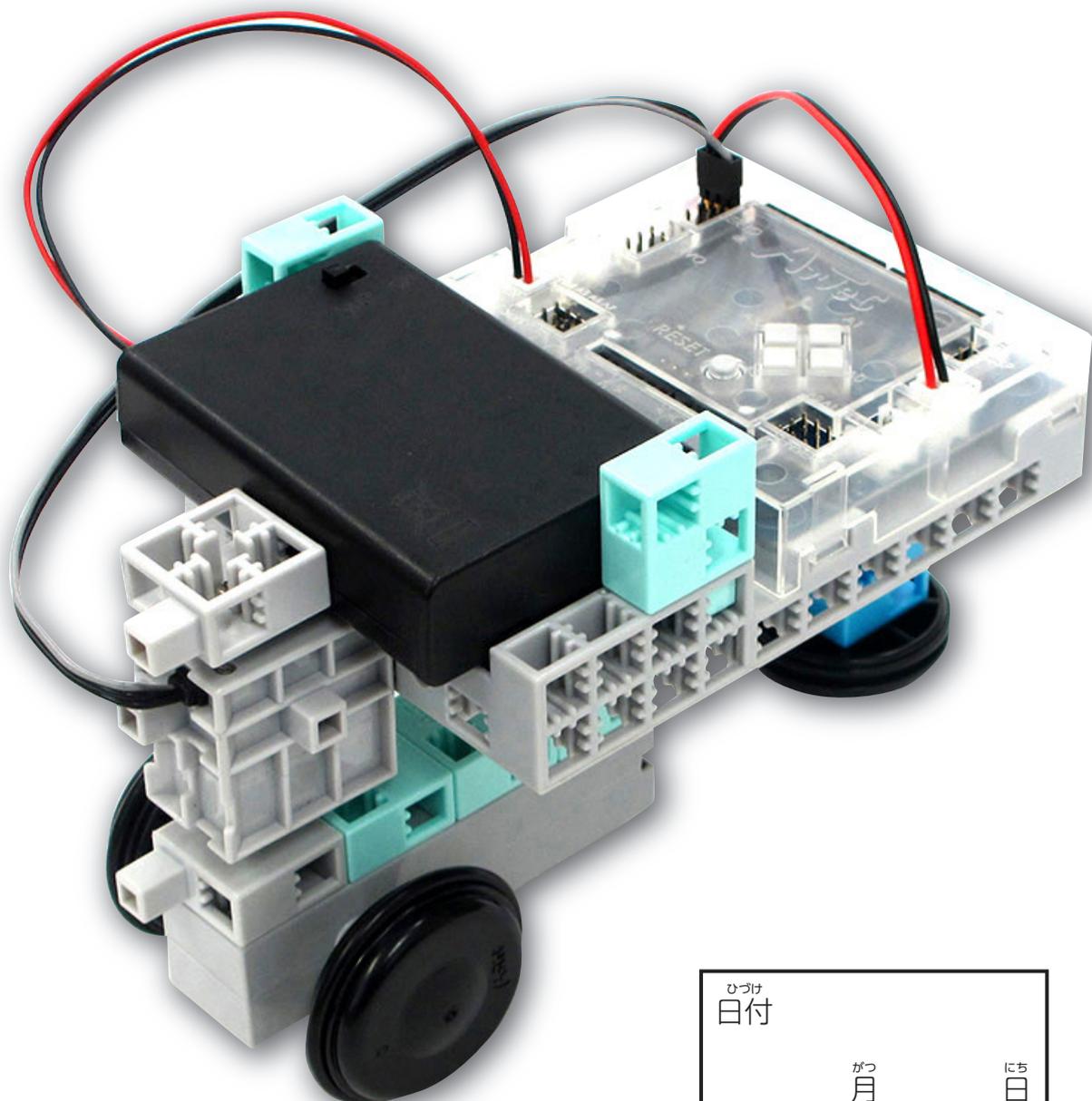


しょう がっ こう よう

小学校用テキスト

教師用指導書



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

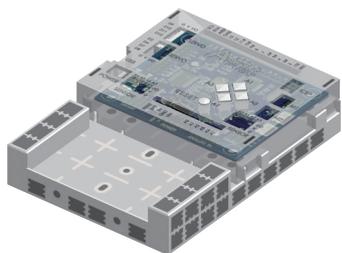
● 使用するパーツ	22
+ ロボットカーの動きを組み合わせよう	23
● まっすぐ走る、左右に曲がるプログラムを組み合わせよう	23
● コースにチャレンジ！	24
■ 発表しよう	24
● つまづいたときに確認しよう	25

■ 巻末資料	27
--------	----

- 授業を開始する前に、コンピュータの起動やキットの配布を済ませておきましょう。
- プログラムしていない空のデータを基板本体に転送し、あらかじめデータを消しておきましょう。

1時間目

使用するパーツ



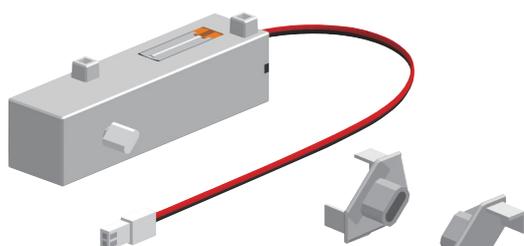
基板本体 ×1



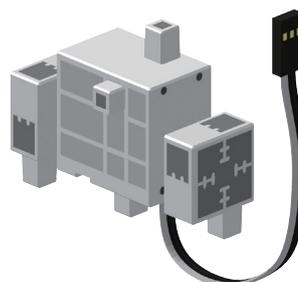
電池ボックス ×1



ユーエスピー USBケーブル ×1



DCモーター ×1



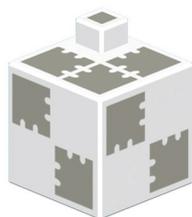
サーボモーター ×1



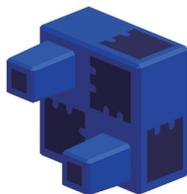
タイヤ ×3



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



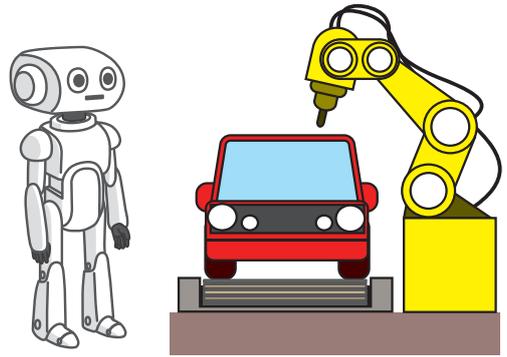
ブロック 水 ×2

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？

みんなの周りにあるロボットはどんな形をしていて、どんなことができ、どんな風にみんなの役に立っているのでしょうか。同じロボットといっても、使われる目的が違えば、できることも違う見た目も大きく変わります。でもどんなロボットにも必ず共通した特徴があります。それは「コンピュータ」「出力装置」の2つの部品が組み込まれているということです。

