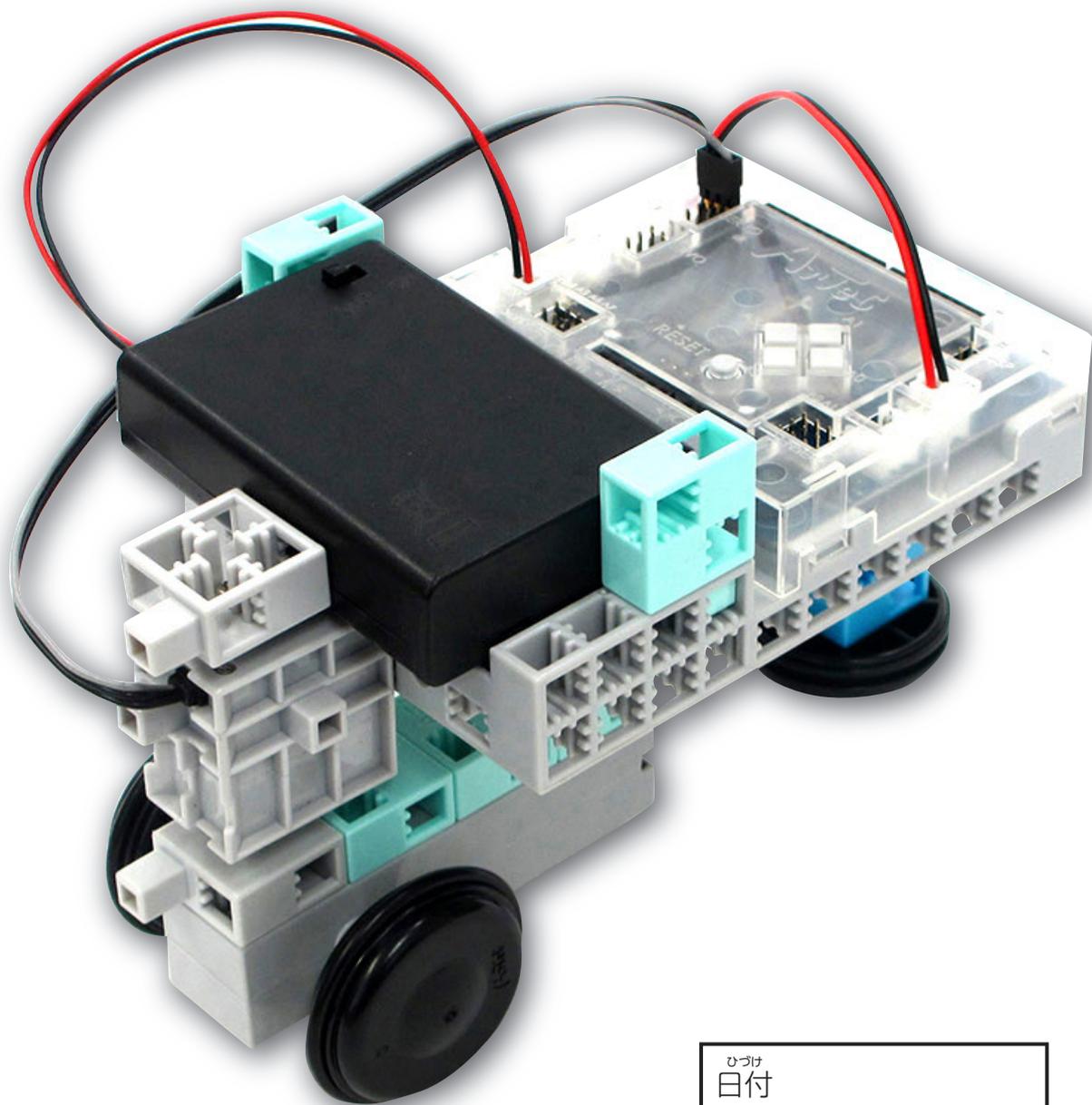


しょう がっ こう よう

小学校用テキスト



ひつげ 日付	
がっ 月	にち 日

なまえ

1時間目

- 使用するパーツ 2
- + 身の回りのロボット 3
 - ロボットってなんだろう? 3
 - ロボットはどうやって動いているんだろう? 4
- + プログラムを作ってモーターを動かそう 4
 - プログラムを作るためのソフトウェアの使い方 4
 - DCモーターを動かそう 8
- + ロボットカーのプログラムをつくろう 11
 - ロボットカーを組み立てよう 11
 - ロボットカーを走らせよう 13
 - 走行時間を決めてロボットカーを走らせよう 14
 - ロボットカーを自由に走らせよう 14

2時間目

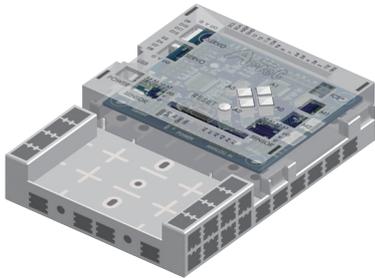
- 使用するパーツ 15
- + ロボットカーを自由に動かそう 16
 - サーボモーターの制御 16
 - ロボットカーを25cmまっすぐ走らせよう 18
 - ロボットカーを旋回させよう 19
 - ロボットカーを左に90度曲がらせよう 20
 - ロボットカーを右に90度曲がらせよう 20
- 話し合おう ■ 考えよう 21

3時間目

- 使用するパーツ 22
- + ロボットカーの動きを組み合わせよう 23
 - まっすぐ走る、左右に曲がるプログラムを組み合わせよう 23
 - コースにチャレンジ! 24
- 発表しよう 24
 - つまづいたときに確認しよう 25

じかん め
1時間目

しよう
使用するパーツ



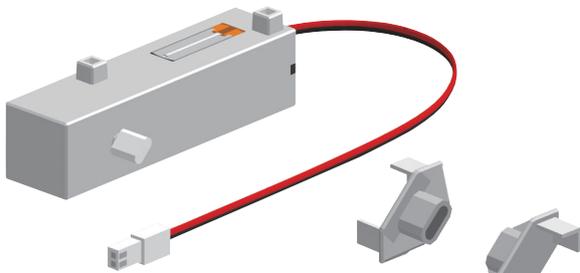
基板本体 ×1



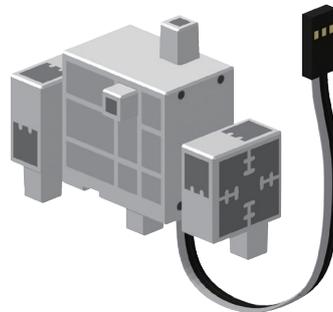
電池ボックス ×1



ユーエスピー USBケーブル ×1



DCモーター ×1



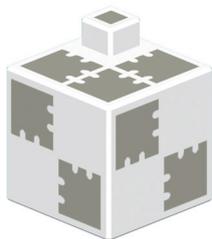
サーボモーター ×1



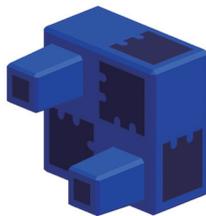
タイヤ ×3



タイヤゴム ×2



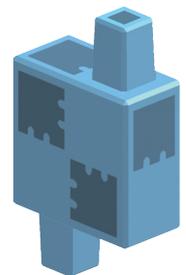
ブロック 白 ×1



ブロック 青 ×1



ブロック 薄水 ×5



ブロック 水 ×2

ちゆうい ディーシー むり やり まわ
注意：DC モーターやサーボモーターを無理矢理回さないようにしましょう。



身の回りのロボット

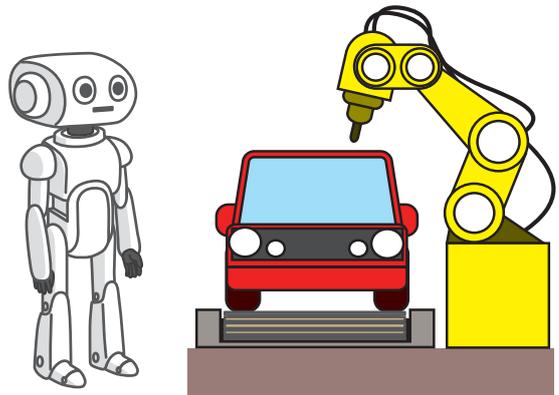


●ロボットってなんだろう？

みんなの周りまわにあるロボットはどんな形かたちをしていて、どんなことことができて、どんな風ふうにみんなの役やくに立たっているのでしょうか。

同じロボットといっても、使つかわれる目的もくてきが違ちがえば、できることことも違ちがうし見た目めも大おおきく変かわります。

でもどんなロボットにも必かならず共きょう通つうした特とく徴ちょうがあります。それは「コンピュータ」「出力装置しゅつりょくそうち」の2つの部ぶ品ひんが組くみ込こまれているということです。



