USB-CAMAC デバイスドライバ外部仕様

- User's Guide of USB-CAMAC Library for Linux -

仲吉一男

kazuo.nakayoshi@kek.jp

高エネルギー加速器研究機構
素粒子原子核研究所オンライングループ

目 次

1 はじめに **1**

2 システム概要 **1**
2.1 ハードウエア構成 1
2.2 ソフトウエア構成 2

3 インストレーション **2**
3.1 ハードウエアのインストール 2
3.2 ソフトウエアのインストール 2

4 Linux USB-CAMAC ライブラリ **3**
4.1 CAMAC セットアップファンクション 4
4.2 CAMAC シングルアクション (コントロール) 4
4.3 CAMAC シングルアクション (データ転送) 5
4.4 CAMAC ブロック転送 6

5 USB-CAMAC ライブラリの使用方法 **7**
5.1 ドライバのロード 7
5.2 CAMAC シングルアクション 7
5.3 CAMAC ブロック転送 8

6 パフォーマンス **10**

参考文献 **10**

1 はじめに

SCSI CAMAC クレートコントローラと USB-SCSI 変換ケーブルを使用して Linux をインストールした PC から USB 接続で CAMAC 操作を行なう方法について説明する. 小規模な原子核・素粒子実験のデータ収集 (DAQ) システムでは, 計算機とバスアダプタを使用して CAMAC や VME とデータ転送を行なうスタイルが一般的である. 従って DAQ で使用可能な計算機はおのずと限定 (専用カードをインストール済みか, もしくは新たにインストールする必要があるという意味で) されていた. しかし, PC とその周辺機器とのインターフェイスとしてデファクトスタンダードな USB を利用することで専用カードを必要とせず USB ポートを持つ PC から DAQ が可能となる. より具体的なイメージとしては, 本システムのソフトウエア (USB-CAMAC ドライバおよび CAMAC ライブラリ) をインストールすれば買ったばかりのノート PC と後述の SCSI CAMAC クレートコントローラを USB-SCSI 変換ケーブルで接続すれば CAMAC のデータ収集が可能となるのである.

この外部仕様 (User's Guide) では 2 節でハードウエアとソフトウエアの構成について説明し, 3 節ではハードウエアとソフトウエアのインストールの方法について述べる. 4 節では USB-CAMAC デバイスドライバのユーザインタフェイスである USB-CAMAC ライブラリ関数について説明し, 5 節では CAMAC シングルアクションやブロック転送についてサンプルプログラムを用いて説明する. 6 節では USB-CAMAC デバイスドライバの性能について述べる.

2 システム概要

2.1 ハードウエア構成

現時点で USB 対応の CAMAC クレートコントローラは存在しないので, SCSI CAMAC クレートコントローラを USB-SCSI 変換ケーブルを介して PC と USB 接続して使用することにした. 具体的には次のハードウエアから構成される.

- USB インターフェイス付の PC
- KineticSystems 社 SCSI CAMAC Crate Controller 3929[1]
- Logitec 社 USB-SCSI 変換ケーブル LUB-SC[2]
- SCSI 変換アダプタ (ピンハーフピッチ 50pin メス– アンフェノール 50pin オス)[3]

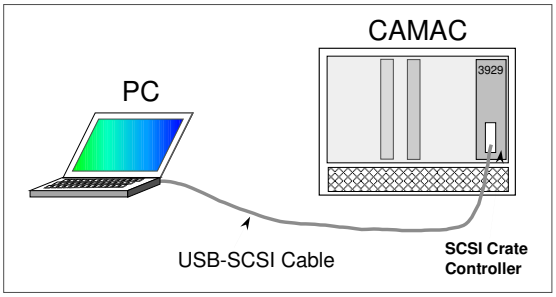

図 1: USB-CAMAC DAQ システムの概念図. 図の左から USB ポートを持つ PC, USB と SCSI を接続するための USB-SCSI 変換ケーブル. CAMAC クレートの右端には SCSI CAMAC クレートコントローラがある. USB-SCSI 変換ケーブルとクレートコントローラは SCSI 変換アダプタを介して接続する.

2.2 ソフトウエア構成

本システムは次のソフトウエアにより構成される.

- Linux カーネル 2.4.0 以上
- USB-CAMAC 用ドライバ
- USB-CAMAC 用 CAMAC ライブラリ

Linux カーネル は USB を正式にサポートしているバージョン 2.4.0 以上が必要である. Linux カーネルの中には USB Core ドライバおよび USB ホストコントローラ用ドライバが含まれる [4]. さらに今回開発した USB-CAMAC 用ドライバとそのユーザ・インタフェイスである USB-CAMAC 用 CAMAC ライブラリにより構成される.

3 インストレーション

3.1 ハードウエアのインストール

1. CAMAC クレートの右端に 3929 をセットする.
2. USB-SCSI 変換ケーブルの SCSI 側のコネクタ (50 pin オス) と SCSI 変換アダプタ (50 pin メス) を接続し SCSI 変換アダプタ (アンフェノール・オス) と 3929 を接続する.
3. USB-SCSI 変換ケーブルの USB 側のコネクタを PC の USB ポートへ接続する.

