

ALGO省配線システム

ユーザーズマニュアル

**CPU付きA - Link
PCIマスタユニットDLL**

ALGO

目 次

はじめに

DLL の概要

第 1 章 アプリケーション開発

1 - 1	フォルダ構成	3
1 - 2	A-Link 動作環境	4
1 - 3	アプリケーション開発の準備	5

第 2 章 DLL 関数

2 - 1	DLL 関数概要	6
2 - 2	DLL 使用手順	9

第 3 章 構造体について

3 - 1	構造体の説明	10
-------	--------------	----

はじめに

- 本書において対象とするマスタユニットは、「CPU 付き A-Link PCI マスタユニット」です。
- 本書では、対象とするマスタユニットを Windows 上よりアクセスを容易に行うために用意されている DLL (CALink.DLL) の使用方法について説明されています。
- 「CPU 付き A-Link PCI マスタユニット」では、スレーブとのデータ入出力は、ST 区画構造体単位で行い、これをモジュール方式と呼びます。スレーブの入出力で主に使用する関数として、コマンド領域の ST 区画構造体単位で書込む<CmdWrite 関数>とレスポンス領域の ST 区画構造体単位で読む<ResRead 関数>が用意されています。
- ST 区画とは、D.P.RAM 上のコマンド領域とレスポンス領域を 63 の区画に分割し、分割された区画を ST (サテライト) 区画と呼び、ST1、2...と表現します。ST 区画番号は、A-Link のスレーブアドレスに相当します。
- スレーブアドレスに相当するコマンド領域の ST 区画にデータをセットするとスレーブに出力され、レスポンス領域の ST 区画を見ることによってスレーブからの入力を取得することができます
- A-Link マスタユニットにつながるにスレーブユニットの識別は、物理 ID を指定することによって行います。指定する ID は、「ボード番号」「ライン番号」「スレーブアドレス」となります。
- 「ボード番号」は、PCI バスに接続されている「CPU 付き A-Link PCI マスタユニット」を識別するために使用します。「ライン番号」は、「CPU 付き A-Link PCI マスタユニット」にある 2 つの A-Link 通信を識別するために使用します。「スレーブアドレス」は、A-Link に接続されるスレーブユニットを識別するために使用します。
- 関数のリターン値については、本ドキュメントでは詳しく取り扱いません。別紙「CPU 付き A-Link PCI マスタユニット DLL リファレンスマニュアル」を参照下さい。
- システム構成情報作成のためのソフトウェア「CPU 付 A-Link 設定ファイル作成ツール」は、詳しく取り扱いません。別紙「CPU 付 A-Link システム構成情報作成ツール 操作マニュアル」を、ご参照下さい。

関連図書

- ・ 「CPU 付き A-Link PCI マスタユニット DLL リファレンスマニュアル」
- ・ 「CPU 付き A-Link PCI マスタユニットドライバ インストールマニュアル」
- ・ 「A-Link スレーブ別コマンド・レスポンス ユーザーズマニュアル」
- ・ 「CPU 付 A-Link システム構成情報作成ツール 操作マニュアル」

