

実用EPICS入門-5

技術部専門講習

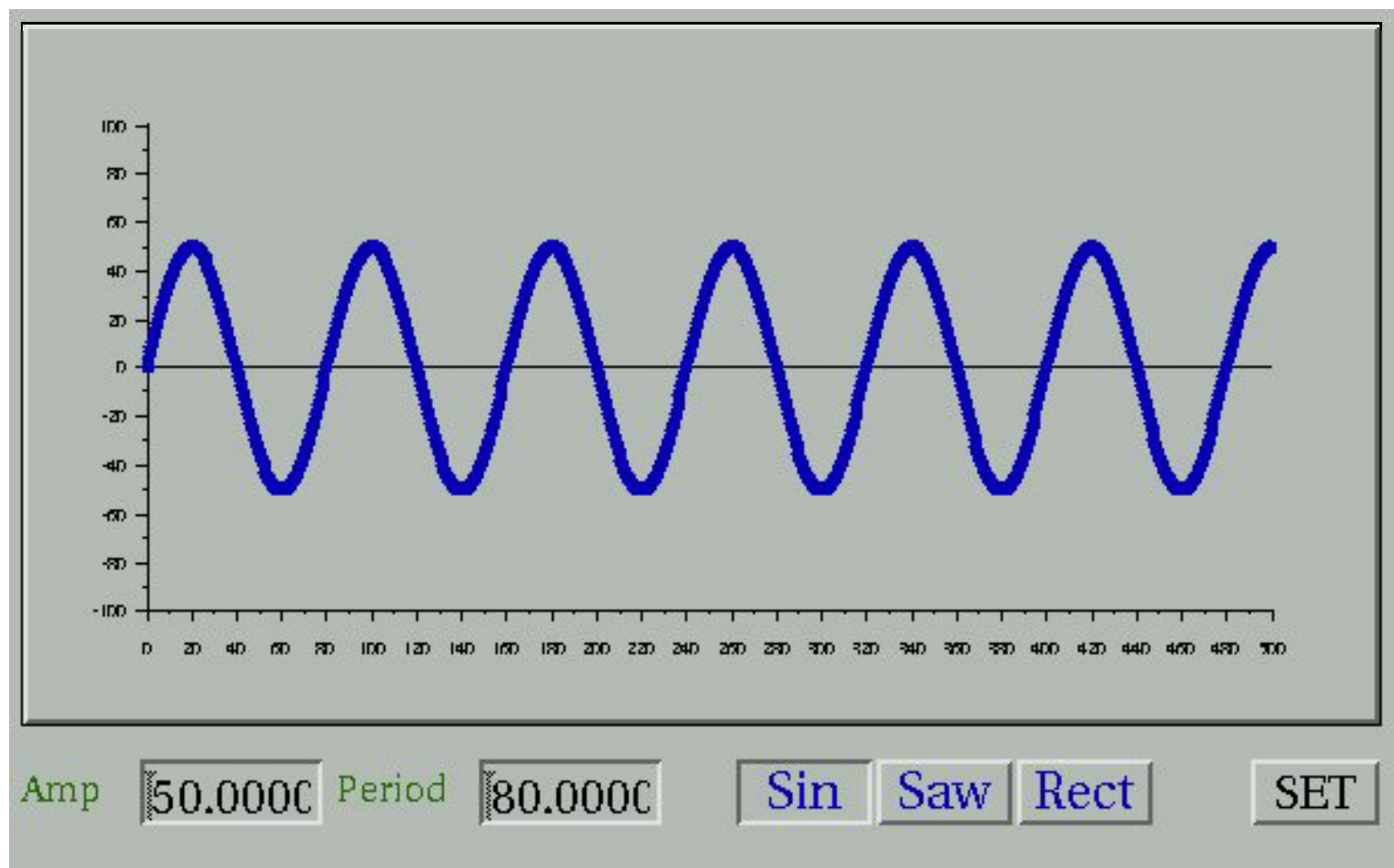
飛山真理

帯名崇

EPICSシーケンサを作ってみよう

- パターンgeneratorをつくる
 - sin波、三角波、矩形波のwaveformをつくる
 - 振幅、周期を指定できるようにする
 - SETボタンを押すとwaveformを生成し、dm2kで表示する
- はじめに、sin波だけのパターンgeneratorを作ってみる

