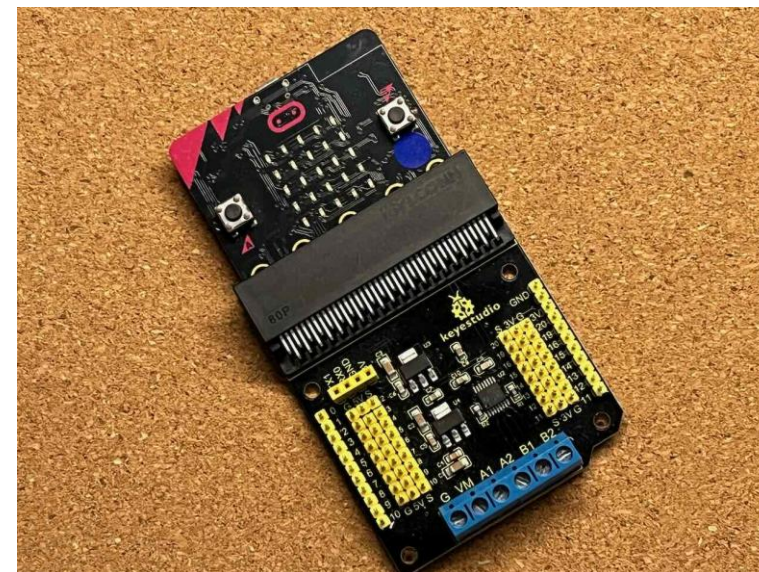
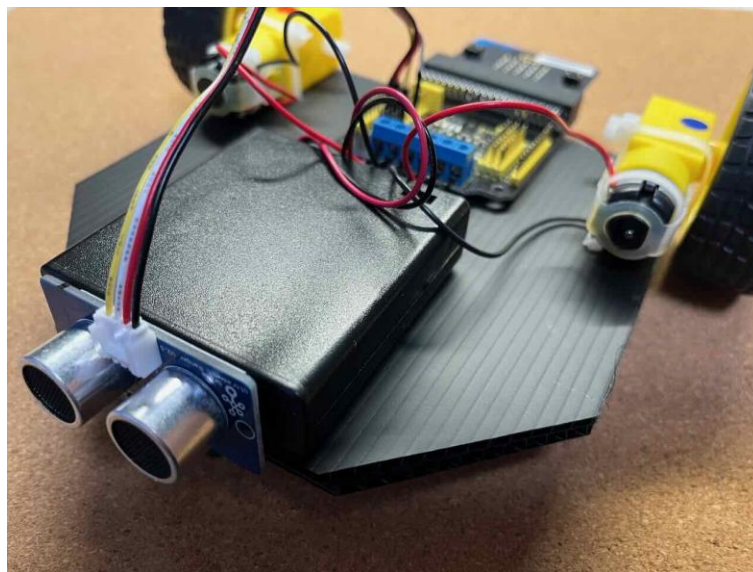


