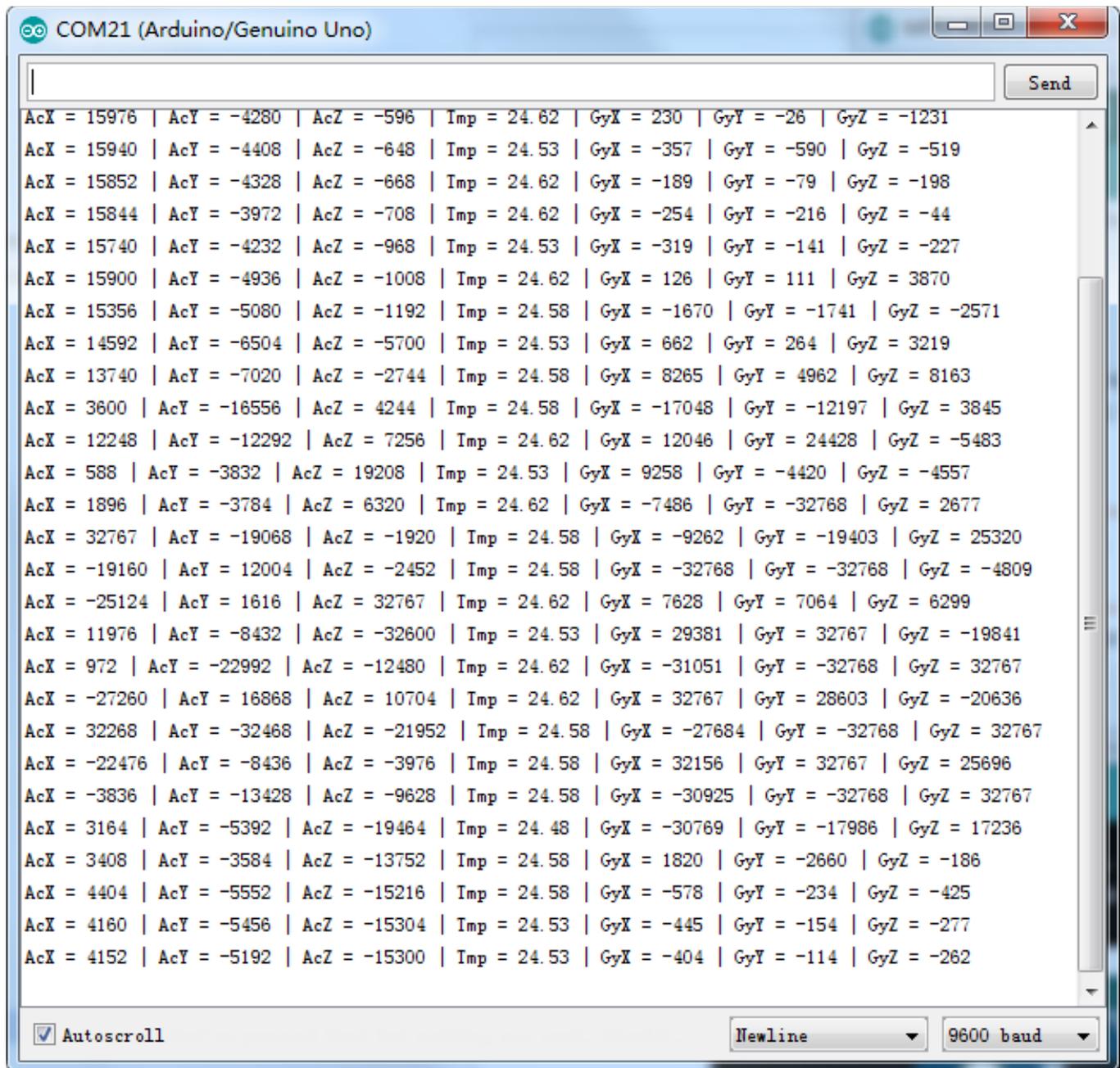
ライブラリファイルのロードの詳細については、[レッスン 1](#) を参照してください。

Example picture



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。



The screenshot shows the 'COM21 (Arduino/Genuino Uno)' serial monitor window. The data displayed is a series of lines, each containing eight numerical values separated by vertical bars. The values appear to be sensor readings. The data is as follows:

```
AcX = 15976 | AcY = -4280 | AcZ = -596 | Imp = 24.62 | GyX = 230 | GyY = -26 | GyZ = -1231
AcX = 15940 | AcY = -4408 | AcZ = -648 | Imp = 24.53 | GyX = -357 | GyY = -590 | GyZ = -519
AcX = 15852 | AcY = -4328 | AcZ = -668 | Imp = 24.62 | GyX = -189 | GyY = -79 | GyZ = -198
AcX = 15844 | AcY = -3972 | AcZ = -708 | Imp = 24.62 | GyX = -254 | GyY = -216 | GyZ = -44
AcX = 15740 | AcY = -4232 | AcZ = -968 | Imp = 24.53 | GyX = -319 | GyY = -141 | GyZ = -227
AcX = 15900 | AcY = -4936 | AcZ = -1008 | Imp = 24.62 | GyX = 126 | GyY = 111 | GyZ = 3870
AcX = 15356 | AcY = -5080 | AcZ = -1192 | Imp = 24.58 | GyX = -1670 | GyY = -1741 | GyZ = -2571
AcX = 14592 | AcY = -6504 | AcZ = -5700 | Imp = 24.53 | GyX = 662 | GyY = 264 | GyZ = 3219
AcX = 13740 | AcY = -7020 | AcZ = -2744 | Imp = 24.58 | GyX = 8265 | GyY = 4962 | GyZ = 8163
AcX = 3600 | AcY = -16556 | AcZ = 4244 | Imp = 24.58 | GyX = -17048 | GyY = -12197 | GyZ = 3845
AcX = 12248 | AcY = -12292 | AcZ = 7256 | Imp = 24.62 | GyX = 12046 | GyY = 24428 | GyZ = -5483
AcX = 588 | AcY = -3832 | AcZ = 19208 | Imp = 24.53 | GyX = 9258 | GyY = -4420 | GyZ = -4557
AcX = 1896 | AcY = -3784 | AcZ = 6320 | Imp = 24.62 | GyX = -7486 | GyY = -32768 | GyZ = 2677
AcX = 32767 | AcY = -19068 | AcZ = -1920 | Imp = 24.58 | GyX = -9262 | GyY = -19403 | GyZ = 25320
AcX = -19160 | AcY = 12004 | AcZ = -2452 | Imp = 24.58 | GyX = -32768 | GyY = -32768 | GyZ = -4809
AcX = -25124 | AcY = 1616 | AcZ = 32767 | Imp = 24.62 | GyX = 7628 | GyY = 7064 | GyZ = 6299
AcX = 11976 | AcY = -8432 | AcZ = -32600 | Imp = 24.53 | GyX = 29381 | GyY = 32767 | GyZ = -19841
AcX = 972 | AcY = -22992 | AcZ = -12480 | Imp = 24.62 | GyX = -31051 | GyY = -32768 | GyZ = 32767
AcX = -27260 | AcY = 16868 | AcZ = 10704 | Imp = 24.62 | GyX = 32767 | GyY = 28603 | GyZ = -20636
AcX = 32268 | AcY = -32468 | AcZ = -21952 | Imp = 24.58 | GyX = -27684 | GyY = -32768 | GyZ = 32767
AcX = -22476 | AcY = -8436 | AcZ = -3976 | Imp = 24.58 | GyX = 32156 | GyY = 32767 | GyZ = 25696
AcX = -3836 | AcY = -13428 | AcZ = -9628 | Imp = 24.58 | GyX = -30925 | GyY = -32768 | GyZ = 32767
AcX = 3164 | AcY = -5392 | AcZ = -19464 | Imp = 24.48 | GyX = -30769 | GyY = -17986 | GyZ = 17236
AcX = 3408 | AcY = -3584 | AcZ = -13752 | Imp = 24.58 | GyX = 1820 | GyY = -2660 | GyZ = -186
AcX = 4404 | AcY = -5552 | AcZ = -15216 | Imp = 24.58 | GyX = -578 | GyY = -234 | GyZ = -425
AcX = 4160 | AcY = -5456 | AcZ = -15304 | Imp = 24.53 | GyX = -445 | GyY = -154 | GyZ = -277
AcX = 4152 | AcY = -5192 | AcZ = -15300 | Imp = 24.53 | GyX = -404 | GyY = -114 | GyZ = -262
```

The window also features a 'Send' button at the top right, an 'Autoscroll' checkbox checked at the bottom left, and dropdown menus for 'Newline' and '9600 baud' at the bottom right.

Lesson 17 HC-SR501 PIR 検知器

概要

このレッスンでは、UNO で PIR 移動検出器を使用する方法を学習します。UNO はこのプロジェクトの中心です。それは PIR センサを「リッスン」し、動きが検出されると LED の点灯または消灯を指示します。

必要な構成部品:

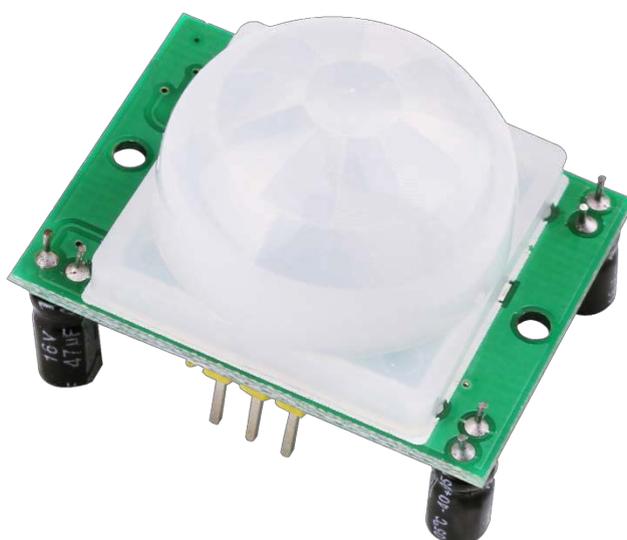
- (1) x Elegoo Uno R3
- (1) x HC-SR501 PIR motion sensor
- (3) x F-M wires (Female to Male DuPont wires)

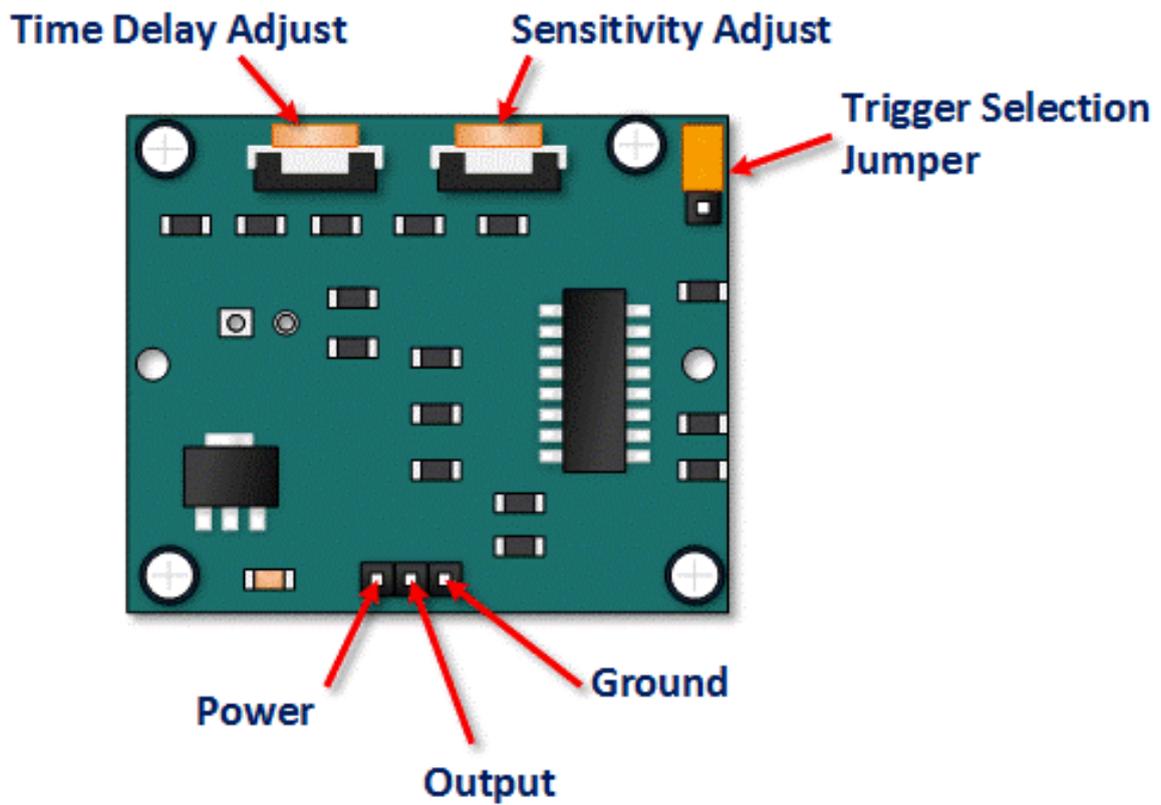
部品の紹介

PIR SENSOR:

PIR センサーは、センサーの入力と出力に影響を与える複数の変数があるため、このチュートリアルで説明されている他の多くのセンサー（フォトセル、FSR、および傾斜スイッチなど）よりも複雑です。

PIR センサ自体には 2 つのスロットがあります。各スロットは IR に敏感な特別な素材で作られています。ここで使用されているレンズはあまり効果がないので、2 つのスロットがある距離（基本的にはセンサーの感度）を「見る」ことができることがわかります。センサーがアイドル状態のとき、両方のスロットは同じ量の IR、室内または壁または屋外から放射された周囲の量を検出します。人間や動物のような暖かい体が通り過ぎると、最初に PIR センサの半分を傍受し、2 つの半分の間に正の差動変化を引き起こす。暖かい物体が感知領域を離れると、逆のことが起こり、それによってセンサーは負の微分変化を生成する。これらの変化パルスは検出されたパルスである。





Pin or Control	Function
タイマー調整	動作を検出した後、出力がどれくらい長く続くかを設定します。どこからでも 5 秒から 5 分です。
感度調整	検出範囲を 3 メートルから 7 メートルに設定します
トリが選択ジャンパ	単一または反復可能なトリガーに設定します。
Ground pin	Ground input
出力 Pin	動きが検出されない場合は低、動きが検出された場合は高です。ハイは 3.3V です
Power Pin	5 to 20 VDC Supply input

HC SR501 PIR Functional Description

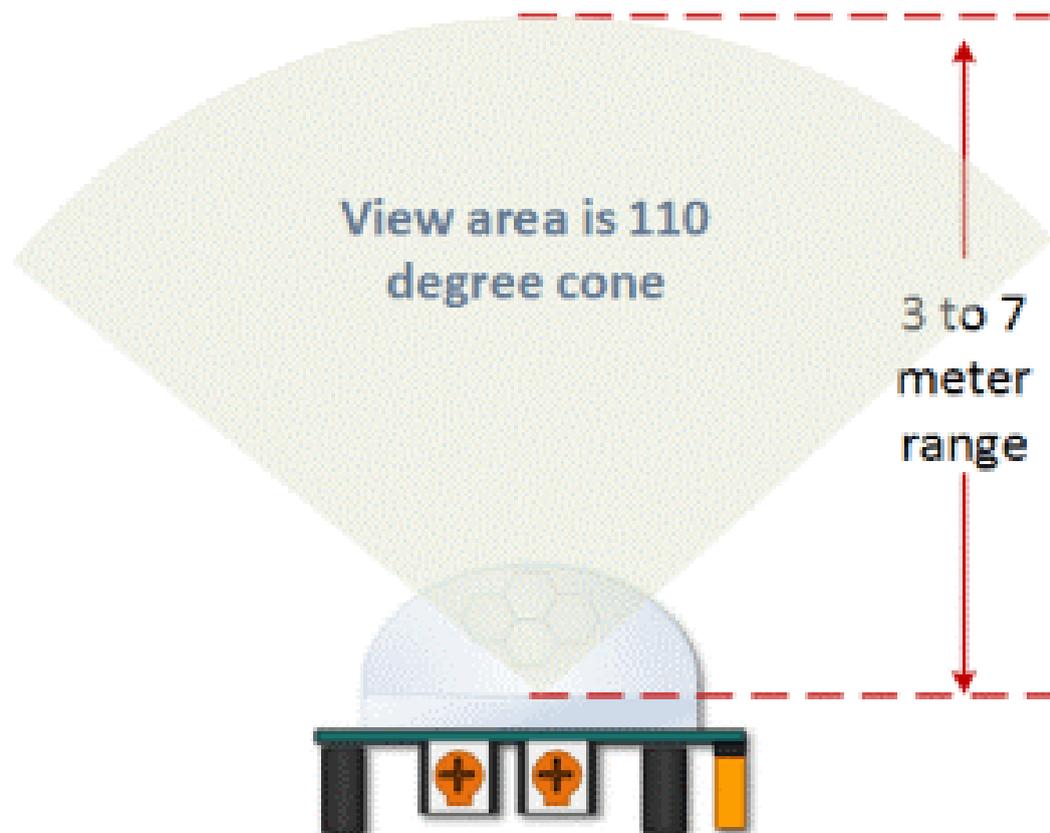
SR501 は赤外線の変化を検出し、動きとして解釈されると出力を低く設定します。動きと解釈されるかどうかは、ユーザーの設定と調整に大きく依存します。

Device Initialization

デバイスを初期化するのにほぼ 1 分かかります。この期間中、誤検出信号を数回出力します。回路またはコントローラのロジックは、この初期化期間を考慮する必要があります。

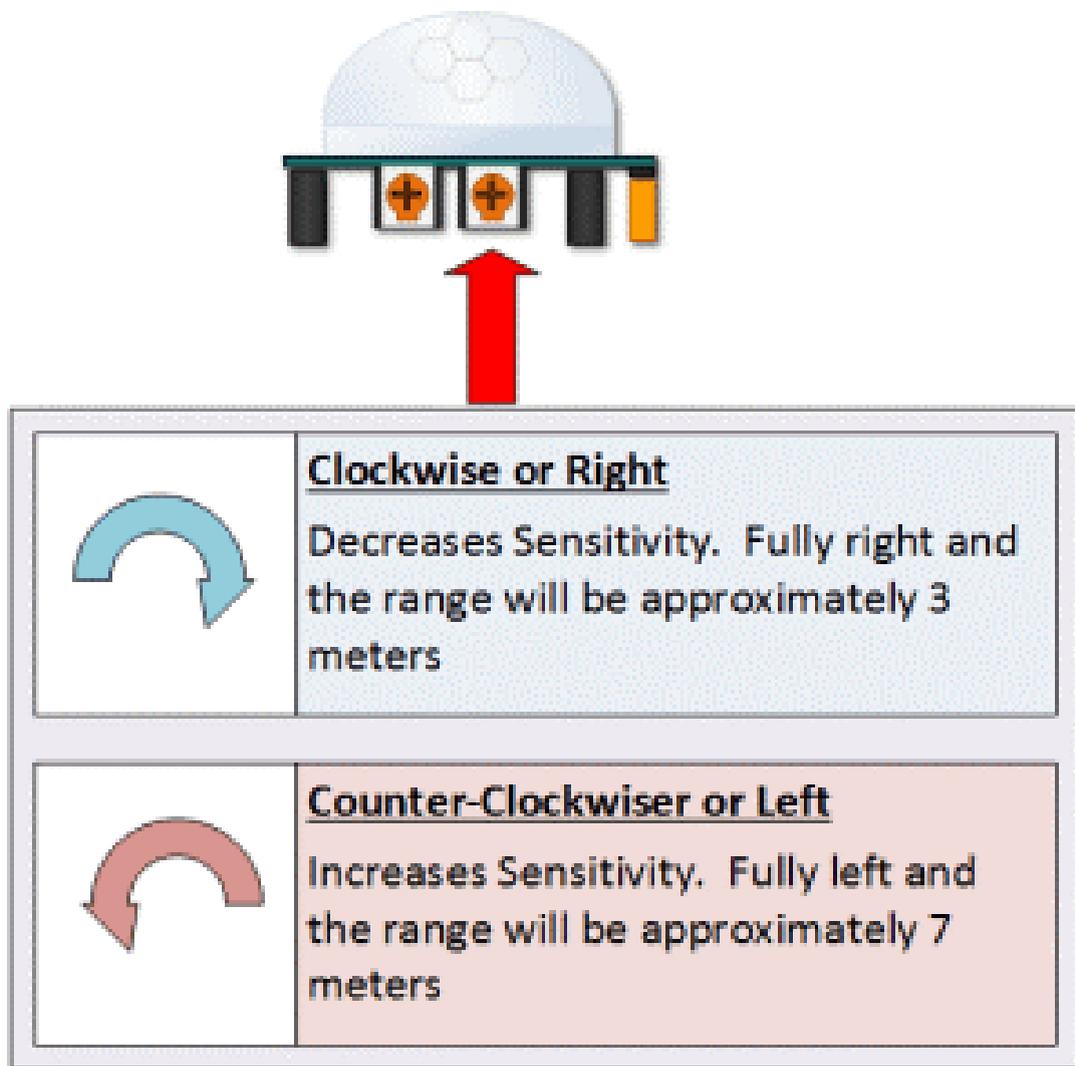
Device Area of Detection

このデバイスは、3~7メートルの範囲で 110 度の円錐の内側の動きを検出します。



HC SR501 View Area

PIR レンジ（感度）調整 前述したように、調節可能な範囲は約 3~7 メートルである。下の図は、この調整を示しています。



HC SR501 感度調整時間遅延調整

時間遅延調整は、検出動作後に PIR センサモジュールの出力がどれだけ長く続くかを決定する。範囲は約 3 秒~5 分です。

☆**Clockwise or Right** 時計回りまたは右回り

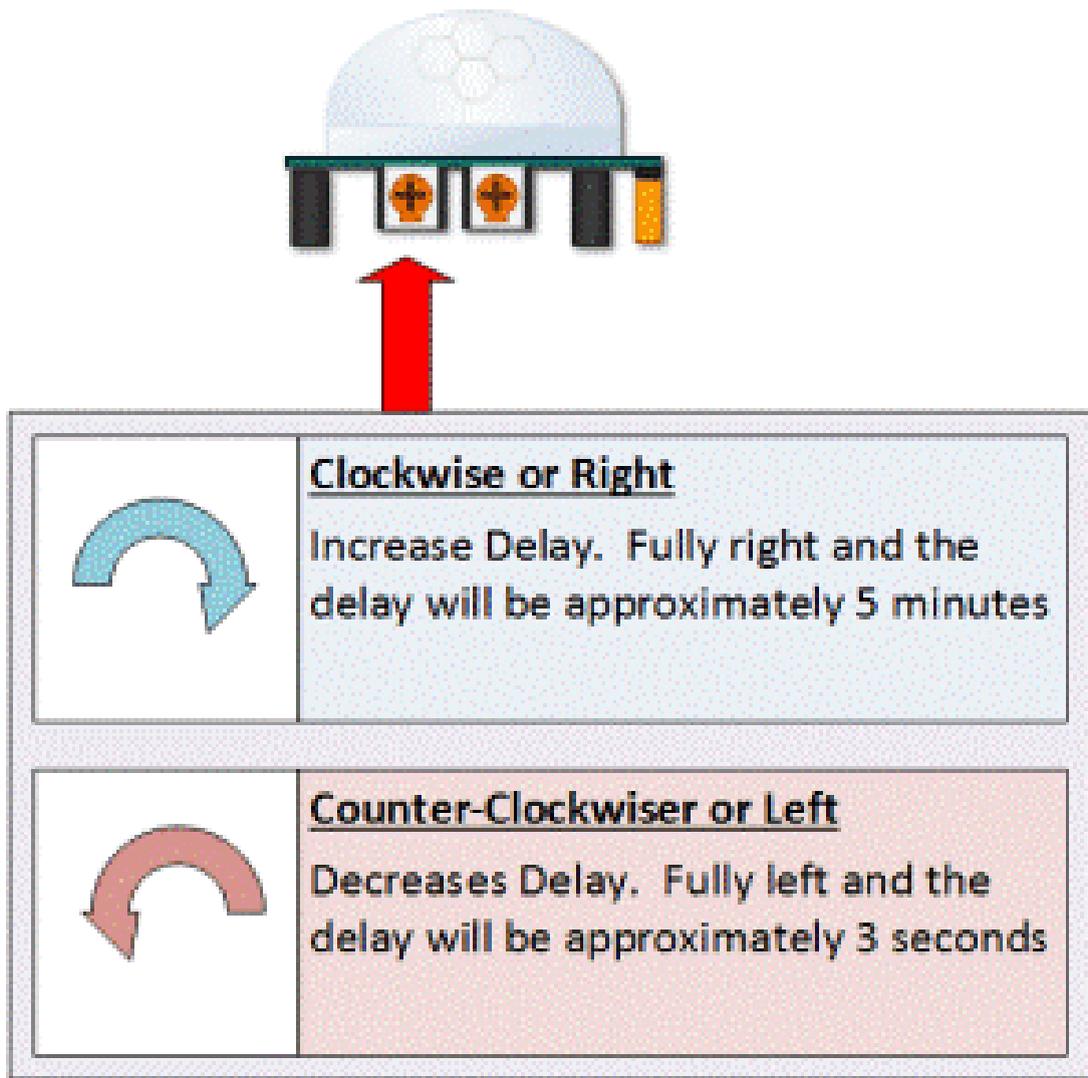
Decreases Sensitivity. Fully right and the range will be approximately 3 meters.

感度が低下します。完全に正しいと範囲は約 3 メートルになります。

Counter-Clockwise or Left 反時計回りまたは左

Increases Sensitivity. Fully left and the range will be approximately 7 meters.

感度を上げる。完全に左折し、範囲は約 7 メートルになります。



HC SR501 Time Delay Adjustment

時間遅延が完了してから 3 秒後に消灯 - 重要

このデバイスの出力は、遅延が完了した後、約 3 秒間 Low (または Off) になります。つまり、この 3 秒間にすべての動き検出がブロックされます。

For Example:

あなたがシングルトリガーモードで、あなたの時間遅延が 5 秒に設定されているとします。PIR はモーションを検出し、5 秒間ハイに設定します。

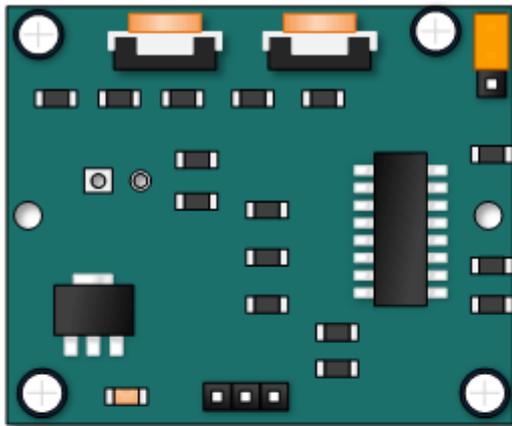
5 秒後、PIR は約 3 秒間その出力を低く設定します。3 秒間、PIR は動きを検出しません。

3 秒後、PIR は動きを再び検出し、検出された動きは再び出力を高く設定します。

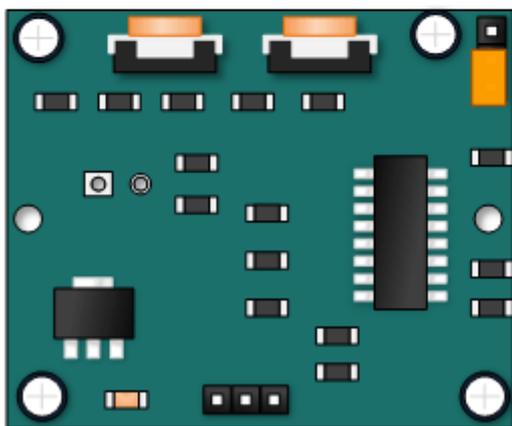
Trigger Mode Selection Jumper

トリガーマード選択ジャンパーでは、単一トリガーとリピート可能トリガーを選択できます。このジャンパ設定の影響は、時間遅延の開始時期を判断することです。

- **SINGLE TRIGGER** –時間遅延は、動きが最初に検出されたときに直ちに開始されます。
- **REPEATABLE TRIGGER** –検出された各モーションは時間遅延をリセットします。従って、時間遅延は、最後に検出された動きから始まる。



Single Trigger Mode – Time Delay is started immediately upon detecting motion. Continued detection is blocked



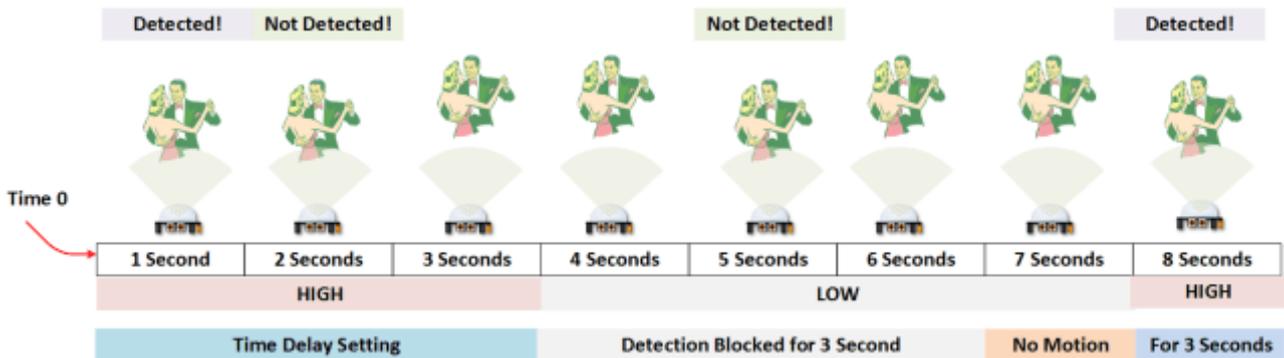
Repeatable Trigger Mode – Time Delay is re-started every time motion is detected.

