

EPICS Stream Device

佐藤政則 (KEK 加速器研究施設)

Stream Device とは (1)

- スイスの研究所 PSI (Paul Scherrer Institute) にて開発された.

- Author : Dirk Zimoch

- Web

<http://epics.web.psi.ch/software/streamdevice/>

EPICS: EPICS R3.13.10, R3.14.8, R3.14.12, R7.0.3

サポート OS: Linux, Windows, VxWorks, ~~cygwin, solaris~~

Stream Device とは (2)

- RS-232C, GPIB, ソケット通信などを用いて, コマンド文字列で制御するデバイス用のEPICS IOCを作成する仕組み. (asynDriverを使いやすくするためのツール.)
- SCPIデバイスをEPICSで制御することに適している.
 - SCPI (Standard Commands for Programmable Instruments): オシロスコープなどの測定をPCなどの計算器を使って制御する際のコマンドに関する国際規格
 - SCPIは”:WAV:DATA?;”といったコマンドの文法と、”*IDN?”などの標準コマンドを規定している.
- コマンド文字列は, protocolファイルで定義する.
- フォーマット形式は, C言語のフォーマット指定書式に似ている.

asynDriver

- デバイスとの通信を非同期化するためのモジュール.
- 同期通信の場合, 応答があるまでIOC (プログラム) が待ち状態になるため, その間処理が進まなくなる. 効率が悪い.
- 非同期通信を使用することで, デバイスとの通信中に他の処理を進めることができる. 効率が良い.

なぜStream Deviceを使うのか？

- 簡単なEPICS IOCであっても，ゼロから新規に作るのは手間がかかり，初心者にはハードルが高い。
 - C言語でのプログラミングの知識
 - 開発環境の知識
 - EPICSライブラリ関数の知識
- EPICS IOCの作成は，経験者であっても手間がかかり，デバッグも簡単ではない。
- asynDriverを直接利用する，あるいはdevGpibやAsynPortDriverなどを使用する方法もあるが，手間がかかりわかりにくい。
- 測定器とのコマンド通信は，シリアル，GPIB，ソケット通信でASCII文字列をやり取りすることが多い．Stream deviceを使用すれば，単純なコマンドであれば短時間で作成可能である。

利点と欠点

• 利点

- 導入のためのハードルが比較的低い。
- protocolファイルの修正後に再コンパイルが不要であり，効率的な開発が可能である。
- C言語の知識をあまり必要としない。（あればなお良い）
- ユーザーがやるべきことは，dbファイルとprotocolファイルの編集のみ。protocolファイルにデバイス制御用コマンドを定義する。

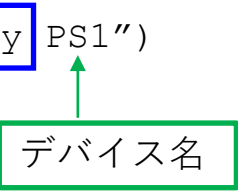
• 欠点

- ASCII文字列であっても複雑なデータを扱うのは困難である。
- 1つのコマンドから，複数のデータに分割するのは難しい。
- 文字列のパースが遅い。
 - 処理速度の速いサーバーなら問題ないが，貧弱な組み込みCPUでは問題になる。
- binaryの取り扱いは，単純なもの以外は難しい。

dbファイルとprotocolファイルの関係

stream.db

```
record(ai, "$(user):freq:get") {  
  field(DTYP, "stream")  
  field(INP, "@stream.proto getFrequency PS1")  
}  
record(ao, "$(user):freq:set") {  
  field(DTYP, "stream")  
  field(OUT, "@stream.proto setFrequency PS1")  
}
```



stream.proto

```
Terminator = CR LF;  
getFrequency {  
  out "FREQ?";  
  in "%f";  
}  
setFrequency {  
  out "FREQ %f";  
}
```

dbでのStreamDeviceの設定

- DTYPは"stream"
- INP/OUTは"@filename function device"
- デバイスの定義 (この例では, PS1という名前)

st.cmd (スタートアップスクリプト) にて,

```
drvAsynIPPortConfigure("PS1", "192.168.15.100:9999", 0, 0, 0)
```

protocolファイルの設定

- function {
 command "string";
}
- "%~"がレコードのVALに対応する。

drvAsynIPPortConfigure

- `drvAsynIPPortConfigure(portName, hostInfo, priority, noAutoConnect, noProcessEos)`
 - `portName`: asyn port名
 - `hostInfo`: IPアドレス:ポート番号 (文字列)
 - `priority`: asyn プロセスのpriority(整数)
 - `noAutoConnect`: 1の時自動再接続しない。
 - `noProcessEos`: Eos (End of String)文字の処理をする(0)しない(1)

protocolファイル設定

Command

- In 入力
- out 出力
- wait 待ち

format

- double(%f, %e,%g)
- long(%d, %u, %i)
- 8進数(%o)
- 16進数(%x, %X)
- string, char(%s,%c)
- enum(%{str0|str1|..})
- raw long (%r)

System

- Terminator 入出力ターミネータ
- OutTerminator 出力ターミネータ
- InTerminator 入力ターミネータ
- WriteTimeout 書込タイムアウト
- ReadTimeout 読込タイムアウト
- ReplyTimeout 応答タイムアウト
- MaxInput 入力最大長
- Separator 分割文字列
- ExtraInput Error/Ignore

例外

- @mismatch フォーマットエラー
- @writetimeout 書込タイムアウト
- @replytimeout 応答タイムアウト
- @readtimeout 読込タイムアウト
- @init 初期化

実習

- 目的：波形発生器 (Function Generator; FG) シミュレータをStream Deviceで制御する.
- 手順：
 - FGシミュレータにtelnetログインし，動作試験（コマンド送受信）をおこなう.
 - templateからIOCを作成する.
 - stream device用のprotocol file, これに対応するdatabaseファイルを編集してコマンドを実装する.
 - まずは, *IDN? コマンドを実装する.
 - 他のコマンドも実装する.
 - medmあるいはCS-Studioで電圧(VOLT?応答)を表示し，横軸時間，縦軸電圧のグラフを描く.
- 詳細手順

<http://kekb-co-gitlab.kek.jp/epics-lecture/epics-lecture-2023/-/blob/main/docs/day2-stream/stream>

参考文献

- PSI stream device: <http://epics.web.psi.ch/software/streamdevice/>
- Asyn: <http://www.aps.anl.gov/epics/modules/soft/asyn/>
- T. Obina, “Stream Device のすすめ”.
https://cerldev.kek.jp/trac/EpicsUsersJP/raw-attachment/wiki/epics/streamdevice/Intro_StreamDevice.pdf
- 路川徹也, “EPICS StreamDevice”, EPICS Training 2018 KEK.
https://cerldev.kek.jp/trac/EpicsUsersJP/raw-attachment/wiki/intro/20181101_KEK/ET2018_StreamDevice_20181029.pdf
- Noboru Yamamoto, “SCPI装置をEPICS-StreamDeviceで制御する”, 2020.4.16 (更新 : 2022.6.13) .
<http://j-parc.jp/ctrl/documents/articles/StreamDevice/SCPI%20equipment%20with%20EPICS-StreamDevice.pdf>
- Mailing List : epics-users@ml.post.kek.jp