Digital
Human Machine Interface

Pro-face

パネルコンピュータ
PL-6920/PL-7920 シリーズ
ユーザーズマニュアル

本マニュアルを印刷する際は、プリンタを高品位に設定してください。
正しく印刷されない場合があります。

株式会社デジタル

はじめに

このたびは、(株)デジタル製のパネルコンピュータ PL-6920/PL-7920 シリーズ(これより「PL」と称します。)をお買い上げいただき、誠にありがとうございます。PL は、低価格で高性能の最新アーキテクチャを実現した多目的 FA コンピュータです。

ご使用にあたっては、本書をよくお読みいただき、PL の正しい取り扱い方法と機能をご理解いただきますようお願いいたします。

お断り

(1) 本製品および本書の内容の、一部または全部を無断で転載することは禁止されています。

(2) 本製品および本書の内容に関しては、将来予告なしに変更することがありますのでご了承ください。

(3) 本製品および本書の内容に関しては、万全を期して作成いたしましたが、万一誤りや記載もれなど、ご不審な点がありましたらご連絡ください。

(4) 本製品を使用したことによるお客様の損害その他の不利益、または第三者からのいかなる請求につきましても、当社はその責任を負いかねますので、あらかじめご了承ください。

本書に記載の商品名は、それぞれの権利者の商標または登録商標です。

安全に関する使用上の注意

本書には、㈱デジタル製のパネルコンピュータ PL-6920/7920 シリーズ（これより「PL」と称します）を正しく安全にお使いいただくために安全表記が記述されています。本書ならびに関連マニュアルをよくお読みいただき、PL の正しい取り扱い方法と機能を十分にご理解いただきますようお願いします。

■絵表示について

本書では、PL を正しく使用していただくために、注意事項に次のような絵表示を使用しています。ここで示した注意事項は、安全に関する重大な内容を記載しています。

その表示と意味は次のようになっています。

⚠警 告	この表示を無視して誤った取り扱いをすると、人が死亡または重傷を負う可能性が想定される内容を示します。
⚠注 意	この表示を無視して誤った取り扱いをすると、人が傷害を負ったり、物的損害の発生が想定される内容を示します。

⚠警 告

- **電源ケーブルの取り付けは必ず電源が供給されていないことを確認してから行ってください。感電の恐れがあります。**
- **表示された電源電圧以外の電圧で使用しないでください。火災、感電の恐れがあります。**
- **PL の本体カバーを開けるときは、必ず電源を切ってください。内部には高電圧部分があり危険です。**
- **PL は改造しないでください。火災、感電の恐れがあります。**
- **装置の安全性にかかわるタッチスイッチをPL上に設けないでください。非常停止スイッチなどの安全性に関わるスイッチは、別系統のハードウェアスイッチを設けてください。**
- **人的損害や物的損害をもたらす可能性があるスイッチは、絶対にタッチパネル上に作らないでください。本体、ユニット、ケーブル等の故障により、意図しない出力信号が出て重大な事故につながる可能性があります。重大な動作を行うスイッチはPL本体以外の装置より行うようにシステム設計をしてください。**
- **バックライトが切れると、画面が真っ暗になって表示が見えなくなりますが、バックライト消灯機能作動時と異なり、タッチスイッチの入力は有効なままです。操作者がバックライト消灯状態と間違えてタッチパネルを押した場合、不当なタッチパネル操作となる恐れがあります。不当な操作による人的・物的損害が生じる恐れのあるタッチスイッチを PL 上に設けないでください。**
  バックライトが切れた場合は以下のような現象が発生します。

  ①バックライト消灯スクリーンセーバーを設定していないのに画面の表示が消える

  ②バックライト消灯スクリーンセーバーを設定していて画面の表示が消えた際に、一度タッチなどの入力を行っても表示が復帰しない

⚠警 告

- 万一、異物（金属片、水、液体）が機器の内部に入った場合は、すぐにPLの電源を切り電源ケーブルを外して、販売店または当社までご連絡ください。
- PLを設置する際には、本書の「第4章 設置と配線」をよく読んで、適切な場所に正しく設置してください。
- 各ボードやインターフェイスの挿入および抜き取りは、必ず電源を切ってから行ってください。
- 可燃性ガスのあるところでは使用しないでください。爆発の恐れがあります。
- PLは航空機器、航空宇宙機器、幹線通信機器、原子力制御機器、生命の維持に関わる医療機器などの極めて高度な信頼性・安全性が求められる用途への使用を想定しておりません。これらの用途には使用できません。
- PLを運送機器（列車、自動車、船舶等）、防災防犯装置、各種安全装置、生命の維持に関わらない医療機器などの、機能・精度において高い信頼性・安全性が求められる用途で使用する場合は、組み込まれるシステム機器全般として、冗長設計、誤動作防止設計等の安全設計を施す必要があります。

⚠注 意

- PLの表示部を強い力や硬い物質で押さえないでください。表示部が割れ危険です。シャープペンシルやドライバーのように先が鋭利なもので、タッチパネルを押さないでください。破損のおそれがあります。
- PLの表面が汚れた場合は乾いたやわらかい布に薄めた中性洗剤をしみ込ませ、硬くしぼってふき取ってください。シンナーや有機溶剤などでふかないでください。
- PLを直射日光の当たる場所や、高温、粉塵、湿気もしくは振動の多いところで使用および保管しないでください。
- 温度変化が急激で結露するような場所での使用はお避けください。故障の原因となります。
- PLの温度上昇を防ぐため、PLの通風孔をふさいだり熱がこもるような場所での使用は避けてください。
- 薬品が気化し、発散している空気や薬品が付着する場所での使用および保管は避けてください。

⚠注 意

ハードディスクユニットに記録された内容(データやソフトウェア)が失われた場合

- **いかなる原因によるものでも弊社ではそれら記録内容に関する補償の責任は負いかねます。重要なデータやソフトウェアについては、外部記憶装置へのバックアップなど、お客様において対策していただきますようお願いいたします。**
- **お客様が運用した結果の影響については、責任を負いかねますのでご了承ください。**
- **ソフトウェア・ハードウェアトラブルによって発生した機会損失に関しても補償は一切できかねますのでご了承ください。**
- **ハードディスクは寿命部品です。データのバックアップや保持、メンテナンスを計画的に実施していただきますようお願いします。**
- **ファイル破損を防ぐため、必ずOSを終了してから、コンピュータの電源を切るようにしてください。**
- **コンピュータの電源を切った後、ハードディスクの回転が完全に止まるまでは、電源を再投入しないでください。再投入まで約5秒必要です。**

液晶パネルに関する注意とお願い

以下の記載事項以外の仕様につきましては弊社営業担当までお問い合わせください。

- **液晶ディスプレイの内部には、刺激性物質が含まれています。万一の破損により液状の物質が流出して皮膚に付着した場合は、すぐに流水で15分以上洗浄してください。また、目に入った場合は、すぐに流水で15分以上洗浄した後、医師にご相談ください。**
- **液晶ディスプレイは表示内容やコントラスト調整などにより、明るさのムラが生じることがありますが、故障ではありませんのでご了承ください。**
- **液晶ディスプレイの素子には、微細な斑点（黒点、輝点）が生じることがあります。これは故障ではありませんのでご承知ください。**
- **液晶ディスプレイの画面を視野角外から見ると表示色が変化して見えます。これは液晶ディスプレイの基本的特性ですのでご了承ください。**
- **同一画面を長時間表示していると表示されていたものが残像として残ることがあります。このような場合は、いったん電源を切り、しばらくしてから再度電源を入れると戻ります。これは液晶ディスプレイの基本的特性ですのでご了承ください。**

  残像を防ぐには以下のようにしてください。

  *** 同一画面で待機する場合は、表示OFF機能を使用する。**

  *** 表示画面を周期的に切り替えて、同一画面を長時間表示しない。**

もくじ

第 1 章 概要

第 2 章 仕様

第 6 章 PL のセットアップ

第 7 章 保守と点検

付録

マニュアル表記上の注意

本書で使用している用語や記号等の意味は以下のとおりです。

重要	この表示の説明に従わない場合、機器の異常動作やデータの消失などの不都合が起こる可能性があります。
	参考事項です。補足説明や知っていると便利な情報です。
※1	脚注で説明している語句についています。
参照→	関連事項の参照ページを示します。
① ②	操作手順です。番号に従って操作を行ってください。
PL-X920シリーズ	パネルコンピュータ PL-6920、PL-7920、PL-B920シリーズの総称です。

シリーズ構成一覧

型式

PL*92*-T4*

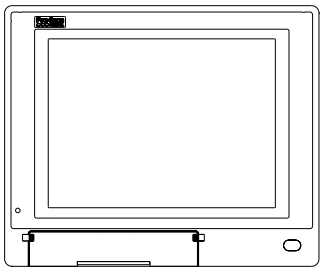
A B C D E

A	6	PL-6920シリーズ
	7	PL-7920シリーズ
B	0	4スロットタイプ
	1	2スロットタイプ
C	T	TFTカラーLCDタイプ
D	4	CEマーキング、UL/c-UL(CSA)規格対応
E	1	CPU:700MHz
	2	CPU:1GHz

梱包内容

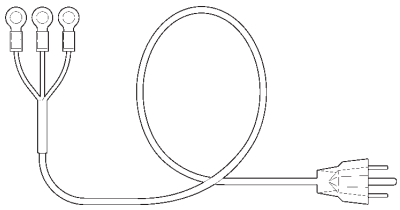
梱包箱には、以下のものが入っています。ご使用前に必ず確認してください。

■ PL本体
(PL-6920/PL-6921/PL-7920/PL-7921)



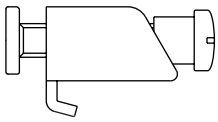
重要 **・ハードディスク組み込みタイプは、取り扱いに注意してください。**

■ 電源ケーブル



重要 **・AC100V専用です。その他の電圧では規格に合ったケーブルを使用してください。**

■ 取り付け金具



PL-6920シリーズは8個1組
PL-7920シリーズは12個1組

■ 取扱説明書
（日本語1部／英語1部）

■ CD-ROM　1枚
「PL-X920 Series User Manual & Driver CD」

MEMO ・ CDにはユーザーズマニュアルとPL-X920用ドライバおよびユーティリティが収録されています。詳細につきましては **第6章　PLのセットアップ**をご覧ください。

オプション品組み込み出荷の場合、オプション品の取扱説明書も入っています。各オプション品の取扱説明書に記載の梱包内容も合わせて確認してください。

特長

PL-6920/PL-7920 シリーズには、次のような特長があります。

● 高性能最新アーキテクチャを実現

CPU として Pentium® Ⅲ(700MHz/1GHz)を採用しています。これにより、PC/AT 互換機として求められる高性能のアーキテクチャを実現できます。また Windows®XP 等、負荷の大きな OS に対しても快適な使用環境です。

● 高輝度・広視野角のカラーディスプレイ

大画面・高輝度・広視野角の LCD ディスプレイには、TFT カラー LCD を搭載し、優れたスペックを実現しています。

・ TFT カラー LCD は、高輝度・広視野角の 64K 色カラー表示で優れた表現力を持っています。

● 機器組み込み専用前面取り付けタイプ

本体を前面から取り付ける機器組み込み専用タイプです。また、FA などの過酷な環境でもご使用いただけるよう、耐環境性にも優れています。(IP65f 相当)

● 高分解能アナログ抵抗膜方式タッチパネル搭載

1024×1024 の高分解能タッチパネルを搭載しています。タッチパネルはマウス機能をエミュレーションしており、マウス相当のオペレーション操作が可能です。

● 高い拡張性

ISA バスの拡張スロット数により、2 スロットタイプ（PL-6921/PL-7921）と 4 スロットタイプ（PL-6920/7920）を用意しています。2 スロットタイプでは第 2 スロットを、4 スロットタイプでは第 2 スロット、第 3 スロットを PCI バスとしても使用できます。また、（株）デジタル製オプション品や市販の拡張ボードを使用できます。-5V/-12V 電源ユニットや DIM モジュールなどのオプション品も用意しています。

UL/c-UL(CSA)認定について

PL6920-T4*、PL6921-T4*、PL7920-T4*、PL7921-T4*はUL/c-UL(CSA)1950部品認定品です（UL File No.E171486）。PLを組み込んだ機器をUL申請する際は、以下の事項にご注意ください。PLを組み込んだ機器は、PLとの組み合わせの適合性がULによって審査されなければなりません。

・ PLは以下の規格に部品として適合しています。
UL1950第3版1998年3月1日（電気式事務機器を含む情報技術機器の安全性に関する規格）
CAN/CSA-C22.2 No.950-95　（電気式事務機器を含む情報技術機器の安全性に関する規格）

PL6920-T4*（UL登録型式:2780054-04）、PL7920-T4*(UL登録型式:2780054-02)
PL6921-T4*（UL登録型式:2780054-03）、PL7921-T4*(UL登録型式:2780054-01)

以下の条件が満たされていないと、PLがUL/c-UL規格の要求を満たさなくなる可能性があります。

・ 機器に組み込んで使用してください。
・ 室内専用機として使用してください。
・ 電源を接続する際は、電流・電圧を考慮し、導体部の太さが0.75㎟以上のケーブルを使用してください。
・ PLを組み込んだ機器には、オペレータが容易に操作できる位置にPLの電源を切断できるスイッチなどを設けてください。スイッチには電流・電圧を考慮したものを使用してください。
・ バックアップ用電池を誤って交換すると、爆発する危険性があります。製造者の指定する製品か、それと同じタイプの製品と交換してください。使用後の電池を破棄する際は、製造者の指示に従ってください。
・ PLを組み込んだ機器はUL1950に適合した筐体構造にしてください。

CEマーキングについて

PL6920-T4*、PL6921-T4*、PL7920-T4*、PL7921-T4*は、EMC指令に適合したCEマーキング製品です。

＜適合している規格＞

Safety
EN60950

EMI
EN55011(Group1 ClassA)、EN61000-3-2、EN61000-3-3

EMS<EN61000-6-2>
EN61000-4-2、EN61000-4-3、EN61000-4-4、EN61000-4-5、EN61000-4-6、EN61000-4-8、EN61000-4-11

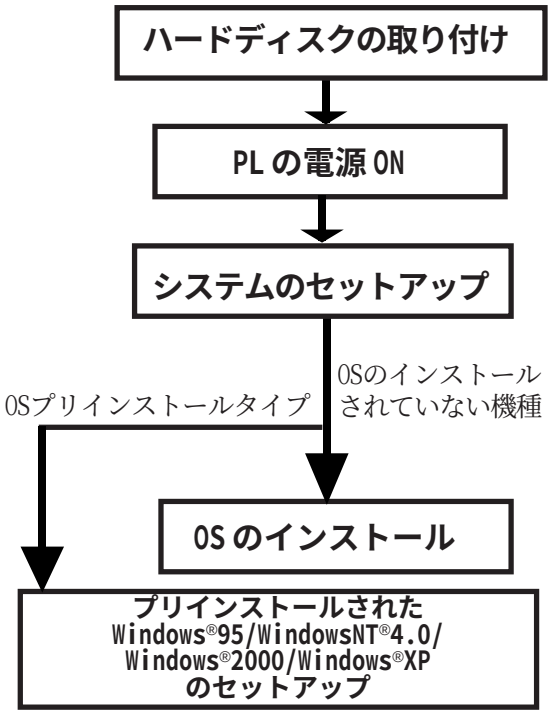
以下の条件が満たされていないと、PLがEN60950の要求を満たさなくなる可能性があります。

・ 機器に組み込んで使用してください。
・ 室内専用機として使用してください。
・ 電源を接続する際は、電流・電圧を考慮し、導体部の太さが0.75㎟以上のケーブルを使用してください。
・ PLを組み込んだ機器には、オペレータが容易に操作できる位置にPLの電源を切断できるスイッチなどを設けてください。スイッチには電流・電圧を考慮したものを使用してください。
・ バックアップ用電池を誤って交換すると、爆発する危険性があります。製造者の指定する製品か、それと同じタイプの製品と交換してください。使用後の電池を破棄する際は、製造者の指示に従ってください。
・ PLを組み込んだ機器はEN60950に適合した筐体構造にしてください。

第 1 章 概要

1.1 ご使用になる前に

PL本体をご使用になる前に、必ず以下の処理を行ってください。



参照→ OSプリインストールタイプの場合は、**プリインストール取扱説明書**
OSなしタイプの場合は、**PL-HD220取扱説明書**

参照→ 4.2 配線について

参照→ 第5章 システムのセットアップ

参照→ 6.2.1 OSなしタイプのセットアップ

参照→ 6.2.2 OSプリインストールタイプのセットアップ

重要

- **ハードウエアセットアップの後、実際にハードディスクにデータやアプリケーションを記録するためには、使用するオペレーティングシステム(MS-DOS®やWindows®等)パーティション（記録区画）の作成とフォーマット（初期化）が必要です。ご使用になるオペレーティングシステムの取扱説明書をよくお読みになり、正しくご使用ください。**
- **PLの電源を切った後、PLの電源を再投入する場合は、5秒以上の間隔を置いてください。正常に起動しない場合があります。**
- **PLで対応しているOSは、Windows®95 OSR2以上、Windows®98 Second Edition、WindowsNT®4.0、Windows®2000、Windows®XPです。それ以外のOSでは、ドライバなどのユーティリティソフトがサポートされていません。**

1.1.1 タッチパネルの接続について

タッチパネル接続方法には、シリアル接続(RS-232C)、もしくはUSB接続があります。タッチパネル接続方法により、対応するOSが異なります。

タッチパネル接続方法	対応OS
USB接続	Windows® 98 Second Edition Windows® 2000 Windows® XP
シリアル (RS-232C)接続	Windows® 95 Windows® 98 Second Edition Windows NT®4.0 Windows® 2000 Windows® XP

出荷時の設定は、シリアル接続です。タッチパネルの接続方法をUSB接続にする場合は、以下のように設定を変更してください。

◆システム情報の設定

システム情報の詳細については、**5.2 システム情報の設定内容**をご参照ください。

システム情報メニュー	設定項目	USB接続設定
Integrated Peripherals	USB Controller	Enabled
	Onboard Serial Port 4	Disabled
PnP/PCI Configuration	Assign IRQ For USB	Enabled

◆タッチパネルI/Fセレクトスイッチ(T-MODE)

タッチパネルI/Fセレクトスイッチ(T-MODE)をUに設定します。
タッチパネルI/Fセレクトスイッチ(T-MODE)の詳細については、**2.4 各部名称とその機能**をご参照ください。

◆マウスエミュレーションソフトウェア(UPDD)

マウスエミュレーションソフトウェアのインストール時にUSBを選択します。

1.1.2 USB I/Fの使用について

出荷時の設定では、USB I/Fを使用することはできません。USB I/Fに周辺機器を接続する場合は、システム情報の設定を以下のように変更する必要があります。

システム情報メニュー	設定項目	USB接続設定
Integrated Peripherals	USB Controller	Enabled
PnP/PCI Configuration	Assign IRQ For USB	Enabled

重要 **・ 市販のUSB HUBを使用する場合、2段以上のHUB接続はできません。USB HUBによるUSBデバイス中継は1段までとしてください。**

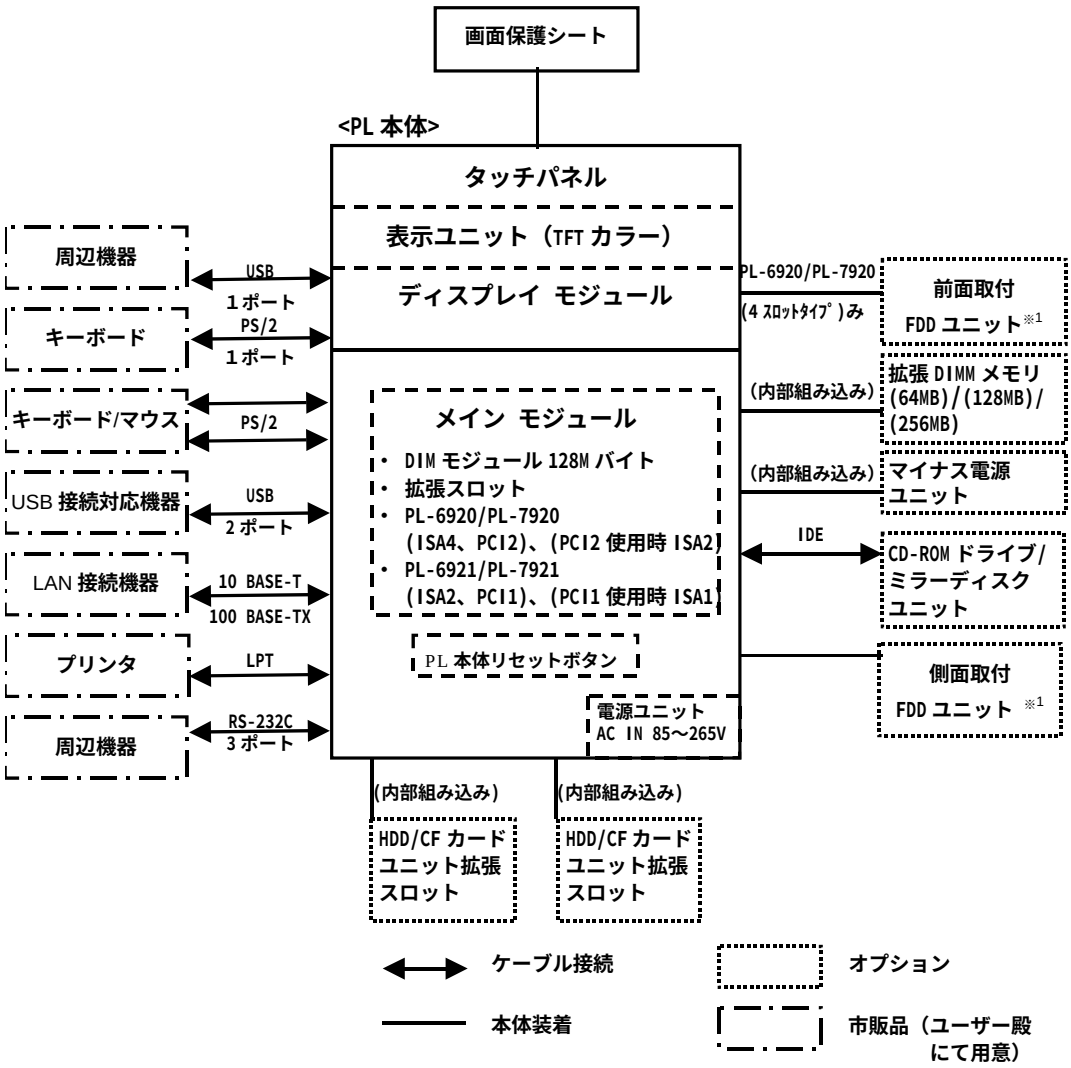
1.1.3 LAN I/Fの使用について

出荷時の設定では、LAN I/Fを使用することはできません。LAN I/Fを使用する場合は、システム情報の設定を以下のように変更する必要があります。

システム情報メニュー	設定項目	LAN接続設定
Integrated Peripherals	Onboard LAN	Enabled

1.2 システム構成図

PL と接続する周辺機器を示します。



重要 ・ 上図は、PLの内部処理の流れや周辺機器との接続について示したものです。PL の実際の部品配置とは異なります。

※1 FDD ユニットは前面用 FDD ユニットか側面用 FDD ユニットのいずれか一方しか取り付けることはできません。

1.3 オプション機器一覧

(株) デジタルのオプション品です。

■オプション

DIMモジュール	PL-EM500	SDRAM(DIMM)。容量は64Mバイトです。
	PL-EM128	SDRAM(DIMM)。容量は128Mバイトです。
	PL-EM256	SDRAM(DIMM)。容量は256Mバイトです。
FDDユニット ※1	PL-FD200	PC/AT互換、3.5型のFDDユニットです。 側面取り付けタイプ
	PL-FD210	PC/AT互換、3.5型のFDDユニットです。 前面取り付けタイプ（PL-6920/PL-7920<4スロットタイプ>用）
-5V/-12V 電源ユニット	PL-PW100	拡張スロットに-5Vと-12Vを供給するユニットです。2スロットの合計で各200mAまでの電流を取ることができます。
CD-ROMドライブユニット	PL-DK200	IDE(ATAPI)規格対応の外付けCD-ROMドライブユニットです。 (ケーブル付属)
CFカードユニット	PL-CF200	5V電源仕様のCFカード専用ユニットです。
HDDユニット	PL-HD220	2.5型のハードディスクを搭載した専用HDDユニットです。容量は20Gバイトです。OSは含まれていません。
Windows®95プリインストールHDDユニット	PL-HDX920-W95	2.5型のハードディスクを搭載した専用HDDユニットです。容量は20Gバイト。(Cドライブとして20Gバイトを設定済み) Windows®95プリインストールタイプです。（本体とセットでのみ販売可能）
WindowsNT®4.0プリインストールHDDユニット	PL-HDX920-NT40	2.5型のハードディスクを搭載した専用HDDユニットです。容量は20Gバイト。(Cドライブとして2Gバイトを設定済み) WindowsNT®4.0プリインストールタイプ<Service Pack 6a>です。（本体とセットでのみ販売可能）
Windows ®2000プリインストールHDDユニット	PL-HDX920-W2K	2.5型のハードディスクを搭載した専用HDDユニットです。容量は20Gバイト。(Cドライブとして20Gバイトを設定済み) Windows®2000 Professionalプリインストールタイプ<Service Pack 2>です。（本体とセットでのみ販売可能）
Windows ®2000プリインストールHDDユニット	PL-HDX920-W2K/ML	2.5型のハードディスクを搭載した専用HDDユニットです。容量は20Gバイト。(Cドライブとして20Gバイトを設定済み) Windows®2000 Professional Multi Language プリインストールタイプ<Service Pack 2>です。（本体とセットでのみ販売可能）
Windows ®XPプリインストールHDDユニット	PL-HDX920-WXP	2.5型のハードディスクを搭載した専用HDDユニットです。容量は20Gバイト。(Cドライブとして20Gバイトを設定済み) Windows®XP Professionalプリインストールタイプ<Service Pack 1>です。（本体とセットでのみ販売可能）
ソフトミラーユーティリティ	PL-SM900	ミラーディスクユニットを使用せずに、RAID 1(ミラーリング)を構築するソフトです。
フルサイズボードカバー	PL-FC200	拡張スロットにISAバスフルサイズボードを使用する時のカバーです。（PL-6921/PL-7921シリーズ<2スロットタイプ>用）
	PL-FC210	拡張スロットにISAバスフルサイズボードを使用する時のカバーです。（PL-6920/PL-7920シリーズ<4スロットタイプ>用）
マウスエミュレーションソフトウェア ※2	UPDD	タッチパネルエミュレータおよびキーボードエミュレータです。(Windows®95、 WindowsNT®4.0、Windows®98 Second Edition、Windows®2000、Windows®XPで使用可能)
RS-232C/RS-485変換ユニット	PL-RC500	RS-232CインターフェイスをRS-485インターフェイスに変換するユニットです。COM2もしくはCOM3に装着して使用します。

※1 PL-FD200 と PL-FD210 を同時に使用することは、できません。

※2 マウスエミュレーションソフトウェア(UPDD)は(株)デジタルのウェブサイト (http://www.proface.co.jp/otasuke/)からダウンロードしてください。

■オプション

画面保護シート	PL-CS100	表示面の保護および防汚用の使い捨てシートです。表示面に貼ったままでタッチパネルの使用も可能です。（5枚1セット）＜PL-6920/PL-7920シリーズ共用＞
CFカード	GP077-CF20	16MバイトのCFカードです。CFカードユニット(PL-CF200)が必要です。
	GP077-CF30	32MバイトのCFカードです。CFカードユニット(PL-CF200)が必要です。
インダストリアルHUB	SPIDER8TX-PRO	産業用イーサネットHUB　DC24V

■メンテナンスオプション

ミラーディスクユニット保守用HDD	PL-MD200-MD01	2.5型2.1Gバイトのミラーディスクユニットの保守用ハードディスクです。
取り付け金具	GP070-AT01	PLの取り付け時に使用する金具です。本体に同梱されているものと同じです。（4個1セット）
防滴パッキン	PL6900-WP00	PLの取り付け時に使用する防滴パッキンです。本体に取り付けられているものと同じです。
	PL7900-WP00	
交換用バックライト	PL6920-BL00	交換用バックライトです。
	PL7900-BL00-MS	

MEMO

- PL-6920/PL-7920シリーズ（4スロットタイプ）でFDDユニットを使用する場合、側面取り付けタイプのPL-FD200と前面取り付けタイプのPL-FD210のどちらか一方のみ使用できます。両方を同時に使用することはできません。
- PLにはIDEのインターフェイスとして、HDDユニットを接続するコネクタが2つ、CD-ROMドライブユニットを接続するコネクタが1つあります。物理的には3つのIDEドライブを接続できますが、IDEインターフェイスの仕様では、1つのコントローラに対し、マスタードライブ、スレーブドライブとして各1台ずつしか同時に使用することはできません。以下に2つのIDE ドライブを使用する場合のオプション機器の組み合わせとマスタードライブ、スレーブドライブの組み合わせを示します。

HDDユニット	MS	M	M	S		
CD-ROMドライブユニット		S			S	
CFカードユニット			S	M	M	MS

MS：マスターおよびスレーブとして2台使用
M：マスターとして使用
S：スレーブとして使用

重要
- **ハードディスクには、寿命があります。万一の故障も考え、定期的なデータのバックアップや交換用HDDユニットの用意をお勧めします。**
- **ハードディスクの寿命は使用条件や環境により前後しますが、目安として周囲温度20 ℃で20,000 時間（通電時間）または5 年間のいずれか早い到達期限までです。**

■市販品

市販の拡張ボード（PCI/ISA バス互換ボード）、キーボード、マウス、プリンタなどが使用できます。USB接続の場合、USB接続対応機器も使用できます。ただし、市販のパソコン用機器にはPLで使用できないものもあります。

MEMO
- PLの拡張スロットでは、-5Vまたは-12Vの供給は行っておりません。-5Vまたは-12Vを使用したPCI/ISAバス互換ボードを使用する場合はオプションのPL-PW100をご使用ください。
- メインメモリには（株）デジタル製のDIMモジュールをご使用ください。市販のDIMモジュールの中には正常に動作しないものがあります。
- USB接続対応機器を使用する際は、各USB接続対応機器の取扱説明書をよくお読みください

第 2 章 仕様

PLの一般仕様、性能仕様、インターフェイス仕様などの仕様と名称と外観図を説明しています。

2.1 一般仕様

2.1.1 電気的仕様

	PL-6920/PL-7920 (4スロットタイプ)	PL-6921/PL-7921 (2スロットタイプ)
定格電圧	AC100V～240V	
電圧許容範囲	AC85V～265V	
定格周波数	50/60Hz	
許容瞬時停電時間	1サイクル以下(ただし、瞬時停電の間隔は1s以上)	
消費電力	150VA以下	120VA以下
絶縁耐力	AC1500V　20mA　1分間(充電部端子とFG端子間)	
絶縁抵抗	DC500Vで10MΩ以上(充電部端子とFG端子間)	

2.1.2 環境仕様

<table>
<tr><td rowspan="9">使用周囲温度</td><td rowspan="2">PL692*-T41
(CPU:700MHz)</td><td colspan="2">ファン使用</td><td>5℃～50℃(HDD使用時)</td></tr>
<tr><td colspan="2">ファン未使用※1</td><td>5℃～40℃(HDD使用時)</td></tr>
<tr><td rowspan="3">PL792*-T41
(CPU:700MHz)</td><td rowspan="2">盤内</td><td>ファン使用</td><td>5℃～50℃(HDD使用時)</td></tr>
<tr><td>ファン未使用※1</td><td>5℃～40℃(HDD使用時)</td></tr>
<tr><td>表示面側</td><td colspan="2">5℃～40℃</td></tr>
<tr><td rowspan="2">PL692*-T42
(CPU:1GHz)</td><td colspan="2">ファン使用</td><td>5℃～45℃(HDD使用時)</td></tr>
<tr><td colspan="2">ファン未使用※1</td><td>使用不可</td></tr>
<tr><td rowspan="3">PL792*-T42
(CPU:1GHz)</td><td rowspan="2">盤内</td><td>ファン使用</td><td>5℃～45℃(HDD使用時)</td></tr>
<tr><td>ファン未使用※1</td><td>使用不可</td></tr>
<tr><td>表示面側</td><td colspan="2">5℃～40℃</td></tr>
<tr><td colspan="2">保存周囲温度</td><td colspan="3">-10℃～+60℃</td></tr>
<tr><td colspan="2">使用周囲湿度</td><td colspan="3">10%RH～85%RH(結露しないこと)</td></tr>
<tr><td colspan="2">保存周囲湿度</td><td colspan="3">10%RH～85%RH(結露しないこと)</td></tr>
<tr><td colspan="2">最大湿球温度</td><td colspan="3">29℃</td></tr>
<tr><td colspan="2">じんあい</td><td colspan="3">じんあいがないこと</td></tr>
<tr><td colspan="2">汚染度</td><td colspan="3">汚染度2</td></tr>
<tr><td colspan="2">腐食性ガス</td><td colspan="3">腐食性ガスがないこと</td></tr>
<tr><td colspan="2">耐振動</td><td colspan="3">19.6m/s^2（10Hz～25Hz X,Y,Z方向 各30分）
（HDD使用時は4.9m/s^2、FDD使用時は9.8m/s^2）</td></tr>
<tr><td colspan="2">耐ノイズ（インパルスノイズ）</td><td colspan="3">ノイズ電圧：1500V
パルス幅：50ns、500ns、1μs
立ち上がり時間：1ns
（ノイズシミュレータによる）</td></tr>
<tr><td colspan="2">耐静電気放電</td><td colspan="3">4kV　IEC 61000-4-2</td></tr>
<tr><td colspan="2">ノイズイミュニティ（ファーストトランジェント・バーストノイズ）</td><td colspan="3">電源ライン：2kV　IEC 61000-4-4
COMポート：1kV　IEC 61000-4-4</td></tr>
</table>

重要

- **オプション使用時は、オプション品の仕様値も併せてご確認ください。**
- **フルサイズカバー使用の場合は、装着するボードの寸法や形状によって耐振動等の環境仕様が異なります。**
- **ハードディスクには、寿命があります。万一の故障も考え、定期的なデータのバックアップや交換用ハードディスクユニットの用意をお勧めします。**
- **ハードディスクの寿命は使用条件や環境により前後しますが、目安として周囲温度 20℃で 20,000 時間(通電時間)または 5 年間のいずれか早い到達期限までです。**
- **ハードディスクを高温・高湿度の環境で使用すると、寿命を縮める原因となります。最大湿球温度 29℃での使用を推奨します。この条件は、例えば気温 35℃で湿度 64%RH、40℃で 44%RH 程度に相当します。**

※1 本体内部にある電源ファンを取り外した場合。

2.1.3 外観仕様

		PL-6920シリーズ		PL-7920シリーズ	
		PL-6920	PL-6921	PL-7920	PL-7921
接地		保護接地（D種接地）、機能接地（D種接地）			
構造		保護構造※1：JEM1030 IP65f相当 形状：一体形 取付方法：パネル埋込取付			
冷却方法	CPU(700MHz)モデル	ヒートパイプおよび電源ファンによる空冷			
	CPU(1GHz)モデル	CPUファン一体型ヒートパイプおよび電源ファンによる空冷			
質量（HDD、FDDを含む）		9.5kg以下	8.5kg以下	10.5kg以下	9.5kg以下
外形寸法（背面突出部を含まない）		W346×H287×D170mm	W346×H287×D123mm	W374×H325×D180mm	W374×H325×D134mm
フルサイズボードカバー取り付け時の寸法（背面突出部を含まない）		W393×H287×D170mm	W393×H287×D123mm	W422×H325×D180mm	W422×H325×D134mm
RS-232C/RS-485変換ユニット取り付け時の寸法（背面突出部およびケーブル部を含ま		W346×H287×D170mm	W346×H287×D145mm	W374×H325×D180mm	W374×H325×D156mm

※1 PLをパネルに取り付けたときのフロント部分に関する保護構造です。当該試験条件で適合性を確認していますが、あらゆる環境での使用を保証しているものではありません。特に試験に規定されている油であっても、長時間にわたり噴霧状態で本機がさらされている場合や極端に粘度の低い切削油にさらされている場合などは、フロント部のシートのはがれにより油の浸入が発生することがあります。その場合は別途対策が必要となります。また、規定外の油でも同様の浸入やプラスチックが変質することがあります。本機を使用する前にあらかじめご使用の環境をご確認ください。

また、長時間使用した防滴パッキンや一度パネル取り付けした防滴パッキンはキズや汚れが付き、十分な保護効果を得られない場合があります。安定した保護効果を得るためには、防滴パッキンの定期的な交換をお勧めします。

2.2 性能仕様

2.2.1 性能仕様

CPU		PentiumⅢ(700MHz/1GHz)		
DRAM(SDRAM DIMM)		標準128Mバイト(DIMMソケット×2：最大512Mバイト)		
BIOS		AWARD PC/AT互換		
2次キャッシュメモリ		256Kバイト内蔵		
グラフィック		PL-6920ｼﾘｰｽﾞ	SVGA(800×600ドット)	
		PL-7920ｼﾘｰｽﾞ	XGA(1024×768ドット)	
		VESA 16色/256色/64K色		
ビデオメモリ		UMA方式		
パネルタッチ	方式	アナログ抵抗膜方式		
	分解能	1024×1024		
	インターフェイス	COM4	マウスエミュレーションソフトウェア(UPDD)のインストール時にシリアル(COM4)接続、USB接続のどちらかのインターフェイスを選択※1	
		USB		
インターフェイス	シリアル	RS-232C (FIFO付き)	COM1	Dsub 9ピン オス
			COM2	Dsub 9ピン オス(RI/+5V切替可)
			COM3	Dsub 9ピン オス(RI/+5V切替可)
	プリンタ	セントロニクス規格準拠(ECP/SPP/EPP対応)(Dsub 25ピン メス)		
	キーボード	PS/2インターフェイス(ミニDIN6ピン メス) 前面 1ポート/側面 1ポート		
	マウス	PS/2インターフェイス(ミニDIN6ピン メス) 側面 1ポート		
	RAS	RASインターフェイス(Dsub 25ピン オス)		
	ディスク	FDD I/F	本体側面 2モード3.5インチFD I/F	
			本体前面 2モード3.5インチFD I/F (PL-6920/PL-7920<4スロットタイプ>のみ)	
		IDE I/F	2.5インチHDD I/F ・CD-ROMドライブユニット(PL-DK200)	
	USB※2	USB 1.1対応 前面 1ポート/側面 2ポート		
	LAN※2	IEEE802.3準拠(10 Base-T/100 Base-TX自動切替)		

2.2.2 表示仕様

	PL-6920シリーズ	PL-7920シリーズ
表示デバイス	TFTカラーLCD(12インチ)	TFTカラーLCD(15インチ)
表示ドット数	800×600ドット	1024×768ドット
ドットピッチ	0.3075×0.3075mm	0.297×0.297mm
有効表示寸法	246.0×184.5mm	304.1×228.1mm
表示色、階調	64K色	
バックライト	冷陰極管(交換可能) 平均寿命:連続点灯50,000時間以上	
輝度調整	なし	
コントラスト調整	なし	

※1 OSがWindows® 95、もしくはWindows NT® 4.0の場合は、シリアル(COM4)接続で固定。

※2 BIOS設定の変更が必要です。参照→ 5.2.5 Integrated Peripherals

2.2.3 拡張スロット

		PL-6920/PL-7920 4スロットタイプ	PL-6921/PL-7921 2スロットタイプ	使用可能ボードサイズ フルサイズ カバー未使用	使用可能ボードサイズ フルサイズ カバー使用	スロット 間隔	拡張ボード の部品実装 高さ
第1スロット		ISA	ISA	163×122mm	163×122mm	20mm	13mm以下
第2スロット		PCI	PCI	250×122mm	250×122mm	25mm	18mm以下
		ISA	ISA		338×122mm		
第3スロット		PCI/ISA	なし	250×122mm	338×122mm	25mm	18mm以下
第4スロット		ISA	なし	250×122mm	338×122mm	20mm	13mm以下
供給電源	CPU 700MHz モデル	5V:4A 12V:1A (4スロット合計)	5V:2A 12V:0.5A (2スロット合計)	――――	――――	――――	――――
	CPU 1GHz モデル	5V:3.5A 12V:1A (4スロット合計)	5V:1A 12V:0.5A (2スロット合計)	――――	――――	――――	――――

MEMO ・ PL-6920/PL-7920(4スロットタイプ)の第2・3スロット、PL-6921/PL-7921(2スロットタイプ)の第2スロットはPCIまたはISAのいずれかを選択することが可能です。

2.2.4 時計精度

PLに内蔵されている時計（RTC）には誤差があります。常温無通電状態での誤差は、1ヶ月±180秒です。温度差や使用年数によっては1ヶ月に±300秒の誤差になることがあります。時計の誤差が問題となるシステムでご使用になる場合、定期的に正確な時間の設定を行ってください。

時計精度	±180秒／月

2.3 インターフェイス仕様

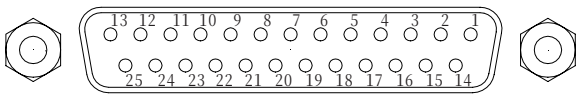
重要 ・ 拡張スロット(ISA/PCI)に接続されているオプションボードの消費電流が「2.2.3 拡張スロット」に示す仕様上限値の場合、ご使用になるI/O機器の5V消費電流値の合計が下表を満たすようにしてください。

	PLX920T-41 (CPU:700MHz) 4スロットタイプ	PLX921T-41 (CPU:700MHz) 2スロットタイプ	PLX920T-42 (CPU:1GHz) 4スロットタイプ	PLX921T-42 (CPU:1GHz) 2スロットタイプ
外部I/O機器の 消費電流値合計※1	1.5A	1.0A	0.5A	0.5A

※1 RS-232C I/F(COM2/COM3)、RAS I/F、USB I/F、マウスI/Fの+5Vより供給される電流の合計値。

2.3.1 プリンタインターフェイス(LPT1)

Dsub25 ピン（メス）



インチネジ
#4-40UNC

ピン番号	SPP/ECPモード信号名	EPPモード信号名	方向	電気的仕様	ピン番号	SPP/ECPモード信号名	EPPモード信号名	方向	電気的仕様
1	$\overline{\text{STRB}}$	$\overline{\text{WRITE}}$	入出力	O.D/T.S	14	$\overline{\text{AUTOFD}}$	$\overline{\text{DSTRB}}$	入出力	O.D/T.S
2	DATA0	DATA0	入出力	T.S	15	$\overline{\text{ERROR}}$	$\overline{\text{ERROR}}$	入力	TTL
3	DATA1	DATA1	入出力	T.S	16	$\overline{\text{INIT}}$	$\overline{\text{INIT}}$	入出力	O.D/T.S
4	DATA2	DATA2	入出力	T.S	17	$\overline{\text{SLCTIN}}$	$\overline{\text{ADSTRB}}$	入出力	O.D/T.S
5	DATA3	DATA3	入出力	T.S	18	GND	GND		
6	DATA4	DATA4	入出力	T.S	19	GND	GND		
7	DATA5	DATA5	入出力	T.S	20	GND	GND		
8	DATA6	DATA6	入出力	T.S	21	GND	GND		
9	DATA7	DATA7	入出力	T.S	22	GND	GND		
10	$\overline{\text{ACKNLG}}$	$\overline{\text{ACKNLG}}$	入力	TTL	23	GND	GND		
11	BUSY	$\overline{\text{WAIT}}$	入力	TTL	24	GND	GND		
12	PE	PE	入力	TTL	25	GND	GND		
13	SLCT	SLCT	入力	TTL					

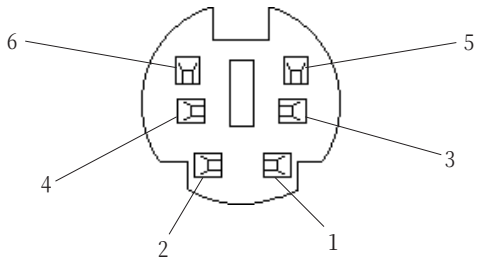
電気的仕様 … O.D:オープンドレイン、T.S:3 ステート入出力、TTL:TTL 入力

MEMO

・ 1、14、16 および 17 ピンでは、電気的仕様は SPP モードの場合、O.D となり、ECP モードおよび EPP モードの場合、T.S となります。

2.3.2 キーボードインターフェイス

ミニ DIN 6 ピン(メス)

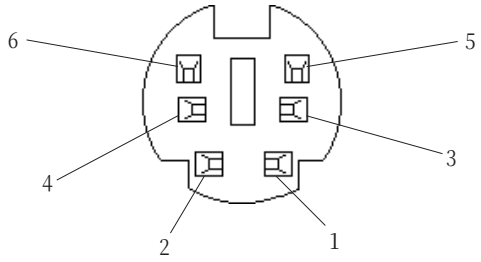


(フロント、サイド共通)

ピン番号	信号名
1	KEY DATA
2	NC
3	GND
4	＋5V
5	KEY CLK
6	NC
SHIELD	GND

2.3.3 マウスインターフェイス

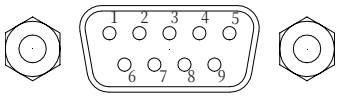
ミニDIN 6ピン(メス)



ピン番号	信号名
1	Mouse DATA
2	NC
3	GND
4	＋5V
5	Mouse CLK
6	NC
SHIELD	GND

2.3.4 RS-232C インターフェイス(COM1/COM2/COM3)

Dsub 9ピン(オス)



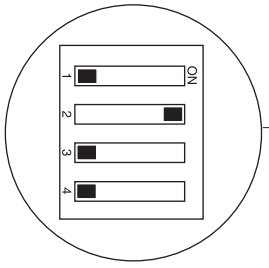
インチネジ
#4-40UNC

ピン番号	信号名	ピン番号	信号名
1	CD	6	DSR
2	RXD	7	RTS
3	TXD	8	CTS
4	DTR	9	RI/+5V
5	GND		

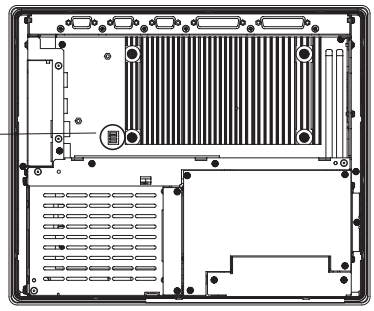
重要 ・ **GND端子は信号グランドです。接続相手のSG(信号グランド)端子と接続してください。**

9番ピンの「RI/+5V」の切替えはCOM2、COM3のみです。COM1は「RI」となります。COM2、COM3の「RI/+5V」の切替えは、本体のリアメンテナンスカバーを開け、基板上にあるメインファンクションスイッチにて行います。COM2を切替える場合は、メインファンクションスイッチのSW2をONにすると「+5V」に切り替わります。初期設定はOFFで「RI」です。COM3を切り替える場合は、同様にSW3をONにしてください。

参照→ 3.1.1 リアメンテナンスカバーの取り外し



メインファンクションスイッチ

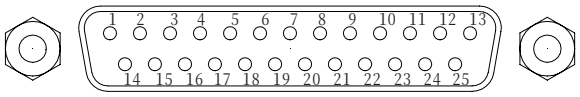


PL-6920/PL-7920(4スロットタイプ)内部

重要 ・ **メインファンクションスイッチのSW1、SW4は予約です。設定は変更しないでください。**
・ **接続相手のインターフェイス仕様を確認の上、切り替えを行ってください。誤った設定を行うと故障、誤動作の原因となります。**
・ **切り替えは必ずPL本体の電源を切った状態で行ってください。誤動作の原因となります。**

2.3.5 RAS インターフェイス

Dsub25 ピン（オス）



インチネジ
#4-40UNC

ピン番号	信号名	ピン番号	信号名
1	GND	14	GND
2	+5V（最大100mA）	15	+5V
3	+12V（最大100mA）	16	NC
4	NC	17	NC
5	リセット入力(+)	18	NC
6	DIN0(+)	19	NC
7	DOUT(-)	20	NC
8	DOUT(+)	21	ランプ出力(-)
9	アラーム出力(-)	22	ランプ出力(+)
10	アラーム出力(+)	23	NC
11	リセット入力(-)	24	DIN1(-)
12	DIN0(-)	25	NC
13	DIN1(+)		

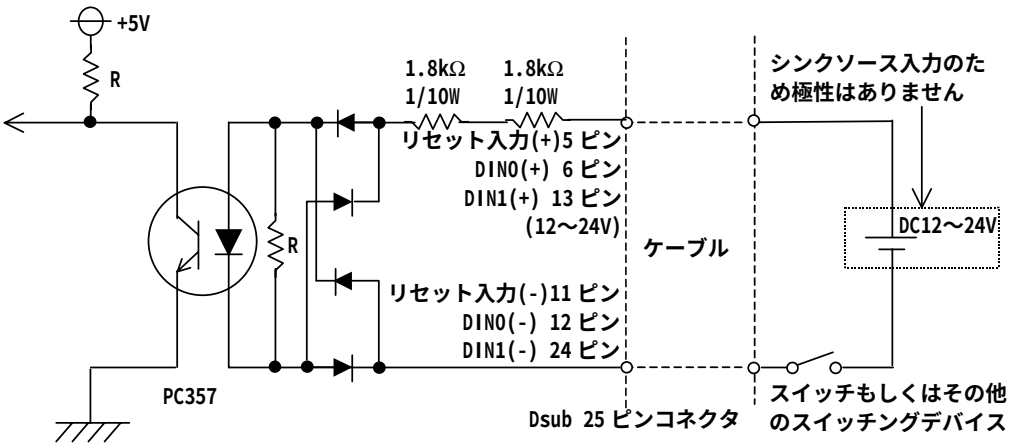
重要 **・2 番(+5V)、3 番(+12V)の外部電源出力をご使用の場合は定格電流を守ってご使用ください。誤動作、故障の原因となります。**

MEMO ・ RAS機能の詳細については、**付.2 RAS機能について**をご覧ください。

■外部入力信号（DIN、リモートセット入力共通）

入力電圧	DC12V～24V
入力電流	7mA
動作電圧	ON電圧:9V（min）、OFF電圧:3V（max）
絶縁方式	フォトカプラによる絶縁

（インターフェイス回路）　　（接続例）



重要
- **汎用信号入力(DIN)は、入力レベルを 1.5S 以上保持してください。1.5S以下では検出できないことがあります。**
- **端子間の電圧値は、入力電圧で決められた範囲内で使用してください。入力電圧範囲を超えますと故障の原因となります。**

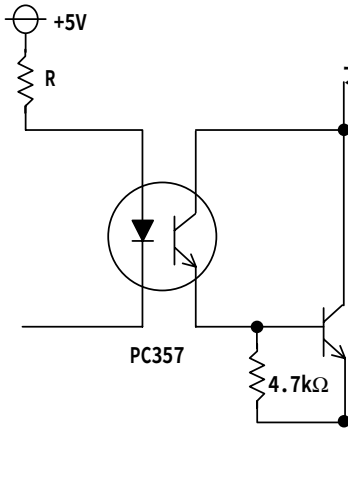
重要

- **シンクソース入力のため、D(-)、RESET(-)が正極、D(+)、RESET(+)が負極となっても問題ありません。この場合も、上記入力電圧範囲内で使用してください。**

■外部出力信号（DOUT、アラーム出力、ランプ出力共通）

定格負荷電圧	DC12V～24V
最大負荷電流	100mA/点
端子間最大降下電圧	1.5V（負荷電流100mA時）
絶縁方式	フォトカプラによる絶縁

（インターフェイス回路）



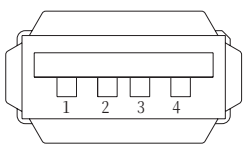
（接続例）

重要

- **最大負荷電流内で使用してください。最大負荷電流を超えて使用すると故障の原因となります。**
- **負荷の電流値および電圧値は、端子間電圧を加味したうえで設計してください。負荷電流を大きくとりますと、端子間にて最大1.5Vの電圧降下が生じます。**
- **誘導性負荷を接続する場合は上図※1の保護用ダイオードを接続してください。**

2.3.6 USBインターフェイス（USB）

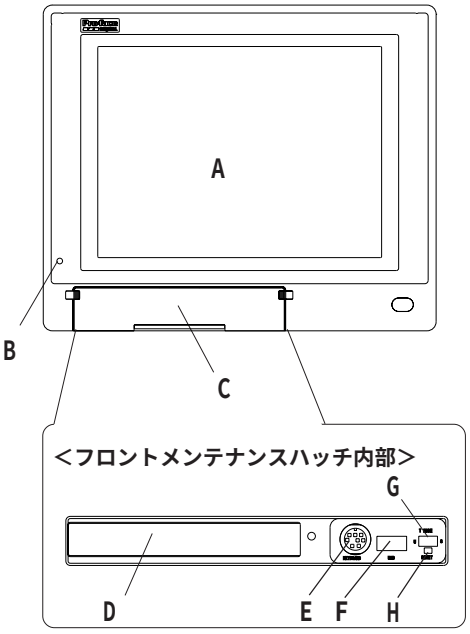
レセプタクル

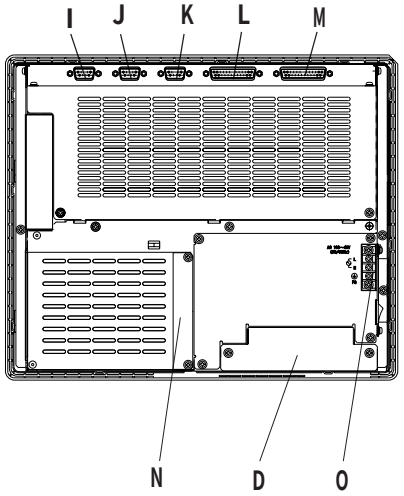


ピン番号	信号名
1	Vcc
2	- Data
3	+ Data
4	GND

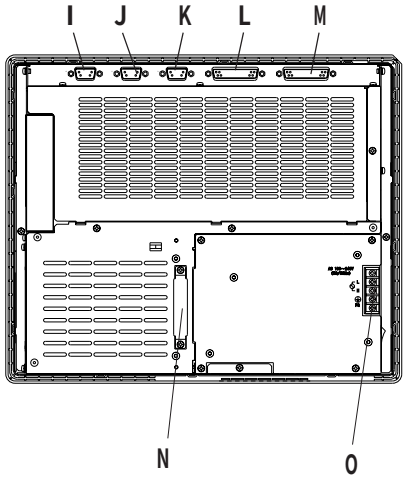
2.4 各部名称とその機能

PLの各部名称とその機能について説明します。図は、PL-6920シリーズを例にしています。





PL-6920(4スロットタイプ)背面図



PL-6921(2スロットタイプ)背面図

A: 表示部 / タッチパネル
表示出力部および高分解能のアナログ式タッチパネルです。SVGA、またはXGAコントローラを内蔵しています。キーボードレスで操作可能なシステムを構築できます。

B: 電源LED/RASステータスランプ
RAS機能のLEDインジケートと共用化されたパワーランプです。RAS機能のアラームにより点灯状態が変化します。
参照→ 付2.2 RAS機能詳細

C: フロントメンテナンスハッチ
フロントキーボードコネクタ､USBコネクタなどを使用する場合は、このハッチ(カバー)を開きます。

D: 前面取り付けFDDスロット
FDDユニット（PL-FD210）を装着するスロットです（PL-6920/PL-7920<4スロットタイプ>のみ)。

E: キーボードコネクタ(KEYBOARD)
PS/2タイプのキーボードを接続します。

F: USBコネクタ(USB)
USB 1.1対応のUSB I/Fです。USB接続対応機器を接続します。

G：タッチパネルI/Fセレクトスイッチ(T-MODE)
U- タッチデータをUSB I/Fで送信します。
S- タッチデータをシリアルI/F(COM4)で送信します。

H:ハードウェアリセットスイッチ(RESET)

