

高校生ものづくりコンテスト  
電子回路組立部門  
指 導 書



ZENJYOUKEN

平成22年1月

全国情報技術教育研究会

## ご 挨拶

埼玉県立久喜工業高等学校長 新井 誠  
(前全国情報技術教育研究会長)

全国情報技術教育研究会では、(社)全国工業高等学校長協会主催の高校生ものづくりコンテスト「電子回路組立」部門を、第5回東京大会から支援させて頂いています。これからのものづくりは「組込み技術」であり、ソフトウェア技術とハードウェア技術それに両者を結ぶインターフェース技術が情報技術教育には必須であると考えたからです。予選会である各地区大会の運営には、会員の皆様のご協力で毎年活発になっていますのは大変喜ばしいことです。

第9回神奈川大会が、私個人にとってお手伝いできる最後の大会となってしまいました。皆様のご支援に深く感謝申し上げます。今後の「組込み技術」教育の一助になればと思い指導資料をまとめてみました。先生方の研修に授業の指導資料に活用いただけると幸いです。

ご協力いただきました皆様をご紹介しますと共に深く感謝申し上げます。

東海大学専門職大学院組込み技術研究科 研究科長 教授 大原 茂之 様  
(株)ルネサスソリューションズ 第二応用技術本部 主管技師 磯 佳実 様  
日産ディーゼル工業株式会社 技能五輪指導員 田中 耕介 様  
全国情報技術教育研究会「電子回路組立」部門課題作成委員会  
第9回高校生ものづくりコンテスト「電子回路組立」部門実行委員会

平成22年1月吉日

## 第9回高校生ものづくりコンテスト全国大会 電子回路組立部門 課題

### 1. 課題

競技時間中に製作する『設計製作回路①』と大会当日に配布する『制御対象回路③』を、事前に製作したケーブルにより『持参コンピュータ②』と接続し、競技時間内に『制御プログラム④』を作成し、目的の動作を行うシステムを完成させる。

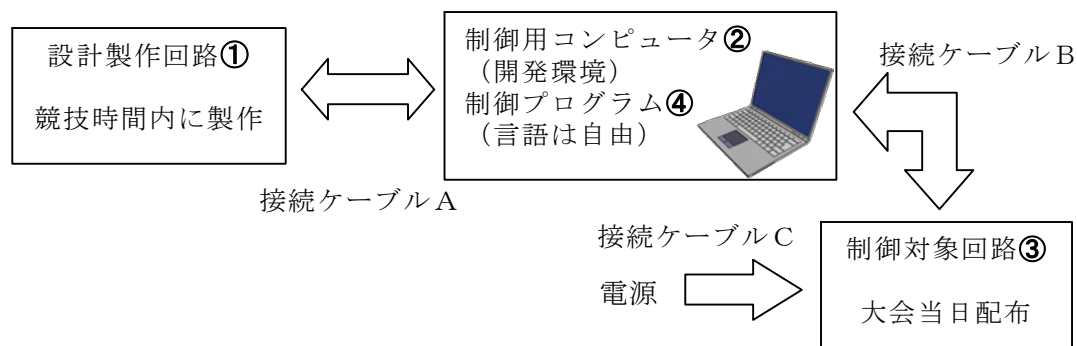


図1 課題概略図

#### (1) 設計製作回路①

大会当日に示す設計仕様に基づく電子回路を設計し、ユニバーサル基板を用いて電子回路基板を製作する。配線はスズメッキ線を使用し、設計製作回路は以下の部品を使用する。

- ①ユニバーサル基板（サンハヤト ICB293 相当品）
- ②フォトインタラプタ、スイッチ、コネクタ、0.5φスズメッキ線等

#### (2) 制御用コンピュータ②

開発環境を含め全て持参する。コンピュータの性能・形状等に制限はない。

#### (3) 制御対象回路③

大会当日に競技委員から配布する。制御対象回路には制御対象駆動回路と制御対象が含まれている。なお、制御対象としては、次のようなものが考えられる。

- ①LED
- ②7セグメントLED
- ③DCモータ
- ④ステッピングモータ等

#### (4) 制御プログラム④

大会当日に提示する仕様に基づいたプログラムを作成する。使用する言語は自由である。なお、プログラムの仕様例として、次のようなものがある。

- ①ストップウォッチのプログラム
- ②回転制御のプログラム

(5) 接続ケーブル

①接続ケーブルA（設計製作回路①用）

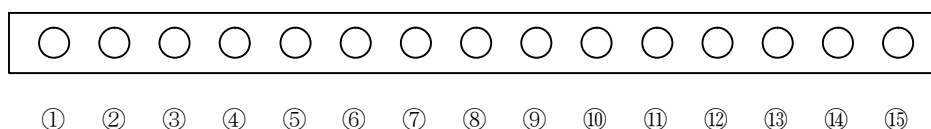
I C ピッチ 1 列 5 ピンコネクタ・メス（ストレートピンヘッダハウジング）



図 2 接続ケーブルA用コネクタのピン配置

②接続ケーブルB（制御対象回路③用）

I C ピッチ 1 列 1 5 ピンコネクタ・メス（ストレートピンヘッダハウジング）



①	GND	②	出力 0	③	出力 1	④	出力 2	⑤	出力 3
⑥	出力 4	⑦	出力 5	⑧	出力 6	⑨	出力 7	⑩	出力 8
⑪	出力 9	⑫	N C	⑬	N C	⑭	N C	⑮	N C

図 3 接続ケーブルB用コネクタのピン配置

③接続ケーブルC（電源供給用）

I C ピッチ 1 列 3 ピンコネクタ・メス（ストレートピンヘッダハウジング）

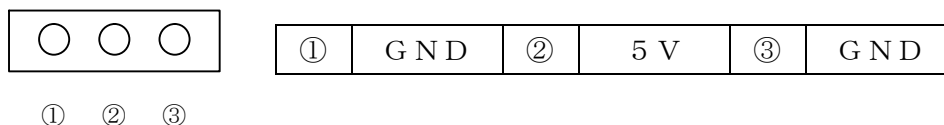


図 4 接続ケーブルC用コネクタのピン配置

注) コネクタ・メスの具体的なイメージは、ヒロセ電機のホームページを参照。

([http://www.hirose.co.jp/catalogj\\_hp/j21800074.pdf](http://www.hirose.co.jp/catalogj_hp/j21800074.pdf))

写真は HNC2-2.5P-8DS

(I C ピッチ 1 列 8 ピンコネクタ・メス)



## 2. 作業条件

(1) 競技時間 2 時間 30 分 (150 分)

(2) 競技委員から配布するもの

- ・『設計製作回路①』で使用する部品および材料等
- ・『制御対象回路③』
- ・コンテストで使用する部品の規格表
- ・AC100V コンセント (2 口)
- ・ソースリスト印刷用パソコンおよびプリンタ
- ・A 4 サイズ方眼紙

(3) 競技者が準備するもの

- ・『制御用コンピュータ②』および開発環境
- ・接続ケーブル A、接続ケーブル B、接続ケーブル C
- ・+ 5 V の電源 (Max1000mA 程度)
- ・工具類およびテーブルタップ
- ・ソースリスト提出用の記録媒体 (USB メモリまたは FD)
- ・筆記用具および定規・テンプレート類

工具類とは、各自の作業に必要なもので、はんだごて、こて台、ニッパ、リードペンチ、+ドライバ、テスタ、保護メガネ、基板支持台 等

(4) 競技者の服装等

- ・競技中は、各学校で使用している作業服を着用する。
- ・はんだ付けの作業時には、保護メガネを着用する。ただし、メガネをかけている場合はこの限りではない。

(5) 注意事項

- ①作業を行うにあたっては、安全に十分注意する。
- ②配布された部品および材料以外のものは、使用しない。
- ③規格表・命令表が必要な場合は各自で準備し、大会前日に承認を受ける。
- ④事前に製作したヘッダーファイルは、大会前日に申請し内容の承認を受ける。
- ⑤接続ケーブル A・B・C は、図 2・3・4 を参考に事前に製作し準備しておく。  
ただし、ケーブルの長さは自由である。
- ⑥ソースリストは、テキスト形式で記録媒体に保存・提出する。

### 3. 審査対象

- (1) 『設計製作回路①』の設計図
- (2) 『設計製作回路①』の製作基板
- (3) 仕様に対応する動作
- (4) プログラムのソースリスト
- (5) その他（作業態度等）

### 4. 採点基準

- (1) 採点項目と観点

項 目	配点	観 点
プログラミング技術	40	・ 動作 ・ 構造 ・ 書式 ・ 読みやすさ
組み立て技術	30	・ 工具類の使い方 ・ 部品処理 ・ はんだの状態 ・ 配線 ・ 配置
設計力	20	・ 正確さ ・ 配置 ・ 記号 ・ 文字
その他	10	・ 作業態度 ・ 作業工程
合 計	100	

- (2) 順位の決定方法

- ①合計得点の高い順に、1位、2位、3位…とする。
- ②同点の場合は、「プログラミング技術」の得点の高い選手を高位とする。
- ③「プログラミング技術」の得点も同点の場合は、「組み立て技術」の得点の高い選手を高位とする。
- ④さらに同点の場合は、「設計力」の得点の高い選手を高位とする。それでもなお同点の場合は、全体の完成度から順位を決定する。

- (3) その他

設計製作回路①の製作に関して、別紙『標準的なはんだ付けについて』を参照。。

## 5. その他

### (1) 鉛フリーはんだについて

無鉛（鉛フリー）はんだ（Sn-3.0Ag-0.5Cu、0.8mm）を使用する。

### (2) 動作確認について

プレ審査時に競技委員の指示に従い、競技者が操作して課題の動作確認を行う。

### (3) 設計製作回路・制御対象回路・当日の課題プログラムについて

『設計製作回路①』・『制御対象回路③』の回路図・使用部品の規格等については、事前公開しない。

また、当日作成する制御プログラムに関しても、事前公開はしない。

### (4) その他

大会の参考として、下記の全国情報技術教育研究会ホームページを参照。

<http://www.zenjouken.com/area/kantou/2008/index.html>

## 標準的なはんだ付けについて

### （１）はんだのぬれ性

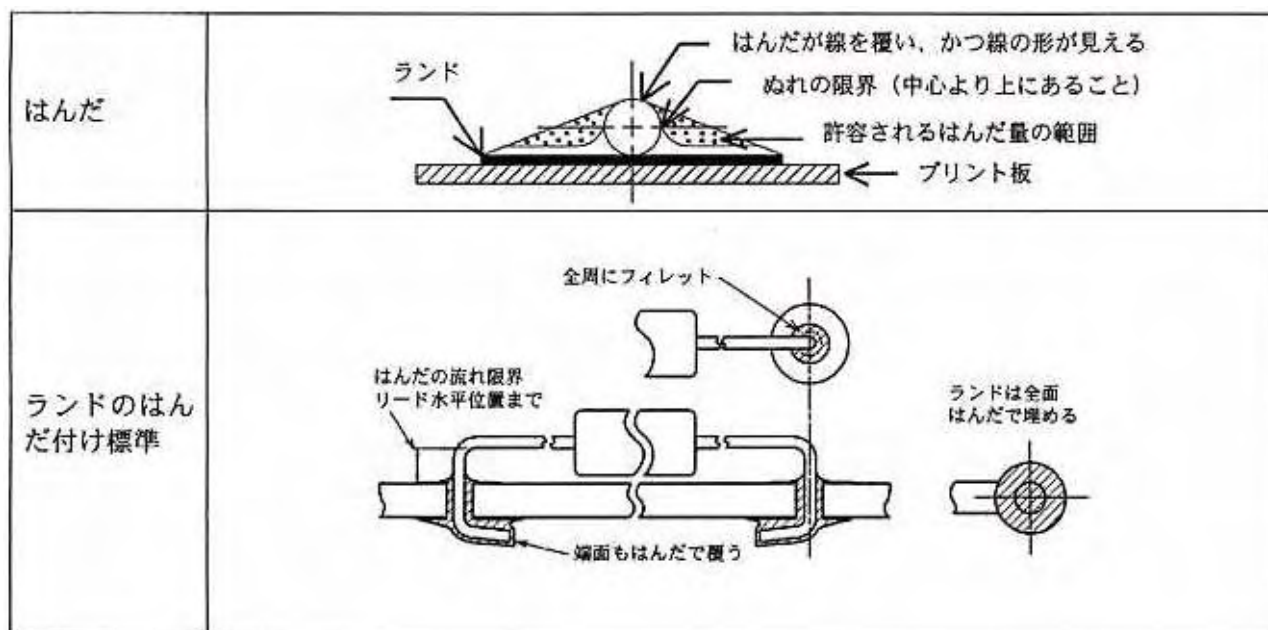
- イ．はんだが接合するリード線、銅箔によく流れ、長くすそを引いていること。
- ロ．部品穴のはんだ付けは、ランドの表面にはんだのぬれ性があること。

### （２）はんだの量

- イ．はんだの量は、部品リード線の折り曲げ部分、線の切り口等をはんだで覆い、かつ、肉厚が薄く線の形がわかるものとする。（図を参照）
- ただし、折り曲げず、かつ、切断せずに取り付ける部品リードのはんだ付けを行う場合は、リードの先端まで、全面はんだで覆わなくてもよい。

### （３）その他

- イ．部品端子の線材接合部は、穴あきのないようにはんだ付けすること。
- ロ．ランドのないところで、線又は部品リードを接続しないこと。  
（空中配線接続をしてはならない）
- ハ．ランドをはく離させないこと。
- ニ．ジャンパー線を用いず、裏面のみで配線をおこなうこと。

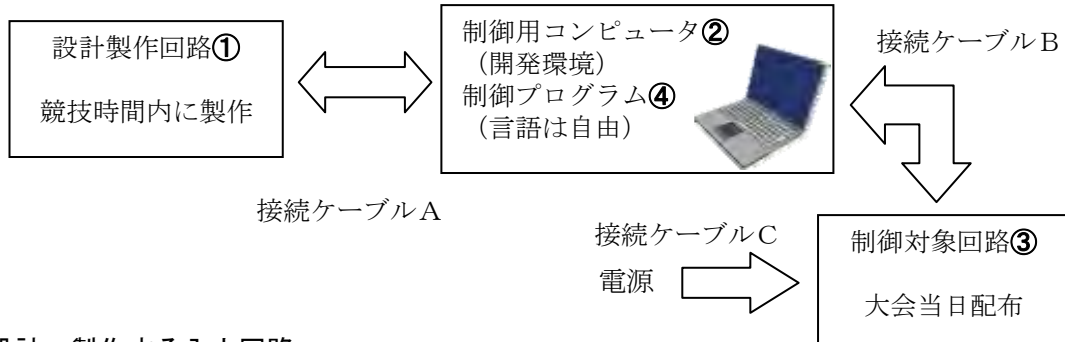




# 第9回高校生ものづくりコンテスト全国大会 電子回路組立部門 課題

## 1. システム構成

『制御用コンピュータ②』に『設計製作回路①』と『制御対象回路③』を接続し、その『制御プログラム④』を作成し、コンピュータ制御システムを完成させる。

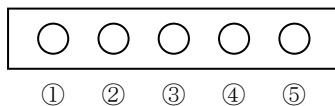


## 2. 設計・製作する入力回路

以下の条件を満たした回路を設計し、完成させなさい。

- ① 下図に示されたピン配置による入力回路を設計する。

I C ピッチ 1 列 5 ピン



①	GND	②	5V	③	PS
④	TSR	⑤	TSB		

PS : 透過型フォトインタラプタ  
TSR : タクトスイッチ赤  
TSB : タクトスイッチ青

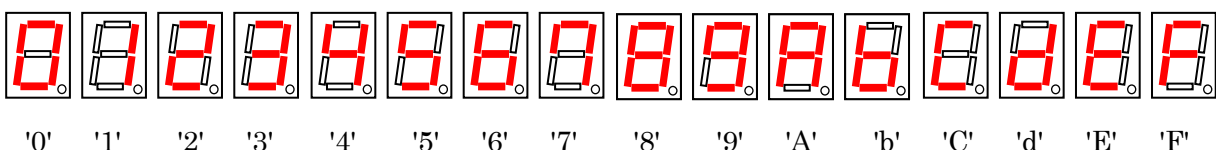
- ② 支給する方眼紙(A4)に回路図を書き提出する。
- ③ 支給された部品を使って、設計した回路を製作する。フォトインタラプタはユニバーサル基板上の部品面に直接取付けること。

## 3. 制作するプログラム

課題1～7のプログラムを完成させなさい。

注意事項

- ① 文中の表現で「ON」はタクトスイッチ (TSB または TSR) を押している状態、「OFF」はタクトスイッチが押されていない状態をいう。「ON/OFF」はタクトスイッチを押して離すことをいう。
- ② 7セグメント LED の表示は次のように点灯すること。(図の赤が点灯状態)  
また 16 進数の B と D についても下のように小文字で点灯させること。



- ③ 課題の中にある動作概要図については大まかな流れが図示してあるので、詳細な条件については課題文章のとおり動作すること。

## 課題 1

- (1) 左側の7セグメントLEDは、タクトスイッチ TSB が ON のとき'A'を表示し OFF のときは消灯する。
- (2) 右側の7セグメントLEDは、タクトスイッチ TSR が ON のとき'b'を表示し OFF のときは消灯する。
- (3) 2つのモータはいずれも停止していること。
- (4) フォトインタラプタの状態に関係なく動作すること。

## 課題 2

- (1) 左側7セグメントLEDがフォトインタラプタとタクトスイッチの状態によって次のように表示されること。

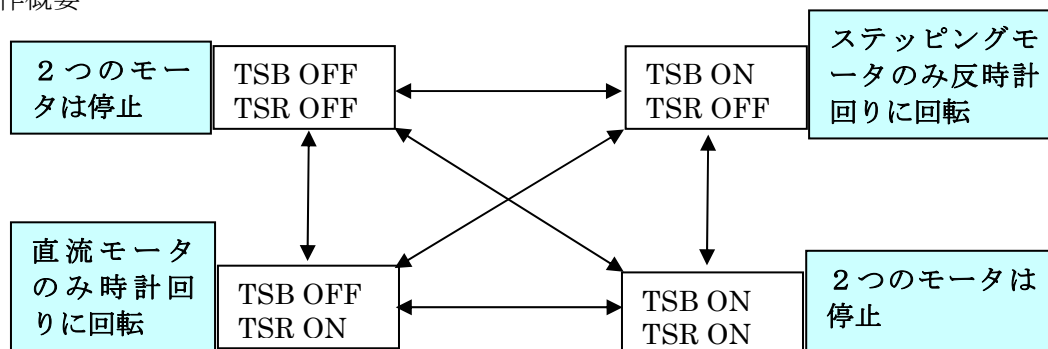
フォトインタラプタ	タクトスイッチ TSB	タクトスイッチ TSR	左側 LED
透過	OFF	OFF	0
透過	OFF	ON	1
透過	ON	OFF	2
透過	ON	ON	3
遮断	OFF	OFF	4
遮断	OFF	ON	5
遮断	ON	OFF	6
遮断	ON	ON	7

- (2) 2つのモータはいずれも停止していること。
- (3) 右側7セグメントLEDは常に消灯していること。

## 課題 3

- (1) タクトスイッチ TSB のみが ON のときステッピングモータが反時計回りに回転する。
- (2) タクトスイッチ TSR のみが ON のとき直流モータが時計回りに回転する。
- (3) タクトスイッチの状態が上の (1)、(2) 以外のときのときはステッピングモータ、直流モータともに停止状態になる。
- (4) モータの回転速度はいずれも目視で確認できること。
- (5) 2つの7セグメントLEDは消灯していること。

動作概要



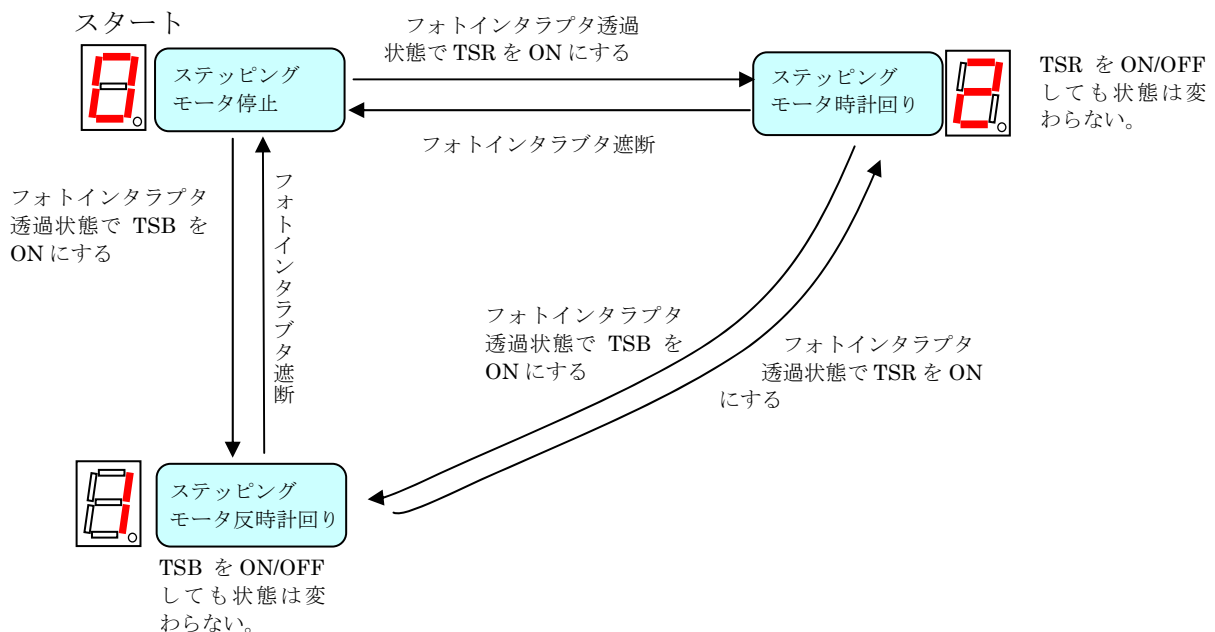
## 課題 4

- (1) フォトインタラプタの遮断回数を右側の 7 セグメント LED に表示する。  
ただし遮断回数の表示は 16 進数 1 桁で 0 から F までとし、F を超えたら 0 に戻ってカウントを継続すること。
- (2) プログラムスタート時は右側 7 セグメント LED は 0 が表示されているものとする。  
遮断回数のカウント方法は遮断→透過で 1 回とする。
- (3) 2つのモータはいずれも停止していること。
- (4) 左側 7 セグメント LED は常に消灯していること。
- (5) タクトスイッチ TSB, TSR の状態に関係なく動作すること。

## 課題 5

- (1) プログラムスタート時、ステッピングモータは停止している。
- (2) フォトインタラプタが透過状態のとき
  - (a) タクトスイッチ TSB を一度 ON にするとステッピングモータが反時計回りに回転し、タクトスイッチ TSB を OFF にしてもステッピングモータはそのまま回転を継続する。
  - (b) タクトスイッチ TSR を ON にするとステッピングモータが時計回りに回転し、タクトスイッチ TSR を OFF にしてもステッピングモータはそのまま回転を継続する。
- (3) フォトインタラプタが遮断状態のとき、  
タクトスイッチの状態に関わらずステッピングモータは停止する。
- (4) 停止したステッピングモータを再び動作させるためには (2) の (a)、(b) の状態にする。
- (5) 左側 7 セグメント LED は常に消灯していること。直流モータは常に静止していること。
- (6) 右側 7 セグメント LED はステッピングモータの状態によって次のように点灯すること。
  - (a) 停止状態のとき '0' を表示する。
  - (b) 反時計回りのとき '1' を表示する。
  - (c) 時計回りのとき '2' を表示する。
- (7) フォトインタラプタが透過状態で TSR, TSB が同時に ON になった場合は、(2) の (a) または (b) のどちらか一方の動作をすること。

### 動作概要

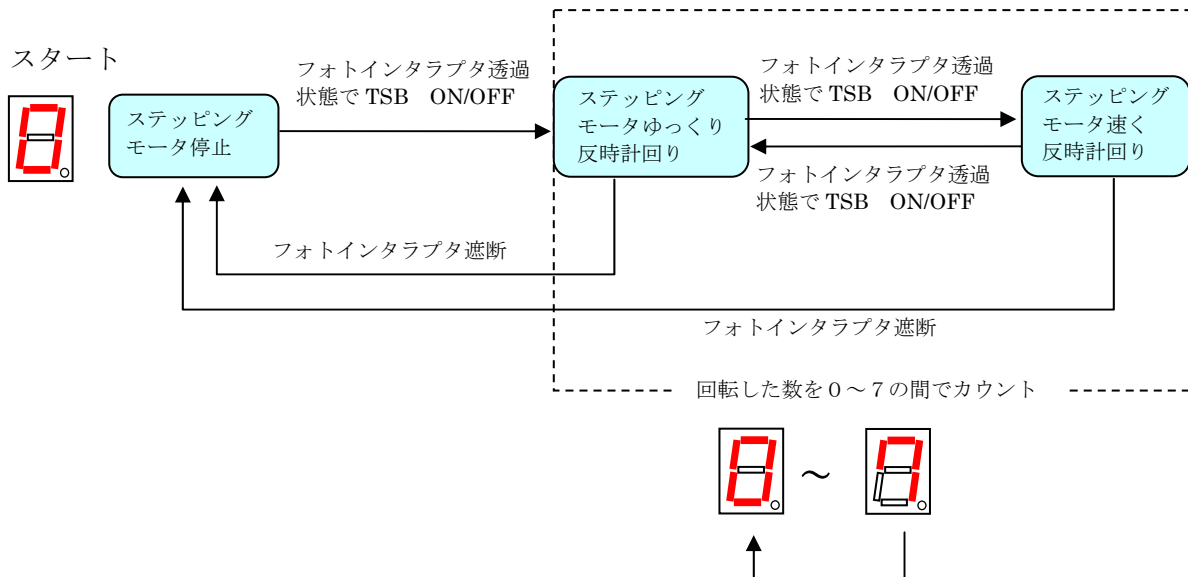


## 課題 6

- (1) プログラムスタート時、2つのモータは停止していること。
- (2) フォトインタラプタが透過状態のとき  
タクトスイッチ TSB を ON/OFF するたびに状態が次の(a)と(b)の間を移る。
  - (a) ステッピングモータが反時計方向にゆっくりとまわる。
  - (b) ステッピングモータが(a) よりも反時計方向に速くまわる。なお最初は(a)から始めるものとする。
- (3) フォトインタラプタが遮断状態のとき  
ステッピングモータは停止する。停止したモータは(2)の条件を満たすまで動作しない。
- (4) 右側7セグメント LED はステッピングモータの回転した数をカウントする。  
プログラムスタート時は0を表示する。7を超えたら再び0からカウントする。  
モータが途中で停止した場合はカウンタの値をそのまま表示しつづけ、回転が再開したらカウントを継続する。
- (5) 左側7セグメント LED は常に消灯していること。  
ステッピングモータは目視でゆっくり回ると速く回る違いが区別できること。