HC-SR501 Dance Floor Application Examples

ダンサーがダンスする場所に基づいてダンスフロアの照明を制御したいとします。タイムディレイとトリガモードがどのように作用するかを理解することは、あなたが望む方法で照明を制御するために必要です。

Example One

この最初の例では、時間遅延が3秒に設定され、トリガーモードが1に設定されています。下の図のように、モーションは常に検出されるわけではありません。実際、動きが検出されない約6秒の期間があります。画像をクリックすると気軽に拡大できます。



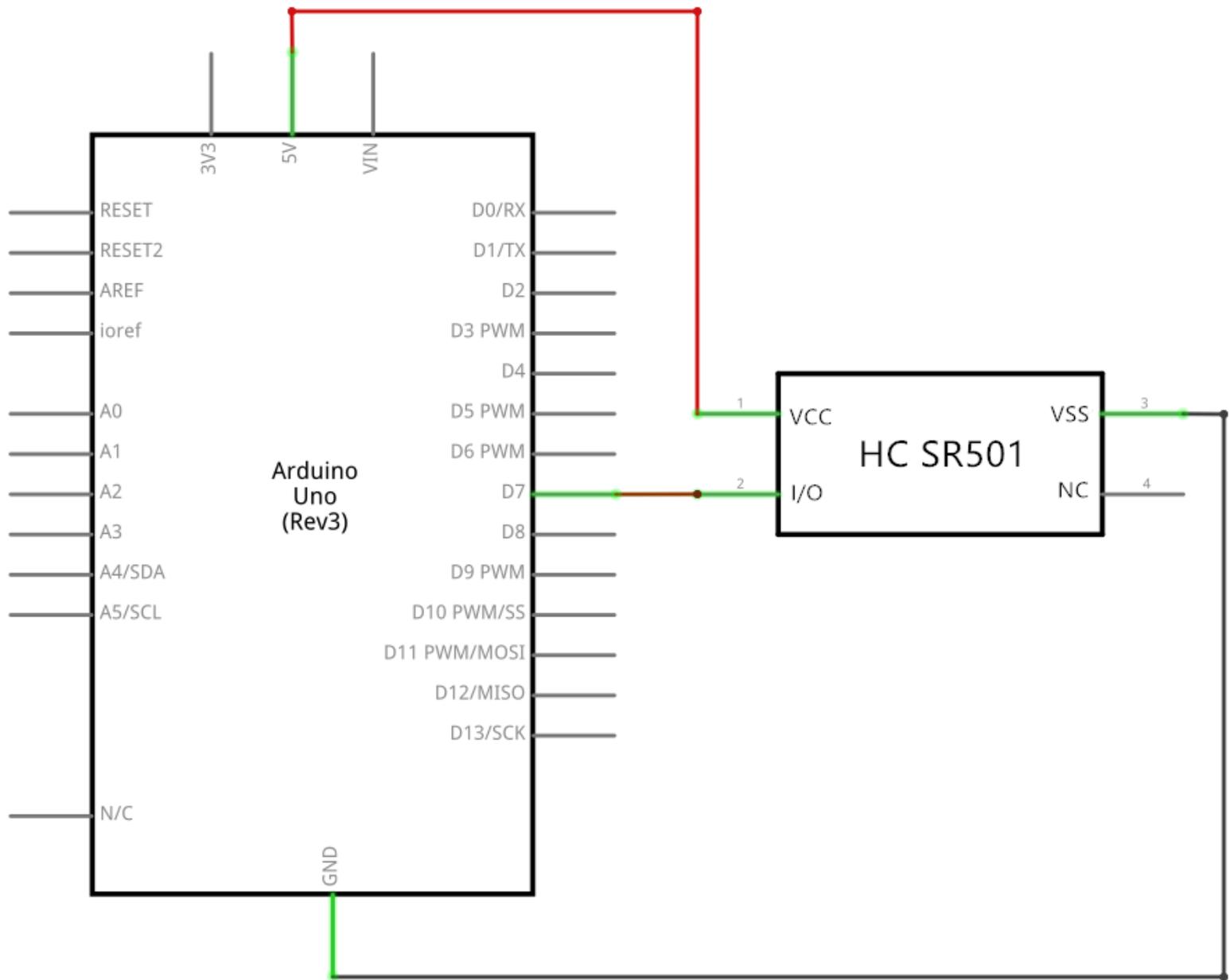
Example Two

次の例では、時間遅延はまだ3秒で、トリガーは繰り返し可能に設定されています。以下の図では、時間遅延期間が再開されたことがわかります。しかし、3秒後には、3秒間検出がブロックされます。

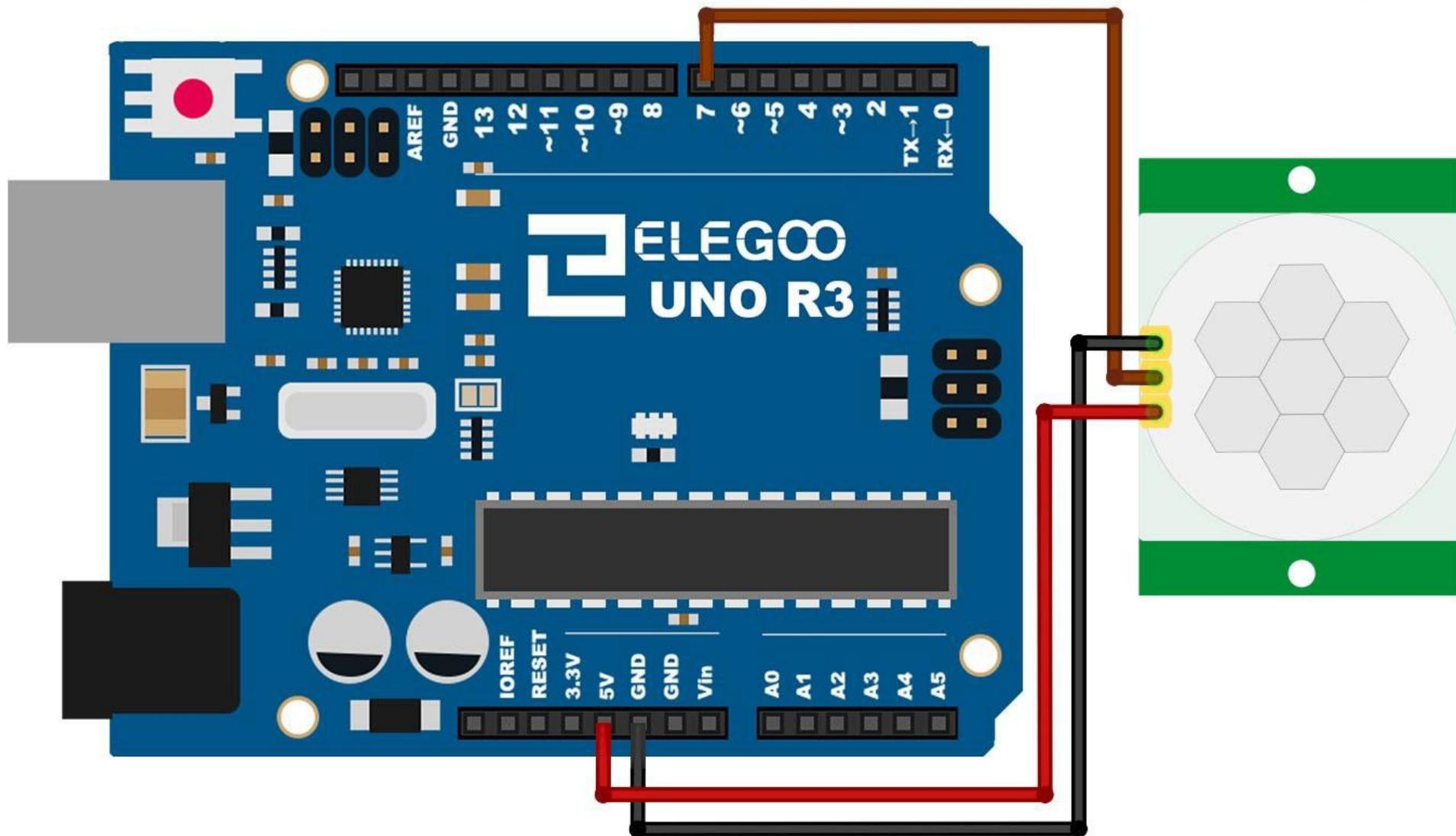
前述のように、3秒のブロック期間をいくつかのクリエイティブコードで上書きすることはできませんが、それを考慮する必要があります。あなたが使用するエレクトロニクスの中には、オンとオフの揺れが好きではないかもしれません。3秒間は、バックアップを開始する前に少し休憩を取ることができます。



Connection Schematic



Wiring diagram



PIR センサーをマイクロコントローラーに接続するのは本当に簡単です。 PIR はデジタル出力として機能するので、ピンが高い (検出された) か低い (検出されない) かを聞くだけです。

それはあなたが再トリガする必要がある可能性が高いので、ジャンパを H の位置に置いてください! 5V で PIR に給電し、グラウンドをグラウンドに接続します。 次に、出力をデジタルピンに接続します。 この例では、ピン 7 を使用します。

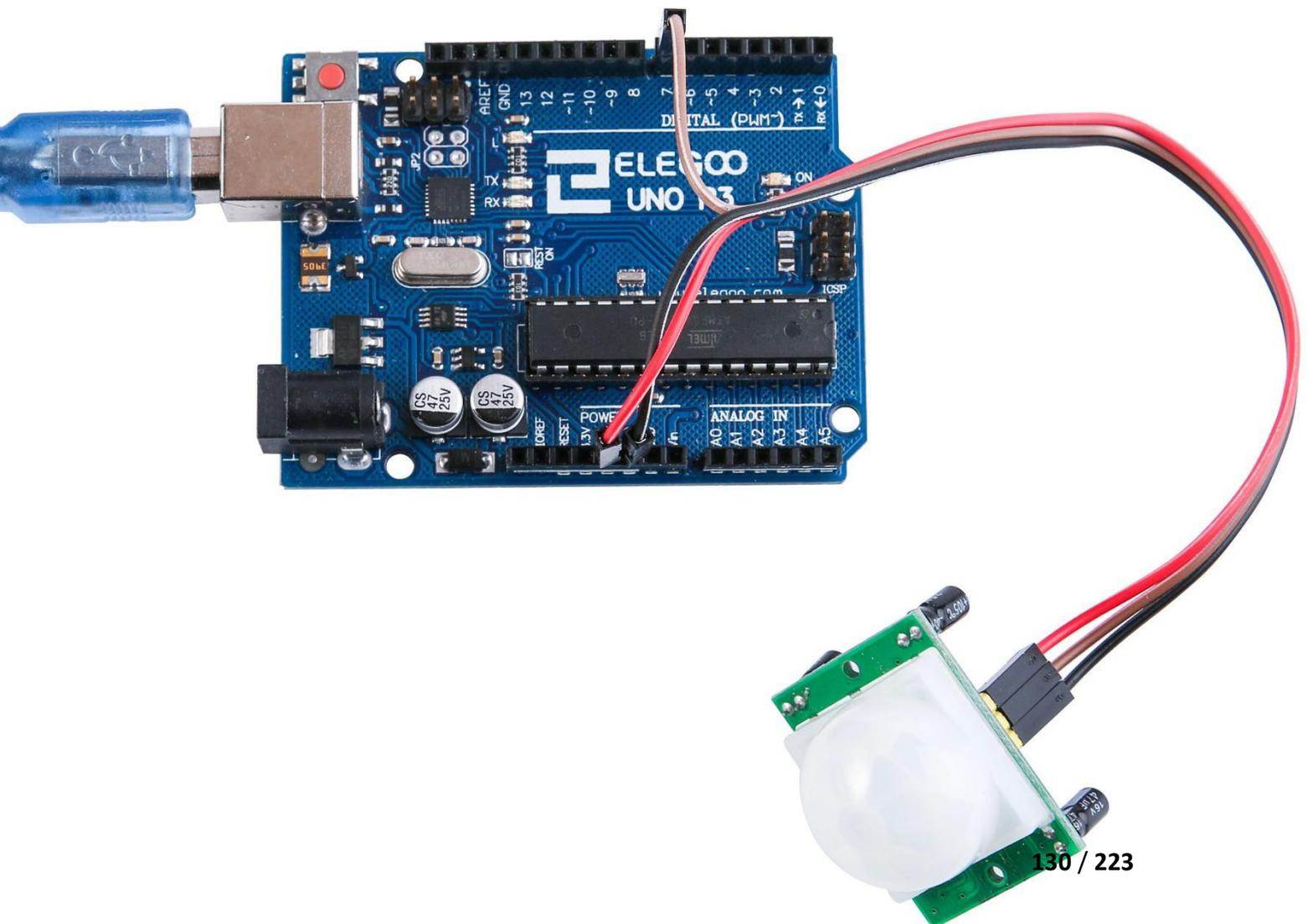
Code

配線後、コードフォルダのレッスン 17 HC-SR501 PIR センサ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。

スケッチは動きが検出されるたびにピン 13 に接続された Arduino LED を点灯させるだけです。

あなたが開発するどのアプリケーションでも、1 分間の初期設定に注意し、何とか処理してください。

Example picture



Lesson 18 水位検出検知器

概要

このレッスンでは、水位検出センサーモジュールの使用方法を学習します。このモジュールは水の深さを知覚することができ、コア構成要素はトランジスタといくつかの櫛状 PCB 配線からなる増幅回路である。水の中に置かれると、これらの経路は水の深さの変化と共に変化する抵抗を提示する。水の深さの信号は電気信号に変換され、UNO R3 の ADC 機能によって水深の変化を知ることができます。

必要な構成部品:

- (1) x Elegoo Uno R3
- (3) x F-M wires (Female to Male DuPont wires)
- (1) x Water lever detection sensor module

部品の紹介



Water sensor:

水検出器は水の検出のために設計されており、降水量、水位、液漏れの検出に広く使用できます。レングは主に 3 つの部分で構成されています。電子レングのコネクタ、1MΩの抵抗器、数本の裸の導線です。

このセンサーは一連の露出したトレースを地面に接続することで機能します。接地されたトレースの間にインターレースされるのは、センスのトレースです。

センサトレースには 1MΩの弱いプルアップ抵抗があります。抵抗器は、水滴がセンサトレースを接地トレースに短絡するまで、センサトレース値を高くします。信じられないかもしれませんが、この回路は UNO R3 ボードのデジタル I/O ピンで動作するか、アナログピンと一緒に使用して接地トレースとセンサトレースの間の水誘導接触量を検出できます。

このアイテムは、水滴/水のサイズを測定するために一連の露出した平行ワイヤステッチで水位を判断することができます。それは簡単にアナログ信号に水のサイズを変更することができ、出力アナログ値は、プログラム機能で直接使用することができます水位警報の機能を達成します。

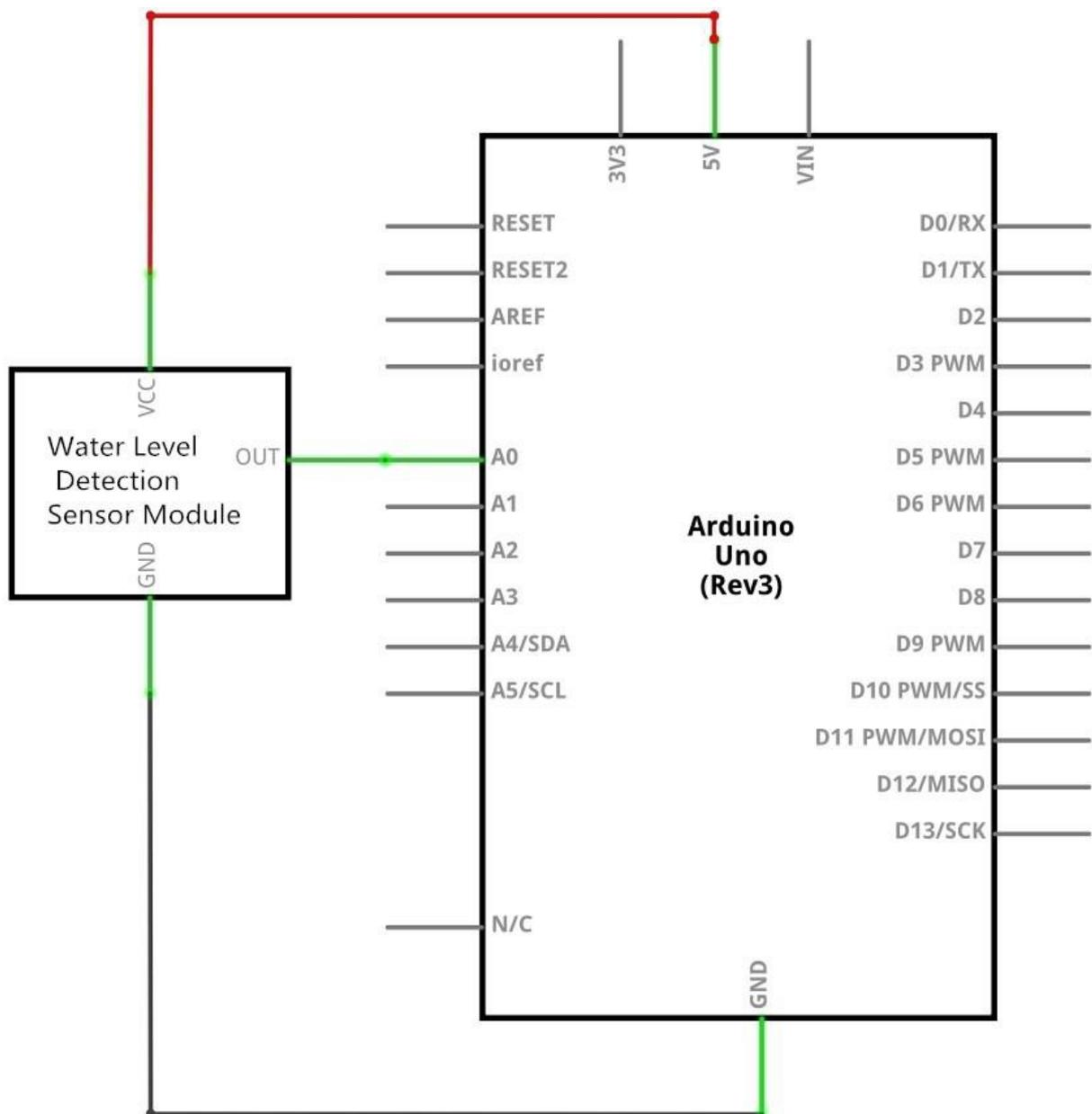
低消費電力で高感度です。

特徴:

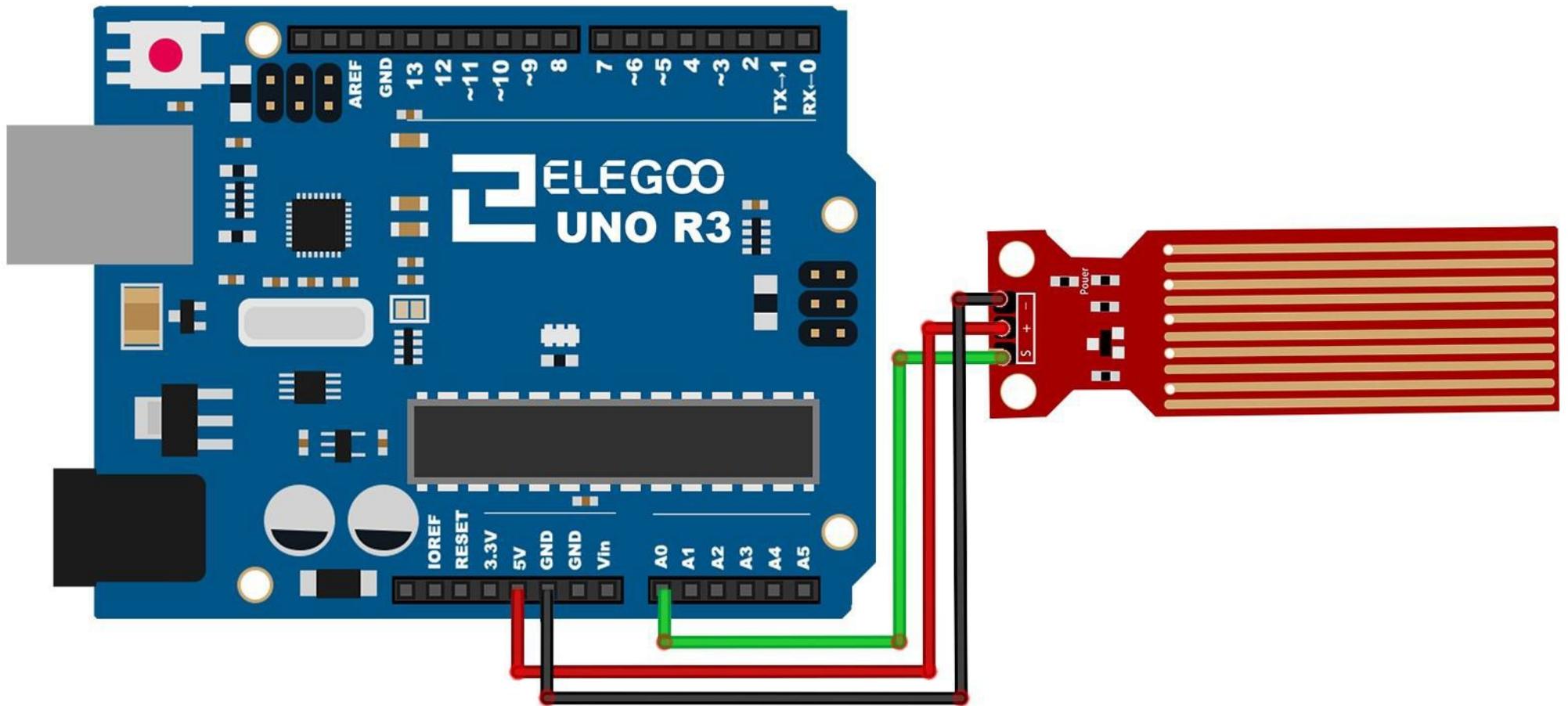
- 1、 Working voltage: 5V
- 2、 Working Current: <20ma
- 3、 Interface: Analog
- 4、 Width of detection: 40mm×16mm
- 5、 Working Temperature: 10°C~30°C
- 6、 Output voltage signal: 0~4.2V

Connection

Schematic



Wiring diagram

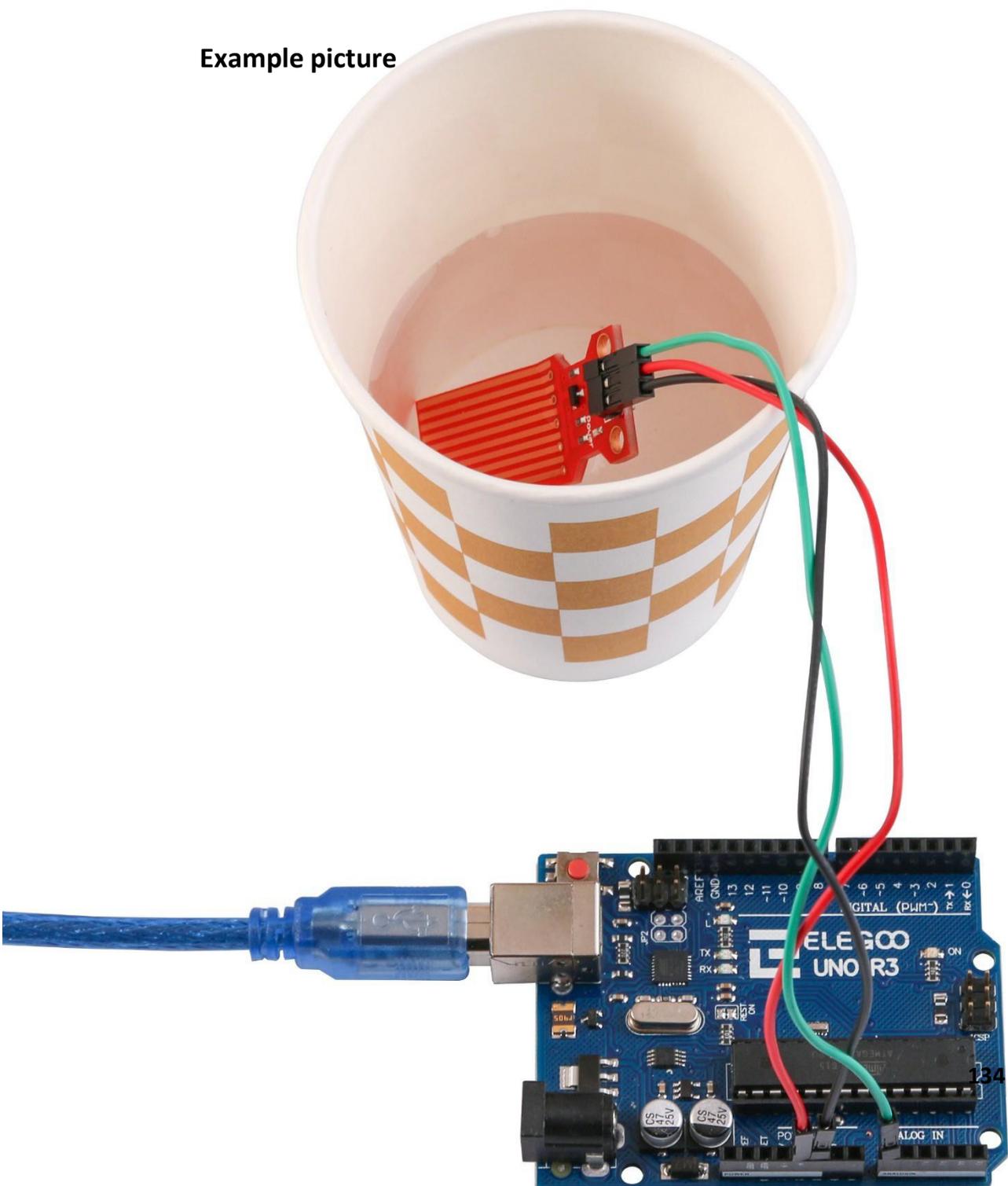


配線のヒント：電源 (+) は UNO R3 ボードの 5V に接続され、グランド電極 (-) は GND に接続されています。信号出力 (S) は UNO R3 ボードのアナログ信号を入力する機能を持つポート (A0 ~A5) に接続されていますが、ランダムなものは OK ですが、ルーチンと同じデモコードを定義する必要があります。

Code

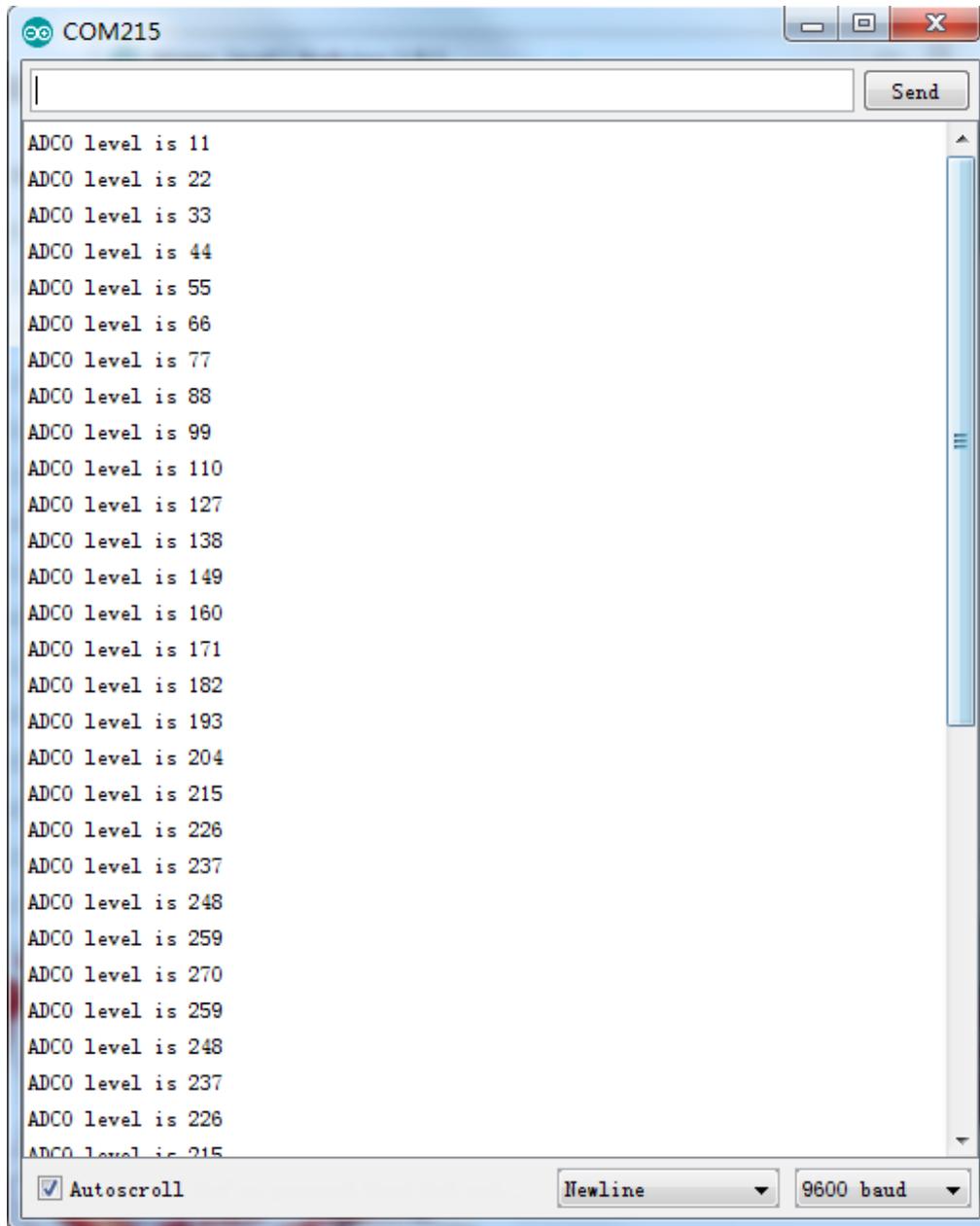
結線後、プログラムをコードフォルダ - レッスン 18 水位検知センサーモジュールで開き、**UPLOAD** をクリックしてプログラムをアップロードしてください。エラーがある場合のプログラムアップロードの詳細については、[レッスン 2](#) を参照してください。

Example picture



モニタを開くと、以下のようなデータが表示されます:

[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 19 実時間モジュール

概要

このレッスンでは、年、月、日、時、分、秒、および週を表示するクロックモジュール DS3231 の使用方法を学習します。サポートはバックアップバッテリーリトリクルチャージャを介して行われ、3本のデータケーブルのみで UNO に接続されていない限り使用できます。

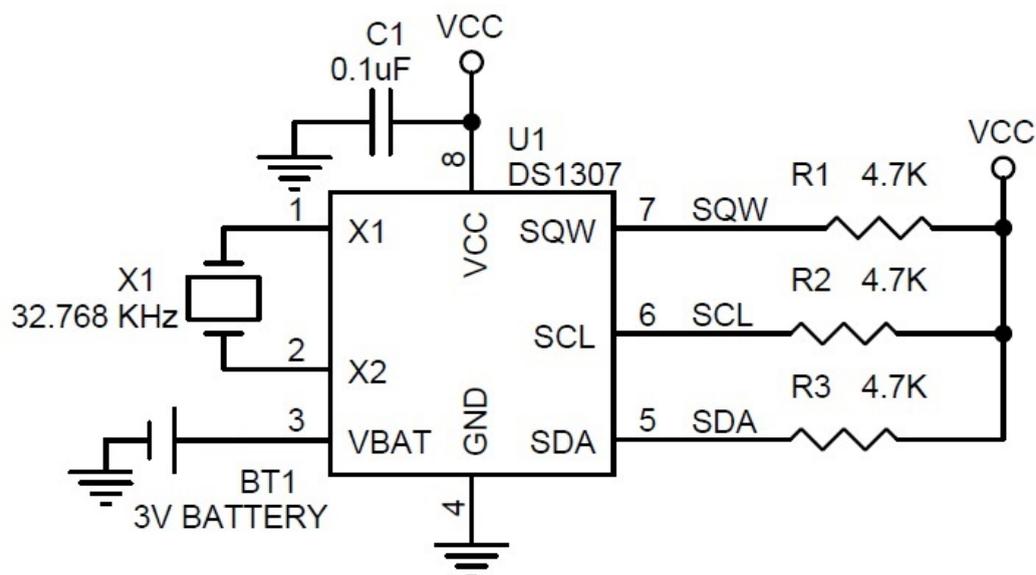
必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x DS3231 RTC module
- (4) x F-M wires (Female to Male DuPont wires)

部品の紹介

DS3231

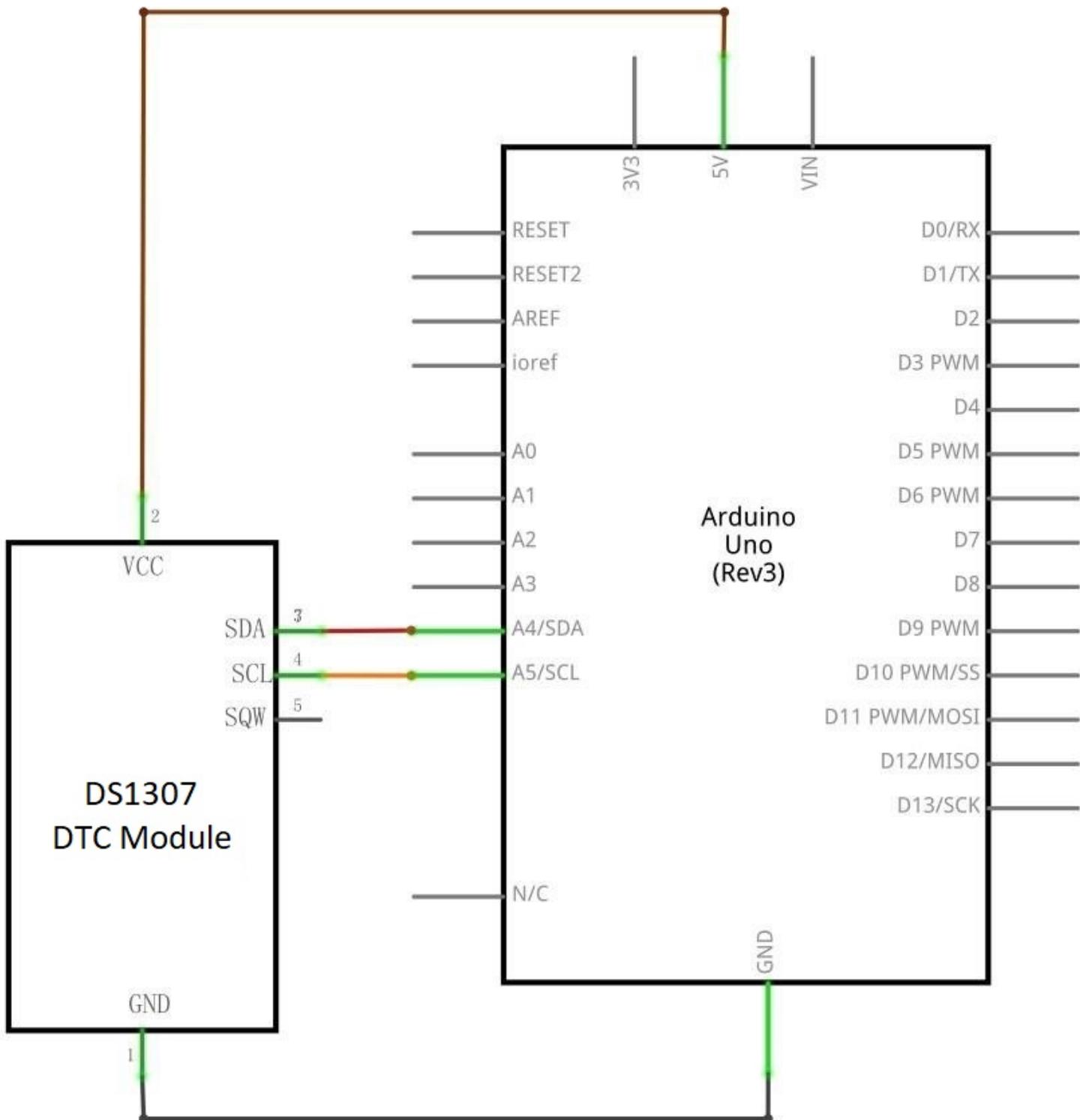
DS3231 はシンプルな時間管理チップです。内蔵バッテリーを備えているため、電源を切っても時計は時間を維持できます。



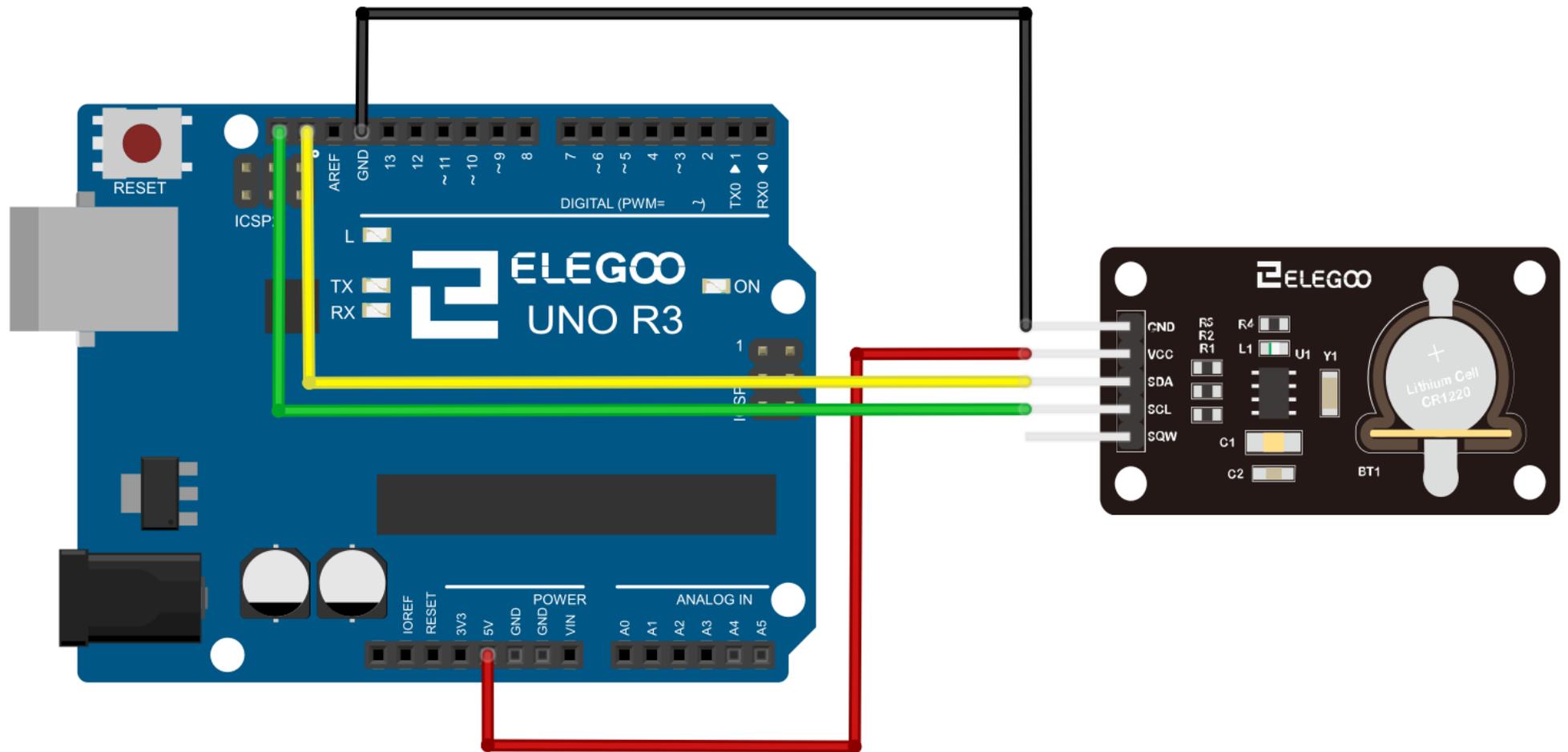
注意: キットにモジュールの左側のバージョンがある場合でも、心配しないでください。その機能とピン名は新しいバージョンと同じです。配線図に従って、下のチュートリアルスケッチを実行してください。

