

実用EPICS入門-2

技術部専門講習

飛山真理

帯名崇

EPICSデータベースの実際

- よく使うEPICSレコードのタイプ
 - ai ao (analog input / output)
 - bi bo (binary input / output)
 - calc (calculation)
 - fanout (trigger signalをfanoutする)
 - longin longout (long input / output)
 - mbbi mbbo (multibit binary input / output)
 - mbbiDirect mbboDirect
 - stringin stringout (string input / output)
 - waveform(配列)
 - subarray(配列の一部)

このほかにも

- EPICSシステムにはじめから用意されているもの
 - eg, event, scan, waitなど
 - コントロールグループなどにより独自に作成、拡張された物
 - bpm record, compactSubarrayなど
- などがあります。

参考: EPICS Record Reference Manual

準備

- st.cmd内の不要レコードをコメントアウト
 - 今の状態でIOC shellを立ち上げ、dbfで見ると
 - sluserHost:aiExample
 - sluserHost:aiExample1
 - sluserHost:aiExample2
 - sluserHost:aiExample3
 - sluserHost:calcExample
 - sluserHost:calcExample1
 - sluserHost:calcExample2
 - sluserHost:calcExample3
 - sluserHost:compressExample
 - sluserHost:subExample
 - sluserHost:xxxExample

st.cmdを編集

```
##Load record instances
```

```
dbLoadTemplate "db/userHost.substitutions"
```

```
dbLoadRecords("db/dbSubExample.db","user=sluserHost")
```

上記2行をコメントアウト(頭に#を付ける)

```
##dbLoadTemplate "db/userHost.substitutions"
```

```
##dbLoadRecords("db/dbSubExample.db","user=sluserHost")
```

とりあえずaiレコード(ソフトレコード)を 作ってみる

- 名前: 28文字以内の文字列
 - 大文字小文字区別あり
 - もちろん2byte codeは使用不可
 - .(ドット)もだめ
 - 意味がある名前にすべき
 - (システム全体で)規則性があった方がよい
 - 全コントロールシステム内で唯一の名前に

データベース作成

- ターミナルを開き、epicsApp/trainApp/Dbへディレクトリ移動
- editorで、test01.dbというファイルを作成

test01.dbの中身

```
record(ai, "TDL:TEST:A1") {  
    field(DESC, "analog input record")  
    field(SCAN, "Passive")  
    field(DTYP, "Soft Channel")  
    field(FLNK, "0.0")  
    field(INP, "0.0")  
    field(PREC, "5")}
```


Makefileの編集

- DB += dbExample1.dbから
- DB += userHost.substitutions
までをコメントアウト
DB += test01.db
追加、file保存
- ターミナルからmake

IOC shellの起動

- st.cmdファイルを編集
 - 先ほどコメントアウトしたところの近くに
`dbLoadRecords("db/test01.db")`
追加
- ./st.cmdとしてIOC shell起動
 - dbコマンドで、データベース確認

作ったレコードにアクセスする

- ターミナルをひらき
 - caget TDL:TEST:A1
 - 別のターミナルを開き camonitor TDL:TEST:A1
 - caput TDL:TEST:A1 100.0
- IOC shellから
 - dbl
 - dbpr "TDL:TEST:A1",5
 - dbgrep "TDL*"