### 動作概要

スタート



## 課題 7

(1)プログラムスタート時2つのモータは停止、右側7セグメントLEDはプログラムスタート後ただちに16進数0～Fまでを約1秒間隔で繰り返し表示し続ける。

(2)フォトインタラプタが透過状態のとき

(a)タクトスイッチ TSB を一度 ON にすると右側7セグメントLEDのカウンタが0から7までの間は、ステッピングモータが時計回り、8からFまでは反時計回りに回転すること。  
また直流モータは停止させること。

(b)タクトスイッチ TSR を一度 ON にすると右側セグメントLEDのカウンタが0から7までの間は、直流モータが時計回り、8からFまでは反時計回りに回転すること。  
またステッピングモータは停止させること。

(c)モータは同時に二つ動かさない。

(a)ではステッピングモータが停止するまで TSR の ON を受け付けないこと。

(b)では直流モータが停止するまで TSB の ON を受け付けないこと。

(3)フォトインタラプタを遮断したとき

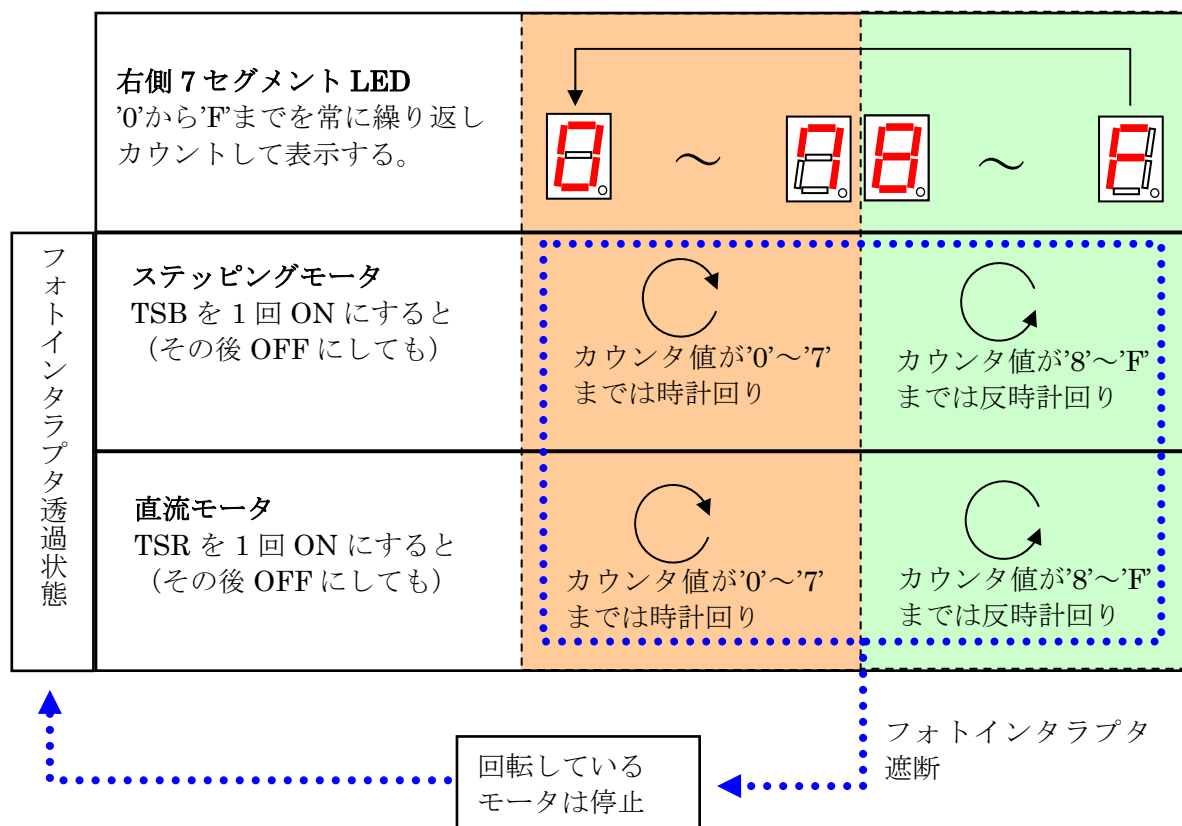
回転しているステッピングモータまたは直流モータは停止させること。

停止したモータは(2)の条件を満たすまで動かないこと。

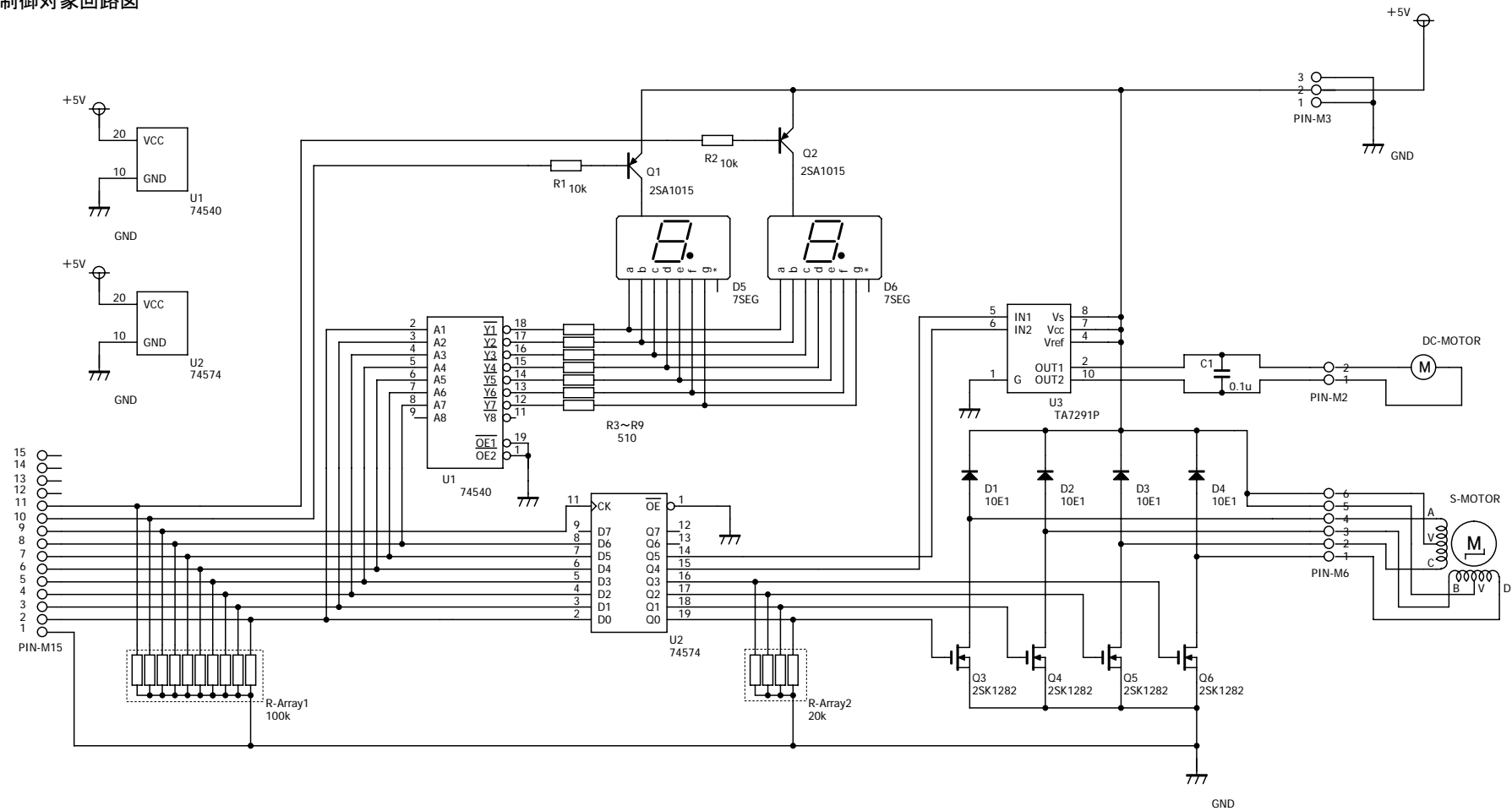
(1)のカウンタは継続させること。

(4)左側7セグメントLEDは常に消灯していること

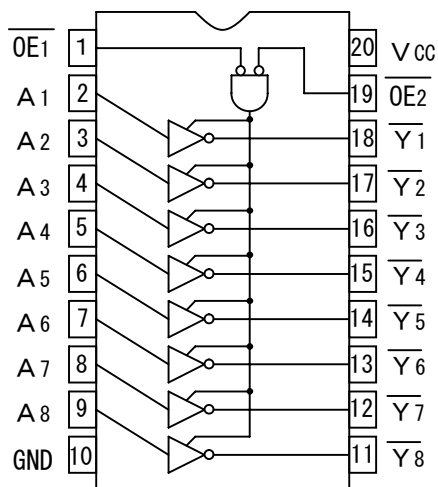
動作概要



制御対象回路図



74HC540



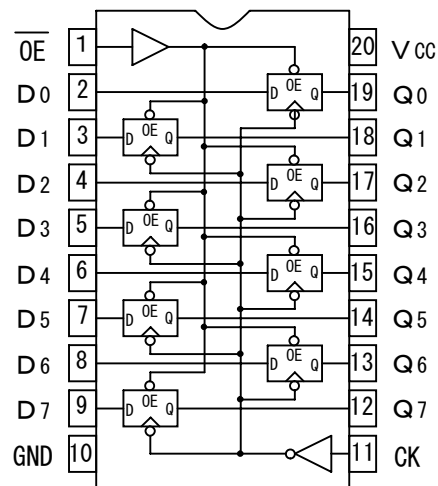
Function Table

Input			Output
OE1	OE2	An	Yn
L	L	L	H
L	L	H	L
x	H	x	Z
H	x	x	Z

Notes

H:HIGH voltage level  
L:LOW voltage level  
x:don't care  
Z:high impedance OFF-state

74HC574



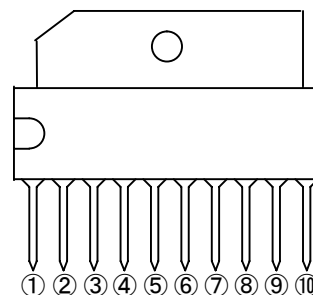
Function Table

Input			Output
OE	CK	Dn	Qn
L	↑	H	H
L	↑	L	L
H	↓	x	Qn
H	x	x	Z

Notes

H:HIGH voltage level  
L:LOW voltage level  
x:don't care  
Z:high impedance OFF-state  
Qn:Hold

TA7291P



Pin Explanation

①	②	④	⑤	⑥	⑦	⑧	⑩
GND	OUT1	Vref	IN1	IN2	VCC	VS	OUT2

③⑨:NC

Function Table

INPUT		OUTPUT		MODE
IN1	IN2	OUT1	OUT2	
L	L	Z	Z	STOP
H	L	H	L	CW/CCW
L	H	L	H	CCW/CW
H	H	L	L	BRAKE

Notes

H:HIGH voltage level  
L:LOW voltage level  
Z:high impedance

Stepping Motor

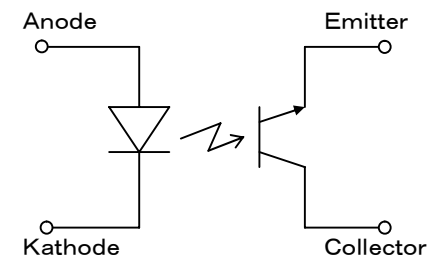
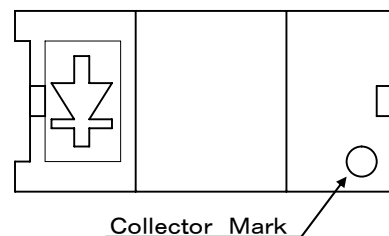
48M048C1U-N

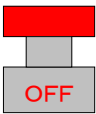
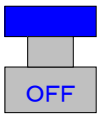
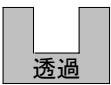
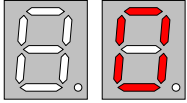
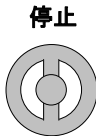
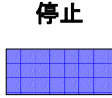
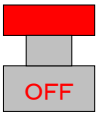
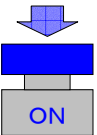
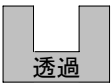
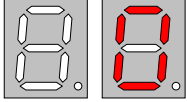
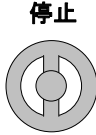
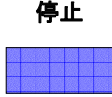
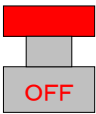
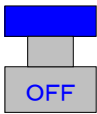
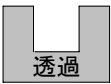
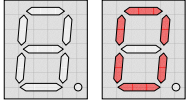
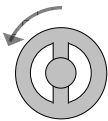
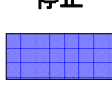
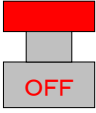
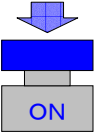
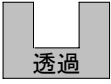
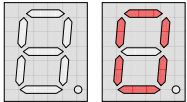
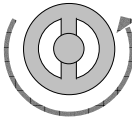

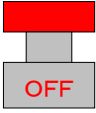
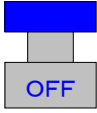
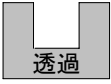
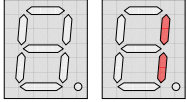
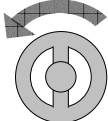

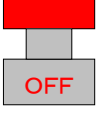
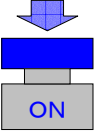

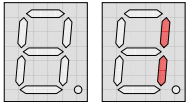
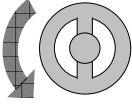
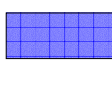
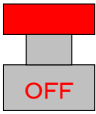
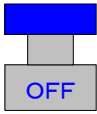
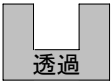
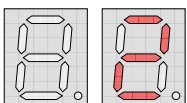
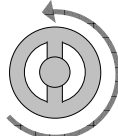
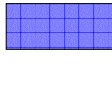
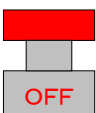
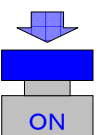

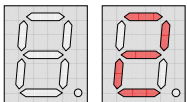
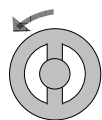

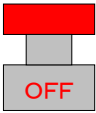
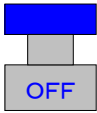
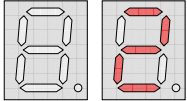
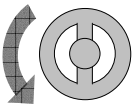
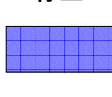
DC Operating Voltage 5V

Step Angle 7.5°

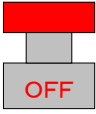
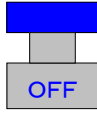
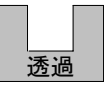
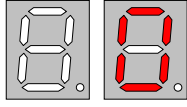
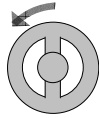
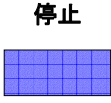
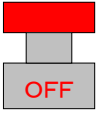
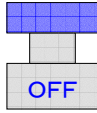
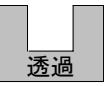
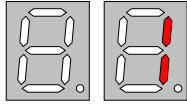
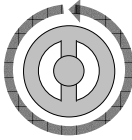
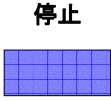
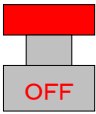
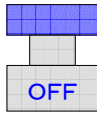
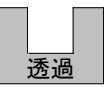
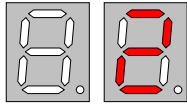
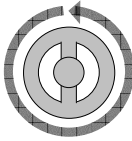
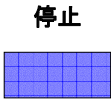

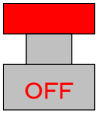
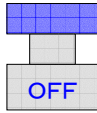
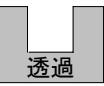
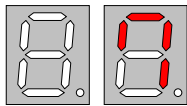
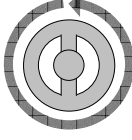
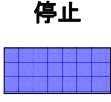
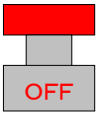
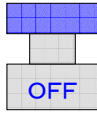
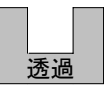
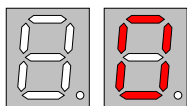
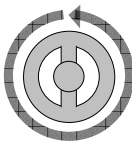
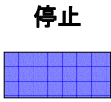
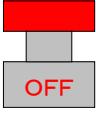
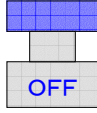
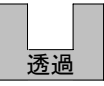
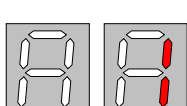
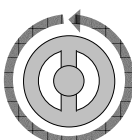
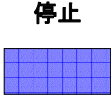
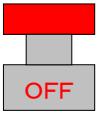
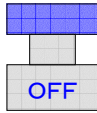
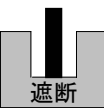
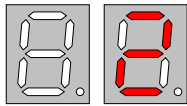
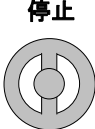

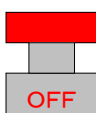
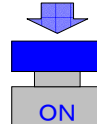

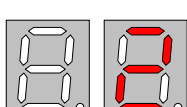
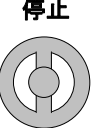
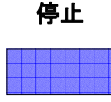
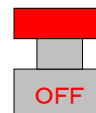
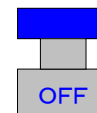

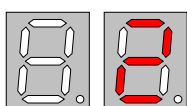
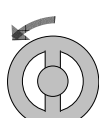
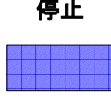
Excitation method 4Phases Unipolar

SG206



課  題  6	TSR	TSB	PS	7segLED	STM	DCM	CK
							<input type="checkbox"/>
							<input type="checkbox"/>
							<input type="checkbox"/>
							<input type="checkbox"/>
							<input type="checkbox"/>
							<input type="checkbox"/>
							<input type="checkbox"/>
							<input type="checkbox"/>
							<input type="checkbox"/>



課  題  6	TSR	TSB	PS	7segLED	STM	DCM	CK
							
							<input type="checkbox"/>
							<input type="checkbox"/>
							
							
							<input type="checkbox"/>
							
							<input type="checkbox"/>
							<input type="checkbox"/>
							<input type="checkbox"/>

## 競技上の注意

- 1 入力回路の製作は、下表 1 に示す支給部品を使用すること。

表 1 支給部品表

番号	名 称	型式等	個数
1	ユニバーサル基板	ICB-293 (Sunhayato)	1
2	透過型フォトインターラプタ	SG206 (KODENSHI)	1
3	タクトスイッチ	赤 (秋月電子通商)	1
4	タクトスイッチ	青 (秋月電子通商)	1
5	ピンヘッダ	5 Pin ストレート	1
6	抵 抗	10k $\Omega$ 1/4W	2
7	抵 抗	180 $\Omega$ 1/4W	1
8	抵 抗	22k $\Omega$ 1/4W	1
9	ネ ジ	M3 $\times$ 10mm	4
10	ナット	M3	4
11	ゴム足	$\phi$ 14(相当品)	4
12	スズメッキ線	$\phi$ 0.5	1
13	鉛フリーハンダ	Sn-3Ag-0.5Cu 0.8mm	1
14	遮光チェック用アルミ板	20mm $\times$ 110mm	1

- 2 制御プログラムのソースリストの提出は、下表 2 に示すファイル名とする。

表 2 ファイル名

問	提出ファイル名の例			
1	kadai1.c	kadai1.asm	kadai1.txt	等
2	kadai2.c	kadai2.asm	kadai2.txt	等
3	kadai3.c	kadai3.asm	kadai3.txt	等
...	...			

- 3 DCモータおよびステッピングモータの回転は目視および触って確認できること。

## 第9回高校生ものづくりコンテスト全国大会

### 電子回路組立部門 Q & A

<H21.11.05>

Q 1 制御対象回路③は、今回支給となっていますが、昨年同様どこかのメーカーから提供していただけるのでしょうか？メーカー名、型名、価格などもわかればお願いします。

秋田県立大曲工業高等学校

A 1 大会要項に記載の通り、制御対象回路③の回路図・使用部品の規格等について事前公開はいたしません。

Q 2 (1)3年前の埼玉県大会の要綱にもどったのはどうしてでしょうか？  
(2)静岡県や大阪府大会では、実施要綱変更の際や技能検定2級レベルのハード重視に変更になったときは、理由を明確に文書で提示していたのですが今回はないのでしょうか？  
(3)今後とも、ソフト重視でいくことになるのでしょうか？  
(4)回路図作成がまた、復活したのであれば、回路図に使用する図記号(どの回路図を使用するのか)を提示していただけないでしょうか？  
(5)使用コネクタですが、一般に入手するのが面倒なものであり、ピン間隔がインチではなく、通常使用する基板のインチ間隔にはなじまない。入手が簡単なボックスタイプのコネクタに変更していただけないでしょうか。  
(6)ソフト、ハード採点基準をもっと明確に提示してほしい。(生徒に指導する関係上納得できるようにしてください。)

山形県立山形工業高等学校

A 2 (1)要綱に関しては、静岡及び大阪大会時のものと、ほぼ同じ程度と考えています。  
競技課題は、専門高校で学習するレベルのものを予定しています。  
(2)文書での提示は行いません。表面実装の部品を使用する予定がないので、あえて技能検定2級レベルは求めません。  
(3)設計製作回路および制御プログラムが課題となりますので、ハードおよびソフトの両面を問うものと考えています。  
(4)図記号は、JIS C 0617を基本とします。  
(5)使用するピンヘッダ・コネクタは、ユニバーサル基板用のインチ間隔のものです。申し訳ありませんが、本大会では、課題要項記載のものを使用します。  
(6)採点基準は、観点を示していますので、それを参考にご指導頂けたらと思います。

Q 3 (1)ネットで見てますが、ものづくり大会の制御回路、製作回路について、当日まで分からないとのことですが、製作について、何かヒントになることは教えていただけないでしょうか。  
(2)要項にあった接続ケーブルコネクタは、ヒロセ電機のコネクタを使用していますがそこに用いる電線も、コネクタのカタログに書かれている指定の電線でなければいけないのでしょうか。

新潟県立柏崎工業高等学校

- A 3 (1)「電子回路組立部門 課題」 5.その他(3)に記載のとおりです。
- (2)「電子回路組立部門 課題」 1.課題(5)で示した接続ケーブル・コネクタを使用します。電線は接続可能なものを準備して下さい(長さは自由です)。イメージは、ヒロセ電機のホームページを参照下さい。

- Q 4 (1) 競技者準備物について  
予備のケーブル・開発環境等の持参は可能か。
- (2) プログラミング技術の採点基準について  
プログラミング技術の観点が具体化されていない。書式・読みやすさは主観によって観点が変わってくることが考えられる。具体的な文例を出していただきたい。
- (3) 設計力の採点基準について  
製図を行うのであれば、用紙を教えてください(○ミリ方眼or 無地ケント紙など)。記号・枠などの書き方の基準は？考えられる部品記号一覧と、設計ガイドラインを示していただきたい。
- (4) 設計製作回路① について  
課題に示されている部品以外は使用しないのか？たとえば可変抵抗などによるアナログ入力は不要か？
- (5) 課題の提出について  
開発環境・設計製作回路①・制御対象回路③ を1枚のパネル上に固定する必要はあるか？
- (6) 制御対象回路③ の公開について  
大会中のどの時間で公開されるのか。14日に公開されれば、ある程度プログラムの準備が可能である。しかし15日の競技時間内であれば、設計製作回路やプログラム課題作成の時間を使うことが考えられる。
- (7) プログラム課題の提出・作成方法について  
プログラム課題は課題ごとに独立したソースファイルでの提出が可能か。

岐阜県立岐阜工業高等学校

- A 4 (1) 予備の工具・制御用コンピュータ・接続ケーブル・開発用コンピュータは、持参しても構いません。
- (2) 本競技においては、制御用コンピュータの制限はなく、また、使用するプログラミング言語は自由です。したがって、プログラミング技術の採点は、動作・構造・書式・読みやすさの4つの観点により、それぞれの環境に応じて総合的に評価します。
- ① 動作・・・各課題ごとに、完全に動作しているか。
- ② 構造・・・プログラムの関数、サブルーチンなどが機能ごとに適切に構造化されているか。
- ③ 書式・・・改行、入れ子の字下げ等、コーディングスタイルが適切であるか。
- ④ 読みやすさ・・・プログラムがわかりやすく表現されているか。
- (3) 1ミリ方眼紙を使用し、図記号は、JIS C 0617 を基本とします。

- (4) 競技に必要な部品は、すべて競技の直前に配布します。また、部品の仕様書は、競技開始時に出题します。配布する部品に関しては、事前の公開はいたしません。
- (5) 必要はありません。
- (6) 制御対象回路③は競技課題です。制御対象回路及び回路図・部品の仕様書は競技開始時に出题します。
- (7) プログラム課題は、課題ごとに評価・採点します。独立したソースファイルを提出することになります。

- Q 5** (1)1. 課題 (5)図2(設計製作回路)と図3(制御対象回路)の入出力は、TTL 論理レベルの信号に準拠していますか。制御コンピュータが学校教材の場合、A/D や D/A といった機能がないこともありますので、配慮が必要かと思われます。加えて、制御コンピュータの選定や準備に大きな影響がありますので公開されるべき情報であると考えます。
- (2)2. 作業条件(5)③の規格表や命令表の持ち込み資料のガイドラインはありますか。また、前日の承認はどのような方法で行われますか。手書きのノートは持ち込み可能でしょうか。
- (3)同④の事前作成のヘッダファイルの作成可能範囲などのガイドラインはありますか。また、申請や確認の方法はどの様に行いますか。付随して、練習で制作したプログラムの参照及び活用は可能ですか。
- (4)5. その他(2)のプレ審査は競技時間内、それとも時間外で行いますか。

長野県松本工業高等学校

- A 5** (1)制御用コンピュータと設計製作回路、制御用コンピュータと制御対象回路は、TTL 論理レベルの信号に準拠しています。アナログ入出力に関する配慮は、必要有りません。
- (2)電子部品や開発環境等の規格表や命令表のみ持ち込みを認めます。手書きおよびワープロで作成した書類等は、持ち込み禁止です。開発用コンピュータの記憶装置内の格納もこれに準じます。
- (3)ヘッダファイルの内容は、ポートの定義等とします。プログラム類は禁止です。また、競技中に事前に制作したプログラムを参照する行為は禁止です。競技において、開発環境に付属したヘッダファイルのみ用いる場合は、ヘッダファイルの申請は不要です。自作のヘッダファイルを用いる場合は、内容を確認しますので、必ず申請し使用許可を得てください。
- (4)プレ審査は、時間外に行います。

