



Arduinoからステップアップ！ 電子工作趣味者のための TOPPERS入門



2020年6月27日

NPO法人TOPPERSプロジェクト
堀 武司



TOPPERSプロジェクトとは？

TOPPERS = Toyohashi OPen Patform for
Embedded and Real-time Systems

- プロジェクトの活動内容
 - ITRON仕様の技術開発成果を出発点として、組込みシステム構築の基盤となる各種の高品質なオープンソースソフトウェアを開発するとともに、その利用技術を提供

**組込システム分野において、Linuxのように広く使われる
オープンソースOSの構築を目指す！**

- プロジェクトの推進主体
 - 産学官の団体と個人が参加する産学官民連携プロジェクト
 - 2003年9月にNPO法人として組織化



本日のおしながき

- **対象:** Arduino を使って電子工作している人
(ソフト・マイコン開発初心者の方)
- Arduino の不満を解決！ TOPPERSのRTOS
- 使ってみよう！ TOPPERSの始め方
 - マイコンボードを選ぼう！
 - 開発ツールをインストールしよう！
 - プログラムをビルド・動かしてみよう



Arduino の不満を解決！ TOPPERS のリアルタイムOS



Arduino スケッチ

```
void setup()
{
    // 最初に一度だけ実行する処理（初期化など）
}

void loop()
{
    // 繰り返し何度も実行される処理
}
```



LED チカチカ

```
void setup()
{
    // ポートの初期化
    pinMode(13, OUTPUT);
}
```

一秒ごとに LEDを点滅
とっても簡単！

```
void loop()
{
    digitalWrite(13, HIGH);    // 点灯
    delay(1000);               // 1秒（1000ミリ秒）待つ
    digitalWrite(13, LOW);    // 消灯
    delay(1000);               // 1秒（1000ミリ秒）待つ
}
```



2つのLEDを同時にチカチカ？

```
void setup()
{
    // ポートの初期化
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
}
```

LED 1 を一秒ごとに点滅
LED 2 を二秒ごとに点滅
させたい。どう書けばいい？

```
void loop()
{
    // ?????????????????
}
```



力ずくで書いてみる

```
void loop()
{
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    delay(1000);
    digitalWrite(LED1, LOW);
    delay(1000);

    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, LOW);
    delay(1000);
    digitalWrite(LED1, LOW);
    delay(1000);
}
```

一回のLoop で、

- LED1は2周期点滅
- LED2は1周期点滅

面倒くさい！



カウンタを数えてみる

```
int count = 0;
void loop()
{
    if (count==0) {
        digitalWrite(LED1, HIGH);
        digitalWrite(LED2, HIGH);
    } else (count==1) {
        digitalWrite(LED1, LOW);
    } else (count==2) {
        digitalWrite(LED1, HIGH);
        digitalWrite(LED2, LOW);
        /* . . . 省略 . . . */

        delay(1000);

        count=count+1;
        if (count==4) count=0; // 4までいったら戻る
    }
}
```

とっても面倒くさい！



もっと複雑になったら？

```
int count = 0;
```

果てしなく面倒くさい！

「Arduino の XXライブラリを使えば出来るよ」
「YYタイマ機能を使えば出来るよ」

のような話も色々ありますが……



こう書きたいと思いませんか？

```
void loop()  
{ // LED1 の点滅  
    digitalWrite(LED1, HIGH);  
    delay(1000);  
    digitalWrite(LED1, LOW);  
    delay(1000);  
}
```

```
void loop2()  
{ // LED2 の点滅  
    digitalWrite(LED2, HIGH);  
    delay(2000);  
    digitalWrite(LED2, LOW);  
    delay(2000);  
}
```

すっきり！

2つの処理が「混ざって」いないので、わかりやすい。



そこで TOPPERS!

リアルタイムOSを使うと、本当にこう書けます

- TOPPERS のリアルタイムOSでできること
 - **マルチタスク**: ふたつ以上の仕事（タスク）を別々に書いて、同時に動かす
 - **優先度**: より重要なタスクから先に処理する
 - **タスク同士の通信・待ち合わせ**
 - **周期実行、割り込み**



つかってみよう! TOPPERSの始め方

ふだんArduino を使っている人むけに、最短コースで
TOPPERS を導入する方法を説明します。



マイコンボードを選ぼう

× Arduino Uno など (AVRマイコン)

- 残念ながら動きません (とても小さなマイコンなので)
- まずは、AVR → Arm マイコンにステップアップしてみよう！

◎ Arduino M0, M0 Pro, Zero, etc. (Armマイコン搭載)