DLL の概要

- 本 DLL「CALink.DLL」は、アルゴシステム省配線システムである「A-Link」を用いたシステムの Windows アプリケーションを作成するユーザが、容易に作成できる便宜をはかる為に提供されます。使用するマスタユニットは、「CPU 付き A-Link PCI マスタユニット」です。
- ユーザは、Microsoft Visual C++、Borland C++Builder、Borland Delphi 等の開発言語から、DLL 関数をコールすることによって A-Link 入出力を行うアプリケーションを作成することができます。
- 本 DLL は、「CPU 付き A-Link PCI マスタユニット」にアクセスする為のデバイスドライバ「CALink.sys」を内部でコールするユーザーアプリケーションがアクセスする「CALink.DLL」です。
- 各 A-Link スレーブへのデータ入出力を行うために使用する関数は、主に<CmdWrite 関数>、<ResRead 関数>です。データ入出力は、ST 区画構造体のバッファを介して行います。
- ST 区画構造体は、A-Link スレーブに割り当てられる D.P.RAM 上の ST 区画（6 ワード）を一つの構造体と捉えます。ST 区画単位でデータ入出力を行います。
- 位置決めユニット、シリアル通信ユニット等のコマンド・レスポンス型のユニットは、返答があるまで<ResRead 関数>でポーリングすることにより、レスポンスを得られます。
- 本 DLL では、割込み関数もサポートされています。割込みは、PCI マスタユニットのライン毎に発生します。割込みの種類は、「通信スキャン毎」、「通信異常（CHK2 発生）」、「8 点の入力データ変化（任意に指定可能）」をサポートします。
- モードレジスタエリア内の情報をアクセスする関数は、機能単位に準備されています。またシステム構成情報を設定する関数を準備します。
- 関数のリターン値については、本ドキュメントでは詳しく取り扱いません。別紙「CPU 付き A-Link PCI マスタユニット DLL リファレンスマニュアル」を参照下さい。
- システム構成情報を設定する関数（「ALink_SetSystemConfig」で必要となる「転送レート」「全ノ半二重」「接続されているユニット」といったシステム構成情報は、「ALGOSYSTEM CPU 付 A-Link 設定ファイル作成ツール」を用いてシステム情報ファイル（ALinkC0nfig.dat）を使用して作成します。
- 本 DLL を使用する前に、ユーザは、CPU 付き A-Link PCI マスタユニットを実行マシンの PCI スロットに挿入し、デバイスドライバのインストール作業を行っておく必要があります。