ロボットの「コンピュータ」「出力装置」には人間の体と同じような働きをするものがたくさんあります。

コンピュータ

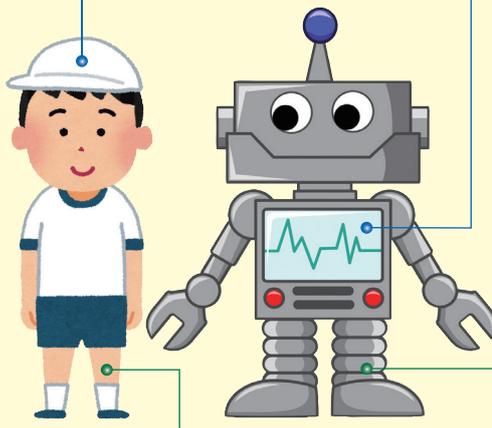
例

パーソナルコンピュータ

マイクロコンピュータ

スーパーコンピュータ

人間の頭脳ずのうにあたる部分ぶぶんで、情報じょうほうを処理しりしたり、出力装置しゅつりょくそうちを動かうごかす命令めいれいを出だします。



出力装置

例

サーボモーター DCモーター

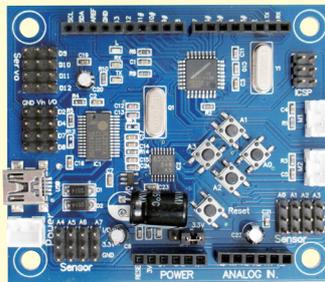
LED ブザー

人間の手てや足あしにあたる部分ぶぶんです。モーターなどコンピュータからの命令めいれいをもとに実際じっさいに動うごくものは「アクチュエーター」と呼よべられます。モーター以外いがいにもLEDやブザーなど音おとや光ひかりなどを出だすものは「インジケータ」などと呼ばれます。

授業じゅぎょうで使つかう部ぶ品ひんにあてはめると…

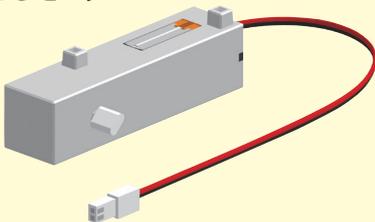
コンピュータ

スタディーノ



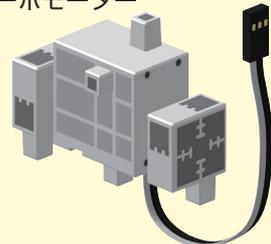
出力装置

ディーシー DCモーター



出力装置

サーボモーター



動き方を伝える

●ロボットはどうやって動いているんだろう？

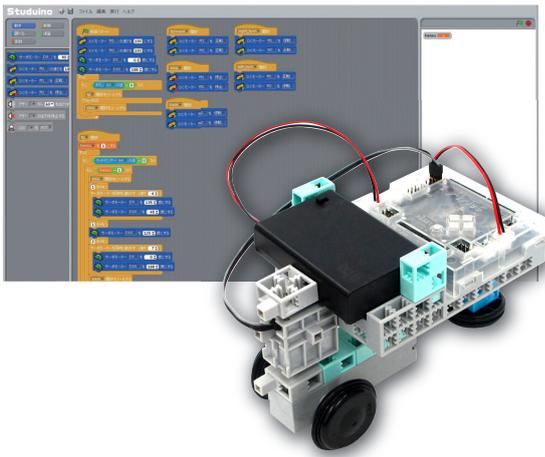
ロボットや電気製品の多くはコンピュータがさまざまな動きの命令を出すことによって動いています。しかし、コンピュータが「どんなときに、どういった命令を出すか」といった手順は、あらかじめ人間によって覚えさせる必要があります。この手順を示したものをプログラムといい、そのために使われる言語をプログラム言語、プログラムを作ることをプログラミングといいます。

```
f ifNil: |
  "Turn setting language"
  self setLanguage: ltmp.
  + self!.

"Head and Footer file"
Default English.
headCode + StandardFileStream new open: 'code/head-en.txt' forWrite: false.
footCode + StandardFileStream new open: 'code/foot-en.txt' forWrite: false.

((ltmp = 'ja') | (ltmp = 'ja+HIRA')) ifTrue: |
  Transcript show: '----- Trans to Japan -----'. cr.
  headCode + StandardFileStream new open: 'code/head.txt' forWrite: false.
  footCode + StandardFileStream new open: 'code/foot.txt' forWrite: false.
!
```

▲プログラミング言語で書かれたプログラム



プログラミング言語で書かれたプログラムを見るとむずかしそうに感じるかもしれませんが、だれでもかんたんにプログラミングができるソフトウェアを使うので心配はいりません。プログラムを作り、思い通りにロボットを動かす楽しさを体感しましょう。

＋ プログラムを作ってモーターを動かそう

●プログラムを作るためのソフトウェアの使い方

ロボットを動かすためのプログラムを作るソフトウェアの使い方を覚えましょう。

1 ブロックプログラミング環境の起動

①デスクトップにある右図のショートカットアイコンをダブルクリックすると、ソフトウェアが起動します。

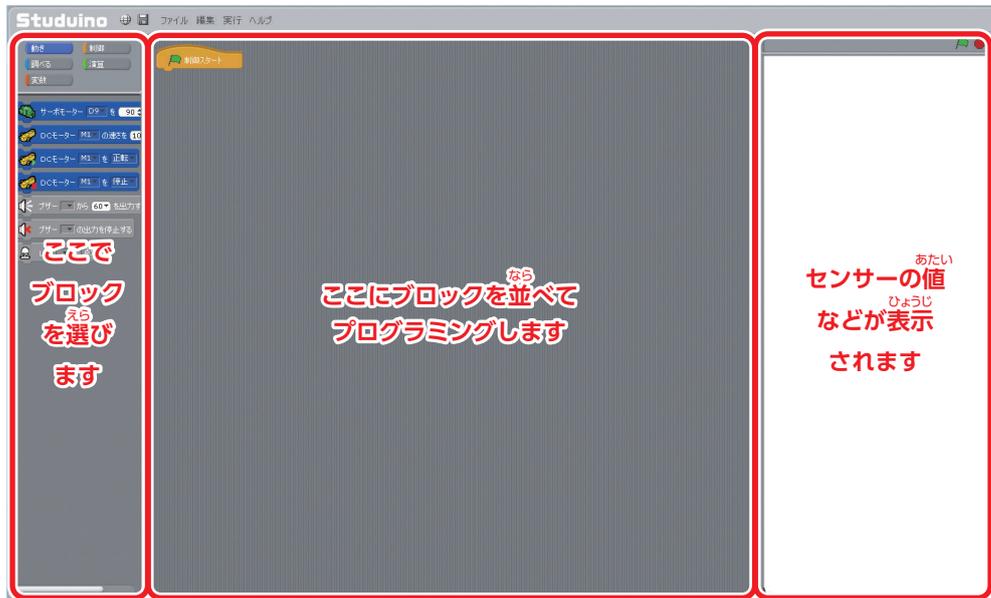


②ソフトウェアを起動すると、右の画面が表示されます。授業では、ブロックプログラミング環境を使います。選択して起動してください。



2 ブロックプログラミング環境の使い方

このソフトウェアでは、さまざまな命令をブロックのように並べてつなぎあわせてプログラムを作ることができます。



3 マウスの使い方の確認

クリック (左クリック)	右クリック	ドラッグ
 ●メニューを選ぶ ●ブロックを選ぶ	 ●ブロックの複製 ●ブロックの削除	 ●選んだブロックを動かす

※ここから先は、左クリックを「**クリック**」、右クリックを「**右クリック**」と表記します。

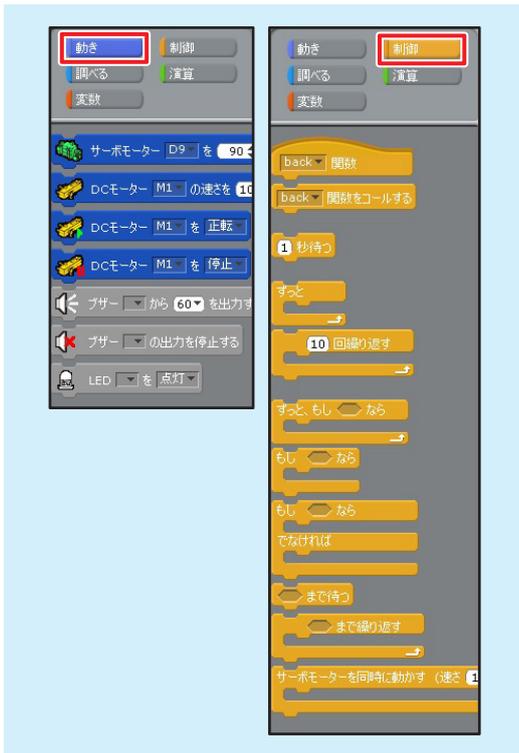
4 使用するブロックの選び方

左上の5つのボタンをおすと、選べるブロックが切りかわります。

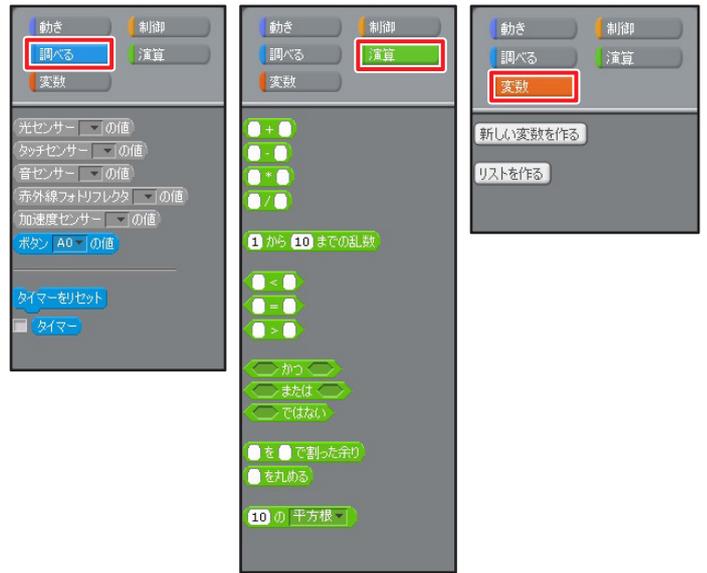
それぞれのブロックの使い方は、毎回のテーマにそって説明をしていきます。ここではたくさん種類があることを確認しましょう。



