

RTミドルウェアによるロボットプログラミング技術

2. プログラミングの基礎



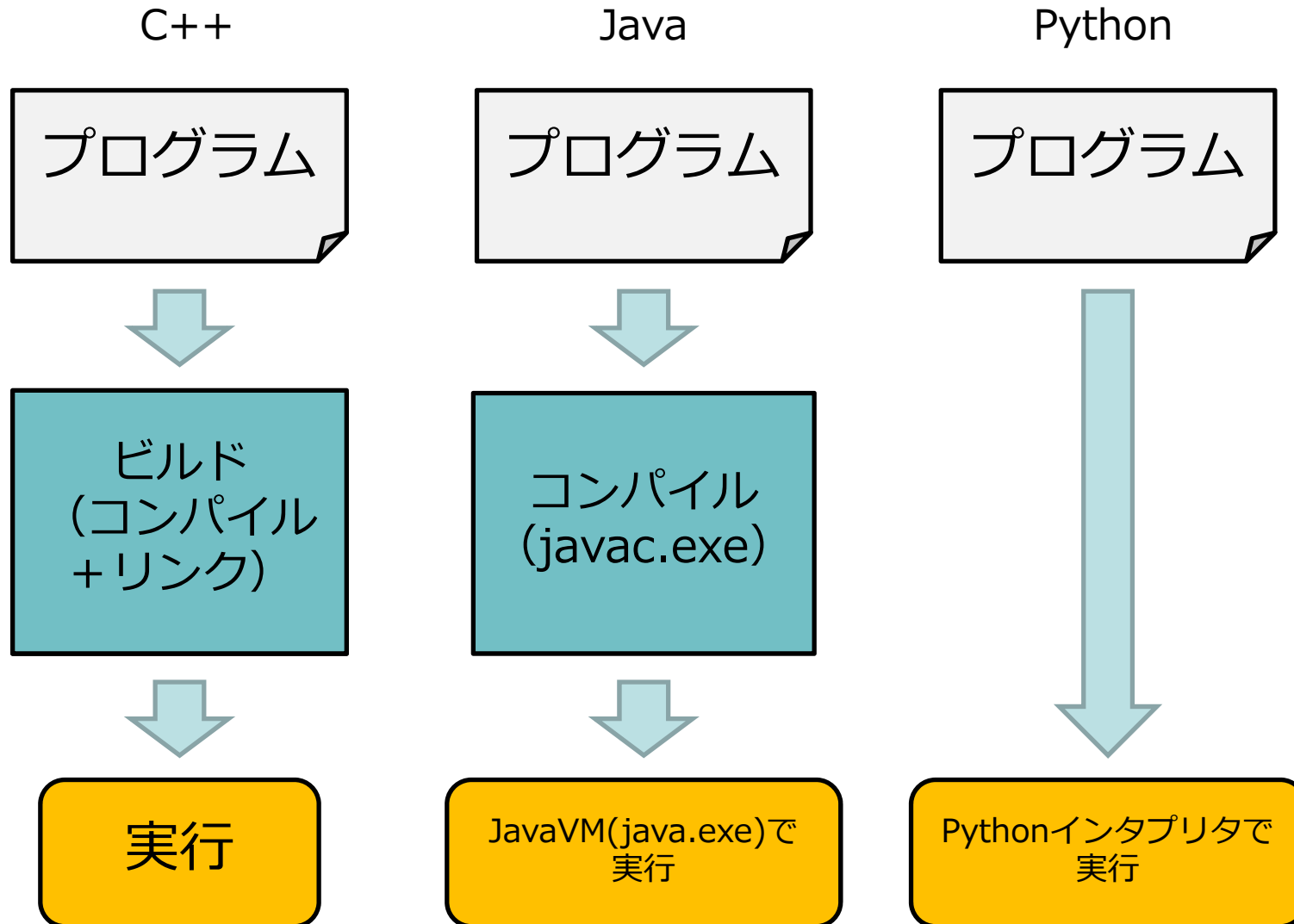
概要

1. プログラミングの基礎
2. Linuxでのプログラミング
3. Windowsでのプログラミング

目標:

1. LinuxおよびWindowsでの開発手法(主にC++)を学ぶ
2. CMakeを利用して、同じプログラムをLinuxとWindowsとでコンパイルして動作させる。

プログラミングの流れ

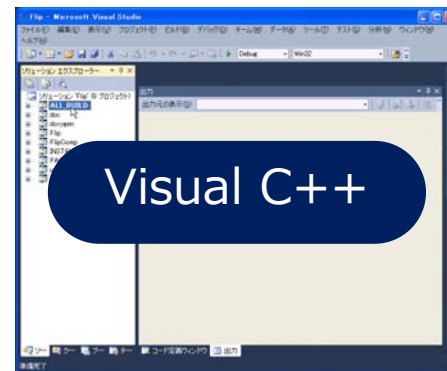
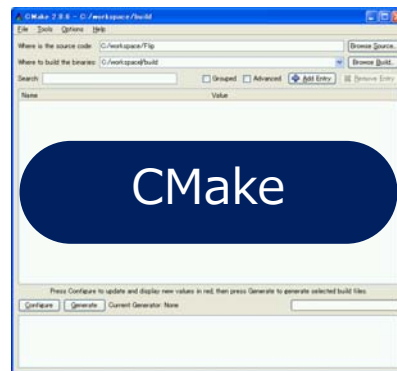
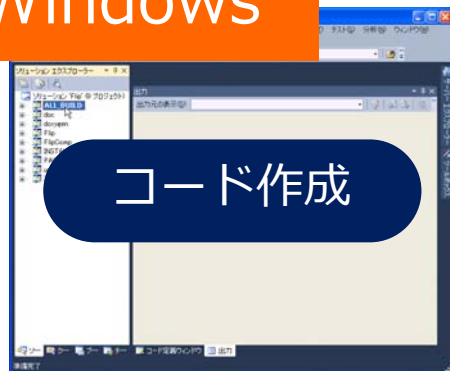


CMake

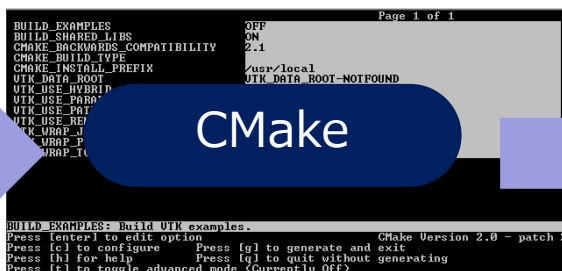
- コンパイラに依存しないビルド自動化のためのフリーソフトウェア
- 様々なOS上の様々な開発環境用ビルドファイルを生成することができる
 - Linux では Makefileを生成
 - Windows ではVC(Visual C++)のプロジェクトファイルを生成
- 最近のオープンソースソフトウェアではCMakeでビルドするようになっているものが多数。

