

### USB I/Oコントローラ 取り扱い説明書

#### 概要

パソコンのUSBポートを通して10個のリレーをON/OFFできる汎用入出力ボードです。

コントローラにはワンチップCPU (PIC16F628) を搭載しており、ホストパソコンからは文字コマンド (ASCII 文字列) を送るだけで簡単にリレーをコントロールできます。また、「ロボットアーム・MR-999」にダイレクト接続できるコネクタも備えていますので、ロボットプログラミングの入門としても最適です。

付属のドライバは、仮想的に、RBIO-2Uを汎用のRS-232Cポートとして扱います。

この機能により、殆どのRS-232C用のソフトがそのまま利用可能です。

(一部、特殊な仕様のソフトでは利用できない場合があります)

#### 仕様

電源	: DC9 ~ 12V
電流	: 230mA (9V時最大) \ 270mA (12V時最大)
電源端子	: 2.1 センタ+ (プラス) ACアダプタ入力
出力	: 10組 / リレー接点 /NC,NO,COMの3線
式接点容量	: DC24V1A/AC125V0.5A
入力	: 4組 / 対GND接続接点入力 / 予備5V電源端子付き
USB I/F	: USB1.1 準拠 基板付けBタイプコネクタ
動作環境	: Windows (7,8,10,11)
専用端子	: ロボットアーム「MR-999」用 : 5 関節、正転、逆転、停止
コマンド	: 個別I/Oのコントロールコマンドおよびロボットアーム「MR-999」 専用 コマンド
その他	: リモート用電源ON/OFFモード

#### 必要な機器

本ボードを動作 (確認動作) させるのに必要な機材は以下の通りです。

電源 9V ~ 12V の安定化された300mA程度を供給できる事。

プラグはセンターが+で2.1 の事

USB が機能するWindowsマシン。

(MAC 及びLinuxマシンでの動作は未確認)

USB A-B ケーブル。



\* 注意 : 本基板の電源は最大12Vで、車のバッテリーでも動作可能です。しかし、車のバッテリーの様な供給能力の大きい電源から供給する場合は、短絡事故等で火災になる場合がありますので、必ず途中にヒューズを挿入してご利用ください。

# 目次

概要 .....	1
仕様 .....	1
必要な機器 .....	1
概要 .....	3
RB10-2U の接続 .....	3
デバイスドライバのインストール .....	3
ドライバの削除 .....	4
COM ポート番号と複数枚の RB10-2U の使用に付いて .....	4
付属 CD の内容 .....	5
動作試験 .....	5
本結線 .....	6
個別の I / O として使う .....	6
ロボットを接続する .....	7
入力端子 .....	8
コントロールの方法 .....	8
コントロールコマンド .....	8
コマンド : PCRnp .....	9
コマンド : PCDst .....	9
コマンド : PCTnttt .....	9
コマンド : PCAn .....	10
サーチコマンド .....	11
実時間入力 .....	11
AT コマンド .....	11
緊急停止 .....	11
コマンドステータス .....	11
リリースステータス .....	11
Windows でコントロールする場合 .....	12
うまく動作しない場合は .....	12
技術資料 .....	13
高容量リレーを追加する .....	19
付録 .....	20
コマンド一覧表 .....	21

## 概要

本基板ではインターフェースにUSBを採用しています。

ドライバをインストールする事により、仮想的なシリアルポートが作成され、従来のシリアルポート用のソフトがそのまま動作します。(全てのアプリケーションがそのまま動作する事を保障するものではありません)

RB10-1 (注: 弊社製 I/O コントローラでインターフェースがシリアルになっている以外は RB10-2U と同一) では PC に対してシリアル (COM ポート経由) で接続しました。

一方、RB10-2U は USB による接続となります。USB コントローラには FTDI 社の FT232AM を使用し、RB10 のコントローラとは、ボード上でシリアル接続されています。

通信信号の流れが RB10-1 では、・プログラム COM ポート RB10 のコントローラとなっています。一方 RB10-2U では、・プログラム 仮想 COM ポート USB ホスト RB10-2U 上の USB デバイス RB10 のコントローラと通過する経路が複雑化しています。

しかしながら、デバイスドライバの仮想 COM ポート機能が実装されるため、・プログラム 仮想 COM ポート RB10 のコントローラと、RB10-1 と同じ形態となります。\*

\* 仮想 COM ポートは PC 上のソフト (OS とデバイスドライバ) が、存在しないハードウェアをソフトウェアにより、仮想的に作成します。これにより、アプリケーションからは、ハード的な COM ポートが増設されたのと同じ様に見え、またコントロールも通常の COM ポートと同様に行う事ができます。

## RB10-2U の接続

RB10-2U はページ 6 の図 1 の様に、USB ポートに接続して利用します。

USB に接続した時点で、RB10-2U の電源の有無に関わらず、USB 機器として認識されます (USB コントローラ部はバスパワー動作)

初めての接続では、USB のドライバを組み込む画面が表示されます。

ドライバの組み込みは次項を参照してください。

## デバイスドライバのインストール

初めて RB10-2U を PC に接続する場合は、デバイスドライバの組み込みが必要になります。

RB10-2U の USB インターフェースは FTDI 社 (<http://www.ftdichip.com>) の USB インターフェース IC を使用しています。

またデバイスドライバも同社提供のドライバを使用しています。

最新のドライバが必要な場合は FTDI 社のホームページから入手する事ができます。

使用しているチップは「FT232BM」で、それ用の VCP ドライバをダウンロードしてください。また、FTDI 社のホームページには Windows 用以外に MAC や Linux 用のドライバも掲載されています。

RB10-2U でも動作すると思われませんが、弊社では動作試験は行っていません。これらのドライバ (または Windows 以外の環境) に付いての質問にはお答えできませんのでご了承ください。

RB10-2U 付属 CD の Drivers フォルダに、二つの種類のドライバを収録したフォルダがあります。

CD から直接インストールできますが、RB10-2U を追加した場合、再度ドライバ CD を要求される場合があります。

トータルで数 M バイトに満たないファイルですので、ハードディスクに余裕がある場合、適当なフォルダにコピーしておくとう便利です。ドライバのインストールは、先に USB ポートに RB10-2U を接続し、認識した新しいハードウェアに対するドライバを組み込む手順になります。

インストール自体は USB を扱うドライバと仮想の COM ポートを作るドライバの 2 段階で行われます。

USB の接続に際して、RB10-2U の電源は、入っていても、切れていても、かまいません。付属 CD には、二つのタイプのドライバが収録されています。

それは、PnP に対応した物と、非対応の物です。

PnP 対応のドライバは WindowsXP で、非対応

は全ての Windows での利用を想定しています。共に、全ての Windows (95 は USB が無いので非対象) に対応しますが、次の様な現象が発生します。

FTDI 社の最新ドライバを基にしていますが、同社がドライバ仕様を変更した場合、上記の通りの反応にならない可能性があります)

## フォルダ名とドライバの種類

添付 CD の Driver フォルダ内に「PnP 対応」と「PnP 非対応」のフォルダがあります。

## ドライバの組み込み

PC と RB10-2U の USB コネクタを接続してください。

「新しいハードウェアが見つかりました」が表示されます。

続いて、ドライバの検索画面が表示されます。ドライバの場所に、CD またはコピーしたフォルダの名前を指定してください。

XP では「一覧または特定の場所からインストールする (詳細)」を選択しないと、場所の指定ができません。

正常にドライバが組み込まれると、続けて仮想 COM ポートのインストールが始まります。OS の種類により、自動的に最後まで組み込まれる場合と、再度インストール画面になる場合があります。

再度インストール画面になった場合は、先の例と同じく、ドライバーのフォルダの場所を指定する様にしてください。

## ドライバの削除

再度、最初から認識をやり直す場合や、ドライバが不要になった場合は、ドライバ CD かコピーしたドライバのフォルダ内にある「Ftdiunin(.exe)」を実行してください。

## COM ポート番号と複数枚の RB10-2U の使用について

### COM ポートの確認

インストールが完了した時点で (追加の場合も含めて) RB10-2U が接続された、COM ポートの番号を調べる必要があります。

スタートより 設定 コントロールパネルと選択して、開いたコントロールパネルフォルダ内のシステムアイコンをダブルクリックして起動してください。

システムプロパティのウインドウのタブからデバイスマネージャを選択してください。

(OS によっては、スタートより コントロール

パネル システム ハードウェアのタブ デバイスマネージャ)

