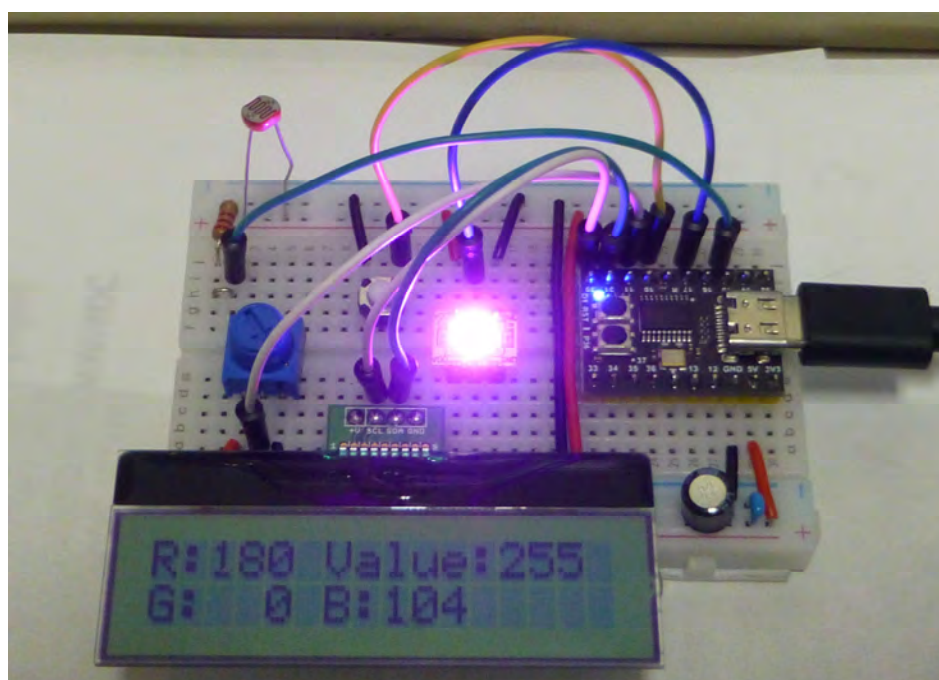


神戸高専 夏季公開講座

2023年8月17日

マイコンでカラーLEDを制御してみよう



電子工学科

笠井 正三郎

本講座で行うこと

• マイコンを用いてカラーLEDを制御

キーワード

- マイコン（マイクロコントローラ）
 - プログラムにより動作が変えられる
 - ➔ PCでプログラムを作って、マイコンに書き込む
- カラーLED
 - 様々な色は、R(赤)、G(緑)、B(青)の合成で表現される。
 - 3つのLEDが1つになったLEDモジュール
 - * 実はカラーLEDの中にもマイコンが. . .

