平成19年 一日大学院

コンピュータでレゴを操ろう!!

知識科学教育研究センター
金井 秀明

授業の流れ

- レゴについて
- 動かしてみよう！！
- プロジェクト
  - 触覚をつける
  - 視覚をつける
- まとめ

レゴについて

レゴ（LEGO)

- レゴ
  - 語源は，デンマーク語のLeg godt（よく遊べ)
  - プラスチック製の組み立てブロック

レゴ（LEGO)#2

- 歴史
  - 1949年デンマークの玩具会社がが「Automatic Binding Bricks」と名付けて発売
  - 1953年「LEGO Bricks」として発売

授業で使うレゴは

- レゴマインドストームNXT
  - パソコンで，レゴブロックで作ったロボットを動かすことができる「ロボット開発キット」
  - 日本では2006年６月に登場

デモ

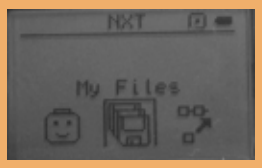
動かしてみよう!!

動かしてみよう

「選択」

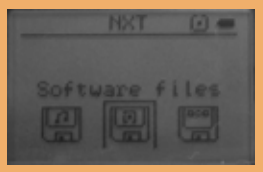
「矢印」

選択を押す



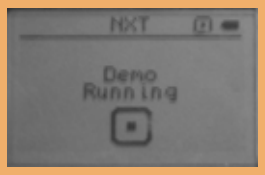
「NXT」起動

選択を押す



矢印で「Demo」を選び，選択を押す

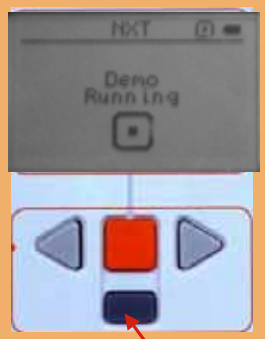
選択を押す



Demo実行中

「実行」をとめるには

Demo実行中

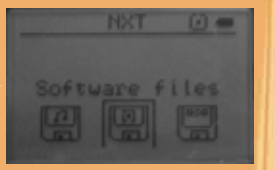


キャンセル

キャンセルを1回押す

動作がとまる

キャンセルを2回押す



レゴを操る方法

レゴを操る＃１

- 「レゴを操る」＝レゴを「ある状況（状態）に応じて，特定の動作をさせる」こと
- 予め，「操る内容」をレゴに覚えさせなければならない．
- 「操る内容」が記述されたもの＝「プログラム」

レゴを操る#2

- 要素
  - 状況を認識する部分（入力部）：センサー
  - 「操る内容」を記憶・処理する部分：コンピュータ
  - 動作する部分（出力部）：モーターなど

センサー#１

- タッチセンサー（触覚，皮膚）
  - 何かにぶつかって押されたり，離れたりすることを感知できる．
- 光センサー（視覚，目）
  - 明暗を感知できる．

タッチセンサー

光センサー

センサー#2

- 超音波線センサー（視覚，聴覚）
  - 物体との距離を計測できる．±3cmの誤差（255cmまでの距離）
- サウンドセンサー（聴覚，耳）
  - 周囲の音の大きさを計測できる．

超音波センサー

サウンドセンサー

モータ（動作をする部分）

- レゴに何らかの動作を与えるもの
- モータの回転角度，回転数が変えられる．
- 例：
  - タイヤを回す．
  - ロボットの関節を動かす．
  - ・・・

コンピュータブロック(NXT)

- 「操る内容」（プログラム）に従って，センサーからの情報の受信やモータの制御を行う．

NXTの構成

出力ポート：
モータなど接続

パソコン接続

ディスプレイ

スピーカー

操作ボタン

項目の左移動

項目の右移動

キャンセル

メニュー選択の決定

入力ポート：
センサー接続

まとめ

- 「レゴを操る」には，予め，「操る内容」を決めなければならない．
- 「操る内容」が記述されたもの＝「プログラム」
- 「プログラム」を「コンピュータ」に覚えさせる．

レゴは，「プログラム」に従って動く．

プログラミング

プログラミングとは

- 「プログラム」を作ること
- 英語：
  - Programming = Program+ing

プログラミング言語: NXC

- Not eXactly C (NXC)
  - C言語ライクな言語
- 関連リンク
  - http://bricxcc.sourceforge.net/nbc/nxcdoc/index.html

