

FLX-PCI 2 ***User's Manual***

Rev.1.01

日本語版

Photron
PHOTRON LIMITED
2010

Table of contents

Chapter 1 概要	5
1.1. はじめに	6
1.2. 保証について	7
1.3. このマニュアルの使用方法	8
1.4. 注意事項	9
1.5. 仕様	10
1.6. 動作環境	12
Chapter 2 準備	14
2.1. 梱包について	15
2.2. ドライバのインストール方法	16
2.2.1 FLX-PCI 2 ボードを装着する	16
2.2.2 付属 CD-ROM を準備する	17
2.2.3 ドライバソフトをインストールする (Windows2000/XP の場合)	18
2.2.4 ドライバソフトをインストールする (Linux (RedHat7.3) の場合)	25
Chapter 3 サンプルソフトの使用法	32
3.1. VidCap の使い方	33
3.2. AmCap の使い方	36
3.3. xawtv の使い方	41
Chapter 4 FLX-PCI 2 用ライブラリ	47
4.1. FLX-PCI API 関数一覧	48
4.2. FLX-PCI API 関数リファレンス	50
4.3. フレームメモリのデータ構造	75

Memo

Chapter 1 概要

1.1 はじめに

1.2 保証について

1.3 このマニュアルの使用方法

1.4 注意事項

1.5 仕様

1.6 動作環境

1.1 はじめに

FLX-PCI 2 は PC/AT 互換機の PCI バスに装着される拡張ボードです。

FLX-PCI 2 は、テレビやビデオ、CCD カメラ等の映像信号を取り込み、パソコンで扱える画像ファイルとして保存することができます。カメラからの信号をパソコンに入力できるボードは、ビデオ編集などの用途として民生用としても数多く販売されています。フォトロン社の FDM シリーズ (FDM-XXX、FLX-PCI 2) に代表される産業用途のボードと、民生用のボードのもっとも大きな違いは、

- ・ ボードをプログラムによって制御できるようにドライバ・開発環境(SDK)を用意している
- ・ 製品の安定・長期供給に配慮していること

の 2 点といえます。

ハードウェアの機能としては、民生用・産業用ともそれほど大きな違いはありませんが、産業用の画像ボードは、各種産業機器での使用を考慮した開発環境の公開・提供、および産業機器への組み込みの際に必要な安定供給体制に重点をおいた製品となっています。

「注意」

- (1) 本装置の性能を十分に発揮させて正しくお使い頂くために、取扱説明書(本書)を必ず最後までお読みいただくようお願い致します。
- (2) 本書の内容につきましては万全を期して作成しましたが、万一ご不審な点や記載漏れ、誤りなどお気づきの点がございましたらご連絡ください。
尚 運用の結果につきましては、責任を負いかねますのでご了承ください。
- (3) 本書の内容の一部または全部を無断転載する事は禁じられています。
- (4) 本書の内容に関しては将来予告無しに変更する事があります。
- (5) 当社では本装置の運用を理由とする損失・逸失利益等の請求につきましては、上記項目にかかわらず、いかなる責任も負いかねますので予めご了承ください。
- (6) 本装置を使用され人身事故・財産損害などが生じても当社はいかなる責任も負いかねます。
- (7) 本機におけるソフトウェアを全部または一部を著作者の許可なく複製したり複製物を頒布したりすると、著作権の侵害となります。

1.2 保証について

新規購入時の保証 PHOTRON FLX-PCI 2

「フォトロン FLX-PCI 2」及びそのアクセサリ類についての保証期間は、出荷日より一年間です。ただし、保証カードが、納入後 30 日以内に弊社へ返送されている事を条件とします。機器保証サービスは次の通りです。

1. 修理サービス： 当方へ送られた場合、無償で修理いたします。
2. パーツの交換： 保証期間中に交換されるパーツは無償で提供させていただきます。

尚、以下のような場合、この保証は適用されません：

- ・本製品を株式会社フォトロンの指示通りに正しくご使用にならなかった場合（当マニュアルに示した環境仕様に従っていただく必要があります）。
- ・本製品が「不慮のダメージを受けている」、「誤操作が行われた」、「間違って使用された」などの事故、誤用または乱用の痕跡が認められるとき。
- ・本製品がフォトロンのスタッフ、フォトロンで研修を受けたお客様側のスタッフ以外の人によって、あるいはフォトロンの了承なしに修理、変更、又は不正使用された場合。
- ・輸送中の損傷はこの保証の範囲に含まれません。輸送中の損傷に関する損害賠償は、お客様の責任において行ってください。

フォトロンは特定目的のための商品性や適合性を含め、明示暗示を問わず、他の保証をいたしません。もしこのフォトロン FLX-PCI 2 が保証期間中に正常動作をしない場合は、上記条件により無償で修理致します。当保証に関する株式会社フォトロンの義務は、無償修理に限らせていただきます。

1.3 このマニュアルの使用方法

用語の定義

このマニュアルにはいくつかの情報が、「注」、「注意」、「警告」として提示されています。これら三つの用語の意味を認識することは重要です。

「注」

当 FLX-PCI 2 の操作に関して強調したい情報を含んでいる注意書きです。

「注意」

注意とは、ある操作または状況が FLX-PCI 2 に対してダメージを与え得るということの注意を喚起します。

「警告」

FLX-PCI 2 を操作する人の安全に取って重要な警告であり、いかなる場合でも無視されるべきではありません。

Chapter 1, 概要

保証、注意事項、はじめに、およびこのマニュアルの使用方法を含みます。 .

Chapter 2, 準備

FLX-PCI 2 を構成する各部分の概要を説明します
また、カメラとグラブボードの各コネクタについても説明します。

Chapter 3, 付録

カメラとグラブボードの性能仕様を説明します。

1.4 注意事項

温度

フォトロン FLX-PCI 2 は、周囲温度摂氏 5～35 度の温度、かつ結露のない環境で正常動作するよう設計されています。

動作に必要な PC の動作温度に関しましては、ご使用になられる PC の仕様に準じます。

フォトロン FLX-PCI 2 とご使用になられる PC との動作保証温度が異なる場合は、共に動作保証を満たす範囲（上限に付きましては低い方／下限に付きましては高い方の範囲）となります

輸送

当装置の輸送の際には、PC からいったんグラブボード、カメラを取り外し、装置がもともと入ってきた輸送カートンをご使用願います。また、輸送中に結露しないように注意してください。

警告

この製品は、パソコンの電源コードを介してグラウンドに接続されています。この保安用グラウンドの接続は当装置の安全操作に不可欠です。正しく配線したコンセントに電源プラグを指し込むことにより、感電を防ぎます。理由の如何にかかわらず、防護接地がなくなると感電の原因になります。正常な電源コードを使用し、電源コードの状態が正常であることを確認してください。

1.5 仕様

対応バス	PCI
コントローラ	CONEXANT 社製 Bt878KHF 又は、FUSION 878A
対応キャプチャフォーマット	RGB555、RGB24、YUY2
ビデオ入力フォーマット	NTSC_J,NTSC,PAL_M
ビデオサイズ	80x60、88x72、128x96、160x120、176x144、240x176、240x176、240x180、320x240、352x288、640x240、640x480
ビデオフレームレート	最大 30fps (次ページ「注意」参照)
入出力端子	コンポジットビデオ入力 (RCA ジャック) ×4 系統
Plug & Play	対応
電源	PCI バスより供給
電源電圧	5V/12V
消費電流	最大 300mA (5V)、最大 120mA (12V)
外形寸法	W135×H125mm
動作環境	温度 5～35°C、湿度(結露なきこと)

「注意」

FLX-PCI 2 の画像取り込み速度は、コンピュータの速度や取り込み解像度に大きく依存します。
 下記に FLX-PCI 2 の画像取り込み速度の一例を示します。

【環境】

MotherBoard	RICOH® FB6
Chipset	Intel® 875E
CPU	Intel® Pentium®IV 2.80GHZ
Memory	DDR 512Mbyte
VideoBoard	ELSA® GADIAC FX736
OS	Microsoft® Windows XP Professional SP1 RedHat® Linux 7.3
Camera	SONY® XC-77 CCD VIDEO CAMERA MODULE

【計測結果】

VGA : 640x480 [pixel]
 QVGA : 320x240 [pixel]

Windows XP (標準ドライバ)

	QVGA	VGA
AMCAP	15	15
VIDCAP	30	30
FLXCAP	30	15

※30fps,3sec で撮影
 ※FLXCAP は 15bit color

RedHat Linux 7.3

	QVGA	VGA
xawtv (15bit)	3~17	3~8
xawtv (24bit)	8~10	非サポート
xawtv (jpeg)	25~30	非サポート
xawtv (mpeg)	28	23~24

※取り込む映像フォーマットは
 Microsoft AVI(RIFF)format
 no sound,44100Hz,30fps

1.6 動作環境

FLX-PCI 2 を使用するには、下記の環境が必要です。

コンピュータ	PCI バス搭載 PC/AT 互換機
CPU	Pentium 166MHz 以上
PCI バス	PCI バス Revision2.1 以上 (バースト転送モードをサポートしているもの) UMA 型チップセットを搭載した機種では、あらかじめ動作確認を行う必要があります。動作確認無しでの不具合につきましては保証致しかねます(※1)
メモリ	64MB 以上
OS(※2)	Microsoft® Windows 2000 / XP Microsoft® Windows 2003 Server(※3) Microsoft® Windows Vista Microsoft® Windows 7(※4) RedHat Linux 7.3 (Kernel Version 2.4 of Red Hat 7.3)
使用リソース	IRQ 1 チャンネル および 1KB メモリ領域

※1 「注意」

PCI バスが上記環境以外の場合、画像の取り込みができない等、満足な結果が得られないことがあります。

※2 「注意」

動作可能 OS は全て 32bit が対象となります。64bit OS での動作は対応しておりません。

※3 「注意」

Service Pack2 以降がインストールされた環境下におきましては、PAE モードを OFF にするか、搭載メモリを 2GB 以下にしてご利用下さい。

※4 「注意」

Windows7 におきましては、RGB24bit でキャプチャを行った際、320x240 の解像度でのキャプチャとなります。VGA (640x480) でのキャプチャを行う際は、RGB555 (15bit) の色モードのみ対応となります。

Memo

Chapter 2 準備

2.1 梱包について

2.2 ドライバのインストール方法

2.1 梱包について

本製品の標準構成は以下の通りです。
製品を梱包から取り出し、本体・付属品を確認してください。

- | | | |
|----|-------------------------|-----|
| 1. | グラブボード (PCI ボード) | 1 台 |
| 2. | FLX-PCI 2 Drivers Disk | 1 枚 |
| 3. | FLX-PCI 2 User's Manual | 1 冊 |
| 4. | お客様登録カード | 1 枚 |
| 5. | ソフトウェア使用許諾契約書 | 1 枚 |
| 6. | 保証書 | 1 部 |