デバイスマネージャが表示されたら、ポート (COM/LPT) の項目をダブルクリックして展開してください。

新たに USB Serial Port (COMx) の項目増えていれば、正常にインストールが完了しています。

COMx の x の場所には数字が入ります。この項目が COM ポートの番号を表していますので追加したボードに記入しておいてください。以後、対象の RB10-2U をコントロールする場

合に、この COM 番号を使用します。

### 多数枚の使用

RB10-2U は USB ハブを使用する事により、多数枚数を一つの PC に接続できます。

割り当てられる仮想 COM ポートの番号は、USB のコネクタ位置ではなく、RB10-2U ボード固有になります。

今 3 枚の RB10-2U を接続するとします。

仮に A,B,C と名前を付けます。

A を接続した際に COM6、さらに B を接続すると COM7 となったとします。

ここで B を外すと COM7 がデバイスの一覧から消えます。

そして C を挿入すると、割り当てられる COM 番号は消滅した COM7 ではなく、新たな番号、例えば COM8 になります。

もちろん B を挿入すると COM7 が現れます。すなわち、一度割り当てられた COM ポートの番号は、そのボード固有の値として記憶される事になります。

注意：割り当てられる COM ポートの番号は Windows が管理しています。

RB10-2U を異なる PC に接続した場合は、その機械固有の割り当てが行われます。その番号は前に接続していた PC とは関係ありません。同じ機械でも XP と Win98 のデュアルブート構成の様に起動する Windows が異なる場合、Windows 毎に異なる割り当てが行われます。

## 付属 CD の内容

フォルダ名 ... 内容

Drivers ... デバイスドライバ

Document ... PDF 形式による本マニュアルと RB10-1(参考用)のマニュアル。

\*PDF 形式の書類を開くには、Adobe 社の

Acrobat Reader (無料) が必要です。

<http://www.adobe.co.jp> より入手できます。

## 動作試験

(接続が完了し、デバイスドライバのインストールが完了している事、COM 番号が判明している必要があります)

図 1 の様に、USB の接続及び電源の接続を行ってください。

この時点では本基板上の緑色 LED ランプが点灯します。

Windows のハイパーターミナルを起動します。ターミナルモードを 9600bps、8bit、パリティ無し、RS・CS フローコントロール、ローカルエコー ON、行終端送信を CR に設定してください。本器は USB 接続ですが、最終段階のインターフェースがシリアルのため、下図の様なシリアルの通信条件を設定する必要があります。

(ハイパーターミナル以外のターミナルをご利用の場合で、仕様上、電話をかけないと接続できない様な場合は AT コマンドの項を参照してください)

Windows のバージョン及び、認識された COM 番号の数値に拠っては、ハイパーターミナルで設定できない COM 番号の可能性があります。

この場合は、ページ 15「3、Windows のハイパーターミナルで操作する場合」を参照してください。

次にターミナルから「AT」をタイプして RETURN (または行送信) を押してください。ディスプレイ上に「OK」が表示されれば正常です。

うまく行かない場合は、12 ページの「うまく動作しない場合は」とページ 15「3、Windows のハイパーターミナルで操作する場合」を参照してください。

## シリアルの設定条件

設定項目	設定値
伝送速度	9 6 0 0 b p s
ビット長	8 b i t
パリティ	無し ( N O N E )
フロー制御	R S ・ C S
エコーモード	ローカルエコー
行末処理	C R のみ送信

## 本結線

以下の作業は必ず、電源を切って、コネクタを外した状態で行ってください。

### 個別のI/Oとして使う

J1 から J15 のショートソケット(短絡ソケット)を全て I/O 側に挿入してください。

これにより、CN1 から CN10 の端子はそれぞれ独立したリレー接点になります。

ただし、ロボット用コネクタ CN11 にも一部の信号が接続されたままになります。必ず CN11 から、コネクタを外してご利用ください。

リレーは機械的な接点ですので、最大電圧、最

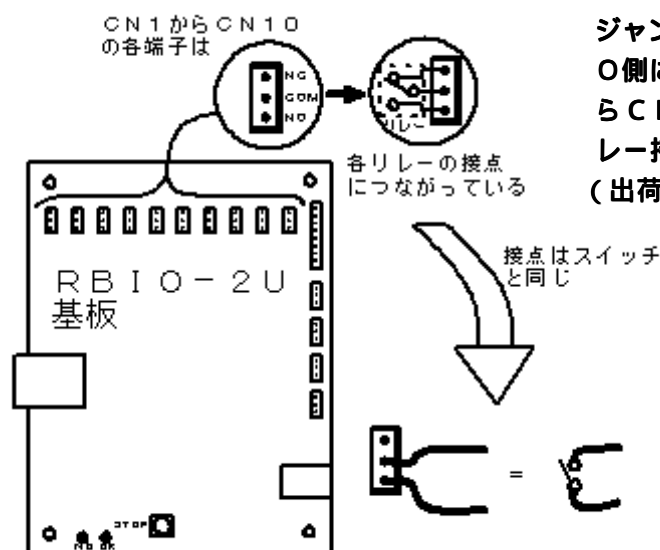
大電流の範囲内であればスイッチ代わりにそのまま使用できます(図2、次ページ図3)

ページ 20 の表 2 に端子番号を示しますが、COM が共通端子で、動作していない場合は NC 端子とつながっています。リレー動作時は COM、NC 間の導通が無くなり、COM、NO 間が導通します。リレー接点の耐電圧は AC100V になっていますが、基板耐圧の都合で AC30V または DC30V 程度を上限にしてください。

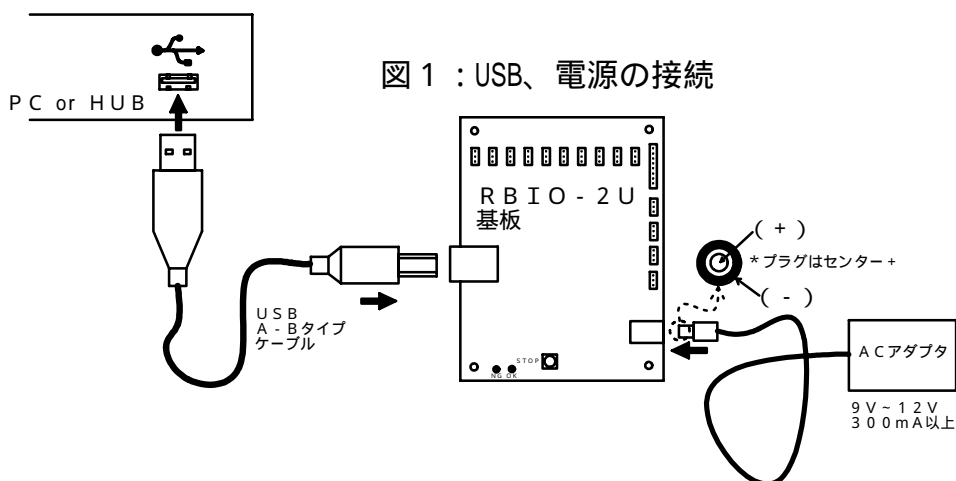
この電圧での最大許容電流は 1A までです。

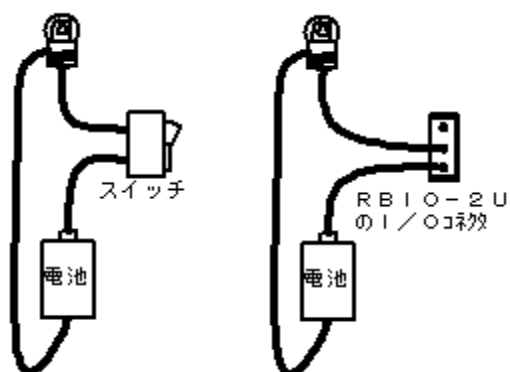
AC100V や高負荷の開閉には、巻末の「高容量リレーの追加」を参照してください。