- きょう紹介するOS(TOPPERS/R2CA) が動きます。
- これを選びましょう。

△ ↑以外の Arm 搭載 Arduino / Arduino 互換マイコン

- 今日は説明しませんが、頑張れば動きます。

Arduino M0 シリーズ



Arduino M0 Pro

- オンボードデバッグ回路(EDBG)が搭載されていて、プログラム書き込みが楽
- 製造中止、店頭在庫限りです。見かけたら買っておきましょう。(6,000円くらい)

Programming Port



Arduino M0 (Proでない方)

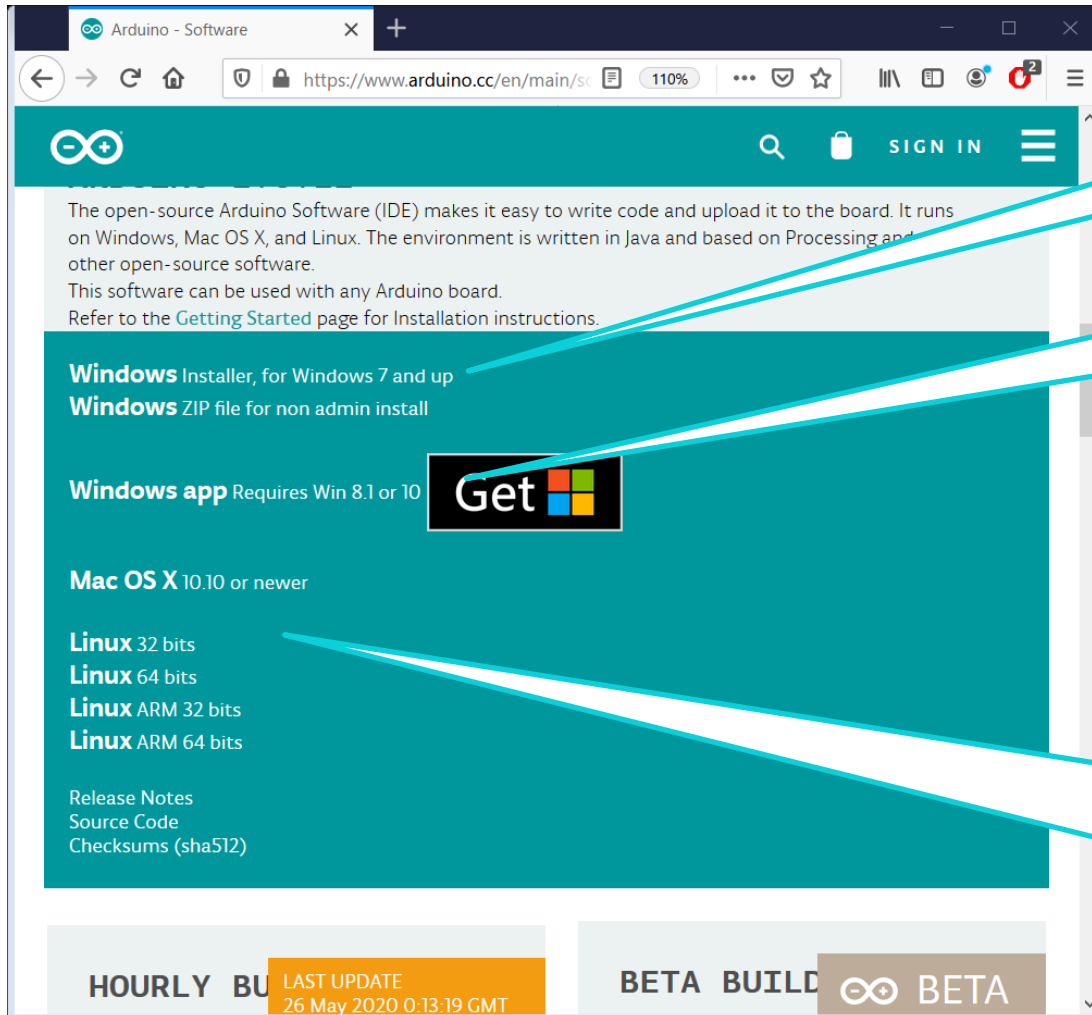
- 普通の電子工作なら、これで十分。
- 今も入手可能 (3,500円くらい)
- 互換製品もあるようです。

Programming Port が無い





Arduino IDE をインストール



Windows インストーラ版を選びましょう。

Windows App版、ZIP版も使えますが、追加で環境設定が必要

Mac OS X, Linux でも動きます。
(自力でなんとか出来る人向け)



ボードマネージャの設定

- Arduino IDE の [ツール]→[ボード]→[ボードマネージャ]を開く
- 2つのボードを追加する

The screenshot shows the Arduino Board Manager window. At the top, there's a search bar and a 'Type' dropdown set to 'All'. Below, there are three sections of boards:

- Arduino SAM Boards (32-bits ARM Cortex-M3)**
by **Arduino** バージョン **1.6.12** **INSTALLED**
このパッケージに含まれているボード:
Arduino Due.
[Online Help](#)
[More Info](#)
- Arduino SAMD Boards (32-bits ARM Cortex-M0+)**
by **Arduino** バージョン **1.8.4** **INSTALLED**
このパッケージに含まれているボード:
Arduino MKR WiFi 1010, Arduino Zero, Arduino MKR1000, Arduino MKRZERO, Arduino MKR FOX 1200, Arduino MKR WAN 1300, Arduino MKR WAN 1310, Arduino MKR GSM 1400, Arduino MKR NB 1500, Arduino MKR Vidor 4000, Arduino Nano 33 IoT, Arduino M0 Pro, Arduino M0, Arduino Tian, Adafruit Circuit Playground Express.
[Online Help](#)
[More Info](#)
- Arduino SAMD Beta Boards (32-bits ARM Cortex-M0+)**
by **Arduino**
このパッケージに含まれているボード:

Two callout boxes provide additional context:

- A box pointing to the top section states: "今回使うボード用ではないですが、こっちのパッケージも使うのでインストール (※ 別途 Cygwin, MinGW などを入れている人は不要かも)"
- A box pointing to the middle section states: "今回使うボード(Arduino M0) 用のパッケージ"

TOPPERS/R2CA



https://dev.toppers.jp/trac_user/contrib/wiki/rtos_arduino

- Arduino マイコンで、TOPPERS の OSを「Arduinoっぽい」感覚で簡単に使える。
 - setup()/loop()を複数書ける。
 - リアルタイムOS (TOPPERS/ASP カーネル) の機能が使える。
- Arduino スケッチ、ライブラリが、だいたいそのまま動く。

TOPPERS/R2CA Compatible with Arduino

◆TOPPERS RTE/RTOS Compatible with Arduino : R2CA とは？

- Arduino/ハードウェア上でTOPPERS/ASPカーネルを実行
- 組み込みソフトウェア開発のエントリからのステップアップ用

◆Arduino

- ラビッドプロトタイピング環境
- メリット
 - シンプルなIDEとスケッチによるプログラミング
 - Web/書籍等の情報が豊富
 - 豊富なライブラリ・シールド
- デメリット
 - シングルタスク
 - デバッグが用意されていない

→ 拡張

◆R2CA

- ASPによるマルチタスク環境
- ASPのAPIを発行可能
- スケッチとの互換性
- Arduinoライブラリを使用可能
- 細かいツールの設定は必要無し
 - Arduino IDEでインストールされるツールを使用
- 無償のソースコードデバッグをサポート(Atmel Studio)

◆デモ構成

- 6個のタスクにより構成
- ライブラリはArduino向を少し変更で使用
- 処理のみに注力して開発が可能

Arduino M0 Pro

- Cortex-M0+ 48MHz
- ROM 256KB
- RAM 32KB

◆プログラム例

```
void setup() {  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);  
  delay(1000);  
  digitalWrite(13, LOW);  
  delay(1000);  
}  
  
void task1_setup() {  
  Serial.begin(9600);  
}  
  
void task1_loop() {  
  int sv = analogRead(A0);  
  Serial.println(sv);  
  delay(1);  
}
```

TOPPERS
Toyohashi Open Platform for Embedded Real-time Systems



TOPPERS/R2CA をダウンロード

- 公式配布版
 - https://dev.toppers.jp/trac_user/contrib/attachment/wiki/rtos_arduino/
 - 少し古い（最終更新 2017年）ので、今の Arduino IDE ではそのまま動きません。
 - Arduino M0（Proでない）で動かす場合も、工夫が必要です。
- アップデート版（私が修正を入れた非公式版）
 - https://github.com/hori-takeshi/rtos_arduino/
 - 最新版の Arduino IDE で動くように調整しました。
 - Arduino M0（Proでない）でも簡単に動きます。

ここからダウンロードして下さい



https://github.com/hori-takeshi/rtos_arduino/releases/tag/osc-nagoya-20200530

The screenshot shows the GitHub release page for the repository `hori-takeshi/rtos_arduino`. The release is titled "TOPPERS/R2CA for Arduino 1.8.x OSC2020 Nagoya Online" and is marked as a "Pre-release". The release was made by `hori-takeshi` 3 minutes ago. The page lists three assets: `rtos_arduino-20200530-osc-nagoya.zip` (13.5 MB), `Source code (zip)`, and `Source code (tar.gz)`. The page also includes a "Compare" button and a "Releases" tab.

