

実用EPICS入門-7

技術部専門講習

飛山真理

帯名崇

Agilent 33220Aデバイスサポート

devHP33220A.dbd(自動生成)

```
device(ai,      GPIB_IO, devAiHP33220A,  "HP33220A")
device(ao,      GPIB_IO, devAoHP33220A,  "HP33220A")
device(bi,      GPIB_IO, devBiHP33220A,  "HP33220A")
device(bo,      GPIB_IO, devBoHP33220A,  "HP33220A")
device(event,   GPIB_IO, devEvHP33220A,  "HP33220A")
device(longin,  GPIB_IO, devLiHP33220A,  "HP33220A")
device(longout, GPIB_IO, devLoHP33220A,  "HP33220A")
device(mbbi,    GPIB_IO, devMbbiHP33220A, "HP33220A")
device(mbbiDirect,GPIB_IO, devMbbidHP33220A, "HP33220A")
device(mbbo,    GPIB_IO, devMbboHP33220A, "HP33220A")
device(mbboDirect,GPIB_IO, devMbbodHP33220A, "HP33220A")
device(stringin, GPIB_IO, devSiHP33220A,  "HP33220A")
device(stringout, GPIB_IO, devSoHP33220A,  "HP33220A")
device(waveform, GPIB_IO, devWfHP33220A,  "HP33220A")
```

```
include "asyn.dbd"
```

devHP33220A.c

```
static char *onOffList[]={"Off","On"};  
static struct devGpibNames onOff  
    ={2,onOffList,0,1};
```

```
static char *FuncList[] = {  
    "SIN", "SQU", "RAMP", "PULS", "NOIS", "DC", "USER"};  
static unsigned long FuncVal[]={0,1,2,3,4,5,6};  
static struct devGpibNames Func = {7, FuncList, FuncVal, 3};
```

```
static char *VUnitList[] = {  
    "VPP", "VRMS", "DBM"};  
static unsigned long VUnitVal[] = {0,1,2};  
static struct devGpibNames VUnit  
    = {3, VUnitList, VUnitVal, 2};
```

```
static char *FuncSet[] = {"FUNC SIN¥n","FUNC SQU¥n",  
    "FUNC RAMP¥n","FUNC PULS¥n","FUNC NOIS¥n","FUNC DC¥n",  
    "FUNC USER¥n"};  
static char *VUnitSet[] =  
    {"VOLT:UNIT:VPP¥n","VOLT:UNIT:VRMS¥n","VOLT:UNIT:DBM¥n"}  
    ;  
static char *onOffSet[] ={"OUTP OFF¥n","OUTP ON¥n"};  
static char *FuncRSet[]  
    ={"SIN¥n","SQU¥n","RAMP¥n","PULS¥n","NOIS¥n","DC¥n",  
    "USER¥n"};  
static char *VUnitRset[] = {"VPP¥n","VRMS¥n","DBM¥n"};  
static char *onOffRSet[] = {"0¥n","1¥n"};
```

```

static struct gpibCmd gpibCmds[] = {
    /* Param 0 -- initialize command */
    {&DSET_BO, GPIBCMD, IB_Q_LOW,
     "APPL:SIN 0.1 HZ, 0.3 VPP,0 V¥n", NULL, 0, 511, NULL, 0, 0,
     NULL, NULL, NULL},
    /* Param 1 - Function Set */
    {&DSET_MBBO, GPIBEFASTO, IB_Q_LOW, NULL, NULL, 0, 511,
     NULL, 0, 0, FuncSet, &Func, NULL},
    /* Param 2 -- Function Read */
    {&DSET_MBBI, GPIBEFASTI, IB_Q_LOW, "FUNC?¥n", NULL,0,
     511,NULL,0,0,FuncRSet,&Func, NULL},
    /* Param 3 -- Frequency Set */
    {&DSET_AO, GPIBWRITE, IB_Q_LOW, NULL, "FREQ %lf¥n", 0,
     511,NULL, 0,0,NULL, NULL, NULL},
    /* Param 4 -- Frequency Read */
    {&DSET_AI, GPIBREAD, IB_Q_LOW, "FREQ?¥n", "%lf¥n", 0, 511,
     NULL, 0, 0, NULL, NULL, NULL},

