

IOシリーズ

USB18FIO (USB—IO)

ユーザーズ・マニュアル

目 次

1.	概要	2
1.1.	デジタル入出力仕様	2
1.2.	AD入力仕様	2
1.3.	USB仕様	2
1.4.	外形寸法	2
1.5.	電源	2
1.6.	動作条件	3
1.7.	対応OS	3
1.8.	添付品	3
1.9.	動作設定	4
1) 1.9.1.	デバイス番号指定	4
2) 1.9.2.	CN12配置図	4
3) 1.9.3.	電源設定方法	4
1.10.	ビット・ポートNO指定方法 (CN12)	5
1.11.	ポート入出力について	6
1.12.	AD変換データ	6
1.13.	デジタル入出力参考回路図	7
1.14.	ネットワーク抵抗の実装	9
2.	インストール	1
2.1.	フォルダ構造	1
2.2.	デバイスドライバインストール	1
2.3.	データロガーソフトウェア	2
3.	プログラム作成	3
3.1.	言語別関数定義ファイル	3
3.1.1.	Visual Basic 6.0	3
3.1.2.	Visual Basic.NET, 2005	3
3.1.3.	Visual C++ (6, 2005)	3
3.1.4.	Visual C# (2005)	3
3.2.	コーディングフロー	4
4.	サンプルプログラム説明	5
4.1.	DIOタイプ用サンプル	5
4.1.1.	構成	5
4.1.2.	前提条件	5
4.1.3.	プログラム作成における実行手順	5
4.1.4.	説明	6
4.2.	ADタイプ用サンプル	7
4.2.1.	構成	7
4.2.2.	前提条件	7
4.2.3.	プログラム作成における実行手順	7
4.2.4.	説明	8
5.	関数一覧	9
5.1.	デバイス操作	9
5.2.	入出力操作	9
5.3.	AD入力	9
6.	関数リファレンス (アルファベット順)	10

1)	Cdusb18_ADGet (AD 入力).....	10
2)	Cdusb18_ADGetArray (AD 入力データ取得)	11
3)	Cdusb18_ADStart (AD サンプリング開始)	11
4)	Cdusb18_ADStatus (AD サンプリング状態確認)	12
5)	Cdusb18_ADStop (AD サンプリング停止)	12
6)	Cdusb18_ADTime (AD 変換時間設定)	13
7)	Cdusb18_Close (デバイスクローズ)	14
8)	Cdusb18_CloseAll (デバイスクローズALL)	14
9)	Cdusb18_InBit (ポートから1点読み込み)	15
10)	Cdusb18_InPort (ポートから1ポート読み込み)	15
11)	Cdusb18_InDport (ポートからダブルポート読み込み)	16
12)	Cdusb18_InFport (ポートから4ポート読み込み)	16
13)	Cdusb18_InAport (ポート全て読み込み)	17
14)	Cdusb18_Open (デバイスオープン)	17
15)	Cdusb18_OutBit (ポートへ1点出力)	18
16)	Cdusb18_OutPort (ポートへ1ポート出力)	18
17)	Cdusb18_OutDport (ポートへダブルポート出力)	19
18)	Cdusb18_OutFport (ポートへ4ポート出力)	19
19)	Cdusb18_OutAport (全ポート出力)	20
20)	Cdusb18_SelDevice (デバイス選択)	20
21)	Cdusb18_SetDir (デバイス出力方向設定) DIO タイプのみ有効.....	21

1. 概要

WindowsパソコンのUSBポートと接続することで、デジタル入出力/AD変換入力が容易にできます。IO24点をポート（4点ずつ）毎に入出力方向設定ができるので（DIOタイプのみ）、制御対象に合わせて自由に利用いただけます。

ソフトウェア開発もアクセス用ライブラリ（DLL）が標準添付されていますので、Visual Basic、Visual C++、Visual C#などでプログラミングが容易にできます。

また、DSWを設定し同時に8台まで使用できます。

1.1. デジタル入出力仕様

- ・ TTL <DIOタイプ> 24点
<ADタイプ> 入力8点・出力8点
(入出力はソフトウェアにて設定可能<DIOタイプのみ>ポート毎（4点ずつ）)

電圧 (V_{dd} 5V時) :

入力Lo電圧 0.8V最大、入力Hi電圧 2.0V最小

出力Lo電圧 0.6V最大、出力Hi電圧 V_{dd}-0.7V最小

- ・ 回路 入力：非絶縁TTLレベル（正論理）
出力：非絶縁TTLレベル（正論理）
1ピンあたり シンク/ソース電流 25mA以下
全体 180mA以下

1.2. AD入力仕様

- ・ チャンネル 4ch
- ・ 分解能 10bit
- ・ 変換時間 MAX 32 μ Sec (C_{usb18_ADTime} 関数にて4段階に変換時間を選択可)
- ・ 入力方法 シングルエンド入力
- ・ 入力電圧 0～5V
- ・ 基準電圧 USBバスパワー時 USBポートからの電圧
セルフパワー時 セルフパワー電源電圧（外部電源）

1.3. USB仕様

- ・ USB 2.0 High-Speed (480Mbps) モード対応

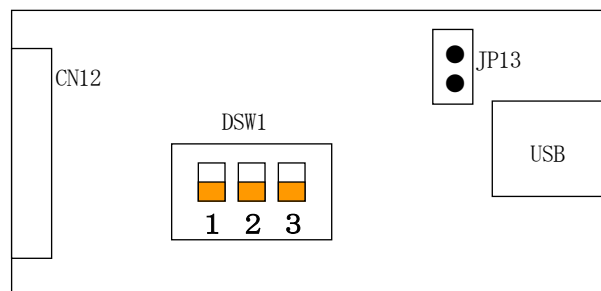
1.4. 外形寸法

- ・ 75×50×30 (mm)、70g

1.5. 電源

- ・ バスパワー／セルフパワー駆動切替可能
(ジャンパーにて設定可 詳細は“1.9. 電源設定方法”を参照してください)
セルフパワー駆動時 外部入力電圧 DC 5V
- ・ 消費電流 80mA（本体動作電流）

1.9. 動作設定

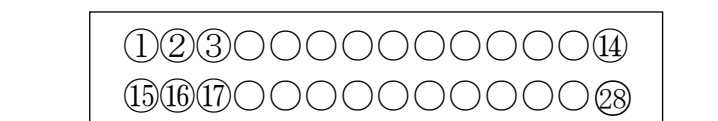


1) デバイス番号指定

DSW1にてデバイス番号設定となります。設定方法は下表を参照してください。

デバイス番号	DSW1-1	DSW1-2	DSW1-3
0 (出荷時設定)	OFF	OFF	OFF
1	ON	OFF	OFF
2	OFF	ON	OFF
3	ON	ON	OFF
4	OFF	OFF	ON
5	ON	OFF	ON
6	OFF	ON	ON
7	ON	ON	ON

2) CN12配置図



CN12を挿入方向からみた配置

3) 電源設定方法

JP13にて、バスパワー／セルフパワー（外部電源）の設定を行います。

バスパワー : ショート（出荷時設定）

セルフパワー : オープン

※電源設定によりCN12-1・CN12-2 (+5V) / CN12-27・CN12-28 (GND) の動作が以下の様に異なります。電源設定と異なる接続をした場合デバイス等が破損する恐れがありますのでご注意ください。

バスパワー設定時 : USBバス電源 (+5V) がデバイスで使用されます。
CN12-1・CN12-2 +5V出力
CN12-27・CN12-28 出力用GND

セルフパワー設定時 : 外部接続電源 (+5V) がデバイスで使用されます。
CN12-1・CN12-2 +5V入力
CN12-27・CN12-28 入力用GND

1. 概要

1.10. ビット・ポートNO指定方法 (CN12)

