

# TOPPERS 活用アイデア・アプリケーション開発 コンテスト

部門 : 活用アイデア部門  
アプリケーション開発部門  
がじえるね IoT 部門

作品のタイトル : TOPPERS\_ASP3 カーネルと astah\*-UML による  
状態マシン図でのソフト設計と動作検証の試み

作成者 : 塩出 武(しおで たけし)

共同作業者 :

対象者 : Astah を使用してソフト設計をされている方、  
状態マシン図について学習中の方など

使用する開発成果物 : TOPPERS\_ASP3 カーネル  
astah\*-UML(もしくは Professional も可)  
astah\*-m2t プラグイン  
astah\*-toppers\_flow プラグイン  
QEMU(バージョン 2.12.0)

## 目的・狙い

まだ簡易的ではありますが、astah\*-UML と m2t プラグインを用いて TOPPERS\_ASP3 カーネルのサンプルコード上で動くプログラムを出力します。加えて、サンプルプログラム上の状態を astah\*の状態マシン図にフィードバックして、F/W と状態マシン図が連動して動く(色が変わる)ようにしています。

設計モデルからコード生成することで実装作業を短縮し、かつシミュレーションによってその設計モデルを実機レスで検証することで、ソフトウェアの開発工程を短縮できないか検討してみました。

## 目次

1	アイデアアプリケーションの概要	3
2	アプリケーションの狙い	3
2.1	実機レスでのソフト開発	3
2.2	設計～実装～動作検証の作業を短縮	3
2.3	設計ドキュメントとソースコードの紐付け	3
3	アプリケーションの構成	4
3.1	<b>Toppers_Flow</b> プラグイン	4
3.1.1	<b>Toppers_Flow</b> の概要	4
3.1.2	制限事項： <b>Toppers_Flow</b> 向け共有 <b>DLL</b> ファイルの置き場について	4
3.2	<b>QEMU</b> への機能追加	5
4	サンプルプログラムの説明	5
4.1	環境準備	5
4.1.1	<b>TOPPERS-ASP3</b> シミュレーション環境	5
4.1.2	<b>astah*</b> 側 <b>m2t</b> プラグイン	6
4.1.3	<b>TOPPERS</b> テンプレートの制限事項について	6
4.1.4	<b>astah*</b> 側 <b>Toppers_Flow</b> プラグイン	7
4.2	サンプルプログラム実行	7
4.2.1	実行方法	7
4.3	制限事項	8
5	まとめ	8
6	付録：デバッグについて	9

## 1 アイデアアプリケーションの概要

astah\*-UML上で作成した状態マシン図から、Toppers\_ASP3上で動作するプログラムを出力し、その動作がリアルタイムでastah\*上のモデルにフィードバックされます。これにより状態マシン図から作成したプログラムが意図通りに動作するかをPC上で確認できます。図1参照。

