

情報・制御教材

型番

オーロラクロック 2N

UC-7
UC-8

 Webアプリ版

文字入力プログラムワーク集


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】

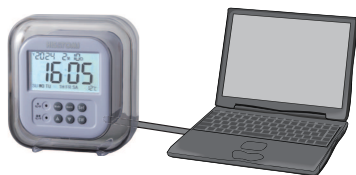
年 組 番

氏名

ワーク No.1

「文字プログラム」を使ってみよう

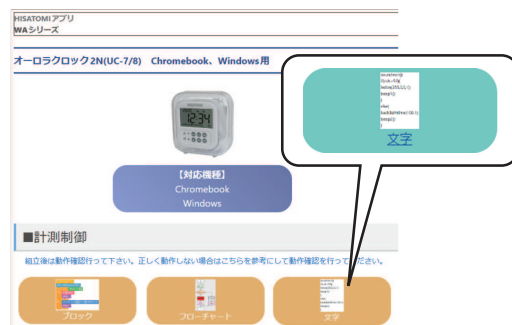
①パソコンと接続



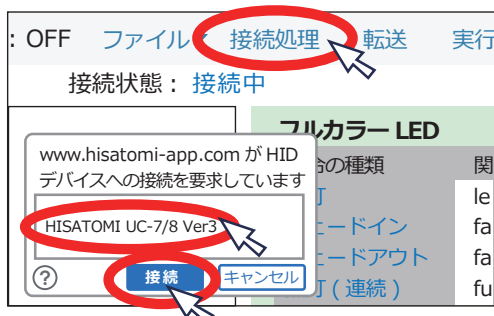
②QRコードでwebアプリを開き機種を選択



③計測・制御の「文字」を選択



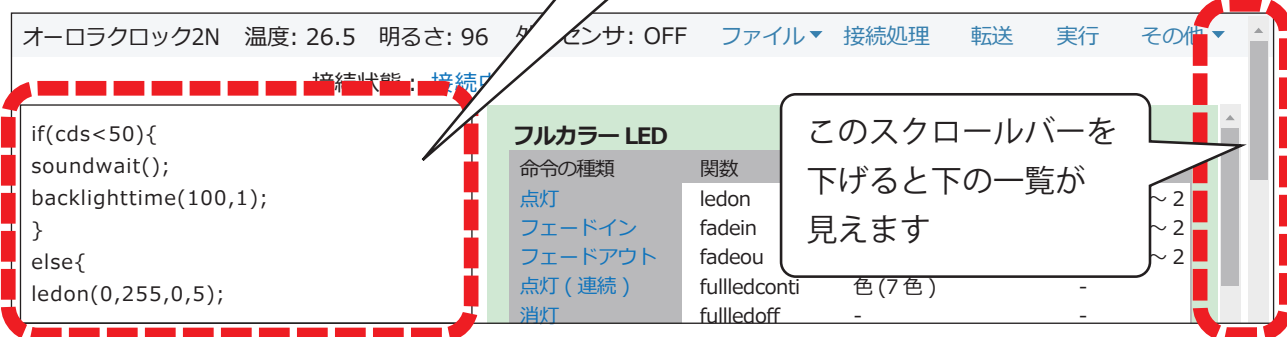
④「接続処理」、「UC-7/8」 「接続」をクリック



⑤左の入力画面でキーボードからプログラムを入力

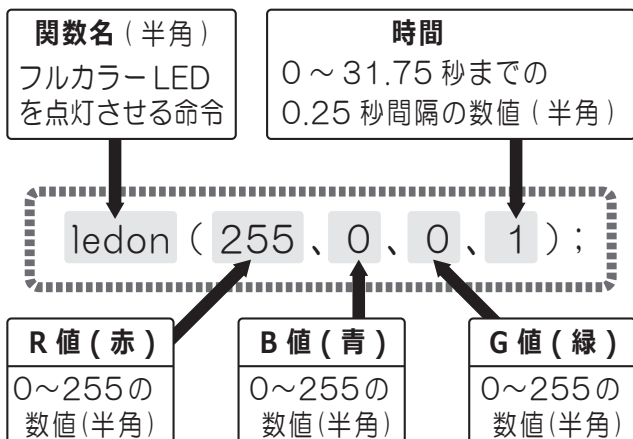
点線で囲まれたプログラム画面で、命令の文字入力をキーボードから行います。

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。



このスクロールバーを下げると下の一覧が見えます

関数名 (R 値、 G 値、 B 値、 時間);



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。



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 (半角)

0 ~ 31.75 秒までの
0.25 秒間隔の数値

`ledon ( 255、0、0、1 );`

R 値 (赤)

0~255(半角)

B 値 (青)

0~255(半角)