```

```
/* Param 5 -- Output Unit Set*/
{&DSET_MBBO,GPIBEFASTO,IB_Q_LOW,NULL,NULL,
0,511,NULL,0,0,VUnitSet, &VUnit, NULL},
/* Param 6 -- Output Unit Read */
{&DSET_MBBI,
GPIBEFASTI,IB_Q_LOW,"VOLT:UNIT?¥n",NULL,0,511,
NULL,0,0,VUnitRset,&VUnit,NULL},
/* Param 7 -- Volt Set */
{&DSET_AO, GPIBWRITE, IB_Q_LOW, NULL,
"VOLT %lf¥n",0,511,NULL,0,0,NULL,NULL,NULL},
/* Param 8 -- Volt Read */
{&DSET_AI,
GPIBREAD,IB_Q_LOW,"VOLT?¥n","%lf¥n",0,511,NULL,
0,0,NULL,NULL,NULL},
```

```

/* Param 9 -- Volt Offset Set*/
{&DSET_AO, GPIBWRITE, IB_Q_LOW, NULL,
 "VOLT:OFFS %lf¥n",0,511,NULL,0,0,NULL,NULL,NULL},
/* Param 10 -- Volt Offset Read */
{&DSET_AI,
 GPIBREAD,IB_Q_LOW,"VOLT:OFFS?¥n","%lf¥n",0,511,NULL,0,0,
 NULL,NULL,NULL},
/* Param 11 -- Out ON/OFF */
{&DSET_BO,GPIBEFASTO,IB_Q_LOW,NULL,NULL,0,80,NULL,0,
 511,onOffSet, &onOff,NULL},
/* Param 12 -- Out ON/OFF read */
{&DSET_BI, GPIBEFASTI,
 IB_Q_LOW,"OUTP?¥n",NULL,0,511,NULL,0,0,onOffRSet,&onOff,
 NULL}
};

```


EPICSデータベース

```
record(bo,"$(USER):FG:INIT"){  
    field(DTYP,"HP33220A")  
    field(OUT,"#L0 $(ADDR) @0")  
}  
record(mbbo,"$(USER):FG:FUNC:SET"){  
    field(DTYP,"HP33220A")  
    field(OUT,"#L0 $(ADDR) @1")  
}  
record(mbbsi,"$(USER):FG:FUNC:READ"){  
    field(DTYP,"HP33220A")  
    field(SCAN,"10 second")  
    field(INP,"#L0 $(ADDR) @2")  
}
```

```
record(ao,"$(USER):FG:FREQ:SET"){  
    field(DTYP,"HP33220A")  
    field(OUT,"#L0 $(ADDR) @3")  
    field(PREC,"5")  
}
```

```
record(ai,"$(USER):FG:FREQ:READ"){  
    field(DTYP,"HP33220A")  
    field(SCAN,"10 second")  
    field(INP,"#L0 $(ADDR) @4")  
    field(PREC,"5")  
}
```

```
record(mbbo,"$(USER):FG:OUT:UNIT:SET"){  
    field(DTYP,"HP33220A")  
    field(OUT,"#L0 $(ADDR) @5")  
}
```

```
record(mbbi,"$(USER):FG:OUT:UNIT:READ"){  
    field(DTYP,"HP33220A")  
    field(SCAN,"10 second")  
    field(INP,"#L0 $(ADDR) @6")  
}  
record(ao,"$(USER):FG:VOLT:SET"){  
    field(DTYP,"HP33220A")  
    field(OUT,"#L0 $(ADDR) @7")  
    field(PREC,"5")  
}  
record(ai,"$(USER):FG:VOLT:READ"){  
    field(DTYP,"HP33220A")  
    field(SCAN,"10 second")  
    field(INP,"#L0 $(ADDR) @8")  
    field(PREC,"5")  
}
```

```
record(ao,"$(USER):FG:VOLT:OFFSET:SET"){  
    field(DTYP,"HP33220A")  
    field(OUT,"#L0 $(ADDR) @9")  
    field(PREC,"5")  
}  
record(ai,"$(USER):FG:VOLT:OFFSET:READ"){  
    field(DTYP,"HP33220A")  
    field(SCAN,"10 second")  
    field(INP,"#L0 $(ADDR) @10")  
    field(PREC,"5")  
}
```

```
record(bo,"$(USER):FG:OUT:SET"){  
    field(DTYP,"HP33220A")  
    field(OUT,"#L0 $(ADDR) @11")  
}  
record(bi,"$(USER):FG:OUT:READ"){  
    field(DTYP,"HP33220A")  
    field(SCAN,"10 second")  
    field(INP,"#L0 $(ADDR) @12")  
}
```