epics> dbpr "TDL:TEST:A1",5

ACKS: NO_ALARM	ACKT: YES	ADEL: 0	ALST: 5
AOFF: 0	ASG:	ASLO: 1	ASP: 0x00000000
BKPT: 0x00	DESC: analog input record	DISA: 0	
DISP: 0	DISS: NO_ALARM	DISV: 1	DPVT: 0x00000000
DSET: 0xac816c00	DTYP: Soft Channel	EGU:	EGUF: 0
EGUL: 0	EOFF: 0	ESLO: 1	EVNT: 0
FLNK:CONSTANT 0	HHSV: NO_ALARM	HIGH: 0	HIHI: 0
HOPR: 0	HSV: NO_ALARM	HYST: 0	INIT: 0
INP:CONSTANT 0.0	LALM: 5	LBRK: 0	LCNT: 0
LINR: NO CONVERSION	LLSV: NO_ALARM	LOLO: 0	LOPR: 0
LOW: 0	LSET: 0x48e11308	LSV: NO_ALARM	MDEL: 0
MLIS: 0x000000000000000000000000		MLOK: 0x88e01308	MLST: 5
NAME: TDL:TEST:A1	NSEV: NO_ALARM	NSTA: NO_ALARM	ORAW: 0
PACT: 0	PBRK: 0x00000000	PHAS: 0	PINI: NO
PPN: 0x00000000	PPNR: 0x00000000	PREC: 5	PRI0: LOW
PROC: 0	PUTF: 0	RDES: 0xc81a1408	ROFF: 0
RPRO: 0	RSET: 0x8063cb00	RVAL: 0	SCAN: Passive
SDIS:CONSTANT	SEVR: NO_ALARM	SIML:CONSTANT	SIMM: NO
SIMS: NO_ALARM	SIOL:CONSTANT	SM00: 0	SPVT: 0x00000000
STAT: NO_ALARM	SVAL: 0	TIME: 2007-11-06 08:23:51.214915000	
TPRO: 0	TSE: 0	TSEL:CONSTANT	UDF: 0
VAL: 5			

epics>

EPICSレコードの構造

- 1つのレコードの中には多くのfieldがある
 - 意識する必要がある物は多くない
- 全てのrecordにあるもの
 - **NAME** DESC ASG **SCAN** PINI PHAS EVNT
TSE TSEL **DTYP** DISV DISA SDIS MLOK
MLIS DISP **PROC** STAT **SEVR** NSTA NSEV
ACKS ACKT DISS LSET LCNT PACT PUTF
RPRO ASP PPN PPNN SPVT RSET DSET
DPVT **PRIO** TPRO BKPT UDF **TIME FLNK**

- 多くのinputもののrecordにあるもの
 - INP DTYP RVAL VAL SIMM SIML SVAL
SIOL SIMS
- 多くのoutputもののrecordにあるもの
 - OUT DTYP VAL OVAL RVAL RBV DOL
OMSL OIF SIMM SIML SIOL SIMS IVOA
IVOV

フィールドにアクセスするには

- `caget TDL:TEST:A1.SCAN`
- `caput TDL:TEST:A1.HIGH 100.0`

名前置き換え機能

- test01.dbを以下のように変更

```
record(ai, "$ (USER) :TEST:A1") {  
    field(DESC, "analog input record")  
    field(SCAN, "Passive")  
    field(DTYP, "Soft Channel")  
    field(FLNK, "0.0")  
    field(INP, "0.0")  
    field(PREC, "5")}
```
- ・ 忘れずにmake

st.cmdで名前を決定

st.cmdを編集

```
dbLoadRecords("db/test01.db","USER=TDL")
```

IOC schellを立ち上げて、データベース名を確認

aiレコードを2つ作り、リンクする-1

- TDL:TEST:A1とTDL:TEST:A2
 - どちらもaiレコード、cut-pasteで作る
- TDL:TEST:A2のINPフィールドを
 - field(INP,"TDL:TEST:A1")
 - あるいはfield(INP,"\$(USER):TEST:A1")
- これでTDL:TEST:A1の値を変えてみる
 - TDL:TEST:A2をモニターすると？
 - TDL:TEST:A2をプロセスすると？
 - caput TDL:TEST:A2.PROC 1
 - TDL:TEST:A2のSCANフィールドを2 second にしてみる
 - caput TDL:TEST:A2.SCAN 5

