

計測・制御・双方向通信教材

型番

オーロラロック3

UD-1/UD-2

文字入力プログラム ワーク集 (Webアプリ版)



HISATOMI
久富電機産業株式会社

〒720-0003 広島県福山市御幸町森脇989
TEL: 084-955-6889 FAX: 084-955-1551
URL: <http://www.hisatomi-kk.com>
e-mail: info@hisatomi-kk.com

【禁転載】

年 組 番

名前

①パソコンと接続



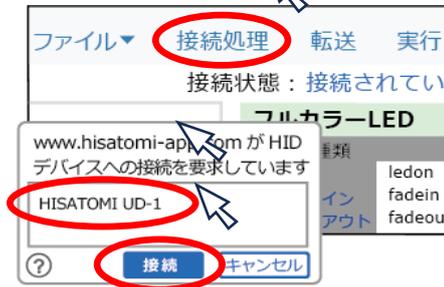
②QRコードで Webアプリを開く (機種を選択する)



③計測・制御の「文字」を選択



④「接続処理」、「UD-1」、「接続」をクリック



⑤左の入力画面でキーボードからプログラムを入力

点線で囲まれたプログラム画面で、命令の文字入力をキーボードから行います。命令の最後の; (セミコロン) の後でエンターキー (改行キー) を押すと2列目のプログラムが入力できます。



関数名 (R値 , G値 , B値 , 時間);

関数名 (半角) フルカラーLED を点灯させる命令	時間 0~31.75秒までの 0.25秒間隔の数値(半角)
-----------------------------------------	--------------------------------------------

ledon (255 , 0 , 0 , 1) ;

R値(赤) 0~255の 数値(半角)	R値(赤) 0~255の 数値(半角)	R値(赤) 0~255の 数値(半角)
----------------------------------	----------------------------------	----------------------------------

プログラム入力欄に必ず「半角英数」で全ての文字・数字・記号を間違えないように入力して下さい。

命令や数字、記号の間にスペースは入れないでください。

R値、G値、B値は0から255まで
時間は、0.25秒間隔で31.75秒まで入力できます。



No.2

年

組

名前

点灯(フルカラーLED)

ledon(数値,数値,数値,数値);

関数名 (R値 , G値 , B値 , 時間);

関数名 (半角)

フルカラーLED
を点灯させる命令

時間

0~31.75秒までの
0.25秒間隔の数値(半角)

ledon (255 , 0 , 0 , 1);

R値(赤)

0~255(半角)

G値(緑)

0~255(半角)

B値(青)

0~255(半角)

フェードイン(フルカラーLED)

fadein(数値,数値,数値,数値);

関数名 (R値 , G値 , B値 , 時間);

関数名 (半角)

フルカラーLED
を点灯させる命令

時間

0~31.75秒までの
0.25秒間隔の数値(半角)

fadein (255 , 0 , 0 , 1);

R値(赤)

0~255(半角)

G値(緑)

0~255(半角)

B値(青)

0~255(半角)

フェードアウト(フルカラーLED)

fadeout(数値,数値,数値,数値);

関数名 (R値 , G値 , B値 , 時間);

関数名 (半角)

フルカラーLED
を点灯させる命令

時間

0~31.75秒までの
0.25秒間隔の数値(半角)

fadeout (255 , 0 , 0 , 1);

R値(赤)

0~255(半角)

G値(緑)

0~255(半角)

B値(青)

0~255(半角)

点灯(連続)(フルカラーLED)

fullledconti(色);

関数名 (半角)

フルカラーLED消灯

7色 (半角)

red green
blue yellow
White purple
lightblue

fullledconti(色);

消灯(フルカラーLED)

fullledoff();

関数名 (半角) フルカラーLED消灯

fullledoff ();

No.3

年

組

名前

音入力があるまで停止

soundwait();

関数名 (半角) 音入力あるまで停止

soundwait ();

SWが押されるまで停止

swwait();

関数名 (半角) SWが押されるまで停止

swwait ();

信号入力があるまで停止

inputwait();

関数名 (半角) 信号入力があるまで停止

inputwait ();

アラーム時刻になるまで停止

alermwait();

関数名 (半角) アラーム時刻になるまで停止

alermwait ();

暗くなるまで停止

lightvaluwait(1~100);