本日のコース

1. マイコンの動作確認 Lチカ

PCとマイコンボードを接続して、LEDを点滅動作させるプログラムを書き込み、基本動作の確認

2. カラーLEDの接続と点灯動作

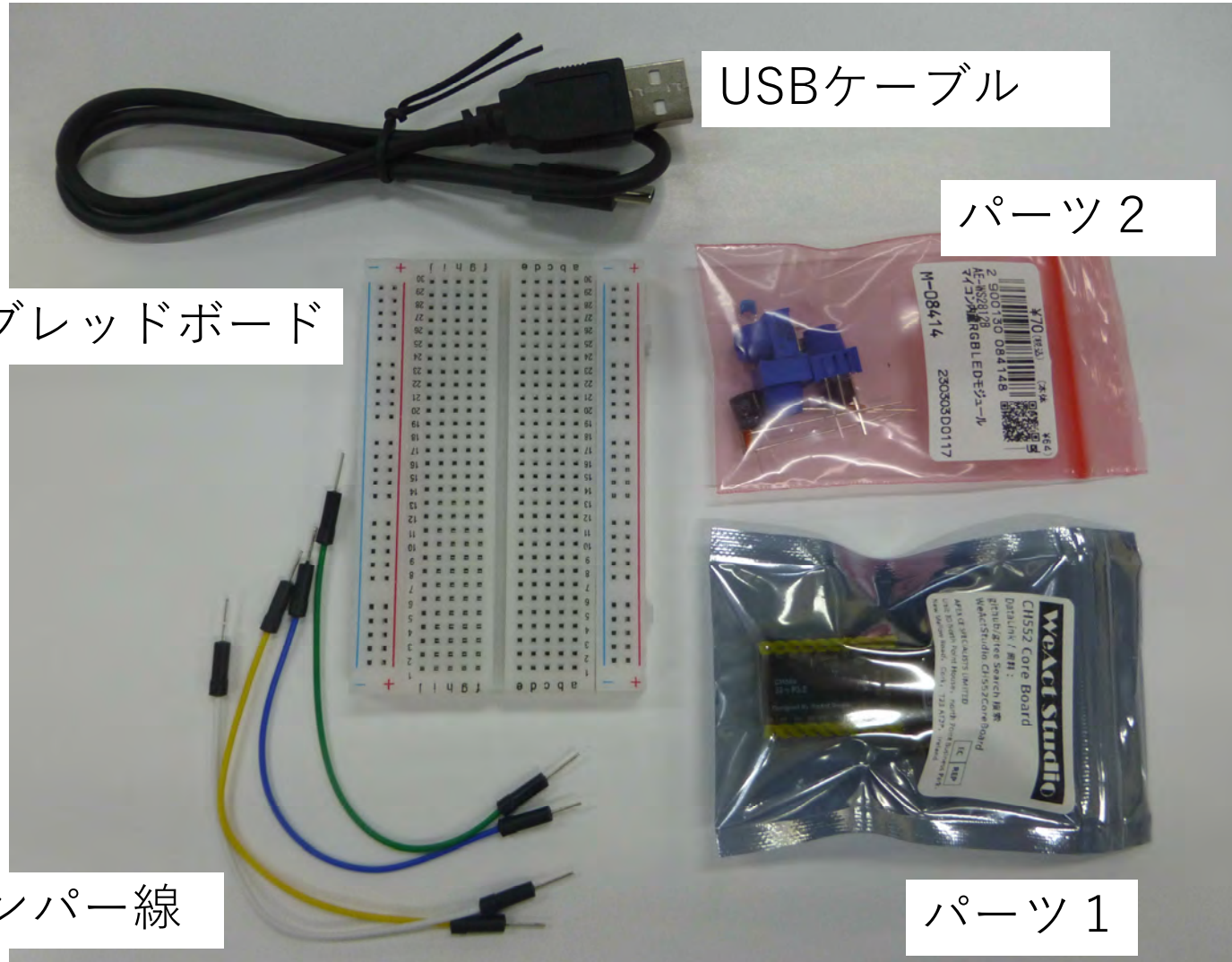
青・黄・赤の3色を1秒間隔で移動

3. 3個の可変抵抗器によりR・G・Bの色を制御

可変抵抗器により様々な色を表現できることを確認

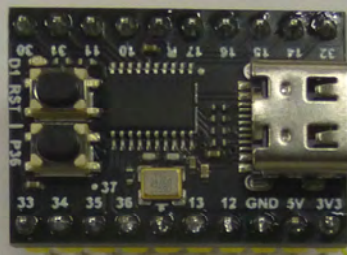
4. 1個の可変抵抗器あるいは明るさセンサで色を変化させる

配布パーツ

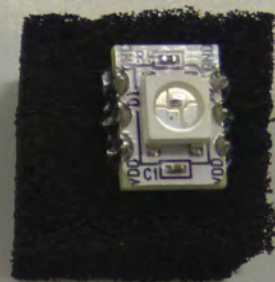


パーツ1

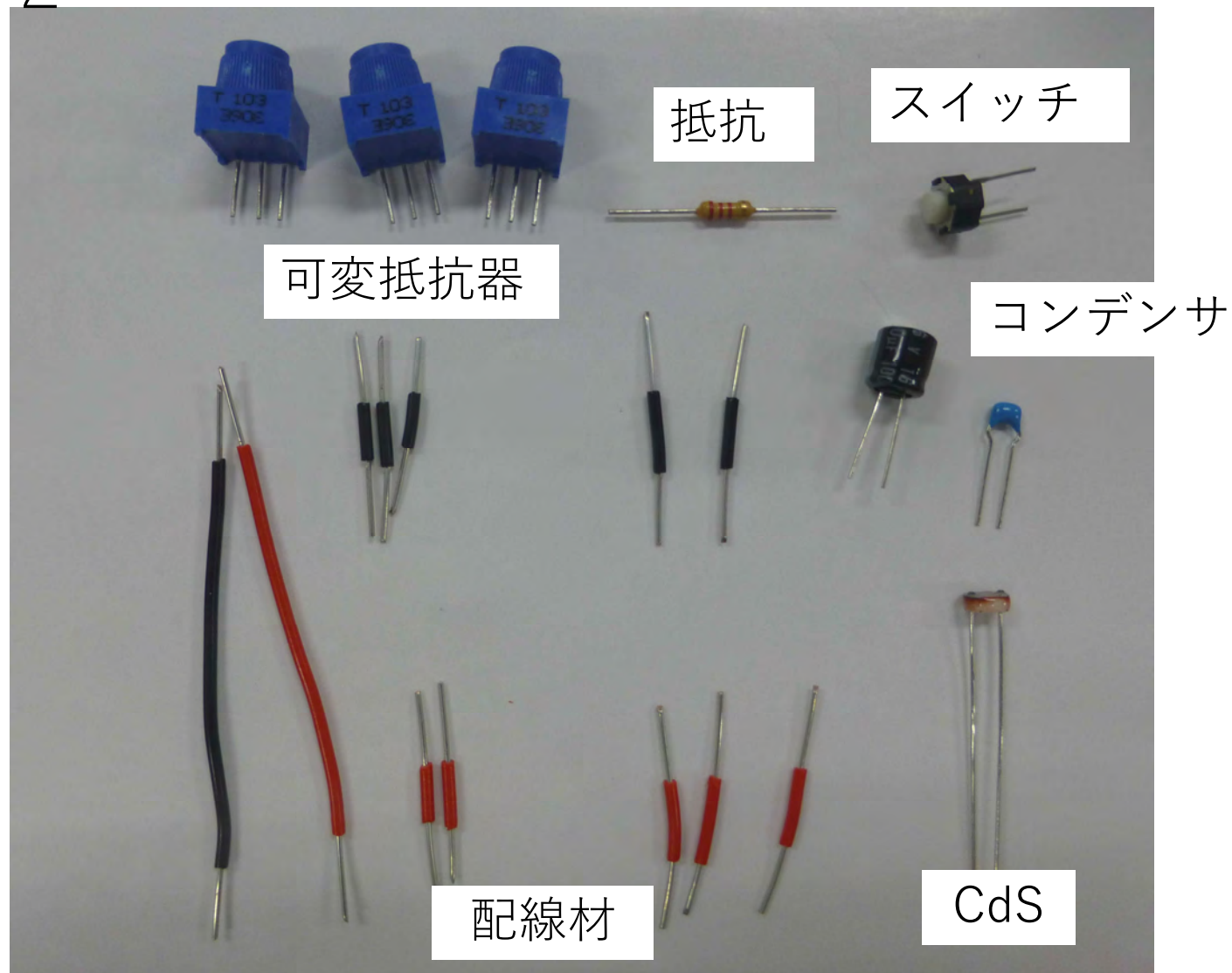
マイコンボード



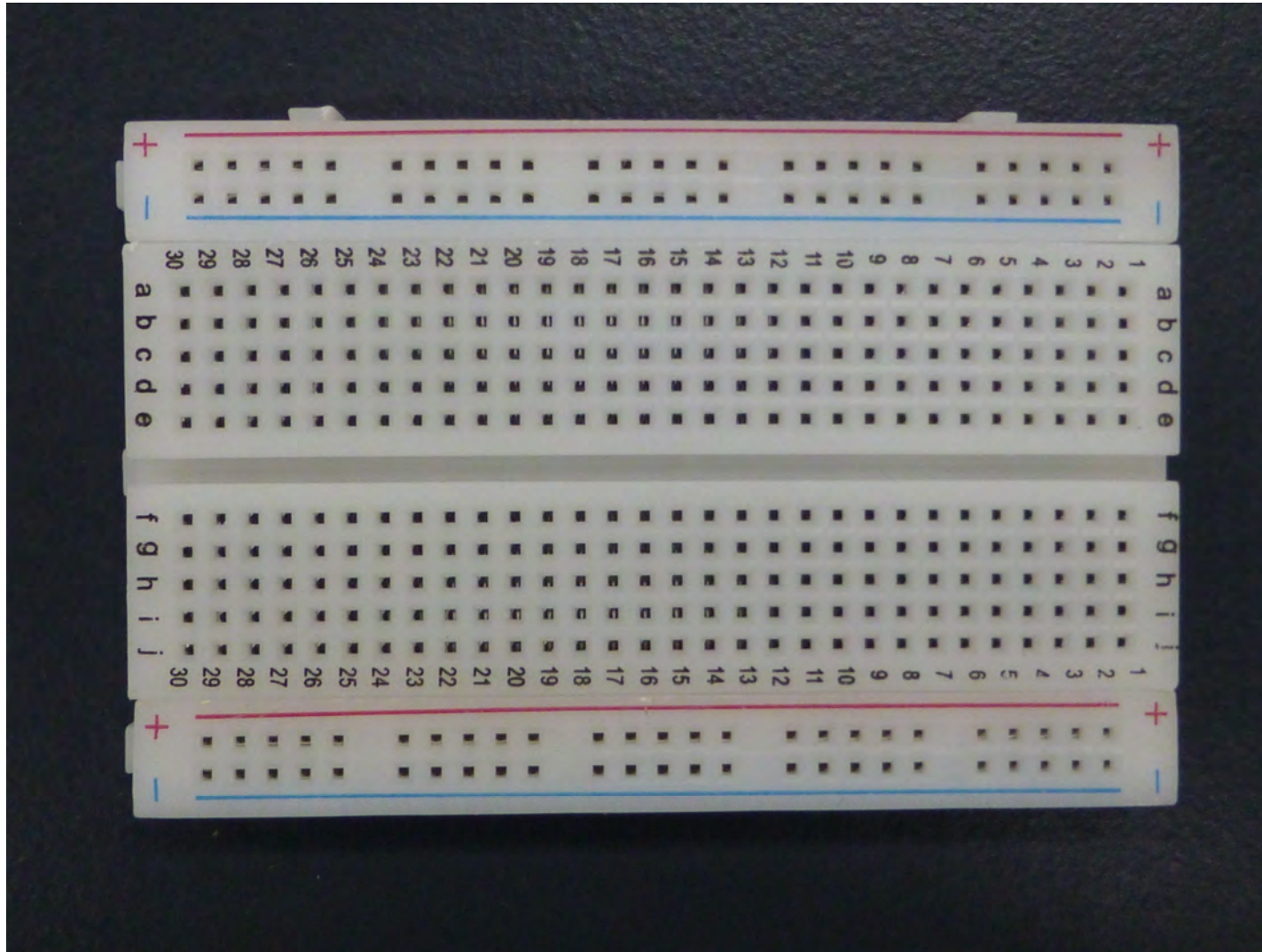
カラーLED



パーツ2



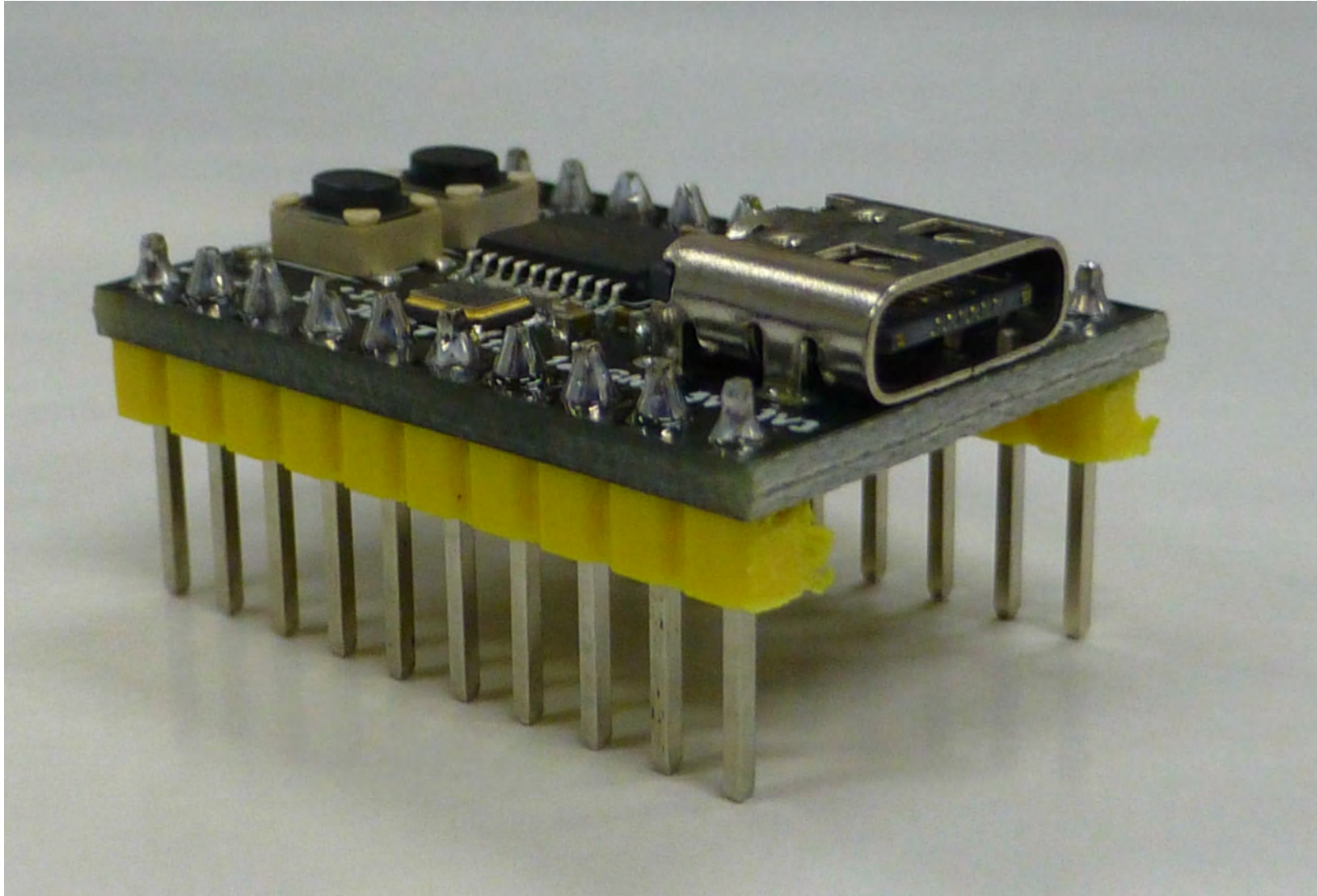
ブレッドボード（マイコンと素子を接続）



上下の赤・青
横につながっています。
*電源用として使用

真ん中の溝を境に
縦につながっています。

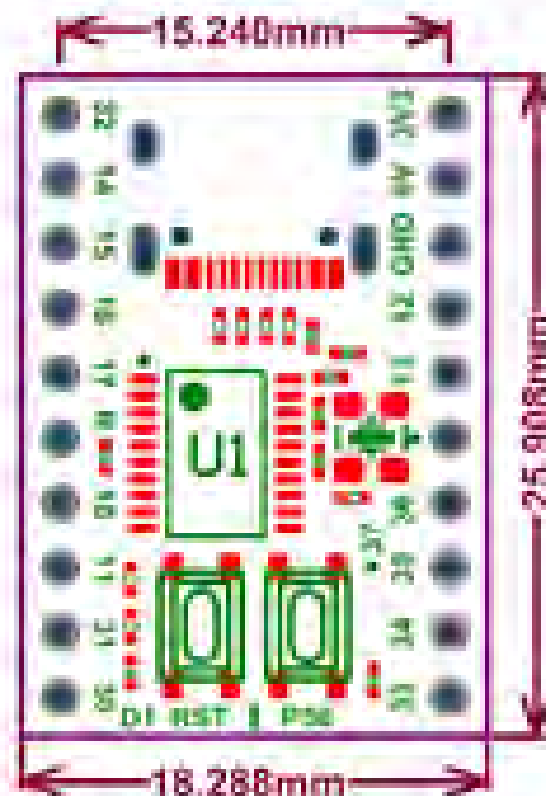
マイコンボード



WeAct Studio

AliExpress.com

1個～250円ぐらい



Freq. 24Mhz Max

ROM 16KB

RAM 1280B

USB2.0 FS/PWM/ADC

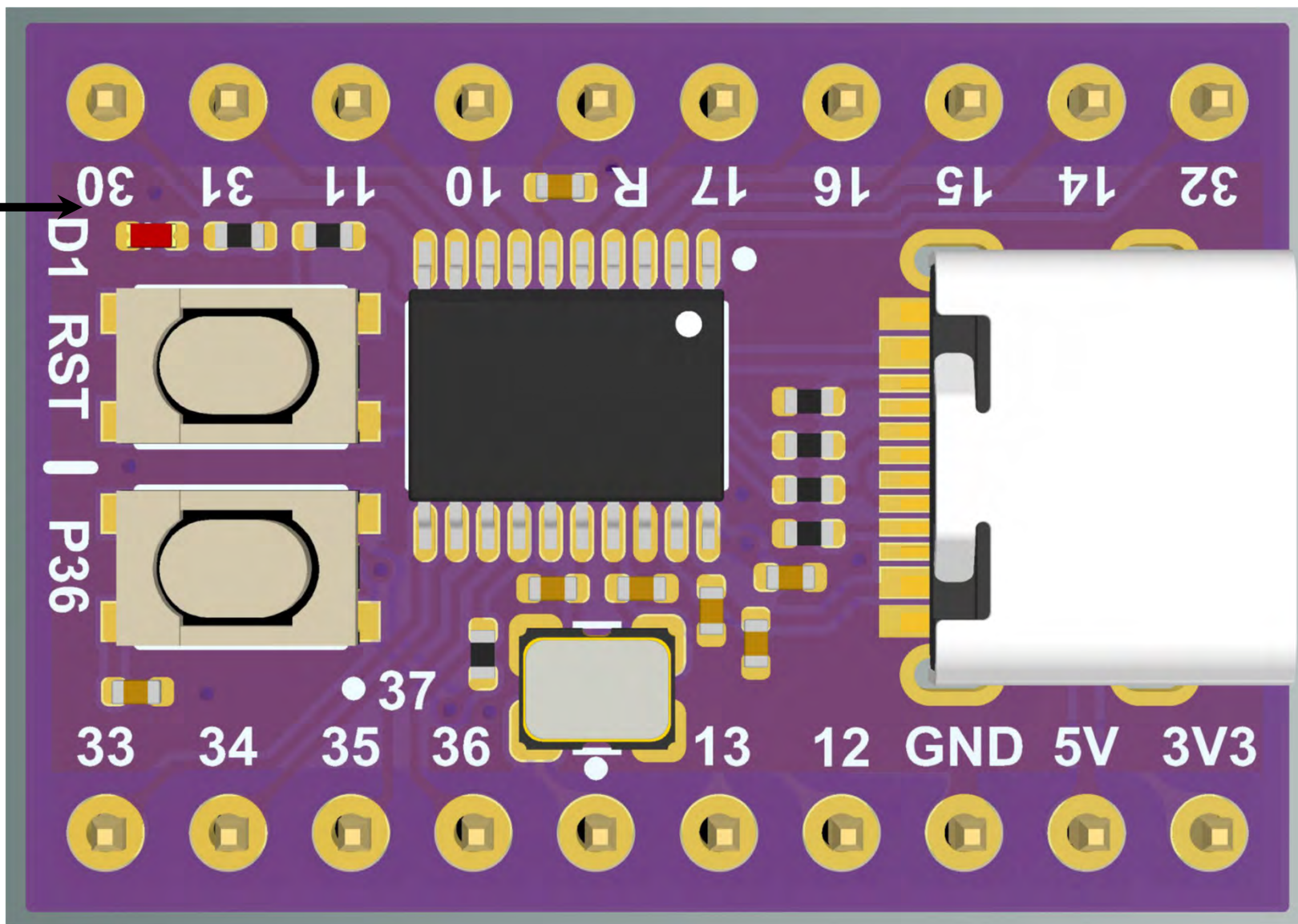
Touch Key

SPI/Uart/Watch-dog

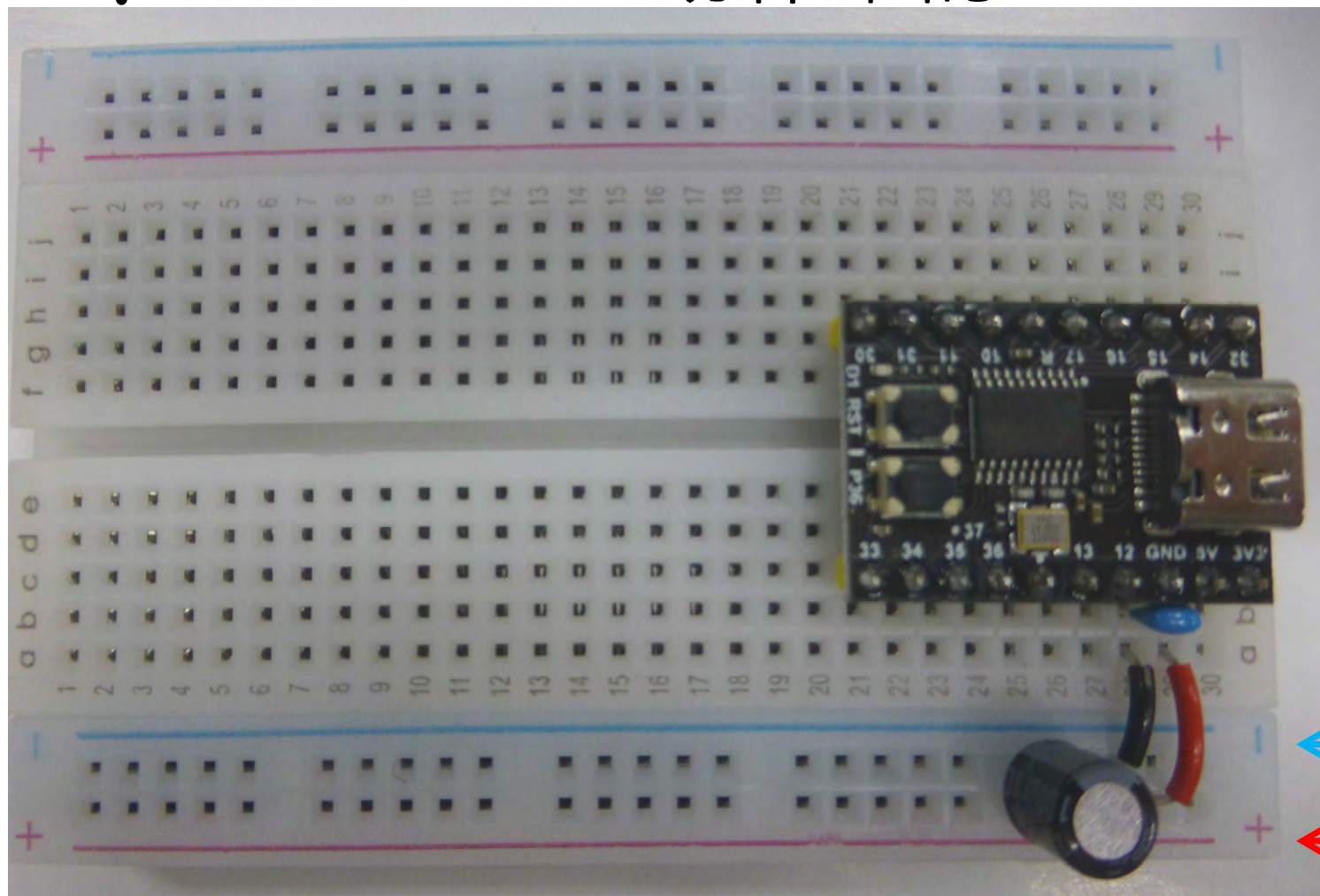
3.3-5V Power Supply

Support Arduino/SDCC

LED



1. マイコンの動作確認 Lチカ



接続

マイコンボード

コンデンサ

配線2本

赤：5V 黒：GND

← 青；GND (0V)

← 赤：5V

プログラムの開発環境

- マイコン

最終的には機械語と呼ばれる命令に従って動作します。

機械語：人間にはわかりにくい 1 0 0 1 . . .

人間にわかりやすい言語で記述 ➡ 機械語に変換

アセンブラ、C、C++、Pythonなど

今回の環境

ノートPC (Windows10)

開発環境ソフト：Arduino IDE

* 言語：Cに似たArduino言語



Arduino IDE

Blink | Arduino IDE 2.1.1

ファイル(F) 編集 スケッチ ツール ヘルプ

新規スケッチ Ctrl+N

新規クラウドスケッチ Alt+Ctrl+N

開く... Ctrl+O

最近使った項目を開く

スケッチブック

スケッチ例

閉じる Ctrl+W

保存 Ctrl+S

名前を付けて保存... Ctrl+Shift+S

基本設定... Ctrl+カンマ

詳細

終了 Ctrl+Q

付属のスケッチ例

01.Basics

02.Digital

03.Analog