+5V (ピンNo1・2)、GND (ピンNo27・28) は電源設定にて入出力が異なります。
 詳細は“1.9. 電源設定方法”を参照してください。

<DIOタイプ>

		+5V																									GND	
		1	2																								27	28
ピンNo	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26				
*1ビット No	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I				
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23				
ポート No	IO_P0			IO_P1				IO_P2				IO_P3				IO_P4				IO_P5								
ダブル ポート No	IO_DP0								IO_DP1								IO_DP2											
4ポート	←												→															

※1: ビットNoは、IO_Bx となります
 ※GND(ピンNo27, 28)は入力/出力共通です

<ADタイプ>

		+5V																									GND	
		1	2																								27	28
ピンNo	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26				
*1ビット No	A	A	A	A	*2	*2	*2	*2	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I				
	D	D	D	D	N	N	N	N																				
	0	1	2	3	C	C	C	C	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B				
	0	1	2	3	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26					
ポート No									IO_P0				IO_P1				IO_P0				IO_P1							
ダブル ポート No									IO_DP0								IO_DP0											
	AD				デジタル出力								デジタル入力															

※1: ビットNoは、IO_Bx となります
 ※2: NC (ピンNo7~10) には接続しないでください。
 ※GND(ピンNo27, 28)は入力/出力共通です

1.1.1. ポート入出力について

ポート入出力時のデータ指定方法は下記例を参考にしてください。

ピンNo	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
状態	0 N	0 N	0 F F	0 F F	0 N	0 F F	0 F F	0 F F	0 F F	0 F F	0 N	0 F F	0 F F	0 F F	0 F F	0 F F	0 F F	0 F F	0 F F	0 N	0 N	0 N	0 N	0 N
データ	0x3			0x1			0x4				0x0				0x8				0xF					
	①																							
	②																							
	③																							
	④																							

① ポート0入力時・・・0x3

② ダブルポート0入力時・・・0x13

③ 4ポート入力時 (D I Oタイプのみ)・・・0x0413

④ 全ポート入力時 (D I Oタイプのみ)・・・0xF80413

1.1.2. AD変換データ

AD変換データは10ビット (000h～3FFh) です。未使用ビットは‘0’です。したがって、基準電圧が0～5Vの時は、約4.8828mV (5V÷2¹⁰(1024)) 単位で計測できます。基準電圧については1.2. AD入力仕様を参照してください。

【データ精度について】

- ◎基準電圧差が3.0V以上ない場合は10ビット分解能の精度は保証されません。
- ◎アナログ入力源の出力インピーダンスが2.5kΩより高くなると精度が悪化します。
- ◎基準電圧源の出力インピーダンスが250Ωより高くなると精度が悪化します。

【電圧データ値】

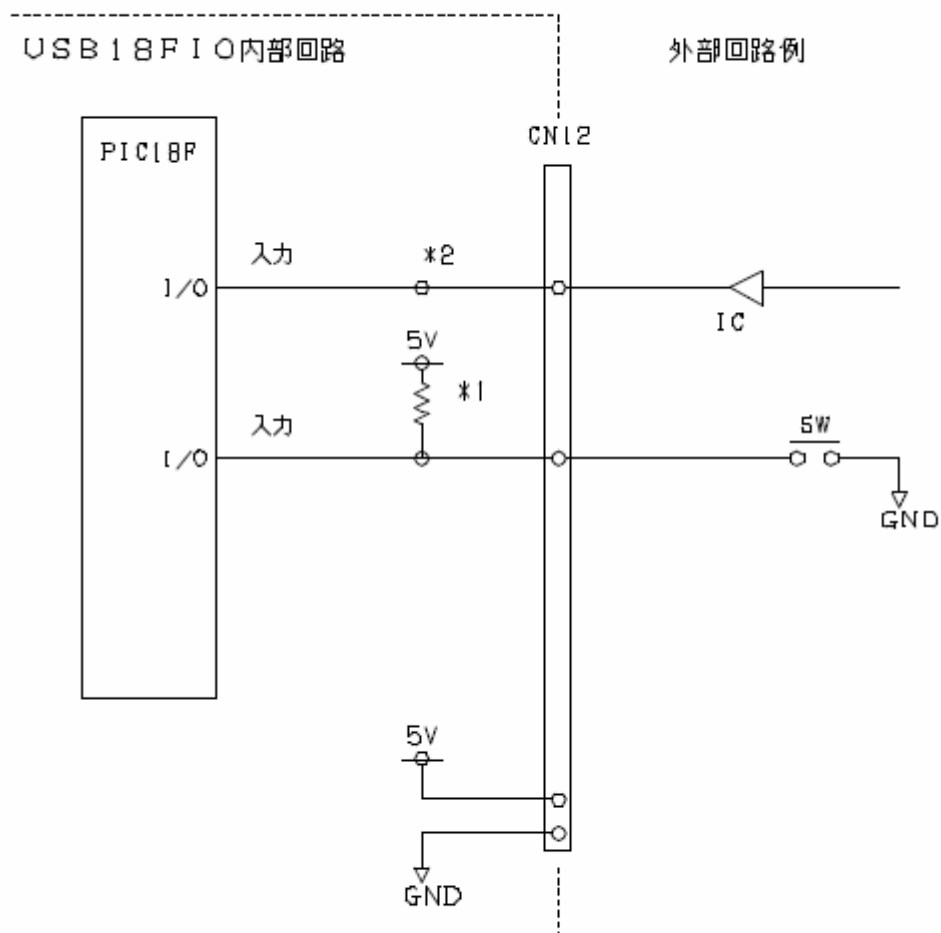
◎入力関数(Cdusb18_ADGet など)にて取得したデータは以下の式にて電圧値に変換できます。

