

1. PX-1200 の概要

PX-1200 は、シグナルプロセッサ Aproc-1 Plus の拡張インターフェースを手軽に利用できるように準備されたプロトタイプ作成ボードです。PX-1200 の構成を 図 1 に、外形・寸法を 図 2 に示します。PX-1200 に実装されている論理デバイス CPLD に所望のロジックを書き込みますと、任意のデジタル信号を Aproc-1 Plus のプログラムから入出力することができるようになります。信号処理システム開発ツール AprocS には、PX-1200 を経由して信号を入出力するための関数と動作条件を設定する宣言文が準備されていますので、ユーザは PX-1200 を使った信号処理プログラムを容易に作成することができます。

PX-1200 の主な仕様を 表 1 および 表 2 に示します。

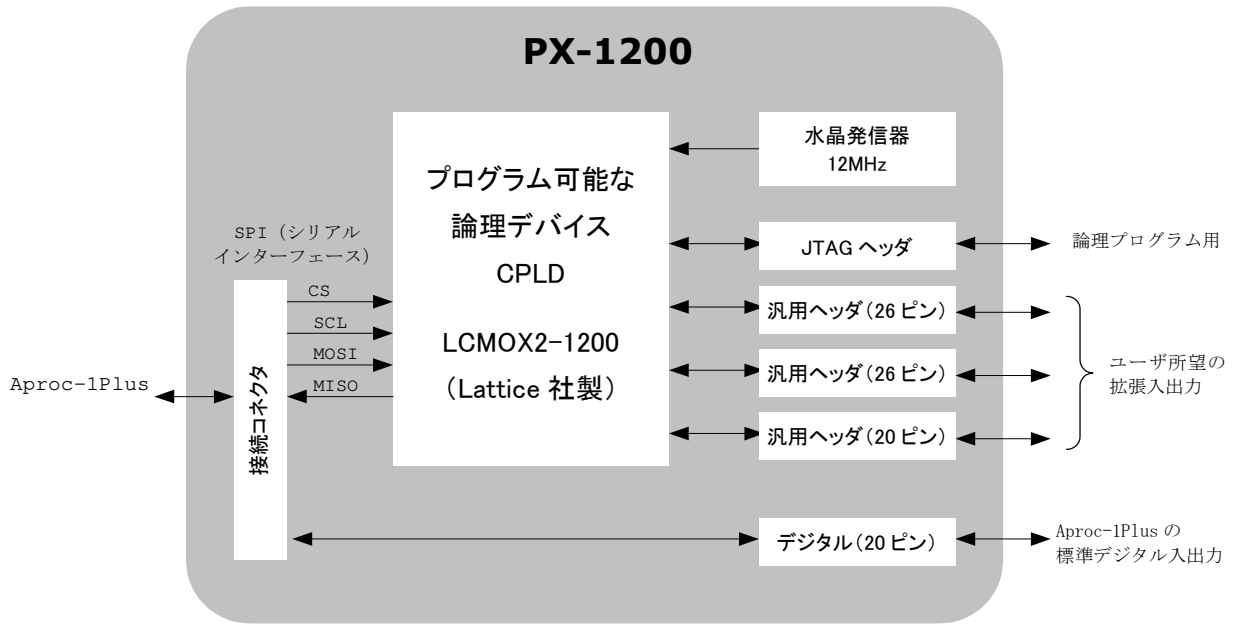


図 1 PX-1200 のブロックダイアグラム

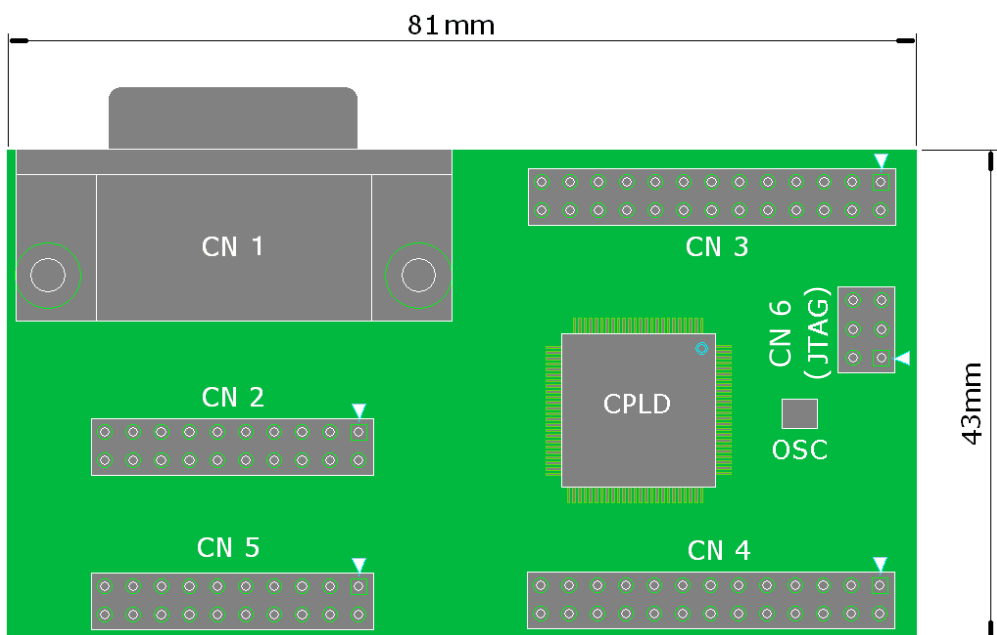


図 2 PX-1200 の外形・寸法および部品配置

2. PX-1200 の仕様

表 1 は、PX-1200 の機械的仕様です。コネクタの番号 CN1～6 は 図 2 に対応しています。

表 1 PX-1200 の機械的仕様

項目	仕様	備考
コネクタ	CN1 : 26 ピン倍密度 DSUB (プラグ)	Aproc-1Plus との接続用
	CN2 : 20 ピンヘッダ (2.54mm ピッチ 10×2)	Aproc-1Plus の標準 DIO 用
	CN3 : 26 ピンヘッダ (2.54mm ピッチ 13×2)	ユーザ信号用 (表 3 参照)
	CN4 : 26 ピンヘッダ (2.54mm ピッチ 13×2)	
	CN5 : 20 ピンヘッダ (2.54mm ピッチ 10×2)	
	CN6 : 6 ピンヘッダ (2.54mm ピッチ 3×2)	JTAG プログラミング用
寸法	81mm×43mm、高さ 15mm 以下	図 2 参照
重量	約 50g	