関数名 (半角)
明るくなるまで停止明るさ (半角)
0~100までの数値

lightvaluwait (数値);

暗くなるまで停止

darkvaluwait(1~100);

関数名 (半角)
暗くなるまで停止明るさ (半角)
0~100までの数値

darkvaluwait (数値);

指定時刻になるまで停止

timewait(0~23 : 0~59);

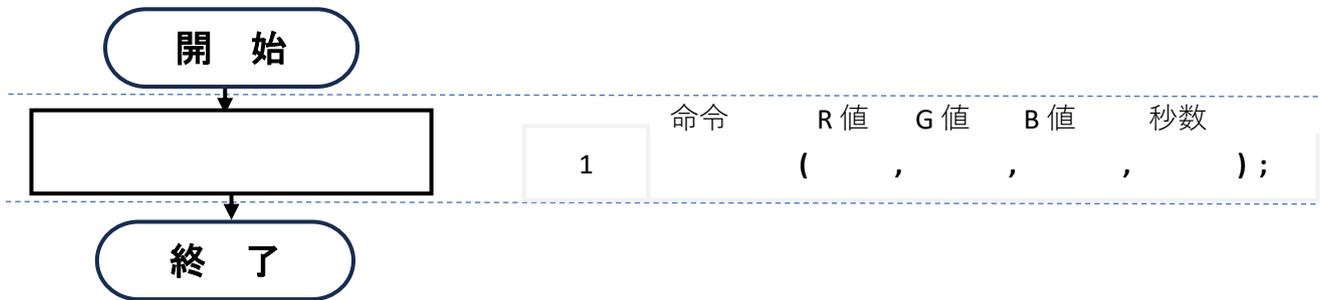
関数名 (半角)
指定時刻になるまで停止時 (半角)
0~23までの数値分 (半角)
0~59まで

timewait(数値:数値);

順次処理のプログラム

フローチャートを用いて、文字でのプログラムを行います。 命令を横にそろえて

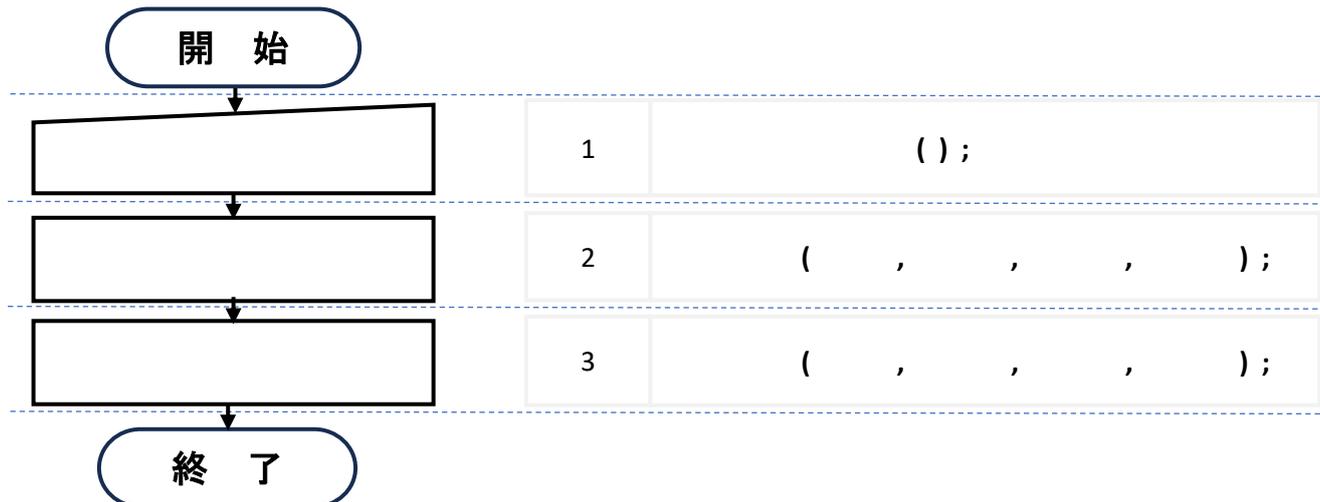
課題 4-1 赤を1秒ずつ点灯させてみよう。メニューの"転送(F6)"で、作成したプログラムを本体に転送します。 "実行(F7)"を押すと、転送したプログラムが動き出します。



