

USB ホストデバイスコントローラ搭載 4ch モータドライバ

AT32PWM

シリアルコマンドフォーマット

ver 1.1

2010/11/26

 **有限会社 恵比寿電機**

※本書ご利用にあたっての注意事項

- ・本書の内容は将来予告なく変更することがあります。
- ・本書の内容はサンプルプログラムのシリアル通信のコマンドフォーマットを示したものです。ソフトウェアのバージョンおよび使用されるパソコン環境によって正常に動作しない場合があります。
- ・本書に記載されている内容についての質問等のサポートは受けておりません。ご了承ください。
- ・本書に起因するすべての損害に対して弊社では一切責任も負いません。
- ・本書を無断で複写、引用、配布することは禁止します。

目次

1. NOP	2
2. Control (コントロール)	2
2.1 Write Joystick Control Data	
2.2 Read Joystick Control Data	
2.3 Write Button Control Data	
2.4 Read Button Control Data	
2.5 Write Speed Control Data	
2.6 Read Speed Control Data	
3. Register Edit (レジスタエディット)	4
3.1 Write Register	
3.2 Read Register	
4. Parameter Edit (パラメータエディット)	5
4.1 Write Parameter	
4.2 Read Parameter	

1. NOP

コマンドへのレスポンスを返すだけのコマンドです。

Transmit Data

Byte	1	2	3
Type	Start	ID	End
Value	0x61	0x00	0x0A

Receive Data

Byte	1	2	3
Type	Start	ID	End
Value	0x61	0x00	0x0A

2. Control (コントロール)

USB ゲームパッドでの制御をシリアル通信で仮想的に実行するコマンドです。

2.1 Write Joystick Control Data

Transmit Data

Byte	1	2	3	4	5	6	7
Type	Start	ID	X1	Y1	X1	Y1	End
Value	0x61	0x01	0x00~0xFF	0x00~0xFF	0x00~0xFF	0x00~0xFF	0x0A

Receive Data

Byte	1	2	3
Type	Start	ID	End
Value	0x61	0x01	0x0A

仮想ゲームパッドのジョイスティックのデータを書き込みます。

X1: ジョイスティック 1 の X 軸に相当するデータ 左 0x00 ゼロ 0x80 右 0xFF

Y1: ジョイスティック 1 の Y 軸に相当するデータ 上 0x00 ゼロ 0x80 下 0xFF

X2: ジョイスティック 2 の X 軸に相当するデータ 左 0x00 ゼロ 0x80 右 0xFF

Y2: ジョイスティック 2 の Y 軸に相当するデータ 上 0x00 ゼロ 0x80 下 0xFF

2.2 Read Joystick Control Data

Transmit Data

Byte	1	2	3
Type	Start	ID	End
Value	0x61	0x01	0x0A

Receive Data

Byte	1	2	3	4	5	6	7
Type	Start	ID	X1	Y1	X2	Y2	End
Value	0x61	0x01	0x00~0xFF	0x00~0xFF	0x00~0xFF	0x00~0xFF	0x0A

仮想ゲームパッドのジョイスティックのデータを読み出します。

パラメータは” 2.1 Write Joystick Control Data ”と同様です。

2.3 Write Button Control Data

Transmit Data

Byte	1	2	3	4
Type	Start	ID	Btn	End
Value	0x61	0x02	0x00~0xFF	0x0A

Receive Data

Byte	1	2	3
Type	Start	ID	End
Value	0x61	0x02	0x0A

仮想ゲームパッドのボタンのデータを書き込みます。

Btn: ボタンに相当するデータ

0x00 で未入力

0x01 でブレーキ

0x00,0x01 以外はプログラムを書き換えることで拡張可能

2.4 Read Button Control Data

Transmit Data

Byte	1	2	3
Type	Start	ID	End
Value	0x61	0x02	0x0A

Receive Data

Byte	1	2	3	4
Type	Start	ID	Btn	End
Value	0x61	0x02	0x00~0xFF	0x0A

仮想ゲームパッドのボタンのデータを読み出します。

パラメータは” 2.3 Write Button Control Data”と同様です。

2.5 Write Speed Control Data

Transmit Data

Byte	1	2	3	8
Type	Start	ID	Speed	End
Value	0x61	0x03	0x00~0xFF	0x0A

Receive Data

Byte	1	2	3
Type	Start	ID	End
Value	0x61	0x03	0x0A

仮想ゲームパッドで制御する際の速度パラメータを書き込みます。

Speed: 速度パラメータ

2.6 Read Speed Control Data

Transmit Data

Byte	1	2	3
Type	Start	ID	End
Value	0x61	0x03	0x0A

Receive Data

Byte	1	2	3	4
Type	Start	ID	Speed	End
Value	0x61	0x03	0x00~0xFF	0x0A

仮想ゲームパッドで制御する際の速度パラメータを読み出します。

パラメータは” 2.5 Write Speed Control Data”と同様です。

3. Register Edit (レジスタエディット)

AT32UC3B のレジスタにアクセスするコマンドです。

3.1 Write Register

Transmit Data

Byte	1	2	3	4	5	6	7	8	9
Type	Start	ID	Offset (2byte)		Data (4byte)				End
Value	0x61	<i>ref Table.1</i>	0x0000~0xFFFF		0x00000000~0xFFFFFFFF				0x0A

