

ICS3.5/3.6 ソフトウェアマニュアル コマンドリファレンス

©KONDO KAGAKU CO.,LTD 2016年 8月 第2版

【免責事項】

- ・このコマンドリファレンスは参考資料として公開されるものです。ご利用の際は、ご自身の責任でご使用頂きますようお願い致します。
- ・内容についての著作権など法的な権利は近藤科学株式会社にありますが、ご使用になって生じた結果については一切責任は負いません。ご理解とご了承をお願い申し上げます。
- ・誤字脱字などについては、弊社窓口までお申し出ください。
- ・内容についてのご質問、及びプログラミングについてのご質問についてはお応え致しかねますのでご了承ください。

ICS について

ICS (Interactive Communication System) は、モジュール・コントロールボード間の双方向データ通信規格です。コントロールボードでのサーボの制御時の通信や、PCなどを使用してのサーボモータの設定変更などが可能です。

ロボット用サーボモータの規格である、従来の【ICS2.0】【ICS3.0】の上位規格として、機能の追加・拡張を行ったものがICS3.5です。シリアル通信によるサーボ内の各種パラメータの変更を可能としたICSですが、新たに次の点が拡張されています。

■ICS3.0から3.5への変更点

<“シリアル” ⇔ “PWM”の切り替えが可能>

【ICS3.5】では、【ICS3.0】のシリアル制御機能はそのままに、PWM信号での制御が可能になりました。

■ICS3.5シリアル通信の主な特徴

- ・最大1.25Mbpsの高速通信が可能
- ・「スピード」や「ストレッチ」のほか、「温度制限」や「電流制限」などサーボモータの様々な特性を動作中に任意に変更可能
- ・モジュールを接続するマルチドロップ接続では最大32個接続可能(モジュール、コントロールボードの能力によって最大数が変わります)

■ICS3.6について

- ・ICS3.5の上位規格です
- ・どのタイミングでも角度を取得できるようにコマンドを追加しました
詳細は「パラメータ読み出し」をご覧ください。

■送信コマンドのループバック

送信側(PC またはマイコンなどのコントロール側)で受信されるデータは、最初に自分自身が送出したデータが受信されます。これはICS のインターフェースが送信線と受信線を共通の線で使用しているためのエコーです。ほとんどのコマンドにおいて、デバイス側は最初に受信したコマンドをそのまま返した後で、内容のデータを送出します。

■マルチドロップ接続

ICS対応デバイスはシリアル信号の送信線と受信線が共通ですので、コントロールボードのシリアル対応端子に対して1対1の接続ではなく、デバイス同士を連結して接続してもデータ通信が可能です。この連結接続をマルチドロップ接続と呼びます。

■ICS3.5対応サーボモータ

【HVサーボモータ(電源電圧:9~12V)】

- ・KRS-6003RHV ICS
- ・KRS-4034HV ICS
- ・KRS-4033HV ICS
- ・KRS-4032HV ICS
- ・KRS-4031HV ICS
- ・KRS-2572HV ICS
- ・KRS-2552RHV ICS
- ・KRS-2542HV ICS

【6Vサーボモータ(電源電圧:6~7.4V)】

- ・KRS-3304 ICS
- ・KRS-3204 ICS

いずれのタイプも信号線の電圧仕様は同じです。

■ICS3.6対応サーボモータ

【6Vサーボモータ(電源電圧:6~7.4V)】

- ・KRS-3301 ICS

※KRS-3301は、「温度」「電流値」を取得することはできません。

■各機能の紹介

※各パラメータの初期設定値は、サーボにより異なります。初期値につきましては、KRSサーボシリーズ取扱説明書をご覧ください。

【ID】

サーボにID番号を設定します。

設定範囲 0～31

【通信速度】

サーボモータとの通信速度を設定します。

設定範囲 115200bps, 625000bps, 1.25Mbps

【ストレッチ】

サーボモータの保持特性を変更します。
数値を下げるごとに、保持力が弱まりばねのように柔らかくなります。

設定範囲 (Soft) 1 ～ 127 (Hard)

ストレッチ (SET1)(SET2)(SET3)はHeartToHeart3のキャラクタースティックチェンジで使用する値です。このパラメータ範囲は上記と同じです。

【スピード】

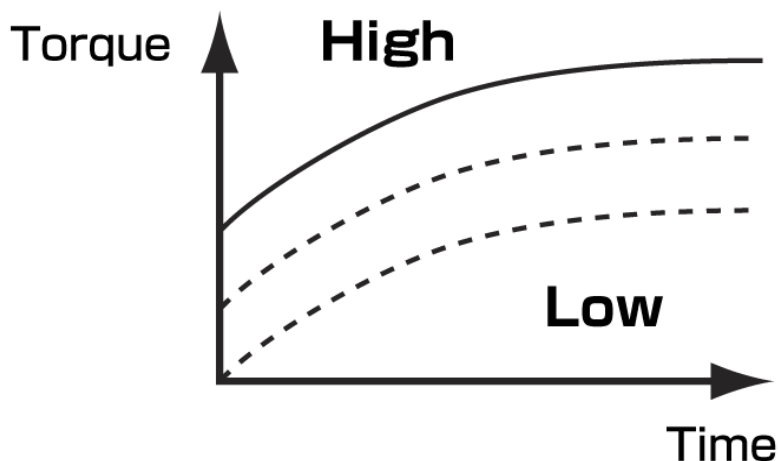
サーボモータの最高回転速度を設定します。

設定範囲 (Slow) 1 ～ 127 (Fast)

【パンチ】

サーボモータ初動のトルクオフセットを設定します。
数値を上げるごとに動き出しから瞬時にパワーを出すことができます。

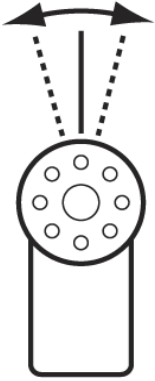
設定範囲 (Low) 0 ～ 10 (High)



【デッドバンド】

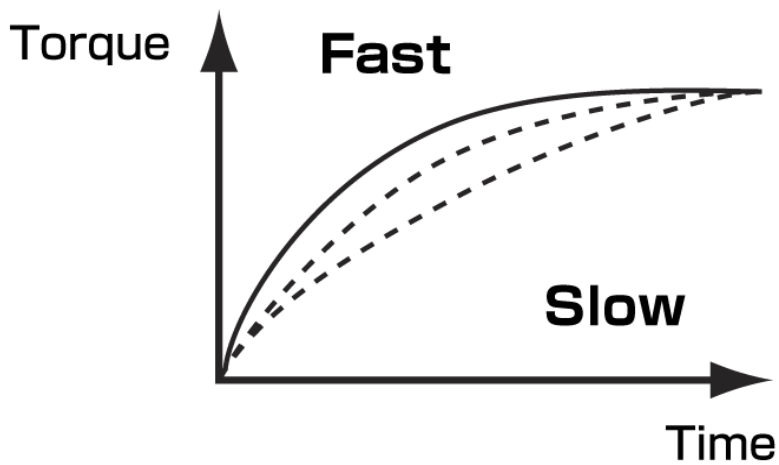
サーボモータのニュートラル帯域(不感帯)の範囲を設定します。数値が上がるほど範囲が広くなり、ニュートラル位置があいまいになります。サーボがハンチングする場合にニュートラル範囲を広げることで止めることができます。

