

# プロポ信号受信 FET 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 FET モータドライブ基板
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 1.02 版

2016.02.08

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

<b>1. 概要</b> .....	<b>1</b>
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 基板、コネクタ寸法.....	5
1.6 部品面パターン図.....	6
1.7 半田面パターン図.....	7
1.8 回路図.....	8
<b>2. 組み立てに必要な工具、部品</b> .....	<b>10</b>
2.1 工具.....	10
2.2 セット以外で組み立てに必要な部品.....	12
<b>3. プロポ FET 基板の組み立て</b> .....	<b>13</b>
3.1 部品表.....	13
3.2 部品面のシルク.....	17
3.3 抵抗(4.7k $\Omega$ )の取り付け.....	18
3.4 抵抗(3.3k $\Omega$ )の取り付け.....	19
3.5 抵抗(1k $\Omega$ )の取り付け.....	20
3.6 抵抗(100 $\Omega$ )の取り付け.....	21
3.7 積層セラミックコンデンサ(1 $\mu$ F)の取り付け.....	22
3.8 積層セラミックコンデンサ(10 $\mu$ F)の取り付け.....	23
3.9 汎用整流用ダイオード(1N4007)の取り付け.....	24
3.10 ショットキーバリアダイオード(11EQS04)の取り付け.....	25
3.11 R8C マイコン(M120AN)の取り付け.....	26
3.12 ハーフブリッジドライバ(IR2302PBF)の取り付け.....	27
3.13 LED(EBR3338S)の取り付け.....	28
3.14 タクトスイッチ(B3F-1000)の取り付け.....	29
3.15 スライドスイッチ(SS-12D00-G5)の取り付け.....	30
3.16 XH コネクタ 4ピンオス(B4B-XH-A)の取り付け.....	31
3.17 抵抗内蔵トランジスタ(RN1201)の取り付け.....	32
3.18 ピンヘッダ(1列 $\times$ 40P)の取り付け.....	33
3.19 ターミナルブロック(2ピン・緑・縦)の取り付け.....	34
3.20 ターミナルブロック(3ピン・青・縦)の取り付け.....	35
3.21 ターミナルブロック(2ピン・青・縦)の取り付け.....	36
3.22 電解コンデンサ(1000 $\mu$ F)の取り付け.....	37
3.23 電気二重層コンデンサ(1.5F)の取り付け.....	38
3.24 三端子レギュレータ(LM2940CT-5.0)の取り付け.....	39
3.25 Nch パワーFET(2SK3163)の取り付け.....	41
3.26 プロポ受信機接続ケーブルの作成.....	44
3.27 パターンの強化.....	46
<b>4. 接続</b> .....	<b>47</b>
4.1 接続図.....	47
4.2 マイコンボードの電源を別にする.....	49

4.3 複数のプロポ FET 基板を使って、多くのモータを制御する .....	50
4.4 センサの接続方法 .....	51
4.4.1 NPN 出力タイプの場合 (12V 動作のセンサ) .....	51
4.4.2 2 線式の場合 (12V 動作のセンサ) .....	52
4.4.3 スイッチを接続する場合 .....	53
4.4.4 5V で動作するアナログ出力のセンサの場合 .....	53
<b>5. 動作確認 .....</b>	<b>54</b>
5.1 動作確認の結線 .....	54
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 .....	55
5.3 動作確認 .....	58
<b>6. ワークスペース「r8cm12a_propo_fet」 .....</b>	<b>59</b>
6.1 プロジェクトの構成 .....	59
<b>7. プログラム「propo_fet.c」 .....</b>	<b>60</b>
7.1 概要 .....	60
7.2 プログラムリスト .....	60
7.3 関数 .....	68
7.4 プログラムの解説 .....	70
7.4.1 グローバル変数部分 .....	70
7.4.1 シンボル定義(チャンネル A のパルス幅定義部分) .....	71
7.4.2 main 関数 .....	72
7.4.3 調整のポイント .....	73
7.4.4 モータを動作させている部分 .....	74
<b>8. 付録 .....</b>	<b>76</b>
8.1 ターミナルブロックについて .....	76
8.2 モータの加工 .....	77
<b>9. 参考文献 .....</b>	<b>78</b>

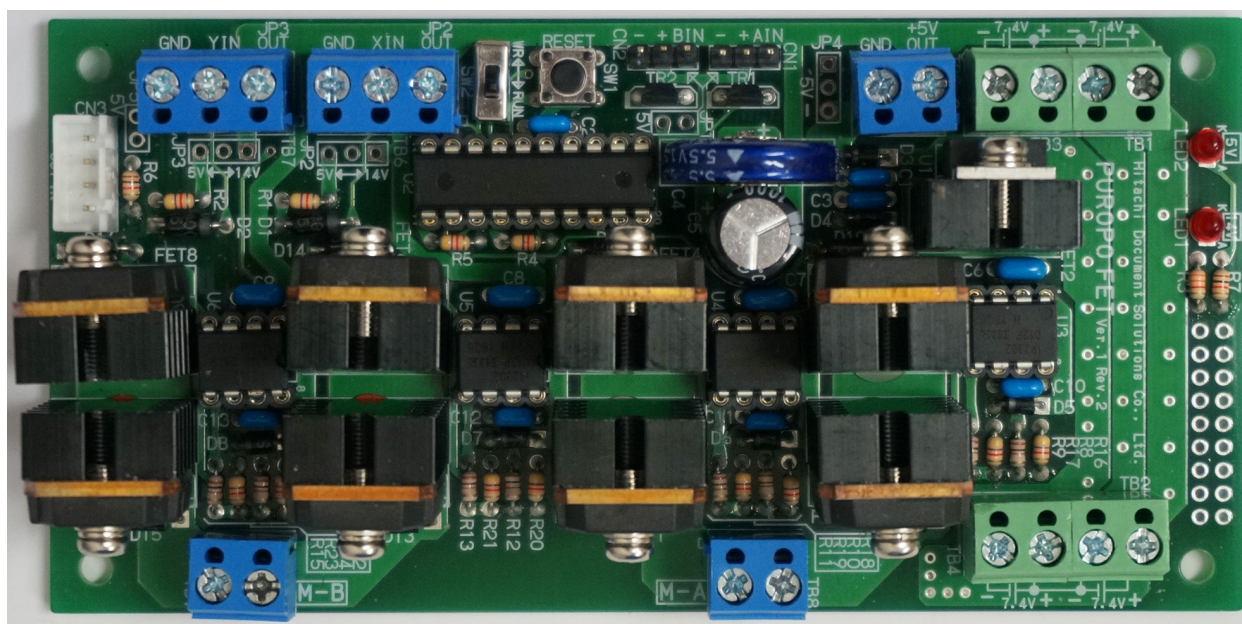
## 1. 概要

本マニュアルは、プロポ信号受信 FET モータドライブ基板(以下、プロポ FET 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 FET 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 FET 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。ただし、プロポ FET 基板自体は、7~20Vで作動しますので、この範囲の電圧であればどのような電源でも問題ありません。
- FET(2SK3163)は75A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 FET モータドライブ基板

1. 概要

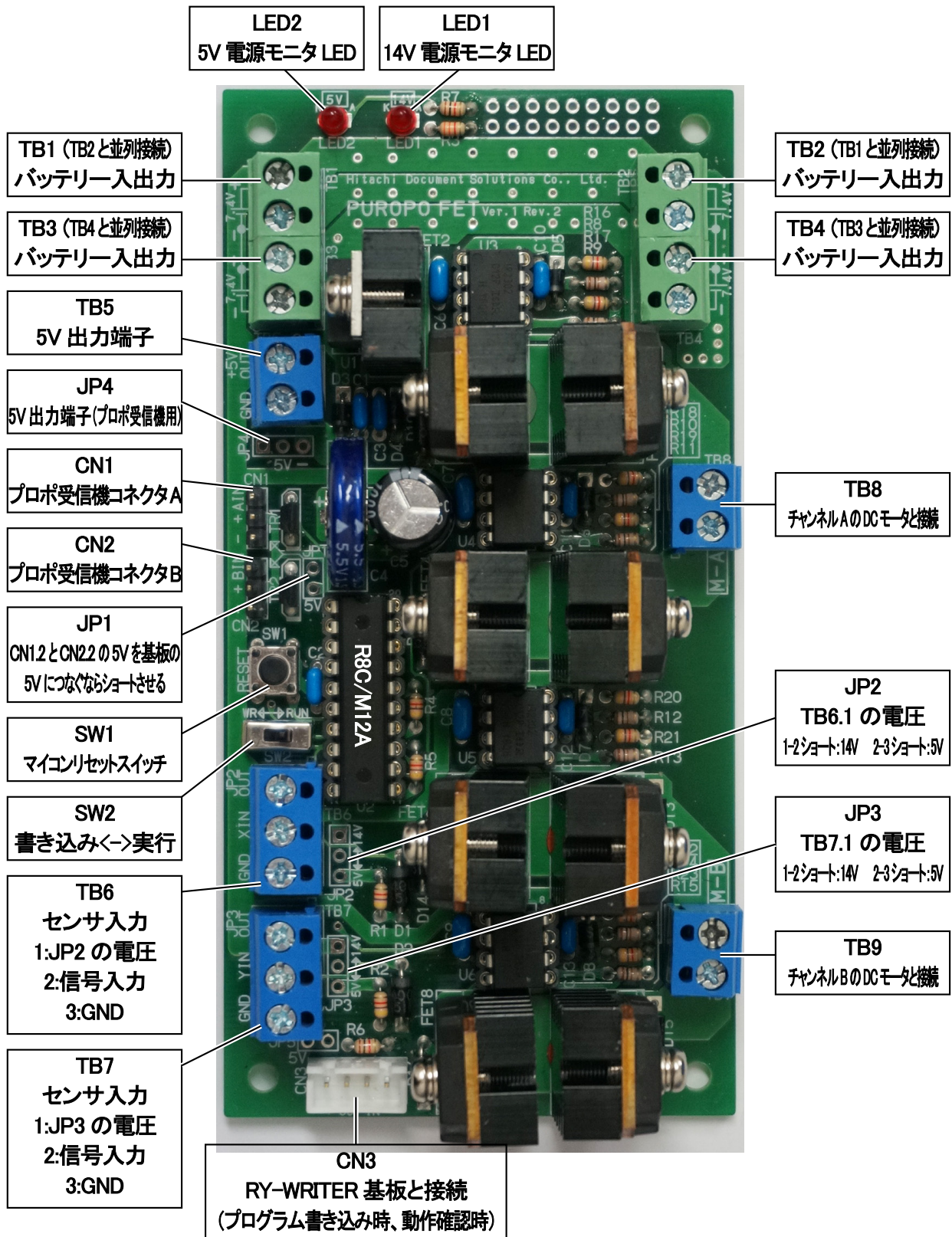
1.2 仕様

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
制御できるモータ数	2 個
モータの制御周期、分解能	制御周期(PWM 周期):10[ms] 分解能:正転、逆転ともに 10%ごと(0%、±10%、±20%、……、±90%、±100%)
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 27mm
重量(完成品の実測)	約 170g ※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

---

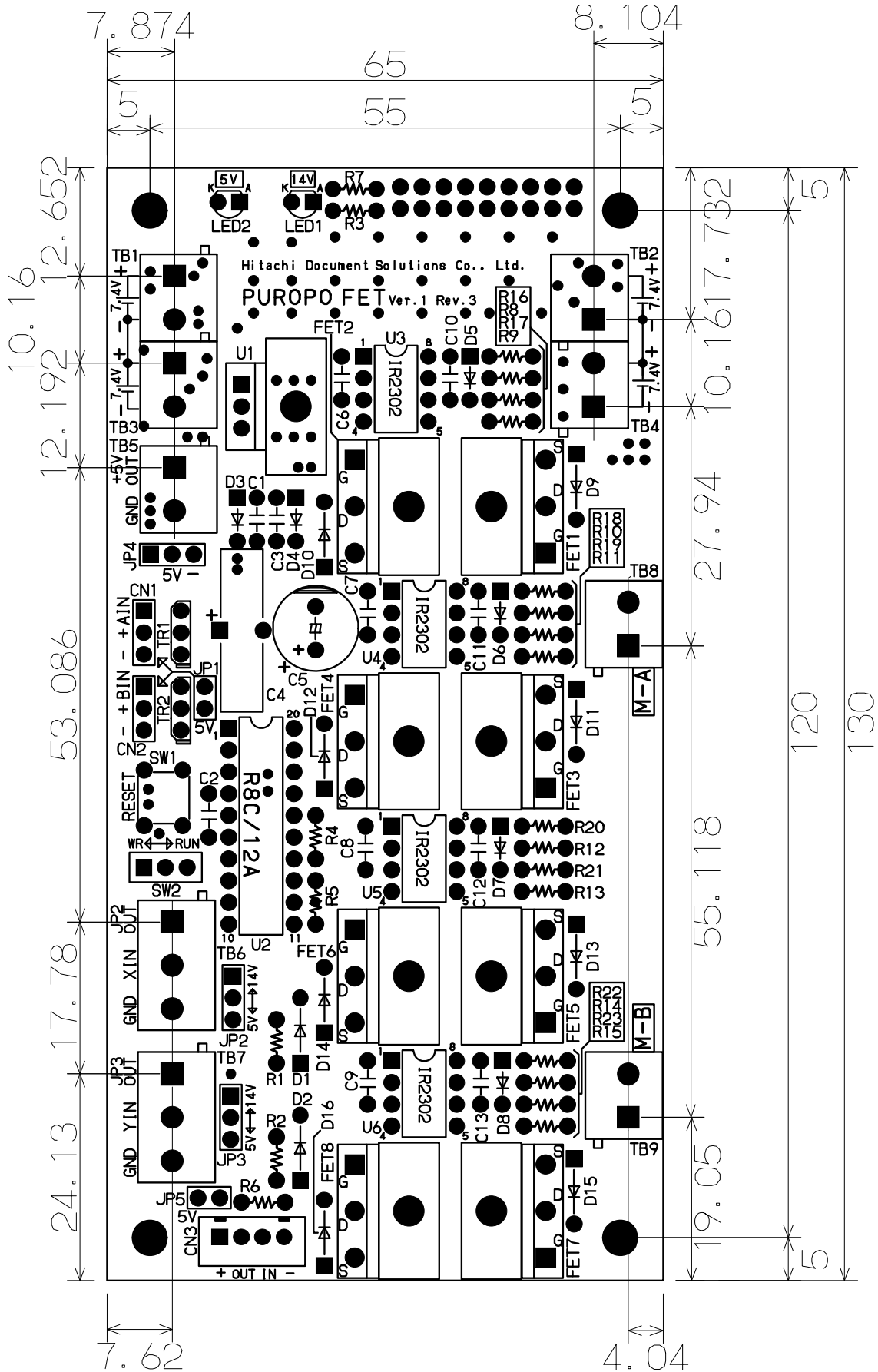
1.4 各部(コネクタ、スイッチなど)の説明

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。
JP2	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。
JP3	JP2と同じで、TB7が対象となります。

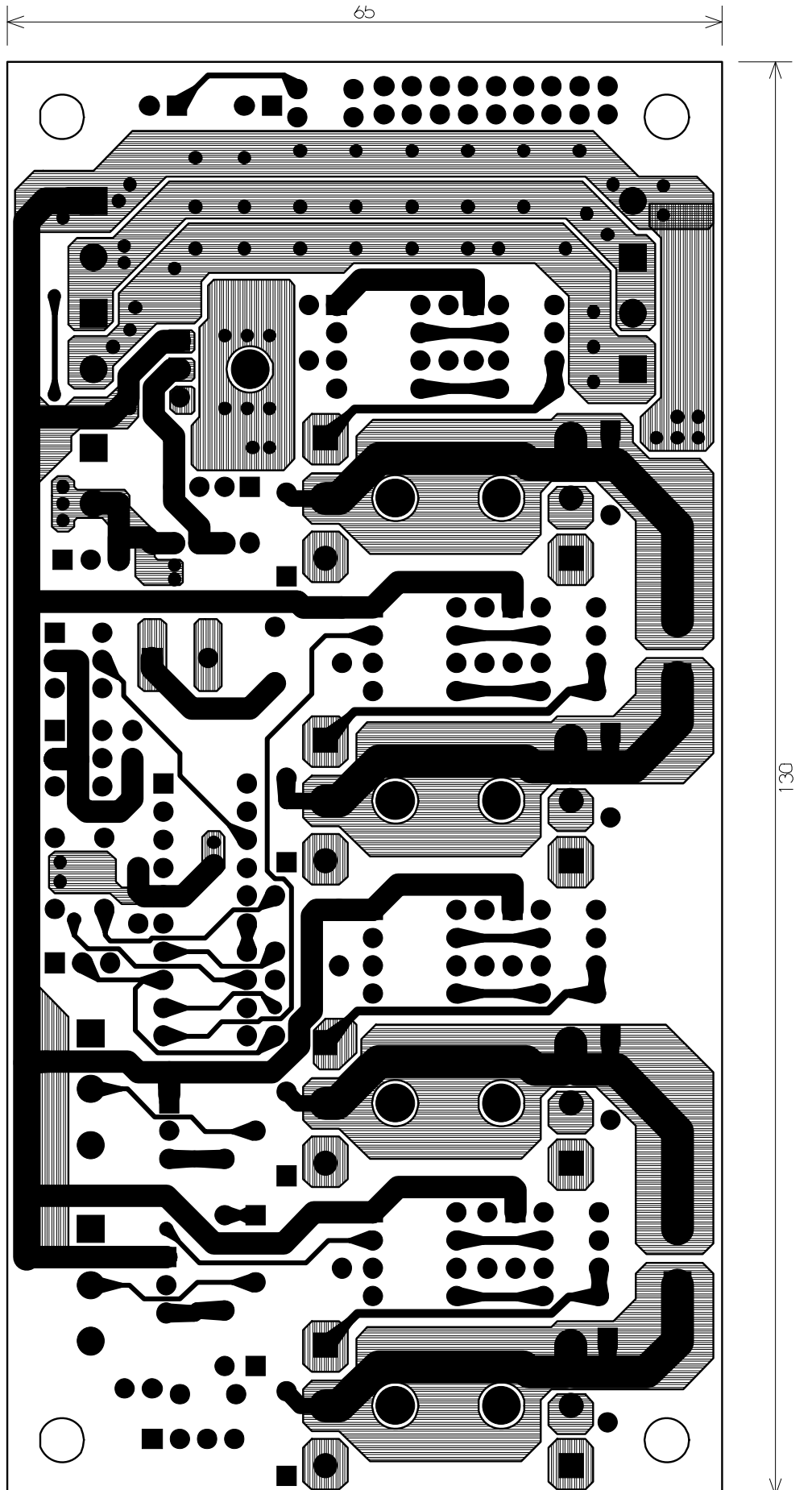


1. 概要

1.5 基板、コネクタ寸法

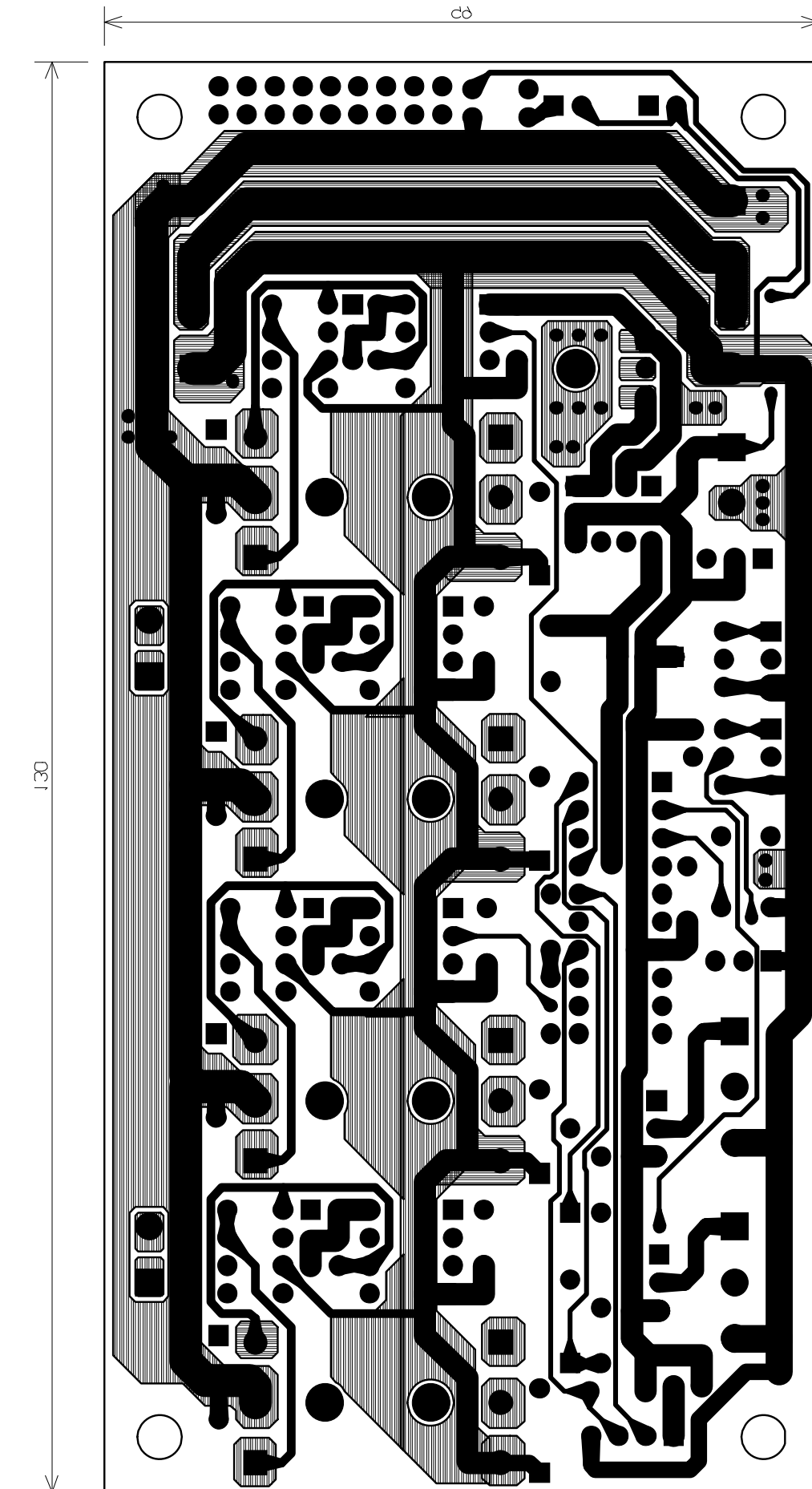


1.6 部品面パターン図



1. 概要

1.7 半田面パターン図



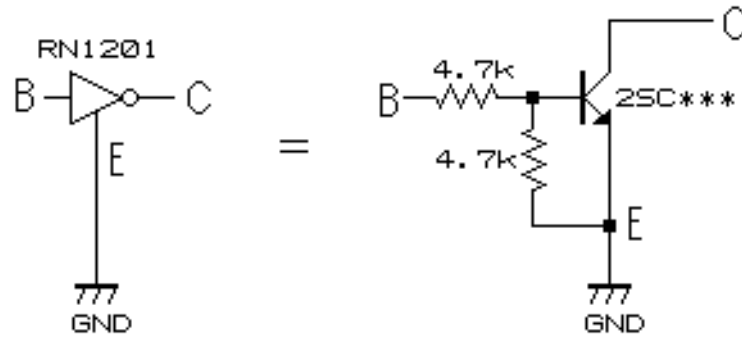


1. 概要

---

※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

プロポ FET 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 (<a href="https://www2.himdx.net/mcr">https://www2.himdx.net/mcr</a>) で販売しています。</p> <ul style="list-style-type: none"> <li>●M-S327: はんだこて(HAKKO FX-600-02)</li> <li>●M-S328: はんだこて台(HAKKO 633-01)</li> </ul>
<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。マイコンカーラー販売では、下記を取りそろえています。</p> <ul style="list-style-type: none"> <li>•M-S329:糸半田(千住金属 スパークルソルダ φ0.6 150g)</li> <li>•M-S330:糸半田(千住金属 スパークルソルダ φ0.8 150g)</li> <li>•M-S331:糸半田鉛フリー(千住金属 ESCF3M705 φ0.6 100g)</li> <li>•M-S332:糸半田鉛フリー(千住金属 ESCF3M705 φ0.8 100g)</li> </ul>
<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 FET 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_FET_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, 16,17,18,19, 20,21,22,23	抵抗	CFS1/4C 4.7kΩ (黄・紫・赤・金)		コーア(株)	12
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
R8, 9,10,11, 12,13,14,15	抵抗	CFS1/4C 100Ω (茶・黒・茶・金)		コーア(株)	8
C1, 2, 3,10, 11,12,13	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	7
C6, 7, 8, 9	積層セラミックコンデンサ	RDEC71H106K3K1H03B 10 μ F/50V 5.08mm ピッチ		(株)村田製作所	4
D1, 2, 9,10, 11,12,13,14, 15,16	汎用整流用ダイオード	1N4007 1000V/1A		PANJIT INTERNATIONAL INC.	10
D3, 4, 5, 6, 7, 8	ショットキーバリアダイオード	11EQS04 40V/1A V <sub>FM</sub> (せん頭順電圧):0.55V		日本インター(株)	6