第 1 章 アプリケーション開発

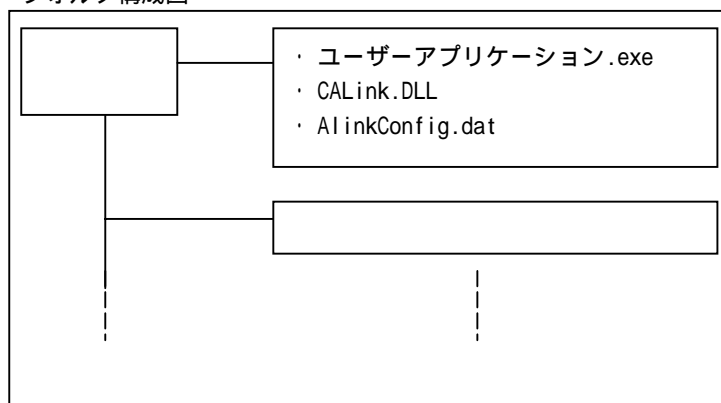
アプリケーションの開発を行うために必要な基本環境について説明します。

1 - 1 フォルダ構成

「ユーザーアプリケーション」、「CALink.DLL」、「AlinkConfig.dat」をどのようなフォルダ構成で使用すれば良いのかを示します。

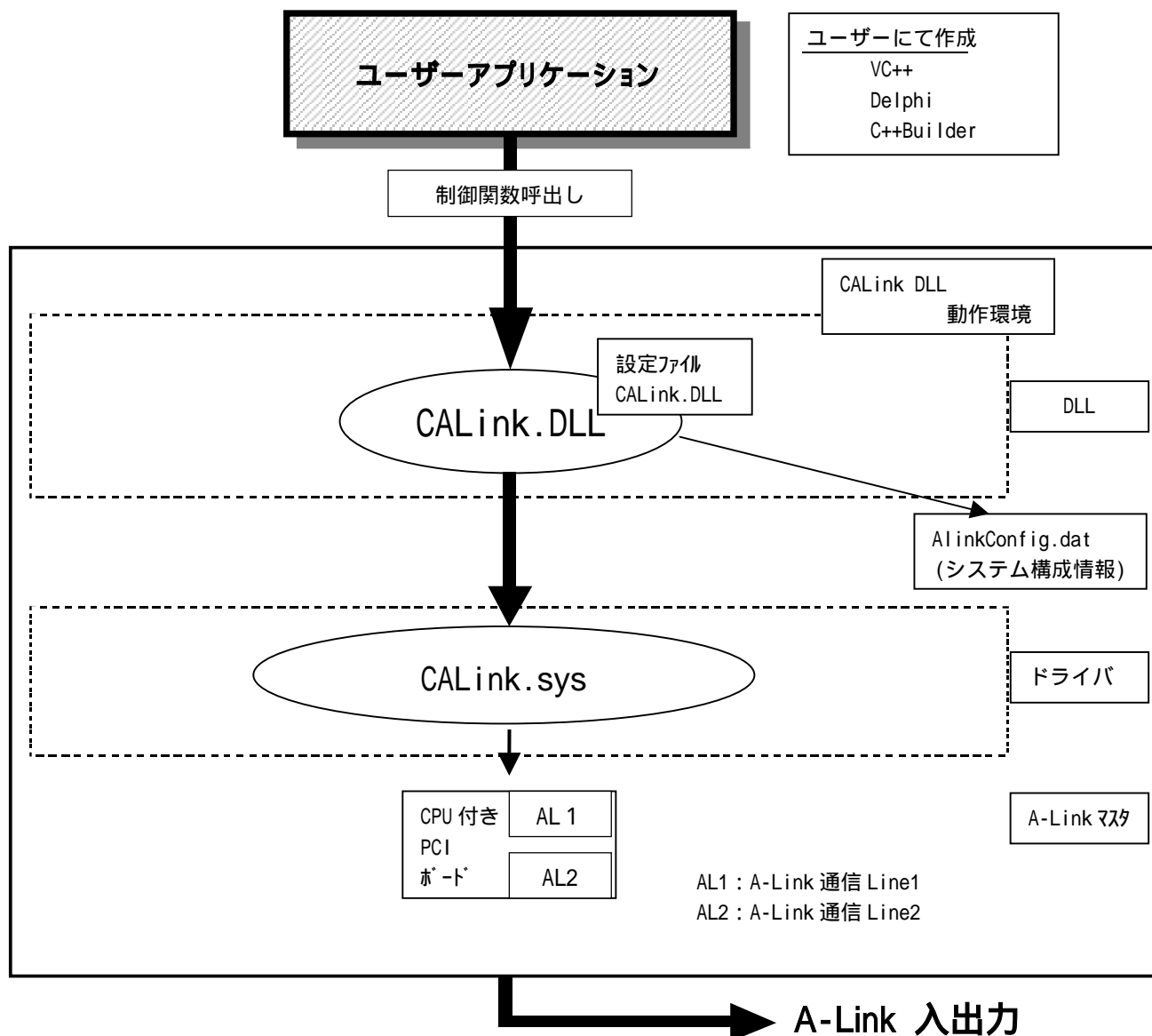
作成したアプリケーションと DLL 及びシステム構成情報ファイルは、同一フォルダに入れて下さい。

・ フォルダ構成図



1 - 2 A-Link 動作環境

ユーザは作成するアプリケーション内でCALink.DLLの関数をコールすることによりA-Linkスレーブの入出力を処理します。



1 - 3 アプリケーション開発の準備

開発アプリケーションから DLL をコールできるようにする為に、開発ユーザは、下記の手順を実行します。

1) Microsoft Visual C++

プロジェクトのソースファイルがあるフォルダに、AlgCALink.cpp、AlgCALink.H、CALinkDef.H をコピーします。

DLL の関数をコールするソースファイルへ、AlgCALink.H をインクルードします。

プロジェクトへ AlgCALink.CPP を追加します。

プログラム起動時に走る部分で、次の関数をコールして下さい。LoadCALinkDll("CALink.DLL");

プログラム終了時に走る部分で、次の関数をコールして下さい。UnloadCALinkDll();

上記以外に C++ のコンパイル設定で「プリコンパイルヘッダを使用しない」を指定して下さい。但し、「プリコンパイルヘッダを使用する」を指定する場合（ヘッダ指定が stdafx.h の時、つまりスイッチが /Yu"stdafx.h" の時）は、AlgCALink.cpp の上部 ヘッダ指定に次の 1 行追加して下さい。

```
例：      #include <windows.h>
           #include "stdafx.h"          < - - 追加行
           #include "AlgCALink.h"
```

2) Borland C++Builder

プロジェクトのソースファイルがあるフォルダに、AlgCALink.CPP、AlgCALink.H、CALinkDef.H をコピーします。

DLL の関数をコールするソースファイルへ、AlgCALink.H をインクルードします。

プロジェクトへ、AlgCALink.CPP を追加します。

プログラム起動時に走る部分で、次の関数をコールして下さい。LoadCALinkDll("CALink.DLL");

プログラム終了時に走る部分で、次の関数をコールして下さい。UnloadCALinkDll();

