

USB デジタル I/O ボードの使い方

~簡単に使える I/O ボードを目指して~

【概要】

Smart-IO2 デジタル I/O ボード(以下、I/O ボード)は、USB を利用して PC からデジタル I/O を簡単に制御できるボードです。1 ポートあたり 8 ビットの I/O を 8 ポート備え、合計 64 ビットの I/O 制御ができます。40pin コネクタ CN1 と CN2 に 4 ポートずつ收容しています。I/O 電圧は 3.3V ですが、5V-I/O トレラントなので既存の 5V-TTL システムとインタフェースが可能です。

この資料では、I/O ボード付属の制御アプリケーションソフト”RefIO7.exe”により、制御ソフトをカスタマイズすることなく、様々な制御ができることを示します。

【制御アプリケーションソフトによるレジスタ制御】

I/O ボードには、以下の内蔵レジスタがあり、それぞれのレジスタを ”RefIO7.exe” から制御できます。

ポート	レジスタ No.	用途	Bit	I/O 方向	備考
A	0	汎用	8	R/W	ボードのコネクタ CN1 と CN2 に接続する I/O。合計 64 本。
B	1		8	R/W	
C	2		8	R/W	
D	3		8	R/W	
E	4		8	R/W	
F	5		8	R/W	
G	6		8	R/W	
H	7		8	R/W	
A-OE	8	各ポートに対応したアウトプット・インネーブル設定	8	R/W	対応する bit を”1”に設定すると、その I/O は出力になる。リセットや電源投入直後は、すべてのレジスタ値は”0”のため、全ての I/O ピンは入力になります。
B-OE	9		8	R/W	
C-OE	10		8	R/W	
D-OE	11		8	R/W	
E-OE	12		8	R/W	
F-OE	13		8	R/W	
G-OE	14		8	R/W	
H-OE	15		8	R/W	
SW	16	ボード上のスイッチ状態読み取り	5	R	タクト SW1個、ディップ SW 4bit
ADRS	17	ボード・アドレス	4	R	0~F 出力のロータリーSW
LED	18	ボード上の LED 表示	8	R/W	高輝度 LED 8個