さとやまプログラミングクラブ2025

マイクロビットで「ロボカー」をつくろう！

第3回

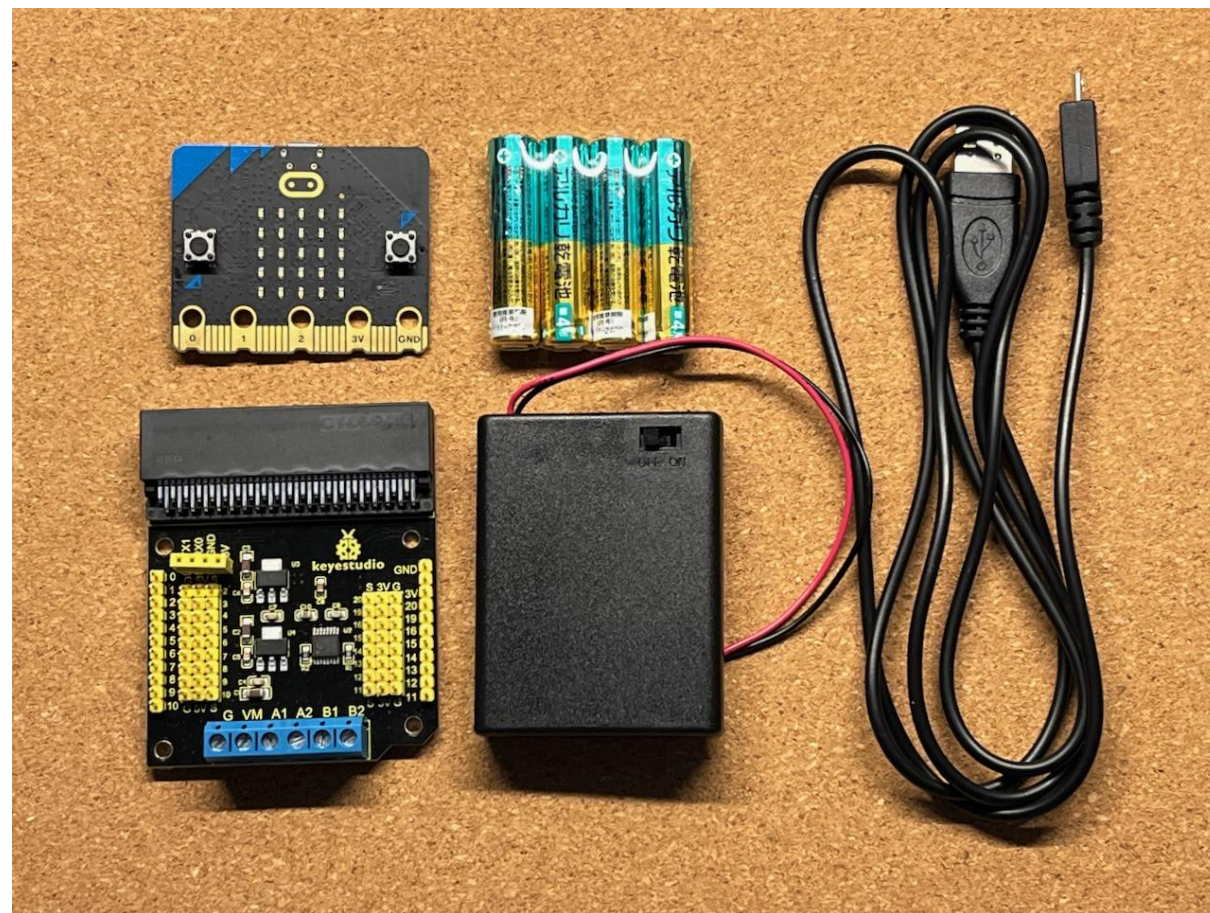


今日やること

- マイクロビットのプログラムでロボカーを制御します。
- プログラムの仕様は以下のとおりです。
 - ふだんはロボカーは「前進」します。