コネクタの不足分は別途お求めください。適合コネクタはページ 20 の表 5 です。



ジャンパー J1 から J15 を I/O 側に設定した場合は、CN1 から CN10 の各端子は独立にリレー接点と接続されます。(出荷時状態はこの状態です)





リレー接点はスイッチの代わりに  
使用できます（最大定格以内  
で使用してください）

図 3：スイッチ結線

### ロボットを接続する

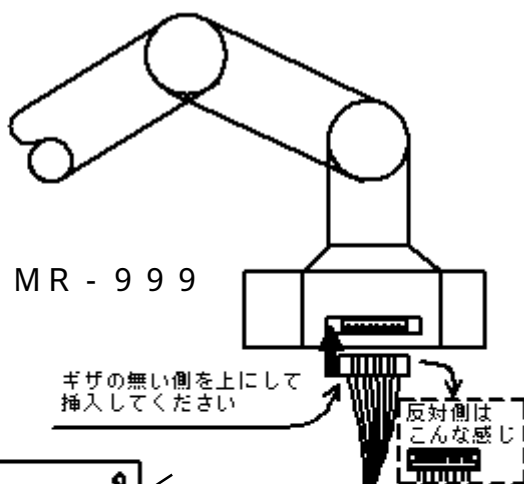
J1 から J15 のジャンパーピンを全て RBT 側  
に挿入してください。

CN11 とロボットアーム「MR-999」を付属の  
コネクタで接続します。

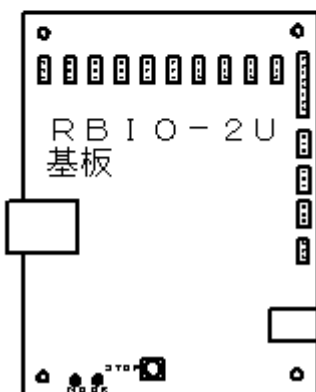
ただし、I/O 用コネクタ CN1 から CN10 にも  
一部の信号が接続されたままになります。  
必ずこれらの端子から、コネクタを外して  
ご利用ください。



### ロボットアームに接続

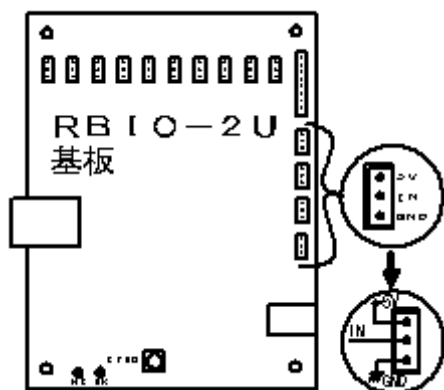


注意：CN1 ～ CN10 と CN11 は  
ジャンパー J1 ～ J15 の切り  
替え如何に関わらず、同時  
接続しないでください。  
CN1 ～ CN10 と CN11 **同時に**  
**接続**しますと回路上の衝突  
で接続機器やロボットを壊  
す場合があります。



## 入力端子

本器にはCN12～CN15の4個の入力端子があります。コネクタ番号はページ20の表3、端子番号は表4です。



スイッチの様な、接点信号を接続する場合はGNDとIN間に接続してください。

この接続では電源が不要なため、5V端子は解放とします。本器の入力（IN端子）は内部で10Kの抵抗で5Vに接続されています。

スイッチを「切」状態（解放と同じ）にすると、入力（IN端子）は

この抵抗により5Vとなります。またスイッチを「閉」（GNDとショートさせる）にすると、入力（IN端子）は0Vとなります。

FBコマンドにより、入力端子の状態問い合わせを行うと、結果として、

スイッチ開 入力 5V 255

スイッチ閉 0V 0

と「解放で255」、「ショートで0」が戻されますので注意してください。

また入力（IN端子）は直接、電圧を入力することができます。この場合の耐電圧は±24VでRS-232の様な信号を直接接続できます。論理の切り替わり電圧は+2.5V程度ですが、素子の固有値や温度で変化するため、なるべく0～5V以上の振幅を入力してください。

入力端子に出ている5Vはセンサ等の動作電源ですが、出力可能な電流が4個の端子の合計で**最大50mA**と大変少なくなっていますので注意してください。

## コントロールの方法

USBによる接続ですが、PC（コントロールプログラム）からは、COMポート（仮想）を通してシリアル通信によってRBIO-2Uを制御します。

デバイスドライバが仮想的にWindows内に作成したCOMポートを利用する訳ですが、本物のCOMポートを制御するのと同じ手法で制御可能です。

RBIOシリーズは、ASCIIキャラクタによる命令発行及び受信で動作します。

COMポートに対して、文字列をプリント（送信）する手法が取れば、駆動するソフトの種類は関係しません。

Windows付属のハイパーターミナルを使用して命令を送り出して動作確認が可能です。

## コントロールコマンド

注意1：一行に送信できる文字数は37文字までです。詳細は13ページ「1-2: シリアル受信バッファ容量」を参照してください。

注意2：一行の送信完了後、OKまたはERRORの返送文字を確認した後、次の行を送信するようにしてください。

注意3：ローカルエコーをONに設定してください。

本器のコマンドは文字列と終端記号で構成されます。

文字列は先頭に区分文字2文字とパラメータで構成され、終端記号として改行コード（CR: 0x0D）を送ります。多くのターミナルではこの改行コードはENTERキーを押す事で発生させる事ができます。説明は英大文字を使用していますが、文字自体をコードに対応させるPCDコマンド（下記参照）を除いて、英小文字でも同様に扱われます。

「PCR51」「pcr51」「pcR51」は全て同じ意味になります。

コマンドに対しては、RBIO-2Uから応答が返されます。応答は正常を示す「OK」とエラーを示す「ERROR」、結果要求を行うコマンドに対する応答の3種です。

なお、改行のみを入力した場合は何も返送されません。



本器をコントロールするコマンドは二系統に分かれます。

「PC」系：リレーを個別にコントロールします。I/O 端子を使用している場合に利用します。

「RC」系：ロボットアームをコントロールします。回転方向（正逆動作）と動作、停止を行います。

上記「PC」「RC」は論理上の区別で物理的な接続には関係しません。

ロボットを接続している場合でも「PC」でコントロールできますし、個別 I/O 接続に「RC」を使用してもかまいません。あくまで、使用しやすいコマンド体系で区分しているだけです。

### 個別コントロール用

は改行コードを示します。

「PC」引き続き「R」、「D」、「T」、「A」の一字とパラメータで構成されます。

「PC」の文字以降のコマンドは、「D」コマンドを除き、一行内に複数記述できます。「D」コマンドのみ、他のコマンドの一番最後が単独の行が必要です。一行内で同一リレーに対する操作を行った場合は最終の状態に従います。

例：PCR21R20 同一のリレー 2 に対して ON の指定と OFF の指定を行っています。この場合は最終の OFF 状態が結果になります。

例：PCR21XYR20 上の例と同じく、同一のリレー 2 に対して ON の指定と OFF の指定を行っていますが、途中でエラーになる文字 XY が存在します。この場合は最初の ON を実行した時点でエラーになるため、最終結果は ON になります。

#### コマンド：PCRnp

指定リレーを ON,OFF させるコマンドです。

n はリレー番号で 0 ~ 9、p は動作指定で 0 で OFF、1 で ON になります。

リレー番号に数字以外及び動作指定が 0,1 以外ではエラーになります。

一行での複数指定が可能です

例：0,1,3 を ON,2 を OFF にする場合

PCR01R11R31R20 順番に意味はありません。

任意の順に記述してください。

もし同一リレーに対する異なった指定がある場合は最後に出現した指定に従います。

PCR51R50 リレー 5 は OFF になる。

応答：OK 正常 ERROR 不正な文字がある。

#### コマンド：PCDst

リレーの直接 ON/OFF 指定

s と t はリレー番号 9 から 5 と 4 から 0 の ON,OFF を一文字で直接指定します。

指定には ASCII コードの知識があれば便利ですが、対応コード表を参照して実行してください。どちらかと言うと、直接キー入力に使用するより、機械が外部をコントロールするのに便利なコマンドです。

