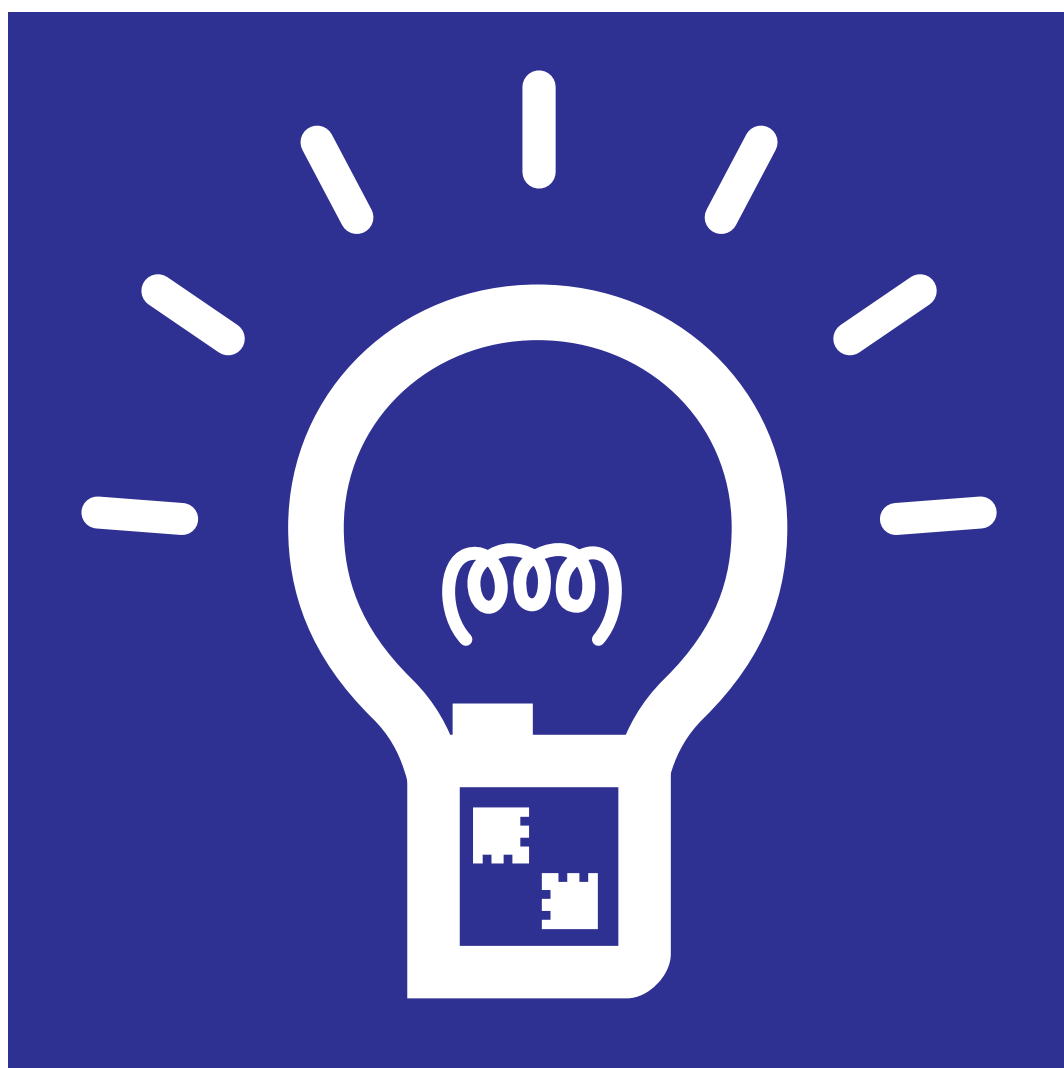




計測と制御 クリエイトキット スタディガイド



目次

※本テキストに掲載の部材表記は、テキスト内で使用する教材の部材表記と統一しております。
このため、一部教科書表記とは異なるものがあります。あらかじめご了承ください。

Stduino ダウンロードサイトのご紹介 1

1. 各種パーツの取扱説明..... 2

- 1.1. アーテックブロック 2
- 1.2. Stduino 3
- 1.3. サーボモーター 5
- 1.4. LED 5
- 1.5. 電子ブザー 5
- 1.6. DC モーター 5
- 1.7. タッチセンサー 6
- 1.8. 光センサー 6
- 1.9. 赤外線フォトリフレクタ 6
- 1.10. 加速度センサー 6

2. 接続方法..... 7

- 2.1. 各種センサー /LED/ ブザーの接続方法 7
- 2.2. DC モーターの接続方法 7
- 2.3. サーボモーターの接続方法 8
- 2.4. 電池ボックスの接続方法 8

3. ブロックプログラミング環境の

取扱説明..... 9

- 3.1. ブロックプログラミング環境の起動 9

4. ソフトウェアの基本動作..... 10

- 4.1. ソフトウェアの説明 10
- 4.2. ブロックを選ぶ 10
- 4.3. ブロックの並べ方 11
- 4.4. ブロックの消し方 12
- 4.5. ファイルの保存 12

5. 入出力設定..... 13

- 5.1. 入出力設定 13・14

授業の流れ..... 15

ワークシート集..... 16 ~ 19

巻末資料

制御ブロックの使い方..... 20 ~ 27

エネルギー変換に関する技術 機構例 ... 28・29

機構をつかった作例..... 30

Stduino ダウンロードサイトのご紹介

スタディーノ

Stduino[®] とは？

モーターやセンサーなど全てのパーツをコネクタで接続できるロボット製作に特化した Arduino 互換基板です。

下記 URL より、Stduino をご利用いただくために必要なソフトウェアや説明書をダウンロードできます。

<http://www.artec-kk.co.jp/stduino/ja/stduino.php>

●ソフトウェア：Stduino ソフトウェアのインストール方法

Stduino 上で動作するプログラムを作成するためのソフトウェアです。お使いの PC の OS バージョンに対応したものをインストールしてください。

●USB デバイスドライバ：デバイスドライバのインストール方法

Stduino と PC を接続するために必要なソフトウェアです。お使いの PC の OS バージョンに対応したものをダウンロードしてください。Raspberry Pi をお使いの場合、USB デバイスドライバのインストールは不要です。

●説明書：Stduino プログラミング環境取扱説明書

●入門ガイド：

①アイコンプログラミング環境（本テキストでは使用しません）

掲載の電子パーツ：LED/ ブザー / プッシュスイッチ / 光センサー / 赤外線フォトリフレクタ / タッチセンサー / 音センサー / 加速度センサー / DC モーター / サーボモーター

②ブロックプログラミング環境（前編）

掲載の電子パーツ：LED/ ブザー / プッシュスイッチ / 光センサー / 赤外線フォトリフレクタ / タッチセンサー / 音センサー

③ブロックプログラミング環境（後編）

掲載の電子パーツ：DC モーター / サーボモーター

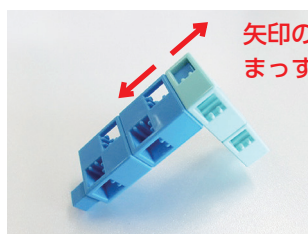
1. 各種パーツの取扱説明

1.1. アーテックブロック

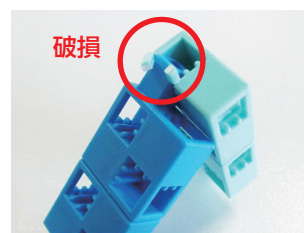
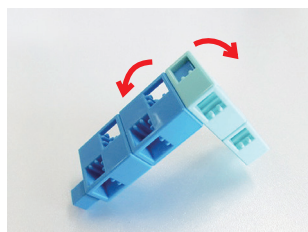
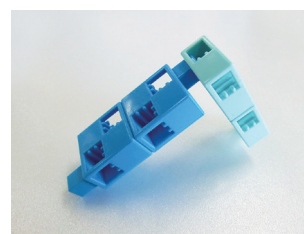
アーテックブロックはその形状の特長から、組立の際の突起の位置を注意して組み立てる必要があります。



アーテックブロックをはずすときは、まっすぐ引っ張るようにしてください。無理に曲げるように力を加えると破損する場合があります。



矢印の方向に
まっすぐ引っ張る



破損

ブロックのつなげかたによって、手で外すのが難しい場合は、ブロックリムーバーを使用して外してください。

アーテックブロックがかたくて
外れないときは...

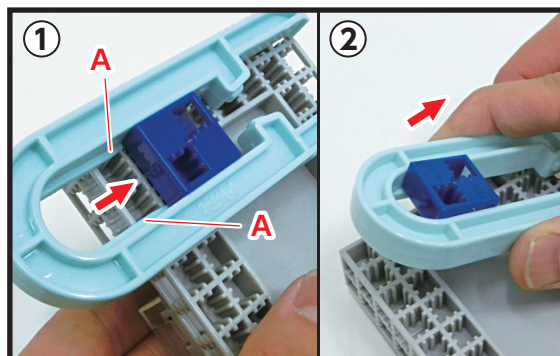
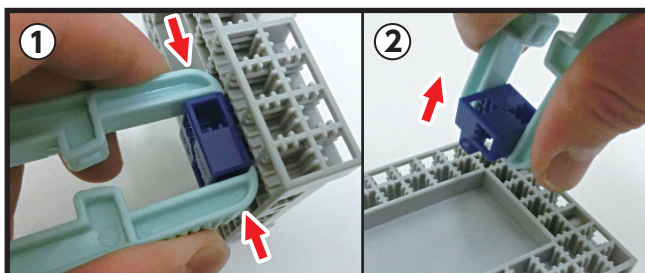
ブロックリムーバー
を使いましょう！

- ① ツメでブロックのつけねを強く挟むとパチッと音が鳴リスキマがひろがる
- ② そのまま引きぬく



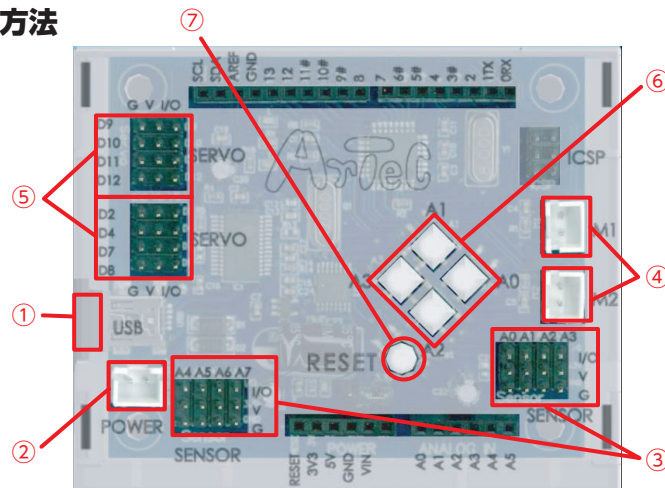
それでも外れないときは...

