

# EPICS セミナー： EPICS Database の基礎

S. Sasaki (KEK, Accelerator Division 4)

2018/11/01(木) 13:00 - 14:30

## 概要

EPICS 入門セミナー第 1 日目; 第 4 コマ。IOC がどのようなレコードを持ち、それがどのように振る舞うかは EPICS Database 上で管理される。ここでは、EPICS Database の基礎とその構文に関して実習する。

## 目次

1	softloc コマンドの使い方	2
2	仮想電源 IOC の EPICS Database	2
2.1	作成するレコード	2
2.2	EPICS Database の作成	2
2.3	softIoc の起動	5
3	課題	7

## 1 softloc コマンドの使い方

ハードウェア I/O の無い IOC は soft IOC と呼ばれる。本実習では softIoc コマンドで soft IOC を立ち上げ、EPICS Database について学ぶ。softIoc コマンドは EPICS Base に含まれる実行ファイルであり、単純な soft IOC であれば、このコマンドで簡単に起動することができる。softIoc コマンドを `-h` オプションと共に実行すると、以下の通りヘルプが表示される。

```
$ softIoc -h
Usage: softIoc [-D softIoc.dbd] [-h] [-S] [-a ascf]
           [-m macro=value,macro2=value2] [-d file.db]
           [-x prefix] [st.cmd]
```

ヘルプにある通り、`-d` で起動する softIoc のレコードデータベースを指定して起動することができる。起動時に実行するスタートアップスクリプト (`st.cmd`) も指定できる。`-m` ではマクロの定義が可能。`-x` を指定すると `prefix:exit` というレコードが作成され、このレコードに 1 を put すると IOC が停止する。

## 2 仮想電源 IOC の EPICS Database

ここでは電源をシミュレートした EPICS Database の作成を通して、EPICS Database に関して学ぶ。

### 2.1 作成するレコード

表 1 に示すレコードを作成していく。図 1 にレコードのリンクを図示したものを示す。

表 1 仮想電源のレコード

タイプ	レコード名	役割	備考
ao	\$(head):PS.SIM:SET_CURRENT	電流設定値の書き込み	0.0~100.0 のレンジ
ai	\$(head):PS.SIM:SET_CURRENT_RB	電流設定値の読み出し	
calc	\$(head):PS.SIM:CURRENT	電流値 (擬似電流の値)	設定値を中心に値をランダムに上下
ai	\$(head):PS.SIM:READ_CURRENT	電流値の読み出し	
bo	\$(head):PS.SIM:POWER_SW	ON/OFF スイッチ	1 で ON、0 で OFF とする
bi	\$(head):PS.SIM:POWER_SW_RB	ON/OFF 状態の読み出し	

### 2.2 EPICS Database の作成

はじめに作業用のディレクトリを作成し、移動する。

```
$ mkdir -p ~/epics/app/pssim
$ cd ~/epics/app/pssim
```