3.2 ソフトウエアのインストール

1. http://onlax2.kek.jp/~nakayosi/USB/usb-camac.html から USB-CAMAC パッケージをダウンロードする.

2. パッケージを展開する. (xxx はバージョン番号が入ります)
   % tar -xvfz usb-camac-xxx.tgz
   展開後, drv, lib, example というディレクトリができていることを確認する.
3. drv ディレクトリに移る. スーパーユーザになり USB-CAMAC ドライバーを make する.
   % cd drv
   % su
   # make
4. lib ディレクトリに移り USB-CAMAC ライブラリを make する.
   % cd lib
   % make

4 Linux USB-CAMAC ライブラリ

Linux USB-CAMAC ライブラリの関数は大きく次の 4 つに分けられる. CAMAC デバイスの open や close 等を行なうためのセットアップ関数, データを伴わない CAMAC 操作関数, データ転送を伴う CAMAC 操作関数および CAMAC ブロック転送関数である. これらの関数は, KEK オンライングループによりこれまで使用されている関数 [5] と同じ形式とした. 従ってユーザは, かつて DECstation や SPARCstation で使用した CAMAC ライブラリとほぼ同じ形式で USB から CAMAC へアクセスすることができる. 表 1 にその一覧を示す.

function	Description
camopen	Open CAMAC device
camcls	Close CAMAC device
naf	Set CAMAC NAF Code
cgenz	Generate CAMAC Initialize
cgenc	Generate CAMAC Clear
cseti	Set CAMAC Inhibit
cremi	Remove CAMAC Inhibit
camac	Execute CAMAC Single Action(24bit)
camacw	Execute CAMAC Single Action(16bit)
cdma	Execute CAMAC Block Transfer(24bit)
cdmaw	Execute CAMAC Block Transfer(16bit)

表 1: USB-CAMAC ライブラリで用意されている CAMAC 関数一覧

4.1 CAMAC セットアップファンクション

- camopen
  - Syntax: int camopen()
  - Description: CAMAC をオープンする．CAMAC にアクセスする前に 1 度呼ばなければならない．
  - Arguments: なし
  - Return value: 成功すれば 0, 失敗すると 0 以外の負の値が返る.
- camcls
  - Syntax: int camcls()
  - Description: CAMAC をクローズする．プログラム終了時に 1 度呼ばなければならない．
  - Arguments: なし
  - Return value: 成功すれば 0, 失敗すると 0 以外の負の値が返る.
- naf
  - Syntax: int naf(int n, int a, int f)
  - Description: CAMAC の N, A, F をデバイスドライバで使用する NAF コードに変換する．
  - Arguments:
    - n: ステーション番号
    - a: サブアドレス
    - f: ファンクション
    - Return value: NAF コード

4.2 CAMAC シングルアクション (コントロール)

- cgenz
  - Syntax: int cgenz()
  - Description: CAMAC を初期化する．
  - Arguments: なし
  - Return value: 成功すれば 0, 失敗すると 0 以外の負の値が返る.

- cgenc
  - Syntax: int cgenc()
  - Description: CAMAC をクリアする．
  - Arguments: なし
  - Return value: 成功すれば 0, 失敗すると 0 以外の負の値が返る.
- cseti
  - Syntax: int cseti()
  - Description: CAMAC の inhibit をセットする．
  - Arguments: なし
  - Return value: 成功すれば 0, 失敗すると 0 以外の負の値が返る.
- cremi
  - Syntax: int cremi()
  - Description: CAMAC の inhibit をクリアする．
  - Arguments: なし
  - Return value: 成功すれば 0, 失敗すると 0 以外の負の値が返る.

4.3 CAMAC シングルアクション (データ転送)

- camac
  - Syntax: int camac(int naf, int *dat, int *q, int *x)
  - Description: NAF コードで指定された CAMAC 24bit シングルアクション を実行し，24bit データの転送を行なう．データは dat で指定されるメモリ領域に read/write される．
  - Arguments:
    - naf: NAF コード
    - dat: データ
    - q: CAMAC Q Response
    - x: CAMAC X Response
  - Return value: 成功すれば 0, 失敗すると 0 以外の負の値が返る.
- camacw

- Syntax: int camacw(int naf, int *dat, int *q, int *x)
- Description: NAF コードで指定された CAMAC 24bit シングルアクション を実行し，16bit データの転送を行なう．データは dat で指定されるメモリ領域に read/write される．
- Arguments:
  * naf: NAF コード
  * dat: データ
  * q: CAMAC Q Response
  * x: CAMAC X Response
- Return value: 成功すれば 0, 失敗すると 0 以外の負の値が返る.

