

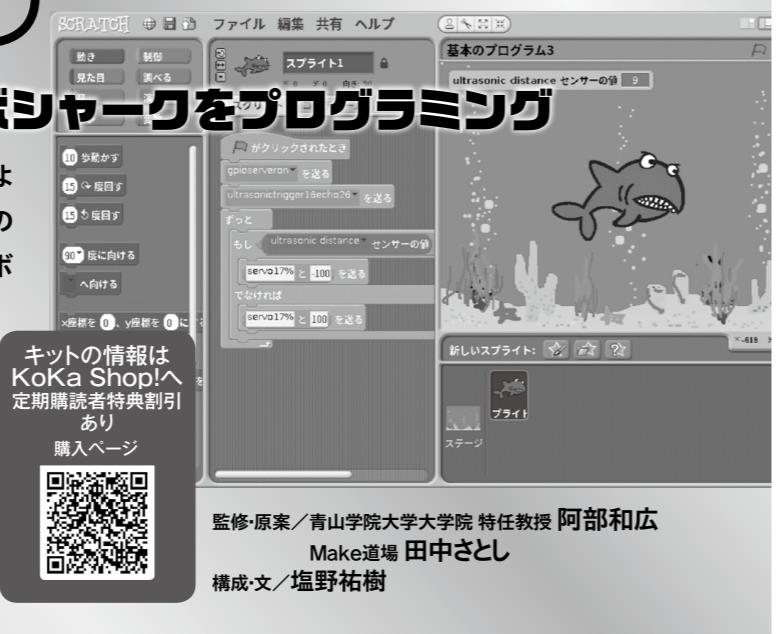
ゲーム 工作 プログラミング

ラズベリーパイ Raspberry Pi 大活用

第30回 スクラッチでロボシャークをプログラミング

3回にわたるロボシャークづくりもいよいよクライマックスだ! 7月号ではロボット制御の基本と、電子回路の配線、8月号ではロボシャーク本体の組み立てをしたね。今回はこれらをすべて合わせ、ロボシャークをプログラムで動かす!

ジブン専用パソコン特設サイト
prog.kodomonokagaku.com/jibun

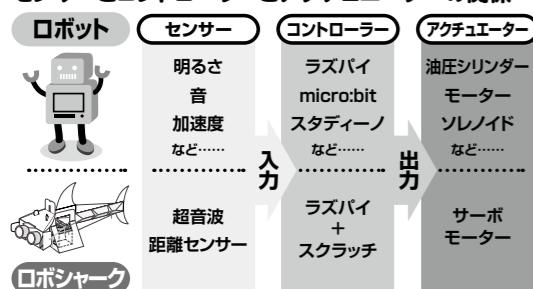


前々回の復習

少し間があいたので、前々回(7月号)の復習をしておこう。

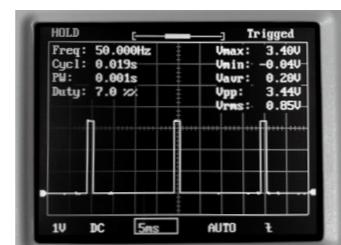
ロボットは、入力となるセンサー、出力となるアクチュエーター、それらを制御するコントローラーから構成されている。ロボシャークでは、センサーとして超音波距離センサー、アクチュエーターとしてサーボモーター、コントローラーとしてラズパイとスクラッチが使われているぞ。

センサーとコントローラーとアクチュエーターの関係



超音波距離センサーには、トリガー(Trigger:引き金)とエコー(Echo:反射波)というピンがある。トリガーに信号を送ると、それが引き金となって超音波が発射される。対象に当たって跳ね返ってきた超音波を受け取ると、かかった時間の分だけ、エコーの信号がHIGH(5V)になる。これにより、音の進む速さと時間の関係から、対象までの距離がわかるわけだ。