- ① リムーバーの内側のAの部分ブロックのすき間に入れる
- ② リムーバーをスライドさせそのまま引きぬく



1.2.Studuino

各部の機能および接続方法



① 通信ケーブル接続コネクタ

(USB mini B コネクタ)

パソコンとの通信を行うためにUSBケーブルを接続するコネクタです。コネクタの形状は USB の mini-B 規格です。製品に付属のUSBケーブルだけでなく、市販のUSB ケーブルを接続することもできます。

② 電源コネクタ(外部電源用コネクタ)

付属の電池ボックスを接続して、電源を接続するコネクタです。センサー・LED・ブザー・スイッチは、USB からの供給電源で動作しますが、DC モーターやサーボモーターを動かす場合および、USBケーブルを接続せずにセンサー・LED・ブザー・スイッチを動かす場合は、電源コネクタから電力を供給する必要があります。また、USBケーブルを外し、Studuinoを動作させるときにも電源コネクタから電力を供給する必要があります。

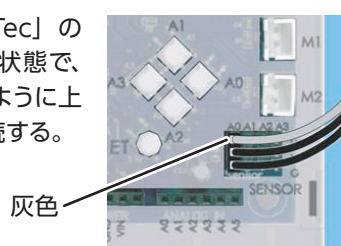
③ センサー・LED・ブザー用コネクタ

(アナログピン用コネクタ)

センサー・LED・ブザーのいずれかを接続するコネクタです。コネクタには、A0～A7 の番号がふられています。
※音センサー・光センサー・赤外線フォトリフレクタは、A0～A7に接続できます。
※タッチセンサー・LED・ブザーはA0～A5に接続できます。
※加速度センサーは、A4とA5にまたがって接続します。
※A0～A3のコネクタ使用時は、プッシュスイッチA0～A3が使用できなくなります。

センサー接続コードを取りつける向きに注意！

透明カバー上の「ArTec」の文字が水平に読める状態で、灰色のコードが図のように上向きになるように接続する。



④ DCモーター用コネクタ

※DC モーター用コネクタM1とサーボモーター用コネクタD2・D4 は同時に使用できません。
※DC モーター用コネクタM2とサーボモーター用コネクタD7・D8は同時に使用できません。

⑤ サーボモーター用コネクタ

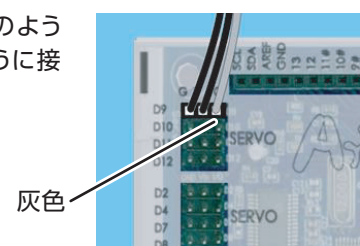
(デジタルピン用コネクタ)

サーボモーターを接続するコネクタです。サーボモーターは8つまで接続可能で、コネクタ名は、D2・D4・D7・D8・D9・D10・D11・D12 です。

※サーボモーターについての詳細は、4ページをご確認ください。

サーボモーターのコードを取りつける向きに注意！

灰色のコードが図のように右向きになるように接続する。



⑥ プッシュスイッチ

ボタンスイッチです。A0～A3 まであります。プッシュスイッチを使用する場合は、センサー・LED・ブザー用コネクタのA0～A3は使用できません。

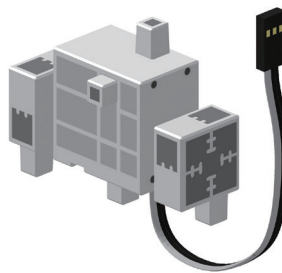
⑦ リセットスイッチ

プログラムをリセットするスイッチです。Studuinoを初期化した後や、パソコンが Studuinoを正しく認識できないとき、その他 Studuinoの動作がおかしいときなどには、このスイッチを押してStuduinoを再起動してください。

1.3.サーボモーター

サーボモーターは指定された角度に回転させることができます。接続コネクタはD2・D4・D7・D8・D9・D10・D11・D12が対応しています。

※サーボモーターを動かす場合はUSBケーブル接続時でも電源コネクタから電力を供給する必要があります。



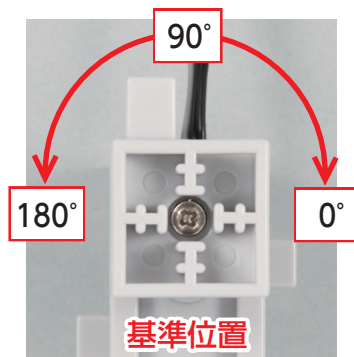
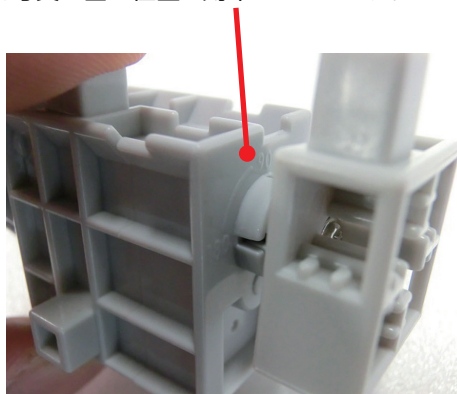
サーボモーターの向きに注意!

実際に電気信号により回転する方とネジで留められているだけのブロックである方を間違えないように組み立ててください。

サーボモーターの回転について

サーボモーターは基準位置を90°として、左右に0°~180°の範囲で回転します。

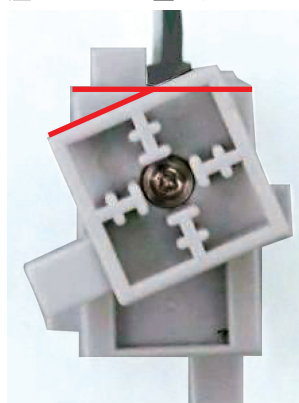
回転の方向は、写真の図の位置に刻印されています。



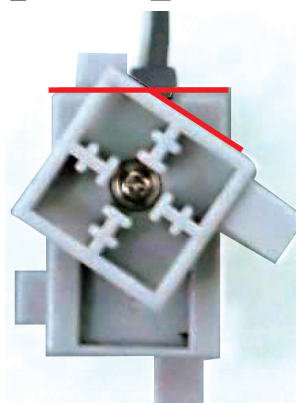
サーボモーターのスリップ機構について

内部ギヤ破損を防ぐため、軸部に一定以上の負荷がかかるとカチカチという音とともにスリップする構造となっています。何度もスリップが起こる場合は負荷がかからないように組立を見直す必要があります。スリップを起こすと、サーボモーターの基準位置がずれてしまい、ロボットを組み立てた時の動きに支障がでます。以下の確認を行ったうえで、ずれた方向と逆方向に手動でスリップさせて矯正してください。※むやみにスリップさせないでください。サーボモーターの劣化、破損の原因となります。

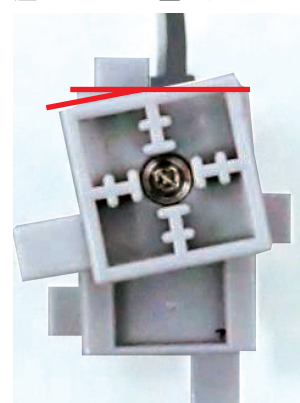
左いっぱい回したとき



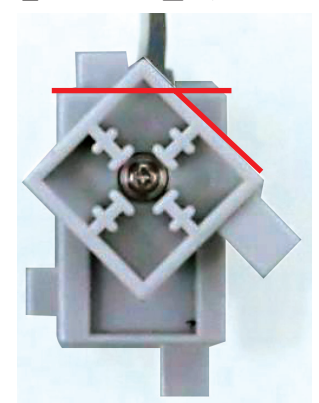
右いっぱい回したとき



左いっぱい回したとき



右いっぱい回したとき



正常：可動域が左右対称

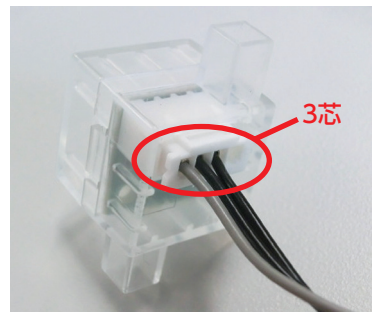
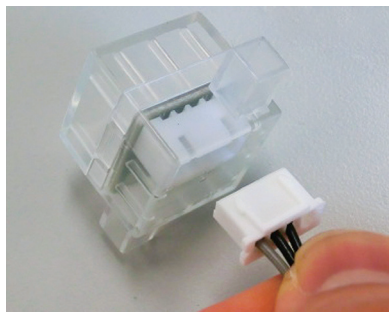
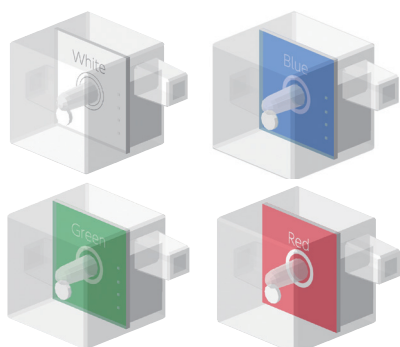


異常：可動域が左右対称でない

上の図の場合、右回転方向にスリップしているため、左回転方向いっぱい回して、負荷をかけて、正常位置に戻すように矯正する必要があります。

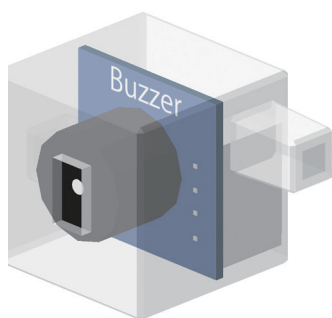
1.4.LED

LEDは白、青、緑、赤の4色あり、基板の色および英語表記の印刷で識別できます。線が3本出ている3芯のセンサー接続コードでStuduinoと接続します。接続コネクタはA0～A5が対応しています。



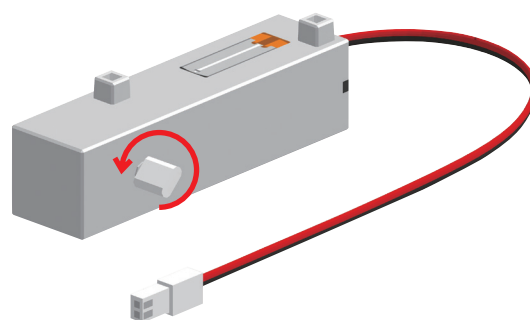
1.5. 電子ブザー

電子ブザーは電気信号を音に変換します。赤外線フォトリフレクタと混同しやすいので、注意してください。英語表記の印刷で識別できます。3芯のセンサー接続コードにてStuduinoと接続します。接続コネクタはA0～A5が対応しています。



1.6.DC モーター

DCモーターは電流の流れる量や時間で、モーターの回転する速さや時間をコントロールできます。内部には複数のギヤが入っており、モーターの回転を軸へ伝えていきます。接続コネクタはM1・M2 が対応しています。※DC モーターを動かす場合はUSB ケーブル接続時でも、電源コネクタから電力を供給する必要があります。



DCモーターの回転方向

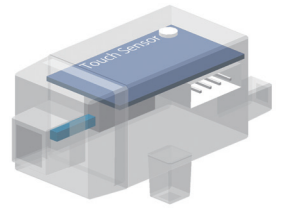
右図の矢印の向きに回転することを正転、反対方向に回転することを逆転と表記しています。

DCモーターのスリップ機構について

内部ギヤ破損を防ぐため、軸部に一定以上の負荷がかかるとカチカチという音とともに空回りする構造となっています。スリップが起こる場合は負荷がかからないように組み立てを見直す必要があります。スリップを長時間起こし続けると、摩耗によりスリップが起こりやすくなり小さな負荷でもスリップが起こるようになってしまいます。

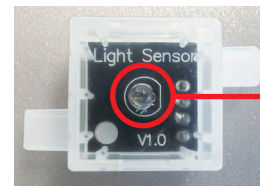
1.7. タッチセンサー

タッチセンサーは先端にスイッチのついた物理センサーです。スイッチを押しこんだときにONになり、離すとOFFになります。3芯のセンサー接続コードでStuduinoと接続します。接続コネクタはA0～A5が対応しています。



1.8. 光センサー

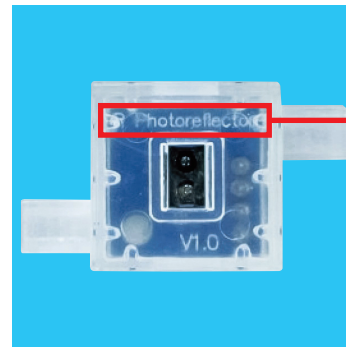
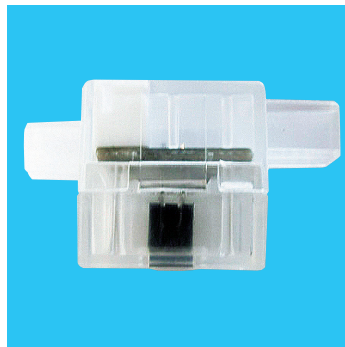
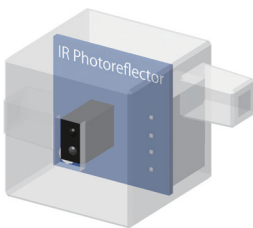
光センサーは光を電気に変える「フォトランジスタ」とよばれる電子部品によって光の強弱を判別します。明るいほどセンサーの値が大きくなります。見た目はLED と酷似していますが、基板の色(黒)および英語表記の印刷で識別します。3芯のセンサー接続コードでStuduino と接続します。接続コネクタはA0～A7が対応しています。



フォトランジスタ

1.9. 赤外線フォトリフレクタ

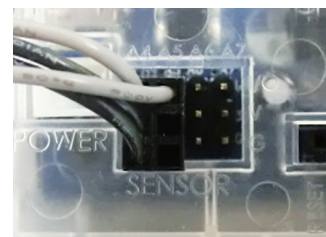
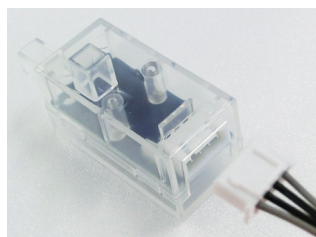
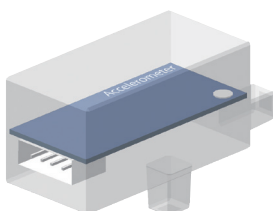
形状がブザーと混同しやすいため注意が必要です。英語表記の印刷で識別できます。3芯のセンサー接続コードでStuduino と接続します。接続コネクタはA0～A7 が対応しています。