課題 4-2 赤、緑、青を1秒ずつ点灯させてみよう。メニューの"転送(F6)"で、作成したプログラムを本体に転送します。 "実行(F7)"を押すと、転送したプログラムが動き出します。



課題 4-3 信号待ち命令を使ってみよう。音入力あるまで待ち命令の後に好きな色を2色点灯させてみよう。それぞれの信号待ち命令の動作が理解できたら、チェックをしましょう。



点灯 (バックライト)

`backlighttime (数値);`

関数名 (明るさの数値 , 時間);

関数名(半角)

バックライトを点灯させる命令

関数名(半角)

0~31.75秒までの
0.25秒間隔の数値

`backlighttime (1);`

点灯(連続) (バックライト)

`backlightconti ();`

関数名 (半角) 信号入力があるまで停止

`backlightconti ();`

消灯 (バックライト)

`backlightoff ();`

関数名 (半角) アラーム時刻になるまで停止

`backlightoff ();`

反 復 命 令

フローチャート の場合

繰り返し命令の指定した回数分だけ

「繰り返し (回数)」と
「繰り返し終了」で囲まれた命令を繰り返します。

開始

繰り返し 5 回

命令

命令

繰り返し終了

終了

文字プログラム の場合

繰り返し命令の指定した回数分だけ

「loopstart (回数)」と
「loopend ()」で囲まれた命令を繰り返します。

`loopstart (5);`

命 令

命 令

`loopend ();`

繰り返しの関数は「loopstart」
「loopend」の二つでワンセット

`loopstart (半角);`

この2つで対になります

`loopend ();`

`loopstart (回数);`

命令

ループで挟んだ
命令を”回数”分
実行します

`loopend ();`

順次処理のプログラム

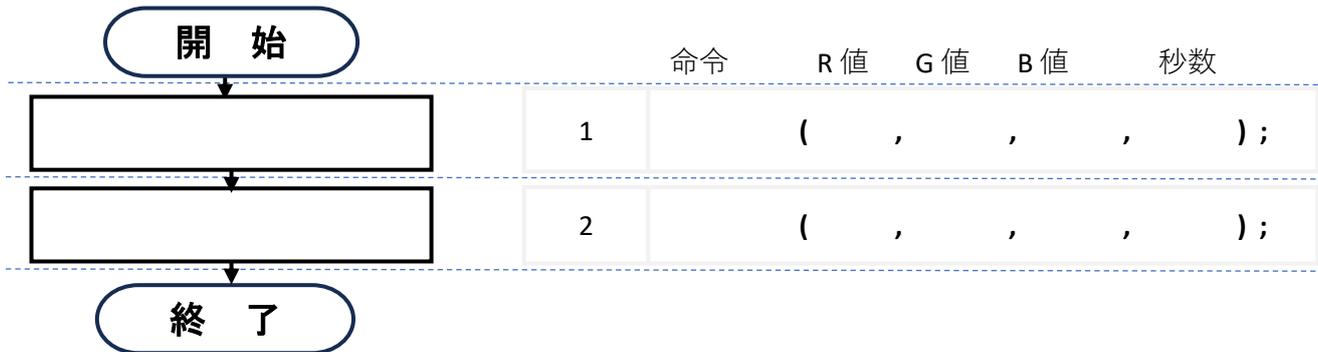
プログラムには「順次処理」、「反復処理」、「分岐処理」があることを以前に学びました。