応答：OK 正常 ERROR 不正な文字がある。

s と t には任意の文字を記述できますが、ASCII コードの下 5bit がそのままリレーの ON/OFF に対応します。

#### コマンド：PCTnttt

時間指定でリレー ON

指定のリレーを、0.1 秒から 25.4 秒間で指定された時間 ON にするコマンドです。

n はリレー番号で 0 ~ 9、ttt は時間を指定します。

ttt の単位は 0.1 秒で 1 から 254 (25.4 秒)まで指定できます。

254 を越える 999 までの数値は最大の 254 と見なされます。

ttt の次が改行コード(行の終了)の場合は、ttt の桁数は数字として意味のある桁数、1,2,3 桁で可能ですが、4 桁ある場合はたとえ先頭が 0 でもエラーとなります。

例えば ttt が 5,05,005 これは全て OK で同じ 0.5 秒を表します。0005 はエラーとなります。

リレー 9 を 0.5 秒 ON させる操作、PCT95 PCT905

PCT9005 これらは OK PCT90005 は ×。

このコマンドも一行に複数記述可能ですが、複数記述の場合は ttt の部分が 3 桁の固定文字数でなければなりません(不足する場合は 0 を入れる)

例：リレー 3,4,5 を 1 秒 2 秒 3 秒間 ON させる

PCT3010T4020T5030

応答：OK 正常 ERROR 不正な文字がある。

## コマンド : PCAn

### 動作報告

指定リレーの動作状況報告を要求するコマンドです。

n はリレー番号で 0 ~ 9 です。

応答として、一つの要求に付き、動作時は '1' 停止時は '0' の文字が返送されます。

さらに改行後に OK または続きの文字列に対する ERROR が返されます。

例：リレー 0,1,3 が動作中 (ON) リレー 2 が停止中 (OFF) として

PCA0      結果は 1

PCA1      結果は 1

PCA2      結果は 0

PCA3      結果は 1

同一行に任意の順番で続けて指定する事もできます。結果は出現順に戻されます。

PCA0A1A2A3      結果は 1 行に 1101 が戻されます。

さらに、他のコマンドと同時に使用してもかまいませんが、操作を伴う場合は評価 (PCA コマンドの記述) 時点での結果が得られます。

例：リレー 3 に対して、リレー ON、状態報告、リレー OFF、状態報告を指示した場合。

PCR31A3R30A3      結果は 10 になります。一回目の報告要求の前に同リレーを ON するコマンドがあるため、一回目の報告時には ON になっています。また二回目の報告要求までに OFF の指示があるため、二回目の報告は OFF になります。

実際のリレー動作は最終結果の OFF となります。

最終まで正常実行された場合は、改行した後、OK の文字が、続きのコマンドがエラーになった場合は改行後 ERROR の文字が戻されます。

例：先の例と同じ動作で、

PCR31A3R30A3ABCDEF G と入力した場合は、PCR31A3R30A3 まで正常に実行されるため結果 10 が得られ、リレー 5 も OFF になりますが、次の ABCDEF G の文字列はエラーのため、この時点で中断され、次の行に ERROR が戻ってきます。

は改行コードを示します。

RC に引き続き RR, RL, SU, SD, AU, AD, NR, NL, FO, FC の文字とパラメータで構成されます。

PC コマンドと異なり、一行には、一つの指令しか記述できません。

結果は OK, ERROR, 0, +, - が返されます。

全ての RC コマンドは同一の形式で、RC に続けて 2 文字の指示とパラメータを記述します。例えば土台を右回転させるコマンドは RCRRppp となります。

ppp はパラメータで動作の指定で以下の様になります。

停止を行う場合は数字の 0 または省略。

連続回転を行う場合は文字 C

動作状態を調べる場合は ?

指定時間のタイマー動作を行う場合は 0.1 秒を単位とする、1 ~ 254 の数字を指定します。255 ~ 999 の数字でもエラーにならず、最大の 254 と解釈されます。

注意: 数字の指定は最大 3 桁です。4, 04, 004 は全て同じ値になりますが、0004 はエラーになります。

? に対して下記の応答があります。

数字の 0 停止中、記号 + 正方向動作、記号 - 逆方向動作。

なお、? と停止 (時間指定を 0 にする) コマンドは、同じ軸の動作で 2 種類のコマンドがありますが、結果は同じになります。(RCRR? と RCRL?, RCRR0 と RCRL0 等)

正方向と逆方向の定義ですが下のコマンド表の ( ) の中に (+) と (-) で示します。+- の基準はロボットアームのモータの回転方向 (モータに送る電圧の極性) を基準にしています。

また右、左はロボットアームから見ての方向を示します。

RC に続く二文字は以下の通りです。RC も含めて示します。

RCRR 土台を右回転させます (-)

RCRL 土台を左回転させます (+)

RCSU 肘を上に出します (-)

RCSD 肘を下に出します (+)

RCAU 腕を上に出します (+)

RCAD 腕を下に出します (-)

RCNR 手首を右に回転させます (+)

## ロボットコントロール用

RCNL 手首を左に回転させます (-)

RCFO 指を開きます (-)

RCFC 指を閉じます (+)

例:

手首を連続で右回転 RCNR

土台を 10 秒間左回転 RCRL100

腕の上げ下げを停止します RCAU0 または  
RCAD0 (RCAU または RCAD でも停止し  
ます)

土台の回転を調べる RCRR? または RCRL?

## サーチコマンド

基板上の 4 個の入力状態を調べるコマンドです  
FB に続き入力チャンネル番号 0 ~ 3 を指定  
します。

PC コマンドと異なり、一行には、一つの指令し  
か記述できません。

結果は 0 か 255 の数字で返されます。

入力端子と GND 間を短絡 (電圧 = 0V) の場合は  
0 に、入力端子を解放または 5V に接続 (電圧 =  
5V) の場合は 255 が戻されます。

例: ch2 の状態を調べる FB2

## 実時間入力

RB10-2U ではサーチコマンド発行による、  
ポーリング形式に加え、端子の状態変化が  
あった時点で送信を行う、リアルタイム形式  
が利用可能です。

変化時に送信される内容は、先頭から I の文  
字 (0 x 4 9 ) 入力番号 '0'..'3'、結果  
5V='H'・0V='L'、CR コード、LF コード  
ただし、入力番号の 0 ~ 3 は ASCII コードの  
数字 0x30 ~ 0x33 です。

入力の変化は秒 50 回の割りで検査されます。  
変化検出で送信すべき状態になった際、RTS  
線で送信が禁止されている場合は、その時点  
の送信データは破棄されます。

リアルタイム形式を使用するには、特殊モー  
ドで S6 パラメータを 1 に設定してください。  
ポーリングに比べて、実時間応答が得られる点  
と、余分な操作を必要としない点で便利ですが、  
RB10-2U からの送信タイミングを、受け手  
でコントロールできません。

(変化があれば、送信されて来ます)

## AT コマンド

AT コマンドはモデムのコントロールに使用する  
汎用的なコマンド体系です。

本基板をターミナルに接続した場合、ターミナ  
ルの仕様で、AT コマンドが送られてくる場合が  
あります。これに対処するため、ATD を除く、AT  
で始まる文字列に対して OK を返送します。

また ATD は電話をかけるための AT コマンドで、  
一部のターミナルでは電話をかけないと先に進  
まない仕様となっています。これに対処するた  
め、ATD で開始される文字列が送られて来た場  
合には、約 1 秒後に「CONNECT 9600」の文字列  
を返送する様になっています。ターミナルは、  
この文字列により、電話先が応答した状態と認  
識します。

なお ATD 以降は何を書いても無視されますの  
で、適当な電話番号を設定しておけば接続でき  
ます。

## 緊急停止

基板の STOP ボタンを押してください。全ての  
リレーは OFF になり、ロボットアームは停止し  
ます。なお停止と同時に、ターミナル上に  
「RB10-2 I/O Control Ver.x.x」(x.x はバージ  
ョン番号) を表示します。

後述の電源コントロール接続を行っている場合  
も一度出力が OFF になり、その後入力 0 の状態  
に従いますが、この時点でシリアル線の送信が許  
可されていない場合は、「RB10-2...」のメッ  
セージを送信しようとして停止してしまいま  
す。