$$\text{電圧値 (V)} = \text{取得したデータ (10bit)} * (5 / 1023)$$

1. 概要

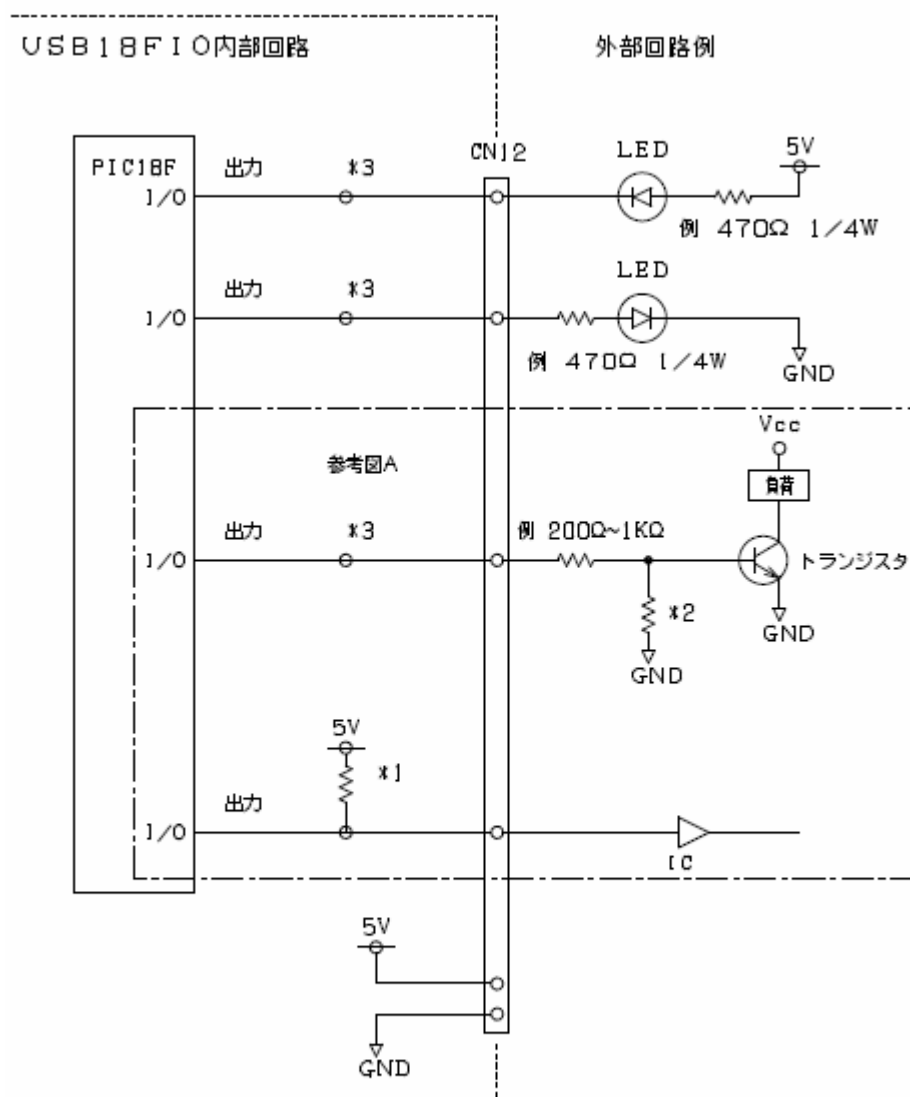
1.1.3. デジタル入出力参考回路図

デジタル入力参照回路図



- *1 : 入力がオープン状態にならないためのプルアップ抵抗
- *1 : 付属品の抵抗を取り付けた時 M5-1-103J (BIテクノロジー)
- *2 : 購入時はプルアップ抵抗は付いていません

デジタル出力参照回路図



- *1：出力がハイインピーダンス状態にさせないためのプルアップ抵抗
- *1：付属品の抵抗を取り付けた時 M5-1-103J (BIテクノロジー)
- *2：出力がハイインピーダンス状態にさせないためのプルダウン抵抗(例 10KΩ)
- *3：購入時はプルアップ抵抗は付いていません

注意 デジタル入出力ポートは電源投入またはリセット直後はすべて入力モードとなります。従って、出力ピンとして使う場合には、リセットした直後出力がハイインピーダンス状態となるので、接続相手がハイインピーダンス状態で不安定動作しないように対策しておく必要があります。

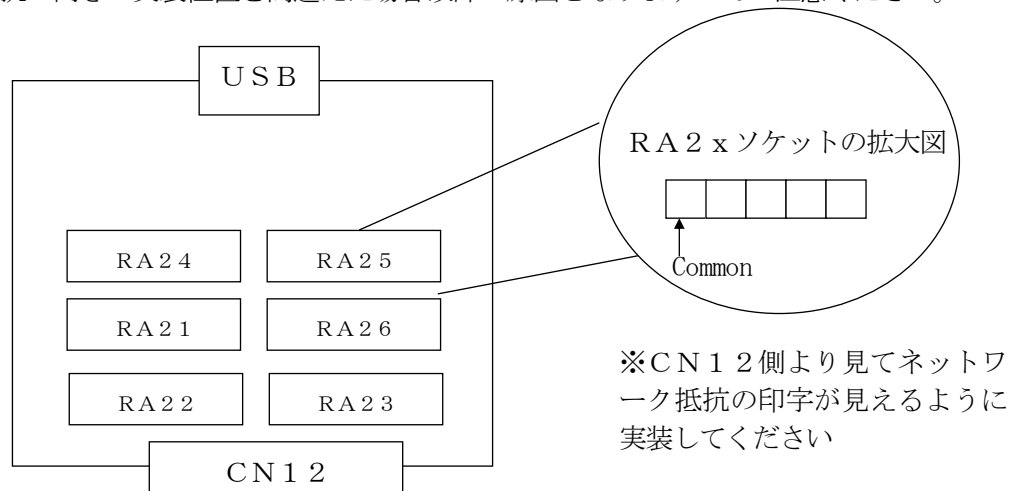
この対策としては、トランジスタ、ICの負荷のような時には、参考図Aのように、プルダウンかプルアップする必要があります。これによりハイインピーダンス状態を無くし誤動作を避けることができます。

プルアップが必要な時は、付属のネットワーク抵抗をボード内に取り付けられ、外部に取り付ける必要がなくなります。(プルダウンの時は外部に取り付けて下さい。)

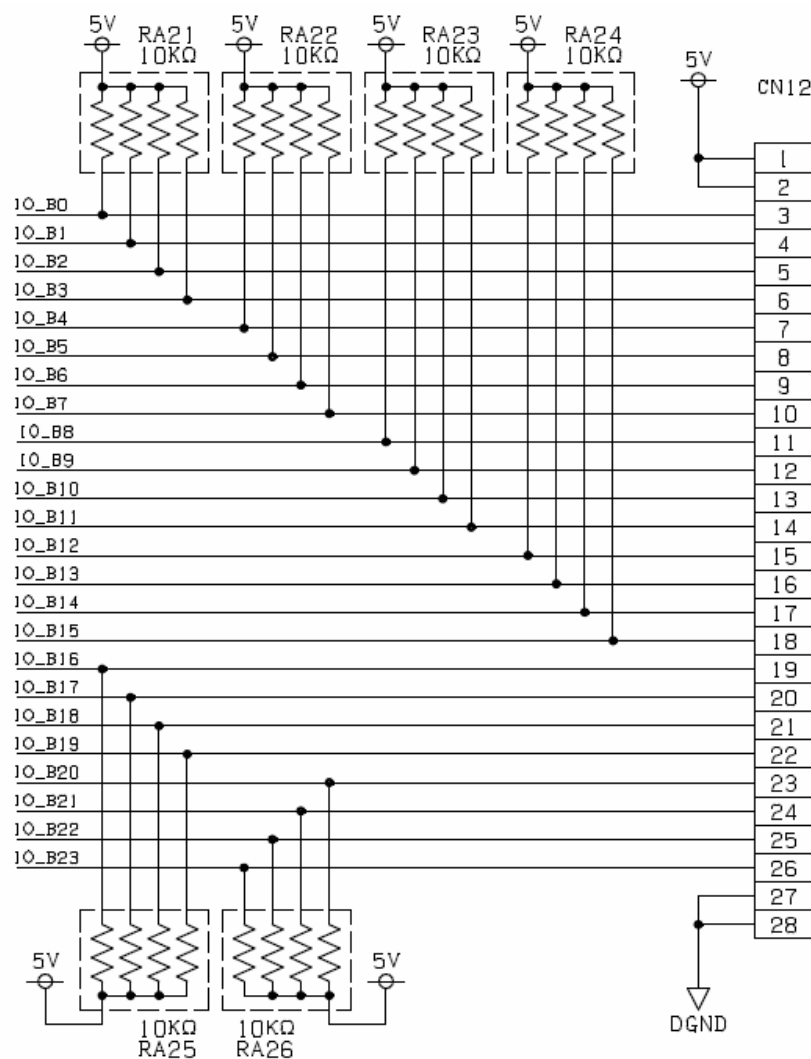
1. 概要

1.1.4. ネットワーク抵抗の実装

本製品はボックス内の基板にプルアップ抵抗(10KΩ)を実装できます。ボックスの蓋を開けて、ソケットにネットワーク抵抗を挿して下さい。1個のネットワーク抵抗で4ビットをプルアップできます。下図は、USB 18F I/Oを上部より見た場合のネットワーク抵抗実装ソケットの配置図です。付属のネットワーク抵抗を実装する場合は、以下の回路図を参考にしてください。ネットワーク抵抗の向き・実装位置を間違えた場合故障の原因となりますのでご注意ください。



【回路図】



2. インストール

2.1. フォルダ構造

インストールを実行すると以下のフォルダ、ファイルがコピーされます。

インストールフォルダ (Default:c:\¥CHUBU ELECTRIC¥USB18FIO)

USB18FIO

DOC	ユーザーズマニュアル
driver	USBデバイスドライバ
i386	32bit OS用USBデバイスドライバ
amd64	64bit OS用USBデバイスドライバ
DtLog	データロガープログラム
INC	VC用ヘッダファイル (cdusb18.h)、 VB6用標準モジュール (cdusb18.bas)、 VB.NET用標準モジュール (cdusb18.vb) VC#用定義ファイル (cdusb18.cs)
LIB	アクセス用 (DLL) ファイル、LIBファイル
x32	32bit アプリケーション用
x64	64bit アプリケーション用

