

ETロボコン公式トレーニング モデリング入門

モデルを使った開発を体験しよう！



この教材について



■ 目的

- この教材は、ETロボコンに参加されるみなさんに、モデル作成に必要となる知識やスキル取得の機会を提供することを目的に作成しています

■ 著作

- この教材はETロボコン実行委員会が作成したものです
- この教材の著作権は、ETロボコン実行委員会に帰属します

■ 使用について

- ETロボコンの参加資格（企業・大学・個人）の範囲内に限り、ご自由に活用していただいてもかまいません

■ 本コースの想定する受講者

- C言語/C++言語、あるいはこれらに類似の言語のプログラミング経験はあるが、モデルを使った開発経験のないエンジニア

■ トレーニングのゴール

- モデルの意義・価値を理解できる
- モデルの読み書きができる
- モデルを使った開発の流れが理解できる

■ トレーニングの進め方

- モデルと部品を使った開発の必要性を学びます
- ライントレーサの開発を例にして、モデルの書き方からモデルをソースコードとして実装する方法までを、段階的に学びます
- ライントレースへの機能の追加や変更などを題材にして、モデルと部品を使った差分開発を、演習を通して実際に体験します

1. なぜ、モデルが必要なのか？
2. ソフトウェアの開発の進め方とモデルの活用
3. モデルと部品を使ってソフトウェアを設計する
4. モデルからソースコードを作る
5. 演習1) ライントレーサに機能を追加する
6. 演習2) ライントレーサの機能を変更する
7. 演習3) これまでの部品で新たな機能を実現する
8. 参考文献、その他の情報
9. 演習のまとめ

- EV3を使ったシステムを開発するために使用する開発環境を構築しましょう
- モデリングに使うツールをインストールして、参加者向けライセンスを設定しましょう
- 演習用データを入手しましょう

0. トレーニングの準備

0-1. 開発環境を構築する

EV3



- 使用するロボットキットに必要な開発環境を入手し、インストールしておきましょう
- Mindstorms EV3を使う場合は次の構築手順に従います
 - ETロボコン EV3開発環境構築ガイド
 - ◆ <https://github.com/ETrobocon/etroboEV3/wiki>
 - 開発環境構築の課題と対処
 - ◆ https://github.com/ETrobocon/etroboEV3/wiki/problem_and_coping
- Windows利用者は、Cygwinが使えることを確認しておきましょう
 - Cygwinのシェルは起動しますか
 - Unixの基本コマンド（cd, ls, makeなど）は動作しますか

0-2. モデリングツールを用意する



- **astah* Professionalの参加者向けライセンスが提供されるので、モデリング演習にはこのツールを使います**
 - 演習進行の円滑を意図したもので、提出するモデルを作成するツールを指定するものではないことに注意しましょう
- **astah* Professionalを入手します**
 - 参加者向けの入手方法に従ってツールをダウンロードします
 - ◆ astah* SysMLや、astah* GSNもダウンロードできます
- **astah* Professionalをインストールします**
 - インストール手順に沿ってインストールします
 - ◆ astah* を使うには、JRE (Java Runtime Environment) が必要です
 - ◆ Windows PCだけでなく、MacやLinuxでも利用できます
- **astah*製品群向けの参加者向けライセンスを入手します**
 - 参加者向けの入手方法に従ってライセンスを入手します
 - 提供される手順に従って、astah* Professionalを起動し、ライセンスを設定します
 - 同じライセンスで、astah* SysMLや、astah* GSNも利用できます

astah*製品群のダウンロードやライセンス設定に関する詳細は、(株)チェンジビジョンから提供されます

0-3. テキストエディタを用意する



- プログラムの作成には、プログラミング用のテキストエディタや統合環境（IDE）を使いましょう
 - メモ帳で済ませようなんて、**ダメ、ダメ、絶対**
 - このトレーニングでは、テキストエディタとコマンドラインツールを使ってプログラムを作る方法で演習を進めます
 - ◆ 他の環境やツールを使いたい場合も、トレーニング中は同じやり方で進めましょう
- テキストエディタの使い方に慣れておきましょう
 - ファイルの作成、削除、保存、別名での保存、扱えるファイル拡張子の確認
 - テキストの入力、削除、コピーやペースト、検索、置換
- Windows用のテキストエディタの例
 - Notepad++、TeraPad、サクラエディタ、Atom、Brackets、Emacs、Vim、Sublime Text、Visual Studio Codeなど
- Mac用のテキストエディタの例
 - CotEditor、mi、TextMate、Atom、Brackets、Emacs、Vim、Sublime Text、Visual Studio Codeなど

