

マイコンカーキットVer.5.1 動作確認マニュアル R8C/38A版

本マニュアルは、
・モータドライブ基板 Ver.5
・センサ基板 Ver.5
に対応したマイコンカーキット Ver.5.1 の動作確認手順を説明しています。

第 1.02 版

2015.04.20

ジャパンマイコンカーラリー実行委員会
株式会社日立ドキュメントソリューションズ

注意事項 (rev.6.0J)

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、文書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。
- ・すべての商標および登録商標は、それぞれの所有者に帰属します。

連絡先

株式会社 日立ドキュメントソリューションズ

〒135-0016 東京都江東区東陽六丁目 3 番 2 号 イースト 21 タワー

E-mail:himdx.m-carrally.dd@hitachi.com

目次

1. 概要.....	1
2. プログラムのインストール.....	2
3. マイコンカーキットの構成.....	4
4. 動作確認用プログラムの書き込み.....	5
4.1 ワークスペース「kit12_38a」を開く.....	5
4.2 動作確認用プログラムの書き込み.....	6
5. 動作確認.....	10
5.1 動作確認一覧.....	10
5.2 LED の動作確認.....	11
5.3 プッシュスイッチの動作確認.....	12
5.4 サーボの動作確認.....	13
5.5 右モータの動作確認.....	15
5.6 左モータの動作確認.....	16
5.7 センサ基板の動作確認.....	17
5.8 直進性の確認.....	22
5.9 動作確認終了.....	23
6. プログラムソース.....	24
6.1 「kit12test_38a.c」のプログラム内容.....	24

1. 概要

1. 概要

本マニュアルは、下記マニュアルで製作、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。

- モータドライブ基板 Ver.5 製作マニュアル
- センサ基板 Ver.5 製作マニュアル
- マイコンカーキット Ver.5.1 本体組み立て製作マニュアル

動作確認は、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

パソコンにルネサス統合開発環境(M16C、R8C 版)をインストールします。詳しくは、「ルネサス統合開発環境操作マニュアル(R8C/38A 版)」を参照して、インストールしてください。なお、ルネサス統合開発環境(H8 版)では動作しません。

2. サンプルプログラム(動作確認プログラム)をインストールします

パソコンにワークスペース「kit12_38a」をインストールします。これから行います。

3. 動作確認プログラムを、マイコンカーの RY_R8C38 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確認します

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

マイコンカーキット Ver.5.1 kit12_38a プログラム解説マニュアル(R8C/38A 版)の「7. 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

マイコンカーキット Ver.5.1 kit12_38a プログラム解説マニュアル(R8C/38A 版)を参照しながら、プロジェクト「kit12_38a」の kit12_38a.mot ファイルを RY_R8C38 ボードに書き込み、コースを走らせてみてください。

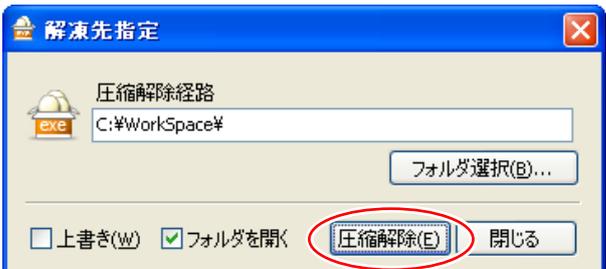
2. プログラムのインストール

2. プログラムのインストール

1		<p>マイコンカーラリーサイト 「http://www.mcr.gr.jp/」の技術情報→ダウンロード内のページへ行きます。</p>
---	---	--

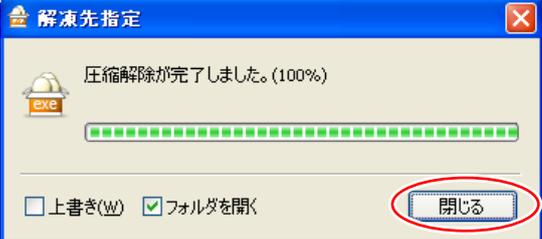
2	<p>免責事項</p> <p>「マニュアル」、「ソフトウェア」は万全な体制で制作されており、通常の使用環境においては正常に動作するように作成されていますが、万が一「マニュアル」、「ソフトウェア」による損失・損害が発生した時には、『ジャパンマイコンカーラリー実行委員会』はいかなる場合も責任を負いません。個人の免責が取れる範囲内であらかじめ了承の上でご使用くださるようお願いをいたします。</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>対象マイコン</th> <th>内容</th> <th>更新日</th> </tr> </thead> <tbody> <tr> <td>R8C/38A</td> <td>R8C/38Aマイコン(RY_R8C38ボード)に関する資料</td> <td>2013.06.03 NEW!!</td> </tr> <tr> <td>H8/3048F-ONE</td> <td>H8/3048F-ONEマイコン(RY3048Foneボード)用のサンプルプログラム、書き込みソフト</td> <td>2010.10.07</td> </tr> <tr> <td>H8/3048F</td> <td>H8/3048F-ONEマイコン(RY3048Foneボード)に</td> <td>2009.09.25</td> </tr> </tbody> </table>	対象マイコン	内容	更新日	R8C/38A	R8C/38Aマイコン(RY_R8C38ボード)に関する資料	2013.06.03 NEW!!	H8/3048F-ONE	H8/3048F-ONEマイコン(RY3048Foneボード)用のサンプルプログラム、書き込みソフト	2010.10.07	H8/3048F	H8/3048F-ONEマイコン(RY3048Foneボード)に	2009.09.25	<p>「R8C/38A マイコン(RY_R8C38 ボード)に関する資料」をクリックします。</p>
対象マイコン	内容	更新日												
R8C/38A	R8C/38Aマイコン(RY_R8C38ボード)に関する資料	2013.06.03 NEW!!												
H8/3048F-ONE	H8/3048F-ONEマイコン(RY3048Foneボード)用のサンプルプログラム、書き込みソフト	2010.10.07												
H8/3048F	H8/3048F-ONEマイコン(RY3048Foneボード)に	2009.09.25												

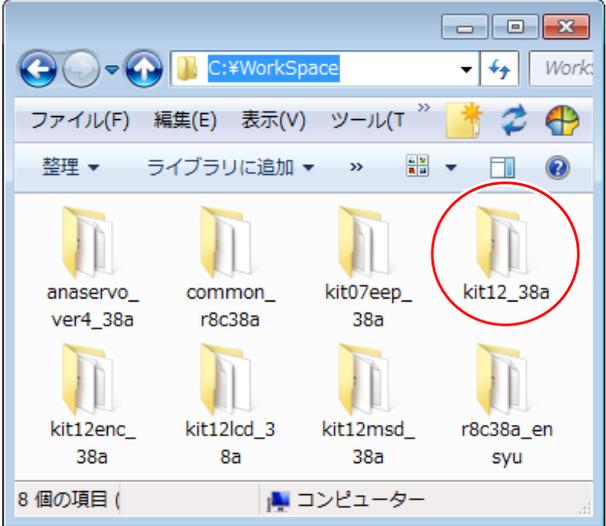
3	<p>■マイコンカーキットに関する資料</p> <p>マイコンカーキットVer.5.1に関する資料は、下記からダウンロードしてください。</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>本体製作</th> <th>センサ基板</th> <th>モータドライブ基板</th> <th>動作確認</th> <th>プログラム解説</th> <th>プログラム</th> </tr> </thead> <tbody> <tr> <td>マイコンカーキットVer.5.1本体組み立て製作マニュアル 第1.01版 2012.09.13</td> <td>センサ基板Ver.5 製作マニュアル 第1.00版 2012.06.18</td> <td>モータドライブ基板Ver.5 製作マニュアル 第1.00版 2012.06.18 <small>※「LM350追加セット」、「フリー追加セット」についても本マニュアルをご覧ください。</small></td> <td>マイコンカーキットVer.5.1動作確認マニュアル 第1.00版 2012.06.20</td> <td>kit12_38aプログラム解説 第1.01版 2012.10.18</td> <td>kit12_38a.exe 2013.04.01</td> </tr> </tbody> </table> <p>マイコンカーキットVer.5に関する資料は、下記</p> <p style="text-align: center; border: 1px solid black; padding: 5px;">kit12_38a.exe</p>	本体製作	センサ基板	モータドライブ基板	動作確認	プログラム解説	プログラム	マイコンカーキットVer.5.1本体組み立て製作マニュアル 第1.01版 2012.09.13	センサ基板Ver.5 製作マニュアル 第1.00版 2012.06.18	モータドライブ基板Ver.5 製作マニュアル 第1.00版 2012.06.18 <small>※「LM350追加セット」、「フリー追加セット」についても本マニュアルをご覧ください。</small>	マイコンカーキットVer.5.1動作確認マニュアル 第1.00版 2012.06.20	kit12_38aプログラム解説 第1.01版 2012.10.18	kit12_38a.exe 2013.04.01	<p>マイコンカーキット Ver.5.1 に関する資料の「kit12_38a.exe」をダウンロードします。</p>
本体製作	センサ基板	モータドライブ基板	動作確認	プログラム解説	プログラム									
マイコンカーキットVer.5.1本体組み立て製作マニュアル 第1.01版 2012.09.13	センサ基板Ver.5 製作マニュアル 第1.00版 2012.06.18	モータドライブ基板Ver.5 製作マニュアル 第1.00版 2012.06.18 <small>※「LM350追加セット」、「フリー追加セット」についても本マニュアルをご覧ください。</small>	マイコンカーキットVer.5.1動作確認マニュアル 第1.00版 2012.06.20	kit12_38aプログラム解説 第1.01版 2012.10.18	kit12_38a.exe 2013.04.01									

