



DAQ-MIDDLEWARE トレーニングコース 実習

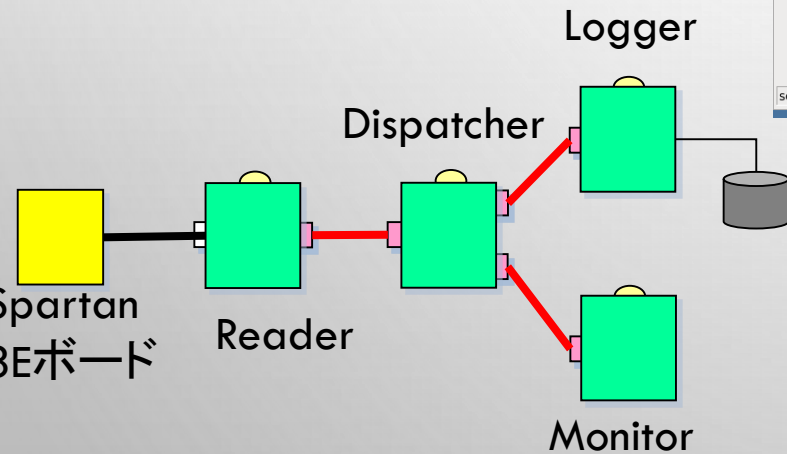
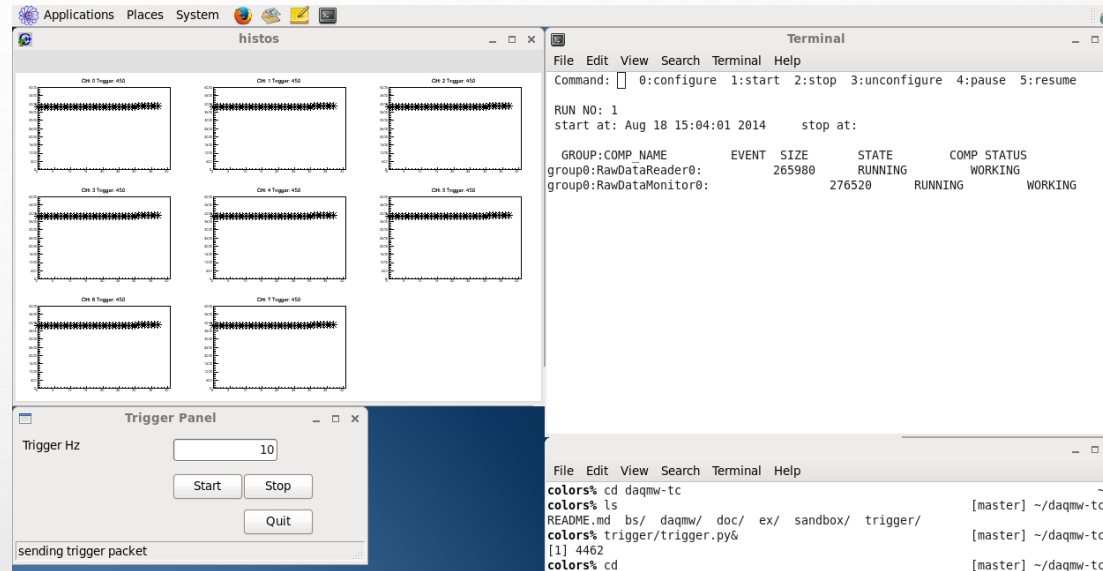
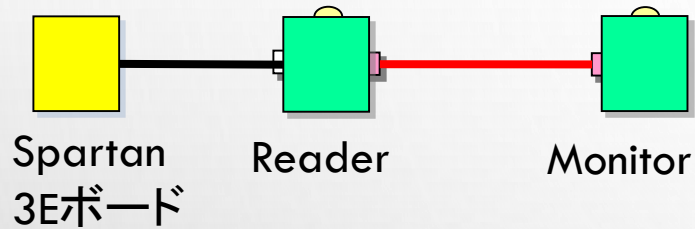
濱田英太郎