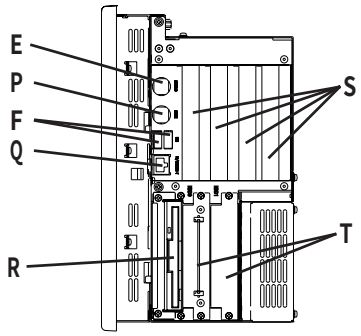
I: RS-232Cコネクタ(COM1)
J: RS-232Cコネクタ(COM2)(RI/+5V切替可)
K: RS-232Cコネクタ(COM3)(RI/+5V切替可)
RS-232Cのインターフェイスです(Dsub9ピン オス)。
他機種との通信を行ったり、周辺機器を接続します。

L: プリンタコネクタ(LPT1)
セントロニクス規格準拠のインターフェイスです(Dsub 25ピンメス)。プリンタなどパラレル通信を行う機器を接続します(ECP/SPP/EPP対応)。

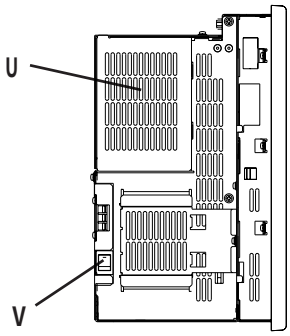
M: RASコネクタ(RAS)
DIN、DOUT、ウォッチドッグ、リモートリセットのインターフェイスです(Dsub25ピン オス)。

N: IDE I/Fカバー
CD-ROMドライブユニット（PL-DK200）を装着する場合、このカバーを取り外します。

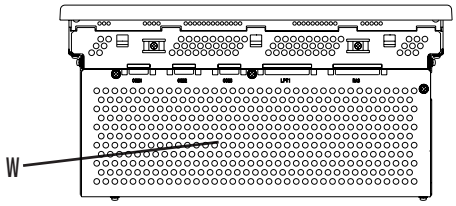
O:電源入力用端子台
AC100V/240Vの電源ケーブルを接続します。



PL-6920(4 スロットタイプ)右側面図



PL-6920(4 スロットタイプ)左側面図



PL-6920(4 スロットタイプ)上面図

P: マウスコネクタ(MOUSE)
PS/2 タイプのマウスを接続します。

Q：LAN コネクタ(10/100 BASE-T)
IEEE802.3 準拠のイーサネットインターフェイスです。(10 Base-T/100 Base-TX 自動切替)

R: 側面取り付け FDD スロット
FDD ユニット（PL-FD200）を装着するスロットです。

S: 拡張スロット

T: HDD/CF カードユニット拡張スロット
HDDユニットまたはCFカードユニットを拡張するスロットです。

U: ハーフカバー
オプション品のDIMモジュールや各種拡張ボードを取り付ける場合、このカバーを取り外します。

V: 電源スイッチ
PL の電源 ON/OFF を行います。

W: リアメンテナンスハッチ
オプション品のDIMモジュールや各種拡張ボードを取り付ける場合、このカバーを取り外します。

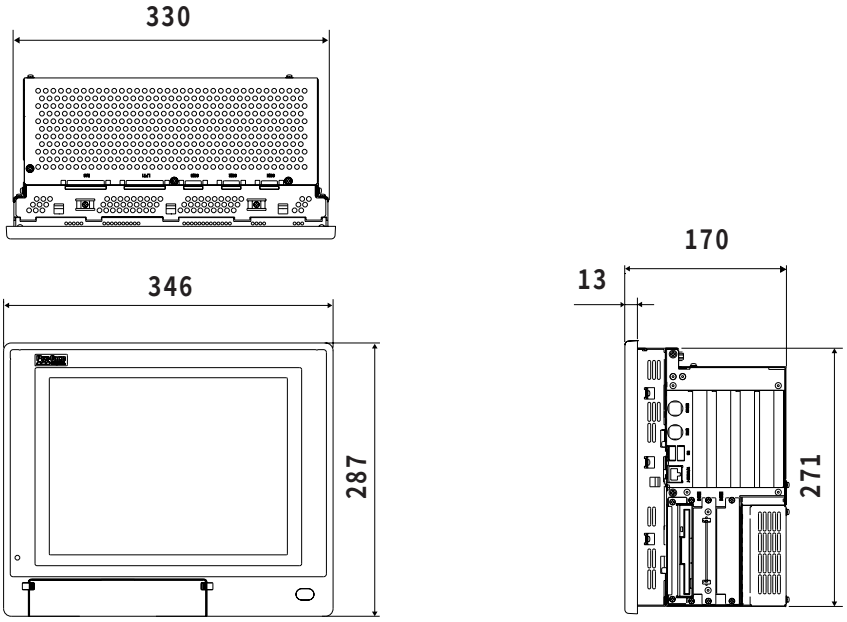
2.5 外観図と各部寸法図

PL-6920 シリーズと PL-7920 シリーズの外観図と各部の寸法図を示します。

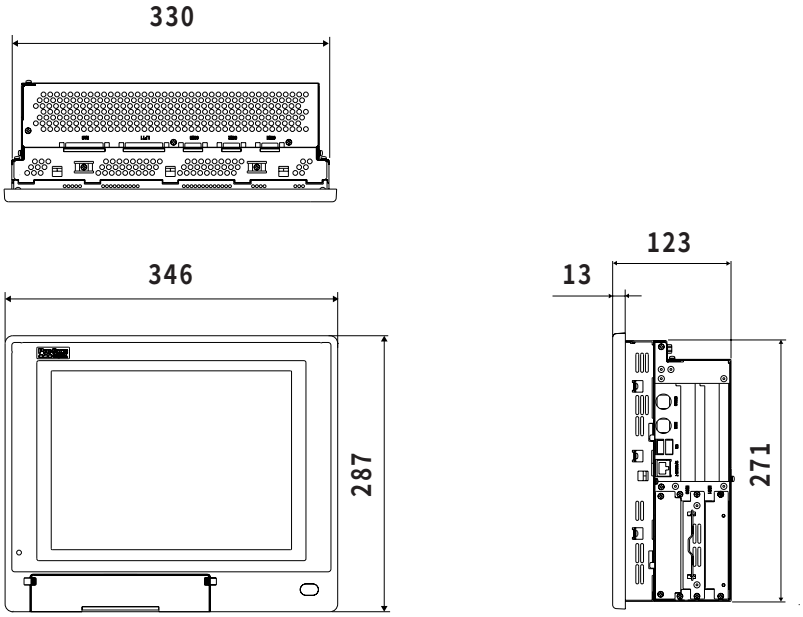
2.5.1 外観図

単位：mm
（突出部を除く）

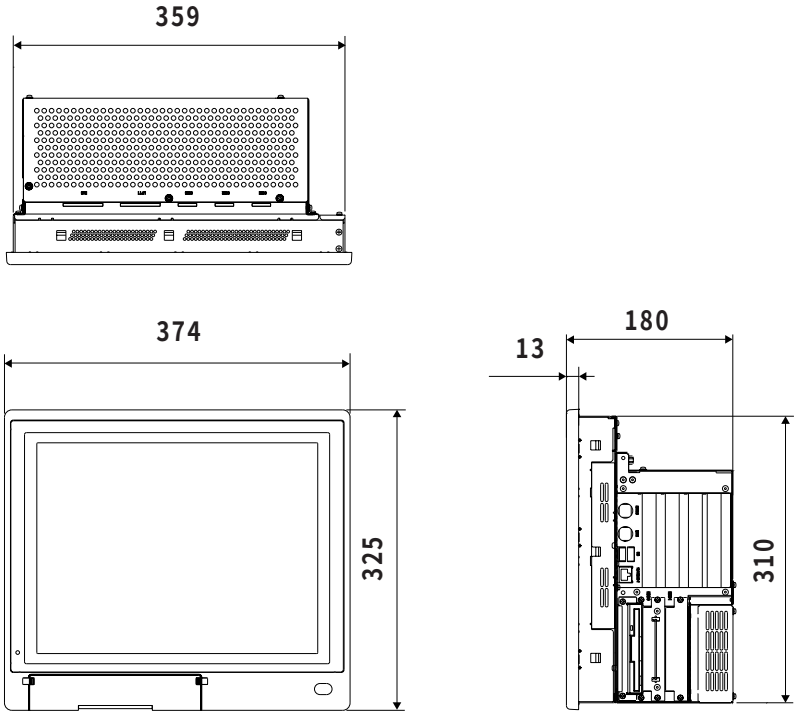
■ PL-6920



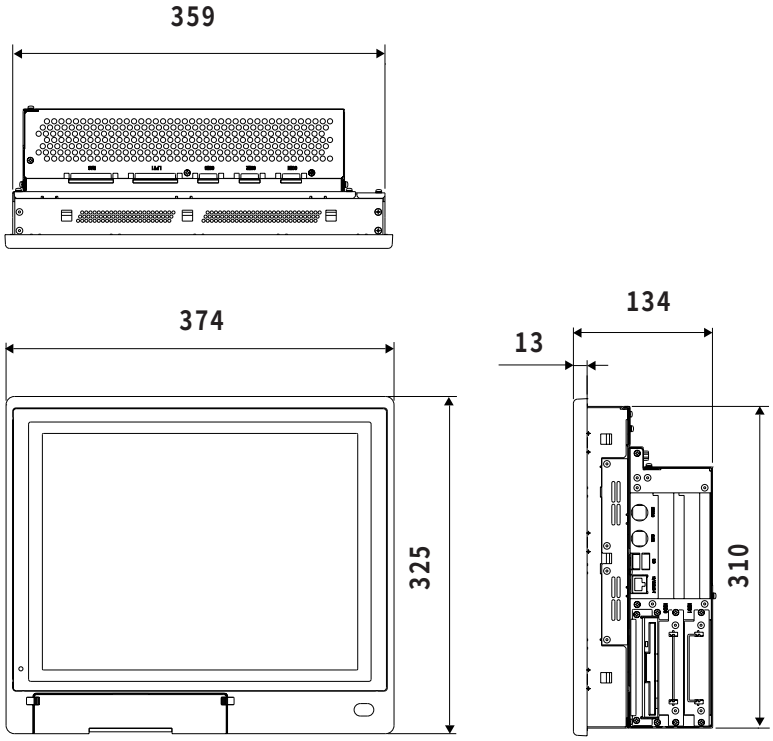
■ PL-6921



■ PL-7920

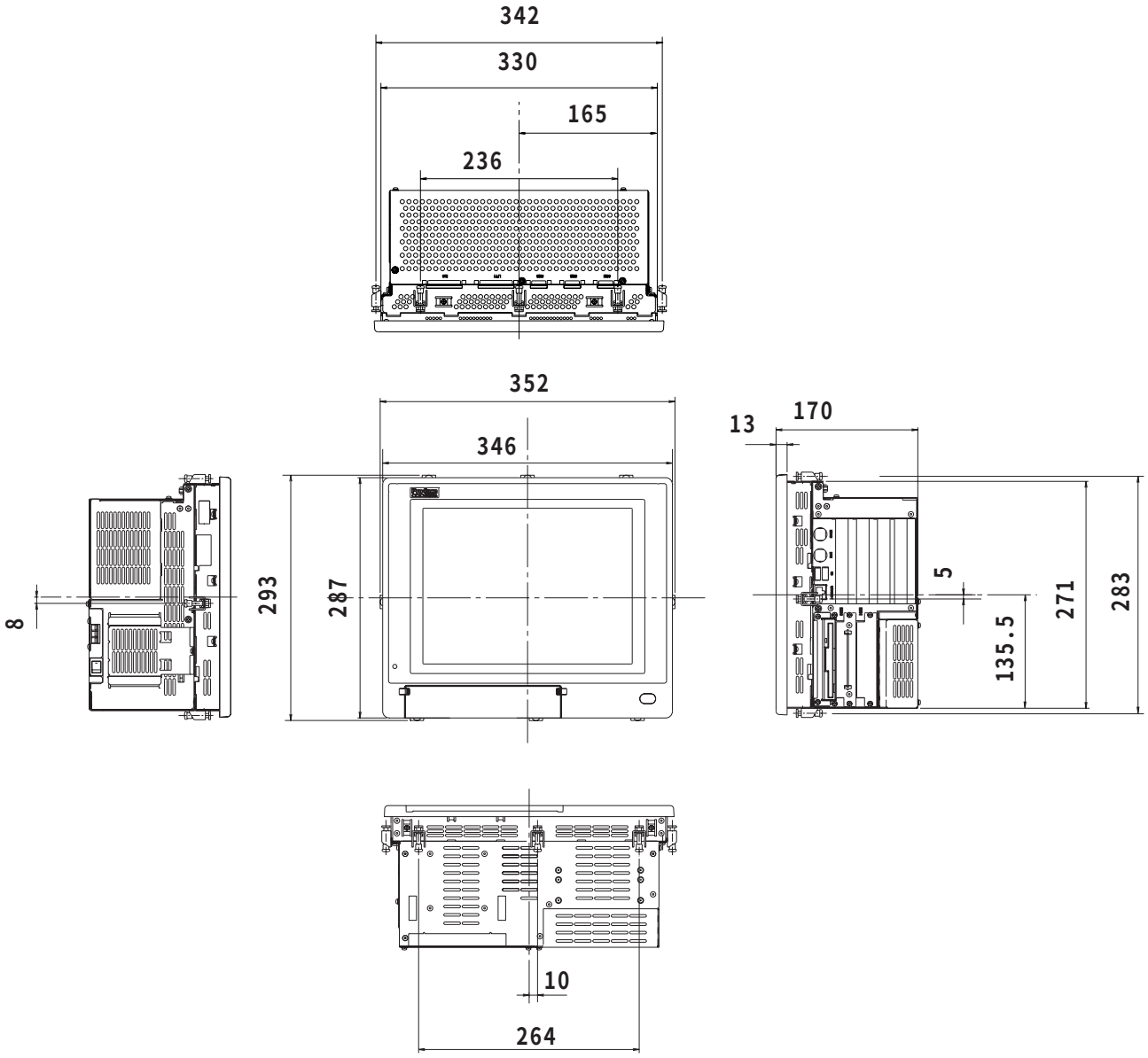


■ PL-7921

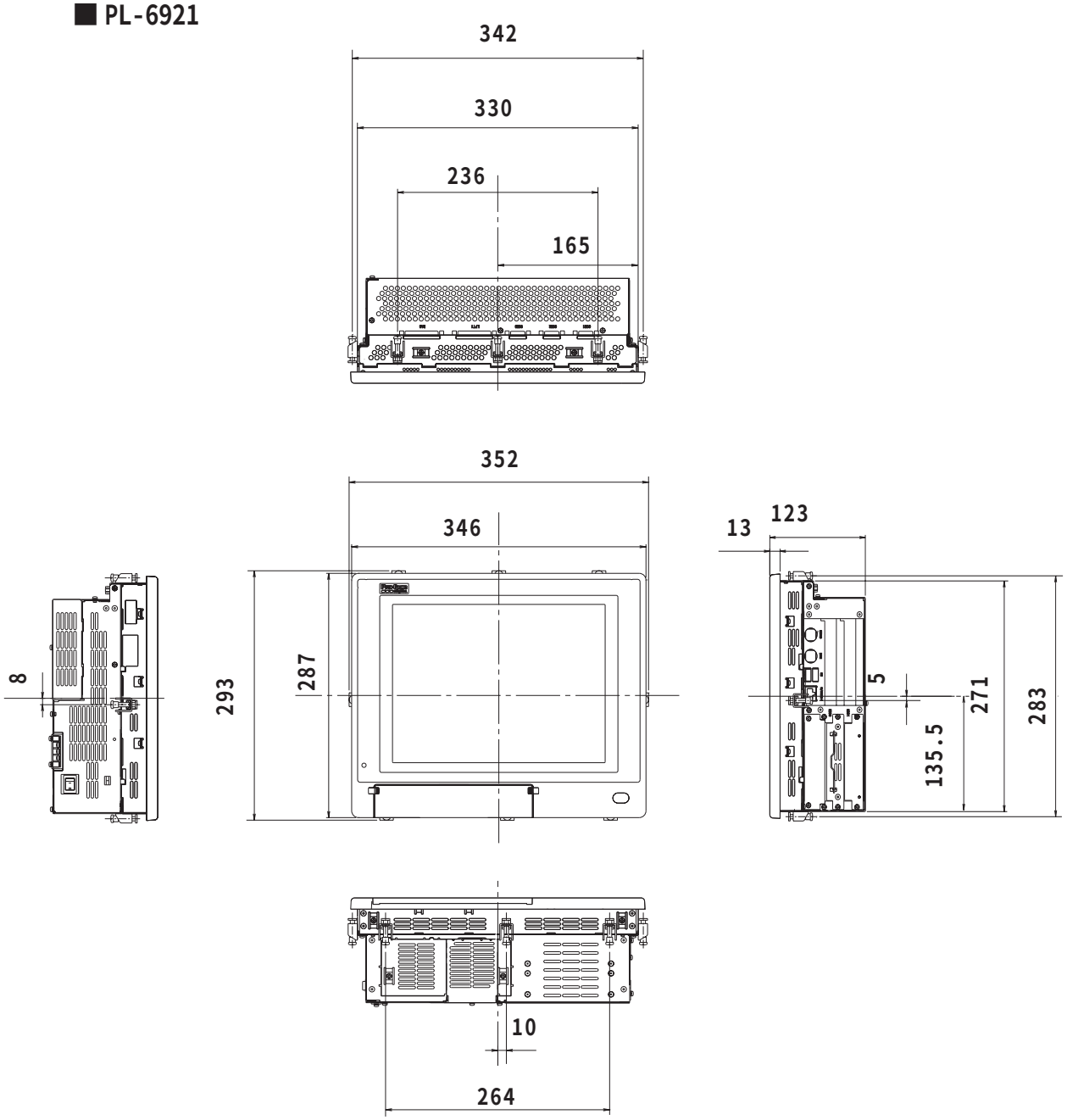


2.5.2 取り付け金具付き外観図

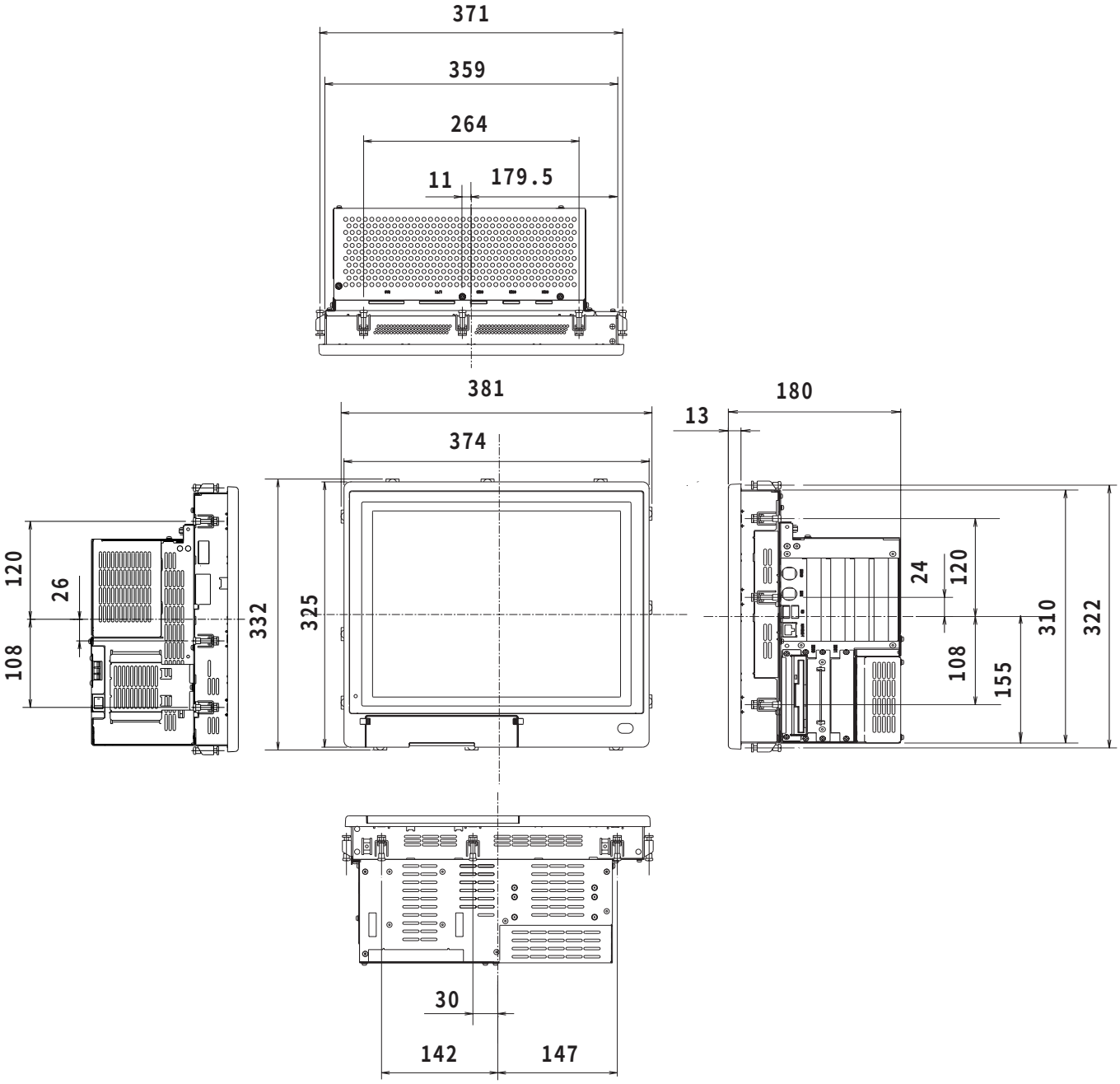
■ PL-6920



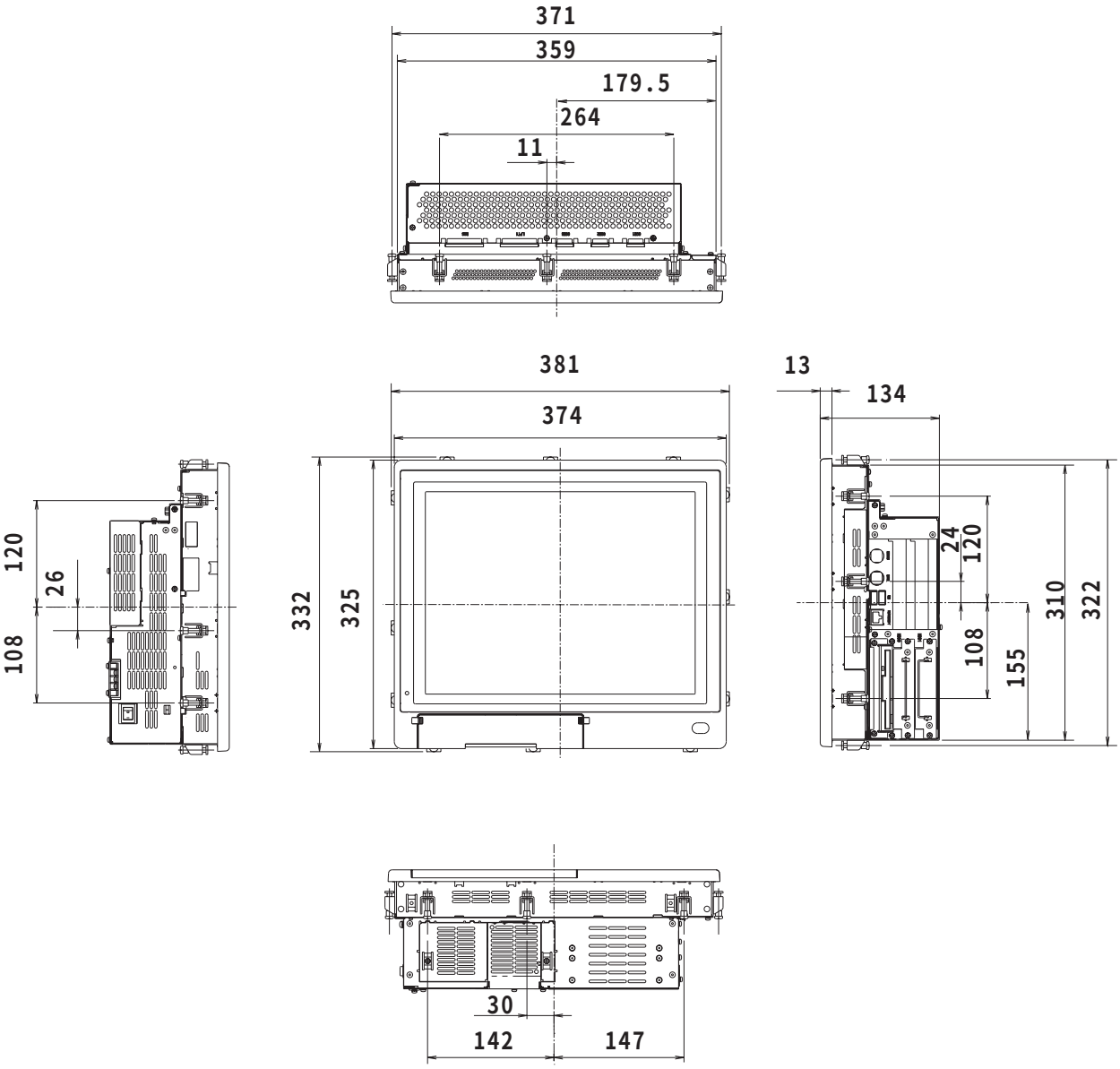
■ PL-6921



■ PL-7920


371
359
264
11
179.5
381
374
180
13
120
26
108
332
325
24
120
108
155
310
322
30
142
147

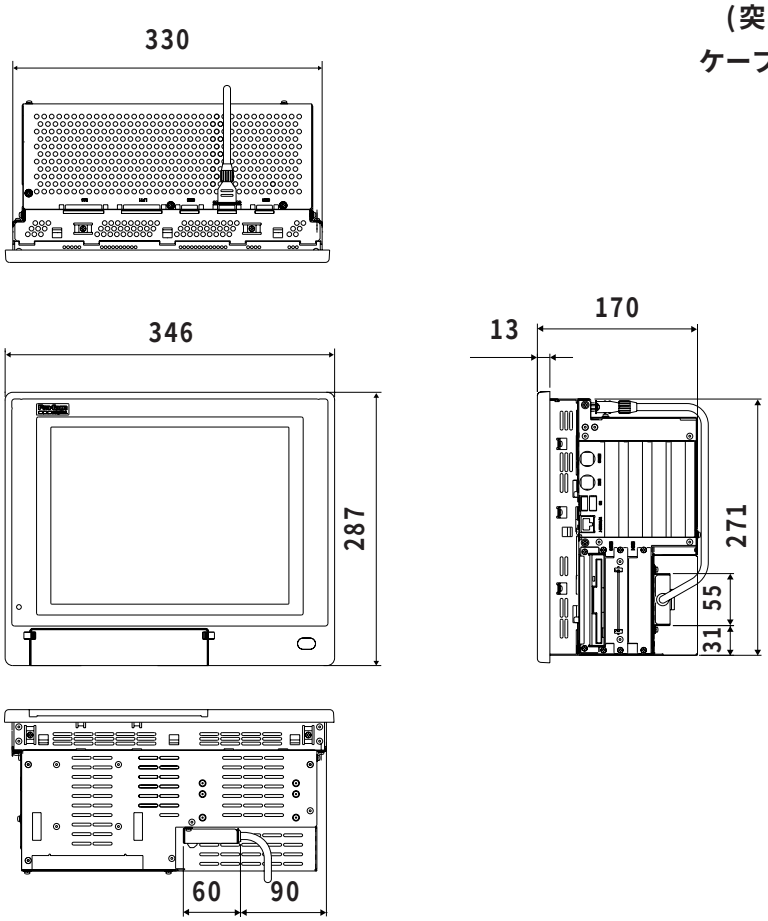
■ PL-7921


371
359
179.5
264
11
381
374
332
325
13
134
24
120
310
322
108
155
120
26
108
30
142
147

2.5.3 RS-232C/RS-485 変換ユニット(PL-RC500)取り付け時の外観図

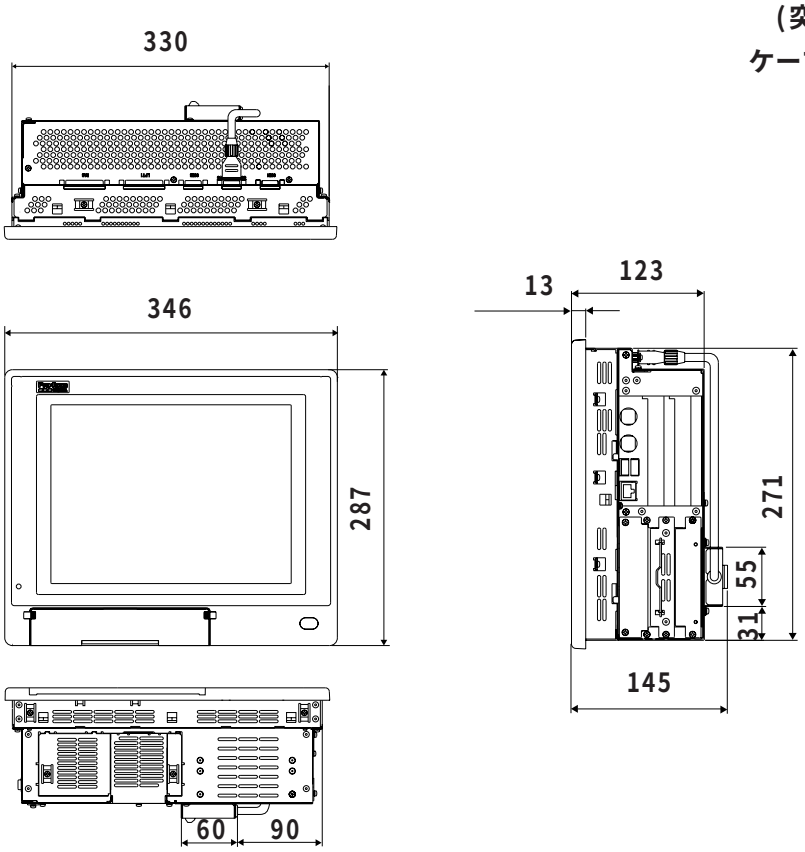
■ PL-6920

単位:mm
(突出部および
ケーブル部を除く)

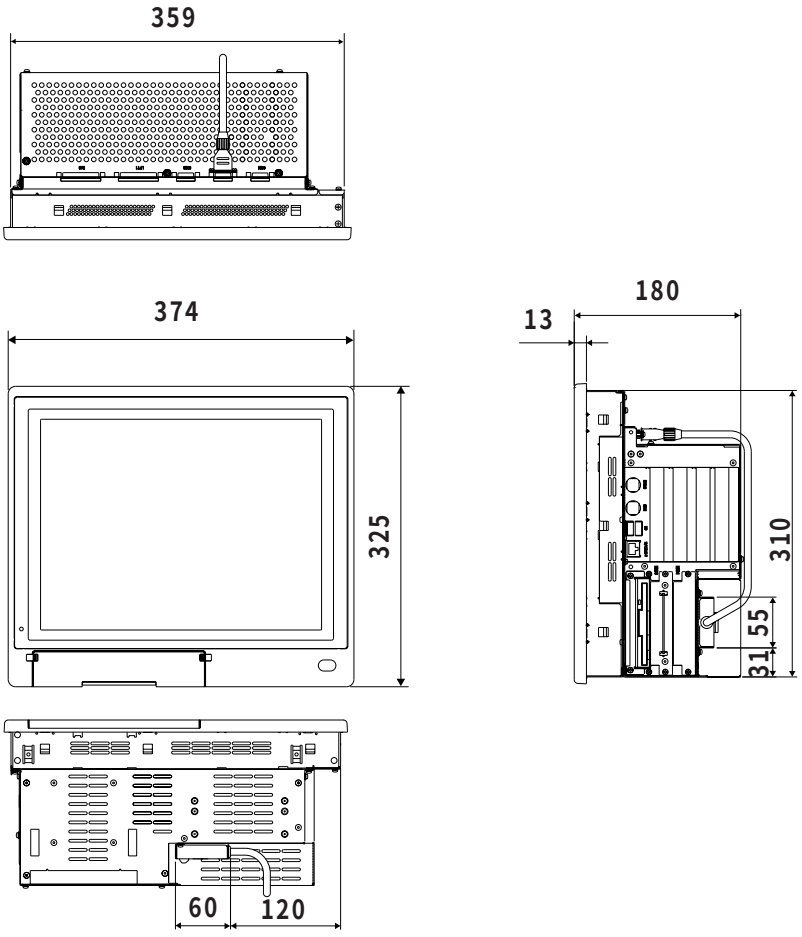


■ PL-6921

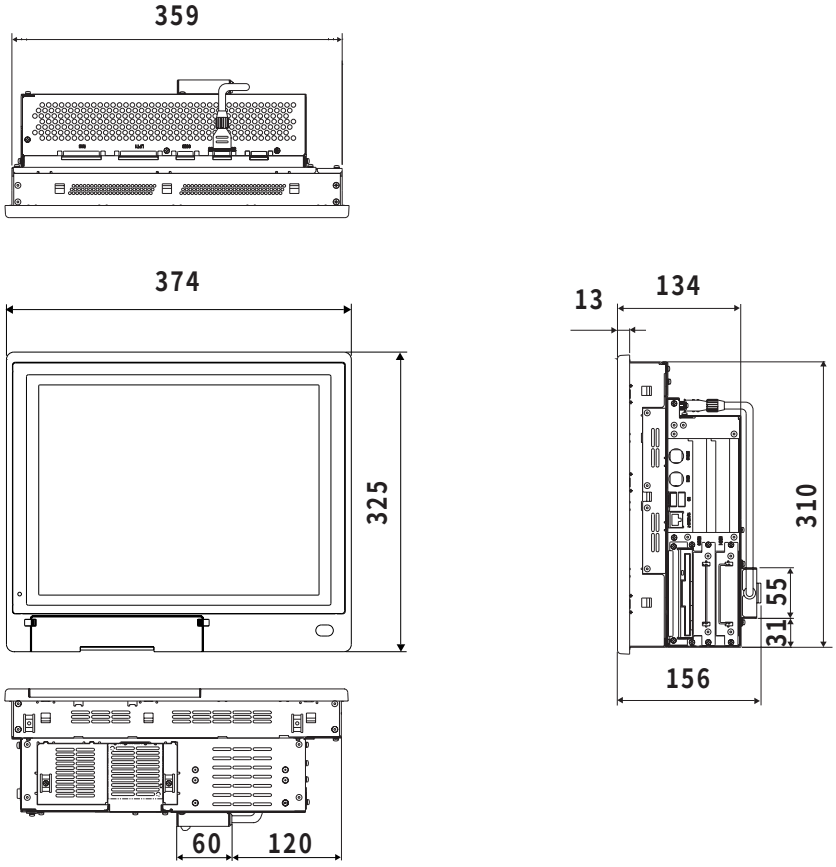
単位:mm
(突出部および
ケーブル部を除く)



■ PL-7920



■ PL-7921

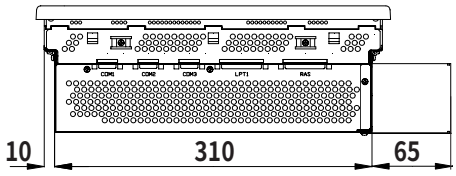


2.5.4 フルサイズボードカバー取り付け時の外観図

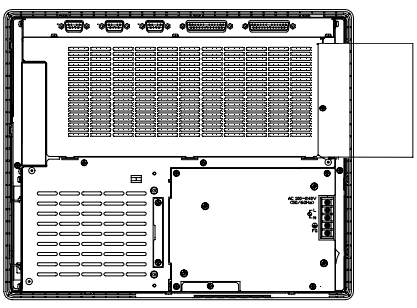
■ PL-6920 シリーズ

単位:mm

PL-6921 をモデルにしています。



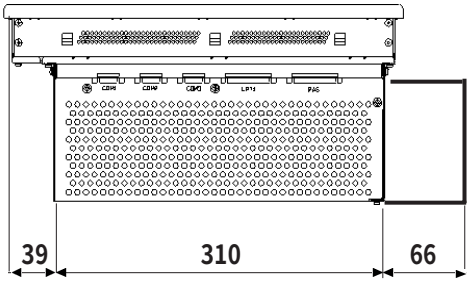
上面図



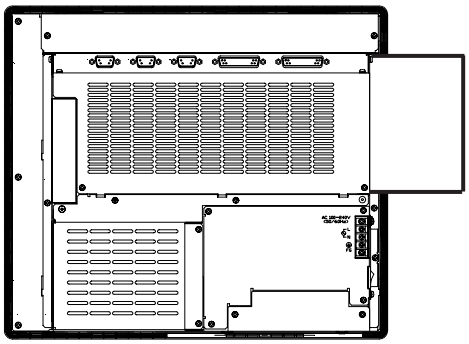
背面図

■ PL-7920 シリーズ

PL-7920 をモデルにしています。



上面図



背面図

重要

- **フルサイズ拡張ボードおよびフルサイズボードカバーを使用する場合は、先に PL 本体をパネルに取り付けてから装着してください。フルサイズ拡張ボードおよびフルサイズボードカバーを先に取り付けると PL をパネルに取り付けることができません。**
- **フルサイズボードカバーを使用する場合は、装着するボードの寸法や形状によって耐振動等の環境仕様が異なります。**

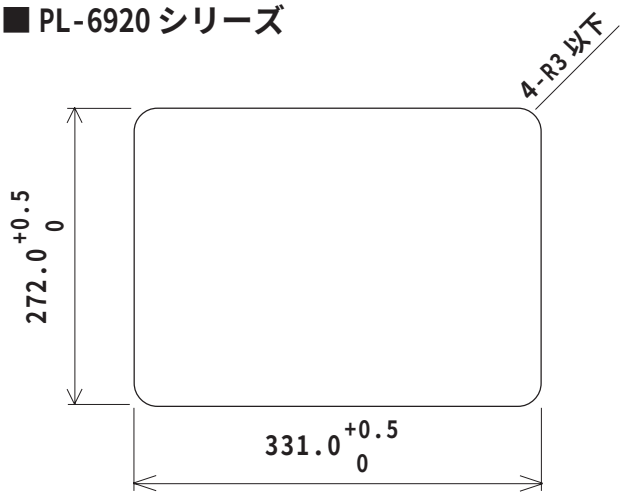
MEMO

- フルサイズボードカバーには 4 スロットタイプ（PL-6920/7920）用と 2 スロットタイプ（PL-6921/7921）用があります。
  4 スロットタイプ用：PL-FC210
  2 スロットタイプ用：PL-FC200
  参照→1.3 オプション機器一覧

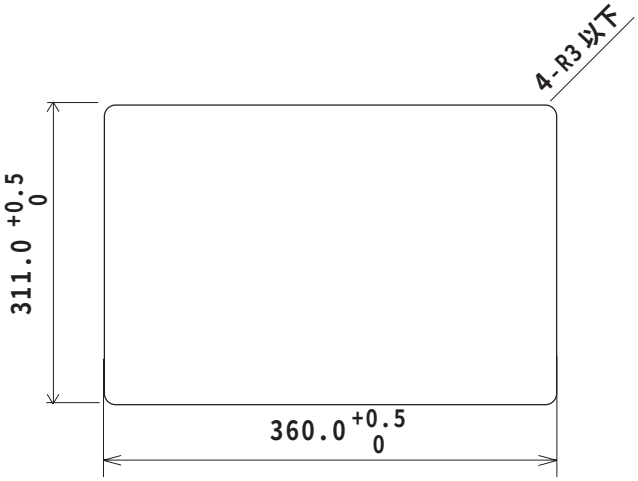
2.5.5 パネルカット寸法

単位：mm

■ PL-6920 シリーズ



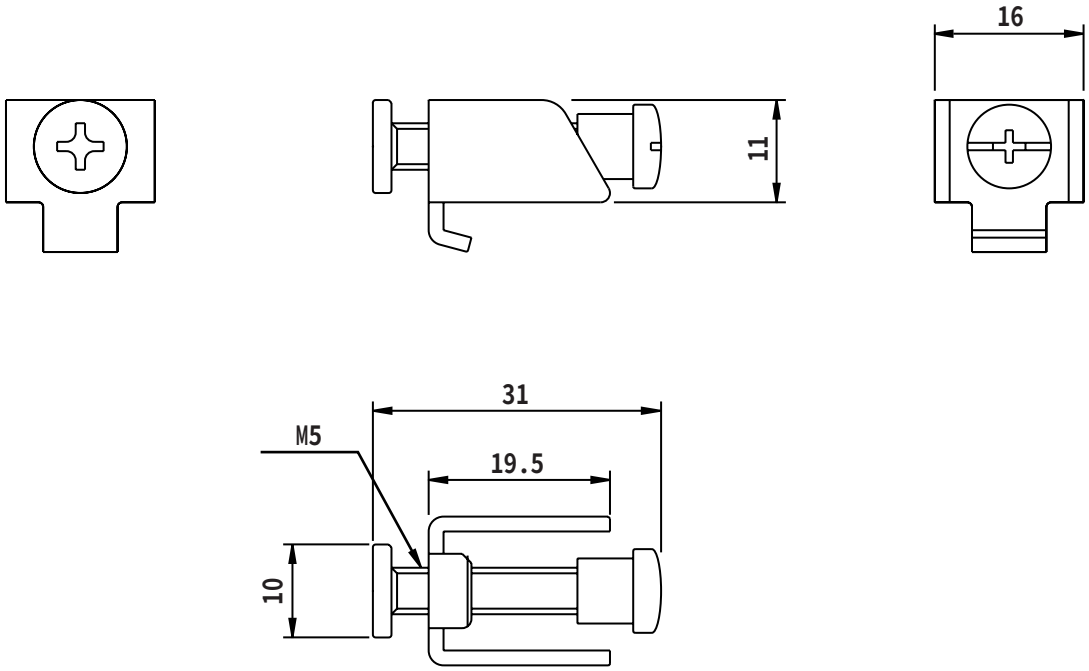
■ PL-7920 シリーズ



重要
- パネル厚範囲は 1.6㎜ ～ 10.0㎜ です。
- パネルの形状によっては、補強等の対策が必要です。特に、振動が発生する場所、扉等の稼働場所に取り付ける場合は、PLの質量を十分に考慮してパネルを設計してください。
  参照→ 2.1.3 外観仕様
- 防滴効果を得るため、取り付け部は傷がなく良好な平面にしてください。
- 取り付け公差は必ず守ってください。脱落の恐れがあります。

2.5.6 取り付け金具寸法図

単位:mm



MEMO

このページは、空白です。
ご自由にお使いください。

第 3 章 ユニット・拡張ボードの組み込み

1. ユニット・拡張ボードの取り付け

PLでは、(株)デジタルがオプションとして用意する各種ユニットや拡張ボード、市販のISA(AT)バス互換ボードが使用できます。

この章では、ユニットや拡張ボードをPLに組み込んで使用する方法について説明します。

3.1 ユニット・拡張ボードの取り付け

ここでは、DIMモジュール（PL-EM500/PL-EM128）、FDDユニット（PL-FD200/PL-FD210）、HDDユニット（PL-HD220/PL-HDX920-W95/PL-HDX920-NT40/PL-HDX920-W2K/PL-HDX920-W2K/ML/PL-HDX920-WXP）、拡張ボード、CD-ROMドライブユニット（PL-DK200）、および電源ファンユニットの取り付け/取り外しについて説明します。

その他のオプションユニットについては、各オプションユニットの「取扱説明書」を参照してください。

⚠ 警 告

■ ユニット・拡張ボードの取り付け時は、電源ケーブルを取り外し、必ずPLに電源が供給されていないことを確認してから行ってください。感電のおそれがあります。

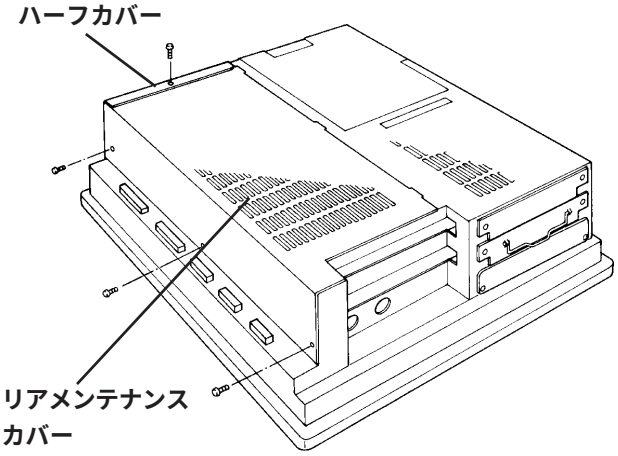
重要

- **ネジの取り外し、取り付けにはドライバを使用してください。**
  ネジは強くしめつけすぎると破損するおそれがありますので、ご注意ください。
- **PL本体内へのネジの脱落に注意し作業を行ってください。**

3.1.1 リアメンテナンスカバーの取り外し

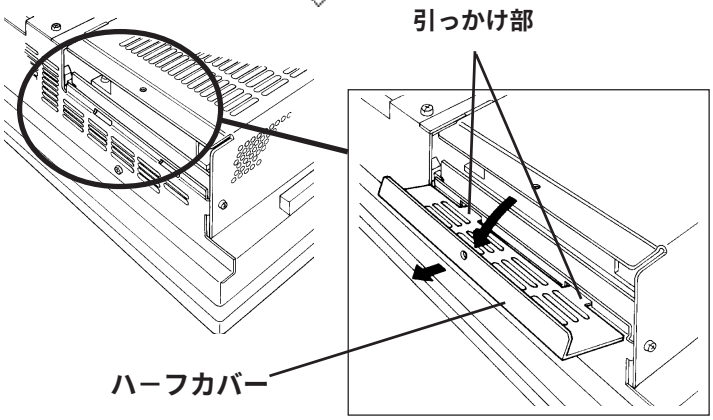
重要 ・ **リアメンテナンスカバーはアルミ製です。変形しやすいので、取り扱いには十分に注意してください。**

■ PL-6921/PL-7921（2 スロットタイプ）の場合

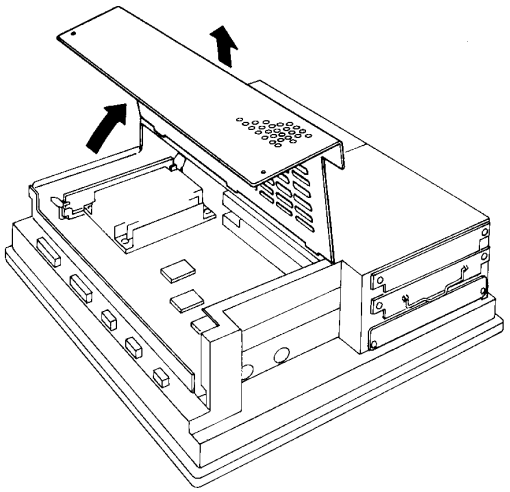


① PLのハーフカバーとリアメンテナンスカバーのネジ(4ヵ所)を外します。

重要 **・ハーフカバーを外してからリアメンテナンスカバーを外してください。**

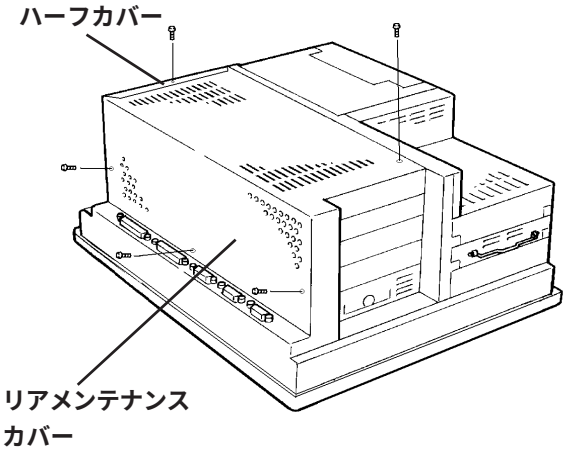


② ハーフカバーの引っかけ部を引き上げ、ハーフカバーを取り外します。



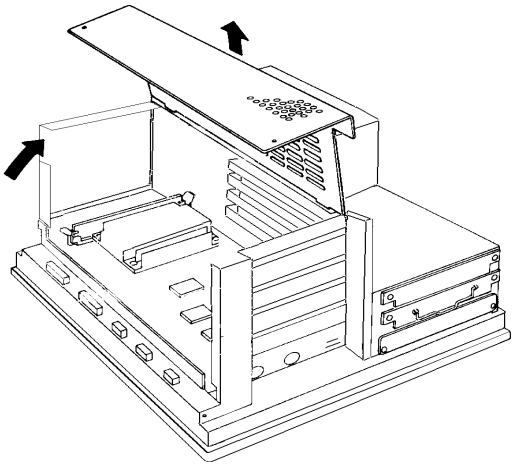
③ リアメンテナンスカバーを取り外します。

■ PL-6920/PL-7920（4 スロットタイプ）の場合

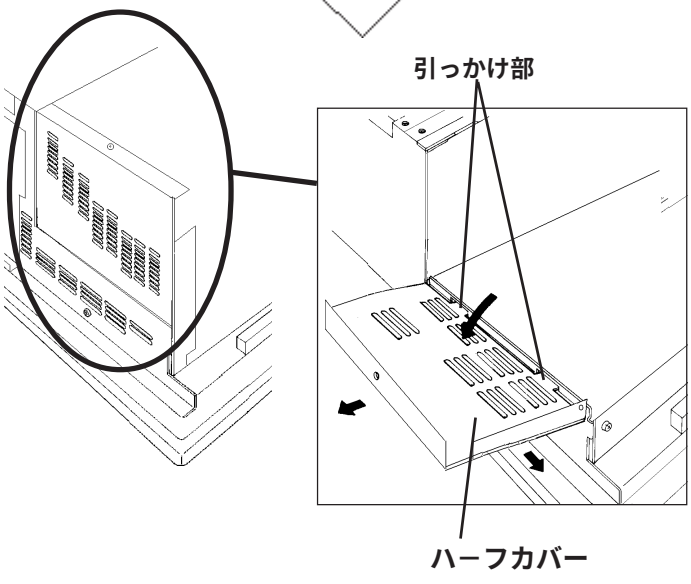


① PLのハーフカバーとリアメンテナンスカバーのネジ(5ヵ所)を外します。

重要 **・ リアメンテナンスカバーを外してからハーフカバーを外してください。**



② リアメンテナンスカバーを取り外します。



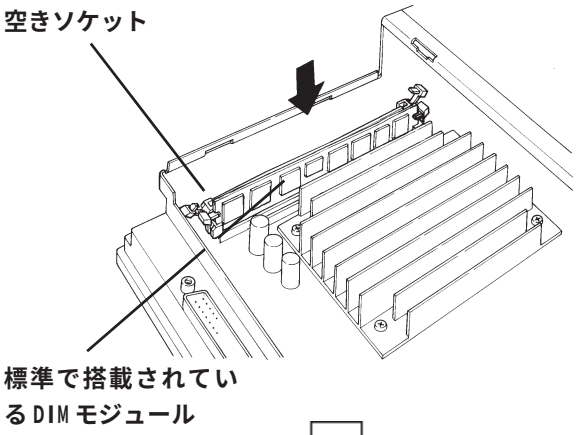
③ ハーフカバーの引っかけ部を引き上げ、ハーフカバーを取り外します。

3.1.2 DIMモジュール(PL-EM500/PL-EM128/PL-EM256)の取り付け

重要

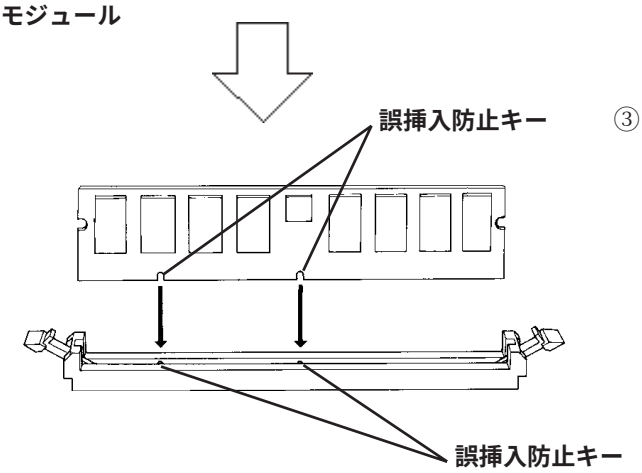
- **DIMモジュールのソケットは非常に壊れやすい部品ですので、取り扱いには十分ご注意ください。**
- **標準で搭載されているDIM モジュールは、挿入するソケットの位置を変更しないでください。**

PLには、2つのDIMモジュールソケットがあり、標準ではDIMモジュールが1枚搭載されています。空きソケットにDIMモジュールを取り付けることで、メインメモリを拡張することができます。以下の順序で取り付けてください。

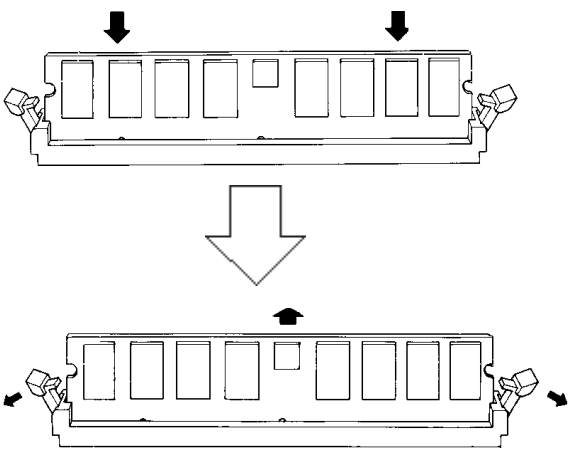


① リアメンテナンスカバーとハーフカバーを取り外します。**参照→ 3.1.1 リアメンテナンスカバーの取り外し**

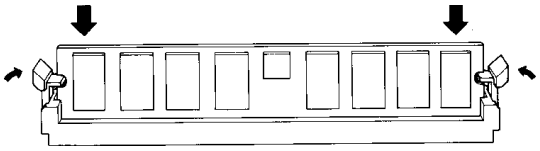
② 空きソケットに拡張用DIMモジュールを取り付けます。



③ 誤挿入防止キーの位置を合わせます。



④ DIMモジュールをDIMモジュールソケットの溝に沿って差し込みます。

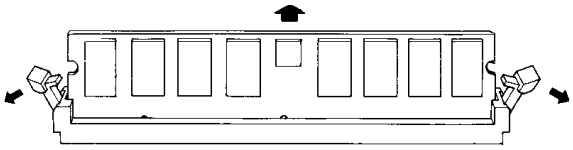


⑤ DIMモジュールを両側のツメがロックするまで押し下げます。

⑥ リアメンテナンスカバーとハーフカバーを元に戻しネジを止めます。

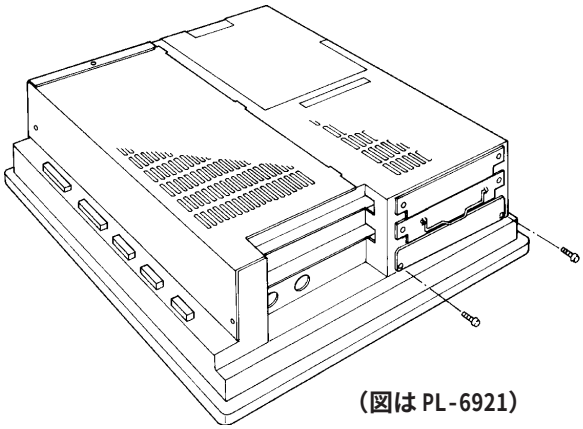
<取り外し方法>

DIMモジュールソケットのツメを矢印の方向に開き、DIMモジュールを外します。



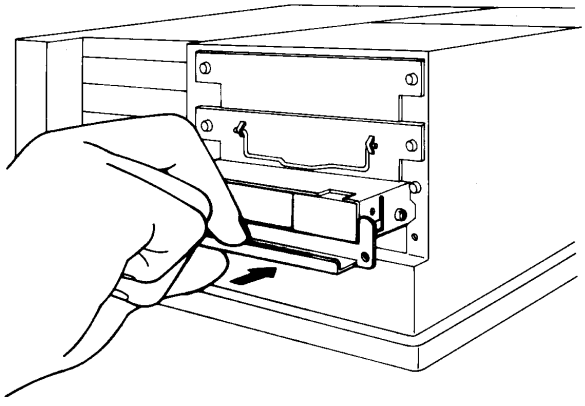
3.1.3 FDDユニット(PL-FD200)の取り付け

MEMO・ PL-FD200 と PL-FD210 を同時に使用することはできません。



（図は PL-6921）

① FDDユニットの挿入口のブランクパネルのネジ(2ヵ所)を外し、ブランクパネルを取り外します。



② FDDユニットをガイドレールに沿うように挿入し、コネクタが完全に接続されるよう差し込みます。

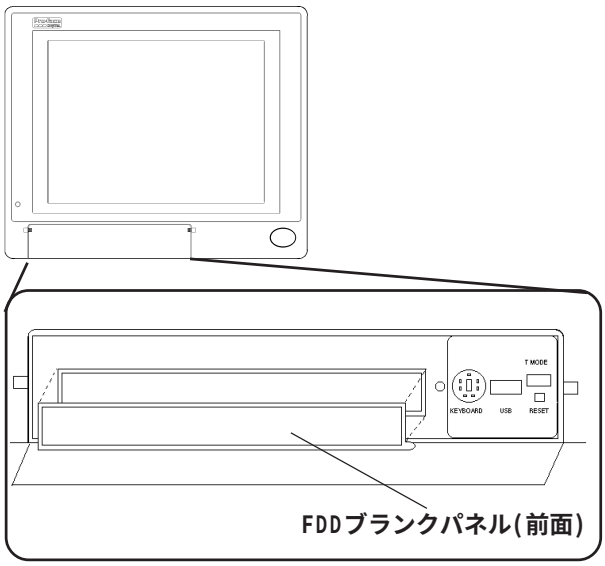
③ ネジ（2ヵ所）で固定します。

3.1.4 FDDユニット(PL-FD210)の取り付け

MEMO

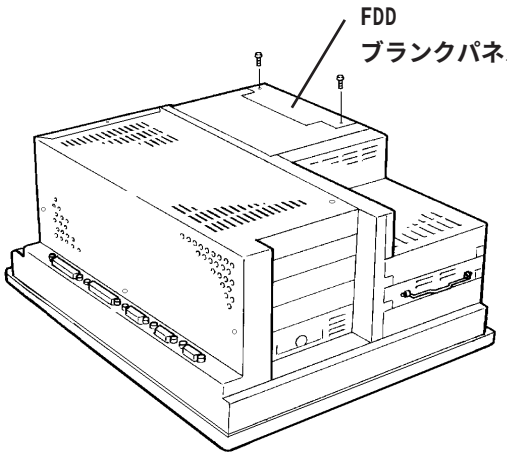
・ PL-FD210はPL-6920/PL-7920（4スロットタイプ）のみ対応しています。PL-6921/PL-7921（2スロットタイプ）には使用できません。