2.2 ドライバのインストール方法

FLX-PCI 2 のコンポーネントについて、

「警告」 および 「注意」

- ・強い衝撃を与えたりしないで下さい。
- ・引火性ガスや粉塵が発生する場所では使用しないで下さい。
- ・ぐらついた台の上や傾いた場所など、不安定な場所には置かないで下さい。
- ・分解・改造をしないで下さい。
- ・水などの液体で濡らさないで下さい。
- ・無理な力がかかるような使い方はしないで下さい。

2.2.1 FLX-PCI 2 ボードを装着する

FLX-PCI 2 ボードをパソコンに装着する方法について説明します。

1. パソコンおよび周辺機器の電源を切ってください。
2. パソコンのカバーを取り外してください。
※ 取り外し方法については、パソコンの付属マニュアルをご参照ください。
3. 空いている PCI スロットのカバーを外してください。
※ カバーを固定してあるネジはボードの固定に使用しますので、なくさないようにしてください。
4. FLX-PCI 2 ボードを PCI スロットに装着して、ネジで固定してください。
5. パソコンのカバーを取り付けてください。

2.2.2 付属 CD-ROM を準備する

FLX-PCI 2 ボードに付属する、CD-ROM は 1 枚です。

FLX-PCI 2 Drivers Disk の内容は以下の通りです。

¥¥App¥AMCam¥	DirectShow を使用したサンプルソフト
¥¥App¥VidCap¥	VFW を使用したサンプルソフト
¥¥App¥Xawtv¥	Linux 用のサンプルソフト
¥¥Driver¥Windows2000¥	Windows2000/XP/2003/Vista/7 用標準ドライバ
¥¥Driver¥Windows2000Ex¥	Windows2000/XP/2003/Vista/7 用追加ドライバ(※1)
¥¥Driver¥Linux¥	Linux 用ドライバ
¥¥FLXCAP¥flxcap¥	FLX 用ライブラリ「FDMPCI.DLL」を使用したサンプル (Visual C/C++)
¥¥FLXCAP¥flxcap¥Release¥	FLXCAP サンプルソフト実行ファイル
¥¥VisualBasic¥fdmVB¥	FLX 用ライブラリ「FDMPCI.DLL」を使用したサンプル (Visual Basic)

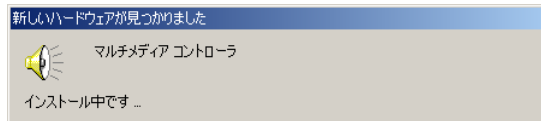
サンプルソフト実行ファイルについては、FLX-PCI 2 ボードが装着されている環境でご使用ください。

※1 追加ドライバに関して

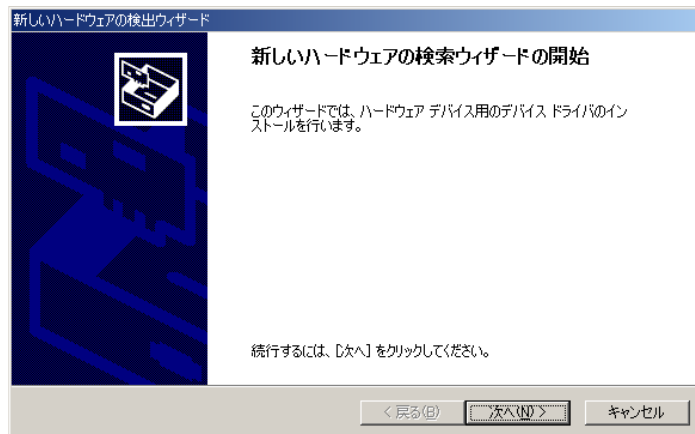
お客様の PC 環境によって標準ドライバを導入しても動作しない場合、追加ドライバを導入することにより動作する可能性がありますのでご利用ください。

2.2.3 ドライバソフトをインストールする (Windows2000/XP の場合)

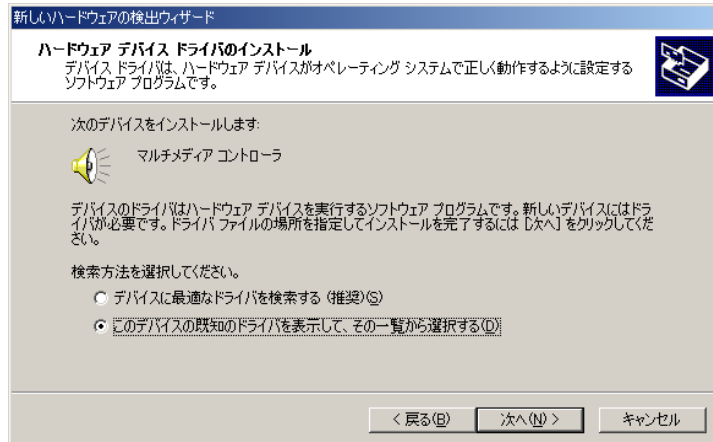
1. FLX-PCI 2 ボードをパソコンに装着後、パソコンの電源を入れます。Windows が起動したら、次のウィンドウが表示されます。



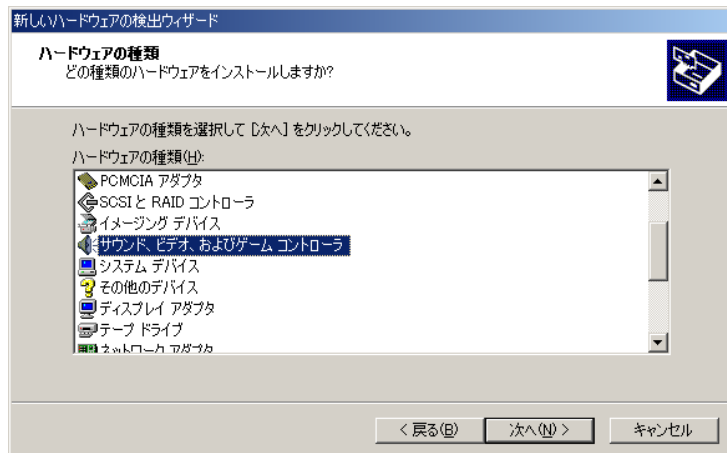
その後、次のウィンドウが表示されますので「次へ」をクリックします。



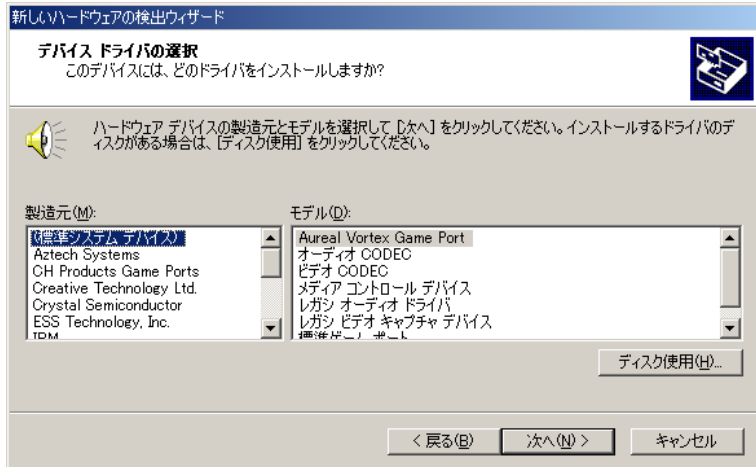
- 次にインストール方法を選択します。「このデバイスの既知のドライバを表示して、その一覧から選択する」を選択し、「次へ」を押します。



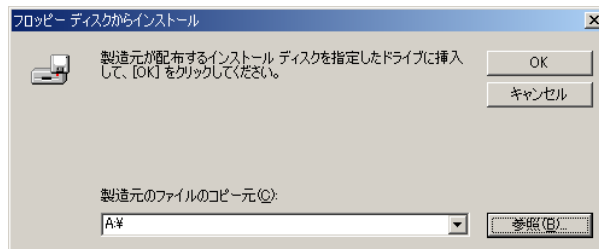
- 次にハードウェアの種類を選択します。「サウンド、ビデオ、およびゲームコントローラ」を選択して、「次へ」を押します。



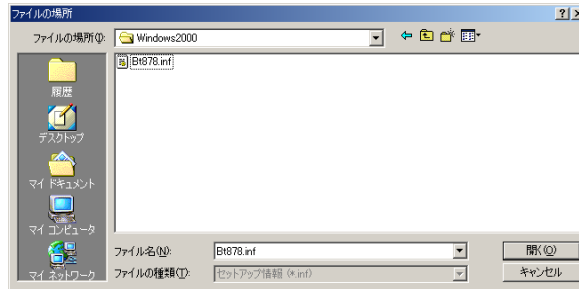
4. デバイスドライバーを選択します。ここでは「ディスク使用」を押します。



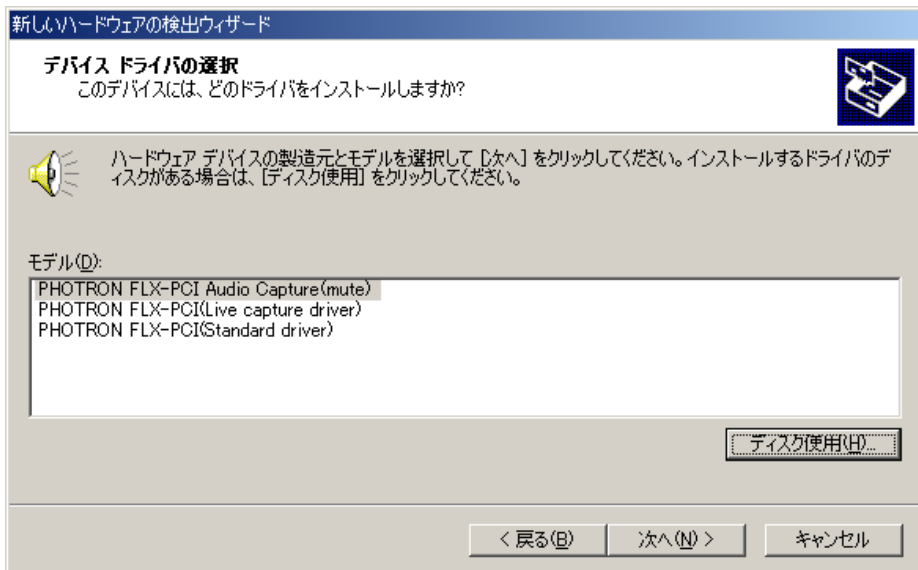
5. CD-ROM を入れて、「参照」を押します。



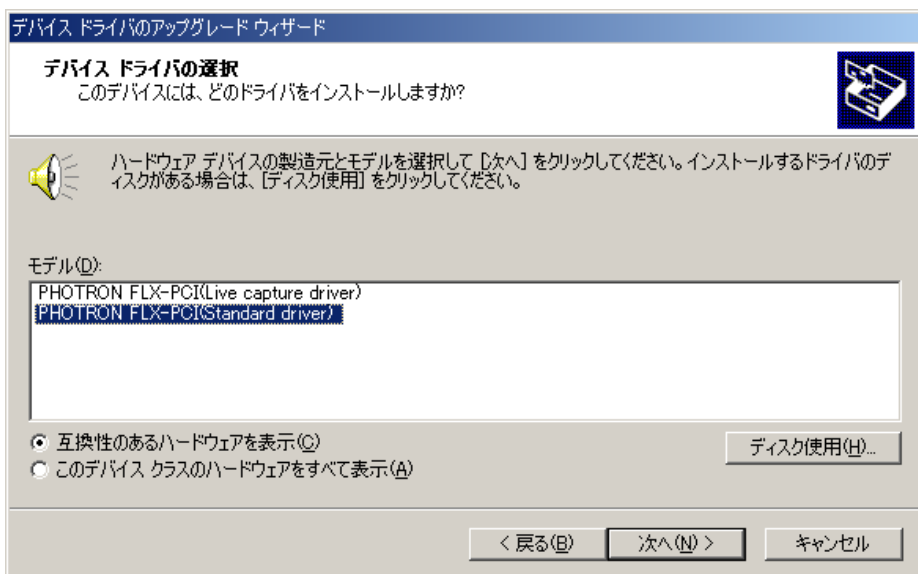
6. 次のウィンドウで「D:¥Driver¥Windows2000¥」の下の「bt878.inf」ファイルを選択し「開く」を押します。
 - ※ ドライブ名はお客様の環境によって変わります。CD-ROMが割り当てられているドライブ名を指定してください。