IR Photorelector
という文字が書かれています

1.10. 加速度センサー

センサーの傾きや動きの変化を数値化するセンサです。4芯のセンサー接続コードでStuduinoと接続します。このセンサーの接続コネクタはA4 とA5 にまたがるように接続します。

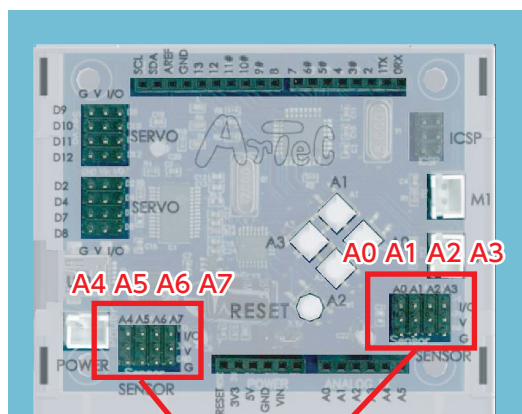
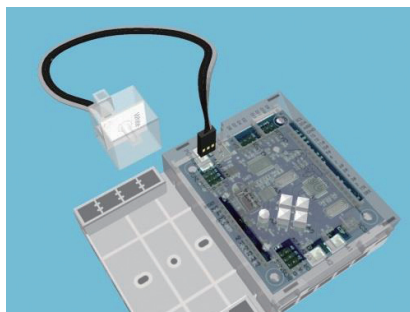
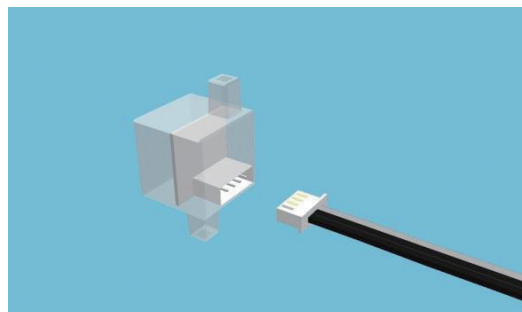


2. 接続方法

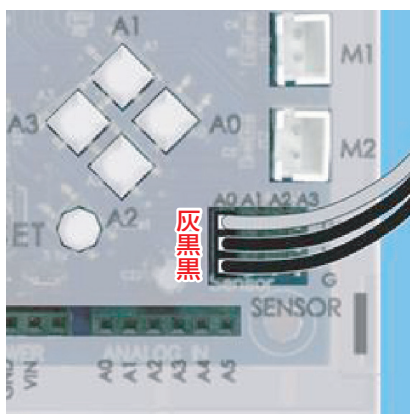
2.1. 各種センサー /LED/ ブザーの接続方法

センサーパーツに下図のようにセンサー接続コードを接続し、Studuino基板のセンサー/LED/ブザー用コネクタに接続します。

※加速度センサー以外のセンサーは3芯のセンサー接続コードを使用し、加速度センサーは4芯のセンサー接続コードを使用してください。



センサー/LED/ブザー用コネクタ



センサー接続コードは左図のように灰色のコードが上側にくるように接続します。

※音センサー、光センサー、赤外線フォトリフレクタはA0～A7に接続できます。

※タッチセンサー、LED、ブザーはA0～A5に接続できます。

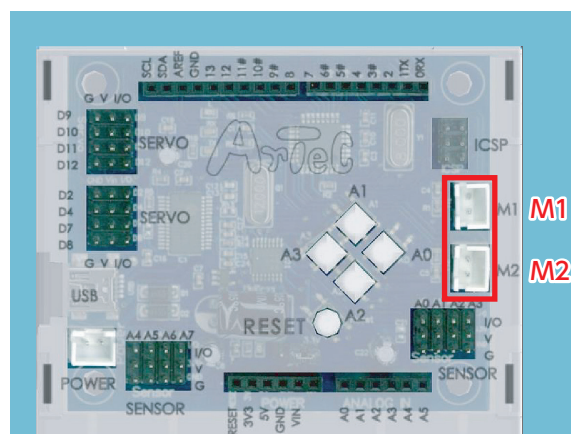
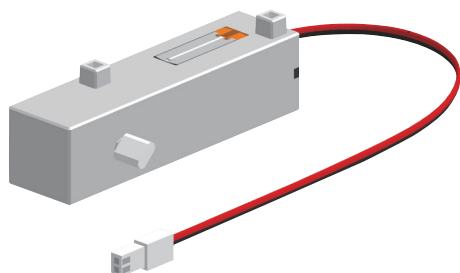
※加速度センサーは、A4とA5にまたがって接続します。

※A0～A3のコネクタ使用時は、プッシュスイッチA0～A3が使用できなくなります。

2.2.DC モーターの接続方法

DCモーターは2つまで接続可能で、DCモーター用コネクタ(M1、M2)に接続します。

※DCモーターのコネクタは逆方向に接続できない構造になっています。

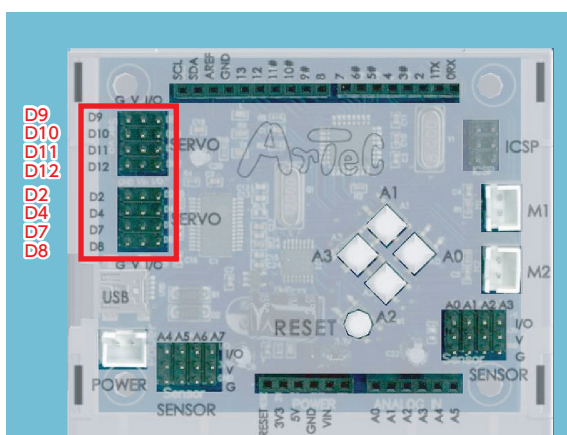
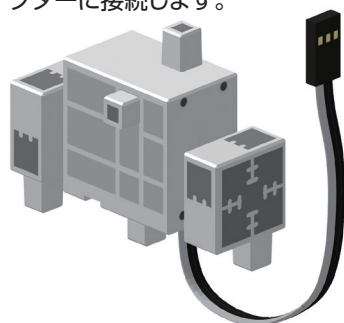


※DCモーター用コネクタM1と、サーボモーター用コネクタD2、D4は同時に使用できません。

※DCモーター用コネクタM2と、サーボモーター用コネクタD7、D8は同時に使用できません。

2.3. サーボモーターの接続方法

サーボモーターは8つまで接続可能で、サーボモーター用コネクタに接続します。



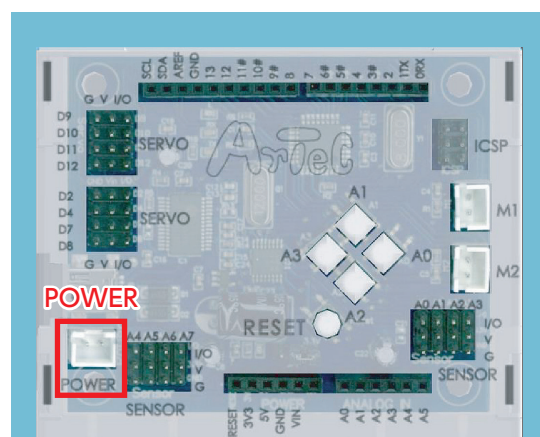
サーボモーターのコードは上図のように灰色のコードが右側にくるように接続します。

※DCモーター用コネクタM1と、サーボモーター用コネクタD2、D4は同時に使用できません。

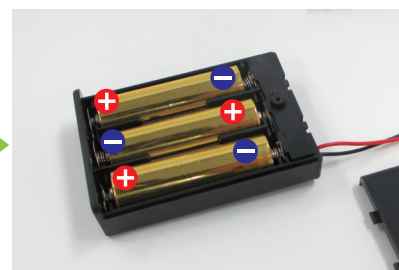
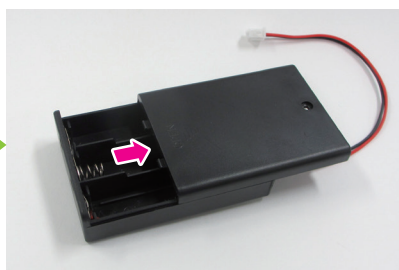
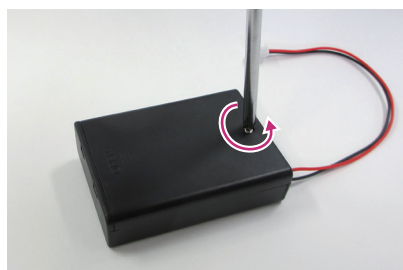
※DCモーター用コネクタM2と、サーボモーター用コネクタD7、D8は同時に使用できません。

2.4. 電池ボックスの接続方法

電池ボックスは電源コネクタ（POWER）に接続します。センサー/LED/ブザーはUSBからの供給電源で動作しますが、DCモーターやサーボモーターを動かす場合は、電源コネクタから電力を供給する必要があります。また、USBケーブルを外して動作させるときにも、電源コネクタから電力を供給する必要があります。



電池の挿入方法



⚠️ + プラス - マイナスの向きに注意しましょう

3. ブロックプログラミング環境の取扱説明

3.1. ブロックプログラミング環境の起動

Windows の場合

① デスクトップにある右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。Windows では2種類のプログラム環境が選択できます。本講座では、ブロックプログラミング環境を選択してください。



③ 3つのバージョンのブロックプログラミング環境が選択できます。それぞれの用途にあわせて使用するブロックプログラミング環境を選択して起動してください。

※「キャラクター」「電気実験」の2つのバージョンについては、「Studuino プログラミング環境 Ver1.5.0.0」以降に追加された機能になります。Ver1.4.1.1以前のソフトウェアをご使用の場合は、ソフトウェアのアップデートを行ってください。



Mac OS X の場合

インストールした「StuduinoBPE.app」をダブルクリックすると、ブロックプログラミング環境が起動します。

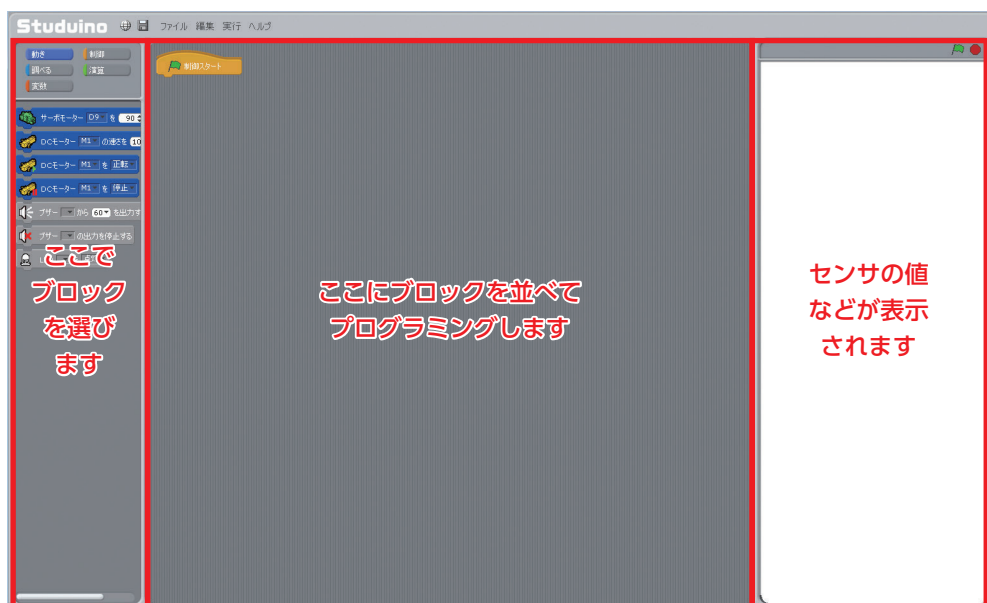
※ Mac OS X では「ロボット」のみ使用可能です。「キャラクター」「電気実験」は未対応です。