G 値 (緑)

0~255(半角)

フェードイン(フルカラーLED)

`fadein(数値,数値,数値);`

関数名 (R 値、G 値、B 値);

関数名 (半角)

フルカラーLEDのフェードイン命令

`fadein ( 255、0、0 );`

R 値 (赤)

0~255(半角)

B 値 (青)

0~255(半角)

G 値 (緑)

0~255(半角)

フェードアウト(フルカラーLED)

`fadeout(数値,数値,数値);`

関数名 (R 値、G 値、B 値);

関数名 (半角)

フルカラーLEDのフェードアウト命令

`fadeout ( 255、0、0 );`

R 値 (赤)

0~255(半角)

B 値 (青)

0~255(半角)

G 値 (緑)

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点灯(連続)

`fullledconti ( 色 );`

7 色 (半角)

red green
blue yellow
White purple
lightblue

消灯(フルカラーLED)

`fullledoff();`

関数名 (半角)

フルカラーLED消灯

`fullledoff();`

ワーク No.3

信号待ち命令について

音入力があるまで停止

soundwait();

関数名 (半角) 音入力があるまで停止

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

alarmwait();

関数名 (半角) アラーム時刻になるまで停止

alarmwait ();

明るくなるまで停止

lightvaluawait(1~100);

関数名 (半角)

明るくなるまで待つ

明るさ (半角)

0 ~ 100 までの数値

lightvaluawait (数値);

暗くなるまで停止

darkvaluawait(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

darkvaluawait (数値);

指定温度になるまで停止

tempwait(1~100);

関数名 (半角)

指定温度になるまで停止

明るさ (半角)

0 ~ 100 までの数値

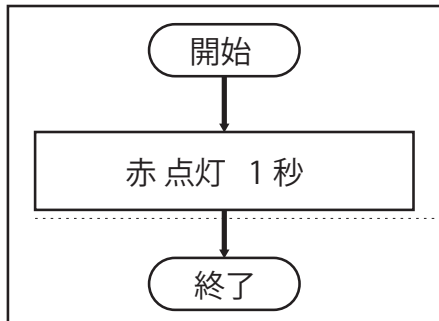
tempwait (数値);

ワーク No.4

フルカラーLED命令

課題 4-1: 赤を 1 秒ずつ点灯させてみよう。

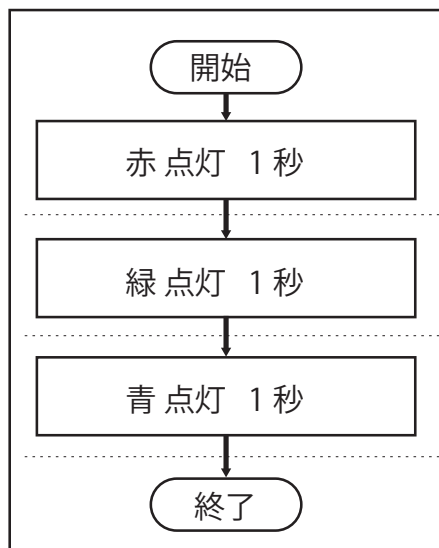
メニューの“**転送 (F6)**” で、作成したプログラムを本体に転送します。“**実行 (F7)**” を押すと、転送したプログラムが動き出します。



命令	R 値	G 値	B 値	時間
1	(	,	,	) ;

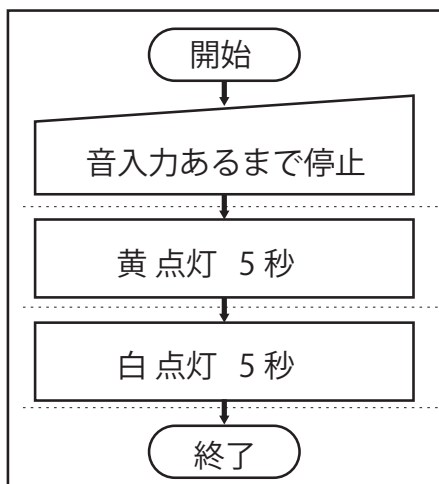
課題 4-2: 赤、緑、青を 1 秒ずつ点灯させてみよう。

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。



命令	R 値	G 値	B 値	時間
1	(	,	,	) ;
2	(	,	,	) ;
3	(	,	,	) ;

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。



1	() ;
2	(, , ,) ;
3	(, , ,) ;

ワークNo.5-1 バックライト命令について

点灯 (バックライト)

backlighttime

点灯(連続) (バックライト)

backlightconti

消灯 (バックライト)

backlightoff

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

明るさ (半角)

0 ~ 100
までの数値

時間 (半角)

0 ~ 31.75秒までの
0.25秒間隔の数値

backlighttime (100 , 0);