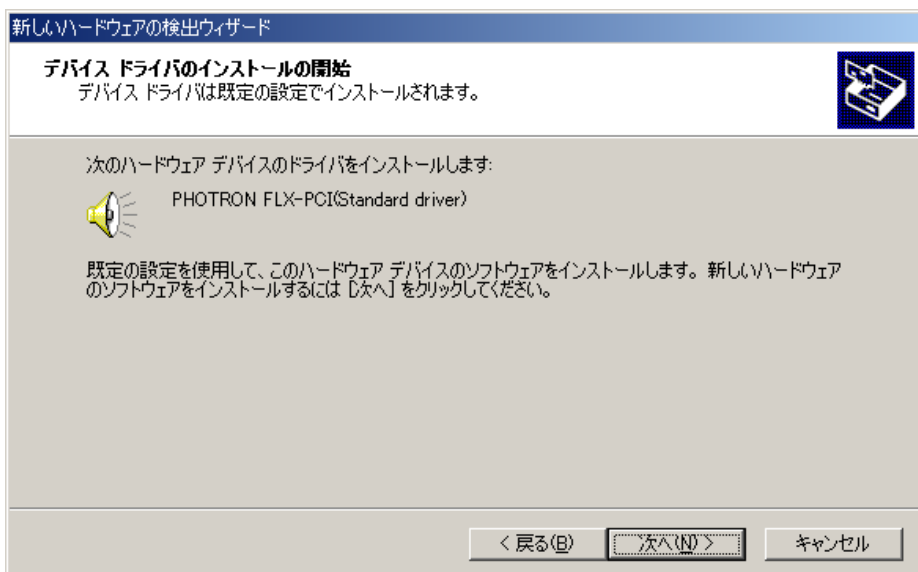
7. 「開く」を押して、下のように複数のドライバが表示された場合には一度[戻る]を押します。



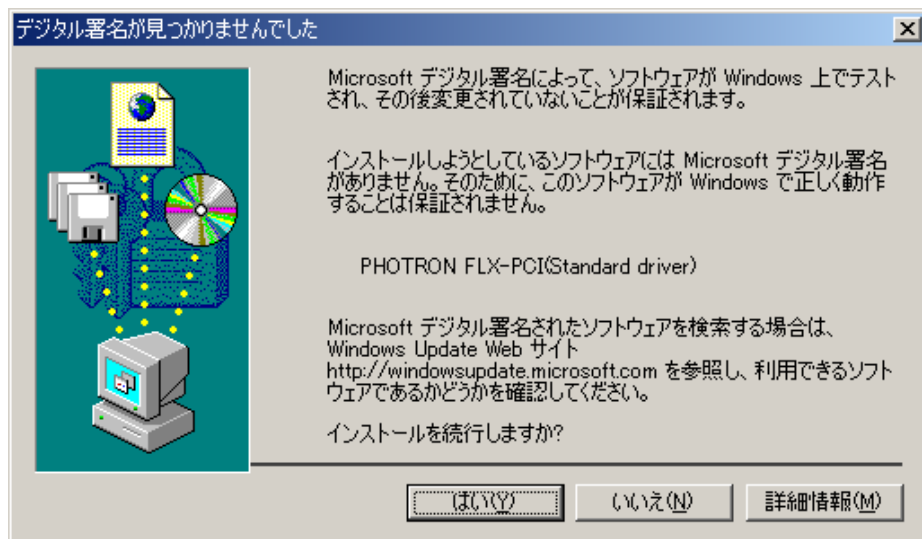
- 一度戻り、もう一度「次へ」を押すと、下のように互換性のあるドライバのみ表示されます。この状態で「次へ」を押します。



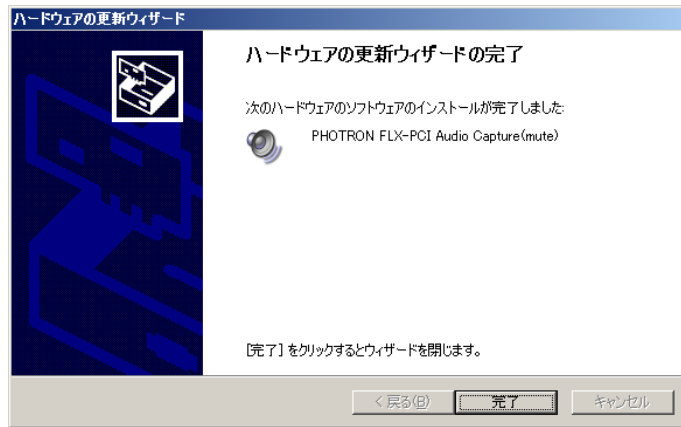
- 下のようなウィンドウが出たら、「次へ」を押します。これでインストールが始まります。



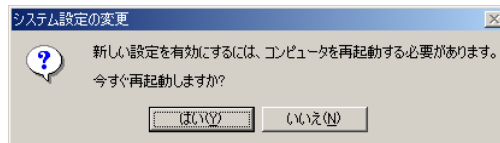
10. 一つのドライバのインストールが終ると、以降のドライバは次のようなウィンドウで「はい」を押すことにより、インストールできます。このように「PHOTRON FLX-PCI(Standard driver)」と「PHOTRON FLX-PCI Audio Capture(mute)」の2つのドライバをインストールします。



11. すべてのドライバのインストールが終ると以下のウィンドウが表示されますので、「完了」を押します。



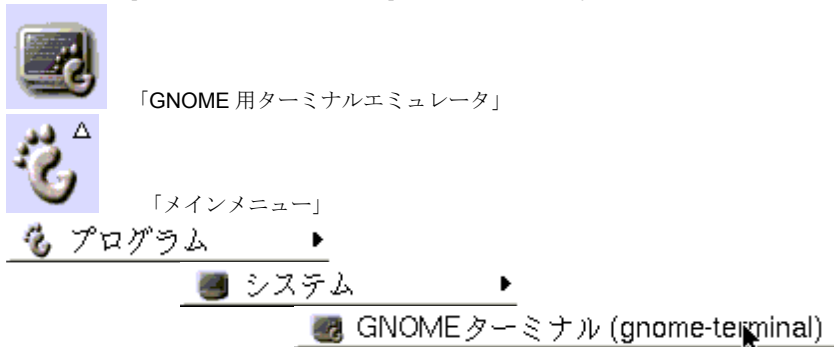
12. コンピュータを再起動させます。
下記のようなコンピュータを再起動させるような注意が促されない場合もありますが、必ずコンピュータを再起動させて下さい。



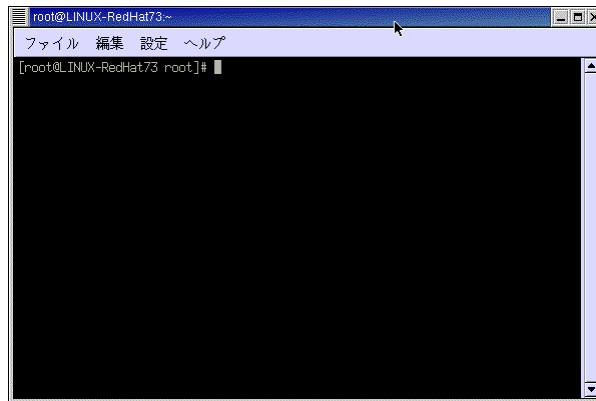
※インストールが途中で失敗する、インストールが完了しても動作しない等の問題が発生した場合、CDの「¥Driver¥Windows2000¥¥」に含まれる追加ドライバの導入をお試しください。

2.2.4 ドライバソフトをインストールする (Linux(RedHat7.3)の場合)

1. FLX-PCI 2 ボードをパソコンに装着後、パソコンの電源を入れます。
2. ログイン画面が表示されたら、「root」でログインします。デスクトップが表示された後、左下の「GNOME 用ターミナルエミュレータ」
3. をクリックして、ターミナルを表示させます。または「メインメニュー」→「プログラム」→「システム」→「GNOME ターミナル」を選んでください。



4. 「GNOME ターミナル」のウィンドウが表示されます。これ以降の四角で囲った操作はこのウィンドウの中で行います。「root@*****」の*****の部分はホスト名を示していますので、インストールしているマシンによって変わります。ここではホスト名を「LINUX-RedHat73」としています。



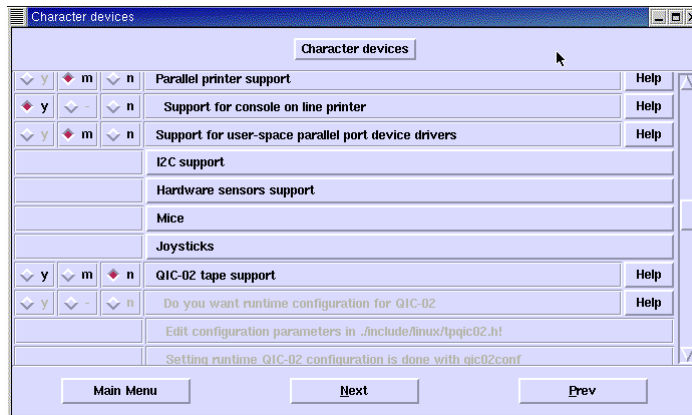
5. ドライバをビルドのための準備をします。すでに行われている場合は必要ありません。

- 正しい.config をコピーした後、xconfig を立ち上げます。

```
[root@LINUX-RedHat73 root]# cd /usr/src/linux-2.4  
[root@LINUX-RedHat73 root]# cp .config ~  
[root@LINUX-RedHat73 root]# cp /boot/config-2.4.18-3 .config  
[root@LINUX-RedHat73 root]# make xconfig
```

