

Pepper プログラミング講座

第1回 Pepper でどんなことができる？

講師 静岡産業大学情報学部教授 佐野典秀 博士（工学）

TA（ティーチングアシスタント）佐野ゼミ（3年生）石川公康、石田竜士

学校名		名前	
-----	--	----	--

1. Pepper プログラミング講座の流れ

講座は合計3回、第1回(7/28) Pepper プログラミングを通して Pepper を知る

第2回(8/11) チームごとにテーマを設定、プログラムを考える

第3回(8/24) プログラムの完成、発表会（各チーム3分程度）

Pepper プログラミングコンテスト藤枝大会 11/9 BiVi

ソフトバンク成果発表会 2月 ソフトバンク本社（東京）

昨年の藤枝代表の結果報告 中学部門 全国優勝 葉梨中学

副賞 シリコンバレー訪問

2. Pepper について知ろう。

口・・・スピーカー 左右の丸いところに2個（耳と思っている人が多いけど。）

耳・・・マイク 頭の前後左右に4個（話しかけるときは頭に話かけようね）

目・・・RGB カメラ 額（ひたい）と口のところ

3D センサー 右目のところ

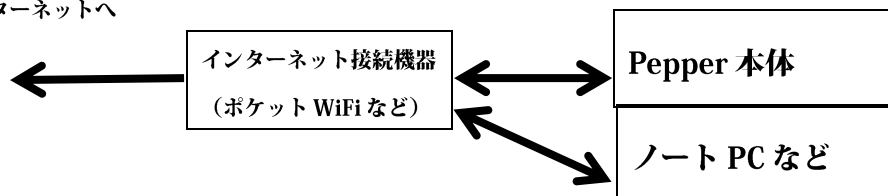
レーザー 足元でレーザー線を発射して反射を検知して周囲の状況を確認

触覚・・・タッチセンサ 頭に3個、左右の手の手甲（こう）

3. Pepper で使う機器の構成

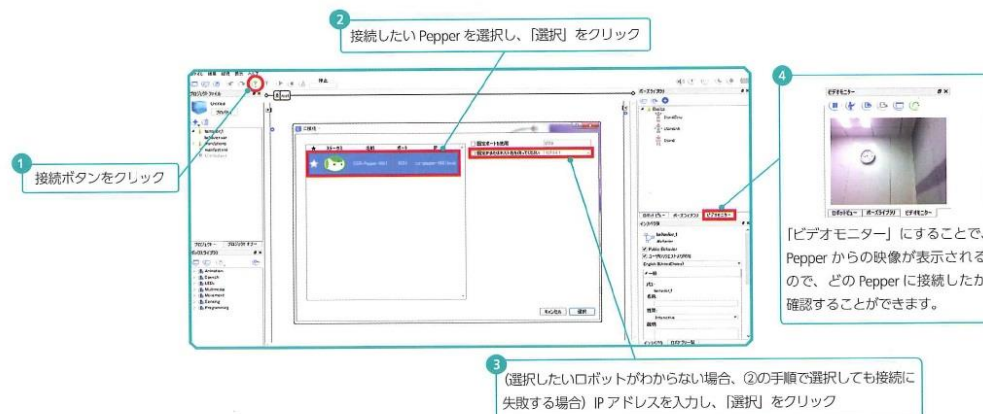
Pepper は必ず、インターネットに接続して使います。以下の通信機器を使います。

インターネットへ



4. Pepper のプログラミングに使うソフトウェア（コレグラフ）

4. 1 Pepper とノート PC を接続します。




※ 胸のボタンを1回軽く押すと、Pepper の IP アドレスを確認することができます。

4. 2 Pepper のモードを切替えます。

Pepper には2つのモードがあります。オートノマスライフモードとプログラム開発モードの2つです。

プログラムを開発するときは、プログラム開発モードにします。(オートノマスライフ機能をオフ (OFF) に)

1. Choregraphe のツールバーから「オートノマスライフ」ボタン  をクリックし設定を OFF にする



Pepper がセーフレストの姿勢になります。

! ・オートノマスライフボタンをクリックしてもすぐに Pepper がセーフレストの姿勢にならない場合があります。信号の転送に時間がかかっているので、何度もボタンをクリックせずそのまましばらく待ちます。