Connection

Schematic



Wiring diagram



次の画像に従って設定します。

32K ピンと SQW ピンは無視してください。 あなたはそれらを必要としません。 SCL ピンを UNO R3 ボードの SCL ポートに接続し、SDA ピンを SDA ポートに接続します。 VCC ピンは 5V ポートに接続され、GND は GND ポートに接続されます。

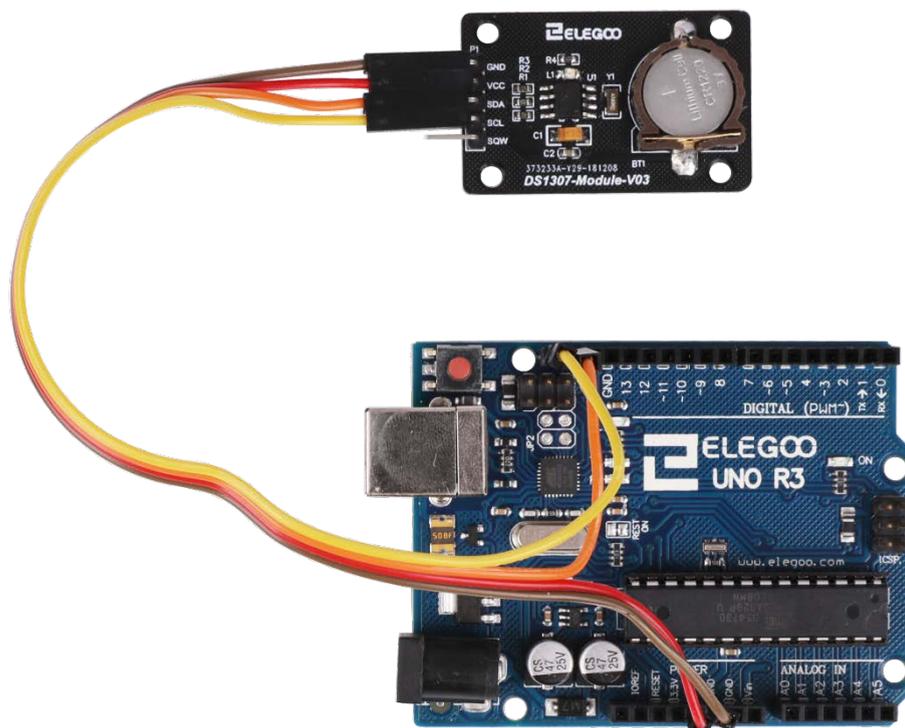
Code

配線後、コードフォルダのレッスン 19 リアルタイムクロックモジュール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、<DS3231>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。

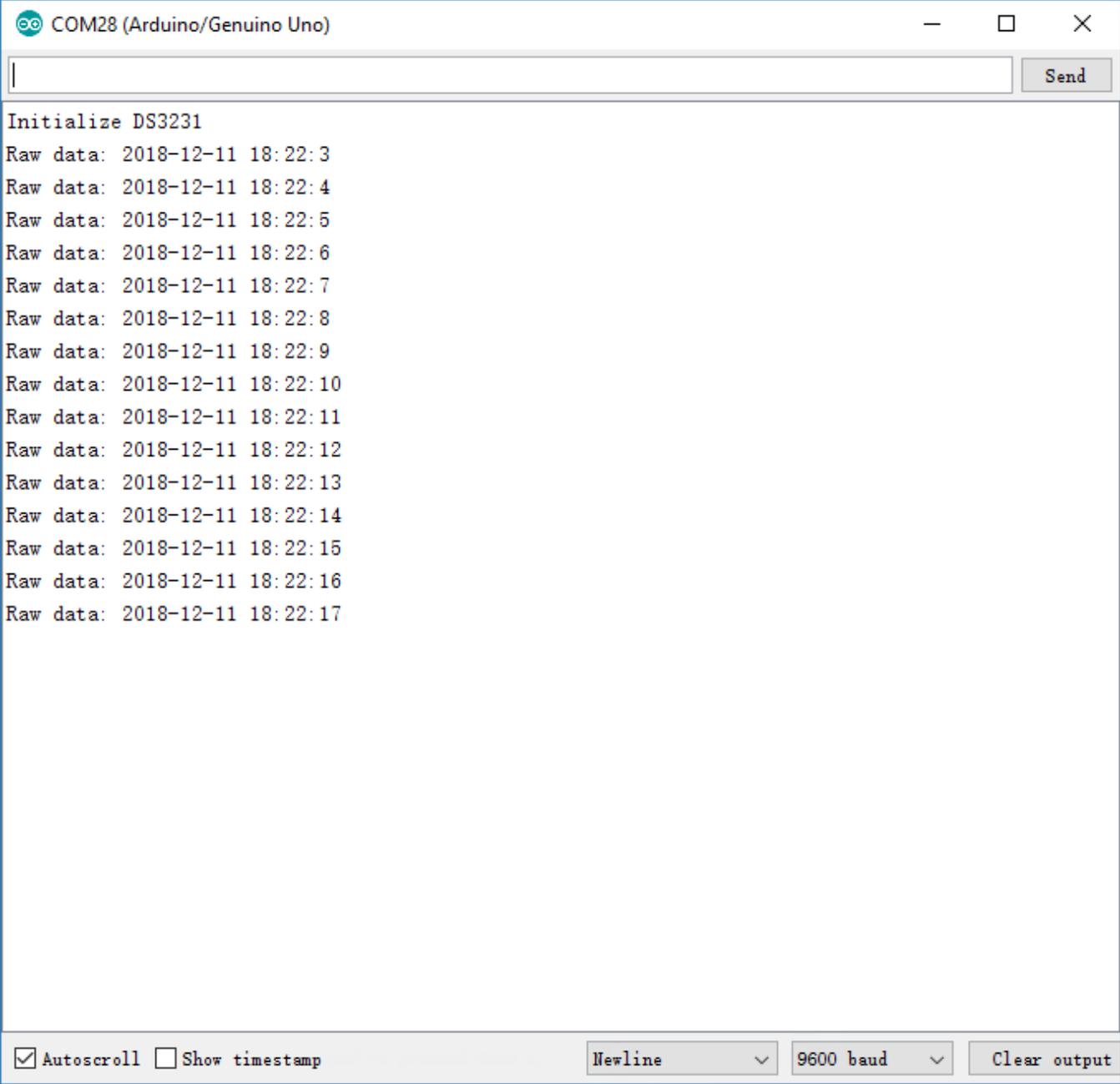
ライブラリファイルのロードの詳細については、[レッスン 1](#) を参照してください。

Example picture



モニタを開くと、モジュールが次のように時刻を読み取ることができます:

[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。



```
COM28 (Arduino/Genuino Uno)
Initialize DS3231
Raw data: 2018-12-11 18:22:3
Raw data: 2018-12-11 18:22:4
Raw data: 2018-12-11 18:22:5
Raw data: 2018-12-11 18:22:6
Raw data: 2018-12-11 18:22:7
Raw data: 2018-12-11 18:22:8
Raw data: 2018-12-11 18:22:9
Raw data: 2018-12-11 18:22:10
Raw data: 2018-12-11 18:22:11
Raw data: 2018-12-11 18:22:12
Raw data: 2018-12-11 18:22:13
Raw data: 2018-12-11 18:22:14
Raw data: 2018-12-11 18:22:15
Raw data: 2018-12-11 18:22:16
Raw data: 2018-12-11 18:22:17
```

Autoscroll Show timestamp Newline 9600 baud Clear output

Lesson 20 Sound Sensor Module

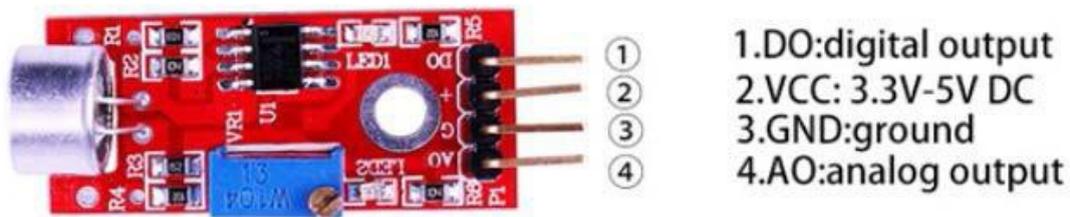
概要

このレッスンでは、サウンドセンサモジュールの使用方法を学習します。このモジュールには2つの出力があります。

AO: アナログ出力、マイクのリアルタイム出力電圧信号

DO: 音の強さがある閾値に達すると、出力は高レベルまたは低レベルの信号になります。閾値感度は、ポテンショメータを調整することによって達成することができる。

マイクがあなたの音声を正常に検出できるようにするには、モジュール上の青色の正確なポテンショメータを回して感度を変更してみてください。その正確さを考えると、少なくとも10のサークルが必要です。



必要な構成部品:

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

部品の紹介

Microphone

トランスデューサは、エネルギーをある形から他の形に変換する装置です。マイクは、音エネルギーを電気信号に変換するトランスデューサです。それは話者の反対に働く。マイクはさまざまな形とサイズで利用できます。アプリケーションによっては、マイクロフォンが異なる技術を使用して音を電気信号に変換する場合があります。ここでは、携帯電話、ノートパソコンなどで広く使われているエレクトレットコンデンサーマイクについて議論します。

その名前が示唆するように、エレクトレットコンデンサーマイクロホン平行平板

可変キャパシタンスの原理に基づいて動作します。それは 2 つのプレートから構成され、一方は固定されており（バックプレートと呼ばれています）、他方のプレートはダイヤフラムと呼ばれ、小さな隙間があります。電位はプレートを充電する。音がダイヤフラムにぶつくと、振動が始まり、プレート間の静電容量が変化し、可変電流が流れる。

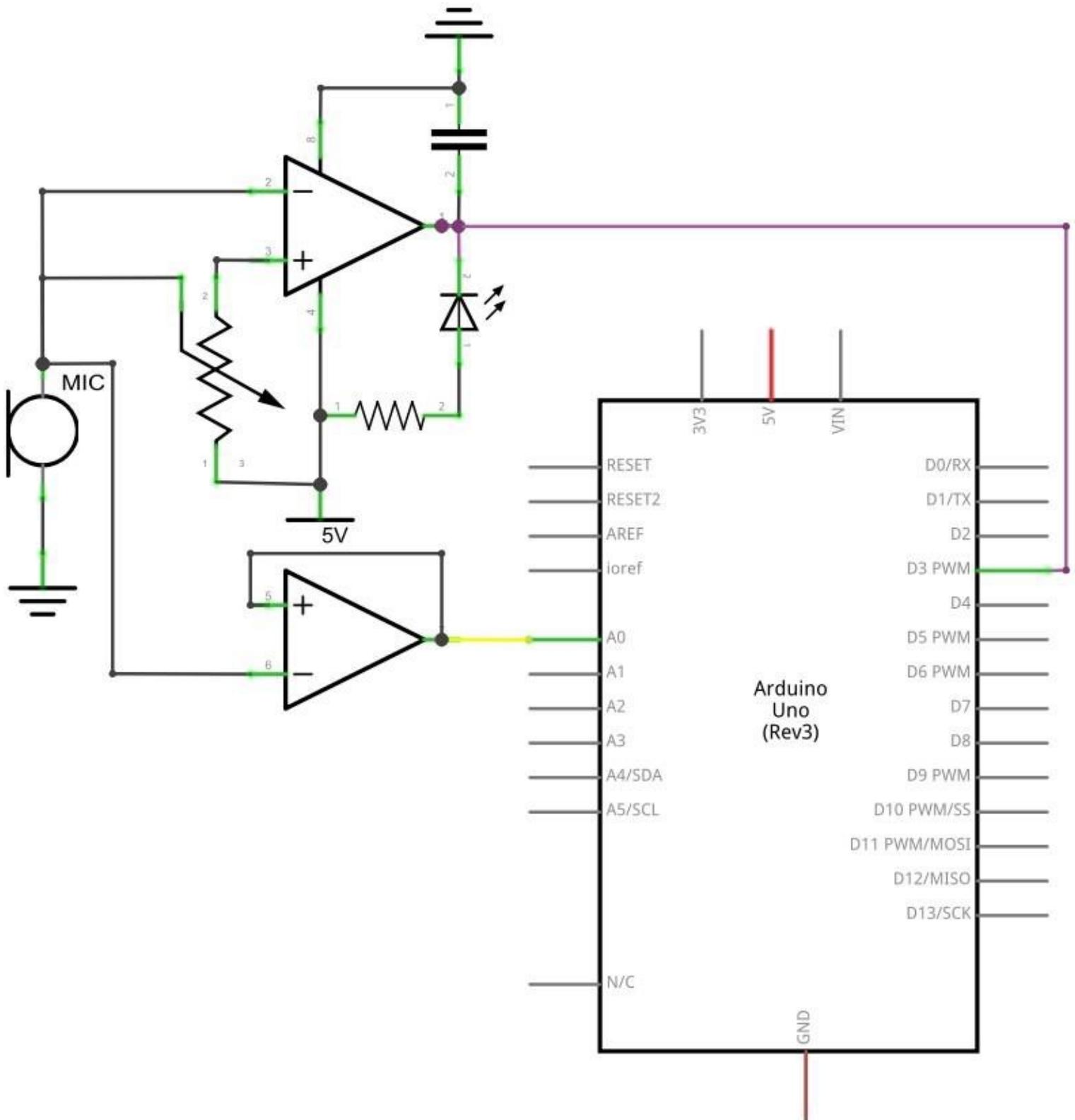


これらのマイクロホンは、マイナーな音や空気振動を検出するために電子回路に広く使用されており、これらの振動はその後の使用のために電気信号に変換される。上記の図に示されている 2 本の脚は、回路との電気接続に使用されます。

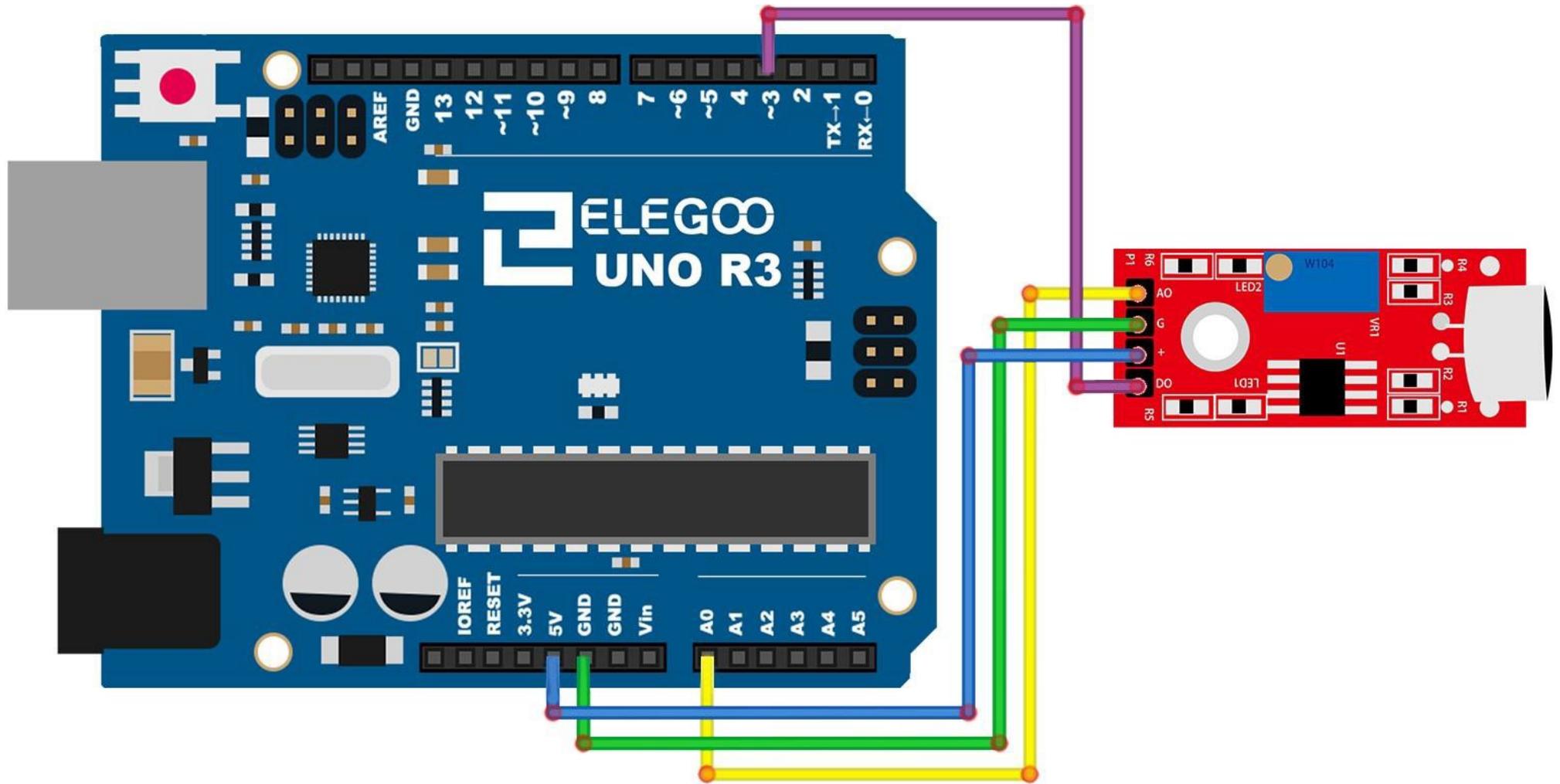


固体の導電性の金属体が、マイクロフォンの様々な部分をカプセル化する。上面は、接着剤の助けを借りて多孔質材料で覆われている。ダスト粒子のフィルターとして機能します。音響信号/空気振動は、多孔質材料を通過し、上の図に示された穴を通過してダイヤフラム上に落ちる。

Connection Schematic



Wiring diagram

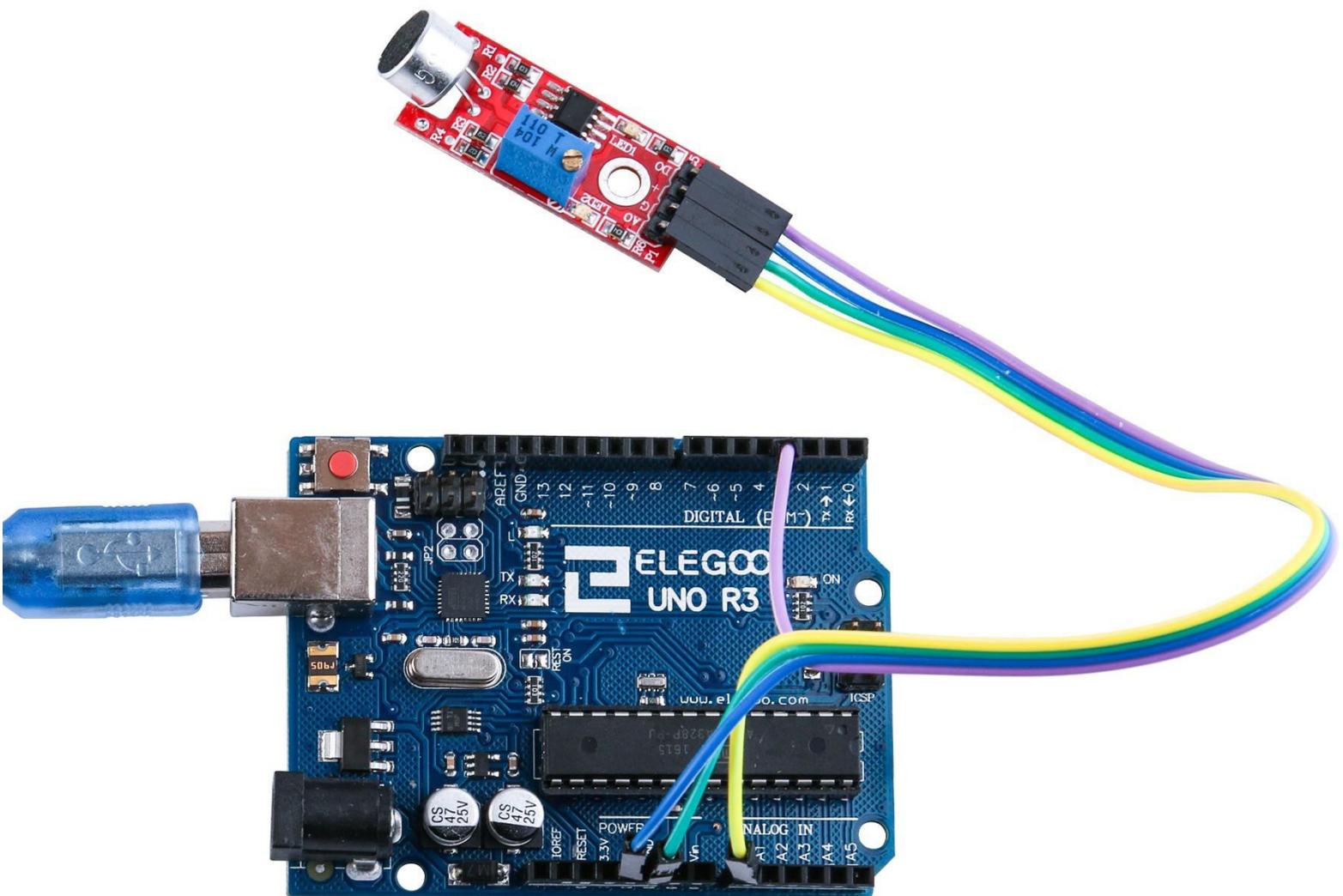


The code

配線後、プログラムをコードフォルダ - レッスン 20 サウンドセンサーモジュールで開き、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。

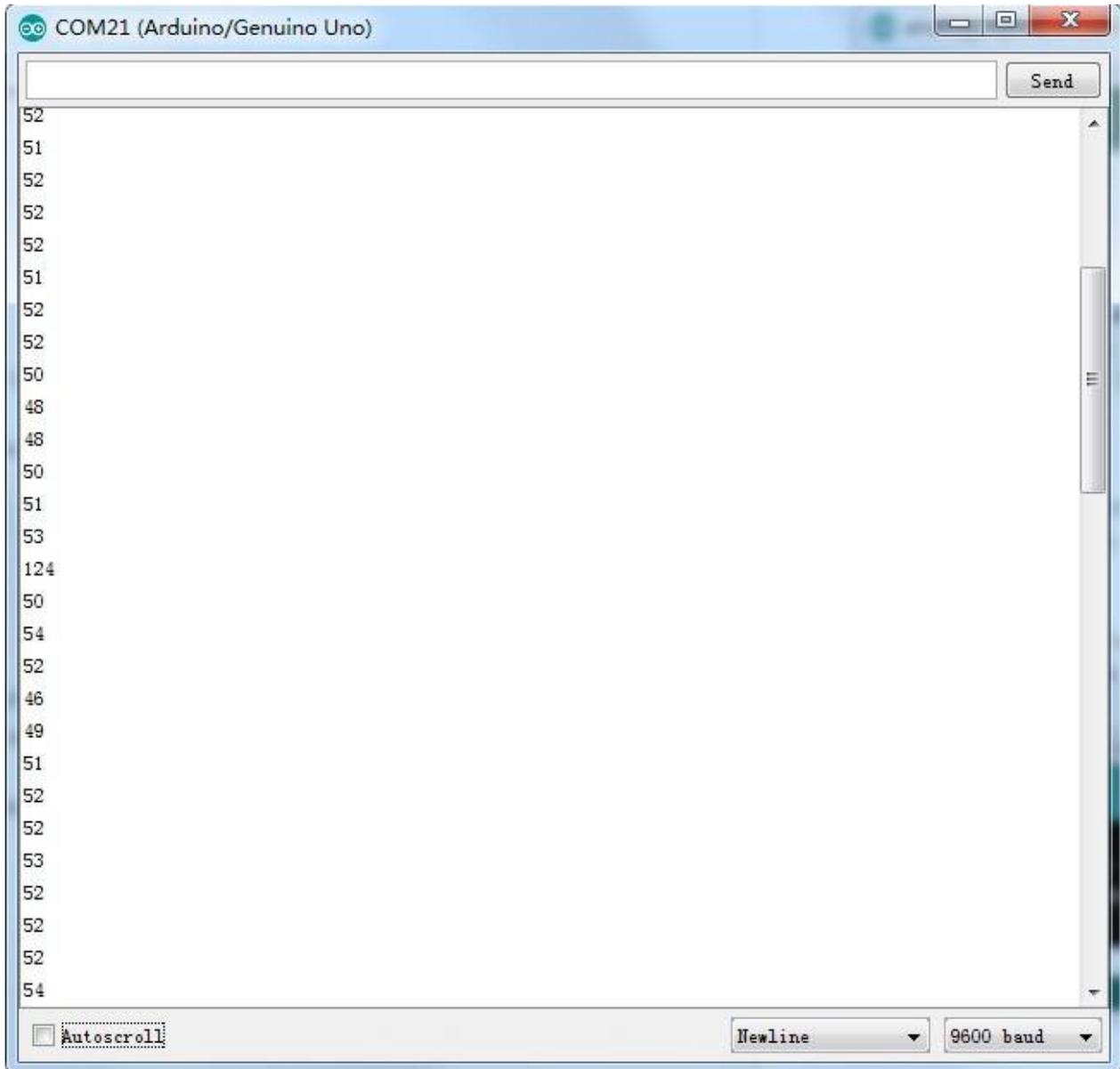
このモジュールは、2 つの信号出力モードを提供します。2 つのコード: **digital_signal_output** と **analog_signal_output** があります。 **digital_signal_output** のコードは、音声 that 一定の値に達すると動作し、デジタル信号をトリガーし、Arduino の dig #11 ピンがハイレベルを出力し、インジケータ L が同時に点灯します。 このトリガ値は、上述した感度調整方法に従って変更することができます。 **analog_signal_output** のコードは、モジュールのアナログ値を読み取り、シリアルモニタに直接表示します。同様に、この値は、上記の感度調整方法に従って変更することもできます。

Example picture



モニタを開くと、以下のようなデータが表示されます:

[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。



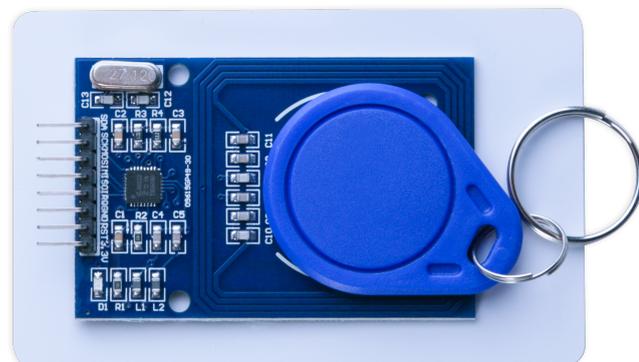
Lesson 21 RC522 RFID Module

概要

このレッスンでは、UNO R3にRC522 RFIDリーダーモジュールを適用する方法を学習します。このモジュールは、Arduino、Raspberry Pi、ビーグルボードなどのコントローラーと通信するために、SPI (Serial Peripheral Interface) バスを使用します。

必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x RC522 RFID module
- (7) x F-M wires (Female to Male DuPont wires)



部品の紹介

RC522

MFRC522 は、非接触通信用の高集積リーダ/ライタです。

13.56MHz。MFRC522 リーダーは ISO 14443A /MIFARE®モードをサポートしています。

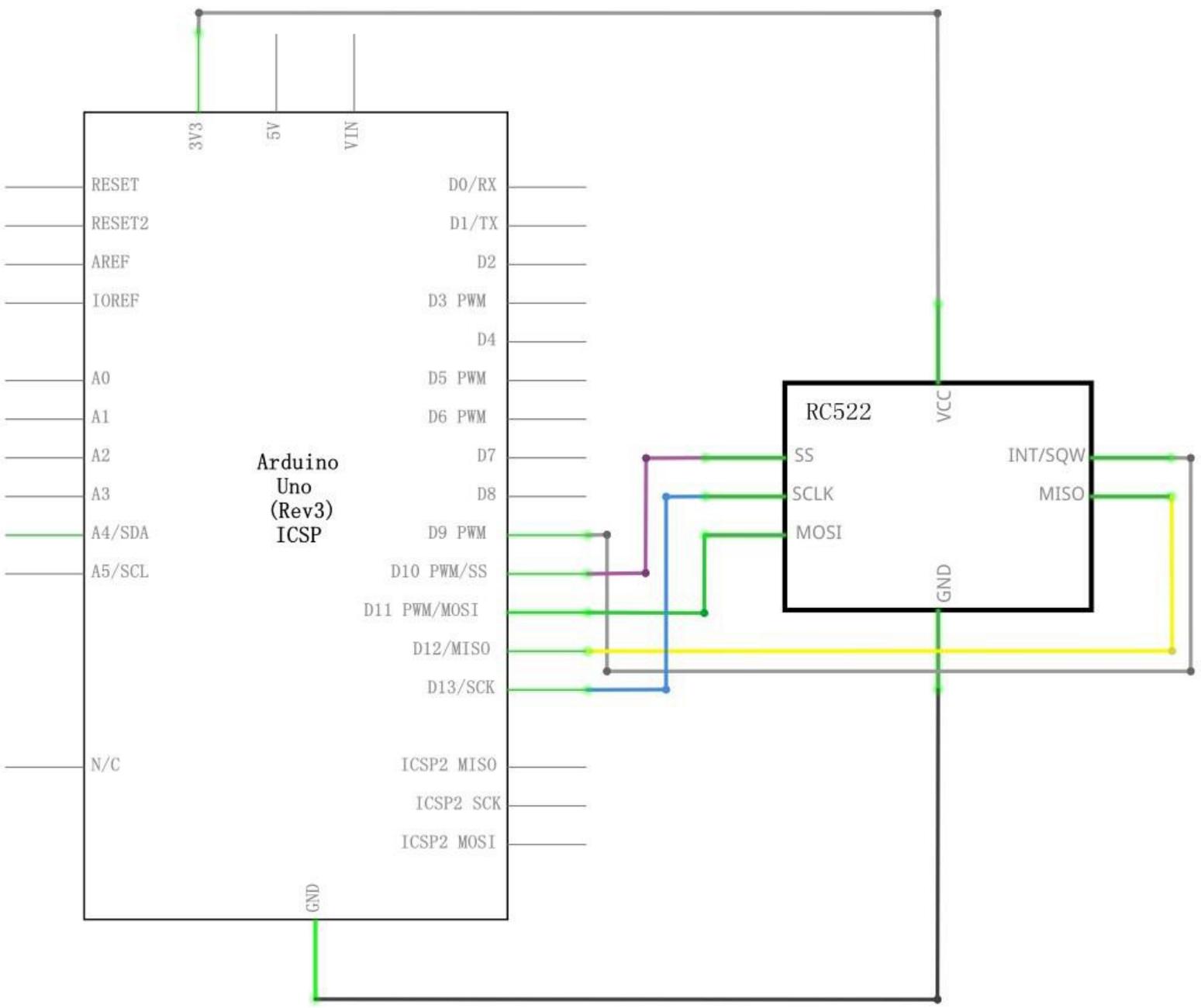
MFRC522 の内部送信部は、ISO / IEC 14443A /MIFARE®カードとトランスポンダと通信するように設計されたリーダライタアンテナを、追加のアクティブ回路なしで駆動することができます。受信機部分は、ISO / IEC14443A /MIFARE®互換カードおよびトランスポンダからの信号用の復調および復号回路の堅牢で効率的な実装を提供します。デジタル部分は完全な ISO / IEC 14443A フレーミングとエラー検出 (パリティ & CRC) を処理します。MFRC522 は MIFARE®Classic (例: MIFARE®Standard) 製品をサポートしています。MFRC522 は、双方向で最高 848kbit / s の MIFARE®高速転送速度を使用した非接触通信をサポートしています。

さまざまなホストインターフェイスが実装されています。インターフェイスとは接続や操作方法のこと:

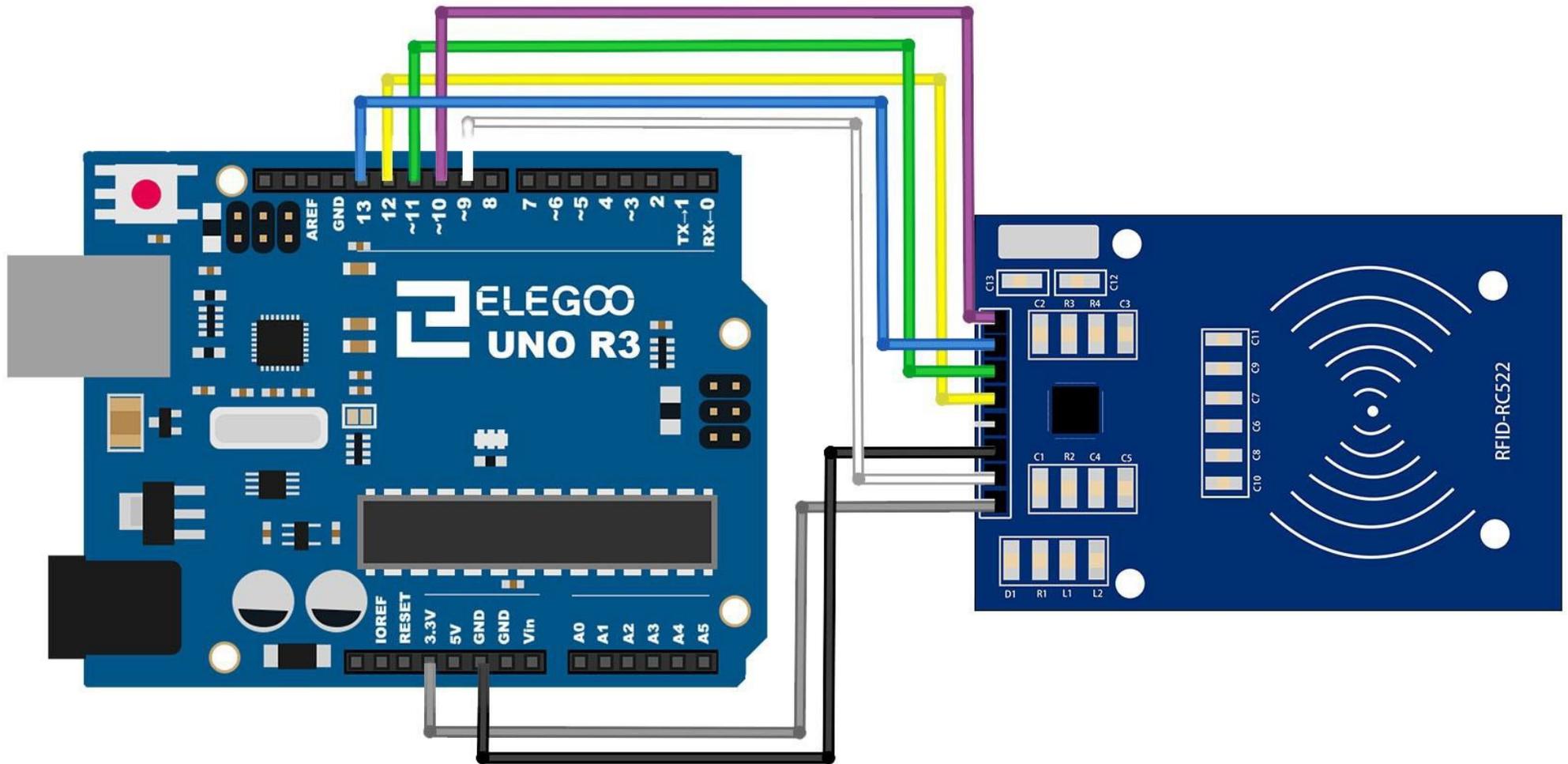
- SPI interface
- Serial UART (パッド電圧供給に応じた電圧レベルの RS232 に似ています)
- I2C interface.