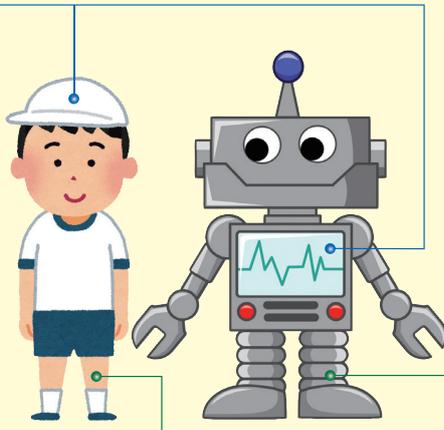


ロボットの「コンピュータ」「出力装置」には人間の体と同じような働きをするものがたくさんあります。

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

人間の頭脳にあたる部分です。情報を処理したり、出力装置を動かす命令を出します。

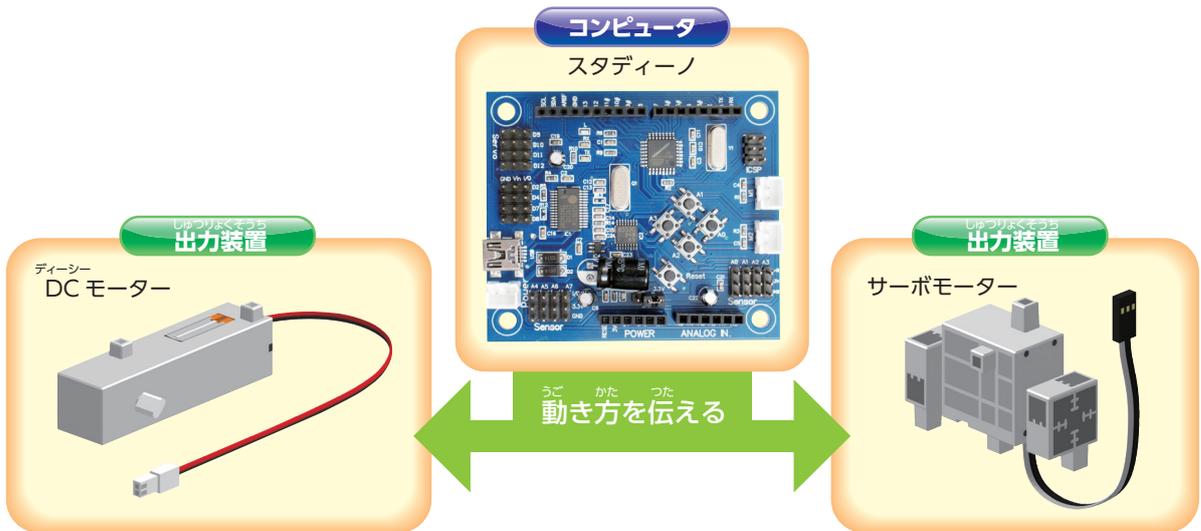


出力装置

- 例
- サーボモーター
 - DC モーター
 - エリイデー LED
 - ブザー

人間の手や足にあたる部分です。モーターなどコンピュータからの命令をもとに実際に動くものは「アクチュエーター」と呼ばれます。モーター以外にもLED やブザーなど音や光などを出すものは「インジケータ」などと呼ばれます。

