

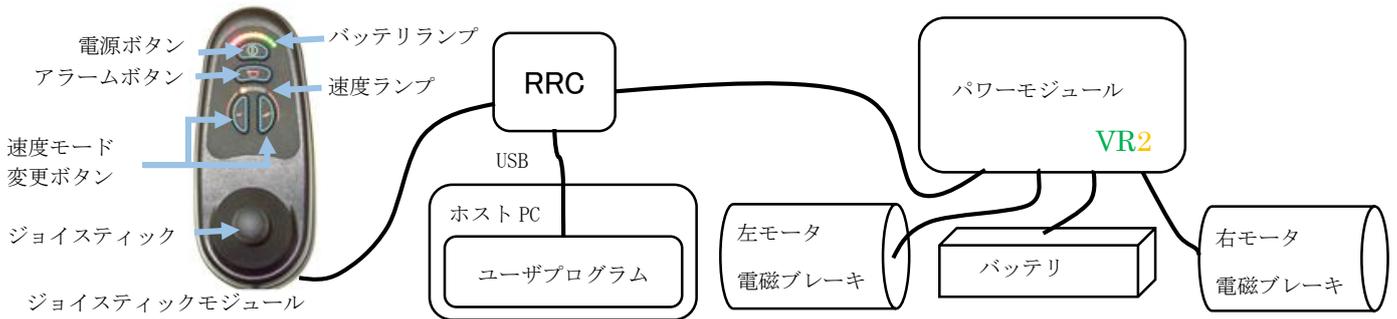
# RRC : Robotic Relay Communicator for VR2

2015年8月23日版 株式会社 Doog

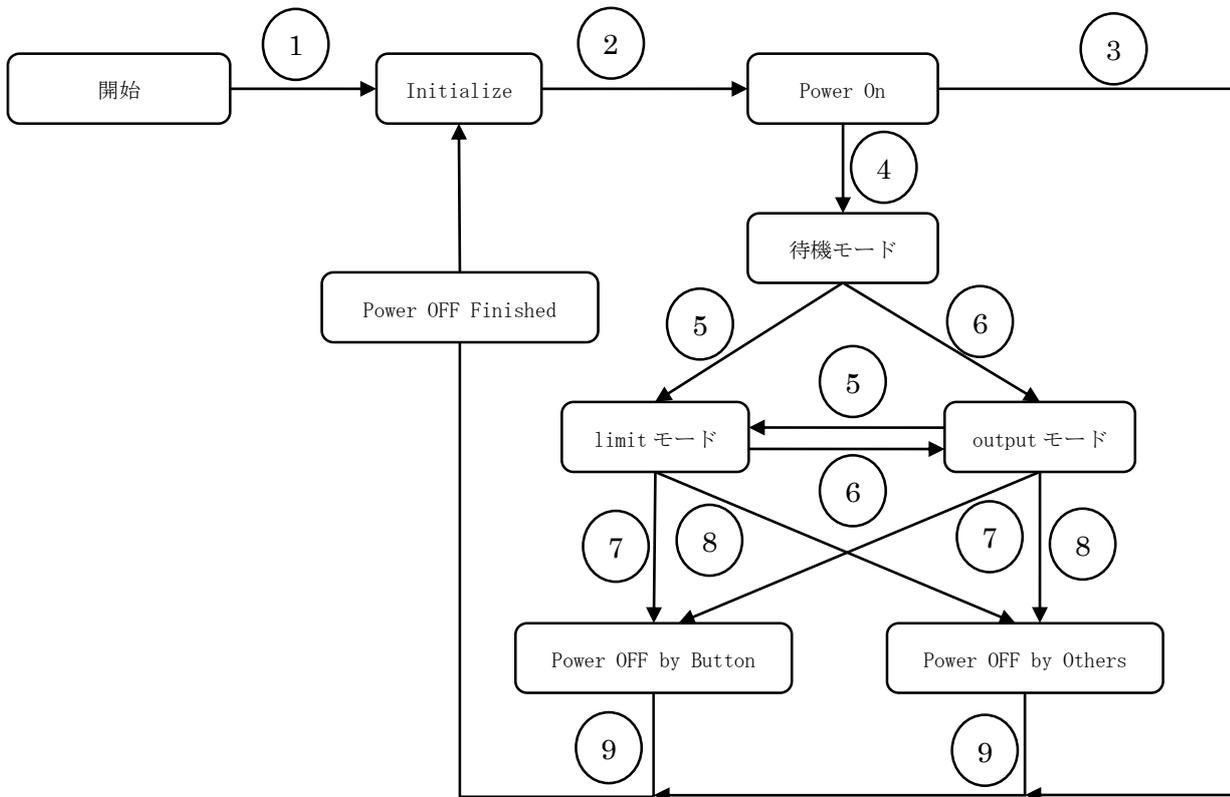
## 概要

本デバイスは、ジョイスティックとパワーユニットの間に挿入することで通信信号の内容を調整するものである。また、PC等をホストとして、各設定の変更、出力指令の送信、各状態の受信をすることが出来る。さらに、VR2システムの全機能の利用だけでなく、新たに追加された機能を自由にハンドリングすることで、車両のロボット化を実現するための簡単で強力なデバイスである。