powerSupply.db ファイルを作成し、ソースコード 1 の通りに記述する。レコードの定義の仕方、フィール

## Power Supply Simulate EPICS Database

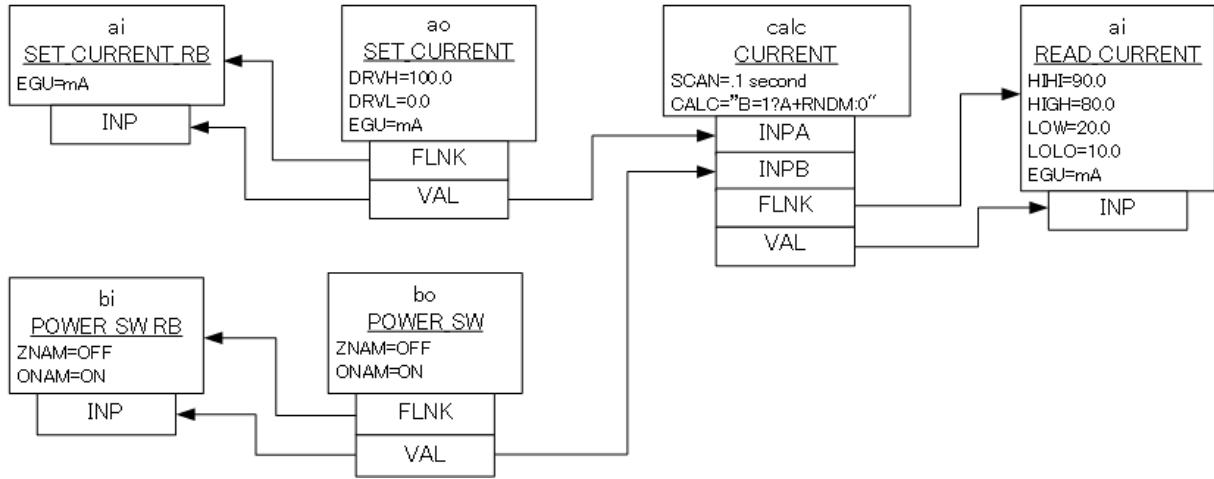


図1 仮想電源の EPICS Database

ドの設定方法などを確認すること。厳密な文法に関して知りたい場合には [1] を参照すること。また、代表的なレコードタイプやフィールドは別紙の EPICS quick reference を参照すること。

ソースコード 1 powerSupply.db

```

1 # Record Definition Syntax
2 # ===
3 # record(record_type, record_name) {
4 # field(field_name, "value")
5 # ...
6 # #comment
7 # }
8
9 record(ao, "$(head):PS_SIM:SET_CURRENT")
10 {
11   field(DESC, "Set the current value") # Description
12   field(SCAN, "Passive") # Scanning Rate
13   field(VAL, "0.0") # Value
14   field(DRVH, "100.0") # Drive High
15   field(DRVL, "0.0") # Drive Low
16   field(HOPR, "100.0") # High Operating Range
17   field(LOPR, "0.0") # Low Operating Range
18   field(PREC, "1") # Display Precision
19   field(EGU, "mA") # Engineering Units
20   field(FLNK, "$(head):PS_SIM:SET_CURRENT_RB PP NMS") # Forward Link
21 }
22

```

```

23 record(ai, "$(head):PS_SIM:SET_CURRENT_RB")
24 {
25     field(DESC, "Read back set the current value")
26     field(SCAN, "Passive")
27     field(INP, "$(head):PS_SIM:SET_CURRENT") # Input Link
28     field(HOPR, "100.0")
29     field(LOPR, "0.0")
30     field(PREC, "1")
31     field(EGU, "mA")
32 }
33
34 record(calc, "$(head):PS_SIM:CURRENT")
35 {
36     field(DESC, "The current value")
37     field(SCAN, ".1 second")
38     field(CALC, "B=1?A+RNDM:0") # If B=1, is generated
39     #field(CALC, "B=1?VAL+(A-VAL)*0.1+RNDM:0") # If B=1, is generated
40     field(INPA, "$(head):PS_SIM:SET_CURRENT")
41     field(INPB, "$(head):PS_SIM:POWER_SW")
42     field(PREC, "1")
43     field(FLNK, "$(head):PS_SIM:READ_CURRENT PP NMS") # Forward Link
44 }
45
46 record(ai, "$(head):PS_SIM:READ_CURRENT")
47 {
48     field(DESC, "Read the current value")
49     field(SCAN, "Passive")
50     field(INP, "$(head):PS_SIM:CURRENT")
51     field(HIHI, "90.0") # Highest Alarm Limit
52     field(HIGH, "80.0") # High Alarm Limit
53     field(LOW, "20.0") # Low Alarm Limit
54     field(LOLO, "10.0") # Lowest Alarm Limit
55     field(HOPR, "100.0")
56     field(LOPR, "0.0")
57     field(PREC, "1")
58     field(EGU, "mA")
59 }
60
61 record(bo, "$(head):PS_SIM:POWER_SW")
62 {
63     field(DESC, "The power switch")
64     field(SCAN, "Passive")
65     field(VAL, "0")
66     field(ZNAM, "OFF") # Zero Name
67     field(ONAM, "ON") # One Name
68     field(FLNK, "$(head):PS_SIM:POWER_SW_RB PP NMS")