3) Borland Delphi

プロジェクトのソースファイルがあるフォルダに、CALinkLib.pas と CALinkConst.pas をコピーします。

プロジェクトへ CALinkLib.pas と CALinkConst.pas を追加します。

DLL の関数をコールするソースファイルの uses 節に CALinkLib.pas と CALinkConst.Pas を追加して下さい。

第 2 章 DLL 関数

本章では、用意されている DLL 関数の種類、機能及び、DLL 関数を利用した A-Link スレーブユニットとの通信の基本的な流れについて説明します。

用意されている DLL 関数の詳細については、

「CPU 付き A-Link PCI マスタユニット DLL リファレンスマニュアル」を参照下さい。

2 - 1 DLL 関数概要

DLL 関数は、マスタユニットを制御する関数とスレーブユニットを制御する関数が用意されています。関数形式は、CALink_[コマンド](引数 1, 引数 2, …) になります。

各関数の詳細は「CPU 付き A-Link PCI マスタユニット DLL リファレンスマニュアル」を参照下さい。

ここでは、関数の概要について説明します。

1) マスタ制御関数

マスタユニットの状態を操作します。

スレーブ制御関数を使用するには、先ずボードのオープンを行い、システム構成情報の設定、ラインの通信を開始させる必要があります。正常リターンを確認してから、スレーブ制御関数をコールするようにして下さい。また、通信を全て終了させる場合には、通信の停止及びボードのクローズを行って下さい。

・ マスタ制御関数

ALink_Open()	ボードをオープンします。
ALink_Close()	ボードをクローズします。
ALink_Start()	通信の開始・停止を制御します。
ALink_SetCommInfo()	通信設定を行います。
ALink_SetSystemConfig()	システム構成情報を設定します。
ALink_GetLineStatus()	ライン状態を取得します。
ALink_Write()	D.P.RAM にデータを書込みます。
ALink_Read()	D.P.RAM のデータを読込みます。

・ ALink_Open ()

- 指定したボードをオープンします。それにより、他の関数を使用できるようになります。

・ ALink_Close()

- 指定したボードをクローズします。それにより、他の関数が使用できなくなります。

・ ALink_Start()

- 「CPU 付 A-Link PCI マスタユニット」には、2 つの独立した A-Link が搭載されており、どちらの A-Link 通信ラインを使用するのかを決めます。

・ ALink_SetCommInfo()

- 通信設定構造体に「通信速度」「フレームモード」「全/半二重」「リトライ回数」情報をセットし、信設定を行います。
- 通信を開始していると設定を行えません。通信を停止した状態で使用して下さい。

- ・ ALink_SetSystemConfig()
 - ・ DLL と同じフォルダに置かれている、「ALGOSYSTEM CPU 付 A-Link 設定ファイル作成ツール」で作成したシステム情報ファイル (ALinkCOnfig.dat) を読み込みマスタユニットに登録します。
 - ・ システム情報ファイルには、A-Link 通信設定 (ボーレート等) 及び A-Link ラインに接続されているスレーブ情報が書かれています。
 - ・ A-Link に接続されているスレーブユニットが変更された場合には、再度システム情報ファイルを作成し、設定し直す必要があります。
- ・ ALink_GetLineStatus()
 - ・ 「マスタの状態」「コントロール状態」「スレーブ最終アドレス」「リトライ回数」「インクリメンタルデータ」を知ることができます。
- ・ ALink_Write()
 - ・ 指定した ALink 通信ライン用 D.P.RAM の指定したアドレスへデータを書込みます。
- ・ ALink_Read()
 - ・ 指定した ALink 通信ライン用 D.P.RAM の指定したアドレスのデータを読み込みます。

2) スレーブ制御関数

これらの関数を利用して、マスタユニットにつながる A-Link スレーブに対して、データ入出力を行います。マスタユニットに接続されたスレーブを特定するために、「ボード番号」、「ライン番号」、「スレーブアドレス」を指定してスレーブにアクセスを行います。

・ スレーブ制御関数

ALink_CmdWrite()	コマンドエリアにデータを書込みます。
ALink_CmdWriteWord()	コマンドエリアにワードデータを書込みます。
ALink_CmdRead()	コマンドエリアからデータを読み込みます。
ALink_CmdReadWord()	コマンドエリアからワードデータを読み込みます。
ALink_ResWrite()	レスポンスエリアにデータを書込みます。
ALink_ResWriteWord()	レスポンスエリアにワードデータを書込みます。
ALink_ResRead()	レスポンスエリアからデータを読み込みます。
ALink_ResReadWord()	レスポンスエリアからワードデータを読み込みます。
ALink_GetSlaveStatus()	スレーブの通信状態を取得します。