高エネルギー加速器研究機構

素粒子原子核研究所

実習最終目標

SPARTAN 3Eボードからデータを読んでグラフを画面に表示するシステム



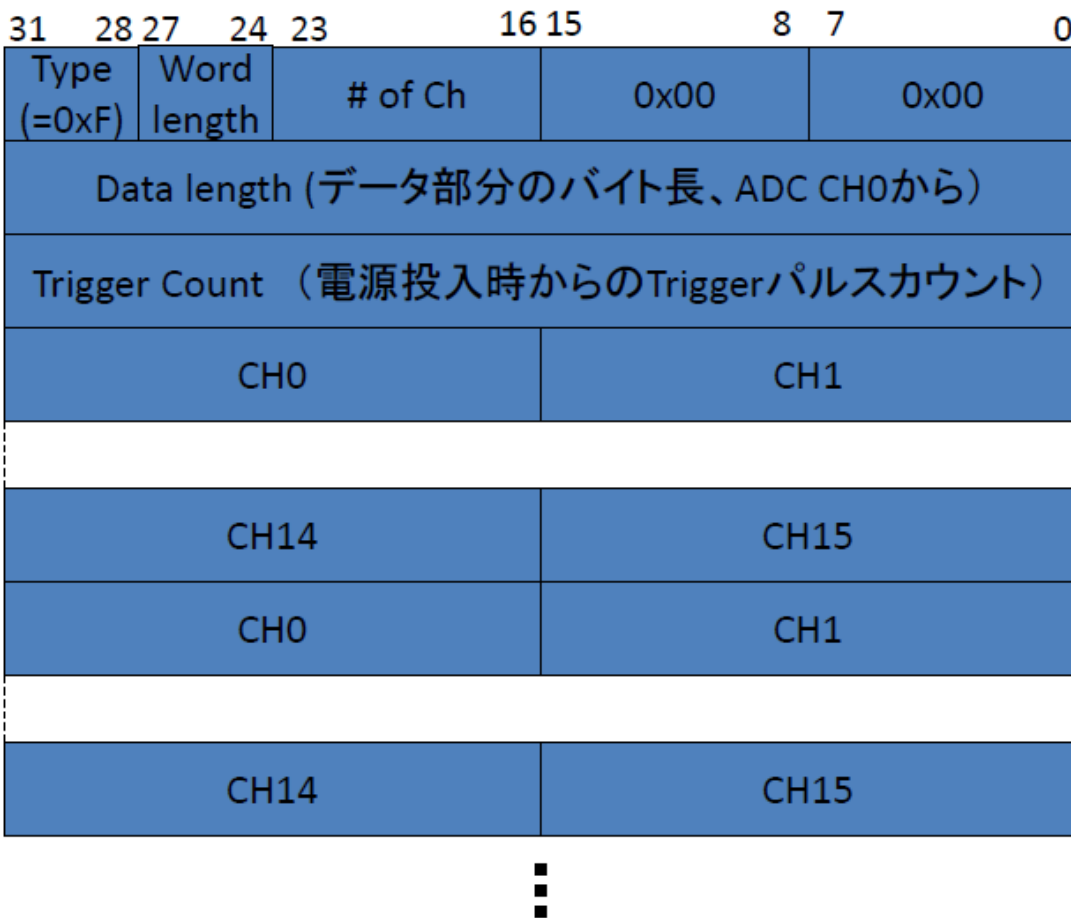
実習で行う事項

- Spartan3E評価ボードセットアップ
- ひな形、解答例ファイルダウンロード
- 実習1 (DAQ-Middlewareを利用しない)
 - ex01 実習環境確認
 - ex02 C++の簡単な復習(クラス)
 - ex03 ネットワークバイトオーダー
 - ex04 char bufferからの数値の取り出し
 - ex05 バイナリファイルの読みだし
 - ex06 ファイルを読んでデコード
 - ex07 ROOTを使ってグラフを書く
 - ex08 ファイルを読みながらグラフを画面に表示する
 - ex09 ネットワークからデータを読みデコードする
 - ex10 ncコマンドでデータを読みグラフを画面に表示する

