

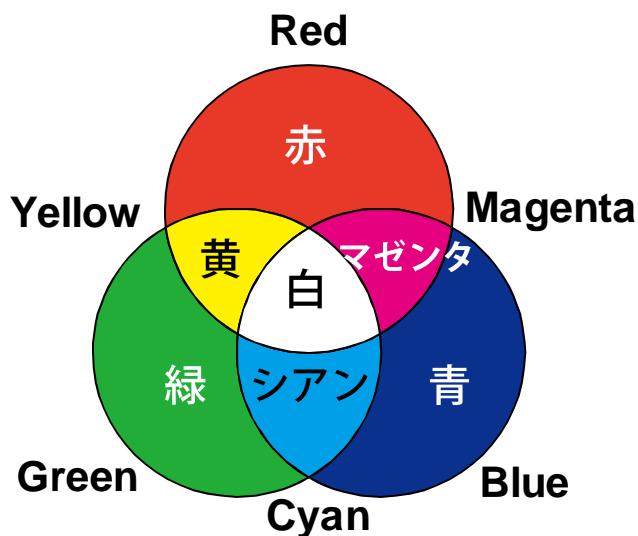
光の三原色

赤R、緑G、青Bを光の三原色と呼びます。

赤と緑が重なると黄色Yに、緑と青が重なるとシアンCに、青と赤が重なるとマゼンタMになります。そして赤、青、緑が重なると白 White になります。これを加法混色と言います。

赤・青・緑の光の割合を少しずつ変化させていくと、その光の色は徐々に変化していきます。混ぜる光の色を制御すれば、ほとんど全ての色を作り出すことができます。

色素系(絵の具やインク等)の三原色では、減法混色となり赤・青・緑の絵の具を重ねると黒になります。



自然な白色（太陽光）発光は、赤 3：緑 6：青 1

の割合になります。

赤色の光度を 300 ミリカンデラとすると
緑色の光度は

$$300 \text{ ミリカンデラ} \times 6/3 = 600 \text{ ミリカンデラ}$$

青色の光度は

$$300 \text{ ミリカンデラ} \times 1/3 = 100 \text{ ミリカンデラ}$$

赤と緑で黄色を発光させると

$$300 + 600 = 900 \text{ ミリカンデラ}$$

赤と青でマゼンタを発光させると

$$300 + 100 = 400 \text{ ミリカンデラ}$$

緑と青でシアンを発光させると

$$600 + 100 = 700 \text{ ミリカンデラ}$$

白色光は赤単色に比べて約 3 倍の明るさになります。人工的に作り出す光は、自然光に比べて青色が強くなりがちのため、光度の調整が必要です。

フルカラーLED は、このように白色光をつくることができ、他にも様々な色の光を発光します。

フルカラー RGB LED の仕様	
標準電流：20mA	
光 度	
赤	1560 ミリカンデラ
緑	2180 ミリカンデラ
青	1120 ミリカンデラ

フルカラーLED 拡張ボード for micro:bit

micro:bit にフルカラーLED を簡単に接続できる拡張ボードです。

5つのスペーサーを使って、micro:bit の P0、P1、P2、GND、3V 電源と接続できます。

フルカラーLED には、内部に小さな赤・緑・青の小さな LED が内蔵されており、これらを明るさを調整して光らすことで、多くの色を発光することができます。

接続は以下のようになっています。

- ・ P0 ピンと赤 LED (750Ω)
- ・ P1 ピンと緑 LED (330Ω)
- ・ P2 ピンと緑 LED (330Ω)