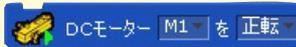
これから授業で使うブロック



授業では使わないブロック



動き … LED、電子ブザー、DC モーター、サーボモーターなど出力装置を動かすためのブロックが用意されています。今回は 3 種類のブロックを使います。



制御 … プログラムの処理に関する色々なブロックが用意されています。今回は 1 種類のブロックを使います。

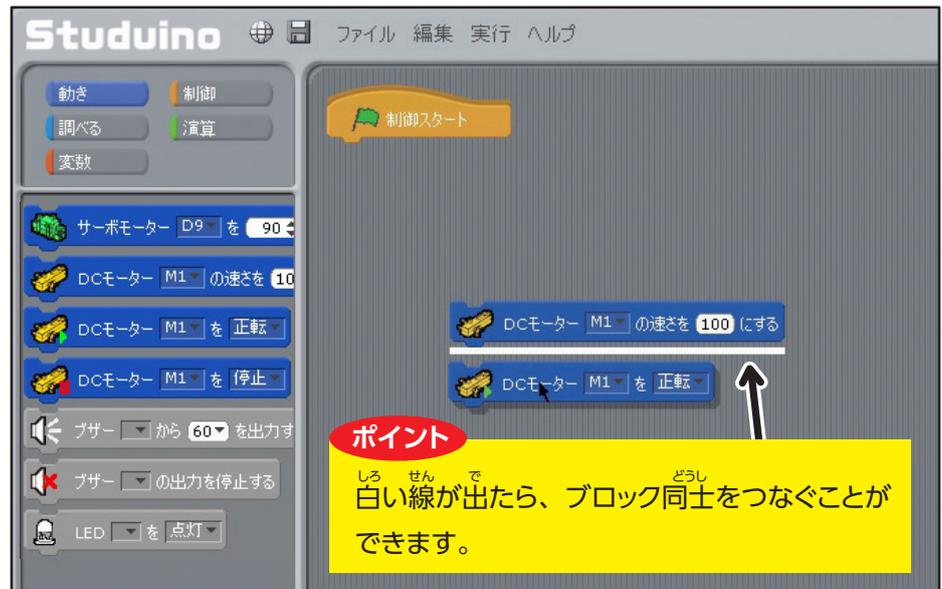


5 プログラムの並べ方

選んだブロックをマウスでクリックしながら、真ん中の画面の上にドラッグし、クリックをはなすと、ブロックを並べることができます。



すでに並べられているブロックの下に別のブロックをドラッグすると、右図のように白い線が現れて、ブロック同士をつなげることができます。

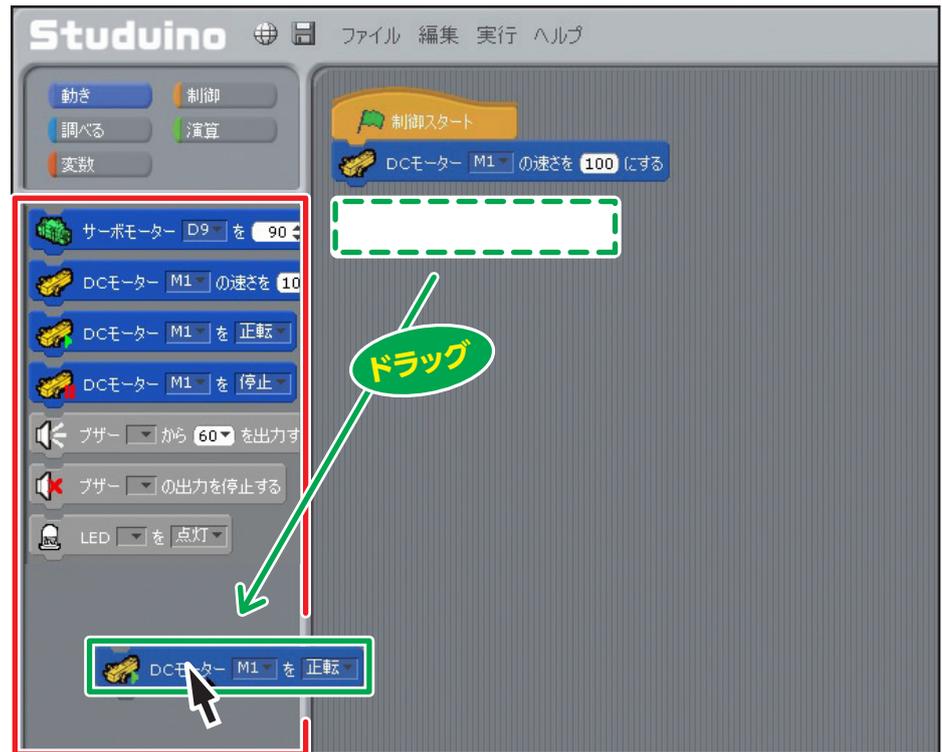


並べられたブロックをドラッグすることで、移動させることができます。その時、下につながっているブロックもいっしょに動きます。



6 ブロックの消し方

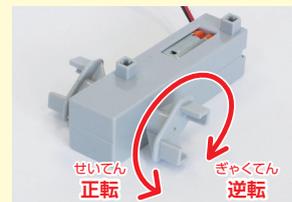
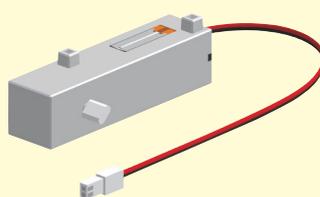
赤い線の中のブロックを選ぶスペースに、消したいブロックをドラッグしてクリックをはなすとブロックを消すことができます。



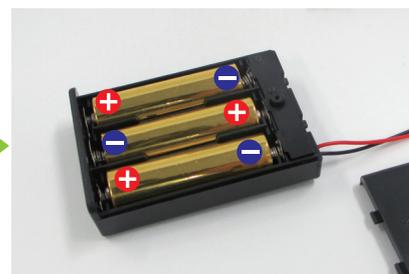
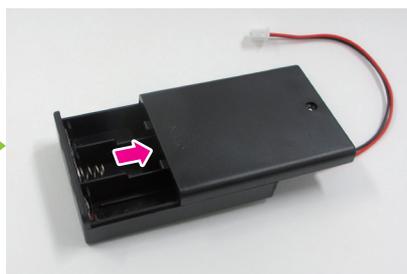
●DCモーターを動かそう

DCモーターってなに？

DC モーターは電気を流すと回る電気部品です。DC モーターは電気を流す向きによって回る向きを変えることができます。また、流す電気の量によって回る速さも変えることができます。



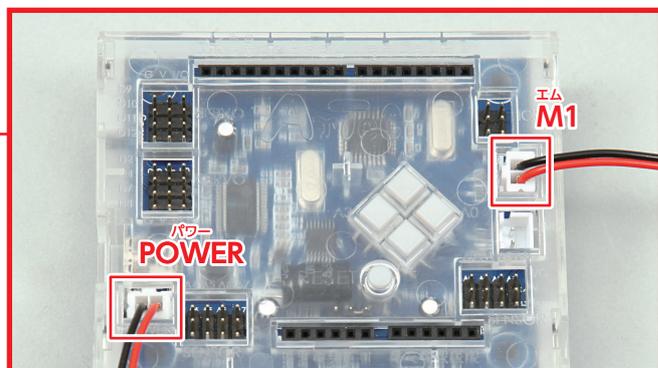
DC モーターを動かすために、電池ボックスに電池（単3、アルカリ）を入れましょう。



⚠️ **+** プラス **-** マイナスの向きに注意しましょう

1 準備をしよう

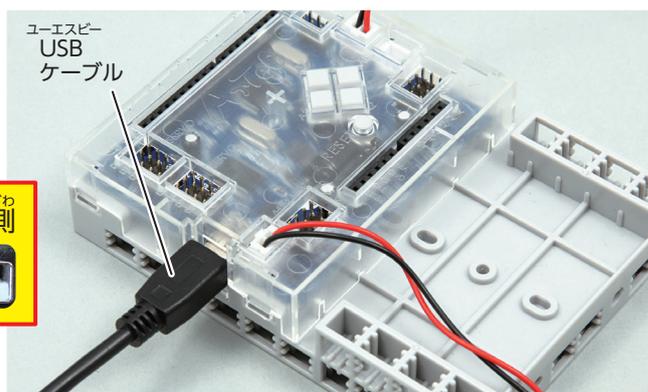
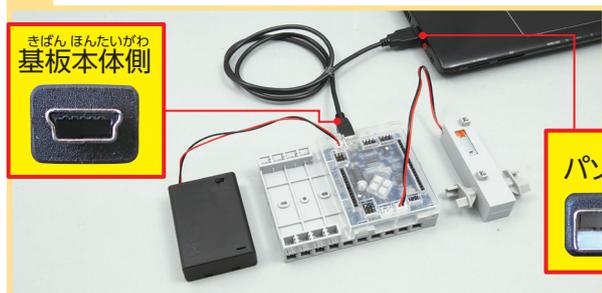
1 電池ボックスを基板本体の **POWER** に、DC モーターを **M1** にとりつけます。