実習で行う事項

- 実習2 (DAQ-Middlewareを利用する)
 - ex11 DAQ-Middleware付属サンプルコンポーネントを動かしてみる
 - ex12 Webモードでシステムを動かす
 - ex13 ログの確認
 - ex14 ボードを読むシステム(DAQ-Middleware使用)を動かしてみる
(Reader - Logger)
 - ex15 ボードを読んでモニターするシステムをDAQ-Middlewareで作る
(Reader - Monitor)

ボードからのデータフォーマット



Header

Type: 他のフォーマットと見分けるためにある

Word Length: 各チャンネル毎、Window毎のデータバイト長(単位バイト)
(2)

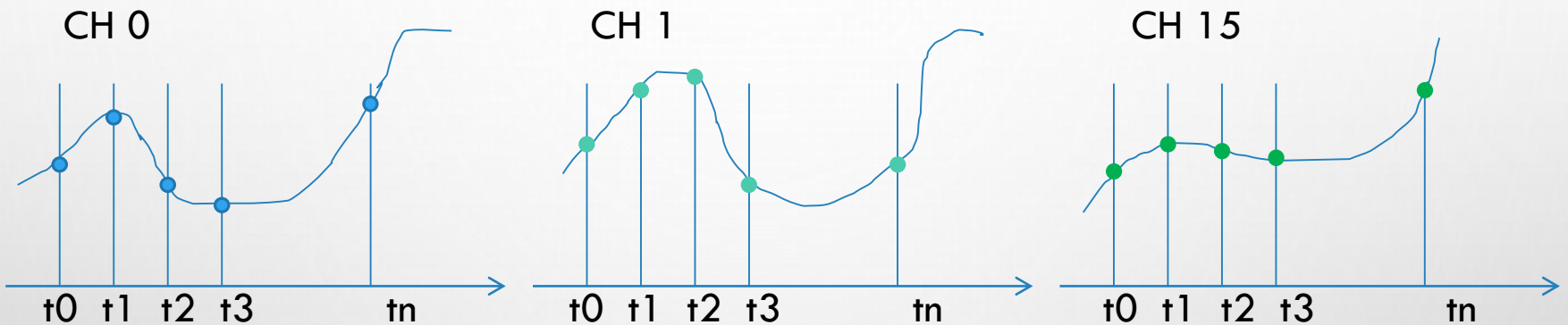
データ並びについては次のページ

Data

Time windowの長さだけ繰り返し先に送られてくるデータが時間的に古い

データ並び

1トリガーでやってくるデータ



t0 ch0	t0 ch1	t0 ch2	t0 ch3	...	t0 ch15
t1 ch0	t1 ch1	t1 ch2	t1 ch3	...	t1 ch15
tn ch0	tn ch1	tn ch2	tn ch3	...	tn ch15

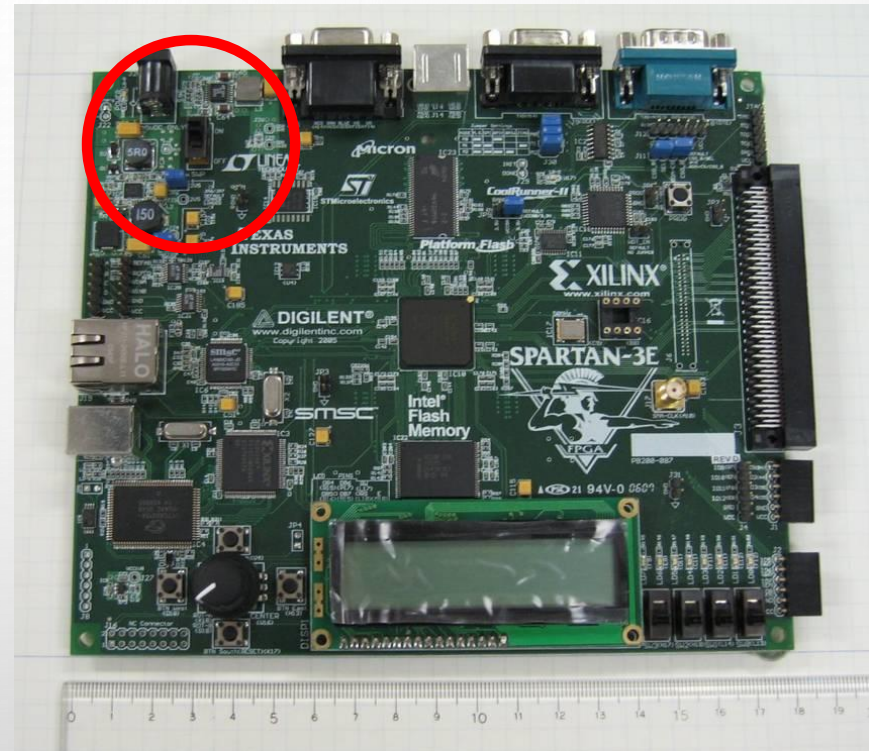
Data Length(単位バイト) / (Word length(単位バイト) × チャンネル数)
で1チャンネルあたり、1トリガーで何個データがくるかわかる