ちょっと高等技術1

- GP-IBデバイスが配列、あるいはバイナリーデータを返す場合
 - オシロスコープのトレース
 - 多チャンネルDMMの出力
- デバイスサポートでこれをwaveform(あるいは該当するデータタイプ)に変換する必要があります。

例

- HP8590 spectrum analyzerの画面データ (trace A)
- TDF P;AUNITS DBM;TA;¥r¥nでtrace AのデータをdBm単位で送り返す
 - データ点数は399点
 - データは¥r¥nで切れたテキストファイル

¥r¥nで区切られた文字配列をデータ列に変換する関数

```
static int rd_data(char wav[], float data[])
{
    int i;
    char wa1[32767];
    strcpy(wa1, strtok(wav, "¥r¥n"));
    data[0] = (float) atof(wa1);
    for (i=1; i<399; i++)
    {
        strcpy(wa1, strtok(NULL, "¥r¥n"));
        data[i] = (float) atof(wa1);
    }
    return 0;
}
```


データ変換関数

```
STATIC int rd_wf_data(struct gpibDpvt *pdpvt, int p1, int p2, char **p3)
{
    struct waveformRecord *pwf = (struct waveformRecord *) (
        pdpvt->precord);
    char *craw;
    float *temparray;
    float temp[500], *ptemp;
    int ok_flag;
    unsigned long numElem;

    temparray = (float *) pwf->bptr; (最終データを返す先)
    craw = pdpvt->msg; (rawデータが入っている先)
```

```
if (craw == NULL)
{
    devGpibLib_setPvSevr(pwf, READ_ALARM, INVALID_ALARM);
    return(ERROR);
}
```

gpib機器からの返事が空だった時、recordをinvalidにして、errorメッセージを返す

```
ok_flag = rd_data(craw,temp);  
if (ok_flag != 0)  
{  
    devGpibLib_setPvSevr(pwf,READ_ALARM, INVALID_ALARM);  
    return(ERROR);  
}
```

- 前に定義した、文字列を数字配列に変換するfunctionを呼び、データ変換をする。正常終了出来なかったときは、waveformをinvalidにして、エラーだと言う

```
numElem = 399;
if (numElem > pwf->nelm)    numElem = pwf->nelm;
pwf->nord = numElem;
ptemp=temp;
while (numElem--)
{
    *temparray++ = (float) *ptemp++;
}
return(OK);
}
```

- データをwaveformレコードに入れていく

gpibCmd構造体

```
{&DSET_WF, GPIBREAD, IB_Q_LOW,  
  "TDF P;AUNITS DBM;TA;¥r¥n",  
  NULL, 0, 8191, rd_wf_data, 0,  
  0, NULL, NULL, NULL},
```