 - 障害物にちかづくときすこしのあいだだけ「右折」または「左折」して、ロボカーのむきをかえます。
 - むきをかえたらまた「前進」します。
- ロボカーを制御するプログラムを改良します。

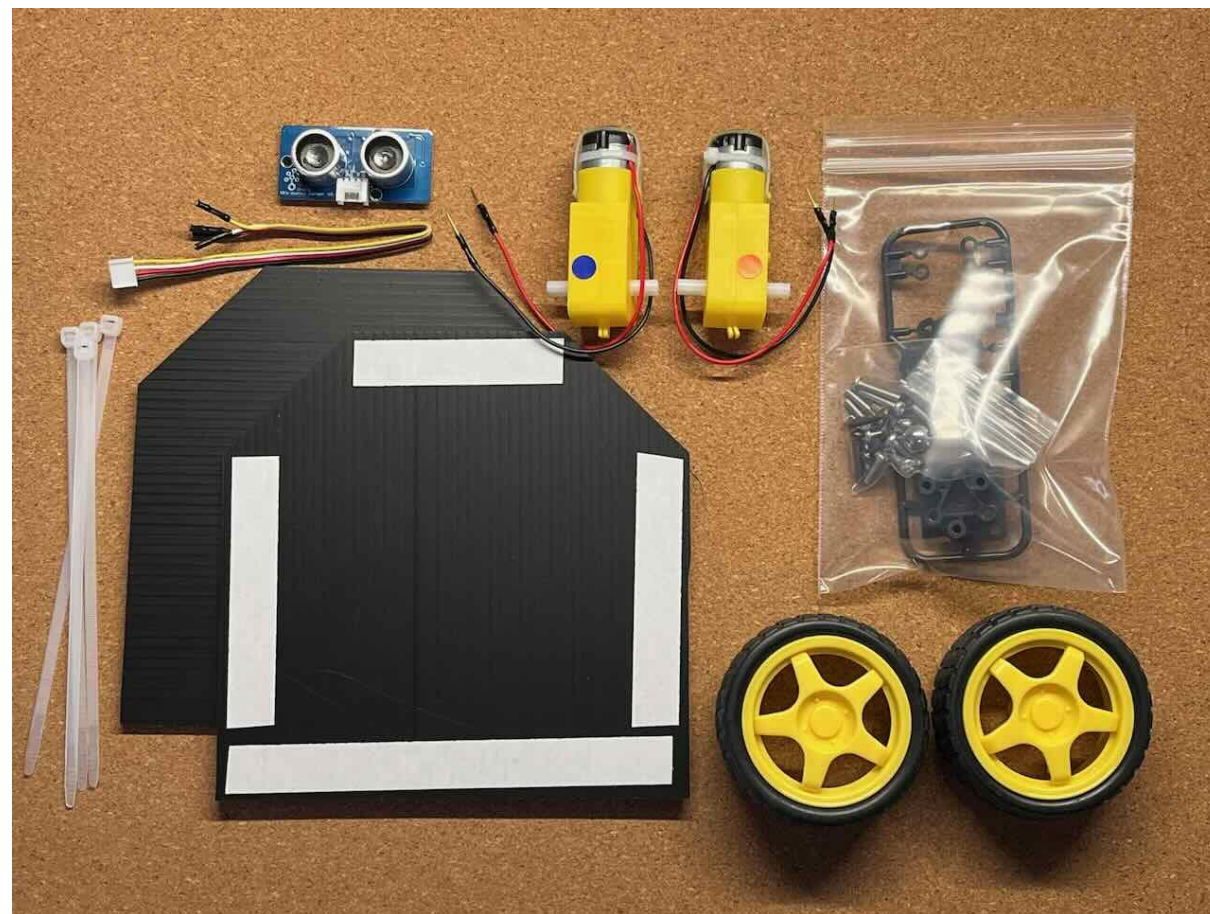
部品の確認 (マイクロビットセット)