Sample

【DIOタイプ用】

VB	VB6 サンプル
VBNET	VB.NET (2005) サンプル
VC	VC6 サンプル
VC2005	VC2005 サンプル
VCS2005	VC#2005 サンプル

【ADタイプ用】

VBNET_AD	VB.NET (2005) サンプル
VC2005_AD	VC2005 サンプル
VCS2005_AD	VC#2005 サンプル

2.2. デバイスドライバインストール

デバイスドライバは、添付CDソフトと同時にインストールできます。

同時にインストールしない場合は、以下の2つの方法いずれかでインストールしてください。

① 接続前にインストール (推奨)

PCへUSB18FIO-DIO/ADをUSB接続する前に、“インストールフォルダ¥Driver”内 setup.exe を実行 (ダブルクリック) してダイアログに従ってインストールします。

(OS 32/64bit は自動で判定しインストールされます)

その後、USB18FIO-DIO/ADをPCへ接続しプラグアンドプレイにて“ソフトウェアを自動的にインストールする”でインストールします。

② プラグアンドプレイ機能

PCへUSB18FIO-DIO/ADをUSB接続すると、デバイスインストールのダイアログが表示されます。ダイアログに従って“インストールフォルダ¥Driver”を指定してインストールします。(OS 32/64bitによりドライバが異なりますので、OSの種類を確認の上選択してください。)

2.3. データロガーソフトウェア

データロガーソフトウェアは、インストール後セットアップを実行してご使用ください。セットアップを実行すると、実行ファイルとヘルプファイルがコピーされます。また、プログラムメニューへ“データロガー”が追加されます。

ご使用される際は、必ずヘルプを参照の上最初に「設定」をしてください。

【インストールについて】

インストールフォルダ¥Dtlog¥ DtlogSetup.msi を実行（ダブルクリック）してインストールしてください。

追加されるファイル・・・USB18F-DtLog.exe、USB18F-DtLog.chm

【注意】

データロガーソフトのデータ収集において実行しても収集ファイルが作成されない場合は、指定したフォルダへの書き込み権限がある事を確認してください。Ver1.1.0以前のバージョンは保存ファイル名のデフォルト値がインストールフォルダになっているため Program Files に設定されている場合があります。Windows Vista以降ではこのフォルダへの書き込み権限が制約されていますので、確認してください。なお、Ver1.2.0以降はマイドキュメントがデフォルトフォルダになっています。

3. プログラム作成

3. 1. 言語別関数定義ファイル

プログラムを作成する時は、以下の準備をしてください。

※説明の標記について

INST・・・インストールフォルダ名 に置き換えてください。

※dll ファイルは、作成するアプリケーションのタイプ(32bit/64bit)により異なります。
64bit OS へインストールした場合、32bit 用/64bit 用共それぞれのシステム
フォルダへインストールされています。

3. 1. 1. Visual Basic 6.0

Visual Basic 6.0 に必要なファイルをお客様のフォルダへコピーしてプロジェクトへ追加しご使用
ください。

なお、cdusb18.dll を使用していますがこのファイルはインストール時に Windows のシステムフォル
ダへ既にコピーされていますので、特にコピーの必要はありません。

ファイル名	コピー元	コピー先
cdusb18.bas	INST¥INC	お客様のプロジェクトフォルダ

3. 1. 2. Visual Basic.NET, 2005

Visual Basic.NET, 2005 に必要なファイルをお客様のフォルダへコピーしてプロジェクトへ追加しご
使用ください。

なお、cdusb18.dll を使用していますがこのファイルはインストール時に Windows のシステムフォル
ダへ既にコピーされていますので、特にコピーの必要はありません。

ファイル名	コピー元	コピー先
Cdusb18.vb	INST¥INC	お客様のプロジェクトフォルダ

3. 1. 3. Visual C++ (6, 2005)

Visual C++に必要なファイルをお客様のフォルダへコピーしてプロジェクトへ追加しご使用くださ
い。

なお、cdusb18.dll を使用していますがこのファイルはインストール時に Windows のシステムフォル
ダへ既にコピーされていますので、特にコピーの必要はありません。

ファイル名	コピー元	コピー先
cdusb18.h	INST¥INC	お客様のプロジェクトフォルダ
cdusb18.lib	INST¥LIB¥x32 Or INST¥LIB¥x64	お客様のプロジェクトフォルダ

3. 1. 4. Visual C# (2005)

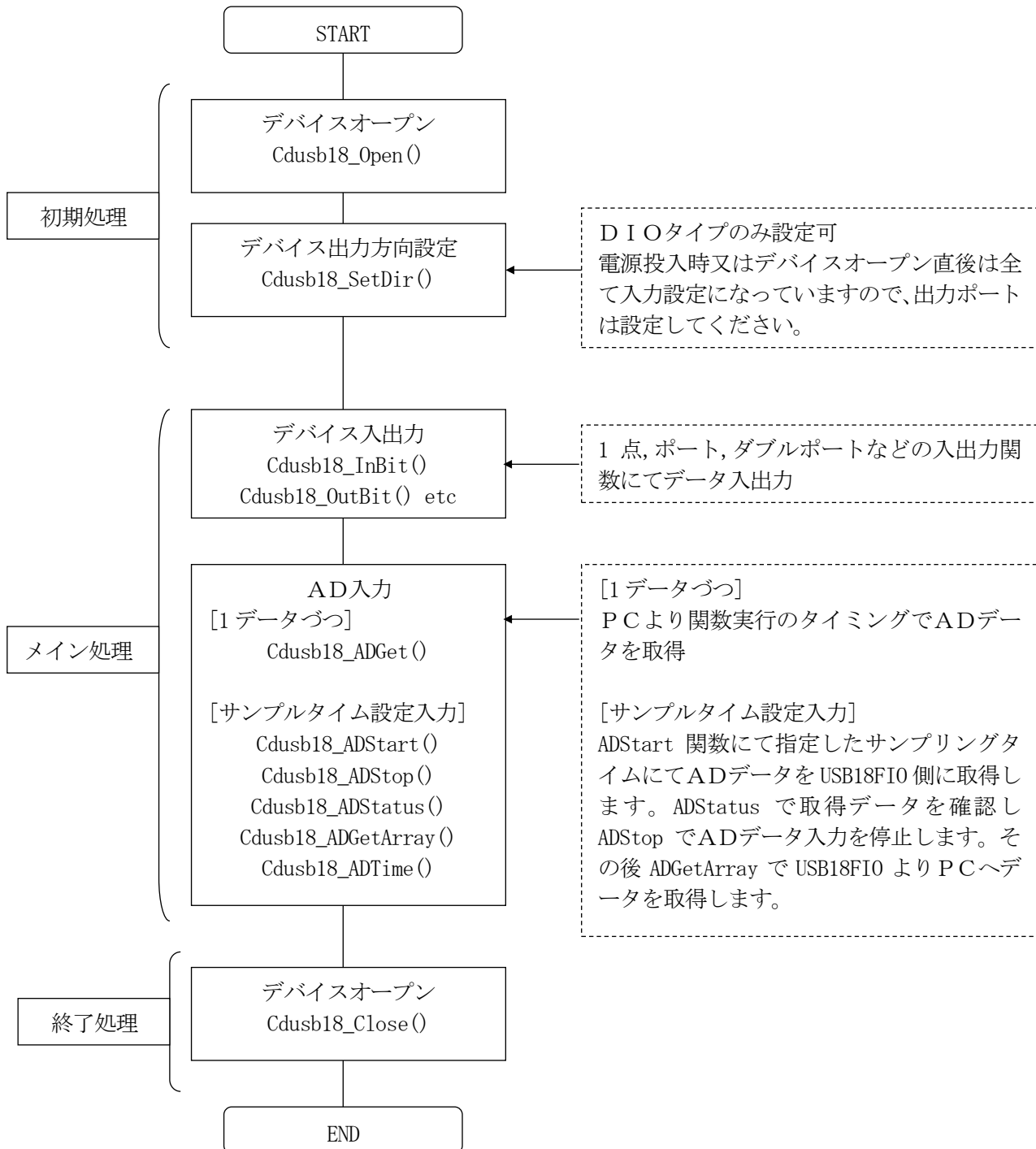
Visual C#に必要なファイルをお客様のフォルダへコピーしてプロジェクトへ追加しご使用くださ
い。