ここからは、これらの処理方法を実際のプログラムで試してみましょう。

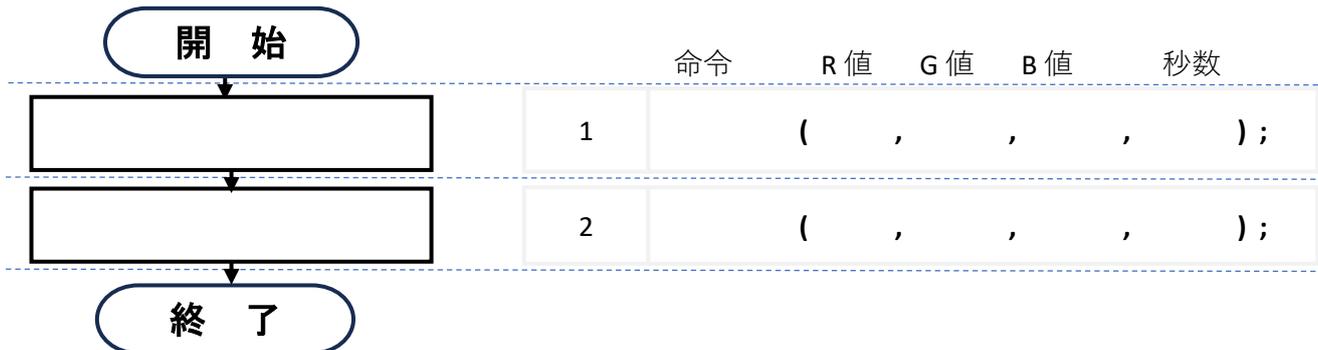
最終的には、信号機のプログラムを作ることを目指します。課題は簡単なものから少しずつ難しくなっていきます。これらの課題を通して、信号機の動作やプログラムを考えてみましょう。最初に、「順次処理」のプログラムを作ってみましょう。



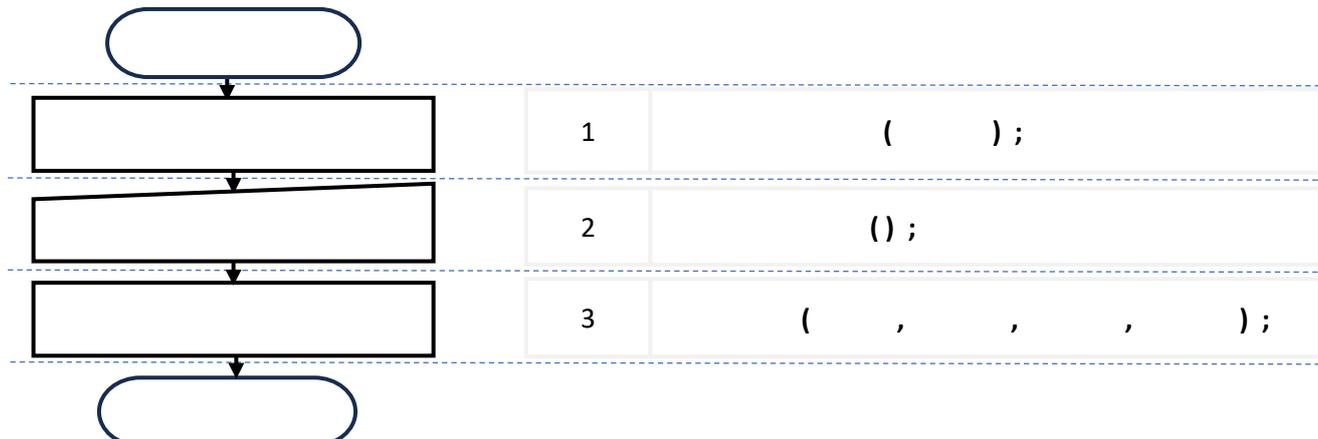
課題 6-1 歩行者用信号機の点灯を想像してください。赤が15秒点灯青が10秒点灯するプログラムを作ってみよう。



課題 6-2 課題6-1で作ったプログラムを元に、道路の幅が倍になった場合を考えてみましょう。(道路の幅が倍なので、歩行者が渡る時間も倍になります。)



課題 6-3 押しボタン式信号機を考えてみましょう。普段は赤色が点灯していて歩行者が渡りたい時に、ボタンを押すと青に変わります。この動作をプログラムしてみよう。(ヒントは、「LED連続命令」と「SWが押されるまで停止」命令を使うことです。青の点灯時間は、10秒とします。)