部品	個数
マイクロビット	1
拡張ボード (KEYE STUDIO KS4033)	1
USBケーブル	1
単三x4電池ボックス	1
単三乾電池	4



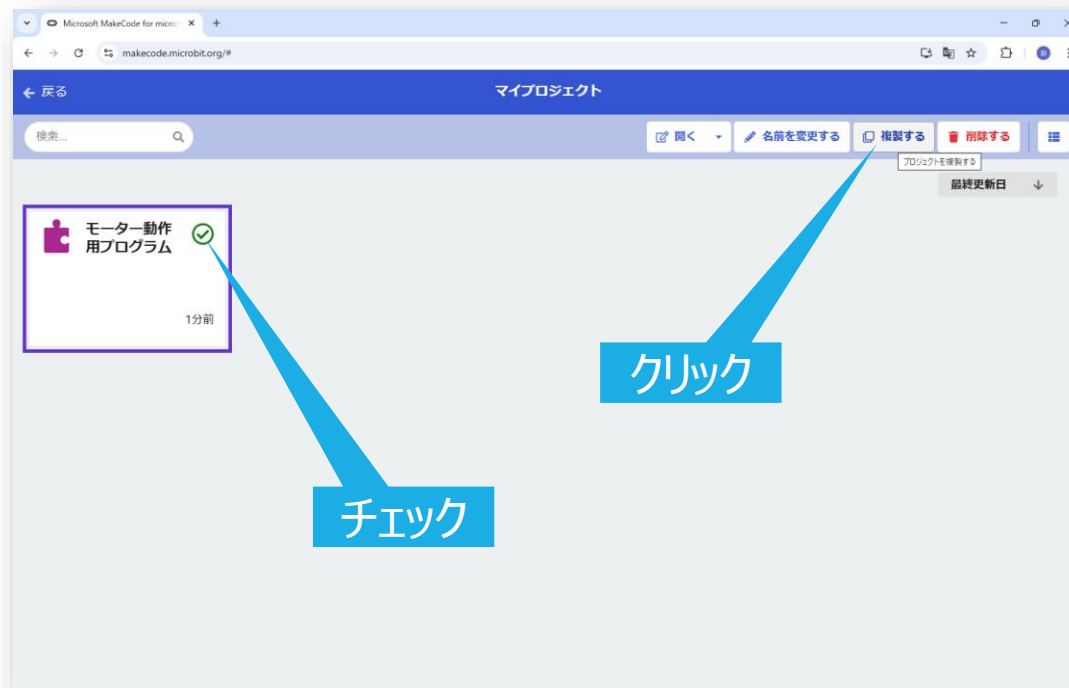
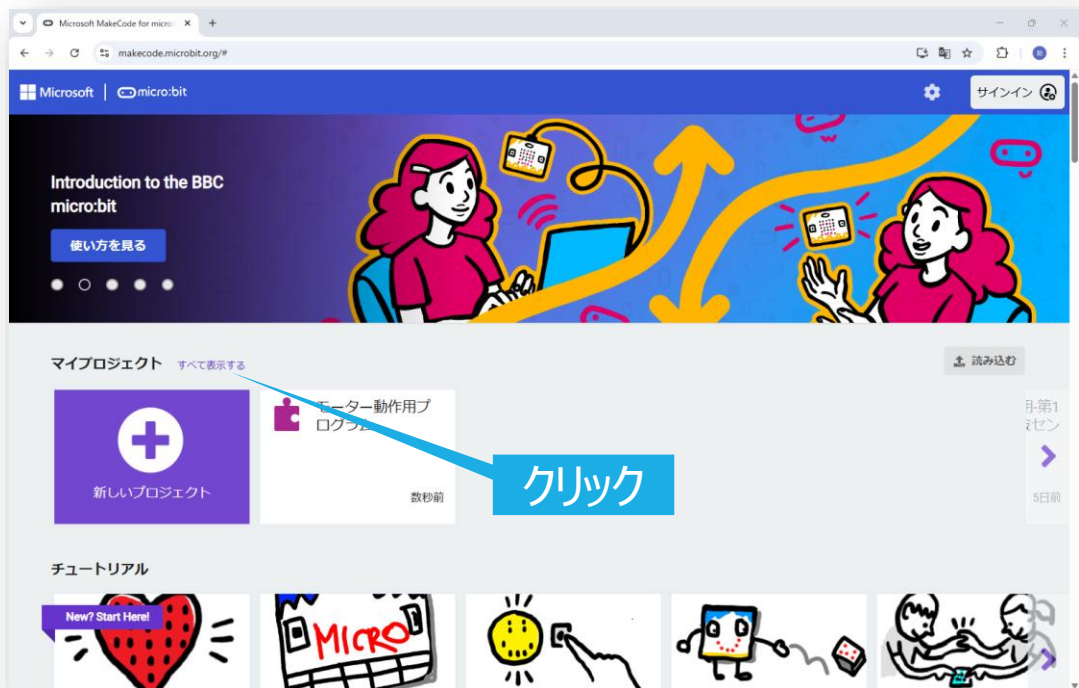
部品の確認（ロボカー）

