

1 SR620 の設定

1.1 ローカル設定

SR620 のフロントパネルで所定の測定パラメータを設定する。

1.2 リモート設定

RS-232C または GPIB により測定パラメータを任意に変更する。
(具体的な設定方法は、後述のソースプログラムに記載してある。)

2 PC と SR620 間の通信パラメータ

SR620 側の通信パラメータはフロントパネルで設定する。例えば、9600 8 N (2)。
PC 側はソフトウェアで SR620 側に対応するパラメータを設定する。特にストップビットを '2' とする理由は、SR620 側の送信が '2' に固定されているためである。

3 ソフトウェア

自作ソフトウェア 'sr620-232.exe' は、Windows(32 ビット版)のコンソール画面で動作する。その機能は、

- 1 分の周期でデータを表示する。データは SR620 の計測最終データ (平均計算の場合はその結果) である。
- ユーザが終了キー (Ctrl+C) を押すまで継続する。
- データは、年月日および時刻と共にデータ格納ファイル (sr620-232.txt) に文字形式で記録される。

sr620meas.txt の例:

```
09/29 23:58 1.01686233794E3  
09/29 23:59 1.01674460771E3  
09/30 00:00 1.01658324724E3
```

4 数値データへの変換

数値データから gnuplot が扱える数値データへの変換は、Excel または LibreOffice Calc によって行う。コラムの内容を一挙に数値データ化するのである。なお、年月日および時刻は文字データのままとする。

コラムごとのセパレータは空白文字とする。

5 gnuplot の設定

グラフの形式を定めるための sr620meas.fmt を次のとおり準備する。

```
#A parameter file to draw "sr620meas.dat" for GNUPLOT      (c)2018 JA5FP
set terminal wxt
#set terminal pdfcairo color size 6in,4in font "Arial, 10"
#set terminal pngcairo color size 6in,4in font "Arial, 10"
#set output 'sr620meas.pdf'
#set output 'sr620meas.png'
set title "Frequency drift of Audio Oscillator      (c)2018 JA5FP" font "Arial, 12"
set grid back
set xdata time
set format x "%H:%M"
set timefmt "%H:%M"
set xtics "00:00", "00:10", "00:59"
set xrange ["00:00":"00:59"]
set xlabel "Hour:minute (H:M) on 2018/09/30"
set ylabel "different frequency (Hz) from 100kHz"
unset key
```

gnuplot のコマンドラインに次を入力する。

```
load 'sr620meas.fmt'
```

6 gnuplot の描画

前項の sr620meas.fmt により gnuplot の出力先が wxt になっているので、グラフィック画面に描画される。そこで次のコマンドで sr620meas.dat のデータを一挙動で入力する。

```
plot 'sr620meas.dat' using 1:2 with lines
```

