

ポコロボスタジオ 操作ガイド

文書バージョン: v1.9 / 発行日: 2026年10月

製造元: 株式会社ロボットスポーツゲームズ

目次

1. このガイドについて
2. スタジオを開く
3. 画面の見方
4. 基本操作
5. 各ブロックの説明
6. よくある質問
7. 用語集
8. 付録A: ゲームパッドボタン対応表
9. 付録B: ブロッカー一覧早見表
10. 付録C: 作ってみよう（例題）

クイックスタート：最初にやる3つ

本体のセットアップが完了していることを確認してから始めてください。（まだの場合は「ポコロボ 本体セットアップ・安全ガイド」を先に読んでください）

ステップ1：スタジオが開いているか確認する

1. パソコンで「ポコロボ コネクト」が動いていることを確認します（通常は自動で起動しています。入っていない場合は「ポコロボ 本体セットアップ・安全ガイド」を参照）
2. Chrome のアドレス欄に `https://studio.robotic-sports.com` と入力してスタジオを開きます
3. 画面右側の操作パネルに「**接続中**」と表示されていれば準備完了です
 - 「未接続」や画面上部に黄色いお知らせが出る場合 → §2「画面の上に黄色いお知らせが出たとき」、または「ポコロボ 本体セットアップ・安全ガイド」 §8を参照してください

ステップ2：LEDを光らせてみる（約3分）

1. 左のブロック置き場の「繰り返す・待つ」から「ずっと繰り返す」ブロック（緑）をドラッグしてワークスペースに置きます
2. その中に、一番上の「動かす」にある「LED」ブロック（青）を入れます
3. LED ブロックの赤・緑・青の数字をクリックして好きな値（0～100）に変更します
4. 操作パネルの「▶ 実行」をクリックします
5. ポコロボのLEDが光れば成功です

「▶ 実行」を押しても光らない場合 → §6「よくある質問」を参照してください

ステップ3：次にやること

- DCモータを動かす → 付録C「例題② DCモータを動かそう」
- ゲームパッドで操縦する → 付録C「例題③ ゲームパッドで操縦しよう」
- ブロックの使い方を調べる → 付録B「ブロッカー一覧早見表」

1. このガイドについて

このガイドで分かること

このガイドは、ポコロボスタジオ（Webブラウザで動くプログラミング画面）の操作方法を説明します。

次のことが分かります：

- スタジオをブラウザで開く手順
- 画面の各部の名前と役割
- ブロックを置いてプログラムを組み立てる基本操作
- 各ブロックの意味・使い方・入力できる値の範囲
- LEDを光らせる・DCモータを動かす・ゲームパッドで操縦するなどの例題

前提：本体の準備ができていること

このガイドを使う前に、以下の準備が完了していることを確認してください。

- ☐ 電源が用意されている（スタンダードは電池ケースに単4形電池×2本。ミニはパワー用電源かUSB給電）
- ☐ 必要であればパワー用電源も接続されている（DCモータ・サーボを使う場合）
- ☐ DCモータやサーボなど使う周辺機器が接続されている

上記の準備方法は「ポコロボ 本体セットアップ・安全ガイド」をご覧ください。

分からない単語が出てきたら、巻末の「7.用語集」を参照してください。

このガイドの読み方

- **初めて使う方**：「2. スタジオを開く」から順番に読んでください
- **特定の操作を知りたい方**：「5. 各ブロックの説明」または「6. よくある質問」を直接参照してください
- **例題をやってみたい方**：「付録C: 作ってみよう」から始めてください

スタンダードとミニの違い

ポコロボには **スタンダード** と **ミニ** の2種類があります。スタジオの操作は共通ですが、使える機能が一部異なります。機能の違いの一覧は「ポコロボ 本体セットアップ・安全ガイド」§1「スタンダードとミニの違い」を参照してください。

ミニをつないでいるときは、ミニに無いブザーとエンコーダのブロックはブロック置き場に出ません。スタンダード用に作ったプログラムを読み込むなどしてミニでブザーやエンコーダのブロックを使うと、実行中にエラーが表示されてプログラムが止まります。また、DCモータの番号はミニでは0・1のみが使えます。

2. スタジオを開く

必要なもの

- ポコロボ本体（電源ON済み）
- 通信対応のUSB Type-A to Type-Cケーブル（充電専用ケーブルは使えません。ケーブルの選び方は「ポコロボ 本体セットアップ・安全ガイド」の「使うもの」参照）
- インターネットにつながるパソコン（**Chrome 推奨**）
- パソコン用アプリ「ポコロボ コネクト」（導入方法は「ポコロボ 本体セットアップ・安全ガイド」参照）

スタジオを開く手順

1. ポコロボの電源スイッチを **ON** にします

2. **通信対応の** USB Type-A to Type-Cケーブルでポコロボとパソコンをつなぎます（充電専用ケーブルでは接続できません）。**インターネットで使うときもこのケーブルは必要です**
3. パソコンでChromeブラウザを開きます
4. アドレス欄に `https://studio.robotic-sports.com` と入力して **Enterキー** を押します
5. 初めて開いたときは「ポコロボと接続するには、ブラウザの許可が必要です」と表示されます。「**接続する**」を押し、画面の左上に出る Chrome の確認で「**許可する**」を押します
6. スタジオの画面が表示されたら完了です

Chrome の確認で「ブロック」を押してしまうと「接続が「ブロック」されています」と表示されます。アドレスバー左のアイコンを押して「ローカルネットワーク」を許可に変え、「再読み込み」を押すと接続できます。

初めてつないだとき、WindowsパソコンはUSBネットワークドライバを自動でインストールします。少し待ってからアクセスしてください。

画面の上に黄色いお知らせが出たとき

スタジオがポコロボを見つけられないとき、画面の上部に黄色いお知らせが表示されます。表示される文ごとの対処は「ポコロボ 本体セットアップ・安全ガイド」§8「困ったときは」を参照してください。

インターネットに接続できないとき

ポコロボは本体の中にもスタジオを持っています。インターネットがない場所では、USB Type-A to Type-Cケーブルでつないだ状態で Chrome のアドレス欄に `http://192.168.20.1` と入力すると、同じスタジオが開きます。

- この方法では「ポコロボ コネクト」もインターネットも不要です（USB Type-A to Type-Cケーブルは必要です）
- アドレスは `https://` ではなく `http://` で始まることに注意してください

