

プロポ信号受信 リレーモータドライブ基板 製作・プログラム解説 マニュアル

※本キットは、組み立て材料一式をセットにした未完成品です。組み立てには、半田付けが必要です。

本マニュアルで説明しているセット内容	プロポ信号受信リレーモータドライブ基板製作キット
本基板の対象マイコンボード	本基板内の R8C/M12A マイコンを使いますので、別途マイコンボードは必要ありません。ただし、プログラムの書き込みに、RY-WRITER 基板を使用します。別途ご用意ください。
本基板のプログラムについての説明	本マニュアルで説明しています。

第 1.00A 版
2015.10.29

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.6.0H)

著作権

- ・本マニュアルに関する著作権は株式会社日立ドキュメントソリューションズに帰属します
- ・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、文書による株式会社日立ドキュメントソリューションズ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、株式会社日立ドキュメントソリューションズ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、株式会社日立ドキュメントソリューションズは、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
1.1 特徴.....	1
1.2 仕様.....	2
1.3 外観.....	3
1.4 各部(コネクタ、スイッチなど)の説明.....	4
1.5 リレーによる、速度の切り替え方法.....	4
1.6 基板、コネクタ寸法.....	5
1.7 部品面パターン図.....	6
1.8 半田面パターン図.....	7
1.9 回路図.....	8
2. 組み立てに必要な工具、部品	10
2.1 工具.....	10
2.2 セット以外で組み立てに必要な部品.....	12
3. プロポリレー基板の組み立て	13
3.1 部品表.....	13
3.2 部品面のシルク.....	16
3.3 JP3、JP4 の半田付け.....	17
3.4 抵抗(4.7k Ω)の取り付け.....	18
3.5 抵抗(3.3k Ω)の取り付け.....	19
3.6 抵抗(1k Ω)の取り付け.....	20
3.7 積層セラミックコンデンサ(1 μ F)の取り付け.....	21
3.8 汎用整流用ダイオード(1N4007)の取り付け.....	22
3.9 ショットキーバリアダイオード(11EQS04)の取り付け.....	23
3.10 R8C マイコン(M120AN)の取り付け.....	24
3.11 LED(EBR3338S)の取り付け.....	25
3.12 タクトスイッチ(B3F-1000)の取り付け.....	26
3.13 スライドスイッチ(SS-12D00-G5)の取り付け.....	27
3.14 抵抗内蔵トランジスタ(RN1201)の取り付け.....	28
3.15 XH コネクタ 4ピンオス(B4B-XH-A)の取り付け.....	29
3.16 ピンヘッダ(1列 \times 40P)の取り付け.....	30
3.17 ターミナルブロック(2ピン・緑・縦)の取り付け.....	31
3.18 ターミナルブロック(3ピン・青・縦)の取り付け.....	32
3.19 ターミナルブロック(2ピン・青・縦)の取り付け.....	33
3.20 電解コンデンサ(1000 μ F)の取り付け.....	34
3.21 リレーの取り付け.....	35
3.22 電気二重層コンデンサ(1.5F)の取り付け.....	36
3.23 三端子レギュレータ(LM2940CT-5.0)の取り付け.....	37
3.24 プロポ受信機接続ケーブルの作成.....	39
3.25 パターンの強化.....	41
4. 接続	42
4.1 接続図.....	42
4.2 マイコンボードの電源を別にする.....	44
4.3 複数のプロポリレー基板を使って、多くのモータを制御する.....	45

4.4 センサの接続方法.....	46
4.4.1 NPN 出力タイプの場合(12V 動作のセンサ).....	46
4.4.2 2線式の場合(12V 動作のセンサ).....	47
4.4.3 スイッチを接続する場合.....	48
4.4.4 5V で動作するアナログ出力のセンサの場合.....	48
5. 動作確認.....	49
5.1 動作確認の結線.....	49
5.2 ワークスペース(プログラム)のダウンロードとプログラムの書き込み.....	50
5.3 動作確認.....	53
6. ワークスペース「r8cm12a_propo_relay」.....	54
6.1 プロジェクトの構成.....	54
7. プログラム「propo_relay.c」.....	55
7.1 概要.....	55
7.2 プログラムリスト.....	55
7.3 関数.....	62
7.4 プログラムの解説.....	64
7.4.1 グローバル変数部分.....	64
7.4.1 シンボル定義(チャンネル A のパルス幅定義部分).....	65
7.4.2 main 関数.....	66
7.4.3 調整のポイント.....	67
7.4.4 モータを動作させている部分.....	68
8. 付録.....	71
8.1 ターミナルブロックについて.....	71
8.2 モータの加工.....	72
9. 参考文献.....	73

1. 概要

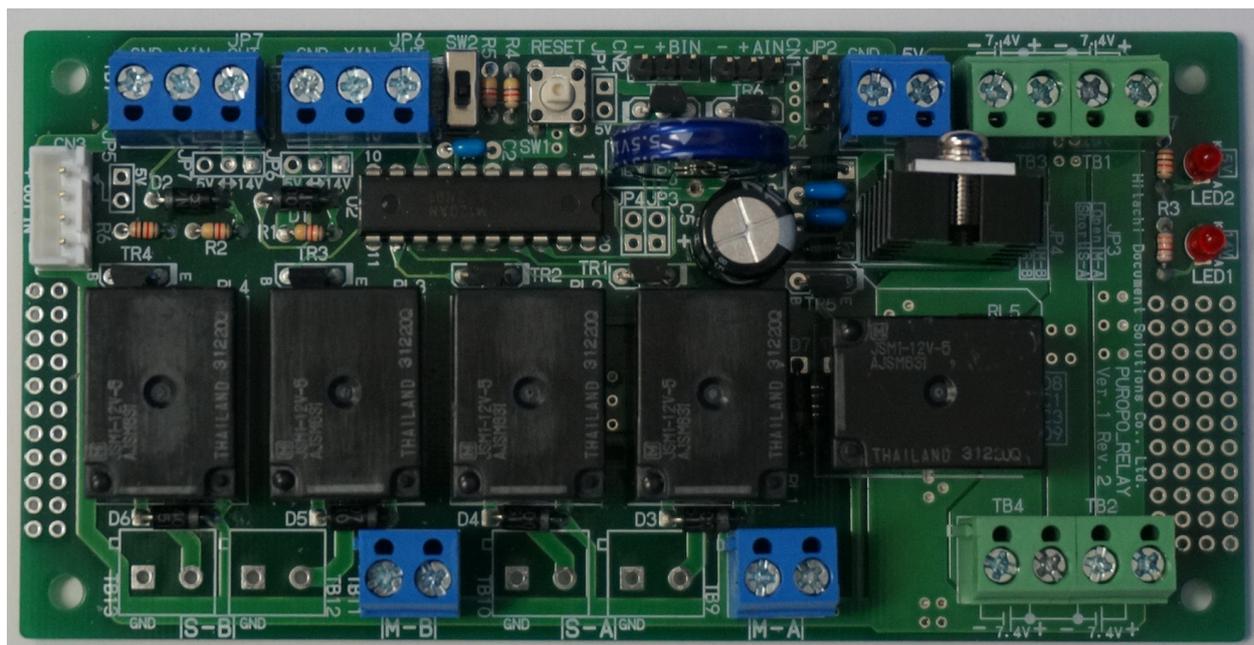
1. 概要

本マニュアルは、プロポ信号受信リレーモータドライブ基板(以下、プロポリレー基板)の製作、およびプログラム解説マニュアルです。

1.1 特徴

プロポリレー基板の特徴を下記に示します。

- プロポ受信機から出力される信号(2チャンネル)をプロポリレー基板で入力して、DC モータ(2個)を制御することができます。
- 電源は7.2Vまたは7.4Vのバッテリーを2本直列で使用することを想定しています。2本のバッテリーを直列で接続できるよう、基板には端子台(ターミナルブロック)を搭載しています。
- 本基板のリレー(JSM1-12V-5)は25A(2分以内)まで電流を流すことができます(ただし、20A以上の電流を流すときはジャンパ線を追加して、基板のパターンを強化する必要があります)。
- PNPトランジスタ出力のセンサを2個、外付け回路無しで接続することができます。公開されているプログラムに、各自でセンサの信号を入力してモータを制御するプログラムを追加してください。



▲プロポ信号受信リレーモータドライブ基板 実装例

1. 概要

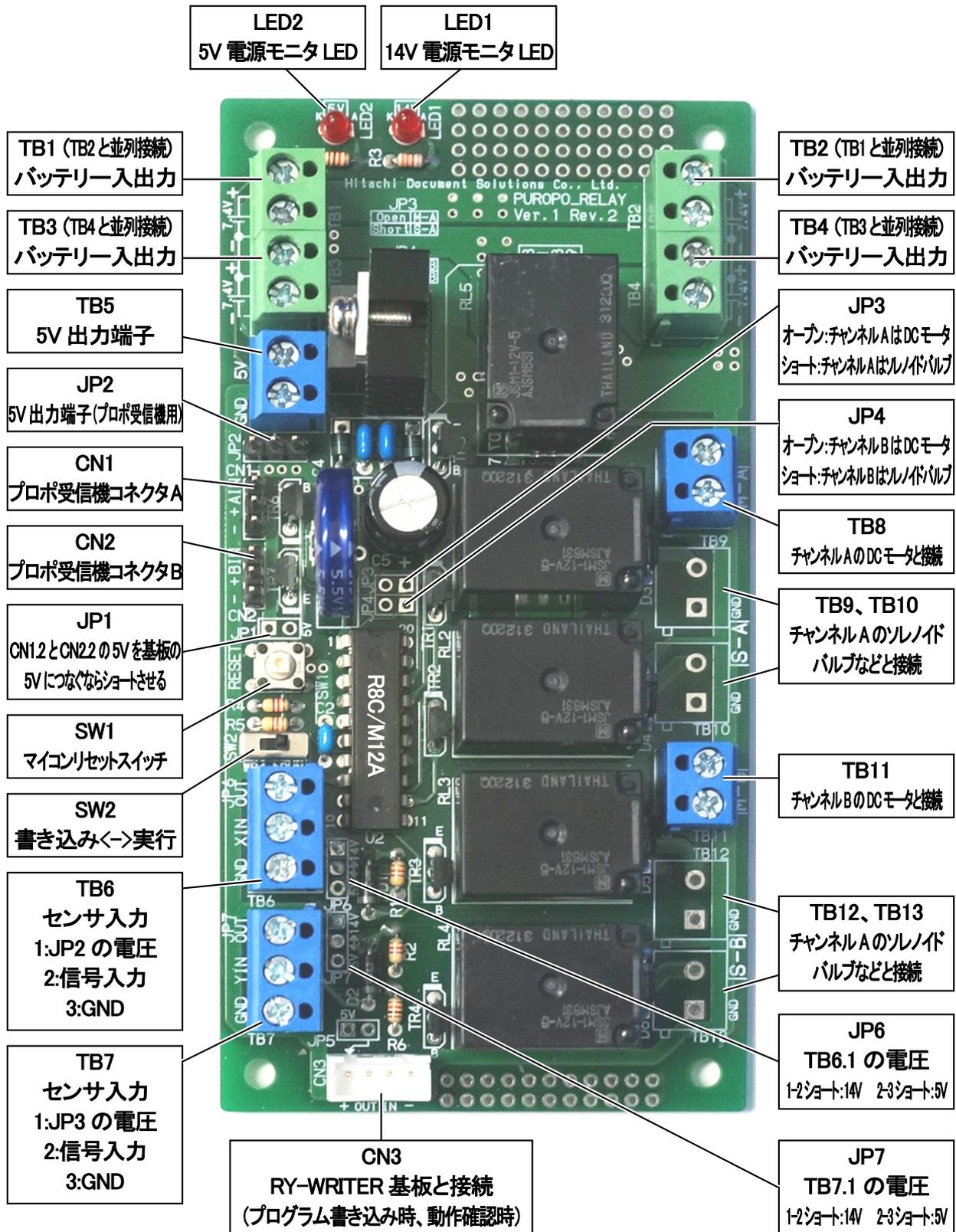
1.2 仕様

項目	内容	
電源電圧	7～20V ・7.2V または 7.4V のバッテリーを 2 本直列で使用することを想定 ・基板にはバッテリー 2 本分の端子台を搭載	
制御マイコン	ルネサス エレクトロニクス製 R8C/M12A	
プロボ信号受信機からの信号を受信できる数	2 チャンネル	
制御対象	チャンネル A	DC モータ、またはソレノイドバルブ
	チャンネル B	DC モータ、またはソレノイドバルブ
モータの制御	リレーの ON、OFF によるモータ制御 モータに供給する電圧は、2段階(High/Low)切り替え可能 ※High/Low切り替えリレー(基板の部品番号:RL5)により、バッテリー1本分の電圧(低速:Low)、バッテリー2本分の電圧(高速:High)のどちらかを供給するか選択可能 ※High、Low は、チャンネル A とチャンネル B のモータ共通です。どちらか Low、どちらか High というような制御はできません。	
信号入力	PNP トランジスタ出力型のセンサ信号を 2 チャンネル入力可能 ※入力した信号を使う場合、各自でプログラムの追加が必要です	
基板外形	130×65×厚さ 1.6mm、銅厚 70 μ m	
完成時の寸法(実測)	130×65×高さ約 23mm	
重量(完成品の実測)	約 135g ※リード線の長さや半田の量で変わります	

1. 概要

1.3 外観

外観を、下図に示します。



1. 概要

1.4 各部(コネクタ、スイッチなど)の説明

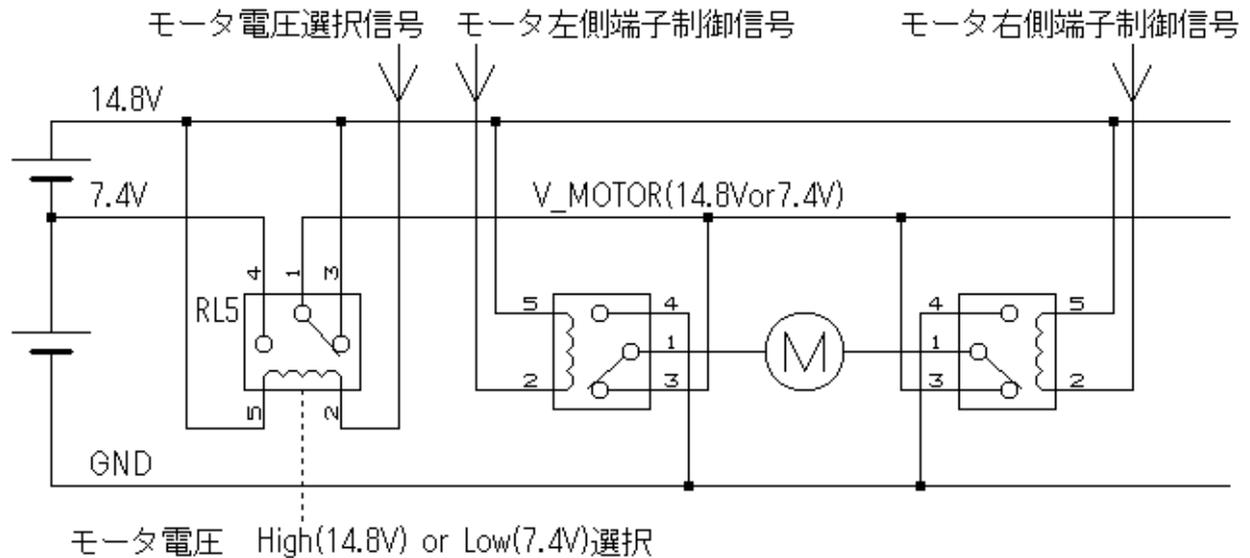
SW1	R8C/M12A マイコンのリセットスイッチです。
SW2	書き込み・実行の切り替えスイッチです。電源が切れている状態で、プログラム書き込み時はスイッチを基板の外側へ、プログラム実行時はスイッチを内側にして、電源を入れてください。
JP1	プロボ受信機から供給される5Vを使う場合、ジャンパをショートさせます。このとき、三端子レギュレータ(U1)は必要ありません。
JP3	オープン:チャンネル A の制御は DC モータ ショート:チャンネル A の制御はソレノイドバルブ
JP4	オープン:チャンネル B の制御は DC モータ ショート:チャンネル B の制御はソレノイドバルブ
JP6	TB6 にセンサをつなぐとき、センサへの供給電源が 12V の場合は 1-2 ピンを、5V の場合は 2-3 ピンをショートさせます。
JP7	JP2 と同じで、TB7 が対象となります。

1.5 リレーによる、速度の切り替え方法

モータに供給する電圧は、2段階切り替え可能です。

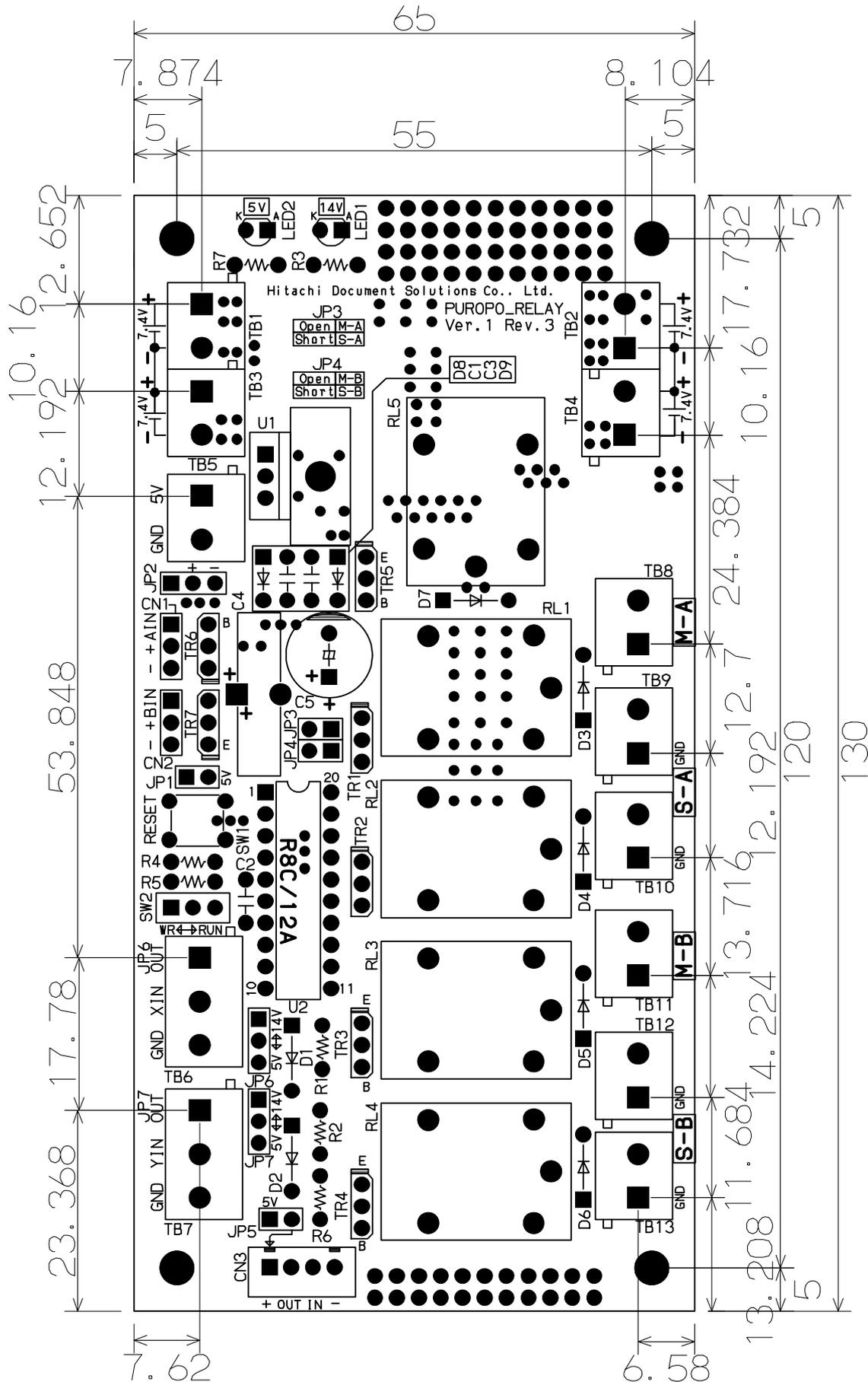
High/Low 切り替えリレー(基板の部品番号:RL5)により、電池 1 本分の電圧(低速:Low)、電池2本分の電圧(高速:High)のどちらかを供給するか選択します。

High、Low は、チャンネル A とチャンネル B のモータ共通です。どちらか Low、どちらか High というような制御はできません。両モータとも Low か、High となります。



