

ロボット博士養成講座

ロボティクスプロフェッサーコース

不思議アイテムI-2①

第2回

色で遊ぶ

講師用

目 次

0. 色で遊ぶ

- 0.0. 「色で遊ぶ」でやること
- 0.1. 必要なもの
- 0.2. カラーセンサー使用時の注意点

1. パーツの組み立てと接続

- 1.0. カラーセンサーの組み立てと接続
- 1.1. スピーカーと電池ボックスの接続
- 1.2. 無線受信モジュールの接続とペアリング

2. 色について知る

- 2.0. 3原色を混ぜる
- 2.1. 光の強さを変える
- 2.2. 色を変化させる
- 2.3. 色の表現方式

3. カラーセンサーで色を読み取る

- 3.0. 周囲の色を測定する
- 3.1. 白黒を判別する

4. ワンドットカメラ

- 4.0. 色を撮影する
- 4.1. 撮影した色を混ぜる

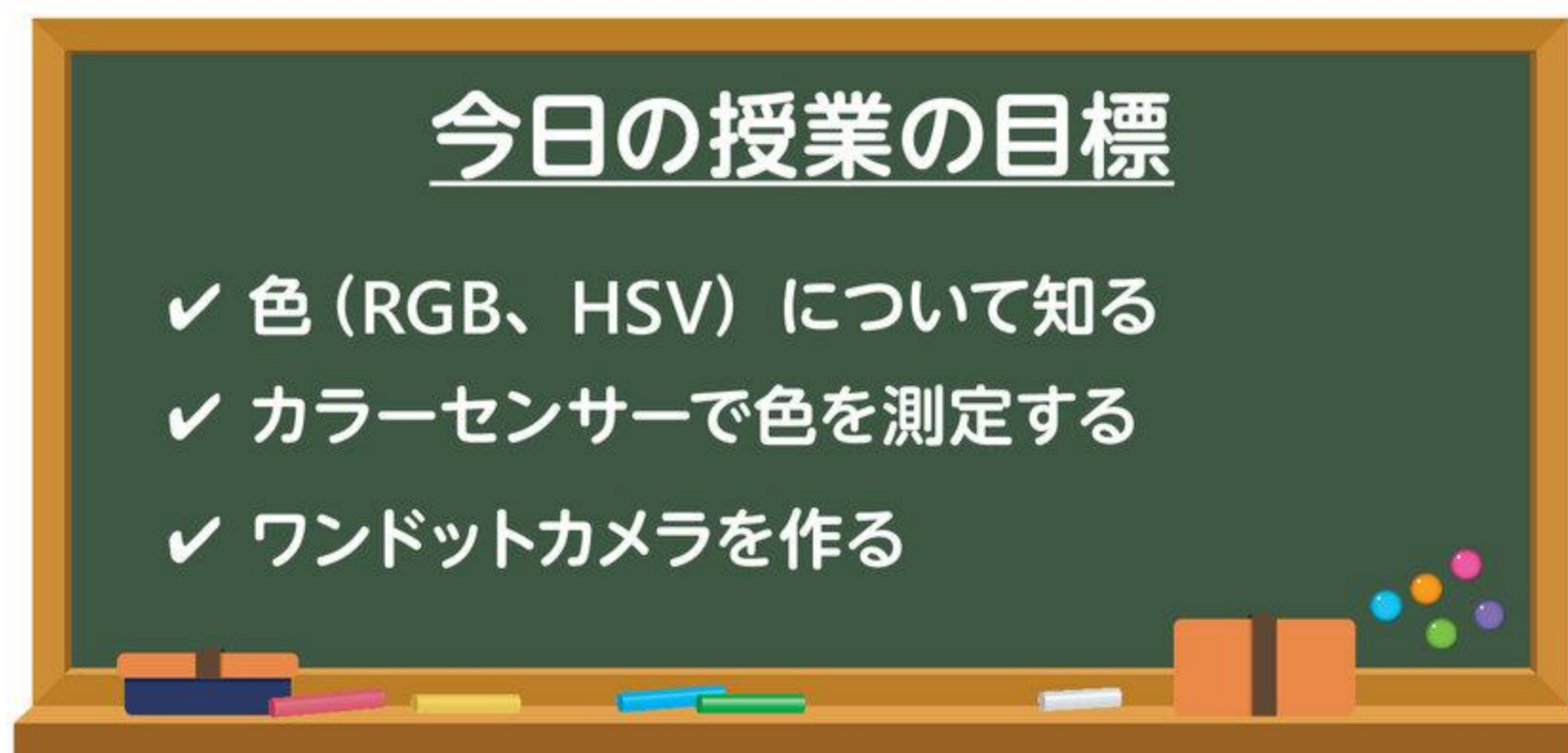
5. まとめ

- 授業開始にあたって
授業のはじめは、着席させ、大きな声であいさつしてから始めます。
- 今回の目標をパネルで用意するか、黒板に予め書いておきます。
(授業の目標を明確化することは大変重要なことですので、生徒によく理解させます)

目安時間は授業時間 120 分のうち、休憩 10 分程度を取ることを想定しています。
生徒の進捗状況により、休憩時間などを調整して授業を行ってください。

0. 色で遊ぶ (目安 10 分)

0.0. 「色で遊ぶ」 でやること



ここでは、色について勉強していきます！

みなさんは、図工や美術などの授業で、絵の具を混ぜてみたことはありませんか？ 絵の具を混ぜると、さまざまな色に変化しますね。実は、コンピューターでも同じで、混ぜることによってさまざまな色を表現します。今回は、そんなコンピューターの世界における色について学びます。「RGB」や「HSV」といった専門用語も出てきますが、後ほど詳しく説明します。実際にさまざまな色をLEDに表示させ、コンピューターでの色のあつかい方を学んでいきましょう。また、カラーセンサーを使って、一点だけ撮影できるワンドットカメラというものを作ります。いろいろな場所を撮影してみて、その場所の色がどのようにカラーセンサーによって読み取られているかを試してみましょ^う。どれも、コントローラーを使った操作で、簡単に色をあつかうことができるようになっています。

ではコンピューターの色の世界に飛び込んでいきましょう！

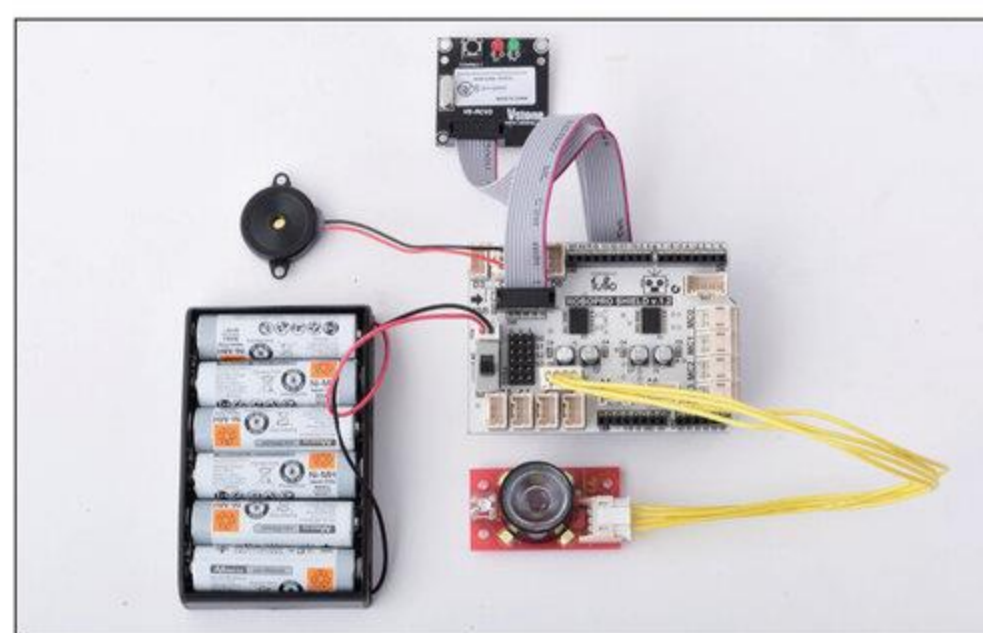
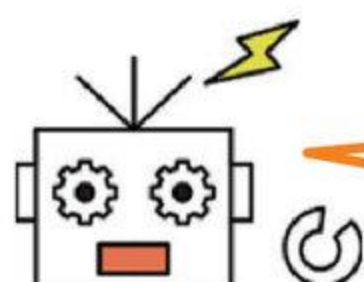


図0-0 接続完成図



ボクたちロボットの見ている色の世界と、
ヒトが見ている色の世界ってどう違うのかな？

0.1. 必要なもの

以下のパーツを準備しておきましょう。なお、前回使用したユニットから、マトリクスLEDシールドとマトリクスLEDを外したユニット（マイコンボードとロボプロシールドを積み重ねたもの）を今回も使います。

ラジオペンチ	1	ドライバー	1	USBケーブル	1	マイコンボード	1
							
ロボプロシールド	1	電池ボックス	1	リボンケーブル	1	コントローラー	1
							
無線受信モジュール	1	スピーカー	1	カラーセンサー	1	センサーケーブル	1
							

図0-1 必要なもの

0.2. カラーセンサー使用時の注意点

カラーセンサーのLEDは光が強いため、長い時間直視すると目にダメージを与えてしまいます。光の状態を確認するときは、紙で透かすか紙に光を投影するなどして、絶対に直視しないようにしましょう。

⚠️ 注意! LEDは直視しないこと!