4.4 CAMAC ブロック転送

- cdma
  - Syntax: int cdma(int mode, int naf, int *buf, int len, int *retlen)
  - Description: mode, naf で指定された CAMAC 24bit ブロック転送を行なう．データは buf で指定されるメモリ領域に read/write される．
  - Arguments:
    * mode: 次の中から一つを指定する．QSTOP，QIGNORE，QREPEAT，QSCAN
    * naf: NAF コード
    * buf: data buffer
    * len: data buffer length(Byte)
    * retlen: return data buffer length
  - Return value: 成功すれば 0, 失敗すると 0 以外の負の値が返る.
- cdmaw
  - Syntax: int cdmaw(int mode, int naf, int *buf, int len, int *retlen)
  - Description: mode, naf で指定された CAMAC 16bit ブロック転送を行なう．データは buf で指定されるメモリ領域に read/write される．
  - Arguments:
    * mode: 次の中から一つを指定する．QSTOP，QIGNORE，QREPEAT，QSCAN

    * naf: NAF コード
    * buf: data buffer
    * len: data buffer length(Byte)
    * retlen: return data buffer length
  - Return value: 成功すれば 0, 失敗すると 0 以外の負の値が返る.

5 USB-CAMAC ライブラリの使用方法

以下のような手順で CAMAC モジュールからデータを読んでみる. できればスイッチレジスタのようなテスト専用モジュールを使うことをお勧めします. (以下ではスイッチレジスタのデータを読むことを想定している)

5.1 ドライバのロード

1. ルートになり USB ホストドライバ (usb-uhci.o) をカーネルに組み込む.
   # insmod usb-uhci.o
2. USB-CAMAC ドライバ (xcamac.o) をカーネルに組み込む.
   # insmod xcamac.o
3. 必要なドライバがロードされているか lsmod コマンドで確認する. 以下のように usb-uhci, xcamac が表示されれば正常にロードされている.

```
# lsmod
Module                  Size  Used by
usb-uhci               22196   0  (unused)
xcamac                  7020   0
```

5.2 CAMAC シングルアクション

1. 次のプログラムをファイルに single.c という名前で保存する. ここでは CAMAC ステーション#2 のスイッチレジスタの値 (16bit) をシングルアクションで読むことを想定しています. 別のモジュールにアクセスする場合は NAF の値を変更してください. またスイッチレジスタの値は 0x1111 に設定していると仮定します.

```
#include "camlib.h"

   int main()
   {
          int naf1, data, q, x;
          naf1 = naf(2,1,0);

          camopen();
          cgenz();
          cgenc();
          cremi();
          camacw(naf1, &data, &q, &x);
          camcls();
          printf("data = %d (0x%x)\n", data, data);
          printf("q = %d  x = %d\n", q, x);
   }
```

2. プログラムのコンパイルとリンクをする.
   % gcc -o single single.c camlib.o
3. single を実行する.
   % ./single
4. 正常に終了した場合, 次のような出力結果となる.

```
data = 4369 (0x1111)
q = 1  x = 1
```

5.3 CAMAC ブロック転送

1. 次のプログラムをファイルに block.c という名前で保存する. ここではCAMACステーション#2 のスイッチレジスタの値 (16bit) を Q-ignore モードのブロック転送で読むことを想定しています. 別のモジュールにアクセスする場合は NAF の値を変更してください. またスイッチレジスタの値は0x1111 に設定していると仮定します.

```
#include "camlib.h"

int main()
{
        int i, naf1, len = 10, rlen;
        unsigned short data[10];
        naf1 = naf(2,1,0);

        camopen();
        cgenz();
        cgenc();
        cremi();
        cdmaw(IGNORE, naf1, data, len, &rlen);
        camcls();

        for(i = 0; i < len; i++)
           printf("data[%d] = %hu(0x%x)\n", i, data[i], data[i]);
        printf("rlen = %d\n", rlen);
 }
```

2. プログラムのコンパイルとリンクをする.
   % gcc -o block block.c camlib.o
3. block を実行する.
   % ./block
4. 正常に終了した場合, 次のような出力結果となる.

```
data[0] = 4369(0x1111)
data[1] = 4369(0x1111)
data[2] = 4369(0x1111)
data[3] = 4369(0x1111)
data[4] = 4369(0x1111)
data[5] = 4369(0x1111)
data[6] = 4369(0x1111)
data[7] = 4369(0x1111)
data[8] = 4369(0x1111)
data[9] = 4369(0x1111)
rlen = 20
```

6 パフォーマンス

本ライブラリを使用して CAMAC シングルアクションとブロック転送のデータ転送速度を測定した. その結果を次の表に示す. 測定には Pentium II 300MHz, メモリ 190MB の PC を使用した.

	24bit data	16bit data
READ(ms)	4.9	4.9
WRITE(ms)	4.9	4.9

表 2: CAMAC シングルアクション性能

	16bit data
READ(KB/s)	919
WRITE(KB/s)	923
Overhead(ms)	4.9

表 3: CAMAC ブロック転送性能

参考文献

[1] Model 3929-Z1A SCSI Crate Controller Instruction Manual, KineticSystems Corporation.

[2] ロジテック USB-SCSI 変換ケーブル LUB-SC.
http://www.logitec.co.jp/

[3] 例えばエレコム SCSI 変換 アダプタ AD-PFAM.
http://www.elecom.co.jp/

[4] Programming Guide for Linux USB Device Drivers,
http://usb.in.tum.de/usbdoc/.

[5] Yoshiji Yasu, Usage Guide of CAMAC Library for UNIX(Version 0.8), 1993.