- Q 6** ヘッダファイルの事前審査について、ここでいうヘッダファイルとは何を意味するのでしょうか？たとえば、よく使う処理をまとめてヘッダファイルとして持ち込むことが可能とも読めます。またヘッダファイルは承認となっていますが、申請した後、持ち込み禁止となることもありえるのでしょうか？そうすると生徒への指導方法が変わるので、ヘッダファイルについての具体的な指示があると幸いです。

北海道旭川工業高等学校

- A 6** A 5 (3)のとおりです。

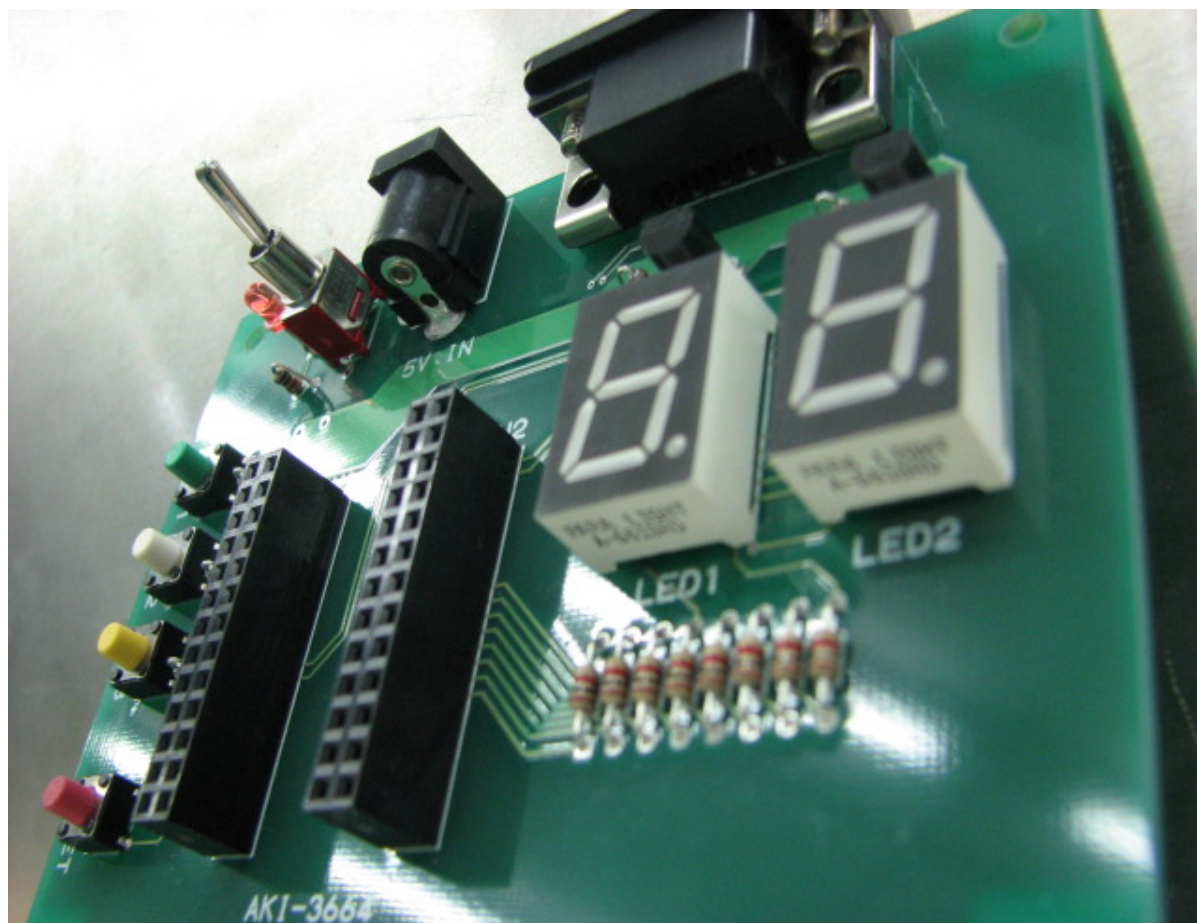
高校生ものづくりコンテスト全国大会 電子回路組立部門 採点表

番 号				1	2	3	4	5	6	7	8	9	10
学 校 名													
学 年													
氏 名													
プログラミング技術	動 作		25										
	構 造		5										
	書 式		5										
	読 み 易 さ		5										
	計		40										
設計力	回 路	正 し さ	12										
		配 置	2										
		記 号	2										
		文 字	2										
		配 線 図	2										
	計		20										
組立技術	作 業（工具の使い方・手袋）		6										
	出来栄え	部 品（足処理、曲方）	6										
		半 田（形状、ブリッジ、ツヤ）	6										
	実 装	配 線	6										
		配 置	6										
	計		30										
その他	計		10										
合 計			100										
順 位													

100点満点：プログラミング技術50点・設計力20点・組立技術20点・その他（安全作業の配慮等）10点

平成21年度高校生ものづくりコンテスト  
電子回路組立部門 埼玉県予選会

## 電子回路の組立て(はんだ付け)のポイント



※写真は『平成20年度ものづくり埼玉大会』の組立基板

### 講 師

UDトラックス(株) 技能五輪電子機器組立て

・田中 耕介

氏名		学校名	
----	--	-----	--

## はじめに

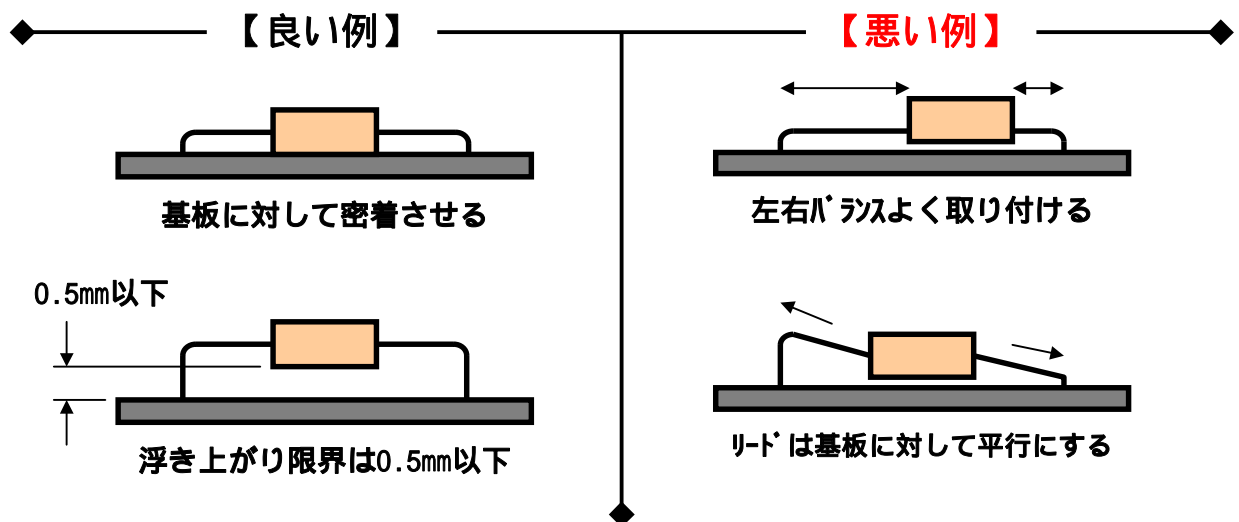
『ものづくりコンテスト埼玉大会』の電子回路の組立てには、基板の表面に電子部品を取り付ける「部品取り付け」と裏面にはんだ付けを行う「はんだ付け」の二つがあります。  
本資料では、電子回路の組立て(はんだ付け)のポイントを説明します。

## 部品取り付け (基板の表面に電子部品を取り付ける)

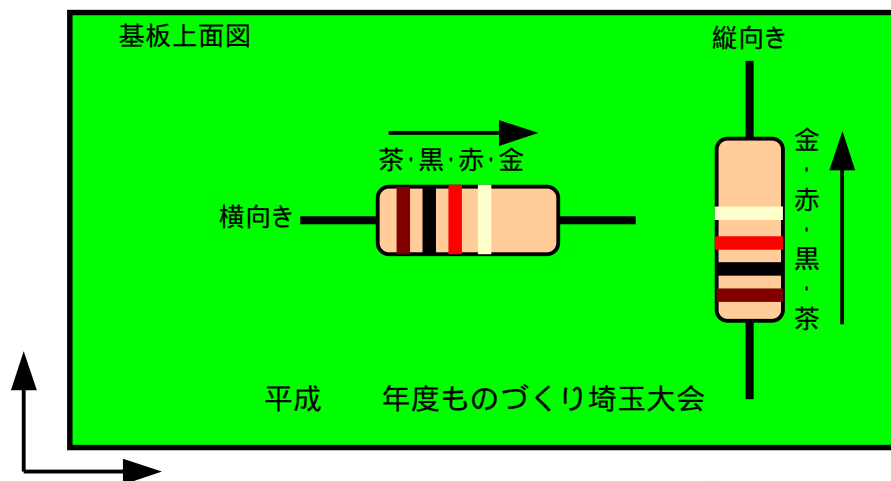
### 抵抗を取り付ける

#### 【水平に取り付ける場合】

- 抵抗を基板に対して水平に取り付ける場合は、基板に**密着させる**ように取り付けること。  
尚、浮き上がり限界は0.5mm以下とする。
- 抵抗を基板に取り付けた後のリード部分(部品の脚)の長さは、左右バランスよく曲げ、基板に対して平行に伸ばすこと。



- 抵抗のカラーコードは、基板を自分に対して正面に向けて、**横向きの場合は「左から右」**へ、**縦向きの場合は「下から上」**へ読み取れるように取り付けること。

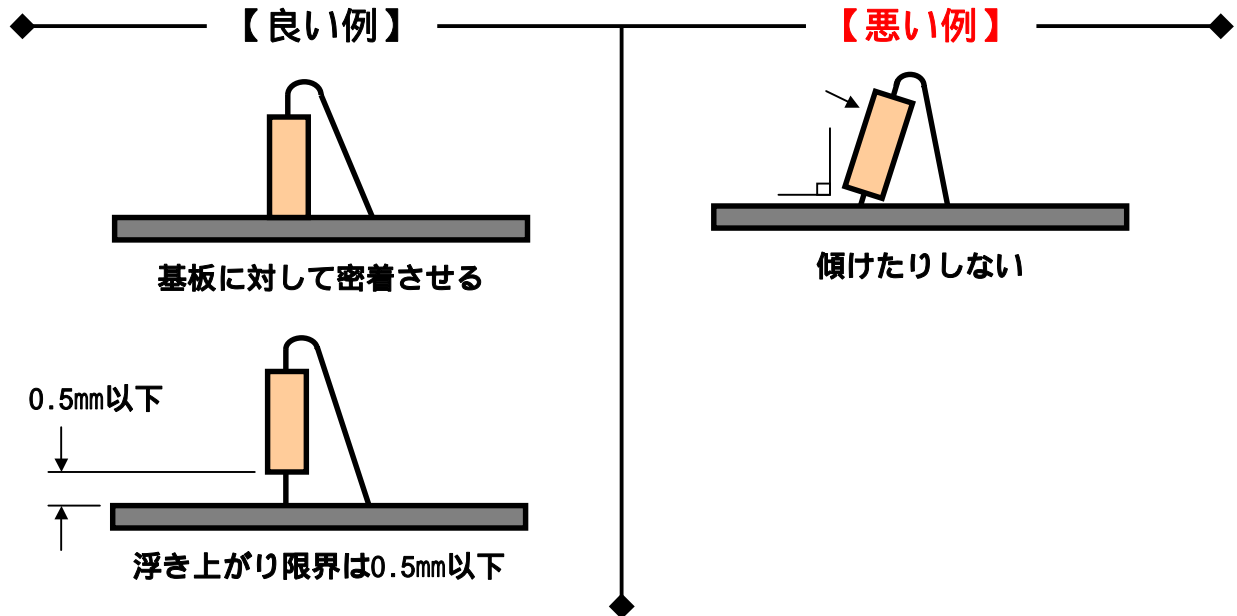


例) 1[k] 抵抗を基板に取り付ける場合

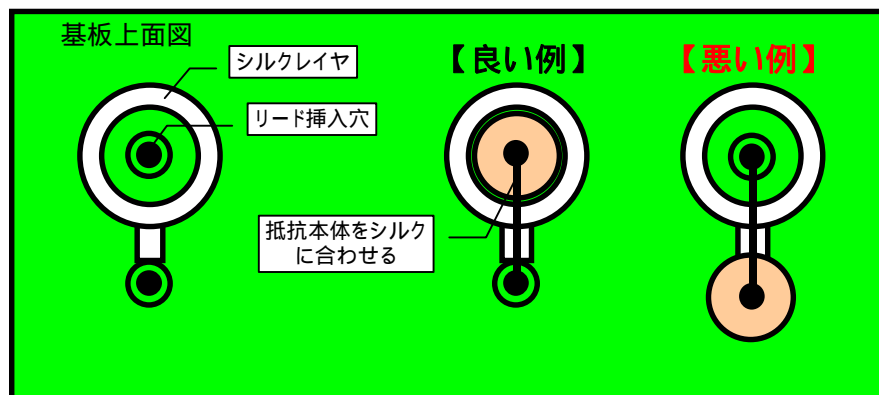


### 【垂直に取り付ける場合】

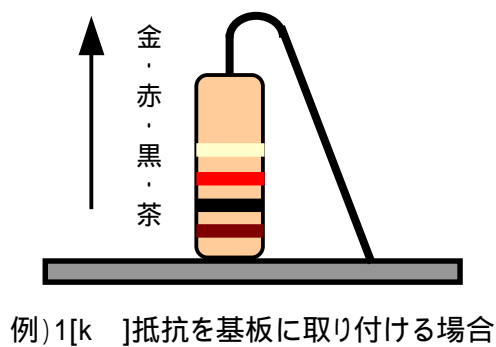
- 抵抗を基板に対して垂直に取り付ける場合は、基板に**密着させる**ように取り付け、傾けたりしないこと。  
尚、浮き上がり限界は0.5mm以下とする。



- 抵抗は、シルクレイヤの描いてある方へ『抵抗本体』を取り付けること。

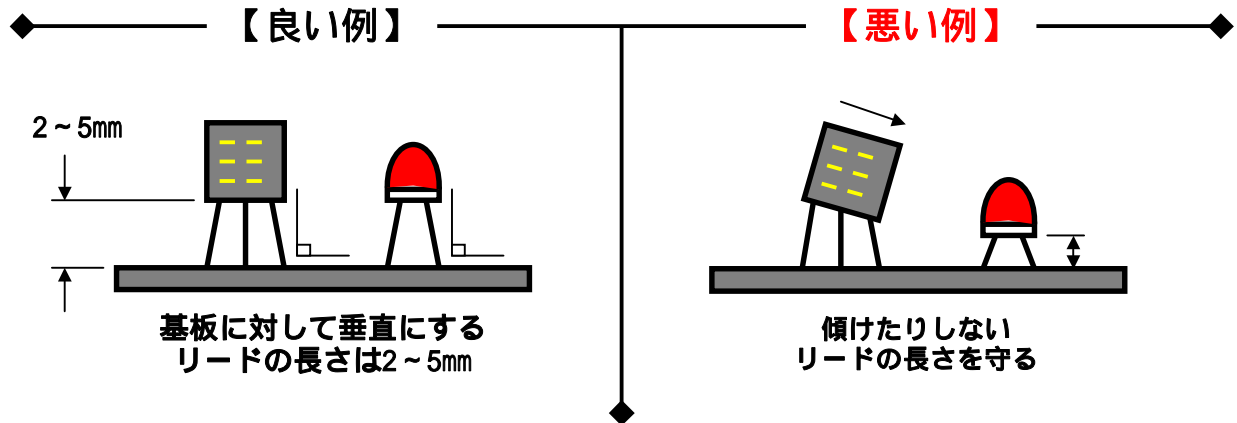


- 抵抗のカラーコードは、**基板を側面から見て「下から上」**へ読み取れるように取り付けすること。



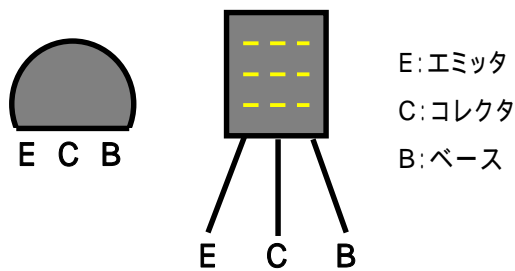
## トランジスタ・LEDを取り付ける

- ・ トランジスタ・LEDは、基板に対して垂直に取り付けること。
- ・ 部品下面から基板までのリードの長さは2～5mmとする。



- ・ トランジスタ・LEDは、シルクレイヤに従い極性に注意して取り付けること。

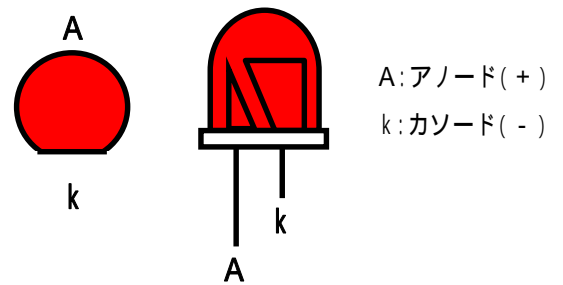
【トランジスタの極性】



見方1: 上面から見て切り欠きがある方を正面に  
向けて、左から順番に「ECB」

見方2: 側面から見て型式の書いてある方を正面に  
向けて、左から順番に「ECB」

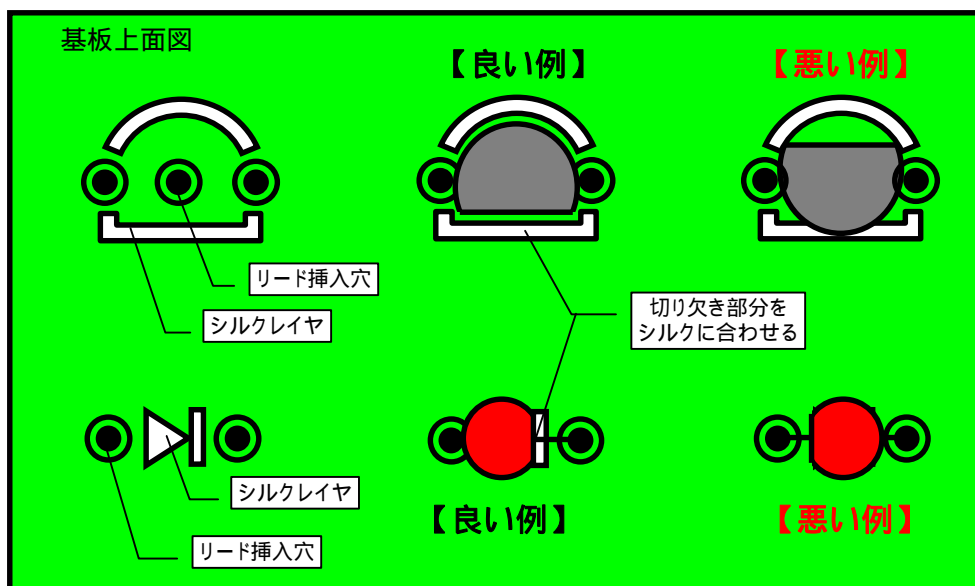
【LEDの極性】



見方1: 上面から見て切り欠きがある方が  
k(カソード(-))

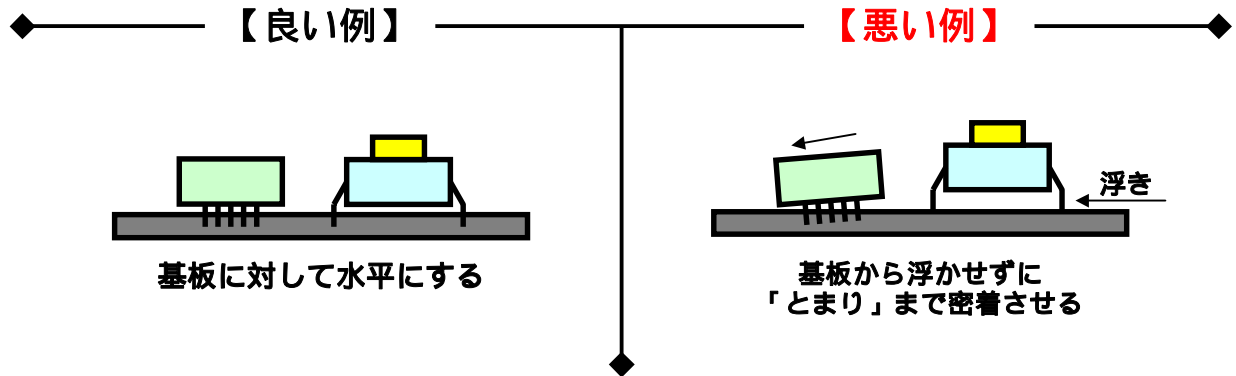
見方2: 側面から見てリード部分が長い、  
もしくは、内部素子が三角形になって  
いる方がA(アノード(+))

**極性の見方については、図解とは異なる部品も存在する**



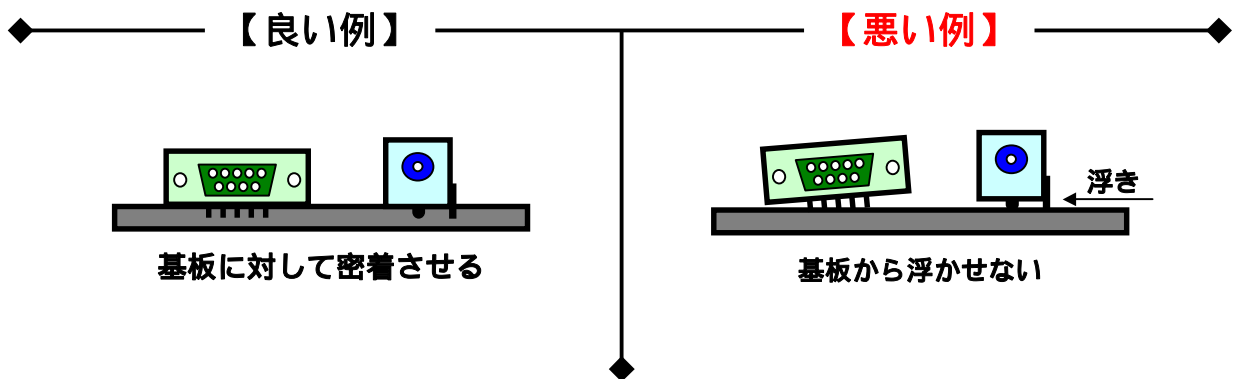
## 7セグメントLED・スイッチ類を取り付ける

- ・ 7セグメントLED・スイッチ類は、基板に対して水平に取り付けること。
- ・ 基板から浮かせずに「とまり」まで密着させること。



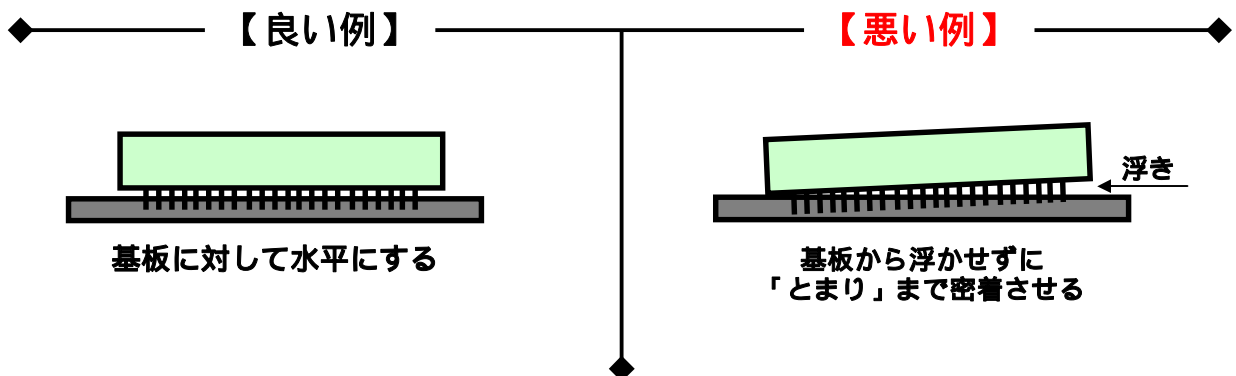
## D-sub9ピンコネクタ・DCジャックを取り付ける

- ・ D-sub9ピンコネクタ・DCジャックは、基板に対して密着させるように取り付けること。



## ピンソケット類を取り付ける

- ・ ピンソケット類は、基板に対して水平に取り付けること。
- ・ 基板から浮かせずに「とまり」まで密着させること。

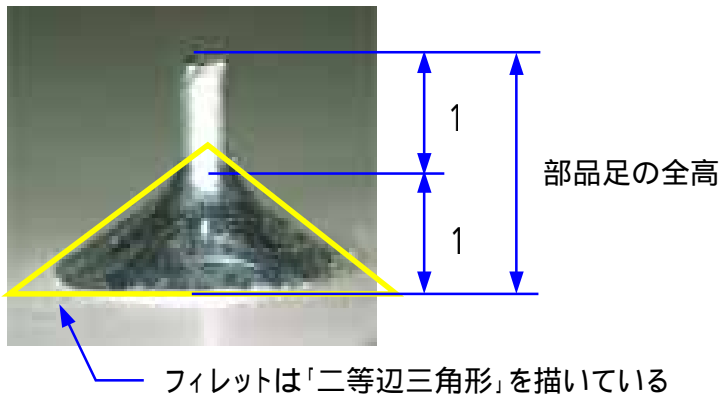


## はんだ付け (基板の裏面にはんだ付けを行う)

### はんだ付けの量、フィレット

- ・ はんだを部品の足に盛り付ける量は、部品足の全高の約半分程度を目安とすること。(1:1)  
(7セグメントLEDのような部品足の長いものは、部品足の全高の約1/3程度を目安とする。)
- ・ はんだのフィレットは、側面から見たときに部品の足を中心として「二等辺三角形」を描くような形状が良い。

