

マイクロビット ロボカーの応用例

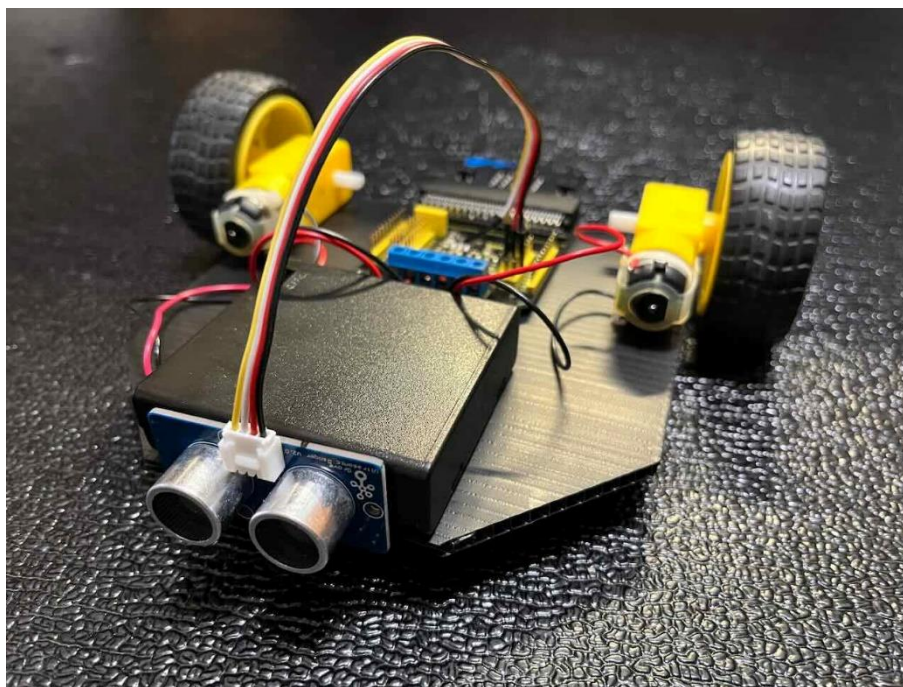
マイクロビットロボカーは、プログラムの内容によってさまざまな動きをさせることができます。
ここでは、ロボカーにほかの部品をとりつけた作品の例を紹介します。