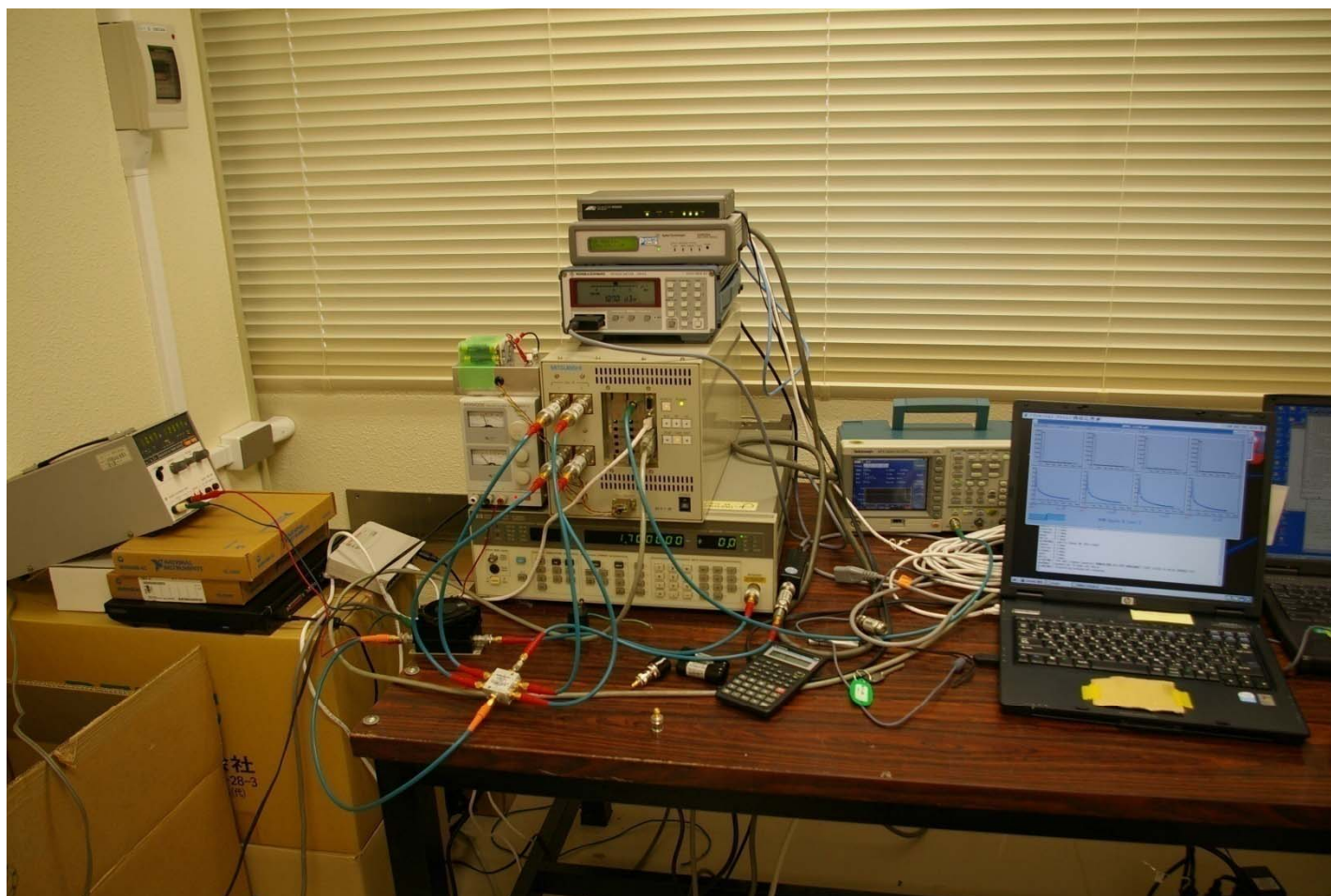
サービスリクエスト

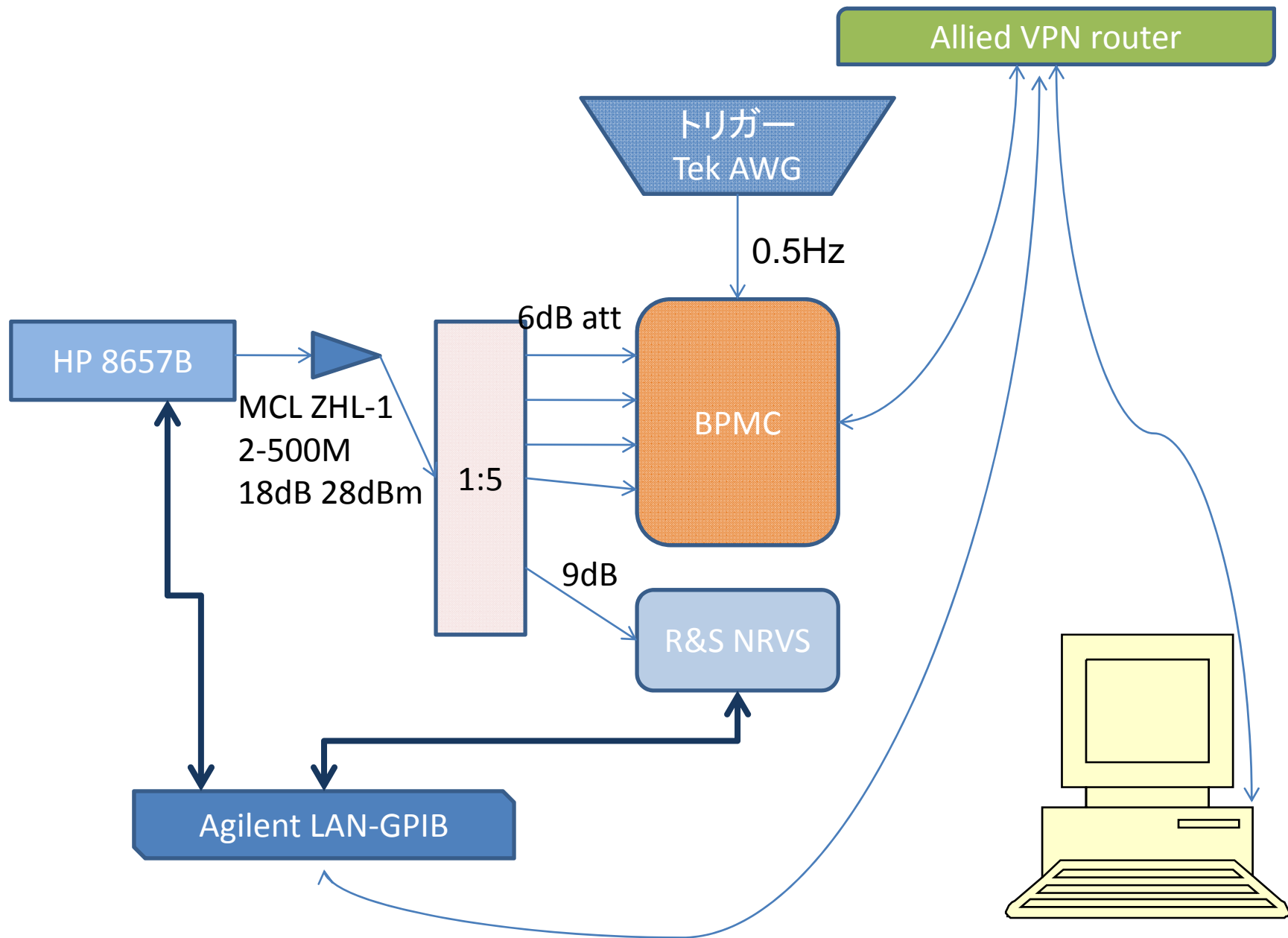
- GPIB機器側から、コントローラへactionを求める割り込み信号を送る
 - 機器側がSRQラインをlowにする
 - コントローラは、SRQがlowになったら、SRQを出した機器をシリアルポールあるいはパラレルポールで探し、プログラムで決められた動作をする
 - 自動で何かをする訳ではない。プログラムで動作を決めない限り、SRQが出ても何もしない。
 - パラレルポールする機器は見たことがない。

サービスリクエストの例

- ローデシュワルツNRVSパワーメータ
 - RFパワーを測定する器械
 - パワーが低いときは、データを読み取れるまで時間がかかる(～秒程度)
 - 読み取り準備が出来たら、サービスリクエストを出すことができる

セットアップ





GPIBコマンドを送り、SRQが出るまで待ってデータを取る

```
{&DSET_AI,GPIBREADW,IB_Q_HIGH,  
"X2¥r¥n","%lf",0,2047,0,0,0,NULL,NULL,0},
```

- X2コマンドを送り、SRQが出たら値を読みに行く。

何でもないときに突然SRQが出る

- エラーとか、故障とか。
- これに対応するデバイスサポートは面倒なので、専門家にお願いした方が吉。
- 実際上は、SRQを使わずにすませるコードを書く(例えば、シーケンサで十分待たせるとか)ことの方が多い。