関数名 (半角)

バックライト点灯 (連続)

backlightconti ();

関数名 (半角)

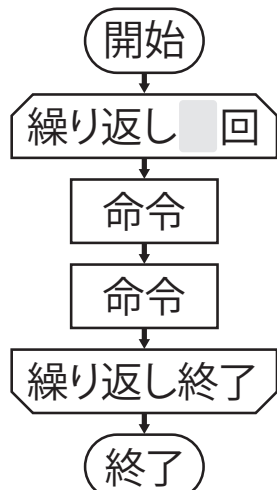
バックライト消灯

backlightoff ();

ワークNo.5-2 STEP3(文字)反復命令について

反復命令

フローチャート
の場合



文字プログラム
の場合

loopstart (回数);

命令

⋮

loopend ();

loopstart (回数); (半角)

命令

loopend (); (半角)

loopstart (回数);

命令

⋮

loopend ();

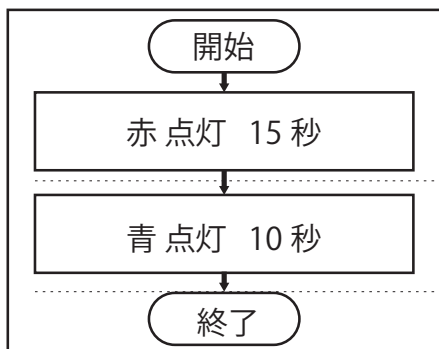
命令を
"回数"
分実行

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。

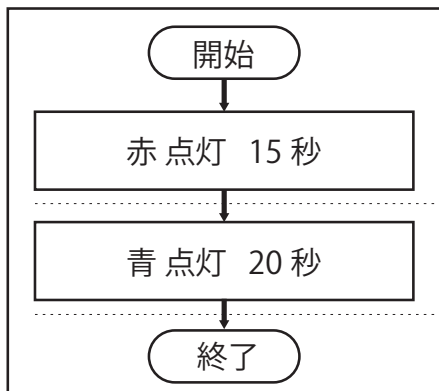


赤が 15 秒点灯青が 10 秒点灯するプログラムを作ってみよう。

命令	R 値	G 値	B 値	時間
1	(	,	,	,) ;
2	(	,	,	,) ;

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。

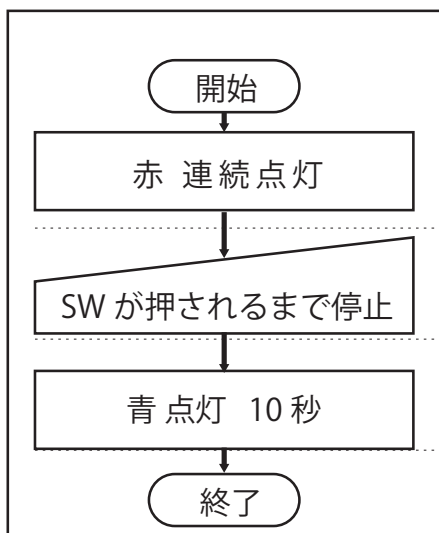
(道路の幅が倍なので、歩行者が渡る時間も倍になります。)



命令	R 値	G 値	B 値	時間
1	(	,	,	,) ;
2	(	,	,	,) ;

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。)



1	(	) ;
2	() ;	
3	(, , ,) ;	

