

# USB 標準規格について ~ USB ビデオクラス ~

平成 20 年 2 月 5 日

## 1 はじめに

従来の画像処理のような外部のセンサを利用してデータを取得する場合には、外界センサであるカメラといった周辺機器を USB によって計算機である PC に接続してきた。しかし、これまで使用されてきたビデオカメラには IEEE 1394 を用いて接続する機種が多く、USB での接続を実現するにはメーカーがそれぞれ独自のドライバを用意する必要があるという問題があった。[1] そこで、このような問題を解決するために USB の標準化団体である "USB Implementers Forum" が USB カメラを PC に接続するための標準規格である USB ビデオクラスを策定した。本稿では、この USB 標準規格である USB ビデオクラスの概要と実際に USB ビデオクラスを導入してカメラを Linux で接続する方法について述べていく。

## 2 USB ビデオクラス 概要

USB ビデオクラスとは USB 標準化団体である "USB Implementers Forum" が 2003 年 9 月に公開した USB2.0 を拡張した家庭用ビデオカメラ、WebCam と呼ぶ映像を扱う機器である USB カメラを PC に USB で接続するための規格であり、Motion-JPEG、DV、MPEG-2 ないしは MPEG-4 などの圧縮データ、あるいは非圧縮のデータを、USB を通じて転送することが可能となる。前項でも述べたが、従来の USB カメラは IEEE 1394 で接続される機種が多かったため、USB 接続を実現するために機種毎のドライバを用意する必要がある。しかし USB ビデオクラスの策定によってビデオカメラ機器は 1 つのクラスとして定義され、ドライバとしては OS が提供する同一クラスのドライバを利用すれば良いことになり、これによって機器メーカーは必要に応じて機器個別のドライバを提供すればよく、機器の開発の手間は大幅に省かれることとなった。USB は現在販売されているほぼすべての PC が備えるインターフェースであるため、今後は IEEE 1394 対応の機器と並んで広く普及すると見込まれている。[1][2]

## 3 PC 環境

USB ビデオクラスを導入する前に本稿を記述するにあたって使用した PC 環境について述べる。以下、この PC 環境の仕様に基づいて述べる。

表 1: PC 環境

PC	東芝製ノート PC dynabook SS
OS	Linux Debian etch kernel-Ver 2.6.22-2-686
CPU	Intel Pentium M 1.0GHz
MEMORY	256 + 512 = 768MB
画像処理ライブラリ	Intel OpenCV1.0

## 4 USB ビデオクラスの導入

### 4.1 サポートされている USB カメラ

Linux 用の USB ビデオクラスのドライバを配布している”berliOS”によれば、2007 年現在にサポートしている USB カメラは図 1 の通りである.[3]

✔ Device works    ⚠ Device works with issues    ? Device is untested

Device ID	Name	Manufacturer	Status
041e:4057	Creative Live! Cam Optia	Creative Labs	✔
041e:4058	Creative Live! Cam Optia AF	Creative Labs	✔
045e:00f8	Microsoft LifeCam NX-6000	Microsoft	✔
046d:08c1	Logitech Quickcam Fusion	Logitech	⚠ [1,2]
046d:08c2	Logitech Quickcam Orbit MP	Logitech	⚠ [1,2]
046d:08c3	Logitech Quickcam Pro for Notebooks	Logitech	⚠ [1,2]
046d:08c5	Logitech Quickcam Pro 5000	Logitech	⚠ [1,2]
046d:08c6	Logitech Quickcam OEM Dell Notebook	Logitech	⚠ [1,2]
046d:08c7	Logitech Quickcam OEM Cisco VT Camera II	Logitech	⚠ [1,2]
046d:08c9	Logitech Quickcam Ultra Vision	Logitech	⚠ [1,2]
046d:08ca	Logitech Quickcam Fusion (2006 model)	Logitech	⚠ [1,2]
046d:08cb	Logitech Quickcam Pro for Notebooks (2006 model)	Logitech	⚠ [1,2]
046d:08cc	Logitech Quickcam Orbit MP (2006 model)	Logitech	⚠ [1,2]
046d:08ce	Logitech Quickcam Pro 5000 (2006 model)	Logitech	⚠ [1,2]
046d:0990	Logitech Quickcam Pro 9000	Logitech	✔
046d:0991	Logitech Quickcam Pro for Notebooks (2007 model)	Logitech	✔
046d:0992	Logitech Quickcam Communicate STX	Logitech	? Device is untested
046d:09b0	Acer Orbicam (found in notebooks)	Logitech	✔
046d:09b2	Fujitsu Webcam (found in notebooks)	Logitech	? Device is untested
046d:09c0	Quickcam for Dell Notebooks (found in Dell notebooks)	Logitech	⚠ [1,2]
046d:09c1	Logitech Quickcam Deluxe for Notebooks	Logitech	⚠ [1,2]
04cb:0172	Fujifilm FinePix E550	Fujifilm	✔
04da:2318	Panasonic Camcorder NV-GS11	Panasonic	✔
04da:231d	Panasonic Camcorder NV-GS27EG	Panasonic	✔
04f2:b012	Chicony 1.3M UVC Webcam (Asus G1S notebooks)	Chicony Electronics	✔
04f2:b015	Chicony VGA UVC Webcam (HP notebooks)	Chicony Electronics	✔
04f2:b018	Chicony 2M UVC Webcam (Compal notebooks)	Chicony Electronics	✔
05a9:2640	Omnivision OV2640 (Dell Inspiron 1420/1720 notebooks)	Omnivision	✔
05c8:0103	Omnivision OV2640 (Dell Inspiron 1420/1720 notebooks)	Omnivision	✔
064e:a100	F013FF-65 PC-CAM	Foxlink	✔
090c:b371	Silicon Motion SM371	Silicon Motion	✔
0ac8:c302	Vega USB 2.0 Camera (Samsung Q45 notebook)	Vimicro	✔
0c45:62c0	Sonix USB 2.0 Camera (Acer Aspire 5050 and HP Pavilion DV6000 notebooks)	Sonix Technology	✔
18cd:cafe	Pico iImage	Ecamm	✔
19ab:1000	Bodelin ProScope HR	Bodelin	✔
5986:0100	Acer Orbicam (found in notebooks)	Unknown	✔
5986:0200	Acer Orbicam (found in notebooks)	Unknown	✔

