

・NeoPixel RGB LED テープを使う

《何ができる?》光の 3 原色 (RGB = 赤緑青) を一つの LED にまとめたものを RGB LED という。

通常は弾丸型の LED の中に RGB の 3 本のアナログ出力を LED の数の分だけ制御する必要があったが、最近では、データ出力 1 本でシリアル制御するタイプの RGBLED が増えてきた。

今回は、NeoPixel というタイプの LED テープを用いて複数の RGB LED を制御する方法を学ぶ。



右図にあるような長いテープを LED が必要な個数分だけ切り取って利用する。演習では 5 個の RGB LED を、IO ピン 1 本だけで制御する。

通常の RGB LED だと、RGB 3 本×5 個の LED=15 本の制御ピンが必要になる。

制御するピンの本数が 1 本になるメリットがあるが、あらかじめライブラリを準備して読み込ませる必要がある。

図 1 ALITOVE LED テープ WS2812B アドレス指定可能 LED テープライト 5m300LED 5050RGB SMD ピクセル ドリームカラー 非防水 ホワイト PCB D C 5 V

演習

【目標】

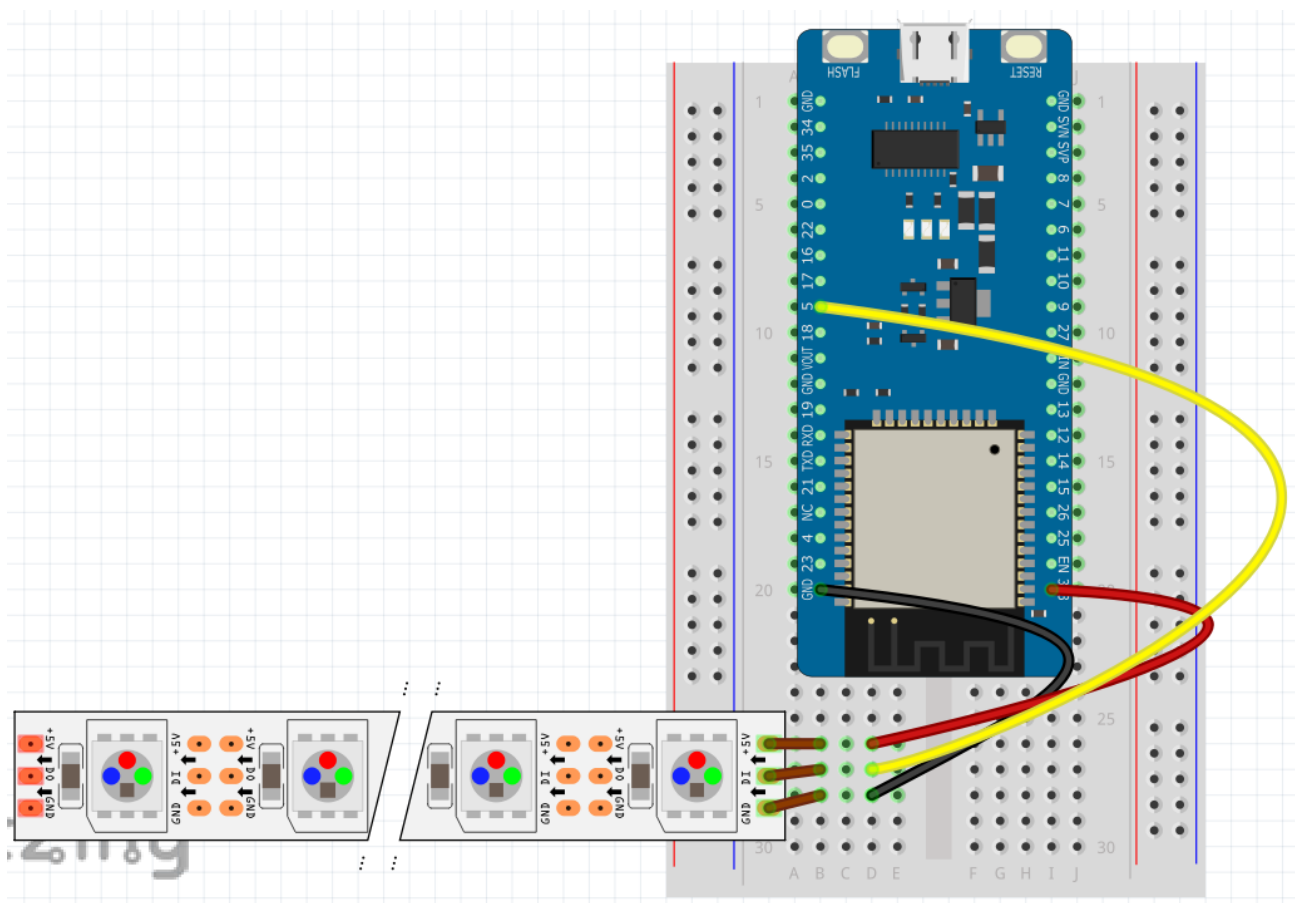
NeoPixel RGB LED を用いて、複数の LED のフルカラー制御ができるようになる。NeoPixel もそうだが、デバイス用のライブラリをあらかじめ入手して、ESP32 で利用できるようにする必要がある。インターネットからライブラリを入手し適切なインストールを行ってデバイスを活用できるようになることが目標である。