・ PL-FD200 と PL-FD210 を同時に使用することはできません。



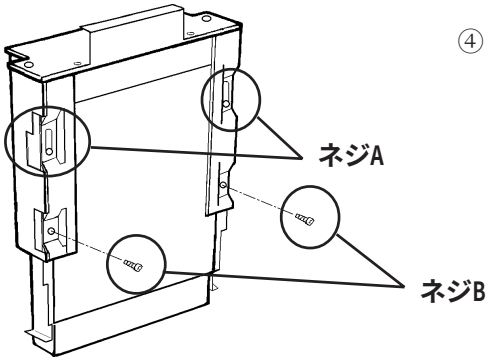
① フロントメンテナンスハッチ(カバー)を開き、FDDブランクパネルを取り外します。

② フロントメンテナンスハッチ(カバー)を閉じます。

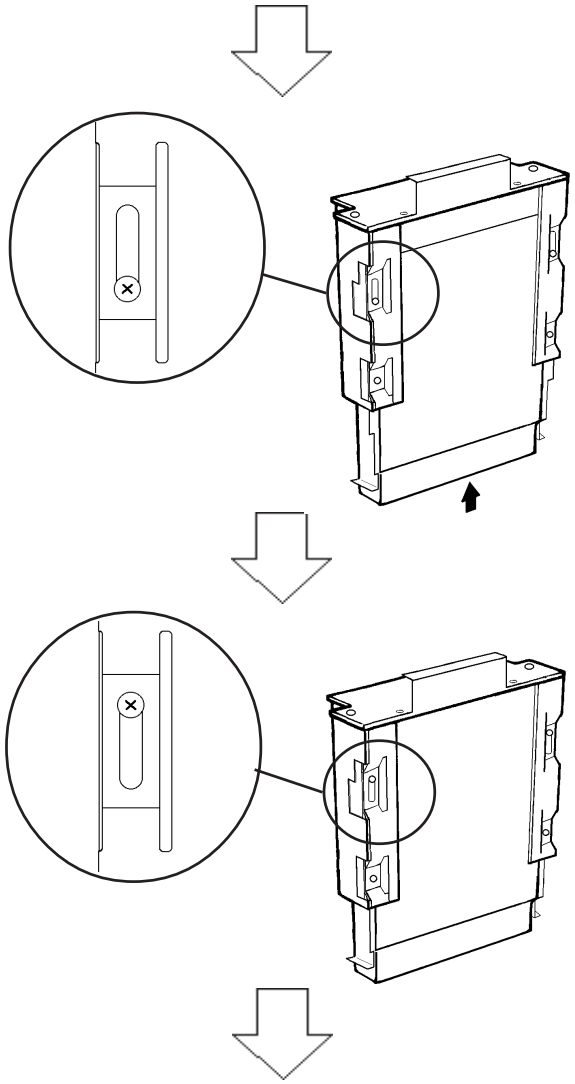


③ FDDユニットの挿入口のブランクパネルのネジ(2ヵ所)を外し、ブランクパネルを取り外します。

◆この手順は、PL-6920シリーズのみ必要

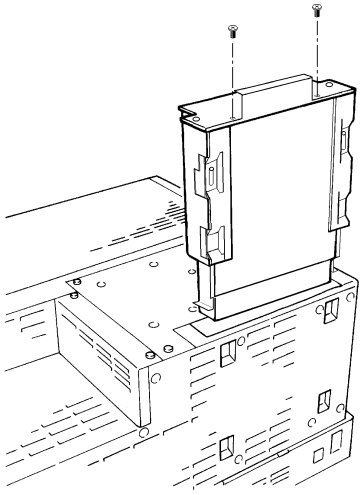


④ 左図に示すFDDユニットのネジA（2カ所）をゆるめ、下部のネジB（2カ所）を外します。



◆この手順は、PL-6920 シリーズのみ必要

⑤ FDDユニット内部を矢印の方向へスライドさせ、ネジAおよびネジBを締めて固定します。



⑥ FDDユニットをガイドレールに沿うように挿入し、コネクタが完全に接続されるよう差し込みます。

⑦ PL-FD210 に付属のネジ（2 カ所）で固定します。

重要

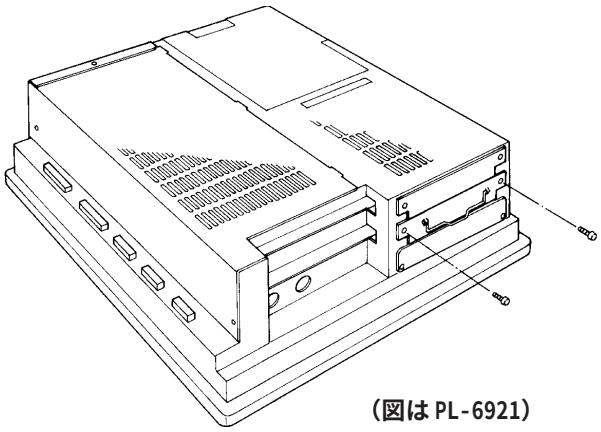
- **PL-FD210をPL本体に組み込む際には、PL-FD210をPL本体にゆっくり押し込みながら、正しく装着されているかを確認してください。**

⑧ 最後に、手順③で取り外したブランクパネルを取り付けます。

3.1.5 HDDユニット(PL-HD220/PL-HDX920-W95/PL-HDX920-NT40/PL-HDX920-W2K(/ML)/PL-HDX920-WXP)の取り付け / 取り外し

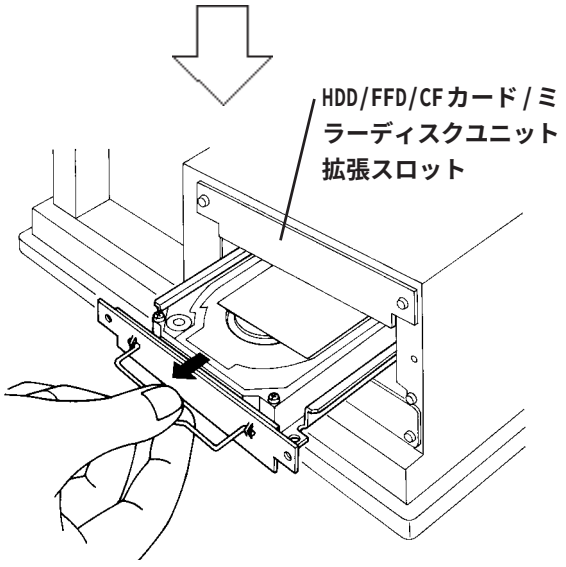
MEMO
- FFDユニット(PL-FF210)およびCFカードユニット(PL-CF200)の取り付け / 取り外しもHDDユニットと同様に行います。
- HDD ユニット、FFD ユニット、ミラーディスクユニット、CFカードユニットおよびCD-ROM ドライブユニットは使用する組み合わせに制限があります。**参照→1.3 オプション機器一覧**

重要 **・ HDDユニットは精密機器ですので、衝撃を与えないでください。**



(図は PL-6921)

① HDD ユニットのネジ(2ヵ所)を外します。



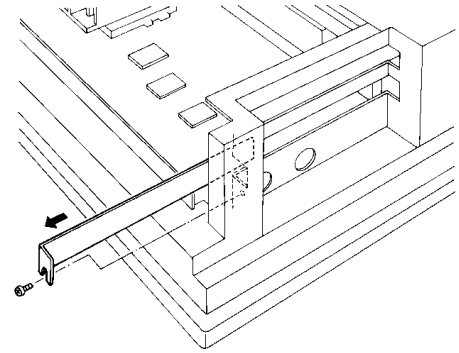
② HDD ユニットの取手を持ち、衝撃を与えないようにゆっくりとPL本体から引き出します。

③ 新たに組み込むHDDユニットをガイドレールに沿うようにPL本体に挿入し、コネクタが完全に接続されるよう差し込みます。

④ ネジ(2ヵ所)で固定します。

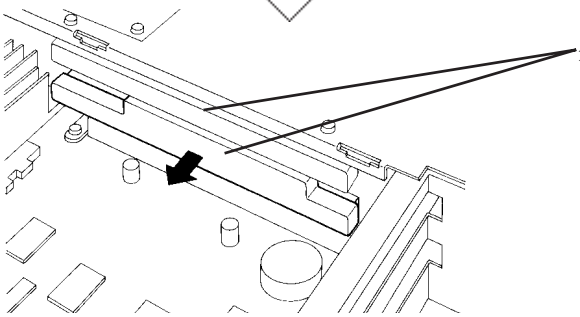
MEMO
- HDD/FFD/CFカードユニット拡張スロットへの脱着も、同様の手順で行います。

3.1.6 拡張ボードの取り付け

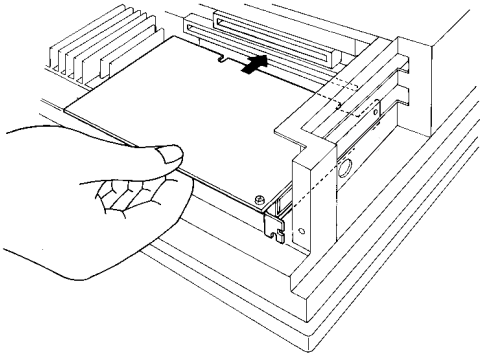


① PLのリアメンテナンスカバーを取り外します。**参照→ 3.1.1 リアメンテナンスカバーの取り外し**

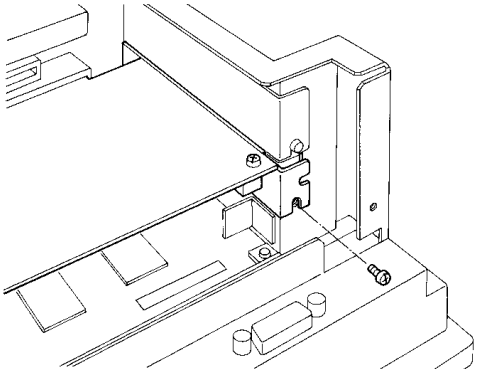
② ブランクパネルのネジ(1ヵ所)を外し、ブランクパネルを取り外します。



③ ダスターカバーを取り外します。



④ 拡張ボードを拡張スロットに差し込みます。

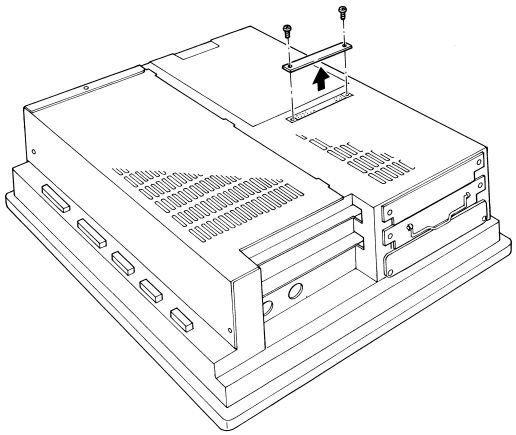


⑤ 拡張ボードの板金部を拡張ボード付属のネジ(1ヵ所)で固定します。

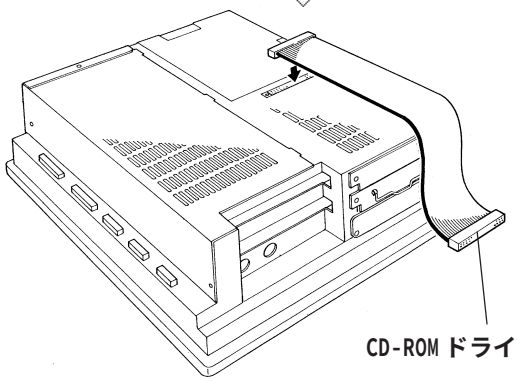
⑥ リアメンテナンスカバーとハーフカバーを元に戻しネジを止めます。

3.1.7 CD-ROM ドライブユニット(PL-DK200)の接続

■ PL-6921/PL-7921（2 スロットタイプ）の場合



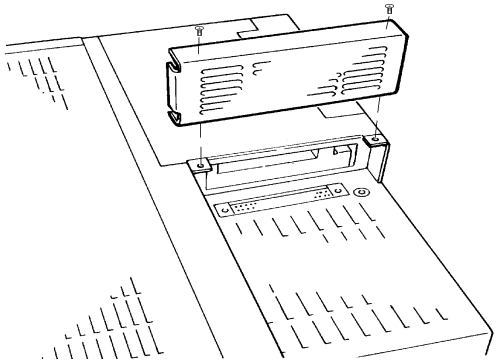
① IDE I/F カバーのネジ(2ヵ所)を外し、IDE I/F カバーを取り外します。



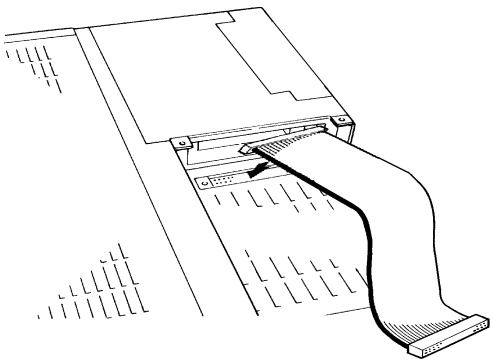
② CD-ROM ドライブユニットに付属のケーブル(PL-X920 シリーズ用)を IDE I/F に接続します。

重要 **• ケーブルがコネクタに正しく接続されていることを確認の上、電源を投入してください。**

■ PL-6920/PL-7920（4 スロットタイプ）の場合



① IDE I/F カバーのネジ(2ヵ所)を外し、IDE I/F カバーを取り外します。



② CD-ROM ドライブユニットに付属のケーブル(PL-X900 シリーズ用)を IDE I/F に接続します。

重要 **• ケーブルがコネクタに正しく接続されていることを確認の上、電源を投入してください。**

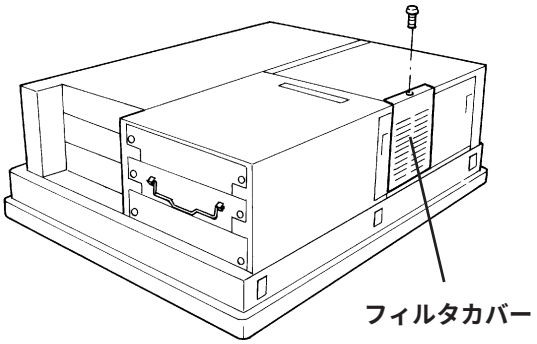
3.1.8 電源ファンユニットの取り外し

PLは本体底面部のファンユニットを取り外して使用することができます。

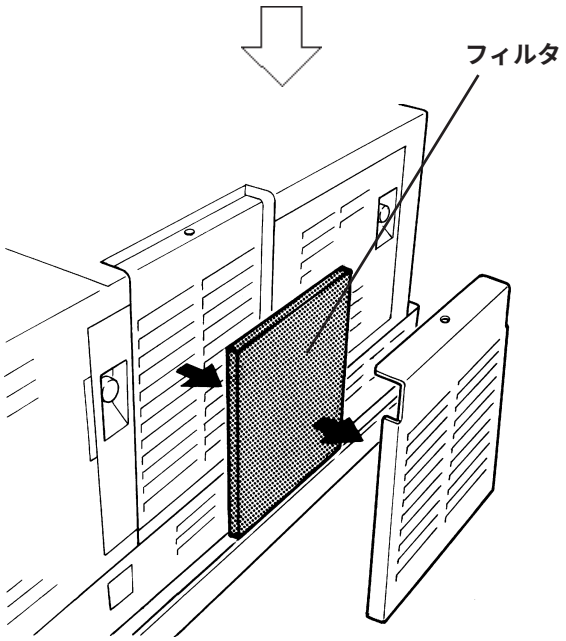
この場合、使用周囲温度はファン未使用時の温度となります。参照→ 2.1.2 環境仕様

重要 **• CPUが1GHzの機種(PL692*-T42/PL792*-T42)は電源ファンを取り外して使用することはできません。**

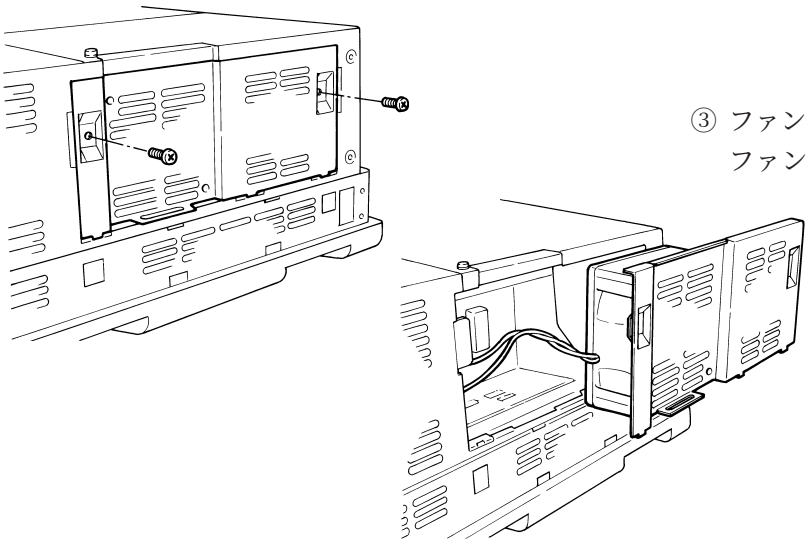
■ PL-6921/PL-7921（2 スロットタイプ）の場合



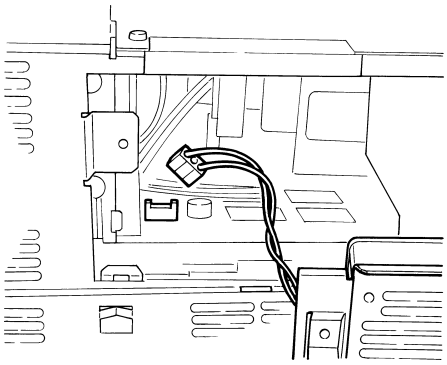
① フィルタカバーのネジ(1ヵ所)を取り外し、フィルタカバーを取り外します。



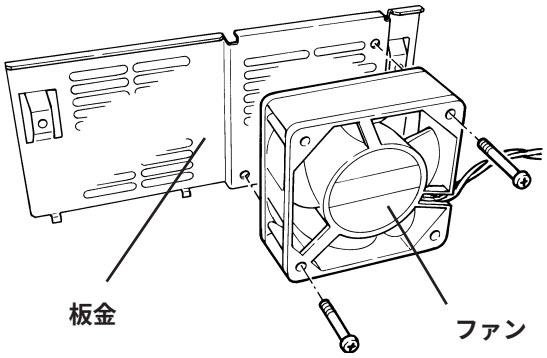
② フィルタを取り外します。



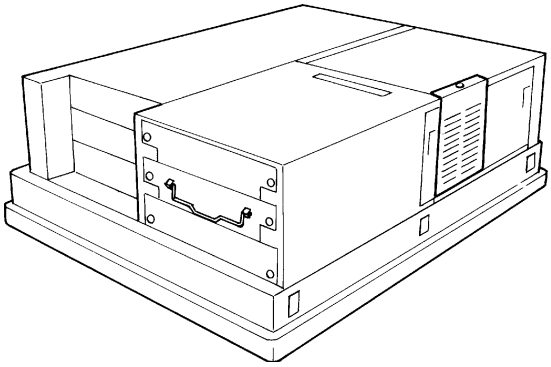
③ ファンユニットのネジ(2ヵ所)を外し、ファンユニットを取り外します。



④ ファンの電源ケーブルのコネクタを抜き取ります。

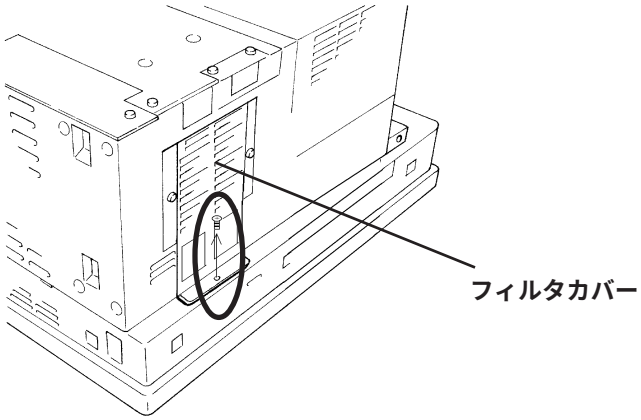


⑤ ファンを取り付けているネジ(2ヵ所)を外し、ファンユニットの板金からファンを取り外します。

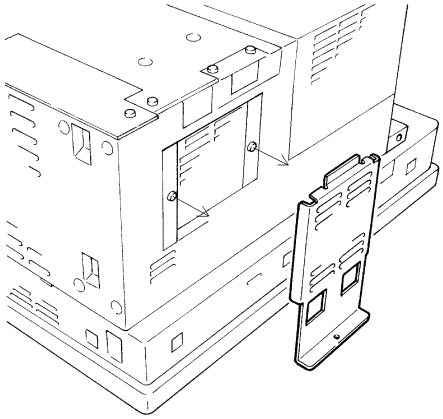


⑥ ファンの付いていた板金と、フィルタカバーをPL本体に装着します。

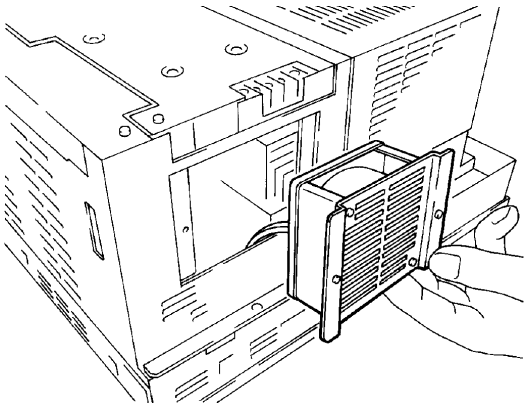
■ PL-6920/PL-7920（4 スロットタイプ）の場合



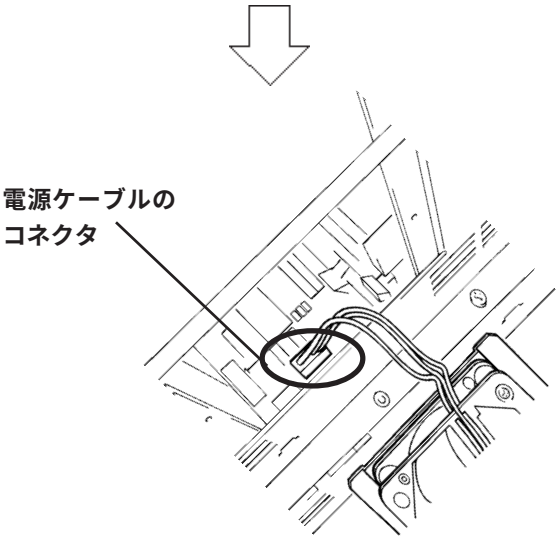
① フィルタカバーのネジ(1ヵ所)を取り外し、フィルタカバーとフィルタを取り外します。



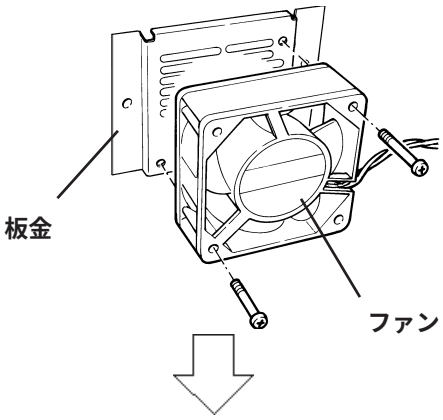
② ファンユニットのネジ(2ヵ所)を外します。



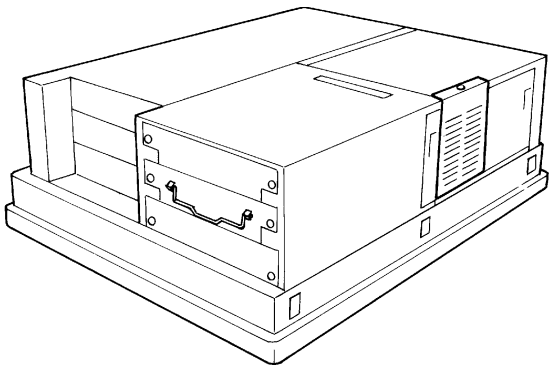
③ PL本体からファンユニットを取り外します。



④ ファンの電源ケーブルのコネクタを抜き取ります。



⑤ ファンを取り付けているネジ(2ヵ所)を外し、ファンユニットの板金からファンを取り外します。



⑥ ファンの付いていた板金と、フィルタカバーをPL本体に装着します。

第 4 章
設置と配線

1.PL の設置
2. 配線について

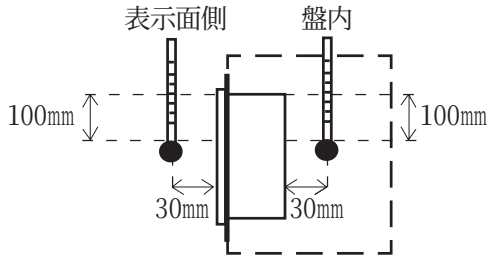
PLの取り付け方法と配線方法について説明します。

4.1 PL の設置

4.1.1 PL 設置上の注意

■使用周囲温度について

- PL は垂直取り付けで自然冷却ではなく、空冷ファンでの冷却を基本にしています。
- 故障の原因になりますので、使用周囲温度は下表の範囲内で使用してください。使用周囲温度の確認は下図の位置で行ってください。(使用周囲温度とは、盤内と表示面側の両方です。)



＜使用周囲温度範囲＞

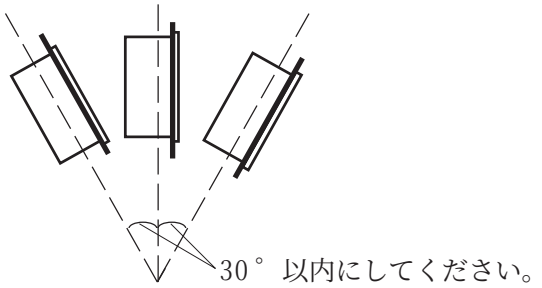
使用周囲温度	PL692*-T41 (CPU:700MHz)	ファン使用		5℃～50℃(HDD使用時)
		ファン未使用※1		5℃～40℃(HDD使用時)
	PL792*-T41 (CPU:700MHz)	盤内	ファン使用	5℃～50℃(HDD使用時)
			ファン未使用※1	5℃～40℃(HDD使用時)
		表示面側	5℃～40℃	
	PL692*-T42 (CPU:1GHz)	ファン使用		5℃～45℃(HDD使用時)
		ファン未使用※1		使用不可
	PL792*-T42 (CPU:1GHz)	盤内	ファン使用	5℃～45℃(HDD使用時)
			ファン未使用※1	使用不可
		表示面側	5℃～40℃	

※1 本体内部にある電源ファンを取り外した場合。

■取り付け角度について

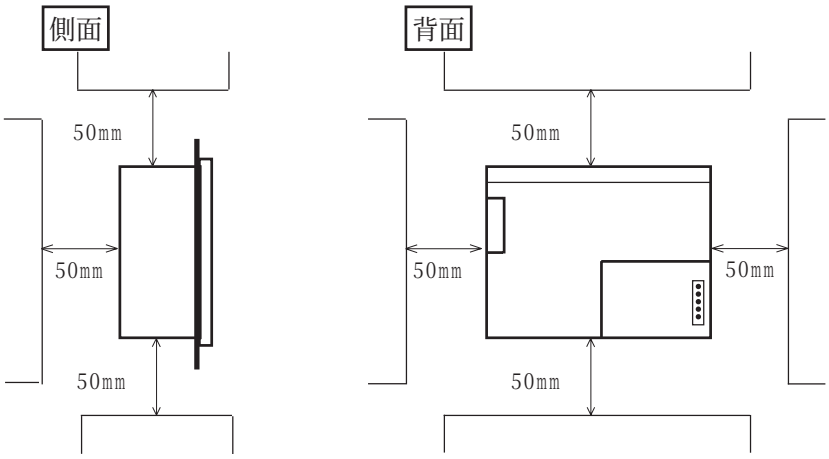
上記の使用周囲温度範囲内で、極力垂直に取り付けてください。

傾けて取り付ける場合は、本機内部での熱ごもりを最小限にするために垂直から前後30°以内で取り付けてください。



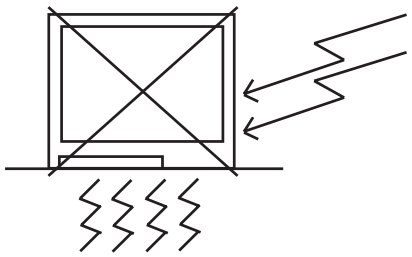
■設置場所について

- 他の機器の発熱でPLが過熱しないようにしてください。
- 電磁開閉機やノーヒューズブレーカーなどのアークを発生させる機械からは遠ざけて設置してください。
- 腐食性ガスが発生する環境では使用しないでください。
- 保守性、操作性、および風通しを良くするため、PLと構造物や部品との間は、50㎜以上としてください。PLを取り付けた状態で拡張ボードの抜き差しを行う場合や使用されるコネクタ等の形状を考慮し、十分な間隔を確保してください。



■振動・衝撃について

- 盤の扉の開閉時や、キャスター付きラックに組み込んだ場合の移動時には、ハードディスクに大きな振動や衝撃が加わる可能性があります。取り扱いには十分注意してください。



	耐震動
HDD使用時	4.9m/s^2
FDD使用時	9.8m/s^2
ドライブ非装着時	19.6m/s^2

重要

- **ハードディスクは精密機器ですので衝撃を与えないでください。特にPLが通電中のときには、机の上などでも向きを変えたり、移動させないでください。ハードディスクの故障の原因になります。**
- **PLをファン等で強制空冷する場合は、ハードディスク部分に直接強い風をあてないでください。ハードディスクの誤動作の原因となります。**

4.1.2 取り付け手順

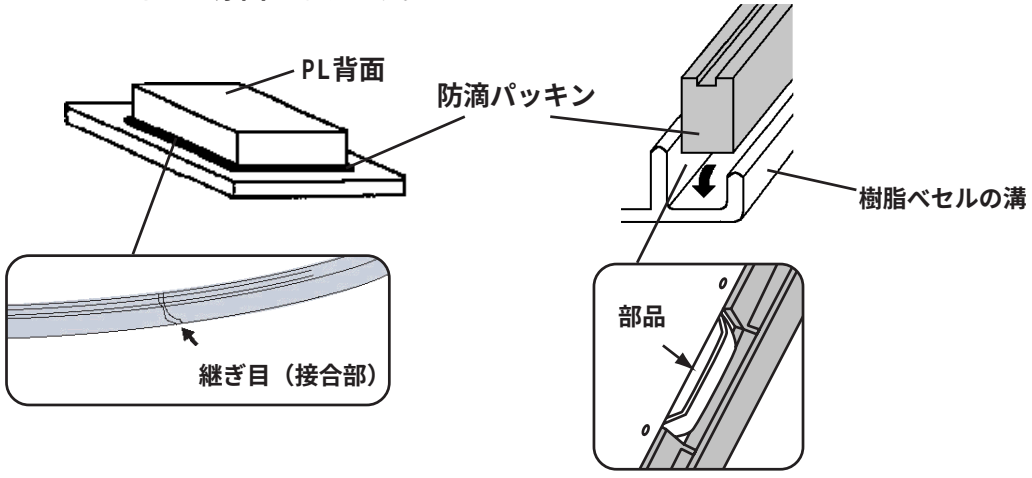
以下の方法で取り付けを行ってください。

■防滴パッキンを取り付ける

防滴効果を必要としないような環境においても防滴パッキン(本体付属)は、必ず使用してください。PLの表示面を下にして水平なところに置き、付属の防滴パッキンを背面部から樹脂ベゼルの溝に取り付けます。

重要

- **取り付けをする前に、パッキンがPLに装着されているか必ず確認してください。**
- **長期間使用した防滴パッキンはキズや汚れがつき防塵・防滴効果が得られない場合があります。定期的(キズや汚れが目立ってきた場合)に交換してください。**
- **防滴パッキンを必要としないような環境においても、防滴パッキン(本体付属)は必ず使用してください。**
- **パッキンは伸縮性がないため、引っ張らないでください。無理に引っ張るとちぎれる恐れがあります。**
- **PLの角部にパッキンの継ぎ目(接合部)を挿入しないでください。挿入すると、継ぎ目に引っ張る力が加わり、パッキンがちぎれる原因となります。**



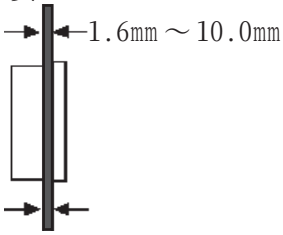
MEMO
・ パッキンには凹型のくぼみがあります。水平面が下になるように取り付けてください。
・ 樹脂ベセルの溝には上図のような部品がついています。防滴パッキンが部品に引っかかった状態にならないように、溝の底まで挿入してください。

重要
・ パッキンが均等に2㎜程度、溝から表面に現れていれば、正しく装着された状態です。取り付けの際は、必ず装着状態を確認してください。
・ パッキンが溝に正しく装着されていないと、防滴効果(IP65f相当)は得られません。

■取り付け穴をあける

取り付け穴図に従い、取り付け部分に加工を行います。取り付けには、防滴パッキン、取り付け金具が必要です。**参照→ 2.5.6 パネルカット寸法**

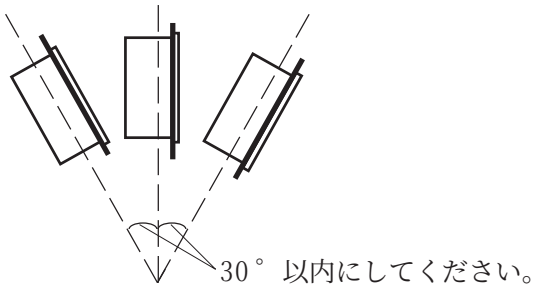
MEMO
・ 防滴効果を得るため、取り付け部（パネル）には反りや傷、凹凸のない良好な平面を選んでください。反りを防止するためには補強板をつけることも有効です。
・ パネル厚許容範囲は、1.6㎜～10.0㎜です。



重要
・ パネルの強度を十分考慮の上、パネル厚を決定してください。

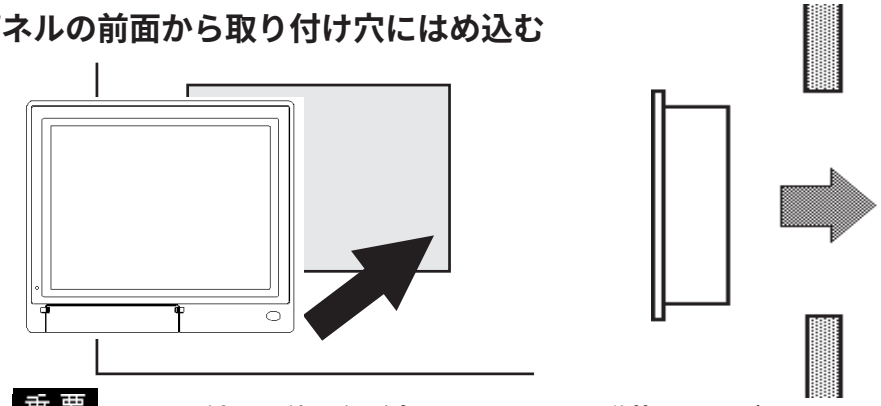
■角度に注意して取り付ける

・ PLは垂直取り付けで自然冷却ではなく、空冷ファンでの冷却を基本にしています。斜めに設置する場合の取り付け角度は、垂直より30°以内にしてください。



・ 他の機器の発熱でPLが過熱しないようにしてください。
・ 使用周囲温度以上の環境下では使用しないでください。
・ 電磁開閉機やノーヒューズブレーカーなどのアークを発生させる機械からは遠ざけて設置してください。
・ 腐食性ガスが発生する環境では使用しないでください。

■パネルの前面から取り付け穴にはめ込む



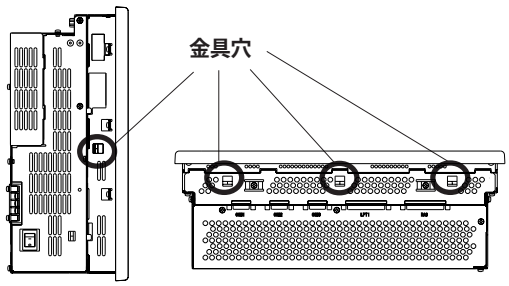
重要 ・ 取り付け公差は必ず守ってください。脱落の恐れがあります。

参照→ 2.5.6 パネルカット寸法

■パネルの裏側を取り付け金具で固定する

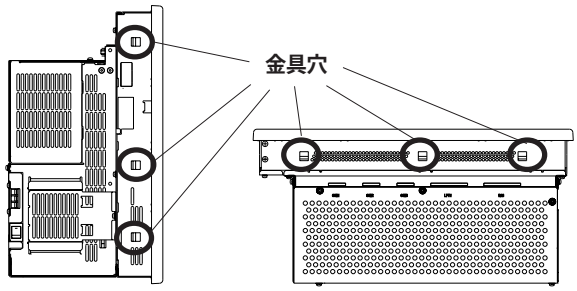
① PL の上下左右にある金具穴(PL-6920 シリーズは 8 カ所、PL-7920 シリーズは 12 カ所)に、取り付け金具のフックを入れます。下図は上面と左側面図です。底面と右側面図にも同様の取り付け穴があります。

PL-6920 シリーズ



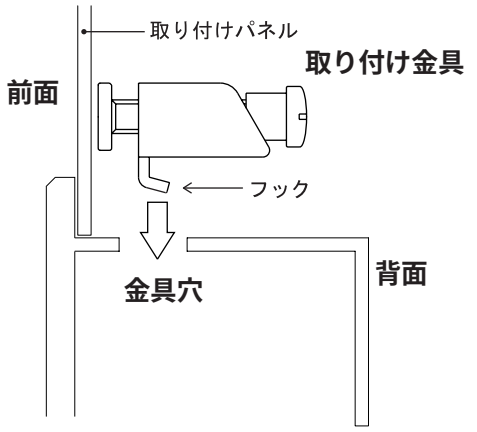
左側面　　上面

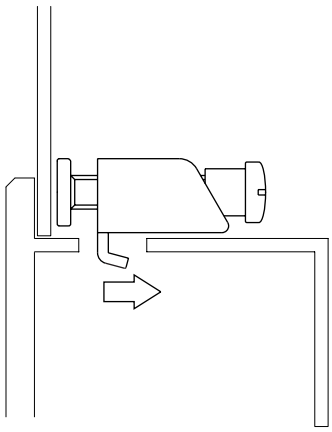
PL-7920 シリーズ



左側面　　上面

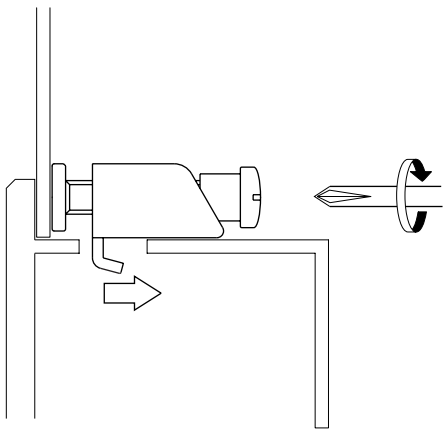
②金具穴に入れたら、金具を後ろへずらします。





③取り付け金具のネジを締めます。

4カ所のネジを対角に少しずつ締めてください。

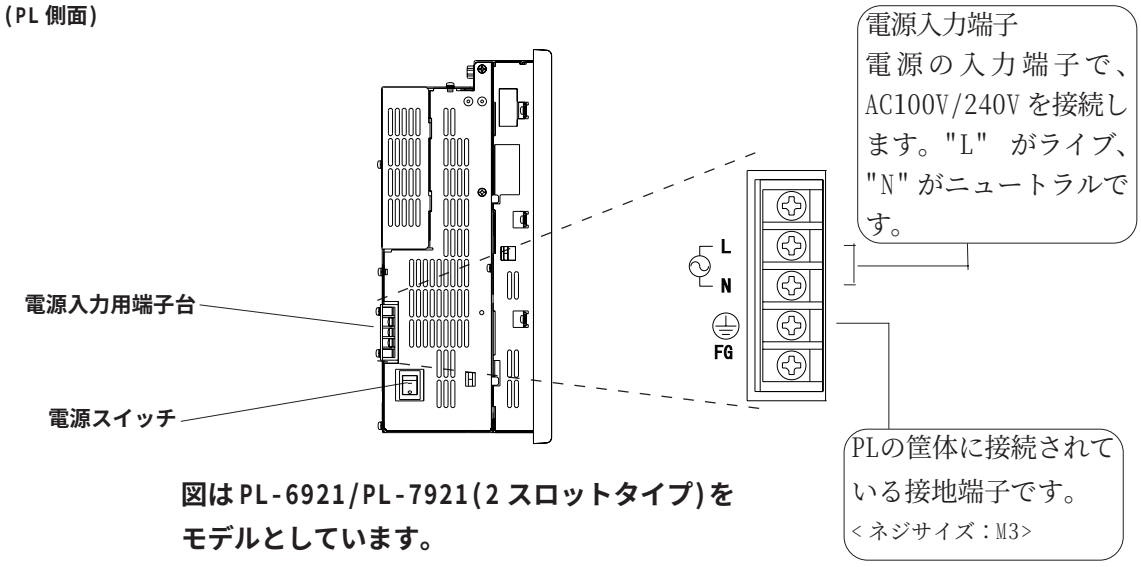


重要 **・強く締めすぎると破損する恐れがあります。防滴性確保のための適正締め付けトルクは0.5N・mです。**

4.2 配線について

4.2.1 電源ケーブルの接続

電源ケーブルは、PL背面にある電源入力用端子台に接続します。



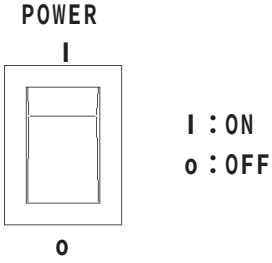
図はPL-6921/PL-7921(2スロットタイプ)をモデルとしています。

電源ケーブルは、以下の手順に従って接続してください。

⚠ 警 告

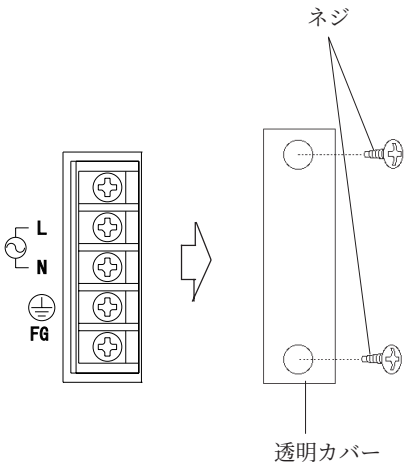
■ 電源ケーブルの取り付けは､必ず電源が供給されていないことを確認して取り付けてください。感電のおそれがあります。

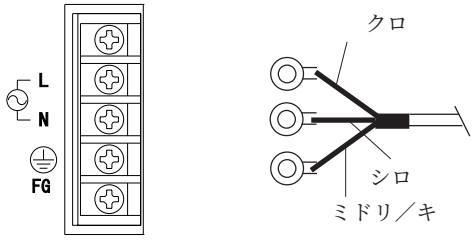
■ 記載の電源電圧以外の電圧で使用しないでください｡火災､感電､および破損のおそれがあります。



I：ON
o：OFF

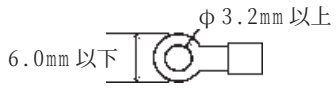
① 電源スイッチがOFFになっていることを確認した後、PLの背面にある電源入力用端子台の透明カバーを外します。

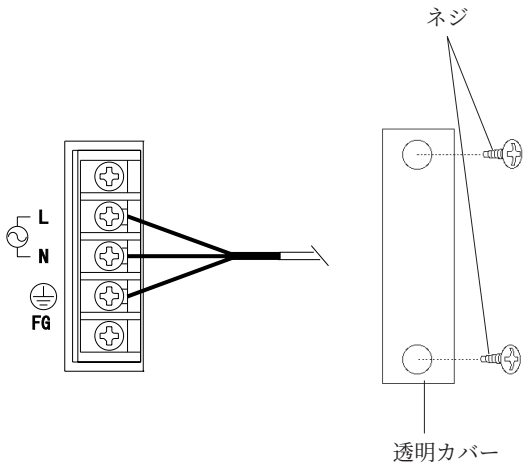




② 端子台の中央3ヵ所のネジを外し、圧着端子をネジ穴にあわせた後、ネジ止めします。

MEMO
・使用圧着端子:V1.25-3 相当品
(日本圧着端子製造(株)製)
(JIS 規格型番 RAV1.25-3)
・端子寸法は、以下の条件のものを使用してください。



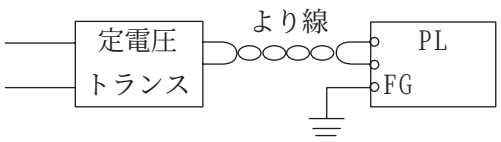


重要
・図のケーブル色は、付属のケーブルを使用した場合の色です。
・付属のケーブルは AC100V 専用です。他の電圧ではそれぞれの各規格に合ったケーブルを使用してください。

③ 透明カバーを電源入力用端子台にネジ止めします。

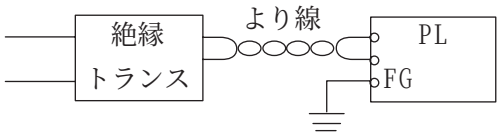
4.2.2 電源供給時の注意事項

電源供給時の注意事項です。下記の注意事項を守り、PL 背面の電源入力用端子台に電源ケーブルを接続してください。



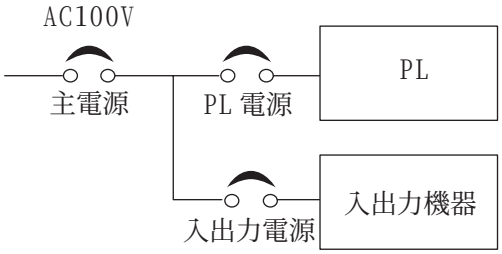
・ 電圧変動が規定値以上の場合は、定電圧トランスを接続してください。

電圧の規定値については、 **参照→2.1 一般仕様**

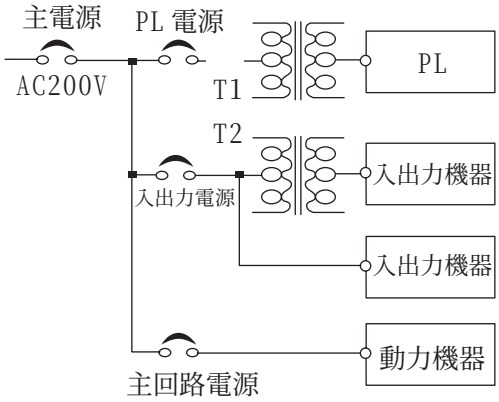


・ 線間や大地間は、ノイズの少ない電源を使用してください。ノイズが多い場合は、絶縁トランス（ノイズカットトランス）を接続してください。

MEMO ・定電圧トランス、絶縁トランスは、容量 200VA 以上のものを使用してください。

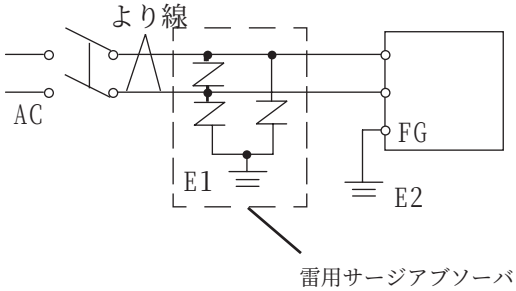


・ PL の電源と入出力機器、および動力機器とは、系列を分離して配線してください。



・ 耐ノイズ性を高めるために、電源ケーブルにフェライトコアを取り付けることをお勧めします。

・ 主回路（高電圧、大電流）線、入出力信号線、電源ケーブルは、束線、近接をしないでください。



・ 雷のサージ対策に、雷用サージアブソーバを接続してください。

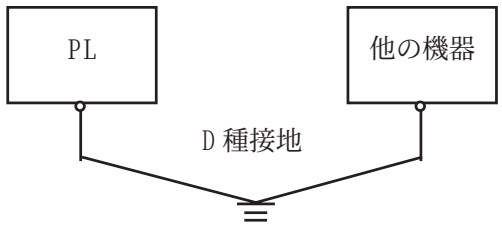
重要 **・雷用サージアブソーバの接地(E1）と PL の接地（E2）とは分離して行ってください。**

・電源電圧最大上昇時でも、サージアブソーバの最大許容回路電圧を超えないような雷用サージアブソーバを選定してください。

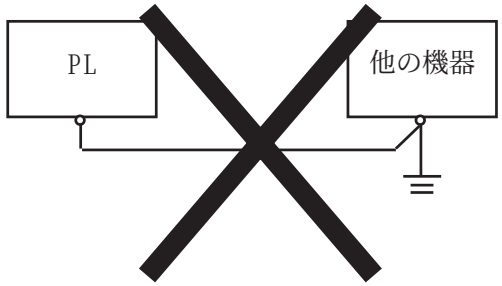
4.2.3 接地時の注意事項

（a）専用接地　最良

（b）共用接地　良



（c）共用接地　不可



- PL 背面にある FG 端子からの接地は、専用接地としてください。「図（a）、接地工事はD種接地、接地抵抗100 Ω以下」
- 専用接地がとれないときは、図（b）の共用接地としてください。
- 本機は内部で SG（シグナルグラウンド）と FG（フレームグラウンド）が接続されています。
- 接続装置と SG を接続する場合は、短絡ループが形成されないようにシステムを設計してください。
- 2mm²以上の接地用電線を使用してください。

  接地点は、PL の近くで接地線の距離を短くしてください。接地線が長くなる場合は、太い絶縁線を使用し、電線管を通して敷設してください。

4.2.4 入出力信号接続時の注意事項

- 入力信号線、および出力信号線は、動力回路のケーブルとは別の配線系統に布線をしてください。
- 動力回路ケーブルをどうしても別の配線系統にできないときは、シールドケーブルを使用して、シールド端を PL の FG に落としてください。
- 耐ノイズ性を高めるために、通信ケーブルにフェライトコアを取り付けることをおすすめします。

第 5 章
システムのセットアップ

1. システムセットアップ手順
2. システム情報の設定内容

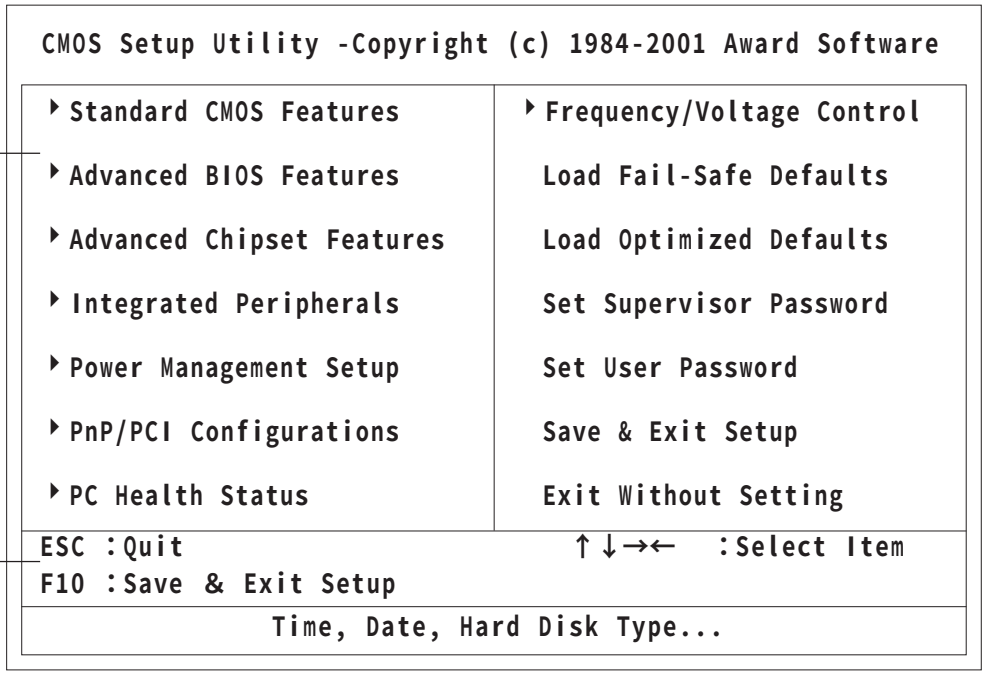
システムのセットアップ手順と設定内容について説明します。

5.1 システムセットアップ手順

以下の全説明画面は、出荷時設定(初期設定)画面です。

重要 **・通常は、出荷時設定(初期設定)で使用してください。**

① PL にキーボードを接続します。
② PL の電源を ON にします。
③ 画面左下 "Press <DEL> to Enter SETUP" のメッセージが表示されたら、[DEL]キーを押し続けます。
セットアップユーティリティが起動し、次のようなメニュー画面が表示されます。