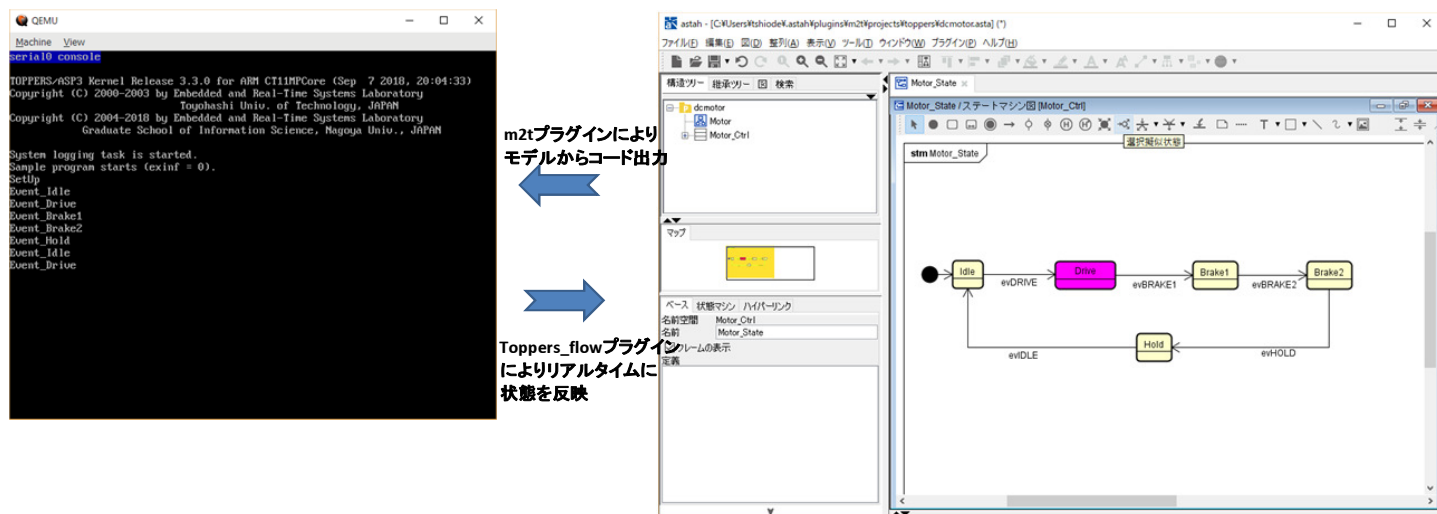


図1 アプリケーションの概要

## 2 アプリケーションの狙い

### 2.1 実機レスでのソフト開発

通常のソフト開発では設計と実装が完了後、基板上で動作確認を行います。ICEや基板上的LED、外付けのPrintf端末を用います。今回のアプリケーションではソフト動作をQEMU上でシミュレーションし、かつastah\*モデルに直接フィードバックするため、これらの機材は不要です。

### 2.2 設計～実装～動作検証の作業を短縮

m2tプラグインにより設計モデルから直接コードを出力出来るので、実装の大半は省略されます。が今回の試行では環境によるヘッダファイルの調整や、一部手作業によるソフト編集が必要でした(プラグイン設定で調整しきれませんでした)。m2tテンプレートの調整に慣れれば解消するかもしれませんが今後の課題です。

### 2.3 設計ドキュメントとソースコードの紐付け

忙しくなってくると、納期に間に合わせるためにどうしても実装と動作検証を優先せざるを得なくなります。その結果、当初の設計からの変更や調整は資料に反映できなくなるため、ドキュメントとソフトウェアの乖離が大きくなります。私はよくそうなります。今回のアプリケーションはモデル(設計)からソフト(実装)を直接出力することで乖離をなくするという試みでもあります。

### 3 アプリケーションの構成

アプリケーションを構成するソフトウェア (開発成果物) を、表 1 にまとめます。

表 1 使用した開発成果物まとめ

ソフトウェア名	役割	備考
ASP3 カーネル	F/W の本体	
astah*-UML(64bit 版)	状態マシン図作成用	※ community 版は不可です* <sup>1</sup>
m2t プラグイン	モデル→コード出力用	
Topeprs_Flow プラグイン (自作)	F/W →状態マシンモデル反映用	astah*プラグインと DLL ファイル* <sup>2</sup>
asta*-Plugin-SDK	↑ Toppers_Flow 作成用	プラグイン開発環境
Java-jdk	↑ Toppers_Flow 作成用	astah-build の際に必要
jna ライブラリ	Java ⇔ Win32 連携用	Java-Native-Access の略
QEMU	F/W シミュレーション用	2.12.0 バージョン使用 (64bit 環境でビルド)
MSYS2	QEMU ビルド用	64bit 版
VisualStudioExpress2015	Toppers_Flow 用 DLL 作成	64bit 環境でビルド