眠った Pepper を起こしてあげます

2. Choregraphe のツールバーから「Wake UP」ボタン  をクリックする



Pepper が胸を張ったような姿勢になります。

4. 3 Pepper を基本姿勢にします。

3. 「ポーズライブラリ」から、Pepper の姿勢を「Stand」にする

Choregraphe の表示メニューから「ポーズライブラリ」を選択

ロボットビューの横の「ポーズライブラリ」をクリック

「ポーズライブラリ」から「Stand」を選択

「Stand」の姿勢になる



5. 実際にコレグラフィを使ってプログラミングしてみましょう。

コレグラフィでは、ボックスという箱のようなものを線でつないでプログラミングしていきます。

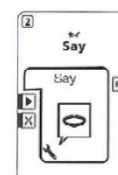
5.1 Pepper が話す

使うボックス (ボックスは左下のボックスライブラリの中に色々なボックスがそろっています。)

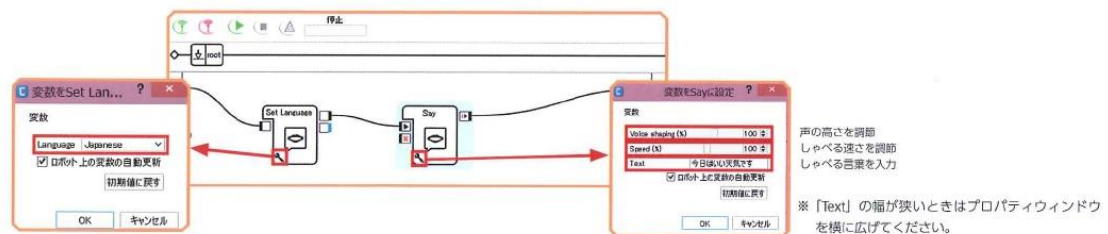
Set Languages・・・どこの国の言葉を話すかを選びます。

Say・・・話す内容や、声の高さ、スピードを設定します。

VoiceShaping 声の高さを調整 「130」が Pepper らしい声
Speed しやべるスピード 「110」が Pepper らしい声



では、この2つのボックスをコレグラフの左下から真ん中の広いフローダイアグラムというところにマウスでドラッグして（マウスの矢印をボックスの名前の上ののせて、マウスの左ボタンを押しながら）移動しましょう。適当な場所に来たら、マウスの左ボタンを離しましょう。あとは、フローダイアグラムの左上のプログラムの始まりから、ドラッグで順番にボックスをつないでいき、最後に右上のプログラムの終わりとつなぎましょう。



Set Language ボックスの左下のスパナ（工具）のマークをクリックしましょう。「変数を Set Language に設定」のダイアログボックスという箱が開きます。Language の右の下向きの三角をクリックして「Japanese」日本語を設定しましょう。

同じようにして、Say ボックスの左下のスパナ（工具）のマーク（プロパティといいます）をクリックして、声の高さや、スピード、話す内容を設定しましょう。話す内容は四角の枠をクリックして、その中に直接、話させたい言葉を入力していくことができます。日本語の入力に時間がかかるので、今回は、代表的なあいさつなどをあらかじめ入れたファイルを用意しました。コレグラフの右上の三つのボタンの左はしの「一」を押すと、一旦、コレグラフが閉じます（見た目、下に隠れるだけでプログラムを終了したわけではないです）。そして「Pepper の話す言葉」のアイコンをダブルクリックしてファイルを開きます。

好きな言葉を選んでドラッグして、メニューの「編集」→「コピー」をクリックします。これでメモリに言葉が記憶されました。次に一旦閉じていたコレグラフを開きます。そのためには画面の下に並んでいる中から「C」を選んでクリックします。開いたコレグラフで、さっきの Say ボックスの中の話す言葉を入れる枠の中で言葉を追加するところをクリックして選択し、マウスを「右クリック」→「貼り付け」で話す言葉が入力できます。

