通信モードでのプロッチの制御

通信モードでは、パソコンとプロッチをUSBケーブルで接続した状態でプロッチの制御を行う。 この時、ドリトルは記述したプログラムの命令を一つ実行する度に、USBのケーブルを介してプロッチに専用の制御命令を送信する。 制御命令を受信したプロッチが、その内容に合わせて様々な動作を随時実施することで、プログラムに書かれた動作を実現する。
プロッチから情報を受け取ることも可能なため、プロッチに接続されたセンサなどの値をパソコン上に表示することも可能である。

1. 最初の一歩（プログラムの作り方・実行の仕方の確認）

ここではプロッチのLEDを交互に点滅するプログラムを使って、ドリトルのプログラムの入力から実行までの手順を説明する。

手順1:プログラムを「編集画面」に入力
次のプログラム(protchrmled.dtl)を入力する。

```
システム!"protchrm"使う。
最初に実行＝「
  左ライト!消灯。
」
繰り返し実行=「
  左ライト!点灯。
  プロッチ!1　待つ。
  左ライト!消灯。
  プロッチ!1　待つ。
」
プロッチ！転送。
```

手順2:通信用のプログラムをプロッチに転送する
* プログラムを実行すると通信プログラムを転送しますかのダイアログが表示するので、「はい」を押す
* 転送が完了しましたが出たら「はい」を押す
* 3秒立つとプログラムの動作が開始する

手順3:動作を終了する

- 実行画面の「終了」ボタンを押す(ドリトルからの制御命令の送信を終了する)

今回のプログラムでは、左のLEDが1秒ごとに点滅すれば正しく動作している。

2. プロッチの制御プログラムを書く時のルール

ドリトルからプロッチを制御するプログラムを説明する。

ルール1
　プロッチを**使う**プログラムを作成することを示すため、先頭の行に記述する。プロッチ本体を表す「プロッチ」[1]というオブジェクトや各種センサやモータなどを表すオブジェクトが使えるようになる。

```
システム!"protchrm"使う。
```

ルール2
　一度だけ実行したい処理を**最初に実行**のブロックに書く。モータの速度を設定することや、動作の開始を合図するためのブザー音などの処理を書くことが考えられる。

```
最初に実行＝「
□□□□□□
□□
```

ルール3
　何度も実行したい処理を**繰り返し実行**のブロックに書く。LEDの点滅やセンサの計測値によって動作を分岐するような処理を書くことが考えられる。

```
繰り返し実行=「
□□□□
□□
```

ルール4
　最初に実行や繰り返し実行のブロックに書いたプログラムをプロッチに送るために**転送**を使用する。通信モードの場合、制御命令を一つずつプロッチに送る。

```
プロッチ!転送。
```

3. センサやアクチュエータの利用

3.1 LEDライトを点灯しよう

ここでは、プロッチの左右に搭載しているLEDライトを制御する。
各ライトは「左ライト!点灯。」や「右ライト！点灯。」を使うことでライトを点灯することができる。
同様に、「左ライト!消灯」や「右ライト！消灯」を使うことでライトを消灯することができる。

下記のプログラムを実行することで、左ライトが１秒間点灯する。

```
システム!"protchrm"使う。
最初に実行=「
    左ライト!点灯。
    プロッチ!1　待つ。
    左ライト!消灯。
    プロッチ!1　待つ。
□□
プロッチ！転送。
```

左ライト・右ライトの命令一覧

命令	機能
点灯	LEDを点灯する

命令	機能
消灯	LEDを消灯する

3.2 スイッチの利用

ここでは、プロッチの左右に搭載しているスイッチを利用して、プログラムの動きを変える。
各スイッチは、「左スイッチ!接触?」や「右スイッチ!接触?」を使うことでスイッチのONとOFFを検出できる。

下記のプログラム(protchrmswitch.dtl)では、左スイッチが押されている時に左ライトが点灯し、押されていないときは消灯する。 ここでは、スイッチを利用した条件分岐のプログラムを、「繰り返し実行」のブロックの中に書いた。
こうすることで、スイッチを利用する分岐プログラムを繰り返し実行することができる。

```
システム "protchrm" 使う。

繰り返し実行＝「
    「(左スイッチ！接触?)==はい」！ なら「
        左ライト！ 点灯。
    」そうでなければ「
        左ライト！ 消灯。
    」実行。
」
プロッチ！転送。
```

左スイッチ・右スイッチの命令一覧

命令	機能
接触?	スイッチの入力を「はい」か「いいえ」で取得する

3.3 光センサの利用

ここでは、プロッチの左右に搭載しているLEDライトを制御する。
プロッチに搭載している光センサは明るさ?を使うことで周辺の光量を取得できる。 左側の光センサを制御するオブジェクトを「左光センサ」、右側の光センサを制御するオブジェクトを「右光センサ」とした。
センサの計測値はドリトルのラベルオブジェクトを使うとドリトルの実行画面で確認することできる。

