

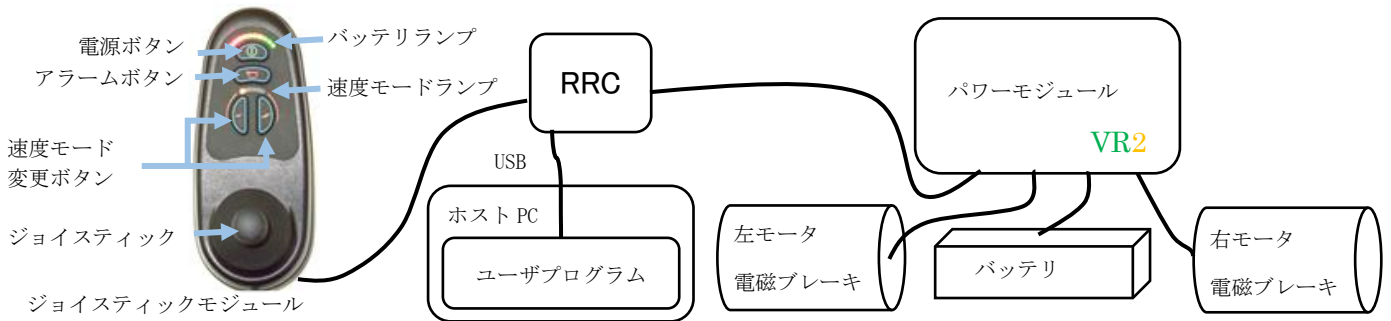
RRC : Robotic Relay Communicator for VR2