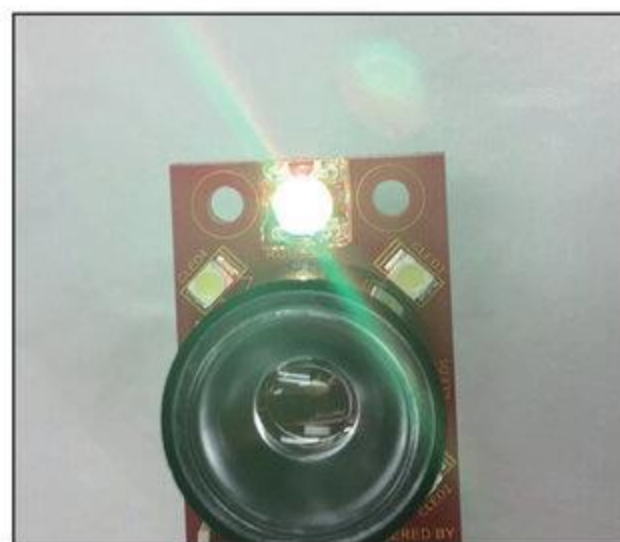


図0-2 カラーセンサー使用時の注意

1. パーツの組み立てと接続 (目安 15分)

1.0. カラーセンサーの組み立てと接続

<組み立て手順①>

まずは、カラーセンサーを組み立てます。カラーセンサーのレンズなどと一緒に入っているM2L6タッピングネジ(×2)を使って、レンズをカラーセンサーの基板に取り付けます。

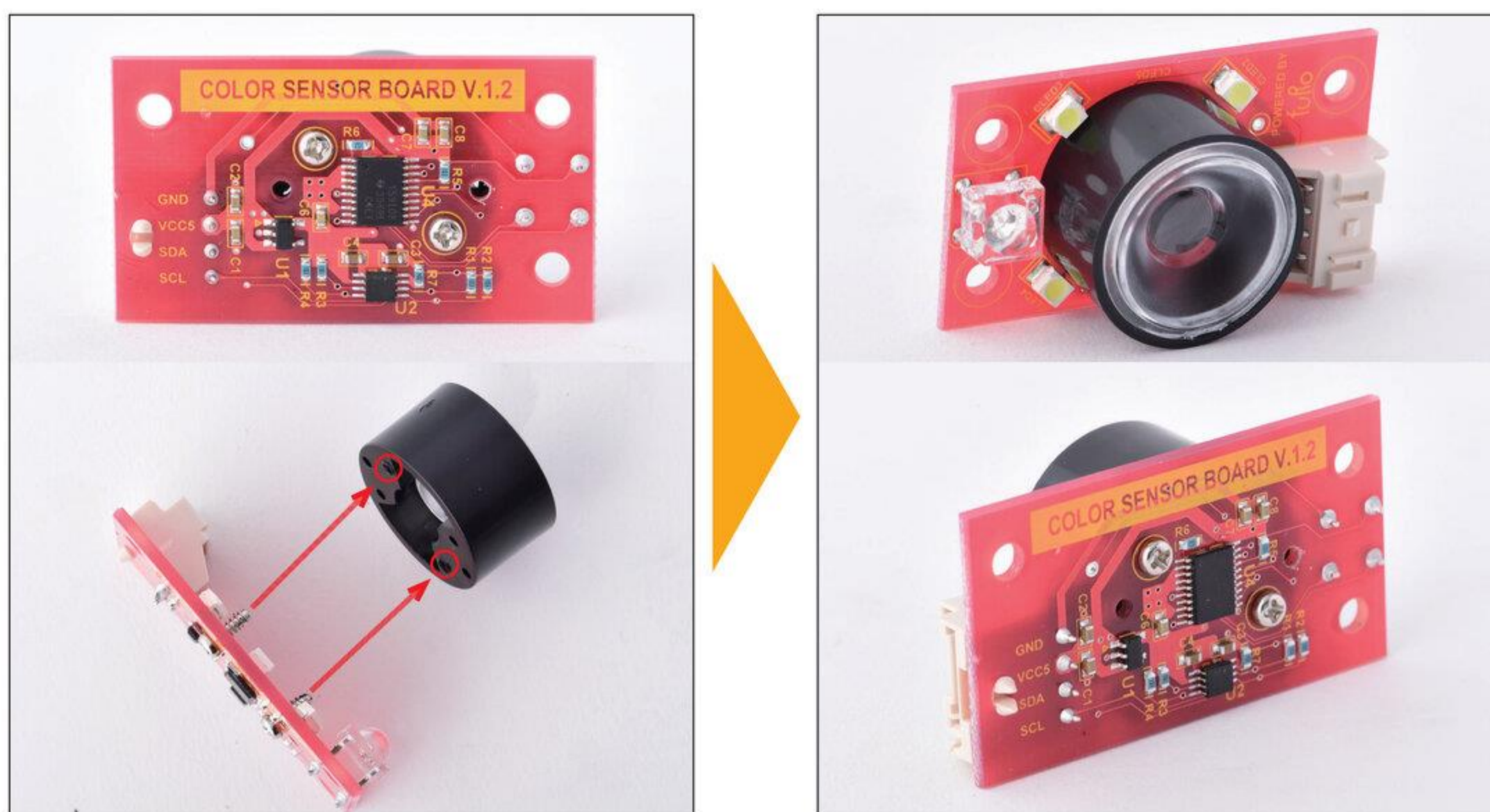


図1-0 カラーセンサーの組み立て

<組み立て手順②>

組み立てたカラーセンサーを、センサーケーブルを使って、[図1-1](#)のようにロボプロシールドのIICに接続しましょう。

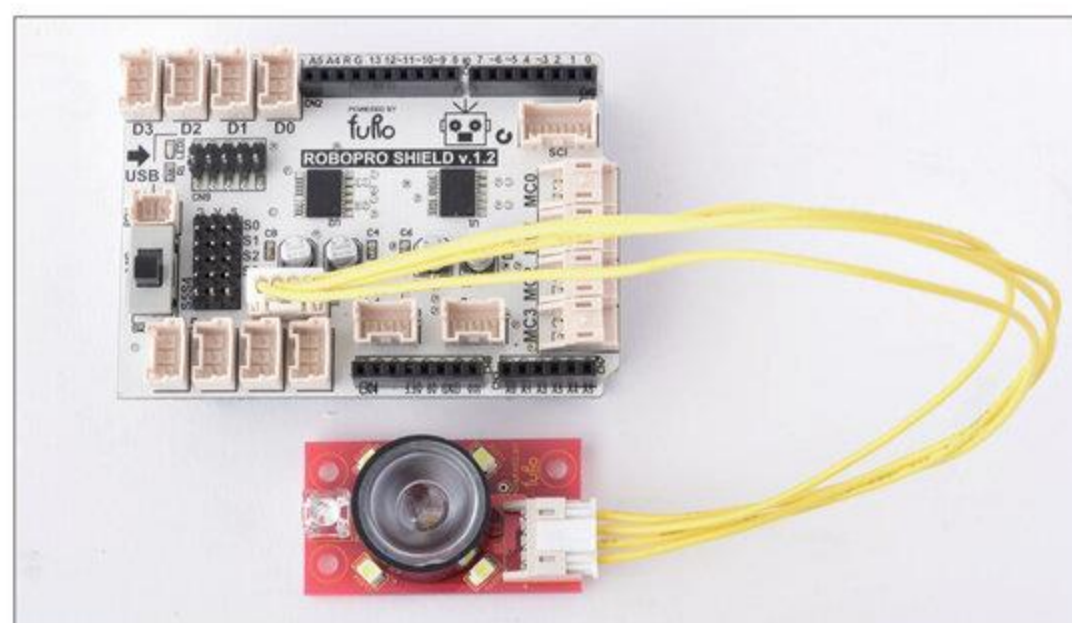


図1-1 カラーセンサーの接続

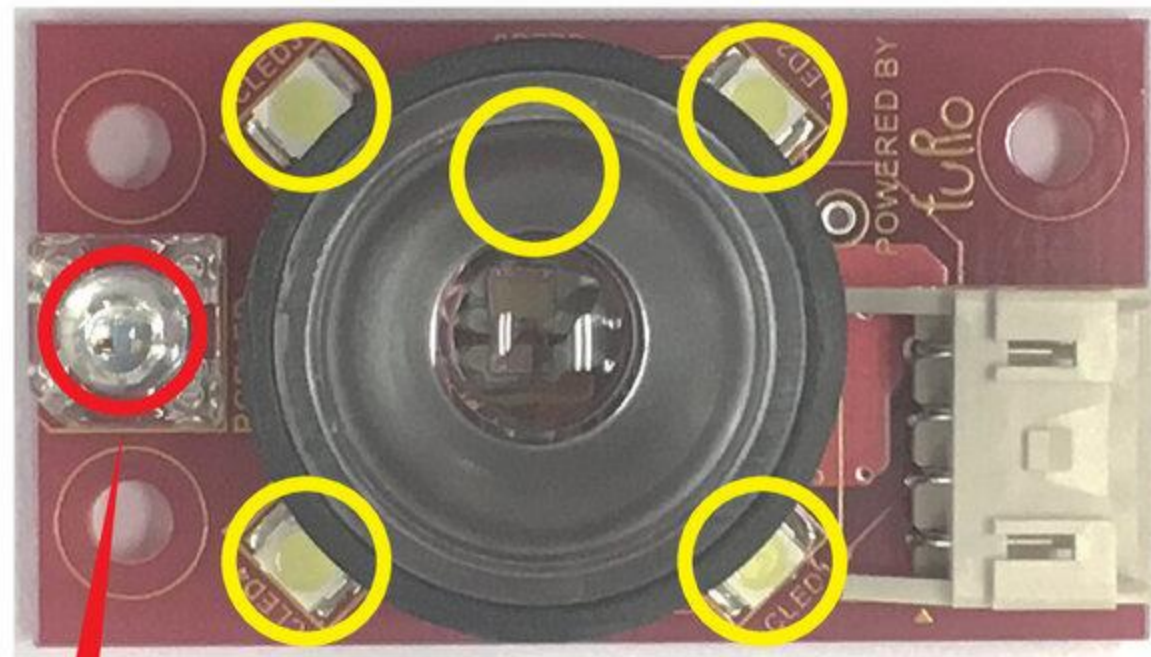
カラーセンサーの動作確認を行うため、以下のプログラムを実行してください。

∞ プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemB2 > ColorLED0

接続が上手くできていれば、カラーセンサーが何かしらの動作をしているはずです。どのように動作しましたか？

✎ カラーセンサーの6つの場所が順番に光る。(カラーセンサーが8回点滅する。)