赤 LED は、流す電流に対して明るく光るので、ピンと LED の間の抵抗を大きくしてあります。

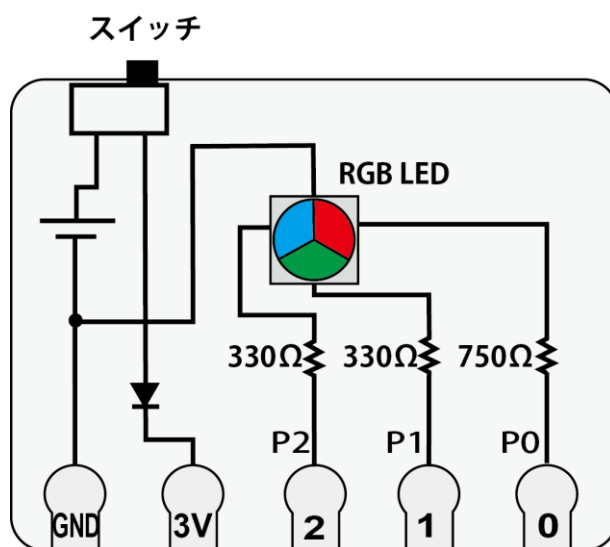
実際、3つの色を点灯した場合、LED の特性が異なるため、MakeCode の アナログピン出力値を調整して、目的の色を光らせてください。

フルカラーLED 拡張ボードは、ボタン電池(CR2032)で micro:bit を動作させる機能もあります。

ボードの裏側に、micro:bit を接続する前に、セットする必要があります。micro:bit 接続後は、はずれない構造になっています。電池を交換する場合は、micro:bit を1度取り外します。

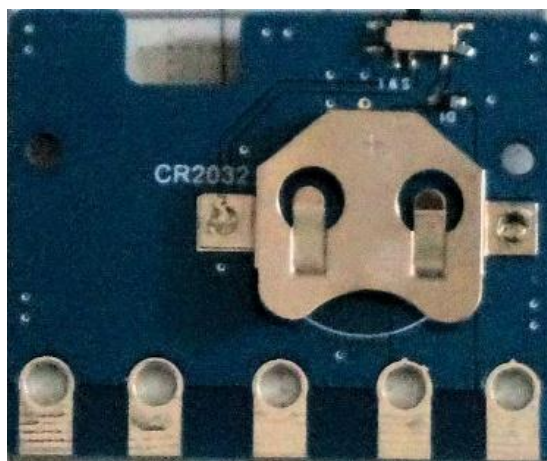
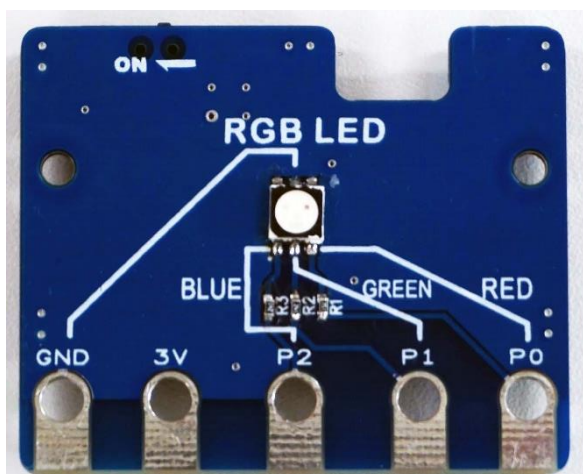
電池は、ダイオードを介して micro:bit につながっているので、micro:bit 側から電池へ電流が流れることはないのです、ボード接続した状態でも USB ケーブルをつなげることができます。

小型スイッチを ON にすると、USB ケーブルの接続がなくても micro:bit が動作します。ボタン電池は出力電流が小さいので、LED の明るさは少し暗くなります。単独で、micro:bit が動作するので、通信機能を使った LED の制御等のプログラムが作成・動作可能です。



