ロボコンにチャレンジしよう

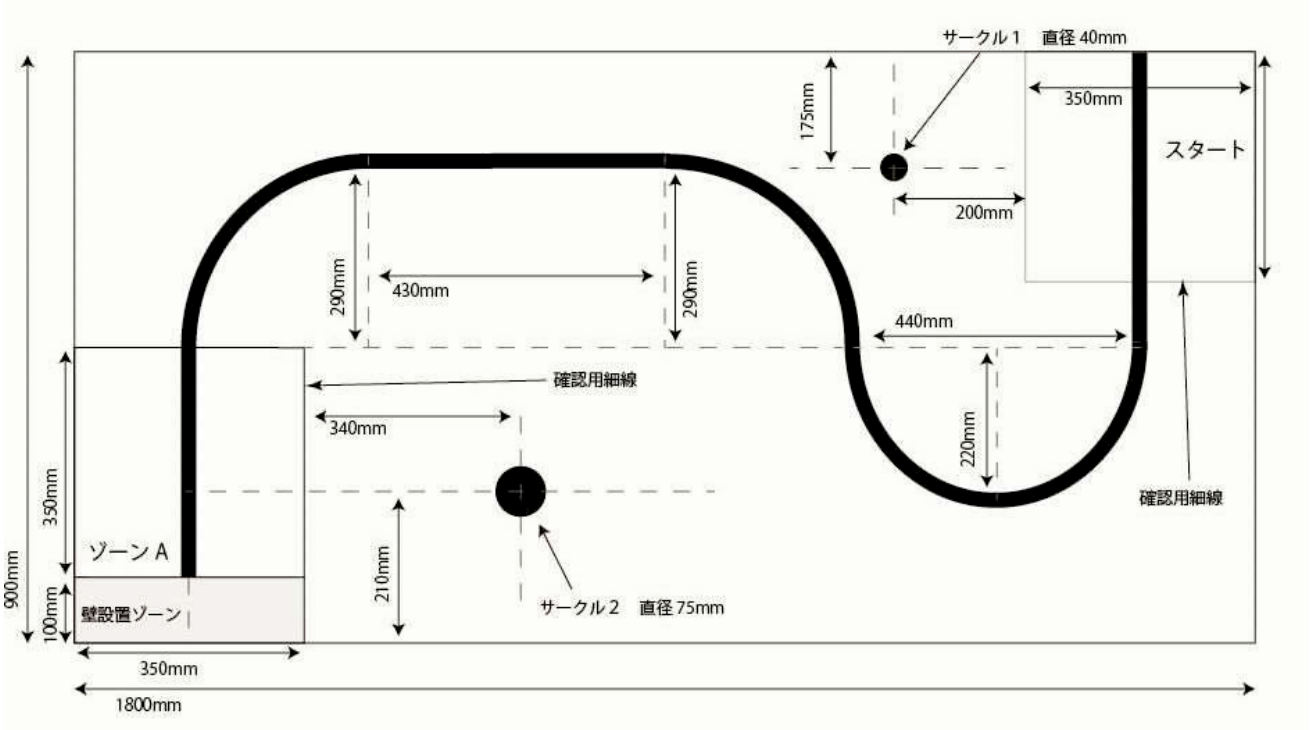
目次

第1章　競技ルール

１－１　競技ルールの説明

本テキストでは、下図のコースで競技を行います。

- 競技コース



１－２　競技内容

競技は、WROパイロット競技をベースにしています。
競技内容は以下の通りです。

1. 競技エリアのスタートゾーンからロボットをスタートさせる
2. ラインとレースをしてゾーンAに入る
3. サークル1に設置されたオブジェクトをつかむ
4. オブジェクトをサークル2の中に入れる
5. オブジェクト投入後、3秒間静止した時点で競技終了とする

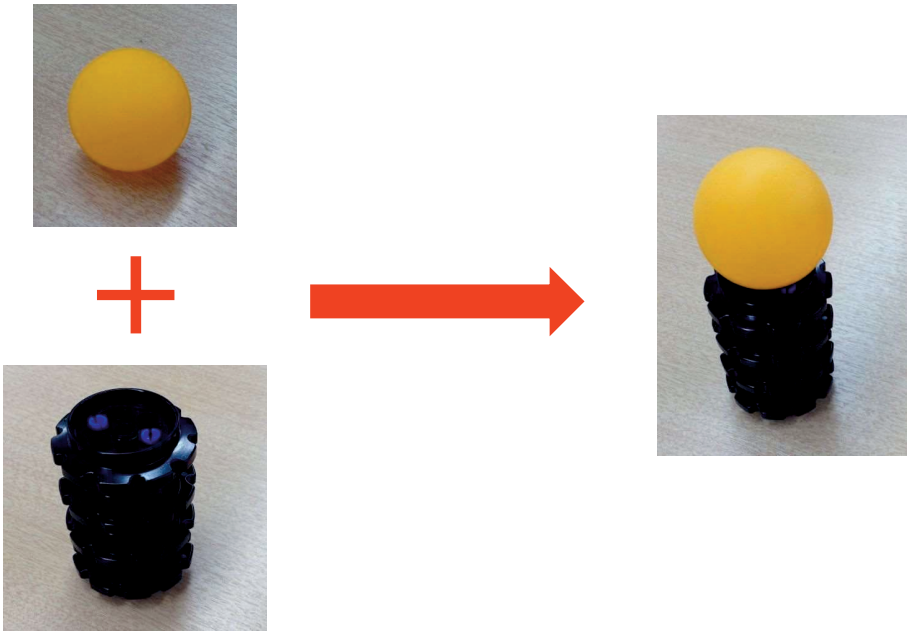
ポイント　ラインとレース	20点	
ゾーンAに入った	20点	
オブジェクトの移動	20点	
オブジェクト投入後3秒間停止	20点	合計80点

ポイントが同点の場合は、競技タイム（スタートからゴールまでの経過時間）を元に順位を決める。
※ゾーンAには、ロボット全体が完全に入ること

１－３　競技に使用するオブジェクト

サークル1には、40径のピンポン玉を、レゴブロックで組み立てた台に設置します。

- 台は付属資料の組み立てガイドにそって組み立ててください。

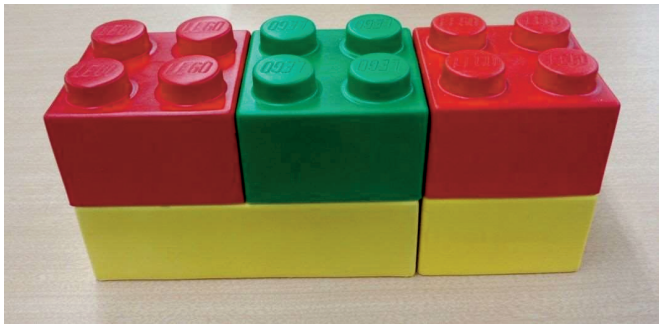


サークル2には、直径75mm、高さ50mmの筒を設置します。

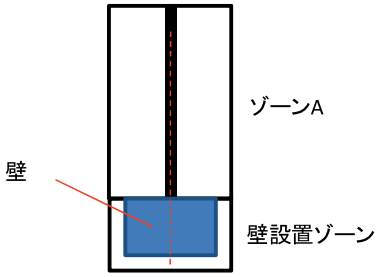
- テープの芯を使用、もしくは段ボール等で作成します。
  例：テープの芯

ゾーンAには、縦50～100、横250～350、高さ100～200（mm）の壁を設置します。

- ソフトブロックや、ティッシュ箱、段ボール等を使用します。
  例：ソフトブロック



壁の設置位置
・ゾーンAに面するように設置
・壁の中心とラインの中心を合わせる



１－４　制約条件

競技の制約条件を確認します。利用可能なハードウェアとソフトウェアは以下の通りです。

- ハードウェア
  - LEGO MINDSTROMS EV3基本セット（1個）
    - インテリジェントブロックEV3　×1
    - カラーセンサー　×1
    - 超音波センサー　×1
    - タッチセンサー　×2
    - Lモーター　×2
    - Mモーター　×1
- ソフトウェア
  - 教育版EV3ソフトウェア

１－５　競技要素の分解

競技を要素別に分解します。要素に分けることで、解決すべき問題を明らかにします。

- ラインレースをしてゾーンAに入る
  - ラインレース
    - 色や、明るさによって行動を変える
- サークル1に設置されたオブジェクトをつかむ
  - 目的地まで正確に動く
  - オブジェクトをつかむ
- オブジェクトをサークル2の中に入れる
  - 目的地まで正確に動く
  - オブジェクトをはなす

以上より、競技をクリアするのに必要な要素は以下のとおりです。

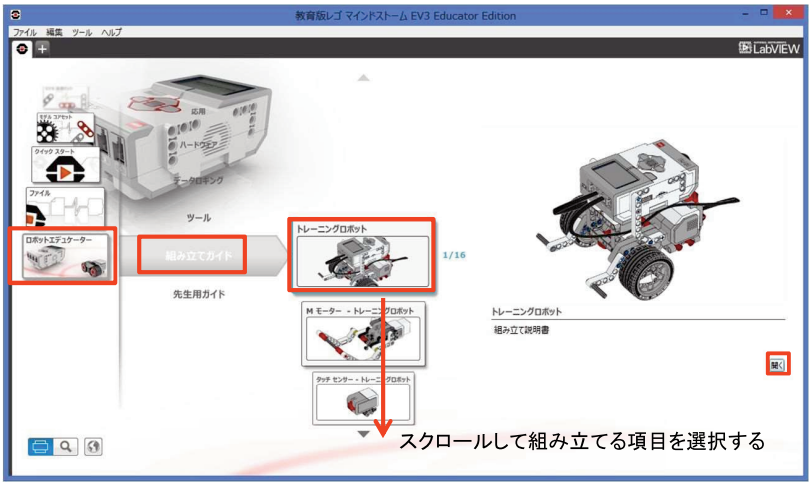
- ラインレース
- 目的地まで正確に動く
- オブジェクトをつかむ、はなす

第2章　ロボットの組み立て

２－１　トレーニングロボットの作成

学習にはトレーニングロボットを使用します。トレーニングロボットの組み立て図は教育版EV3ソフトウェアの中に入っています。
以下の手順で組み立て図を開き、組み立てを行ってください。

1. 教育版EV3ソフトウェアを起動します。
2. 左から「ロボットエデュケーター」「組み立てガイド」「トレーニングロボット」の順にメニューを選択し、「開く」をクリックし、トレーニングロボットを組み立てます。
3. 次に、「タッチセンサー」「カラーセンサー下向き」「超音波センサー」の順で組み立てを進めます。