青、緑、赤の3つの色が順番に点滅します

図1-2 カラーセンサーの点滅位置

1.1. スピーカーと電池ボックスの接続

スピーカーを、**図1-3**のように、ロボプロシールドの **[D2]** に接続しましょう。さらに電池ボックスも接続しましょう。

なお電池ボックスを接続せず、USBケーブルを通してのPCからの電源供給でも動作はします。ただし電池ボックスを使用しないと、カラーセンサーの色の読み取りの精度が悪くなるので、できれば使用しましょう。

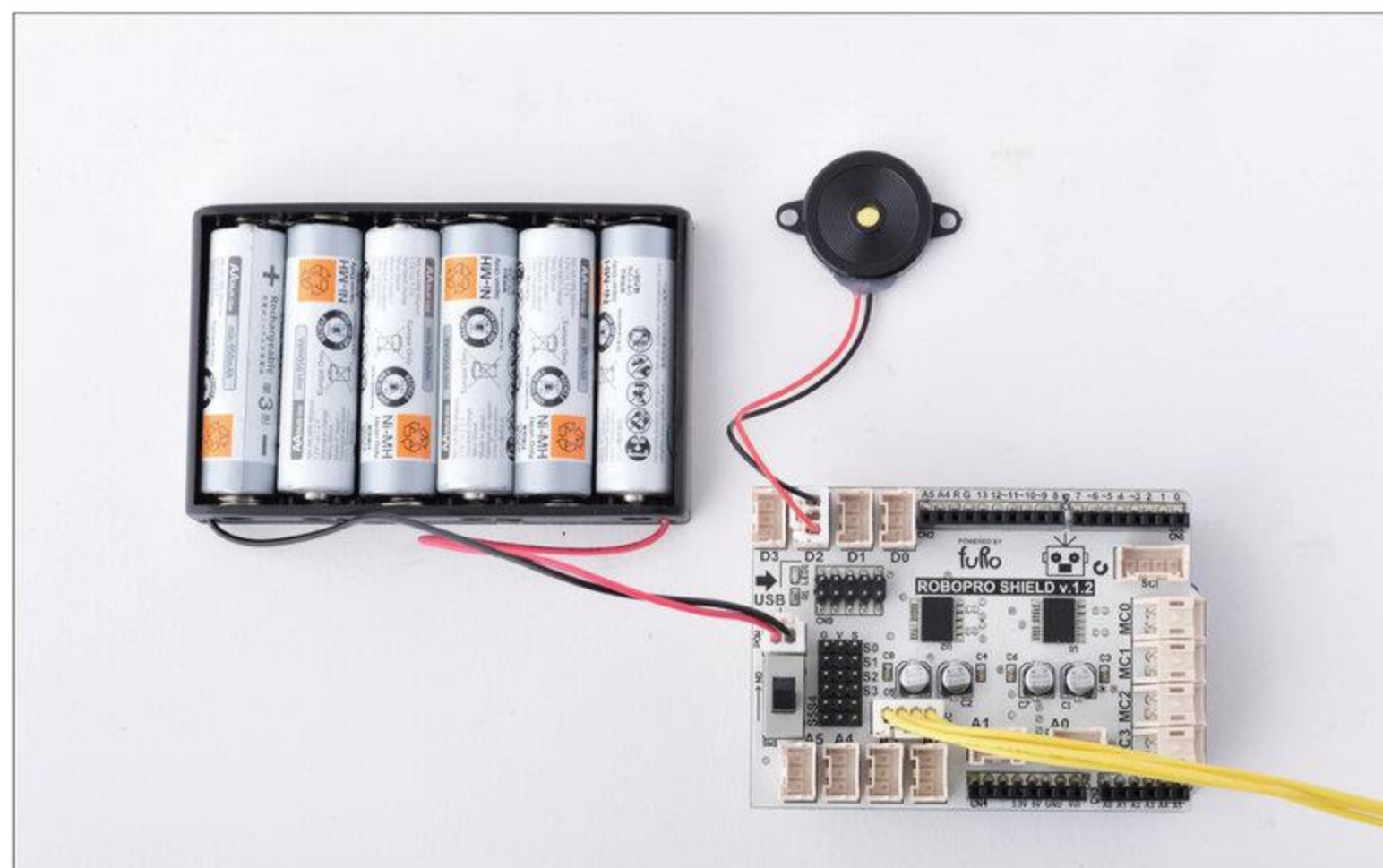


図1-3 スピーカーと電池ボックスの接続

1.2. 無線受信モジュールの接続とペアリング

続いて、コントローラーを使用するための準備をします。

1) 無線受信モジュールの接続

ロボプロシールドの [CN9] のコネクタにリボンケーブルを取り付けます。リボンケーブルの向きは、赤いラインが図1-4のように「ROBOPRO SHIELD v.1.2」とかいてある側にくるようにしましょう。