04.Communication

05.Control

06.Sensors

07.Display

08.Strings

09.USB

10.StarterKit_BasicKit

11.ArduinoISP

CH552 Board用のスケッチ例

Ethernet

Firmata

Generic_Examples

Keyboard

LiquidCrystal

SD

Servo

SoftI2C

SPI

Stepper

TFT

Touch Key

WS2812

www.arduino.cc/en/Tutorial/Blink

D_BUILTIN 30

up function runs once when you press reset or power the board

() {

alize digital pin LED_BUILTIN as an output.

LED_BUILTIN, OUTPUT);

p function runs over and over again forever

() {

rite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)

00); // wait for a second

rite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW

00); // wait for a second

01.Basics

AnalogReadSerial

02.Digital

SerialMinimum

03.Analog

Blink

04.Communication

DigitalReadSerial

05.USB

Fade

06.EEPROM

ReadAnalogVoltage

Arduino Mbed OS

Edge Boards by Arduino

Boards included in this packa

Arduino Edge Control

詳細情報

4.0.4

インストール

Arduino Mbed OS

Giga Boards by Arduino

Boards included in this packa

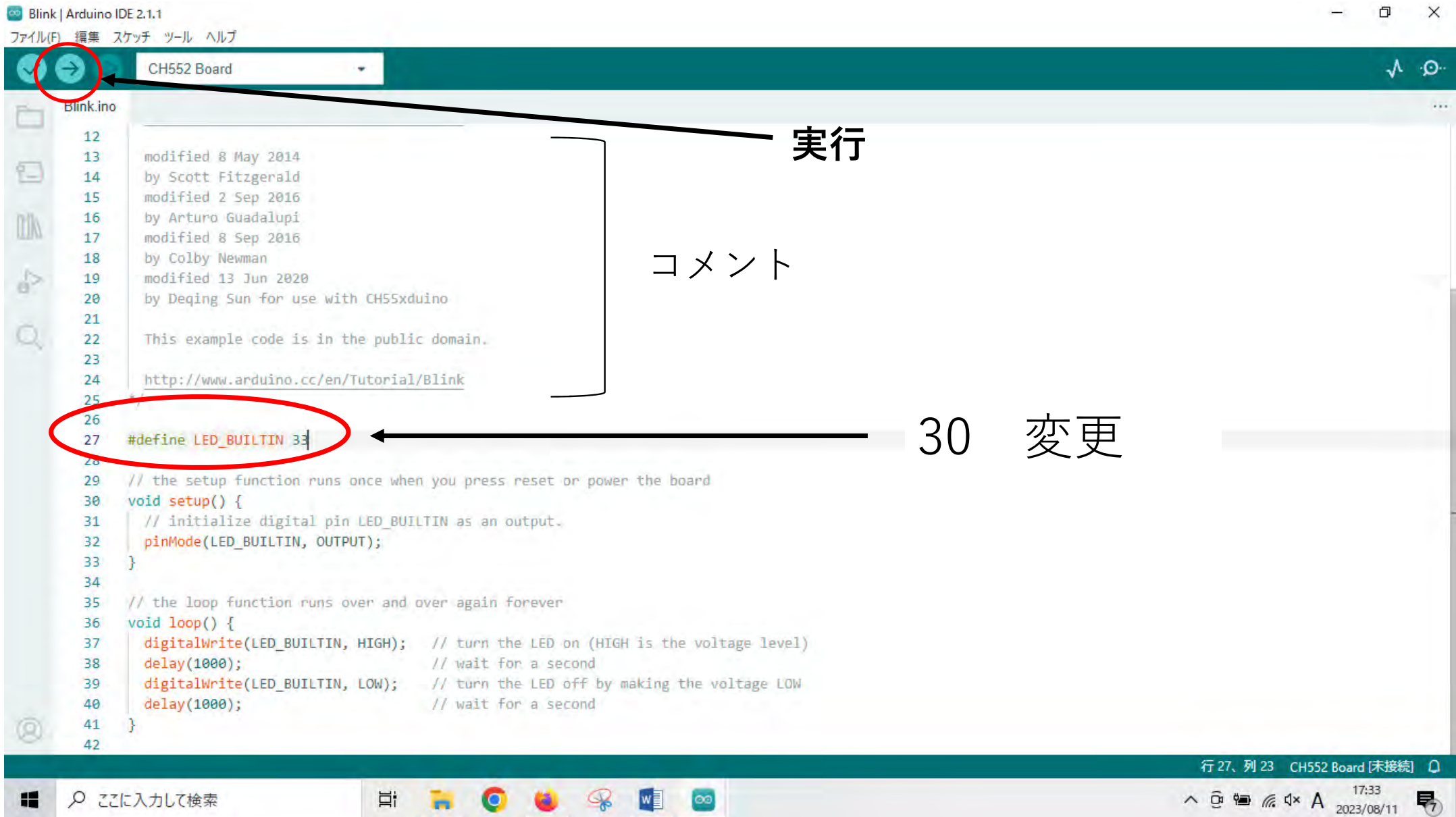
Arduino Giga

詳細情報

行 32、列 5 CH552 Board [未接続]

ここに入力して検索

17:32 2023/08/11



プログラムの約束事

- ・コメント

```
/*      */ で挟まれた間（複数行でも可）  
// この記号から後ろ（その行のみ）
```

- ・プログラムの構成

- * 初期設定

```
#define LED_BUILTIN 30                // 定義文
```

- * 最初1回実行

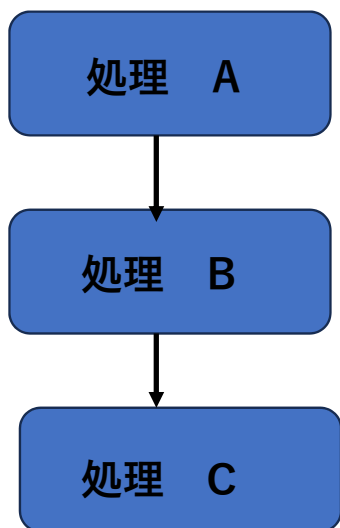
```
void setup() {  
    pinMode(LED_BUILTIN, OUTPUT);    // ボード端子の出力定義  
}
```

