

## 第2章

### ファームウェアの使いかた

…各モードの機能とベンダ・リクエスト・コマンドの詳細

EzFirm/FX2は、サイプレス社のUSBターゲット・デバイスであるEZ-USB/FX2シリーズ用に作成した汎用I/Oファームウェアです。USBを利用してI/O制御を行う場合、USBのベンダIDを取得し、USBプロトコルやターゲット・コントローラの仕様を理解し、ファームウェアを作成しなくてはなりません。

EzFirm/FX2はこのような問題に対応し、USBプロトコルなどの知識を必要とせずに、USB2.0で簡単にI/O操作が行えるようにしたものです。

#### 2-1 ファームウェア EzFirm/FX2 の機能

8ビット単位でのリード/ライトが行えるPIOモードのほか、FX2チップがもつGPIFやスレーブFIFO機能にも対応し、USB2.0の特徴を生かした高速大容量のデータ転送が可能です。

GPIF (General Programmable Interface) をサポートしたことにより、リード/ライト信号などのストロープ信号やウェイト/レディ信号が必要なバス型のインターフェースをもったI/Oデバイスとの接続も簡単に行えます。

簡易的な非同期シリアル・ポートもサポートしており、EIA-232対応の表示器やキー入力装置なども簡単に接続することができるようになっています。また、内部/外部クロック選択可能な16ビット・カウンタ機能もサポートしており、内部クロックや外部クロックの分周出力や、パルス・カウント用などとして利用することができます。

**【注】** 本書添付のボード (UCT-203) に使用した56ピン・パッケージ品では、シリアル・ポートやカウンタ/タイマ入出力ピンをもたないため、これらの機能は使用できません。

#### ●おもな仕様

ファームウェア (EzFirm/FX2) のおもな仕様を表1に示します。

#### ●EzFirm/FX2におけるI/Oモード

EzFirm/FX2では、FX2のもつパラレルI/Oポートに対し、次の3種類の動作モードをサポートしています。

表1 ファームウェア (EzFirm/FX2) のおもな仕様

分類	項目	仕様
I/O点数 (*1)	汎用I/O	最大39点 (PORTA ~ PORTE. PE.2はタイマ出力として使用)
	CTL出力	6点 (3点)
	RDY入力	6点 (2点)
高速転送モード	動作モード	GPIFモード (内部バス・マスタ) スレーブFIFOモード (外部バス・マスタ)
	データ・バス幅	16ビット
	最大I/O速度	96Mバイト/秒 (ただし、ホストとの伝送はUSBバスによって制約)
シリアル・ポート (*2)	ビットレート	2400 / 4800 / 9600 / 19200 / 28800 / 38400 / 57600 bps
	データ長	8ビット
	ストップ・ビット	1ビット
	パリティ	なし
	フロー制御	なし
タイマ (*2)	クロック	内部、または外部。内部クロックは4 MHz/12 MHzから選択
	カウント値	16ビット
	タイマ出力	オーバーフロー時に1クロックぶんのパルスを出力

\*1: 本書添付基板では、PORTA/B/Dの24点、CTL出力は3点、RDY入力は2点。

\*2: シリアル・ポート、タイマ機能は本キットのボードでは使用不可 (ピンがないため)

### (1) PIOモード

PORTA ~ PORTE (本キットのボードではPORTA/B/D) の各グループ単位で入出力方向を指定して、ホストからのコマンドで8ビット単位の入出力が行えます。

### (2) GPIFモード (FX2マスタ型の高速データ転送モード)

CPUと独立して動作するFX2内蔵のGPIFユニットが、バルクIN/OUTエンド・ポイントとターゲット機器の間の転送動作を自動実行します。CPUを経由せず、FX2がバス・マスタとなって外部とのデータ転送を自動的に行うため、高速なデータ転送が可能です。また、波形生成テーブル (ウェーブフォーム・データ) を定義することにより、単純なストロブだけでなく、外部からのステータス入力を利用したハンドシェイクなども行えます。

### (3) スレーブFIFOモード (外部マスタ型の高速データ転送モード)

バルクIN/OUTエンド・ポイントを外部からFIFOメモリのようにみせかけることで、ターゲットはFIFOメモリをリード/ライトする感覚でホスト-ターゲット間の転送が行えます。

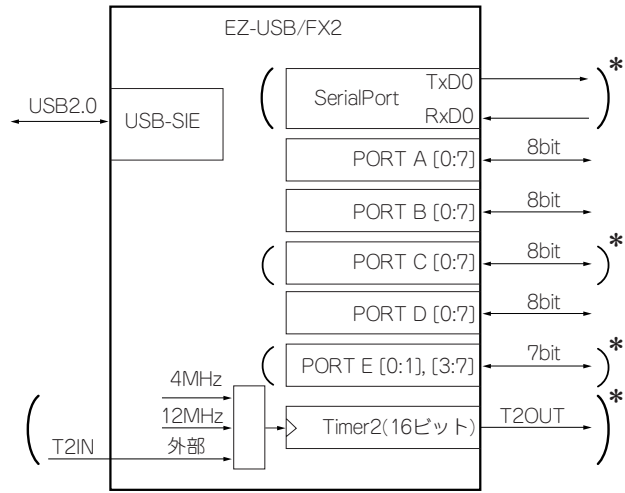
\*

\*

それぞれの動作モードにおけるポート割り付けは図1のようになっています。各モードの詳細については後で説明します。

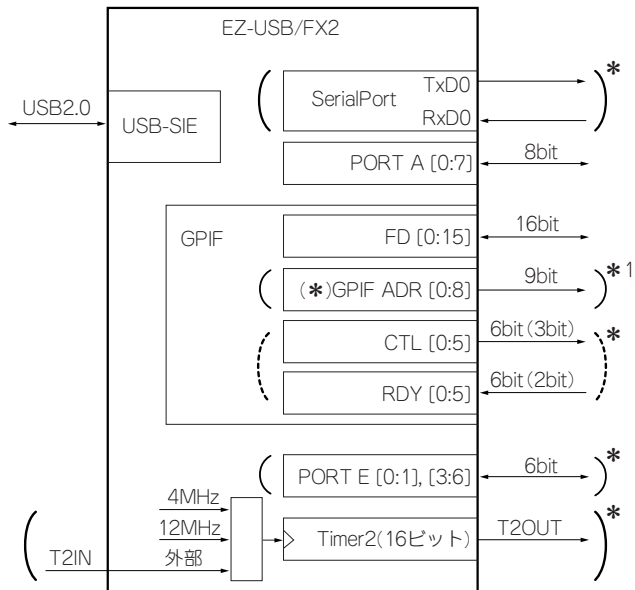
## ● EzFirm/FX2のエンド・ポイント設定

EzFirm/FX2では、FX2のもつエンド・ポイントを表2のように設定しています。エンド・ポイントは、ホストとターゲットの間に設けられた通信バッファ・メモリのようなものです。EPOはホストとターゲット間のデバイス・リクエスト (コマンド) の送受信に使われるエンド・ポイントです。EzFirm/FX2で用意しているコマン