メッセージが表示されます。

- 「LINUX Kernel Configuration」のウィンドウが表示されます。



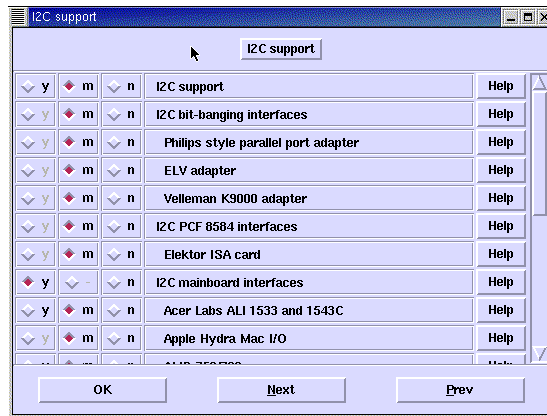
- 「Character devices」をクリックします。「Character devices」のウィンドウが開きます。

Character devices

9. 下に行って「I2C support」をクリックします。「I2C support」のウィンドウが開きます。

I2C support

10. 「I2C support」「I2C bit-banging interfaces」を「m」にします。その後「OK」を押して、ウィンドウを閉じ、「Character devices」のウィンドウの「Main menu」を押して、メインメニューに戻ります。



11. 「Multimedia devices」をクリックします。

Multimedia devices

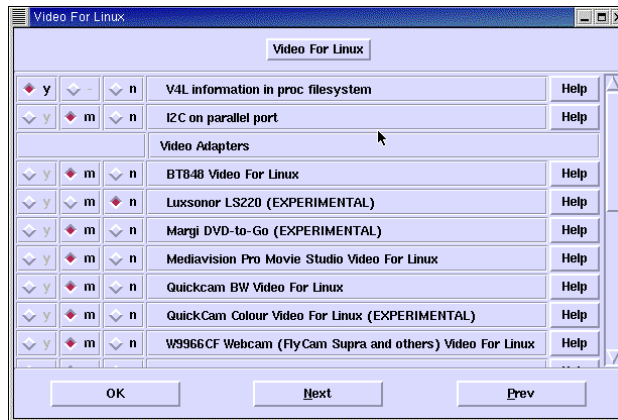
12. 「Multimedia devices」のウィンドウが表示されます。「m」の左のラジオボタンをクリックします。既に押下されている場合、押す必要がありません。



13. 「Video For Linux」をクリックします。「Video For Linux」のウィンドウが表示されます。

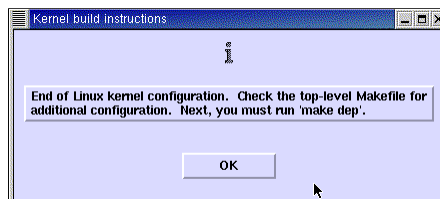
Video For Linux

14. 「BT848 Video For Linux」を「m」にします。「OK」を押して、「Multimedia devices」のウィンドウに戻り、「Main menu」を押します。「Linux Kernel Configuration」に戻ったら、「Save and Exit」を押して設定を保存します。



Save and Exit

15. 「Kernel build instructions」のウィンドウが表示されたら「OK」を押します。



16. ターミナルに戻り、下のように入力をビルドします。

```
[root@LINUX-RedHat73 root]# make dep clean modules
[root@LINUX-RedHat73 root]#
```

17. FD ディスクからファイルをコピーします。FD ディスクをマシンに挿入し、マシンから読めるようにするためにファイルシステムにマウントします。ディスクのデータを HDD にコピーします。コピーが終わったら、アンマウントします。

```
[root@LINUX-RedHat73 root]# mount /dev/fd0 /mnt/floppy
[root@LINUX-RedHat73 root]# cd /mnt/floppy/Driver/LINUX
[root@LINUX-RedHat73 root]# cp * ~
[root@LINUX-RedHat73 root]# umount /mnt/floppy
```

18. ドライバをコンパイルした後、インストールします。ホームディレクトリに移動して、ファイルを展開します。その後、ドライバをコンパイルし、インストールします。

```
[root@LINUX-RedHat73 root]# cd ~
[root@LINUX-RedHat73 root]# tar zxvf bttv-0.7.95.tgz
```

展開されたファイルの情報が表示されます。

```
[root@LINUX-RedHat73 root]# cd bttv-0.7.95/driver
[root@LINUX-RedHat73 root]# make
```

コンパイル過程の情報が表示されます。

```
[root@LINUX-RedHat73 root]# make install
```

インストール過程の情報が表示されます。

19. ドライバの依存関係を明示する。エディターで「/etc/modules.conf」を編集する。以下は `emacs` を使う場合。`emacs` の使い方については他の資料を参照して下さい。

```
[root@LINUX-RedHat73 root]# emacs /etc/modules.conf
```

以下の 6 行を追加して下さい。

```
# i2c
alias char-major-89 i2c-dev
#bttv
alias char-major-81 videodev
alias char-major-81-0 bttv
options bttv card=81
```

編集が終わったら下のように打ち込み、ドライバの依存関係を更新します。

```
[root@LINUX-RedHat73 root]# /sbin/depmod -a
```

20. ドライバをロードして確かめてみます。

```
[root@LINUX-RedHat73 root]# modprobe bttv
[root@LINUX-RedHat73 root]# lsmod
```

ロードされているドライバの情報が表示されます。

以下のような「`bttv`」の行があることを確認します。

Module	Size	Used
<code>bttv</code>	***	0
<code>videodev</code>	***	2
<code>l2c-algo-bit</code>	***	1
<code>l2c-core</code>	***	0

ドライバをアンロードするには以下のようにします。

```
[root@LINUX-RedHat73 root]# modprobe -r bttv
```

21. ドライバを起動時にロードする方法。ドライバを起動時にロードさせたい場合は以下のよう
に設定してください。

```
[root@LINUX-RedHat73 root]# cd ~/bttv-0.7.95  
[root@LINUX-RedHat73 root]# cp bttvmk /etc/init.d/  
[root@LINUX-RedHat73 root]# chkconfig --add bttvmk  
[root@LINUX-RedHat73 root]# chkconfig bttvmk on
```

Chapter 3

サンプルソフトの使用方法

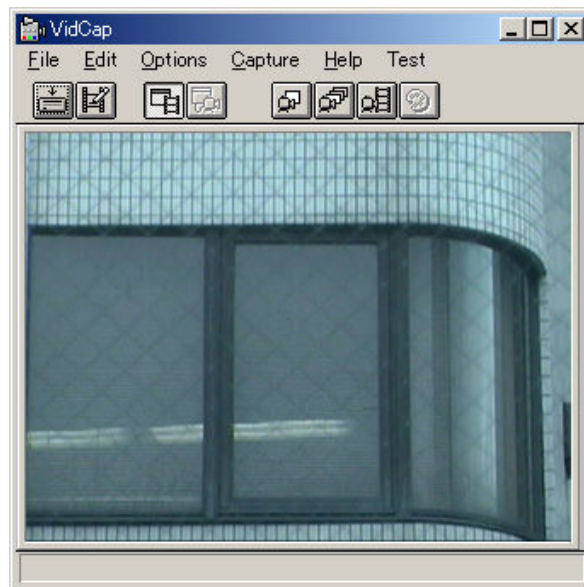
3.1. VidCap の使い方

3.2. AmCap の使い方

3.3. xawtv の使い方

3.1. VidCap の使い方

Windows 2000/XP のドライバは VFW に対応しています。VFW を使用したサンプルアプリとして VidCap が添付されています。これは正常にボードが設定できたことやその他ビデオ入力を確認するためなど一般的な目的にお使いいただけます。以下の一連の操作により、このアプリケーションの基本的な機能を使うことができます。



【プレビュー】

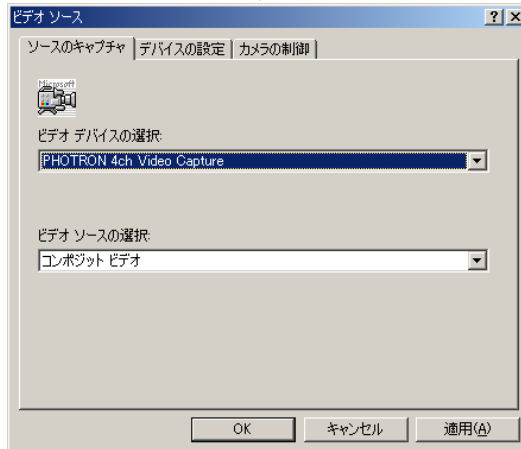
「Preview」ボタン（ツールバーの一番左から3番目のボタン）でプレビューの ON/OFF ができます。プレビューが ON のときにはビデオ入力からの映像が画面に表示され定期的にアップデートされます。

【シングルフレームキャプチャ】

「Single Frame」ボタン（ツールバーの右から4番目のボタン）はシングルフレームキャプチャに使用されます。このボタンを押すたびに一枚の画像がキャプチャされウィンドウに表示されます。この画像をコピーして、他のアプリケーションで貼り付けることができます。

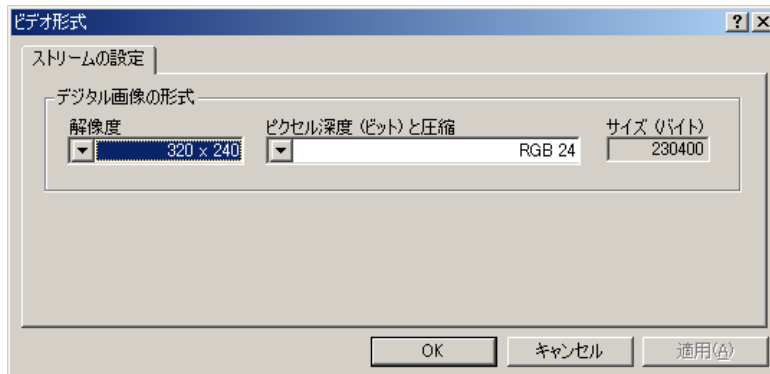
【ビデオキャプチャドライバの設定】

VidCap32 のアプリからドライバの設定ダイアログを表示することができます。メニューの [Options]-[Video Source...] でビデオ入力ソースの選択やカラーの調整などを設定できます。それぞれ、以下のようなダイアログが表示されます。



【キャプチャサイズ、画像フォーマットの設定】

ビデオのキャプチャサイズを設定する場合にはメニューの [Options]-[Video Format] でビデオ形式を設定するウィンドウが表示されます。このウィンドウでビデオの画像サイズ、キャプチャフォーマットを設定できます。デフォルトでは 320x240 となっています。