反復処理のプログラム

次に、反復処理のプログラムを作ってみましょう。

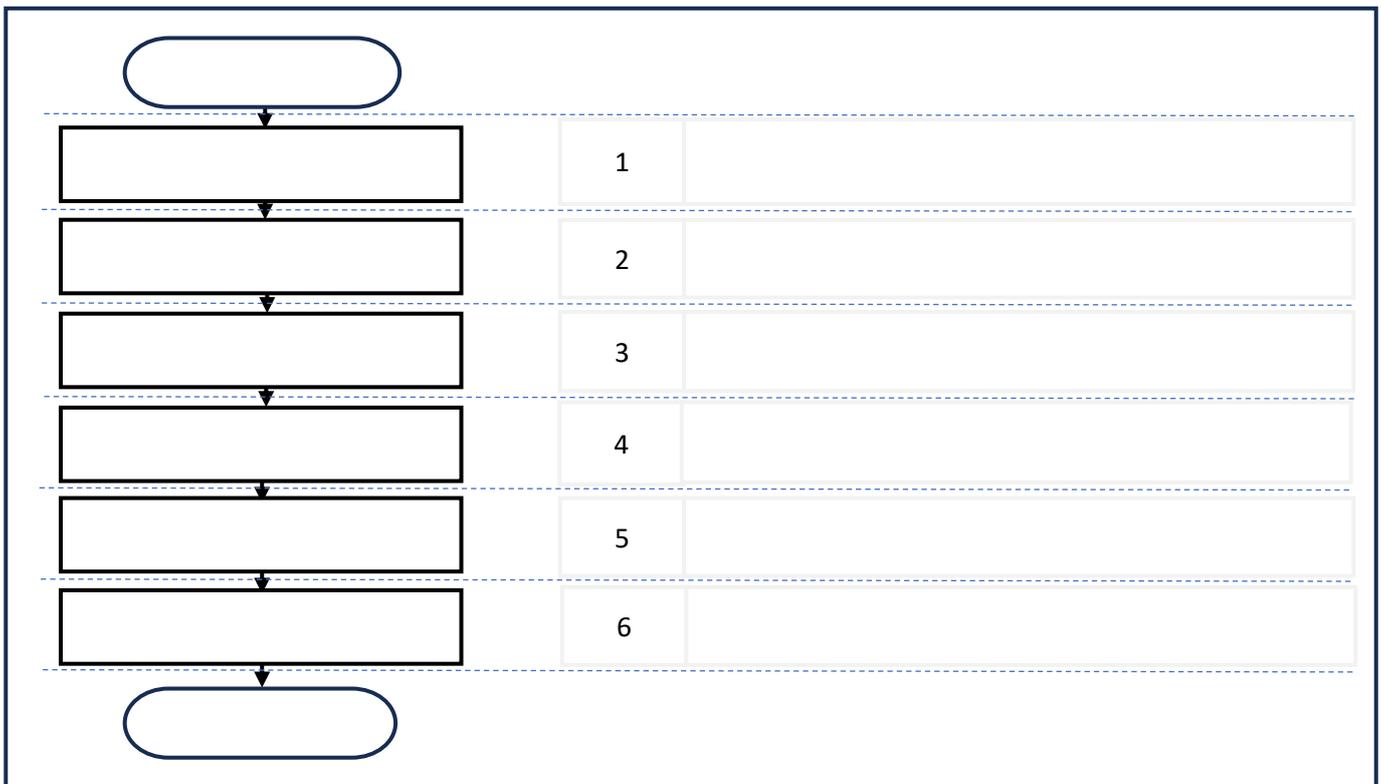
歩行者用信号機ではどのようなところで反復処理が使われているでしょうか？



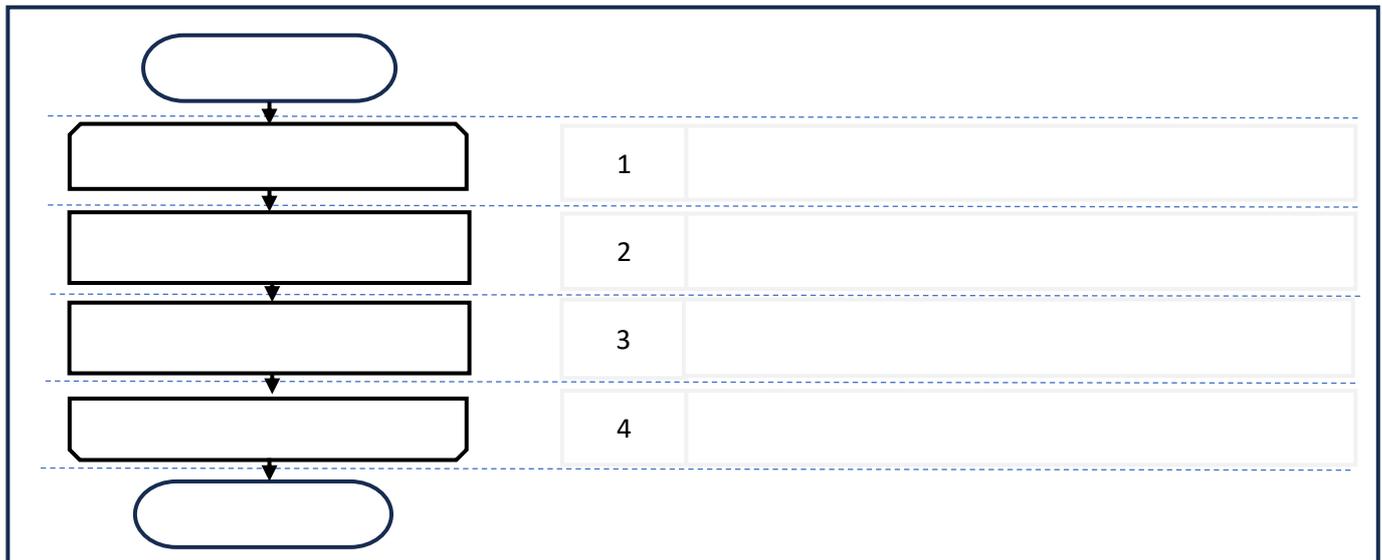
※繰り返し命令の中に繰り返し命令を入れることはできません。(二重ループはできません。)