プログラミングの流れ

Step 1

プログラムの作成

Step 2

プログラムをNXTに記憶させる

Step 3

プログラムを動かす

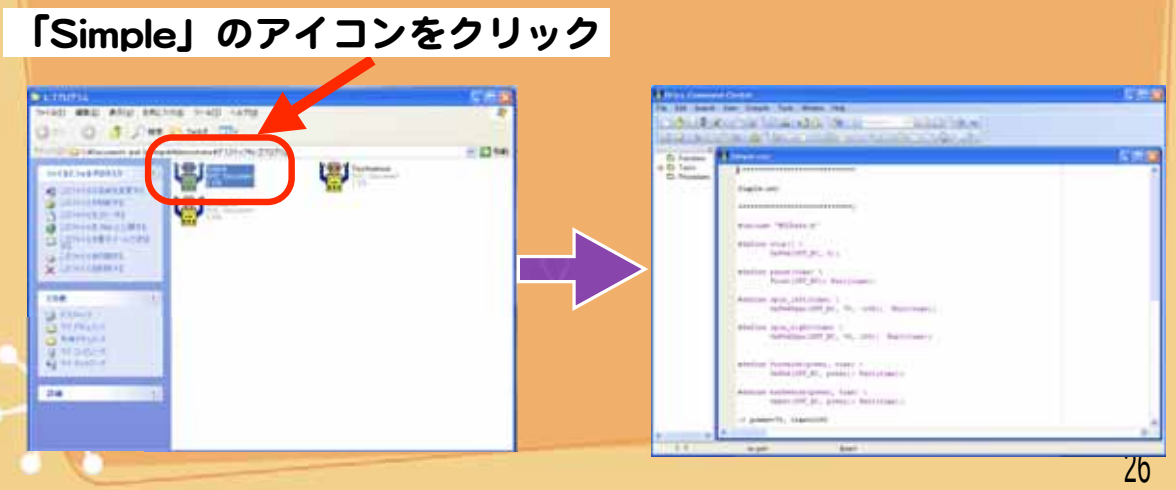
プログラミングの練習

練習:「Simple」

- プログラム「Simple」を開く
- NXTとパソコンの接続
- NXTに記憶させる
- プログラムを動かす

プログラムを開く

- プログラム「Simple」をクリックする.

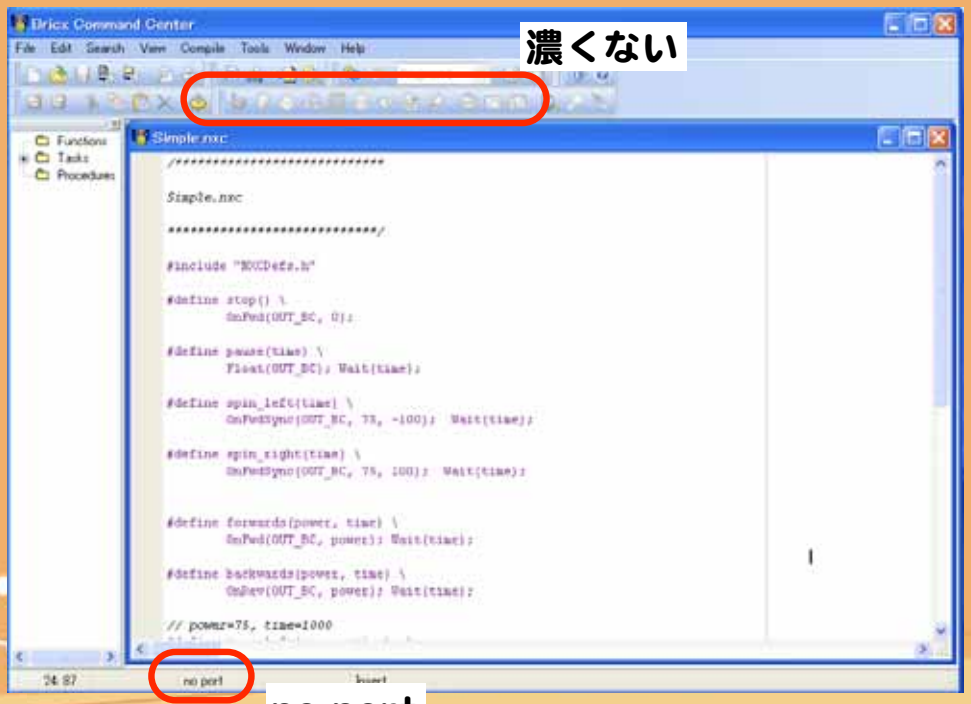


NXTとパソコンの接続

- NXTとパソコンをUSBケーブルでつなぐ.

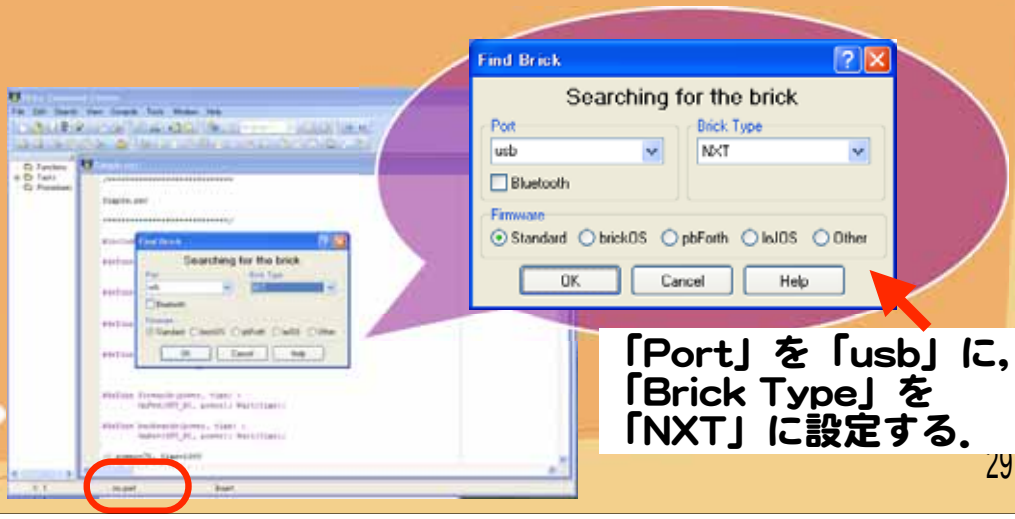
- NXTの電源をONする.
  - NXTの「オレンジのボタン」を押す.

