

# USB-CAMAC 変換アダプタによる ポータブル CAMAC データ収集

仲吉一男      安 芳次      高梨充朗<sup>A</sup>      関根英樹<sup>A</sup>

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

<sup>A</sup> 株式会社東陽テクニカ

## 概要

東陽テクニカ社製 CC/USB は同社製の CAMAC クレートコントローラ CC/7700 に取り付け、PC から USB を介して CAMAC の制御を可能にする USB-CAMAC 変換アダプタである。CC/USB は KEK オンライングループが提案し東陽テクニカ社と協議を行って仕様を決定し、同社が製作した。現在、USB はデスクトップ型の PC のみならずノート型の PC にもほぼ標準的に採用されており、このアダプタを用いることで簡便でポータブルな CAMAC データ収集 (DAQ) システムの構築が可能になる。このような DAQ システムの構築のため Linux で利用できる CC/USB のデバイスドライバおよび CAMAC ライブラリの開発を行った。また、このデバイスドライバを使用して CC/USB による CAMAC データ転送速度を測定した。その結果、CAMAC シングル転送の実行時間が平均 1.5msec、CAMAC ブロック転送のスピードが、約 420KBytes/sec(16bit Q-Ignore モード・リード) であった。Linux カーネル 2.4.x に対応したデバイスドライバおよび CAMAC ライブラリを配付キットとしてまとめ WEB 上で公開している。

# 目次

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	背景と目的	1
1.2	CC/USBに要求した仕様	1
1.3	CC/USBでサポートするCAMAC動作	2
<b>2</b>	<b>ハードウェアおよびソフトウェアの構成</b>	<b>3</b>
2.1	ハードウェアの構成	3
2.2	ソフトウェアの構成	3
<b>3</b>	<b>ハードウェア仕様</b>	<b>4</b>
<b>4</b>	<b>CC/USBの転送プロトコルの仕様</b>	<b>4</b>
4.1	CC/USBで用いる転送プロトコル仕様の方針	5
4.2	CC/USBのエンドポイント構成	5
4.3	データフォーマット	6
4.3.1	OUT1 エンドポイント	6
4.3.2	OUT4 エンドポイント	7
4.3.3	IN1 エンドポイント	7
4.3.4	IN4 エンドポイント	7
4.4	データ転送シーケンス	7
4.4.1	CAMAC シングル転送	7
4.4.2	CAMAC ブロック転送	8
4.4.3	CAMAC LAM ステータス	9
4.4.4	CAMAC 制御コマンド	9
<b>5</b>	<b>CC/USBのLinux用デバイスドライバ</b>	<b>9</b>
5.1	サポートしているCC/USBの制御コマンド	9
5.2	サポートするファイル操作関数	10
5.3	CAMAC マルチクレートのサポート	10
<b>6</b>	<b>CC/USBの性能</b>	<b>11</b>
6.1	CAMAC シングル転送性能	11
6.2	CAMAC ブロック転送性能	12
<b>7</b>	<b>まとめ</b>	<b>13</b>
	謝辞	13
	参考文献	13

<b>A</b>	<b>CC/USB の各エンドポイントで用いるデータフォーマット</b>	<b>14</b>
A.1	OUT1 エンドポイント . . . . .	14
A.2	OUT4 エンドポイント . . . . .	16
A.3	IN1 エンドポイント . . . . .	17
A.4	IN4 エンドポイント . . . . .	19
<b>B</b>	<b>CC/USB セットアップ方法</b>	<b>20</b>
B.1	ハードウェアのセットアップ . . . . .	20
B.2	ソフトウェアのセットアップ . . . . .	20
<b>C</b>	<b>Linux 用 CAMAC ライブラリ</b>	<b>21</b>
C.1	CAMAC ライブラリを使ったプログラム例 . . . . .	21

# 1 Introduction

## 1.1 背景と目的

KEK オンラインおよび回路グループではシリアル転送を用いた次世代データ収集 (DAQ) システムの研究・開発を行っている。我々の注目しているシリアル転送技術を表 1 にまとめた。筆者らは、その中から USB(Universal Serial Bus) [1] に注目し、これまで調査・研究を行ってきた。USB はコンピュータとその周辺機器を接続するためのシリアル転送技術で、活線挿抜等ユーザの使い易さが考慮されている。近年 USB はデスクトップ型 PC/AT 互換機 (PC) のみならずノート型 PC にもほぼ標準的に採用されており、これを利用すればポータブルな DAQ システムの構築が可能になると考えた。そこで KineticSystems 社 SCSI クレートコントローラ Model 3929-Z1A(3929) [2] と市販の USB-SCSI 変換ケーブルを用いて USB から CAMAC [3] の制御を行った [4]。これでノート PC を使用したポータブルな CAMAC DAQ システムが構築可能であるが 3929 は高価であり、新規にシステムを構築する利用者にとっては現実的でないと考えた。そこで KEK 周辺で比較的多く利用されている東陽テクニカ社の CAMAC クレートコントローラ TOYO CC/7700 [5](CC/7700) に対して同様の USB 変換ケーブルが実現できないか同社へ提案した。その後、東陽テクニカ社と協議を行って仕様を決定し同社が USB アダプタ TOYO CC/USB(CC/USB) を製作した。

	最大転送速度 (Mbps)	接続装置および応用例
USB1.x	12	キーボード, マウス等
USB2.0	480	ハードディスク, マルチメディア等
IEEE1394.a	400	デジタルビデオカメラ, ハードディスク等
IEEE1355.2	400	マルチメディア・ネットワーク等
Fast Ethernet	100	ネットワーク機器
Gigabit Ethernet	1000	ネットワーク機器

表 1: 我々が注目しているシリアルリンク技術。またその最大転送速度と応用例。

CC/USB 開発の目的は「USB を用いたポータブルな CAMAC DAQ システムの実現」である。ポータブル CAMAC DAQ システムとは、例えばノート PC を用いた可搬な DAQ システムであり、また活線挿抜のような USB の特長を利用した柔軟なシステムのことである。これは CAMAC モジュールのチェックやテスト実験等での使用を想定している。

## 1.2 CC/USB に要求した仕様

KEK オンライングループでは CC/USB の仕様として次のことを要求した。

1. CC/USB の形状は極力小さくし、CC/7700 に直接取り付け可能であること。
2. CC/USB は外部電源を必要としないバスパワー・デバイスとして動作すること。
3. CAMAC ブロック転送機能をサポートすること。
4. CC/USB で使用する転送プロトコルは、CC/7700 に依存しないこと。

USBを用いたポータブルな DAQ システムの構築には、上記1と2の条件は特に重要である。

### 1.3 CC/USB でサポートする CAMAC 動作

CC/USB では表2の CAMAC 動作をサポートしている。

CAMAC シングル転送	ステータス付き
	ステータス無し
CAMAC ブロック転送	Q-Ignore モード
	Q-Repeat モード
	Q-Stop モード
	Address Scan モード
CAMAC 制御	CAMAC Z
	CAMAC C
	Set Inhibit
	Remove Inhibit
	Read LAM Pattern
	Read Status

表 2: CC/USB でサポートしている CAMAC 動作.