⚠️ **コネクタの向きに注意**

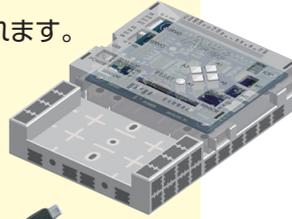
コネクタの向きに注意しましょう。**赤色・黒色のコード**は写真と同じ向きに接続します。

2 ユーエスピー USB ケーブルで基板本体とパソコンをつなぎます。



⚠️ **コネクタの向きに注意**

「ブロックプログラミング環境」で作ったプログラムは、
基板^{かんばん}上のコンピュータ^{コンピューター}が理解^{りかい}できる言葉^{ことば}にパソコン上^{じょう}
で変換^{へんかん}されて、USBケーブルを通して転送^{てんそう}されます。



```
// -----  
void artecRobotSetup() {  
  board.InitDCMotorPort(PORT_M1);  
  board.InitServomotorPort(PORT_D9);  
  board.InitSensorPort(PORT_A0, FIDLED);  
  board.InitSensorPort(PORT_A1, FIDBUZZER);  
  board.SetServomotorCalibration(SvCalibrationData);  
  board.SetDCMotorCalibration(DCCalibrationData);  
}  
  
// -----  
// Artec robot mainroutine  
// -----  
void artecRobotMain() {  
  board.LED(PORT_A0, ON);  
  board.Timer(1);  
  board.LED(PORT_A0, OFF);  
  BUZZER_START(PORT_A1, 62);  
  board.Timer(1);  
  BUZZER_STOP(PORT_A1);  
  DCMOTOR_POWER(PORT_M1, 100);  
  board.DCMotorControl(PORT_M1, NORMAL);  
  board.Timer(3);  
  DCMOTOR_STOP(PORT_M1, BRAKE);  
  board.Servomotor(PORT_D9, SVRANGE(90));  
}
```

2 プログラミングしてDCモーターを動かそう

① 画面上の**実行**をクリックし、**テストモード開始**を選びましょう。



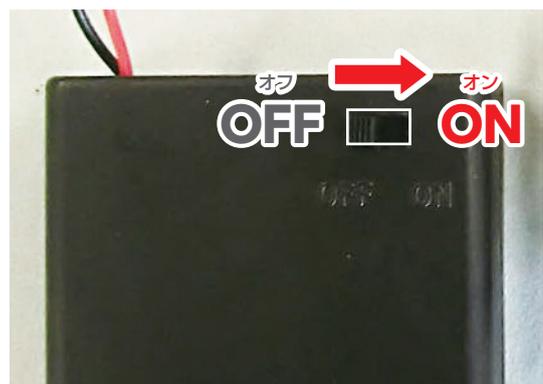
コンピュータに DC モーターを動かす命令を出すときは、次の 3 つのブロックを使います。

	……DC モーターの回る速さを決める。
	……回る向きを決めて DC モーターを回す。
	……DC モーターの回転を止める。

② 図のようにブロックをドラッグして並べましょう。



③ 電池ボックスのスイッチを ON にしましょう。

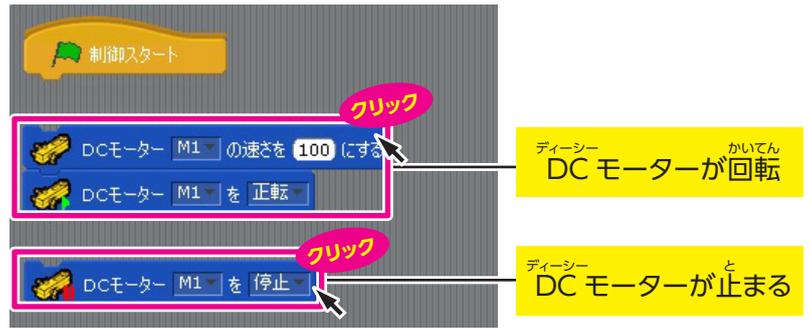


- ④ DCモーター M1 の速さを 100 にする
 DCモーター M1 を 正転

をクリックすると、DCモーターが回転します。

- DCモーター M1 を 停止

をクリックすると、DCモーターが止まります。



DCモーターの止め方

DCモーターを止める時は「停止」と「解放」の2種類の止め方が選べます。
 自転車に例えると、「停止」がブレーキをかけて止める、「解放」がこがずに自然に止まるのを待つことに似ています。

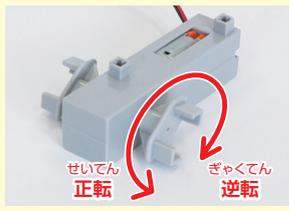
「停止」



「解放」



- ⑤ DCモーター M1 の速さを 50 にする
 DCモーター M1 を 逆転
- の 部分の 数値 を変えると回転の速さが変わり(最大 100)、
 部分を 逆転 に変えると、反対方向に回転します。回転の速さや回転方向を自由に変えて動かしてみましょう。



DCモーターの速さは 0 ~ 100 の間の数字で設定できますが、20 より小さい数字を設定するとDCモーターに流れる電気が足りなくなって動かなくなります。

DCモーターの動きを確認できたら、USBケーブルを抜く前に、画面上の 実行 をクリックし、テストモード終了を選び、テストモードを終了させてください。



基板本体につないだ DCモーターと電池ボックスを一度はずしておきましょう。

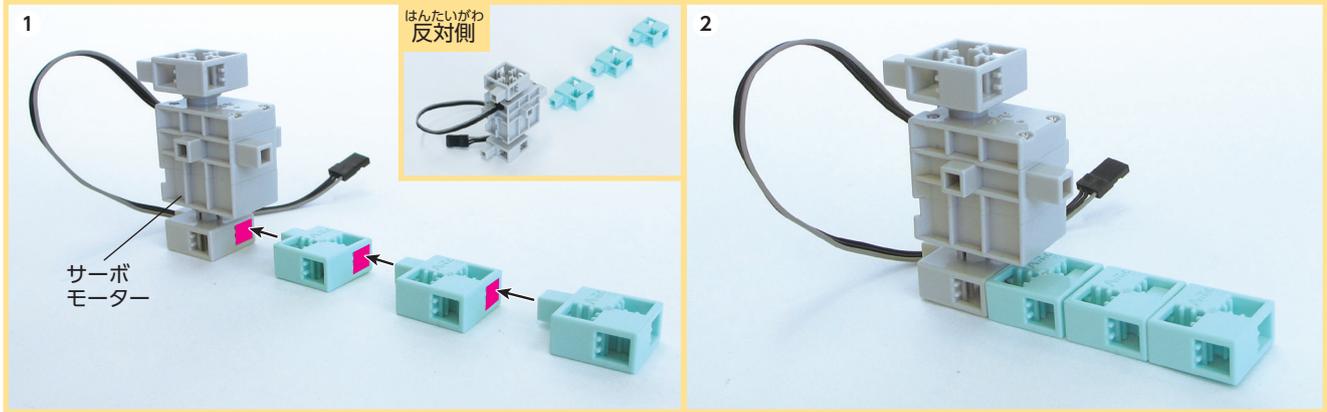


ロボットカーのプログラムをつくろう

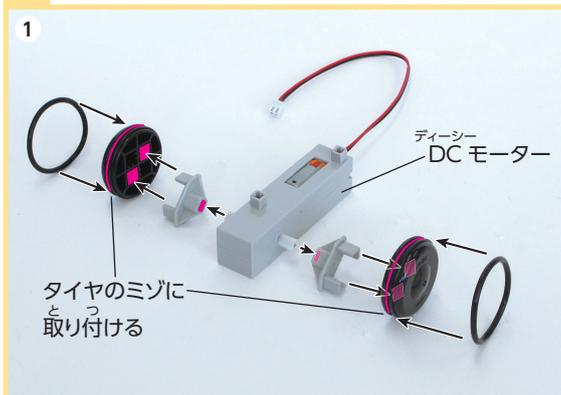


●ロボットカーを組み立てよう

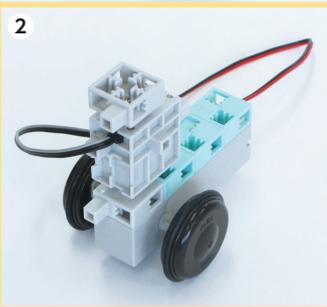
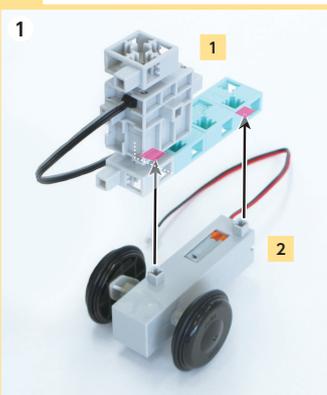
1 サーボモーターにブロックを取り付けます。