スタジオが開かない場合の確認事項

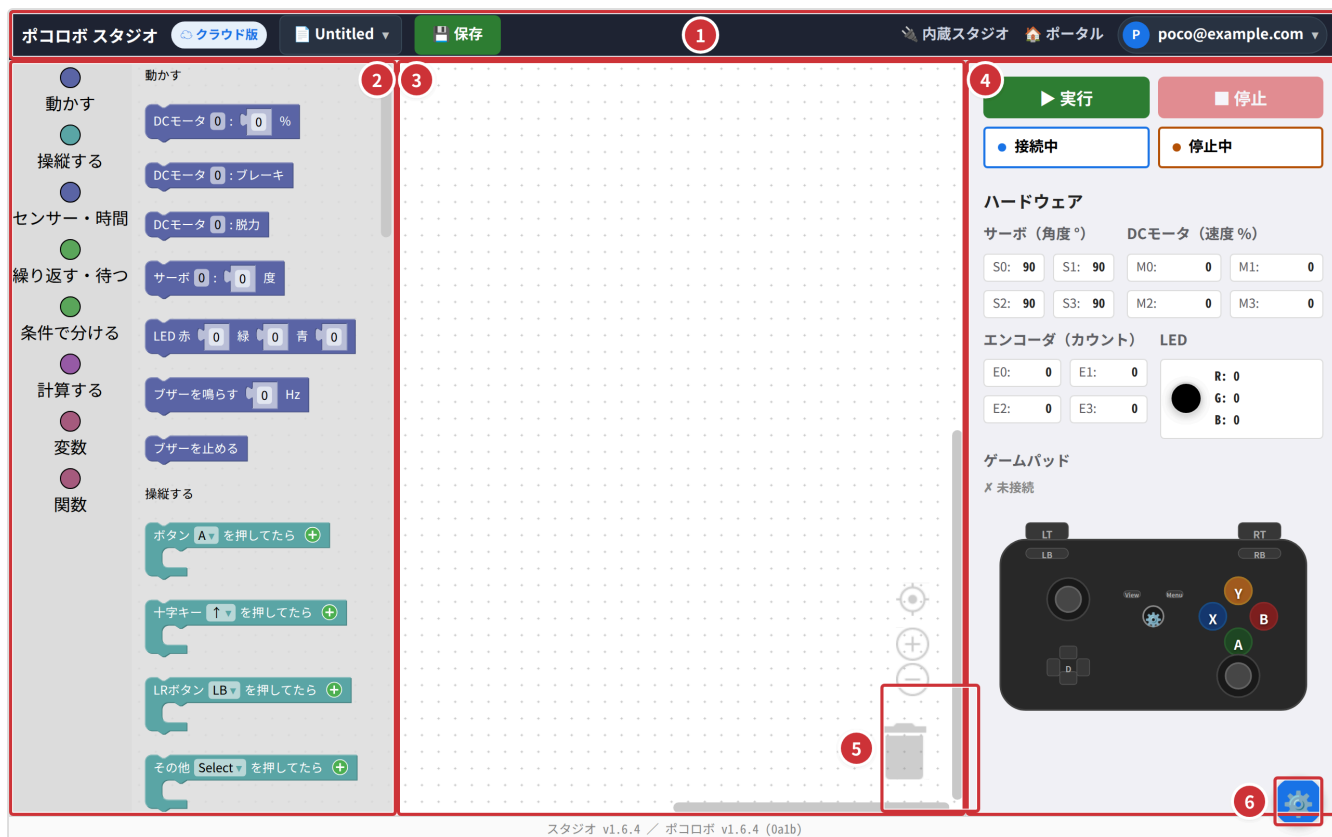
本体や接続まわりのトラブルは「ポコロボ 本体セットアップ・安全ガイド」の「困ったときは」を参照してください。

主な原因として次が挙げられます：

- ポコロボの電源が入っていない
- USB Type-A to Type-Cケーブルがしっかり刺さっていない
- 「ポコロボ コネクト」が起動していない（上の「画面の上に黄色いお知らせが出たとき」を参照）
- アドレスが間違っている（`https://studio.robotic-sports.com` を確認。内蔵スタジオの場合は `https://` ではなく `http://192.168.20.1`）
- Chrome 以外のブラウザを使っている（Chrome 推奨。他のブラウザは動作確認していません）

3. 画面の見方

スタジオを開くと、次のような画面が表示されます。



画面は大きく6つのエリアに分かれています。

番号	エリア名	説明箇所
①	ヘッダ	→「① ヘッダ（画面の上）」
②	ブロック置き場	→「② ブロック置き場（左側）」
③	ワークスペース	→「③ ワークスペース（中央）」
④	操作パネル	→「④ 操作パネル（右側）」
⑤	ゴミ箱	→「⑤ ゴミ箱」
⑥	設定ボタン	→「⑥ 設定ボタン」

① ヘッダ（画面の上）



プログラムに名前を付けて保存したり、アカウントを確認したりする場所です。

表示	意味
クラウド版	インターネット経由で開いているスタジオであることを示します（ http://192.168.20.1 で開いたときは「内蔵スタジオ」と表示されます）
プログラム名▼	開いているプログラムの名前です。クリックすると新規・開く・保存・書き出しのメニューが出ます
●未保存	プログラム名の横に出ます。保存していない変更があるという印です
保存	開いているプログラムを保存します（→4節「プログラムを保存する」）
内蔵スタジオ	本体の中のスタジオ（ http://192.168.20.1 ）へ移動します。インターネットが無い場所で使います
ポータル	ロボスポのポータルサイトへ移動します
アカウント	ログイン中のメールアドレスです。クリックするとアカウント画面とログアウトが出ます

表示	意味
ログイン / 新規登録	ログインしていないときに「アカウント」の代わりに出ます。押すとログインとアカウント作成の画面へ移動します

② ブロック置き場（左側）

使えるブロックがここに並んでいます。ここからドラッグしてワークスペースに持っていきます。

ブロックは次の**8つの島**（目的ごとのまとまり）に分かれ、上から順に1本の一覧に並んでいます。左端の列が島の**目次**で、島の名前を押すと一覧がその島の位置へ移動します。一覧を上下にスクロールしても全ブロックを見ることができます。

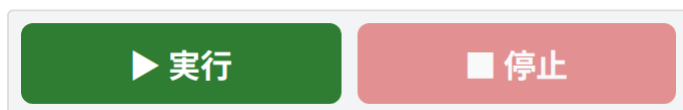
順番	島の名前	主な内容
①	動かす	DCモータ・サーボ・LED・ブザー
②	操縦する	ゲームパッドのボタン・十字キー・スティック
③	センサー・時間	エンコーダ・プログラム開始からの時間
④	繰り返す・待つ	繰り返し・待つ
⑤	条件で分ける	条件分岐・比較・AND/OR/NOT
⑥	計算する	四則演算・三角関数・平方根・絶対値・べき乗
⑦	変数	変数の作成・設定・読み出し
⑧	関数	関数の作成・実行・結果・引数

「まず動かす」ところから始められるように、「動かす」の島が一番上にあります。

③ ワークスペース（中央）

ブロックを配置してプログラムを組み立てる場所です。ブロックを上から下へつなげた順番に実行されます。マウスホイールで拡大縮小できます。ワークスペースの空いている部分をドラッグすると表示位置を移動できます。

④ 操作パネル（右側）



ボタン	機能
▶ 実行	プログラムをポコロボに送って実行する
■ 停止	実行中のプログラムを停止する

実行中はブロックの編集ができなくなります。編集したい場合は「■ 停止」を押してください。

接続状態の表示：ボタンのすぐ下に、ポコロボとの接続状態とプログラムの状態が並びます。**実行前に必ず確認**してください。



左側の表示	意味	対処
接続中	ポコロボと通信できています	実行できます

左側の表示	意味	対処
未接続	ポコロボと通信できていません	USB Type-A to Type-Cケーブルと電源を確認（「ポコロボ 本体セットアップ・安全ガイド」 § 8 「困ったときは」を参照）

右側にはプログラムの状態が出ます。停止しているときは**停止中**、動いているときは**実行中**、エラーで止まったときは**エラー**です。

本体とスタジオのバージョンが一致しないときは「ファームウェアの更新が必要です」の画面が出て、実機の操作ができなくなります。「1. アップデートする」「2. 内蔵スタジオで使う」「3. 実機なしで使う」のいずれかを選びます。「実機なしで使う」を選ぶとブロックの編集だけ続けられ、「▶ 実行」の下に「ファームウェアのバージョンが一致しないため、実機の操作は無効です。」と表示されます。