3. プロポ FET 基板の組み立て

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
U3, 4, 5, 6	ハーフブリッジ ドライバ	IR2302PBF	 0のへこみがある側の手前が1ピン	International Rectifier	4
LED1,2	LED	EBR3338S φ3mm・赤	 A(長) (短)K	スタンレー電気(株)	2
SW1	タクトスイッチ	B3F-1000		オムロン(株)	1
SW2	スライドスイッチ	SS-12D00-G5		Zhejiang Jianfu Electronics Co., Ltd	1
CN3	XHコネクタ4ピンオス	B4B-XH-A		日本圧着端子製造(株)	1
TR1, 2	抵抗内蔵トランジスタ	RN1201		(株)東芝セミコンダクター社	2
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
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

3. プロポ FET 基板の組み立て

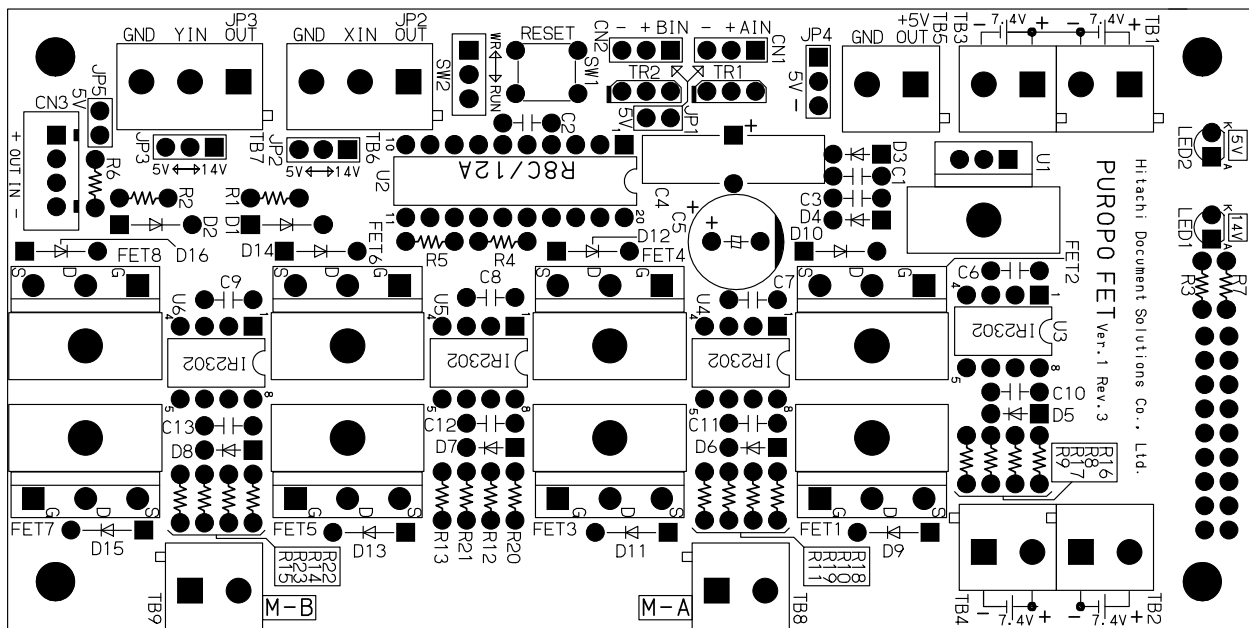
TB6, 7	ターミナルブロック 3ピン(青)(縦)	TB112-2-3-U-1		Alphaplus Connectors & Cables Corp.	2
TB5, 8, 9	ターミナルブロック 2ピン(青)(縦)	TB112-2-2-U-1		Alphaplus Connectors & Cables Corp.	3
C5	電解コンデンサ	SMG160E102MJ16S 1000 $\mu$ F/16V		日本ケミコン (株)	1
C4	電気二重層 コンデンサ	SE-5R5-D155VYV 1.5F/5.5V(タテ型) ※プラス、マイナスは、 スポット溶接部分の 形で判断します		KAMCAP	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
FET1,2,3, 4, 5, 6, 7, 8	Nch パワーFET	2SK3163 60V75A		ルネサス エレ クトロニクス(株)	8
FET1~8 の 放熱器の下 に入れる ※U1の放熱器の下 には入れません	ベークワッシャ	XX-0306-05 内径 $\phi$ 3、外径 $\phi$ 6 厚さ 0.5mm		(株)廣杉計器	8
U1、 FET1~8 用	放熱器	PC2116-7-PB		(株)放熱器のオ ーエス	9

3. プロポ FET 基板の組み立て

放熱器 固定用	座金ネジ	φ 3×8mm		各社	9
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 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。

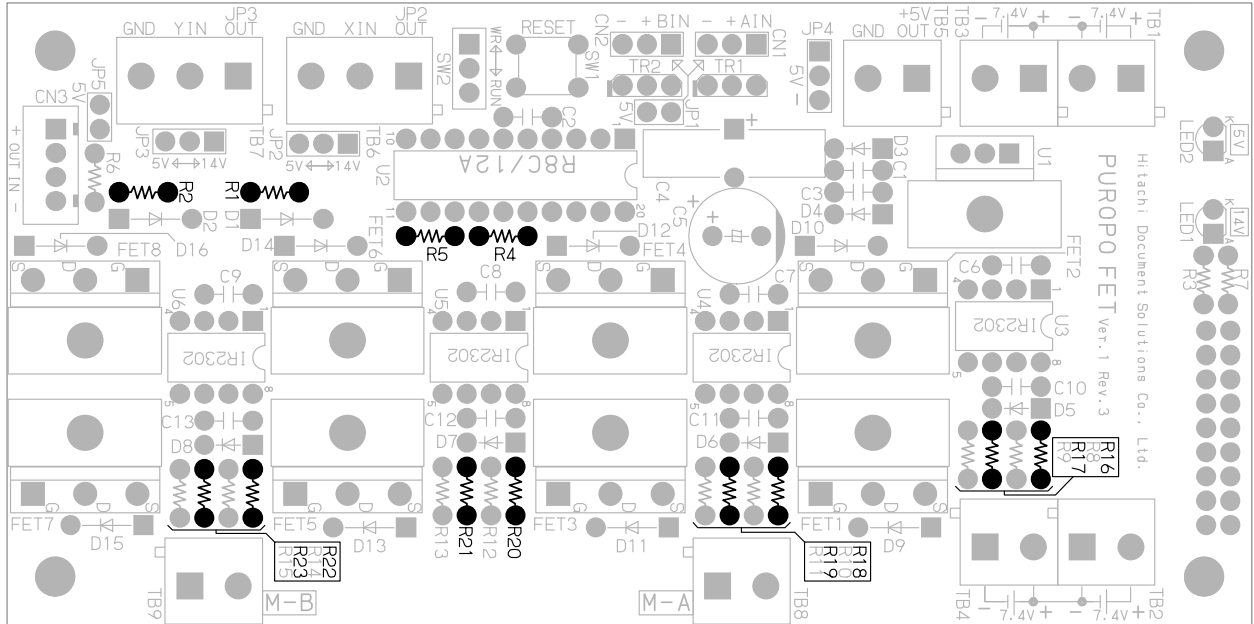


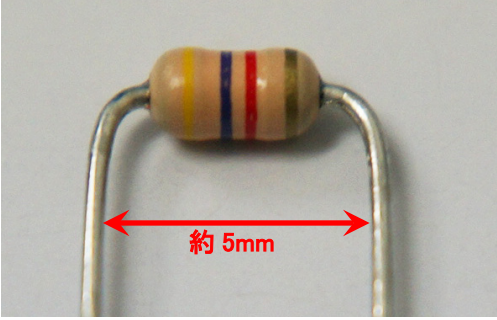
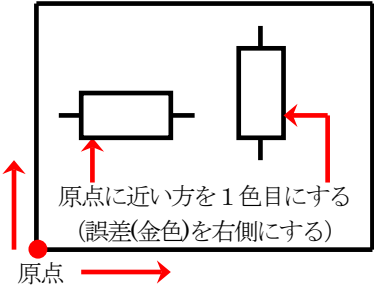
3. プロポ FET 基板の組み立て

3.3 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, 16,17,18,19, 20,21,22,23	抵抗	CFS1/4C 4.7kΩ (黄・紫・赤・金)		コーア(株)	12


抵抗(4.7kΩ)を半田付けします。



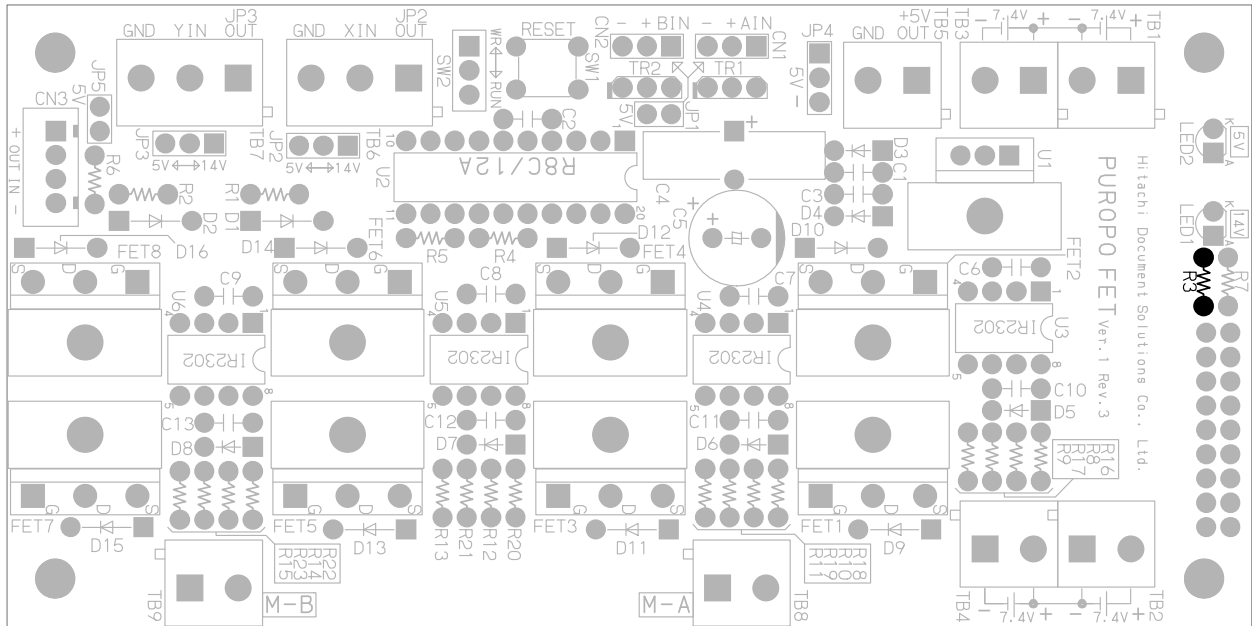
1		抵抗を写真のように曲げ加工します。リードの間隔は、約 5mm にします。 抵抗の根元からリードを曲げると、約 5mm になります。
2		抵抗の実装方法は、基板の左下を原点として、原点に近い方を1色目にして半田付けします。

3. プロポ FET 基板の組み立て

3.4 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。

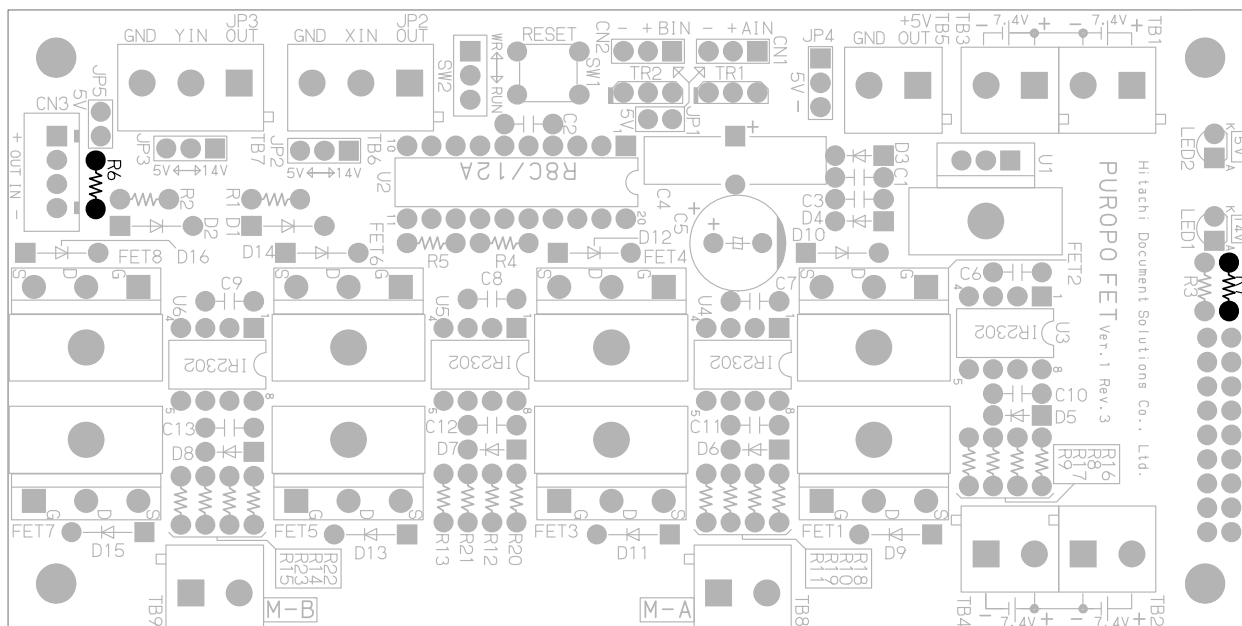


3. プロポ FET 基板の組み立て

3.5 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。



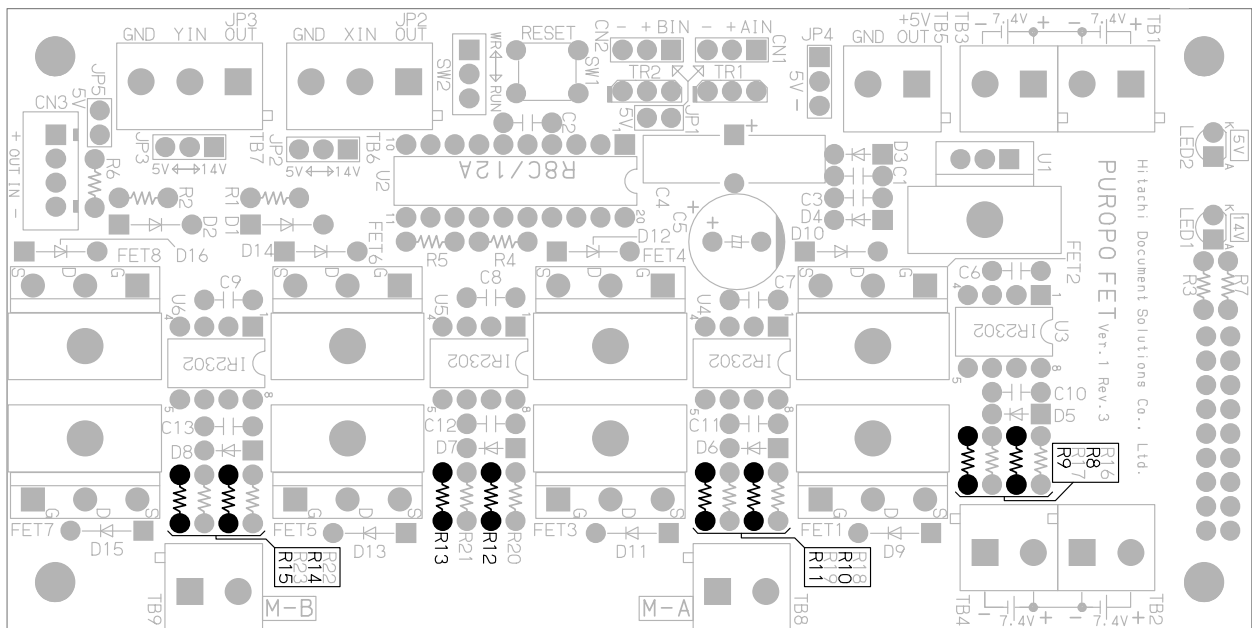


3. プロポ FET 基板の組み立て

3.6 抵抗(100Ω)の取り付け

部品番号	名称	型式	写真	メーカー	数量
R8, 9,10,11, 12,13,14,15	抵抗	CFS1/4C 100Ω (茶・黒・茶・金)		コーア(株)	8

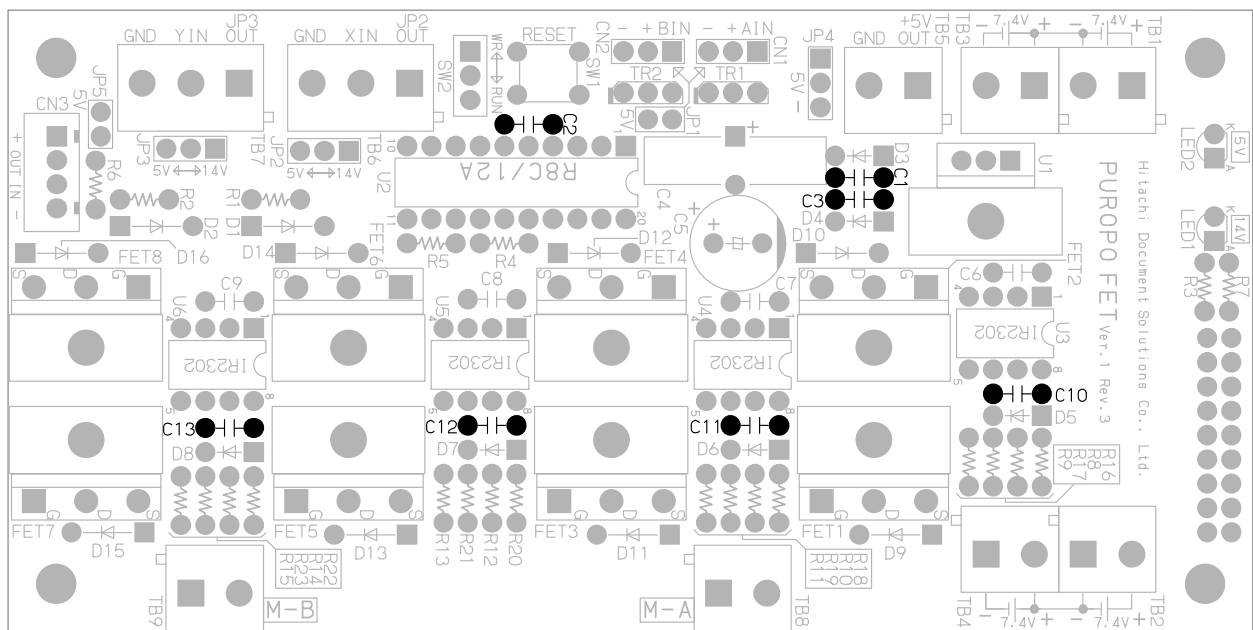
抵抗(100Ω)を半田付けします。



### 3.7 積層セラミックコンデンサ(1 $\mu$ 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, 10, 11, 12, 13	積層セラミックコンデンサ	RDER71H105K2K1H03B 1 $\mu$ F/50V 5.08mm ピッチ		(株)村田製作所	7

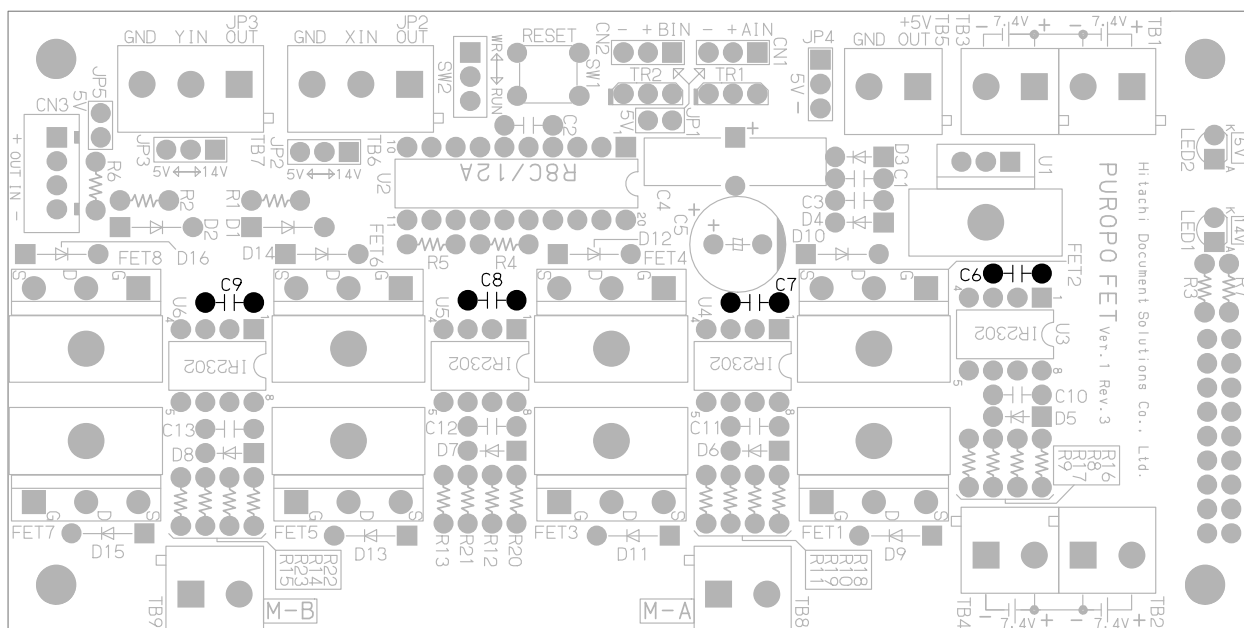
積層セラミックコンデンサ(1  $\mu$  F)を半田付けします。取り付け向きはありません。



### 3.8 積層セラミックコンデンサ(10 $\mu$ F)の取り付け

部品番号	名称	型式	写真	メーカー	数量
C6, 7, 8, 9	積層セラミックコンデンサ	RDEC71H106K3K1H03B 10 $\mu$ F/50V 5.08mm ピッチ		(株)村田製作所	4

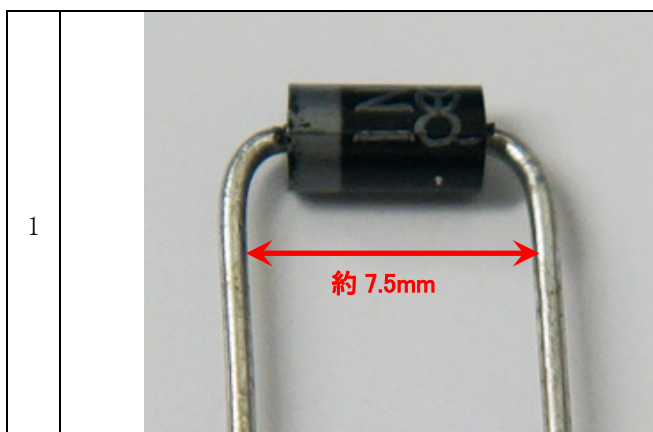
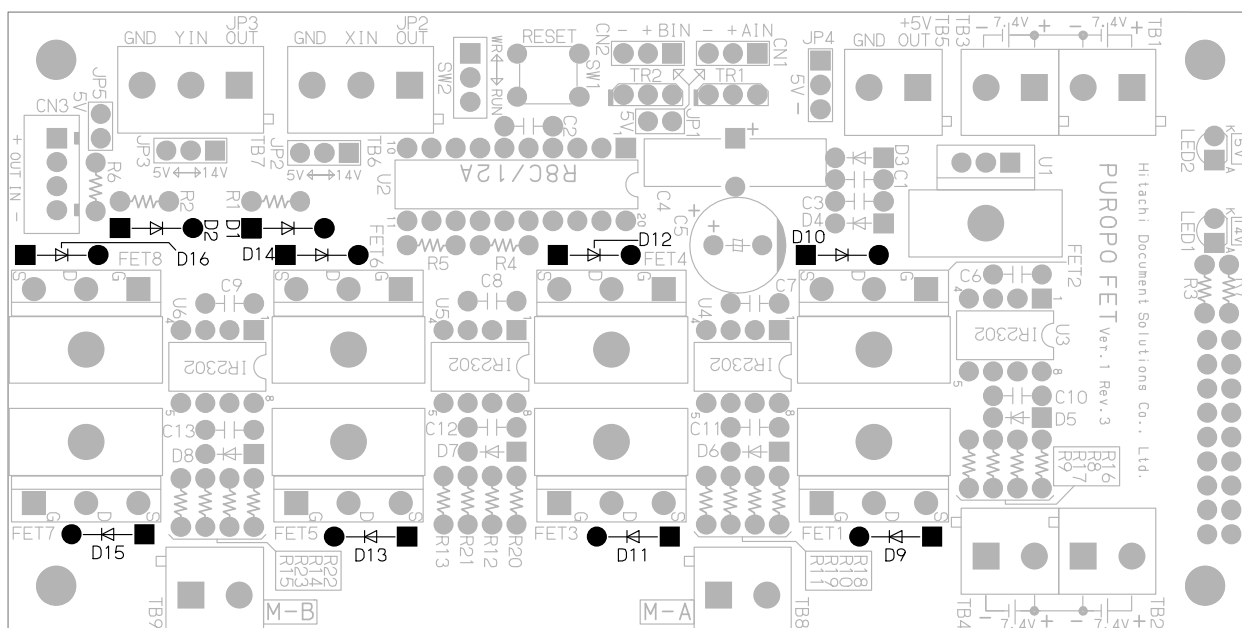
積層セラミックコンデンサ(10 $\mu$ F)を半田付けします。取り付け向きはありません。