4		<p>ダウンロードしたファイルを実行します。 「圧縮解除」をクリックします。</p>
---	---	---

2. プログラムのインストール

5		<p>ファイルの上書き確認の画面が出てきた場合、「全てのファイルに作用」のチェックを付けて、「はい」をクリックします。</p> <p>※上書きしたくない場合は、元々あるファイルを保存してから実行してください。</p>
---	---	--

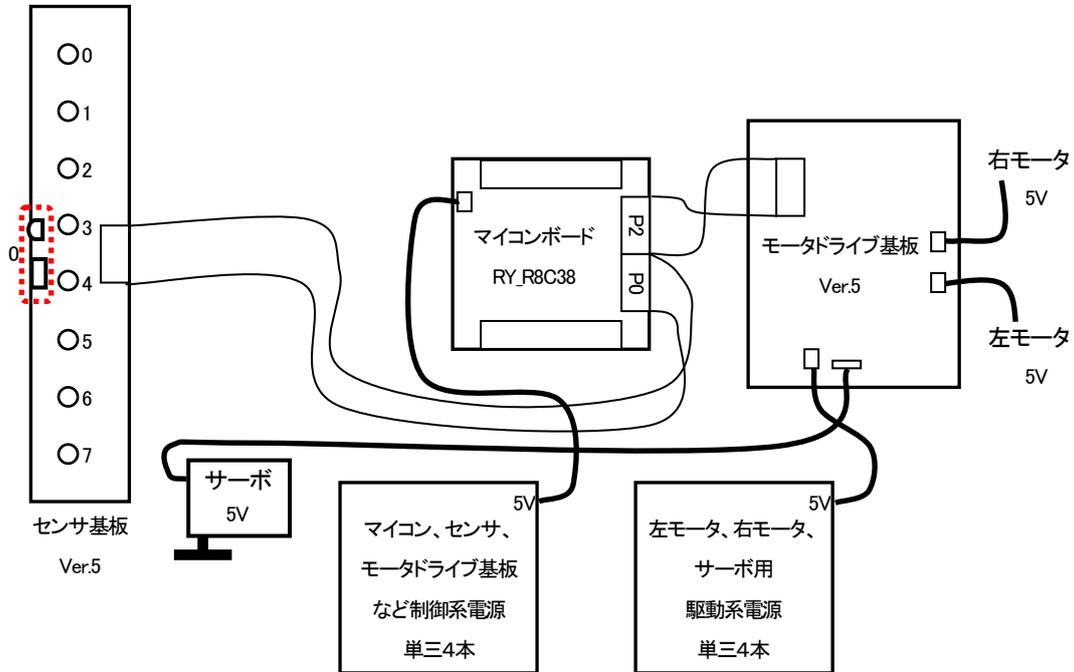
6		<p>インストールが終わりました。「OK」をクリックします。</p>
---	---	------------------------------------

7		<p>「C:\¥Workspace」のフォルダが自動で開きます。</p> <p>「kit12_38a」フォルダの内容が、今回の動作確認用のプログラムが入っているフォルダです。</p>
---	--	---

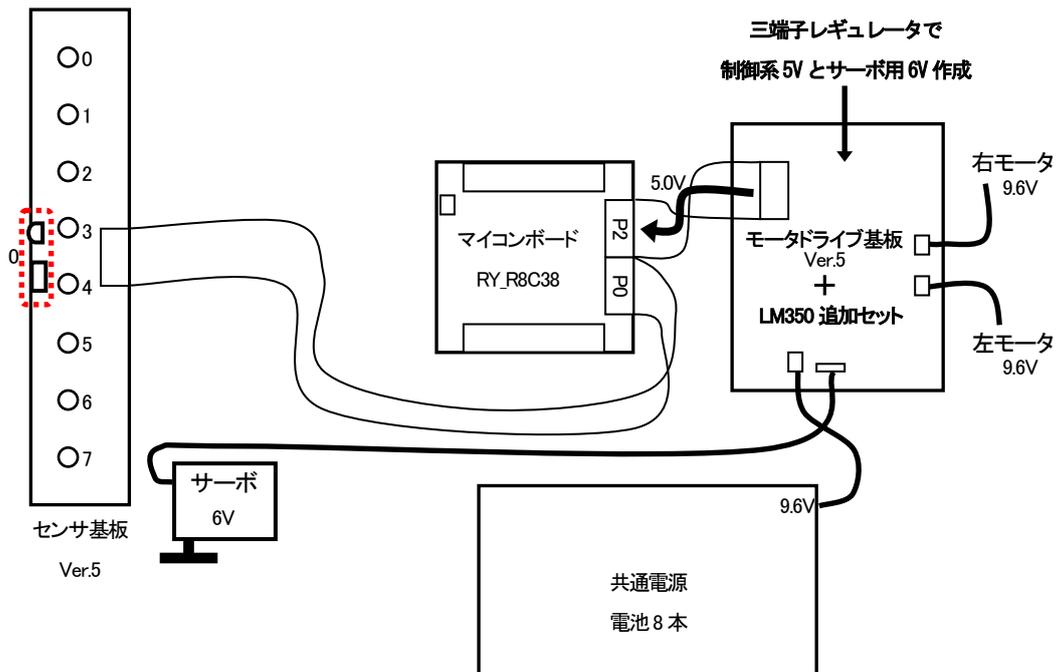
3. マイコンカーキットの構成

3. マイコンカーキットの構成

本マニュアルでは、マイコンカーキット Ver.5.1 の構成のマイコンカーの動作確認を行うことができます。構成を、下記に示します。

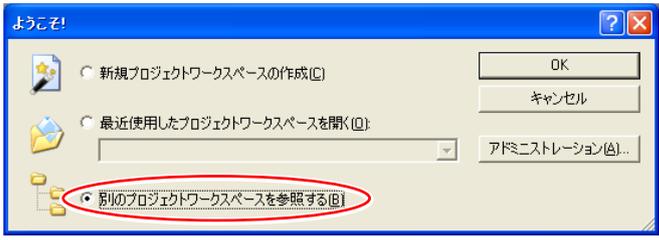
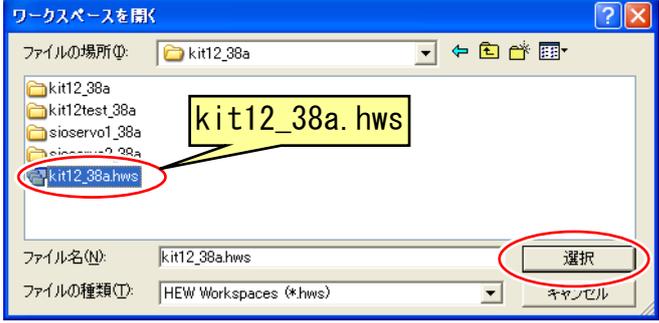
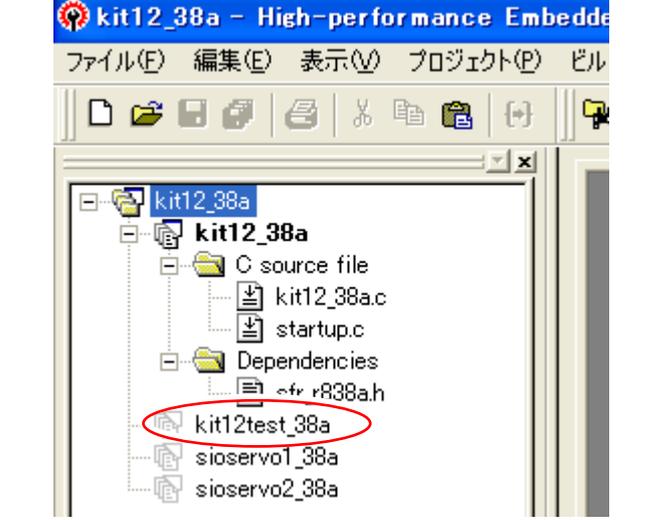