キー操作一覧
セットアップで使用するキーの一覧です。

システム設定エリア
各メニューで設定するシステム項目が表示されます。

④ カーソルを移動してシステム項目を選択し、[Enter]キーで確定します。
各システム設定画面が表示されます。

5.2 システム情報の設定内容

メニュー画面でシステム項目を選択し、システム情報を設定します。各システム項目ごとの詳細設定を示します。

重要 **・通常は出荷時の設定(初期設定)で使用してください。**

5.2.1 Standard CMOS Features

メニュー画面にてStandard CMOS Featuresを選択すると以下の画面が表示されます。

```
        CMOS Setup Utility - Copyright (C) 1984-2001 Award Software
                          Standard CMOS Features

  Date (mm:dd:yy):          Tue,Jul 2 2001          |        Item Help
  Time (hh:mm:ss):          14 : 50 : 3             |
                                                    |  Menu Level   ▸
▸ IDE Primary Master        [IC25N020ATDA04-0]      |
▸ IDE Primary Slave         [None]                  |  Change the day, month,
                                                    |  year and century
  Drive A                   [1.44M, 3.5 in.]        |
  Drive B                   [None]                  |
                                                    |
  Video                     [EGA/VGA]               |
                                                    |
  Halt On                   [All,But Disk/Key]      |
                                                    |
  Base Memory               640K                    |
  Externded Memory          129024K                 |
  Total Memory              130048K                 |

↑↓→←:Move  Enter:Select  +/-/PU/PD:Value  F10:Save  ESC:Exit  F1:General Help
      F5:Previous Values     F6:Fail-Safe Defaults      F7:Optimized Defaults
```

■ Date/Time

PL内蔵のカレンダ時計に日付、時刻を設定します。
時:00 ～ 23
分:00 ～ 59
秒:00 ～ 59

■ IDE Primary Master (Slave)

PLで接続されているIDEタイプハードディスクの容量が表示されます。
[Enter]キーでパラメータ設定メニューが表示されます。**参照→5.2.2 IDE HDD Auto Detection**

■ Drive A (B)

PLに接続するFDドライブの種類をセットします。
[None]or[720K, 3.5in]or[1.44M,3.5in]or[2.88M,3.5in]の選択となります。出荷時の設定はDrive A [1.44M,3.5in]、Drive B [None]です。通常、出荷時の設定でご使用ください。

■ Video

システムで使用する画面モード(ビデオモード)を選択します。
[EGA/VGA]or[CGA40]or[CGA80]or[MONO]の選択が可能です。出荷時の設定は[EGA/VGA]です。
通常、出荷時の設定でご使用ください。

■ Halt On

起動時のセルフテストでエラーが検出された場合の設定です。出荷時の設定は[All, But Disk/Key］です。通常、出荷時の設定でご使用ください。

[All Errors］　　：全てのエラーを表示し停止します。

［No Errors］　　：エラー表示、停止をしません。

［All, But Keyboard］：キーボードを除くエラーのみを表示し停止します。

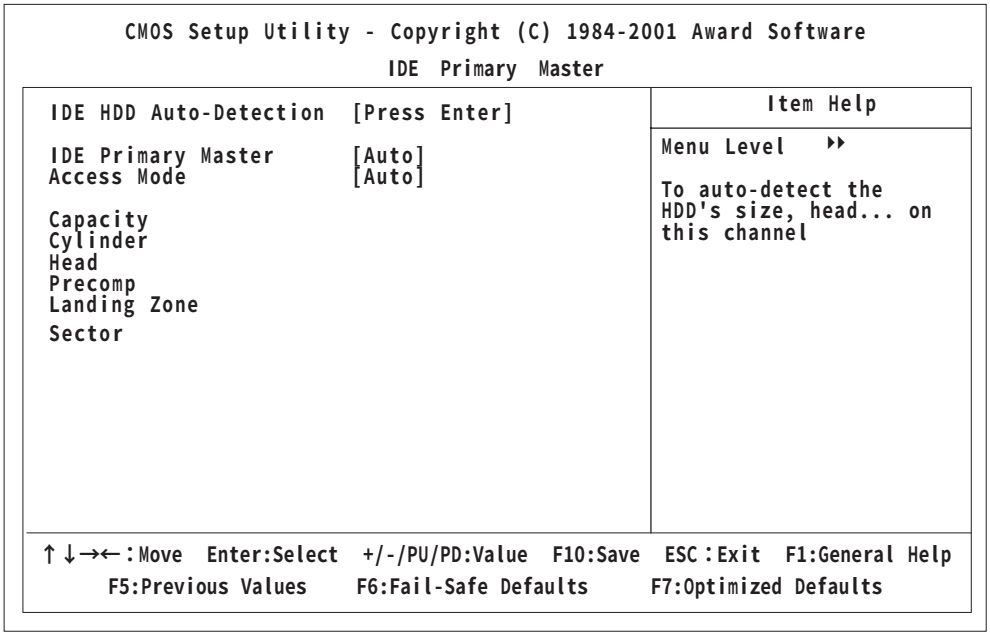
キーボードを接続しないでご使用になる場合はこの設定にしてください。

［All, But Diskette］：FDDを除くエラーのみ表示し停止します。

［All, But Disk/Key］：キーボード、FDDを除くエラーのみ表示し停止します。

5.2.2 IDE HDD AUTO DETECTION

Standard CMOS Features のメニューにて、IDE Primary Master もしくは IDE Primary Slave を選択すると、以下のメニュー画面が表示されます。



■ IDE HDD Auto-Detection

IDE に接続されたハードディスクを自動検出します。通常は使用しません。

■ IDE Primary Master (Slave)

PL に接続する IDE タイプハードディスクのパラメータの設定方法を選択します。[None]or[Auto]or[Manual]の選択となります。出荷時の設定は[Auto]です。通常、出荷時の設定でご使用ください。

■ Access Mode

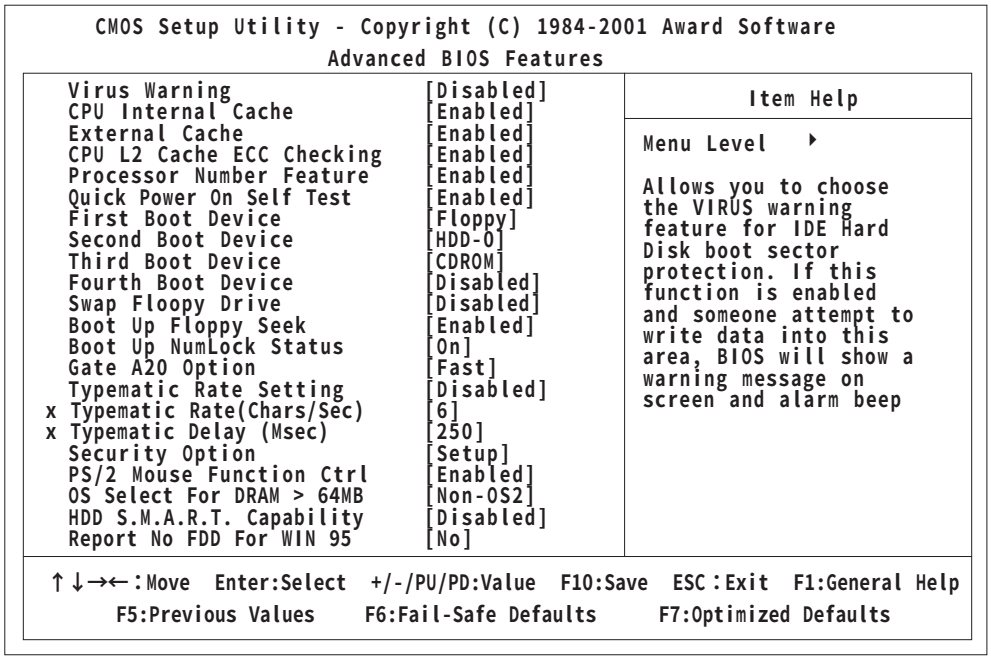
IDE に接続されたハードディスクのアクセスモードを選択します。[CHS]or[LBA]or[Large]or[Auto]の選択となります。出荷時の設定は[Auto]です。通常、出荷時の設定でご使用ください。

■ Capacity / Cylinder / Head / Precomp / Landing Zone / Sector

PL に接続する IDE タイプハードディスクのパラメータを設定します。[IDE Primary Master (Slave)]が[Manual]かつ、Access Mode が[CHS]の場合のみ設定できます。[IDE Primary Master (Slave)]が[Auto]の場合は自動検出された値が表示されます。Capacity の設定は自動です。

5.2.3 Advanced BIOS Features

メニュー画面にてAdvanced BIOS Featuresを選択すると以下の画面が表示されます。



■ Virus Warning

ハードディスクのBoot Sectorへの書き込みが発生した場合、警告表示を行うかどうかを設定します。[Disabled]or[Enabled]の選択となります。出荷時の設定は[Disabled]です。通常、出荷時の設定でご使用ください。

■ CPU Internal Cache

CPU内蔵のキャッシュメモリの使用有無を設定します。[Disabled]or[Enabled]の選択となります。出荷時の設定は[Enabled]です。通常、出荷時の設定でご使用ください。

■ External Cache

外部(L2)キャッシュメモリの使用有無を設定します。[Disabled]or[Enabled]の選択となります。出荷時の設定は[Enabled]です。通常、出荷時の設定でご使用ください。

■ CPU L2 Cache ECC Checking

外部(L2)キャッシュメモリのECC(Error Check Correction) の有効、無効を設定します。[Disabled]or[Enabled]の選択となります。出荷時の設定は[Enabled]です。通常、出荷時の設定でご使用ください。

■ Processor Number Feature

CPUのシリアル番号をチェックします。[Disabled]or[Enabled]の選択となります。出荷時の設定は[Enalbed]です。システムにシリアル番号を知らせたくない場合は、無効にします。通常PentiumⅢのプロセッサ・シリアル・ナンバはOffの状態で出荷されており、[Enabled]にしても有効にはなりません。インテルのWebサイトからプロセッサ・シリアル・ナンバ制御プログラムを用いてOnした時にはじめてこの機能が有効になります。

■ Quick Power On Self Test

電源On時のセルフテストを簡易に行うかどうかを設定します。[Disabled]or[Enabled]の選択となります。出荷時の設定は[Enabled]です。通常、出荷時の設定でご使用ください。

■ First/Second/Third/Fourth Boot Device

OSをどのドライブから起動するかを選択します。選択肢は[Floppy],[LS120],[HDD-0]※1,[SCS1],[CDROM],[HDD-1]※1,[ZIP100],[LAN],[ISA-FDD],[Disabled]の選択となります。

■ Swap Floppy Drive

A、BドライブをB、Aのようにドライブ割当の交換を行うかどうか設定します。[Disabled]or[Enabled]の選択となります。出荷時の設定は[Disabled]です。通常、出荷時の設定でご使用ください。

■ Boot Up Floppy Seek

システム立ち上げ時、フロッピーディスクドライブを装着しているかどうかをチェックする機能を設定します。[Disabled]or[Enabled]の選択となります。出荷時の設定は[Enabled]です。通常、出荷時の設定でご使用ください。

■ Boot Up Numlock Status

起動時点におけるNumLockキーの状態を設定します。[On]or[Off]の選択となります。出荷時の設定は[On]です。通常、出荷時の設定でご使用ください。

■ Gate A20 Option

[Fast]or[Normal]の選択となります。[Normal]を選択した場合は、Gate A20のコントロールにKeyboradコントロールを使用します。[Fast]を選択した場合は、Chipsetを使用します。出荷時の設定は[Fast]です。通常、出荷時の設定でご使用ください。

※1 PL本体の板金に刻印されているスロット位置を示すHDD0やHDD1とは関係がありません。

HDD-0:
- **HDDが一つだけ装着されている場合**
  HDDの設定がマスタ/スレーブに関係なく、そのHDDをHDD-0と認識します。
- **HDDが二つ装着されている場合**
  設定がマスタとなっているHDDをHDD-0と認識します。

HDD-1:
- **HDDが一つだけ装着されている場合**
  OS起動は不可能です。BIOS設定は必ずHDD-0にしてください。
- **HDDが二つ装着されている場合**
  設定がスレーブになっているHDDをHDD-1と認識します。

■ Typematic Rate Setting

キーボードのリピート文字処理の設定を行います。
[Disabled]or[Enabled]の設定ができます。出荷時の設定は[Disabled]です。

■ Typematic Rate (Chars/Sec)

実際のレート（1秒あたりの繰り返し入力文字数）です。出荷時の設定は[6]です。
Typematic Rate Settingが[Enabled]の場合のみ設定できます。

■ Typematic Delay (Msec)

最初の文字のリピートが始まるまでの遅延時間です。単位はミリ秒（msec）です。
出荷時の設定は[250]です。Typematic Rate Settingが[Enabled]の場合のみ設定できます。

■ Security Option

パスワードの入力要求が行われる場所を指定します。BIOSセットアップ時にパスワードの入力要求をする場合は[Setup]を、システム起動時にパスワードの入力要求をする場合は[System]を選択してください。この設定は、[Set Supervisor Password]or[Set User Password]でパスワードが設定されていない場合は無効です。出荷時の設定は[Setup]です。通常、出荷時の設定でご使用ください。
[Set Supervisor Password]については、**5.2.14 Set Supervisor Password**を
[Set User Password]については、**5.2.15 Set User Password**をご参照ください。

■ PS/2 Mouse Function Ctrl

[Disabled]or[Enabled]の選択となります。出荷時の設定は[Enabled]です。

■ OS Select For DRAM >64MB

[Non-OS2]or[OS2]の選択となります。出荷時の設定は[Non-OS2]です。通常、出荷時の設定でご使用ください。

■ HDD S.M.A.R.T Capability

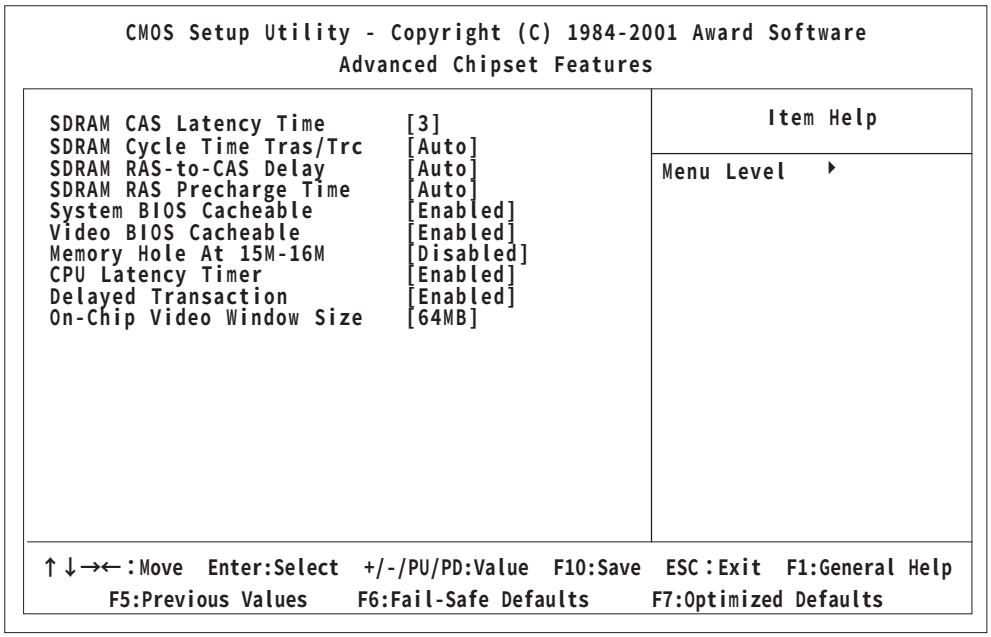
HDDのS.M.A.R.T(Self-Monitoring Analysis and Reporting Technology)機能を設定します。[Disabled]or[Enabled]の設定ができます。出荷時の設定は[Disabled]です。

■ Report No FDD For Win 95

選択肢は[Yes]or[No]です。フロッピードライブなしでWindows®95を使用する場合は[Yes]を選択してください。そうでない場合は[No]を設定してください。出荷時の設定は[No]です。

5.2.4 Advanced Chipset Features

メニュー画面にてAdvanced Chipset Featuresを選択すると以下の画面が表示されます。



■ SDRAM CAS Latency Time

CASが有効になってからバースト転送が始まるまでのクロック数の設定をします。[3]or[2]の設定が可能です。出荷時の設定は[3]です。必ず、出荷時の設定でご使用ください。

■ SDRAM Cycle Time Trans/Trc

バンク活性時間および同一バンクを活性化するために必要な最小時間を設定します。[7/9]or[5/7]or[Auto]の選択です。出荷時の設定は[Auto]です。必ず、出荷時の設定でご使用ください。

■ SDRAM RAS-to-CAS Delay

RASが有効になってからCASが有効になるまでのクロック数を設定します。[3]or[2]or[Auto]の選択です。出荷時の設定は[Auto]です。必ず、出荷時の設定でご使用ください。

■ SDRAM RAS Precharge Time

プリチャージ時間を設定します。[3]or[2]or[Auto]の設定が可能です。出荷時の設定は[Auto]です。必ず、出荷時の設定でご使用ください。

■ System BIOS Cacheable

システムBIOSのキャッシングをするかどうかを設定します。システムBIOSを使用するOSを使用した場合、動作スピードを高速化できます。[Enabled]or[Disabled]の設定が可能です。

出荷時の設定は[Enabled]です。通常、出荷時の設定でご使用ください。

■ Video BIOS Cacheable

Video BIOSのキャッシングをするかどうかの設定をします。選択肢は[Enabled]or [Disabled]です。出荷時の設定は[Enabled]です。[Enabled]にするとシステムのBIOS ROM領域C0000h-F7FFFhがキャッシング可能になり、ビデオパフォーマンスが上がります。しかし、他のプログラムが、このメモリ領域に書き込みをしようとすると、システムエラーが起こる可能性があります。

■ Memory Hole At 15M-16M

ISAカードに15-16MBのメモリ領域を確保してパフォーマンスの向上を図る設定をします。選択肢は[Enabled]or[Disabled]です。出荷時の設定は[Disabled]です。

■ CPU Latency Timer

選択肢は[Enabled]or[disabled]です。出荷時の設定は[Enabled]です。[Enabled]にすると、変更可能なCPUサイクルが、31クロックの間スヌープストール(命令の依存関係などの条件によって、演算ユニットの命令の実行が停止状態になること)になった後に、さらにADS#がきた時にのみ変更されます。[Disabled]の時は、変更可能なCPUサイクルはGMCHがADS#を受けるとすぐに変更されます。

■ Delayed Transaction

プリチャージ時間を設定します。選択肢は[Enabled]or[Disabled]です。出荷時の設定は[Enabled]です。

■ On-Chip Video Window Size

VGAドライバを使用するためにオンチップビデオウィンドウサイズを設定します。[Disabled]or[64MB]の設定が可能です。出荷時の設定は[64MB]です。

5.2.5 INTEGRATED PERIPHERALS

メニュー画面にてINTEGRATED PERIPHERALSを選択すると以下の画面が表示されます。

```
        CMOS Setup Utility - Copyright (C) 1984-2001 Award Software
                          Integrated Peripherals
  On-Chip Primary PCI IDE   [Enabled]          |        Item Help
  IDE Primary Master PIO    [Auto]             |
  IDE Primary Slave PIO     [Auto]             |  Menu Level   ▸
  IDE Primary Master UDMA   [Auto]             |
  IDE Primary Slave UDMA    [Auto]             |
  USB Controller            [Disabled]         |
x USB Keyboard Support       Disabled          |
  Init Display First        [PCI Slot]         |
  AC97 Audio                [Disabled]         |
  Onboard LAN               [Disabled]         |
  IDE HDD Block Mode        [Enabled]          |
  POWER ON Function         [BUTTON  ONLY]     |
x KB Power ON Password       Enter             |
x Hot Key Power On           Ctrl-F1           |
  Onboard FDC Controller    [Enabled]          |
  Onboard Serial Port 1     [3F8/IRQ4]         |
  Onboard Serial Port 2     [2F8/IRQ3]         |
  Onboard Serial Port 3     [3E8H/IRQ9]        |
  Onboard Serial Port 4     [2E8H/IRQ10]       |
  Onboard Parallel Port     [3BC/IRQ7]         |
  Parallel Port Mode        [SPP]              |
x EPP Mode Select            EPP1,7            |
x ECP Mode Use DMA           3                 |
  PWRON After PWR-Fail      [ON]               |
↑↓→←:Move  Enter:Select  +/-/PU/PD:Value  F10:Save  ESC:Exit  F1:General Help
     F5:Previous Values    F6:Fail-Safe Defaults     F7:Optimized Defaults
```

■ On-Chip Primary PCI IDE

チップセットがIDEのファーストチャネルのサポートを有効にするかどうかの設定です。
[Disabled]or[Enabled]の選択が可能です。
出荷時の設定は［Enabled］です。通常、出荷時の設定でご使用ください。

■ IDE Primary Master (Slave) PIO

IDEのプライマリーマスター(プライマリースレーブ)のPIO（Programmed Input/Output）モードを設定します。[Auto]or[Mode 0]or[Mode 1]or[Mode 2]or[Mode 3]or[Mode 4]の選択が可能です。
出荷時の設定は[Auto]です。通常、出荷時の設定でご使用ください。

■ IDE Primary Master (Slave) UDMA

IDEのプライマリーマスター(プライマリースレーブ)のUDMA（Ultra DMA)モードを設定します。[Auto]or[Disabled]の選択が可能です。出荷時の設定は[Auto]です。通常、出荷時の設定でご使用ください。

■ USB Controller

USB周辺機器を使用する場合に設定します。[Disabled]or[Enabled]の選択が可能です。
出荷時の設定は[Disabled]です。USB I/Fを使用する場合やタッチパネルの接続方法をUSB接続にする場合は[Enabled]を選択します。

重要　USB周辺機器を使用する場合は[USB Controller]と[PnP/PCI Configuration]メニューの[Assign IRQ For USB]を[Enabled]に変更してください。

■ USB Keyboard Support

USB キーボードをサポートしない OS で、BIOS が USB キーボードのドライバを用意し PS/2 キーボードと同じように扱う場合に設定します。出荷時の設定は[Disabled]です。[Disabled]or[Enabled]の設定が可能です。USB controller が[Disabled]の場合には変更できません。

重要 **USB周辺機器を使用する場合は[USB Controller]と[PnP/PCI Configuration]メニューの[Assign IRQ For USB]を[Enabled]に変更してください。**

■ Init Display First

PCI と AGP 両方インストールされている場合、どちらかを先に出力するかを設定します。選択肢は[PCI Slot]or[Onboard/AGP]です。出荷時の設定は[PCI Slot]です。

■ AC97 Audio

810 チップセットの AC97 オーディオ機能を設定します。[Disabled]or[Auto]の選択です。出荷時の設定は[Disabled]です。

■ Onboard LAN

[Enabled]or[Disabled]の設定です。出荷時の設定は[Disabled]です。

■ IDE HDD Block Mode

Block Mode をサポートしている HDD において、Block Mode を有効にするかどうかの設定です。[Disabled]or[Enabled]の選択が可能です。
出荷時の設定は［Enalbed］です。通常、出荷時の設定でご使用ください。

■ Power ON Function

サポートしておりません。

■ KB Power ON Password

サポートしておりません。

■ Hot Key Power ON

サポートしておりません。

■ Onboard FDC Controller

オンボードのフロッピーコントローラを有効にするかどうかの設定です。[Enabled]or[Disabled]の選択が可能です。出荷時の設定は[Enabled]です。通常、出荷時の設定でご使用ください。

■ Onboard Serial Port 1

オンボードのシリアルポート 1 がどの I/O アドレスを使用するかを決定します。[Disabled]or[Auto]or[3F8/IRQ4]or[2F8/IRQ3]or[3E8/IRQ4]or[2E8/IRQ3]の選択が可能です。
出荷時の設定は[3F8/IRQ4]です。通常、出荷時の設定でご使用ください。

■ Onboard Serial Port 2

オンボードのシリアルポート2がどのI/Oアドレスを使用するかを決定します。[Disabled]or[Auto]or[3F8/IRQ4]or[2F8/IRQ3]or[3E8/IRQ4]or[2E8/IRQ3]の選択が可能です。

出荷時の設定は[2F8/IRQ3]です。通常、出荷時の設定でご使用ください。

■ Onboard Serial Port 3

オンボードのシリアルポート3がどのポートアドレスを使用するかを決定します。[Disabled]or[3F8H/IRQ9]or[2F8H/IRQ10]or[3E8H/IRQ9]or[2E8H/IRQ10]の選択となります。

出荷時の設定は[3E8H/IRQ9]です。通常、出荷時の設定でご使用ください。

■ Onboard Serial Port 4

オンボードのシリアルポート4(本体内部でタッチパネルに接続されています)がどのポートアドレスを使用するかを決定します。[Disabled]or[3F8H/IRQ9]or[2F8H/IRQ10]or[3E8H/IRQ9]or[2E8H/IRQ10]の選択となります。出荷時の設定は[2E8H/IRQ10]です。通常、出荷時の設定でご使用ください。タッチパネルの接続方法をUSB接続にする場合は[Disabled]を選択します。

■ Onboard Parallel Port

オンボードのパラレルポート1がどのI/Oアドレスを使用するかを決定します。[Disabled]or[3BC/IRQ7]or[378/IRQ7]or[278/IRQ5]の選択が可能です。

出荷時の設定は[3BC/IRQ7]です。通常、出荷時の設定でご使用ください。

■ Parallel Port Mode

オンボードのパラレルポートの動作モードを決定します。[SPP]or[ECP]or[EPP]or[ECP+EPP]の選択が可能です。出荷時の設定は[SPP]です。出荷時設定ではパラレルポートの動作モードは［SPP］と［ECP］しか使用できません。オンボードパラレルポートが[378/IRQ7]もしくは[278/IRQ5]の場合は[SPP]or[ECP]or[EPP]or[ECP+EPP]の選択が可能です。

■ EPP Mode Select

［EPP］または［ECP+EPP］モードで使用するときに、EPPモードプロトコルを選択します。[EPP 1.7]or[EPP 1.9]の選択が可能です。

■ ECP Mode Use DMA

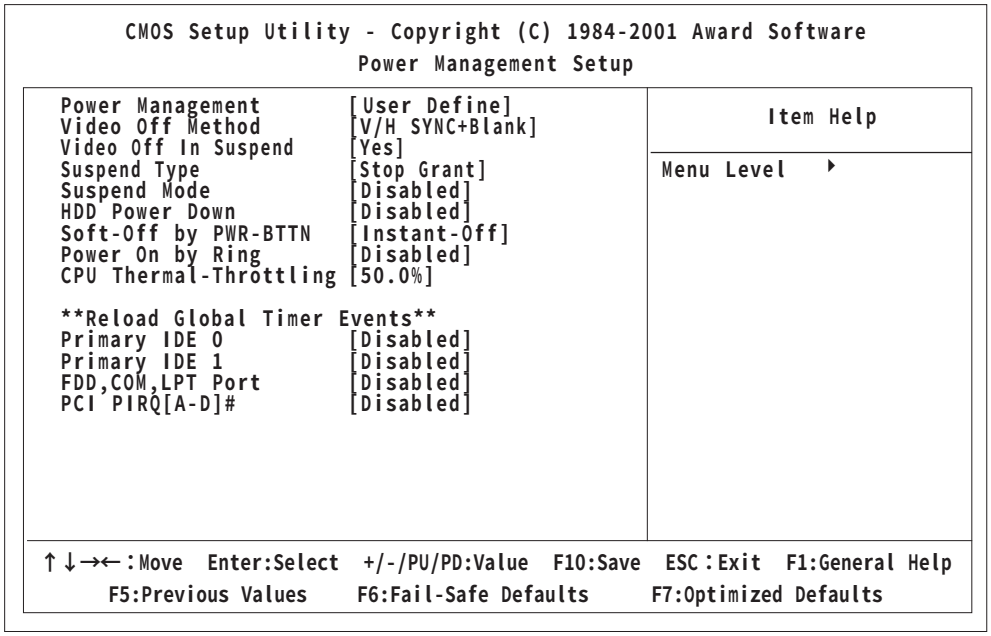
ECPモードで使用するDMAチャンネルを決定します。[1]or[3]の選択が可能です。
ただし、Parallel Port Modeの設定が[ECP]または[ECP+EPP]の場合のみ設定可能です。

■ PWRON After PWR-Fail

サポートしておりません。

5.2.6 POWER MANAGEMENT SETUP

メニュー画面にて POWER MANAGEMENT SETUP を選択すると以下の画面が表示されます。



■ Power Management

3 タイプのパワーマネジメントの設定を行います。[User Define]or[Min Saving]or[Max Saving]の選択となります。

出荷時の設定は[User Define]です。通常、出荷時の設定でご使用ください。

■ Video Off Method

ディスプレイの画面表示を消す方法を設定します。[V/H SYNC+Blank]or[Blank Screen]or[DPMS]の設定が可能です。[V/H SYNC+Blank]は画面表示を消すだけではなく、ディスプレイの水平・垂直同期信号も停止します。[Blank Screen]は画面表示のみを消します。[DPMS]は DPMS に対応した CRT を使用した場合に制御可能です。出荷時の設定は[V/H SYNC+Blank]です。通常、出荷時の設定でご使用ください。

■ Video Off In Suspend

サスペンドモードでモニタを切るかどうかを設定します。選択肢は[Yes]or[No]です。出荷時の設定は[Yes]です。

■ Suspend Type

サスペンドモードの種類を設定します。選択肢は[Stop Grant]or[PWRON Suspend]です。出荷時の設定は[Stop Grant]です。通常、出荷時の設定でご使用ください。

■ Suspend Mode

サスペンドモードに入るまでの連続アイドル時間を設定します。[1Min]or[2Min]or[4Min]or[8Min]or[12Min]or[20Min]or[30Min]or[40Min]or[1Hour]or[Disabeld]の選択です。出荷時の設定は[Disabled]です。通常、出荷時の設定でご使用ください。

■ HDD Power Down

ハードディスクのモーターを停止するまでの時間の設定を行います。[1Min]→[15Min]or [Disabled]の選択となります。出荷時の設定は[Disabled]です。通常、出荷時の設定でご使用ください。

■ Soft-Off by PWR-BTTN

ソフト制御の電源ボタンの設定を行います。選択肢は[Delay 4 sec]or[Instant-off]ですが、PLにはソフト制御の電源ボタンがありませんので[Instant-off]に設定してください。出荷時の設定は[Instant-off]です。

■ Power On by Ring

モデムからのコールがあると、システムを起動するかどうかの設定をします。選択肢は[Enabled]or[Disabled]です。出荷時の設定は[Disabled]です。

■ CPU Thermal-Throttling

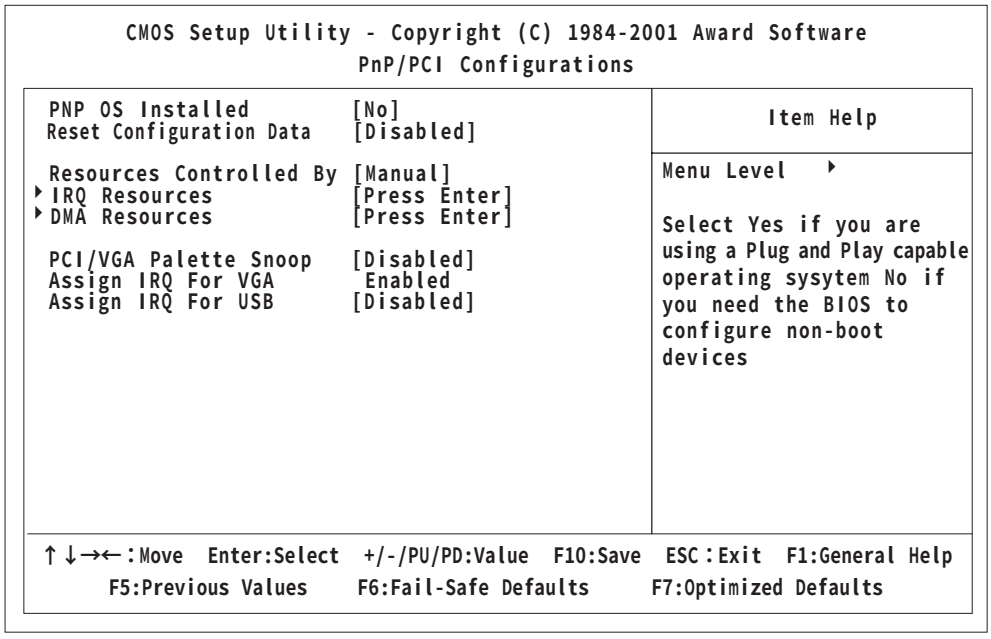
システムがサスペンドモードに入った時にCPUに供給されるクロックを通常のクロックに対する比率で設定します。出荷時の設定は[50.0%]です。

■ **Reload Global Timer Events**

省電力モードに移行するアイドル時間のタイマをリロードするイベントを設定します。[Enabled]に設定した項目の割り込みイベントが発生するとシステムはタイマをリロードします。

5.2.7 PnP/PCI Configurations

メニュー画面にて PnP/PCI Configurations を選択すると以下の画面が表示されます。



■ PNP OS Installed

プラグアンドプレイ対応の OS を使用する場合の設定です。[Yes]or[No]の選択となります。出荷時の設定は[No]です。通常、出荷時の設定でご使用ください。

■ Reset Configuration Data

セットアップユーティリティを終了したときに、プラグアンドプレイで使用する ESCD (Exteneded System Configuration Data)を初期化するかどうかの設定です。[Enabled]or [Disabled]の選択となります。出荷時の設定は[Disabled]です。通常、出荷時の設定でご使用ください。

■ Resources Controlled By

プラグアンドプレイによる I/O ポート、IRQ、DMA のリソース割り当てを自動または手動のどちらで行うかの設定を行います。[Manual]or[Auto(ESCD)]の選択となります。[Auto (ESCD)]を選択すると IRQ Resources と DMA Resources の選択はできなくなります。出荷時の設定は[Manual]です。通常、出荷時の設定でご使用ください。

■ IRQ Resources

Resource Controlled By が Manual に設定されている場合は、各デバイスに手動で割り当てる IRQ を設定する必要があります。**参照→5.2.8 IRQ Resources**

■ DMA Resources

各デバイスに手動で割り当てるDMAリソースを設定します。**参照→5.2.9 DMA Resources**

■ PCI/VGA Palette Snoop

[Enabled]or[Disabled]の選択です。出荷時の設定は「Disabled]です。通常は出荷時の設定でご使用ください。ただし、使用するVGAボード、MPEGボードによっては[Enabled]に設定してください。詳細については、VGAボード、MPEGボードの取扱説明書を参照してください。

■ Assign IRQ For VGA

VGAに対し割り込みを割り当てるかどうかの設定です。[Enabled]固定です。

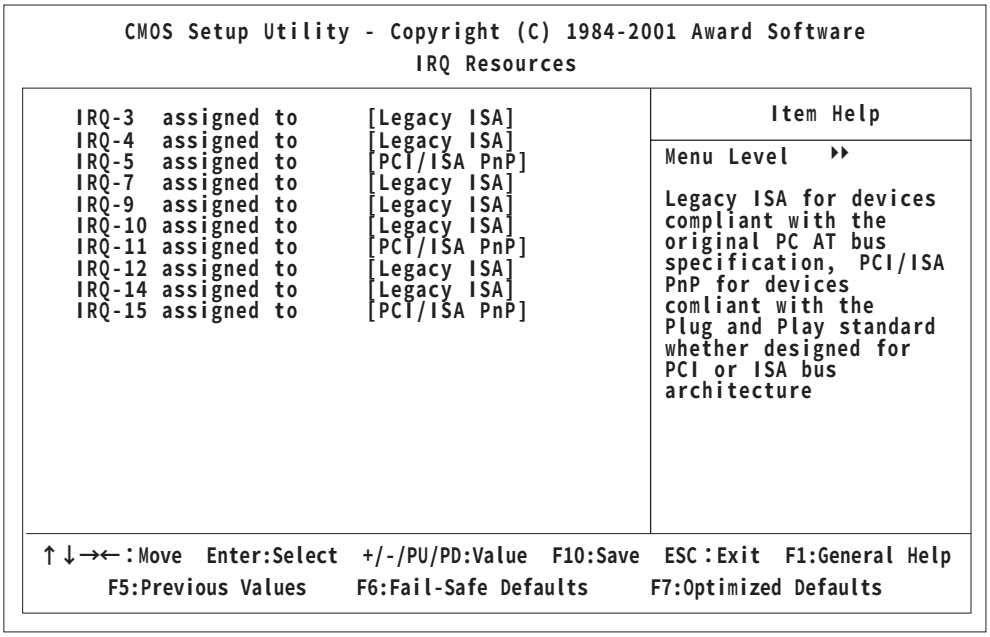
■ Assign IRQ For USB

[Enabled]or[Disabled]の選択となります。出荷時の設定は[Disabled]です。通常、出荷時の設定でご使用ください。USB I/Fを使用する場合やタッチパネルの接続方法をUSB接続にする場合は[Enabled]を選択します。

重要 **USB周辺機器を使用する場合は[INTEGRATED PERIPHERALS]メニューの[USB Controller]と[Assign IRQ For USB]を[Enabled]に変更してください。**

5.2.8 IRQ Resources

PnP/PCI Configurations のメニュー画面にて IRQ Resources を選択すると以下の画面が表示されます。



■ IRQ-3 assigned to ～ IRQ-15 assigned to

IRQ に割り当てられる機器の種類を設定します。[PnP/PCI Configurations]の[Resources Control By]が[Manual]の場合に有効です。

[PCI/ISA PnP] ..... プラグアンドプレイ対応の PCI、または ISA カードを使用する場合

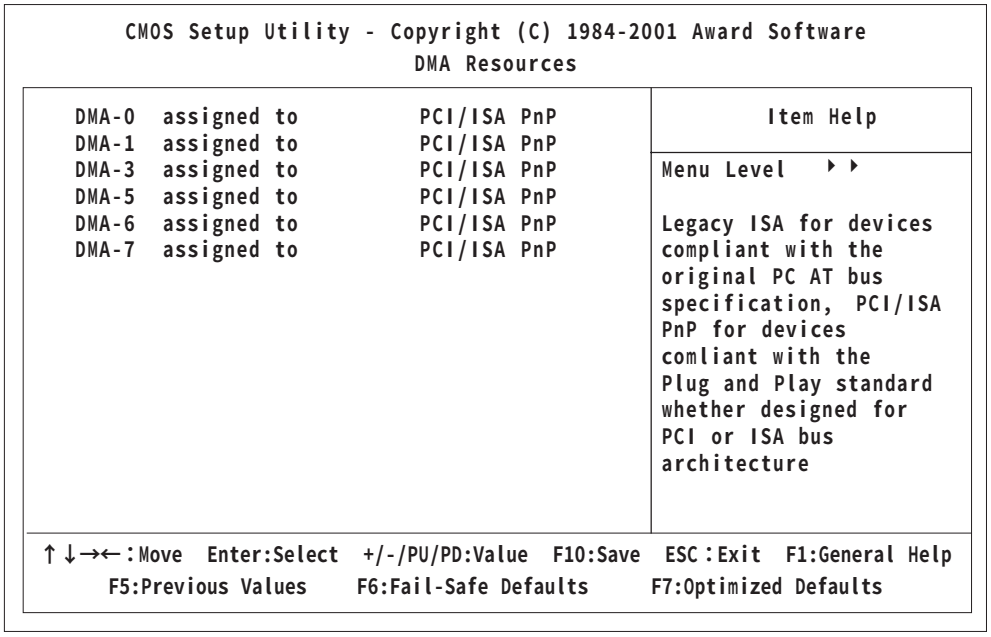
[Legacy ISA] ..... プラグアンドプレイ未対応の ISA カードを使用する場合

初期設定は以下の表のとおりです。

	初期設定		初期設定
IRQ-3 assigned to	Legacy ISA	IRQ-10 assigned to	Legacy ISA
IRQ-4 assigned to	Legacy ISA	IRQ-11 assigned to	PCI/ISA PnP
IRQ-5 assigned to	PCI/ISA PnP	IRQ-12 assigned to	Legacy ISA
IRQ-7 assigned to	Legacy ISA	IRQ-14 assigned to	Legacy ISA
IRQ-9 assigned to	Legacy ISA	IRQ-15 assigned to	PCI/ISA PnP

5.2.9 DMA Resources

PnP/PCI Configurations のメニュー画面にて DMA Resources を選択すると以下の画面が表示されます。



■ DMA-0 assigned to ～ DMA-7 assigned to

ポートアドレスに割り当てられる機器の種類を設定します。[PnP/PCI Configurations]の[Resources Control By]が[Manual]の場合に有効です。

[PCI/ISA PnP] ..... プラグアンドプレイ対応の PCI、または ISA カードを使用する場合

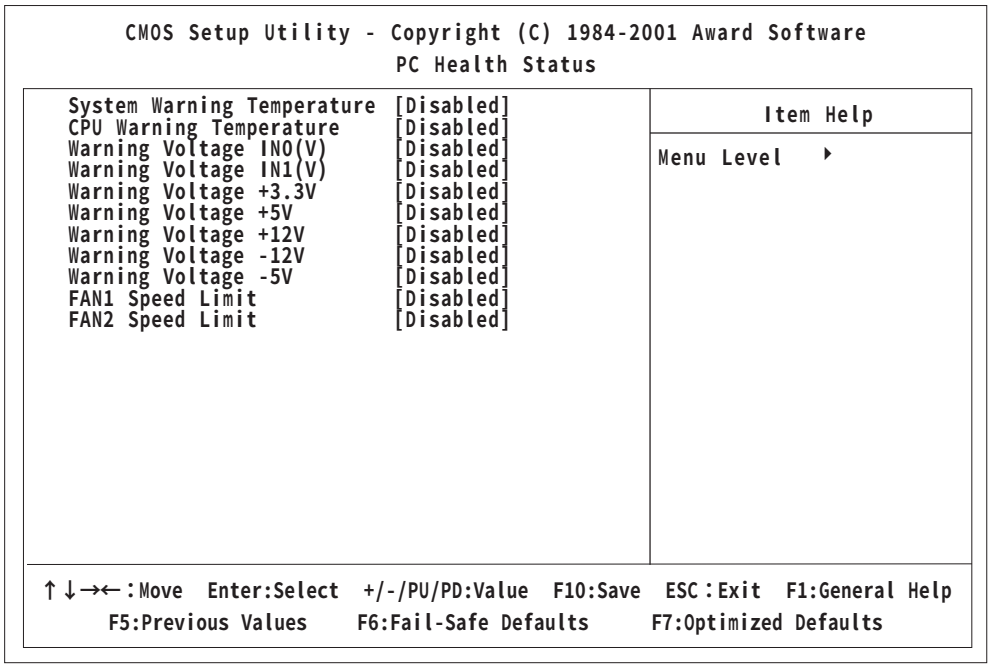
[Legacy ISA] ..... プラグアンドプレイ未対応の ISA カードを使用する場合

初期設定は以下の表のとおりです。

	初期設定		初期設定
DMA-0 assigned to	PCI/ISA PnP	DMA-5 assigned to	PCI/ISA PnP
DMA-1 assigned to	PCI/ISA PnP	DMA-6 assigned to	PCI/ISA PnP
DMA-3 assigned to	PCI/ISA PnP	DMA-7 assigned to	PCI/ISA PnP

5.2.10 PC Health Status

メニュー画面にてPC Health Statusを選択すると以下の画面が表示されます。



■ System Warning Temperature

システム全体の温度が一定値に上がるとシステムモニタに対して警告を出す設定をします。選択肢は[Disabled]or[40°C/104°F]or[45°C/113°F]or[50°C/122°F]or[55°C/131°F]or[60°C/140°F]or[65°C/149°F]or[70°C/158°F]or[75°C/167°F]or[80°C/176°F]or[85°C/185°F]です。ご使用環境に合わせて設定してください。出荷時の設定は[Disabled]です。

■ CPU Warning Temperature

CPUの温度が一定値に上がるとシステムモニタに対して警告を出す設定をします。選択肢は[Disabled]or[40°C/104°F]or[45°C/113°F]or[50°C/122°F]or[55°C/131°F]or[60°C/140°F]or[65°C/149°F]or[70°C/158°F]or[75°C/167°F]or[80°C/176°F]or[85°C/185°F]です。ご使用環境に合わせて設定してください。出荷時の設定は[Disabled]です。

重要 **・ ご使用のCPUがPentium III 1GHzの場合は[75°C/167°F]に、PentiumIII 700MHzの場合は[85°C/185°F]に設定してください。**

■ Warning Voltage IN0(V)

IN0(Vcore)の電圧が許容範囲を超えるとシステムモニタに対して警告を出す設定をします。選択肢は[Disabled]or[±6%]or[±8%]です。出荷時の設定は[Disabled]です。

■ Warning Voltage IN1(V)

IN1の電圧が許容範囲を超えるとシステムモニタに対して警告を出す設定をします。選択肢は[Disabled]or[±6%]or[±8%]です。出荷時の設定は[Disabled]です。

■ Warning Voltage +3.3V

+3.3V の電圧が許容範囲を超えるとシステムモニタに対して警告を出す設定をします。選択肢は[Disabled]or[±6%]or[±8%]です。出荷時の設定は[Disabled]です。

■ Warning Voltage +5V

+5V の電圧が許容範囲を超えるとシステムモニタに対して警告を出す設定をします。選択肢は[Disabled]or[±6%]or[±8%]です。出荷時の設定は[Disabled]です。

■ Warning Voltage +12V

+12V の電圧が許容範囲を超えるとシステムモニタに対して警告を出す設定をします。選択肢は[Disabled]or[±6%]or[±8%]です。出荷時の設定は[Disabled]です。

■ Warning Voltage -12V

-12V の電圧が許容範囲を超えるとシステムモニタに対して警告を出す設定をします。選択肢は[Disabled]or[±6%]or[±8%]です。出荷時の設定は[Disabled]です。

■ Warning Voltage -5V

-5V の電圧が許容範囲を超えるとシステムモニタに対して警告を出す設定をします。選択肢は[Disabled]or[±6%]or[±8%]です。出荷時の設定は[Disabled]です。

■ FAN1 Speed Limit

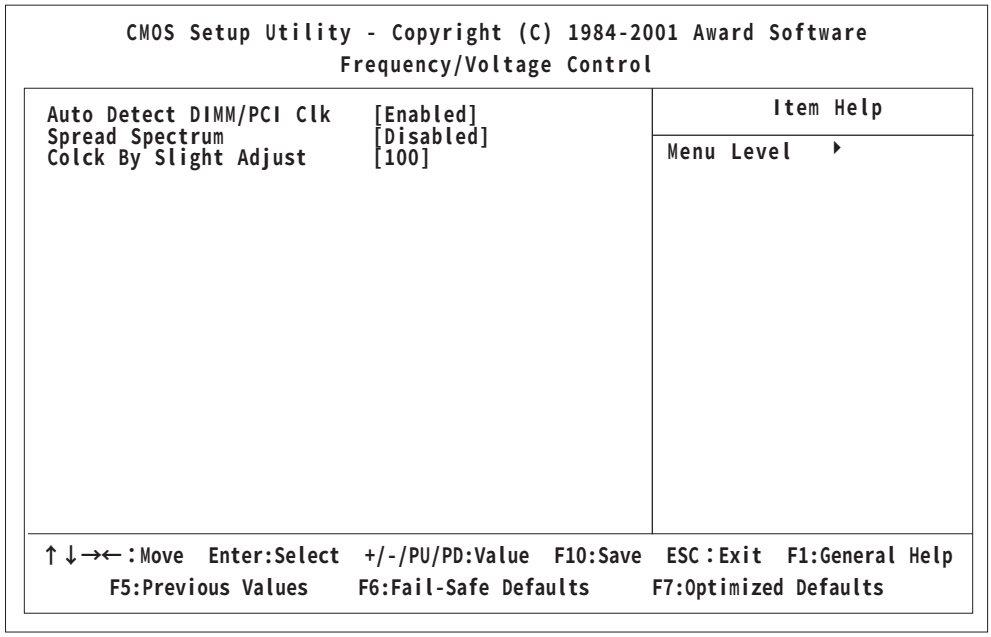
FAN1 のスピードの許容範囲を設定をします。選択肢は[Disabled]or[-30%]or[-50%]です。出荷時の設定は[Disabled]です。FAN1 は、CPU ファン用です。

■ FAN2 Speed Limit

FAN2 のスピードの許容範囲を設定をします。選択肢は[Disabled]or[-30%]or[-50%]です。出荷時の設定は[Disabled]です。FAN2 は、電源ファン用です。

5.2.11 Frequency/Voltage Control

メニュー画面にてFrequency/Voltage Controlを選択すると以下の画面が表示されます。



■ Auto Detect DIMM/PCI CLK

DIMM/PCIクロックの自動認識を設定します。[Enabled]or[Disabled]の選択です。出荷時の設定は[Enabled]です。必ず、出荷時の設定でご使用ください。

■ Spread Spectrum

クロックジェネレータのスペクトラム拡散を設定します。[Enabled]or[Disabled]の選択です。出荷時の設定は[Disabled]です。必ず、出荷時の設定でご使用ください。

■ Clock By Slight Adjust

CPUクロックを133MHzから166MHzの間、100MHzから132MHzの間、または66MHzから100MHzの間で調整することができます。設定可能な範囲は、CPU自体の周波数によります。[100]～[132]の選択です。出荷時の設定は[100]です。必ず、出荷時の設定でご使用ください。

5.2.12 Load Fail-Safe Defaults

メニュー画面でLoad Fail-Safe Defaultsを選択すると、システムが立ち上がるための最低限のシステム設定にセットアップするかどうかを設定できます。[Y]or[N]の選択となります。

5.2.13 Load Optimized Defaults

メニュー画面でLoad Optimized Defaultsを選択すると、PL出荷時の設定にするかどうかを設定します。[Y]or[N]の選択となります。

・ 出荷時の設定では、USB I/Fは使用できません。タッチパネルのUSB接続方法については、**1.1.1 タッチパネルの接続方法について**を、USB I/Fについては、**1.1.2 USB I/Fの使用について**をご参照ください。

5.2.14 Set Supervisor Password

システム情報の設定内容を変更できるパスワードです。システム情報の内容に対して変更許可のないユーザーが、システム情報を変更できないようにするためのものです。最大半角8文字で入力すると、今まで設定していたパスワードに上書きされます。

パスワードを設定しない場合は、[ENTER]を押します。[ENTER]を押すと"PASSWARD DISABLE"と表示され、パスワードが設定されていないことを確認できます。

パスワードの入力要求がどの時点で行われるかは、[Advanced BIOS Features]の[Security Option]で設定することができます。**参照→5.2.3 Advanced BIOS Features**

5.2.15 Set User Password

システム情報の設定内容を見ることができるパスワードです。システム情報の内容に対して閲覧許可のないユーザーが、システム情報を閲覧できないようにするためのものです。最大半角8文字で入力すると、今まで設定していたパスワードに上書きされます。

パスワードを設定しない場合は、[ENTER]を押します。[ENTER]を押すと "PASSWARD DISABLE" と表示され、パスワードが設定されていないことを確認できます。

パスワードの入力要求がどの時点で行われるかは、[Advanced BIOS Features]の[Security Option]で設定することができます。**参照→5.2.3 Advanced BIOS Features**

重要

- **Set Supervisor Password、またはSet User Passwordのどちらか一方のみが設定されている場合は、システム設定の閲覧、変更が可能です。**
- **Set Supervisor Password、およびSet User Passwordの両方が設定されている場合は、パスワード入力要求時にSupervisor Passwordで設定画面に入るとシステム設定の閲覧、および変更が、User Passwordで設定画面に入るとシステム設定の閲覧のみが可能です。**

5.2.16 Save & Exit Setup

セットアップユーティリティで設定した内容を保存し、PLを再起動します。

5.2.17 Exit Without Setting

セットアップユーティリティの内容を保存せずに、PLを再起動します。

MEMO

このページは、空白です。
ご自由にお使いください。

第 6 章

PL のセットアップ

(株)デジタルでは、標準の Windows®95 OSR2 以上、Windows®98 Second Edition、WindowsNT® 4.0、Windows®2000、Windows®XP ではサポートされていない PL-X920 シリーズ専用のプログラムを付属 CD-ROM にて用意しています。

6.1 付属 CD-ROM について

6.1.1 ソフトウェア構成

ここでは付属 CD-ROM に収録されているソフトウェアの種類をフォルダ構成図にて説明します。

PL-X920 Series User Manual & Driver CD		
├── [Driver]		
│ ├── [Chipset]	**Chipset ドライバ**	**< 対応 OS>**
│ │ ├── [Win95]		Windows®95
│ │ ├── [Win98]		Windows®98 Second Edition
│ │ └── [Win2000]		Windows®2000
│ ├── [Display]	**Display ドライバ**	**< 対応 OS>**
│ │ ├── [Win95]		Windows®95
│ │ │ └── [DCOM]		Windows95 用 DCOM ドライバ
│ │ ├── [Win98]		Windows®98 Second Edition
│ │ ├── [Win2000]		Windows®2000
│ │ ├── [WinNT]		WindowsNT®4.0
│ │ └── [WinXP]		Windows®XP
│ └── [LAN]	**LAN ドライバ**	**< 対応 OS>**
│ ├── [Win95]		Windows®95
│ ├── [Win98]		Windows®98 Second Edition
│ ├── [Win2000]		Windows®2000
│ └── [WinNT]		WindowsNT®4.0
├── [Manual]		**< ユーザーズマニュアル >**
│ ├── [ENG]		PDF マニュアル英語版
│ └── [JPN]		PDF マニュアル日本語版
├── [Reader]		**<PDF データリーダ >**
│ ├── [ENG]		Acrobat Reader 英語版
│ └── [JPN]		Acrobat Reader 日本語版
└── [Utility]		**<PL のユーティリティツール >**
├── blset.exe		バックライト管切れ検出機能
├── [Win95]	**ﾕｰﾃｨﾘﾃｨｾｯﾄｱｯﾌﾟ**	**< 対応 OS>**
├── [Win98]		Windows®95
├── [Win2000]		Windows®98 Second Edition
├── [WinNT]		Windows®2000
└── [WinXP]		WindowsNT®4.0
		Windows®XP

6.2 PLのセットアップ

PLには、OSなしタイプとOSプリインストールタイプの2種類があり、それぞれセットアップの手順が異なります。このマニュアルではOSなしタイプを基本に説明します。OSプリインストールタイプのセットアップは、各OSのプリインストールHDDユニットに付属の取扱説明書(以下、プリインストールタイプ取扱説明書と称します)をご参照ください。

6.2.1 OSなしタイプのセットアップ

OSなしタイプは、Windows®95/Windows®98 Second Edition/WindowsNT®4.0/Windows®2000/Windows®XPのオペレーティングシステム、PLを使用するために必要なユーティリティソフトなどをPLにインストールする必要があります。

■HDDユニット取り付け

PLにハードディスクユニットが取り付けられていない場合は、取り付けが必要です。**参照→PL-HD220取扱説明書**

■システム情報の設定

PLにハードディスクユニットが取り付けられていない場合は、システム情報の設定が必要です。システム情報を設定し、ハードディスクを正しく認識させます。**参照→PL-HD220取扱説明書**

■OSのセットアップ

PLに市販のOSをインストールします。セットアップ方法については、各OSのマニュアルをご参照ください。

重要 **・ PLで対応しているOSは以下の5種類です。それ以外のOSでの動作は保証できません。**

Windows®95 OSR2以上
Windows®98 Second Edition
WindowsNT®4.0(Service Pack 3以上)
Windows®2000
Windows®XP Professional

■PLのユーティリティセットアップ

PL本体に付属の「PL-X920 Series User Manual & Driver CD」からドライバ、およびユーティリティソフトなどをインストールする必要があります。

MEMO
- セットアップには、PS/2(ミニDIN)タイプキーボードが必要です。
- PLでタッチパネルを使用するには、マウスエミュレーションソフトウェア(UPDD)が別途必要です。マウスエミュレーションソフトウェア組み込み時、COM4を指定してインストールしてください。マウスエミュレーションソフトウェア(UPDD)は(株)デジタルのウェブサイト(http://www.proface.co.jp/otasuke/) からダウンロードしてください。