#### 【良い例】



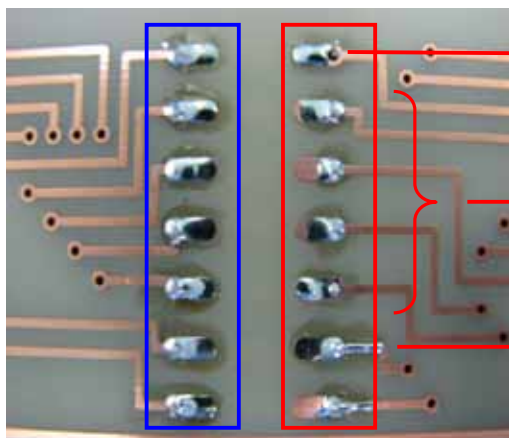
#### 【悪い例】



### はんだの付けぬれ性

- ・ はんだをランドに付けた時のぬれ性は、対象となるランドの表面が全てはんだで覆われていること。
- ・ はんだづけ対象外となるパターンにはんだがはみださないようにする。  
(但し、パターンに絶縁用のレジスト加工(緑色のコーティング)が施してあり場合は、この限りではない)

#### 【良い例】 【悪い例】



リード挿入穴まではんだがぬれていない

ランドの表面が露出している

ランドの表面が全てはんだで覆われているが、パターンにはんだがはみ出している

## はんだ付けのその他の要素

- ・ はんだの量、フィレット、ぬれ性以外にも、「コゲ、角、ブリッジ、色、艶(つや)」などの要素がある。

### 【良い例】



良いはんだづけ

### 【悪い例】



コゲがある



角がある



ブリッジしている



色がくすんでいる



熱を加えすぎて  
艶がなくなっている

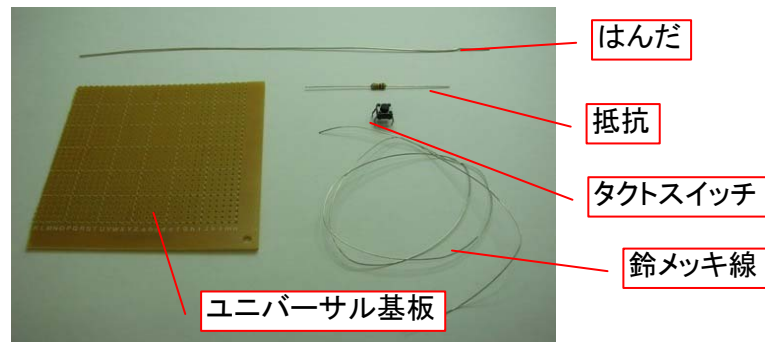


ランドがはがれている

## 『ユニバーサル基板でのタクトスイッチと抵抗の配線方法』

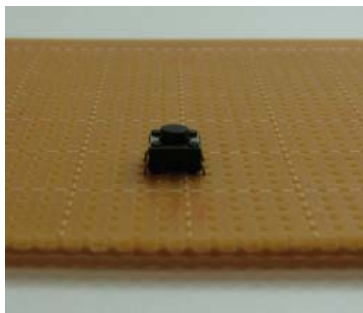
- ここでは、基板製作の一例としてユニバーサル基板にタクトスイッチと抵抗を取り付け、その二つを鈴メッキ線で配線する方法を説明します。

### ①以下の部品を用意する



### ②タクトスイッチを基板に取り付ける

※基板の穴に向かってスイッチを押し込むように挿入する。

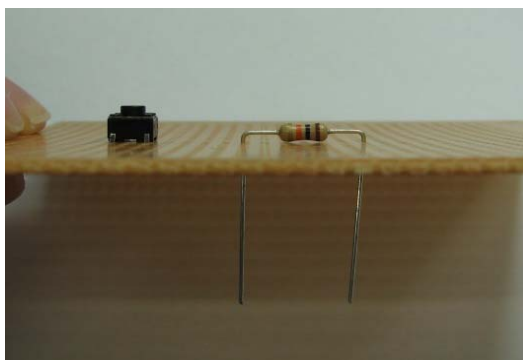


### ③抵抗のリード部分を『コの字型』に曲げる

※折り曲げた肩の部分が直角になっていることがこのましい。

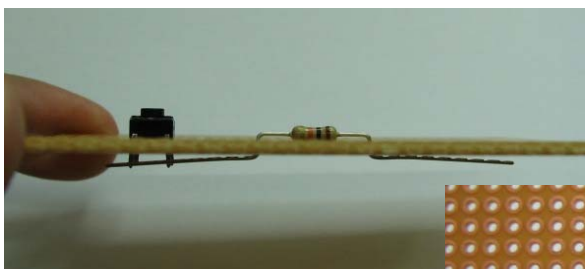


### ④抵抗のリード部分を基板の穴に入れる

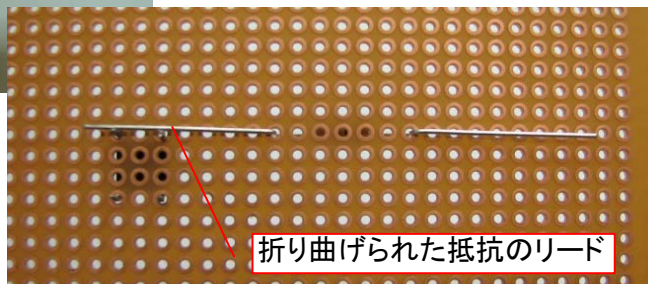


**⑤抵抗のリード部分を曲げる**

※外側に向かって曲げると抵抗の肩を崩さずに曲げられる。



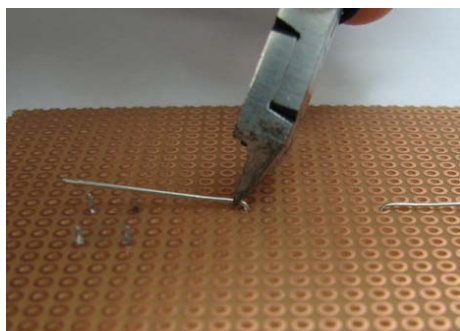
基板の上から見たところ



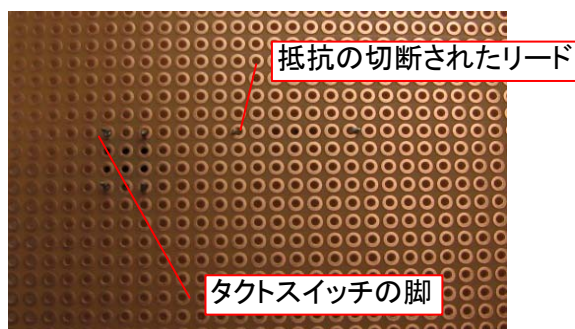
折り曲げられた抵抗のリード

**⑥抵抗のリード部分をニッパで切る**

※ニッパの平らな面を内側に向けて切ると切りやすい。



基板の上から見たところ

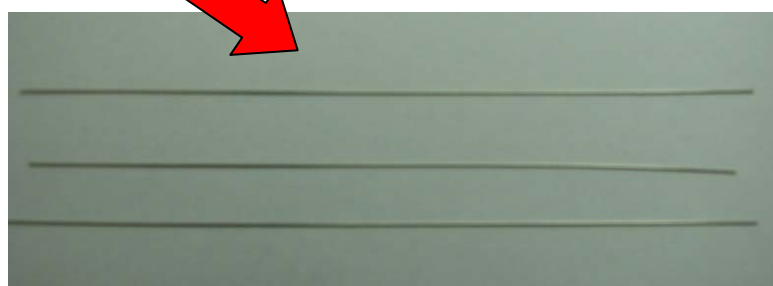
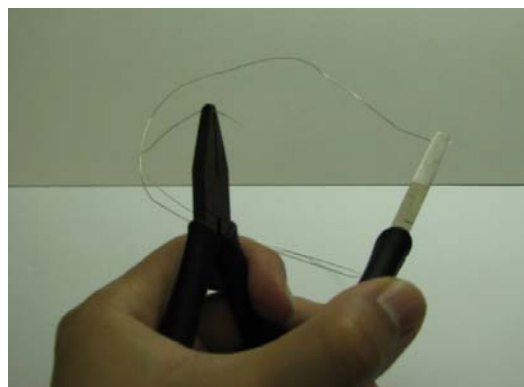
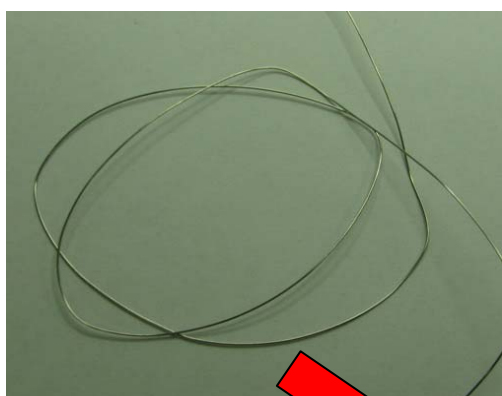


抵抗の切断されたリード

タクトスイッチの脚

**⑦鈴メッキ線をまっすぐに伸ばす**

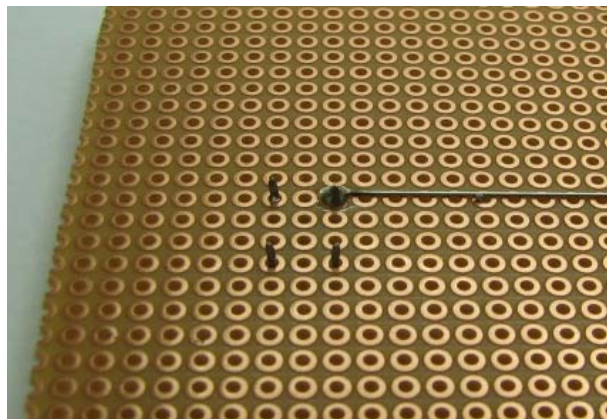
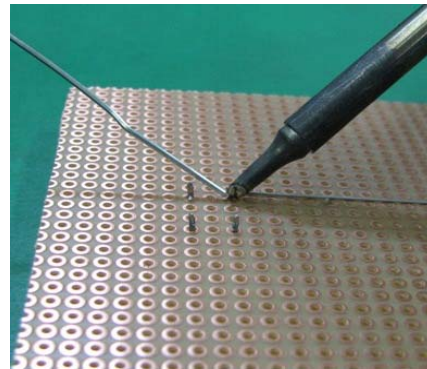
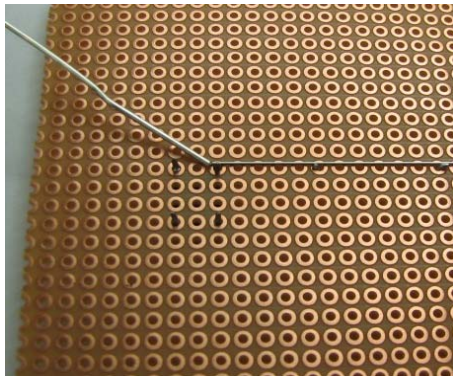
※鈴メッキ線の両端をペンチで強くつまみ、お互いを外側に向かって力を入れて広げる。





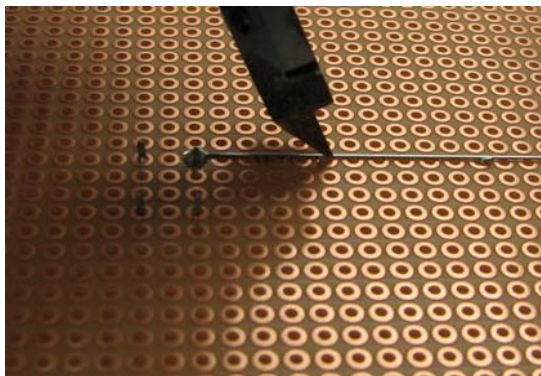
**⑧タクトスイッチの脚と鈴メッキ線をはんだづけする**

※タクトスイッチの脚に鈴メッキ線とはんだをあて、コテではんだづけする。



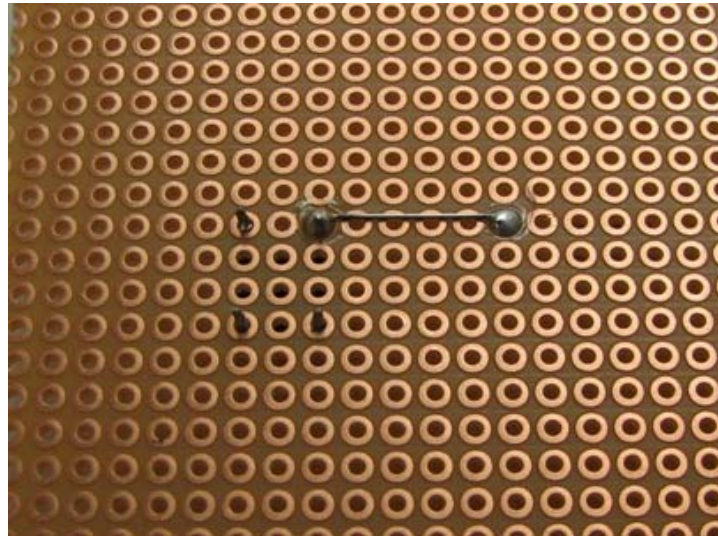
**⑨鈴メッキ線を抵抗のリード部分で切る**

※手順は⑥と同じ。

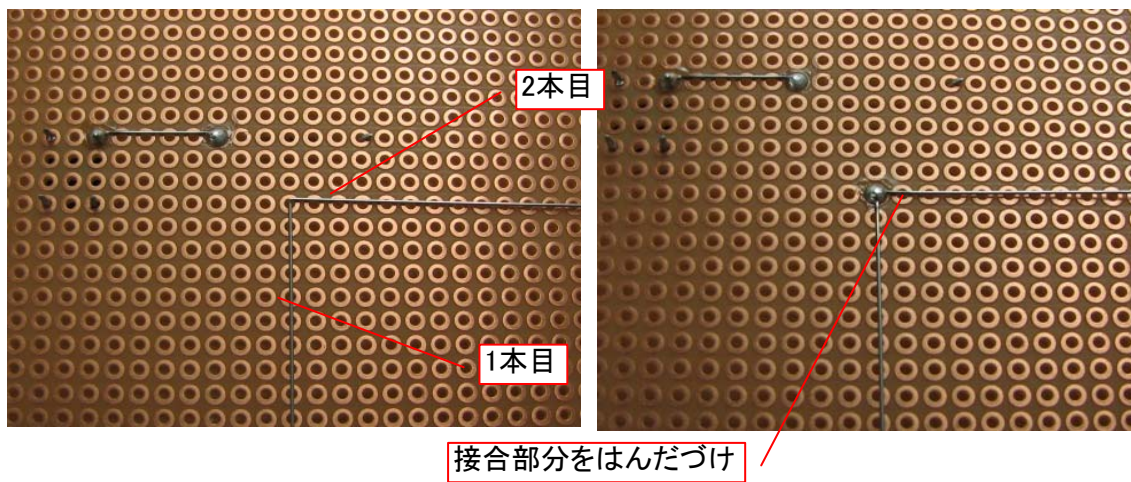




⑩ 鈴メッキ線と抵抗のリード部分をはんだづけする（完成）



その他： 配線を直角に折り曲げる場合は、鈴メッキ線を2本使用して、その接合部分をはんだづけする。



第9回高校生ものづくりコンテスト全国大会

# **電子回路組立部門 解説マニュアル**

第 1.01 版

2010.01.18

ルネサステクノロジ マイコンカーラリー事務局

# 注 意 事 項 (rev.2.0)

## 著作権

- ・本マニュアルに関する著作権はルネサステクノロジ マイコンカーラリー事務局に帰属します。
- ・本マニュアルは著作権法および、国際著作権条約により保護されています。

## 禁止事項

ユーザーは以下の内容を行うことはできません。

- ・第三者に対して、本マニュアルを販売、販売を目的とした宣伝、使用、営業、複製などを行うこと
- ・第三者に対して、本マニュアルの使用権を譲渡または再承諾すること
- ・本マニュアルの一部または全部を改変、除去すること
- ・本マニュアルを無許可で翻訳すること
- ・本マニュアルの内容を使用しての、人命や人体に危害を及ぼす恐れのある用途での使用

## 転載、複製

本マニュアルの転載、複製については、文書によるルネサステクノロジ マイコンカーラリー事務局の事前の承諾が必要です。

## 責任の制限

本マニュアルに記載した情報は、正確を期すため、慎重に制作したのですが万一本マニュアルの記述誤りに起因する損害が生じた場合でも、ルネサステクノロジ マイコンカーラリー事務局はその責任を負いません。

## その他

本マニュアルに記載の情報は本マニュアル発行時点のものであり、ルネサステクノロジ マイコンカーラリー事務局は、予告なしに、本マニュアルに記載した情報または仕様を変更することがあります。製作に当たりましては、最新の内容を確認いただきますようお願いいたします。

## 連絡先

ルネサステクノロジ マイコンカーラリー事務局  
〒162-0824 東京都新宿区揚場町 2-1 軽子坂MNビル  
TEL (03)-3266-8510  
E-mail:official@mcr.gr.jp

※記載されている会社名・製品名は、各社の商標または登録商標です。

# 目 次

1. 概要 .....	1
2. 構成 .....	2
2.1 全体構成 .....	2
2.2 制御用コンピュータ② .....	3
2.3 設計製作回路① .....	4
2.4 制御対象回路③ .....	6
2.5 接続 .....	8
3. プログラム .....	9
3.1 概要 .....	9
3.2 開発環境 .....	9
3.3 プログラム .....	9
3.4 注意事項 .....	13
3.5 課題1 .....	14
3.5.1 課題 .....	14
3.5.2 フローチャート .....	14
3.5.3 プログラム例 .....	15
3.5.4 プログラムの解説 .....	18
3.6 課題2 .....	19
3.6.1 課題 .....	19
3.6.2 フローチャート .....	20
3.6.3 プログラム例 .....	21
3.6.4 プログラムの解説 .....	21
3.7 課題3 .....	22
3.7.1 課題 .....	22
3.7.2 フローチャート .....	23
3.7.3 プログラム例 .....	24
3.7.4 プログラムの解説 .....	24
3.8 課題4 .....	25
3.8.1 課題 .....	25
3.8.2 フローチャート .....	26
3.8.3 プログラム例 .....	27
3.8.4 プログラムの解説 .....	27
3.9 課題5 .....	28
3.9.1 課題 .....	28
3.9.2 フローチャート .....	29
3.9.3 プログラム例 .....	30
3.9.4 プログラムの解説 .....	30
3.10 課題6 .....	31
3.10.1 課題 .....	31
3.10.2 フローチャート .....	32
3.10.3 プログラム例 .....	33
3.10.4 プログラムの解説 .....	34
3.11 課題7 .....	35

3.11.1 課題 .....	35
3.11.2 フローチャート.....	36
3.11.3 プログラム例.....	37
3.11.4 プログラムの解説.....	38

## 1. 概要

本マニュアルは、第9回高校生ものづくりコンテスト全国大会の電子回路組立部門の課題について解説したマニュアルです。電子回路組立部門には、回路製作(半田付け)と課題のプログラム製作がありますが、本マニュアルでは主に、プログラムについて説明します。

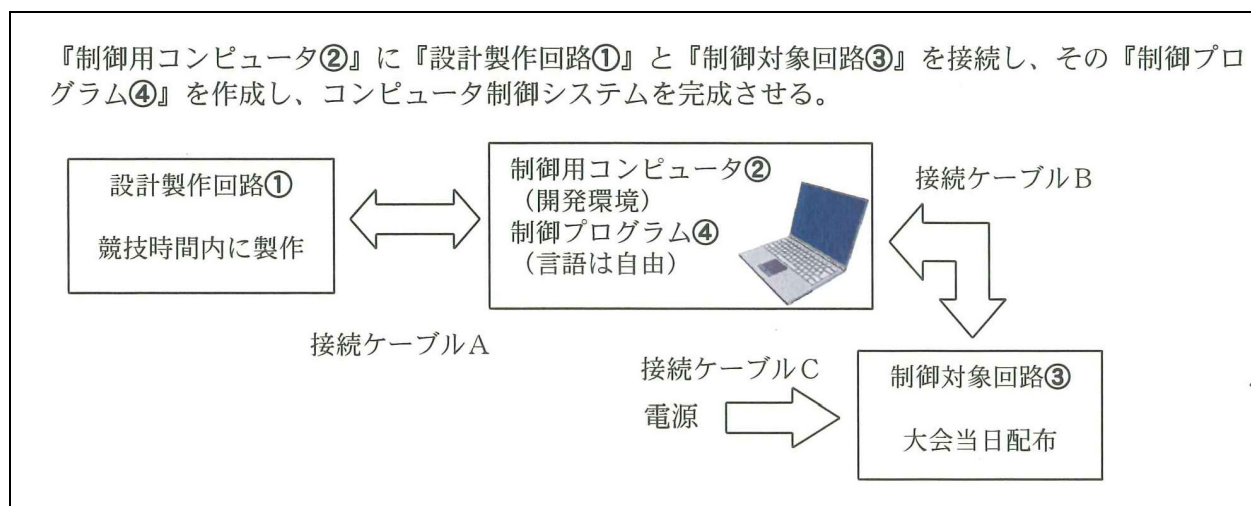
高校生ものづくりコンテストについては、全国工業高等学校長協会のホームページ(アドレス:  
<http://www.zenkoukyo.or.jp/>)を参照下さい。

なお、本マニュアル作成において北海道旭川工業高等学校の下村 幸広先生、西原 翔太君、神戸市立科学技術高等学校の登 弘聡先生、藤原 祐磨君よりご協力いただきました。この場を借りて、お礼申し上げます。

## 2. 構成

### 2.1 全体構成

構成は、下記のとおりです。



▲事前配付資料より抜粋

設計製作回路①	部品を支給され、競技時間内に製作する回路です。
制御用コンピュータ②	<p>制御用コンピュータは競技者が自由に構築することができます。ルネサステクノロジ製マイコン(H8 など)、マイクロチップ・テクノロジー製マイコン(PIC など)、ポケットコンピュータなどが多く使われています。ただ、ポケットコンピュータは、A/D 変換機能を使った課題が 2008 年度に出題されてからほとんど使われなくなりました。</p> <p>今回の制御用コンピュータ②は、下記のような構成とします。</p> <p>パソコン:WindowsXP、または WindowsVista、RS232C 搭載(USB 変換ケーブル可)  マイコンボード:マイコンカーラー承認ボード RY3048Fone ボード  開発環境:ルネサス統合開発環境  開発言語:C 言語</p> <p>マイコンボードは、ジャパンマイコンカーラーの承認ボードである「RY3048Fone」ボードを使用します。本ボードは、ルネサステクノロジ製の H8/3048F-ONE マイコンを搭載したボードです。</p>
制御対象回路③	大会当日に配布される基板です。
制御プログラム④	今回は、C 言語でプログラムを行います。

## 2.2 制御用コンピュータ②

本マニュアルでは、ジャパンマイコンカーラーの承認ボードである「RY3048Fone」ボードを使用します。本ボードは、ルネサステクノロジ製の H8/3048F-ONE マイコンを搭載したボードです。詳しい仕様やサンプルプログラムは、マイコンカーラーサイトにある「H8/3048F-ONE 実習マニュアル(ルネサス統合開発環境版)」を参照ください。

H8/3048F-ONE 実習マニュアル(ルネサス統合開発環境版)は、

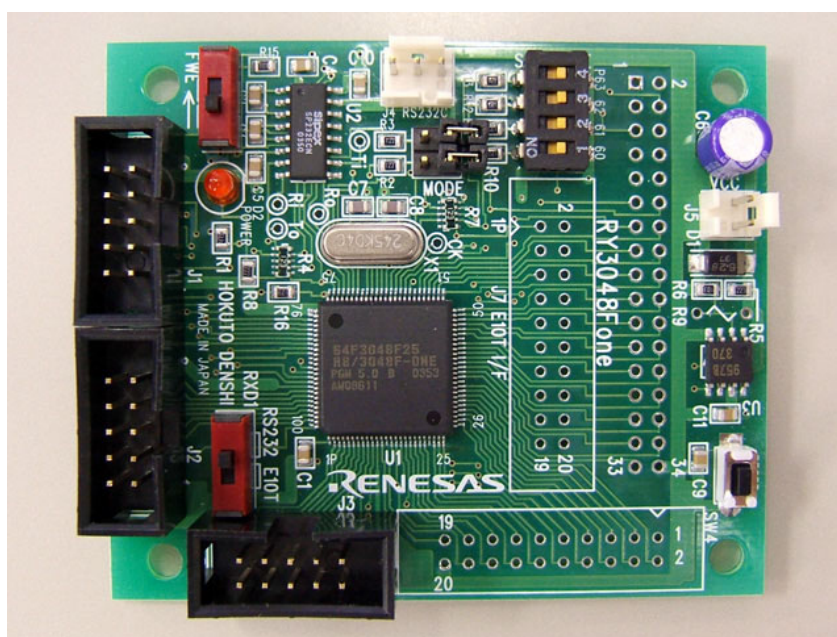
<http://www.mcr.gr.jp/tech/download/main01.html>



マイコンに関する資料

より、ダウンロードできます。

購入先:マイコンカーラー販売サイト URL:<http://www2.himdx.net/mcr/>



▲RY3048Fone ボード (マイコンは H8/3048F-ONE)