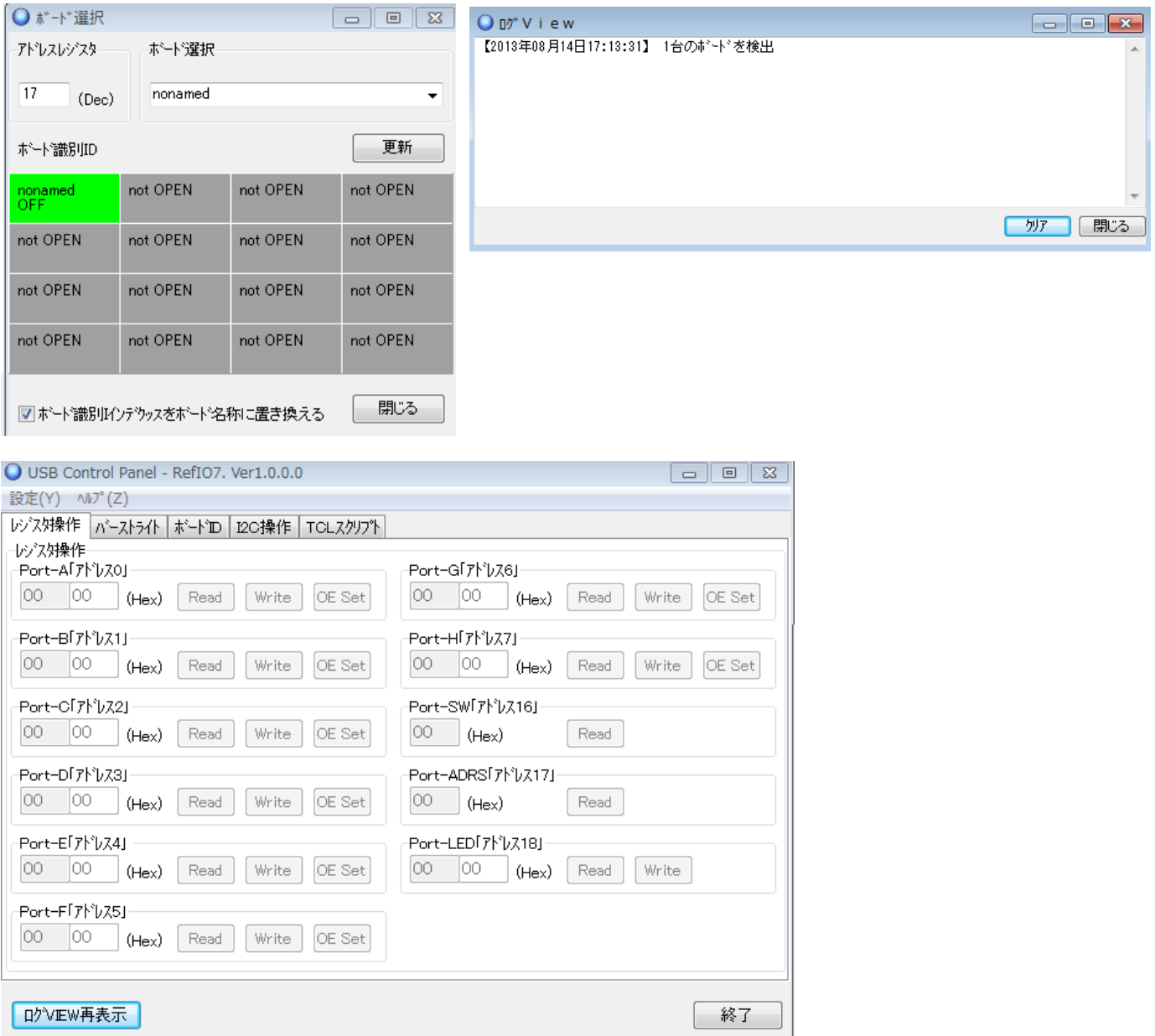
<表1. I/O ボードのアドレスマップ>

【RefIO7.exe の概要】

RefIO7.exe 制御ソフトでは、I/O ボードを最大 16 台まで制御できます。複数台の同じ I/O ボードを識別するために、I/O ボード上のロータリースイッチ(SW1)でボードアドレスを設定します。制御ソフトでは、ボードアドレスに対応した I/O ボードに対して制御を行います。I/O ボードが 1 台の場合にはボードアドレスは Don't care です。

RefIO7.exe は、3 つのパネルで構成しています。

1. **ボード選択パネル** RefIO7 インストールフォルダ内の Index.csv ファイルと連携しています。
2. **USB 制御パネル** 実際に I/O を制御するパネルです。
3. **ログ View パネル** 制御した履歴を閲覧できます。



<図1. RefIO7 パネル構成>

【RefIO7.exe の GUI】

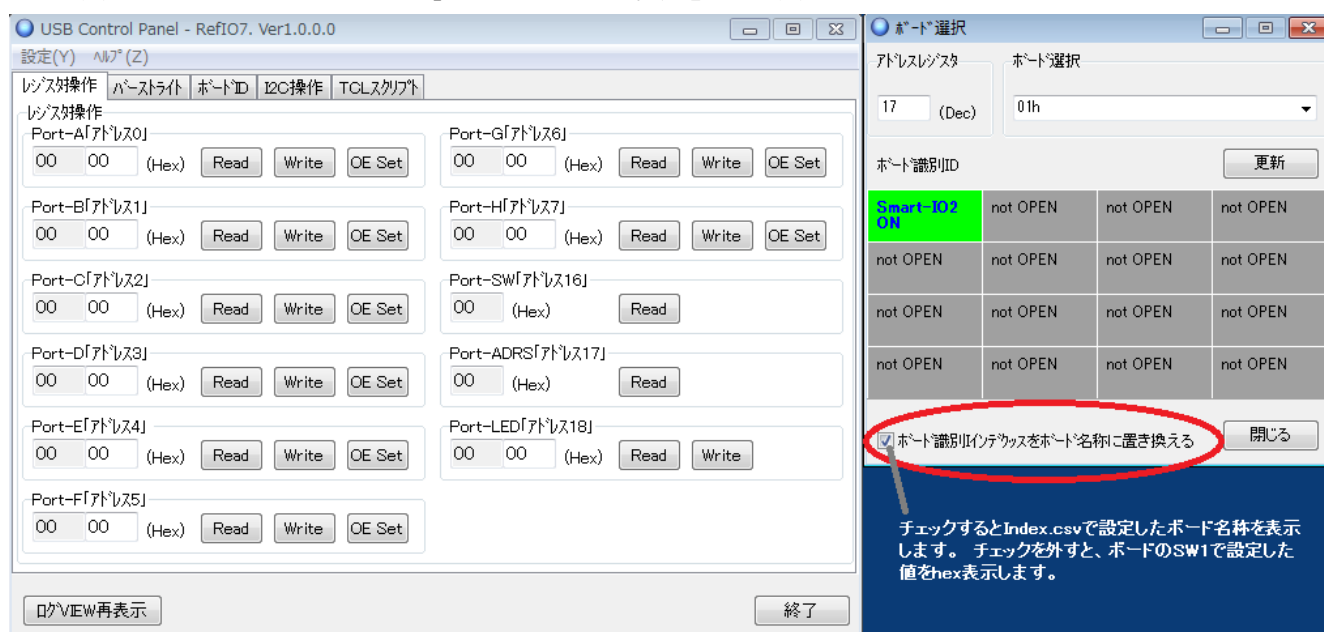
<ボード制御の準備>

RefIO7.exe のインストールフォルダには、Index.csv ファイルがあります。このファイルでは、ボードのアドレスに関連づけしたボード名称の設定ができます。例えば、ボード上のロータリースイッチを“1”にセットし、アドレス“1”のボード名称を Smart-IO2 とした場合、RefIO7.exe のボード選択パネルには、“Smart-IO2”と表示されます。

Index.csv ファイルフォーマット: ボードアドレス No. , ボード名称

```
0,nonamed
1,Smart-IO2
2,nonamed
3,nonamed
4,nonamed
```

RefIO7.exe を起動し、図 2、右側の「ボード選択」パネルから、制御するボードを指定します。この例では PC に 1 台のボードだけを USB 接続しています。Index.csv で設定した“Smart-io2”という名称のボードを接続しているため、欄が緑に変化し、“ON”表示されています。この状態で「コントロールパネル」によりレジスタへの設定を行います。