(a) PIOモード

\* : EzIO/FX2ボードではPORTA/B/Dのみ  
(SerialPort, PORTC, PORTE, Timer2はなし)

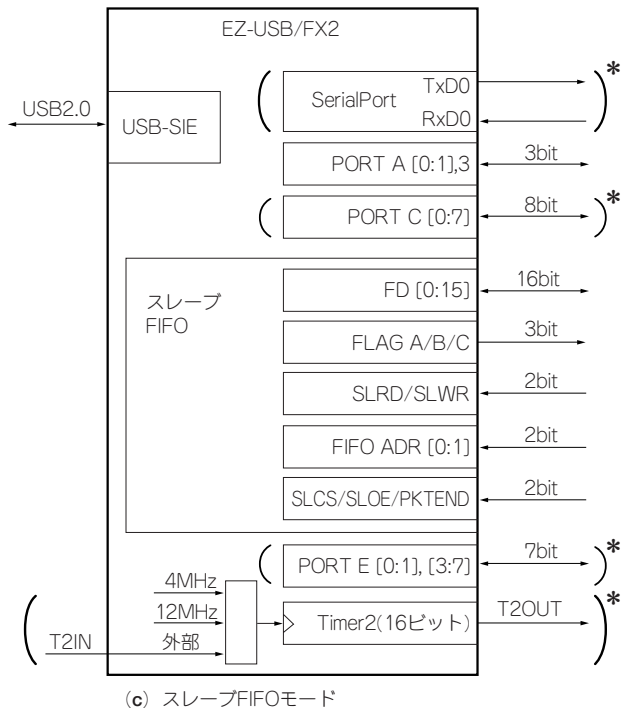


(b) GPIFモード

\*1 : GPIF ADRは設定によってPIOとしても利用可

\* : EzIO/FX2ボードではPORTA, FD[0..15], CTL[0..2], RDY[0..1]のみ  
(Serial Port,GPIFADR[0..8],CTL[3..5], RDY[2..5], PORTE[0..1], Timer 2 はなし)

図1 動作モードとポートの割り付け (次頁へつづく)



\* : UCT-203ボードではSerialPort, PORTC/E/Timer2はなし

図1 動作モードとポートの割り付け (前頁からのつづき)

ども、EP0を利用したベンダ定義のリクエスト（コマンド）として実装されています。

EP0以外のエンド・ポイントはデータ転送のエンド・ポイントです。EzFirm/FX2では、EP1OUTをシリアル・ポートの送信データ用、EP1INをシリアル・ポートの受信データ用、EP2を高速伝送モード（GPIFモード/スレーブFIFOモード）時のデータ送出（ホストからターゲットの方向）用に、EP6を高速伝送モード時のデータ受信（ターゲットからホストの方向）用として使用しています。

これらのエンド・ポイントはWindowsドライバ側では「パイプ」と呼ばれ、パイプ番号を使ってそれぞれのエンド・ポイントを区別しています。

**【注】**本キットのボード（UCT-203）に使用した56ピン・パッケージ品では、シリアル・ポートやカウンタ/タイマー入出力ピンをもたないため、EP1OUTとEP1INは使用できません。

### ●シリアル・ポート（注：UCT-203では使用できない）

EzFirm/FX2では、EP1OUTとEP1INエンド・ポイントをFX2内蔵のシリアル・ポート0とのコミュニケーション用に割り付けており、エンド・ポイントのリード/ライトを行うだけで、シリアル・ポートを利用した周辺機器とのインターフェースを行うことができます。ボーレートはSetModeConfigリクエストによって、2400～

表2 EzFirm/FX2でのエンド・ポイント設定

パイプ番号	エンド・ポイント	種別	フルスピード (12 Mbps)	ハイスピード (480 Mbps)	用途
-	EP0	コントロール (IN/OUT)	64 バイト	64 バイト	デバイス・リクエスト
0	EP1OUT*	バルク OUT	64 バイト	64 バイト	シリアル・ポート送信データ
1	EP1IN*	バルク IN	64 バイト	64 バイト	シリアル・ポート受信データ
2	EP2	バルク OUT	64 バイト×4 バンク	512 バイト×4 バンク	GPIF / スレーブ FIFO 出力
3	EP6	バルク IN	64 バイト×4 バンク	512 バイト×4 バンク	GPIF / スレーブ FIFO 入力

\*: 本キットのボード (UCT-203) では使用できない

57600 bps の間で設定可能です (詳細は **SetModeConfig** リクエストの説明を参照のこと)。通信パラメータは、  
データ長: 8 ビット, ストップ・ビット: 1 ビット, パリティ: なし  
に固定です。フロー制御はありません。

送信方向は、EP1OUT エンド・ポイント・バッファ (64 バイト) が送信バッファとなり、受信方向は、EzFirm/FX2 内部に 64 バイトの受信バッファがありますので、1 パケットぶんのデータをまとめて受け渡すことができます。

ファームウェア起動後のデフォルトでは 9600 bps に設定されています。ターゲットの通信速度が 9600 bps であれば、起動後に EP1OUT エンド・ポイント (パイプ番号 0) と EP1IN (パイプ番号 1) のライト/リードを行うだけで、ターゲットとの通信が行えます。

**【注】** EZ-USB/FX2 のシリアル・ポート 0 は受信 FIFO をもたないため、EzFirm/FX2 では 1 キャラクターずつ割り込みを使って処理しています。ビットレートが高い場合やシリアル・データ受信中にベンダ・リクエストを行った場合などには、データを取りこぼす可能性があります。また、FX2 のシリアル・ポートが初期化されたあと、1 バイトの 0x00 が読み出されることがあります。