FLNKでリンクする

- TDL:TEST:A1のFLNKフィールドをA2にする
 - field(FLNK,"TDL:TEST:A2")
 - あるいは
 - field(FLINK,"\$(USER):TEST:A2")
- これで、TDL:TEST1:A1の値を変え、A2をモニターする

longinレコード

```
record(longin,"$(USER):TEST:L1"){  
    field(DESC,"long input record")  
    field(SCAN,"Passive")  
    field(DTYP,"Soft Channel")  
    field(FLNK,"0.0")  
    field(INP,"0.0")  
}
```

long型整数、-2147483648～2147483647

biレコード

```
record(bi,"$(USER):TEST:B1"){  
  field(DESC,"binary input record")  
  field(SCAN,"Passive")  
  field(DTYP,"Soft Channel")  
  field(FLNK,"0.0")  
  field(INP,"0.0")  
  field(ZNAM,"OFF")  
  field(ONAM,"ON")  
}
```

binary入力レコード、ON/OFF等制御によく用いる
値は0か1かだけ

演習

- longin、biレコードを作って動作を確認する
 - 範囲外の値を入れたときの動作は？

calcレコード

```
record(calc,"$(USER):TEST:C1"){  
  field(DESC,"calculation record")  
  field(SCAN,"Passive")  
  field(FLNK,"0.0")  
  field(CLAC,"A*0.5-B")  
  field(INPA,"$(USER):TEST:A1.VAL NPP MS")  
  field(INPB,"$(USER):TEST:A2.VAL NPP MS")  
  field(INPC,"0.0")  
  field(INPD,"0.0")  
  field(INPE,"0.0")  
  field(INPF,"0.0")  
  field(INPG,"0.0")  
  field(INPH,"0.0")  
  field(INPI,"0.0")  
  field(INPJ,"0.0")  
  field(INPK,"0.0")  
  field(INPL,"0.0")}
```

calcレコード(続き)

- CALCの計算で使える表現
(fieldがRPCLなら逆ポーランド記法になる)
 - A～L
 - RNDM (0～1の乱数)
 - ABS、SQR、MIN、MAX、CEIL、FLOOR、LOG、LOGE、EXP
 - ^、**、+、-、*、/、%、NOT
 - SIN、SINH、ASIN、COS、COSH、ACOS、TAN、TANH、ATAN
 - >=、>、<=、<、#(not equal to)、=
 - &&、||、!
 - |、&、OR、AND、XOR、~(complement)、<<、>>

calc計算の例

- $A+B$
A+Bになる
- $(A+B)<(C+D)$
これが成立するなら1、しないなら0
- $(A+B)<(C+D)?E:F$
これが成立するならE、しないならF
- $A\&B$
A、Bいずれもまず整数に変換し、 $A\&B$ のbit演算をし、結果をfloatingに変換する。

演習

1. 1秒ごとに1ずつ増え、50になったら0に戻り動作を続けるcalcレコードを作る
2. このレコードをbiレコードが1のときだけcountするようにする
3. さらに、biレコードでリセットできるようにする

1 の例

```
record(calc,"$(USER):TEST:C1){  
  field(DESC,"calculation record")  
  field(SCAN,"1 second")  
  filed(CALC,"(A<50)?A+1:0")  
  field(INPA,"$(USER):TEST:C1 NPP MS")}
```

2の例

```
record(calc,"$(USER):TEST:C1){  
  field(DESC,"calculation record")  
  field(SCAN,"1 second")  
  filed(CALC,"B?((A<50)?A+1:0):A")  
  field(INPA,"$(USER):TEST:C1 NPP MS")  
  field(INPB,"$(USER):TEST:B1 NPP MS")}
```

calcレコードのtip

- あんまり複雑な計算をさせない方が吉
 - 見通しが悪くない程度までの計算
- epicsバージョン、実装によってはバグがある
 - 超常現象がおきたときは、悩まずコントロール専門家に相談