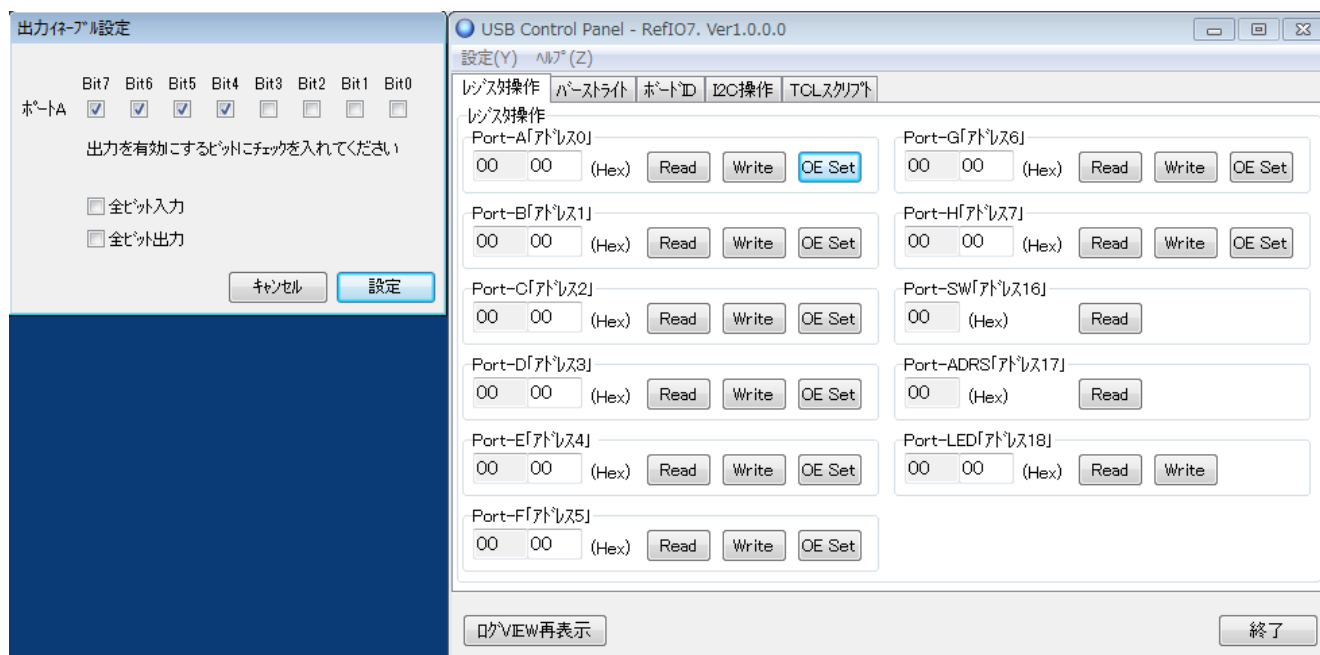


<図2. コントロールパネルとボード選択パネル>

※ ボードの複数台制御については、[こちらの資料](#)も参考にしてください。

<ポート設定>

電源投入時とボードリセット時、汎用ポートの A~H レジスタはすべて入力になっています。I/O ボードのコネクタ CN1 と CN2 に何も接続していない場合に「Read」すると不定値が得られます。ポートを出力として運用する場合には、各ポートの「OE Set」ボタンをクリックして、設定してください。



＜図3. 出力イネーブル設定画面＞

図3の例では、ポートAの上位4bitだけ出力に設定し、下位4bitは入力に設定しています。すべてのビットを出力に設定したい場合は、「全ビット出力」を1箇所チェックしてください。

＜ポート制御＞

ポートの制御方向を決めた後は、それぞれのポートで「Write」または「Read」ボタンをクリックして制御を行います。Aポートを「全ビット出力」にした場合、「FF」を「Write」ボタンで書き込むと、ポートAのI/Oピンを収容するCN1の対応するI/OピンがHighレベルになります。ポートを出力に設定した直後は初期値の「00」なので、コネクタCN1の対応するI/OピンはLowレベルになっています。

＜バーストライト機能＞

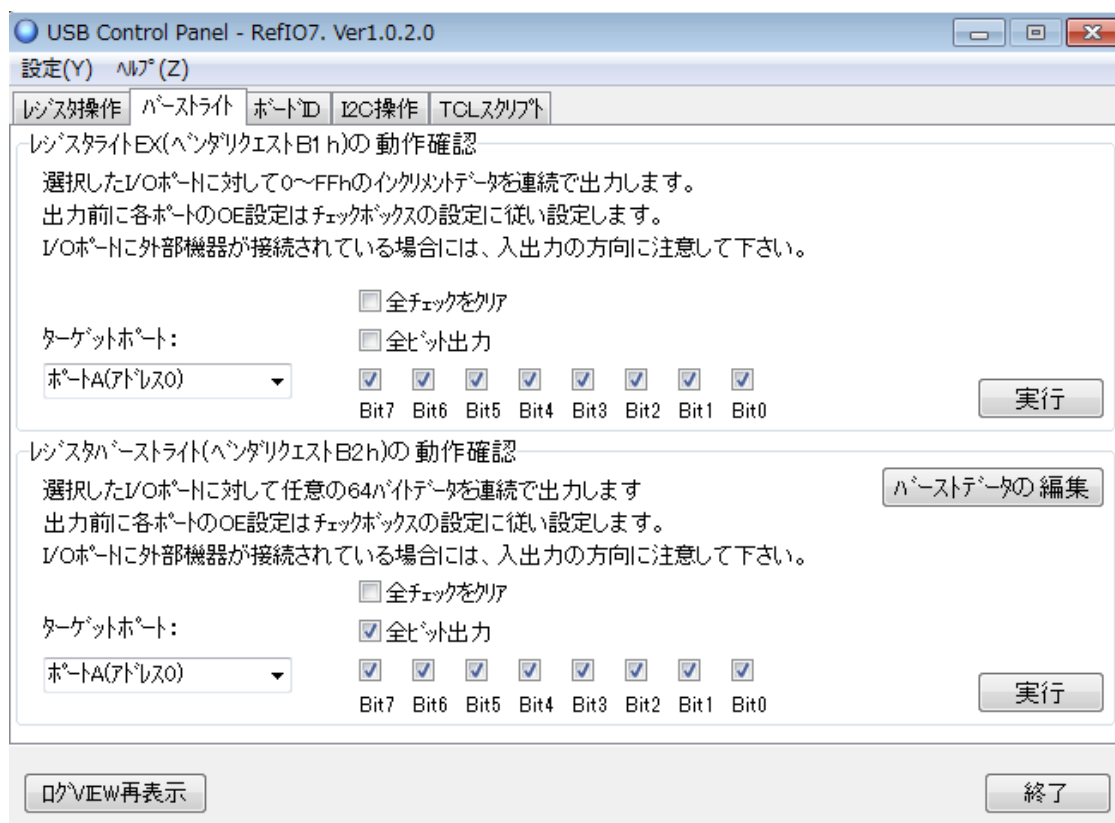
ポートA～Hのいずれか1つのポートに、64バイトの連続したデータを書き込むことができます。試験用として、2種類のバーストライトができます。