なお、cdusb18.dll を使用していますがこのファイルはインストール時に Windows のシステムフォル
ダへ既にコピーされていますので、特にコピーの必要はありません。

ファイル名	コピー元	コピー先
cdusb18.cs	INST¥INC	お客様のプロジェクトフォルダ

3.2. コーディングフロー

プログラム作成例です。各関数の使用方法については、“5. 関数一覧”を参照してください。デバイスオープン後デバイスクローズまでに、USB18FI0 本体がPCより切り離された場合PCが不安定な状態になる場合があります。デバイスオープン後はデバイスクローズをしてから切り離してください。



4. サンプルプログラム説明

サンプルプログラムは、USB 18F IOをパソコンから操作するためのサンプルソフトウェアです。サンプルプログラムは、各開発環境がインストールされているPCにて動作可能です。

4.1. DIOタイプ用サンプル

4.1.1. 構成

Microsoft社製 Visual Basic6(VB)、Visual Studio.NET2002/2005 Basic(VBNET)、Visual C++6(VC)、VisualC++2005(VC2005)、VisualC#2005(VCS2005)にて作成しています。

プログラムは()内のフォルダ名にてインストールされています。

4.1.2. 前提条件

このサンプルプログラムは、DIOタイプ用です。ポート2・3を出力/4・5を入力ポートとして作成されています。この設定はデバイスに合わせて必ず変更してご使用ください。

DIOサンプルプログラムをADタイプにて使用される場合は、以下の修正をしてご使用ください。
<VB>

入出力方向設定削除

SCdusb18_SetDir(2)、SCdusb18_SetDir(3)の命令を削除

ポート番号変更

SCdusb18_OutDport(1, data)→SCdusb18_OutDport(0, data)

SCdusb18_InDport(2, data) →SCdusb18_InDport(0, data)

<VC>

入出力方向設定削除

_Cdusb18_SetDir(2)、_Cdusb18_SetDir(3)の命令を削除

ポート番号変更

_Cdusb18_OutDport(IO_DP1, Data)→_Cdusb18_OutDport(IO_DP0, Data)

_Cdusb18_InDport(IO_DP2, &buf) →_Cdusb18_InDport(IO_DP0, &buf)

4.1.3. プログラム作成における実行手順

① デバイスオープン (Cdusb18_Open)

オープン直後は該当デバイスが選択された状態です。複数デバイス同時使用時はデバイス選択(Cdusb18_SelDevice)してご使用ください。

デバイス選択された状態の時はデバイス番号を指定しない関数にて操作できます。
(ex. _Cdusb18_InBit/SCdusb18_InBit など)

デバイス選択せずに使用する場合は、デバイス番号を指定する関数にて操作してください。
(ex. Cdusb18_InBit など)

② デジタル入出力方向設定(Cdusb18_SetDir)

電源投入(USBポート接続)時は全て入力モードになっています。出力使用ポートのみ設定してください。

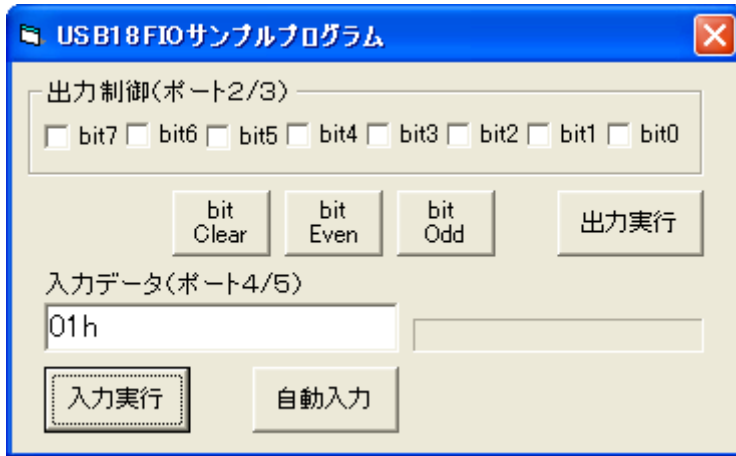
③ 入出力実行(Cdusb18_InBit, Cdusb18_OutBit など)

必要に応じて各入出力関数実行

④ デバイスクローズ(Cdusb18_Close, Cdusb18_CloseAll)

プログラムの最後に必ず①でオープンしたデバイスをクローズしてください。なおCdusb18_CloseAllを使用すると現在オープンしているデバイスを検索し全てクローズします。

4.1.4. 説明



① 出力実行

出力8点をbit0～bit7としチェックした点へデータ“1”を「出力実行」ボタンを押下して出力します。

「bit Clear」 「bit Even」 「bit Odd」 ボタンはチェックする際に使用します。

②入力実行

「入力実行」を押下したタイミングで入力関数 `Cdusb18_InDport()` を実行しデータを表示します。

③自動入力

タイマーにて 1mSec 毎に入力関数 `Cdusb18_InDport()` を実行し、データを表示します。
一部この機能に実装していないサンプルプログラムがあります。

4.2. ADタイプ用サンプル

4.2.1. 構成

Microsoft 社製 Visual Studio.NET2002/2005 Basic(VBNET_AD)、VisualC++2005(VC2005_AD)、VisualC#2005(VCS2005_AD)にて作成しています。

プログラムは()内のフォルダ名にてインストールされています。

4.2.2. 前提条件

このサンプルプログラムは、ADタイプ専用です。

4.2.3. プログラム作成における実行手順

① デバイスオープン (Cdusb18_Open)

オープン直後は該当デバイスが選択された状態です。複数デバイス同時使用時はデバイス選択 (Cdusb18_SelectDevice) してご使用ください。

デバイス選択された状態の時はデバイス番号を指定しない関数にて操作できます。
(ex. _Cdusb18_InBit/SCdusb18_InBit など)

デバイス選択せずに使用する場合は、デバイス番号を指定する関数にて操作してください。
(ex. Cdusb18_InBit など)

② 入出力実行(Cdusb18_InBit, Cdusb18_OutBit, Cdusb18_ADGet など)

必要に応じて各入出力関数実行

③ デバイスクローズ(Cdusb18_Close, Cdusb18_CloseAll)

プログラムの最後に必ず①でオープンしたデバイスをクローズしてください。なお Cdusb18_CloseAll を使用すると現在オープンしているデバイスを検索し全てクローズします。

4.2.4. 説明



①AD入力

AD入力に `Cdusb18_ADGet()` 関数を使用し任意のタイミングで取得し、電圧値へデータを変換して表示しています。チャンネル番号はコンボボックスより選択できます。

②D I O出力

出力8点を bit0～bit7 としチェックした点へデータ“1”を「出力実行」ボタンを押下して出力します。

「bit Clear」 「bit Even」 「bit Odd」 ボタンはチェックする際に使用します。

入力は、「入力実行」押下したタイミング又は「自動入力」タイマーにて 1mSec 毎に実行します。取得した結果は1点ずつの内容は入力欄へ”1”は青色、”0”は白色で表示され、8点を1ポートとしての内容は”InDport”欄へ表示されます。入力には `Cdusb18_InPort()` , `Cdusb18_InBit()` ,

`Cdusb18_InDport()` 関数を使用しています。

5. 関数一覧

5. 1. デバイス操作

Cdusb18_Open (デバイスオープン)	17
Cdusb18_Close (デバイスクローズ)	14
Cdusb18_CloseAll (デバイスクローズALL).....	14
Cdusb18_SelDevice (デバイス選択)	20