2016年8月2日版 株式会社 Doog

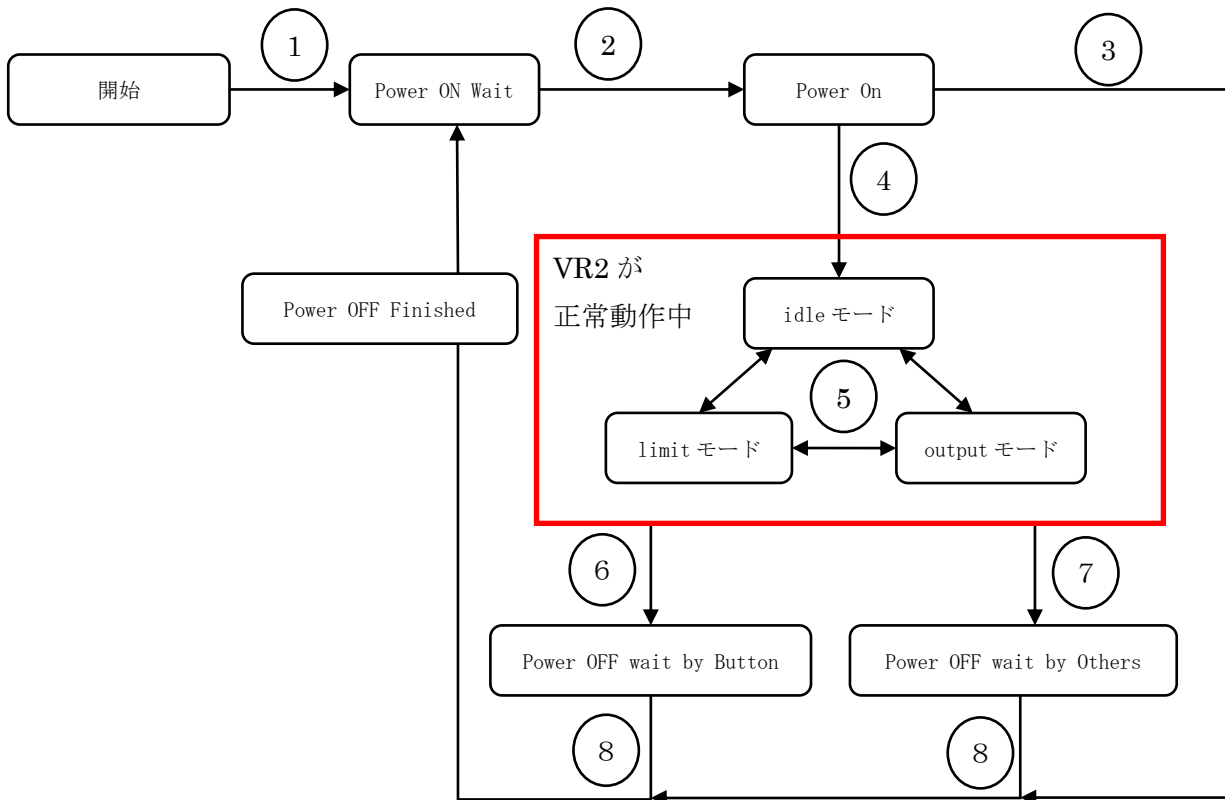
概要

本デバイスは、ジョイスティックモジュール（以下、JM と記載）とパワーモジュール（以下、PM と記載）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
Power ON Wait	JM の電源ボタンが押されて ON 状態になるのを待つ
idle 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 Wait 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 Wait 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、JM の電源ボタンを押して VR2 を OFF 状態にする必要がある ※poweroff コマンドのみ VR2 システムの中継通信を強制終了できます
Power OFF Finished	ホストに VR2 の通信が止まったことを報告して PowerOnWait に戻る ※poweroffdelay コマンドを設定すると終了を遅れさせることができます

・ 遷移条件

1	RRC に電源が供給された時
2	JM の電源ボタンが押された時
3	RRC の中継通信を介して ON 状態へ正常に切り替わることが出来なかった時
4	RRC の中継通信を介して 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, output, idle コマンドが RRC に送信された時
6	JM の電源ボタンが押されて VR2 が停止した時
7	その他の理由により VR2 が停止した時 ・ ”poweroff”のコマンドが RRC に送信された時 ・ PC と RRC 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 ・ 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 ・ 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
8	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 ON の場合のみ RRC が動作し、VR2 電池側から 24V6.5mA を消費する。 このため VR2 を電源 SW として代替可。USB ホスト側へ 5V 最大 1.5A まで出力。
電源構成 C : 連動 VR2 (USB 切断)	VR2 が ON の場合のみ RRC が動作し、VR2 電池側から 24V6.5mA を消費する。 このため VR2 を電源 SW として代替可。USB ホスト側と RRC で 5V 電源を切り離せる。
電源構成 D : 常時 VR2 (USB 供給)	VR2 の OFF でも RRC が動作し、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 ホスト側へ 5V 最大 1.5A まで出力。
電源構成 E : 常時 VR2 (USB 切断)	VR2 が OFF でも RRC が動作し、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 ホスト側と RRC で 5V 電源を切り離せる。

※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。

※電源構成 C と電源構成 E は供給を廃止しました。お客様にて USB ケーブルの 5V 線を切断してください。

通信仕様

ドライバ	Windows では mbed.org の Windows serial configuration より取得しインストールする 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 なお、下記コマンド表では PC からの送信側の改行コードを LF としているが CR+LF でも良い
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 CHECK の値が 10~15 の際は大文字で‘A’~‘F’ 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、処理能力が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 程度にするのが良い。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。
 - status コマンド(typeS)では、データ長を s1set コマンド等で変更することが出来る。コマンドの処理には 1 周期あたり 5msec を割り当てているため、理想的には $115200\text{bps} * 0.005\text{s} = 576\text{bit} \approx 57\text{byte}$ を送出することが出来る。ただし、他の通信と重なることで処理時間を溢れる可能性について考慮する必要がある。
- ・ 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、安全確保に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. JM と PM、RRC など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. JM 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
limit3232ffff	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
stablemode1	1	CHECK*1 が必要になり、OK コマンドが送信されなくなる
stablemode0	0	CHECK*1 が不要になり、OK コマンドが送信されるようになる
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