設定範囲 (Min) 0 ~ 10 (Max)

**【レスポンス】**

出力軸の動作開始時の立ち上がり特性を設定します。数値が小さいほど動き出しがなめらかになります。

設定範囲 (Slow) 1 ~ 5 (Fast)

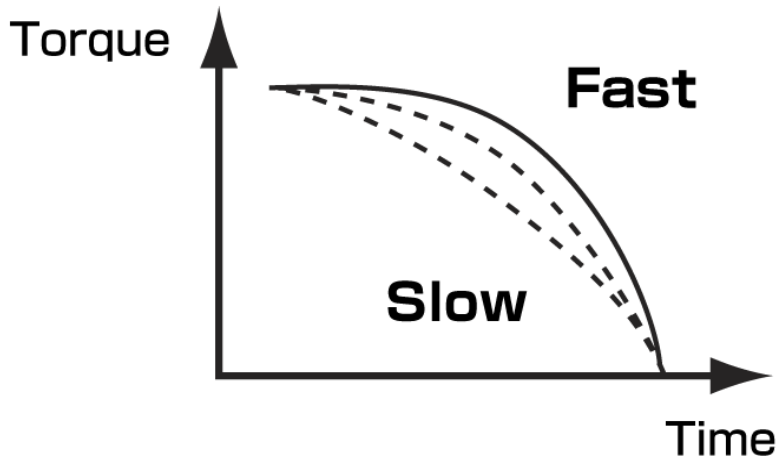


【ダンピング】

出力軸の動作停止時のブレーキ特性を設定します。

数値が小さいほどブレーキが早くかかるようになり、停止までの動きがなめらかになります。

設定範囲 (Slow) 1 ~ 255 (Fast)

**【プロテクション】**

プロテクション動作開始までの時間を設定します。

プロテクション機能は、ロックした場合の保護機能です。起動後はサーボのパワーを自動的に50%ダウンさせます。復帰後は、ロックなどを解消した時点で自動的に行われます。

なお、プロテクションはサーボのスピードのパラメータが127に設定されている時にのみ有効な機能です。

設定範囲 (Short) 10 ~ 255 (Long)

パラメータ1あたりの時間
約0.056sec

※使用状況、機種によっては動作開始までの時間が前後する場合があります。

【リミッタ】

サーボモータの最大動作角の範囲を設定します。

設定範囲 正転 (Min) 8000 ~ 11500 (Max)
逆転 (Min) 3500 ~ 7500 (Max)

【温度制限】

サーボモータ本体の温度の閾値を設定します。

サーボの基板に実装されたセンサが、設定した温度より高い値が出力された場合、サーボが脱力状態になります。閾値を上回れば復帰します。

設定範囲 (High) 1 ~ 127 (Low)

温度	設定値
100°C	30
90°C	47
80°C	60
70°C	75
60°C	87

※これらの値は目安です。状況により実際の動作とは異なる場合があります。

【電流制限】

電流の閾値を設定します。

サーボの基板に実装されたセンサが、設定した値より高い電流を検知した場合、サーボが脱力状態になります。閾値を下回れば復帰します。

設定範囲 正転 (Low) 0 ~ 63 (High)
逆転 (Low) 64 ~ 127 (High)

電流値	設定値
0A	0
0.1A	1
0.5A	5
1.0A	10
1.5A	15
2.0A	20

※これらの値は目安です。状況により実際の動作とは異なる場合があります。

【フラグ】

リバース、シリアル専用(PWMINH)、スレーブ、回転モードの使用を選択します。

名称	機能
リバース	設定すると指定と反対方向へ動作します。
シリアル専用(PWMINH)	PWMモードでの動作を禁止します。
スレーブ	送信コマンドの返信コマンドを返さないようにします。
回転モード	設定すると出力軸が無限回転します。

▼ダブルサーボについて

ダブルサーボとは、2つのサーボを背中合わせに組み付けて使用する状態をさします。両軸をそれぞれのサーボにすることでトルクを約2倍にすることが可能です。リバースとスレーブを設定することでダブルサーボとして使用することができます。

それぞれのサーボ同士をつなげ同じIDに設定し、同じコマンドを受け取って動作します。

ただし、サーボはコマンドを受け取ると返信コマンドを発信するため、2つのサーボが同時に返信コマンドを発信することになり、通信が混信してしまいます。そこで、片方のサーボを「スレーブ」に設定し返信コマンドを禁止にすることによって、同じコマンドを受信しても混信をしないようにすることができます。

また、背中合わせに組み付けるため、同じコマンドではそれぞれが逆方向に動作しますが、スレーブにしたサーボに「リバース」に設定して片方を逆方向に動かすことにより、同じ方向へ動作させることでサーボアームなどを組み付けることが可能になります。

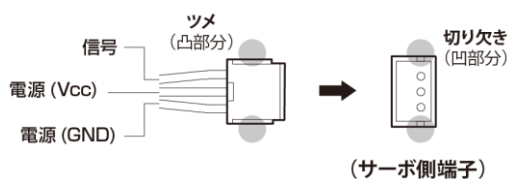
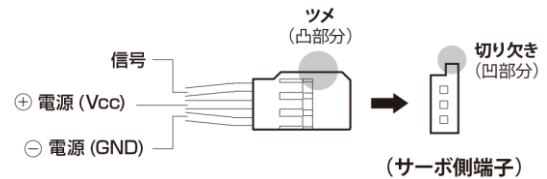


ダブルサーボ使用例
サーボ: KRS-6000シリーズ

【ユーザーオフセット】

出力軸の初期位置をユーザーが任意に設定できます。

設定範囲 (逆転) -127 ~ 127 (正転)

■コネクタの極性について**ZHコネクタ****サーボコネクタ**

通信を行う前に

■PCから通信する場合

USBアダプターを使用することで、PCから直接サーボを制御することができます。

用意するもの

●USBアダプター

- ・Dual USBアダプターHS (No.02116)
- ・ICS USBアダプターHS (No.02042)
- ・ICS USBアダプター (No.01106)

※ 商品名のHSはHigh Speedの略です。最大1250000bps(1.25Mbps)の高速通信が可能です。
 ※ ICS USBアダプター(No.01106)は高速通信に対応していないため115200bps以外は使用できません。
 ※ Dual USBアダプターHSのセットに必要なケーブルが全て含まれています。