ASYNドライバ

- Agilent 33220AにはEthernetの口がついていて、gp-ibとおなじメッセージベースコマンドで動作するので、EPICSのASYNドライバを直接使ってコントロールも出来ます
 - いまひとつ動作が不安定な欠点有り
 - いまだに発展途上の印象有り

やり方

```
dbLoadRecords("db/hpfg.db","USER=TDL, PORT=#L1, ADDR=A10)
```

```
drvAsynIIPortConfigure("L1","192.168.0.4:5025",0,0,0)
```

- vxi11ConfigureでL0を使っていないなら、L0でもOK
- ADDRの数値の意味は無い(でも書いておく必要有り)
- 5025:Agilent 33220Aが通信に使うport番号

今回までで

- 自分用epics環境がつくれる
- データベースがつくれる
- 簡易表示ディスプレイが作成できる
- 自分でデータをarchiveできる
- シーケンサが作れる
- 簡単なGP-IBデバイスサポートが作れる

身の回りのもので、EPICSを動作させてみよう

- 持っていない機器(例えばLAN-GPIB)などは、誰かに借りるとかして、自分の研究、仕事に関係あるものを制御するシステムを作ってみよう

次回は

- その他のdevice supportの紹介
 - VME(VxWorks)
 - ASYN
 - netDev