ステータス付きシングル転送でリードを行うと、データの後に転送ステータスが付加される。同様のシングル・ライトではステータスのみが返される。

Q-Ignore モードのブロック転送では、CAMAC モジュールから返される Q の値にかかわらず要求した転送ワード数の転送を行う。Q-Repeat モードのブロック転送では、モジュールから Q=0 が返される場合は Q=1 になるまで再実行し、要求した転送ワード数に達するまで転送を行う。Q-Stop モードのブロック転送では要求した転送ワード数に達するか、Q=0 になるまでデータ転送を行う。Address Scan モードのブロック転送では、指定した開始ステーション番号 (N) から終了ステーション番号のモジュールまでサブアドレス (A) を1ずつ増加させて転送を行う。転送の際に Q=0 の場合、A を 0 にして次のステーション (N+1) のスキャンを行う。指定した終了ステーション番号に達し、かつ Q=0 となるサブアドレスに達するか、あるいは指定の転送ワード数に達するまで転送を行う。

各転送のデータフォーマットについては 4.3 で、その転送手順については 4.4 で説明する。

## 2 ハードウェアおよびソフトウェアの構成

### 2.1 ハードウェアの構成

CC/USB を用い USB 接続で CAMAC を制御するために必要なハードウェアを次に示す。ハードウェアのセットアップについて付録 B.1 で具体的に説明する。

- USB インターフェイスをもつ PC/AT 互換機。
- TOYO CAMAC クレートコントローラ CC/7700
- TOYO CC/USB(図 1 参照)
- USB ケーブル

CC/USB は USB コントローラとして Cypress 社の CY7C64613(EZ-USB FX) チップ [6] と低消費電力タイプの CPLD を搭載している。

### 2.2 ソフトウェアの構成

上記のハードウェアを用いて Linux で CAMAC を制御するために必要なソフトウェアを次に示す。

- Linux (Linux カーネル 2.4.x およびその配付パッケージ)。
- CC/USB 用デバイスドライバ ccusb。
- CAMAC 用 ライブラリ camlib。

利用者は camlib を利用することにより、これまで慣れ親しんできた camac() 等の関数を使用することができる。ソフトウェアのインストールについての具体的な説明は付録 B.2 で、camlib については付録 C で説明する。



図 1: CC/USB の全体図。CC/USB は直接 CC/7700 へ取り付けが可能である。

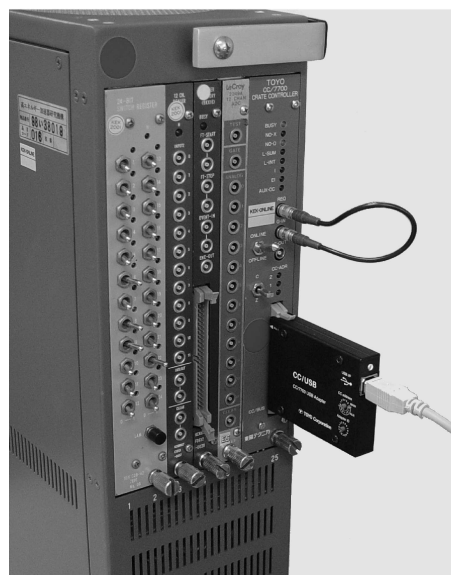


図 2: CC/USB を CC/7700 へ取り付け付けた状態。

### 3 ハードウェア仕様

CC/USB のハードウェア仕様を表 3 に示す。

項目	仕様
対応規格	USB1.1
電源供給	USB ポートからのバスパワー供給で動作
電源容量	最大 350mA
外形寸法	83.5(W) × 58.5(H) × 15.5(D) mm (突起部含まず)
重量	約 80g
コネクタ	40pin メス MIL 準拠ヘッダ, USB コネクタ (シリーズ B 型)
スイッチ	ロータリースイッチ × 2, リセットスイッチ
インジケータ	USB 接続時に LED が点灯
付属品	USB ケーブル, 延長用 40pin フラットケーブル

表 3: CC/USB のハードウェア仕様。

表 3 のロータリースイッチとリセットスイッチについて説明する。

- ロータリースイッチ
  - CC アドレス設定スイッチ (図 1 参照)
  - アダプタ ID 設定スイッチ (図 1 参照)

図 1 の写真の前面にロータリースイッチが 2 個ついている。上は CC アドレス設定用で CC/7700 のアドレスと一致させなければならない。CC/7700 をメインのクレートコントローラとして使う場合「MAIN」と書かれた番号の範囲に、また補助クレートコントローラとして使う場合は「AUX」と書かれた番号の範囲に CC/7700 のアドレスをそれぞれ設定する。下は CC/USB 自身の ID を設定するアダプタ ID 設定スイッチである。両者はそれぞれ独立して設定可能であるが、後述する Linux で利用可能なデバイスドライバでは CC/USB のアダプタ ID は CC アドレスと同じ番号であることを前提にしている。これは複数台の CC/7700 と CC/USB によるマルチクレートを実現する場合に必要であり、詳細は 5.3 で説明する。

- リセットスイッチ

図 1 写真の背面にリセットスイッチを押すための小さな穴がある。通常これを利用する必要はないが、CC/USB の制御が不可能になった場合に使用する。USB ケーブルを抜いても同様の効果が得られる。

