

Toppers-Contest2017

部門 : がじえるねIoT部門

作品のタイトル :

初心者向け、プログラムを同時に動かすとは



チーム名 : ひぐらし

発表者の紹介

森脇 秀樹

- 概歴 還暦を迎えた、78系が大好き
自称システムエンジニア

角田 米弘

- 概歴 ARMがソフトバンクに買収され
ショックを受けているTK-80で育ったおじさん

概要

コンピュータは、プログラムを順次に、
高速に、実行するように設計された道具ですが

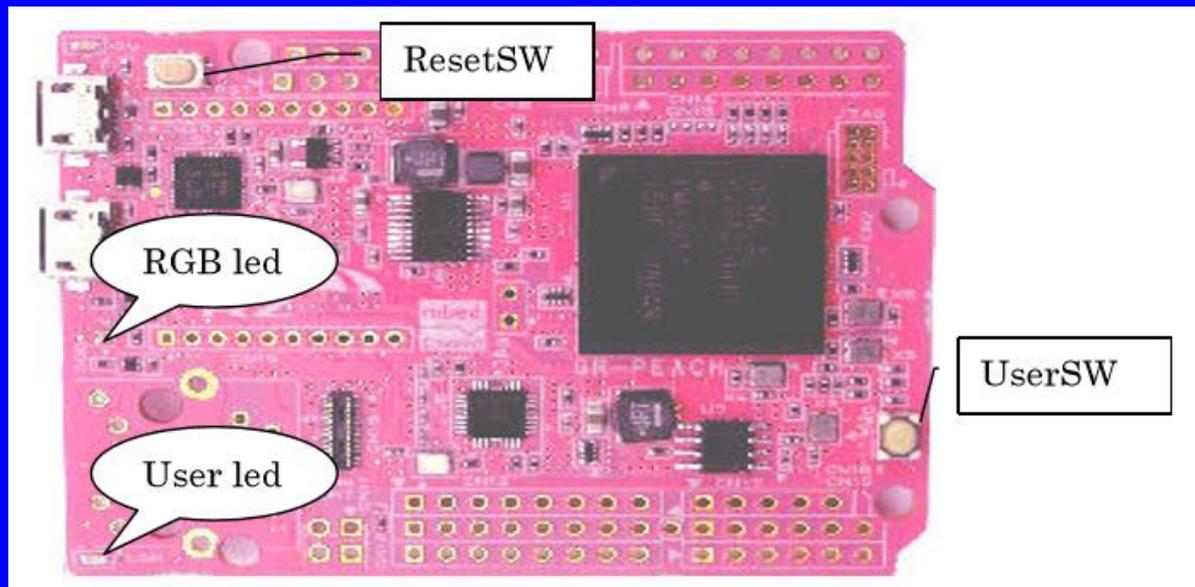
同時に違う動作のプログラムを動作させる
ためには、
どのように考えればよいのでしょうか？

使用ボード GR-PARCH

GR-PARCHには、いろいろな機能が満載ですが
今回は、SWとLEDを使用します。

SWと内部のカウンタ(タイマー)で、時間制御を行い
各LEDを点滅します。

プログラムの中で使用する名前は
自由に付けることができますが、次のように付けて
ありますので、これに従います。



接続機能端子 入力

UserSWはポート6の0に入力として
接続されています。

押すと'0'が読めます。

(普段は'1'です)

プログラムの中では、UserSW を PIN_SW
と呼んでいます。

接続機能端子 出力

User ledは ポート6の12に出力として、

RGB ledのそれぞれの色は

R(赤)ポート6の13の出力として

G(緑)ポート6の14の出力として

B(青)ポート6の15の出力として

接続されています。

'0'で消灯、'1'で点灯します。

接続機能端子 出力呼び名

プログラムの中では、
User led を PIN_USER_LED

RGB ledの赤 を PIN_LED_RED
RGB ledの緑 を PIN_LED_GREEN
RGB ledの青 を PIN_LED_BLUE
と呼んでいます。

サンプルプログラムsp1

がじえるねページのアイテムのGR-PEACH下方に用意されている
(<http://gadget.renesas.com/ja/product/peach.html>)、
プログラム作成 特設ページ
Arduinoライクなスケッチ peach_sp1.html