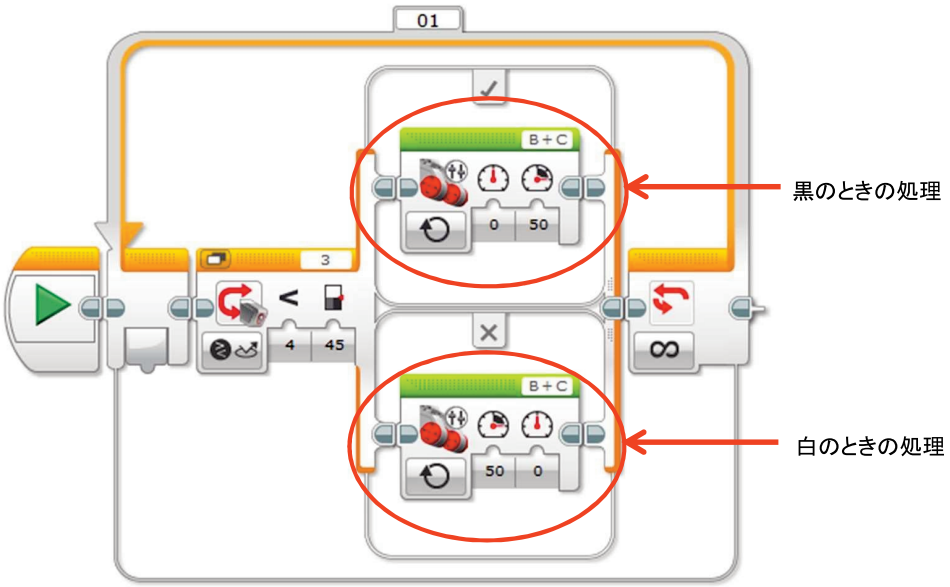


第3章　ライントレース

３－１　ライントレースをするには

ライントレースをするには色や、明るさによって行動を変える必要があります。
黒なら左へと動く、白なら右へ、基本的なライントレースを行ってみましょう。

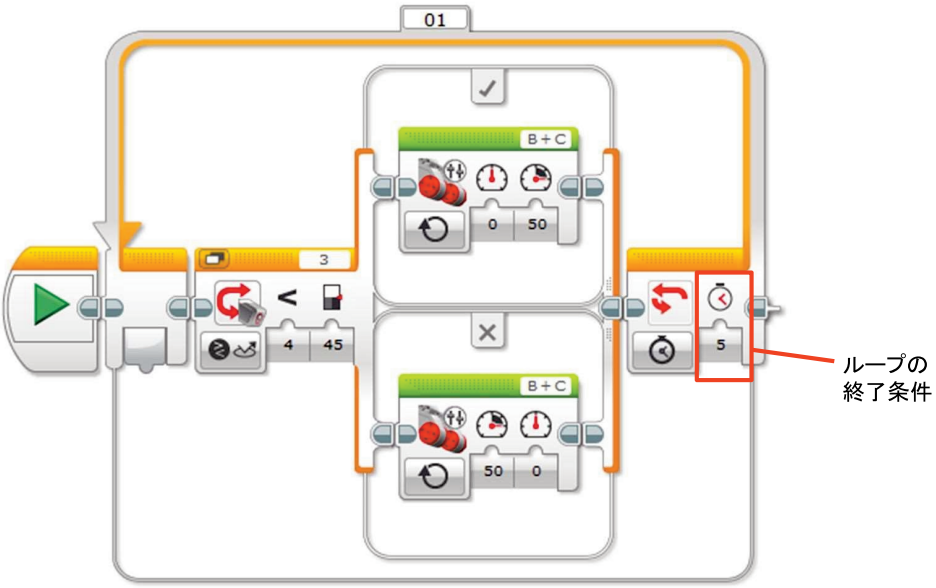
- 黒なら左へ、白なら右へと動くライントレース



３－２　ループを終了させるには（１）

ライントレースをやめるには、ループ中断する必要があります。
ループを中断する方法はいくつかあります。ここでは、秒数でループを中断してみましょう。

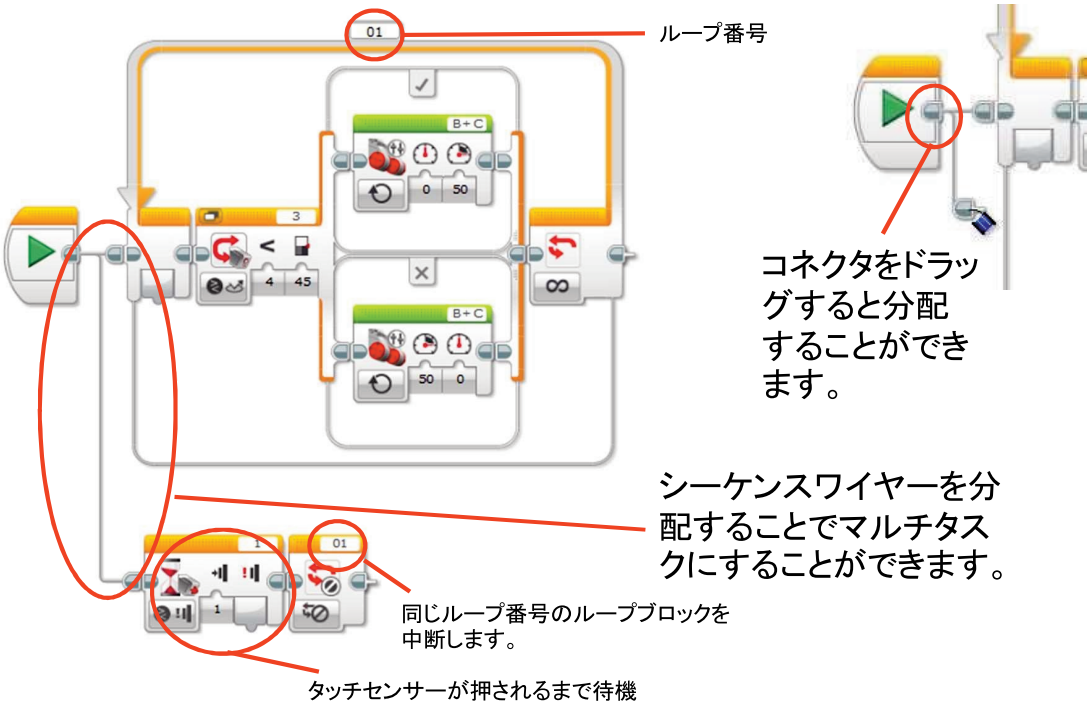
- ５秒間ライントレースするプログラム



３－３　ループを終了させるには（２）

次は、マルチタスク機能を使ってループを中断してみましょう。
マルチタスクとは、複数の処理を並行して行うことです。

- マルチタスクでループを中断するプログラム



３－４　ロボットのスタート位置

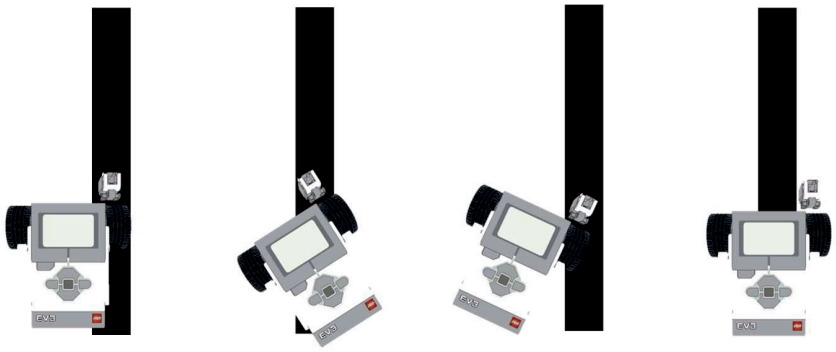
ライントレースを行う際、ロボットのスタート位置は毎回同じでしたか？

ロボットのスタート位置は非常に重要です。スタート位置が変わると、動作も変わります。

何が原因で失敗、または成功したのか、それを正確に把握するために、スタート位置は毎回同じ場所にする必要があります。

- スタート位置による動きの違い

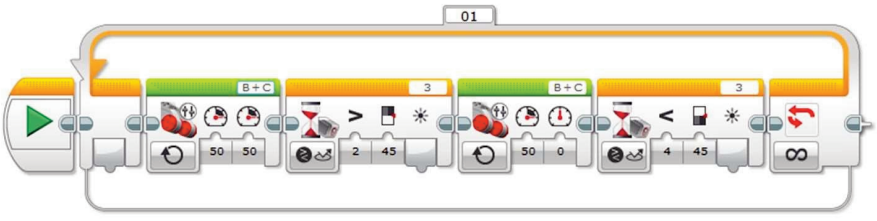
  それぞれのスタート位置で動きがどう変わるのか見てみましょう。



３－５　プログラムの記述方法

今は、「黒なら左へ、白なら右へ」という方法で線を追いました。しかし、プログラムの記述方法は一つではありません。ライントレースにも様々な方法があります。いろんなライントレースの方法を考えてみましょう。

直進して線から外れたら戦に戻るまで右に曲がるライントレース



- その他のライントレースのヒント

  スピードが速くなると，制御が難しくなり，スピードが遅いと制御しやすくなります。

  もっと速くライントレースするには？

  複雑なコースをライントレースするには？

  など、目的に合った方法を考える必要があります。

  汎用性を持たせなくてよければ、そのコースのみで速くライントレースできる方法を考えても良いでしょう。

第４章　正確な動き

４－１　正確な距離を進む方法（１）

競技では、目的の位置まで正確な距離移動するべき場面があります。
正確な距離を移動するためにはモーターの回転角度と距離の関係を知る必要があります。

- モーターの回転角度と距離

  左右のタイヤが等しく動くとき、ロボットが進む距離はタイヤの回転した距離と等しくなります。モーターが１回転したときのタイヤの回転数は、ロボットの機構によって変わります。

  トレーニングロボットはモーターが１回転すると、タイヤも１回転します。つまり、モーターとタイヤの回転比は１：１となります。よってモーターが１回転したときの進む距離は、タイヤの円周と等しくなります。進む距離の計算式は次のようになります。

  ロボットが進む距離＝１×タイヤの円周＝１×タイヤの直径×円周率

  このロボットのタイヤの直径は56mmなので、モーター１回転で進む距離は次の通りです。

  モーター１回転で進む距離＝１×５６×３．１４・・・≒１７５.８４（mm）

  １回転＝３６０度なので、求めた距離を３６０で割ると、モーターの回転角度１度あたりの距離を求めることができます。

  モーターの回転角度１度あたりに進む距離＝１７５．８４÷３６０≒０．４９（mm）

  これでモーターの回転角度と距離の関係を知ることができました。

４－２　正確な距離を進む方法（2）

モーターの回転角度と距離の関係を用いて、正確な距離を進んでみましょう。

- 正確に10cm前進するプログラム

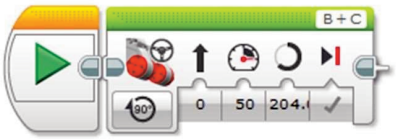
10cm前進するために必要な、モーターの回転角度を求めます。
計算式は次のようになります。

モーターの回転角度＝進みたい距離÷回転角度1度あたりに進む距離

よって、100÷0．49＝204．08（度）になります。

※単位を統一しなければいけないことに注意しましょう。

50のパワーで10cm進むプログラムです。
正確に10cm進むか確かめてみましょう。



また、以下のことを試してみましょう。
・20cm後進するプログラムを作成してみましょう。
・1mm進むために必要な回転角度を求めてみましょう。

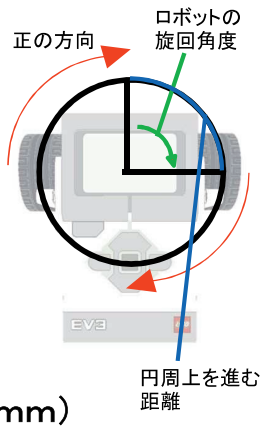
４－３　正確な角度旋回する方法（1）

目的の位置が直進方向にない場合には、どうすればよいでしょうか？
解決方法のひとつとして、旋回する方法があります。

- ロボットの旋回角度と距離

ロボットが旋回した場合、その軌跡はロボットのタイヤ間の距離を直径とした円となります。
ロボットの旋回角度と円周上進む距離の関係式は次の通りです。

円周上を進む距離＝タイヤ間の距離×円周率
×（ロボットの旋回角度÷360）



ロボットが1度旋回するための距離を求めてみましょう。
トレーニングロボットの場合、タイヤ間の距離は約12cmなので計算は次のようになります。

ロボットが1度旋回するために進む距離
＝120×3．14×（1÷360）＝1．05（mm）

これでロボットの旋回角度と距離の関係を知ることができました。

４－４　正確な角度旋回する方法（2）

ロボットの旋回角度と距離の関係を用いて、正確な角度旋回してみましょう。

- ロボットが正確に90度旋回するためのプログラム

※時計回りの回転方向を正とします。

ロボットが90度旋回するために必要な、ロボットの進む距離を求めます。

計算式は次のようになります。

ロボットの進む距離＝ロボットの旋回角度

÷ロボットが1度旋回するために進む距離

距離のままでは、扱うことができないので、４－１で学習した「モーターの回転角度1度あたりに進む距離」を用いて、距離からモーターの回転角度を求めます。

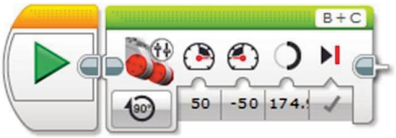
モーターの回転角度＝ロボットの進む距離

÷モーターの回転角度1度あたりに進む距離

よって、90÷1．05÷4．09≒174．93（度）になります。

50のパワーで90度旋回するプログラムです。

正確に90度旋回するか確かめてみましょう。



また、以下のことを試してみましょう

・－60度旋回するプログラムを作成しましょう。

・180度旋回するプログラムを作成しましょう。

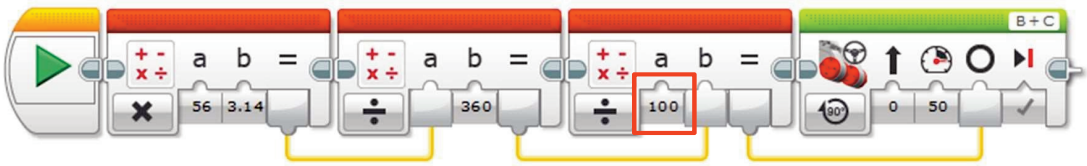
４－５　計算ブロックの使い方（1）

正確に移動するための計算を、毎回手計算していては大変です。計算ブロックを使って、プログラム上で計算するようにしてみましょう。

- 正確な距離を進む計算

計算ブロックを使うと様々な計算をプログラム上で行うことができます。

計算ブロックを使った正確に10cm前進するプログラム

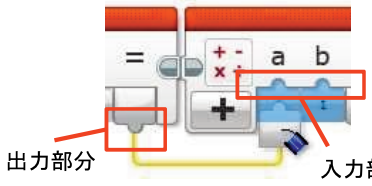


正確に10cm進むか確かめてみましょう。

計算ブロックを使うと、進む距離を簡単に変えることができます。赤枠内の数値を変えることで、進む距離を変えることができます。

また、赤枠内の数値にマイナスの値を入れたらどうなるか試してみましょう。

データの受け渡し



データワイヤーを使うとデータを渡すことができます。データワイヤーは、出力の部分と入力部分の形が同じであれば、繋ぐことができます。出力部分をドラッグして入力部分へ繋ぎます。

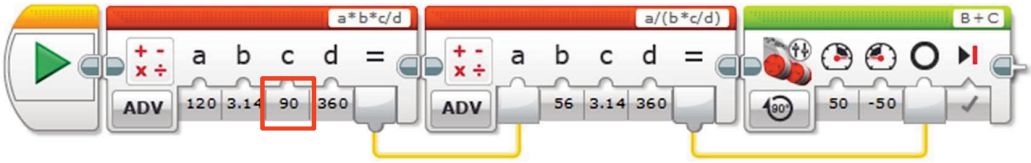
４－６　計算ブロックの使い方（２）

計算ブロックの拡張機能モードでは、まとめて計算したり、複雑な計算をしたりすることができます。

- 正確な角度旋回する計算

  拡張機能モードをつかってまとめて計算してみましょう。

  計算ブロックを使った正確に90度旋回するプログラム



  正確に90度旋回するか確かめてみましょう。

  拡張機能モードを使うと、ブロックが多くなり、複雑なプログラムになることを避けることができます。

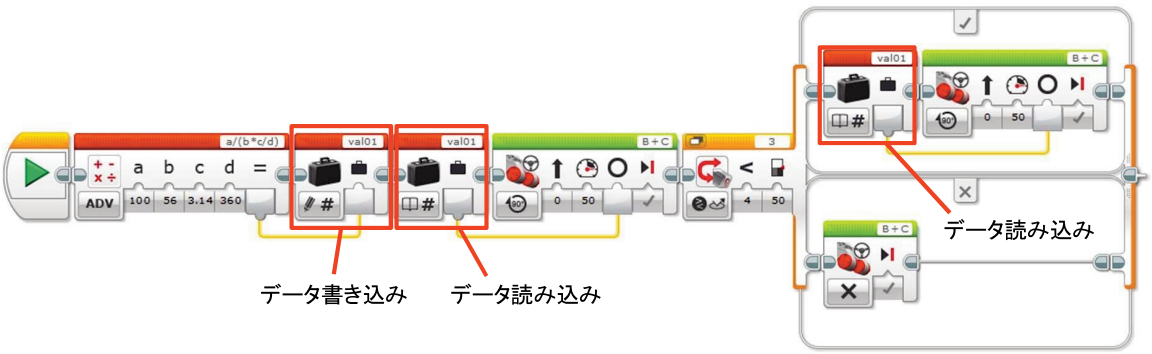
  赤枠内の数値にマイナスの値を入れたらどうなるか試してみましょう。

４－７　変数ブロックの使い方

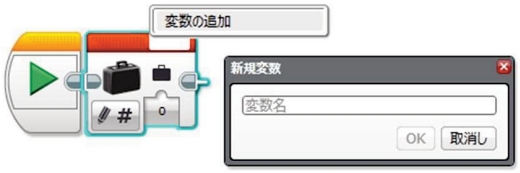
計算した値をもう一度使いたくなったときには、変数ブロックを使います。
変数ブロックを使うと、離れた場所や、スイッチ分の中でも値を呼び出すことができます。