① レジスタライト EX(ベンダリクエスト B1h)の動作確認

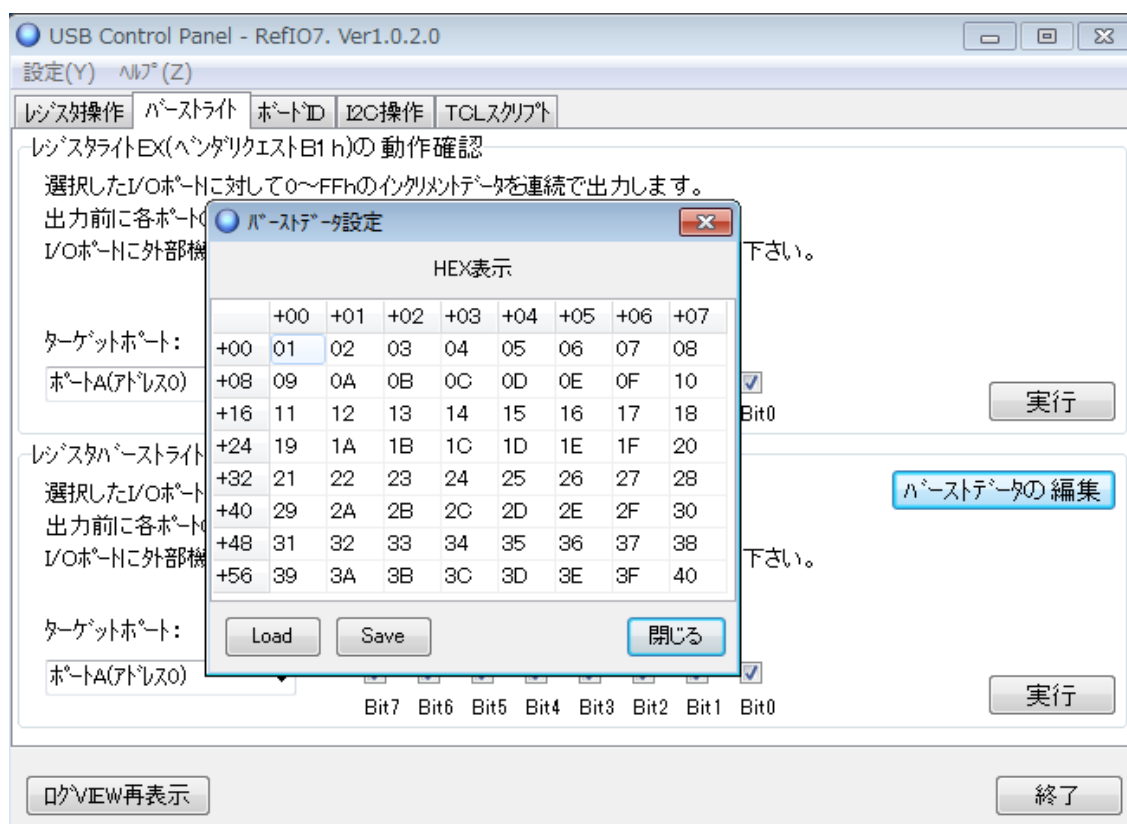
ここでは、指定したポートに、インクリメントするデータを出力できます。インクリメントデータ以外のデータ設定はできません。通常のレジスタライト・コマンドにくらべてUSBバス帯域に占める割合が減る関数になっています。コマンド発行間隔はソフトウェア処理なので、コマンド発行間隔が常に均一になりません。制御ソフトをカスタマイズする場合には、このコマンド形式が便利です。

② レジスタバーストライト((ベンダリクエスト B2h)の動作確認

指定したポートに64バイト長のデータを一括出力できます。図5のようにバーストデータを編集して、信号出力ができます。ソフトウェアでは1回のコマンド発行になり、ハードウェア側で64バイトのデータを処理するので、信号周期はほぼ一定です。図6を参照してください。