部品	個数
モーター	2
タイヤ	2
超音波センサ	1
超音波センサ用ケーブル	1
プラダン	2
ボールキャスター	1
結束バンド	4



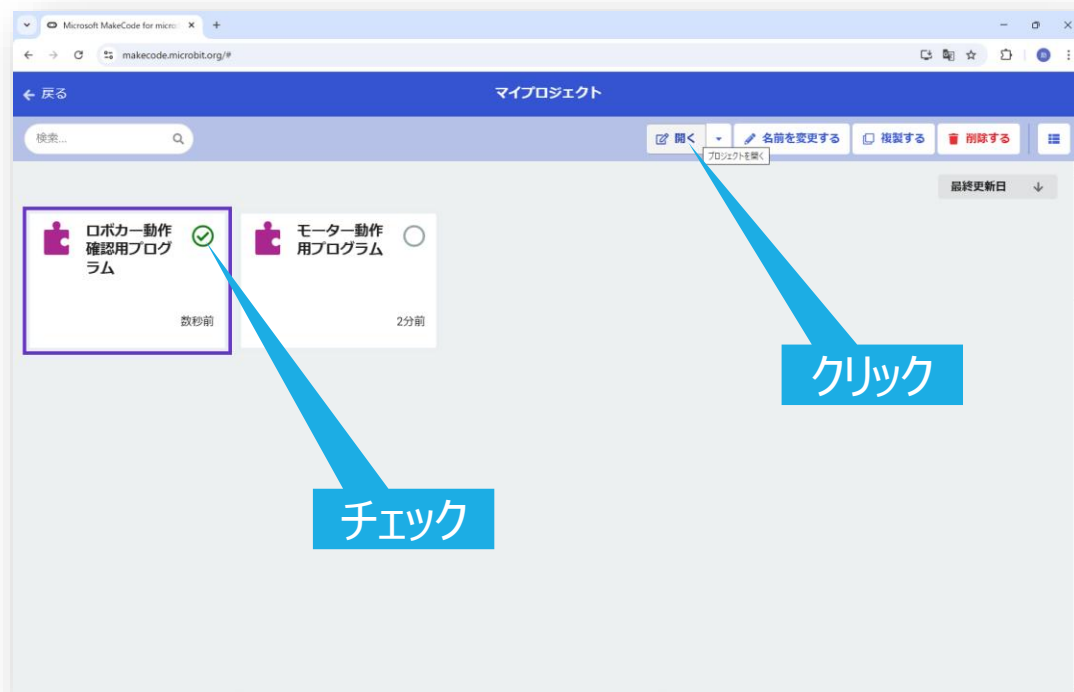
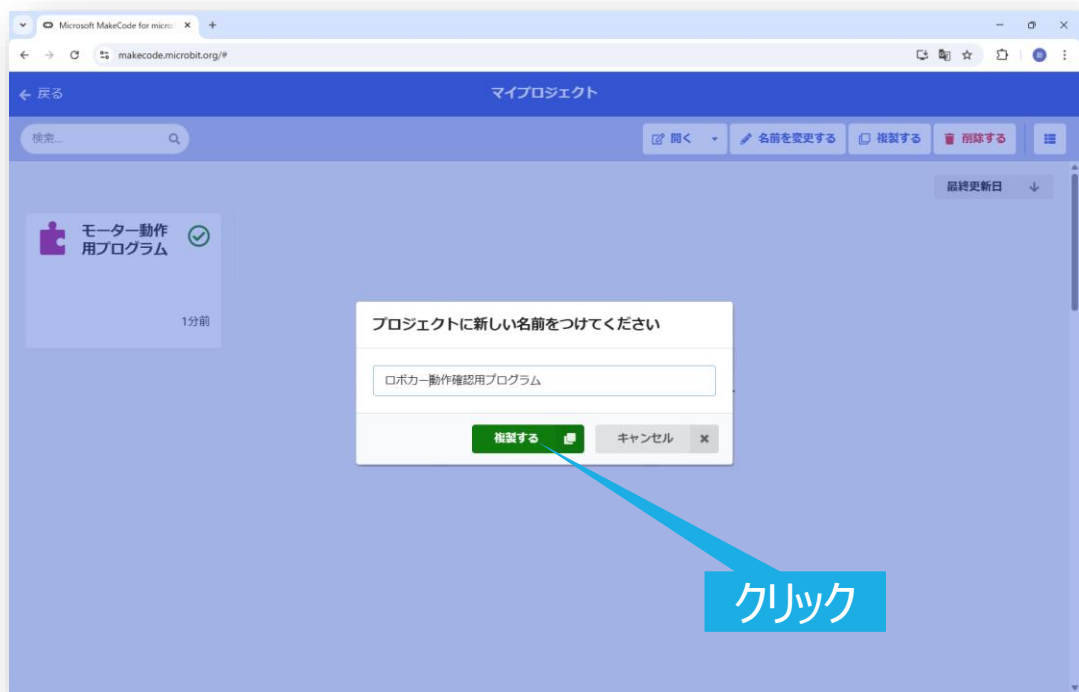
前回つくったプログラムのコピー

- 前回つくった関数を、今回も活用します。
- MakeCodeエディタをひらき「すべて表示する」をクリックします。
- 前回つくったモーターのプログラムにチェックをいれ、画面右上の「複製する」をクリックします。



前回つくったプログラムのコピー

- あたらしいなまえをつけて「複製する」をクリックします。
- プログラムがコピーされたので、それにチェックをいれて、「開く」をクリックします。



前回つくったプログラムの確認

- 前回、これらの関数をつくりました。
- これらの関数は今回もつかうので、消さずにそのままのこしておきます。

The image displays a collection of function blocks from a block-based programming environment, arranged in two rows. Each block is a blue rounded rectangle with a white header containing the function name and a small upward arrow icon. The body of each block contains red rounded rectangles representing code blocks.

Row 1:

- 右モーター正転 (Right Motor Forward):** Contains two red blocks: "アナログで出力する 端子 P12 値 256" and "アナログで出力する 端子 P13 値 0".
- 右モーター逆転 (Right Motor Reverse):** Contains two red blocks: "アナログで出力する 端子 P12 値 0" and "アナログで出力する 端子 P13 値 256".
- 右モーター停止 (Right Motor Stop):** Contains two red blocks: "アナログで出力する 端子 P12 値 0" and "アナログで出力する 端子 P13 値 0".
- 前進 (Forward):** Contains two red blocks: "呼び出し 右モーター正転" and "呼び出し 左モーター正転".
- 後進 (Reverse):** Contains two red blocks: "呼び出し 右モーター逆転" and "呼び出し 左モーター逆転".
- 停止 (Stop):** Contains two red blocks: "呼び出し 右モーター停止" and "呼び出し 左モーター停止".

Row 2:

- 左モーター正転 (Left Motor Forward):** Contains two red blocks: "アナログで出力する 端子 P15 値 256" and "アナログで出力する 端子 P16 値 0".
- 左モーター逆転 (Left Motor Reverse):** Contains two red blocks: "アナログで出力する 端子 P15 値 0" and "アナログで出力する 端子 P16 値 256".
- 左モーター停止 (Left Motor Stop):** Contains two red blocks: "アナログで出力する 端子 P15 値 0" and "アナログで出力する 端子 P16 値 0".
- 左折 (Left Turn):** Contains two red blocks: "呼び出し 右モーター正転" and "呼び出し 左モーター逆転".
- 右折 (Right Turn):** Contains two red blocks: "呼び出し 右モーター逆転" and "呼び出し 左モーター正転".