表側

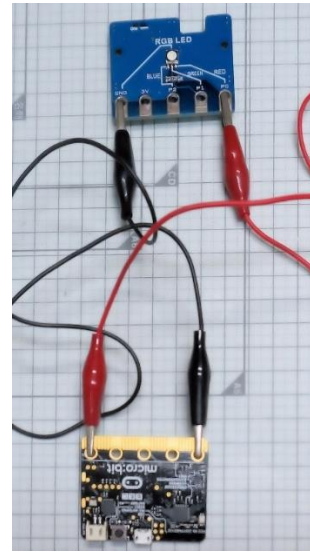
裏側



Action1: 赤・緑・青の LED を光らせてみよう。

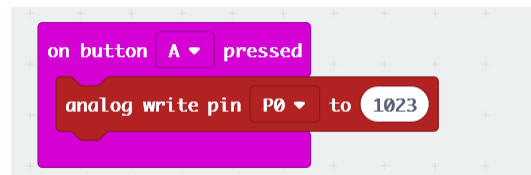
micro:bit と LED ボードを接続します。

- ① ワニクリップケーブル黒で GND 同士を接続します。
- ② ワニクリップケーブル赤で P0 同士を接続します。

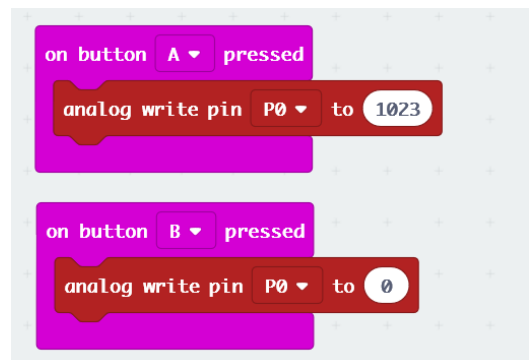


- ③ 接続ができたなら、USB ケーブルを接続し、プログラミングを行います。

A ボタンを押すと、赤の LED が光ります。

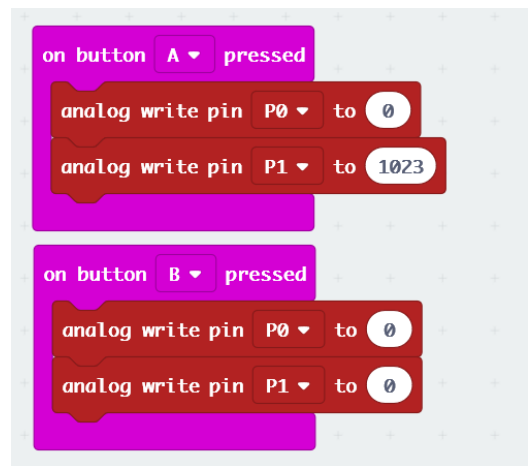


- ④ B ボタンを押すと消える処理を追加します。



応用：パラメータを変化させて暗く光らせてみます。

- ⑤ ワニクリップケーブルで P1 同士を接続します。
- ⑥ P1 への出力のプログラムを作成します。P0 の出力は 0 とします。
緑の明るさを確認します。