サーボモーターは、送られるパルス信号の幅によって、回転する角度が決まる。PWM(パルス幅変調)と呼ばれる方法だよ。



オシロスコープで見たサーボモーターの信号。3本立っている山のそれが1つのパルス。横軸は時間で1つのマスが5m秒、縦軸は電圧で1つのマスが1V。

基本のプログラム

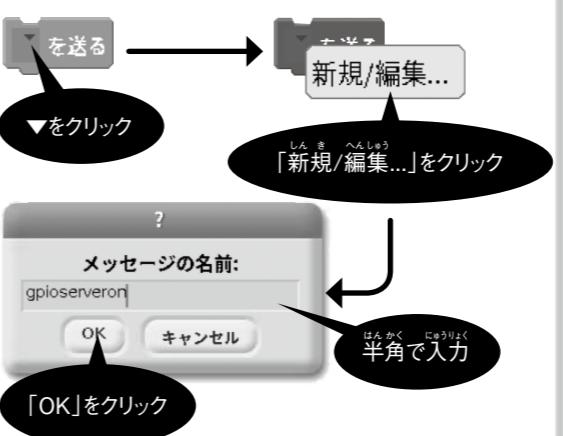
まずは、基本となるプログラムをつくろう。ロボシャークが完成し、動作テストも終わっているという前提で進めるよ。まだつくれていない人は、特設サイトのバックナンバーを読んでね。

ロボシャークは、「固定モード」と「動くモード」のどちらでもOK。ラズパイの電源が入っているときは、中の歯車が壊れてしまうので、絶対にサーボモーターのホーンを手で動かさないでね。

では、スタートメニューの「プログラム」から「Scratch」を選ぼう。「Scratch 2」ではないぞ。Scratchが立ち上がったら、次のようにプログラムを組み立てよう。これは、GPIOを使えるようにして、超音波距離センサーで使うピンをセットするプログラムだ。



「緑の旗がクリックされたとき」と「[▼]を送る」は、「制御」カテゴリーにあるよ。「[▼]を送る」にはまだメッセージ名が入っていないので、「gpiserveron」と半角で入力しよう。



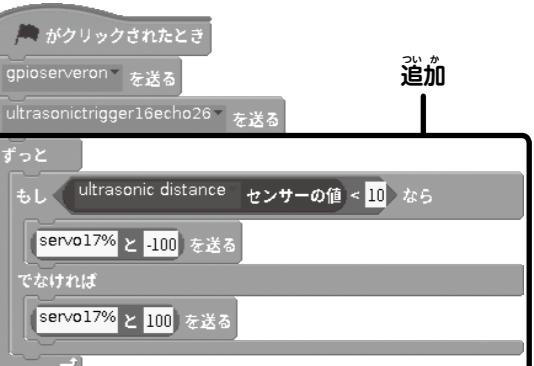
ちょっと長いけど、1文字でも間違えると動かないで、慎重に入力しよう。同じように

「ultrasonictrigger16echo26」も入力しよう。「ultrasonic」は「超音波」という意味だ。

もし、「OK」した後に間違いに気づいたら、もう一度「▼」をクリックして「新規/編集...」から打ち直せるよ。

ここまでできたら、緑の旗をクリックしよう。画面上は何も起こらないけど、それで大丈夫だ。

それでは、超音波距離センサーの値が10よりも小さいときは尾びれを左、大きいときは尾びれを右



に振るようにプログラムしてみよう。

「ずっと」と「もし～なら～でなければ～」は「制御」カテゴリーにある。

「[ultrasonic distance▼] センサーの値」は、「調べる」カテゴリーの「[スライダー▼] センサーの値」の「▼」をクリックして、メニューから「ultrasonic distance」を選ぼう。「distance」は「距離」という意味だ。

メニューの中に「ultrasonic distance」が見つからない場合は、さきに入力した「ultrasonic trigger16echo26」を打ち間違えているか、緑の旗をクリックしていないことが原因だ。「full datetime」などもなかったら、「gpiserveron」を打ち間違えているか、緑の旗をクリックしていないことが原因。もう一度メッセージ名を確認して、緑の旗をクリックしよう。

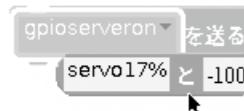
スライダー
明るさ
音
抵抗-A
抵抗-B
抵抗-C
抵抗-D
傾き
距離
fulldatetime
hours
minutes
ultrasonic distance

「[] < []」ブロックは「演算」カテゴリーにある。左の穴には「[ultrasonic distance▼] センサーの値」をはめよう。右の穴には半角の数字で「10」と入力しよう。これが「超音波距離センサーの値が10より小さい」を表しているよ。



「[] と []」のブロックは、「演算」カテゴリーにある「[ハロー]」と「[ワールド]」を使う。左の「ハロー」を半角の「servo17%」に、右の「ワールド」を「-100」に書き換えよう。

出来上がった「[servo17%]と[-100]」は、「制御」カテゴリーにある「[gpioserveron▼] を送る」の、「gpioserveron」の部分にはめよう。