ゆめほたる環境科学技術クラブ

ぶつからない車

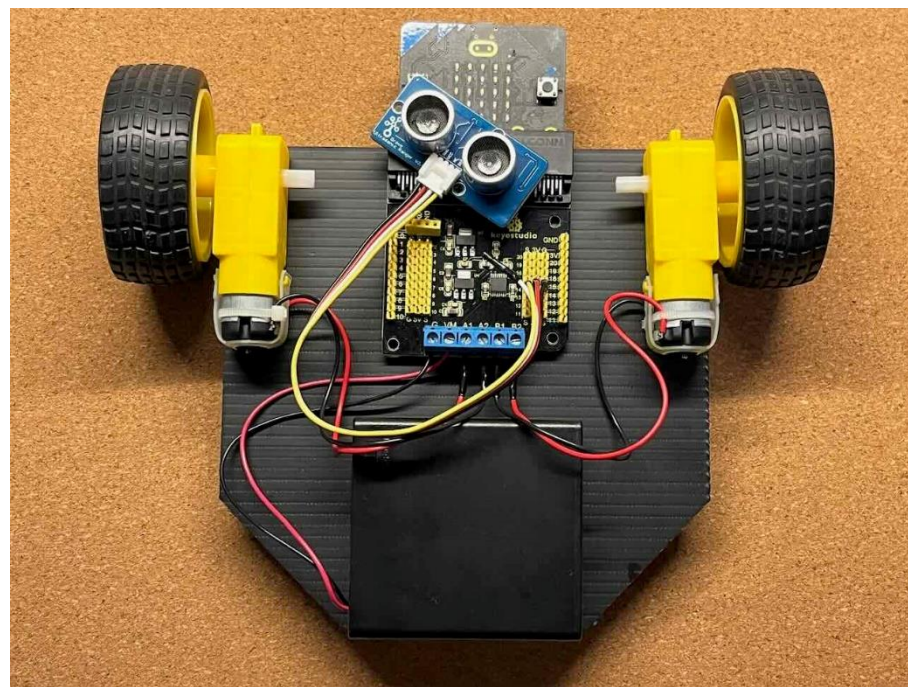
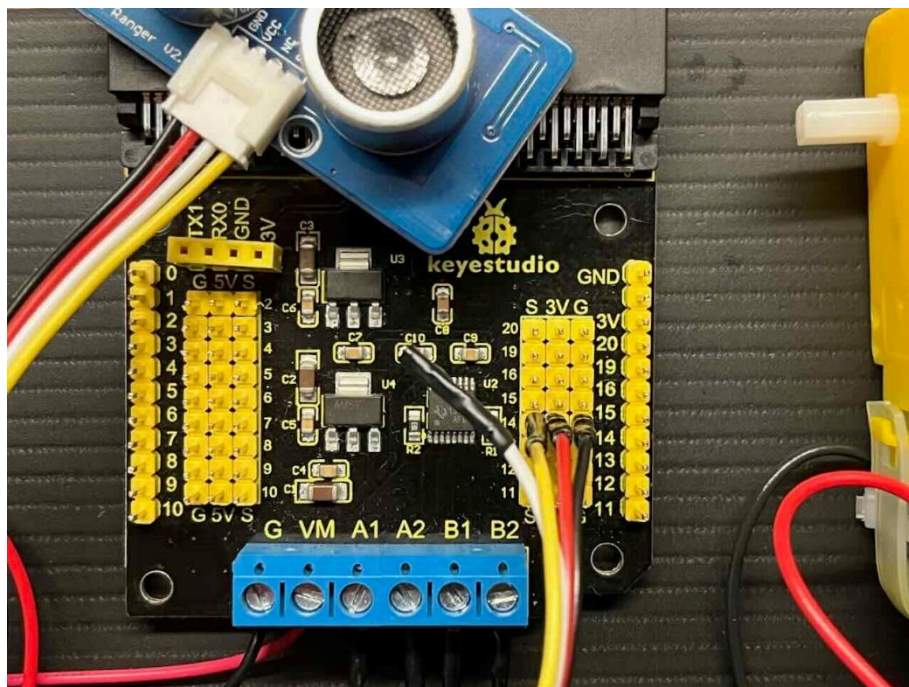
(くわしくはこちら : <https://sample.msr-r.net/microbit-robotcar-ultrasonic/>)

- ロボカーの前面に「超音波センサ（前にある障害物との距離をはかる部品）」をとりつけます。
- 障害物にちかづくときロボカーの向きをかえることで、障害物にぶつからずに動きつづけることができます。



ぶつからない車（くみたて）

- 超音波センサの「GND (黒)」「VCC (赤)」「SIG (黄)」を、拡張ボードの「G」「3V」「14」につなぎます。
- 超音波センサをロボカーの前面にはりつけます。



ぶつからない車（プログラム）

- ロボカーを動かすための11個の関数は「ロボカーきほんどうさ」と同じものです。
- 以下のプログラムをマイクロビットにかきこみ、スイッチをオンにすると、ロボカーがレンバのように動きつづけます。

The image shows a Scratch-style block diagram for a program. It consists of two main sections: '最初だけ' (Only at the start) and 'ずっと' (Forever loop).

- 最初だけ (Only at the start):** Contains a single '呼び出し 前進' (Call Forward) block.
- ずっと (Forever loop):** Contains a sequence of blocks:
 - A 'もし' (If) block with the condition: '[Grove - 超音波距離センサー] 距離(cm)を読み取る 端子 P14 < 50 なら' (If distance from Grove ultrasonic sensor on pin P14 is less than 50 cm).
 - '呼び出し 後進' (Call Reverse)
 - '一時停止(ミリ秒) 100' (Wait 100 ms)
 - '呼び出し 右回り' (Call Turn Right)
 - '一時停止(ミリ秒) 100' (Wait 100 ms)
 - '呼び出し 前進' (Call Forward)