リボンケーブルのもう一方の先には、無線受信モジュールを取り付けます。赤いラインは、「Vstone」のマークがついている側にくるようにします。

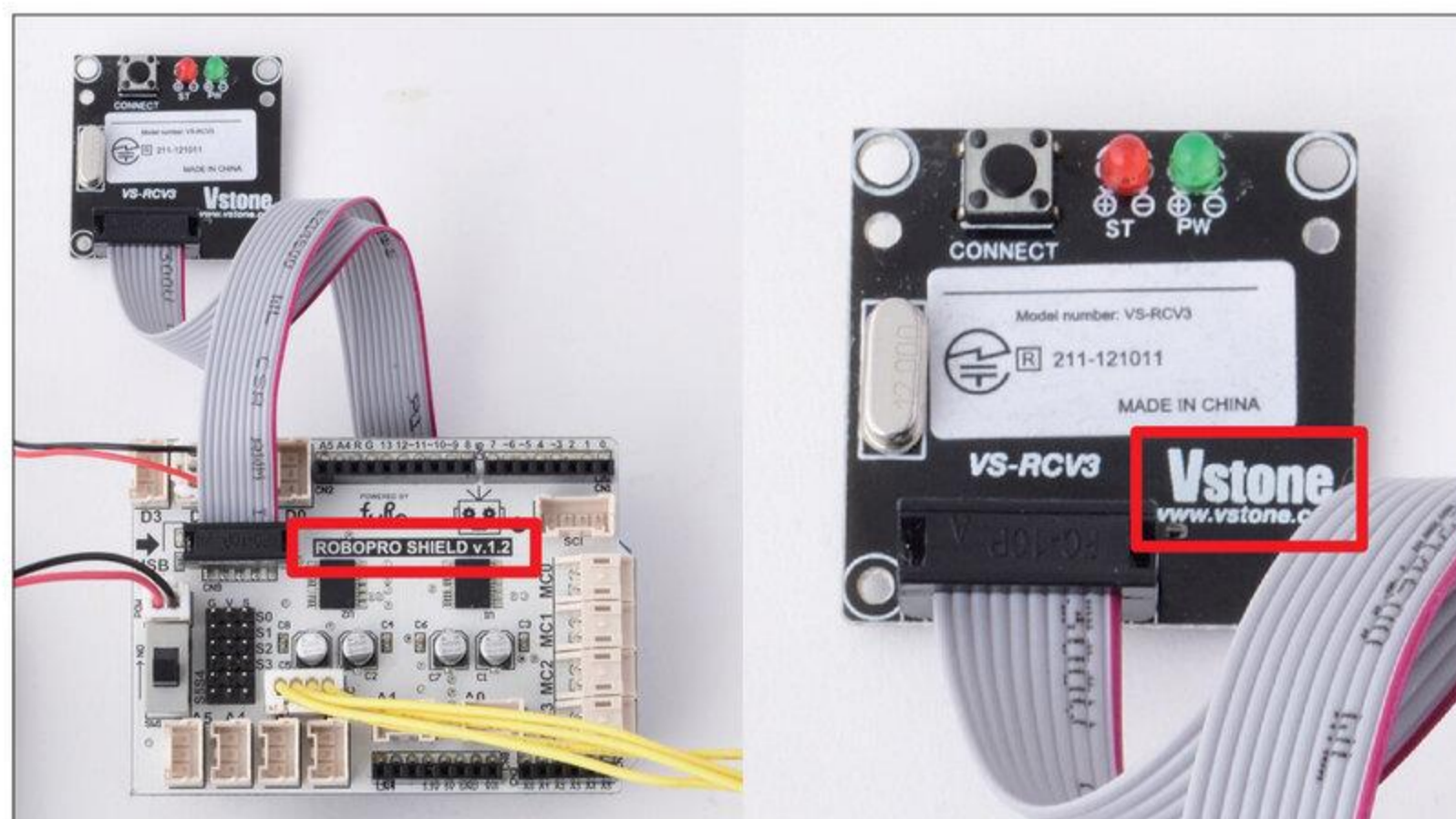


図1-4 無線受信モジュールの接続

正しく接続されていれば、電源が供給されたときに、無線受信モジュールの緑色LEDは点灯し、赤色LEDは点滅して^{てんめつ}いるはずですが、

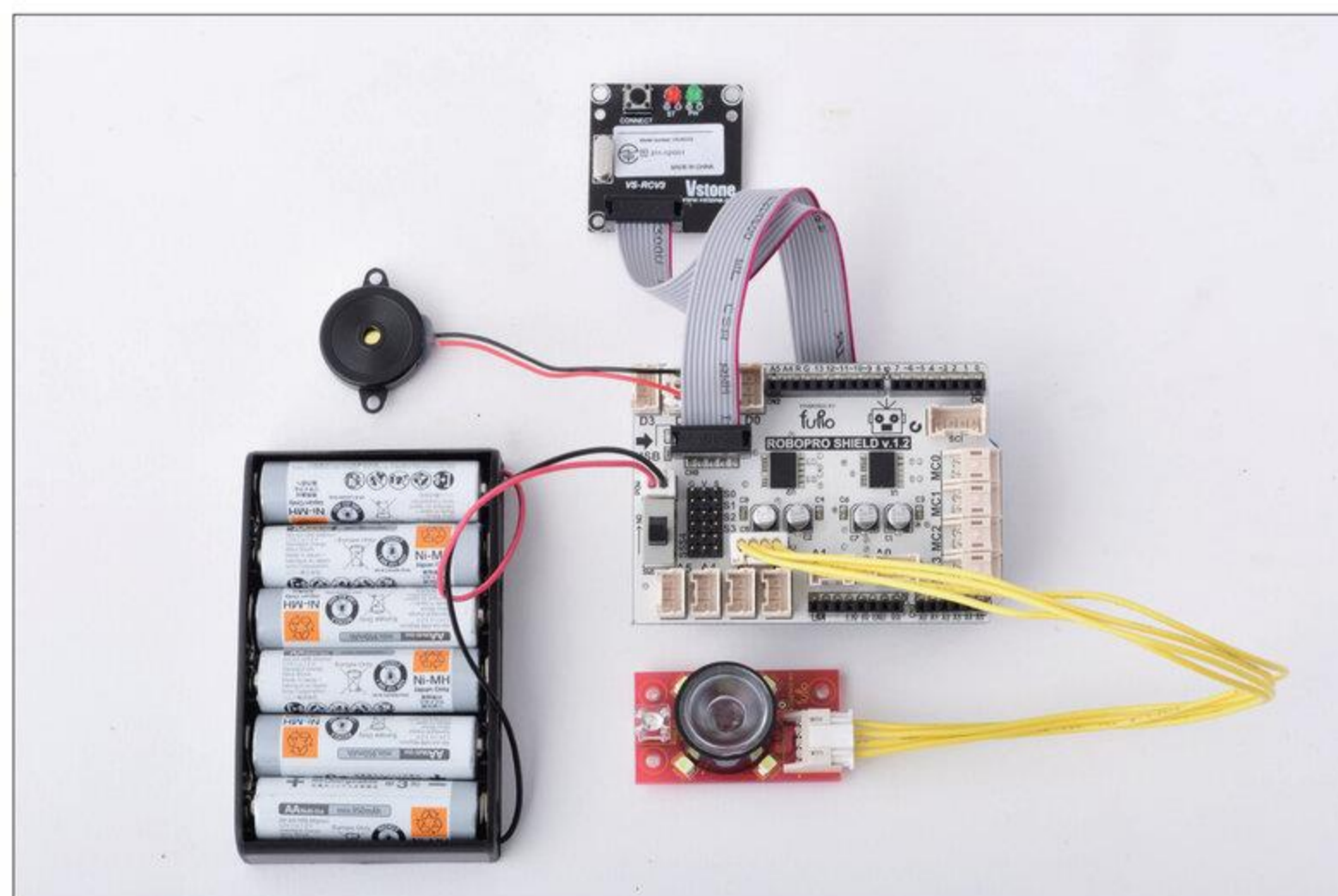


図1-5 接続完成図

2) ペアリング

コントローラーと無線受信モジュールを1組のペアに設定し、通信ができるように設定することを、ペアリングといいます。



POINT

●ペアリングのやり方

まず、無線受信モジュールの黒い「CONNECT」ボタンを押します。

次に、コントローラーの裏面の「POWER」をONにし、表面の「ANALOG」ボタンを押します。

- ・無線受信モジュールの赤色LEDが、点滅から点灯に変わる
- ・コントローラーの緑色LEDと赤色LEDが、点滅から点灯に変わる状態になれば、ペアリング成功です。



●コントローラー使用時の注意点

コントローラーはしばらく放っておくと、自動的に接続が切れて、無線受信モジュールの赤色LEDが点滅に変わり、コントローラーのLEDは消灯します。その場合はコントローラーの「START」ボタンを押せば、再びペアリングの状態になります。

なおコントローラーの詳しい使い方は、付属の説明書にもかかれています。

やってみよう!

コントローラーの動作確認をしよう。以下のプログラムを実行し、コントローラーの△○×ボタンを押すと、カラーセンサーのLEDが点灯するよ。どのボタンを押すと何色に光るかかき込んでみよう。

∞ プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemB2 > ColorLED1



これで、準備はすべて終わりました。いよいよ、色について学んでいきます。

2. 色について知る (目安 50 分)

2.0. 3原色を混ぜる

導入で少しふれましたが、コンピューターで色を扱うときは、絵の具と同じようにさまざまな色を混ぜて表現します。ただし、絵の具と違って、多くの色を使うことはしません。赤、緑、青の、3色だけを使います。この3色は、光の「3原色」といわれ、組合せ方によってさまざまな色を表現することが可能なのです。

1) プログラムの実行

それでは早速、その3色を混ぜて、どのような色に見えるのか体験してみましょう。コントローラーの動作確認で使ったプログラムがそのまま使えます。もう一度、以下のプログラムを実行してください。

プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemB2 > ColorLED1

すでに、どのボタンを押すと何色が光るかは確認しましたね？ 結果は、表2-0の通りです。また、複数のボタンを同時に押すと、混ざった色が表示されます。

表2-0 プログラム「ColorLED1」のボタンの割り当て

ボタン	表示される色
○	赤
△	緑
×	青

プログラムを見てみると、前回学習した「if文」が使われていることがわかります。

2) プログラムの確認

□ プログラム「ColorLED1」より抜粋 ぼっすい

```
void loop(){
  ps2x.read_gamepad();

  (中略)

  if(ps2x.Button(P SB_RED)){
    red = 100;
  }

  if(ps2x.Button(P SB_GREEN)){
    green = 100;
  }

  if(ps2x.Button(P SB_BLUE)){
    blue = 100;
  }
}
```

`ps2x.read_gamepad();` という命令で、コントローラーの各ボタンが押されているのかどうかという状況をチェックします。あとはもし `ps2x.Button(P SB_RED)`、つまり○ボタンが押されているなら `red = 100;`、つまり `red` という変数の値を0から100に変更へんこうします。他のボタンが押されたときの動きも、残りの部分を読んでみるとわかりますね！

ここで登場した3つの変数 `red`、`green`、`blue` は、最後に3色LEDを点灯させる命令で使います。

□ プログラム「ColorLED1」より抜粋 ぼっすい

```
ColorSensor.ledRGB(red, green, blue);
```

命令 「ColorSensor.ledRGB(a, b, c);」

実行結果：カラーセンサー上の赤色LED、緑色LED、青色LEDをそれぞれa、b、cの明るさで点灯する

使い方：ColorSensor.ledRGB(100, 0, 50);