課題 7-1 歩行者用信号機が青から赤に変わる前に、青は点滅します。

消灯0.5秒、青点灯0.5秒、消灯0.5秒、青点灯0.5秒、消灯0.5秒、青点灯0.5秒、の順で点滅するプログラムを作ってみよう。ただし、繰り返し命令は使わないこと。



課題 7-2 課題7-1を繰り返し命令で使って作ってみよう。

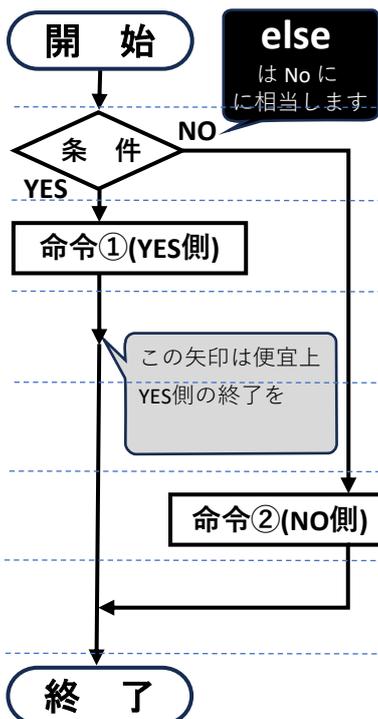


フローチャートの場合

設定した条件で動作する命令はYES側とNO側に左右へ分岐します。

このとき、どちらか一方しか動作しません。

YES側の命令の終わりに矢印がありますが、解り易くするためです。本来の表記にはありません。



文字プログラムの場合

フローチャートと異なりタテにプログラムが並びます。

「if (条件)」の命令 (YES側)、その後「else」の命令 (NO側) で表します。

YESであれば、YES側のみ動作し、NO側の命令は動作しません。また、NOであれば、NO側の命令だけ動作し、YES側の

```

if (条件) {
    命令① (YES側)
}
else {
    命令② (NO側)
}
    
```

条件の関数は

「 if (条件) { 命令 } 」 YES側
 「 else { 命令 } 」 NO側



もし～ならば
 YESなら
 YES側のみが動作
 (NO側は動作しない)

でなければ
 NOなら
 NO側のみが動作
 (YES側は動作しない)

UDで使用できる条件の種類と解説

sw = on

スイッチが押されているかどうか

押されていれば"YES"
 押されていなければ"NO"

sw = off

スイッチが押されているかどうか

押されていなければ"YES"
 押されていれば"NO"

c ds > 【数値】

【数値】よりも明るいかどうか

(【数値】：0～100までの数値 (半角))
 明るければ"YES"、暗ければ"NO"