- 計算した値を別の場所で使う方法

  10cm進んだ先が暗ければ10cm進み、明るければ止まるプログラム



変数の作成



変数を扱うには、変数を作成する必要があります。
変数名はわかりやすいものにしましょう。呼び出す際は、変数名を指定します。
変数は何度でも呼び出すことができます。

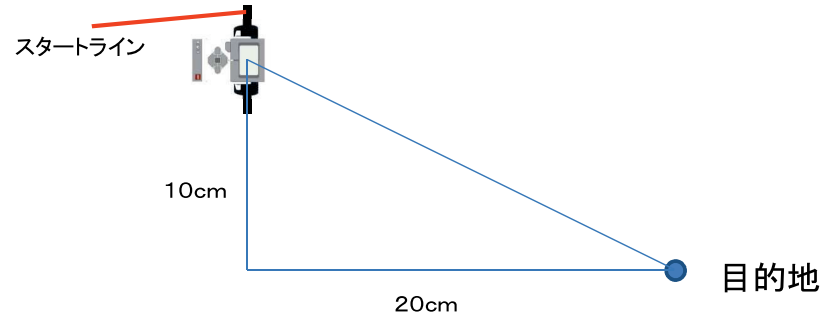
４－８　目的地まで移動する

４章で学んだことを活かして、斜め先にある目的地まで移動しましょう。
床の材質によって、計算を調整する必要があることに注意しましょう。

- 斜め先の目的地まで移動する
  - ルール
    - ・スタート位置はスタートライン上にタイヤを乗せること
    - ・目的地に到達時、目的地の印がタイヤの間にあること
    - ・目的地に到達したときの向きは問わない

目的地例

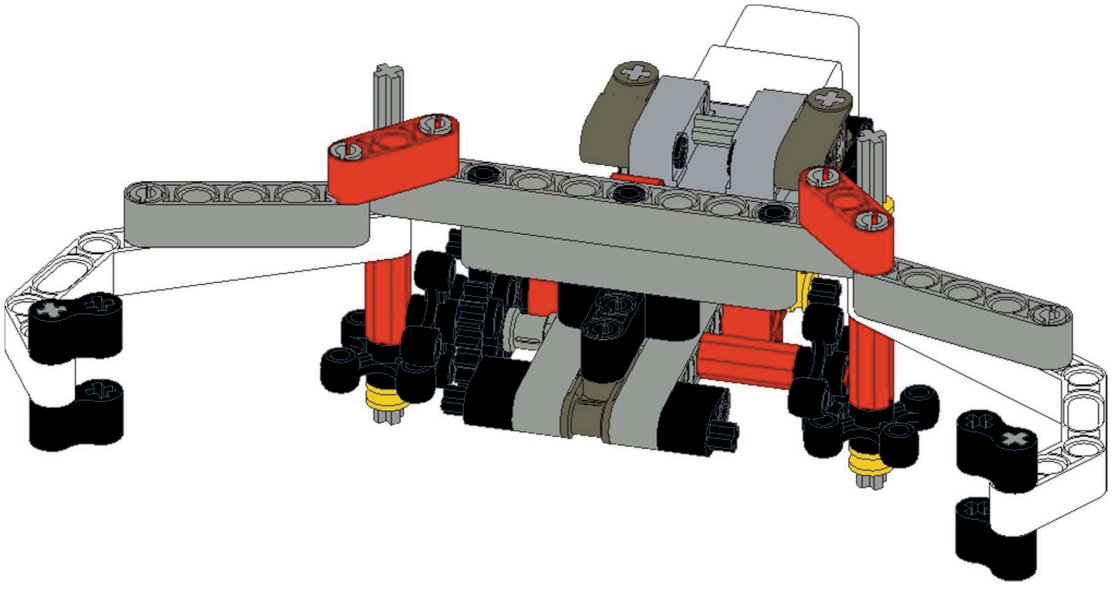
自由に目的地を設定して、そこまで到達してみましょう。



第5章　つかむ

５－１　ものをつかむ機構の作成

オブジェクトを運ぶためには、まずものをつかむ機構を作成する必要があります。
付属資料の組み立てガイドに沿って、組み立てを行ってください。



５－２　ものをつかむ方法

ものをつかむときには、未調整のモーターブロックを使います。未調整のモーターブロックを用いると、モーターに負荷が掛かったときに、モーターの回転を止めることができます。モーターに負荷が掛かり機構やモーターが壊れるのを、防ぐことができます。

- ものをつかんでから、離す

  物をつかんだとき、ものをつかんでないときの動きを見比べてみましょう。

  アームを閉じてから、開くプログラム

  未調整のモーターブロックを用いても、パワーが強すぎると機構に負荷が掛かり過ぎることがあります。機構によって、適切なパワーを設定するようにしましょう。

第６章　競技にチャレンジ

６－１　競技ルールの確認

1. 競技エリアのスタートゾーンからロボットをスタートさせる
2. ライントレースをしてゾーンAに入る
3. サークル1に設置されたオブジェクトをつかむ
4. オブジェクトをサークル2の中に入れる
5. オブジェクト投入後、3秒間静止した時点で競技終了とする

ポイント　ライントレース	20点	
ゾーンAに入った	20点	
オブジェクトの移動	20点	
オブジェクト投入後3秒間停止	20点	合計80点

※スタートゾーンの範囲からはみ出さないようにロボットを設置します。
※ロボットはゾーンAに全体が入りきる必要があります。

付録

A－１　配列ブロックの使い方

ここでは、配列ブロックの使い方を紹介します。配列ブロックを使って、複数の色を保存し、読み取った色によって行動を変えてみましょう。

- 配列とは

講座中では、値を保存するには変数を使っていました。変数は1つの値しか保存することができないので、複数の値を保存するとき、沢山の変数を用意することになり、大変です。そんなとき、配列を使うと便利です。配列はいくつもの変数の集まりなので1つの配列で、複数の値を保存することが出来ます。

配列を構成する変数には、番号が割り振られます。これをインデックスといいます。インデックスは配列の最初の変数を0として、順番に数字が割り振られます。
例えば、4つの変数で構成された配列「test」は下図のようになります。

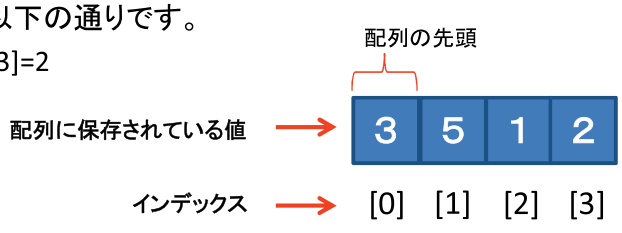
配列：test

この場合配列「test」の構成要素は以下の通りです。

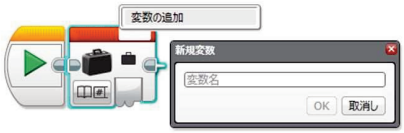
test[0]=3, test[1]=5 , test[2]=1 , test[3]=2

※配列名 [インデックス]=値

n個の要素を持つ配列の場合、
インデックスは[0]~[n-1]となります。



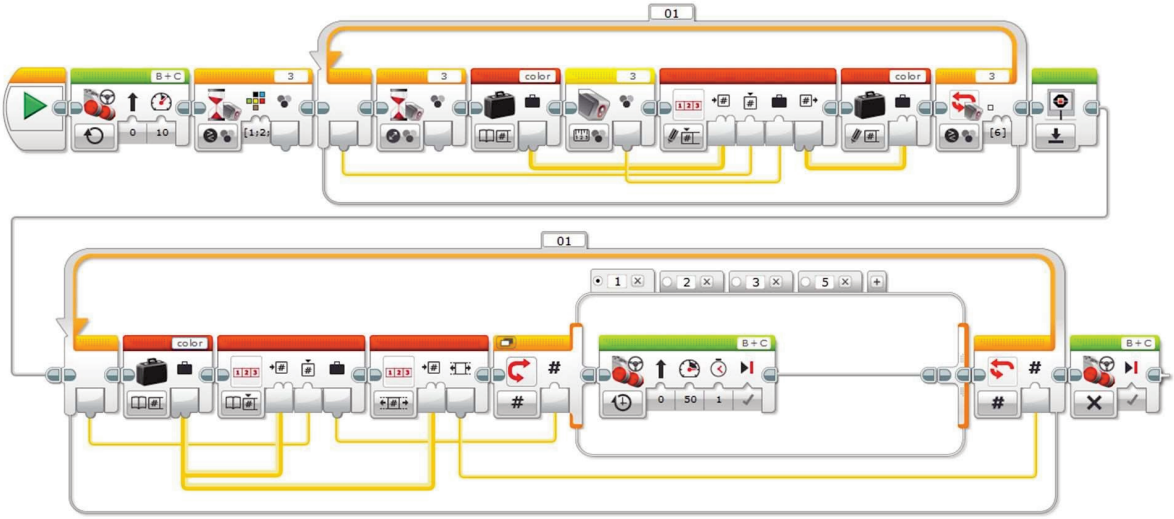
配列の作成



配列も、変数と同じように、値を入れるための箱を作る必要があります。
配列の名前とインデックスを指定することにより、配列の中身を操作します。

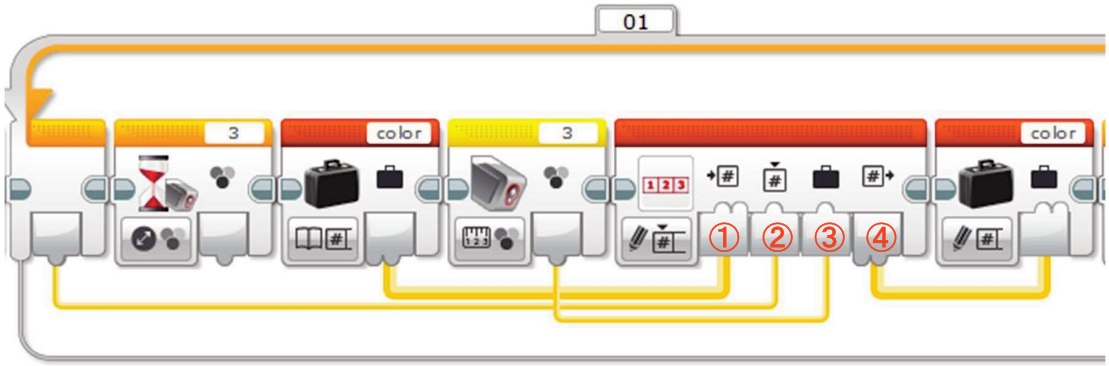
- 複数の色を保存する方法

  色を配列に保存し、色によって行動を変えるプログラム

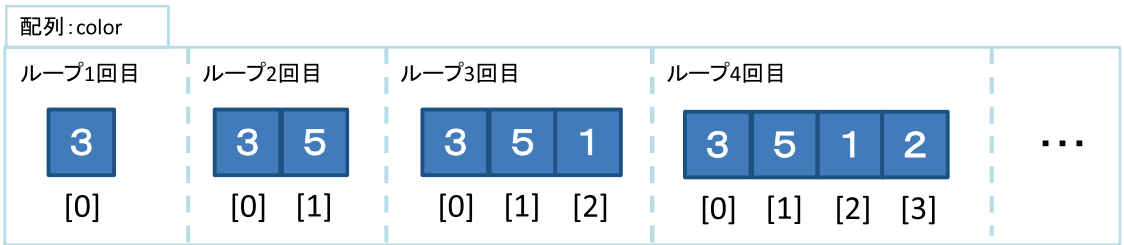


- 値を保存する方法

  配列の先頭から順番にカラーセンサーの値を保存しています。

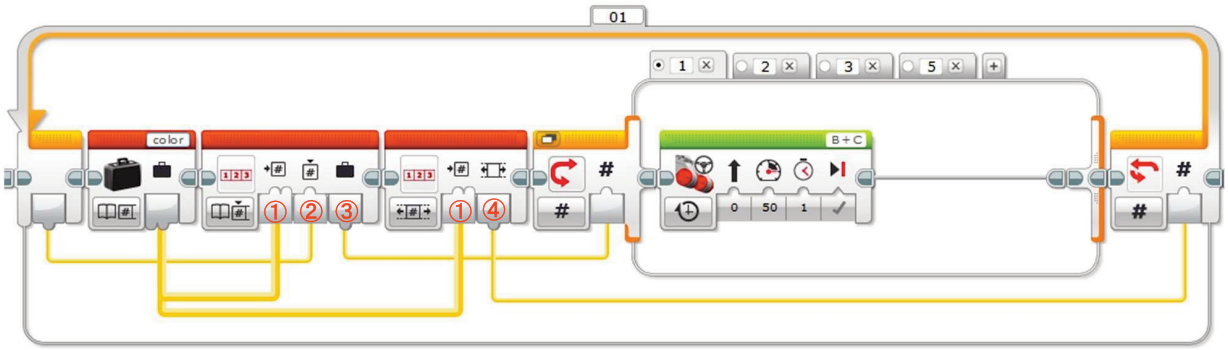


①配列入力・・・値を保存する配列を指定します
②インデックス・・・値を保存するインデックスを指定します
③値・・・配列に保存する値を指定します
④配列出力・・・値を保存した後の配列の値です