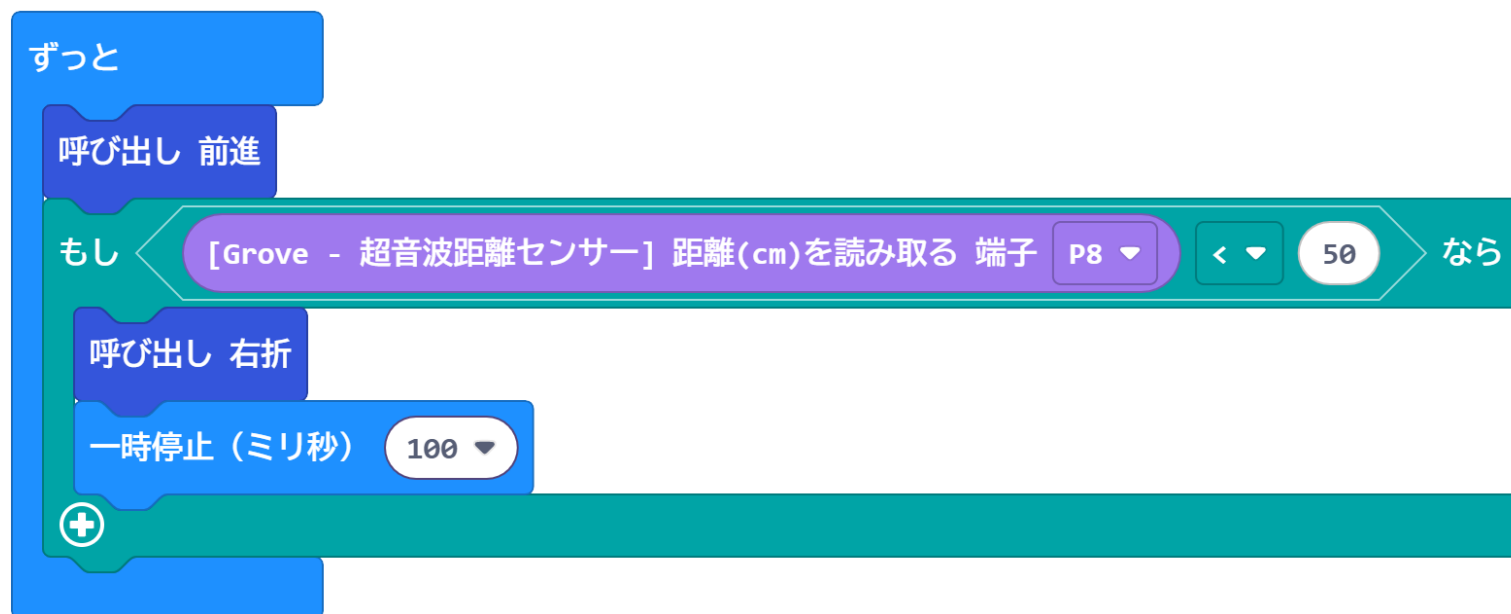
超音波センサをつかうための拡張機能をいれる

- ツールボックスで「拡張機能」をクリックします。
- 検索窓に「grove」と入力します。
- 表示された中から「grove」をクリックします。

The screenshot shows the Scratch IDE interface. On the left, the 'Grove' extension is selected in the '拡張機能' (Extensions) category. The main workspace displays a script for a '4-Digit' display. The script includes several 'strip' blocks for controlling the display. A red arrow points to the 'strip' block that is currently being added to the script area, which is labeled '[Grove - 超音波距離センサー] 距離(cm)を読み取る 端子 P0'.