プログラム作成の流れ

Windows



Linux



コンポーネントの
仕様の入力

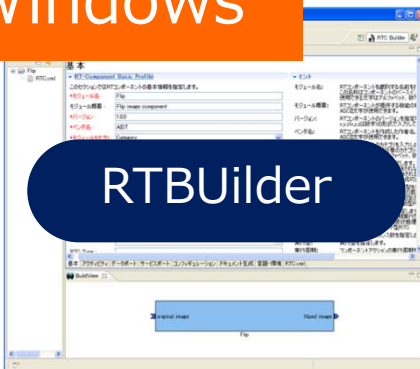
VCプロジェクトファイル
またはMakefileの生成

実装およびコンパイル
実行ファイルの生成

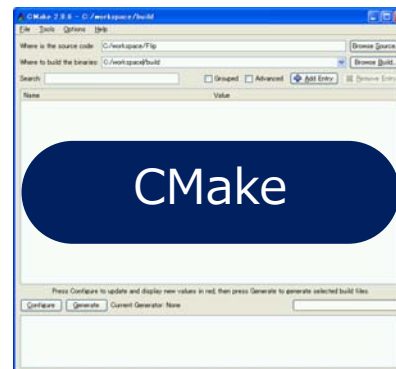
途中まで流れは同じ、コンパイラが異なる

コンポーネント作成の流れ

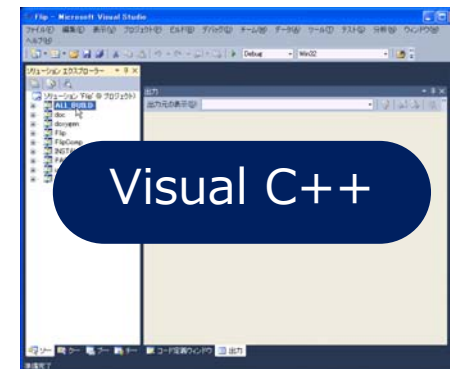
Windows



RTBuilder

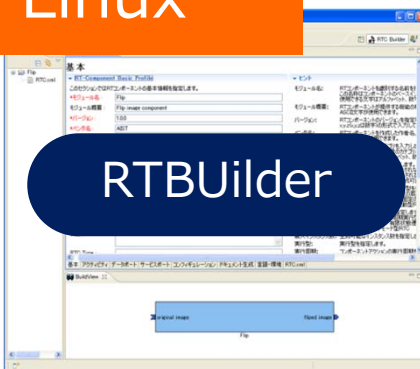


CMake

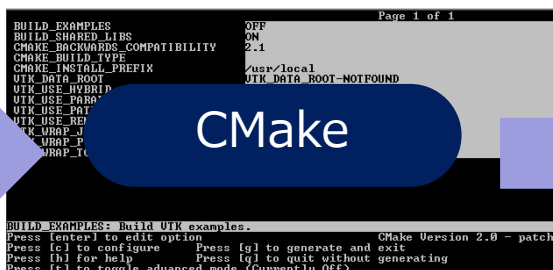


Visual C++

Linux



RTBuilder



CMake



make
+
gcc (g++)

コンポーネントの
仕様の入力

VCプロジェクトファイル
またはMakefileの生成

実装およびコンパイル
実行ファイルの生成

途中まで流れは同じ、コンパイラが異なる

演習：CMakeを使ってみよう

- CMakeを使って同じプログラムをLinuxとWindowsの両方でビルドしてみる。
- 手順：
 - サンプルプログラムをダウンロード
 - CMakeLists.txtを編集
 - cmake (cmake-gui)
 - make or VC++ でコンパイル
 - 実行

ターミナルを開く

ブラウザ (Firefox) も開いておく。

左上ボタンを押して、検索窓に "terminal" と入力し terminal を起動

このような窓が開く以降コマンドを入力して作業します。

ソフトウェアのインストール (Linux)

- LinuxでFirefoxを開く
- openrtm.orgポリテクセンター
ページへアクセス
 - <https://bit.ly/2yD1SXX>
- pkg_install_ubuntu.sh をダウンロード
- sudo sh
pkg_install_ubuntu.sh を実行
- cmakeもインストール

11月6日 (火)	
10:00 -11:00	1. コース概要 (1) ロボットシステムプログラミングの現状 (2) ロボットOS・ミドルウェア (3) RTミドルウェア(RTM)を用いたロボット開発 資料: 181106-01.pdf
11:00 -12:00 13:00-14:00	2. プログラミングの基礎 (1) プログラミングの基礎 (2) Linuxでのプログラミング (3) Windowsでのプログラミング インストールスクリプト: pkg_install_ubuntu.sh サンプルコード: arm2dof_ver001.zip 資料: 181106-02.pdf

【使用コマンド一覧】

wget: ファイルなどをダウンロード
 sudo: 管理者権限で実行
 sh: シェルコマンド
 apt-get: パッケージインストール

```

$ cd ダウンロード or Download
$ sudo sh pkg_install_ubuntu.sh -c
$ sudo apt-get install cmake
  
```

サンプルダウンロード

Linux

- ブラウザで先ほどのページから
 - arm2dof_ver001.zip をダウンロード
- Unzip コマンドでZIP ファイルを展開
- 中に2つのファイルがあることを確認

11月6日 (火)	
10:00 -11:00	1. コース概要 (1) ロボットシステムプログラミングの現状 (2) ロボットOS・ミドルウェア (3) RTミドルウェア(RTM)を用いたロボット開発 資料: 181106-01.pdf
11:00 -12:00 13:00-14:00	2. プログラミングの基礎 (1) プログラミングの基礎 (2) Linuxでのプログラミング (3) Windowsでのプログラミング インストールスクリプト: pkg_install_ubuntu.sh サンプルコード: arm2dof_ver001.zip 資料: 181106-02.pdf

【使用コマンド一覧】

mkdir: フォルダ(ディレクトリ)を作成
 wget: ファイルなどをダウンロード
 unzip: ZIPファイルを展開
 cd: ディレクトリに移動
 ls: ファイル一覧を表示

```
$ 例: cd arm2dof_ver001.zipをダウンロードしたディレクトリ
$ cd ~/Download or ~/ダウンロード
$ unzip arm2dof_ver001.zip
Archive: arm2dof_ver001.zip
  creating: arm2dof/
  inflating: arm2dof/arm2dof.cpp
  inflating: arm2dof/CMakeLists.txt
$ cd arm2dof/
$ ls
CMakeLists.txt  arm2dof.cpp
$
```