## 配線図



## 状態遷移図



### 状態説明

開始	RRC に電源が入っていない状態
Initialize	ここでは、以下の処理が行われる <ul style="list-style-type: none"> <li>速度モードを強制的にもっとも低速な 1 にする</li> <li>コマンドにより設定されていた情報がリセットされる</li> </ul>

wait Power ON by Button	コントローラの電源ボタンが押されて ON 状態になるのを待つ
待機モード	コマンドにより limit か output が指定されるまで待機する
limit モード	ユーザのジョイスティック操縦で動作する状態 “joystick” コマンドは変更されると即座に適用される
output モード	ユーザプログラムの設定値で動作する状態 “joystick” コマンドは変更できるが本モードでは適用されない
Power OFF by Button	VR2 システムの通信が正常に終了することを待つ
Power OFF By Others	VR2 システムの通信が正常に終了することを待つ ただし、以下の点でボタンによる終了と異なることに注意すること <ul style="list-style-type: none"> <li>VR2 システムの通信を強制的に停止させる（モータ停止）</li> <li>コントローラの LED が点滅するエラー状態が表示される</li> <li>コントローラの電源ボタンを押して VR2 を OFF 状態にする必要がある</li> </ul>
Power OFF Finished	ホストに VR2 の通信が止まったことを報告し、即座に Initialize に戻る

・ 遷移条件

1	RRC に電源が供給された時
2	RRC の初期化が終了した時
3	コントローラの電源ボタンが押されたが ON 状態へ正常に切り替わることが出来なかった
4	コントローラの電源ボタンが押されて OFF 状態から ON 状態へ正常に切り替わった時
5	PC (ユーザプログラム) によって limit コマンドが RRC に送信された時
6	PC (ユーザプログラム) によって output コマンドが RRC に送信された時
7	コントローラの電源ボタンが押されて VR2 が停止した時
8	その他の理由により VR2 が停止した時 <ul style="list-style-type: none"> <li>ユーザプログラムから”poweroff” のコマンドが RRC に送信された時</li> <li>ユーザプログラムからの単位時間あたりの通信量が多すぎて RRC の処理が間に合わなかった時</li> <li>ユーザプログラムと RRC の通信にウォッチドッグタイマを設定しており、その時間内に通信が行われなかった時</li> <li>VR2 システムで通信される信号が一定時間以上の間、正常でない時</li> </ul>
9	VR2 の通信が正しく終了したのを確認した際

## 電源構成

電源構成 A : USB バスパワー	標準的な構成。USB ホスト側を接続している状態で動作し、5V20mA を消費する。 USB ホスト側が独立電源を持ち、コマンドによってのみ動かす場合にオススメ。
電源構成 B : 連動 VR2 (USB 供給)	VR2 が正常に ON の場合のみ動作し、VR2 電池側から 24V6.5mA を消費する。 RRC のための電源 SW が不要。USB ホスト側が 5V 系で動く場合に <b>1.5A まで利用可。※</b>
電源構成 C : 連動 VR2 (USB 切断)	VR2 が正常に ON の場合のみ動作し、VR2 電池側から 24V6.5mA を消費する。 RRC のための電源 SW が不要。USB ホスト側が独立した電源を持つ場合にオススメ。
電源構成 D : 常時 VR2 (USB 供給)	VR2 の動作状態に関わらず常に動作し、VR2 電池側から 24V6.5mA を消費し続ける。 RRC のための電源 SW が必要。USB ホスト側が 5V 系で動く場合に <b>1.5A まで利用可。※</b>
電源構成 E : 常時 VR2 (USB 切断)	VR2 の動作状態に関わらず常に動作し、VR2 電池側から 24V6.5mA を消費し続ける。 RRC のための電源 SW が必要。USB ホスト側が独立した電源を持つ場合にオススメ。

※電源構成 B と電源構成 D において、USB ホスト側からバスパワー電源が供給されるポートには接続しないこと。逆流による機器の破損の恐れが生じる。

## 通信仕様

ドライバ	<u>Windows では mbed.org の Windows serial configuration より取得しインストールする</u> Linux では追加インストール不要で/dev/ttyACM*が割り当てられる
通信設定	仮想 COM ポート, bitrate=115200bps, data=8bit, parity=0bit, stop=1bit
通信内容	可読文字, LF=LineFeed=0x0A, CR=CarriageReturn=0x0D のためテキスト通信可能
表記形式	“(ダブルクォーテーション囲み) = 囲まれた文字列を通信 2hex = 16 進数の 2 字(00~FF, 00~ff)を可読文字で通信 1hex = 16 進数の 1 字(0~F, 0~f)を可読文字で通信 CHECK = 1hex(‘0’~‘F’, ‘0’~‘f’)で表記されて、コマンドの先頭から CHECK 直前までの総和 RRC→ホストにおいては <u>CHECK の値が 10~15 の際は大文字で‘A’~‘F’</u> となる CHECK*1 = 初期状態では CHECK を含まず、stablemode コマンドで有効となる このため、コンソールからのデバッグ時は CHECK が不要となり、 ソフトウェアにより安定した連続通信する際は CHECK を必要とできる

