マニュアルに戻る。

Studuinoと通信しよう

Studuinoというデバイスを使って、LEDやセンサなど外部デバイスでの入出力をしてみよう。Studuinoを使用するためには、ローカル版のドリトルが必要である。

Studuinoの入手

Studuinoは、株式会社アーテックが開発した汎用入出力ボードである。Arduinoと互換性があり、同社のロボティストシリーズのセンサーやモーターなどと接続して使うことができる。今回は、「ロボティスト アドバンス」という製品で動作を確認した。下図は同社のサイトによる説明図である。

ドリトルと通信するための設定

Windowsでの設定手順

Windowsが導入されたコンピュータでドリトルとStuduinoを通信するには、以下の設定をおこなう。このとき、デバイス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。なお、Windows7でのインストールを例として以下の手順で設定する。

1. USBデバイスドライバをインストールする
    - 株式会社アーテックのサイト（http://www.artec-kk.co.jp/studuino/）のダウンロードページから、USBデバイスドライバ(Windows)のインストーラをダウンロードする。
    - ダウンロードしたファイルを展開し、インストーラを実行する。画面の指示に従い、USBデバイス(PL2303)のドライバをインストールする。

- Studuinoの接続
    - PCのUSBポートにStuduinoを接続する。
    - 自動的にドライバがインストールされ、「Prolific USB-to-Serial Comm Port (COM) 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。[1)]
- 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。
- ドリトルの起動
    - エクスプローラからdolittle.batを実行する。[2)]

Mac(Mac)での設定手順

以下の手順で設定する。画面に確認ダイアログが表示された場合は、すべて承認して作業を進める必要がある。

1. USBデバイスドライバをインストールする
    - 株式会社アーテックのサイト（http://www.artec-kk.co.jp/studuino/）のダウンロードページ

から□□USBデバイスドライバ(Mac)□のインストーラをダウンロードする。
    - ダウンロードしたファイルを展開し、インストーラを実行する。画面の指示に従い、USBデバイス(PL2303)のドライバをインストールする。

- 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）のダウンロードページからMac用Java同梱版のドリトルをダウンロードする。
    - ダウンロードしたファイルをファインダーからクリックして実行する。
    - 表示された□Dolittle□を「アプリケーション」フォルダに入れる。[3)]
    - 外部機器を利用するための設定を行う。詳細は「はじめに」を□p.**mac□install**)を参照。
- Studuinoの接続
    - USBポートにStuduinoを接続する。
- ドリトルの起動
    - アプリケーションフォルダからDolittleを実行する。

ドリトルからプログラムを転送する

ドリトルで記述したプログラムは次の手順で転送する。

1. ドリトルの編集画面でプログラムを記述する。
2. プログラムを実行すると□Studuinoへの転送を実行しますか？」と表示されるので「はい(Y)□を選択する。[4)]
3. 10秒程度待つとStuduino基板に転送が行われ「転送完了」が表示される□□OK□を選択して閉じる。

センサーなどのパーツを接続する

以下に各種パーツを接続するポートを示す。[5)]

- ◎と○は使用可能、×は使用不能を表す。
- プッシュスイッチ◎を使用するとき、○の同名のポートは使用できない。たとえばプッシュスイッチA0を使用するとき、他のパーツはA0を使用できない。
- DCモーター◎を使用するとき、サーボモーター○の対応するポートは使用できない。たとえばDCモータ□M1を使用するときサーボモータ□D2, D4は使用できない。[6)]
- 加速度センサーはA4,A5を同時に使用する。

□	A0,A1	A2,A3	A4,A5	A6,A7	D2,D4	D7,D8	D9□D11	D12
□	□	□	□	□	M1	M2	□	□
加速度センサー	×	×	○	×	×	×	×	×
タッチセンサー	○	○	○	×	×	×	×	×
プッシュスイッチ	◎	◎	×	×	×	×	×	×
それ以外のセンサー	○	○	○	○	×	×	×	×
LED	○	○	○	×	×	×	○	×
DCモーター	×	×	×	×	◎	◎	×	×
サーボモーター	×	×	×	×	○	○	○	○

デジタル出力による動作の確認

動作を確認するには□A0ポートにLEDを接続した状態でStuduinoとPC本体をUSBケーブルと接続し、ド

リトルに次のプログラムを入力して実行する。接続したLEDが1秒ごとに点滅すれば、プログラムの転送は正常に行われている。

```
システム□"studuino" 使う。

最初に実行＝「
 □ST□"A0" デジタルLED□
 □□

繰り返し実行＝「
 □ST□"A0" 1 書く。
 □ST□1000 待つ。
 □ST□"A0" 0 書く。
 □ST□1000 待つ。
 □□
  ST□転送。
```

ドリトルからStuduinoを制御するプログラムは、次の形で記述する。先頭の行ではStuduinoを使うプログラムを作成することを示している。以後Studuino本体を表す「ST□」[7]というオブジェクトを使うことができるようになる。

```
システム□"studuino" 使う。
最初に実行＝「
 □...
 □□
繰り返し実行＝「
 □...
 □□
  ST□転送。
```