### 4 CC/USB の転送プロトコルの仕様

まず転送プロトコルの仕様を決定する際の方針を説明し、続いてエンドポイントの構成、データフォーマットについて述べ、最後に転送のシーケンスについて説明する。

## 4.1 CC/USB で用いる転送プロトコル仕様の方針

我々は次に述べる方針のもとに CC/USB の転送プロトコル仕様を決定した。

- **CC/USB で使用する転送プロトコルは，CC/7700 に依存しないこと**  
他のクレートコントローラにも応用が可能なように CC/7700 に依存した転送プロトコルの実装はしない。
- **ホストと CC/USB 間の USB 転送はバルク転送を用いる**  
USB の転送には制御，割り込み，アイソクロナス，バルクという転送方式があるがハードディスクやスキャナのような大量のデータを転送するデバイスで用いられているバルク転送を採用する。バルク転送と同様に大量のデータ転送用にアイソクロナス転送があるが，これはデータ転送幅と転送時間は保証するがデータのエラーに対する保証(再送)は行わない。我々がバルク転送を採用した最大の理由は，バルク転送には再送によるデータの保証があるからである。
- **CAMAC シングル転送の実装には USB 転送の回数を極力減らす**  
USB1.1 の転送速度はフルスピードモードで 1.5MB/s である。しかし USB1.1 ではホストコントローラは 1msec 毎に転送のスケジューリングを行うため 1 回の転送にかかるオーバーヘッドが大きい。CAMAC シングル転送実現のため USB 転送を多く使うとその実行時間が数 msec におよぶ可能性があるため，より少ない回数の USB 転送で実現する方法を考えた。例えば，シングルライトの場合は，コマンドと一緒にデータを送る。ステータス付きシングルリード場合は，データと一緒に Q, X を含む転送ステータスを読む等である。
- **CAMAC LAM によるハードウェア割り込みは実装しない**  
CC/7700 には LAM 検出によるコンピュータへのハードウェア割り込み機能があるが，CC/USB ではサポートしない。USB では全ての転送の開始はホストコントローラにより行われるためデバイスが CPU に対して直接割り込みを通知することは原理的に不可能であるからである。USB 割り込み転送による実装も検討したが，割り込み転送は最小でも 1msec のポーリングで行われるためレイテンシが大きくメリットがないと判断し行わなかった。LAM の情報を取得したい場合は，ステーションを指定して CAMAC ファンクション F(8) を発行するか，CC/USB の制御コマンドを発行して全ステーションの LAM パターンを取得する方法を用いる<sup>1</sup>。

## 4.2 CC/USB のエンドポイント構成

USB にはエンドポイントという概念がある。エンドポイントとは転送データを格納するための FIFO のことであり，ホストとデバイスのデータの受渡しは必ずこのエンドポイントを介して行われる。デバイスは複数のエンドポイントを持つことができ，それぞれ番号がづ

---

<sup>1</sup> CC/7700 の ROM を交換することにより，CC/USB のステータスで LAM SUM の情報を取得することも可能である。



いている。このエンドポイントの構成は、USB デバイスを設計する上で重要である。一般的に USB デバイスは複数のエンドポイントを持ち、それらと上述の制御、割り込み、アイソクロナス、バルクという転送方式の組合せで成り立っている。例えば USB ハードディスクは、複数のエンドポイントを持ちデータ転送にはバルク転送をエンドポイントの STALL<sup>2</sup> 状態の解消には制御転送を用いる。またハードディスクやフロッピーディスク等は USB マスストレージ・クラスとして定義されており共通のデバイスドライバを利用できる。我々が CC/USB のエンドポイント構成を決定する上で考慮したことは、使用するエンドポイントの数は少なくし、シンプルな構成にするということである。また転送ステータス取得のためのエンドポイントとブロック転送で使用するエンドポイントは別にした。これは、ブロック転送時に帯域を占有されることなく、ステータス取得等の動作の実装が容易になると考えたからである。このエンドポイントの構成とその機能の設計が成功すれば、今後作られる USB-CAMAC インターフェイスの標準になり得ると考えている。後述の IN, OUT はデータの流れる方向を表し、IN エンドポイントはデバイスからホストへ送るデータを生成し、OUT エンドポイントはホストからデバイスへのデータを受け入れる。

表 4 に CC/USB のエンドポイント構成を示す。

転送方向	エンドポイント番号	データ内容
IN	1	シングルリード・データ, 転送ステータス, LAM ステータス
	4	ブロックリード・データ
OUT	1	制御コマンド, CAMAC コマンド, シングルライト・データ
	4	ブロックライト・データ

表 4: CC/USB のエンドポイント構成.

### 4.3 データフォーマット

4.2 で説明したエンドポイント毎に USB 転送で用いるデータのフォーマットについて説明する。各データフォーマットは表にして付録 A とした。

#### 4.3.1 OUT1 エンドポイント

OUT1 エンドポイントでは CC/USB 制御コマンド, CAMAC コマンド, シングルライト・データを受け付ける。

- 制御コマンドのフォーマットを表 9 に示す。bit4 で転送データ幅 (16bit/24bit) の指定を行う。bit3 から bit0 で動作を指定する (表 10)。
- CAMAC コマンドのデータフォーマットを表 11 に示す。CAMAC コマンドはシングル転送および Address Scan モード以外のブロック転送は N, A, F を指定する。Address

<sup>2</sup> STALL とはデバイスのエンドポイントがエラーのため停止状態のことをいう。

Scan モード・ブロック転送の場合は、bit15 から bit8 にスキャンを開始するステーション番号を bit23 から bit16 にスキャンを終了するステーション番号を指定する。ブロック転送の場合は更に転送ワード個数を指定する。

- CAMAC シングルライト・データのフォーマットを表 12 に示す。

#### 4.3.2 OUT4 エンドポイント

- CAMAC ブロックライト・データ (16bit) のフォーマットを表 13 に示す。
- CAMAC ブロックライト・データ (24bit) のフォーマットを表 14 に示す。

#### 4.3.3 IN1 エンドポイント

- CAMAC シングルリード・データのフォーマットを表 15 に示す。
- CAMAC 転送ステータス・データのフォーマットを表 16 に示す。また表 17 に表 16 の bit7, bit6 により示されるステータス情報を示す。
- CAMAC LAM ステータス・データのフォーマットを表 18 に示す。
- CC/USB ID データのフォーマットを表 19 に示す。
- ブロック転送ステータス・データのフォーマットを表 20 に示す。

#### 4.3.4 IN4 エンドポイント

- CAMAC ブロックリード・データ (16bit) のフォーマットを表 21 に示す。
- CAMAC ブロックリード・データ (24bit) のフォーマットを表 22 に示す。

### 4.4 データ転送シーケンス

指定した CAMAC 動作を実行するためのデータ転送のシーケンスを次に示す。

#### 4.4.1 CAMAC シングル転送

- シングルリード
  1. エンドポイント OUT1 へ制御コマンド (表 9), CAMAC コマンド (表 11) を送る。
  2. エンドポイント IN1 からデータ (表 15) を読み込む。ステータスを要求した場合はデータの後に付加されるステータス (表 16) も同時に読み込む。
- シングルライト

1. エンドポイント OUT1 へ制御コマンド (表 9), CAMAC コマンド (表 11), データ (表 12) を送る.
2. ステータスを要求した場合は, エンドポイント IN1 からステータス (表 16) を読み込む.