<sup>1</sup> asta\*-api の編集機能が無効のため色が変わりません

<sup>2</sup> toppers\_flow.dll ファイルの置き場に制限があります

#### 3.1 Toppers\_Flow プラグイン

QEMU 上で動いている F/W の状態を参照して、astah\*のモデルに反映するプラグインを作成しました。TOPPERS アプリ用ということで Toppers\_Flow としましたが、もともとは astah\*スクリプトエディタプラグインのサンプルコードがベースです。<sup>\*3</sup> 本家は ECMAScript(JavaScript) ですが、QEMU 側との DLL 共有の仕方がわからないのでプラグイン化しました。ただ、今回のプラグインアプリでは.dll ファイルの置き場をハードコーディングしているため、JavaScript で共有出来るならその方がシンプルかもしれません。

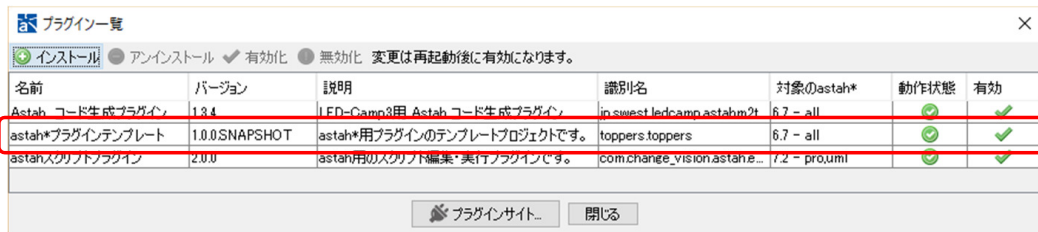


図 2 Toppers\_Flow プラグイン

##### 3.1.1 Toppers\_Flow の概要

1. 100ms 毎に QEMU の共有メモリを参照する。メモリには状態を示す文字列が記録されている。(Idle、Drvie、Hold...)
2. 読み出した文字列を astah\*状態マシン図のステート名と比較する
3. 一致するステートの色を変更する
4. 一致するステートが変わったら前回のステートの色を元に戻して新しいステートの色を変更する
5. メニューの Toppers\_Flow を選択すると開始ダイアログを表示してプラグインを開始する
6. 再度選択すると終了ダイアログを表示して終了する

##### 3.1.2 制限事項：Toppers\_Flow 向け共有 DLL ファイルの置き場について

QEMU との共有メモリのために.dll ファイルを作成しています。これを astah\*側の Toppers\_Flow プラグインから参照するのですが、現時点でどうしても相対パス設定がうまくいかないため、やむを得ず絶対パスとさせてもらっています。申し訳ございませんが、以下のように、C ドライブにディレクトリを作成して、toppers\_flow.dll ファイルを置いてください。  
C:\work\toppers\toppres\_flow.dll

\*3 <https://ja.astahblog.com/sample-scripts/#change-color>

### 3.2 QEMU への機能追加

ASP3 カーネル向けのシミュレーション環境に使用している QEMU(realview-eb-mpcore ベース) に状態参照のための共有メモリを追加しています。サンプルプログラムがここに現在の状態を記録し、それを **Toppers\_Flow** が参照します。図 3 にレジスタマップを記載します。今回は、ボード名を"realview-eb-toppers"として現行環境と分けました。