課題 1

好きな言葉を Pepper が話すプログラムを作ろう。

Pepper の話す言葉の声の高さやスピード、強弱などをいろいろと調整してみよう。

5.2 プログラムを実行してみよう。



今回の講座では、2組で1台のPepperを交替で使います。お互いに譲り合って使うことにご協力ください。

どちらの組が最初にPepperとつながかを決めましょう。


最初につなが組の人は、先ほどの 3.1 PepperとPCを接続します を参考にしてPepperとつながみましょう。

■Pepperで実行する

実行する前にPepperとPCがつながっていることを確認してください。

作成したアプリをPepperで実行するには、ツールバーの「アップロードして再生」ボタンをクリックしてください。また、プログラムを停止する場合は、「停止」ボタンをクリックしてください。



- ・Pepperへの転送に時間がかかることがあります。連続して再生ボタンを押すと接続が不安定になりChoregrapheの再起動が必要になる場合がありますので注意してください。
- ・動きに関するプログラムを途中で停止した場合、Pepperの動きも途中で止まります。
その場合はモーターのオーバーヒート対策として、「ポーズライブラリ」から「Stand」姿勢に戻してください。
- ・プログラムが実行されない場合がありますので、再生ボタンを押す前に必ず「停止」ボタンを押してから再生するようにしてください。

Pepperが思い通りに話しましたか。

私たちの話す日本語にも地域によって方言があったり、イントネーション（言葉の切り方や、リズム、音の高さなど）が違うことがありますね。

Pepperの話す言葉も時々、あれえ、というような話し方の場合があります。

その場合は、わざと、「ひらがな」や「カタカナ」、違う感じなどを使うとしっかりとすることがあります。

いらっしゃいませ	→ いらっしゃいませーっ
準備はいいですか？	→ 準備はいいですかぁ？
はい	→ 葉あーアイツツ、
やったー	→ ヤッタアーーーーっ、
わかりました	→ わかりましたーあ！、
残念	→ ざぁんねーんっ！

など、余裕がある高学年の方は、いろいろと試してみてくださいね。

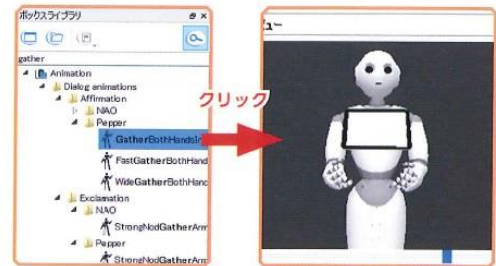
5.3 Pepper に動きをつけてみよう。

使うボックス

-  GatherBothHandsInFront_01 両手を前に持ってくる
-  PointFrontLeftArm_LeanLeft_01 左手をさし出す
-  StrongHeadShakeSpreadArms_FlexEnd_01 ... 頭を振りながら両手を広げる
-  OfferLeftHandThenRightHand_01 両手を腰まで上げて、右手と左手を交互にさし出す



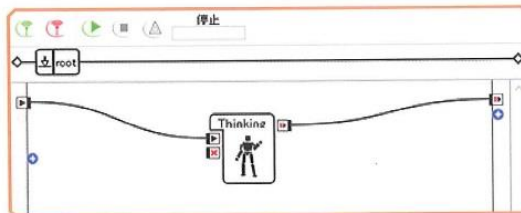
- ・ Pepper の近くに人や机などの障害物がある場合、自動的に動きが制御されます。
- ・ 動作する Pepper の腕などに当たらないように注意します。
- ・ Pepper の腕などを手で無理やり動かさないように注意します。



インスペクタで動きの確認ができます。

例えば、Thinking・・・「頭をかく」動作をさせてみましょう。

「Thinking」ボックスを使い、下記の通り線で接続します。Pepper が左手を上げ、頭をかく動作をします。動作が終わった後、Pepper は左手を上げた状態で停止するので、ポーズライブラリから「Stand」をダブルクリックして、モーターに負荷のかからない姿勢に戻します。



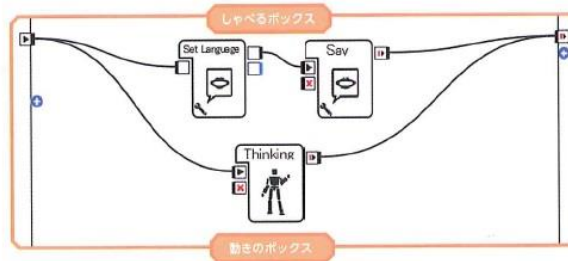
- ・ Pepper は人間と同じで、腕を上げ続けると疲れてしまいます（モーターに負荷がかかり続ける）。ポーズライブラリから「Stand」をダブルクリックします。