未接続のときの画面



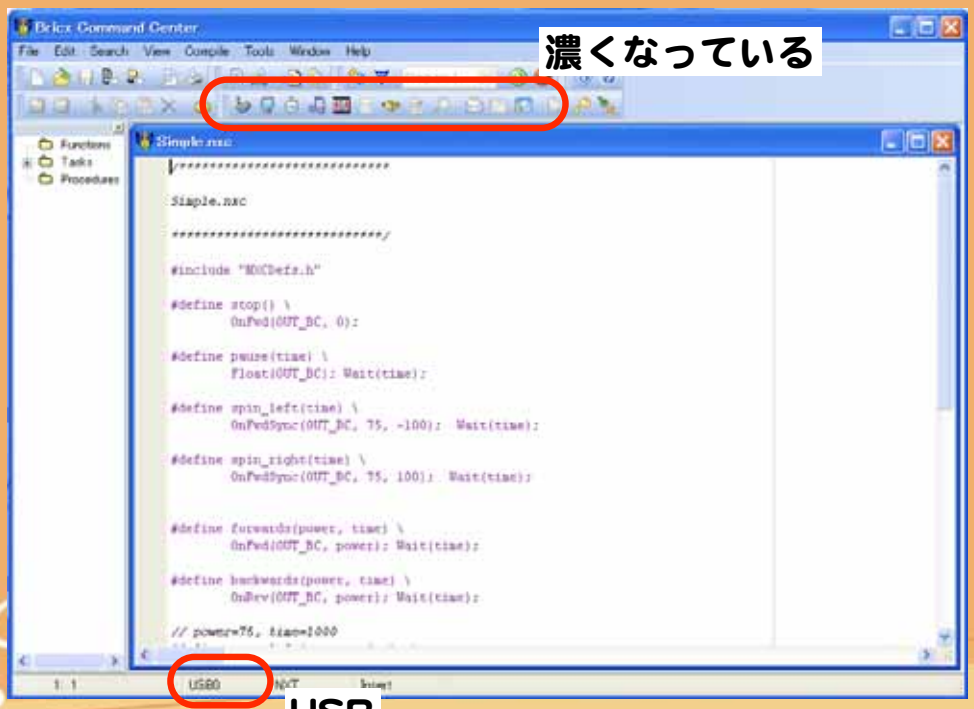
未接続時の対処法

- メニュー「Tools」から「Find Brick」をクリックする.
- 「Find Brick」ウィンドウが開く



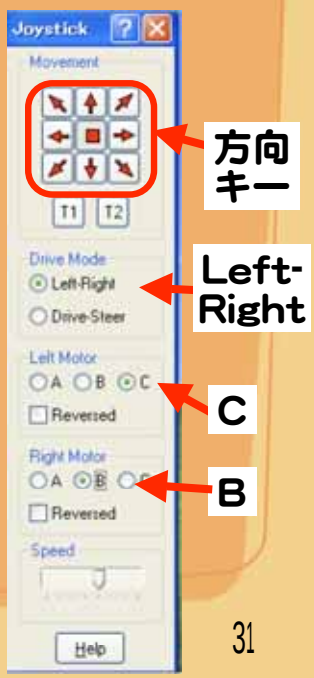
「Port」を「usb」に,「Brick Type」を「NXT」に設定する.