Fanout

- 6個までの他のrecordをprocessする

```
record(fanout,"$(USER):TEST:F1"){  
  field(DESC,"fanout record")  
  field(SCAN,"Passive")  
  field(LNK1,"$(USER):TEST:A1")  
  field(LNK2,"$(USER):TEST:A2")  
  field(LNK3,"0.0")  
  field(LNK4,"0.0")  
  field(LNK5,"0.0")  
  field(LNK6,"0.0")}
```

EPICSのtrigger信号

- fanout以外のレコードは、FLNK出来る先は1つだけ
 - いくつかのレコードをtriggerしたければ、fanoutを使うか、seriesにflinkするか
 - flinkはstackを食うので、あまり長いとうましくない
- triggerされるレコードは、いくつでもtrigger信号を受け付ける
 - scanがpassiveのときのみ

mbbiレコード

```
record(mbbi,"$(USER):TEST:MB1"){  
  field(DESC,"mbbi record")  
  field(SCAN,"Passive")  
  field(DTYPE,"Soft Channel")  
  field(INP,"0.0")  
  field(FLNK,"0.0")  
  field(ZRVL,"0")..field(FFVL,"0.0")  
  field(ZRST,"")..field(FFST,"")}
```

binary valueを(最大)16ステートにする

演習

- mbbiレコードを作って、動作を確認する
 - calcレコードで0～15までの値をつくり、mbbiレコードにfInk
 - mbbiレコードをモニターして、値がどうなるか見る

mbbiDirect

```
record(mbbiDirect,"$(USER):TEST:MD") {  
  field(DESC,"mbbiDirect record")  
  field(SCAN,"Passive")  
  field(DTYP,"Soft Channel")  
  field(FLNK,"0.0")  
  field(INP,"0.0")  
  field(B0,"")  
  field(B1,"")..field(BF,"")}
```

mbbiと違い、biが16個まとまったようなもの。各ビット操作できる。

mbbiDirect操作

- bitごとの操作

caput TDL:TEST:MD.B0 1

caget TDL:TEST:MD.BF

- 全体一気操作

caput TDL:TEST:MD 255

caget TDL:TEST:MD

演習

- mbbiDirectをつくり(scan passive)、値を入れてみてbit patternがどうなるか見る
- calcレコードで0～15の値を1秒ごとに更新させ、これをmbbiレコードとmbbiDirectレコードに入れ、結果をそれぞれモニターする
 - fanoutを使ってみる
 - あるいは、flnkでつないでみる

例1

```
record(calc,"$(USER):TEST:C1"){  
    field(SCAN,"1 second")  
    field(FLNK,$(USER):TEST:FAN")  
    field(CALC,"(A<15)?A+1:0")  
    field(INPA,"$(USER):TEST:C1 NPP MS")}  
record(fanout,"$(USER):TEST:FAN"){  
    field(LNK1,"$(USER):TEST:MB1")  
    field(LNK2,"$(USER):TEST:MD")}  
record(mbbi,"$(USER):TEST:MB1"){  
    field(INP,"$(USER):TEST:C1.VAL NPP MS")}  
record(mbbiDirect,"$(USER):TEST:MD"){  
    field(INP,"$(USER):TEST:C1.VAL NPP MS")}
```

例2

```
record(calc,"$(USER):TEST:C1"){  
    field(SCAN,"1 second")  
    field(FLNK,$(USER):TEST:MB1")  
    field(CALC,"(A<15)?A+1:0")  
    field(INPA,"$(USER):TEST:C1 NPP MS")}  
record(mbbi,"$(USER):TEST:MB1"){  
    field(FLNK,"$(USER):TEST:MD.VAL")  
    field(INP,"$(USER):TEST:C1.VAL NPP MS")}  
record(mbbiDirect,"$(USER):TEST:MD"){  
    field(INP,"$(USER):TEST:C1.VAL NPP MS")}
```

次回は

- medmを使った簡易表示
- データのarchive
- データセーブをお忘れ無く