

QtCreator を用いた コントローラのデバッグ方法

株式会社コレオノイド
中岡 慎一郎

背景

- ロボットプログラミング(C/C++)におけるデバッグの実態
 - エラーメッセージやログをみる
 - 気合で修正 😊
 - printfデバッグ
- デバッガも活用しよう！
- デバッガを使っている人は案外少ない？
 - ロボット用に使用することの多いLinux環境ではよいデバッガがあまり知られていないのが原因？

デバッグ事情

- Windows
 - Visual Studio (C++) のデバッグが標準で使い勝手もよい
- Linux
 - 基本はGDB
 - ただしCUI (テキスト+コマンド)
 - GUIのデバッグ (内部でGDBを使用)
 - DDD
 - KDevelop
 - Eclipse
 - QtCreator **イチオシ!**
 - Visual Studio Code

QtCreator

- Ubuntuでは以下のコマンドでインストール可能
 - `sudo apt install qtcreator`
- 起動
 - `qtcreator`
- 「新しいプロジェクト」で「既存プロジェクトのインポート」
 - ファイル選択で既存ソースツリーを取り込む

デバッグの準備

- ビルド時にデバッグオプションを付けてビルド
 - CMakeではCMAKE_BUILD_TYPE を “Debug” にする
- QtCreatorの「ビルド設定」は無効化する
 - 「Buildステップ」「Cleanステップ」を削除
 - これまでどおり自前でビルドすればOK
- QtCreatorの「実行時の設定」で以下を設定
 - 実行ファイル
 - コマンドライン引数
 - 作業ディレクトリ

デバッグの実行

- デバッグ開始ボタンを押して開始
- デバッグしたい個所のソースコードを表示
- ソースコードにブレークポイントを付与
- 呼び出し履歴（バックトレース）の確認／移動
- ステップ実行
- 変数の内容を確認
- 落ちた箇所の特定

※ 操作方法はVisual Studioのデバッガとほぼ同じ

Tips

- QtCreatorの「ヘルプ」 - 「プラグインについて」で表示される「インストール済みプラグイン」で、以下をオフにしておく
 - “C++” - “ClangCodeModel”
 - テキストエディタ上での言語エラー検知機能が無効化される
- Ubuntu 20.04の場合、std::stringの内容が表示されない
 - 右クリックメニューで”Change Value Display Format”を“生データ”にするとなんとか確認できる
 - GCC9に対応できていないのが原因。以下のページから最新のQtCreatorをダウンロードしてインストールすれば解決
 - <https://www.qt.io/offline-installers>
- ROSでlaunchファイルを使って起動されるプログラムのデバッグなどには「実行中のアプリケーションにアタッチ」を利用

参考資料

- Choreonoid マニュアル
 - Eclipse を用いて Choreonoid をデバッグする方法
 - <https://choreonoid.org/ja/manuals/latest/tips/eclipse/debug.html>
 - Visual Studio を用いて Choreonoid をデバッグする方法
 - <https://choreonoid.org/ja/manuals/latest/tips/VisualStudio/debug.html>
- QtCreator ダウンロードページ
 - <https://www.qt.io/offline-installers>
- QtCreator マニュアル
 - <https://doc.qt.io/qtcreator/index.html>