## 注意事項

- ・ RRC のビットレートは 115200bps であり内部のコマンド処理周期は 10msec であるが、処理能力に限られるため次の基準を目安にコマンド通信を実施すること。PowerOnSuccess の期間は約 2 コマンド/1 周期を RRC 受け取り処理することが出来る、また同時に 1 周期毎に RRC は status を垂れ流し送信することが出来る。ただし、ホスト側から受信したコマンドに対し、stablemode の実施が無い場合正常応答(OK コマンド)の送信を含む場合や、コマンドの記載ミスによるエラー応答(ER コマンド)の送信を含む場合は処理時間が不足することで VR2 を強制終了することがある。なお、PowerOnSuccess 以外の期間も通信は可能であるが、最大のパフォーマンスではない瞬間も存在するため、1 コマンド/20msec 程度にするのが望ましい。またホストとの通信開始時には、ダミーのコマンドを送信して、100msec 程度のウェイトをし、受信バッファを空にしてから通信することで安定するケースがある。
- ・ ホスト→RRC のコマンドは小文字のみで表記される
- ・ ホスト→RRC のコマンドがエラー受信の場合、付随するすべての設定値は変化しない  
ただし、beep コマンドのみエラー受信であっても再生中のアラーム音は停止する
- ・ RRC→ホストのコマンドは 1 文字目が必ず大文字であり、複数の大文字を含むことがある
- ・ RRC→ホストにおいては CHECK の値が 10~15 の際は大文字で‘A’~‘F’となる
- ・ RRC または VR2 の電源を再投入すると、すべての設定が初期化される。よって、stablemode コマンドもリセットされるため、安定した連続通信を再開する場合は、再び stablemode コマンドを送信すること
- ・ limit モードと output モードを切り替える際に、ホスト側から speedmode1 を実施すると安全である
- ・ 車両事故防止のため、運用時は必ず stablemode コマンドと watchdog コマンドを併用すること  
watchdog は短い時間を設定し、安全確保必要なコマンドの応答が一定回数連続してエラー応答であれば、コマンド送信を打ち切ることで watchdog エラーを発生させるか、poweroff コマンドの送信に切り替える

## テスト手順

1. コントローラとパワーユニット、RRCなどを配線図のように接続する
2. PC上でターミナルソフト（TeraTermなど）を起動し、通信設定を確認してRRCと接続する
3. コントローラの電源ボタンを押し、VR2が起動できることを確認する  
この時、ターミナルに“EV:PowerOnTrigger”や“EV:PowerOnSuccess”と表示されているはずである
4. ターミナル上で、以下のサンプルコマンドを送信しRRCの動作が正しいことを確認する  
この段階では、stablemodeコマンドを送らずに、CHECKなしでテストするのが簡単である
5. ユーザプログラムを用いて、以下のサンプルコマンドを送信しRRCの動作が正しいことを確認する  
※ 注意事項に従い、安全な運用を心掛けること  
※ stablemodeコマンドを設定した以降は、コマンドの最後にCHECKを付け加えること

サンプルコマンド	CHECK	結果
limit3232ffffff	D	ジョイスティックを倒した時の最大速度が0x32=50[%]になる
output1e00	7	前進方向に0x1e=30[%]の速度で走行する
speedmode40	A	速度モードが4に変更され、コントローラボタンのロック無し
joystick00006464003051e2d	8	モード3 ジョイスティック値が半径5以内の場合、モータは動作しない 後ろ方向に30度までは、その場旋回を行う 後ろ方向に45度からは、ジョイスティックを倒した方向に後退する
watchdog64	B	0x64=100[×10msec]=1[sec]以内にコマンドを正常受信しなければ、VR2を強制的に停止させる
statusA1	6	RRCの内部状態が10msec周期で送信される
stablemode1	1	CHECK*1が必要になり、OKコマンドが送信されなくなる
poweroff	8	VR2を強制的に停止させる
beep646464640a0a0a0a0a0a	a	0x64=100[×10msec]=1[sec]おきにブザーが2度鳴った後、 0x0a=10[×10msec]=0.1[sec]おきにブザーが3度鳴る

## RRCの接続状態に、ホストからの通信なしでジョイスティック操作する手順

VR2のコントローラで電源をON状態にしてから待機状態の期間は、下記の通り任意のボタンを押しながらアラームボタンを押すことで、通常のVR2システムのようにPCを介さずに即座にジョイスティックから動かすことが出来る。なお、この状態からでもlimitモードおよびoutputモードへの移行は可能である。また、次の電源ON状態では改めて待機状態になった際はジョイスティックから操作できない状態へ戻る。

追加で押すボタン	選択される操作性
なし	VR2の通常の操作性
速度モード低下	後退操作では並進速度をゼロとし回転操作のみ継続 (VR2の後退最大速度がゼロの際と同様)
速度モード増加	ギア切替の模擬(自動アラームあり)&後退時はVR2と同様に車両が動く
速度モード低下+増加	ギア切替の模擬(自動アラームあり)&後退時はVR2と逆転し操作方向へ車両が動く

## ホスト→RRC：コマンドリスト

### limit コマンド

ユーザのジョイスティック操縦を許可し、所定の制限を行う

“limit”	2hex #1	2hex #2	2hex #3	2hex #4	2hex #5	CHECK *1	LF
---------	---------	---------	---------	---------	---------	----------	----

#1 = [0, 100] = 並進入力制限値

#2 = [0, 100] = 回転入力制限値

#3 = [0, 255] = 並進[%]+回転[%]の制限値（車体外周速度相当値）

#4 = [0, 255] = 並進[%]×回転[%]の制限値（遠心加速度相当値）

#5 = [0, 255] =  $\sqrt{(\text{並進}[\%])^2 + (\text{回転}[\%])^2}$ の制限値

### output コマンド

ユーザのジョイスティック操作を禁止し、ホストから出力指令を行う

“output”	2hex #1	2hex #2	CHECK *1	LF
----------	---------	---------	----------	----