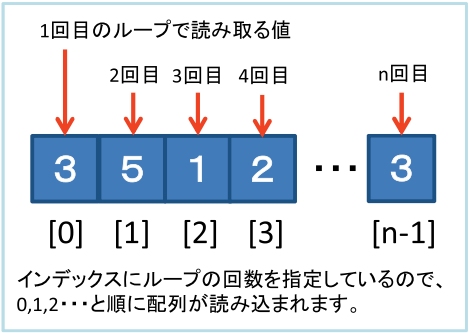
- 値を読み出す方法

  配列の先頭から順番に値を読み込み、その値によって行動を変えます。
  配列に保存されている値の数だけ、ループを繰り返します。



①配列入力・・・値を読み込む配列を指定します
②インデックス・・・値を読み込むインデックスを指定します
③値・・・配列に保存されている値を出力します
④長さ・・配列に保存されている値の個数

読み取った値による行動
1：黒・・・1秒直進
2：青・・・1秒後退
3：緑・・・1秒右に旋回
5：赤・・・1秒左に旋回

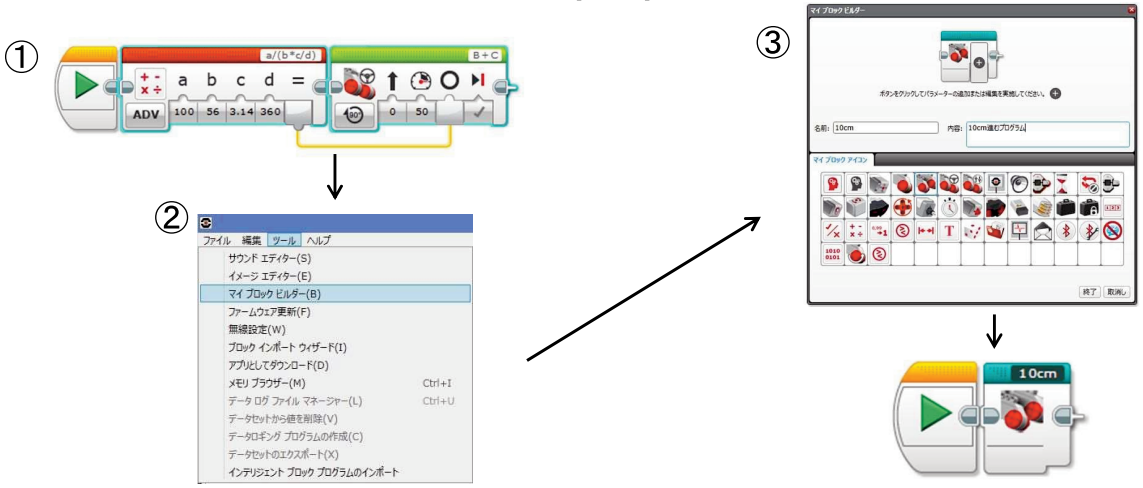


A－2　マイブロックの使い方

ここでは、マイブロックの使い方を紹介します。マイブロックは、処理を繰り返す際に便利です。また、処理をまとめるので、プログラムの複雑化を避けることができます。

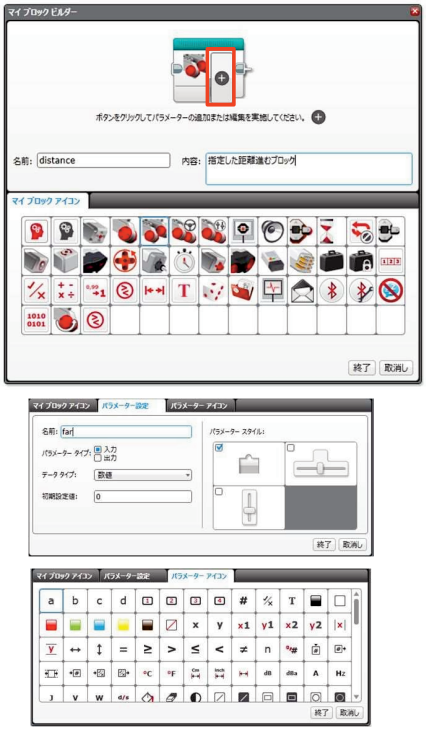
- 指定した距離を正確に進むマイブロック

  10cm正確に進むプログラムをマイブロック化してみましょう。
  マイブロックの作製手順は以下の通りです。
  ①ひとまとめにしたい処理のブロックを全て選択する。（[開始]ブロックは含まない）
  ②[ツール]から[マイブロックビルダー]を選択する。
  ③名前と内容を入力し，アイコンを選択後[終了]をクリックする。



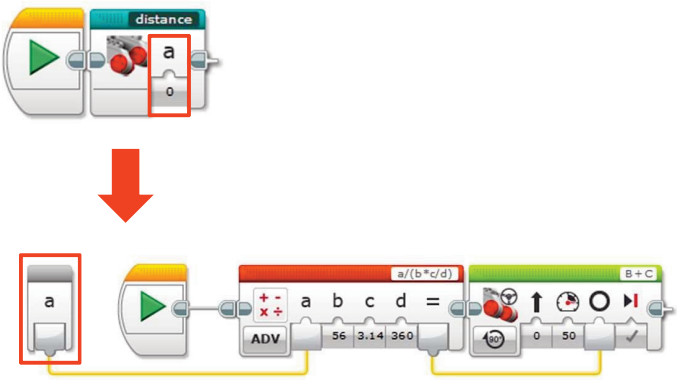
- パラメータ付きマイブロックの作成方法

  頻繁に値を変更する場合があるブロックの場合、ブロックにパラメータを追加すると便利です。
  指定した距離を正確に進むプログラムを作成してみましょう。
  もう一度先ほどのプログラムを作成し、マイブロックビルダーを開くところまで
  進みます。



赤枠の部分をクリックして、マイブロックにパラメータを追加します。
パラメータを追加すると、「マイブロックアイコン」の横に、「パラメーター設定」「パラメーターアイコン」のタブが追加されます。目的に応じたパラメーターを設定します。
全ての設定が終了したら、「終了」を押すと、マイブロックが作成されます。

マイブロックの中に、作成したパラメーターブロックがあるので、パラメーターを設定したい部分に繋ぎます。



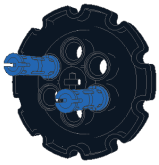
付属資料

オブジェクト台組み立てガイド

■完成図（オブジェクト台）

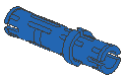
■組み立て図

1.

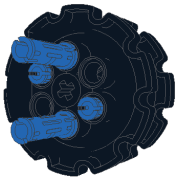


1x

2x

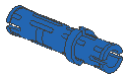


2.

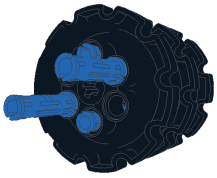


1x

2x

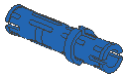


3.

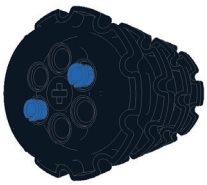


1x

2x

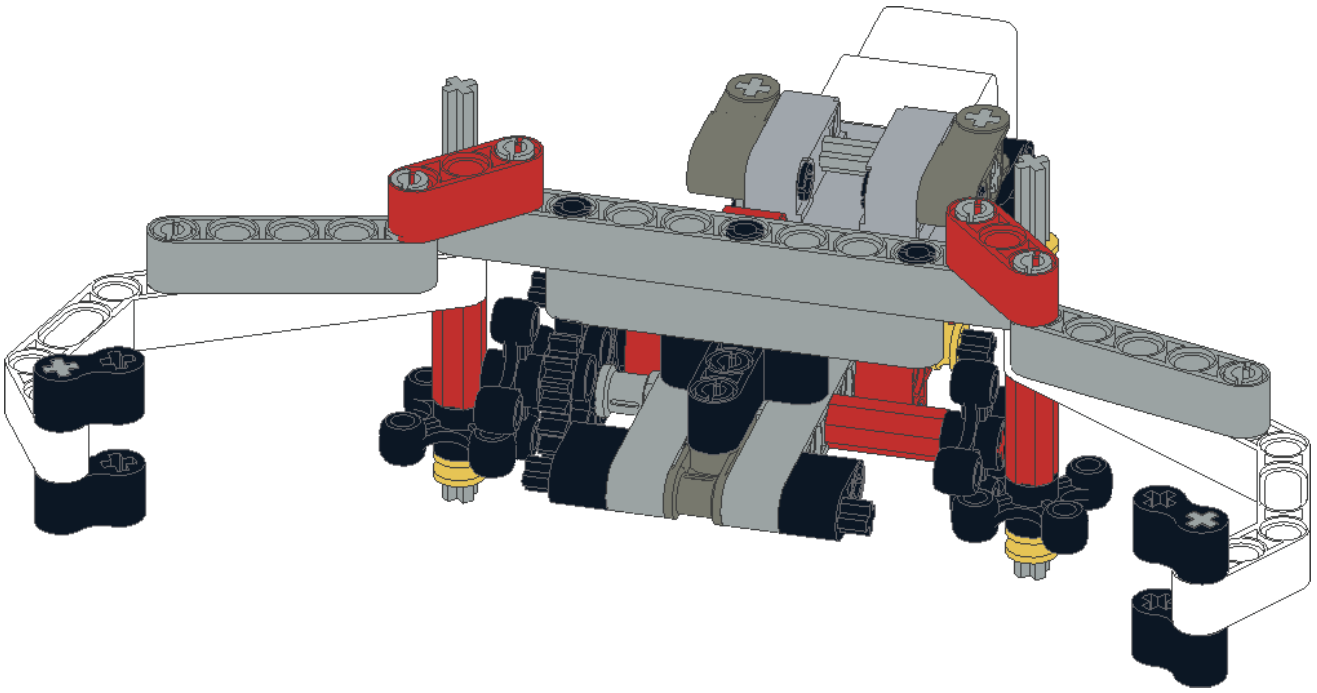


4 .

