

マイコンカーラリーキット  
動作確認マニュアル  
マイコンカーキット  
Ver.4 版

・モータドライブ基板 Vol.3  
・センサ基板 Ver.4  
に対応したマイコンカーキットの動作確認手順を説明しています。

第 1.00 版  
2007.06.05  
ジャパンマイコンカーラリー実行委員会

# 注意事項 (rev.1.2)

## 著作権

- ・本マニュアルに関する著作権はジャパンマイコンカーラリー実行委員会に帰属します。
- ・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、文章によるジャパンマイコンカーラリー実行委員会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、ジャパンマイコンカーラリー実行委員会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、ジャパンマイコンカーラリー実行委員会は、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。製作に当たっては、こと前にマイコンカー公式ホームページ(<http://www.mcr.gr.jp/>)などを通じて公開される情報に常にご注意ください。

## 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

# 目 次

1. 概要.....	1
2. キットの構成.....	2
3. 動作確認用プログラムの書き込み.....	3
3.1 ワークスペース「kit07」を開く.....	3
3.2 プロジェクトの変更.....	4
3.3 テストプログラムの書き込み.....	5
4. 動作テスト.....	8
4.1 動作テスト一覧.....	8
4.2 LED のテスト.....	9
4.3 タクトスイッチのテスト.....	9
4.4 サーボのテスト.....	9
4.5 右モータのテスト.....	11
4.6 左モータのテスト.....	11
4.7 センサのテスト.....	12
4.8 直進テスト.....	14
4.9 テスト終了.....	14
5. プログラムソース.....	15
5.1 「kit07test.c」のプログラム内容.....	15
5.2 「kit07teststart.src」のプログラム内容.....	19

## 1. 概要

本マニュアルは、

センサ基板 Ver.4 製作マニュアル  
モータドライブ基板 Vol.3 製作マニュアル  
本体組み立て製作マニュアル(Ver.4 版)

で製作、組み立てたマイコンカーの動作確認方法を説明するマニュアルです。  
動作確認をするためには、下記の手順で行います。

### 1. パソコンにルネサス統合開発環境をインストールします

既にインストールしている場合、必要ありません。インストールしていない場合は、「ルネサス統合開発環境操作マニュアル 導入編」を参照して、インストールしてください。

### 2. サンプルプログラム(動作テストプログラム)をインストールします

既にインストールしている場合、必要ありません。インストールしていない場合は、「ルネサス統合開発環境操作マニュアル 導入編」を参照して、インストールしてください。

### 3. プログラムをマイコンカーの CPU ボードに書き込みます

これから行います。

### 4. 本マニュアルに従って動作テストします

これから行います。

### 5. サーボセンタと最大切れ角を調整します

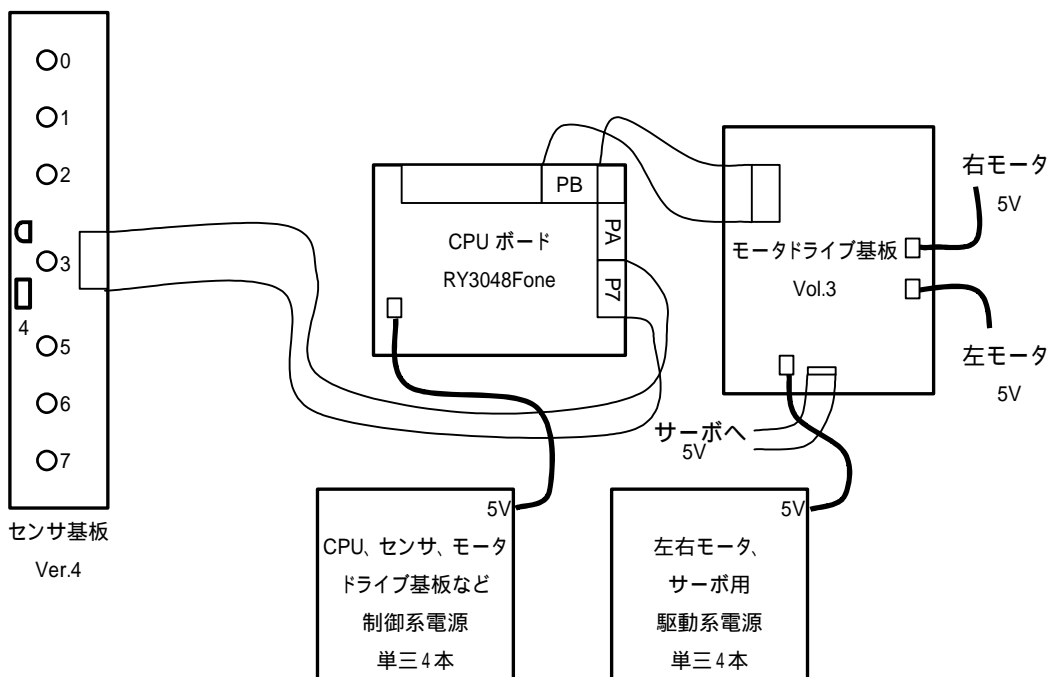
プログラム解説マニュアル kit07 版の「13. サーボセンタと最大切れ角の調整」を参照して、調整してください。

### 6. 走行プログラムを書き込みます

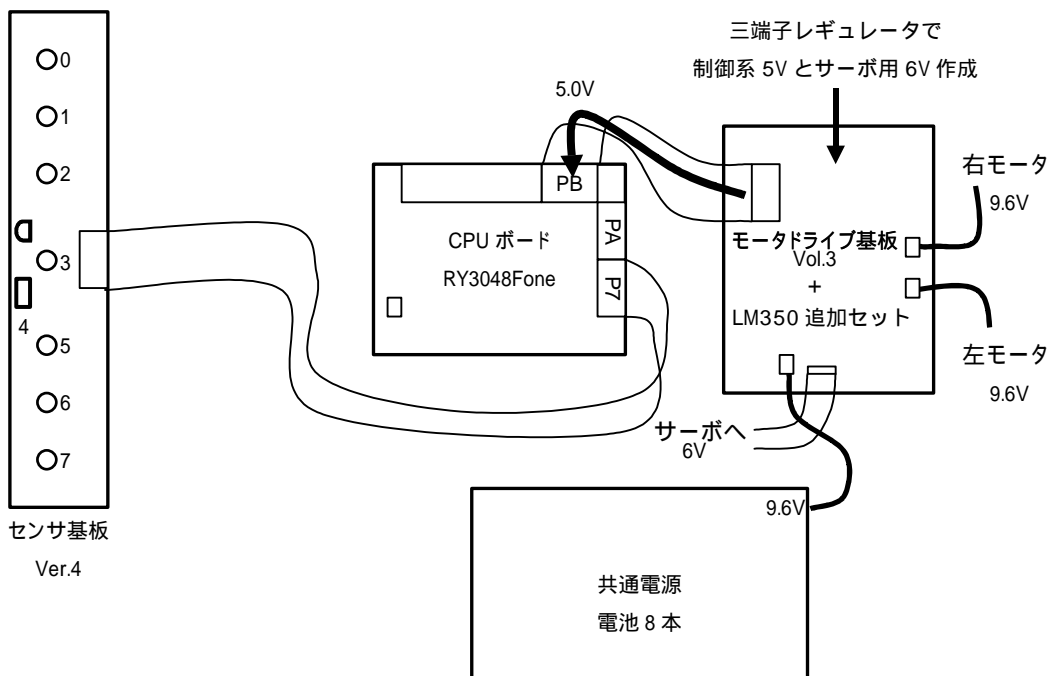
プログラム解説マニュアルを参照しながら、プロジェクト「kit07」の kit07.mot ファイルを CPU ボードに書き込み、コースを走らせてみてください。