【1. ESP32 と電子工作部品との接続】

1. 1. 必要な部品

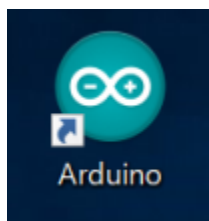
パーツ名	必要個数
NeoPixel RGB LED テープ (5)	1 本
ジャンパーコード	赤色、黄色、黒色 各 1 本ずつ ※上記の色が揃わなければ別の色でも良い

以下の接続を行う。以下の実体配線図を元に接続し、動作を確認すること。



- ① NeoPixelRGB LED をブレッドボード 1 (ESP32 が載っている方) の a26 から a28 に差し込む。NeoPixel テープのシルク印刷で 5V と印字されたピンが g26 に刺さるようにする。
- ② ジャンパーワイヤー (赤色) をブレッドボード 1 の c26 へ、もう片方を ESP32 の 3V3 ピンへ。
- ③ ジャンパーワイヤ (黄色) をブレッドボード 1 の c27 へ、もう片方を ESP32 の I05 ピンへ。
- ④ ジャンパーワイヤ (黒色) をブレッドボード 1 の c28 へ、もう片方を ESP32 の GND ピンへ。

【2. Arduino スケッチのサンプルプログラムを実行】

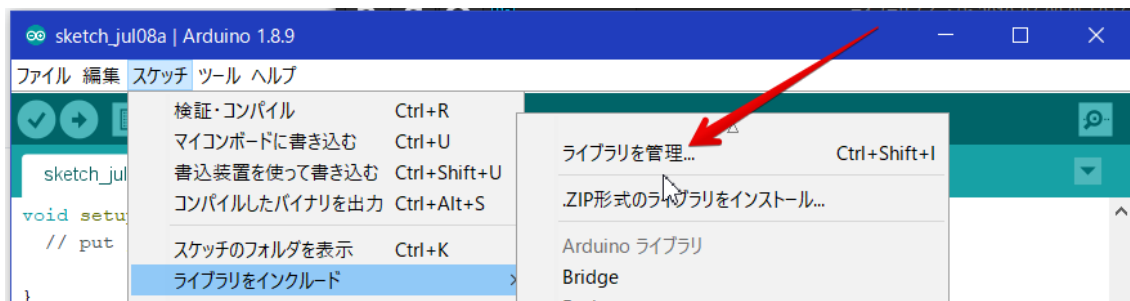


デスクトップのアイコンをダブルクリックして
Arduino IDE を起動する。

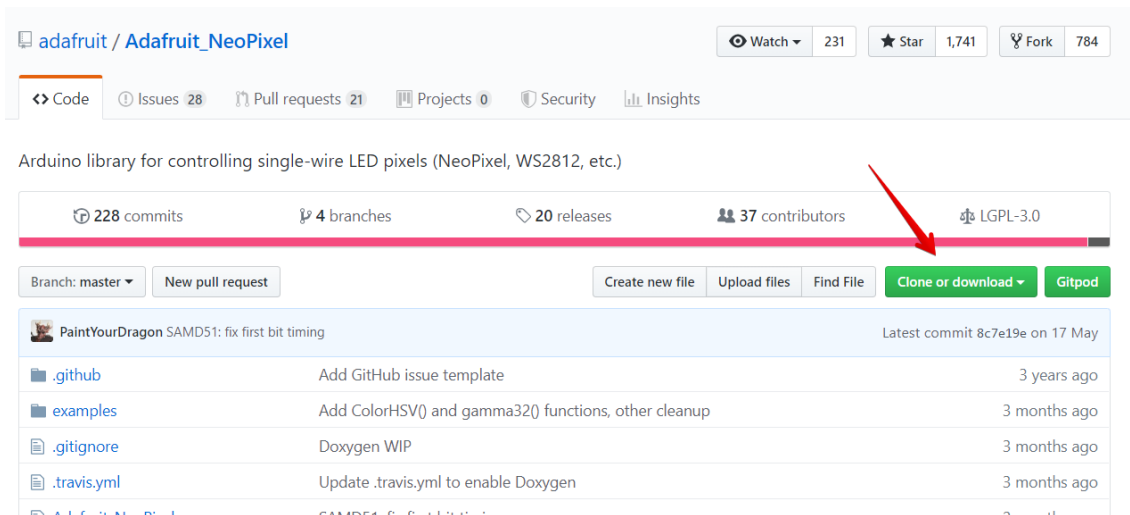
【課題】プロジェクト名「kad17_strandtest」

NeoPixel 接続用のライブラリをダウンロードして libraries フォルダに格納しておく必要がある。
ライブラリの追加方法は2パターンある。

1. Arduino IDE の[スケッチ]メニューから「ライブラリを管理」を選んでダウンロードする方法

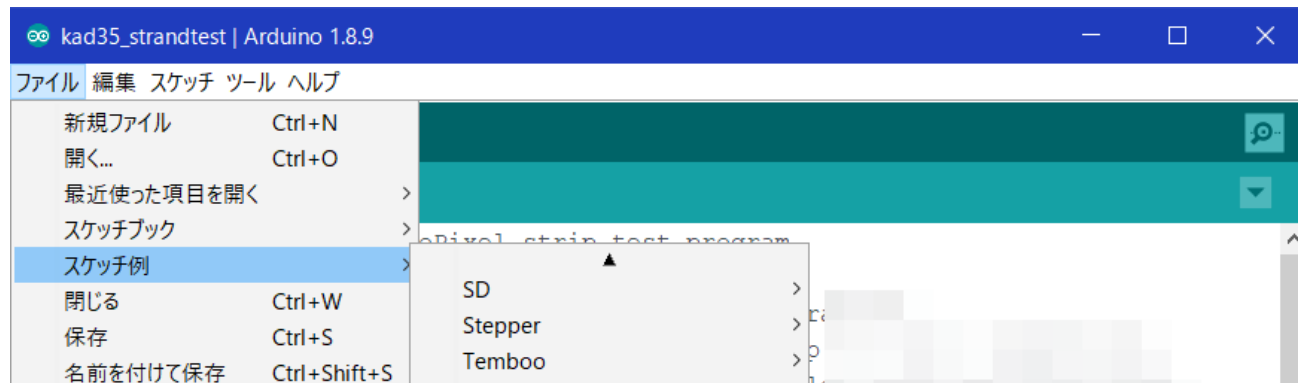


2. Github などで公開されているライブラリのソースコードをダウンロードして libraries フォルダに配置する方法 (Arduino IDE の再起動が必要)

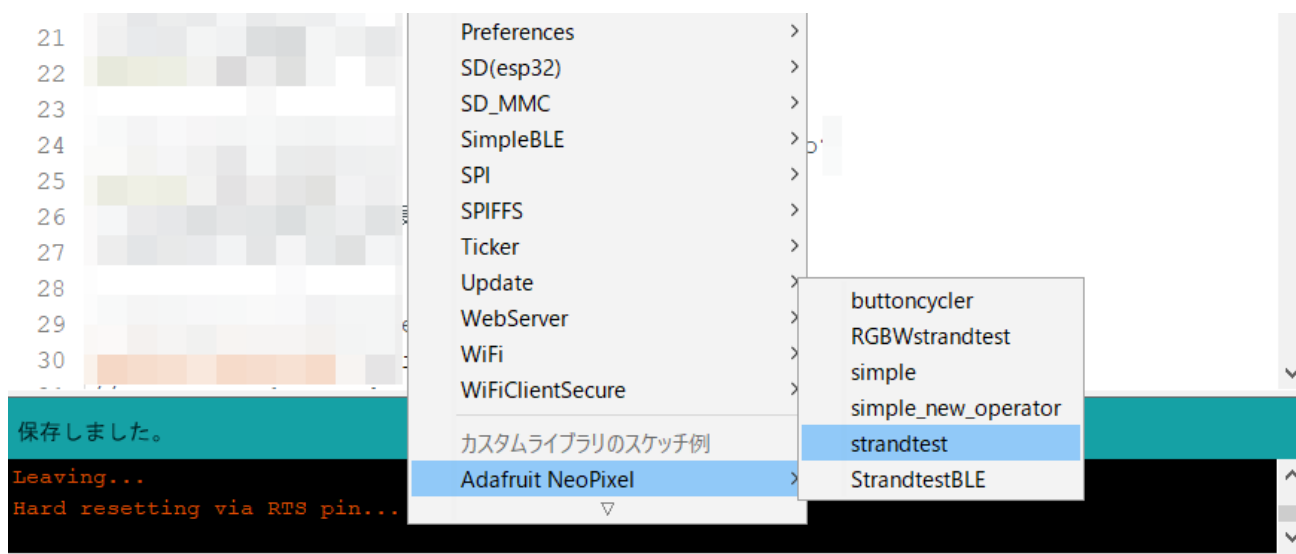


いずれにしても自分でやろうとなると、Google 検索でライブラリ入手方法を見つける必要がある。自分で検索して、NeoPixel のライブラリを自分の Arduino IDE 開発環境で使えるようにする。

もし代表的なライブラリが入手出来たら、「スケッチ例」には以下のようなサンプルソースが並ぶ。



<中略>



このうち、「strandtest」という名前のサンプルスケッチがあればそれを使う（もし、サンプルスケッチ「strandtest」が無い場合は先生に申告する）。

サンプルスケッチ「strandtest」を開いたら、そのままでは保存できないので、[ファイル]-[名前をつけて保存...]で、「kad17_strandtest」というプロジェクト名に変えて保存する。

また、そのままでは実行できない。少なくとも以下の2点を変更してからボードへの書き込みを行う。

- DI (DATA INPUT) : データ線は、GPIO の 6 番ピンに接続する指定になっている
- RGB LED の個数は 60 個指定になっている

RGB LED が、さまざまな光をだすようになれば OK。

【課題】 プロジェクト名「kad18_function」

前の課題を元に、以下の関数を作成して、任意の LED を光らせるようにする。

```
void colorDotOn(int i, uint32_t color)
```