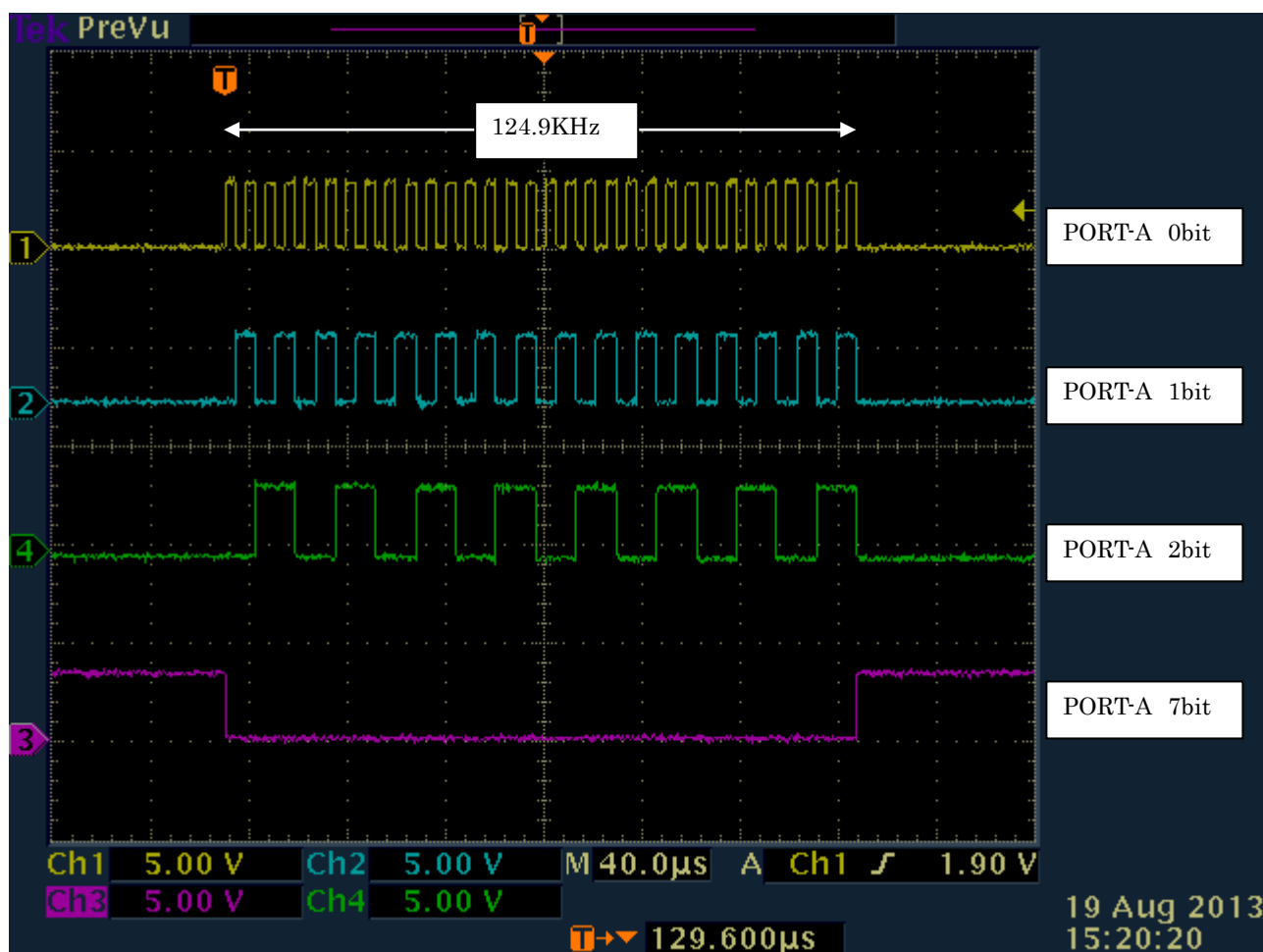


<図4. バーストライト機能パネル>



<図5. バーストデータの編集>

図5は、ポート A にカウントアップする数値を書き込んだ結果です。64 バイトの連続データ書き込みなので、bit0 には 32 個のパルスを観測できます。

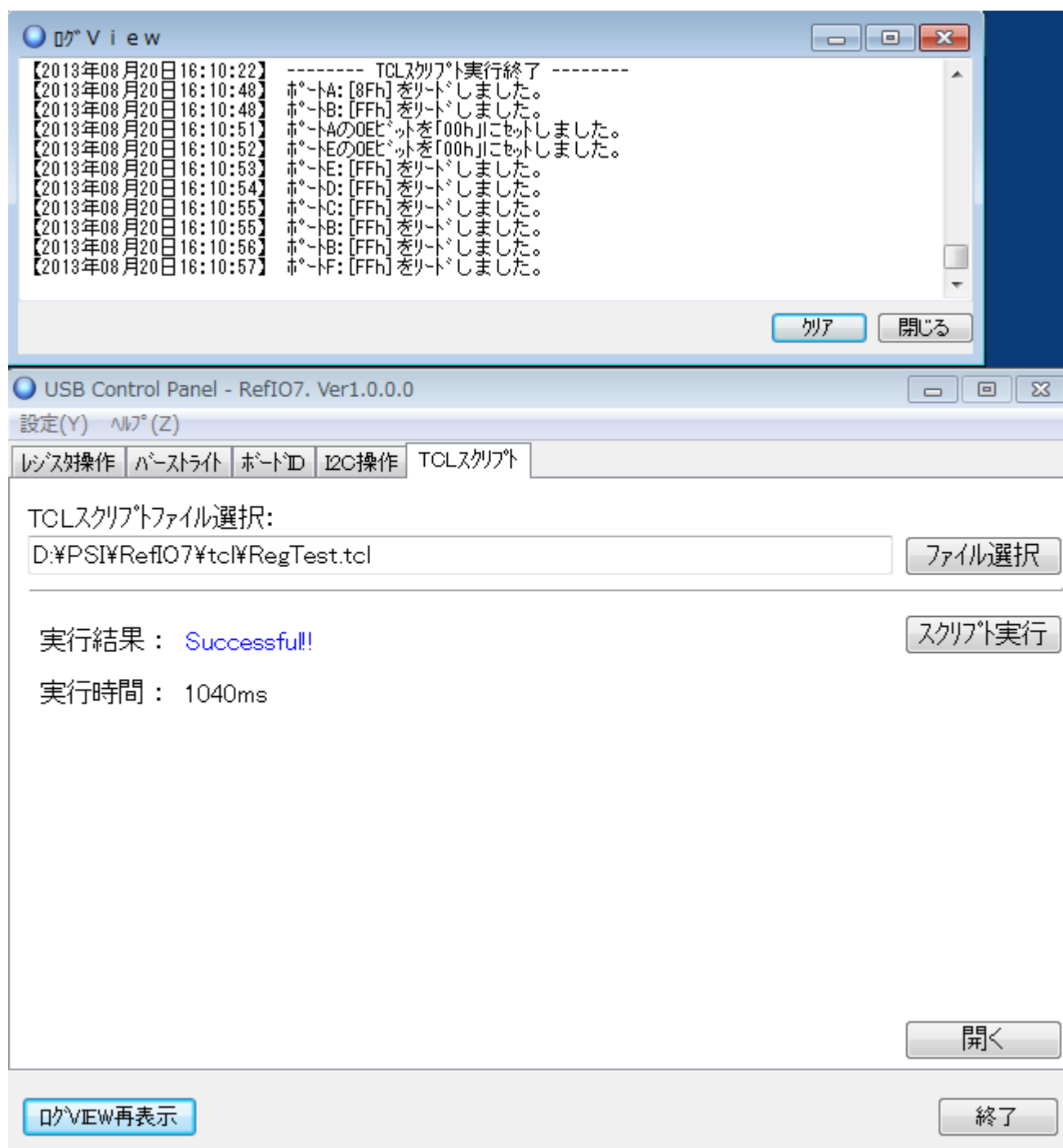


<図6. パーストライト時の波形 (CN1 の対応ピンで観測)>

【RefIO7.exe の TCL スクリプト機能】

RefIO7.exe には、一連のレジスタ設定をバッチ・ファイルとして取り扱うことができる「TCL 機能」を備えています。図5に「TCL 機能」パネルを示します。

TCL 機能はだれでも簡単にテキストエディタで記述できるスクリプトです。図5では、レジスタ番号“4”(ポート A)への書き込みを記述した Sample.tcl (表2参照)を実行した結果です。図6に示す観測波形は、TCL 機能により、ポート A の 4 ビット分にパルス波形を 8 個出力した例です(I/O ボードの CN1 で観測した結果です)。