名前の通り、先月のOSC Online/Nagoya 向けに急遽つくりました。

TOPPERS/R2CA を展開する



File Explorer window showing the directory structure of the TOPPERS/R2CA project.

Address bar: C:\Users\hori\Documents\Projects\TOPPERS\rtos_arduino

Navigation pane (left):

- rtos_arduino
- .git
- .svn
- arduino_lib
- asp_1.9.2
- examples
- lib
- SSP3
- temp_work
- toppers-base-st
- TPN

Main pane (right):

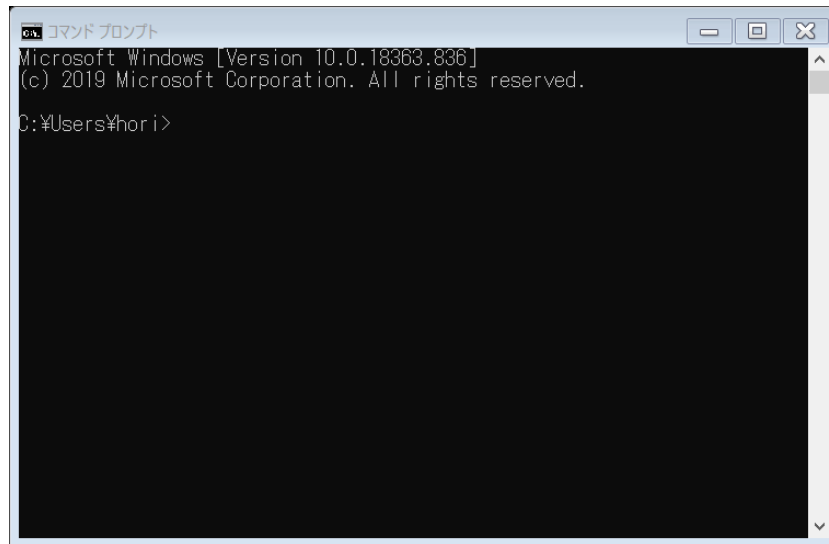
名前	更新日時	種類	サイズ
.git	2020/05/26 8:33	ファイル	
.svn	2020/05/19 11:23	ファイル	
arduino_lib	2020/05/19 11:23	ファイル フォルダー	
asp_1.9.2	2020/05/19 11:23	ファイル フォルダー	
examples	2020/05/19 19:16	ファイル フォルダー	
lib	2020/05/19 18:36	ファイル フォルダー	
.gitignore	2020/05/19 11:31	GITIGNORE ファイル	1 KB
arduino-1-8.txt	2020/05/27 15:19	テキスト ドキュメント	1 KB
README.md	2020/05/20 20:44	MD ファイル	2 KB
README.txt	2020/05/19 11:23	テキスト ドキュメント	13 KB

Annotation: A red box highlights the 'examples' folder, with a red arrow pointing to it from the text: 例題プログラムの置き場所 (自分のスケッチもここに置く)



コマンドプロンプトを開く

- Arduino IDE だけでラクラク開発……と言いましたが、スケッチのビルド・実行には IDE を**使いません**。
- 代わりに「**コマンドプロンプト**」から操作します。



- ~~6月末ころ（OSC Hokkaido）には、全ての作業を Arduino IDE でできるようにします。~~



サンプルをビルドしてみる

rtos_Arduino¥Examples¥<<サンプル名>>
に移動 する。

コマンド プロンプト

```
C:¥Users¥hori¥Documents¥Projects¥TOPPERS>cd rtos_arduino
C:¥Users¥hori¥Documents¥Projects¥TOPPERS¥rtos_arduino>cd examples¥MultiTask
C:¥Users¥hori¥Documents¥Projects¥TOPPERS¥rtos_arduino¥examples¥MultiTask>do_make
```

do_make.bat

もしくは
make
(事前に examples¥do_path.bat
を実行しておく)