引数 : int i ...LED の番号。0 から 4 までの値。0 が一番入力端子に近い側になる

引数 : uint32_t color ...RGB 値 (32 ビット符号なし整数で表される)

戻り値 : なし

```
void colorDotOff(int i)
```

引数 : int i ...LED の番号。0 から 4 までの値。0 が一番入力端子に近い側になる

戻り値 : なし

メインループとなる loop() 関数と上記 2 つのユーザー定義関数は以下の様なスケッチをひな型として使って、任意の場所の LED のオン・オフができるようにする。

プロジェクト名「kad18_function」の loop 関数と 2 つのユーザー定義関数のテンプレ(ひな型)

```
void loop() {
    //colorWipe(strip.Color(255, 0, 0), 50); // Red
    colorDotOn(0, strip.Color(255, 0, 0)); //0 番目(一番下) が赤色に光る
    delay(1000);
    colorDotOff(0); //Red 0 番目(一番下) が消える
    delay(1000);

    colorDotOn(4, strip.Color(0, 255, 0)); //4 番目(一番上) が緑色に光る
    delay(1000);
    colorDotOff(4); //Red 4 番目(一番上) が消える
    delay(1000);
}

/**
 * colorDotOn
 */
void colorDotOn(int i, uint32_t color){
    //ここに i 番目の LED が color 色に光る処理を入れる
}

/**
 * colorDotOff
 */
void colorDotOff(int i){
    //ここに i 番目の LED が消灯する処理を入れる
}

/**
 * colorWipe
 */
void colorWipe(uint32_t color, int wait) {
    for(int i=0; i<strip.numPixels(); i++) { // For each pixel in strip...
        strip.setPixelColor(i, color);        // Set pixel's color (in RAM)
        strip.show();                          // Update strip to match
        delay(wait);                          // Pause for a moment
    }
}
```

上記テンプレはコピー＆ペーストすればよい。

【実行結果】

以下の処理をくり返す

- 1 番下の LED が赤色に 1 秒光る
- 1 番下の LED が消灯し 1 秒経過
- 1 番上の LED が緑色に 1 秒光る
- 1 番上の LED が消灯し 1 秒経過

【課題】 プロジェクト名「kad19_function2」

前の課題を元に、以下の関数を作成して、任意の LED を光らせるようにする。

```
void colorRun(uint32_t color)
```

引数 : uint32_t color …RGB 値。ここで指定された色が LED テープ上を走る

引数 : int wait …光る時間(ミリ秒)

戻り値 : なし

メインループとなる loop() 関数と上記のユーザー定義関数は以下の様なスケッチをひな型として使って、LED の点滅が下から上に向かって走るようなスケッチに仕上げる。

プロジェクト名「kad19_function2」の loop 関数と 1 つのユーザー定義関数のテンプレ(ひな型)

```
void loop() {
  //colorWipe(strip.Color(255, 0, 0), 50); // Red

  colorRun(strip.Color(255, 0, 0), 50); //赤が 50 ミリ秒おきに下から上へひとつずつ点灯
  delay(1000);
  colorRun(strip.Color(0, 255, 0), 50); //緑が 50 ミリ秒おきに下から上へひとつずつ点灯
  delay(1000);
}

/**
 * colorRun
 */
void colorRun(uint32_t color, int wait) {
  // ここに color 色の LED が 0 番目から wait ミリ秒おきに点滅する処理を入れる
}
```

上記テンプレはコピー＆ペーストすればよい。

【実行結果】

以下の処理をくり返す

LED が一番下の 0 番目から赤色に光 50 ミリ秒おきに下から上へひとつずつ点灯、消灯し
一番上まで行ったら 1 秒経過

LED が一番下の 0 番目から緑色に光 50 ミリ秒おきに下から上へひとつずつ点灯、消灯し
一番上まで行ったら 1 秒経過

上記課題ができたなら、loop 関数内を書き換えて色々な模様が出る処理に書き換えてみる。