## 2. キットの構成

下記のようなマイコンカーをテストすることができます。マイコンカーキット Ver.4 の構成です。



モータドライブ基板に、「LM350 追加セット」を追加し、電源電圧を上げても同じくテストできます。

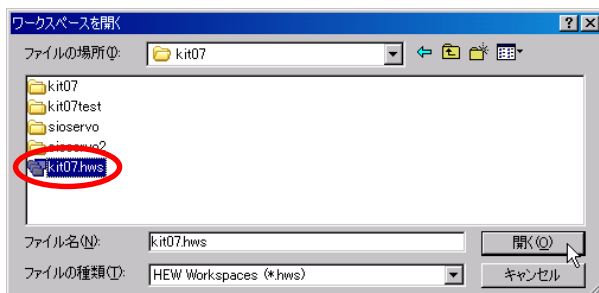
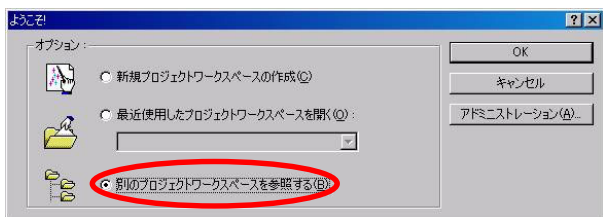


### 3. 動作確認用プログラムの書き込み

#### 3.1 ワークスペース「kit07」を開く

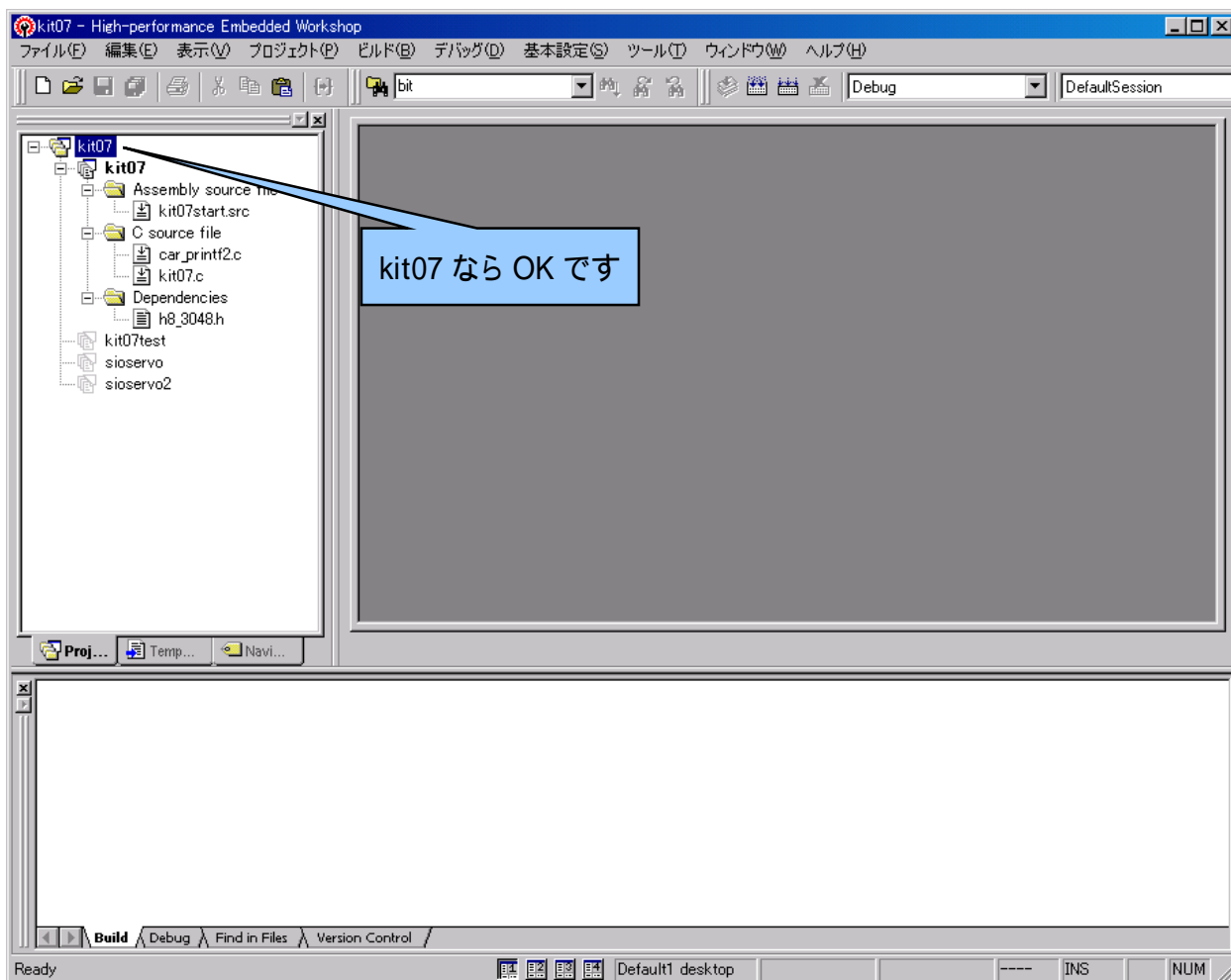


ルネサス統合開発環境を実行します。



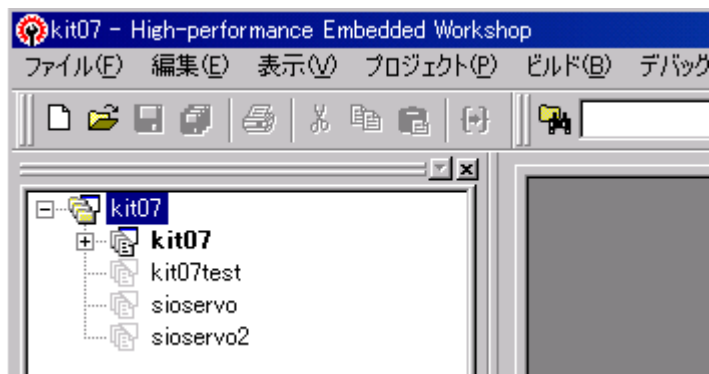
「別のプロジェクトワークスペースを参照する」を選択します。

Cドライブ Workspace kit07 の「kit07.hws」を選択します。



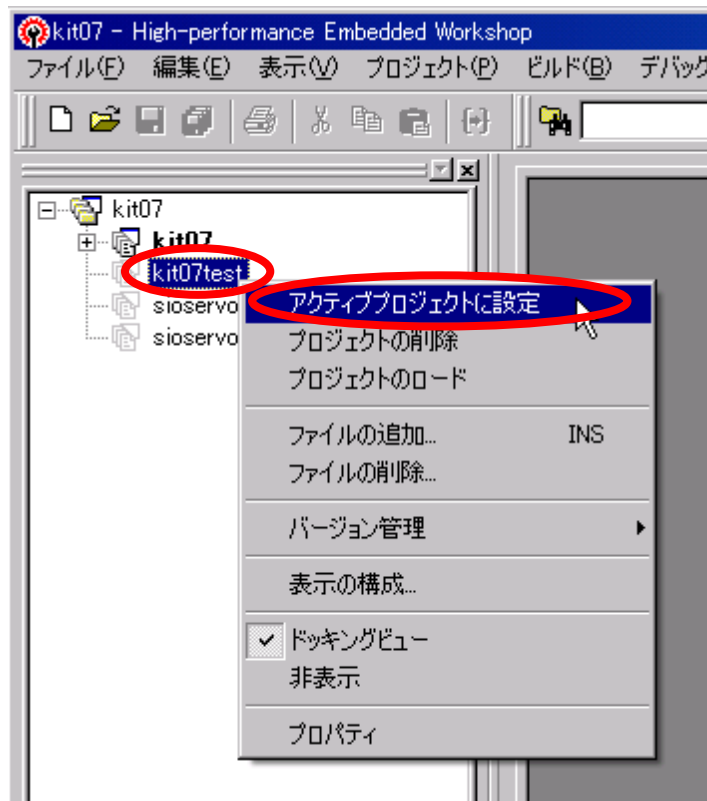
kit07 というワークスペースが開かれます。

### 3.2 プロジェクトの変更



ワークスペース「kit07」には、4つのプロジェクトが登録されています。

プロジェクト名	内容
kit07	マイコンカー走行プログラムです。
kit07test	<b>製作したマイコンカーのモータドライブ基板やセンサ基板が正しく動作するかテストします。</b>
sioservo	サーボのセンタを調整するプログラムです。後述します。
sioservo2	サーボの最大切れ角を見つけるためのプログラムです。後述します。