授業で使う部品にあてはめると…



●実際にキットからそれぞれの部品を取り出して確認させましょう。

●ロボットはどうやって動いているんだろう？

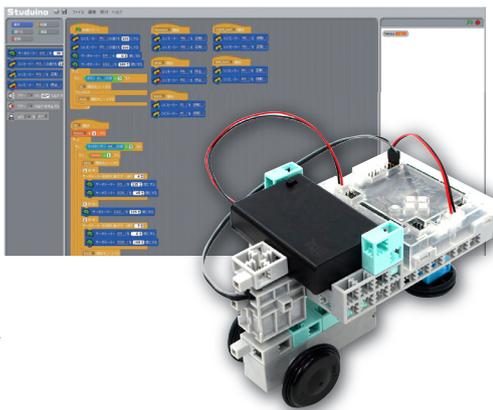
ロボットや電気製品の多くはコンピュータがさまざまな動きの命令を出すことによって動いています。しかし、コンピュータが「どんなときに、どういった命令を出すか」といった手順は、あらかじめ人間によって覚えさせる必要があります。この手順を示したものをプログラムといい、そのために使われる言語をプログラム言語、プログラムを作ることをプログラミングといいます。

```
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.

(0tmp = 'ja') (0tmp = '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。

累計時間
5分

目安時間
3分

+ 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。

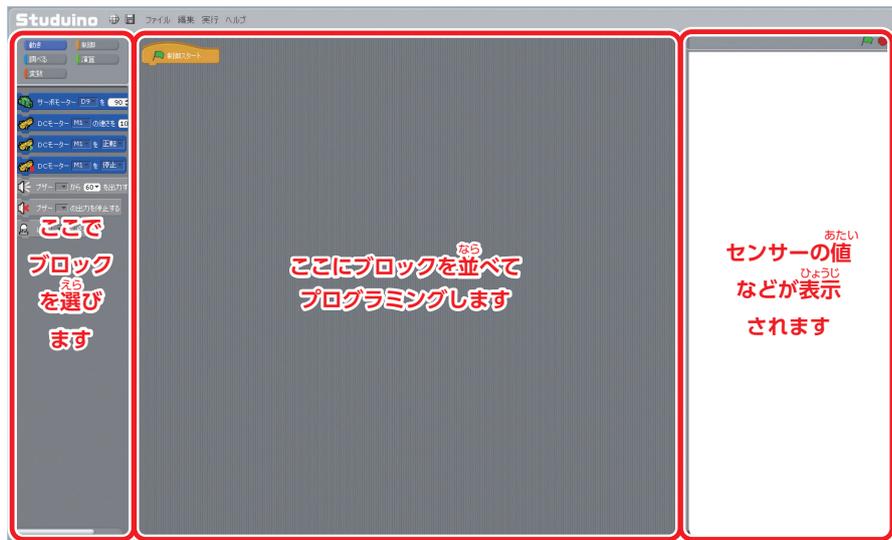


②ソフトウェアを起動すると、右の画面が表示されます。授業では、ブロックプログラミング環境を使います。選択して起動してください。



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



※ここからは、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。

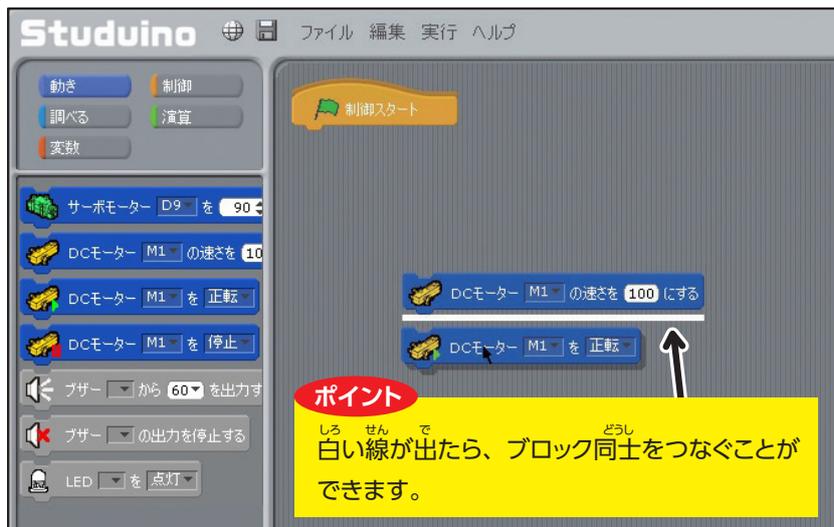
4 使用するブロックの選び方

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。ここではたくさんの種類があることを確認しましょう。



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。

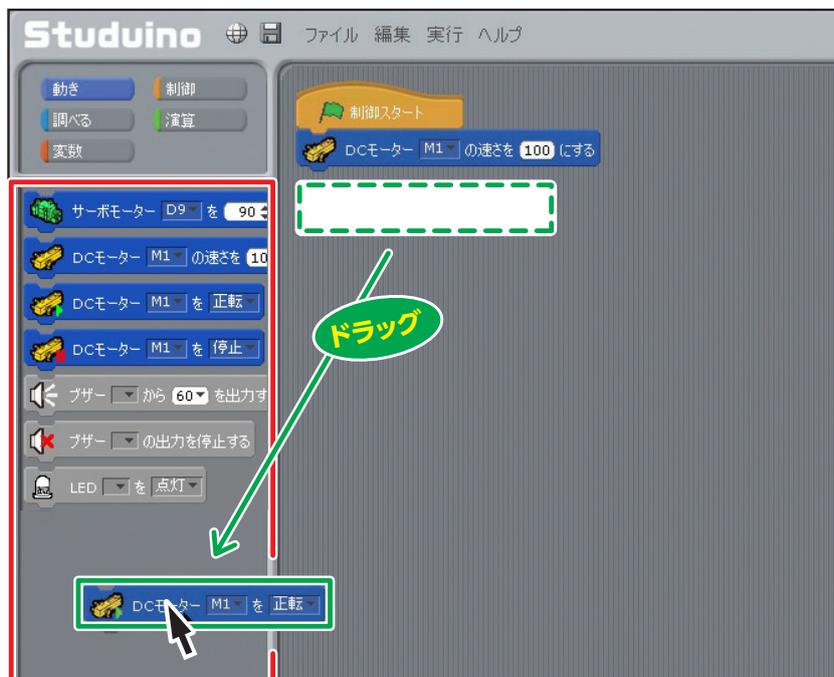


並べられたブロックをドラッグすることで、移動させることができます。その時、下につながっているブロックもいっしょに動きます。



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。

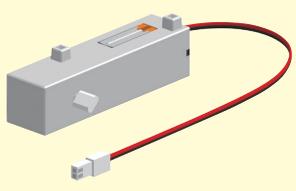


- ブロックプログラミング環境の操作に慣れてもらうために、児童に自由に触ってもらう時間を確保しましょう。また、次の課題にうつる前には、メニューバーの「ファイル」から「新規」を選び、プログラムを白紙に戻します。

●DCモーターを動かそう

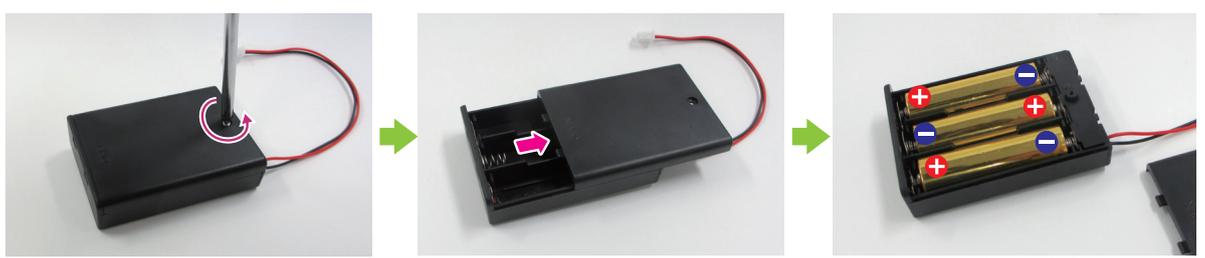
DCモーターってなに？

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モーターは、手で無理矢理回さないように注意しましょう。

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。

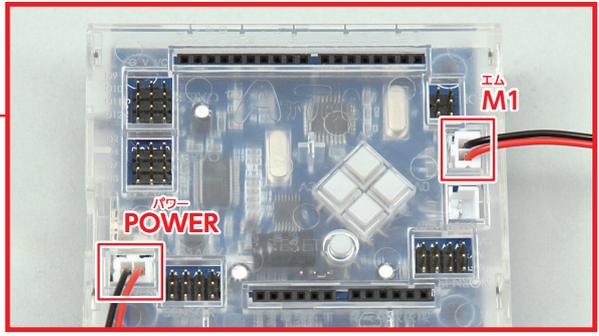
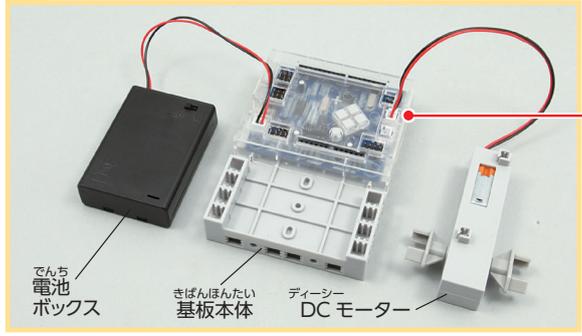


⚠️ + プラス - マイナスの向きに注意しましょう

●授業の前にあらかじめ電池を電池ボックスに入れておくと、スムーズに進行できます。

1 準備をしよう

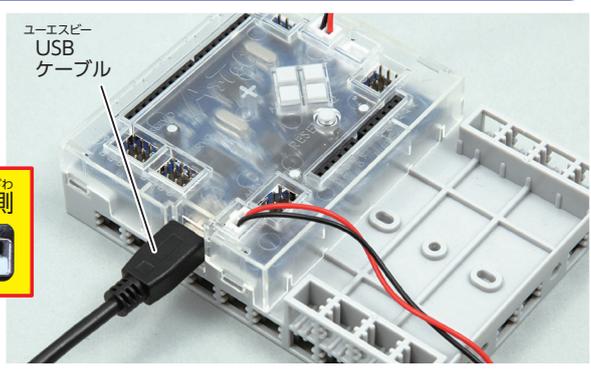
1 電池ボックスを基板本体の POWER に、DCモーターを M1 にとりつけます。



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。⚠️ コネクタの向きに注意

●電池ボックスとDCモーターを挿すポートを間違えないように注意しましょう。また、電池ボックスのコネクタの向きを逆向きに挿した場合、基板が破損する恐れがあるので特に注意します。

2 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

ユーエスピー やくわり
USBケーブルの役割

「ブロックプログラミング環境」で作ったプログラムは、
基板上的コンピュータが理解できる言葉にパソコン上
で変換されて、USBケーブルを通して転送されます。