図 1: サポートされている USB カメラ一覧

## 4.2 使用した USB カメラ

本稿を記述するにあたっては図 2 に挙げられている”logitech Quickcam Pro for Notebook”を使用した。



図 2: logitech Quickcam Pro for Notebook (amazon より引用)

## 4.3 USB ビデオクラスドライバの導入のための諸準備

この項からは、実際に USB ビデオクラスドライバの導入について述べていく。まず、USB ビデオクラスを配布している”berlios Developer” [4] からのドライバの取得には”svn”コマンドを使用するため、apt-get コマンドを用いて”subversion”を PC にインストールする

```
apt-get install subvesion
```

subversion のインストールが完了したら、次のコマンドを用いてドライバをダウンロードする。

```
svn checkout svn://svn.berlios.de/linux-uvc/linux-uvc/trunk
```

するとホームディレクトリの下のユーザーディレクトリ直下に trunk というディレクトリが生成される。trunk ディレクトリ内に移動すると Makefile が存在するので、後は”make”コマンドよりディレクトリ内のファイルをコンパイルするだけであるが、本稿ではこのコンパイル～ドライバのインストールを自動および手動で行う二種類の方法について述べる。

## 4.4 自動インストールの場合

trunk ディレクトリ内のファイルをコンパイルするには、現在使用しているカーネルのバージョンのヘッダファイルが必要であるため”apt-get”コマンドを用いてヘッダファイルの取得を行うが、ここでヘッダファイルの取得に用いる”apt-get”のサーバーの参照先を確認する。

”apt-get”の参照先を記述しているファイル”/etc/apt/sources.list”をエディタで開く。すると次のような画面が立ち上がるので、参照先に Debian の大元であるプライマリを参照するように次の一行を加える。

```
deb ftp://ftp2.jp.debian.org/debian/ testing main contrib non-free
```

```
# deb cdrom:[Debian GNU/Linux 4.0 r0 _Etch_ - Official i386 NETINST Binary-1 2007-04-07-11:29]/ etch contrib main
# deb cdrom:[Debian GNU/Linux 4.0 r0 _Etch_ - Official i386 NETINST Binary-1 2007-04-07-11:29]/ etch contrib main
#deb http://ring.asahi-net.or.jp/archives/linux/debian/debian/ etch main
#deb-src http://ring.asahi-net.or.jp/archives/linux/debian/debian/ etch main
#deb http://security.debian.org/ etch/updates main contrib
#deb-src http://security.debian.org/ etch/updates main contrib
#deb http://ring.asahi-net.or.jp/pub/linux/debian/debian etch main contrib non-free
#deb http://security.debian.org/ etch/updates main contrib non-free
#deb http://ring.asahi-net.or.jp/pub/linux/debian/debian/ testing main contrib non-free
#deb http://ftp2.jp.debian.org/debian/ testing main contrib non-free
#deb http://security.debian.org/ testing/updates main contrib non-free
```