下の図は、MFRC522 との相補的なアンテナ接続を使用した一般的な回路図を示しています。

Connection Schematic



Wiring diagram



Code

配線後、コードフォルダのレッスン 21 RC522 RFID Module 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、<rfid>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。

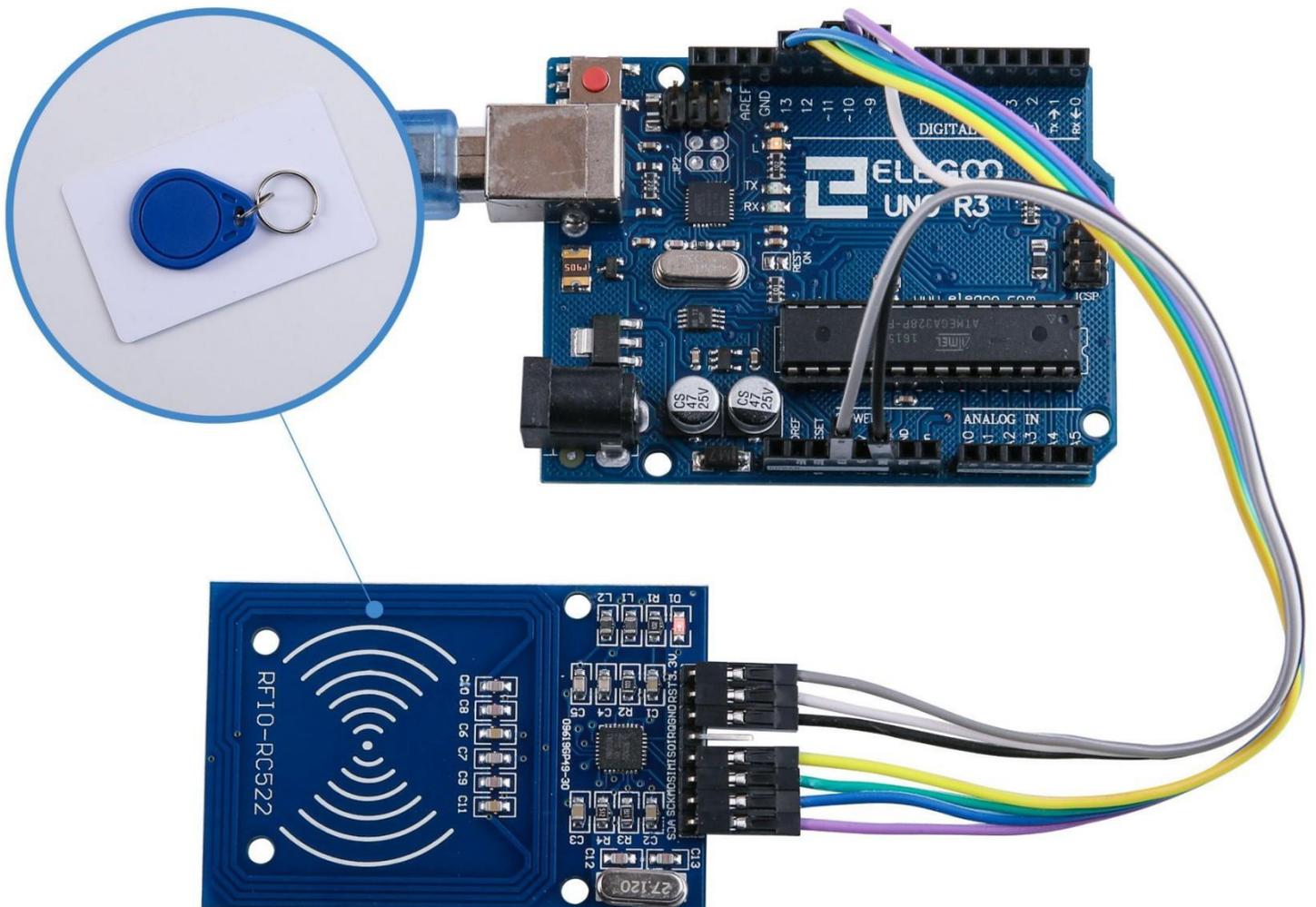
	MFRC522	Arduino	Arduino	Arduino	Arduino	Arduino
	Reader/PCD	Uno	Mega	Nano v3	Leonardo/Micro	Pro Micro
Signal	Pin	Pin	Pin	Pin	Pin	Pin
RST/Reset	RST	9	5	D9	RESET/ICSP-5	RST
SPI SS	SDA (SS)	10	53	D10	10	10
SPI MOSI	MOSI	11 / ICSP-4	51	D11	ICSP-4	16
SPI MISO	MISO	12 / ICSP-1	50	D12	ICSP-1	14
SPI SCK	SCK	13 / ICSP-3	52	D13	ICSP-3	15

```
#define RST_PIN 9 // Configurable, see typical pin layout above
```

```
#define SS_PIN 10 // Configurable, see typical pin layout above
```

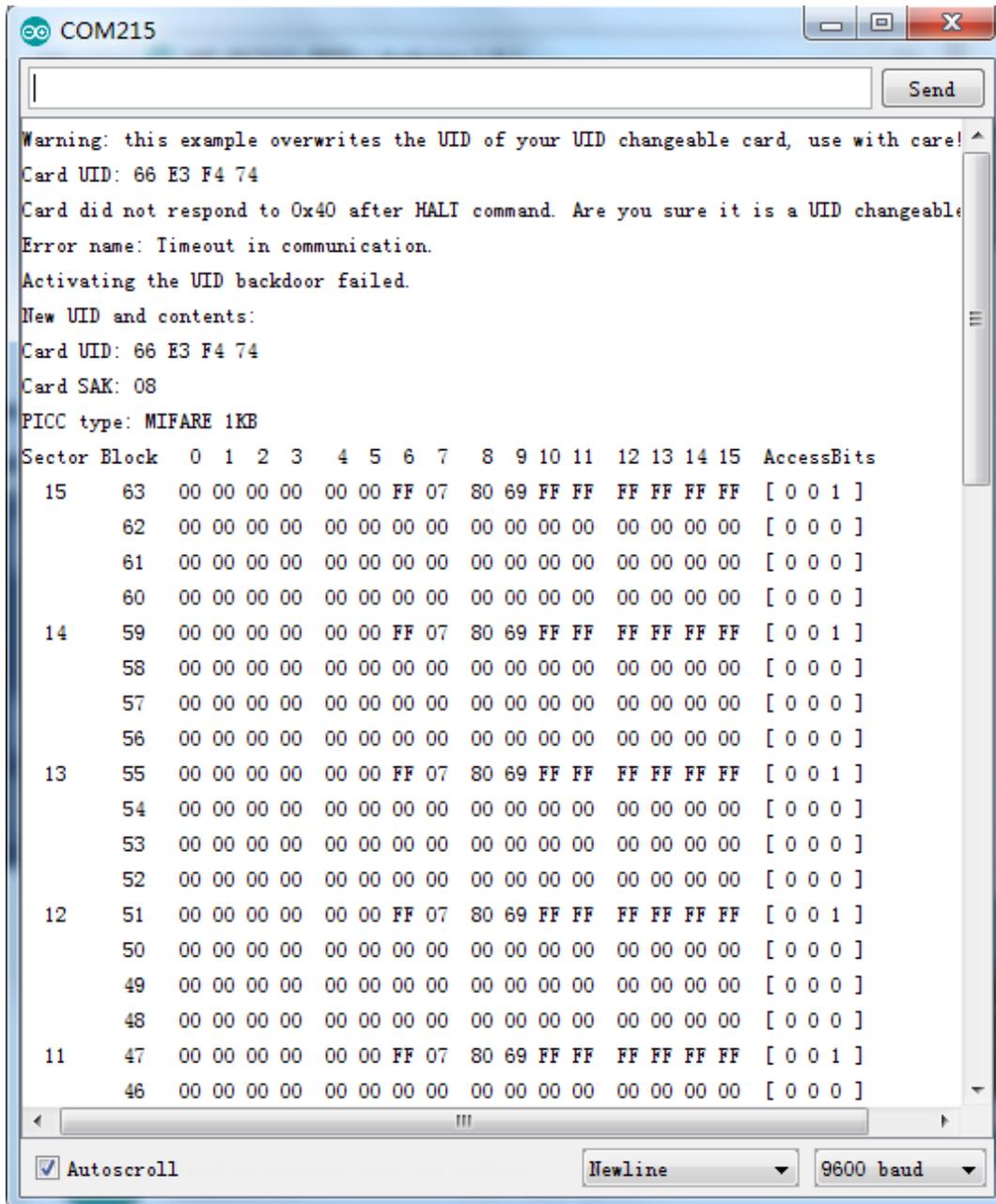
SPI ピンの位置はチップによって異なりますので、機能を少し変更する必要があります。

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。



Warning: this example overwrites the UID of your UID changeable card, use with care!
Card UID: 66 E3 F4 74
Card did not respond to 0x40 after HALI command. Are you sure it is a UID changeable?
Error name: Timeout in communication.
Activating the UID backdoor failed.
New UID and contents:
Card UID: 66 E3 F4 74
Card SAK: 08
PICC type: MIFARE 1KB

Sector	Block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AccessBits
15	63	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]
	62	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	61	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
14	59	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]
	58	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	57	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	56	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
13	55	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]
	54	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	53	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	52	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
12	51	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]
	50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	49	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
	48	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]
11	47	00	00	00	00	00	00	FF	07	80	69	FF	FF	FF	FF	FF	FF	[0 0 1]
	46	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	[0 0 0]

Autoscroll Newline 9600 baud

Lesson 22 LCD Display

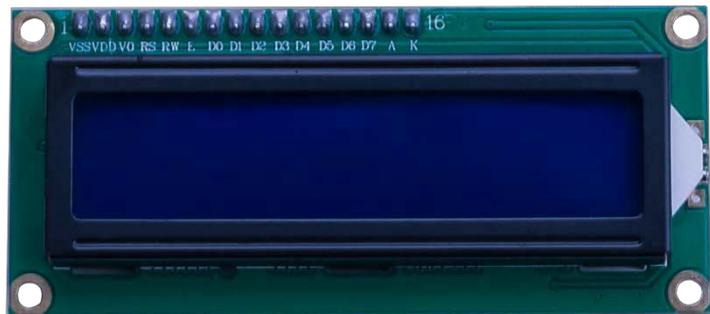
概要

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 ライブラリ用の 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やるかの指示を参照する) を選択できます。

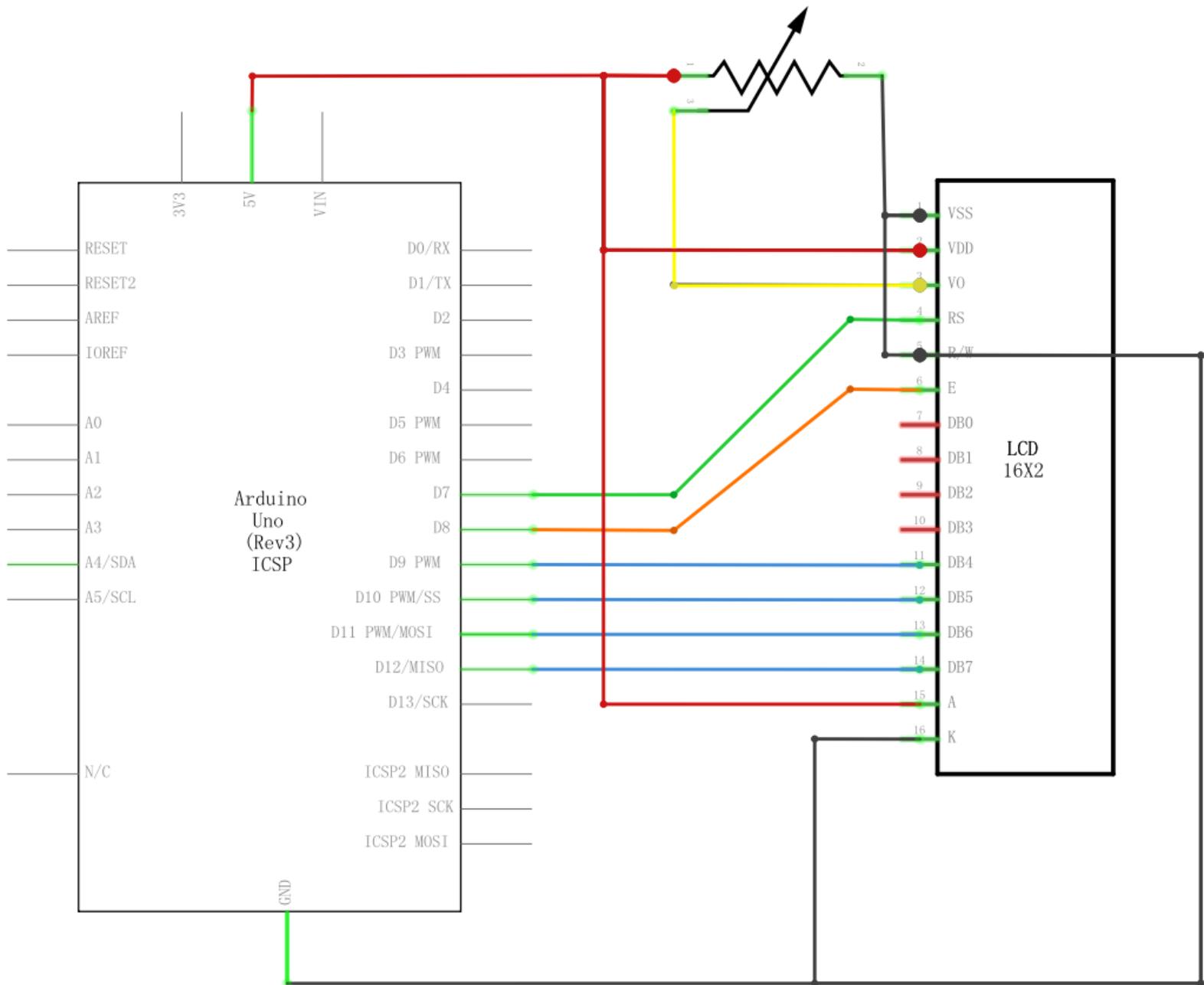
R/W: 読み出しモードまたは書き込みモードを選択する読み出し/書き込みピン

E: ローレベルのエネルギーが供給されると、LDC 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。

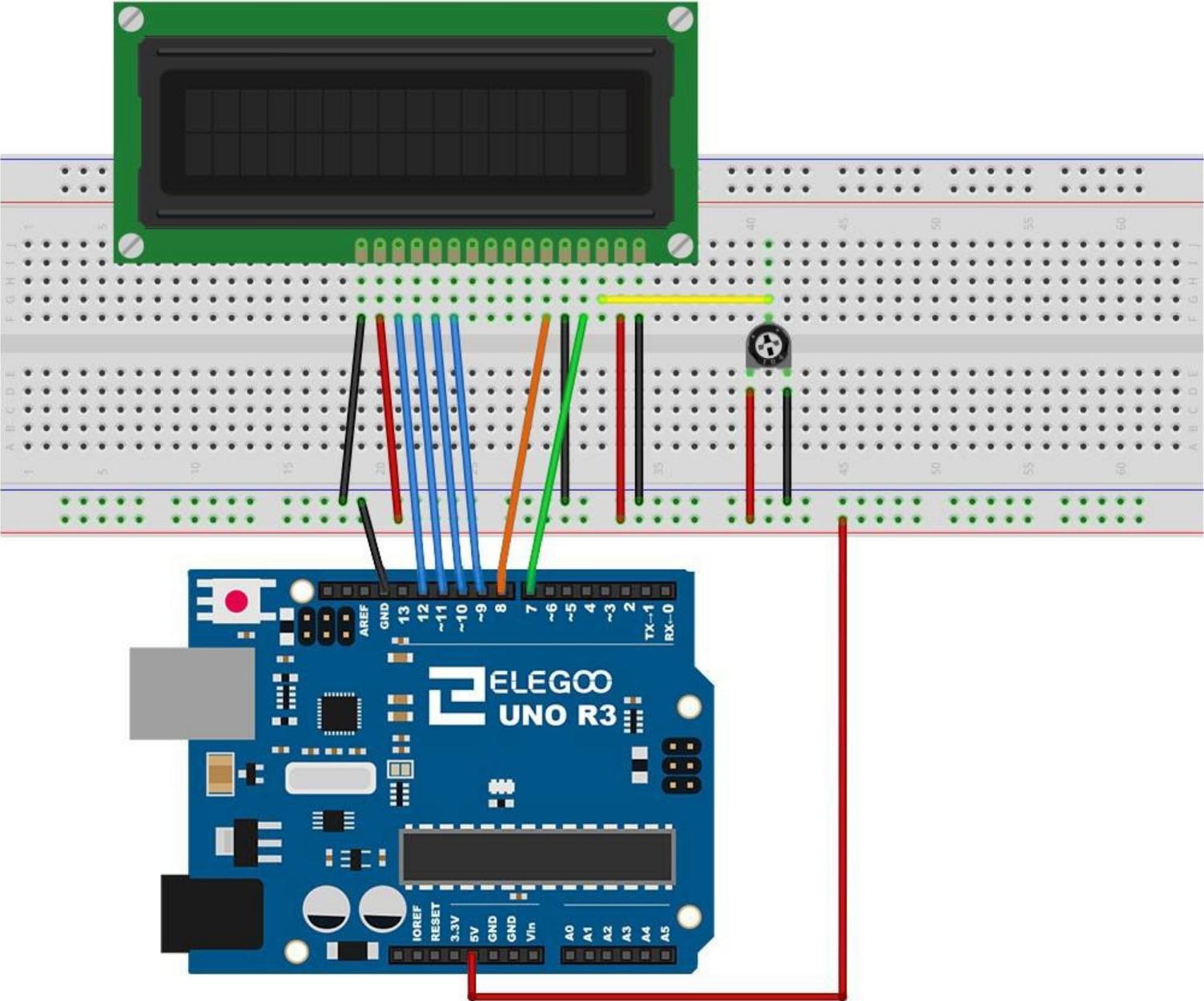
D0-D7: データを読み書きするためのピン

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:

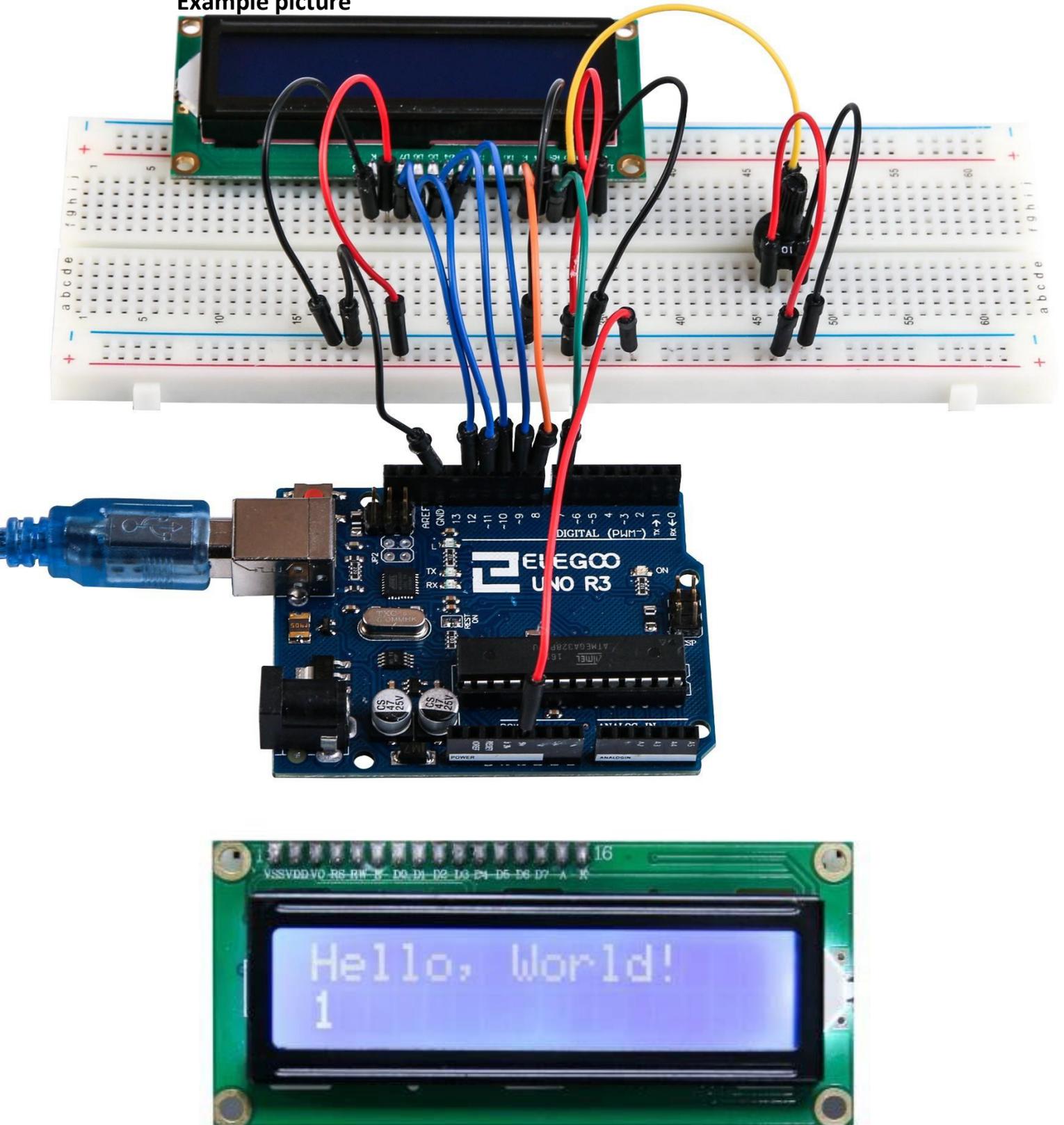
```
lcd.setCursor(0, 1);
```

```
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 23 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 jumper wires)

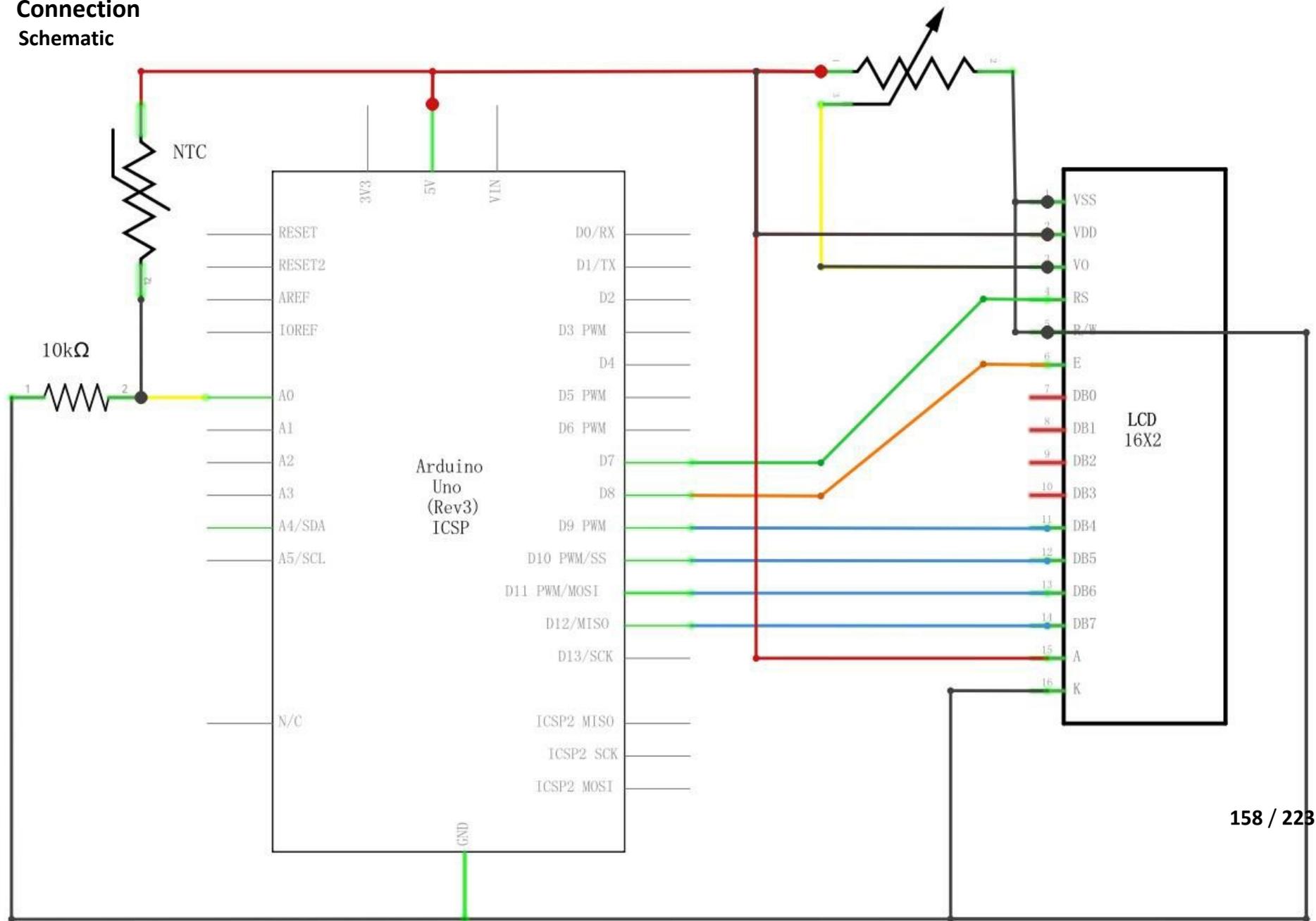
部品の紹介

Thermistor

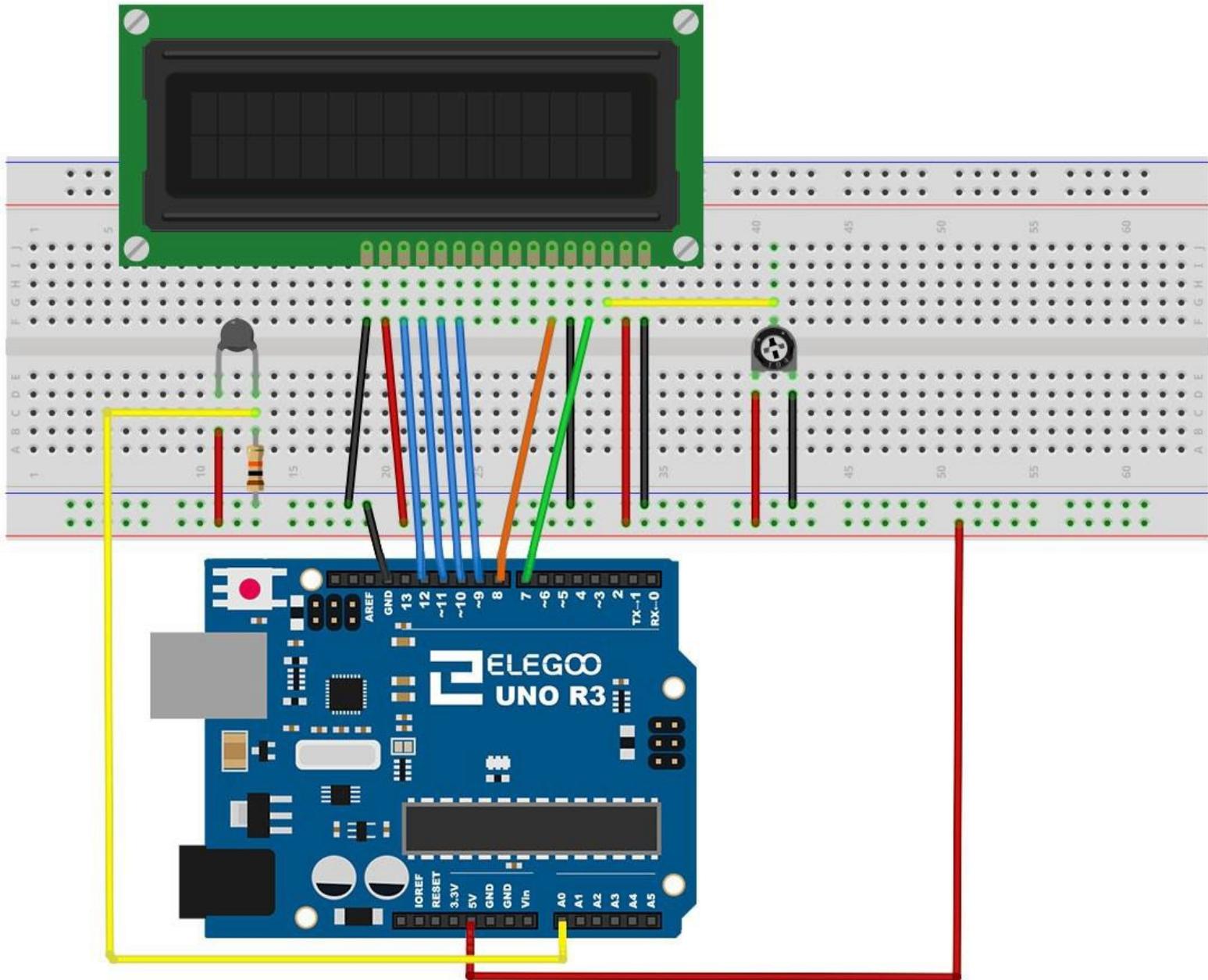
サーミスタはサーマル抵抗で、温度によって抵抗が変化する抵抗です。技術的には、すべての抵抗器はサーミスタです。抵抗は温度によってわずかに変化しますが、その変化は通常非常に小さく、測定するのが困難です。サーミスタは温度によって抵抗値が大きく変化するように作られており、1 度あたり 100 オーム以上変化させることができます!

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 はリセット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、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いくつかのジャンパーワイヤーがあります。

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。

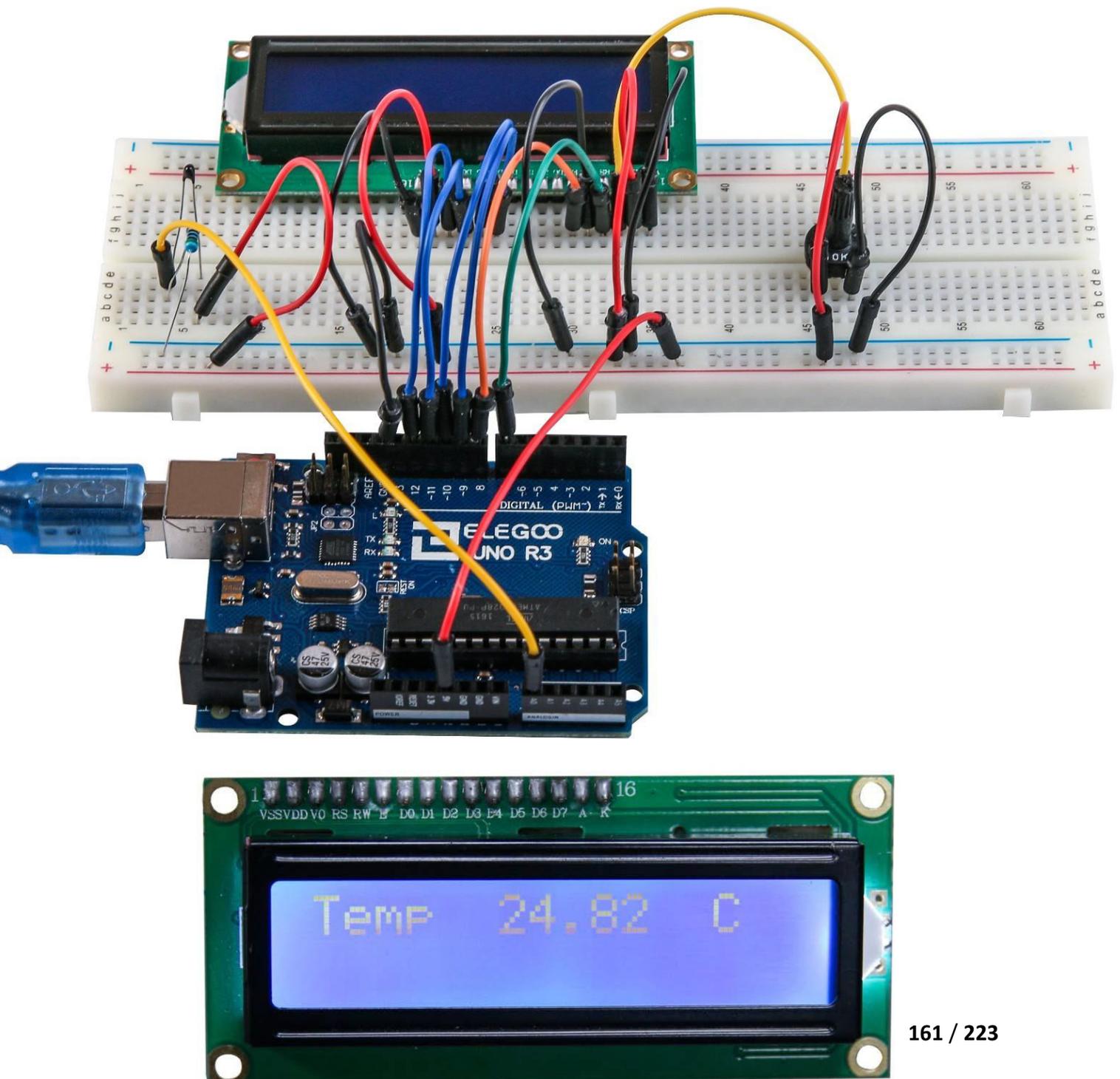
これを避けるには、ループの周りに LCD の行全体を書きます。