ビルド完了

```
コマンド プロンプト
WString.o IPAddress.o Tone.o target_serial.o banner.o syslog.o kernel_cfg.
o libkernel.a -lc -lgcc -lm -lstdc++
"C:\Users\hori\AppData\Local\Arduino15\packages\arduino\tools\arm-none-eabi-gcc\7
-2017q4\bin\arm-none-eabi-nm" -n r2ca.elf > r2ca.syms
"C:\Users\hori\AppData\Local\Arduino15\packages\arduino\tools\arm-none-eabi-gcc\7
-2017q4\bin\arm-none-eabi-objcopy" -O srec -S r2ca.elf r2ca.srec
"../../asp_1.9.2/cfg/cfg/cfg" --pass 3 --kernel asp -I. -I../../asp_1.9.2/include
-I../../asp_1.9.2/arch -I../../asp_1.9.2 -I"../../arduino/lib/hardware/tools/CMS
IS/CMSIS/Inc/Include/" -I"../../arduino/lib/hardware/tools/CMSIS/Device/ATMEL/" -I"
../../arduino/lib/hardware/arduino/samd/cores/arduino/" -I"../../arduino/lib/hardwa
re/arduino/samd/variants/arduino_zero/" -I/lib -I../../asp_1.9.2/target/arduino_m
0_gcc -I../../asp_1.9.2/arch/arm_m_gcc/common -I../../asp_1.9.2/arch/gcc -I../../
lib -I../../arduino/lib/hardware/arduino/samd/cores/arduino/ -I../../arduino/lib/
hardware/arduino/samd/cores/arduino/avr -I../../arduino/lib/hardware/arduino/samd
/cores/arduino/USB -I../../arduino/lib/hardware/arduino/samd/variants/arduino_zer
o/ -I../../asp_1.9.2/library ¥
--rom-image r2ca.srec --symbol-table r2ca.syms ¥
-T ../../asp_1.9.2/target/arduino_m0_gcc/target_c
heck.tf --api-table ../../asp_1.9.2/kernel/kernel_api.csv --cfg1-def-table ../../
asp_1.9.2/kernel/kernel_def.csv --cfg1-def-table ../../asp_1.9.2/target/arduino_
m0_gcc/target_def.csv --cfg1-def-table ../../asp_1.9.2/arch/arm_m_gcc/common/core
_def.csv r2ca_app.cfg
check complete

C:\Users\hori\Documents\Projects\TOPPERS\rtos_arduino\examples\MultiTask>
```

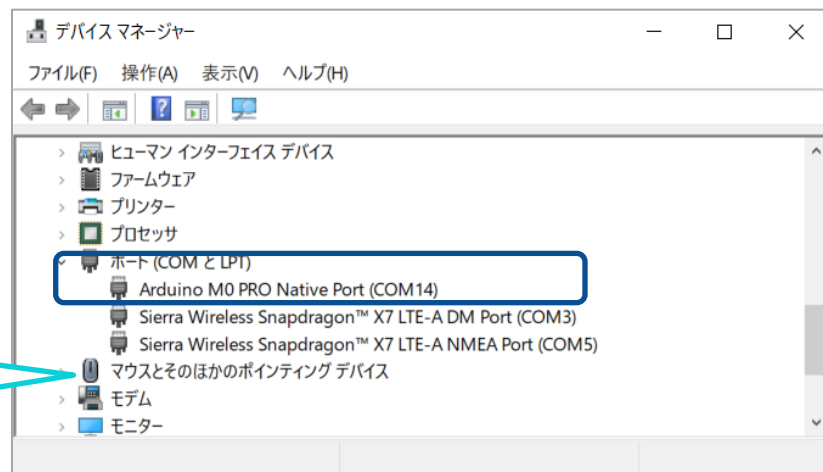
こうなったらビルド成功

プログラムの動かす(Arduino M0)