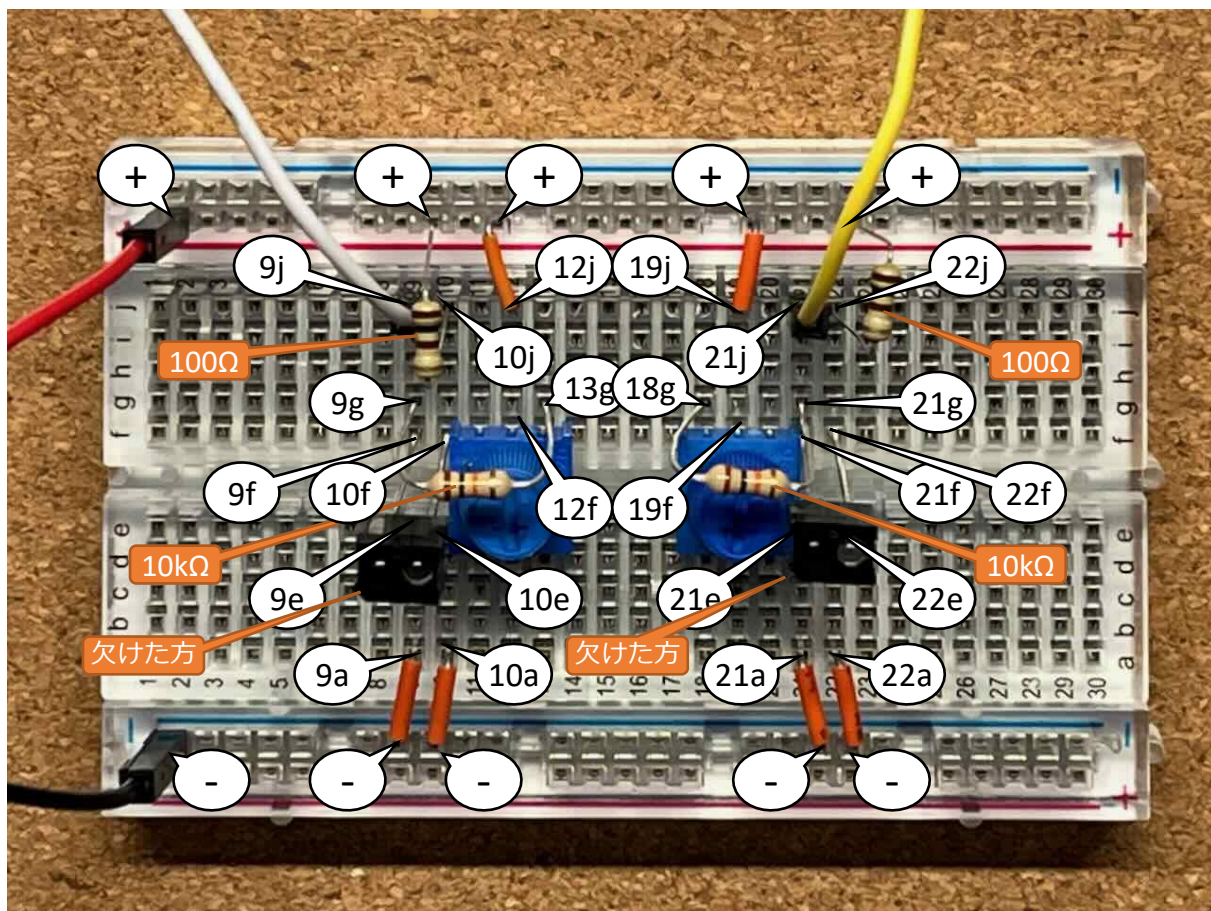
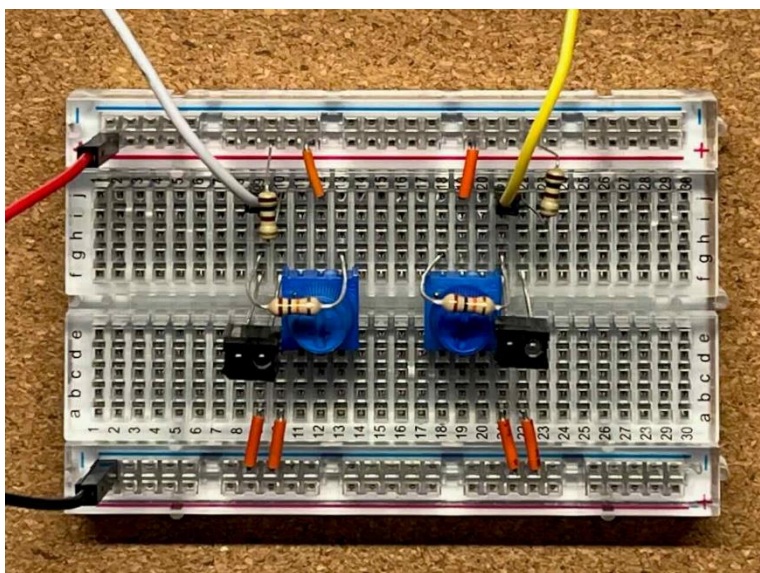
ライントレースカー