◆付属のCD-ROMからインストール

MEMO・ あらかじめCD-ROMドライブユニット(PL-DK200)をPLに接続しておいてください。

付属のCD-ROM「PL-X920 Series User Manual & Driver CD」の以下のフォルダにあるSetup.exeを起動してください。

例)Windows®98 Second Edition

D:\Utility\Win98※1\Disk1\Setup.exe (CD-ROMドライブがDの場合)

└ OSによって異なります。

◆ハードディスクの内容

PLのユーティリティセットアップを行うと、Cドライブに[Proface]フォルダが作成されます。[Proface]フォルダの構成は以下のとおりです。

[Proface]

- [69api] **API-DLL**
- [Blsaver] **バックライト消灯スクリーンセーバ**
- [Chipset] **Chipsetドライバ** (WindowsNT®4.0、Windows®XPには含まれない)
- [Disp] **表示On/Offユーティリティ**
- [Display] **グラフィックアクセラレータドライバ**
- [Keyclick] **キーボードエミュレータ**
- [Lan] **LANドライバ** (Windows®XPには含まれない)
- [Sysmon] **システムモニタ/RASアプリケーション**

MEMO
- 使用するOSによって、上記の内容と異なる場合があります。
- PLのユーティリティセットアップを行うと、以下の制御用ドライバは、自動的にシステムフォルダにコピーされます。
  - PLSYSMON.VXD ハードウエア制御用ドライバ (Windows® 95、Windows® 98 Second Edition)
  - PLSYSMON.SYS ハードウエア制御用ドライバ (WindowsNT® 4.0、Windows® 2000、Windows® XP)
  - BLCTRL.VXD バックライト制御用ドライバ (Windows® 95、Windows® 98 Second Edition)
  - BLCTRL.SYS バックライト制御用ドライバ (WindowsNT® 4.0、Windows® 2000、Windows® XP)

※1 OSがWindows®95の場合は「Win95」、Windows®2000の場合は「Win2000」、WindowsNT®4.0の場合は「WinNT」、Windows®XPの場合は[WinXP]を入力してください。

6.2.2 OSプリインストールタイプのセットアップ

OSプリインストールタイプには、Windows®95/Windows®2000/WindowsNT®4.0/Windows® XPのオペレーティングシステムの他に、PLを使用するために必要なドライバ、およびユーティリティソフトなどがあらかじめPLにインストールされています。

- プリインストールされているOSは、PLシリーズ専用のものです。
- Windows®NT 4.0のOSプリインストールタイプには、Service Pack 6aが入っています。WindowsNT®4.0のシステム構成を変更した場合は、Service Pack再セットアップが必要です。
- Windows®2000のOSプリインストールタイプには、Service Pack 4が入っています。
- Windows®XPのOSプリインストールタイプには、Service Pack 2が入っています。

■HDDユニット取り付け

PLにハードディスクユニットが取り付けられていない場合は、取り付けが必要です。**参照→プリインストールタイプ取扱説明書**

■システム情報の設定

PLにハードディスクユニットが取り付けられていない場合は、PLにシステム情報の設定が必要です。システム情報を設定し、ハードディスクを正しく認識させます。**参照→プリインストールタイプ取扱説明書**

■ OS プリインストールタイプのセットアップ

PLにプリインストールされているOSをセットアップします。セットアップ方法については、プリインストールタイプ取扱説明書をご参照ください。

◆ハードディスクの内容

OSプリインストールタイプHDのCドライブに[Proface]フォルダが用意されています。
[Proface]フォルダの構成は以下のとおりです。

[Proface]

[69api]	**API-DLL**
[Blsaver]	**バックライト消灯スクリーンセーバ**
[Chipset]	**Chipset ドライバ** (WindowsNT® 4.0、Windows® XP には含まれない)
[Disp]	**表示 On/Off ユーティリティ**
[Display]	**グラフィックアクセラレータドライバ**
[Keyclick]	**キーボードエミュレータ**
[Lan]	**LAN ドライバ** (Windows® XP には含まれない)
[Setup]	**セットアップ** (Windows®95には含まれない)
[Update]	**Windows® 95 用のアップデートモジュール**(Windows®95 のみ)
[Sysmon]	**システムモニタ/RAS アプリケーション**
[Updd]	**マウスエミュレータ** (PL-B920シリーズには含まれない)

MEMO
- 使用するOSによって、上記の内容と異なる場合があります。
- PLのハードディスクのシステムフォルダには、以下の制御用ドライバが用意されています。

  PLSYSMON.VXD　ハードウエア制御用ドライバ(Windows® 95)
  PLSYSMON.SYS　ハードウエア制御用ドライバ
  (WindowsNT® 4.0、Windows® 2000、Windows® XP)
  BLCTRL.VXD　バックライト制御用ドライバ(Windows® 95)
  BLCTRL.SYS　バックライト制御用ドライバ
  (WindowsNT® 4.0、Windows® 2000、Windows® XP)

6.3 ドライバの組み込み

PL専用のドライバとして、CHIPSET、グラフィックアクセラレータ、LANの3種類を用意しています。

OSなしタイプやリカバリーメディアで修復させたPLには、各ドライバを必要に応じて組み込んでください。

OSプリインストールタイプの場合も、LANドライバは組み込まれていないため、必要に応じて組み込んでください。

ここではあらかじめPLのハードディスクに[Proface]フォルダがあるものとして説明しています。

重要 **・組み込んだドライバは、アンインストールできません。**

■ Chipset ドライバの組み込み

PL-X920シリーズにChipsetドライバを組み込みます。組み込むことでOSに対してハードディスクを認識させます。ただし、Windows NT®4.0、Windows® XPには対応していません。

①ご使用のOSに応じて以下のファイルを起動します。画面の指示に従って進めてください。

C:\Proface\Chipset\infinst_enu.exe (Windows® 95)

C:\Proface\Chipset\infinst_autol.exe (Windows® 98, Windows® 2000)

■グラフィックアクセラレータドライバの組み込み

PL-X920シリーズにグラフィックアクセラレータドライバを組み込みます。組み込むことでWindows®の画面表示を専用ハードウエアの機能で高速化します。

◆OSがWindows® 95の場合の手順

①ドライバを組み込む前に以下のファイルを起動します。

C:\Proface\Display\Dcom\Dcom95.exe

②C:\Proface\Display\win9xm66.exeを起動します。画面の指示に従って進めてください。

◆OSがWindows® 98 Second Edition、Windows NT®4.0、 Windows® 2000、Windows® XPの場合の手順

①ご使用のOSによって、以下のファイルを起動し、画面の指示に従って進めてください。

C:\Proface\Display\win9xm67.exe (Windows® 98)

C:\Proface\Display\winnt4m67.exe (WindowsNT® 4.0)

C:\Proface\Display\win2k_xpm67.exe (Windows® 2000, Windows® XP)

MEMO ・ OSがWindows NT®4.0の場合、Windows NT®4.0 Service Pack 3以上がインストールされているか確認してください。

■ LAN ドライバの組み込み

PL に LAN ドライバを組み込みます。組み込むことで LAN が使用できます。ご使用の OS や、OS なしタイプや OS プリインストールタイプによって手順が異なりますので、ご使用の PL に合わせて組み込みを行ってください。

◆ LAN ドライバを組み込む前に

① システム情報の設定の[Integrated Peripherals]の[Onboad LAN]の設定を[Enabled]に変更してください。**参照→5.2.5 Integrated Peripherals**

② OS が起動します。Windows® 95、Windows® 98 の場合は、OS が起動した後にデバイスドライバウィザードが表示されます。

・ Window®XP をご使用の場合は、システム情報の設定変更後、OS 起動時に自動的に LAN ドライバが組み込まれます。

◆ OS が Windows® 95 の場合の手順

・OS なしタイプ

・ OS なしタイプでは、手順の前に PL に CD-ROM ドライブユニット(PL-DK200)を接続し、ご使用の OS の CD-ROM を挿入しておいてください。

デバイスドライバウィザードが表示されます。

1 **[次へ]をクリックします。**

2 **[場所の指定(O)]をクリックします。**
場所の指定ウィザードが表示されます。

3 **[C:\Proface\Lan] と入力し、[OK] をクリックします。**
場所の指定ウィザードが表示されます。

4 **もう一度[C:\Proface\Lan] と入力し、[OK] をクリックします。**
デバイスドライバウィザードが表示されます。

5 **[完了] をクリックします。**
ファイルのコピーのダイアログが表示されます。

6 **[D:\Win95]と入力し、[OK] をクリックします。**
Windows® 95 の CD-ROM から PL にファイルがコピーされます。

7 **[はい(Y)]をクリックし、PL を再起動すれば終了です。**

・OS プリインストールタイプ

デバイスドライバウィザードが表示されます。

1 **[次へ]をクリックします。**

2 **[場所の指定(O)]をクリックします。**
場所の指定ウィザードが表示されます。

3 [C:\Proface\Lan] と入力し、[OK] をクリックします。
場所の指定ウィザードが表示されます。

4 もう一度[C:\Proface\Lan] と入力し、[OK] をクリックします。
デバイスドライバウィザードが表示されます。

5 [完了]をクリックします。
ファイルのコピーのダイアログが表示されます。

6 [C:\Windows\Options\Cabs] と入力し、[OK] をクリックします。
プリインストールされている Windows® 95 からファイルがコピーされます。

7 **[はい(Y)]をクリックし、PL を再起動すれば終了です。**

◆OS が Windows® 98 Second Edition の場合の手順

Windows® 98 Second Edition には、OS プリインストールタイプはありません。

・OS なしタイプ

・ OS なしタイプでは、手順の前に PL に CD-ROM ドライブユニット(PL-DK200)を接続し、ご使用の OS の CD-ROM を挿入しておいてください。

新しいハードウェアの追加のウィザードが表示されます。

1 **[次へ(N)]をクリックします。**

2 **[使用中のデバイスに最適なドライバを検索する(推奨)]を選択し、[次へ]をクリックします。**

3 **検索場所の指定(L)に[C:\Proface\Lan]と入力し、[次へ]をクリックします。**

4 **[次へ]をクリックします。**
ファイルのコピーダイアログが表示され、Windows® 98 の CD-ROM から PL にファイルがコピーされます。

5 **[完了]をクリックします。**
システム設定のダイアログが表示されます。

6 **[はい(Y)]をクリックして、PL を再起動すれば終了です。**

◆OSがWindows NT®4.0の場合の手順

・OSなしタイプ

・ OSなしタイプでは、手順の前にPLにCD-ROMドライブユニット(PL-DK200)を接続し、ご使用のOSのCD-ROMを挿入しておいてください。

[スタート]ボタンをクリックし、[設定(S)]をポイントし、[コントロールパネル(C)]をクリックします。

1 **[コントロールパネル]の[ネットワーク]をダブルクリックします。**
ネットワーク構成のダイアログが表示されます。

2 **[はい(Y)] をクリックします。**
ネットワークセットアップウィザードが表示されます。

3 **[ネットワーク接続(W)]を選択し、[次へ(N)] をクリックします。**

4 **[一覧から選択(S)] をクリックします。**
ネットワークアダプタの選択のダイアログが表示されます。

5 **[ディスク使用(H)] をクリックします。**
フロッピーディスクの挿入のダイアログが表示されます。

6 **[C:\Proface\Lan]と入力し、[OK] をクリックします。**
OEMオプションの選択のダイアログが表示されます。

7 **[OK] をクリックします。**
ネットワークセットアップウィザードが表示されます。

8 **[次へ(N)] をクリックします。**

9 **使用するネットワークプロトコルを選択して[次へ(N)] をクリックします。**

10 **インストールされるサービスを選択して、[次へ(N)] をクリックします。**

11 **[次へ(N)] をクリックします。**
WindowsNTのセットアップのダイアログが表示されます。

12 **[D:\i386]と入力し、[続行] をクリックします。**

13 **[C:\Proface\Lan]と入力し、[続行] をクリックします。**
Speed Duplex modeのダイアログが表示されます。

14 **[Continue]をクリックします。**
Input Network Addressのダイアログが表示されます。

15 **[OK]をクリックします。**
Input Tx Early Thresholdのダイアログが表示されます。

16 **[OK]をクリックします。**
TCP/IPのダイアログが表示されます。

17 **お客様のネットワークの設定状況に合わせて設定してください。**
ネットワークセットアップウィザードが表示されます。

18 **[次へ(N)]をクリックします。**

19 **[次へ(N)]をクリックします。**

20 **コンピュータ名とワークグループ名を入力し、[次へ(N)]をクリックします。**

21 **[完了]をクリックします。**
ネットワーク設定の変更のダイアログが表示されます。

22 **[はい(Y)]をクリックすると、PLが再起動されます。**
サービスコントロールマネージャーのダイアログが表示されます。

23 **再起動すると、エラーメッセージが出ますので、お客様がインストールしたサービスパックを再インストールします。**

24 **再インストールが完了すれば、PLを再起動して終了です。**

・OSプリインストールタイプ

[スタート]ボタンをクリックし、[設定(S)]をポイントし、[コントロールパネル(C)]をクリックします。

1 **[コントロールパネル]の[ネットワーク]をダブルクリックします。**
ネットワーク構成のダイアログが表示されます。

2 **[はい(Y)] をクリックします。**
ネットワークセットアップウィザードが表示されます。

3 **[ネットワーク接続(W)]を選択し、[次へ(N)] をクリックします。**

4 **[一覧から選択(S)] をクリックします。**
ネットワークアダプタの選択のダイアログが表示されます。

5 **[ディスク使用(H)] をクリックします。**
フロッピーディスクの挿入のダイアログが表示されます。

6 **[C:\Proface\Lan]と入力し、[OK] をクリックします。**
OEMオプションの選択のダイアログが表示されます。

7 **[OK] をクリックします。**
ネットワークセットアップウィザードが表示されます。

8 **[次へ(N)] をクリックします。**

9 **使用するネットワークプロトコルを選択して[次へ(N)] をクリックします。**

10 **インストールされるサービスを選択して、[次へ(N)] をクリックします。**

11 **[次へ(N)] をクリックします。**
WindowsNTのセットアップのダイアログが表示されます。

12 **[C:\Proface\Setup]と入力し、[続行] をクリックします。**

13 **[C:\Proface\Lan]と入力し、[続行] をクリックします。**
Speed Duplex modeのダイアログが表示されます。

14 **[Continue]をクリックします。**
Input Network Addressのダイアログが表示されます。

15 **[OK]をクリックします。**
Input Tx Early Thresholdのダイアログが表示されます。

16 **[OK]をクリックします。**
TCP/IPのダイアログが表示されます。

17 **お客様のネットワークの設定状況に合わせて設定してください。**
ネットワークセットアップウィザードが表示されます。

18 **[次へ(N)]をクリックします。**

19 **[次へ(N)]をクリックします。**

20 **コンピュータ名とワークグループ名を入力し、[次へ(N)]をクリックします。**

21 **[完了]をクリックします。**
ネットワーク設定の変更のダイアログが表示されます。

22 **[はい(Y)]をクリックすると、PLが再起動されます。**
サービスコントロールマネージャーのダイアログが表示されます。

23 **エクスプローラを開き[C:\Proface\Setup\Sp6\i386\Update\Update.exe]をダブルクリックで起動します。**
Windows NT Service Pack セットアップダイアログが表示されます。

24 **[同意する(A)]を選択し、[インストール(I)]をクリックします。**
Service Pack 6aのセットアップが開始されます。

25 **[再起動(R)]をクリックし、PLを再起動すれば終了です。**

◆OSがWindows® 2000の場合の手順

Windows® 2000は、OSなしタイプとOSプリインストールタイプの手順が共通です。

・OSなしタイプ/OSプリインストールタイプ

[スタート]ボタンをクリックし、[設定(S)]をポイントし、[コントロールパネル(C)]をクリックします。

1 **[コントロールパネル]の[ネットワークとダイヤルアップ接続]をダブルクリックします。**
ネットワークとダイヤルアップ接続のウィンドウが表示されます。

2 **[ローカルエリア接続]を右クリック**し、**[プロパティ]を開きます。**
ローカルエリア接続プロパティが表示されます。

3 **[構成(C)] をクリックします。**
Realtek RTL8139(A) PCI Fast Ethernet Adapterのプロパティが表示されます。

4 **[ドライバの更新(P)] をクリックします。**
デバイスドライバのアップグレードウィザードが表示されます。

5 **[次へ(N)] をクリックします。**

6 **[デバイスに最適なドライバを検索する(推奨)(S)]を選択し、[次へ(N)] をクリックします。**

7 **[場所を指定(S)]を選択し、[次へ(N)] をクリックします。**

8 **[C:\Proface\Lan]と入力し、[OK] をクリックします。**
ドライバの検索が始まります。

9 **[次へ(N)] をクリックします。**

10 **[完了] を選択し、PLを再起動して終了です。**

6.4 アプリケーション機能

PL専用の機能としてプログラムを用意しています。ここではあらかじめPLのハードディスクに[Proface]フォルダがあるものとして、ファイルの格納されている場所を下記の表に示します。

ファイル名	Windows®95/Windows®98 Second Edition	Windows NT®4.0/ Windows®2000	Windows®XP
PL_BLIOC.DLL	C:\Windows\System	C:\Winnt\System32	C:\Windows\System32
PL_DLL.DLL			
PL_IOC.DLL			
Backlight Control.scr			
Disp.exe	C:\Proface\Disp		
Keyclick.exe	C:\Proface\Keyclick		
PL_Smon.exe	C:\Proface\Sysmon		
PL_Wps.exe	C:\Proface\Sysmon		
BLSET.EXE	付属のCD-ROM「PL-X920 Series User Manual & Driver CD」内のUtility\Blset.exe		

■ API-DLL

PL上で動作するRAS機能を、お客様が作成したアプリケーションから利用するためのダイナミックリンクライブラリです。API-DLLには、以下の3種類を用意しています。

◆バックライトコントロール PL_BLIOC.DLL

このPL_BLIOC.DLLは、PL上で動作するバックライトコントロール機能をユーザーが作成したアプリケーションから利用するためのダイナミックリンクライブラリです。詳しくは「付.4　バックライトコントロール機能API-DLL」を参照してください。

◆システムモニタ PL_DLL.DLL

このPL_DLL.DLLは、PL上で動作するシステムモニタ機能を利用するためのダイナミックリンクライブラリです。

◆RAS機能 PL_IOC.DLL

このPL_IOC.DLLは、PL上で動作するRAS機能を、お客様が作成したアプリケーションから利用するためのダイナミックリンクライブラリです。

詳しくは「付.3　システムモニタ/RAS機能API-DLL」を参照してください。

■バックライト消灯スクリーンセーバ Backlight Control.scr

設定時間オペレーションがない場合、バックライトを消灯することによって寿命を延ばします。このプログラムはWindows®で実行します。

MEMO

・ 実行中のアプリケーションによっては設定時間になってもバックライトが消灯しない場合があります。ご使用のアプリケーションで動作を確認してからご使用ください。

■表示On/Offユーティリティ Disp.exe

バックライト表示をOn/Offするコマンドラインユーティリティです。このプログラムはコマンドプロンプトで動作します。

起動方法　　**DISP ON** ↵ または **DISP OFF** ↵

オプションスイッチ　ON:表示 /OFF:非表示

リターン値　　0:正常終了 /-1:オプションスイッチエラー

重要
- **OSがWindows®でバックライト表示を連続してOn/Offするアプリケーションを作成する場合は、バックライトコントロール PL_BLIOC.DLL をお使いください。**

■キーボードエミュレータ Keyclick.exe

マウスオペレーションでキーボード入力をサポートします。Windows®で実行すると、PLの画面上にキーボードが表示されます。

IN-fINITY soft製 Keyclick32使用許諾書に同意が必要です。**参照→付 .6 使用許諾書**

MEMO
- 実行中のアプリケーションによってはキー入力できない場合がありますのでご使用のアプリケーションで動作を確認してからご使用ください。
- Windows®起動時のユーザー名、パスワードの入力はできません。
- Keyclickのフォントポイント変更にはキーボードが必要です。
- 使用方法の詳細は、画面キーボードの[HELP]ボタンをクリックしオンラインヘルプを参照してください。

■システムモニタ /RASアプリケーション　Pl_Smon.exe　/　Pl_Wps.exe

RAS機能、システムモニタ機能を使用し、温度や電圧、ファンの異常を監視することができます。このプログラムはWindows®で実行します。

◆システムモニタプログラム　Pl_Smon.exe

詳しくは**付3.3 システムモニタの動作**をご参照ください。

◆監視パラメータ設定用プログラム　Pl_Wps.exe

詳しくは**付3.2 システムモニタプロパティの設定**をご参照ください。

■ バックライト管切れ検出機能 BLSET.EXE (PL-6920シリーズのみ)

バックライト切れた時のタッチパネル操作を有効にするか無効にするかを設定します。このプログラムは「PL-X920 Series User Manual & Driver CD」の[Utility]フォルダに格納されています。FDまたはPLのハードディスクにコピーし、使用します。このプログラムはMS-DOS®で動作します。

起動方法　　**BLSET ON** ↵ または **BLSET OFF** ↵

「ON」でバックライトが切れた時にタッチパネル操作を無効にし、
「OFF」でバックライトが切れた時にタッチパネル操作を有効にします。
出荷時の設定は「OFF」です。検出機能の詳細については**付 .2 RAS機能について**をご参照ください。

6.4.1 アンインストール

PLのユーティリティソフトをアンインストールします。

①[コントロールパネル]をクリックします。

②[アプリケーションの追加と削除]で[PL-X920 Driver and Utility]を選択し、削除します。

重要 **・組み込んだドライバは、アンインストールできません。**

6.5 Windows NT®4.0、Windows® 2000、Windows® XP 使用時の注意

OSがWindows NT®4.0、Windows®2000、またはWindows®XPの場合は、必要に応じて以下の設定を行ってください。

6.5.1 システムへの自動ログオンの設定方法

Windows®を起動したときに出るパスワード入力を省略して、Windows®を起動する設定です。

■Windows NT®4.0の場合

① [スタート]メニューから[ファイル名を指定して実行(R)]を選択し、以下のコマンド(レジストリエディタ)を実行します。

C:\WINNT\REGEDIT.EXE

② 次のサブキーを選択します。

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon

③ [DefaultUserName]に自動ログオンするユーザー名を設定します。

④ レジストリエディタの[編集(E)]メニューの[新規作成(N)]で[文字列(S)]を選択します。

⑤ データ型が文字列のエントリ[AutoAdminLogon]を追加し、[値のデータ(V)]に1を設定します。

⑥ データ型が文字列のエントリ[DefaultPassword]を追加し、[DefaultUserName]に設定されたユーザーのパスワードを設定します。

重要 **・ パスワードなしのユーザーの場合、自動ログオンできません。**

⑦ レジストリエディタを終了します。

MEMO

- 自動ログオンするユーザーが「Administrators」グループに所属していない場合に、自動ログオンを設定するとレジストリ編集による自動ログオンの解除ができなくなってしまいます。その場合[Shift]キーを押しながらログオフすると、[ログオン情報]ダイアログボックスが表示され、他の管理者権限を持つユーザーでログオンし直すことができます。
- 自動ログオンの設定をしない場合は、ログオン時にPS/2(ミニDIN)タイプキーボードが必要です。

■ Windows® 2000 の場合

① コントロールパネルの[ユーザーとパスワード]を起動します。

② 自動ログオンするユーザーを選択し、[このコンピュータを使うには、ユーザー名とパスワードを入力する必要があります(E)]チェックボックスのチェックを解除した状態にします。

③ [詳細]タブをクリックし、[ユーザーがログオンする前に必ずCtrl+Alt+Delキーを押す(R)]チェックボックスのチェックを解除した状態にします。

④ [適用(A)]ボタンを押すと自動ログオンのダイアログボックスが表示されるので、パスワードを入力します。

■ Windows® XP の場合

① [スタート]メニューから[ファイル名を指定して実行(R)]を選択します。

② "Control userpasswords2" と入力し、OKボタンをクリックします。

③ [ユーザーがこのコンピュータを使うには、ユーザー名とパスワードの入力が必要(E)]チェックボックスのチェックを解除した状態にし、適用ボタンをクリックします。

④ 自動ログイン設定するユーザー名とパスワードを入力し、OKボタンをクリックします。

6.5.2 無停電電源装置について

Windows NT®4.0、Windows® 2000、またはWindows® XPはシステムの電源を切る前にシャットダウンを行う必要があります。突然の電源障害からデータを守るために無停電電源装置の使用をお勧めします。

Windows® に対応した無停電電源装置を使用すると、電源障害発生時にバックアップ用電源に切り替わり安全にシャットダウンするまでの時間を確保したり、自動的にWindows®をシャットダウンすることができます。

詳細については、無停電電源装置の販売元にお問い合わせください。

6.5.3 システム構成を変更する場合

LANやプリンタを増設した場合、Windows®のシステム構成を変更する必要があります。

■ Windows NT®4.0 の場合

◆システム構成の変更

Windows®のシステム構成を変更する場合、次のメッセージが表示されます。

> Windows NT セットアップ
> いくつかのWindows NT ファイルをコピーする必要があります。セットアップは、次の場所でファイルを検索します。ほかの場所を検索させた場合は、新しい場所を入力し、[続行]をクリックしてください。

> ファイルが必要
> Windows NT Workstation CD-ROM上の一部のファイルが必要です。Windows NT Workstation CD-ROMを指定したドライブに入れて、[OK]をクリックしてください。

いずれのメッセージが表示された場合も新しい場所としてシステム構成を変更するフォルダを入力して[続行]をクリックしてください。

・OSなしタイプの場合

Windows NT® 4.0のCD-ROM内の[I386]のフォルダを指定します。

D:\I386 (CD-ROMドライブがDの場合)

・OSプリインストールタイプの場合

C:\Proface\Setup\I386

◆ Service Packの再セットアップ

Windows NT®のシステム構成を変更した場合は、システムファイルがService Pack1の古いファイルに上書きされてしまいます。必ず以下の手順でService Packの再セットアップを行ってください。

・OSなしタイプの場合

お客様がインストールしたService Packの再セットアップを行ってください。

・OSプリインストールタイプの場合

C:\Proface\Setup\Sp6\I386\Update\Update.exe

MEMO

・ Windows NT®のシステム構成を変更したあと、Service Pack 6aの再セットアップを行わなかった場合、システムは正常に動作しません。

■ Windows® 2000/Windows® XP の場合

◆システム構成の変更

Windows® のシステム構成を変更する場合、次のメッセージが表示されます。
新しい場所としてシステム構成を変更するフォルダを入力して[続行]をクリックしてください。

'Windows ○○※1 Professional CD-ROM' のラベルの付いた CD を CD-ROM ドライブ (D:)に挿入して、[OK]をクリックしてください。 フロッピーディスクやネットワークサーバなど、別の場所からファイルをコピーする場合も、[OK]をクリックしてください。

・OS なしタイプの場合

Windows® の CD-ROM 内の[I386]のフォルダを指定します

D:\I386　(CD-ROM ドライブが D の場合)

・OS プリインストールタイプの場合

C:\Proface\Setup\I386

6.5.4 NTFS ファイルシステムへの変換方法

■ Windows NT®4.0/Windows® 2000/Windows® XP の場合

OS プリインストールタイプの場合や、ご使用の OS が、Windows® の DOS 互換ファイルシステム(FAT32)でフォーマットされている場合は、NTFS ファイルシステムへ変換することができます。

NTFS ファイルシステムへの変換は、Windows® を起動させ、コマンドプロンプトを使用してください。

convert X: /fs:ntfs(X にはドライブ名を入力します)

MEMO ・ NTFS ファイルシステムへ変換してしまうと、Windows® の DOS 互換ファイルシステム(FAT32)へは戻すことはできません。

※1 ○○は OS により異なります。

例)Windows®XP の場合：'Windows XP Professional CD-ROM'

MEMO

このページは、空白です。
ご自由にお使いください。

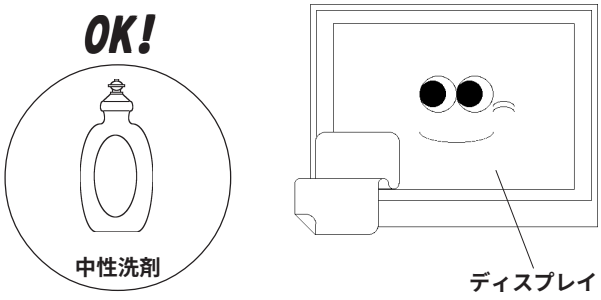
第 7 章

保守と点検

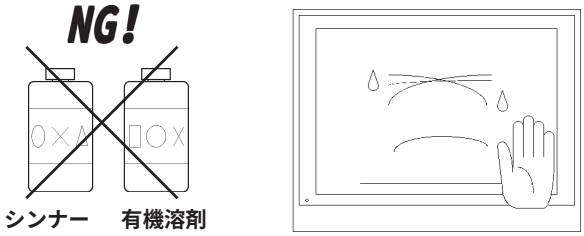
PLを快適に使用するための注意や点検基準を説明しています。

7.1 通常の手入れ

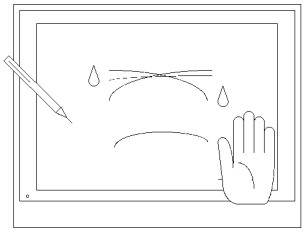
7.1.1 ディスプレイの手入れ



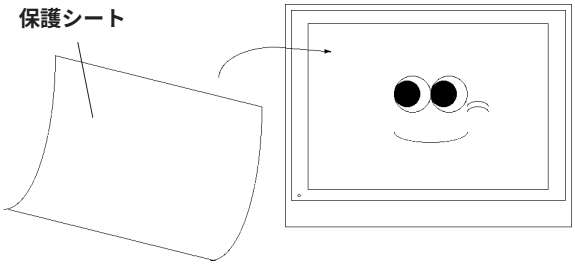
ディスプレイの表面、およびフレームが汚れた時には、柔らかい布に水でうすめた中性洗剤をしみこませて固く絞り、ディスプレイの表面やフレームの汚れを拭き取ります。



シンナー、有機溶剤、強酸系などは使用しないでください。



シャープペンシルなどの先が鋭利なもので画面に触れないでください。キズの原因になります。



表示面がすぐに汚れるような場所でご使用になる場合には、保護シートをご利用ください。

7.1.2 防滴パッキンについて

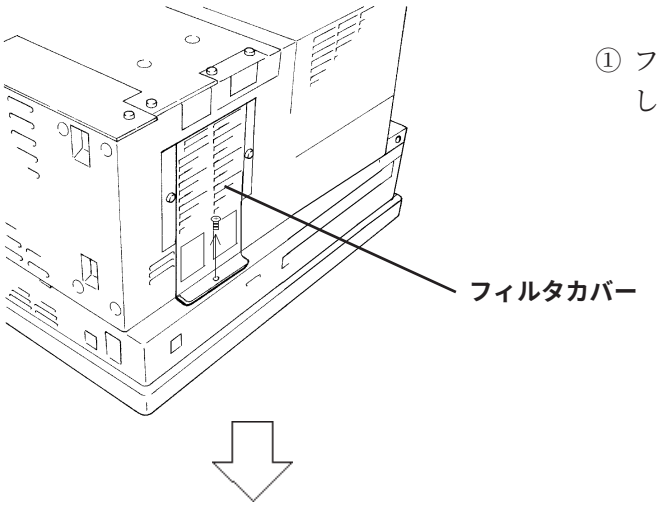
防滴パッキンは、防塵・防滴効果を得るために使います。防滴パッキンの取り付け方法は、**4.1.2取り付け手順**を参照してください。

重要 **・ 長期間使用した防滴パッキンはキズや汚れがつき防塵･防滴効果が得られない場合があります。定期的（キズや汚れが目立ってきた場合）に交換してください。**

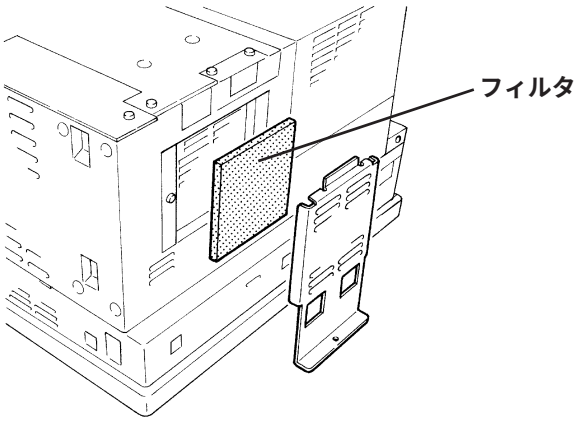
7.2 ファンフィルタの清掃方法

PLには、本体の冷却のため電源ファンを使用しておりますが、そのファンフィルタが汚れますと本来の機能を十分に発揮できませんので、定期的にフィルタのチェックおよび清掃を行ってください。

■ PL-6920/PL-7920（4スロットタイプ）の場合

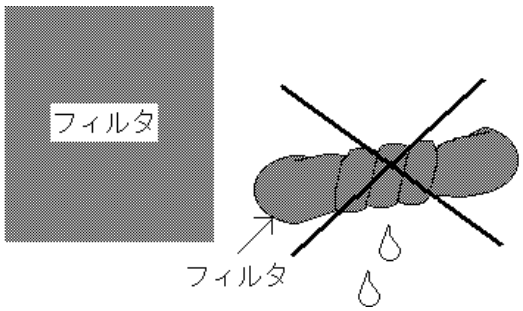


① フィルタカバーのネジ(1ヵ所)を取り外し、フィルタカバーを取り外します。



② フィルタを取り外します。

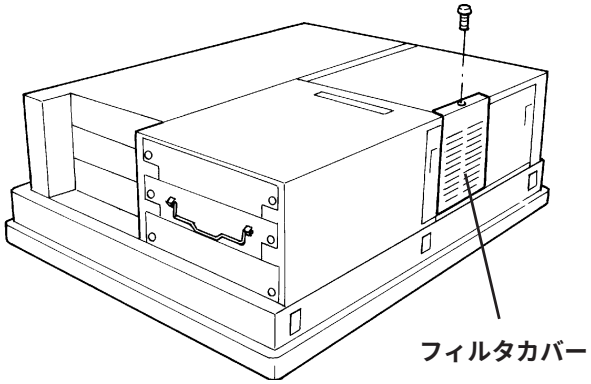
③ フィルタに付いたゴミは、掃除機などで除去してください。汚れがひどい場合は、中性洗剤で水洗いしてください。



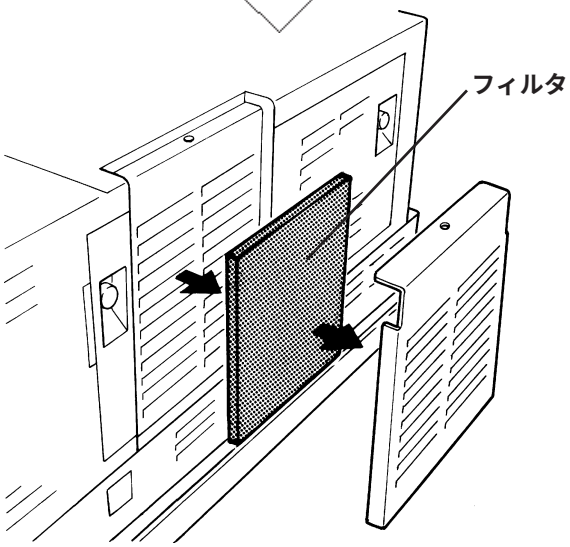
重要
- **フィルタはねじらないでください。**
- **フィルタを乾かす場合は、直射日光を避け陰干ししてください。**
- **フィルタは、十分乾いた後に取り付けてくだい。**

④ フィルタをセットし、 本体にファンカバーをネジ1本で取り付けます。

■ PL-6921/PL-7921(2 スロットタイプ)の場合

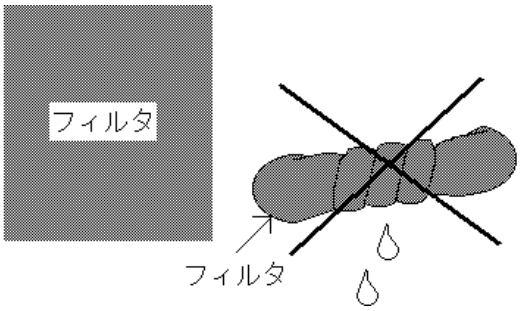


① フィルタカバーのネジ(1ヵ所)を取り外し、フィルタカバーを取り外します。



② フィルタを取り外します。

③ フィルタに付いたゴミは、掃除機などで除去してください。汚れがひどい場合は、中性洗剤で水洗いしてください。



重要

- **フィルタはねじらないでください。**
- **フィルタを乾かす場合は、直射日光を避け陰干ししてください。**
- **フィルタは、十分乾いた後に取り付けてくだい。**

④ フィルタをセットし、本体にフィルタカバーをネジ1本で取り付けます。

7.3 バックライトの交換方法

PL-6920/PL-7920 シリーズでは、ユーザーでのバックライト（ランプ部分）交換が可能です。

以下に交換方法を説明します。

MEMO ・ご使用の PL とバックライトの型式が適合しているかご確認ください。

PL	バックライトの型式
PL-6920	PL6920-BL00
PL-6921	
PL-7920	PL7900-BL00-MS
PL-7921	

重要 **・ バックライトまたは表示ユニットが故障した場合、表示画面が消えます。画面が消えていても、タッチパネルは、正常に動作している可能性があります。このような状態でタッチ操作を行うと意図しない結果を招き、危険を伴いますのでお避けください。**

⚠ 警 告

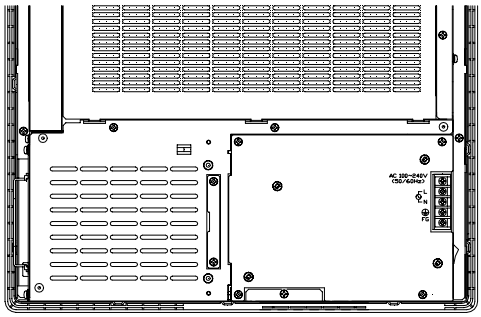
【感電】
- **作業を始める前に、PLの電源を切っておいてください。**
- **バックライトには高電圧がかかっています。PLの電源が入った状態では絶対にバックライトの交換作業を行わないでください。**

【ヤケド】
- **バックライトは、点灯中熱くなります。ヤケドの恐れがありますので、点灯中および消灯直後のバックライトやその周辺にはふれないでください。作業の際には、必ず手袋を着用してください。**

【ガラス】
- **バックライトは細いガラス管でできています。強い力がかかると、割れることがあり危険です。取り外し・取り付け時に、強い力で引っ張ったり押し込んだりしないようご注意ください。**

以下の手順にしたがってください。作業は必ず手袋を着用してください。PLが組み込まれている機器から外し、表示面を下にして作業してください。

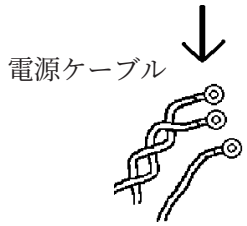
■ PL-6920 シリーズの場合



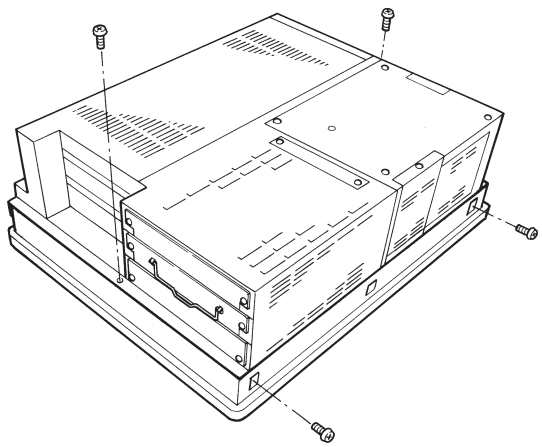
（PL-6921 をモデルとしています）

① PL の電源を切ってください。また、電源ケーブルに電源が供給されていないことを確認してください。感電のおそれがあります。

② 電源ケーブルを取り外します。

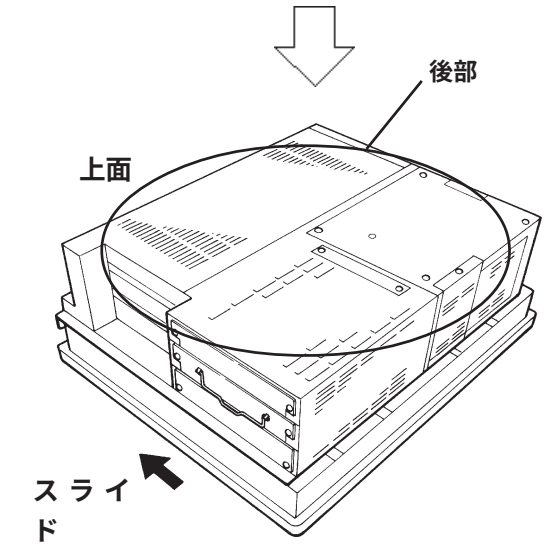


重要 **・ 作業は平らな場所で行ってください。不安定な場所での作業はケーブルの断線や PL の破損につながります。**

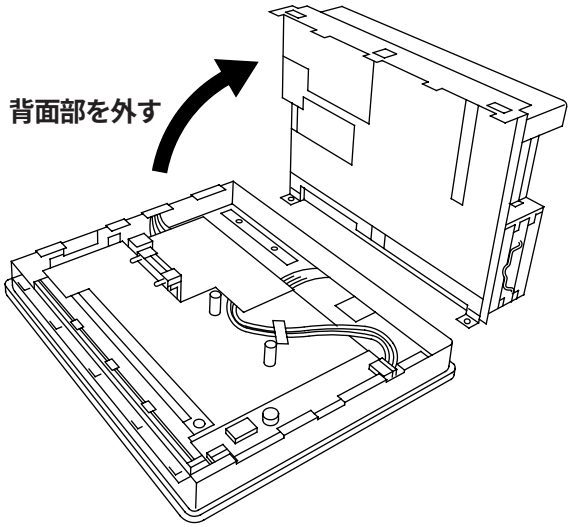


③ 本体にあるネジ(4ヵ所)をドライバではずします。

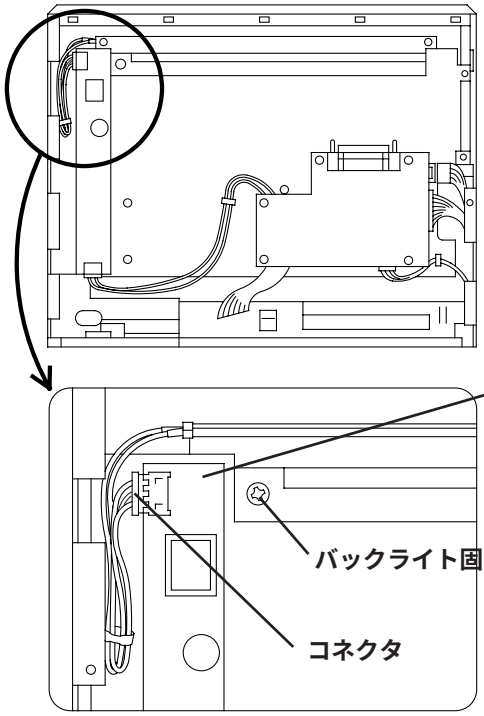
重要 **・ PL-FD210 を装着している場合は、PL-FD210 を外してください。(PL-6920<4 スロットタイプ > のみ装着可能)**



④ 背面部を上面側にスライドさせます。



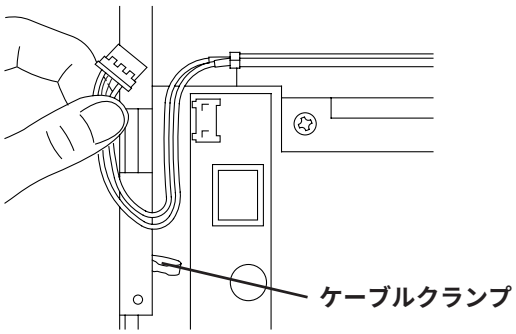
⑤ 背面部を上面側に引き上げ、前面から取り外します。



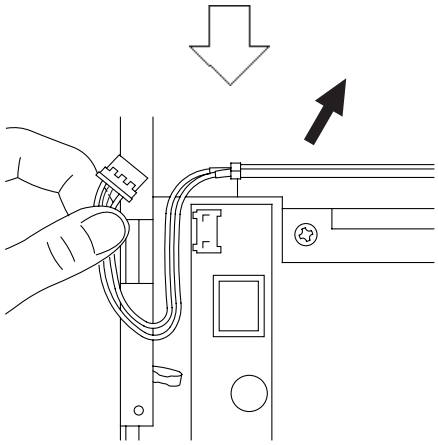
⑥ インバータ基板からコネクタを外します。

⑦ バックライトユニットはLCD本体とバックライト固定ネジで一カ所のみ固定されています。
バックライト固定ネジ(落下防止ワッシャー付き)をドライバでゆるめます。

重要
- **バックライト固定ネジはプラスドライバ No.1が適しています。**
- **ネジを紛失しないようにしてください。**
- **ネジを本体に混入させないでください。破損の恐れがあります。**

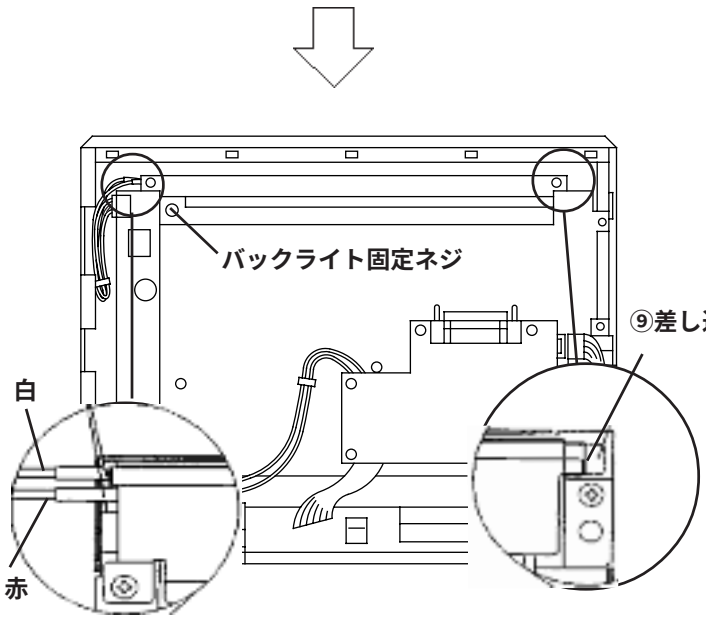


⑧ケーブルクランプからケーブルを外します。



⑨ バックライトユニットを持ち上げて、フレームから斜めに引き抜きます。

重要
- **バックライトは、バックライトユニットごと交換してください。**



⑩凹部に分けてケーブルをはめ込む。

⑩ 新しいバックライトユニットのケーブルとは反対側のツメをフレームの穴に差し込みます。バックライトユニットをフレームの溝に合わせてはめ込みます。(⑦と逆の手順)

⑪バックライトユニットのケーブルはフレームの2ヵ所の凹部にはめ込みます。(上凹部に白ケーブル1本、下凹部に赤ケーブル2本)

⑫バックライト固定ネジ(1ヵ所)をドライバで固定します。締め付けトルクは0.19N･mです。

⑬インバータ基板にコネクタを差し込み、ケーブルをケーブルクランプで固定します。(⑥⑧と逆の手順)

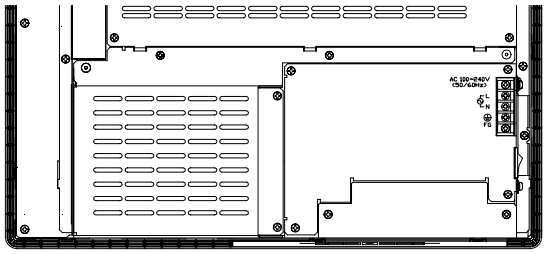
重要・ インバータ基板へコネクタを差し込む際は、奥まで確実に挿入してください。破損の恐れがあります。

⑭取り外したPL背面部はケーブルを挟まないよう前面部にかぶせます。(④⑤と逆の手順)

重要・ ケーブルランプはPL本体にケーブルがはさまらないために設置されています。閉口の際は、必ずケーブルはケーブルクランプに引っかけてから後部をフロント部にかぶせてください。

⑮背面ネジ(4ヵ所)をドライバで固定します。(③と逆の手順)

■ PL-7920 シリーズの場合

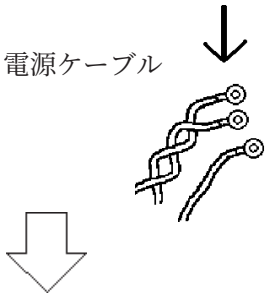


① PL の電源を切ってください。また、電源ケーブルに電源が供給されていないことを確認してください。感電のおそれがあります。

② 電源ケーブルを取り外します。

PL-7920(4 スロット)をモデルとしています。

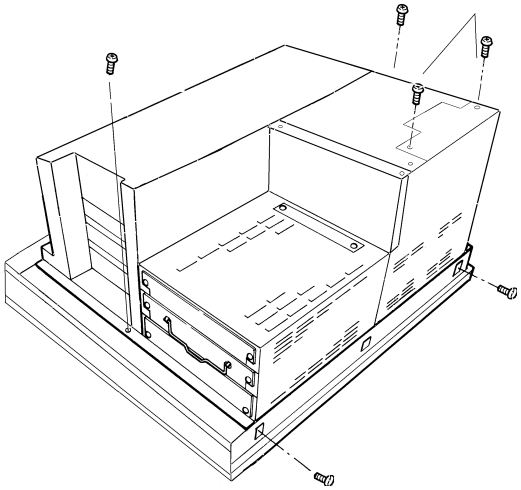
重要 **• 作業は平らな場所でおこなってください。不安定な場所での作業はケーブルの断線や PL の破損につながります。**



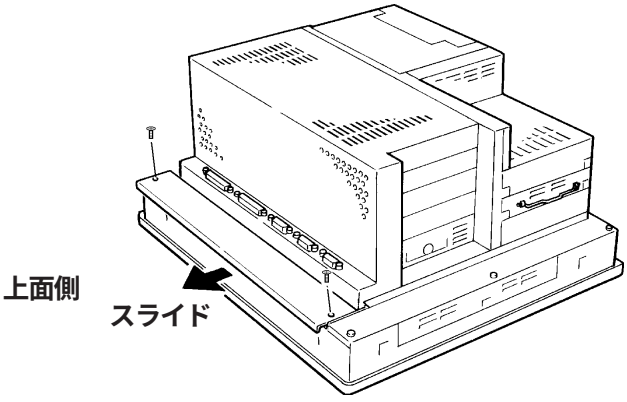
PL-FD210を装着の場合、ネジを外し、PL-FD210を外してください。

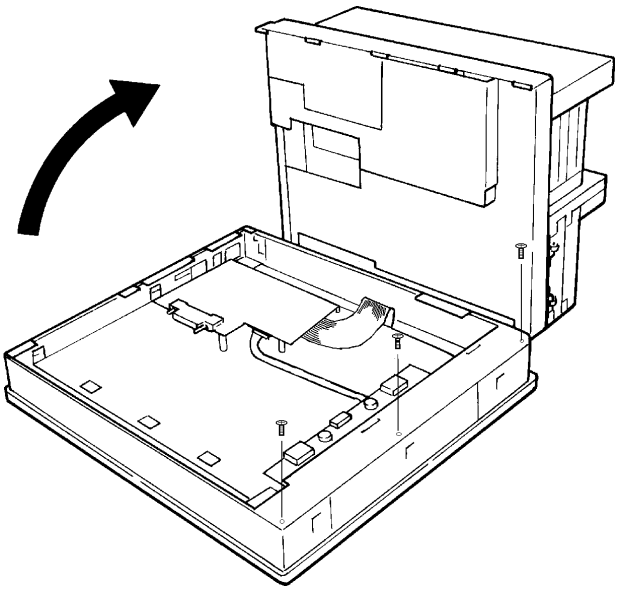
③ 本体にあるネジ(4ヵ所)をドライバではずします。

重要 **• PL-FD210 を装着している場合は、PL-FD210 を外してください。(PL-7920<4 スロットタイプ> のみ装着可能)**

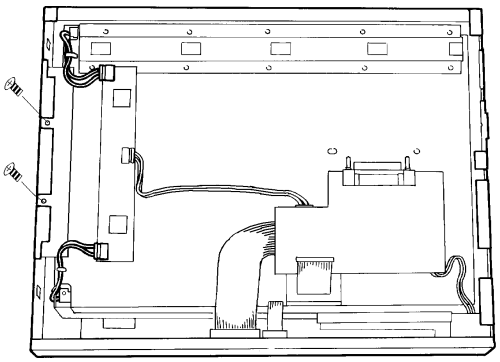


④ ネジ(2ヶ所)を取り外し、カバーを外し、後部を上面側にスライドされます。

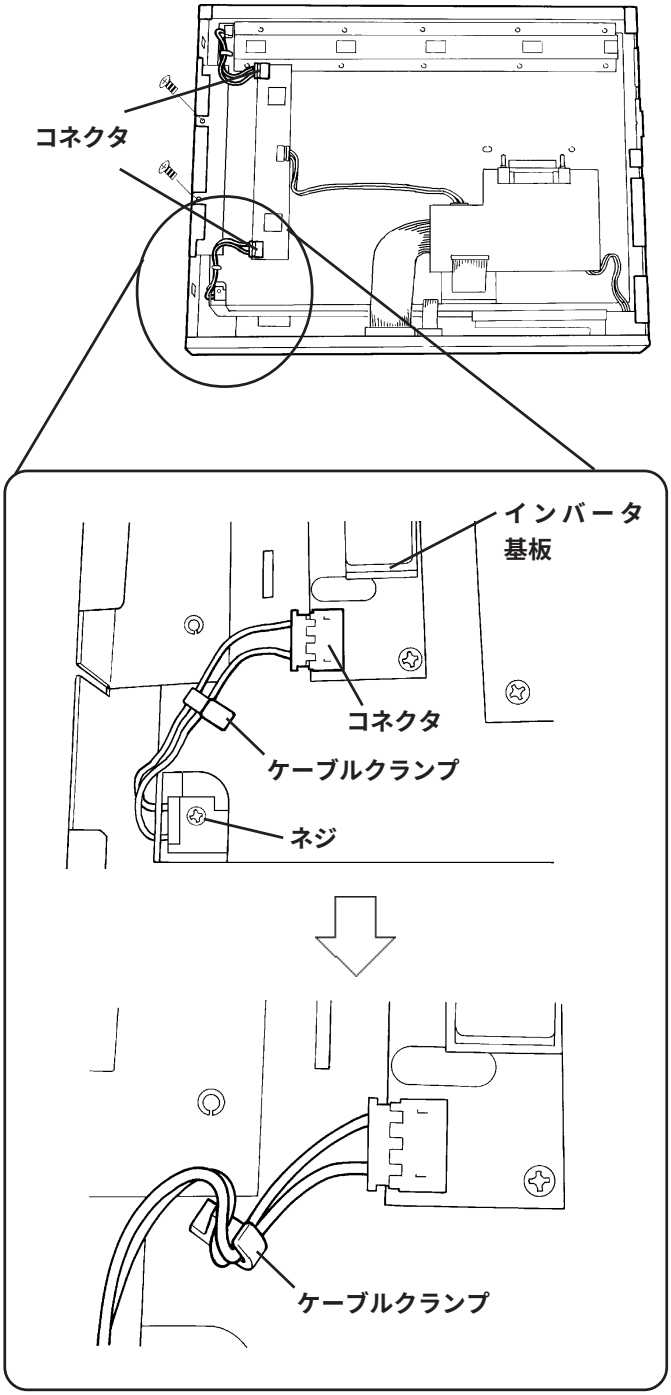




⑤ フロント部から後部を取り外し、 カバーのネジ(3ヶ所)を取り外します。



⑥ 表示部LCD固定ネジ（2カ所)をドライバで外します。



⑦ インバータ基板からコネクタを外します。

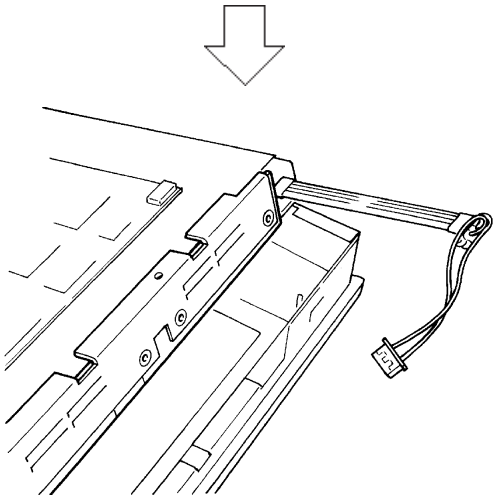
⑧ バックライトはLCD本体にネジで固定されています。
固定ネジ(2ヵ所)をドライバで外します。

