プロッチを使ってみよう

□LEDや各種センサ、通信機能を持つロボット「**プロッチ**」を使ってプログラムで操作してみよう。

プロッチについて

プロッチは、山崎教育システム株式会社と株式会社**モノ・グラム**が開発・販売する教育教材である□LEDの点灯や、センサを利用した各種計測、モータの制御を行うことができる。**またプロッチを2台**使用することにより２つのパソコン間での文字や数字のやりとりをするプログラムを作ることもできる。
なお、ドリトルでは「オリジナルプロッチ」に対応している。[1)]

プロッチの各部品の名称

プロッチで利用できるセンサ、アクチュエータ、拡張パーツを下図や表に示す。**なお、図については、**センサを赤色、アクチュエータを青色、拡張パーツを緑色、電源やUSBなどプログラムの実行に必要な部品を黒色で示している。

部品名	説明
電源ボタン	プロッチの電源をオン・オフする
リセットボタン	プロッチに書き込まれたプログラムをもう一度実行する
外部電源	サーボモータの制御に必要な外部電源を接続する
USB	パソコンとプロッチをUSBポートで接続し、パソコンからプログラムの転送やプロッチを制御する
ライト	LEDライトを点灯、消灯する
スイッチ	スイッチのON□OFFを検出し、障害物の衝突を検出したり、スイッチをコントローラにしたりする
光センサ	センサの周辺の明るさを計測する
ブザー	音階や音長を指定して、ブザーで演奏する
ラインセンサ	赤外線の反射量を計測し、計測値からラインの白と黒を判断できる
モータ	左右のタイヤを回転させて、プロッチを動かす
サーボモータ	指定した角度にモータを制御する。※利用する場合、外部電源が必要になる
超音波センサ	超音波を出力し、壁や障害物から跳ね返ってくるまでの時間から距離を計測する
通信端子	プロッチ2台の通信端子をケーブルで接続し、それぞれをパソコンとUSBで接続して、文字や数値のやり取りをする

ドリトルとプロッチの導入手順

ドリトルでプロッチを制御するには、ドリトルのダウンロードの他、プロッチエディタ及びUSBデバイスドライバのインストールが必要になる。以下がインストールの手順である。**なお、ソフトウェアとドライバのインストールを行うため**□OSの管理者権限が必要となる場合がある。

1. **プロッチエディタをインストールする**
   - 株式会社山崎教育システムのプロッチ公式サイト(http://protch.jp/#section2)の指示に沿ってインストールを行う。
2. **プロッチのデバイスドライバのインストールする**

- プロッチの電源を入れた状態でPCとプロッチをUSBケーブルで接続する。自動的にドライバがインストールされる。
- Prolific USB-to-Serial Comm Port (COM)[2] デバイスドライバーソフトウェアが正しくインストールされました。」と表示される。\\ 写真を貼る。

3. **ドリトルをインストールする**
   - ドリトルのサイト（http://dolittle.eplang.jp）のダウンロードから最新版のWindows用のドリトルをダウンロードして任意のフォルダに展開する(C:直下が好ましい)
4. **追加パッケージをダウンロードする**
   - ダウンロードから追加パッケージ「ロボット制御ライブラリ」をダウンロードし、ファイルを解凍する。
   - robotPSフォルダ内のファイル(iniファイル,exeファイル等)を、ドリトルのルートフォルダ(dolittle.jarやdolittle.batが置いてあるフォルダ）に上書き保存する。
5. **ドリトルの起動**
   - エクスプローラからdolittle.batを実行する。[3]

ドリトルによるプロッチの制御方法の種類

ドリトルによるプロッチの制御は通信モードと自立モードがある。下表にそれぞれのメリット・デメリットを示す。

	メリット	デメリット
通信モード	ドリトルのGUIが利用できる ・「ボタン」、「ラベル」、「テキストエリア」	PCとUSBケーブルで接続できる範囲でしか動作できない ケーブルを抜くとエラーになる可能性がある
自立モード	プログラムを転送後はPCから離れた場所でも動作できる	センサやアクチュエータの動作の確認するすべがない

実際の開発にあたって、最初に通信モードでセンサの性質やアクチュエータの動作を確認しながら、プログラムを作成し、めどが立ってから自立モードでロボットを単独で動作させることが考えられる。

通信モードでのプロッチの制御方法

ここでは、通信モードでの制御方法について説明する。なお、自立モードでの制御方法に関しては、〜〜を参照してください。

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。ドリトルでは、プログラムを実行するとUSBシリアル通信を用いてプロッチに制御命令を送信している。送信された命令をプロッチで受信し、命令に合わせて動作を行う。