モータドライブ基板に「LM350 追加セット」を追加した構成のマイコンカーも、動作確認することができます。追加したときの構成を、下記に示します。



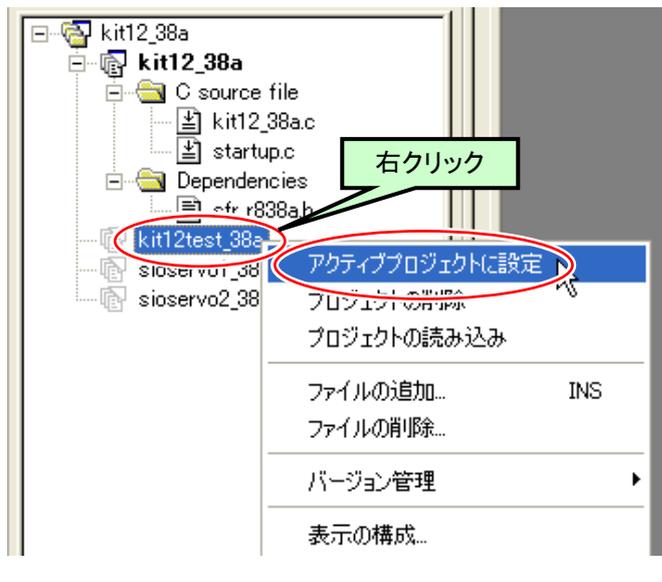
4. 動作確認用プログラムの書き込み

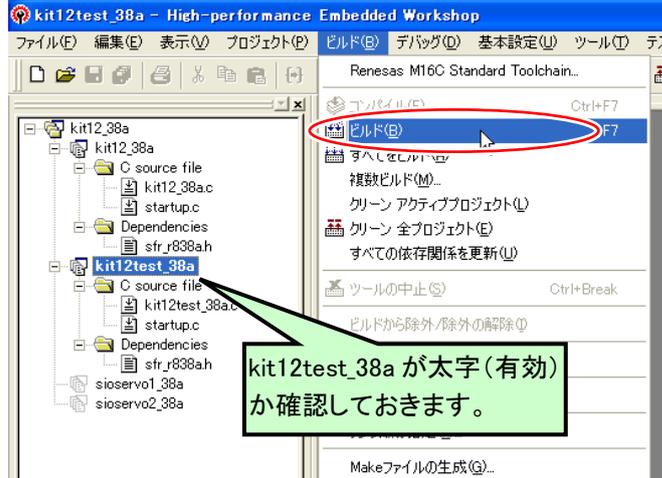
4.1 ワースペース「kit12_38a」を開く

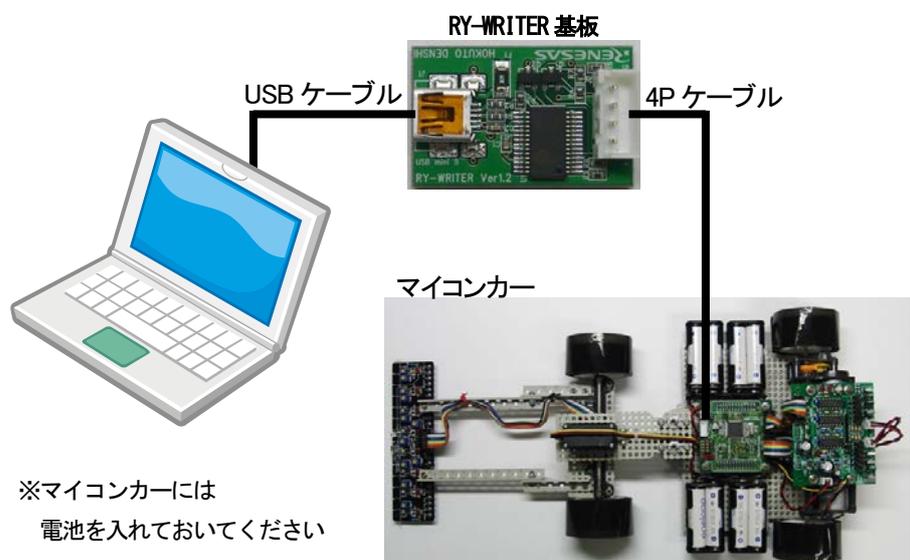
1		ルネサス統合開発環境を実行します。
2		「別のプロジェクトワークスペースを参照する」を選択します。
3		Cドライブ → Workspace → kit12_38a の「kit12_38a.hws」を選択します。
4		ワークスペース「kit12_38a」が開きます。このワークスペースには、4つのプロジェクトがあります。 <ul style="list-style-type: none">●kit12_38a マイコンカー走行プログラムです。●kit12test_38a 今回使うプロジェクトで、モータドライブ基板、センサ基板が正しく動作するか確認します。●sioservo1_38a サーボのセンタを調整するプログラムです。●sioservo2_38a サーボの最大切れ角を見つけるためのプログラムです。

4. 動作確認用プログラムの書き込み

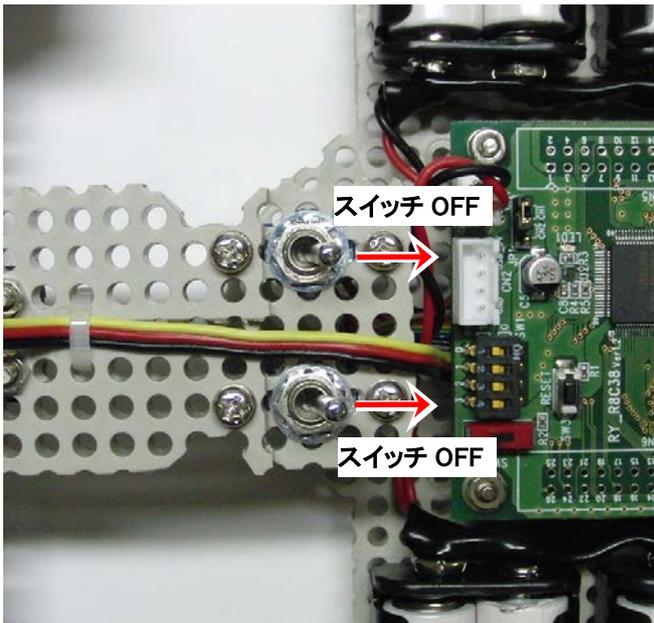
4.2 動作確認用プログラムの書き込み

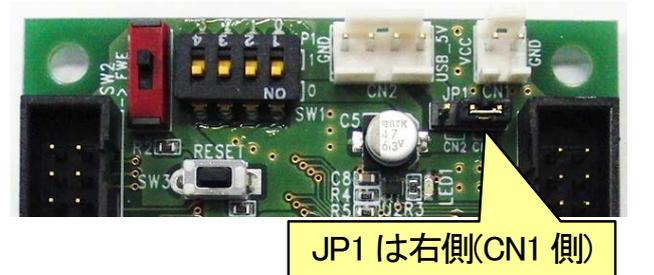
1		<p>「kit12test_38a」をアクティブプロジェクトに設定します。</p> <p>「kit12test_38a」上で右クリックします。</p> <p>「アクティブプロジェクトに設定」を選択します。</p> <p>「kit12test_38a」が太字になります。</p>
---	---	--

2		<p>「ビルド→ビルド」で MOT ファイルを作ります。</p>
---	--	----------------------------------

3	 <p>※マイコンカーには電池を入れておいてください</p>	<p>パソコンとマイコンカー (RY_R8C38 ボード) の間に USB 信号を TTL レベル信号に変換する基板を入れます。</p> <p>左図は、RY-WRITER 基板をパソコンとマイコンカーキット Ver.5.1 の間に入れた場合の結線です。詳しくは、「マイコン実習マニュアル(R8C/38A 版)」を参照してください。</p>
---	--	---

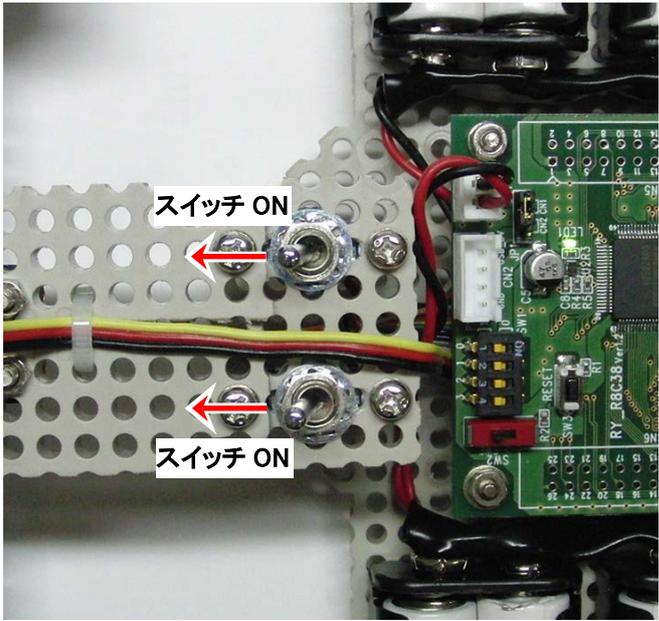
4. 動作確認用プログラムの書き込み

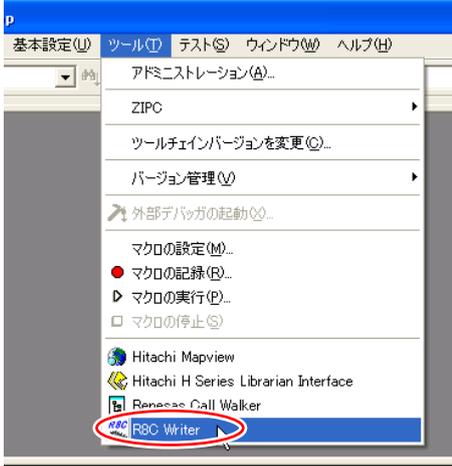
4		マイコンカーの電源スイッチは、2 つとも切っておきます。
---	---	------------------------------

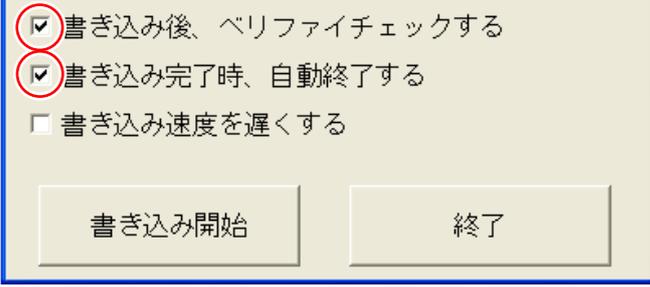
5		RY_R8C38 ボードの JP1 は、写真のように右側をショート(CN1 側)してください。 ※CN1 から電源供給する設定です。CN2 にすると、パソコンの USB 電源から電源供給する設定になります。パソコンの USB 電源は、電流を取りすぎると壊れるおそれがあるので使わないでください。
---	--	--