#### ▶ シリアル・ポートの送受信開始タイミング

EzFirm/FX2 では、送信時は EP1OUT エンド・ポイントへの書き込みが行われるとすぐに送信が開始されます。

受信方向は、不要な受信割り込みの発生を抑えるため、電源投入後や、**SetPortConfig** コマンドによる設定を行った直後は受信禁止状態になっており、EP1IN エンド・ポイントのリードが行われた段階で受信を開始するようになっています。この受信開始のリード操作に対して、EzFirm/FX2 はサイズ 0 のデータを返します。

シリアル・ポート受信を利用する場合には、ポート設定を行ったあとに必ず EP1 をダミー・リードして、受信開始を指示するようにしてください。

## 2-2 パラレルI/O動作

EzFirm/FX2 では外部インターフェースとして、単純な入出力を行う PIO モードと、高速伝送をサポートする GPIF モード、スレーブ FIFO モードをサポートしています。デフォルトでは PIO モードで、汎用 I/O はすべて入力に設定されます。

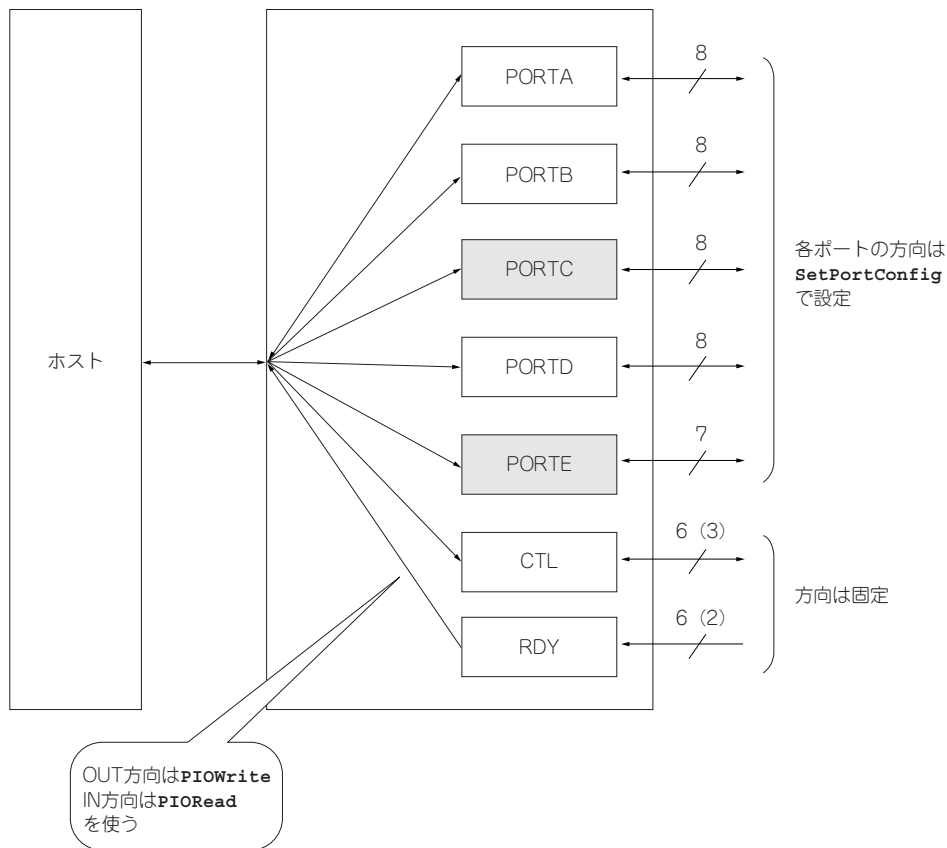


図2 PIOモードのI/Oポート構成

### ●PIOモード

PIOモードでは、汎用I/Oをそれぞれのグループ（PA～PE）ごとに入出力方向を指定することができます。CTL [5：0] は出力専用、RDY [5：0] は入力専用です（図2）。

汎用I/Oの入出力設定はSetModeConfigで行います。出力ポートへのデータ・セットはPIOwriteリクエスト、入力ポートの読み出しはPIOReadリクエストで行います。

### ●GPIFモード

GPIFモードでは、バルクOUTエンド・ポイント（EP2）、またはバルクINエンド・ポイント（EP6）と外部機器間のデータ転送を、FX2内部のGPIF（General Programmable Interface）が制御するモードです。

エンド・ポイントと外部の入出力操作はGPIFがすべて行いますので、GPIFTrigリクエストでGPIFを起動したあとは、EP2へのライト、あるいはEP6のリードを行うだけで、データ転送が行われます。FX2の内蔵CPUを