(くわしくはこちら : <https://sample.msr-r.net/microbit-linetracecar/>)

- 白い紙にひかれた黒い線にそってロボカーが動きます。
- 「フォトリフレクタ」という部品が紙の白と黒を検知し、それによってロボカーの向きをかえます。
- つかう部品は以下のとおりです。
 - ブレッドボード 1枚
 - フォトリフレクタ 2個
 - 10k Ω 半固定抵抗 2個
 - 10k Ω 抵抗 2個
 - 100 Ω 抵抗 2個
 - みじかいジャンパーワイヤ 6個
 - ジャンパーワイヤ (オス~メス) 4本

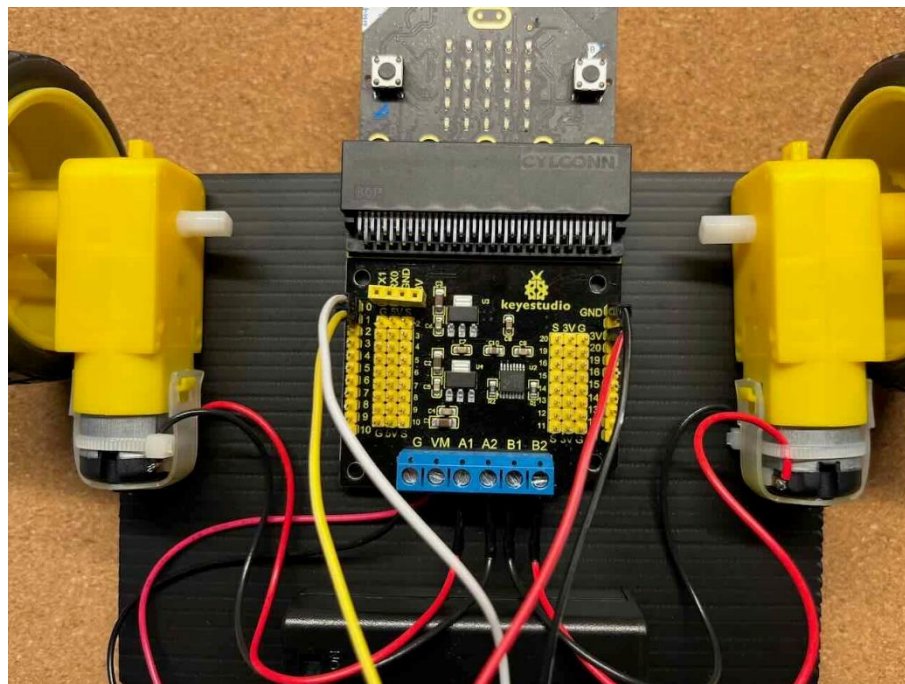
ライントレースカー（くみたて）

- ブレッドボードで回路をつくります。



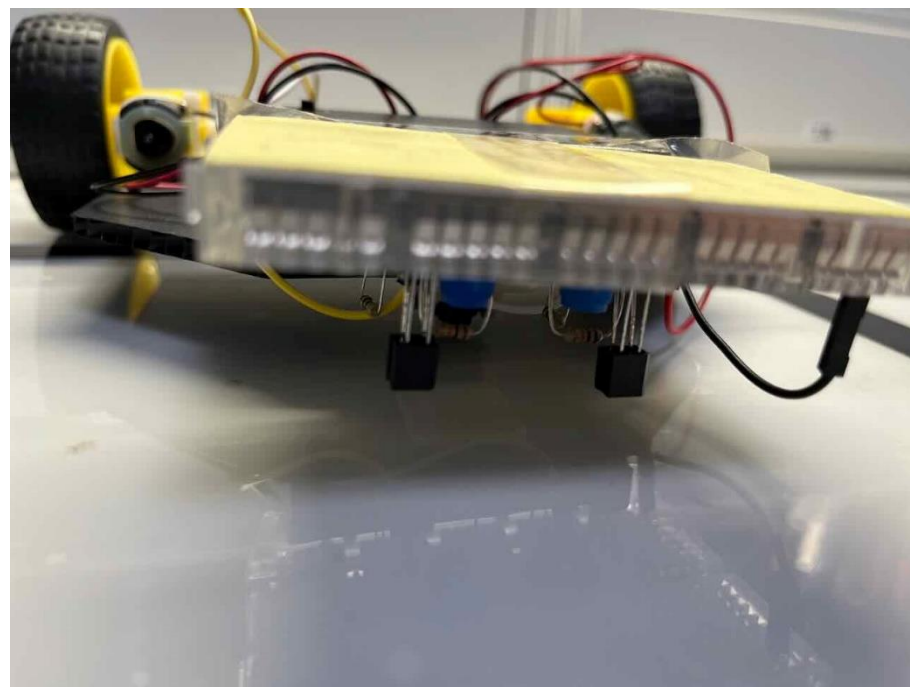
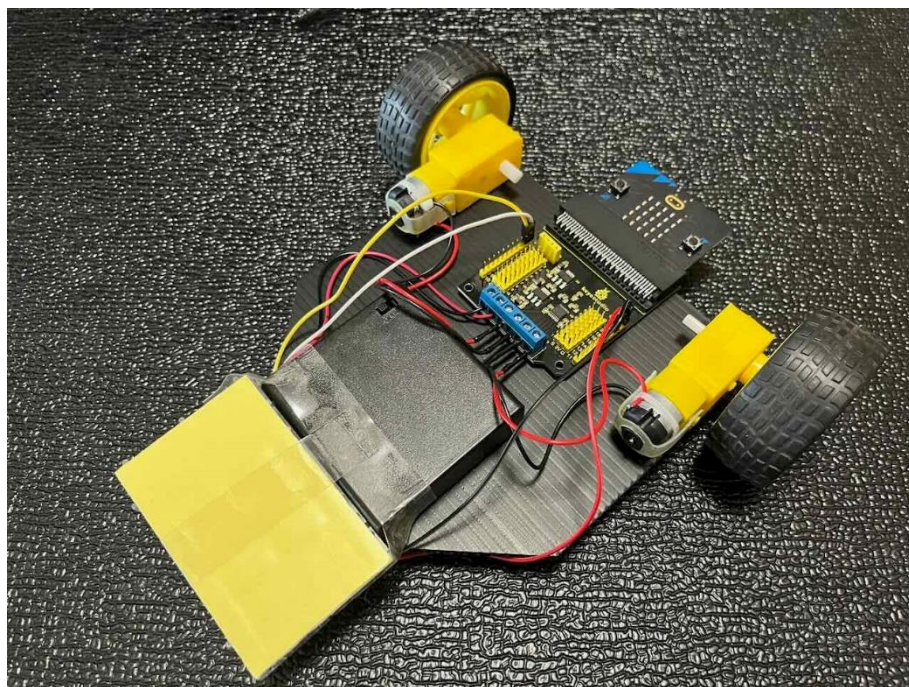
ライトレースカー（くみたて）