<http://gadget.renesas.com/ja/sketch>
スケッチを実行してください



GR-PEACH

1. Webコンパイラでスケッチ(Win/Mac)
2. LED、ボタンで遊ぶ！
3. SDカードを使ってロガーを作る！
- 4a. Webサーバー
その1 LEDを遠隔操作！
- 4b. Webサーバー
その2 センサーをグラフ表示！
- 4c. Webサーバー
その3 Webカメラを作る！
5. TOPPERS/ASPでマルチタスク！

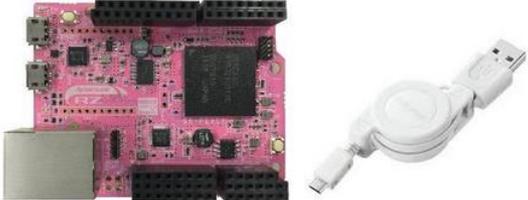
GR-PEACH 特設 Webコンパイラでスケッチ

概要

GR-PEACHのスケッチ(プログラム)作成をWebコンパイラで行う方法を紹介します。WindowsとMacで利用可能です。

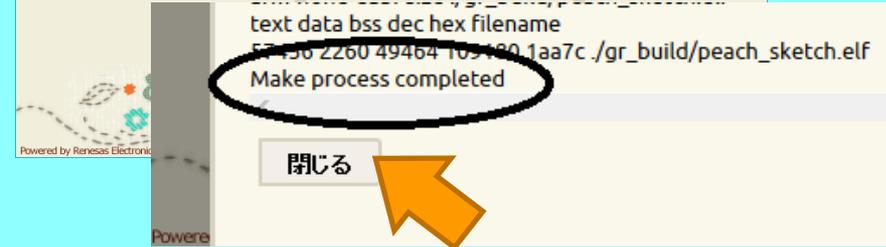
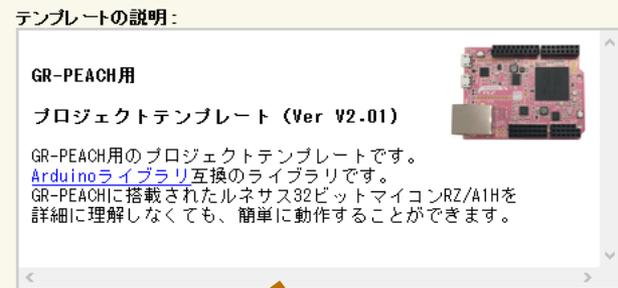
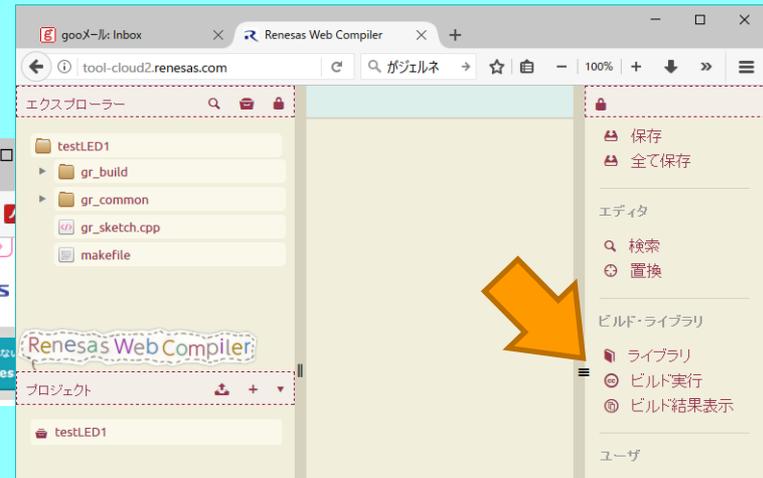
準備

GR-PEACH、USBケーブル(マイクロBタイプ)の2つを準備します。



(スケッチの実行方法)

がじえるねホームページにて <http://gadget.renesas.com/ja/>



プログラム説明

```
/* GR-PEACH Sketch Template V2.00 */
```

```
#include <Arduino.h>
```

```
#define INTERVAL 100
```

```
void setup()
```

```
{  
  pinMode(PIN_LED_RED , OUTPUT);  
  pinMode(PIN_LED_GREEN ,  
    OUTPUT);  
  pinMode(PIN_LED_BLUE , OUTPUT);  
  pinMode(PIN_LED_USER , OUTPUT);  
  pinMode(PIN_SW      , INPUT);  
}
```

```
void loop()
```

```
{  
  while(digitalRead(PIN_SW) == 0){  
    digitalWrite(PIN_LED_USER, 1);  
    delay(INTERVAL);  
    digitalWrite(PIN_LED_USER, 0);  
    delay(INTERVAL);  
  }  
  digitalWrite(PIN_LED_RED, 1);  
  delay(INTERVAL);  
  digitalWrite(PIN_LED_RED, 0);  
  digitalWrite(PIN_LED_GREEN, 1);  
  delay(INTERVAL);  
  digitalWrite(PIN_LED_GREEN, 0);  
  digitalWrite(PIN_LED_BLUE, 1);  
  delay(INTERVAL);  
  digitalWrite(PIN_LED_BLUE, 0);  
}
```