図 3: apt-get の設定ファイル sources.list

設定が完了したら一度, "apt-get" コマンドでアップデートしておく.

apt-get update

アップデート終了後, 次のコマンドを入力し, カーネルのヘッダファイルをインストールする.

apt-get install linux-headers-2.6-686

ここで linux-headers-以降のバージョン数は使用しているカーネルにより異なるので, "uname -a" で確認し環境に応じて変更すること. 上記のコマンドにより, ヘッダファイルのインストールが完了すると図のように "/usr/src/" に使用しているカーネルのバージョンのヘッダファイルが生成される.



```
b0104399@debian: ~/work/rtex
b0104399@debian:~/work/rtex$ ls /usr/src/
linux-headers-2.6.21-2      linux-kbuild-2.6.22
linux-headers-2.6.21-2-686  linux-source-2.6.22
linux-headers-2.6.22-2      linux-source-2.6.22.tar.bz2
linux-headers-2.6.22-2-686  rpm
linux-kbuild-2.6.21
b0104399@debian:~/work/rtex$
```

図 4: 生成された linux-headers

ヘッダファイルがインストールされているのを確認したら, trunk ディレクトリ内に移動し "make" コマンドによりディレクトリ内のファイルのコンパイルを行う. コンパイルが終了すると, 図のように trunk ディレクトリ内に "uvcvideo.ko" というファイルが生成される.



```
b0104399@debian: ~/trunk
b0104399@debian:~/trunk$ ls
Makefile      uvc_compat.h  uvc_driver.o  uvc_v4l2.o  uvcvideo.ko
Module.symvers  uvc_ctrl.c   uvc_queue.c   uvc_video.c  uvcvideo.mod.c
dyncntrl.txt   uvc_ctrl.o   uvc_queue.o   uvc_video.o  uvcvideo.mod.o
log           uvc_driver.c  uvc_v4l2.c   uvcvideo.h   uvcvideo.o
b0104399@debian:~/trunk$
```

図 5: コンパイル後の trunk ディレクトリ

最後に "make install" と入力すると, 自動で USB ビデオクラスのドライバがインストールされる. 以上の操作で PC に USB ビデオクラスがインストールされ, 図 1 に挙げられている USB ビデオクラスに準拠する USB カメラを USB 端子に接続するだけで使用可能となる.

## 4.5 手動インストールの場合

大抵の場合は前項で述べた自動インストールで問題無いが、上手くいかない場合はこの項で述べる手動によるインストールを行う。まず、trunk ディレクトリの中身をコンパイルするためにはカーネルのソースファイルが必要なので、”synaptic”あるいは”aptitude”を用いてカーネルのソースファイルをダウンロードする。今回用いた PC のカーネルのバージョンは”2.6.22-2-686”なので”linux-source2.6.22”をダウンロードする。

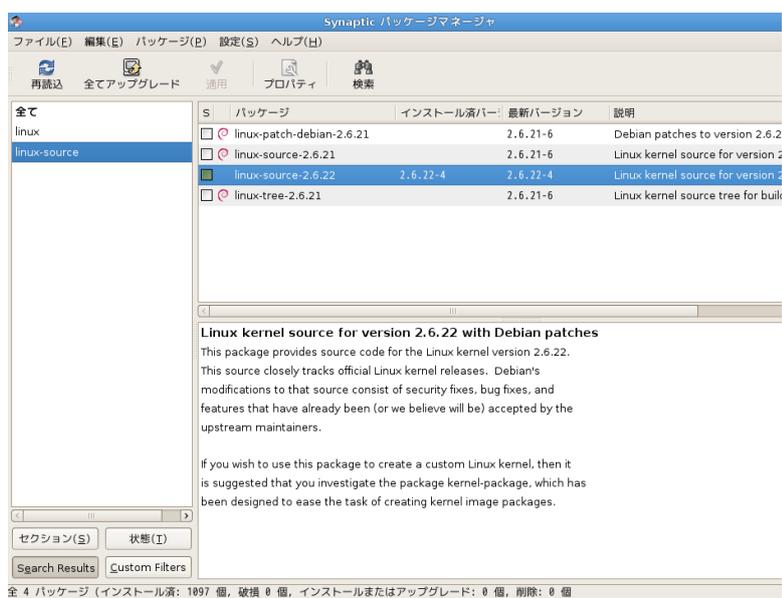


図 6: linux-source2.6.22 のダウンロード

ダウンロードが完了すると”/usr/src/”に”linux-source-2.6.22.tar.bz2”が生成されているので、tar コマンドを用いてファイルの解凍を行う。

```
tar xjf linux-source-2.6.22.tar.bz2
```

