# 高校用テキスト



目 次	
1時間目	使用するパーツ・・・・・・・・・・・・・・・・・・2
	1 身の回りの計測・制御3
	2 計測・制御システムとキットの対応・・・・・・・・・・・4
	3 ソフトウェアの使用
	4 Studuino基板と各部の接続方法·····8
	5 DCモータを動かしてみよう・・・・・・・・・・・・・・・・・9
	6 条件分岐、繰り返しのプログラムを作ろう・・・・・・11
	7 自動車を組み立てよう・・・・・・・・・・・・・・・・・・・・・・・16
	課題 タッチセンサーで自動車の速さを制御する18
2時間目	使用するパーツ・・・・・・19
	1 自動車における計測・制御システム・・・・・・・・・20
	<ol> <li>直進する自動車の制御・・・・・・・・・・・・・・・・・・・・・・・20</li> </ol>
	<ol> <li>曲がる自動車の制御・・・・・・・・・・・・・・・・・・・・・・・21</li> </ol>
	4 ゆっくり曲がる自動車の制御·····23
	5 赤外線フォトリフレクタを用いた計測・・・・・・・24
	6 衝突回避自動車のプログラムの作成 ······27
	課題 障害物を避けて走らせる30
3時間目	<b>使用するパーツ</b> ·····31
	1 産業用ロボットにおける計測・制御システム・・・・・・32
	2 アームロボットを組み立てよう ····································
	3 アームロボットのプログラムの作成・・・・・・・・34
	ロボットコンテスト・・・・・・・・・・・・・・・37



## 1時間目



ブロック 白×1

A

ブロック 青×1





P

A示で<sup>®</sup> 特許登録済・意匠登録済 JP-PAT.No.5595492 JP-DES.No.1473448 ※国際特許申請中

注意:DCモータやサーボモータを無理矢理回さないようにしましょう

## 🚹 身の回りの計測・制御

私たちの身の回りにある家電製品は、計測・制御の仕組みを使って自動的にさまざまな仕事をしています。決まった 動作をくり返したり、外部の情報を元に柔軟に対応したりすることができるのは、コンピュータやセンサが組み込まれ ているからです。炊飯器の例を見てみましょう。



炊飯器では、連続的に変化する温度を温度センサを通して電気的なアナログデータに変え、それをコンピュータで扱えるようにディジタルデータに変換しています。また、温度センサからの情報の変化をコンピュータで監視し、ヒータで温度の調節を行い、最適な温度を保っています。

このように身の回りの家電製品では、コンピュータやセンサ、仕事をする部分(アクチュエータなど)が組み込まれ、計測と制御を行っています。



## 2 計測・制御システムとキットの対応



※アクチュエータとはエネルギーを動きに変える物を指すため、光るだけのLEDはアクチュエータには含まれません。

# 3 ソフトウェアの使用

コンピュータが様々な処理を実行するための仕組みや、手順、プログラムのことをソフトウェアといいます。今回はプログラムを作成するために、ブロックプログラミング環境というソフトウェアを使用します。

## 3-1:ブロックプログラミング環境の起動

#### Windows の場合

- 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種類のプログラミング環境が選択できます。授業では、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。