#1 = [-100, 100] = 並進出力値

#2 = [-100, 100] = 回転出力値

### device コマンド

正常応答（OK コマンド）の後で、続けてデバイス名を答える

“device”	CHECK *1	LF
----------	----------	----

### version コマンド

正常応答（OK コマンド）の後で、続けてバージョン情報を答える

“version”	CHECK *1	LF
-----------	----------	----

### date コマンド

正常応答（OK コマンド）の後で、続けてコンパイル日時を答える

“date”	CHECK *1	LF
--------	----------	----

### speedmode コマンド

コントローラボタンで選択可能な速度モードをホストから変更する

“speedmode”	1hex #1	1hex #2	CHECK *1	LF
-------------	---------	---------	----------	----

#1 = [1, 5] = 速度モード(Profile)選択

#2 = [0, 1] = コントローラボタンによる変更ロックの無効(0)または有効(1)の指定

### speedmodearea コマンド

コントローラボタンで選択可能な速度モードの範囲をホストから変更する

“speedmodearea”	1hex #1	1hex #2	CHECK *1	LF
-----------------	---------	---------	----------	----

#1 = [1, 5] = 速度モード(Profile)の下限值

#2 = [1, 5] = 速度モード(Profile)の上限値

### joystick コマンド

ユーザのジョイスティック操縦時の挙動全般を調整する

“joystick”	2hex #1	2hex #2	2hex #3	2hex #4	1hex #5	1hex #6
1hex #7	2hex #8	2hex #9	2hex #10	CHECK *1	LF	

- #1 = [0, 100] = 並進下側不感体, 必ず下側<上側とする, 本値の入力値を並進出力の 0 とする  
 #2 = [0, 100] = 回転下側不感体, 必ず下側<上側とする, 本値の入力値を回転出力の 0 とする  
 #3 = [0, 100] = 並進上側不感体, 必ず下側<上側とする 本値の入力値を並進出力の 100 とする  
 #4 = [0, 100] = 回転上側不感体, 必ず下側<上側とする, 本値の入力値を回転出力の 100 とする  
 #5 = [0, 15] = 並進べき乗指数, 入力値<sup>(1+本値\*0.1)</sup>が出力値となる  
 #6 = [0, 15] = 回転べき乗指数, 入力値<sup>(1+本値\*0.1)</sup>が出力値となる  
 #7 = [0, 3] = 後退動作のモード選択  
     0 : VR2 の通常の操作性  
     1 : 後退操作では並進速度をゼロとし回転操作のみ継続 (VR2 の後退最大速度がゼロの際と同様)  
     2 : ギア切替の模擬(自動アラームあり)&後退時は VR2 と同様に車両が動く  
     3 : ギア切替の模擬(自動アラームあり)&後退時は VR2 と逆回転するため操作方向へ車両が動く  
 #8 = [0, 100] = ギア切替の模擬におけるニュートラル (モードリセット) 半径  
 #9 = [0, 90] = ギア切替の模擬における前後モードの切替閾値角度 (後方を 0deg とし絶対値で適用される)  
 #10 = [0, 90] = ギア切替の模擬における後退モードの減衰開始角度 (後方を 0deg とし絶対値で適用される)  
 ※ #1~#6 は入力の正負ともに適用される  
 ※ #6 の回転べき乗指数は、並進入力 $\pm 100$ の時に比率 1 で適用され、並進入力が 0 の時に比率 0 で適用され、その中間は線形補間される  
 ※ #7 が 0 or 1 を選択している際は#8~#10 無視される  
 ※ ギア切替の模擬ではニュートラル半径内に 0.2 秒以上ジョイスティックが留まるとモードリセット  
 ※ ギア切替の模擬における前進モードでは後退方向に操作した際に並進速度をゼロとして回転操作のみ継続  
 ※ ギア切替の模擬における後退モードでは前進方向に操作した際に並進・回転速度の両方がゼロ停止

## joystickadjust コマンド

ユーザのジョイスティック操縦時の操作感度のみを**符号毎**に調整する

“joystickadjust”	2hex #1	2hex #2	2hex #3	2hex #4	2hex #5	2hex #6	
2hex #7	2hex #8	1hex #9	1hex #10	1hex #11	1hex #12	CHECK *1	LF

- #1 = [0, 100] = +並進下側不感体, 必ず下側<上側とする, 本値の入力値を並進出力の 0 とする  
 #2 = [0, 100] = +並進上側不感体, 必ず下側<上側とする, 本値の入力値を並進出力の 100 とする  
 #3 = [0, 100] = -並進下側不感体, 必ず下側<上側とする, 本値の入力値を並進出力の 0 とする  
 #4 = [0, 100] = -並進上側不感体, 必ず下側<上側とする, 本値の入力値を並進出力の 100 とする  
 #5 = [0, 100] = +回転下側不感体, 必ず下側<上側とする, 本値の入力値を回転出力の 0 とする  
 #6 = [0, 100] = +回転上側不感体, 必ず下側<上側とする, 本値の入力値を回転出力の 100 とする  
 #7 = [0, 100] = -回転下側不感体, 必ず下側<上側とする, 本値の入力値を回転出力の 0 とする  
 #8 = [0, 100] = -回転上側不感体, 必ず下側<上側とする, 本値の入力値を回転出力の 100 とする  
 #9 = [0, 15] = +並進時の並進べき乗指数, 入力値<sup>(1+本値\*0.1)</sup>が出力値となる  
 #10 = [0, 15] = -並進時の並進べき乗指数, 入力値<sup>(1+本値\*0.1)</sup>が出力値となる  
 #11 = [0, 15] = +並進時の回転べき乗指数, 入力値<sup>(1+本値\*0.1)</sup>が出力値となる, **並進方向で切り替わる**  
 #12 = [0, 15] = -並進時の回転べき乗指数, 入力値<sup>(1+本値\*0.1)</sup>が出力値となる, **並進方向で切り替わる**  
 ※ #1~#8 の並びが joystick コマンドと異なることに注意すること