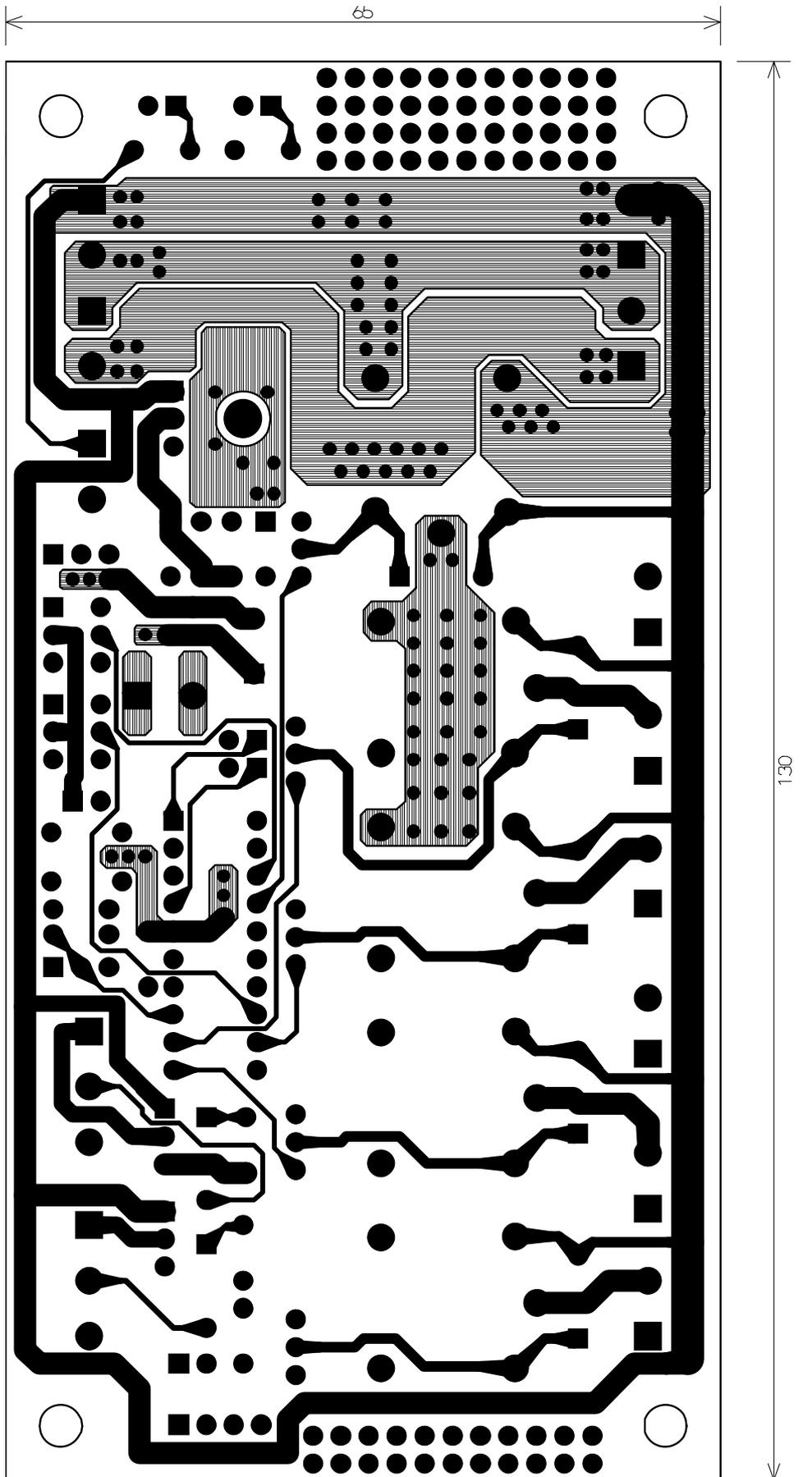
1. 概要

1.6 基板、コネクタ寸法



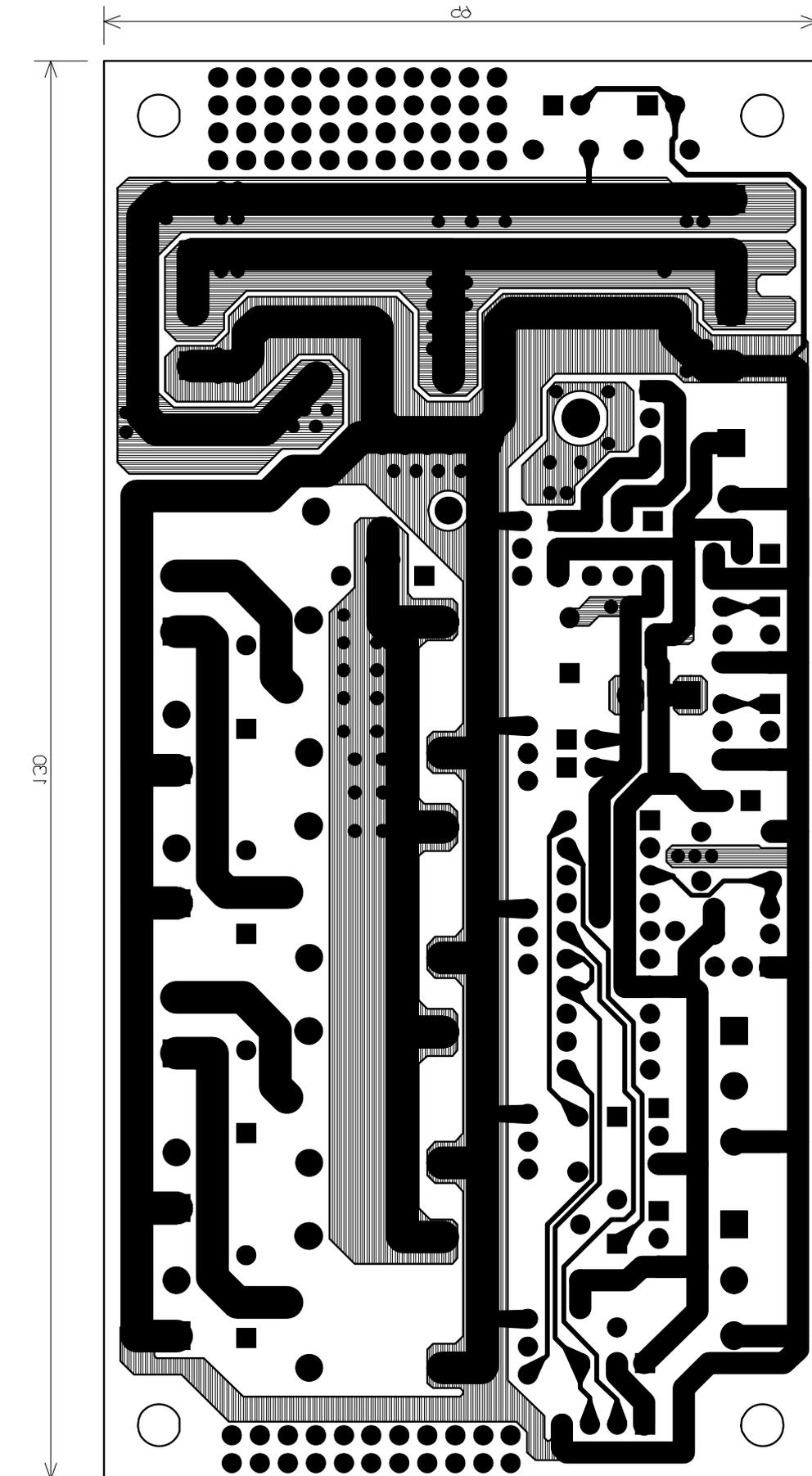
1. 概要

1.7 部品面パターン図



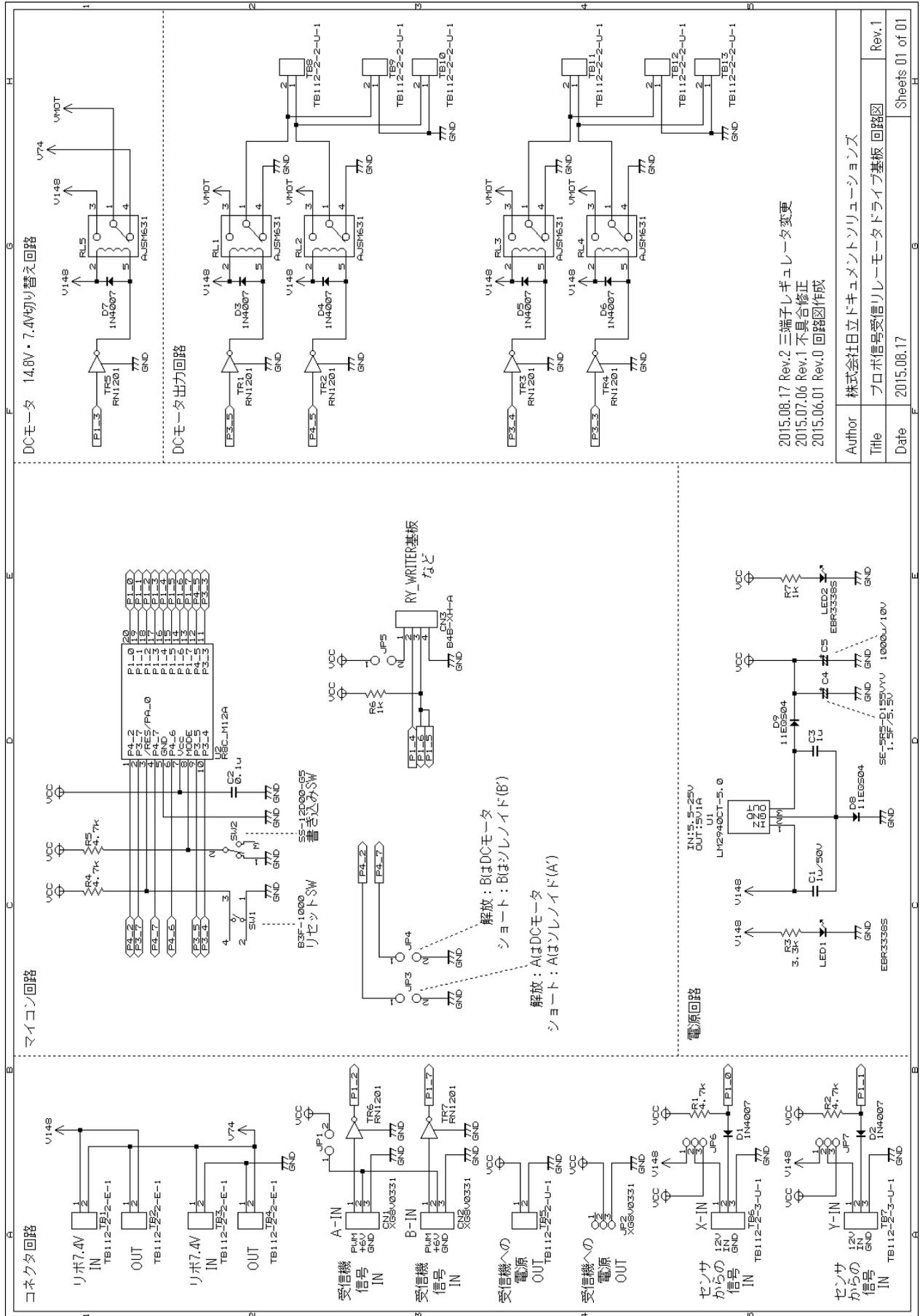
1. 概要

1.8 半田面パターン図



1. 概要

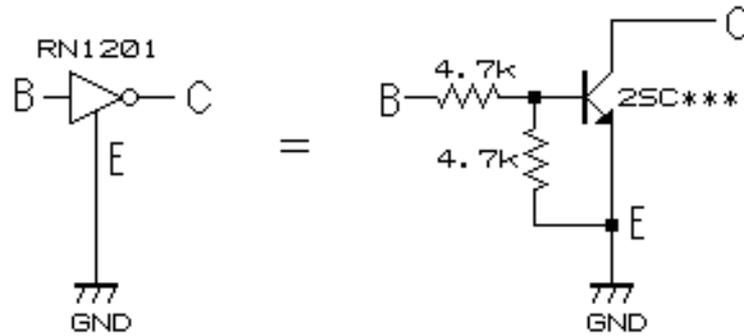
1.9 回路図



1. 概要

※RN1201(TR1、TR2)について

RN1201 は抵抗を内蔵しているトランジスタです。B(ベース)に 5V を加えるとトランジスタが ON して C(コレクタ)が 0V になるため E(エミッタ)を 0V に接続している場合、今回の回路図では NOT 回路のように書いています。



2. 組み立てに必要な工具、部品

2.1 工具

プロボリレー基板の組み立てに必要な工具類を、下表に示します。

<p>ミニチュア ニッパ</p>		<p>リード線を切るのに使います。</p>
<p>ミニチュア ペンチ</p>		<p>リード線を曲げたり、部品を固定するのに使います。</p>
<p>ワイヤ ストリッパ</p>		<p>線の被覆を剥くのに使います。 ※写真はホーザン(株)の P-946 です。現在販売されているのは、P-960 です。</p>
<p>圧着 ペンチ</p>		<p>コネクタのコンタクトピンを圧着するのに使います。 ※写真はホーザン(株)の P-706 です。</p>
<p>ピンセット</p>		<p>部品を持つのに使います。</p>

2. 組み立てに必要な工具、部品

<p>半田こて</p>		<p>部品を半田付けするのに使います。50～100W 程度の半田こてを用意します。</p> <p>※写真は、HAKKO FX600-02 です。手元のダイヤルで、200℃、270℃、320℃、370℃、420℃、500℃に温度調節することができる半田こてです。マイコンカーラー販売サイト (https://www2.himdx.net/mcr) で販売しています。</p> <ul style="list-style-type: none"> ●M-S327: はんだこて(HAKKO FX-600-02) ●M-S328: はんだこて台(HAKKO 633-01)
<p>半田こて台</p>		<p>半田こてを置くときに使います。 写真は、マイコンカーラー販売の「M-S328 はんだこて台 (HAKKO 633-01)(1 台)」です。</p>
<p>はさみ</p>		<p>セットの袋を切るなどで使います。</p>
<p>テスタ</p>		<p>回路の導通チェックなどをします。</p>

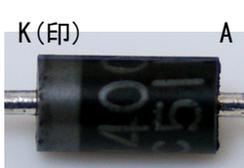
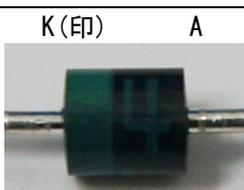
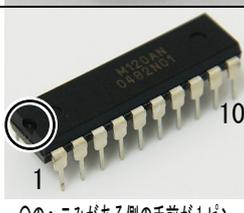
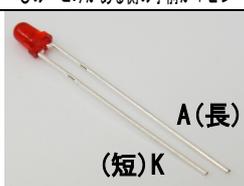
2.2 セット以外で組み立てに必要な部品

今回のセットに同封されておらず、別途必要な部品を下記に示します。

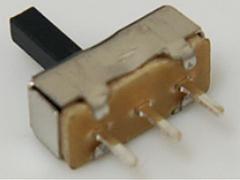
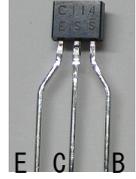
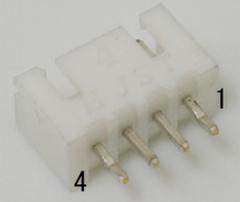
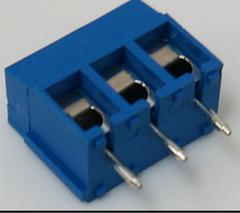
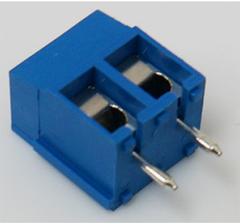
<p>半田</p>		<p>部品を基板に半田付けするときに使用します。マイコンカーラリー販売では、下記を取りそろえています。</p> <ul style="list-style-type: none"> •M-S329:糸半田(千住金属 スパークルソルダ φ0.6 150g) •M-S330:糸半田(千住金属 スパークルソルダ φ0.8 150g) •M-S331:糸半田鉛フリー(千住金属 ESCF3M705 φ0.6 100g) •M-S332:糸半田鉛フリー(千住金属 ESCF3M705 φ0.8 100g)
<p>スズメッキ線</p>	<p>No image</p>	<p>パターンを強化するために必要です。φ0.6～φ1.0mm 程度のスズメッキ線 1m 程度、必要です。</p>

3. プロポリレー基板の組み立て

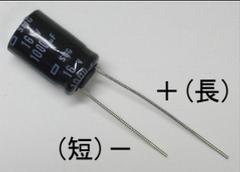
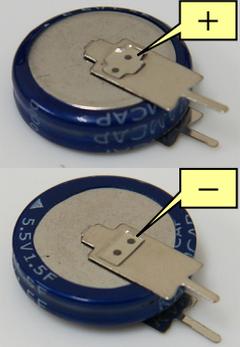
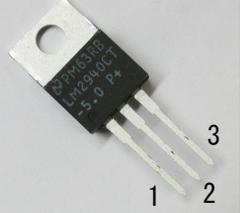
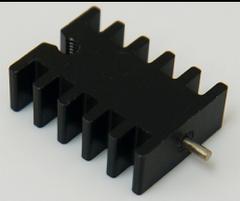
3.1 部品表

部品番号	名称	型式	写真	メーカー	数量
	基板	PUROPO_RELAY_ VER1_REV3 65×130×t1.6mm パターン厚 70 μ m		(株)日立ドキュメントソリューションズ	1
R1, 2, 4, 5	抵抗	CFS1/4C 4.7kΩ (黄・紫・赤・金)		コーア(株)	4
R3	抵抗	CFS1/4C 3.3kΩ (橙・橙・赤・金)		コーア(株)	1
R6, 7	抵抗	CFS1/4C 1kΩ (茶・黒・赤・金)		コーア(株)	2
C1, 2, 3	積層セラミック コンデンサ	RDER71H105K2K1H03B 1 μ F/50V 5.08mm ピッチ		(株)村田製作所	3
D1, 2, 3, 4, 5, 6, 7	汎用整流用ダイオード	1N4007 1000V/1A		PANJIT INTERNATIONAL INC.	7
D8, 9	ショットキーバリア ダイオード	11EQS04 40V/1A V _{FM} (せん頭順電圧):0.55V		日本インター (株)	2
U2	R8C マイコン	R8C/M12A M120AN	 0のへこみがある側の手前が1ピン	ルネサス エレクトロニクス(株)	1
LED1,2	LED	EBR3338S φ3mm・赤		スタンレー電気 (株)	2

3. プロボリレー基板の組み立て

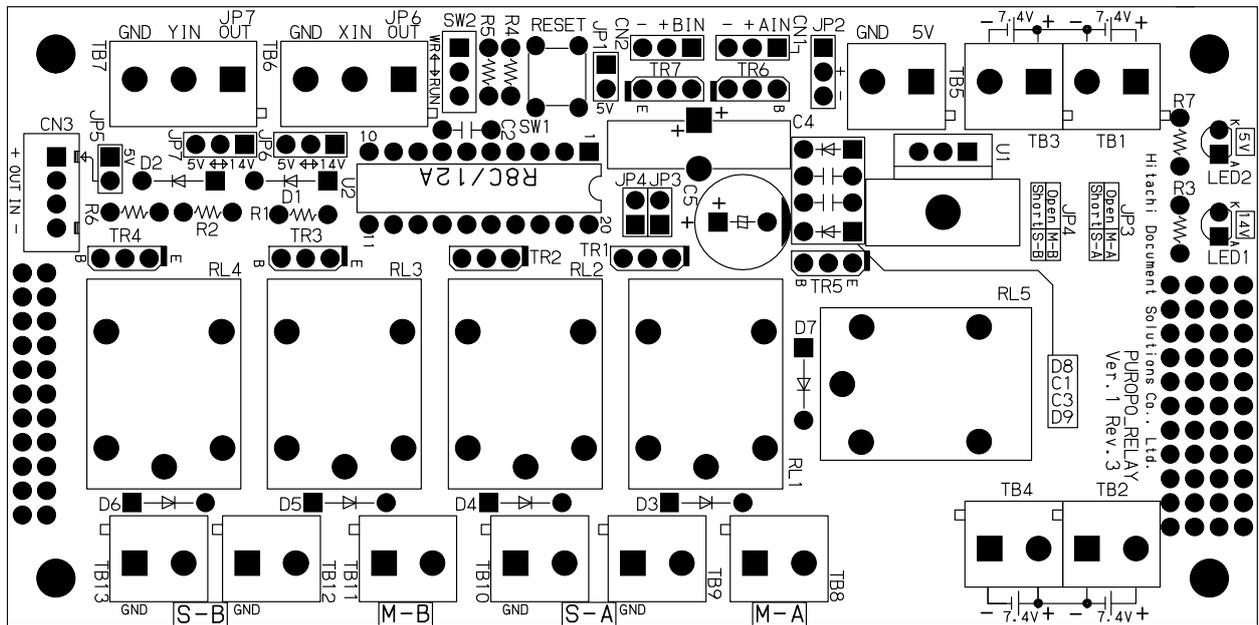
SW1	タクトスイッチ	B3F-1000	 この数字は関係ありません	オムロン(株)	1
SW2	スライドスイッチ	SS-12D00-G5		Zhejiang Jianfu Electronics Co., Ltd	1
TR1,2,3,4, 5,6,7	抵抗内蔵トランジスタ	RN1201		(株)東芝セミコンダクター社	7
CN3	XHコネクタ4ピンオス	B4B-XH-A		日本圧着端子製造(株)	1
CN1,2, JP2	ピンヘッダ	1列×40P ※カットして使用します		(株)秋月電子通商	1
TB1, 2, 3, 4	ターミナルブロック 2ピン(緑)(縦)	TB112-2-2-E-1		Alphaplus Connectors & Cables Corp.	4
TB6, 7	ターミナルブロック 3ピン(青)(縦)	TB112-2-3-U-1		Alphaplus Connectors & Cables Corp.	2
<ul style="list-style-type: none"> ●共通 TB5 ●チャンネル A TB8, または TB9,10 ●チャンネル B TB11, または TB12,13 <p>※チャンネル A とチャンネル B の実装位置 は、後述します</p>	ターミナルブロック 2ピン(青)(縦)	TB112-2-2-U-1		Alphaplus Connectors & Cables Corp.	5

3. プロポリレー基板の組み立て

C5	電解コンデンサ	SMG160E102MJ16S 1000 μ F/16V		日本ケミコン (株)	1
RL1, 2, 3, 4, 5	リレー	JSM1-12V-5 ・定格制御容量(抵抗負荷) 15A16V DC ・接点最大通電電流(20 度時) 25A/2 分間 15A/1 時間		パナソニック電 工(株)	5
C4	電気二重層コ ンデンサ	SE-5R5-D155VYV 1.5F/5.5V(タテ型) ※プラス、マイナスは、 スポット溶接部分の 形で判断します		KAMCAP	1
U1	三端子レギュレ ータ	LM2940CT-5.0		ナショナル セミ コンダクター ジ ャパン(株)	1
U1 用	放熱器	PC2116-7-PB		(株)放熱器のオ ーエス	9
放熱器 固定用	座金ネジ	ϕ 3 \times 8mm		各社	9
受信機 接続用	分割ロングピン ソケット	1 列 \times 42 ピン		(株)秋月電子 通商	1
受信機 接続用	10 芯スダレ フラット ケーブル	FLEX-S410-7/0.127 1.27mm ピッチ		沖電線(株)	0.6 m
ピン ソケット用	熱収縮 チューブ	スミチューブ C 透明 ϕ 1		住友電工ファイ ンポリマー(株)	0.5 m

3.2 部品面のシルク

部品面は、白い文字のある面です。この面から部品を取り付けます。

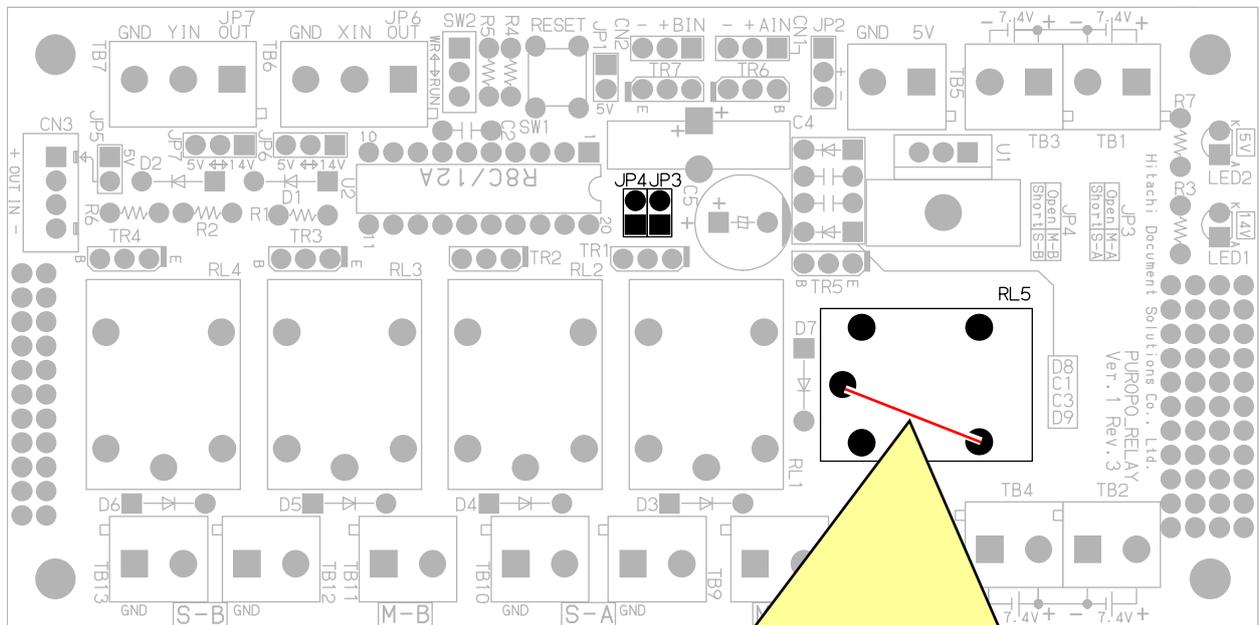


3. プロボリレー基板の組み立て

3.3 JP3、JP4 の半田付け

DC モータを制御するか、ソレノイドバルブを制御するかによって、JP3 と JP4 を、それぞれオープン(何もしない)にするか、ショートさせるか決めます。

JP3	オープン(何もしない)	チャンネル A の制御は DC モータを使用します。
	シ ョ ー ト	チャンネル A の制御はソレノイドバルブを使用します。 ソレノイドバルブとして使用する場合、RL5 の下記の位置をショートさせてください。
JP4	オープン(何もしない)	チャンネル B の制御は DC モータを使用します。
	シ ョ ー ト	チャンネル B の制御はソレノイドバルブを使用します。 ソレノイドバルブとして使用する場合、RL5 の下記の位置をショートさせてください。



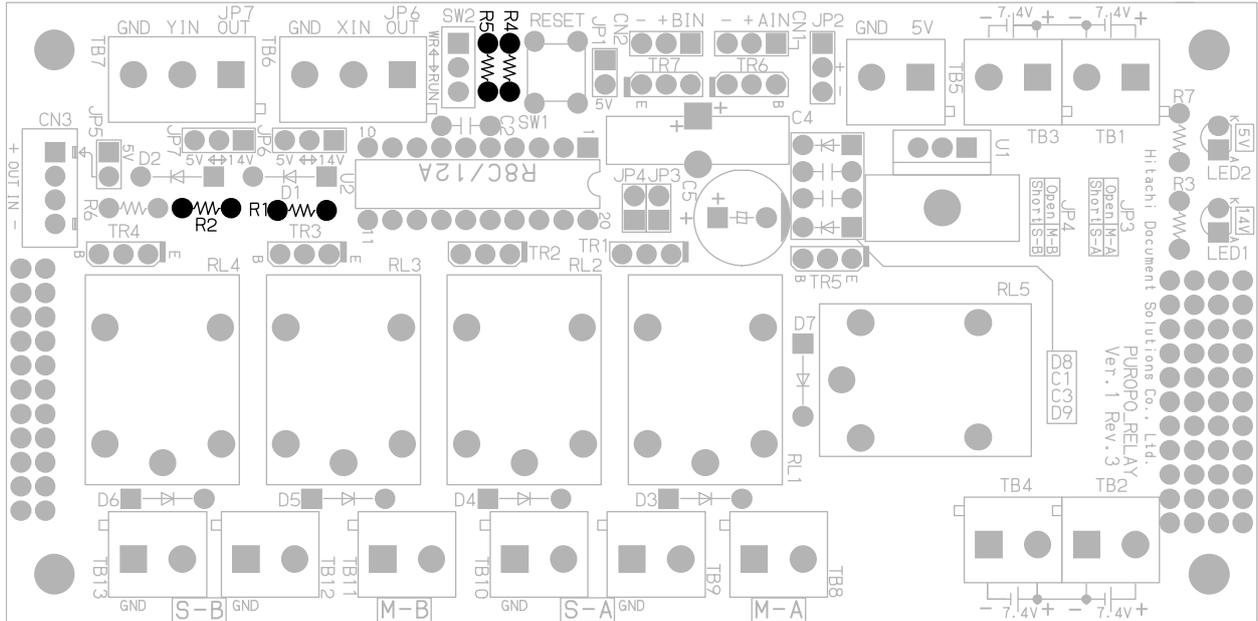
ソレノイドバルブを 1 個以上、使用する場合は、RL5 のこの部分を線でショートさせます。
RL5 部分にリレーは実装しません。
ソレノイドバルブを 1 個以上、使用する場合は、DC モータは必ず High になります。Low にはできません。
※DC モータを 1 個、ソレノイドバルブを 1 個使う場合、DC モータに流す電流がこの線を通りますので、電流に見合った線の太さにしてください。

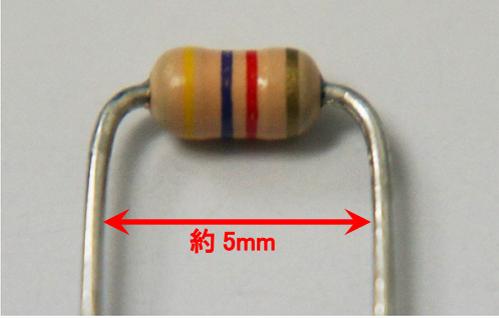
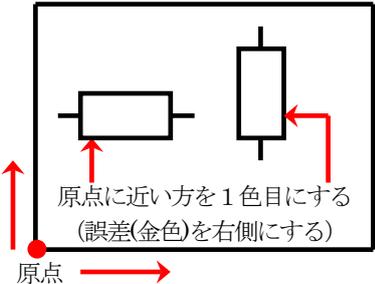
3. プロポリレー基板の組み立て

3.4 抵抗(4.7kΩ)の取り付け

部品番号	名称	型式	写真	メーカー	数量
R1, 2, 4, 5	抵抗	CFS1/4C 4.7kΩ (黄・紫・赤・金)		コーア(株)	4

抵抗(4.7kΩ)を半田付けします。



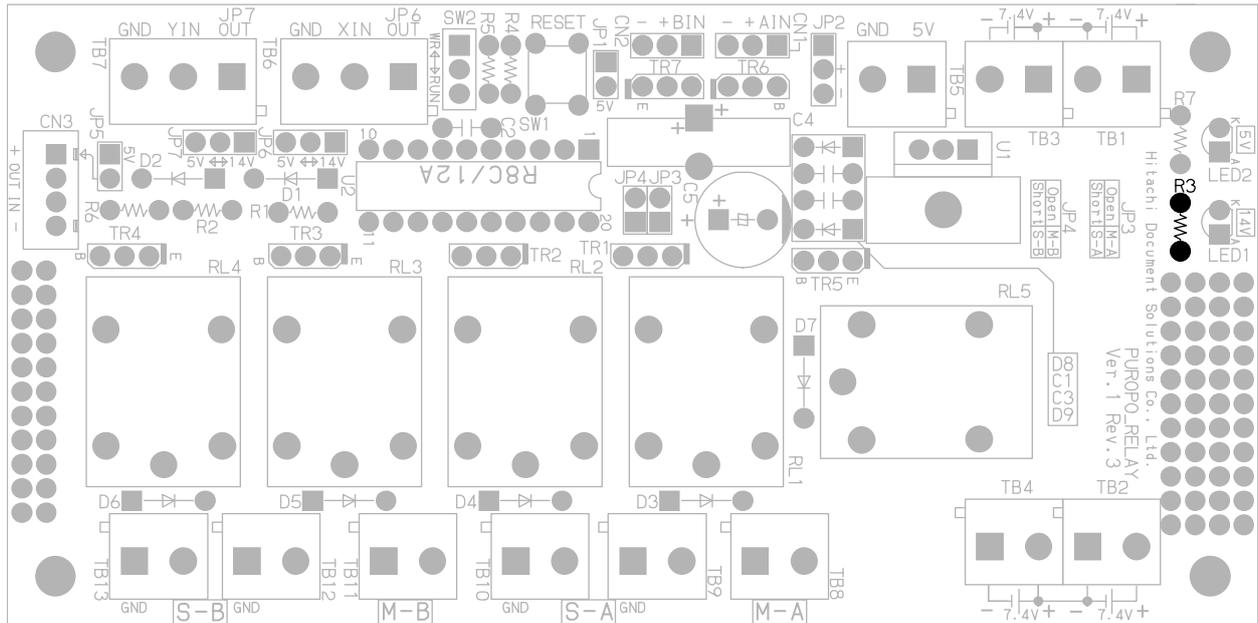
1		抵抗を写真のように曲げ加工します。リードの間隔は、約 5mm にします。 抵抗の根元からリードを曲げると、約 5mm になります。
2		抵抗の実装方法は、基板の左下を原点として、原点に近い方を1色目にして半田付けします。