🔤 Stuchaireプログラミング制作	ean a	II
Studuir	です 10 プログラミング環境	a Arte
		<b>クロックブログラミング地域</b> アクラミング集めなりた。 クロッキング集めなりた。 マクラミングをのクリングに、 マクラングをのかった。 マクシングをのかった。 マクラングをのかった。 マクラングをのかった。 マクラングをのかった。 マクラングをのかった。 マクラングをのかった。 マクラングをのかった。 マクラングをのかった。 マクラングをのかった。 マクラングをのかった。 マクラングをのかった。 マクラングでのかった。 マクラングをのかった。 マクラングをのかった。 マクラングをのかった。 マクラングをのかった。 マクラングをのかった。 マクラングをのかった。 マクラングをのかった。 マクラングをのかった。 マクラングをのかった。 マクラングをのかった。 マクラングをのかった。 マクラングでのかった。 マクラングでのかった。 マクラングでのかった。 マクラングでのかった。 マクラングでのかった。 マクラングでのかった。 マクラングでのかった。 マクラングでのかった。 マクラングでのかった。 マクラングでのかった。 マクラングでのかった。 マクラングでのかった。 マクラングでのかった。 マクラングでのかった。 マクラングでのかった。 マクラング マククング マククング マククン マククング マククン マク マククン マククン マククン マククン マククン マククン マククン マククン マク マクン マク マク マク マク マク マク マク マク マククン マククン マク マク マク マク マク マク マク マク マク マク
УЭнэтратали Нас		

## 3-2:ソフトウェアの説明

「ブロックプログラミング 環境」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、ロボット を動かすプログラムをつ くることができます。





## 3-4:ブロックの並べ方

ブロックを選んで真ん中の画面上にドラッグ &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 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。 そのとき、下につながっているブロックも一 緒に動きます。



## <u>3-5:ブロックの消しかた</u>

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&ドロップするとブロックを 消すことができます。



## 3-6: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。

		プロジェクトを保存	<del>7</del>
1	Compu	ter	💽 £ 🚹
C			
	С E: С G: Барнот С I:		
	:0		
	新しいファイノ	1/2	
			OK HHUTEL



## Studuino基板と各部の接続方法



# 5 DCモータを動かしてみよう

プログラムを作成し、DCモータを動かします。DCモータの速さや回転方向を変え、モータの様子を確認しましょう。その際どのような制御が行われているか考えてみましょう。

## 5-1:準備をしよう

1 電池ボックスをStuduino基板の POWER に、DCモータを M1に接続します。



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

2 USBケーブルでStuduino基板とパソコンを接続します。



## 5-2:プログラミングしてDCモータを動かそう



2 図のようにブロックを ドラッグ して並べましょう。





▲ 図のようにブロックを クリック するとそれぞれDCモータが 回転 と 停止 の動作をします。
▶ 1000 にする DCモーター M1 の速さを 1000 にする DCモーター M1 を 正転
■ DCモータが 回転 します。
■ DCモータが 停止 します。

 5 右図のブロックの ○部分の 数値 を変えると回転の速さが 変わります(最大100)。
 ○部分を逆転にかえると、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モータの動きを確認できたら、 USBケーブルを抜く前に、メニュー バー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。



DCモーター M1 の速さを 50 にする

DCモーター M1 を 逆転

DCモータの正転・逆転

DCモータの正転・逆転の制御は、モータに流れる電流の向きを基板で変えることによって行われています。



#### DCモータの速さ

DCモータの回転速度の制御には、パルス幅制御(PWM制御)が利用されています。通常DCモータでは、電圧を 変えることで、流れる電流の大きさが変化し、回転速度が変わります。しかし、この方法は電圧の変化に伴い発生 する熱を捨てる必要があり、効率の悪い制御方法といえます。また、コンピュータからの出力は基本的にHighか Lowであるため、電圧の細かい変化を行う際には専用の回路を設計する必要があります。

パルス幅制御(PWM制御)ではコンピュータから出力する電圧を一定にし、その電圧を加える時間や加えない時間を変化させることで回転速度を制御しています。



HIgnが短く、LOWが長い。少し回転して 止まろうとする。→ ゆっくり回転する。



Highが長く、Lowが短い。長く回転して、 止まる時間が短い。→速く回転する。

## 各件分岐、繰り返しのプログラムを作ろう

プログラムには、実行させたい処理を順番に実行する「順次処理(順次構造)」のほかに、条件によって処 理を変える「条件分岐(分岐構造)」や同じ処理を繰り返し実行させる「繰り返し(反復構造)」があります。