プログラムをつくる

- 以下のプログラムをつくり、マイクロビットにかきこみます。
 - ふだんはロボカーは「前進」します。
 - もしも障害物との距離が50センチよりちかづくとき、0.1秒の間だけ「右折」します。
 - そのあとはプログラムの先頭にもどり、また「前進」します。



プログラムを改良する 1

- 障害物にちかづいたときにまがるむきがランダムにかわるようにします。
 - 「まがるむき」という変数をつくり、そこに「0」か「1」のランダムな数をいれます。
 - 「まがるむき」が「0」だったら「左折」し、「1」だったら「右折」するようにします。

```
ずっと
  呼び出し 前進
  もし < [Grove - 超音波距離センサー] 距離 (cm)を読み取る 端子 P8 < 50 なら
    変数 まがるむき を 0 から 1 までの乱数 にする
    もし まがるむき = 0 なら
      呼び出し 左折
    でなければ
      呼び出し 右折
    +
    一時停止 (ミリ秒) 100
  +
```

プログラムを改良する 2

- 障害物にちかづいたときにまがる時間がランダムにかわるようにします。
 - 「一時停止」の時間が、さきほどまでは「100」になっていましたが、ここを「100 から 2500 までの乱数」にかえます。
 - これでまがる時間が「0.1秒」から「0.1秒～2.5秒のランダムな時間」にかわります。
 - どのぐらいむきをかえるかが、そのときによってランダムにかわるようになります。

```
ずっと
  呼び出し 前進
  もし [Grove - 超音波距離センサー] 距離(cm)を読み取る 端子 P8 < 50 なら
    変数 まがるむき を 0 から 1 までの乱数 にする
    もし まがるむき = 0 なら
      呼び出し 左折
    でなければ
      呼び出し 右折
    一時停止 (ミリ秒) 100 から 2500 までの乱数
```

プログラムを改良する 3

- より確実にうごくようにするために、以下の処理をくわえます。
 - 障害物にちかづいたとき、むきをかえるまえにすこしだけ「後進」するようにします。
 - ことなる種類のうごきにかわるまえに、いったんロボカーを「停止」させます。

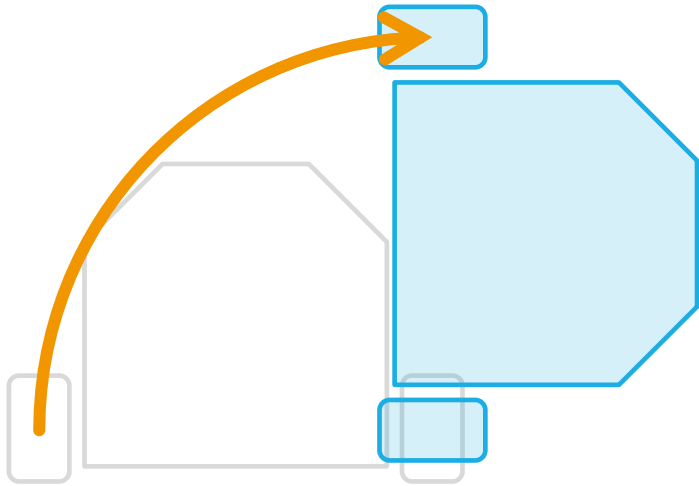
```
ずっと
  呼び出し 前進
  もし < [Grove - 超音波距離センサー] 距離(cm)を読み取る 端子 PB < 50 > なら
    呼び出し 停止
    一時停止 (ミリ秒) 100
    呼び出し 後進
    一時停止 (ミリ秒) 100
    変数 まがるむき を 0 から 1 までの乱数 にする
    もし まがるむき = 0 なら
      呼び出し 左折
      でなければ
        呼び出し 右折
    一時停止 (ミリ秒) 100 から 2500 までの乱数
    呼び出し 停止
    一時停止 (ミリ秒) 100
```

プログラムの調整

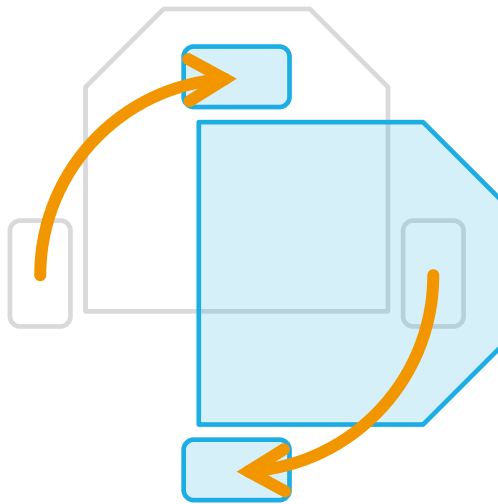
- モーターの性能のすこしのちがいや、ロボカーのくみたてかたのちがいにより、おなじプログラムでもロボカーのうごきはかわってきます。
- モーターのまわるスピードやうごかす時間を変えて、自分のロボカーがスムーズにうごくように調整しましょう。