接続完了の時の画面



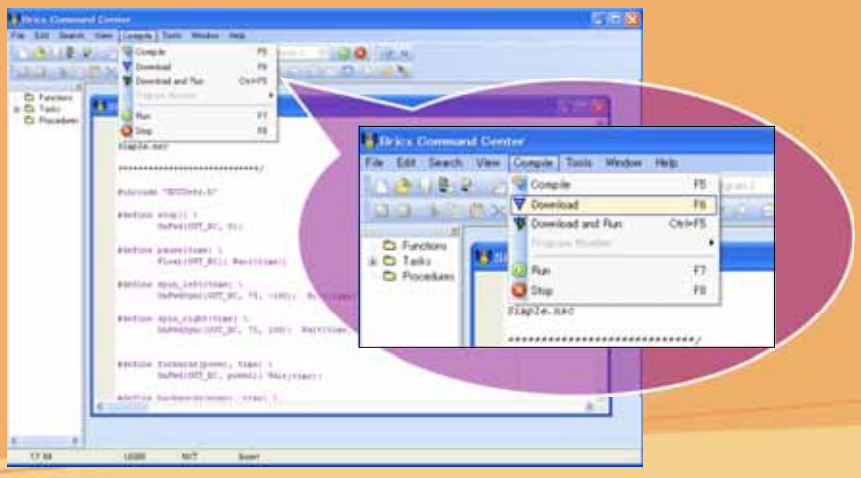
接続確認方法

- メニュー「Tools」から「Brick Joystick」をクリックする．「Joystick」ウィンドウが開く．
  - 方向キーでレゴが動く場合，接続完了している．
  - 方向キーでレゴが動かない場合，「未接続時の対処法」の作業を行う．



NXTに記憶させる

- プログラムをNXTに記憶させることを，「ダウンロード(Download)」という．

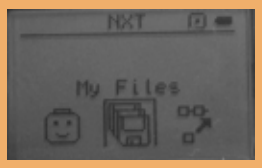


動かしてみよう

「選択」

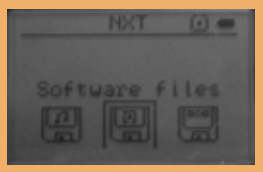
「矢印」

選択を押す

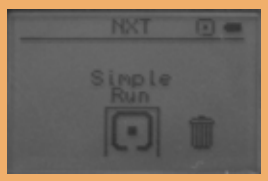


「NXT」起動

選択を押す



矢印で「Simple」を選び，選択を押す



選択を押す

Simple実行中

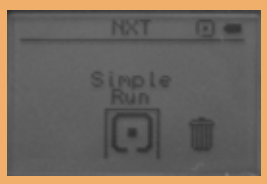
「実行」をとめるには

Simple実行中

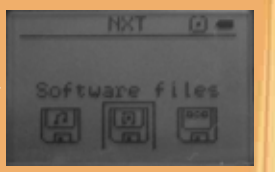
キャンセル

キャンセルを1回押す

動作がとまる



キャンセルを2回押す



プログラムの説明

プログラム:Simple

- 複数の動作命令でできている.

```
task main()
{
  forwards(75, 1000);
  turn_right(75, 1000);
  forwards(100, 1000);
  turn_left(75, 1000);
  forwards(100, 1000);
  spin_right(3000);
  forwards(100, 1000);
  spin_left(3000);
}
```

動作命令

動作命令（モータ制御）

前進	：forwards(スピード, 時間);
後進	：backwards (スピード, 時間);
停止	：stop();
一時停止	：pause(時間);
右折	：turn_right(スピード, 時間);
左折	：turn_left(スピード, 時間);
右回転	：spin_right(時間);
左回転	：spin_left(時間);

スピード：モータのスピード（0から100）
時間　　：動作する時間（1000=1秒）

プログラムの説明

```
task main()
{
  forwards(75, 1000);
  turn_right(75, 1000);
  forwards(100, 1000);
  turn_left(75, 1000);
  forwards(100, 1000);
  spin_right(3000);
  forwards(100, 1000);
  spin_left(3000);
}
```

前進（75のスピード，１秒間）
右折（75のスピード，１秒間）
前進（100のスピード，１秒間）
左折（75のスピード，１秒間）
前進（100のスピード，１秒間）
右回転（3秒間）
前進（100のスピード，１秒間）
左回転（3秒間）

動作命令

プログラムの改造

プログラムの改造

- プログラムに従って，レゴは動く．
- つまり，「プログラムを改造すること」で，「レゴに異なった動作をさせること」ができる．

プログラムの改造#１

- 動作命令の変更：削除

```
task main()
{
  forwards(75, 1000);
  turn_right(75, 1000);
  forwards(100, 1000);
  turn_left(75, 1000);
  forwards(100, 1000);
  spin_right(3000);
  forwards(100, 1000);
  spin_left(3000);
}
```

```
task main()
{
  forwards(75, 1000);
  turn_right(75, 1000);
  forwards(100, 1000);
  turn_left(75, 1000);
  forwards(100, 1000);
  spin_right(3000);
  forwards(100, 1000);
  spin_left(3000);
}
```


プログラムの改造#2

- 動作命令の変更：追加

```
task main()
{
  forwards(75, 1000);
  turn_right(75, 1000);
  forwards(100, 1000);
  turn_left(75, 1000);
  forwards(100, 1000);
  spin_right(3000);
  forwards(100, 1000);
  spin_left(3000);
}
```

```
task main()
{
  turn_right(75, 1000);
  pause(2000);
  turn_left(75, 1000);
  forwards(100, 1000);
  spin_right(3000);
  forwards(100, 1000);
  spin_left(3000);
}
```