6-1:準備をしよう

タッチセンサにセンサ接続コード(短)を 接続します。





↑ Studuino基板の A4 のコネクターに タッチセンサ を接続します。



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

## 6-2:入出力設定をしよう



- 現れたウィンドウで使う部品を設定しましょう。
  - ① チェックをすべてはずす  $\checkmark \rightarrow \square$

DCモーター	サー;	ポモーター			ボタン
□ M1 □		2 🔲 D4 9 🛄 D10	D7	D8	A0 A2
センサール	ED/ブザー				
🗖 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	光センサー	
□ A3	光センサー		🗆 A7	光センサー	

② DCモータM1とA4にチェック 🗸 を ③ A4の光センサをタッチセンサに 入れる

DCE-Q-		サーボモ	-9-			ボタン	
	M2	02	D4	D7	D8	□ A0 □ A1	A2
2ンサー/L	ED/J	<b>f</b> -					
□ A0	光セン	<del>1</del> 9 -		( 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	光センサー		
🗆 A3	光セン	サー		🗆 A7	光センサー		

変更し、OKをクリックする

DCモーター	-	サーポモ	-9-			ボタン	
✓ M1	M2	D2	D4	D7	D8	🗖 A0	🗆 A2
		D9	D10	D11	D12	□ A1	🗆 A3
センサーノレ	ED/プリ	<b>f</b> –					
□ A0	光セン	<del>7</del> -		<b>₽</b> A4	タッチセンサ	- \	•
🗆 A1	光セン	<del>7</del> –		□ A5	光センサー		
□ A2	光セン	<del>7</del> –		🗖 A6	光センサー		
🗆 A3	光セン	<del>7</del> –		🗆 A7	光センサー		

## 6-3:条件分岐(分岐構造)

タッチセンサを使用し、始めにタッチセンサを押しているとDCモータが正転し、押していないと逆転する プログラムを作成します。

処理の流れは、右の図のようになります。

プログラムが実行され、始めにタッチセンサが押され ているかどうかを判断します。タッチセンサが押され ているときと押されていないときに、コンピュータ上 でどのような数値(ディジタルデータ)を扱っている か、テストモードを使用して変化を見てみましょう。



#### 6-4:タッチセンサの数値の変化をみてみよう

Studuino基板とパソコンをUSBケーブルでつなげましょう。
 メニューバーの 実行 を クリック して、テストモードを開始 を選びましょう。

2 テストモードになったら、画面右側に センサ・ボード が表示されます。その中の [A4]タッチセンサ の数 値に注目しましょう。 タッチセンサを押したときと押していないときの数値がどのようにかわるか確認 しましょう。



※テストモード中はUSBケーブルを抜かないでください。

## 6-5:タッチセンサを使って処理を分けよう

タッチセンサを使用し、タッチセンサが押されているとき、つまり値が0のときDCモータを正転、そうでなければ、つまり値が1のときDCモータを逆転させるプログラムを作成しましょう。



3 (調べる の中から タッチセンサー ▲4 ▼の値)を (○= ○)の ○部分にドラッグしましょう。

4 ()=() の()部分に0を入力します。

タッチセンサA4=0と入力したとき、このプログラムは、「もし、タッチセンサA4 =0ならば、上の段のプログラムが実行され、そうでなければ、下の段のプログラム が実行される」という意味になります。



- 5 「もし~なら」の上の段に「動き」から「DCモータ M1の速さを100にする」と「DCモータM1を 正転」ブロックをドラッグします。
- Studuino 🕀 🗟 ファイル 編集 実行 ヘルプ 動き || 制御スタート 調べる ) [演算 ] - A4 - の値 = 0 M1 の速さを 💋 DCモーター M1 の速さを 100 ( DCモーター M1 を 正転 モーター M1 \* を 停止 ブザー 💽 から (60マ)を出力 🥢 ブザー 💌 の出力を停止する 🔜 LED 🔽 を 点灯 🕶
- 6 「もし~なら」の下の段に「動き」から「DCモータ M1の速さを100にする」と「DCモータM1を 正転」ブロックをドラッグし、逆転に変えます。