【キャプチャファイルの設定】

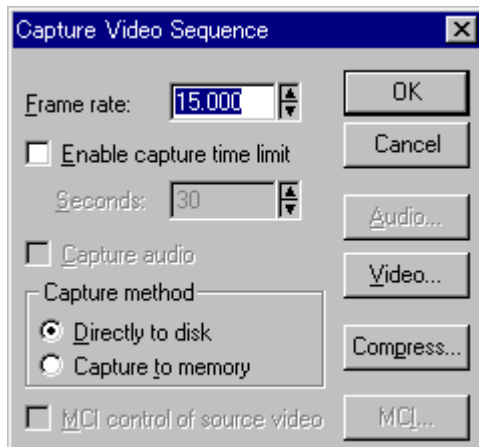


[Set Capture File...]ボタン（ツールバーの一番左のボタン）でキャプチャしたデータを保存するファイルを設定します。

【キャプチャ開始】



[Video]ボタン（ツールバーの一番右から2番目のボタン）を押すと、[Capture Video Sequence]のウィンドウが表示されます。



このウィンドウの[Frame rate]でフレームレートを設定します。[Enable capture time limit]をチェックして、[Second]に秒数を設定すれば、指定した時間だけキャプチャすることができます。設定が終わったら[OK]を押します。

すると、下のダイアログが表示され、[OK]を押すとキャプチャが開始されます。下の例ではキャプチャした画像を保存するファイルを「C:\CAPTURE.AVI」に指定しています。



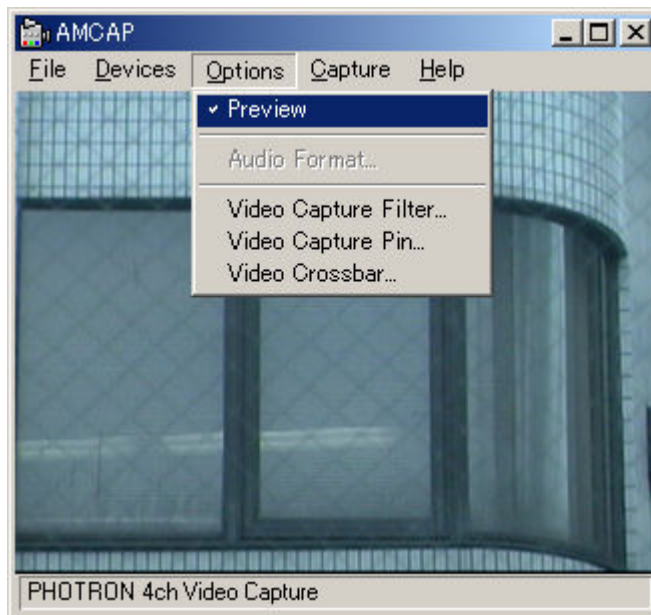
【その他】

VidCap 及び VFW の使い方については Platform SDK のドキュメントを参照して下さい。以下のページから Platform SDK をダウンロードできます。

<http://www.microsoft.com/msdownload/platformsdk/sdkupdate/>

3.2. AmCap の使い方

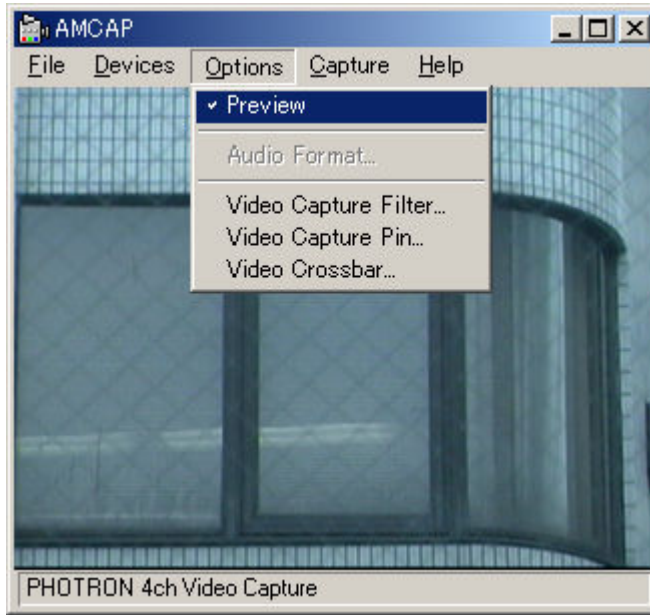
AMCap は FLX-PCI 2 のパッケージに含まれているビデオを見るためのサンプルアプリケーションです。これは正常にボードが設定できたことやその他ビデオ入力を確認するためなど一般的な目的にお使いいただけます。AMCap と VidCap32 の主な違いとして、AMCap は VFW API の代わりに DirectShow API を使用している点です。その他 2 GB を超えるキャプチャファイルを扱うことができます。



【デバイスの選択】

AMCap はプレビューやキャプチャの前の入力デバイスを選択する必要があります。メニューの [Device] で使用可能なオーディオやビデオキャプチャーデバイスのリストが表示されますので、[Device]-[Photron 4ch Video Capture] を選択します。

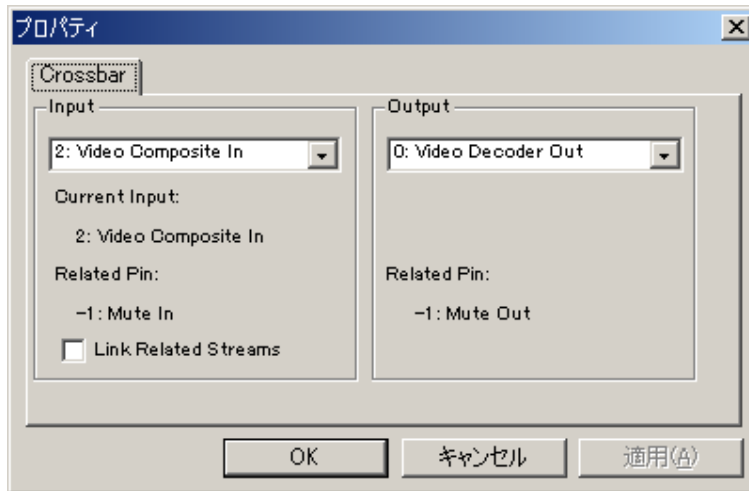
【プレビュー】



最初 AMCap を立ち上げた時点ではビデオ入力の映像は表示されません。メニューの [Options]-[Preview]によりプレビューモードを選択する必要があります。プレビューが ONになればビデオ入力が画面に表示され、定期的に更新されるようになります。このとき表示される画像はプレビューに特化しているのでキャプチャによって保存される画像と若干異なる場合があります。

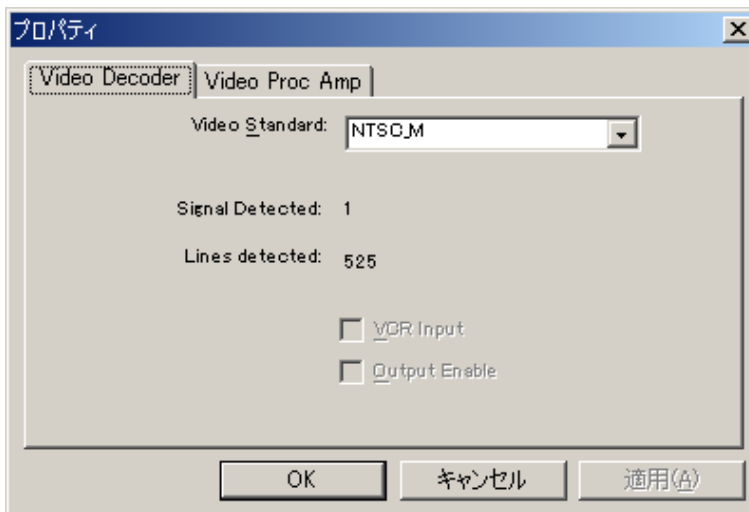
【ビデオ入力端子の選択】

AMCap のアプリからドライバの設定ダイアログを表示することができます。メニューの [Options]-[Video Crossbar...] でビデオ入力ソースの選択できる下のようなダイアログが表示されます。左側の [Input] のコンボボックスから入力端子を選択します。



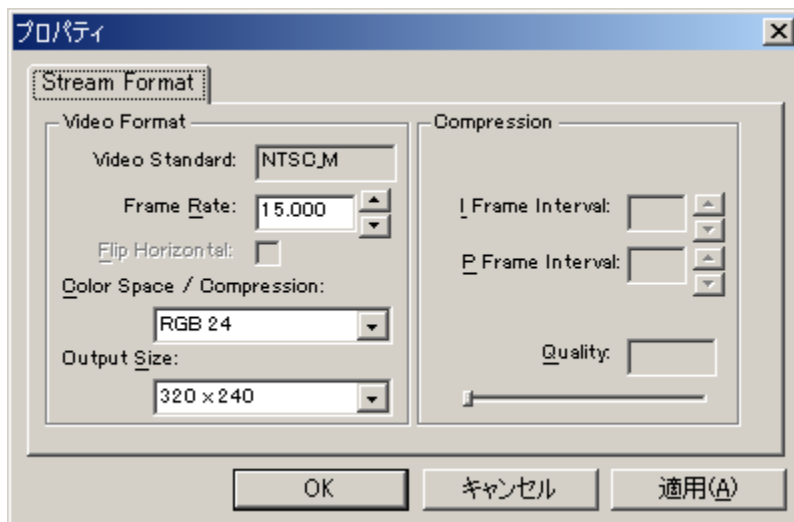
【ビデオフォーマットの設定、カラーの調整】

AMCap のアプリからドライバの設定ダイアログを表示することができます。メニューの [Options]-[Video Capture Filter...] で表示される下のようなウィンドウの [Video Decoder] のタブを選択し、入力されるビデオ信号を選択でき、[Video Proc Amp] のタブを選択することで、カラーの調整などを設定できます。



【ビデオのキャプチャサイズ、フォーマット、フレームレートの選択】

ビデオのキャプチャサイズを設定する場合にはメニューの[Options]-[Video Capture Pin...]でストリーム形式を設定するウィンドウが表示されます。このウィンドウでビデオの画像サイズ (Output Size)、キャプチャフォーマット(ColorSpace/Compression)、フレームレート(Frame Rate)を設定できます。デフォルトの解像度は 320×240 となっています。



【キャプチャファイルの設定】

メニューの[File]-[Set Capture File]でキャプチャファイル設定ダイアログが表示されます。状況によってはかなりのコマ落ちがおこりますが、これはフレームサイズやフォーマット、システムのスピードなどに依ります。これらはキャプチャファイルを予め確保しデフラグメントすることにより改善できます。

キャプチャファイルを予め確保することの意味はハードディスクのある領域を前もって予約しておくことです。メニューの[File]-[Allocate Disk Space]で表示されるダイアログでファイルを作り、キャプチャしたい最大の映像に十分なハードディスクのスペースを予約します。複数の映像を保持するために複数のファイルを確保しておくことができます。

予め確保された領域を効率的に利用するためにハードディスクをデフラグメントする必要があります。デフラグメントとはハードディスク内に分散、断片化して保存されているファイルを一箇所の連続領域に置かれるように整理することです。