```

```

69 }
70
71 record(bi, "$(head):PS_SIM:POWER_SW_RB")
72 {
73   field(DESC, "Read back the power switch")
74   field(SCAN, "Passive")
75   field(INP, "$(head):PS_SIM:POWER_SW")
76   field(ZNAM, "OFF")
77   field(ONAM, "ON")
78 }

```

## 2.3 softloc の起動

soft IOC を起動して、作成した Database をロードする。マクロには ET\_kektaro のように指定して起動する。

```

$ softIoc -m user=ET_kektaro -d powerSupply.db
Starting iocInit
#####
## EPICS R3.15.6
## EPICS Base built Oct 18 2018
#####
epicsThreadRealtimeLock Warning: Unable to lock the virtual address space.
VM page faults may harm real-time performance.
iocRun: All initialization complete
epics>

```

望んだレコードが作成されているか dbl コマンド (database list) で確認する。

```

epics> dbl
ET_kektaro:PS_SIM:POWER_SW_RB
ET_kektaro:PS_SIM:POWER_SW
ET_kektaro:PS_SIM:SET_CURRENT
ET_kektaro:PS_SIM:SET_CURRENT_RB
ET_kektaro:PS_SIM:READ_CURRENT
ET_kektaro:PS_SIM:CURRENT

```

dbpr コマンド (database print record) でレコードのフィールドの値を確認する。

```

epics> dbpr ET_kektaro:PS_SIM:SET_CURRENT
ASG:          DESC: Set the current value          DISA: 0
DISP: 0       DISV: 1          NAME: ET_kektaro:PS_SIM:SET_CURRENT
OMOD: 0       OVAL: 0          RBV: 0             RVAL: 0
SEVR: NO_ALARM  STAT: UDF      TPRO: 0             VAL: 0

```

レコード名の後に 0~4 の数値を指定すると、どこまで詳細にフィールドを表示するかを指定できる。

```

epics> dbpr ET_kektaro:PS_SIM:SET_CURRENT 2
ACKS: NO_ALARM  ACKT: YES          ADEL: 0          AOFF: 0
ASG:          ASLO: 0          BKPT: 00
DESC: Set the current value          DISA: 0          DISP: 0
DISS: NO_ALARM  DISV: 1          DOL:CONSTANT    DRVH: 100
DRVL: 0        DTYP: Soft Channel  EGU: mA          EGUF: 0
EGUL: 0        EOFF: 0          ESLO: 1          EVNT:
FLNK:DB_LINK ET_kektaro:PS_SIM:SET_CURRENT_RB  HHSV: NO_ALARM
HIGH: 0        HIHI: 0          HOPR: 100       HSV: NO_ALARM
HYST: 0        IVOA: Continue normally  IVOV: 0
LCNT: 0        LINR: NO CONVERSION  LLSV: NO_ALARM  LOLO: 0
LOPR: 0        LOW: 0          LSV: NO_ALARM   MDEL: 0
NAME: ET_kektaro:PS_SIM:SET_CURRENT  NSEV: NO_ALARM  NSTA: NO_ALARM
OIF: Full      OMOD: 0          OMSL: supervisory  OROC: 0
OUT:CONSTANT   OVAL: 0          PACT: 0          PHAS: 0
PINI: NO       PREC: 1          PRIO: LOW        PUTF: 0
RBV: 0        ROFF: 0          RPRO: 0          RVAL: 0
SCAN: Passive  SDIS:CONSTANT   SDLY: -1         SEVR: NO_ALARM
SIML:CONSTANT  SIMM: NO        SIMS: NO_ALARM   SIOL:CONSTANT
SSCN: <nil>    STAT: UDF      TIME: <undefined>  TPRO: 0
TSE: 0        TSEL:CONSTANT   UDF: 0           UDFS: INVALID
VAL: 0