●電源

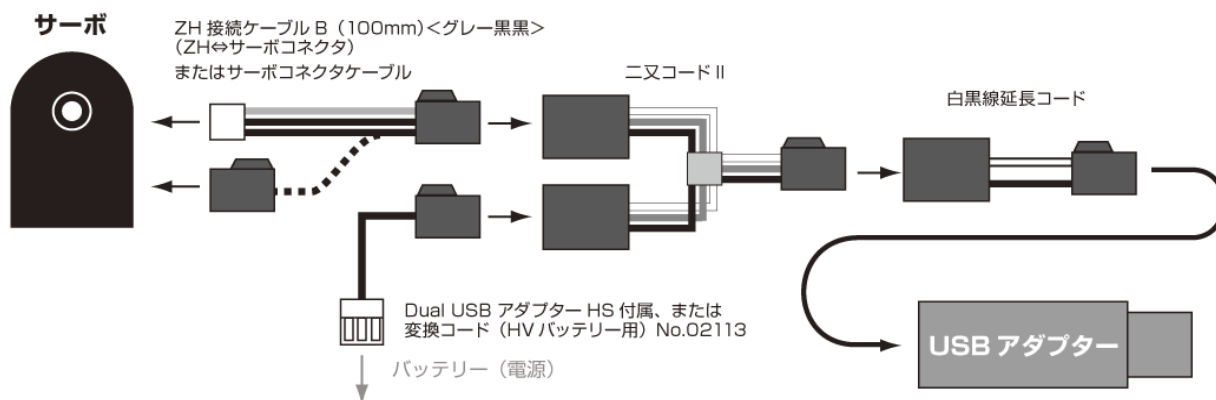
【HVサーボ用】

- ROBOパワーセル F3-850タイプ(Li-Fe)
- ROBOパワーセル F3-1450タイプ(Li-Fe)
- ROBOパワーセル F3-2100タイプ(Li-Fe)
- ROBOパワーセル HV Cタイプ 9N-300mAh Ni-MH
- ROBOパワーセル HV Dタイプ 9N-800mAh Ni-MH

【6Vサーボ用】

- ROBOパワーセル F2-850タイプ(Li-Fe)
- ROBOパワーセル F2-1450タイプ(Li-Fe)

接続方法



※ ドライバーのインストール方法などは、KO Driver付属のマニュアルをご覧ください。

▼【KONDO website】カスタマーセンター→サポート情報→ソフトウェア→「KO Driver2015」

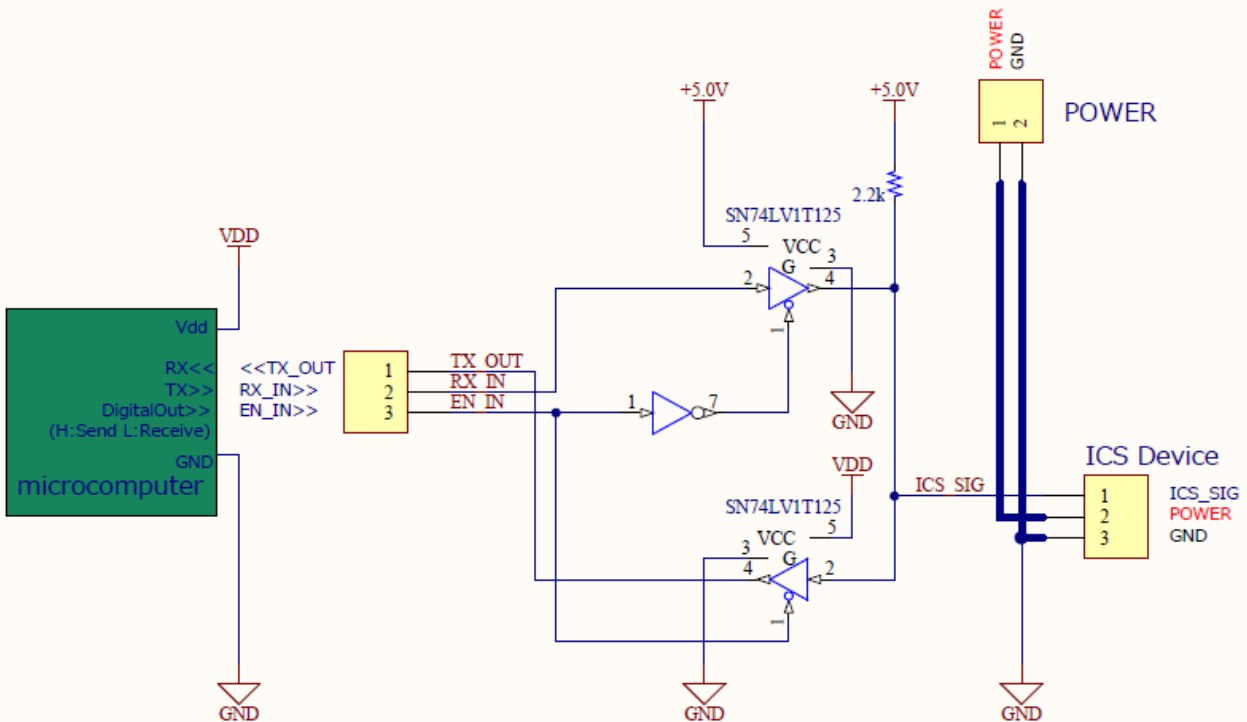
<http://kondo-robot.com/faq/ko-driver-2015>

■お手持ちのボードからUARTにて通信する場合

UART端子を持っているマイコンよりICS機器(KRSサーボ)を動作させる際は、以下の回路をご用意ください。

回路図のICS Deviceにサーボや受信機を接続します。各番号は、1番が信号線、2番が電源線、3番がグランド線となっています。弊社製ロボット用シリアルサーボモータは、電源電圧は9～12V対応品と6～7.8V対応品の2種類あります。通信は、1本の通信線を送信、受信を切り替えて使います。

第1版の回路では、5.0Vで駆動しているマイコンでしか使用できませんでしたが、3.3V以下で駆動するマイコンが増えてきたため、それに対応した回路になっています。



※ロジックICのバイパスコンデンサは省略しています

●回路について

送信側の回路の影響を与えないため、3ステートバッファを用います。今回用いる3ステートバッファは、制御ピンをLにすると信号を通し、HIにすると電氣的に出力が切り離されるものになります。また、ICSの信号は5Vの信号のため、低電圧のマイコンでは直接信号を受けることができません。3ステートバッファはその電圧の変換にも使用しています。

ICSでは、同時に送信、受信をすることができません。送信側の3ステートバッファの制御ピンにNOT回路を入れることにより、受信と逆の制御ができるようになりますので、EN_IN信号一つで送信受信を切り替えられるようになります。

●動作説明

コマンドを送信するときはEN_INをHIにします。

ICS機器からの信号を受信するときはEN_INをLIにします。送信をしない場合はEN_INのピンを常にLIにしてください。

※ICS機器からデータが返ってきている最中に次のコマンドを送らないでください

※この回路では、コールバックのデータが返ってきません

●通信手順

(初期: EN_INをLにします)

通信する時信号が流れてないのを確認し、EN_INをHにします。

↓

マイコンのTXからコマンドを送信します。

↓

TXからコマンドを送信終了後すぐにEN_INをLにします。

(ICS機器から信号が送られてくるのですぐ受信に切り替えます)

↓

マイコンのRXよりデータを受信します。

●ICS変換アダプターについて

上記で紹介した回路を製品化します。この変換基板にお手持ちのマイコンのUARTの各端子を接続するだけで、回路を用意することなくサーボとシリアル通信を行うことができます。