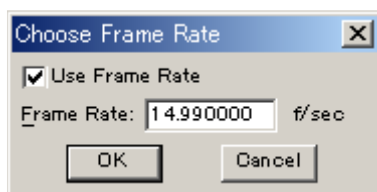
AMCap ではデフラグメントを行えません。Windows に付属している機能かサードパーティによって提供されているアプリケーションを必要とします。Windows 95/98/98SE/ME/2000/XP ではOS にデフラグメント機能がついていますのでそれを利用することができます。デフラグメントは必ずキャプチャファイルを確保した後に行います。

デフラグメントが終わったら、AMCap に戻ります。ファイルは連続的な領域に移動しています。このハードディスク内での物理的な配置はファイルの大きさが変わったり削除されたりしない限り維持されます。デフラグメントを使わないやり方としてはビデオキャプチャ用にパーティションを用意するやりかたもあります。

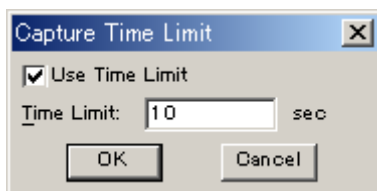
デフラグメントは大変時間がかかりますが、圧縮しないなどの高品質のビデオキャプチャにまつわる問題などには効果的です。まず必要とするファイルのサイズ、数を確認してください。一度、デフラグメントを行ったら間違えて消さないよう注意してください。それは半永久的な作業領域と考えて、特定の映像ファイルとして取り扱わないことをお勧めします。これらのファイルは別な名前を付けて他の領域にコピーして編集などを行ってください。

【AMCap を使ったビデオキャプチャについて】

AMCap はキャプチャを行う前にいくつか設定する項目があります。まず、メニューの [Capture][Set Frame Rate...] でフレームレートを設定します。最高品質でのキャプチャではコマ落ちを防ぐために十分遅いフレームレートを選択してください。



次にメニューの [Capture]-[Set Time Limit...] でキャプチャ時間を設定します。



正しい設定を設定ができていないか確認した後、メニューの [Capture]-[Start Capture] を選び表示されるダイアログで [OK] の確認ボタンを押してキャプチャを開始して下さい。もし、キャプチャ時間を設定していない場合には [Capture]-[Stop Capture] か [ESC] を押すことによりキャプチャを終了させます。

【再生】

一番簡単な方法は「マイ コンピュータ」から保存したキャプチャファイルを選び、ダブルクリックします。これにより Windows Media Player が立ち上がり、ファイルを自動的に再生します。Windows Media Player には再生、停止、一時停止などがあります。詳しくは Windows Media Player のオンラインヘルプを参照して下さい。

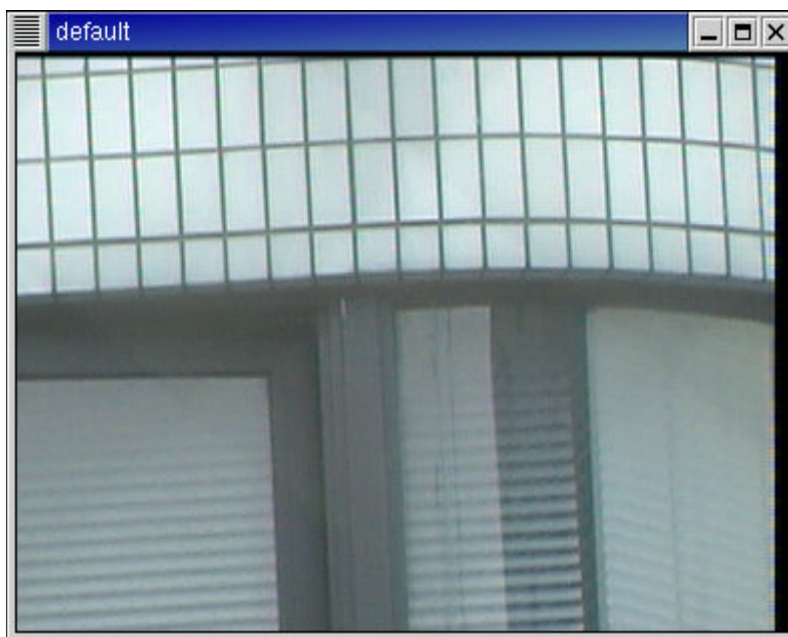
【その他】

AMCap 及び DirectShow の使い方については DirectX SDK のドキュメントを参照して下さい。以下のページから DirectX SDK をダウンロードできます。

<http://www.microsoft.com/japan/msdn/directx/downloads.asp>

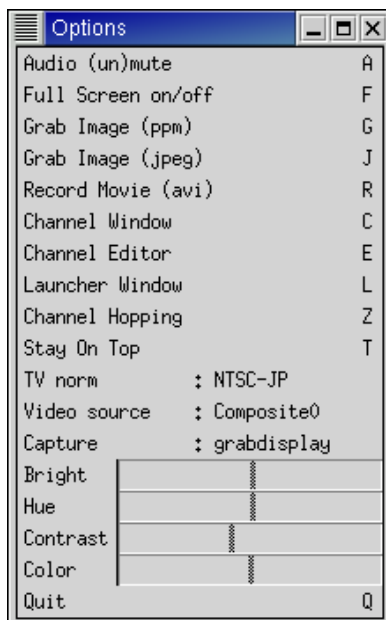
3.3. *xawtv* の使い方

xawtv は FLX-PCI 2 のパッケージに含まれているビデオを見るためのサンプルアプリケーションです。これは正常にボードが設定できたことやその他ビデオ入力を確認するためなど一般的な目的にお使いいただけます。下の一連の操作にこのアプリケーションの基本的な機能を使うことができます。下の画像はメインウインドウの様子です。

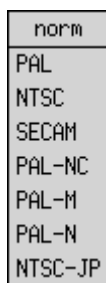


【操作パネル】

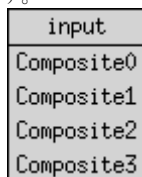
メインウインドウを右クリックすると、下のような操作パネルが表示されます。「TV norm」で入力信号のフォーマットを、「Video source」でどの端子の信号を入力するかを決めます。下のウインドウではそれぞれ「NTSC-JP」「Composite 0」が選択されています。



「TV norm」をクリックすると下のようなポップアップメニューが表示されます。それぞれの文字をクリックすることでフォーマットを選択できます。

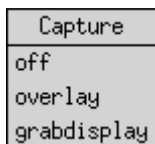


「Video source」をクリックすると下のようなポップアップメニューが表示され、それぞれの文字をクリックすると、端子を選択できます。



【プレビュー】

プレビューは操作パネルの「Capture」で「grabdisplay」「overlay」のいずれかを選ぶことで表示できます。リモートのPCから操作する場合などでは「grabdisplay」を選択して下さい。

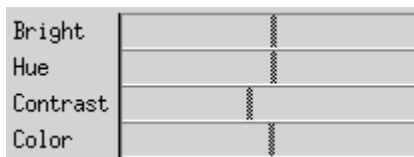


【シングルキャプチャ】

シングルフレームキャプチャは操作パネルの「Grab Image (ppm)」「Gram Image (jpeg)」のいずれかを押すことで実行できます。キャプチャされたファイルは「snap****」(****はキャプチャの度が変わる)という名前でカレントディレクトリに保存されます。キャプチャは最大サイズの 640x480 で行われます。[Ctrl+G][Ctrl+J]を押すとそれぞれ ppm、Jpeg でウインドウサイズのシングルフレームキャプチャが行われます。

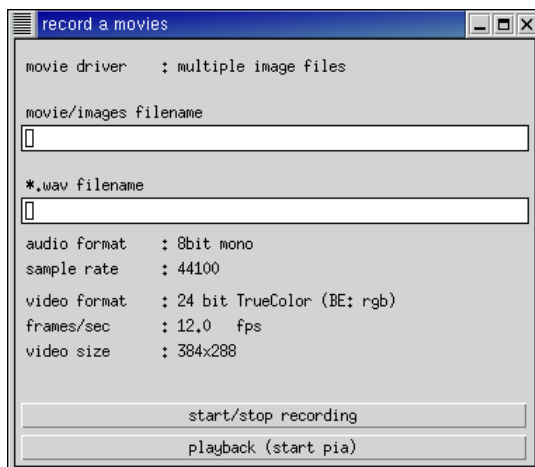
【その他の設定】

操作パネルから明るさ、コントラスト、鮮やかさ、色合いなどを設定することができます。これらのパラメータはそれぞれ操作パネルの「bright」「contrast」「hue」「color」の右のエリアで、右クリックで上がり、左クリックで下がります。

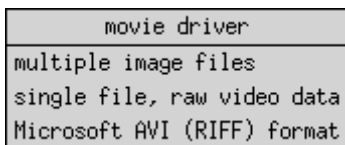


【キャプチャの準備】

キャプチャする画面のサイズはウィンドウサイズで決まるのでウィンドウサイズを調節します。操作パネルの「Record Movie (avi)」をクリックします。すると、下のようなウィンドウが表示されます。



まず、初めに「movie driver」をクリックして、保存する形式を選びます。この例では「Microsoft AVI (RIFF) format」を選びます。

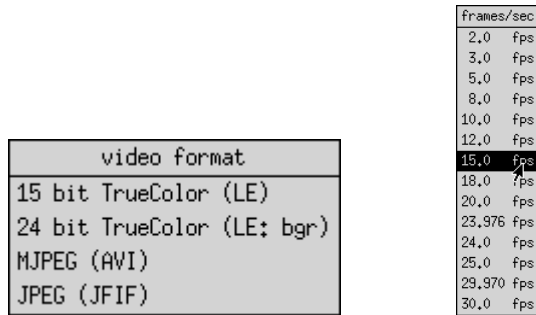


次に保存されるファイル名を指定します。「movie/images filename」の下にあるウィンドウの上にカーソルを持っていき、入力します。ここでは「~/movie.avi」としてホームディレクトリにmovie.aviというファイルを作ります。このファイル名を入力するウィンドウは複数行に渡って入力できてしまうため、入力する前にバックスペースキーですべての文字が削除されていることを確認してください。