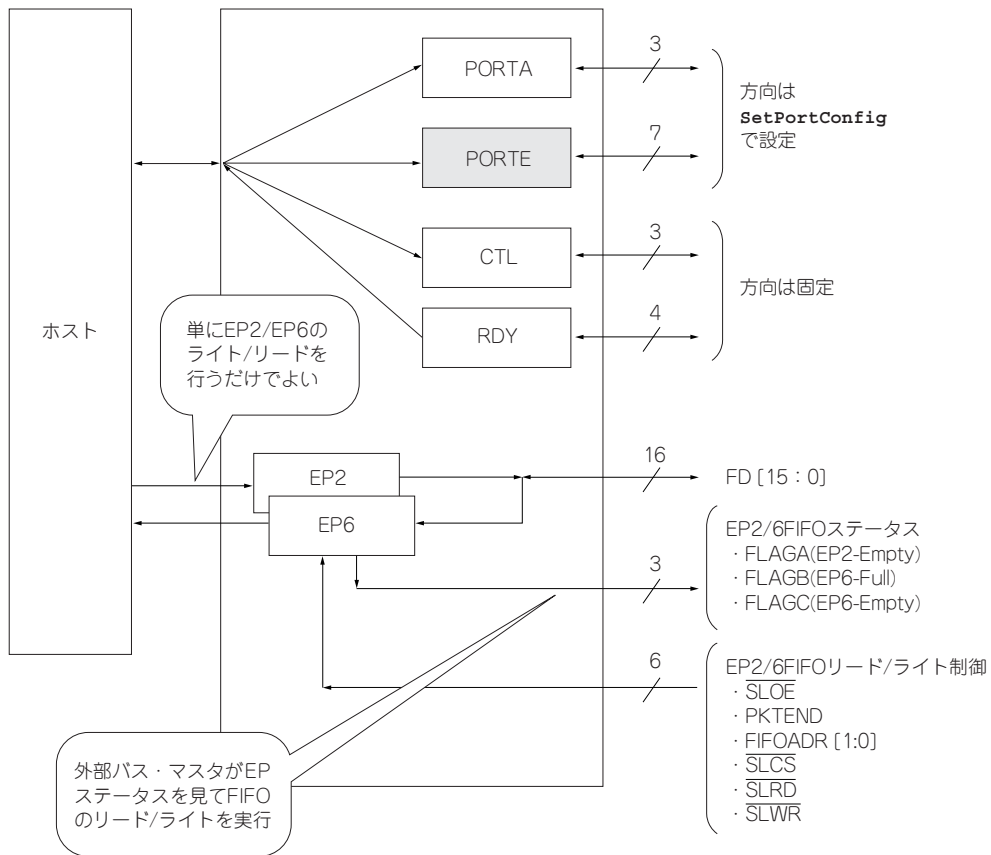


図3 スレーブFIFOモードのI/Oポート構成

いっさい介在せずに動作することから、高速で大容量のデータ転送に向いています。

GPIFは8ステート（うち1個はアイドル状態用としてリザーブ）のステート・マシンで、ウェーブフォーム・ディスクリプタと呼ばれる波形定義テーブルにしたがって、エンド・ポイントと外部データ・バス（FD [16 : 0]）との間の入出力や、CTL出力端子の制御、RDY入力による動作の分岐などを行います。

GPIFを使うことで、あたかも16ビット・バスをもったCPUが外部回路にライト/リード動作を行うようなシケンスを作成することができます。また、外部に特別なハードウェアを用意することなく、セットアップ・タイムやホールド・タイムを確保したり、外部機器とのハンドシェイクを行わせることも容易になっています。

### ●スレーブFIFOモード

スレーブFIFOモードは、外部機器に対してエンド・ポイントをあたかもFIFOメモリであるかのように見せかけるモードです（図3）。

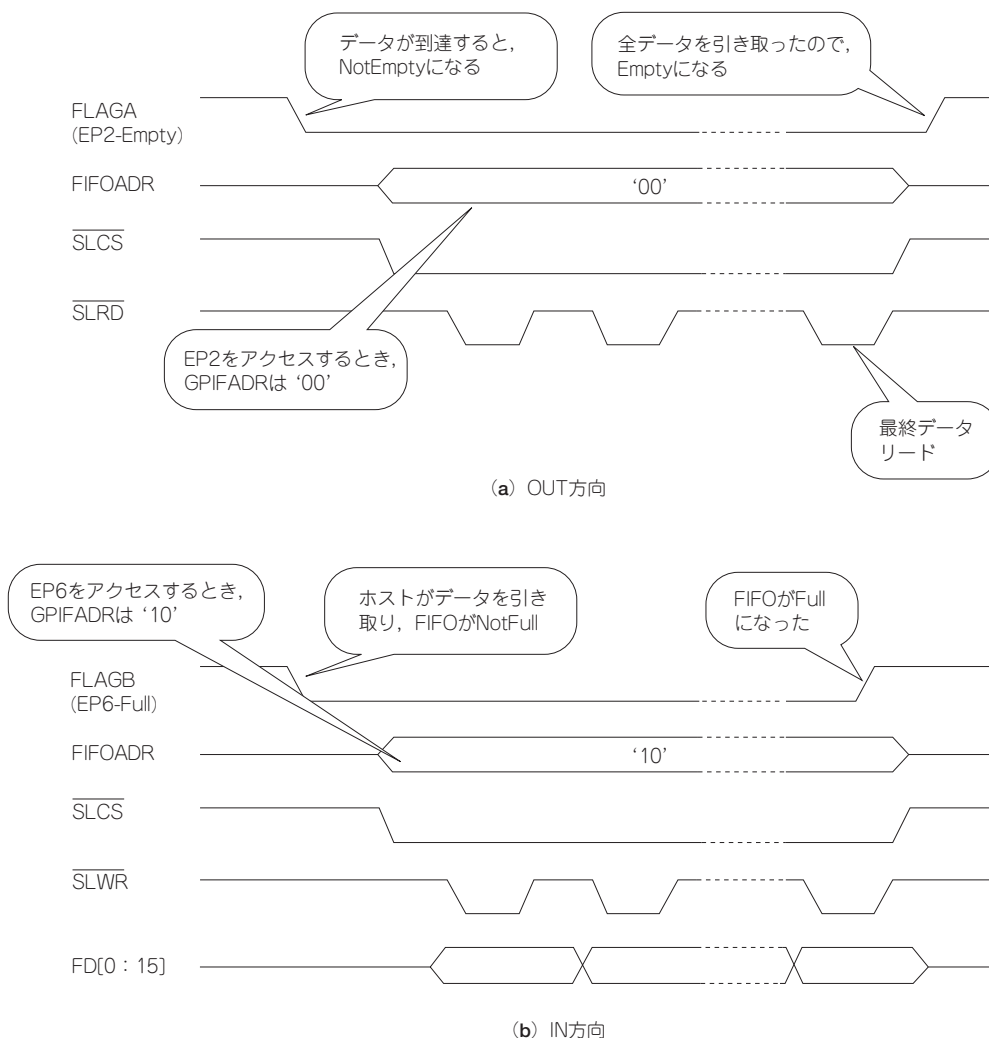


図4 スレーブFIFOモードの動作

EP2OUT エンド・ポイントが空である、すなわちホストからまだ新規のデータが書き込まれていない場合、FLAGA ピン (EP2-Empty) はH レベルになっています。ホストから1パケットぶんのデータを受信すると、FLAGA はL レベルになりますので、周辺機器側は FIFOADR [1 : 0] = 00 として、 $\overline{\text{SLCS}}$ 、 $\overline{\text{SLOE}}$ 、 $\overline{\text{SLRD}}$  をL レベルにアサートしてデータを引き取ります。1回のリードごとにバイト数カウンタが減少し、ゼロになった時点でFLAGA はH レベルに復帰します [図4(a)]。

EP6IN エンド・ポイントの場合はこれとは逆で、EP6 エンド・ポイントが空の場合、FLAGB (EP6-Full) フラグはL レベルになっていますので、周辺機器側が $\overline{\text{SLCS}}$ 、 $\overline{\text{SLWR}}$  を利用してデータを書き込みます。1パケットぶ