「Thinking」のボックスをいろいろなボックスに変えて、さまざまな動作を Pepper にさせてみましょう。

5.4 Pepper が動きながら話すプログラムを考えてみよう。

線を並列に（2本に分かれて）つなぐと「〇〇しながら」「△△する」という同時に2つのことを行うプログラムを作ることができます。

例えば、さきほどの 4.1 Pepper が話す と 4.3 Pepper に動きをつけてみよう の2つを並列でつないでみましょう。例えば、こんな感じで

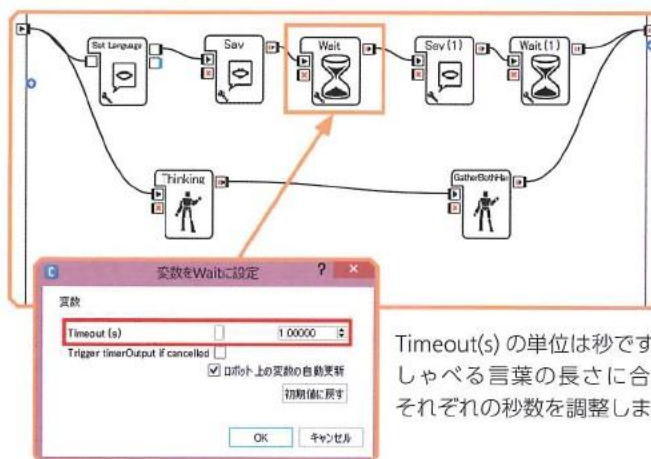


- ・しゃべる言葉が短く、動いている途中でしゃべり終わると動きが途中で止まります（このプログラミング時の正常な動作）。しゃべる言葉の長さを色々変えて試してください。
- ・直列でつないだ場合、しゃべり終わってから動くという動作になります。

コレグラフでは、二つの分かれたプログラムのどちらか早い方の処理が終了するとプログラムが終わります。そうすると、短い言葉話す場合は、話が終わると動作の途中で Pepper は動きが止まってしまいます。また、長い言葉話す場合は、動作が終わると話している途中でも、話を中断して止まってしまいます。話す言葉の長さを変えて、試してみましょう。