①

②

プログラム変更

プログラムの作りを少し変更して、while を if にする修正して、コンパイルして、実行してみてください。

```
14 }  
15  
16 void loop()  
17 {  
18     if(digitalRead(PIN_SW) == 0){  
19         digitalWrite(PIN_LED_USER, 1);  
20         delay(INTERVAL);  
21         digitalWrite(PIN_LED_USER, 0);  
22         delay(INTERVAL);  
23     }
```

よく動きを、観察して
みてください。
点滅間隔が、息継ぎ
をしていませんか？

コーヒーブレイク？

コンピュータは、本来プログラムで記述されている命令を順番に、1つずつ実行する機械です。

順番に実行するのですから、多数の事を同時には行いません。

ただし命令実行速度が、高速な場合、外部から見ていると、同時に多数の処理をこなしているように見えます。

作り方

なにかないかなー

ありました！！！！

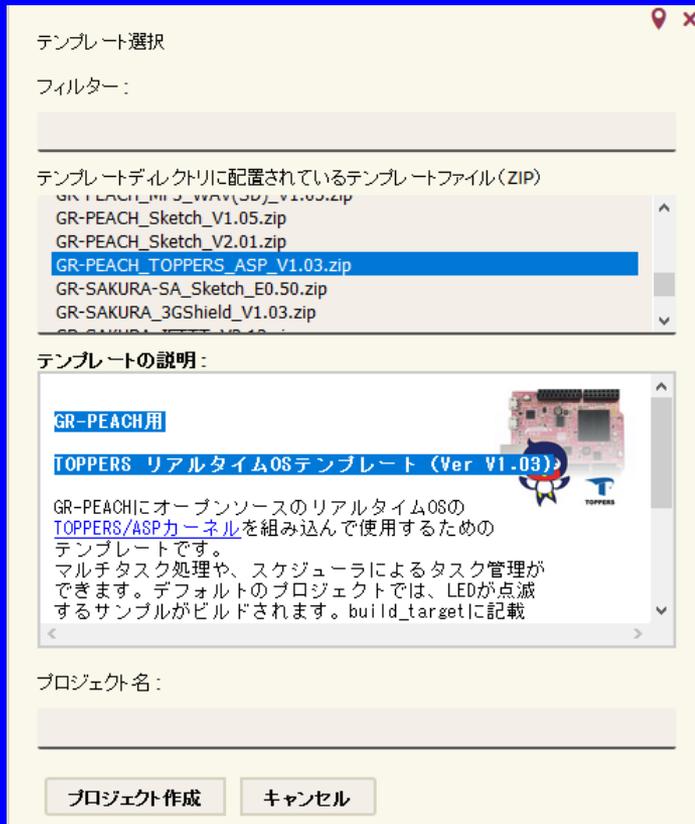
GR-PEACH用

TOPPERS リアルタイムOS

テンプレート (Ver V1.03)