4. ソフトウェアの基本動作

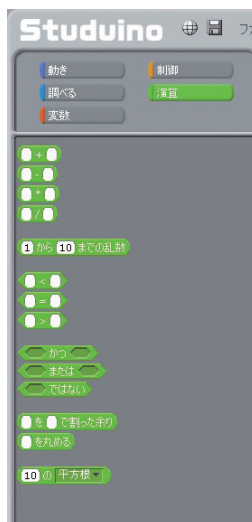
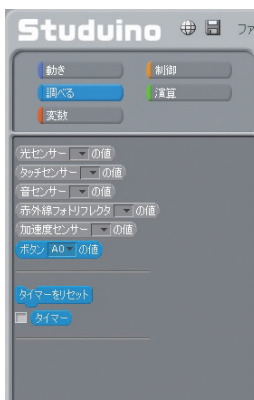
4.1. ソフトウェアの説明

ロボットを動かすプログラムをつくるソフトウェアを「ブロックプログラミング環境」といいます。このソフトでは、さまざまな命令をブロックのように並べてつなぎ合わせ、プログラムをつくることができます。



4.2. ブロックを選ぶ

左上の5つのボタンを押すと、選べるブロックが切り替わります。

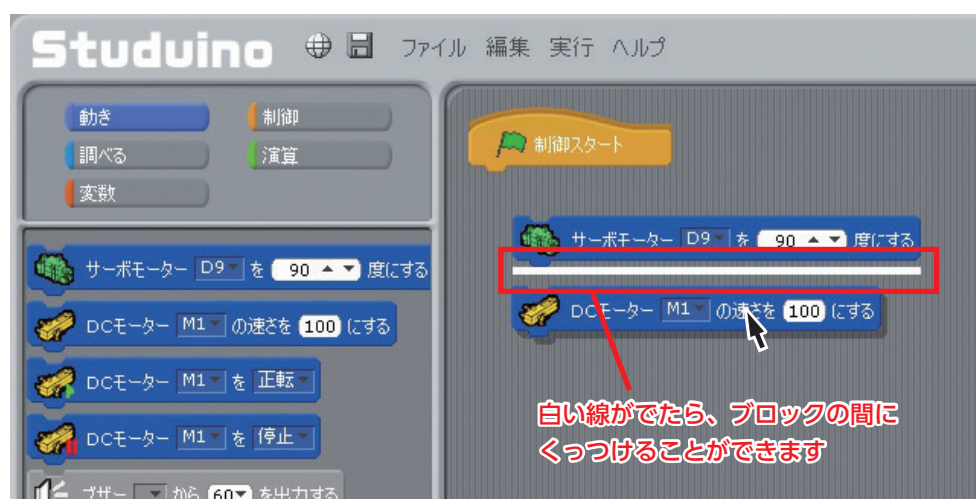


4.3. ブロックの並べ方

ブロックを選んでマウスで左クリックしながら、真ん中の画面の上にドラッグして、ドロップすると、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、右の図のように白い線があらわれて、ブロック同士をつなぐことができます。



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。そのとき、下につながっているブロックも一緒に動きます。



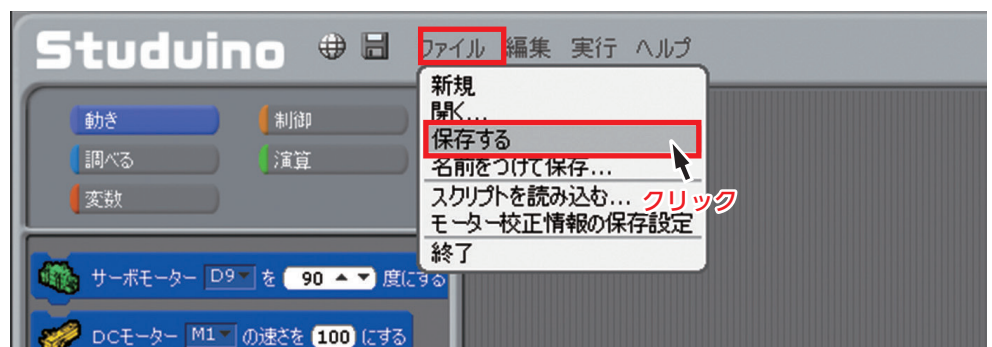
4.4. ブロックの消し方

消したいブロックを左端のスペースにドラッグ&ドロップするとブロックを消すことができます。

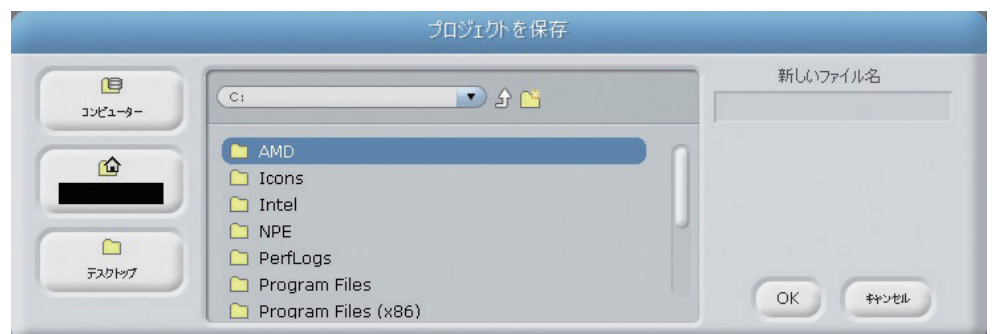


4.5. ファイルの保存

① メニューバーの「ファイル」より、「保存する」を選択します。



② 以下のようなウィンドウが現れたら、保存したいフォルダを指定し、ファイル名を付けて保存します。

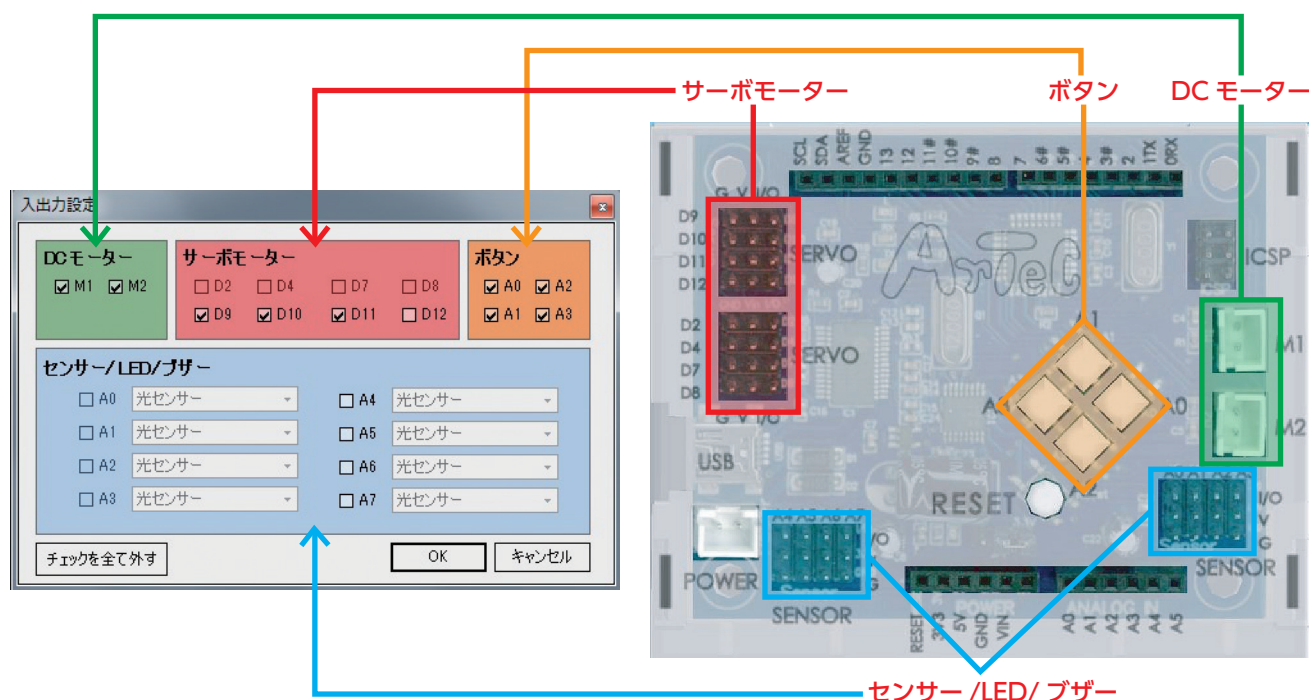


5. 入出力設定

5.1. 入出力設定

Studuinoに接続しているパーツ情報を、設定します。メニューバーの「編集」より、「入出力設定」を選択すると「入出力設定ダイアログボックス」が表示されます。

このダイアログボックスのチェックボックスは、下記のように Studuinoのコネクタに対応していますので、パーツが接続されている Studuinoのコネクタに対応するチェックボックスにチェックを入れてください。



また、7～8ページにもあるように、下記コネクタの組み合わせは同時に使用できません。

- DCモーター用コネクタM1と、サーボモーター用コネクタD2、D4
- DCモーター用コネクタM2と、サーボモーター用コネクタD7、D8
- ボタン(=プッシュスイッチ)A0～A3と、センサー/LED/ブザー用コネクタA0～A3

この組み合わせは「入出力設定ダイアログボックス」でも同時に設定できなくなっていますので、一方を使用する場合は片方を無効にしてください。たとえば、サーボモーターのD2、D4を使用する場合は、DCモーターのM1のチェックボックスのチェックを外すとD2、D4が有効になり、チェックボックスにチェックを入れることができます。

注意!

コネクタを2つ使用する加速度センサーに関しては、対応したコネクタ全てにチェックが入っている必要があります。

複数コネクタを使用するデバイスの
コネクタ組み合わせ

接続パーツ	コネクタの組み合わせ
加速度センサー	A4、A5

上表の「コネクタの組み合わせ」に対応するコネクタ名のチェックボックスにチェックを入れた状態でアイテム選択を行うと、対応する接続パーツが表示され、パーツを選択すると自動的に他のコネクタのアイテムも選択されます。

入出力設定

DCモーター
☒ M1 ☒ M2

サーボモーター
☐ D2 ☐ D4 ☐ D7 ☐ D8
☒ D9 ☒ D10 ☒ D11 ☒ D12

ボタン
☒ A0 ☒ A2
☒ A1 ☒ A3

センサー/LED/プザー

☐ A0 光センサー ☒ A4 光センサー
☐ A1 光センサー ☒ A5 光センサー
☐ A2 光センサー ☒ A6 タッチセンサー
☐ A3 光センサー ☒ A7 音センサー
赤外線フォトリフレクタ
加速度センサー
LED
プザー

チェックを全て外す OK キャンセル

A4で加速度センサーを選択すると、自動的にA5でも選択されます。

入出力設定

DCモーター
☒ M1 ☒ M2

サーボモーター
☐ D2 ☐ D4 ☐ D7 ☐ D8
☒ D9 ☒ D10 ☒ D11 ☒ D12

ボタン
☒ A0 ☒ A2
☒ A1 ☒ A3

センサー/LED/プザー

☐ A0 光センサー ☒ A4 加速度センサー
☐ A1 光センサー ☒ A5 加速度センサー
☐ A2 光センサー ☒ A6 光センサー
☐ A3 光センサー ☒ A7 光センサー