※ #11～#12 の回転べき乗指数は、並進入力 $\pm 100$ の時に比率 1 で適用され、並進入力 $0$ の時に比率 0 で適用され、その中間は線形補間される

## joystickback コマンド

ユーザのジョイスティック操縦時の後退動作のみを調整する

“joystickback”	1hex #1	2hex #2	2hex #3	2hex #4	CHECK *1	LF
----------------	---------	---------	---------	---------	----------	----

#1 = [0, 3] = 後退動作のモード選択

0 : VR2 の通常の操作性

1 : 後退操作では並進速度をゼロとし回転操作のみ継続 (VR2 の後退最大速度がゼロの際と同様)

2 : ギア切替の模擬(自動アラームあり)&後退時は VR2 と同様に車両が動く

3 : ギア切替の模擬(自動アラームあり)&後退時は VR2 と逆回転するため操作方向へ車両が動く

#2 = [0, 100] = ギア切替の模擬におけるニュートラル (モードリセット) 半径

#3 = [0, 90] = ギア切替の模擬における前後モードの切替閾値角度 (後方を 0deg として絶対値で適用される)

#4 = [0, 90] = ギア切替の模擬における後退モードの減衰開始角度 (後方を 0deg として絶対値で適用される)

※ #1 が 0 or 1 を選択している際は#2～#3 無視される

※ ギア切替の模擬ではニュートラル半径内に 0.2 秒以上ジョイスティックが留まるとモードリセット

※ ギア切替の模擬における前進モードでは後退方向に操作した際に並進速度をゼロとして回転操作のみ継続

※ ギア切替の模擬における後退モードでは前進方向に操作した際に並進・回転速度の両方がゼロ停止

## watchdog コマンド

ホストから RRC に指定時間以上においてコマンドの正常受信が無い場合に VR2 を強制終了する

コマンドをエラー受信した場合には安全のため保持時刻は更新されない

“watchdog”	2hex #1	CHECK *1	LF
------------	---------	----------	----

#1 = [0, 255] = 時間[ $\times 10\text{ms}$ ], 0 は未設定と見なす

## stablemode コマンド

本コマンドの有効指定に対する OK コマンドを送った以降は下記の挙動に変わる

- ・ [ホスト $\rightarrow$ RRC]の CHECK \*1 が有効となる
- ・ [RRC $\rightarrow$ ホスト]の OK コマンドが省略される

“stablemode”	1hex #1	CHECK *1	LF
--------------	---------	----------	----

#1 = [0, 1] = 無効(0)または有効(1)の指定

## status コマンド

RRC 側で得られる各種ステータス情報の垂れ流し送信を開始する

- ・ 内部処理周期と同期したタイミング (約 10msec ごと)
- ・ 送信されるのは、状態遷移図において待機モード $\sim$ PowerOffWait までの間に限られる

“status”	1hex #1	1hex #2	CHECK *1	LF
----------	---------	---------	----------	----

#1 = [‘A’ or ‘B’] = type A または typeB を指定

#2 = [0, 1] = 無効(0)または有効(1)の指定

## encoderresistance コマンド

ロータリーエンコーダ接続端子 (2 モータ 2 相) の端子抵抗を選択する

“encoderresistance”	1hex #1	CHECK *1	LF
---------------------	---------	----------	----

#1 = [0, 2] = 0 はハイインピーダンス, 1 はプルアップ, 2 はプルダウン

## relayerror コマンド

ジョイスティックモジュールまたはパワーモジュールとの通信エラーが連続した場合に

EV:PowerOffWaitByJoystickModuleError または EV: PowerOffWaitByPowerModuleError を発生する

“relayerror”	1hex #1	CHECK *1	LF
--------------	---------	----------	----

#1 = [1, 9] = エラーイベントを発生させる通信エラーの連続回数

## unlockbrake コマンド

output モードにおいて、並進出力値と回転出力値の両方の絶対値が本値未満である場合に、電磁ブレーキをロック解除状態にするために、パワーモジュールにゼロ値と本値以上となる出力値を交互に自動送出する

“unlockbrake”	2hex #1	CHECK *1	LF
---------------	---------	----------	----

#1 = [0, 100] = 電磁ブレーキのロックが解除される出力値 (0 だと本機能は無効化)

※弊社調べでは、並進出力値と回転出力値の両方が±10 未満の場合に電磁ブレーキがロックします

※本コマンドを安定動作するため、実機による事前の動作確認を入念に実施してください

※本電磁ブレーキを解除している場合は、モータを外部から駆動しても VR2 はエラーを生じません

## poweroff コマンド

VR2 を強制終了する

“poweroff”	CHECK *1	LF
------------	----------	----

## beep コマンド

RRC に付属するブザーからパターン音を鳴らす

“beep”	2hex #1	2hex #2	2hex #3	2hex #4	…	2hex #9	2hex #10	CHECK *1	LF
--------	---------	---------	---------	---------	---	---------	----------	----------	----

#1, #3, #5, #7, #9 = [0, 255] = ブザーオン時間[×10ms]

#2, #4, #6, #8, #10 = [0, 255] = ブザーオフ時間[×10ms]