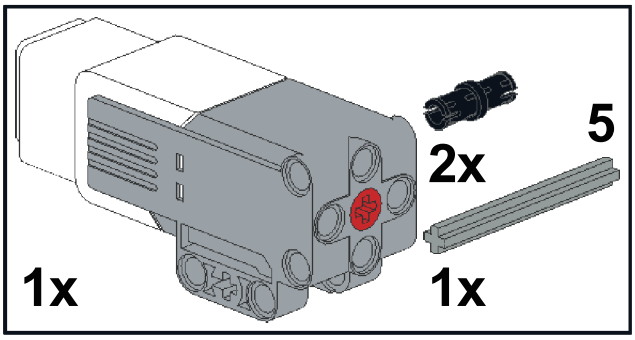


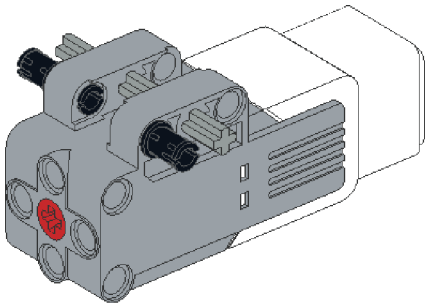
1x

つかむ機構組み立てガイド



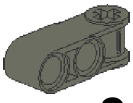
1



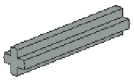


2

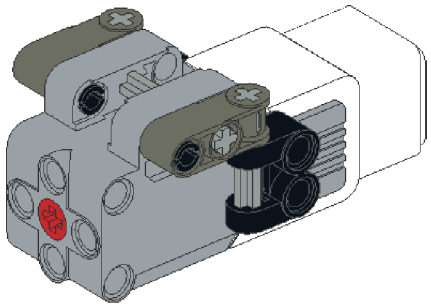
2x



2x 3

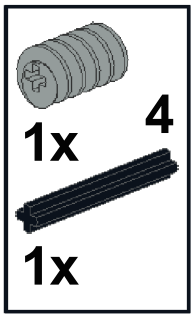


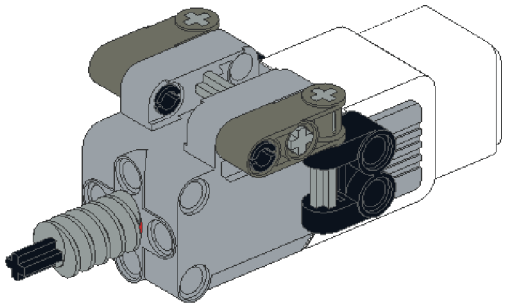
2x



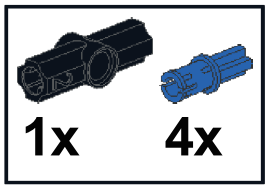
反対側も同様に作成する

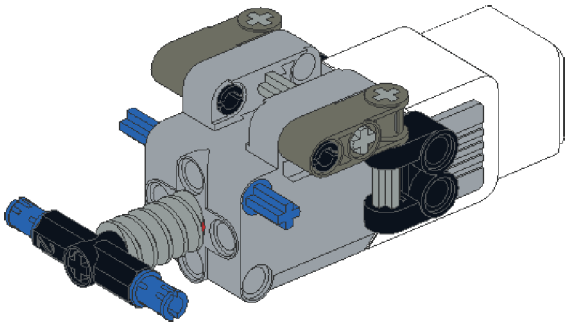
3



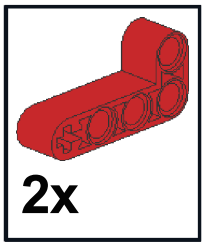


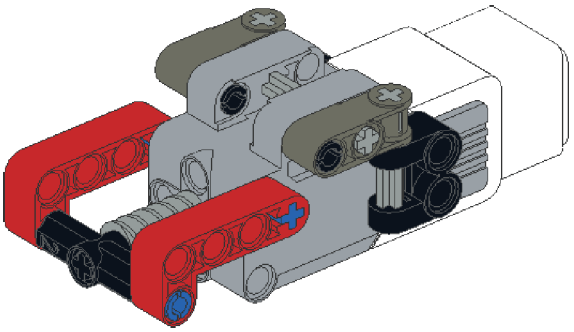
4



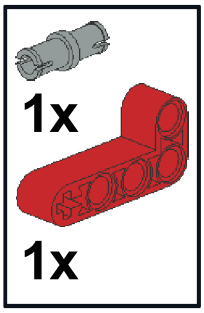


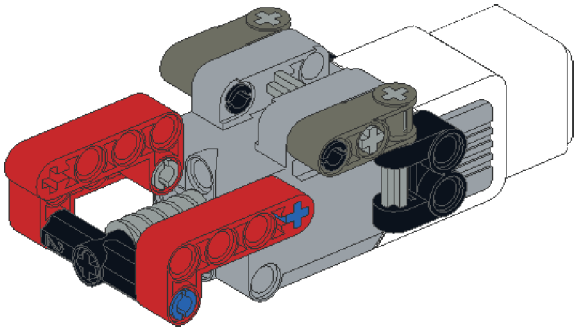
5

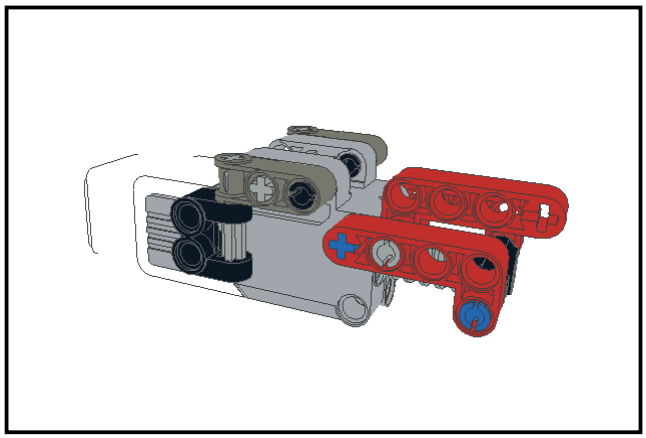




6

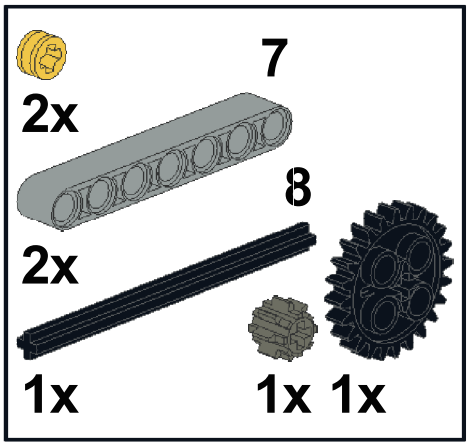


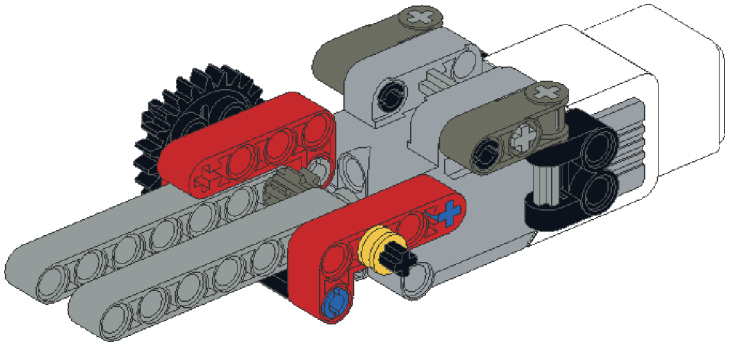


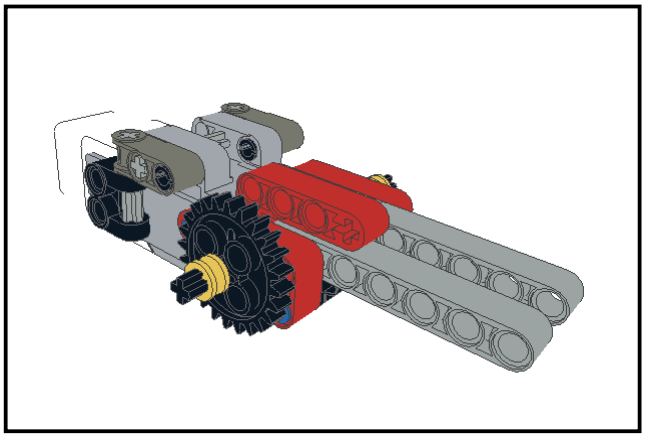


反対側からみた図

7

