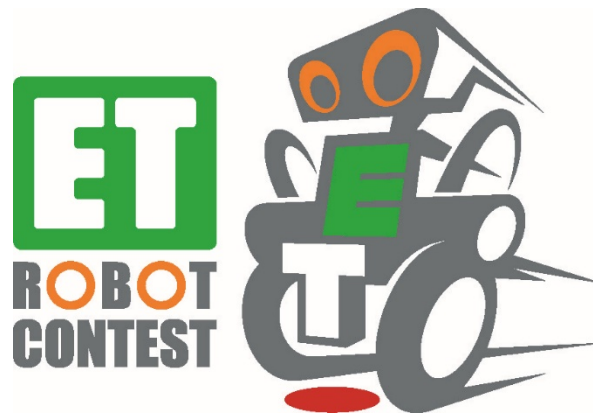


## EV3 開発環境の構築

---

ここで、EV3開発環境をちゃちゃっと構築して、これから始まる”技術教育”に挑もうではないか！



# 本日のルール



- グループ内 (参加者同士) で助け合うこと
  - グループ内で解決できなければ、実行委員にサポートをお願いします
- グループ全員で”(本日の) ゴール”しましょう
- このスライドの見方
  - EV3 開発環境構築ガイド (TOPPERS/EV3RT) を主とします  
[http://dev.toppers.jp/trac\\_user/ev3pf/wiki/WhatsEV3RT](http://dev.toppers.jp/trac_user/ev3pf/wiki/WhatsEV3RT)
  - スライドでは各工程のポイントとなることを紹介します



- 本日は動的ローディング形式でビルドします
- 1. ビルドは、“workspace” フォルダで行います

# 本日のゴール (おさらい)



- グループ全員で以下のことを達成すること！
  - TOPPERS/EV3RT 開発環境を構築する
  - サンプルプログラムをビルドし、正常にコンパイルできることを確認する
  - コンパイル済みアプリケーションを走行体上で実行し、正常に動作することを確認する
  
- 参加者同士”なかよし”になること！



# EV3 開発環境の構築 – アジェンダ



1. 持ち物確認
2. 開発環境 (クロスコンパイラー, ツール) のインストール
3. 開発プラットフォーム EV3RT のダウンロード & インストール
4. サンプルプログラムのダウンロード
5. サンプルアプリケーションのビルドと実行
6. サンプルコースの試走

- まずは、忘れ物がないか確認しましょう
- 

# 1. 持ち物確認

# ここがポイント!!

## ■ 開発PC

- ✓ Windows / Mac

## ■ EV3走行体（+本体とパーツをつなげるケーブル）

- ✓ EV3way / HackEV

走行体は組立図どおりに組み立てたもの

## ■ microSDカード (+ SDカードアダプター)

- ✓ SDHC 4GB 以上

## ■ インターネットに接続できる機器

- ✓ もしくは事前にダウンロード

## ■ サンプルコース

- ✓ 持参できる方のみ

---

## 2. 開発環境 のインストール (Windows編)

- [http://dev.toppers.jp/trac\\_user/ev3pf/wiki/DevEnvWin](http://dev.toppers.jp/trac_user/ev3pf/wiki/DevEnvWin)
  - 開発環境のインストール（Windows + Cygwin編）



The screenshot shows the Toppers Wiki page for 'DevEnvWin'. The page title is '開発環境のインストール (Windows + Cygwin編)'. The content describes how to set up a development environment on Windows using Cygwin. It includes a list of instructions for downloading and installing Cygwin, and a small image of the Cygwin Setup window.

**開発環境のインストール (Windows + Cygwin編)**

Windows + Cygwinで開発する場合、以下の手順で開発環境を構築することができます。

① Cygwinのインストーラをダウンロードします

- ⇒ Cygwinのサイトからインストーラをダウンロードすることができます
- 64-bit版Cygwinのインストーラ(⇒ setup-x86\_64.exe)をオススメします
- 32-bit版Cygwin(⇒ setup-x86.exe)でも動作できる報告があります
- Cygwinのインストーラ(⇒ setup-x86\_64.exe又は setup-x86.exe)を起動します





手順番号は、参照ページの番号と紐づけられています。

👉 ここがポイント!! (Windows) -1/2

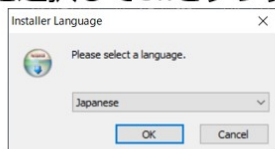


## ■ 手順⑥

- ダウンロードサイト → “<http://ftp.jaist.ac.jp>”

## ■ 手順⑩

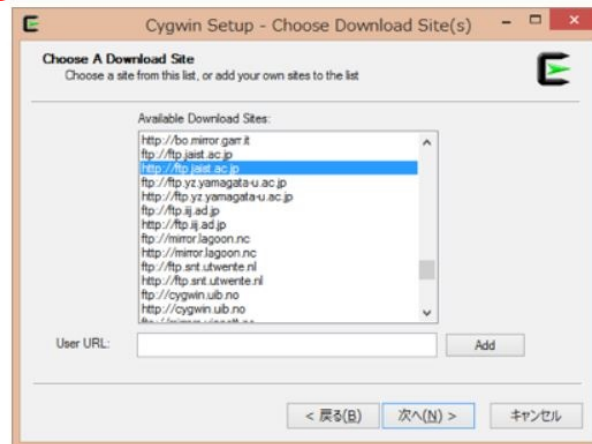
- 下記URLからgcc-arm-none-eabi-6-2017-q1-update-win32.exeをダウンロードします
  - [https://developer.arm.com/-/media/Files/downloads/gnu-rm/6\\_1-2017q1/gcc-arm-none-eabi-6-2017-q1-update-win32.exe](https://developer.arm.com/-/media/Files/downloads/gnu-rm/6_1-2017q1/gcc-arm-none-eabi-6-2017-q1-update-win32.exe)
- GCC ARMのインストーラを起動します
- 「Japanese」を選択してOKをクリックします