次にオーディオをOFFにします。「audio format」の右をクリックして「no sound」を選択します。



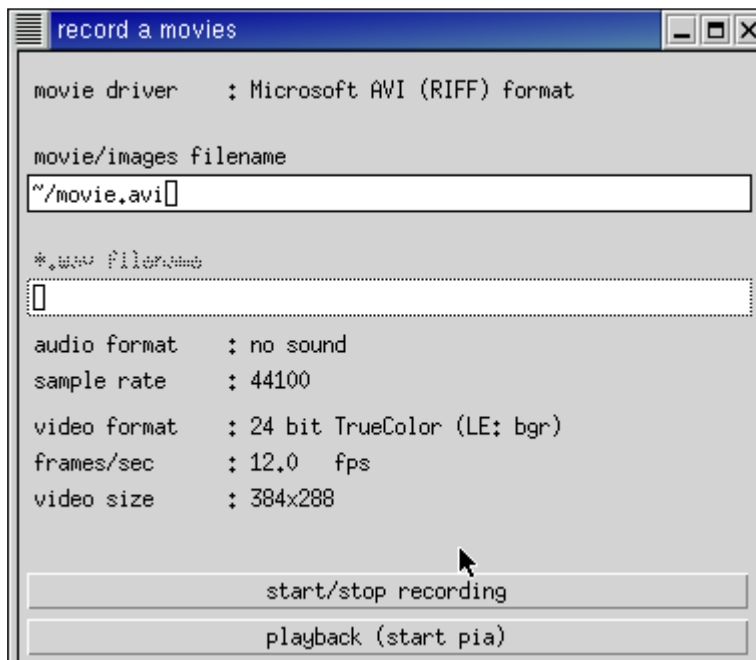
次に「video format」で保存するフォーマットを、「frames/sec」でキャプチャするレートを指定します。それぞれ、文字の右をクリックして表示されるウィンドウで指定します。ここでは「24 bit TrueColor (LE: bgr)」と「15.0 fps」をそれぞれ指定します。



「video size」はメインウィンドウのサイズであらかじめ調節しておきます。

【キャプチャの開始】

キャプチャする準備が整ったら、下の「start/stop recording」を押します。



キャプチャを止めるにはもう一度「start/stop recording」を押します。

【JPEG ファイルでのキャプチャの準備】

キャプチャは J P E G ファイルとして行うこともできます。「movie driver」をクリックして、「multiple image files」を選択します。「movie/images filename」は「~/image000.jpg」のように数字の部分が含まれるようにしておきます。キャプチャファイルはこの数字の部分が一つずつ増えた名称のファイルとして保存されます。「audio format」は「no sound」を選択します。「video format」は「JPEG(JFIF)」を選択し、「frames/sec」から保存したいフレームレートを指定します。「video size」はメインウィンドウのサイズであらかじめ調節しておきます。保存されたファイルは RedHat7.3 に付属している GQView などのアプリで確認できます。

Chapter 4

FLX-PCI 2 用ライブラリ

4.1. FLX-PCI API 関数一覧

4.2. FLX-PCI API 関数リファレンス

4.3. フレームメモリのデータ構造

4.1. FLX-PCI API 関数一覧

fdmpciInit()	: 初期化
fdmpciExit()	: 後処理
fdmpciSetInputResolution()	: 入力解像度の設定
fdmpciGetInputResolution()	: 入力解像度の取得
fdmpciSetOutputResolution()	: 出力解像度の設定
fdmpciGetOutputResolution()	: 出力解像度の取得
fdmpciSetStartPosition()	: 開始点の設定
fdmpciGetStartPosition()	: 開始点の取得
fdmpciSelectChannel()	: チャンネルの設定
fdmpciGetChannelNum()	: チャンネルの取得
fdmpciGetNumChannel()	: チャンネル数の取得
fdmpciSelectFrame()	: カレントフレームの設定
fdmpciGetFrameNum()	: カレントフレームの取得
fdmpciGetBuffer()	: カレントフレームのバッファのアドレスの取得
fdmpciReadPixel()	: ピクセルの読み込み
fdmpciWritePixel()	: ピクセルの書き込み
fdmpciRectRead()	: 領域の読み込み
fdmpciRectWrite()	: 領域の書き込み
fdmpciStopCapture()	: キャプチャの中止
fdmpciSingleCapture()	: シングルキャプチャ
fdmpciCapture()	: キャプチャ
fdmpciIsCapture()	: キャプチャ中かどうか調べる
fdmpciSetBrightness()	: ブライツネスパラメータの設定
fdmpciGetBrightness()	: ブライツネスパラメータの取得
fdmpciSetContrast()	: コントラストパラメータの設定
fdmpciGetContrast()	: コントラストパラメータの取得
fdmpciSetColorMode()	: カラーモードの設定
fdmpciGetColorMode()	: カラーモードの取得
fdmpciSaveBitmap()	: カレントフレームのセーブ
fdmpciLoadBitmap()	: カレントフレームへのロード
fdmpciCleanFrame()	: カレントフレームの消去
fdmpciCleanAll()	: すべてのフレームの消去
fdmpciSaveAVIFile()	: AVI ファイルのセーブ
fdmpciSaveBitmaps()	: BMP ファイルのセーブ
fdmpciLoadAVIFile()	: AVI ファイルのロード
fdmpciCaptureStart()	: キャプチャ開始
fdmpciIsCompleted()	: キャプチャ終了確認
fdmpciRecord()	: 範囲録画
fdmpciIs2ch()	: FDM-PCI 2CH かどうかの取得

FLX-PCI 2 User's Manual

fdmpcilsRgb()	: FDM-PCI RGB かどうかの取得
fdmpcilsRgb2()	: FDM-PCI RGB2 かどうかの取得
fdmpcilsTriggerOut()	: トリガ出力があるかどうかの取得
fdmpcilsTriggerIn()	: トリガ入力があるかどうかの取得
capSetVideoAmpHue()	: 選択中のチャンネルの HUE(色相)を設定
capGetVideoAmpHue()	: 選択中のチャンネルの HUE(色相)を取得
capSetVideoAmpSaturation()	: 選択中のチャンネルの Saturation(彩度)を指定
capGetVideoAmpSaturation()	: 選択中のチャンネルの Saturation(彩度)を取得
capGetIsVideoPresent()	: 選択中のチャンネルに信号が来ているかを取得

4.2. FLX-PCI API 関数リファレンス

int fdmpciInit(void)

- 機能
ボードの初期化を行う。
- 引数
なし
- 戻り値
FDMPCI_NOERROR : 正常終了
FDMPCI_ERROR : 異常終了

void fdmpciExit(void)

- 機能
ボードの後処理を行う。
ハングアップ防止のため必ず実行してください。
- 引数
なし
- 戻り値
なし

int fdmpciSetInputResolution(UINT nWidth, UINT nHeight)

- 機能
入力解像度の設定を行う。デフォルト解像度は 640 x 480 です。
最大解像度は 640 x 480 です。
- 引数
nWidth 入力解像度の幅
nHeight 入力解像度の高さ
- 戻り値
FDMPCI_NOERROR : 正常終了
FDMPCI_ERROR : 引数が範囲外

void fdmpciGetInputResolution(UINT *pWidth, UINT *pHeight)

- 機能
入力解像度の取得を行う。
- 引数
pWidth 入力解像度の幅を格納するためのポインタ
pHeight 入力解像度の高さを格納するためのポインタ
- 戻り値
なし

int fdmpciSetOutputResolution(UINT nWidth, UINT nHeight)

○機能

出力解像度の設定を行う。デフォルトは 640 x 480 です。

最大解像度は 640 x 480 です。

解像度の変更があった場合にはキャプチャは中止され、フレーム 0 番が選択されます。

○引数

nWidth 出力解像度の幅

nHeight 出力解像度の高さ

○戻り値

FDMPCI_NOERROR : 正常終了

FDMPCI_ERROR : 引数が範囲外

void fdmpciGetOutputResolution(UINT *pWidth, UINT *pHeight)

○機能

出力解像度の取得を行う。

○引数

pWidth 出力解像度の幅を格納するためのポインタ

pHeight 出力解像度の高さを格納するためのポインタ

○戻り値

なし

int fdmpciSetStartPosition(UINT x, UINT y)

○機能

開始点の設定を行う。デフォルトは(1, 16)です。
最大値は共に 0x3FFF です。

○引数

x 点の X 座標 (奇数のみ)
y 始点の Y 座標

○戻り値

FDMPCI_NOERROR : 正常終了
FDMPCI_ERROR : 引数が範囲外

注) x の値は必ず奇数に設定してください。
偶数に設定した場合には色が変化いたします。

void fdmpciGetStartPosition(UINT *x, UINT *y)

○機能

開始点の取得を行う。

○引数

x 始点の X 座標を格納するためのポインタ
y 点の Y 座標を格納するためのポインタ

○戻り値

なし

void fdmpciSelectChannel(int no)

- 機能
入力チャンネルの設定を行う。デフォルトは CH1 です。
- 引数
no チャンネル番号 (0: CH1)
- 戻り値
なし

int fdmpciGetChannelNum(void)

- 機能
入力チャンネルの取得を行う。
- 引数
なし
- 戻り値
チャンネル番号 (0:CH1)

int fdmpciGetNumChannel(void)

- 機能
入力チャンネル数の取得を行う。
- 引数
なし
- 戻り値
チャンネル数

BYTE *fdmpciSelectFrame(int no)

- 機能
カレントフレームを選択する。デフォルトはフレーム 0 です。
このコマンドを実行するとキャプチャは中止されます。
- 引数
no フレーム番号
- 戻り値
NULL 以外 : バッファのポインタ
NULL : 引数がフレーム数を越えた

int fdmpciGetFrameNum(void)

- 機能
カレントフレーム番号の取得を行う。
- 引数
なし
- 戻り値
カレントフレーム番号

int fdmpciGetNumFrame(void)

- 機能
フレーム数を取得する。
- 引数
なし
- 戻り値
フレーム数

BYTE *fdmpciGetBuffer(void)

- 機能
カレントフレームのバッファアドレスの取得を行う。データの並びにつきましては「**第5章 フレームメモリのデータ構造**」を参照してください。
- 引数
なし
- 戻り値
カレントフレームのバッファのアドレス

int fdmpciReadPixel(UINT x, UINT y, DWORD *pValue)

- 機能
カレントフレームの (x, y) 座標のピクセルの値を読み込む。
- 引数

x	X 座標
y	Y 座標
pValue	ピクセル値を格納するためのポインタ 「カラーモードの設定」に関わらず 32 ビットカラーにて読み込まれます。 X7X6X5X4X3X2X1X0 R7R6R5R4R3R2R1R0 G7G6G5G4G3G2G1G0 B7B6B5B4B3B2B1B0
- 戻り値

FDMPCI_NOERROR	: 正常終了
FDMPCI_ERROR	: 引数が範囲外

int fdmpciWritePixel(UINT x, UINT y, DWORD nValue)

○機能

カレントフレームの (x, y) 座標にピクセル値を書き込む。

○引数

x X 座標

y Y 座標

nValue ピクセル値

「カラーモードの設定」に関わらず 32 ビットカラーで設定します。

X7X6X5X4X3X2X1X0

R7R6R5R4R3R2R1R0

G7G6G5G4G3G2G1G0

B7B6B5B4B3B2B1B0

○戻り値

FDMPCI_NOERROR : 正常終了

FDMPCI_ERROR : 引数が範囲外

int fdmpciRectRead(UINT x1, UINT y1, UINT x2, UINT y2, DWORD *pBuff)