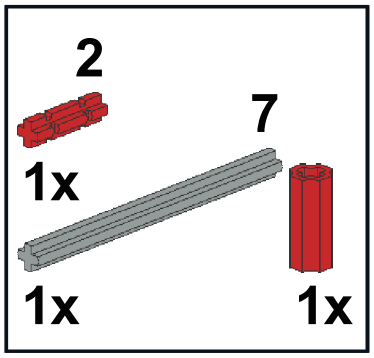


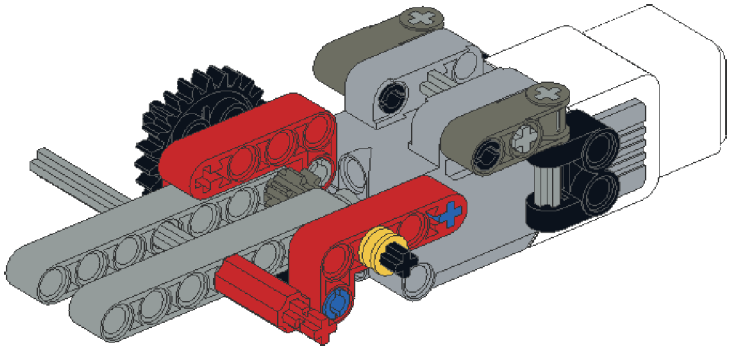


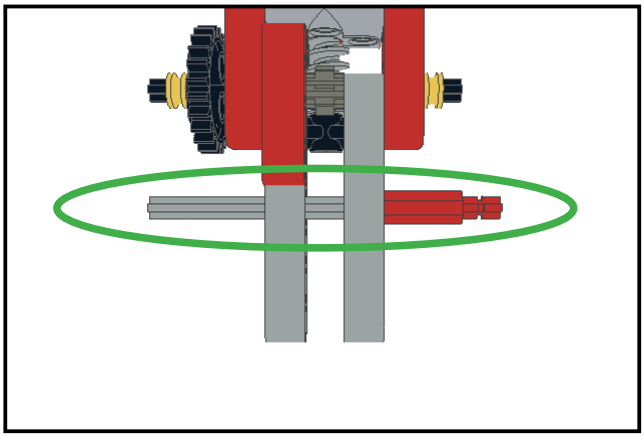


反対側からみた図

8

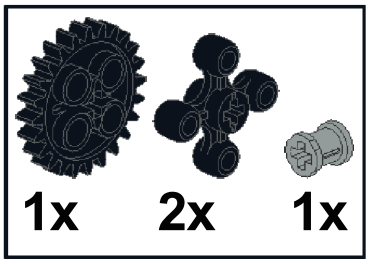


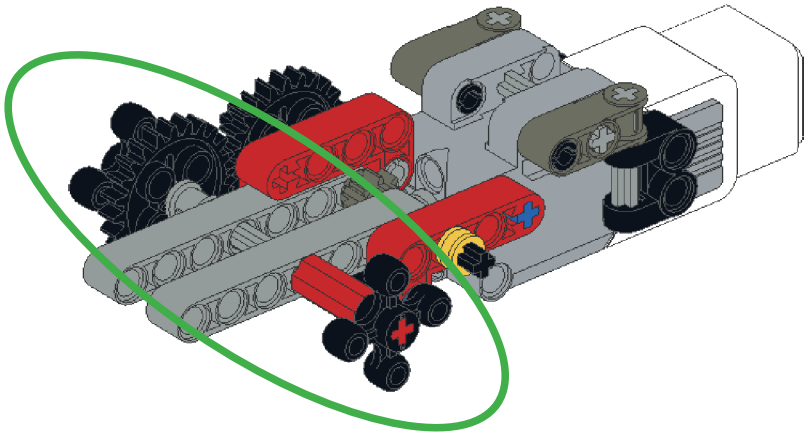




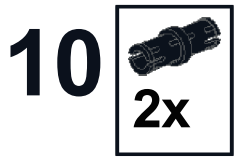
上側からみた図

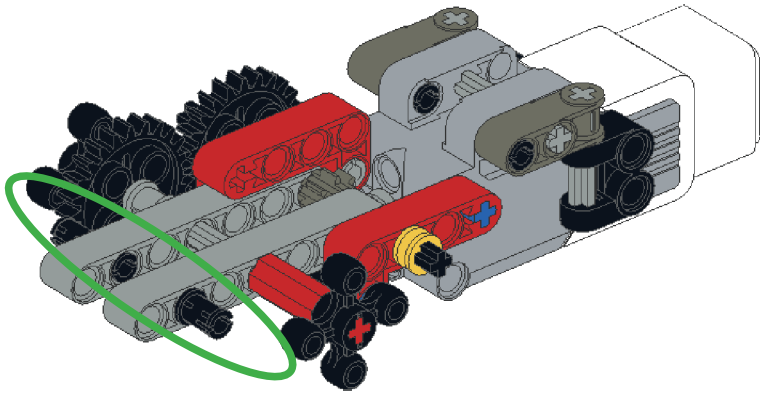
9



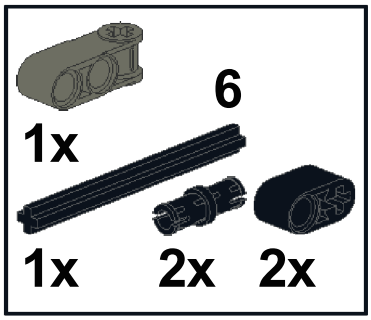


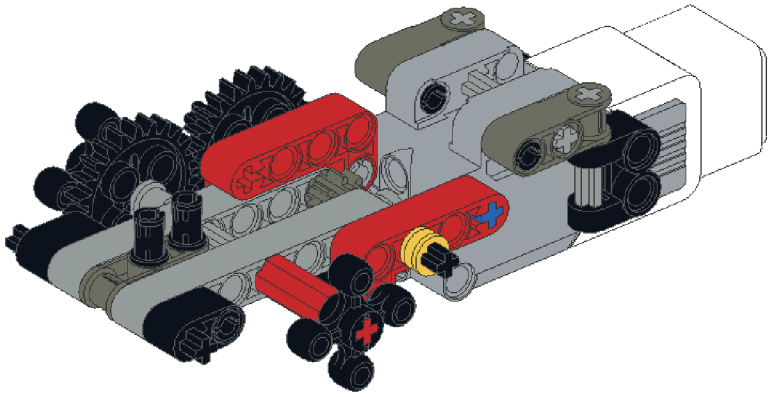
10



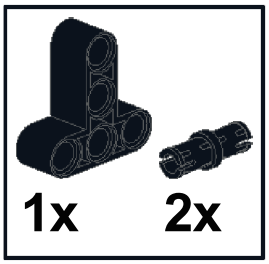


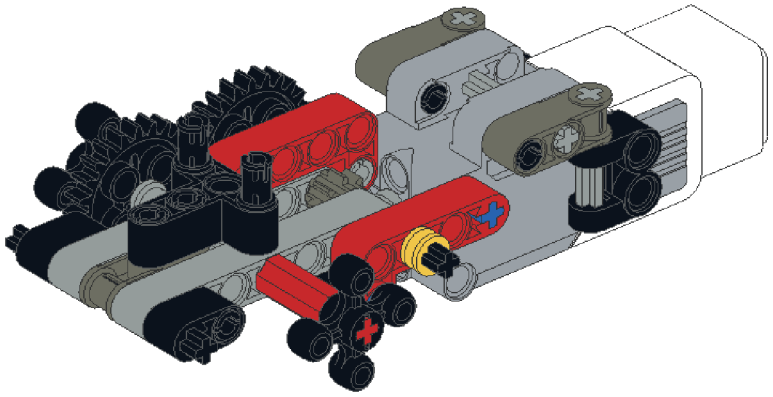
11



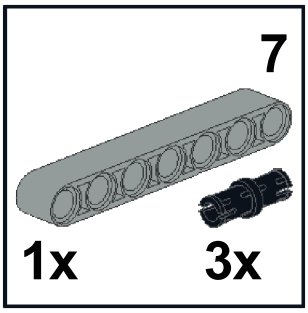


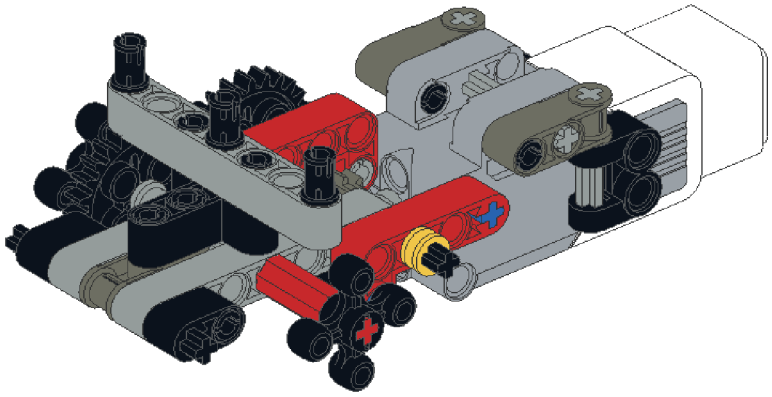
12



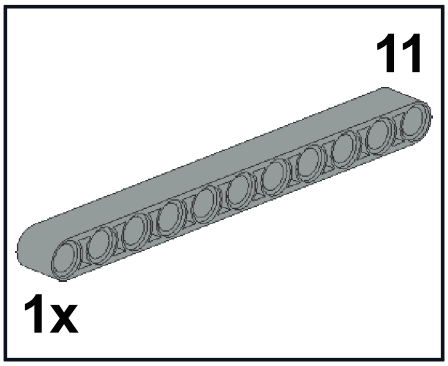


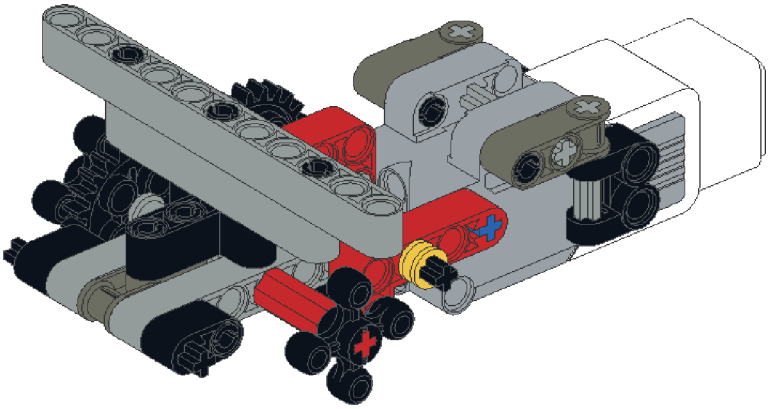
13



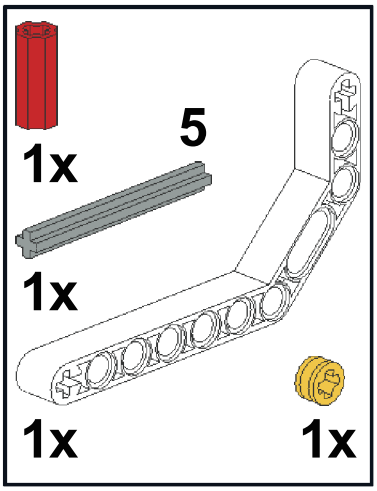


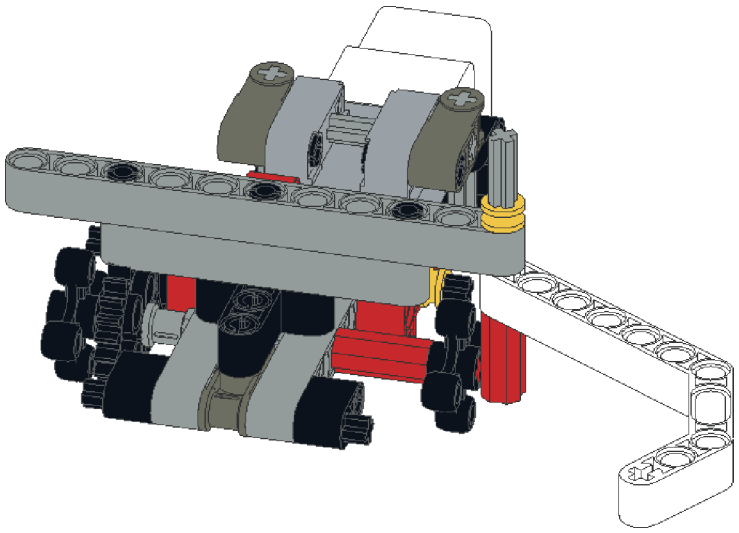
14



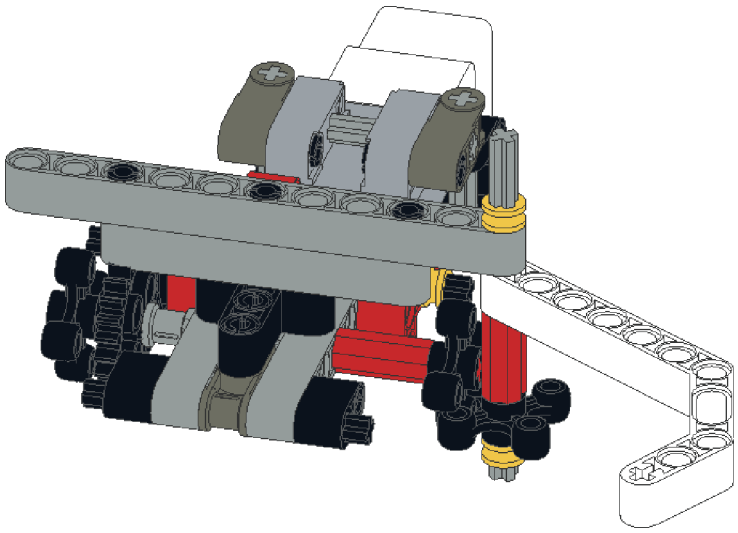


15