表 2 は PX-1200 の電氣的仕様、表 3 は各コネクタのピン対応表です。

表 2 PX-1200 の電氣的仕様

項目	仕様	備考
ホストインターフェース	同期シリアル通信 (SPI)	Aproc-1Plus がマスタになります
SPI 通信クロック	10.5MHz	マスタより供給されます
発信器周波数	12MHz	PLL-IP により周波数変換可能
CPLD 型名	LCMXO2-1200-5	Lattice 社製 CLPD
入出力信号数	60	表 3 参照
プログラミング方式	JTAG	
電源電圧	5V	Aproc-1 Plus より供給

PX-1200 では、Aproc-1 Plus の標準デジタル入出力信号がコネクタ CN2 へ延長されていますので、PX-1200 を接続した状態で標準入出力も使うことができます。コネクタ CN2 のピンアサインを表 4 に示します。

PX-1200 の回路接続を 図 3 に示します。

表 3 コネクタ CN3, CN4, CN5, CN6 各ピンの接続先 CPLD ピン番号

コネクタ ピン番号	CN3 に接続された CPLD のピン番号	CN4 に接続された CPLD のピン番号	CN5 に接続された CPLD のピン番号	CN6 に接続された CPLD のピン番号
1	96	75	(GND)	90 (TMS)
2	97	74	(GND)	91 (TCK)
3	98	71	47	92 (GND)
4	99	70	48	93 (VIO)
5	1	69	43	94 (TDI)
6	2	68	45	95 (TDO)
7	3	67	41	
8	4	66	42	
9	(GND)	(GND)	39	
10	(GND)	(GND)	40	
11	7	65	(GND)	
12	8	64	(GND)	
13	9	63	37	
14	10	62	38	
15	12	61	35	
16	13	60	36	
17	14	59	32	
18	15	58	34	
19	(GND)	(GND)	29	
20	(GND)	(GND)	30	
21	16	57		
22	17	54		
23	18	53		
24	19	52		
25	21	51		
26	24	49		