Studuino 🕀 🗄	ファイル 編集 実行 ヘルプ
<u>動き</u>   割御   叫べる   変数	
<b>₩</b> サーボモーター ▼ を 90 <b>\$</b>	もし - 多ッチセンサー A4 一の値 = ① なら
DCモーター M1 の速さを 11	OCモーター M1 の速さを 100 にする     Oの速さを 100 にする     OOD にする     ODD     ODD      ODD     ODD      ODD
DCモーター M1 を止転。 DCモーター M1 を停止	
①     【     「         ・         ・         ・	DCモーター M1 1 00度とを 100 にする トラック
ブザー 💽 の出力を停止する	
LED Tを 点灯T	

StuduinoとPCのUSB接続を切り離さないでください

Studuino 🕀 🗄 ファイル 編集 実行 🕻

「逆転」を入力

プログラム作成・転送

テストセード開始

ハーツ

7 Studuino基板とパソコンがUSBケーブルでつ ながっているかを確認し、メニューバーの実行 をクリックし プログラム作成・転送を選び、プ ログラムを転送します。

🥂 プログラム転送中は

USB ケーブルを

抜かないでください

制御 調べる ) [演算 ] タッチセンサー A4 の値 = 0 クリン 変数 🊀 DCモーター M1 の速さを 100 にする 🙀 サーボモーター 💌 を 🛛 90 🜩 DCモーター M1 を 正転 💌 🚀 DСモーター 🎦 の速さを 10 DCモーター M1 の速さを 100 にする 🚀 DCモーター M1▼ を 正転▼ DCモーター M1 を 逆転 🔹 🔏 DCモーター M1 🛛 を 停止

動き

8 転送終了後、タッチセンサを押した状態でリ セットボタンを押しプログラムを始めから実行 させてみたりしましょう。



リセットボタン

#### 条件分岐の種類

条件分岐のプログラムでは、分岐の方法によって使うブロックが変わります。 下に2種類の条件分岐を使ったプログラムのフローチャート例があります。 それぞれの分岐方法の違いをみつけましょう。





#### 6-6:繰り返し(反復構造)

6-5のプログラムでは、プログラムの処理が始まった時点で タッチセンサを押していなければDCモータは正転しませ んでした。プログラムは基板に転送後すぐに実行され、タッ チセンサを押していなければDCモータが逆転した状態の ままプログラムが終了します。いつでもタッチセンサを押 すとDCモータが正転し、離すとDCモータが逆転するよう な動作を実行するためには、6-5で作成した処理を常に繰り 返す必要があります。このとき使用するのが繰り返し(反復 構造)です。6-5で作成した処理をずっと繰り返すときの手 順を表したフローチャートは右の図のようになります。





メニューバーの「実行」から「プログラム作成・転送」を選択し、プログラムを実行しましょう。



### 7-1:DCモータ、サーボモータを組み合わせ自動車をつくろう

DCモータで前進後退、サーボモータで左右の方向を切り替える自動車を組み立てます。



▲ Studuino基板の向きに注意



8 3と 2を合体し、サーボモータをD9に、DCモータをM1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す。