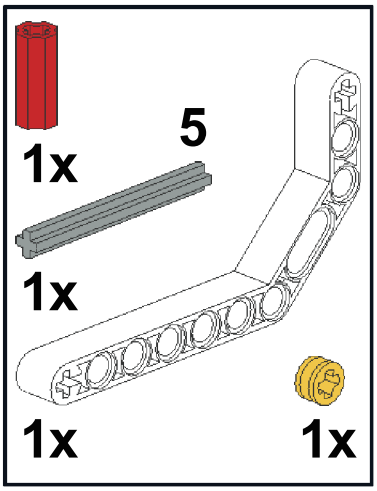


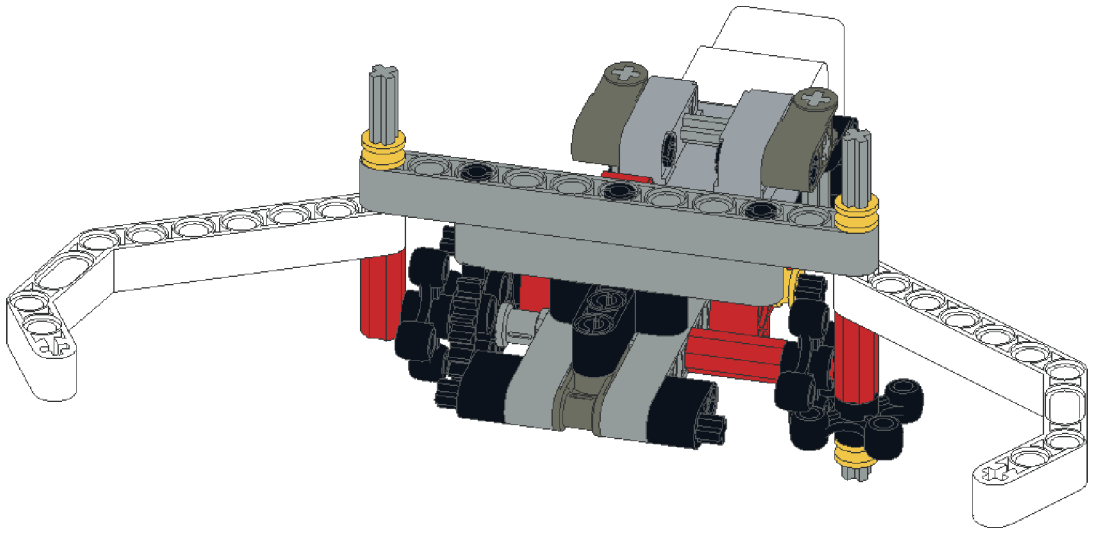


16

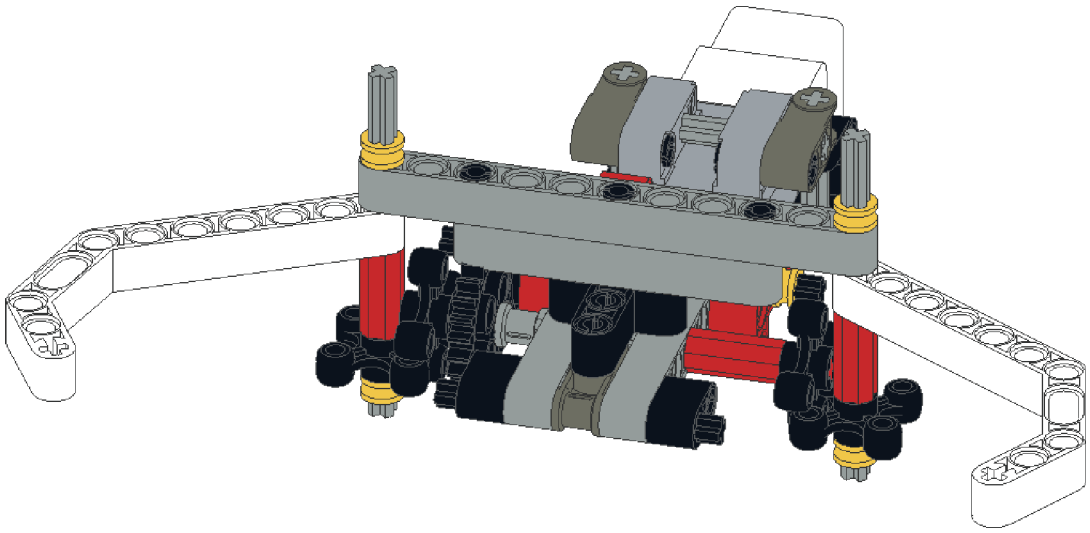


17

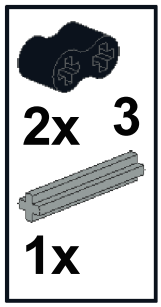


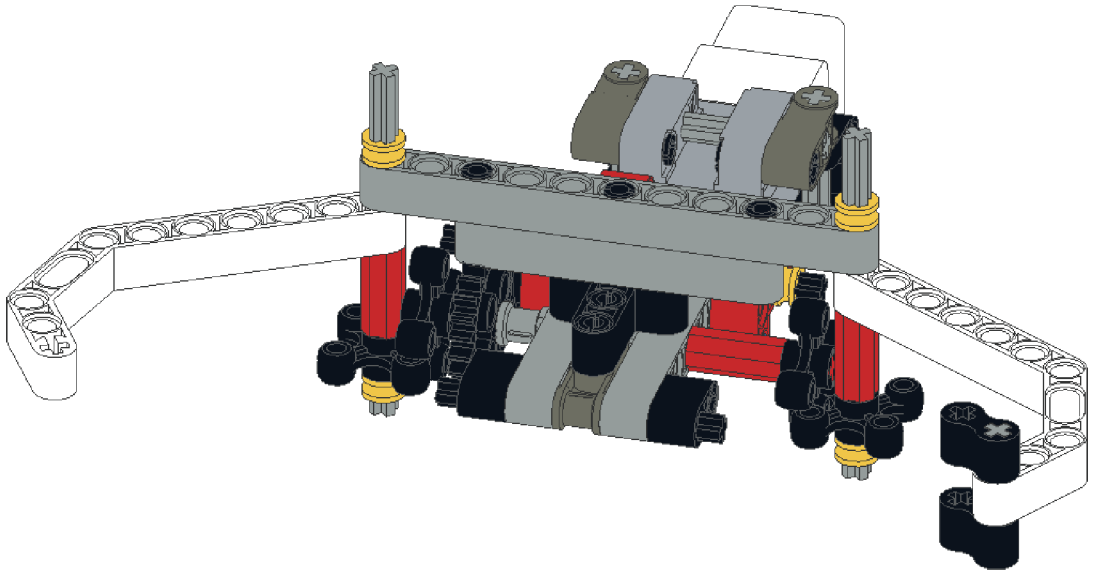


18

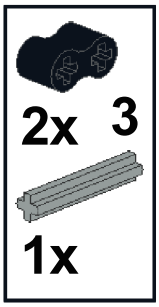


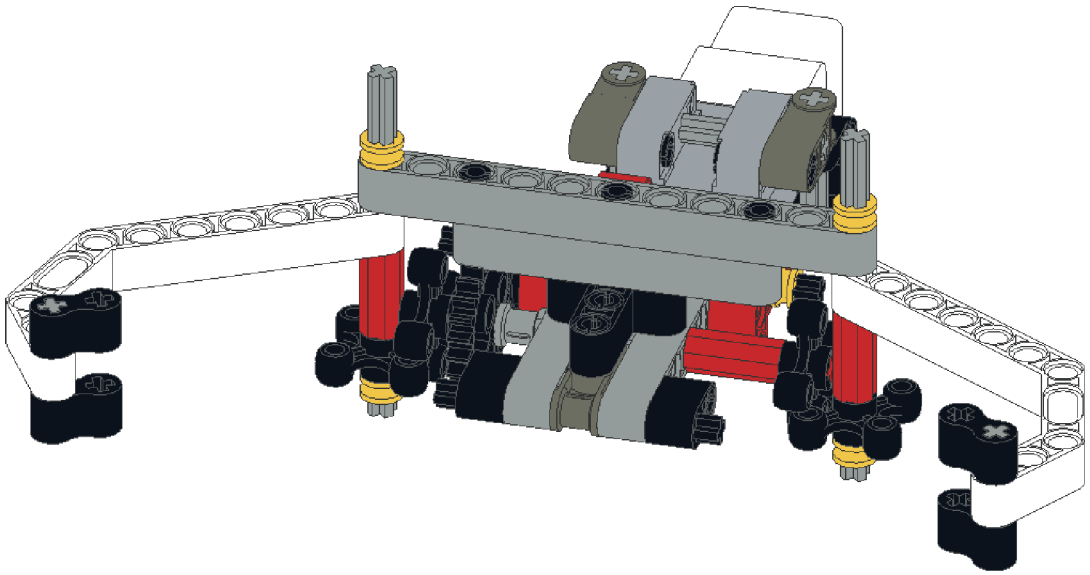
19



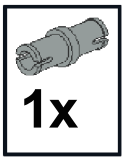


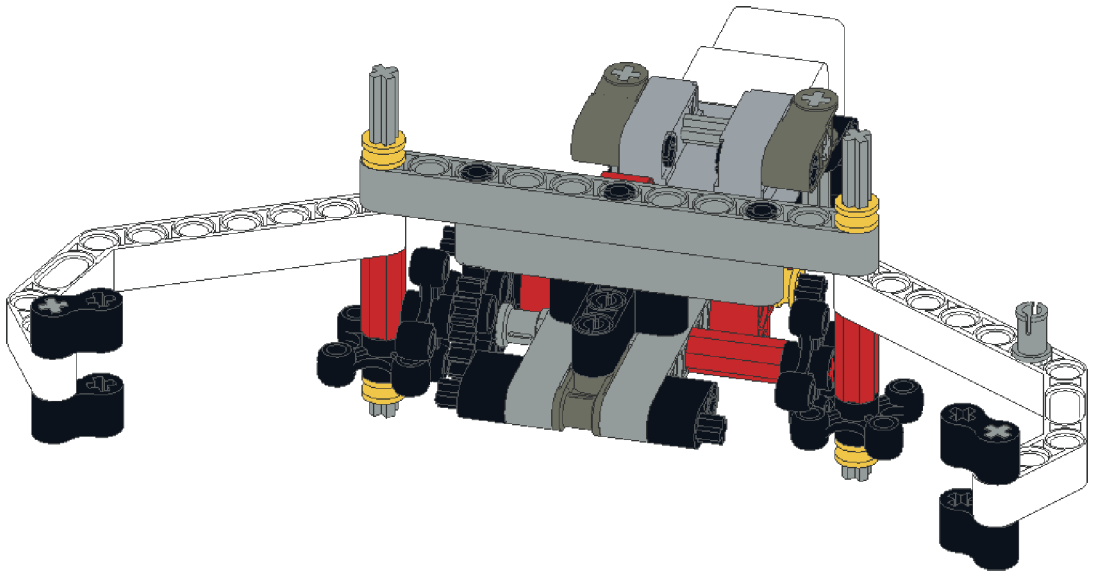
20



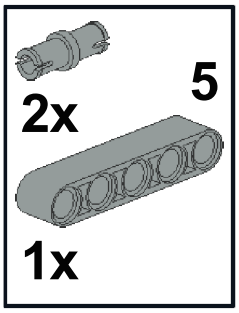


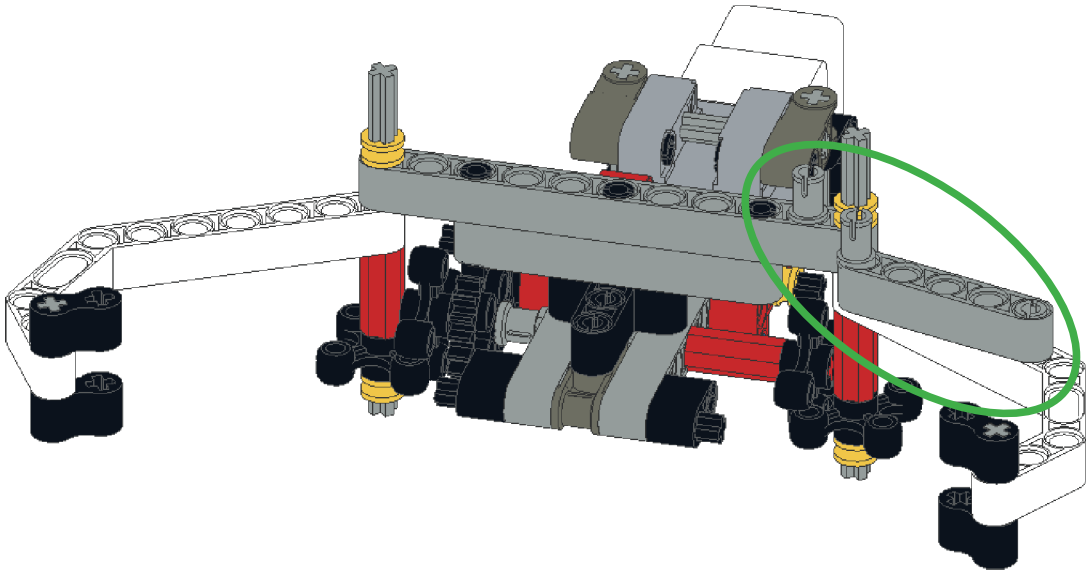
21



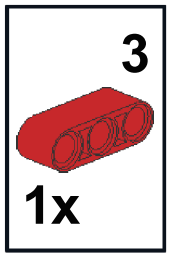


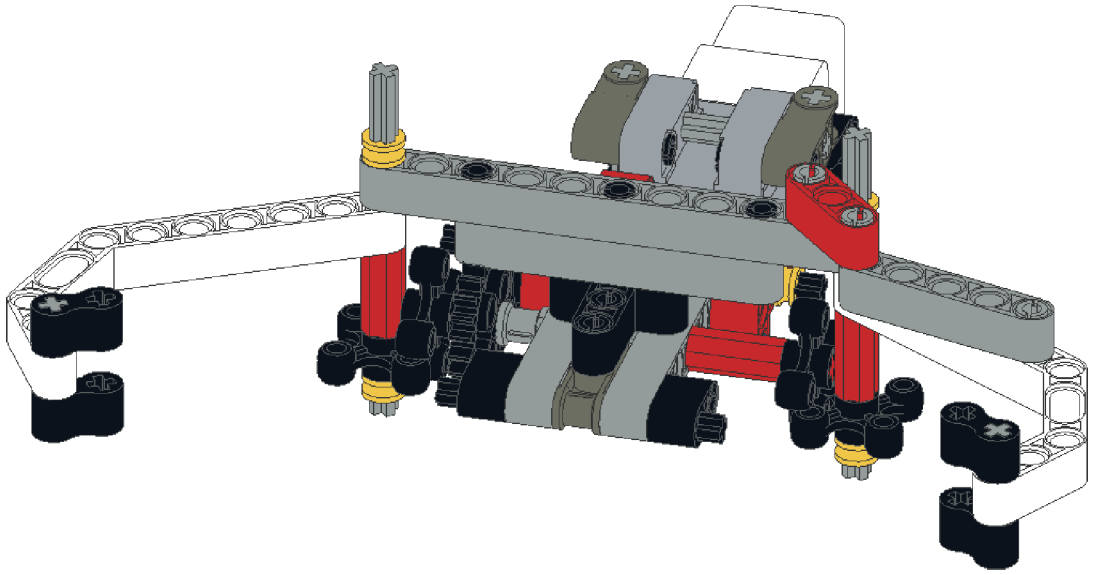
22

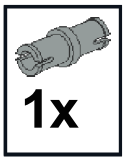


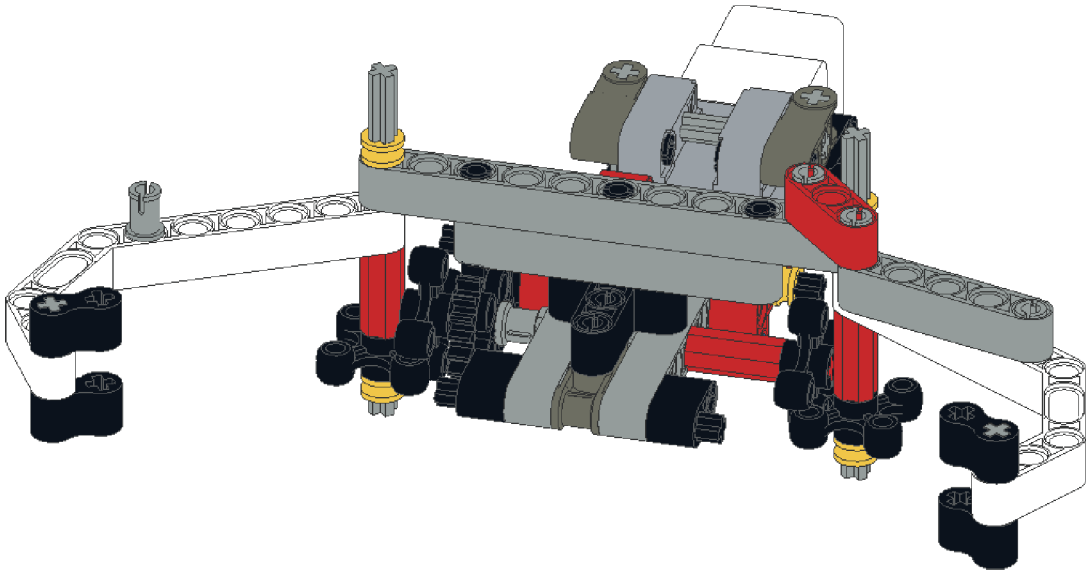


23

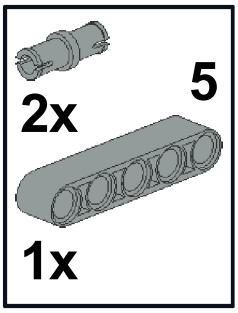


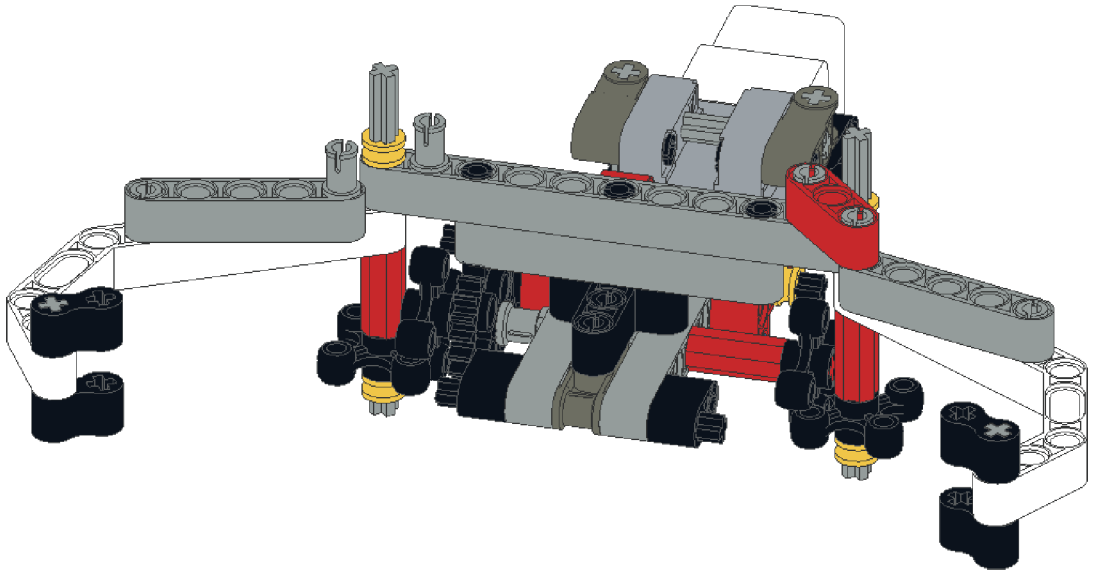


24 

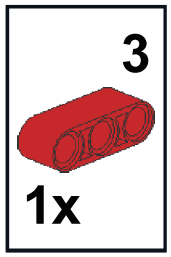


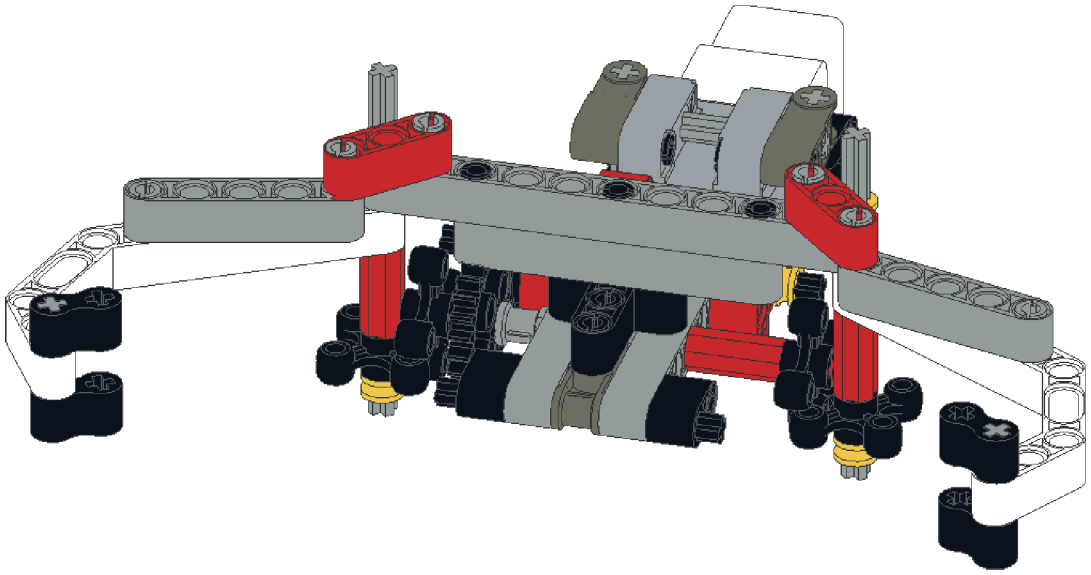