チェックを全て外す OK キャンセル

授業例

生活課題型システム*をつくり他者の生活を便利にしよう（全 18 時間）

	テーマ	時間	授業形態	学習活動	備考	参考資料
1	生活の中の計測・制御システム	第1時	言語学習	「計測・制御システム」とはどのようなものか、「計測・制御システム」が身の回りでどのように使われているかを調べ、まとめる。		調査シート
		第2時	言語学習	第1時で調査した計測・制御システムでは何をセンサーで計測し、何を制御しているのかを調べる。その内容から計測・制御システムが有効な場面とそうでない場面を考える。		
2	センサーを使ったDCモーターやサーボモーターの制御	第3時	実技	Studuinoの使い方を学ぶ。光センサーとDCモーターを使い、明るくなったらDCモーターを正転、暗くなったらDCモーターを逆転するプログラムをつくる。		ブロックプログラミング環境(後編)
		第4時	実技	Studuinoの使い方を学ぶ。音センサーとサーボモーターを使い、音に反応してサーボモーターの角度を変えるプログラムをつくる。		http://www.artec-kk.co.jp/studuino/ja/studuino.php
		第5・6時	実技	DCモーターやサーボモーターを組み合わせたようなものができるか、グループで討論し子どもたちが制作したいものをつくる。DCモーターやサーボモーターの活用例を紹介し合い、次時に行う生活課題型システム*の追求に活かす。		巻末資料 制御ブロックの使い方
3	他者の生活をよりよくする生活課題型システム*を考える	第7時	言語学習	他者の生活を想定し、だれが、どのような場面で、どのような困難があるかアイデア整理シートにまとめる。まとめた内容を基に、作成する生活課題型システム*についてグループで追求する。	事前に3人一組のグループを作成し、グループ内で一つの生活課題型システム*をつくるように指示する。	アイデア設計シート
		第8時	言語学習	作成する生活課題型システム*を具体的にするために、KJ法を用い目的を明確化する。アイデア整理シートを提出する。		
4	生活課題型システム*の実現	第9・10時	実技	Studuinoを使い第8時で検討した製作物を実際に制作する。	グループ内で主な役割(機構面でのリーダー、プログラミング面でのリーダー、プレゼンテーション面でのリーダー)を決め、スムーズに制作できるように指示する。	巻末資料 エネルギー変換に関する技術機構例
5	使用者を意識したシステムの改良	第11時	実技	製作した生活課題型システム*を他者が使いやすくするためにはどうしたらいいか考え、改良する。		
		第12時	実技	実際に使用する者だけではなく、その周囲にいる人にとっても安全なシステムになるように改良する。		
		第13時	言語学習	製作物をクラスの中で発表する。発表グループに対し、現状のシステムに対する意見や改良点を伝え、他者の意見を取り入れる機会とする。		
		第14・15時	実技	これまであがった改良点をもとにシステムを完成させる。		
6	製作した生活課題型システム*の調査、評価	第16時	言語学習	製作した生活課題型システム*を無料動画サイトに公開するために、説明動画を撮影し、サイトに公開する。		
		第17時	言語学習	製作した生活課題型システム*を作品発表シートにまとめる。		発表シート
7	今後の計測・制御について考える	第18時	言語学習	調査したものを発表し合う。異なるグループで調査結果を共有し合う。		

この授業例は、2015年度に静岡大学教育学部附属静岡中学校 本部康司先生が実施した授業をもとに作成しています。

*生活課題型システム = 自分以外の生活をよりよくする課題を達成するシステム

ワークシート集

ワークシートはエクセルで作成されていますので、
ご自由に編集してお使いいただけます。

ダウンロードはこちら↓

<http://www.artec-kk.co.jp/dl/createkit/>

生活の中の計測・制御システム 調査シート

生活の中に存在する「計測・制御システム」を使っている製品や装置をあげ、どのような情報を読み取り(計測し)、どのような動きをする(制御している)のかまとめましょう。また、それぞれ計測する部分(センサー)と仕事をする部分(アクチュエータなど)もあげましょう。

身の回りの製品・装置

例) 自動ドア
1)
2)
3)

計測しているもの		どのような制御をしているか
例) 人がいるかどうか		例) ドアの開閉
1)	→	1)
2)		2)
3)		3)

上記した製品や装置で使用されているセンサーやアクチュエータを調べ、下にまとめましょう

計測している部分(センサー)		仕事をする部分(アクチュエータなど)
例) 赤外線センサー		例) DCモーター
1)	→	1)
2)		2)
3)		3)

センサー例

計測する部分(センサー)	計測するもの
光センサー	周囲の明るさ
赤外線センサー	対象物の有無
音センサー	周囲の音の大きさ
カラーセンサー	対象物の色
ジャイロセンサー	対象物の傾き

アクチュエータなどの例(※)

仕事をする部分(アクチュエータなど)	動きや働き
LED	光を発する
電子ブザー	音を発する
サーボモーター	角度を変える
DCモーター	回転させる

※アクチュエータとは、エネルギーを機械的な動きに変換するものを指す。

アイデア設計シート

身の回りの人の「困ったこと」を考え、どのようなシステムを開発をすればその問題を解決できるのかを考えましょう。また、そのシステムのイメージを簡単な図でまとめましょう。

対象者
(困っている人)

対象者の生活で困難が予想される場面

困難を解決するために開発するシステム

システムを開発することで改善されること

開発するシステムのイメージ図



年 組 番 名前(

月 日)

発表シート

システム名

--

システムのイメージ

--

用途・機能

--

システムが動作する条件と動作内容

条件	動作

プログラムの作成に関する工夫

--

プログラム以外の制作に関する工夫

--

巻末資料 制御ブロックの使い方

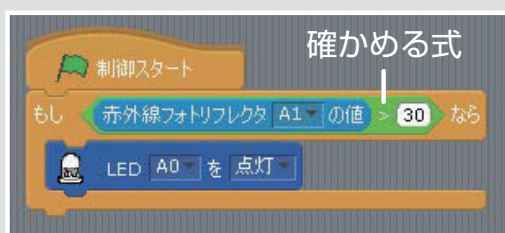
①



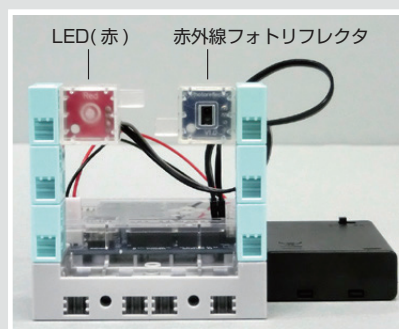
の中の式を確かめ、式が成り立つ場合だけ、ブロックで囲まれた処理を実行する。

例) 手をかざしたときだけ LED が点灯するプログラム

作成したプログラム



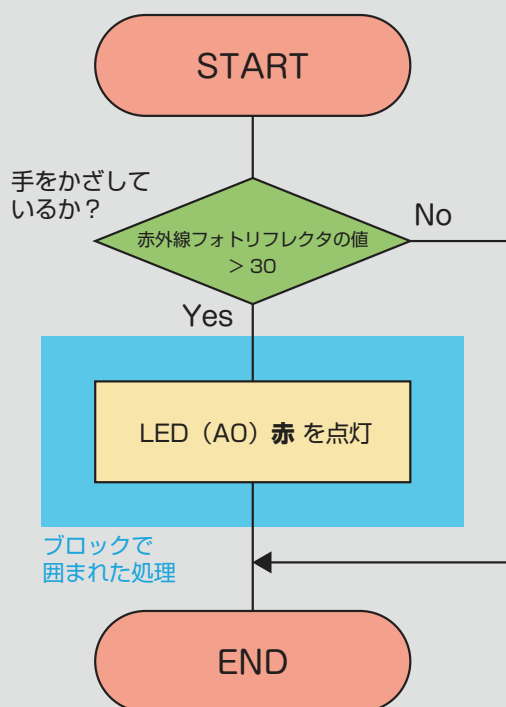
製作例



入出力設定

- ・ A0 : LED (赤)
- ・ A1 : 赤外線フォトリフレクタ

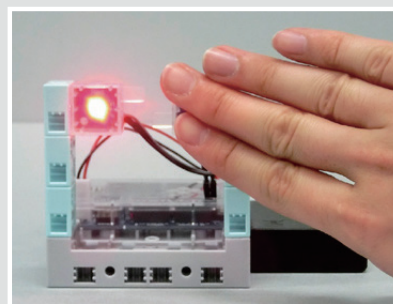
プログラムの処理の流れ



プログラム実行

赤外線フォトリフレクタに手をかざしたあと、電源を入ると、LED(赤)が点灯する。

※電源を入れたあと、赤外線フォトリフレクタに手をかざしても既にプログラムが終了しているため、反応しない。



応用

○ 2 つの条件を指定したいとき



「かつ」、「または」のブロックを使う。

「かつ」は2つの式の両方が成り立つ場合、

「または」はどちらか一方でも式が成り立つ場合にブロックで囲まれた処理を実行する。

○ ブロックを入れ子にする

「もし～なら」のブロックは入れ子にできる。

下の2つのプログラムは同じ処理を表す。



=



②



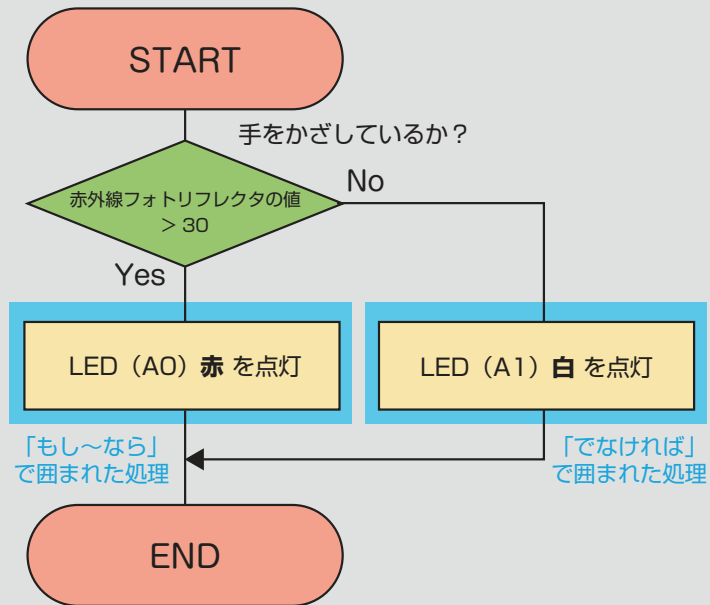
の中の式を確かめ、式が成り立つ場合とそうでない場合で、実行する処理を分ける。

例) 手をかざしたときは LED (赤) が点灯し、そうでない場合は LED (白) が点灯するプログラム

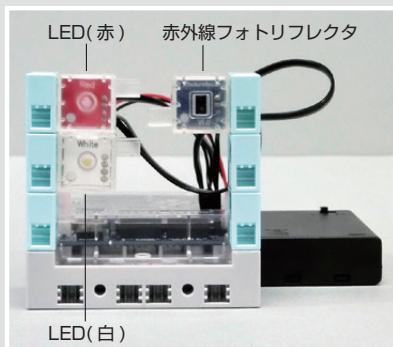
作成したプログラム



プログラムの処理の流れ



製作例



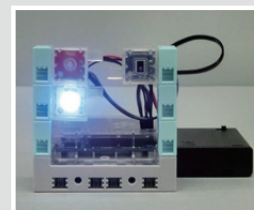
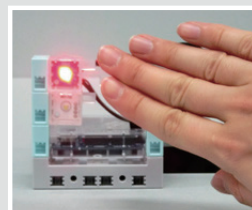
入出力設定

- ・ A0 : LED (赤)
- ・ A1 : LED (白)
- ・ A2 : 赤外線フォトリフレクタ

プログラム実行

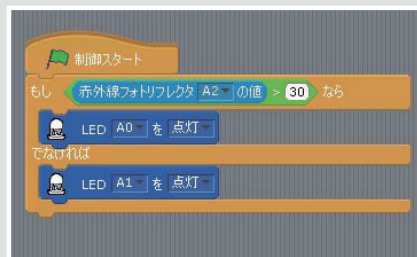
赤外線フォトリフレクタに手をかざし、電源を入れると、LED (赤) が点灯し、そうでない場合 LED (白) が点灯する。

※電源を入れたあと、赤外線フォトリフレクタに手をかざしても既にプログラムが終了しているため、LED (赤) は点灯しない。



応用

「もし～なら」を2つ並べた左図のプログラムを考えたときは、「もし～なら、でなければ」を使った右図のプログラムに変えると、わかりやすくまとめることができる。



③



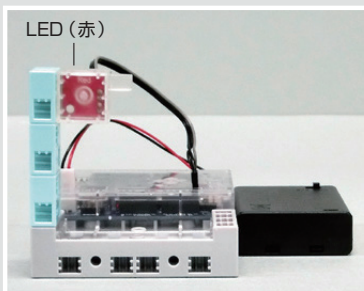
○ の中に指定した回数だけ、ブロックで囲まれた処理を繰り返し実行する。

例) LED (赤) を3回繰り返し点滅させるプログラム

作成したプログラム

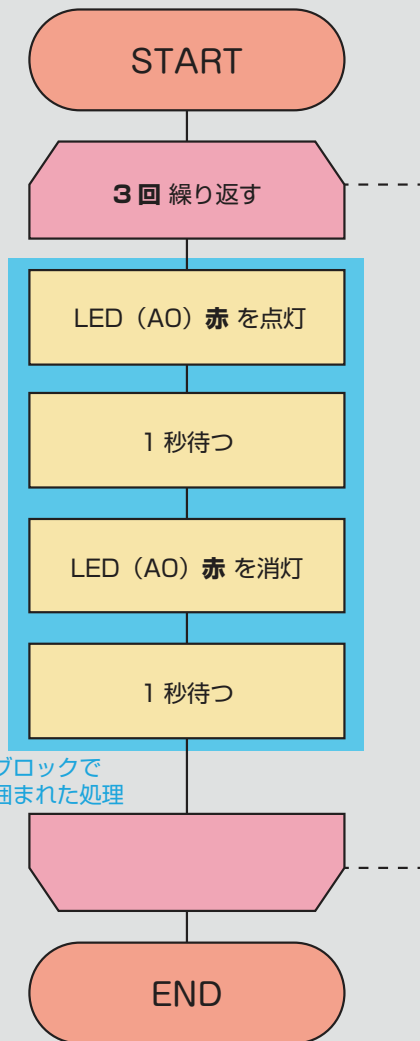


製作例



入出力設定
・ AO : LED (赤)

