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。プロッチのプログラミングは、ダウンロード版のドリトルのみ対応している。

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が開発したロボット教材である□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パソコン間での文字のやりとり(チャット)などのプログラムを作ることが可能である。現在ドリトルで対応しているプロッチは「オリジナルプロッチ[1)]」という製品である。

プロッチの各部品の名称

プロッチは標準でセンサが6つ□LEDライトが2つ、モータが2つ、ブザーが1つ搭載している。また、拡張パーツには距離センサ、サーボモータ、通信コードがある。それぞれの配置を下図[2)]に示す。

各部品の説明を下表に示す。

部品名	説明
電源ボタン	プロッチを起動するボタン。プロッチを動かすときは電源を入れる必要がある
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ボタン
外部電源	サーボモータを利用する際に必要になる
USBポート	パソコンとUSBを接続するためのポート。プログラムの転送やプロッチを動かすときに接続する必要がある
ライト	LEDライトを点灯、消灯ができる
スイッチ	スイッチのON□OFFを検出できる。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できる
光センサ	センサの周辺の明るさを計測できる
ブザー	音を出力できる。音の音階に合ったブザー音がなる。
ラインセンサ	赤外線の反射量を計測できる。計測値からラインの白と黒を判断できる
モータ	左右のモータを回転させて、プロッチを動かすことができる。コースを走らすことができる
サーボモータ	[拡張パーツ]角度を指定できるモータ
距離センサ	[拡張パーツ]超音波が跳ね返ってくるまでの時間から距離を計測できる
通信コード	[拡張パーツ]パソコンとパソコンの間に通信コードで繋げたプロッチを2台を入れることでパソコン間の文字の交換ができる

ドリトルでプログラムをする手順

ドリトルによるプロッチのプログラミングは以下の手順で行う。

1. ドリトルとプロッチのプログラム環境の導入(初回のみ必要な作業)
2. 開発モードでのプログラミング(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グラムの動作を確認する)

    - プロッチに通信用のプログラムを転送する(最初に一度だけ)
    - プログラムの作成・実行を行う
    - センサの計測値をドリトルの画面で確認しながらプログラムの修正・改良を行う
3. 独立モードでのプログラミング(プログラムをプロッチに転送して□USBケーブルを外した状態で動作を確認する)
    - プログラムを作成する
    - 作成したプログラムをプロッチに転送する
    - 転送後はケーブルを外して動作を確認する

モード別の利用可能な機能

以下に、各モードの利用可能な機能を示す。基本的にドリトルのGUIを使ったプログラムに関しては、開発モードでのみ利用可能となる。

	部品	開発モード	独立モード
アクチュエータの制御	LEDの点灯・消灯制御	○	○
	モータの回転制御	○	○
	サーボモータの角度制御	○	○
	ブザーの音程制御	×[3]	○
センサの入力値の計測	スイッチの入力計測	○	○
	光センサの光量計測	○	○
	ラインセンサの赤外線の反射量の計測	○	○
	距離センサの距離計測	○	○
通信機能	文字の交換	○	×
ドリトルGUI機能	センサ値の画面表示	○	×

ドリトルとプロッチの導入手順

□Windowsが導入されたコンピュータでドリトルとプロッチを通信するには、以下の設定をおこなう。このとき、ソフトウェア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. プロッチの標準ソフトウェアをインストールする
    - 株式会社山崎教育システムのサイト(http://protch.jp/#section2)のインストールページのインストール手順に沿って行う。
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のUSBポートにプロッチを接続する。1.が正しくインストールされていれば、自動的にドライバがインストールされる
    - ドライバのインストールが完了するのを待つ。完了すると□Prolific USB-to-Serial Comm Port (COM□)[4] 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
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。(可能であれば□□C:□直下に入れる[5])
4. ドリトルの起動
    - エクスプローラからdolittle.batを実行する。[6]

プロッチを開発モードで利用するための準備

開発モードでは、ドリトルからプロッチを制御するための通信プログラムを転送しておく必要がある。

通信プログラムは、独立モードのプログラムを転送しない限り一度だけ転送すればよい。通信プログラムの転送手順は(https://dolittle.eplang.jp/ch_protchrm/remote)を参照してください。

プログラムの入力から実行まで

ここでは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。

```
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。
」
```

手順2:プログラムの実行(実行ボタンを押す)
手順3:動作を開始する

- 実行画面の「動作開始」ボタンを押す(ドリトルからプロッチを制御する命令の送信が開始する)

手順4:動作を確認できたら動作を終了する

- 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。

プロッチのプログラムの基本

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。

書き方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」[7]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。
```

書き方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。開発モードでは、「動作終了」のタイミングによってはLEDが点灯した状態やモータが回転した状態が残ってしまうので□LEDを消す処理やモータを停止する処理を書くことが考えられる。

```
最初に実行＝「
□□□□□□
□□
```

書き方3
　今回は、LEDが点灯した状態が残ってしまうことを考慮し、最初に実行にLEDの**消灯**命令を書いた。**プロッチはLEDが２つ搭載されているため、**左側のLEDを制御するオブジェクトを「左ライト」、右側のLEDを制御するオブジェクトを「右ライト」にした。

```
左ライト!消灯。
```

書き方4
　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=「
□□□□
□□
```

書き方5
　今回は、LEDの点滅をするために、**点灯**命令と**消灯**命令を書いた。点灯と消灯命令の間には、プログラムの実行を止めるプロッチオブジェクトの**待つ**命令を書いている。待つの単位は秒である。
　「点灯」「1秒待つ」「消灯」「1秒待つ」を繰り返しているので1秒間隔でLEDが点滅していることになる。

```
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。
```