## RRC→ホスト：コマンドリスト

### EV コマンド (RRC や VR2 の状態が変わるイベントが生じた際)

RRC が初期化された際に送られる

“EV:Initialized”	LF
------------------	----

VR2 の電源が ON にされた際に送られる

“EV:PowerOn”	LF
--------------	----

VR2 の電源が ON にされて正常に RRC の通信を開始できた際に送られる

※このイベント以降に諸々の設定コマンドを実施し、確認が取れた後に連続通信を開始する構成が望ましい

“EV:PowerOnSuccess”	LF
---------------------	----

VR2 の電源が ON にされたが RRC の通信を開始できない問題が生じた際に送られる (要因 1)

“EV:PowerOnError1”	LF
--------------------	----

VR2 の電源が ON にされたが RRC の通信を開始できない問題が生じた際に送られる (要因 2)

“EV:PowerOnError2”	LF
--------------------	----

VR2 の電源が ON にされたが RRC の通信を開始できない問題が生じた際に送られる (タイムアウト)

“EV:PowerOnTimeout”	LF
---------------------	----

ユーザがコントローラの電源ボタンを押した際に送られる

“EV:PowerOffWaitByButton”	LF
---------------------------	----

通信系に連続で異常が生じたことにより RRC を終了する際に送られる

“EV:PowerOffWaitByCommunicationError”	LF
---------------------------------------	----

ジョイスティックモジュールとの通信内容に連続で異常が生じたことにより RRC を終了する際に送られる

“EV:PowerOffWaitByJoystickModuleError”	LF
--	----

パワーモジュールとの通信内容に連続で異常が生じたことにより RRC を終了する際に送られる

“EV:PowerOffWaitByPowerModuleError”	LF
-------------------------------------	----

VR2 がエラーを検知したことにより RRC を終了する際に送られる

“EV:PowerOffWaitBySystemError”	LF
--------------------------------	----

ホストから poweroff コマンドを受け取り、RRC を強制終了する際に送られる

“EV:PowerOffWaitByHost”	LF
-------------------------	----

watchdog コマンドが設定済みで、設定時間を超えたことにより RRC を終了する際に送られる

“EV:PowerOffWaitByWatchdog”	LF
-----------------------------	----

内部処理やホストとの通信時間が規定値を超えたことにより RRC を終了する際に送られる

“EV:PowerOffWaitByOverTime”	LF
-----------------------------	----

コントローラとパワーユニットと通信が途絶えた際に送られる (コネクタが抜けた際)

“EV:PowerOffWaitByNoCommunication”	LF
------------------------------------	----

VR2 の電源が完全に OFF になった際に送られる

“EV:PowerOffFinished”	LF
-----------------------	----

## OK コマンド (ホストからの受信を処理できた際の正常応答)

“OK:”	*****	LF
-------	-------	----

\*\*\*\*\* = ホスト→RRC のコマンド名

※stablemode コマンドの有効を正常応答した以降、本コマンドは省略される

## ER コマンド (ホストからの受信が処理できない際のエラー応答)

“ER:”	*****	LF
-------	-------	----

\*\*\*\*\* = ホスト→RRC のコマンド名

その他の ER コマンドとして下記が存在する。

CommandChecksum : チェックサムの不一致 ⇒ 計算方式を見直すこと

CommandUnknown : 未定義のコマンドを受信 ⇒ 通信内容を再確認すること

ReceiveBufferOverflow : 受信バッファ溢れ ⇒ 送信データ量を注意事項に従い見直すこと

## SA コマンド(Status - type A, 垂れ流し応答)

“SA:”	2hex #1	2hex #2	2hex #3	2hex #4	1hex #5	1hex #6
1hex #7	1hex #8	1hex #9	CHECK	LF		

#1 = [-100, 100] = 並進入力値 (ジョイスティックの入力を RRC が受け取った直後の値)

#2 = [-100, 100] = 回転入力値 (ジョイスティックの入力を RRC が受け取った直後の値)

#3 = [-100, 100] = 並進出力値 (RRC がパワーユニットへ出力を渡す直前の値)

#4 = [-100, 100] = 回転出力値 (RRC がパワーユニットへ出力を渡す直前の値)

#5 = [1, 10] = 電圧ゲージ値

#6 = [1, 5] = 速度モード(Profile)選択値

#7 = [0, 15] = 速度モード増加ボタン入力カウンタ (循環する増加カウンタ、押すと奇数、離すと偶数)

#8 = [0, 15] = 速度モード減少ボタン入力カウンタ (循環する増加カウンタ、押すと奇数、離すと偶数)

#9 = [0, 15] = アラームボタン入力カウンタ (循環する増加カウンタ、押すと奇数、離すと偶数)

※本コマンドは **PowerOnSuccess の期間でのみ** 送出される

※本コマンドの CHECK は常に有効

※**CHECK の値が 10~15 の際は**大文字で'A'~'F'となる

## SB コマンド(Status - type B, 垂れ流し応答)

“SB:”	2hex #1	2hex #2	2hex #3	2hex #4	1hex #5	1hex #6	
1hex #7	1hex #8	1hex #9	2hex #10	2hex #11	CHECK	LF	