表 4 コネクタ CN2 のピン信号

番号	記号	信号名	備考
1	GND	グラウンド	
2			
3	Din7	デジタル入力 7 またはエンコーダ入力 B3	エンコーダ入力と兼用 LVTTTL レベル (3.3V) ただし 5V まで許容
4	Din6	デジタル入力 6 またはエンコーダ入力 A3	
5	Din5	デジタル入力 5 またはエンコーダ B2 入力	
6	Din4	デジタル入力 4 またはエンコーダ A2 入力	
7	Din3	デジタル入力 3 またはエンコーダ B1 入力	
8	Din2	デジタル入力 2 またはエンコーダ A1 入力	
9	Din1	デジタル入力 1 またはエンコーダ B0 入力	
10	Din0	デジタル入力 0 またはエンコーダ A0 入力	
11	GND	グラウンド	
12			
13	Dout7	デジタル出力 7 または PWM 出力 7	PWM 出力と兼用 LVTTTL レベル (3.3V)
14	Dout6	デジタル出力 6 または PWM 出力 6	
15	Dout5	デジタル出力 5 または PWM 出力 5	
16	Dout4	デジタル出力 4 または PWM 出力 4	
17	Dout3	デジタル出力 3 または PWM 出力 3	
18	Dout2	デジタル出力 2 または PWM 出力 2	
19	Dout1	デジタル出力 1 または PWM 出力 1	
20	Dout0	デジタル出力 0 または PWM 出力 0	