## コマンドステータス

コマンドの実行結果を示す赤と緑の LED ランプ  
が基板に実装されています。

赤色 LED ランプが点灯 <- コマンドエラーの場  
合 (ターミナルに ERROR 文字が返送される)  
緑色 LED ランプが点灯 <- 正常実行された場合  
(ターミナルに結果が OK 文字が返送される)  
表示は次のアクションがあるまで、変化しま  
せん。

## リレーステータス

I/O コネクタとリレーの間にある赤色の LED ラ  
ンプはリレー動作のモニタ用です。

リレーがONになっていると点灯します。

## Windowsでコントロールする場合

予め、ドライバの組み込みが正常に行われている必要があります。

動作させる方法に、COMポートを直接制御する方法と、モデムとして制御する方法があります。

初めてRBIO-2Uをモデムとして使用の場合は、コントロールパネルのモデムと電話内のモデムタブから「追加」を行ってください。

自動では認識できませんので、モデム一覧から「標準 9600 bps モデム」を選んでください。

**ターミナルソフト**を立ち上げます。

Windowsにはハイパーターミナルが付属しています。スタートメニュー、プログラム、アクセサリとたどって、ハイパーターミナルを起動（フォルダが開く）します。ハイパーターミナルを格納しているフォルダが見あたらない場合はインストールされていないので、アプリケーションの追加と削除からインストールしてください。ハイパーターミナルは通信のサブカテゴリにあります。

初めて起動させる場合は、ハイパーターミナルのホルダからHyperterm(.exe)を起動させます。接続名を入れる所に適当な名前を付けます「RBIO」等、アイコンの選択もできます。「OK」で電話番号のダイアログが表示されますので「電話番号」に適当な数字「0000」等を入れます。「接続方法」から、先ほど選択した、標準 9600 bps モデム、選択します。

「OK」で「接続」のダイアログが表示されますので、「ダイヤル」をクリックします。

やがて接続が完了してターミナル画面になります。「接続」のダイアログが何時までも表示されている場合は接続ミスが考えられます。次の項目を参照して対処してください。

うまくターミナルが起動できた場合は、「ファイル」メニューから「プロパティ」を選択し、「設定」のタブから「ASCII 設定」を選択し、「ローカル」にチェックを入れて、閉じてください。

ターミナル画面でAT又はatをタイプして、画面に表示される事と、ENTERを押す事でOKが表示される事を確認してください。

キーを押した瞬間にカーソルが消滅して、以後

のキー入力を受け付けなくなった場合は、接続ミスです。次の項目を参照して対処してください。

ハイパーターミナルを終了する際に、保存を選択しておけば、指定した名前のターミナルが作成されます。次回からは、専用のターミナルを起動すれば設定の必要がなくなります。

## うまく動作しない場合は

電源は正常ですか

電源には、**安定化**された9V～12Vを必要とします。またコネクタのセンターは+になっています。通常市販されている、ラジカセ等のACアダプタは9Vで300mA以上の表示があれば、動作する可能性もありますが、非安定アダプタとなっているため、お勧めできません。

**緑色LEDランプが点灯**しますか

電源が正常に供給されると基板上の緑色LEDが点灯します。

USB機器として認識され、正常にドライバがインストールされていますか。

デバイスマネージャを表示させた際に、黄色の？マークが表示された項目が無いでしょうか。

？マークが、RBIO-2Uの抜き差しに合わせて、出現する場合は、ドライバの組み込みミスです。？マークの項を削除した後、RBIO-2Uの抜き差しを行ってください。新しい機器として、ドライバの組み込み画面が表示されます。

ターミナル設定は大丈夫でしょうか

本器は基板から入力文字に対するエコーバックを返しません。ターミナル設定では必ずローカルエコーの項目にチェックを入れてください。また通信条件である9600bps、データ8bit長、パリティ無し、ハードウェアフロー制御(RS,CS制御、RTS,CTS制御と記載されている場合もある)に設定してください。

接続、ターミナルソフトに問題点が見あたらない場合は、基板のSTOPボタンを押してみてください。ターミナル上に「RBIO-2 I/O Control Verx.x」と表示されれば、接続は正常と見なせ

ます。

この場合は一方向だけですが、基板 -> PC で通信できた事になります。

ここで AT とリターンキーを押して OK が表示されれば双方向とも通信が正常と見なせます。

=====

これ以降の項目は、通常の使用では必ずしも必要としません。

うまく動作しない場合や、コントロールソフトの制作が必要な場合に参照してください。

## 技術資料

### 1、弊社 RB10-1 との相違点

#### 1-1: 通信インターフェース

基本的に RB10-1 用に製作した、シリアル経由の通信制御プログラムは、RB10-2U においてもそのまま動作する可能性があります。

さらに、パフォーマンスを上げる場合は、以下の相違点を考慮する必要があります。

RB10-2U では、PC から送り出され、USB デバイスから伝送されて来たキャラクタは最終段階で、基板上のコントローラがシリアル信号として受信します。

このシリアル信号を受信する事においては、RB10-1 と RB10-2U は同一の方式と言えます。

しかし、RB10-1 用のコントローラではこの受信動作を、ソフト的に行っていました。

一方、RB10-2U においては、シリアル信号をコントローラ内蔵のハードウェアが受信します。

ソフトで行うシリアル通信は、通信線が全二重用であっても、送信か、受信の一方通行（半二重通信）となっており、通信方向の調停に RS-CS（RTS-CTS）ハードフローコントロールを利用していました。

現象的には、RB10 に対してコマンド + 終端コードを送信した場合、一定時間、必ず RB10 がビジーとなります。

CTS 線はディスエーブルされますので、続けてコマンドを送る場合でも、一定時間の送信禁止期間がありました。

一方、RB10-2U においては、ハードウェアが通信を処理するため、必ずしも、CTS をディスエーブルする必要がなくなりました。

以前の RB10-1 との互換性のため、出荷状態の RB10-2U は、受信完了後に CTS をディスエーブルとしていますが、特殊モードを使用して、S1 パラメータを書き換える事により、変更できます。

これにより、完全な全二重通信が可能となります。バッファの容量内なら、RB10-2U からの応答を待たずに、次のコマンドを送信する事ができます。

#### 1-2: シリアル受信バッファ容量

RB10-1 の受信バッファ容量は 38 バイトでしたが、RB10-2U ではビジー状態でも受信可能なバッファを 64 バイト余分に持っています。これは、全二重通信に設定した場合に有効です。

この場合でも一組分（先頭から終端文字まで）の最大長さは、RB10-1 と同じ 38 バイトです。

#### 1-3: RB10-2U が USB インタフェースを搭載した事で発生する現象

RB10-2U 上には USB インタフェースと制御用 CPU が、別々の系統として実装されています。これらの系統は電源も別になっており、USB インタフェースは USB のバスパワーで、RB10 のコントロール用 CPU は、基板上の DC ジャックから供給される様になっています。

各々の電源は同時に投入される事がないため、次の様な現象が発生します。

1-3-1: USB が先に PC と接続され、後から基板の DC ジャックに電源を供給した場合。

既に、PC の通信ソフトが起動している場合、PC 上にゴミデータが受信される場合があります。

特に基板の DC ジャックに電源を供給した瞬間と切断した瞬間に多く発生します。

基板上の電源供給状態は DSR ステータスで知ることができる様になっていますので、適時ソフトで処理してください。

1-3-2: 基板上の DC ジャックに電源を供給した後、USB を接続した場合。

先に RB10 のコントローラが受信動作を開始するため、RB10 の受信バッファにゴミデータが受信されています。

USB と接続後にそのままコマンドを送信しても、先に受信したゴミデータの続きに本来の

コマンドが受信されるため、最初のコマンドがエラーとなります。

PC 側のソフトが動作を開始した時点で、一回だけ、ダミーのコマンド（エラーになるが）を送るか、CTRL-X キャラクタ（0x18 コード）を送信して、バッファに溜まったゴミデータをクリアしてから、本来の操作を実行してください。

#### 1-4: 通信条件

RBIO-1 と同じ条件です。

以下の設定を行ってください。

伝送速度 9600bps

ビット長 8bit

パリティ 無し(NONE)