0-4. 配布パッケージの説明（1）

■ 配布パッケージのファイル構成

- 次のようなファイルを配布しています

配布ファイル名	含まれるもの
chap-00-04.zip	0章から4章のテキスト、4章のサンプルモデルとサンプルコード
chap-05-07.zip	5章から7章のテキスト（解答例は伏せてあります）
chap-05-ans.zip	5章の解答例つきテキスト、演習後のモデルとコードの例
chap-06-ans.zip	6章の解答例つきテキスト、演習後のモデルとコードの例
chap-07-ans.zip	7章の解答例つきテキスト、演習後のモデルとコードの例
chap-08-apx.zip	8章、9章、補足資料のテキスト

配布パッケージのファイル構成

0-5. 演習の準備（1）

■ 0章から4章で使う配布ファイル

- ファイル名：chap-00-04.zip
- テキストと、演習で使うサンプルコードとサンプルモデルが含まれています

■ 圧縮ファイルchap-00-04.zipを展開します

- 展開したディレクトリの中身を確認しましょう

```

chap-00-04
├── pdf
│   ├── chap-00.pdf
│   ├── chap-01.pdf
│   ├── chap-02.pdf
│   ├── chap-03.pdf
│   └── chap-04.pdf
├── asta
│   └── etrobo_tr.asta
├── sample_code
│   ├── Makefile
│   └── etrobo_tr/*

```

テキストのディレクトリ

0章のテキスト

1章のテキスト

2章のテキスト

3章のテキスト

4章のテキスト

サンプルモデルのディレクトリ

3、4章で使うサンプルモデル

サンプルコードのディレクトリ

ワークスペースに必要なMakefile

3、4章で使うサンプルモデル

0-5. 演習の準備（2）

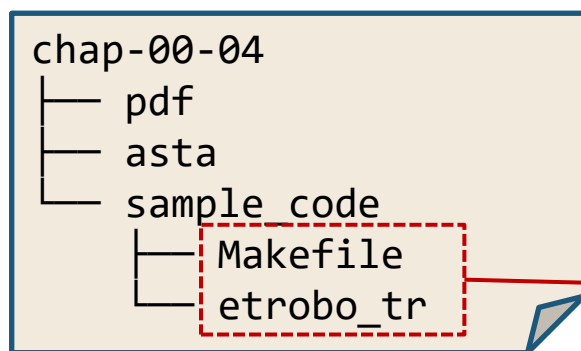
EV3

ET
ROBOT
CONTEST

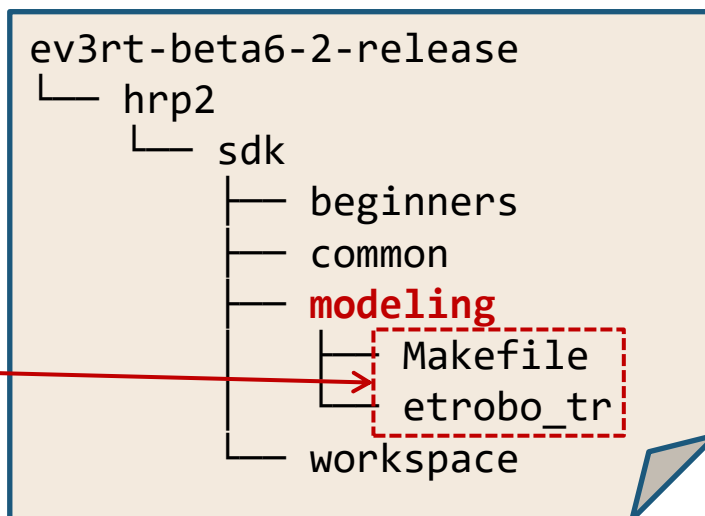


■ EV3RT環境にワークスペースを作成します

- EV3RTをインストールしたディレクトリの中に「hrp2¥sdk」の中に「**modeling**」ディレクトリを作成します
- 「sample_code」ディレクトリの中に含まれる全てのディレクトリとファイルを、いま作成した「modeling」ディレクトリの中へコピーします
- コピー先ディレクトリの内容を確認します



