

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

作品のタイトル : GoogleTest と FakeFunctionFramework による
TOPPERS アプリケーションの単体テスト自動化

作成者 : 仙石 友輝

共同作業者 : 薄井 茜、高橋 佑梨、村山 晴基

対象者 : 組込みソフトウェア初心者、
単体テストの自動化にご興味のある方

使用する開発成果物 : TOPPERS/ASP3 カーネル

目的・狙い

・目的

組込みソフトウェア開発では、実機が用意されないと通信系やレジスタ・メモリ系のテストが困難です。ソースコードを改変すれば出来なくもないですが、製品に載せるソースコードに手を加えることは避けたいです（品質が担保できなくなる）。ソースファイルを疑似テスト用と本番用で入れ替えすることも考えられますが、かなり手間となります。

これらの課題を解決する為に、GoogleTest と FakeFunctionFramework(FFF)を用いて、本番用ソースコードに手を加えず、実機不要の単体テスト環境を構築しました。

・狙い

昨今の半導体不足や短納期開発における実機不足の改善。

誤ってテストコードが本番用ソフトに混入することの防止。

いきなり大規模開発に単体テスト自動化を適用しようとするとう失敗しがちのため、先ず小規模なカップラーメンタイマに適用することで、基本を理解する。

アイデア/アプリケーションの概要

私共 4 人は、パイオニアシステムテクノロジー(株)の 2022 年度新入社員です。
新入社員研修にて、名古屋大学情報学研究科附属組込みシステム研究センター (NCES)の「組込みソフトウェア開発技術の基礎」(以降、「組込みソフトウェア開発技術の基礎」と略)を教材として学習しました。
次に、弊社の 2021 年度新入社員が第 11 回 TOPPERS 活用アイデア・アプリケーション開発コンテスト活用アイディア部門銀賞を頂いた「データキュー機能と固定長メモリプール機能を用いたタスク間通信に関する教材」に沿って「組込みソフトウェア開発技術の基礎」に掲載のカップラーメンタイマを改造しました。
これらの学習内容を基にプロジェクト演習にて、カップラーメンタイマの単体テスト自動化環境を作成しました。今回の応募作品は、このプロジェクト演習をまとめたものになります。

1.著作権表記

コンテスト応募に伴い、「組込みソフトウェア開発技術の基礎」を改変・再配布する形になるため、著作権を以下のとおり示します。

NEP基礎コース01
組込みソフトウェア開発技術の基礎
ITRONプログラミング実習編
(STM32)

Copyright (C) 2014-2018 by 名古屋大学組込みシステム人材育成プログラム
Copyright (C) 2006-2007 by 名古屋大学組込みソフトウェア技術者人材養成プログラム
Copyright (C) 2006-2008 by 高田広章, 本田晋也
Copyright (C) 2014 by 松浦光洋

上記著作権者は、以下の(1)~(4)の条件を満たす場合に限り、本コンテンツ（本コンテンツを改変・翻訳したものを含む、以下同じ）を使用・複製・改変・翻訳・再配布（以下、利用と呼ぶ）することを無償で許諾する。

- (1) この枠内の著作権表記等が、そのままの形でコンテンツ中に含まれていること。
- (2) 本コンテンツを再配布する場合には、再配布の形態等を、以下のウェブサイトから報告すること。
<https://www.nces.i.nagoya-u.ac.jp/NEP/materials/forms/01.html>
- (3) 本コンテンツを改変・翻訳する場合には、コンテンツを改変・翻訳した旨の記述を、コンテンツ中に含めること。また、改変・翻訳者の著作権表記等は、この枠内の著作権表記等とは別に行うこと。
- (4) 本コンテンツの利用により直接的または間接的に生じるいかなる損害からも、上記著作権者を免責すること。

※ 本コンテンツ中の一部は、文部科学省特別経費により、名古屋大学組込みシステム人材育成プログラム(NEP)の一環として作成しました。

※ 本コンテンツの一部は、文部科学省科学技術振興調整費により、名古屋大学組込みソフトウェア技術者人材養成プログラム(NEXCESS)の一環として作成しました。

※ 本コンテンツ中に記載されている商品名やサービス名などは、各社の商標または登録商標です。

2.テスト対象

テスト対象はカップラーメンタイマです。カップラーメンタイマの外観や仕様、開発環境等は、「データキュー機能と固定長メモリプール機能を用いたタスク間通信に関する教材」を参照願います。