- * 繰り返し処理

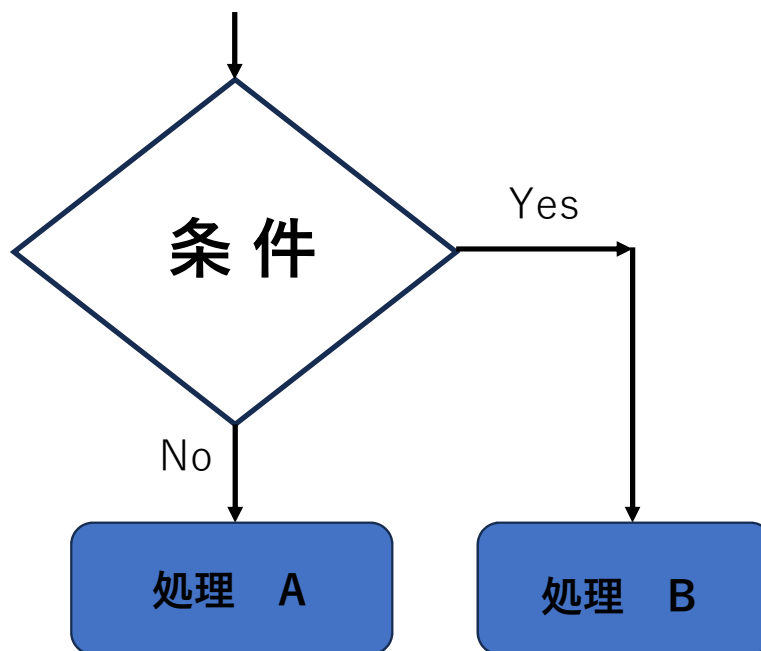
```
void loop() {  
    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
    delay(1000);                     // wait for a second  
    digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW  
    delay(1000);                     // wait for a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

プログラムの基本構成要素

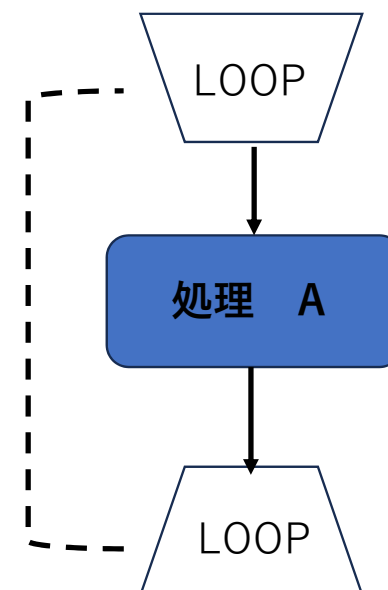
接続

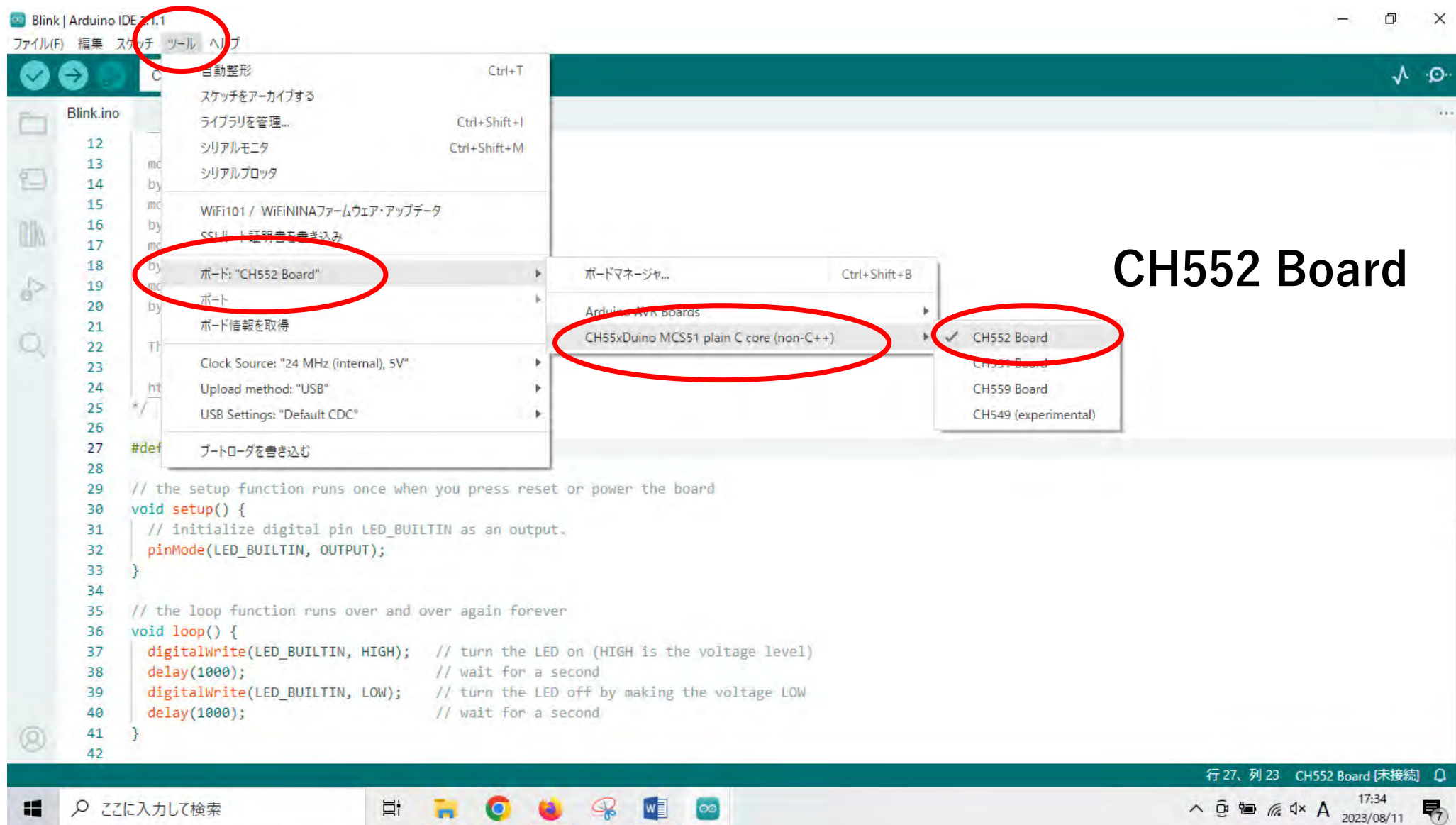


分岐



繰り返し





Blink | Arduino IDE 2.1.1

ファイル(F) 編集 スケッチ ツール ヘルプ

自動整形 Ctrl+T

スケッチをアーカイブする

ライブラリを管理... Ctrl+Shift+I

シリアルモニタ Ctrl+Shift+M

シリアルプロッタ

WiFi101 / WiFininaファームウェア・アップデート

SSLルート証明書を書き込み

ボード: "CH552 Board" ▶

ポート ▶

ポート情報を取得

COM1

Clock Source: "24 MHz (internal), 5V" ▶

Upload method: "USB" ▶

USB Settings: "Default CDC" ▶

ブートローダを書き込む

ツール ヘルプ

自動整形 Ctrl+T

スケッチをアーカイブする

ライブラリを管理... Ctrl+Shift+I

シリアルモニタ Ctrl+Shift+M

シリアルプロッタ

WiFi101 / WiFininaファームウェア・アップデート

SSLルート証明書を書き込み

ボード: "CH552 Board" ▶

ポート: "COM1" ▶

ポート情報を取得

ボードマネージャ... Ctrl+Shift+B

Arduino AVR Boards ▶

CH55xDuino MCS51 plain C core (non-C++) ▶

Clock Source: "24 MHz (internal), 5V" ▶

Upload method: "USB" ▶

USB Settings: "Default CDC" ▶

ブートローダを書き込む

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

行 27、列 23 CH552 Board [未接続]

ここに入力して検索

17:40 2023/08/11

Blink | Arduino IDE 2.1.1

ファイル 編集 スケッチ ツール ヘルプ

CH552 Board

Blink.ino

```
21
22 This example code is in the public domain.
23
24 http://www.arduino.cc/en/Tutorial/Blink
25 */
26
27 #define LED_BUILTIN 30
28
29 // the setup function runs once when you press reset or power the board
30 void setup() {
31   // initialize digital pin LED_BUILTIN as an output.
32   pinMode(LED_BUILTIN, OUTPUT);
33 }
34
35 // the loop function runs over and over again forever
36 void loop() {
37   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
38   delay(1000); // wait for a second
39   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
40   delay(1000); // wait for a second
41 }
42
```

書き込み失敗

出力