```
// -----
void artecRobotSetup() {
  board.InitDCMotorPort(PORT_M1);
  board.InitServomotorPort(PORT_D9);
  board.InitSensorPort(PORT_A0, PIDLED);
  board.InitSensorPort(PORT_A1, PIDBUZZER);
  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 画面上的実行をクリックし、テストモード開始を選びましょう。



● USBが接続できていないエラーメッセージが表示された時は、1度USBケーブルを抜き挿しし、再度実行し直してください。

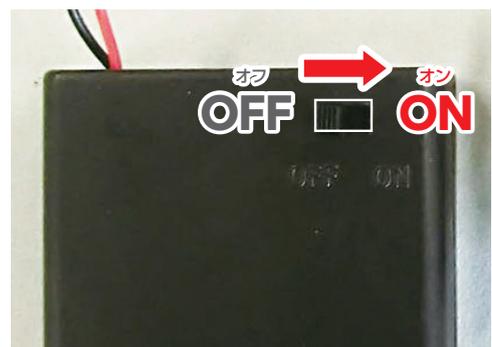
コンピュータに DC モーターを動かす命令を出すときは、次の 3 つのブロックを使います。

- DCモーター M1 の速さを 100 にする DC モーターの回る速さを決める。
- DCモーター M1 を 正転 回る向きを決めて DC モーターを回す。
- DCモーター M1 を 停止 DC モーターの回転を止める。