- ブレッドボードのジャンパーワイヤ「黒」「赤」「白」「黄」を拡張ボードの「GND」「3V」「0」「1」につなぎます。



ライントレースカー（くみたて）

- ブレッドボードを下向きにして、ロボカーの前面にはりつけます。
- フォトリフレクタと地面のあいだが数ミリになるようにたかさを調整します。



ライトレースカー（プログラム）

- ロボカーを動かすための11個の関数のうち5個から、画面表示に関するブロックをけします。
 - ✓画面表示させると、処理がすこしおそくなり、うまくライトレースできませんでした。



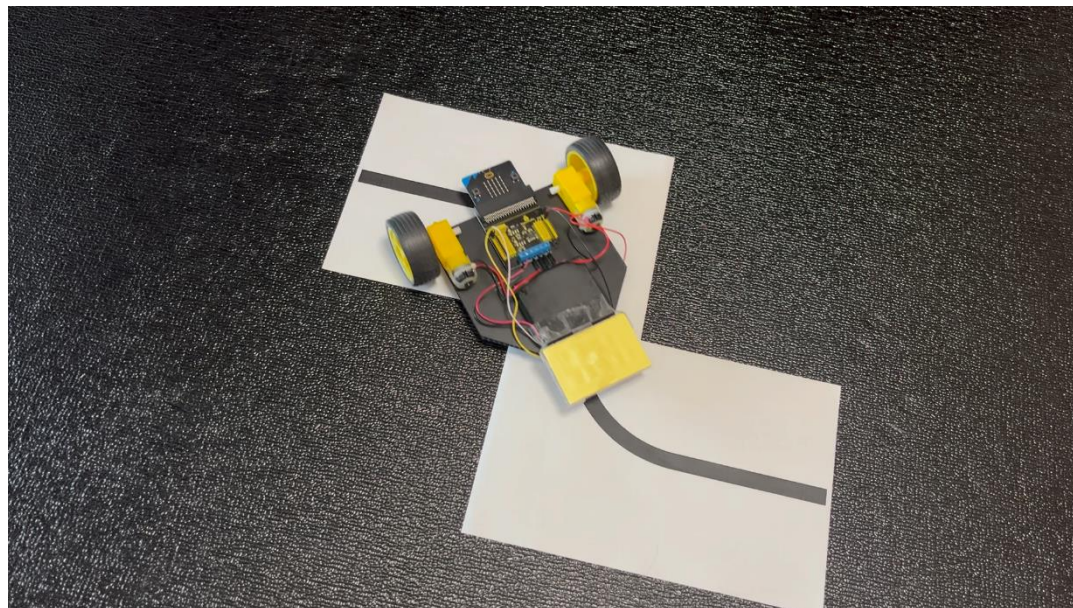
ライントレースカー (プログラム)

- メインプログラムは以下のとおりです。

The diagram shows a Scratch-style block diagram for a line tracking program. It starts with a 'ずっと' (Forever) loop block. Inside the loop, there is a 'もし' (If) block with two conditions: 'アナログ値を読み取る 端子 P0 > 300' and 'かつ' (and) 'アナログ値を読み取る 端子 P1 > 300'. If both conditions are met, it calls '呼び出し 後進' (Call Backward). If not, it checks 'でなければもし' (If not) 'アナログ値を読み取る 端子 P0 > 300'. If true, it calls '呼び出し 左回り' (Call Turn Left). If not, it checks 'でなければもし' (If not) 'アナログ値を読み取る 端子 P1 > 300'. If true, it calls '呼び出し 右回り' (Call Turn Right). If not, it calls '呼び出し 前進' (Call Forward). Finally, there is a '一時停止(ミリ秒) 10' (Pause 10 milliseconds) block.