表3 EzFirm/FX2のサポートするベンダ・リクエスト

リクエスト名	bm Request Type	bRequest	wValue	wIndex	wLength	データ・ステージ	動作
SetPortConfig	0xC0	0x00	動作モード指定	0x0000	0x0002	0x0000 (Read)	I/Oポートの方向や動作モードの指定
SetTimer2	0xC0	0x01	タイマ・カウント値	クロック選択	0x0002	0x0000 (Read)	Timer2のカウント値, 動作モード設定
PIOWrite	0xC0	0x02	ライト・データ (8ビット)	ポート選択	0x0002	0x0000 (Read)	PIOポートへの8ビット・データ・セット
PIORead	0xC0	0x03	0x0000	ポート選択	0x0002	リード・データ (8ビット)	PIOポートからの8ビット・データ・リード (2バイト目は無効)
SetWaveData	0x40	0x05	0x0000	テーブル選択	0x0002	ウェーブフォーム・データ (32バイト)	ディスクリプタ・テーブル設定
GPIFAbot	0xC0	0x07	0x0000	0x0000	0x0020	0x0000 (Read)	GPIFの強制停止
GPIFSingle Write	0xC0	0x08	ライト・データ (16ビット)	GPIFADR値	0x0002	リード・データ (16ビット)	GPIFのシングル・ライト実行
GPIFSingle Read	0xC0	0x09	0x0000	GPIFADR値	0x0002	0x0000 (Read)	GPIFのシングル・リード実行
GPIFTrig	0xC0	0x10	転送カウンタ(下位)	転送カウンタ(上位)	0x0002	0x0000 (Read)	GPIFのバースト・ライト実行
GPIFTrig	0xC0	0x12	転送カウンタ(下位)	転送カウンタ(上位)	0x0002	0x0000 (Read)	GPIFのバースト・リード実行

んのデータを書き込むとFLAGBがHレベルになり、ホスト側で書き込まれたデータを読めるようになります。ホストがデータを引き取ると再びFLAGBがLレベルになり、新規のデータが書き込めるようになったことを外部機器に通知します [図4(b)]。

なお、EP2, EP6とも4バンク・バッファになっているため、最大4パケットぶんのデータがFX2内部に蓄積可能です。

## 2-3 EzFirm/FX2のベンダ・リクエスト

### ● EzFirm/FX2のベンダ・リクエスト一覧

EzFirm/FX2では表3に示すベンダ・リクエスト・コマンドを用意しています。

### ● ベンダ・リクエスト・コマンド

以下に、個々のベンダ・リクエスト・コマンドの動作とパラメータの詳細について解説します。

## ■ SetPortConfig (動作モード設定, I/Oポート設定)

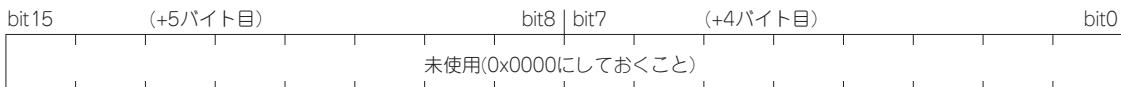
bmRequestType : 0xC0

bRequest : 0x00

wValue

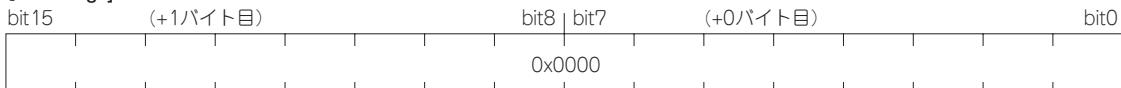


wIndex



wLength : 0x02

[DataStage]



### ●動作

EzFirm/FX2の動作モード (PIOモード/GPIFモード/スレーブFIFOモード), シリアル・ポートのビットレート, PIOポートの入出力方向設定を行います。

**wValue [15 : 14] …MODE (動作モード設定)**

FX2の動作モード (PIOモード/GPIFモード/スレーブFIFOモード) を設定します。GPIFモード, スレーブFIFOモードのときにはPORTBとPORTC, およびPORTAとPORTEの一部がGPIF/スレーブFIFO動作用に切り替わります。これらのポートは汎用I/Oポートとしては利用できなくなりますので注意してください。

MODE値の設定

MODE	設定
00	PIOモード
01	(未使用)
10	GPIFモード
11	スレーブFIFOモード

動作モードとPIOピンの割り当て

		MODEビットの設定値				
		MODE=00		MODE=10		MODE=11
ポート	ビット	PIOモード		GPIFモード		スレーブFIFOモード
PA	7	⇔	PA7	⇔	PA7	← SLCS
	6	⇔	PA6	⇔	PA6	← PKTEND
	5	⇔	PA5	⇔	PA5	

PA	4	⇔	PA4	⇔	PA4	←	FIFOADR1
	3	⇔	PA3	⇔	PA3	←	FIFOADR0
	2	⇔	PA2	⇔	PA2	←	SLOE
	1	⇔	PA1	⇔	PA1	⇔	PA1
	0	⇔	PA0	⇔	PA0	⇔	PA0
PB	7	⇔	PB7	⇔	FD7	⇔	FD7
	6	⇔	PB6	⇔	FD6	⇔	FD6
	5	⇔	PB5	⇔	FD5	⇔	FD5
	4	⇔	PB4	⇔	FD4	⇔	FD4
	3	⇔	PB3	⇔	FD3	⇔	FD3
	2	⇔	PB2	⇔	FD2	⇔	FD2
	1	⇔	PB1	⇔	FD1	⇔	FD1
	0	⇔	PB0	⇔	FD0	⇔	FD0
PC	7	⇔	PC7	→	(GPIFADR7)	⇔	PC7
	6	⇔	PC6	→	(GPIFADR6)	⇔	PC6
	5	⇔	PC5	→	(GPIFADR5)	⇔	PC5
	4	⇔	PC4	→	(GPIFADR4)	⇔	PC4
	3	⇔	PC3	→	(GPIFADR3)	⇔	PC3
	2	⇔	PC2	→	(GPIFADR2)	⇔	PC2
	1	⇔	PC1	→	(GPIFADR1)	⇔	PC1
	0	⇔	PC0	→	(GPIFADR0)	⇔	PC0
PD	7	⇔	PD7	⇔	FD15	⇔	FD15
	6	⇔	PD6	⇔	FD14	⇔	FD14
	5	⇔	PD5	⇔	FD13	⇔	FD13
	4	⇔	PD4	⇔	FD12	⇔	FD12
	3	⇔	PD3	⇔	FD11	⇔	FD11
	2	⇔	PD2	⇔	FD10	⇔	FD10
	1	⇔	PD1	⇔	FD9	⇔	FD9
	0	⇔	PD0	⇔	FD8	⇔	FD8
PE	7	⇔	PE7	⇔	(GPIFADR8)	⇔	PE7
	6	⇔	PE6	⇔	PE6	⇔	PE6
	5	⇔	PE5	⇔	PE5	⇔	PE5
	4	⇔	PE4	⇔	PE4	⇔	PE4
	3	⇔	PE3	⇔	PE3	⇔	PE3
	2	→	T2OUT	→	T2OUT	→	T2OUT
	1	⇔	PE1	⇔	PE1	⇔	PE1
	0	⇔	PE0	⇔	PE0	⇔	PE0
CTL	5			→	CTL5		
	4			→	CTL4		
	3			→	CTL3		
	2			→	CTL2	→	FLAGC (未使用)
	1			→	CTL1	→	FLAGB (EP6_FULL)
	0			→	CTL0	→	FLAGA (EP2_EMPTY)
RDY	5			←	RDY5		
	4			←	RDY4		
	3			←	RDY3		
	2			←	RDY2		
	1			←	RDY1	←	SLWR
	0			←	RDY0	←	SLRD