3. プロポリレー基板の組み立て

3.5 抵抗(3.3kΩ)の取り付け

部品番号	名称	型式	写真	メーカー	数量
R3	抵抗	CFS1/4C 3.3kΩ (橙・橙・赤・金)		コーア(株)	1

抵抗(3.3kΩ)を半田付けします。

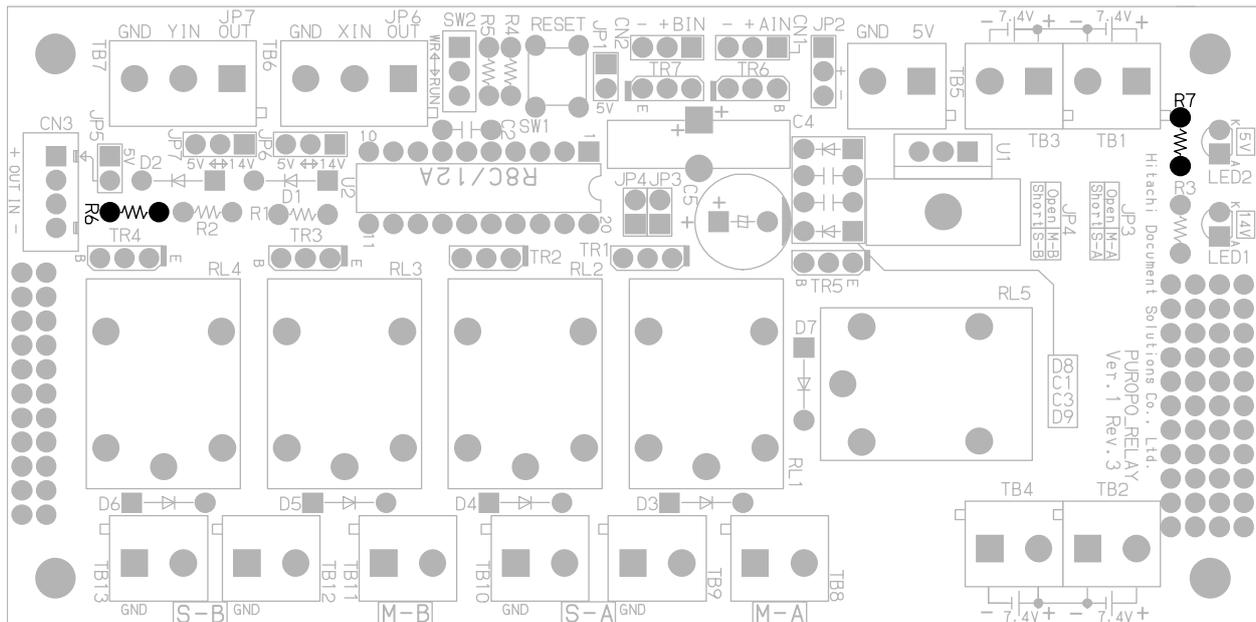


3. プロボリレー基板の組み立て

3.6 抵抗(1kΩ)の取り付け

部品番号	名称	型式	写真	メーカー	数量
R6, 7	抵抗	CFS1/4C 1kΩ (茶・黒・赤・金)		コーア(株)	2

抵抗(1kΩ)を半田付けします。

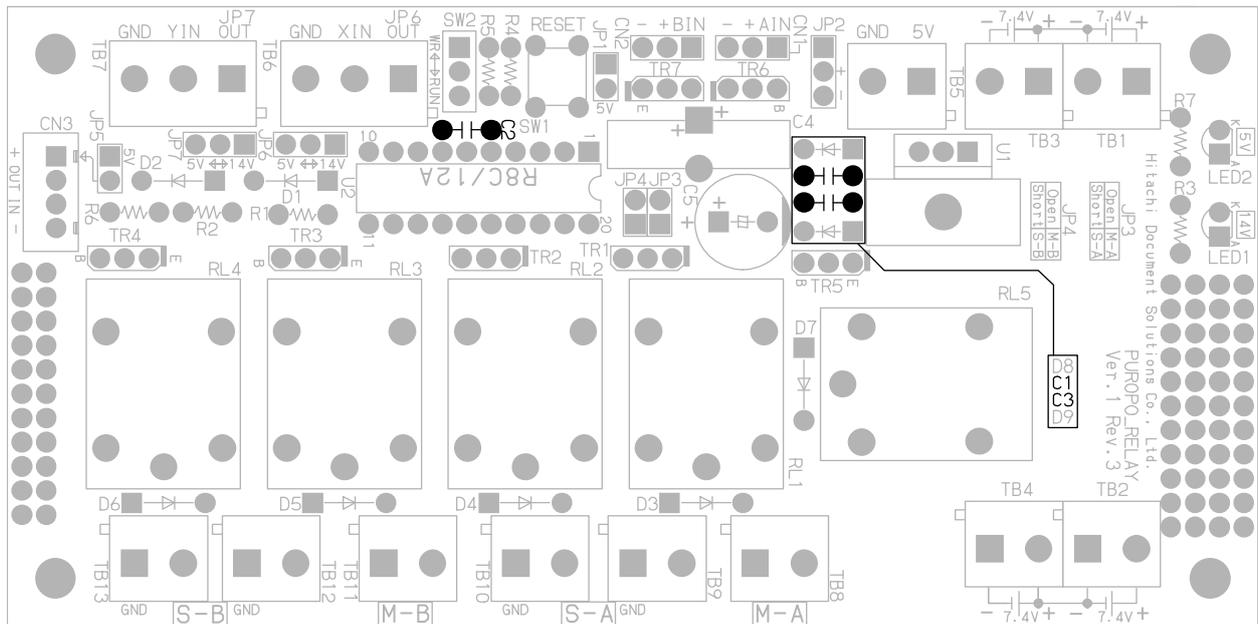


3. プロポリレー基板の組み立て

3.7 積層セラミックコンデンサ(1 μ F)の取り付け

部品番号	名称	型式	写真	メーカー	数量
C1, 2, 3	積層セラミックコンデンサ	RDER71H105K2K1H03B 1 μ F/50V 5.08mm ピッチ		(株)村田製作所	3

積層セラミックコンデンサ(1 μ F)を半田付けします。取り付け向きはありません。

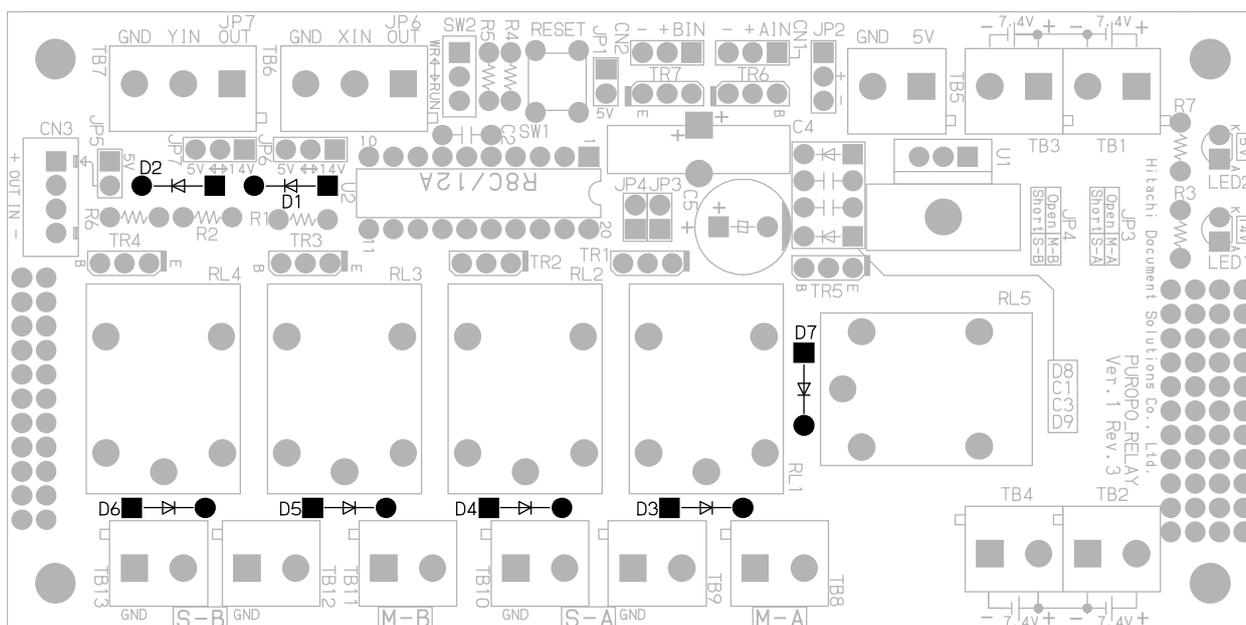


3. プロボリレー基板の組み立て

3.8 汎用整流用ダイオード(1N4007)の取り付け

部品番号	名称	型式	写真	メーカー	数量
D1, 2, 3, 4, 5, 6, 7	汎用整流用ダイオード	1N4007 1000V/1A		PANJIT INTERNATIONAL INC.	7

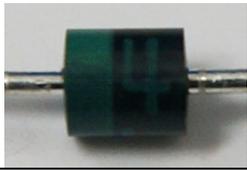
汎用整流用ダイオード(1N4007)を半田付けします。向きに合わせて取り付けてください。



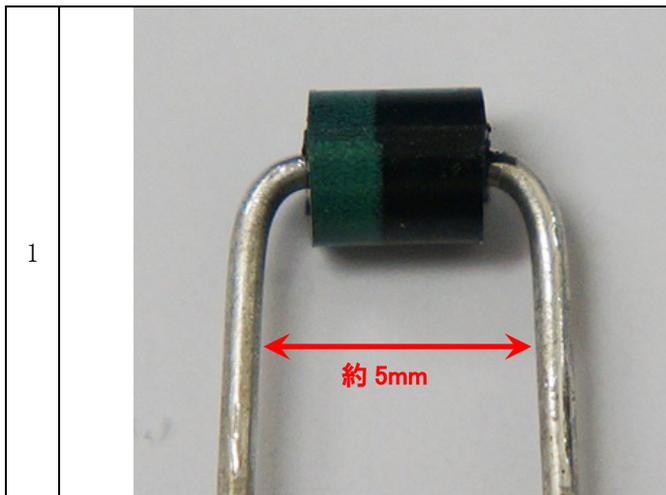
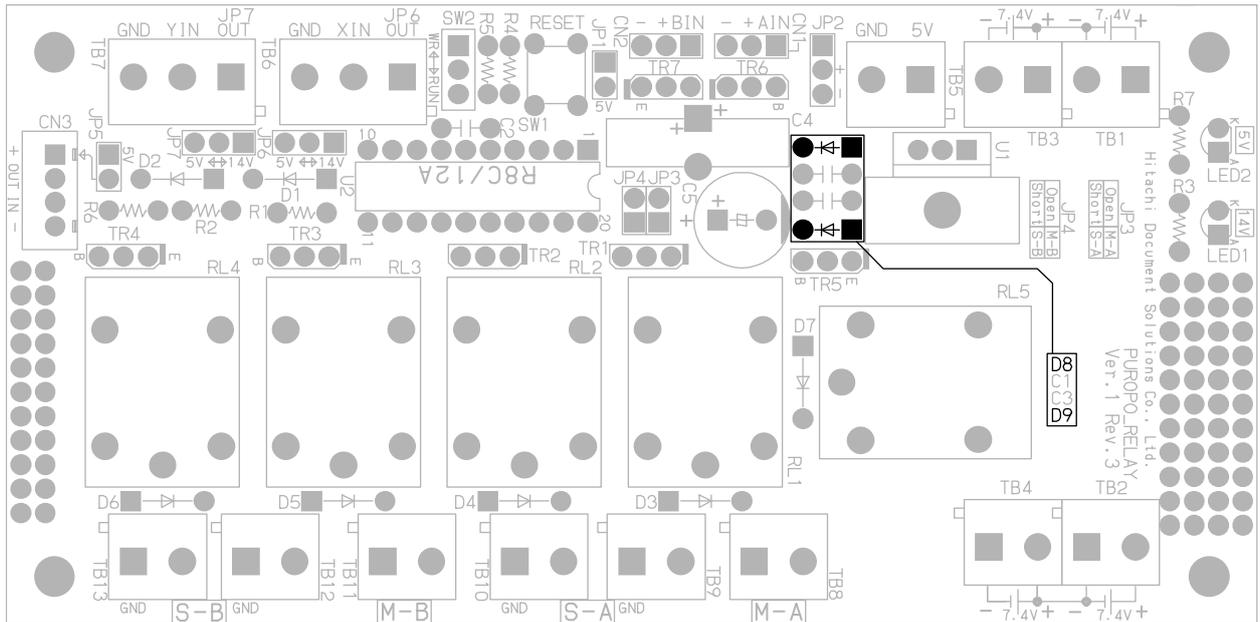
ダイオードを写真のように曲げ加工します。リードの間隔は、約 7.5mm にします。ダイオードの根元からリードを曲げると、約 7.5mm になります。

3. プロボリレー基板の組み立て

3.9 ショットキーバリアダイオード(11EQS04)の取り付け

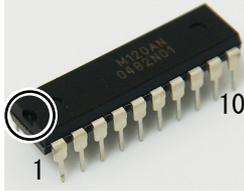
部品番号	名称	型式	写真	メーカー	数量
D8, 9	ショットキーバリアダイオード	11EQS04 40V/1A V _{FM} (せん頭順電圧):0.55V	<p>K(印) A</p> 	日本インター(株)	2

ショットキーバリアダイオード(11EQS04)を半田付けします。向きに合わせて取り付けてください。

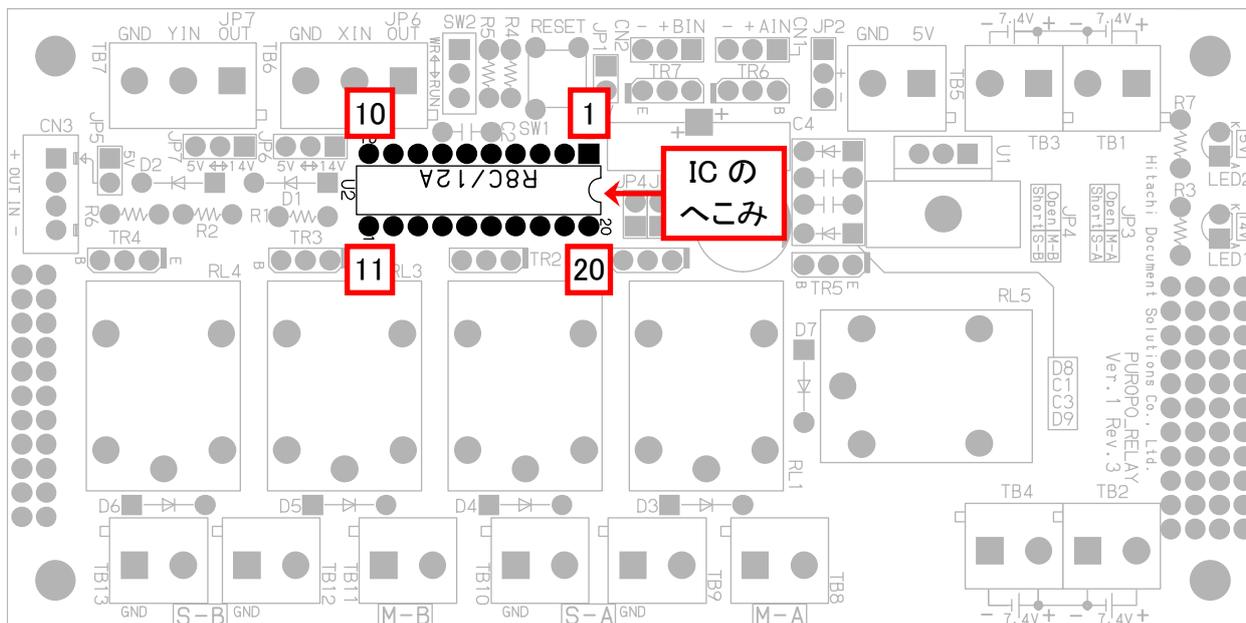


ダイオードを写真のように曲げ加工します。リードの間隔は、約 5mm にします。ダイオードの根元からリードを曲げると、約 5mm になります。

3.10 R8C マイコン(M120AN)の取り付け

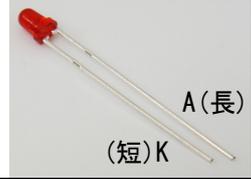
部品番号	名称	型式	写真	メーカー	数量
U2	R8C マイコン	R8C/M12A M120AN	 0のへこみがある側の手前が1ピン	ルネサス エレ クトロニクス(株)	1

R8C マイコン(M120AN)を半田付けします。向きに合わせて取り付けてください。

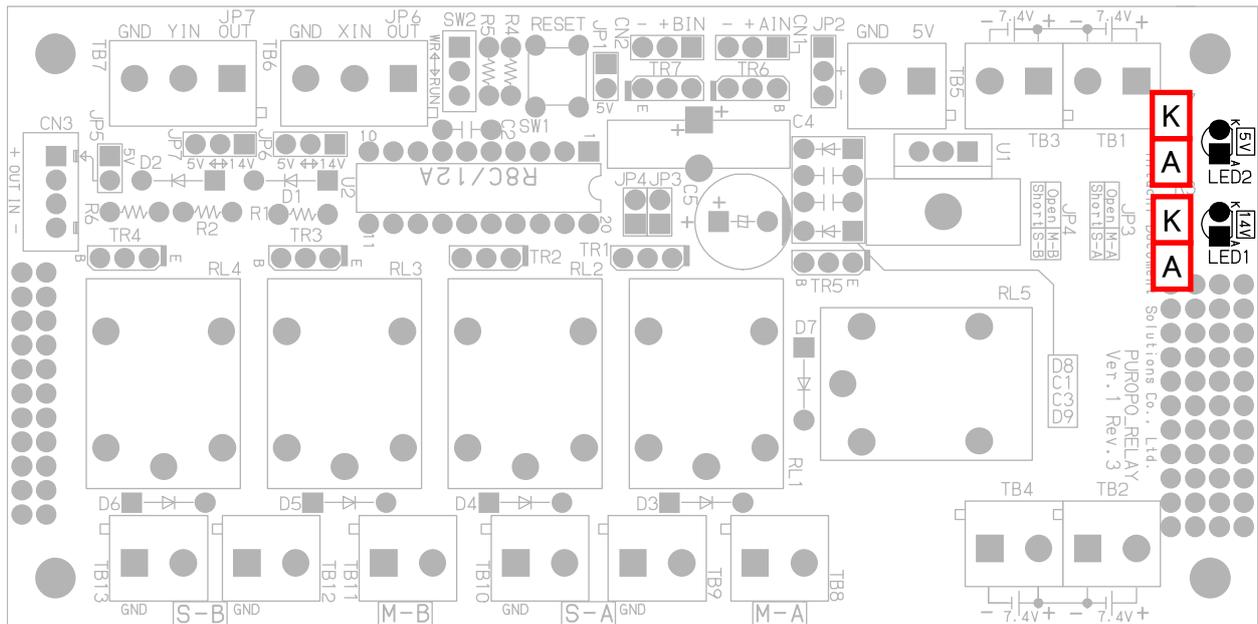


3. プロポリレー基板の組み立て

3.11 LED(EBR3338S)の取り付け

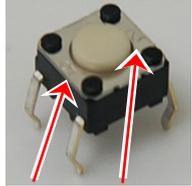
部品番号	名称	型式	写真	メーカー	数量
LED1,2	LED	EBR3338S φ3mm・赤		スタンレー電気 (株)	2

LED(EBR3338S)を半田付けします。向きに合わせて取り付けてください。



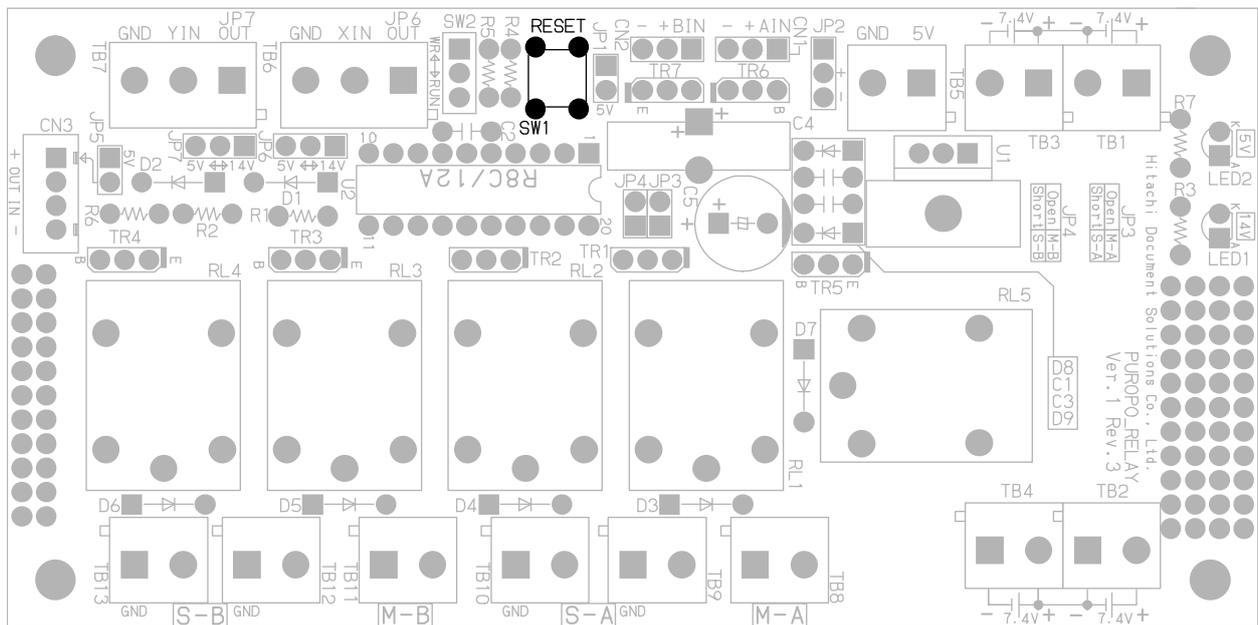
3. プロボリレー基板の組み立て

3.12 タクトスイッチ(B3F-1000)の取り付け

部品番号	名称	型式	写真	メーカー	数量
SW1	タクトスイッチ	B3F-1000	 <p>この数字は関係ありません</p>	オムロン(株)	1

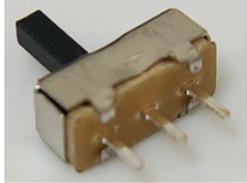
タクトスイッチ(B3F-1000)を半田付けします。取り付け向きはありません。

※スイッチに書いてある数字は、生産ロットです。向きとは関係ありません。

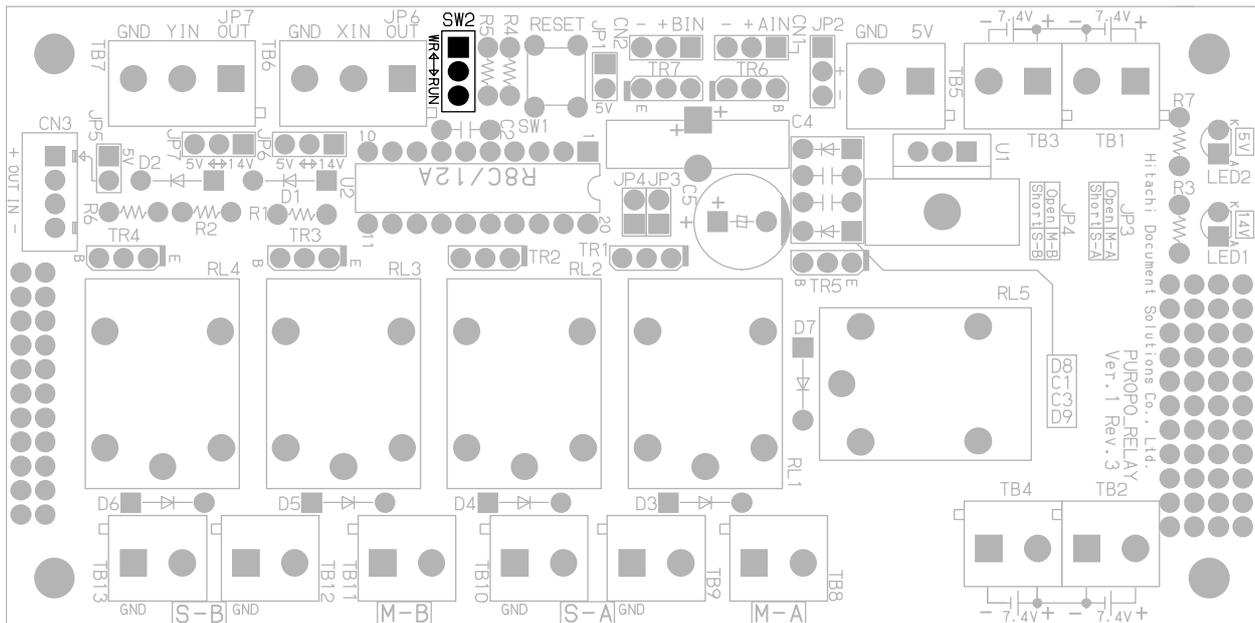


3. プロポリレー基板の組み立て

3.13 スライドスイッチ(SS-12D00-G5)の取り付け

部品番号	名称	型式	写真	メーカー	数量
SW2	スライドスイッチ	SS-12D00-G5		Zhejiang Jianfu Electronics Co., Ltd	1

スライドスイッチ(SS-12D00-G5)を半田付けします。向きはありません。

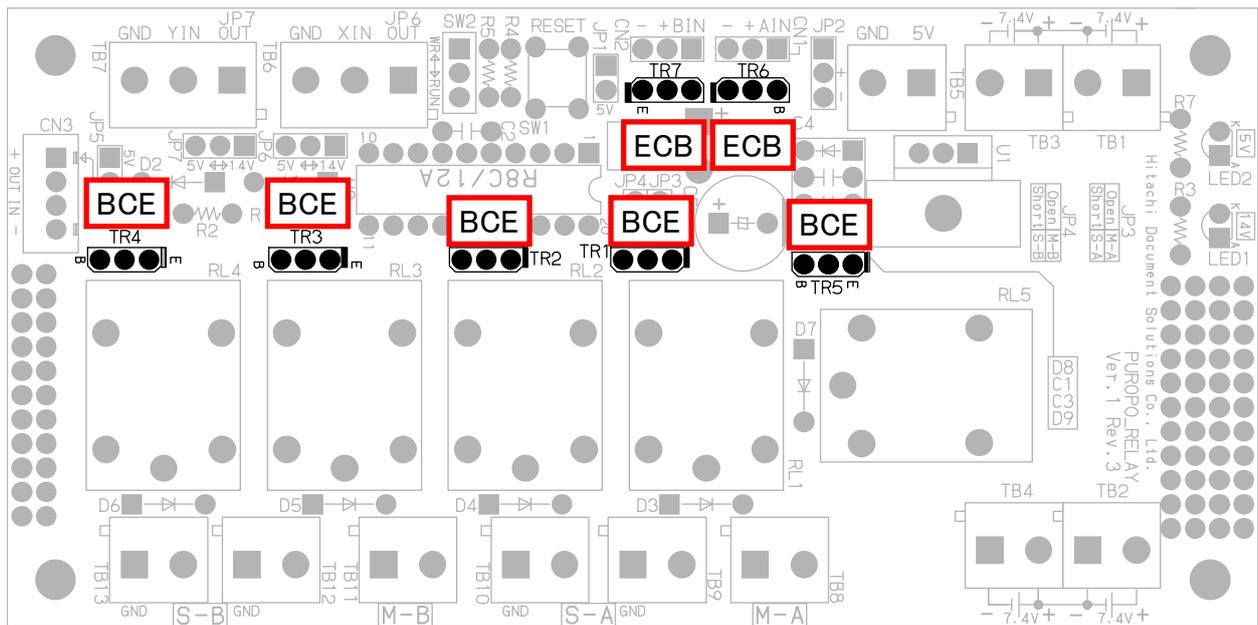


3. プロポリレー基板の組み立て

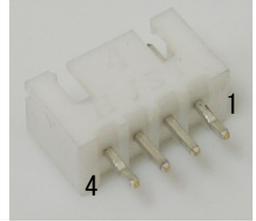
3.14 抵抗内蔵トランジスタ(RN1201)の取り付け

部品番号	名称	型式	写真	メーカー	数量
TR1,2,3,4, 5,6,7	抵抗内蔵トランジスタ	RN1201		(株)東芝セミコンダクター社	7

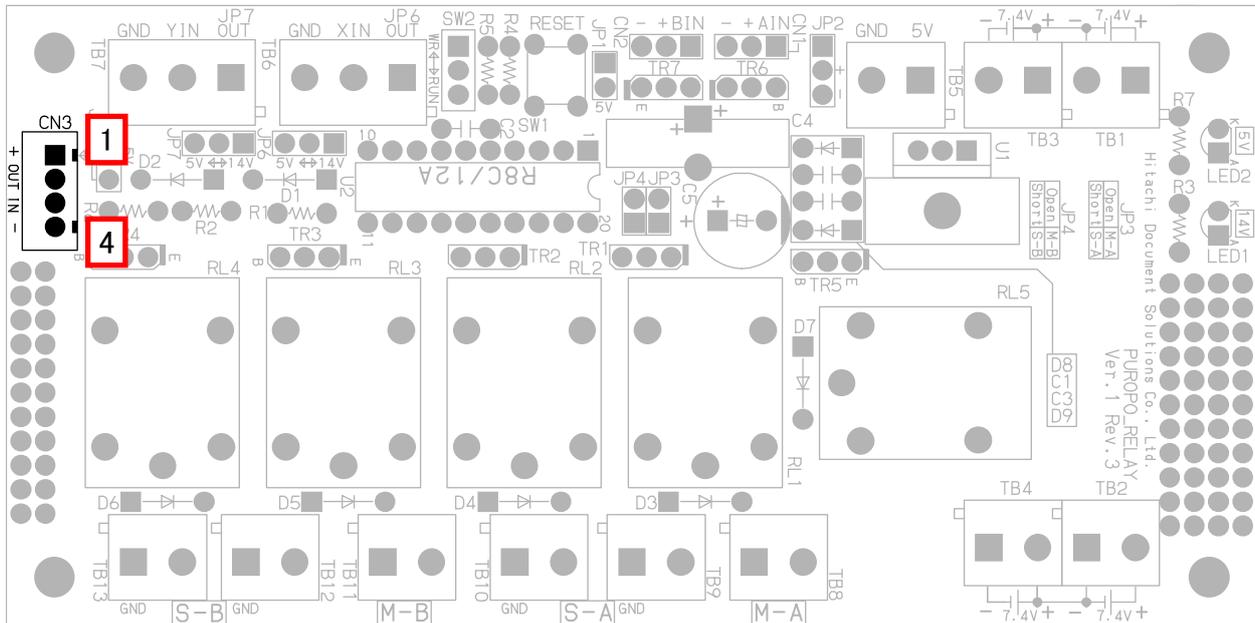
抵抗内蔵トランジスタ(RN1201)を半田付けします。向きに合わせて取り付けてください。