<https://www.toppers.jp/contest.html>

2021年度

第11回TOPPERS活用アイデア・アプリケーション開発コンテストの審査結果は、以下の通りです。受賞された皆様、おめでとうございます。

▪ 活用アイデア部門

銀賞
データキュー機能と固定長メモリプール機能を用いたタスク間通信に関する教材 バイオニアシステムテクノロジー㈱2021年度新入社員一同
 コンテスト応募資料
 教材スライド
 サンプルプログラム

3.おすすめの学習順

(1)組込みソフトウェア初心者の場合、先ず「組込みソフトウェア開発技術の基礎」にて学習するとよいでしょう。この教材は初心者向けの教材で、とても分かりやすいです。またこの教材は、初心者が最初に躓きがちな開発環境の構築方法についても分かりやすく記載されています。

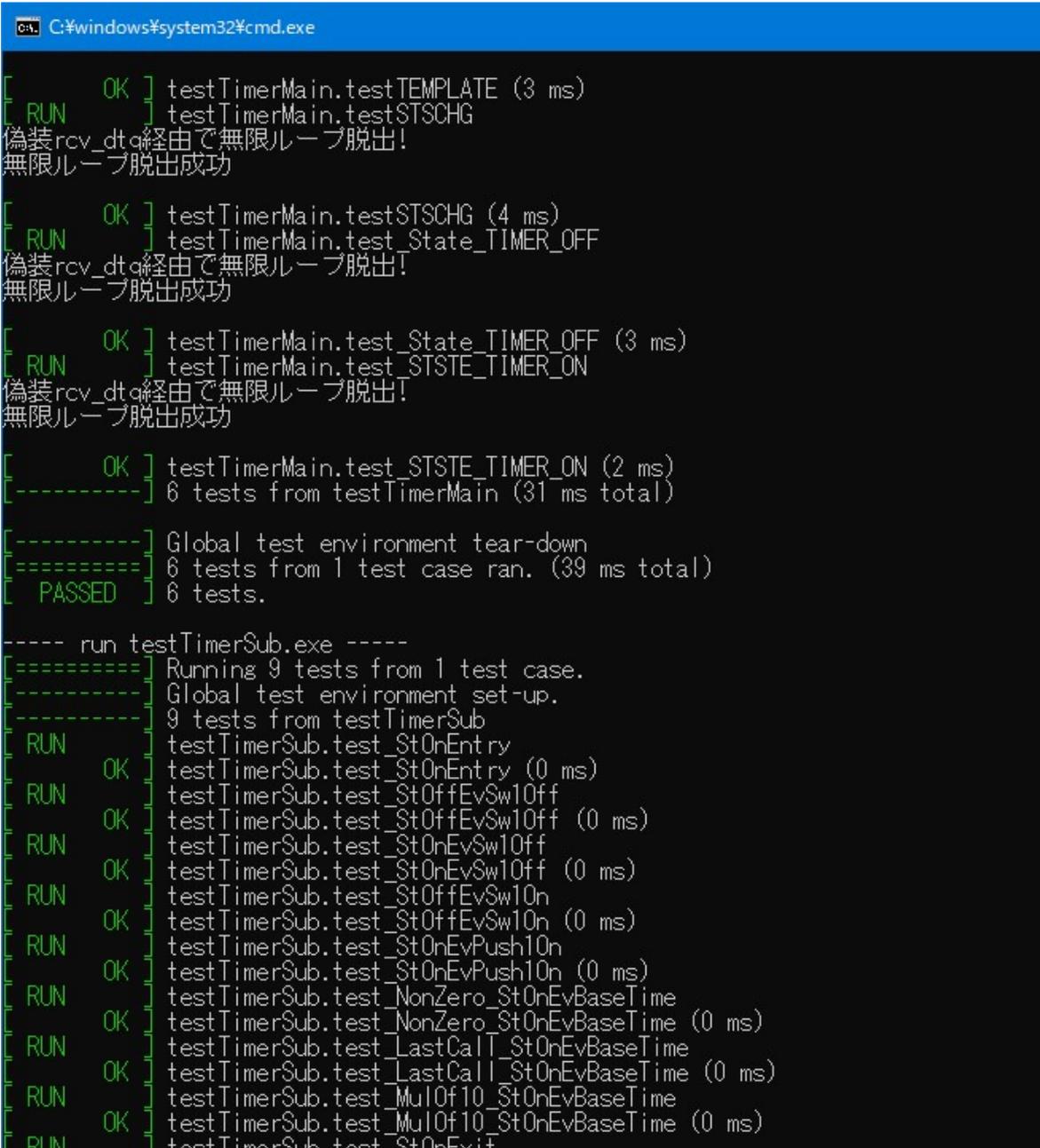
(2)次に前項で紹介した「データキュー機能と固定長メモリプール機能を用いたタスク間通信に関する教材」でテスト対象を理解してください。この教材は「組込みソフトウェア開発技術の基礎」と同一の開発環境、及び同一のハードウェア構成のため、シームレスに学習することが出来ます。また、カップラーメンタイマの仕様も「組込みソフトウェア開発技術の基礎」と同一のため、正しく動作しているか否かがすぐ分かります。