http://gadget.renesas.com/ja/product/peach_sp5.html

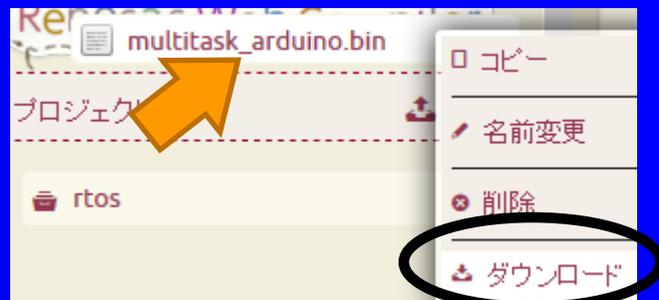
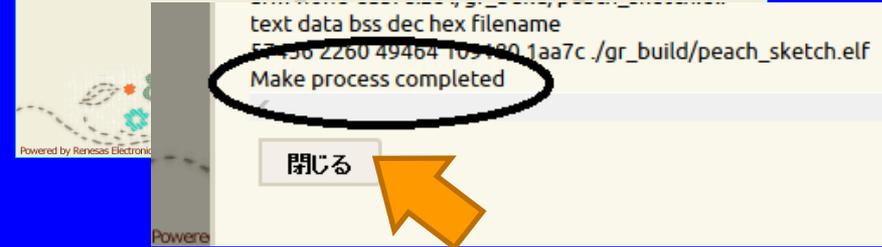
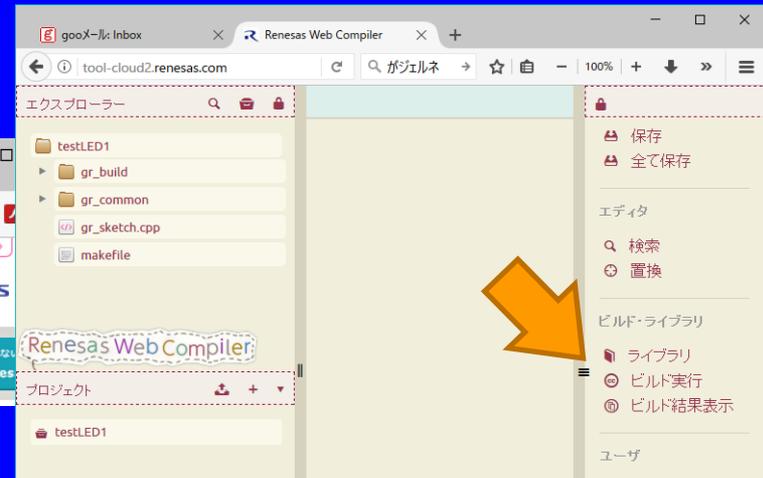
サンプルプログラム



Renesas WebCompilerの
GR-PEACH用
TOPPERS リアルタイムOS
テンプレート (Ver V1.03) を選んで
マルチタスクなArduinoスケッチ
を参考にして
動作させてみてください。

(スケッチの実行方法)

がじえるねホームページにて <http://gadget.renesas.com/ja/>



プログラム修正

```
#include <kernel.h>
#include <t_syslog.h>
#include <t_stdlib.h>
#include "syssvc/serial.h"
#include "syssvc/syslog.h"
#include "kernel_cfg.h"

#include "arduino_app.h"
#include "arduino_main.h"

/* GR-PEACH Sketch Template V1.00 */
#include <Arduino.h>

#define INTERVAL    100
#define INTERVAL_RED  100
#define INTERVAL_BLUE 150
#define INTERVAL_GREEN 200

void setup()
{
    pinMode(PIN_LED_RED , OUTPUT);
    pinMode(PIN_LED_GREEN , OUTPUT);
    pinMode(PIN_LED_BLUE , OUTPUT);
    pinMode(PIN_LED_USER , OUTPUT);
    pinMode(PIN_SW      , INPUT);
}

void loop()
{
    while(digitalRead(PIN_SW) == 0){
        digitalWrite(PIN_LED_USER, 1);
        delay(INTERVAL);
        digitalWrite(PIN_LED_USER, 0);
        delay(INTERVAL);
    }
}

void loop1(){
    digitalWrite(PIN_LED_BLUE, 1);
    delay(INTERVAL_BLUE);
    digitalWrite(PIN_LED_BLUE, 0);
    delay(INTERVAL_BLUE);
    digitalWrite(PIN_LED_RED, 1);
    delay(INTERVAL_RED);
    digitalWrite(PIN_LED_RED, 0);
    delay(INTERVAL_RED);
    digitalWrite(PIN_LED_GREEN, 1);
    delay(INTERVAL_GREEN);
    digitalWrite(PIN_LED_GREEN, 0);
    delay(INTERVAL_GREEN);
}

void loop2(){
    /*
     * Cyclic Handler
     *
     * This handler is called every 10 [ms] as specified in
     * multitask_arduino.cfg.
     */
    void cyclic_handler(intptr_t exinf) {
        irot_rdq(LOOP_PRI); /* change the running loop. */
    }
}
}
```

まとめ

いかがでしたか、簡単に、同時に別々の動作を行わせるプログラム記述を行えたことでしょうか

マルチタスクに興味が出てきたら
理論である
TOPPERS
を学んでみてください



御清聴ありがとうございます