3.15 XH コネクタ 4ピンオス(B4B-XH-A)の取り付け

部品番号	名称	型式	写真	メーカー	数量
CN3	XHコネクタ4ピンオス	B4B-XH-A		日本圧着端子製造(株)	1

XHコネクタ4ピンオス(B4B-XH-A)を半田付けします。向きに合わせて取り付けてください。



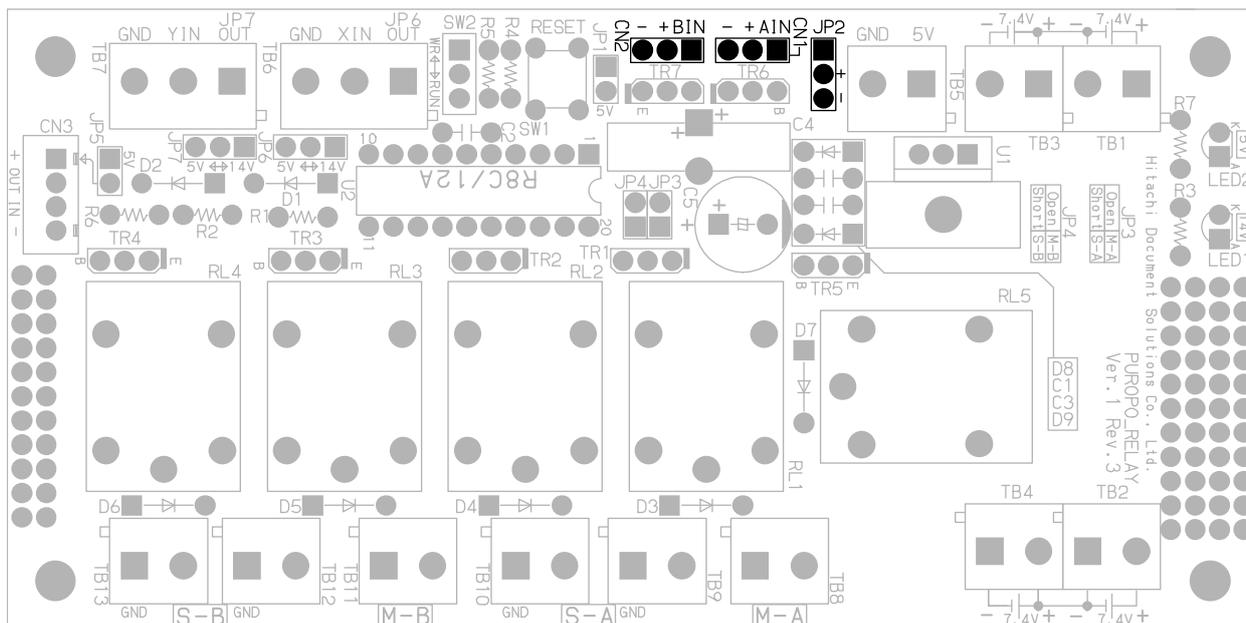
3. プロポリレー基板の組み立て

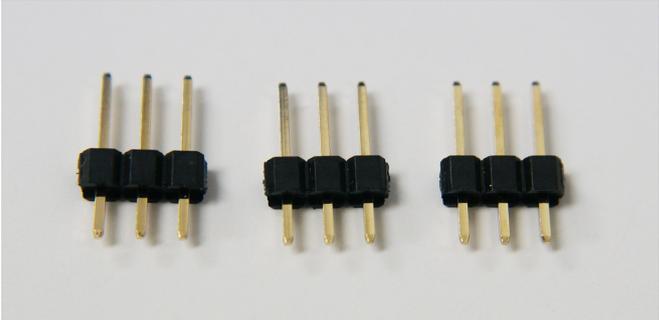
3.16 ピンヘッダ(1列×40P)の取り付け

部品番号	名称	型式	写真	メーカー	数量
CN1, 2, JP4	ピンヘッダ	1列×40P ※カットして使用します		(株)秋月電子通商	1

ピンヘッダ(1列×40P)を半田付けします。

1列×40Pのピンヘッダを3ピンにカットして、CN1、CN2、JP2 に実装してください。取り付け向きはありません。



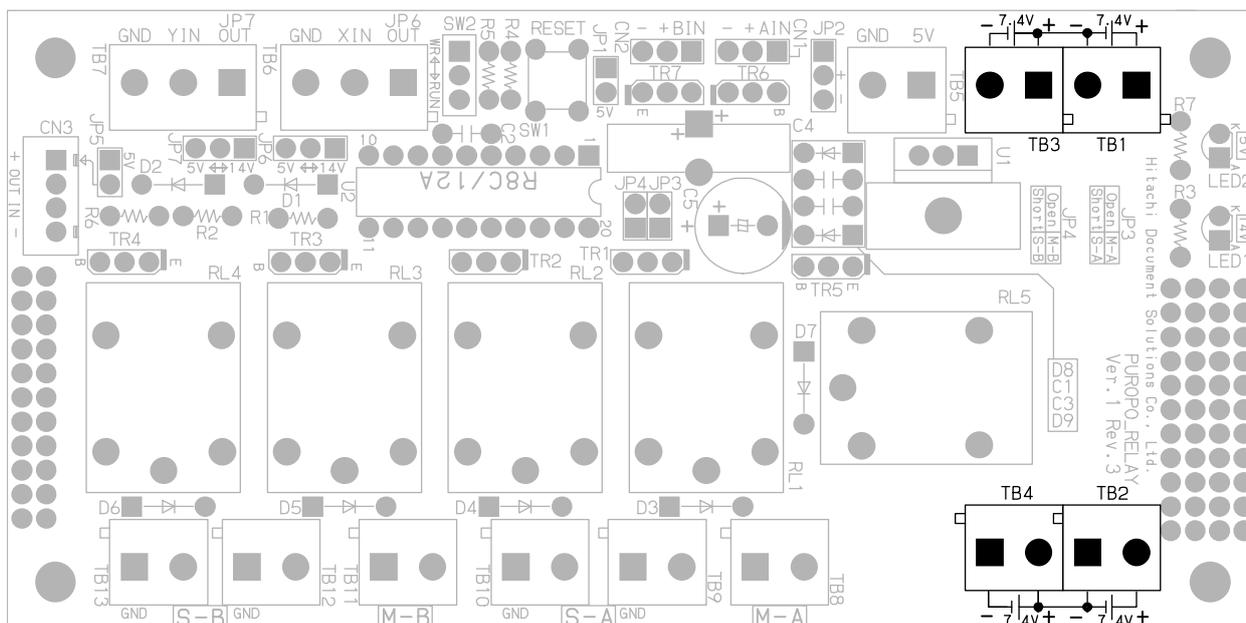
1		<p>1列×40Pのピンヘッダを3ピンにカットして、CN1、CN2、JP4 に実装してください。短いピンを、半田付けします。</p>
---	---	--

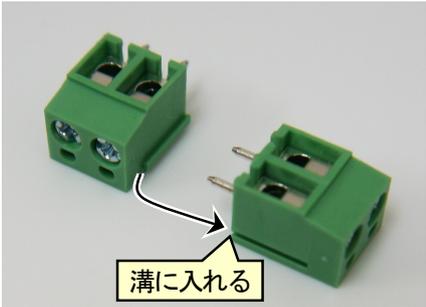
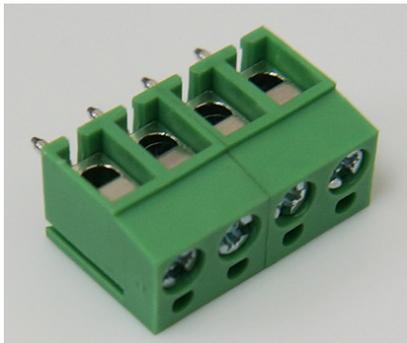
3. プロボリレー基板の組み立て

3.17 ターミナルブロック(2ピン・緑・縦)の取り付け

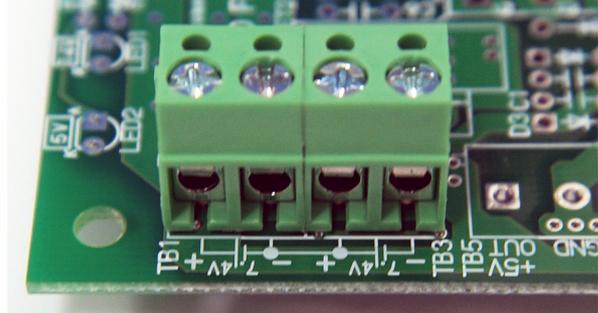
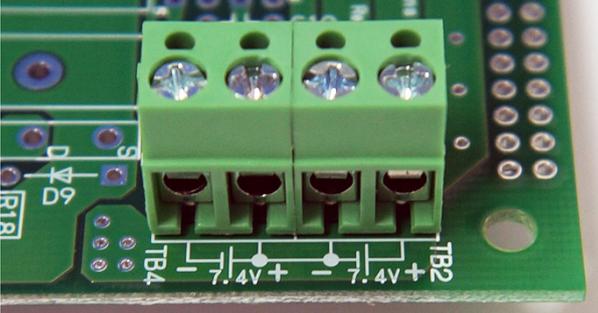
部品番号	名称	型式	写真	メーカー	数量
TB1, 2, 3, 4	ターミナルブロック 2ピン(緑)(縦)	TB112-2-2-E-1		Alphaplus Connectors & Cables Corp.	4

ターミナルブロック(2ピン・緑・縦)を半田付けします。向きに合わせて取り付けてください。

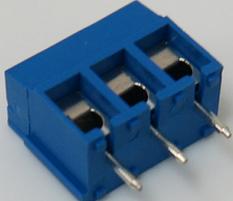


1		<p>TB1 と TB3 は、左写真のようにターミナルブロックの溝に差し込み、2 個つながっている状態で実装します。 TB2 と TB4 も同様です。</p>
2		<p>溝に入れたところです。</p>

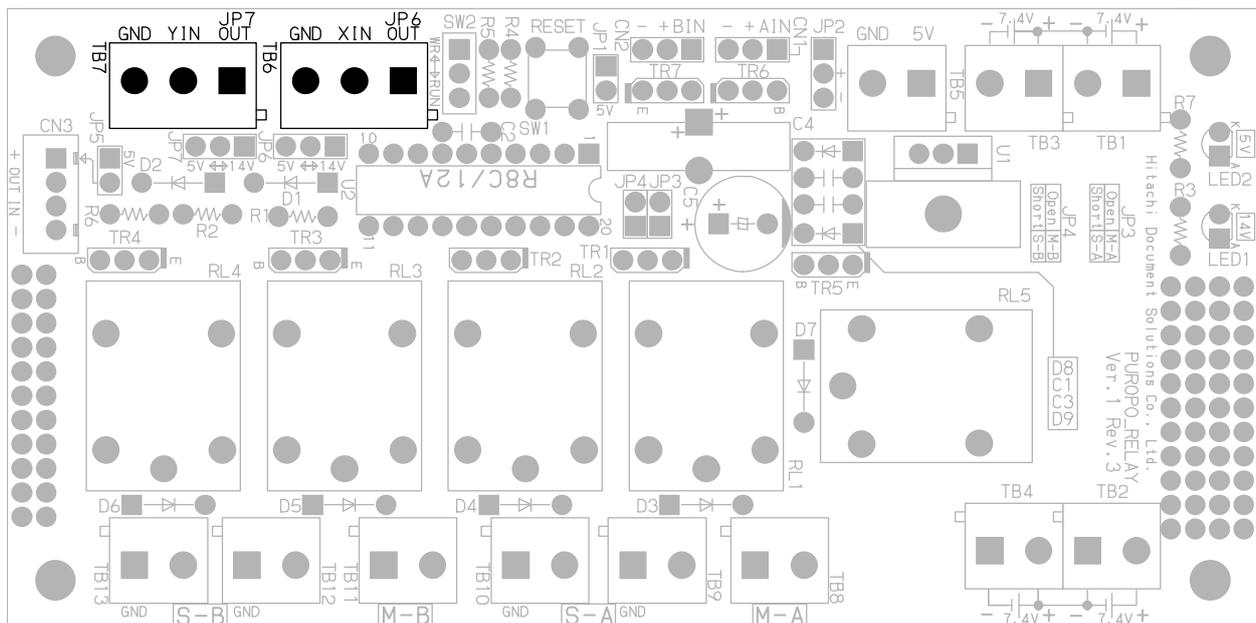
3. プロポリレー基板の組み立て

3		TB1、TB3 を実装したところです。
4		TB2、TB4 を実装したところです。

3.18 ターミナルブロック(3ピン・青・縦)の取り付け

部品番号	名称	型式	写真	メーカー	数量
TB6, 7	ターミナルブロック 3ピン(青)(縦)	TB112-2-3-U-1		Alphaplus Connectors & Cables Corp.	2

ターミナルブロック(3ピン・青・縦)を半田付けします。向きに合わせて取り付けてください。

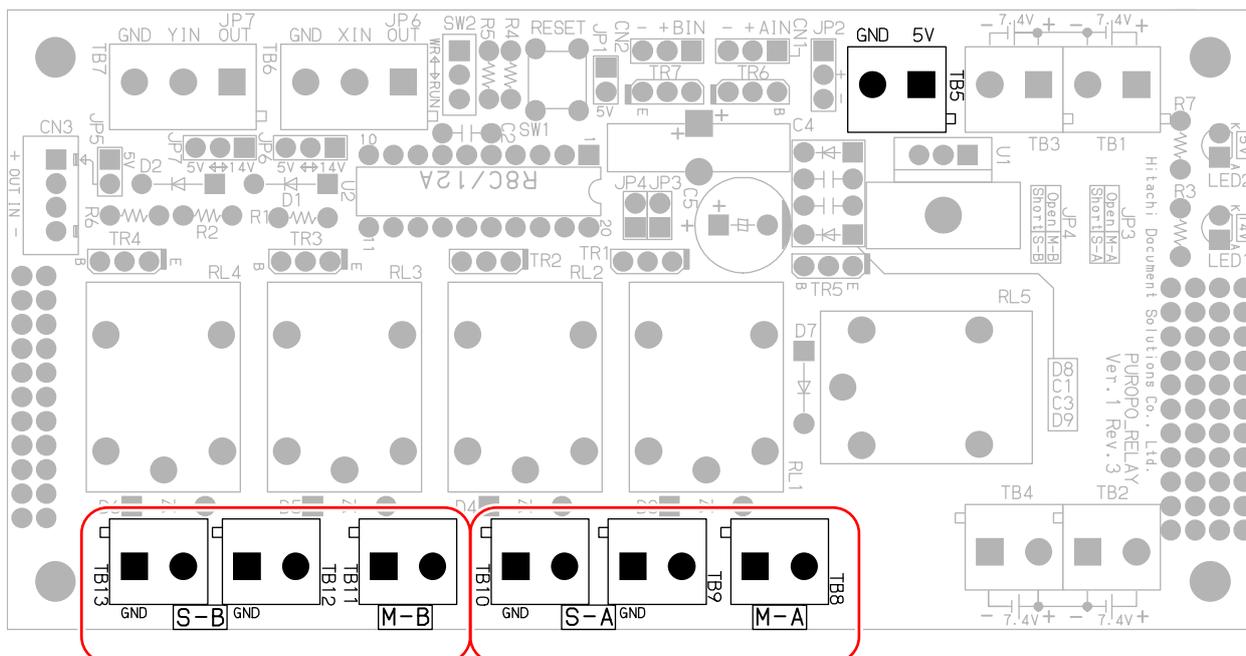


3. プロボリレー基板の組み立て

3.19 ターミナルブロック(2ピン・青・縦)の取り付け

部品番号	名称	型式	写真	メーカー	数量
●共通 TB5	ターミナルブ ック 2ピン(青)(縦)	TB112-2-2-U-1		Alphaplus Connectors & Cables Corp.	5
●チャンネル A TB8, または TB9,10					
●チャンネル B TB11, または TB12,13					
※チャンネル A とチャンネル B の実装位置 は、後述します					

ターミナルブロック(2ピン・青・縦)を半田付けします。向きに合わせて取り付けてください。



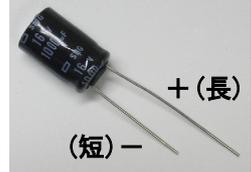
チャンネルBで何を制御するかによって、実装位置が異なります。

- DC モータを使う場合は TB11 に実装してください。
- ソレノイドバルブを使うときは、TB12、TB13 に実装してください。

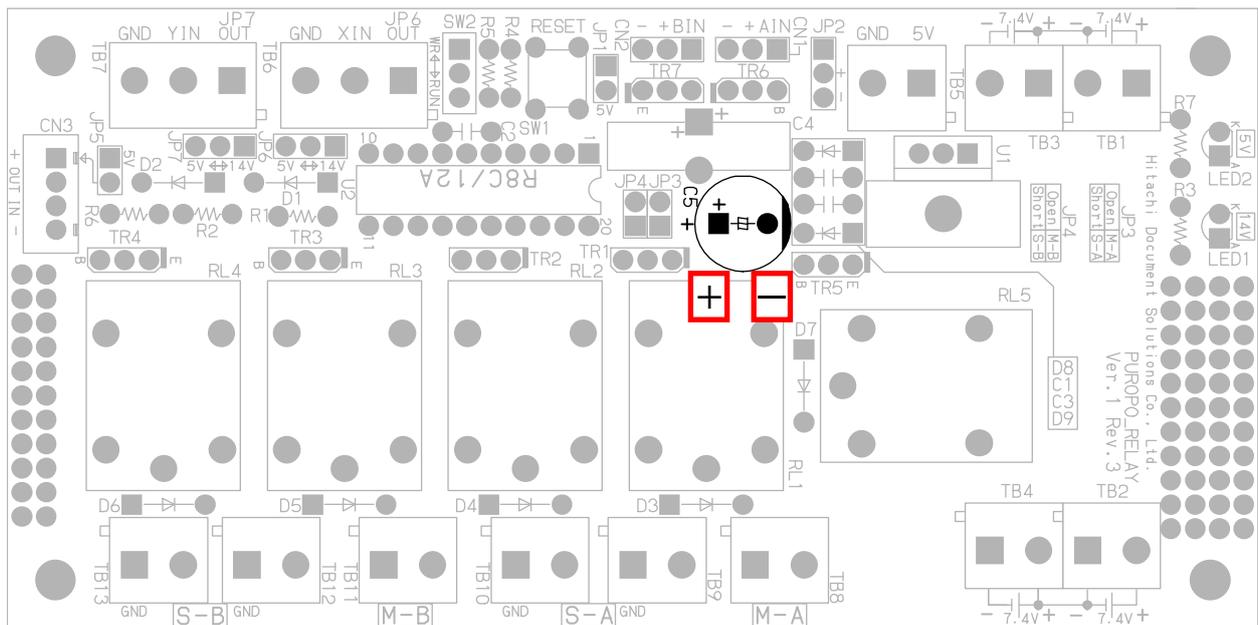
チャンネルAで何を制御するかによって、実装位置が異なります。

- DC モータを使う場合は TB8 に実装してください。
- ソレノイドバルブを使うときは、TB9、TB10 に実装してください。

3.20 電解コンデンサ(1000 μ F)の取り付け

部品番号	名称	型式	写真	メーカー	数量
C5	電解コンデンサ	SMG160E102MJ16S 1000 μ F/16V		日本ケミコン (株)	1

電解コンデンサ(1000 μ F)を半田付けします。向きに合わせて取り付けてください。



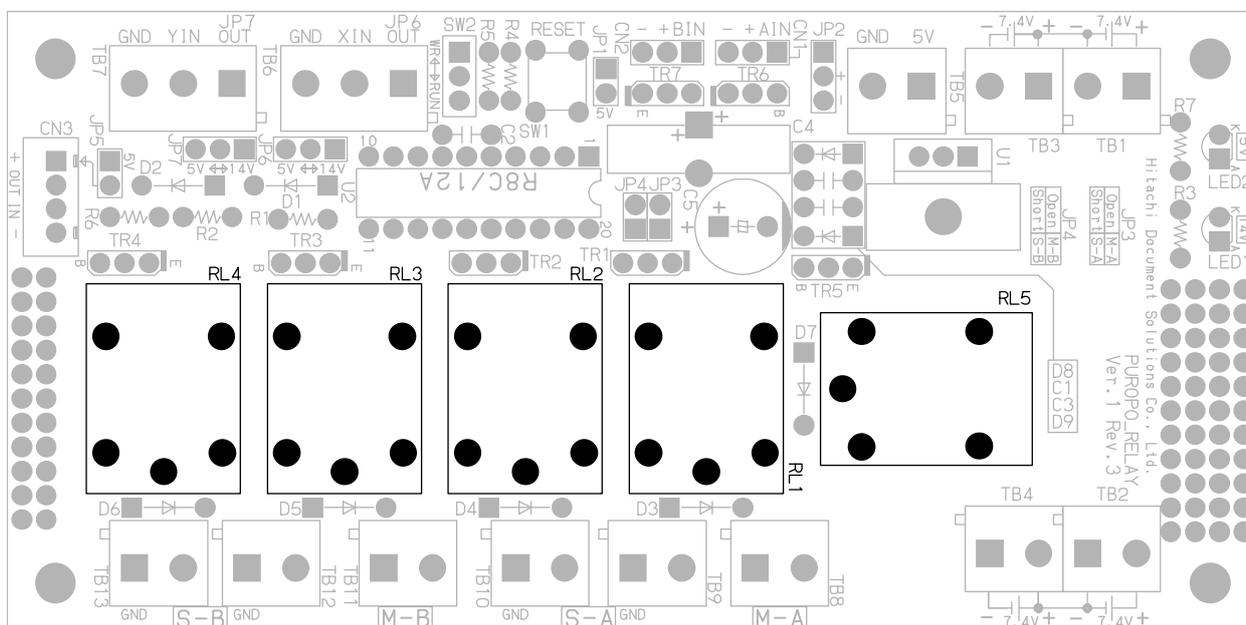
3. プロポリレー基板の組み立て

3.21 リレーの取り付け

部品番号	名称	型式	写真	メーカー	数量
RL1, 2, 3, 4, 5	リレー	JSM1-12V-5 ・定格制御容量(抵抗負荷) 15A16V DC ・接点最大通電電流(20度時) 25A/2分間 15A/1時間		パナソニック電工(株)	5

※ソレノイドバルブを 1 個以上使う場合(JP3、または JP4 をショートしている場合)、RL5 にリレーは実装せず、線を1本、ショートさせます。詳しくは、「3.3 JP3、JP4 の半田付け」を参照してください。

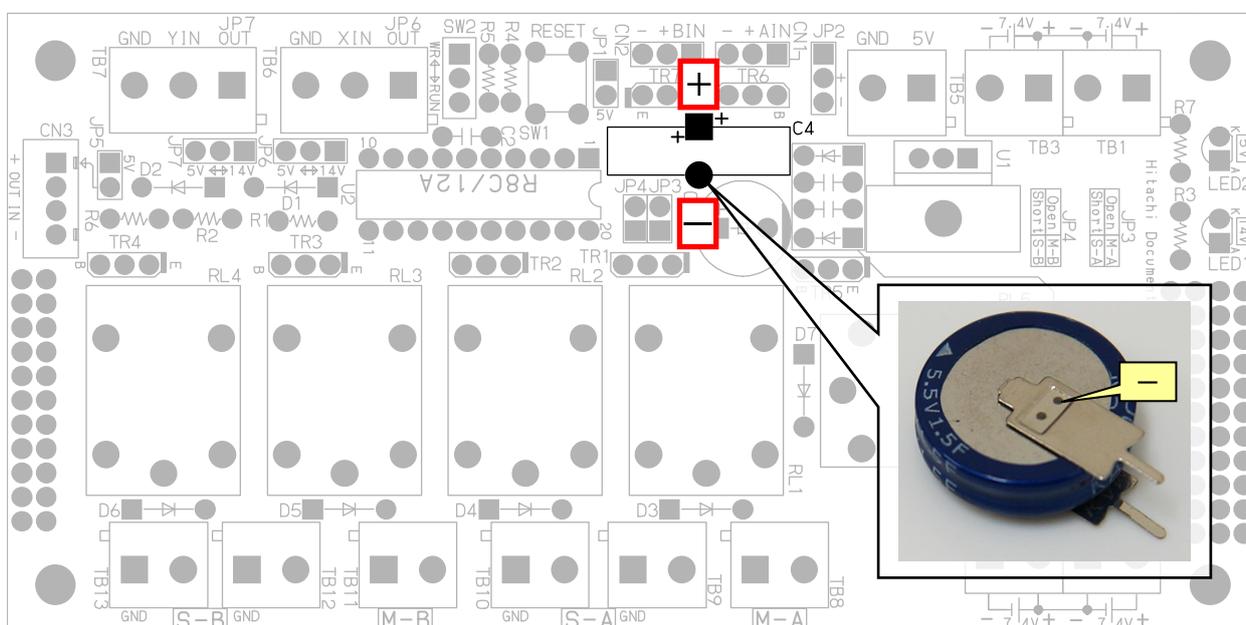
リレーを半田付けします。向きに合わせて取り付けてください。



3.22 電気二重層コンデンサ(1.5F)の取り付け

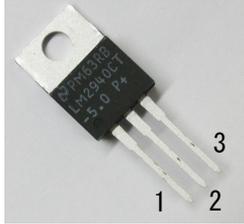
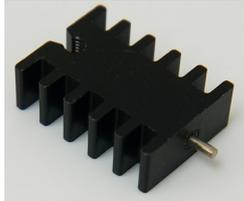
部品番号	名称	型式	写真	メーカー	数量
C4	電気二重層コンデンサ	SE-5R5-D155VYV 1.5F/5.5V(タテ型) ※プラス、マイナスは、 スポット溶接部分の 形で判断します		KAMCAP	1