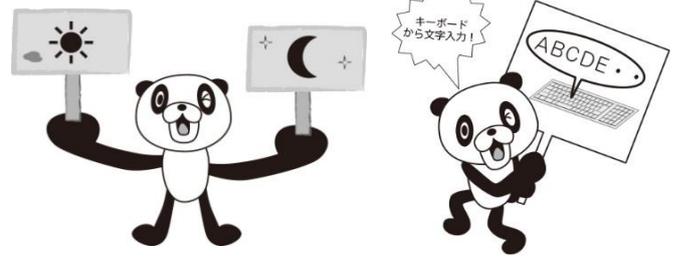
c ds < 【数値】

【数値】よりも暗いかどうか

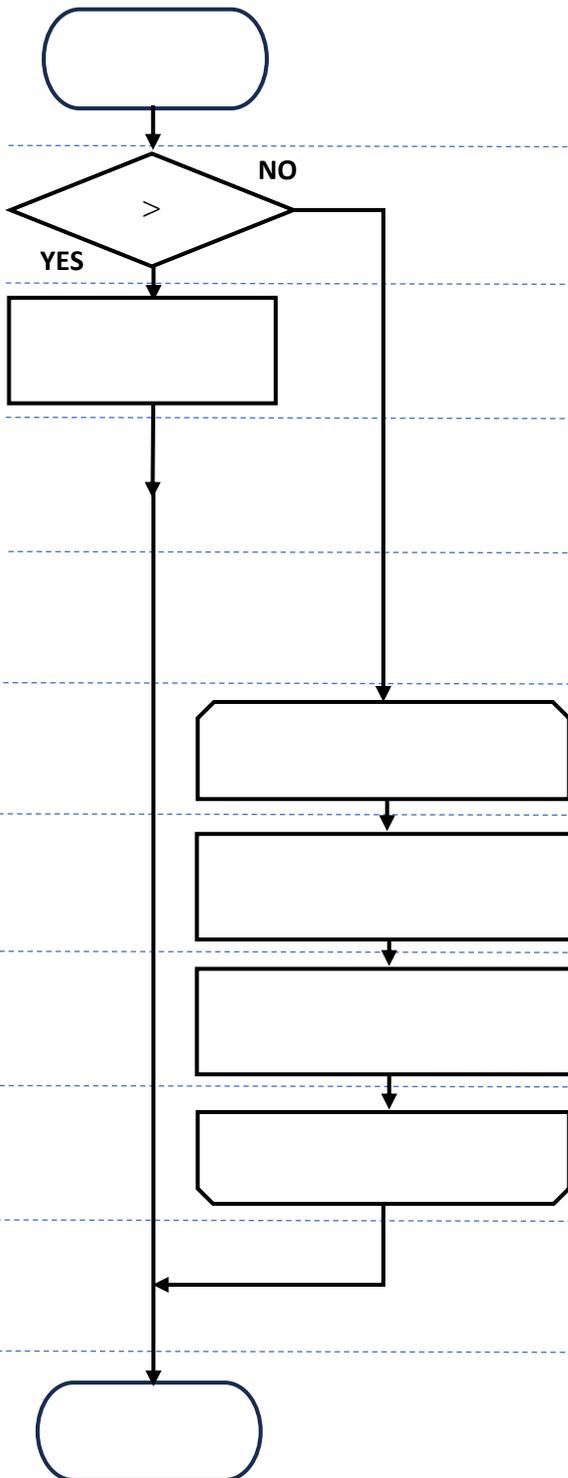
(【数値】：0～100までの数値 (半角))
 暗ければ"YES"、明るければ"NO"

分岐処理のプログラム

次に、分岐処理のプログラムを作ってみましょう。
 信号機では、夜になると点滅するものがあります。
 夜と昼をどのようにして見分けているのでしょうか？

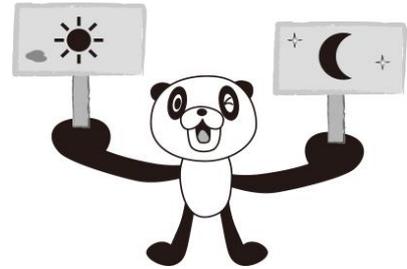


課題 10 明るい時は、赤が15秒点灯、暗い時は、赤が5回点滅するプログラムを作ってみましょう。
 (明るさの値や点滅時間は自由に設定してください。)



1	(>) {
2	(, , ,);
3	}
4	{
5	();
6	(, , ,);
7	(, , ,);
8	();
9	}

分岐処理のプログラム



次に、分岐処理のプログラムを作ってみましょう。

信号機では夜になると点滅するものがあります。

夜と昼をどのようにして見分けているのでしょうか？

課題 11 明るい時は、赤が15秒点灯、青が10秒点灯する。 暗い時は、赤が10回点滅するプログラムを作ってみましょう。(明るさの値は自由に設定してください。)