ETロボコン技術教育資料 / ETロボコン実行委員会

ETロボコン2019 10

## ⑥ パッケージをダウンロードするサイトを選択します



- 任意のサイトを選択しても構いませんが、日本国内のサイト（末尾がjp）をオススメします

[http://dev.toppers.jp/trac\\_user/ev3pf/wiki/DevEnvWin](http://dev.toppers.jp/trac_user/ev3pf/wiki/DevEnvWin)



# ここがポイント!! (Windows) -1/2

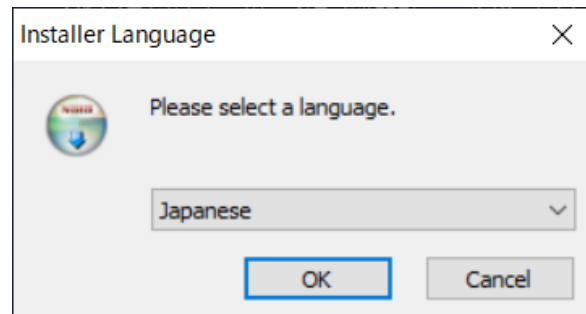


## ■ 手順⑥

- ダウンロードサイト → “<http://ftp.jaist.ac.jp>”

## ■ 手順⑩

- 下記URLからgcc-arm-none-eabi-6-2017-q1-update-win32.exeをダウンロードします
  - [https://developer.arm.com/-/media/Files/downloads/gnu-rm/6\\_1-2017q1/gcc-arm-none-eabi-6-2017-q1-update-win32.exe](https://developer.arm.com/-/media/Files/downloads/gnu-rm/6_1-2017q1/gcc-arm-none-eabi-6-2017-q1-update-win32.exe)
- GCC ARMのインストーラを起動します
- 「Japanese」を選択してOKをクリックします