プログラムの改造#3

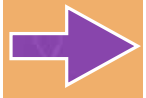
- 動作する「スピード」や「時間」を変更する.
  - スピード：０から１００，整数
  - 時間：０以上の値，整数，1000=1秒

```
task main()
{
   turn_right(75, 1000);
   pause(2000);
   turn_left(75, 1000);
   forwards(100, 1000);
   spin_right(3000);
   forwards(100, 1000);
   spin_left(3000);
}
```


プログラムの改造#4

- 動作を繰り返す
  - while(true) {繰り返す命令}

```
task main()
{
   turn_right(75, 1000);
   pause(2000);
   turn_left(75, 1000);
   forwards(100, 1000);
   spin_right(3000);
   forwards(100, 1000);
   spin_left(3000);
}
```



```
task main()
{
   while(true) {
   turn_right(75, 1000);
   pause(2000);
   turn_left(75, 1000);
   forwards(100, 1000);
   spin_right(3000);
   forwards(100, 1000);
   spin_left(3000);
   }
}
```

実習

- 「プログラムの改造#1～4」のような改造を，「Simple」に行ってください．
- そして，実際にレゴの動きが変わるか確認してください．

プロジェクト

プロジェクト

- 触覚をつける
  - タッチセンサを使って，「障害物に衝突した後に回避する動作」をつくる．
- 視覚をつける
  - 超音波センサーを使って，「障害物に衝突する前に回避する動作」をつくる．

プロジェクト

- 触覚をつける
  - タッチセンサを使って，「障害物に衝突した後に回避する動作」をつくる．
- 視覚をつける
  - 超音波センサーを使って，「障害物に衝突する前に回避する動作」をつくる．

触覚をつける

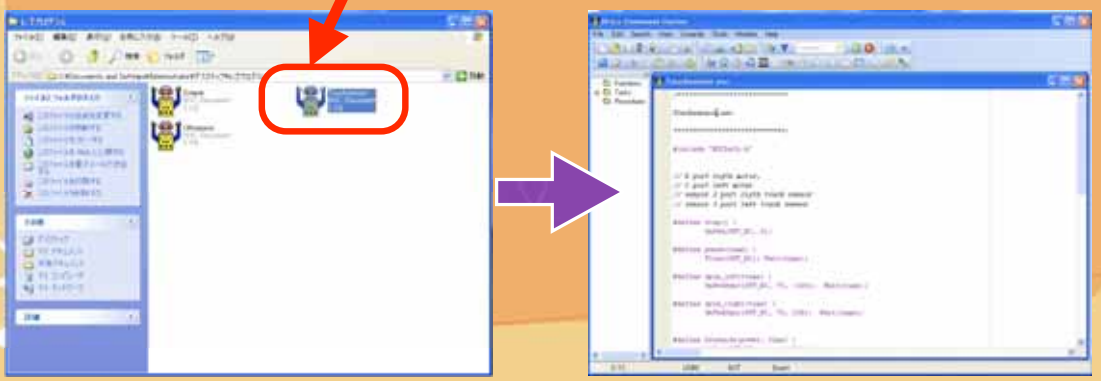
- タッチセンサの装着
- プログラム「Touchsensor」 を動かす
- プログラムの改造
  - 回避動作の変更

タッチセンサの装着

- タッチセンサから出ているコードを入力ポートにさしこむ.