電気二重層コンデンサ(1.5F)を半田付けします。向きに合わせて取り付けてください。

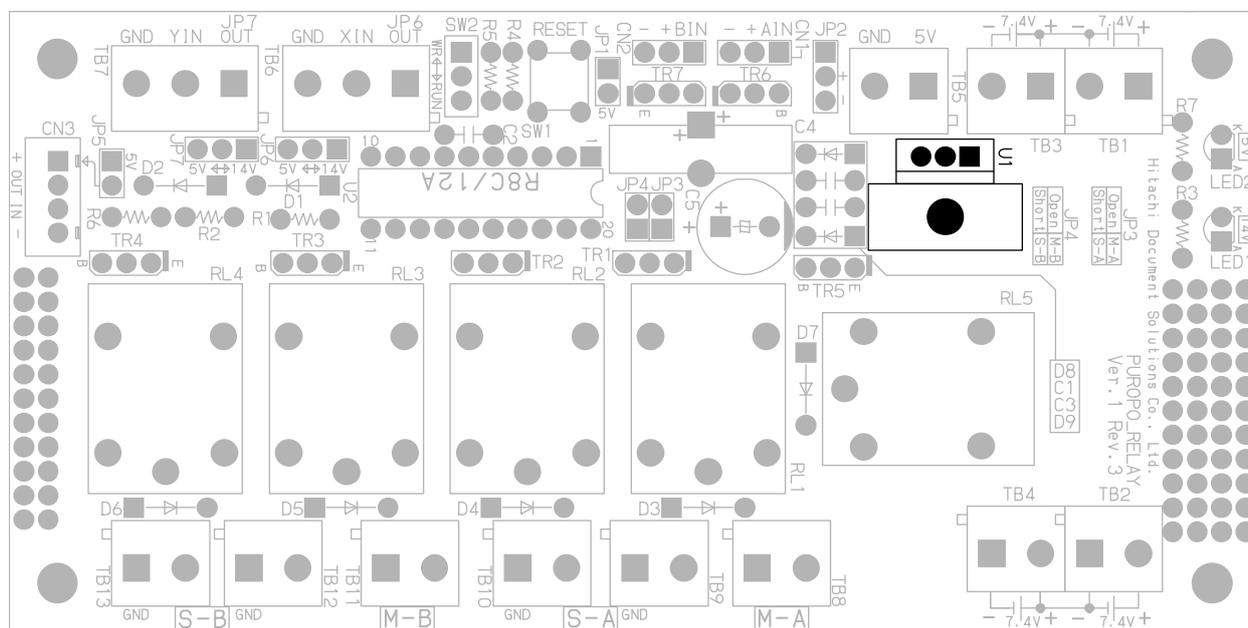


3. プロポリレー基板の組み立て

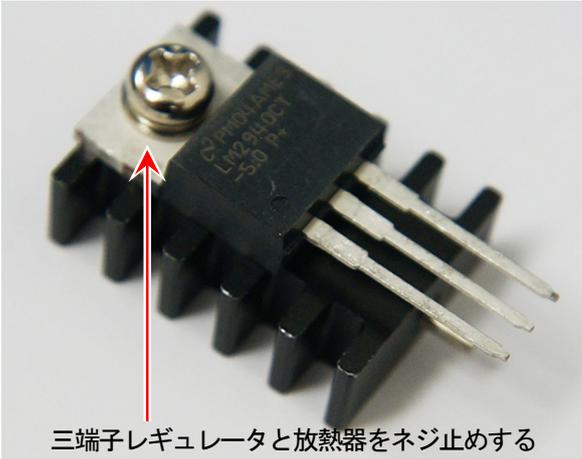
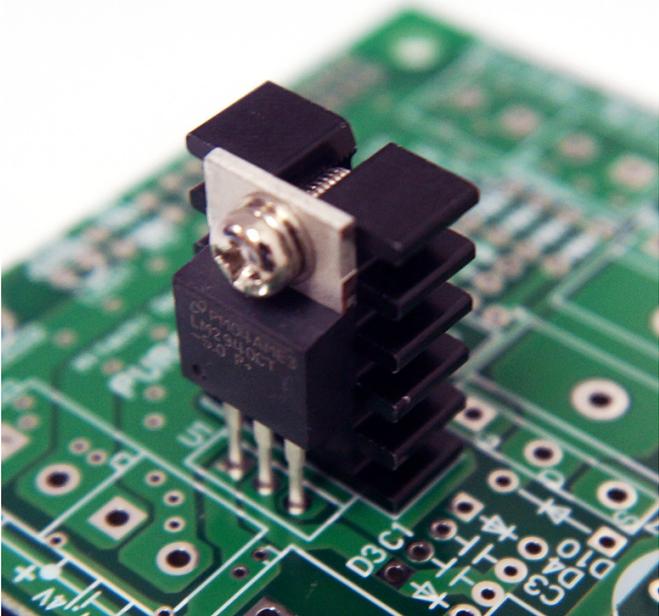
3.23 三端子レギュレータ(LM2940CT-5.0)の取り付け

部品番号	名称	型式	写真	メーカー	数量
U1	三端子レギュレータ	LM2940CT-5.0		ナショナル セミコンダクター ジャパン(株)	1
U1 用	放熱器	PC2116-7-PB		(株)放熱器のオーエス	1
放熱器固定用	座金ネジ	φ3×8mm		各社	1

三端子レギュレータ(LM2940CT-5.0)を半田付けします。三端子レギュレータと放熱器をネジ止めしてから、半田付けしてください。



3. プロボリレー基板の組み立て

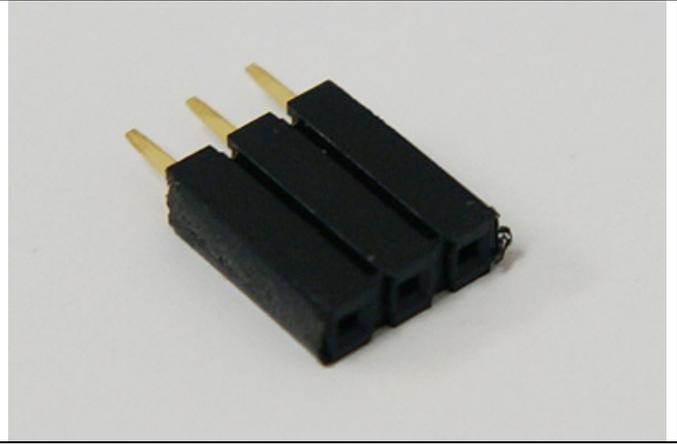
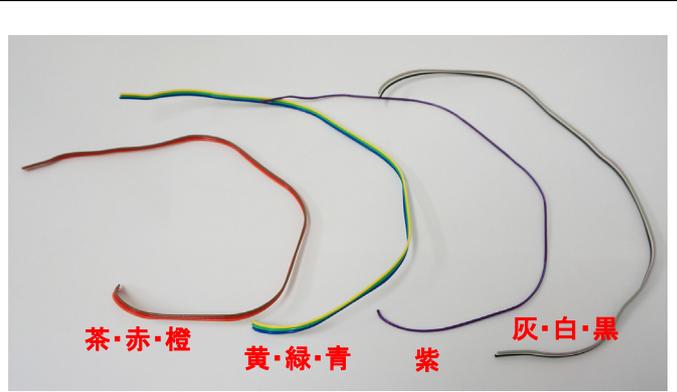
1	 <p>三端子レギュレータと放熱器をネジ止めする</p>	<p>三端子レギュレータ(LM2940CT-5.0)、放熱器、座金ネジを用意します。 左写真のように、三端子レギュレータを放熱器に座金ネジで固定します。 この状態で、U1 に実装してください。</p>
2		<p>実装したところです。 ※基板は、別な基板です</p>

3. プロポリレー基板の組み立て

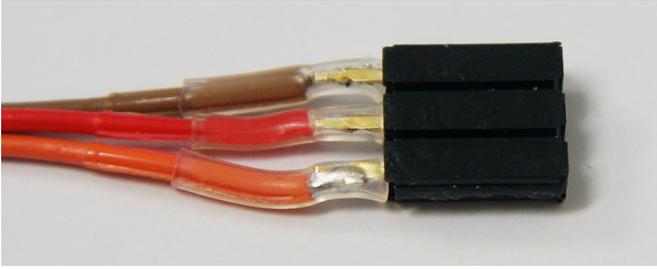
3.24 プロポ受信機接続ケーブルの作成

部品番号	名称	型式	写真	メーカー	数量
受信機 接続用	分割ロングピン ソケット	1列×42ピン		(株)秋月電子 通商	1
受信機 接続用	10芯スダレ フラット ケーブル	FLEX-S410-7/0.127 1.27mm ピッチ		沖電線(株)	0.6 m
ピン ソケット用	熱収縮 チューブ	スミチューブ C 透明 φ1		住友電工ファイ ンポリマー(株)	0.5 m

プロポ受信器と、プロポリレー基板をつなぐケーブルを3本作ります。

1		分割ロングピンソケットを3ピンでカットします。これを6個、作ります。カットした切り口は、ヤスリなどで慣らしておきましょう。
2		<p>10芯スダレフラットケーブルを必要な長さに切り、下記のように4つに裂きます。</p> <p>①茶・赤・橙の3色で裂きます ②黄・緑・青の3色で裂きます ③紫の1色で裂きます(使いません) ④灰・白・黒の3色で裂きます</p> <p>※必要な長さとは、プロポ受信器とプロポリレー基板との距離です。このケーブルは、プロポ受信器とプロポリレー基板をつなぐケーブルです。</p>

3. プロポリレー基板の組み立て

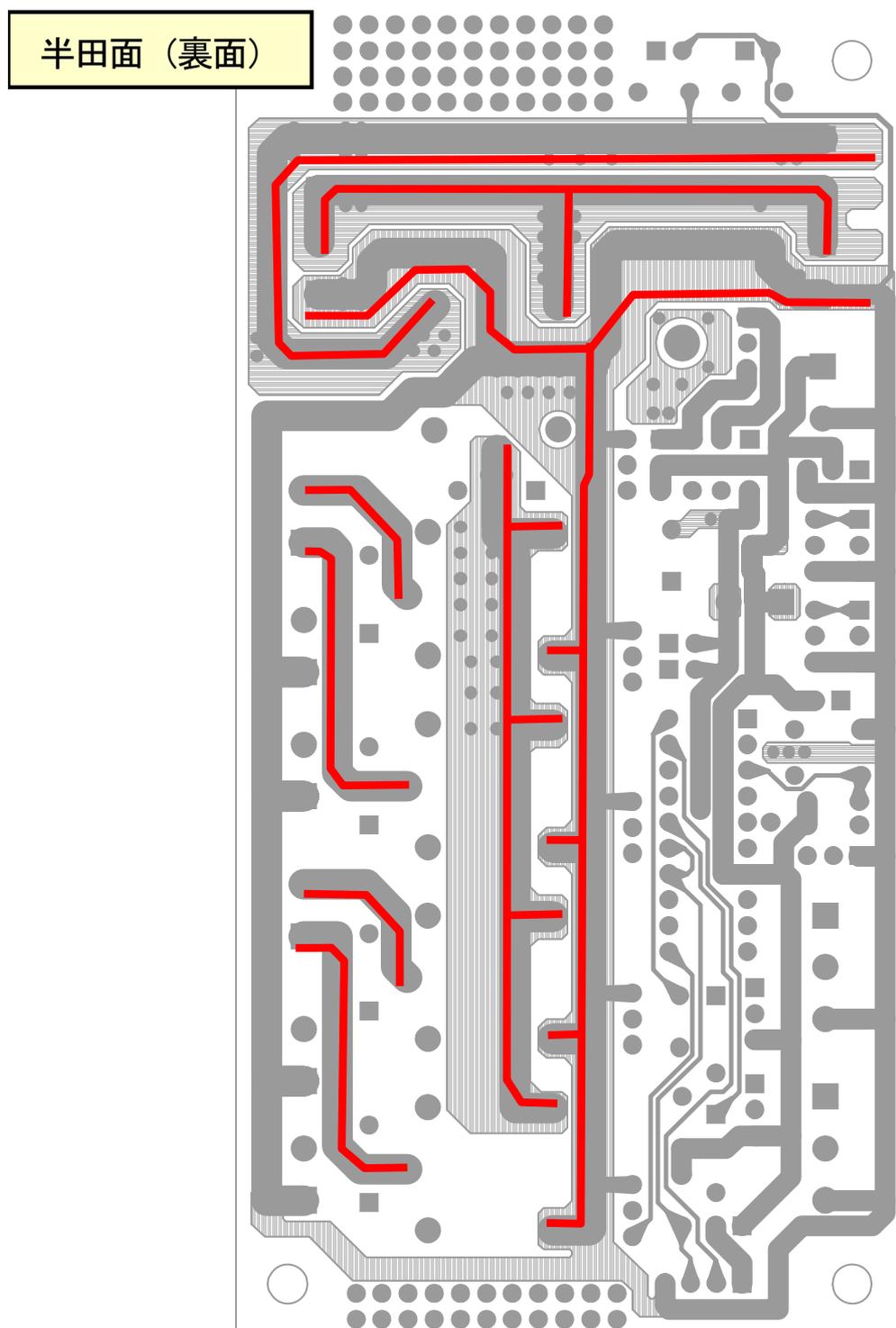
3		<p>分割ロングピンソケットと、フラットケーブルを半田付けします。半田付けした部分が、他の部分に触ったときにショートしないように熱収縮チューブをかぶせておきます。</p>
---	---	---

4		<p>同様に、逆側も分割ロングピンソケットを半田付けします。写真は、完成例です。</p> <p>これを、</p> <ul style="list-style-type: none">①茶・赤・橙の3色のケーブル②黄・緑・青の3色のケーブル④灰・白・黒の3色のケーブル <p>で3本、作っておきます。</p> <p>③紫のケーブル は使いません。</p>
---	---	--

3.25 パターンの強化

太線(赤色線)部分は、半田面のレジスト(緑色の部分)が無い部分を示しています。この部分に **0.5 スケア程度**の裸線と一緒に半田を盛って強化しておきます。強化しないと、大電流が流れたときに電流を流せず、パターンが燃えてしまいます。

※半田だけの場合、基板に強い力が加わると半田が割れることがあります。必ず裸線と一緒に半田付けしてください。

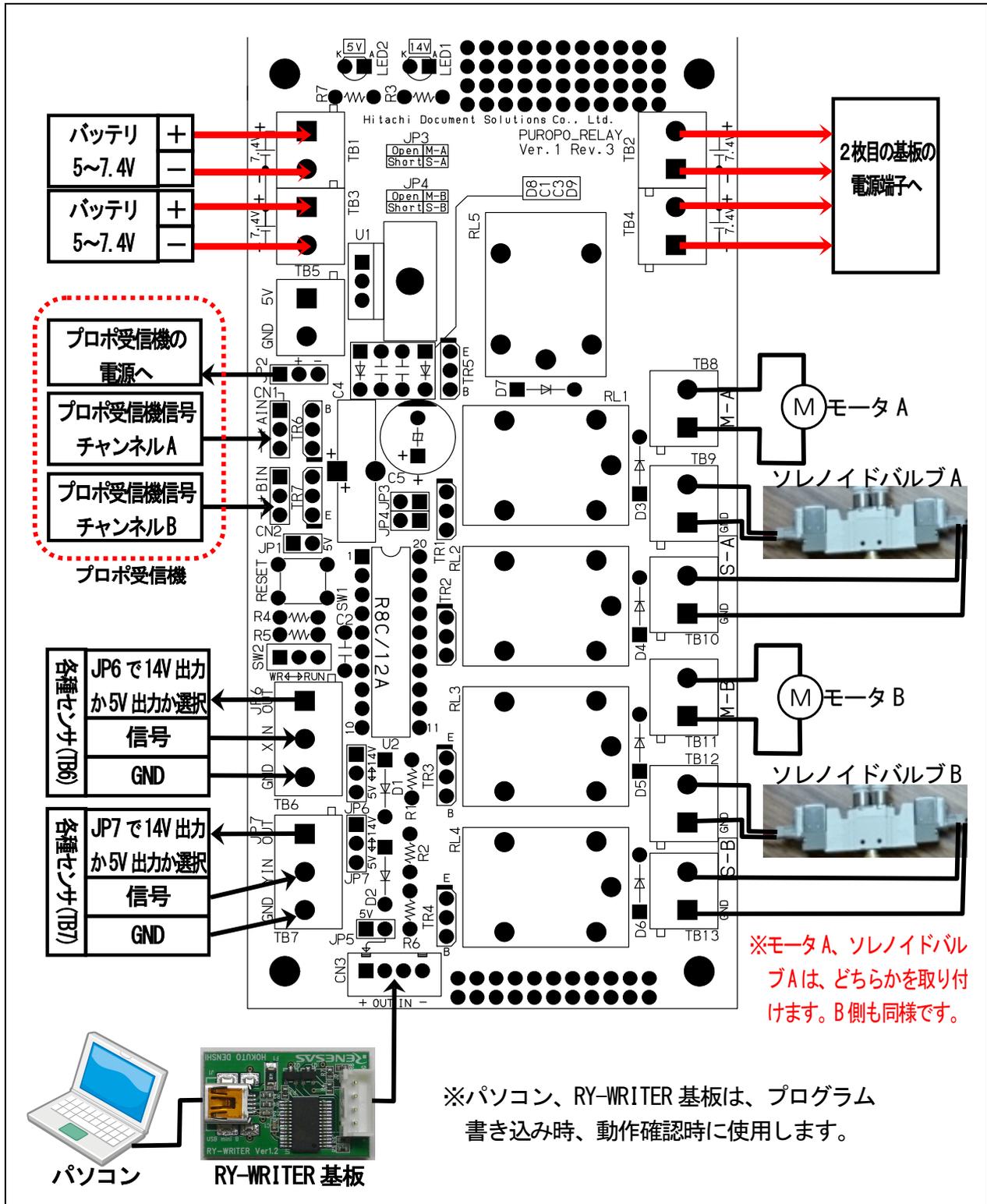


4. 接続

4. 接続

4.1 接続図

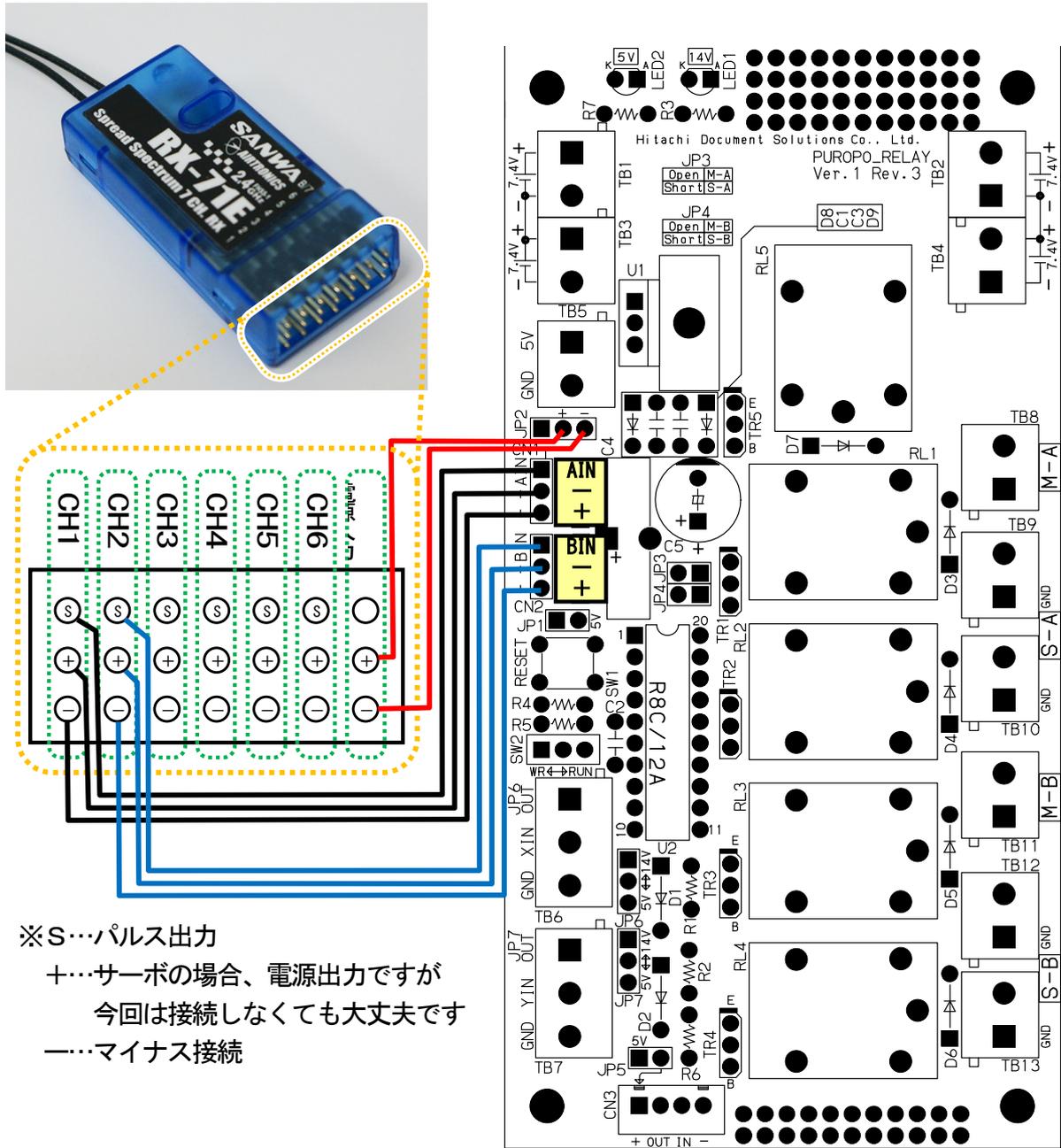
プロポリレー基板の接続構成を下記に示します。



4. 接続

プロポ受信器との接続例を、下図に示します。例として、サンワのRX-71Eというプロポからの信号を7チャンネル受信する受信器との接続例です。

プロポの操作とチャンネルの関係は、プロポの説明書を参照してください。



4. 接続

4.2 マイコンボードの電源を別にする

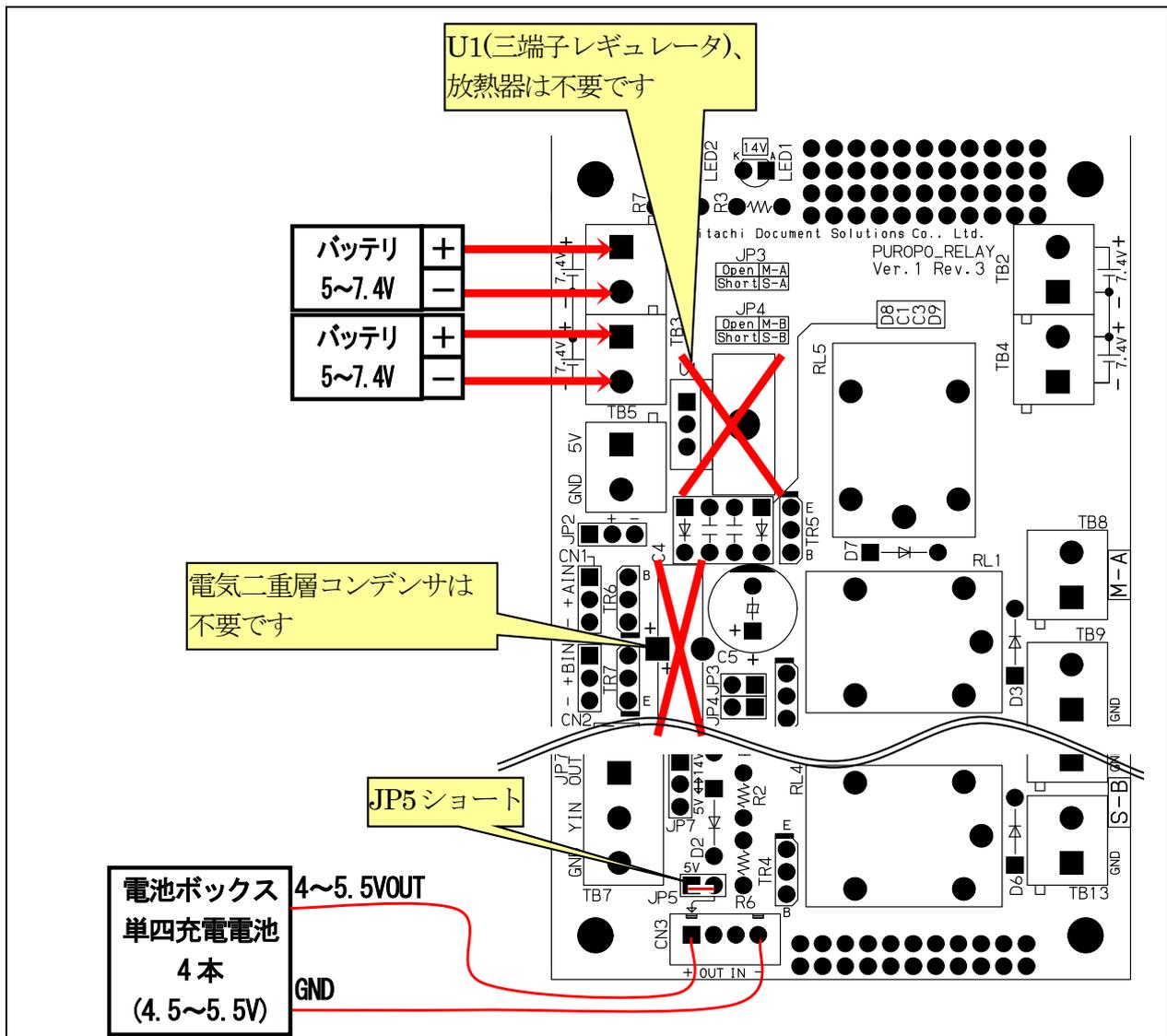
モータによる電圧降下が大いいと、マイコンへの供給電圧も低下してリセットしてしまい、ロボット制御できなくなります。この場合、マイコンの電源を別にして下さい。

別電源として、(株)秋月電子通商で販売している「電池ボックス 単4×4本用(フタ付プラスチック・スイッチ付)」がスイッチ付きでお勧めです。

<http://akizukidenshi.com/catalog/g/gP-00735/>



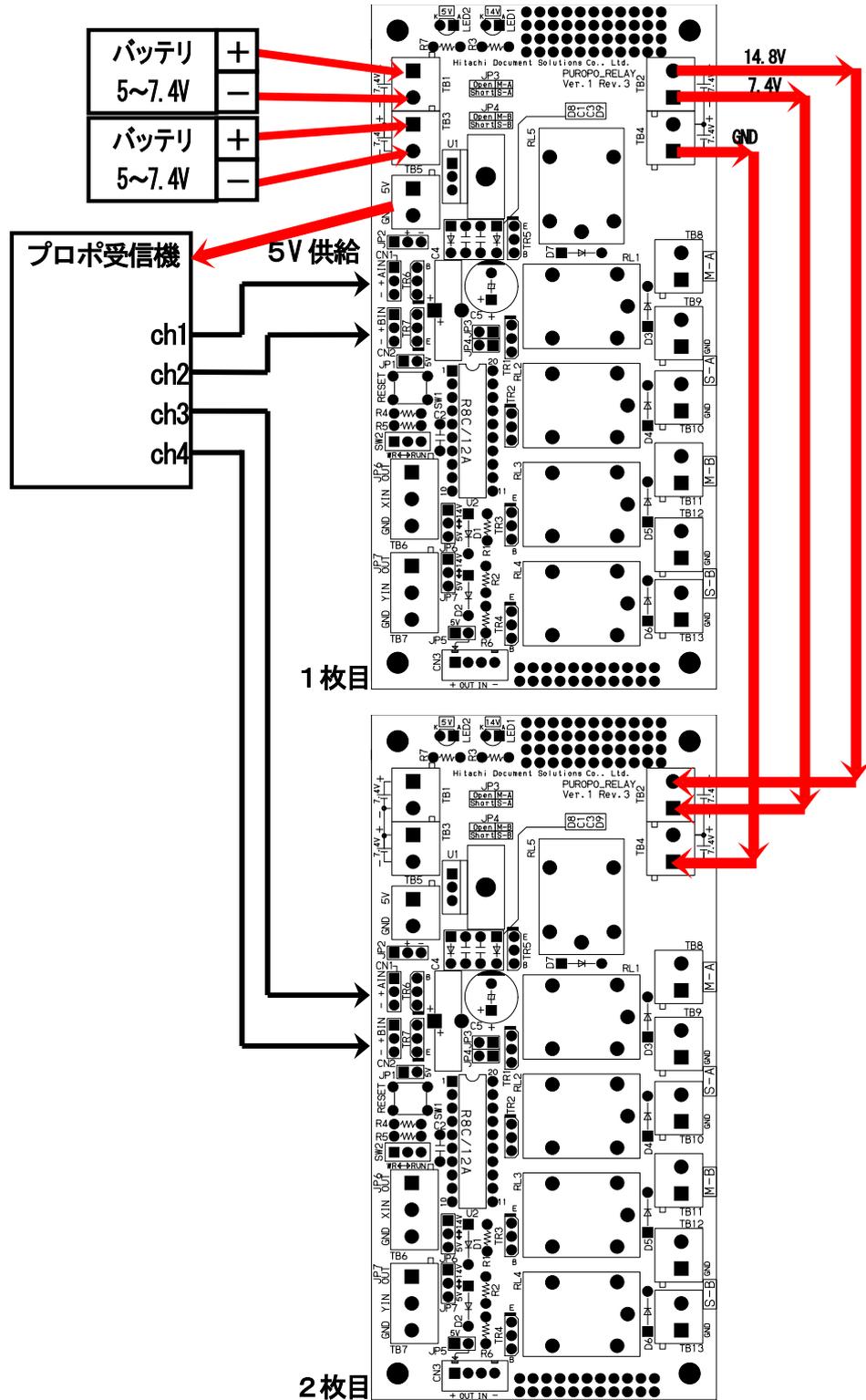
電池は、**充電式**単四電池を使用して下さい。アルカリ電池の場合は $1.5V \times 4 = 6.0V$ となり、**マイコンの動作保証範囲外となります**。