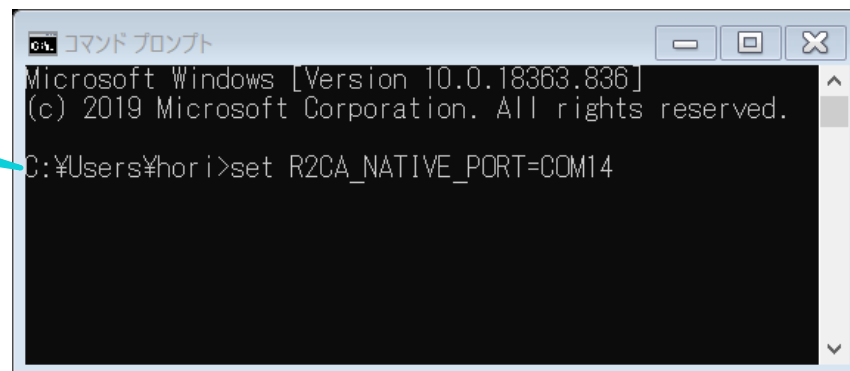
- マイコンのシリアルポートを探す

デバイス マネージャ を起動し、
ポート (COMとLPT) から、
“Arduino M0 Pro Native Port”を
探す



- ポートの名前を環境変数 R2CA_NATIVE_PORT に設定する

set R2CA_NATIVE_PORT=**COM14**



プログラムを動かす(Arduino M0)



```
コマンド プロンプト
C:\Users\hori\Documents\Projects\TOPPERS\rtos_arduino\examples\MultiTask>make run-native
C:\Users\hori\AppData\Local\Arduino15\packages\arduino\tools\avrdude\6.3.0-arduino17\bin\avrdude.exe -p -P COM7 -C C:\Users\hori\AppData\Local\Arduino15\packages\arduino\tools\avrdude\6.3.0-arduino17\etc\avrdude.conf -v -patmega2560 -cstk500v2 -P COM7 -b57600 -Uflash:w:./r2ca.srec

arduino\tools\avrdude\6.3.0-arduino17\bin\avrdude.exe: Version 6.3-20190619
Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com
Copyright (c) 2007-2014 Joerg Wunsch

System wide configuration file is "C:\Users\hori\AppData\Local\Arduino15\packages\arduino\tools\avrdude\6.3.0-arduino17\etc\avrdude.conf"

Using Port                : COM7
Using Programmer           : stk500v2
Overriding Baud Rate      : 57600
AVR Part                   : ATmega2560
Chip Erase delay           : 9000 us
PAGEL                      : PD7
BS2                        : PA0
RESET disposition          : dedicated
RETRY pulse                : SCK
serial program mode        : yes
parallel program mode      : yes
Timeout                   : 200
StabDelay                  : 100
CmdexeDelay                : 25
SyncLoops                  : 32
```

make run-native

```
arduino\tools\avrdude\6.3.0-arduino17\bin\avrdude.exe: load data flash data from input file ./r2ca.srec:
arduino\tools\avrdude\6.3.0-arduino17\bin\avrdude.exe: input file ./r2ca.srec auto detected as Motorola S-Record
arduino\tools\avrdude\6.3.0-arduino17\bin\avrdude.exe: input file ./r2ca.srec contains 38728 bytes
arduino\tools\avrdude\6.3.0-arduino17\bin\avrdude.exe: reading on-chip flash data:

Reading | ##### | 100% 0.53s

arduino\tools\avrdude\6.3.0-arduino17\bin\avrdude.exe: verifying ...
arduino\tools\avrdude\6.3.0-arduino17\bin\avrdude.exe: 38728 bytes of flash verified

arduino\tools\avrdude\6.3.0-arduino17\bin\avrdude.exe: safemode: lfuse reads as 0
arduino\tools\avrdude\6.3.0-arduino17\bin\avrdude.exe: safemode: hfuse reads as 0
arduino\tools\avrdude\6.3.0-arduino17\bin\avrdude.exe: safemode: efuse reads as 0
arduino\tools\avrdude\6.3.0-arduino17\bin\avrdude.exe: safemode: Fuses OK (E:00, H:00, L:00)

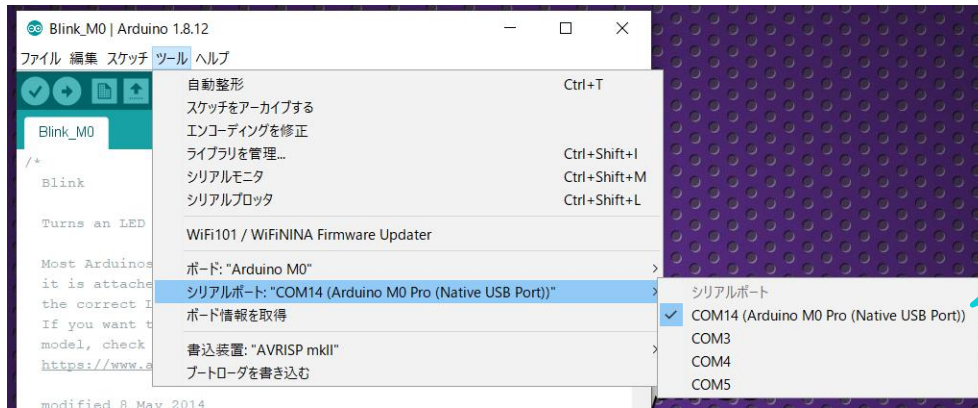
arduino\tools\avrdude\6.3.0-arduino17\bin\avrdude.exe done. Thank you.

C:\Users\hori\Documents\Projects\TOPPERS\rtos_arduino\examples\MultiTask>
C:\Users\hori\Documents\Projects\TOPPERS\rtos_arduino\examples\MultiTask>
```

