

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 : 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
- 作成者 : パイオニアシステムテクノロジー(株)2021年度新入社員一同
(鍵谷 慧、狩野 海大、齋藤 冬樹、佐々木 未来、佐藤 国渡)
- 共同作業 :
- 対象者 : RTOS 初心者、メールボックス機能の代替案をお探しの方
- 使用する開発成果物 : TOPPERS/ASP3 カーネル

目的・狙い

TOPPERS 第 3 世代カーネル(ITRON 系)ではメールボックス機能が廃止されました。

TOPPERS 第 3 世代カーネル(ITRON 系)統合仕様書では、メールボックス機能の代わりとして、データキュー機能を用いてメモリ領域へのポインタを送受信する方法が記載されております。この方法について、初級者向けの教材があるとよいのではと考え、教材を作成しました。

開発環境とアプリケーション(カップラーメンタイマ)仕様は、「名古屋大学情報学研究所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)と同一としました。

なお、この応募作品は、ET ロボコン 2020 で TOPPERS 賞を受賞した弊社の昨年度の新人研修を基に、今年度の新人研修で実施したプロジェクト演習「カップラーメンタイマの改造」を教材として再構成したものです。

アイデア/アプリケーションの概要

私達新入社員は、全員 RTOS 初心者です。新人研修のプロジェクト演習にて、サンプルコードも無い中でメモリプール・データキューや関数ポインタ等、複数の機能を一気に実装した結果、動作させるまで非常に苦勞しました。

この経験を踏まえて、新入社員目線で初心者に分かりやすい教材の作成を検討しました。検討の結果、10 のステップに分けて段階的に理解しながら実装していく教材としました。

教材と共に作成したサンプルプログラムは、NCES TRAINING BOARD(RX63N マイコン搭載)、及び NUCLEO-F401RE(STM32 マイコン搭載)で動作確認しております。

上記 2 種類のマイコンボード向けにそれぞれ cup_timer2~8 までの 7 つのプロジェクトとして作成しており、「組込みソフトウェア開発技術の基礎」の教材ディレクトリ ¥program¥program_asp 下にコピーするだけでビルド可能です。

1.著作権表記

コンテンツ応募に伴い、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.動作環境

2.1.ハードウェアの入手先

次のいずれかのマイコンボードを用意してください。

- ・NCES TRAINING BOARD(ルネサスエレクトロニクス RX63N マイコン搭載)

http://ma2.la.coocan.jp/miconboard/nces_shield/

- ・NUCLEO-F401RE(ST マイクロ STM32 マイコン搭載)※

<https://www.st.com/ja/evaluation-tools/nucleo-f401re.html>

※NUCLEO-F401RE の場合、組込みソフトウェア開発技術の基礎「13.回路作成実習編」の「パーツリスト」に記載のパーツも必要になります。

2.2.開発環境の入手先

開発環境は、「組込みソフトウェア開発技術の基礎」と同一です。次の URL から教材のダウンロードの申し込みができます。

- ・組込みソフトウェア開発技術の基礎（日本語版）

<https://www.nces.i.nagoya-u.ac.jp/NEP/materials/about.html#dltext1>

- ・組込みソフトウェア開発技術の基礎（英語版）

<https://www.nces.i.nagoya-u.ac.jp/NEP/materials/about.html#dltext2>

- ・組込みソフトウェア開発技術の基礎（日本語版）（STM32）

<https://www.nces.i.nagoya-u.ac.jp/NEP/materials/about.html#dltext3>

- ・組込みソフトウェア開発技術の基礎（英語版）（STM32）

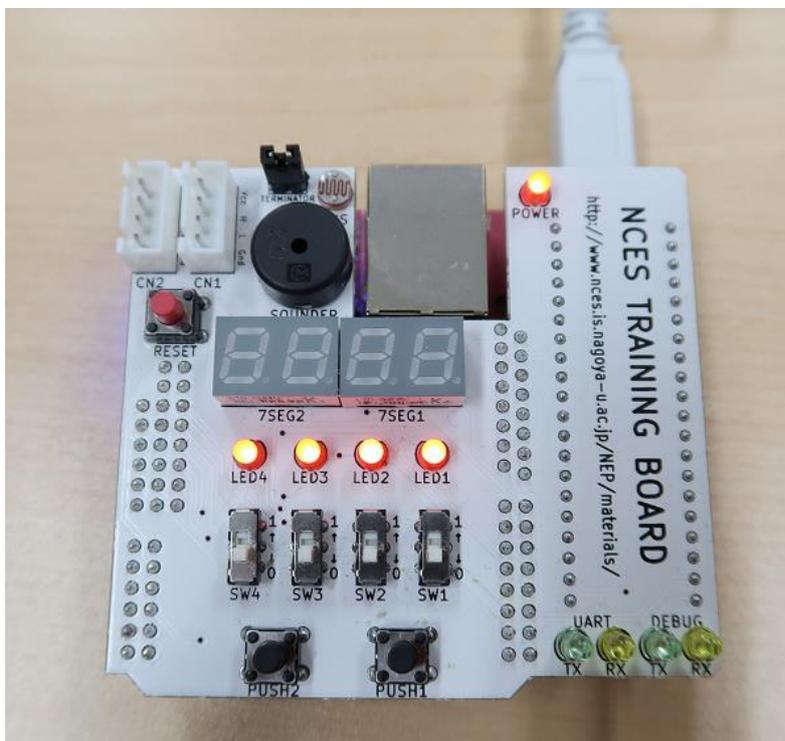
<https://www.nces.i.nagoya-u.ac.jp/NEP/materials/about.html#dltext4>