リカバリ状態：RRC の接続状態で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

JM の電源ボタンにより ON 状態にしてから **2 秒以内**に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 **2 秒以上の長押し**することで、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、次回の電源 ON 後に idle モードとなるまで limit モードおよび output モードへの**移行はできない**。

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作を許可し、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

idle コマンド

ユーザのジョイスティック操作を禁止し、JM のランプにより待機状態であることを示す。

"idle"	CHECK *1	LF
--------	----------	----

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
--------	----------	----

state コマンド

正常応答（OK コマンド）の後で、続けて現在の RRC 状態を答える

※次の種類が存在する poweronwait, poweron, poweroffwait, idle, limit, output, actuator(※検証不十分)

"state"	CHECK *1	LF
---------	----------	----

timestamp コマンド

正常応答（OK コマンド）の後で、続けて現在の RRC の内蔵タイマ時刻[ms]を 2hex 形式で答える

"timestamp"	CHECK *1	LF
-------------	----------	----

speedmode コマンド

JM で選択可能な速度モード(Profile)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] = JM の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

JM で選択可能な速度モード(Profile)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

※本コマンド適用時に joysticklength の設定値は破棄される

“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^(1+本値*0.1)が出力値となる

#6 = [0, 15] = 回転べき乗指数, 入力値^(1+本値*0.1)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 として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 として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 ± 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

※本コマンド適用時に joysticklength の設定値は破棄される

“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^(1+本値*0.1)が出力値となる

#10 = [0, 15] = -並進時の並進べき乗指数, 入力値^(1+本値*0.1)が出力値となる

#11 = [0, 15] = +並進時の回転べき乗指数, 入力値^(1+本値*0.1)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^(1+本値*0.1)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、並進入力に±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

joysticklength 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 **半径長さ単位** で調整する

※本コマンド適用時に joystick や joystickadjust の不感帯とべき乗指数の設定値は破棄される

“joysticklength”	2hex #1	2hex #2	1hex #3	CHECK *1	LF
------------------	---------	---------	---------	----------	----

#1 = [0, 100] = 長さ下側不感体, 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] = 長さ上側不感体, 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, 15] = 長さべき乗指数, 入力値^(1+本値*0.1)が出力値となる

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

poweroffdelay コマンド

ジョイスティックモジュールの電源ボタンを押した際と、forcepoweroff コマンドを受信した際に VR2 システムの中継通信を強制終了し、その後 PowerOnWait になるまでの遅れ時間を設定する。遅れ時間中は、電源構成が B,C の場合には RRC 側システムは保持されて遅れ時間後に OFF になる。

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

#1 = [0, 255] = 時間[×100ms], 0 は無効 (未設定) と見なす

watchdog コマンド

ホストから RRC に指定時間以上のコマンド正常受信が無い場合に RRC をエラー状態にする

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

#1 = [0, 255] = 時間[×10ms], 0 は無効 (未設定) と見なす

※本コマンドをエラー受信した場合には安全のため内部保持時刻は更新されない

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が変わる

- ・ [ホスト→RRC]の CHECK *1 が有効となる
- ・ [RRC→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を約 10msec ごとに垂れ流し送信する

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

#1 = [‘S’] = typeS を指定

#2 = [0, 1] = 無効(0)または有効(1)の指定

【typeS】

- ・ モード切替時に一時的に送信不可になる場合を除いて常に送信される
- ・ 応答は[S]から始まるコマンドとなる (通信量削減のため[SS:]でないことに注意)

s1set コマンド

S コマンド(statusS の応答)の順序 1 データ (基準情報) の選択する

“s1set”	1hex #1	CHECK *1	LF
---------	---------	----------	----

#1 = [0, 1] = 「タイムスタンプ(2hex)」の無効(0)または有効(1)の指定

s2set コマンド

S コマンド(statusS の応答)の順序 2 データ (標準 JM 情報) の選択する

“s2set”	1hex #1	1hex #2	1hex #3	1hex #4	CHECK *1	LF
---------	---------	---------	---------	---------	----------	----