Receive Data

Byte	1	2	3	4	5
Type	Start	ID	Offset (2byte)		End
Value	0x61	<i>ref Table.1</i>	0x0000~0xFFFF		0x0A

指定したレジスタにデータを書き込みます。レジスタは”Peripheral”と”Offset”で指定します。

ID: 書き込むレジスタが所属する”Peripheral”の ID を指定します。Table.1 を参照ください。

Offset: ”Peripheral”のベースアドレスからのオフセット。値は AT32UC3B の仕様書を参考ください。

Data: レジスタに書き込むデータ

Table.1 “Peripheral”の ID

Name	ID	Name	ID	Name	ID	Name	ID
USB	0x0B	PM	0x10	USART0	0x15	PWM	0x1A
HMATRIX	0x0C	RTC	0x11	USART1	0x16	SSC	0x1B
HFLASHC	0x0D	WDT	0x12	USART2	0x17	TC	0x1C
PDMA	0x0E	EIM	0x13	SPI0	0x18	ADC	0x1D
INTC	0x0F	GPIO	0x14	TWI	0x19	ABDAC	0x1E

3.2 Read Register

Transmit Data

Byte	1	2	3	4	5
Type	Start	ID	Offset (2byte)		End
Value	0x61	<i>ref Table.1</i>	0x0000~0xFFFF		0x0A

Receive Data

Byte	1	2	3	4	5	6	7	8	9
Type	Start	ID	Offset (2byte)		Data (4byte)				End
Value	0x61	<i>ref Table.1</i>	0x0000~0xFFFF		0x00000000~0xFFFFFFFF				0x0A

指定したレジスタのデータを読み出します。レジスタは”Peripheral”と”Offset”で指定します。

パラメータは”3.1 Write Register”と同様です。

4. Parameter Edit (パラメータエディット)

ファームウェアで使用されている内部変数にアクセスします。

4.1 Write Parameter

Transmit Data

Byte	1	2	3	4	5
Type	Start	ID	Data (2byte)		End
Value	0x61	ref Table.2,3	0x0000~0xFFFF		0x0A

Receive Data

Byte	1	2	3
Type	Start	ID	End
Value	0x61	ref Table.2,3	0x0A

指定したパラメータにデータを書き込みます。

ID: 書き込むパラメータを指定します。Table.2,3 を参照ください。

Data: パラメータに書き込むデータ

4.2 Read Parameter

Transmit Data

Byte	1	2	3
Type	Start	ID	End
Value	0x61	ref Table.2,3	0x0A

Receive Data

Byte	1	2	3	4	5
Type	Start	ID	Data (2byte)		End
Value	0x61	ref Table.2,3	0x0000~0xFFFF		0x0A

指定したパラメータのデータを読み出します。

パラメータは” 4.1 Write Parameter”と同様です。

Table.2 制御関連のパラメータの ID

Name	ID	Function
制御有効/無効	0x1F	0x00 : 制御有効 0x00 以外 : 制御無効
sampling_time	0x20	メインループ処理の刻み時間 単位:[ms]
Limit_voltage	0x21	電源電圧の下限。 AD[3]の測定値が設定値以下で電源電圧低下と判定。 値 = (設定したい停止電圧値[V])*0.1/3.3*1024 例: 電源電圧を 6[V]にする場合、値=6*0.1/3.3*1024÷186
Limit_voltage_cnt	0x22	設定値の回数だけ連続で電源電圧低下を検知すると、 ドライバ機能を停止。