⑦ ワニクリップケーブルで P2 同士を接続します。

⑧ P2 への出力のプログラムを作成します。P0 と P1 の出力は 0 とします。

青の明るさを確認します。最も暗く表示されるはずです。



Action2: 黄・マゼンタ（明るい赤紫）・シアン（明るい水色）を光らせてみよう。

① P0 と P1 への出力のパラメータを 500、P2 は 0 にします。

黄色、やまぶき色、オレンジが表示されます。

② P0 と P2 への出力のパラメータを 500、P1 は 0 にします。

マゼンタが表示されます。

③ P1 と P2 への出力のパラメータを 500、P0 は 0 にします。

シアンが表示されます。

④ P1,P2,P3 のパラメータを調整して白色（太陽光）を表示させてみよう。

最初の説明を確認して、数値を変えます。

Action3: 7 つの色の表示を関数にしてみよう。

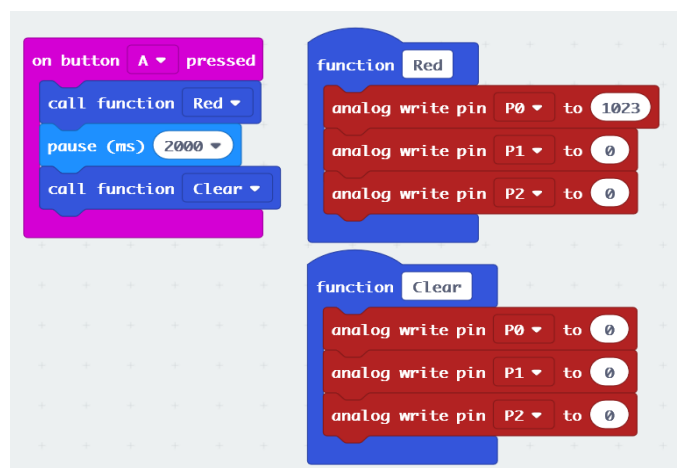
同じ処理を繰り返す場合や、プログラムの流れを見やすくするために関数(Function)を使います。Advanced をクリックして表示される Functions カテゴリーを使います。

① Make a Function で 7 色の表示関数を作成します。

Red, Green, Blue, Yellow

Magenta,Cyan

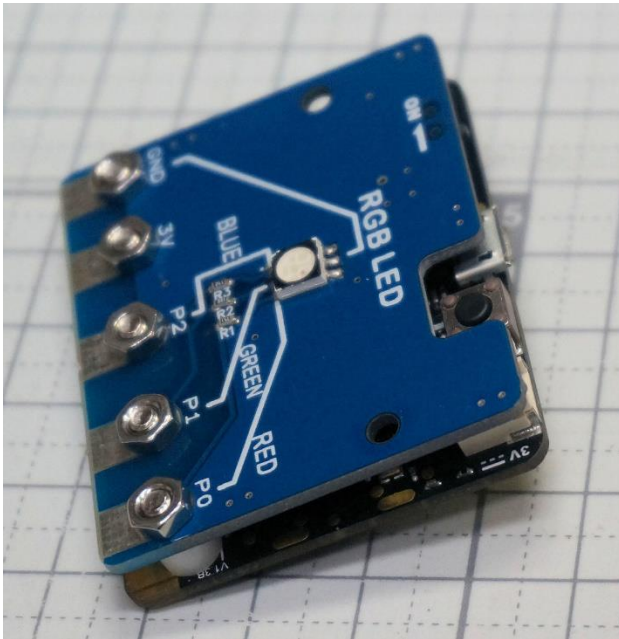
の関数を作成します。



- ② PAUSE ブロックを挟んで、連続的にすべての色を光らせます。
- ③ ファイル名を RGBLED として保存しておきます。



micro:bit を接続



micro:bit の通信機能

micro:bit は Bluetooth モジュールを搭載していて 2 つの micro:bit の間でメッセージの送信・受信をすることができます。無線通信をするためには、例えば radio グループのブロックを使います。ここでは「送信者が A ボタンを押した時にメッセージを送信し、受信者がメッセージを受け取った時に表示するプログラム」を作ってみましょう。

Action4: 2 つの MICRO:BIT で通信をしよう。

はじめに、送受信双方で、無線のチャンネル（グループ）を設定します。次に、送信側では、A ボタンを押した時に、あるメッセージを送る処理を行い、受信側では、メッセージを受け取った時に、それを画面に表示する処理を行います。

① on start ブロックに無線通信をするためのグループを設定します。（数値はグループ番号）

② on button A pressed ブロックを配置して、そこに、ボタン A が押された時に起こるイベントのブロックを当てはめます。ここでは、radio send string ブロックを用い

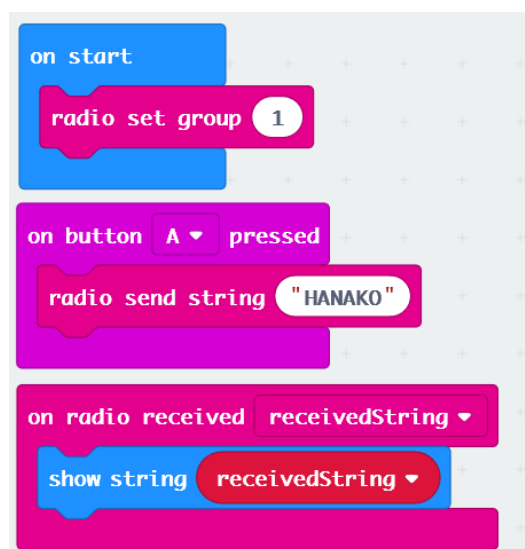
て、自分の名前の文字列を送信します。

③ on radio received を配置して、そこに、メッセージを受け取った時に起こるイベントのブロックを当てはめます。ここでは、show string ブロックを用いて、受け取った