【商品情報】

ICS変換基板

No.03121 価格:未定

※2016年秋発売予定

シリアル通信設定

通信速度	115200bps, 625000bps, 1.25Mbps
ビット長	8bit
スタート	1bit
ストップ	1bit
フロー制御	無し
パリティ	EVEN(偶数)
極性	反転なし
信号レベル	5V TTL

シリアル動作設定

サーボをシリアル通信させるには、電源投入時に信号線を500ms間、Hレベルに保つことで通信が可能になります。ただし、電源の瞬断が起きた際にPWMモードに切り替わることを防ぐため、ロボットなどでの実動時には”PWMINH”フラグを1にしておくことをお勧めします。(フラグの詳細はEEPROM項目参照)

データ構造

バイト数	1 (CMD)	2 (SC)	3~N-1 (DATA)
内容	コマンドヘッダ (メインコマンド) +ID番号	サブコマンド	データ

■コマンドヘッダ (CMD)

コマンドヘッダ (CMD) 部分は、メインコマンド4種類(ポジション、読み出し、書き込み、ID設定)とサーボモータにセットされたID番号を連結したものです。下記のメインコマンド一覧にある数値で、「0b」がついているものは2進数での表記となっています。またこれ以降「0x」とついているものは16進数表記です。

	CMD (1BYTE)							
	コマンド			ID				
	7bit	6bit	5bit	4bit	3bit	2bit	1bit	0bit
ポジション (サーボモータの回転 角度を決める)	1	0	0	x	x	x	x	x
読み出し (パラメータを読み 出す。種類はサブコ マンドで決める)	1	0	1	x	x	x	x	x
書き込み (パラメータを書き 込む。種類はサブコ マンドで決める)	1	1	0	x	x	x	x	x
ID (ID番号の読み出し、 または書き出しを行 う)	1	1	1	x	x	x	x	x

XXXXXには5ビットのID番号が入ります。例えばID=12(5ビットの2進数で01100)のサーボにポジションをセットする場合は、ICSコマンド第1バイト目(コマンドヘッダ、CMD)は「0b10001100」となります。

■サブコマンド(SC)

サブコマンド(SC)は、メインコマンドのオプション設定で、サーボモータのスピード、ストレッチ、電流値などがあります。ポジションコマンドにはサブコマンドはありません。

※1はICS3.5以降のサブコマンドです。

※2はICS3.6限定の機能です。

EEPROM	0x00	サーボモータのROMデータに直接アクセスする
STRC	0x01	ストレッチデータを扱う
SPD	0x02	スピードデータを扱う
CUR ※1	0x03	電流値を読み出す、または電流制限値を書き込む
TMP ※1	0x04	サーボモータの温度値を読み出す、または温度制限値を書き込む
TCH ※2	0x05	サーボモータの現在値を読み出します (書き込みには対応していません)

EEPROMの中身の詳細はp.21をご参照ください。

■データ(DATA)

データ(DATA)は読み出しの場合は指定しません。書き込みの場合にサーボモータに書き込みたいデータを指定します。ポジションコマンドのデータ部分だけは特殊なデータ構造になります。下記データ構造を作る方法は「ポジション設定」をご覧ください。

POS_H	ポジションデータの上位7ビット
POS_L	ポジションデータの下位7ビット

※コマンドヘッダ以外はMSBを0するよう構成されていますが、IDの書き込み時のみ例外となっています。

コマンド一覧

これから解説する表は以下の構成になっています。

TX or RX	BYTE1	BYTE2	BYTE3	…
	内容	内容	内容	

TX = コマンド送信

RX = コマンド受信

ポジション設定

機能

○ポジション設定コマンド

角度の指定によりサーボを動作させることができます。

サーボモータのポジションデータに指定できる値は3500～11500となります。7500がニュートラル位置です。ポジションデータに0を入れたときだけ特殊操作となり、サーボモータが脱力(Free)します。

構成

TX	1	2	3
	CMD	POS_H	POS_L

CMD ポジション設定コマンド
POS_H / POS_L サーボの設定舵角

RX	1	2	3	4	5	6
	送信コマンドのループバック			R_CMD	TCH_H	TCH_L

TCH_H / TCH_L 現在のサーボの角度

解説

MSB		CMD					LSB
7bit	6bit	5bit	4bit	3bit	2bit	1bit	0bit
1	0	0	x	x	x	x	x
ポジション設定コマンド #100xxxxxb (xxxxxはID番号)			サーボのID 0x00(0) ~ 0x1F(31)				

MSB		R_CMD					LSB
7bit	6bit	5bit	4bit	3bit	2bit	1bit	0bit
0	0	0	x	x	x	x	x
ポジション返信コマンド #000xxxxxb (xxxxxはID番号)			サーボのID 0x00(0) ~ 0x1F(31)				

※サーボからの返事ではホストからのコマンドと間違わないように

CMDのMSBをマスクして返ってきます。

※従来のICS2.0との互換性のため、ID0で通信速度が115.2Kの場合、RXのMSBは1になります。
ただし、こちらはポジション設定コマンドの場合のみです。

MSB POS_H LSB							
7bit	6bit	5bit	4bit	3bit	2bit	1bit	0bit
0	x	x	x	x	x	x	x
0固定	設定舵角(上位7bit)						

MSB POS_L LSB							
7bit	6bit	5bit	4bit	3bit	2bit	1bit	0bit
0	x	x	x	x	x	x	x
0固定	設定舵角(下位7bit)						

MSB TCH_H LSB							
7bit	6bit	5bit	4bit	3bit	2bit	1bit	0bit
0	x	x	x	x	x	x	x
0固定	現在角度(上位7bit)						

MSB TCH_L LSB							
7bit	6bit	5bit	4bit	3bit	2bit	1bit	0bit
0	x	x	x	x	x	x	x
0固定	現在角度(下位7bit)						

サーボの舵角の範囲は0 ~ 16383 です。ただし、舵角270° のサーボの設定範囲は3500~115000、センターは7500になります。

コマンドヘッダ以外はMSBを0にしなければならない約束がありますので、2バイト(16ビット)データのうち、下位14ビットのみを使用します。取り出した14ビットを2分割して、上位をPOS_H、下位をPOS_Lとします。

例えばニュートラル位置7500の場合は、

7500=0b00011101_01001100(0b[00][011101_0][1001100])となりますので、
 POS_H=0b000111010=0x3A、POS_L=0b01001100=0x4Cとなります。

返値は現在位置データが送信コマンドと同じように、7ビットずつに分解されて返ってきますので、プログラムでデータを戻すには7ビットずつ取り出してつなぎ合わせる必要があります。

舵角に0x00 をセットすると、サーボがFree になります。

シリアルモードでは、従来のPWM 信号の場合のようなポジションキャプチャー専用のコマンドがありません。軸の現在の位置(角度)は、動作位置を指定した際の戻り値として値が返ってきます。現在の位置が不定の場合には、一旦動作位置を”0(Free)”で指定し、位置を取得してから希望の位置に移動することで安全な起動時の動作が可能です。