フロー制御 RS・CS

行末処理 CR のみ送信

なお、マニュアルにあるエコーモードとは、ターミナル等を使って、人間が直接操作する場合の、キー入力操作に付いての項目です。通常のターミナルでは、キー操作は、相手の機械に送信されるだけで、画面上には表示されません。この状態をセンターエコーと呼びます。ターミナルは、相手に送ったキャラクタがそのまま送り返される事を期待して動作しています。

しかし、RBIO では受信したキャラクタを送り返す操作を行わないため、キー入力は画面に表示されません。

そこで、ターミナル設定をローカルエコーに変更します。

ローカルエコーでは、キー入力した文字が相手に送信されると共に、自身の画面にも出力されます。

#### 2: RBIO-1 と -2U を同じソフトでコントロールする場合。

ターミナルでは無く、VB の様な言語を使って制御する場合に RBIO-1 を接続する COM ポートと RBIO-2U を接続する仮想 COM ポートは、通常の使用においては、番号の違い位しか相違はありません。

しかしながら、次の 2 点を考慮する必要があります。

#### 2-1: タイミング

RBIO-1 を PC に装備されているハード的な COM ポートに接続した場合に比べ、USB 接続の方

が、データの遅延が多くなる傾向にあります。Windows では、シリアルキャラクター送受信と言っても、1 キャラク毎にシリアル用のデバイスとやり取りする訳ではなく、ある程度のサイズになるまで、文字を溜め込んだ後、まとめてデバイスとやり取りを行っています。実際にコマンド（+ 終端キャラクタ）を送信した後、結果を受け取るまでの時間は、直接 COM ポートに接続した場合と、USB 経由では異なってきます。

送信から結果の受信までの時間をぎりぎりまで詰めている場合、USB デバイスを経由すると、タイムアウトと判断される事があります。USB、直結両方に対応するソフトを製作する場合、タイムアウトの処理は少し長い目で行ってください。

#### 2-2: 起動直後のゴミデータ

先の 1 項で説明した様に、RBIO-2U では、バッファ内に不要なデータが受信されてしまう事があります。RBIO-1 では、論理レベルの違いで、ゴミデータを受信する事が殆どないため、特別な配慮は必要ありませんでした。RBIO-1、-2U 双方に対応したソフトを製作する場合、次の様な操作を行ってください。

- ・ 1、シリアルデバイスを OPEN し、DTR 線を FALSE に設定する。

- ・ 2、DSR 線のステータスをモニタし TRUE の場合に受信したデータは破棄する。

- ・ 3、DSR 線のステータスが FALSE であれば、次の初期化を行う。

CTRL-X キャラクタ（0x18）を送信、BS（0x08）を 3 回送信。

- ・ 4、通常の操作。

RBIO-2U の電源状態を示す、DSR ステータスは、ハード上の制限から一般的な論理とは逆になっています。

電源 ON : DSR=FALSE      電源 OFF : DSR=TRUE

項目・1 は、RBIO-1 で接続状態時に DSR ステータスが FALSE になる様にするための処置です。

RBIO-1 の DSR 線はそのまま DTR 線に繋がっているため、DTR=FALSE にすると、DSR 線は FALSE になります。

項目 2 は RBIO-2U の電源 ON までの、ゴミ処理です。

項目3は、RB10-2上の受信バッファークリアする手順です。

RB10-2UはCTRL-Xキャラクタの受信で、バッファを空にしますが、CTRL-Xを解釈しないRB10-1にとっては、CTRL-Xが逆にゴミデータとなってしまいます。

そこで、BSコード(RB10-1、-2Uともある)を数回送って、CTRL-Xキャラクタを削除しています。

### 3、Windowsのハイパーターミナルで操作する場合

RB10をターミナルから操作する場合、デバイスの扱いで二種類の方法があります。

一つは直接COMポート番号を指定する方法、もう一つは、RB10をモデムとして扱い、接続先をモデムとして指定する方法です。

Windows98/MEのハイパーターミナルは、直接接続できるCOMポートが、COM1～COM4となっています。

新たに認識されたUSB Serial Portの番号がこの範囲にある場合や、COM255まで扱えるターミナルソフトの場合は、接続方法の選択で、COM?ヘダイレクトを選んでください。

新たに認識されたUSB Serial PortのCOMポート番号がCOM1～COM4以外の場合は、モデムとして設定する事により、Windows98/MEのハイパーターミナルでも操作可能です。

(COM1～COM4に入っている場合、または直接COM4以上を扱うターミナルを利用する場合でも、モデムとして扱う手法は有効です。本説明書の「Windowsでコントロールする場合」の項目では、最初からモデムとして扱う手順で説明しています)

モデムとして、扱う場合は、先にモデムの追加を自動認識で行って見てください。

新しいモデムとして、自動で認識される場合があります。

自動で認識できた場合は「標準モデム」の名前になります。

自動で見つからない場合は、「モデムの一覧から選択」を選んで、標準モデムの種類から「標準 9600 bps モデム」を選択してください。

ハイパーターミナルの設定では、接続方法の項目に、認識したモデム名「標準モデム」ま

たは「標準 9600 bps モデム」を入れて下さい。

COMポートと直接接続している場合は、操作開始の注意点として、操作開始時にENTERキーを押して、RB10-2U上のゴミデータを取り除いてください。

その際ERRORが表示される場合もありますが、次の指令から正常に動作します。

"PC"とENTERを押してOKが表示されれば、正常動作中です。

一方、モデムとして組み込んでいる場合は、上記の初期化手順を行う前に接続(電話をかける)手順が入ります。一回目の電話でエラーになった場合は、再度「ダイヤル」をクリックして、電話をかけてください。

### 特殊モード

特殊モードは本基板の動作を変更するためのモードで、設定によっては希望の動作にならない場合がありますので、注意して利用してください。

なお、基本動作状態以外に設定を変更した場合の動作は保証対象外とさせていただきます。

基板の動作モードはコントロール用IC、PIC16F628のEEPROMに記録されています。

これを便宜上S0からSFと呼ぶ事にします。S0からSFはそのままPIC16F628のEEPROM、0～9、A～F番地に対応しています。また内容はASCIIコード'0'～'9','A'～'F'に対して、バイナリの数値00～0Fを使用しています。

この内容を書き換える事で基板の動作を変更します。書き換えはターミナルから送る文字列と基板のSTOPボタンを使います。

ターミナルから文字を送る必要がありますので、最低限の接続が完了していないと利用できません。

#### 特殊モードに入るには

基板上のSTOPボタンを押したまま電源を入れます。

赤と緑の両方のLEDランプが点灯します。ここでSTOPボタンを離します。

#### 特殊モードの動作

特殊モード中はシリアル通信の内、基板に向かう方向の信号しか使用しません。

基板から信号を送り返す事はありません。  
書き換えには次の文字列「Sn=p」を使用します。  
Sは英大文字で小文字のsはエラーになります。  
次のnには'0' ~ '9' 及び 'A' ~ 'F' の英数字 (HEX 数) が入ります。

= は英文字の = です。次のpにはパラメータとして'0' ~ '9' 及び 'A' ~ 'F' の英数字が入ります。

終端の改行(enter)は不要ですが、入力しても無視します。

例えばS3の内容を2にするには「S3=2」とタイプする事になります。

正常に入力した場合は基板上のLEDが緑点灯のみとなり、赤色は消えます。

ここで、二つの選択が可能です。

1:書き込みを実行する場合 基板上のSTOPボタンを押します。一瞬LEDが消えてすぐに赤、緑の両方が点灯します。これで該当パラメータが書き換えられています。続けて作業が可能です。

2:中止する場合 ターミナルからさらに文字を送ります(スペース等が適当)これで、パラメータ入力がエラーとなりますので、次のエラーの場合に進んでください。

エラーの場合

入力の文字列は上記以外では全てエラーになります。すなわち先頭がSではない、次の文字が数字ではない、さらに=が続かない、パラメータが数字ではない、5文字以上入力した。

以上の場合、エラーが判明した時点で赤色LEDが点灯し、緑色LEDが消灯します。

エラー状態は基板のSTOPボタンを押す事でクリアされ、最初の赤、緑、両LEDの点灯に戻りますので作業を続ける事ができます。

### 特殊モードの終了

電源を切断してください。

### パラメータと動作内容

S0 STOPボタン押した場合の文字列送出。

S0=0 STOPを押すと RB10-2 I/O Control....  
の文字を返送 (初期値)