- ・ ALink_CmdWrite()
 - ・ コマンドエリアの指定した ST 区画へデータを書込みます。
- ・ ALink_CmdWriteWord()
 - ・ コマンドエリアの指定した ST 区画にワードアクセスを行います。
 - ・ コマンドエリアの ST 区間の先頭から指定した場所にワードデータを書込みます。
- ・ ALink_CmdRead()
 - ・ コマンドエリアの指定した ST 区間のデータを読み込みます。
- ・ ALink_CmdReadWord()
 - ・ コマンドエリアの指定した ST 区間の先頭から指定した場所のワードデータを読み込みます。

- ・ ALink_ResWrite()
 - ・ レスポンスエリアの指定した ST 区画ヘータを書込みます。
- ・ ALink_ResWriteWord()

レスポンスエリアの指定した ST 区画にワードアクセスを行います。

 - ・ レスポンスエリアの ST 区間の先頭から指定した場所にワードデータを書込みます。
- ・ ALink_ResRead()
 - ・ レスポンスエリアの指定した ST 区画のデータを読込みます。
- ・ ALink_ResReadWord()
 - ・ レスポンスエリアの指定した ST 区画の先頭から指定した場所のワードデータを読込みます。
- ・ ALink_GetSlavesStatus()
 - ・ スレーブの通信状態を知ることができます。