6		SW2 を FWE 側(外側)にします。 SW2 を外側の状態で RY_R8C38 ボードの電源を入れると、マイコンはプログラム書き込みモードになります。 ※SW2 は、必ず RY_R8C38 ボードの電源が OFF の状態で操作してください。
---	---	--

4. 動作確認用プログラムの書き込み

7		<p>マイコンカーの電源スイッチを、2つとも ON にします。</p>
---	---	-------------------------------------

8		<p>「ツール→R8C Writer」で R8C Writer を立ち上げます。</p> <p>※「R8C Writer」が無い場合は、「ルネサス統合開発環境操作マニュアル(R8C/38A 版)」を参照して、登録してください。</p>
---	--	---

9		<p>R8C Writer のチェックボックスの設定を確認します。</p> <p>2 カ所とも、チェックは ON(☑)にしておきます。</p>
---	---	---

4. 動作確認用プログラムの書き込み

10		<p>「kit12test_38a.mot」を書き込みます。 書き込みが正常に終わったら、R8C Writer は自動で終了します。もしエラー画面が出てきた場合は、ケーブルの接続などを確認して再度書き込んでください。</p> <p>※Tera Term が立ち上がっていると、通信ポートを使うので書き込みができません。 Tera Term を終了させて、再度書き込んでください。</p>
----	--	---

11		<p>マイコンカーの電源スイッチは、2 つとも切っておきます。</p>
----	--	-------------------------------------

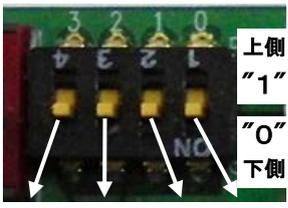
12		<p>SW2 を内側にします。 SW2 を内側の状態で RY_R8C38 ボードの電源を入れると、先ほど書き込んだプログラムを実行します。</p> <p>※SW2 は、必ず RY_R8C38 ボードの電源が OFF の状態で操作してください。</p>
----	--	---

5. 動作確認

5. 動作確認

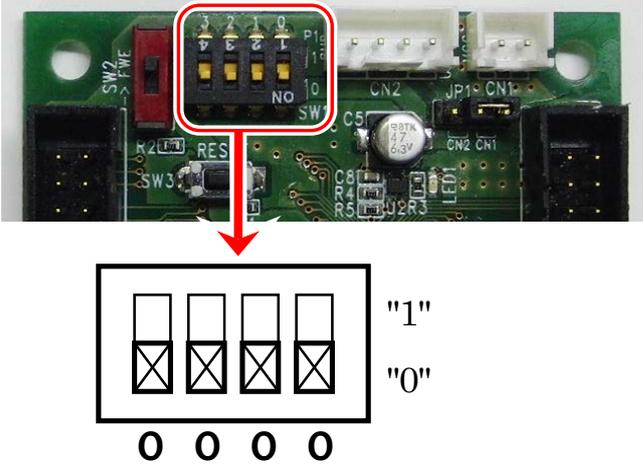
5.1 動作確認一覧

RY_R8C38 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動作確認するか選択し、動作確認を行います。

				内容
P1_3	P1_2	P1_1	P1_0	
0	0	0	0	LED の動作確認をします。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動作確認をします。 スイッチ OFF で LED3 が点灯、スイッチ ON で LED2 が点灯します。
0	0	1	0	サーボの動作確認をします。 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	右モータの動作確認をします。 「正転→ブレーキ」動作を繰り返します。
0	1	0	1	右モータの動作確認をします。 「逆転→ブレーキ」動作を繰り返します。
0	1	1	0	左モータの動作確認をします。 「正転→ブレーキ」動作を繰り返します。
0	1	1	1	左モータの動作確認をします。 「逆転→ブレーキ」動作を繰り返します。
1	0	0	0	センサ基板の bit1、bit0 の動作確認をします。 センサ bit1、bit0 の状態を LED2、LED3 に出力します。 ※bit0 はスタートバー検出センサと兼用です。
1	0	0	1	センサ基板の bit3、bit2 の動作確認をします。 センサ bit3、bit2 の状態を LED2、LED3 に出力します。
1	0	1	0	センサ基板の bit5、bit4 の動作確認をします。 センサ bit5、bit4 の状態を LED2、LED3 に出力します。
1	0	1	1	センサ基板の bit7、bit6 の動作確認をします。 センサ bit7、bit6 の状態を LED2、LED3 に出力します。
1	1	0	0	まっすぐに走るか確認します。 PWM50%で前進、2 秒後に停止します。
1	1	0	1	まっすぐに走るか確認します。 PWM50%で前進、5 秒後に停止します。
1	1	1	0	まっすぐに走るか確認します。 PWM100%で前進、2 秒後に停止します。
1	1	1	1	まっすぐに走るか確認します。 PWM100%で前進、5 秒後に停止します。

5. 動作確認

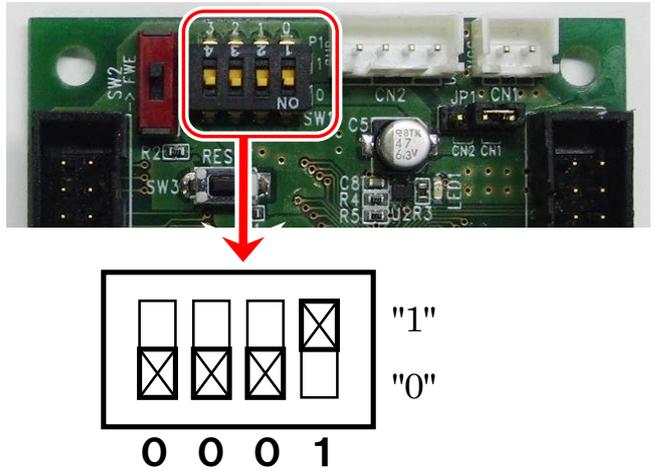
5.2 LED の動作確認

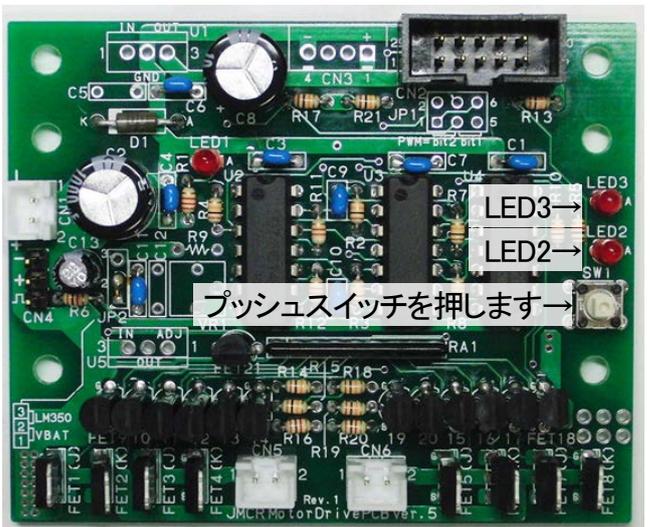
<p>1</p>	 <p>The image shows a microcontroller board with a 4-pin DIP switch highlighted by a red box. Below the board is a diagram of the switch with four positions. The top row of the diagram is labeled "1" and the bottom row is labeled "0". Below the diagram are four "0" characters, indicating that all four pins are in the "0" position.</p>	<p>モータドライブ基板の LED が点灯・消灯するか確認します。 ディップスイッチを”0000”の状態に、マイコンカーの電源スイッチを 2 つとも ON にします。</p>
----------	---	--

<p>2</p>	 <p>The image shows a microcontroller board with a yellow callout box pointing to LED1. The callout box contains the text: "電源 LED です。電源 ON で常に点灯します" (This is the power LED. It is always lit when the power is ON). Labels "LED2" and "LED3" point to other LEDs on the board.</p>	<p>モータドライブ基板の LED2 と LED3 が 0.5 秒毎に交互に点灯します。 動作確認が終わったら電源スイッチ 2 つを OFF にします。</p> <p>LED2、または LED3 が点灯しない場合は、RY_R8C38 ボードとモータドライブ基板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、テスタなどで原因を突き止めてください。</p>
----------	--	---

5. 動作確認

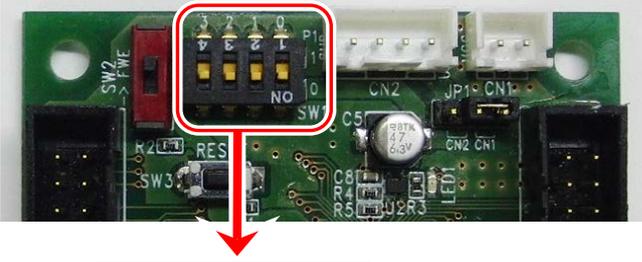
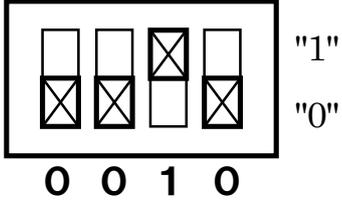
5.3 プッシュスイッチの動作確認