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。

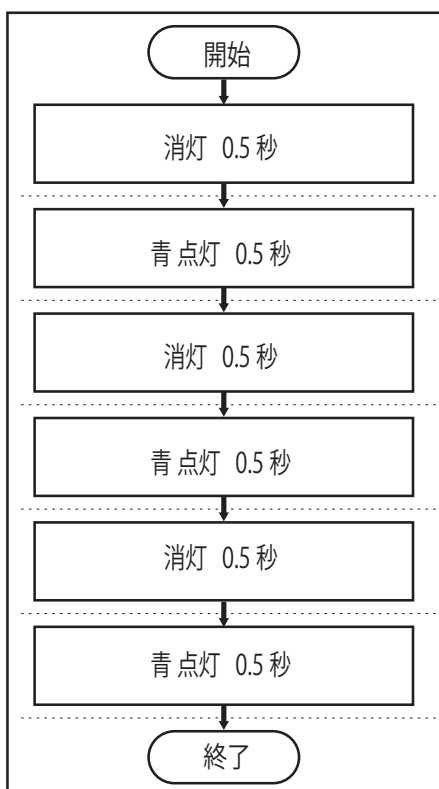
歩行者用信号機ではどのようなところで反復処理が使われているでしょうか？



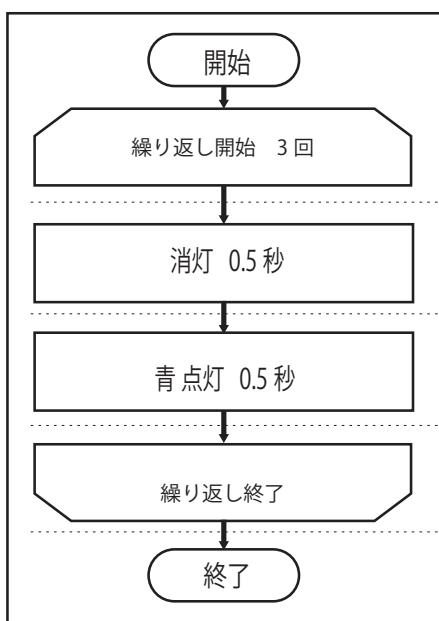
※繰り返し命令の中に繰り返し命令を入れることはできません。(二重ループはできません。)

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。



命令	R 値	G 値	B 値	時間
(, , ,) ;				
(, , ,) ;				
(, , ,) ;				
(, , ,) ;				
(, , ,) ;				
(, , ,) ;				



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。

() ;
(, , ,) ;
(, , ,) ;
() ;

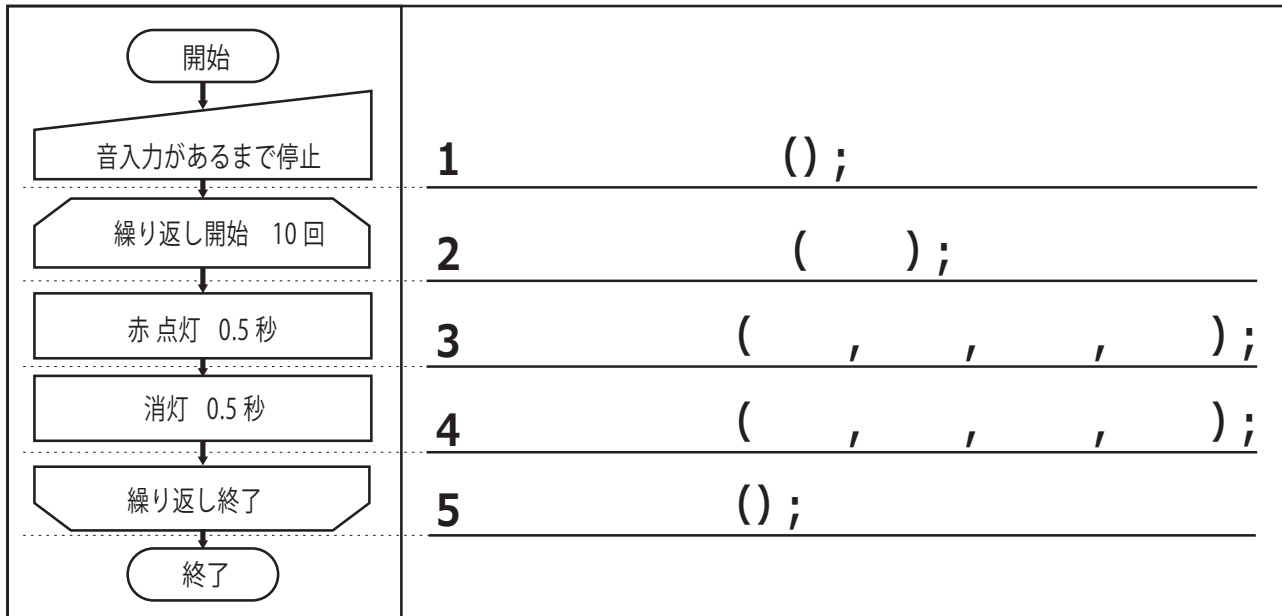
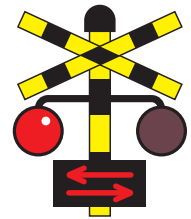
ワーク No.8