プログラム「Tocuhsensor」を開く

「Touchsensor」のアイコンをクリック

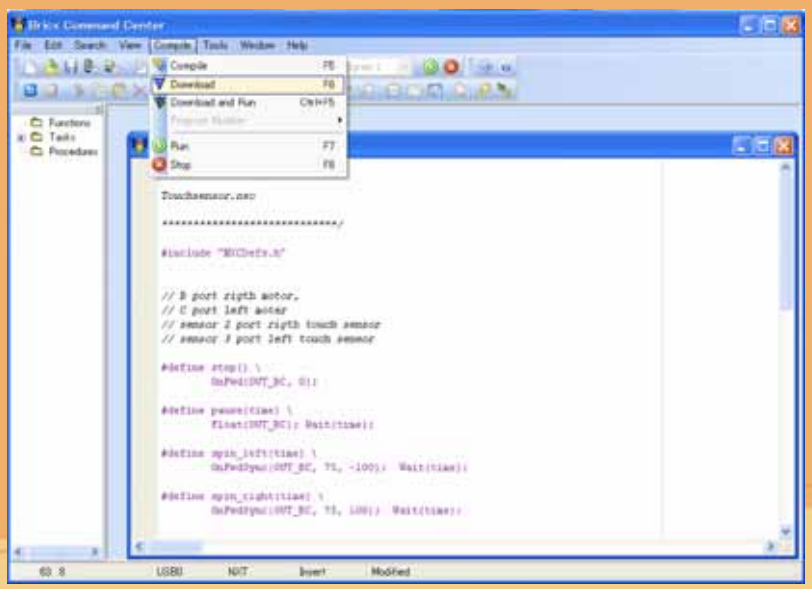


NXTとパソコンとが未接続の場合

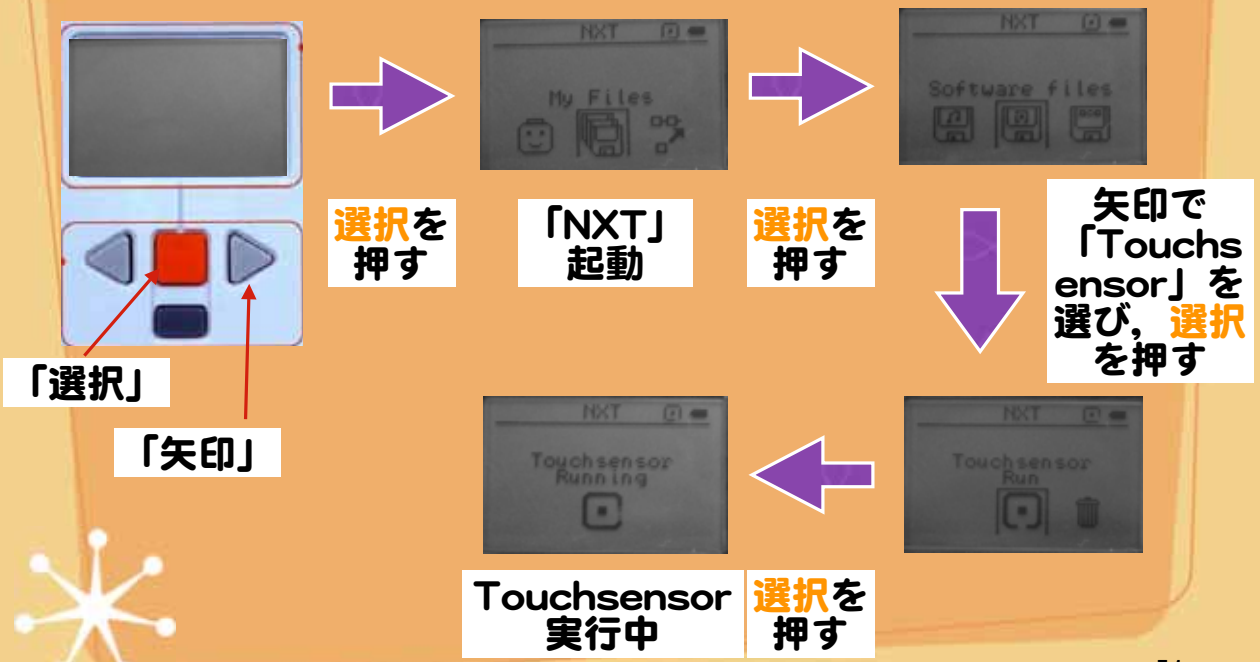
- 「２７ページから３１ページ」の手順で，NXTとパソコンを接続し直してください．

NXTに記憶させる

- プログラム「Touchsensor」をNXTにダウンロード



動かしてみよう



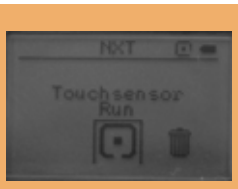
「実行」をとめるには

Touchsensor実行中

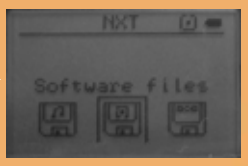
キャンセル

キャンセルを1回押す

動作がとまる



キャンセルを2回押す



プログラムの全体

```
task main()
{
  . . .
    while( true ){
        //Right sensor touched
        if( Sensor(IN_2) ){
            backwards(100, 500);
            turn_left(100, 250);
        //Left sensor touched
        }else if( Sensor(IN_3) ){
            backwards(100, 500);
            turn_right(100, 250);
        }else{
             forwards(100, 100);
        }
    }
}
```

動作の流れを制御する

- 条件分岐：
  - 「ある条件（場合），特定の動作を実行させる」
- 例：
  - 「もし壁に衝突したら，後進して右折する」
    - 条件：壁に衝突したら？
    - 動作：後進し右折する．

条件分岐の構文

```
//Right sensor touched
if( Sensor(IN_2) ){
    backwards(100, 500);
    turn_left(100, 250);
}

//Left sensor touched
else if( Sensor(IN_3) ){
    backwards(100, 500);
    turn_right(100, 250);
}

// otherwise
else{
    forwards(100, 100);
}
```

```
if( 条件1){
    動作1-1
    動作1-2
    …
}
else if( 条件2){
    動作2-1
    動作2-2
    …
}
//それ以外
else{
    動作3-1
    動作3-2
    …
}
```

回避動作手順

```
//Right sensor touched
if( Sensor(IN_2) ){
    backwards(100, 500);
    turn_left(100, 250);
}

//Left sensor touched
else if( Sensor(IN_3) ){
    backwards(100, 500);
    turn_right(100, 250);
}

// otherwise
else{
    forwards(100, 100);
}
```