#### 4.4.2 CAMAC ブロック転送

CAMAC ブロック転送で転送モードに依存した Q の扱いとモジュールに対する転送制御は, CC/USB 内部で行っているためホストでは意識する必要はない. ただし, Q-Stop, Address Scan モードでは 1.3 で説明したように要求した転送ワード数に達する前に転送が終了する場合があります, その際にリードの場合は 64Byte 未満のデータ長になるまでデータを読み, またライトの場合はエンドポイントが STALL するまでデータを送るといった動作をさせている.

- Q-Ignore および Q-Repeat リード

1. エンドポイント OUT1 へ制御コマンド (表 9), CAMAC コマンド (表 11), 転送ワード数 (表 11) を送る.
2. 要求した転送ワード数に達するまでエンドポイント IN4 からデータ (表 21,22) を読み込む.
3. 転送途中または転送後にステータスを調べる場合はエンドポイント OUT1 へステータス読みだしのための制御コマンド (表 9) を送り, エンドポイント IN1 から転送ステータス (表 16) を読み込む.

- Q-Stop および Address Scan リード

1. エンドポイント OUT1 へ制御コマンド (表 9), CAMAC コマンド (表 11), 転送ワード数 (表 11) を送る.
2. 要求した転送ワード数に達するか, 64byte 未満のデータ長になるまでエンドポイント IN4 からデータ (表 21,22) を読み込む.
3. 転送途中または転送後にステータスを調べる場合はエンドポイント OUT1 へステータス読みだしのための制御コマンド (表 9) を送り, エンドポイント IN1 から転送ステータス (表 16) を読み込む.

- Q-Ignore および Q-Repeat ライト

1. エンドポイント OUT1 へ制御コマンド (表 9), CAMAC コマンド (表 11), 転送ワード数 (表 11) を送る.
2. 要求した転送ワード数に達するまでエンドポイント OUT4 にデータ (表 13,14) を送る.

3. 転送途中または転送後にステータスを調べる場合はエンドポイント OUT1 へステータス読みだしのための制御コマンド (表 9) を送り, エンドポイント IN1 から転送ステータス (表 16) を読み込む.

- Q-Stop および Address Scan ライト

1. エンドポイント OUT1 へ制御コマンド (表 9), CAMAC コマンド (表 11), 転送ワード数 (表 11) を送る.
2. 要求した転送ワード数に達するか, STALL するまでエンドポイント OUT4 へデータ (表 13,14) を送る.
3. 転送途中または転送後にステータスを調べる場合はエンドポイント OUT1 へステータス読みだしのための制御コマンド (表 9) を送り, エンドポイント IN1 から転送ステータス (表 16) を読み込む.

#### 4.4.3 CAMAC LAM ステータス

1. エンドポイント OUT1 へ LAM ステータス読みだしのための制御コマンド (表 9) を送る.
2. エンドポイント IN1 から LAM ステータス・データ (表 18) を読み込む.

#### 4.4.4 CAMAC 制御コマンド

1. エンドポイント OUT1 へ制御コマンド (表 9) を送る.
2. ステータスを確認する場合は, エンドポイント OUT1 からステータス読みだしのための制御コマンド (表 9) を送り, エンドポイント IN1 からステータス (表 16) を読み込む.

## 5 CC/USB の Linux 用デバイスドライバ

Linux から CC/USB 経由で CAMAC を制御するためのデバイスドライバを開発した. このデバイスドライバは Linux カーネル 2.4.x に対応している. このデバイスドライバはカーネルソースの中に入っている usb-skeleton.c を参考にした.

### 5.1 サポートしている CC/USB の制御コマンド

表 10 の「ステータス無しシングル転送」を除いた制御コマンドをサポートしている. シングル転送では, 転送後にステータスを読んで Q, X の情報および転送中のエラーの有無を確認している.

## 5.2 サポートするファイル操作関数

本デバイスドライバは、Linux のキャラクタ型デバイスドライバでファイル操作関数として `open()`, `close()`, `ioctl()` をサポートしている。利用者は `ioctl` 関数の 3 番目の引数として CAMAC 制御用の構造体を定義しデバイスドライバへ渡す。 `ioctl()` 関数は表 5 のコマンドをサポートしている。デバイスドライバは、その構造体から CAMAC コマンドを抽出して発行した後、データと転送ステータスを読み出し、それらを構造体に入れて利用者へ返す。本デバイスドライバは、この方法により CAMAC シングルリードに対し 1 回のシステムコールでデータを読み出せる。一方、`read()`, `write()` 関数で実装する場合は `write()` でコマンドを送り、その後 `read()` によりデータを読み出す。このようにシステムコール発行によるオーバーヘッドが 2 回発生するため `read()`, `write()` 関数はサポートしていない。

ioctl コマンド	説明
CAM_WORD	CAMAC 16bit シングル転送を実行する
CAM_LONG	CAMAC 24bit シングル転送を実行する
CAM_BLKW	CAMAC 16bit ブロック転送を実行する
CAM_BLKL	CAMAC 24bit ブロック転送を実行する
CAM_INIT	CAMAC の初期化を行う
CAM_CLEAR	CAMAC クリアを行う
CAM_SETI	CAMAC インヒビットをセットする
CAM_CLRI	CAMAC インヒビットを解除する
CAM_SETLAM	CC/7700 の LAM をセットする
CAM_CLRLAM	CC/7700 の LAM を解除する
CAM_STATUS	CC/7700 の ON/OFF-Line, CAMAC のステータスを読む
CAM_READLAM	CAMAC の LAM パターンを読む
CAM_READID	CC/USB の ID を読む