これは、17番ピン(#17)につないだサーボモーターを、上から見て右いっぱいに回す命令だ。サーボモーターとロボシャークは糸でつながれていながら、サーボモーターを右に回すと尾びれは左に振れる。「-100」は角度ではなく、中立の位置から右回りで約45°回るよ。ブロックの「送る」のあたりをクリックすると尾びれが動くはずだ。動かないときは、文字の打ち間違いを確認しよう。



センサーの値を確認する

ブロックパレットにある「[スライダー▼] センサーの値」を「[ultrasonic distance▼] センサーの値」にしてから、左にあるチェックボックスをクリックすると、ステージに超音波距離センサーの値が表示されるよ。

同じようにして、「[servo17%]と[100]」のブロックもつくろう。これは、中立の位置から左回りで約45°回るので、尾びれは右に振れる。同じようにブロックの「送る」のあたりをクリックして、尾びれの動きを確認しよう。

右側に入れる値を「100」以上や「-100」以下にしても、角度はそれ以上変わらない。また、「0」になると中立の位置、つまり、尾びれがまっすぐの状態になるよ。

これで、基本のプログラムは完成だ。緑の旗をクリックしてから、ロボシャークの頭に手を近づけてみよう。尾びれが左右に動くかな？

このとき、「動くモード」にすると頭も一緒に動くため、距離が微妙に変化する。手の位置を10cmぎりぎりにすると尾びれが往復運動するよ。

超音波距離センサーの入力をもとに、スクランチのプログラムで、サーボモーターの出力をコントロールできたね。特設サイトでは、ステージの背景を海にしたり、ネコのキャラクターをサメに変えたりする方法も紹介しているよ。

尻尾を激しく振る

基本のプログラムでは、10cmを境にして、尾びれを左右いっぱいに振るだけだった。これを、距離に応じて振る角度を変えながら、左右に振り続けるようにしてみよう。

プログラムはこんな感じだ。さっきとどこが違うかわかるかな？



まず、「もし～なら～でなければ～」がなくなったね。また、サーボモーターの角度を指定する数が、「(-200) / [[ultrasonic distance▼] センサーの値]」と「(200) / [[ultrasonic distance▼] センサーの値]」になっている。「() / ()」は割り算の記号だ。「演算」カテゴリーにあるよ。

例えば、超音波距離センサーの値が200のとき、それぞれの答えは、-1と1になる。同じように、値が20なら-10と10、値が2なら-100と100だね。つまり、距離が近いほど尾びれを振る角度が大きくなるということだ。

超音波距離センサーの値とサーボモーターの値の関係

超音波距離センサーの値	左に振る値	右に振る値
200	-1	1
20	-10	10
2	-100	100

サーボモーターを動かした後（メッセージを送った後）に「(0.2) 秒待つ」しているのは、サーボモーターが動く時間が必要だからだ。これがないと尾びれはうまく動かない。「(0.2) 秒待つ」を入れない場合も試してみよう。

「[ultrasonictrigger16echo26▼] を送る」の下にも「(1) 秒待つ」がある。実は、このメッセージが送られてから実際に距離が測定されるまでには、少し時間がかかる。その間、超音波距離センサーの値は「0.0」なので、角度を指定する計算が「-200/0」と「200/0」になってしまう。

Scratch 1.4では、0で割る割り算は「ゼロ除算エラー」というエラーになってしまうため、測定が始まるとここで待っているんだ。

測定中でも0になることがあるのでは？ と思った人は、実際に測定して値を確かめてみよう。また、その対策も考えてみるといいぞ。

もっと滑らかに動かす

尾びれを動かす速さを、もっと滑らかに変えることもできる。画面のネコの向きも連動して動くようにしてみたよ。プログラムだけ載せておくので、なぜそうなるのか考えてみてね。変数の「angle」と「speed」は、「変数」カテゴリーの「新しい変数を作る」ボタンでつくれるよ。



ロボシャークは他にもいろいろなことができるので、何をつくるかはキミ次第だ。夏休みの自由研究で、自分だけのロボシャークをつくれてみてほしい。例えば、誰かが近づいてきたらロボシャークが動いて警告する装置ができそうだ。また、特設サイトでは、ロボシャークを、周囲にあるモノとの距離を扇形のグラフにしてディスプレイに表示する「サメレーダー」にする応用例を紹介しているので、こちらもチェックしよう。