ICS3.6対応のサーボモータは角度取得コマンドを追加しましたのでそちらをお使いください

例

ID=1のサーボモータのポジションを7500にする送信コマンド

TX	1	2	3
	CMD 0x81	POS_H 0x3A	POS_L 0x4C

16ビットデータの下位14ビットを7ビットずつ分解して、上位をPOS_H、下位をPOS_Lとする

RX	1	2	3	4	5	6
	送信コマンドのループバック			R_CMD 0x01	TCH_H 0x3A	TCH_L 0x4C

5,6のデータは現在の位置が戻る

※サーボからの返事ではホストからのコマンドと間違わないように
CMDのMSBをマスクして返ってきます。
(IDが0で通信速度が115.2Kbps以外の場合)

ここで「送信コマンドのループバック」とは、自分で送信した3バイトのデータがそのまま返ってくることを示しています。ICS規格では送信線と受信線が同じであるため、データを送ると同時に受信してしまうからです。普通のマイコンでは送信時に受信はできません(フロー制御がないとき)ので、ループバックは無視して3バイトのみの受信でかまわないのですが、パソコンなどのOSが搭載されているものでは、ループバックはバッファに自動的に保存されますので、サーボモータから来た返事は6バイト受け取るようにしてください。なお4バイト目のサーボモータから返ってきたCMDはMSBが0となっています。この例では送信CMD=0x81に対して、R_CMD=01が返ります。ただし、ID=0で通信速度が115.2kbpsの場合のみ、CMDのMSBはマスクせず返ってきます。これは、ポジションコマンドのみの仕様です。

パラメータ読み出し

機能

○パラメータ読み出しコマンド

各種設定値を読み出すことができます。読み出せるデータは、スピード、ストレッチ、現在の電流値と温度値、EEPROMのデータです。

○電流値の読み出しについて

電流値の読取コマンドでは現在の電流値と方向が読み取れます。

正方向では電流値が0から63まで、逆方向では64から127として読み込まれます。これは逆方向のときに6bit目が1になるためです。

○温度値の読み出しについて

温度値の読取コマンドでは現在の温度値が読み取れます。

温度パラメータは0から127までの値で、小さいほど温度が高いことを示します。目安として、パラメータ60で温度が約80度、パラメータ30で約100度です。

○角度読み出しについて

ICS3.6からの機能です。今までポジションコマンドを使用して角度を取得しましたが、Freeのポジション送ったり現在の指令値を送らないといけない等、動かす時不都合な場合があります。今回実装されたコマンドを使用することで、動作を止めることなく角度を読み出すことが可能になりました。

値はポジションコマンドで送る角度データと同じフォーマットで返事が返ってきます。

構成

TX	1	2
	CMD	SC

EEPROM

RX	1	2	3	4	5から68
	送信コマンドのループバック		R_CMD	SC	EEPROM 64bytes

ストレッチ

RX	1	2	3	4	5
	送信コマンドのループバック		R_CMD	SC	STRC

スピード

RX	1	2	3	4	5
	送信コマンドのループバック		R_CMD	SC	SPD

電流

RX	1	2	3	4	5
	送信コマンドのループバック		R_CMD	SC	CUR

温度

RX	1	2	3	4	5
	送信コマンドのループバック		R_CMD	SC	TMP

角度 (ICS3.6限定機能)

RX	1	2	3	4	5	6
	送信コマンドのループバック		R_CMD	SC	THC_H	THC_L

解説

MSB		CMD						LSB
7bit	6bit	5bit	4bit	3bit	2bit	1bit	0bit	
1	0	1	x	x	x	x	x	
パラメータ読み出しコマンド #101xxxxxb (xxxxxはID番号)			サーボのID 0x00(0) ~ 0x1F(31)					

MSB		R_CMD						LSB
7bit	6bit	5bit	4bit	3bit	2bit	1bit	0bit	
0	0	1	x	x	x	x	x	
読み出し返信コマンド #001xxxxxb (xxxxxはID番号)			サーボのID 0x00(0) ~ 0x1F(31)					

※サーボからの返事ではホストからのコマンドと間違わないように
CMDのMSBをマスクして返ってきます。

SC	
EEPROM	0x00
ストレッチ	0x01
スピード	0x02
電流値	0x03
温度値	0x04
角度値	0x05

(ICS3.6限定機能)

DATA	
EEPROM	EEPROM データ参照
STRC	ストレッチデータ 1(2) ~ 127(254) カッコ内の数値はEEPROM の設定値
SPD	スピードデータ 1(1) ~ 127(127) カッコ内の数値はEEPROM の設定値
CUR	電流値 正転時0 ~ 63,逆転時64~127
TMP	温度値 1 ~ 127
TCH_H TCH_L	現在の角度値 3500~7500~11500 (7500がニュートラル地点です)

例

ID=1のサーボモータのストレッチデータを読み出す送信コマンド

TX	1	2
	CMD 0xA1	SC 0x01

RX	1	2	3	4	5
	送信コマンドのループバック		R_CMD 0x21	SC 0x01	STRC 0x1E

サーボからの返事ではホストからのコマンドと間違わないように
CMDのMSBをマスクして返ってきます。ここでは0xA1が0x21になっています。
※ストレッチが30(0x1E)になっている場合

パラメータ書き込み

機能

○パラメータ書き込みコマンド

各設定値を書き換え(変更)することができます。書き込めるデータはスピード、ストレッチ、電流制限値、温度制限値、EEPROMのデータです。

○ストレッチ、スピード、電流制限値、温度制限値、通信速度の設定について

書き込みコマンドを使用すると、EEPROMに保存されているスピード、ストレッチ、電流制限値、温度制限値の内容も変更されます。

その他通信速度等を書き込む場合は専用のEEPROMの書き込みコマンドを用いて一括で書き換えてください。IDについては専用コマンドがあります。「IDコマンド」の章を参照してください。

※ICS USBアダプター(No.01106)ご使用上の注意

通信速度のパラメータを書き換える際は充分気をつけて書き換えてください。一度書き換えてしまうと次回からは書き換えたスピードでの通信が必要になるため、625kbps,1.25Mbpsの高速通信に変更した場合、高速通信非対応の製品(ICS USBアダプター(No.01106)など)でデバイスとの通信ができなくなります。

構成

EEPROM

TX	1	2	3から66
	CMD	SC	EEPROM 64bytes

RX	1	...	66	67	68
	送信コマンドのループバック			R_CMD	SC

ストレッチ

TX	1	2	3
	CMD	SC	STRC

RX	1	2	3	4	5	6
	送信コマンドのループバック			R_CMD	SC	STRC

スピード

TX	1	2	3
	CMD	SC	SPD

RX	1	2	3	4	5	6
	送信コマンドのループバック			R_CMD	SC	SPD

電流制限値

TX	1	2	3
	CMD	SC	CURLIM

RX	1	2	3	4	5	6
	送信コマンドのループバック			R_CMD	SC	CURLIM

温度制限値

TX	1	2	3
	CMD	SC	TMPLIM

RX	1	2	3	4	5	6
	送信コマンドのループバック			R_CMD	SC	TMPLIM

解説

MSB		CMD					LSB
7bit	6bit	5bit	4bit	3bit	2bit	1bit	0bit
1	1	0	x	x	x	x	x
パラメータ書き込みコマンド #110xxxxxb (xxxxxはID番号)			サーボのID 0x00(0) ~ 0x1F(31)				