プログラムの入力から実行まで

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

次のプログラムを入力する。(サンプルプログラム(.dtl))

```
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。
  左ライト!1  待つ。
  左ライト!消灯。
  左ライト!1  待つ。
□□
プロッチ！転送。
```

手順2:プログラムの実行(実行ボタンを押す)
手順3:動作を確認する
手順4:動作を終了する

- 実行画面の「動作終了」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。プログラムの説明は、後にしている。

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」[4]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。
```

ライトを点灯しよう

　プロッチに搭載しているライトは、「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。また、「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。

```
システム!"protch"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。
```

上記のプログラムでは、

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
消灯	LEDを消灯する

スイッチの利用

　プロッチに搭載しているスイッチは、「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。

```
システム "protchrm" 使う。
最初に実行＝「
  左ライト!消灯。
」
繰り返し実行＝「
    「(左スイッチ！ 押された?)==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。
」
```

　上記のプログラムでは、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。プログラムは繰り返し実行の中に書いているため、何度の実行される。例えば、プロッチが箱に衝突したことを知らせるために左ライトを点灯するなどができる。

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

光センサの利用

　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。光センサは左右に搭載しているので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」にした。
　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。次のプログラムでは最初に実行で「計測結果」という名前のラベルを作り、繰り返し実行で光センサの明るさ?で取得した計測値を計測結果ラベルに書いている。

```
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。
□□
```

　次のプログラムでは、周辺の光量が100より大きい時に左のライトが点灯する。プログラムは繰り返し実行に書いているため、何度の実行される。

```
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。
□□
```

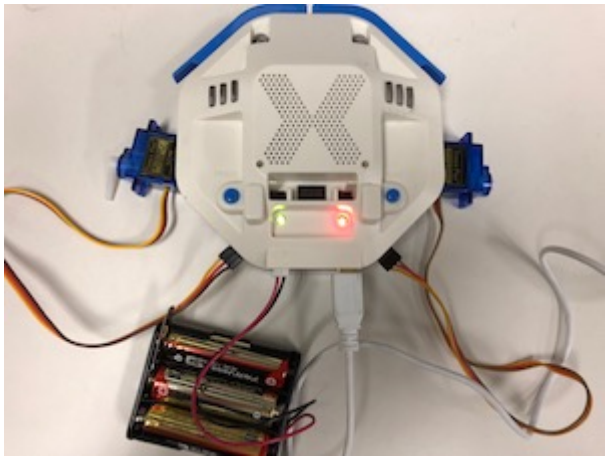
下図は、左光センサを手で隠すと暗くなるのでLEDが点灯している。

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

サーボモータの利用

プロッチにはサーボモータ用の端子がある。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。端子は左右にあり、左側のサーボモータを制御するオブジェクトを「左サーボモータ」、右側のサーボモータを制御するオブジェクトを「右サーボモータ」にした。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。



次のプログラムでは、左右のサーボモータの角度を1秒ごとに180度と0度に動作する。

```
システム!"protchrm"使う。
最初に実行=「
  左サーボ！180　角度。
  右サーボ！180　角度。
  プロッチ!1　待つ。
  左サーボ！0　角度。
  右サーボ！0　角度。
□□
```

右サーボモータ・左サーボモータの命令一覧

命令	機能
角度	0度~180度の間で角度を指定する

モータの利用

プロッチのモータは**前進**□**後進**□**左折**□**右折**□**左回り**□**右回り**を使うことで動かすことができる。また、**停止**を使うことでモータを止めることができる。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。
次のプログラムでは、1秒ごとに前進、後進、左折、右折、左回り、右回りをしている。

```
システム□ "protchrm" 使う。
最初に実行＝「
    プロッチ！1　前進。
    プロッチ！1　後進。
    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。
」
```

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**速度設定**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100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。
最初に実行＝「
    プロッチ!停止。
」

繰り返し実行=「
    「左スイッチ!押された?」！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!押された?」なら「
        プロッチ!100 100 速度。
        プロッチ!1　前進。
    」実行。
」
```

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
後進	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
速度設定	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100になっている

ラインセンサの利用

　プロッチに搭載しているラインセンサは赤外線を出し、物体から跳ね返ってきた赤外線の反射量を取得する仕組みである。読むを使うことで赤外線の反射量を取得できる。ラインセンサは左右に搭載しているので左側のラインセンサを制御するオブジェクトを「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を「右ラインセンサ」にした。
　次のプログラムでは、ドリトルの画面でラインセンサが計測した値を確認できる。

```
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。
□□
```

ラインとレースカー

　ラインセンサの反射量は、物体の色によって変化する。この特性を使うことでラインの白と黒を判断することができる。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。
最初に実行=「
　　プロッチ!停止。
□□
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。
□□
```

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
明るさ?	赤外線の反射量を0〜255の範囲で取得する

距離センサの利用

　プロッチでは距離センサをつけることができる。距離センサは距離?を使うことで、プロッチの正面にあるモノまでの距離をmmで取得できる。次のプログラムでは、障害物が近くにあると停止する。