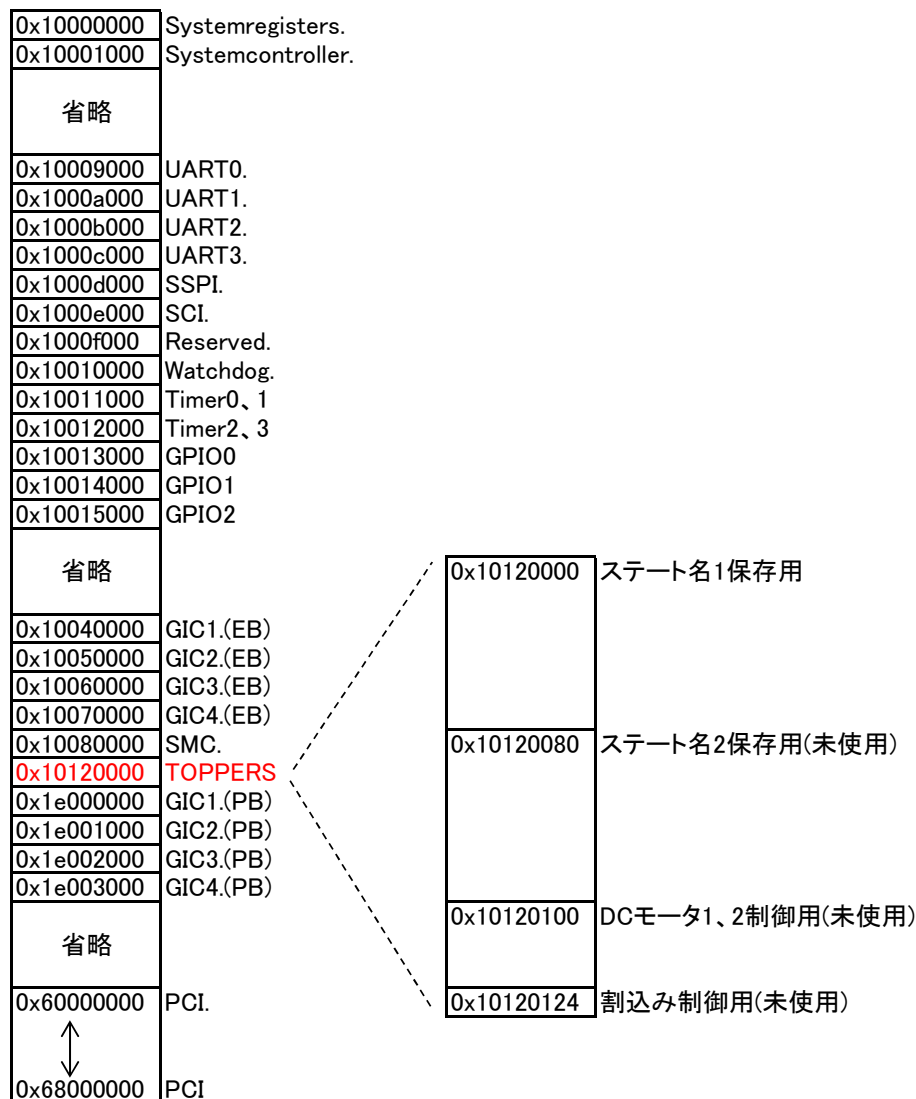


図 3 QEMU への機能 (共有レジスタ) 追加

## 4 サンプルプログラムの説明

### 4.1 環境準備

#### 4.1.1 TOPPERS-ASP3 シミュレーション環境

ASP3 カーネルの環境 (今回は **Core Tile for ARM11 MPCore (ARM)** 簡易パッケージ) を落としてきて、任意のフォルダに展開し、コンフィグレーションと **make** を行います。QEMU については **Toppers\_Flow** 用に機能追加したもの (事務局様に提出) を使用します。以下簡単ですが手順です。

1. `./configure.rb -T ct11mpcore_gcc -O "-DTOPPERS_USE_QEMU"` ← コンフィグレーション
2. `export PATH=$PATH:D:\GNU Tools ARM Embedded\4.9 2015q2\bin` ← ツールのパスを通す (D ドラの場合)
3. `make(m2t プラグインからは.cpp(C++) 形式で出力されますので C++ 向けの make 設定をしています。`
4. `./qemu-system-arm -M realview-eb-toppers -semihosting -m 128M -smp 1 -serial vc -kernel asp` ← 実行

## 4.1.2 astah\*側 m2t プラグイン

1. m2t プラグインを **astah\*-UML** にインストール
2. **dcmotor.asta** の状態マシン図を開く (事務局様に提出)
3. **TOPPERS** 向けのテンプレートファイル (事務局様に提出)\*<sup>4</sup>を設定する。図 4 参照
4. m2t プラグインでコードを出力する→※一部手動で編集する制限事項有り
5. **TOPPERS** 側で **make**