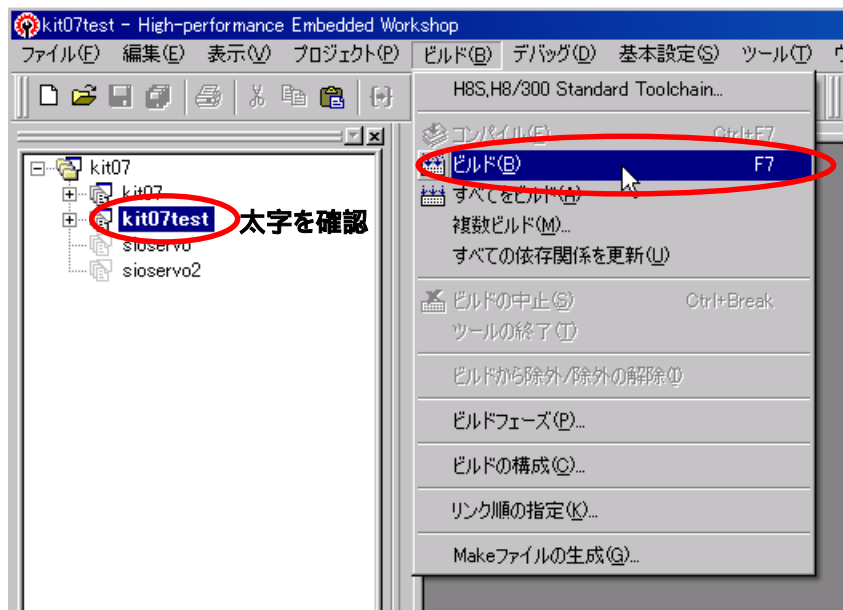


「kit07test」をアクティブプロジェクトに設定します。

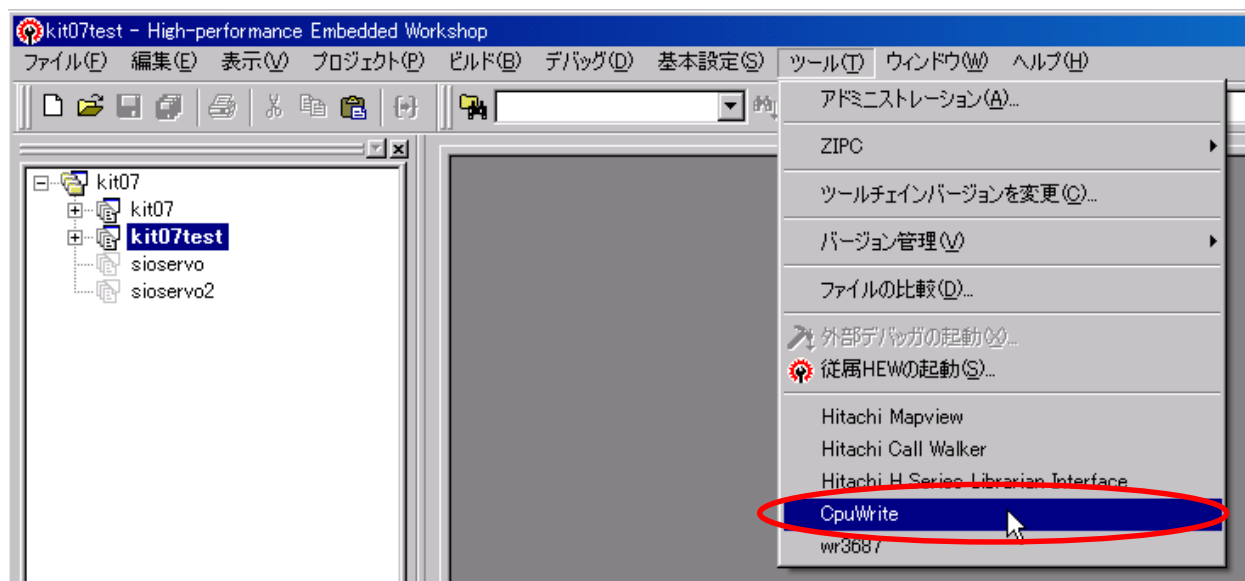
「kit07test」上で右クリックします。

「アクティブプロジェクトに設定」を選択します。

### 3.3 テストプログラムの書き込み



再度、「kit07test」が有効(太字)になっていることを確認して、「ビルド ビルド」を実行します。

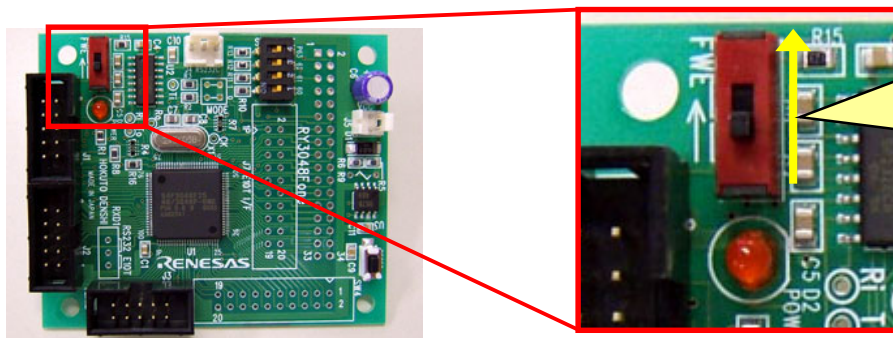


「ツール CpuWrite」で書き込みソフトを起動します。もし、CpuWrite コマンドが無い場合は、ルネサス統合開発環境操作マニュアル 導入編の「3.3 RY3048Fone ボードの場合のインストール、設定」を参照して、登録してください。





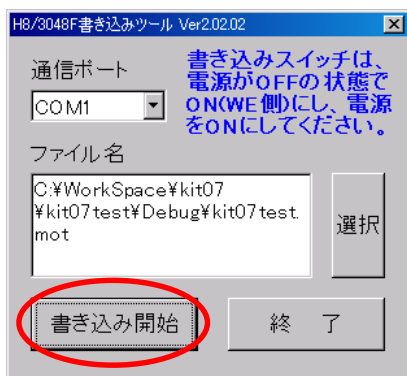
通信ケーブルをパソコンと CPU ボードの 3 ピンコネクタ(左写真)に接続します。



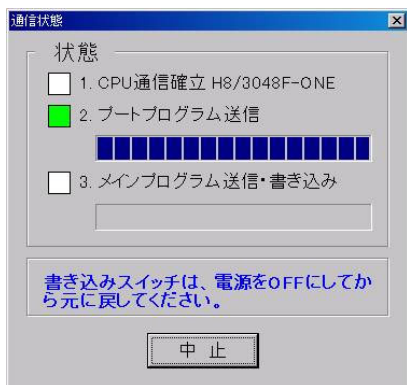
電源 OFF の状態で FWE 側に切り換えます。この状態がプログラム書き換え状態です。

CPU ボードの電源が切れていることを確認し、赤い書き込みスイッチを FWE 側に切り換えます。この状態がプログラム書き換え状態です。

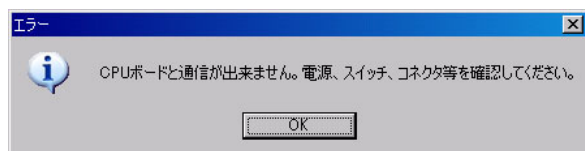
**！電源が入っている状態で、書き込みスイッチを切り換えると CPU ボードが壊れる可能性があります。必ず電源が切れていることを確認してください。CPU ボード上の LED が消えている状態です。**