### 3.9 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, 9,10, 11,12,13,14, 15,16	汎用整流用ダイオード	1N4007 1000V/1A		PANJIT INTERNATIONAL INC.	10

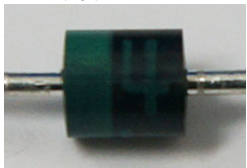
汎用整流用ダイオード(1N4007)を半田付けします。向きに合わせて取り付けてください。



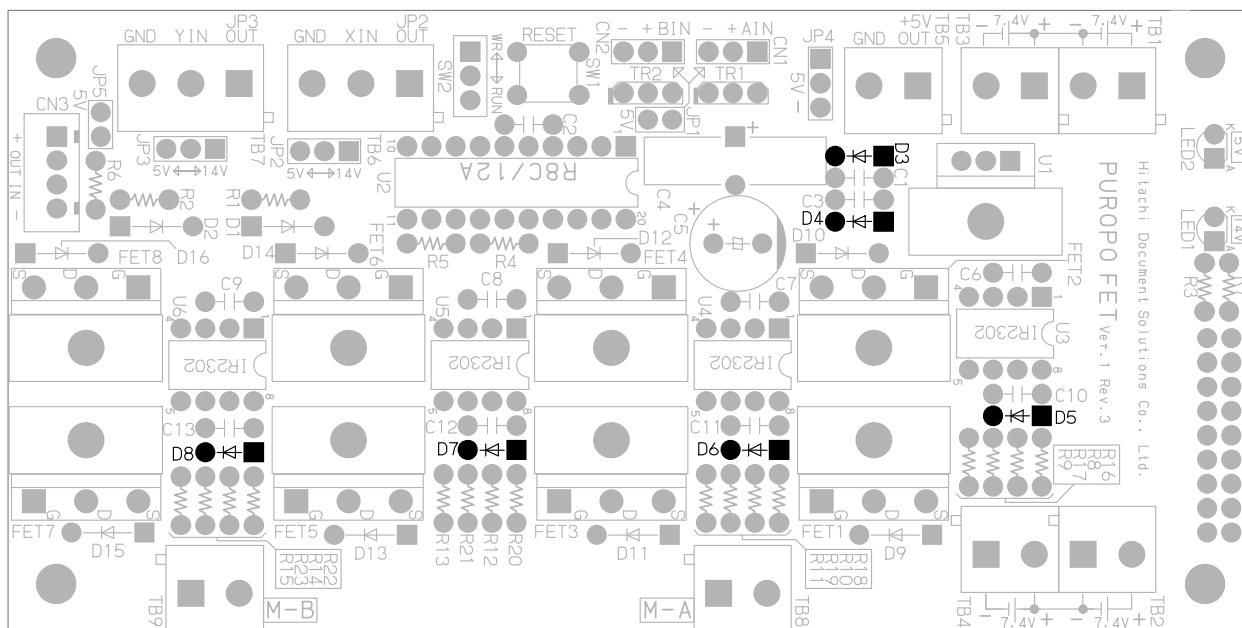
ダイオードを写真のように曲げ加工します。リードの間隔は、約 7.5mm にします。ダイオードの根元からリードを曲げると、約 7.5mm になります。

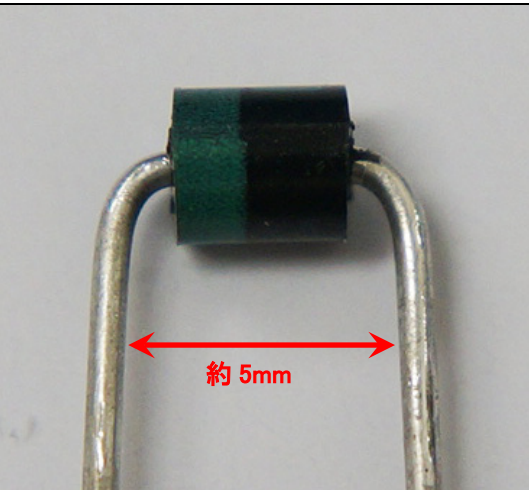
3. プロポ FET 基板の組み立て

3.10 ショットキーバリアダイオード(11EQS04)の取り付け

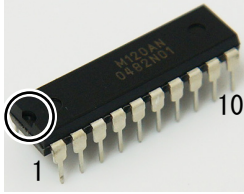
部品番号	名称	型式	写真	メーカー	数量
D3, 4, 5, 6, 7, 8	ショットキーバリアダイオード	11EQS04 40V/1A V <sub>FM</sub> (せん頭順電圧):0.55V	<p>K(印)      A</p> 	日本インター(株)	6

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。

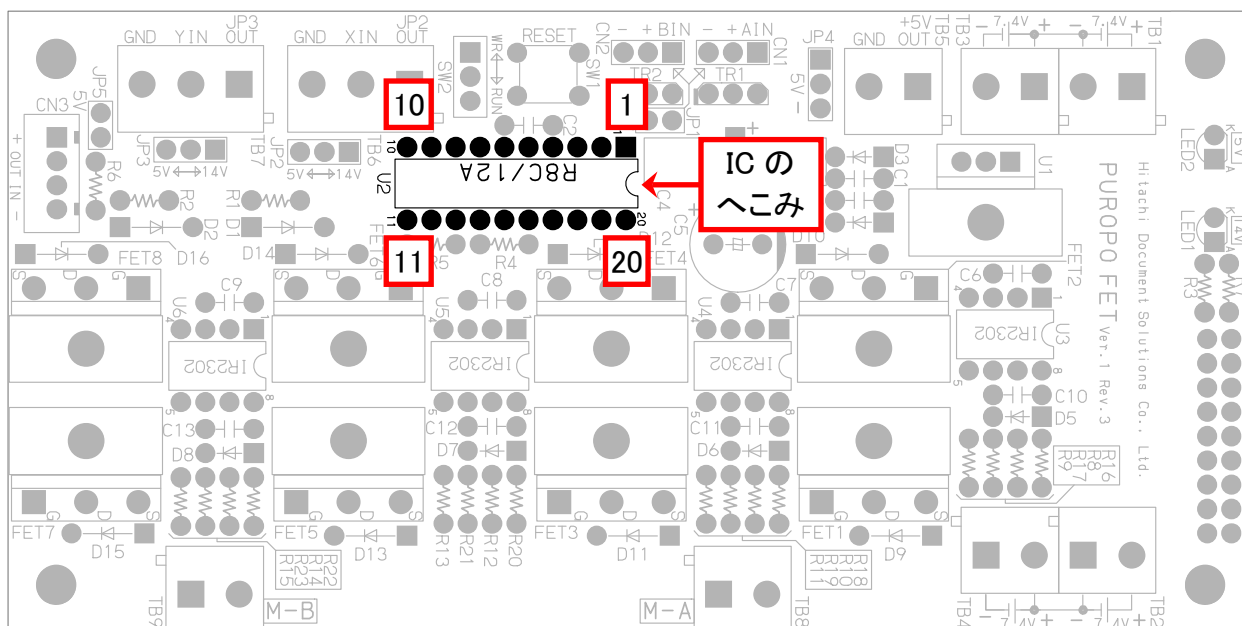


1		<p>ダイオードを写真のように曲げ加工します。リードの間隔は、約 5mm にします。ダイオードの根元からリードを曲げると、約 5mm になります。</p>
---	---	---

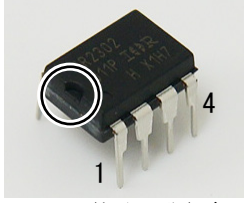
### 3.11 R8C マイコン(M120AN)の取り付け

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

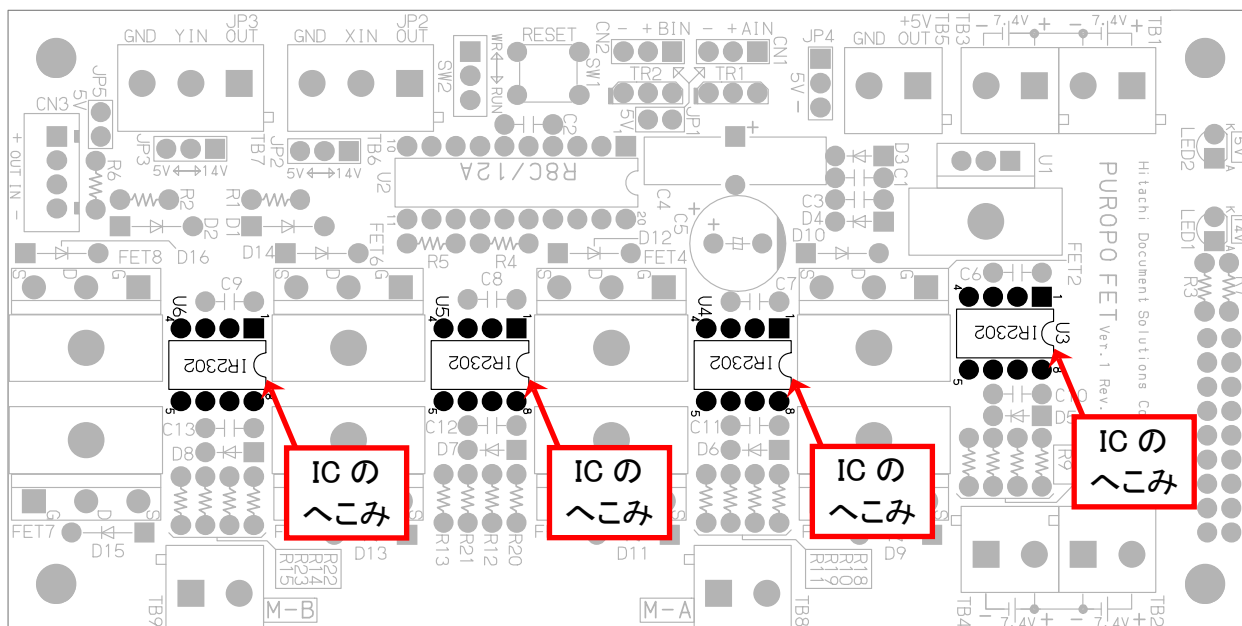
R8C マイコン(M120AN)を半田付けします。向きに合わせて取り付けてください。



### 3.12 ハーフブリッジドライバ(IR2302PBF)の取り付け

部品番号	名称	型式	写真	メーカー	数量
U3, 4, 5, 6	ハーフブリッジ ドライバ	IR2302PBF	 <p>0のへこみがある側の手前が1ピン</p>	International Rectifier	4

ハーフブリッジドライバ(IR2302PBF)を半田付けします。向きに合わせて取り付けてください。

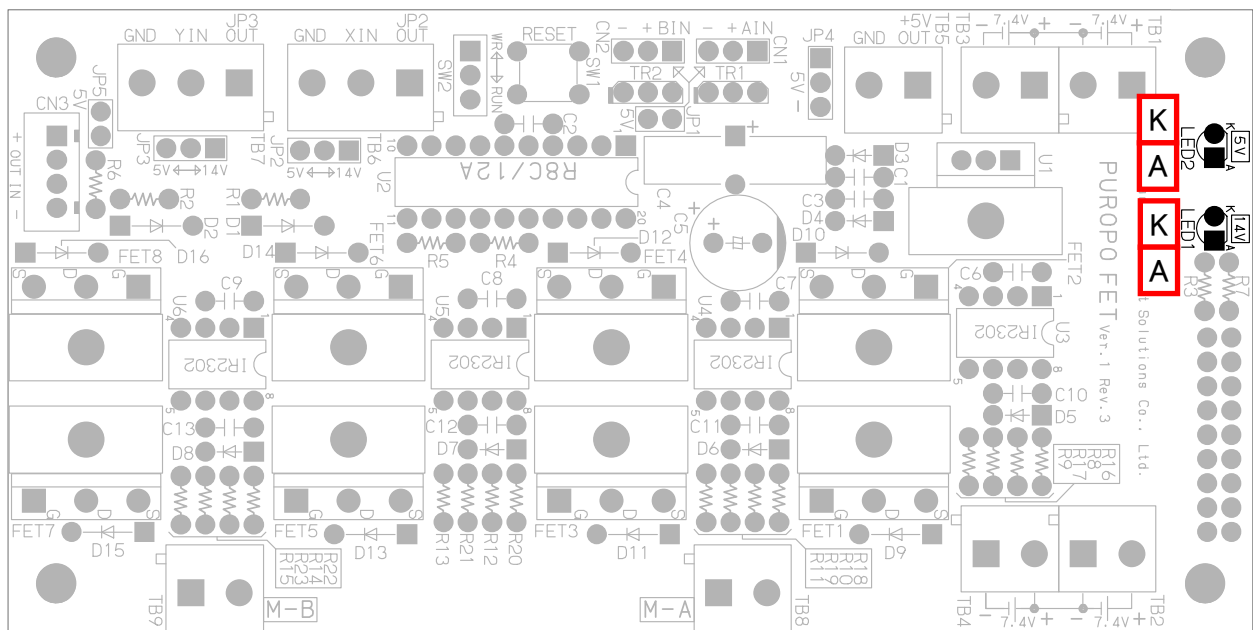


3. プロポ FET 基板の組み立て

3.13 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

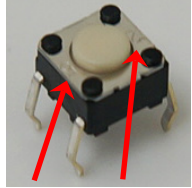
LED(EBR3338S)を半田付けします。向きに合わせて取り付けてください。





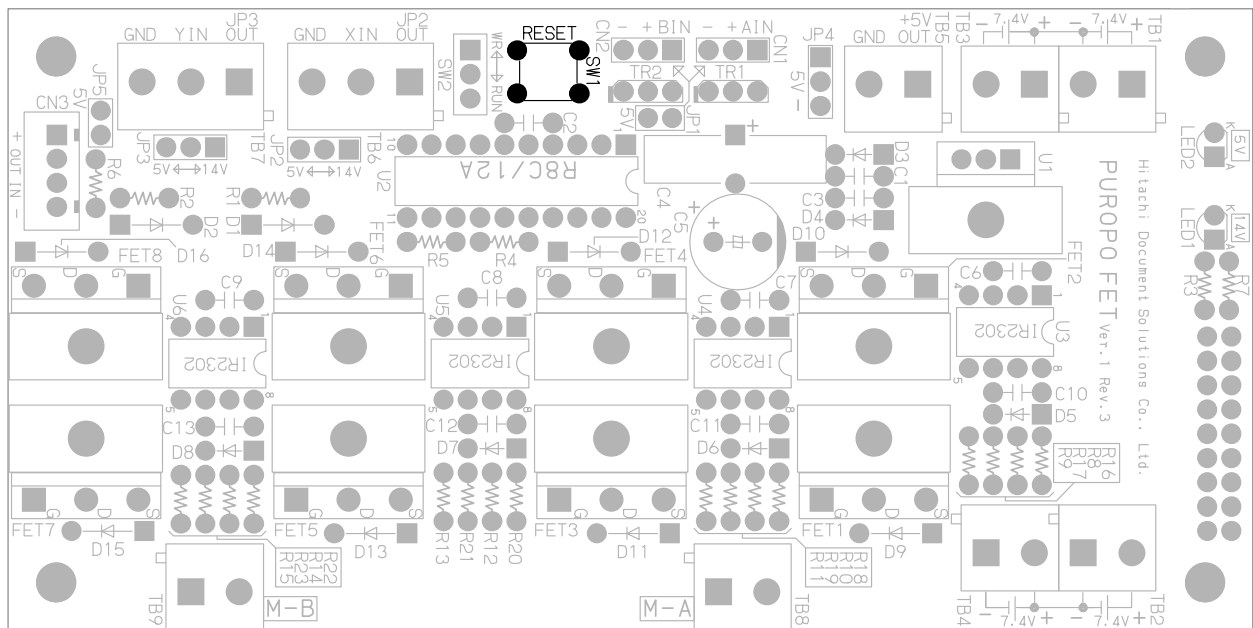
3. プロポ FET 基板の組み立て

3.14 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

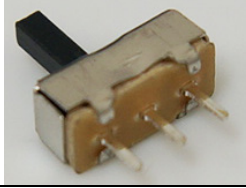
タクトスイッチ(B3F-1000)を半田付けします。取り付け向きはありません。

※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。

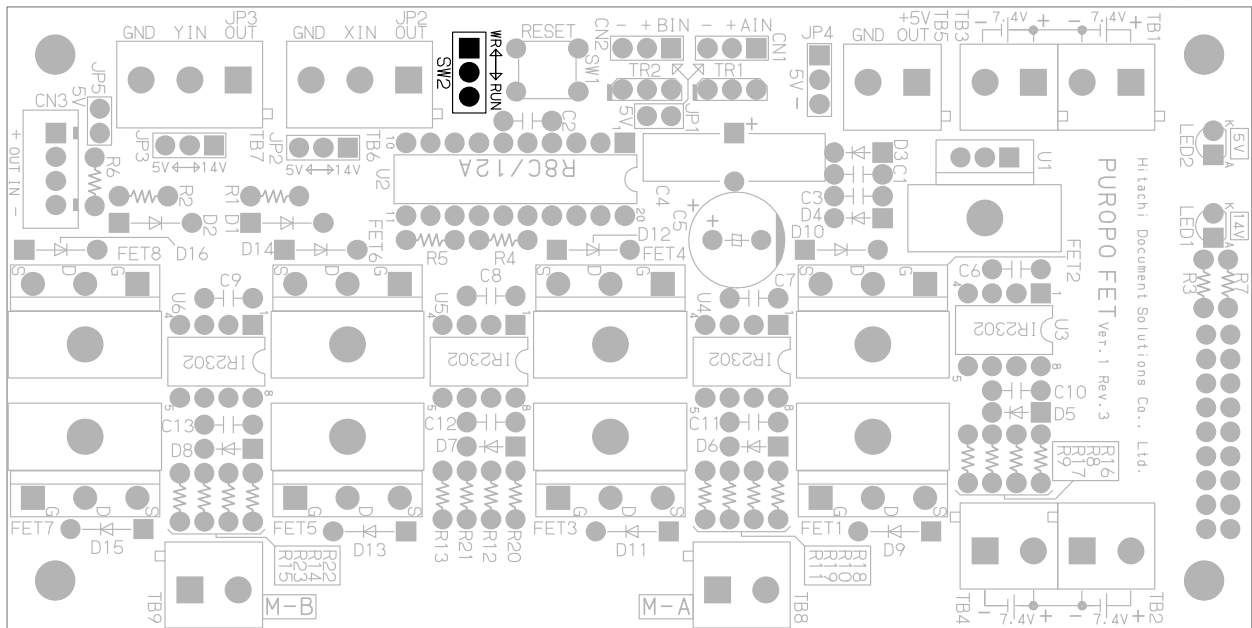


3. プロポ FET 基板の組み立て

3.15 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

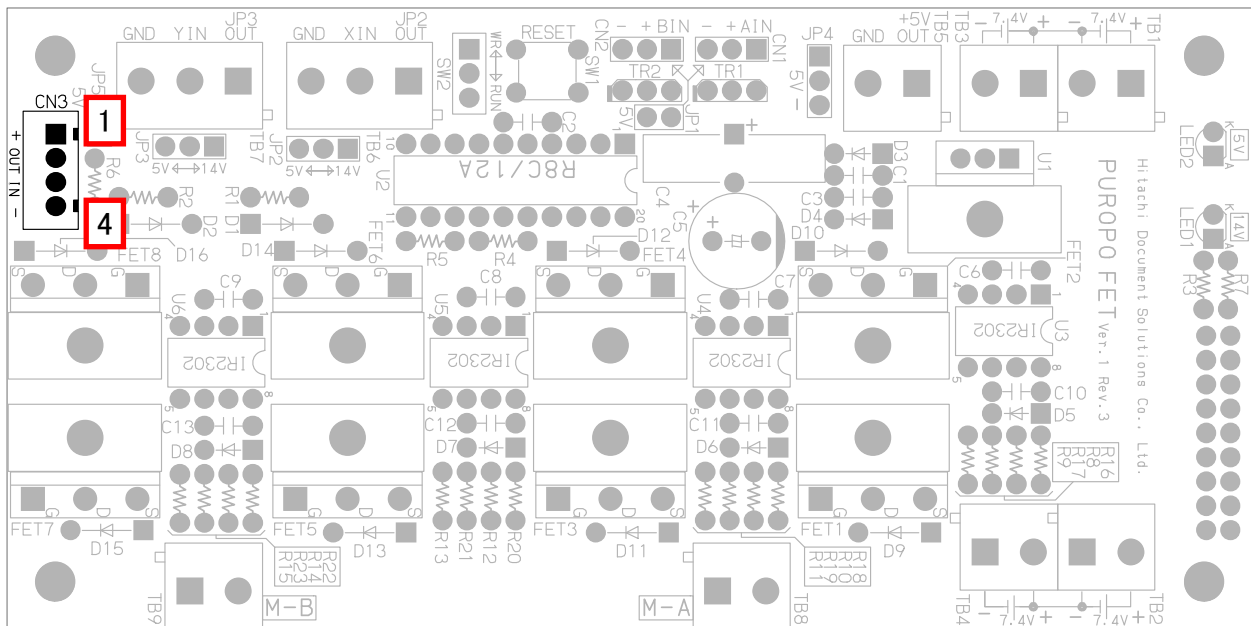
スライドスイッチ(SS-12D00-G5)を半田付けします。向きはありません。



### 3.16 XH コネクタ 4 ピンオス(B4B-XH-A)の取り付け

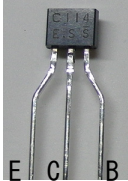
部品番号	名称	型式	写真	メーカー	数量
CN3	XHコネクタ4ピンオス	B4B-XH-A		日本圧着端子製造(株)	1

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。

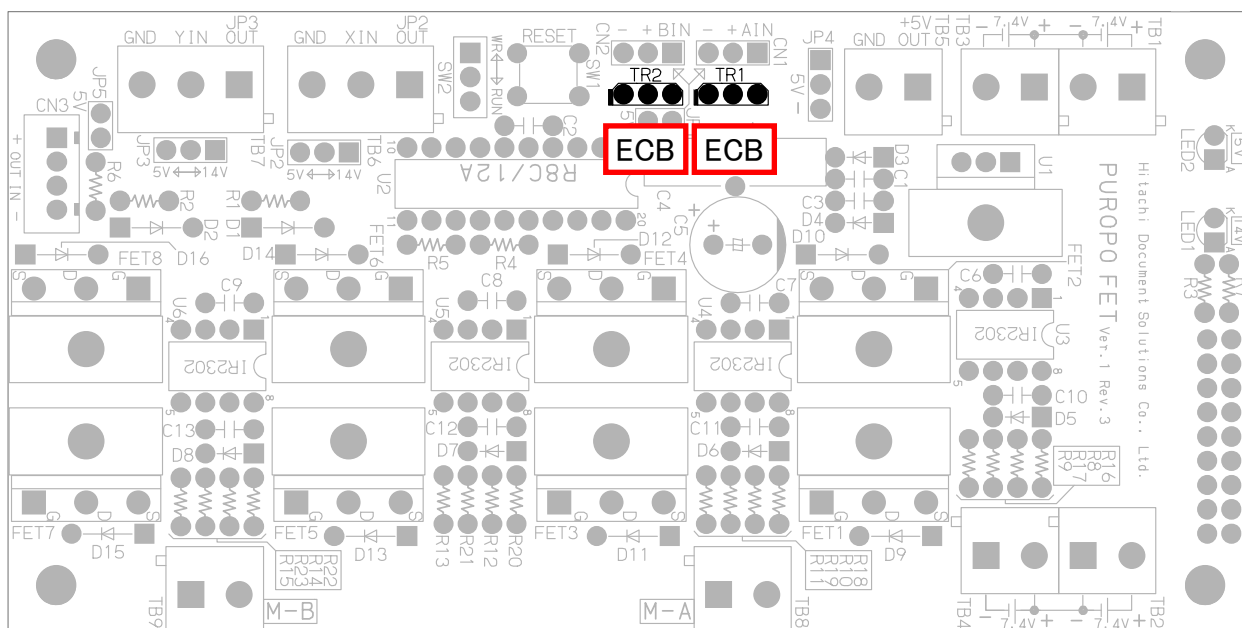


3. プロポ FET 基板の組み立て

3.17 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	抵抗内蔵トランジスタ	RN1201		(株)東芝セミコンダクター社	2