2.3.開発環境の構築

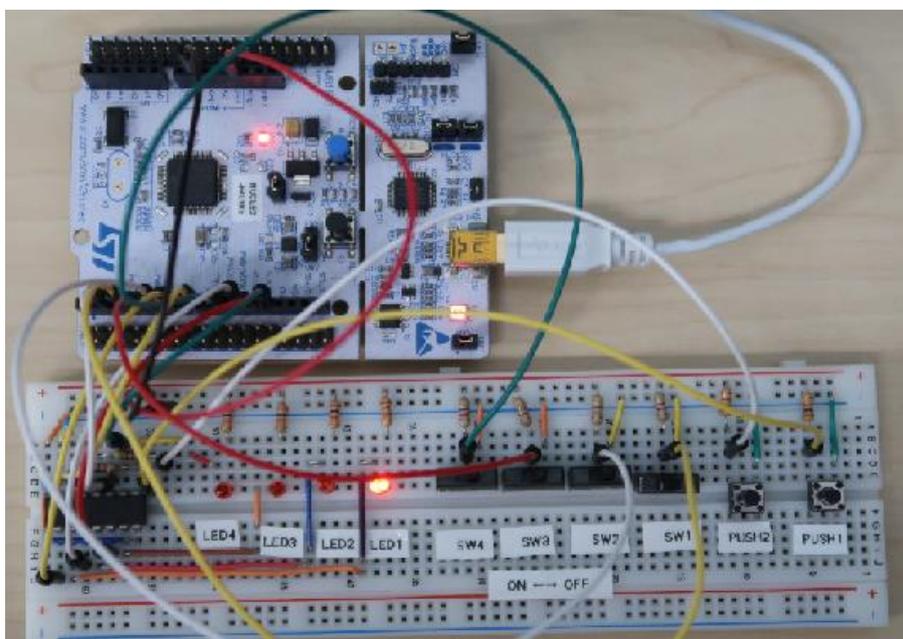
組込みソフトウェア開発技術の基礎「5.開発環境の確認編」に記載の方法により、開発環境の構築が可能です。

3.ハードウェアの外観

- ・ NCES TRAINING BOARD



- ・ NUCLEO-F401RE



4.アプリケーションの仕様

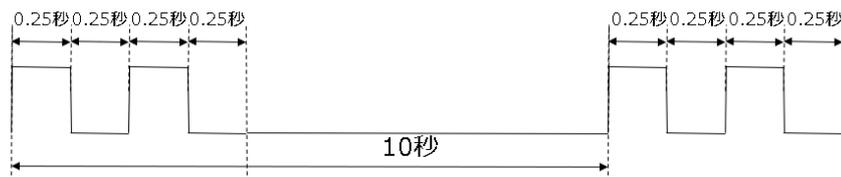
組込みソフトウェア開発技術の基礎「12.ITRON プログラミング実習編」の「カップラーメンタイマの実装 1.仕様」に記載されています。以下に引用いたします。

カップラーメンタイマの仕様

- 概要
 - タイマを起動してから、設定した時間が経過したことを知らせる
 - 設定時間は30秒刻みで設定できる
- スイッチとLEDの使い方
 - LED1 : 電源がONであること
(プログラムが動いていること) を表示
 - LED2 : PUSH1が押されたことを表示
 - LED4 : タイマが動いていることを表示
 - SW1 : タイマを起動・停止
 - PUSH1 : 設定時間を30秒延長

カップラーメンタイマの仕様

- 外部仕様
 - 電源をONの間（プログラムが動いている間）、LED1の点灯・消灯を1秒間隔で繰り返す
 - SW1がONになると設定時間を30秒にして、タイマを起動する
 - SW1がOFFになると、タイマを停止する
 - タイマが動いている間にPUSH1がONになると、ONの間LED2を点灯させ、設定時間を30秒延長する
 - タイマが動いている間は、10秒間隔で、LED4を2回点滅させる。LED4の2回点滅は、点灯・消灯を0.25秒間隔で2回繰り返すことで行う
 - 設定時間が経過すると、LED4を15秒間点滅させた後、消灯する。LED4の点滅は、点灯・消灯を0.25秒間隔で繰り返すことで行う



5.教材の構成

サンプルプログラムと 10Step の段階的学習のスライドです。

詳細は sample.zip と教材スライド.pptx を参照願います。

サンプルプログラム

- 演習で作成するプログラムの解答例になります。
- 添付のsample.zipを解凍し、 5.開発環境の確認編で作成した program¥program_aspフォルダにコピーしてください。

cup_timer2	データキューの実装と状態遷移表のコード化(switch case文)
cup_timer3	状態遷移表のコード化(関数ポインタのテーブル)
cup_timer4	メモリプールの制約と解決策
cup_timer5	メモリプールの実装
cup_timer6	状態遷移の局所化とメッセージ転送機能の実装
cup_timer7	ファイル分割
cup_timer8	状態遷移時のentry/exitアクションの実装

10Stepに分けることで、段階的に学習

- Step0 仕様と状態遷移図を把握する
- Step1 抜け漏れなく改造するために
- Step2 外部I/Fの変更(データキュー)
- Step3 タスク・ハンドラ内部の変更
- Step4 状態遷移表のコード化(switch case文)
- Step5 状態遷移表のコード化(関数ポインタのテーブル)
- Step6 外部I/Fの変更(データキュー+メモリプール)
- Step7 状態遷移の局所化とメモリブロックの転送
- Step8 ファイル分割
- Step9 状態遷移時のentry/exit

以上