4. 接続

4.3 複数のプロポリレー基板を使って、多くのモータを制御する

2枚以上のプロポリレー基板を使う場合、下記のように結線します。プロポ受信機からの信号を、各基板に接続します。



4. 接続

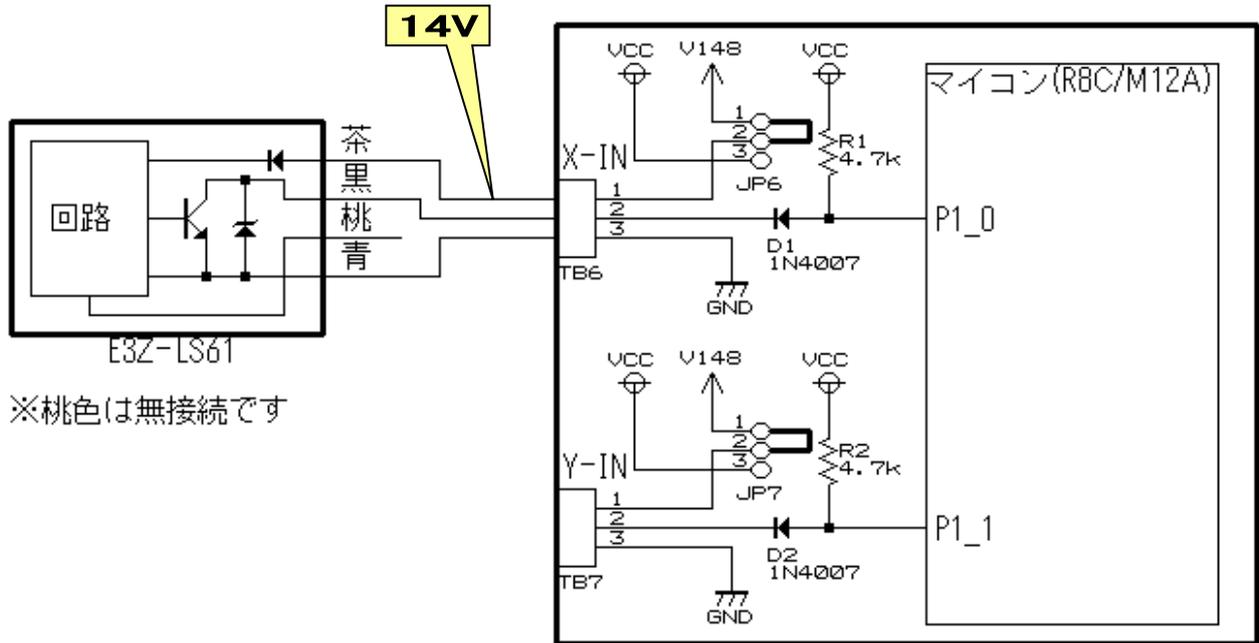
4.4 センサの接続方法

TB6、TB7 に接続するセンサについて説明します。

4.4.1 NPN 出力タイプの場合 (12V 動作のセンサ)

NPN 出力タイプのセンサの場合、外付け部品なしで配線することができます。

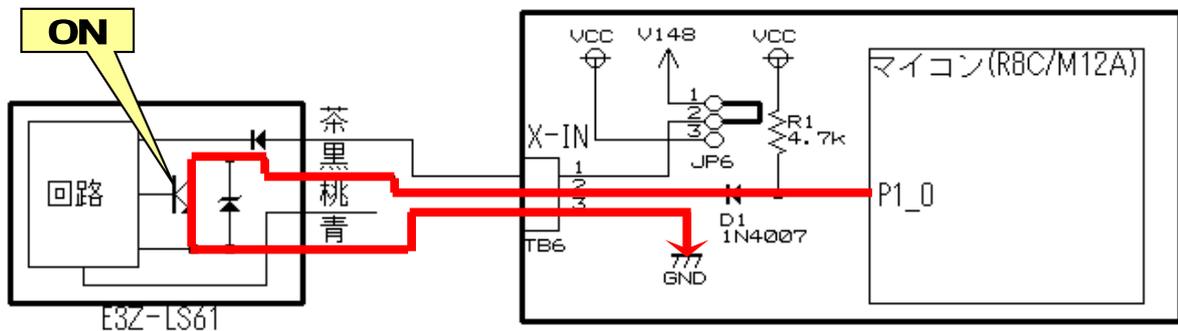
例として、オムロンの「細ビーム反射形 コード引き出しタイプ 検出距離 90±30mm NPN 出力 (E3Z-L61)」を接続するときの回路を、下記に示します。



※桃色は無接続です

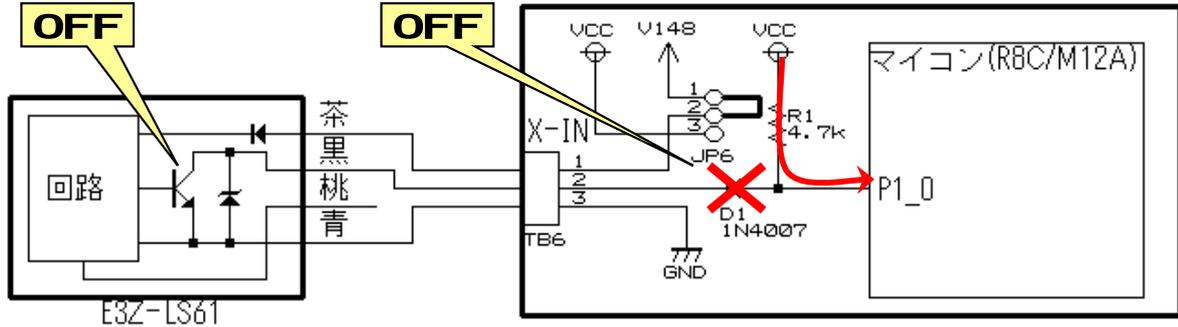
※動作原理

センサが ON になると、トランジスタが ON するため、P1_0 端子は矢印のように 0V に接続され、「0」が入力されます(下図)。



4. 接続

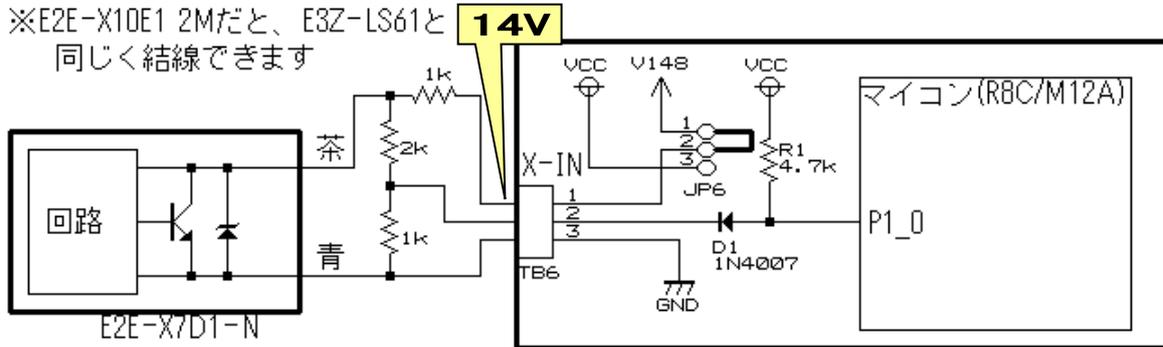
センサが OFF になると、トランジスタが OFF して、D1 は OFF になるため、R1 のプルアップ抵抗を通して、P1_0 端子は矢印のように 5V に接続され、「1」が入力されます(下図)。※センサによってはトランジスタが OFF しても 8 ~ 12V の電圧が出力される場合があるので、D1 は必ず付けてください。マイコンが壊れることがあります。



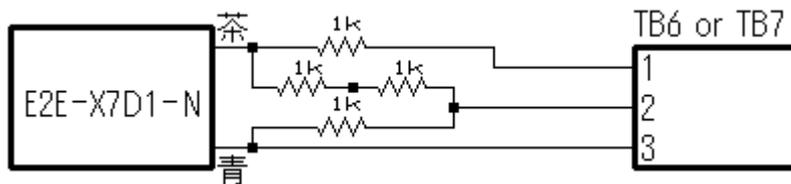
4.4.2 2線式の場合(12V 動作のセンサ)

例として、オムロンの「直流2線式 シールド M18 検出距離 7mm NO 出力 (E2E-X7D1-N 2M)」を接続するときの回路を、下記に示します(二線式は、外付け回路が多くなるので、NPN 出力型がお勧めです)。

※E2E-X10E1 2Mだと、E3Z-LS61と同じく結線できます



1kΩ の抵抗を 4 個使い、下記のように配線すると、良いでしょう。



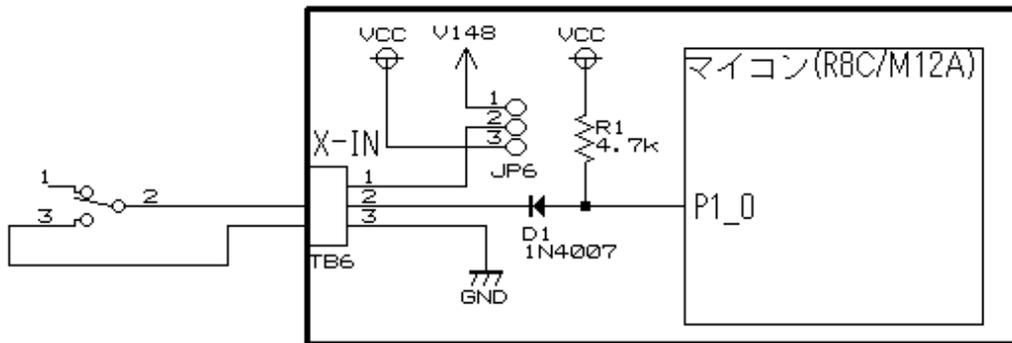
実際の半田付け例を、下記に示します。



4. 接続

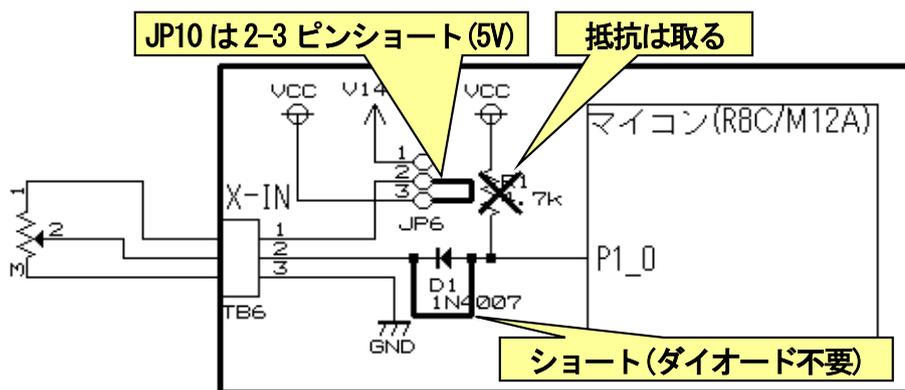
4.4.3 スイッチを接続する場合

TB6 または TB7 の 2 ピン (IN) と 3 ピン (GND) に、スイッチをつなぎます。スイッチが ON で "0"、OFF で "1" になります。接続例を下記に示します。



4.4.4 5V で動作するアナログ出力のセンサの場合

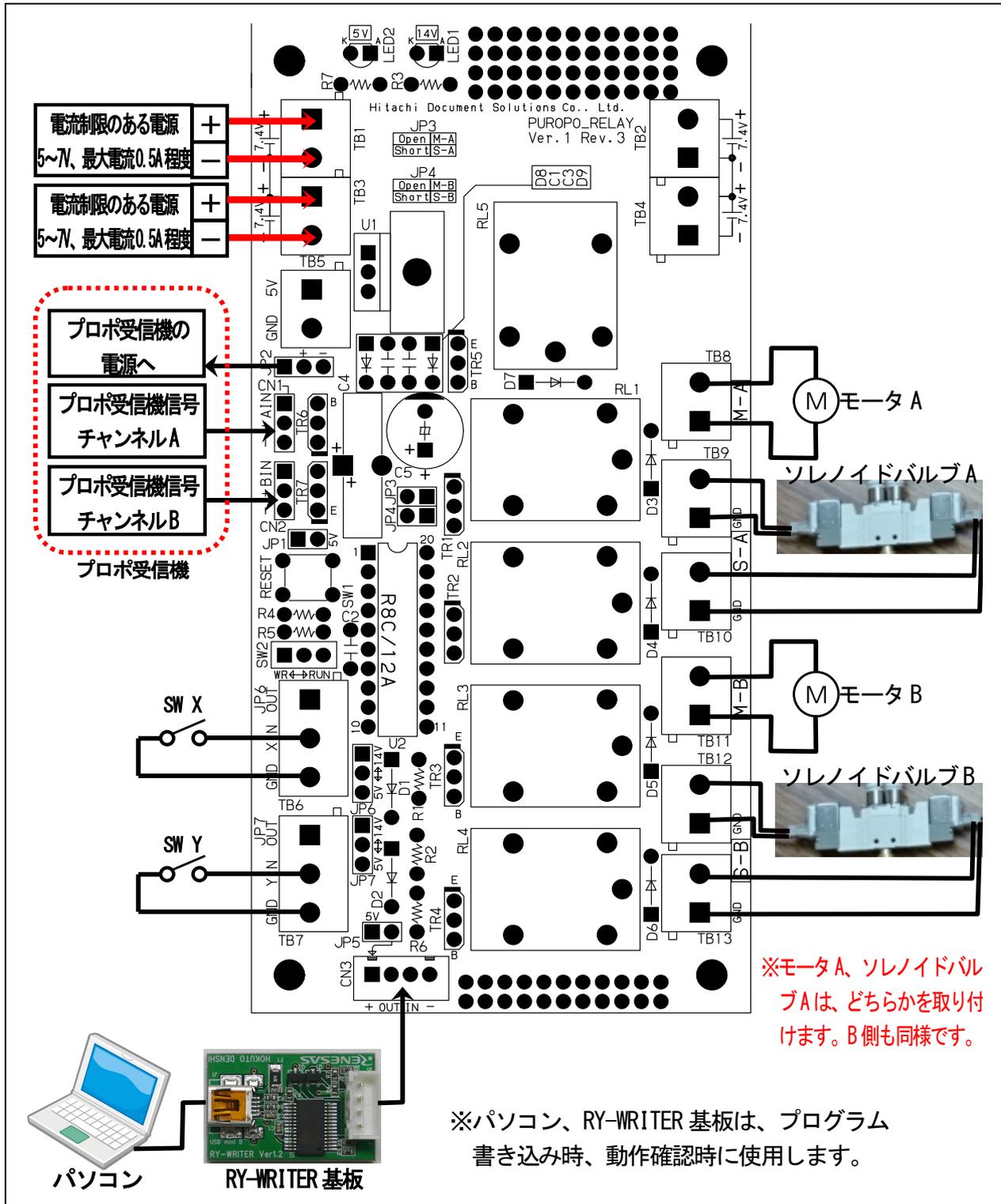
0~5V の信号を入力する場合、D1 はショート、R1 は取ります。また、JP2 は 5V 側にします (2-3 ピン側をショート)。接続例を下記に示します。



5. 動作確認

5.1 動作確認の結線

動作確認の結線を、下記に示します。**必ず電流制限ができる電源 2 台を使ってください。**モータを回しているとき以外に電流リミットがかかった場合、半田付け不良、部品の付け間違いなど確認してください。



5. 動作確認

5.2 ワークスペース(プログラム)のダウンロードとプログラムの書き込み

1	<p>日立ドキュメントソリューションズは、マイコンカー製作キットをはじめとして、ものづくりから基本的なマイコン制御を試行錯誤しながら学習ができるマイコン学習教材の開発・販売を行っています。</p>	<p>HITACHI Inspire the Next</p> <p>マイコンカーラーリー販売 サイト https://www2.himdx.net/mcr/</p> <p>にアクセスします。 「ダウンロード」をクリックします。</p>
---	--	--

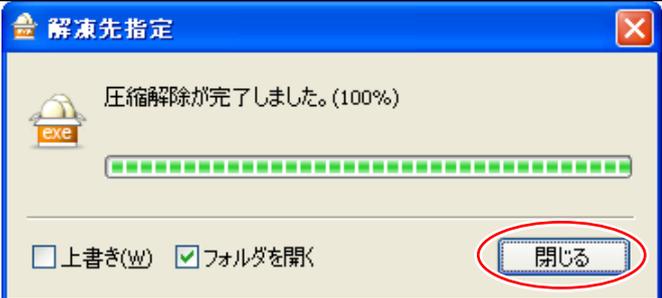
2	<p>ダウンロード(R8C,RXマイコンに関する資料)</p> <p>↓ 開発環境に関する資料 ↓ マイコンカーキットに関する資料 ↓ 各種基板に関する資料</p> <p>↓ ミニマイコンカーVer.2に関する資料 ↓ TypeS基板に関する資料 ↓ 基板マイコンカーに関する資料</p> <p>↓ マトリクス・ジュニア製作キットに関する資料 ↓ R8C/M12Aマイコンに関する資料</p> <p>↓ RMC-RX62Gボードに関する資料 ↓ その他資料</p> <p>「マニュアル」「ソフトウェア」は万全な体制で制作されており、通常の使用環境においては正常に動作するように作成されていますが、万が一「マニュアル」「ソフトウェア」による損失・損害が発生したときには、当社はいかなる場合も責任を負いません。ご利用者の自己責任においてご利用をお願いいたします。</p>	<p>「各種基板に関する資料」をクリックします。</p>
---	---	------------------------------

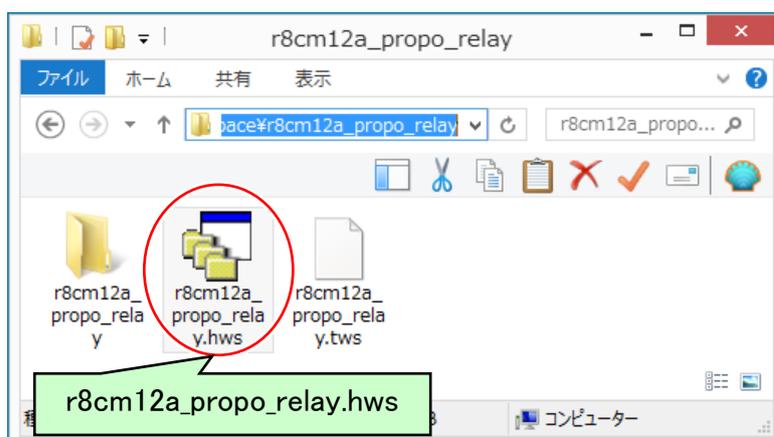
3	<table border="1"> <tr> <td>を受信して、定格75AのFETでモータ2個を制御することのできる基板です。</td> <td>第1.01版 2015.10.19</td> <td>tet.zip 2015.10.05</td> </tr> <tr> <td>プロボ信号受信リレーモータドライブ基板 プロボ受信機からの信号(PWM信号)を受信して、定格25A(2分)、または15A(1時間)のリレーでモータ2個を制御することのできる基板です。</td> <td>プロボ信号受信リレーモータドライブ基板製作 製作・プログラム解説マニュアル 第1.00版 2015.10.29</td> <td>r8cm12a_propo_relay.zip 2015.10.27</td> </tr> </table> <p style="text-align: right;">↑ ページの先頭へ</p>	を受信して、定格75AのFETでモータ2個を制御することのできる基板です。	第1.01版 2015.10.19	tet.zip 2015.10.05	プロボ信号受信リレーモータドライブ基板 プロボ受信機からの信号(PWM信号)を受信して、定格25A(2分)、または15A(1時間)のリレーでモータ2個を制御することのできる基板です。	プロボ信号受信リレーモータドライブ基板製作 製作・プログラム解説マニュアル 第1.00版 2015.10.29	r8cm12a_propo_relay.zip 2015.10.27	<p>「r8cm12a_propo_relay.zip」をダウンロードして、解凍します。</p>
を受信して、定格75AのFETでモータ2個を制御することのできる基板です。	第1.01版 2015.10.19	tet.zip 2015.10.05						
プロボ信号受信リレーモータドライブ基板 プロボ受信機からの信号(PWM信号)を受信して、定格25A(2分)、または15A(1時間)のリレーでモータ2個を制御することのできる基板です。	プロボ信号受信リレーモータドライブ基板製作 製作・プログラム解説マニュアル 第1.00版 2015.10.29	r8cm12a_propo_relay.zip 2015.10.27						

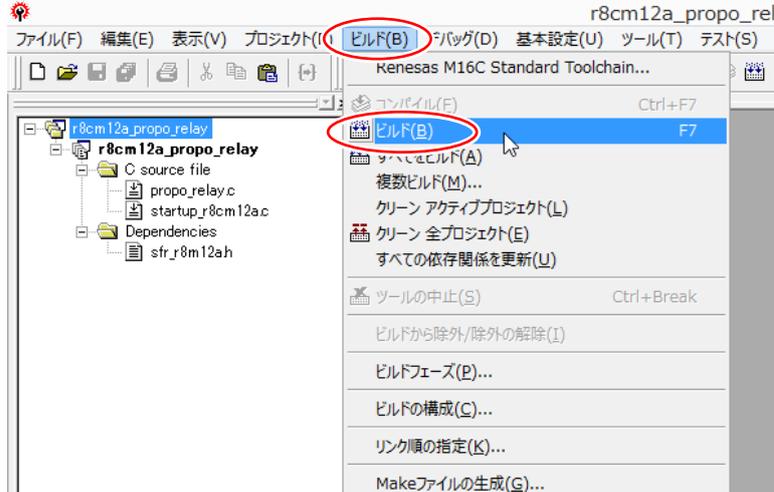
4		<p>「r8cm12a_propo_relay.exe」を実行して、圧縮解除をクリックします。</p> <p>※フォルダは変更できません。変更した場合は、ルネサス統合開発環境の設定を変更する場合があります。</p>
---	--	--

5. 動作確認

5		<p>解凍が終わったら、自動的に「Cドライブ→Workspace」フォルダが開かれます。今回使用するのは、「r8cm12a_propo_relay」です。</p>
---	--	---

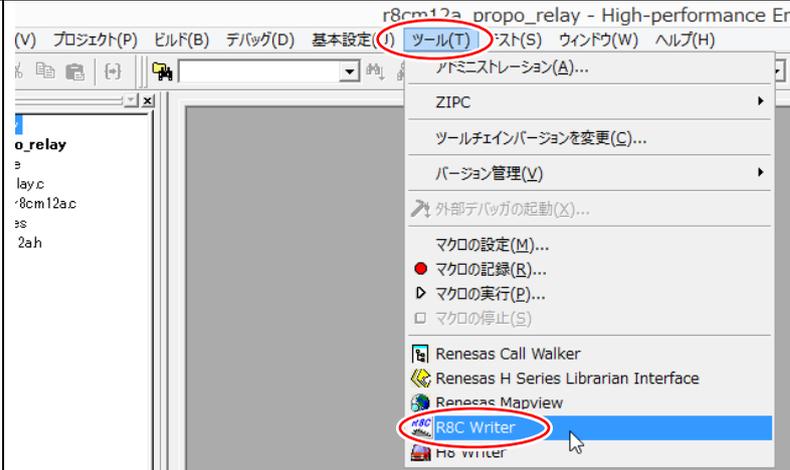
6		<p>閉じるをクリックして終了です。</p>
---	---	------------------------

7		<p>「Cドライブ→ Workspace→ r8cm12a_propo_relay→ r8cm12a_propo_relay.hws」 をダブルクリックすると、ルネサス 統合開発環境が立ち上がります。</p>
---	--	--

8		<p>「ビルド→ビルド」でマイコンに書き込むファイル(MOT ファイル)を作成します。</p>
---	--	---

5. 動作確認

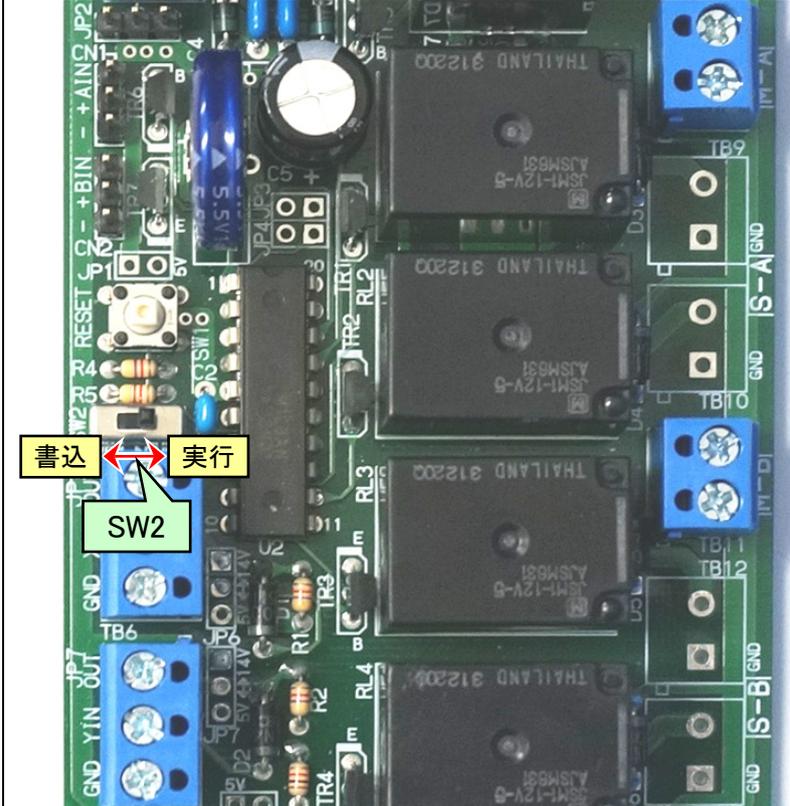
9



The screenshot shows the 'Tools' menu in the Renesas IDE. The 'Tools' menu item is circled in red. The 'R8C Writer' option is highlighted in blue, and a mouse cursor is pointing at it.

「ツール→R8C Writer」で書き込みソフトを立ち上げます。

10



The photograph shows the PCB with a switch labeled 'SW2'. Two yellow callout boxes with arrows point to the switch: one labeled '書込' (Write) and one labeled '実行' (Execute).