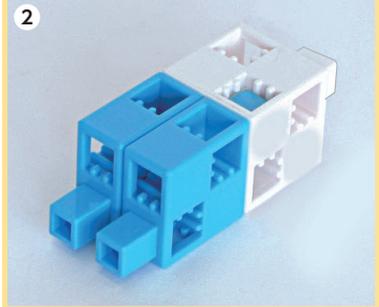
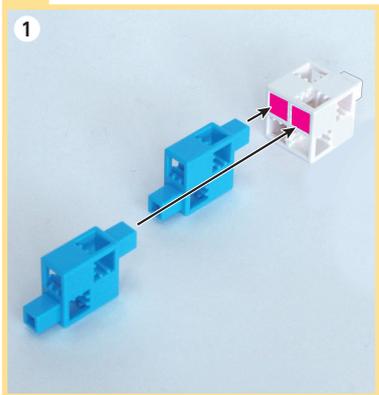
2 タイヤを組み立て、DCモーターに取り付けます。



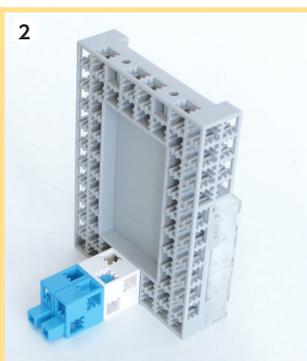
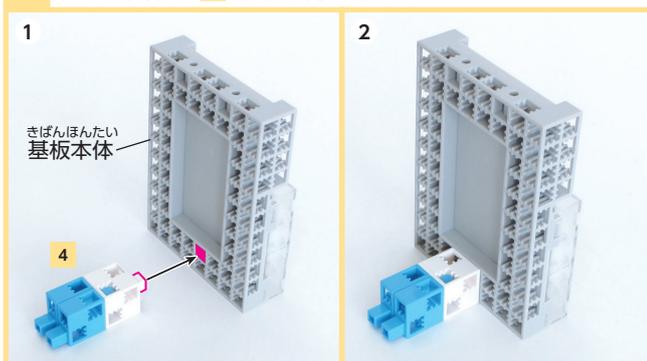
3 1 と 2 を合体します。



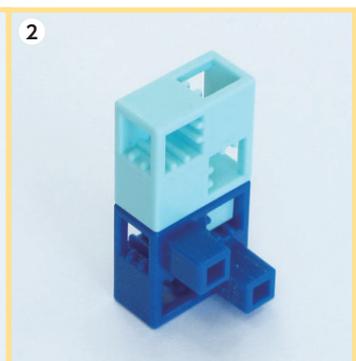
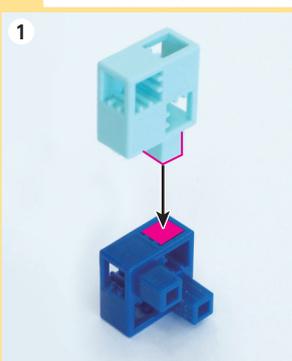
4 ブロックを組み立てます。