CMakeLists.txtの編集

Linux

- CMakeLists.txt をgedit(エディタ)で開く
- 2行コメントイン
 - 行頭の '#' を削除
- 保存・終了

add_executable(arm arm2dof.cpp)
の行は、

- 実行ファイル(executable) arm2dof を作成せよ
- そのためのソースコードは arm2dof.cpp である
ということを意味している。

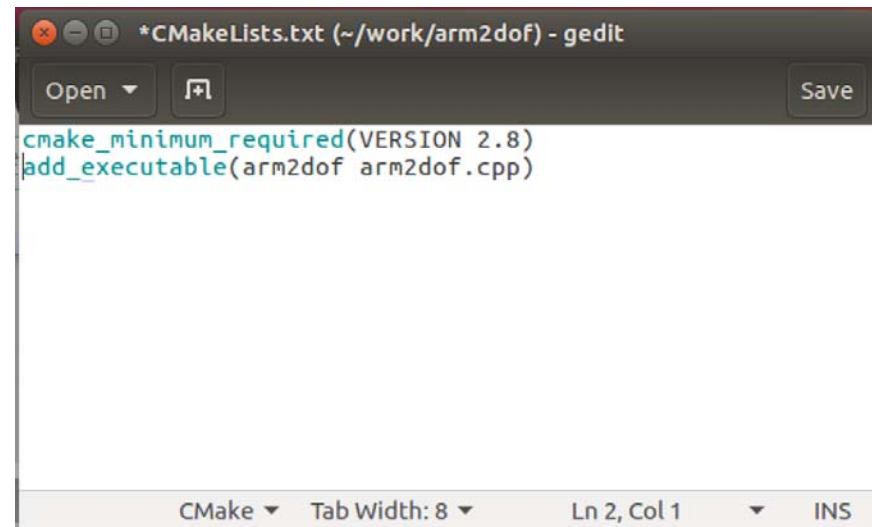
```
$ gedit CMakeLists.txt
```

または

```
$ vi CMakeLists.txt
```

または

```
$ emacs CMakeLists.txt
```



geditの編集画面

cmake & make & 実行

Linux

- build ディレクトリを作成
- cmake .. を実行
- make を実行
- arm2dofが生成される
- arm2dofを実行

【使用コマンド一覧】

cmake: コマンド

make: Makefileに基づいてコンパイル・リンクするためのコマンド

arm2dof: 今回作成する実行ファイル

【ディレクトリ指定】

. or ./ : 現在のディレクトリ(カレントディレクトリ)

.. or ../ : 現在のディレクトリの一つ上のディレクトリ(親ディレクトリ)

```
n-ando@Ubuntu1604-64:~/work/arm2dof/build$ mkdir build
n-ando@Ubuntu1604-64:~/work/arm2dof/build$ cmake ..
-- The C compiler identification is GNU 5.4.0
-- The CXX compiler identification is GNU 5.4.0

: 中略

-- Configuring done
-- Generating done
-- Build files have been written to: /home/n-ando/work/arm2dof/build

n-ando@Ubuntu1604-64:~/work/arm2dof/build$ make
Scanning dependencies of target arm2dof
[ 50%] Building CXX object CMakeFiles/arm2dof.dir/arm2dof.cpp.o
[100%] Linking CXX executable arm2dof
[100%] Built target arm2dof

n-ando@Ubuntu1604-64:~/work/arm2dof/build$ ls
arm2dof CMakeCache.txt CMakeFiles cmake_install.cmake
Makefile

n-ando@Ubuntu1604-64:~/work/arm2dof/build$ ./arm2dof
pos (x, y): -1, 1      ==> angle (th0, th1): 0, 0
pos (x, y): -0.5, 1   ==> angle (th0, th1): 0, 0
pos (x, y): 0, 1      ==> angle (th0, th1): 0, 0
pos (x, y): 0.5, 1    ==> angle (th0, th1): 0, 0
pos (x, y): 1, 1      ==> angle (th0, th1): 0, 0
n-ando@Ubuntu1604-64:~/work/arm2dof/build$
```

ソフトウェアのインストール (Windows)

- <https://bit.ly/2yD1SXX>

にアクセス

- 「インストールするソフトウェア」に従い必要なソフトウェアをインストール
 - Python
 - OpenRTM-aist
 - PyYAML
 - Cmake
 - Doxygen
 - TeraTerm

インストールするソフトウェア

あらかじめインストールしておくべきソフトウェアは以下のとおりです。以下のリンクをクリックし、ファイルをダウンロード・インストールしてください。
一部のリンクはダウンロードページへ飛びますので、飛んだ先のページ内で適切なファイルをそれぞれダウンロードしてください。