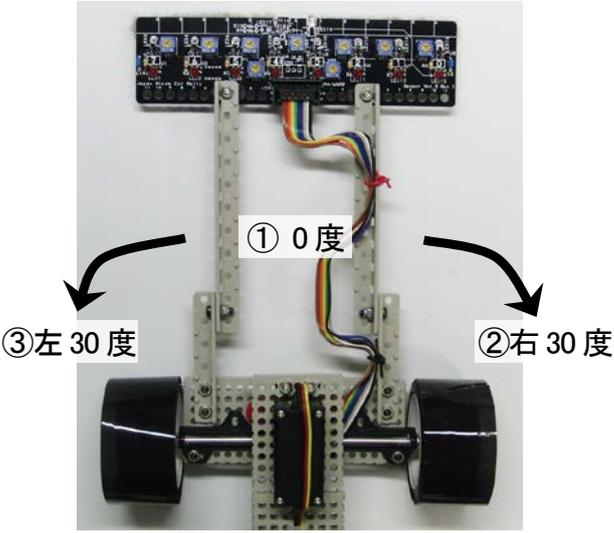
<p>1</p>		<p>モータドライブ基板のプッシュスイッチが反応するか確認します。 ディップスイッチを”0001”の状態に設定し、マイコンカーの電源スイッチを2 つとも ON にします。</p>
----------	---	---

<p>2</p>		<p>モータドライブ基板のプッシュスイッチが押されていない状態で LED3 が点灯、押された状態で LED2 が点灯します。 動作確認が終わったら電源スイッチ 2 つを OFF にします。</p> <p>LED3 のみしか点灯しない場合は、プッシュスイッチまでの回路が半田付け不良、LED2 が付きっぱなしの場合は半田ブリッジなどが考えられます。目視チェック、テスタなどで原因を突き止めてください。</p>
----------	--	---

5. 動作確認

5.4 サーボの動作確認

1		<p>モータドライブ基板に接続したサーボが動作するか確認します。フロント部分が動きますので、マイコンカーの回りには何も置かずに、浮かせた状態にしてください。</p> <p>ディップスイッチを”0010”の状態、マイコンカーの電源スイッチを2 つとも ON にします。</p>
		

2		<p>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。動作確認が終わったら電源スイッチ 2 つを OFF にします。</p> <p>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などが考えられます。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、テスタなどで原因を突き止めてください。</p>
---	--	--

3	<pre> 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 </pre> <pre> /***** /* サーボハンドル操作 /* 引数 サーボ操作角度：-90~90 /* -90で左へ90度、0でまっすぐ、90で右へ90度[/***** void handle(int angle) { /* サーボが左右逆に動く場合は、「-」を「+」に替 trdgrd1 = SERVO_CENTER - angle * HANDLE_STEP; } /***** /* end of file /***** </pre> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-top: 10px;"> 「-」を 「+」にし </div>	<p>キット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、「kit12test_38a.c」の handle 関数内の 425 行の「- (マイナス)」を「+ (プラス)」に変更すれば左右が入れ代わり、動作が「0 度→右 30 度→左 30 度」となります。</p>
---	--	--

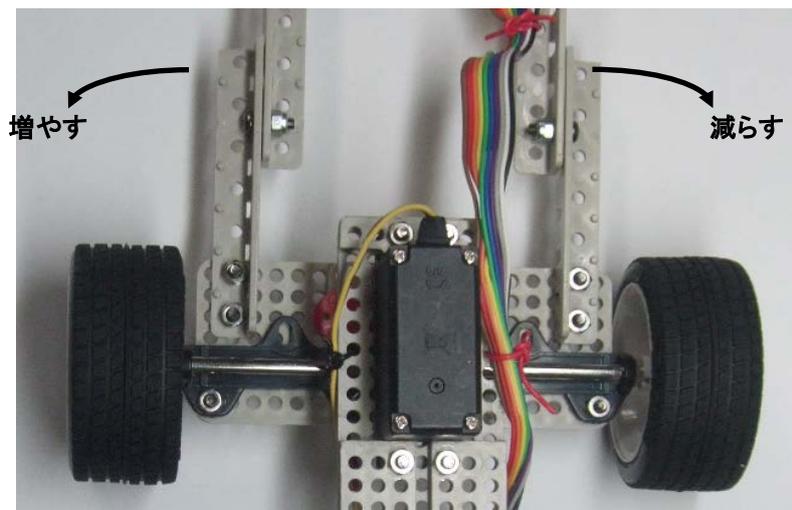
5. 動作確認

※サーボのセンタ調整

マイコンカーの電源を入れたとき、サーボは 0 度になっていなければいけません。しかし、ほとんどの場合は電源を ON にしても 0 度になっていないと思います。これは、サーボのセンタ値(まっすぐにする値)が、サーボによってそれぞれ違うのでマイコンカー 1 台 1 台で調整する必要があるためです。「kit12test_38a.c」の SERVO_CENTER の「3750」という値を変えてまっすぐになるように調整します。この数値は、26 で約 1 度です。

```
46 : #define SERVO_CENTER    3750          /* サーボのセンタ値          */
```

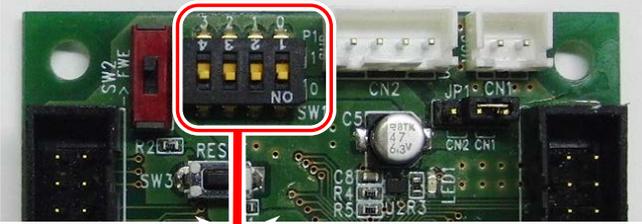
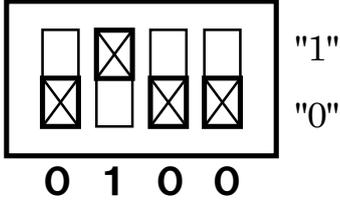
値を増やせば進行方向に向かって左側、減らせば右側に向きます。

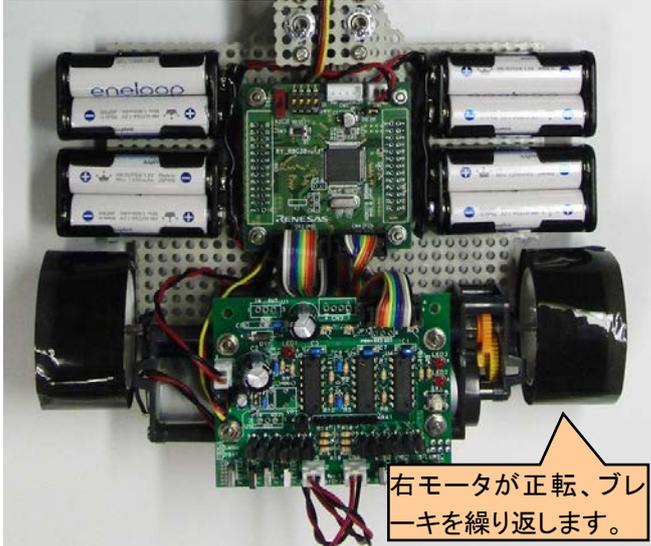


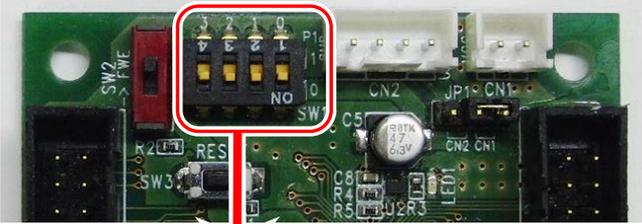
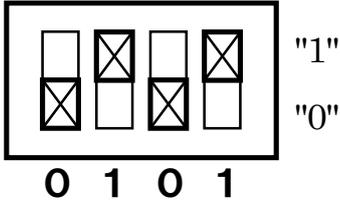
サーボセンタ値の調整は、マイコンカーキット Ver.5.1 kit12_38a プログラム解説マニュアル(R8C/38A 版)の「7. サーボセンタと最大切れ角の調整」を参照しながら行くと便利です。

5. 動作確認

5.5 右モータの動作確認

1		<p>右モータが正転、逆転、停止するか確認します。右モータが回りますので、マイコンカーを持ち上げた状態にしてください。ディップスイッチを”0100”の状態に、マイコンカーの電源スイッチを2つともONにします。</p>
		

2	 <p style="text-align: center;">右モータが正転、ブレーキを繰り返します。</p>	<p>右モータが1秒ごとに「正転→ブレーキ」を繰り返します。動作確認が終わったら電源スイッチ2つをOFFにします。</p> <p>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、テスタなどで原因を突き止めてください。</p> <p>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。</p>
---	--	---

3		<p>ディップスイッチを”0101”の状態に、マイコンカーの電源スイッチを2つともONにします。</p> <p>右モータが1秒ごとに「逆転→ブレーキ」を繰り返します。動作確認が終わったら電源スイッチ2つをOFFにします。</p> <p>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、テスタなどで原因を突き止めてください。</p>
		

5. 動作確認

5.6 左モータの動作確認

1		<p>左モータが正転、逆転、停止するか確認します。左モータが回りますので、タイヤを持ち上げた状態にしてください。</p> <p>ディップスイッチを”0110”の状態、マイコンカーの電源スイッチを2 つとも ON にします。</p>
---	--	---

2		<p>左モータが1秒ごとに「正転→ブレーキ」を繰り返します。</p> <p>動作確認が終わったら電源スイッチ 2 つを OFF にします。</p> <p>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、テスタなどで原因を突き止めてください。</p> <p>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。</p>
---	--	---

3		<p>ディップスイッチを”0111”の状態、マイコンカーの電源スイッチを2 つとも ON にします。</p> <p>左モータが1秒ごとに「逆転→ブレーキ」を繰り返します。</p> <p>動作確認が終わったら電源スイッチ 2 つを OFF にします。</p> <p>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、テスタなどで原因を突き止めてください。</p>
---	--	---