完成図



まず、EPICSデータベースを作る

w_gene.db

- aiレコード
 - 振幅 **TDL**:WFG:AMP
 - 周期 **TDL**:WFG:PERIOD
- biレコード
 - セット **TDL**:WFG:SET
- waveformレコード
 - 波形 **TDL**:WFG:WAVE
- **TDLのところは、各自の名前に！**

aiレコード

```
record(ai,"$(USER):WFG:AMP"){  
  field(DESC,"analog input record")  
  field(SCAN,"Passive")  
  field(DTYP,"Soft Channel")  
  field(FLNK,"0.0")  
  field(INP,"0.0")  
  field(PREC,"5")  
}
```

aiレコード

```
record(ai,"$(USER):WFG:PERIOD"){  
  field(DESC,"analog input record")  
  field(SCAN,"Passive")  
  field(DTYP,"Soft Channel")  
  field(FLNK,"0.0")  
  field(INP,"0.0")  
  field(PREC,"5")  
}
```

biレコード

```
record(bi,"$(USER):WFG:SET"){  
    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")  
}
```

waveformレコード

```
record(waveform,"$(USER):WFG:WAVE"){  
  field(DESC,"waveform record")  
  field(SCAN,"Passive")  
  field(INP,"0.0")  
  field(FTVL,"DOUBLE")  
  field(HOPR,"100.0")  
  field(LOPR,"-100.0")  
  field(NELM,"500")  
}
```


データベースをseve、Makefileを 変更、make

- DB+= w_gene.db
- make(習慣として、Makefile、データベースを変更したら常にmakeすること)

sequencerを使えるよう設定

- `cd` でホームディレクトリへ
- `cd epicsApp/configure`
- `emacs RELEASE`

#If using the sequencer, point SNCSEQ at its top directory:
#SNCSEQ=\$(EPICS_BASE)/../modules/soft/seq

を

#If using the sequencer, point SNCSEQ at its top directory:
SNCSEQ=\$(EPICS_BASE)/../modules/soft/seq/2.0.11

epics directoryを再メイク

- `cd ~/epicsApp`
- `make distclean`
- `make` (ちょっと時間がかかります)

sourceディレクトリに移動

- `cd ~/epicsApp/trainApp/src`
- `emacs WFG_SRC.dbd`
 - 中身は
`registrar(WFGRegistrar)`
- `emacs WFG_SRC.stt`
 - sequencer本体

program WFG

%%#include <math.h>

%%#define PI 3.14159265

int SET;

assign SET to "TDL:WFG:SET";

monitor SET;

double WAVE[500];

assign WAVE to "TDL:WFG:WAVE";

double AMP;

double PERIOD;

assign AMP to "TDL:WFG:AMP";

assign PERIOD to "TDL:WFG:PERIOD";

```
int i;
```

```
ss function_generator
```

```
{
```

```
    state init
```

```
    {
```

```
        when()
```

```
        {
```

```
            for (i=0;i<500;i++) {
```

```
                WAVE[i]=0; }
```

```
        } state fg
```

```
    }
```

```
state fg
{
when(SET == 1){
    SET=0;
    pvPut(SET);
    pvGet(AMP);
    pvGet(PERIOD);
    if (PERIOD <1e-3) PERIOD=1e-3;
```

```
%{  
    for (i=0;i<500;i++)  
    {  
        WAVE[i]=AMP*sin(2*PI/PERIOD*i);  
    }  
}%  
pvPut(WAVE);  
} state fg  
}  
}
```


srcのMakefileを変更

```
ifneq ($(SNCSEQ),)
    #This builds sncExample as a component of train
    train_SNCFLAGS += +r
    train_DBD += sncExample.dbd
    train_SRCS += sncExample.stt
    train_DBD += WFG_SRC.dbd
    train_SRCS += WFG_SRC.stt
    train_LIBS += seq pv
```

makeする

- top(~epicsApp/)にもどってmake distclean、makeをした方が安心
- エラーが出たら、表示される行数を参考にエラーを修正する
- O.linux-x86にWFG_SRC.cファイルが出来ているはず

st.cmdを変更する

一番下に追加

```
## Start any sequence programs
```

```
#seq senExampe,"user=sluserHost"
```

```
seq WFG
```