【ノウハウ】むきのかえかた いろいろ

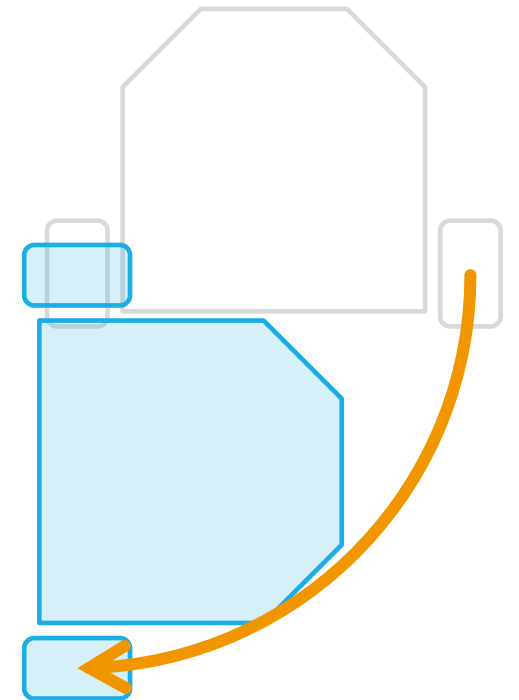
- ロボカーのむきのかえかたにもいろいろあります。



左モーター正転
右モーター停止



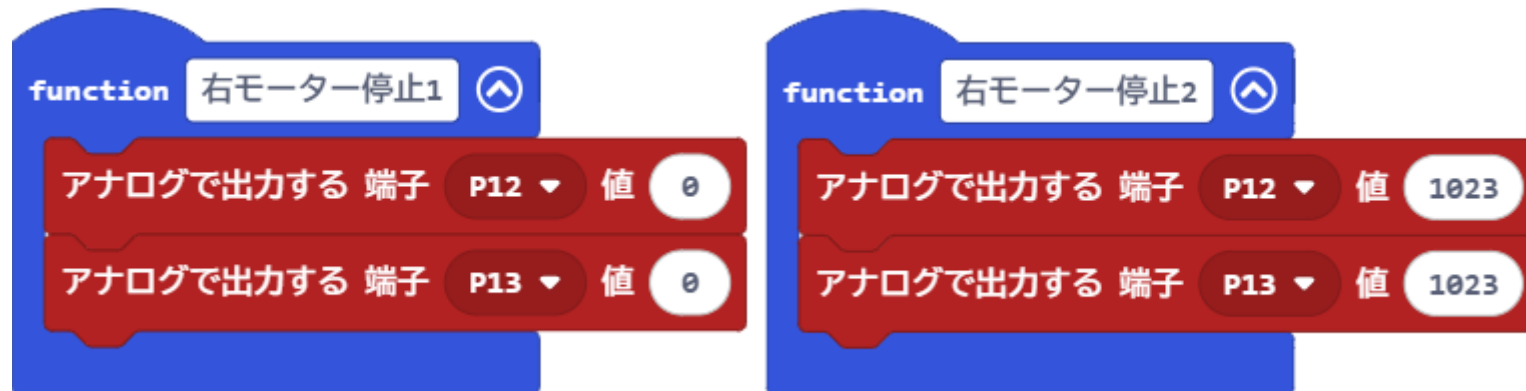
左モーター正転
右モーター逆転



左モーター停止
右モーター逆転

【ノウハウ】停止のさせかた いろいろ

- モーターを制御するふたつの端子をどちらも「0（0V）」にすると、モーターはしぜんにとまります。
- モーターを制御するふたつの端子をどちらも「1023（6V）」にすると、モーターはブレーキがかかります。



【ノウハウ】スピードのかえかた

- 「スピード」という変数をつくり、モーター用端子に「スピード」変数で値をあたえます。
- プログラム内で「スピード」変数の値をかえると、モーター用端子にあたえる値もかわります。

function 右モーター正転
アナログで出力する 端子 P12 値 **スピード**
アナログで出力する 端子 P13 値 0

function 右モーター逆転
アナログで出力する 端子 P12 値 0
アナログで出力する 端子 P13 値 **スピード**

function 右モーター停止
アナログで出力する 端子 P12 値 0
アナログで出力する 端子 P13 値 0

function 左モーター正転
アナログで出力する 端子 P15 値 **スピード**
アナログで出力する 端子 P16 値 0

function 左モーター逆転
アナログで出力する 端子 P15 値 0
アナログで出力する 端子 P16 値 **スピード**

function 左モーター停止
アナログで出力する 端子 P15 値 0
アナログで出力する 端子 P16 値 0

ずっと
変数 **スピード** を 500 にする ← 変数の値をかえる
呼び出し 前進
もし [Grove - 超音波距離センサー] 距離(cm)を読み取る 端子 P8 < 50 なら
変数 **スピード** を 300 にする ← 変数の値をかえる
呼び出し 停止