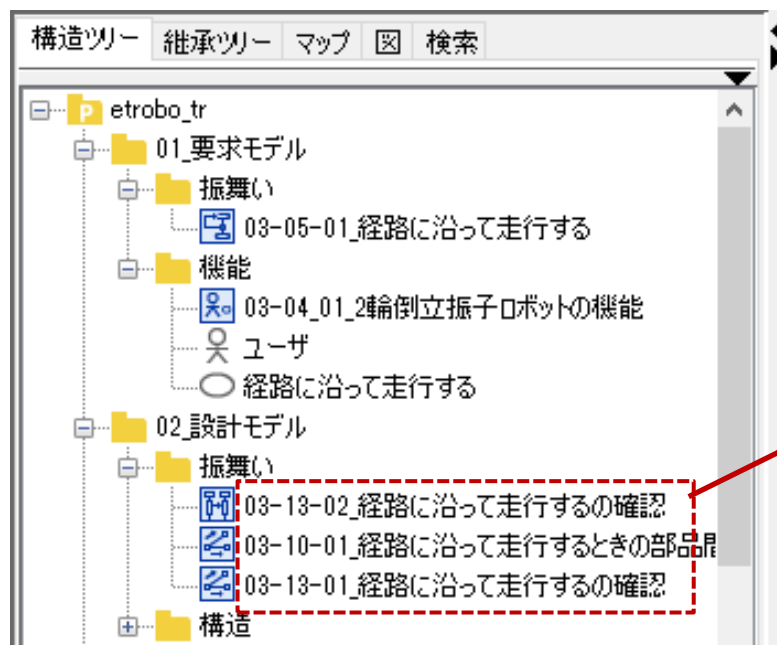
chap-00-04.zipを展開した様子（一部省略）



0-5. 演習の準備（3）

■ 演習用モデルを格納するディレクトリを作成します

- chap-00-04.zipを展開したディレクトリの中に「asta」ディレクトリがあることを確認します
- astah*を起動して「asta」ディレクトリに含まれる「etrobo_tr.asta」が読めるか確認しましょう



サンプルモデル図の各図には、
どのページの図が分かるよう、
図の名前に章・節番号をつけて
あります
演習のときは、サンプルモデル
図を参照しながら進めましょう

0-6. 環境の使い方に慣れておきましょう



- Cygwin環境（Macではターミナル）の操作方法
 - ファイルやディレクトリの配置の理解
 - Unixのシェルとコマンドの使い方
- テキストエディタの操作方法
 - ファイルを作る、開く、保存する、閉じる
 - 文章やコードを入力する、削除する、コピー&ペーストする
 - 文字列を検索、置換する
- モデリングツールの操作方法
 - プロジェクトファイルの作成、保存
 - モデル図を追加、複製、削除する
 - モデル図の要素の追加、移動、修正、削除
- モデリング言語の基本
 - UMLのよく使う図や構成要素の名前や役割

0-7. プログラムのビルドと実行（1）

EV3

ET
ROBOT
CONTEST



- EV3のプログラムをビルド・実行方法する方法は2つあります
 - 動的ローディング形式（ローダを使って指定したプログラムを起動する方式）
 - ◆ このトレーニングでは「動的ローディング形式」を使用します
 - スタンドアローン形式（プログラム自身がブートイメージとなる方式）

- 動的ローディング形式のビルド手順
 - シェルを起動してアプリケーションのソースコードがあるワークスペースへ移動します
 - ◆ （インストールディレクトリ） /hrp2/sdk/modeling
 - ◆ ワークスペース中のアプリケーションごとのディレクトリ中にソースコードがあります
 - 次のコマンドを実行します
 - ◆ `make app=アプリケーションディレクトリ名`
 - ビルドに成功すると、modelingディレクトリ中に「app」というファイル名のロードモジュールができます

実際のビルドと転送は、演習の中でやってみましょう

0-7. プログラムのビルドと実行 (2)