Visual Studio

- Visual Studio 2013推奨： [こちらのページ](#) から無償版をダウンロードできます。
- ポリテクセンターのPCにはインストール済みです。

Python

- Python2.7.10
- Python2.7.11もすでにリリースされていますが非推奨です。
- OpenRTM-aistやPyYAMLをインストールする前にインストールしてください

OpenRTM-aist 1.1.2-RELEASE版 (C++版、Python版)

- インストーラには 32bit/64bit の区別がありますが今回は32bit版を使用します。
- Windows用インストーラ(32bit)

その他

以下のソフトウェアも必須です。忘れずにインストールしてください。

- PyYAML
- CMake
- Doxygen
- TeraTerm
- 使い慣れたエディタ: EclipseやPythonに付属のエディタでも構いませんが、使い慣れたエディタが入っていた方が良いでしょう
 - Vim
 - sublime text3
 - 日本語化の方法
 - atom

ダウンロード

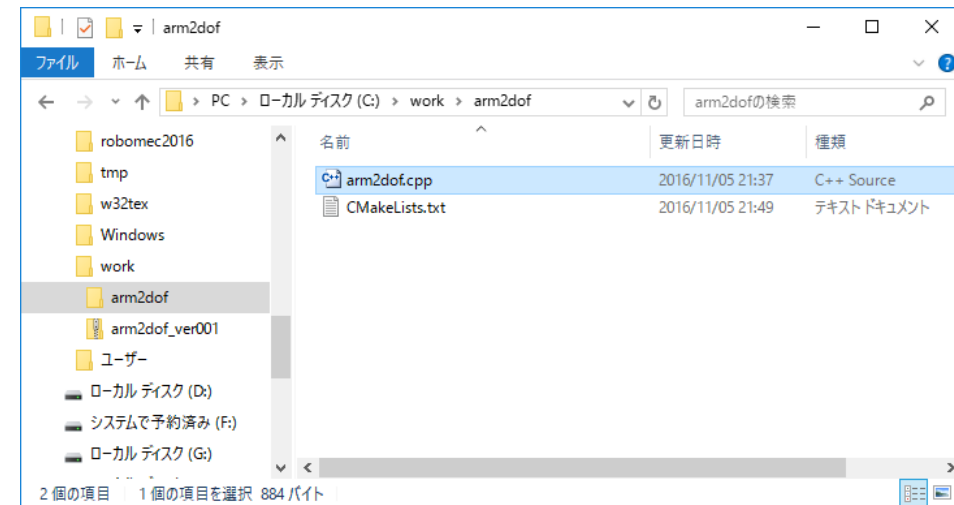
Windows

<https://bit.ly/2yD1SXX>

Windows

- ブラウザからダウンロード
- ZIPを展開
- arm2dofフォルダの下に
 - arm2dof.cpp
 - CMakeLists.txt
 の2つのファイルが展開される

11月6日 (火)	
10:00 -11:00	1. コース概要 (1) ロボットシステムプログラミングの現状 (2) ロボットOS・ミドルウェア (3) RTミドルウェア(RTM)を用いたロボット開発 資料: 181106-01.pdf
11:00 -12:00 13:00-14:00	2. プログラミングの基礎 (1) プログラミングの基礎 (2) Linuxでのプログラミング (3) Windowsでのプログラミング インストールスクリプト: pkg_install_ubuntu.sh サンプルコード: arm2dof_ver001.zip 資料: 181106-02.pdf