(3)テスト対象を理解した上で、添付のスライドとサンプルコードにて単体テストの自動化に取り組むとよいでしょう。

4.アプリケーションの概要

GoogleTest + FFF (Fake Function Framework) によるカップラーメンタイマの単体テスト自動環境になります。タスクのエントリーポイント等で多用される無限ループを含む関数のテストにも、TOPPERS サービスコールの偽装によって対応しております。

テスト結果は Dos プロンプト(図 1)に表示されるとともに xml へ出力(図 2)、xml をパースして csv に変換します(表 1,表 2)。



```
C:\windows\system32\cmd.exe

[ OK ] testTimerMain.testTEMPLATE (3 ms)
[ RUN ] testTimerMain.testSTSCHG
偽装rcv_dtq経由で無限ループ脱出!
無限ループ脱出成功

[ OK ] testTimerMain.testSTSCHG (4 ms)
[ RUN ] testTimerMain.test_State_TIMER_OFF
偽装rcv_dtq経由で無限ループ脱出!
無限ループ脱出成功

[ OK ] testTimerMain.test_State_TIMER_OFF (3 ms)
[ RUN ] testTimerMain.test_STSTE_TIMER_ON
偽装rcv_dtq経由で無限ループ脱出!
無限ループ脱出成功

[ OK ] testTimerMain.test_STSTE_TIMER_ON (2 ms)
[-----] 6 tests from testTimerMain (31 ms total)

[-----] Global test environment tear-down
[=====] 6 tests from 1 test case ran. (39 ms total)
[ PASSED ] 6 tests.

---- run testTimerSub.exe ----
[=====] Running 9 tests from 1 test case.
[-----] Global test environment set-up.
[-----] 9 tests from testTimerSub
[ RUN ] testTimerSub.test_StOnEntry
[ OK ] testTimerSub.test_StOnEntry (0 ms)
[ RUN ] testTimerSub.test_StOffEvSw10ff
[ OK ] testTimerSub.test_StOffEvSw10ff (0 ms)
[ RUN ] testTimerSub.test_StOnEvSw10ff
[ OK ] testTimerSub.test_StOnEvSw10ff (0 ms)
[ RUN ] testTimerSub.test_StOffEvSw10n
[ OK ] testTimerSub.test_StOffEvSw10n (0 ms)
[ RUN ] testTimerSub.test_StOnEvPush10n
[ OK ] testTimerSub.test_StOnEvPush10n (0 ms)
[ RUN ] testTimerSub.test_NonZero_StOnEvBaseTime
[ OK ] testTimerSub.test_NonZero_StOnEvBaseTime (0 ms)
[ RUN ] testTimerSub.test_LastCall_StOnEvBaseTime
[ OK ] testTimerSub.test_LastCall_StOnEvBaseTime (0 ms)
[ RUN ] testTimerSub.test_MulOf10_StOnEvBaseTime
[ OK ] testTimerSub.test_MulOf10_StOnEvBaseTime (0 ms)
[ RUN ] testTimerSub.test_StOnEvif
```