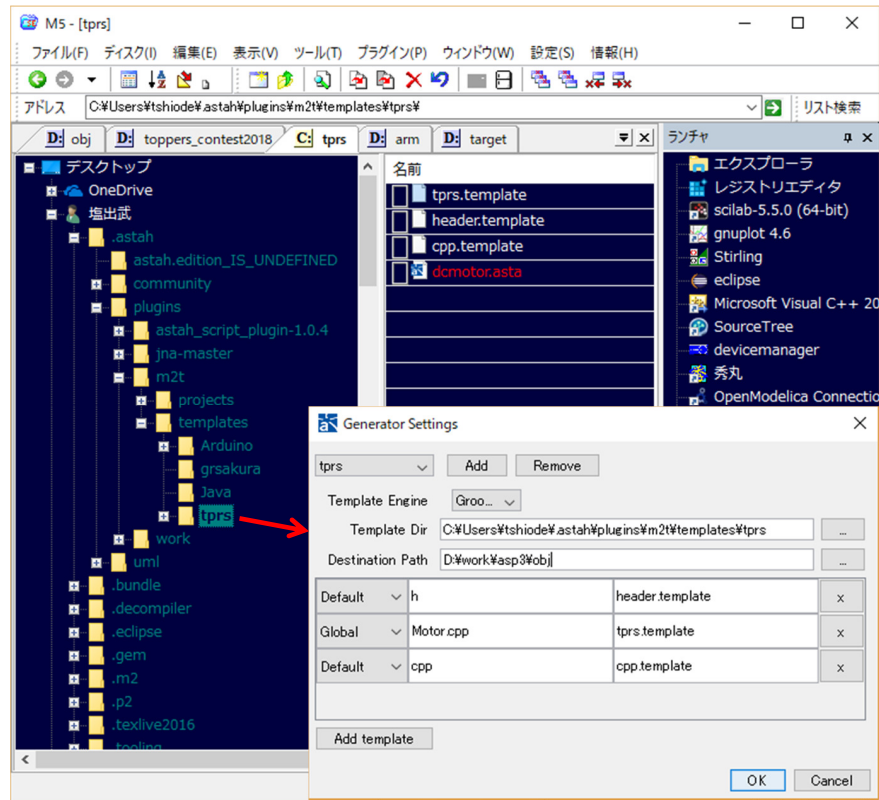


図 4 TOPPERS 向け m2t プラグインの設定

## 4.1.3 TOPPERS テンプレートの制限事項について

**m2t** テンプレート設定を手に乗せ切れていないため、コード出力を完全自動化できませんでした。以下、手動で編集した箇所を記載します (青字)。事務局様に提出のものは編集済みです。

## ■手動調整部分 Motor.cpp

```
void tprs_event(TPRS_EVENT event)
{
    switch(event){
        // set case for event..
        case EVENT_IDLE:    motor_Ctrl->transition(Motor_Ctrl::evIDLE);    break;
        case EVENT_DRIVE:   motor_Ctrl->transition(Motor_Ctrl::evDRIVE);   break;
        case EVENT_BRAKE1:  motor_Ctrl->transition(Motor_Ctrl::evBRAKE1); break;
        case EVENT_BRAKE2:  motor_Ctrl->transition(Motor_Ctrl::evBRAKE2); break;
        case EVENT_HOLD:    motor_Ctrl->transition(Motor_Ctrl::evHOLD);    break;
        default:
            break;
    }
    motor_Ctrl->doAction();
}
```

\*<sup>4</sup> Arduino をベースに改編しました

## 4.1.4 astah\*側 Toppers\_Flow プラグイン

**m2t** プラグイン同様にインストールください。図 2 参照。3.1.2 節に前述のように.dll ファイルの配置に制限があります。ご了承ください。**topers\_flow.dll** をこちらに置いてください。→ C:\work\toppers\toppres\_flow.dll