```
No CH55x USB Found, retry...
No CH55x USB Found, retry...
No CH55x USB Found, retry...
No CH55x USB Found, retry...
Found no CH55x USB
Time limit reached, exit process
Failed uploading: uploading error: exit status 1
```

Upload error: Failed uploading: uploading error: exit status 1

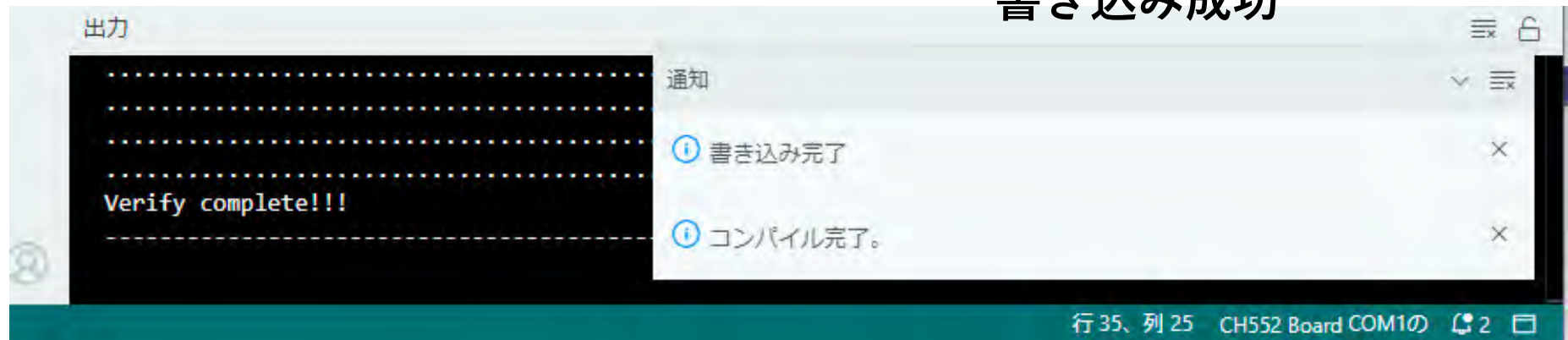
エラーメッセージをコピー

行 27、列 23 CH552 Board COM1の 2

ここに入力して検索

17:42 2023/08/11

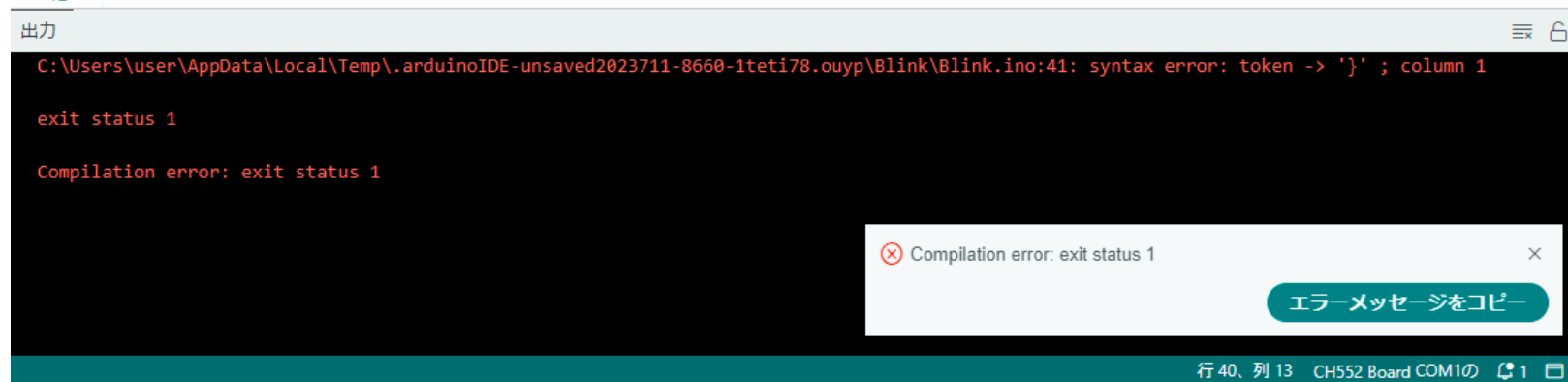
書き込み成功



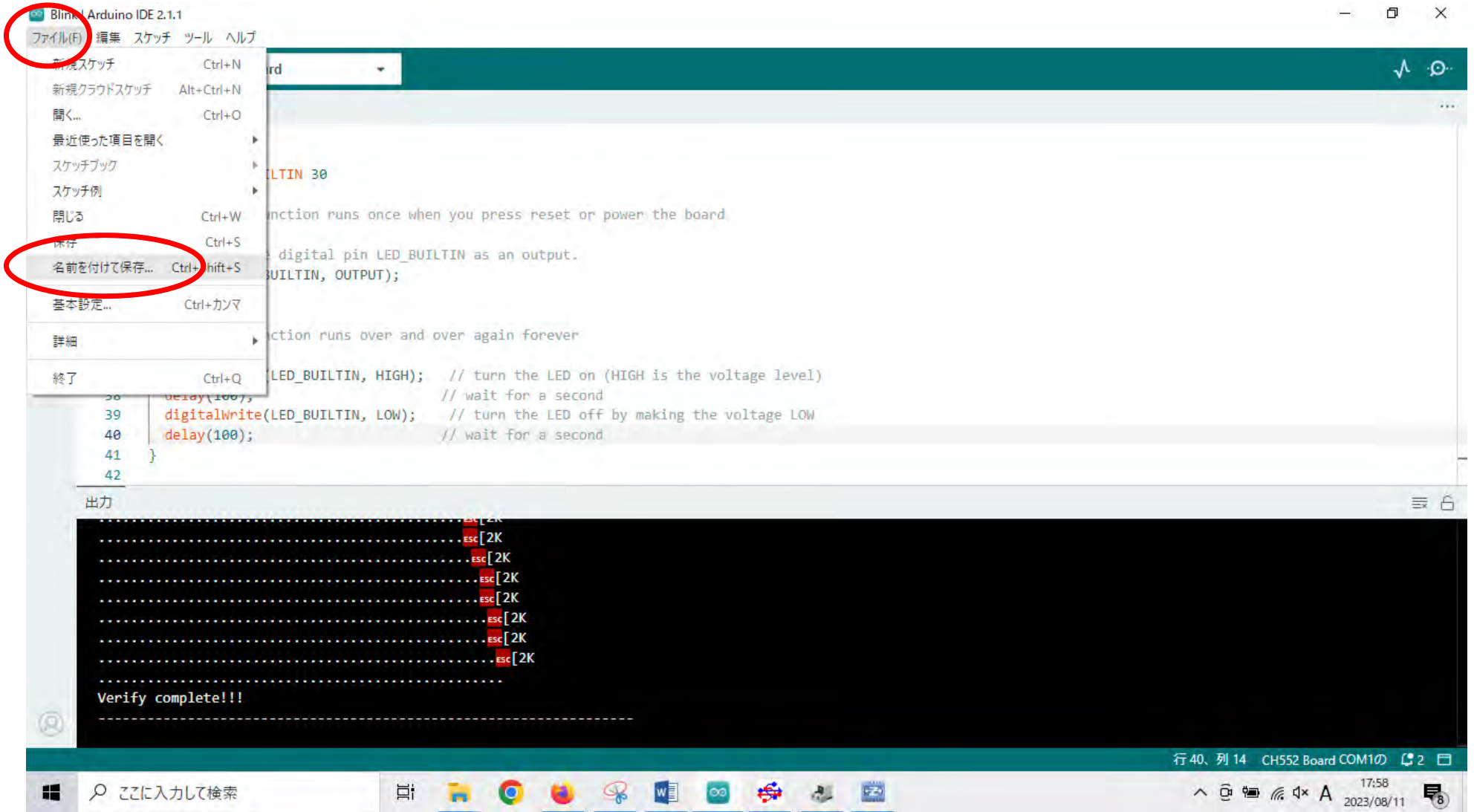
```
35 // the loop function runs over and over again forever
36 void loop() {
37   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
38   delay(100); // wait for a second
39   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
40   delay(100); // wait for a second
41 }
42
```

コンパイルエラー