プログラムの処理の流れ

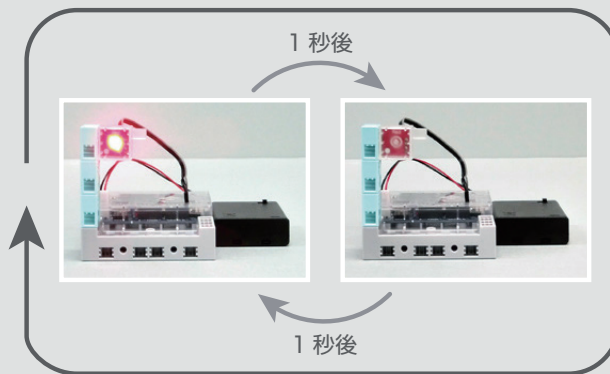


ブロックで
囲まれた処理

プログラム実行

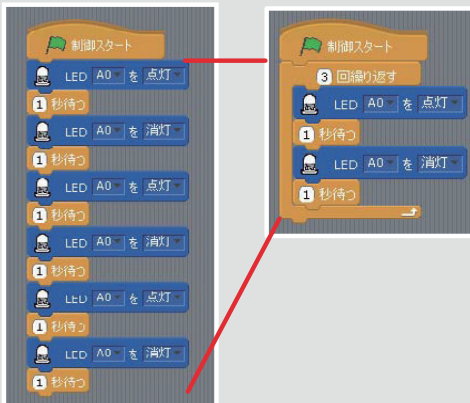
電源を入れると、LED (赤) が 1 秒間置きに点灯・消灯を 3 回繰り返す。

3 回繰り返す



応用

「～回繰り返す」のブロックを使うと、同じ処理を何度も実行する場合にプログラムを短くまとめることができる。



4



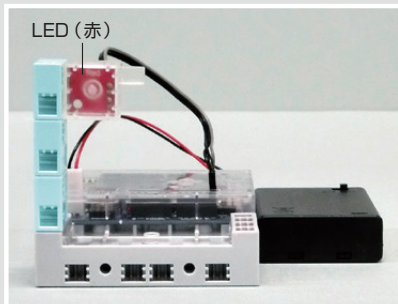
ブロックで囲まれた処理を電源が落ちるまでずっと繰り返す。

例) LED（赤）を無限に繰り返し点滅させるプログラム

作成したプログラム



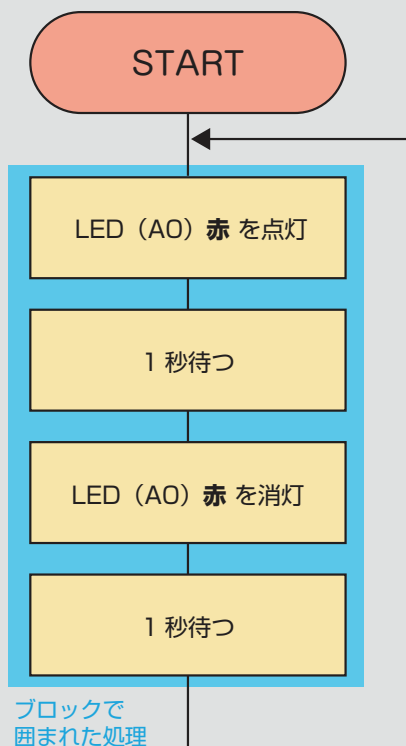
製作例



入出力設定
・ AO : LED（赤）

プログラムの処理の流れ

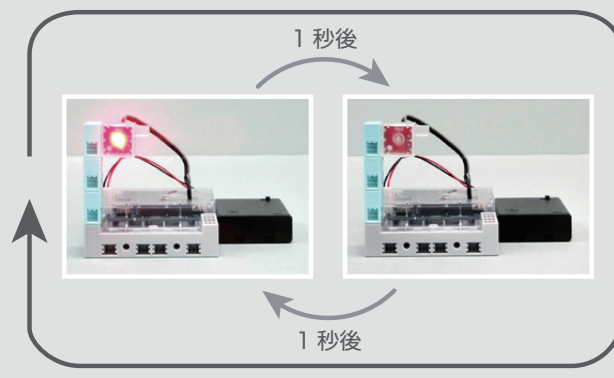
ブロックで囲まれた処理を無限に繰り返す。



プログラム実行

電源を入れたら、LED（赤）が1秒間置きに点灯・消灯を電源が落ちるまで無限に繰り返す。

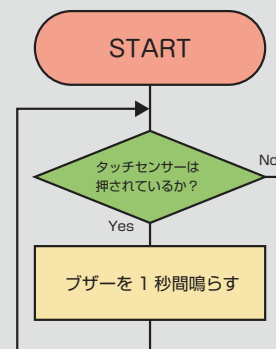
無限に繰り返す



応用

「ずっと」のブロックを「もし～なら」のブロックと組み合わせることで、常にセンサーの値を監視し、処理を実行するプログラムが作成できる。

例) タッチセンサーを押すと、1秒間ブザー音が鳴るプログラム



5

まで待つ

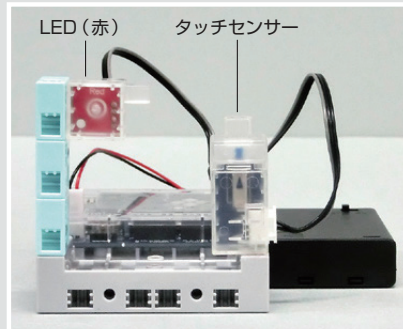
の中の式を確かめ、式が成り立つまで次の処理を遅らせる。

例) タッチセンサーが押されると 1 回だけ LED (赤) が点滅するプログラム

作成したプログラム



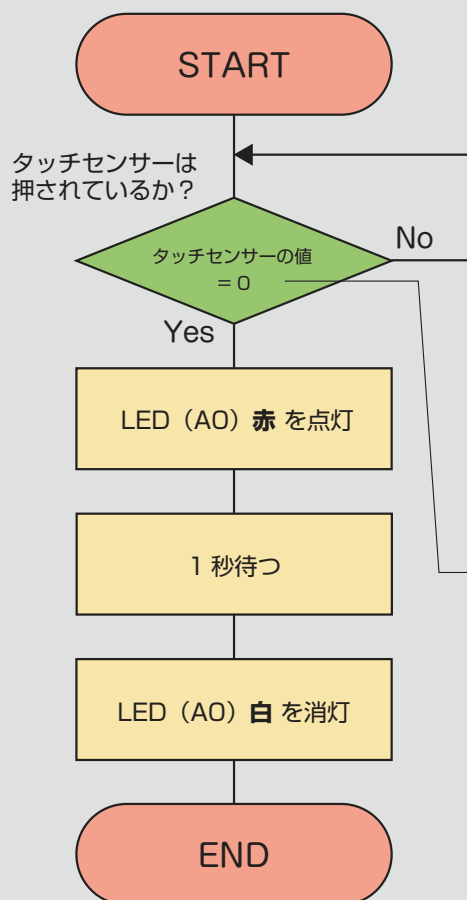
製作例



入出力設定

- ・ A0 : LED (赤)
- ・ A1 : タッチセンサー

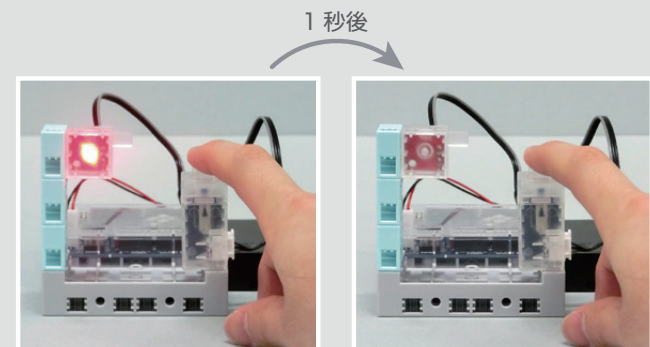
プログラムの処理の流れ



プログラム実行

電源を入れ、タッチセンサーを押すと、LED (赤) が点灯し、1 秒後に消灯する。

※もう一度タッチセンサーを押しても、既にプログラムが終了しているため、LED (赤) は点滅しない。



※タッチセンサーは押されているときに 0 の値を示す。

6



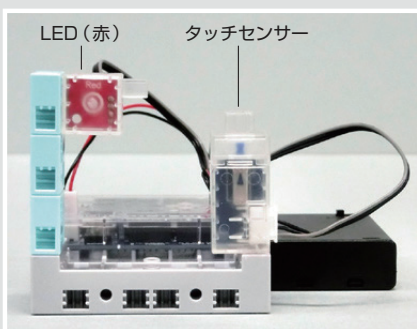
の中の式を確かめ、式が成り立つまでブロックで囲まれた処理を繰り返し実行する。

例) タッチセンサーが押されるまで、LED（赤）を繰り返し点滅させるプログラム

作成したプログラム



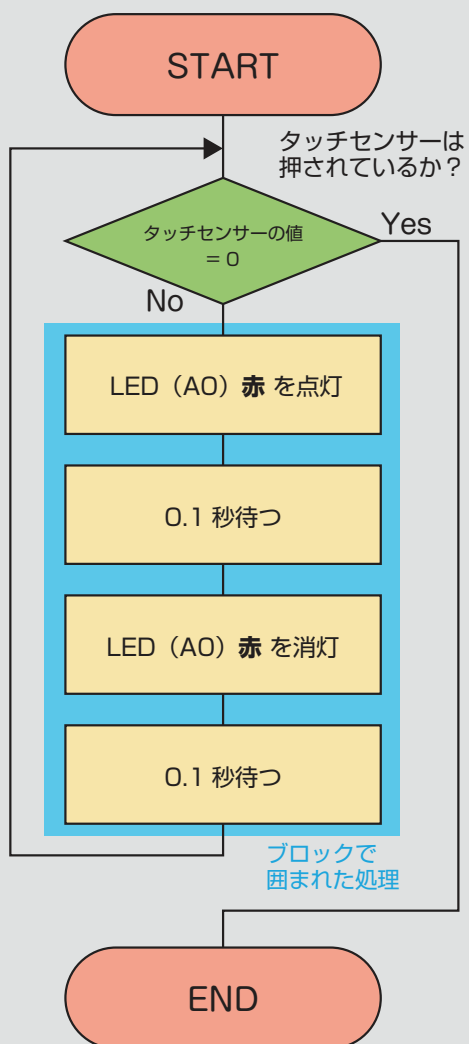
製作例



入出力設定

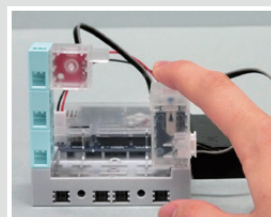
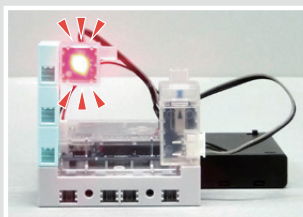
- ・ A0 : LED (赤)
- ・ A1 : タッチセンサー

プログラムの処理の流れ




プログラム実行

電源を入れたら、LED（赤）が点滅を繰り返し、タッチセンサーを押すと、消灯する。



7

ずっと、もし  なら

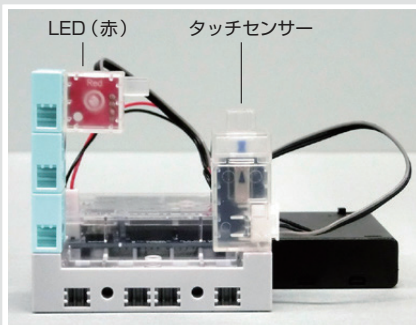
電源が落ちるまで、繰り返し  の中の式を確かめ、式が成り立つ場合にブロックで囲まれた処理を実行する。