st.cmdを実行してみる

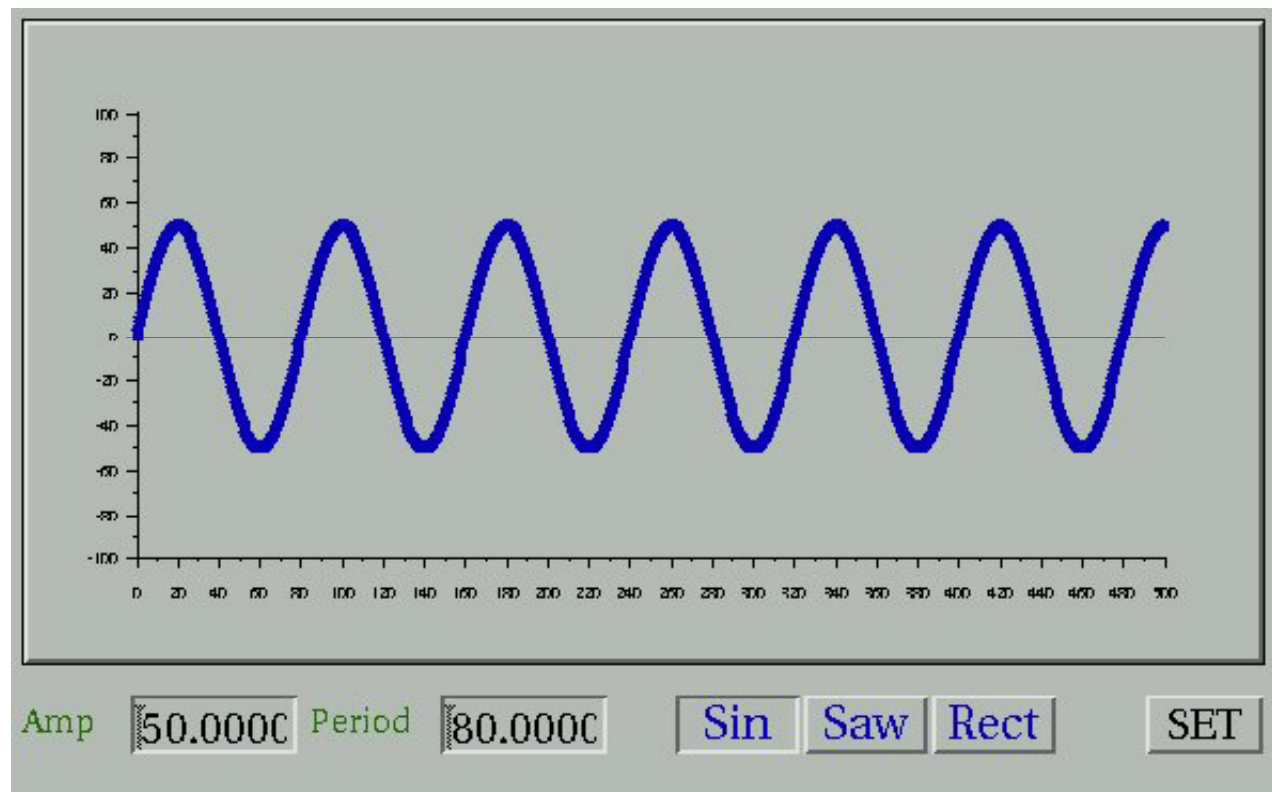
camonitor TDL:WFG:WAVE

で、TDL:WFG:AMPとTDL:WFG:PERIODをセ
ットし、TDL:WFG:SETを1にする

dm2kでセット、表示画面を作り、操作する

演習

- このsequenceを拡張し、当初の目標であるsin波、三角波、矩形波のwaveformをつくれるようにする



ヒント

- functionの選択は、mbboレコードを使ってみる

```
record(mbbo,"$(USER):WFG:FUNC"){  
    field(SCAN,"Passive")  
    field(OUT,"0.0")  
    field(ZRST,"Sin")  
    field(ONST,"Saw")  
    field(TWST,"Rect")}
```

ちょびっと高等技術

- sequencerでも名前置き換え機能を使用できます。

int SET;

assign SET to "{USER}:WFG:SET";

monitor SET;

で、st.cmdの中で

seq WFG, "USER=TDL"

次回

- EPICSデバイスサポート(1)
 - LAN－GPIBを使って、GP-IBデバイスをコントロールするのが目標
 - GP-IBの簡単な説明
 - デバイスサポートの構造、文法