CPU ボードの電源を入れます。LED が点灯するはずですが、書き込みソフトの「書き込み開始」をクリックします。



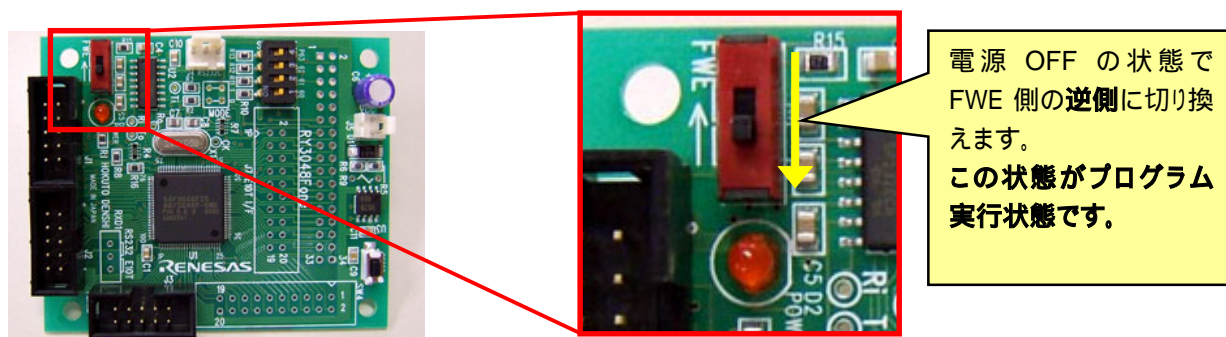
書き込み中です。正常に書き込みが終われば、自動的に書き込みソフトは終了します。



正常に書き込みができなければ、エラーメッセージが出力されます。正常に書き込まれない理由としては、

- (1)RS232C ケーブルの接続間違い、断線
- (2)パソコン側の問題。通信ポートが無効になっている、ポート番号が違っている、別の機器(赤外線ポートなど)がすでに通信ポートを使用しているなど
- (3)CPU ボード側の問題。書き込みスイッチの切り換え間違い、CPU が書き込み制限回数を超えている(メーカー保証回数は 100 回)、RXD1 切り換えスイッチが RS232 側になっていない( を参照してください)、CPU 電源電圧が 4.5 ~ 5.5V ではないなど

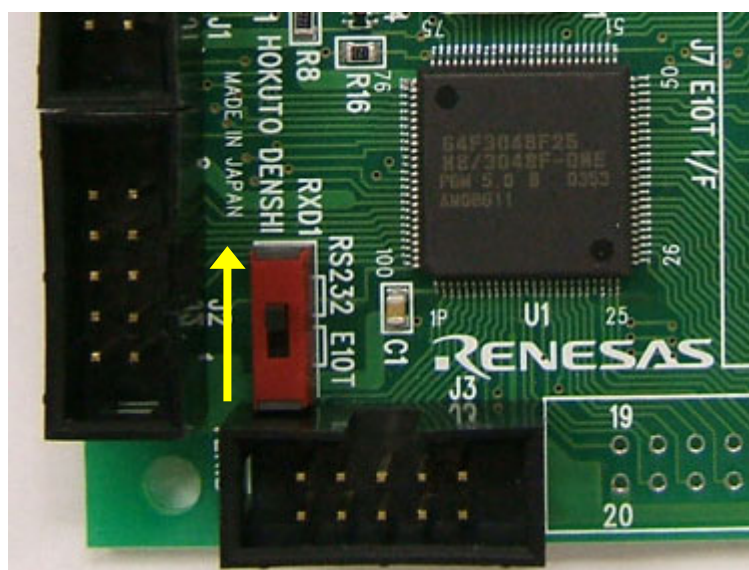
の理由が考えられます。これらの原因を解決して再度書き込みを行ってください。



書き込みが終わったら、CPU ボードの電源が切れていることを確認し、赤い書き込みスイッチを FWE 側とは**逆側**に切り換えます。**この状態がプログラム実行状態です。**

これでプログラムの書き込みは完了です。CPU ボードの電源を入ると、書き込んだプログラムが実行されます。

#### RXD1 部分に赤いスイッチのある CPU ボードの場合



2005 年度以前の RY3048Fone ボードには、RXD1 部分に赤いスイッチが付いています。その場合、**スイッチが RS232C 側になっているか確認してください。** E10T 側になっているとプログラムの書き込みができません。

ちなみに、2006 年度以降の CPU ボードは、このスイッチが無くなり、強制的に RS232 側になっています(パターンでショートしています)。

順調に書き込まれたならマイコンカーの電源を OFF にして、書き込みスイッチを通常側に戻します。

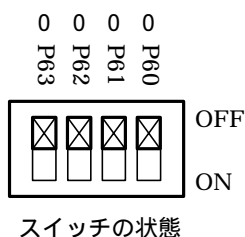
## 4. 動作テスト

### 4.1 動作テスト一覧

CPU ボードのディップスイッチの状態を変更することにより、マイコンカーのどの部分をテストするか選択し、動作のテストを行います。

				内容
P63	P62	P61	P60	
0	0	0	0	LED のテストをします。LED が 0.5 秒間隔で交互に点灯します。
0	0	0	1	プッシュスイッチのテストをします。 スイッチ OFF で LED0 が点灯、スイッチ ON で LED1 が点灯します。
0	0	1	0	サーボのテストをします。 サーボが、「0° 右 30° 左 30° の繰り返し」の動作をします。
0	0	1	1	何もしません。
0	1	0	0	右モータのテストをします。 「正転 ブレーキ」動作を繰り返します。
0	1	0	1	右モータのテストをします。 「逆転 ブレーキ」動作を繰り返します。
0	1	1	0	左モータのテストをします。 「正転 ブレーキ」動作を繰り返します。
0	1	1	1	左モータのテストをします。 「逆転 ブレーキ」動作を繰り返します。
1	0	0	0	コースのセンサテストをします。 センサ bit1,0 の状態を LED1,0 に出力します。
1	0	0	1	コースのセンサテストをします。 センサ bit3,2 の状態を LED1,0 に出力します。
1	0	1	0	コースのセンサテストをします。 センサ bit5,4 の状態を LED1,0 に出力します。
1	0	1	1	コースのセンサテストをします。 センサ bit7,6 の状態を LED1,0 に出力します。
1	1	0	0	直進テストをします。 PWM50%で前進、2 秒後にストップします。
1	1	0	1	直進テストをします。 PWM50%で前進、5 秒後にストップします。
1	1	1	0	直進テストをします。 PWM100%で前進、2 秒後にストップします。
1	1	1	1	直進テストをします。 PWM100%で前進、5 秒後にストップします。

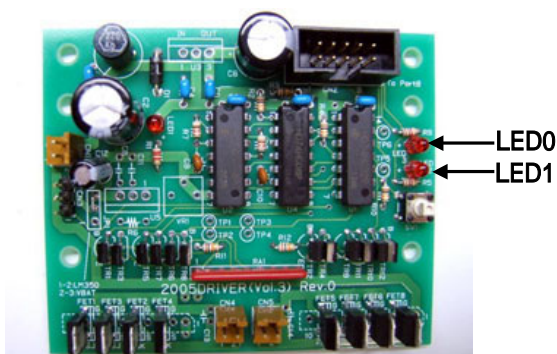
## 4.2 LED のテスト



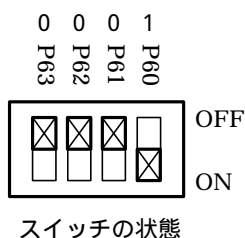
ディップスイッチを”0000”の状態ですべての CPU ボード用電源を ON します。ディップスイッチは、ON が”1”、OFF が”0”です。  
モータドライブ基板の LED0 と LED1 が 0.5 秒毎に交互に点灯します。

点灯しない場合は、CPU ボードとモータドライブ基板とを接続するフラットケーブルの不良、LED の半田付け不良、半田ブリッジ(ショート)、LED の向きが逆など考えられます。目視チェック、テスタ等で原因を突き止めてください。

モータドライブ基板の LED0、LED1 は下記の位置です。



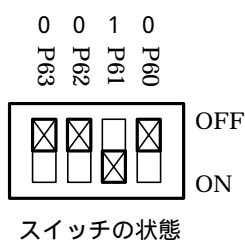
## 4.3 タクトスイッチのテスト



ディップスイッチを”0001”の状態ですべての CPU ボード用電源を ON します。  
モータドライブ基板上のスイッチが押されていない状態なら LED0 が点灯、押されたら LED1 が点灯します。

LED0 のみしか点灯しない場合は、スイッチまでの回路が半田付け不良、LED1 が付きっぱなしの場合は半田ブリッジが考えられます。目視チェック、テスタ等で原因を突き止めてください。

## 4.4 サーボのテスト



ディップスイッチを”0010”の状態ですべての CPU ボード用電源、モータ用電源を ON します。  
サーボが 1 秒毎に、「0 度 右 30 度 左 30 度」の動作を繰り返します。

サーボが動作しない場合は、サーボまでの回路が半田付け不良、サーボコネクタの向きが逆等が考えられます。また、モータドライブ基板の電源 LED が点灯しているかも確認してください。目視チェック、テスタ等で原因を突き止めてください。