- ①プロポリレー基板の電源は入れません。
- ②SW2 を基板の外側にします。
- ③電源を入れます。
- ④R8C Writer の書き込み開始ボタンをクリックして、プログラムを書き込みます。
- ⑤正常に書き込みができれば、電源を OFF にします。書き込みができない場合は、RY-WRITER 基板との結線や、プロポリレー基板に半田付けした部品が正しく取り付けられているか、半田付け不良がないかなど確認してください。
- ⑥SW を基板の内側にします。

5. 動作確認

5.3 動作確認

1		<p>プロポリレー基板の電源はまだ、OFF にしておいてください。</p> <p>Tera Term を立ち上げます。「シリアル」を選択、ポートを RY-WRITER 基板の番号にして、OK をクリックします。</p> <p>※Tera Term がインストールされていない場合、下記からダウンロード、インストールしてください。 https://osdn.jp/projects/ttssh2/</p>
---	--	--

2	<p>PUROPO Relay PCB Ver. 1</p> <p>A=<u>03759</u>(<u>00297</u>) B=<u>03757</u>(<u>00297</u>) TB6=<u>1</u>(<u>1023</u>) TB7=<u>1</u>(<u>0914</u>)</p> <p style="text-align: center;">① ② ③ ④ ⑤ ⑥ ⑦ ⑧</p> <p>プロポリレー基板の電源を入ると、上記のようなメッセージが Tera Term の画面に表示されます。下記の確認を行ってください。</p> <p>①プロボのスティックを動かすと、①の値が動きます。 上下(または左右)に動かすと、中心位置から±1000 程度値が変化します。 変化しない場合は、プロボ受信機のチャンネル番号が間違っているなど考えられます。</p> <p>②ここでは使いません(受信回数を表示します)。</p> <p>③プロボのスティックを動かすと、③の値が動きます。 上下(または左右)に動かすと、中心位置から±1000 程度値が変化します。 変化しない場合は、プロボ受信機のチャンネル番号が間違っているなど考えられます。</p> <p>④ここでは使いません(受信回数を表示します)。</p> <p>⑤SW X をショートさせると"0"に、ショートさせないと"1"になります。</p> <p>⑥ここでは使いません(A/D 値を表示します)。</p> <p>⑦SW Y をショートさせると"0"に、ショートさせないと"1"になります。</p> <p>⑧ここでは使いません(A/D 値を表示します)。</p> <p>値の調整については、「7.4.3 調整のポイント」を参照してください。</p>
---	--

6. ワークスペース「r8cm12a_propo_relay」

プログラムのダウンロードは、「5.2 ワークスペース(プログラム)のダウンロードとプログラムの書き込み」を参照してください。

6.1 プロジェクトの構成



1	puropo_relay.c	実際に制御するプログラムが書かれています。R8C/M12A マイコンの内蔵周辺機能(SFR)の初期化も行います。 ファイルの位置→C:\¥Workspace¥r8cm12a_propo_relay¥r8cm12a_propo_relay¥puropo_relay.c
2	startup_r8cm12a.c	固定割り込みベクタアドレスの設定、スタートアッププログラム、RAMの初期化(初期値のないグローバル変数、初期値のあるグローバル変数の設定)などを行います。 ファイルの位置→C:\¥Workspace¥r8cm12a_propo_relay¥r8cm12a_propo_relay¥startup_r8cm12a.c
3	sfr_r8m12a.h	R8C/M12A マイコンの内蔵周辺機能を制御するためのレジスタ(Special Function Register)を定義したファイルです。 ファイルの位置→C:\¥Workspace¥r8cm12a_propo_relay¥r8cm12a_propo_relay¥sfr_r8m12a.h

7. プログラム「propo_relay.c」

7.1 概要

このファイルの中に main 関数があり、プロボリレー基板を制御するプログラムが入っています。プログラムを改造するときは、通常このファイルを修正/します。他の C ファイルや H ファイルは、ほとんど修正しません。

7.2 プログラムリスト

```

1 : //*****
2 : // 対象マイコン R8C/M12A
3 : // ファイル内容 プロボ信号受信リレーモータドライブ基板 制御プログラム
4 : // バージョン Ver. 1.00
5 : // Date 2015.10.27
6 : // Copyright 株式会社日立ドキュメントソリューションズ
7 : //*****
8 :
9 : //=====
10 : // インクルード
11 : //=====
12 : #include "sfr_r8m12a.h" // R8C/M12A SFRの定義ファイル
13 :
14 : //=====
15 : // シンボル定義
16 : //=====
17 : // CH_A のプロボ信号 受信パルス幅
18 : #define CH_A_PLUS_LIMIT 5050 // これ以上は無効とする(CH_A_PLUS+300くらい)
19 : #define CH_A_PLUS 4750 // いちばん上にしたとき
20 : #define CH_A_PLUS_HIGH 4550 // HIGHモードで正転
21 : #define CH_A_PLUS_LOW 4050 // LOWモードで正転
22 : #define CH_A_STOP 3750 // プロボステイック ニュートラル
23 : #define CH_A_MINUS_LOW 3450 // LOWモードで逆転
24 : #define CH_A_MINUS_HIGH 2950 // HGIIHモードで逆転
25 : #define CH_A_MINUS 2750 // いちばん下にしたとき
26 : #define CH_A_MINUS_LIMIT 2450 // これ以下は無効とする(CH_A_MINUS-300くらい)
27 :
28 : #define CH_A_SORE_UE 4250 // ソレノイドバルブ 上側ON
29 : #define CH_A_SORE_SHITA 3250 // ソレノイドバルブ 下側ON
30 :
31 : // CH_B のプロボ信号 受信パルス幅
32 : #define CH_B_PLUS_LIMIT 5050 // これ以上は無効とする(CH_A_PLUS+300くらい)
33 : #define CH_B_PLUS 4750 // いちばん上にしたとき
34 : #define CH_B_PLUS_HIGH 4550 // HIGHモードで正転
35 : #define CH_B_PLUS_LOW 4050 // LOWモードで正転
36 : #define CH_B_STOP 3750 // プロボステイック ニュートラル
37 : #define CH_B_MINUS_LOW 3450 // LOWモードで逆転
38 : #define CH_B_MINUS_HIGH 2950 // HGIIHモードで逆転
39 : #define CH_B_MINUS 2750 // いちばん下にしたとき
40 : #define CH_B_MINUS_LIMIT 2450 // これ以下は無効とする(CH_A_MINUS-300くらい)
41 :
42 : #define CH_B_SORE_UE 4250 // ソレノイドバルブ 上側ON
43 : #define CH_B_SORE_SHITA 3250 // ソレノイドバルブ 下側ON
44 :
45 : // ターミナルブロックとI/O、A/Dのチャンネル
46 : #define TB6 p1_0 // TB6の入力ポート
47 : #define TB7 p1_1 // TB7の入力ポート
48 :
49 : #define TB6_AD 0 // TB6のA/Dチャンネル
50 : #define TB7_AD 1 // TB7のA/Dチャンネル
51 :
52 : //=====
53 : // プロトタイプ宣言
54 : //=====
55 : void init( void );
56 : void set_trcioc( int out );
57 : void ch_a( int mode );
58 : void ch_b( int mode );
59 :
60 : int get_ad( int ch );
61 :
62 : int get_uart0( char *s );
63 : int put_uart0( char r );
64 : void put_uart0_str( char *str );
65 : void put_uart0_num( long cnt, int keta );
66 : void put_uart0_hex( unsigned long value, int keta );
67 :

```

7. プログラム「propo_relay.c」

```

68 : //=====
69 : // グローバル変数
70 : //=====
71 : int cnt_rb; // 1msごとに+1する変数
72 :
73 : long pulse_a; // プロボ受信信号CH_A パルス幅
74 : long pulse_b; // プロボ受信信号CH_B パルス幅
75 :
76 : int pulse_a_inc; // プロボ受信信号CH_A 受信回数(確認用)
77 : int pulse_b_inc; // プロボ受信信号CH_B 受信回数(確認用)
78 :
79 : int cnt_a; // パルス幅 CH_A 受信が無い時間
80 : int cnt_b; // パルス幅 CH_B 受信が無い時間
81 :
82 : //*****
83 : // メインプログラム
84 : //*****
85 : void main( void )
86 : {
87 :     init(); // 初期化
88 :     asm("FSET I"); // 全体割り込み許可
89 :
90 :     put_uart0_str( "¥n¥nPUR0P0 Relay PCB Ver.1¥n¥n" );
91 :
92 :     while( 1 ) {
93 :         if( cnt_rb >= 200 ) {
94 :             cnt_rb = 0;
95 :
96 :             put_uart0_str( "A=" );
97 :             put_uart0_num( pulse_a, 5 );
98 :             put_uart0_str( "(" );
99 :             put_uart0_num( pulse_a_inc, 5 );
100 :            put_uart0_str( ") B=" );
101 :            put_uart0_num( pulse_b, 5 );
102 :            put_uart0_str( "(" );
103 :            put_uart0_num( pulse_b_inc, 5 );
104 :            put_uart0_str( ") TB6=" );
105 :            put_uart0_num( pl_0, 1 );
106 :            put_uart0_str( "(" );
107 :            put_uart0_num( get_ad(TB6_AD), 4 );
108 :            put_uart0_str( ") TB7=" );
109 :            put_uart0_num( pl_1, 1 );
110 :            put_uart0_str( "(" );
111 :            put_uart0_num( get_ad(TB7_AD), 4 );
112 :            put_uart0_str( ")");
113 :
114 :            put_uart0_str( "¥r" ); // 先頭に戻る
115 :        }
116 :    }
117 : }
118 :
119 : //*****
120 : // R8C/M12A スペシャルファンクションレジスタ(SFR)の初期化
121 : //*****
122 : void init( void )
123 : {
124 :     // 内蔵ブルアップ設定
125 :     pu1_2 = 1; // 受信機信号Aのトランジスタ出力
126 :     pu1_7 = 1; // 受信機信号Bのトランジスタ出力
127 :     pu4_2 = 1; // JP3(チャンネルAの選択)
128 :     pu4_7 = 1; // JP4(チャンネルBの選択)
129 :
130 :     // ポートの入出力設定
131 :     pd1 = 0x18; // ポート1の入出力設定
132 :     pd3 = 0x38; // ポート3の入出力設定
133 :     pd4 = 0x20; // ポート4の入出力設定
134 :     pda = 0x00; // ポートAの入出力設定
135 :
136 :     // タイマRBの設定 1msごとに割り込みを発生させる
137 :     mstrb = 0; // タイマRB2を有効にする
138 :     trbmr = 0x00; // 動作モード、分周比設定
139 :     trbpre = 200-1; // プリスケアラレジスタ
140 :     trbpr = 100-1; // プライマリレジスタ
141 :     trbir = 0xc0; // タイマ割り込み要求
142 :     ilvlc = 0x03; // 割り込み優先レベル設定
143 :     trbcr = 0x01; // カウント開始
144 :
145 :     // タイマRCの設定
146 :     mstrc = 0; // タイマRCを有効にする
147 :     pl2sel1 = 0; // TRCIOB端子の選択
148 :     pl2sel0 = 1; // P1_2をTRCIOB端子にする
149 :     iob2_trcior0 = 1; // TRCIOB端子はインプットキャプチャ(両エッジ)で使用
150 :     iob1_trcior0 = 1;
151 :     iob0_trcior0 = 0;
152 :     dfck1_trcdf = 1; // デジタルフィルタ f1
153 :     dfck0_trcdf = 0;
154 :     dfb_trcdf = 1; // TRCIOB端子はデジタルフィルタ使用
155 :     trcrl = 0xb0; // カウントソース = f8
156 :     trcgra = 25000-1; // インプットキャプチャの周期 (1/20M*8)*25000=10ms
157 :     trcgrb = 0; // TRCIOB端子のON幅の設定
158 :     eb_trcoer = 1; // TRCIOBの出力 出力許可

```

7. プログラム「propo_relay.c」

```

159 :     imiea_trcier = 1;
160 :     imieb_trcier = 1;
161 :     ilvl35 = 1;           // 割り込みレベルの設定
162 :     ilvl34 = 0;
163 :     cts_trcmr = 1;       // TRCCNT カウント開始
164 :
165 :     // タイマRJ2の設定
166 :     mstrj = 0;           // タイマRJ2を有効にする
167 :     p17sel1 = 1;        // P1_7をTRJIO端子にする
168 :     p17sel0 = 0;        // "
169 :     tedgsel_trjioc = 0; // Lレベル幅を測定
170 :     tipfl_trjioc = 0;   // フィルタ有り f1
171 :     tipf0_trjioc = 1;
172 :     trjmr = 0x13;       // パルス幅測定モードに設定 f8
173 :     trj = 0xffff;      // trjのカウンタはダウンカウント
174 :     ilvlb1 = 1;        // 割り込みレベルの設定
175 :     ilvlb0 = 0;
176 :     trjie_trjir = 1;    // 割り込み許可
177 :     tstart_trjcr = 1;   // タイムスタート
178 :
179 :     // UART0の設定
180 :     mstuart = 0;        // UART0を有効にする
181 :     p15sel2 = 0;        // P1_5 = RXD0端子にする
182 :     p15sel1 = 0;
183 :     p15sel0 = 1;
184 :     p14sel2 = 0;        // P1_4 = TXD0端子にする
185 :     p14sel1 = 0;
186 :     p14sel0 = 1;
187 :     u0mr = 0x05;        // UARTモード8bit
188 :                             // 1ストップビット,パリティなし
189 :     u0c0 = 0x10;        // U0BRGカウンタ: f1
190 :     u0brg = 129;        // f1 / (bps*16) - 1
191 :                             // = 20*10^6/(9600*16)-1=129.2
192 :     te_u0c1 = 1;        // 送信許可ビット:送信許可
193 :     re_u0c1 = 1;        // 受信許可ビット:受信許可
194 : }
195 :
196 : //*****
197 : // タイマRB 1msごとの割り込み処理
198 : //*****
199 : #pragma interrupt intTRB(vect=24)
200 : void intTRB( void )
201 : {
202 :     static int speed_high_a = 0; // CH_A 1=HIGH 0=LOW
203 :     static int speed_high_b = 0; // CH_B 1=HIGH 0=LOW
204 :
205 :     trbif_trbir = 0;           // フラグのクリア
206 :
207 :     // カウンタインクリメント
208 :     cnt_rb++;
209 :     if( cnt_a < 30000 ) cnt_a++;
210 :     if( cnt_b < 30000 ) cnt_b++;
211 :
212 :     // チャンネルA制御
213 :     if( p4_2 == 1 ) {
214 :         // JP3 オープンならチャンネルAはモータモード
215 :         if( cnt_a >= 200 ) {
216 :             // 指定時間以上、パルス幅入力がなれば停止
217 :             speed_high_a = 0;
218 :             ch_a( 0 );
219 :         } else if( pulse_a >= CH_A_PLUS_LIMIT ) {
220 :             // リミット以上なら停止
221 :             speed_high_a = 0;
222 :             ch_a( 0 );
223 :         } else if( pulse_a >= CH_A_PLUS_HIGH ) {
224 :             // HIGHで正転
225 :             speed_high_a = 1;
226 :             ch_a( 1 );
227 :         } else if( pulse_a >= CH_A_PLUS_LOW ) {
228 :             // LOWで正転
229 :             speed_high_a = 0;
230 :             ch_a( 1 );
231 :         } else if( pulse_a <= CH_A_MINUS_LIMIT ) {
232 :             // リミット以下なら停止
233 :             speed_high_a = 0;
234 :             ch_a( 0 );
235 :         } else if( pulse_a <= CH_A_MINUS_HIGH ) {
236 :             // HIGHで逆転
237 :             speed_high_a = 1;
238 :             ch_a( -1 );
239 :         } else if( pulse_a <= CH_A_MINUS_LOW ) {
240 :             // LOWで逆転
241 :             speed_high_a = 0;
242 :             ch_a( -1 );
243 :         } else {
244 :             // それ以外は停止
245 :             speed_high_a = 0;
246 :             ch_a( 0 );
247 :         }

```

7. プログラム「propo_relay.c」

```
248 :     } else {
249 :         // JP3 ショートならチャンネルAはソレノイドバルブモード
250 :         if( cnt_a >= 200 ) {
251 :             // 指定時間以上、パルス幅入力がなれば何もしない
252 :         } else if( pulse_a >= CH_A_PLUS_LIMIT ) {
253 :             // リミット以上なら何もしない
254 :         } else if( pulse_a >= CH_A_SORE_UE ) {
255 :             // ソレノイドバルブ、上側ONにする
256 :             ch_a( 1 );
257 :         } else if( pulse_a <= CH_A_SORE_SHITA ) {
258 :             // ソレノイドバルブ、下側ONにする
259 :             ch_a( -1 );
260 :         } else if( pulse_a <= CH_A_MINUS_LIMIT ) {
261 :             // リミット以下なら何もしない
262 :         } else {
263 :             // それ以外はOFF
264 :             ch_a( 0 );
265 :         }
266 :     }
267 :
268 :     // チャンネルB制御
269 :     if( p4_7 == 1 ) {
270 :         // JP4 オープンならチャンネルBはモータモード
271 :         if( cnt_b >= 200 ) {
272 :             // 指定時間以上、パルス幅入力がなれば停止
273 :             speed_high_b = 0;
274 :             ch_b( 0 );
275 :         } else if( pulse_b >= CH_B_PLUS_LIMIT ) {
276 :             // リミット以上なら停止
277 :             speed_high_b = 0;
278 :             ch_b( 0 );
279 :         } else if( pulse_b >= CH_B_PLUS_HIGH ) {
280 :             // HIGHで正転
281 :             speed_high_b = 1;
282 :             ch_b( 1 );
283 :         } else if( pulse_b >= CH_B_PLUS_LOW ) {
284 :             // LOWで正転
285 :             speed_high_b = 0;
286 :             ch_b( 1 );
287 :         } else if( pulse_b <= CH_B_MINUS_LIMIT ) {
288 :             // リミット以下なら停止
289 :             speed_high_b = 0;
290 :             ch_b( 0 );
291 :         } else if( pulse_b <= CH_B_MINUS_HIGH ) {
292 :             // HIGHで逆転
293 :             speed_high_b = 1;
294 :             ch_b( -1 );
295 :         } else if( pulse_b <= CH_B_MINUS_LOW ) {
296 :             // LOWで逆転
297 :             speed_high_b = 0;
298 :             ch_b( -1 );
299 :         } else {
300 :             // それ以外は停止
301 :             speed_high_b = 0;
302 :             ch_b( 0 );
303 :         }
304 :     } else {
305 :         // JP4 ショートならチャンネルBはソレノイドバルブモード
306 :         if( cnt_b >= 200 ) {
307 :             // 指定時間以上、パルス幅入力がなれば何もしない
308 :         } else if( pulse_b >= CH_B_PLUS_LIMIT ) {
309 :             // リミット以上なら何もしない
310 :         } else if( pulse_b >= CH_B_SORE_UE ) {
311 :             // ソレノイド、上側ONにする
312 :             ch_b( 1 );
313 :         } else if( pulse_b <= CH_B_SORE_SHITA ) {
314 :             // ソレノイド、下側ONにする
315 :             ch_b( -1 );
316 :         } else if( pulse_b <= CH_B_MINUS_LIMIT ) {
317 :             // リミット以下なら何もしない
318 :         } else {
319 :             // それ以外はOFF
320 :             ch_b( 0 );
321 :         }
322 :     }
323 :
324 :     // RL5のリレー制御
325 :     if( p4_2 == 1 && p4_7 == 1 ) {
326 :         // チャンネルA、チャンネルB、両方ともモータモードなら
327 :         if( speed_high_a == 1 || speed_high_b == 1 ) {
328 :             // CH_AとCH_BのどちらかがHIGHモードなら、HIGHにする
329 :             pl_3 = 1;
330 :         } else {
331 :             // どちらもLOWモードなら、LOWにする
332 :             pl_3 = 0;
333 :         }
334 :     } else {
335 :         // 1つ以上ソレノイドバルブなら、リレーはOFFにする
336 :         pl_3 = 0;
337 :     }
338 : }
```

7. プログラム「propo_relay.c」

```

339 :
340 : //*****
341 : // CH_Aのリレー
342 : // 引数 1:正転 -1:逆転 0:停止
343 : // 戻り値 無し
344 : //*****
345 : void ch_a( int mode )
346 : {
347 :     if( mode > 0 ) {
348 :         p3_5 = 1;           // 正転
349 :         p4_5 = 0;
350 :     } else if( mode < 0 ) {
351 :         p3_5 = 0;           // 逆転
352 :         p4_5 = 1;
353 :     } else {
354 :         p3_5 = 0;           // 停止
355 :         p4_5 = 0;
356 :     }
357 : }
358 :
359 : //*****
360 : // CH_Bのリレー
361 : // 引数 1:正転 -1:逆転 0:停止
362 : // 戻り値 無し
363 : //*****
364 : void ch_b( int mode )
365 : {
366 :     if( mode > 0 ) {
367 :         p3_4 = 1;           // 正転
368 :         p3_3 = 0;
369 :     } else if( mode < 0 ) {
370 :         p3_4 = 0;           // 逆転
371 :         p3_3 = 1;
372 :     } else {
373 :         p3_4 = 0;           // 停止
374 :         p3_3 = 0;
375 :     }
376 : }
377 :
378 : //*****
379 : // タイマRC 割り込み処理
380 : //*****
381 : #pragma interrupt intTRC(vect=7)
382 : void intTRC( void )
383 : {
384 :     static unsigned int pulse_a_start = 0;
385 :
386 :     if( imfb_trcsr == 1 ) {
387 :         imfb_trcsr = 0;
388 :
389 :         if( p1_2 == 0 ) {
390 :             pulse_a_start = trcgrb;
391 :         } else {
392 :             if( pulse_a_start < trcgrb ) {
393 :                 pulse_a = trcgrb - pulse_a_start;
394 :             } else {
395 :                 pulse_a = (long)25000 + trcgrb - pulse_a_start;
396 :             }
397 :             pulse_a_inc++;
398 :             cnt_a = 0;
399 :         }
400 :     }
401 : }
402 :
403 : //*****
404 : // タイマRJ2 割り込み処理
405 : //*****
406 : #pragma interrupt intTRJ2(vect=22)
407 : void intTRJ2( void )
408 : {
409 :     static long pulse_b_start = 0xffff;
410 :
411 :     unsigned char c;
412 :
413 :     trjif_trjir = 0;
414 :
415 :     if( tundf_trjcr == 1 ) {
416 :         // アンダーフロー
417 :         c = trjcr & 0xdf;
418 :         trjcr = c;
419 :         pulse_b_start += 0x10000;
420 :     }
421 :
422 :     if( tedgf_trjcr == 1 ) {
423 :         // 有効エッジ(立ち下がりがり)あり
424 :         c = trjcr & 0xef;
425 :         trjcr = c;
426 :
427 :         tstart_trjcr = 0;           // タイマ停止
428 :         pulse_b = pulse_b_start - trj;
429 :         trj = 0xffff;

```

7. プログラム「propo_relay.c」

```

430 :         pulse_b_start = 0xffff;
431 :         tstart_trjcr = 1;           // タイマスタート
432 :
433 :         pulse_b_inc++;
434 :         cnt_b = 0;
435 :     }
436 : }
437 :
438 : //*****
439 : // TRCIOC端子から出力するON幅を設定
440 : // 引数 0:0% ~ TRCGRAの値(19999):100%
441 : //*****
442 : void set_trcioc( int out )
443 : {
444 :     if( out == 0 ) {
445 :         // 0%出力
446 :         if( trcgrc < trcgra ) while( trcgrc >= trccnt );
447 :         trcgrc = trcgra;
448 :     } else if( out == trcgra ) {
449 :         // 100%出力
450 :         trcgrc = trcgra + 1;
451 :     } else {
452 :         // 0%,100%以外
453 :         if( trcgrc < trcgra ) while( trcgrc >= trccnt );
454 :         trcgrc = out;
455 :     }
456 : }
457 :
458 : //*****
459 : // AN端子をA/D変換
460 : // 引数 チャンネル番号(0~4,7)
461 : // 戻り値 A/D変換値(0~1023) 9999:チャンネル選択エラー
462 : //*****
463 : int get_ad( int ch )
464 : {
465 :     int i;
466 :
467 :     // A/Dコンバータの設定
468 :     mstad = 0;           // A/Dコンバータを有効にする
469 :     admod = 0x03;       // 単発モードに設定
470 :     switch( ch ) {
471 :         case 0: adinsel = 0x00; break; // 入力端子AN0(P1_0)を選択
472 :         case 1: adinsel = 0x01; break; // 入力端子AN1(P1_1)を選択
473 :         case 2: adinsel = 0x40; break; // 入力端子AN2(P1_2)を選択
474 :         case 3: adinsel = 0x41; break; // 入力端子AN3(P1_3)を選択
475 :         case 4: adinsel = 0x80; break; // 入力端子AN4(P1_4)を選択
476 :         case 7: adinsel = 0x81; break; // 入力端子AN7(P1_7)を選択
477 :         default: mstad = 1; return 9999;
478 :     }
479 :     adcon0 = 0x01;      // A/D変換スタート
480 :
481 :     while( adcon0 & 0x01 ); // A/D変換終了待ち
482 :
483 :     switch( ch ) {
484 :         case 0: i = ad0; break; // 入力端子AN0(P1_0)を選択
485 :         case 1: i = ad1; break; // 入力端子AN1(P1_1)を選択
486 :         case 2: i = ad0; break; // 入力端子AN2(P1_2)を選択
487 :         case 3: i = ad1; break; // 入力端子AN3(P1_3)を選択
488 :         case 4: i = ad0; break; // 入力端子AN4(P1_4)を選択
489 :         case 7: i = ad1; break; // 入力端子AN7(P1_7)を選択
490 :     }
491 :
492 :     mstad = 1;         // A/Dコンバータを無効にする
493 :     return i;
494 : }
495 :
496 : //*****
497 : // 1文字受信
498 : // 引数 受信文字格納アドレス
499 : // 戻り値 -1:受信エラー 0:受信なし 1:受信あり 文字は*sに格納
500 : //*****
501 : int get_uart0( char *s )
502 : {
503 :     int ret = 0, data;
504 :     char c;
505 :
506 :     if (ri_u0c1 == 1){ // 受信データあり?
507 :         data = u0rb;
508 :         *s = (char)data;
509 :         ret = 1;
510 :         if( data & 0xf000 ) { // エラーあり?
511 :             // エラー時は再設定
512 :             re_u0c1 = 0;
513 :             te_u0c1 = 0;
514 :             c = u0mr;
515 :             u0mr &= 0xf8;
516 :             u0mr = c;
517 :             re_u0c1 = 1;
518 :             te_u0c1 = 1;
519 :             ret = -1;
520 :         }

```

7. プログラム「propo_relay.c」

```
521 :     }
522 :     return ret;
523 : }
524 :
525 : //*****
526 : // 1文字出力
527 : // 引数 送信データ
528 : // 戻り値 0:送信中のため、送信できず 1:送信セット完了
529 : //*****
530 : int put_uart0( char r )
531 : {
532 :     if(ti_u0c1 == 1) { // 送信データなし?
533 :         u0tbl = r;
534 :         return 1;
535 :     } else {
536 :         // 先に送信中(今回のデータは送信せずに終了)
537 :         return 0;
538 :     }
539 : }
540 :
541 : //*****
542 : // 文字列出力
543 : // 引数 送信データ
544 : // 戻り値 無し
545 : //*****
546 : void put_uart0_str( char *str )
547 : {
548 :     while(*str != '\0') {
549 :         // '\n'は'\r\n'に変換する
550 :         if( *str == '\n' ) while( !put_uart0( '\r' ) );
551 :         while( !put_uart0( *str ) ); // 1文字送信
552 :         str++;
553 :     }
554 : }
555 :
556 : //*****
557 : // 10進数出力
558 : // 引数 long 値, int 桁(1~8)
559 : // 戻り値 無し
560 : //*****
561 : void put_uart0_num( long value, int keta )
562 : {
563 :     int i = keta;
564 :     long temp = 0;
565 :
566 :     do {
567 :         temp <<= 4;
568 :         temp += value % 10;
569 :         value /= 10;
570 :     } while( --i );
571 :
572 :     do {
573 :         while( !put_uart0( (temp & 0xf) + '0' ) ); // 1文字送信
574 :         temp >>= 4;
575 :     } while( --keta );
576 : }
577 :
578 : //*****
579 : // 16進数出力
580 : // 引数 unsigned long 値, int 桁(1~8)
581 : // 戻り値 無し
582 : //*****
583 : void put_uart0_hex( unsigned long value, int keta )
584 : {
585 :     char temp;
586 :
587 :     do {
588 :         temp = value >> (keta * 4 - 4);
589 :         temp &= 0xf;
590 :         temp += (temp < 10) ? '0' : ('a'-10);
591 :
592 :         while( !put_uart0( temp ) ); // 1文字送信
593 :     } while( --keta );
594 : }
595 :
596 : //*****
597 : // end of file
598 : //*****
```