5. 動作確認

5.7 センサ基板の動作確認

1

ディップスイッチを"1000"の状態に、マイコンカーの電源スイッチを2つともONにします。

2

スタートパ検出センサ: 0

感度調整用ボリューム

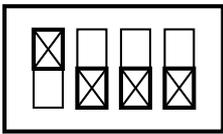
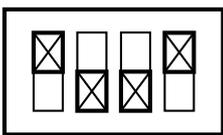
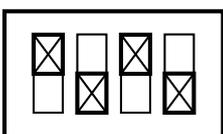
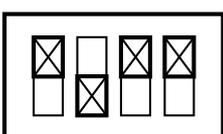
LED3
LED2

写真のように、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に出力されます。センサ基板上のLEDも点灯しますので、同じ反応か確認します。センサの感度はボリュームで調整します。

もしセンサ基板上の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、部品の逆差しなどが考えられます。センサ基板上のLEDは点灯するのにモータドライブ基板上の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、テストなどで原因を突き止めてください。

※スタートパ検出センサは、センサ0のLED(LED15)に出力されます。ディップスイッチ"1000"のときに(今回)、スタートパ検出センサの確認を行ってください。

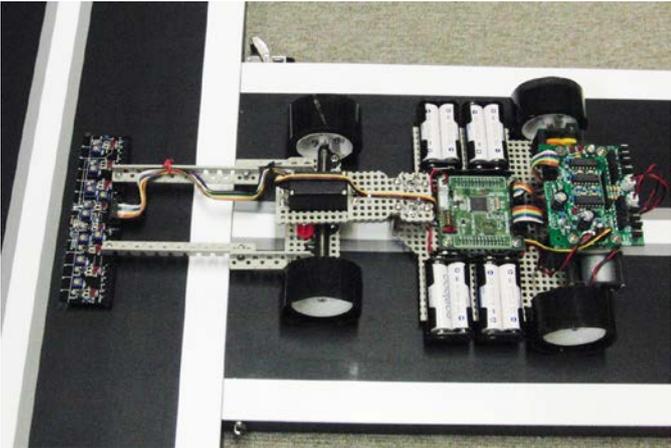
5. 動作確認

3	ディップスイッチ  "1" "0" 1 0 0 0	モータドライブ基板の LED2 に出力される センサ番号 1	モータドライブ基板の LED3 に出力される センサ番号 0 スタートバー検出センサの 反応でも光ります	同じように左の表の ようにディップスイッ チを切り替えて、セ ンサ基板のセンサ 7 ～0 の合計 8 個を確 認します。 ※センサ 0 とスター トバー検出センサ は、OR 接続されて います。同じ LED が 光りますが、故障で はありません。
	ディップスイッチ  "1" "0" 1 0 0 1	3	2	
	ディップスイッチ  "1" "0" 1 0 1 0	5	4	
	ディップスイッチ  "1" "0" 1 0 1 1	7	6	

5. 動作確認

※センサの調整方法

1

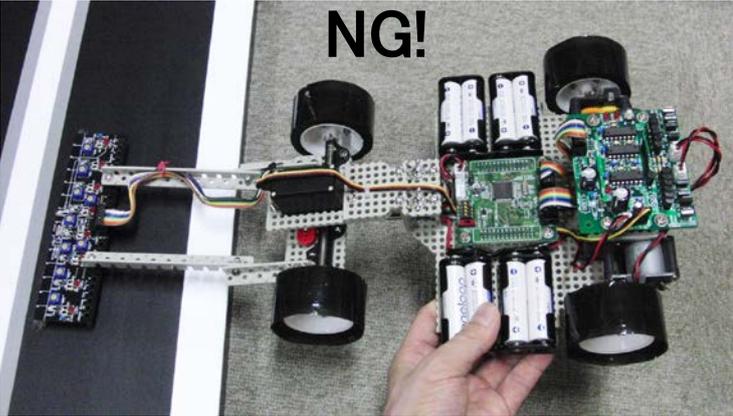


※横から見たところ

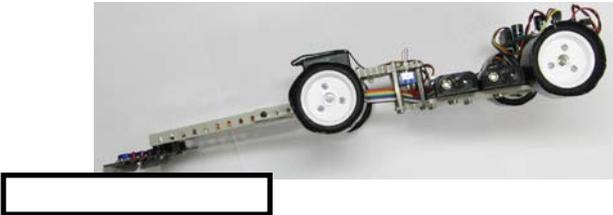


写真のようにコース中心の灰色線とセンサ基板を平行に置きます。このときマイコンカーは、コース同一面上に置き、走っている状態と同じにします。

2



※横から見たところ



このように、手で持ちながらセンサの調整をしようとしてもセンサとコースとの間隔が一定にならないため、きちんと調整できません。必ずコース同一面上に置いてください。

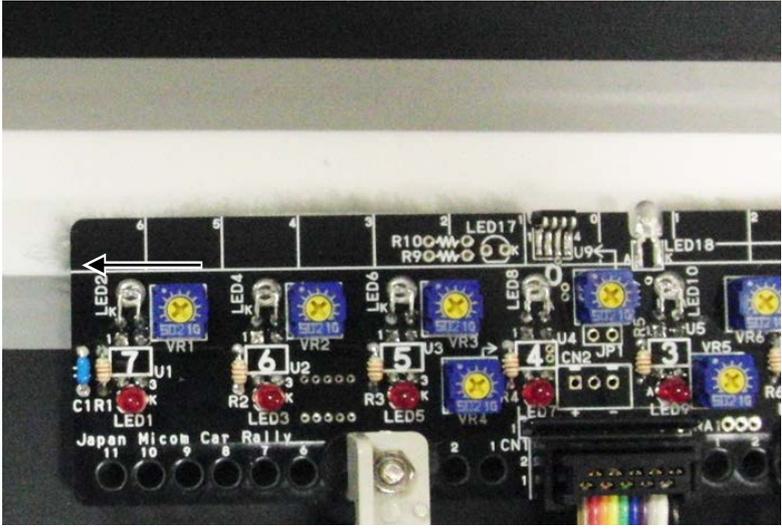
3



9 個のボリュームをすべて、反時計回りに回します。

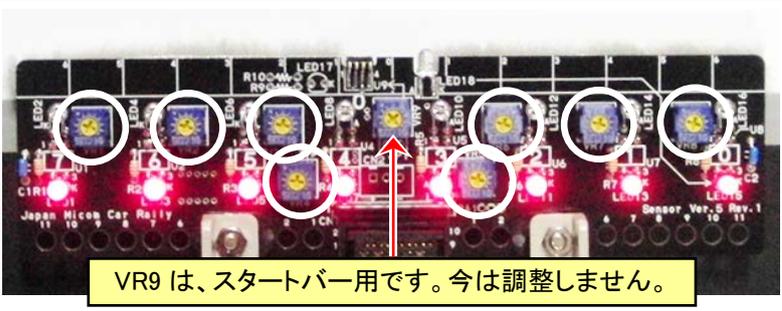
5. 動作確認

4



基板の横線とコースの白色と灰色の切り替わり部分を合わせます。真上から見て合わせるようにしてください。

5



8 個のボリュームを時計回りに回して LED が点くようにします。一つ一つゆっくりと回して、LED が点いた瞬間回すのを止めます。今回の調整で灰色も反応するように調整します。マイコンカーキットは、白色・灰色で反応するように調整します。

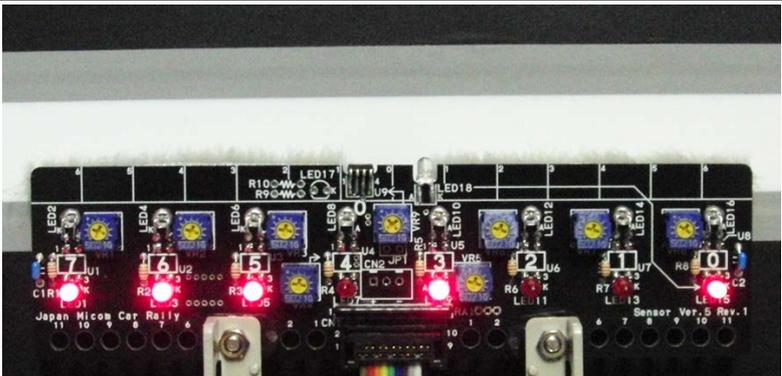
VR9 は、スタートバー用です。今は調整しません。

6



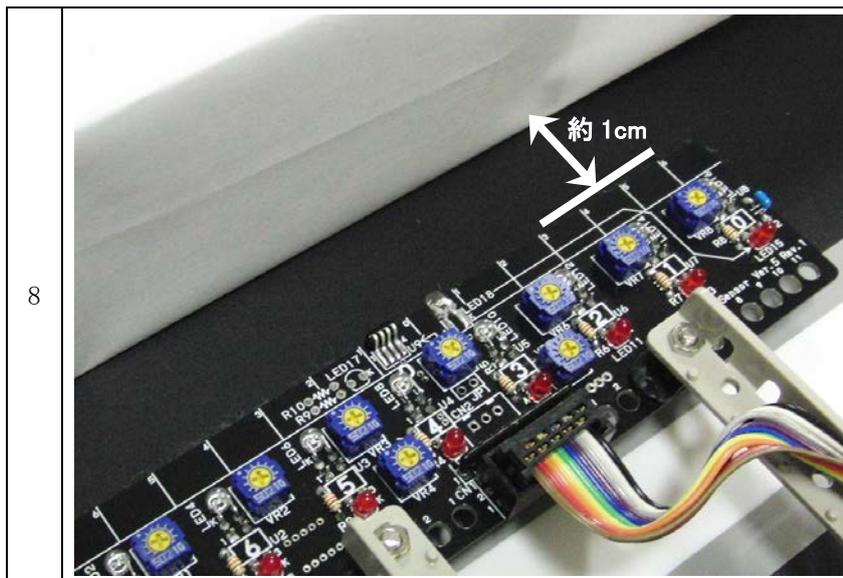
センサを少し下げます。すべて消えます。

7



再度センサを灰色の位置にゆっくりと平行に近づけます。点かない LED は感度を上げます(時計回り)。他の LED より先に点く場合、感度を下げます(反時計回り)。ほぼ同時に LED が点くように何度も調整します。