ところで、前回if文を学んだとき、2つ目以降の条件には `else if～` というかき方を使っていました。

しかし、プログラム「ColorLED1」をよく見てみると `else` が登場せず、`if～`、`if～`、`if～` というかき方になっています。

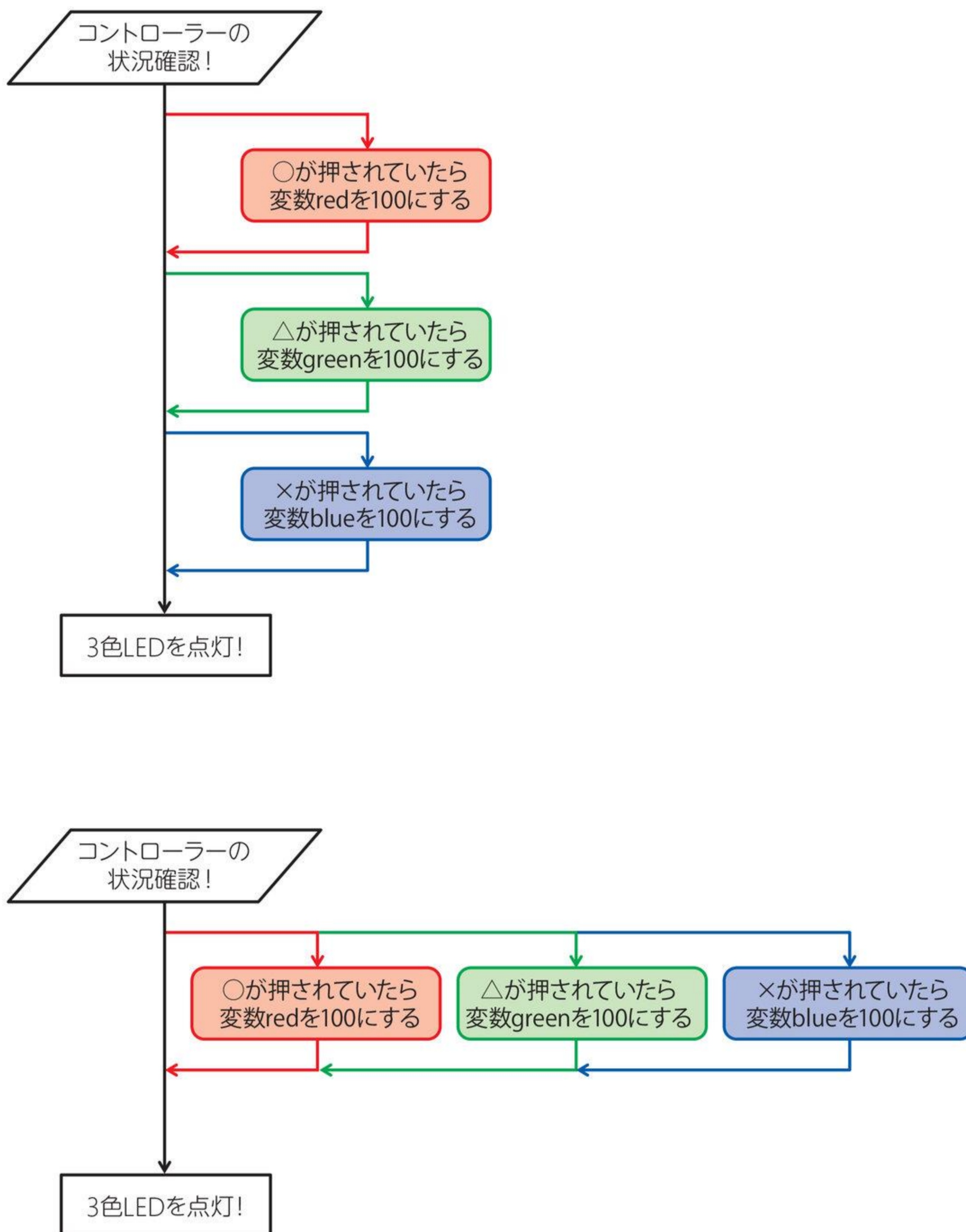



図2-0 elseがあるとき・ないときの処理の流れ

図2-0のうち、プログラム「ColorLED1」は上の図、2つ目以降の `if` を `else if` に変更したプログラムは下の図のように処理を進めます。どちらの図も同じブロックでできていますが、ブロックの組み合わせ方がちがいますね。実際の動作にもちがいは出るのでしょうか。

やってみよう!

`if` を `else if` に変更すると、動作にはどんなちがいが出るかな?
まずは図2-0から予想してみよう!

 ヒント

実際には最初のブロックから終わりのブロックまで、どのような道を通っていくのか矢印をなぞってみるといいかもしれないね!

動作はどう変わる?

 複数のボタンを同時に押しても1色しか点灯しなくなる

予想がかけた人や、どうなるか予想が出来ず困っている人は、「ColorLED1」を以下のようにかきかえて確認してみましょう!

 プログラム「ColorLED1」より**抜粋**

```
if(ps2x.Button(PSB_RED)){
    red = 100;
}

if(ps2x.Button(PSB_GREEN)){
    green = 100;
}

if(ps2x.Button(PSB_BLUE)){
    blue = 100;
}
```

```
if(ps2x.Button(PSB_RED)){
    red = 100;
}

else if(ps2x.Button(PSB_GREEN)){
    green = 100;
}

else if(ps2x.Button(PSB_BLUE)){
    blue = 100;
}
```

`else` を追加することで、ボタンを同時押ししても1色しか点灯しなくなりましたね！

図2-0の上の図ではLEDを点灯させるまでに赤、緑、青すべてのブロックを通るチャンスがありますが、下の図になるとたとえば赤のブロックを通ったら他のブロックにはたどり着けません。そのため、どれか1色のLEDしか点灯させることができないのです。今回は複数の色を混ぜて3色以外の色もつくりたいので、`else` をつけないプログラムのほうが合っているわけです。

講

なおこの場合、ボタンを同時押ししたときは「よりプログラムの上の行にかかっている処理」が優先されます。たとえば△と×を同時に押せば、△（緑色）が点灯することになります。

3) 表示される色

ではここで混ぜることによって表示される色をあらためて確認しておきます。色の表現には、さまざまな言葉があります。また、人の感じる色には個人差があるため、これが正解といった表現はないのですが、組み合わせでできる色の例は、表2-1のようになります。

なお、光の世界では、赤と青を組合わせたものを「マゼンタ」、緑と青を組合わせたものを「シアン」と呼びます。

表2-1 色の組合わせと表示される色

色の組合せ	表示される色
赤・緑	黄
赤・青	マゼンタ (ピンクに近い ^{むらさき} 紫、あざやかな ^{むらさき} 紫)
緑・青	シアン (あざやかな水色)
赤・緑・青	白

赤・緑・青を混ぜると、色のない白い光が、光っているように見えるのですね！

2.1. 光の強さを変える

ここまでで確認したとおり、3原色を混ぜて色を表現するとなると、赤、緑、青の単色に加えて、黄、マゼンタ、シアン、白の、計7色しか表現できていませんね。では、もっと多くの色を表現するにはどうしたらよいのでしょうか？

そこでヒントとなるのが光の強さです。実は、LEDは光の強さを変えることができます。LEDの光の強さ（輝度^{きど}）を変える主な方法は2つあります。



POINT

1. LEDに流れる電流量を変える
2. 点灯・消灯を短い時間で切りかえて、一定時間に点灯している回数を変える

1は、LEDの「流れる電流量が多いほど、明るく光る」という性質を利用した方式です。

2は、人間の目の性質を利用した方式です。人間の目は、光が短い時間に点灯・消灯をくり返していると、点滅しているようには見えません。例えば、蛍光灯も、1秒間に100～120回ほど点滅をくり返していますが、ずっと点灯しているように感じますよね？人の目が点滅を認識できるのは、1秒間に50～60回程度です。ですから、1秒間に100～120回ほど点滅している蛍光灯は、人の目には、ずっと点灯しているように見えるのです。そして、点滅している場合は、全部の時間点灯しているのと比べて、光が弱く、暗くなっていると人は感じるのです。

LEDの輝度を変化させる方法としては、どちらの方式でも可能です。では、マイコンが得意とするのはどちらでしょうか？これまで学んできたように、マイコンは、ON/OFFを高速に切りかえるのは得意でした！

というわけで、マイコンを使用する今回は、2の方式でLEDの輝度を変化させます。



豆知識

2は、「パルス幅変調 (PWM: Pulse Width Modulation)」という方式です。パルスとは、短時間に変化する電気信号のことです。ON (点灯) とOFF (消灯) のON の時間幅を調整・変化させるわけです。

プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemB2 > ColorLED2

このプログラムは、プログラム「ColorLED1」と同じく、コントローラーのボタンを押すと、LEDが点灯・消灯します。さらに、ボタンを押す度に、少しずつLEDの光り方が強くなったり、弱くなったりします。



図2-1 プログラム「ColorLED2」のボタンの割り当て

プログラムを見てみると、「ColorLED1」と同じくif文がたくさん登場します。ただ、「ColorLED2」のほうがifの中身が複雑になっていますね。ためしに、○ボタンが押されているときの動作を見てみましょう。

□ プログラム「ColorLED2」より**抜粋**^{ばっすい}

```

if(ps2x.ButtonPressed(PSB_RED)){
    if(red < 100){
        red = red + 1;
    }
    if(red == 100){
        tone(Speaker, NOTE_A3, 10);
    }
}

```

if の中に、さらに if が2つ入っています。ややこしいですが、じっくり読めば意味が分かるのではないのでしょうか。

変数 red の値の変えかたが先ほどと大きくちがうので、ここだけ確認しておきましょう。

□ プログラム「ColorLED1」より**抜粋**^{ばっすい}

```
red = 100;
```

□ プログラム「ColorLED2」より**抜粋**^{ばっすい}

```
red = red + 1;
```

red = 100; は、「変数 red の値を100にする」という命令でした。もとの red が0だろうが、99だろうが、24だろうが、とにかく処理後の値は100にしかありません。

red = red + 1; は「変数 red の値に1を足す」という意味で、もとの red が0だったなら1に、99だったなら100に、24だったなら25になります。

なお、red = red + 1; という式をまるまる red++; という文字列にかきかえてもまったく同じ動作になります。「インクリメント演算子^{えんざんし}」という記号になるので覚えておきましょう。ちなみに red--; とかくと「変数redの値から1をひく」という処理になります。こちらは「デクリメント演算子^{えんざんし}」とよびます。

やってみよう!

各色の輝度を増やすボタンをそれぞれ20回連打した色を混ぜてみよう。そして混ぜた色は何色になるかを観察して、以下の表に観察できた色をかき込んでみよう。

色の組合せと ボタンを押す回数	表示される色
赤20・緑20	黄
赤20・青20	マゼンタ
緑20・青20	シアン
赤20・緑20・青20	白

また、表示される色は、プログラム「ColorLED1」で色を混ぜたときと、どのようなちがいがああるかな？ 以下に記入してみよう。

 ちがいはない。(変わらない。)

プログラム「ColorLED1」と色の明るさは異なりましたが、色は、同じような「黄、マゼンタ、シアン、白」が表示されたと思います。各LEDの輝度が変わっても、同じ色を混ぜれば、同じ混ぜた色（黄、マゼンタ、シアン、白）になるということです。

つまり、色の性質には、「色味」、「色の明るさ」が、別々に独立した要素としてあるわけです。

やってみよう!