7.3 関数

■ch_a 関数

書式	void ch_a(int mode);
内容	チャンネル A のモータ、またはソレノイドバルブを制御します。
引数	int チャンネル A の動作 ●モータの場合 ●ソレノイドバルブの場合 -1:逆転 -1:下側 ON にする 0:停止 0:どちらも OFF 1:正転 1:上側 ON にする
戻り値	無し
例	ch_a(1); // 正転、または上側 ON ch_a(-1); // 逆転、または下側 ON ch_a(0); // 停止、またはどちらも OFF

■ch_b 関数

書式	void ch_b(int mode);
内容	チャンネル B のモータ、またはソレノイドバルブを制御します。
引数	int チャンネル B の動作 ●モータの場合 ●ソレノイドバルブの場合 -1:逆転 -1:下側 ON にする 0:停止 0:どちらも OFF 1:正転 1:上側 ON にする
戻り値	無し
例	ch_b(1); // 正転、または上側 ON ch_b(-1); // 逆転、または下側 ON ch_b(0); // 停止、またはどちらも OFF

■get_ad 関数

書式	int get_ad(int ch);
内容	A/D 変換値を取得します。値は 0~1023 になります。0 が 0V、1023 が電源電圧(5V)になります。
引数	int チャンネル番号 A/D 値を取得する端子名を指定します。 TB6_AD : TB6 の入力端子の A/D 値を取得します TB7_AD : TB6 の入力端子の A/D 値を取得します
戻り値	無し
例	get_ad(TB6_AD); // TB6 の入力端子の電圧を測定(0~1023) get_ad(TB7_AD); // TB7 の入力端子の電圧を測定(0~1023)

■put_uart0_str 関数

書式	void put_uart0_str(char *str);
内容	UART0 に文字列を出力します。RY-WRITER 基板を使ってパソコンと接続、パソコンには TeraTerm などの通信ソフトを使ってマイコンと接続すると、TeraTerm 上に文字が表示されます。デバッグとして使用します。
引数	char* 文字列 出力したい文字列を指定します。
戻り値	無し
例	<pre>// パソコンの TeraTerm などの通信ソフトに、文字列を表示 put_uart0_str("¥n¥nPUROPO Relay PCB Ver.1¥n¥n");</pre>

■put_uart0_num 関数

書式	void put_uart0_num(long value, int keta);
内容	UART0 に値を 10 進数で出力します。デバッグとして使用します。
引数	<p>●long value UART0 で出力する変数を設定します。 値は、0～99,999,999 (8 桁)まで変換できます。マイナスの値は文字化けするので、プログラムでプラスの値に直してから、この関数で表示させてください。</p> <p>●int keta 出力したい桁数を設定します 1～8 桁まで設定できます。</p>
戻り値	無し
例	<pre>i = 12345; put_uart0_num(i , 6); // 「012345」と表示されます i = -1234; if(i > 0) { put_uart0_num(i , 4); } else { put_uart0_str("-"); // 「-」を表示 put_uart0_num(-i , 4); // 「1234」と表示 合わせて「-1234」と表示</pre>

■put_uart0_hex 関数

書式	void put_uart0_hex(unsigned long value, int keta);
内容	UART0 に値を16進数で出力します。デバッグとして使用します。
引数	<p>●unsigned long value UART0 で出力する変数を設定します。 値は、0～ffffffff (8 桁)まで変換できます。</p> <p>●int keta 出力したい桁数を設定します 1～8 桁まで設定できます。</p>
戻り値	無し
例	<pre>i = 0xabcd; put_uart0_hex (i , 6); // 「01abcd」と表示されます</pre>

7.4 プログラムの解説

7.4.1 グローバル変数部分

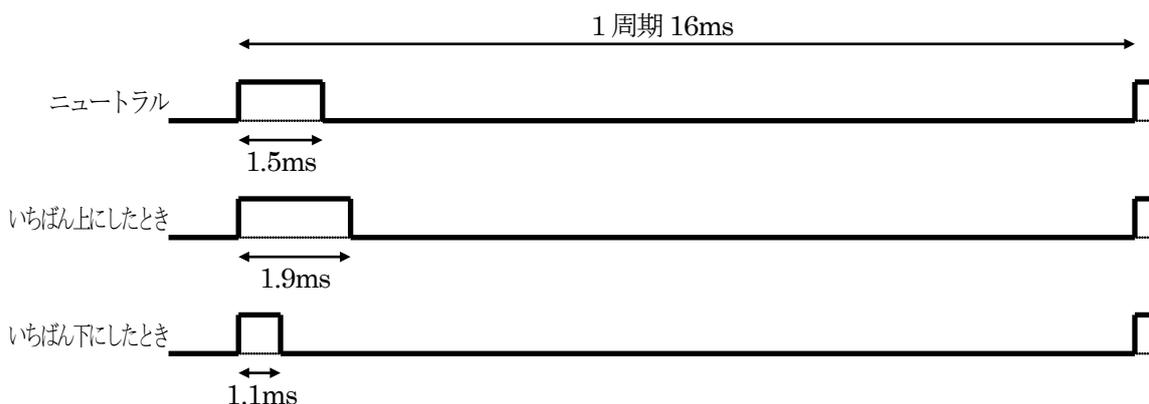
```

68 : //=====
69 : // グローバル変数
70 : //=====
71 : int cnt_rb;                // 1ms ごとに+ 1 する変数
72 :
73 : long pulse_a;            // プロポ受信信号 CH_A パルス幅
74 : long pulse_b;            // プロポ受信信号 CH_B パルス幅
75 :
76 : int pulse_a_inc;          // プロポ受信信号 CH_A 受信回数(確認用)
77 : int pulse_b_inc;          // プロポ受信信号 CH_B 受信回数(確認用)
78 :
79 : int cnt_a;                // パルス幅 CH_A 受信が無い時間
80 : int cnt_b;                // パルス幅 CH_B 受信が無い時間
    
```

73 行	プロポ受信機入力コネクタのチャンネル A (CN1) に入力したパルスの ON 幅が代入される変数です (実際は、反転した信号をマイコンで入力するので OFF 幅を測定しています)。この変数は、1 あたりパルス幅が、 $0.4 \mu s$ です。パルス幅が、 $1.5ms (=1500 \mu s)$ の場合は、変数の値 $=1500 / 0.4 = 3750$ となります。
74 行	プロポ受信機入力コネクタのチャンネル B (CN2) に入力したパルスの ON 幅が代入される変数です (実際は、反転した信号をマイコンで入力するので OFF 幅を測定しています)。この変数の計算は、pulse_a 変数と同様です。

プロポの操作と、パルス幅の関係を下記に示します。

※SANWA のプロポ「Aquila-6」(アキーラ-6) の解析結果です。その他のメーカーは異なることがあります。



7. プログラム「propo_relay.c」

パルス幅と、変数の値の関係を、下表に示します。

プロポのジョイスティック	PWM の ON 幅	pulse_a、pulse_b の計算
ニュートラル(中心)	1.5[ms] (1500[μs])	変数は、1 あたりパルス幅が、0.4 μs です。よって、 変数の値=ON 幅 / 0.4 μs = 1500 / 0.4 = 3750 となります。
いちばん上にしたとき	1.9[ms] (1900[μs])	変数の値=ON 幅 / 0.4 μs = 1900 / 0.4 = 4750 となります。
いちばん下にしたとき	1.1[ms] (1100[μs])	変数の値=ON 幅 / 0.4 μs = 1100 / 0.4 = 2750 となります。

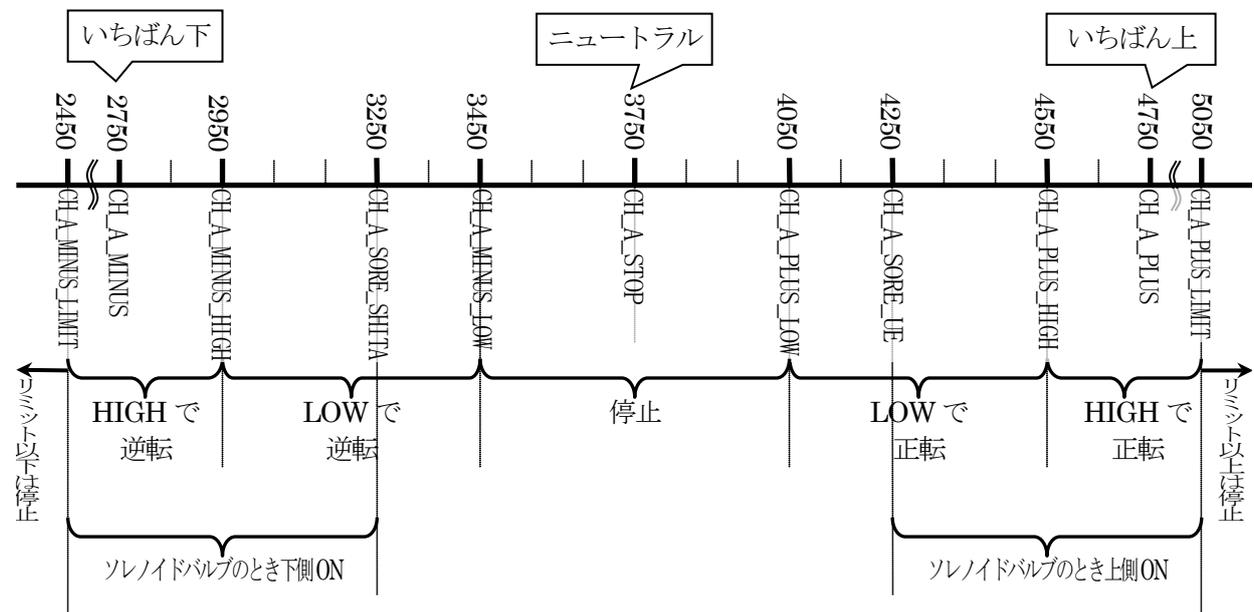
pulse_a 変数、pulse_b 変数の値を if 文などでチェックすることによって、モータ、またはソレノイドバルブを動作させます。

7.4.1 シンボル定義(チャンネル A のパルス幅定義部分)

```

17 : // CH_A のプロポ信号 受信パルス幅
18 : #define CH_A_PLUS_LIMIT    5050    // これ以上は無効とする(CH_A_PLUS+300くらい)
19 : #define CH_A_PLUS          4750    // いちばん上にしたとき
20 : #define CH_A_PLUS_HIGH    4570    // HIGHモードで正転
21 : #define CH_A_PLUS_LOW     4050    // LOWモードで正転
22 : #define CH_A_STOP         3750    // プロポスティック ニュートラル
23 : #define CH_A_MINUS_LOW    3450    // LOWモードで逆転
24 : #define CH_A_MINUS_HIGH   2950    // HIGHモードで逆転
25 : #define CH_A_MINUS        2750    // いちばん下にしたとき
26 : #define CH_A_MINUS_LIMIT  2450    // これ以下は無効とする(CH_A_MINUS-300くらい)
27 :
28 : #define CH_A_SORE_UE       4250    // ソレノイドバルブ 上側ON
29 : #define CH_A_SORE_SHITA   3250    // ソレノイドバルブ 下側ON
    
```

「CH_A_xxxx」は、パルス幅に対して、モータをどう制御するか定義しています。サンプルプログラムは、下記のようになっていますので、適宜値を書き換えて、自分のロボットに合うようにしてください。



7.4.2 main 関数

```

82 : //*****
83 : // メインプログラム
84 : //*****
85 : void main( void )
86 : {
87 :     init();                // 初期化
88 :     asm("FSET I");        // 全体割り込み許可
89 :
90 :     put_uart0_str( "YnYnPUROPO Relay PCB Ver. 1YnYn" );
91 :
92 :     while( 1 ) {
93 :         if( cnt_rb >= 200 ) {
94 :             cnt_rb = 0;
95 :
96 :             put_uart0_str( "A=" );
97 :             put_uart0_num( pulse_a, 5 );
98 :             put_uart0_str( "(" );
99 :             put_uart0_num( pulse_a_inc, 5 );
100 :            put_uart0_str( ") B=" );
101 :            put_uart0_num( pulse_b, 5 );
102 :            put_uart0_str( "(" );
103 :            put_uart0_num( pulse_b_inc, 5 );
104 :            put_uart0_str( ") TB6=" );
105 :            put_uart0_num( p1_0, 1 );
106 :            put_uart0_str( "(" );
107 :            put_uart0_num( get_ad(TB6_AD), 4 );
108 :            put_uart0_str( ") TB7=" );
109 :            put_uart0_num( p1_1, 1 );
110 :            put_uart0_str( "(" );
111 :            put_uart0_num( get_ad(TB7_AD), 4 );
112 :            put_uart0_str( ")") );
113 :
114 :            put_uart0_str( "Yr" );    // 先頭に戻る
115 :        }
116 :    }
117 : }

```

main 関数内では、TeraTerm などの通信ソフトへ、デバッグ用の内容を入力するプログラムです。表示例を下記に示します。

```

PUROPO Relay PCB Ver. 1

A=03759(00297) B=03757(00297) TB6=1(1023) TB7=1(0914)
  ①   ②       ③   ④           ⑤   ⑥       ⑦   ⑧

```

①～⑧の内容について、下表に示します。

①	プロポ受信機入力コネクタ A から受信したパルスのパルス幅を表示します。
②	プロポ受信機入力コネクタ A から受信したパルスの受信回数を表示します。
③	プロポ受信機入力コネクタ B から受信したパルスのパルス幅を表示します。
④	プロポ受信機入力コネクタ B から受信したパルスの受信回数を表示します。

7. プログラム「propo_relay.c」

⑤	TB6 の入力レベルを“0”か“1”かで表示します。
⑥	TB6 に入力されている電圧を A/D 変換した値 (0~1023) で表示します。 0=0V、1023=5V(電源電圧)です。アナログセンサを接続したときに、A/D 値を使います。
⑦	TB7 の入力レベルを“0”か“1”かで表示します。
⑧	TB7 に入力されている電圧を A/D 変換した値 (0~1023) で表示します。 0=0V、1023=5V(電源電圧)です。アナログセンサを接続したときに、A/D 値を使います。

プロポのジョイスティックをいちばん**上**にしたときの例を下記に示します。

```
A=04764 (00217) B=03761 (00217) TB6=1 (1022) TB7=1 (0913)
```

パルス幅は、 $4764 * 0.4 \mu s = 1.9056ms$ となります。

プロポのジョイスティックをいちばん**下**にしたときの例を下記に示します。

```
A=02741 (00136) B=03758 (00137) TB6=1 (1023) TB7=1 (0908)
```

パルス幅は、 $2741 * 0.4 \mu s = 1.0964ms$ となります。

7.4.3 調整のポイント

プログラムのシンボル定義部分を、Tera Term で表示された実際の値に変えてください。

```
17 : // CH_A のプロポ信号 受信パルス幅
18 : #define CH_A_PLUS_LIMIT    5050    // 「CH_A_PLUS」より300くらい大きい値にして
                                         // ください これ以上の値は、不正な値としてモ
                                         // ータ(ソレノイドバルブ)を停止します
19 : #define CH_A_PLUS        4750    // スティックをいちばん上 (右) にしたときの
                                         // 値にしてください
20 : #define CH_A_PLUS_HIGH    4550    // HIGHモードで正転する値ににしてください
21 : #define CH_A_PLUS_LOW     4050    // LOWモードで正転する値ににしてください
22 : #define CH_A_STOP        3750    // プロポスティックを何も操作していないとき
                                         // の値 (ニュートラル) にしてください
23 : #define CH_A_MINUS_LOW    3450    // LOWモードで逆転する値ににしてください
24 : #define CH_A_MINUS_HIGH   2950    // HIGHモードで逆転する値ににしてください
25 : #define CH_A_MINUS        2750    // スティックをいちばん下 (左) にしたときの
                                         // 値にしてください
26 : #define CH_A_MINUS_LIMIT  2450    // 「CH_A_MINUS」より300くらい小さい値にして
                                         // ください これ以上の値は、不正な値として
                                         // モータ(ソレノイドバルブ)を停止します
27 :
28 : #define CH_A_SORE_UE       4250    // ソレノイドバルブの上側をONにする値にして
                                         // ください
29 : #define CH_A_SORE_SHITA    3250    // ソレノイドバルブの下側をONにする値にして
                                         // ください
```

CH_B の値も同様に変更してください。

モータの回転を逆にしたいときは、モータの配線を入れ替えてください。

215～247 行が、チャンネル A で DC モータを動かすプログラムです。

```

215 :         if( cnt_a >= 200 ) { パルス入力が200ms以上なければ
216 :             // 指定時間以上、パルス幅入力がなれば停止 予期せぬ状態と判断し停止
217 :             speed_high_a = 0;
218 :             ch_a( 0 );
219 :         } else if( pulse_a >= CH_A_PLUS_LIMIT ) { リミット値以上なら予期せぬ状態
220 :             // リミット以上なら停止 と判断し、停止
221 :             speed_high_a = 0;
222 :             ch_a( 0 );
223 :         } else if( pulse_a >= CH_A_PLUS_HIGH ) { HIGH以上ならHIGHで正転
224 :             // HIGHで正転
225 :             speed_high_a = 1;
226 :             ch_a( 1 );
227 :         } else if( pulse_a >= CH_A_PLUS_LOW ) { LOW以上ならHIGHで正転
228 :             // LOWで正転
229 :             speed_high_a = 0;
230 :             ch_a( 1 );
231 :         } else if( pulse_a <= CH_A_MINUS_LIMIT ) { リミット値以下なら予期せぬ状態
232 :             // リミット以下なら停止 と判断し、停止
233 :             speed_high_a = 0;
234 :             ch_a( 0 );
235 :         } else if( pulse_a <= CH_A_MINUS_HIGH ) { HIGH以下ならHIGHで逆転
236 :             // HIGHで逆転
237 :             speed_high_a = 1;
238 :             ch_a( -1 );
239 :         } else if( pulse_a <= CH_A_MINUS_LOW ) { LOW以下ならHIGHで逆転
240 :             // LOWで逆転
241 :             speed_high_a = 0;
242 :             ch_a( -1 );
243 :         } else { それ以外は停止
244 :             // それ以外は停止
245 :             speed_high_a = 0;
246 :             ch_a( 0 );
247 :         }

```

250～265 行が、チャンネル A でソレノイドバルブを動かすプログラムです。

```
250 :         if( cnt_a >= 200 ) {
251 :             // 指定時間以上、パルス幅入力があれば何もしない
252 :         } else if( pulse_a >= CH_A_PLUS_LIMIT ) {
253 :             // リミット以上なら何もしない
254 :         } else if( pulse_a >= CH_A_SORE_UE ) {
255 :             // ソレノイドバルブ、上側ONにする
256 :             ch_a( 1 );
257 :         } else if( pulse_a <= CH_A_SORE_SHITA ) {
258 :             // ソレノイドバルブ、下側ONにする
259 :             ch_a( -1 );
260 :         } else if( pulse_a <= CH_A_MINUS_LIMIT ) {
261 :             // リミット以下なら何もしない
262 :         } else {
263 :             // それ以外はOFF
264 :             ch_a( 0 );
265 :         }
```

チャンネル A、チャンネル B の両方とも DC モータの場合、LOW と HIGH のどちらか検出し、LOW なら DC モータにはバッテリー 1 本分の電圧 (7.2～7.4V×1) を加えます。HIGH なら DC モータにはバッテリー 2 本分の電圧 (7.2～7.4V×2) を加えます。

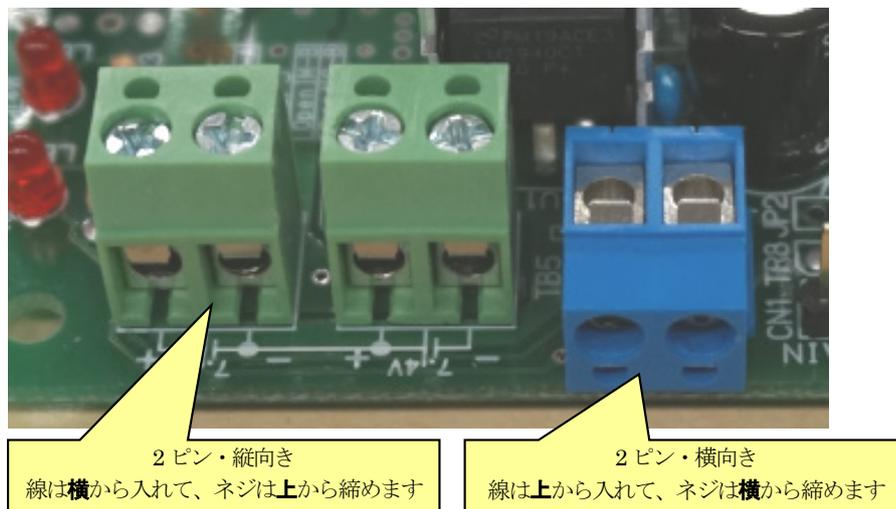
```
324 :         // RL5 のリレー制御
325 :         if( p4_2 == 1 && p4_7 == 1 ) {
326 :             // チャンネル A、チャンネル B、両方ともモータモードなら
327 :             if( speed_high_a == 1 || speed_high_b == 1 ) {
328 :                 // CH_A と CH_B のどちらかが HIGH モードなら、HIGH にする
329 :                 p1_3 = 1;
330 :             } else {
331 :                 // どちらも LOW モードなら、LOW にする
332 :                 p1_3 = 0;
333 :             }
334 :         } else {
335 :             // 1つ以上ソレノイドバルブなら、リレーは OFF にする
336 :             p1_3 = 0;
337 :         }
```

モータ B を動作させるプログラムは、上記「CH_A」が「CH_B」に変わるだけなので、説明は省略します。

8. 付録

8.1 ターミナルブロックについて

ターミナルブロックは、縦向き、横向きがあります。組み込んだときに、ネジが締めやすいターミナルブロックを選んでください。

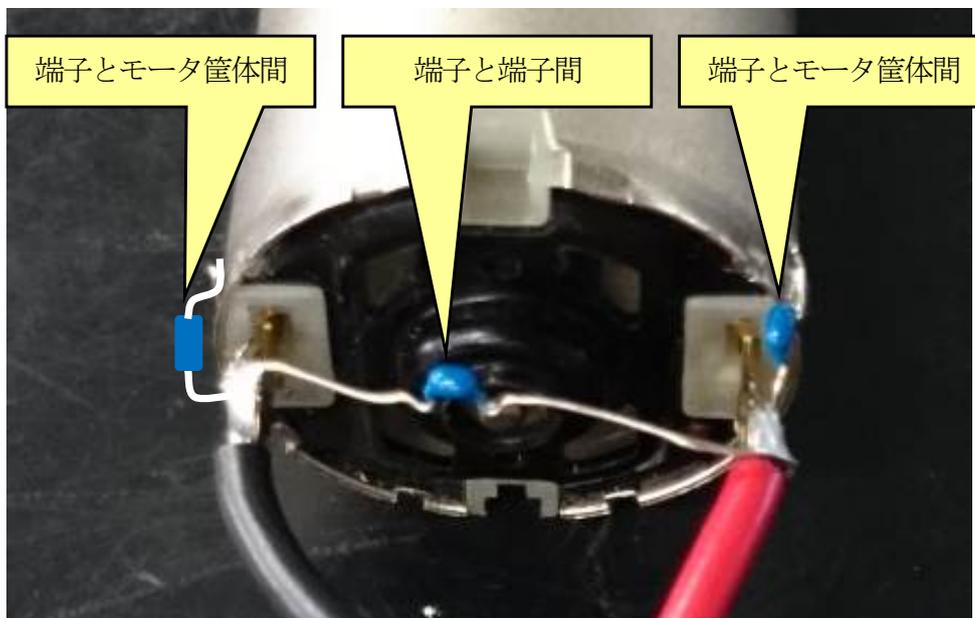


ターミナルブロックは、(株)秋月電子通商で販売しています。ピン数、向きと、型式の関係を下記に示します。

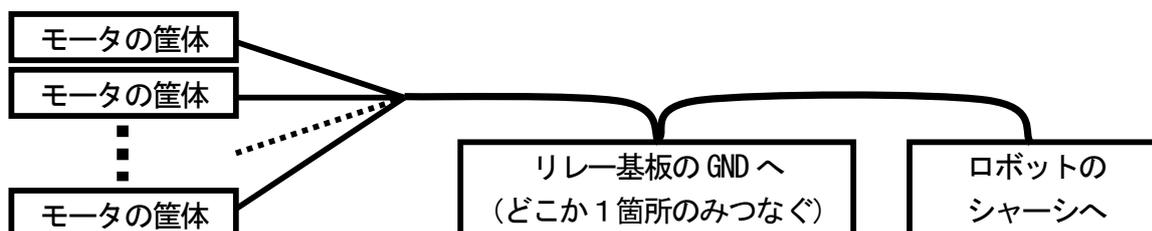
端子の色、向き		2ピン	3ピン
緑	縦向き	TB112-2-2-E-1 通販コード P-01308	TB112-2-3-E-1 通販コード P-01310
	横向き	TB112a-2-2-E-1 通販コード P-01312	TB112a-2-3-E-1 通販コード P-01314
青	縦向き	TB112-2-2-U-1 通販コード P-01309	TB112-2-3-U-1 通販コード P-01311
	横向き	TB112a-2-2-U-1 通販コード P-01313	TB112a-2-3-U-1 通販コード P-01315

8.2 モータの加工

モータを回すと、かなりノイズが発生し、マイコンが誤動作します。そこで、モータ1個につき、 $0.01\mu\text{F}$ ~ $0.1\mu\text{F}$ のコンデンサを3個取り付け、ノイズを吸収させます。



それでもモータを回したときにリレーが誤動作するときは、全モータの筐体間を線で結び、その結んだ線をリレー基板の GND とロボットのシャーシに繋いでください。



9. 参考文献

- ルネサス エレクトロニクス(株)
R8C/38C グループ ユーザーズマニュアル ハードウェア編 Rev.1.10
- ルネサス エレクトロニクス(株)
R8C/M11A グループ、R8C/M12A グループ ユーザーズマニュアル ハードウェア編 Rev.1.00
- ルネサス エレクトロニクス(株)
M16C シリーズ,R8C ファミリー用 C/C++コンパイラパッケージ V.6.00
C/C++コンパイラユーザーズマニュアル Rev.1.00
- ルネサス エレクトロニクス(株)
High-performance Embedded Workshop V.4.09 ユーザーズマニュアル Rev.1.00
- ルネサス半導体トレーニングセンター C言語入門コーステキスト 第1版
- 電波新聞社 マイコン入門講座 大須賀威彦著 第1版
- ソフトバンク(株) 新C言語入門シニア編 林晴比古著 初版
- 共立出版(株) プログラマのための ANSI C 全書 L.Ammeraal 著
吉田敬一・竹内淑子・吉田恵美子訳 初版

マイコンカーラリー、販売部品についての詳しい情報は、マイコンカーラリー販売サイトをご覧ください。

<https://www2.himdx.net/mcr/>

R8C マイコンについての詳しい情報は、ルネサス エレクトロニクスのホームページをご覧ください。

<http://japan.renesas.com/>

の「製品情報」欄→「マイコン」→「R8C」でご覧頂けます