下記のプログラムは、光センサの計測値をPCの画面上に表示する。
このプログラム(protchsensorcheck.dtl)では、まず、プログラムを実行した最初（「最初に実行」の中）で、 画面上に計測値を表示するためのラベル（「計測結果」）を作っている。 そして、「繰り返し実行」の中に、光センサの明るさ?で取得した計測値を「計測結果」のラベルに書くプログラムを記述することで、 リアルタイムの光センサの計測値を表示できるようにしている。

```
システム□ "protchrm" 使う。
最初に実行=「
  計測結果=ラベル!作る。
□□
繰り返し実行=「
    計測結果!(左光センサ!明るさ?)書く。
□□
プロッチ！転送。
```

また、下記はセンサを制御に利用したプログラムの例である。 このプログラム(protchrmlight.dtl)では、周辺の光量が100より大きい時に左のライトが点灯する。

```
システム□ "protchrm" 使う。

繰り返し実行=「
    「(左光センサ！ 明るさ?)>100」！ なら「
        左ライト！ 点灯。
    」そうでなければ「
        左ライト！ 消灯。
    」実行。
□□
プロッチ！転送。
```

下図は、左光センサを手で隠した時の例である。センサ周辺が暗くなるのでLEDが点灯している。

左光センサ・右光センサの命令一覧

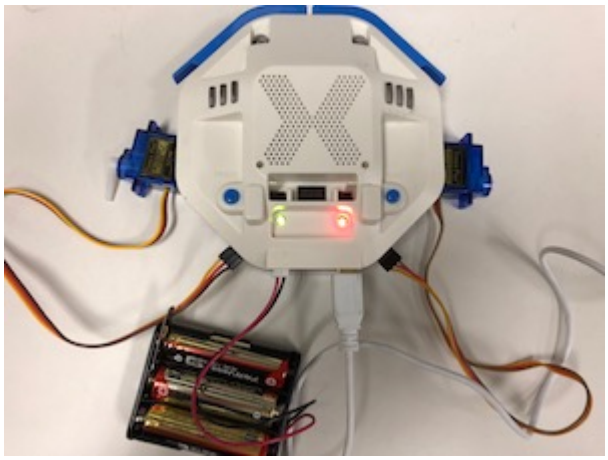
命令	機能
読む	光量を0〜255の範囲で取得する。暗い程、取得する値は大きくなる
明るさ?	光量を0-255の範囲で取得する。暗い程、取得する値は大きくなる

3.4 サーボモータの利用

ここでは、プロッチに接続したサーボモータの制御を行う。
プロッチには本体の左右にサーボモータ用の端子がある。左側のサーボモータを制御するオブジェクトが「左サーボ」、右側のサーボモータを制御するオブジェクトを「右サーボ」である。 サーボモータは角

度を使うことで指定した角度に**モータ**を動かすことができる。

なお、サーボモータの接続方法は下図のとおりである。
サーボモータを利用するときは、外部電源に電池**ボックス**を接続する必要がある点に注意する。



次の**プログラム**では、左右の**サーボモータ**の角度を**180**度から**0**度に変更する**プログラム**である。 **サーボモータ**の動作には時間がかかる。 このため、動作の完了までの待ち時間を作るために「**プロッチ!1** 待つ。」の命令で角度を**180**度に設定した後、１秒間の待ち時間を設けている。 繰り返すの命令で繰り返し回数の指定が可能になる。今回は、**3**回に指定している。

```
システム!"protchrm"使う。
最初に実行=「
□□□
        左サーボ！90　角度。
        プロッチ!1　待つ。
        左サーボ！180　角度。
        プロッチ!1　待つ。
    」！３　繰り返す。
□□
プロッチ!転送。
```

右サーボ・左サーボの命令一覧

命令	機能
角度	0度~180度の間で角度を指定する

3.5 モータの利用

ここでは、**プロッチ**のモータ（車輪）の制御を行う。
プロッチのモータは、「**プロッチ**」オブジェクトに対して**前進□後退□左折□右折□左回り□右回り**の命令を使うことで、 各動作を行うとができる。 また、**停止**を使うことでモータを止めることができる。
なお、これらの命令は秒数を指定することでその時間だけ動作することができる。
次の**プログラム**は、**プロッチ**に前進、後退、左折、右折、左回り、右回りを**1**秒ごとに切り替えながら実行するための**プログラム**である。

```
システム□ "protchrm" 使う。
最初に実行＝「
    プロッチ！1　前進。
    プロッチ！1　後退。
    プロッチ！1　左折。
    プロッチ！1　右折。
    プロッチ！1　左回り。
```

```
    プロッチ！1　右回り。
    プロッチ!停止。
□□

プロッチ！転送。
```