10 9 にタイヤを取り付けて完成です。



課題 タッチセンサーで自動車の速さを制御する

タッチセンサーのオン/オフで自動車の走る速さが切り替わるプログラムを作成しましょう。

#### 準備

1 組み立てた自動車にタッチセンサーをつなぎます。 2 下の図のように入出力設定を行います。

にゅうしゅつりょくせってい

DCモーター

👿 M1 🗖 M2

センサー/LED/ブサー

□ A1 ひかりセンサー

□ A2 ひかりセンサー

■ A3 ひかりセンサー

チェックをすべてはずす

サーボモーター

D2 D4

x

ボタン

🗆 A0 🔲 A2

🗌 A1 🔲 A3

-

キャンセル

🗆 D7 👘 D8

☑ A4 タッチセンサー

□ A6 ひかりセンサ~

□ A7 ひかりセンサー

OK

🗖 A5

ひかりセンサ・

☑ D9 □ D10 □ D11 □ D12

Ŧ



## <u>プログラムの作成</u>

赤枠内のブロックを使用してプログラムを完成させましょう。



メニューバーの「実行」から「プログラム作成・転送」を選択し、プログラムを実行しましょう。

# 2時間目



## | 自動車における計測・制御システム

私たちの身の回りでは、計測・制御システムを利用したものの例として、家電製品のほかに自動車が挙 げられます。従来の自動車ではドライバーが全ての情報を判断し、最適な動作を行っていました。 近年では、自動車における計測・制御システムの新しい技術として衝突事故軽減システムというものが 活躍し、ドライバーをサポートしています。衝突事故軽減システムとは、自動車につけられたセンサや カメラが前を走る車や障害物の情報を受け取り、コンピュータが解析し、ドライバーへの警告やブレー キの補助操作などを行うシステムです。

実際のシステムでは、カメラによる画像認識や電波を使用したレーダー、赤外線レーザーをつかった レーダーなどで障害物を感知していますが、この授業ではキットの中に入っている赤外線フォトリフ レクタを使用します。



前回の授業で製作した自動車のモデルを使用し、衝突回避場面におけるセンサを用いた計測・制御シス テムを作成していきましょう。

# 2 直進する自動車の制御

衝突回避を行う計測・制御システムを作成するにあたり、初めに自動車の走行プログラムを作成します。 直進する自動車のプログラムを作成しましょう。

自動車をまっすぐ前に走らせるためには、 サーボモータを90°、DCモータを正転に 設定しなければなりません。





▲ 電池ボックスのスイッチを ON にすると、車が直進します。直進しない場合は、サーボモータの角度 を 調整して、もう一度プログラムを転送しましょう。

## 3 曲がる自動車の制御

1秒間直進し、その後左右に曲がるプログラムを作成しましょう。



自動車を右に曲がらせるためには、サーボモータの角度を90°より大きい値に設定します。



自動車を左に曲がらせるためには、サーボモータの角度を90°より小さい値に設定します。









3ニューバーの 実行 をクリックし、プログラム 作成・転送 を選びましょう。



- 4 電池ボックスのスイッチを ON にすると、自動車が1秒間直進した後、サーボモータの角度が変化し、右に 曲がるように走ります。
- 5 サーボモータの角度を 30° にしてプログラムを転送しましょう。



6 電池ボックスのスイッチを ON にすると、自動車が1秒間直進した後、サーボモータの角度が変化し、左に 曲がるように走ります。

## 4 ゆっくり曲がる自動車の制御

前項の2つのプログラムでは、サーボモータの角度が急に変わってしまうため、自動車であれば安全に 曲がれているとは言えません。サーボモータの動く速さを変えて、安全にゆっくりと曲がるようにして みましょう。

右の図のようにブロックを ドラッグ し、
 4 サーボーター DPT を 30 € 魚にする を囲みましょう。



2 プログラムを転送しましょう。

3 右の図のように サーボモータを同時に動かす (速さ)の数値を変えて、動く速さが変わること を確認しましょう。 数値は0~20の間で設定することができます。 数値が大きくなるほど速く動きます。



## 5 赤外線フォトリフレクタを用いた計測

衝突回避を行う計測・制御システムを作成するにあたり、自動車の前に障害物があるかを計測するため、 赤外線フォトリフレクタというセンサを使用します。

赤外線フォトリフレクタをよく見てみましょう。小さなレンズのようなものが2つ並んでいることがわかります。透明な方が赤外線を出す赤外線LEDで、黒い方が赤外線を受ける赤外線検出器です。