MSB		R_CMD					LSB
7bit	6bit	5bit	4bit	3bit	2bit	1bit	0bit
0	1	0	x	x	x	x	x
書き込み返信コマンド #010xxxxxb (xxxxxはID番号)			サーボのID 0x00(0) ~ 0x1F(31)				

※サーボからの返事ではホストからのコマンドと間違わないようにCMDのMSBをマスクして返ってきます。

SC	
EEPROM	0x00
ストレッチ	0x01
スピード	0x02
電流値	0x03
温度値	0x04

DATA	
EEPROM	EEPROM データ参照
STRC	ストレッチデータ 1(2) ~ 127(254) カッコ内の数値はEEPROM の設定値
SPD	スピードデータ 1(1) ~ 127(127) カッコ内の数値はEEPROM の設定値
CUR	電流制限値 1(1) ~ 63(63) カッコ内の数値はEEPROM の設定値
TMP	温度上限値 1(1) ~ 127(127) カッコ内の数値はEEPROM の設定値

例

ID=10のサーボモータにスピードとして100を書き込む送信コマンド

例として、ID番号が10のサーボモータのスピードを変更するICSコマンドを作成してみます。リファレンスマニュアルから、設定できるスピード値は0~127(数値が小さいほど速度が下がる)であるので、ここでは100と設定します。スピードをサーボモータにセットするには書き込みコマンド(0b110XXXXX)を使います。

ID番号は10(2進数で0b00001010)です。

スピード設定サブコマンドは2です。

メインコマンドが「0b110XXXXX」で、ID番号は「0b00001010」ですので、CMDは11001010=202(0xCA)となります。

TX	1	2	3
	CMD 0xCA	SC 0x02	SPD 0x64

スピード変更などの書き込みコマンド実行時には、サーボモータから次のような返事が来ます。

RX	1	2	3	4	5	6
	送信コマンドのループバック			R_CMD 0x4A	SC 0x02	SPD 0x64

サーボからの返事ではホストからのコマンドと間違わないようにMSBをマスクして返ってきます。ここでは0xC2が0x42になっています。

ID コマンド

機能

○シリアルサーボのID を読み書きできます。

構成

TX	1	2	3	4	
	CMD	SC	SC	SC	
RX	1	2	3	4	5
	送信コマンドのループバック				R_CMD

解説

MSB							CMD	LSB	
7bit	6bit	5bit	4bit	3bit	2bit	1bit	0bit		
1	1	1	x	x	x	x	x		
ID設定コマンド #111xxxxb (xxxxxはID番号)			ID読込=#11111b					ID書込=書き込むID をセットする 0x00(0) ~0x1F(31)	

MSB							R_CMD	LSB	
7bit	6bit	5bit	4bit	3bit	2bit	1bit	0bit		
1	1	1	x	x	x	x	x		
ID設定コマンド #111xxxxb (xxxxxはID番号)			ID読込=現在設定してあるID がセットされる。					ID書込=送ったときのID がセットされる。	

IDコマンドのみサーボからの返事でもMSBのマスクはありません。

SC	
読込	0x00
書込	0x01

ID コマンドを使用する際は、必ず送り側とシリアルサーボを1対1で接続すること！

マルチドロップ接続になっているデバイスにIDコマンドを送ると、全てのデバイスがコマンドに対して返事をしてしまうため信号が混ざり合い、不正なデータとなります。またIDを書き込んだ場合は、全てのデバイスが同じIDになってしまいます。

例

1対1で接続されたサーボモータのIDを読み出すコマンド

ID番号を読み出すコマンドは他のコマンドと構造が違います。サーボからIDを読み取る時には、CMD=0xFFとしてください(※IDが分からないサーボモータからも読み出すため)。SCは0固定で、3回繰り返します。

TX	1	2	3	4
	CMD 0xFF	SC 0x00	SC 0x00	SC 0x00

サーボモータからは下のようなデータが返ってきます(PCで受信した場合のみ5バイト、そうでないなら1バイトのみ返る)。

RX	1	2	3	4	5
	送信コマンドのループバック				R_CMD 0xF4

R_CMDには、ID読み出しコマンド0b111XXXXXとサーボモータのID番号0b000XXXXXが合わさった1バイトのデータで、MSBが0になったものが返ります。例えば読み出したサーボモータのID番号が25(0b000110011)だった場合は、R_CMD=0b011110011(243=0xF3)が返ってきます。