重要
- **バックライト固定ネジはプラスドライバ No.1が適しています。**
- **ネジを紛失しないようにしてください。**
- **ネジを本体内に混入させないでください。破損の恐れがあります。**

⑨ ケーブルクランプからケーブルを外します。

MEMO

ケーブルクランプはPL本体にケーブルを挟み込まないために設置されています。必ずケーブルはケーブルクランプに引っかけてから後部をフロント部にかぶせてください。



⑩ バックライトを取り外します。

⑪ 交換用バックライトを差し込みます。

⑫ バックライトはLCDの上下に各１本ずつ使用しているので ⑨～⑪の手順に従い、両方のバックライトを交換します。交換後、バックライト固定ネジおよび、LCDパネル固定ネジの合計4ヵ所をネジ止めし、インバータ基板にコネクタを差し込みます。

重要

- **バックライトは LCD の上下に各１本ずつ使用しています。交換の際には、2 本同時に交換してください。**
- **インバータ基板にコネクタを差し込む場合は、奥まで確実に差し込んでください。破損につながります。**
- **PL-FD210 は、PL-7920 でのみ使用することができます。**

⑬ 取り外した後部はケーブルをはさまないようフロント部にかぶせます。

⑭ PL-FD210を取り付け、2カ所をネジ止めして固定します。(PL-7920のみ)

⑮ 取り外したネジをすべてネジ止めして固定します。

7.4 定期点検

PLを最良の状態で使用するために定期的に点検を行ってください。

■周囲環境の点検

□周囲温度は環境仕様内か？

<table>
<tr><td rowspan="10">使用周囲温度</td><td rowspan="2">PL692*-T41
(CPU:700MHz)</td><td colspan="2">ファン使用</td><td>5℃～50℃(HDD使用時)</td></tr>
<tr><td colspan="2">ファン未使用※1</td><td>5℃～40℃(HDD使用時)</td></tr>
<tr><td rowspan="3">PL792*-T41
(CPU:700MHz)</td><td rowspan="2">盤内</td><td>ファン使用</td><td>5℃～50℃(HDD使用時)</td></tr>
<tr><td>ファン未使用※1</td><td>5℃～40℃(HDD使用時)</td></tr>
<tr><td>表示面側</td><td colspan="2">5℃～40℃</td></tr>
<tr><td rowspan="2">PL692*-T42
(CPU:1GHz)</td><td colspan="2">ファン使用</td><td>5℃～45℃(HDD使用時)</td></tr>
<tr><td colspan="2">ファン未使用※1</td><td>使用不可</td></tr>
<tr><td rowspan="3">PL792*-T42
(CPU:1GHz)</td><td rowspan="2">盤内</td><td>ファン使用</td><td>5℃～45℃(HDD使用時)</td></tr>
<tr><td>ファン未使用※1</td><td>使用不可</td></tr>
<tr><td>表示面側</td><td colspan="2">5℃～40℃</td></tr>
</table>

□周囲湿度は環境仕様内（10 ～ 85%RH）か？

□雰囲気は適当（腐食性ガスなし）か？

■電気的仕様の点検

□電圧は適当（AC85 ～ 265V 50/60Hz）か？

■取り付け状態の点検

□接続ケーブルのコネクタは完全に差し込まれている（ゆるみがない）か？

□ PL を取り付けている金具にゆるみがないか？

□防滴パッキンにキズや汚れが目立ってきていないか？

■使用状態の点検

□画面が暗くて見づらくないか？

・ バックライト交換が必要な場合は、(株) デジタル サービス・リペアセンター窓口までお問い合わせください。

参照→7.5 アフターサービス

※1 本体内部にある電源ファンを取り外した場合。

7.5 アフターサービス

■サービス・リペアセンター

(株)デジタル製品の故障、修理などのご相談に対応いたします。

お問い合わせの際には問題点、現象などをあらかじめご確認の上、ご連絡ください。また製品送付時には、問題点、現象を書き留めた修理依頼書を同封してください。その際、輸送時の振動で製品が破損しないよう、梱包状態には十分ご注意くださいますようお願いいたします。(修理依頼書は下記受け付け窓口へご請求ください。)

■お問い合わせ先

サービス・リペアセンター　大阪　(月～金　9:00 ～ 17:00)

TEL：(06) 6613-1638　　FAX：(06) 6613-1639

以下のサービスの受付け窓口は、お買い求めの代理店、(株)デジタルの営業担当、または(株)デジタル サービス・リペアセンターです。

■契約保守

製品ご購入時に年間一定料金で契約を結ぶことにより、不具合に対して無償でサービス・リペアセンター修理をするシステムです。

■サービス・リペアセンター修理

お客様より修理品をサービス・リペアセンターへ返却していただき、修理するシステムです。故障した製品を宅配便等でお送りいただき、修理後ご指定の場所へお返しいたします。処置内容により修理費用は異なります。

■保証および修理について

1. 無償保証期間

無償保証期間は、納入後12ヶ月とさせていただきます(有償修理品の故障に対しては、同一部位のみ修理後3ヶ月)。無償保証期間終了後は有償での修理となります。

2. 無償保証範囲

(1) 無償保証につきましては、上記無償保証期間中、弊社製品の使用環境・使用状態・使用方法などがマニュアル・取扱説明書・製品本体注意ラベル等に記載された諸条件や注意事項に従っていた場合にのみ限定させていただきます。

(2) 無償保証期間内であっても、次のような場合には、有償修理とさせていただきます。

1. 納入後の輸送(移動)時の落下、衝撃等、貴社の取扱い不適当により生じた故障損傷の場合。
2. カタログ・マニュアル記載の仕様範囲外でご使用された場合。
3. 取扱説明書に基づくメンテナンス、消耗部品の交換保守が正しく行われていれば防げたと認められる故障の場合。
4. 火災、地震、水害、落雷、その他天災地変、公害や異常電圧による故障及び損傷。
5. 接続している他の機器、及び不適当な消耗品やメディアの使用に起因して本製品に生じた故障及び損傷。
6. 消耗部品の交換。
7. 販売当時の科学・技術の水準では予見できない原因による故障の場合。
8. その他、貴社による故障、損傷または不具合の責と認められる場合。

(3) 次のような場合には、たとえ有償であっても修理をお断りすることがございます。
弊社以外で修理、改造等をされたと認められる場合。

3．生産中止について

(1) 弊社製品の生産中止は、弊社ホームページ上で、最終出荷の6ヶ月前に掲示いたします。

(2) ただし、使用部品の生産中止に伴う弊社製品の生産中止に関しましては、部品メーカーからの生産中止の連絡があり次第、弊社ホームページ上に掲示いたします。

4．生産中止後の修理期間（有償修理）

(1) 生産中止を弊社ホームページで掲示した月を起点として7年間は、弊社サービスリペアセンターにて当該製品の修理を行います（2005年10月現在)。2005年9月以前に生産中止となった製品は、最終出荷日より5年間が修理期間となります。

(2) 上記期間に限らず、交換部品が入手不可能となった場合には、修理できなくなることがございますのでご了承ください。

5．修理条件

(1) 修理は、弊社製品のみを対象といたします。 オプション品は対象外となります。

(2) 修理に際し、お客様のプログラムやデータが消失することがありますので、予めデータを保存するようにしておいてください。

(3) 弊社製品に記憶されているお客様のデータにつきましては、取扱には十分に注意をいたしますが、お客様の重要機密に関する事項等は、修理前に消去いただくようお願いいたします。消去できない故障の場合は、その旨を予めご連絡いただくようお願いいたします。

(4) 修理は、センドバックによる弊社工場修理を原則とさせていただきます。この場合、弊社工場への送料はお客様負担にてお願いいたします。

(5) 修理にて交換された部品の所有権は(株)デジタルに帰属するものとします。

■技術ご相談窓口

PLご使用時の技術的なご相談を承ります。

1　お問い合わせの前に
まずマニュアルの該当するページをご覧ください。

2　お問い合わせの際には次の点についてお知らせください。
①ご担当者名　②ご連絡先電話番号　③ご使用機種　④シリアル　No.　⑤ご使用環境

問題点・現象・操作を行った手順などを、あらかじめ書き留めてからご連絡くださるようお願いいたします。

3　お問い合わせ先
■月～金　9:00～17:00
TEL　大阪：(06) 6613-3115　　東京：(03) 5821-1105　　名古屋：(052) 932-4093

FAXでお問い合わせの場合は、次頁の「PL-6920/PL-7920シリーズお問い合わせFAX」をコピーし、質問事項をご記入のうえ、（株）デジタルまでご返送ください。

■ホームページからのアクセス

ホームページからのお問い合わせには随時承ります。

URL　http://www.proface.co.jp

宛先
株式会社 デジタル
サポートダイヤル宛

FAX : 06(6613)5982

PL-6920/PL-7920 シリーズお問い合わせ FAX　　年　月　日　枚

ご連絡先

貴社名＿＿＿＿＿＿＿＿＿＿＿＿　TEL＿＿＿＿＿＿＿＿
ご所属＿＿＿＿＿＿＿＿＿＿＿＿　FAX＿＿＿＿＿＿＿＿
ご氏名＿＿＿＿＿＿＿＿＿＿＿＿　E-Mail＿＿＿＿＿＿＿＿
ご住所 〒＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿
製品型式＿＿＿＿＿＿＿＿＿＿＿＿　ご購入先＿＿＿＿＿＿＿＿
シリアル№＿＿＿＿＿＿＿＿＿＿＿＿　お買上日＿＿＿＿＿＿＿＿

※シリアルNo.(本体後面の定格銘板に記載)が記入されていないと質問にお答えできません。

ご使用環境

＜システム構成＞

□本体（□PL-6920 □PL-6921 □PL-7920 □PL-7921）
□拡張メモリ（□PL-EM500 □PL-EM128）
□ハードディスク(□PL-HD220 □PL-HDX920-W95 □PL-HDX920-NT40
□PL-HDX920-W2K □PL-HDX920-W2K/ML □PL-HDX920-WXP)
□フロッピーディスク(□PL-FD200 □PL-FD210)
□PL-FF210 □PL-DK200 □PL-MD***
□その他（オプション品、市販品)＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

＜使用ソフト環境＞

□Windows®95 プリインストールタイプ
□WindowsNT®4.0 プリインストールタイプ
□Windows®2000 Professional プリインストールタイプ
□Windows®2000 Professional Multi Language プリインストールタイプ
□Windows®XP Professional プリインストールタイプ
□その他OS＿＿＿＿＿＿＿＿＿＿　Version＿＿＿＿＿＿
□アプリケーション＿＿＿＿＿＿
□その他 ＿＿＿＿＿＿＿＿＿＿

お問い合わせ内容（エラーメッセージ等は正確に記入してください。）

デジタル記入欄	処　理	受　付

付録

I/Oマップ、メモリマップ、割り込みマップなどのハードウェア構成とRAS機能について説明します。

付 .1 ハードウェア構成

付 .1.1 I/O マップ

アドレス	ATシステムデバイス	システム固有デバイス
0000H-001FH	DMAコントローラ(8237)	
0020H-003FH	割り込みコントローラ(8259A)	
0040H-005FH	システムタイマ(8254)	
0060H-006FH	キーボードコントローラ	
0070H-007FH	RTC、NMIマスク	
0080H-009FH	DMAページレジスタ	
00A0H-00BFH	割り込みコントローラ2(8259A)	
00C0H-00DFH	DMAコントローラ2(8237)	
00F0H-00FFH	数値演算プロセッサ	
01F0H-01FFH	ハードディスク(IDE)	
0200H-0207H	ゲームI/O	
0290H-029FH	リザーブ	
02E8H-02EFH	リザーブ	タッチパネル シリアルポート4(COM4)
02F8H-02FFH	シリアルポート2(COM2)：汎用	
03B0H-03BBH	ビデオコントローラ(VGA)	
03BCH-03BFH	パラレルポート1(LPT1)	
03C0H-03DFH	ビデオコントローラ(VGA)	
03E8H-03EFH	リザーブ	シリアルポート3(COM3)
03F0H-03F7H	フロッピーディスクコントローラ	
03F8H-03FFH	シリアルポート1(COM1):汎用	

付 .1.2 メモリマップ


FFFFFFFFH
システム ROM
FFFF0000H
RAM 空間
00100000H
システム ROM
000E0000H
拡張 ROM エリア
000CA000H
VGA BIOS
000C0000H
ディスプレイアダプタ
000A0000H
RAM 空間
640K バイト
システムボード DRAM
000000H

付 .1.3 割り込みマップ

MEMO
・ 割り込み、DMAチャンネルはPCI/ISAのPnPの機能によって変化する場合があります。

■ハードウェア割り込み一覧

	要因
NMI	パリティエラーまたはI/Oチャネル・チェック
IRQ 0	タイマ(チップセット内)
1	キーボード
2	コントローラ2からのカスケード
3	シリアルポート2（COM2)：汎用ポート
4	シリアルポート1（COM1)：汎用ポート
5	ユーザー使用可　※
6	フロッピーディスクコントローラ
7	パラレルポート1（LPT1)：プリンタポート
8	リアルタイムクロック
9	シリアルポート3（COM3)：汎用ポート
10	シリアルポート4（COM4)：タッチパネル
11	ユーザー使用可　※
12	PS/2マウス
13	数値演算プロセッサ
14	ハードディスク（IDE)
15	ユーザー使用可　※

重要
・ ※はお客様の設定によってはプラグアンドプレイデバイスである下記デバイスが自動的に割り当てられます。

Display Controller
SMBus Controller
Multimedia Device
Network Controller
USB Controller

■ DMA チャンネル一覧

	要因	
DMA 0		8 ビット転送用
1		
2	フロッピーディスクコントローラ	
3		
4	コントローラ1へのカスケード	16 ビット転送用
5		
6		
7		

付 .2 RAS 機能について

付 .2.1 PL の RAS 機能

RAS（Reliability Availability Serviceability）機能とは、システムの信頼性を向上することを目的に用意された機器監視機能を中心とする様々な機能の総称です。

一般的にサポートされている機能は機器により異なり、PL では RAS 機能として下記の異常監視と外部入力信号をサポートしています。

異常監視	電源電圧異常 冷却ファン回転異常 内部温度異常 ウォッチドッグタイマタイムアップ ソフトミラーディスク異常 ※3 ハードミラーディスク異常 ※1 ※3 タッチパネル異常 バックライト切れ検出機能※4 SMART監視
外部入力信号	汎用信号入力（DIN 2ビット） リモートリセット入力 ※2

重要 **・ 汎用信号入力(DIN)に関しては信号レベルを 1.5S 以上保持しないと検出できない場合があります。**

また、PL では上記異常発生および外部信号入力時のアラーム処理出力として、下記の外部出力信号と各種処理機能をサポートしています。

外部出力信号	汎用信号出力（DOUT 1ビット） アラーム出力（1点） ランプ出力（1点）
各種処理機能	LEDインジケート（3色発光 1点） ポップアップメッセージ出力 ブザー出力 システムシャットダウン処理 システムリセット

さらにPLでは添付のシステムモニタ（ソフトウェアユーティリティ）を使用することにより、上記の異常監視項目や外部入力信号ごとに監視機能の有効無効およびアラーム処理内容を設定できます。**参照→付 .3 システムモニタ**

また、システムモニタに他のアプリケーションから利用するためのダイナミックリンクライブラリ（API-DLL）を用意しています。

***1 ハードミラーディスク異常については監視機能は常に有効で、異常検出時 LED は赤色、緑色、橙色の点滅表示します。**

***2 リモートリセットについては入力の有効無効設定は可能ですが、強制的にハードウェアリセットがかかるため、アラーム出力状態の設定はできません。**

***3 ハードミラーディスク、ソフトミラーディスクは別途購入が必要です。**

***4 PL-6920 シリーズのみの対応です。**

付 .2.2 RAS 機能詳細

PL の RAS 機能詳細を示します。

■異常監視

◆電源電圧異常

PL の内蔵電源および内部での CPU 供給電源の状況を監視します。

◆冷却ファン回転異常

PL 本体の電源冷却ファンおよび CPU 冷却ファンの回転数を監視します。

◆内部温度異常

PL本体の内部温度およびCPU周辺の温度を監視します。

上記3つの異常検知の有効無効はシステムのセットアップで設定します。異常検知設定の詳細については**参照→ 5.2.10 PC Health Status** をご覧ください。

システムモニタでもこれらの異常監視の有効無効および異常処理内容を設定できます。

◆ウォッチドッグタイマタイムアップ

内蔵のRAS機能専用プログラマブルタイマにアプリケーションから定期的にタイムアップカウント値の書き込みを繰り返すことによってCPUの正常動作を監視します。アプリケーションからのカウント値書き込みが停止し、タイマがオーバーフローした場合に異常検知されます。

ウォッチドッグタイマタイムアップの有効/無効および異常処理内容はシステムモニタで設定します。

◆ソフトミラーディスク異常

オプションのソフトミラーディスクによりハードディスクの異常を検出します。

◆ハードミラーディスク異常

オプションのミラーディスクにディスククラッシュなどのエラーが発生した場合、LEDインジケートで異常を知らせます。

◆タッチパネル異常

タッチパネルの異常検出を行います。タッチパネル異常が発生した場合、LEDが橙色に点灯します。

◆バックライト切れ検出機能(PL-6920シリーズのみ)

バックライト切れの検出を行います。
バックライト切れが発生した場合、タッチパネルの操作を無効にし、LED インジケートにて通知します。出荷時の設定は有効です。不当なタッチパネル操作による誤動作を防止するため有効にすることをお勧めします。設定は BLSET.EXE にて行います。(BLSET.EXE は MS-DOS® 用ユーティリティです)有効、無効の切り替えについては、**参照→6.4 アプリケーション機能**をご参照ください。

◆ SMART 異常

ハードディスクの状態を監視します。ハードディスクが故障する前兆を検出します。

重要

- **SMART 監視を行うには Administrator 権限が必要です。**
- **SMART 監視を行うには OS が Windows NT®、Windows® 2000、または Windows® XP である必要があります。**
- **CF カードは SMART に対応していないため、CF カードの状態を監視することはできません。**
- **(株)デジタル製オプション品以外のハードディスクを使用した場合、SMART 監視の動作は保証できません。**
  以下のハードディスクに対応しています。

ハードディスク型式	Rev.
PL-HD220	Rev.C以降
PL-HDX920-NT40	ハードRev.A以降
PL-HDX920-W2K	
PL-HDX920-WXP	
PL-HDX920-W2K/ML	

- **IDE で接続されたハードディスクのみ SMART 監視を行うことができます。USB や SCSI で接続されたハードディスクの状態を監視することはできません。**
- **Rev.C 以前のソフトミラーユーティリティ PL-SM900 使用時は SMART 監視を行うことはできません。**

■外部入力信号

PL 本体の RAS インターフェイスコネクタに下記の入力信号が用意されています。

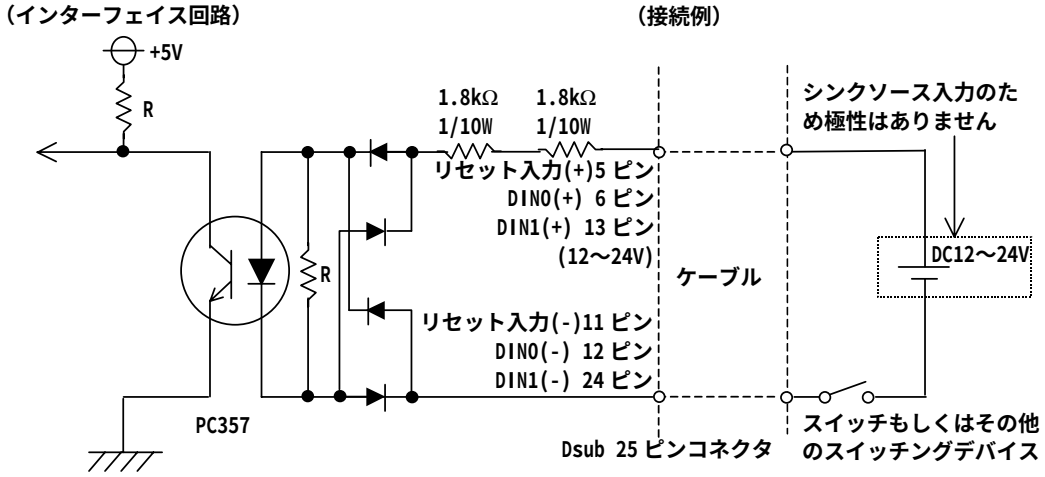
◆汎用信号入力（DIN）

外部機器の異常検知用に用意された汎用デジタル入力です。入力は 2 ビット用意されています。
システムモニタで本信号の有効無効および処理内容を設定します。

◆リモートリセット入力

外部機器による PL のハードウェアリセット信号です。本信号が有効になった場合に強制的にハードウエアリセットがかかります。
システムモニタで本信号の有効無効を設定します。

入力電圧	DC12V～24V
入力電流	7mA
動作電圧	ON電圧:9V（min）、OFF電圧:3V（max）
絶縁方式	フォトカプラによる絶縁



重要

- **汎用信号入力（DIN）は、入力レベルを 1.5S 以上保持してください。1.5S 以下では検出できないことがあります。**
- **端子間の電圧値は、入力電圧で決められた範囲内で使用してください。入力電圧範囲を超えますと故障の原因となります。**
- **シンクソース入力のため、D(-)、RESET(-)が正極、D(+)、RESET(+)が負極となっても問題ありません。この場合も、上記入力電圧範囲内で使用してください。**

MEMO

- コネクタピン配列については**2.3.5 RAS インターフェイス**をご覧ください。

■外部出力信号

PL 本体の RAS インターフェイスコネクタに下記の出力信号が用意されています。

◆汎用信号出力（DOUT）

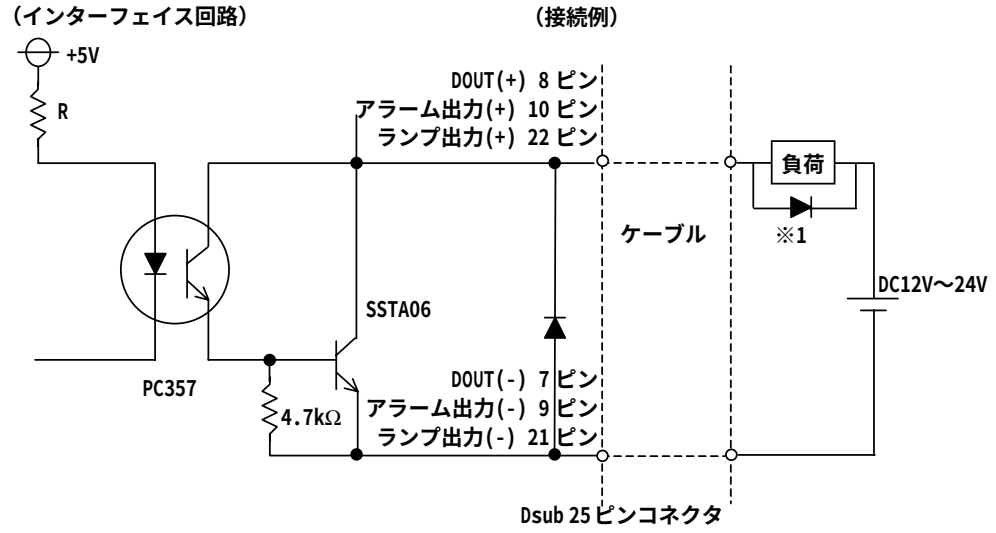
本信号は、外部機器にシステムの状態を通知するために準備されたデジタル出力信号です。
システムモニタの API-DLL でアプリケーションからコントロールできます。

◆アラーム出力（1 点）

◆ランプ出力（1 点）

これらの信号は、外部機器にシステムの状態を通知するために準備された汎用デジタル出力です。
システムモニターで出力の有効無効の設定が可能です。
なお、アラーム出力を有効にした場合は LED インジケートも同時に橙色に点灯します。

定格負荷電圧	DC12V～24V
最大負荷電流	100mA/点
端子間最大降下電圧	1.5V（負荷電流100mA時）
絶縁方式	フォトカプラによる絶縁



重要

- **最大負荷電流内で使用してください。最大負荷電流を超えて使用すると故障の原因となります。**
- **負荷の電流値および電圧値は、端子間電圧を加味したうえで設計してください。負荷電流を大きくとりますと、端子間にて最大 1.5V の電圧降下が生じます。**
- **誘導性負荷を接続する場合は上図※ 1 の保護用ダイオードを接続してください。**

MEMO

- コネクタピン配列については**2.3.5 RAS インターフェイス**をご覧ください。

■各種処理機能

PLでは下記の状態通知処理を行うことができます。

◆LED インジケート

電源の ON/OFF を表示するパワーランプと共用化された 3 色発光 LED で、下記の発光色でシステムの状態を通知します。PL-6920/PL-7920シリーズのLEDは前面パネルの左下部にあります。

発光色	システム状態	出力条件
橙色 点灯	何らかのRAS異常発生	システムモニタでアラーム出力の設定が有効
緑色 点灯	正常動作中（電源ON）	無し
橙色/緑色 点滅	ソフトミラーディスク異常発生 または ハードミラーディスク異常発生	無し
橙色/赤色 点滅	バックライト異常 (PL-6920シリーズのみ)	無し
赤色/緑色 点滅	ハードミラーディスク異常発生 ＋ バックライト異常 (PL-6920シリーズのみ)	無し

◆ポップアップメッセージ出力

Windowsのポップアップメッセージでシステムの状態を通知する機能です。

システムモニタで表示の有効無効を設定します。

◆ブザー出力

PLの内蔵スピーカの出力にてシステムの状態を通知する機能です。

システムモニターで出力の有効無効を設定します。

◆システムシャットダウン処理

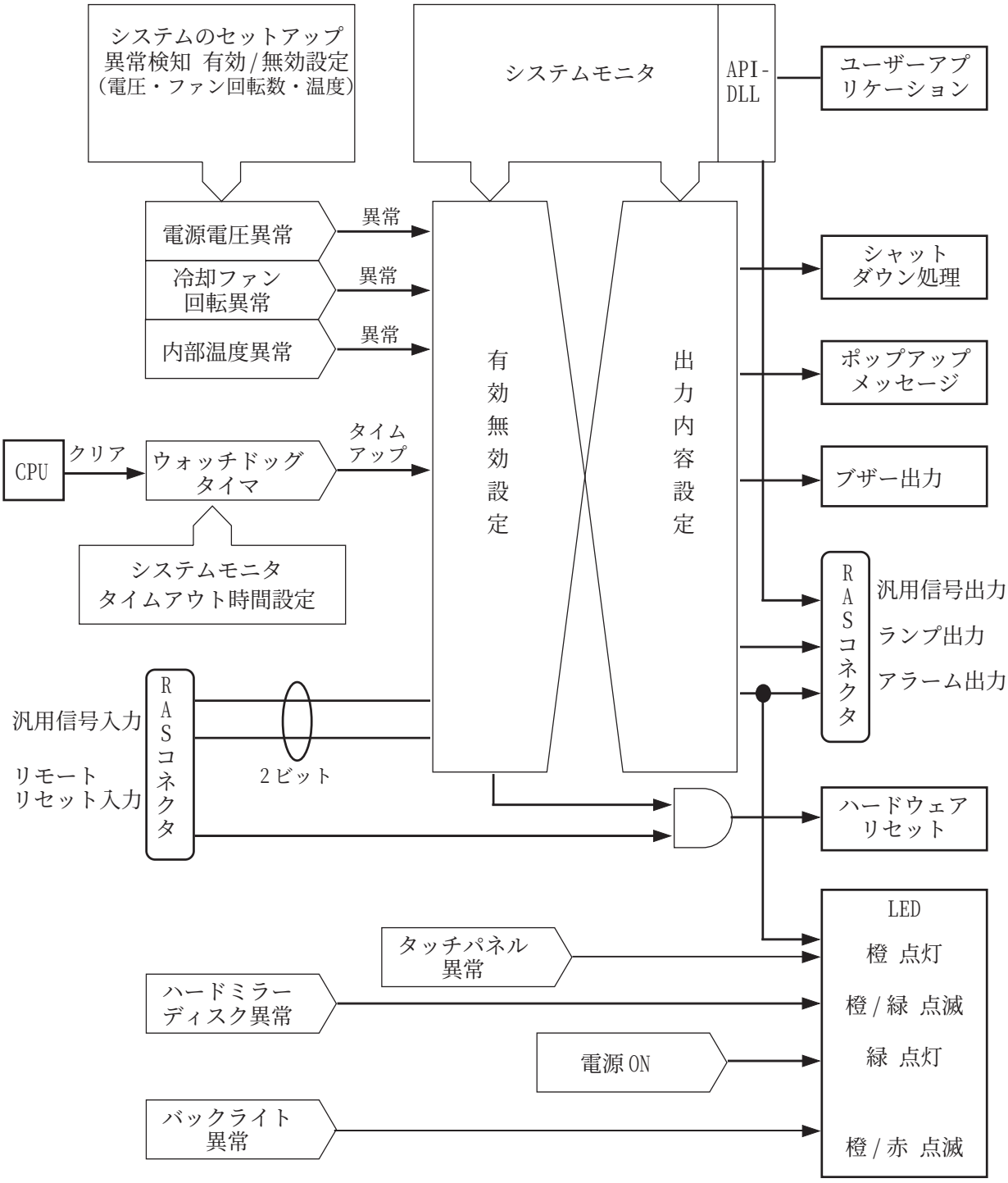
OSのシャットダウンを行う機能です。

システムモニタで本処理の有効無効を設定します。

◆システムリセット

ウォッチドッグタイマがタイムアップした場合にシステムをリセットする機能です。

付 .2.3 RAS 機能概念図

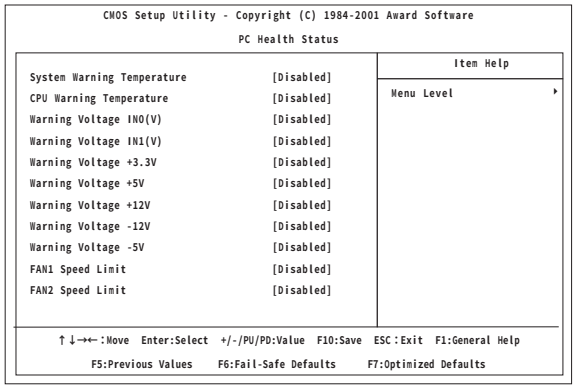


付 .3 システムモニタ

付 .3.1 設定方法

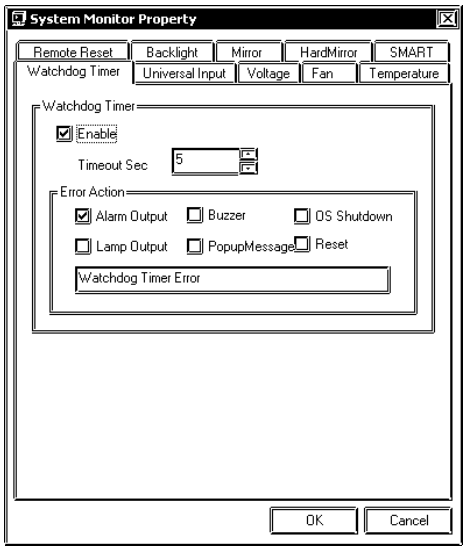
システムモニタ /RAS 機能を使用する為には、次のステップで設定を行ってください。

■ BIOS 画面での設定

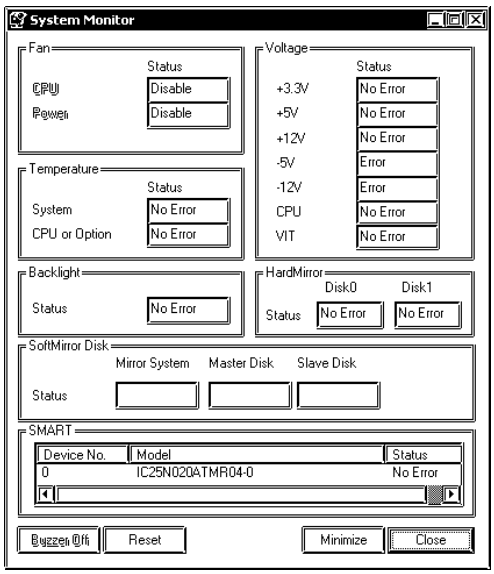


BIOS のセットアップで PC Health Status メニューで各監視機能の有効/無効を設定します。

■アプリケーションでの設定



①OS を起動し[スタート]->[プログラム]->[System Monitor]->[System Monitor Property]を起動し、システムモニタ/RASイベント発生時の動作を設定します。



②[スタート]->[プログラム]->[System Monitor]->[System Monitor]を起動することにより、監視することができます。

MEMO

- SMART 監視を行うには Administrator 権限が必要です。Administrator 権限を持たないユーザーがログインした場合、SMART の項目の欄には何も表示されません。
- SMART 監視を行うには OS が Windows NT®、Windows® 2000、または Windows® XP である必要があります。Windows® 95、Windows® 98 の場合、SMART の項目の欄には何も表示されません。

付 .3.2 システムモニタプロパティの設定（PL_Wps.exe）

PC Health Statusメニューで設定した各監視機能の有効範囲を越えた場合の動作の設定します。各機能に対して以下の動作設定が可能です。

○：設定可能
×：設定不可能

	Alarm Output	Lamp Output	Buzzer	Popup Message	OS Shutdown	Reset
Watchdog Timer	○	○	○	○	○	○
Universal Input	○	○	○	○	○	×
Voltage	○	○	○	○	○	×
Fan	○	○	○	○	○	×
Temperature	○	○	○	○	○	×
Remote Reset *1	×	×	×	×	×	○
Backlight *2	○	○	○	○	×	×
Mirror	○	○	○	○	×	×
HardMirror	○	○	○	○	×	×
SMART	○	○	○	○	×	×

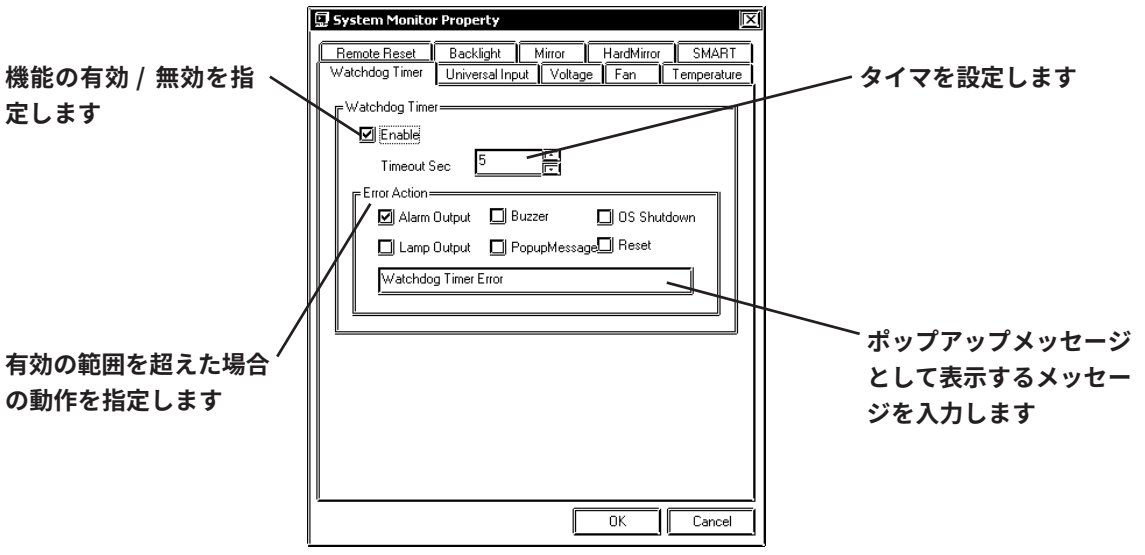
*1 システムモニタプロパティで Enabled を指定すると、Reset と同じ動作となります。

*2 PL-6920 シリーズのみ使用できます。

各項目の動作内容を下記に示します。

項目	動作内容
Alarm Output	RASインターフェースのアラーム出力(9番-10番)から信号が出力されます。
Lamp Output	RASインターフェースのアラーム出力(21番-22番)から信号が出力されます。
Buzzer	警告音としてBEEP音を鳴らします。(OS Shutdownがチェックされている場合を除きます。)
Popup Message	エラーメッセージがポップアップメッセージで表示されます。(画面上にポップアップします。)
OS Shutdown	OSを終了します。終了確認メッセージが表示される設定と強制終了設定ができます。デフォルトは終了確認メッセージを表示します。
Reset	ハードウェアをリセットします。強制終了します。
Enable	各監視設定を許可します。

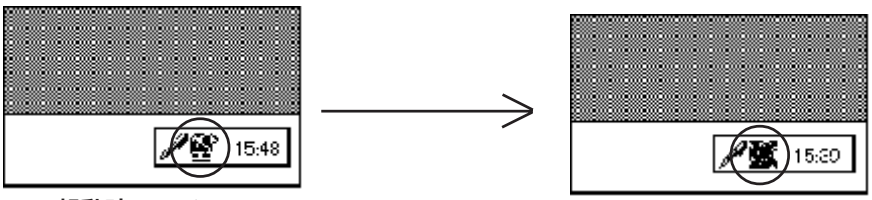
システムモニタプロパティの設定画面の概要を示します。



付 .3.3 システムモニタの動作(PL_Smon.exe)

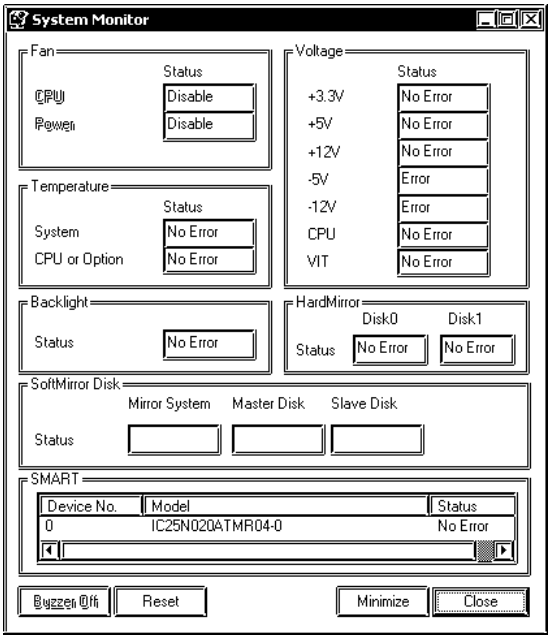
システムモニタの起動直後は、システムモニタ画面は表示されず、アイコンがシステムトレイに格納された状態となります。

異常を検出した場合、システムモニタプロパティで設定された「動作」を行い、システムトレイのアイコンが「×」マークの異常ありを示すアイコンに変わります。システムトレイのアイコンが異常ありに変化した場合は、システムトレイのアイコンをダブルクリックし異常内容を確認してください。



起動時のアイコン　　異常検出時のアイコン

システムモニタ画面を以下に示します。



システムモニタ画面

MEMO
- SMART の Device No. は「0」がマスタ、「1」がスレーブを表しています。

システムモニタ画面下部には「Buzzer Off」「Reset」「Minimize」「Close」のボタンがあり、以下の機能を持ちます。

ボタン	動作
Buzzer Off	通常動作のブザーを停止
Reset	通常動作およびシステムモニタ内部での異常状態保持クリア
Minimize	システムモニタをアイコン化
Close	システムモニタ終了

システムモニタ画面内ではファン/温度/電圧/バックライト/ハードミラー/ソフトミラー/SMARTの各監視要素について、それらが正常であるか異常であるか、または非監視であるかを表示します。表示するステータスには以下のものがあります。

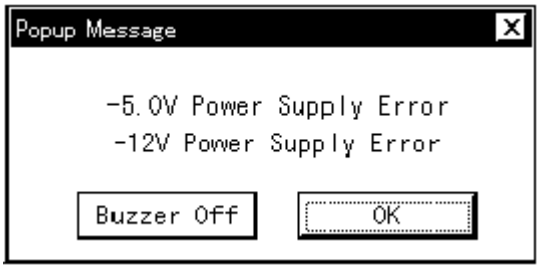
表示	意味
No Error	正常
Error	異常
Disable	監視しない
Not Support	未対応

MEMO
・「Not Support」はCFカードなどSMARTに対応していないデバイスが検出された場合に表示されます。

それぞれの監視要素について異常を検出した場合、および外部入力(Universal Input)からの入力を検出した場合、システムモニタプロパティで設定された通知動作(Error Action)を行います。**参照→付 .3.2 システムモニタプロパティの設定**

通知動作は、異常/入力を検出した時、それぞれの監視要素について1回のみ行われます。

+3.3V電圧と+5.0電圧を例に上げると、電圧監視に関してポップアップメッセージを選択していた場合、+3.3V電圧が異常となると、+3.3V異常を示すポップアップメッセージが表示されます。「OK」を押してメッセージ画面を閉じるとその後+3.3V異常に対するメッセージは表示されないが、+5.0V電源が異常となれば+5.0V異常を示すポップアップメッセージが表示されます。ポップアップメッセージは、エラーになった監視要素とエラー内容を表示します。ブザーを選択していた場合は、ポップアップメッセージの「Buzzer OFF」ボタンを押すと、ブザーを停止することができます。「OK」ボタンを押すとポップアップメッセージを閉じます。



ポップアップメッセージの出力画面

OS Shutdownを選択した場合には、ユーザに特に許可を求めずにシャトダウン処理に入ります。

通常時にシステムモニタ画面を表示し、現在の状態を確認する場合は、システムトレイ内のアイコンをダブルクリックすることによりシステムモニタ画面を表示します。

警告動作としてブザーが鳴った場合、システムモニタ画面内に通常は非表示の「Buzzer Off」ボタンが表示されます。ポップアップメッセージが表示されている場合にはポップアップメッセージ画面も「Buzzer Off」ボタンが表示されます。

重要 **・一度異常を検知すると、システムモニタは「異常」状態を保持します（異常状態を示すアイコン表示）この状態から復帰するためには、システムモニタ画面の「Reset」ボタンを押すか、一度PL本体の電源を切り、その異常要因を取り除くメンテナンス作業をした後、電源を再投入する必要があります。**

付 .3.4 メッセージ

システムモニタおよび、システムモニタプロパティにて表示されるエラーメッセージ、終了時のメッセージ内容を表記します。

■システムモニタ

◆異常時ポップアップメッセージ

Error ActionでPopup Messageを有効にし、エラーが発生した場合、出荷状況では以下のメッセージがポップアップメッセージの出力画面に表示されます。

エラー発生場所	メッセージ
CPU電圧	"CPU Power Supply Error"
電圧+3.3V	"+3.3V Power Supply Error"
電圧+5.0V	"+5V Power Supply Error"
電圧+12V	"+12V Power Supply Error"
電圧-12V	"-12V Power Supply Error"
電圧-5V	"-5V Power Supply Error"
CPU電圧 2	"VIT Power Supply Error"
Power FAN	"Power FAN Error"
CPU FAN	"CPU or OPTION FAN Error"
温度 System	"System Temperature Error"
温度 CPU or Option	"CPU Temperature Error"
Universal Input 0	"Universal Input 0"
Universal Input 1	"Universal Input 1"
ウォッチドッグ	"Watch Dog Timer Error"
ソフトミラー	"A Mirror disk error occurred"
ハードミラー	"A Mirror disk error occurred"
バックライト	"Back Light Blowout Error"
SMART	"SMART Error"

◆ドライバ　非動作エラー

"The system monitor driver not found."

"Install the latest driver."

◆ドライババージョンエラー

"The old system monitor driver version."

"Update the driver."

◆2 重起動メッセージ

"System monitor has started. "

"Terminate the system monitor in starting."

◆終了確認メッセージ

"The system monitor is terminated."

"Are you sure?"

■システムモニタプロパティ

◆2 重起動メッセージ

"System monitor property has started."

"Terminate the system monitor property in starting."

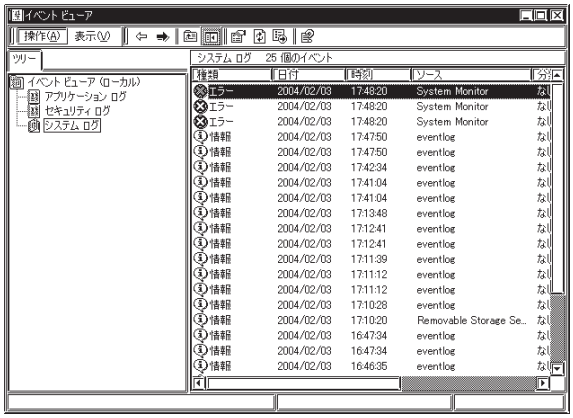
◆終了確認メッセージ

"Save Changes to the registry?"

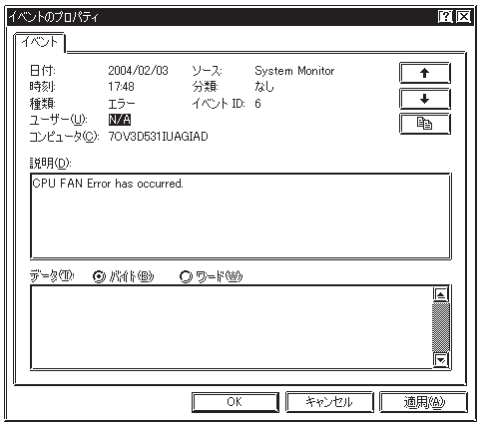
付 .3.5 イベントビューアを使用したエラーの表示

エラー発生場所とエラー発生時の動作はシステムログにエラーイベントとして記録されます。
エラーイベントの内容はイベントビューアで確認します。

■エラーメッセージの表示



①[コントロールパネル]->[管理ツール]->[イベントビューア]を起動し、[システムログ]を選択します。



②System Monitor のエラーを選択し、[プロパティ]アイコンをクリックします。[イベントのプロパティ] ダイアログボックスの[説明]にエラーメッセージが表示されます。

◆エラー発生場所

イベントビューアで表示されるエラー発生場所は次のとおりです。

エラー発生場所	エラーメッセージ
+3.3V	+3.3V Error has occurred.
+5.0V	+5.0V Error has occurred.
+12V	+12V Error has occurred.
-12V	-12V Error has occurred.
-5.0V	-5.0V Error has occurred.
CPU valtage	CPU voltage Error has occurred.
Vit valtage	Vit valtage Error has occurred.
CPU FAN	CPU FAN Error has occurred.
Power FAN	Power FAN Error has occurred.
CPU Temp	CPU Temperature Error has occurred.
System Temp	System Temperature Error has occurred.
Universal Input 0	Universal Input 0 Error has occurred.
Universal Input 1	Universal Input 1 Error has occurred.
Watch Dog Timer	Watch Dog Timer Error has occurred.
Backlight	Backlight Error has occurred.
Soft Mirror	Mirror Disk Error has occurred.
Hard Mirror	Mirror Disk Error has occurred.
SMART	SMART Error has occurred. Attribute (No.) (Attribute Name) Device (No.) (HD Model). ( ) 内は発生するエラーの内容およびエラーが発生したデバイス（0：マスター、1：スレーブ）によって異なります。

◆エラー発生時の動作

イベントビューアで表示されるエラー発生時の動作は次のとおりです。

MEMO
・表中の□□はエラー発生場所を表します。
・エラー発生時の動作はシステムモニタプロパティで設定します。

エラー発生時の動作	エラーメッセージ
Buzzer	Buzzer has sounded because of □□ error.
Popup Message	Popup message has been shown because of □□ error.
OS Shutdown	Windows has been shut down because of □□ error.
ALARM	ALARM has output because of □□ error.
LAMP	LAMP has output because of □□ error.