計測と制御の学習 グループ学習

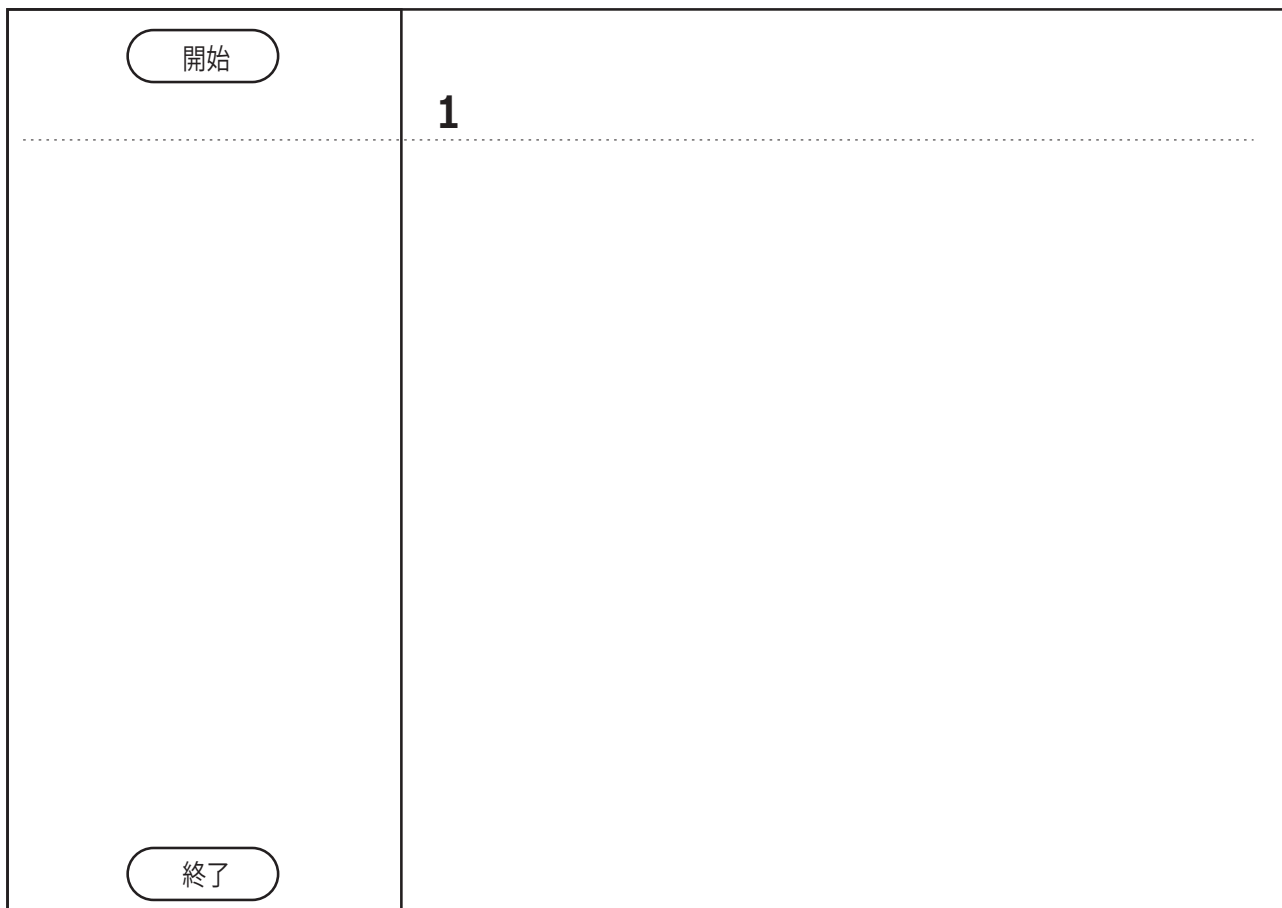
2人で踏切プログラムを作ってみよう。

課題 8-1：二人一組でプログラムを作ります。 0.5 秒毎に赤 LED が交互に 10 回点滅するプログラムを作ってみよう。

(プログラムを同時に開始するには、どうすれば良いか考えよう)



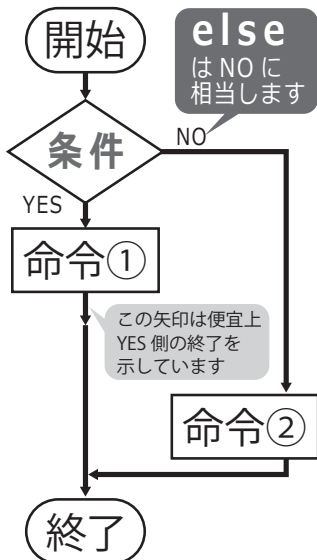
課題 8-2：二人以上で 10 秒以内のイルミネーションプログラムを作ってみよう。出来上がったならグループで発表し、さらに良くなるようにグループで考え工夫しよう。