5. 2. 入出力操作

Cdusb18_InBit (ポートから1点読み込み)	14
Cdusb18_InPort (ポートから1ポート読み込み)	15
Cdusb18_InDport (ポートからダブルポート読み込み)	16
Cdusb18_InFport (ポートから4ポート読み込み)	16
Cdusb18_OutBit (ポートへ1点出力)	18
Cdusb18_OutPort (ポートへ1ポート出力)	18
Cdusb18_OutDport (ポートへダブルポート出力)	19
Cdusb18_InAport (ポート全て読み込み)	17
Cdusb18_OutFport (ポートへ4ポート出力)	19
Cdusb18_OutAport (全ポート出力)	20

5. 3. AD入力

Cdusb18_ADStart (AD サンプリング開始)	11
Cdusb18_ADStop (AD サンプリング停止)	12
Cdusb18_ADStatus (AD サンプリング状態確認)	11
Cdusb18_ADGetArray (AD 入力データ取得)	11

6. 関数リファレンス (アルファベット順)

この関数リファレンスは、Microsoft 社製 VisualStudio VC++/VB での使用方法について説明しています。VB については、Microsoft 社製 VisualStudio のバージョンによりデータ型の長さがことなりますので、注意してください。

VC++の場合、関数名称の先頭に ‘_’ が有る関数は現在のボード (Cdbus18_SelDevice で選択したボード) を対象に実行される関数です。

VB の場合、関数名称の先頭に ‘S’ が有る関数は現在のボード (Cdbus18_SelDevice で選択したボード) を対象に実行される関数です。

※UINT…unsigned int

1) Cdbus18_ADGet (AD 入力)

【機能】 AD変換し、データを取得します。

【引数】 devno デバイス番号
 ch チャンネル番号 (0～3)
 *buf データ受信バッファ (1 データ分)

【書式】 VC :

```
BOOL Cdbus18_ADGet(int devno, UINT ch, UINT *buf)
BOOL _Cdbus18_ADGet(UINT ch, UINT *buf)
```

VB (Ver6.0) :

```
Function Cdbus18_ADGet(ByVal devno As Long, ByVal ch As Long,
                      ByRef buf As Long) As Long
Function SCdbus18_ADGet(ByVal ch As Long, ByRef buf As Long) As Long
```

VB (.NET 以降) :

```
Function Cdbus18_ADGet(ByVal devno As Integer, ByVal ch As Integer,
                      ByRef buf As Integer) As Integer
Function SCdbus18_ADGet(ByVal ch As Integer,
                      ByRef buf As Integer) As Integer
```

【戻値】 0 異常終了
 0 以外 正常終了

【参考】 Cdbus18_ADTime

2) Cdusb18_ADGetArray (AD 入力データ取得)

【機能】 Cdusb18_ADStart/Cdusb18_ADStop 関数にて AD 入力したデータを指定長受信します。

【引数】 devno デバイス番号
 *buf データ受信バッファ
 size データ長 (1～300)

【書式】 VC :

```
BOOL Cdusb18_ADGetArray(int devno, UINT *buf, UINT size)
BOOL _Cdusb18_ADGetArray(UINT *buf, UINT size)
```

VB (Ver6.0) :

```
Function Cdusb18_ADGetArray(ByVal devno As Long, ByRef buf As Long,
                             ByVal size As Long) As Long
Function SCdusb18_ADGetArray (ByRef buf As Long, ByVal size As Long) As Long
```

VB (.NET 以降) :

```
Function Cdusb18_ADGetArray(ByVal devno As Integer, ByRef buf As Integer,
                             ByVal size As Integer) As Integer
Function SCdusb18_ADGetArray(ByRef buf As Integer,
                             ByVal size As Integer) As Integer
```

【戻値】 0 異常終了、0 以外 正常終了

3) Cdusb18_ADStart (AD サンプルング開始)

【機能】 指定したサンプルング時間で連続してAD変換をし最長300データまで保持します。変換データは Cdusb8_GetArray 関数にて受信できます。(同時に複数チャンネルのサンプルング開始はできません)

【引数】 devno デバイス番号
 ch チャンネル番号 (0～3)
 tm サンプルング時間 (μSec 単位 ～1,300,000)
 最小値は Cdusb18_ADTime 設定値に依存します

【書式】 VC :

```
BOOL Cdusb18_ADStart(int devno, UINT ch, UINT tm)
BOOL _Cdusb18_ADStart(UINT ch, UINT tm)
```

VB (Ver6.0) :

```
Function Cdusb18_ADStart(ByVal devno As Long, ByVal ch As Long,
                          ByVal tm As Long) As Long
Function SCdusb18_ADStart(ByVal ch As Long, ByVal tm As Long) As Long
```

VB (.NET 以降) :

```
Function Cdusb18_ADStart(ByVal devno As Integer, ByVal ch As Integer,
                          ByVal tm As Integer) As Integer
Function SCdusb18_ADStart(ByVal ch As Integer, ByVal tm As Integer) As Integer
```

【戻値】 0 異常終了、0 以外 正常終了

4) Cdusb18_ADStatus (AD サンプルング状態確認)

【機能】 Cdusb18_ADStart で開始したAD変換のバッファ状態を確認します。現在のサンプルングデータ数 (0～300) を取得できます。

【引数】 devno デバイス番号

【書式】 VC :

```
UINT   Cdusb18_ADStatus(int devno)
UINT   _Cdusb18_ADStatus(void)
```

VB (Ver6.0) :

```
Function Cdusb18_ADStatus (ByVal devno As Long) As Long
Function SCdusb18_ADStatus() As Long
```

VB (.NET 以降) :

```
Function Cdusb18_ADStatus(ByVal devno As Integer) As Integer
Function SCdusb18_ADStatus() As Integer
```

【戻値】 データ数 (0～300)

5) Cdusb18_ADStop (AD サンプルング停止)

【機能】 Cdusb18_ADStart で開始したAD変換を停止します。

【引数】 なし

【書式】 VC :

```
BOOL   Cdusb18_ADStop(void)
BOOL   _Cdusb18_ADStop(void)
```

VB (Ver6.0) :

```
Function Cdusb18_ADStop() As Long
Function SCdusb18_ADStop() As Long
```

VB (.NET 以降) :

```
Function Cdusb18_ADStop() As Integer
Function SCdusb18_ADStop() As Integer
```

【戻値】 0 異常終了、0 以外 正常終了

6) Cdusb18_ADTime (AD 変換時間設定)

【機能】 Cdusb18_ADStart・Cdusb18_ADGet でAD変換をする際の変換時間を4段階で設定します。デフォルト値は3です。

【引数】 devno デバイス番号
idx 変換時間設定 (0～3)

【書式】 VC :

```
BOOL Cdusb18_ADTime(int devno, UINT idx)
BOOL _Cdusb18_ADTime(UINT idx)
```

VB (Ver6.0) :

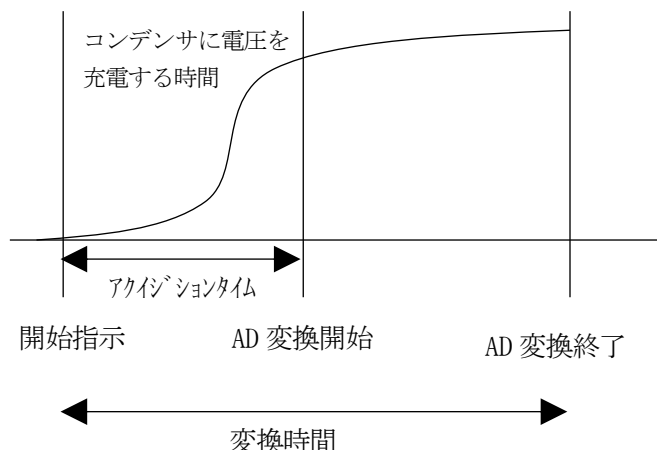
```
Function Cdusb18_ADTime(ByVal devno As Long, ByVal idx As Long) As Long
Function SCdusb18_ADTime(ByVal idx As Long) As Long
```