○機能

カレントフレームの四角形領域のピクセル値を読み込む。

○引数

x1 左

y1 上

x2 右

y2 下

pBuff ピクセル値を格納するためのポインタ

「カラーモードの設定」に関わらず 32 ビットカラーで読み込まれます。

X7X6X5X4X3X2X1X0

R7R6R5R4R3R2R1R0

G7G6G5G4G3G2G1G0

B7B6B5B4B3B2B1B0

○戻り値

FDMPCI_NOERROR : 正常終了

FDMPCI_ERROR : 引数が範囲外

int fdmpciRectWrite(UINT x1, UINT y1, UINT x2, UINT y2, DOWRD *pBuff)

○機能

カレントフレームの四角形領域にピクセル値を書き込む。

○引数

x1 左

y1 上

x2 右

y2 下

pBuff ピクセル値のポインタ

「カラーモードの設定」に関わらず 32 ビットカラーで設定します。

X7X6X5X4X3X2X1X0

R7R6R5R4R3R2R1R0

G7G6G5G4G3G2G1G0

B7B6B5B4B3B2B1B0

○戻り値

FDMPCI_NOERROR : 正常終了

FDMPCI_ERROR : 引数が範囲外

void fdmpciStopCapture(void)

- 機能 キャプチャの中止を行う。
- 引数 なし
- 戻り値 なし

void fdmpciSingleCapture(void)

- 機能 カレントフレームに一度だけキャプチャする。
- 引数 なし
- 戻り値 なし

void fdmpciCapture(void)

- 機能
カレントフレームにキャプチャし続けます。(ライブ)
- 引数
なし
- 戻り値
なし

int fdmpcilsCapture(void)

- 機能
キャプチャをしているかどうか調べる。(ライブ)
- 引数
なし
- 戻り値
 - 0 : ストップ
 - 1 : キャプチャ中 (ライブ)

int fdmpciSetBrightness(UINT value)

- 機能
輝度パラメータの設定を行う。デフォルトは0です。
value は-32 から 30 の間でなければならない。
- 引数
value 輝度パラメータ
- 戻り値
FDMPCI_NOERROR : 正常終了
FDMPCI_ERROR : 引数が範囲外

int fdmpciGetBrightness(void)

- 機能
輝度パラメータの取得を行う。
- 引数
なし
- 戻り値
輝度パラメータ

int fdmpciSetContrast(UINT value)

- 機能
コントラストパラメータの設定を行う。デフォルトは 128 です。
value は 0 から 255 の間でなければならない。
- 引数
value コントラストパラメータ
- 戻り値
FDMPCI_NOERROR : 正常終了
FDMPCI_ERROR : 引数が範囲外

int fdmpciGetContrast(void)

- 機能
コントラストパラメータの取得を行う。
- 引数
なし
- 戻り値
コントラストパラメータ値

void fdmpciSetColorMode(int mode)

○機能

フレームバッファのカラーモードの設定を行う。
デフォルトは 24 ビットカラーです。モードの変更があった
場合にはキャプチャは中止され、フレーム 0 が選択されます。

○引数

mode	0 : FDMPCI_RGB888	32 ビットカラー
	1 : FDMPCI_RGB555	16 ビットカラー
	2 : FDMPCI_RGB888_PACK	24 ビットカラー
	3 : FDMPCI_YUV422	
	4 : FDMPCI_RGB565	16 ビットカラー

○戻り値

なし

int fdmpciGetColorMode(void)

○機能

フレームバッファのカラーモードの取得を行う。

○引数

なし

○戻り値

0	: FDMPCI_RGB888	32 ビットカラー
1	: FDMPCI_RGB555	16 ビットカラー
2	: FDMPCI_RGB888_PACK	24 ビットカラー
3	: FDMPCI_YUV422	
4	: FDMPCI_RGB565	16 ビットカラー

int fdmpciSaveBitmap(char *fileName)

- 機能
カレントフレームをビットマップファイルに保存する。
- 引数
fileName ファイル名
- 戻り値
FDMPCI_NOERROR : 正常終了
FDMPCI_ERROR : 異常終了

int fdmpciLoadBitmap(char *fileName)

- 機能
ビットマップファイルをカレントフレームに読み込む。
ただしセーブした時の解像度と現在の解像度が違う場合にはロードできません。
- 引数
fileName ファイル名
- 戻り値
FDMPCI_NOERROR : 正常終了
FDMPCI_ERROR : 異常終了

void fdmpciCleanFrame

- 機能
カレントフレームの消去（黒）を行う。
- 引数
なし
- 戻り値
なし

void fdmpciCleanAll(void)

- 機能
すべてのフレームの消去（黒）を行う。
- 引数
なし
- 戻り値
なし

int fdmpciSaveAVIFile(char *fileName, int nFrom, int nTo)

- 機能
nFrom フレームから nTo フレームまでを fileName に AVI ファイル形式で保存する。
- 引数
fileName ファイル名
nFrom 開始フレーム番号
nTo 終了フレーム番号
- 戻り値
FDMPCI_NOERROR : 正常終了
FDMPCI_ERROR : 異常終了

int fdmpciSaveBitmaps(char *fileName, int nFrom, int nTo)

- 機能
nFrom フレームから nTo フレームまでを fileName で指定した BMP ファイル形式で保存する。fileName は printf の書式指定 (%を使用した)を用いる。
- 引数
fileName ファイル名 (printf 書式指定 例 “abcd%04d.bmp”)
nFrom 開始フレーム番号
nTo 終了フレーム番号
- 戻り値
FDMPCI_NOERROR : 正常終了
FDMPCI_ERROR : 異常終了

void fdmpciCaptureStart(void)

- 機能
シングルキャプチャを開始させる。fdmpciSingleCapture()のようにキャプチャ終了を待たない。またはキャプチャ終了は fdmpciIsCompleted()を使用する。
- 引数
なし
- 戻り値
なし

注) この関数はキャプチャの終了を待ちませんが、フレームの開始まで待機しますので1フレーム未満の時間がかかることになります。

int fdmpciIsCompleted(void)

- 機能
fdmpciCaptureStart()で開始したシングルキャプチャの終了確認を行う。
- 引数
なし
- 戻り値
0 : キャプチャ中
1 : キャプチャ終了

int fdmpciRecord(int nFrom, int nTo, nStep)

- 機能
範囲指定の録画を行う。
- 引数
 - nFrom 開始フレーム
 - nTo 終了フレーム
 - nStep 駒飛ばし数。駒飛ばし無しはゼロを指定する
- 戻り値
 - FDMPCI_NO_ERROR : 正常終了
 - FDMPCI_ERROR : 指定フレームが範囲外

int fdmpcils2ch(void)

- 機能
FDM-PCI 2CH ボードかどうかの取得。
- 引数
なし
- 戻り値
0 : FDM-PCI 2CH 以外のボード
1 : FDM-PCI 2CH ボード

int fdmpcilsRgb(void)

- 機能
FDM-PCI RGB ボードかどうかの取得。
- 引数
なし
- 戻り値
0 : FDM-PCI RGB 以外のボード
1 : FDM-PCI RGB ボード

int fdmpcilsRgb2(void)

- 機能
FDM-PCI RGB2 ボードかどうかの取得。
- 引数
なし
- 戻り値
0 : FDM-PCI RGB2 以外のボード
1 : FDM-PCI RGB2 ボード

int fdmpcilsTriggerOut(void)

- 機能
トリガ出力をサポートしているかどうか取得します。
- 引数
なし
- 戻り値
0 : サポートされていない
1 : サポートされている

int fdmpcilsTriggerIn(void)

- 機能
トリガ入力をサポートしているかどうか取得します。
- 引数
なし
- 戻り値
0 : サポートされていない
1 : サポートされている

BOOL capSetVideoAmpHue(long Value)

- 機能
現在のチャンネルの HUE(色相)を Value に設定する。
- 引数
-180,000 ~ 180,000 (-180 ~ +180 度)
- 戻り値
TRUE : 設定に成功
FALSE : 設定に失敗

BOOL capGetVideoAmpHue(long *Value)

- 機能
現在のチャンネルの HUE(色相)を *Value に得る。
- 引数
-180,000 ~ 180,000 (-180 ~ +180 度)
- 戻り値
TRUE : 取得に成功
FALSE : 取得に失敗

BOOL capSetVideoAmpSaturation(long Value)

- 機能
現在のチャンネルの Saturation(彩度)を Value に指定する。
- 引数
0 ~ 10000
- 戻り値
TRUE : 設定に成功
FALSE : 設定に失敗

BOOL capGetVideoAmpSaturation(long *Value)

- 機能
現在のチャンネルの HUE(彩度)を *Value に得る。
- 引数
0 ~ 10000
- 戻り値
TRUE : 取得に成功
FALSE : 取得に失敗

BOOL capGetIsVideoPresent(long *Present)

- 機能
現在選択されているチャンネルに信号が来ているかを取得する。
- 引数
信号が来ていれば *Present に TRUE が、来ていなければ FALSE が得られる。
- 戻り値
TRUE : 取得に成功
FALSE : 取得に失敗

4.3. フレームメモリのデータ構造

フレームメモリのデータ構造はフレームバッファのカラーモードによって異なります。

現在のフレームバッファのカラーモードは `fdmpciGetColorMode()` によって取得できます。

FDMPCI_RGB888(MODE=0)の時

アドレス	データ
+0x00	B7B6B5B4B3B2B1B0
+0x01	G7G6G5G4G3G2G1G0
+0x02	R7R6R5R4R3R2R1R0
+0x03	X7X6X5X4X3X2X1X0
+0x04	B7B6B5B4B3B2B1B0
+0x05	G7G6G5G4G3G2G1G0
+0x06	R7R6R5R4R3R2R1R0
+0x07	X7X6X5X4X3X2X1X0

注) X7-X0 は不定です。

FDMPCI_RGB555(MODE=1)の時

アドレス	データ
+0x00	G2G1G0B4B3B2B1B0
+0x01	X0R4R3R2R1R0G4G3
+0x02	G2G1G0B4B3B2B1B0
+0x03	X0R4R3R2R1R0G4G3
+0x04	G2G1G0B4B3B2B1B0
+0x05	X0R4R3R2R1R0G4G3

FDMPCI_RGB888_PACK(MODE=2)の時

アドレス	データ
+0x00	B7B6B5B4B3B2B1B0
+0x01	G7G6G5G4G3G2G1G0
+0x02	R7R6R5R4R3R2R1R0
+0x03	B7B6B5B4B3B2B1B0
+0x04	G7G6G5G4G3G2G1G0
+0x05	R7R6R5R4R3R2R1R0

Photron FLX-PCI 2 User's Manual

日本語版

2010年10月 Rev1.01

発行

株式会社フォトロン

〒102-0071 東京都千代田区富士見 1-1-8 千代田富士見ビル
TEL 03-3238-2107
FAX 03-3238-2107

本書の収録内容は、すべて株式会社フォトロンに著作権があります。無断転載・複製は固くお断りします。

製品の仕様やその他の記載内容については予告なく変更する場合がありますのであらかじめご了承ください。

- ※ 無断転載禁止
- ※ 乱丁・落丁はお取替えいたします。