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、**押された?**を使うことでスイッチのONとOFFを検出できる。スイッチは左右に搭載しており、左側のスイッチを制御するオブジェクトを「左スイッチ」、右側のスイッチを制御するオブジェクトを「右スイッチ」にした。
　次のプログラムでは、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＝「
```

```
    「左スイッチ！ 押された?」！ 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、プロッチが箱に衝突し、左スイッチが押されたので左ライトが点灯している。



左スイッチ・右スイッチの命令一覧

命令	機能
読む	スイッチの入力を0か1で取得する
接触?	スイッチの入力を0か1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。
□□
```

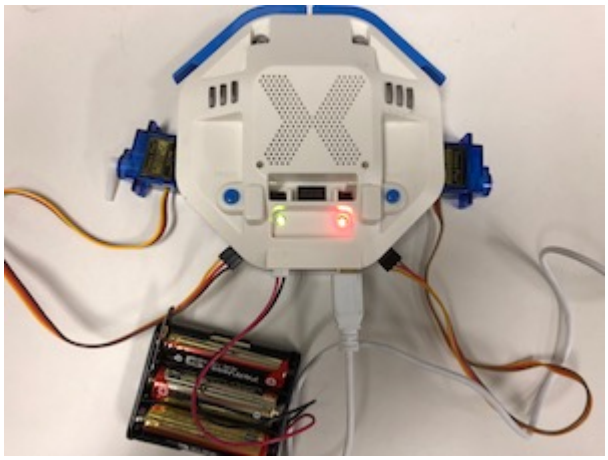
下図は、左光センサを手で隠すと暗くなるのでLEDが点灯している。

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=「
  左サーボモータ！180　角度。
  右サーボモータ！180　角度。
```

```
  プロッチ!1　待つ。
  左サーボモータ！0　角度。
  右サーボモータ！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＝「
    モータ！1　前進。
    モータ！1　後進。
    モータ！1　左折。
    モータ！1　右折。
    モータ！1　左回り。
    モータ！1　右回り。
    モータ!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＝「
    モータ!停止。
□□

繰り返し実行=「
    「左スイッチ!押された?」！なら「
        モータ!255 255 速度設定。
        モータ!1　前進。
    」そうでなければ「右スイッチ!押された?」なら「
        モータ!100 100 速度設定。
    モータ!1　前進。
    」実行。
□□
```

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。
□□
```

ラントレースカー

　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=「
    モータ!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「
        モータ！左回り。
  」そうでなければ「
        モータ！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。

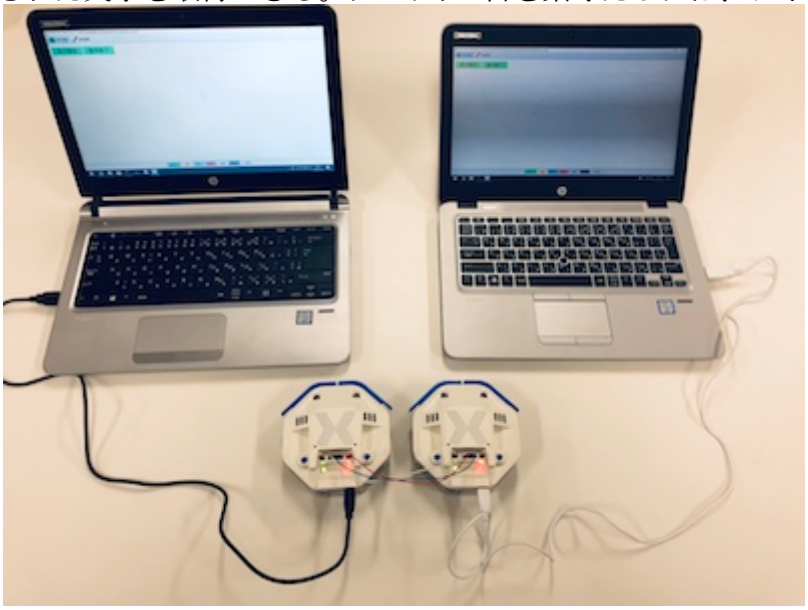
```
システム!"protchrm"使う。
最初に実行＝「
    モータ!停止。
」
繰り返し実行=「
    「(距離センサ!距離?)<40」！なら「
        モータ！1 停止。
    」そうでなければ「
        モータ!前進。
    」実行。
」
```

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。

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＝「
　シリアル!"こんにちは"出力。
□□
```

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!(シリアル!読む)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ボタンの動作で出力している。また、読む命令を0.1秒間隔で実行することで受信した文字をテキストエリアに書いている。

```
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。
　　送信ボタン＝ボタン!"送信"作る。
　　受信エリア＝テキストエリア！ 作る。
　　送信ボタン：動作＝「シリアル！ (送信フィールド！ 読む)出力」。
　　前回受信値="".
□□□
　　受信値＝シリアル！ 読む。
　　「前回受信値！ ＝受信値」！ なら「
　　　受信エリア！ (受信値)書く。
　　」実行。
　　前回受信値＝受信。
　　プロッチ!0.1 待つ。
　」!1000 繰り返す。
□□
```

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)
赤色：センサ類、青色：モータ類、緑色：拡張パーツ、黒色：ロボットを動かすために必要な部分
3)
現在対応を進めている
4)
COM□の部分は環境によって異なる。
5)
ファイルパスが256文字を超える場合はドリトルが起動しない問題がある
6)
「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。
7)
□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=1540117383

Last update: **2018/10/21 19:23**