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。



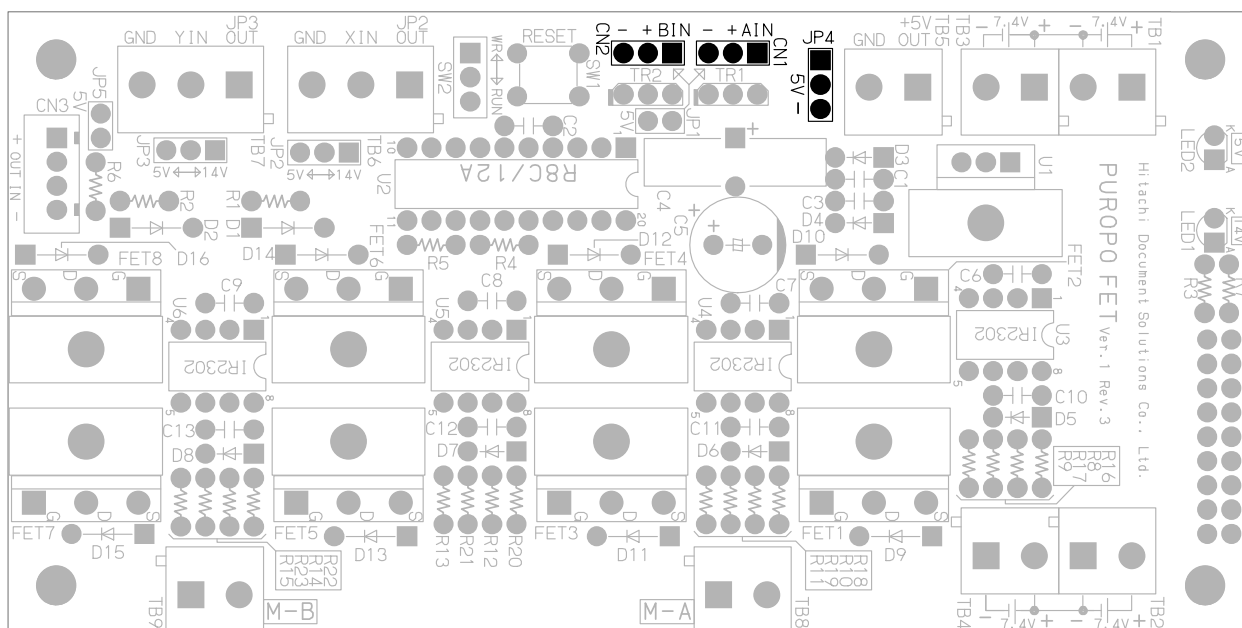
3. プロポ FET 基板の組み立て

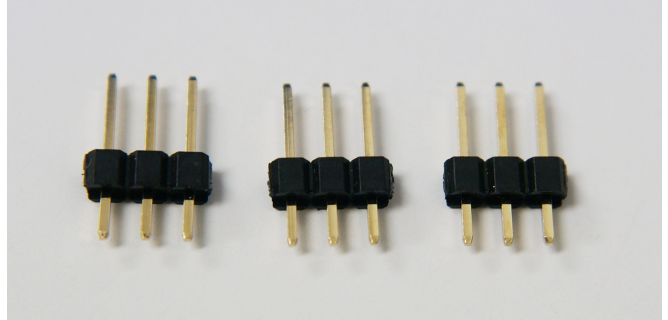
3.18 ピンヘッダ(1 列 × 40P)の取り付け

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

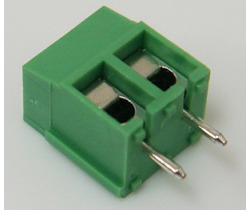
ピンヘッダ(1 列 × 40P)を半田付けします。

1 列 × 40P のピンヘッダを 3 ピンにカットして、CN1、CN2、JP4 に実装してください。取り付け向きはありません。

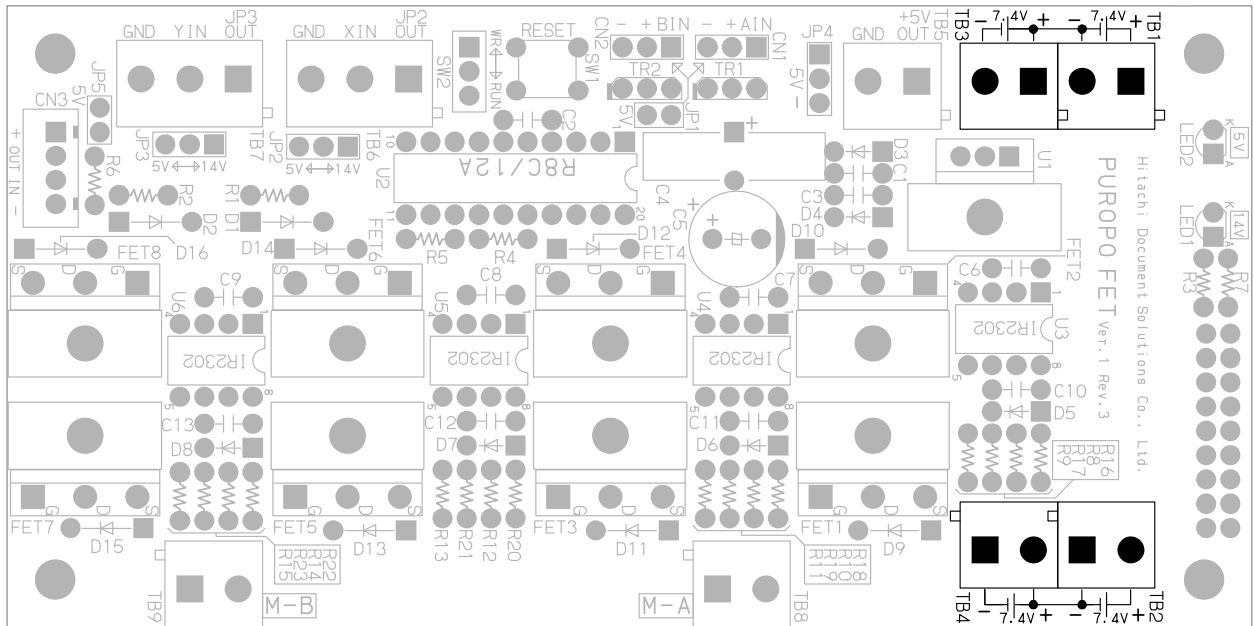


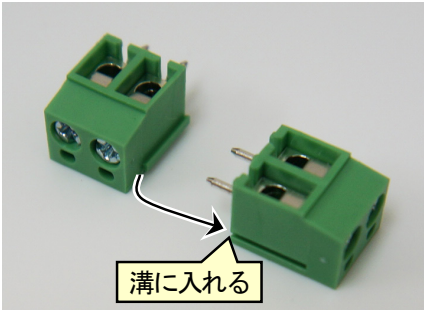
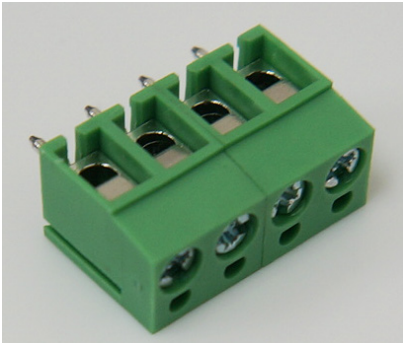
1		<p>1 列 × 40P のピンヘッダを 3 ピンにカットして、CN1、CN2、JP4 に実装してください。短いピンを、半田付けします。</p>
---	---	--

### 3.19 ターミナルブロック(2ピン・緑・縦)の取り付け

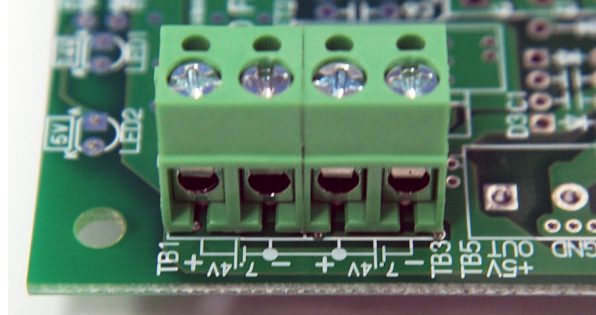
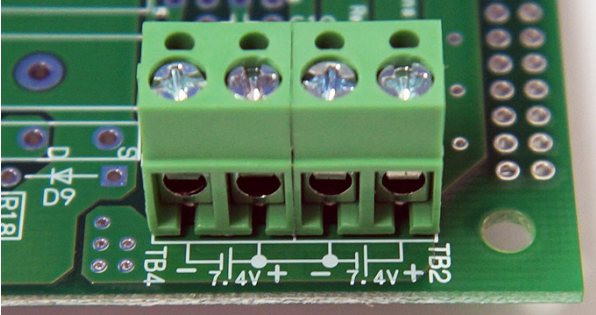
部品番号	名称	型式	写真	メーカー	数量
TB1, 2, 3, 4	ターミナルブロック 2ピン(緑)(縦)	TB112-2-2-E-1		Alphaplus Connectors & Cables Corp.	4

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。

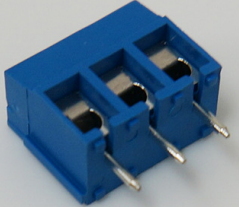


1		TB1 と TB3 は、左写真のようにターミナルブロックの溝に差し込み、2 個つながっている状態で実装します。 TB2 と TB4 も同様です。
2		溝に入れたところです。

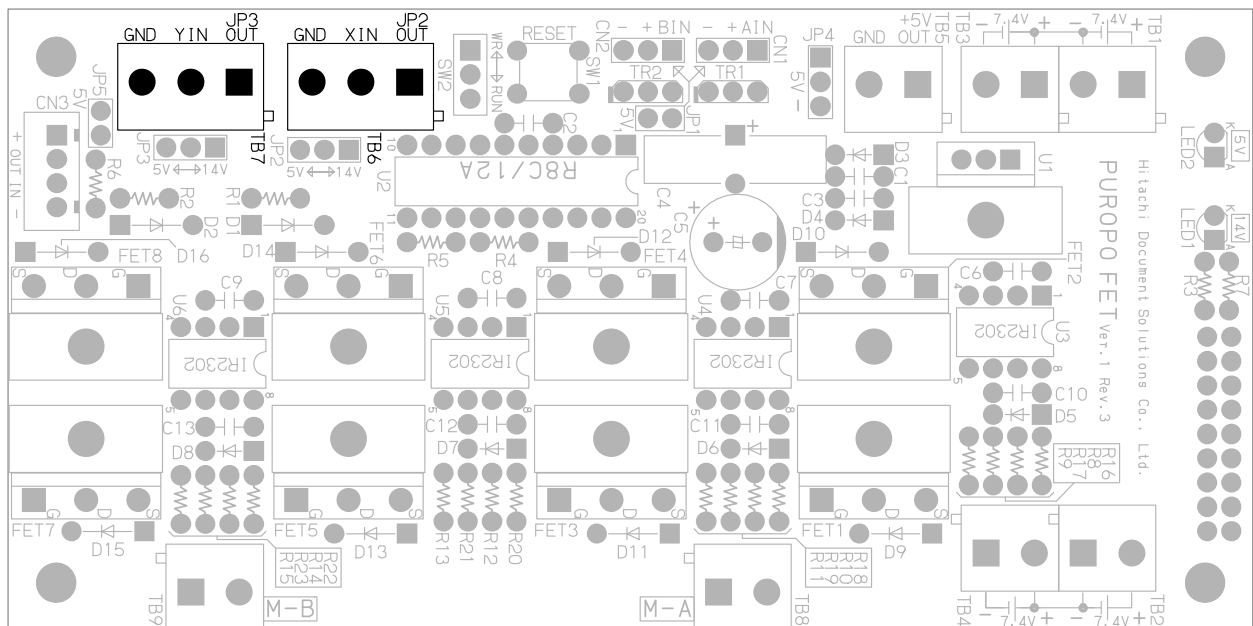
3. プロポ FET 基板の組み立て

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。

3.20 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。

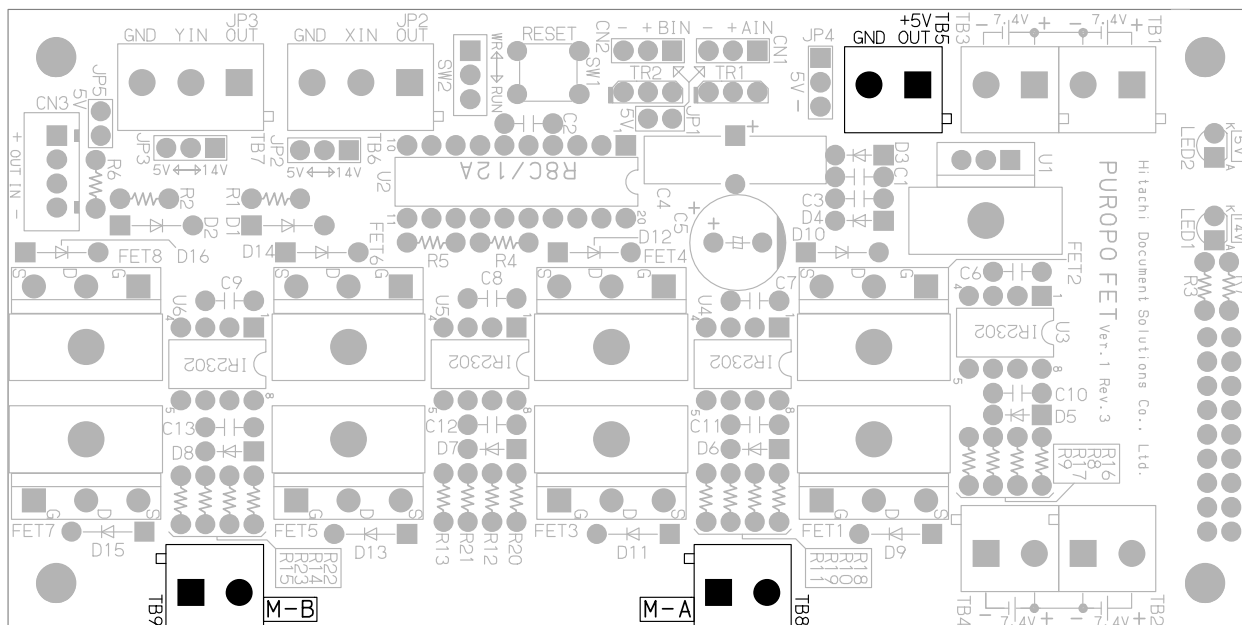


3. プロポ FET 基板の組み立て

3.21 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
TB5, 8, 9	ターミナルブロック 2ピン(青)(縦)	TB112-2-2-U-1		Alphaplus Connectors & Cables Corp.	3

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。

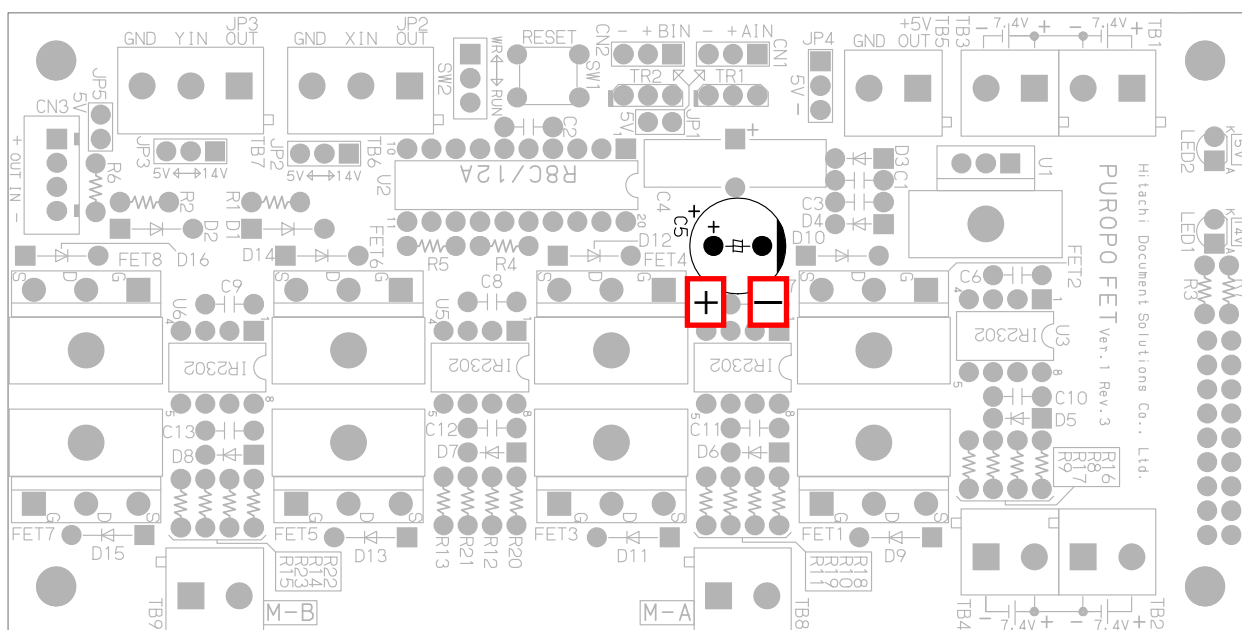




### 3.22 電解コンデンサ(1000 $\mu$ F)の取り付け

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 $\mu$ F/16V		日本ケミコン (株)	1

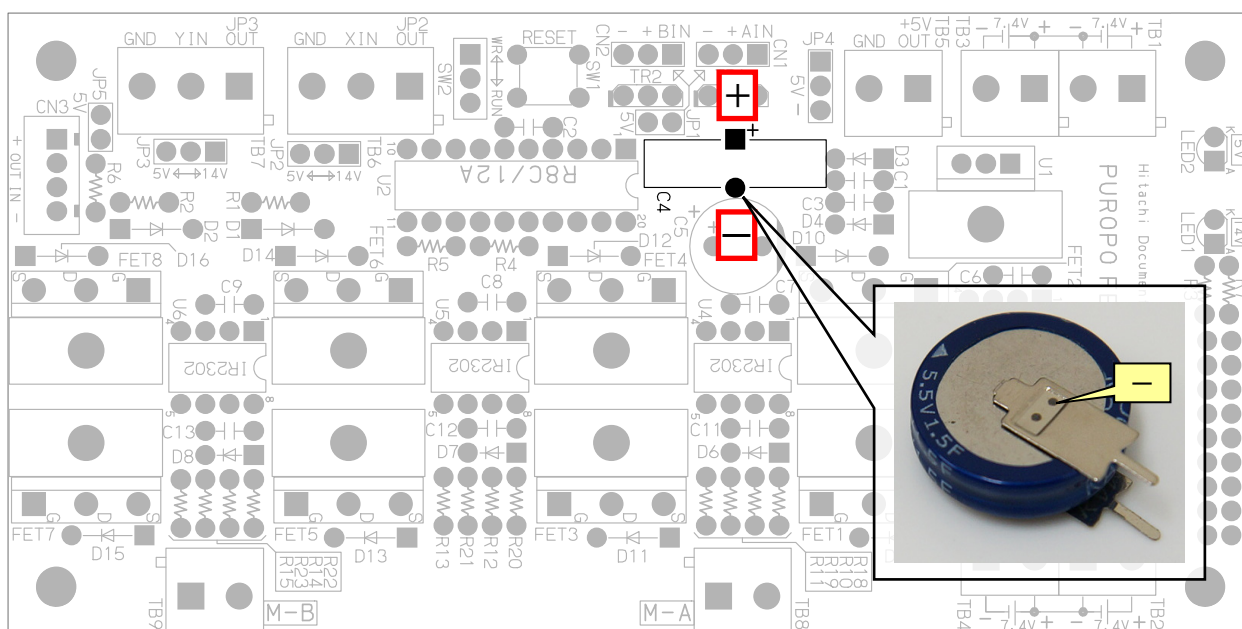
電解コンデンサ(1000  $\mu$ F)を半田付けします。向きに合わせて取り付けてください。



### 3.23 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。

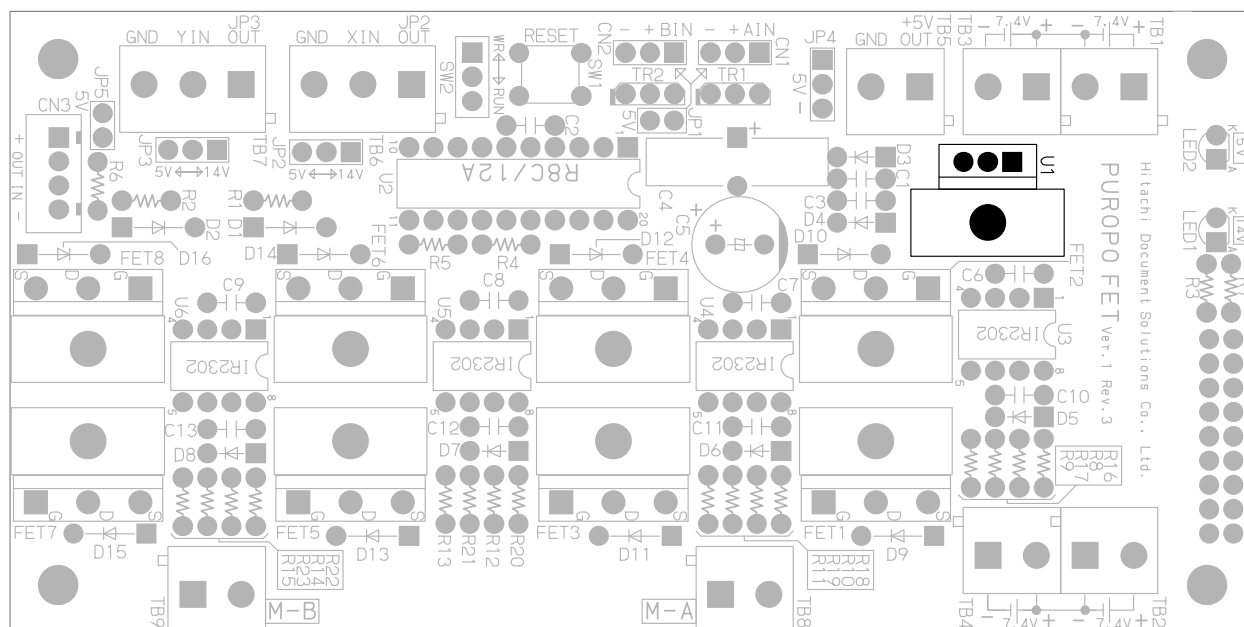


3. プロポ FET 基板の組み立て

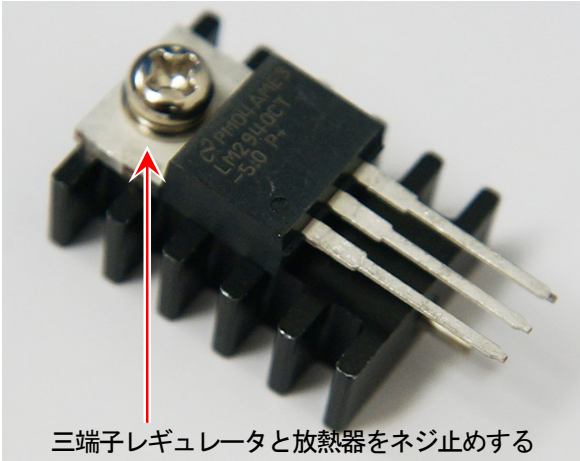
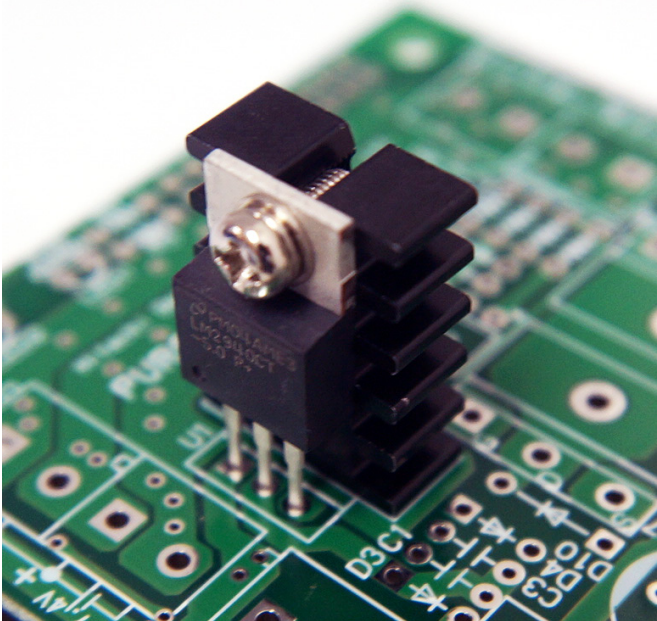
3.24 三端子レギュレータ(LM2940CT-5.0)の取り付け