最初に実行の「…」の部分には、使用するセンサーの定義など、最初に1回だけ実行されるプログラムを書く。今回は最初に、LEDを接続するためにA0ポートを初期化した。 繰り返し実行の「…」の部分には、何度も繰り返して実行されるプログラムを書く。今回はA0ポートに接続したLEDを1で光らせた後、1秒間（1000ミリ秒間）待ってからLEDを0で消灯し、1秒間待っている。この動作を繰り返す。 転送を実行すると、プログラムがコンパイルされStuduinoに転送される。

アナログ出力

アナログ出力では、出力する値として0から255の数値を指定できる。次のプログラムでは、最初にD10ポートに接続されたLEDを「0」で消灯した後、1秒後に「127」の明るさで点灯させ、1秒後に「255」の明るさで点灯している。この動作を繰り返す。

```
システム□"studuino" 使う。

最初に実行＝「
 □ST□"D10" アナログLED□
 □□

繰り返し実行＝「
 □ST□"D10" 0 書く。
 □ST□1000 待つ。
```

```
□ST□"D10" 127 書く。
□ST□1000 待つ。
□ST□"D10" 255 書く。
□ST□1000 待つ。
□□
 ST□転送。
```

デジタル入力

デジタル入力は、センサーが接続されたポートの入力値を0と1で表すことで、スイッチ等のONとOFFを検出できる。ポートの入力値は、読むなどを使用することで取得することができる。 次のプログラムでは□A1ポートに入力があるとき□Studuino基板上のスイッチA1が押されて値が0になったとき）に□A0ポートのLEDを1で点灯している。入力がないときは0で消灯する。

```
システム□"studuino" 使う。

最初に実行＝「
 □ST□"A1" スイッチ。
 □ST□"A0" デジタルLED□
 □□

繰り返し実行＝「
 □□(ST□"A1" 読む) == 0」！なら「
 □□ST□"A0" 1 書く。
  」そうでなければ「
 □□ST□"A0" 0 書く。
  」実行。
 □□
  ST□転送。
```

デジタル入力を利用する場合は部品に応じた初期化命令を実行する必要がある。利用可能な部品名、ポート番号、初期化命令の使用例を表に示す。 **デジタル入力の利用可能部品**

部品名	ポート番号	初期化命令の使用例
スイッチ	A0□A3	ST□“A0”□スイッチ。
タッチセンサー	A0□A5	ST□“A1”□タッチセンサー。

アナログ入力

アナログ入力は、特定のポートの電圧を0から255の値で返すことで、各種センサーの入力値を検出できる。ポートの入力値は、読むなどを使用することで取得することができる。 次のプログラムでは、アナログのA3ポートに接続された光センサー の入力値によって□LEDの点灯と消灯の制御をしている。（光センサーを照明に向けたり手で隠すことで明るさを変えられる。明暗を判断する値（この例では120）は必要に応じて変更する必要がある。）

```
システム□"studuino" 使う。

最初に実行＝「
 □ST□"A3" 光センサー。
 □ST□"A0" デジタルLED□
```

```
□□

繰り返し実行＝「
 □□(ST□"A3" 読む) ＜ 120」！なら「
 □□ST□"A0" 1 書く。
  」そうでなければ「
 □□ST□"A0" 0 書く。
  」実行。
 □□
  ST□転送。
```

アナログ入力を利用する場合は部品に応じた初期化命令を実行する必要がある。利用可能な部品名、**ポート**番号、初期化命令の使用例を表に示す。 **アナログ入力の利用可能部品**

部品名	ポート番号	初期化命令の使用例
光センサー	A0□A7	ST□“A3”□光センサー。
音センサー	A0□A7	ST□“A4”□音センサー。
赤外線センサー	A0□A7	ST□“A5”□赤外線センサー。

その他のセンサ例（加速度センサー）

加速度センサーはA4□A5ポートの二つを同時に使用する。加速度センサーは、センサーに対するX軸、Y軸、Z軸の加速度を取得するセンサーである。重力加速度を検知することで傾きを取得することもできる。次のプログラムでは、加速度センサーのX軸方向の傾きを取得し、傾きに応じてLEDの点灯と消灯の制御をしている。（この例では計測した傾きが0以上なら消灯、そうでなければ点灯している。）

```
システム□"studuino" 使う。

最初に実行＝「
 □ST□加速度センサー。
 □ST□"A0" デジタルLED□
 □□

繰り返し実行＝「
 □□(ST□"x" 読む) ＜ 0」！なら「
 □□ST□"A0" 1 書く。
  」そうでなければ「
 □□ST□"A0" 0 書く。
  」実行。
 □□

 ST□転送。
```

サーボモーター

サーボモーターを使うと、0度から180度の範囲で任意の角度に回転させることができる。 サーボモーターが回転している間に並行して次の命令が実行される。複数のサーボモーターを回転させる命令を続けて書くと、それらは同時に回転する。サーボモーターの回転が終るのを待ちたいときは、待つなどを使って適切な時間だけプログラムの実行を止める必要がある。 次のプログラムでは□D10ポートとD11ポートに接続したサーボモーターの角度を1秒おきに変更することで回転させている。

```
システム□"studuino" 使う。

最初に実行＝「
 □ST□"D10" サーボモーター。
 □ST□"D11" サーボモーター。
 □□

繰り返し実行＝「
 □ST□"D10" 180 書く。
 □ST□"D11" 0 書く。
 □ST□1000 待つ。
 □ST□"D10" 90 書く。
 □ST□"D11" 90 書く。
 □ST□1000 待つ。
 □□

 ST□転送。
```