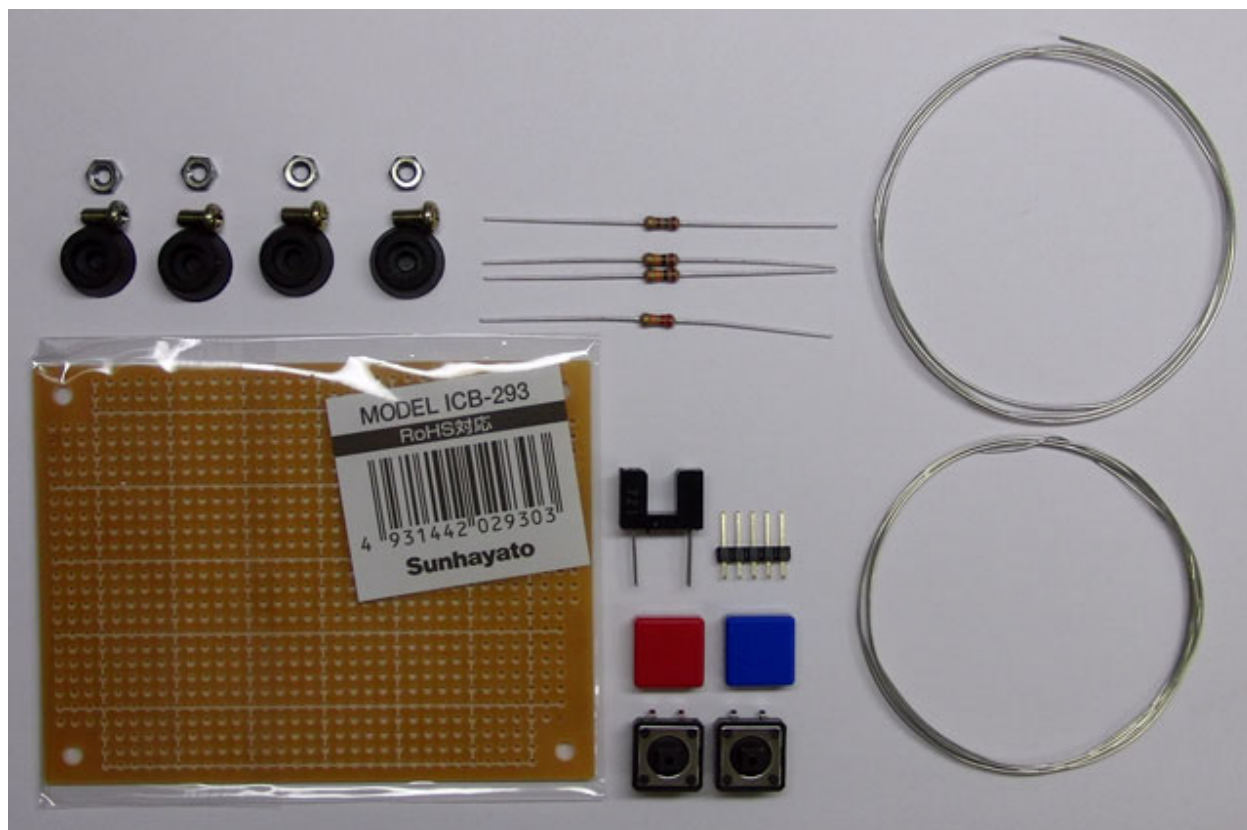


## 2.3 設計製作回路①

設計製作回路①は部品が支給され、競技時間内に競技者が製作します。支給部品を下記に示します。

番号	名 称	型式等	個数
1	ユニバーサル基板	ICB-293 (Sunhayato)	1
2	透過型フォトインタラプタ	SG206 (KODENSHI)	1
3	タクトスイッチ	赤 (秋月電子通商)	1
4	タクトスイッチ	青 (秋月電子通商)	1
5	ピンヘッダ	5 Pin ストレート	1
6	抵 抗	10k $\Omega$ 1/4W	2
7	抵 抗	180 $\Omega$ 1/4W	1
8	抵 抗	22k $\Omega$ 1/4W	1
9	ネ ジ	M3×10mm	4
10	ナット	M3	4
11	ゴム足	$\phi$ 14(相当品)	4
12	スズメッキ線	$\phi$ 0.5	1
13	鉛フリーハンダ	Sn-3Ag-0.5Cu 0.8mm	1
14	遮光チェック用アルミ板	20mm×110mm	1

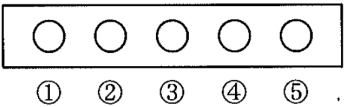
▲大会当日配付資料より抜粋



▲大会当日支給された部品

5ピンのヘッダピンの接続は、下記のように指定されています。

ICピッチ1列5ピン



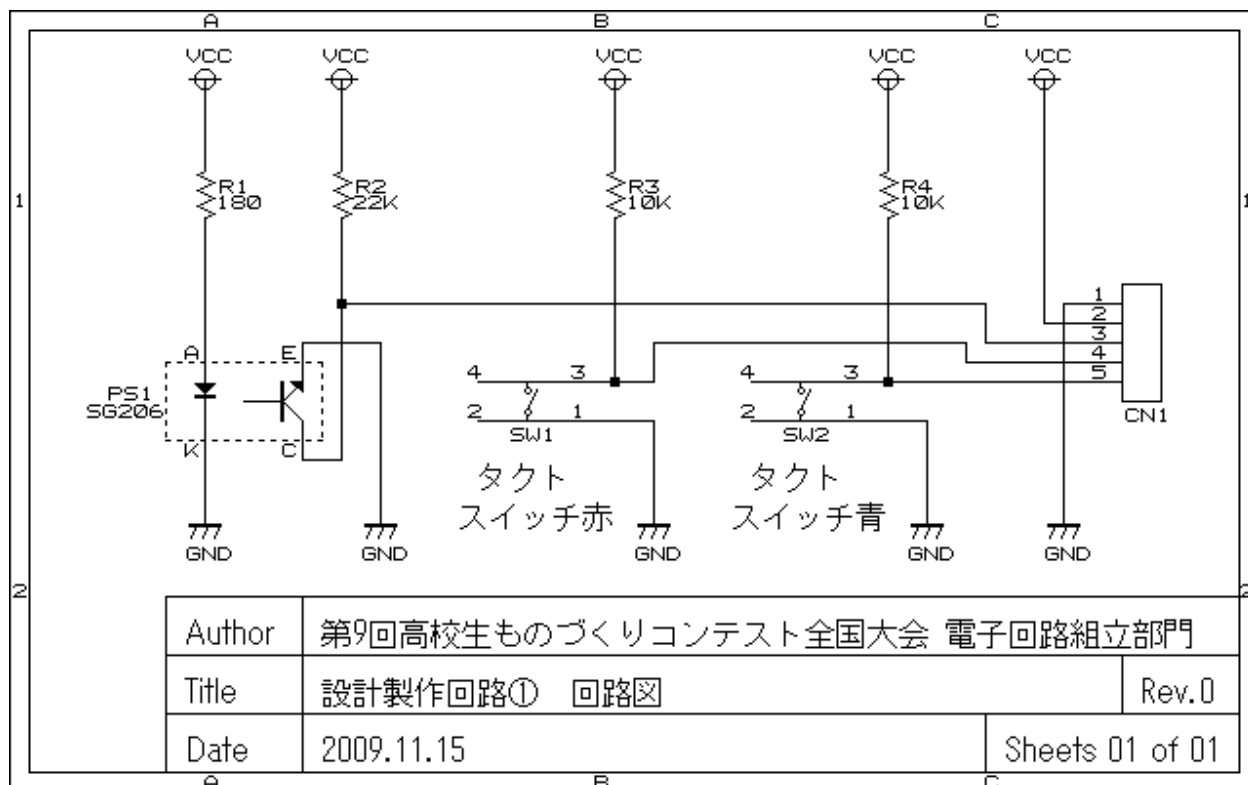
①	GND	②	5V	③	PS
④	TSR	⑤	TSB		

PS : 透過型フォトインタラプタ  
 TSR : タクトスイッチ赤  
 TRB : タクトスイッチ青

▲大会当日配付資料より抜粋

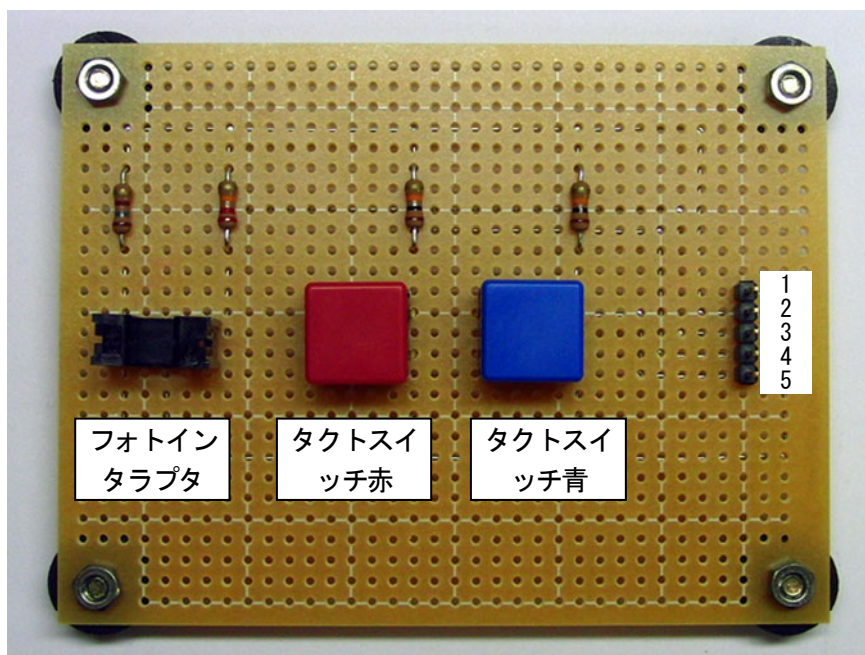
回路図はありませんので、競技者は部品から回路を予想して組み立てます。

下記に、回路製作例を示します。



▲回路製作例

下記に、完成した基板の製作例を示します。

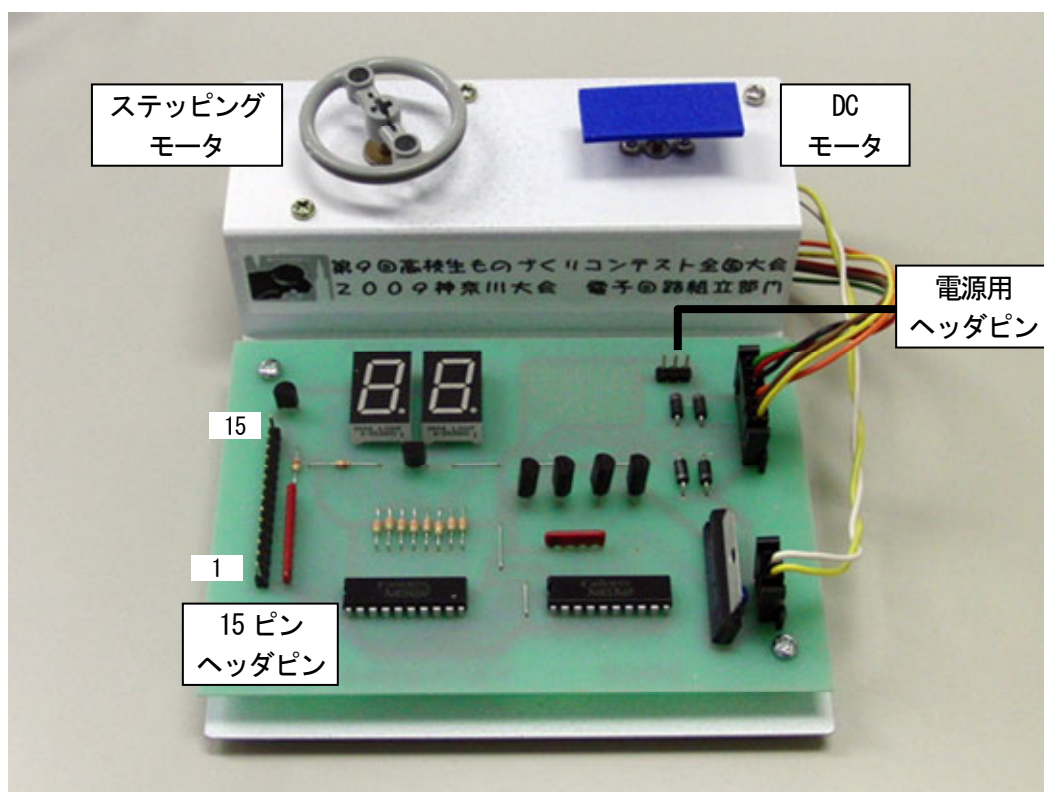


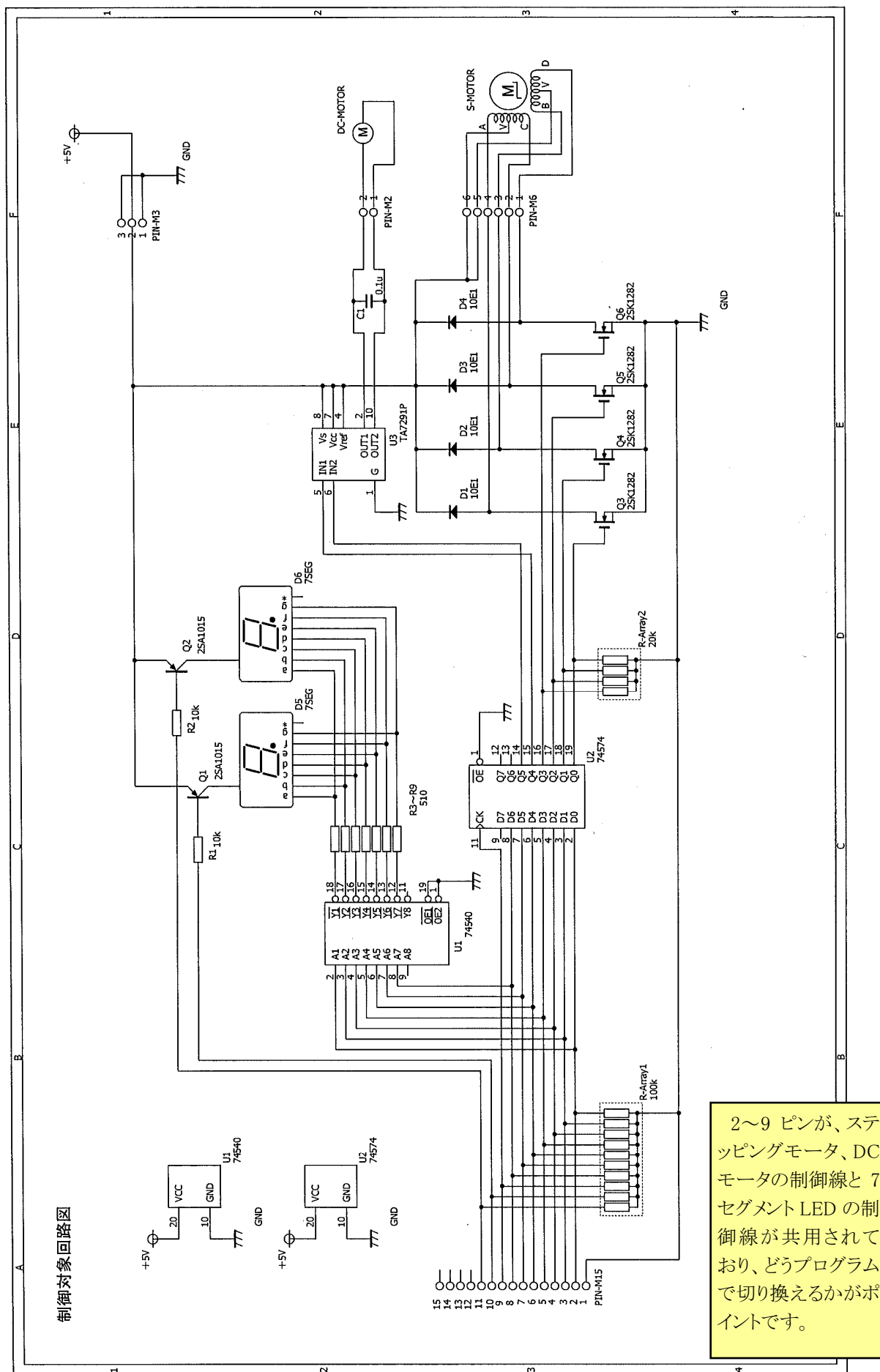
▲基板製作例

## 2.4 制御対象回路③

制御対象回路③は、大会当日に配布される基板です。

下記に制御対象回路③の写真、回路図を示します。

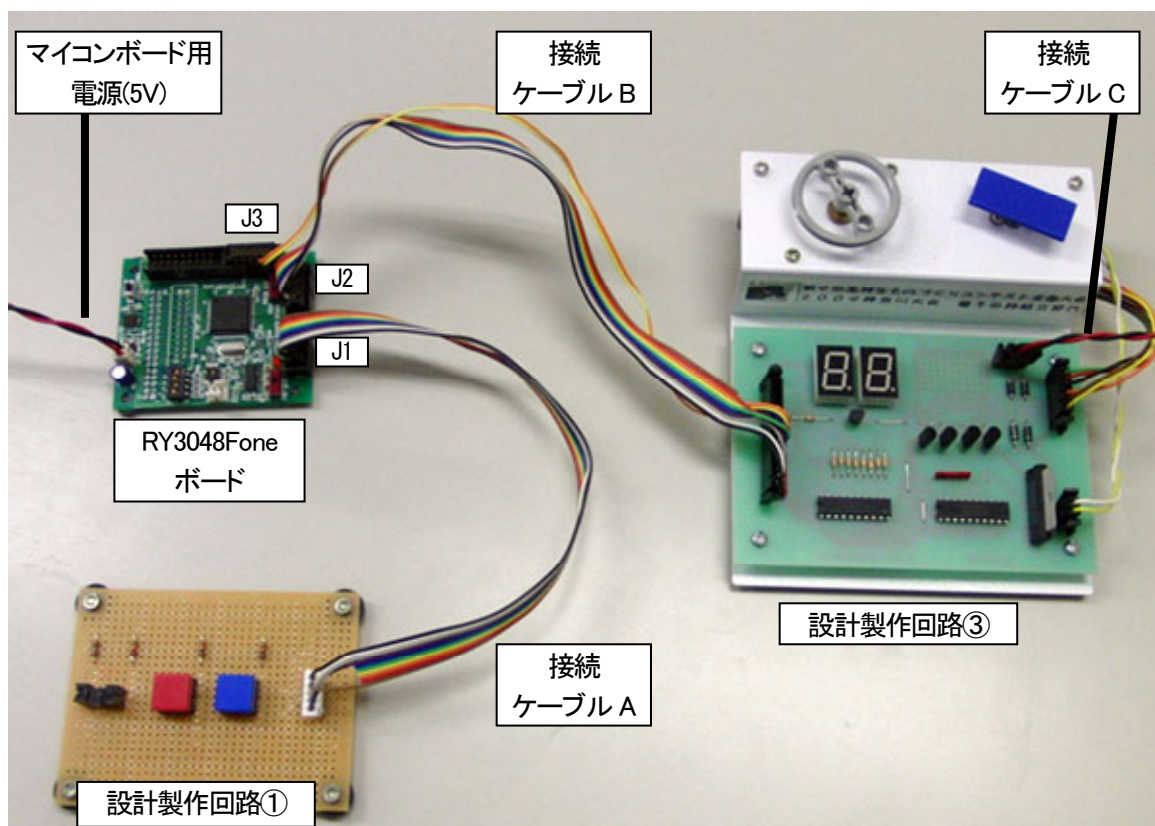
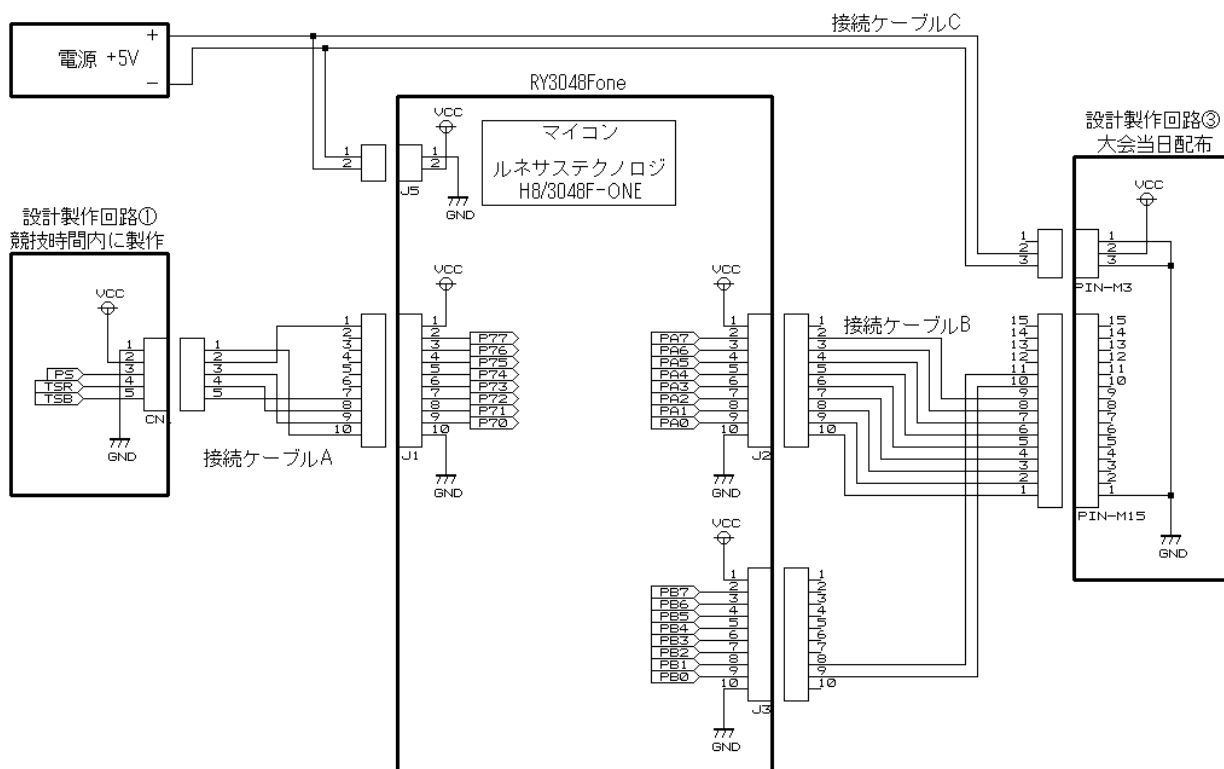




2～9 ピンが、ステップモータ、DCモータの制御線と7セグメントLEDの制御線が共用されており、どうプログラムで切り換えるかがポイントです。

## 2.5 接続

設計製作回路①(競技時間内に製作)、設計製作回路③(大会当日配布)、RY3048Fone ボードとの接続を下記に示します。接続ケーブル A、接続ケーブル B は、事前に製作しておきます。今回のプログラムでは、下記のように結線されているものとします。





## 3. プログラム

### 3.1 概要

課題のプログラムは7問あります。基板製作、回路図記載、プログラム含め制限時間である2時間30分で終わらせなければいけません。そのため、今回は初期設定や考え方が複雑になる割り込みを使っておりません。タイマは、for文によるソフトウェアタイマを使っています。

### 3.2 開発環境

本マニュアルでは、ルネサス統合開発環境(無償評価版)を使用します。

ルネサス統合開発環境やその他ファイルの入手、インストール、操作方法については、マイコンカーラーサイトにある「ルネサス統合開発環境 操作マニュアル導入編」を参照ください。

ルネサス統合開発環境 操作マニュアル導入編は、

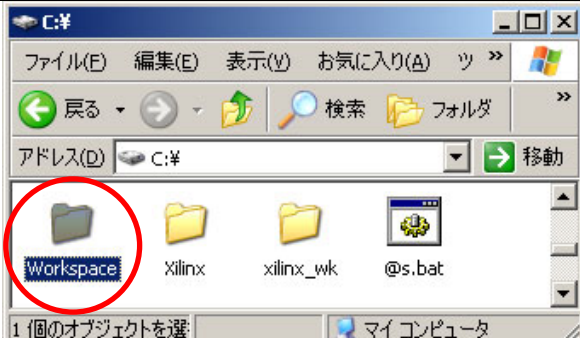
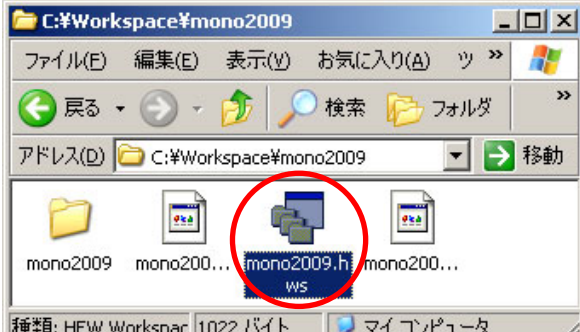
<http://www.mcr.gr.jp/tech/download/main01.html>

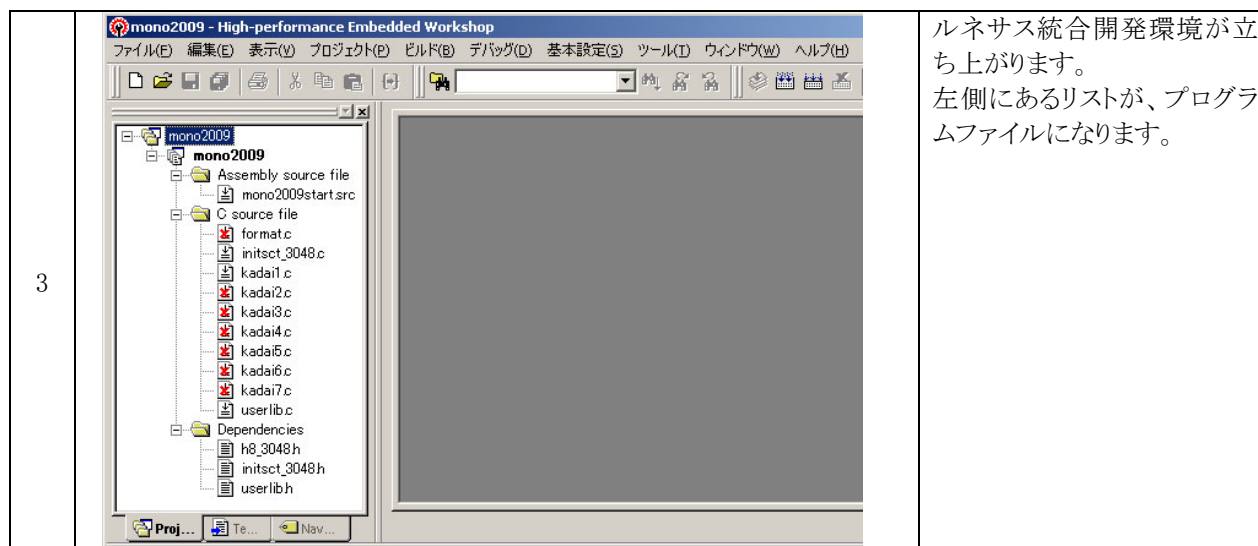


開発環境、サンプルプログラムの資料より、ダウンロードできます。

### 3.3 プログラム

ルネサス統合開発環境で開くワークスペースなどが、「workspace」フォルダ内にあります。このフォルダ内にプログラムがあります。このフォルダの扱いを説明します。

1		<p>付属している「workspace」フォルダを、Cドライブにコピー、または移動します。</p>
2		<p>「C ドライブ→workspace→mono2009」フォルダにある、「mono2009.hws」を実行します。</p>