+3.3V にエラーが発生してブザーが鳴った場合、イベントビューアには「+3.3V Error has occurred.」と「Buzzer has sounded because of +3.3V error.」の２つのエラーが表示されます。

付 .4 システムモニタ /RAS 機能 API-DLL

付 .4.1 動作環境

システムモニタ /RAS機能をPL-X920シリーズ上で動作させるためのダイナミックリンクライブラリ(API-DLL)について説明します。

API-DLL は、アプリケーケーションからシステムモニタ /RAS 機能を「システムモニタ /RAS デバイスドライバ」経由でアクセスするためのインターフェースを提供します。アプリケーションは、この DLL を経由し、以下の機能を使用することが可能になります。

1. ドライバのバージョン管理
2. システムモニタ監視状態
3. 監視用パラメータ取得(電圧、ファン、温度)
4. システムモニタ現在情報(電圧、ファン、温度)
5. ウォッチドッグパラメータ
6. 警告処理
7. 汎用入力処理
8. リセット処理
9. ミラーリング処理
10. イベント処理

■オペレーティングシステム

CD-ROM に付属の API-DLL が動作する OS は以下のとおりです。

- Microsoft Windows® 95
- Microsoft Windows® 98
- Microsoft WindowsNT® 4.0
- Microsoft Windows® 2000
- Microsoft Windows® XP

また、それぞれのOS用の「システムモニタ /RASデバイスドライバ」が動作していなければなりません。

■対応言語

- Microsoft Visual C
- Microsoft Visual C++
- Microsoft Visual Basic

◆必要ファイル

このDLLを使用すためには、各開発言語毎に以下のファイルが必要です。

・Visual C

ファイル名	説明
PL_Iocif.h	ドライバインタフェイス定義インクルードファイル
PL_Ioc.LIB	ライブラリ定義ファイル
PL_Ioc.dll	ダイナミックリンクライブラリファイル

・Visual C++

ファイル名	説明
PL_Iocif.h	ドライバインタフェイス定義インクルードファイル
PL_Iocall.h	CPL_Iocallクラス定義インクルードファイル
PL_Ioctl.h	CPL_Ioctlクラス定義インクルードファイル
PL_Ioc.LIB	ライブラリ定義ファイル
PL_Ioc.dll	ダイナミックリンクライブラリファイル
Sm.h	ソフトミラー定義ファイル(ソフトミラー使用時のみ)
PL_SmiIoctl.h	CPL_SmiIocrlクラス定義インクルードファイル (ソフトミラー使用時のみ)

*インクルードするヘッダファイルの順番は以下の通りです。

#include PL_Iocif.h

#include PL_Ioctl.h

PL_Iocall.hは自動でインクルードされるので、直接インクルードしないでください。

・Visual Basic

ファイル名	説明
PL_Ioc.bas	ドライバインタフェイス定義ファイル
PL_Ioc.LIB	ライブラリ定義ファイル
PL_Ioc.dll	ダイナミックリンクライブラリファイル

■ Dynamic Link Library(DLL)

作成したアプリケーションからPL_Ioc.dllを使用するために、以下の位置にDLLを格納する必要があります。

OS	位置
Windows 95/Windows 98	C:\Windows\System
Windows NT/Windows 2000	C:\Winnt\System32
Windows XP	C:\Windows\System32

付 .4.2 クラス内容

■ CPL_Ioctl クラス

CPL_Ioctl クラスは CPL_Ioctl クラスでデバイスドライバアクセスするためのパラメータをセットします。

キーワード	型	変数名	説明
public	HANDLE	m_Drvhandle	デバイスドライバハンドル

■ CPL_Iocall クラス

CPL_Ioctlでセットされたパラメータを使用し、DeviceIoControl(ドライバアクセス関数)を呼び出します。

ただし、このクラスはCPL_Ioctlから継承されているので直接使用することはありません。

キーワード	型	変数名	説明
public	HANDLE	m_h	デバイスドライバハンドル
public	LONG	m_long	実行する操作の制御コード
public	void *	m_ibp	入力データバッファアドレス
public	ULONG	m_ibsize	入力データバッファサイズ
public	void *	m_obp	出力データバッファアドレス
public	ULONG	m_obsize	出力データバッファサイズ
public	DWORD	m_retsize	実際の出力バイト数のアドレス
public	LPOVERLAPPED	m_ovlp	オーバーラップ構造体のアドレス

■ CPL_Smiloctl クラス

CPL_SmiIoctlクラスは、CPL_SmiIoctlクラスでデバイスドライバアクセスをするためのパラメータをセットします。

ソフトミラードライバを使用する場合にのみ、使用します。

キーワード	型	変数名	説明
public	HANDLE	m_Drvhandle	デバイスドライバハンドル

付 .4.3 Visual C 用関数仕様一覧

関数名	説明
InitIoctl	CPL_Ioctlオブジェクト作成
EndIoctl	CPL_Ioctlオブジェクト破棄
GetDrvHandle	ドライバハンドル取得
CloseDrvHandle	GetDrvHandle取得ハンドル破棄
GetDrvVersion	ドライババージョン取得
GetMonitorSetup	モニタ許可／禁止設定取得
GetVoltParam	電圧監視用パラメータ取得
GetCurrentVolt	現在電圧値取得
GetFanParam	FAN監視用パラメータ取得
GetCurrentFan	現在FAN値取得
GetTempParam	温度監視用パラメータ取得
GetCurrentTemp	現在温度値取得
SetWdtCounter	ウォッチドックタイマカウンタ値設定
GetWdtCounter	ウォッチドックタイマカウンタ取得
SetWdtMask	ウォッチドックタイマタイムアウト時の警告マスク設定
GetWdtMask	ウォッチドッグタイマタイムアウト時の警告マスク取得
StartWdt	ウォッチドッグタイマ開始
StopWdt	ウォッチドッグタイマ停止
RestartWdt	ウォッチドッグタイマ再開
RunningWdt	ウォッチドッグタイマ動作状況取得
SetWarningOut	警告出力設定
GetWarningOut	警告出力取得
GetUniversalIn	汎用入力取得
ClearUniversalIn	汎用入力ラッチ状態解除
SetUniversalInMask	汎用入力マスク設定
GetUniversalInMask	汎用入力マスク取得
SetResetMask	リセットマスク設定
GetResetMask	リセットマスク取得
SetIdeErr	ミラーリングエラー（ソフト）設定
GetIdeErrHard	ミラーリングエラー（ハード）取得
GetLightblowErr *1	バックライト切れ状態取得
GetEvent	エラーイベント取得
ClearEvent	エラーイベント消去
StartInsideBuzzer	内部Buzzer開始
StopInsideBuzzer	内部Buzzer停止
ChkInsideBuzzer	内部Buzzer状態チェック
GetWdtTimeout	ウォッチドッグタイマのタイムアウト状態取得
ClearWdtTimeout	ウォッチドッグタイマのタイムアウト状態クリア
SetWarningDOUT	警告出力DOUT設定
GetWarningDOUT	警告出力DOUT取得
GetSmiDrvHandle	SoftMirror ドライバハンドル取得
CloseSmiDrvHandle	SoftMirror ドライバハンドル破棄
GetSmiAryStatus	SoftMirror Array Status 取得
GetSmiDevStatus	SoftMirror Device Status 取得
SetWdtResetMask	リセットマスク設定
GetWdtResetMask	リセットマスク取得

*1 PL-6920 シリーズでのみ使用できます。

付 .4.4 Visual C 用関数仕様詳細

InitIoctl

・呼び出し形式　　void WINAPI InitIoctl(void)
・戻り値　　なし
・引数　　なし
・処理概要　　CPL_Loctlオブジェクトを作成する。作成されたオブジェクトはEndloctl関数が呼ばれるまで破棄されない。
・例　　InitIoctl();

EndIoctl

・呼び出し形式　　void WINAPI EndIoctl(void)
・戻り値　　なし
・引数　　なし
・処理概要　　InitIoctl関数で作成したオブジェクトを破棄する。
・例　　EndIoctl();

GetDrvHandle

・呼び出し形式　　int WINAPI GetDrvHandle(HANDLE * pHndl)
・戻り値　　0:正常
1：エラー
・引数　　(I/O) HANDLE *pHndl　　デバイスドライバハンドルへのポインタ
・処理概要　　デバイスドライバとのやり取りを行なうためのデバイスドライバハンドルを取得する。
・例

```
int   ret;
HANDLE     hndl;
ret = GetDrvHandle( &hndl );
```

・ システムモニタ/RASデバイスドライバが動作していない場合はエラーになります。

CloseDrvHandle

・ 呼び出し形式　　BOOL WINAPI CloseDrvHandle(void)
・ 戻り値　　TRUE:正常
FALSE:エラー
・引数　　なし
・ 処理概要　　GetDrvHandle関数で取得したハンドルを破棄する。
・例

```
BOOL  ret;
// ハンドル破棄
ret = CloseDrvHandle();
```

GetDrvVersion

・呼び出し形式　BOOL WINAPI GetDrvVersion(int *pMajor, int *pMinor)

・戻り値　TRUE:正常
FALSE：エラー

・引数　(I/O) int *pMajor　バージョン情報(Major,0～99)へのポインタ
(I/O) int *pMinor　バージョン情報(Minor,0～99)へのポインタ

・処理概要　ドライババージョン情報を取得する。

・例
```
BOOL  ret;
int   Major, Minor;
ret = GetDrvVersion( &Major, &Minor );
```

・ バージョンが 1.10 の場合は、
Major:1　(10 進数)
Minor：10　(10 進数)
となります。

GetMonitorSetup

・呼び出し形式　BOOL WINAPI GetMonitorSetup(int Selector, int *pSetup)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I) int Selector　取得パラメータ

MONITOR_VOLT_CPU	CPU コア電圧
MONITOR_VOLT_P33	+3.3V 電圧
MONITOR_VOLT_P50	+5.0V 電圧
MONITOR_VOLT_P12	+12V 電圧
MONITOR_VOLT_M12	-12V 電圧
MONITOR_VOLT_M50	-5.0V 電圧
MONITOR_VOLT_VIT	CPU コア電圧 2
MONITOR_TEMP_SYSTEM	SYSTEM 温度
MONITOR_TEMP_CPU	CPU 温度
MONITOR_TEMP_OPT	OPTION 温度
MONITOR_FAN_CPU	CPU FAN
MONITOR_FAN_POWER	POWER FAN
MONITOR_FAN_OPT	OPTION FAN

(I/O) int *pSetup　取得データへのポインタ
0:Disable
1:Enable

・処理概要　現在のモニタ許可／禁止状態を取得する。

・例
```
BOOL  ret;
int   Setup;
// CPU コア電圧セットアップ状態取得
ret = GetMonitorSetup( MONITOR_VOLT_CPU, &Setup );
```

GetVoltParam

・呼び出し形式　BOOL WINAPI GetVoltParam
(int Selector, int *pULimit, int *pLLimit)

・戻り値　TRUE:正常
FALSE:エラー

・ 引数　(I) int Selector　取得パラメータ

MONITOR_VOLT_CPU	CPU コア電圧
MONITOR_VOLT_P33	+3.3V 電圧
MONITOR_VOLT_P50	+5.0V 電圧
MONITOR_VOLT_P12	+12V 電圧
MONITOR_VOLT_M12	-12V 電圧
MONITOR_VOLT_M50	-5.0V 電圧
MONITOR_VOLT_VIT	CPU コア電圧 2

(I/O) int *pULimit　電圧上限値(単位:mV)へのポインタ
(I/O) int *pLLimit　電圧下限値(単位:mV)へのポインタ

・処理概要　電圧監視用パラメータを取得する。

・例
```
BOOL  ret;
int   ULimit, LLimit;
// CPU コア電圧上限下限値取得
ret = GetVoltParam( MONITOR_VOLT_CPU, &ULimit, &LLimit );
```

・ 関数から取得されたデータはmV(ミリボルト)単位になっているためV(ボルト)単位で使用する時は下記のような変換をする必要があります。
ボルト単位データ = ミリボルト単位データ /1000

GetCurrentVolt

・呼び出し形式　BOOL WINAPI GetCurrentVolt(int Selector, int *pData)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I) int Selector　取得パラメータ

MONITOR_VOLT_CPU	CPU コア電圧
MONITOR_VOLT_P33	+3.3V 電圧
MONITOR_VOLT_P50	+5.0V 電圧
MONITOR_VOLT_P12	+12V 電圧
MONITOR_VOLT_M12	-12V 電圧
MONITOR_VOLT_M50	-5.0V 電圧
MONITOR_VOLT_VIT	CPU コア電圧 2

(I/O) int *pData 電圧値(単位:mV)へのポインタ

・処理概要　現在の電圧値を取得する。

・例

```
BOOL  ret;
int   Data;
// CPU コア電圧値取得
ret = GetCurrentVolt( MONITOR_VOLT_CPU, &Data );
```

・ 関数から取得されたデータはmV(ミリボルト)単位になっているためV(ボルト)単位で使用する時は下記のような変換をする必要があります。
ボルト単位データ = ミリボルト単位データ /1000

GetFanParam

・呼び出し形式　BOOL WINAPI GetFanParam (int Selector, int *pLLimit)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I) int Selector　取得パラメータ

MONITOR_FAN_CPU	CPU FAN
MONITOR_FAN_POWER	POWER FAN
MONITOR_FAN_OPT	OPTION FAN

(I/O) int *pLLimit　FAN 下限回転数(単位:RPM)へのポインタ
(RPM:1 分あたりの回転数)

・処理概要　FAN 監視用のパラメータを取得する。

・例

```
BOOL  ret;
int   LLimit;
// CPU FAN 下限回転数取得
ret = GetFanParam( MONITOR_FAN_CPU, &LLimit );
```

GetCurrentFan

・呼び出し形式　BOOL WINAPI GetCurrentFan(int Selector, int *pData)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Selector	取得パラメータ	
	MONITOR_FAN_CPU	CPU FAN
	MONITOR_FAN_POWER	POWER FAN
	MONITOR_FAN_OPT	OPTION FAN
(I/O) int *pData	FAN 回転数(単位:RPM)へのポインタ	
	(RPM:1 分あたりの回転数)	

・処理概要　現在の FAN 回転数を取得する。

・例

```
BOOL  ret;
int   Data;
// CPU FAN 回転数取得
ret = GetCurrentFan( MONITOR_FAN_CPU, &Data );
```

GetTempParam

・ 呼び出し形式　BOOL WINAPI GetTempParam(int Selector, int *pULimit)

・ 戻り値　TRUE:正常
FALSE:エラー

・ 引数

(I) int Selector	取得パラメータ	
	MONITOR_TEMP_SYSTEM	SYSTEM 温度
	MONITOR_TEMP_CPU	CPU 温度
	MONITOR_TEMP_OPT	OPTION 温度
(I/O) int *pULimit	温度上限値(単位:℃)へのポインタ	

・ 処理概要　温度監視用のパラメータを取得する。

・例

```
BOOL  ret;
int   ULimit;
// SYSTEM 温度上限値取得
ret = GetTempParam( MONITOR_TEMP_SYSTEM, &ULimit );
```

GetCurrentTemp

・呼び出し形式　BOOL WINAPI GetCurrentTemp(int Selector, int *pData)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Selector	取得パラメータ	
	MONITOR_TEMP_SYSTEM	SYSTEM 温度
	MONITOR_TEMP_CPU	CPU 温度
	MONITOR_TEMP_OPT	OPTION 温度
(I/O) int *pData	温度値(単位:℃)へのポインタ	

・処理概要　現在の温度値を取得する。

・例

```
BOOL  ret;
int   Data;
// SYSTEM 温度値取得
ret = GetCurrentTemp( MONITOR_TEMP_SYSTEM, &Data );
```

SetWdtCounter

・呼び出し形式　BOOL WINAPI SetWdtCounter(int Counter)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Counter	ウォッチドッグタイマ初期カウンタ値 (5 ～ 255)(単位:秒)	

・処理概要　ウォッチドッグタイマの初期カウンタ値を設定する。

・例

```
BOOL  ret;
// ウォッチドッグタイマ初期カウンタ値を10秒に設定
ret = SetWdtCounter( 10 );
```

GetWdtCounter

・呼び出し形式　BOOL WINAPI GetWdtCounter(int *pCounter)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I/O) int *pCounter	ウォッチドッグタイマの初期カウンター値 (単位:秒)へのポインタ	

・処理概要　現在のウォッチドッグタイマの初期カウンタ値を取得する。

・例

```
BOOL  ret;
int   Counter;
ret = GetWdtCounter( &Counter );
```

SetWdtMask

・呼び出し形式　BOOL WINAPI SetWdtMask(int Selector, int Mask)

・戻り値　TRUE:正常
FALSE:エラー

・ 引数

(I) int Selector	設定項目	
	WARNING_LAMP	LAMP
	WARNING_ALARM	ALARM
(I) int Mask	マスク情報	
	MASK_OFF	マスク解除
	MASK_ON	マスク

・処理概要　ウォッチドッグタイマタイムアウト時に出力する警告のマスクを設定する。

・例

```
BOOL  ret;
// LAMP出力をマスクする
ret = SetWdtMask( WARNING_LAMP, MASK_ON );
// ALARM出力のマスクを解除する
ret = SetWdtMask( WARNING_ALARM, MASK_OFF );
```

GetWdtMask

・呼び出し形式　BOOL WINAPI GetWdtMask(int Selector, int *pMask)

・ 戻り値　TRUE:正常

FALSE:エラー

・引数

(I) int Selector	設定項目	
	WARNING_LAMP	LAMP
	WARNING_ALARM	ALARM
(I/O) int *pMask	マスク情報へのポインタ	
	MASK_OFF	マスク解除
	MASK_ON	マスク

・処理概要　ウォッチドッグタイマタイムアウト時の警告出力マスク情報を取得する。

・例

```
BOOL  ret;
int   Mask;
// LAMP のマスク情報取得
ret = GetWdtMask( WARNING_LAMP, &Mask );
// ALARM のマスク情報取得
ret = GetWdtMask( WARNING_ALARM, &Mask );
```

StartWdt

・呼び出し形式　BOOL WINAPI StartWdt (void)

・戻り値　TRUE:正常

FALSE:エラー

・引数　なし

・処理概要　ウォッチドッグタイマのカウントダウンを開始する。

・例

```
BOOL ret:
ret = StartWdt();
```

StopWdt

・呼び出し形式　BOOL WINAPI StopWdt (void)

・戻り値　TRUE:正常

FALSE:エラー

・引数　なし

・処理概要　ウォッチドッグタイマのカウントダウンを停止する。

・例

```
BOOL ret:
ret = StopWdt();
```

RestartWdt

・呼び出し形式　BOOL WINAPI RestartWdt (void)
・戻り値　TRUE:正常
FALSE:エラー
・引数　なし
・処理概要　ウォッチドッグタイマのカウント値を初期値に戻し、再カウントダウンを始める。
・例
```
BOOL ret:
ret = RestartWdt();
```

・ ウォッチドッグタイマが停止状態の場合は、何も処理しません。

RunningWdt

・呼び出し形式　BOOL WINAPI RunningWdt (int *pRunFlag)
・戻り値　TRUE:正常
FALSE:エラー
・引数　(I/O) int *pRunFlag　ウォッチドッグタイマの動作状態へのポインタ

WATCHDOG_STOP	停止中
WATCHDOG_COUNTDOWN	カウントダウン中

・処理概要　ウォッチドッグタイマの動作状態を取得する。
・例
```
BOOL ret；
int RunFlag;
ret = RunningWdt(&RunFlag);
```

SetWarningOut

・呼び出し形式　BOOL WINAPI SetWarningOut (int Selector, int WarnOut)
・戻り値　TRUE:正常
FALSE:エラー
・引数　(I) int Selector　設定項目

WARNING_LAMP	LAMP
WARNING_ALARM	ALARM

(I) int WarnOut　出力状態

OUTPUT_OFF	出力 OFF
OUTPUT_ON	出力 ON

・処理概要　設定項目(LAMP、ALARM)の警告情報を設定する。
・例
```
BOOL ret:
//LAMP の出力状態を ON に設定
ret = SetWarningOut(WARNING_LAMP, OUTPUT_ON);
//ALARM の出力状態を OFF に設定
ret = SetWarningOut(WARNING_ALARM, OUTPUT_OFF);
```

GetWarningOut

・呼び出し形式　BOOL WINAPI GetWarningOut (int Selector, int *pWarnOut)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Selector	設定項目	
	WARNING_LAMP	LAMP
	WARNING_ALARM	ALARM
(I/O) int *pWarnOut	出力状態へのポインタ	
	OUTPUT_OFF	出力 OFF
	OUTPUT_ON	出力 ON

・処理概要　現在の設定項目(LAMP,ALARM)の警告状態を取得する。

・例

```
BOOL ret:
int = WarnOut;
//LAMP の出力状態取得
ret = GetWarningOut (WARNING_LAMP, &WarnOut);
//ALARM の出力状態取得
ret = GetWarningOut (WARNING_ALARM, &WarnOut);
```

GetUniversalIn

・呼び出し形式　BOOL WINAPI GetUniversalIn (int Selector, int *pUniIn)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Selector	対象ポート	
	PORT_UNI0	Universal Input 0
	PORT_UNI1	Universal Input 1
(I/O) int*pUniIn	入力状態へのポインタ	
	INPUT_OFF	入力なし
	INPUT_ON	入力あり

・処理概要　対象ポート(Universal Input 0, Universal Input 1)の入力状態を取得する。

・例

```
BOOL ret;
int UniIn;
//Universal Input 0の入力状態取得
ret = GetUniversalIn(PORT_UNI0, &UniIn);
//Universal Input 1の入力状態
ret = GetUniversalIn(PORT_UNI1, &UniIn);
```

ClearUniversalIn

・呼び出し形式　BOOL WINAPI ClearUniversalIn (int Selector)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I) int Selector　対象ポート
PORT_UNI0　Universal Input 0
PORT_UNI1　Universal Input 1

・処理概要　対象ポート(Universal Input 0, Universal Input 1)の入力状態をキャンセルする。

・例

```
BOOL ret:
//Universal Input 0の入力状態をキャンセルする
ret = ClearUniversalIn(PORT_UNI0);
//Universal Input 1の入力状態をキャンセルする
ret = ClearUniversalIn(PORT_UNI1);
```

SetUniversalInMask

・呼び出し形式　BOOL WINAPI SetUniversalInMask (in Selector, int Mask)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I)　in Selector　対象ポート
PORT_UNI0　Universal Input 0
PORT_UNI1　Universal Input 1
(I)　int Mask　マスク情報
MASK_OFF　マスク解除
MASK_OFF　マスク

・処理概要　対象ポート(Universal Input 0, Universal Input 1)のマスク情報を設定する。

・例

```
BOOL ret；
//Universal Input 0をマスク解除
ret = SetUniversalInMask(PORT_UNI0, MASK_OFF);
//Universal Input 1をマスク
ret = SetUniversalInMask(PORT_UNI1, MASK_ON);
```

GetUniversalInMask

・呼び出し形式　BOOL WINAPI GetUniversalInMask(int Selector, int *pMask)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Selector	対象ポート	
	PORT_UNI0	Universal Input 0
	PORT_UNI1	Universal Input 1
(I/O) int *pMask	マスク情報へのポインタ	
	MASK_OFF	マスク解除
	MASK_ON	マスク

・処理概要　対象ポート(Universal Input 0, Universal Input 1)のマスク情報を取得する。

・例

```
BOOL  ret;
int   Mask;
// Universal Input0 マスク情報取得
ret = GetUniversalInMask( PORT_UNI0, &Mask );
// Universal Input1 マスク情報取得
ret = GetUniversalInMask( PORT_UNI1, &Mask );
```

SetResetMask

・呼び出し形式　BOOL WINAPI SetResetMask (int Mask)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Mask	マスク情報	
	MASK_OFF	マスク解除
	MASK_ON	マスク

・処理概要　リセットマスクを設定する

・例

```
BOOL ret:
//リセットマスク解除
ret = SetResetMask(MASK_OFF);
```

GetResetMask

・呼び出し形式　BOOL WINAPI GetResetMask(int *pMask)

・戻り値　TRUE:正常
FALSE:エラー

・ 引数

(I/O) int *pMask	マスク情報へのポインタ	
	MASK_OFF	マスク解除
	MASK_ON	マスク

・処理概要　現在のリセットマスク情報を取得する。

・例

```
BOOL  ret;
int   Mask;
ret = GetResetMask( &Mask );
```

SetIdeErr

・呼び出し形式　BOOL WINAPI SetIdeErr (int ldeErr)
・戻り値　TRUE:正常
FALSE:エラー
・引数　(I) int IdeErr　エラー出力情報
IDE_ERROR_OFF　エラー出力しない
IDE_ERROR_ON　エラー出力する
・処理概要　ソフトウェア制御での IDE エラー出力を設定する。
・例
```
BOOL ret;
//IDE エラー出力しないように設定
ret = SetIdeErr(IDE_ERROR_OFF);
```

GetIdeErrHard

・呼び出し形式　BOOL WINAPI GetIdeErrHard(int Selector, int *pIdeErr)
・戻り値　TRUE:正常
FALSE:エラー
・引数　(I) int Selector　取得パラメータ
IDE_ERROR_1　IDE_ERR1
IDE_ERROR_2　IDE_ERR2
(I/O) int *pIdeErr　エラー信号へのポインタ
IDE_ERROR_OFF　正常
IDE_ERROR_ON　エラー
・処理概要　現在のハードウェアの出力する IDE エラー信号を取得する。
・例
```
BOOL  ret;
int   IdeErr;
// IDE_ERR1 の信号取得
ret = GetIdeErrHard( IDE_ERROR_1, &IdeErr );
```

GetLightblowErr

・呼び出し形式　BOOL WINAPI GetLightblowErr (int *pLightErr)
・戻り値　TRUE:正常
FALSE:エラー
・引数　(I/O)int *pLightErr　エラー出力情報へのポインタ
BACKLIGHT_OK　バックライト正常
BACKLIGHT_ERR　バックライト管切れ
・処理概要　現在の LCD バックライト管切れエラー出力を取得する。
・例
```
BOOL ret;
int LightErr;
//バックライト管切れ状態取得
ret = GetLightblowErr(&LightErr);
```

・ PL-6920 シリーズでのみ使用できます。

GetEvent

・呼び出し形式　BOOL WINAPI GetEvent (int Selector, int *pEvent)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Selector	取得パラメータ	
	EVENT_VOLT_CPU	CPU コア電圧
	EVENT_VOLT_P33	+3.3V
	EVENT_VOLT_P50	+5.0V
	EVENT_VOLT_P12	+12V
	EVENT_VOLT_M12	-12V
	EVENT_VOLT_M50	-5.0V
	EVENT_VOLT_VIT	CPU コア電圧 2
	EVENT_FAN_CPU	CPU FAN
	EVENT_FAN_POWER	POWER FAN
	EVENT_FAN_OPT	OPTION FAN
	EVENT_TEMP_SYSTEM	SYSTEM 温度
	EVENT_TEMP_CPU_OPT	CPU or OPTION温度
	EVENT_UNI_IN0	Universal Input 0
	EVENT_UNI_IN1	Universal Input 1
	EVENT_WDT_TIMEOUT	Watchdot Timeout
(I/O) int *pEvent	エラーイベント情報へのポインタ	
	ERROR_EVENT_OFF	エラーイベントなし
	ERROR_EVENT_ON	エラーイベントあり

・処理概要　マシンの電圧、FAN、温度の異常、また、Universal Input 動作の情報 (イベント)、Watchdog Timeout 情報をチェックする。

・例

```
BOOL ret;
int Evnet
//CPU コア電圧のエラーイベント情報取得
ret = GetEvent(EVENT_VOLT_CPU, &Event);
```

ClearEvent

・呼び出し形式　BOOL WINAPI ClearEvent (int Selector)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I) int Selector　エラーイベントキャンセル対象パラメータ

EVENT_VOLT_CPU	CPU コア電圧
EVENT_VOLT_P33	+3.3V
EVENT_VOLT_P50	+5.0V
EVENT_VOLT_P12	+12V
EVENT_VOLT_M12	-12V
EVENT_VOLT_M50	-5.0V
EVENT_VOLT_VIT	CPU コア電圧 2
EVENT_FAN_CPU	CPU FAN
EVENT_FAN_POWER	POWER FAN
EVENT_FAN_OPT	OPTION FAN
EVENT_TEMP_SYSTEM	SYSTEM 温度
EVENT_TEMP_CPU_OPT	CPU or OPTION温度
EVENT_UNI_IN0	Universal Input 0
EVENT_UNI_IN1	Universal Input 1
EVENT_WDT_TIMEOUT	Watchdot Timeout

・処理概要　エラーイベントをキャンセルする。

・例
```
BOOL ret;
//CPU コア電圧エラーイベントキャンセル
ret = ClearEvent(EVENT_VOLT_CPU);
```

StartInsideBuzzer

・呼び出し形式　BOOL WINAPI StartInsideBuzzer(int hz, int ms)

・戻り値　BOOL TRUE:正常
FALSE:エラー

・引数
(I) int hz　Buzzer 音周波数(Hz)
(I) int ms　Buzzer 音長(ms)

・処理概要　指定された Buzzer 周波数、Buzzer 音長を元に内部 Buzzer を開始する。

・例
```
BOOL ret;
int  hz = 600;
int  ms = 1000;
//Buzzer 周波数 600Hz を 1 秒間鳴らすよう設定。
ret = StartInsideBuzzer( hz, ms );
```

・ Windows®95,Windows®98 の関数です。Windows NT®4.0、Windows®2000、Windows®XP で使用してもエラーとなります。

StopInsideBuzzer

・呼び出し形式　BOOL WINAPI StopInsideBuzzer(void)
・戻り値　BOOL TRUE:正常
　　FALSE:エラー
・引数　なし
・処理概要　内部 Buzzer を停止する。
・例
```
BOOL ret;
// 内部 Buzzer を停止する
ret = StopInsideBuzzer();
```

・ Windows®95,Windows®98 の関数です。Windows NT®4.0、Windows®2000、Windows®XP で使用してもエラーになります。

ChkInsideBuzzer

・呼び出し形式　BOOL WINAPI ChkInsideBuzzer (int *BuzzerParam)
・戻り値　BOOL TRUE:正常
　　FALSE:エラー
・引数　(I/O) int *BuzzerParam　Buzzer 状態へのポインタ
　　BUZZER_ON　Buzzer 開始中
　　BUZZER_OFF Buzzer 停止中
・処理概要　内部 Buzzer の開始 / 停止状態をチェックする。
・例
```
BOOL ret:
int BuzzerParam;
//Buzzer 状態チェック
ret = ChkInsideBuzzer(&BuzzerParam)
```

・ Windows®95,Windows®98 の関数です。Windows NT®4.0、Windows®2000、Windows®XP で使用してもエラーになります。

GetWdtTimeout

・呼び出し形式　BOOL WINAPI GetWdtTimeout(int *pTimebuf)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I/O) int *pTimebuf　ウォッチドッグタイムアウト状態へのポインタ
TIMEOUT_OK　タイムアウトしていない
TIMEOUT_ERROR　タイムアウトしている

・処理概要　ウォッチドッグのタイムアウト状態を取得する

・例
```
BOOL  ret;
int   Timebuf;
// ウォッチドッグのタイムアウト状態取得
ret = GetWdtTimeout( &Timebuf );
```

ClearWdtTimeout

・呼び出し形式　BOOL WINAPI ClearWdtTimeout (void)

・戻り値　TRUE:正常
FALSE:エラー

・引数　なし

・処理概要　ウォッチドッグのタイムアウト状態をクリアする。

・例
```
BOOL ret;
//ウォッチドッグタイムアウト状態クリア
ret = GetWdtTimeout();
```

SetWarningDOUT

・呼び出し形式　BOOL WINAPI SetWarningDOUT (int WarningOut)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I)　int WarningOut　出力状態
OUTPUT_OFF　出力 OFF
OUTPUT_ON　出力 ON

・処理概要　現在の設定項目(DOUT)の警告状態を設定する。

・例
```
BOOL ret;
//DOUT の出力状態を OFF に設定
ret = SetWarningDOUT(OUTPUT_OFF);
```

GetWarningDOUT

・呼び出し形式　BOOL WINAPI GetWarningDOUT(int* pWarningOut)
・戻り値　TRUE:正常
FALSE:エラー
・引数　(I/O)　int *pWarningOut　出力状態へのポインタ
OUTPUT_OFF　出力 OFF
OUTPUT_ON　出力 ON
・処理概要　現在の設定項目(DOUT)の警告状態を取得する。
・例

```
BOOL ret;
int  WarningOut
//DOUT の出力状態を取得
ret = GetWarningDOUT(&WarningOut);
```

GetSmiDrvHandle

・呼び出し形式　int WINAPI GetSmiDrvHandle (void)
・戻り値　0:正常
1:エラー
・引数　なし
・処理概要　ソフトミラーデバイスドライバとのやり取りを行うためのデバイスドライバハンドルを取得する。
・例

```
int  ret;
ret = GetSmiDrvHandle();
```

・ ソフトミラーデバイスドライバが動作していない場合はエラーになります。

CloseSmiDrvHandle

・呼び出し形式　BOOL WINAPI CloseSmiDrvHandle (void)
・戻り値　TRUE:正常
FALSE:エラー
・引数　なし
・処理概要　GetSmiDrvHandle 関数で取得したハンドルを破棄する。
・例

```
BOOL ret;
// ハンドル破棄
ret = CloseSmiDrvHandle();
```

GetSmiAryStatus

・呼び出し形式　BOOL WINAPI GetSmiAryStatus (int* pStatus)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I/O) int *pStatus	ミラーディスクへのポインタ	
	ARYSTAT_GOOD	正常
	ARYSTAT_NOTEXIST	出力されません
	ARYSTAT_UNCONFIG	未構築状態
	ARYSTAT_REBUILD	再構築中
	ARYSTAT_REDUCE	縮退通
	ARYSTAT_DEAD	ミラー状態破壊

・処理概要　ソフトミラーの状態を取得する。

・例

```
BOOL ret;
int Status;
// ソフトミラーの状態取得
ret = GetSmiAryStatus (&Status);
```

GetSmiDevStatus

・呼び出し形式　BOOL WINAPI GetSmiDevStatus (int Id, int* pType, int* pStatus)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Id	デバイス ID	
	0：Master HDD	
	1：Slave HDD	
(I/O)int* pType	デバイスタイプ	
	ATADEVICE	ATA デバイス
	ATAPIDEVICE	CD-ROM
	UNKNOWNDEVICE	不明なデバイス
	NODEVICE	未接続
(I/O)int* pStatus	デバイスステータス	
	DEVSTAT_GOOD	正常
	DEVSTAT_NOTEXIST	未接続
	DEVSTAT_BROKEN	故障

・処理概要　ソフトミラーのデバイス状態を取得する。

・例

```
BOOL ret;
int   Id, Type, Status
// デバイス状態の取得
Id = 0;
ret = GetSmiDevStatus(Id, &Type, &Status);
```

SetWdtResetMask

・呼び出し形式　BOOL WINAPI SetWdtResetMask(int Mask)
・戻り値　TRUE:正常
FALSE:エラー
・引数　(I/O) int Mask　マスク情報
Mask_OFF　マスク解除
Mask_ON　マスク
・処理概要　WDT タイムアウト時の H/W リセットマスクの設定。
・例
```
BOOL ret;
//WDTタイムアウト時のリセットをマスク解除する
ret = SetWdtResetMask(MASK_OFF);
```

GetWdtResetMask

・呼び出し形式　BOOL WINAPI GetWdtResetMask (int* pMask)
・戻り値　TRUE:正常
FALSE:エラー
・引数　(I/O) int* pMask　マスク情報へのポインタ
MASK_OFF　マスク解除
MASK_ON　マスク
・処理概要　WDT タイムアウト時の H/W リセットマスク情報の取得
・例
```
BOOL ret;
int Mask;
//WDTタイムアウト時のリセットマスク情報取得
ret = GetWdtResetMask(&Mask);
```

付 .4.5 Visual C++ 用関数一覧

関数名	説明
GetDrvHandle	ドライバハンドル取得
CloseDrvHandle	GetDrvHandle取得ハンドル破棄
GetDrvVersion	ドライババージョン取得
GetMonitorSetup	モニタ許可／禁止設定取得
GetVoltParam	電圧監視用パラメータ取得
GetCurrentVolt	現在電圧値取得
GetFanParam	FAN監視用パラメータ取得
GetCurrentFan	現在FAN値取得
GetTempParam	温度監視用パラメータ取得
GetCurrentTemp	現在温度値取得
SetWdtCounter	ウォッチドッグタイマカウンタ値設定
GetWdtCounter	ウォッチドッグタイマカウンタ取得
SetWdtMask	ウォッチドッグタイマタイムアウト時の警告マスク設定
GetWdtMask	ウォッチドッグタイマタイムアウト時の警告マスク取得
StartWdt	ウォッチドッグタイマ開始
StopWdt	ウォッチドッグタイマ停止
RestartWdt	ウォッチドッグタイマ再開
RunningWdt	ウォッチドッグタイマ動作状況取得
SetWarningOut	警告出力設定
GetWarningOut	警告出力取得
GetUniversalIn	汎用入力取得
ClearUniversalIn	汎用入力ラッチ状態解除
SetUniversalInMask	汎用入力マスク設定
GetUniversalInMask	汎用入力マスク取得
SetResetMask	リセットマスク設定
GetResetMask	リセットマスク取得
SetIdeErr	ミラーリングエラー（ソフト）設定
GetIdeErrHard	ミラーリングエラー（ハード）取得
GetLightblowErr *1	バックライト切れ状態取得
GetEvent	エラーイベント取得
ClearEvent	エラーイベント消去
StartInsideBuzzer	内部Buzzer開始
StopInsideBuzzer	内部Buzzer停止
ChkInsideBuzzer	内部Buzzer状態チェック
GetWdtTimeout	ウォッチドッグタイマのタイムアウト状態取得
ClearWdtTimeout	ウォッチドッグタイマのタイムアウト状態クリア
SetWarningDOUT	警告出力DOUT設定
GetWarningDOUT	警告出力DOUT取得
GetSmiDrvHandle	SoftMirror ドライバハンドル取得
CloseSmiDrvHandle	SoftMirror ドライバハンドル破棄
GetSmiAryStatus	SoftMirror Array Status 取得
GetSmiDevStatus	SoftMirror Device Status 取得
SetWdtResetMask	リセットマスク設定
GetWdtResetMask	リセットマスク取得

*1 PL-6920 シリーズでのみ使用できます。

付 .4.6 Visual C++ 用関数仕様詳細

GetDrvHandle

・呼び出し形式　int GetDrvHandle(void) または int GetDrvHandle(HANDLE *pHndl)

・戻り値　0:正常
1:エラー

・引数　なし

・処理概要　デバイスドライバとのやり取りを行なうためのデバイスドライバハンドルを取得する。取得されたハンドルはメンバ変数m_handleに格納される。

・例 1
```
CPL_Ioctl  m_Ioc;
m_Ioc.GetDrvHandle();
```

・例 2
```
int   ret;
HANDLE     hndl;
ret = ::GetDrvHandle( &hndl );
```

・システムモニタ/RASデバイスドライバが動作していない場合はエラーになります。

CloseDrvHandle

・呼び出し形式　BOOL CloseDrvHandle(void)

・戻り値　TRUE:正常
FALSE:エラー

・引数　なし

・処理概要　GetDrvHandle関数で取得したハンドルを破棄する。

・例 1
```
CPL_Ioctl  m_Ioc;
BOOL       ret;
// ハンドル破棄
ret = m_Ioc.CloseDrvHandle();
```

・例 2
```
BOOL  ret
// ハンドル破棄
ret = ::CloseDrvHandle();
```

GetDrvVersion

・呼び出し形式　BOOL GetDrvVersion(int *pMajor, int *pMinor)

・戻り値　TRUE：正常
FALSE：エラー

・引数

(I/O)	int *pMajor	バージョン情報(Major,0 ～ 99)へのポインタ
(I/O)	int *pMinor	バージョン情報(Minor,0 ～ 99)へのポインタ

・処理概要　ドライババージョン情報を取得する。

・例 1
```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int        Major, Minor;
ret = m_Ioc.GetDrvVersion( &Major, &Minor );
```

・例 2
```
BOOL  ret;
int   Major, Minor;
ret = ::GetDrvVersion( &Major, &Minor );
```

・バージョンが 1.10 の場合は
Major:1　(10 進数)
Minor:10　(10 進数)
となります。

GetMonitorSetup

・呼び出し形式　BOOL GetMonitorSetup(int Selector, int *pSetup)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Selector	取得パラメータ	
	MONITOR_VOLT_CPU	CPU コア電圧
	MONITOR_VOLT_P33	+3.3V 電圧
	MONITOR_VOLT_P50	+5.0V 電圧
	MONITOR_VOLT_P12	+12V 電圧
	MONITOR_VOLT_M12	-12V 電圧
	MONITOR_VOLT_M50	-5.0V 電圧
	MONITOR_TEMP_SYSTEM	SYSTEM 温度
	MONITOR_TEMP_CPU	CPU 温度
	MONITOR_TEMP_OPT	OPTION 温度
	MONITOR_FAN_CPU	CPU FAN
	MONITOR_FAN_POWER	POWER FAN
	MONITOR_FAN_OPT	OPTION FAN
	MONITOR_VOLT_VIT	CPU コア電圧 2
(I/O) int *pSetup	取得データへのポインタ	
	0:Disable	
	1:Enable	

・処理概要　現在のモニタ許可／禁止状態を取得する。

・例 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int        Setup;
// CPU コア電圧取得セットアップ状態取得
ret = m_Ioc.GetMonitorSetup( MONITOR_VOLT_CPU, &Setup );
```

・例 2

```
BOOL  ret;
int   Setup;
// CPU コア電圧取得セットアップ状態取得
ret = ::GetMonitorSetup( MONITOR_VOLT_CPU, &Setup );
```

GetVoltParam

・呼び出し形式　BOOL GetVoltParam (int Selector, int *pULimit, int *pLLimit)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Selector	取得パラメータ	
	MONITOR_VOLT_CPU	CPU コア電圧
	MONITOR_VOLT_P33	+3.3V 電圧
	MONITOR_VOLT_P50	+5.0V 電圧
	MONITOR_VOLT_P12	+12V 電圧
	MONITOR_VOLT_M12	-12V 電圧
	MONITOR_VOLT_M50	-5.0V 電圧
	MONITOR_VOLT_VIT	CPU コア電圧 2
(I/O) int *pULimit	電圧上限値(単位:mV)へのポインタ	
(I/O) int *pLLimit	電圧下限値(単位:mV)へのポインタ	

・処理概要　電圧監視用パラメータを取得する。

・例 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int        ULimit, LLimit;
// CPU コア電圧上限下限値取得
ret = m_Ioc.GetVoltParam( MONITOR_VOLT_CPU, &ULimit, &LLimit );
```

・例 2

```
BOOL  ret;
int   ULimit, LLimit;
// CPU コア電圧上限下限値取得
ret = ::GetVoltParam( MONITOR_VOLT_CPU, &ULimit, &LLimit );
```

MEMO ・関数から取得されたデータはmV(ミリボルト)単位になっています。V(ボルト)単位で使用する時は下記のような変換をする必要があります。
ボルト単位データ = ミリボルト単位データ /1000

GetCurrentVolt

・呼び出し形式　BOOL GetCurrentVolt(int Selector, int *pData)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I) int Selector　取得パラメータ

MONITOR_VOLT_CPU	CPU コア電圧
MONITOR_VOLT_P33	+3.3V 電圧
MONITOR_VOLT_P50	+5.0V 電圧
MONITOR_VOLT_P12	+12V 電圧
MONITOR_VOLT_M12	-12V 電圧
MONITOR_VOLT_M50	-5.0V 電圧
MONITOR_VOLT_VIT	CPU コア電圧 2

(I/O) int *pData 電圧値(単位:mV)へのポインタ

・処理概要　現在の電圧値を取得する。

・例 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int        Data;
// CPU コア電圧値取得
ret = m_Ioc.GetCurrentVolt( MONITOR_VOLT_CPU, &Data );
```

・例 2

```
BOOL  ret;
int   Data;
// CPU コア電圧値取得
ret = ::GetCurrentVolt( MONITOR_VOLT_CPU, &Data );
```

MEMO

・ 関数から取得されたデータはmV(ミリボルト)単位になっています。V(ボルト) 単位で使用する時は下記のような変換をする必要があります。
ボルト単位データ = ミリボルト単位データ /1000

GetFanParam

・呼び出し形式　BOOL GetFanParam (int Selector, int *pLLimit)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Selector	取得パラメータ	
	MONITOR_FAN_CPU	CPU FAN
	MONITOR_FAN_POWER	POWER FAN
	MONITOR_FAN_OPT	OPTION FAN
(I/O) int *pLLimit	FAN 下限回転数(単位:RPM)へのポインタ (RPM:1 分あたりの回転数)	

・処理概要　FAN 監視用のパラメータを取得する。

・例 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int        LLimit;
// CPU FAN 下限回転数取得
ret = m_Ioc.GetFanParam( MONITOR_FAN_CPU, &LLimit );
```

・例 2

```
BOOL  ret;
int   LLimit;
// CPU FAN 下限回転数取得
ret = ::GetFanParam( MONITOR_FAN_CPU, &LLimit );
```

GetCurrentFan

・呼び出し形式　BOOL GetCurrentFan(int Selector, int *pData)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Selector	取得パラメータ	
	MONITOR_FAN_CPU	CPU FAN
	MONITOR_FAN_POWER	POWER FAN
	MONITOR_FAN_OPT	OPTION FAN
(I/O) int *pData	FAN 回転数(単位:RPM)へのポインタ (RPM:1 分あたりの回転数)	

・処理概要　現在の FAN 回転数を取得する。

・例 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int        Data;
// CPU FAN 回転数取得
ret = m_Ioc.GetCurrentFan( MONITOR_FAN_CPU, &Data );
```

・例 2

```
BOOL  ret;
int   Data;
// CPU FAN 回転数取得
ret = ::GetCurrentFan( MONITOR_FAN_CPU, &Data );
```

GetTempParam

・呼び出し形式　BOOL GetTempParam(int Selector, int *pULimit)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I) int Selector　取得パラメータ
MONITOR_TEMP_SYSTEM　SYSTEM 温度
MONITOR_TEMP_CPU　CPU 温度
MONITOR_TEMP_OPT　OPTION 温度
(I/O) int *pULimit　温度上限値(単位:℃)へのポインタ

・処理概要　温度監視用のパラメータを取得する。

・例 1
```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int        ULimit;
// SYSTEM 温度上限値取得
ret = m_Ioc.GetTempParam( MONITOR_TEMP_SYSTEM, &ULimit );
```

・例 2
```
BOOL  ret;
int   ULimit;
ret = ::GetTempParam( MONITOR_TEMP_SYSTEM, &ULimit );
```

GetCurrentTemp

・呼び出し形式　BOOL GetCurrentTemp(int Selector, int *pData)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I) int Selector　取得パラメータ
MONITOR_TEMP_SYSTEM　SYSTEM 温度
MONITOR_TEMP_CPU　CPU 温度
MONITOR_TEMP_OPT　OPTION 温度
(I/O) int *pData　温度値(単位:℃)へのポインタ

・処理概要　現在の温度値を取得する。

・例 1
```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int        Data;
// SYSTEM 温度値取得
ret = m_Ioc.GetCurrentTemp( MONITOR_TEMP_SYSTEM, &Data );
```

・例 2
```
BOOL  ret;
int   Data;
// SYSTEM 温度値取得
ret = ::GetCurrentTemp( MONITOR_TEMP_SYSTEM, &Data );
```

SetWdtCounter

・呼び出し形式　BOOL SetWdtCounter(int Counter)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I) int Counter　ウォッチドッグタイマ初期カウンタ値 (5 ～ 255)(単位:秒)

・処理概要　ウォッチドッグタイマの初期カウンタ値を設定する。

・例 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
// ウォッチドッグタイマ初期カウンタ値を 10 秒に設定
ret = m_Ioc.SetWdtCounter( 10 );
```

・例 2

```
BOOL  ret;
// ウォッチドッグタイマ初期カウンタ値を 10 秒に設定
ret = ::SetWdtCounter( 10 );
```

GetWdtCounter

・呼び出し形式　BOOL GetWdtCounter(int *pCounter)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I/O) int *pCounter　ウォッチドッグタイマの初期カウンター値 (単位:秒)へのポインタ

・処理概要　現在のウォッチドッグタイマの初期カウンタ値を取得する。

・例 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int        Counter;
ret = m_Ioc.GetWdtCounter( &Counter );
```

・例 2

```
BOOL  ret;
int   Counter;
ret = ::GetWdtCounter( &Counter );
```

SetWdtMask

・呼び出し形式　BOOL SetWdtMask(int Selector, int Mask)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Selector	設定項目	
	WARNING_LAMP	LAMP
	WARNING_ALARM	ALARM
(I) int Mask	マスク情報	
	MASK_OFF	マスク解除
	MASK_ON	マスク

・処理概要　ウォッチドッグタイマタイムアウト時に出力する警告のマスクを設定する。

・例 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
// LAMP 出力をマスクする
ret = m_Ioc.SetWdtMask( WARNING_LAMP, MASK_ON );
// ALARM 出力のマスクを解除する
ret = m_Ioc.SetWdtMask( WARNING_ALARM, MASK_OFF );
```

・例 2

```
BOOL  ret;
// LAMP 出力をマスクする
ret = ::SetWdtMask( WARNING_LAMP, MASK_ON );
// ALARM 出力のマスクを解除する
ret = ::SetWdtMask( WARNING_ALARM, MASK_OFF );
```

GetWdtMask

・呼び出し形式　BOOL GetWdtMask(int Selector, int *pMask)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Selector	設定項目	
	WARNING_LAMP	LAMP
	WARNING_ALARM	ALARM
(I/O) int *pMask	マスク情報へのポインタ	
	MASK_OFF	マスク解除
	MASK_ON	マスク

・処理概要　ウォッチドッグタイマタイムアウト時の警告出力マスク情報を取得する。

・例 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int        Mask;
// LAMP のマスク情報取得
ret = m_Ioc.GetWdtMask( WARNING_LAMP, &Mask );
// ALARM のマスク情報取得
ret = m_Ioc.GetWdtMask( WARNING_ALARM, &Mask );
```

・例 2

```
BOOL  ret;
int   Mask;
// LAMP のマスク情報取得
ret = ::GetWdtMask( WARNING_LAMP, &Mask );
// ALARM のマスク情報取得
ret = ::GetWdtMask( WARNING_ALARM, &Mask );
```

StartWdt

・呼び出し形式　BOOL StartWdt(void)

・戻り値　TRUE:正常
FALSE:エラー

・引数　なし

・処理概要　ウォッチドッグタイマのカウントダウンを開始する。

・例 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
ret = m_Ioc.StartWdt();
```

・例 2

```
BOOL  ret;
ret = ::StartWdt();
```

StopWdt

・呼び出し形式　BOOL StopWdt(void)
・戻り値　TRUE:正常
FALSE:エラー
・引数　なし
・処理概要　ウォッチドッグタイマのカウントダウンを停止する。
・例 1
```
CPL_Ioctl  m_Ioc;
BOOL       ret;
ret = m_Ioc.StopWdt();
```
・例 2
```
BOOL  ret;
ret = ::StopWdt();
```

RestartWdt

・呼び出し形式　BOOL RestartWdt(void)
・戻り値　TRUE:正常
FALSE:エラー
・引数　なし
・処理概要　ウォッチドッグタイマのカウンタ値を初期値に戻し、再カウントダウンを始める。
・例 1
```
CPL_Ioctl  m_Ioc;
BOOL       ret;
m_Ioc.RestartWdt();
```
・例 2
```
BOOL  ret;
ret = ::RestartWdt();
```

・ ウォッチドッグタイマが停止状態の場合は、何も処理しません。

RunningWdt

・呼び出し形式　BOOL RunningWdt(int *pRunFlag)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I/O) int *pRunFlag　ウォッチドッグタイマの動作状態へのポインタ
WATCHDOG_STOP　停止中
WATCHDOG_COUNTDOWN　カウントダウン中

・処理概要　ウォッチドッグタイマの動作状態を取得する。

・例 1
```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int        RunFlag;
ret = m_Ioc.RunningWdt( &RunFlag );
```

・例 2
```
BOOL  ret;
int   RunFlag;
ret = ::RunningWdt( &RunFlag );
```

SetWarningOut

・呼び出し形式　BOOL SetWarningOut(int Selector, int WarnOut)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I) int Selector　設定項目
WARNING_LAMP　LAMP
WARNING_ALARM　ALARM
(I) int WarnOut　出力状態
OUTPUT_OFF 出力 OFF
OUTPUT_ON　出力 ON

・処理概要　設定項目(LAMP、ALARM)の警告情報を設定する。

・例 1
```
CPL_Ioctl  m_Ioc;
BOOL       ret;
// LAMP の出力状態を ON に設定
ret = m_Ioc.SetWarningOut( WARNING_LAMP, OUTPUT_ON );
// ALARM の出力状態を OFF に設定
ret = m_Ioc.SetWarningOut( WARNING_ALARM, OUTPUT_OFF );
```

・例 2
```
BOOL  ret;
// LAMP の出力状態を ON に設定
ret = ::SetWarningOut( WARNING_LAMP, OUTPUT_ON );
// ALARM の出力状態を OFF に設定
ret = ::SetWarningOut( WARNING_ALARM, OUTPUT_OFF );
```

GetWarningOut

・呼び出し形式　BOOL GetWarningOut(int Selector, int *pWarnOut)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Selector	設定項目	
	WARNING_LAMP	LAMP
	WARNING_ALARM	ALARM
(I/O) int *pWarnOut	出力状態へのポインタ	
	OUTPUT_OFF	出力 OFF
	OUTPUT_ON	出力 ON

・処理概要　現在の設定項目(LAMP,ALARM)の警告状態を取得する。

・例 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int        WarnOut;
// LAMP の出力状態取得
ret = m_Ioc.GetWarningOut( WARNING_LAMP, &WarnOut );
// ALARM の出力状態取得
ret = m_Ioc.GetWarningOut( WARNING_ALARM, &WarnOut );
```

・例 2

```
BOOL  ret;
int   WarnOut;
// LAMP の出力状態取得
ret = ::GetWarningOut( WARNING_LAMP, &WarnOut );
// ALARM の出力状態取得
ret = ::GetWarningOut( WARNING_ALARM, &WarnOut );
```

GetUniversalIn

・呼び出し形式　BOOL GetUniversalIn(int Selector, int *pUniIn)

・戻り値
TRUE:正常
FALSE:エラー

・引数

(I) int Selector	対象ポート	
	PORT_UNI0	Universal Input 0
	PORT_UNI1	Universal Input 1
(I/O) int *pUniIn	入力状態へのポインタ	
	INPUT_OFF	入力なし
	INPUT_ON	入力あり

・処理概要　対象ポート(Universal Input 0, Universal Input 1)の入力状態を取得する。

・例 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int        UniIn;
// Universal Input 0の入力状態取得
ret = m_Ioc.GetUniversalIn( PORT_UNI0, &UniIn );
// Universal Input 1の入力状態取得
ret = m_Ioc.GetUniversalIn( PORT_UNI1, &UniIn );
```

・例 2

```
BOOL ret;
int  UniIn;
// Universal Input 0の入力状態取得
ret = ::GetUniversalIn( PORT_UNI0, &UniIn );
// Universal Input 1の入力状態取得
ret = ::GetUniversalIn( PORT_UNI1, &UniIn );
```

ClearUniversalIn

・呼び出し形式　BOOL ClearUniversalIn(int Selector)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I) int Selector　対象ポート

PORT_UNI0	Universal Input 0
PORT_UNI1	Universal Input 1

・処理概要　対象ポート(Universal Input 0, Universal Input 1)の入力状態をキャンセルする。

・例 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
// Universal Input 0の出力をキャンセルする
ret = m_Ioc.ClearUniversalIn( PORT_UNI0 );
// Universal Input 1の出力をキャンセルする
ret = m_Ioc.ClearUniversalIn( PORT_UNI1 );
```

・例 2

```
BOOL  ret;
// Universal Input 0の出力をキャンセルする
ret = ::ClearUniversalIn( PORT_UNI0 );
// Universal Input 1の出力をキャンセルする
ret = ::ClearUniversalIn( PORT_UNI1 );
```

SetUniversalInMask

・呼び出し形式　BOOL SetUniversalInMask(int Selector, int Mask)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I) int Selector　対象ポート

PORT_UNI0	Universal Input 0
PORT_UNI1	Universal Input 1

(I) int Mask　マスク情報

MASK_OFF	マスク解除
MASK_ON	マスク

・処理概要　対象ポート(Universal Input 0, Universal Input 1)のマスク情報を設定する。

・例 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
// Universal Input 0をマスク解除
ret = m_Ioc.SetUniversalInMask( PORT_UNI0, MASK_OFF );
// Universal Input 1をマスク
ret = m_Ioc.SetUniversalInMask( PORT_UNI1, MASK_ON );
```

・例 2

```
BOOL  ret;
// Universal Input 0をマスク解除
ret = ::SetUniversalInMask( PORT_UNI0, MASK_OFF );
// Universal Input 1をマスク
ret = ::SetUniversalInMask( PORT_UNI1, MASK_ON );
```

GetUniversalInMask

・呼び出し形式　BOOL GetUniversalInMask(int Selector, int *pMask)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Selector	対象ポート	
	PORT_UNI0	Universal Input 0
	PORT_UNI1	Universal Input 1
(I/O) int *pMask	マスク情報へのポインタ	
	MASK_OFF	マスク解除
	MASK_ON	マスク

・処理概要　対象ポート(Universal Input 0, Universal Input 1)のマスク情報を取得する。

・例 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int        Mask;
// Universal Input 0マスク情報取得
ret = m_Ioc.GetUniversalInMask( PORT_UNI0, &Mask );
// Universal Input 1マスク情報取得
ret = m_Ioc.GetUniversalInMask( PORT_UNI1, &Mask );
```

・例 2

```
BOOL  ret;
int   Mask;
// Universal Input 0マスク情報取得
ret = ::GetUniversalInMask( PORT_UNI0, &Mask );
// Universal Input 1マスク情報取得
ret = ::GetUniversalInMask( PORT_UNI1, &Mask );
```

SetResetMask

・呼び出し形式　BOOL SetResetMask(int Mask)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Mask	マスク情報	
	MASK_OFF	マスク解除
	MASK_ON	マスク

・処理概要　リセットマスクを設定する

・例 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
// リセットマスク解除
ret = m_Ioc.SetResetMask( MASK_OFF );
```

・例 2

```
BOOL  ret;
// リセットマスク解除
ret = ::SetResetMask( MASK_OFF );
```

GetResetMask

・呼び出し形式　BOOL GetResetMask(int *pMask)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I/O) int *pMask　マスク情報へのポインタ
MASK_OFF　マスク解除
MASK_ON　マスク

・処理概要　現在のリセットマスク情報を取得する。

・例 1
```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int        Mask;
ret = m_Ioc.GetResetMask( &Mask );
```

・例 2
```
BOOL  ret;
int   Mask;
ret = ::GetResetMask( &Mask );
```

SetIdeErr

・呼び出し形式　BOOL SetIdeErr(int IdeErr)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I) int IdeErr　エラー出力情報
IDE_ERROR_OFF　エラー出力しない
IDE_ERROR_ON　エラー出力する

・処理概要　ソフトウェア制御での IDE エラー出力を設定する。

・例 1
```
CPL_Ioctl  m_Ioc;
BOOL       ret;
// エラー出力しないように設定
ret = m_Ioc.SetIdeErr( IDE_ERROR_OFF );
```

・例 2
```
BOOL  ret;
// エラー出力しないように設定
ret = ::SetIdeErr( IDE_ERROR_OFF );
```

GetIdeErrHard

・呼び出し形式　BOOL GetIdeErrHard(int Selector, int *pIdeErr)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Selector	取得パラメータ	
	IDE_ERROR_1	IDE_ERR1
	IDE_ERROR_2	IDE_ERR2
(I/O) int *pIdeErr	エラー信号へのポインタ	
	IDE_ERROR_OFF	正常
	IDE_ERROR_ON	エラー

・処理概要　現在のハードウェアの出力する IDE エラー信号を取得する。

・例 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int        IdeErr;
// IDE_ERR1 の信号取得
ret = m_Ioc.GetIdeErrHard( IDE_ERROR_1, &IdeErr );
```

・例 2

```
BOOL  ret;
int   IdeErr;
// IDE_ERR1 の信号取得
ret = ::GetIdeErrHard( IDE_ERROR_1, &IdeErr );
```

GetLightblowErr

・呼び戻し形式　BOOL GetLightblowErr(int *pLightErr)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I/O) int *pLightErr	エラー出力情報へのポインタ	
	BACKLIGHT_OK	バックライト正常
	BACKLIGHT_ERR	バックライト管切れ

・処理概要　現在の LCD バックライト管切れエラー出力を取得する。

・例 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int        LightErr;
// バックライト管切れ状態取得
ret = m_Ioc.GetLightblowErr( &LightErr );
```

・例 2

```
BOOL  ret;
int   LightErr;
// バックライト管切れ状態取得
ret = ::GetLightblowErr( &LightErr );
```

・ PL-6920 シリーズでのみ使用できます。

GetEvent

・呼び出し形式　BOOL GetEvent(int Selector, int *pEvent)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Selector	取得パラメータ	
	EVENT_VOLT_CPU	CPU コア電圧
	EVENT_VOLT_P33	+3.3V
	EVENT_VOLT_P50	+5.0V
	EVENT_VOLT_P12	+12V
	EVENT_VOLT_M12	-12V
	EVENT_VOLT_M50	-5.0V
	EVENT_VOLT_VIT	CPU コア電圧 2
	EVENT_FAN_CPU	CPU FAN
	EVENT_FAN_POWER	POWER FAN
	EVENT_FAN_OPT	OPTION FAN
	EVENT_TEMP_SYSTEM	SYSTEM 温度
	EVENT_TEMP_CPU_OPT	CPU or OPTION 温度
	EVENT_UNI_IN0	Universal Input 0
	EVENT_UNI_IN1	Universal Input 1
	EVENT_WDT_TIMEOUT	Watchdog Timeout
(I/O) int *pEvent	エラーイベント情報へのポインタ	
	ERROR_EVENT_OFF	エラーイベントなし
	ERROR_EVENT_ON	エラーイベントあり

・処理概要　マシンの電圧 , FAN, 温度の異常、また、Universal Input 動作の情報 (イベント)、WatchDog Timeout 情報をチェックする。

・例 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int        Event;
// CPU コア電圧のエラーイベント情報取得
ret = m_Ioc.GetEvent( EVENT_VOLT_CPU, &Event );
```

・例 2