色を混ぜて、次の色を作ってみよう。各色LEDの輝度を増やすボタンをそれぞれ何回押したかかき込んでみよう。

表示する色	ボタンを押した回数	
黄	赤 回	緑 回
マゼンタ	赤 回	青 回
シアン	緑 回	青 回

講

ボタンを押す回数は、混ぜ合わせる色の組合せ同士では同じになります。例えば黄を表示するためには、「赤が10回のはきは緑も10回」、「赤が20回のはきは緑も20回」、押す必要があります。

色の明るさが指定されていなかったため、黄を作るのに、「赤20回、緑20回」で作った人や、「赤100回、緑100回」で作った人もいます。

色の要素に、「色味」、「色の明るさ」があることがわかったでしょうか？ 指定の色を作るときに、色味が一緒でも、色の明るさがちがう場合もあるわけです。



図2-2 「色味」は同じだが「色の明るさ」がことなる例

2.2. 色を変化させる

ここまで、LEDの輝度^{きど}を変化させましたが、各色のボタンの押す回数が各ペアで同じだと、表現できる色は、「3原色（赤、緑、青）+4色（黄、マゼンタ、シアン、白）」の7種類から増えませんね。

しかし、みなさんは、上の7色以外の色をもう目撃^{もくげき}しているはずです。たとえば赤を20回押した後に、緑の輝度^{きど}を0から20に変えると、緑が増える度に、赤から徐々に黄^{じょじょ}に色が変化していく様子が観察できたと思います。

チャレンジ課題

プログラム「ColorLED2」を使って、各色のLEDの輝度^{きど}を変化させてみよう。2色めを0から20、20から40まで増やしていったとき、何色から何色に変化したかかいてみよう。どの色か迷ったら、近いと思う色をかいてね。また、変化の間の色を観察しよう！

色の組合せと ボタンを押す回数	表示される色		
	0	20	40
赤20・緑0～20～40	赤	黄	緑
赤20・青0～20～40	赤	マゼンタ	青
緑20・青0～20～40	緑	シアン	青

輝度^{きど}が異なる各色LEDを混ぜると、黄、マゼンタ、シアン以外の、さまざまな色が作れることが体験できました。

2.3. 色の表現方式

1) RGBと色の要素

これまでやってきたような、光の3原色（赤、緑、青）の輝度^{きど}を変化させてさまざまな色を表現する方式を「RGB」と呼びます。「RGB」とは、「赤（Red）」、「緑（Green）」、「青（Blue）」の英語の頭文字を取ったものです。

ただし「RGB」は思った通りの色をピタリと指定するのが難しいため、実は別の表現方式もあります。

それが、先ほどから何度か出てきた「色の要素」を指定する方法です。これまで出てきた色の要素は、赤やシアンといった「色味」と、0や20や40などと指定した「色の明るさ」でしたが、他にもう一つ「色のあざやかさ」というものがあります。この3つの要素を指定してもさまざまな色を表現することができるのです。なお、3つの要素は専門用語ではそれぞれ「色相（色味）」、「明度（色の明るさ）」、「彩度（色のあざやかさ）」と記します。

2) 色相環^{しきそうかん}

さて、3要素を指定して色を表現するためには、それらを数字にする必要があります。

ここでは「色相^{しきそう}」について学びましょう。

まず、3つの原色のマスを円周上におきます。

3つのマスの間隔^{かんかく}を一定にして、赤が0度の場所にあるとすれば、緑は120度、青は240度の場所にきます。

さらに、マスのあいだに、原色をまぜてできる色のマスを付けたしていきます。図2-3では、追加で9種類の色をつくり付けたしてあります。たとえば0度（赤）と120度（緑）のちょうど中間の60度には、赤と緑を同じ量だけまぜてできた黄色のマスがあります。

こうすればそれぞれの色をその角度、つまり0から360までの数値に置きかえることができますね。「色相^{しきそう}」を表すときには、このような数値を用いることが多いです。ちなみに、図2-3のような円形の図を「色相環^{しきそうかん}」といいます。

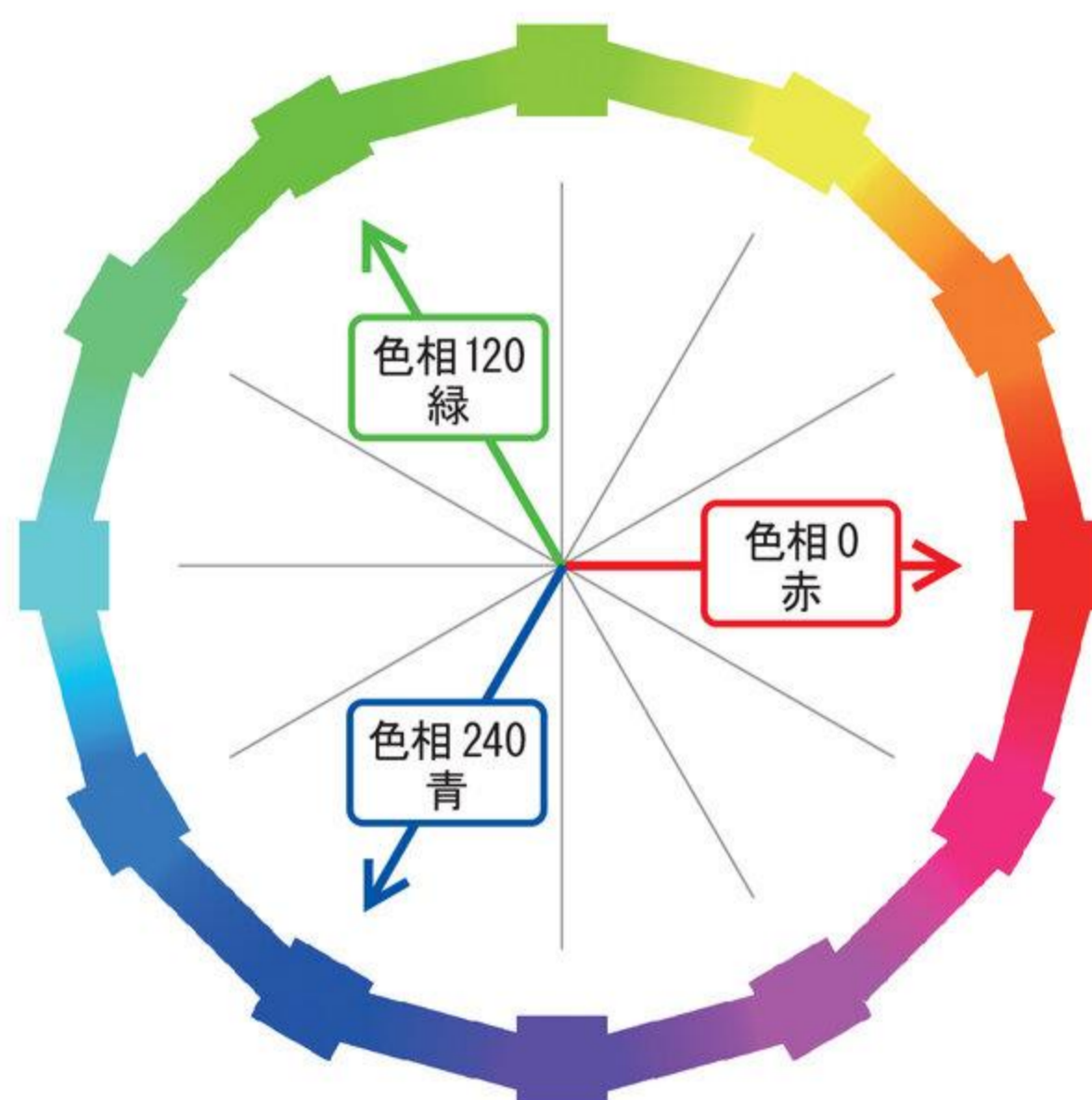


図2-3 色相環^{しきそうかん}

では早速、色の変化を体験してみましょう。
以下のプログラムを実行してください。

∞ プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemB2 > ColorLED3

実行結果：コントローラーの右アナログスティックをグルグル回すと、スティックのたおれている角度にあわせて、フルカラーLEDの色が変化する。さらに、L、Rボタンを押す度に、色の明るさも変化する。

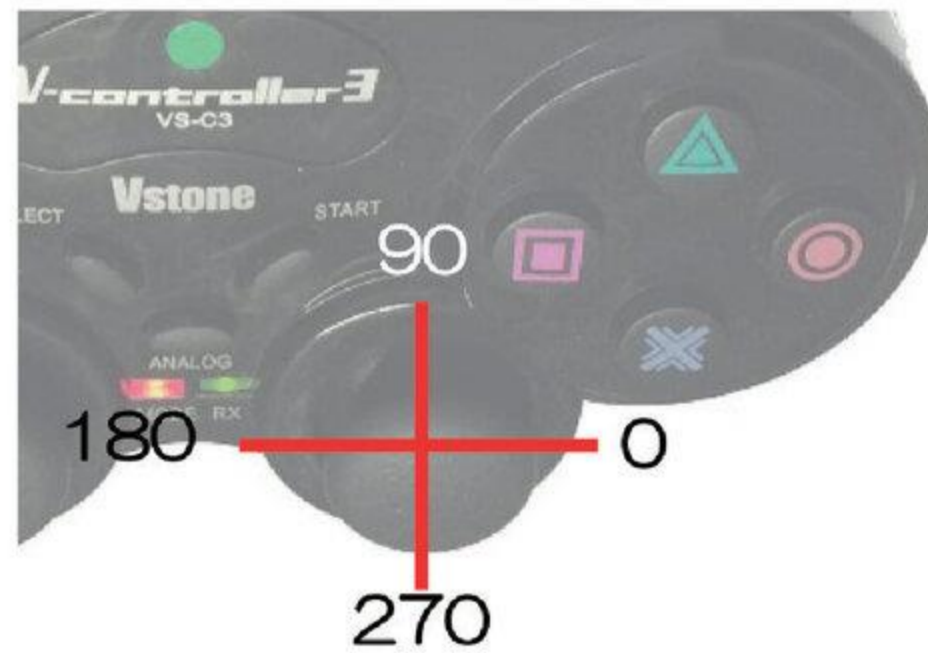
色の明るさは、最初は一番明るく設定されています。そのため、L、Rボタンを押す場合は、まずは、色の明るさを減らしてみてください。



図2-4 プログラム「ColorLED3」のボタンの割り当て

やってみよう!