スイッチを使ってモータの速度を変更する

モータの回転速度は速度を使うことで変更できる。モータの回転速度は0～255の範囲で設定でき、標準では150に設定している。
次のプログラムでは、左スイッチを押した時に回転速度を255にして1秒前進し、右スイッチを押した時に速度を100にして、1秒前進する。

```
システム□ "protchrm" 使う。
繰り返し実行=「
    「（左スイッチ!接触?）＝＝はい」！なら「
        プロッチ!255 255 速度。
        プロッチ!1　前進。
    」そうでなければ「（右スイッチ!接触?）＝＝はい」なら「
        プロッチ!100 100 速度。
        プロッチ!1　前進。
    」実行。
□□

プロッチ！転送。
```

プロッチのモータの命令一覧

命令	機能
前進	左右のモータを正回転する
後退	左右のモータを逆回転する
停止	左右のモータを停止する
左折	右側のモータを正回転する
右折	左側のモータを正回転する
左回り	右側のモータが標準の速度で正回転し、左側のDCモータが標準の速度/5の速度で正回転する
右回り	左側のモータが標準の速度で正回転し、右側のDCモータが標準の速度/5の速度で正回転する
速度	DC モータの速度を0 から255 までの値で設定する。標準では150になっている

3.6 ラインセンサの利用

ここでは、プロッチの底部にあるラインセンサを用いた制御を行う。
プロッチに搭載しているラインセンサは、赤外線光を放射して跳ね返ってきた赤外線の反射量を取得することで、路面の黒と白の違いを識別する。
ラインセンサが計測した赤外線の反射量は、「ラインセンサ」オブジェクトに対し、**読む**を使うことで赤外線の反射量を取得できる。 左側のラインセンサを制御するオブジェクトが「左ラインセンサ」、右側のラインセンサを制御するオブジェクトが「右ラインセンサ」である。
次のプログラムにより、ドリトルの画面でラインセンサが計測した値を確認することができる。

```
システム!"protchrm" 使う。
最初に実行=「
  計測結果=ラベル!作る。
□□
```

```
繰り返し実行＝「
   計測結果!(左ラインセンサ!読む)書く。
   プロッチ！0.5 待つ。
□□
プロッチ！転送。
```

ライントレースカー

ライントレースカーは、ラインセンサで地面に描かれた線(黒色)を判断し、コースに沿って動くようにモータを制御する。
次のプログラムでは、左右のラインセンサを比較し、線の上にいなければ右折し、線の上にいれば左折する。

```
システム!"protchrm" 使う。

繰り返し実行＝「
    「(左ラインセンサ!読む)<(右ラインセンサ!読む)」!なら「
        プロッチ！左回り。
  」そうでなければ「
        プロッチ！右回り。
  」実行。
□□
プロッチ！転送。
```

ラインセンサの命令一覧

命令	機能
読む	赤外線の反射量を0～255の範囲で取得する

3.7 超音波センサの利用

ここでは、プロッチの上部に接続する距離センサを用いた制御を行う。
プロッチに接続する距離センサは、超音波を発生し、その反射時間を用いて、反射物までの距離を測ることができる。
「超音波センサ」オブジェクトに対して距離?の命令を使うことで、プロッチの正面にあるモノまでの距離を測ることができる。
(なお、単位はmmとなる。)
次のプログラムでは、障害物が近くになければ前進し、障害物に近づくと停止するプログラムである。

```
システム!"protchrm"使う。

繰り返し実行=「
    「(超音波センサ!距離?)<100」！なら「
        プロッチ！1 停止。
    」そうでなければ「
        プロッチ!前進。
    」実行。
□□
プロッチ！転送。
```

超音波センサの命令一覧

命令	機能
距離?	超音波センサの計測値をmmで取得する

3.8 音楽演奏

プロッチに搭載しているブザーを利用して音楽の演奏が可能である。次のプログラムでは、ブザーを使ってドレミの音階を鳴らしている。

ブザーの命令一覧

命令	機能
演奏	引数の音階を演奏する

```
システム "protchrm" 使う。
最初に実行＝「
  ブザー！"ドレミ"　演奏。
」
プロッチ！ 転送。
```

また、ブザーを用いてメロディの演奏が可能である。ここでは、譜面に相当するメロディオブジェクトに旋律を書き込み、それを演奏する。次のプログラムでは、「きらきらぼし」という名前のメロディオブジェクトを生成し、「“…“」の形で旋律を追加してから、ブザーに演奏させている。

メロディの命令一覧

命令	機能
作る	メロディの旋律を作る

```
システム!"protchrm"使う。
最初に実行＝「
    きらきらぼし＝メロディ！"ドドソソララソ～ファファミミレレド～ソソファファミミレ～ソソファファミミレ～ドドソソララソ～ファファミミレレド～"　作る。
    ブザー！（きらきらぼし） 演奏。
」
プロッチ！ 転送。
```

1)

protch「プロ」も使用できる。

From:
https://dolittle.eplang.jp/ - **プログラミング言語「ドリトル」**

Permanent link:
https://dolittle.eplang.jp/ch_protchremote?rev=1552670614

Last update: **2019/03/16 02:23**