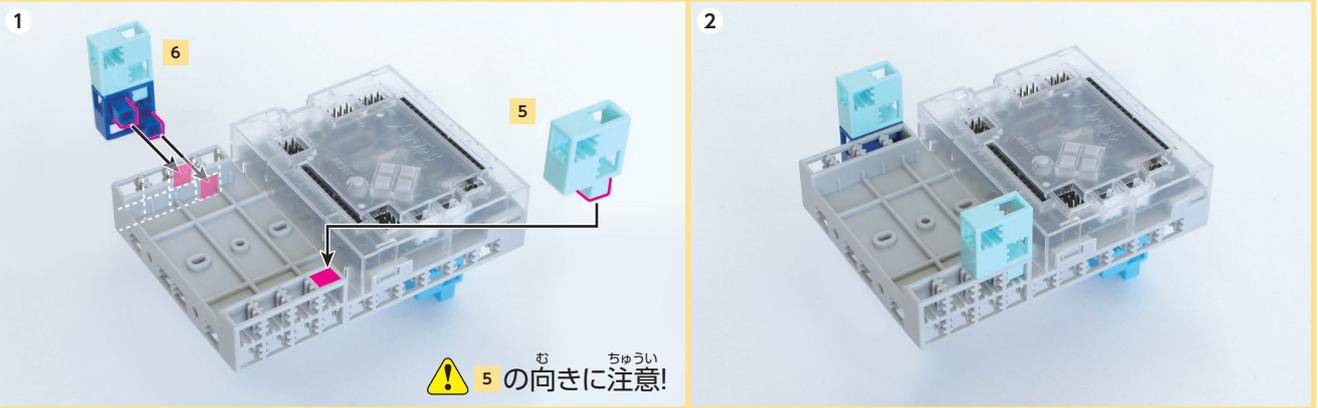
5 きばんほんたい 基板本体に 4 を取り付けます。



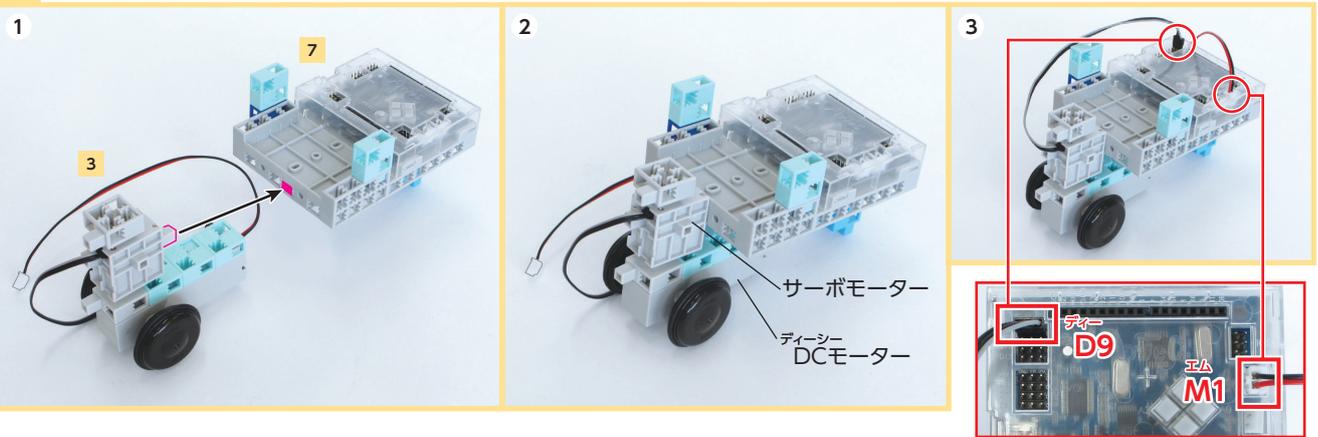
6 ブロックを組み立てます。



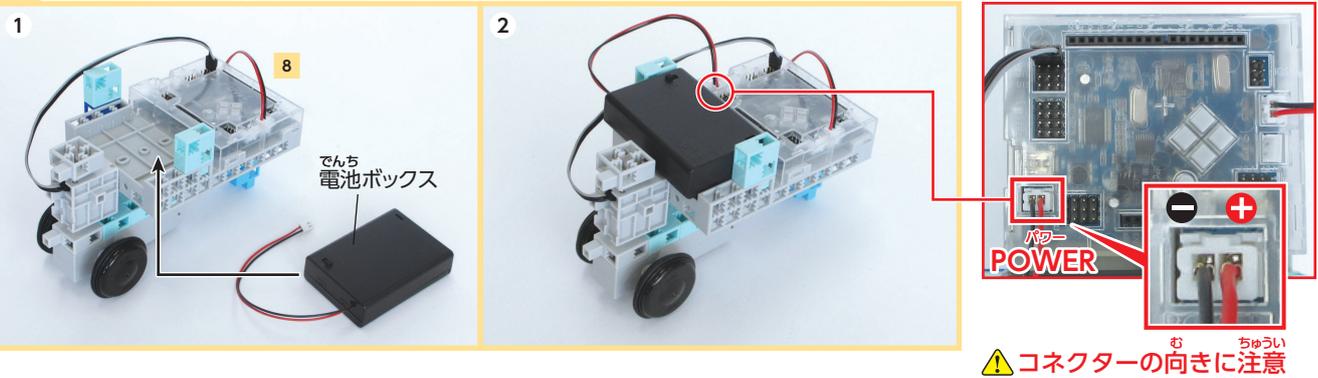
7 5と6を合体し、さらにブロックを取り付けます。



8 3と7を合体し、サーボモーターをD9に、DCモーターをM1に接続します。

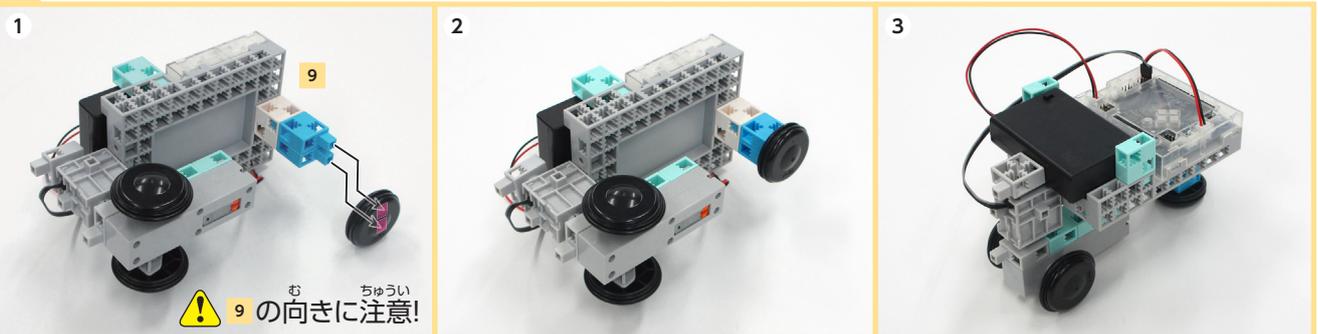


9 8に電池ボックスをセットし、POWERに接続します。



コネクタの向きに注意しましょう。赤色・黒色・灰色のコードは写真と同じ向きに接続します。

10 9にタイヤを取り付けて完成です。

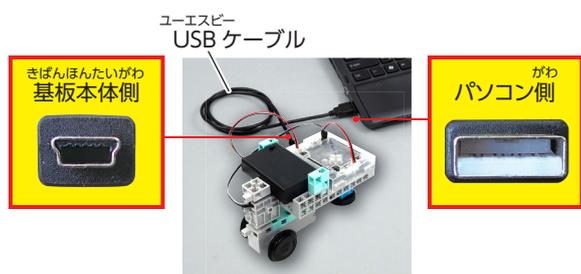


●ロボットカーを走らせよう

①右図のようにブロックをドラッグし、並べましょう。



②USB ケーブルで基板本体とパソコンをつなぎましょう。

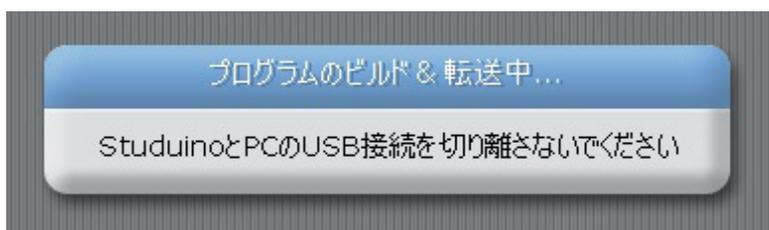


⚠ コネクタの向きに注意

③画面上の**実行**をクリックし、**プログラム作成・転送**を選びましょう。



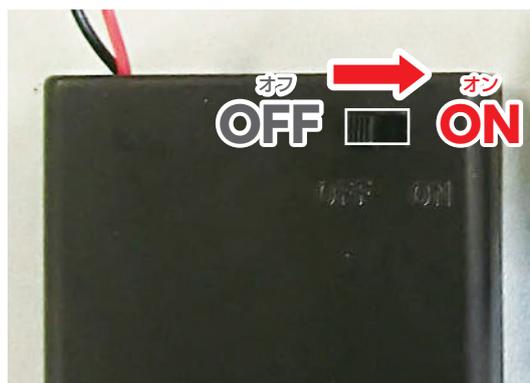
下の画面が表示されます。表示が消えるまで USB ケーブルを抜かずに待ちましょう。



④電池ボックスのスイッチを**ON**にして、ロボットカーを走らせてみましょう。

ロボットカーを停止したい時は電池ボックスのスイッチを**OFF**にします。

※ロボットカーを走らせる時以外は、電池ボックスのスイッチは**OFF**にしておきましょう。



「プログラム作成・転送」と「テストモード」の違い

「プログラム作成・転送」はプログラムを基板に転送するので、パソコンと USB ケーブルでつながっていても、ロボットを動かさずとも、「テストモード」はパソコンと USB ケーブルでつながっていないと動きませんが、作ったプログラムをすぐに試すことができます。

● 走行時間を決めてロボットカーを走らせよう

走行時間を決めるときは次のブロックを使います。

1 秒待つ …………… 指定した時間だけ次のブロックの処理を待つ。

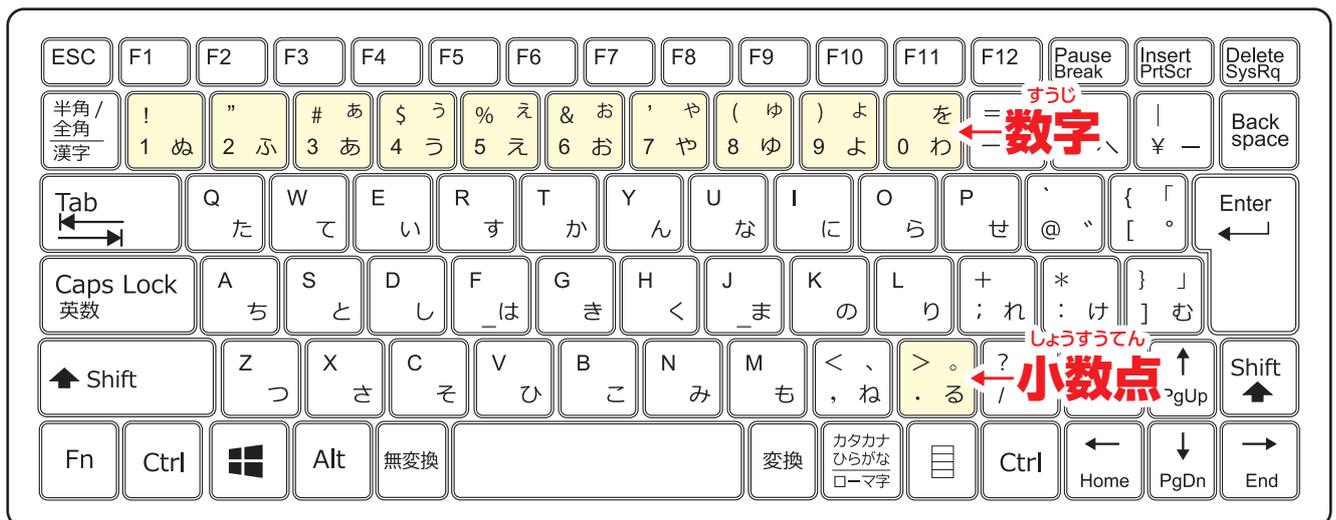
下の図のようにブロックをドラッグし、並べましょう。



1 秒待つ の数値を変更することで、ロボットカーの走行時間を設定することができます。

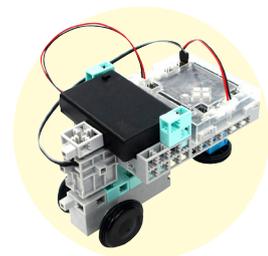
また、時間は小数も設定することができるので、自由に変えて動かしてみましょう。

数字と小数点の場所



● ロボットカーを自由に走らせよう

これまで学習してきたことを使って、ロボットカーを自由に走らせてみましょう。

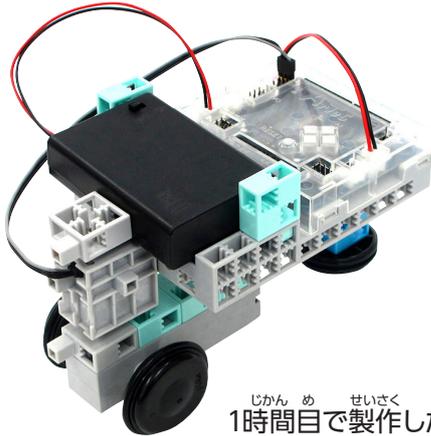




じかんめ 2時間目



しよう 使用するパーツ



じかんめ せいさく くるま
1時間目で製作した車 ×1



ユーエスピー
USBケーブル ×1

ちゅうい ディーシー
注意：DC モーターやサーボモーターを無理矢理回さないようにしましょう。
むり やり まわ



ロボットカーを自由に動かそう



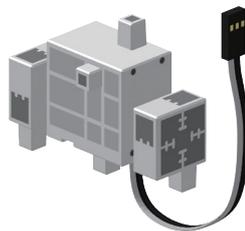
●サーボモーターの制御

組み立てたロボットカーの進行方向を変えるためには、ロボットカーをどのように制御すればよいでしょうか。私たちがふだん乗っている自動車はハンドルを回すことでタイヤの向きを変え、進行方向を変えます。また自転車も体をかたむけたり、ハンドルを操作しタイヤの向きを変えることで進行方向を変えます。

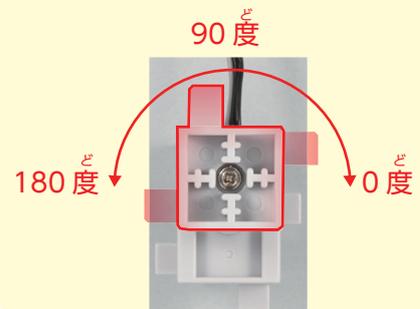
ロボットカーもサーボモーターを使用することで、自転車のようにタイヤ（DC モーター）の向きを変え、進行方向を変えることができます。