キットに付属ではないサーボに交換したとき、「0度 左 30度 右 30度」の動作になる場合、左右の回転が逆なサーボです。その場合、handle 関数内の 412 行を下記のように変更します。

412 行	変更前	ITU4_BRB = SERVO_CENTER - angle * HANDLE_STEP;
	変更後	ITU4_BRB = SERVO_CENTER $\pm$ angle * HANDLE_STEP;

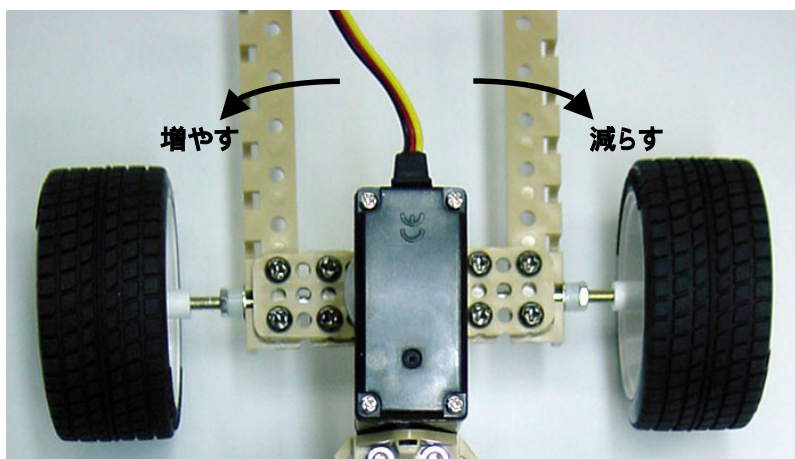
左右が入れ代わり、動作が「0度 右 30度 左 30度」となります。

### サーボのセンタ調整

ほとんどの場合、電源を ON にしても 0 度のはずなのにまっすぐ向いていないと思います。サーボのセンタ値は、サーボによってそれぞれ違うのでマイコンカー 1 台 1 台違う値になります。「kit06test.c」の SERVO\_CENTER の「5000」という値を変えてまっすぐになるように調整します。26 が約 1 度です。

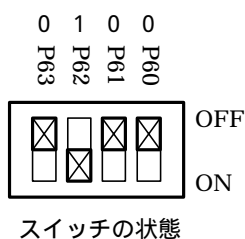
```
53 : #define          SERVO_CENTER    5000    /* サーボのセンタ値          */
```

値を増やせば進行方向に向かって左側、減らせば右側に向きます。



サーボの調整は、プログラム解説マニュアルの「13. サーボセンタと最大切れ角の調整」を参照しながら行うと便利です。

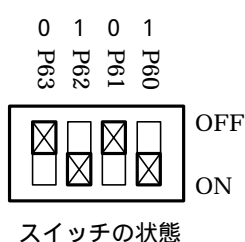
## 4.5 右モータのテスト



ディップスイッチを”0100”の状態に CPU ボード用電源、モータ用電源を ON します。  
右モータが 1 秒ごとに「正転 ブレーキ」を繰り返します。

右モータが正転しない場合は、右モータ制御回路の半田付け不良が考えられます。回転し続ける場合は、半田ブリッジしている可能性があります。目視チェック、テスト等で原因を突き止めてください。

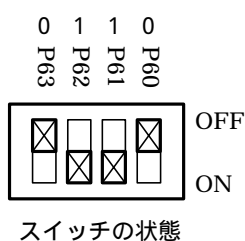
また、タイヤが逆転した場合は、モータのケーブルが逆です。コネクタの 1 ピンと 2 ピンの端子を入れ替えてください。



一度電源を OFF して、ディップスイッチを”0101”の状態に CPU ボード用電源、モータ用電源を ON します。  
右モータが 1 秒ごとに「逆転 ブレーキ」を繰り返します。

右モータが逆転しない場合は、半田付け不良やショートが考えられます。目視チェック、テスト等で原因を突き止めてください。

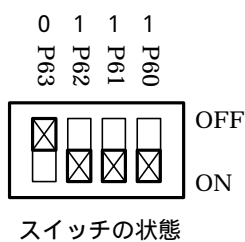
## 4.6 左モータのテスト



ディップスイッチを”0110”の状態に CPU ボード用電源、モータ用電源を ON します。  
左モータが 1 秒ごとに「正転 ブレーキ」を繰り返します。

左モータが正転しない場合は、左モータ制御回路の半田付け不良が考えられます。回転し続ける場合は、半田ブリッジしている可能性があります。目視チェック、テスト等で原因を突き止めてください。

また、タイヤが逆転した場合は、モータのケーブルが逆です。コネクタの 1 ピンと 2 ピンの端子を入れ替えてください。

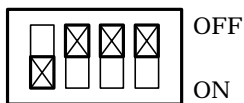


一度電源を OFF して、ディップスイッチを”0111”の状態に CPU ボード用電源、モータ用電源を ON します。  
左モータが 1 秒ごとに「逆転 ブレーキ」を繰り返します。

左モータが逆転しない場合は、半田付け不良やショートが考えられます。目視チェック、テスト等で原因を突き止めてください。

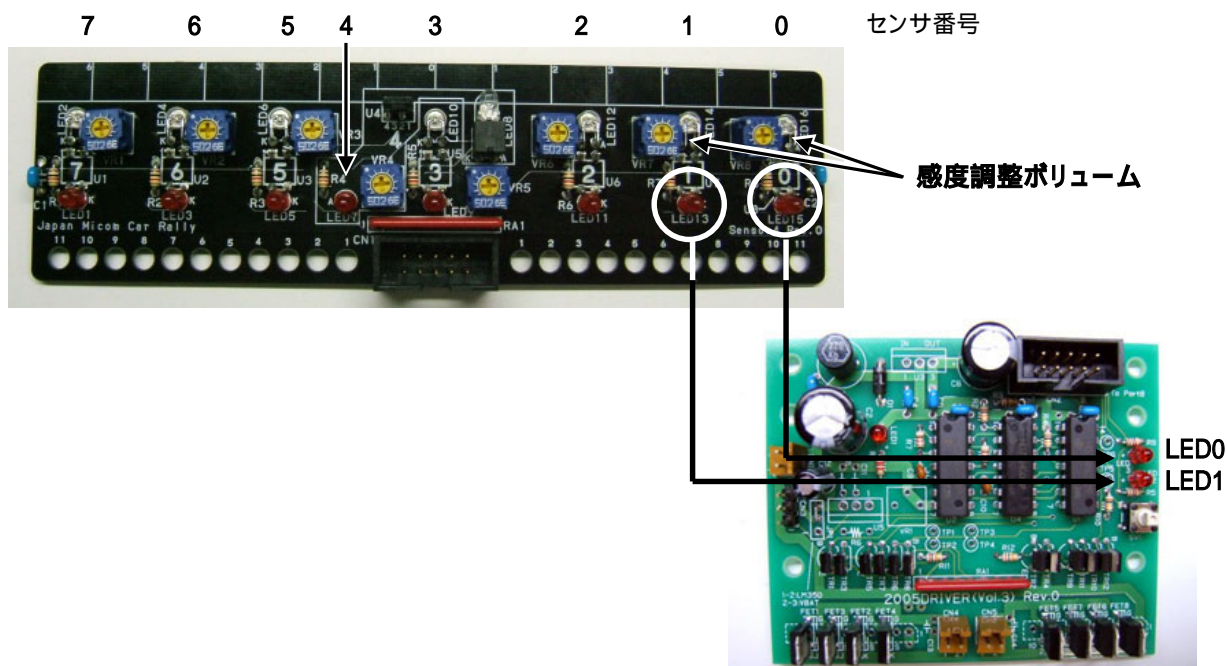
### 4.7 センサのテスト

1 0 0 0  
P63 P62 P61 P60



スイッチの状態

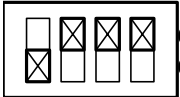
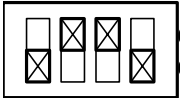
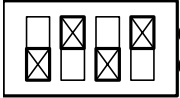
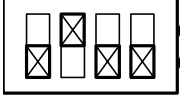
ディップスイッチを”1000”の状態に CPU ボード用電源を ON します。  
下図のようにセンサ基板のセンサ 0 とセンサ 1 の状態が、モータドライブ基板の 2 つの LED にモニタされます。センサ基板の LED も点灯しますので、同じ反応がテストします。センサの感度はボリュームで調整します。