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てから、半田付けしてください。



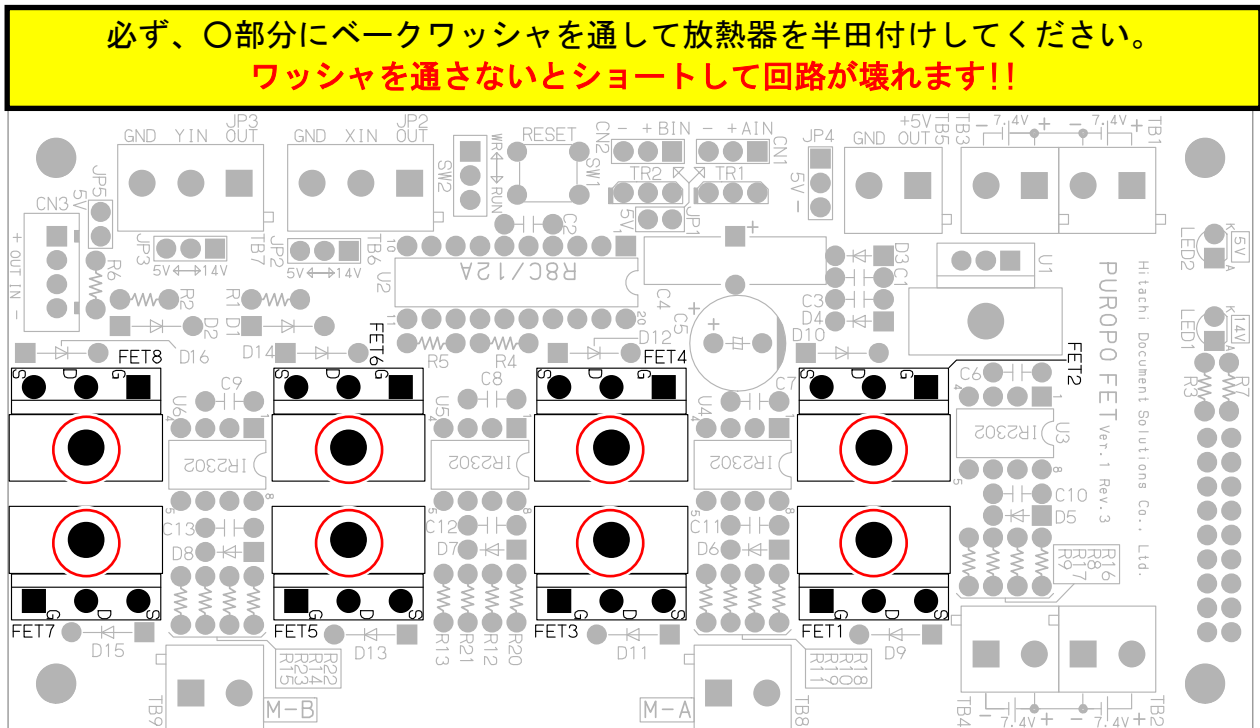
3. プロポ FET 基板の組み立て

<p>1</p>	 <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>
<p>2</p>		<p>実装したところです。</p>

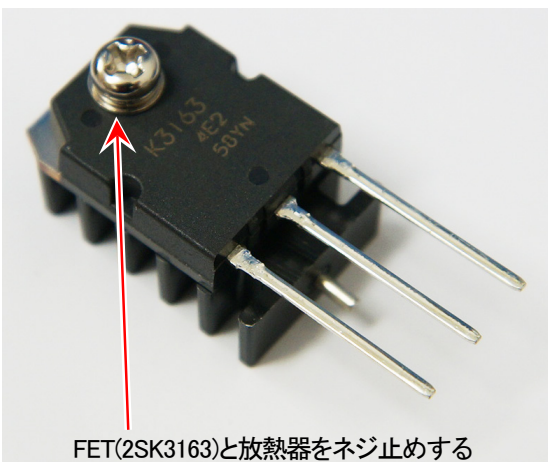
### 3.25 Nch パワーFET(2SK3163)の取り付け

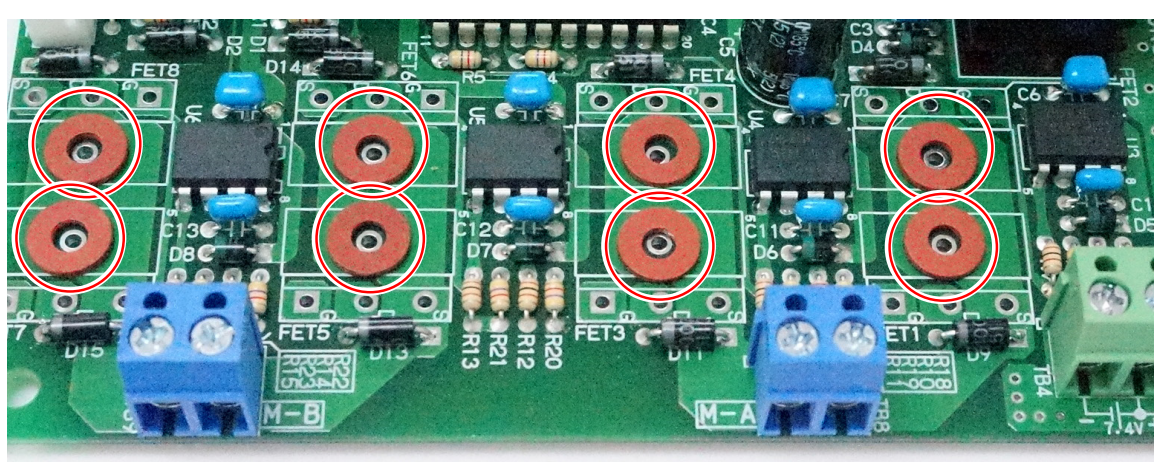
部品番号	名称	型式	写真	メーカー	数量
FET1,2,3, 4, 5, 6, 7, 8	Nch パワーFET	2SK3163 60V75A		ルネサス エレクトロニクス(株)	8
FET1~8 の放熱器の下に入れる	ベークワッシャ	XX-0306-05 内径φ3、外径φ6 厚さ0.5mm		(株)廣杉計器	8
FET1~8 用	放熱器	PC2116-7-PB		(株)放熱器のオーエス	8
放熱器固定用	座金ネジ	φ3×8mm		各社	8

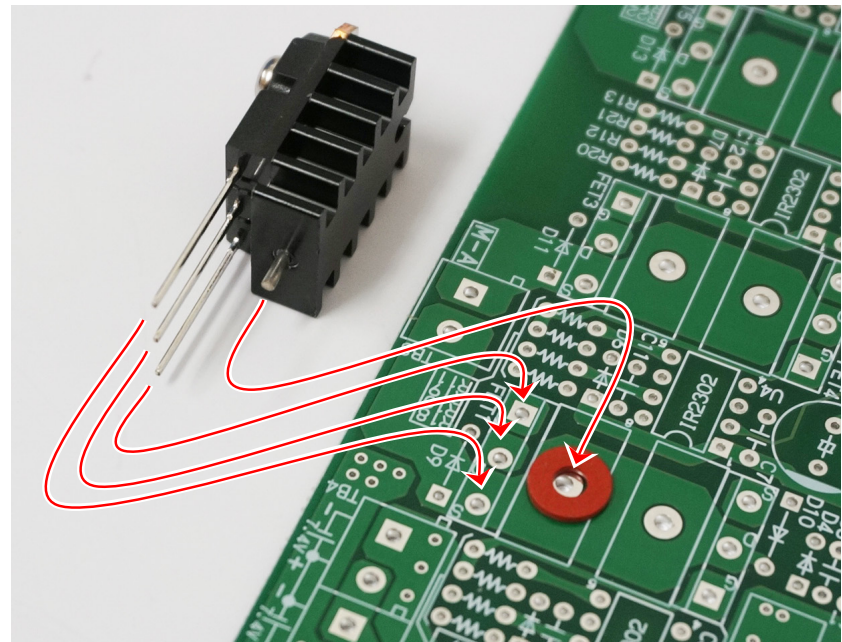
Nch パワーFET(2SK3163)を半田付けします。パワーFET(2SK3163)と放熱器をネジ止めしてから、半田付けしてください。半田付けするとき、放熱器の下には必ずベークワッシャを入れてください。




3. プロポ FET 基板の組み立て

1	 <p>FET(2SK3163)と放熱器をネジ止めする</p>	<p>Nch パワーFET(2SK3163)、放熱器、座金ネジ、ベークワッシャを用意します。</p> <p>左写真のように、Nch パワーFET(2SK3163)を放熱器に座金ネジで固定します。</p>
---	--	---

2	 <p>写真のように、ベークワッシャを FET1~8 の放熱器を固定するランド上に載せます。</p>
---	---

3		<p>基板と放熱器の間にベークワッシャがある状態で実装します。</p> <p>写真は、FET1 部分に実装しているところです。</p> <p>FET1~FET8 のすべて、ベークワッシャを間に入れて実装してください。</p> <p><b>ベークワッシャを通さないと、放熱器と部品面のパターンが触れてショートし、基板や部品、線材などが壊れます!!</b></p>
---	--	--

3. プロポ FET 基板の組み立て

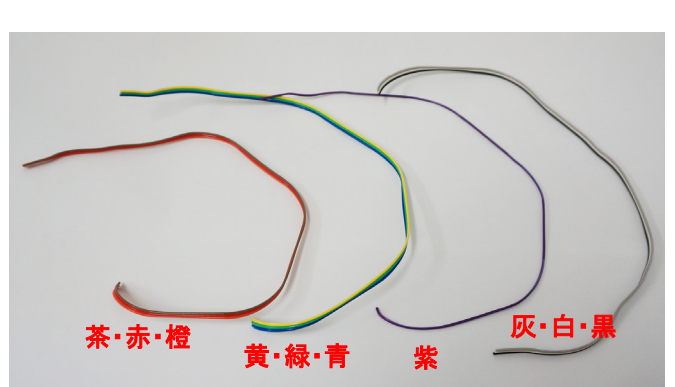
4	 <p>ランドがショートしている例(矢印部分)</p>	<p>放熱器のランドは、周りのベタとショートさせないように半田付けしてください。</p> <p><b>放熱器のランドは、FETのドレイン(D)とつながっており、ランドのショートは FET のショートと同じです。ショートさせると、FET や関係部品が壊れます。</b></p>
---	--	---

### 3.26 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

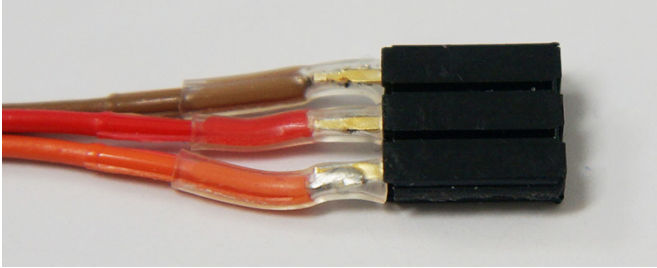
プロポ受信器と、プロポ FET 基板をつなぐケーブルを 3 本作ります。


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 FET 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 FET 基板をつなぐケーブルです。</p>
---	---	--



3. プロポ FET 基板の組み立て

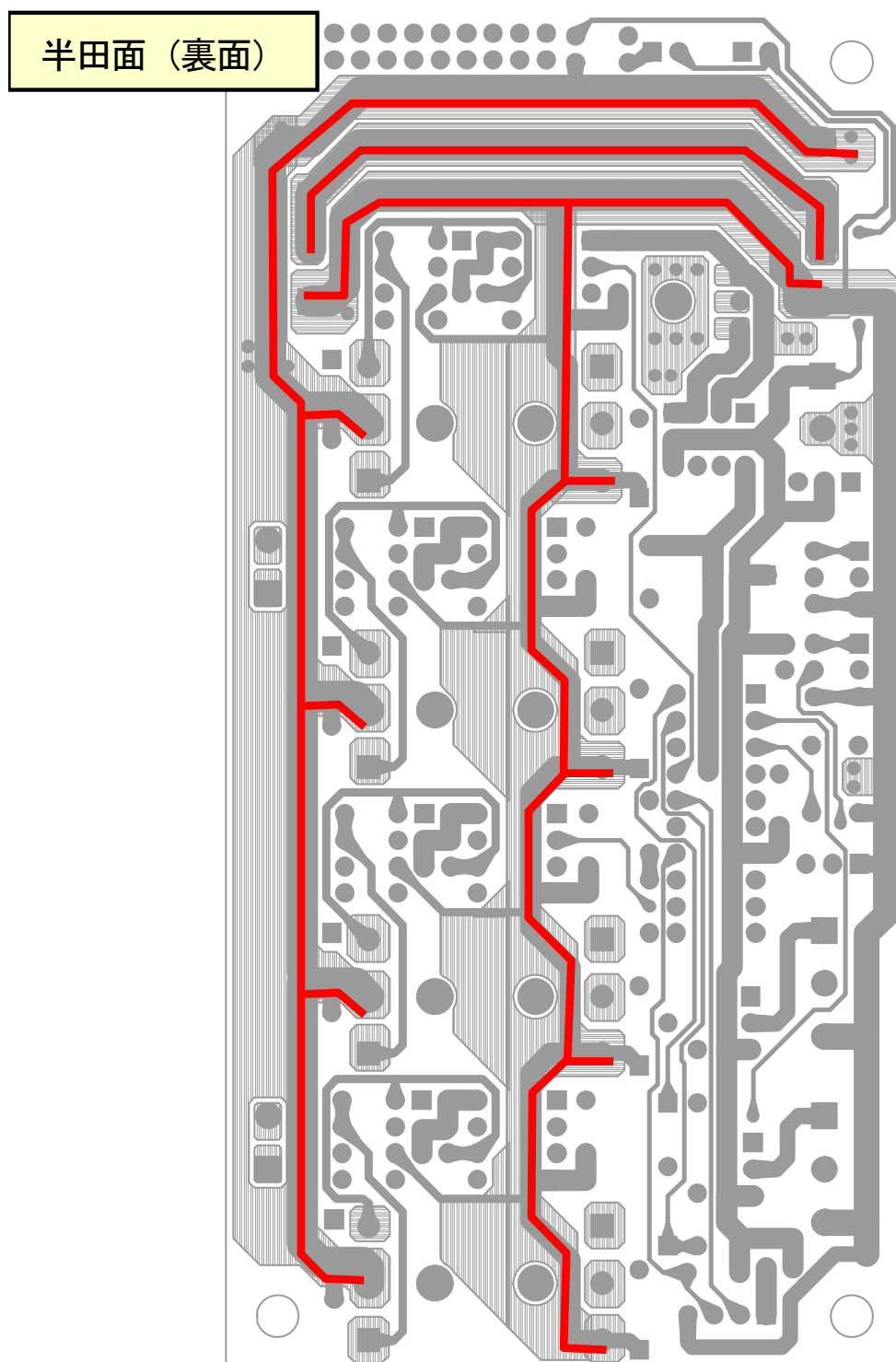
3		<p>分割ロングピンソケットと、フラットケーブルを半田付けします。半田付けした部分が、他の部分に触ったときにショートしないように熱収縮チューブをかぶせておきます。</p>
---	---	---

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。</p> <p>これを、</p> <ul style="list-style-type: none"><li>①茶・赤・橙の3色のケーブル</li><li>②黄・緑・青の3色のケーブル</li><li>④灰・白・黒の3色のケーブル</li></ul> <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>
---	---	--

### 3.27 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。

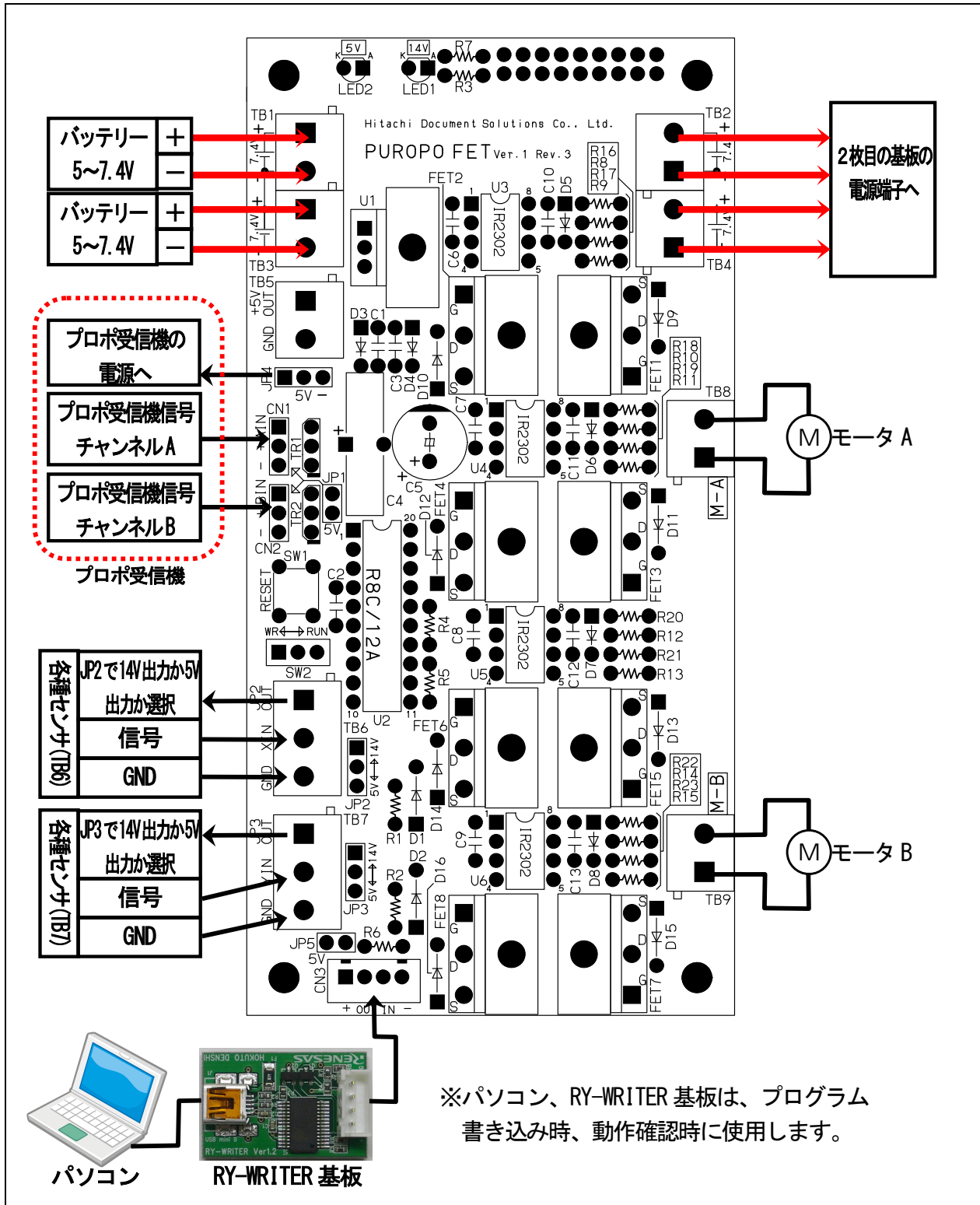


4. 接続

4. 接続

4.1 接続図

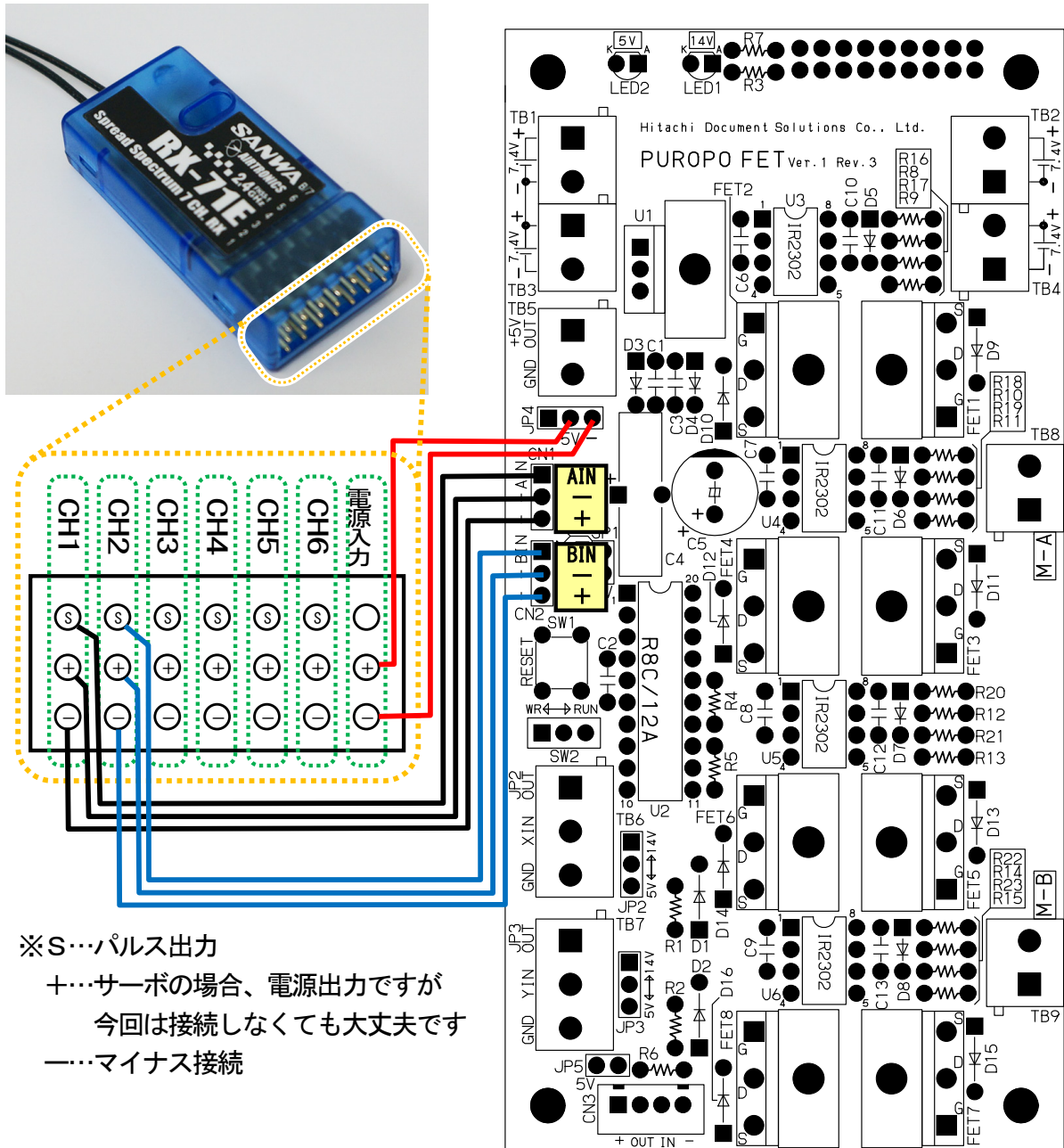
プロボ FET 基板の接続構成を下記に示します。



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。

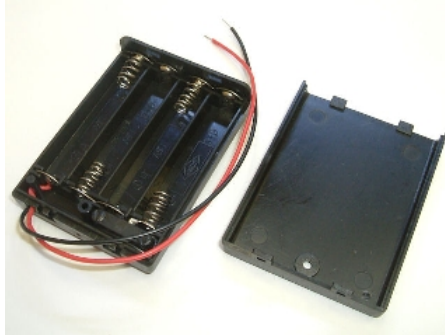


## 4.2 マイコンボードの電源を別にする

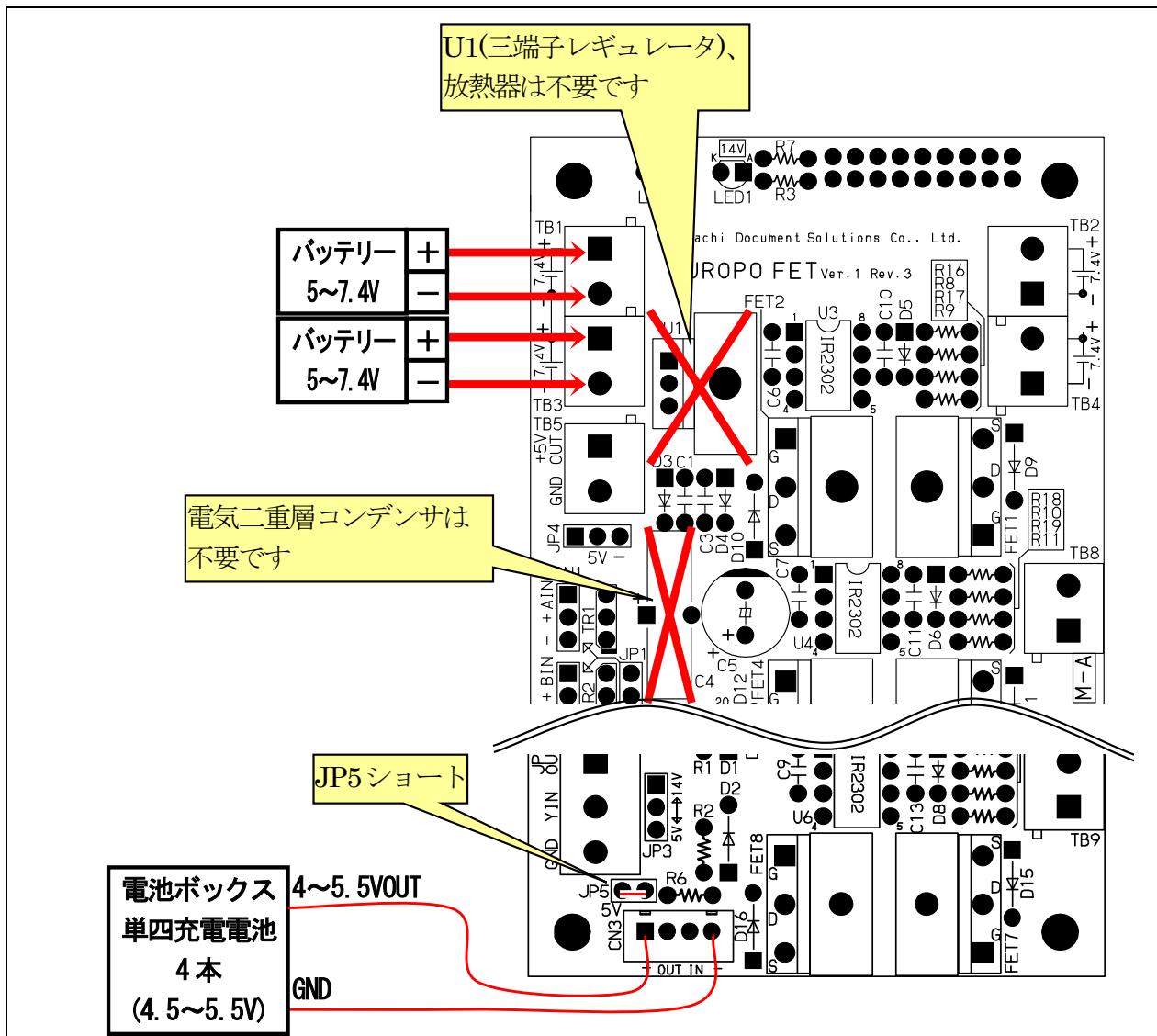
モータによる電圧降下が大いいと、マイコンへの供給電圧も低下してリセットしてしまい、ロボット制御できなくなります。この場合、マイコンの電源を別にしてください。