サーボモーターってなに？

サーボモーターは DC モーターと同じ電気を流すと回る電気部品ですが1つちがうところがあります。DC モーターが速さや向きを決めて回るのに対し、サーボモーターはねらった角度まで回すことができます。



使用するサーボモーターでは 0～180度の範囲でねらった角度まで回すことができます。



1 プログラミングしてサーボモーターを動かそう

サーボモーターの角度を制御してタイヤの向きを変えてみましょう。60度⇒90度⇒120度の順番に動かすプログラムを作ります。

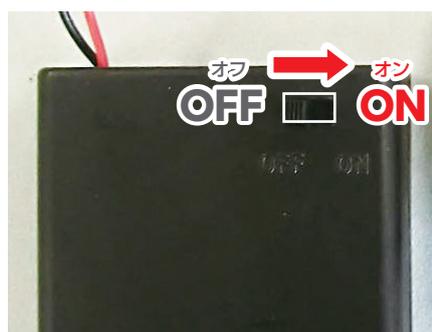
① USB ケーブルで基板本体とパソコンをつなぎます。



② 画面上の「実行」をクリックし、「テストモード開始」を選択します。



③ 電池ボックスのスイッチを ON にしましょう。



サーボモーターの特性

電池ボックスのスイッチを入れると、プログラムによる指示がなければ自動で 90 度の方向に向きます。

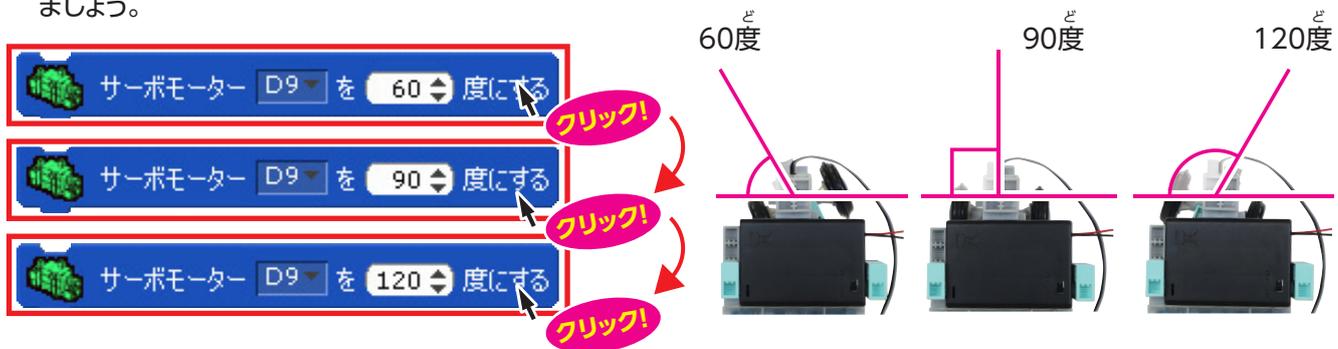
- ④ **動き** から **サーボモーター D9** を **90度にする** をドラッグし、3つのブロックどうしがくっつかないようにおきましょう。その後、それぞれのブロックの に60、90、120と角度の数字を入力します。



サーボモーターは0度から180度まで1度ずつ変更できます。

ここでサーボモーターの角度を決めます。
上から60、90、120と入力してみましょう。

- ⑤ (60度にする)⇒(90度にする)⇒(120度にする)の順にクリックして、サーボモーターの角度が変わることを確認しましょう。



記入しよう

サーボモーターの角度とタイヤの向きとの関係

サーボモーターの角度が90度より小さい時、タイヤは(左 ・ 右)を向く
 サーボモーターの角度が90度より大きい時、タイヤは(左 ・ 右)を向く

2 サervoモーターを順番に動かそう

- ①サーボモーターのブロックを60度⇒90度⇒120度の順につなぎましょう。
 つないだら、一番上の **サーボモーター D9** を **60度にする** をクリックし、プログラムを実行します。サーボモーターは60度⇒90度⇒120度の順に動くでしょうか？

つながったブロックをクリックすると、上から順番にブロックが実行されます。



①のプログラムでは、思った通りにロボットが動作しません。なぜでしょうか？

コンピュータはひとつひとつのブロックをととても短い時間で処理できます。

①のプログラムではサーボモーターの角度が変わる前に次のブロックが処理されてしまうので、最後につながったブロックだけが実行されたかのような動作をしてしまうのです。

②各ブロックの間に「制御」から「1秒待つ」をドラッグして、つなぎましょう。
つないだら、一番上の「サーボモーター D9 を 60 度にする」をクリックし、プログラムが上から順番に実行されることを確認しましょう。



●ロボットカーを25cmまっすぐ走らせよう

走行時間を調整して、ロボットカーがまっすぐ 25cm 走るようにプログラミングしてみましょう。

①下の図のようにプログラムをつくりましょう。

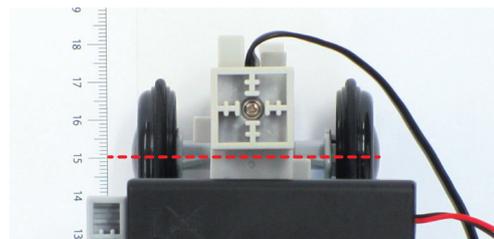
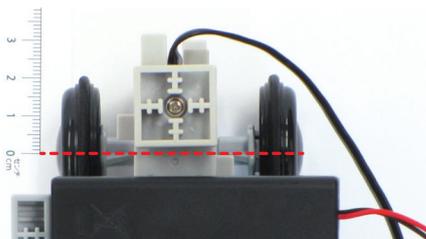
②2 つめの「1秒待つ」の数値を変えて走行時間を調整します。



③プログラムを転送し、USB ケーブルを外します。

④P27 の定規の右側にロボットカーを置き、タイヤの中心を 0cm に合わせます。

⑤電池ボックスのスイッチを ON にしてロボットカーを走らせましょう。止まった時のタイヤの中心位置がロボットカーが走った距離となります。



⑥ ②～⑤をくり返し行い、ロボットカーを 25cm ちょうど走らせてみましょう。

25cm 走ったときの時間
びよう
秒

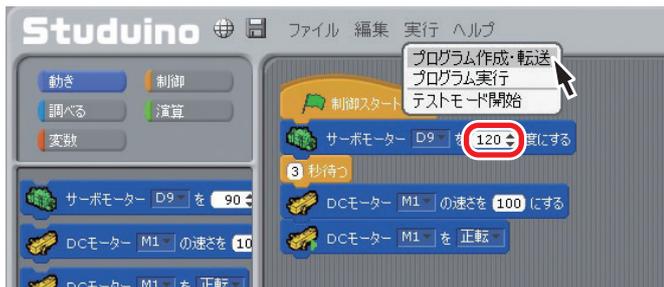
●ロボットカーを旋回させよう

①下の図のようにブロックをドラッグし、並べましょう。

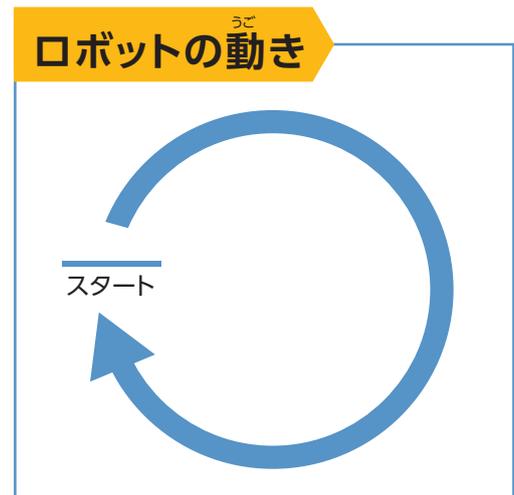


3 秒待つ を入れることで、サーボモーターの角度が変わった後でロボットカーが走り出すようにしています。

②サーボモーターの角度を「120」に変えて、プログラムを転送します。転送終了後 USB ケーブルを外します。



③電池ボックスのスイッチを ON にし、ロボットカーを走らせます。スイッチを入れると、まずサーボモーターが 120 度の方向を向き、3 秒後に DC モーターが回り始めます。スタート位置を決め、1 周するまで走らせましょう。



DC モーターが回転する前に、写真のようにタイヤの先端を決めたスタートの位置に合わせて、走らせましょう。

サーボモーター D9 を 120 度にする の角度を変えることで、ロボットカーが回る円の大きさを変えることができます。サーボモーターの角度を 150 度に変え、120 度の時と比べて回る円の大きさがどのように変わるか確認しましょう。また、サーボモーターの角度を 30 度、60 度に変えて、左回りの時にロボットカーが回る円の大きさがどのように変わるかも確認しましょう。