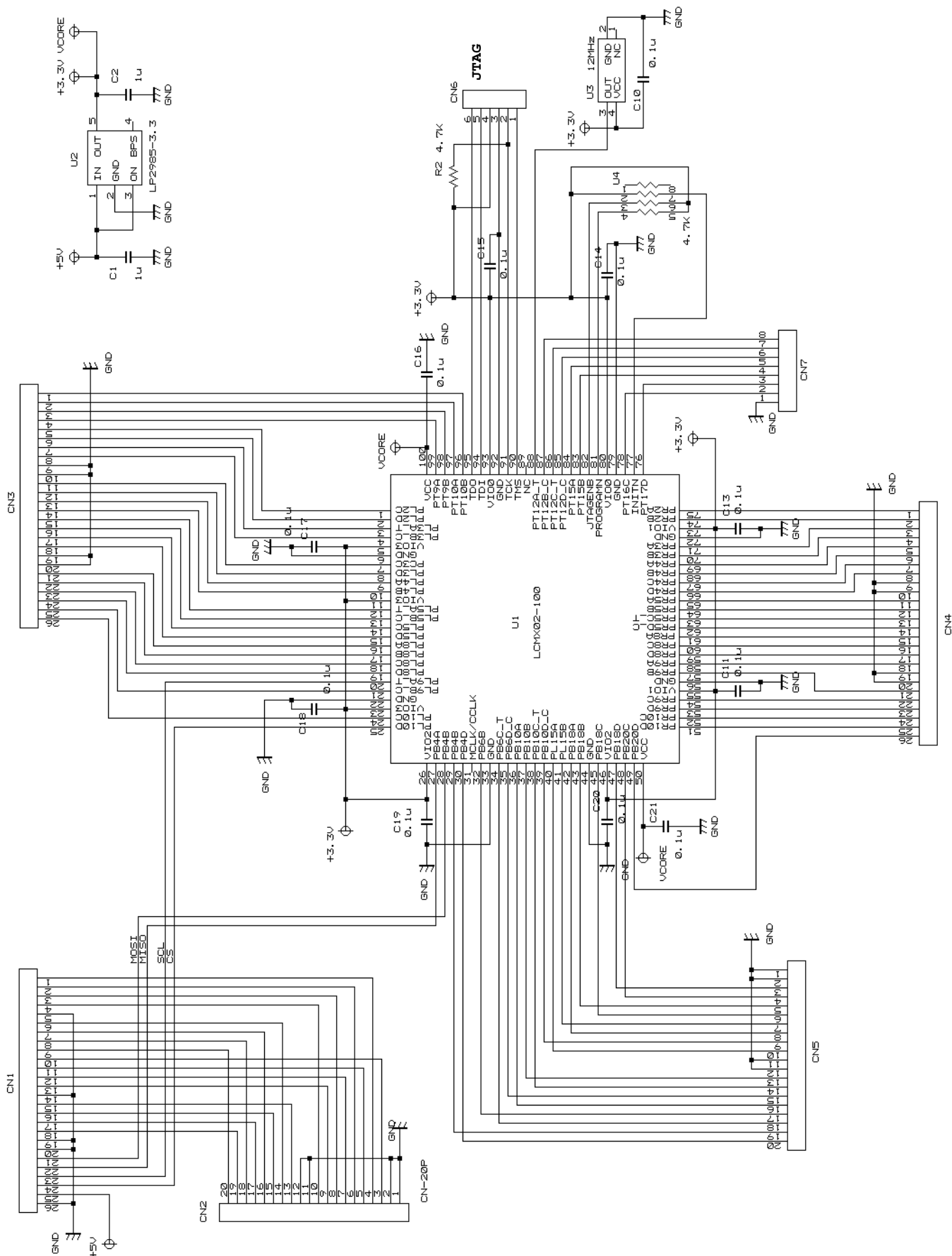


図 3 PX-1200 回路接続

3. プロジェクトの作成例

(1) 設計仕様

PX-1200 の使用例として、**図 4** に示す機能を PX-1200 の CPLD に作成し、そのプロジェクト名を DX-001 とします。DX-001 は以下の機能を有します。

- (1) アップダウンカウンタは指定された周期の三角波を発生します。
- (2) SPI (シリアル信号インターフェース) を介して Aproc1 Plus から受信した 3 つのパルス幅指令 pwm () は、それぞれパルス幅レジスタに格納されます。
- (3) 三角波のカウント値とパルス幅指令を比較して、互いに逆位相のパルス幅変調波 pout () と nout () を生成します。
- (4) 内部信号 pout () をそのまま信号 ref () として出力します。
- (5) 内部信号 pout () の 0→1 と変化する時刻を指定時間遅らせた信号 pdout () を出力します。
- (6) 内部信号 nout () の 0→1 と変化する時刻を指定時間遅らせた信号 ndout () を出力します。
- (7) 8 ビットのデジタル信号 ByteIn () を入力し、Aproc-1 Plus へ送ります。