文字列（receivedString）を表示させます。

送信者が A ボタンを押すと送信者の名前が受信者の micro:bit に表示されます。

micro:bit の通信は数値を送受信することもできます。

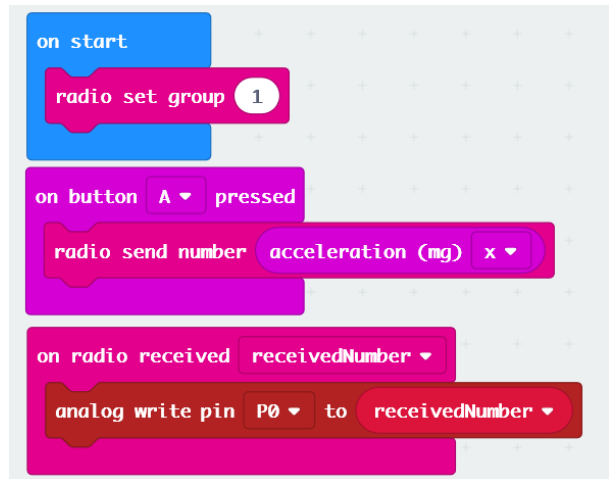


Action5: 他の MICRO:BIT の LED を通信により光らせてみよう。

リアルタイムに加速度センサーの X 方向の値(-1023~+1023)を送信して、その値を受信した側は、赤の LED のパラメータとして光らせます。

① on start ブロックに無線通信をするためのグループを設定します。

- ② forever ブロックに加速度センサーの X 方向の値を送るためのブロックを配置します。
- ③ on radio received を配置して、その中に数値を受け取った時に赤色の LED の出力パラメータを変えて光らせるブロックを配置します。



送信側がブロックを傾けると、受信側の赤色 LED の明るさが変わります。

-1023~0 の数値が送られると LED は消えます。

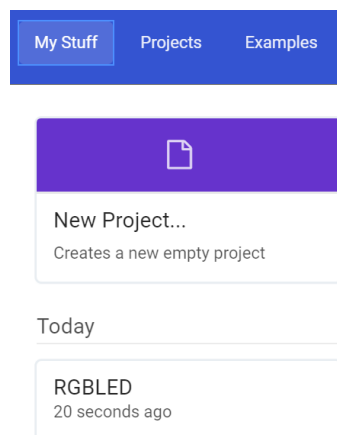
発展：

-1023~+1023 の送信値すべてで、連続的に明るさが変化する受信側のプログラムを作ってみよう。

Action6: 全員の MICRO:BIT の LED を通信により 7 色に光らせてみよう。

- ① Action3 で保存したプログラムを読み込みます。

左上の Project をクリックすると RGBLED が表示されますので、それを選択します。
読み込んだら、ファイル名を COMLED に変更します。



- ② on button A pressed ブロックを配置して、ボタン A が押された時に加速度センサーの X 方向の値を送るためのブロックを配置します。

Action5 の on button A pressed ブロックと同様です。

- ③受信側の処理を作成します。

ブロックが多いので、作業量が多くなりますが、動作を理解して進めると間違えずにプログラミングが行えます。

if ブロックは最初に配置したときに elseif を必要な数だけ作成しておきます。

- ④ 最後の on start ブロックに無線通信をするためのグループを設定します。

(全員が 100 にします)



※全体ができるまで、A ボタンを押さないでください。