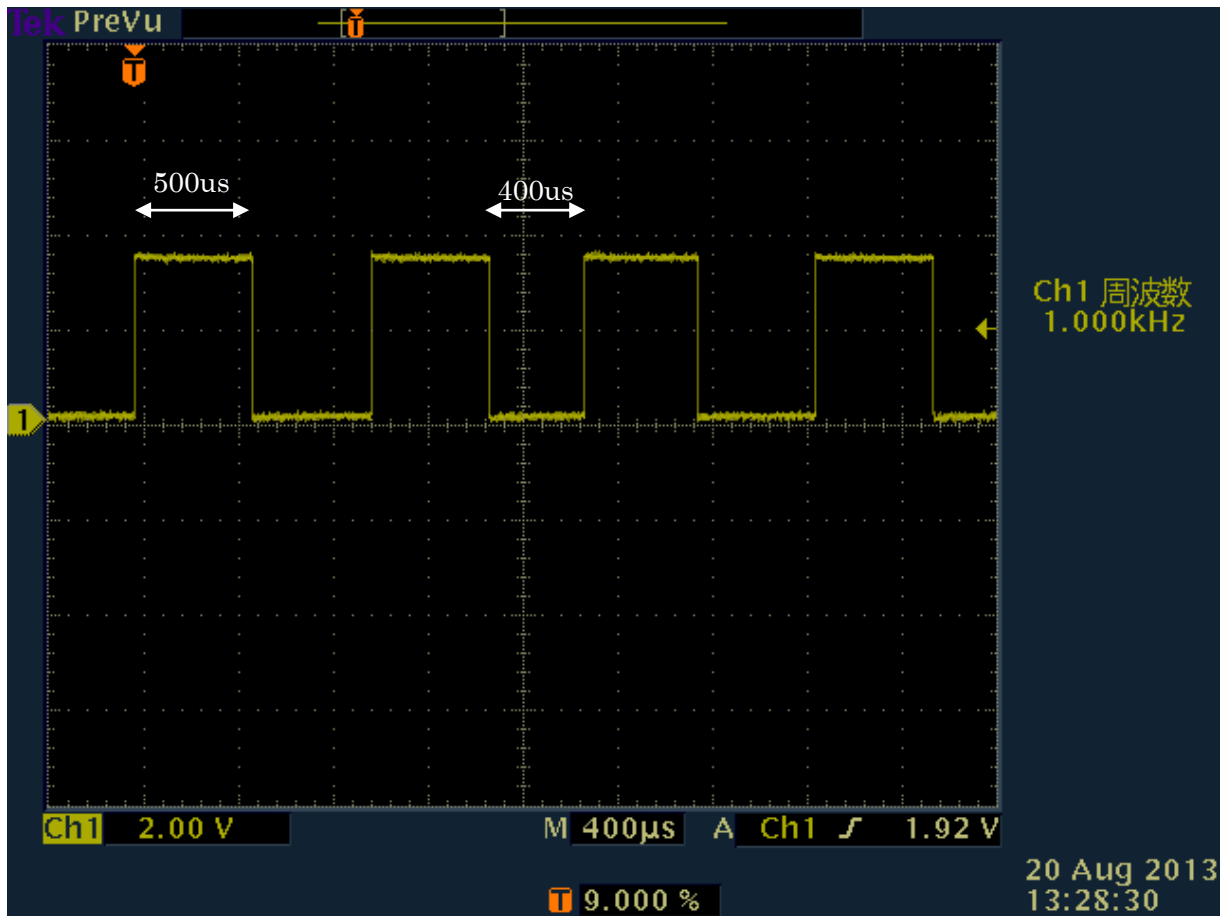


<図7. マクロ機能パネル>

TCL 機能パネルの「ファイル選択」ボタンにより、ユーザがテキスト・エディタで記述した TCL(.tcl)を選択し、「実行」ボタンをクリックして実行します。実行した内容は、「ログ View」画面により確認することができます。TCLコマンドにより、ログ View 画面に表示するマクロコマンド実行結果を非表示にすることもできます。

RefIO7.exe をインストールしたフォルダ内に¥tcl フォルダがあります。このフォルダ内に TCL サンプルスクリプトがありますので、参考にしてください。

TCL スクリプトでレジスタ・ライトを繰り返すと、I/O トグル周波数は約 1KHz になります。すなわち、コマンド発行間隔が約 2KHz (約 500us)になります。尚、この数値は使用する PC 仕様により変化します。



<図8. TCL スクリプトによる IO トグル>

図 6 はレジスタライトコマンドにより、1 と 0 を繰り返し書き込んで波形観測した例です。コマンド発行は約 500us 毎に発行されていることがわかります。しかし、発行間隔は PC,OS のソフトウェア管理によるものなので、常に一定ではありません。

TCL スクリプトで利用できるコマンドの一覧を以下に示します。

コマンドのフォーマットは、PSI_COMMAND “RW -a18 -d55” です。アドレス 18(LED)に 55h を書き込む命令です。

コマンド名	コマンド	機能
レジスタ・ライト	RW	レジスタへの書き込み
レジスタ・リード	RR	レジスタの読み出し
レジスタ・リード・チェック	RRC	読み出したデータを期待値と照合
レジスタ・リード・ポーリング	RRP	読み出したデータを期待値と照合し、一致するまでレジスタ・リード動作を続ける。不一致時のタイムアウト時間の設定も可能。
バースト・レジスタ・ライト	BW	1~64 バイトまでの連続データを、指定したポートに書き出す。CSV ファイルで記述したデータを指定する。
デバイス・セレクト	DS	PC に接続した I/O ボードのアドレスを読み出し、そのボードに対してレジスタ・アクセスができるようにする。
アウトプット・イネーブル設定	PD	レジスタ 0~7 に対応するアウトプット・イネーブル設定。RW コマンドでも代用できます。

<表3. 主なマクロコマンドの一覧>

【備考】

実際に使用した PC 仕様:

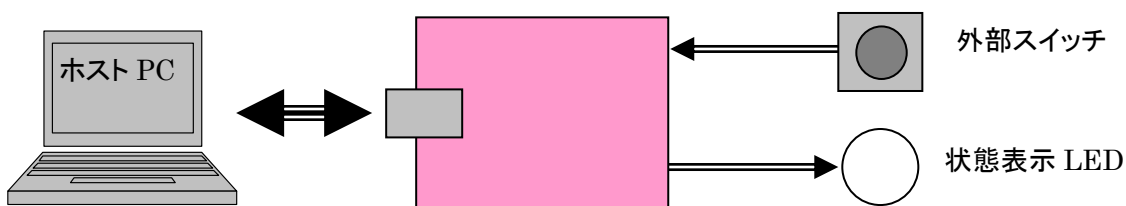
CPU: Intel Core i3 2100 (3.1GHz)

メインメモリ: DDR3 4GB

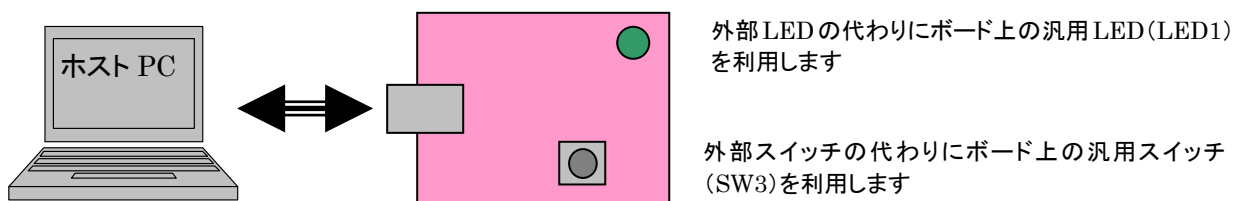
この資料で示す全てのデータは、I/O ボードを USB2.0 ポートに接続し、High Speed モードで運用した場合です。USB2.0 ポートに接続した場合でも、Full Speed モードで運用した場合と、USB1.1 ポートに接続した場合は、I/O トグル周波数として示した数値は大幅に低下します(但し、バーストライト時のトグル周波数は、PC 仕様に大きく依存しません、ハードウェア管理のためです)。