赤外線LEDから出た赤外線を障害物に反射させて、 はね返ってきた赤外線を検出器で読み取ります。

赤外線を用いたセンサは、衝突事故軽減システム以外にもさまざまな場所で身近な製品に使用されています。どのような製品に、どのような目的で使われているか考えてみましょう。

赤外線センサを使用した製品	どのような目的で使われているか

#### 5-1:準備をしよう

赤外線フォトリフレクタに、センサ接続コード(短)を接続します。



3 製作した車の前方に2を 写真のように取り付けます。 2 赤外線フォトリフレクタにブロックを写真のように取り付けます。



4 Studuino基板の A4 のコネクターに赤外線フォトリフレクタ を接続します。





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

#### 5-2:赤外線フォトリフレクタの数値の変化をみてみよう

1 編集から入出力設定を選択し、A4 に赤外線フォトリフレクタを設定します。



DCモーター	サーボ	モーター			ボタン
₩1 ¥ M	2 🗆 D2	🗆 D4	🗆 D7	🗆 D8	🔽 A0 🔽 A2
	<b>₽</b> D9	☑ D10	☑ D11	D12	🗹 A1 🔽 A3
センサー/LE	D/ブザー				
🗆 A0 🚦	光センサー	*	🖌 A4	赤外線フォ	トリフレクタ 🔹
🗆 A1 🚦	光センサー	Ŧ	🗆 A5	光センサー タッチセンサ	-
🗆 A2 🚦	光センサー	-	🗖 A6	赤外線7ォ	トリフレクタ 🔪
- A0 -	光ヤンサー	*	□ A7	ブザー	

A4にチェックを入れ 赤外線フォトリフレクタを選び、 OKをクリック!





3	テストモードになると、画面右側の センサ・ボード が表示されます。	センサー・ボード	
	その中の [A4] 赤外線フォトリフレクタ の数値に注目しましょう。	[A0] ボタン	1
	※テストモード中は USB ケーブル を抜かないでください。	[A1] ボタン	1
		[A2] ボタン	1
		[A3] ボタン	
	数値に注目	[A4] 赤外線フォトリフレクタ	18
		[A5] 未接続	
		[A6] 未接続	
		[A7] 未接続	

4 赤外線フォトリフレクタに白色のブロックを近づけた時の数値がどのように変わるか計測しましょう。

#### ●計測した値を記入しましょう。

きょり 距離	10cm	8cm	6cm	4cm	2cm	1cm	0cm	
あたい 値								※計測する数値は環境により 多少変動します



#### AD変換

赤外線フォトリフレクタでは、赤外線LEDから出た赤外線が障害物に反射し、どれだけはね返ってきたかを検出 器で読みとることで障害物の有無を計測することができました。このとき検出器で読みとった赤外線量は常に連 続的に変化するため、「アナログデータ」と呼ばれます。コンピュータでは、取り扱う情報をHigh(1)やLow(0)な どの数値の系列で表した「ディジタルデータ」として処理することに長けているため、「アナログデータ」をコン ピュータの「ディジタルデータ」にそのまま対応させると無限個の数値の系列を扱うことになります。しかし、コ ンピュータで扱える数値の系列には限りがあるため、「アナログデータ」を四捨五入や切り捨て、切り上げなどの 方法によって有限個のとびとびの数値に変換する必要があります。このように「アナログデータ」を「ディジタル データ」に変換することを「AD変換」といいます。



前項のテストモードで計測された数値もAD変換後の「ディジタルデータ」になります。

## 6 衝突回避自動車のプログラムの作成

センサで障害物を検知し、衝突を回避する自動車のプログラムを 作成しましょう。プログラムを作成するためには、目的の達成や 問題解決のための手順を明確化する必要があります。これらの手 順を、順次や繰り返し、条件分岐などの単純な手順に分割して表現 し、その手順が有限回の処理で終了するものを「アルゴリズム」と いいます。「アルゴリズム」は「フローチャート」などの図式を用い て表現することができます。