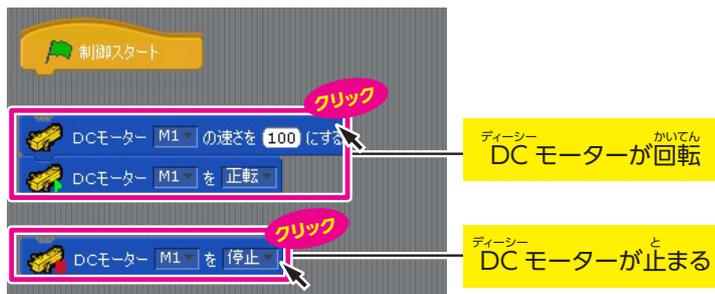
② 図のようにブロックをドラッグして並べましょう。



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。

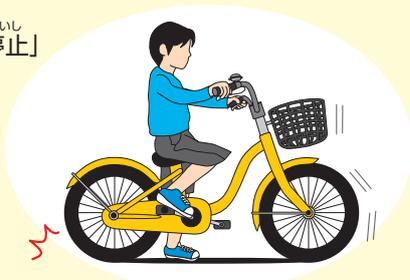


●DCモーターが動かない時は、テストモードが実行されているか、電池ボックスのスイッチがオンになっているかを確認しましょう。

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。

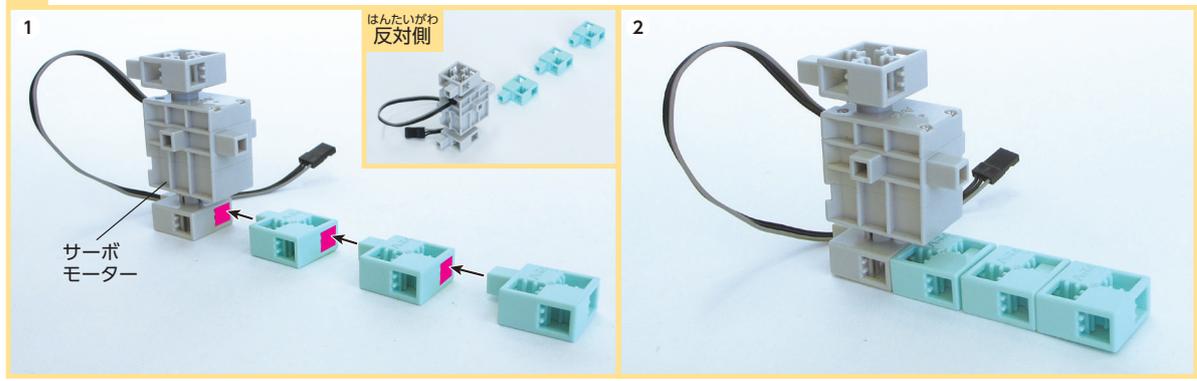


基板本体につないだDCモーターと電池ボックスを一度はずしておきましょう。

+ ロボットカーのプログラムをつくろう +

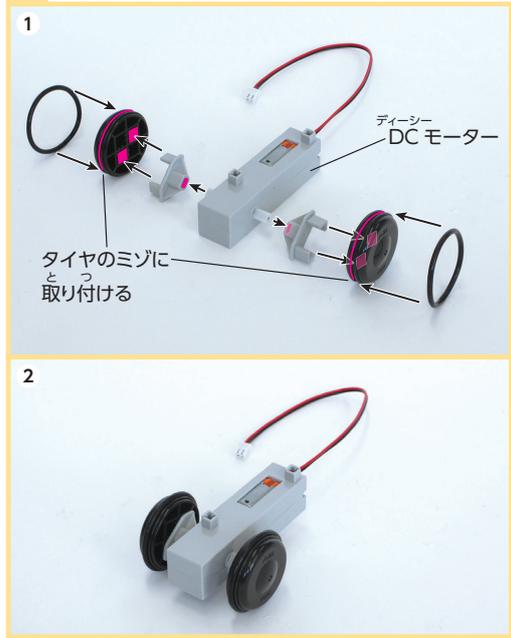
● ロボットカーを組み立てよう

1 サーボモーターにブロックを取り付けます。

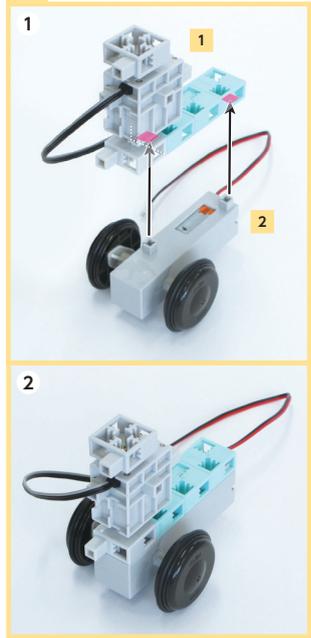


● サーボモーターにブロックを取り付ける時は、電気信号により回転する方(手で回した時に重たい方)に取り付けましょう。確認する場合は、手動で無理に回さないように注意しましょう。