1. 下図の角度を基準にして、赤、緑、青になっている角度(色相)は何度か調べてみよう。



💡 ヒント

各3原色付近になるとスピーカーから音が出るよ! 緑、青は音が出る範囲がせまいので、LEDの色にあわせて、ゆっくりとアナログスティックを動かしてみよう!

LEDの色	角度
赤	0 度
緑	120 度
青	240 度

2. 下の表の色は、右アナログスティックが何度のときに表示されるかな? なお、以下の色は、アナログスティックがその色付近の角度になっても、特にスピーカーから音は出ないので注意しよう。

LEDの色	角度
黄	60 度
シアン	180 度
マゼンタ	300 度

アナログスティックのたおれている角度（色相）にあわせて、LEDの色味が変化している様子を体験できたと思います。

まとめると、表2-2のようになります。

表2-2 LEDの色と色相（角度）

LEDの色	色相（角度）
赤	0度
黄	60度
緑	120度
シアン	180度
青	240度
マゼンタ	300度

3) HSV

これで、色相を角度という数字で表現できることがわかったと思います。

上の表は色相のみですが、色の3つの要素（色相、彩度、明度）を数値化して色を表現する方式を、「HSV」と呼びます。「HSV」とは、「色相（Hue）」、「彩度（Saturation）」、「明度（Value）」のそれぞれの英語の頭文字からとったものです。

なお、一般的には、コンピューターであつかいやすいのが「RGB」で、人がわかりやすいのが「HSV」であるといわれています。



豆知識

プログラム「ColorLED3」には、「アナログスティックの横方向&縦方向の倒れ具合をチェックし、色相を計算して求める」という処理がかいてあります。

```
hue = 180.0 * atan2(ry, rx) / 3.14;
```

求めた色相は `hue` という変数に入ります。計算式の中に見えない文字列が入っていますが、これは高校数学で学ぶ「三角関数」という知識を使っている部分です。

3. カラーセンサーで色を読み取る (目安 15 分)

3.0. 周囲の色を測定する

では、色についてわかったところで、私たちの身の回りのものが何色なのかを読み取ってみましょう！ 色の読み取りにも、カラーセンサーを使います。以下のプログラムを実行してください。

プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemB2 > ColorSensor1

プログラムを実行すると、カラーセンサーのLEDが点灯して、何かしらの色を表示していると思います。カラーセンサーは、**図3-0**のような、丸いレンズの下の部分で色を読み取っています。色を読み取りたい対象にレンズを向けてみてください。

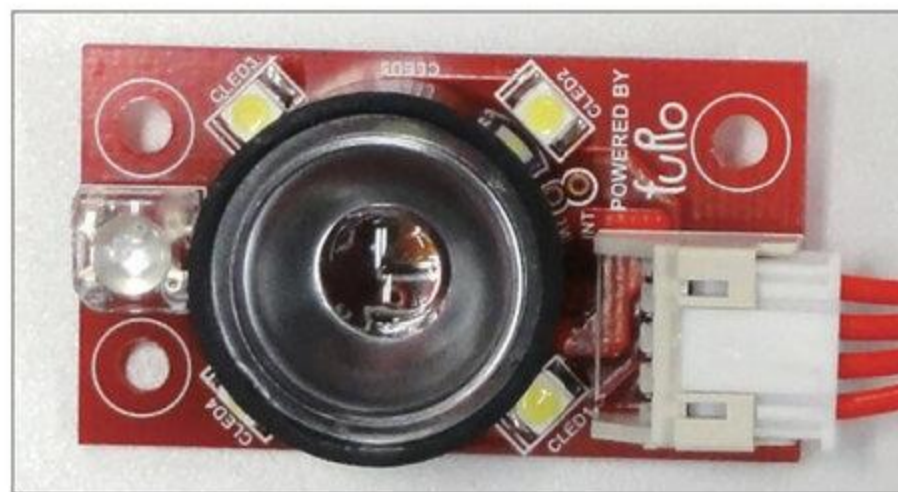


図3-0 カラーセンサー

やってみよう！

レンズを何も無い方向(天井など)に向けてみよう。LEDの色は何色に光っているかな？



結果は、環境によって変わります。例えば、かべや天井が白で、照明が蛍光灯であれば、緑から黄に点灯します。結果に多少の違いがあっても、基本的に特に何も無い方向に向けていますよね。私たちが白色や透明と感じている空間の色を、カラーセンサーは、LEDで表示している色として測定しています。

これは、光の要素である「明度」と「彩度」を捨てて、純粋に「色相」だけを表示しているためです。私たちの生活空間の色(主に照明の色)は、「色相」だけを取り出してみると意外な色ですよね！

さて、せっかくの色を読み取るためのカラーセンサーですから、今度は、色のはっきりしている対象を読み取ってみましょう。

やってみよう！

二人一組になって、一人はプログラム「ColorLED3」を実行しよう。そして、アナログスティックで操作して、さまざまな色を表示してみよう。

もう一人は、プログラム「ColorSensor1」を実行しよう。そして、相手の光っているLEDにカラーセンサーのレンズを向けて、色を読み取ってみよう。

色を読み取れる距離は、環境や対象により変化するよ。カラーセンサーと対象のLEDとの距離をいろいろ変えてみよう。

読み取った結果は、プログラム「ColorSensor1」をかき込んだ方のLEDに表示されるぞ！

どうでしたか？ 読み取る側と読み取られる側が適切な距離になっていれば、両方のLEDの色が同じになるはずですよ。

やってみよう！

プログラム「ColorSensor1」をかき込んでから、カラーセンサーを机の上に置こう。そして、テキストの巻末にある色見本のページを、カラーセンサーレンズの真上にかざそう。適切な距離になっていれば、色見本のページと同じような色がLEDに表示されるよ。

さらに、両方が同じ色になったと思える最大の距離をおおまかに調べて、次の表にかき込もう。

LEDの色	距離
赤	センチ
黄	センチ
緑	センチ
シアン	センチ
青	センチ
マゼンタ	センチ

講

基本的に、距離が近づくほど同じような色に近づきますが、教室の光の環境によって変わります。読み取れるだいたいの目安は5～15センチです。

また、色見本のページのうち黒だけは色相がないため色は表示されません。

3.1. 白黒を判別する

今回のカラーセンサーでは、色だけでなく、光の明るさも測定することもできます。この機能を使って、白（明るい）・黒（暗い）を判別してみましょう。
では、以下のプログラムを実行してください。

∞ プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemB2 > ColorSensor2

このプログラムでは、黒（暗い）と判定されると、スピーカーから音が出ます。

やってみよう！

巻末にある色見本の黒ページを机に置き、1センチくらいの距離^{きょり}のところでカラーセンサーを黒ページに向けてみよう。

黒ページの何も印刷のない白のところにカラーセンサーを向けても、かき込んだ^こプログラムのままでは、白・黒がうまく判別できずに、まちがって黒だと判別している音が出てしまうよ。

プログラム内の数字をかきかえて、白・黒が上手く判定できる数値を探そう。

💡 ヒント

プログラムの以下の部分を修正しよう。上手く判定できる数値は、1000よりも小さいよ。

```
// 1000の値を小さくしていくとセンサーと黒い色見本が  
// 1cmのところで白黒判定できる値が見つかります。  
bool isBlack = ColorSensor.colorIsBlack (1000);  
if(isBlack){  
    digitalWrite(Led, HIGH);  
    tone(Speaker, NOTE_F2, 50);  
}  
else {  
    digitalWrite(Led, LOW);  
}
```

この1000を小さくしていく

講

「500」にすると10センチ位、「100」にすると1センチ位の距離で反応しますが、数値は環境により違いがでます。

4. ワンドットカメラ (目安 15 分)

4.0. 色を撮影する

色を読み取るセンサーと色を表示するLEDがあれば、撮影範囲は1点分だけに限られますが、デジタルカメラが作れます！これを「ワンドットカメラ」といいます。早速、撮影を試みましょう。以下のプログラムを実行してください。

プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemB2 > OneDotCamera

撮影と表示には、コントローラーのボタンを使います。

○ボタンは、押した瞬間の色を撮影します。

×ボタンは押されている間、撮影された色を表示します。

巻末の色見本をはじめ、さまざまなものの色を撮影してみてください。



図4-0 プログラム「OneDotCamera」のボタンの割り当て

4.1. 撮影した色を混ぜる

ワンドットカメラで表示した結果は、プログラム「ColorSensor1」とほとんど同じだったのではないのでしょうか。そこで次は、2箇所を撮影して、その2箇所の色を混ぜることができるデジタルカメラにしてみましょう。

以下のプログラムを実行してください。

プログラムの書き込み

RoboticsProfessorCourse1 > MagicItemB2 > OneDotCameraAverage

プログラム「OneDotCamera」と操作はほとんど同じです。ちがうところは、もう1箇所の色さつえいを撮影できる点と、色を表示するとき、撮影した2箇所の色相かの平均しきそうを計算して表示してくれる点です。さまざまな色さつえいを撮影して、フュージョンしてみてください。



図4-1 プログラム「OneDotCameraAverage」のボタンの割り当て

やってみよう!