← : IN 方向 (外部からFX2)    → : OUT 方向 (FX2から外部)    ⇔ : 双方向

**wValue [12 : 10] … BRATE** (シリアル・ポートのビットレート設定)

シリアル・ポートのビットレート (bps) を設定します。データ長は8ビット、パリティなし、ストップ・ビットは1ビットに固定です。

BRATE値の設定

BRATE	設定 (bps)
000	2400
001	4800
010	9600
011	19200
100	28800
101	34800
110	57600
111	(予約)

**wValue [9 : 8] … PE** (PORTEの動作モード設定)

FX2の汎用I/OポートであるPORTEの入出力方向を設定します。ただし、PE.2 (PORTEのビット2) は常にT2OUT出力端子として使用されますので、本ビットによる設定は無効です。また、**MODE** = '10' (GPIFモード) で **PC** = '11' (GPIFADRとして使用) に設定したときは、PE.7はGPIFADR8に設定され、本ビットによる設定は無効です。

PE値と設定

PE	設定
00	入力ポート
01	出力ポート
10	(予約)
11	(予約)

**wValue [7 : 6] … PD** (PORTDの動作モード設定)

FX2の汎用I/OポートであるPORTDの入出力方向を設定します。**MODE** = '10' (GPIFモード) または **MODE** = '11' (スレーブFIFOモード) のときは、**PD**はFD [15 : 0] (FIFOデータの上位8ビット) になり、本ビットによる設定は無効です。

PD値と設定

PD	設定
00	入力ポート
01	出力ポート
10	(予約)
11	(予約)

**wValue [5 : 4] … PC** (PORTCの動作モード設定)

FX2の汎用I/OポートであるPORTCの入出力方向を設定します。**MODE** = '10' (GPIFモード) のときは、さ

らに GPIF の専用アドレス信号 (GPIFADR) に設定することもできます。本ビットを '11' (GPIFADR) にした場合、PE.7は GPIFADR8 (GPIFADRの最上位アドレス・ビット) になります。

PC値と設定

PC	設定	備考
00	入力ポート	
01	出力ポート	
10	(予約)	
11	GPIFADR	MODE=10のときのみ有効

#### wValue [3 : 2] … PB (PORTBの動作モード設定)

FX2の汎用I/OポートであるPORTBの入出力方向の設定を行います。MODE = '10' (GPIFモード), または MODE = '11' (スレーブFIFOモード) のときは, PBはFD [7 : 0] (FIFOデータの低位8ビット) になり, 本ビットによる設定は無効です。

PB値と設定

PB	設定
00	入力ポート
01	出力ポート
10	(予約)
11	(予約)

#### wValue [1 : 0] … PA (PORTAの動作モード設定)

FX2の汎用I/OポートであるPORTAの入出力方向を設定します。MODE = '11' (スレーブFIFOモード) のときは, PA [7 : 4], およびPA2がスレーブFIFO制御用の信号になり, 本ビットによる設定は無効です。

PA値と設定

PA	設定
00	入力ポート
01	出力ポート
10	(予約)
11	(予約)

MODE=11 (スレーブFIFO時)の割り付け

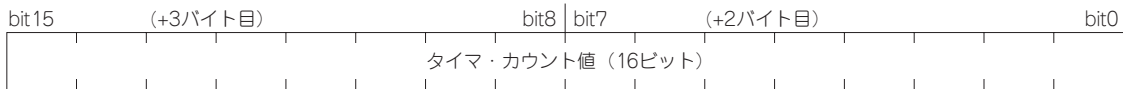
PORTA	信号名
7	SLCS
6	PKTEND
5	FIFOADR1
4	FIFOADR0
3	PA3
2	SLOE
1	PA1
0	PA0

## ■ SetTimer2 (タイマ動作設定, 注: UCT-203では機能しない)

bmRequestType : 0xC0

bRequest : 0x01

wValue

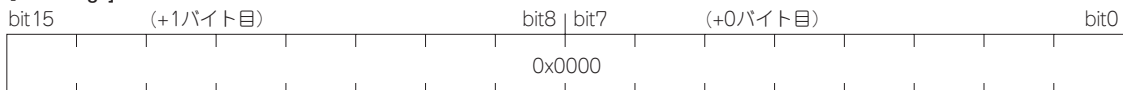


wIndex



wLength : 0x02

[DataStage]



### ●動作

EzFirm/FX2ではFX2内蔵のタイマ2をサポートしています。タイマの動作クロックは外部クロック入力 (T2), または4 MHzか12 MHzの内部クロックから選択できるようになっています。

カウンタはアップカウンタで, 0xFFFFまで進むと次のクロックでwValue値で設定した値が自動的に再ロードされます。このとき1クロック期間だけT2OUT端子がHレベルになります。

T2M/C/T2の設定

T2M	C/T2	信号名
1	1	外部クロック (T2)
1	0	内部クロック (12MHz)
0	1	外部クロック (T2)
0	0	内部クロック (4MHz)