例) タッチセンサーが押されている間だけ、LED（赤）を繰り返し点滅させるプログラム

作成したプログラム



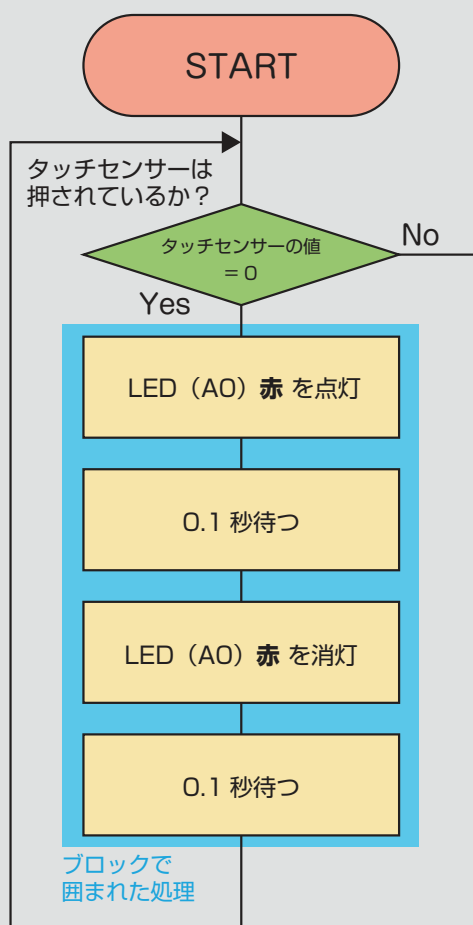
製作例



入出力設定

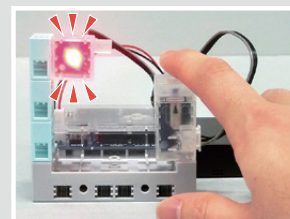
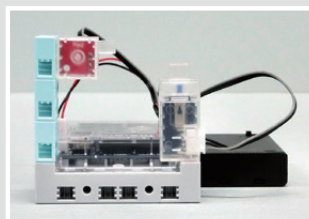
- ・ A0 : LED (赤)
- ・ A1 : タッチセンサー

プログラムの処理の流れ



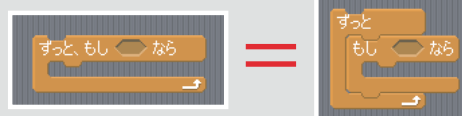
プログラム実行

電源を入れたあと、タッチセンサーを押すと、押している間だけ LED（赤）が点滅を繰り返す。一度タッチセンサーを離れたあと、もう一度押しても同じように LED（赤）が点滅を繰り返す。

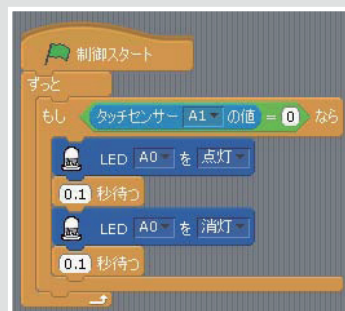


応用

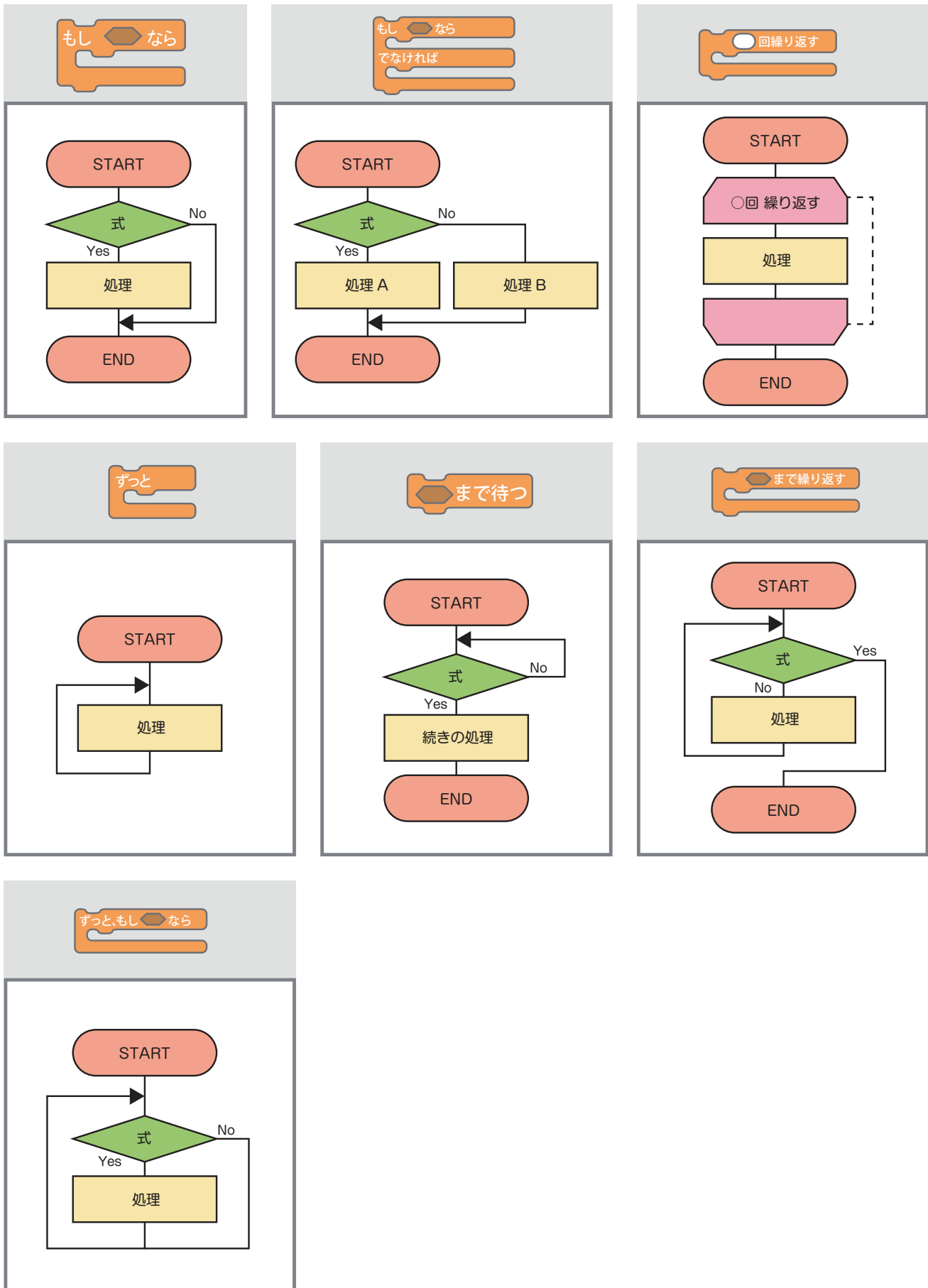
「ずっと、もし～なら」のブロックは「ずっと」のブロックと「もし～なら」のブロックを組み合わせたと同じ処理を行う。



そのため、代わりに下図のプログラムを転送しても同じ結果になる。

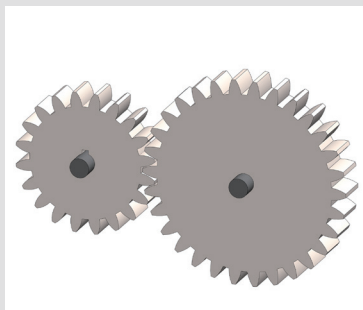


フローチャート対応表

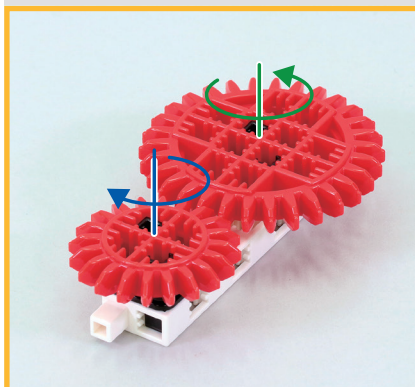


1 ギヤ

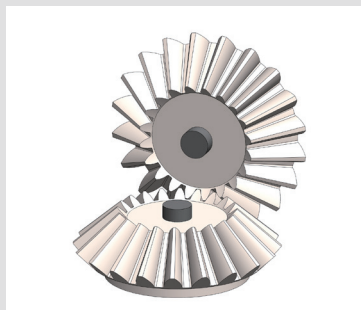
スパーギヤ



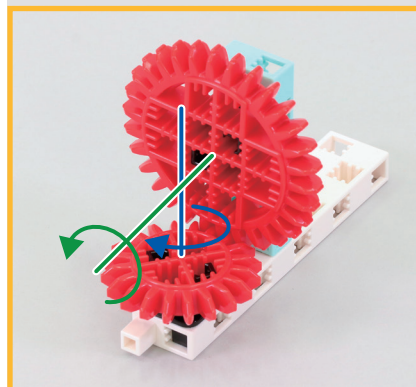
円のまわりに平らにならんだ歯をもつ最も一般的なギヤです。さきほど組み立てたそう置でもこれをつかっていました。



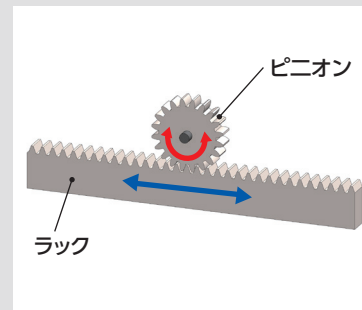
マイタギヤ／ベベルギヤ



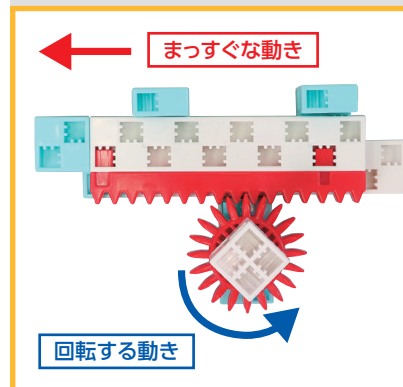
回転じくの向きをかえて動きを伝えるときにつかいます。歯の数が同じものの組み合わせを**マイタギヤ**、歯の数がちがうものの組み合わせを**ベベルギヤ**とよびます。



ラック&ピニオン

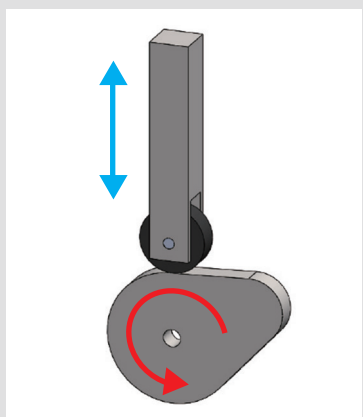


ラックはまっすぐに歯がならんだギヤです。**ピニオン**とよばれる小さなギヤと組み合わせ、回転する動きをまっすぐな動きにかえることができます。

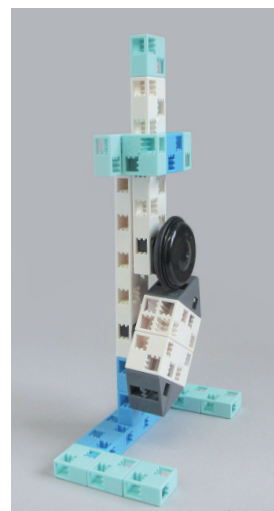
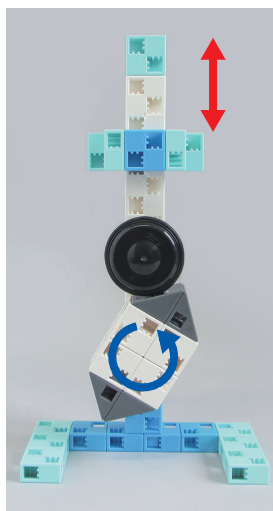
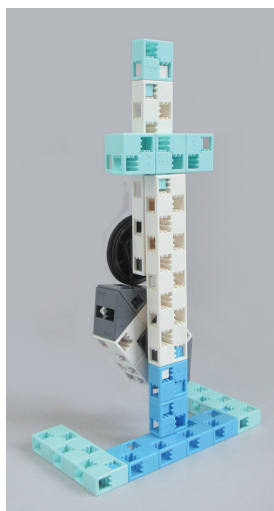