【 IO ボードの使用例】

例として IO ボードにスイッチを接続し、制御アプリケーションからスイッチの状態を監視してスイッチが ON 状態の時に、外部の LED を点灯する方法を示します。



IO ボードに接続した外部スイッチの状態は、PC から定期的にポーリングします。USB 仕様上、スイッチを押したときに、IO ボードが Host PC に割り込み要求することはできません。このため、Host PC は IO ボードに接続しているスイッチの状態を定期的に読み出さなくてはなりません。読み出し間隔は TCL スクリプトで実行する場合、約 500us でスイッチの状態を読み出すことができます。Host PC 側でスイッチが ON 状態になったと認識後、外部 LED に接続している IO ボードのポートを制御して LED を点灯させます。上記構成の場合、Smart-IO2 ボード本体のタクトスイッチと LED により置き換えることができます。ボード上のスイッチと LED を利用することにして、以下の構成に置き換えてみます。

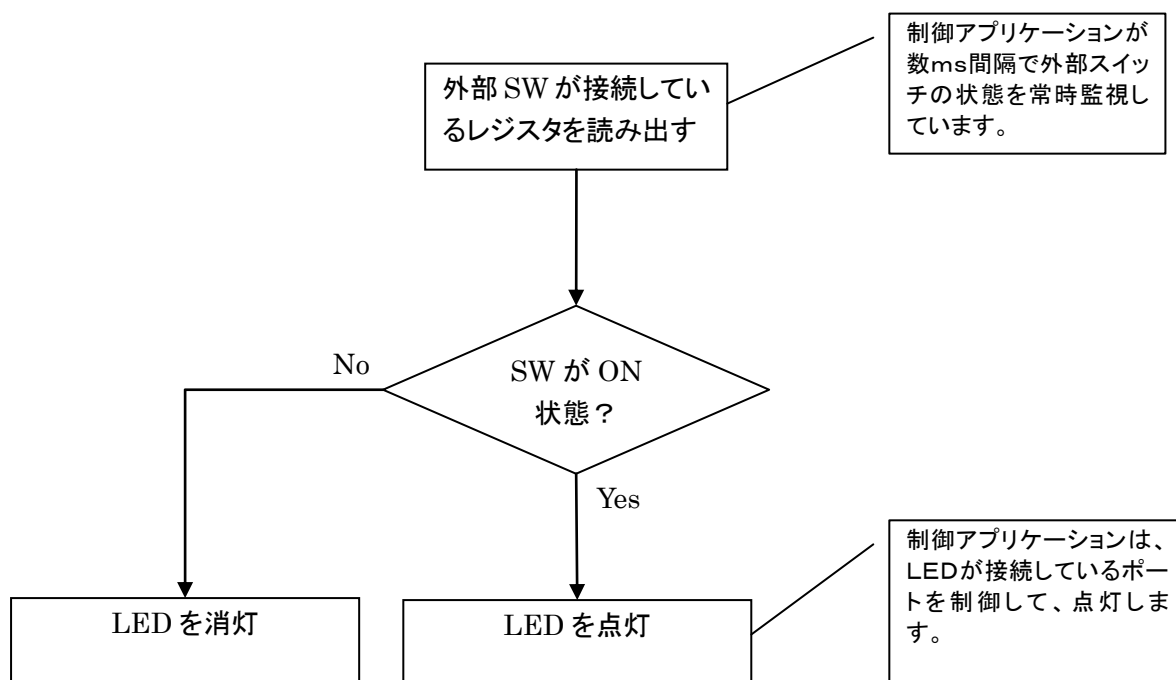


制御ソフトウェアでは以下の様にスイッチが接続しているレジスタの値を一定間隔(500us 前後)で読み出し続け、レジスタの値を元にスイッチが ON されたかどうかを判定します。スイッチが ON していれば、LED が接続しているポート(レジスタ)を制御して LED を点灯します。

TCL スクリプト内では、while 文や for 文を利用することもできます。その中で PSI_COMMAND “RR -a16”を繰り返し実行し、値の変化によりループを抜け、LED を点灯するコマンド PSI_COMMAND “RW -a18 -d01”により LED を点灯させることができます。

TCL スクリプトの例: レジスタ 16 の値の値を読み出し続け、値が変化したら LED を点灯させる。

```
# 以下 3行 予約 変更不可
set result 0
set read 0
set sts 1
  #追加した変数
set polling 0
#TCLバージョンの表示
TPRINT "TCL Version [info tclversion]"
#アドレス 18(LED)をクリア(消灯)
PSI_COMMAND "RW -a18 -d00"
#変数 polling が 0 の間は以下のコマンドを発行し続ける
#アドレス 16 の値を読み出して、read 変数に格納
while { $polling == 0 } {
  PSI_COMMAND "RR -a16"
  TPRINT "Reading tact SW $read"
  #read 変数の値を polling 変数に代入
  set polling $read
}
#polling 変数が 0 以外なら、アドレス 18(LED)の 1ビット目を点灯して終了
PSI_COMMAND "RW -a18 -d01"
```



使用する PC の性能にも依存しますが、スイッチを押してから LED が点灯するまで、1ms 前後の時間がかかります。