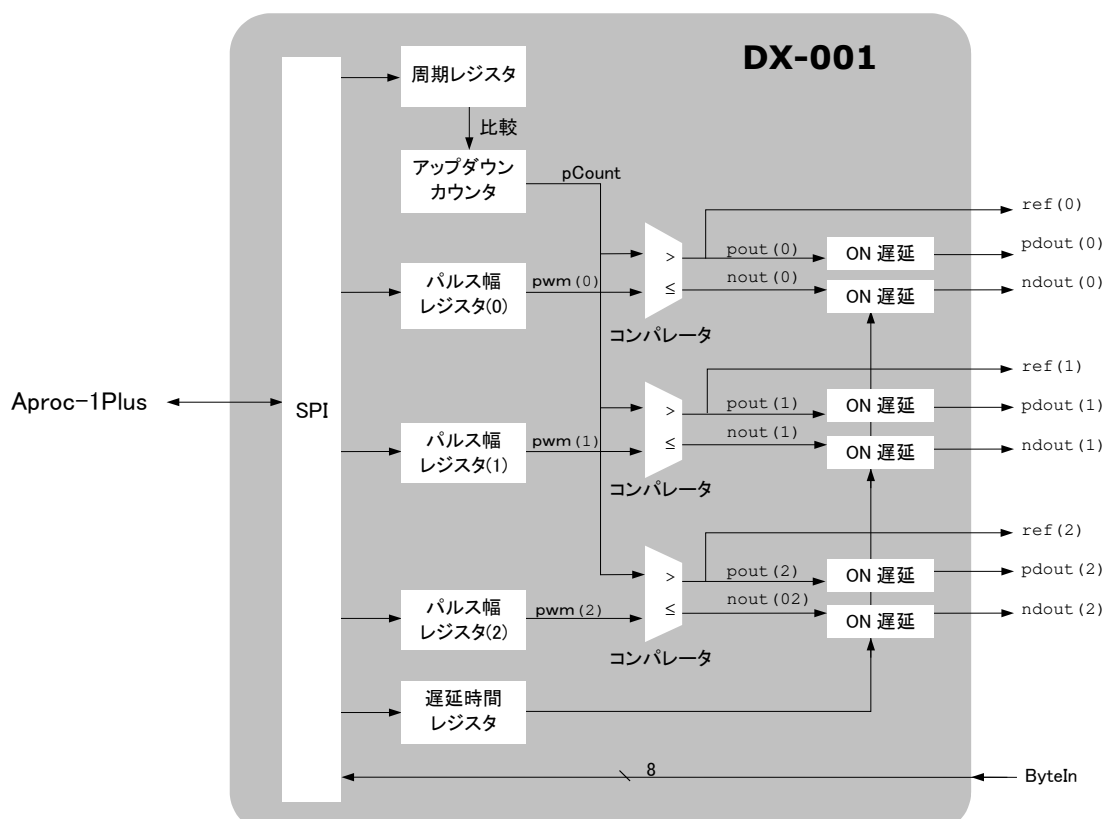


図 4 プロジェクト DX-001 の機能ブロック図

(2) Aproc-1 Plus との通信処理

Aprioc-1Plus からは SPI 形式で 3 種類のコマンドのいずれかが送られてきます。各コマンドについて PX-1200 側が行うべき処置を以下に説明します。また、Dx-001 での処理を **表 5** に示します。Aproc-1 Plus の SPI 信号については、Aproc-1Plus のデータシートを参照してください。

1) プロジェクト ID 要求と初期設定コマンド (ID 要求、ヘッダ値=0xC3)

ユーザプログラムの運転開始時にこのコマンドを受信します。このとき Dx-001 はプロジェクト ID ('D'で始まる 4 桁の 16 進数) を返信し、ユーザプログラムからデータがあればそれらを初期設定に使用します。Aproc-1 Plus 側では、返信で得た ID と xType 宣言の ID とが一致しない場合、エ

ラーとなります。

2) 拡張信号入力コマンド (入力指令、ヘッダ値=0xC5)

AprocS のユーザプログラムに拡張信号入力の定義がある場合、毎運転サイクル開始時 (A/D 変換開始とほぼ同時刻) にこのコマンドを受信します。これに対し PX-1200 のプロジェクトは、入力したデータ、またはその入力データに係わる処理結果などを Aproc-1 Plus へ返信します。

3) 拡張信号出力コマンド (出力指令、ヘッダ値=0xC9)

AprocS のユーザプログラムに拡張信号出力の定義がある場合、毎運転サイクル終了時 (D/A 変換開始とほぼ同時刻) にこのコマンドを受信します。これに対し、PX-1200 のプロジェクトは、受信したデータ、またはそのデータに係わる処理結果を CPLD から外部へ出力します。

表 5 SPI 通信のコマンドと DX-001 での処理

コマンド	ヘッダ	説明	DX-001 での処理
ID 要求	C3 ₁₆	xType 宣言の ID 値を返信し、必要なら受信した値で動作条件等を設定します	ID 値 0xD001 を返信し、受信した 3 バイトを PWM 周期レジスタ (16bit) と ON 遅延時間レジスタ (8bit) へ格納します。
拡張信号入力	C5 ₁₆	Aproc-1 Plus へ送信した数値が関数 Xin () の戻り値になります。信号数は AprocS のユーザプログラムで宣言します。	8bit ポート ByteIn の入力値を Aproc-1 Plus へ送信します。データの並びはリトルエンディアンなので、16bit データの場合は下位ビットを先に送ります。
拡張信号出力	C9 ₁₆	関数 Xout () の引数値が送られてくるので、それを使って所望の処理を行います。信号数は AprocS のユーザプログラムで宣言します。	受信した 6 バイトを PWM 波への指令値として、3 つの 16bit レジスタへ格納します。リトルエンディアンなので下位バイトを先に読みます。