#1 = [0, 1] = 「JM からの入力値(4hex)」の無効(0)または有効(1)の指定

#2 = [0, 1] = 「PM への出力値(4hex)」の無効(0)または有効(1)の指定

#3 = [0, 1] = 「JM の LED 表示 (2hex, 電圧ゲージ値, 速度モード選択値)」の無効(0)または有効(1)の指定

#4 = [0, 1] = 「JM のボタン入力カウンタ (3hex)」の無効(0)または有効(1)の指定

s3set コマンド

S コマンド(statusS の応答)の順序 3 データ (ロータリーエンコーダ&ラジコン用プロポ情報) の選択する

"s3set"	1hex #1	1hex #2	CHECK *1	LF
---------	---------	---------	----------	----

#1 = [0, 1] = 「ロータリーエンコーダのカウント値(4hex)」の無効(0)または有効(1)の指定

#2 = [0, 1] = 「ラジコン用プロポの入力値(4hex)」の無効(0)または有効(1)の指定

※ #2 は propo コマンドの設定で機能 ON になっている場合のみ

s4set コマンド

S コマンド(statusS の応答)の順序 4 データ (汎用 IO の 8 ピン情報) の選択する

"s4set"	1hex #1	2hex #2	CHECK *1	LF
---------	---------	---------	----------	----

#1 = [0, 1] = 「汎用 IO ピンの電位状態(2hex)」の無効(0)または有効(1)の指定

#2 = [0, 255] = 「汎用 IO ピンのカウント値(1hex×8 ピン)」の無効(0)または有効(1)をビット指定

s5set コマンド(予定)

S コマンド(statusS の応答)の順序 5 データ (VR2 アクチュエータ情報) の選択する

pinmode コマンド

汎用ピン (8 ピン) の動作モードを設定する

"pinmode"	1hex #1	1hex #2	...	1hex #7	1hex #8	CHECK *1	LF
-----------	---------	---------	-----	---------	---------	----------	----

#1~8 = [0, 5] = ピン 7~0 (#1 がピン 7、#8 がピン 0 に対応) を下記の通り設定変更する

値 0 = 設定変更しない

値 1 = 入力設定で PullNone に変更する

値 2 = 入力設定で PullUp に変更する

値 3 = 入力設定で PullDown に変更する

値 4 = 出力設定で Low に変更する

値 5 = 出力設定で High に変更する

※初期値=1 で入力設定で PullNone となっています

※ピン 0 とピン 1 は入力設定において PullUp/PullDown が設定不可のため外部回路で対応してください

※ピン 0 とピン 1 は出力設定において OpenDrain 端子となりますので外部回路で対応してください

※設定変更したピンは、s4set #2 で応答するカウント値がゼロクリアされます

※encoderresistance コマンドと異なり、値 1 から設定が変更されます

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 PullNone, 1 は PullUp, 2 は PullDown として設定する

※pinmode コマンドと異なり、値 0 から設定が変更されます

relayerror コマンド

JM または PM 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

※初期値は 5 となっています

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、PM 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

※VR2 の設定または個体によっては、ロック解除時に僅かに動作してしまう場合があります

※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

forcepoweroff コマンド

VR2 の中継通信を強制終了する

※電源構成が B,C の場合には RRC 側システムは連動して OFF になる

※poweroffdelay コマンドを設定していると上記連動 OFF と EV:PowerOffFinished が遅れます

“forcepoweroff”	CHECK *1	LF
-----------------	----------	----

forceerror コマンド

VR2 の中継通信をエラー状態に強制移行する

※エラー表示はホスト側から更新することが出来る

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

#1 = [0, 255] = バッテリランプのエラー表示状態 (未定義)