ハードウェアの状態：ポコロボにつながっていると、いま出ている値が並びます。ブロックが思ったとおりに効いているかをここで確かめられます。

ハードウェア

サーボ（角度°）

DCモータ（速度 %）

S0: 90

S1: 90

M0: 0

M1: 0

S2: 90

S3: 90

M2: 0

M3: 0

エンコーダ（カウント）

LED

E0: 0

E1: 0

E2: 0

E3: 0

R: 0

G: 0

B: 0

ゲームパッド

X 未接続

欄	内容
サーボ（角度°）	S0～S3に出している角度。動かしていないときは ---
DCモータ（速度 %）	M0～M3に出している速度（-100～100）。ブレーキ中は「ブレーキ」と表示
エンコーダ（カウント）	E0～E3の累積カウント
LED	いま光らせている色と R/G/B の値
ゲームパッド	つないでいる機器の種類と、押しているボタン

ミニでの違い：ミニはDCモータが2つ（M0・M1）で、エンコーダはありません。そのため「エンコーダ（カウント）」の欄は表示されません。サーボは4つ（S0～S3）で共通です。

⑤ ゴミ箱

ブロックをここにドラッグすると削除できます。

⑥ 設定ボタン

画面右下の歯車マークのボタンです。起動時設定やコントローラのペアリングなど、本体の設定を変更できます。

設定

×

コントローラペアリング

ワイヤレスコントローラをペアリングします。『ペアリング開始』を押し、ペアにしたいコントローラのボタンを押してください。

ペアリング開始

ペア解除

起動時設定

☐ 起動時にプログラム自動実行

保存

リセット

閉じる

設定	内容
コントローラペアリング	専用コントローラをペアリング・解除します（→「ポコロボ コントローラ説明書」）
起動時にプログラム自動実行	電源を入れたとき、本体に保存されたプログラムを自動で実行します

変更したら **【保存】** を押してください。

4. 基本操作

ブロックを置く・繋げる

- ②ブロック置き場から使いたいブロックを **ドラッグ** して、③ワークスペースに持っていきます
- ブロックを離すと配置されます
- ブロックの **凸（出っ張り）** と **凹（くぼみ）** を合わせると接続されます
- 上から下へつなげた順番に実行されます

ポコロボスタジオ 操作ガイドー 7 / 28

ブロックを削除する

方法は2つあります：

- ブロックを **⑤ゴミ箱までドラッグ** して離す
- ブロックを **右クリック** → 「ブロックを削除」を選択する

繋がっているブロックをまとめて削除したい場合は、一番上のブロックをゴミ箱へドラッグしてください。繋がった下のブロックも一緒に削除されます。

数値を変更する（シャドウブロック）

ブロックの中にある **灰色の数字** をクリックすると、直接数値を入力して変更できます。

この灰色の数字ブロックを「シャドウブロック」（初期値が入った編集可能な灰色ブロック）と呼びます。シャドウブロックは削除できませんが、計算ブロックや変数ブロックを上から重ねるとその値に置き換えることができます。重ねたブロックを外すと、元の灰色ブロックに戻ります。

ブロックを複製する

ブロックを **右クリック** → 「ブロックを複製」を選択すると、同じブロックが複製されます。中にはめ込んだブロックと一緒に複製されますが、下につながっているブロックは複製されません。

ワークスペースを移動・拡大縮小する

- **移動**：ワークスペースの空いている部分をドラッグする
 - **拡大縮小**：マウスホイールを回す
 - **全体を表示**：右下のズームアウトボタンを繰り返し押す
-

プログラムを実行する

1. **④操作パネルの「▶ 実行」** ボタンをクリックします
 2. プログラムがポコロボに送信され、実行が始まります
 3. 実行中はブロックの編集ができなくなります
-

プログラムを停止する

操作パネルの **「■ 停止」** ボタンをクリックします。実行が止まり、ブロックの編集ができるようになります。

プログラムを保存する

保存のしかたは2通りあります。ヘッダの **プログラム名▼** をクリックすると、どちらもこのメニューから選べます。



メニュー	何をするか
保存	ロボスポのアカウントに保存します。別のパソコンからログインしても続きから使えます
コピーを保存	いま開いているプログラムを、別の名前でもう1つ作ります
名前を変更	保存したプログラムの名前を変えます
↓ .json で書き出し	プログラムを JSON（データ保存用のファイル形式）ファイルとしてパソコンに保存します

- ヘッダの 保存 ボタンは、メニューの「保存」と同じです（よく使うため外に出してあります）
- アカウントへの保存には**ログインが必要です**。ログインしていないと、保存の項目に鍵マーク（）が付きます
- ↓ .json で書き出し の保存先は通常、ブラウザの「ダウンロード」フォルダです（ブラウザの設定で変更可能）

本体の中のスタジオ（内蔵スタジオ）では：アカウントへの保存はできません。操作パネルに ダウンロード / アップロード のボタンが出るので、JSON ファイルとして保存・読み込みします。ヘッダの サンプルを読み込む からサンプルプログラムを開けます。

ダウンロード

アップロード

プログラムを読み込む

- **アカウントに保存したもの**：ヘッダの **プログラム名 ▼** → 開く… の「マイプロジェクト」から選びます
- **サンプル**：「 開く…」の「サンプル」から選びます。開いたサンプルはプログラム名の横に「読込専用」と付き、ヘッダの保存ボタンが コピーとして保存 に変わります
- **JSON ファイル**：同じメニューの ↑ .json から読み込み をクリックし、保存済みのファイルを選択します
- **つないだポコロボに保存されているもの**：ポコロボをつないだとき、画面にブロックが無ければ自動で開きます。ブロックがあるときは「つないだポコロボにプログラムがあります」のお知らせが出るので、「つないだポコロボのプログラムを開く」を押すと開きます（内蔵スタジオではこのお知らせは出ません）

注意：読み込むと現在のプログラムは上書きされます。必要なプログラムは先に保存しておいてください。プログラム名の横に●未保存と出ているときは、まだ保存していない変更があります。

5. 各ブロックの説明

このセクションでは、スタジオで使えるすべてのブロックを、ブロック置き場に並んでいる島の順に説明します。

このセクションの見方

各ブロックの説明は次の形式で書かれています：

1. ブロックの画像（実際の見た目）
2. 機能の説明
3. 入力できる値の範囲
4. 使い方の例または注意点

動かす

ポコロボのDCモータ・サーボ・LED・ブザーを動かすブロックです。

ミニをお使いの方へ：ブザーのブロックはミニでは使えません（実行するとエラーで止まります）。DCモータの番号も0・1のみです。

DCモータ

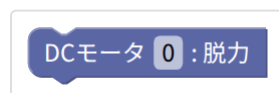


指定した番号のDCモータを、指定した速度で動かします。

入力	範囲	説明
番号	0～3	本体のコネクタ（M0～M3）に対応。4以上を指定して実行すると「無効なDCモータID」のエラーで止まります。 ミニは0・1のみ （2以上は同じエラー）
速度	-100～100（%）	正の値で正転、負の値で逆転、0で停止

DCモータを動かすには**パワー用電源**が必要です。ロジック用電源だけでは動きません。

DCモータのブレーキ・脱力



ブロック置き場の「動かす」にあります。DCモータの止め方を選べます。

ブロック	動き
ブレーキ	モータに急ブレーキをかけて止めます。次に「DCモータ」ブロックで速度を指定するまで、ブレーキがかかったままになります
脱力	モータの力を抜き、勢いで回りながら止まります。速度0と同じ動きです