```
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に取り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 24 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にたくさんのも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場合がありますが、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、一度に少しず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:

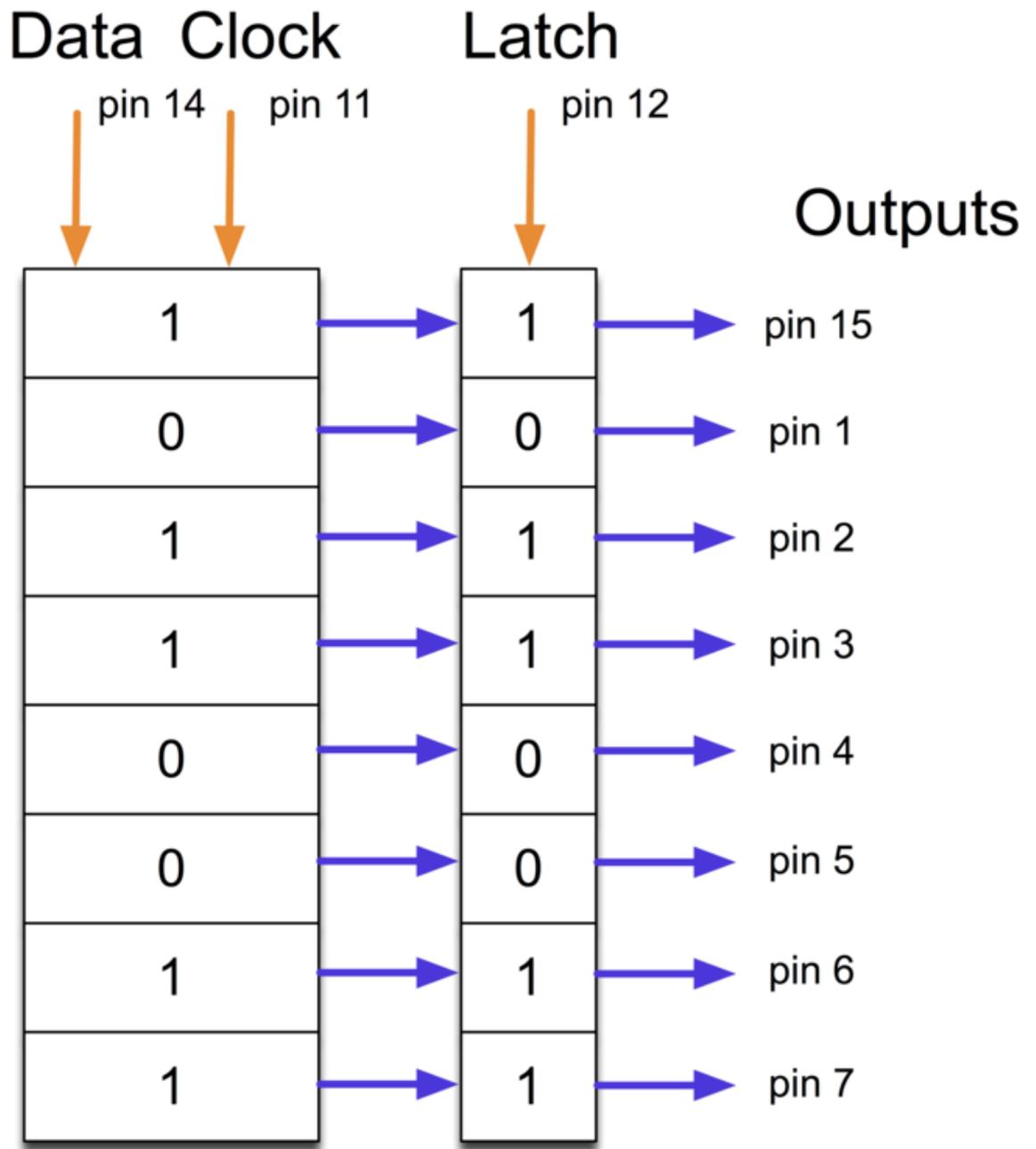
- (1) x Elegoo Uno R3
- (1) x 830 tie-points breadboard
- (8) x leds
- (8) x 220 ohm resistors
- (1) x 74hc595 IC
- (14) x M-M wires (Male to Male jumperwires)



部品の紹介

74HC595 Shift Register:

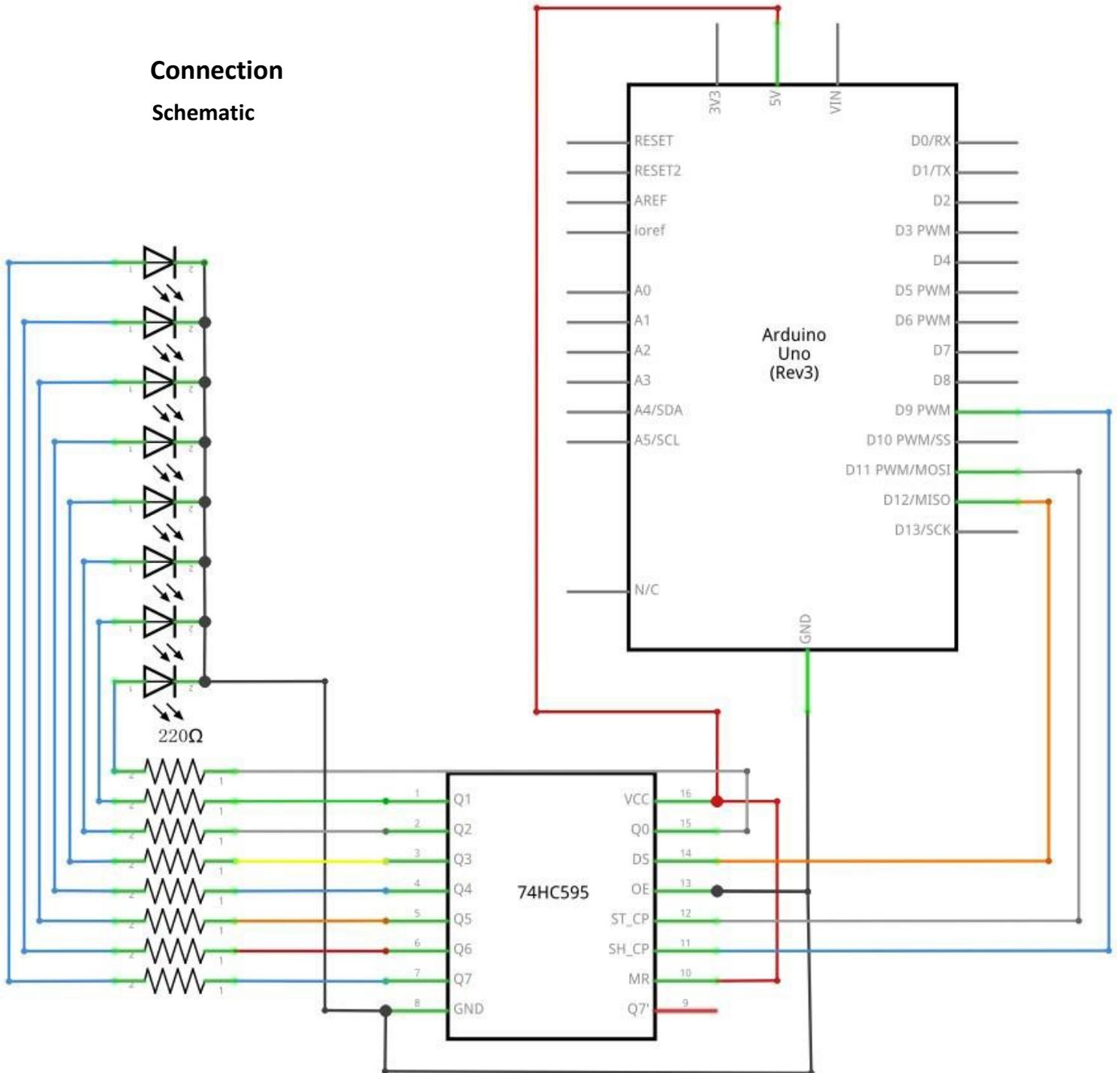
シフトレジスタは、8つのメモリ位置として考えられるものを保持する一種のチップであり、それぞれが1または0のいずれかになります。これらの値をそれぞれオンまたはオフに設定するには、チップの「データ」ピンと「クロック」ピンです。



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、間違っただ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すべて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



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ー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）。

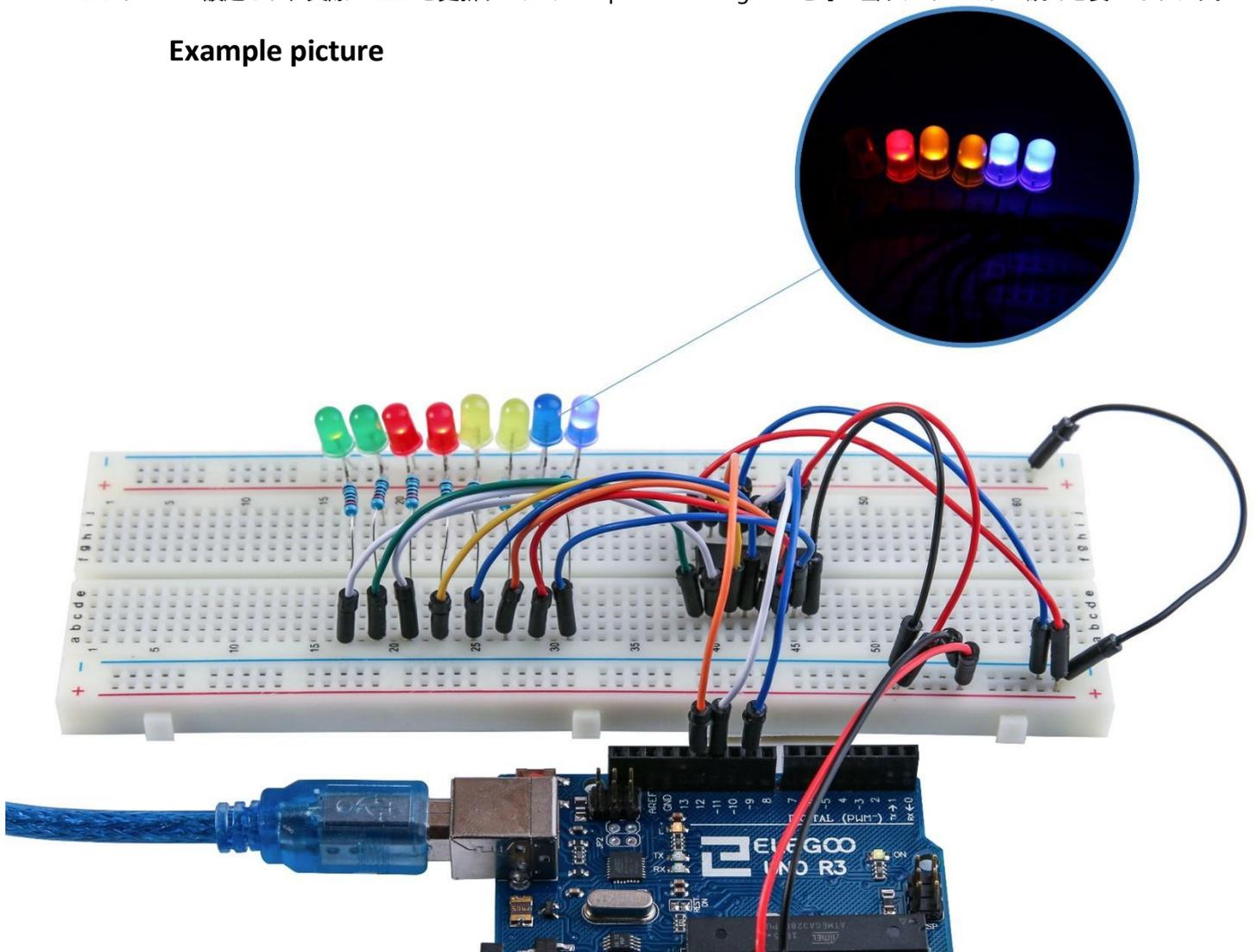
```

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 picture



Lesson 25 シリアルモニタ

概要

このレッスンでは、レッスン 24 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 24 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 24 に従っていない場合は、レッスン 24 に従ってください。

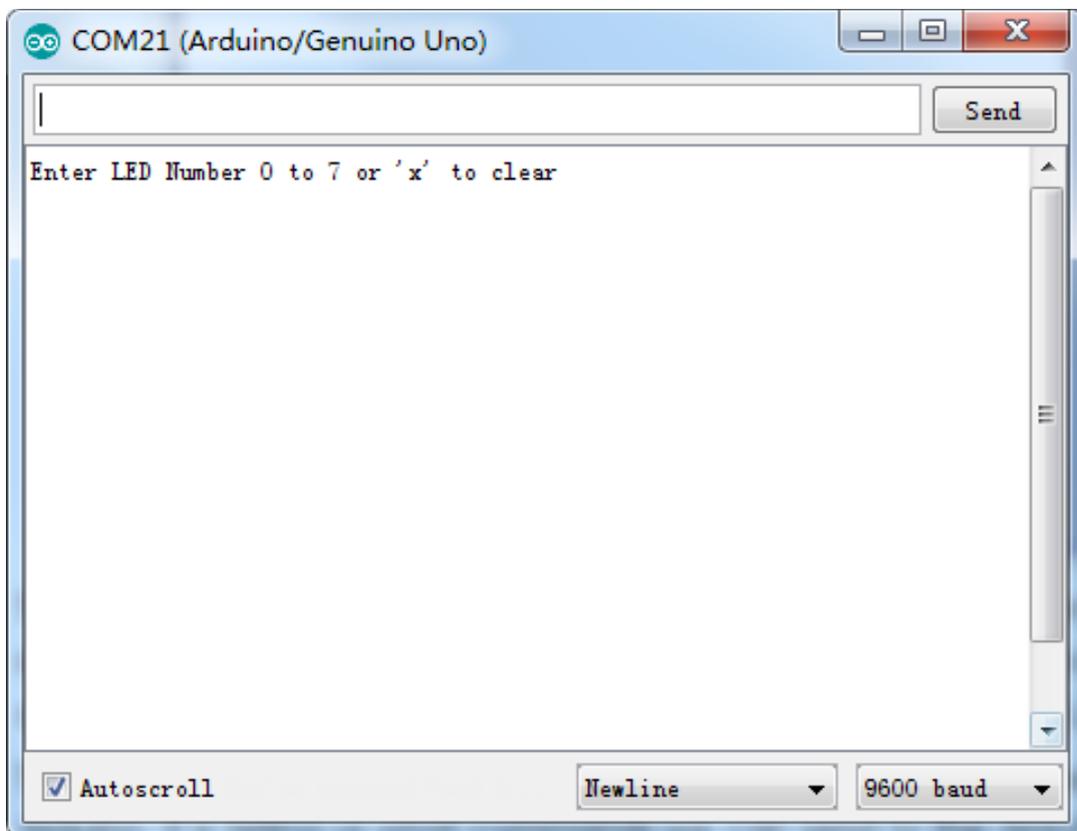
Steps taken

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。

```
The_Serial_Monitor $
1 //www.elegoo.com
2 //2016.12.9
3
4 int latchPin = 11;
5 int clockPin = 9;
6 int dataPin = 12;
7
8 byte leds = 0;
9 void updateShiftRegister ()
10 {
11     digitalWrite(latchPin, LOW);
12     shiftOut(dataPin, clockPin, LSBFIRST, leds);
13     digitalWrite(latchPin, HIGH);
14 }
15 void setup ()
16 {
17     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。

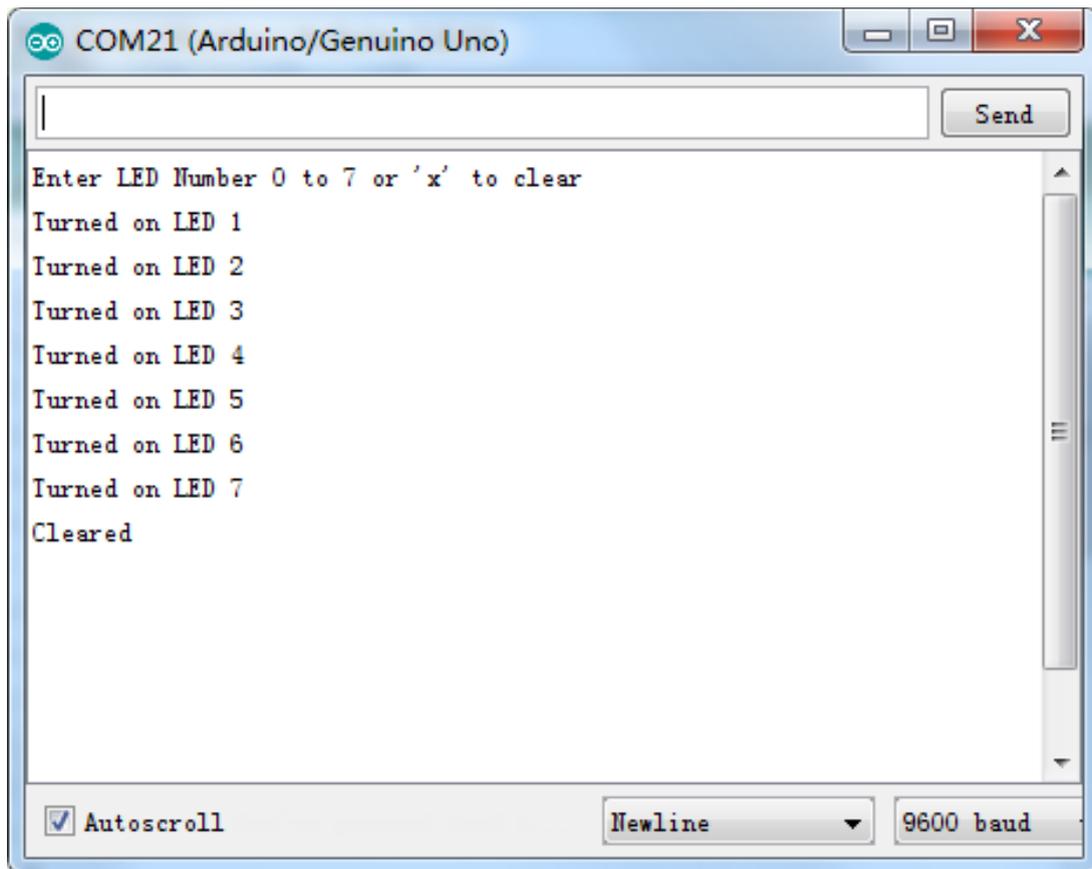


このウィンドウはシリアルモニタと呼ばれ、Arduino IDE ソフトウェアの一部です。その仕事は、コンピュータから UNO ボード (USB 経由) へのメッセージの送信と、UNO からのメッセージの受信の両方を可能にすることです。

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「ボーレート」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が、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 桁の数字か '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 桁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 26 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 24 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、銅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)



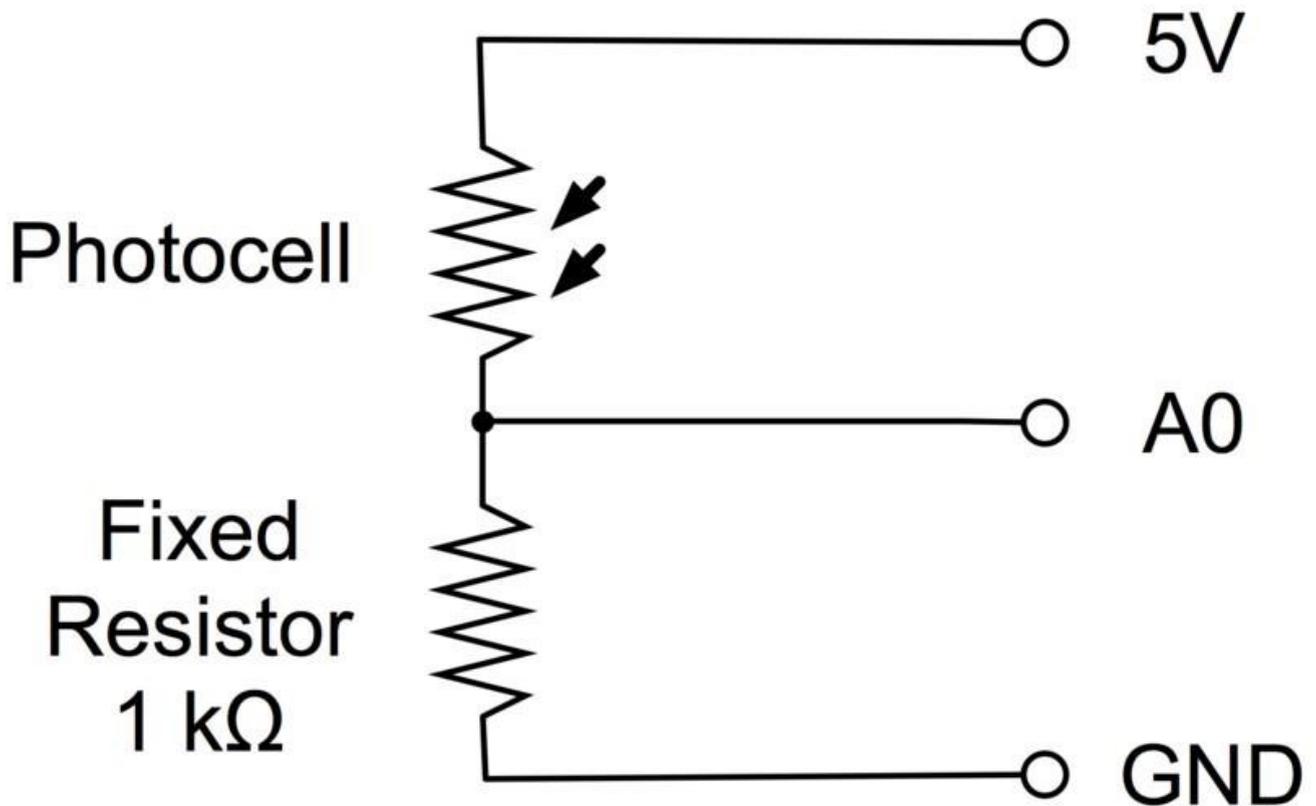
部品の紹介

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 50k Ω 、明るい光では 500 Ω 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。

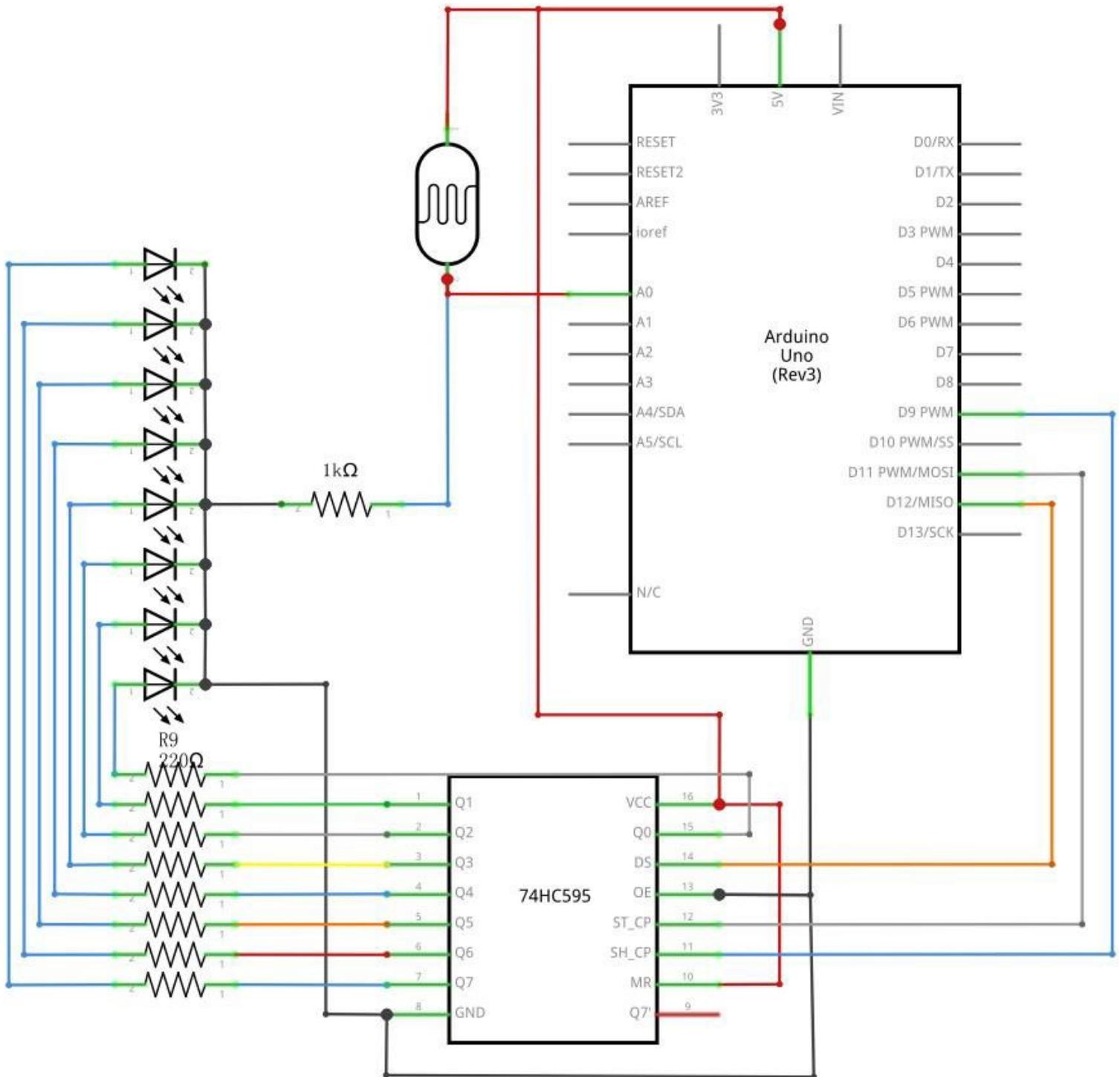


抵抗と光電池は一緒にポットのように振る舞います。光が非常に明るい場合、光電池の抵抗は固定値の抵抗と比較して非常に低く、ポットが最大になったかのようにです。

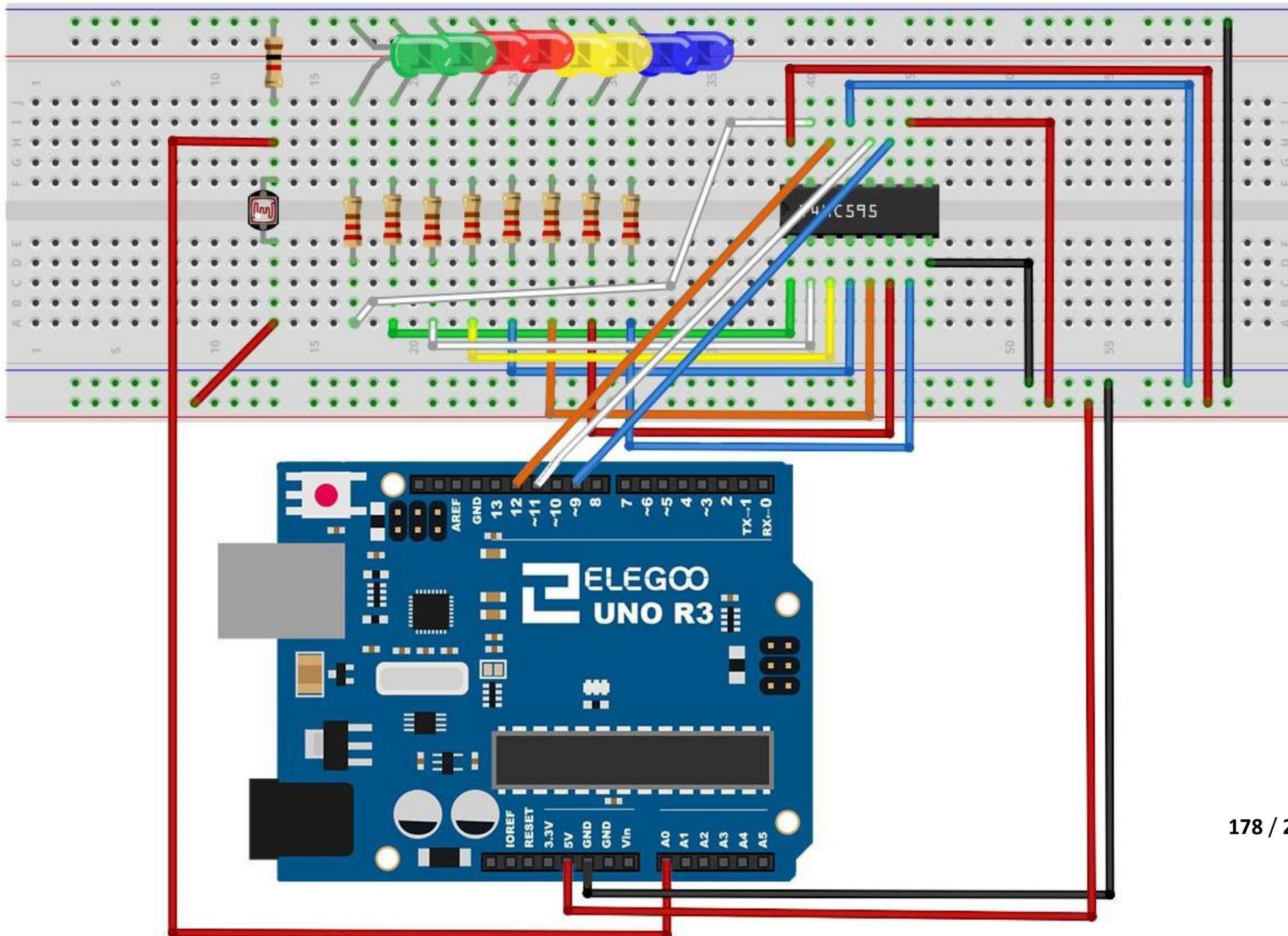
フォトセルが鈍い光の中にあるとき、抵抗は固定された $1\text{k}\Omega$ の抵抗よりも大きくなり、ポットが GND に向かって回転しているかようになります。