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/>

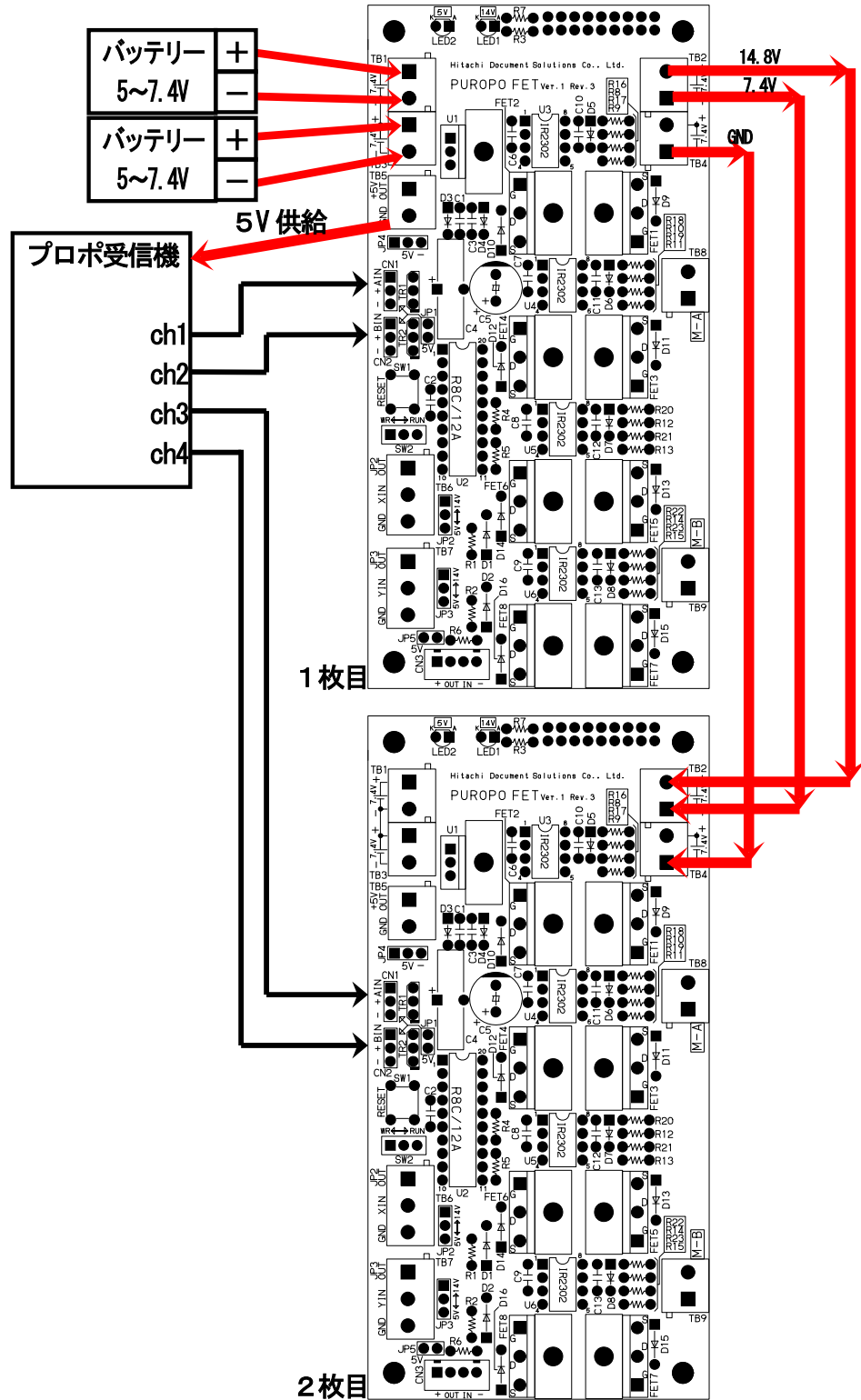


電池は、**充電式**単四電池を使用してください。アルカリ電池の場合は  $1.5V \times 4 = 6.0V$  となり、**マイコンの動作保証範囲外**となります。



### 4.3 複数のプロポ FET 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 FET 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

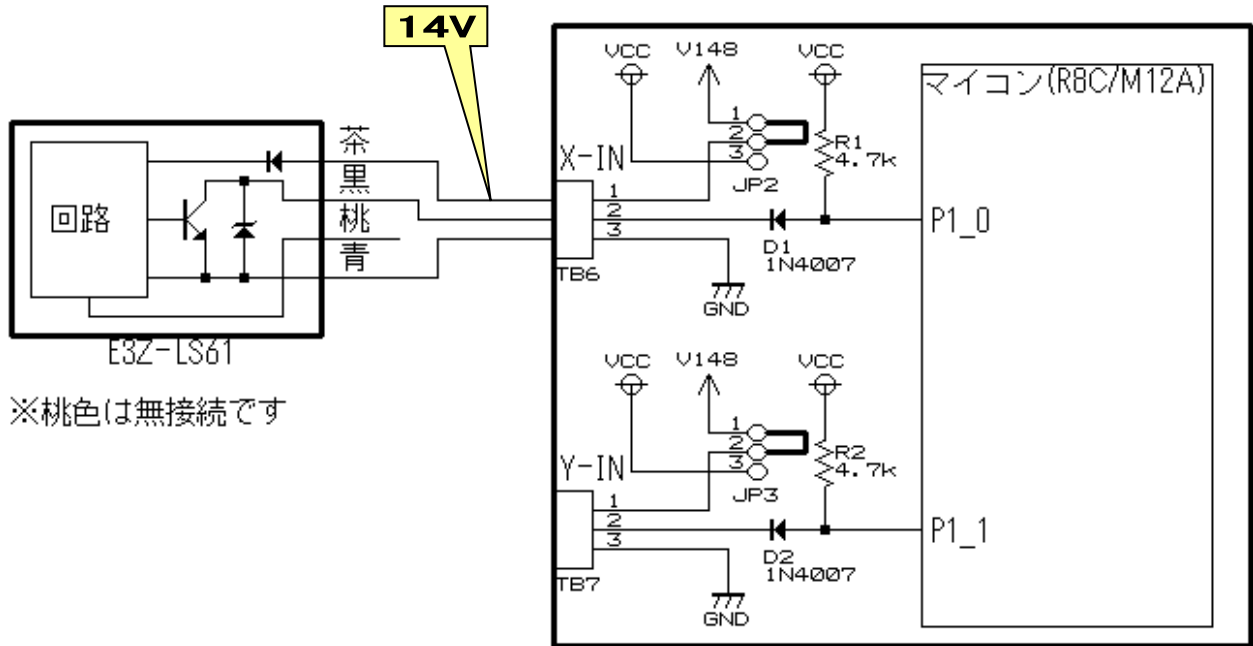
4.4 センサの接続方法

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。

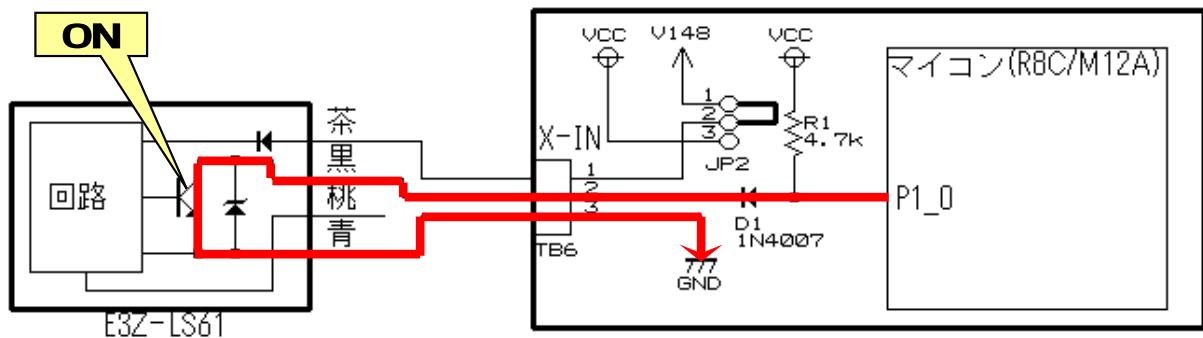
例として、オムロンの「細ビーム反射形 コード引き出しタイプ 検出距離 90±30mm NPN 出力 (E3Z-L61)」を接続するときの回路を、下記に示します。



※桃色は無接続です

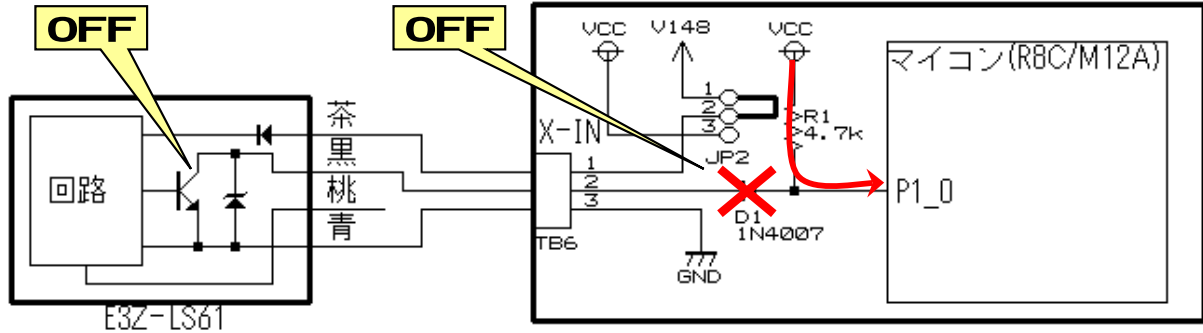
※動作原理

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

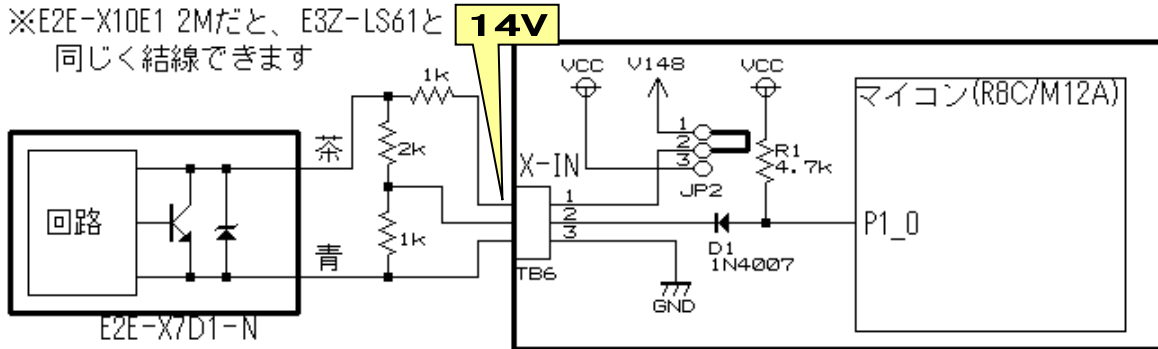
センサが OFF になると、トランジスタが OFF して、D1 は OFF になるため、R1 のプルアップ抵抗を通して、P1\_0 端子は矢印のように 5V に接続され、「1」が入力されます(下図)。※センサによってはトランジスタが OFF しても 8 ~12V の電圧が出力される場合があるので、D1 は必ず付けてください。マイコンが壊れることがあります。



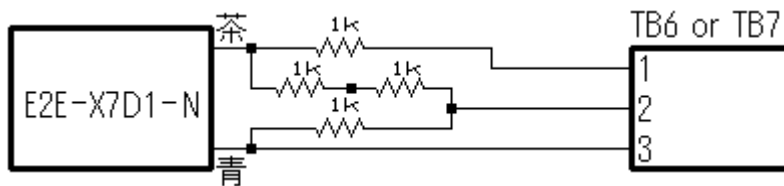
4.4.2 2 線式の場合 (12V 動作のセンサ)

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。

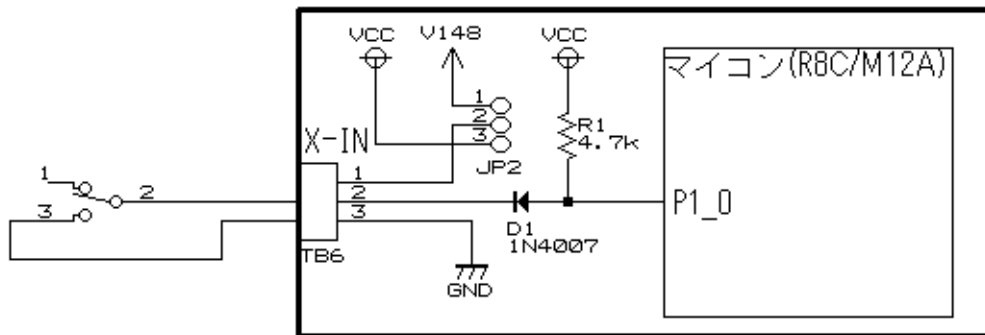




4. 接続

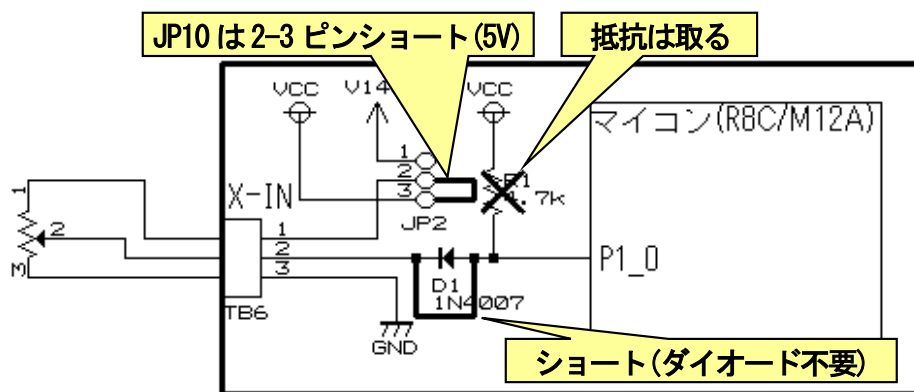
4.4.3 スイッチを接続する場合

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

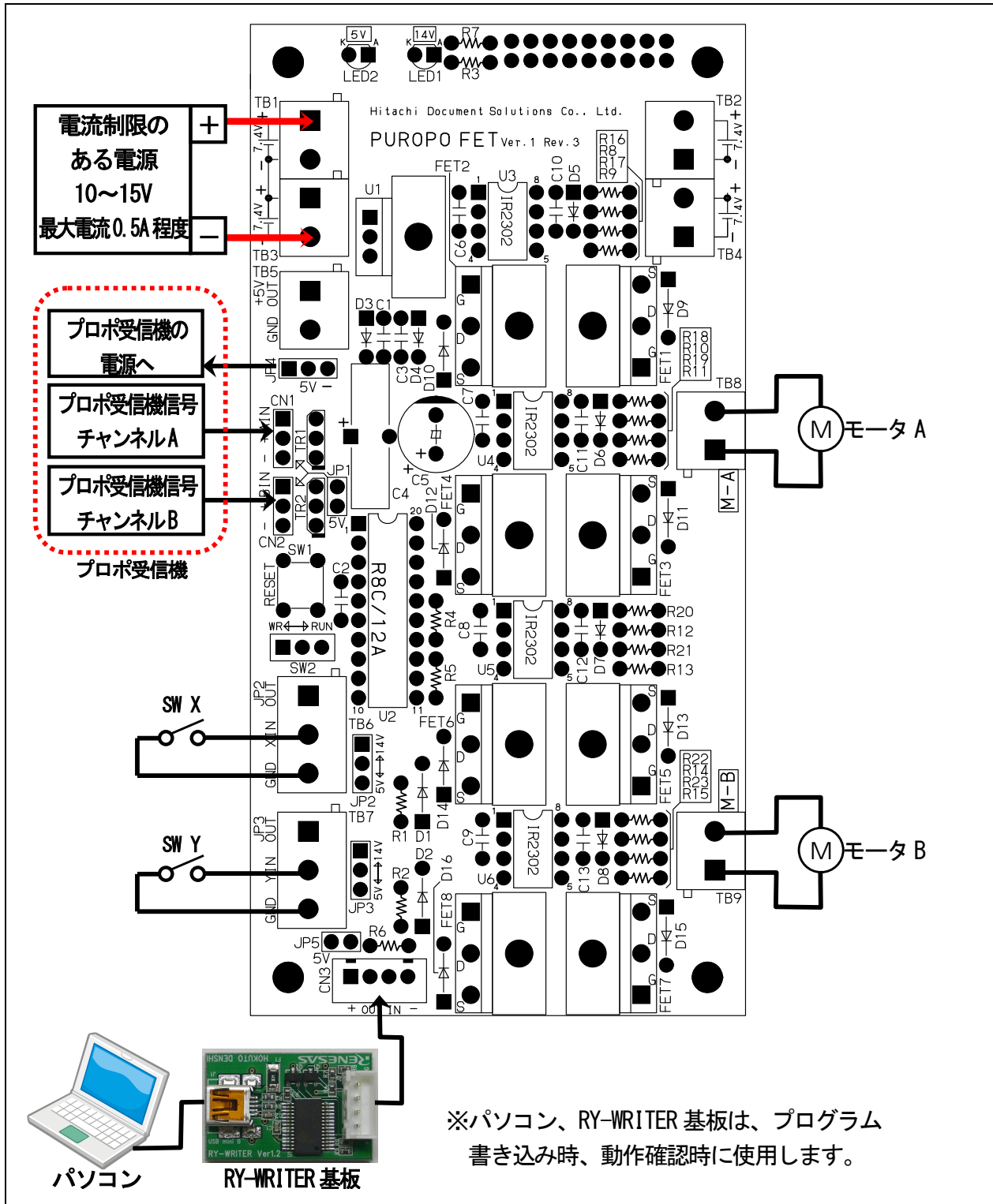
0~5V の信号を入力する場合、D1 はショート、R1 は取り除きます。また、JP2 は 5V 側にします (2-3 ピン側をショート)。接続例を下記に示します。



## 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

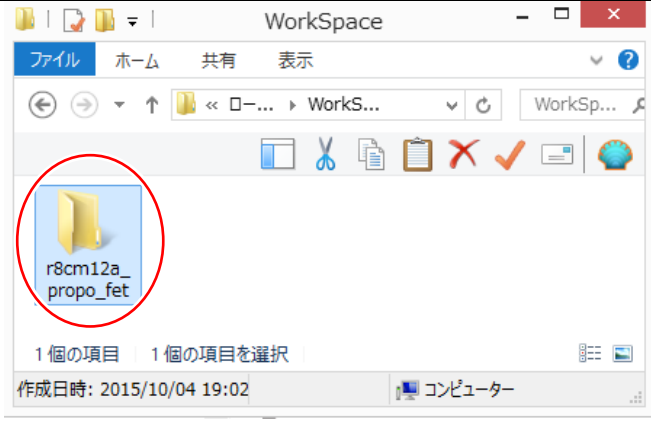
1		<p>HITACHI Inspire the Next</p> <p>マイコンカーラリー販売 サイト <a href="https://www2.himdx.net/mcr/">https://www2.himdx.net/mcr/</a></p>
	<p>日立ドキュメントソリューションズは、マイコンカー製作キットをはじめとして、ものづくりから基本的なマイコン制御を試行錯誤しながら学習ができるマイコン学習教材の開発・販売を行っています。</p>	<p>にアクセスします。「ダウンロード」をクリックします。</p>

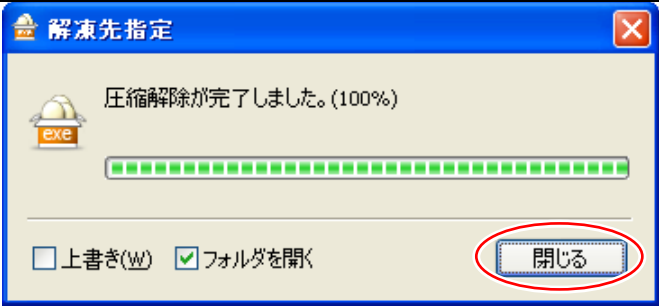
2	<p>ダウンロード(R8C,RXマイコンに関する資料)</p> <p>↓ <a href="#">開発環境に関する資料</a> ↓ <a href="#">マイコンカーキットに関する資料</a> ↓ <a href="#">各種基板に関する資料</a></p> <p>↓ <a href="#">ミニマイコンカーVer.2に関する資料</a> ↓ <a href="#">TypeS基板に関する資料</a> ↓ <a href="#">基板マイコンカーに関する資料</a></p> <p>↓ <a href="#">マトリクス・ジュニア製作キットに関する資料</a> ↓ <a href="#">R8C/M12Aマイコンに関する資料</a></p> <p>↓ <a href="#">RMC-RX62Gボードに関する資料</a> ↓ <a href="#">その他資料</a></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>
---	---	------------------------------

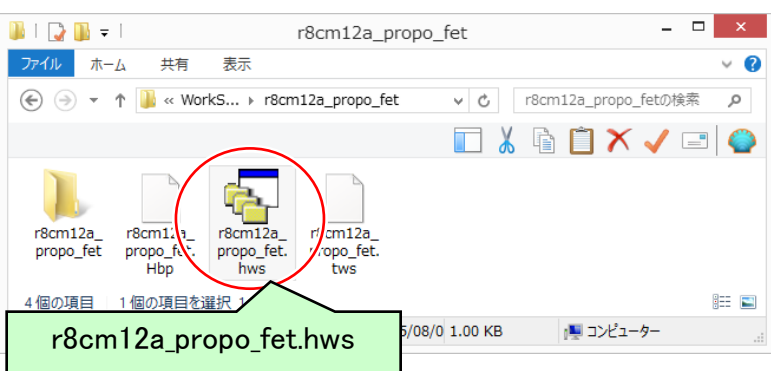
3	<p>ものづくりコンテスト電子回路組立 入力回路練習基板</p> <p>ものづくりコンテスト電子回路組立の入力回路練習基板です。 ※実際の大会のとは、一切関係ありません。大会で使用する基板ではありません。</p>	<p><a href="#">ものづくりコンテスト電子回路組立入力回路練習基板 製作マニュアル</a></p> <p>ル</p> <p>第1.00版 2015.09.28</p>	<p>test.zip 2015.09.28</p> <p>●課題プログラム 作成中です</p>	<p>「r8cm12a_propo_fet.zip」をダウンロードして、解凍します。</p>
	<p>プロボ信号受信FETモータドライブ基板</p> <p>プロボ受信機からの信号(PWM信号)を受信して、大容量モータ2個を制御することのできる基板です。</p>	<p><a href="#">プロボ信号受信FET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</a></p> <p>第1.00版 2015.10.05</p>	<p>r8cm12a_propo_fet.zip 2015.10.05</p>	