「DCモータ」ブロックで速度を0にしたときは、脱力と同じ止まり方になります。

サーボ



指定した番号のサーボを、指定した角度に動かします。

入力	範囲	説明
番号	0～3	本体のコネクタ（S0～S3）に対応。4以上を指定して実行すると「無効なサーボID」のエラーで止まります
角度	0～180（度）	0～180度の範囲。90°が中央。範囲外の値は0または180として動きます

LED



本体内蔵のフルカラーLEDの色を設定します。赤・緑・青の3色を組み合わせることで自由な色を作れます。

入力	範囲	説明
赤（R）	0～100	0で消灯、100で最大輝度
緑（G）	0～100	0で消灯、100で最大輝度
青（B）	0～100	0で消灯、100で最大輝度

色の組み合わせ例：

色	赤	緑	青
赤	100	0	0
緑	0	100	0
黄	100	100	0
白	100	100	100
消灯	0	0	0

ブザーを鳴らす



指定した周波数で音を鳴らします。Hz（ヘルツ：1秒間の振動回数）で音の高さを指定します。鳴らせる範囲は100～10000 Hzで、範囲外の値は近いほうの端の値になります。0以下を指定すると音が止まります。

ミニにはブザーがありません。「ブザーを鳴らす」/「ブザーを止める」のブロックをミニで使うと、実行中にエラーが表示されてプログラムが止まります。

周波数	音程の目安
262 Hz	ド（中央C）
330 Hz	ミ
392 Hz	ソ

周波数	音程の目安
440 Hz	ラ（A音）

音を止めるには「ブザーを止める」ブロックを使います。

ブザーを止める



鳴っているブザーを止めます。「ブザーを鳴らす」で鳴らした後、このブロックで停止させます。

操縦する

操縦ブロックはゲームパッドの操作を読み取るブロックです。専用コントローラ（専用無線）でつないだときも、同じブロックでその操作を読み取れます。

ゲームパッドの2つの接続方法

接続先	用途	備考
ポコロボの USB Type-A（大きい方）	ロボットが直接ゲームパッドを読む	USB Type-C に何かをつないでいる間は自動的に無効になる
パソコンの USB ポートに直接接続	パソコン経由でスタジオを通じて操作する	USB Type-C接続中でも使用可能

パソコンに直接つないだ場合：ゲームパッドをパソコンに接続すると、Chrome がゲームパッドの入力を読み取り、スタジオを通じてポコロボに送ります。ポコロボの Type-A ポートにつながなくても動作します。

操縦ブロックには2種類あります：

- **～を押したら**（中にブロックを入れる形）：押している間だけ中のブロックを実行する
- **～の値**（丸い形）：押している間は 1、離れていると 0 を返す。「もし～なら」や計算ブロックにはめ込んで使う

ボタンを押したら（A/B/X/Y）



A・B・X・Yのいずれかのボタンを押している間、中のブロックを実行します。

ブロック右上の ⊕ **ボタン** をクリックすると「でなければ」の枠が追加されます。ボタンを押していないときの動作を指定できます。⊖ **ボタン** で削除できます。

十字キーを押したら



十字キー（↑↓←→）のいずれかを押している間、中のブロックを実行します。

LRボタンを押したら



LB・RB・LT・RTのいずれかを押している間、中のブロックを実行します。

その他を押したら



「その他」のボタン（Select / Start / L3 / R3 / Home）を押している間だけ、中のブロックを実行します。使い方は「ボタンを押したら」と同じで、ドロップダウンで選べるボタンが違います。

ドロップダウンの選択肢	一般的なゲームパッドでの呼び名
Select	セレクト／ビュー／シェア
Start	スタート／メニュー／オプション
L3	左スティックの押し込み
R3	右スティックの押し込み
Home	ホーム／ガイド

ボタンの値（A/B/X/Y）



ボタンを押している間は **1**、離れていると **0** を返します。「もし～なら」や比較ブロックにはめ込んで使います。

十字キーの値



十字キー（↑↓←→）の押下状態を **0** または **1** で返します。押している間は **1**、離れていると **0** です。

LRボタンの値



LRボタン（LB / RB / LT / RT）の状態を数値で返します。押している間は **1**、離れていると **0** です。「もし～なら」や計算ブロックにはめ込んで使います。

トリガーの押し込み量



LT / RT のトリガーを、どれだけ押し込んでいるかを **0～100** の数値で返します。離れていると **0**、いっぱいまで押し込むと **100** です。DCモータの速度にそのまま渡せば、押し込むほど速く走るアクセルになります。

押し込み量を測れないゲームパッド（トリガーを押したか押していないかだけが分かるもの）では、**0** か **100** になります。ゲームパッドをつないでいないときは **0** です。

その他の値

その他 Select ▼

「その他」のボタン（Select / Start / L3 / R3 / Home）の状態を数値で返します。押している間は **1**、離れていると **0** です。

スティックの軸の値

スティック 左X ▼

スティックの傾き具合を **-100～100** の数値で返します。左X・左Y・右X・右Y から選択できます。この範囲はDCモータの速度（-100～100）とそのまま同じなので、計算せずにDCモータへ渡せます。

方向	値の目安
前（奥）に倒す（左Y / 右Y）	マイナス（-100に近づく）
後ろ（手前）に引く（左Y / 右Y）	プラス（100に近づく）
左に倒す（左X / 右X）	マイナス（-100に近づく）
右に倒す（左X / 右X）	プラス（100に近づく）
中央（何も触らない）	0

注意：スティックを前に倒すと値が**マイナス（-100方向）**になります。「前倒し → DCモータ前進」にしたい場合は、スティックの値にマイナスを掛けてください（計算ブロックで $\times -1$ ）。これはゲームパッドの標準仕様によるものです。

センサー・時間

エンコーダの値や、プログラムを実行し始めてからの時間を読むブロックです。

ミニをお使いの方へ：エンコーダのブロックはミニでは使えません（実行するとエラーで止まります）。

エンコーダ 0 のカウント

エンコーダ 0 ▼ のカウント

DCモータの回転量を数える **エンコーダ** の累積カウントを返します。番号（0～3）で読み取るエンコーダを選びます。4以上を指定して実行すると「無効なエンコーダID」のエラーで止まります。

値	説明
カウント	DCモータが正転すると増え、逆転すると減ります。プログラムを実行し始めたときに0にリセットされます

スタンダードのみ。 ミニにはエンコーダがないため、このブロックを使うと実行中にエラーが表示されてプログラムが止まります。

エンコーダ 0 のカウントをリセット

エンコーダ 0 のカウントをリセット

指定したエンコーダ（0～3）の累積カウントを 0 に戻します。4以上を指定して実行すると「無効なエンコーダ ID」のエラーで止まります。距離や回転数を数え直したいときに使います。

スタンダードのみ。 ミニにはエンコーダがないため、このブロックを使うと実行中にエラーが表示されてプログラムが止まります。

プログラム開始からの時間（秒）

プログラム開始からの時間（秒）

プログラムを実行し始めてからの時間を秒で返します。0.001 のような小数も返るため、細かい時間の計測に使えます。

場面	返る値
実行した直後	0 に近い値
3秒後	3 に近い値
「■ 停止」のあと「▶ 実行」した直後	0 に近い値（実行のたびに 0 から数え直します）

前回からの経過時間を求める

このブロックの値を変数に覚えておき、次の周回で引き算すると、1周にかかった時間が求められます。速度（進んだ量 ÷ 時間）の計算に使います。

1. 「今の時間」に **プログラム開始からの時間（秒）** を入れる
2. 「経過時間」に 「今の時間」－「前の時間」 を入れる
3. 「前の時間」に 「今の時間」 を入れる