プログラムファイルと内容を、下記に示します。

ファイル名	詳細
h8_3048.h	H8/3048F-ONE マイコンのレジスタの定義をしているファイルです。全課題共通で使用します。 ファイルの場所: C:\¥Workspace¥common¥h8_3048.h
mono2009start.src	マイコン起動時のプログラム(スタートアップルーチン)、ベクタアドレス(割り込み発生時の実行先の定義、今回は割り込みは使用していません)が定義されているファイルです。全課題共通で使用します。 ファイルの場所: C:\¥Workspace¥mono2009¥mono2009¥mono2009start.src
initsct_3048.c initsct_3048.h	H8/3048F-ONE マイコンを使う上で、メモリの初期化を行うルーチンが入っているファイルです。全課題共通で使用します。 ファイルの場所: C:\¥Workspace¥common¥initsct_3048.c
userlib.c userlib.h	課題プログラムを作成する上で、ポートの定義など、マイコンに特化したプログラムを入れているファイルです。全課題共通で使用します。 ファイルの場所: C:\¥Workspace¥mono2009¥mono2009¥userlib.c
format.c	課題のフォーマットです。いちばん最初、この状態から始めます(空のファイルです)。 ファイルの場所: C:\¥Workspace¥mono2009¥mono2009¥format.c
kadai1.c	課題1の回答例が入っているファイルです。 ファイルの場所: C:\¥Workspace¥mono2009¥mono2009¥kadai1.c
kadai2.c	課題2の回答例が入っているファイルです。 ファイルの場所: C:\¥Workspace¥mono2009¥mono2009¥kadai2.c
kadai3.c	課題3の回答例が入っているファイルです。 ファイルの場所: C:\¥Workspace¥mono2009¥mono2009¥kadai3.c
kadai4.c	課題4の回答例が入っているファイルです。 ファイルの場所: C:\¥Workspace¥mono2009¥mono2009¥kadai4.c
kadai5.c	課題5の回答例が入っているファイルです。 ファイルの場所: C:\¥Workspace¥mono2009¥mono2009¥kadai5.c
kadai6.c	課題6の回答例が入っているファイルです。 ファイルの場所: C:\¥Workspace¥mono2009¥mono2009¥kadai6.c
kadai7.c	課題7の回答例が入っているファイルです。 ファイルの場所: C:\¥Workspace¥mono2009¥mono2009¥kadai7.c

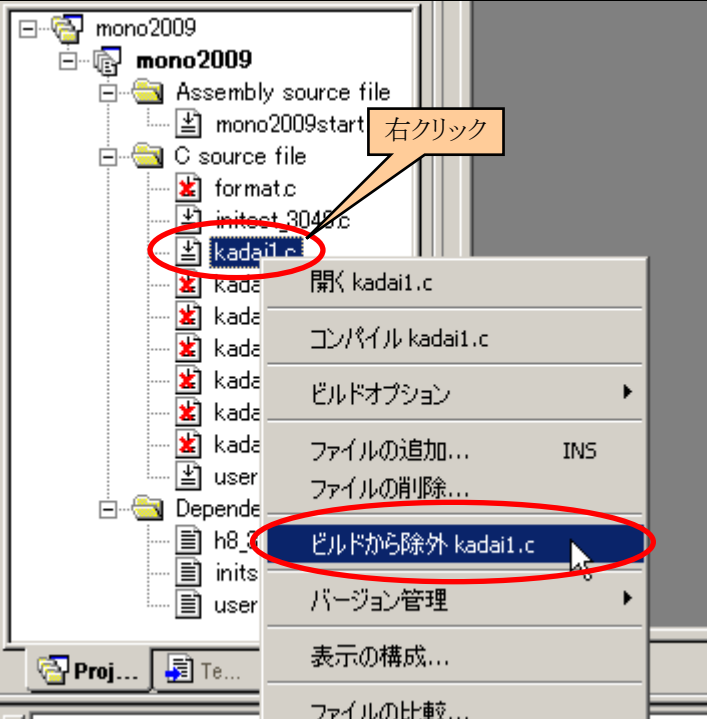
本プロジェクトには、課題1～7、フォーマットのファイルをすべて登録していますが、この中で有効にできるのは1つだけです。例えば、課題1のときは、「kadai1.c」のみ有効、「kadai2.c～kadai7.c、format.c」はビルドから除外（ファイル左の赤い×マーク）にしておきます。

各課題のプログラムをビルドするときに、ビルドから除外するファイルを下記に示します。

ファイル名	課題1	課題2	課題3	課題4	課題5	課題6	課題7
mono2009start.src	○	○	○	○	○	○	○
initsct_3048.c	○	○	○	○	○	○	○
userlib.c	○	○	○	○	○	○	○
format.c	×	×	×	×	×	×	×
kadai1.c	○	×	×	×	×	×	×
kadai2.c	×	○	×	×	×	×	×
kadai3.c	×	×	○	×	×	×	×
kadai4.c	×	×	×	○	×	×	×
kadai5.c	×	×	×	×	○	×	×
kadai6.c	×	×	×	×	×	○	×
kadai7.c	×	×	×	×	×	×	○

○:有効 ×:ビルドから除外するファイル

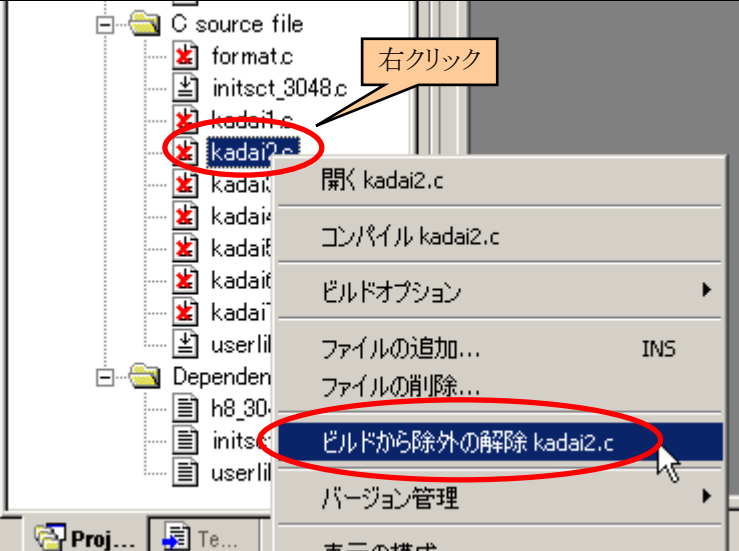
4



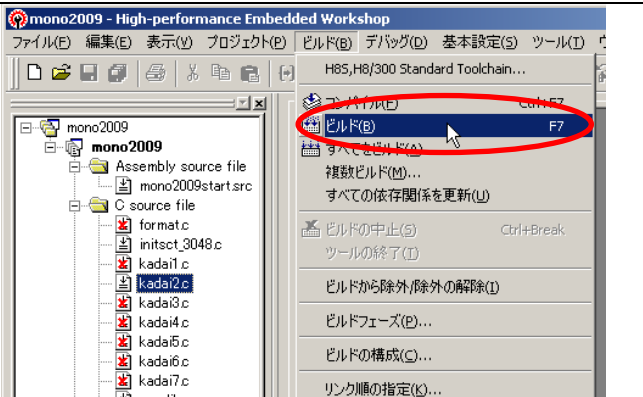
例えば、課題2のファイルである、「kadai2.c」をビルド(MOTファイルの作成)したい場合、次の操作を行います。

「kadai1.c」の上で右クリックし、「ビルドから除外」をクリックします。



5		<p>「kadai2.c」の上で右クリックし、「ビルドから除外の解除」をクリックします。</p>
---	--	--

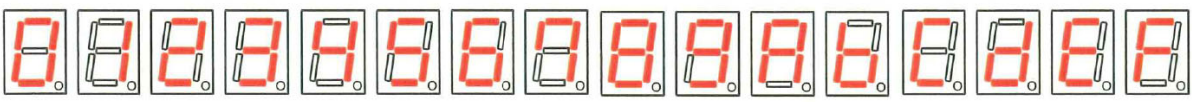
6		<p>リストが、左画面のようになれば完了です。</p>
---	--	-----------------------------

7		<p>「ビルド→ビルド」で、kadai2.c などの登録されているファイルがビルド(アセンブル、コンパイル、リンク)され最終ファイル(MOT ファイル)ができあがります。</p>
---	---	---

8		<p>MOT ファイルは、 「C:\Workspace\mono2009\mono2009\Debug」フォルダ内にできます。</p>
---	---	---

### 3.4 注意事項

課題 1～7 の注意事項を下記に示します。

<p>① 文中の表現で「ON」はタクトスイッチ（TSB または TSR）を押している状態、「OFF」はタクトスイッチが押されていない状態をいう。 「ON/OFF」はタクトスイッチを押して離すことをいう。</p> <p>② 7セグメント LED の表示は次のように点灯すること。（図の赤が点灯状態） また 16 進数の B と D についても下のように小文字で点灯させること。</p> <div style="text-align: center;">  </div> <p>③ 課題の中にある動作概要図については大まかな流れが図示してあるので、詳細な条件については課題文章のとおり動作すること。</p>
---

▲大会当日配付資料より抜粋

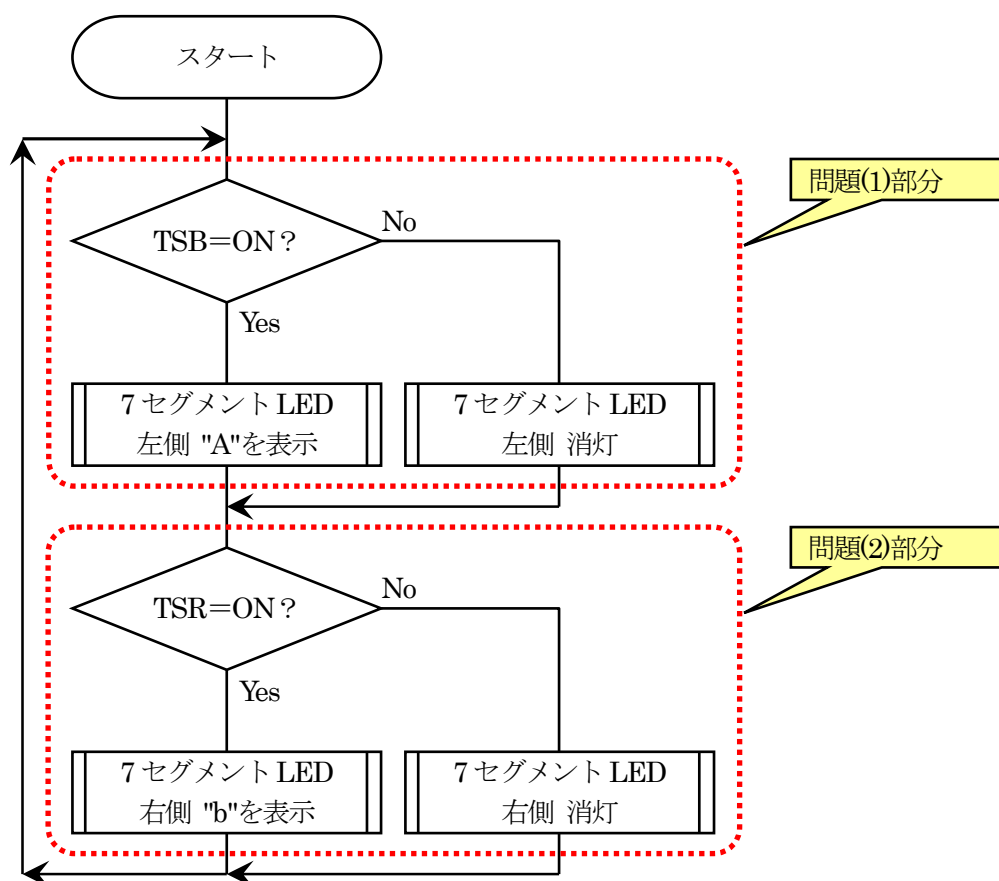
### 3.5 課題1

#### 3.5.1 課題

- (1) 左側の7セグメントLEDは、タクトスイッチ TSB が ON のとき'A'を表示し OFF のときは消灯する。
- (2) 右側の7セグメントLEDは、タクトスイッチ TSR が ON のとき'b'を表示し OFF のときは消灯する。
- (3) 2つのモータはいずれも停止していること。
- (4) フォトインタラプタの状態に関係なく動作すること。

▲大会当日配付資料より抜粋

#### 3.5.2 フローチャート



## 3.5.3 プログラム例

```

1 : //////////////////////////////////////
2 : // 第9回高校生ものづくりコンテスト全国大会 電子回路部門 課題1
3 : // Copyright (C) 2009 ルネサステクノロジ マイコンカラー事務局
4 : //////////////////////////////////////
5 :
6 : #include <machine.h>                // H8 マイコン特有の命令取り込み
7 : #include "h8_3048.h"                // H8/3048F-ONE 用 I/O レジスタ定義
8 : #include "userlib.h"                // ユーザライブラリ
9 : #include "initsect_3048.h"          // セクション(RAM)の初期化
10 :
11 : // タクトスイッチの状態
12 : #define ON      1
13 : #define OFF     0
14 :
15 : // フォトインタラプタの状態
16 : #define SYADAN  1
17 : #define TOUKA   0
18 :
19 : // 7セグメントLEDの桁
20 : #define LEFT    1
21 : #define RIGHT   0
22 :
23 : // 7セグメントLEDの表示データ
24 : unsigned char seg_data[] =
25 :     { 0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x27,
26 :       0x7f, 0x6f, 0x77, 0x7c, 0x39, 0x5e, 0x79, 0x71 };
27 :
28 : // ステッピングモータの出力信号
29 : unsigned char stepper_data[] =
30 :     { 0x01, 0x02, 0x04, 0x08 };
31 :
32 : // DCモータ、ステッピングモータ出力信号
33 : unsigned char motor_out;
34 :
35 : // ステッピングモータに加えたパルス数
36 : long      s_pulse;
37 :
38 : //////////////////////////////////////
39 : // ウェイト
40 : // 引数:ウェイト数
41 : //////////////////////////////////////
42 : void wait( int i )
43 : {
44 :     int j;
45 :
46 :     while( i-- ) {
47 :         for( j=0; j<50; j++ );
48 :     }
49 : }
50 :
51 : //////////////////////////////////////
52 : // 透過型フォトインタラプタの状態取得
53 : // 0:障害物無し 1:障害物あり
54 : //////////////////////////////////////
55 : int getPS( void )
56 : {
57 :     int ret;
58 :
59 :     if( P7DR & 0x01 ) {                // 透過型フォトインタラプタの状態確認
60 :         ret = 1;
61 :     } else {
62 :         ret = 0;
63 :     }
64 :     return ret;
65 : }
66 :

```

```

67 : //////////////////////////////////////
68 : // タクトスイッチ赤(TSR)の状態取得
69 : // 0:OFF 1:ON
70 : //////////////////////////////////////
71 : int getTSR( void )
72 : {
73 :     int ret;
74 :
75 :     if( P7DR & 0x02 ) {                // TSR の状態確認
76 :         ret = 0;
77 :     } else {
78 :         ret = 1;
79 :     }
80 :     return ret;
81 : }
82 :
83 : //////////////////////////////////////
84 : // タクトスイッチ青(TSB)の状態取得
85 : // 0:OFF 1:ON
86 : //////////////////////////////////////
87 : int getTSB( void )
88 : {
89 :     int ret;
90 :
91 :     if( P7DR & 0x04 ) {                // TSB の状態確認
92 :         ret = 0;
93 :     } else {
94 :         ret = 1;
95 :     }
96 :     return ret;
97 : }
98 :
99 : //////////////////////////////////////
100 : // 7セグメントLED点灯処理
101 : // 桁(1:10の位 0:1の位), 表示データ 0~f(0以下で消灯)
102 : //////////////////////////////////////
103 : void set7seg( int keta, int i )
104 : {
105 :     PBDR = 0x03;                      // 全消灯
106 :
107 :     if( i >= 0 ) {
108 :         PADR = seg_data[ i ];          // 値のセット
109 :     } else {
110 :         PADR = 0x00;                  // 値がマイナスなら消灯
111 :     }
112 :
113 :     if( keta == 1 ) {
114 :         PBDR = 0x02;                  // 10の桁点灯
115 :     } else {
116 :         PBDR = 0x01;                  // 1の桁点灯
117 :     }
118 :     wait( 10 );
119 : }
120 :
121 : //////////////////////////////////////
122 : // 直流モータの制御
123 : // 1:時計回り -1:反時計回り それ以外:停止
124 : //////////////////////////////////////
125 : void dc_motor( int i )
126 : {
127 :     if( i == 1 ) {                    // 時計回りか?
128 :         motor_out &= 0xef;
129 :         motor_out |= 0x20;
130 :     } else if( i == -1 ) {            // 反時計回りか?
131 :         motor_out &= 0xdf;
132 :         motor_out |= 0x10;
133 :     } else {                          // それ以外は停止
134 :         motor_out &= 0xcf;
135 :     }
136 : }
137 :

```

```

138 : //////////////////////////////////////
139 : // ステッピングモータの制御
140 : // 回転速度 0:停止、値が小さいと速い、値が大きいと遅い)
141 : // 方向      1:時計回り -1:反時計回り 0:停止
142 : //////////////////////////////////////
143 : void stepper_motor( int w, int i )
144 : {
145 :     static int s_wait;
146 :     static int s_count;
147 :
148 :     if( w == 0 ) return;          // 回転速度が0なら停止
149 :
150 :     s_wait++;
151 :     if( s_wait <= w ) return;    // 回転速度により処理しない
152 :
153 :     s_wait = 0;
154 :
155 :     motor_out &= 0xf0;
156 :
157 :     if( i != 0 ) {
158 :         motor_out |= stepper_data[ s_count ]; // ステッピングモータ回転データセット
159 :     }
160 :
161 :     if( i == 1 ) {                // 時計回りか?
162 :         s_pulse++;
163 :         s_count++;
164 :         if( s_count > 3 ) s_count = 0;
165 :     } else if( i == -1 ) {       // 反時計回りか?
166 :         s_pulse--;
167 :         s_count--;
168 :         if( s_count < 0 ) s_count = 3;
169 :     }
170 : }
171 :
172 : //////////////////////////////////////
173 : // DC モータ、ステッピングモータへ信号出力
174 : //////////////////////////////////////
175 : void motor( void )
176 : {
177 :     unsigned char i, j;
178 :
179 :     i = PBDR;          // 現在の表示桁保存
180 :     j = PADR;          // 現在の表示値保存
181 :
182 :     PBDR = 0x03;       // 7セグメントLED 消灯
183 :
184 :     PADR = motor_out;  // DC モータ、ステッピングモータへ信号出力
185 :
186 :     PADR |= 0x80;       // CK HIGH
187 :     PADR &= 0x7f;       // CK LOW
188 :
189 :     PBDR = i;          // 表示桁戻す
190 :     PADR = j;          // 表示値戻す
191 : }
192 :

```

```

193 : //////////////////////////////////////
194 : // メイン関数
195 : //////////////////////////////////////
196 : void main( void )
197 : {
198 :     init();
199 :
200 :     while( 1 ) {
201 :         if( getTSB() == ON ) {
202 :             set7seg( LEFT , 0xa );    // "A"表示
203 :         } else {
204 :             set7seg( LEFT , -1 );    // 消灯
205 :         }
206 :
207 :         if( getTSR() == ON ) {
208 :             set7seg( RIGHT, 0xb );    // "B"表示
209 :         } else {
210 :             set7seg( RIGHT, -1 );    // 消灯
211 :         }
212 :     }
213 : }
214 :
215 : //////////////////////////////////////
216 : // End of File
217 : //////////////////////////////////////

```

### 3.5.4 プログラムの解説

行	詳細
201	getTSB 関数は、タクトスイッチブルー(青)の現在の状態を取得する関数です。ON なら 202 行を実行、OFF なら 204 行を実行します。
202	7 セグメント LED に値を表示します。 set7seg 関数の使い方を次に示します。 set7seg( 点灯位置, 値 ) 点灯位置: 左側を点灯するなら「LEFT」、右側を点灯するなら「RIGHT」を設定します。 値: 0～15 を設定します。消灯する場合は、-1 を設定します。 例) 左側に「5」を点灯 set7seg( LEFT, 5 ); 右側を消灯 set7seg( RIGHT, -1 );
207	getTSR 関数は、タクトスイッチレッド(赤)の現在の状態を取得する関数です。ON なら 208 行を実行、OFF なら 210 行を実行します。

## 3.6 課題2

### 3.6.1 課題

- (1) 左側7セグメントLEDがフォトインタラプタとタクトスイッチの状態によって次のように表示されること。

フォトインタラプタ	タクトスイッチ TSB	タクトスイッチ TSR	左側LED
透過	OFF	OFF	0
透過	OFF	ON	1
透過	ON	OFF	2
透過	ON	ON	3
遮断	OFF	OFF	4
遮断	OFF	ON	5
遮断	ON	OFF	6
遮断	ON	ON	7

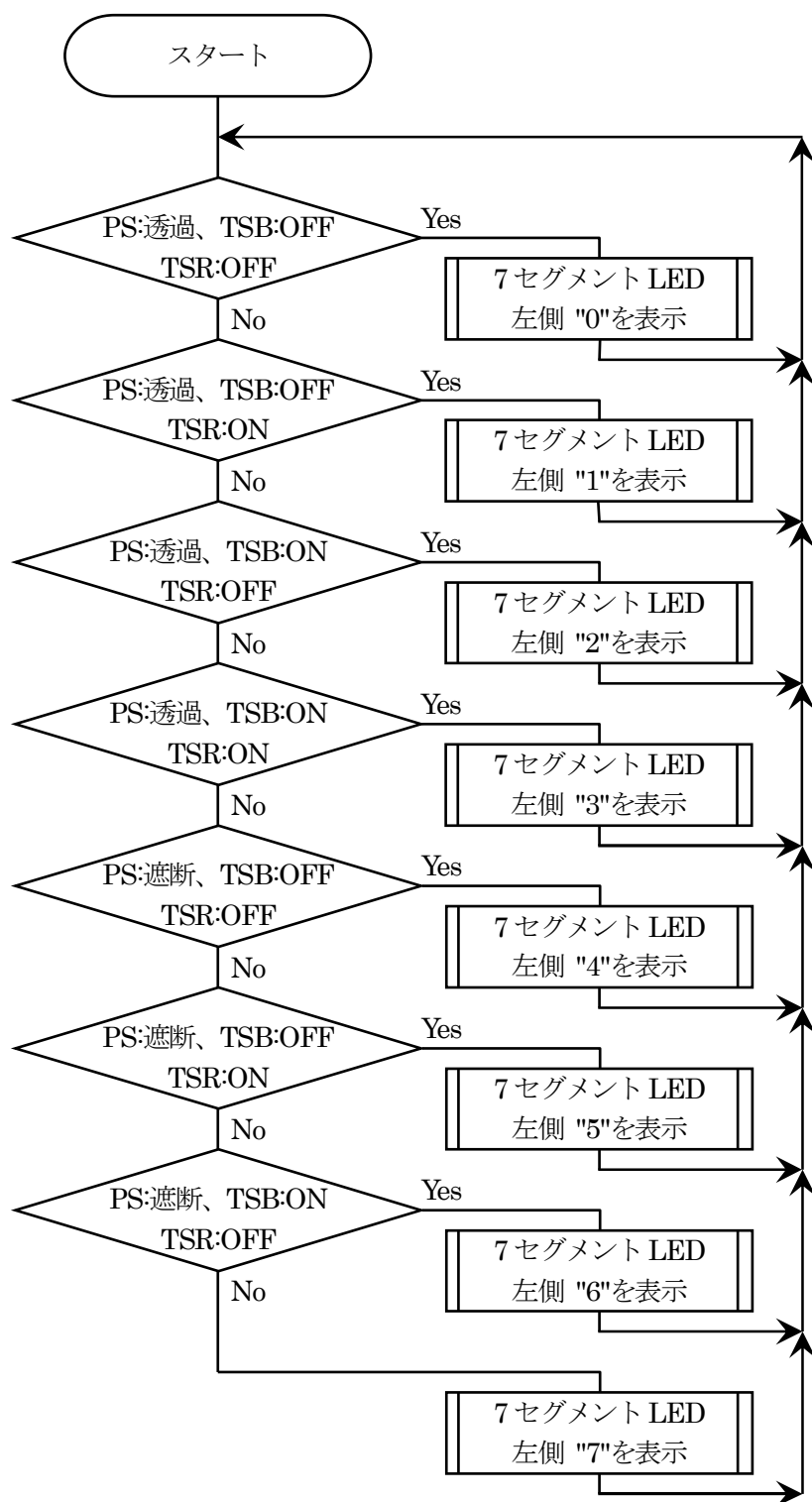
- (2) 2つのモータはいずれも停止していること。

- (3) 右側7セグメントLEDは常に消灯していること。

▲大会当日配付資料より抜粋



### 3.6.2 フローチャート



※判断式は、すべてAND条件(「PS:〇〇」かつ「TSB:〇〇」かつ「TSR:〇〇」)

## 3.6.3 プログラム例

```

1 : //////////////////////////////////////
2 : // 第9回高校生ものづくりコンテスト全国大会 電子回路部門 課題2
3 : // Copyright (C) 2009 ルネサステクノロジ マイコンカーラリー事務局
4 : //////////////////////////////////////

```

5～192行は、課題1と同じため省略

```

193 : //////////////////////////////////////
194 : // メイン関数
195 : //////////////////////////////////////
196 : void main( void )
197 : {
198 :     init();
199 :
200 :     set7seg( LEFT , -1 );
201 :     set7seg( RIGHT , -1 );
202 :
203 :     while( 1 ) {
204 :         if( getPS() == TOUKA && getTSB() == OFF && getTSR() == OFF ) {
205 :             set7seg( LEFT , 0 );
206 :             set7seg( RIGHT , -1 );
207 :         } else if( getPS() == TOUKA && getTSB() == OFF && getTSR() == ON ) {
208 :             set7seg( LEFT , 1 );
209 :             set7seg( RIGHT , -1 );
210 :         } else if( getPS() == TOUKA && getTSB() == ON && getTSR() == OFF ) {
211 :             set7seg( LEFT , 2 );
212 :             set7seg( RIGHT , -1 );
213 :         } else if( getPS() == TOUKA && getTSB() == ON && getTSR() == ON ) {
214 :             set7seg( LEFT , 3 );
215 :             set7seg( RIGHT , -1 );
216 :         } else if( getPS() == SYADAN && getTSB() == OFF && getTSR() == OFF ) {
217 :             set7seg( LEFT , 4 );
218 :             set7seg( RIGHT , -1 );
219 :         } else if( getPS() == SYADAN && getTSB() == OFF && getTSR() == ON ) {
220 :             set7seg( LEFT , 5 );
221 :             set7seg( RIGHT , -1 );
222 :         } else if( getPS() == SYADAN && getTSB() == ON && getTSR() == OFF ) {
223 :             set7seg( LEFT , 6 );
224 :             set7seg( RIGHT , -1 );
225 :         } else {
226 :             set7seg( LEFT , 7 );
227 :             set7seg( RIGHT , -1 );
228 :         }
229 :     }
230 : }
231 :
232 : //////////////////////////////////////
233 : // End of File
234 : //////////////////////////////////////