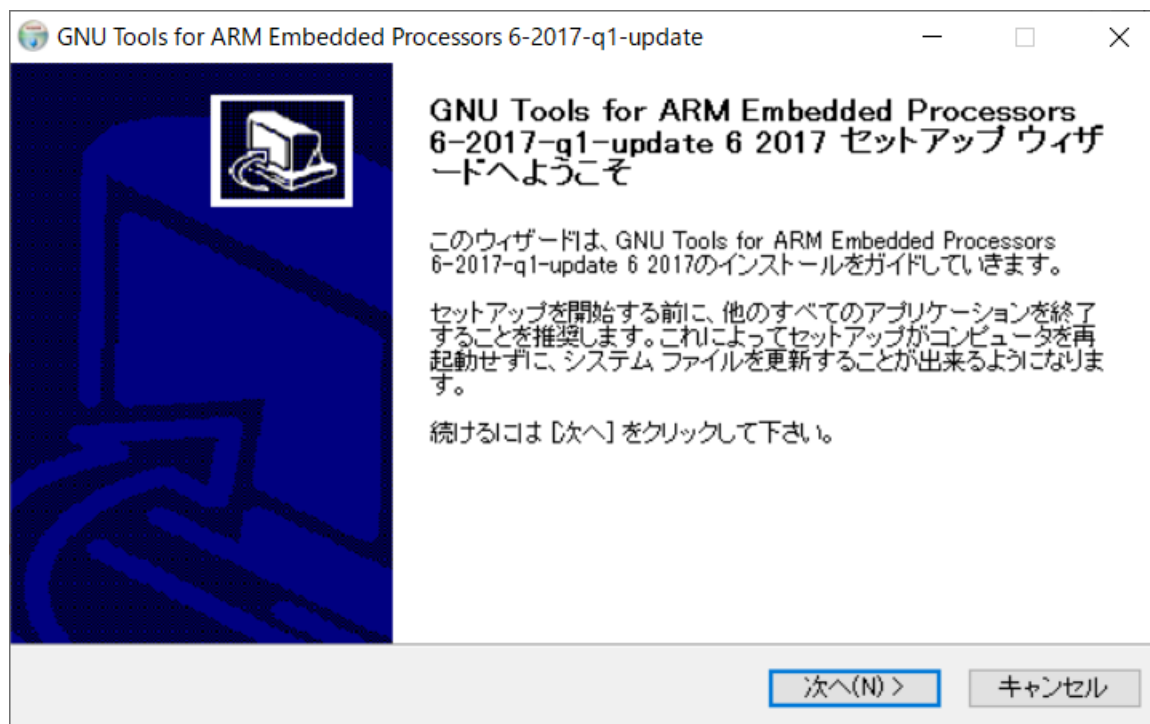


# ここがポイント!! (Windows) -1/2



## ■ 手順⑩

- 「次へ」をクリックします



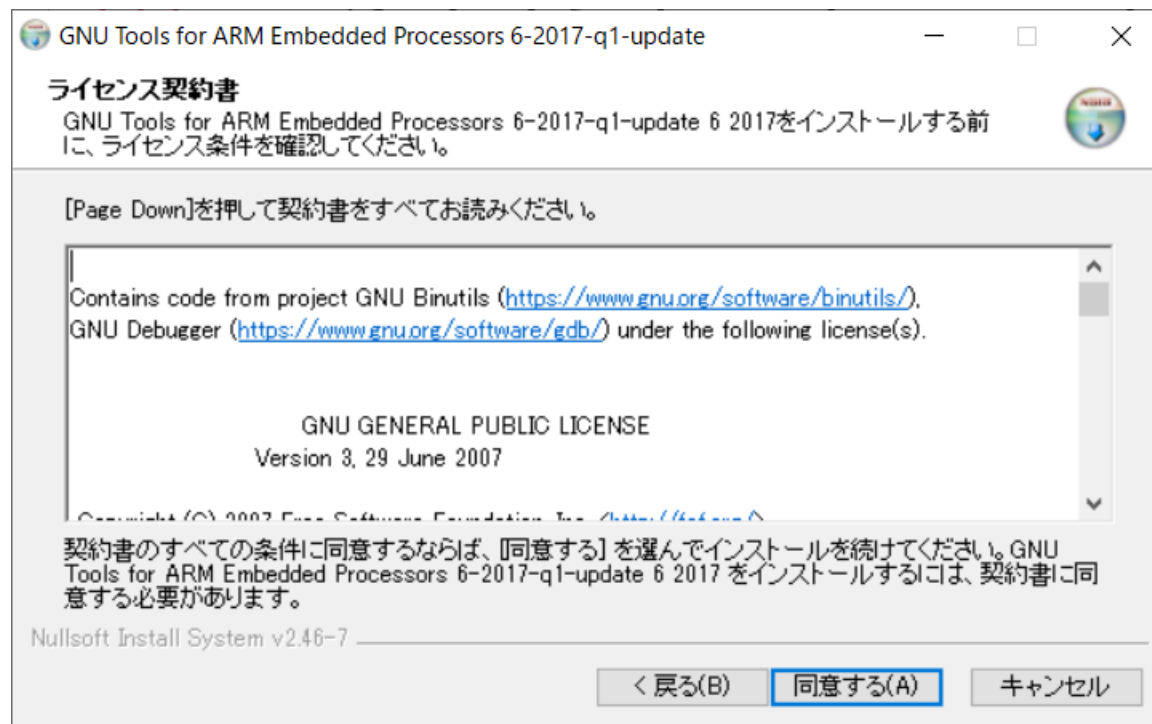


# ここがポイント!! (Windows) -1/2



## ■ 手順⑪

- 「同意する」をクリックします



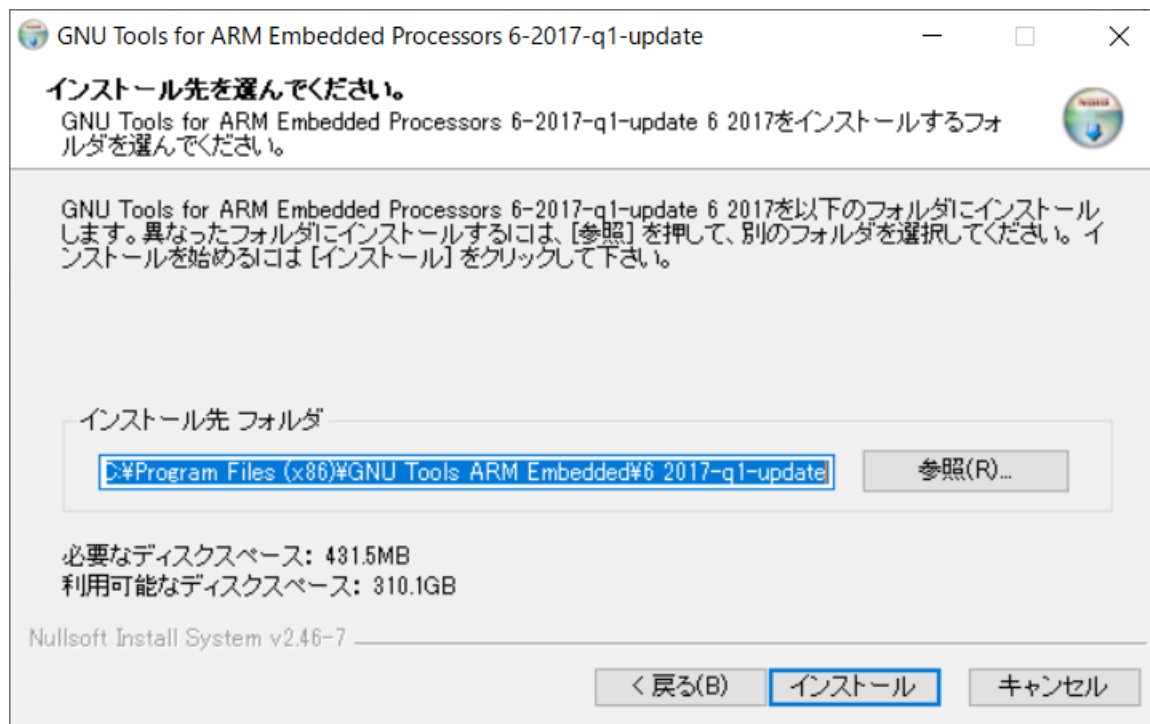


# ここがポイント!! (Windows) -1/2



## ■ 手順⑫

- 「インストール」 をクリックします



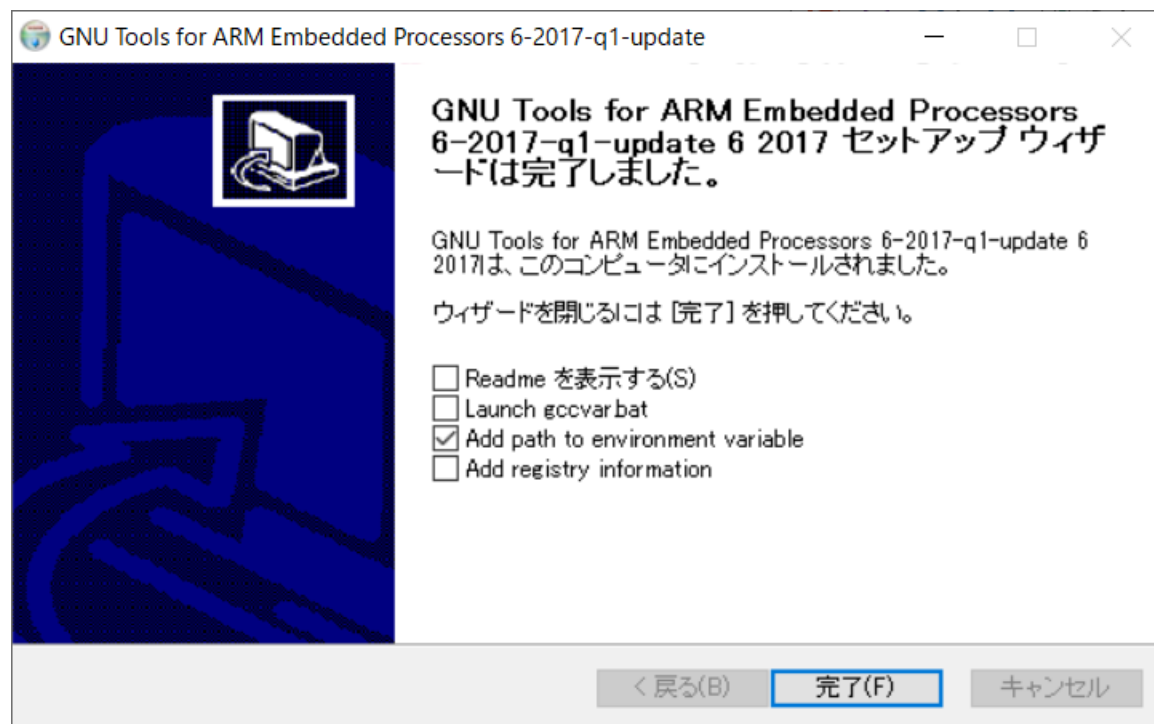


# ここがポイント!! (Windows) -1/2



## ■ 手順⑭

- 上記の画面が表示されたら、インストールは完了です
- **必ず「Add path to environment variable」だけ選択して「完了」をクリックします**





# ここがポイント!! (Windows) -1/2



## ■ 手順⑭

- Cygwin を起動して以下のように入力し、  
「gcc version 6.3.1」と表示されていること

```
$ arm-none-eabi-gcc -v
Using built-in specs.
COLLECT_GCC=C:\Program Files (x86)\GNU Tools ARM Embedded\6 2017-q1-update\bin\arm-none-eabi-gcc.exe
COLLECT_LTO_WRAPPER=c:/program files (x86)/gnu tools arm embedded/6 2017-q1-update/bin/.../lib/gcc/arm-none-eabi/6.3.1/lto-wrapper.exe
Target: arm-none-eabi
Configured with: /tmp/jenkins-GCC-6-buildandreg-104_20170216_1487268972/src/gcc/configure --build=x86_64-linux-gnu --host=i686-w64-mingw32 --target=arm-none-eabi --prefix=/tmp/jenkins-GCC-6-buildandreg-104_20170216_1487268972/install-mingw --libexecdir=/tmp/jenkins-GCC-6-buildandreg-104_20170216_1487268972/install-mingw/lib --infodir=/tmp/jenkins-GCC-6-buildandreg-104_20170216_1487268972/install-mingw/share/doc/gcc-arm-none-eabi/info --mandir=/tmp/jenkins-GCC-6-buildandreg-104_20170216_1487268972/install-mingw/share/doc/gcc-arm-none-eabi/man --htmldir=/tmp/jenkins-GCC-6-buildandreg-104_20170216_1487268972/install-mingw/share/doc/gcc-arm-none-eabi/html --pdfdir=/tmp/jenkins-GCC-6-buildandreg-104_20170216_1487268972/install-mingw/share/doc/gcc-arm-none-eabi/pdf --enable-languages=c,c++ --enable-mingw-wildcard --disable-decimal-float --disable-libffi --disable-libgomp --disable-libmudflap --disable-libquadmath --disable-libssp --disable-libstdc++-pch --disable-nls --disable-shared --disable-threads --disable-tls --with-gnu-as --with-gnu-ld --with-headers=yes --with-newlib --with-python-dir=share/gcc-arm-none-eabi --with-sysroot=/tmp/jenkins-GCC-6-buildandreg-104_20170216_1487268972/install-mingw/arm-none-eabi --with-libiconv-prefix=/tmp/jenkins-GCC-6-buildandreg-104_20170216_1487268972/build-mingw/host-libs/usr --with-gmp=/tmp/jenkins-GCC-6-buildandreg-104_20170216_1487268972/build-mingw/host-libs/usr --with-mpfr=/tmp/jenkins-GCC-6-buildandreg-104_20170216_1487268972/build-mingw/host-libs/usr --with-mpc=/tmp/jenkins-GCC-6-buildandreg-104_20170216_1487268972/build-mingw/host-libs/usr --with-isl=/tmp/jenkins-GCC-6-buildandreg-104_20170216_1487268972/build-mingw/host-libs/usr --with-libelf=/tmp/jenkins-GCC-6-buildandreg-104_20170216_1487268972/build-mingw/host-libs/usr --with-host-libstdc++=' -static -libgcc -Wl,-Bstatic,-lstdc++,-Bdynamic -lm' --with-pkgversion='GNU Tools for ARM Embedded Processors 6-2017-q1-update' --with-multilib-list=rmprofile
Thread model: single
gcc version 6.3.1 20170215 (release) [ARM/embedded-6-branch revision 245512] (GNU Tools for ARM Embedded Processors 6-2017-q1-update)
```