ワーク No.9

条件分岐命令について

フローチャートの場合



文字プログラムの場合

if(条件) {

命令①

YES 側の命令を実行

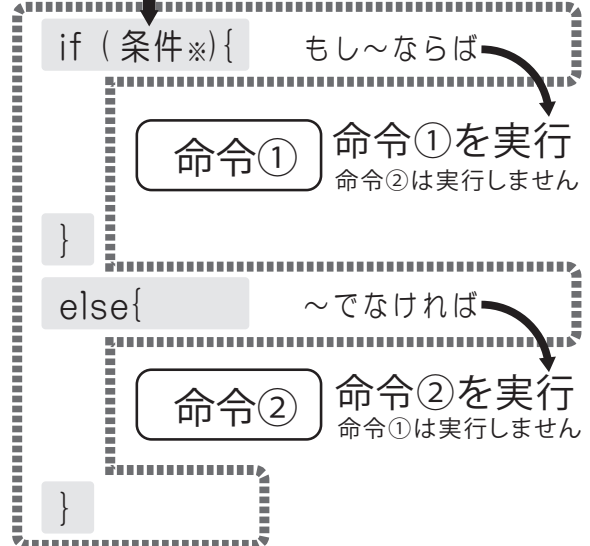
} else {

命令②

NO 側の命令を実行

}

if(条件){ 命令 } (半角) ある条件で
else { 命令 } (半角) プログラムが分岐



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"

cds>【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"

cds<【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))
暗ければ"YES"、明るければ"NO"

temp>【数値】

温度が【数値】度よりも高いかどうか
(【数値】:1~50度までの数値(半角))
高ければ"YES"、低ければ"NO"

temp<【数値】

温度が【数値】度よりも低いかどうか
(【数値】:1~50度までの数値(半角))
低ければ"YES"、高ければ"NO"

temp=【数値】

温度が【数値】度かどうか
(【数値】:1~50度までの数値(半角))
【数値】度なら"YES"、そうでなければ"NO"

変数xに温度を代入 gettemp();

「変数xに温度を代入」とは、プログラムを実行したとき
その時の温度を取得して記憶し、変数「x」となります。
その変数「x」を元に以下に続く条件命令を利用できます。

temp>x

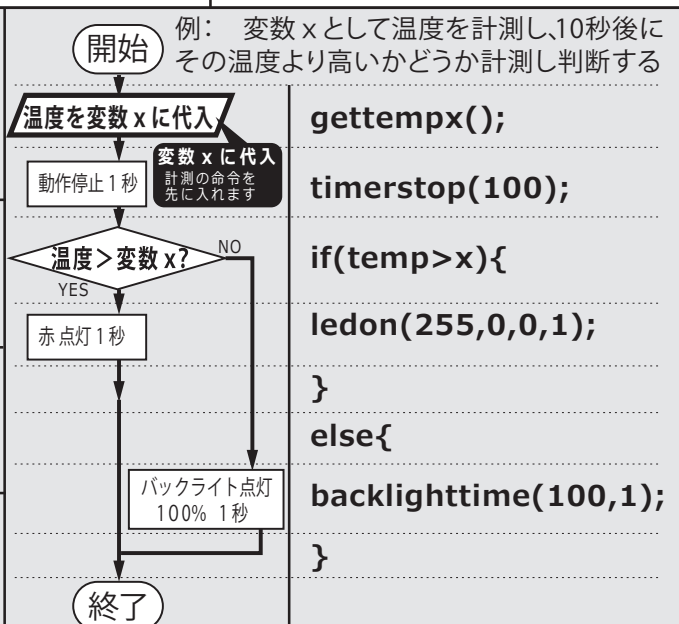
温度が変数 x よりも高いかどうか
変数 x より高ければ"YES"、低ければ"NO"
(分岐より前に「変数xに温度を代入」の命令が必要)

temp<x

温度が変数 x よりも低いかどうか
変数 x より低ければ"YES"、高ければ"NO"
(分岐より前に「変数xに温度を代入」の命令が必要)

temp=x

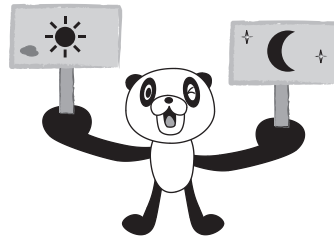
温度が変数 x かどうか
変数 x と同じなら"YES"、そうでなければ"NO"
(分岐より前に「変数xに温度を代入」の命令が必要)