もしセンサ基板の LED が点灯しない場合は、センサ基板の半田付け不良、半田ブリッジ、部品の逆差し等が考えられます。センサ基板の LED は点灯するのにモータドライブ基板の LED が点灯しない場合は、コネクタ周りで不具合が発生している可能性があります。目視チェック、テスト等で原因を突き止めてください。

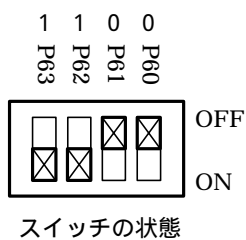
動作確認マニュアル(マイコンカーキット Ver.4 版)

次にスイッチを”1001”に換えるとセンサ 2, センサ 3, ”1010”ならセンサ 4, センサ 5, ”1011”ならセンサ 6, センサ 7 がモータドライブ基板の LED にモニタされます。

スイッチ	モータドライブ基板の LED1 に出力されるセンサ番号	モータドライブ基板の LED0 に出力されるセンサ番号
<p>1 0 0 0</p> 	1	0
<p>1 0 0 1</p> 	3	2
<p>1 0 1 0</p> 	5	4 (スタートバー検出)
<p>1 0 1 1</p> 	7	6



## 4.8 直進テスト



ディップスイッチを”1100”の状態に CPU ボード用電源、モータ用電源を ON します。

2秒後に PWM 値 50%で 2秒間直進します。廊下など平らで直線の長い場所でマイコンカーを走らせ、まっすぐに進むかテストします。曲がってしまう場合は、「4.4 サーボのテスト」で説明したとおり、SERVO\_CENTER の値を調整して直進するように調整します。直進性はマイコンカーのスピードが上がると非常に重要になりますので必ず行います。最終的には1ずつ値を微調整するぐらいの心がけで調整すると OK です。

直進テストは、PWM 値とストップするまでの時間の違いで 4 パターンあります。テスト走行する場所の広さに合わせて調整してください。

スイッチ	PWM 値	停止するまでの時間
1 1 0 0 	50%	2 秒
1 1 0 1 	50%	5 秒
1 1 1 0 	100%	2 秒
1 1 1 1 	100%	5 秒

## 4.9 テスト終了

全機能が正常に動作したなら、走行プログラムを書き込み、コースを走らせてみましょう。その前に、調整項目があります。

- ・プログラム解説マニュアル「13. サーボセンタと最大切れ角の調整」でサーボセンタと最大切れ角を調整してください。
- ・プロジェクト kit07 の「kit07.mot」ファイルを書き込み、コースを走らせてみてください。

## 5. プログラムソース

### 5.1 「kit07test.c」のプログラム内容

```

1 :  /******
2 :  /* マイコンカーテストプログラム "kit07test.c"
3 :  /* 2007.05 ジャパンマイコンカーラリー実行委員会
4 :  /******
5 :
6 :  /*
7 :  キット用センサ基板Ver.4、モータドライブ基板Vol.3の
8 :  テストを行います。
9 :  CPU基板のディップスイッチによりテスト内容を変更します。
10 :  DipSW
11 :  bit3 2 1 0
12 :  0 0 0 0 LEDのテスト LEDが0.5秒間隔で交互に点灯
13 :  0 0 0 1 プッシュスイッチのテスト OFF時：LED0点灯 ON時：LED1点灯
14 :  0 0 1 0 サーボのテスト 0° 右30° 左30°の繰り返し
15 :  0 0 1 1 動作無し
16 :  0 1 0 0 右モータのテスト 正転 プレーキの繰り返し
17 :  0 1 0 1 逆転 プレーキの繰り返し
18 :  0 1 1 0 左モータのテスト 正転 プレーキの繰り返し
19 :  0 1 1 1 逆転 プレーキの繰り返し
20 :
21 :  1 0 0 0 センサテスト センサbit1,0をLED1,0に出力
22 :  1 0 0 1 センサbit3,2をLED1,0に出力
23 :  1 0 1 0 センサbit5,4をLED1,0に出力
24 :  1 0 1 1 センサbit7,6をLED1,0に出力
25 :
26 :  1 1 0 0 直進テスト PWM 50%で前進、2秒後ストップ
27 :  1 1 0 1 直進テスト PWM 50%で前進、5秒後ストップ
28 :  1 1 1 0 直進テスト PWM 100%で前進、2秒後ストップ
29 :  1 1 1 1 直進テスト PWM 100%で前進、5秒後ストップ
30 :  */
31 :
32 :  /*=====*/
33 :  /* インクルード */
34 :  /*=====*/
35 :  #include <machine.h>
36 :  #include "h8_3048.h"
37 :
38 :  /*=====*/
39 :  /* シンボル定義 */
40 :  /*=====*/
41 :
42 :  /* 定数設定 */
43 :  #define TIMER_CYCLE 3071 /* タイマのサイクル 1ms */
44 :  /* /8で使用する場合、 */
45 :  /* /8 = 325.5[ns] */
46 :  /* TIMER_CYCLE = */
47 :  /* 1[ms] / 325.5[ns] */
48 :  /* = 3072 */
49 :  #define PWM_CYCLE 49151 /* PWMのサイクル 16ms */
50 :  /* PWM_CYCLE = */
51 :  /* 16[ms] / 325.5[ns] */
52 :  /* = 49152 */
53 :  #define SERVO_CENTER 5000 /* サーボのセンタ値 */
54 :  #define HANDLE_STEP 26 /* 1°分の値 */
55 :
56 :  /*=====*/
57 :  /* プロトタイプ宣言 */
58 :  /*=====*/
59 :  void init( void );
60 :  unsigned char sensor_inp_test( unsigned char mask );
61 :  unsigned char dipsw_get( void );
62 :  unsigned char pushsw_get( void );
63 :  void led_out( unsigned char led );
64 :  void speed( int accele_l, int accele_r );
65 :  void handle( int angle );
66 :
67 :  /*=====*/
68 :  /* グローバル変数の宣言 */
69 :  /*=====*/
70 :  unsigned long cnt0; /* timer関数用 */
71 :  unsigned long cnt1; /* main内で使用 */
72 :
73 :  /******
74 :  /* メインプログラム */
75 :  /******
76 :  void main( void )
77 :  {
78 :      unsigned char now_sw; /* 現在ディップスイッチ記憶 */
79 :      unsigned char before_sw; /* 前回ディップスイッチ記憶 */
80 :      unsigned char c; /* 作業用 */