このように、二つ以上の動作を上手くタイミングを合わせる（同期をとるといいます）ことは意外と難しいのです。

上手くタイミングを合わせるには、処理の時間を待たせて（Wait を使います）合わせることができます。以下のプログラムで確認してみましょう。



Timeout(s) の単位は秒です。
しゃべる言葉の長さに合わせて、
それぞれの秒数を調整します。

これまでは一方的に Pepper が話したり動いたりしていたね。でも、私たちは
相手の反応を見て、話したり動いたりするよね。この反応を見るのにセンサを使うよ。

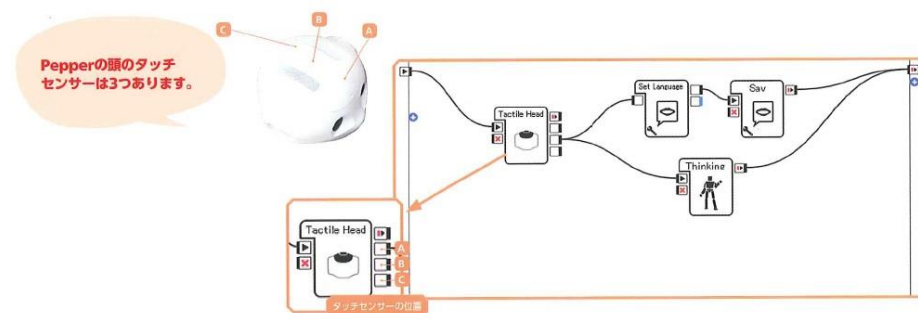
5.5 Pepper のセンサーを使ってみよう。

5.5.1 Pepper のタッチセンサーを使ってみよう

Pepper には、頭と、手の甲にタッチセンサーがあります。今回は頭のタッチセンサーを使ってみます。

新しく使うボックス

Tactile Head …頭の前(A)、中央(B)、やや後ろ(C)のいずれかを触ると以降のプログラムを実行する



Point

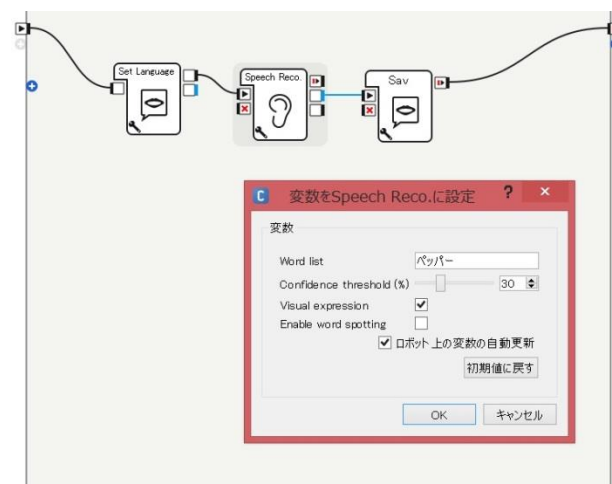
- ・頭のタッチセンサーの前と後ろも試してみて、センサーの範囲の違いを実際に感じてください。
- ・タッチセンサーにはたたいたりせず、やさしく触れるように注意します。
- ・頭のタッチセンサーはどれか1か所のみ使うようにします。3か所を使って分岐処理をしようとすると、同時にタッチセンサーにさわる場合があり、Pepper が正常に動作しません。

上記のプログラムでは、頭の中央を触ることでプログラムがスタートします。

5.5.2 Pepper の音センサーを使ってみよう。

新しく使うボックス

Speech Reco … 聞き取った言葉に応じて、反応する

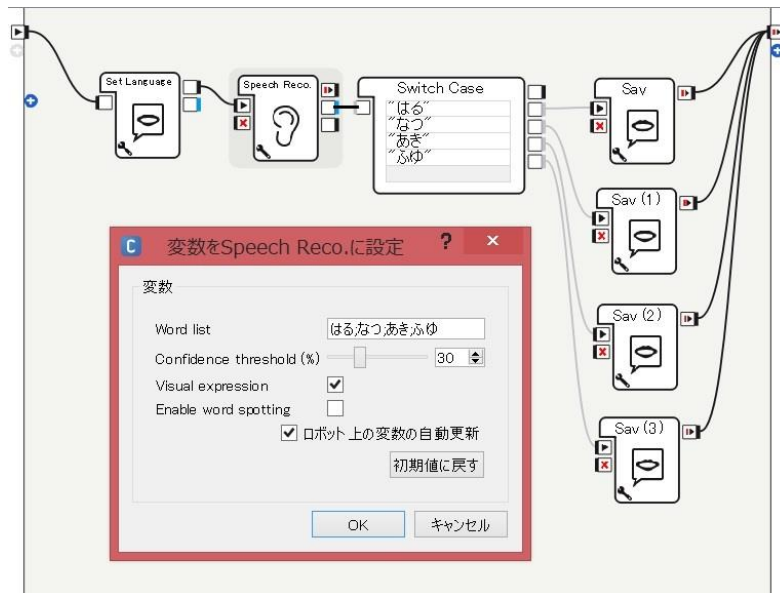


Say ボックスには、「はい」と入れてみよう。ペッパーと呼びかけると「はい」と応えるよ。

音センサーの応用編

さらに Switch Case を使ってみよう

「はる;なつ;あき;ふゆ」を聞き分けて答えるプログラム はる; ←半角のセミコロン



“はる” ←半角のダブルクォーテーション

Speech Reco Switch Case Say

例： はる; “はる” → 春は花がきれいでいいですね (Say ボックス)

なつ; “なつ” → 夏は暑いですね (Say ボックス)

あき; “あき” → 秋は紅葉がきれいでいいですね (Say ボックス)

ふゆ; “ふゆ” → 冬は寒いですね (Say ボックス)