実習準備状況確認

- VirtualBoxのセットアップ
- Spartan 3Eの配布
- Spartan 3Eのセットアップ

ACアダプタ、LANケーブルをさすだけ。

電源スイッチはACアダプタコネクタそば



実習2

- 実習2 (DAQ-Middlewareを利用する)

- ex11 DAQ-Middleware付属サンプルコンポーネントを動かしてみる
- ex12 Webモードでシステムを動かす
- ex13 ログの確認

→ DAQ-Middlewareが提供しているサンプルコンポーネントを動かす

- ex14 ボードを読むシステム(DAQ-Middleware使用)を動かしてみる
(Reader - Logger)
- ex15 ボードを読んでモニターするシステムをDAQ-Middlewareで作る
(Reader - Monitor)

Readerの理解が必要

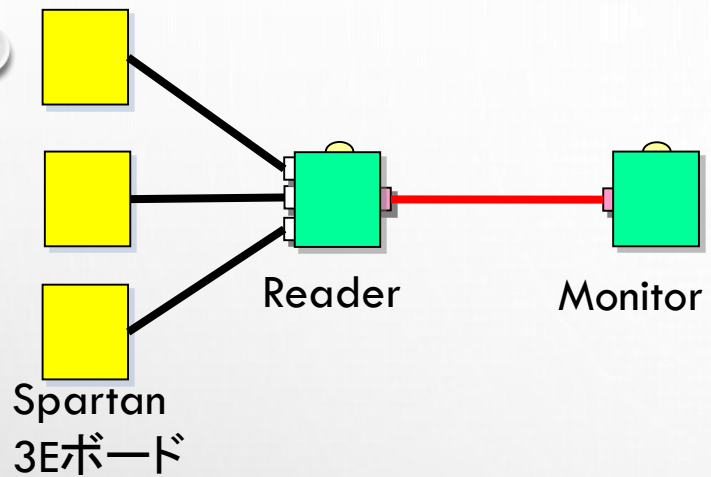
Monitorの中身を変更して、目的のシステムを作る

多重読み出し

```
read(sockfd0, buf_0, read_bytes0);  
read(sockfd1, buf_1, read_bytes1);  
read(sockfd2, buf_2, read_bytes2);
```

- sockfd0のread()が終了するまでsockfd1のread()は実行されない
- sockfd1にデータが来ていてもsockfd0にデータがきていなければsockfd1は読めない
- 解決方法
 - select() あるいは epoll()
 - 読めるようになったものがあれば通知してくれる関数
 - 1 sockfdあたり1個のスレッドをわりあてて独立に読めるようにする。