```
BOOL  ret;
int   Event;
// CPL コア電圧のエラーイベント情報取得
ret = ::GetEvent( EVENT_VOLT_CPU, &Event );
```

ClearEvent

・呼び戻し形式　BOOL ClearEvent(int Selector)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I) int Selector

エラーイベントキャンセル対象パラメータ	
EVENT_VOLT_CPU	CPU コア電圧
EVENT_VOLT_P33	+3.3V
EVENT_VOLT_P50	+5.0V
EVENT_VOLT_P12	+12V
EVENT_VOLT_M12	-12V
EVENT_VOLT_M50	-5.0V
EVENT_VOLT_VIT	CPU コア電圧 2
EVENT_FAN_CPU	CPU FAN
EVENT_FAN_POWER	POWER FAN
EVENT_FAN_OPT	OPTION FAN
EVENT_TEMP_SYSTEM	SYSTEM 温度
EVENT_TEMP_CPU_OPT	CPU or OPTION温度
EVENT_UNI_IN0	Universal Input0
EVENT_UNI_IN1	Universal Input1
EVENT_WDT_TIMEOUT	Watchdog Timeout

・処理概要　エラーイベントをキャンセルする。

・例 1
```
CPL_Ioctl  m_Ioc;
BOOL  ret;
// CPU コア電圧エラーイベントキャンセル
ret = m_Ioc.ClearEvent( EVENT_VOLT_CPU );
```

・例 2
```
BOOL  ret;
// CPU コア電圧エラーイベントキャンセル
ret = ::ClearEvent( EVENT_VOLT_CPU );
```

StartInsideBuzzer

・呼び出し形式　BOOL StartInsideBuzzer(int hz, int ms)
・戻り値　BOOL TRUE:正常
　　FALSE:エラー
・引数　(I) int hz　Buzzer 音周波数(Hz)
　(I) int ms　Buzzer 音長(ms)
・処理概要　指定された Buzzer 周波数、Buzzer 音長を元に内部 Buzzer を開始する。
・例 1

```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int   hz = 600;
int   ms = 1000;
//Buzzer 周波数 600Hz を 1 秒間鳴らすよう設定。
ret = m_Ioc.StartInsideBuzzer( hz, ms );
```

・例 2

```
BOOL  ret;
int   hz = 600;
int   ms = 1000;
//Buzzer 周波数 600Hz を 1 秒間鳴らすよう設定。
ret = ::StartInsideBuzzer( hz, ms );
```

・ Windows®95,Windows®98 の関数です。Windows NT®4.0、Windows®2000、Windows®XP で使用してもエラーになります。

StopInsideBuzzer

・呼び出し形式　BOOL StopInsideBuzzer(void)
・戻り値　BOOL TRUE:正常
　　FALSE:エラー
・引数　なし
・処理概要　内部 Buzzer を停止する。
・例 1

```
CPL_Ioctl  m_Ioc;
BOOL  ret;
// 内部 Buzzer を停止する
ret = m_Ioc.StopInsideBuzzer();
```

・例 2

```
BOOL  ret;
// 内部 Buzzer を停止する
ret = ::StopInsideBuzzer();
```

・ Windows®95,Windows®98 の関数です。Windows NT®4.0、Windows®2000、Windows®XP で使用してもエラーになります。

ChkInsideBuzzer

・呼び出し形式　BOOL ChkInsideBuzzer(int *BuzzerParam)

・戻り値　BOOL TRUE:正常
FALSE:エラー

・引数　(I/O) int *BuzzerParam　Buzzer 状態へのポインタ
BUZZER_ON　Buzzer 開始中
BUZZER_OFF　Buzzer 停止中

・処理概要　内部 Buzzer の開始／停止状態をチェックする。

・例 1
```
CPL_Ioctl  m_Ioc;
BOOL       ret;
Int   BuzzerParam;
//Buzzer 状態チェック
ret = m_Ioc.ChkInsideBuzzer( &BuzzerParam )
```

・例 2
```
BOOL  ret;
//Buzzer 状態チェック
ret = ::ChkInsideBuzzer( &BuzzerParam )
```

・ Windows®95,Windows®98 の関数です。Windows NT®4.0、Windows®2000、Windows®XP で使用してもエラーになります。

GetWdtTimeout

・呼び戻し形式　BOOL GetWdtTimeout(int *pTimebuf)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I/O) int *pTimebuf　ウォッチドッグタイムアウト状態へのポインタ
TIMEOUT_OK　タイムアウトしていない
TIMEOUT_ERROR　タイムアウトしている

・処理概要　ウォッチドッグのタイムアウト状態を取得する。

・例 1
```
CPL_Ioctl  m_Ioc;
BOOL       ret;
int        Timebuf;
// ウォッチドッグのタイムアウト状態取得
ret = m_Ioc.GetWdtTimeout( &Timebuf );
```

・例 2
```
BOOL  ret;
int   Timebuf;
// ウォッチドッグのタイムアウト状態取得
ret = ::GetWdtTimeout( &Timebuf );
```

ClearWdtTimeout

・呼び戻し形式　BOOL ClearWdtTimeout(void)
・戻り値　TRUE:正常
FALSE:エラー
・引数　なし
・処理概要　ウォッチドッグのタイムアウト状態をクリアする。
・例 1
```
CPL_Ioctl  m_Ioc;
BOOL       ret;
// ウォッチドッグのタイムアウト状態クリア
ret = m_Ioc.ClearWdtTimeout();
```
・例 2
```
BOOL  ret;
// ウォッチドッグのタイムアウト状態クリア
ret = ::ClearWdtTimeout()
```

SetWarnigDOUT

・呼び出し形式　BOOL SetWarningDOUT(int WarningOut)
・戻り値　TRUE:正常
FALSE:エラー
・引数　(I) int WarningOut　出力状態
OUTPUT_OFF 出力 OFF
OUTPUT_ON　出力 ON
・処理概要　現在の設定項目(DOUT)の警告状態を設定する。
・例 1
```
CPL_Ioctl  m_Ioc;
BOOL       ret;
// DOUT の出力状態を OFF に設定
ret = m_Ioc.SetWarningDOUT( OUTPUT_OFF );
```
・例 2
```
BOOL  ret;
// DOUT の出力状態を OFF に設定
ret = ::SetWarningDOUT( OUTPUT_OFF );
```

GetWarningDOUT

・呼び出し形式　BOOL GetWarningDOUT(int* pWarningOut)
・戻り値　TRUE:正常
FALSE:エラー
・引数　(I/O) int *pWarningOut　FALSE:エラ出力状態へのポインタ
OUTPUT_OFF 出力 OFF
OUTPUT_ON　出力 ON
・処理概要　現在の設定項目(DOUT)の警告状態を取得する。
・例 1
```
CPL_Ioctl  m_Ioc;
BOOL  ret;
int   WarningOut
// DOUT の出力状態を取得
ret = m_Ioc.GetWarningDOUT( &WarningOut );
```
・例 2
```
BOOL  ret;
int   WarningOut;
// DOUT の出力状態を取得
ret = ::GetWarningDOUT( &WarningOut );
```

GetSmiDrvHandle

・呼び出し形式　int GetSmiDrvHandle(void)
・戻り値　0:正常
1:エラー
・引数　なし
・処理概要　ソフトミラーデバイスドライバとのやり取りを行なうためのデバイスドライバハンドルを取得する。
・例 1
```
CPL_SmiIoctl     m_SmiIoc;
BOOL  ret;
// ソフトミラードライバハンドルの取得
ret = m_SmiIoc.GetSmiDrvHandle();
```
・例 2
```
BOOL  ret;
// ソフトミラードライバハンドルの取得
ret = ::GetSmiDrvHandle();
```

・ ソフトミラーデバイスドライバが動作していない場合はエラーになります。

CloseSmiDrvHandle

・呼び出し形式　BOOL CloseSmiDrvHandle(void)
・戻り値　TRUE:正常
FALSE:エラー
・引数　なし
・処理概要　GetSmiDrvHandle 関数で取得したハンドルを破棄する。
・例 1
```
CPL_SmiIoctl     m_SmiIoc;
BOOL  ret;
// ハンドル破棄
ret = m_SmiIoc.CloseSmiDrvHandle();
```
・例 2
```
BOOL  ret;
// ハンドル破棄
ret = ::CloseSmiDrvHandle();
```

GetSmiAryStatus

・呼び出し形式　BOOL GetSmiAryStatus(int* pStatus)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I/0) int *pStatus　ミラーステータスへのポインタ

ARYSTAT_GOOD	正常
ARYSTAT_NOTEXIST	出力されません
ARYSTAT_UNCONFIG	未構築状態
ARYSTAT_REBUILD	再構築中
ARYSTAT_REDUCE	縮退中
ARYSTAT_DEAD	ミラー状態破壊

・処理概要　ソフトミラーの状態を取得する。

・例 1

```
CPL_SmiIoctl     m_SmiIoc;
BOOL  ret;
int   Status;
// ソフトミラーの状態取得
ret = m_SmiIoc.GetSmiAryStatus( &Status );
```

・例 2

```
BOOL  ret;
int   Status;
// ソフトミラーの状態取得
ret = ::GetSmiAryStatus( &Status );
```

GetSmiDevStatus

・呼び出し形式　BOOL GetSmiDevStatus(int Id, int* pType ,int* pStatus)

・戻り値　TRUE:正常
FALSE:エラー

・引数

(I) int Id	デバイス ID	
	0 : Master HDD	
	1 : Slave HDD	
(I/O) int* pType	デバイスタイプ	
	ATADEVICE	ATA デバイス
	ATAPIDEVICE	CD-ROM
	UNKNOWNDEVICE	不明なデバイス
	NODEVICE	未接続
(I/O) int* pStatus	デバイスステータス	
	DEVSTAT_GOOD	正常
	DEVSTAT_NOTEXIST	未接続
	DEVSTAT_BROKEN	故障

・処理概要　ソフトミラーのデバイス状態を取得する。

・例 1

```
CPL_SmiIoctl     m_SmiIoc;
BOOL  ret;
int   Id, Type, Status
// デバイス状態の取得
Id = 0;
ret = m_SmiIoc.GetSmiDevStatus( Id, &Type, &Status );
```

・例 2

```
BOOL  ret;
int   Id, Type, Status
// デバイス状態の取得
Id = 0;
ret = ::GetSmiDevStatus( Id, &Type, &Status );
```

SetWdtResetMask

・呼び出し形式　BOOL SetWdtResetMask(int Mask)
・戻り値　TRUE:正常
FALSE:エラー
・引数　(I/O) int Mask　マスク情報
MASK_OFF　マスク解除
MASK_ON　マスク
・処理概要　WDT タイムアウト時の H/W リセットマスクの設定。
・例 1

```
CPL_Ioctl  m_Ioc;
BOOL  ret;
// WDTタイムアウト時のリセットをマスク解除する
ret = m_Ioc.SetWdtResetMask( MASK_OFF );
```

・例 2

```
BOOL  ret;
// WDTタイムアウト時のリセットをマスク解除する
ret = ::SetWdtResetMask( MASK_OFF );
```

GetWdtResetMask

・呼び出し形式　BOOL GetWdtResetMask(int* pMask)
・戻り値　TRUE:正常
FALSE:エラー
・引数　(I/O)　int* pMask　マスク情報へのポインタ
MASK_OFF　マスク解除
MASK_ON　マスク
・処理概要　WDT タイムアウト時の H/W リセットマスク情報の取得。
・例 1

```
CPL_Ioctl  m_Ioc;
BOOL  ret;
int   Mask;
// WDTタイムアウト時のリセットマスク情報取得
ret = m_Ioc.GetWdtResetMask( &Mask );
```

・例 2

```
BOOL  ret;
int   Mask;
// WDTタイムアウト時のリセットマスク情報取得
ret = m_Ioc.GetWdtResetMask( &Mask );
```

付 .4.7 Visual Basic 用関数一覧

関数名	説明
InitIoctl	CPL_Ioctlオブジェクト作成
EndIoctl	CPL_Ioctlオブジェクト破棄
GetDrvHandle	ドライバハンドル取得
CloseDrvHandle	GetDrvHandle取得ハンドル破棄
GetDrvVersion	ドライババージョン取得
GetMonitorSetup	モニタ許可／禁止設定取得
GetVoltParam	電圧監視用パラメータ取得
GetCurrentVolt	現在電圧値取得
GetFanParam	FAN監視用パラメータ取得
GetCurrentFan	現在FAN値取得
GetTempParam	温度監視用パラメータ取得
GetCurrentTemp	現在温度値取得
SetWdtCounter	ウォッチドックタイマカウンタ値設定
GetWdtCounter	ウォッチドックタイマカウンタ取得
SetWdtMask	ウォッチドックタイマタイムアウト時の警告マスク設定
GetWdtMask	ウォッチドッグタイマタイムアウト時の警告マスク取得
StartWdt	ウォッチドッグタイマ開始
StopWdt	ウォッチドッグタイマ停止
RestartWdt	ウォッチドッグタイマ再開
RunningWdt	ウォッチドッグタイマ動作状況取得
SetWarningOut	警告出力設定
GetWarningOut	警告出力取得
GetUniversalIn	汎用入力取得
ClearUniversalIn	汎用入力ラッチ状態解除
SetUniversalInMask	汎用入力マスク設定
GetUniversalInMask	汎用入力マスク取得
SetResetMask	リセットマスク設定
GetResetMask	リセットマスク取得
SetIdeErr	ミラーリングエラー（ソフト）設定
GetIdeErrHard	ミラーリングエラー（ハード）取得
GetEvent	エラーイベント取得
ClearEvent	エラーイベント消去
StartInsideBuzzer	内部Buzzer開始
StopInsideBuzzer	内部Buzzer停止
ChkInsideBuzzer	内部Buzzer状態チェック
GetWdtTimeout	ウォッチドッグタイマのタイムアウト状態取得
ClearWdtTimeout	ウォッチドッグタイマのタイムアウト状態クリア
SetWarningDOUT	警告出力DOUT設定
GetWarningDOUT	警告出力DOUT取得
GetSmiDrvHandle	SoftMirror ドライバハンドル取得
CloseSmiDrvHandle	SoftMirror ドライバハンドル破棄
GetSmiAryStatus	SoftMirror Array Status取得
GetSmiDevStatus	SoftMirror Device Status取得
SetWdtResetMask	ウォッチドッグタイマのリセットマスク設定
GetWdtResetMask	ウォッチドッグタイマのリセットマスク取得

付 .4.8 Visual Basic 用関数仕様詳細

IntIoctl

・呼び出し形式　Declare Sub InitIoctl Lib "PL_Ioc.dll" ()
・戻り値　なし
・引数　なし
・処理概要　CPL_Ioctl オブジェクトを作成する。作成されたオブジェクトは EndIoctl 関数が呼ばれるまで破棄されない。
・例　InitIoctl()

EndIoctl

・呼び出し形式　Declare Sub EndIoctl Lib "PL_Ioc.dll" ()
・戻り値　なし
・引数　なし
・処理概要　InitIoctl 関数で作成したオブジェクトを破棄する。
・例　EndIoctl()

GetDrvHandle

・呼び出し形式　Declare Function GetDrvHandle Lib "PL_Ioc.dll" (ByRef hndl As Long) As Long
・戻り値　0：正常
　1：エラー
・引数　hndl As Long　デバイスドライバハンドル(参照渡し)
・処理概要　デバイスドライバとのやり取りを行なうためのデバイスドライバハンドルを取得する。
・例

```
Dim ret As Long
Dim hndl As Long
ret = GetDrvHandle(hndl)
```

・システムモニタ/RASデバイスドライバが動作していない場合はエラーになります。

CloseDrvHandle

・呼び出し形式　Declare Function CloseDrvHandle Lib "PL_Ioc.dll"() As Long
・戻り値　0 以外:正常
　0:エラー
・引数　なし
・処理概要　GetDrvHandle 関数で取得したハンドルを破棄する。
・例

```
Dim ret As Long
// ハンドル破棄
ret = CloseDrvHandle()
```

GetDrvVersion

・呼び出し形式　Declare Function GetDrvVersion Lib "PL_Ioc.dll"
(ByRef Major As Long, ByRef Minor As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数
Major As Long　バージョン情報(Major,0 ～ 99) (参照渡し)
Minor As Long　バージョン情報(Minor,0 ～ 99) (参照渡し)

・処理概要　ドライババージョン情報を取得する。

・例
```
Dim ret As Long
Dim Major As Long
Dim Minor As Long
ret = GetDrvVersion(Major, Minor)
```

・ バージョンが1.10の場合は、
Major:1　(10 進数)
Minor:10　(10 進数)
となります。

GetMonitorSetup

・呼び出し形式　Declare Function GetMonitorSetup Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByRef Setup As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Selector As Long	取得パラメータ(値渡し)	
	MONITOR_VOLT_CPU	CPU コア電圧
	MONITOR_VOLT_P33	+3.3V 電圧
	MONITOR_VOLT_P50	+5.0V 電圧
	MONITOR_VOLT_P12	+12V 電圧
	MONITOR_VOLT_M12	-12V 電圧
	MONITOR_VOLT_M50	-5.0V 電圧
	MONITOR_VOLT_VIT	CPU コア電圧 2
	MONITOR_TEMP_SYSTEM	SYSTEM 温度
	MONITOR_TEMP_CPU	CPU 温度
	MONITOR_TEMP_OPT	OPTION 温度
	MONITOR_FAN_CPU	CPU FAN
	MONITOR_FAN_POWER	POWER FAN
	MONITOR_FAN_OPT	OPTION FAN
Setup As Long	取得データ(参照渡し)	
	0:Disable	
	1:Enable	

・処理概要　現在のモニタ許可／禁止状態を取得する。

・例
```
Dim ret As Long
Dim Setup As Long
// CPU コア電圧セットアップ状態取得
ret = GetMonitorSetup(MONITOR_VOLT_CPU, Setup )
```

GetVoltParam

・呼び出し形式　Declare Function GetVoltParam Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByRef ULimit As Long, ByRef LLimit As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Selector As Long	取得パラメータ(値渡し)	
	MONITOR_VOLT_CPU	CPU コア電圧
	MONITOR_VOLT_P33	+3.3V 電圧
	MONITOR_VOLT_P50	+5.0V 電圧
	MONITOR_VOLT_P12	+12V 電圧
	MONITOR_VOLT_M12	-12V 電圧
	MONITOR_VOLT_M50	-5.0V 電圧
	MONITOR_VOLT_VIT	CPU コア電圧 2
Ulimit As Long	電圧上限値(単位:mV) (参照渡し)	
LLimit As Long	電圧下限値(単位:mV) (参照渡し)	

・処理概要　電圧監視用パラメータを取得する。

・例

```
Dim ret As Long
Dim ULimit  As Long
Dim LLimit As Long
// CPU コア電圧上限下限値取得
ret = GetVoltParam( MONITOR_VOLT_CPU, ULimit, LLimit )
```

・関数から取得されたデータはmV(ミリボルト)単位になっています。V(ボルト)単位で使用する時は下記のような変換をする必要があります。
ボルト単位データ = ミリボルト単位データ /1000

GetCurrentVolt

・呼び出し形式　Declare Function GetCurrentVolt Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByRef Data As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Selector As Long	取得パラメータ(値渡し)	
	MONITOR_VOLT_CPU	CPU コア電圧
	MONITOR_VOLT_P33	+3.3V 電圧
	MONITOR_VOLT_P50	+5.0V 電圧
	MONITOR_VOLT_P12	+12V 電圧
	MONITOR_VOLT_M12	-12V 電圧
	MONITOR_VOLT_M50	-5.0V 電圧
	MONITOR_VOLT_VIT	CPU コア電圧 2
Data As Long	電圧値(単位:mV) (参照渡し)	

・処理概要　現在の電圧値を取得する。

・例
```
Dim ret As Long
Dim Data As Long
// CPU コア電圧値取得
ret = GetCurrentVolt(MONITOR_VOLT_CPU, Data)
```

・関数から取得されたデータはmV(ミリボルト)単位になっています。V(ボルト)単位で使用する時は下記のような変換をする必要があります。
ボルト単位データ = ミリボルト単位データ /1000

GetFanParam

・呼び出し形式　Declare Function GetFanParam Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByRef LLimit As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Selector As Long	取得パラメータ(値渡し)	
	MONITOR_FAN_CPU	CPU FAN
	MONITOR_FAN_POWER	POWER FAN
	MONITOR_FAN_OPT	OPTION FAN
LLimit As Long	FAN 下限回転数(単位:RPM) (参照渡し)	
	(RPM:1 分あたりの回転数)	

・処理概要　FAN 監視用のパラメータを取得する。

・例
```
Dim ret As Long
Dim LLimit As Long
// CPU FAN 下限回転数取得
ret = GetFanParam( MONITOR_FAN_CPU, LLimit )
```

GetCurrentFan

・呼び出し形式　Declare Function GetCurrentFan Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByRef Data As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Selector As Long	取得パラメータ(値渡し)	
	MONITOR_FAN_CPU	CPU FAN
	MONITOR_FAN_POWER	POWER FAN
	MONITOR_FAN_OPT	OPTION FAN
Data As Long	FAN 回転数(単位:RPM) (参照渡し)	
	(RPM:1 分あたりの回転数)	

・処理概要　現在の FAN 回転数を取得する。

・例

```
Dim ret As Long
Dim Data As Long
// CPU FAN 回転数取得
ret = GetCurrentFan( MONITOR_FAN_CPU, Data )
```

GetTempParam

・呼び出し形式　Declare Function GetTempParam Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByRef ULimit As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Selector As Long	取得パラメータ(値渡し)	
	MONITOR_TEMP_SYSTEM	SYSTEM 温度
	MONITOR_TEMP_CPU	CPU 温度
	MONITOR_TEMP_OPT	OPTION 温度
Ulimit As Long	温度上限値(単位:℃) (参照渡し)	

・処理概要　温度監視用のパラメータを取得する。

・例

```
Dim ret As Long
Dim ULimit As Long
// SYSTEM 温度上限値取得
ret = GetTempParam( MONITOR_TEMP_SYSTEM, ULimit )
```

GetCurrentTemp

・呼び出し形式　Declare Function GetCurrentTemp Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByRef Data As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Selector As Long	取得パラメータ(値渡し)	
	MONITOR_TEMP_SYSTEM	SYSTEM 温度
	MONITOR_TEMP_CPU	CPU 温度
	MONITOR_TEMP_OPT	OPTION 温度
Data As Long	温度値(単位:℃) (参照渡し)	

・処理概要　現在の温度値を取得する。

・例
```
Dim ret As Long
Dim Data As Long
// SYSTEM 温度値取得
ret = GetCurrentTemp( MONITOR_TEMP_SYSTEM, Data )
```

SetWdtCounter

・呼び出し形式　Declare Function SetWdtCounter Lib "PL_Ioc.dll"
(ByVal Counter As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Counter As Long	ウォッチドッグタイマ初期カウンタ値(値渡し) (5 ～ 255) (単位:秒)

・処理概要　ウォッチドッグタイマの初期カウンタ値を設定する。

・例
```
Dim ret As Long
// ウォッチドッグタイマ初期カウンタ値を 10 秒に設定
ret = SetWdtCounter( 10 )
```

GetWdtCounter

・呼び出し形式　Declare Function GetWdtCounter Lib "PL_Ioc.dll"
(ByRef Counter As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Counter As Long	ウォッチドッグタイマの初期カウンター値 (参照渡し)(単位:秒)

・処理概要　現在のウォッチドッグタイマの初期カウンタ値を取得する。

・例
```
Dim ret As Long
Dim Counter As Long
ret = GetWdtCounter(Counter)
```

SetWdtMask

・呼び出し形式　Declare Function SetWdtMask Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByVal Mask As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Selector As Long	設定項目(値渡し)	
	WARNING_LAMP	LAMP
	WARNING_ALARM	ALARM
Mask As Long	マスク情報(値渡し)	
	MASK_OFF	マスク解除
	MASK_ON	マスク

・処理概要　ウォッチドッグタイマタイムアウト時に出力する警告のマスクを設定する。

・例
```
Dim ret As Long
// LAMP 出力をマスクする
ret = SetWdtMask( WARNING_LAMP, MASK_ON )
//ALARM 出力のマスクを解除する。
ret = SetWdtMask( WARNING_ALARM, MASK_OFF )
```

GetWdtMask

・呼び出し形式　Declare Function GetWdtMask Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByRef Mask As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Selector As Long	設定項目(値渡し)	
	WARNING_LAMP	LAMP
	WARNING_ALARM	ALARM
Mask As Long	マスク情報(参照渡し)	
	MASK_OFF	マスク解除
	MASK_ON	マスク

・処理概要　ウォッチドッグタイマタイムアウト時の警告出力マスク情報を取得する。

・例
```
Dim ret As Long
Dim Mask As Long
// LAMP のマスク情報取得
ret = GetWdtMask( WARNING_LAMP, Mask )
// ALARM のマスク情報取得
ret = GetWdtMask( WARNING_ALARM, Mask )
```

StartWdt

・呼び出し形式　Declare Function StartWdt Lib "PL_Ioc.dll" () As Long
・戻り値　0 以外:正常
0:エラー
・引数　なし
・処理概要　ウォッチドッグタイマのカウントダウンを開始する。
・例
```
Dim ret As Long
ret = StartWdt()
```

StopWdt

・呼び出し形式　Declare Function StopWdt Lib "PL_Ioc.dll" () As Long
・戻り値　0 以外:正常
0:エラー
・引数　なし
・処理概要　ウォッチドッグタイマのカウントダウンを停止する。
・例
```
Dim ret As Long
ret = StopWdt()
```

RestartWdt

・呼び出し形式　Declare Function RestartWdt Lib "PL_Ioc.dll" () As Long
・戻り値　0 以外:正常
0:エラー
・引数　なし
・処理概要　ウォッチドッグタイマのカウンタ値を初期値に戻し、再カウントダウンを始める。
・例
```
Dim ret As Long
ret = RestartWdt()
```

MEMO
・ ウォッチドッグタイマが停止状態の場合は、何も処理しません。

RunningWdt

・呼び出し形式
```
Declare Function RunningWdt Lib "PL_Ioc.dll"
(ByRef RunFlag As Long) As Long
```
・戻り値　0 以外:正常
0:エラー
・引数　RunFlag As Long　ウォッチドッグタイマの動作状態(参照渡し)

WATCHDOG_STOP	停止中
WATCHDOG_COUNTDOWN	カウントダウン中

・処理概要　ウォッチドッグタイマの動作状態を取得する。
・例
```
Dim ret As Long
Dim RunFlag As Long
ret = RunningWdt( RunFlag )
```

SetWarningOut

・呼び出し形式　Declare Function SetWarningOut Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByVal WarnOut As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Selector As Long	設定項目(値渡し)	
	WARNING_LAMP	LAMP
	WARNING_ALARM	ALARM
WarnOut As Long	出力状態(値渡し)	
	OUTPUT_OFF	出力 OFF
	OUTPUT_ON	出力 ON

・処理概要　設定項目(LAMP、ALARM)の警告情報を設定する。

・例
```
Dim ret As Long
// LAMP の出力状態を ON に設定
ret = SetWarningOut( WARNING_LAMP, OUTPUT_ON )
// ALARM の出力状態を OFF に設定
ret = SetWarningOut( WARNING_ALARM, OUTPUT_OFF )
```

GetWarningOut

・呼び出し形式　Declare Function GetWarningOut Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByRef WarnOut As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Selector As Long	設定項目(値渡し)	
	WARNING_LAMP	LAMP
	WARNING_ALARM	ALARM
WarnOut As Long	出力状態(参照渡し)	
	OUTPUT_OFF	出力 OFF
	OUTPUT_ON	出力 ON

・処理概要　現在の設定項目(LAMP,ALARM)の警告状態を取得する。

・例
```
Dim ret As Long
Dim WarnOut As Long
// LAMP の出力状態取得
ret = GetWarningOut( WARNING_LAMP, WarnOut )
// ALARM の出力状態取得
ret = GetWarningOut( WARNING_ALARM, WarnOut )
```

GetUniversalIn

・呼び出し形式　Declare Function GetUniversalIn Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByRef UniIn As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Selector As Long	対象ポート(値渡し)	
	PORT_UNI0	Universal Input 0
	PORT_UNI1	Universal Input 1
UniIn As Long	入力状態(参照渡し)	
	INPUT_OFF	入力なし
	INPUT_ON	入力あり

・処理概要　対象ポート(Universal Input 0, Universal Input 1)の入力状態を取得する。

・例

```
Dim ret As Long
Dim UniIn As Long
// Universal Input 0の入力状態取得
ret = GetUniversalIn( PORT_UNI0, UniIn )
// Universal Input 1の入力状態取得
ret = GetUniversalIn( PORT_UNI1, UniIn )
```

ClearUniversalIn

・呼び出し形式　Declare Function ClearUniversalIn Lib "PL_Ioc.dll"
(ByVal Selector As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Selector As Long	対象ポート(値渡し)	
	PORT_UNI0	Universal Input 0
	PORT_UNI1	Universal Input 1

・処理概要　対象ポート(Universal Input 0, Universal Input 1)の入力状態をキャンセルする。

・例

```
Dim ret As Long
// Universal Input 0の入力状態をキャンセルする
ret = ClearUniversalIn( PORT_UNI0 )
// Universal Input 1の入力状態をキャンセルする
ret = ClearUniversalIn( PORT_UNI1 )
```

SetUniversalInMask

・呼び出し形式　Declare Function SetUniversalInMask Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByVal Mask As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Selector As Long	対象ポート(値渡し)	
	PORT_UNI0	Universal Input 0
	PORT_UNI1	Universal Input 1
Mask As Long	マスク情報(値渡し)	
	MASK_OFF	マスク解除
	MASK_ON	マスク

・処理概要　対象ポート(Universal Input 0, Universal Input 1)のマスク情報を設定する。

・例

```
Dim ret As Long
// Universal Input 0 をマスク解除
ret = SetUniversalInMask( PORT_UNI0, MASK_OFF )
// Universal Input 1 をマスク
ret = SetUniversalInMask( PORT_UNI1, MASK_ON )
```

GetUniversalInMask

・呼び出し形式　Declare Function GetUniversalInMask Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByRef Mask As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Selector As Long	対象ポート(値渡し)	
	PORT_UNI0	Universal Input 0
	PORT_UNI1	Universal Input 1
Mask As Long	マスク情報(参照渡し)	
	MASK_OFF	マスク解除
	MASK_ON	マスク

・処理概要　対象ポート(Universal Input 0, Universal Input 1)のマスク情報を取得する。

・例

```
Dim ret As Long
Dim Mask As Long
// Universal Input0 マスク情報取得
ret = GetUniversalInMask( PORT_UNI0, Mask )
// Universal Input1 マスク情報取得
ret = GetUniversalInMask( PORT_UNI1, Mask )
```

SetResetMask

・呼び出し形式　Declare Function SetResetMask Lib "PL_Ioc.dll"
(ByVal Mask As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数　Mask As Long　マスク情報(値渡し)
MASK_OFF　マスク解除
MASK_ON　マスク

・処理概要　リセットマスクを設定する。

・例
```
Dim ret As Long
// リセットマスク解除
ret = SetResetMask( MASK_OFF )
```

GetResetMask

・呼び出し形式　Declare Function GetResetMask Lib "PL_Ioc.dll"
(ByRef Mask As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数　Mask As Long　マスク情報(参照渡し)
MASK_OFF　マスク解除
MASK_ON　マスク

・処理概要　現在のリセットマスク情報を取得する。

・例
```
Dim ret As Long
Dim Mask As Long
ret = GetResetMask( Mask )
```

SetIdeErr

・呼び戻し形式　Declare Function SetIdeErr Lib "PL_Ioc.dll"
(ByVal IdeErr As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数　IdeErr As Long　エラー出力情報(値渡し)
IDE_ERROR_OFF　エラー出力しない
IDE_ERROR_ON　エラー出力する

・処理概要　ソフトウェア制御での IDE エラー出力を設定する。

・例
```
Dim ret As Long
// IDEエラー出力しないように設定
ret = SetIdeErr( IDE_ERROR_OFF )
```

GetIdeErrHard

・呼び出し形式　Declare Function GetIdeErrHard Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByRef IdeErr As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Selector As Long	取得パラメータ(値渡し)	
	IDE_ERROR_1	IDE_ERR1
	IDE_ERROR_2	IDE_ERR2
IdeErr As Long	エラー信号(参照渡し)	
	IDE_ERROR_OFF	正常
	IDE_ERROR_ON	エラー

・処理概要　現在のハードウェアの出力する IDE エラー信号を取得する。

・例

```
Dim ret As Long
Dim IdeErr As Long
// IDE_ERR1 の信号取得
ret = GetIdeErrHard( IDE_ERROR_1, IdeErr )
```

GetLightblowErr

・呼び戻し形式　Declare Function GetLightblowErr Lib "PL_Ioc.dll"
(ByRef LightblowErr As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

LightblowErr As Long	エラー情報(参照渡し)	
	BACKLIGHT_OK	正常
	BACKLIGHT_ERR	エラー

・処理概要　現在のバックライトエラー情報を取得する。

・例

```
Dim ret As Long
Dim LightblowErr As Long
// バックライトエラー情報取得
ret = GetLightblowErr( LightblowErr )
```

・ PL-6920 シリーズでのみ使用できます。

GetEvent

・呼び出し形式　Declare Function GetEvent Lib "PL_Ioc.dll"
(ByVal Selector As Long, ByRef Event As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Selector As Long	取得パラメータ(値渡し)	
	EVENT_VOLT_CPU	CPU コア電圧
	EVENT_VOLT_P33	+3.3V
	EVENT_VOLT_P50	+5.0V
	EVENT_VOLT_P12	+12V
	EVENT_VOLT_M12	-12V
	EVENT_VOLT_M50	-5.0V
	EVENT_VOLT_VIT	CPU コア電圧 2
	EVENT_FAN_CPU	CPU FAN
	EVENT_FAN_POWER	POWER FAN
	EVENT_FAN_OPT	OPTION FAN
	EVENT_TEMP_SYSTEM	SYSTEM 温度
	EVENT_TEMP_CPU_OPT	CPU or OPTION温度
	EVENT_UNI_IN0	Universal Input 0
	EVENT_UNI_IN1	Universal Input 1
	EVENT_WDT_TIMEOUT	Watchdog Timeout
Event As Long	エラーイベント情報(参照渡し)	
	ERROR_EVENT_OFF	エラーイベントなし
	ERROR_EVENT_ON	エラーイベントあり

・処理概要　マシンの電圧，FAN，温度の異常、また、Universal Input 動作の情報（イベント)、WatchDog Timeout 情報をチェックする。

・例

```
Dim ret As Long
Dim Event As Long
// CPU コア電圧のエラーイベント情報取得
ret = GetEvent( EVENT_VOLT_CPU, Event )
```

ClearEvent

・呼び出し形式　Declare Function ClearEvent Lib "PL_Ioc.dll"
(ByVal Selector As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数　Selector As Long　エラーイベントキャンセル対象パラメータ(値渡し)

EVENT_VOLT_CPU	CPU コア電圧
EVENT_VOLT_P33	+3.3V
EVENT_VOLT_P50	+5.0V
EVENT_VOLT_P12	+12V
EVENT_VOLT_M12	-12V
EVENT_VOLT_M50	-5.0V
EVENT_VOLT_VIT	CPU コア電圧 2
EVENT_FAN_CPU	CPU FAN
EVENT_FAN_POWER	POWER FAN
EVENT_FAN_OPT	OPTION FAN
EVENT_TEMP_SYSTEM	SYSTEM 温度
EVENT_TEMP_CPU_OPT	CPU or OPTION温度
EVENT_UNI_IN0	Universal Input0
EVENT_UNI_IN1	Universal Input1
EVENT_WDT_TIMEOUT	Watchdog Timeout

・処理概要　エラーイベントをキャンセルする。

・例
```
Dim ret As Long
// CPU コア電圧エラーイベントキャンセル
ret = ClearEvent( EVENT_VOLT_CPU )
```

StartInsideBuzzer

・呼び出し形式　Declare Function StartInsideBuzzer Lib "PL_Ioc.dll"
(ByVal hz As Long, ByVal ms As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数
hz As Long　ブザー周波数(値渡し)
ms As Long　ブザー音長(値渡し)

・処理概要　指定された周波数、音長で内部ブザーを開始する。

・例 1
```
Dim ret As Long
Dim hz As Long
Dim ms As Long
// 600Hz で 1 秒間ブザーを鳴らす。
hz = 600
ms = 1000
ret = StartInsideBuzzer( hz, ms )
```

・ Windows®95、Windows®98 の関数です。WindowsNT®4.0、Windows®2000、Windows®XP で使用してもエラーとなります。

StopInsideBuzzer

・呼び戻し形式　Declare Function StopInsideBuzzer Lib "PL_Ioc.dll" () As Long
・戻り値　0 以外:正常
0:エラー
・引数　なし
・処理概要　内部 Buzzer を停止する。
・例
```
Dim ret As Long
// 内部 Buzzer を停止する
ret = StopInsideBuzzer()
```

・ Windows®95、Windows®98 の関数です。WindowsNT®4.0、Windows®2000、Windows®XP で使用してもエラーとなります。

ChkInsideBuzzer

・呼び戻し形式　Declare Function ChkInsideBuzzer Lib "PL_Ioc.dll" (ByRef buff As Long) As Long
・戻り値　0 以外:正常
0:エラー
・引数　BuzzerParam As Long　ブザー状態(参照渡し)
BUZZER_ON　ブザー開始中
BUZZER_OFF　ブザー停止中
・処理概要　内部ブザーの開始 / 停止状態をチェックする。
・例
```
Dim ret As Long
Dim BuzaerParam As Long
// ブザー状態チェック
ret = ChkInsideBuzzer( BuzzerParam )
```

・ Windows®95、Windows®98 の関数です。WindowsNT®4.0、Windows®2000、Windows®XP で使用してもエラーとなります。

GetWdtTimeout

・呼び出し形式　Declare Function GetWdtTimeout Lib "PL_Ioc.dll"
(ByRef Timebuf As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数　Timebuf As Long　ウォッチドッグタイムアウト状態(参照渡し)

・処理概要　ウォッチドッグのタイムアウト状態を取得する。

・例
```
Dim ret As Long
Dim Timebuf As Long
// ウォッチドッグのタイムアウト状態取得
ret = GetWdtTimeout( Timebuf )
```

ClearWdtTimeout

・呼び出し形式　Declare Function ClearWdtTimeout Lib "PL_Ioc.dll () As Long

・戻り値　0 以外:正常
0:エラー

・引数　なし

・処理概要　ウォッチドッグのタイムアウト状態をクリアする。

・例
```
 Dim ret As Long
// ウォッチドッグのタイムアウト状態クリア
ret = ClearWdtTimeout()
```

SetWarningDOUT

・呼び出し形式　Declare Function SetWarningDOUT Lib "PL_Ioc.dll"
(ByVal WarningOut As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数　WarningOut As Long　出力状態(値渡し)
OUTPUT_OFF　出力 OFF
OUTPUT_ON　出力 ON

・処理概要　現在の設定項目(DOUT)の警告状態を設定する。

・例
```
Dim ret As Long
// DOUT の出力状態を OFF に設定
ret = SetWarningDOUT( OUTPUT_OFF )
```

GetWarningDOUT

・呼び出し形式　Declare Function GetWarningDOUT Lib "PL_Ioc.dll"
(ByRef WarningOut As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数　WarningOut As Long　出力状態(参照渡し)
OUTPUT_OFF　出力 OFF
OUTPUT_ON　出力 ON

・処理概要　現在の設定項目(DOUT)の警告状態を取得する。

・例 1
```
Dim ret As Long
Dim WarningOut As Long
ret = GetWarningDOUT( WarningOut )
```

GetSmiDrvHandle

・呼び出し形式　Declare Function GetSmiDrvHandle Lib "PL_Ioc.dll" () As Long

・戻り値　0:正常
1:エラー

・引数　なし

・処理概要　ソフトミラーデバイスドライバとのやり取りを行なうためのデバイスドライバハンドルを取得する。

・例 1
```
Dim ret As Long
//
ret = GetSmiDrvHandle()
```

MEMO

・ ソフトミラーデバイスドライバが動作していない場合はエラーになります。

CloseSmiDrvHandle

・呼び出し形式　Declare Function CloseSmiDrvHandle Lib "PL_Ioc.dll" () As Long

・戻り値　0 以外:正常
0:エラー

・引数　なし

・処理概要　GetSmiDrvHandle 関数で取得したハンドルを破棄する。

・例
```
Dim ret As Long
// ハンドルを破棄する
ret = CloseSmiDrvHandle()
```

GetSmiAryStatus

・呼び出し形式　Declare Function GetSmiAryStatus Lib "PL_Ioc.dll"
(ByRef Status As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Status As Long	ソフトミラーステータス(参照渡し)	
	ARYSTAT_GOOD	正常
	ARYSTAT_NOTEXIST	出力されません
	ARYSTAT_UNCONFIG	未構築状態
	ARYSTAT_REBUILD	再構築中
	ARYSTAT_REDUCE	縮退中
	ARYSTAT_DEAD	ミラー状態破壊

・処理概要　ソフトミラーの状態を取得する。

・例

```
Dim ret As Long
Dim Status As Long
// ソフトミラーの状態を取得する
ret = GetSmiAryStatus( Status )
```

GetSmiDevStatus

・呼び出し形式　Declare Function GetSmiDevStatus Lib "PL_Ioc.dll"
(ByVal Id As Long, ByRef Type As Long, ByRef Status As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数

Id As Long	デバイス ID(値渡し)	
	0 : Master HDD	
	1 : Slave HDD	
Type As Long	デバイスタイプ(参照渡し)	
	ATADEVICE	ATA デバイス
	ATAPIDEVICE	CD-ROM
	UNKNOWNDEVICE	不明なデバイス
	NODEVICE	未接続
Status As Long	デバイスステータス(参照渡し)	
	DEVSTAT_GOOD	正常
	DEVSTAT_NOTEXIST	未接続
	DEVSTAT_BROKEN	故障

・処理概要　ソフトミラーのデバイス状態を取得する。

・例

```
Dim ret As Long
Dim Id As Long
Dim Type As Long
Dim Status As Long
// ソフトミラーのデバイス状態を取得する。
Id = 0
ret = GetSmiDevStatus( Id, Type, Status )
```

SetWdtResetMask

・呼び出し形式　Declare Function SetWdtResetMask Lib "PL_Ioc.dll"
(ByVal Mask As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数　Mask As Long　マスク情報(値渡し)
MASK_OFF　マスク解除
MASK_ON　マスク

・処理概要　WDT タイムアウト時の H/W リセットマスクの設定。

・例
```
Dim ret As Long
// WDT タイムアウト時のリセットをマスク解除する
ret = SetWdtResetMask( MASK_OFF )
```

GetWdtResetMask

・呼び出し形式　Declare Function GetWdtResetMask Lib "PL_Ioc.dll"
(ByRef Mask As Long) As Long

・戻り値　0 以外:正常
0:エラー

・引数　Mask As Long　マスク情報(参照渡し)
MASK_OFF　マスク解除
MASK_ON　マスク

・処理概要　WDT タイムアウト時の H/W リセットマスク情報の取得。

・例
```
Dim ret As Long
Dim Mask As Long
// WDT タイムアウト時の H/W リセットマスク情報の取得
ret = GetWdtResetMask( Mask )
```

付 .5 バックライトコントロール API-DLL

付 .5.1 動作環境

バックライトコントロール機能をPL-X920シリーズ上で動作させるためのダイナミックリンクライブラリ(API-DLL)について説明します。

API-DLLは、アプリケーケーションからバックライトコントロール機能を「バックライトコントロールデバイスドライバ」経由でアクセスするためのインターフェースを提供します。アプリケーションは、このDLLを経由し、以下の機能を使用することが可能になります。

・バックライトコントロールON/OFF機能

■オペレーティングシステム

CD-ROMに付属のAPI-DLLが動作するOSは以下のとおりです。

・Microsoft Windows®95

・Microsoft Windows®98

・Microsoft WindowsNT®4.0

・Microsoft Windows®2000

・Microsoft Windows®XP

また、それぞれのOS用の「バックライトコントロールデバイス」が動作していなければなりません。

■対応言語

・Microsoft Visual C

・Micorsoft Visual C++

・Microsoft Visual Basic

◆必要ファイル

このDLLを使用すためには、各開発言語毎に以下のファイルが必要です。

・Visual C

ファイル名	説明
PL_BLIocif.h	ドライバインタフェイス定義ファイル
PL_BLIoc.LIB	ライブラリ定義ファイル
PL_BLIoc.dll	ダイナミックリンクライブラリファイル

・Visual C++

ファイル名	説明
PL_BLIocif.h	ドライバインタフェイス定義インクルードファイル
PL_BLIocall.h	CPL_BLIocallクラス定義インクルードファイル
PL_BLIoctl.h	CPL_BLIoctlクラス定義インクルードファイル
PL_BLIoc.LIB	ライブラリ定義ファイル
PL_BLIoc.dll	ダイナミックリンクライブラリファイル

*インクルードするヘッダファイルの順番は以下の通りです。

#include PL_BLIocif.h

#include PL_BLIoctl.h

PL_BLIocall.hは自動でインクルードされるので、直接インクルードしないでください。

・Visual Basic

ファイル名	説明
PL_BLIoc.bas	ドライバインタフェイス定義インクルードファイル
PL_BLIoc.LIb	ライブラリ定義ファイル
PL_BLIoc.dll	ダイナミックリンクライブラリファイル

■ Dynamic Link Library(DLL)

作成したアプリケーションからPL_BLIoc.dllを使用するために、以下の位置にDLLを格納する必要があります。

OS	位置
Windows 95/Windows 98	C:\Windows\System
WindowsNT4.0/Windows2000	C:\Winnt\System32
Windows XP	C:\Windows\System32

付 .5.2 クラス内容

■ CPL_BLIoctl クラス

CPL_BLIoctlクラスはCPL_BLIoctlクラスでデバイスドライバアクセスするためのパラメータをセットします。

キーワード	型	変数名	説明
public	HANDLE	m_Drvhandle	デバイスドライバハンドル

■ CPL_BLIocal クラス

CPL_BLIoctlでセットされたパラメータを使用し、DeviceloControl(ドライバアクセス関数)を呼び出します。

ただし、このクラスはCPL_BLIoctlから継承されているので直接使用することはありません。

キーワード	型	変数名	説明
public	HANDLE	m_h	デバイスドライバハンドル
public	LONG	m_long	実行する操作の制御コード
public	void *	m_ibp	入力データバッファアドレス
public	ULONG	m_ibsize	入力データバッファサイズ
public	void *	m_obp	出力データバッファアドレス
public	ULONG	m_obsize	出力データバッファサイズ
public	DWORD	m_retsize	実際の出力バイト数のアドレス
public	LPOVERLAPPED	m_ovlp	オーバーラップ構造体のアドレス

付 .5.3 Visual C 用関数仕様一覧

関数名	説明
InitBLIoctl	CPL_BLIoctlオブジェクト作成
EndBLIoctl	CPL_BLIoctlオブジェクト破棄
GetBLDrvHandle	ドライバハンドル取得
GetBLDrvVersion	ドライババージョン取得
SetBLControl	バックライトコントロール設定
GetBLControl	バックライトコントロール状態取得

付 .5.4 Visual C 用関数仕様詳細

InitBLIoctl

・呼び出し形式　　void WINAPI InitBLIoctl(void)
・戻り値　　なし
・引数　　なし
・処理概要　　CPL_BLIoctl オブジェクトを作成する。作成されたオブジェクトは EndBLIoctl 関数が呼ばれるまで破棄されない。
・例　　InitBLIoctl();

EndBLIoctl

・呼び出し形式　　void WINAPI EndBLIoctl(void)
・戻り値　　なし
・引数　　なし
・処理概要　　InitBLIoctl 関数で作成したオブジェクトを破棄する。
・例　　EndBLIoctl();

GetBLDrvHandle

・呼び出し形式　　int WINAPI GetBLDrvHandle(HANDLE * pHndl)
・戻り値　　0:正常
　　1:エラー
・引数　　(I/O) HANDL * pHndl　　デバイスドライバハンドルへのポインタ
・処理概要　　デバイスドライバとのやり取りを行うためのデバイスドライバハンドルを取得する。
・例

```
int   ret;
HANDLE     hndl;

ret = GetBLDrvHandle( &hndl );
```

・ バックライトコントロールデバイスドライバが動作していない場合はエラーになります。

GetBLDrvVersion

・呼び出し形式　BOOL WINAPI GetBLDrvVersion(int *pMajor, int *pMinor)
・戻り値　TRUE:正常
FALSE:エラー
・引数　(I/O) int *pMajor　バージョン情報(Major,0～99)へのポインタ
(I/O) int *pMinor　バージョン情報(Minor,0～99)へのポインタ
・処理概要　ドライババージョン情報を取得する。
・例
```
BOOL  ret;
int   Major, Minor;
ret = GetBLDrvVersion( &Major, &Minor );
```

・ バージョンが 1.10 の場合は
Major:1　(10 進数)
Minor:10　(10 進数)
となります。

SetBLControl

・呼び出し形式　BOOL WINAPI SetBLControl(int BLFlag)
・戻り値　TRUE:正常
FALSE:エラー
・引数　(I) int BLFlag　設定パラメータ
BACKLIGHT_OFF バックライト OFF
BACKLIGHT_ON バックライト ON
・処理概要　バックライトの ON/OFF を設定する。
・例
```
BOOL ret;
//バックライトコントロール ON 設定
ret = SetBLControl( BACKLIGHT_ON );
```

GetBLControl

・呼び出し形式　BOOL WINAPI GetBLControl(int pBLFlag)
・戻り値　TRUE:正常
FALSE:エラー
・引数　(I/0) int *pBLFlag　バックライト状態へのポインタ
BACKLIGHT_OFF　バックライト OFF
BACKLIGHT_ON　バックライト ON
・処理概要　バックライトコントロール状態を取得する。
・例
```
BOOL  ret;
int   BLFlag;
//バックライトコントロール状態取得
ret = GetBLControl( &BLFlag );
```

付 .5.5 Visual C++ 用関数一覧

関数名	説明
GetBLDrvHandle	ドライバハンドル取得
GetBLDrvVersion	ドライババージョン取得
SetBLControl	バックライトコントロール設定
GetBLControl	バックライトコントロール状態取得

付 .5.6 Visual C++ 用関数仕様詳細

GetBLDrvHandle

・呼び出し形式　int GetBLDrvHandle(void) または int GetBLDrvHandle(HANDLE *PHndl)

・戻り値　0:正常
1:エラー

・引数　なし

・処理概要　デバイスドライバとのやり取りを行うためのデバイスドライバハンドルを取得する。取得されたハンドルはメンバ変数m_handleに格納される。

・例 1
```
CPL_BLloctl      m_BLoc;
m_BLloc. GetBLDrvHandle()
```

・例 2
```
int         ret;
HANDLE      hndl;
ret = ::GetBLDrvHandle( &hndl );
```

MEMO
・ バックライトコントロールデバイスドライバが動作していない場合はエラーになります。

GetBLDrvVersion

・呼び出し形式　BOOL GetBLDrvVersion(int *pMajor, int *pManor)

・戻り値　TRUE:正常
FALSE:エラー

・引数　(I/O) int *pMajor　バージョン情報(Major,0～99)へのポインタ
(I/O) int *pMinor　バージョン情報(Minor,0～99)へのポインタ

・処理概要　ドライババージョン情報を取得する。

・例 1
```
CPL_BLloctl      m_BLloc;
BOOL  ret;
int Major, Minor;
ret = m_BLloc.GetBLDrvHndle( &Major, &Minor );
```

・例 2
```
BOOL  ret;
int   Major, Minor;
ret = GetBLDrvVersion( &Major, &Minor );
```

MEMO
・ バージョンが 1.10 の場合は
Major:1　(10 進数)
Minor:10　(10 進数)
となります。

SetBLControl

・呼び出し形式　BOOL SetBLControl(int BLFlag)
・戻り値　TRUE:正常
FALSE:エラー
・引数　(I)　int BLFlag　設定パラメータ
BACKLIGHT_OFF バックライト OFF
BACKLIGHT_ON　バックライト ON
・処理概要　バックライトの ON/OFF を設定する。
・例 1
```
CPL_BLloc m_BLloc
BOOL       ret;
//バックライトコントロール ON 設定
ret = m_BLloc.SetBLControl( BACKLIGHT_ON );
```
・例 2
```
BOOL ret;
//バックライトコントロール ON 設定
ret = ::SetBLControl( BACKLIGHT_ON );
```

GetBLControl

・呼び出し形式　BOOL GetBLControl(int *pBLFlag)
・戻り値　TRUE:正常
FALSE:エラー
・引数　(I/O) int *pBLFlag　バックライト状態へのポインタ
BACKLIGHT_OFF　バックライト OFF
BACKLIGHT_ON　バックライト ON
・処理概要　バックライトコントロール状態を取得する。
・例 1
```
CPL_BLloc m_BLloc;
BOOL  ret;
int   BLFlag;
//バックライトコントロール状態取得
ret = m_BLloc. GetBLControl( &BLFlag );
```
・例 2
```
BOOL  ret;
int   BLFlag;
//バックライトコントロール状態取得
ret = ::GetBLControl( &BLFlag );
```

付 .5.7 Visual Basic 用関数一覧

関数名	説明
InitBLIoctl	CPL_BLIoctlオブジェクト作成
EndBLIoctl	CPL_BLIoctlオブジェクト破棄
GetBLDrvHandle	ドライバハンドル取得
GetBLDrvVersion	ドライババージョン取得
SetBLControl	バックライトコントロール設定
GetBLControl	バックライトコントロール状態取得

付 .5.8 Visual Basic 用関数仕様詳細

InitBLIoctl

・呼び出し形式　Declare Sub InitBLIoctl Lib "PL_BLIoc.dll" ()
・戻り値　なし
・引数　なし
・処理概要　CPL_BLIoctl オブジェクトを作成する。作成されたオブジェクトは EndBLIoctl 関数が呼ばれるまで破棄されない。
・例　Call InitBLIoctl

EndBLIoctl

・呼び出し形式　Declare Sub EndBLIoctl Lib "PL_BLIoc.dll" ()
・戻り値　なし
・引数　なし
・処理概要　InitBLIoctl 関数で作成したオブジェクトを破棄する。
・例　Call EndBLIoctl

GetBLDrvHandle

・呼び出し形式　Declare Function GetBLDrvHandle Lib "PL_BLIoc.dll"
(ByRef hndl As Long) As Long
・戻り値　0:正常
1:エラー
・引数　hndl As Long　デバイスドライバハンドル(参照渡し)
・処理概要　デバイスドライバとのやり取りを行なうためのデバイスドライバハンドルを取得する。
・例

```
Dim ret  As Long
Dim hndl As Long
ret = GetBLDrvHandle( hndl )
```

MEMO
・ バックライトコントロールデバイスドライバが動作していない場合はエラーになります。

GetBLDrvVersion

・呼び出し形式　Declare Function GetBLDrvVersion Lib "PL_BLIoc.dll"
(ByRef Major As Long, ByRef Minor As Long) As Long
・戻り値　0 以外: 正常
0:エラー
・引数　Major As Long　バージョン情報(Major,0 ～ 99)　(参照渡し)
Minor As Long　バージョン情報(Minor,0 ～ 99)　(参照渡し)
・ 処理概要　ドライババージョン情報を取得する。
・例

```
Dim ret As Long
Dim Major As Long
Dim Minor As Long
ret = GetBLDrvVersion(Major, Minor)
```

・ バージョンが1.10の場合は、
Major:1　(10 進数)
Minor:10　(10 進数)
となります。

SetBLControl

・呼び出し形式　Declare Function SetBLControl Lib "PL_BLIoc.dll"
(ByVal BLFlag As Long) As Long
・戻り値　0 以外: 正常
0:エラー
・引数　BLFlag As Long　設定パラメータ(値渡し)
BACKLIGHT_OFF　バックライト OFF
BACKLIGHT_ON　バックライト ON
・処理概要　バックライトの ON ／ OFF を設定する。
・例

```
Dim ret As Long
// バックライトコントロール ON 設定
ret = SetBLControl( BACKLIGHT_ON )
```

GetBLControl

・呼び出し形式　Declare Function GetBLControl Lib "PL_BLIoc.dll"
(ByRef BLFlag As Long) As Long
・戻り値　0 以外: 正常
0:エラー
・引数　BLFlag As Long　バックライト状態(参照渡し)
BACKLIGHT_OFF　バックライト OFF
BACKLIGHT_ON　バックライト ON
・処理概要　バックライトコントロール状態を取得する。
・例

```
Dim ret As Long
Dim BLFlag As Long
// バックライトコントロール状態取得
ret = GetBLControl( BLFlag )
```

付 .6 使用許諾書

■ IN-fINITY soft 製　Keyclick32

著作権は、IN-fINITY soft が所有しています。本プログラムの使用ならびに使用不能におけるいかなる損害に関して一切責任を負わないものとします。また、本プログラムに不具合が発見されても作者は不具合を修正する義務を負わないものとします。