```

動作確認マニュアル(マイコンカーキット Ver.4 版)

```

81 :   int          i;                /* 作業用          */
82 :
83 :   /* マイコン機能の初期化 */
84 :   init();                        /* 初期化          */
85 :   set_ccr( 0x00 );               /* 全体割り込み許可 */
86 :
87 :   /* 変数初期化 */
88 :   before_sw = dipsw_get();
89 :   cnt1 = 0;
90 :
91 :   /* マイコンカーの状態初期化 */
92 :   handle( 0 );
93 :   speed( 0, 0 );
94 :   led_out( 0x0 );
95 :
96 :   while( 1 ){
97 :     /* ディップスイッチ読み込み */
98 :     now_sw = dipsw_get();
99 :
100 :    /* 前回のスイッチ値と比較 */
101 :    if( before_sw != now_sw ) {
102 :      /* 不一致なら前回値更新、タイマ値のクリア */
103 :      before_sw = now_sw;
104 :      cnt1 = 0;
105 :    }
106 :
107 :    /* ディップスイッチの値によりテストモードの選択 */
108 :    switch( now_sw ) {
109 :
110 :      /* LEDのテスト LEDが0.5秒間隔で交互に点灯 */
111 :      case 0:
112 :        if( cnt1 < 500 ) {
113 :          led_out( 0x1 );
114 :        } else if( cnt1 < 1000 ) {
115 :          led_out( 0x2 );
116 :        } else {
117 :          cnt1 = 0;
118 :        }
119 :        break;
120 :
121 :      /* プッシュスイッチのテスト OFF時：LED0点灯 ON時：LED1点灯 */
122 :      case 1:
123 :        led_out( pushsw_get() + 1 );
124 :        break;
125 :
126 :      /* サーボのテスト 0° 右30° 左30°の繰り返し */
127 :      case 2:
128 :        if( cnt1 < 1000 ) {
129 :          handle( 0 );
130 :        } else if( cnt1 < 2000 ) {
131 :          handle( 30 );
132 :        } else if( cnt1 < 3000 ) {
133 :          handle( -30 );
134 :        } else {
135 :          cnt1 = 0;
136 :        }
137 :        break;
138 :
139 :      /* 何もしない */
140 :      case 3:
141 :        break;
142 :
143 :      /* 右モータのテスト 正転 ブレーキの繰り返し */
144 :      case 4:
145 :        if( cnt1 < 1000 ) {
146 :          speed( 0, 100 );
147 :        } else if( cnt1 < 2000 ) {
148 :          speed( 0, 0 );
149 :        } else {
150 :          cnt1 = 0;
151 :        }
152 :        break;
153 :
154 :      /* 右モータのテスト 逆転 ブレーキの繰り返し */
155 :      case 5:
156 :        if( cnt1 < 1000 ) {
157 :          speed( 0, -100 );
158 :        } else if( cnt1 < 2000 ) {
159 :          speed( 0, 0 );
160 :        } else {
161 :          cnt1 = 0;
162 :        }
163 :        break;
164 :
165 :      /* 左モータのテスト 正転 ブレーキの繰り返し */
166 :      case 6:
167 :        if( cnt1 < 1000 ) {
168 :          speed( 100, 0 );
169 :        } else if( cnt1 < 2000 ) {
170 :          speed( 0, 0 );
171 :        } else {

```

動作確認マニュアル(マイコンカーキット Ver.4 版)

```

172 :         cnt1 = 0;
173 :     }
174 :     break;
175 :
176 :     /* 左モータのテスト 逆転 プレーキの繰り返し */
177 :     case 7:
178 :         if( cnt1 < 1000 ) {
179 :             speed( -100, 0 );
180 :         } else if( cnt1 < 2000 ) {
181 :             speed( 0, 0 );
182 :         } else {
183 :             cnt1 = 0;
184 :         }
185 :         break;
186 :
187 :     /* センサテスト センサbit1,0をLED1,0に出力 */
188 :     case 8:
189 :         c = sensor_inp_test( 0x03 );
190 :         led_out( c );
191 :         break;
192 :
193 :     /* センサテスト センサbit3,2をLED1,0に出力 */
194 :     case 9:
195 :         c = sensor_inp_test( 0x0c );
196 :         c = c >> 2;
197 :         led_out( c );
198 :         break;
199 :
200 :     /* センサテスト センサbit5,4をLED1,0に出力 */
201 :     case 10:
202 :         c = sensor_inp_test( 0x30 );
203 :         c = c >> 4;
204 :         led_out( c );
205 :         break;
206 :
207 :     /* センサテスト センサbit7,6をLED1,0に出力 */
208 :     case 11:
209 :         c = sensor_inp_test( 0xc0 );
210 :         c = c >> 6;
211 :         led_out( c );
212 :         break;
213 :
214 :     /* 直進テスト PWM 50%で前進、 2秒後ストップ */
215 :     case 12:
216 :         if( cnt1 < 2000 ) {
217 :             speed( 0, 0 );
218 :         } else if( cnt1 < 4000 ) {
219 :             speed( 50, 50 );
220 :         } else {
221 :             speed( 0, 0 );
222 :         }
223 :         break;
224 :
225 :     /* 直進テスト PWM 50%で前進、 5秒後ストップ */
226 :     case 13:
227 :         if( cnt1 < 2000 ) {
228 :             speed( 0, 0 );
229 :         } else if( cnt1 < 7000 ) {
230 :             speed( 50, 50 );
231 :         } else {
232 :             speed( 0, 0 );
233 :         }
234 :         break;
235 :
236 :     /* 直進テスト PWM 100%で前進、 2秒後ストップ */
237 :     case 14:
238 :         if( cnt1 < 2000 ) {
239 :             speed( 0, 0 );
240 :         } else if( cnt1 < 4000 ) {
241 :             speed( 100, 100 );
242 :         } else {
243 :             speed( 0, 0 );
244 :         }
245 :         break;
246 :
247 :     /* 直進テスト PWM 100%で前進、 5秒後ストップ */
248 :     case 15:
249 :         if( cnt1 < 2000 ) {
250 :             speed( 0, 0 );
251 :         } else if( cnt1 < 7000 ) {
252 :             speed( 100, 100 );
253 :         } else {
254 :             speed( 0, 0 );
255 :         }
256 :         break;
257 :
258 :     /* どれも無いなら */
259 :     default:
260 :         break;
261 : }
262 : }

```

動作確認マニュアル(マイコンカーキット Ver.4 版)

```

263 : }
264 :
265 : /******
266 : /* H8/3048F内蔵モジュール 初期化 */
267 : /******
268 : void init( void )
269 : {
270 :     /* ポートの入出力設定 */
271 :     P1DDR = 0xff;
272 :     P2DDR = 0xff;
273 :     P3DDR = 0xff;
274 :     P4DDR = 0xff;
275 :     P5DDR = 0xff;
276 :     P6DDR = 0xf0;           /* CPU基板上のDIP SW */
277 :     P8DDR = 0xff;
278 :     P9DDR = 0xf7;         /* 通信ポート */
279 :     PADDR = 0xff;
280 :     PBDR = 0xc0;
281 :     PBDDR = 0xfe;         /* モータドライブ基板Vol.3 */
282 :     /* センサ基板のP7は、入力専用なので入出力設定はありません */
283 :
284 :     /* ITU0 1msごとの割り込み */
285 :     ITU0_TCR = 0x23;
286 :     ITU0_GRA = TIMER_CYCLE;
287 :     ITU0_IER = 0x01;
288 :
289 :     /* ITU3,4 リセット同期PWMモード 左右モータ、サーボ用 */
290 :     ITU3_TCR = 0x23;
291 :     ITU_FCR = 0x3e;
292 :     ITU3_GRA = PWM_CYCLE;           /* 周期の設定 */
293 :     ITU3_GRB = ITU3_BRB = 0;       /* 左モータのPWM設定 */
294 :     ITU4_GRA = ITU4_BRA = 0;       /* 右モータのPWM設定 */
295 :     ITU4_GRB = ITU4_BRB = SERVO_CENTER; /* サーボのPWM設定 */
296 :     ITU_TOER = 0x38;
297 :
298 :     /* ITUのカウントスタート */
299 :     ITU_STR = 0x09;
300 : }
301 :
302 : /******
303 : /* ITU0 割り込み処理 */
304 : /******
305 : #pragma interrupt( interrupt_timer0 )
306 : void interrupt_timer0( void )
307 : {
308 :     ITU0_TSR &= 0xfe;           /* フラグクリア */
309 :     cnt0++;
310 :     cnt1++;
311 : }
312 :
313 : /******
314 : /* センサ状態検出(テストモード用) */
315 : /* 引数 マスク値 */
316 : /* 戻り値 センサ値 */
317 : /******
318 : unsigned char sensor_inp_test( unsigned char mask )
319 : {
320 :     unsigned char sensor;
321 :
322 :     sensor = P7DR;
323 :     sensor &= mask;
324 :
325 :     return sensor;
326 : }
327 :
328 : /******
329 : /* ディップスイッチ値読み込み */
330 : /* 戻り値 スイッチ値 0~15 */
331 : /******
332 : unsigned char dipsw_get( void )
333 : {
334 :     unsigned char sw;
335 :
336 :     sw = P6DR;           /* ディップスイッチ読み込み */
337 :     sw &= 0x0f;
338 :
339 :     return sw;
340 : }
341 :
342 : /******
343 : /* プッシュスイッチ値読み込み */
344 : /* 戻り値 スイッチ値 ON:1 OFF:0 */
345 : /******
346 : unsigned char pushsw_get( void )
347 : {
348 :     unsigned char sw;
349 :
350 :     sw = PBDR;           /* スイッチのあるポート読み込み */
351 :     sw &= 0x01;
352 :
353 :     return sw;