## 4.2 サンプルプログラム実行

サンプルプログラムは **DC** モータの駆動制御を状態マシン図で設計したという想定で作ってみました。<sup>\*5</sup>

## 4.2.1 実行方法

1. 前述の環境準備に沿って **TOPPES-ASP3** シミュレーション環境を実行してください
2. 次に **astah\*** の **dcmotor.astah** のメニューから **Toppers\_Flow** プラグインを実行します図 5。開始のダイアログが出ますので **OK** を押してください
3. キーボードの **'s'** キーを押して初期セットアップを行います
4. 既存のサンプルプログラムをベースにキーボタンに各種イベントを割り当てています表 2 を参照してイベントを起こして **astah\*** の状態マシン図が反応すれば成功です
5. 再度 **astah\*** の **dcmotor.astah** のメニューから **Toppers\_Flow** プラグインを押すと **End** メッセージが出て、モデルを初期化後終了します。

表 2 キーイベントまとめ (モータ駆動を想定)

キー入力	イベント	備考
's'	状態マシン制御セットアップ	setup() をコールして Idle 状態になります
'i'	Idle イベント発生	Drive にのみ遷移可能
'd'	Drive イベント発生	Brake1 のみ遷移可能
'b'	Brake1 イベント発生	Brake2 のみ遷移可能
'B'	Brake2 イベント発生	Hold のみ遷移可能
'h'	Hold イベント発生	Idle のみ遷移可能
'Shif't + 'Q'	終了	既存サンプルプログラムと同じ

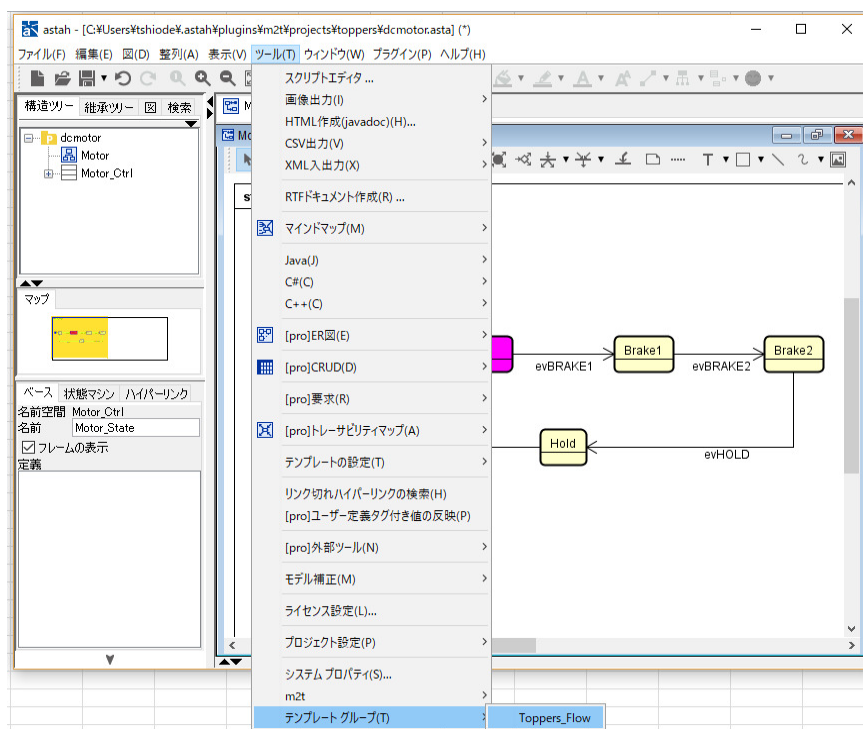


図 5 Toppers\_Flow の開始

\*5 今回はモータ側のシミュレーションが間に合いませんでしたので、モデル側の動きのみとなります

### 4.3 制限事項

動作確認した中で、以下の不具合がわかっています。ご了承ください。

1. **Toppers\_Flow** を再起動する際は、一度 **astah\***を終了して再度起動してください。終了後、再起動を掛けるとエラーが出ます。
2. **Tooppers\_Flow** を起動する際は事前に **TOPPES** のサンプルプログラムを起動しておいてください。**Toppers\_Flow** を先に実行すると、**DLL** の読み込みに失敗します。
3. **Toppers\_Flow** を起動後、一度もイベントを入れずに再度メニューから **Toppers\_Flow** を押すと再度スタートさせようとして失敗します