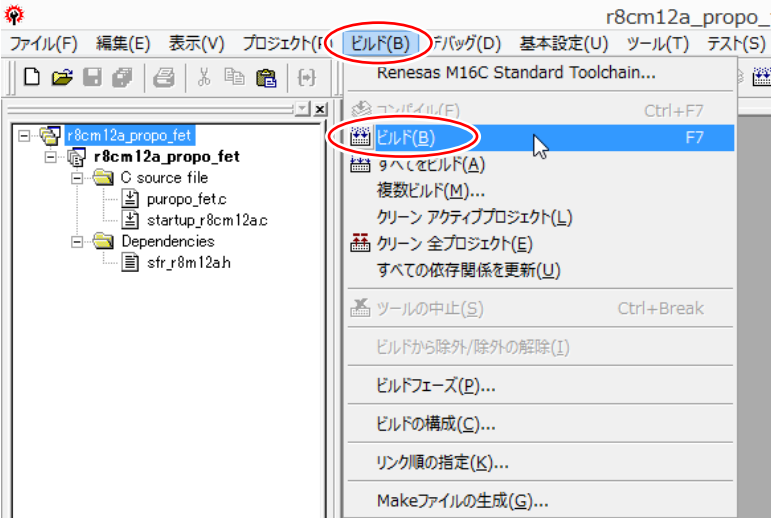
4		<p>「r8cm12a_propo_fet.exe」を実行して、<b>圧縮解除</b>をクリックします。</p> <p>※フォルダは変更できません。変更した場合は、ルネサス統合開発環境の設定を変更する場合があります。</p>
---	--	--

5. 動作確認

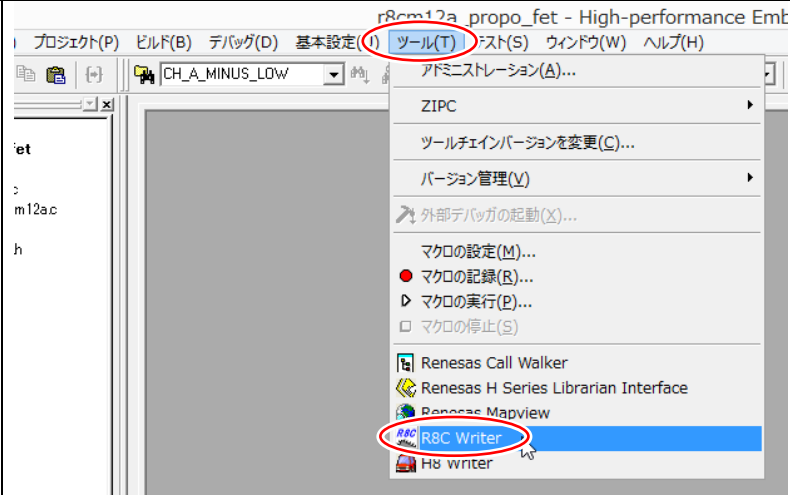
5		<p>解凍が終わったら、自動的に「Cドライブ→Workspace」フォルダが開かれます。今回使用するのは、「r8cm12a_propo_fet」です。</p>
---	---	---

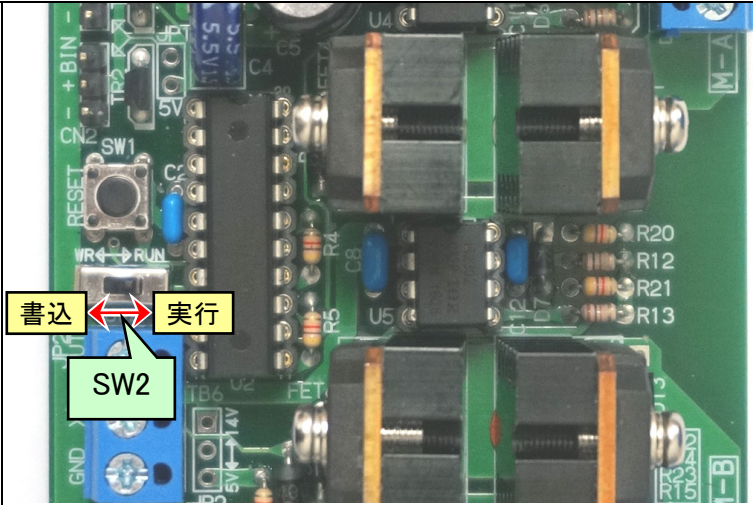
6		<p>閉じるをクリックして終了です。</p>
---	---	------------------------

7		<p>「Cドライブ→ Workspace→ r8cm12a_propo_fet→ r8cm12a_propo_fet.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		<p>「ツール→R8C Writer」で書き込みソフトを立ち上げます。</p>
---	--	---

10		<ol style="list-style-type: none"> <li>①プロポ FET 基板の電源は入れません。</li> <li>②SW2 を基板の外側にします。</li> <li>③電源を入れます。</li> <li>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。</li> <li>⑤正常に書き込みができたなら、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 FET 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。</li> <li>⑥SW を基板の内側にします。</li> </ol>
----	--	--

5. 動作確認

5.3 動作確認

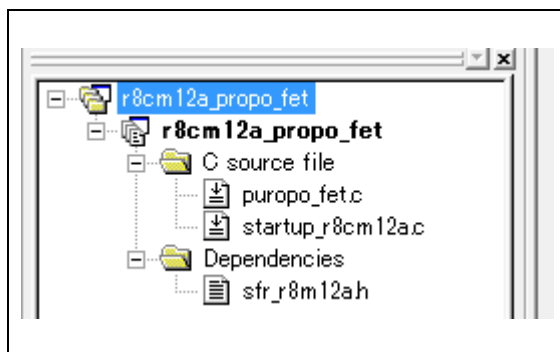
1		<p>プロポ FET 基板の電源はまだ、OFF にしておいてください。</p> <p>Tera Term を立ち上げます。「シリアル」を選択、ポートを RY-WRITER 基板の番号にして、<b>OK</b> をクリックします。</p> <p>※Tera Term がインストールされていない場合、下記からダウンロード、インストールしてください。  <a href="https://osdn.jp/projects/ttssh2/">https://osdn.jp/projects/ttssh2/</a></p>
---	--	--

2	<p>PUROPO FET 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 FET 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プロポ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、プロポ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</p> <p>③プロポ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、プロポ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</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</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</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\_fet」

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_fet.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_fet¥r8cm12a_propo_fet¥puropo_fet.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_fet¥r8cm12a_propo_fet¥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_fet¥r8cm12a_propo_fet¥sfr_r8m12a.h

## 7. プログラム「propo\_fet.c」

### 7.1 概要

このファイルの中に main 関数があり、プロポ FET 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 プロポ信号受信FET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. 08. 09
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_PLUS100 4600 // +100%のパルス値
21 : #define CH_A_STOP 3750 // プロポスティック ニュートラル
22 : #define CH_A_MINUS100 2900 // -100%のパルス値
23 : #define CH_A_MINUS 2750 // いちばん下にしたとき
24 : #define CH_A_MINUS_LIMIT 2450 // これ以下は無効とする(CH_A_MINUS-300くらい)
25 :
26 : // CH_B のプロポ信号 受信パルス幅
27 : #define CH_B_PLUS_LIMIT 5050 // これ以上は無効とする(CH_B_PLUS+300くらい)
28 : #define CH_B_PLUS 4750 // いちばん上にしたとき
29 : #define CH_B_PLUS100 4600 // +100%のパルス値
30 : #define CH_B_STOP 3750 // プロポスティック ニュートラル
31 : #define CH_B_MINUS100 2900 // -100%のパルス値
32 : #define CH_B_MINUS 2750 // いちばん下にしたとき
33 : #define CH_B_MINUS_LIMIT 2450 // これ以下は無効とする(CH_B_MINUS-300くらい)
34 :
35 : // ターミナルブロックとI/O、A/Dのチャンネル
36 : #define TB6 p1_0 // TB6の入力ポート
37 : #define TB7 p1_1 // TB7の入力ポート
38 :
39 : #define TB6_AD 0 // TB6のA/Dチャンネル
40 : #define TB7_AD 1 // TB7のA/Dチャンネル
41 :
42 : //=====
43 : // プロトタイプ宣言
44 : //=====
45 : void init( void );
46 : void set_trcioc( int out );
47 : int get_ad( int ch );
48 :
49 : int get_uart0( char *s );
50 : int put_uart0( char r );
51 : void put_uart0_str( char *str );
52 : void put_uart0_num( long cnt, int keta );
53 : void put_uart0_hex( unsigned long value, int keta );
54 :
55 : void motor_a( int pwm );
56 : void motor_b( int pwm );
57 :
58 : //=====
59 : // グローバル変数
60 : //=====
61 : int cnt_rb; // 1msごとに+1する変数
62 :
63 : long pulse_a; // プロポ受信信号CH_A パルス幅
64 : long pulse_b; // プロポ受信信号CH_B パルス幅
65 :
66 : int pulse_a_inc; // プロポ受信信号CH_A 受信回数(確認用)
67 : int pulse_b_inc; // プロポ受信信号CH_B 受信回数(確認用)
68 :
69 : int pwm_a; // CH_A 設定PWM値 -10~10まで1刻み(10=100%)

```



## 7. プログラム「propo\_fet.c」

```

70 : int pwm_b; // CH_B 設定PWM値 -10~10まで1刻み(10=100%)
71 :
72 : int cnt_a; // パルス幅 chA 受信が無い時間
73 : int cnt_b; // パルス幅 chB 受信が無い時間
74 :
75 : //*****
76 : // メインプログラム
77 : //*****
78 : void main( void )
79 : {
80 :     init(); // 初期化
81 :     asm("FSET I"); // 全体割り込み許可
82 :
83 :     put_uart0_str( "YnYnPUROPO FET PCB Ver.1YnYn" );
84 :
85 :     while( 1 ) {
86 :         if( cnt_rb >= 200 ) {
87 :             cnt_rb = 0;
88 :
89 :             put_uart0_str( "A=" );
90 :             put_uart0_num( pulse_a, 5 );
91 :             put_uart0_str( "(" );
92 :             put_uart0_num( pulse_a_inc, 5 );
93 :             put_uart0_str( ") B=" );
94 :             put_uart0_num( pulse_b, 5 );
95 :             put_uart0_str( "(" );
96 :             put_uart0_num( pulse_b_inc, 5 );
97 :             put_uart0_str( ") TB6=" );
98 :             put_uart0_num( pl_0, 1 );
99 :             put_uart0_str( "(" );
100 :             put_uart0_num( get_ad(TB6_AD), 4 );
101 :             put_uart0_str( ") TB7=" );
102 :             put_uart0_num( pl_1, 1 );
103 :             put_uart0_str( "(" );
104 :             put_uart0_num( get_ad(TB7_AD), 4 );
105 :             put_uart0_str( ")");
106 :
107 :             put_uart0_str( "Yr" ); // 先頭に戻る
108 :         }
109 :     }
110 : }
111 :
112 : //*****
113 : // R8C/M12A スペシャルファンクションレジスタ(SFR)の初期化
114 : //*****
115 : void init( void )
116 : {
117 :     // 内蔵ブルアップ設定
118 :     pul_2 = 1; // 受信機信号Aのトランジスタ出力
119 :     pul_7 = 1; // 受信機信号Bのトランジスタ出力
120 :
121 :     // ポート出力初期設定
122 :     p1_3 = 0; // モータA側のFETドライバ上側
123 :     p3_4 = 0; // モータA側のFETドライバ下側
124 :     p3_5 = 0; // モータB側のFETドライバ上側
125 :     p3_7 = 0; // モータBA側のFETドライバ下側
126 :
127 :     // ポートの入出力設定
128 :     pd1 = 0x18; // ポート1の入出力設定
129 :     pd3 = 0xb0; // ポート3の入出力設定
130 :     pd4 = 0x00; // ポート4の入出力設定
131 :
132 :     // タイマRBの設定 0.5msごとに割り込みを発生させる
133 :     msttrb = 0; // タイマRB2を有効にする
134 :     trbmr = 0x00; // 動作モード、分周比設定
135 :     trbpre = 100-1; // プリスケールレジスタ
136 :     trbpr = 100-1; // プライマリレジスタ
137 :     trbir = 0xc0; // タイマ割り込み要求
138 :     ilvlc = 0x03; // 割り込み優先レベル設定
139 :     trbcr = 0x01; // カウント開始
140 :
141 :     // タイマRCによるPWM出力(3ch出力)
142 :     msttrc = 0; // タイマRCを有効にする
143 :     p12sel1 = 0; // TRCIOB端子の選択
144 :     p12sel0 = 1; // P1_2をTRCIOB端子にする
145 :     iob2_trcior0 = 1; // TRCIOB端子はインプットキャプチャ(両エッジ)で使用
146 :     iob1_trcior0 = 1;
147 :     iob0_trcior0 = 0;
148 :     dfck1_trcdf = 1; // デジタルフィルタ f1
149 :     dfck0_trcdf = 0;
150 :     dfb_trcdf = 1; // TRCIOB端子はデジタルフィルタ使用
151 :     trccr1 = 0xb0; // カウントソース = f8
152 :     tregra = 25000-1; // インプットキャプチャの周期 (1/20M*8)*25000=10ms
153 :     tregrb = 0; // TRCIOB端子のON幅の設定
154 :     imiea_trcier = 1; // trcentがオーバーフローしたときの割り込み許可
155 :     imieb_trcier = 1; // パルスのレベルが変わったときの割り込み許可
156 :     // ※立ち上がり、立ち下がり、両方で割り込み発生
157 :     // 割り込みレベルの設定
158 :     ilvl35 = 1;
159 :     ilvl34 = 0;
160 :     cts_tremer = 1; // TRCCNT カウント開始

```

## 7. プログラム「propo\_fet.c」

```

161 : // タイマRJ2の設定
162 : mstrj = 0; // タイマRJ2を有効にする
163 : p17sel1 = 1; // P1_7をTRJIO端子にする
164 : p17sel0 = 0; // "
165 : tedgsel_trjioc = 0; // Lレベル幅を測定
166 : tipfl_trjioc = 0; // フィルタ有り f1
167 : tipf0_trjioc = 1;
168 : trjmr = 0x13; // パルス幅測定モードに設定 f8
169 : trj = 0xffff; // trjのカウンタはダウンカウント
170 : ilvlb1 = 1; // 割り込みレベルの設定
171 : ilvlb0 = 0;
172 : trjie_trjir = 1; // 割り込み許可
173 : tstart_trjcr = 1; // タイマスタート
174 :
175 : // UART0の設定
176 : mstuart = 0; // UART0を有効にする
177 : p15sel2 = 0; // P1_5 = RXD0端子にする
178 : p15sel1 = 0;
179 : p15sel0 = 1;
180 : p14sel2 = 0; // P1_4 = TXD0端子にする
181 : p14sel1 = 0;
182 : p14sel0 = 1;
183 : u0mr = 0x05; // UARTモード8bit
184 : // 1ストップビット,パリティなし
185 : u0c0 = 0x10; // U0BRGカウンタ値:f1
186 : u0brg = 129; // f1 / (bps*16) - 1
187 : // = 20*10^6/(9600*16)-1=129.2
188 : te_u0c1 = 1; // 送信許可ビット:送信許可
189 : re_u0c1 = 1; // 受信許可ビット:受信許可
190 : }
191 :
192 : //*****
193 : // タイマRB 0.5ms ごとの割り込み処理
194 : //*****
195 : #pragma interrupt /B intTRB(vect=24)
196 : void intTRB( void )
197 : {
198 :     static int cnt0_5ms = 0;
199 :
200 :     trbif_trbir = 0; // フラグのクリア
201 :
202 :     // カウンタインクリメント
203 :     cnt0_5ms++;
204 :     if( cnt_a < 30000 ) cnt_a++;
205 :     if( cnt_b < 30000 ) cnt_b++;
206 :
207 :     switch( cnt0_5ms ) {
208 :     case 1:
209 :         cnt_rb++;
210 :
211 :         // CH_A モータ制御
212 :         if( cnt_a >= 200 ) {
213 :             // 指定時間以上、パルス幅入力となれば停止
214 :             motor_a( 0 );
215 :         } else if( pulse_a >= CH_A_PLUS_LIMIT ) {
216 :             // 上限以上なら0%
217 :             motor_a( 0 );
218 :         } else if( pulse_a <= CH_A_MINUS_LIMIT ) {
219 :             // 下限以上なら0%
220 :             motor_a( 0 );
221 :         } else if( pulse_a >= CH_A_STOP ) {
222 :             // プラスなら
223 :             motor_a( (long)(pulse_a-CH_A_STOP) * 100 / (CH_A_PLUS100-CH_A_STOP) );
224 :         } else {
225 :             // マイナスなら
226 :             motor_a( (long)(pulse_a-CH_A_STOP) * 100 / (CH_A_STOP-CH_A_MINUS100) );
227 :         }
228 :         break;
229 :
230 :     case 2:
231 :         cnt0_5ms = 0;
232 :
233 :         // CH_B モータ制御
234 :         if( cnt_b >= 200 ) {
235 :             // 指定時間以上、パルス幅入力となれば停止
236 :             motor_b( 0 );
237 :         } else if( pulse_b >= CH_B_PLUS_LIMIT ) {
238 :             // 上限以上なら0%
239 :             motor_b( 0 );
240 :         } else if( pulse_b <= CH_B_MINUS_LIMIT ) {
241 :             // 下限以上なら0%
242 :             motor_b( 0 );
243 :         } else if( pulse_b >= CH_B_STOP ) {
244 :             // プラスなら
245 :             motor_b( (long)(pulse_b-CH_B_STOP) * 100 / (CH_B_PLUS100-CH_B_STOP) );
246 :         } else {
247 :             // マイナスなら
248 :             motor_b( (long)(pulse_b-CH_B_STOP) * 100 / (CH_B_STOP-CH_B_MINUS100) );
249 :         }
250 :         break;
251 :     }

```

## 7. プログラム「propo\_fet.c」

```
252 : }
253 :
254 : //*****
255 : // CH_Aのモータ制御
256 : // 引数 -100~100 マイナスは逆転 プラスは逆転
257 : // 戻り値 無し
258 : //*****
259 : void motor_a( int pwm )
260 : {
261 :     static int mode = 0;
262 :     static int pwm_buff = 999;
263 :     static int pwm_max = 0;
264 :     static int pwm_cnt = 0;
265 :
266 :     // 前回と違うなら、更新
267 :     if( pwm != pwm_buff ) {
268 :         pwm_buff = pwm;
269 :         pwm_max = pwm / 10;
270 :
271 :         if( pwm_max >= 10 ) pwm_max = 10;
272 :         if( pwm_max <= -10 ) pwm_max = -10;
273 :
274 :         if( pwm_max == 0 ) {
275 :             mode = 0;
276 :         }
277 :     }
278 :
279 :     pwm_a = pwm_max;
280 :
281 :     // モータ制御
282 :     switch( mode ) {
283 :     case 0: // ブレーキ
284 :         p1_3 = 0;
285 :         p3_4 = 0;
286 :         mode = 1;
287 :         break;
288 :
289 :     case 1: // ブレーキのまま待機、PWM値が変わったらパターン移る
290 :         if( pwm_max > 0 ) {
291 :             // 正転
292 :             p1_3 = 1;
293 :             pwm_cnt = 0;
294 :             mode = 11;
295 :         } else if( pwm_max < 0 ) {
296 :             // 逆転
297 :             p3_4 = 1;
298 :             pwm_cnt = 0;
299 :             mode = 21;
300 :         }
301 :         break;
302 :
303 :     case 11: // 正転
304 :         if( pwm_max == 10 ) {
305 :             // 100%でも、一瞬だけは"0"にする
306 :             p1_3 = 0;
307 :             asm( "nop" );
308 :             asm( "nop" );
309 :             asm( "nop" );
310 :             p1_3 = 1;
311 :             break;
312 :         }
313 :         if( pwm_max < 0 ) {
314 :             // 逆転
315 :             p1_3 = 0;
316 :             p3_4 = 1;
317 :             mode = 21;
318 :             break;
319 :         }
320 :
321 :         pwm_cnt++;
322 :         if( pwm_cnt >= 10 ) {
323 :             // 10msたったら正転にする
324 :             p1_3 = 1;
325 :             pwm_cnt = 0;
326 :             break;
327 :         }
328 :         if( pwm_max == pwm_cnt ) {
329 :             // 一致したらブレーキ
330 :             p1_3 = 0;
331 :             break;
332 :         }
333 :         break;
334 :
335 :     case 21: // 逆転
336 :         if( pwm_max == -10 ) {
337 :             // -100%でも、一瞬だけは"0"にする
338 :             p3_4 = 0;
339 :             asm( "nop" );
340 :             asm( "nop" );
341 :             asm( "nop" );
342 :             p3_4 = 1;
```

## 7. プログラム「propo\_fet.c」

```

343 :         break;
344 :     }
345 :     if( pwm_max > 0 ) {
346 :         // 正転
347 :         p3_4 = 0;
348 :         p1_3 = 1;
349 :         mode = 11;
350 :         break;
351 :     }
352 :
353 :     pwm_cnt++;
354 :     if( pwm_cnt >= 10 ) {
355 :         // 10msたったら逆転にする
356 :         p3_4 = 1;
357 :         pwm_cnt = 0;
358 :         break;
359 :     }
360 :     if( (-pwm_max) == pwm_cnt ) {
361 :         // 一致したらブレーキ
362 :         p3_4 = 0;
363 :         break;
364 :     }
365 :     break;
366 : }
367 : }
368 :
369 : //*****
370 : // CH_Bのモータ制御
371 : // 引数  -100~100 マイナスは逆転 プラスは逆転
372 : // 戻り値 無し
373 : //*****
374 : void motor_b( int pwm )
375 : {
376 :     static int mode = 0;
377 :     static int pwm_buff = 999;
378 :     static int pwm_max = 0;
379 :     static int pwm_cnt = 0;
380 :
381 :     // 前回と違うなら、更新
382 :     if( pwm != pwm_buff ) {
383 :         pwm_buff = pwm;
384 :         pwm_max = pwm / 10;
385 :
386 :         if( pwm_max >= 10 ) pwm_max = 10;
387 :         if( pwm_max <= -10 ) pwm_max = -10;
388 :
389 :         if( pwm_max == 0 ) {
390 :             mode = 0;
391 :         }
392 :     }
393 :
394 :     pwm_b = pwm_max;
395 :
396 :     // モータ制御
397 :     switch( mode ) {
398 :     case 0: // ブレーキ
399 :         p3_5 = 0;
400 :         p3_7 = 0;
401 :         mode = 1;
402 :         break;
403 :
404 :     case 1: // ブレーキのまま待機、PWM値が変わったらパターン移る
405 :         if( pwm_max > 0 ) {
406 :             // 正転
407 :             p3_5 = 1;
408 :             pwm_cnt = 0;
409 :             mode = 11;
410 :         } else if( pwm_max < 0 ) {
411 :             // 逆転
412 :             p3_7 = 1;
413 :             pwm_cnt = 0;
414 :             mode = 21;
415 :         }
416 :         break;
417 :
418 :     case 11: // 正転
419 :         if( pwm_max == 10 ) {
420 :             // 100%でも、一瞬だけは"0"にする
421 :             p3_5 = 0;
422 :             asm( "nop" );
423 :             asm( "nop" );
424 :             asm( "nop" );
425 :             p3_5 = 1;
426 :             break;
427 :         }
428 :         if( pwm_max < 0 ) {
429 :             // 逆転
430 :             p3_5 = 0;
431 :             p3_7 = 1;
432 :             mode = 21;
433 :             break;

```

## 7. プログラム「propo\_fet.c」

```

434 :     }
435 :
436 :     pwm_cnt++;
437 :     if( pwm_cnt >= 10 ) {
438 :         // 10msたったら正転にする
439 :         p3_5 = 1;
440 :         pwm_cnt = 0;
441 :         break;
442 :     }
443 :     if( pwm_max == pwm_cnt ) {
444 :         // 一致したらブレーキ
445 :         p3_5 = 0;
446 :         break;
447 :     }
448 :     break;
449 :
450 :     case 21: // 逆転
451 :     if( pwm_max == -10 ) {
452 :         // -100%でも、一瞬だけは"0"にする
453 :         p3_7 = 0;
454 :         asm( "nop" );
455 :         asm( "nop" );
456 :         asm( "nop" );
457 :         p3_7 = 1;
458 :         break;
459 :     }
460 :     if( pwm_max > 0 ) {
461 :         // 正転
462 :         p3_7 = 0;
463 :         p3_5 = 1;
464 :         mode = 11;
465 :         break;
466 :     }
467 :
468 :     pwm_cnt++;
469 :     if( pwm_cnt >= 10 ) {
470 :         // 10msたったら逆転にする
471 :         p3_7 = 1;
472 :         pwm_cnt = 0;
473 :         break;
474 :     }
475 :     if( (-pwm_max) == pwm_cnt ) {
476 :         // 一致したらブレーキ
477 :         p3_7 = 0;
478 :         break;
479 :     }
480 :     break;
481 : }
482 : }
483 :
484 : //*****
485 : // タイマRC 割り込み処理
486 : //*****
487 : #pragma interrupt intTRC(vect=7)
488 : void intTRC( void )
489 : {
490 :     static unsigned int pulse_a_start = 0;
491 :
492 :     if( imfb_trcsr == 1 ) {
493 :         imfb_trcsr = 0;
494 :
495 :         if( p1_2 == 0 ) {
496 :             pulse_a_start = trcgrb;
497 :         } else {
498 :             if( pulse_a_start < trcgrb ) {
499 :                 pulse_a = trcgrb - pulse_a_start;
500 :             } else {
501 :                 pulse_a = (long)25000 + trcgrb - pulse_a_start;
502 :             }
503 :             pulse_a_inc++;
504 :             cnt_a = 0;
505 :         }
506 :     }
507 : }
508 :
509 : //*****
510 : // タイマRJ2 割り込み処理
511 : //*****
512 : #pragma interrupt intTRJ2(vect=22)
513 : void intTRJ2( void )
514 : {
515 :     static long pulse_b_start = 0xffff;
516 :
517 :     unsigned char c;
518 :
519 :     trjif_trjir = 0;
520 :
521 :     if( tundf_trjcr == 1 ) {
522 :         // アンダーフロー
523 :         c = trjcr & 0xdf;
524 :         trjcr = c;

```

## 7. プログラム「propo\_fet.c」