VB (.NET 以降) :

```
Function Cdusb18_ADTime(ByVal devno As Integer, ByVal idx As Integer) As Integer
Function SCdusb18_ADTime(ByVal idx As Integer) As Integer
```

【戻値】 0 異常終了、0以外 正常終了

=AD変換に要する時間=



設定値	時間 (μ Sec)
0	31
1	37
2	44
3	50

7) Cusb18_Close (デバイスクローズ)

【機能】 指定したUSBポート等を解放します。ユーザープログラム終了直前にこの関数を実行してください。デバイスクローズするとI/Oの入出力設定がデフォルト値(入力)になります。

【引数】 devno デバイス番号

【書式】 VC : void Cusb18_Close(int devno)
 VB (Ver6.0) : Sub Cusb18_Close(ByVal devno as Long)
 VB (.NET 以降) : Sub Cusb18_Close(ByVal devno as Integer)

【戻値】 なし

【参考】 Cusb18_Open

8) Cusb18_CloseAll (デバイスクローズALL)

【機能】 オープンしている全てのUSBポート等を解放します。ユーザープログラム終了直前にこの関数を実行してください。デバイスクローズするとI/Oの入出力設定がデフォルト値(入力)になります。

【引数】 なし

【書式】 VC : void Cusb18_CloseAll(void)
 VB (Ver6.0) : Sub Cusb18_CloseAll()
 VB (.NET 以降) : Sub Cusb18_CloseAll()

【戻値】 なし

【参考】 Cusb18_Open

9) Cdusb18_InBit (ポートから1点読み込み)

【機能】 指定した1点データを読み込みます。

【引数】 devno デバイス番号
 pos ビット番号 (IO_B0~IO_B23)
 *buf データ受信バッファ

【書式】 VC :

```
BOOL Cdusb18_InBit(int devno, UINT pos, UINT *buf)
BOOL _Cdusb18_InBit(UINT pos, UINT *buf)
```

VB (Ver6.0) :

```
Function Cdusb18_InBit(ByVal devno As Long, ByVal pos As Long,
                      ByVal buf As Long) As Long
Function SCdusb18_InBit(ByVal pos As Long, ByVal buf As Long) As Long
```

VB (.NET 以降) :

```
Function Cdusb18_InBit(ByVal devno As Integer, ByVal pos As Integer,
                      ByVal buf As Integer) As Integer
Function SCdusb18_InBit(ByVal pos As Integer,
                      ByVal buf As Integer) As Integer
```

【戻値】 0 異常終了、0以外 正常終了

10) Cdusb18_InPort (ポートから1ポート読み込み)

【機能】 指定した1ポートデータを読み込みます。ポート長は4ビットです。

【引数】 devno デバイス番号
 pos ポート番号 (IO_P0~IO_P5)
 *buf データ受信バッファ

【書式】 VC :

```
BOOL Cdusb18_InPort(int devno, UINT pos, UINT *buf)
BOOL _Cdusb18_InPort(UINT pos, UINT *buf)
```

VB (Ver6.0) :

```
Function Cdusb18_InPort(ByVal devno As Long, ByVal pos As Long,
                      ByVal buf As Long) As Long
Function SCdusb18_InPort(ByVal pos As Long, ByVal buf As Long) As Long
```

VB (.NET 以降) :

```
Function Cdusb18_InPort(ByVal devno As Integer, ByVal pos As Integer,
                      ByVal buf As Integer) As Integer
Function SCdusb18_InPort(ByVal pos As Integer, ByVal buf As Integer) As Integer
```

【戻値】 0 異常終了、0以外 正常終了

11) Cdusb18_InDport (ポートからダブルポート読み込み)

【機能】 指定したダブルポートデータを読み込みます。ダブルポートは8ビット長です。

【引数】 devno デバイス番号
 pos ダブルポート番号(I0_DP0～I0_DP2)
 *buf データ受信バッファ

【書式】 VC :

```
BOOL Cdusb18_InDport(int devno, UINT pos, UINT *buf)
BOOL _Cdusb18_InDport(UINT pos, UINT *buf)
```

VB (Ver6.0) :

```
Function Cdusb18_InDport(ByVal devno As Long, ByVal pos As Long,
                          ByRef buf As Long) As Long
Function SCdusb18_InDport(ByVal pos As Long, ByRef buf As Long) As Long
```

VB (.NET 以降) :

```
Function Cdusb18_InDport(ByVal devno As Integer, ByVal pos As Integer,
                          ByRef buf As Integer) As Integer
Function SCdusb18_InDport(ByVal pos As Integer,
                          ByRef buf As Integer) As Integer
```

【戻値】 0 異常終了、0 以外 正常終了

12) Cdusb18_InFport (ポートから4ポート読み込み)

【機能】 指定した4ポートデータを読み込みます。4ポートは16ビット長です。
 <D I Oタイプ>のみ対応。

【引数】 devno デバイス番号
 *buf データ受信バッファ

【書式】 VC :

```
BOOL Cdusb18_InFport(int devno, UINT *buf)
BOOL _Cdusb18_InFport(UINT *buf)
```

VB (Ver6.0) :

```
Function Cdusb18_InFport(ByVal devno As Long,
                          ByRef buf As Long) As Long
Function SCdusb18_InFport(ByRef buf As Long) As Long
```

VB (.NET 以降) :

```
Function Cdusb18_InFport(ByVal devno As Integer,
                          ByRef buf As Integer) As Integer
Function SCdusb18_InFport(ByRef buf As Integer) As Integer
```

【戻値】 0 異常終了、0 以外 正常終了

13) Cdbus18_InAport (ポート全て読み込み)

【機能】全てのポートデータを読み込みます。ポートは24ビット長です。<D I Oタイプ>のみ対応。

【引数】 devno デバイス番号
 *buf データ受信バッファ

【書式】 VC :

```
BOOL Cdbus18_InAport(int devno, UINT *buf)
BOOL _Cdbus18_InAport(UINT *buf)
```

VB (Ver6.0) :

```
Function Cdbus18_InAport(ByVal devno As Long,
                        ByRef buf As Long) As Long
Function SCdbus18_InAport(ByRef buf As Long) As Long
```

VB (.NET 以降) :

```
Function Cdbus18_InAport(ByVal devno As Integer,
                        ByRef buf As Integer) As Integer
Function SCdbus18_InAport(ByRef buf As Integer) As Integer
```

【戻値】 0 異常終了、0以外 正常終了

【参考】 Cdbus18_InPort, Cdbus18_InDport

14) Cdbus18_Open (デバイスオープン)

【機能】USBポートを使用できるようにオープンします。ユーザープログラム初期化時に1度実行してください。

D I Oタイプは、この関数直後全てのポートは入力設定となります。Cdbus18_SetDir 関数にて出力設定をしてください。

【引数】 devno デバイス番号

【書式】 VC : BOOL Cdbus18_Open(int devno)
 VB (Ver6.0) : Function Cdbus18_Open (ByVal devno as Long) As Long
 VB (.NET 以降) : Function Cdbus18_Open (ByVal devno as Integer) As Integer

【戻値】 0 異常終了、0以外 正常終了

【参考】 Cdbus18_Close

15) Cdbus18_OutBit (ポートへ1点出力)

【機能】 指定した1点へデータを出力します。

【引数】 devno デバイス番号
 pos ビット番号 (IO_B0～IO_B23)
 buf 出力データバッファ

【書式】 VC :

```
BOOL Cdbus18_OutBit(int devno, UINT pos, UINT buf)
BOOL _Cdbus18_OutBit(UINT pos, UINT buf)
```

VB (Ver6.0) :

```
Function Cdbus18_OutBit(ByVal devno As Long, ByVal pos As Long,
                        ByVal buf As Long) As Long
Function SCdbus18_OutBit(ByVal pos As Long, ByVal buf As Long) As Long
```