ワーク No.10 計測と制御の学習 分岐処理

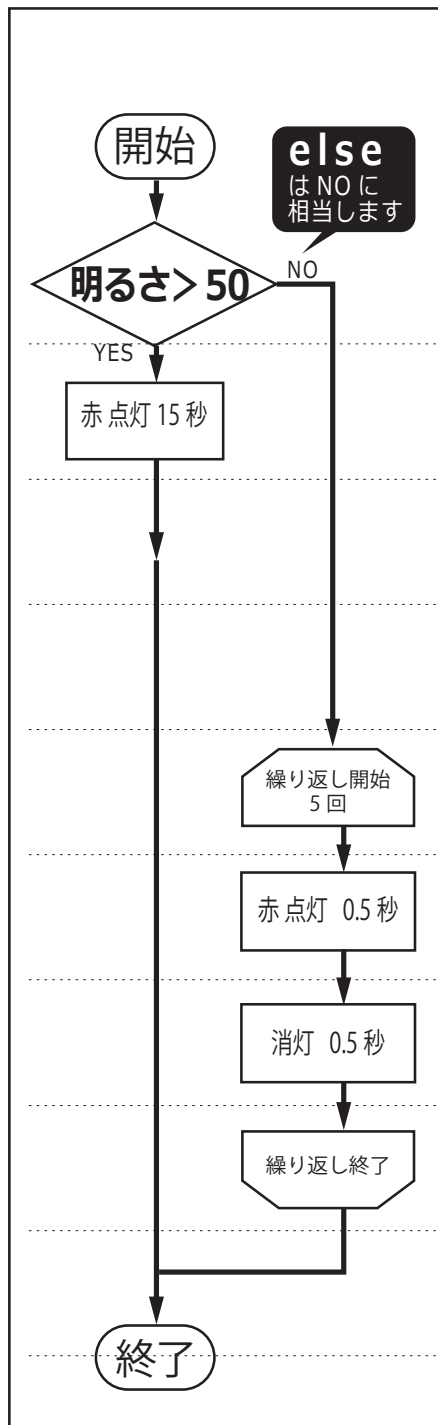
分岐処理のプログラム

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。)



```

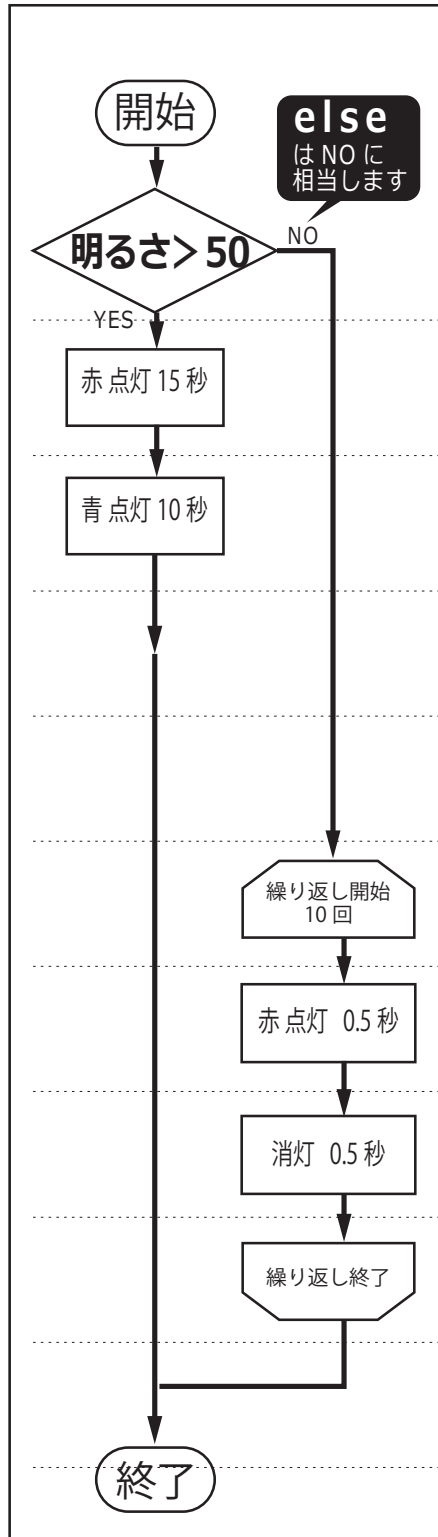
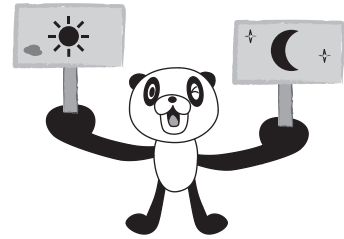
1  (      ) {
2      (      ,      ,      );
3  }
4  else{
5      (      );
6      (      ,      ,      );
7      (      ,      ,      );
8      (      );
9  }
  
```

ワーク No.11 計測と制御の学習 分岐処理

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。)