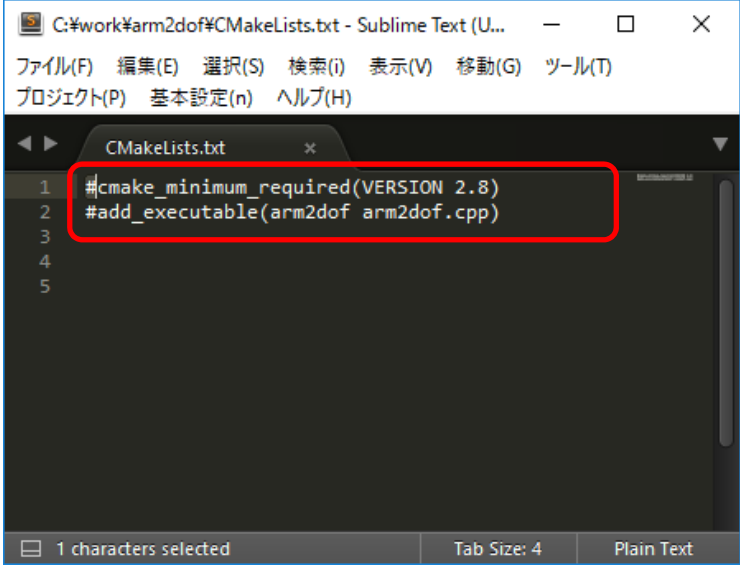
CMakeLists.txtの編集

Windows

- CMakeLists.txt をエディタで開く
- 2行コメントイン
 - 行頭の '#' を削除

add_executable(arm arm2dof.cpp)
の行は、

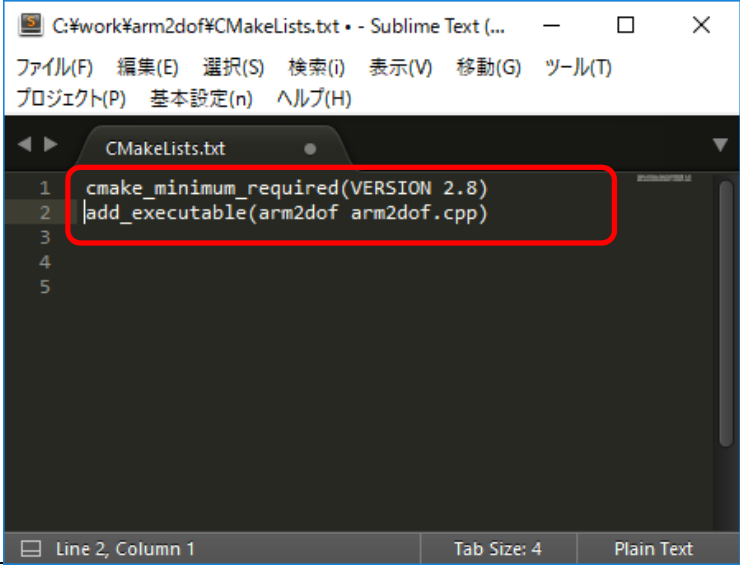
- 実行ファイル(executable) arm2dof を作成せよ
- そのためのソースコードは arm2dof.cpp である
ということを意味している。



A screenshot of the Sublime Text editor window titled 'C:\work\arm2dof\CMakeLists.txt - Sublime Text (U...'. The menu bar includes 'ファイル(F)', '編集(E)', '選択(S)', '検索(i)', '表示(V)', '移動(G)', 'ツール(T)', 'プロジェクト(P)', '基本設定(n)', and 'ヘルプ(H)'. The editor shows a file named 'CMakeLists.txt' with the following content:

```
1 #cmake_minimum_required(VERSION 2.8)
2 #add_executable(arm2dof arm2dof.cpp)
3
4
5
```

The two lines of code are highlighted with a red rectangle. The status bar at the bottom indicates '1 characters selected', 'Tab Size: 4', and 'Plain Text'.



A screenshot of the Sublime Text editor window titled 'C:\work\arm2dof\CMakeLists.txt - Sublime Text (...'. The menu bar is the same as the previous screenshot. The editor shows the same file 'CMakeLists.txt' with the following content:

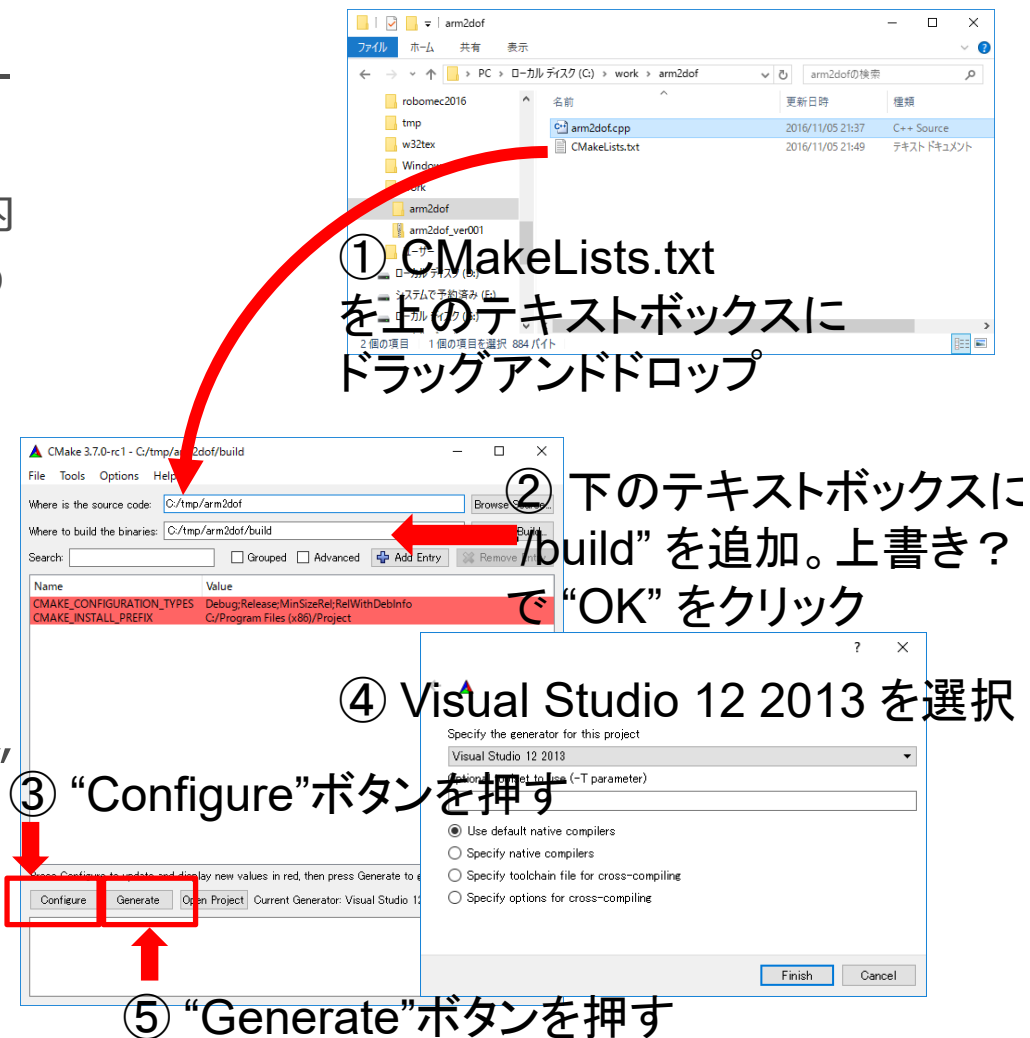
```
1 cmake_minimum_required(VERSION 2.8)
2 add_executable(arm2dof arm2dof.cpp)
3
4
5
```

The two lines of code are highlighted with a red rectangle. The status bar at the bottom indicates 'Line 2, Column 1', 'Tab Size: 4', and 'Plain Text'.

cmake-gui

Windows

- スタートメニューからcmake-gui を起動
 - スタートメニュー “cmake”内
 - 検索窓でcmakeと入力するのが早い
- CMakeLists.txtを受けのテキストBOXにDnD
- “Configure”ボタンを押下
- ダイアログでVisual Studio 12 2013を選択
- 下の窓に “Configuring done” ならOK
- “Generate”ボタンを押下
- 下の窓に “Generating done” と出ていれば完了



コンパイル (ビルド)

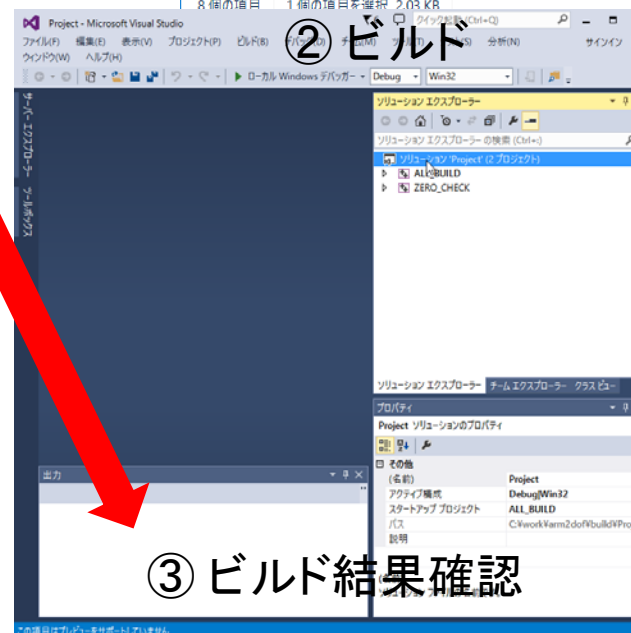
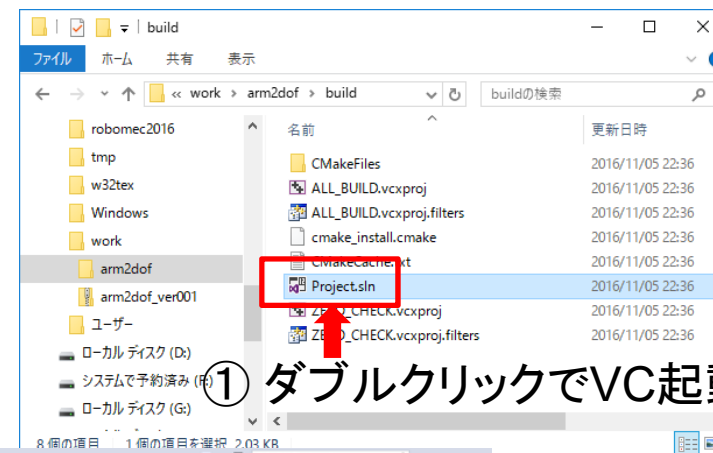
Windows