(3) 三角波生成処理

16bit のアップダウンカウンタが 図 5 の状態遷移で動作します。加算状態のときカウント値が上限値に達すれば減算モードとなり、減算状態のときカウント値が下限値になれば加算モードとなります。周期レジスタに設定した値を上限値、その半分の値を下限値としました。例えば、周期レジスタ値=1200 とすると、カウント値は 600~1200 の範囲で増減を繰り返します。クロック周波数 12MHz なので、三角波の周波数は、12MHz/1200=10KHz となります。

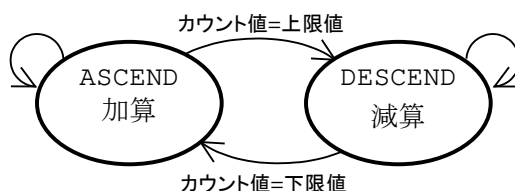


図 5 三角波発生の状態遷移

(4) PWM 生成処理

三角波のカウント値 pCount とパルス幅指令値 pwm () を比較し、pCount > pwm () なら pout ()=1、nout ()=0、さもなければ pout ()=0、nout ()=1 とします。この処理を 3 つの指令値 pwm (0) ~ (2) に対して行い、3 組の pout (0) ~ (2)、nout (0) ~ (2) を得ます。

(5) ON 時間遅延処理

pout、nout が 0→1 と変化する時刻を xType 宣言で指定したサイクル数だけ遅らせます。1→0 と変化する場合はそのまま遅延なく変化させます。この状態遷移を 図 6 に示します。

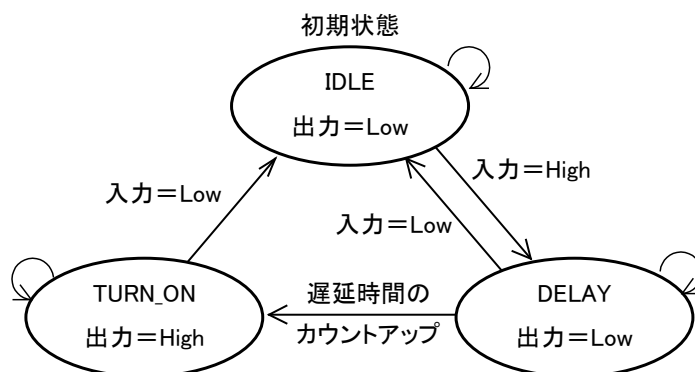


図 6 ON 時間遅延の状態遷移

(6) DX-001 の入出力ピンアサインメント

コネクタ ピン番号 (共通)	CN3		CN4		CN5	
	CPLD ピン番号	Dx001 の 信号名	CPLD ピン番号	Dx001 の 信号名	CPLD ピン番号	Dx001 の 信号名
1	96	ByteIn (0)	75		GND	
2	97	ByteIn (1)	74		GND	
3	98	ByteIn (2)	71		47	
4	99	ByteIn (3)	70		48	
5	1	ByteIn (4)	69		43	
6	2	ByteIn (5)	68		45	pout (0)
7	3	ByteIn (6)	67		41	
8	4	ByteIn (7)	66		42	pout (1)
9	GND		GND		39	
10	GND		GND		40	pout (2)
11	7		65		GND	
12	8		64		GND	
13	9		63		37	
14	10		62	ref (0)	38	nout (0)
15	12		61		35	
16	13		60	ref (1)	36	nout (1)
17	14		59		32	
18	15		58	ref (2)	34	nout (2)
19	GND		GND		29	
20	GND		GND		30	
21	16		57			
22	17		54			
23	18		53			
24	19		52			
25	21		51			
26	24		49			