図 1.テスト結果の表示

```

testTimerSub.xml X
cup_timer10 > unit_test > xml_output > testTimerSub.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <testsuites tests="9" failures="0" disabled="0" errors="0" time="0.21" name="AllTests">
3    <testsuite name="test" tests="9" failures="0" disabled="0" errors="0" time="0.119">
4      <testcase name="test_StOnEntry" status="run" time="0" classname="test" />
5      <testcase name="test_StOffEvSw10ff" status="run" time="0" classname="test" />
6      <testcase name="test_StOnEvSw10ff" status="run" time="0" classname="test" />
7      <testcase name="test_StOffEvSw10n" status="run" time="0" classname="test" />
8      <testcase name="test_StOnEvPush10n" status="run" time="0" classname="test" />
9      <testcase name="test_NonZero_StOnEvBaseTime" status="run" time="0" classname="test" />
10     <testcase name="test_LastCall_StOnEvBaseTime" status="run" time="0" classname="test" />
11     <testcase name="test_MulOf10_StOnEvBaseTime" status="run" time="0" classname="test" />
12     <testcase name="test_StOnExit" status="run" time="0" classname="test" />
13   </testsuite>
14 </testsuites>
15

```

図 2.テスト結果の xml 出力

表 1.テスト結果概要出力

	A	B	C	D	E
1	test name	test case count	failue count	error count	execute time[ms]
2	testTimerMain	6	0	0	0.002
3	testTimerSub	9	0	0	0.119
4	testBaseTimeCyclic	2	0	0	0.001
5	testSwScanCyclic	6	0	0	0.001
6	testBlinkMain	7	0	0	0.003
7	testBlinkSub	8	0	0	0.154
8	testLed4BlinkCyclic	2	0	0	0

表 2.テスト結果詳細出力

	A	B	C	D
1	test case	test status	failure count	execute time[ms]
2	testSwScanHandler	run	0	0
3	testSwScanTaskSw1on	run	0	0
4	testSwScanTaskSw1off	run	0	0
5	testSwScanTaskPushon	run	0	0
6	testSwScanTaskPushoff	run	0	0
7	testSwScanTaskDouble	run	0	0

5.動作環境

この単体テスト環境は、以下の環境で動作確認しております。

Windows10 Pro 64 ビット

Pleiades All in One Eclipse リリース 2021 Ultimate

Windows x64 Ultimate Full Edition

<https://mergedoc.osdn.jp/#pleiades.html>

上記サイトから、Eclipse Ultimate をダウンロードしてインストールしてください(この単体テスト環境は CDT と Python を使用するため、Ultimate 版をインストールしております)。

Python と gcc の環境があれば他の環境でも動作すると推測します。

環境設定の詳細は添付のスライドを参照願います。

6.添付ファイルの構成

- ・教材のスライド:slide.pptx

先ずこのスライドをご覧ください。以下の様に順を追って説明しております。

目次

- ・著作権表記
- ・参考サイト、参考資料

- | | |
|-----------------------------|--------------|
| 1. 概要 | 6. 実行ファイルの生成 |
| 2. テストの全体像 | 7. テストの自動実行 |
| 3. テスト対象 | 8. テスト結果 |
| 4. テスト環境の構築 | 9. テストの応用 |
| 5. テストケース | 10. まとめ |
| 5.1. 各テストケースに共通する箇所の解説 | |
| 5.2. テストケースの解説(1)スイッチキャンタスク | |
| 5.3. テストケースの解説(2)LED4点滅タスク | |

- ・テスト対象のソースコード:

sample¥RX63N¥cup_timer10 -> NCES TRAINING BOARD 用

sample¥stm32¥cup_timer10 -> STM32F401 Nucleo 用

- ・単体テスト環境一式:unit_test フォルダ

以上