開始	
YES	NO
[]	1 () {
[]	2 (, , ,);
[]	3 (, , ,);
[]	4 }
[]	5 {
[]	6 ();
[]	7 (, , ,);
[]	8 (, , ,);
[]	9 ();
[]	10 }
終了	

実用的なプログラム

ここまで、順次、反復、分岐処理を使い、歩行者用信号機のプログラムを作ってきましたが、これらは仮想の信号機でした。ここからは、歩行者用だけではなく、車用の信号機も含めた、いろいろな種類の信号機の動作を調べてみましょう。

課題12-1 信号機について調べてみよう。信号機の特長やオーロラクロックでプログラムする時に必要になりそうな機能を書き出してみよう。

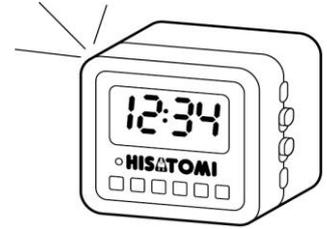
信号機	信号機
特徴	
機能	

```

1      (      ) {
2      (      ,      ,      ,      );
3      (      ,      ,      ,      );
4      (      ,      ,      ,      );
5      }
6      {
7      (      );
8      (      ,      ,      ,      );
9      (      ,      ,      ,      );
10     (      );
11     }
    
```

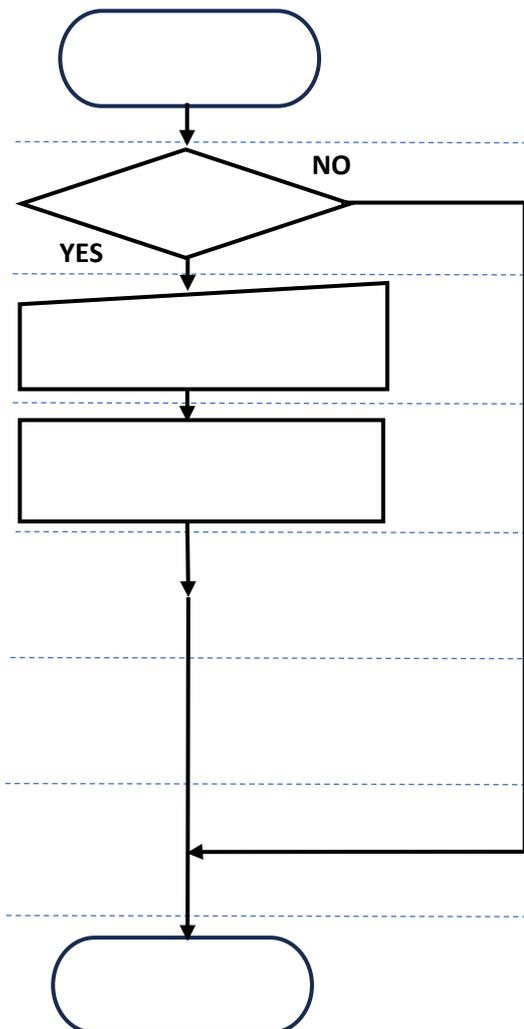
実用的なプログラム

ここまで、順次、反復、分岐処理を使って、プログラムを作ってきました。
ここからは、あると便利で役に立つプログラムを考えて作ってみましょう。



課題 13-1 例えば、暗い時に時間が分かり、明るい時は何もしない時計のプログラムを考えてみましょう。(夜中、部屋で目が覚めた時に時間を確認する時を想定しています。)

プログラムの名前	プログラムの工夫した点
使用するセンサ・アクチュエータ それらの説明	



課題13-2 自分で考えたオリジナルの動作をプログラムしてみよう。

1	()	{
2		();	
3		(,);
4		}		
5		{		
6		}		

実用的なプログラム

これまで、オーロラロックを使って信号機を作ってきました
ここからは使う人の事も考えたプログラムを作ってみましょう。

課題14-1 家族にオーロラロックの機能を説明し、みんなで使えるプログラムを考えてみよう。
使う人の立場に立って考え、今まで習った知識を活かして工夫してみましょう。

プログラムの名前

プログラムの工夫した点

プログラムの目的

課題14-2 上記で考えた動作をプログラムしてみよう。

開始

1

終了