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。

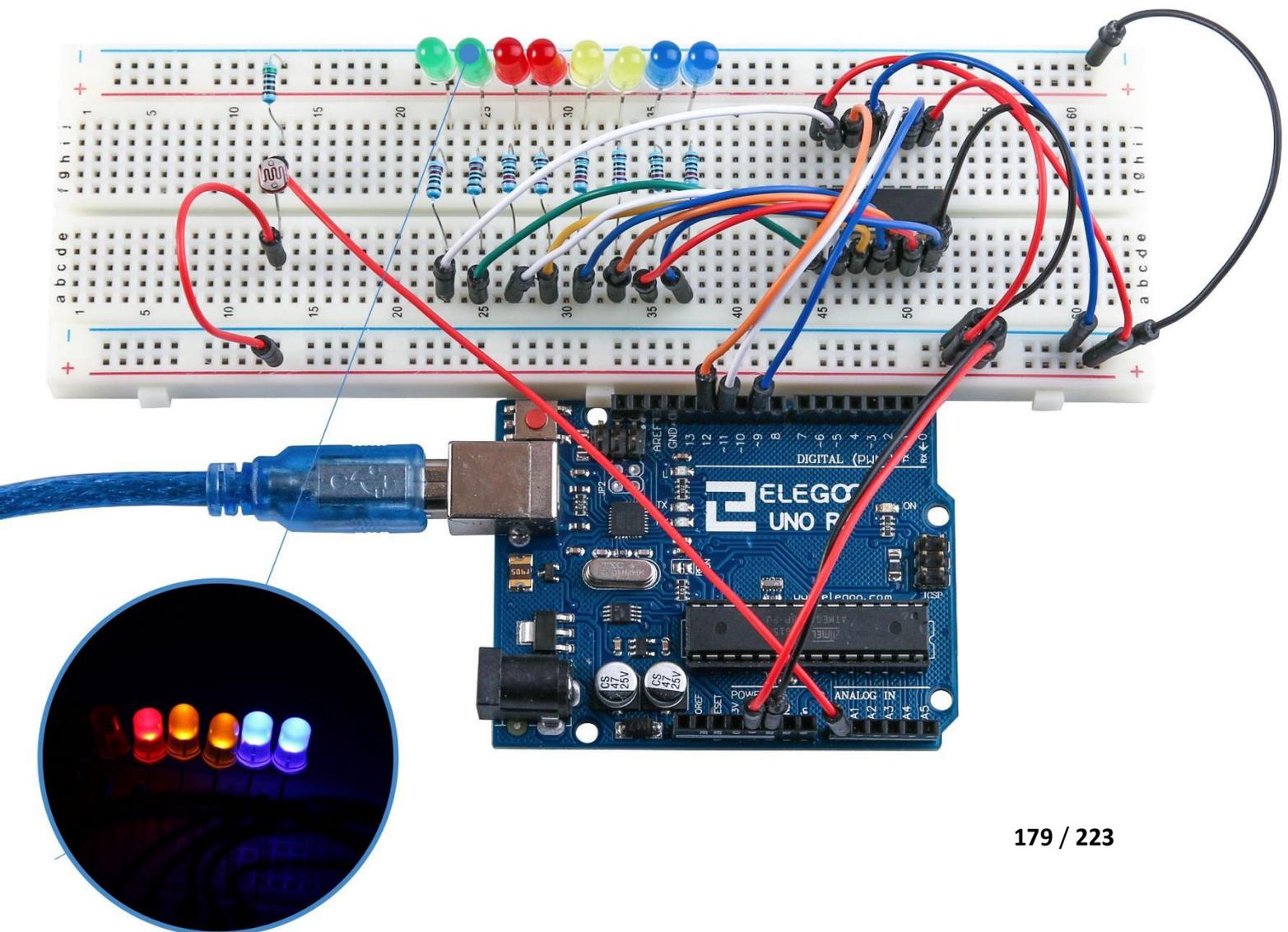
スケッチに対する唯一の他の実質的な変更は、点灯する LED の数を計算する線です:

```
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 1kΩ の抵抗を考慮することです。これは、光電池の抵抗が 1kΩ (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 27 74HC595 と 7 セグメント表示

概要

レッスン 24,25 とレッスン 26 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:

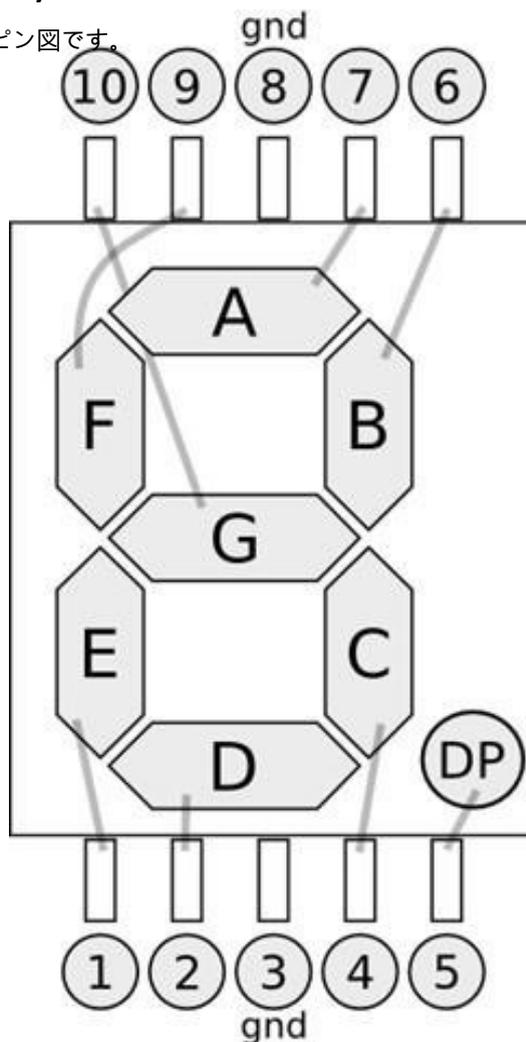
- (1) x Elegoo Uno R3
- (1) x 830 tie-points breadboard
- (1) x 74HC595 IC
- (1) x 1 Digit 7-Segment Display
- (8) x 220 ohm resistors
- (26) x M-M wires (Male to Male jumper wires)



部品の紹介

Seven segment display

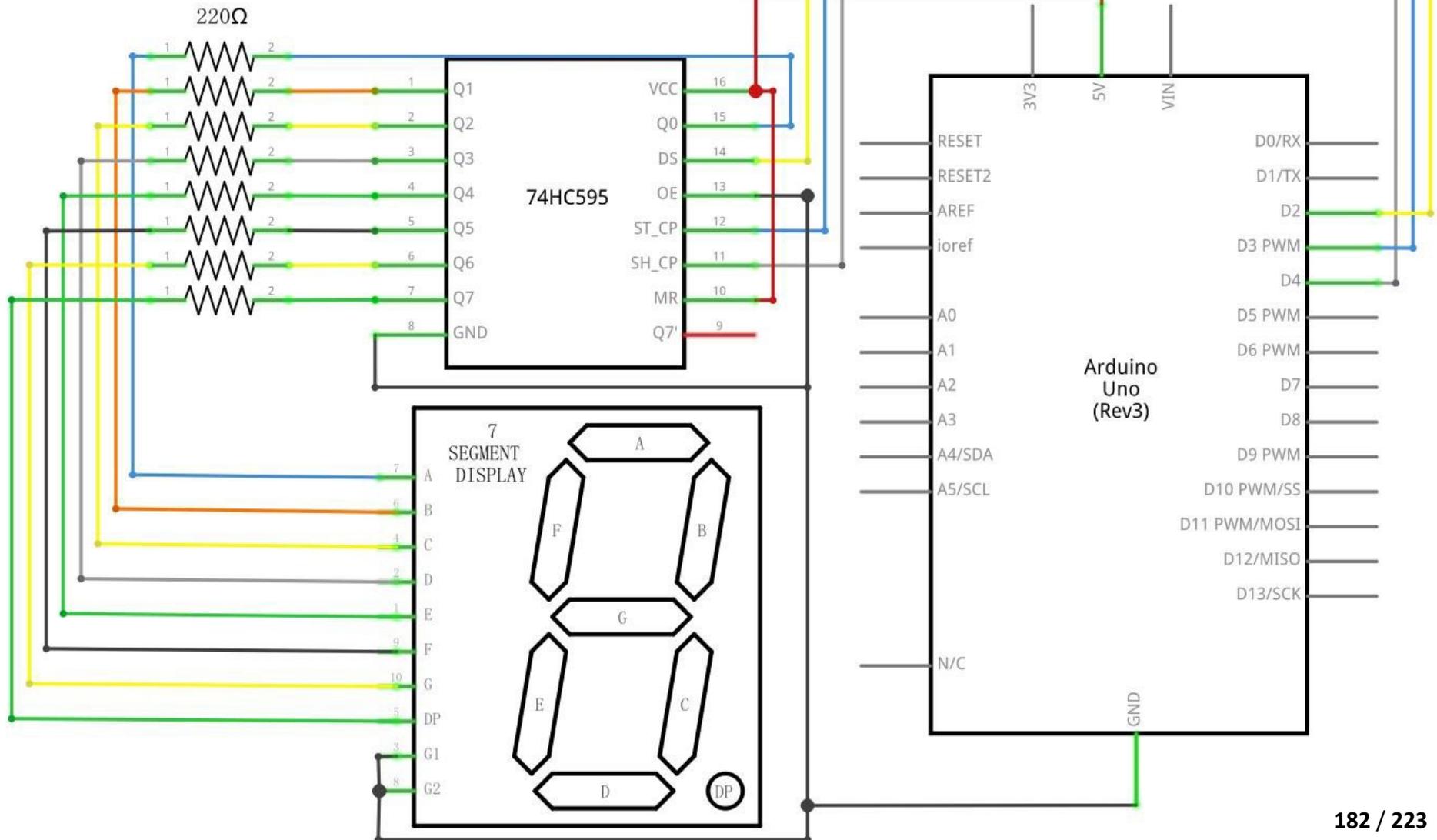
以下は 7 セグメントのピン図です。



0-9 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 1 0 0 をすべて 1 に置き換える必要があります) :

Displaydigital	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



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、配線は電源とグラウ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) はグラウンドに接続

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) 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) 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

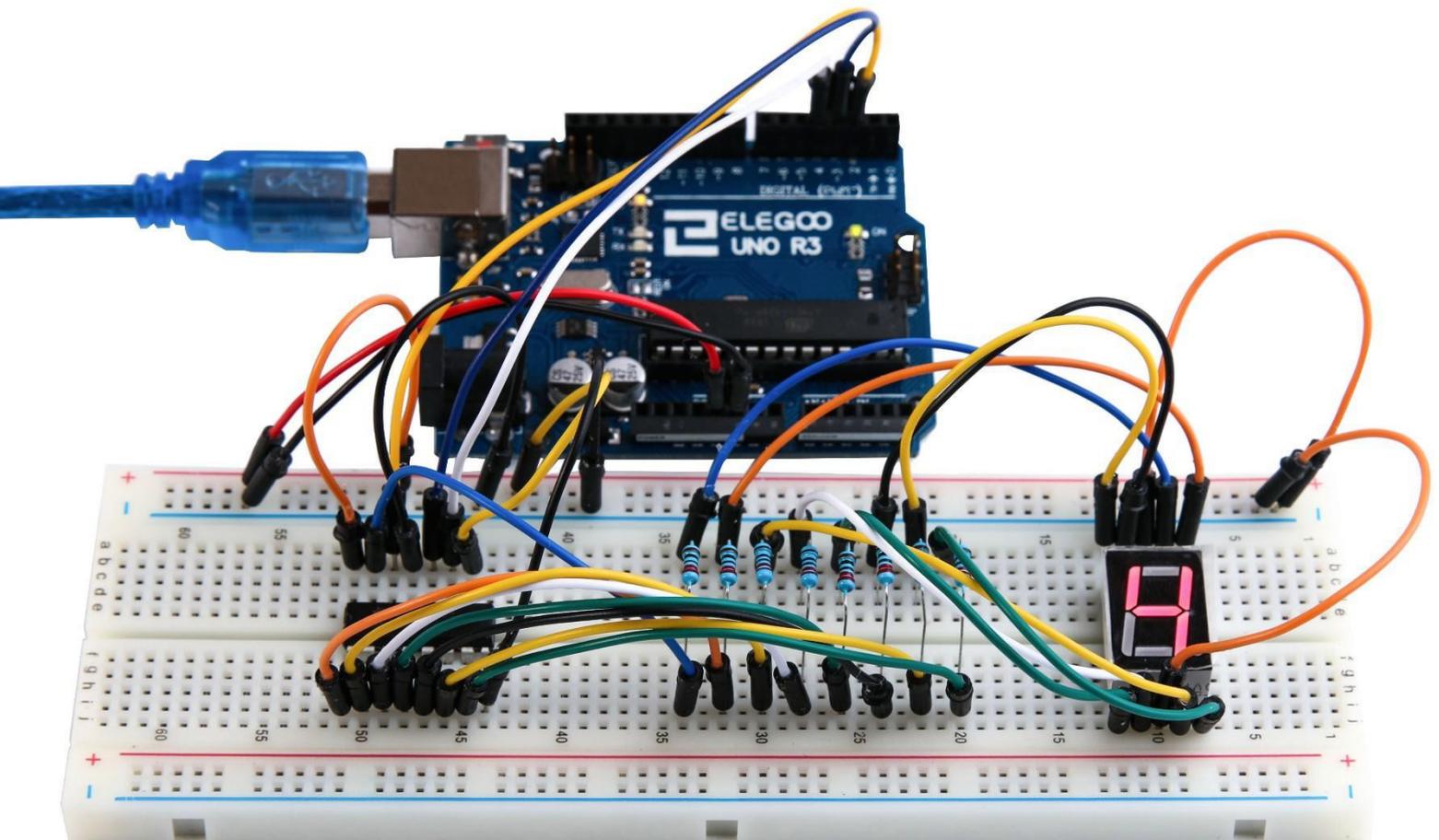
7セグメントディスプレイ 3,8 ピンから UNO R3 ボード GND (この例では共通のアノードを使用していますが、共通のアノードを使用する場合は 3,8 ピンを UNO R3 ボード+ 5V に接続してください)

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 28 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桁の7セグメント表示を使用する方法を学習します。

1桁の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桁の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桁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はできません。

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)

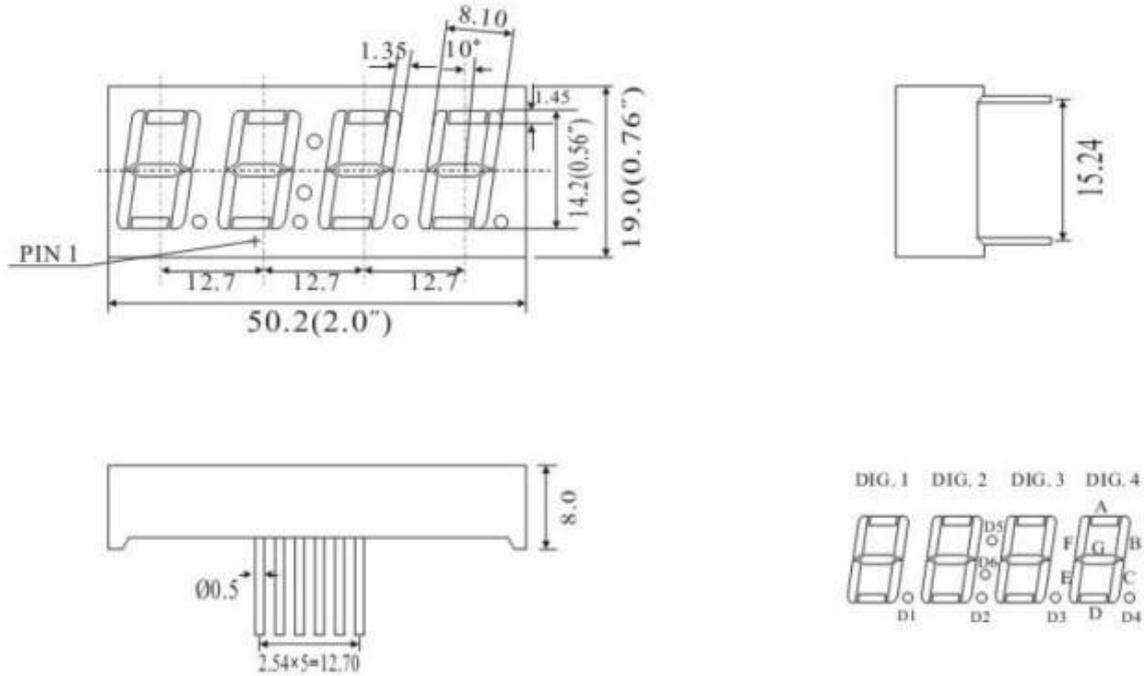


部品の紹介

Four Digital Seven segment display

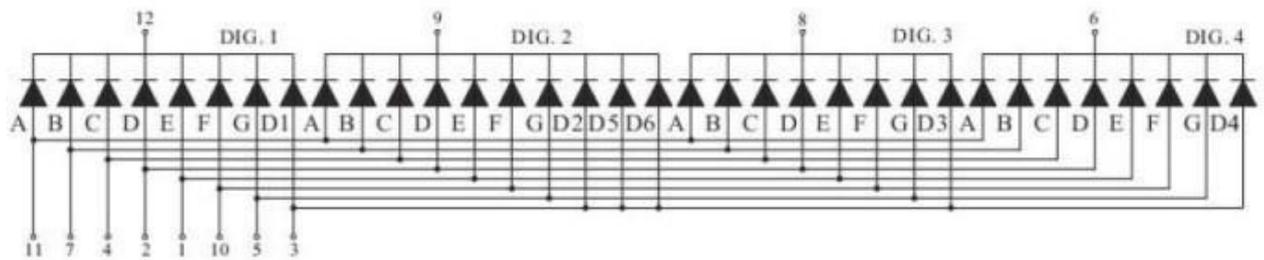
Package Dimensions

CPS05643AB

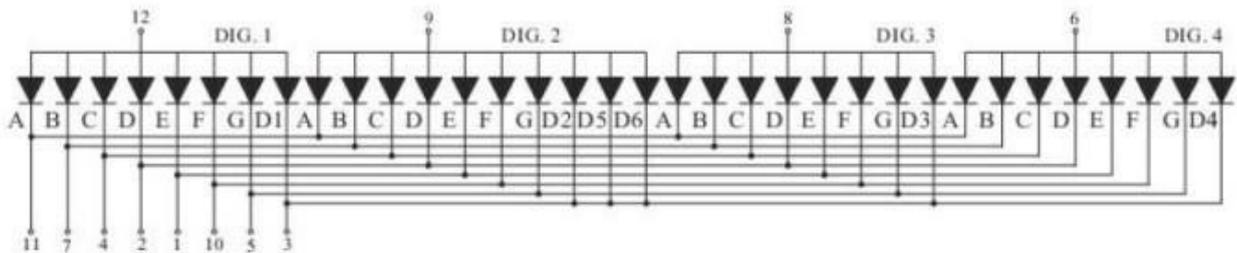


UNIT: MM(INCH) TOLERANCE: $\pm 0.25(0.01)$ "

Internal Circuit Diagram



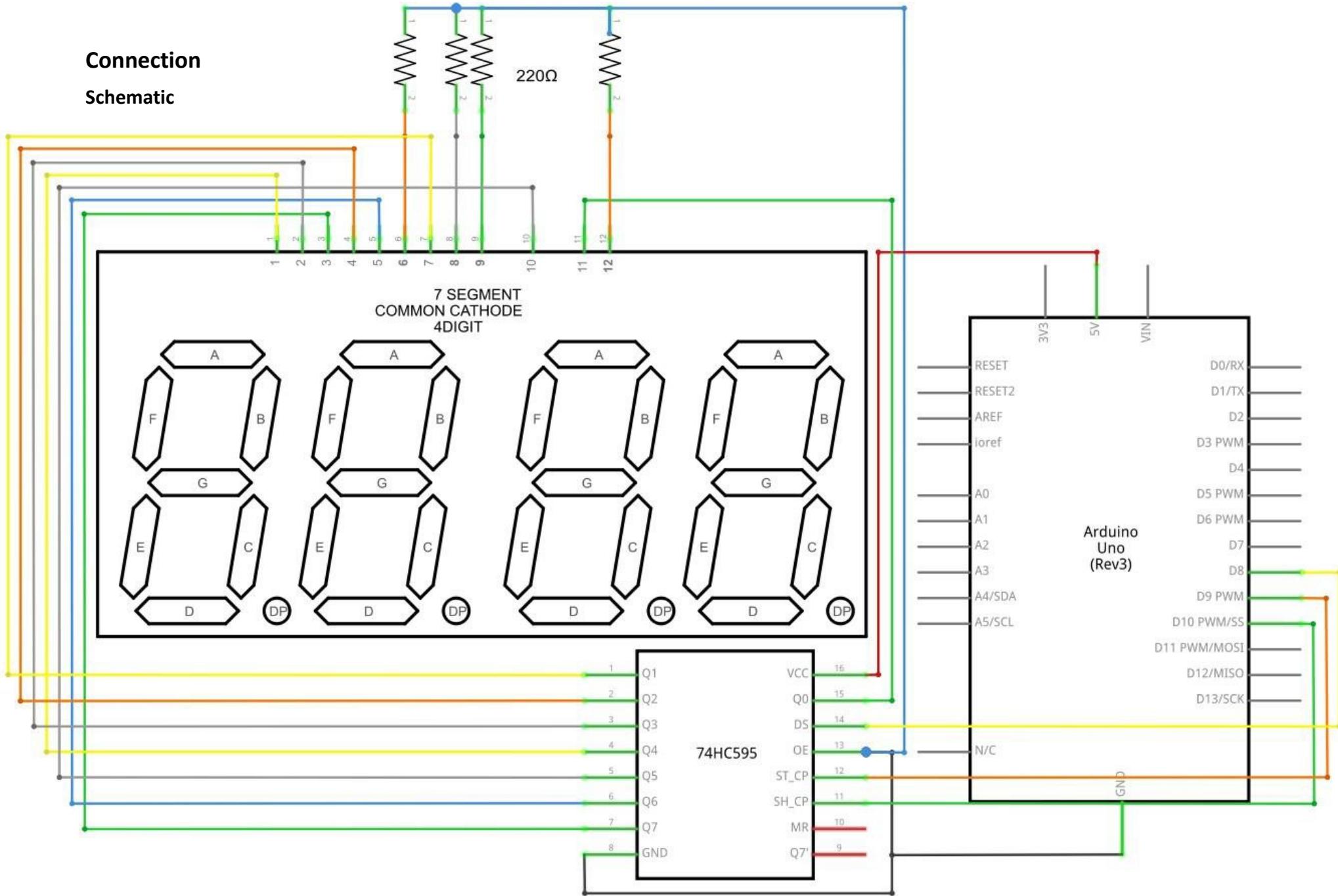
5643A



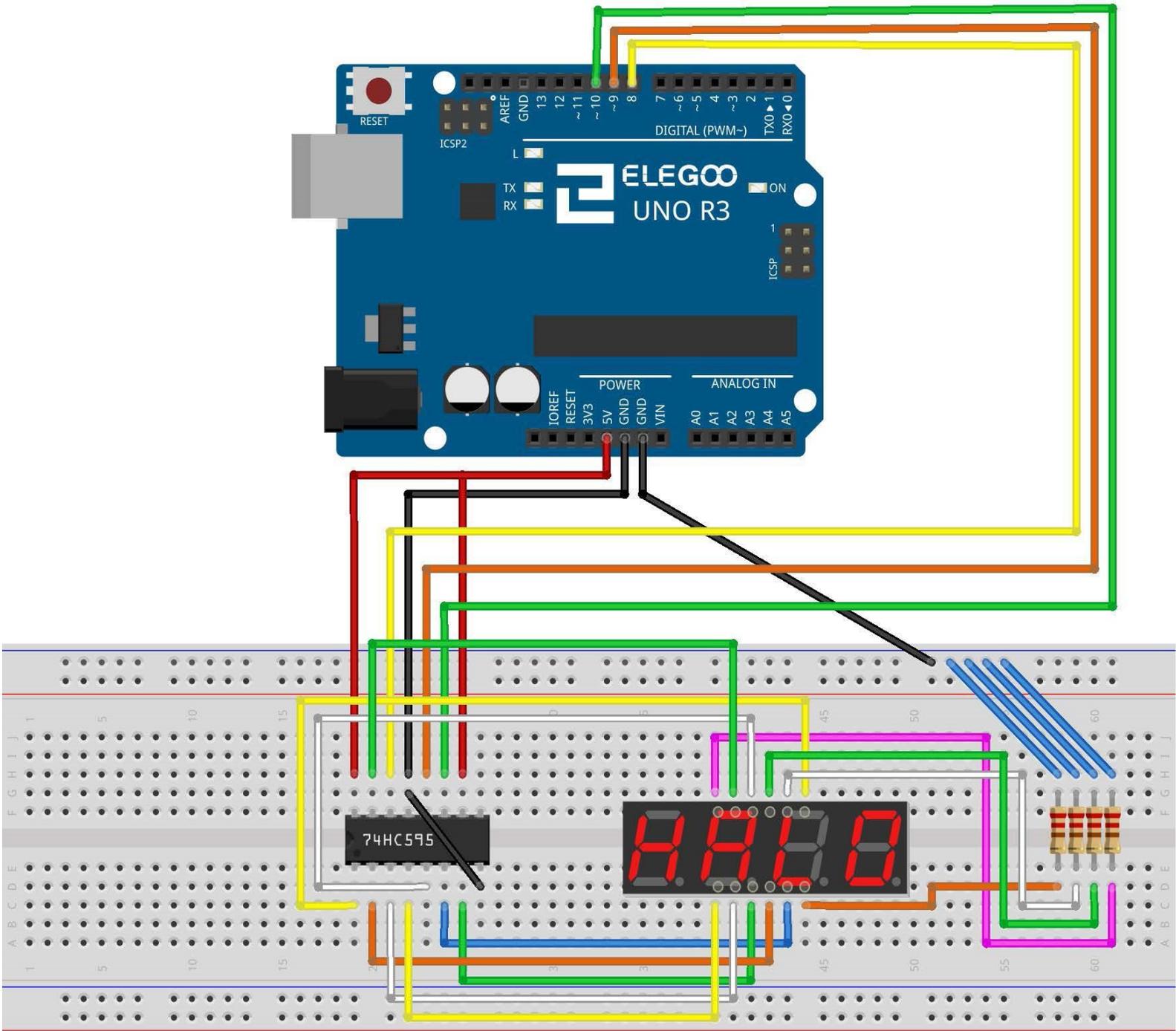
5643B

Four Digits Displays Series

**Connection
Schematic**



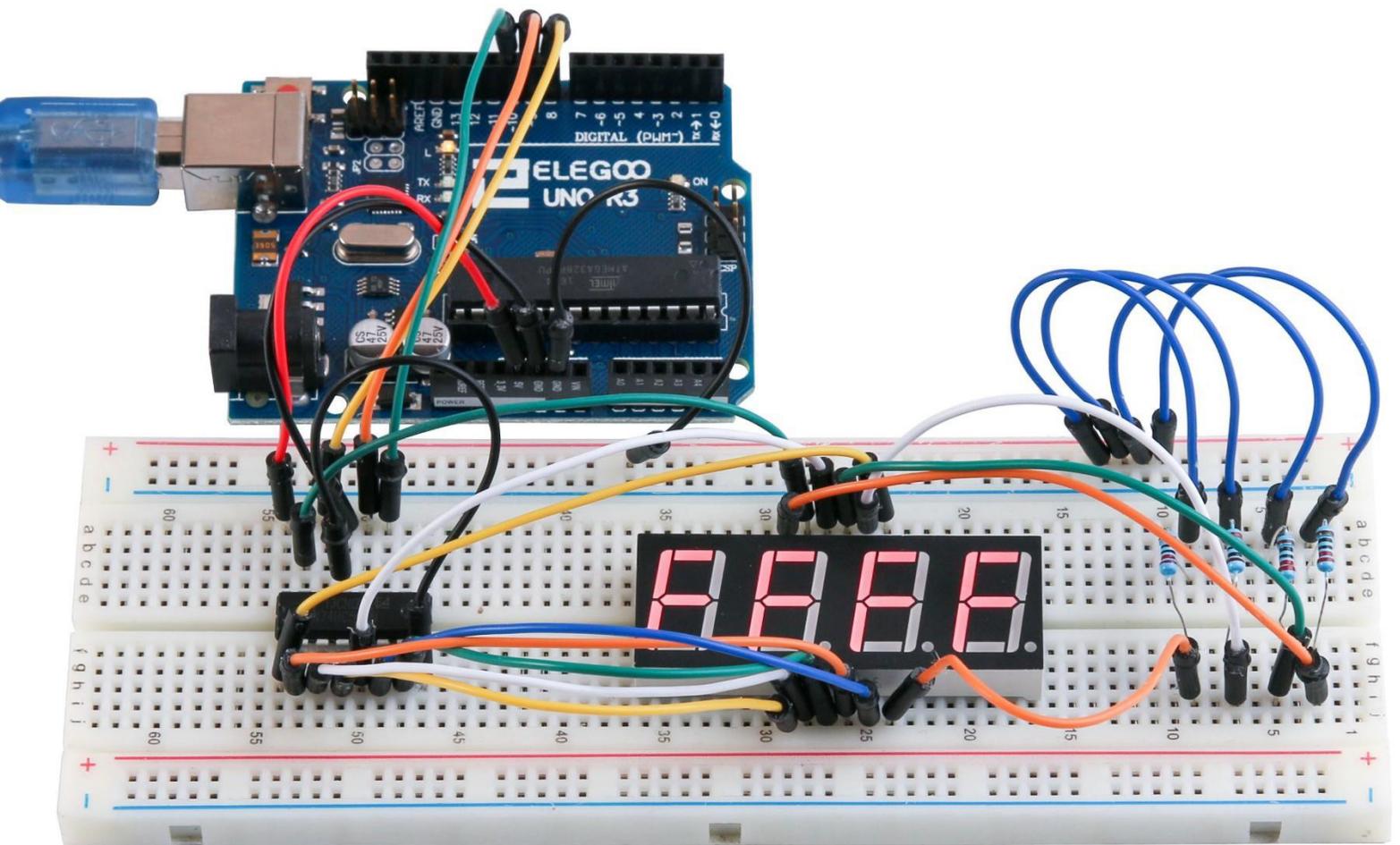
Wiring diagram



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 29 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

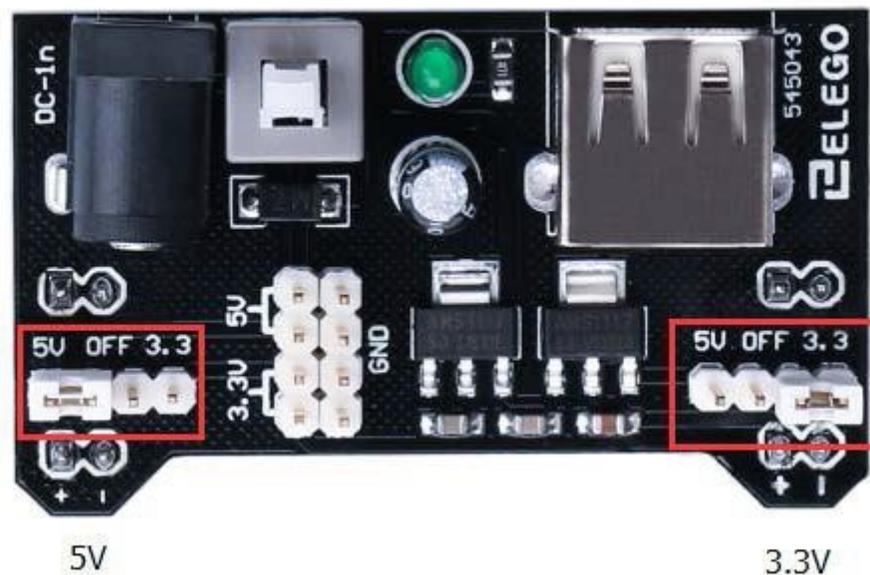
小型の DC モーターは、UNO R3 ボードのデジタル出力よりも多くの電力を直接使用できます。モーターを UNO R3 ボードのピンにまっすぐ接続しようとする、UNO R3 ボードに損傷を与える可能性があります。私たちは電源モジュールを使用して電源を供給します。



製品仕様:

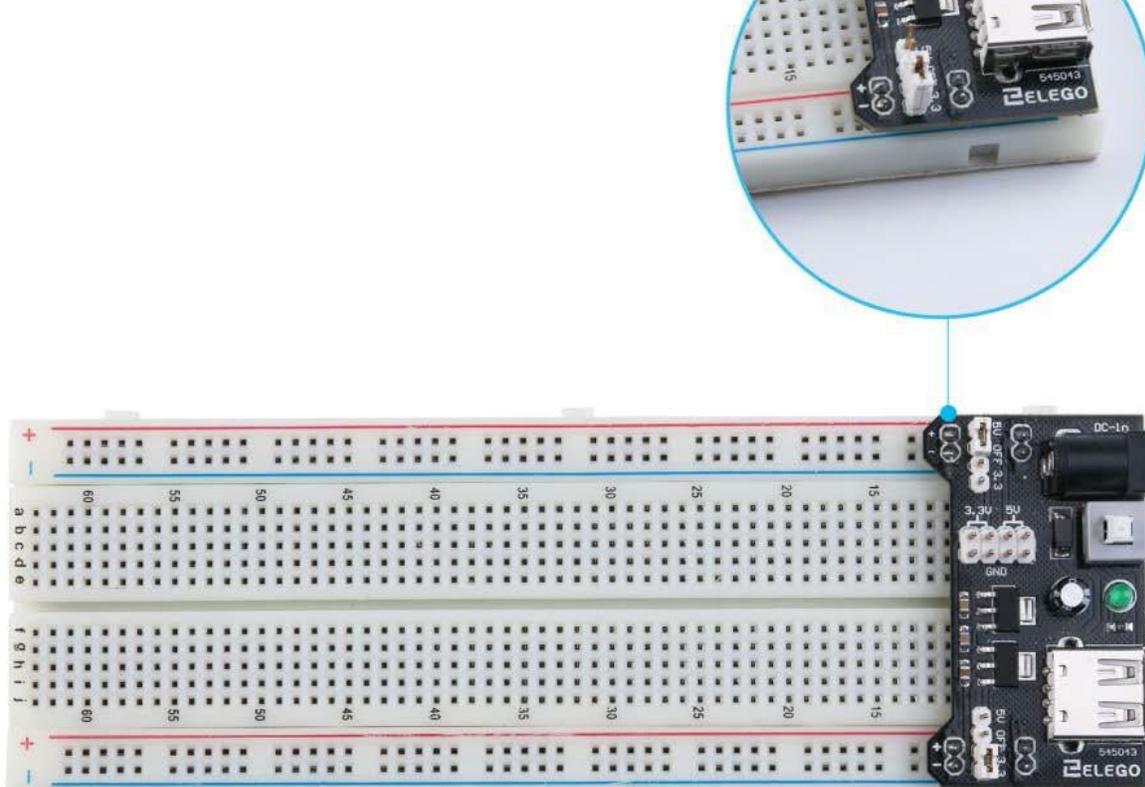
- Locking On/Off Switch
- LED Power Indicator
- Input voltage: 6.5-9v (DC) via 5.5mm x 2.1mm plug
- Output voltage: 3.3V/5v
- Maximum output current: 700mA
- 独立した制御レール出力 0v, 3.3v, 5v to breadboard
- 外部使用に便利な出力
- Size: 2.1 in x 1.4 in
- 外部デバイスに電源を供給するためのオンボード USB デバイスコネクタ

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 LED とブレッドボード電源レールの電源は入らない。

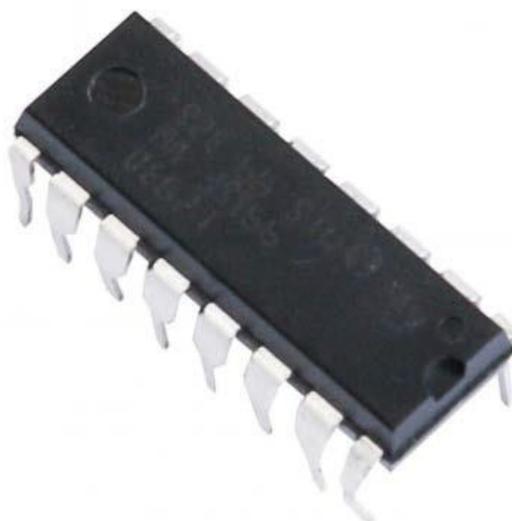


Important note:

ブレッドボードに正しくモジュールが揃っ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ことがあります

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)