VB (.NET 以降) :

```
Function Cdbus18_OutBit(ByVal devno As Integer, ByVal pos As Integer,
                        ByVal buf As Integer) As Integer
Function SCdbus18_OutBit(ByVal pos As Integer, ByVal buf As Integer) As Integer
```

【戻値】 0 異常終了、0 以外 正常終了

16) Cdbus18_OutPort (ポートへ1ポート出力)

【機能】 指定したポートへデータを出力します。ポート長は4ビットです。

【引数】 devno デバイス番号
 pos ポート番号 (IO_P0～IO_P5)
 buf 出力データバッファ

【書式】 VC :

```
BOOL Cdbus18_OutPort(int devno, UINT pos, UINT buf)
BOOL _Cdbus18_OutPort(UINT pos, UINT buf)
```

VB (Ver6.0) :

```
Function Cdbus18_OutPort(ByVal devno As Long, ByVal pos As Long,
                        ByVal buf As Long) As Long
Function SCdbus18_OutPort(ByVal pos As Long, ByVal buf As Long) As Long
```

VB (.NET 以降) :

```
Function Cdbus18_OutPort(ByVal devno As Integer, ByVal pos As Integer,
                        ByVal buf As Integer) As Integer
Function SCdbus18_OutPort(ByVal pos As Integer,
                        ByVal buf As Integer) As Integer
```

【戻値】 0 異常終了、0 以外 正常終了

17) Cdusb18_OutDport (ポートへダブルポート出力)

【機能】 指定したダブルポートヘデータを出力します。ダブルポート長は8ビットです。

【引数】 devno デバイス番号
 pos ダブルポート番号 (IO_DP0~IO_DP2)
 buf 出力データバッファ

【書式】 VC :

```
BOOL Cdusb18_OutDport(int devno, UINT pos, UINT buf)
BOOL _Cdusb18_OutDport(UINT pos, UINT buf)
```

VB (Ver6.0) :

```
Function Cdusb18_OutDport(ByVal devno As Long, ByVal pos As Long,
                           ByVal buf As Long) As Long
Function SCdusb18_OutDport(ByVal pos As Long, ByVal buf As Long) As Long
```

VB (.NET 以降) :

```
Function Cdusb18_OutDport(ByVal devno As Integer, ByVal pos As Integer,
                           ByVal buf As Integer) As Integer
Function SCdusb18_OutDport(ByVal pos As Integer, ByVal buf As Integer) As Integer
```

【戻値】 0 異常終了、0以外 正常終了

18) Cdusb18_OutFport (ポートへ4ポート出力)

【機能】 指定した4ポートヘデータを出力します。4ポート長は16ビットです。
<DIOタイプ>のみ対応。

【引数】 devno デバイス番号
 buf 出力データバッファ

【書式】 VC :

```
BOOL Cdusb18_OutFport(int devno, UINT buf)
BOOL _Cdusb18_OutFport(UINT buf)
```

VB (Ver6.0) :

```
Function Cdusb18_OutFport(ByVal devno As Long,
                           ByVal buf As Long) As Long
Function SCdusb18_OutFport(ByVal buf As Long) As Long
```

VB (.NET 以降) :

```
Function Cdusb18_OutFport(ByVal devno As Integer,
                           ByVal buf As Integer) As Integer
Function SCdusb18_OutFport(ByVal buf As Integer) As Integer
```

【戻値】 0 異常終了、0以外 正常終了

19) Cdbus18_OutAport (全ポート出力)

【機能】全てのポートヘデータを出力します。ポート長は24ビットです。<D I Oタイプ>のみ対応。

【引数】 devno デバイス番号
 buf 出力データバッファ

【書式】 VC :

```
BOOL Cdbus18_OutAport(int devno, UINT buf)
BOOL _Cdbus18_OutAport(UINT buf)
```

VB (Ver6.0) :

```
Function Cdbus18_OutAport(ByVal devno As Long,
                          ByVal buf As Long) As Long
Function SCdbus18_OutAport(ByVal buf As Long) As Long
```

VB (.NET 以降) :

```
Function Cdbus18_OutAport(ByVal devno As Integer,
                          ByVal buf As Integer) As Integer
Function SCdbus18_OutAport(ByVal buf As Integer) As Integer
```

【戻値】 0 異常終了、0 以外 正常終了

20) Cdbus18_SelDevice (デバイス選択)

【機能】指定デバイスを選択状態にします。指定するデバイスは予め Cdbus18_Open() 関数にてデバイスオープンしてください。

【引数】 devno デバイス番号

【書式】 VC : BOOL Cdbus18_SetDevice(int devno)
 VB (Ver6.0) : Function Cdbus18_SelDevice(ByVal devno As Long) As Long
 VB (.NET 以降) : Function Cdbus18_SelDevice(ByVal devno As Integer) As Integer

【戻値】 0 異常終了、0 以外 正常終了

21) Cdusb18_SetDir (デバイス出力方向設定) DIOタイプのみ有効

【機能】 ポート (4ビット) 毎に出力設定をします。電源投入 (USBパソコン接続) またはデバイスオープン (Cdusb18_Open) 直後は、全て入力状態になっています。出力に使用するポートのみ設定してください。

一度出力設定を行うと次にデバイスオープン (Cdusb18_Close 後 Cdusb18_Open) するまでは出力設定が有効です。

【引数】

devno	デバイス番号
pos	ポート番号 (IO_P0~IO_P5)

【書式】 VC :

```
BOOL Cdusb18_SetDir(int devno, UINT pos)
BOOL _Cdusb18_SetDir(UINT pos)
```

VB (Ver6.0) :

```
Function Cdusb18_SetDir(ByVal devno As Long, ByVal pos As Long) As Long
Function SCdusb18_SetDir(ByVal pos As Long) As Long
```

VB (.NET 以降) :

```
Function Cdusb18_SetDir(ByVal devno As Integer, ByVal pos As Integer
) As Integer
Function SCdusb18_SetDir(ByVal pos As Integer) As Integer
```

【戻値】 0 異常終了、0以外 正常終了

参照) 出力設定手順

1. Cdusb18_Open
デバイスをオープン (使用できる状態) にする。
2. Cdusb18_OutPort
デフォルト値はLo(データ=0)なので、Hi(データ=1)にする時はここで設定をする。
3. Cdusb18_SetDir
出力設定をする。

※なお、出力設定前はポート入力のオープン状態なので、出力としてはハイインピーダンスになります。接続方法については1.13. デジタル入出力参考回路図を参照してください。

【 改 訂 履 歴 】

改訂番号	改訂日付	改訂内容
初版	2007. 7. 1	初版
第 3 版	2008. 8. 5	サンプルプログラム出力ポート番号訂正、Cdusb18_Open 戻り値追加
第 4 版	2008. 11. 17	DIO 電圧仕様追加
第 5 版	2009. 7. 7	“AD 変換データ”を追加 一部関数の戻り値 “0 異常終了 0 以外正常終了”に訂正 (マニュアル誤記)
第 7 版	2009. 12. 8	VC++6 サンプルプログラム追加
第 8 版	2010. 5. 1	
第 9 版	2010. 12. 8	データロガーソフト説明追加など
第 10 版	2011. 6. 30	VC#サンプルプログラム追加, ドライバ Win7 (64bit) 対応
第 11 版	2011. 8. 4	3.2 コーディングフロー追加
第 12 版	2017. 10. 10	64bit アプリケーション対応 (Cdusb.dll 64bit 版追加)

- ・本製品及び本マニュアルの内容は製品の改良のため予告無しに変更される事がありますのでご了承下さい。
- ・本製品及び本マニュアルの内容は著作権法により保護されています。許可なく複製する事はできません。

USB18FIO
ユーザーズ・マニュアル

中部電機株式会社

〒440-0004 愛知県豊橋市忠興3丁目2-8
TEL <0532>61-9566 FAX <0532>63-1081
URL : <http://www.chubu-el.co.jp>
E-mail : csg@chubu-el.co.jp

2007. 7 第 1 版発行
2017. 10 第 12 版発行