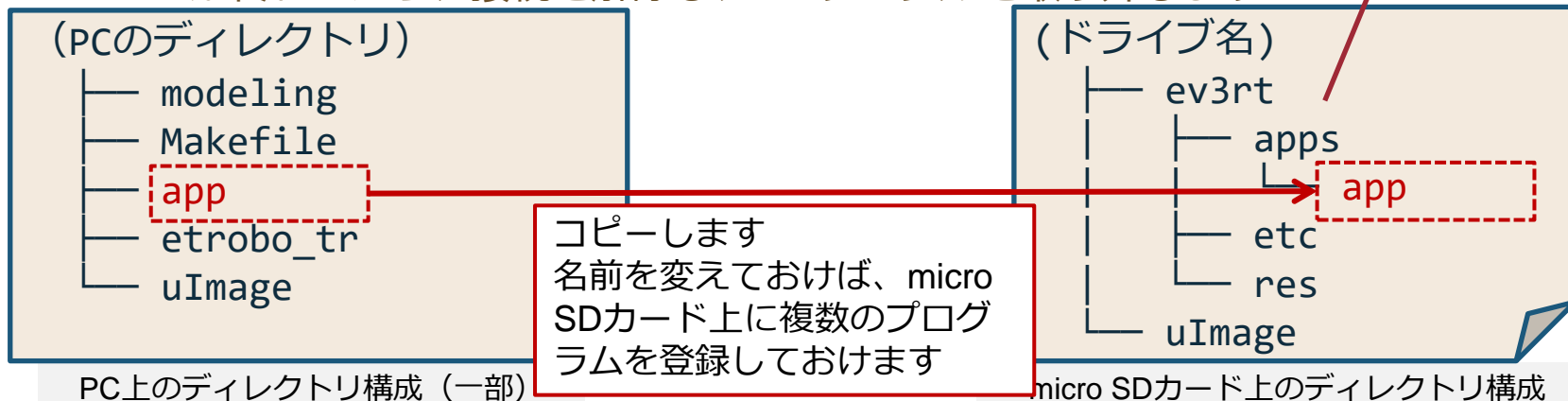
EV3

ET
ROBOT
CONTEST



- micro SDカードにアプリケーションローダが入っていない場合
 - 予めアプリケーションローダを micro SDカードにコピーしておきます
 - ◆ インストールディレクトリのsdcardディレクトリに入っているuImageがローダです
- SDカードをEV3に差し込んで、EV3本体の電源を入れると、アプリケーションローダが起動します
 - ロードのGUIメニューが表示されたら起動は成功です
- 作成したアプリケーションをappsディレクトリにコピーします
 - USBケーブルでPCとEV3本体を接続します
 - SDカードが認識されたら「app」をコピーします
 - コピーが終わったら、接続を解除し、USBケーブルを取り外します

appsディレクトリ
アプリケーションを
登録しておく場所



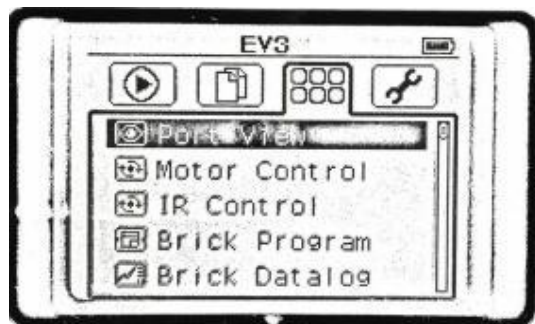
0-8. ジャイロのオフセット値の調べ方

EV3

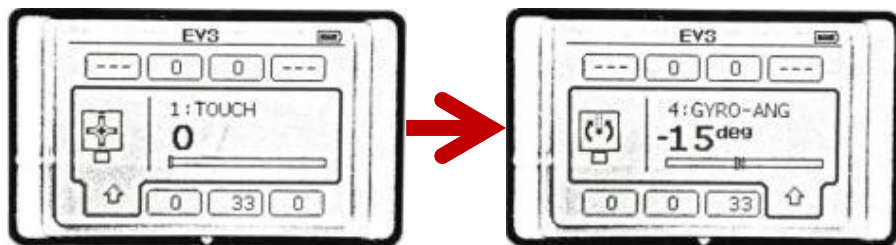
ET
ROBOT
CONTEST



1. MicroSDカードがスロットに刺さっていない状態で電源を投入します
2. 「Port View」を選択し、中央ボタンを押します

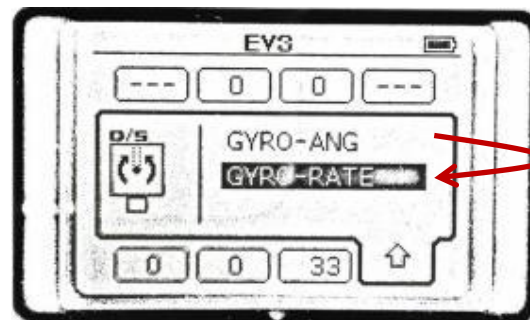


3. 左右ボタンでタブを移動し、ジャイロセンサのポートを選択します

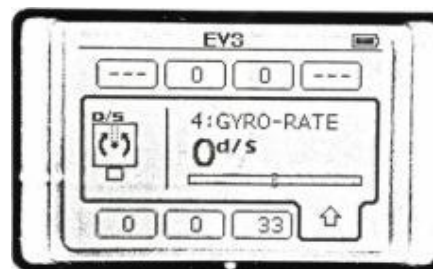


ジャイロセンサには無視できない個体差があります
EV3では、角速度モードに変更した時に自動的に
キャリブレーションされます

4. 中央ボタンを押すと左のメニューが出るので、「GYRO-RATE」を選択し、中央ボタンを押します



5. 「GYRO-RATE」で表示される値がジャイロオフセット値となります



自動的にキャリブレーションされるので、静止した時の値が0になります