```

## 動作確認マニュアル(マイコンカーキット Ver.4 版)

```

354 : }
355 :
356 : /******
357 : /* LED制御
358 : /* 引数 スイッチ値 LED0:bit0 LED1:bit1 "0":消灯 "1":点灯
359 : /* 例)0x3 LED1:ON LED0:ON 0x2 LED1:ON LED0:OFF
360 : /******
361 : void led_out( unsigned char led )
362 : {
363 :     unsigned char data;
364 :
365 :     led = led;
366 :     led <<= 6;
367 :     data = PBDR & 0x3f;
368 :     PBDR = data | led;
369 : }
370 :
371 : /******
372 : /* 速度制御
373 : /* 引数 左モータ:-100~100 , 右モータ:-100~100
374 : /*      0で停止、100で正転100%、-100で逆転100%
375 : /******
376 : void speed( int accele_l, int accele_r )
377 : {
378 :     unsigned char sw_data;
379 :     unsigned long speed_max;
380 :
381 :     sw_data = dipsw_get() + 5; /* デイブスイッチ読み込み */
382 :     speed_max = (unsigned long)(PWM_CYCLE-1) * sw_data / 20;
383 :
384 :     /* 左モータ */
385 :     if( accele_l >= 0 ) {
386 :         PBDR &= 0xfb;
387 :         ITU3_BRB = speed_max * accele_l / 100;
388 :     } else {
389 :         PBDR |= 0x04;
390 :         accele_l = -accele_l;
391 :         ITU3_BRB = speed_max * accele_l / 100;
392 :     }
393 :
394 :     /* 右モータ */
395 :     if( accele_r >= 0 ) {
396 :         PBDR &= 0xf7;
397 :         ITU4_BRA = speed_max * accele_r / 100;
398 :     } else {
399 :         PBDR |= 0x08;
400 :         accele_r = -accele_r;
401 :         ITU4_BRA = speed_max * accele_r / 100;
402 :     }
403 : }
404 :
405 : /******
406 : /* サーボハンドル操作
407 : /* 引数 サーボ操作角度:-90~90
408 : /*      -90で左へ90度、0でまっすぐ、90で右へ90度回転
409 : /******
410 : void handle( int angle )
411 : {
412 :     ITU4_BRB = SERVO_CENTER - angle * HANDLE_STEP;
413 : }
414 :
415 : /******
416 : /* end of file
417 : /******

```

## 5.2 「kit07teststart.src」のプログラム内容

```

1 : ;=====
2 : ; 定義
3 : ;=====
4 : RESERVE: .EQU H'FFFFFF ; 未使用領域のアドレス
5 :
6 : ;=====
7 : ; 外部参照
8 : ;=====
9 : .IMPORT _main
10 : .IMPORT _interrupt_timer0
11 :
12 : ;=====
13 : ; ベクタセクション
14 : ;=====
15 : .SECTION V
16 : .DATA.L RESET_START ; 0 H'000000 レット
17 : .DATA.L RESERVE ; 1 H'000004 システム予約
18 : .DATA.L RESERVE ; 2 H'000008 システム予約
19 : .DATA.L RESERVE ; 3 H'00000c システム予約

```

動作確認マニュアル(マイコンカーキット Ver.4 版)

```

20 :      .DATA.L RESERVE      : 4 H'000010 システム予約
21 :      .DATA.L RESERVE      : 5 H'000014 システム予約
22 :      .DATA.L RESERVE      : 6 H'000018 システム予約
23 :      .DATA.L RESERVE      : 7 H'00001c 外部割り込み NMI
24 :      .DATA.L RESERVE      : 8 H'000020 トラップ 命令
25 :      .DATA.L RESERVE      : 9 H'000024 トラップ 命令
26 :      .DATA.L RESERVE      : 10 H'000028 トラップ 命令
27 :      .DATA.L RESERVE      : 11 H'00002c トラップ 命令
28 :      .DATA.L RESERVE      : 12 H'000030 外部割り込み IRQ0
29 :      .DATA.L RESERVE      : 13 H'000034 外部割り込み IRQ1
30 :      .DATA.L RESERVE      : 14 H'000038 外部割り込み IRQ2
31 :      .DATA.L RESERVE      : 15 H'00003c 外部割り込み IRQ3
32 :      .DATA.L RESERVE      : 16 H'000040 外部割り込み IRQ4
33 :      .DATA.L RESERVE      : 17 H'000044 外部割り込み IRQ5
34 :      .DATA.L RESERVE      : 18 H'000048 システム予約
35 :      .DATA.L RESERVE      : 19 H'00004c システム予約
36 :      .DATA.L RESERVE      : 20 H'000050 WDT MOV1
37 :      .DATA.L RESERVE      : 21 H'000054 REF CMI
38 :      .DATA.L RESERVE      : 22 H'000058 システム予約
39 :      .DATA.L RESERVE      : 23 H'00005c システム予約
40 :      .DATA.L interrupt_timer0 : 24 h'000060 ITU0 IMIA0
41 :      .DATA.L RESERVE      : 25 H'000064 ITU0 IMIB0
42 :      .DATA.L RESERVE      : 26 H'000068 ITU0 OV10
43 :      .DATA.L RESERVE      : 27 H'00006c システム予約
44 :      .DATA.L RESERVE      : 28 H'000070 ITU1 IMIA1
45 :      .DATA.L RESERVE      : 29 H'000074 ITU1 IMIB1
46 :      .DATA.L RESERVE      : 30 H'000078 ITU1 OV11
47 :      .DATA.L RESERVE      : 31 H'00007c システム予約
48 :      .DATA.L RESERVE      : 32 H'000080 ITU2 IMIA2
49 :      .DATA.L RESERVE      : 33 H'000084 ITU2 IMIB2
50 :      .DATA.L RESERVE      : 34 H'000088 ITU2 OV12
51 :      .DATA.L RESERVE      : 35 H'00008c システム予約
52 :      .DATA.L RESERVE      : 36 H'000090 ITU3 IMIA3
53 :      .DATA.L RESERVE      : 37 H'000094 ITU3 IMIB3
54 :      .DATA.L RESERVE      : 38 H'000098 ITU3 OV13
55 :      .DATA.L RESERVE      : 39 H'00009c システム予約
56 :      .DATA.L RESERVE      : 40 H'0000a0 ITU4 IMIA4
57 :      .DATA.L RESERVE      : 41 H'0000a4 ITU4 IMIB4
58 :      .DATA.L RESERVE      : 42 H'0000a8 ITU4 OV14
59 :      .DATA.L RESERVE      : 43 H'0000ac システム予約
60 :      .DATA.L RESERVE      : 44 H'0000b0 DMAC DEND0A
61 :      .DATA.L RESERVE      : 45 H'0000b4 DMAC DEND0B
62 :      .DATA.L RESERVE      : 46 H'0000b8 DMAC DEND1A
63 :      .DATA.L RESERVE      : 47 H'0000bc DMCA DEND1B
64 :      .DATA.L RESERVE      : 48 H'0000c0 システム予約
65 :      .DATA.L RESERVE      : 49 H'0000c4 システム予約
66 :      .DATA.L RESERVE      : 50 H'0000c8 システム予約
67 :      .DATA.L RESERVE      : 51 H'0000cc システム予約
68 :      .DATA.L RESERVE      : 52 H'0000d0 SC10 ER10
69 :      .DATA.L RESERVE      : 53 H'0000d4 SC10 RX10
70 :      .DATA.L RESERVE      : 54 H'0000d8 SC10 TX10
71 :      .DATA.L RESERVE      : 55 H'0000dc SC10 TE10
72 :      .DATA.L RESERVE      : 56 H'0000e0 SC11 ER11
73 :      .DATA.L RESERVE      : 57 H'0000e4 SC11 RX11
74 :      .DATA.L RESERVE      : 58 H'0000e8 SC11 TX11
75 :      .DATA.L RESERVE      : 59 H'0000ec SC11 TE11
76 :      .DATA.L RESERVE      : 60 H'0000f0 A/D ADI
77 :
78 : ;=====
79 : ; スタートアッププログラム
80 : ;=====
81 :      .SECTION P
82 : RESET_START:
83 :     MOV.L #H'FFF10,ER7      ; スタックの設定
84 :     JSR   @_main           ; C言語のmain()関数ヘジャンプ
85 : OWARI:
86 :     BRA  OWARI
87 :
88 :     .END

```