5.6 Pepper のディスプレイに画像を表示してみよう。

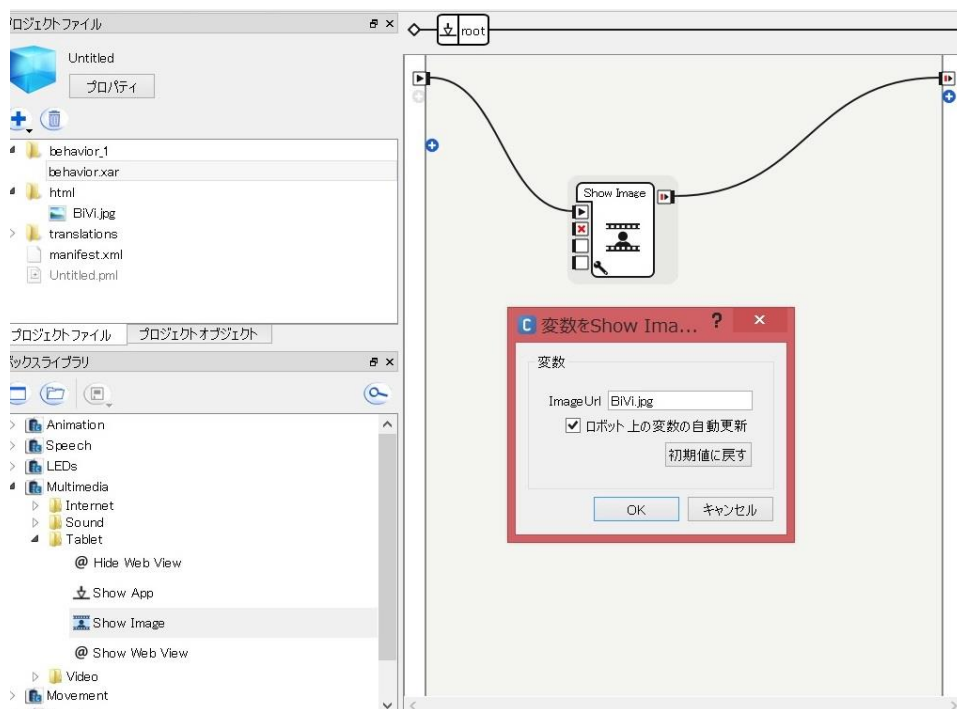
新しく使うボックス

Show Image

準備 0. 画像を用意する。例 BiVi.jpg

画像は、.jpg .png .gif のいずれか

1. プロジェクトパネルの「+」をクリック
2. 新規フォルダを作成「html」 Create クリック
3. プロジェクトパネルの「+」をクリック
4. ファイルのインポート BiVi.jpg を選択
5. BiVi.jpg を「html」にドラッグ&ドロップ
6. Show Image をフローダイアグラムに置く
7. Show Image のプロパティを開いて Image URL に BiVi.jpg を入れる



人間は、いろいろな「入力」を感覚器官（目、耳、触覚など）で感じ、いろいろな「出力」（声、動きなど）を使って相手とさまざまなやり取り（コミュニケーション）をしているよ。

Pepper も同じように「入力」と「出力」を持っているよ。Pepper の入力は

「タッチセンサ」（頭の前後、右手と左手）、「音センサ」（頭部のマイク）、「タッチパネル」（画面を分割してどこを触ったか）などが使えるよ。

Pepper の出力は、「声」、「動作」、「画面」などが使えるよ。

それぞれの入力の特徴

「タッチセンサ」・・・誤作動が少ない。でも、離れたところからは触れない。

「音センサ」・・・離れたところからでも伝わる。でも、誤作動が多い。

「タッチパネル」・・・選択肢の数を増やせる。誤作動が少ない。でも、離れたところからは触れない。

それぞれの出力の特徴

「声」・・・少し離れたところでもわかる。でも、あまり離れると聞こえない。

「動作」・・・離れたところからでも見える。でも、動作の違いが見分けにくい。

「画面」・・・伝えられる情報量が正確で多い。でも、離れると見えない。

以上の特徴を考えながら、自分たちのチームが作る「身の回りで役に立つ Pepper」を次回まで

に考えてこよう。役に立つ Pepper を考えるときは、

「どんな場面で使うの？」

「誰が使うの？（誰（どんな人）の役に立つの？）」

をまず、しっかりと整理してから、どんなサービスを提供すれば「役に立つ」かなと考えを進め

ていこう。次回までに宿題のプリントの枠を埋めて来よう。