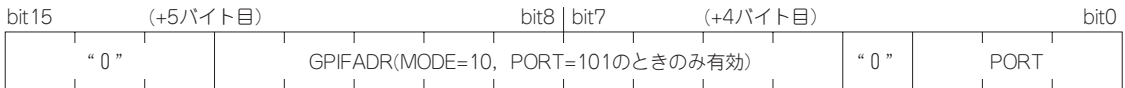
本キットのボード (UCT-203) では機能しません。

## ■ PIOWrite (パラレルI/Oでのデータ出力)

bmRequestType : 0xC0  
 bRequest : 0x02  
 wValue

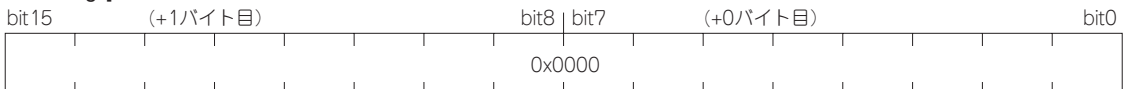


wIndex



wLength : 0x02

[DataStage]



### ●動作

PORTビットで指定されたパラレルI/Oポートにデータ（1バイト）を設定します。該当するポートが入力ポートであったり、動作モードによって別の用途に使用されている場合にはポートから出力されません。MODE = '10' (GPIFモード；SetPortConfigリクエストで設定) かつ PC = '11' (GPIFADR；同) に設定している場合、本コマンドで PORT = '101' (CTLポート) ヘライトする場合、wIndex[12:4]の9ビットはGPIFADR [8:0]への設定値になります。この状態のとき、PORTCはGPIFADRの機能に切り替わっているため、本コマンドでPORTCにライトしても、データは出力されません。

PORTの値と対象ポート

PORT	対象ポート
000	PORTA
001	PORTB
010	PORTC
011	PORTD
100	PORTE
101	RDY
110	(予約)
111	(予約)

\* : UCT-203にはない

## ■ PIORead (パラレルI/Oでのデータ入力)

bmRequestType : 0xC0

bRequest : 0x03

wValue



wIndex



wLength : 0x02

[DataStage]



### ●動作

PORTビットで指定されたパラレルI/Oポートからデータ（1バイト）を読み出し、コントロール伝送のデータINステージでホストに送られます。データINステージでは2バイト（16ビット）のデータが返されますが、上位8ビットは意味をもちません。該当するポートが出力ポートである場合、リードされるデータは無効です。

PORTの値と対象ポート

PORT	対象ポート
000	PORTA
001	PORTB
010	PORTC
011	PORTD
100	PORTE
101	RDY
110	(予約)
111	(予約)

\* : UCT-203にはない



## ■ SetWaveData (ウェーブフォーム・ディスクリプタ・データの設定)

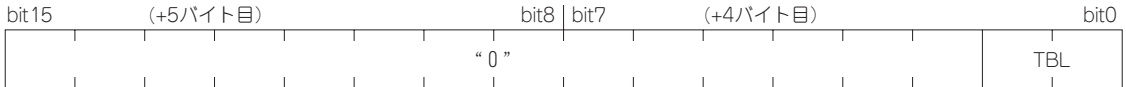
bmRequestType : 0x40

bRequest : 0x05

wValue

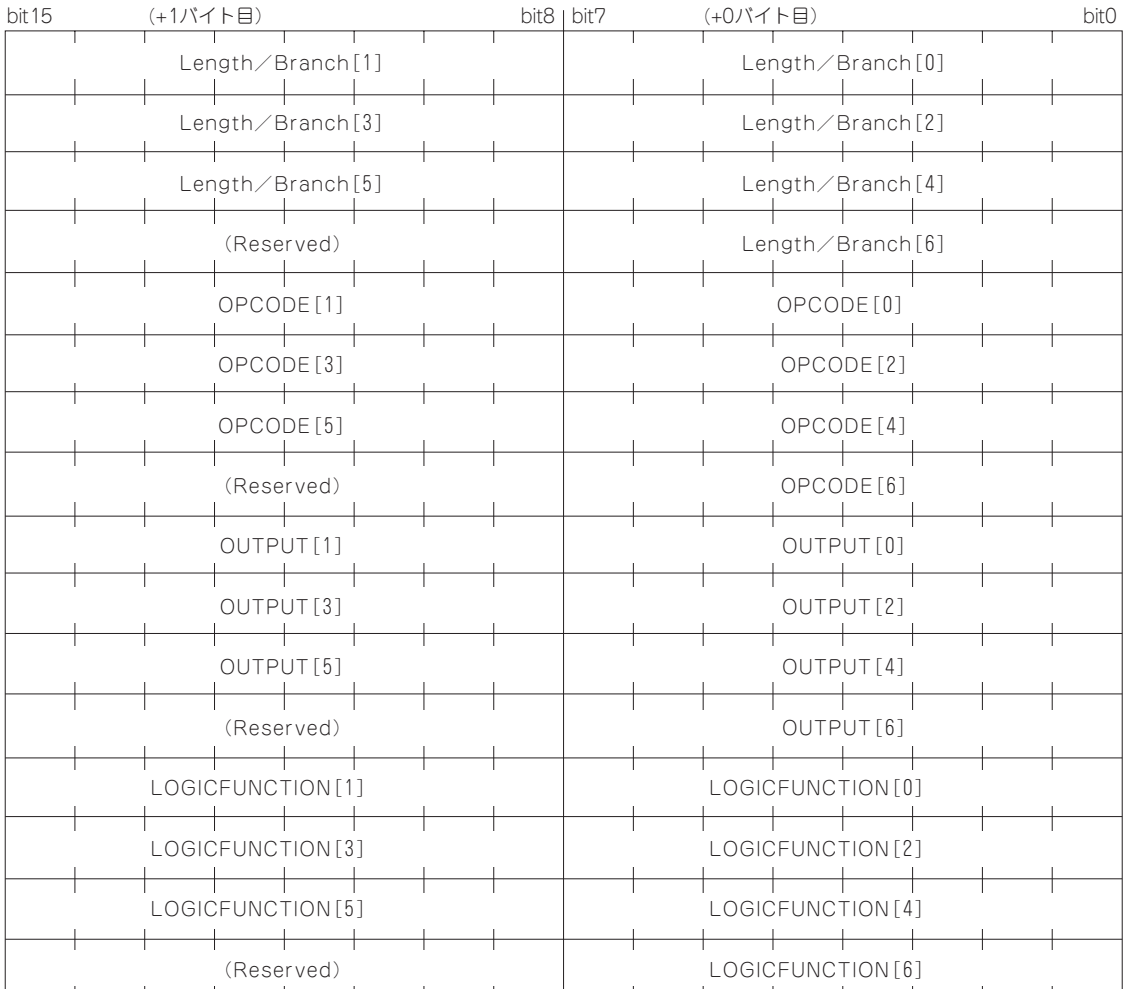


wIndex



wLength : 0x20

[DataStage]