2番のタッチセンサーが反応したとき，

3番のタッチセンサーが反応したとき，

それ以外のとき，

回避動作の変更

- 各タッチセンサーが反応したときの，回避動作（赤いの部分，青の部分，紫色の部分）を変更する．

```
//Right sensor touched
if( Sensor(IN_2) ){
    backwards(100, 500);
    turn_left(100, 250);
}

//Left sensor touched
else if( Sensor(IN_3) ){
    backwards(100, 500);
    turn_right(100, 250);
}

// otherwise
else{
    forwards(100, 100);
}
```

実習

- 「回避動作の変更」に従って，プログラムを改造してください．
- そして，実際にレゴの動作が変わるか確認してください．

プロジェクト

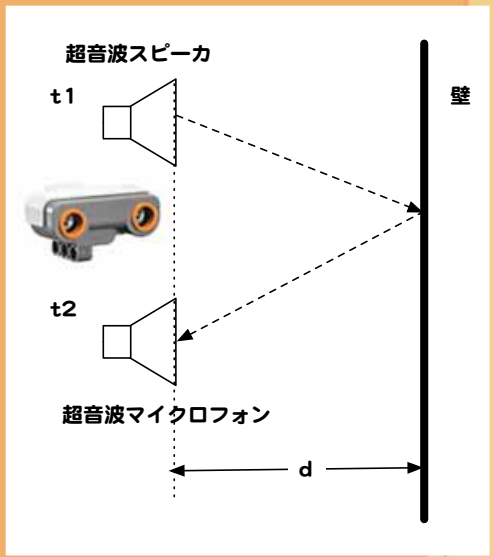
- 触覚をつける
  - タッチセンサを使って，「障害物に衝突した後に回避する動作」をつくる．
- 視覚をつける
  - 超音波センサーを使って，「障害物に衝突する前に回避する動作」をつくる．

視覚をつける

- 超音波センサーを装着する.
- プログラム「Ultrasonic.nxc」を動かす.
- プログラムの改造
  - 回避を開始する障害物までの距離
  - 回避動作の変更

超音波センサの原理

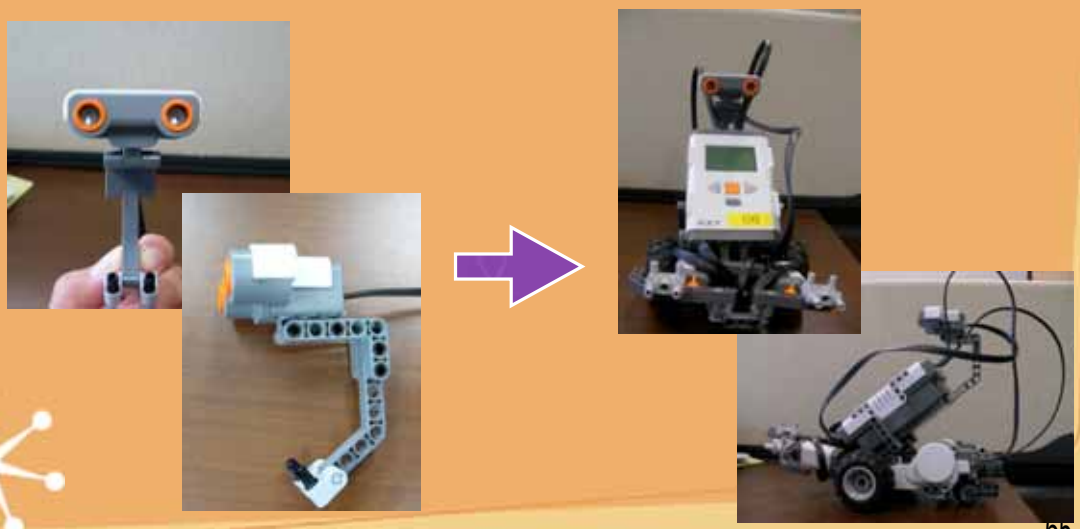
- スピーカから超音波を出し，壁（物体）に反射する超音波をマイクロフォンで検知する.
- その間の時間t(t2-t1)から壁までの距離dが求まる.



d[m]=170*t[s]
ただし，音速340m/sとした

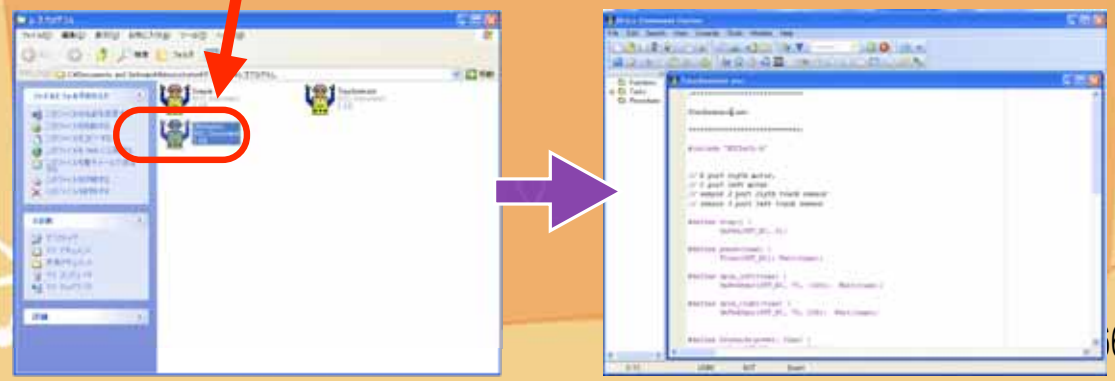
超音波センサの装着

- 超音波センサから出ているコードを４番の入力ポートにさしこむ.