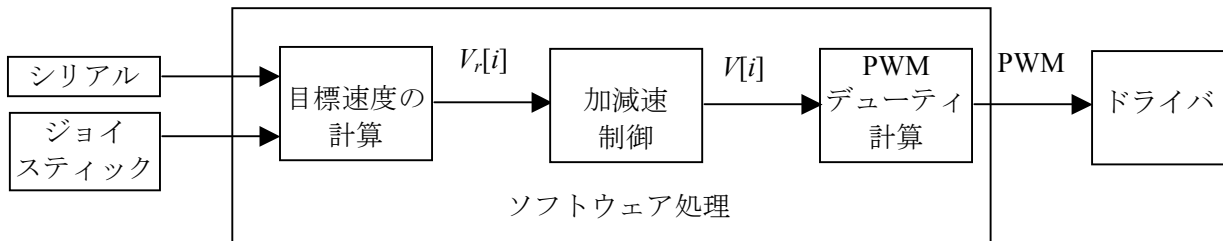
Table.3 速度制御関連のパラメータの ID

Name	ID	Function
$V_r[0]$	0x23	ドライバ 0 の目標速度 (-32768~32767)
$V[0]$	0x24	ドライバ 0 の実行速度 (-32768~32767)
$V_{max}[0]$	0x25	ドライバ 0 の最大速度 (0~32767)
$V_0[0]$	0x26	ドライバ 0 の初速度 (0~32767)
$acc[0]$	0x27	ドライバ 0 の加速度 (0~32767)
$dec[0]$	0x28	ドライバ 0 の減速度 (0~32767)
$dec0[0]$	0x29	ドライバ 0 の停止時の減速度 (0~32767)
$V_r[1]$	0x2A	ドライバ 1 の目標速度 (-32768~32767)
$V[1]$	0x2B	ドライバ 1 の実行速度 (-32768~32767)
$V_{max}[1]$	0x2C	ドライバ 1 の最大速度 (0~32767)
$V_0[1]$	0x2D	ドライバ 1 の初速度 (0~32767)
$acc[1]$	0x2E	ドライバ 1 の加速度 (0~32767)
$dec[1]$	0x2F	ドライバ 1 の減速度 (0~32767)
$dec0[1]$	0x30	ドライバ 1 の停止時の減速度 (0~32767)
$V_r[2]$	0x31	ドライバ 2 の目標速度 (-32768~32767)
$V[2]$	0x32	ドライバ 2 の実行速度 (-32768~32767)
$V_{max}[2]$	0x33	ドライバ 2 の最大速度 (0~32767)
$V_0[2]$	0x34	ドライバ 2 の初速度 (0~32767)
$acc[2]$	0x35	ドライバ 2 の加速度 (0~32767)
$dec[2]$	0x36	ドライバ 2 の停止時の減速度 (0~32767)
$dec0[2]$	0x37	ドライバ 2 の減速度 (0~32767)
$V_r[3]$	0x38	ドライバ 3 の目標速度 (-32768~32767)
$V[3]$	0x39	ドライバ 3 の実行速度 (-32768~32767)
$V_{max}[3]$	0x3A	ドライバ 3 の最大速度 (0~32767)
$V_0[3]$	0x3B	ドライバ 3 の初速度 (0~32767)
$acc[3]$	0x3C	ドライバ 3 の加速度 (0~32767)
$dec[3]$	0x3D	ドライバ 3 の減速度 (0~32767)
$dec0[3]$	0x3E	ドライバ 3 の停止時の減速度 (0~32767)

※ ここで示す速度はすべてプログラムの内部パラメータで実際の速度ではありません。

※速度制御について

ファームウェア内では、下図のようにシリアルもしくはジョイスティックの入力値からドライバを制御する PWM 信号のデューティを制御しています。制御のパラメータはシリアル通信でコマンドを送信することで変更可能です。



・目標速度 $V_r[i]$ の計算

目標速度 $V_r[i]$ の設定は、シリアル通信で設定する方法とジョイスティックの入力(シリアル通信でのコントロールも含む)で設定する方法があります。ジョイスティックのアナログ入力による目標速度 $V_r[i]$ は次式で求められます。

$$V_r[i] = rate[i] \times (V_{max}[i] - V_0[i]) + V_0[i]$$

$$rate[i] = \frac{A_{X1}[i] \times X1 + A_{Y1}[i] \times Y1 + A_{X2}[i] \times X2 + A_{Y2}[i] \times Y2}{128} \times \frac{Speed}{255}$$

ただし、 $0 \leq rate[i] \leq 1$ 、 $i = 0,1,2,3$

$X1, Y1, X2, Y2$ はジョイスティックのアナログ入力データ。(シリアルコマンドのパラメータから 0x80 を 0 として符号付整数に型変換)

$$-128 \leq X1 \leq 127, -128 \leq Y1 \leq 127, -128 \leq X2 \leq 127, -128 \leq Y2 \leq 127$$