#2 = [0, 255] = 速度モードランプのエラー表示状態 (未定義)

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 のみを送る(変数無し)ことで 0.5 秒の音を鳴らすテストができます

“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]

propo コマンド

汎用ピンの 2 と 7 をプロポ入力に設定し、各機能のオンオフを設定する

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

#1 = [0, 255] = プロポ機能を以下に表記する各ビットで状態設定

0x01 → プロポ機能全体 (入力ピン設定を含む)

0x02 → 調整用のデバッグ出力

0x04 → プロポ操作値を limit 入力時のジョイスティック値として上書き (status 応答では区別される)

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)

JM の電源ボタンが押されるのを待つ状態へ移行した際に送られる

“EV:PowerOnWait”	LF
------------------	----

VR2 の電源が ON にされた際に送られる

“EV:PowerOnTrigger”	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を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を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を生じた際に送られる (タイムアウト)

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

んかい

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
---------------------	----

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
------------------------------------	----

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
--------------------------------	----

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
---------------------------	----

ロータリーエンコーダ右側(0 番)の 2 相カウントに飛びが生じた際に送られる

“EV:EncoderSkip0”	LF
-------------------	----

ロータリーエンコーダ左側(1 番)の 2 相カウントに飛びが生じた際に送られる

“EV:EncoderSkip1”	LF
-------------------	----

ホストから forcepoweroff コマンドを受け取り、VR2 の通信を強制終了する際に送られる

“EV:PowerOffWaitByForcePowerOff”	LF
----------------------------------	----

ホストから forceerror コマンドを受け取り、VR2 の通信をエラー状態に移行する際に送られる

“EV:PowerOffWaitByForceError”	LF
-------------------------------	----

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

“EV:PowerOffWaitByOverTime”	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 : チェックサム的一致 ⇒ 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

Recovery : リカバリ状態 ⇒ 改めて電源 ON にして idle モードへ戻すこと s

S コマンド(Status - type S, 垂れ流し応答)