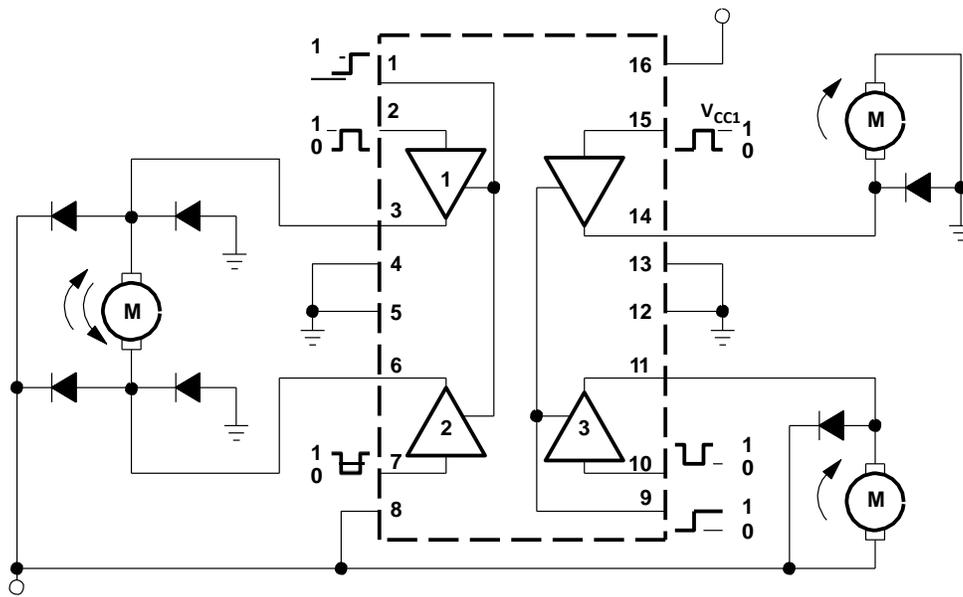


説明/発注情報

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、完全なトータ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ク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。イネーブル入力が高いとき、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。イネーブル入力が高いとき、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読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:

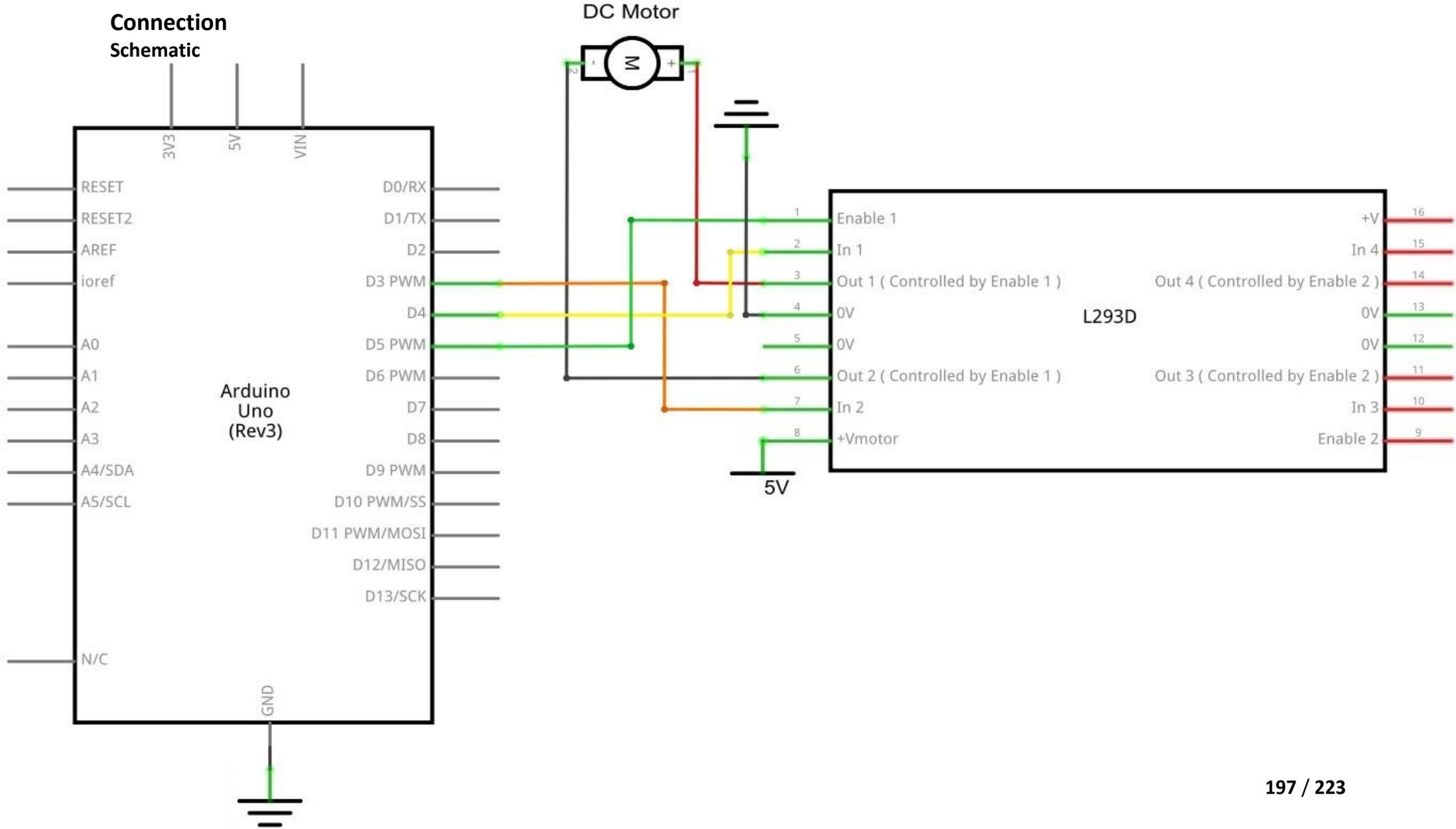
左側は第 1 のモータを処理し、右側は第 2 のモータを処理する。はい、あなたは 1 台のモーターだけを接続して走らせることができます。

Arduino 接続

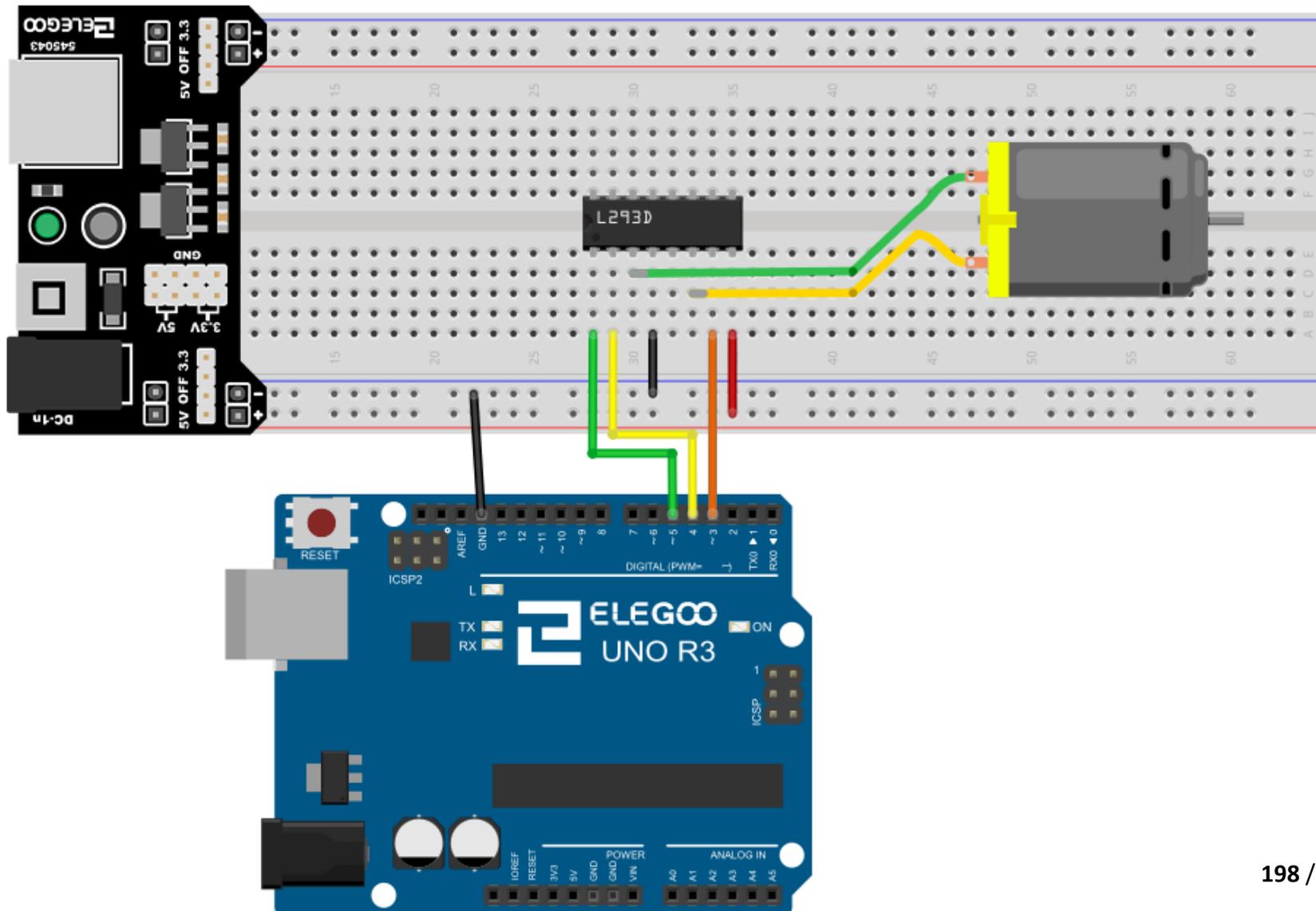
M1 PWM - これを Arduino の PWM ピンに接続します。それらは Uno でラベル付けされています、ピン 5 は例です。0 から 255 までの任意の整数を出力します。0 はオフ、128 は半速度、255 は マックス・スピード。

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 Schematic

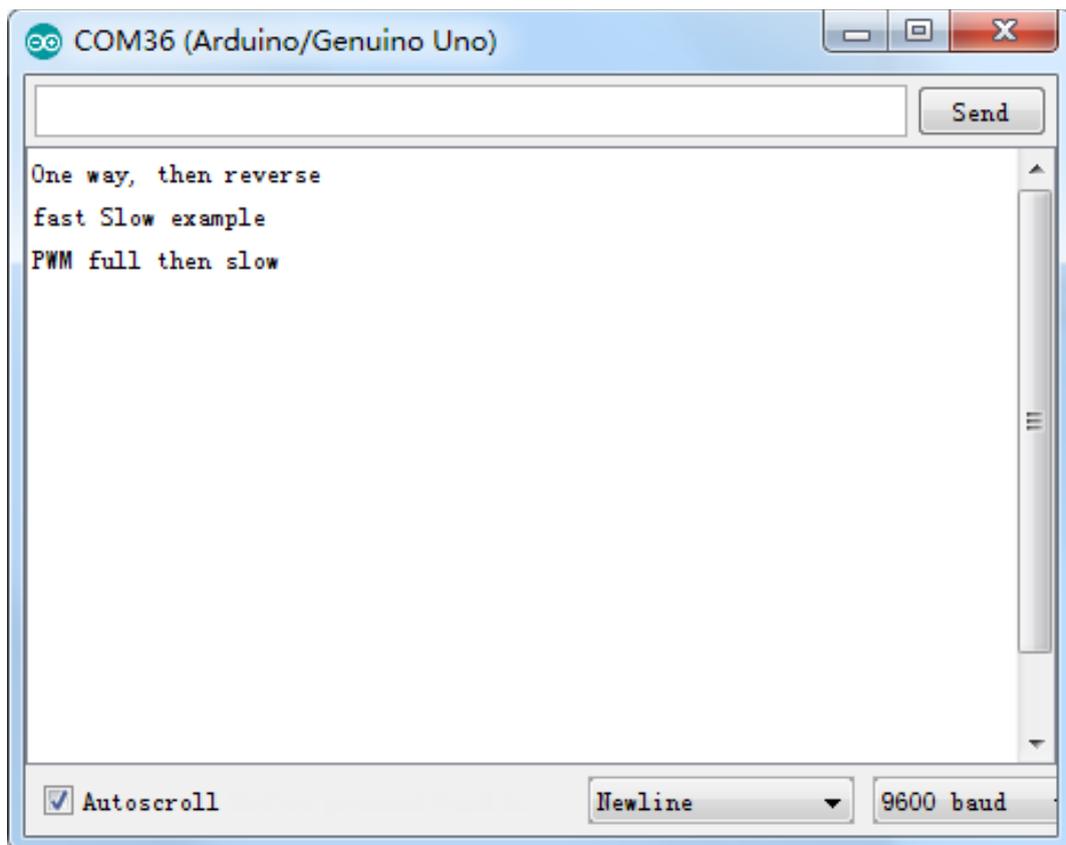


Wiring diagram



以下のコードでは、別の電源（バッテリー）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氣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。

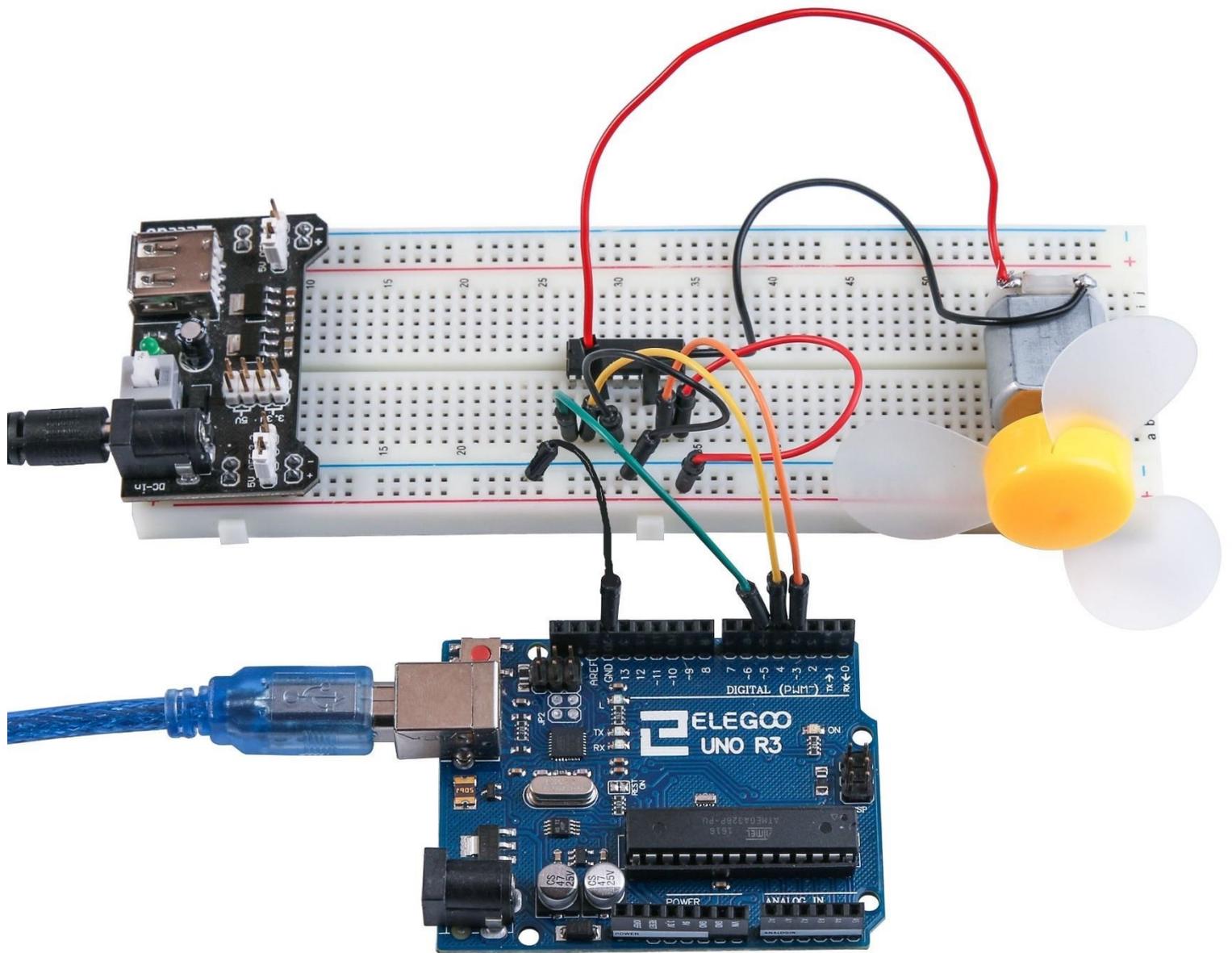


Code

配線後、プログラムをコードフォルダ - レッスン 29 DC モーターで開き、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ーを駆動し、モーターはその最大 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 30 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 jumperwires)



部品の紹介

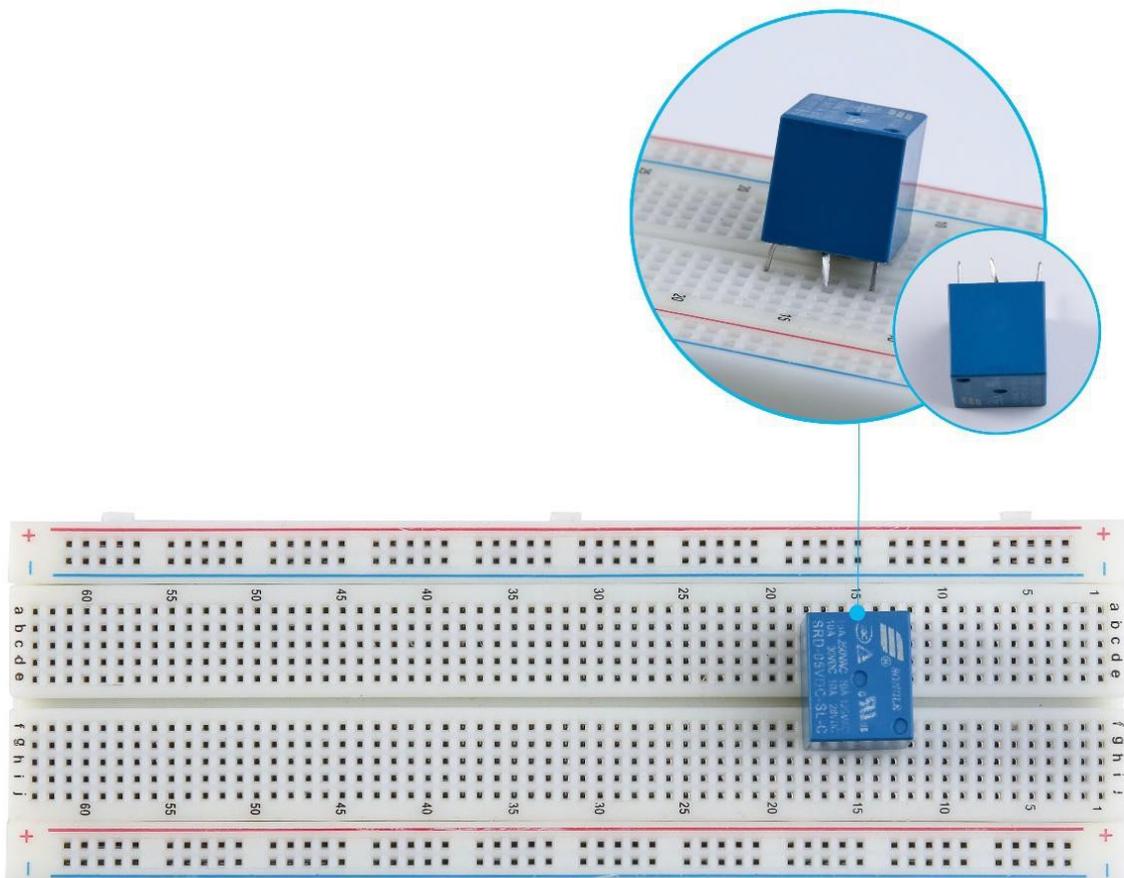
Relay:

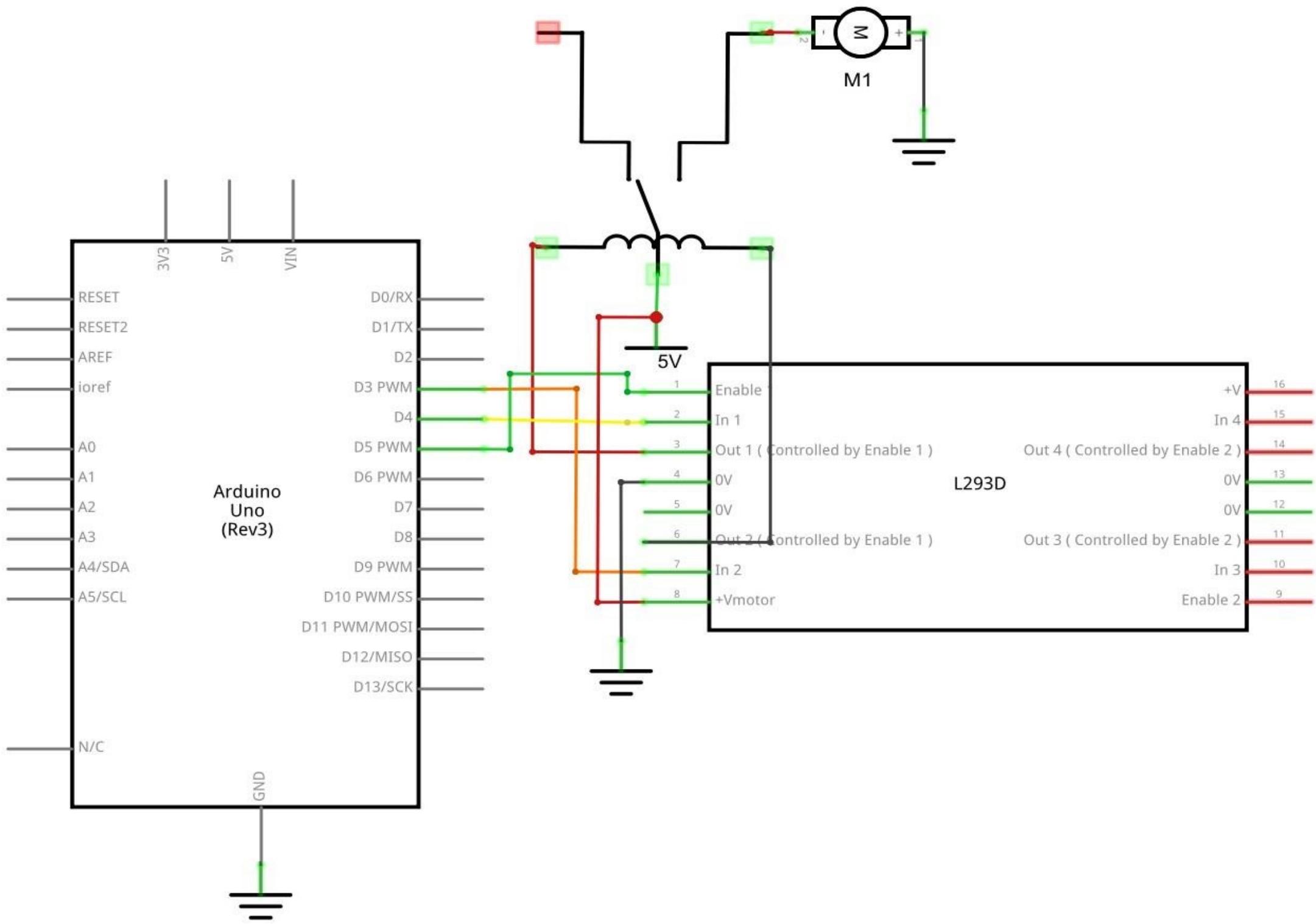
リレーは電氣的に操作されるスイッチである。多くのリレーは電磁石を使用してスイッチを機械的に動作させますが、他の動作原理もソリッドステートリレーのように使用されます。リレーは、低電力信号（制御と制御された回路の間の完全な電氣的絶縁を使用）によって回路を制御する必要がある場合、または複数の回路を1つの信号で制御する必要がある場合に使用されます。第1のリレーは増幅器として長距離電信回路に用いられた。彼らは、ある回路から入ってきた信号を繰り返し、別の回路でそれを再送しました。リレーは、論理的な操作を行うために電話交換機や初期のコンピュータで広く使用されていました。

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トと呼ばれます。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。

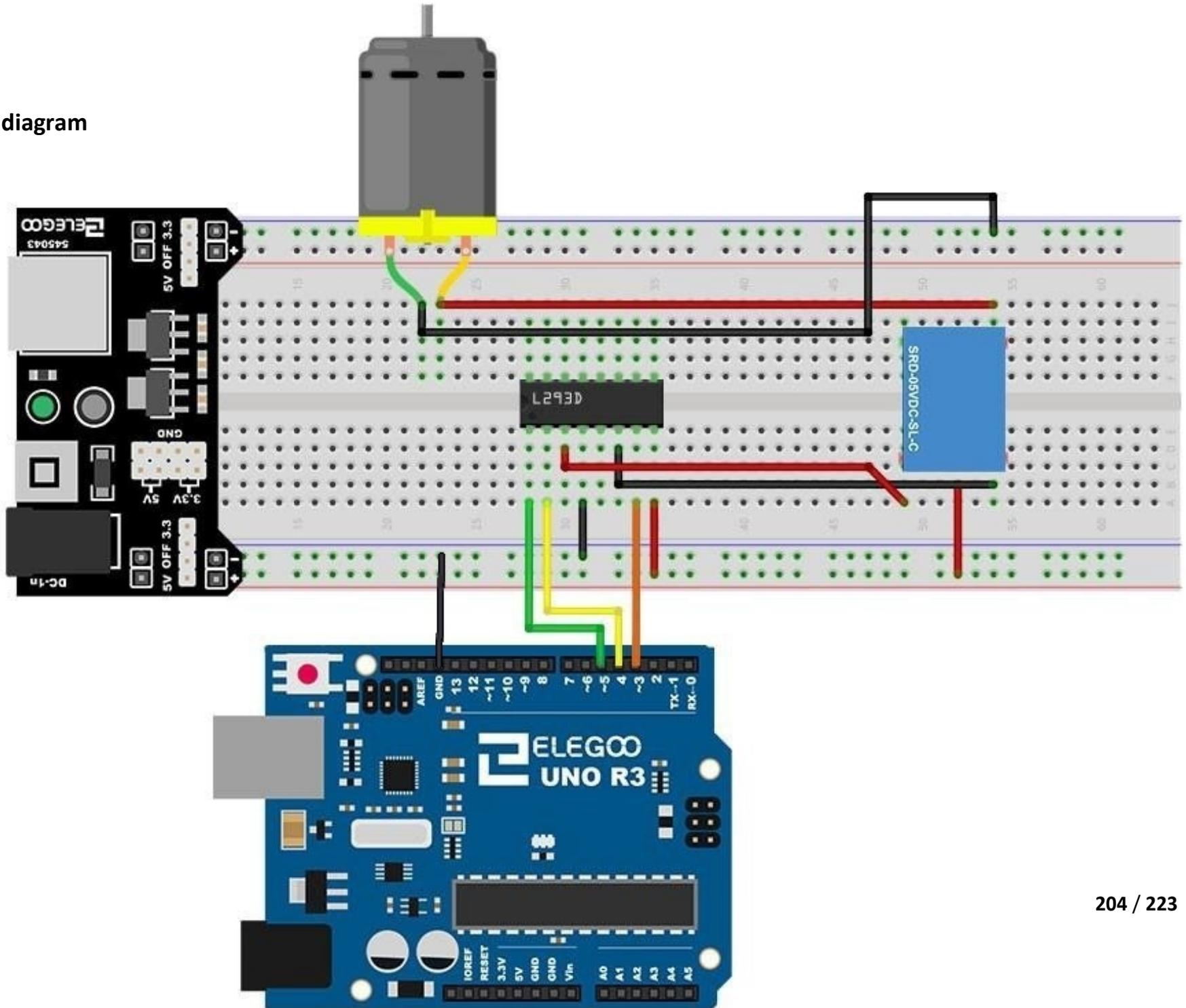
以下は Arduino でリレーを駆動する方法の模式図です。

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。





Wiring diagram

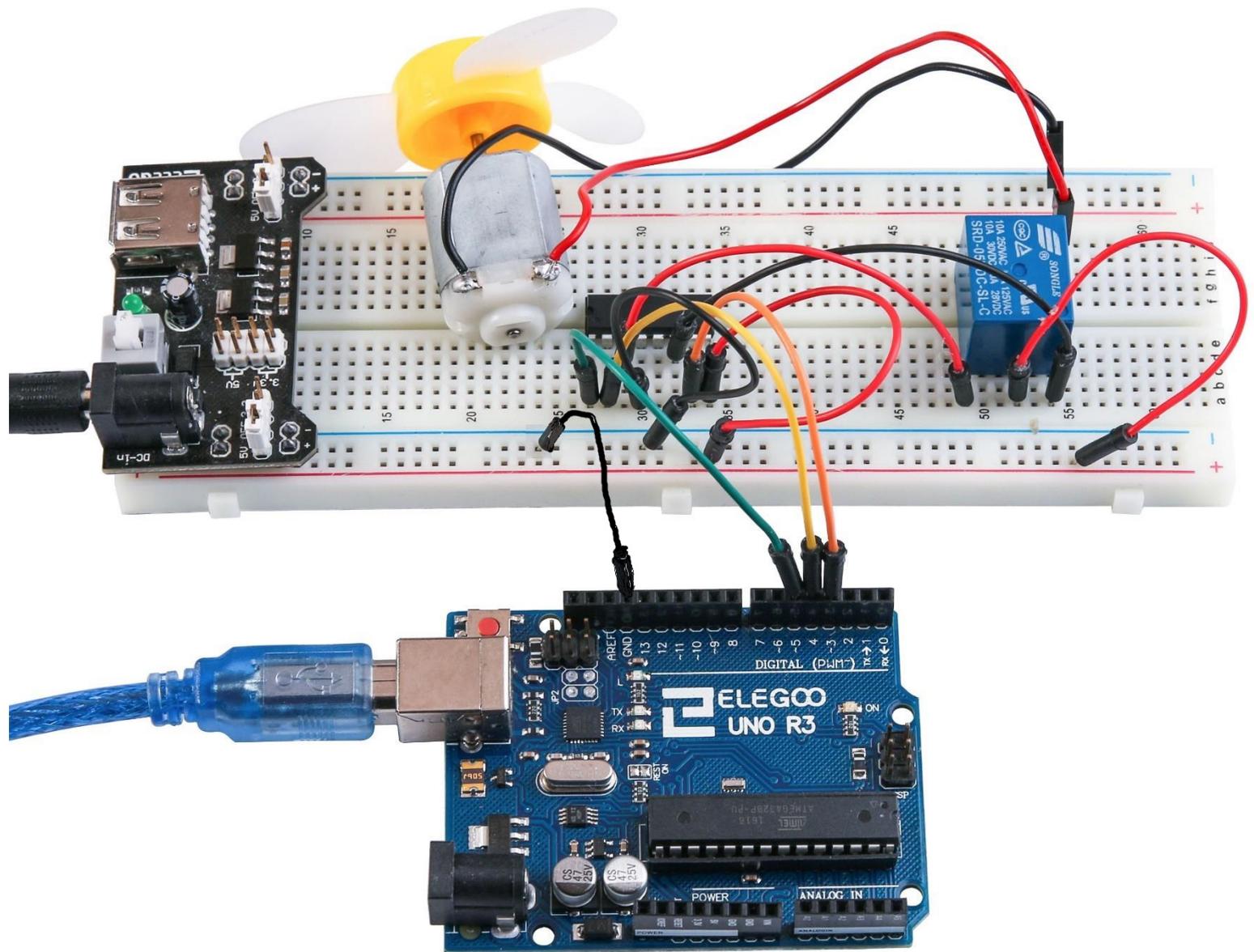


Code

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、モーターが停止します。

Example picture



Lesson 31 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。

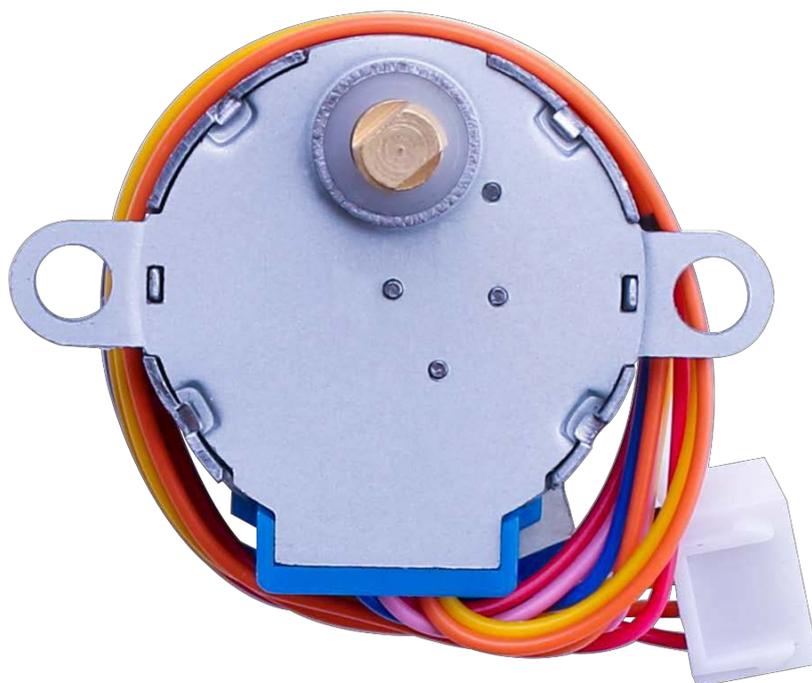
私たちが使用しているステッパーには、独自のドライバボードが付属しており、UNO に簡単に接続できます。

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)

部品の紹介

Stepper Motor



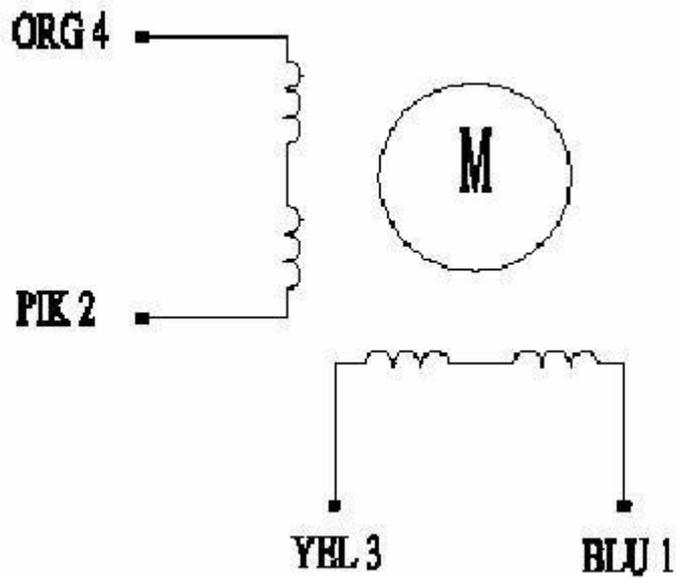
ステッパモータは、電気パルスを個別の機械的動作に変換する電気機械的装置である。ステッパモータのシャフトまたはスピンドルは、電気指令パルスが適切な順序で印加されると、離散的ステップ増分で回転する。モータの回転は、これらの印加された入力パルスに対していくつかの直接関係を持ちます。印加されるパルスのシーケンスは、モータシャフトの回転方向に直接関係します。モータシャフトの回転速度は、入力パルスの周波数に直接関係し、回転の長さは、印加される入力パルスの数に直接関係しています。ステッピングモータの最も重要な利点の1つは、開ループシステムで正確に制御できることです。開ループ制御は、位置に関するフィードバック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° /64
- Frequency: 100Hz
- DC resistance: 50Ω±7%(25°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 >10MΩ(500V)
- Insulated electricity power: 600VAC/1mA/1s
- Insulation grade: A
- Rise in Temperature <40K(120Hz)
- Noise <35dB(120Hz,No load,10cm)

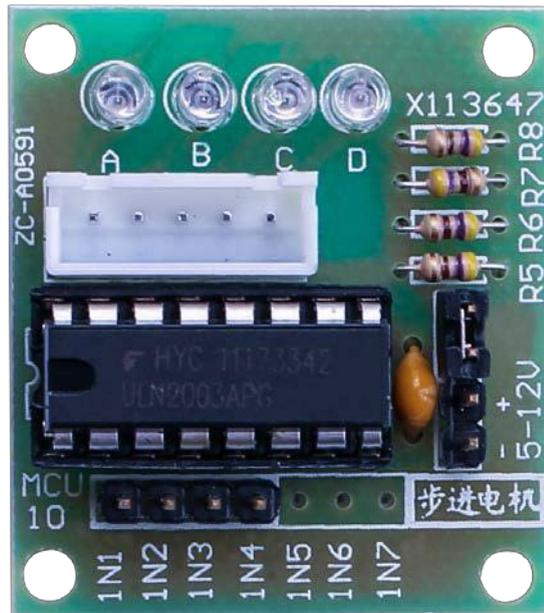
Interfacing circuits