# ここがポイント!! (Windows) -1/2



## ■ 手順⑮

- 「Mkimage.tar.gz」 を下記 に置きます  
32bit 版 OS ・ ・ ・ C:¥cygwin¥tmp  
64bit 版 OS ・ ・ ・ C:¥cygwin64¥tmp
- Cygwin を起動し、Mkimage.tar.gzが置いてある「tmp」フォルダに移動します

```
$ cd /tmp
```





# ここがポイント!! (Windows) -2/2



手順⑮まで終わったら...

## ■ ダウンロードパッケージの確認

- Cygwinを起動して、以下のコマンドを実行します

```
$ cygcheck -c gcc-core make diffutils perl
```

- パッケージの情報が出力されるので、バージョンが  
ていること、StatusがOKであることを確認します

### Cygwin Package Information

Package	Version	Status
diffutils	3.3-3	OK
gcc-core	5.3.0-5	OK
make	4.1-1	OK
perl	5.22.1-2	OK

コマンドの実行結果の例です。  
手順⑦に書かれているバージョンより  
新しいことを確認します。

---

## 2. 開発環境 のインストール (Mac編)

# クロス環境のインストール



- [http://dev.toppers.jp/trac\\_user/ev3pf/wiki/DevEnvMac](http://dev.toppers.jp/trac_user/ev3pf/wiki/DevEnvMac)

**TOPPERS**  
Toyohashi Open Platform  
for Embedded Real-time Systems

Search

Login | Preferences | Help/Guide | About Trac

Wiki | View Tickets | Search

Start Page | Index | History

wiki: DevEnvMac

## OS Xでの環境構築方法

### 前提となる環境

予め以下のソフトウェアがインストールされているものとします。

- gcc
  - Xcodeをインストール(App Store から入手可能)or ⇒ [macport](#) でインストール

### mkimageのインストール

- ⇒ [ここ](#) からバイナリファイルをダウンロードし、パスの通ったディレクトリに置く

```
cp mkimage /usr/local/bin/
```

- ソースコードからインストールする場合は、[こちら](#)を参照してください。

### GNUツールチェーンのインストール

- ⇒ Mac用バイナリが、⇒ [GCC ARMのサイト](#) からダウンロードすることができます



# ここがポイント!! (Mac) - 1/3



## ■ 前提となる環境

- gcc がインストールされているか確認します

```
$ make -v
```

## ■ mkimage のインストール

- “/usr/local/bin” にパスが通っているか確認します

```
$ echo $PASH
```

表示された文字列に「/usr/local/bin」が含まれていることを確認します

- 「mkimage」ファイルを「/usr/local/bin」に置きます

```
$ cd ~/Downloads
```

“mkimage” ファイルをダウンロードしたディレクトリ

```
$ cp mkimage /usr/local/bin/
```



# ここがポイント!! (Mac) - 2/3



## ■ GNUツールチェーンのインストール

- 下記URLからダウンロードします
  - [https://developer.arm.com/-/media/Files/downloads/gnu-rm/6\\_1-2017q1/gcc-arm-none-eabi-6-2017-q1-update-mac.tar.bz2](https://developer.arm.com/-/media/Files/downloads/gnu-rm/6_1-2017q1/gcc-arm-none-eabi-6-2017-q1-update-mac.tar.bz2)
- Finder でダブルクリックして解凍します
- 特にこだわりがなければ、「/usr/local」に置きましょう

```
$ mv gcc-arm-none-eabi-6-2017-q1-update /usr/local/
```

- パスを通します。bash を使っていない方は、ツールに応じた設定を行ってください

実際は1行で記述します

```
$ echo "export PATH=/usr/local/gcc-arm-none-eabi-6-2017-q1-update/bin:¥$PATH" >> ~/.bash_profile
```