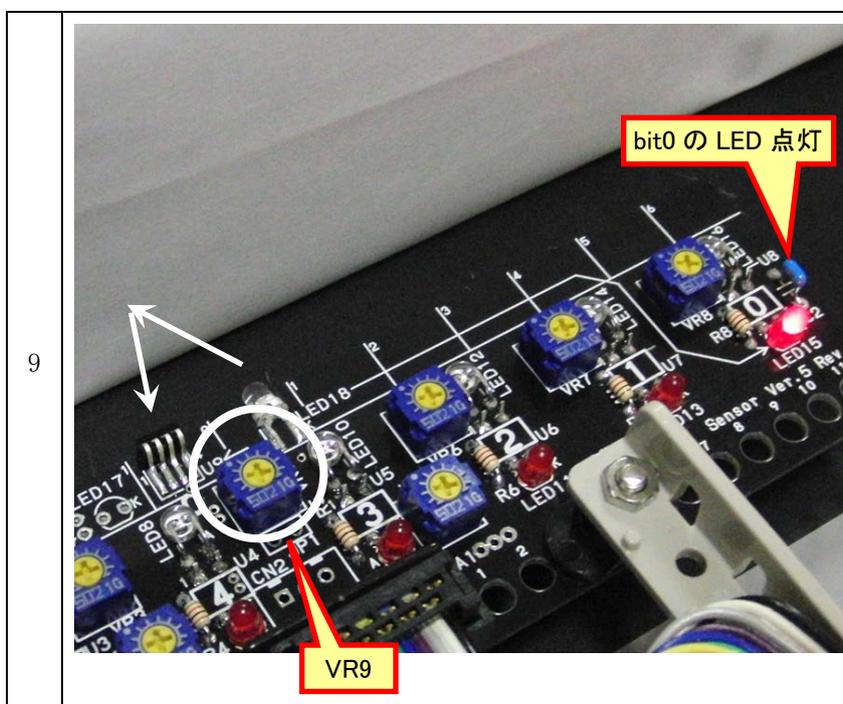
5. 動作確認



次に、スタートバーを検出するセンサの調整をします。

センサの先頭から約1cm離れたところに白色の板か紙を立てておきます。スタートバーの変わりです。

このとき、いちばん右のコース検出センサ(U8)の下が黒色でLED15 が消灯していることを確認してください。



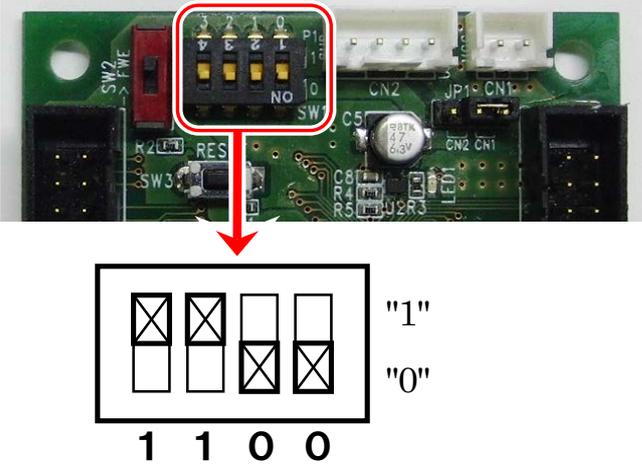
○の VR9 をゆっくりと時計回りに回し、LED15 が点くように調整します。

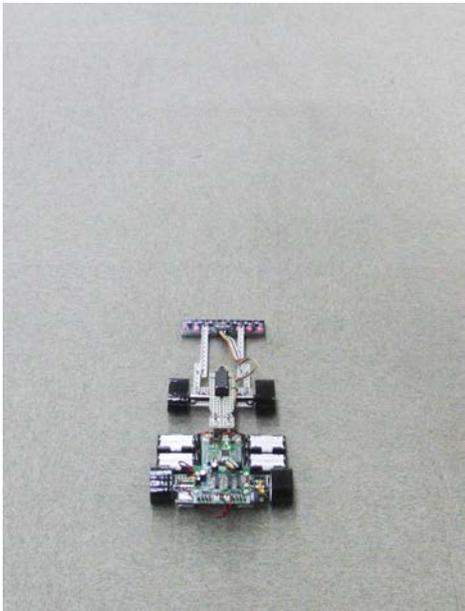
LED15 はいちばん右のコース検出センサ(U8)と兼用なので、このセンサの下が黒色で調整してください。

板や紙などを外したときに、LEDが消えれば完了です。

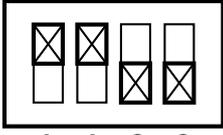
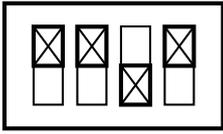
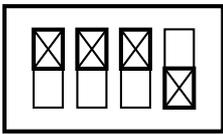
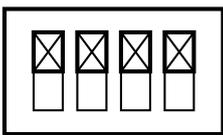
5. 動作確認

5.8 直進性の確認

1		<p>マイコンカーを走らせ、まっすぐに進むか確認します。廊下など平らで直線の長い場所に、マイコンカーを置いてください(ここでは、マイコンカーのコースには置きません)。ディップスイッチを”1100”の状態にします。</p>
---	---	--

2		<p>マイコンカーの電源スイッチを2つとも ON にします。電源を ON にしてから、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、まっすぐに進むか確認してください。</p> <p>曲がってしまう場合は、「5.4 サーボの動作確認」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、必ずまっすぐ進むように調整してください。最終的には、SERVO_CENTER 値を1ずつ調整するぐらいの心がけで調整してください。</p>
---	--	--

5. 動作確認

3	ディップスイッチ	PWM 値	停止するまでの時間	直進性の確認は、PWM 値と停止するまでの時間の違いで 4 パターンあります。PWM 値が大きいほど、また停止するまでの時間が長いほど、長く走ります。確保できる場所の長さに応じて 4 つのどれかから選び、平らで直線の長い場所でマイコンカーを走らせ、まっすぐに進むか確認してください。 ※サーボのセンタ調整は、マイコンカーキット Ver.5 kit12_38a プログラム解説マニュアル(R8C/38A 版)の「7. サーボセンタと最大切れ角の調整」で行うと便利です。
	 <p>"1" "0" 1 1 0 0</p>	50%	2 秒	
	 <p>"1" "0" 1 1 0 1</p>	50%	5 秒	
	 <p>"1" "0" 1 1 1 0</p>	100%	2 秒	
	 <p>"1" "0" 1 1 1 1</p>	100%	5 秒	

5.9 動作確認終了

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。

- マイコンカーキット Ver.5.1 kit12_38a プログラム解説マニュアル(R8C/38A 版)の「7. 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。
- マイコンカーキット Ver.5.1 kit12_38a プログラム解説マニュアル(R8C/38A 版)を参照しながら、ワークスペース「kit12_38a」のプロジェクト「kit12_38a」のプログラムを RY_R8C38 ボードに書き込み、コースを走らせてみてください。

6. プログラムソース

6.1 「kit12test_38a.c」のプログラム内容

```

1 : /******
2 : /* 対象マイコン R8C/38A */
3 : /* ファイル内容 マイコンカーキットVer. 5.1 動作確認プログラム (R8C/38A版) */
4 : /* バージョン Ver. 1.00 */
5 : /* Date 2012.05.07 */
6 : /* Copyright ジャパンマイコンカーラリー実行委員会 */
7 : /******
8 :
9 : /*
10 : マイコンカーキット用センサ基板Ver. 5、モータドライブ基板Ver. 5の
11 : 動作確認を行います。
12 : マイコンボードのディップスイッチにより動作確認する内容を変更します。
13 : DipSW
14 : bit3 2 1 0
15 : 0 0 0 0 LEDの確認 LEDが0.5秒間隔で交互に点灯
16 : 0 0 0 1 プッシュスイッチの確認 OFF時：LED0点灯 ON時：LED1点灯
17 : 0 0 1 0 サーボの確認 0° →右30° →左30° の繰り返し
18 : 0 0 1 1 動作無し
19 : 0 1 0 0 右モータの確認 正転→ブレーキの繰り返し
20 : 0 1 0 1 逆転→ブレーキの繰り返し
21 : 0 1 1 0 左モータの確認 正転→ブレーキの繰り返し
22 : 0 1 1 1 逆転→ブレーキの繰り返し
23 :
24 : 1 0 0 0 センサ確認 センサbit1,0をLED1,0に出力
25 : 1 0 0 1 センサbit3,2をLED1,0に出力
26 : 1 0 1 0 センサbit5,4をLED1,0に出力
27 : 1 0 1 1 センサbit7,6をLED1,0に出力
28 :
29 : 1 1 0 0 直進性の確認 PWM 50%で前進、2秒後ストップ
30 : 1 1 0 1 直進性の確認 PWM 50%で前進、5秒後ストップ
31 : 1 1 1 0 直進性の確認 PWM 100%で前進、2秒後ストップ
32 : 1 1 1 1 直進性の確認 PWM 100%で前進、5秒後ストップ
33 : */
34 :
35 : /*=====*/
36 : /* インクルード */
37 : /*=====*/
38 : #include "sfr_r838a.h" /* R8C/38A SFRの定義ファイル */
39 :
40 : /*=====*/
41 : /* シンボル定義 */
42 : /*=====*/
43 :
44 : /* 定数設定 */
45 : #define PWM_CYCLE 39999 /* モータPWMの周期 */
46 : #define SERVO_CENTER 3750 /* サーボのセンタ値 */
47 : #define HANDLE_STEP 22 /* 1° 分の値 */
48 :
49 : /*=====*/
50 : /* プロトタイプ宣言 */
51 : /*=====*/
52 : void init( void );
53 : unsigned char sensor_inp_all( unsigned char mask );
54 : unsigned char dipsw_get( void );
55 : unsigned char pushsw_get( void );
56 : void led_out( unsigned char led );
57 : void motor( int accele_l, int accele_r );
58 : void handle( int angle );
59 :
60 : /*=====*/
61 : /* グローバル変数の宣言 */
62 : /*=====*/
63 : unsigned long cnt0; /* timer関数用 */
64 : unsigned long cnt1; /* main内で使用 */
65 :
66 : /******
67 : /* メインプログラム */
68 : /******
69 : void main( void )
70 : {
71 :     unsigned char now_sw; /* 現在ディップスイッチ記憶 */
72 :     unsigned char before_sw; /* 前回ディップスイッチ記憶 */
73 :     unsigned char c; /* 作業用 */
74 :     int i; /* 作業用 */
75 :
76 :     /* マイコン機能の初期化 */
77 :     init(); /* 初期化 */
78 :     asm(" fset I "); /* 全体の割り込み許可 */
79 :