DAQ-Middleware 多重読みだしの例



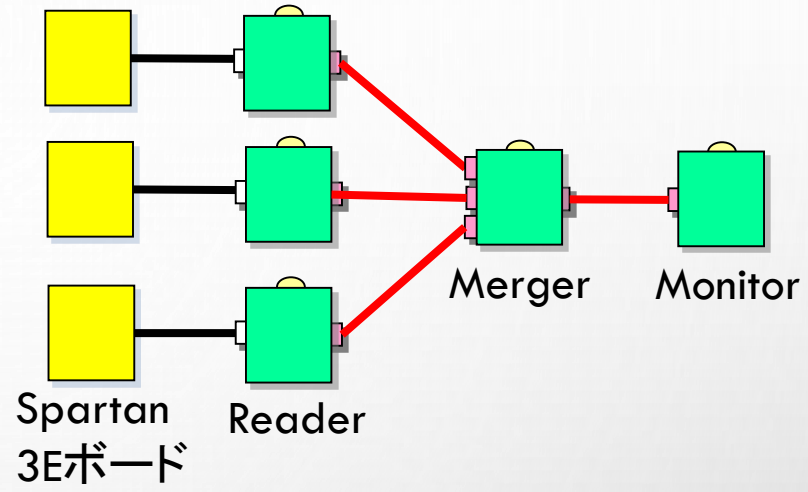
例1 Readerでepoll等を利用して多重読み込みを行う

(メリット)

- コンポーネントが少ないので使用するリソースが少なくても済む

(デメリット)

- Readerの作成が難しい
- Readerの負荷が大きい



例2 複数のReaderとMergerを利用する

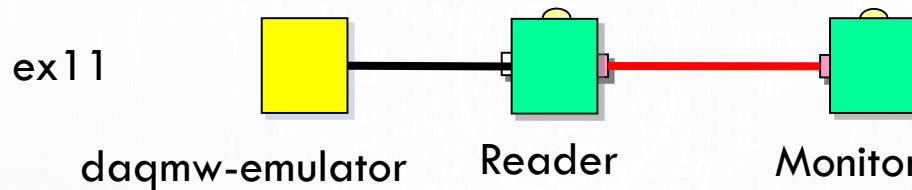
(メリット)

- Readerは全て1台の読み出しなので簡単に作れる。
- Readerの負荷を分散できる

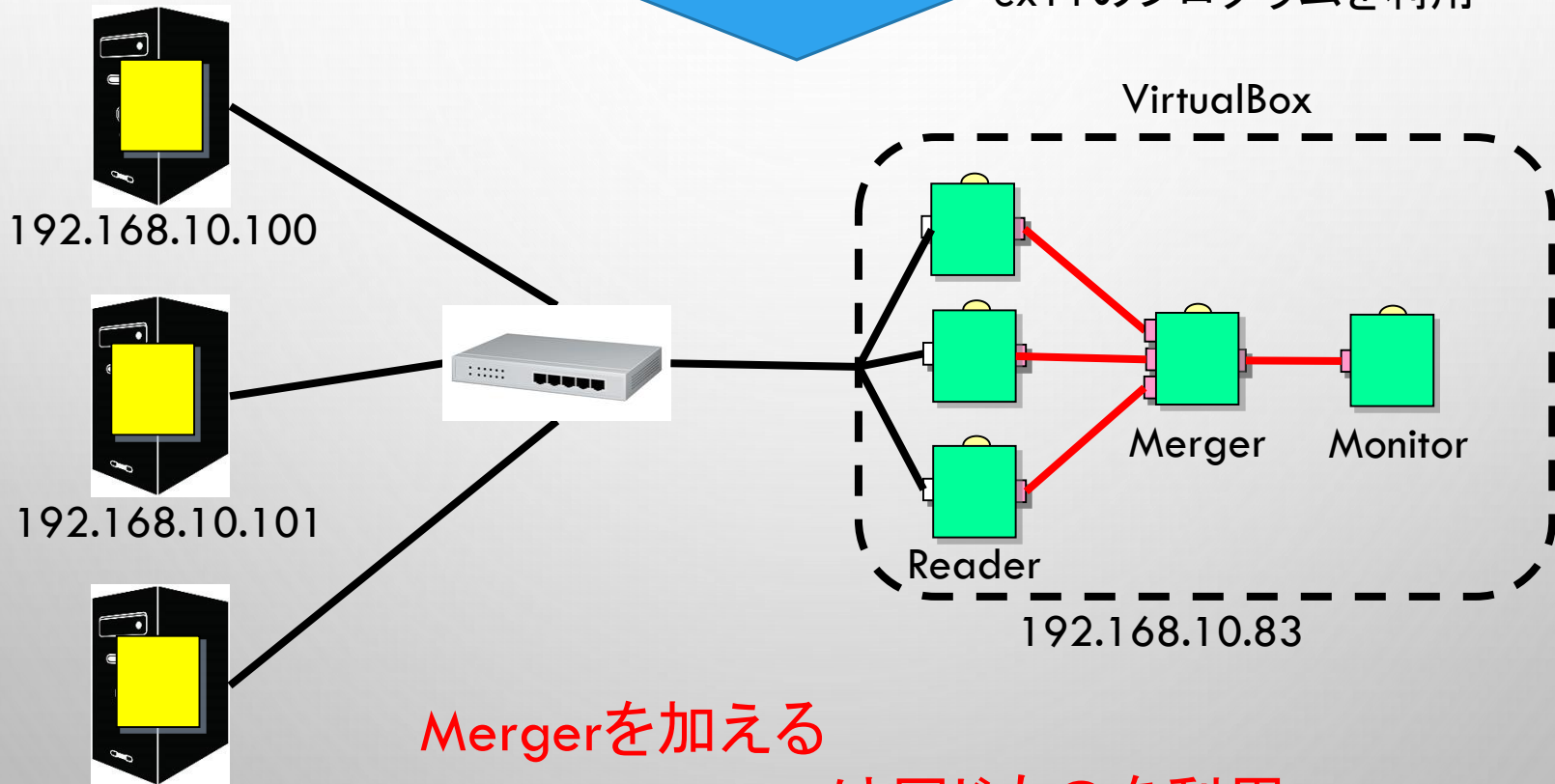
(デメリット)