- 以下のコマンドで、設定を反映します

```
$ source ~/.bash_profile
```



## ここがポイント!! (Mac) - 3/3



### ■ EV3RT のコンフィギュレータ (cfg) のインストール

- 先に「3. EV3RT のインストール」を実施してから、cfgをインストールします  
(後ほど手順が出てくるので、ここでは省略)

---

### 3. EV3RT のインストール

- [http://dev.toppers.jp/trac\\_user/ev3pf/wiki/DevEnv](http://dev.toppers.jp/trac_user/ev3pf/wiki/DevEnv)
  - 開発環境構築のマニュアル



wiki: DevEnv

[Login](#) | [Preferences](#) | [Help/Guide](#) | [About Trac](#)

[Wiki](#)

[View Tickets](#)

[Search](#)

[Start Page](#) | [Index](#) | [History](#)

## 開発環境構築のマニュアル

### 開発プラットフォームEV3RTのダウンロード&インストール

[ダウンロードページ](#)からEV3RTのパッケージをダウンロードし、以下の手順に従ってインストールしてください。

- ① zipファイルを解凍し、プラットフォームのパッケージ (hrp2.tar.xz) を作業フォルダに移動します。
  - 作業フォルダの場所は基本的に自由ですが、パスにスペースや全角文字が含まれている場所は避けてください。

- ② ターミナルを開いて作業フォルダに移動します (もし、Windows環境でCygwinをインストールしていない方は、先に「開発環境 (クロスコンパイラ, ツール) のインストール」を実施してください)。

- ③ 下記のコマンドでプラットフォームの簡易パッケージを解凍します

- 注意: 解凍ソフトを使用して解凍すると、アーカイブに含まれるシンボリックリンクが壊れてしまうことがありますので、以下のように tar コマンドを使用してください。

```
$ tar xvf hrp2.tar.xz
```

作業フォルダに、hrp2 フォルダができれば、プラットフォームのインストールが完了です。

#### Table of Contents

[TOPPERS/EV3RT とは?](#)

[ダウンロード](#)

[開発環境構築のマニュアル](#)

[サンプルアプリケーションのビルドと実行](#)

[ユーザマニュアル \(アプリケーション開発者向け\)](#)

[EV3RT C APIリファレンス](#)

[EV3RT C++ APIリファレンス](#)

[FAQ](#)

[お問い合わせ](#)



# ここがポイント!!

## ■ EV3RT のパッケージ (ev3rt-beta7-2-release.zip)

### ダウンロード

#### パッケージ

リリース	公開範囲	最終更新日	ファイル
β7-2	一般	2018年08月2日	<a href="#">🔗 ev3rt-beta7-2-release.zip</a>
β7-1	一般	2018年04月17日	<a href="#">🔗 ev3rt-beta7-1-release.zip</a>
β7	一般	2017年04月20日	<a href="#">🔗 ev3rt-beta7-release.zip</a>

[http://dev.toppers.jp/trac\\_user/ev3pf/wiki/Download](http://dev.toppers.jp/trac_user/ev3pf/wiki/Download)



# ここがポイント!! (Windows)



## ■ 手順①

- 「ev3rt-beta7-2-release.zip」 をダブルクリックで解凍
  - 「ev3rt-bata7-2-release」 というフォルダができます
- 上記フォルダを下記に移動します
  - 32bit 版 OS . . . C:\cygwin
  - 64bit 版 OS . . . C:\cygwin64
- フォルダの名前が長いので、リネームし「作業フォルダ」とします
  - 「ev3rt-bata7-2-release」 → 「ev3rt」



# ここがポイント!! (Windows)



## ■ 手順①

- Cygwin を起動して、以下のコマンドを実行して「作業フォルダ」に移動します

```
$ cd /ev3rt
```

\$ は表示されているので、cd /ev3rt を入力します。

- 管理者権限が付与されていないログインユーザーでCygwinを起動する際には、Cygwin Terminalアイコンを右クリックして”管理者として実行”を選択して起動してください。
- 以降のCygwinを起動する際も同様です。

## ■ 手順②

- 以下のコマンドを実行して、「hrp2.tar.xz」を解凍します

```
$ tar xvf hrp2.tar.xz
```



# ここがポイント!! (Mac) - 1/3



## ■ 手順①

- 「ev3rt-beta7-2-release.zip」 をダブルクリックで解凍
  - 「ev3rt-bata7-2-release」 というフォルダができます
- 上記フォルダを「ユーザー名のフォルダ」の中にドラッグ & ドロップで移動します
  - Finder の「よく使う項目」一覧の中の「家のアイコン」がついているところです
- フォルダの名前が長いので、リネームし「作業フォルダ」とします
  - 「ev3rt-bata7-2-release」 → 「ev3rt」



# ここがポイント!! (Mac) - 2/3



## ■ 手順②

- ターミナルは、[アプリケーション] > [ユーティリティ] の中にあります
- ターミナルで以下のコマンドを実行し、「作業フォルダ」に移動します

```
$ cd ~/ev3rt
```