記入しよう

90度からの角度の差が大きいほど 回る円の大きさは (大きい ・ 小さい)

$$120 \text{ 度} - 90 \text{ 度} = 30 \text{ 度}$$

$$150 \text{ 度} - 90 \text{ 度} = 60 \text{ 度}$$

なので 150 度のほうが 120 度よりも 90 度からの角度の差は大きい

●ロボットカーを左に90度曲がらせよう

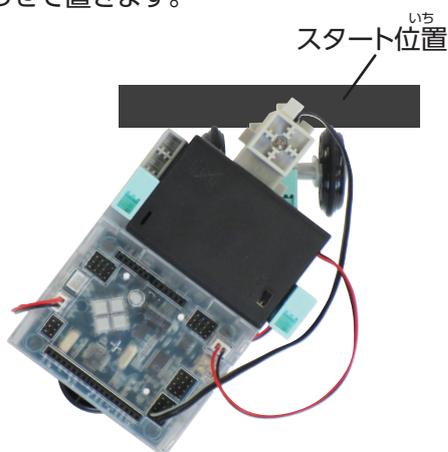
ロボットカーが左に90度曲がるようにプログラミングしてみましょう。

①下の図のプログラムをつくり、曲がる向きに合わせてサーボモーターの角度と走行時間を変えましょう。

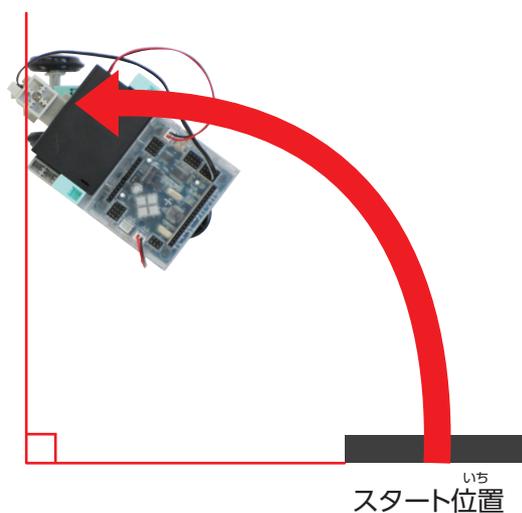


②プログラムを転送し、USBケーブルを外します。

③電池ボックスのスイッチをONにして、サーボモーターが動いたあと、タイヤをスタート位置に合わせて置きます。



④止まったときに90度曲がった位置にタイヤがあるかを確認します。



⑤ロボットカーが左に90度曲がるまで①～④をくり返しましょう。

サーボモーターの角度	90度左に曲がったときの時間
度	秒

●ロボットカーを右に90度曲がらせよう

左のときと同じ手順で、今度はロボットカーが右に90度曲がるようにプログラミングしてみましょう。

サーボモーターの角度	90度右に曲がったときの時間
度	秒

話し合おう

走行時間とロボットカーが走った距離や角度にはどのような関係があるでしょうか？
気づいたことを話し合みましょう。

考えよう

①ロボットカーを 100cm まっすぐ走らせるには、走行時間をおおよそどのくらいにすればよいか考えましょう。

3時間目

使用するパーツ



1時間目で製作した車 ×1



ユーエスピー
USBケーブル ×1

注意：DC モーターやサーボモーターを無理矢理回さないようにしましょう。



ロボットカーの動きを組み合わせよう



●まっすぐ走る、左右に曲がるプログラムを組み合わせよう

前回の授業まで、ロボットカーをまっすぐ走らせる、左右に曲がらせるプログラムをそれぞれ作りました。

ここではそれらをまとめて、次の順番で動くプログラムを作ります。

まっすぐ3秒間走る ⇒ 左に3秒間曲がる ⇒ まっすぐ3秒間走る ⇒ 右に3秒間曲がる ⇒ 止まる

①下の図のようにブロックをドラッグして並べ、数値を変えましょう。



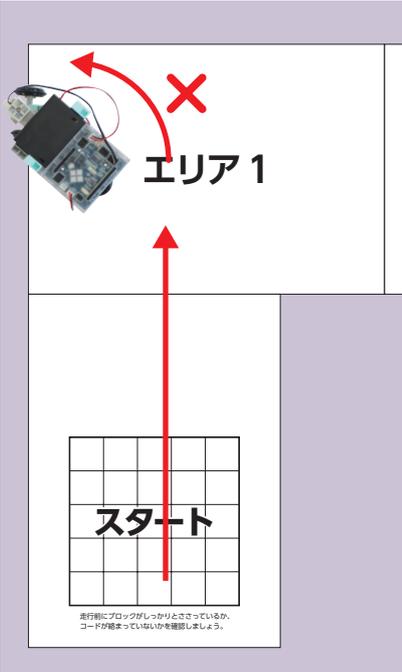
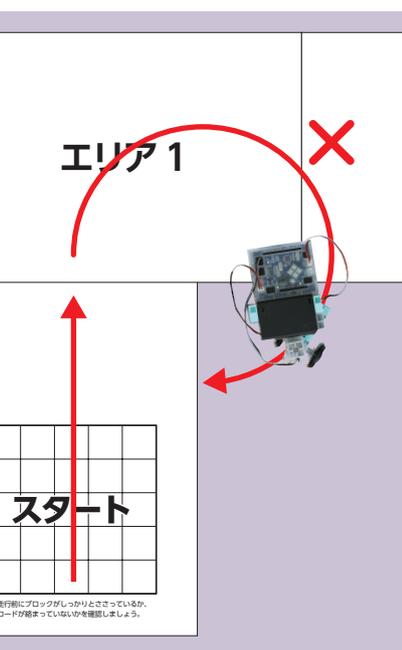
はやむき 速さと向きを決めてモーターを回す
びょうかん 3秒間まっすぐ走らせる
びょうかんひだり 3秒間左に曲がらせる
びょうかん 3秒間まっすぐ走らせる
びょうかんみぎ 3秒間右に曲がらせる
と 止める

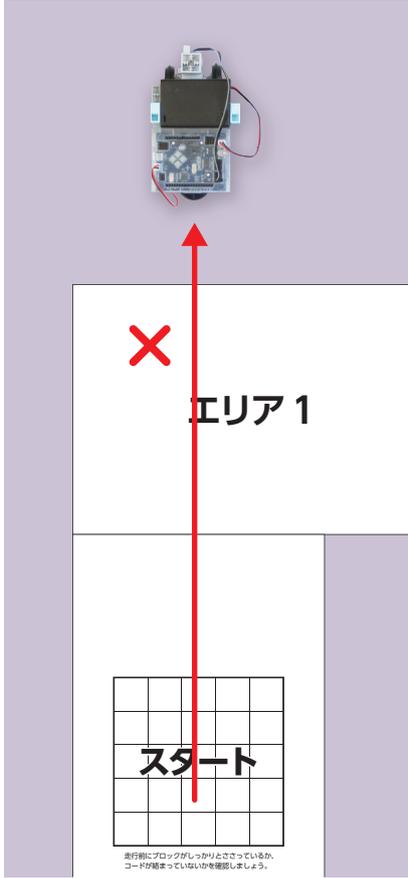
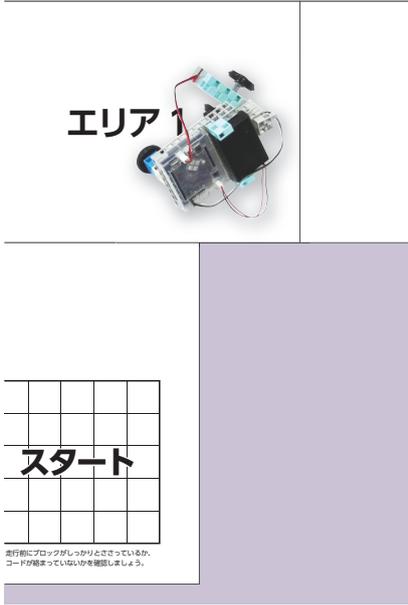
②作成したプログラムを転送し、USBケーブルを外したあと電池ボックスのスイッチをONにしてロボットカーを走らせます。プログラム通りの動きをしているか確認しましょう。



●つまづいたときに確認しよう

課題コースをロボットカーがうまく走らないときは、次のことを確認してプログラムを見直しましょう。

つまづいたこと	つまづいた理由	見直すところ
<p>反対向きに曲がる。</p>  <p>スタート</p> <p><small>進行中にプログラミングがしっかりとできているか、コードが読まれているかを確認しましょう。</small></p>	<p>タイヤが反対の方向を向いている。</p>	<p>テキストの P.17 を振り返り、サーボモーターの角度を見直す。</p>  <p>右や左に曲がるときのサーボモーターの角度はどうなっていたかな？</p>
<p>曲がり続けてコースからはずれる。</p>  <p>スタート</p> <p><small>進行中にプログラミングがしっかりとできているか、コードが読まれているかを確認しましょう。</small></p>	<p>曲がる時間が長い。</p>	<p>テキストの P.20 を振り返り、曲がる時間を見直す。</p>  <p>右や左に曲がる時間はどこで決めていたかな？</p>

つまづいたこと	つまづいた理由	見直すところ
<p>まっすぐに走り続けてコースからはずれる。</p>  <p>進行前にブロックがしっかりとできているか、コードが壊まっていないかを確認しましょう。</p>	<p>まっすぐに走る時間が長い。</p>	<p>テキストの P.18 を振り返り、走る時間を見直す。</p>  <p>まっすぐに走る時間はどこで決めていたかな？</p>
<p>まがる時に倒れる。</p>  <p>進行前にブロックがしっかりとできているか、コードが壊まっていないかを確認しましょう。</p>	<p>タイヤが大きく右や左に向いている。</p>	<p>テキストの P.19 を振り返り、サーボモーターの角度を見直す。</p>  <p>タイヤの向きを小さくするにはサーボモーターの角度をどのように変えればよかったかな？</p>