- Build フォルダ内の
“Project.sln”をダブルクリ
ック
- Visual C++ 2013が起動
- “ビルド”→“ソリューション
のビルド”でコンパイル

ビルド: 2 正常終了、0 失敗、0 更新不要、1 スキップ

と出ていれば正常終了

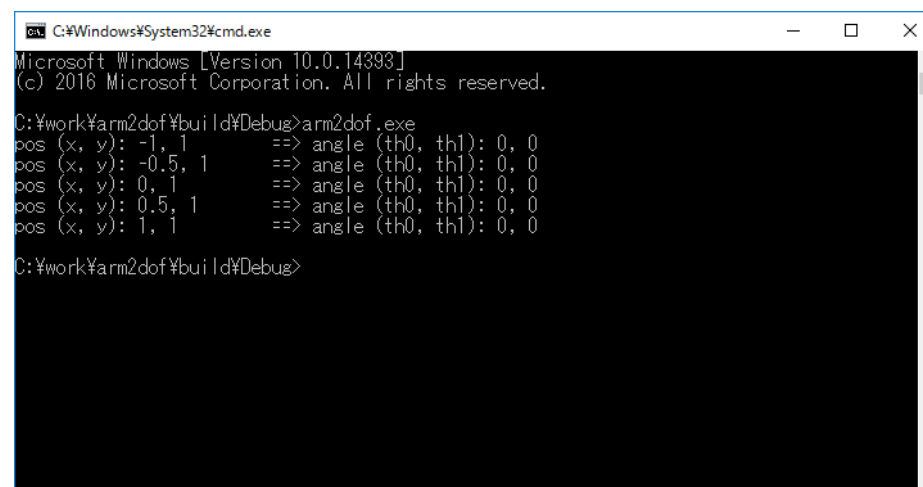
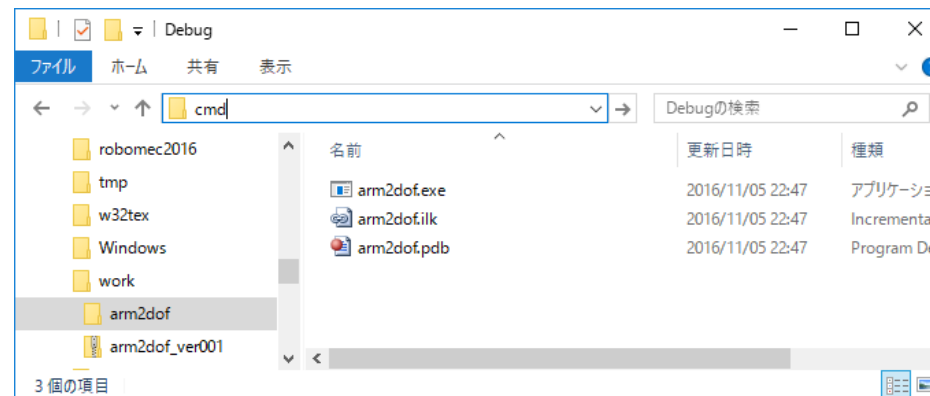
- arm2dof¥build¥Debug の
下に arm2dof.exe ができ
ている



実行

Windows

- コマンドプロンプトを起動
 - Explorerのアドレスバーに“cmd” と入力しEnter
- プロンプトで“arm2dof.exe” と入力
- 結果が表示される。
 - ダミーコードなので、 $\text{angle} = (0, 0)$ でしか表示されない
 - 2日目に練習問題として実装していただきます。



まとめ

- CMakeを利用すると、同じソースコードをWindowsでもLinuxでコンパイルできる
 - CMakeLists.txt をcmake (または cmake-gui)で処理
 - Windowsでは プロジェクトファイルとソリューションファイル
 - Linuxでは Makefile を作成
 - それぞれの方法でビルド