表 5: CC/USB のデバイスドライバでサポートしているファイル操作関数。

## 5.3 CAMAC マルチクレートのサポート

CC/7700 には 1 台のホストに対しディジーチェーン接続による最大 8 台までのマルチクレート機能がある。CC/USB ではディジーチェーン接続はできないが USB Hub を介してマルチクレートとして使用できる。USB デバイスは USB バスに動的に接続/取り外しが行われるためシステムが与える USB の ID では各 CC/USB を同定できない。そのためデバイスドライバでは CC/USB がバスに接続する毎にそのアダプタ ID を読み、それに対応するマイナー番号を割り当てる方法をとっている。この方法でアダプタ ID 0 から 7 に対応してマイナー番号 200 から 207 をアサインする。このマイナー番号に対応したデバイスノード名は `/dev/usb/ccusb00` から `/dev/usb/ccusb07` である。本デバイスドライバではクレート番号と CC/USB が 1 対 1 で対応するようにアダプタ ID と CC/7700 のアドレスは同じにすることを前提にしている。そのため利用者は複数のクレート番号と CC/USB のアダプタ ID をそれぞれ管理する必要はない。

## 6 CC/USBの性能

表 6 に示す環境で 5 で説明したデバイスドライバを用いて CC/USB の CAMAC データ転送の性能を測定した。

### 6.1 CAMAC シングル転送性能

CAMAC スイッチ・レジスタに対してシングルリードを 10000 回行い、転送時間の分布を調べた。その結果を図 3 に示す。シングルライトについても同様の測定を行った。それらの結果を表 7 に示す。結果より CAMAC シングル転送の実行には平均 1.5msec の時間を要することがわかる。これは USB1.x 仕様ではホストコントローラが 1msec 毎のフレームという概念で転送のスケジューリングを行っていることに起因する。

性能テストに使用した PC のスペック		ソフトウェア環境	
CPU	PentiumII 300MHz	Linux Distribution	RedHat7.3
RAM	192MB	Linux Kernel	2.4.19
USB Controller	Intel 82371AB/EB/MB	gcc version	gcc-2.96

表 6: CC/USB の性能テスト環境。表の左側が性能評価に用いた PC のスペックで右側がそのソフトウェア環境。

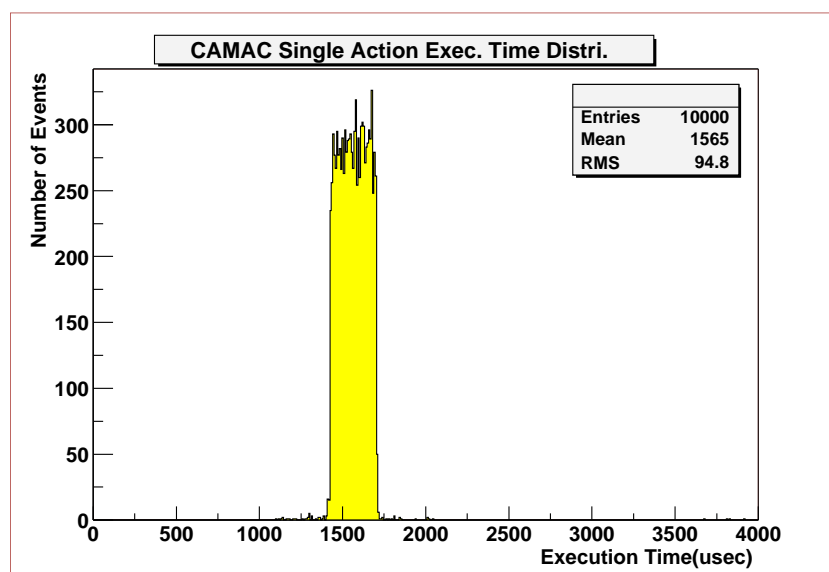


図 3: CC/USB の CAMAC シングル転送実行時間の分布。グラフの横軸はシングル転送の実行時間 ( $\mu\text{sec}$ )。縦軸はイベント数。

## 6.2 CAMAC ブロック転送性能

CAMAC メモリモジュール (8K x 16bit) に対してデータ長を変えながら Q-Ignore モードのブロック転送による書き込みと読み込みを行い、転送に要した時間を測定した。その結果から転送速度を求めグラフにプロットした (図 4)。またそのオーバーヘッドを求めた。その結果を表 8 に示す。

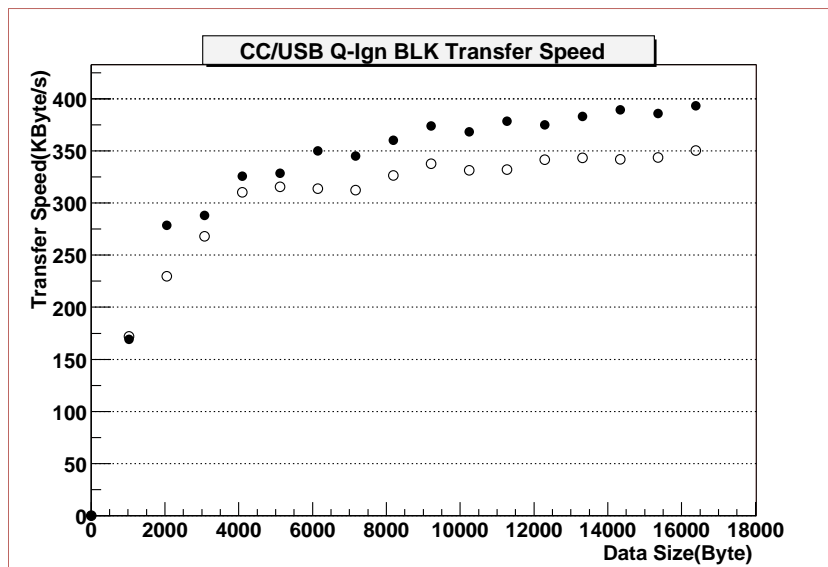


図 4: CC/USB の CAMAC ブロック転送速度。グラフ横軸は、ブロック転送を行ったデータ長 (Byte)。縦軸はデータ転送速度 (KBytes/sec)。グラフ中の黒丸はリード、白丸はライトのデータである。