#1～#9 = SA コマンドを参照すること

#10 = [0, 255] = 左側ロータリーエンコーダのカウンタ値（循環する増減カウンタ）

#11 = [0, 255] = 右側ロータリーエンコーダのカウンタ値（循環する増減カウンタ）

※本コマンドは **PowerOnSuccess の期間でのみ** 送出される

※本コマンドの CHECK は常に有効

※**CHECK の値が 10～15 の際は**大文字で'A'～'F'となる

## オプション品情報：2モータ2相ロータリーエンコーダ

※1周期（10msec）に1モータあたり1byteのデータを送るため、1周期あたりに取得可能なカウンタ量は約120cnt（ $\approx 256/2$ ）です。このため最大読み取り速度は約120cnt/10msecです。

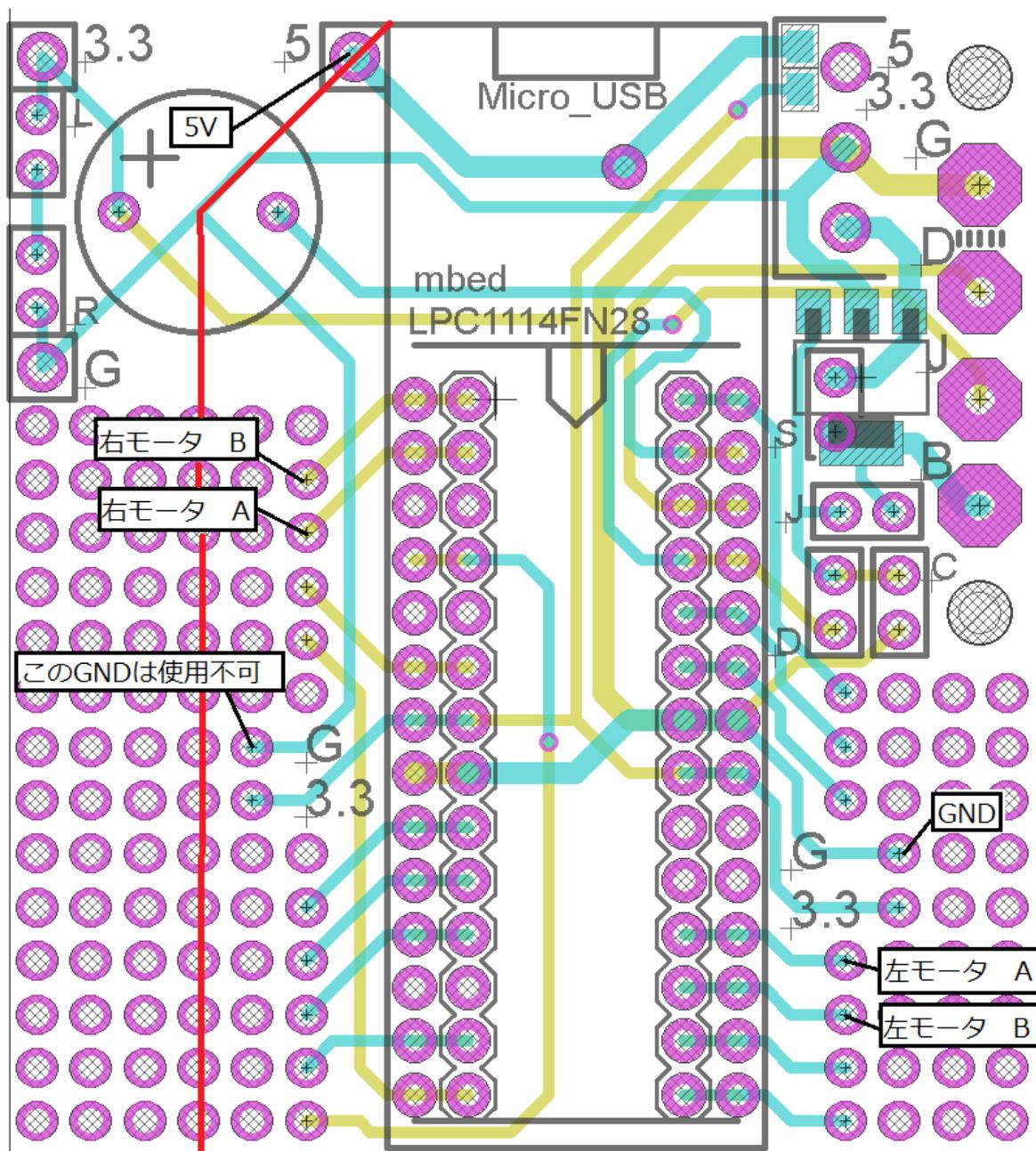
ハードウェア設計時には、最大速度において12000cnt/sを十分に下回るように注意してください。