25



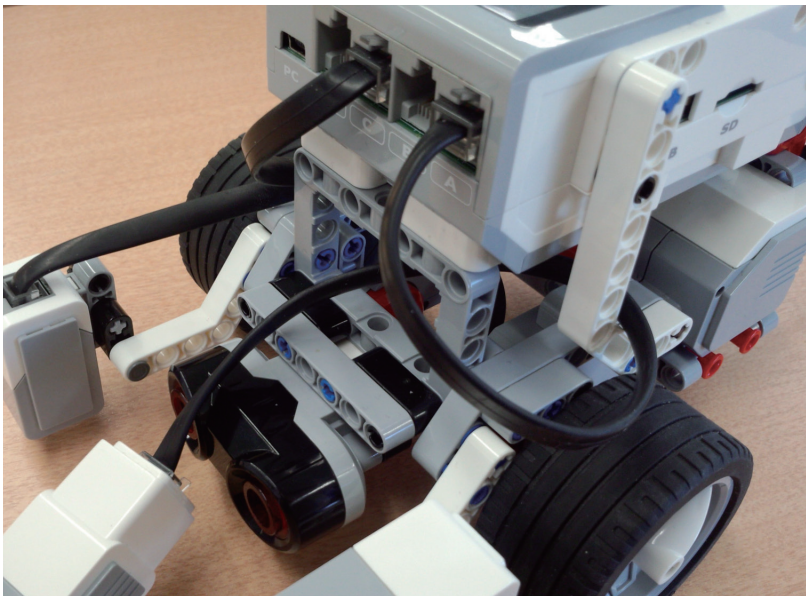


26

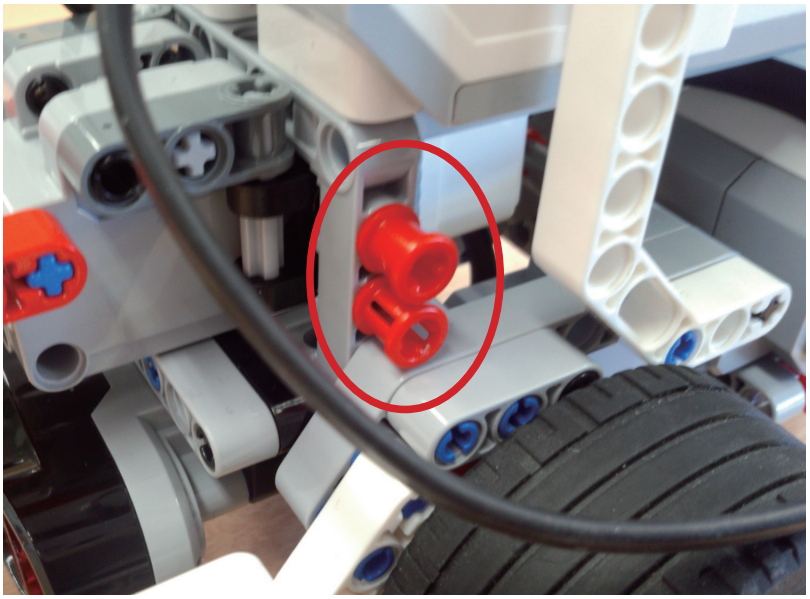




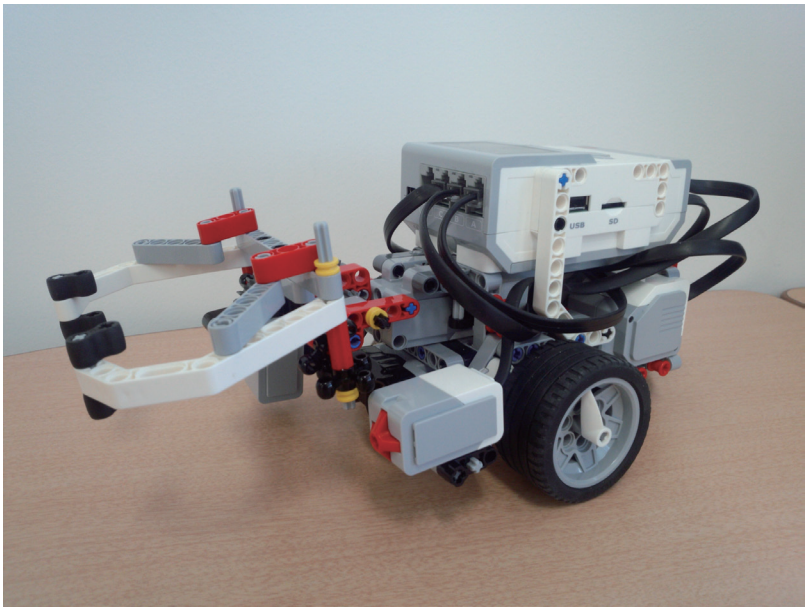
○トレーニングロボットへの取り付け



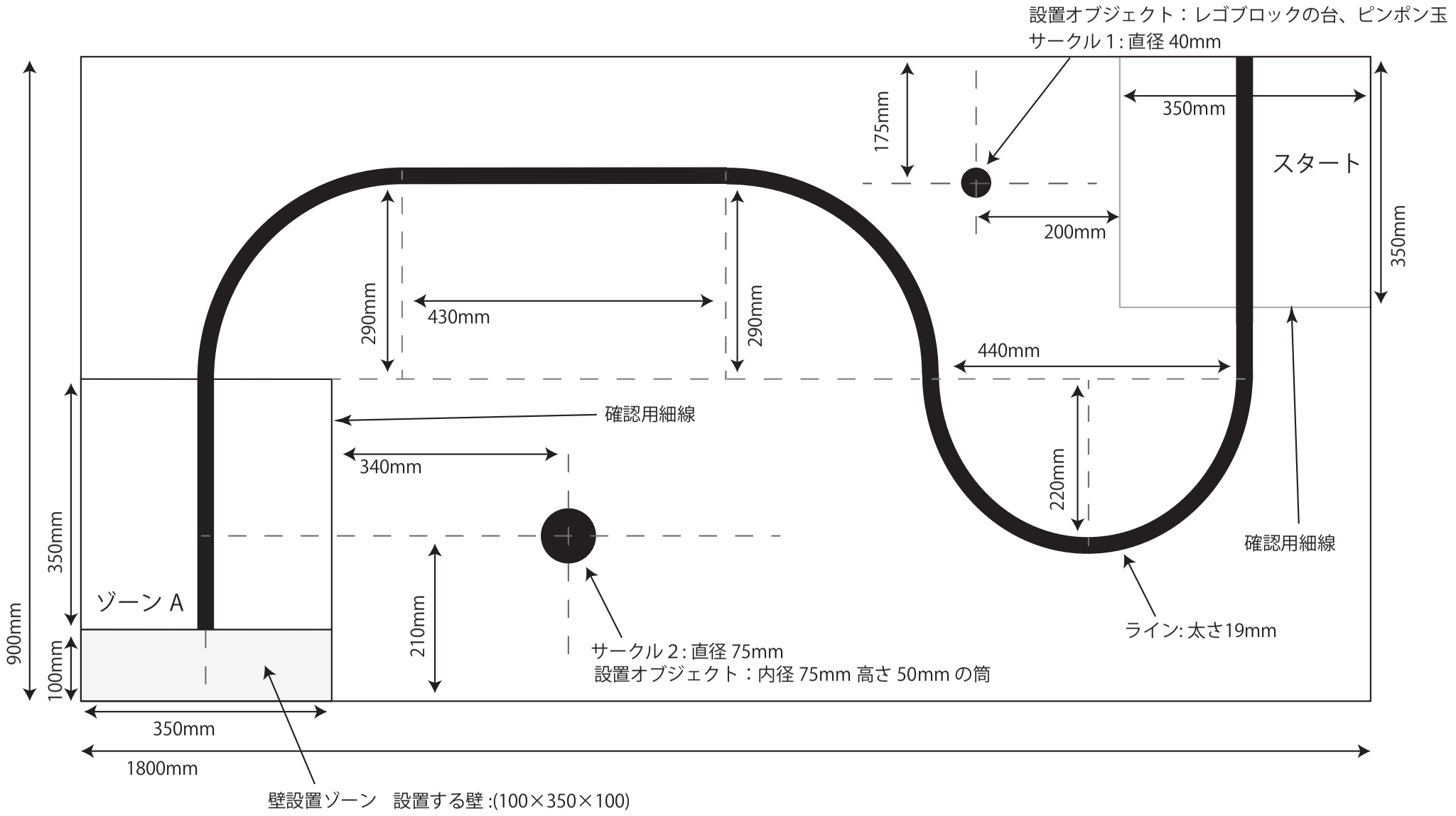
Ｍモーターを Ａポートに接続します。ケーブルはマウンターの間を通して Ｍモーターと接続します。

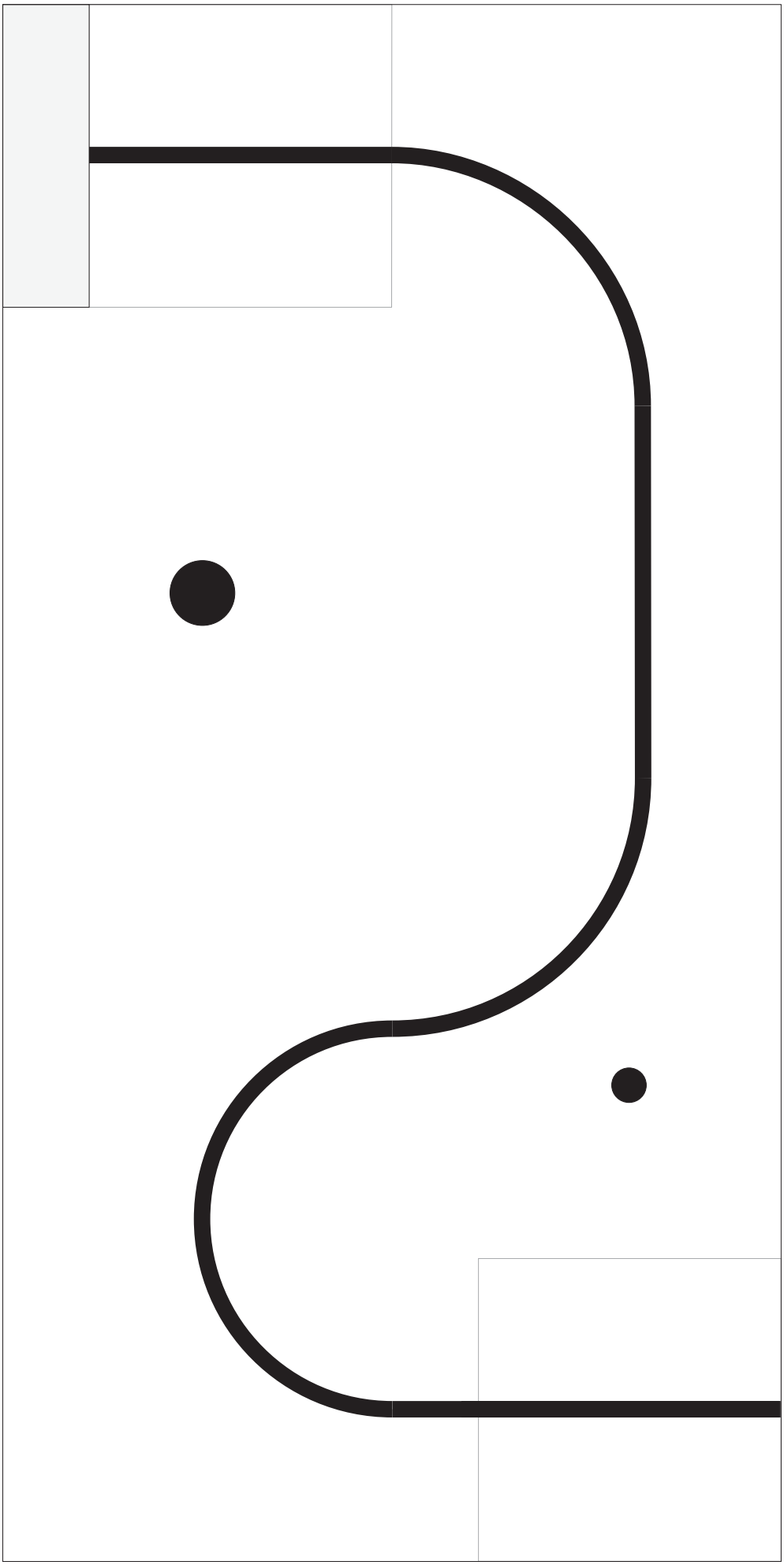


マウンターの間に Ｍモーターを差し込み、ブッシュ付きロングピン（赤）で固定します。反対側も同様です。



完成です。


設置オブジェクト：レゴブロックの台、ピンポン玉
サークル 1 : 直径 40mm
175mm
350mm
スタート
350mm
200mm
290mm
430mm
290mm
440mm
確認用細線
340mm
220mm
確認用細線
350mm
ゾーン A
900mm
100mm
210mm
サークル 2 : 直径 75mm
設置オブジェクト：内径 75mm 高さ 50mm の筒
ライン: 太さ19mm
350mm
1800mm
壁設置ゾーン　設置する壁 :(100×350×100)



平成 27 年度文部科学省委託「東日本大震災からの復興を担う専門人材育成支援事業」
東北の復興･再生を担う自動車組込みエンジニア育成支援プロジェクト

自動車組込み基礎講座演習テキスト

平成 28 年 2 月

東北の復興･再生を担う自動車組込みエンジニア育成支援プロジェクト推進協議会

学校法人日本コンピュータ学園（東北電子専門学校）
〒 980-0013 宮城県仙台市青葉区花京院一丁目3番1号