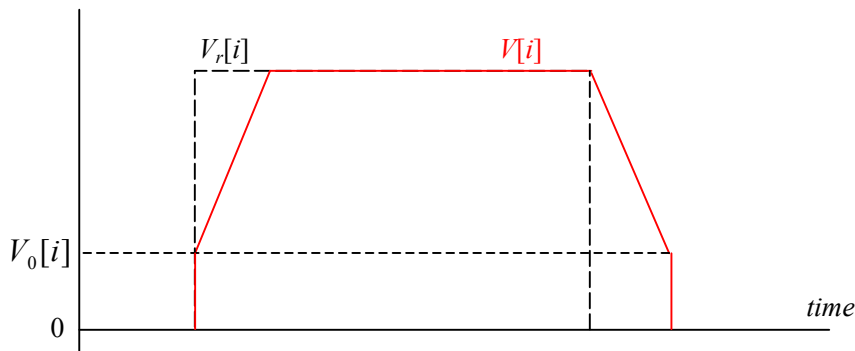
$A_{X1}[i], A_{Y1}[i], A_{X2}[i], A_{Y2}[i]$ は任意のパラメータ。

$$-1 \leq A_{X1}[i] \leq 1, -1 \leq A_{Y1}[i] \leq 1, -1 \leq A_{X2}[i] \leq 1, -1 \leq A_{Y2}[i] \leq 1$$

$Speed$ は速度のパラメータ。(シリアルコマンドのパラメータと同等)

$$0 \leq Speed \leq 255$$

・加減速制御



$acc[i]$ 、 $dec[i]$ は加速度、減速度を設定することで、図のように実行速度を台形制御することができます。

加速時には1 サンプル周期毎に $acc[i]$ で設定した値が $V[i]$ に加算されます。

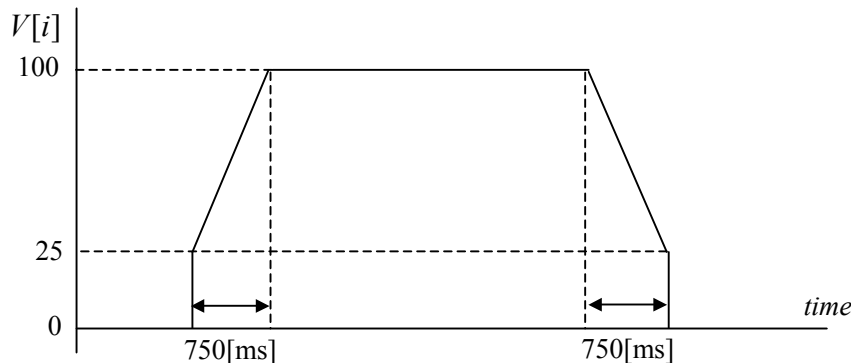
減速時には1 サンプル周期毎に $dec[i]$ で設定した値が $V[i]$ から減算されます。

ただし、停止時は1 サンプル周期毎に $dec0[i]$ で設定した値が $V[i]$ から減算されます。

— $V_0[i] < V[i] < V_0[i]$ の時、 $V[i]=0$ となります。

$acc[i]=0$ 、 $dec[i]=0$ 、 $dec0[i]=0$ の時、 $V_r[i]=V[i]$ となります。

設定例： サンプル周期=10[ms]、 $acc[i]=1$ 、 $dec[i]=1$ 、 $dec0[i]=1$ 、 $V_0[i]=25$ 、 $V_r[i]=100$ の場合



・出力 PWM のデューティ計算

実行速度から出力 PWM のデューティは次式で示されます。

$$cdty[i] = \frac{cprd[i] \times V[i]}{V_{max}[i]}$$

$cdty[i]$ はパルス幅、 $cprd[i]$ は周期。具体的な PWM の設定方法は仕様書を参考ください。

※エスケープ処理について

コマンドの Start、End 以外で 0x0A,0x61,0x7D を送信する場合はエスケープ処理します。

エスケープ処理は、0x7D を送信、次にデータと 0x20 との XOR の値を送信します。

データが 0x0A の場合、0x7D、0x2A を送信

データが 0x61 の場合、0x7D、0x41 を送信

データが 0x7D の場合、0x7D、0x5D を送信

※レジスタエディットについて

レジスタを操作してモータを制御する場合は、速度制御の処理と競合します。必ず制御を無効にしてレジスタを操作してください。

USB ホストデバイスコントローラ搭載 4ch モータドライバ
AT32PWM
シリアルコマンドフォーマット

2010/11/24 ver1.0

2010/11/26 ver1.1 目標速度の計算式を修正

有限会社 恵比寿電機

〒781-5101

高知県高知市布師田 3992-3 企業化支援センター2階B

Tel 088-845-8650

URL <http://www.ebisu-denki.com/>

E-Mail ebisu_denki@yahoo.co.jp