## 5 まとめ

以下今回のまとめです。

1. **astah\***状態マシン図から **TOPPERS** 上で走るコードを出力するテンプレートを **m2t** プラグインに追加しました。
2. **F/W** の状態を **astah\***状態マシンにフィードバックして検証できるようにしました。**Toppers\_Flow** プラグイン
3. これにより設計～実装～検証を効率化するのが狙いです。
4. ドキュメントと最終成果物 (プログラムコード) を紐付けるのも狙いです。

残念ながら作成の過程で、いくつかの制限事項、不具合が見つかっています。引き続き改善したいと思います。今回の内容が、組み込みソフトウェアの生産性向上につながれば幸いです。どうぞよろしくお願い致します。



## 6 付録：デバッグについて

ASP カーネルでも使用していた **eclipse** と **zylin** デバッガで **TOPPERS** 側をデバッグできます。以下手順です。

1. **eclipse** 側で **Zylin** デバッガの設定をする。図 6 参照
2. **QEMU** をデバッグ有効にして起動↓  
`./qemu-system-arm -S -gdb tcp::1234 -M realview-eb-toppers -semihosting -m 128M -smp 1 -serial vc -kernel asp`
3. **eclipse** のデバッガを起動※起動時に **main** がない旨のワーニングが出ますがそのまま続行ください
4. デバッグ開始。図 6 参照