ライントレースカー（動作確認）

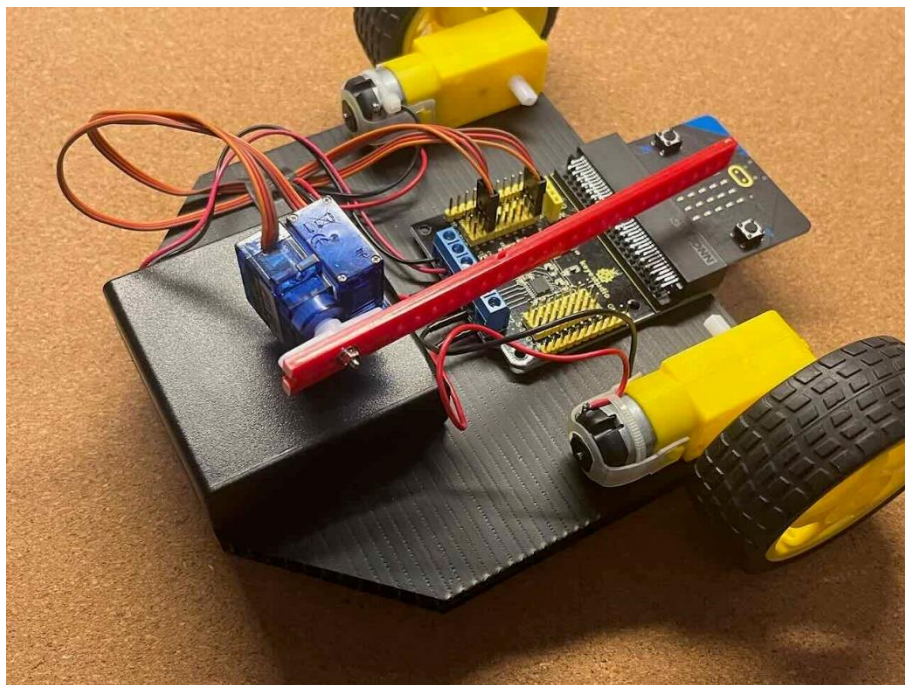
- プログラムをマイクロビットにかきこみ、スイッチをオンにすると、ロボカーが黒い線にそって動きます。
- うまく動かない場合は、「半固定抵抗のツマミをまわす」「ブレッドボードの高さをかえる」「プログラム内の300という値をかえる」などの調整が必要です。