3) 割込関数

これらの関数を利用して、割込みを行います。

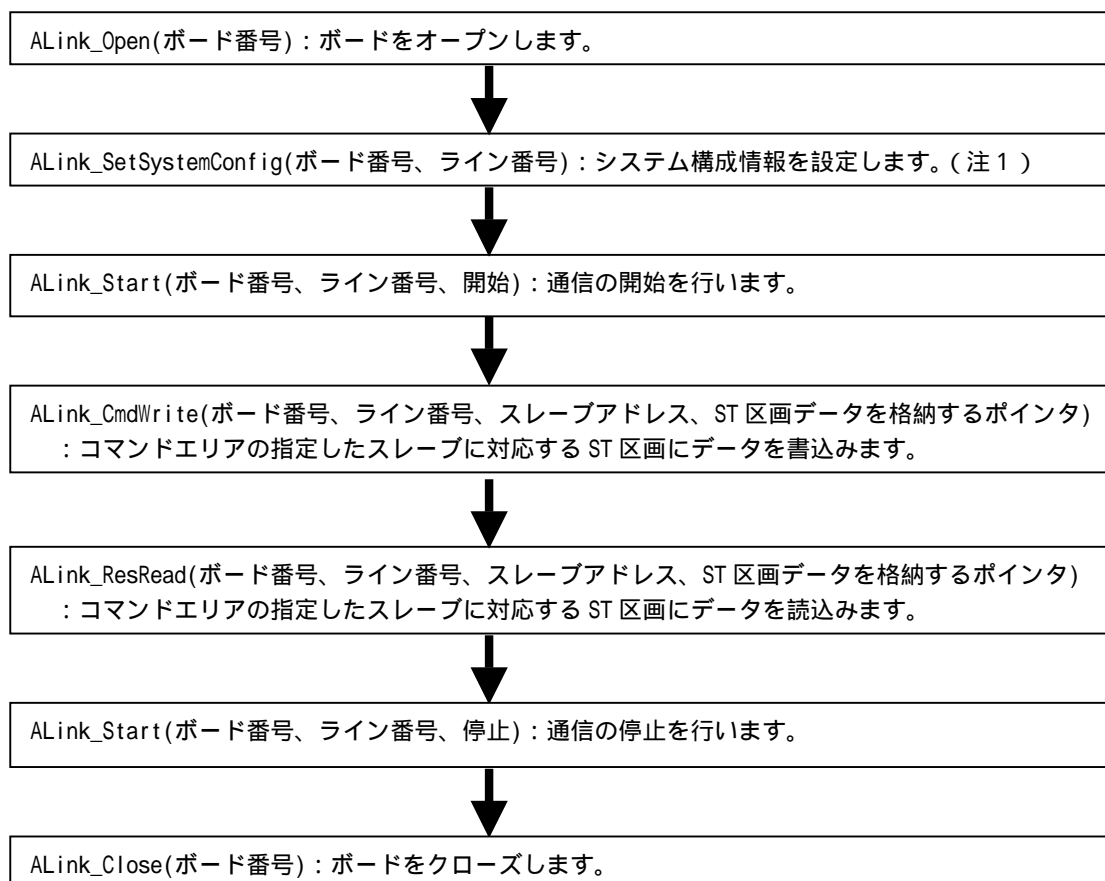
- ・ 割込関数

ALink_SetInterrupt()	割込みの設定をします。
CALINK_INTHANDLER()	割込み処理コールバック関数書式
ALink_SetDataRenewal()	入力変化割込みの設定を行います。

- ・ ALink_SetInterrupt()
 - ・ 指定した A-Link 通信ラインの割込み設定を行います。
 - ・ 割込み設定は通信開始とともに有効となります。
 - ・ 通信通信中は設定を行うことが出来ません。通信を終了させてから設定を行う必要があります。
- ・ CALINK_INTHANDLER()
 - ・ 割込発生時に、コールされる関数で、指定された指定された書式に従ってユーザは、定義できます。
- ・ ALink_SetDataRenewal()
 - ・ 入力変化割込みデータの設定を行います。

2 - 2 DLL 使用手順

マスタ制御関数及びスレーブ関数を使用して、ボードのオープンからスレーブのデータ入出力を行い、ボードのクローズまで基本的な流れを、使用する関数を用いて示します。



(注 1) 一度設定すると、設定情報は、ボードに記憶されます。A-Link に接続されているスレーブユニットの変更、スレーブユニットの通信設定を変更された場合には、再度、実行して下さい。

第 3 章 構造体について

本章では、使用する ST 区画構造体について説明します。

ST 区画に書込まれるデータは、利用されるスレーブユニットのタイプにより異なります。

スレーブタイプ別のコマンド・レスポンスフォーマットは、

別紙「A-Link スレーブ別コマンド・レスポンス ユーザーズマニュアル」を参照下さい。

3 - 1 構造体の説明

本 DLL では、A-Link スレーブユニットとの通信を D.P.RAM を介したコマンド・レスポンス方式で行います。

D.P.RAM 上のコマンド領域の ST 区画にデータをセットするとスレーブに出力され、レスポンス領域の ST 区画を見ることによってスレーブからの入力を取得できます。

ST 区画とは、D.P.RAM のコマンド・レスポンス領域、それぞれを 63 の区画に分割し、分割された区画を ST 区画と呼びます。ST 区画は、ST1,ST2 ... ST63 と表現され、スレーブアドレスと対応します。

本 DLL では、下図の ST 区画を 1 構造体としてデータのやり取りを行います。

コマンドエリア

	D15	D8 D7	D0
+0	コマンド1		
+1	コマンド2		
+2	データ1		
+3	データ2		
+4	データ3		
+5	データ4		

レスポンスエリア

	D15	D8 D7	D0
+0	レスポンス1		
+1	レスポンス2		
+2	データ1		
+3	データ2		
+4	データ3		
+5	データ4		

このユーザーズマニュアルについて

- (1)本書の内容の一部または全部を当社からの事前の承諾を得ることなく、無断で複写、複製、掲載することは固くお断りします。
- (2)本書の内容に関しては、製品改良のためお断りなく、仕様などを変更することがありますのでご了承下さい。
- (3)本書の内容に関しては万全を期しておりますが、万一ご不審な点や誤りなどお気づきのことがございましたらお手数ですが巻末記載の弊社もしくは、営業所までご連絡下さい。

改訂履歴

日時	バージョン	変更点
2005.03.22	Rev 1.00	初版

7 6 D L H 0 0 2 0 A

ALGO 株式会社アルゴシステム

本社

〒587 0021 大阪府堺市美原町小平尾656

TEL (072) 362-5067

FAX (072) 362-4856

大阪営業所

〒542-0081 大阪市中央区南船場1-12-3
船場グランドビル3F

TEL (06) 6263-9575

FAX (06) 6263-9576

東京営業所

〒104-0061 東京都中央区銀座7-15-8
銀座堀ビル2F

TEL (03) 3541-7170

FAX (03) 3541-7175

ホームページ <http://www.algosystem.co.jp/>