	シングル転送時間
READ	1.5 msec
WRITE	1.5 msec

表 7: CAMAC 16bit および 24bit シングル転送性能

	ブロック転送速度	Overhead
READ	422.2 KBytes/sec	3.2 msec
WRITE	370.2 KBytes/sec	3.0 msec

表 8: CAMAC ブロック転送 (16bit Q-Ignore モード) の性能

## 7 まとめ

CAMAC を用いたポータブル DAQ 実現のため, CC/7700 と USB の変換アダプタ CC/USB を提案し東陽テクニカ社と協議して仕様を決定し, 同社が製作した. Linux カーネル 2.4.x で利用できる CC/USB のデバイスドライバを開発し, それを用いて CC/USB の性能を測定した. CC/USB の CAMAC DAQ 性能は シングル転送時間が平均 1.5msec, ブロック転送スピードが約 420KBytes/sec(16bit Q-Ignore モード・リード)であった. 我々の目的であったノート型 PC によるポータブルな CAMAC DAQ システムの実現が可能となった. CC/USB デバイスドライバおよびそのユーザインターフェイスとなる CAMAC ライブラリは, 配付キット [7] としてまとめ WEB 上からダウンロードが可能である.

## 謝辞

藤井啓文氏をはじめとするオンラインおよび回路グループの方々に感謝致します.

## 参考文献

- [1] USB に関する詳細は, 次に示す USB コンソシアムのホームページを参照.  
<http://www.usb.org/>
- [2] Model 3929-Z1A SCSI Crate Controller Instruction manual, KineticSystems Corp., 1991.
- [3] Modular Instrumentation and Digital Interface System (CAMAC) ANSI/IEEE Std. 583-1982.
- [4] KineticSystems 社 SCSI クレートコントローラと市販の USB-SCSI 変換ケーブルを用いて USB で CAMAC 制御を行うための Linux 用デバイスドライバを開発した. 詳細は, 次の URL を参照. <http://www-online.kek.jp/~nakayosi/USB/usb-camac.html>
- [5] CAMAC クレート・コントローラ CC/7700 型 取扱説明書, 株式会社東陽テクニカ.
- [6] Cypress 社 CY7C6413(EZ-USB FX) チップに関する情報は, 次の URL を参照.  
<http://www.cypress.com/>
- [7] CC/USB デバイスドライバと CAMAC ライブラリの配付キットについては, 次の URL を参照のこと. <http://www-online.kek.jp/~nakayosi/>.



## A CC/USB の各エンドポイントで用いるデータフォーマット

4.3 で説明する各エンドポイント毎のデータフォーマットを表にして示す。

### A.1 OUT1 エンドポイント

OUT1 エンドポイントでは、CC/USB 制御コマンド、CAMAC コマンド、シングルライ  
ト・データを取り扱う。

bit7	bit 6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	24bit	COM3	COM2	COM1	COM0

表 9: 制御コマンドのデータフォーマット。24bit のデータ転送を行う場合は、bit4 を 1 にする。

bit3-bit0(16 進表記)	コマンド内容
0x00	ステータス無しシングル転送
0x01	ステータス付きシングル転送
0x02	Q-Stop モード・ブロック転送
0x03	Q-Ignore モード・ブロック転送
0x04	Q-Repeat モード・ブロック転送
0x05	Address Scan モード・ブロック転送
0x06	CAMAC イニシャライズ
0x07	CAMAC クリア
0x08	セット・インヒビット
0x09	クリア・インヒビット
0x0A	セット・LAM
0x0B	クリア・LAM
0x0C	ステータス読みだし
0x0D	LAM ステータス読みだし
0x0E	CC/USB ID 読みだし
0x0F	予約

表 10: 制御コマンドの bit3 から bit0 で指定する値と対応する動作。

シングル転送, ブロック転送 (Q-Stop, Q-Ignore, Q-Repeat)

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
0	0	0	N16	N8	N4	N2	N1

bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
0	0	0	0	A8	A4	A2	A1

bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
0	0	0	F16	F8	F4	F2	F1

ブロック転送 (Adress Scan)

Adress Scan 開始ステーション番号

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
0	0	0	N16	N8	N4	N2	N1

Adress Scan 終了ステーション番号

bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
0	0	0	N16	N8	N4	N2	N1

bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
0	0	0	F16	F8	F4	F2	F1

転送ワード数 (ブロック転送の場合)

bit39	bit38	bit37	bit36	bit35	bit34	bit33	bit32
LEN7	LEN6	LEN5	LEN4	LEN3	LEN12	LEN1	LEN0

bit47	bit46	bit45	bit44	bit43	bit42	bit41	bit40
LEN15	LEN14	LEN13	LEN12	LEN11	LEN10	LEN9	LEN8

表 11: CAMAC コマンドのデータフォーマット.

16bit

bit39	bit38	bit37	bit36	bit35	bit34	bit33	bit32
wdat7	wdat6	wdat5	wdat4	wdat3	wdat2	wdat1	wdat0

bit47	bit46	bit45	bit44	bit43	bit42	bit41	bit40
wdat15	wdat14	wdat13	wdat12	wdat11	wdat10	wdat9	wdat8

24bit

bit39	bit38	bit37	bit36	bit35	bit34	bit33	bit32
wdat7	wdat6	wdat5	wdat4	wdat3	wdat2	wdat1	wdat0

bit47	bit46	bit45	bit44	bit43	bit42	bit41	bit40
wdat15	wdat14	wdat13	wdat12	wdat11	wdat10	wdat9	wdat8

bit55	bit54	bit53	bit52	bit51	bit50	bit49	bit48
wdat23	wdat22	wdat21	wdat20	wdat19	wdat18	wdat17	wdat16

表 12: CAMAC シングルライト・データのフォーマット.