S0=1 STOPを押しても文字列を返送しない

S1 終端文字列の扱い

S1=0 コマンドの終端はCRとLFの両方を認識する (初期値)

S1=1 コマンドの終端をLFのみとする

S1=2 コマンドの終端はCRのみとする

S1=3 使用禁止 (終端が認識できなくなる)

S1=4 コマンドの終端はCRとLFとし、受信後0.5mS待つてRTSをOFF

S1=5 コマンドの終端をLFのみとし、受信後0.5mS待つてRTSをOFF

S1=6 コマンドの終端はCRのみとし、受信後0.5mS待つてRTSをOFF

S1=7 使用禁止 (終端が認識できなくなる)

S1=8 全二重通信でコマンドの終端はCRとLFの両方を認識する

S1=9 全二重通信でコマンドの終端をLFのみとする

S1=A 全二重通信でコマンドの終端はCRのみとする

S1=B 使用禁止 (終端が認識できなくなる)

S1=C 全二重通信でコマンドの終端はCRとLFとし、受信後0.5mS待つてRTSをOFF

S1=D 全二重通信でコマンドの終端をLFのみとし、受信後0.5mS待つてRTSをOFF

S1=E 全二重通信でコマンドの終端はCRのみとし、受信後0.5mS待つてRTSをOFF

S1=F 使用禁止 (終端が認識できなくなる)

### S2 電源コントロール接続指定

S2=0,2 通常の動作

S2=4,6 通常の動作 (リレー0のタイマーだけ1.6秒単位)

S2=1 ch0入力とリレー0の出力をレベルモードでリンク 0.1秒単位のタイマー

S2=3 ch0入力とリレー0の出力をエッジモードでリンク 0.1秒単位のタイマー

S2=5 ch0入力とリレー0の出力をレベルモードでリンク 1.6秒単位のタイマー

S2=7 ch0入力とリレー0の出力をエッジモードでリンク 1.6秒単位のタイマー

S3 エラー時のデバッグ情報



S3=0 エラー時は通常動作

S3=1 エラー時にデバッグ情報を返送

#### S4 返送指定

S4=0 通常動作

S4=1 サイレントモード

#### S5 電源コントロールのタイマー値指定

S5=0: 30 秒                      S5=5: 4 分

S5=1: 1 分                      S5=6: 5 分

S5=2: 1 分 30 秒               S5=7: 6 分

S5=3: 2 分                      S5=8: 6 分 46 秒 (最大値)

S5=4: 3 分                      S5=9: 連続

#### S6 入力形式

S6=0: 通常状態

S6=1: リアルタイム入力

S7 ~ SF 現在未使用

#### 使用例

S0 パラメータ: 本基板では、STOP ボタンはいつでも入力可能です。予定外の文字列が返送されてターミナルが混乱する様な場合には S0=1 にして利用してください。

S1 パラメータ: 一部のターミナルでは処理の関係で RTS、CTS のコントロールが本基板とタイミミングが一致しない場合があります。文字列送信直後にデッドロック (要するに先に進まない) となる様な場合は S1=6 または他のモードをためてみてください。また通常、行終端のコードは改行と表現されていますが、復帰 (CR コード: 0x0d) コードの事を指します。まれに本来の改行 (LF コード: 0x0a) を出力するターミナルがありますので、その様な場合には、変更してみてください。

S2 パラメータ: PC 電源を外部からコントロールする場合に使用するモードです。

\* このモードは RBIO-1 との互換性のために存在します。

Linux においても USB が認識できれば使用可能と思われますが、実験は行っていません。実験される方は自己責任でお願いします。接続等の情報は、CD に収録されている参考用の

RBIO-1 取説を参照してください。

この機能により、外部から電話等の手段で PC の電源を入れて、終われば終了する Linux サーバー等の省エネ化ができます。

電源を入れるには、外部からの電話着信を検出する必要があります。これには、RAS の受付用を使用するモデムの CI 信号が利用できます。この信号を分岐して、本基板の入力 ch0 に接続する事で、着信時にリレー 0 を ON にする事ができます。もちろん本基板は該当 PC のシリアル端子に接続しておく必要があります。RAS サービス中は定期的に本器のリレーに対し時間付きの ON 指令を送信しておけば、シャットダウン後に一定時間 (タイマー時間) 経過後に電源を切る事ができます。

入力の ch0 とリレー 0 除く部分はそのまま I/O として使用できます。また入力 ch0 に対する FB コマンドとリレー 0 に対する全てのコマンドも使用する事ができます。

モデムの CI 信号ではなく、専用のベル検出器が必要な場合はワンダーキット製「BEL-22: 組立キット」が利用できます。

詳しい接続は巻末を参照してください。

S2=1 のレベルモードでリンクと S2=3 のエッジモードでリンクの違いですが、ch0 に入力する信号の取り扱いの違いになります。レベルモードは持続中全般に意味を持ちます。一方エッジモードでは信号が発生した瞬間に意味があり、以後の状態は関係しません。通常電源コントロールはレベルモードが適しています。

信号発生 -> PC 起動 -> 処理 -> 電源断にかかる時間より、ch0 に入力する信号の有効時間の方が長くなる様な場合は、レベルでは電源を切れないため、エッジとする事で入力信号の持続時間と無関係にする事ができます。

\* 信号確認時間に付いて

入力 ch0 に入れる信号の持続時間が 20ms より短いと、有効な入力として判断できない場合があります。モデムの CI 線の様なゆっくりした信号を入力する場合は、特に問題ありません。

#### S3: エラー時のデバッグ

指定した通りにコマンドを送っているのに ER-ROR が戻される様な場合は S3=1 に設定してください。これにより ERROR の場合は、デバッグ用

情報を付けて返送される様になります。  
返送される情報は ERROR に続いて / エラーと認識した文字までの文字数 / 受信した文字列 / 文字 1 の ASCII コード / 文字 2 の ASCII コード / .... と終了まで続きます。  
エラーと認識した文字までの文字数と ASCII コードは共に 2 桁の 16 進数で表されます。  
PCR01NR11 と送信した場合  
ERROR/06/PCR01NR11/50/43/52/30/31/4E/52/31/31  
と返送されます。ERROR/ の次が 06 なので ( 16 進数に注意 ) 前から 6 文字目、N の文字でエラーが発生した事が判ります。また後ろの ASCII コードで、非表示文字が紛れ込んでいるのを発見する事ができます。  
受信文字列がそのまま返された部分の文字が間違っている場合は、伝送エラーなので、ケーブルやコネクタを点検してください。

#### S4: 返送抑止

S1 ともからみますが、RTS, CTS タイミングがどうしても一致しない場合や、本器からの返送がそもそも必要としない様な場合に利用します。  
S1=1 とする事で、全ての返送メッセージは抑止されます。  
この場合、FB コマンドも例外ではなく、リレーの動作状況のチェックや入力チェックも返送されなくなります。