## ■ 手順③

- 以下のコマンドを実行して、「hrp2.tar.xz」を解凍します

```
$ tar xvf hrp2.tar.xz
```



# ここがポイント!! (Mac) - 3/3



## ■ EV3RT のコンフィギュレータ (cfg) のインストール

- [https://dev.toppers.jp/trac\\_user/ev3pf/wiki/DevEnvMac](https://dev.toppers.jp/trac_user/ev3pf/wiki/DevEnvMac) に戻り「EV3RT のコンフィギュレータ (cfg) のインストール」を実施します
- ダウンロードした「cfg-osx-static-1\_9\_4.tar.gz」を Finder でダブルクリックで解凍します
  - 拡張子のない「cfg」というファイルができます
- 「cfg」ファイルを「~/ev3rt/hrp2/cfg/cfg/」に、Finder でドラッグ & ドロップで移動します
  - 「cfg.exe」や「cfg.cpp」などがあるフォルダです

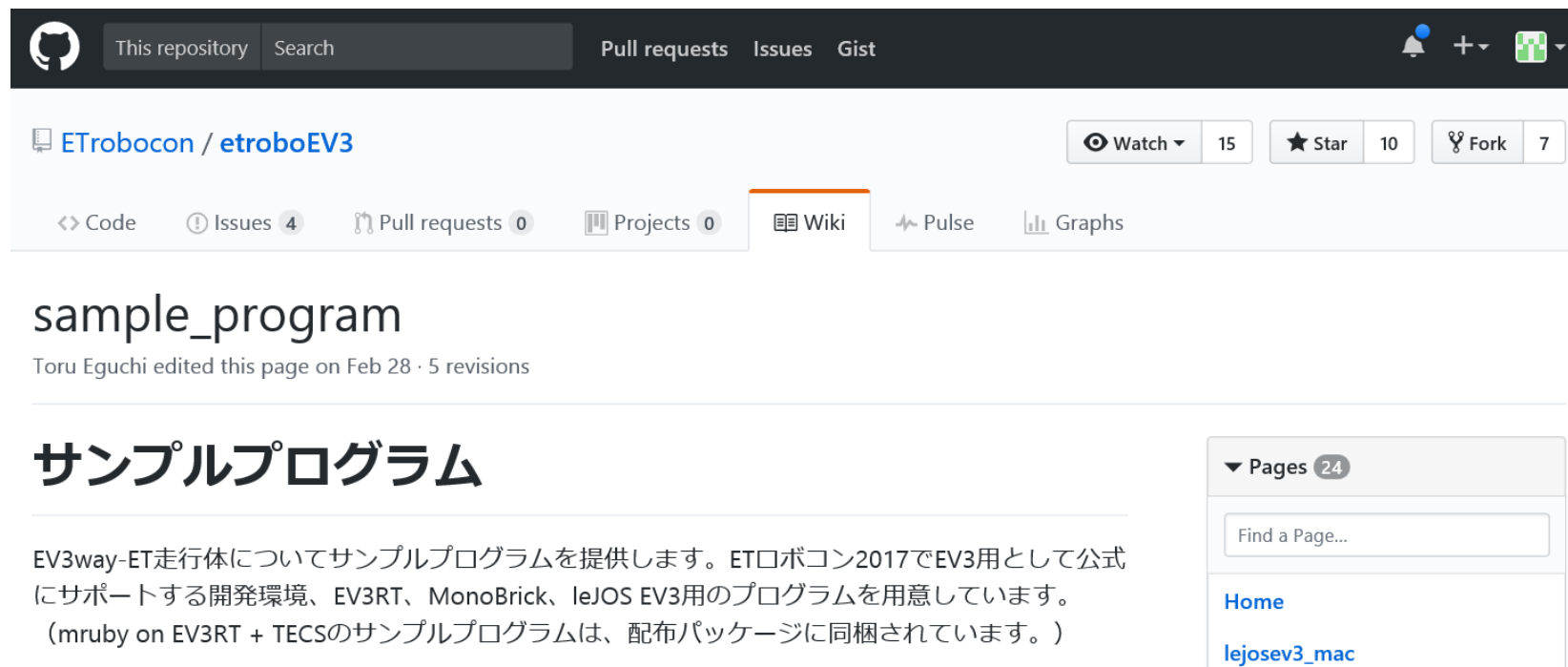
- この章で紹介しているサンプルは、技術教育用ではないので、注意すること

---

## 4. サンプルプログラムのダウンロード

## ■ [https://github.com/ETrobocon/etroboEV3/wiki/sample\\_program](https://github.com/ETrobocon/etroboEV3/wiki/sample_program)

- このサンプルは動作確認用です
  - 技術教育で使うのは別のプログラムなので注意すること



The screenshot shows the GitHub interface for the repository ETrobocon/etroboEV3. The page title is "sample\_program" and it was edited by Toru Eguchi on Feb 28 with 5 revisions. The main heading is "サンプルプログラム" (Sample Program). The text describes that the repository provides sample programs for the EV3way-ET walking body, specifically for use in the ET Robocon 2017 competition. It mentions support for development environments like EV3RT, MonoBrick, and leJOS EV3. A note in parentheses states that sample programs for mruby on EV3RT and TECS are included in the distribution packages. On the right side, there is a "Pages" section with 24 pages, including "Home" and "lejo3ev3\_mac".