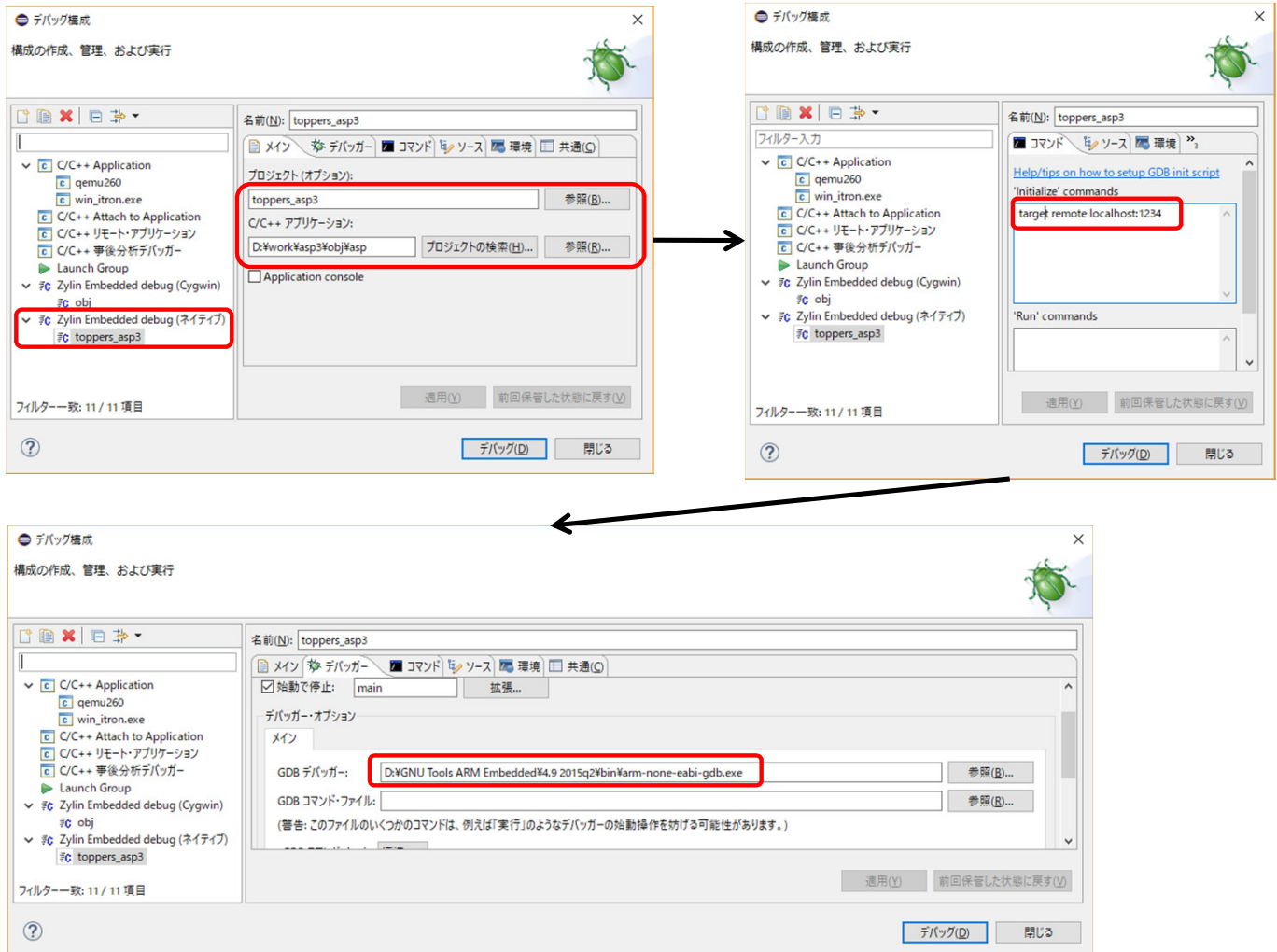


図 6 eclipse のデバッグ設定

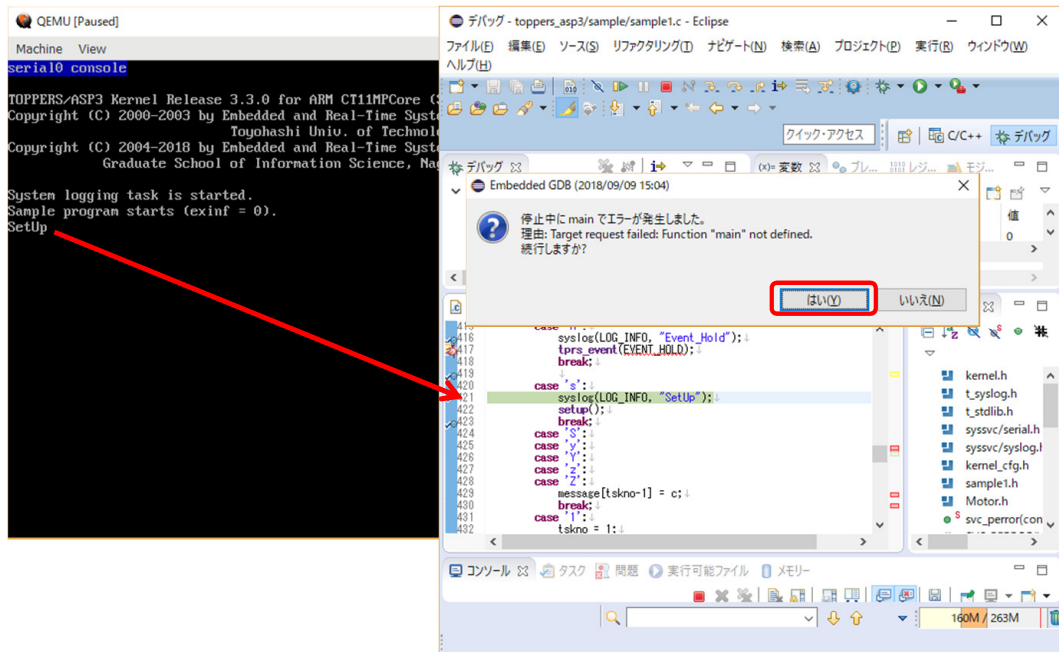


図 7 eclipse によるデバッグ