* プログラムの誤り



プログラムの保存



🔍 スケッチフォルダを別名で保存...



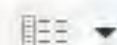
📁 > PC > Windows (C:) > work



🔍 workの検索

整理 ▾

新しいフォルダー



📁 Program Files ^

名前

更新日時

種類

📁 Program Files (

📁 RENRI

📁 test

📁 texlive

📁 TOOL(KCCT)

📁 usr

📁 WCH

📁 Windows

📁 work

📁 work フォルダ

検索条件に一致する項目はありません。

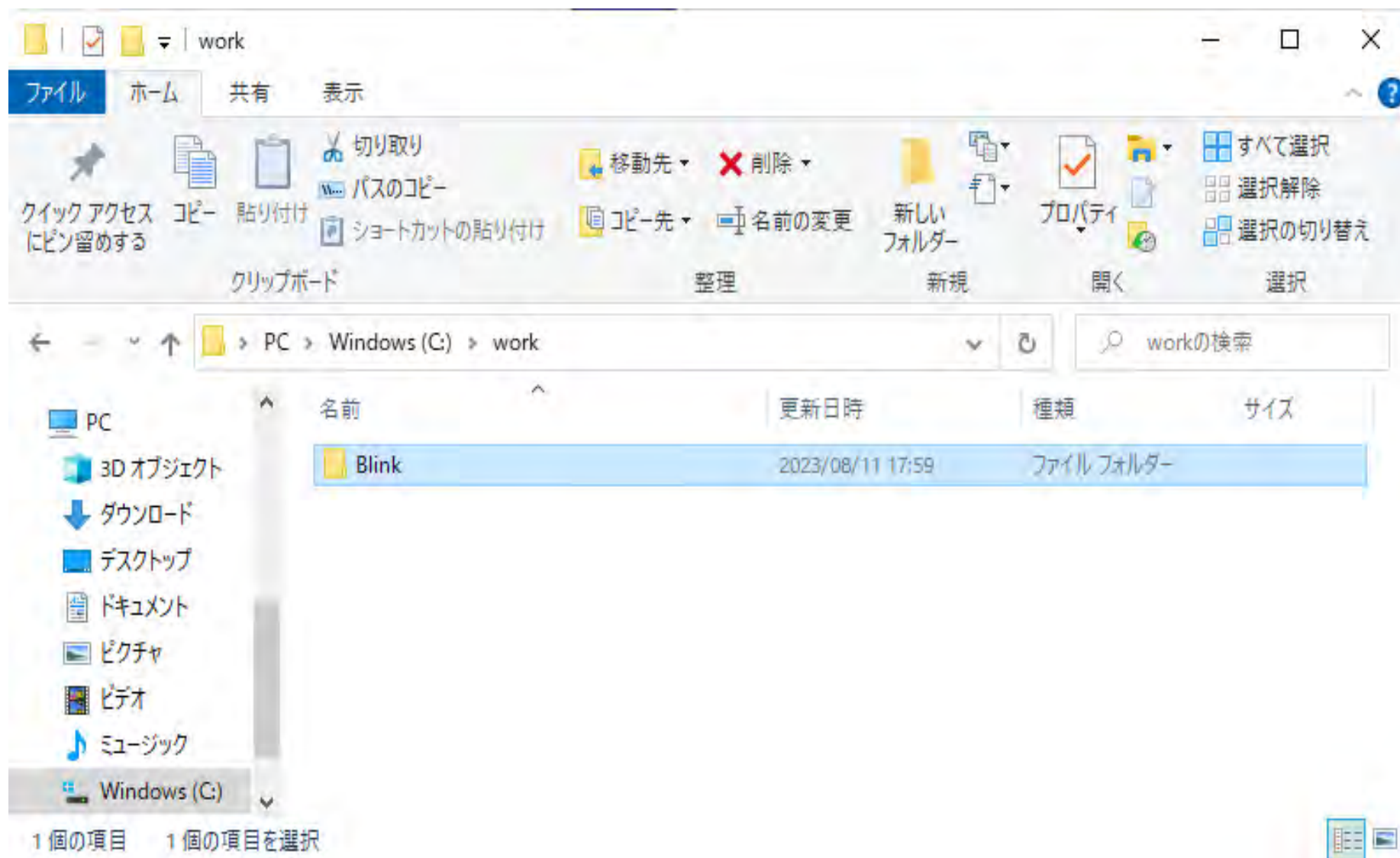
ファイル名(N): Blink

ファイルの種類(T): All Files (*.*)

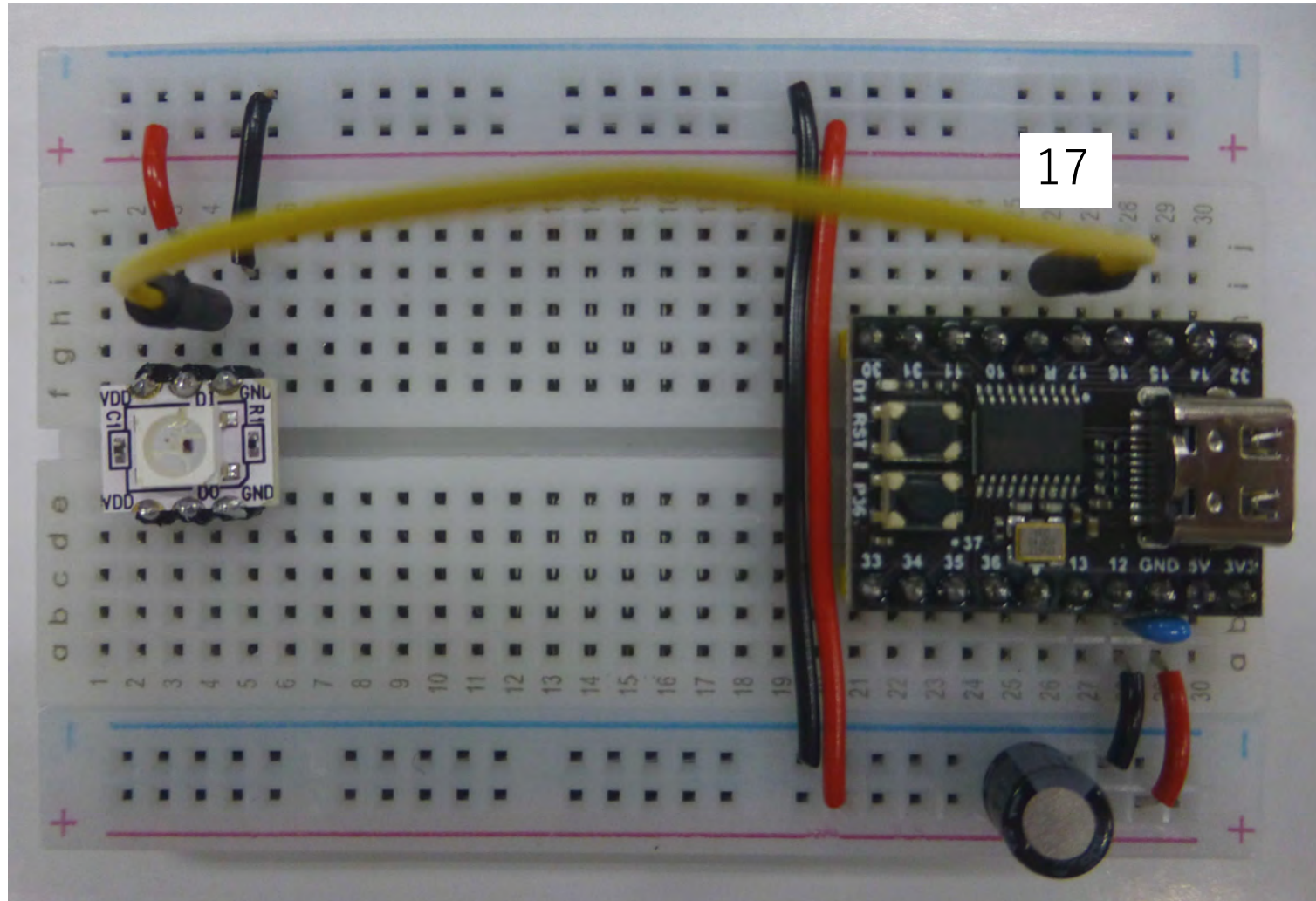
^ フォルダの非表示

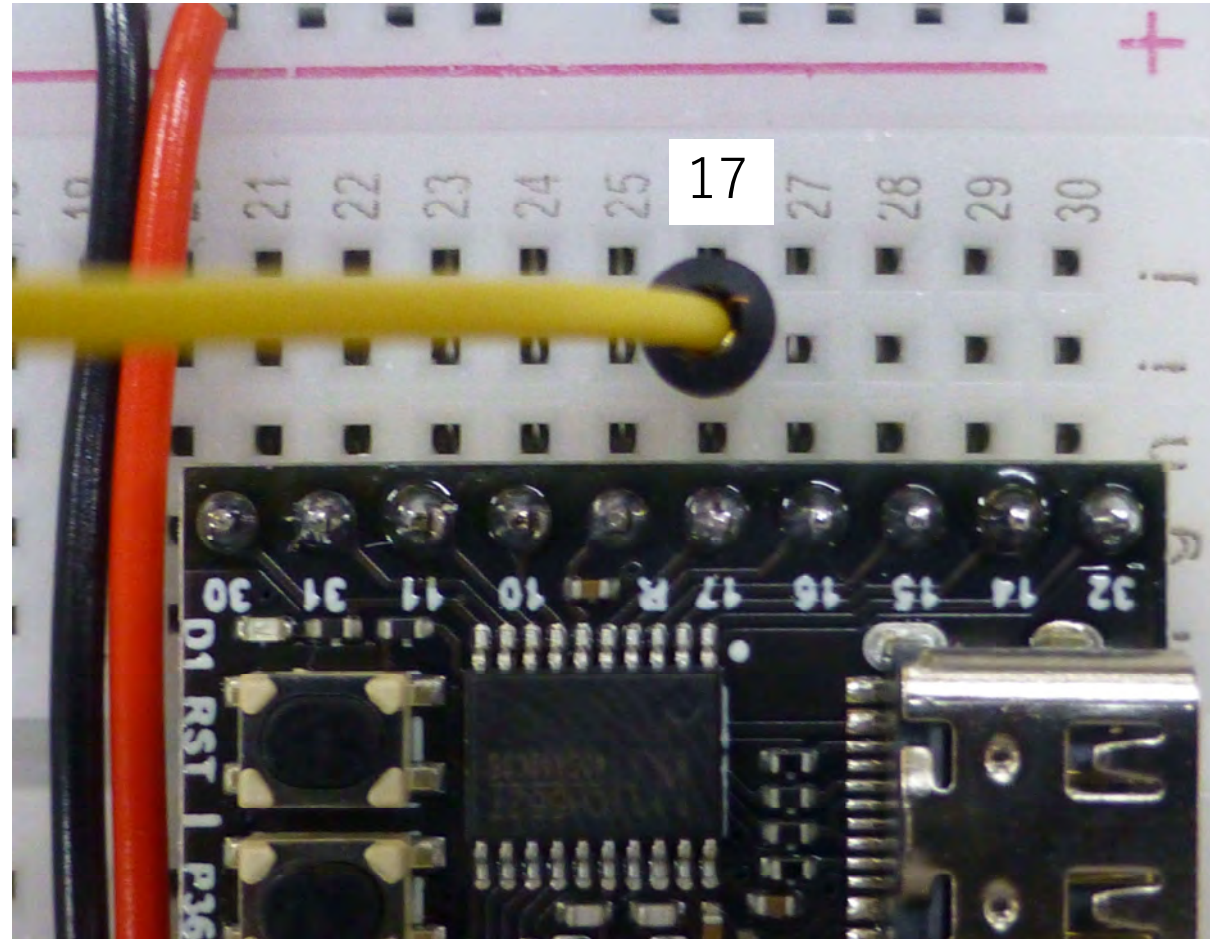
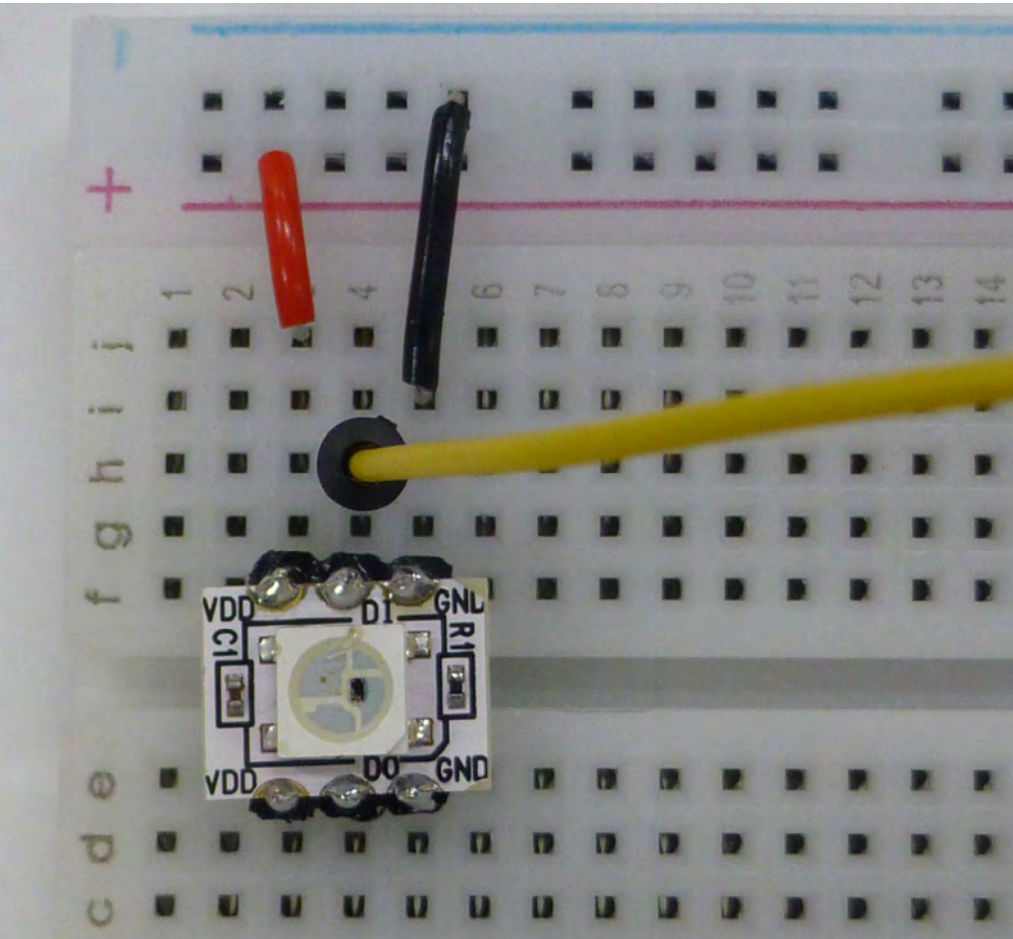
保存(S)

キャンセル



2. カラーLEDの接続と点灯動作





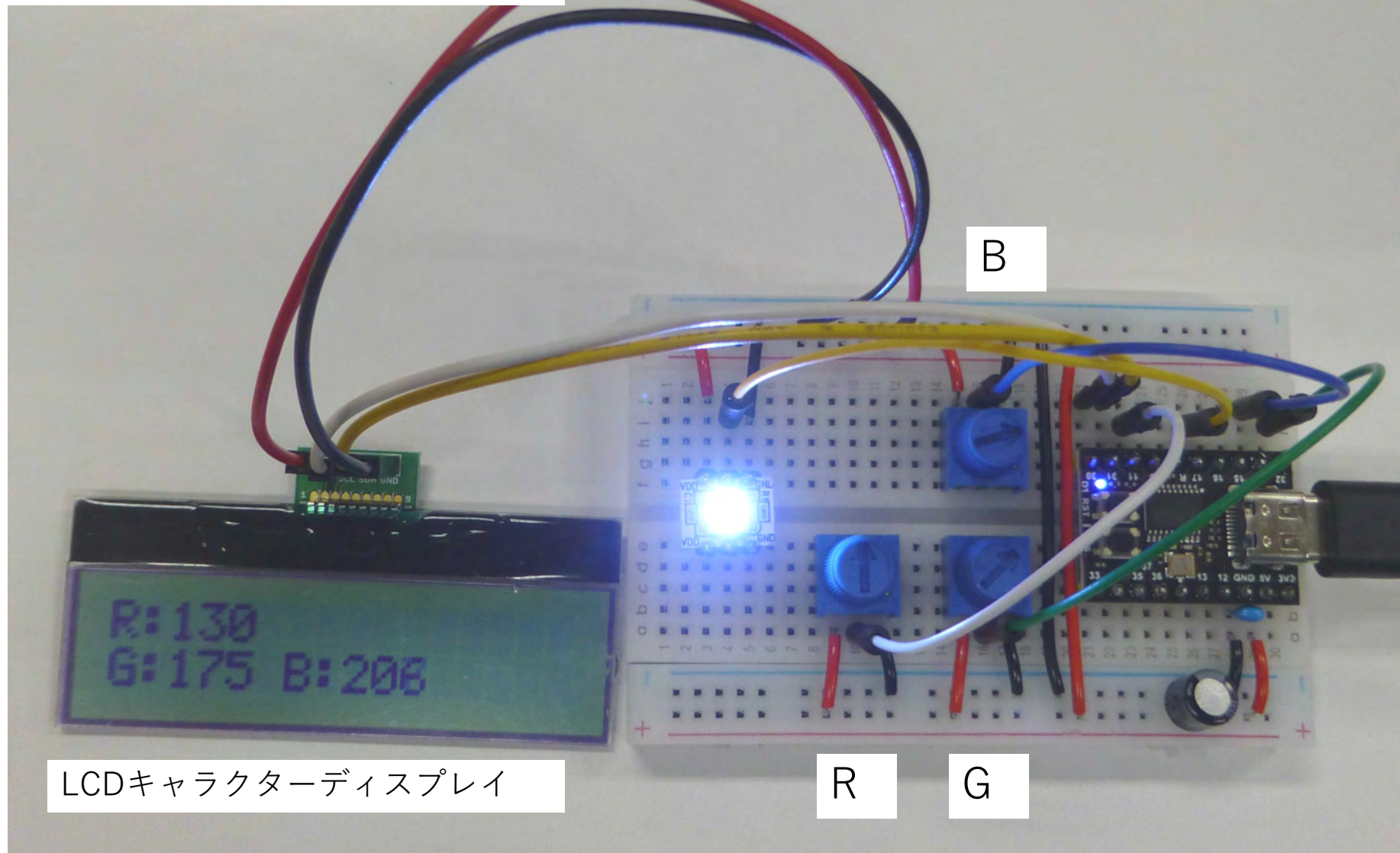
プログラム名：LED17_signal.ino

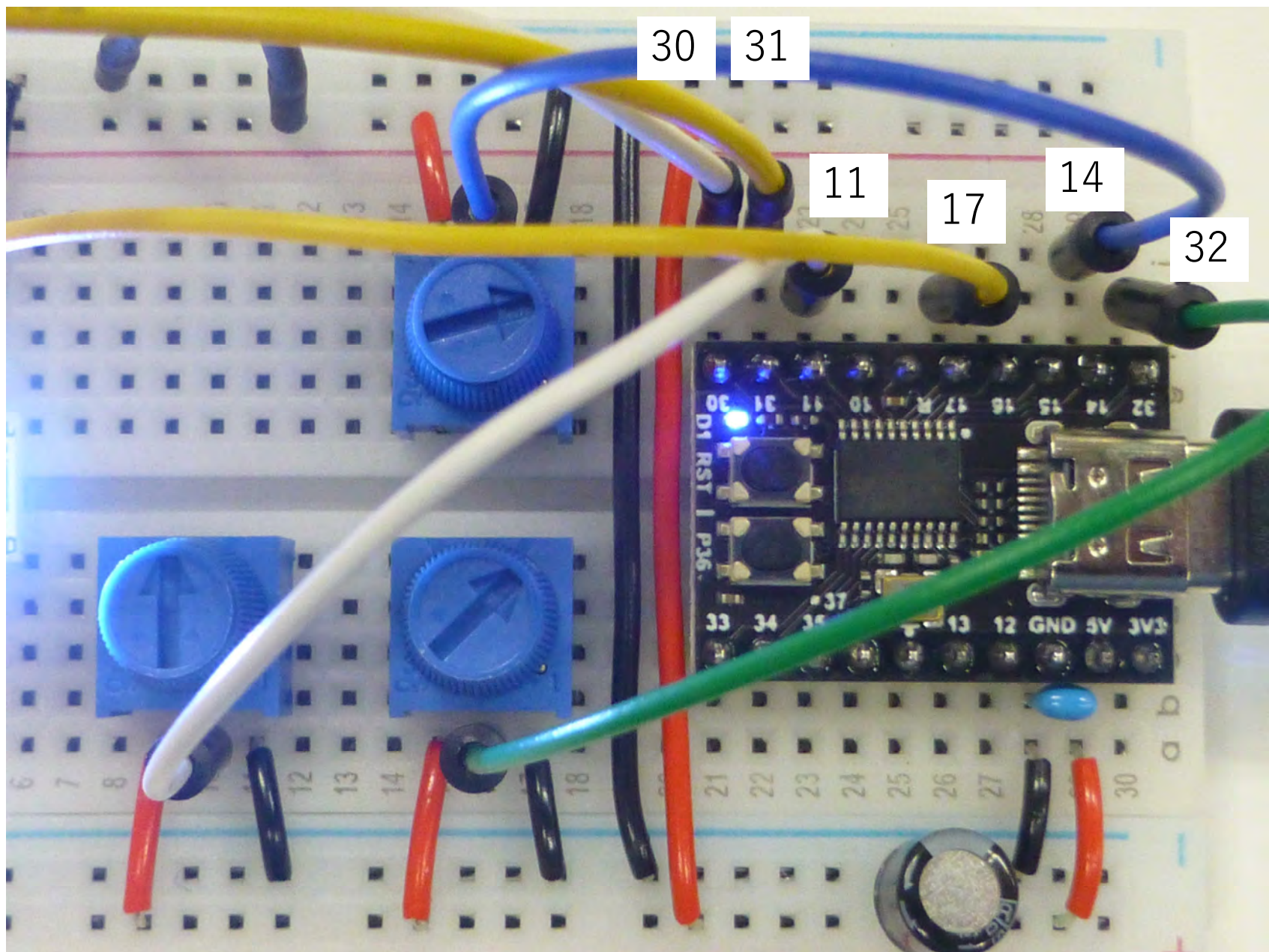
```
#include <WS2812.h>           // カラーLEDを使用するためのヘッダーファイル
. . .
__xdata uint8_t ledData[NUM_BYTES];    // LED表示データを入れる変数