これを繰り返すの中に置くと、「経過時間」に毎週の所要時間が入ります。最初の1周だけは「前の時間」がまだ 0 なので、経過時間が実際より大きく出る点に注意してください。

長時間動かしたときの注意：ポコロボが扱える数の細かさには限りがあるため、実行を始めてから1時間を超えると、この時間の刻みが 0.5ミリ秒ほどまで粗くなります。数分の試合や練習では問題になりません。

繰り返す・待つ

プログラムの流れのうち、繰り返しと待つことを決めるブロックです。

～Hzで繰り返す



1秒間に指定した回数だけ繰り返します。Hz（ヘルツ：1秒間の繰り返し回数）で頻度を指定します。

値	意味
1 Hz	1秒に1回
50 Hz	1秒に50回（モータ制御に適した頻度）
100 Hz	1秒に100回

「ずっと繰り返す」より細かくタイミングを制御したい場合に使います。ブロックを置いた直後は 0 Hz になっています。0 のまま実行するとエラーで止まるので、回数を入れてから実行します。

ずっと繰り返す



中に入れたブロックを **無限に繰り返す** 実行します。

このブロックを使わないと、ブロックは上から一度だけ実行されます。最後まで実行しても「■ 停止」を押すまでは「実行中」のままで、最後に指定した LED の色やDCモータの速度はそのまま保たれます。ゲームパッドの操作のように **変化に合わせて動かし続けたい処理** は、このブロックの中に入れます。

ロボット制御では、基本的にすべての動作をこのブロックで囲む使い方が標準的です。

よくある間違い：「ずっと繰り返す」の**外**にゲームパッドのブロックを置くと、実行した瞬間の状態を一度読むだけで、その後の操作には反応しません。操作に反応させたい処理はこのブロックの**中**に入れてください。

複数置いた場合：ワークスペースに「ずっと繰り返す」ブロックを2つ以上置くと、それぞれが交互に実行されます（並列処理しているように見えますが、内部では1つずつ順番に処理しています）。慣れないうちは1つにまとめることを推奨します。

～の間繰り返す



条件の部分の値が **0以外の間**、中のブロックを繰り返します。値が **0になると停止** します。

「もし～なら」と組み合わせることで、特定の条件が続く間だけ動作させることができます。

待つ



指定した秒数だけ処理を止めて待ちます。**0以上の値**を指定できます（小数も可）。負の値を指定するとエラーになります。

入力値	待ち時間
1	1秒
0.5	0.5秒（500ミリ秒）
0.1	0.1秒（100ミリ秒）

DCモータをある時間だけ動かして止めたいときなどに使います。

よくある間違い：「ずっと繰り返す」の中に「待つ」を入れると、待っている間もループが止まります。ゲームパッド操作のようにリアルタイムで反応させたい処理と同じループに「待つ」を入れると、意図した動作にならない場合があります。

条件で分ける

条件によって動きを変えるブロックと、その条件を組み立てるための比較・論理演算のブロックです。

もし～なら



条件の部分にはめたブロックの値によって動作が変わります。

- 値が **0** → 中を **実行しない**（条件が満たされていない）
- 値が **0以外**（1でも、50でも、-3でも） → 中を **実行する**（条件が満たされている）

ほとんどの場合、ボタンや比較ブロックの結果（0か1）をはめるので、「**1なら動く、0なら動かない**」と考えればOKです。

ブロック右上の **⊕ ボタン** をクリックすると「でなければ」の枠が追加されます。⊖ ボタン で削除できます。

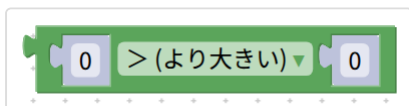
もし（複数条件）



3つ以上の条件で処理を分けたいときに使います。⊕ ボタン で条件を追加、⊖ ボタン で削除できます。

「それ以外」の枠は、どの条件にも該当しなかったときに実行されます。

比較（= ≠ < ≤ > ≥）



2つの値を比較します。条件が **成立すると 1、不成立のとき 0** を返します。

記号	意味
=	等しい
≠	等しくない
<	左が小さい
≤	左が小さいか等しい
>	左が大きい
≥	左が大きい等しい

「もし～なら」ブロックの条件部分にはめ込んで使います。

かつ (AND) / または (OR)



2つの条件を組み合わせます。

演算	意味	結果が1になる条件
AND（かつ）	両方が真のとき	左も右も 0 以外
OR（または）	どちらかが真のとき	左か右の少なくとも一方が 0 以外

「ボタンAを押している **かつ** スティックが前に倒れている」のような複合条件を作るときに使います。

でない (NOT)



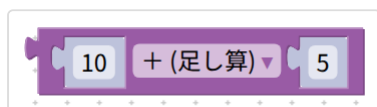
値の真偽を反転します。

入力	出力
0（偽）	1（真）
0以外（真）	0（偽）

「ボタンを押して **いない** とき」のような否定条件を作るときに使います。

計算する

計算（+ - × ÷）



2つの値を計算します。+（足し算）、-（引き算）、×（掛け算）、÷（割り算）から選択できます。÷で0で割るとエラーで止まります。

他のブロックの数値入力にはめ込んで使います。例えばスティックの値（-100～100）に -1 を掛けて、前に倒したときにDCモータが前進するよう向きをそろえるときに使います。

三角関数（sin / cos / tan ほか）



ドロップダウンから **sin**・**cos**・**tan**・**asin**・**acos**・**atan** の6つを選べる計算ブロックです。波のような動きや角度の計算に使います。

関数	入力	出力
sin / cos / tan	角度（度）	比
asin / acos / atan	比	角度（度）

- **角度の単位は「度」です**（ラジアンではありません）。例えば **sin(90)** は 1、**cos(180)** は -1 になります。
- **asin / acos** は入力を **-1～1** の範囲として扱います。範囲を超える値を入れたときは、いちばん近い端（-1 または 1）として計算します。
- **tan** は 90度・270度の近くでとても大きな値になります。

平方根・絶対値・べき乗



数学でよく使う3つの計算ブロックです。

ブロック	計算	例
$\sqrt{\quad}$ (平方根)	入力の平方根 (ルート)	$\sqrt{4} \rightarrow 2$
絶対値 $ x $	マイナスをプラスにする	絶対値(-3) $\rightarrow 3$
$^$ (べき乗)	A を B 回かける (A の B 乗)	$2^8 \rightarrow 1024$

- **平方根** は、マイナスの値を入れると **0** として計算します (例: $\sqrt{-1} \rightarrow 0$)。
- **べき乗** は、マイナスの数や小数の組み合わせ (例: (-2) の 0.5 乗) では計算できない場合があります、そのときは計算できないことを示す値になります。

変数

変数とは

変数 (値に名前を付けて記憶する仕組み) は、**値を入れたり書き換えたりできる名前付きの箱**です。プログラムの実行中に中身が変化します。

なぜ変数を使うのか？ — 同じ値を複数の場所で使いたいときに役立ちます。例えばスティックの値を2つのDCモータに使いたい場合、変数に一度記憶させて使い回せます。変数を使わずに同じ計算を2回書くと、ブロックが増えて読みにくくなるうえ、片方だけ修正し忘れるミスが起きやすくなります。