```

続いて、dbpf コマンド (database put field) でレコードに値を put する。ここでは電流の設定を 30 mA に設定する。

```

epics> dbpf ET_kektaro:PS_SIM:SET_CURRENT 30
DBF_DOUBLE:    30

```

dbpr コマンドで値が更新されたか確認する。VAL フィールドと、TIME フィールドが更新されたことが分かる。

```

epics> dbpr ET_kektaro:PS_SIM:SET_CURRENT 2
ACKS: NO_ALARM      ACKT: YES          ADEL: 0           AOFF: 0
ASG:                 ASLO: 0           BKPT: 00
DESC: Set the current value      DISA: 0           DISP: 0
DISS: NO_ALARM      DISV: 1           DOL:CONSTANT     DRVH: 100
DRVL: 0             DTYP: Soft Channel EGU: mA          EGVF: 0
EGUL: 0             EOFF: 0           ESLO: 1           EVNT:
FLNK:DB_LINK ET_kektaro:PS_SIM:SET_CURRENT_RB      HHSV: NO_ALARM
HIGH: 0             HIHI: 0           HOPR: 100        HSV: NO_ALARM
HYST: 0             IVOA: Continue normally      IVOV: 0
LCNT: 0             LINR: NO CONVERSION LLSV: NO_ALARM   LOLO: 0
LOPR: 0             LOW: 0            LSV: NO_ALARM    MDEL: 0
NAME: ET_kektaro:PS_SIM:SET_CURRENT      NSEV: NO_ALARM   NSTA: NO_ALARM
OIF: Full           OMOD: 0           OMSL: supervisory OROC: 0
OUT:CONSTANT        OVAL: 30          PACT: 0           PHAS: 0
PINI: NO            PREC: 1           PRIO: LOW         PUTF: 0
RBV: 0             ROFF: 0           RPRO: 0           RVAL: 30
SCAN: Passive       SDIS:CONSTANT     SDLY: -1          SEVR: NO_ALARM
SIML:CONSTANT       SIMM: NO          SIMS: NO_ALARM    SIOL:CONSTANT
SSCN: <nil>         STAT: NO_ALARM    TIME: 2018-10-29 13:52:31.645669500
TPRO: 0            TSE: 0           TSEL:CONSTANT     UDF: 0
UDFS: INVALID       VAL: 30

```

その他、代表的なコマンドに関しては別紙の EPICS quick reference を参照すること。

### 3 課題

1. caget、caput、camonitor を利用して、電流値の設定や、スイッチの ON/OFF、電流値のモニターを行う
2. CSS で制御用の OPI を作成する
3. 電流値が少しずつ時間をかけて設定値に近づくようにする
  - calc レコードの CALC フィールドを変更すれば電流値の計算方法を変更することができる
  - 追従率を設定する ao レコードを新規作成すれば、追従の振る舞いを変更することができる
  - CALC フィールドの文法は [2] を参照
4. Busy 状態（電流値が設定値に達するまでを Busy とする）を表すレコードを作成する
  - busy 状態かどうかを計算する calc レコードを新規に作成する
  - 電流値と設定値の差がある値以上（例えば 3mA 以上）の場合には 1、それ以下の場合には 0 となるようにする
5. 電流値がある値以上（例えば 80mA 以上）になった場合に、スイッチが自動的に OFF になるようにする

- ある値以上になったかどうかを計算し、スイッチ入力を 0 にする calcout レコードを作成
- 計算値が 0 の時だけ 0 を put するには OOPT フィールドを使用する

## 参考文献

- [1] <https://epics.anl.gov/base/R3-14/12-docs/AppDevGuide/node7.html>
- [2] [https://wiki-ext.aps.anl.gov/epics/index.php/RRM\\_3-14\\_Calculation](https://wiki-ext.aps.anl.gov/epics/index.php/RRM_3-14_Calculation)