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

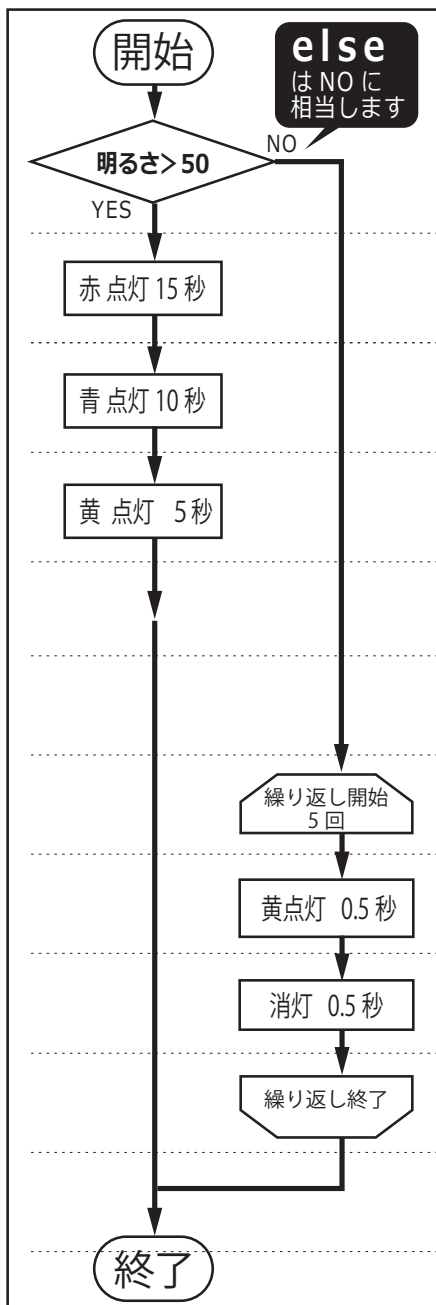
ワーク No.12 計測と制御の学習 分岐処理

信号機プログラムのまとめ

ここまで、順次、反復、分岐処理を使って、歩行者用信号機のプログラムを作ってきました。しかし、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を調べてみましょう。そして、その信号機の動作をオーロラクロックでプログラムしてみましょう。

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ましょう。

信号機	自動車用信号機
特徴	
機能	



課題 12-2：ここまで調べた信号機の動作をプログラムしてみよう。

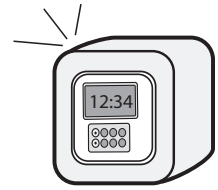
```

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

ワーク No.13 計測と制御の学習 実用的なプログラム

実用的なプログラム

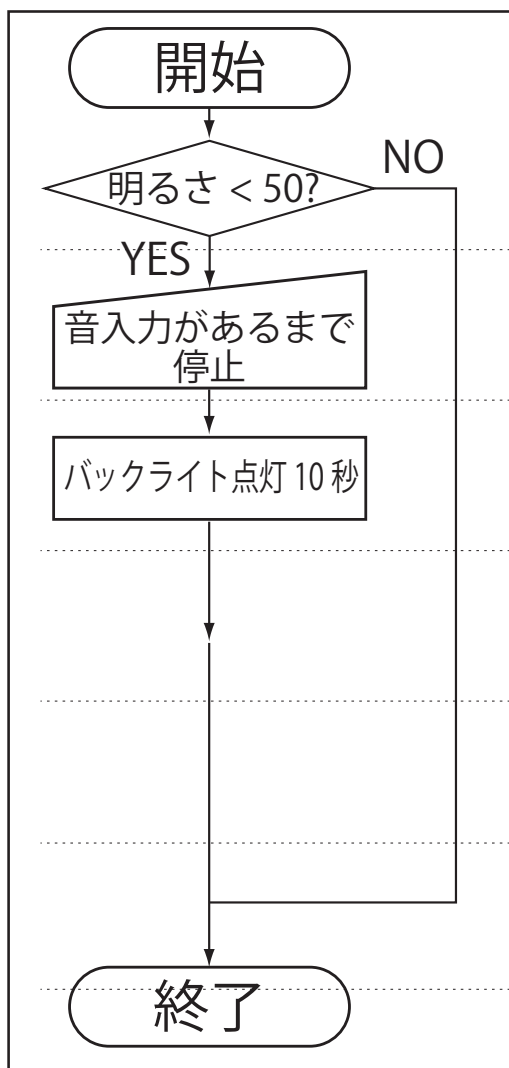
これまで、色々な処理を使って、プログラムを作ってきました。
ここからは、あると便利で役に立つプログラムを作ってみましょう。



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
使用するセンサとアクチュエータその機能

プログラムの工夫した点



課題 13-2: 課題 13-1: で考えた仕様でプログラムを作成しましょう。

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

ワーク No.14 計測と制御の学習 実用的なプログラム

実用的なプログラム 持って帰って使えるプログラムを、いままで学習した内容を生かして、使う人のことも考えたプログラムを作ってみましょう。

課題 14: 家族にオーロラク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たプログラムを工夫してみよう。

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	

開始 ↓	フローチャート
	1
	2
	3
	4
	5
	6
	7
	8
	9
	10
	11
	12
	13
	14
	15
	16
	17
	18
	19
	20
	21
終了	

久富電機産業株式会社