void setup() {
    pinMode(17, OUTPUT);          //LEDの接続ピン：17ピン
}

void loop() {
    set_pixel_for_GRB_LED(ledData, 0, 0, 0, 255);    // 青色の設定
    neopixel_show_P1_7(ledData, NUM_BYTES);          // 表示
    delay(1000);
    set_pixel_for_GRB_LED(ledData, 0, 255, 255, 0);  // 黄色の設定
    neopixel_show_P1_7(ledData, NUM_BYTES);          // 表示
    delay(1000);
    set_pixel_for_GRB_LED(ledData, 0, 255, 0, 0);    // 赤色の設定
    neopixel_show_P1_7(ledData, NUM_BYTES);          // 表示
    delay(1000);
}
```


可変抵抗器でRGB制御





プログラム名：LED17_ADC3_LCD (LCD処理部を除く)

```
#include <WS2812.h>
#include <SoftI2C.h>
```

```
int sensorPinR = 11; // select the input pin for the potentiometer.
```

```
...
```

```
int sensorValueB = 0; // variable to store the value coming from the sensor
```

```
void setup() {
```

```
    pinMode(17, OUTPUT);           // Possible to use other pins.
```

```
    pinMode(sensorPinR, INPUT);    // R用の可変抵抗器
```

```
    pinMode(sensorPinG, INPUT);    // G用の可変抵抗器
```

```
    pinMode(sensorPinB, INPUT);    // B用の可変抵抗器
```

```
}
```

```
void loop() {
```

```
    sensorValueR = analogRead(sensorPinR);    delay(10);        // Rデータ読み込み
```

```
    sensorValueG = analogRead(sensorPinG);    delay(10);        // Gデータ読み込み
```

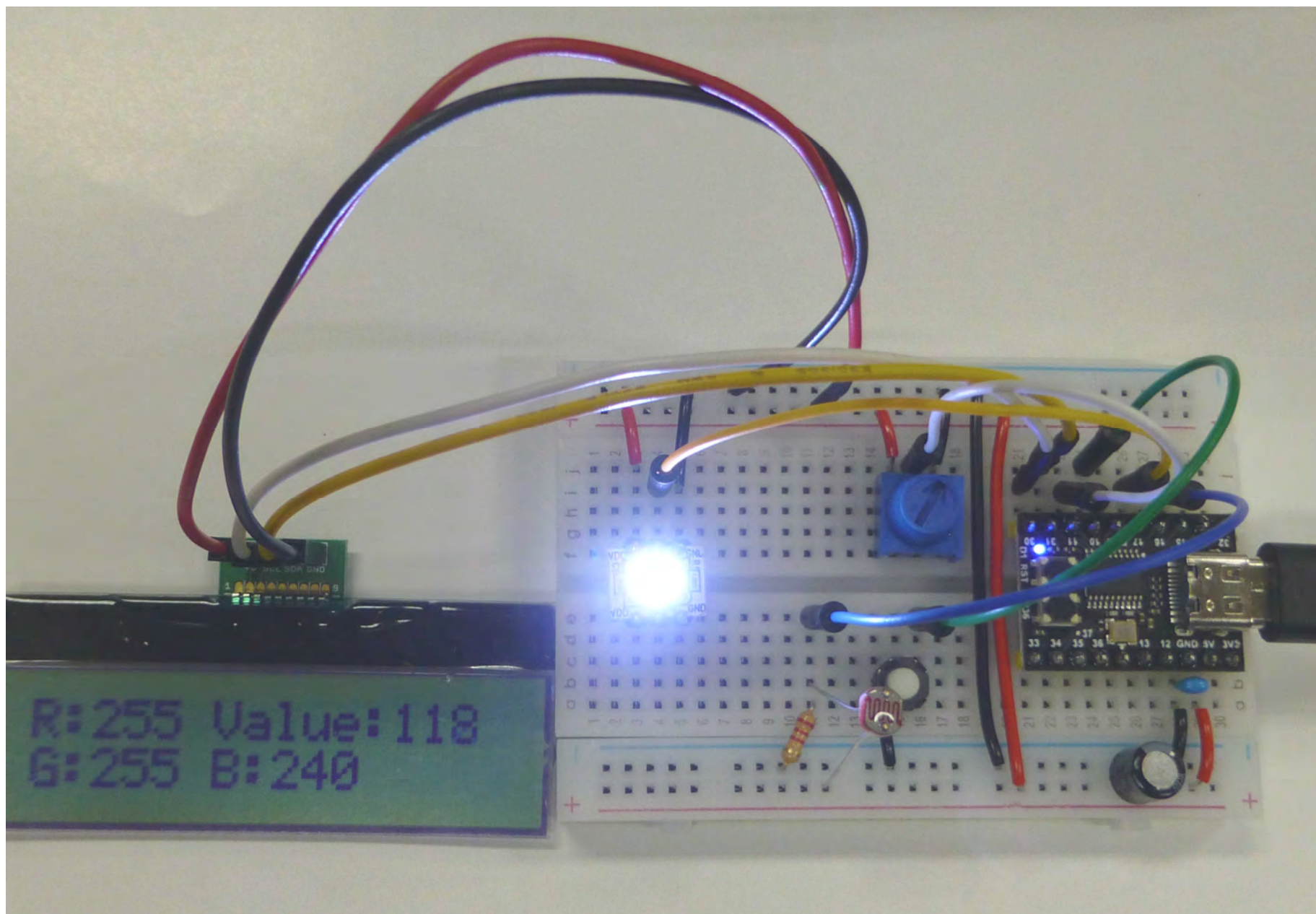
```
    sensorValueB = analogRead(sensorPinB);    delay(10);        // Bデータ読み込み
```

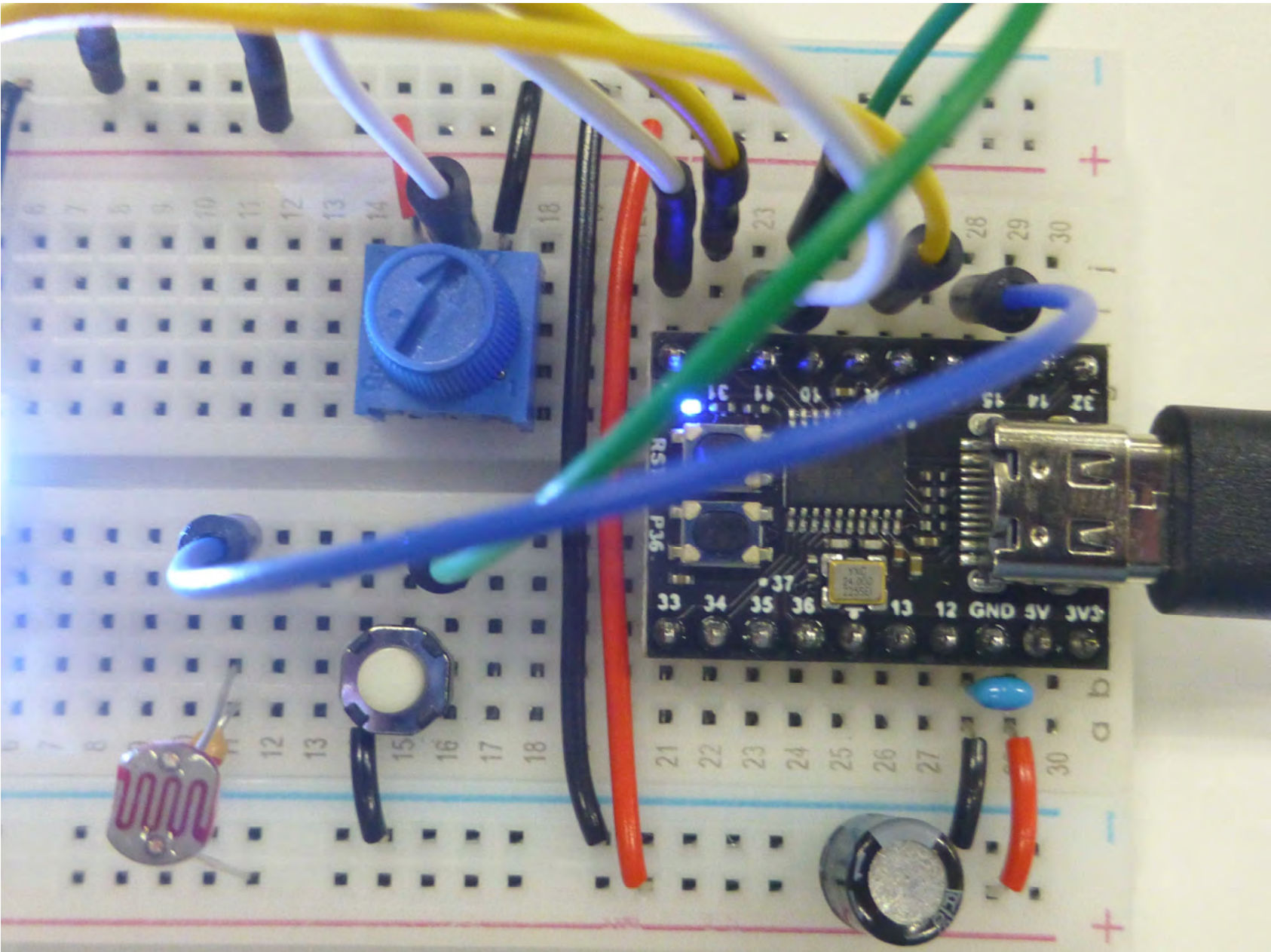
```
    set_pixel_for_GRB_LED(ledData, 0, sensorValueR, sensorValueG, sensorValueB);
```