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とは異なり、バイポーラステッパ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が、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 相ステップモータの動作状態を示します。
- ホワイトジャックは4相ステップ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。

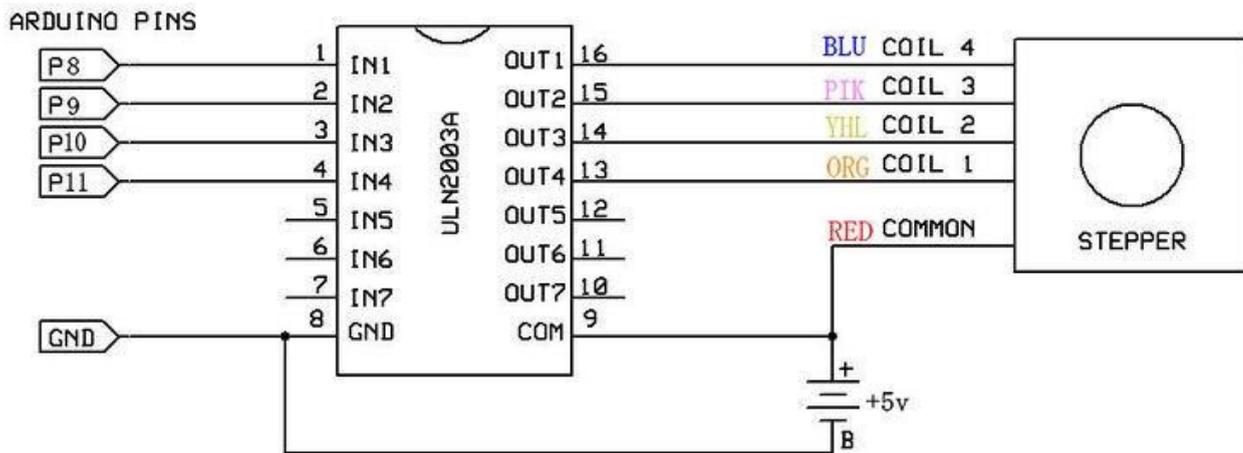
プロトタイプを作成するために、ULN2003 チップの残りのピンを保管しました。

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イクアウトを使用することです。ULN2003A には7つのダーリント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に似ています。ULN2003A は、チャンネルあたり最大 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。

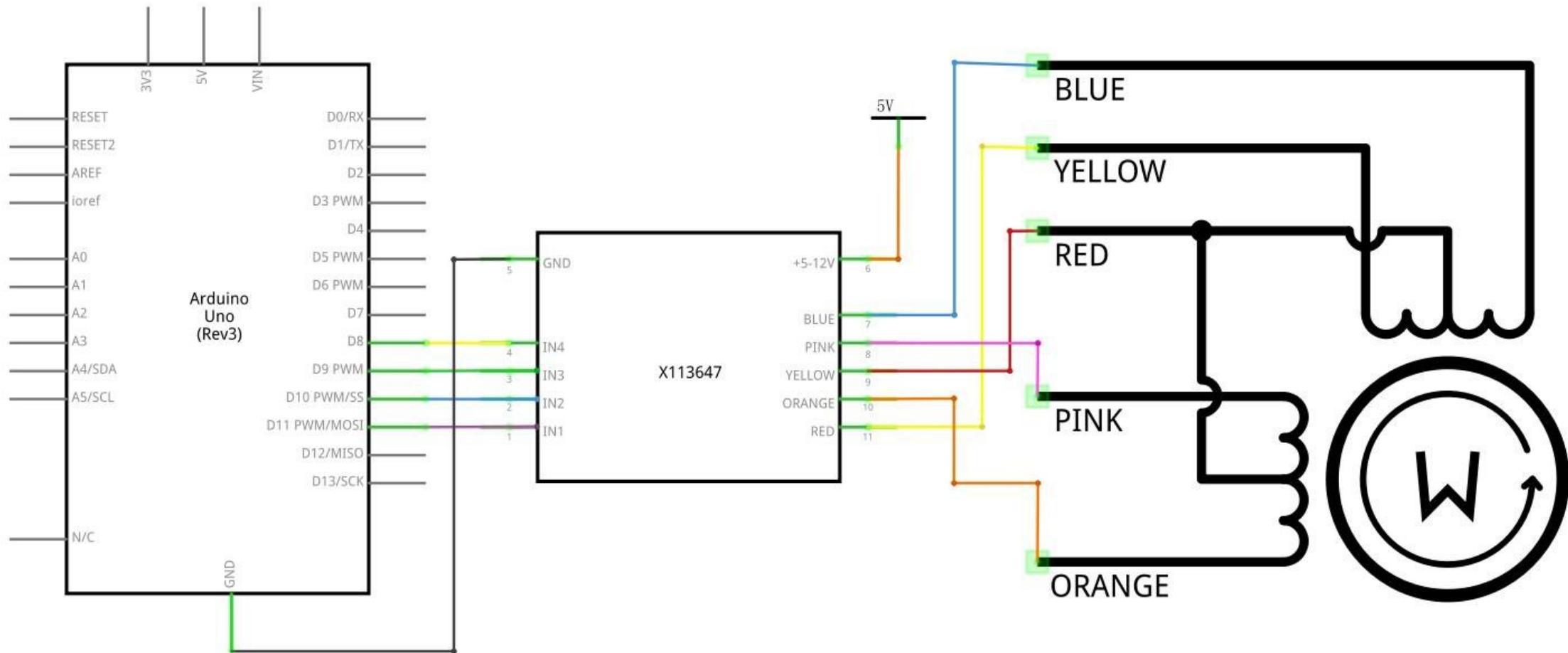
シーケンスは次のようになります:

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



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。グラウンドをUNOからステッパーモーターに接続します。

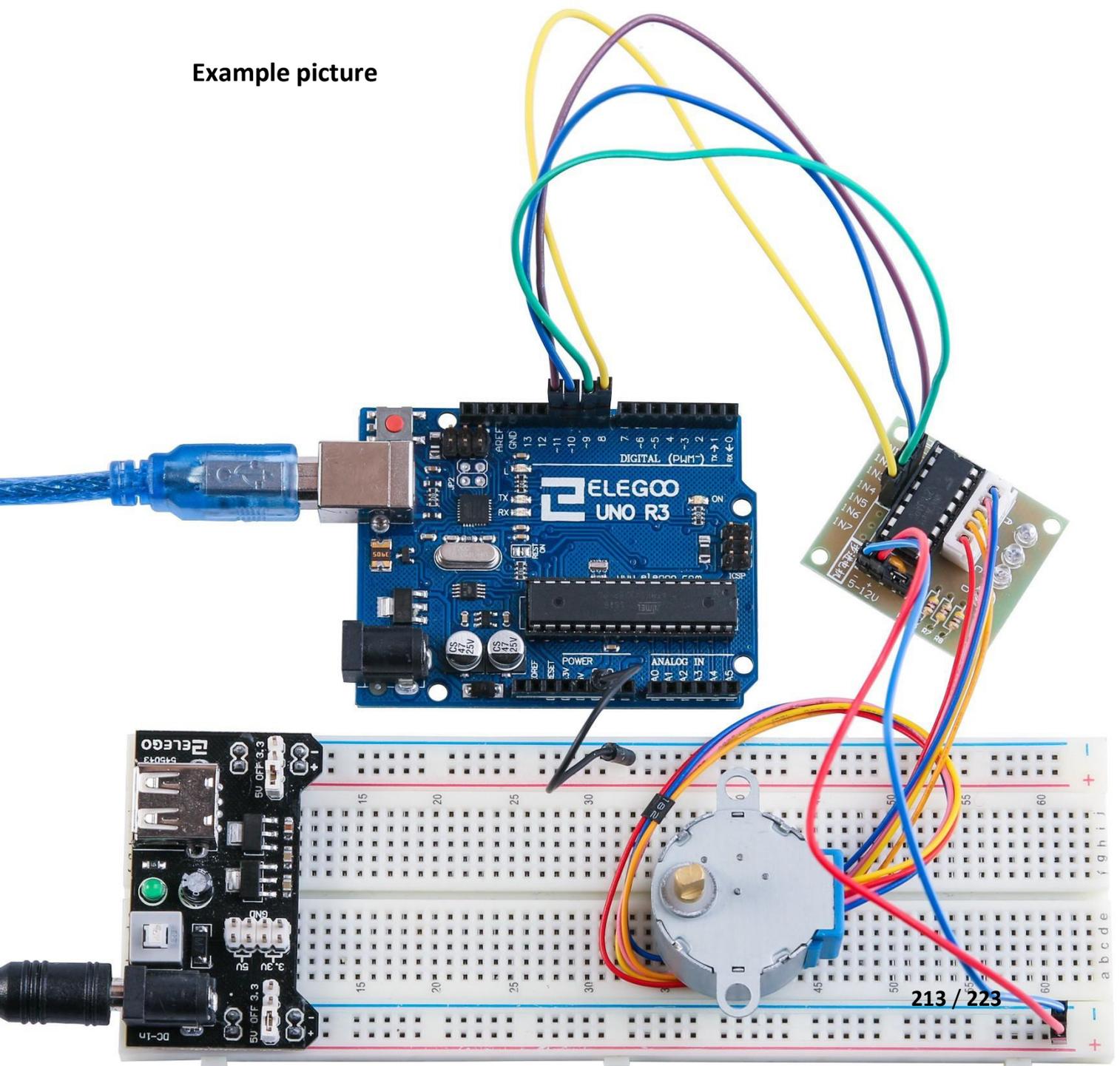
Code

配線後、コードフォルダのレッスン 31 ステッパーモータ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 32 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。

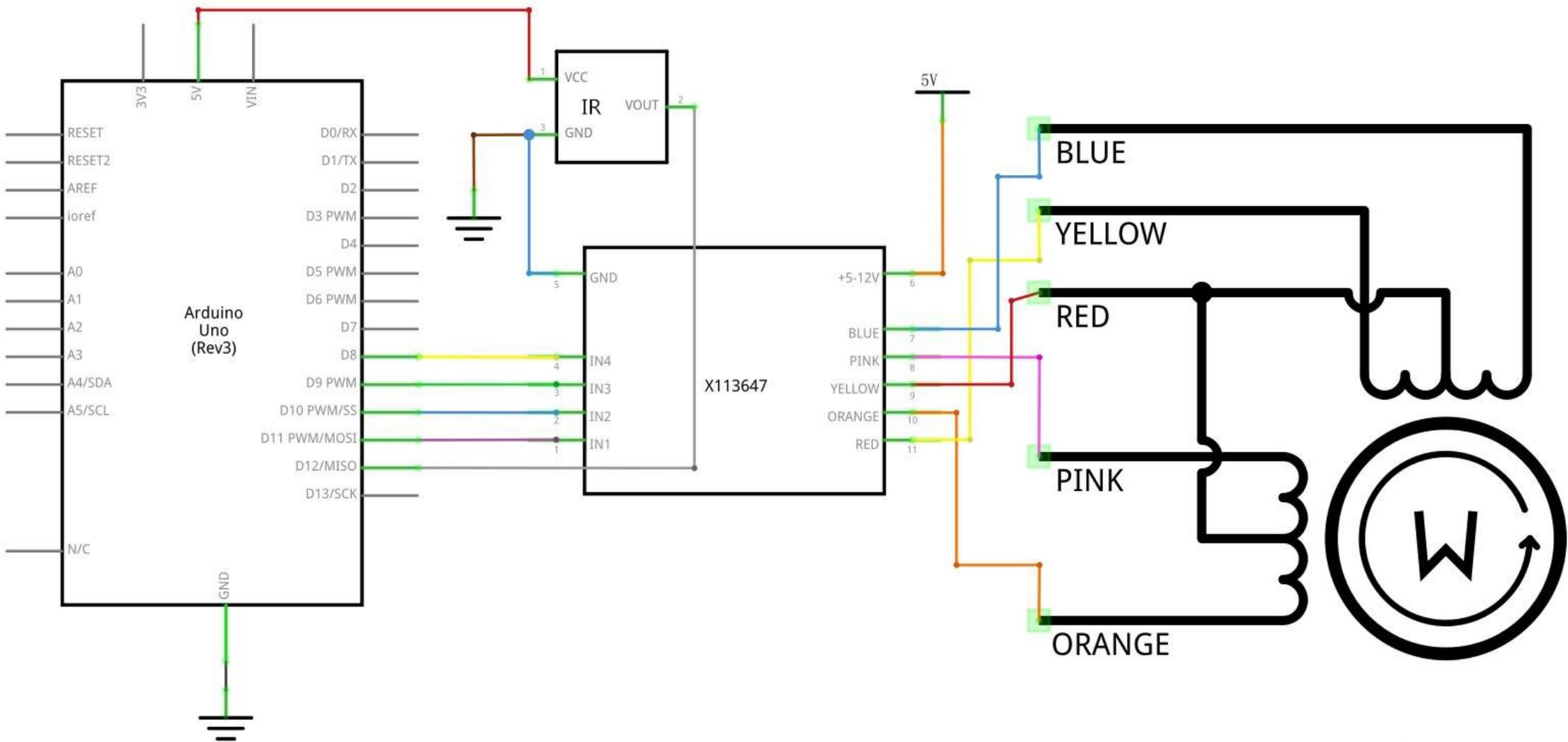
UNO から直接モータを駆動したくないので、ブレッドボードにプラグを差し込んで安価なブレッドボード電源を使用し、9V 1Amp 電源で電源を供給します。

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。

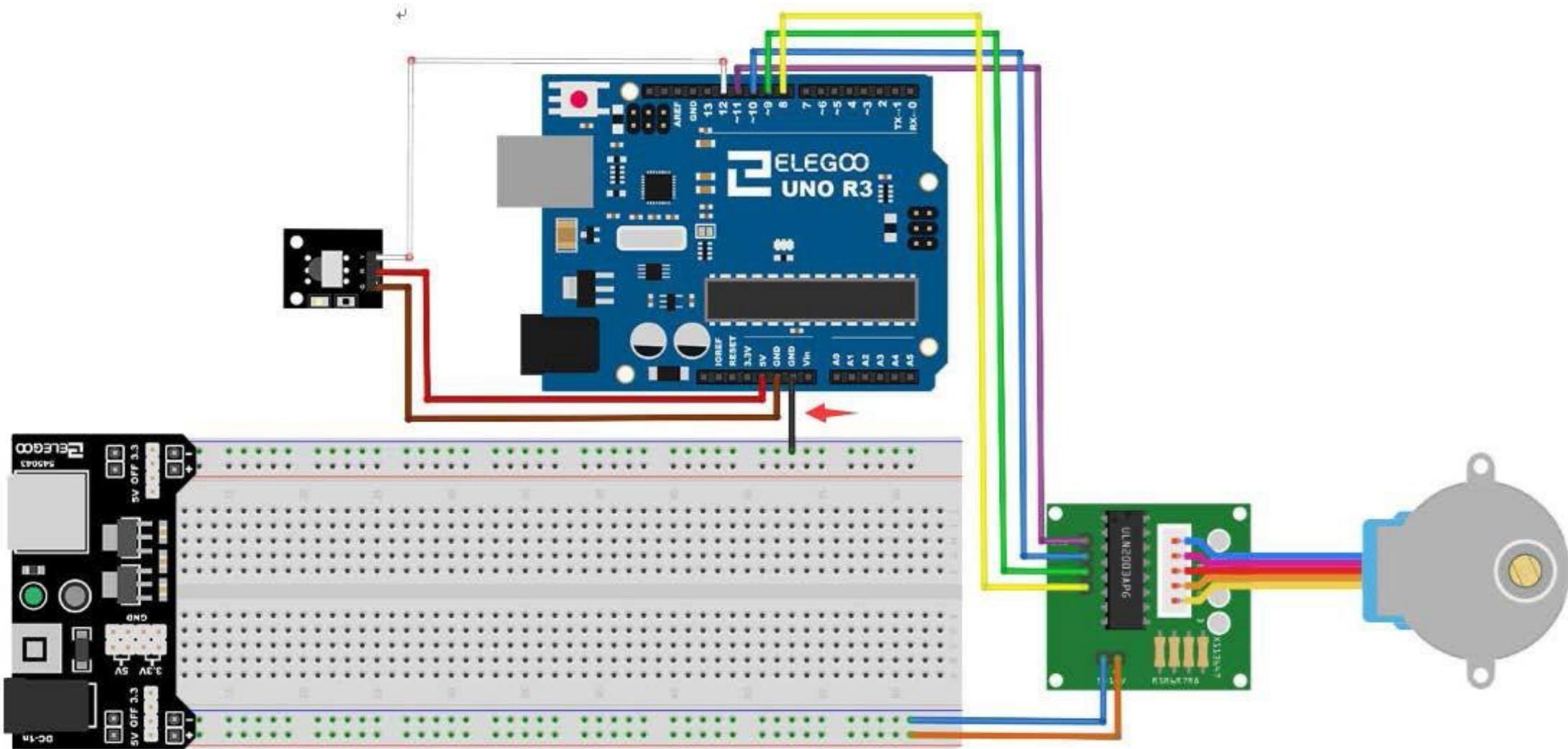
必要な構成部品:

- (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、ブレッドボード電源を使用してステッパモータ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

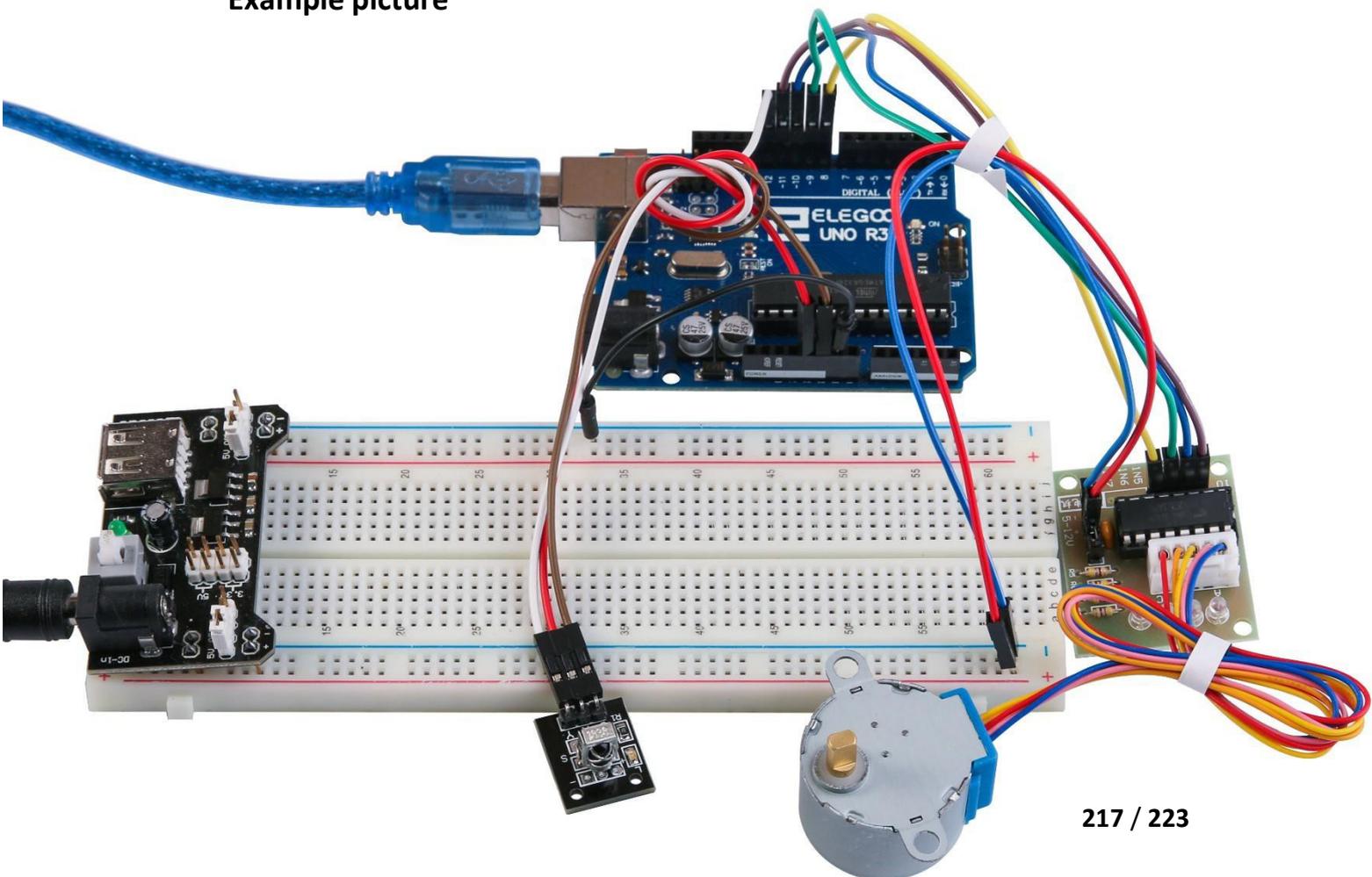
配線後、コードフォルダのプログラムを開きます - レッスン 32 ステッピングモータをリモートで制御し、プログラムをアップロードするにはアップロードをクリックしてください。 エラーがある場合のプログラムアップロードの詳細については、レッスン 2 を参照してください。

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。

Example picture



Lesson 33 ロータリーエンコーダ付きステッピングモータの制御

概要

このレッスンでは、ロータリーエンコーダを使用してステッピングモーターを制御する方法を学習します。独自のコントロールボードである安価で一般的なステッピングモーター：ULN2003 ボード付き 28BYJ-48 ステッパモーターを使用します。

28BYJ-48 モーターはあまり速くないか非常に強いわけではありませんが、初心者にとっては Arduino でステッピングモーターを制御することを試してみるの素晴らしいことです。

ロータリーエンコーダを回転させる方向にモーターを移動させるコードを記述し、取ったステップの数を記録して、モーターを開始位置に戻すことができます ロータリーエンコーダスイッチ。

必要な構成部品:

- (1) x Elegoo Uno R3
- (1) x 830 tie-points breadboard
- (1) x Rotary Encoder Module
- (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)

部品の紹介

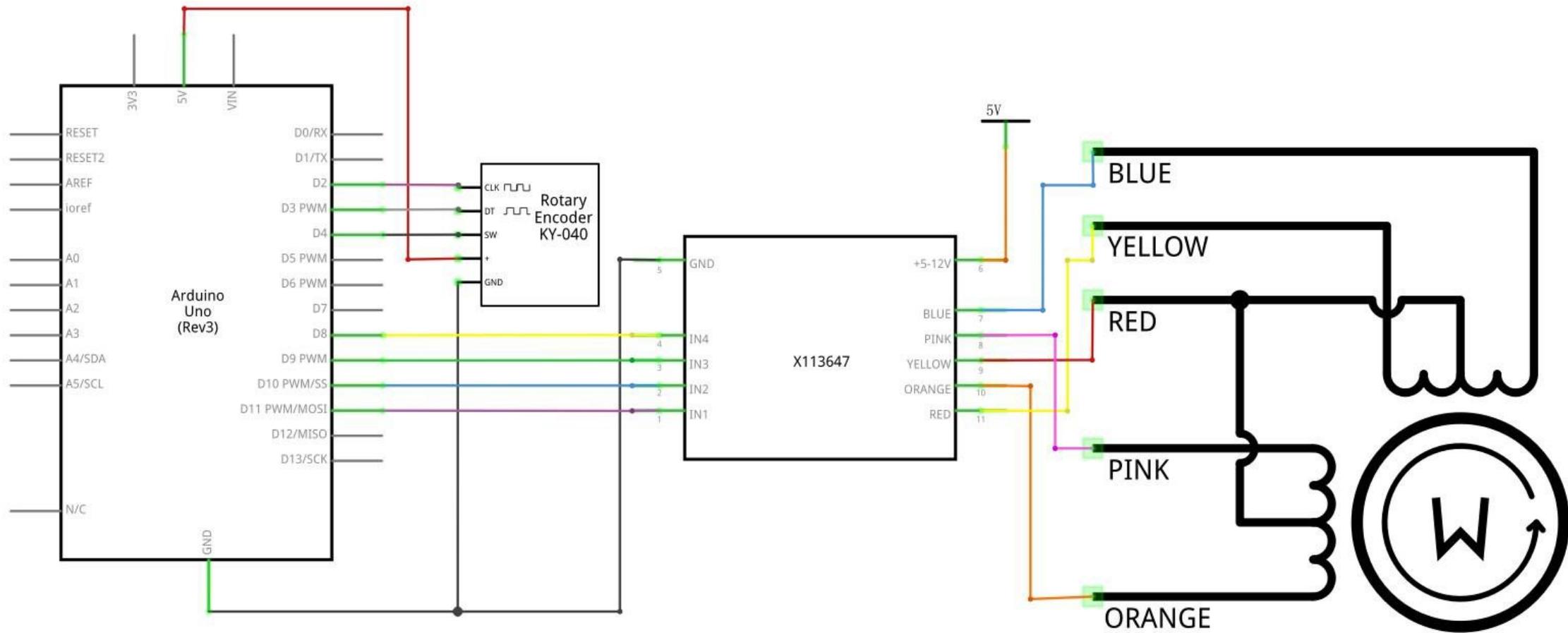
Rotary encoder

軸エンコーダとも呼ばれるロータリエンコーダは、軸または軸の角度位置または動きをアナログまたはデジタルコードに変換する電気機械的装置である。主なタイプには、絶対的なものと増分的なもの（相対的なもの）があります。アブソリュートエンコーダの出力はシャフトの現在位置を示し、角度変換器になります。インクリメンタルエンコーダの出力は、シャフトの動きに関する情報を提供し、典型的には、速度、距離、および位置などの情報にさらに処理される。

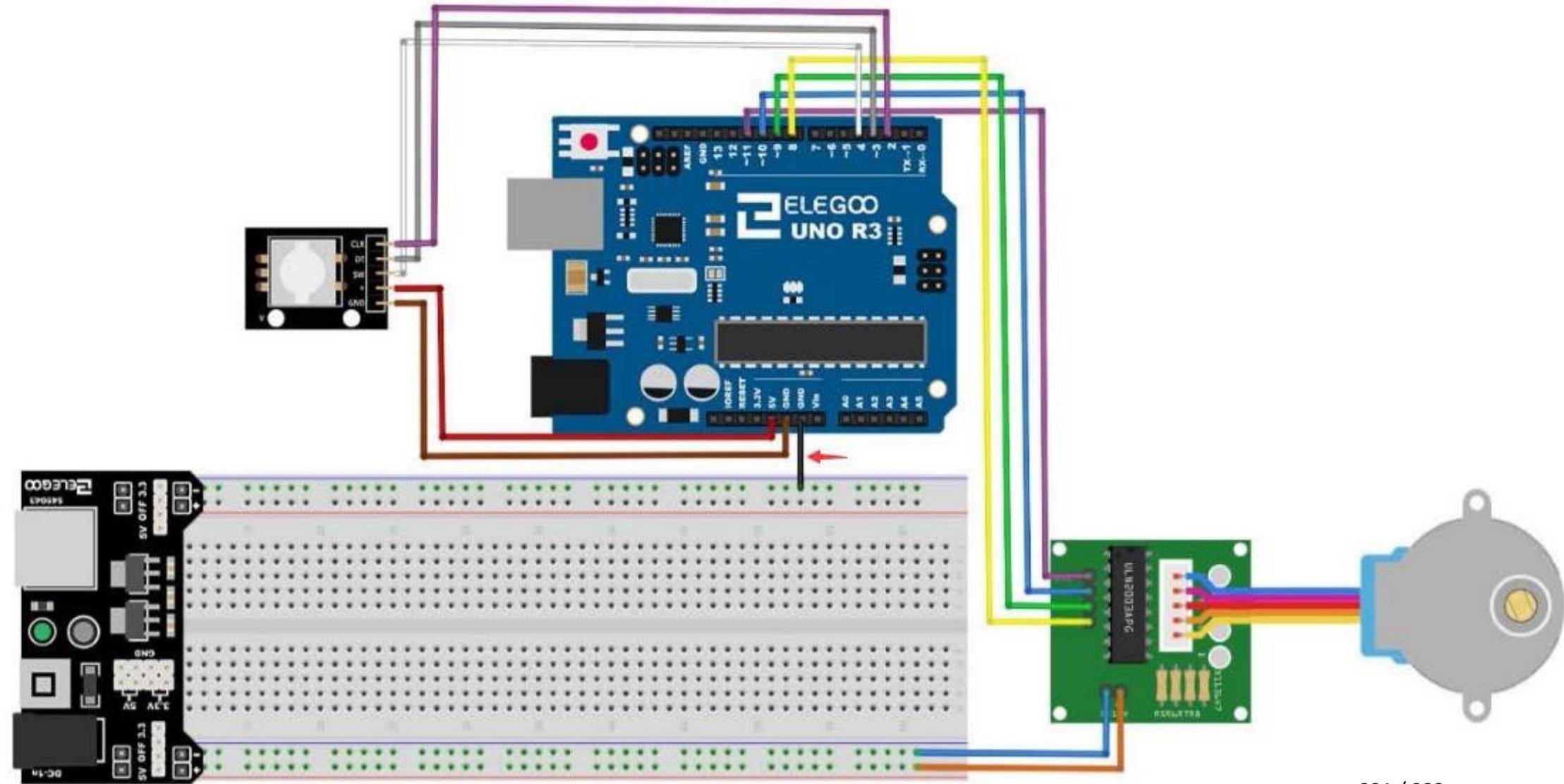
ロータリーエンコーダは、工業用コントロール、ロボット、特殊目的の写真用レンズ、オプトメカニカルマウスやトラックボールなどのコンピュータ入力デバイス、制御された応力レオメータ、回転するレーダプラットフォームなど、精密なシャフトの無限回転を必要とする多くのアプリケーションに使用されています。



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、ロータリーエンコーダモジュール用に3本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、ピン2-4はロータリーエンコーダから情報を受け取ります。

5VとグラウンドをUNOからロータリーエンコーダに接続し、予防措置として、ステッピングモーターに電力を供給するためにブレッドボード電源を使用して、UNOが提供できる電力をより多く使用できるようにします。

UNOグラウンドをブレッドボードに接続して参考にします。

Code

配線後、プログラムをコードフォルダ - レッスン 33 ロータリーエンコーダ付きステッピングモーターの制御で開いて、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、<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。

ステッピングモーターの位置を追跡して開始位置に戻ることができるようにするため、現在の位置を格納する変数をいくつか使用しています。

ロータリーエンコーダにステップがないことを確認するためのエラーチェックコードもいくつか含まれています。これはモーターの位置が不正確になるためです。

Example picture