※出荷時のコネクタはロボットケーブルと3Mの中継用e-CONコネクタメスをを用い下記の配線です。

1番=GND=黒、2番=5V=赤、3番=A=緑、4番=B=白を配線します。

※車両が前進した際に、左右のモータの回転方向が逆になることを見越した配線とソフトウェアです。

ただし、車両が前進した際にエンコーダデータが後退する結果の場合、AB相の配線の入れ替えまたはユーザプログラムにおいて正逆の入れ替えを実施してください。



## オプション品情報：Futaba（株） 互換のラジコン用プロポ

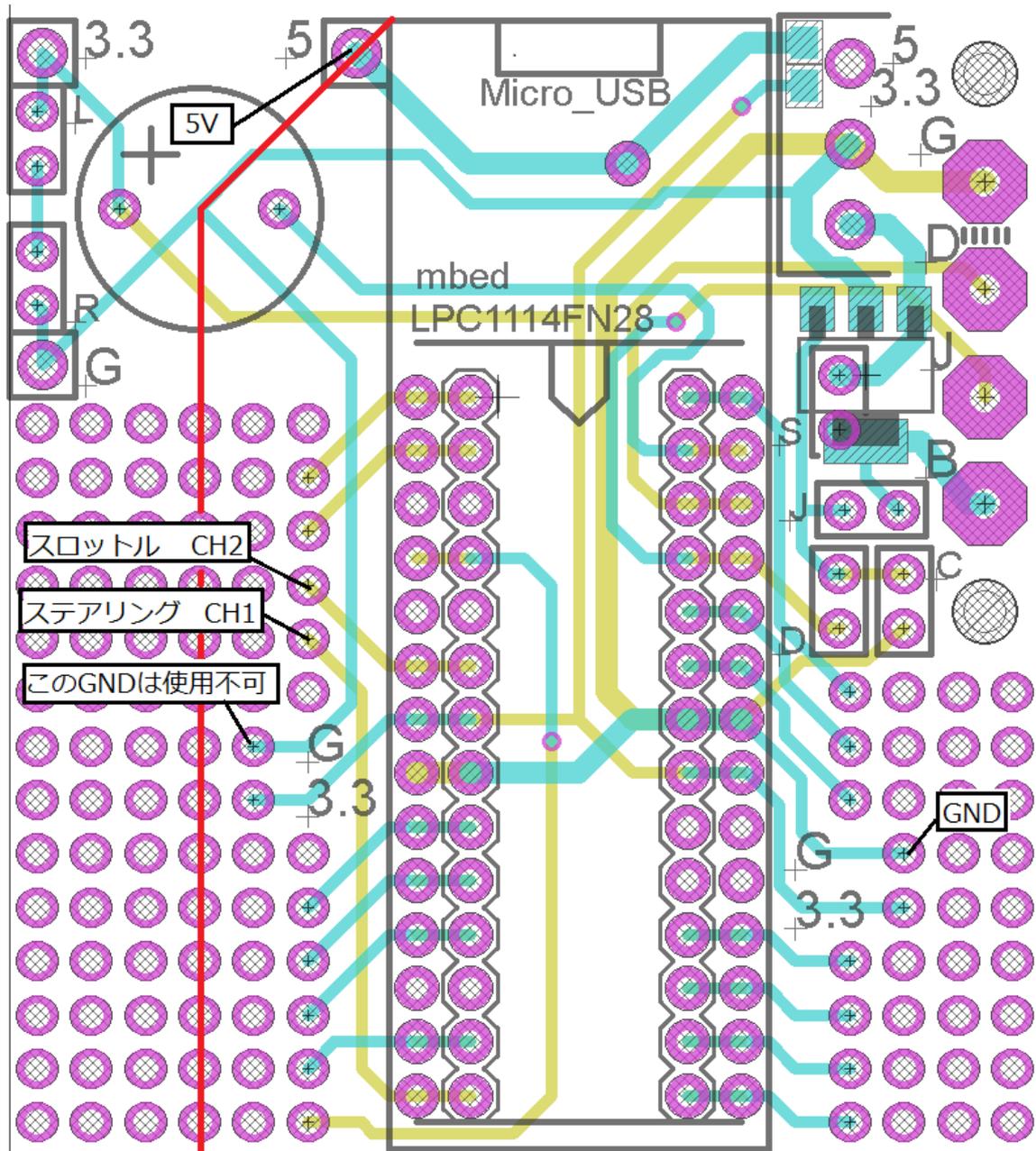
※3PRKA R203GF T/R セットにおいて動作確認済み

※status 応答をしていない状態でプロポと接続が確立した場合に、

ユーザデバッグ用に DE:[スロットル][ステアリング]の形式で表示されます

事前にデバッグ情報を画面で見ながら、原点およびスケールの調整を実施してください

※limit モード動作時にのみ、プロポの値をジョイスティック入力値に上書きしてラジコン操作が可能です



中央のピン列は共通のため1箇所にて5Vを接続

右側のピン列は共通のため1箇所にてGNDを接続

スロットル CH2 のデータピン (左側) を接続

ステアリング CH1 のデータピン (左側) を接続

## 変更履歴

<u>date</u>	<u>version</u>	
2015年8月23日版	2.0	<ul style="list-style-type: none"><li>・ 電源構成を整理して詳細に解説</li><li>・ speedmodearea コマンドの追加</li><li>・ joystickadjust コマンドの追加</li><li>・ joystickback コマンドの追加</li><li>・ status において typeB の要求と応答を追加</li><li>・ encoderresistance コマンドの追加</li><li>・ relayerror コマンドの追加</li><li>・ unlockbrake コマンドの追加</li><li>・ オプション品情報としてロータリーエンコーダを追加</li><li>・ オプション品情報としてラジコン用プロポを追加</li></ul>
2015年4月30日版	1.0	<ul style="list-style-type: none"><li>・ stablemode コマンドの表記ミスを修正</li><li>・ ボーレート→ビットレートに表記を修正</li><li>・ アクチュエータ付きのジョイスティックモジュールおよびパワーモジュールにソフトウェアを対応</li></ul>
2015年4月27日版	1.0(未販売)	<ul style="list-style-type: none"><li>・ EV コマンドに「EV:PowerOffWaitByNoCommunication」を追加</li><li>・ 「ジョイスティック」と「モータドライバ」をそれぞれ「ジョイスティックモジュール」と「パワーモジュール」に名称を変更し、それに伴い EV コマンドの文字列を変更</li></ul>
2015年4月13日版	1.0(未販売)	初版