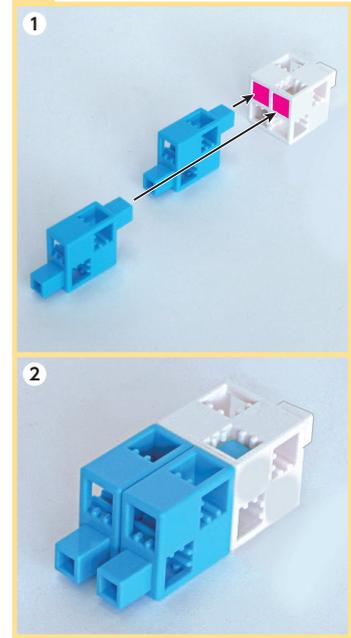
2 タイヤを組み立て、DCモーターに取り付けます。



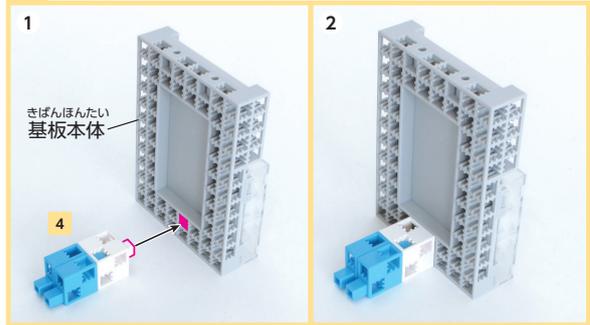
3 1と2を合体します。



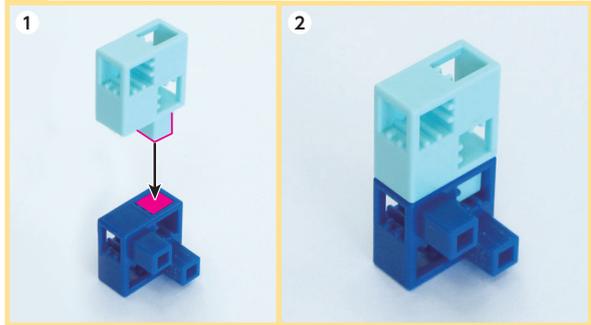
4 ブロックを組み立てます。



5 きばんほんたい基板本体に4を取り付けます。

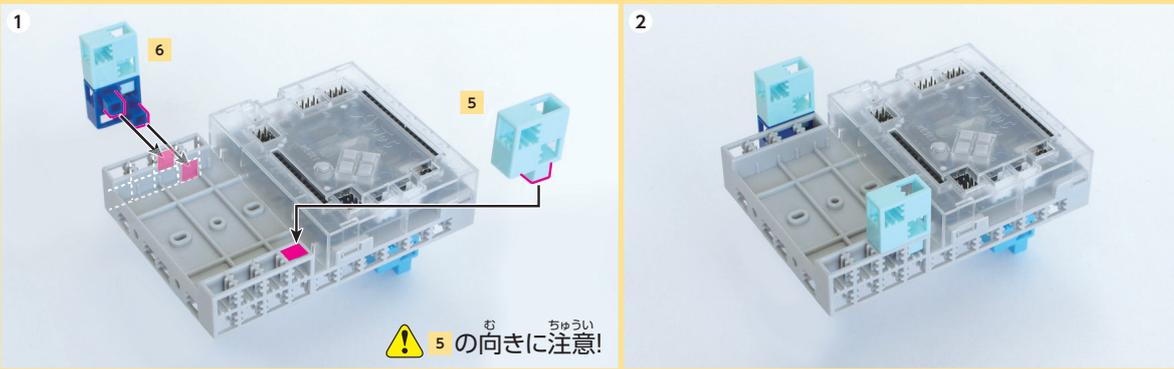


6 ブロックを組み立てます。

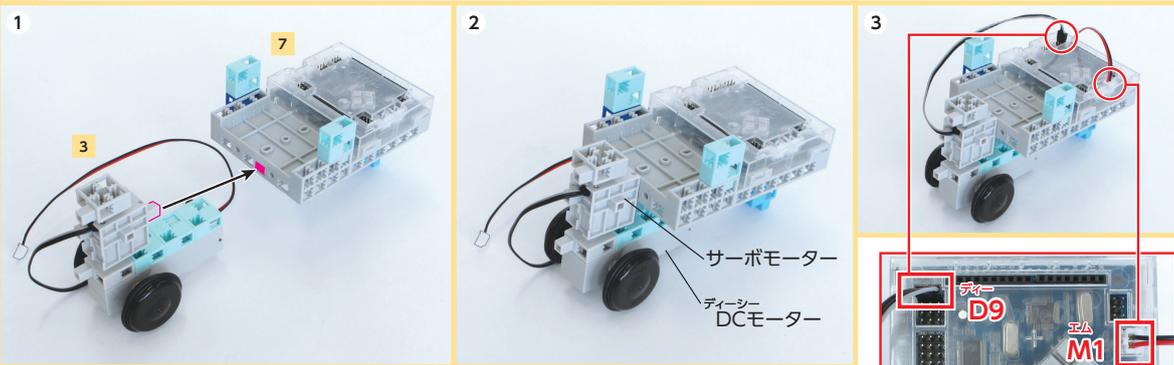


● 4 が台座の中央に取り付けられているかを確認しましょう。

7 5と6を合体し、さらにブロックを取り付けます。

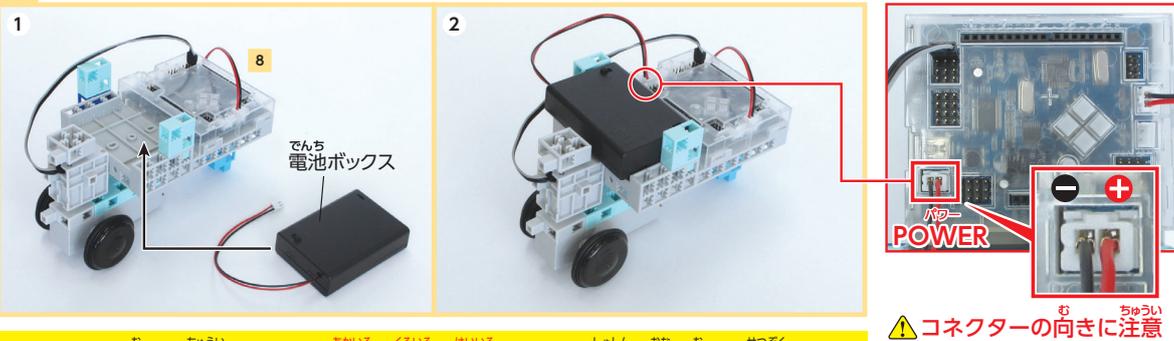


8 3と7を合体し、サーボモーターをD9に、DCモーターをM1に接続します。



●サーボモーターのコネクターがD9に挿せているか、またコードの向きは正しいかの確認を徹底してください。誤っている場合、ロボットカーは正しく動作しません。

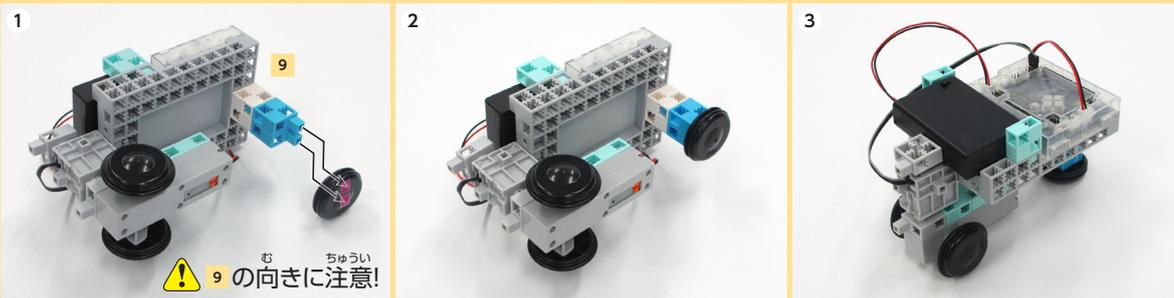
9 8に電池ボックスをセットし、POWERに接続します。



コネクター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。

●電池ボックスをセットする前に、電池ボックスのスイッチがオフになっていることを確認しましょう。

10 9にタイヤを取り付けて完成です。



累計時間
38分

目安時間
7分

●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。

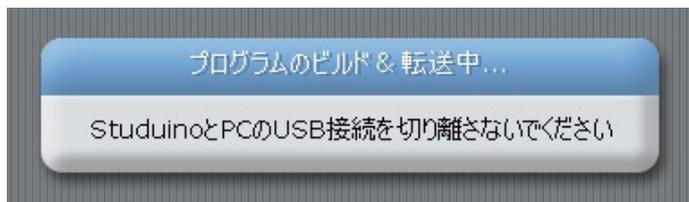


⚠ コネクタの向きに注意

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。



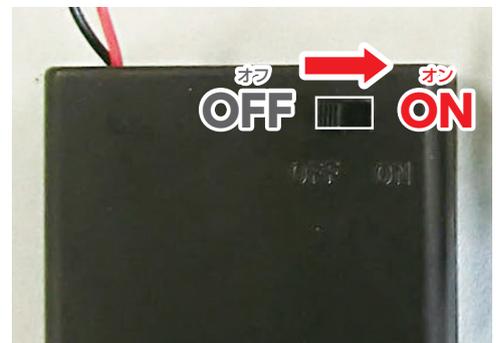
下の画面が表示されます。表示が消えるまで USB ケーブルを抜かずに待ちましょう。



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なくても、ロボットを動かせます。「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。



- 授業を開始する前に、コンピュータの起動やキットの配布を済ませておきましょう。
- プログラムしていない空のデータを基板本体に転送し、あらかじめデータを消しておきましょう。