```

## 3.6.4 プログラムの解説

if 文を使って、フォトインタラプタ、タクトスイッチブルー、タクトスイッチレッドの状態を判断して、条件が成り立ったなら左側の7セグメントLEDへ0～7の値を出力します。

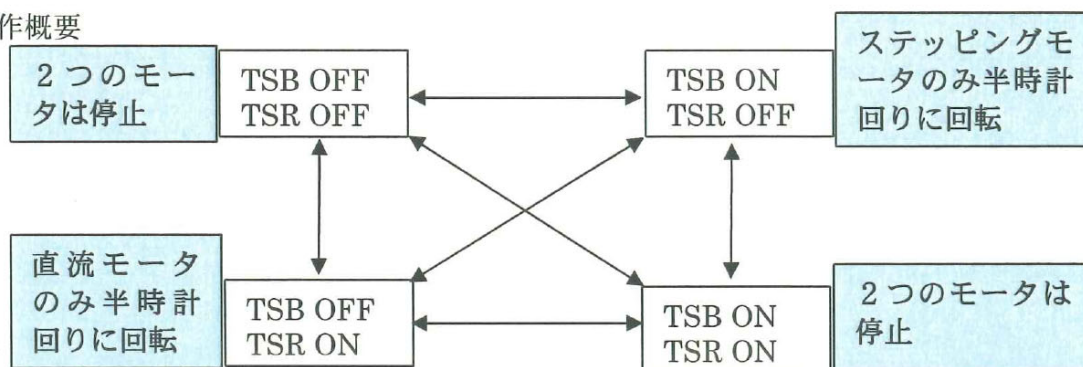
switch～case 文を使った方がシンプルにプログラムできます。switch～case 文を使ったプログラムにもチャレンジしてみてください。

### 3.7 課題3

#### 3.7.1 課題

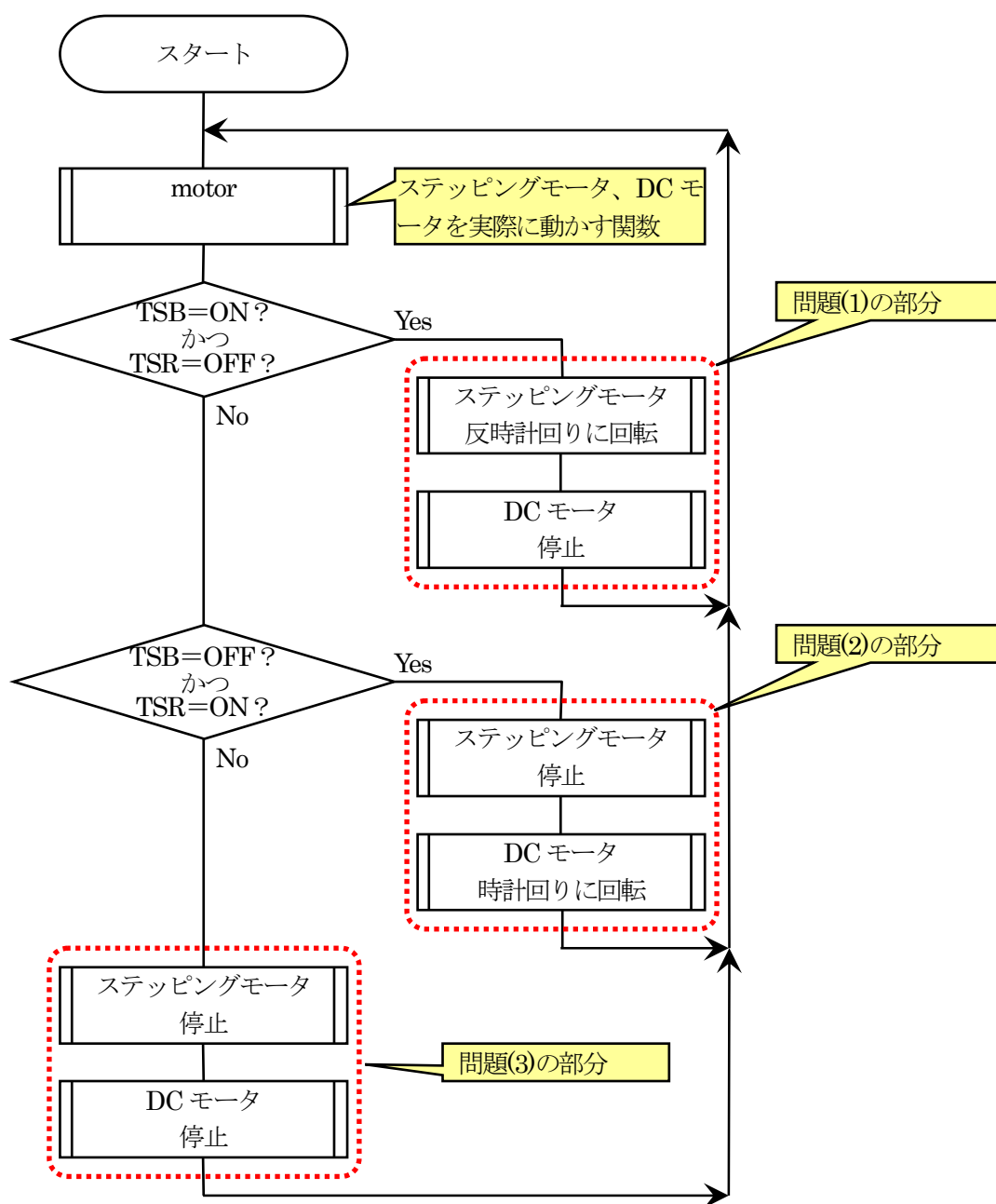
- (1) タクトスイッチ TSB のみが ON のときステッピングモータが反時計回りに回転する。
- (2) タクトスイッチ TSR のみが ON のとき直流モータが時計回りに回転する。
- (3) タクトスイッチの状態が上の (1)、(2) 以外のときのときはステッピングモータ、直流モータともに停止状態になる。
- (4) モータの回転速度はいずれも目視で確認できること。
- (5) 2つの7セグメント LED は消灯していること。

##### 動作概要



▲大会当日配付資料より抜粋

### 3.7.2 フローチャート



## 3.7.3 プログラム例

```

1 : //////////////////////////////////////
2 : // 第9回高校生ものづくりコンテスト全国大会 電子回路部門 課題3
3 : // Copyright (C) 2009 ルネサステクノロジ マイコンカーラリー事務局
4 : //////////////////////////////////////

```

5～192行は、課題1と同じため省略

```

193 : //////////////////////////////////////
194 : // メイン関数
195 : //////////////////////////////////////
196 : void main( void )
197 : {
198 :     init();
199 :
200 :     set7seg( LEFT , -1 );
201 :     set7seg( RIGHT , -1 );
202 :
203 :     while( 1 ) {
204 :         motor();                // DCモータ、ステッピングモータへ信号出力
205 :
206 :         if( getTSB() == ON && getTSR() == OFF ) {
207 :             stepper_motor( 50, -1 );    // ステッピングモータ：反時計回り
208 :             dc_motor( 0 );              // DCモータ：停止
209 :             set7seg( LEFT , -1 );
210 :             set7seg( RIGHT , -1 );
211 :         } else if( getTSB() == OFF && getTSR() == ON ) {
212 :             stepper_motor( 50, 0 );      // ステッピングモータ：停止
213 :             dc_motor( 1 );              // DCモータ：時計回り
214 :             set7seg( LEFT , -1 );
215 :             set7seg( RIGHT , -1 );
216 :         } else {
217 :             stepper_motor( 50, 0 );      // ステッピングモータ：停止
218 :             dc_motor( 0 );              // DCモータ：停止
219 :             set7seg( LEFT , -1 );
220 :             set7seg( RIGHT , -1 );
221 :         }
222 :     }
223 : }
224 :
225 : //////////////////////////////////////
226 : // End of File
227 : //////////////////////////////////////

```

## 3.7.4 プログラムの解説

行	詳細
204	motor 関数は、実際にステッピングモータとDCモータを動かす関数です。 stepper_motor 関数、dc_motor 関数は、それぞれステッピングモータ、DCモータの動作を設定するだけでこれらの関数だけでは動作しません。必ず motor 関数とペアで使います。
207	ステッピングモータの動作を設定します。 stepper_motor 関数の使い方を次に示します。 stepper_motor ( 回転速度, 回転方向 ) 回転速度: 値が小さいほど速く回転します。ただし、小さすぎると脱調します。停止は0です。 回転方向: 1 で時計回り、-1 で反時計回り、0 で停止です。 例) 回転速度 50 で反時計回りに回転 : stepper_motor( 50, -1 );
208	DCモータの動作を設定します。 dc_motor 関数の使い方を次に示します。 dc_motor ( 回転方向 ) 回転方向: 1 で時計回り、-1 で反時計回り、0 で停止です。 例) 時計回りに回転 : dc_motor( 1 );

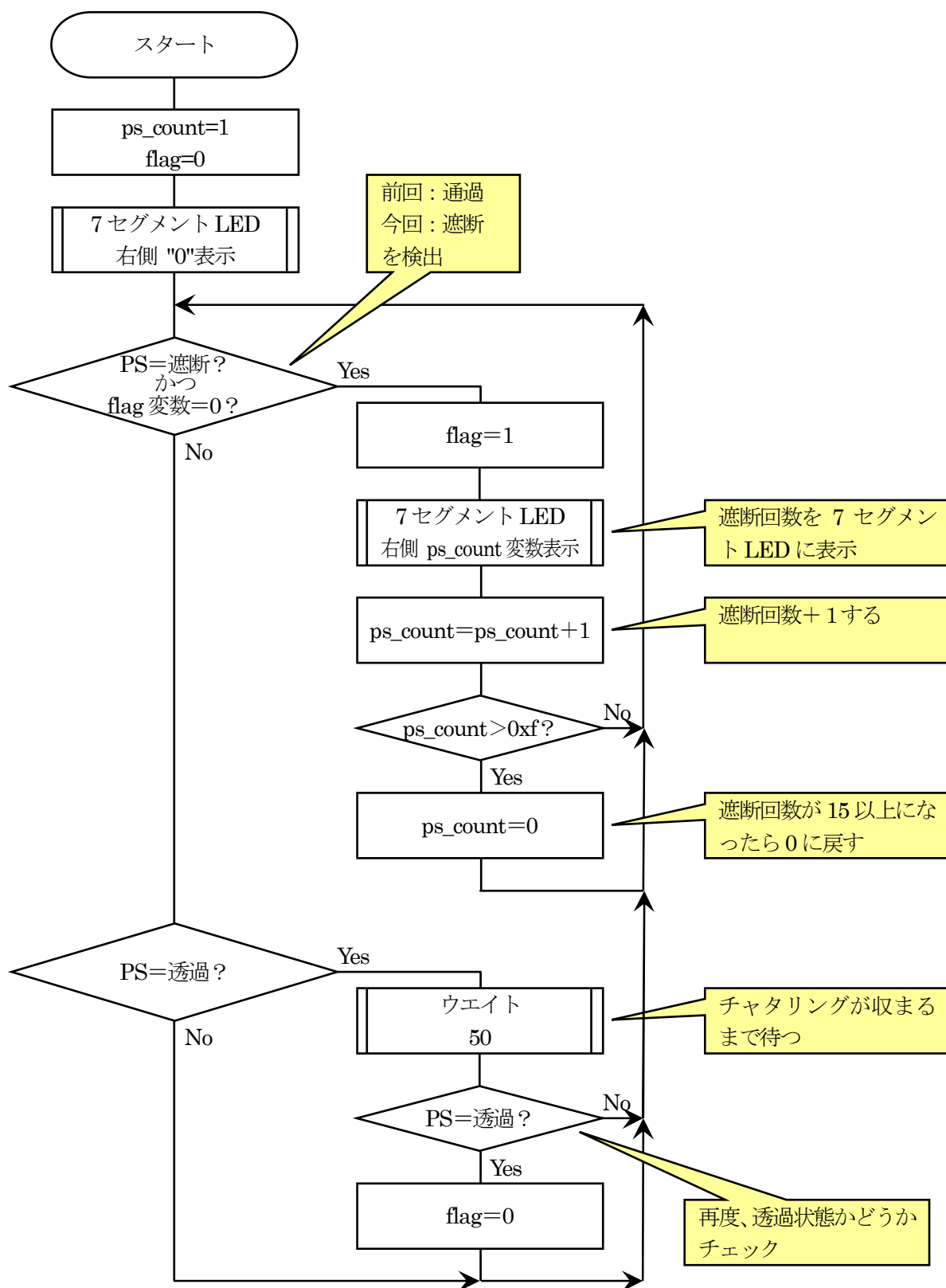
## 3.8 課題4

### 3.8.1 課題

- (1) フォトインタラプタの遮断回数を右側の7セグメントLEDに表示する。  
ただし遮断回数の表示は16進数1桁で0からFまでとし、Fを超えたら0に戻ってカウントを継続すること。
- (2) プログラムスタート時は右側7セグメントLEDは0が表示されているものとする。  
遮断回数のカウント方法は遮断→透過で1回とする。
- (3) 2つのモータはいずれも停止していること。
- (4) 左側7セグメントLEDは常に消灯していること。
- (5) タクトスイッチ TRB,TSR の状態に関係なく動作すること。

▲大会当日配付資料より抜粋

### 3.8.2 フローチャート



## 3.8.3 プログラム例

```

1 : //////////////////////////////////////
2 : // 第9回高校生ものづくりコンテスト全国大会 電子回路部門 課題4
3 : // Copyright (C) 2009 ルネサステクノロジ マイコンカーラリー事務局
4 : //////////////////////////////////////

```

5～192行は、課題1と同じため省略

```

193 : //////////////////////////////////////
194 : // メイン関数
195 : //////////////////////////////////////
196 : void main( void )
197 : {
198 :     int ps_count = 1;           // PSの遮断回数
199 :     int flag = 0;              // PSがONになったら1
200 :
201 :     init();                    // ポートの入出力設定
202 :
203 :     set7seg( LEFT , -1 );
204 :     set7seg( RIGHT , 0 );
205 :
206 :     while( 1 ) {
207 :         if( getPS() == SYADAN && flag == 0 ) { // PSが遮断された瞬間を検出
208 :             flag = 1;
209 :             set7seg( LEFT , -1 );
210 :             set7seg( RIGHT , ps_count );
211 :
212 :             ps_count++;          // カウント数+1
213 :             if( ps_count > 0xf ) ps_count = 0; // "f"を超えたら0にする
214 :
215 :             if( getPS() == TOUKA ) { // PS透過を検出
216 :                 wait( 50 );        // チャタリングが収まるまで待つ
217 :                 if( getPS() == TOUKA ) flag = 0; // 再度チェック、透過ならflagをクリア
218 :             }
219 :         }
220 :     }
221 :
222 :     //////////////////////////////////////
223 :     // End of File
224 :     //////////////////////////////////////

```

## 3.8.4 プログラムの解説

行	詳細
198	ps_count 変数は、フォトインタラプタの遮断回数をカウントする変数です。
199	flag 変数は、フォトインタラプタの遮断された瞬間を検出するための変数です。 207 行ではフォトインタラプタが遮断された状態で flag 変数が 0 なら遮断された瞬間と判断してカッコの中を実行します。208 行で flag 変数を 1 にします。 次に、207 行を実行したとき flag 変数は 1 なので遮断された瞬間ではないと判断してカッコの中は実行しません。このように、遮断した瞬間の 1 回だけカッコの中を実行させるために flag 変数を使います。 フォトインタラプタの遮断が解除された(透過状態)になったら、217 行で flag 変数を 0 にして、次に遮断されたときにカッコの中を実行するようにします。
212	フォトカブラの遮断回数を+1します。0xfを超えたら0に戻して、'f'の次は'0'が表示されるようにします。
216	wait 関数でウェイト(時間稼ぎ)をします。 wait 関数の使い方を次に示します。 wait ( ウェイト時間 1 当たり約 $10\mu s$ ) 今回は、50 を設定していますので約 $500\mu s$ の時間稼ぎとなります。 wait を入れると制御が止まってしまいますので、多用しないようにします。

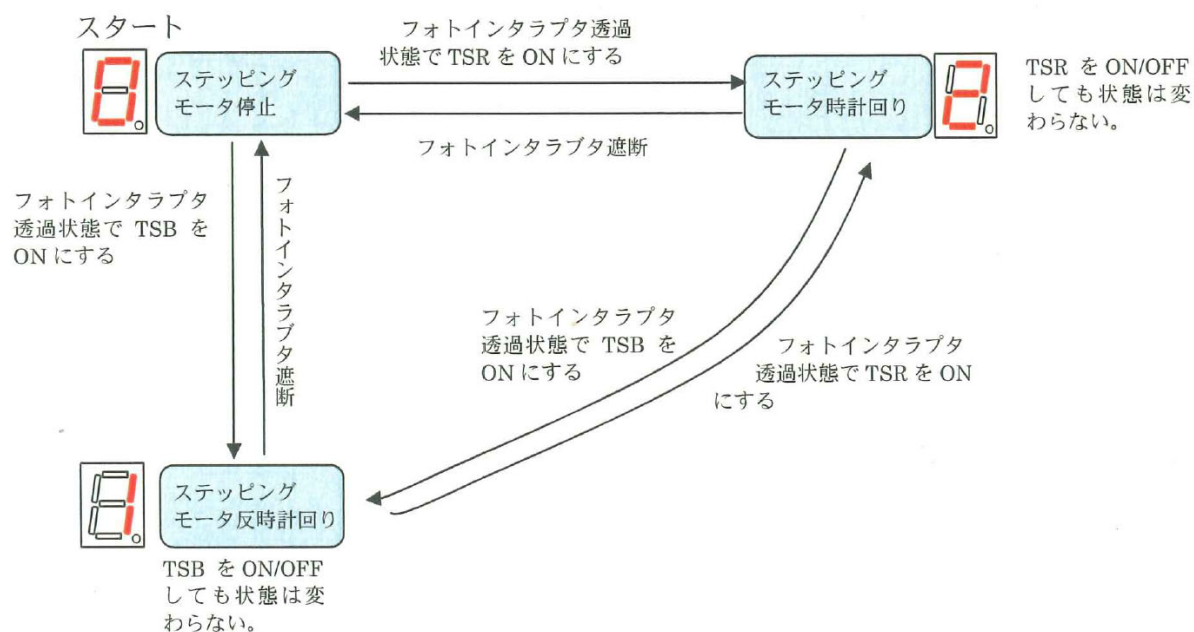


## 3.9 課題5

### 3.9.1 課題

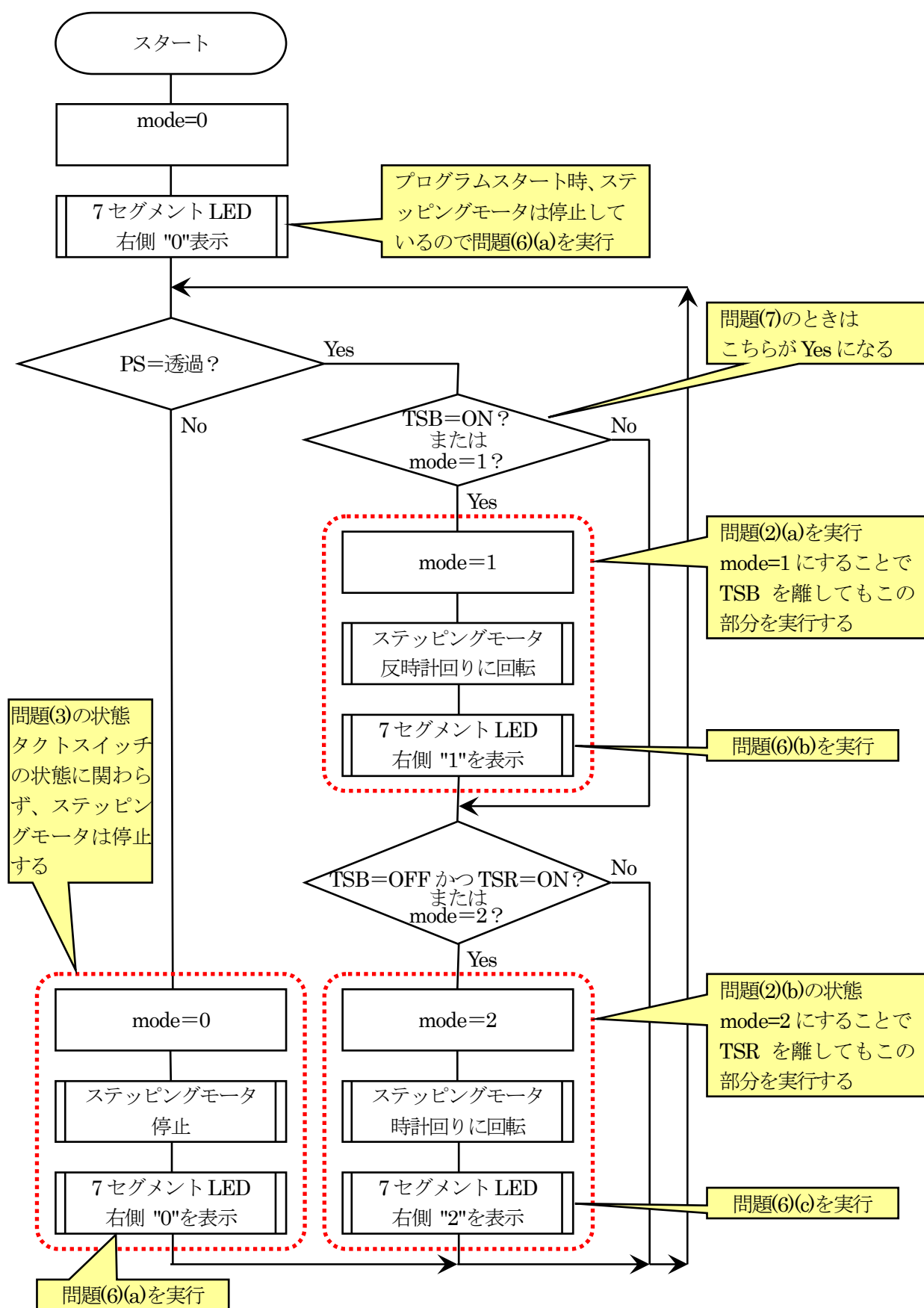
- (1) プログラムスタート時、ステッピングモータは停止している。
- (2) フォトインタラプタが透過状態のとき
  - (a) タクトスイッチ TSB を一度 ON にするとステッピングモータが反時計回りに回転し、タクトスイッチ TSB を OFF にしてもステッピングモータはそのまま回転を継続する。
  - (b) タクトスイッチ TSR を ON にするとステッピングモータが時計回りに回転し、タクトスイッチ TSR を OFF にしてもステッピングモータはそのまま回転を継続する。
- (3) フォトインタラプタが遮断状態のとき、タクトスイッチの状態に関わらずステッピングモータは停止する。
- (4) 停止したステッピングモータを再び動作させるためには (2) の (a)、(b)の状態にする。
- (5) 左側7セグメントLEDは常に消灯していること。直流モータは常に静止していること。
- (6) 右側7セグメントLEDはステッピングモータの状態によって次のように点灯すること。
  - (a) 停止状態のとき '0'を表示する。
  - (b) 反時計回りのとき '1'を表示する。
  - (c) 時計回りのとき '2'を表示する。
- (7) フォトインタラプタが透過状態で TSR,TSB が同時に ON になった場合は、(2) の (a) または (b) のどちらか一方の動作をすること。

#### 動作概要



▲大会当日配付資料より抜粋

### 3.9.2 フローチャート



## 3.9.3 プログラム例

```

1 : //////////////////////////////////////
2 : // 第9回高校生ものづくりコンテスト全国大会 電子回路部門 課題5
3 : // Copyright (C) 2009 ルネサステクノロジ マイコンカーラリー事務局
4 : //////////////////////////////////////

```

5～192行は、課題1と同じため省略

```

193 : //////////////////////////////////////
194 : // メイン関数
195 : //////////////////////////////////////
196 : void main( void )
197 : {
198 :     int mode = 0;
199 :
200 :     init();
201 :     set7seg( LEFT , -1 );
202 :     set7seg( RIGHT , 0 );
203 :
204 :     while( 1 ) {
205 :         motor(); // DC モータ、ステッピングモータへ信号出力
206 :
207 :         if( getPS() == TOUKA ) {
208 :             // 透過型フォトインタラプタ 透過時
209 :             if( getTSB() == ON || mode == 1 ) {
210 :                 mode = 1; // TSB を離してもこの部分を実行するように
211 :                 stepper_motor( 50, -1 ); // ステッピングモータ反時計回り
212 :                 set7seg( LEFT , -1 );
213 :                 set7seg( RIGHT , 1 );
214 :             }
215 :             if( getTSB() == OFF && getTSR() == ON || mode == 2 ) {
216 :                 mode = 2; // TSR を離してもこの部分を実行するように
217 :                 stepper_motor( 50, 1 ); // ステッピングモータ時計回り
218 :                 set7seg( LEFT , -1 );
219 :                 set7seg( RIGHT , 2 );
220 :             }
221 :         } else {
222 :             // 透過型フォトインタラプタ 遮断時
223 :             mode = 0;
224 :             stepper_motor( 50, 0 );
225 :             set7seg( LEFT , -1 );
226 :             set7seg( RIGHT , 0 );
227 :         }
228 :     }
229 : }
230 :
231 : //////////////////////////////////////
232 : // End of File
233 : //////////////////////////////////////