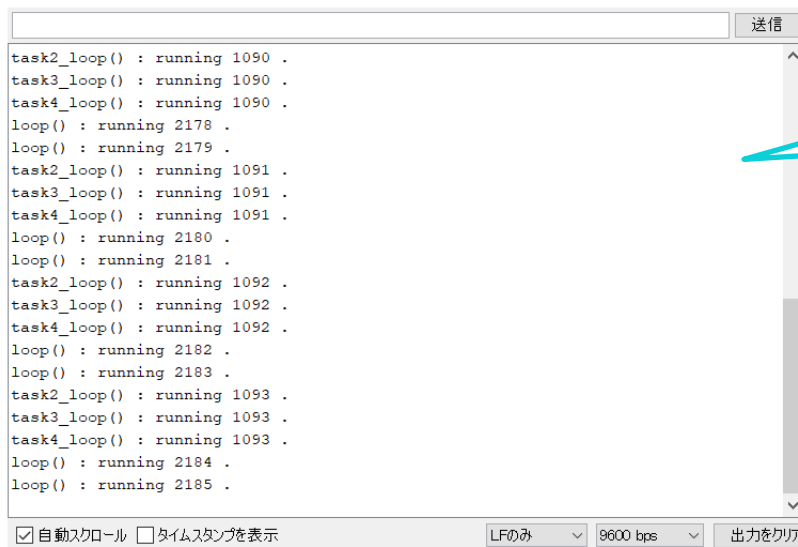
ここまでくれば、書き込み完了



コンソール出力を見てみよう



Arduino IDEを起動し、
Arduino M0 のCOMポートを選択



「シリアルモニタ」を開く

※ 他のターミナルソフト (Tera Term など)を使ってもよい



examples/MultiTask (1)

```
#include "r2ca.h"          // おまじない
#define Serial SerialUSB  // Arduino M0 の場合、Serial の代わりに SerialUSB を使う

/*
 * For MAINTASK
 */
void setup() {
    Serial.begin(115200);
    Serial.println("setup() : running ");

    pinMode(13, OUTPUT);
}

int loop_cnt = 0;

void loop() {
    Serial.print("loop() : running ");
    Serial.print(loop_cnt++);
    Serial.println(" .");
    delay(1000);
}
```

普通の Arduino と同じ初期化処理

メインタスク:
一秒間隔でカウント & 表示



examples/MultiTask (2)

```
/*  
 * For TASK1  
 */  
void loop1() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

タスク1:
一秒間隔で、LED を点滅させる



examples/MultiTask (3)

```
int task2_loop_cnt = 0;

/*
 * For TASK2
 */
void loop2() {
    Serial.print("task2_loop() : running ");
    Serial.print(task2_loop_cnt++);
    Serial.println(" .");
    delay(2000);
}
```

タスク2:
2秒間隔で、カウント&表示
(タスク3, 4も同じ)



コンソール出力

```
loop() : running 1 .  
task3_loop() : running 1 .  
task2_loop() : running 1 .  
task4_loop() : running 1 .  
loop() : running 2 .  
loop() : running 3 .  
task2_loop() : running 2 .  
task3_loop() : running 2 .  
task4_loop() : running 2 .  
loop() : running 4 .  
loop() : running 5 .  
task2_loop() : running 3 .  
task3_loop() : running 3 .  
task4_loop() : running 3 .
```