```
    neopixel_show_P1_7(ledData, NUM_BYTES);  // LED表示
```

```
    delay(100);
```

```
}
```

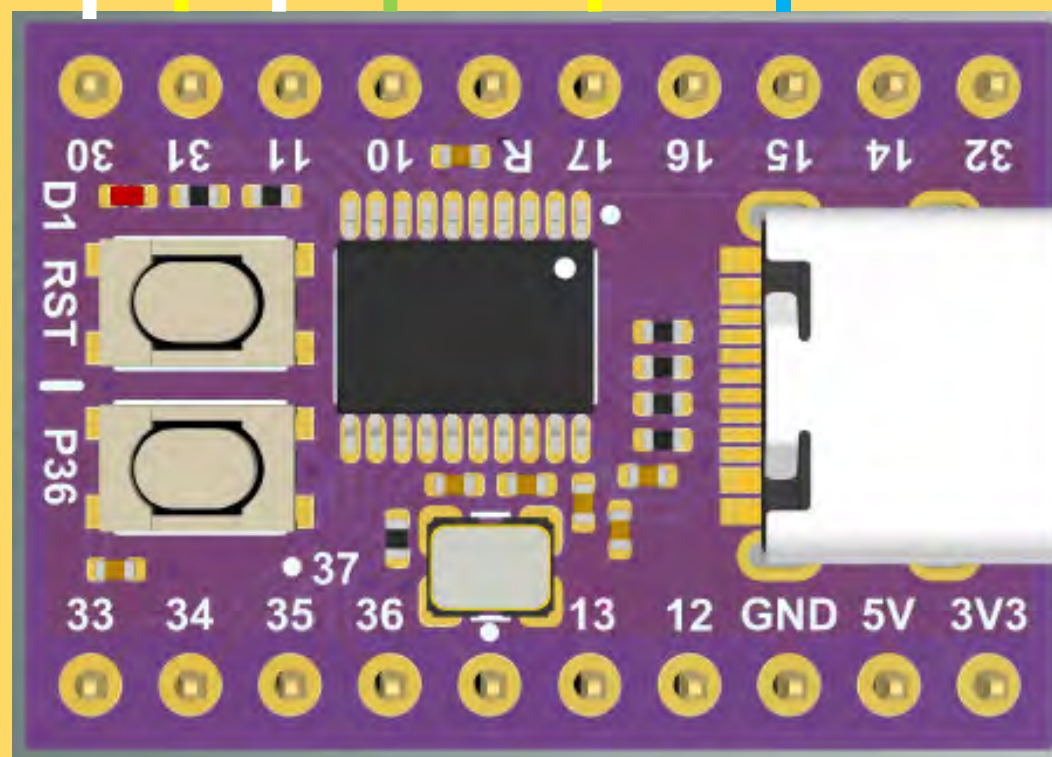


結線図

キャラクタ
ディスプレイ

LED
SW
VR

CdS



プログラム名：LED17_VR11_CDS15_LCD (LCD処理部を除く)

```
#include <WS2812.h>
```

```
#include <SoftI2C.h>
```

```
int flag = 0;          // Switch
```

```
int buttonState = 0;
```

```
uint8_t RGBvalueR[]={0,0,0,185,255,255,250,255,255,180};
```

```
// 0～9番地に初期値設定
```

```
uint8_t RGBvalueG[]={32,65,150,235,255,255,245,153,40,0};
```

```
uint8_t RGBvalueB[]={128,255,255,255,240,150,0,0,0,104};
```

```
uint8_t dispValue;
```

```
int j,mode;
```

```
void setup() {
```

```
    pinMode(17, OUTPUT);          //LED DIN.
```

```
    pinMode(sensorPinVR, INPUT);  // VR
```

```
    pinMode(sensorPinCDS, INPUT); // CdS
```

```
    pinMode(SW, INPUT_PULLUP);    // Switch
```

```
    mode = 0;                     // 0:VR 1:Cds}
```

```
}
```

```

void loop() {
    buttonState = digitalRead(SW);
    if (buttonState == LOW) {                                // SWでVRかCdSか選択
        if (mode == 0) mode = 1;    // CdS
        else      mode = 0;    // VR
        delay(300);
    }
    sensorValueVR = analogRead(sensorPinVR); delay(10);      // VRの値取り込み
    sensorValueCDS = analogRead(sensorPinCDS); delay(10);    // CdSの値取り込み
    if (mode == 0) dispValue = sensorValueVR;                // 使う値をdispValueへ代入
    else      dispValue = sensorValueCDS;
    if(dispValue < 30 )      j=0;                                // 表示レベルの判別 10段階
    else if(dispValue < 58 ) j=1;
    else if(dispValue < 86 ) j=2;
    else if(dispValue < 104 ) j=3;
    else if(dispValue < 132 ) j=4;
    else if(dispValue < 160 ) j=5;
    else if(dispValue < 188 ) j=6;
    else if(dispValue < 216 ) j=7;
    else if(dispValue < 234 ) j=8;
    else      j=9;
    set_pixel_for_GRB_LED(ledData, 0, RGBvalueR[j], RGBvalueG[j], RGBvalueB[j]); //色データの設定
    neopixel_show_P1_7(ledData, NUM_BYTES);    //表示
    delay(100);
}

```

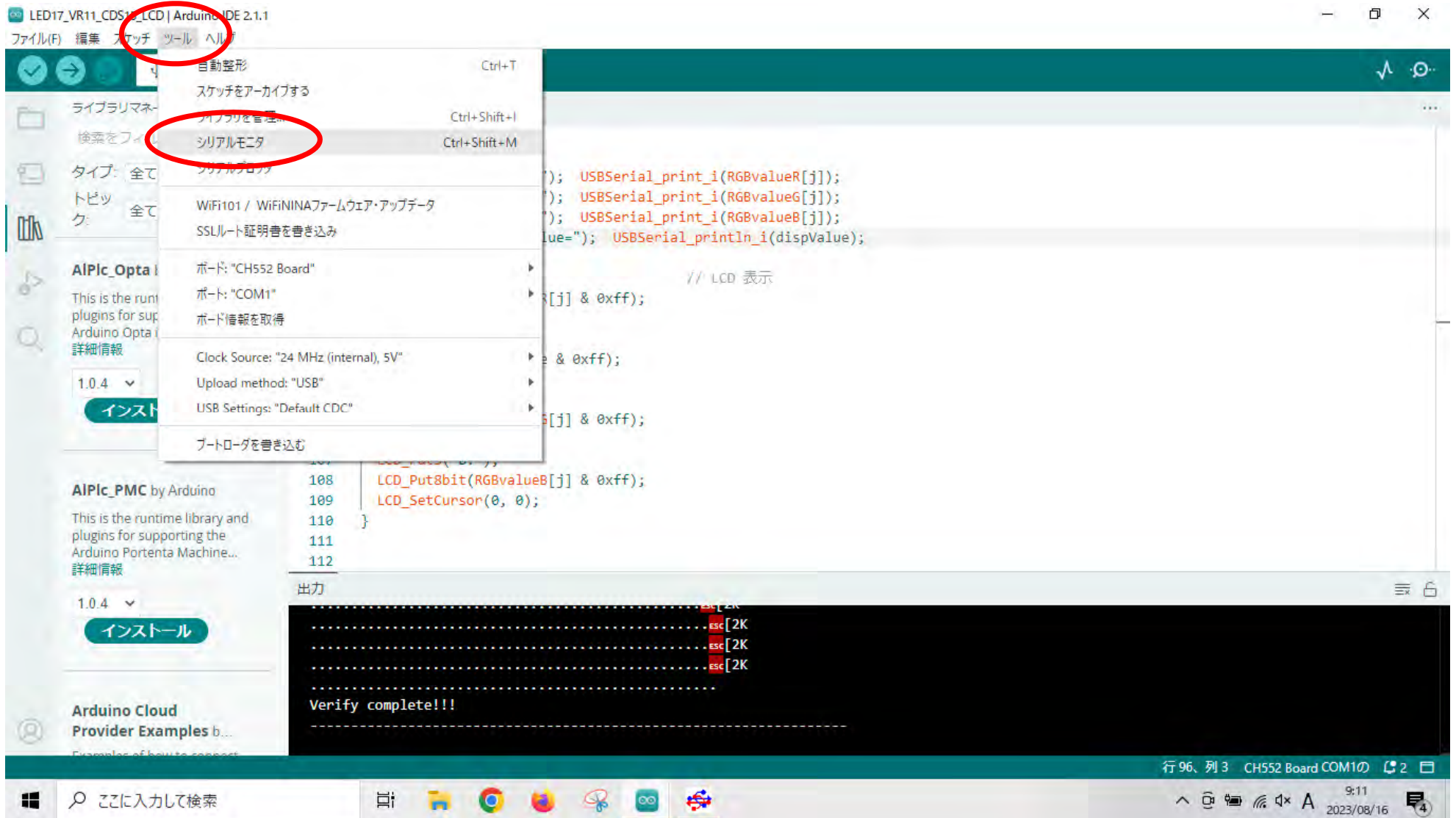

シリアルモニタの利用

マイコンの中のデータはLCDキャラクタディスプレイで表示させているが、これがないときには、Arduino IDE にあるシリアルモニタを使うとよい。

プログラム中に表示させたいところに以下のような命令を追加

```
USBSerial_print(" R="); USBSerial_print_i(RGBvalueR[j]);  
USBSerial_print(" G="); USBSerial_print_i(RGBvalueG[j]);  
USBSerial_print(" B="); USBSerial_print_i(RGBvalueB[j]);  
USBSerial_print(" Value="); USBSerial_println_i(disValue);
```

最初の USBSerial_print(); は文字列の表示
次の USBSerial_print_i(); は値の整数表示
最後の USBSerial_println_i(); は表示後改行
よって、4つの値が1行で表示され、改行



CH552 Board

LED17_VR11_CDS15_LCD.ino

```
88
89   set_pixel_for_GRB_LED(ledData, 0, RGBvalueR[j], RGBvalueG[j], RGBvalueB[j]); //色データの設定
90   neopixel_show_P1_7(ledData, NUM_BYTES);      //表示
91   delay(100);
92
93   USBSerial_print(" R="); USBSerial_print_i(RGBvalueR[j]);
94   USBSerial_print(" G="); USBSerial_print_i(RGBvalueG[j]);
95   USBSerial_print(" B="); USBSerial_print_i(RGBvalueB[j]);
96   USBSerial_print(" Value="); USBSerial_println_i(disValue);
97
98   LCD_Puts("R:");                      // LCD 表示
99   LCD_Put8bit(RGBvalueR[j] & 0xff);
100  LCD_Put8(0x20);
```

出力 シリアルモニタ x

メッセージ ('COM1'のCH552 Boardにメッセージを送信するにはEnter)

LFのみ

9600 baud

```
R=255 G=255 B=240 Value=121
R=255 G=255 B=240 Value=128
R=255 G=255 B=240 Value=120
R=255 G=255 B=240 Value=116
R=255 G=255 B=240 Value=124
```

補足事項

開発環境の構築

「CH55x系ボード WeActStudio CH552CoreBoardをArduinoで使う」

<https://qiita.com/usashirou/items/7513c13dd66c149cca53>

書籍：「CH55 x であろうでしょう」 秋田純一著 株式会社ラトルズ （本体:1600円＋税）

＊ ボードマネージャは最新ではなく、0.0.17 のバージョンを使用してください。

0.0.18 では、カラーLEDのプログラムでエラーが出ます。

CH552ボード追加

URL： https://raw.githubusercontent.com/DeqingSun/ch55xduino/ch55xduino/package_ch55xduino_mcs51_index.json

＊ 開発環境の構築方法については、<http://www.kobe-kosen.ac.jp/~kasai/> に掲載予定(8/20以降)

カラーLEDの使い方

「赤・緑・青のフルカラーLED WS2812Bの使い方」

https://www.marutsu.co.jp/pc/static/large_order/WS2812B_0124

RGBカラーコードチャート

https://www.rapidtables.org/ja/web/color/RGB_Color.html

気象庁ホームページにおける気象情報の配色に関する設定指針

https://www.jma.go.jp/jma/kishou/info/colorguide/120524_hpcolorguide.pdf