**●動作**

GPIFは四つの動作モード（バースト・リード、バースト・ライト、シングル・リード、シングル・ライト）をもち、それぞれの動作モードに対応した場所にあるテーブル（ウェーブフォーム・ディスクリプタ）を参照しながら動作します。

本コマンドは、**TBL**ビットで指定された動作モード用のテーブルに、ウェーブフォーム・ディスクリプタ・データを設定します。ディスクリプタ・データは32バイトで、コントロール伝送のデータOUTフェーズで転送されます。

TBLの値と設定対象テーブル

TBL	テーブル
00	バースト・リード
01	バースト・ライト
10	シングル・リード
11	シングル・ライト

## ■ GPIFAbort (GPIF 動作の強制停止)

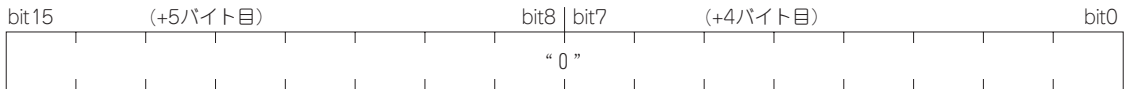
bmRequestType : 0xC0

bRequest : 0x07

wValue

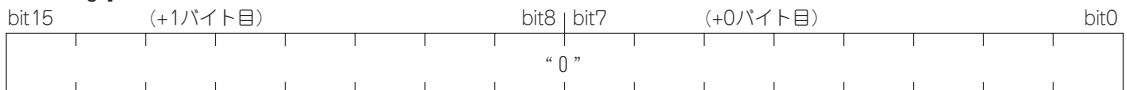


wIndex



wLength : 0x02

[DataStage]



### ●動作

動作中のGPIFを強制停止させて、アイドル・ステート（ステート7）に復帰させます。GPIFが停止中に発行してもかまいません。

SetWaveData リクエストでウェーブフォーム・ディスクリプタを書き換えたり、SetModeConfigなどで動作モードを切り替えるときには、必ずGPIFを停止させた状態で行うようにしてください。

## ■ GPIFSingleWrite (GPIF シングル・ライト実行)

bmRequestType : 0xC0

bRequest : 0x08

wValue

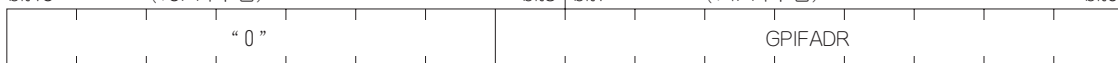


wIndex

bit15 (+5バイト目)

bit8 | bit7 (+4バイト目)

bit0



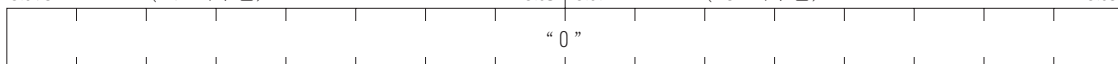
wLength : 0x02

[DataStage]

bit15 (+1バイト目)

bit8 | bit7 (+0バイト目)

bit0



### ●動作

GPIFを使ってシングル・ライト動作を行います。wValueに設定した値がFD [15 : 0] 端子に出力されるデータになります。また、SetPortConfigリクエストによってPORTCをGPIFADRに設定している場合、GPIFADRに設定した値がPORTC (GPIFADR [7 : 0])、およびPORTE.7 (GPIFADR8) に出力されます。

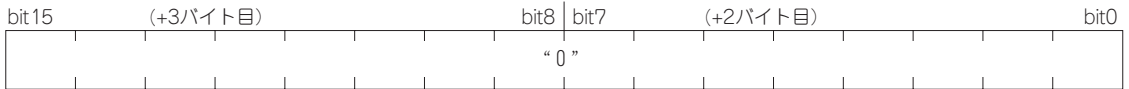
CTL端子の初期状態はPIOWriteコマンドで設定します。

## ■ GPIFSingleRead (GPIF シングル・リード実行)

bmRequestType : 0xC0

bRequest : 0x09

wValue



wIndex



wLength : 0x02

[DataStage]



### ●動作

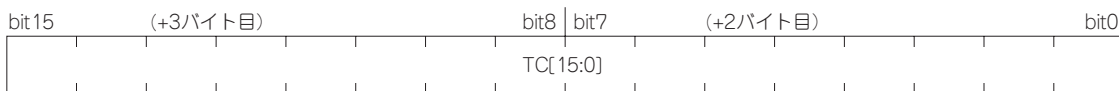
GPIFを使ってシングル・リード動作を行います。リードされた16ビット・データは、つづくデータINステージでホストに渡されます。**SetPortConfig** リクエストによってPORTCをGPIFADRに設定している場合、GPIFADRに設定した値がPORTC (GPIFADR [7:0]), およびPORTE.7 (GPIFADR8) に出力されます。

CTL端子の初期状態は**PIOWrite** コマンドで設定します。

## ■ GPIFTrig (GPIFバースト転送)

bmRequestType : 0xC0  
 bRequest : 0x10 (EP2OUT)  
           0x12 (EP6IN)

wValue



wIndex



wLength : 0x02

[DataStage]



### ●動作

GPIFを使ったバースト転送開始を指示します。TC[31:0]には転送回数を指定します。転送されるデータ・サイズはカウント値×2バイトになることに注意してください。

bRequestが0x10 ('00010000') のときは、本リクエスト発行後にバルク OUT エンド・ポイント (EP6) に到達したデータは、GPIFによって自動的に出力 (バースト OUT) されます。

bRequestが0x12 ('00010010') のときは、本リクエスト発行時にバースト IN 転送がはじまり、FD [15:0] から読み出されたデータが自動的にバルク IN エンド・ポイント (EP6) に転送されます。

指定されたカウント数回の伝送動作を行ったあと、GPIFは自動的に停止します。

EzFirm/FX2では、バルク IN エンド・ポイントを4バンク (Quad Buffer) モードで使用しています。このため本コマンドでバースト IN を指定した場合、エンド・ポイント・サイズ×4 (バイト) ぶんのデータまでは、ホストがバルク IN エンド・ポイントを読まなくても転送されます。

GPIFADR 出力を行っている場合には、PIOwrite コマンドでCTL端子のアイドル時の状態とともに設定します。