# ここがポイント!! (Windows)



## ■ etroboEV3-master.zipを解凍し、sample\_c4フォルダを移動します

### ● 移動元

EV3way 用  
サンプルプログラム

Sample\_c4フォルダを  
workspaceフォルダ配下へ移動します

- etroboEV3-master¥SampleCode¥EV3way\_EV3RT\_sample¥sample\_c4

### ● 移動先

- 32bit 版 OS . . . C:¥cygwin¥ev3rt¥hrp2¥sdk¥workspace
- 64bit 版 OS . . . C:¥cygwin64¥ev3rt¥hrp2¥sdk¥workspace



# ここがポイント!! (Mac)



- etroboEV3-master.zipをFinder でダブルクリックで解凍します
- 解凍してできたsample\_c4フォルダを移動します
  - ドラッグ & ドロップで OK です
  - 移動元
    - etroboEV3-master/SampleCode/EV3way\_EV3RT\_sample/sample\_c4
  - 移動先
    - ev3rt/hrp2/sdk/workspace

EV3way 用  
サンプルプログラム

Sample\_c4フォルダを  
workspaceフォルダ配下へ移動します

---

## 5. サンプルアプリケーションの ビルドと実行

# ビルドと実行の手順



- [http://dev.toppers.jp/trac\\_user/ev3pf/wiki/SampleProgram](http://dev.toppers.jp/trac_user/ev3pf/wiki/SampleProgram)
  - サンプルアプリケーションのビルドと実行

 Search[Login](#) | [Preferences](#) | [Help/Guide](#) | [About Trac](#)[Wiki](#) | [View Tickets](#) | [Search](#)wiki: [SampleProgram](#)[Start Page](#) | [Index](#) | [History](#)

## サンプルアプリケーションのビルドと実行

### 実行形式とビルド方法

本プラットフォームのアプリケーションは、以下の2つの方法で実行することができます。

- 動的ローディング形式：OSとアプリケーションの実行ファイルを分け、OS実行中に、ダイナミックローディング機能を使ってアプリケーションの実行ファイルだけをロードして実行する。
- スタンドアローン形式：ダイナミックローディング機能を使わず、OSとアプリケーションを1つの実行ファイルにして単独（スタンドアローン）で実行する。

### 動的ローディング形式（ダイナミックローディング機能を利用）

#### 動的ローディング形式とは？

動的ローディング形式は、ダイナミックローディング機能をもつアプリケーションローダ（aploader）を使って、SDカードに入れた、またはBluetooth等のシリアル通信で受信したアプリケーションの実行ファイル（ELF形式）をロードして動的に実行する形式です。

#### Table of Contents

[TOPPERS/EV3RT とは？](#)  
[ダウンロード](#)  
[開発環境構築のマニュアル](#)  
[サンプルアプリケーションのビルドと実行](#)  
[ユーザマニュアル（アプリケーション開発者向け）](#)  
[EV3RT C APIリファレンス](#)  
[EV3RT C++ APIリファレンス](#)  
[FAQ](#)  
[お問い合わせ](#)



# ここがポイント!! - 1/2

本日は動的ローディング  
形式でビルドします



## 1. ビルドは、「workspace」フォルダで行います

### a. Windows + Cygwin の場合

```
$ cd /ev3rt/hrp2/sdk/workspace
```

### b. Mac の場合

```
$ cd ~/ev3rt/hrp2/sdk/workspace
```

## 2. サンプルプログラムをビルド（コンパイル）します

```
$ make app=sample_c4
```

EV3way 用  
サンプルプログラム



## ここがポイント!! - 2/2

本日は動的ローディング  
形式でビルドします



### 3. SD カードにコピーします

- a. 動的ローディング形式の場合、初回のみアプリケーションの他にアプリケーションローダーもコピーする必要があります
  - ただし、EV3RT のバージョンが変わった場合は、そのバージョン用のアプリケーションローダーをコピーし直してください
- b. 「ev3rt/sdcard」フォルダの中にある「ev3rt」フォルダと「ulimage」ファイルをSDカードにコピーします
- c. SDカードにコピーした「ev3rt」フォルダの中の「apps」フォルダに、先ほどビルドした「app」ファイルをコピーします
- d. 「USB 経由」と「Bluetooth 経由」は、各自でお試ください

- 走行体上でサンプルプログラムを実行してみ  
ましょう

---

## 6. サンプルコースの試走



# ここがポイント!!



## ■ EV3way のサンプルプログラム

- タッチセンサーを押したときにジャイロセンサーのリセットを行います
- そのため、走行体をできるだけ揺らさないようにタッチセンサーを押すことがうまく走らせるコツです

## ■ HackEV のサンプルプログラム

- HackEVのサンプルプログラムは提供していません
- しかし、光センサーとタイヤの配置がEV3wayと似ているため、EV3wayのサンプルプログラムの倒立制御と尻尾モーターの制御を無効にするとそれなりのものを作れます
- 使用するポートに合わせてソースを修正する必要があります



- ETロボコン EV3開発環境構築ガイド -github.com
  - <https://github.com/ETrobocon/etroboEV3/wiki>
- TOPPERS/EV3RT ウェブサイト – TOPPERS
  - [http://dev.toppers.jp/trac\\_user/ev3pf/wiki/WhatsEV3RT](http://dev.toppers.jp/trac_user/ev3pf/wiki/WhatsEV3RT)
- Mac OS X でETロボコンに出よう！  
（EV3RT 環境構築編） - Qiita
  - <http://qiita.com/yamanekko/items/3cee3991ea9472a1f6b6>

# EV3 開発環境の構築

---

— おしまい —