2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



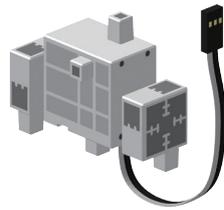
●サーボモーターの制御

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。

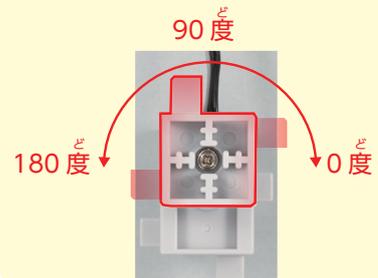
ロボットカーもサーボモーターを使用することで、自転車のようにタイヤ（DC モーター）の向きを変え、進行方向を変えることができます。

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。



●サーボモーターの電気信号により回転する方(手で回したときに重たい方)を無理やり回さないように注意してください。

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。

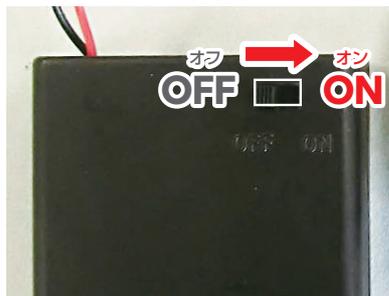
①USB ケーブルで基板本体とパソコンをつなぎます。



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。

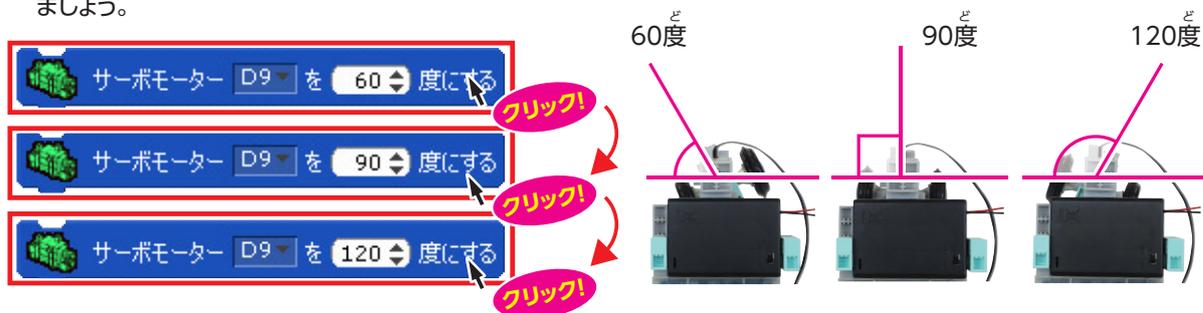
- ④ 動き から サervoモーター D9 を 90 度にする をドラッグし、3つのブロックどうしがくっつかないようにおきましょう。その後、それぞれのブロックの に60、90、120と角度の数字を入力します。

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。



●サーボモーターは個体差により正確に指定した角度にならないものもあります。極端に指定した角度と異なるときは、巻末資料のサーボモーターの校正方法を確認してください。

記入しよう

サーボモーターの角度とタイヤの向き関係

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 サーボ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。



累計時間
10分

目安時間
10分

●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。

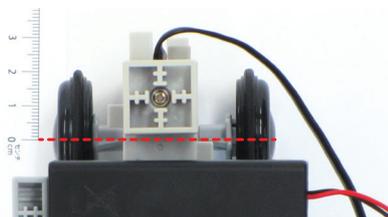


②2つめの「1秒待つ」の数値を変えて走行時間を調整します。

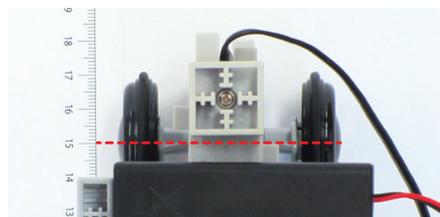


③プログラムを転送し、USB ケーブルを外します。

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。



⑥②～⑤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

びょう
秒

累計時間
20分

●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。

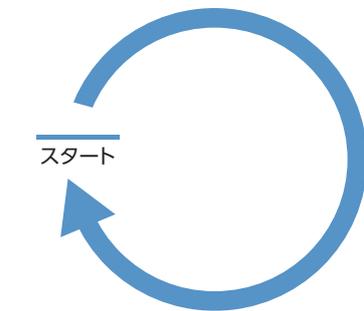


3 秒待つを入れることで、サーボモーターの角度が変わった後でロボットカーが走り出すようにしています。

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。

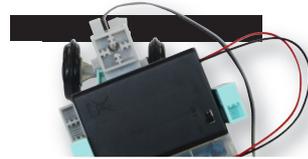


ロボットの動き



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。



- サーボモーターの角度が120度のとき、直径およそ50cmの円を描きます。
- 机で行う場合は落下に注意を促すか、床で行うように指示しましょう。

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

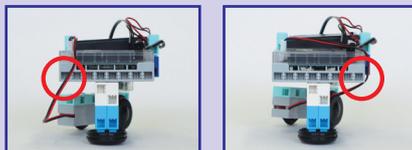
90度からの角度の差が大きいほど

回る円の大きさは(大きい ・ 小さい)

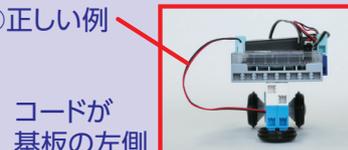
120度 - 90度 = 30度
150度 - 90度 = 60度
なので150度のほうが120度よりも
90度からの角度の差は大きい

●角度を150度や60度にするとDCモーターのケーブルが基板本体に引っかかり、サーボモーターが回転しないことがあります。DCモーターのコードが基板の左側を通るように注意しましょう。