“S”	...	CHECK	LF
-----	-----	-------	----

※データ部には下記表においてトリガを設定したデータが順に含まれる

※本コマンドはいずれの状態遷移においても常に送出される

※[SS:]から始まるのでは無いことに注意

※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

トリガ	データ種類	データ長と値範囲	備考
s1set #1	タイムスタンプ	2hex [0, 255]	msec

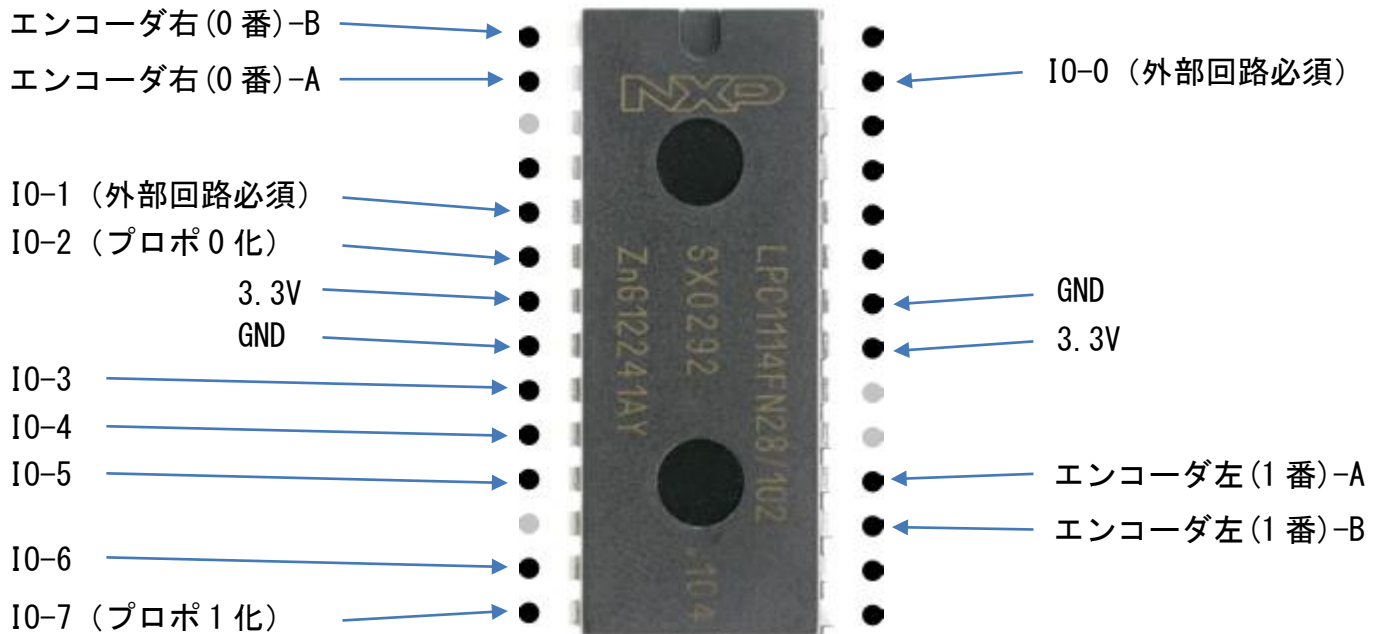
s2set #1	JM から受け取った直後の入力値	2hex [-100, 100] 並進 2hex [-100, 100] 回転	%
s2set #2	PM へ受け渡す直前の出力値	2hex [-100, 100] 並進 2hex [-100, 100] 回転	%
s2set #3	JM の LED 表示値	1hex [1, 10] 電圧ゲージ値 1hex [1, 5] 速度モード(Profile)値	表示と同一
s2set #4	JM のボタン入力カウンタ値	1hex [0, 15] 増加ボタン 1hex [0, 15] 減少ボタン 1hex [0, 15] アラームボタン	単調増加で循環 押すと奇数、離すと偶数
s3set #1	ロータリーエンコーダのカウンタ値	2hex [0,255] 左側(1 番) 2hex [0,255] 右側(0 番)	増減して循環
s3set #2	ラジコン用プロポの入力値	2hex [-100, 100] 並進 2hex [-100, 100] 回転	%
s4set #1	汎用ピン IO の電位状態	2hex [0, 255]	8bit ピン状態(High/Low)
s4set #2	汎用ピン IO のカウンタ値	1hex [0, 15] ×最大 8 ピン分 (設定ピンのみ上位ビットから順)	単調増加で循環 High で奇数、Low で偶数

変更履歴

date	version	
2016年8月2日版	3.0	<ul style="list-style-type: none"> ・ poweroff コマンドを forcepoweroff コマンドに変更し、応答を EV:PowerOffWaitByHost から EV:PowerOffWaitByForcePowerOff へ変更 ・ forceerror コマンド(引数による挙動は未定義)と EV:PowerOffWaitByForceError を追加 ・ VR2 がエラー状態になっても電源構成 B と電源構成 D で電源の自動切断がされないようにした ・ status コマンド(typeA と typeB)を廃止 ・ poweroffdelay を追加 ・ EV: PowerOffWaitByCommunicationError を廃止 (ファームウェアにて EV:PowerOffWaitByJoystickModuleError と EV:PowerOffWaitByPowerModuleError へ分類)
2016年5月10日版	2.3	<ul style="list-style-type: none"> ・ idle コマンドの追加 ・ state コマンドの追加 ・ timestamp コマンドの追加 ・ pinmode コマンドの追加 ・ 新規ステータスコマンド群として status コマンド(typeS)と、S コマンドと、s[1-4]set コマンドの追加 ・ propo コマンドの追加 ・ 旧式ステータスコマンドである status コマンド(typeA と typeB)を利用非推奨として表記 ・ Initialized という状態を Power On Wait と呼称を変えて、EV 応答、state コマンド応答、状態遷移図を変更 ・ Power On Wait (旧 Intialized)で初期化される／されないものを分類 初期化される : VR2 関連、通信(watchdog、stablemode) 初期化されない : ビープ、エンコーダ、プロポ、汎用ピン、ステータス ・ リカバリ状態の初期値として速度モード(Profile)を 3 にする ・ 仕様書末尾の付属情報を整理
2015年9月26日版 2015年9月30日版 2015年10月27日版	2.2	<ul style="list-style-type: none"> ・ 仕様書における EV コマンドの標記順序を修正 ・ 仕様書における EV:EncoderSkip0 と EV:EncoderSkip1 を追記 (以前より実装済み) ・ 仕様書における「待機モード」→「idle モード」に変更 <p>※以下は 10月27日版にのみ適用</p> <ul style="list-style-type: none"> ・ idle モードにおいて速度モード(Profile)ランプをウェーブ状に自動切り替えにすることでユーザへ音と光で知らせる
2015年9月14日版	2.1	<ul style="list-style-type: none"> ・ newVR2-50 に対応 ・ beep コマンドのテスト機能の追加 ・ joysticklength コマンドの追加