2 フローチャート

1 衝突回避アルゴリズムの例 前に進む 赤外線フォトリフレクタを用いた 障害物の有無の計測を行う もし障害物があれば停止する

## 6-1:衝突回避自動車のプログラムの作成

フローチャートをもとに、ソフトウェアでプログラムを作成します。

1 5-2で計測した値をもとに、障害物までどれくらいの距離(値)で自動車を止めるのかを決めましょう。



ボタン A0 \* の値

速度センサー ・ の値



8 Studuino基板とパソコンがUSBケーブルでつながっているかを確認し、プログラムを転送しましょう。

USBケーブルを外し、プログラムの動作を確認します。自動車の進行方向上に障害物を置き、1 で設定した距離まで近づくと停止するか確かめましょう。 他にも、障害物ギリギリで止まるプログラムや余裕をもって止まるプログラムなど、様々な条件で試して



注意
白いブロック以外の障害物で試す場合は、赤外線を反射しにくい黒色でないものを使用しましょう。



赤外線フォトリフレクタで障害物を検知すると、避けて走るプログラムを作成しましょう。



### プログラムの作成

赤枠内にブロックを追加してプログラムを完成させましょう。



メニューバーの「実行」から「プログラム作成・転送」を選択し、プログラムを実行しましょう。

# 3時間目

4

# 使用するパーツ 2時間目で製作したロボットカー ×1 USBケーブル×1 サーボモータ×2 ブロック 白×4 ブロック 水×2 ブロック 薄水×4 ブロック グレー×1 ステー×2 回転軸×1 **入って** (<sup>®</sup> [特許登録済・意匠登録済] JP-PAT.No.5595492 JP-DES.No.1473448 ※国際特許申請中

## 産業用ロボットにおける計測・制御システム

近年、工場では人間のかわりに作業を行う産業用ロボットが 活躍しています。人手では負担の大きい作業や単純な繰り返 しを行う作業などでは、早く正確に繰り返し動作を行うこと ができる産業用ロボットが非常に重宝され、効率的に製品を 製造しています。例えばアームロボットは、人間の腕のような 構造をしており、人間の力では運ぶことが難しい物をつかん で目的地まで運んだり、物を素早く組み立てたりする作業に 役立っています。その他にも災害現場や危険な場所でガレキ を取り除いたり、被災者を救出したりするレスキューロボッ トなど世の中の様々な場面で活躍しています。



今回の授業では、アームロボットの関節部分に使われるサーボモータの特性を理解し、物をつかんで持ち上げる動作のプログラムを作成していきましょう。

#### アームロボットで使用するサーボモータの役割



①上下の動作を行うサーボモータ

②物をつかむ動作を行うサーボモータ

# 2 アームロボットを組み立てよう

1

2

1 ブロックを組み立てます。



4 3 にブロックを取り付けます。

2



- 7 6 を本体に取り付けます。 1 1 7 7 8 サーボモータをD
- 9 л = 9 & D10 · D11 IC つ な きま 3
   1
   2
   010
   011
- 5 サーボモータにブロックと回転軸を 取り付けます。
   1 サーボモータ の向きに注意!
   2