プログラム「Ultrasonic」を開く

「Ultrasonic」のアイコンをクリック

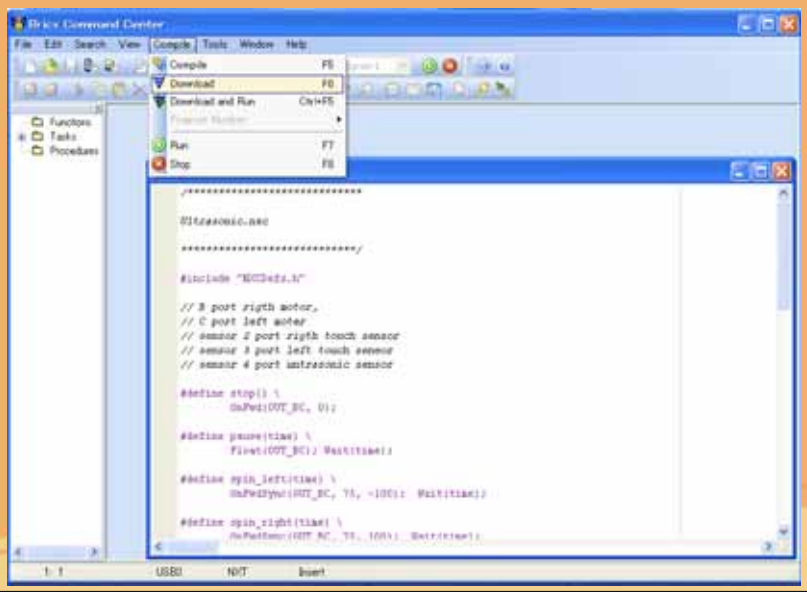


NXTとパソコンとが未接続の場合

- 「２７ページから３１ページ」の手順で，NXTとパソコンを接続し直してください．

NXTに記憶させる

- プログラム「Ultrasonic」をNXTにダウンロード

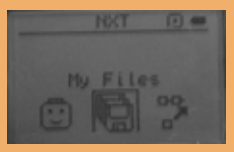


動かしてみよう

「選択」

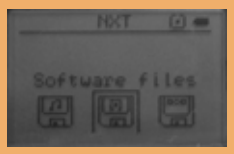
「矢印」

選択を押す



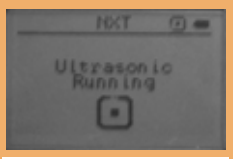
「NXT」起動

選択を押す



矢印で「Ultrasonic」を選び，選択を押す

選択を押す



Ultrasonic実行中

「実行」をとめるには

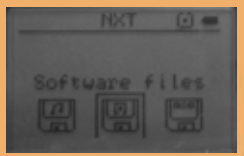
Ultrasonic実行中

キャンセル

キャンセルを1回押す

動作がとまる

キャンセルを2回押す



プログラムの説明

```
#define NEAR 35 //cm

task main()
{
    . . .

      if( SensorUS(IN_4) < NEAR ){
          backwards(100, 1000);
          turn_left(100, 250);

      }else{
          forwards(100, 100);
      }
    . . .
}
```

回避動作を開始するときの障害物までの距離

障害物に近づいた時の回避行動

障害物に近くないとき

プログラムの改造部分

- 回避を開始するときの障害物との距離

```
#define NEAR 35 //cm
```

- 障害物に近づいた時の回避動作

```
if( SensorUS(IN_4) < NEAR ){
    backwards(100, 1000);
    turn_left(100, 250);
```

- 障害物に近くないときの動作

```
}else{
    forwards(100, 100);
}
```

実習

- 「プログラの改造部分」に従って，プログラムを改造してください．
- そして，実際にレゴを動かしてみてください．

まとめ#１

パソコンでプログラミングをし，
レゴ（レゴ車）を操った．

プログラム＝ソフトウェア

レゴ（レゴ車）＝機械（ハードウェア）

「ソフトウェア」で，「ハードウェア」を制御した．

ロボット，車，あらゆる機械がこのようにして，制御されている，

まとめ#2

複数のセンサ（タッチセンサ，超音波センサ）を利用し，「衝突回避」ができるレゴ車を作った．

センサフュージョンによって，賢い機械を作った．

センサフュージョン(Sensor Fusion)＝触覚，視覚，聴覚などのセンサを複数統合的に使って，状況を的確に判断する機械（システム）

質問

？

hideaki@jaist.ac.jp