		<ul style="list-style-type: none"> ・ リカバリ状態の再定義と ER:recovery の追加 ・ 「EV:PowerOn」 コマンドは誤記であり、「EV:PowerOnTrigger」に修正
2015年8月23日版	2.0	<ul style="list-style-type: none"> ・ 電源構成を整理して詳細に解説 ・ speedmodearea コマンドの追加 ・ joystickadjust コマンドの追加 ・ joystickback コマンドの追加 ・ status において typeB の要求と応答を追加 ・ encoderresistance コマンドの追加 ・ relayerror コマンドの追加 ・ unlockbrake コマンドの追加 ・ オプション品情報としてロータリーエンコーダを追加 ・ オプション品情報としてラジコン用プロポを追加
2015年4月30日版	1.0	<ul style="list-style-type: none"> ・ stablemode コマンドの表記ミスを修正 ・ ボーレート→ビットレートに表記を修正 ・ 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
2015年4月27日版	1.0(未販売)	<ul style="list-style-type: none"> ・ EV コマンドに「EV:PowerOffWaitByNoCommunication」を追加 ・ 「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
2015年4月13日版	1.0(未販売)	初版

(付録) ピン解説



- お客様自身で追加配線をされる場合に、ケースの開封／基板の取り出し／による故障は**有償対応**となります。お客様の責任で十分に注意しながら作業してください。**必ず初めにすべての配線を抜いて**から作業してください。基板はブザーから遠い側を斜めに持ち上げることで取り出すことができます。ケースを閉じる際は、スポンジによるケーブルの挟み込みやネジの締め過ぎに注意してください。
- 基板のバージョンによりボードレイアウトが異なります。ボード上で、上記の IC ピンと導通しているホールを見つけてください。なお、故障を招きますので未定義のピンには追加配線をしないでください。
- 5V 電源は USB の 5V 端子とボード上の導通チェックにより見つけてください。ボード上に「5」と記載があります。
- 24V 電源は VR2 から入力された直後のもの、ポリスイッチ(リセットブルヒューズ)を経由したもの、電源構成 B および C においては VR2 の電源と連動したものがあります。いずれもテストにより確認することで利用してください。

(付録) オプション品情報：2モータ2相ロータリーエンコーダ

- 利用には適切なコマンド設定が必要です。
- 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 (≒ 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。

(付録) オプション品情報：Futaba (株) 互換のラジコン用プロポ

- 利用には適切なコマンド設定が必要です。
- 3PRKA R203GF T/R セットにおいて動作確認済みです。
- 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 CH 2 のデータピン (左側) を
IO-2 (プロポ 0 化) に接続

ステアリング CH 1 のデータピン (左側) を
IO-7 (プロポ 1 化) に接続