4. DX-001 を使って入出力を行うユーザプログラムの例

(1) サンプルプログラムのソースコード

Dx001 を PX-1200 上の CPLD へプログラムし、これを Aproc-1 Plus へ接続して **List 1** のプログラムを動作させます。このプログラムは、3 相の正弦波を周波数 10KHz の PWM 波へ変調し、それらの ON 遅延時間を 5 マイクロ秒として出力させます。また、8 ビット入力ポートの数値を 0~255 の整数値として AprocS の操作ウィンドウに表示させます。このプログラムで使われている 3 つの文、宣言文、出力文、および入力文について以下に説明します。

List 1 DX-001 試験用プログラムのソースコード

```
xType = "D001, in=1*1, out=2*3, conf=04-B0-3C"; // ◆ 宣言文
// 0x04B0 = 1200: 周期 1200/12M =100us (10KHz)
// 0x3C = 60 : 遅延 60/12M = 5us
scl = 600;
amp = 0.4;
proc
{
    u = w_sin( period, 0, amp ) + 1.5;
    v = w_sin( period, 1/3, amp ) + 1.5;
    w = w_sin( period, 2/3, amp ) + 1.5;
    Xout( 0, 1/scl, u ); // ◆ 出力文
    Xout( 1, 1/scl, v );
    Xout( 2, 1/scl, w );
    bin = Xin( 0, 1 ); // ◆ 入力文
}
```

【注意】 開発ツール AprocS のバージョンは 2.12 以降となります。

(2) 宣言文 (ID 要求および PWM 周期と遅延時間の設定)

【書式】

```
xType = "ID, in=size*word, out=size*word, conf=x0-x1-x2- ... ";
```

右辺の “ ” で囲まれた文字列はコンマにより区切られた 4 つの文字列を含みます。これらの意味を **表 6** に記します

表 6 設定文字列の意味

文字列	意味	説明
ID	プロジェクトの番号	'D' で始まる 4 桁の 16 進数です。ID 要求コマンドを受信したとき返信する数値がこれと一致しなければエラーになります
in=size*word	入力データのサイズと語数	データサイズが 8bit なら size=1, 16bit なら size=2 で、そのデータの数が word になります。
out=size*word	出力データのサイズと語数	入力、出力のいずれかだけなら、他方は省略します。

文字列	意味	説明
$x_0-x_1-x_2-.....$	初期設定データ	ハイフンで結ばれた 2 桁の 16 進数の並びです。 Aproc-1 Plus はこれを ID 要求コマンドから受信し、プロジェクトは運転開始に必要な初期化を行います。何をどんな順番で並べるかは、プロジェクトがこれらを読む順番に依存します。

List 4 では

```
xType = "D001, in=1*1, out=2*3, conf=04-B8-3C";
```

となっているので、入力は 1 バイトだけ、出力は 2 バイトデータが 3 つになります。また、 x_0-x_1 で表される $0x04B8=1200$ をパルス周期、 $x_2=0x3C=60$ を ON 時間遅延としています。クロックが 12MHz なのでパルス周期は $1200/12M=100\mu s$ (10KHz)、遅延時間は $60/12M=5\mu s$ です。

(3) 出力文 (パルス時間幅の指定)

【書式】

```
Xout( n, scale, var );
```

- 1) n はチャンネル番号を表す番号です。DX-001 の場合、Aproc-1Plus から PX-1200 への出力は 3 つの PWM 指令なので、0、1、2 のいずれかです。宣言文で出力は 16bit としたので
- 2) var は出力したい信号名です。
- 3) scale は PX-1200 が受信するデータの 1 ビットに相当する信号 var の値です。List 1 では、パルス周期が 1200 カウントなので三角波は 600~1200 の範囲で変化します。よって、 $scale=1/600$ とすれば、var=1~2 の範囲で与えたとき PWM のデューティ比は 0~1 の範囲で変化することになります。amp=0.4 なら、正弦波信号 u,v,w の値域は 1.1~1.9 ですから、デューティ比は 0.1~0.9 で変化します。