変数を作る

ブロック置き場の「変数」の先頭にある「**変数を作成**」ボタンをクリックします。名前を入力すると変数が作成され、使えるブロックが追加されます。

新しい変数を作成

変数名

例: count, value, speed

キャンセル

OK

現在、作成した変数の名前変更と削除はできません。

変数を設定する

変数 speed を 0 に設定

変数に値を記憶させます。右側の入力に、記憶させたい値または計算ブロックをはめ込みます。

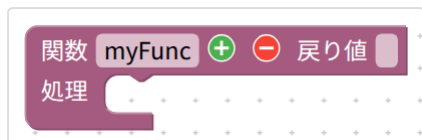
変数を読み出す

変数 speed

記憶されている値を取り出します。DCモータブロックやLEDブロックの数値入力にはめ込んで使います。

関数

関数



関数（ひとまとまりの処理に名前を付けたもの）は、**同じ処理を何度も書かずにすませる**ための仕組みです。

「関数」ブロックに処理を入れておくと、あとから名前呼び出せます。

なぜ関数を使うのか？ — 例えば「前に進む」という動きが3か所に出てくるとき、そのたびにDCモータのブロックを並べると、速度を変えなくなったときに3か所すべてを直すことになります。関数にまとめておけば、直すのは1か所だけで済みます。

「関数」ブロックは、他のブロックと違って**上下にはつながりません**。ワークスペースの空いている場所に単独で置きます。

部分	意味
名前（myFunc と書かれた欄）	関数の名前。分かりやすい名前に書き換えられます
⊕ / ⊖ のボタン	引数（関数に渡す値）を増やす・減らす
戻り値のチェックボックス	チェックを入れると、関数が計算結果を返せるようになります
処理	ここに実行したいブロックを入れます

関数を実行



作った関数を呼び出します。ドロップダウンで呼び出す関数を選びます。関数に引数があるときは、渡す値を入れる差し込み口が自動で増えます。

関数の結果



関数を呼び出して、その**戻り値**を受け取るブロックです。丸い形をしていて、計算ブロックや「もし～なら」にはめ込んで使います。

「関数」ブロック側で**戻り値のチェックボックスにチェックを入れた関数**に対して使います。

引数



関数に渡された値を読み出すブロックです。**関数の中でだけ**使います。ドロップダウンで、どの引数を読むかを選びます。

引数の名前は「関数」ブロックの名前欄の隣で書き換えられます。

6. よくある質問

ブロックが見つからない

ブロック置き場の左端の目次で島の名前を押すと、その島の位置へ移動します。ブロックは「動かす」「操縦する」「センサー・時間」「繰り返す・待つ」「条件で分ける」「計算する」「変数」「関数」の8つの島に分かれています（→ §3「**②** ブロック置き場（左側）」の表）。ミニをつないでいるときは、ブザーとエンコーダのブロックは出ません。

ブロックを削除したい

⑤ ゴミ箱にドラッグ、または右クリック → 「ブロックを削除」で削除できます。

プログラムを保存・読み込みたい

ヘッダの **プログラム名**▼のメニューから行います。

- **保存**：「 保存」でロボスポのアカウントに保存します（要ログイン）。ファイルとして残したいときは「 .json で書き出し」を使います
- **読み込み**：「 開く…」でアカウントに保存したものを、「 .json から読み込み」でファイルを読み込みます

本体の中のスタジオ（内蔵スタジオ）では、操作パネルの **↓ ダウンロード / ↑ アップロード** を使います。

「パソコン用アプリ「ポコロボ コネクト」をインストールしましょう」と表示される

パソコン用アプリ「ポコロボ コネクト」が入っていないか、起動していません。対処は「ポコロボ 本体セットアップ・安全ガイド」§8「困ったときは」を参照してください。

「ファームウェアの更新が必要です」と表示される

本体とスタジオのバージョンが一致していません。画面の「アップデートする (vX.Y.Z)」で本体を更新するか、「実機なしで使う」でブロックの編集だけ続けます（→ §3「**④** 操作パネル（右側）」）。

実行できない・エラーが出る

接続状態の表示が「未接続」の場合、ポコロボと通信できていません。画面の上に黄色いお知らせが出ていないか確認し（→ §2「画面の上に黄色いお知らせが出たとき」）、本体・接続まわりは「ポコロボ 本体セットアップ・安全ガイド」§8「困ったときは」を参照してください。

DCモータが動かない

DCモータは、**パワー用電源**がないと動きません。詳しくは「ポコロボ 本体セットアップ・安全ガイド」§8「DCモータが動かない」を参照してください。

サーボが動かない

「ポコロボ 本体セットアップ・安全ガイド」§8「サーボが動かない」を参照してください。

ゲームパッドが反応しない

接続方法によって確認する内容が異なります。

パターン①：ゲームパッドをポコロボに直接つないでいる場合

「ポコロボ 本体セットアップ・安全ガイド」 §8「ゲームパッドが反応しない」を参照してください。USB Type-C に何かをつないでいる場合は、Type-A が自動的に無効になります → パターン②へ

パターン②：ゲームパッドをパソコンに直接つないでいる場合

- 1. ゲームパッドがパソコンに正しく接続されているか確認します
- 2. Chrome のアドレス欄で `chrome://gamepad-internals` を開き、ゲームパッドが認識されているか確認します
- 3. 認識されていない場合は、ゲームパッドを抜き差しして再接続してください
- 4. それでも認識されない場合は、動作確認済み機種をお使いください

実行中にブロックを編集できない

仕様です。実行中はブロックの編集がロックされます。「■ 停止」ボタンを押すと編集できるようになります。

ゲームパッドとパソコン接続は自動で切り替わる

USB Type-C に何かをつなぐとパソコンとの通信、何もつないでいないときはゲームパッドの読み取りに自動で切り替わります。手動の設定はありません。

変数を削除したい・名前を変えたい

現在、変数の削除と名前変更はできません。不要な変数がある場合は、プログラムを作り直すか、動作に問題がなければそのままにしてください。

7. 用語集

用語	意味
ポコロボ コネクト	パソコンに常駐する小さなアプリ。スタジオの画面とUSBでつないだポコロボの橋渡しをする。初回に一度入れれば、次からは自動で起動する
内蔵スタジオ	インターネットがないとき、 <code>http://192.168.20.1</code> でポコロボ本体の中のスタジオを開く方法。ポコロボ コネクトもインターネットも不要（USB Type-A to Type-Cケーブルは必要）
シャドウブロック	ブロック内に最初から入っている灰色の数値ブロック。クリックして直接編集でき、削除はできない
Hz（ヘルツ）	1秒間に起きる回数。ブザーでは音の振動の回数（数が大きいほど高い音）、繰り返しブロックでは1秒間に処理する回数
JSON	プログラムの保存に使われるデータファイル形式。拡張子は <code>.json</code>
ブラウザ経由のゲームパッド接続	ゲームパッドをパソコンに直接接続すると、Chrome がゲームパッドの入力を読み取り、スタジオを通じてポコロボに送る仕組み
条件分岐	条件に応じて異なる処理を実行すること。「もし～なら」ブロックで実現する
変数	値に名前を付けて記憶する仕組み。実行中に中身を書き換えられる
正転・逆転	DCモータが正方向・逆方向に回ること。DCモータブロックの速度の正負で切り替わる
サーボ	指定した角度まで回転して保持するモータ。0～180度で制御する