```

## 3.9.4 プログラムの解説

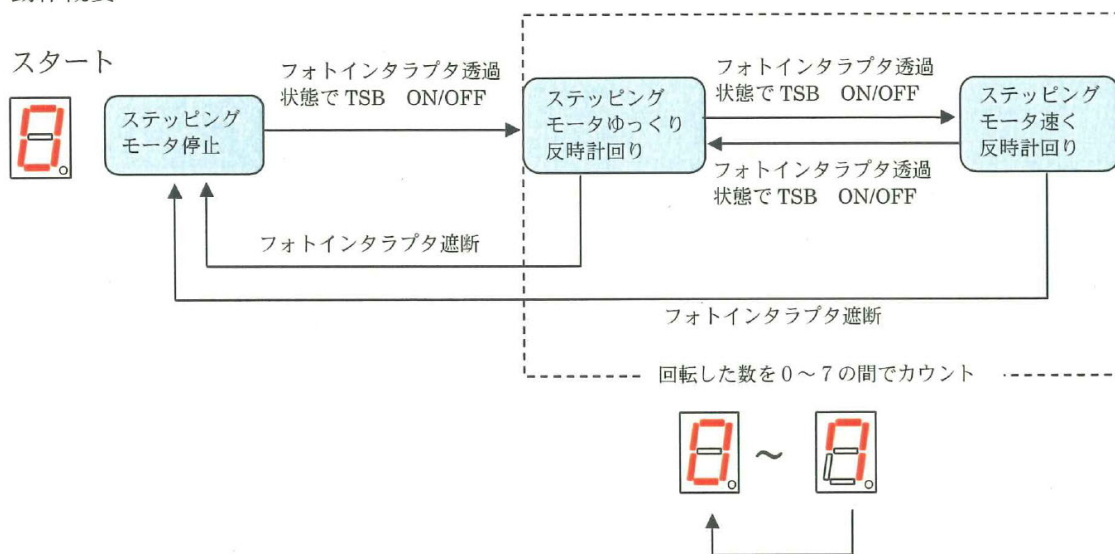
行	詳細
198	<p>mode 変数は、スイッチが押されたときに実行するプログラム部分を、スイッチが離されても実行させるために使用します。</p> <p>209 行でタクトスイッチブルーが ON になると、210 行で mode を 1 にします。タクトスイッチブルーが OFF になっても 209 行の「mode==1」が成り立つため、カッコの中を実行します。</p> <p>同様に、215 行でタクトスイッチブルーが OFF の状態で、タクトスイッチレッドが ON なら、mode を 2 にします。タクトスイッチレッドが OFF になっても 215 行の「mode==2」が成り立つため、カッコの中を実行します。</p> <p>mode 変数の役割が分からない場合は、mode 変数の記述をすべて削除してプログラムを実行してみてください。</p>

## 3.10 課題6

### 3.10.1 課題

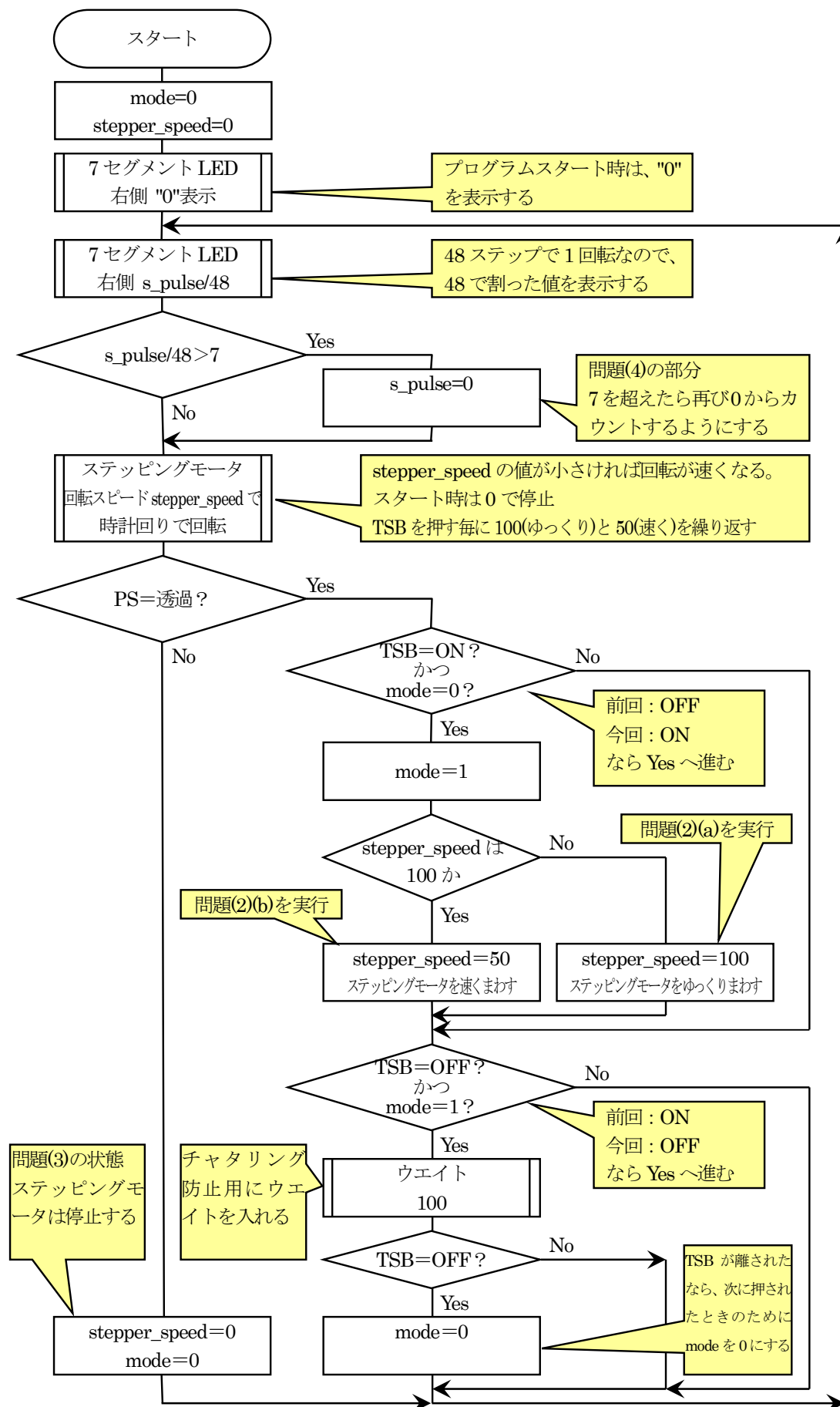
- (1) プログラムスタート時、2つのモータは停止していること。
- (2) フォトインタラプタが透過状態のとき  
タクトスイッチ TSB を ON/OFF するたびに状態が次の(a)と(b)の間を移る。  
(a) ステッピングモータがゆっくりとまわる。  
(b) ステッピングモータが(a) よりも速くまわる。  
なお最初は(a)から始めるものとする。
- (3) フォトインタラプタが遮断状態のとき  
ステッピングモータは停止する。停止したモータは(2)の条件を満たすまで動作しない。
- (4) 右側7セグメントLEDはステッピングモータの回転した数をカウントする。  
プログラムスタート時は0を表示する。7を超えたら再び0からカウントする。  
モータが途中で停止した場合はカウンタの値をそのまま表示しつづけ、回転が再開したらカウントを継続する。
- (5) 左側7セグメントLEDは常に消灯していること。  
ステッピングモータは目視でゆっくり回るのと速く回る違いが区別できること。

#### 動作概要



▲大会当日配付資料より抜粋

### 3.10.2 フローチャート



## 3.10.3 プログラム例

```

1 : //////////////////////////////////////
2 : // 第9回高校生ものづくりコンテスト全国大会 電子回路部門 課題6
3 : // Copyright (C) 2009 ルネサステクノロジ マイコンカーラリー事務局
4 : //////////////////////////////////////

```

5～192行は、課題1と同じため省略

```

193 : //////////////////////////////////////
194 : // メイン関数
195 : //////////////////////////////////////
196 : void main( void )
197 : {
198 :     int mode = 0;
199 :     int stepper_speed = 0;          // ステッピングモータのスピード
200 :
201 :     init();
202 :     set7seg( LEFT , -1 );
203 :     set7seg( RIGHT , 0 );
204 :
205 :     while( 1 ) {
206 :         // ステッピングモータの回転数を7セグに表示
207 :         set7seg( LEFT , -1 );
208 :         set7seg( RIGHT , s_pulse/48 );
209 :         if( s_pulse / 48 > 7 ) s_pulse = 0;    // 7を超えたら0にする
210 :
211 :         motor();                          // DCモータ、ステッピングモータへ信号出力
212 :         stepper_motor( stepper_speed, 1 );
213 :
214 :         if( getPS() == TOUKA ) {
215 :             // 透過型フォトインタラプタ 透過時
216 :             if( getTSB() == ON && mode == 0 ) { // TSBがONになった瞬間を検出
217 :                 mode = 1;
218 :                 if( stepper_speed == 100 ) { // ステッピングモータの速度変更
219 :                     stepper_speed = 50;      // 速く
220 :                 } else {
221 :                     stepper_speed = 100;      // ゆっくり
222 :                 }
223 :             }
224 :             if( getTSB() == OFF && mode == 1 ) {
225 :                 wait( 100 );                // チャタリングが収まるまで待つ
226 :                 if( getTSB() == OFF ) {      // 再度チェック、OFFならmodeをクリア
227 :                     mode = 0;
228 :                 }
229 :             }
230 :         } else {
231 :             // 透過型フォトインタラプタ 遮断時
232 :             stepper_speed = 0;                // ステッピングモータ停止
233 :             mode = 0;
234 :         }
235 :     }
236 : }
237 :
238 : //////////////////////////////////////
239 : // End of File
240 : //////////////////////////////////////

```

## 3.10.4 プログラムの解説

行	詳細
198	<p>mode 変数は、スイッチが押された瞬間を検出するために使用します。</p> <p>最初、mode 変数は 0 です。216 行でタクトスイッチブルーが押されたら、217 行で mode 変数を 1 にします。次に 216 行が実行されても mode 変数は 1 なので、成り立たずカッコの中を実行しません。すなわちタクトスイッチブルーが押されたら 1 回だけカッコの中を実行します。</p> <p>224 行でタクトスイッチブルーが OFF、かつ mode 変数が 1 ならカッコの中を実行します。</p> <p>225 行でウェイト(約 1ms)を入れてチャタリングが収まるまで待ちます。226 行で再度タクトスイッチブルーが OFF かどうかチェックします。OFF なら 227 行で mode 変数を 0 にして、次にタクトスイッチブルーが押されるのを待ちます。</p>
199	<p>stepper_speed 変数は、ステッピングモータの回転速度を設定する変数です。今回のプログラムでは、この変数の値を次のように変化させます。</p> <ul style="list-style-type: none"> <li>・停止 (stepper_speed=0)</li> <li>・速く回転 (stepper_speed=50)</li> <li>・ゆっくり回転 (stepper_speed=100)</li> </ul> <p>218 行で、stepper_speed 変数が 100 の場合、50 に再設定、それ以外の場合は 100 に再設定しています。これは</p> <ul style="list-style-type: none"> <li>・0(停止状態)の場合、stepper_speed を 100(ゆっくり回転)にする</li> <li>・50(速く回転)の場合、stepper_speed を 100(ゆっくり回転)にする</li> <li>・100(ゆっくり回転)の場合、stepper_speed を 50(速く回転)にする</li> </ul> <p>にしています。問題の(1)「なお最初は(a)から始めるものとする」とありますので、停止状態の次は 100 (ゆっくり回転)になります。</p>
208	<p>今回の課題は、ステッピングモータの回転数を右側の 7 セグメント LED に点灯させます。今回のステッピングモータは、48 パルス加えると 1 回転です。</p> <p>s_pulse 変数は、stepper_motor 関数内でステッピングモータに 1 パルス加える毎に +1 されます。そのため、「s_pulse 変数 ÷ 48」がステッピングモータの回転数になります。この値を、7 セグメント LED の右側に表示させます。</p>
209	<p>今回の課題は、「ステッピングモータの回転数を表示する 7 セグメント LED が 7 を超えたら 0 にして再びカウントすること」となっています。「s_pulse 変数 ÷ 48」がステッピングモータの回転数になりますので、この値が 8 以上になったら 0 にします。</p>



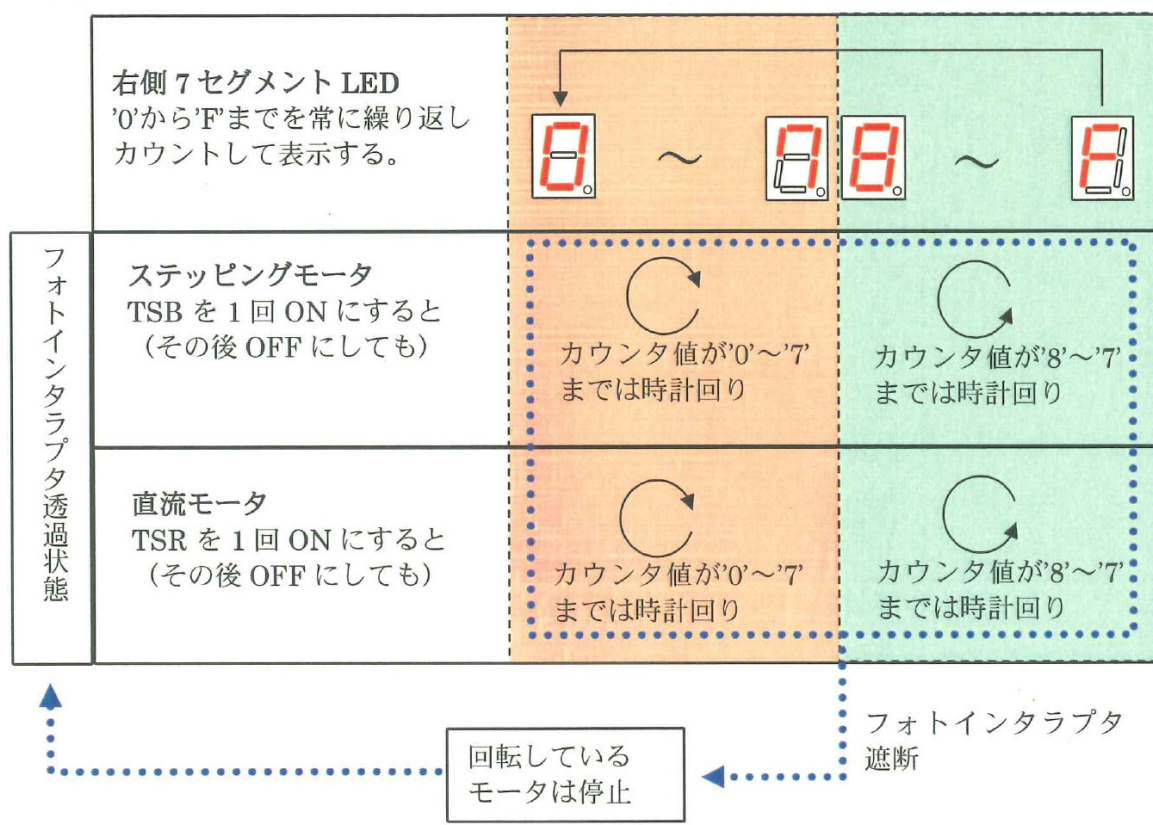
### 3.11 課題7

#### 3.11.1 課題

課題7の問題を、下記に示します。

- (1) プログラムスタート時2つのモータは停止、右側7セグメントLEDはプログラムスタート後ただちに16進数0～Fまでを約1秒間隔で繰り返し表示し続ける。
- (2) フォトインタラプタが透過状態のとき
  - (a) タクトスイッチTSBを一度ONにすると右側7セグメントLEDのカウントが0から7までの間は、ステッピングモータが時計回り、8からFまでは反時計回りに回転すること。  
また直流モータは停止させること。
  - (b) タクトスイッチTSRを一度ONにすると右側セグメントLEDのカウントが0から7までの間は、直流モータが時計回り、8からFまでは反時計回りに回転すること。  
またステッピングモータは停止させること。
  - (c) モータは同時に二つ動かさない。
    - (a) ではステッピングモータが停止するまでTSRのONを受け付けけないこと。
    - (b) では直流モータが停止するまでTSBのONを受け付けけないこと。
- (3) フォトインタラプタを遮断したとき
  - 回転しているステッピングモータまたは直流モータは停止させること。
  - 停止したモータは(2)の条件を満たすまで動かないこと。
  - (1)のカウナは継続させること。
- (4) 左側7セグメントLEDは常に消灯していること

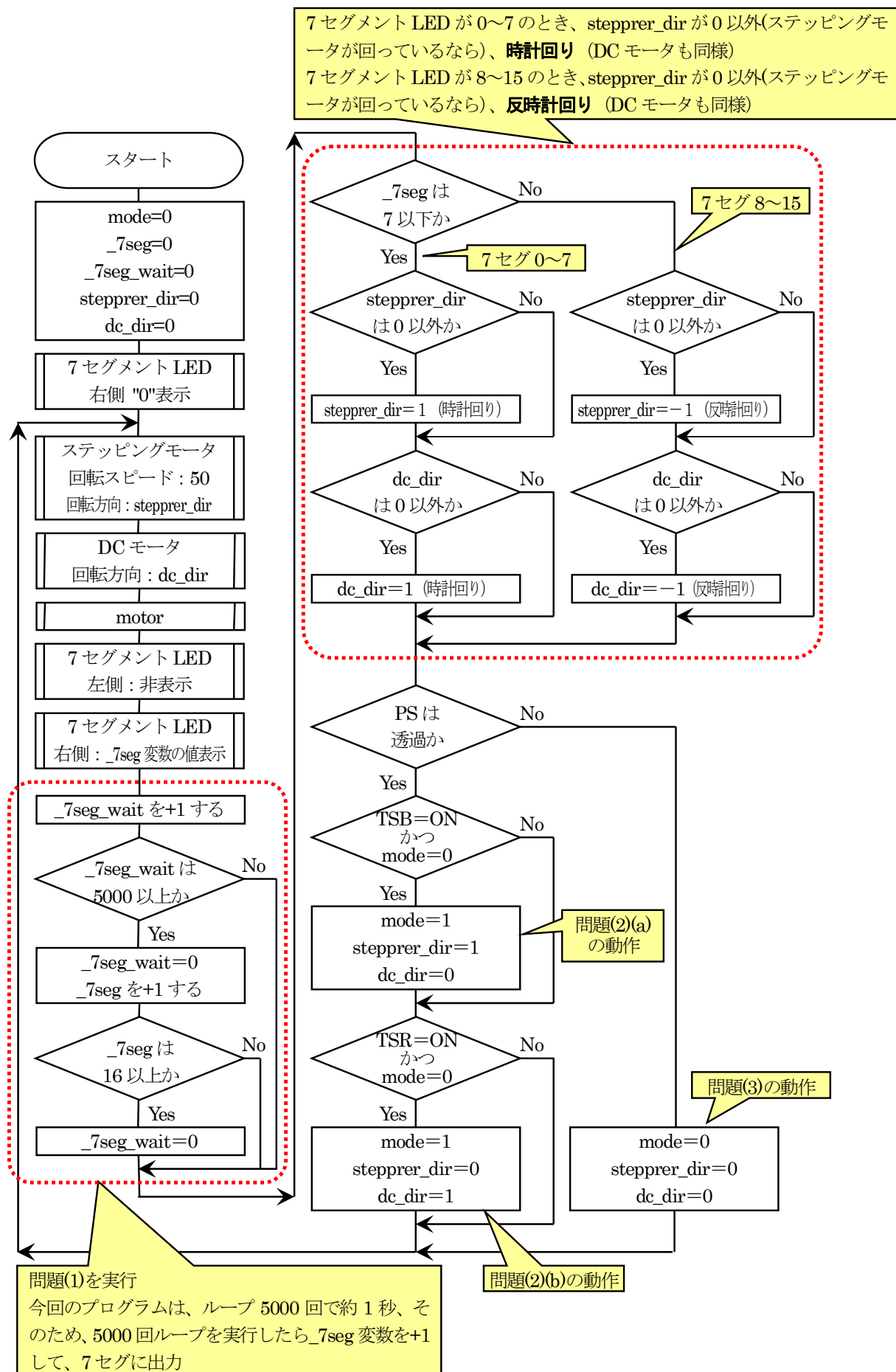
#### 動作概要



▲大会当日配付資料より抜粋



### 3.11.2 フローチャート



## 3.11.3 プログラム例

```

1 : //////////////////////////////////////
2 : // 第9回高校生ものづくりコンテスト全国大会 電子回路部門 課題7
3 : // Copyright (C) 2009 ルネサステクノロジ マイコンカーラリー事務局
4 : //////////////////////////////////////

```

5～192行は、課題1と同じため省略

```

193 : //////////////////////////////////////
194 : // メイン関数
195 : //////////////////////////////////////
196 : void main( void )
197 : {
198 :     int mode = 0;
199 :     int _7seg = 0;                // 7セグメントLEDの出力値
200 :     int _7seg_wait = 0;
201 :     int stepprer_dir = 0;         // ステッピングモータの方向
202 :     int dc_dir = 0;              // DCモータの方向
203 :
204 :     init();
205 :     set7seg( LEFT , -1 );
206 :     set7seg( RIGHT , 0 );
207 :
208 :     while( 1 ) {
209 :         // モータ制御
210 :         stepper_motor( 50, stepprer_dir );
211 :         dc_motor( dc_dir );
212 :         motor();                  // DCモータ、ステッピングモータへ信号出力
213 :
214 :         // 7セグメントLED表示
215 :         set7seg( LEFT , -1 );
216 :         set7seg( RIGHT , _7seg );
217 :
218 :         // 7セグメントLED表示値のインクリメント
219 :         _7seg_wait++;
220 :         if( _7seg_wait >= 5000 ) { // 5000で約1秒
221 :             _7seg_wait = 0;
222 :             _7seg++;
223 :             if( _7seg >= 16 ) _7seg = 0;
224 :         }
225 :
226 :         // ステッピングモータ、DCモータの回転方向制御
227 :         if( _7seg <= 7 ) {        // 0～7なら
228 :             if( stepprer_dir != 0 ) stepprer_dir = 1; // 時計回り
229 :             if( dc_dir != 0 ) dc_dir = 1;           // 時計回り
230 :         } else {                  // 8～15なら
231 :             if( stepprer_dir != 0 ) stepprer_dir = -1; // 反時計回り
232 :             if( dc_dir != 0 ) dc_dir = -1;           // 反時計回り
233 :         }
234 :
235 :         if( getPS() == TOUKA ) {
236 :             // 透過型フォトインタラプタ 透過時
237 :             if( getTSB() == ON && mode == 0 ) {
238 :                 mode = 1;
239 :                 stepprer_dir = 1; // ステッピングモータ：時計回り
240 :                 dc_dir = 0;      // DCモータ：停止
241 :             }
242 :             if( getTSR() == ON && mode == 0 ) {
243 :                 mode = 1;
244 :                 stepprer_dir = 0; // ステッピングモータ：停止
245 :                 dc_dir = 1;      // DCモータ：時計回り
246 :             }
247 :         } else {
248 :             // 透過型フォトインタラプタ 遮断時
249 :             mode = 0;
250 :             stepprer_dir = 0; // ステッピングモータ：停止
251 :             dc_dir = 0;      // DCモータ：停止
252 :         }
253 :     }

```

```

254 : }
255 :
256 : //////////////////////////////////////
257 : // End of File
258 : //////////////////////////////////////

```

### 3.11.4 プログラムの解説

行	詳細
198	<p>mode 変数は、スイッチが押された瞬間を検出するために使用します。</p> <p>最初、mode 変数は 0 です。237 行でタクトスイッチブルーが押されたら、238 行で mode 変数を 1 にします。次に 237 行が実行されても mode 変数は 1 なので、成り立たずカッコの中を実行しません。すなわちタクトスイッチブルーが押されたら 1 回だけカッコの中を実行します。</p> <p>同様に、242 行のタクトスイッチレッドが押されたら 1 回だけカッコの中を実行します。</p> <p>249 行でフォトインタラプタが遮断されたときに mode 変数が 0 になり、次にタクトスイッチブルー、またはタクトスイッチレッドが押されるのを待ちます。</p>
199 200	<p>_7seg 変数は、7 セグメント LED に出力する値を設定する変数です。最初に「_(アンダーバー)」が付いていますが、これは、変数の先頭に数字は使えないため、「_」を入れているだけです。</p> <p>_7seg_wait 変数は、_7seg 変数の値を +1 するタイミングを計算するための変数です。</p> <p>208 行にある while 文の無限ループを 1 回実行するたびに、219 行で _7seg_wiat 変数を +1 します。</p> <p>_7seg_wiat 変数がある値になったら、_7seg 変数を +1 します。</p> <p>今回の課題は、約 1 秒ごとに、7 セグメント LED の値を +1 するという問題です。そのため、220 行目</p> <pre>220 :         if( _7seg_wait &gt;= ???? ) {</pre> <p>の「????」部分をいろいろと変えて、7 セグメント LED が約 1 秒ごとに +1 する値を見つけます。今回は 5000 が約 1 秒でした。そのため、220 行には「5000」を代入します。</p> <p>_7seg_wiat 変数が 5000 になったら約 1 秒経ったと判断して、222 行で _7seg 変数を +1 します。課題は、「16 進数 0～F までは約 1 秒間隔で繰り返し表示し続ける」とあるので、223 行で 16 以上になったら 0 します。</p>
201	<p>stepprer_dir 変数は、ステッピングモータの回転方向を設定する変数です。dir は、「direction: 方向」の略です。値は、0: 停止、1: 時計回り、-1: 反時計回り になります。</p> <p>239 行でタクトスイッチブルーが押されたら、stepprer_dir を 1 にしてステッピングモータを時計回りに回すようにします。</p> <p>227 行～233 行でステッピングモータが回転中の場合、7 セグメント LED が 0～7 なら時計回り、8～F なら反時計回りになるように設定します。stepprer_dir が 0 なら、停止のままにしておきます。</p>
202	<p>dc_dir 変数は、DC モータの回転方向を設定する変数です。dir は、「direction: 方向」の略です。値は、0: 停止、1: 時計回り、-1: 反時計回り になります。</p> <p>245 行でタクトスイッチレッドが押されたら、dc_dir を 1 にして DC モータを時計回りに回すようにします。</p> <p>227 行～233 行で DC モータが回転中の場合、7 セグメント LED が 0～7 なら時計回り、8～F なら反時計回りになるように設定します。dc_dir が 0 なら、停止のままにしておきます。</p>
210～ 212	<p>stepper_motor 関数で、stepprer_dir に設定している方向でステッピングモータの回転を設定します。</p> <p>dc_motor 関数で、dc_dir に設定している方向で DC モータの回転を設定します。</p> <p>motor 関数で、ステッピングモータ、DC モータを回転させます。</p>