```

6. プログラムソース

```

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

```

6. プログラムソース

```

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

```

6. プログラムソース

```

258 : /*****
259 :  * R8C/38A スペシャルファンクションレジスタ (SFR)の初期化
260 :  *****/
261 : void init( void )
262 : {
263 :     int i;
264 :
265 :     /* クロックをXINクロック (20MHz)に変更 */
266 :     prc0 = 1; /* プロテクト解除 */
267 :     cm13 = 1; /* P4_6, P4_7をXIN-XOUT端子にする*/
268 :     cm05 = 0; /* XINクロック発振 */
269 :     for(i=0; i<50; i++ ); /* 安定するまで少し待つ(約10ms) */
270 :     ocd2 = 0; /* システムクロックをXINにする */
271 :     prc0 = 0; /* プロテクトON */
272 :
273 :     /* ポートの入出力設定 */
274 :     prc2 = 1; /* PD0のプロテクト解除 */
275 :     pd0 = 0x00; /* 7-0:センサ基板Ver. 5 */
276 :     pd1 = 0xd0; /* 5:RXD0 4:TXD0 3-0:DIP SW */
277 :     pd2 = 0xfe; /* 7-0:モータドライブ基板Ver. 5 */
278 :     pd3 = 0xff; /* */
279 :     p4 = 0x20; /* P4_5のLED:初期は点灯 */
280 :     pd4 = 0xb8; /* 7:XOUT 6:XIN 5:LED 2:VREF */
281 :     pd5 = 0xff; /* */
282 :     pd6 = 0xff; /* */
283 :     pd7 = 0xff; /* */
284 :     pd8 = 0xff; /* */
285 :     pd9 = 0x3f; /* */
286 :     pur0 = 0x04; /* P1_3~P1_0のブルアップON */
287 :
288 :     /* タイマRBの設定 */
289 :     /* 割り込み周期 = 1 / 20[MHz] * (TRBPRE+1) * (TRBPR+1)
290 :     = 1 / (20*10^6) * 200 * 100
291 :     = 0.001[s] = 1[ms]
292 :     */
293 :     trbmr = 0x00; /* 動作モード、分周比設定 */
294 :     trbpre = 200-1; /* プリスケアラレジスタ */
295 :     trbpr = 100-1; /* プライマリレジスタ */
296 :     trbic = 0x07; /* 割り込み優先レベル設定 */
297 :     trbcr = 0x01; /* カウント開始 */
298 :
299 :     /* タイマRD リセット同期PWMモードの設定*/
300 :     /* PWM周期 = 1 / 20[MHz] * カウントソース * (TRDGRA0+1)
301 :     = 1 / (20*10^6) * 8 * 40000
302 :     = 0.016[s] = 16[ms]
303 :     */
304 :     trdpsr0 = 0x08; /* TRDIOB0, C0, D0端子設定 */
305 :     trdpsr1 = 0x05; /* TRDIOA1, B1, C1, D1端子設定 */
306 :     trdmr = 0xf0; /* バッファレジスタ設定 */
307 :     trdfcr = 0x01; /* リセット同期PWMモードに設定 */
308 :     trdcr0 = 0x23; /* ソースカウントの選択:f8 */
309 :     trdgra0 = trdgrc0 = PWM_CYCLE; /* 周期 */
310 :     trdgrb0 = trdgrd0 = 0; /* P2_2端子のON幅設定 */
311 :     trdgra1 = trdgrc1 = 0; /* P2_4端子のON幅設定 */
312 :     trdgrb1 = trdgrd1 = SERVO_CENTER; /* P2_5端子のON幅設定 */
313 :     trdoer1 = 0xcd; /* 出力端子の選択 */
314 :     trdstr = 0x0d; /* TRD0カウント開始 */
315 : }
316 :
317 : /*****
318 :  * タイマRB 割り込み処理
319 :  *****/
320 : #pragma interrupt intTRB(vect=24)
321 : void intTRB( void )
322 : {
323 :     cnt0++;
324 :     cnt1++;
325 : }
326 :
327 : /*****
328 :  * センサ状態検出(スタートバーセンサを含めすべてのセンサ)
329 :  * 引数 マスク値
330 :  * 戻り値 センサ値
331 :  *****/
332 : unsigned char sensor_inp_all( unsigned char mask )
333 : {
334 :     unsigned char sensor;
335 :
336 :     sensor = ~p0;
337 :     sensor &= mask;
338 :
339 :     return sensor;
340 : }
341 :

```

6. プログラムソース

```

342 : /*****/
343 : /* ディップスイッチ値読み込み */
344 : /* 戻り値 スイッチ値 0~15 */
345 : /*****/
346 : unsigned char dipsw_get( void )
347 : {
348 :     unsigned char sw;
349 :
350 :     sw = p1 & 0x0f;          /* P1_3~P1_0読み込み */
351 :
352 :     return sw;
353 : }
354 :
355 : /*****/
356 : /* プッシュスイッチ値読み込み */
357 : /* 戻り値 スイッチ値 ON:1 OFF:0 */
358 : /*****/
359 : unsigned char pushsw_get( void )
360 : {
361 :     unsigned char sw;
362 :
363 :     sw = ~p2;              /* スイッチのあるポート読み込み */
364 :     sw &= 0x01;
365 :
366 :     return sw;
367 : }
368 :
369 : /*****/
370 : /* LED制御 */
371 : /* 引数 スイッチ値 LED0:bit0 LED1:bit1 "0":消灯 "1":点灯 */
372 : /* 例)0x3→LED1:ON LED0:ON 0x2→LED1:ON LED0:OFF */
373 : /*****/
374 : void led_out( unsigned char led )
375 : {
376 :     unsigned char data;
377 :
378 :     led = ~led;
379 :     led <<= 6;
380 :     data = p2 & 0x3f;
381 :     p2 = data | led;
382 : }
383 :
384 : /*****/
385 : /* モータ速度制御 */
386 : /* 引数 左モータ:-100~100、右モータ:-100~100 */
387 : /* 0で停止、100で正転100%、-100で逆転100% */
388 : /* 戻り値 なし */
389 : /*****/
390 : void motor( int accele_l, int accele_r )
391 : {
392 :     int sw_data;
393 :
394 :     sw_data = dipsw_get() + 5;
395 :     accele_l = accele_l * sw_data / 20;
396 :     accele_r = accele_r * sw_data / 20;
397 :
398 :     /* 左モータ制御 */
399 :     if( accele_l >= 0 ) {
400 :         p2 &= 0xfd;
401 :         trdgrd0 = (long)( PWM_CYCLE - 1 ) * accele_l / 100;
402 :     } else {
403 :         p2 |= 0x02;
404 :         trdgrd0 = (long)( PWM_CYCLE - 1 ) * ( -accele_l ) / 100;
405 :     }
406 :
407 :     /* 右モータ制御 */
408 :     if( accele_r >= 0 ) {
409 :         p2 &= 0xf7;
410 :         trdgrd1 = (long)( PWM_CYCLE - 1 ) * accele_r / 100;
411 :     } else {
412 :         p2 |= 0x08;
413 :         trdgrd1 = (long)( PWM_CYCLE - 1 ) * ( -accele_r ) / 100;
414 :     }
415 : }
416 :
417 : /*****/
418 : /* サーボハンドル操作 */
419 : /* 引数 サーボ操作角度:-90~90 */
420 : /* -90で左へ90度、0でまっすぐ、90で右へ90度回転 */
421 : /*****/
422 : void handle( int angle )
423 : {
424 :     /* サーボが左右逆に動く場合は、「-」を「+」に替えてください */
425 :     trdgrd1 = SERVO_CENTER - angle * HANDLE_STEP;
426 : }
427 :
428 : /*****/
429 : /* end of file */
430 : /*****/

```