付録A: ゲームパッドボタン対応表

名前	番号	一般的なボタン名
A	0	A / ○

名前	番号	一般的なボタン名
B	1	B / ×
X	2	X / △
Y	3	Y / □
LB / RB	4 / 5	L1 / R1
LT / RT	6 / 7	L2 / R2
Select / Start	8 / 9	Back / Menu
L3 / R3	10 / 11	スティック押込
↑ ↓ ← →	12～15	十字キー
Home	16	ホームボタン

軸	番号	範囲
左X / 左Y	0 / 1	-100～100
右X / 右Y	2 / 3	-100～100

※ゲームパッドの機種によってボタン配置が異なる場合があります。

付録B: ブロック一覧早見表

「～したい」から使うブロックを探せます。

やりたいこと	使うブロック	島
LEDの色を変えたい	LED	動かす
LEDを消したい	LED（赤0・緑0・青0）	動かす
DCモータを動かしたい・止めたい	DCモータ	動かす
DCモータを急に止めたい	ブレーキ	動かす
DCモータの力を抜いて止めたい	脱力	動かす
サーボの角度を変えたい	サーボ	動かす
音を鳴らしたい	ブザーを鳴らす	動かす
音を止めたい	ブザーを止める	動かす
ゲームパッドのボタンで動かしたい	ボタンを押してたら	操縦する
十字キーで動かしたい	十字キーを押してたら	操縦する
LB/RB/LT/RT で動かしたい	LRボタンを押してたら	操縦する
Start / Select など で動かしたい	その他を押してたら	操縦する
ボタンの押下を数値で使いたい	ボタンの値 / 十字キーの値 / LRボタンの値 / その他 の値	操縦する
スティックの傾きを取得したい	スティックの軸の値	操縦する
トリガーの押し込み量を使いたい	トリガーの押し込み量	操縦する
DCモータの回転量を測りたい（スタンダードのみ）	エンコーダ0のカウント	センサー・時間
測った回転量をリセットしたい（スタンダードのみ）	エンコーダ0のカウントをリセット	センサー・時間

やりたいこと	使うブロック	島
経過時間を測りたい・速度を計算したい	プログラム開始からの時間（秒）	センサー・時間
ずっと動かし続けたい	ずっと繰り返す	繰り返す・待つ
一定の頻度で繰り返したい	～Hzで繰り返す	繰り返す・待つ
条件が続く間だけ繰り返したい	～の間繰り返す	繰り返す・待つ
少し待ってから次に進みたい	待つ	繰り返す・待つ
条件によって動きを変えたい	もし～なら	条件で分ける
条件を3つ以上に分けたい	もし（複数条件）	条件で分ける
大小を比較したい	比較	条件で分ける
「かつ」「または」を使いたい	かつ (AND) / または (OR)	条件で分ける
条件を反転したい	でない (NOT)	条件で分ける
計算結果を使いたい	計算（＋－×÷）	計算する
三角関数を使いたい（度）	三角関数（sin/cos/tan ほか）	計算する
平方根を求めたい	$\sqrt{\quad}$ （平方根）	計算する
絶対値を求めたい	絶対値 $ x $	計算する
べき乗を求めたい	$\wedge$ （べき乗）	計算する
値を記憶しておきたい	変数を設定 / 変数	変数
同じ処理を何度も使い回したい	関数 / 関数を実行	関数
関数に値を渡したい	関数（ $\oplus$ で引数を追加） / 引数	関数
関数から計算結果を受け取りたい	関数の結果	関数

付録C: 作ってみよう（例題）

ここからは実際にプログラムを組み立てる例題です。各ブロックの仕様は「5. 各ブロックの説明」、困ったときは「6. よくある質問」を参照してください。

例題① LEDを光らせよう

所要時間：約10分 / 使うブロック：LED

この例題を始める前に

- 前提：スタジオが開いていること、操作パネルに「接続中」と表示されていること
- 成功条件：実行ボタンを押すとポコロボのLEDが指定した色で光る

ブロック置き場の「動かす」から「LED」をドラッグして配置し、赤の値を100に変更します。



「▶ 実行」を押すとLEDが赤く光ります。色の組み合わせを変えていろいろな色を試してみましょう。

色	赤	緑	青
赤	100	0	0
緑	0	100	0
黄	100	100	0
白	100	100	100

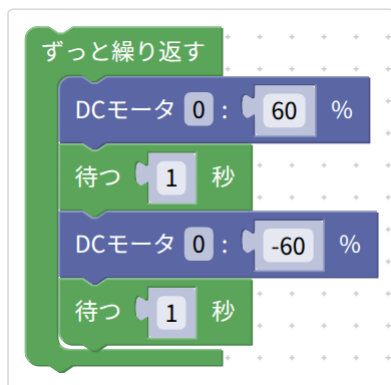
例題② DCモータを動かそう

所要時間：約15分 / 使うブロック：ずっと繰り返す、DCモータ、待つ

この例題を始める前に

- 前提：パワー用電源が接続されていること、DCモータが M0～M3 のいずれか（ミニでは M0・M1 のどちらか）に接続されていること
- 成功条件：実行ボタンを押すとDCモータが回転する
- つまずきやすい点：パワー用電源が未接続 → DCモータが動かない（「ポコロボ 本体セットアップ・安全ガイド」§4「電源の種類と接続方法」を参照）

なぜ「ずっと繰り返す」で囲むのか？ — ブロックは上から一度だけ実行されます。囲まないと、DCモータを正転・逆転させる動きは1回で終わり、最後の速度のまま回り続けます。「ずっと繰り返す」で囲むと、「■ 停止」を押すまでこの動きを繰り返します。



- DCモータの速度は -100～100%（負の値で逆転、0で停止）
- 「待つ」ブロックの単位は 秒（1=1秒、0.5=0.5秒）

動かない場合：DCモータのコネクタと **パワー用電源** を確認してください。ロジック用電源だけでは動きません。

例題③ ゲームパッドで操縦しよう

所要時間：約20分 / 使うブロック：ずっと繰り返す、十字キーを押してたら、DCモータ

この例題を始める前に

- 前提：ゲームパッドがポコロボの Type-A またはパソコンに接続されていること
- 成功条件：ゲームパッドのボタンを押すとDCモータが反応する
- つまずきやすい点：スティックを前に倒すと値がマイナスになる → 前進させたい場合は × -1 で反転する

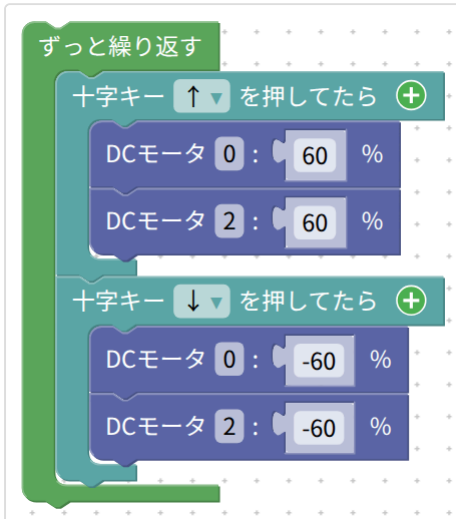
ゲームパッドをポコロボの大きい方のUSBポートに接続します（パソコンに直接接続してもかまいません）。

パソコンとUSB接続中はポコロボ側のUSBポートが使えません。その場合はゲームパッドをパソコンに直接接続してください（ブラウザのChromeがゲームパッドの入力を読み取り、スタジオを通じてポコロボに送ります）。