クレーン車

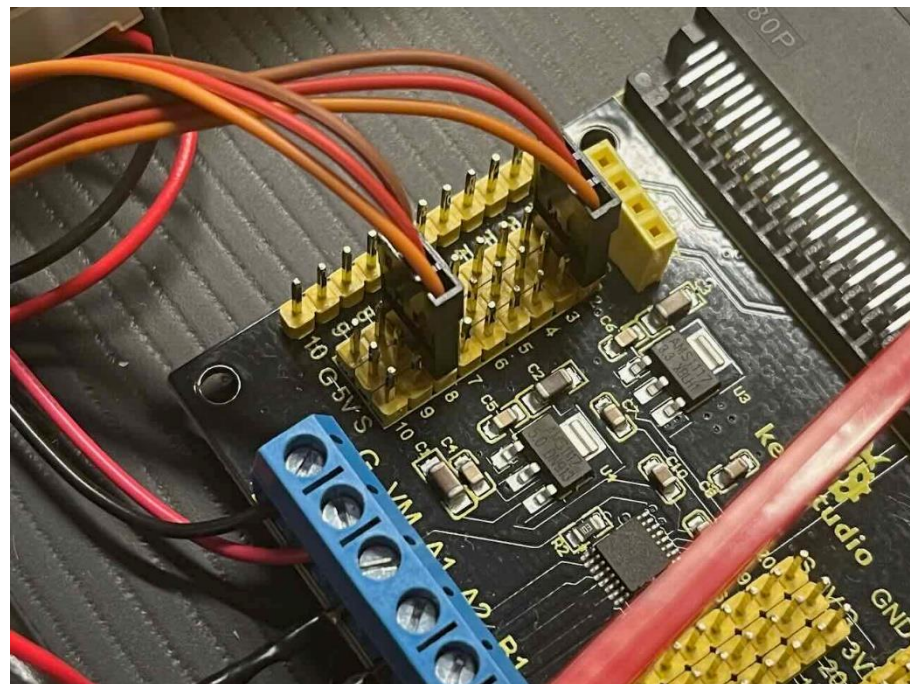
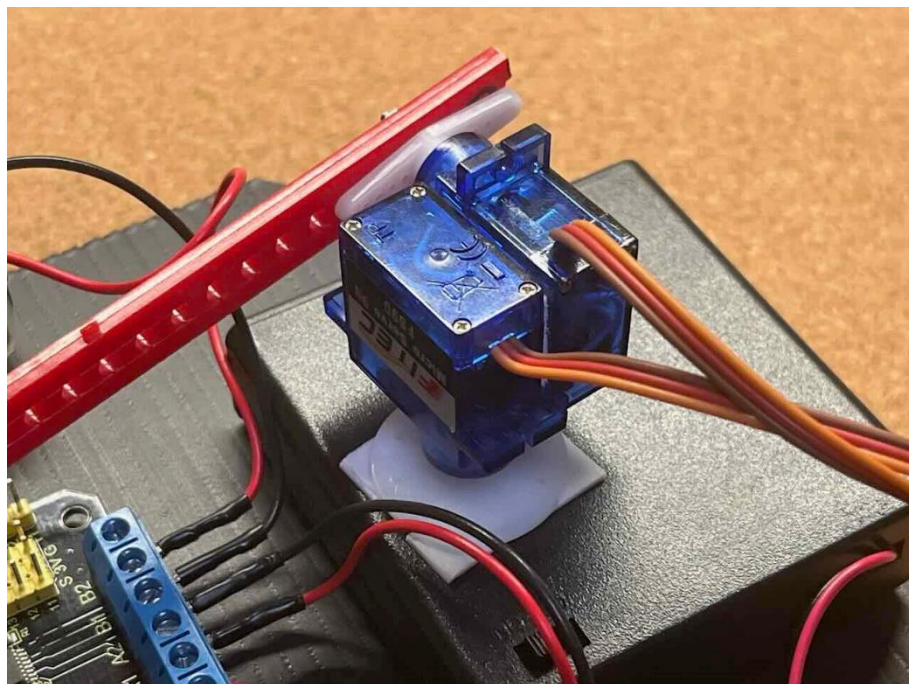
(くわしくはこちら : <https://sample.msr-r.net/microbit-cranetruck/>)

- ロボカーとサーボモーター2コをくみあわせてクレーン車をつくります。
- サーボモーターでアームの回転や上げ下げができます。



クレーン車（くみたて）

- 2コのサーボモーターを90度ずらしてはりあわせ、一方のじくをロボカー、もう一方をアームにはりつけます。
- 2コのサーボモーターのコネクタを、それぞれ拡張ボードの「2」「8」の場所にさしこみます。



クレーン車（プログラム）

- サーボモーターをうごかす関数は以下のとおりです。



関数 **サーボ2を動かす** 数値 数値2

もし **数値2** \geq **数値** ならば

変数 **カウンター** を0~ **数値2** - **数値** に変えてくりかえす

サーボ 出力する 端子 **P2** 角度 **数値** + **カウンター**

一時停止(ミリ秒) **10**

でなければ

変数 **カウンター** を0~ **数値** - **数値2** に変えてくりかえす

サーボ 出力する 端子 **P2** 角度 **数値** - **カウンター**

一時停止(ミリ秒) **10**



関数 **サーボ8を動かす** 数値 数値2

もし **数値2** \geq **数値** ならば

変数 **カウンター** を0~ **数値2** - **数値** に変えてくりかえす

サーボ 出力する 端子 **P8(出力のみ)** 角度 **数値** + **カウンター**

一時停止(ミリ秒) **10**

でなければ

変数 **カウンター** を0~ **数値** - **数値2** に変えてくりかえす

サーボ 出力する 端子 **P8(出力のみ)** 角度 **数値** - **カウンター**

一時停止(ミリ秒) **10**

クレーン車（プログラム）

- メインプログラムは以下のとおりです。
- Aボタンをおすと「前進⇒右回り⇒アーム上げ⇒アーム回転⇒アーム下げ⇒アーム上げ⇒アーム回転⇒アーム下げ⇒左回り⇒後進」を行います。