×引っかかる例



○正しい例



コードが
基板の左側

目安時間
10分

●ロボットカーを左に90度曲がらせよう

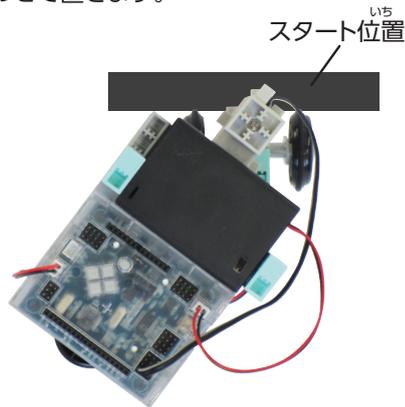
ロボットカーが左に90度曲がるようにプログラミングしてみましょう。

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。

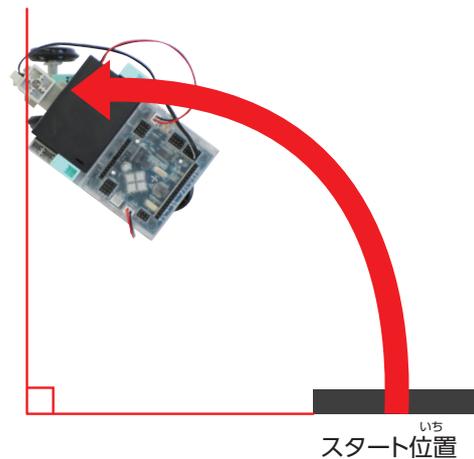


②プログラムを転送し、USBケーブルを外します。

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

累計時間
40分



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しましょう。

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。

累計時間
45分

- 授業を開始する前に、コンピュータの起動やキットの配布を済ませておきましょう。
- プログラムしていない空のデータを基板本体に転送し、あらかじめデータを消しておきましょう。

3時間目

使用するパーツ



1時間目で製作した車 ×1



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てみよう +

● まっすぐ走る、左右に曲がるプログラムを組み合わせてみよう

前回の授業までで、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。



はやむきまわ 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
びょうかんひだりま 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
びょうかんみぎま 3秒間右に曲がらせる
と 止める

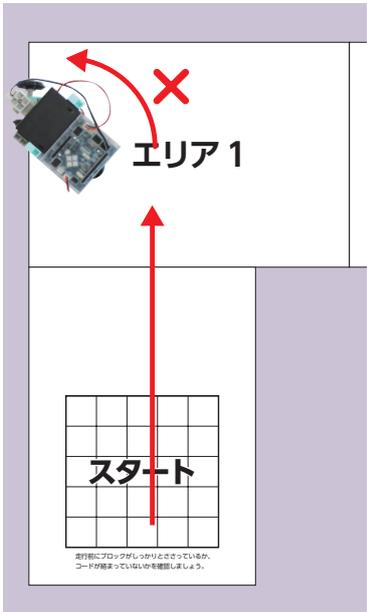
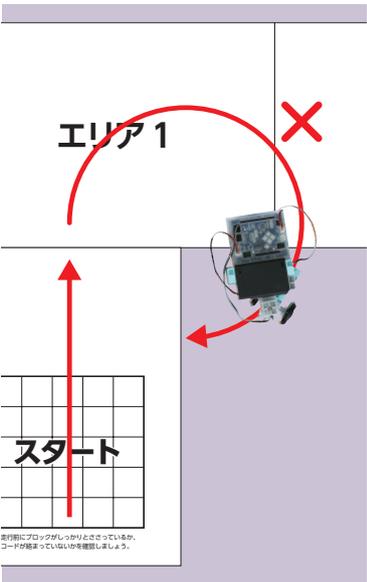
② 作成したプログラムを転送し、USB ケーブルを外したあと電池ボックスのスイッチを ON にしてロボットカーを走らせます。プログラム通りの動きをしているか確認しましょう。

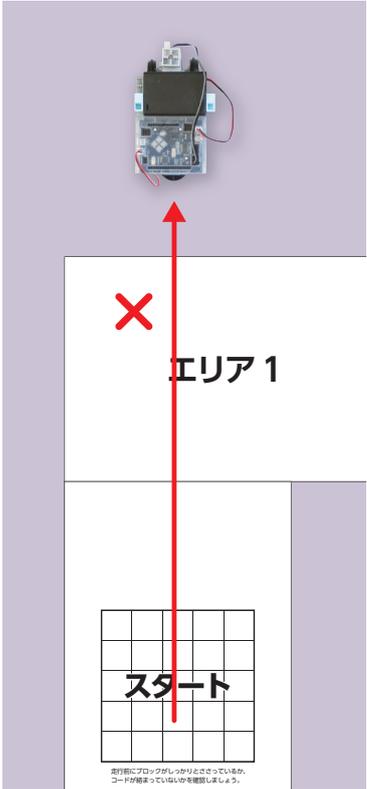
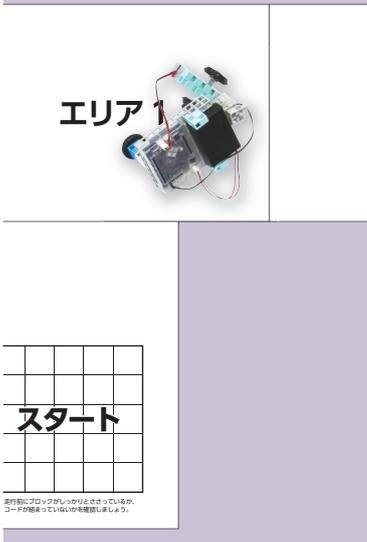


●課題コースのプログラムで児童がつまづいているときは、下記のことを確認してプログラムを改善するように伝えましょう。

●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>タイヤが反対の方向をむいている。</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>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>進行順にブロックがしっかりと並んでいるか、コードが揃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>曲がるときに倒れる。</p>  <p>進行順にブロックがしっかりと並んでいるか、コードが揃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>

サーボモーターの角度校正

サーボモーターの個体差により、微小な角度のズレが生じる場合があります。サーボモーターの角度校正をすることで、微小なズレを補正することができます。作例によっては、サーボモーターの角度を厳密に設定する必要があります。

注意！

角度校正を行った結果は、パソコン上のソフトウェア内に記憶されます。そのため、同じパソコンで同じサーボモーターを使用してプログラミングする場合は、再度の校正は必要ありません。ただし、使用するサーボモーターを変更した場合や、前回使用したパソコンとは異なるパソコンでプログラム作成を行う場合は、再度校正を行う必要があります。

- 基板本体をUSBケーブルでパソコンにつなぎます。
- 入出力設定にて指定したサーボモーターを基板本体に接続します。
- 画面上の **編集** から **モーター校正** を選択します。



- 下図のようなアラート表示がでて、しばらくするとサーボモーター校正のウィンドウが開きます。



- 基板本体に電池ボックスをつなぎ、電源を **ON** にすると、入出力設定されているサーボモーターがすべて90度の位置に固定されます。この時、90度の位置から少しずれている場合は、サーボモーター角度校正画面に数値を入力することで、サーボモーターの角度を補正することができます。90度の位置になるように数値を入力してください。