手順：

1. 緑色の「ずっと繰り返す」ブロックをワークスペースに配置する
2. 水色ブロックから「十字キーを押してたら」（↑を選択）を「ずっと繰り返す」の中にはめる
3. 青の「DCモータ」ブロックを2つ、「十字キー↑を押してたら」の中にはめる
4. 1つ目：DCモータ番号を **0**、速度を **60** に設定
5. 2つ目：DCモータ番号を **2**、速度を **60** に設定
6. 同様に「十字キー↓を押してたら」を追加し、DCモータ速度を **-60** にする

DCモータ番号について：番号は本体のコネクタ（M0～M3）に対応します。自分の配線に合わせて番号を変更してください。**ミニではM0・M1（番号0・1）に読み替えてください（M2はありません）。**



応用：←→を追加して左右旋回、ボタンA/Bでサーボやブザーも試してみてください。

例題④ 条件分岐を使おう

所要時間：約20分 / 使うブロック：ずっと繰り返す、もし～なら、ボタンの値、サーボ、LED

この例題を始める前に

- 前提：ゲームパッドが接続されていること、サーボがS0～S3のいずれかに接続されていること
- 成功条件：ボタンAを押している間だけサーボが動き、離すと戻る
- つまづきやすい点：「ずっと繰り返す」で囲んでいない → ボタンの状態を一度読むだけで、その後の操作に反応しない

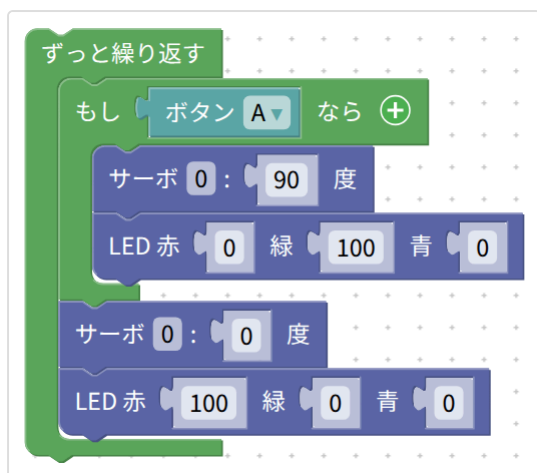
「もし～なら」ブロックは、条件の部分にはめたブロックの値によって動作が変わります。

- 値が **0** → 中を **実行しない**（条件が満たされていない）
- 値が **0以外**（1でも、50でも、-3でも） → 中を **実行する**（条件が満たされている）

ほとんどの場合、ボタンや比較ブロックの結果（0か1）をはめるので、「**1なら動く、0なら動かない**」と考えればOKです。



「ボタンA」の値ブロック **ボタン A** は、押していると **1**、離していると **0** を返します。これを「もし～なら」にはめると、ボタンを押している間だけ中が実行されます。



- ボタンA押下中 → サーボ90度 + LED緑
- 離している間 → サーボ0度 + LED赤

「もし〜なら」ブロックの右上にある **⊕ ボタン** をクリックすると「でなければ」の枠が追加されます。⊖ **ボタン** で削除できます。3つ以上の分岐が必要な場合は、ブロック置き場にある「もし（複数条件）」ブロックを使ってください。

例題⑤ 変数を使おう

所要時間：約25分 / 使うブロック：ずっと繰り返す、変数を設定、変数、計算、スティック、DCモータ

この例題を始める前に

- 前提：ゲームパッドが接続されていること、DCモータが2台（M0・M2など。**ミニでは M0・M1**）接続されていること
- 成功条件：スティックの傾きにに応じて2台のDCモータが同じ速度で動く
- つまづきやすい点：スティックを前に倒すと値がマイナスになるため、DCモータが期待と逆方向に回る → $\times -1$ で反転する

変数（値に名前を付けて記憶する仕組み）は、**値を入れたり書き換えたりできる名前付きの箱**です。

なぜ変数を使うのか？ — 計算結果を1箇所ですしか使わないなら、計算ブロックを直接はめ込んでも構いません。しかし、**同じ値を複数の場所で使いたい場合**に変数が役立ちます。下の例では「スティック左Y $\times -1$ 」の結果をDCモータ0番と2番の両方に使いたいため、変数「speed」に一度記憶させて使い回しています。

ブロック置き場の「変数」の先頭にある「変数を作成」ボタンから新しい変数を作れます。

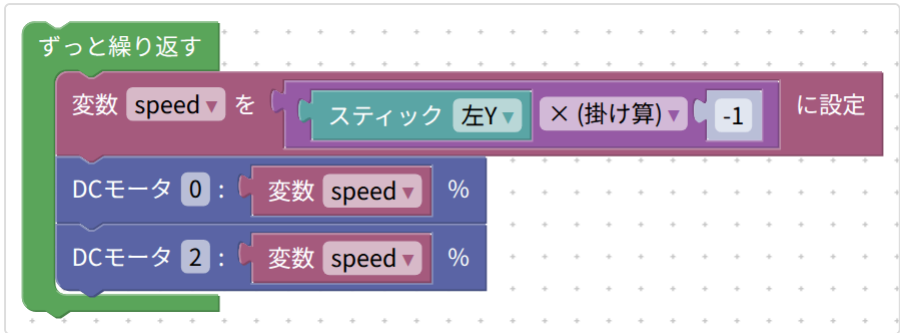
新しい変数を作成

変数名

例: count, value, speed

キャンセル

OK



スティックの値とDCモータの速度は、どちらも **-100～100** の範囲です。

- 前（奥）に倒す → **マイナス**（-100に近づく）
- 後ろ（手前）に引く → **プラス**（100に近づく）
- 中央（何も触らない） → **0**

前に倒したときに前進させたいので、**-1** を掛けて**符号だけ反転**させ、DCモータの速度にします。

製造元: 株式会社ロボットスポーツゲームズ / 問い合わせ: info@robotic-games.com

改訂履歴

文書バージョン	発行	おもな変更
v1.9	2026年10月	ブロック一覧見表にブレーキ・脱力・トリガーの押し込み量を追加。操作パネルのブレーキ中の表示と「ずっと繰り返す」の説明を修正。用語の補足を括弧書きに変更
v1.8	2026年9月	画面の文言を現在のスタジオに合わせて修正（「内蔵スタジオ」）。設定パネルから「アイドル時のLEDアニメーション」を削除。エンコーダの番号が範囲外るときエラーで止まることを追記
v1.7	2026年9月	画面の文言・ブロック名・動作の説明を実装に合わせて修正。接続時のブラウザの許可、ファームウェアの版の確認、サンプルと本体に保存されたプログラムの読み込みを追記。機種の違い・本体側のトラブル対処は「ポコロボ 本体セットアップ・安全ガイド」への参照に統一
v1.6	2026年9月	パソコンとつなぐケーブルの呼び名を「USB Type-A to Type-Cケーブル」に統一（選び方は「ポコロボ 本体セットアップ・安全ガイド」へ集約）
v1.5	2026年8月	機種の呼び名から「版」を外して「スタンダード」「ミニ」に統一。 http://192.168.20.1 でスタジオを開く方法の呼称を「内蔵スタジオ」に変更。冒頭の「対応スタジオ」「対応本体ファームウェア」の行を削除
v1.4	2026年8月	「プログラム開始からの時間（秒）」ブロックの説明と、前回からの経過時間の求め方を追加
v1.3	2026年8月	版のちがい（サーボ・LED）を比較表に追記。DCモータが動かないときの説明を修正。参照先の表記を書名に統一
v1.2	2026年4月	ドキュメントを現在の構成に再設計