## A.2 OUT4 エンドポイント

OUT4 エンドポイントでは、PC から CAMAC ヘブロック転送するデータを取り扱う。

1st word							
bit7	bit6	bit5	bit4	bit3	bit 2	bit1	bit0
wdat7	wdat6	wdat5	wdat4	wdat3	wdat2	wdat1	wdat0
...							
bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
wdat15	wdat14	wdat13	wdat12	wdat11	wdat10	wdat9	wdat8
2nd word							
bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
wdat7	wdat6	wdat5	wdat4	wdat3	wdat2	wdat1	wdat0
...							
bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
wdat15	wdat14	wdat13	wdat12	wdat11	wdat10	wdat9	wdat8
N-th word							
16(N-1)+7	16(N-1)+6	16(N-1)+5	16(N-1)+4	16(N-1)+3	16(N-1)+2	16(N-1)+1	16(N-1)
wdat7	wdat6	wdat5	wdat4	wdat3	wdat2	wdat1	wdat0
...							
16(N-1)+15	16(N-1)+14	16(N-1)+13	16(N-1)+12	16(N-1)+11	16(N-1)+10	16(N-1)+9	16(N-1)+8
wdat15	wdat14	wdat13	wdat12	wdat11	wdat10	wdat9	wdat8

表 13: CAMAC ブロックライト・データ (16bit) のフォーマット。

1st word							
bit7	bit6	bit5	bit4	bit3	bit 2	bit1	bit0
wdat7	wdat6	wdat5	wdat4	wdat3	wdat2	wdat1	wdat0
...							
bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
wdat15	wdat14	wdat13	wdat12	wdat11	wdat10	wdat9	wdat8
bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
wdata23	wdat22	wdat21	wdat20	wdat19	wdat18	wdat17	wdat16
2nd word							
bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
wdat7	wdat6	wdat5	wdat4	wdat3	wdat2	wdat1	wdat0
...							
bit39	bit38	bit37	bit36	bit35	bit34	bit33	bit32
wdat15	wdat14	wdat13	wdat12	wdat11	wdat10	wdat9	wdat8
bit47	bit46	bit45	bit44	bit43	bit42	bit41	bit40
wdata23	wdat22	wdat21	wdat20	wdat19	wdat18	wdat17	wdat16
N-th word							
24(N-1)+7	24(N-1)+6	24(N-1)+5	24(N-1)+4	24(N-1)+3	24(N-1)+2	24(N-1)+1	24(N-1)
wdat7	wdat6	wdat5	wdat4	wdat3	wdat2	wdat1	wdat0
...							
24(N-1)+15	24(N-1)+14	24(N-1)+13	24(N-1)+12	24(N-1)+11	24(N-1)+10	24(N-1)+9	24(N-1)+8
wdat15	wdat14	wdat13	wdat12	wdat11	wdat10	wdat9	wdat8
24(N-1)+23	24(N-1)+22	24(N-1)+21	24(N-1)+20	24(N-1)+19	24(N-1)+18	24(N-1)+17	24(N-1)+16
wdata23	wdat22	wdat21	wdat20	wdat19	wdat18	wdat17	wdat16

表 14: CAMAC ブロックライト・データ (24bit) のフォーマット。

### A.3 IN1 エンドポイント

IN1 エンドポイントでは、シングルリード・データ、転送ステータス、LAM ステータスを取り扱う。

16bit

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
rdat7	rdat6	rdat5	rdat4	rdat3	rdat2	rdat1	rdat0

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
rdat15	rdat14	rdat13	rdat12	rdat11	rdat10	rdat9	rdat8

24bit

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
rdat7	rdat6	rdat5	rdat4	rdat3	rdat2	rdat1	rdat0

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
rdat15	rdat14	rdat13	rdat12	rdat11	rdat10	rdat9	rdat8

bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
rdat23	rdat22	rdat21	rdat20	rdat19	rdat18	rdat17	rdat16

表 15: CAMAC シングルリード・データのフォーマット。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Error	Off-line	Timeout	0	L sum	I Status	No-X	No-Q

表 16: CAMAC 転送ステータス・データのフォーマット。

bit7	bit6	Status
1	-	CC/USB が PC に接続されているが CC/7700 には未接続, または CC/7700 のアドレスと CC/USB のロータリー・スイッチの設定が不一致
0	1	CC/USB が PC に接続され CC/7700 にも接続. CC/7700 が OFF LINE
0	0	CC/USB が PC に接続され CC/7700 にも接続. CC/7700 も ON LINE

表 17: CAMAC 転送ステータス・データの bit7, bit6 が示す CC/USB の接続状態。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
LAM8	LAM7	LAM6	LAM5	LAM4	LAM3	LAM2	LAM1

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
LAM16	LAM15	LAM14	LAM13	LAM12	LAM11	LAM10	LAM9

bit23	bit33	bit21	bit20	bit19	bit18	bit17	bit16
LAM24	LAM23	LAM22	LAM21	LAM10	LAM19	LAM18	LAM17

表 18: CAMAC LAM ステータス・データのフォーマット.

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	0	0	0	ID3	ID2	ID1	ID0

表 19: CC/USB ID データのフォーマット.

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Error	Off-line	Timeout	0	L sum	I Status	0	0

転送ワード 個数

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
LEN7	LEN6	LEN5	LEN4	LEN3	LEN2	LEN1	LEN0

bit23	bit33	bit21	bit20	bit19	bit18	bit17	bit16
LEN15	LEN14	LEN13	LEN12	LEN11	LEN10	LEN9	LEN8

表 20: CAMAC ブロック転送ステータス・データのフォーマット.

## A.4 IN4 エンドポイント

IN4 エンドポイントでは、CAMAC から PC へのブロック転送を取り扱う。

1st word							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
rdat7	rdat6	rdat5	rdat4	rdat3	rdat2	rdat1	rdat0
bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
rdat15	rdat14	rdat13	rdat12	rdat11	rdat10	rdat9	rdat8
2nd word							
bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
rdat7	rdat6	rdat5	rdat4	rdat3	rdat2	rdat1	rdat0
bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
rdat15	rdat14	rdat13	rdat12	rdat11	rdat10	rdat9	rdat8
...							
N-th word							
16(N-1)+7	16(N-1)+6	16(N-1)+5	16(N-1)+4	16(N-1)+3	16(N-1)+2	16(N-1)+1	16(N-1)
rdat7	rdat6	rdat5	rdat4	rdat3	rdat2	rdat1	rdat0
16(N-1)+15	16(N-1)+14	16(N-1)+13	16(N-1)+12	16(N-1)+11	16(N-1)+10	16(N-1)+9	16(N-1)+8
rdat15	rdat14	rdat13	rdat12	rdat11	rdat10	rdat9	rdat8