(4) 入力文 (8 ビットデジタルデータの入力)

【書式】

```
var = Xin( n, scale );
```

- 1) n はチャンネル番号を表す番号です。DX-001 の場合、入力は 1 バイトだけなので、n = 0 しかありません。16bit 単位のデータの場合、例えば xType 文で in=2*4 などと宣言しておけば、16bit 毎の値として入力しますから、その場合、n=0,1,2,3 と 4 回の読出しで全 8 バイトのデータ入力となります。なお、データの並びはリトルエンディアンなので、16bit データの場合は下位 8bit を先に送信するよう CPLD の論理を作成してください。
- 2) scale は入力値 1bit に相当する信号 var の値です。List 1 では、8 ビットデータをそのまま 0~255 で表示しますが、それを 0.0~1.0 の数 (1.0 は含まず) としたいなら

```
bin = Xin ( 0, 1/256 ) ;
```

と記述します。

5. 改訂履歴

バージョン	日付	説明
1.0	Jan 22, 2021	初版
1.2	April 09, 2021	表4の誤りを訂正しました 図Aの誤りを訂正しました

付録1 DX-001のVHDLソースコード

下記からダウンロードできます。

<https://e-arima.com/Aproc-1/exif/dx001.zip>

付録2 CPLDの書き込み回路例

図Aは、下記資料を参考にしてPX-1200用に簡素化したものです。

http://www.vabolis.lt/stuff/lattice_ispdownload.pdf

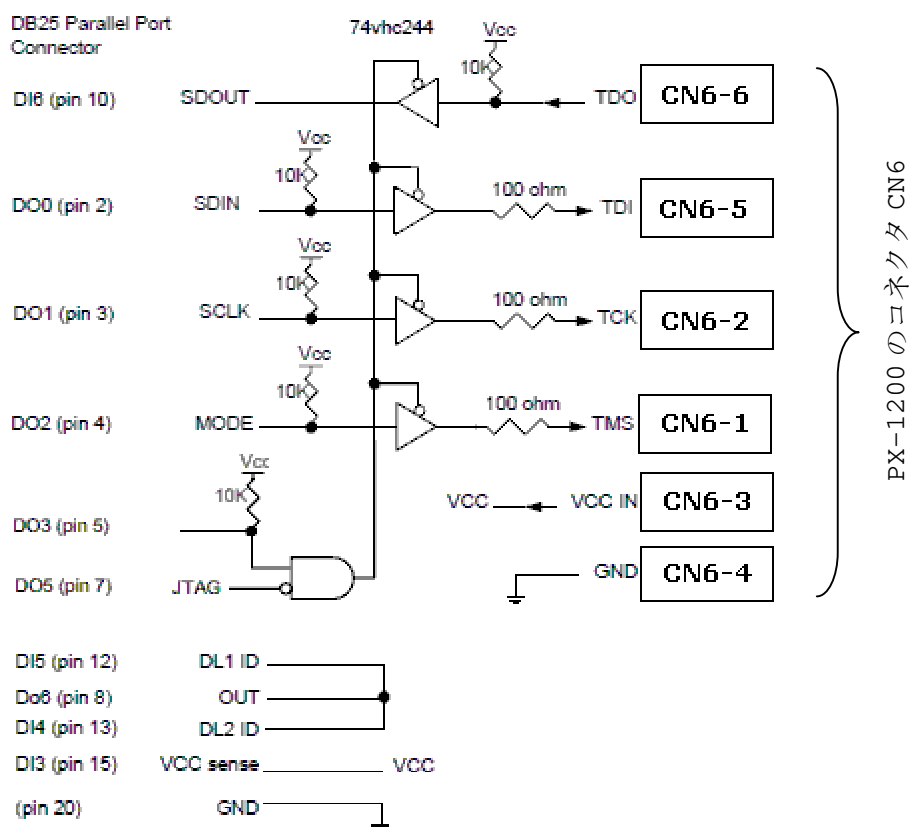


図 A CPLD プログラミング用ライター回路例