コンポーネントが多いので使用するリソースが多くなる

ex16 Mergerを利用して複数台のPCからデータを収集する



ex11のプログラムを利用



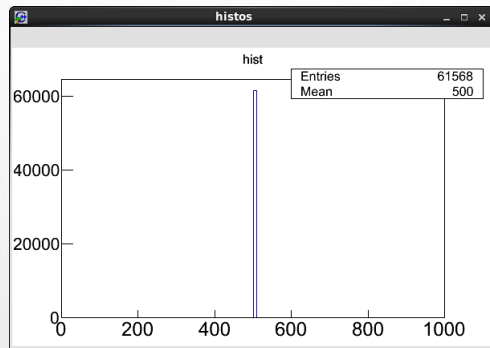
Mergerを加える

Reader、Monitorは同じものを利用

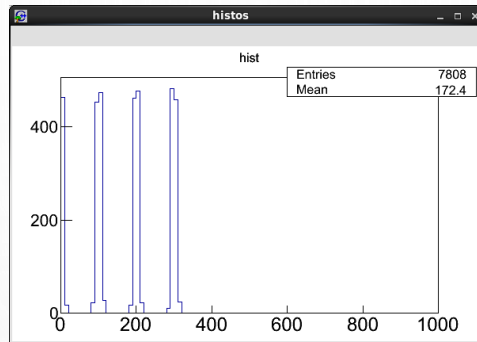
192.168.10.102

ex16 Mergerを利用して複数台のPCからデータを収集する

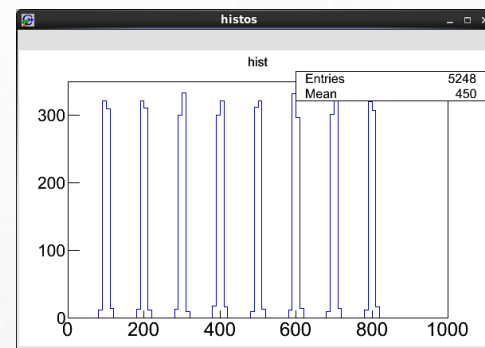
192.168.10.100



192.168.10.101



192.168.10.102



+

+

ALL

=