各入出力の利用例(ライントレースカー)

各入出力の応用としてライントレースカー[8]の制御を考える。ライントレースカーは左右のDCモーターを使い、地面に描かれた線のコースに沿って動く。線の上にいるかどうかを赤外線センサー□赤外線フォトリフレクタ）で判断する。次のプログラムでは□A7ポートに接続された赤外線センサーの入力値によって、線の上にいなければ右折し、線の上にいれば左折する。（床の白黒を判断する値（この例では120）は必要に応じて変更する必要がある。）

```
システム□"studuino" 使う。

最初に実行＝「
 □ST□DCモーター。
 □ST□"A7" 赤外線センサー。
 □□

繰り返し実行＝「
 □□(ST□"A7" 読む) ＜ 120」！なら「
 □□ST□150 右折。
  」そうでなければ「
 □□ST□150 左折。
  」実行。
 □□

 ST□転送。
```

CENTER:&show(http://dolittle.eplang.jp/data/wiki□image/image320/studuino□linetrace.jpg,50%,画像);

各入出力の利用例（センサーコントロールロボ）

複数のモーターとセンサーを組み合わせた例としてセンサーコントロールロボ[9]の制御を考える。このロボットは、３軸の傾きを取得する加速度センサーを使用している。最初に加速度センサーを地面と平

行になるように持ち、次に様々な方向に傾けることで表の動作を行わせることができる。 **操作方法**

操作	動作
前に倒す	前進
後ろに倒す	後進
右に倒す	右折
左に倒す	左折

今回は、以下の部品を使用し、ロボットを動かす。

- サーボモーター
- DCモーター
- 加速度センサー

次のプログラムでは、加速度センサーのX軸、Y軸の傾きを取得し、「前に倒した時」、「後ろに倒した時」、「右に倒した時」、「左に倒した時」のそれぞれの処理を行っている。傾きの値が10以下の場合は、ロボットは停止する。

CENTER:&show(http://dolittle.eplang.jp/data/wiki□image/image320/sensorcontrolrob.jpg,50%,画像);

```
システム□"studuino" 使う。

最初に実行＝「
 □ST□加速度センサー。
 □ST□DCモーター。
 □ST□"D9" サーボモーター。
 □ST□"D10" サーボモーター。
 □ST□"D11" サーボモーター。
 □□

繰り返し実行＝「
 □□(ST!"x" 読む)＞10」！なら「//前に倒した場合の処理
 □□ST□"D9" 140 書く。
 □□ST□"D10" 120 書く。
 □□ST□"D11" 130 書く。
 □□ST□500 後進。
  」そうでなければ□(ST!"x" 読む)＜-10」なら「//後ろに倒した場合の処理
 □□ST□"D9" 70 書く。
 □□ST□"D10" 110 書く。
 □□ST□"D11" 90 書く。
 □□ST□500 前進。
  」そうでなければ□(ST!"y" 読む) ＞10」なら「//右に倒した場合の処理
 □□ST□"D9" 40 書く。
 □□ST□"D10" 60 書く。
 □□ST□"D11" 50 書く。
 □□ST□500 右回り。
  」そうでなければ□(ST!"y" 読む)＜-10」なら「//左に倒した場合の処理
 □□ST□"D9" 140 書く。
 □□ST□"D10" 120 書く。
 □□ST□"D11" 130 書く。
 □□ST□500 左回り。
  」そうでなければ「//どこにも傾けていない場合の処理
```

□□ST□0 停止。
」実行。
□□

ST!転送。

1)
COM□の部分は環境によって異なる。
2)
「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。
3)
古いバージョンが残っているというダイアログが表示された場合は「置き換える」を選択する。
4)
□Windowsセキュリティの重要な警告」が表示された場合は「アクセスを許可する(A)□を選択する。
5)
この章の例では□LEDをA0とD10に、光センサーをA3に、赤外線センサーをA7に、DCモーターをM1とM2に接続している。
6)
V2.39では「ブザー」「カラーセンサー」「温度センサー」「超音波距離センサー」には対応していない。今後のバージョンで対応される予定である。
7)
「スタディーノ□□studuino□□ロボ」も使用できる。
8)
本節で扱うロボットは株式会社アーテックのサイトの作例集と組立説明書のページにあるライントレーサーを想定している。以下にURLを示す。http://www.artec-kk.co.jp/robotist/downloads.html
9)
本節で扱うロボットは株式会社アーテックのサイトの作例集と組立説明書のページにあるセンサーコントロールロボを想定している。以下にURLを示す。http://www.artec-kk.co.jp/robotist/downloads.html

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_studuino?rev=1514996377

Last update: **2018/01/04 01:19**