2 1にサーボモータを取り付けます。

▲ サーボモータ の向きに注意! 1

8 サーボモータをD10・D11につなぎます。

 、 灰色のコードの 向きに注意!
 一 灰色

3 2 にブロックを取り付けます。



## 3 アームロボットのプログラムの作成

車体に取り付けたサーボモータの角度を考え、障害物をつかんで持ち上げるプログラムを作成しましょう。

#### 3-1:アームの動きを考えよう(つかんで持ち上げる)

アームの動きは簡単に表すと下の写真のようになります。

下の写真をヒントにプログラムを考えてみましょう。

このとき、アームのサーボモータの角度をどのように設定すればよいのかテストモードを使い、都度試してみましょう。

※テストモードの使い方は次のページに記載しています。











サーボモータ D10	度
サーボモータ D11	度

#### 3-2:テストモードを使ってサーボモータの動きを確認する

 メニューバーの実行からテスト モード開始を選択します。



2 図のようにブロックを並べます。

3 サーボモータの接続ポートを それぞれD9からD10、D11 に変更します。

動き	制御	C
調べる	演算	A 制御スタート
変数	1	DIO
<u></u>		🎧 サーボモーター 🗖 💿 を 🛛 90 🗢 度にする
● サーボモーター D10 を 90		● サーボモーター □11 を 90 🗢 度にする
🥜 DCE-A	- M1 の速さを 10	DII

4 ▲▼をクリックして、角度を変 更するとリアルタイムにサーボ モータが動きます。

※事前に電池ボックスの電源を ONにしておきます。



54を繰り返し、それぞれの動作を行うときのサーボモータの角度を決めましょう。

## 3-3:つかんで持ち上げるプログラムを作ろう

3-1で考えたサーボモータの角度をプログラムに組み込み、障害物をつかんで持ち上げるプログラム を作成します。

 図のようにブロックをドラッグして並べ、3-1の
 ①から④までの動作を行うサーボモータの角度を それぞれ入力しましょう



ログラムを転送します。
 電池ボックスのスイッチをONにすると、障害物を持ち上げます。



サーボモータを同時に動かすブロックを使用すると、 サーボモータの動かす速さを変更することができます。

#### 試してみよう!

A のように 🧐 サーボモーター 🖭 を 💶 🕈 魔にする だけを使う場合、ブロックの間に 1 秒待つ が必要です。 B のように サーボモーター 福岡崎 (潮知 10 € )) を使うと、サーボモータが指定の角度になるまで次の角度に

#### ならないので必要ありません。

1 秒待つ を間に入れた場合、サーボモータが始めの角度になり、1秒間待ってから次の角度になります。下の図を参考にしてください。



つかんだ障害物を離す動作を考えてプログラムを作成してみましょう。

## ロボットコンテスト

今まで学習した内容を用いて、与えられたミッションを達成することのできるロボット及び プログラムの作成を行いましょう。

#### 

コース上に2種類の火種があり、このままでは被害が拡大してしまうことがわかっている。 被害拡大を防ぐため、無人ロボットを使用し、火種の除去に向かうことにした。



#### 3 ルール説明

- ・スタート時は、スタートライン上にロボットの前輪のタイヤが乗っていなければならない。
- ·スタート後、ロボットは自律的に行動する。
- ・競技時間は3分以内とする。
- ·競技時間内であれば、何度でもスタート位置から再挑戦できる。
- ・競技時間内に全ての火種の除去を行い、最も早くゴールできた時間を最終の評価基準とする。
- ・火種の除去は、火種に見立てた2種類の形状(球・直方体)の材料を、台からコース外に落とす。 (火種1個除去につき到達時間からボーナスとして5秒マイナス)
- ·除去した火種がコース上に落ちた場合、ボーナスを得ることはできない。
- ・走行中はロボットに触れてはいけない。
- ・再挑戦する場合には、その旨を判定員に伝え、スタート位置から再び始める。
- ・再挑戦する場合に機体の修正を行う際は、チーム全員が機体に触れてもよい。
- ・ゴールライン上に前輪のタイヤが乗った時点でゴールとなる。
- ・ゴールまでの到達時間と火種除去のボーナスの合計時間で順位が決まります。

#### 4 競技用ロボット

- ・ロボットの製作台数は各チーム1台とする。
- ・ロボットの製作には「アーテックロボ」を使用し、それ以外のパーツは使用できない。
- ・別のチームとの間でパーツの貸し借りをすることはできない。

## MEMO

B087527 K0217