ファイルの解凍が終了したら、解凍されて生成されたソースのディレクトリ”/usr/src/linux-source-2.6.22/”に移動しドライバのコンパイルを行うための設定を行う。

```
make oldconfig
make prepare
make modules __prepare
```

上記のコマンドにより、ヘッダの更新やシンボリックリンクの解決、Makefile の生成などドライバのコンパイルに必要な設定や準備が行われる。

以上の準備が完了したら、trunk ディレクトリに移動しディレクトリ内部のファイルのコンパイルを行うが、コンパイル時のリンク先をダウンロードしたカーネルソースを参照するようにしなければならぬため、Makefile の 2 行目を次のように書換えた。

```
KERNEL __DIR := /usr/src/linux-source-2.6.22
```

```

KERNEL_VERSION := `uname -r`
#KERNEL_DIR := /lib/modules/$(KERNEL_VERSION)/build
KERNEL_DIR := /usr/src/linux-source-2.6.22
INSTALL_MOD_DIR := usb/media

PWD := $(shell pwd)

obj-m := uvcvideo.o
uvcvideo-objs := uvc_driver.o uvc_queue.o uvc_v4l2.o uvc_video.o uvc_ctrl.o

all: uvcvideo

uvcvideo:
@echo "Building USB Video Class driver..."
@cd $(KERNEL_DIR) && make -C $(KERNEL_DIR) SUBDIRS=$(PWD) CROSS_COMPILE=$(CROSS_COMPILE) modules

install:
@echo "Installing USB Video Class driver..."
@cd $(KERNEL_DIR) && make -C $(KERNEL_DIR) SUBDIRS=$(PWD) INSTALL_MOD_DIR=$(INSTALL_MOD_DIR) INSTALL_MOD_PATH=$(INSTALL_MOD_PATH) modules_install
depmod -ae

clean:
-rm -f *.o *.ko *.cmd *.flags *.mod.c Modules.symvers
-rm -rf *.tmp_versions

```

図 7: trunk 内の Makefile の書き換え

trunk ディレクトリ内で”make”によりコンパイルを行うと、図 4 のように”uvcvideo.ko”というカーネルオブジェクトが生成される。次に手でインストールを行うための準備をする。まず、”/lib/modules/カーネルバージョン/kernel/drivers/USB/”に”media”というディレクトリが存在しているか調べる。存在していないようなら、”mkdir”コマンドによりディレクトリを生成する。次に、コンパイルによって trunk ディレクトリに生成された”uvcvideo.ko”を”cp”コマンドにより”/lib/modules/カーネルバージョン/kernel/drivers/USB/media/”にコピーする。その後、”depmod”コマンドにより USB デバイスを自動で認識するモジュールの設定を行う。

```
depmod -ae /lib/modules/カーネルバージョン/kernel/drivers/USB/media/uvcvideo.ko
```

以上の操作で PC に USB ビデオクラスがインストールされ、図 1 に挙げられている USB ビデオクラスに準拠する USB カメラを USB 端子に接続するだけで使用可能となる。

## 5 OpenCV による画像処理

USB ビデオクラスを導入した PC において、実際に USB カメラが使用可能かどうか、またその USB カメラを用いて intel 社のオープンソース画像処理ライブラリである OpenCV による画像処理が可能かどうかを調べた。その結果、USB カメラを問題無く認識し、私が OpenCV を利用して作成したプログラムを走らせることができた。



図 8: Qcam Pro for Notebook を用いた OpenCV による画像処理

## 6 まとめ

今回は USB の標準規格である USB ビデオクラスの概要および導入について述べた。USB の規格の統一により煩わしい USB カメラ毎のドライバの設定が不要になるだけでなく、本規格が導入されていれば OS を選ばずに USB カメラが使用可能であるために画像処理の研究において非常に有用であると言える。

## 7 参考文献

### 参考文献

- [1] IT 用語辞典 BINARY (USB Video Device Class) <http://www.sophia-it.com/content/USB>
- [2] アスキーデジタル用語辞典 (USB ビデオクラス) <http://yougo.ascii24.com/gh/84/008473.html>
- [3] berliOS (Linux UVC driver and tools) <http://linux-uvc.berlios.de/>
- [4] berliOS developer <http://developer.berlios.de/projects/linux-uvc>

## 8 謝辞

本稿の作成にあたり、橋本研究室ドクターコースの佐々木さんに多大なる助力を頂きました。本稿がこうして無事作成できたのも一重に佐々木さんの助力があったからこそです。ここに記して感謝の意を表したいと思います。また、今回本稿を作成する機会と助言を与えてくださった橋本先生、並びに大阪工業大学の小林さんに感謝します。本当にありがとうございました。