```

525 :     pulse_b_start += 0x10000;
526 : }
527 :
528 : if( tedgf_trjcr == 1 ) {
529 :     // 有効エッジ(立ち下がり)あり
530 :     c = trjcr & 0xef;
531 :     trjcr = c;
532 :
533 :     tstart_trjcr = 0;           // タイマ停止
534 :     pulse_b = pulse_b_start - trj;
535 :     trj = 0xffff;
536 :     pulse_b_start = 0xffff;
537 :     tstart_trjcr = 1;         // タイマスタート
538 :
539 :     pulse_b_inc++;
540 :     cnt_b = 0;
541 : }
542 : }
543 :
544 : //*****
545 : // AN端子をA/D変換
546 : // 引数   チャンネル番号(0~4, 7)
547 : // 戻り値 A/D変換値(0~1023) 9999:チャンネル選択エラー
548 : //*****
549 : int get_ad( int ch )
550 : {
551 :     int i;
552 :
553 :     // A/Dコンバータの設定
554 :     mstad = 0;           // A/Dコンバータを有効にする
555 :     admod = 0x03;       // 単発モードに設定
556 :     switch( ch ) {
557 :         case 0: adinsel = 0x00; break; // 入力端子AN0(P1_0)を選択
558 :         case 1: adinsel = 0x01; break; // 入力端子AN1(P1_1)を選択
559 :         case 2: adinsel = 0x40; break; // 入力端子AN2(P1_2)を選択
560 :         case 3: adinsel = 0x41; break; // 入力端子AN3(P1_3)を選択
561 :         case 4: adinsel = 0x80; break; // 入力端子AN4(P1_4)を選択
562 :         case 7: adinsel = 0x81; break; // 入力端子AN7(P1_7)を選択
563 :         default: mstad = 1; return 9999;
564 :     }
565 :     adcon0 = 0x01;       // A/D変換スタート
566 :
567 :     while( adcon0 & 0x01 ); // A/D変換終了待ち
568 :
569 :     switch( ch ) {
570 :         case 0: i = ad0; break; // 入力端子AN0(P1_0)を選択
571 :         case 1: i = ad1; break; // 入力端子AN1(P1_1)を選択
572 :         case 2: i = ad0; break; // 入力端子AN2(P1_2)を選択
573 :         case 3: i = ad1; break; // 入力端子AN3(P1_3)を選択
574 :         case 4: i = ad0; break; // 入力端子AN4(P1_4)を選択
575 :         case 7: i = ad1; break; // 入力端子AN7(P1_7)を選択
576 :     }
577 :
578 :     mstad = 1;           // A/Dコンバータを無効にする
579 :     return i;
580 : }
581 :
582 : //*****
583 : // 1文字受信
584 : // 引数   受信文字格納アドレス
585 : // 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
586 : //*****
587 : int get_uart0( char *s )
588 : {
589 :     int ret = 0, data;
590 :     char c;
591 :
592 :     if (ri_u0c1 == 1){ // 受信データあり?
593 :         data = u0rb;
594 :         *s = (char)data;
595 :         ret = 1;
596 :         if( data & 0xf000 ) { // エラーあり?
597 :             // エラー時は再設定
598 :             re_u0c1 = 0;
599 :             te_u0c1 = 0;
600 :             c = u0mr;
601 :             u0mr &= 0xf8;
602 :             u0mr = c;
603 :             re_u0c1 = 1;
604 :             te_u0c1 = 1;
605 :             ret = -1;
606 :         }
607 :     }
608 :     return ret;
609 : }
610 :
611 : //*****
612 : // 1文字出力
613 : // 引数   送信データ
614 : // 戻り値 0:送信中のため、送信できず 1:送信セット完了
615 : //*****

```

## 7. プログラム「propo\_fet.c」

```
616 : int put_uart0( char r )
617 : {
618 :     if(ti_u0c1 == 1) { // 送信データなし?
619 :         u0tbl = r;
620 :         return 1;
621 :     } else {
622 :         // 先に送信中(今回のデータは送信せずに終了)
623 :         return 0;
624 :     }
625 : }
626 :
627 : //*****
628 : // 文字列出力
629 : // 引数 送信データ
630 : // 戻り値 無し
631 : //*****
632 : void put_uart0_str( char *str )
633 : {
634 :     while(*str != '\0') {
635 :         // '\n' は '\r\n' に変換する
636 :         if( *str == '\n' ) while( !put_uart0( '\r' ) );
637 :         while( !put_uart0( *str ) ); // 1文字送信
638 :         str++;
639 :     }
640 : }
641 :
642 : //*****
643 : // 10進数出力
644 : // 引数 long 値, int 桁(1~8)
645 : // 戻り値 無し
646 : //*****
647 : void put_uart0_num( long value, int keta )
648 : {
649 :     int i = keta;
650 :     long temp = 0;
651 :
652 :     do {
653 :         temp <<= 4;
654 :         temp += value % 10;
655 :         value /= 10;
656 :     } while( --i );
657 :
658 :     do {
659 :         while( !put_uart0( (temp & 0xf) + '0' ) ); // 1文字送信
660 :         temp >>= 4;
661 :     } while( --keta );
662 : }
663 :
664 : //*****
665 : // 16進数出力
666 : // 引数 unsigned long 値, int 桁(1~8)
667 : // 戻り値 無し
668 : //*****
669 : void put_uart0_hex( unsigned long value, int keta )
670 : {
671 :     char temp;
672 :
673 :     do {
674 :         temp = value >> (keta * 4 - 4);
675 :         temp &= 0xf;
676 :         temp += (temp < 10) ? '0' : ('a' - 10);
677 :
678 :         while( !put_uart0( temp ) ); // 1文字送信
679 :     } while( --keta );
680 : }
681 :
682 : //*****
683 : // end of file
684 : //*****
```

### 7.3 関数

#### ■motor\_a 関数

書式	<code>void motor_a( int pwm );</code>
内容	モータ(チャンネル A)を制御します。
引数	int モータ(チャンネル A)の PWM -100~-1:逆転で回転、割合は 10%ごとに切り捨てられます。例) -9→0%、-95%→-90% 0 :停止 1~100 :正転で回転、割合は 10%ごとに切り捨てられます。例) 9→0%、95%→90%
戻り値	無し
例	<code>motor_a( 65 ); // 60%で正転</code> <code>motor_a( -35 ); // -30%で逆転</code> <code>motor_a( 0 ); // 停止</code>

#### ■motor\_b 関数

書式	<code>void motor_b( int pwm );</code>
内容	モータ(チャンネル B)を制御します。
引数	int モータ(チャンネル A)の PWM -100~-1:逆転で回転、割合は 10%ごとに切り捨てられます。例) -9→0%、-95%→-90% 0 :停止 1~100 :正転で回転、割合は 10%ごとに切り捨てられます。例) 9→0%、95%→90%
戻り値	無し
例	<code>motor_b( 65 ); // 60%で正転</code> <code>motor_b( -35 ); // -30%で逆転</code> <code>motor_b( 0 ); // 停止</code>

#### ■get\_ad 関数

書式	<code>int get_ad( int ch );</code>
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	<code>get_ad( TB6_AD ); // TB6 の入力端子の電圧を測定(0~1023)</code> <code>get_ad( TB7_AD ); // TB7 の入力端子の電圧を測定(0~1023)</code>

#### ■put\_uart0\_str 関数

書式	<code>void put_uart0_str( char *str );</code>
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	<code>// パソコンの TeraTerm などの通信ソフトに、文字列を表示</code> <code>put_uart0_str( "¥n¥nPUROPO FET PCB Ver. 1¥n¥n" );</code>



■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 &gt; 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 = 0x1abcd; 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

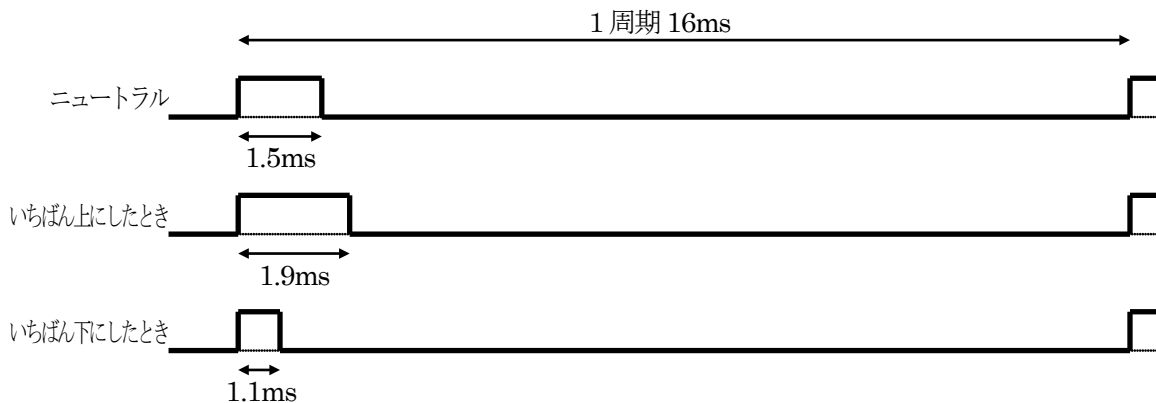
```

58 : //=====
59 : // グローバル変数
60 : //=====
61 : int cnt_rb;                // 1ms ごとに + 1 する変数
62 :
63 : long pulse_a;             // プロポ受信信号 CH_A パルス幅
64 : long pulse_b;             // プロポ受信信号 CH_B パルス幅
65 :
66 : int pulse_a_inc;          // プロポ受信信号 CH_A 受信回数(確認用)
67 : int pulse_b_inc;          // プロポ受信信号 CH_B 受信回数(確認用)
68 :
69 : int pwm_a;                // CH_A 設定 PWM 値 -10~10 まで 1 刻み(10=100%)
70 : int pwm_b;                // CH_B 設定 PWM 値 -10~10 まで 1 刻み(10=100%)
71 :
72 : int cnt_a;                // パルス幅 chA 受信が無い時間
73 : int cnt_b;                // パルス幅 chB 受信が無い時間
    
```

63 行	<p>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、<math>0.4 \mu s</math> です。パルス幅が、<math>1.5ms (=1500 \mu s)</math> の場合は、変数の値 <math>= 1500 / 0.4 = 3750</math> となります。</p>
64 行	<p>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。</p>

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\_fet.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、0.4μsです。 よって、 変数の値=ON幅 / 0.4μs = 1500 / 0.4 = <b>3750</b> 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 / 0.4μs = 1900 / 0.4 = <b>4750</b> 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 / 0.4μs = 1100 / 0.4 = <b>2750</b> となります。

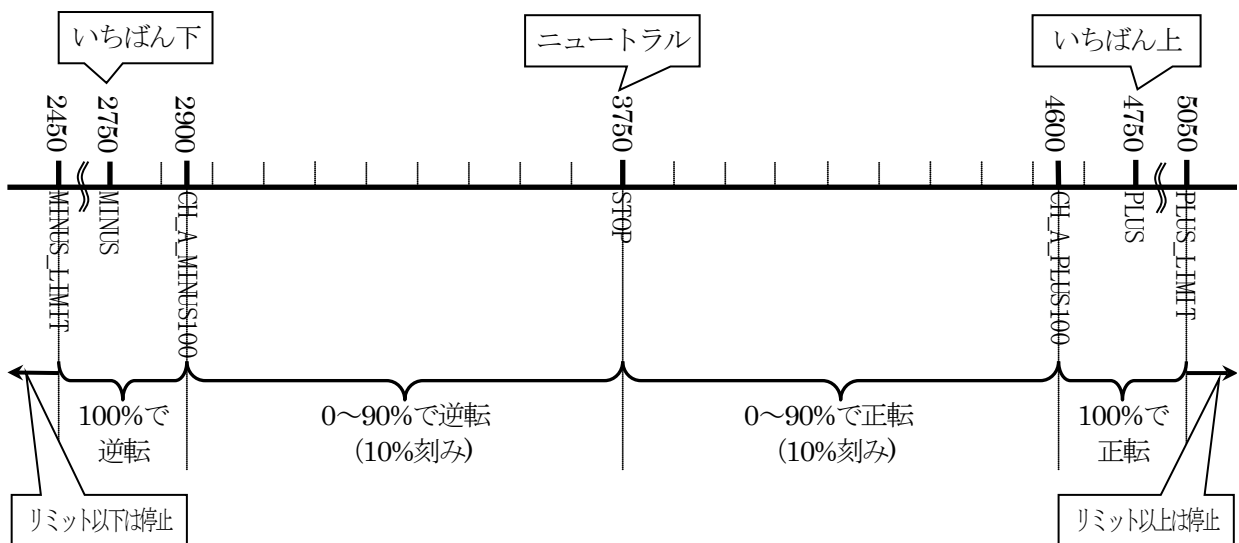
pulse\_a 変数、pulse\_b 変数の値を if 文などでチェックすることによって、モータを動作させます。

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    // 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_PLUS100      4600    // +100%のパルス値
21 : #define CH_A_STOP          3750    // プロポスティック ニュートラル
22 : #define CH_A_MINUS100     2900    // -100%のパルス値
23 : #define CH_A_MINUS        2750    // いちばん下にしたとき
24 : #define CH_A_MINUS_LIMIT  2450    // 以下は無効とする(CH_A_MINUS-300くらい)
    
```

「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

```

75 : //*****
76 : // メインプログラム
77 : //*****
78 : void main( void )
79 : {
80 :     init();                // 初期化
81 :     asm("FSET I");        // 全体割り込み許可
82 :
83 :     put_uart0_str( "¥n¥nPUROPO FET PCB Ver.1¥n¥n" );
84 :
85 :     while( 1 ) {
86 :         if( cnt_rb >= 200 ) {
87 :             cnt_rb = 0;
88 :
89 :             put_uart0_str( "A=" );
90 :             put_uart0_num( pulse_a, 5 );
91 :             put_uart0_str( "(" );
92 :             put_uart0_num( pulse_a_inc, 5 );
93 :             put_uart0_str( ") B=" );
94 :             put_uart0_num( pulse_b, 5 );
95 :             put_uart0_str( "(" );
96 :             put_uart0_num( pulse_b_inc, 5 );
97 :             put_uart0_str( ") TB6=" );
98 :             put_uart0_num( p1_0, 1 );
99 :             put_uart0_str( "(" );
100 :             put_uart0_num( get_ad(TB6_AD), 4 );
101 :             put_uart0_str( ") TB7=" );
102 :             put_uart0_num( p1_1, 1 );
103 :             put_uart0_str( "(" );
104 :             put_uart0_num( get_ad(TB7_AD), 4 );
105 :             put_uart0_str( ")") );
106 :
107 :             put_uart0_str( "¥r" );    // 先頭に戻る
108 :         }
109 :     }
110 : }

```

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 FET 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。

7. プログラム「propo\_fet.c」

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。
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_PLUS100       4600    // 「CH_A_PLUS」の値より100~150小さい値にし
                                         // てください
21 : #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
22 : #define CH_A_MINUS100      2900    // 「CH_A_MINUS」の値より100~150小さい値に
                                         // してください
23 : #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
24 : #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 これ以下の値は、不正な値としてモ
                                         // ータを停止します
    
```

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。

#### 7.4.4 モータを動作させている部分

モータを動作させている部分は、タイマ RB 割り込み内にあり、0.5ms ごとに実行されます。

モータ A とモータ B を交互に制御しています。よって、それぞれのモータは 1ms ごとに制御することになります。割り込みプログラムを下記に示します。

```
192 : //*****
193 : // タイマ RB 0.5ms ごとの割り込み処理
194 : //*****
195 : #pragma interrupt /B intTRB(vect=24)
196 : void intTRB( void )
197 : {
198 :     static int cnt0_5ms = 0;
199 :
200 :     trbif_trbir = 0;                // フラグのクリア
201 :
202 :     // カウンタインクリメント
203 :     cnt0_5ms++;
204 :     if( cnt_a < 30000 ) cnt_a++;
205 :     if( cnt_b < 30000 ) cnt_b++;
206 :
207 :     switch( cnt0_5ms ) {
208 :     case 1:
209 :         cnt_rb++;
210 :
211 :         // CH_A モータ制御
212 :         モータ A を動作させるプログラム
228 :         break;
229 :
230 :     case 2:
231 :         cnt0_5ms = 0;
232 :
233 :         // CH_B モータ制御
234 :         モータ B を動作させるプログラム
250 :         break;
251 :     }
252 : }
```

7. プログラム「propo\_fet.c」

212～227 行が、モータ A を動かすプログラムです。

```

211 : // CH_A モータ制御
212 : if( cnt_a >= 200 ) { パルス入力が200ms以上なければ
213 : // 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
214 : motor_a( 0 );
215 : } else if( pulse_a >= CH_A_PLUS_LIMIT ) { リミット値以上なら予期せぬ状態
216 : // 上限以上なら0% と判断し、停止
217 : motor_a( 0 );
218 : } else if( pulse_a <= CH_A_MINUS_LIMIT ) { リミット値以下なら予期せぬ状態
219 : // 下限以上なら0% と判断し、停止
220 : motor_a( 0 );
221 : } else if( pulse_a >= CH_A_STOP ) { ニュートラル以上なら正転
222 : // プラスなら
223 : motor_a( (long)(pulse_a-CH_A_STOP) * 100 / (CH_A_PLUS100-CH_A_STOP) );
224 : } else { ニュートラル以下なら逆転
225 : // マイナスなら
226 : motor_a( (long)(pulse_a-CH_A_STOP) * 100 / (CH_A_STOP-CH_A_MINUS100) );
227 : }
    
```

ニュートラル以上の場合、下記の計算を行って、パルス幅の値を「0～100%」に変換します。

※motor\_a 関数には 0～100 の値を設定します。motor\_a 関数内で 10%刻みに変換し、モータを制御します。

$$\text{PWM 値} = ( \text{pulse}_a - \text{CH\_A\_STOP} ) / ( \text{CH\_A\_PLUS100} - \text{CH\_A\_STOP} ) * 100$$

pulse\_a 変数の値と PWM の関係を、下表に示します。

pulse_a の値	PWM 値
3750 (ニュートラル)	$\begin{aligned} & ( \text{pulse}_a - \text{CH\_A\_STOP} ) / ( \text{CH\_A\_PLUS100} - \text{CH\_A\_STOP} ) * 100 \\ & = ( 3750 - 3750 ) / ( 4600 - 3750 ) * 100 \\ & = 0 / 850 * 100 \\ & = 0 \end{aligned}$
4600 (100%の値)	$\begin{aligned} & ( \text{pulse}_a - \text{CH\_A\_STOP} ) / ( \text{CH\_A\_PLUS100} - \text{CH\_A\_STOP} ) * 100 \\ & = ( 4600 - 3750 ) / ( 4600 - 3750 ) * 100 \\ & = 850 / 850 * 100 \\ & = 100 \end{aligned}$
4100	$\begin{aligned} & ( \text{pulse}_a - \text{CH\_A\_STOP} ) / ( \text{CH\_A\_PLUS100} - \text{CH\_A\_STOP} ) * 100 \\ & = ( 4100 - 3750 ) / ( 4600 - 3750 ) * 100 \\ & = 350 / 850 * 100 \\ & = 41 \quad \text{※実際は、1の位が切り捨てられ40%で正転します} \end{aligned}$

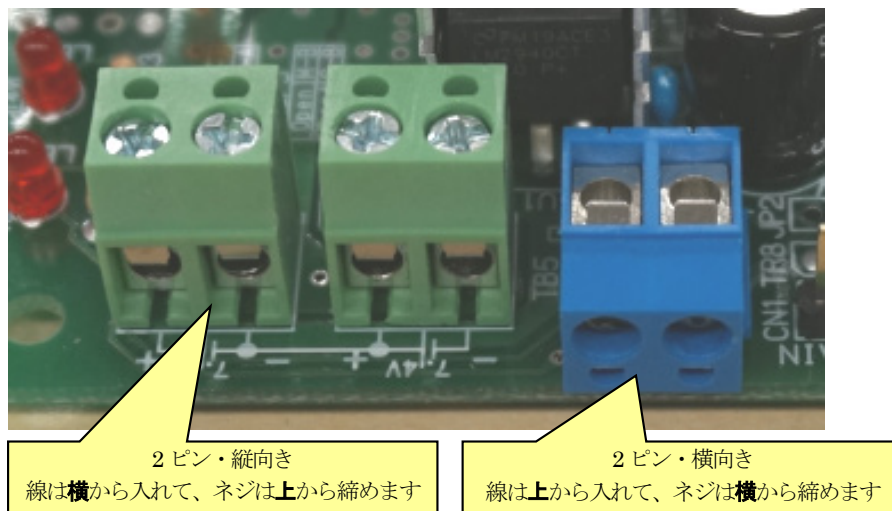
ニュートラル以下の場合も同様の計算です。

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。



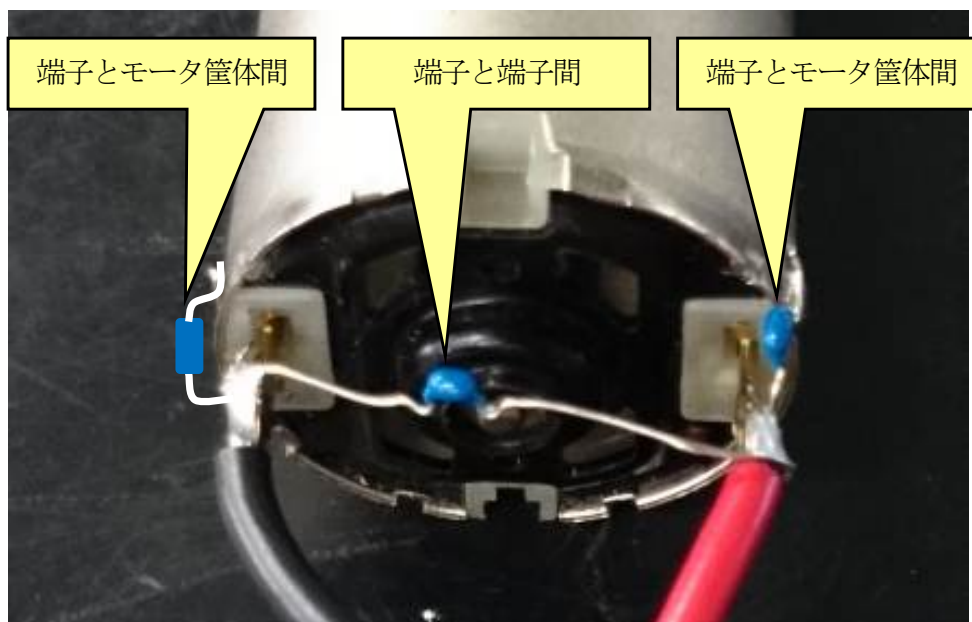
ターミナルブロックは、(株)秋月電子通商で販売しています。ピン数、向きと、型式の関係を下記に示します。

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

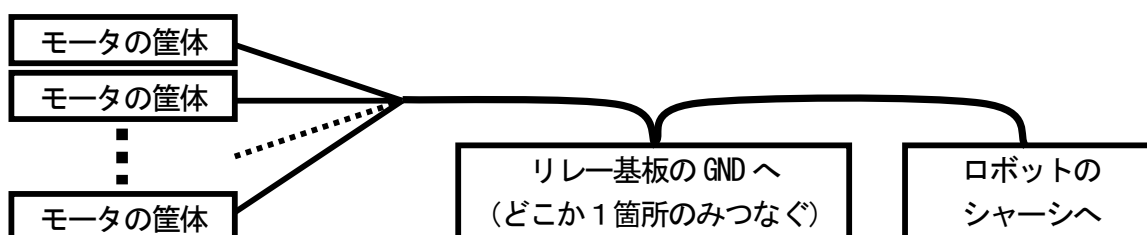


## 8.2 モータの加工

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