1対1で接続されたサーボモータのIDを20(0x14)にする送信コマンド

TX	1	2	3	4
	CMD 0xF4	SC 0x01	SC 0x01	SC 0x01

RX	1	2	3	4	5
	送信コマンドのループバック				R_CMD 0xF4

IDコマンドのみサーボからの返事でもMSBのマスクはありません。

EEPROMデータ

機能

BYTE	設定範囲	出荷時値例	
1	0x5A固定	0x5A	バックアップキャラクタ 上位4bit ここは書き換えてはいけません
2			バックアップキャラクタ 下位4bit ここは書き換えてはいけません
3	2,4...254 偶数のみ2step	60	ストレッチゲイン 上位4bit
4	(0x02~0x100)		ストレッチゲイン 下位4bit
5	1,2,3...127	127	スピード 上位4bit
6	(0x01~0x7F)		スピード 下位4bit
7	0,1,2,3...10	1	パンチ 上位4bit
8	(0x00~0x0A)		パンチ 下位4bit
9	0,1,2,3,4,5	2	デッドバンド 上位4bit
10	(0x00~0x05)		デッドバンド 下位4bit
11	1,2...255	40	ダンピング 上位4bit
12	(0x01~0xFF)		ダンピング 下位4bit
13	10,11...255	250	セイフタイマー 上位4bit
14	(0x01~0xFF)		セイフタイマー 下位4bit
15	「フラグ詳細」参照	0	フラグ 上位4bit 「フラグ詳細」参照
16			フラグ 下位4bit 「フラグ詳細」参照
17			パルスリミット上限 上位バイト上位4bit
18	3500...11500	11500	パルスリミット上限 上位バイト下位4bit
19	(0xDAC~0x2CEC)		パルスリミット上限 下位バイト上位4bit
20			パルスリミット上限 下位バイト下位4bit
21			パルスリミット下限 上位バイト上位4bit
22	3500...11500	3500	パルスリミット下限 上位バイト下位4bit
23	(0xDAC~0x2CEC)		パルスリミット下限 下位バイト上位4bit
24			パルスリミット下限 下位バイト下位4bit
25	変更禁止	変更禁止	出荷時に固体の補正データを書き込んでいます。
26			実際には、読み出したデータをそのまま書き込んでください。
27	0,1,10	10	通信速度 上位4bit 10 = 115200bps, 1 = 625000bps, 0 = 1.25Mbps
28	(0x00 / 0x01 / 0x10)		通信速度 下位4bit
29	1,2...127	80	温度制限 上位4bit
30	(0x01~0x7F)		温度制限 下位4bit
31	1,2...63	63	電流制限 上位4bit
32	(0x01~0x3F)		電流制限 下位4bit
33	変更禁止	変更禁止	出荷時に固体の補正データを書き込んでいます。
34			実際には、読み出したデータをそのまま書き込んでください。
35	変更禁止	変更禁止	出荷時に固体の補正データを書き込んでいます。
36			実際には、読み出したデータをそのまま書き込んでください。
37	変更禁止	変更禁止	出荷時に固体の補正データを書き込んでいます。
38			実際には、読み出したデータをそのまま書き込んでください。
39	変更禁止	変更禁止	出荷時に固体の補正データを書き込んでいます。
40			実際には、読み出したデータをそのまま書き込んでください。
41	変更禁止	変更禁止	出荷時に固体の補正データを書き込んでいます。
42			実際には、読み出したデータをそのまま書き込んでください。
43	変更禁止	変更禁止	出荷時に固体の補正データを書き込んでいます。
44			実際には、読み出したデータをそのまま書き込んでください。
45	変更禁止	変更禁止	出荷時に固体の補正データを書き込んでいます。
46			実際には、読み出したデータをそのまま書き込んでください。
47	変更禁止	変更禁止	出荷時に固体の補正データを書き込んでいます。
48			実際には、読み出したデータをそのまま書き込んでください。
49	変更禁止	変更禁止	出荷時に固体の補正データを書き込んでいます。
50			実際には、読み出したデータをそのまま書き込んでください。
51	1,2,3,4,5	3	レスポンス上位4bit レスポンスの値が大きいほど鋭く立ち上がります。
52	(0x01~0x05)		レスポンス下位4bit
53	0±127	0	ユーザーオフセット上位バイト上位4bit 0が中央です。
54	(-0x7F~0x7F)		ユーザーオフセット上位バイト下位4bit
55	変更禁止	変更禁止	出荷時に固体の補正データを書き込んでいます。
56			実際には、読み出したデータをそのまま書き込んでください。
57	0...31	0	ID 上位4bit
58	(0x00~0x31)		ID 下位4bit
59	2,4...254 偶数のみ2step	120	キャラクタリストティックチェンジのストレッチ1
60	(0x02~0x100)		
61	2,4...254 偶数のみ2step	60	キャラクタリストティックチェンジのストレッチ2
62	(0x02~0x100)		
63	2,4...254 偶数のみ2step	254	キャラクタリストティックチェンジのストレッチ3
64	(0x02~0x100)		

※注意 EEPROMの3, 4番、59から64番のストレッチゲインに直接書き込む時は2から254までの偶数を書きこんでください

フラグの詳細

フラグ 上位4bit							MSB	LSB
7bit	6bit	5bit	4bit	3bit	2bit	1bit	0bit	
0	0	0	0	x	0	0	x	
データなし				スレーブ	データなし		回転モード	

フラグ 下位4bit							MSB	LSB
7bit	6bit	5bit	4bit	3bit	2bit	1bit	0bit	
0	0	0	0	x	1	x	x	
データなし				PWMINH	1固定	FREE	リバース	

機能	フラグ
リバース	0 OFF / 1 ON
FREE	0 OFF / 1 ON 読み出し参照のみ可
1固定	1以外への書き込み禁止
PWMINH	シリアルで使用するときON。
回転モード	0 OFF / 1 ON
スレーブモード	0 OFF / 1 ON

※各機能の詳細は「各機能の紹介」p.7【フラグ】を参照してください。

ユーザーオフセットの設定について

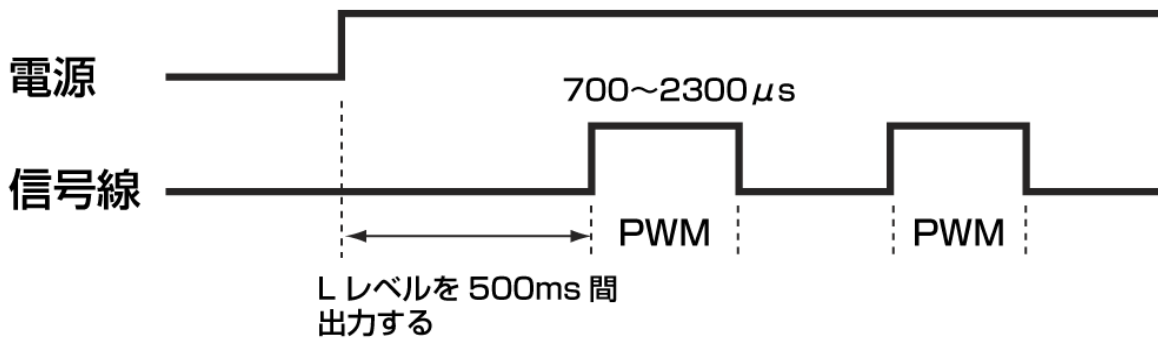
0が中央です。正方向は1,2,3,...,127までです。負の数値は255, 254,,253,,,,,128までです。例として、+1する時は1を+127する時は127を設定します。また、-1する時は255を、-127する時は129を設定します。

※ユーザーオフセットは入力された制御数値に対して出力軸を微調整する機能です。出荷時には調整されておりますので、通常は変える必要はありません。

PWMで使用する場合

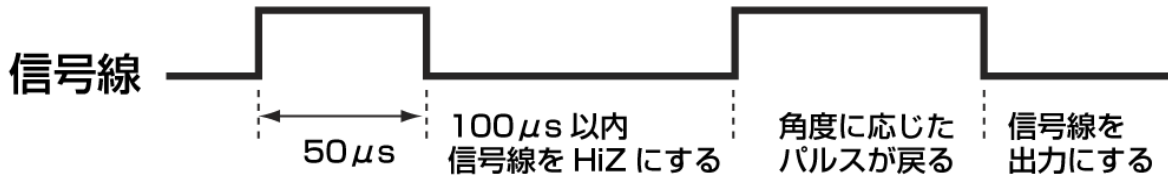
PWM制御方法

PWMで動作させる時はPWMINHフラグを0にして、電源投入時に500ms間、信号線をLレベルにします。PWMの範囲は700usから2300usでサーボの動作角度は270度です。ニュートラルは1500usです。