```
システム!"protchrm"使う。
最初に実行＝「
```

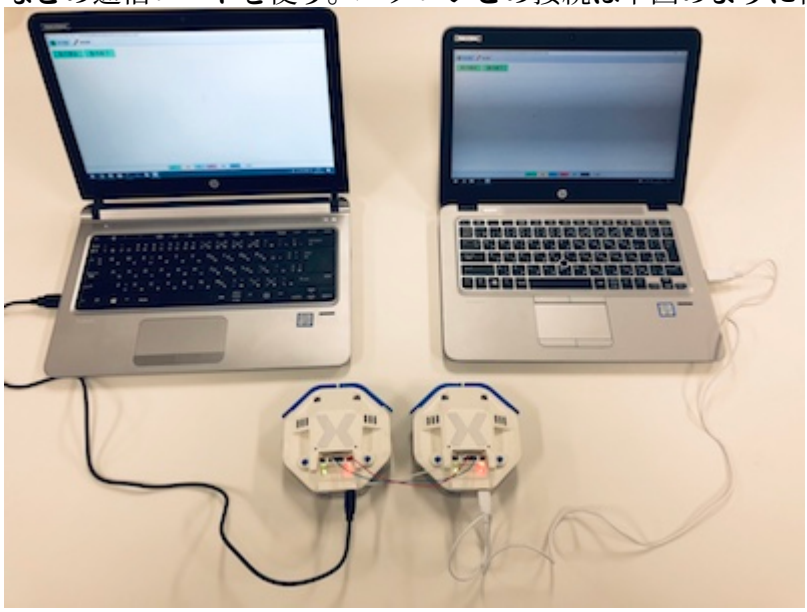
```
    プロッチ!停止。
□□
繰り返し実行=「
    「(距離センサ!距離?)<40」！なら「
        プロッチ！1 停止。
    」そうでなければ「
        プロッチ!前進。
    」実行。
□□
```

距離センサの命令一覧

命令	機能
読む	距離センサの計測値をmmで取得する
距離?	距離センサの計測値をmmで取得する

文字のやりとり

プロッチを2台使うことで文字を送ることができる。文字の送受信はシリアルオブジェクトの出力で文字を送信し、読むで送信された文字を取得できる。プロッチ2台を繋ぐためには、ジャンパワイヤーなどの通信コードを使う。パソコンとの接続は下図のように行う。



ジャンパーワイヤーの配線については、下図のようにすればよい。

メッセージを送信する

メッセージの送信には、シリアルオブジェクトの出力を使う。次のプログラムでは、「こんにちは」を送信している。

```
システム□ "protchrm" 使う。
最初に実行＝「
  通信端子!"こんにちは"送信。
□□
```

メッセージを受信する

　メッセージの受信には、シリアルオブジェクトの**読む**を使う。取得した文字をラベルに書くことでドリトルの実行画面で確認できる。

```
システム□ "protchrm" 使う。
最初に実行＝「
    受信ラベル=ラベル!作る。
□□

繰り返し実行＝「
    受信ラベル!(通信端子!受信)書く。
□□
```

シリアルオブジェクトの命令一覧

命令	機能
出力	文字を出力します
読む	送信された文字を取得します

チャットプログラム

　次のプログラムでは、フィールドオブジェクトに書いた文字をプロッチの左スイッチの押下により送信している。また、

```
システム!"protchrm"使う。
最初に実行＝「
    送信フィールド＝フィールド！ 作る。
    受信エリア＝テキストエリア！ 作る。
□□

繰り返し実行＝「
    「（左スイッチ！接触？）＝＝はい」！なら「
        通信端子！ (送信フィールド！ 読む)送信
    」実行。
    受信値＝通信端子！ 受信。
    「受信値！＝””」！なら「
        受信エリア！ (受信値)書く。
    」実行。
□□

プロッチ！転送。
```

　上記のプログラムを実行すると下図のような実行画面が現れる。①はフィールドに記述した文字を「送信」ボタンを押すことで送信する。②は送られてきた文字が書かれる。

1)
プロッチの仕様については同社のサイト(http://protch.jp/)に詳しく掲載されている。
2)
COM□の部分は環境によって異なる。
3)
「発行元を確認できませんでした。このソフトウェアを実行しますか？」のダイアログが出る場合は

「このファイルを開く前に常に警告する(W)のチェックを外してから「実行(R)を選択する。
[4)]

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_protchrm?rev=1551882150

Last update: **2019/03/06 23:22**