表 21: CAMAC ブロックリード・データ (16bit) のフォーマット。

1st word							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
rdat7	rdat6	rdat5	rdat4	rdat3	rdat2	rdat1	rdat0
bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
rdat15	rdat14	rdat13	rdat12	rdat11	rdat10	rdat9	rdat8
bit23	bit22	bit21	bit20	bit19	bit18	bit17	bit16
rdat23	rdat22	rdat21	rdat20	rdat19	rdat18	rdat17	rdat16
2nd word							
bit31	bit30	bit29	bit28	bit27	bit26	bit25	bit24
rdat7	rdat6	rdat5	rdat4	rdat3	rdat2	rdat1	rdat0
bit39	bit38	bit37	bit36	bit35	bit34	bit33	bit32
rdat15	rdat14	rdat13	rdat12	rdat11	rdat10	rdat9	rdat8
bit47	bit46	bit45	bit44	bit43	bit42	bit41	bit40
rdat23	rdat22	rdat21	rdat20	rdat19	rdat18	rdat17	rdat16
...							
N-th word							
24(N-1)+7	24(N-1)+6	24(N-1)+5	24(N-1)+4	24(N-1)+3	24(N-1)+2	24(N-1)+1	24(N-1)
rdat7	rdat6	rdat5	rdat4	rdat3	rdat2	rdat1	rdat0
24(N-1)+15	24(N-1)+14	24(N-1)+13	24(N-1)+12	24(N-1)+11	24(N-1)+10	24(N-1)+9	24(N-1)+8
rdat15	rdat14	rdat13	rdat12	rdat11	rdat10	rdat9	rdat8
24(N-1)+23	24(N-1)+22	24(N-1)+21	24(N-1)+20	24(N-1)+19	24(N-1)+18	24(N-1)+17	24(N-1)+16
rdat23	rdat22	rdat21	rdat20	rdat19	rdat18	rdat17	rdat16

表 22: CAMAC ブロックリード・データ (24bit) のフォーマット。

## B CC/USB セットアップ方法

CC/USB のセットアップおよびデバイスドライバのインストール方法について説明します。セットアップを行うには本文 2.1 および 2.2 で説明する環境が必要です。

### B.1 ハードウェアのセットアップ

1. CAMAC クレートの電源を OFF にする。
2. CAMAC クレートの右端に CC/7700 をセットする。
3. USB アダプタ (CC/USB) にある "CC address" 設定用のスイッチを CC/7700 と同じにする。同様に "Adapter ID" も同じ番号にする。
4. CC/7700 のコネクタに CC/USB を取り付ける
5. CAMAC クレートの電源を ON にする。
6. CC/USB に USB ケーブルを取り付け PC と接続する。このとき CC/USB の前面にある LED が点灯することを確認する。

### B.2 ソフトウェアのセットアップ

ソフトウェアの設定について説明します。本文中で \$ `typewriter` のような書体の場合、`typewriter` とタイプしてリターンキーを押すことを意味します。\$ は一般ユーザ、# はスーパーユーザのときの Shell プロンプトを表します。

1. <http://www-online.kek.jp/~nakayosi/USB/toyo-usb.html> から TOYO CC/USB パッケージ `ccusb-x.y.z.src.tgz`(x,y,z には数字が入ります) をダウンロードする。
2. パッケージを展開する。  
\$ `tar xvfz ccusb-x.y.z.src.tgz`
3. TOYO CC/USB ドライバー, CAMAC ライブラリ, 例題プログラムを make する。  
\$ `cd ccusb-x.y.z.src`  
\$ `make`
4. スーパーユーザになり, ドライバーを load する。  
# `su # /sbin/insmod ccusb/ccusb.o`
5. ドライバーを正常にロードできたらデバイスファイルをつくる。  
# `ccusb/ccusb.mknod.sh`
6. デバイスファイルができたか確認する。# `ls -al /dev/usb/ccusb*`

```
crw-rw-rw-  1 root    root      180, 200  8月 24 15:46 /dev/usb/ccusb00
crw-rw-rw-  1 root    root      180, 201  8月 24 15:46 /dev/usb/ccusb01
crw-rw-rw-  1 root    root      180, 202  8月 24 15:46 /dev/usb/ccusb02
crw-rw-rw-  1 root    root      180, 203  8月 24 15:46 /dev/usb/ccusb03
crw-rw-rw-  1 root    root      180, 204  8月 24 15:46 /dev/usb/ccusb04
crw-rw-rw-  1 root    root      180, 205  8月 24 15:46 /dev/usb/ccusb05
crw-rw-rw-  1 root    root      180, 206  8月 24 15:46 /dev/usb/ccusb06
crw-rw-rw-  1 root    root      180, 207  8月 24 15:46 /dev/usb/ccusb07
```

## C Linux 用 CAMAC ライブラリ

CC/USB を使用してユーザが容易にプログラムを書けるように、KEK 標準 CAMAC ライブラリ camlib と似た関数をもつ Linux 用のライブラリを作った。ライブラリの構成は、表 23 のようになっている。詳細は <http://www-online.kek.jp/~nakayosi/USB/uguide-toyo.html> を御覧ください。

Function	Description
camopen()	指定したクレーンをオープンする
camse()	指定したクレーンをクローズする
cgenz()	CAMAC Z を発生する
cgenic()	CAMAC C を発生する
cseti()	CAMAC インヒビットをセットする
cremi()	CAMAC インヒビットを解除する
clamstat()	CAMAC LAM ステータスを読む
camac()	NAF を指定して 24bit CAMAC シングル転送を行う
camacw()	NAF を指定して 16bit CAMAC シングル転送を行う
cdma()	24bit CAMAC ブロック転送を行う
cdmaw()	16bit CAMAC ブロック転送を行う

表 23: Linux 用 CAMAC ライブラリの関数。

### C.1 CAMAC ライブラリを使ったプログラム例

表 23 のライブラリ関数を利用したプログラム例を次に示します。クレーン番号 0, ステーション番号 2 にある Switch Register のスイッチの値を読むためのプログラムです。

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include "ccusb.h"
#define CRATE 0
#define SWREG 2

int main()
{
    int ret, a = 1, f = 0, dat, q, x;

    ret = camopen(CRATE);
    if(ret != 0) {
        printf("ERROR: camopen ret = %d\n", ret);
        exit(-1);
    }
    ret = camac(CRATE, SWREG, a, f, &dat, &q, &x);
    if(ret != 0) {
        printf("ERROR: camac ret = %d\n", ret);
        exit(-1);
    }
    printf("data = 0x%x(%d)\n", dat, dat);
    printf("q = %d x = %d\n", q, x);
    camcls(CRATE);
}
```