巻末の色見本を、次の表に示す組合せで撮影すると、どんな色として表示されるか調べてみよう。「3.0.周囲の色を測定する」で調べた距離きょりも参考にしながら、各色さつえいを撮影しよう。

撮影する色	表示される色
赤・緑	黄
赤・青	紫
青・緑	水
黄・シアン	緑
黄・マゼンタ	赤紫
シアン・マゼンタ	青紫

5. まとめ (目安 5 分)

今回は、色について学んできました。色を表現するのに、コンピューターがあつかいやすい「RGB」、人がわかりやすい「HSV」の2種類の方式がありましたね。さらには、色相環しきそうかんという円盤えんばんも出てきましたね。色は身近にあるものですが、意外と知らない側面があったことを、今回発見してもらえたのではないのでしょうか。

次回はセンサーを活用したロボットの組み立てを行います。お楽しみに！

講

- 以下の授業の目標を再確認します。
 - ・色 (RGB、HSV) について知る
 - ・カラーセンサーで色を測定する
 - ・ワンドットカメラを作る
- 次回テーマは「ベースロボットの組み立て」であることを告知します。

《次回必要なもの》

ラジオペンチ 1	ドライバー 1	USBケーブル 1	マイコンボード 1
			
ロボプロシールド 1	電池ボックス 1	ギアドモーター 2	リボンケーブル 1
			
コントローラー 1	無線受信モジュール 1	モーターL字ステイ 2	センサーL字ステイ 1
			
オムニホイール 3	M2.3L20タッピングネジ (A) 2	M3L30ネジ 1	白円形ボード 1
			
M3ナット 21	M3L5ネジ 4	M3L8ネジ 16	M3L25ネジ 4
			
8mm角スペーサー 4	30mm角スペーサー 3	ユニバーサルボード 1	
			

図5-0 次回必要なもの

付録：色見本

付録として色見本を添付^{てんぷ}します。ただし、印刷した色は、厳密には、各色モデルで示した色とは異なりますので、参考程度に使うとよいでしょう。

色：赤

RGB：赤255、緑0、青0

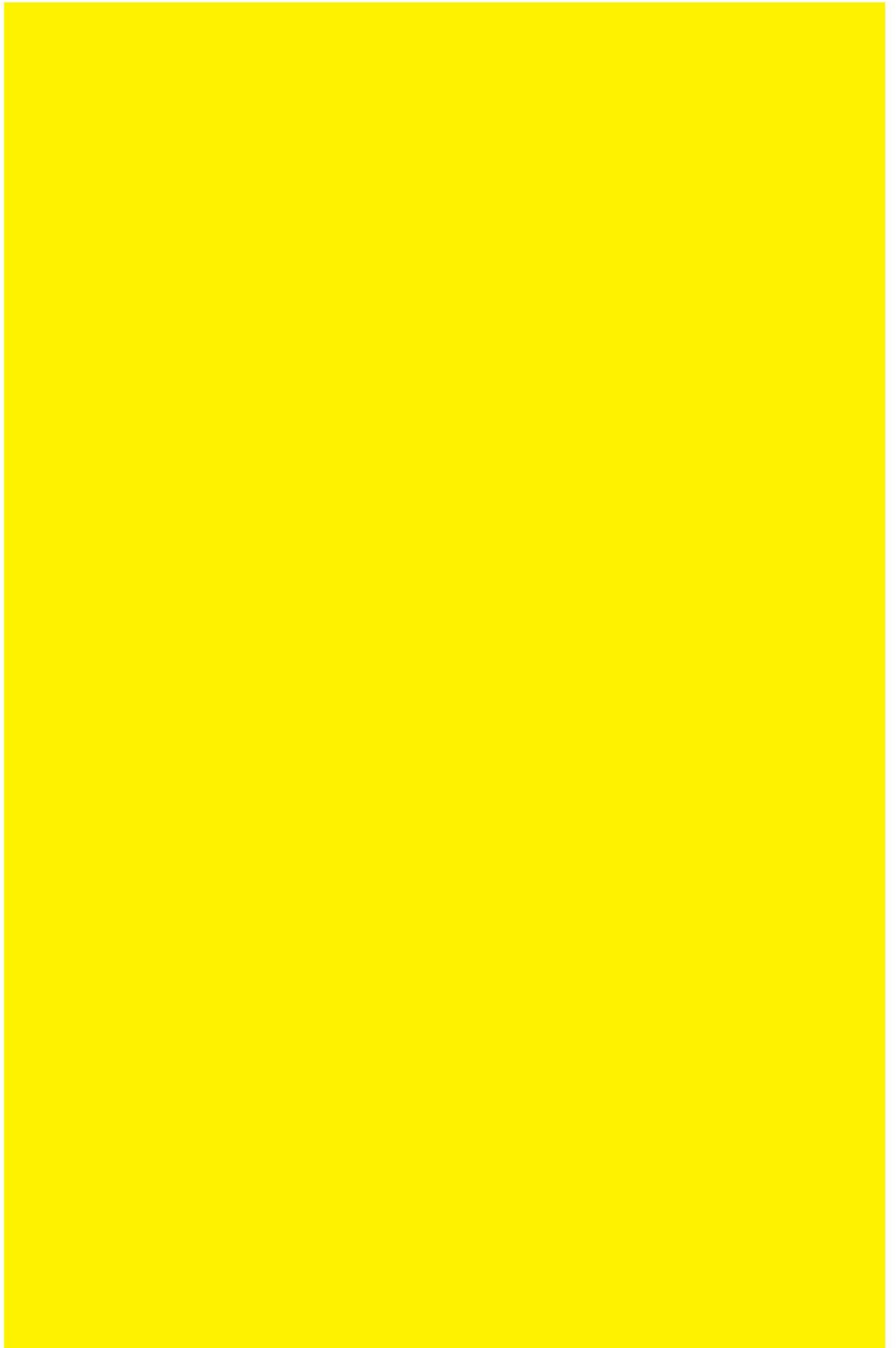
HSV：色相0度、彩度100、明度100



色：黄

RGB：赤255、緑255、青0

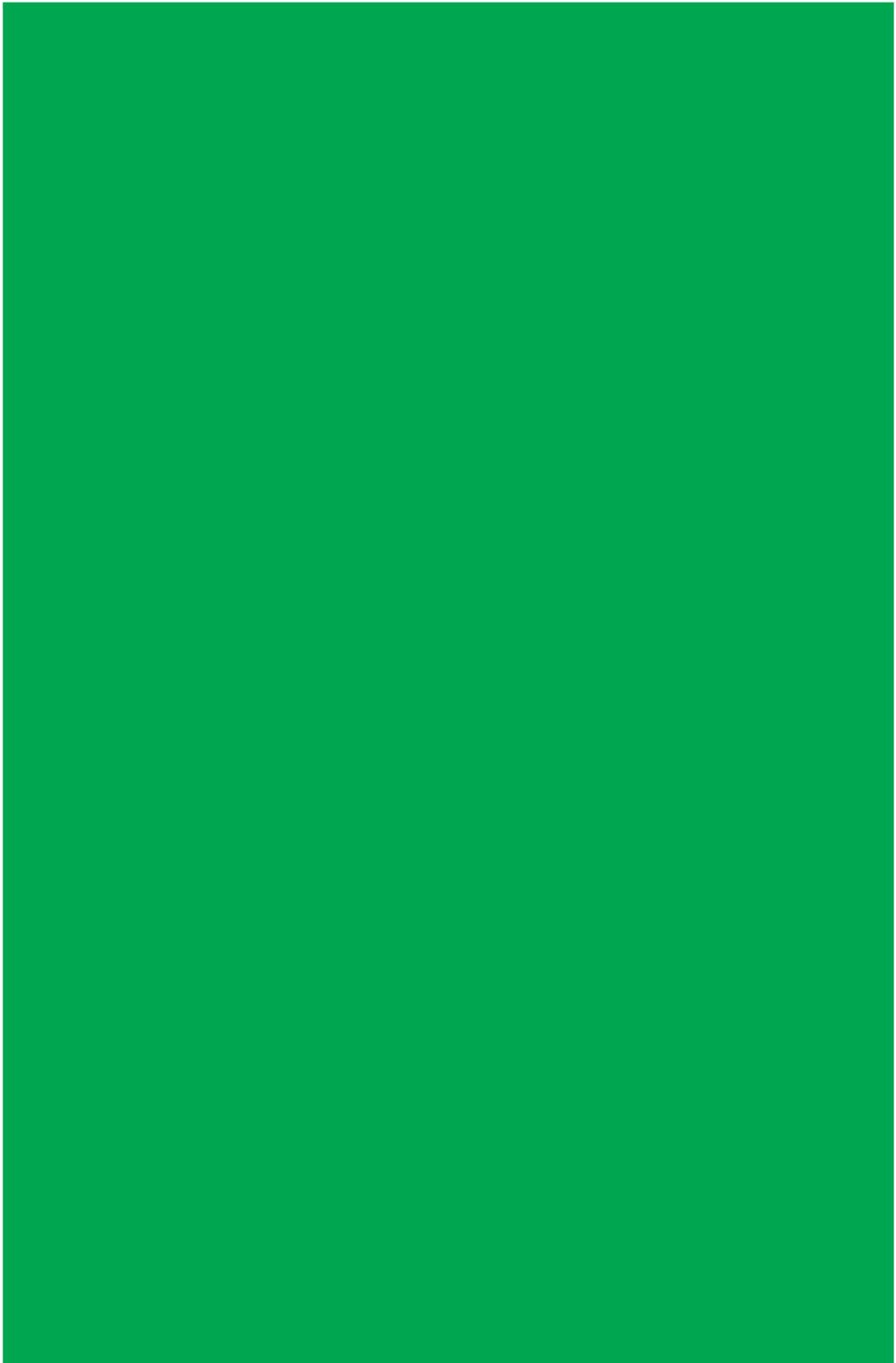
HSV：色相60度、彩度100、明度100



色：緑

RGB：赤0、緑255、青0

HSV：色相 120度、彩度100、明度100



色：シアン

RGB：赤0、緑255、青255

HSV：色相 180度、彩度100、明度100



色：青

RGB：赤0、緑0、青255

HSV：色相 240度、彩度100、明度100



色：マゼンタ

RGB：赤255、緑0、青255

HSV：色相 300度、彩度100、明度100



色：黒

RGB：赤0、緑0、青0

HSV：色相なし(0でもない)、彩度0、明度0