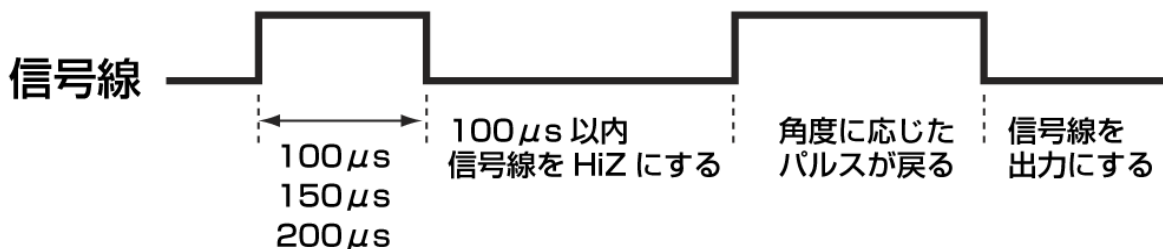
PWMでの教示機能

幅が50us \pm 5usのパルスを入力するとサーボは脱力して現在の出力角度をパルス幅に変換して返します。返ってくるパルスを取得するには、50usのパルスを出してから100us以内に信号線をハイインピーダンスにします。返ってくるパルスを取得後、信号線を出力に戻します。



PWMでのキャラクタースティックチェンジ機能

幅が100us / 150us / 200usのパルスを入力するとサーボはそれぞれSTR1、STR2、STR3のストレッチのデータを現在のストレッチのデータに置き換えます。さらに現在の出力角度をパルス幅に変換して返します。取り込んだストレッチ値は動作に反映されますが、EEPROMIには書き込みませんので、電源を入れなおすとEEPROMのSTRECHの値に初期化されます。



この機能によってPWM動作時でもリアルタイムでストレッチが3種類から選べます。この機能はICS2.0と同等ですが、スピードは一定です。

汎用コマンド（サーボモータ以外）

汎用コマンドはシリアルサーボモータ以外のデバイスをICS3.5へ対応させるためのものです。デバイスの入出力データを仮想メモリエリアにマッピングして使います。

汎用コマンド読み出し

機能

デバイスからデータを読み出す

構成

TX	1	2	3	4
	CMD	SC	ADDR	BYTE

RX	1	2	3	4	5	6	7	8
	送信コマンドのループバック				R_CMD	SC	ADDR	BYTE

9	10	11	12	...	8+(2N-1)	8+2N
DAT1_H	DAT1_L	DAT2_H	DAT2_L	...	DAT(N)_H	DAT(N)_L

解説

CMD								
MSB	7bit	6bit	5bit	4bit	3bit	2bit	1bit	LSB
	1	0	1	x	x	x	x	x
パラメータ読み出しコマンド #101xxxxxb (xxxxxはID番号)				サーボのID 0x00(0) ~ 0x1F(31)				

R_CMD								
MSB	7bit	6bit	5bit	4bit	3bit	2bit	1bit	LSB
	0	0	1	x	x	x	x	x
読み出し返信コマンド #001xxxxxb (xxxxxはID番号)				サーボのID 0x00(0) ~ 0x1F(31)				

※デバイスからの返事ではホストからのコマンドと間違わないようにCMDのMSBをマスクして返ってきます。

SC	
仮想メモリマップ書き込み	0x7F(固定)

ADDR	
仮想メモリマップのアドレス	0x00(0)~0x7F(127)

BYTE	
受け取りデータ数	0x01(1)~0x7F(127)

※ BYTEで指示するデータ数は次ページの仮想メモリーマップ上でのデータ数ですが、実際の送受信では、1バイトデータを上位と下位に2分割して2×BYTE数を実データとして通信します。

DAT1_H~DATA(N)_L	
受け取りデータ(BYTEで指定したバイト数、最大127バイト)	
H、Lはそれぞれデータの上位4ビットと下位4ビット	

例

10bit、4chアナログデバイス(ID=1)

仮想メモリーマップ

ADDR	0	1	2	3	4	5	6	7
0	ch1上位2bit	ch1下位8bit	ch2上位2bit	ch2下位8bit	ch3上位2bit	ch3下位8bit	ch4上位2bit	ch4下位8bit
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-

例1) 全てのデータを読み出す(ID=1)

TX	1	2	3	4
	CMD	SC	ADDR	BYTE
	0xA1	0x7F	0x00	0x08

RX	1	2	3	4	5	6	7	8
	送信コマンドのループバック				R_CMD	SC	ADDR	BYTE
					0x21	0x7F	0x00	0x08

9	10	11	12	13	14	15	16
ch1上位2bit		ch1下位8bit		ch2上位2bit		ch2下位8bit	
上位4bit	下位4bit	上位4bit	下位4bit	上位4bit	下位4bit	上位4bit	下位4bit

17	18	19	20	21	22	23	24
ch3上位2bit		ch3下位8bit		ch4上位2bit		ch4下位8bit	
上位4bit	下位4bit	上位4bit	下位4bit	上位4bit	下位4bit	上位4bit	下位4bit

デバイスはデータ送信時に、メモリーマップ上のデータを1バイト毎に上位4ビット、下位4ビットに分解し、分解した4ビットデータから、上位4ビットが0で下位4ビットがデータとなる1バイトデータを作成します。

結果としてBYTEコマンドでNバイトを要求すると、2Nバイト返ってきます。

例2) CH3のデータのみ読み出す。(ID = 1の場合)

TX	1	2	3	4
	CMD	SC	ADDR	BYTE
	0xA1	0x7F	0x04	0x02

RX	1	2	3	4	5	6	7	8
	送信コマンドのループバック				R_CMD	SC	ADDR	BYTE
					0x21	0x7F	0x04	0x02

9	10	11	12
ch3上位2bit		ch3下位8bit	
上位4bit	下位4bit	上位4bit	下位4bit

汎用コマンド書き込み

機能

デバイスヘータを書き込む

構成

TX	1	2	3	4
	CMD	SC	ADDR	BYTE

5	6	7	8	...	4+(2N-1)	4+2N
DAT1_H	DAT1_L	DAT2_H	DAT2_L	...	DAT(N)_H	DAT(N)_L

RX	1	2	3	4	5	6	7	8
	送信コマンドのループバック				R_CMD	SC	ADDR	BYTE

解説

MSB								CMD								LSB															
7bit				6bit				5bit				4bit				3bit				2bit				1bit				0bit			
1				0				1				x				x				x				x				x			
パラメータ書き込みコマンド #110xxxxxb (xxxxxはID番号)								サーボのID 0x00(0) ~ 0x1F(31)																							

MSB								R_CMD								LSB															
7bit				6bit				5bit				4bit				3bit				2bit				1bit				0bit			
0				0				1				x				x				x				x				x			
書き込み返信コマンド #010xxxxxb (xxxxxはID番号)								サーボのID 0x00(0) ~ 0x1F(31)																							

※デバイスからの返事ではホストからのコマンドと間違わないようにCMDのMSBをマスクして返ってきます。

SC	
仮想メモリマップ書き込み	0x7F(固定)

ADDR	
仮想メモリマップのアドレス	0x00(0)~0x7F(127)

BYTE	
受け取りデータ数	0x01(1)~0x7F(127)

※ BYTEで指示するデータ数は次ページの仮想メモリマップ上でのデータ数ですが、実際の送受信では、1バイトデータを上位と下位に2分割して2×BYTE数を実データとして通信します。

DAT1_H~DATA(N)_L	
受け取りデータ(BYTEで指定したバイト数、最大127バイト) H、Lはそれぞれデータの上位4ビットと下位4ビット	

変更履歴