```
loop() : running 6 .  
loop() : running 7 .  
task2_loop() : running 4 .  
task3_loop() : running 4 .  
task4_loop() : running 4 .  
loop() : running 8 .  
loop() : running 9 .  
task2_loop() : running 5 .  
task3_loop() : running 5 .  
task4_loop() : running 5 .  
loop() : running 10 .
```



自分のスケッチをつくるには？

- examples/MultiTask フォルダをまるごとコピーして、examples/MySketch をつくりましょう。
- あとは、サンプルコードを適当に書き換えていけばOK!
 - r2ca_app.cpp - スケッチ
 - r2ca_app.cfg - OS設定（コンフィギュレーション）



さらに勉強するには？

- Qiita記事: Arduino M0 Pro でマルチタスクプログラミング
 - <https://qiita.com/tags/R2CA>
 - R2CA を開発した本田先生(名古屋大学:当時 @exshonda) による解説記事
 - さらに高度な リアルタイムOSの使い方
- TOPPERS Project
 - <https://www.toppers.jp/>
 - TOPPERS/R2CA 以外にも様々な OS、ミドルウェアがあります。

TOPPERSで楽しく電子工作



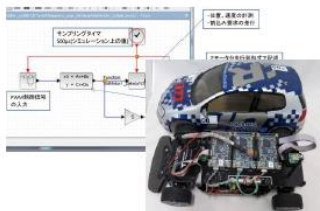


そして楽しい作品ができたら……

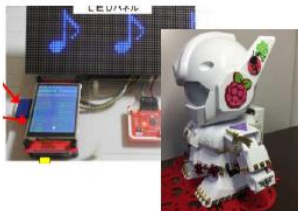
第10回 TOPPERS活用アイデア・アプリケーション開発コンテスト

アプリケーション開発部門

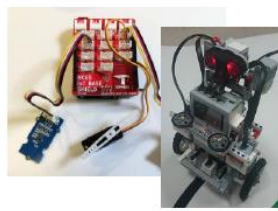
TOPPERS開発成果物を利用した
楽しい・独創的なアプリケーション
実用的なアプリケーション
わかりやすいサンプルアプリケーション
を募集します



実用的なアプリケーション
や開発支援ツールから



たのしい
趣味の電子工作まで！



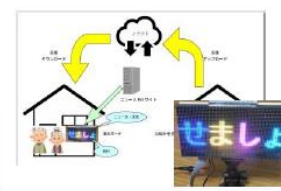
Arduino on TOPPERS
TOPPERS/EV3RTも
こちらの開発部門へ！

活用アイデア部門

TOPPERS開発成果物の利用者にとって
有益な活用アイデアや
TOPPERSをより良くするアイデア
を募集します（アイデアだけでもOK！）



TOPPERSの教材開発、
ビルドツールや文書管理
の改善案なども！



構想段階の企画案や
ソースコード非公開作品
でもOK

応募資格：国内に在住の方（グループ、法人の場合にはその代表者）なら誰でも
（※TOPPERSプロジェクトの会員でない方もご応募頂けます）

応募期間：**2020年6月12日（金）～ 2020年8月31日（月）**

表彰：＜活用アイデア部門＞
＜アプリケーション部門＞
＜学生賞＞

金賞 銀賞 銅賞 各1万円！
金賞 5万円！！ 銀賞 3万円！！ 銅賞 1万円！！
学生の応募作品の中から優秀なものに学生賞（1万円）を授与！

協賛社からの
特別賞も予定
しています

学生や組み
初心者からベテラン
エンジニアまで、
幅広い応募をお待ち
しています！



募集要項、応募規約、FAQ、技術講習会の案内等は以下のページで公開していきます！

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

TOPPERS
Toyohashi Open Platform for Embedded Real-time Systems

前回コンテスト(第9回) 受賞作品の紹介

活用アイデア部門: 金賞

クラウドIDEとエミュレータを利用した実機レス開発環境

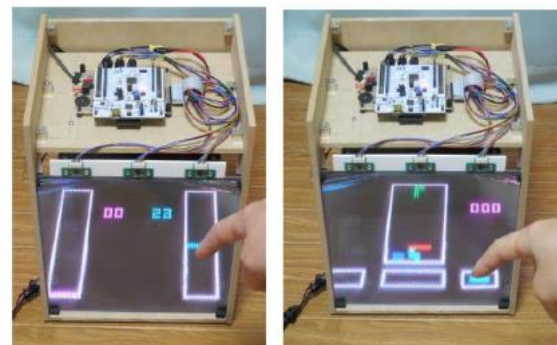
庭野正義 (アイコムシステック株式会社)



アプリケーション開発部門: 銀賞

3Dパネルを使った非接触操作デモ機

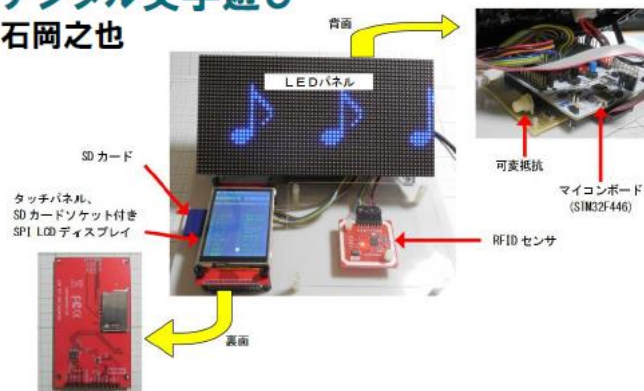
木下浩彰 (ファルコン電子株式会社)



アプリケーション開発部門: 銀賞

デジタル文字遊び

石岡之也

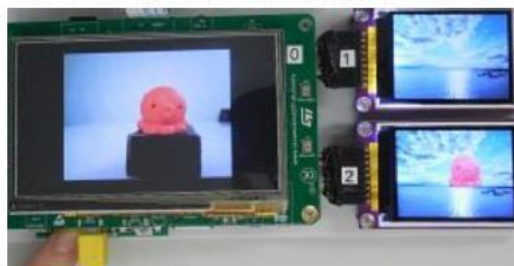


アプリケーション開発部門: 銀賞

ワンチップマイコンによる画像処理システム

羽佐田 理恵

(FCT組込みシステム研究サークル)



アプリケーション開発部門: 銀賞

TOPPERS Configuration Generator

嶋原一人 (富士ソフト株式会社)



第10回 TOPPERS活用アイデア・アプリケーション開発コンテスト

主催: NPO法人TOPPERSプロジェクト

協賛: CQ出版社、(株) アフレル

ウェブサイト: <http://www.toppers.jp/contest.html>

Facebook <http://www.facebook.com/toppersproject>



TOPPERS

ありがとうございました