2 カム

カム機構



円でない板の回転する動きを利用して、動きの向きをかえます。



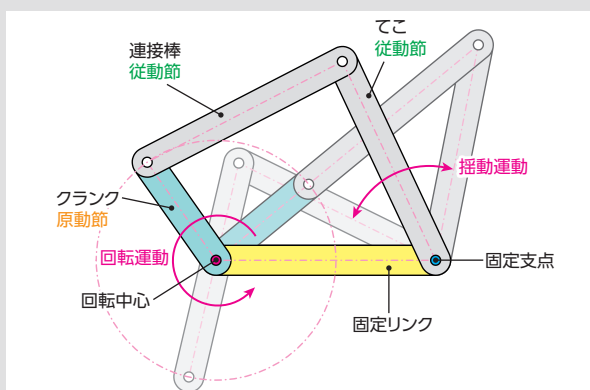
回転する動き



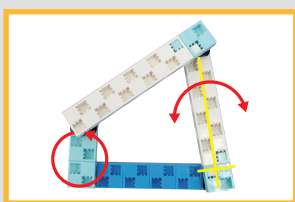
まっすぐな動き

3 リンク

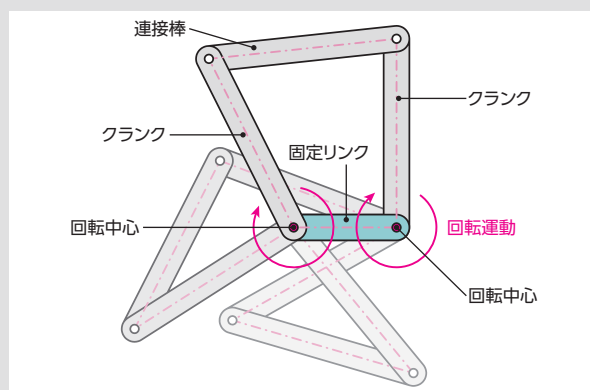
てこ・クランク機構



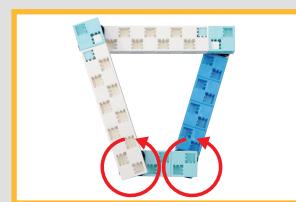
原動節を回転させると、
従動節が揺動運動を行
います。



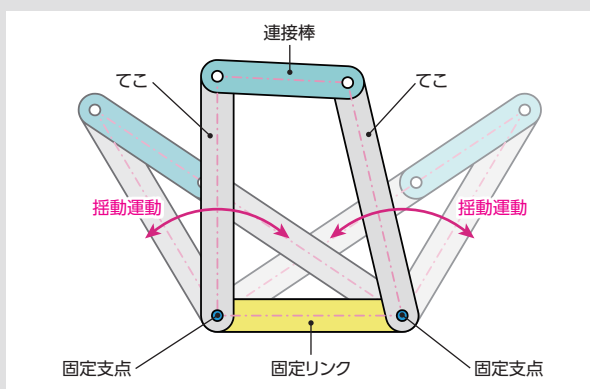
両クランク機構



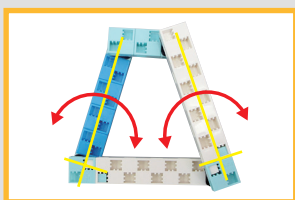
2本のクランクが接続棒
で接続されて、回転運動
を行います。



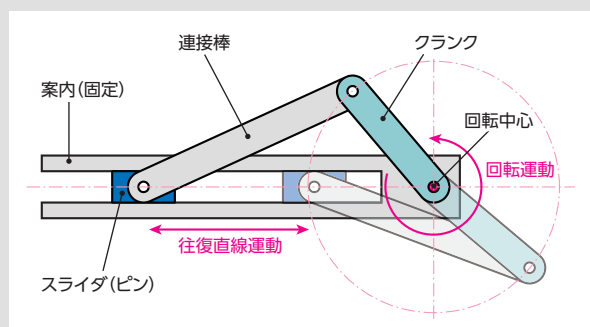
両てこ機構



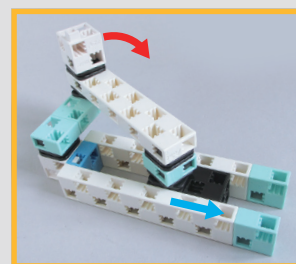
2本のてこが接続棒で接
続されていて、揺動運動
を行います。



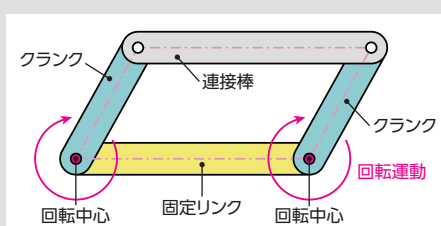
スライダクランク機構



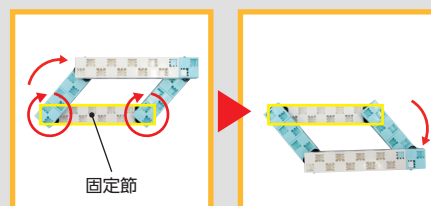
ピンを案内(溝)に沿って
動かすことで、回転運動
を直線運動に変えたり、
逆に直線運動を回転運
動に変えたりすること
ができます。



平行クランク機構

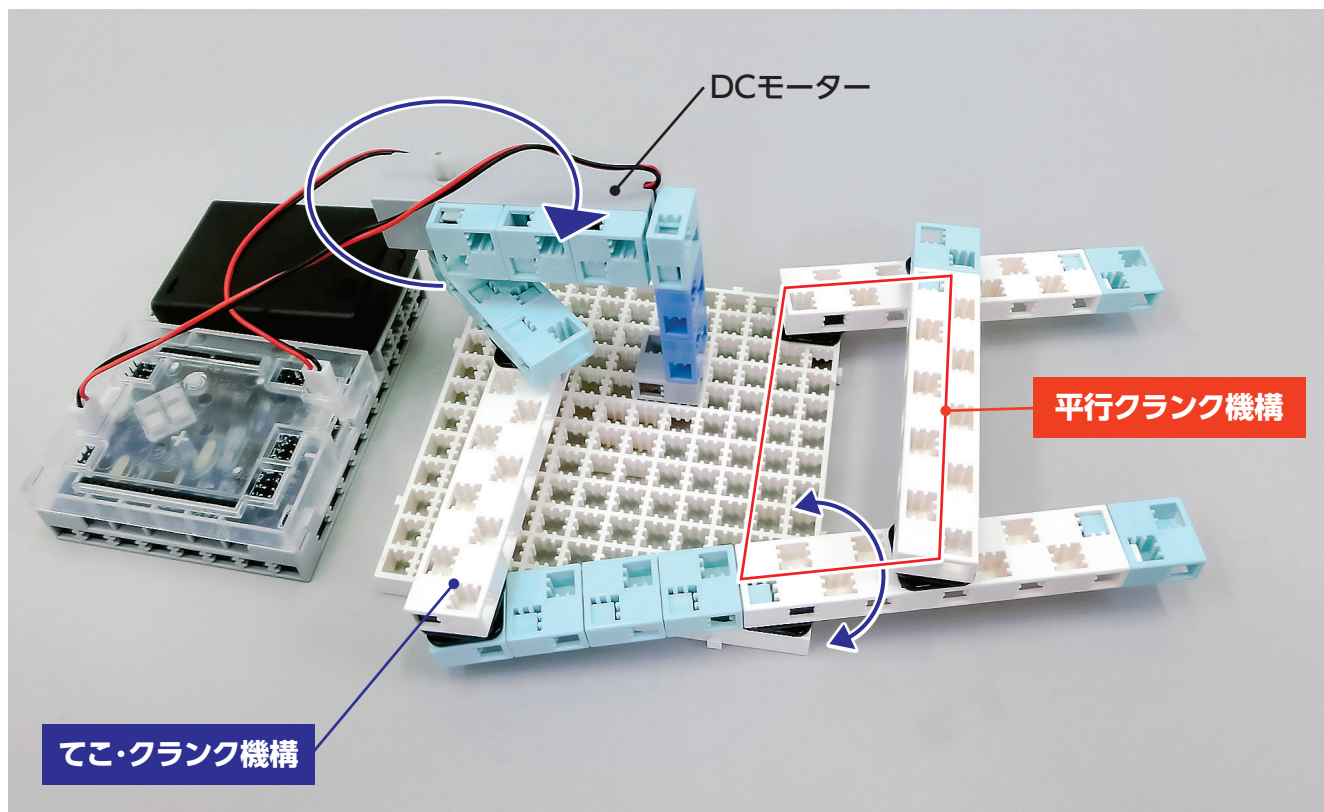


両クランク機構の中でも、向かい
合う筋の長さをそろえたものを
平行クランク機構といい、動く
ときの形がかならず平行四辺形
になるという特徴があります。

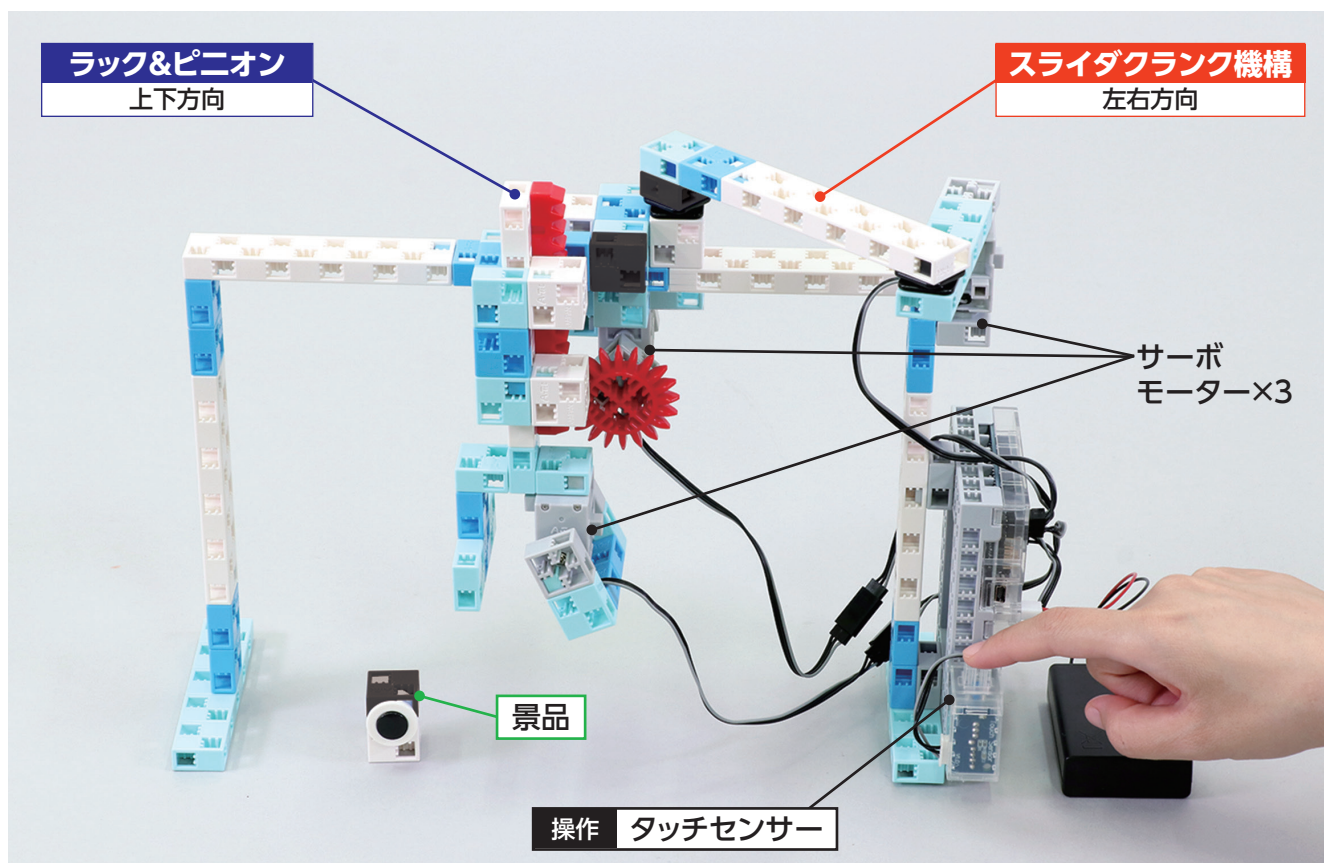


機構をつかった作例

ワイパー



クレーンゲーム





計測と制御クリエイトキット スタディガイド

発行 株式会社アーテック

【本社】 大阪府八尾市北亀井町 3-2-21
TEL : 072-990-5509

【支社】 東京都千代田区丸の内 1-8-1
丸の内トラストタワー N 館 17F
TEL : 03-5208-1882

B084692

K1217