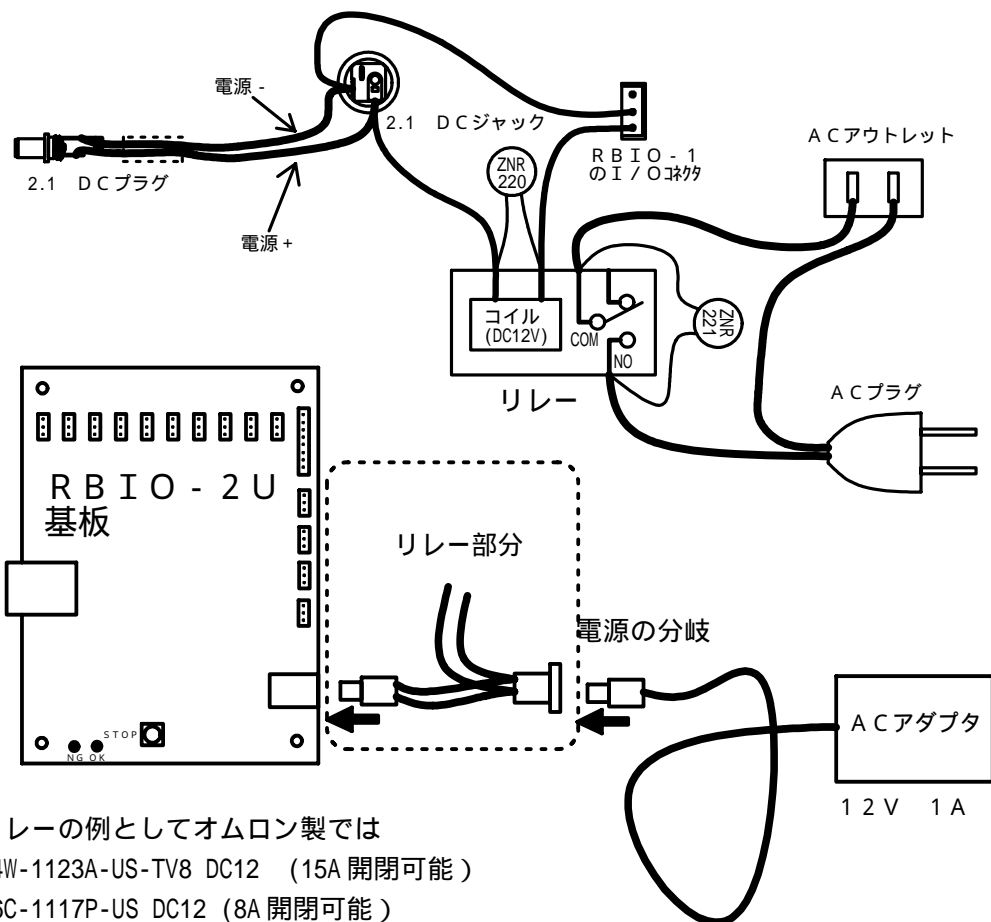
#### S5 電源コントロールのタイマー値指定

S2 で電源コントロールを行っている場合に入力 0 でセットされるリレー 0 タイマーの値です。  
なお、タイマー値は S2 でのモードが 1.6 秒単位の場合の値です。0.1 秒モードでは 1/16 ( 16 倍速くなる ) になります。

#### S6 入力の形式

1 に設定する事により、リアルタイム形式の入力チェックが行えます。  
詳しくは、「実時間入力」の項を参照してください。

## 高容量リレーを追加する



リレーの例としてオムロン製では  
 G4W-1123A-US-TV8 DC12 (15A 開閉可能)  
 G6C-1117P-US DC12 (8A 開閉可能)

AC100Vの開閉や高電流の開閉が必要となった場合は、本器のリレー出力（接点）を使って、さらに容量の大きなリレーを駆動します。上記の回路例ではRBIO-2Uの電源が12Vであることから、リレーのコイル電圧として12Vを選択して、電源を共通化しています。

リレーコイル用の電源を別途用意する場合のコイル電圧は12Vである必要はありません。一般にリレーのコイル電圧は5V、12V、24Vが入手し易くなっています。

例では電源を共通としたので、ACアダプタの12Vを分岐するため、一度DCジャックで受け、RBIO-2Uの電源用DCプラグとリレー動作を取り出します。リレーのコイルに、RBIO-2Uの出力が動作した際に閉（COMとNO間）となる接点を通してDC12Vに接続します。

これにより、RBIO-2Uの出力が、さらに高容量のリレーを駆動できるようになります。

回路例には「ZNR」と記入された円盤状の素子が接続されていますが、これはバリスタと呼ばれる過電圧を吸収する素子です。メーカーにより、呼び名が異なりますが、ZNRやTNRと呼ばれる様です。リレーのコイルと並列に22Vタイプ（記号は220と記載されている）とAC100Vを開閉しているリレーの接点に220V（記号は221と記載されている）をそれぞれ接続しています。

この様な過電圧吸収用素子を実装しないと、接点开閉に伴って発生するノイズで本器搭載のコントローラが暴走してしまいます。

過電圧を吸収できれば、バリスタ以外でもかまいませんが、必ず何らかの処置は行ってください。

## 付録

**表 1 : リレー番号と出力コネクタの対応表**

リレー番号	コネクタ番号
0	C N 1
1	C N 2
2	C N 4
3	C N 3
4	C N 5
5	C N 6
6	C N 8
7	C N 7
8	C N 9
9	C N 1 0

P C コマンドで指定するリレー番号と出力コネクタの対応表です。  
コネクタ番号が、不規則ですが、基板上では順に並んでいます。

**表 2 : 端子番号 (出力用 CN1 ~ CN10)**

端子番号	信号名
1	N C
2	C O M
3	N O

**表 4 : 端子番号 (入力用 CN12 ~ CN15)**

端子番号	信号名
1	G N D
2	入力
3	5 V

**表 3 : 入力番号とコネクタ番号の対応**

入力番号	コネクタ番号
0	C N 1 5
1	C N 1 4
2	C N 1 3
3	C N 1 2

**表 5 : 適合コネクタ表**

コネクタ番号	適合コネクタ
C N 1 ~ C N 1 0 C N 1 2 ~ C N 1 5	E H R - 3 日本圧着端子製
C N 1 1	E H R - 8 日本圧着端子製

**\* 上記コネクタはハウジングのみです。  
別途適合するピンをお求めください**

## コマンド一覧表

コマンド	動作
**個別コントロール	
PC R n p	指定リレーのON/OFF n = リレー番号 p = 0 : OFF p = 1 : ON
PC D s t	リレーのON/OFFを直接指定 s = リレー 0 ~ 4 の指定 p = リレー 5 ~ 9 の指定 指定は 4 0 H を起点とするASCIIコード
PC T n t t t	指定リレーを指定時間ONさせる n = リレー番号 t t t = 3桁の時間指定 時間は 0 . 1 秒単位で 0 ~ 2 5 4 が指定可能
PC A n	指定リレーの動作報告要求 n = リレー番号 動作時 1 停止時 0 が返送される
**ロボットコントロール コマンドの後ろに 0、C、? または p p p 0 = 停止 C = 連続回転 ? = 動作報告 p p p = 時間指定付き回転	
RC R R	土台を右回転
RC R L	土台を左回
RC S U	肘を上上げる
RC S D	肘を下に下げる
RC A U	腕を上上げる
RC A D	腕を下に下げる
RC N R	手首を右回転
RC N L	手首を左回転
RC F O	指を開く
RC F C	指を閉じ
**サーチコマンド	
FB n	指定入力の検査報告要求 n = 0 ~ 3 : 入力番号
**ATコマンド	
AT string	ATで開始されるコマンドに対してOKを返送 stringは任意の文字(文字列)
AT D string	ATDで開始するコマンドに対してCONNECT 9600を 返送する。stringは任意の文字(文字列)

\* 具体的な動きは本文を参照してください。

PCD コマンドで例えば「PCD00」と送った場合は全  
てのリレーがOFFになります。

「PCDUC」と送ると 9 8 7 6 5 4 3 2 1 0

上段リレー番号、下段の - がON、- がOFF

## PCDコマンド用対応コード表 (PCD文字1文字2の順に送る) がONを表します

文字1のリレー対応

文 字	リレー番号				
	9	8	7	6	5
@					
A					
B					
C					
D					
E					
F					
G					
H					
I					
J					
K					
L					
M					
N					
O					
P					
Q					
R					
S					
T					
U					
V					
W					
X					
Y					
Z					
[					
¥					
]					
^					
_					

文字2のリレー対応

文 字	リレー番号				
	4	3	2	1	0
@					
A					
B					
C					
D					
E					
F					
G					
H					
I					
J					
K					
L					
M					
N					
O					
P					
Q					
R					
S					
T					
U					
V					
W					
X					
Y					
Z					
[					
¥					
]					
^					
_					

PCDUC

Uの文字、左の  
表、上から22番  
目、Cの文字、右  
の表上から4番目

\* 当マニュアルの補足等は下記URLにて公開します  
<http://keic.jp/33-9709-1/>

### 本製品のお問い合わせは

〒556-0005 大阪市浪速区日本橋5-8-26  
共立電子産業株式会社、ケイシーズ担当までお願いします  
TEL(06)6644-4447  
FAX(06)6644-4448  
Email:keiseeds@kyohritsu.com

Copyright 1999 ~ 2025 (C) 共立電子産業株式会社