必要に応じて、これまでの操作を修正する。

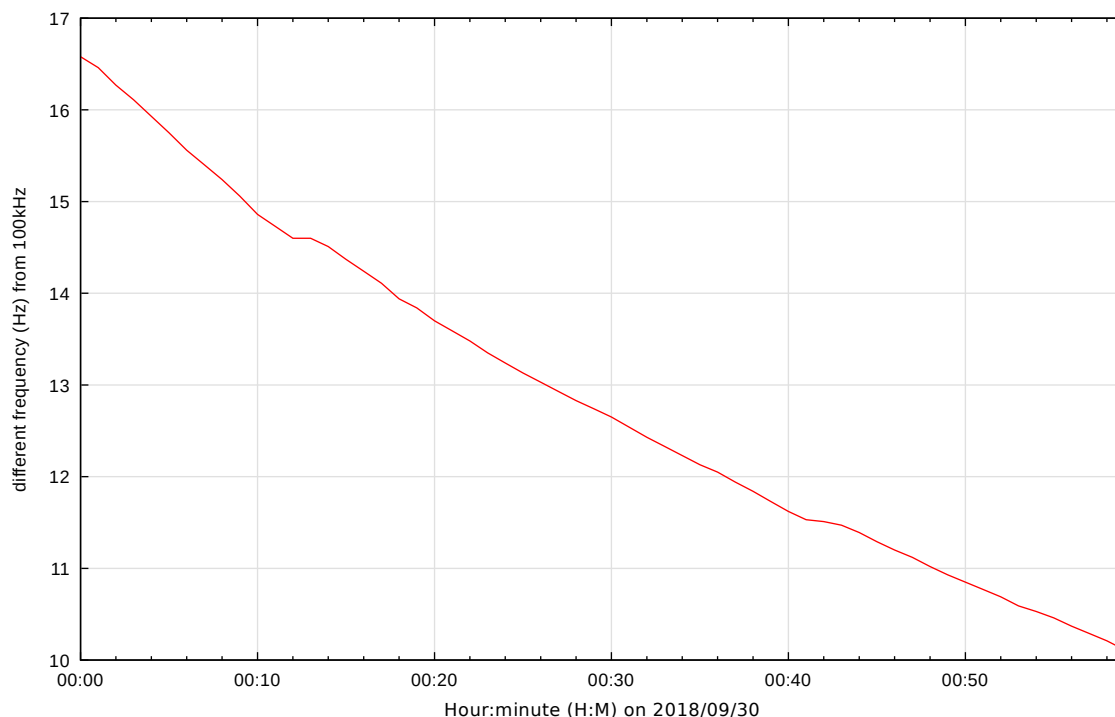
7 出力画像形式の選択と描画

出力先を pdf または png 形式とするばあいには、コマンドラインで次を入力する。

```
set terminal pdfcairo color size 6in,4in font "Arial, 10" set output 'sr620meas.pdf'
または
set terminal pngcairo color size 6in,4in font "Arial, 10" set output 'sr620meas.png'
```

これにより、次の画像が得られる。

Frequency drift of Audio Oscillator (c)2018 JA5FP



8 sr620.exe のソースファイル

```
// sr620-232.c
// A creamsoft program for "Control SR620 and get measured data" using API functions.
// (c) 2018.09.29 JA5FP
// Dev-C++ 4.9.9.2 on WindowsXP
// 仕様 SR620 のパネルまたは外部コマンドによる動作から得られる測定データを、連続して 1
// 分ごとに DOS コマンド画面に表示するとともに、sr620meas.txt ファイルに記録する。
// 外部コマンドを使用するには、setSR620 のコメントアウトを外す。
// sr620meas.txt が存在しない場合は自動的に作成され、存在する場合は上書きされる。
// フォーマットは次のような文字データである。
//           2018/09/29 12:08:00 1.0000000000E3
//           2018/09/29 12:09:00 1.0000000000E3
// 記録を終了するには、Ctrl+C コマンドを使用する。
// 表示および記録間隔を変えるには、ソースコードの%M に関する記述を変更し再コンパイルする。
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include <time.h>
#include <conio.h>
```

```

HANDLE hPort; // ポートのハンドル

int rs232c(void){
    hPort=CreateFile("COM1", GENERIC_READ | GENERIC_WRITE, // COM1 を開く
        0, NULL, OPEN_EXISTING, 0, NULL);

    DCB dcb;
    GetCommState(hPort, &dcb);
    dcb.BaudRate=9600; // パラメータ編集
    dcb.ByteSize=8;
    dcb.Parity=NOPARITY;
    dcb.StopBits=TWOSTOPBITS;
    dcb.fOutxCtsFlow=FALSE;
    dcb.fRtsControl=FALSE;
    SetCommState(hPort, &dcb); // パラメータ設定
}

int main(int argc, char *argv[]){
    rs232c(); // ポートの初期化

    char* setSR620; // SR620 外部コマンド
    // setSR620="MODE3\n;SRCE2\n;ARMM5\n;SIZE1\n;DISPO\n";
    // WriteFile(hPort, setSR620, strlen(setSR620), &dNrSend, NULL);
    // sleep(1000);
    DWORD dErr;
    COMSTAT cStat;
    int lenRcv;

    FILE *fStore; // データ保存ファイル
    fStore=fopen("sr620meas.txt", "a");

    DWORD dNrSend, dNrRcv; // 日時取得
    time_t timer;
    struct tm *timeptr;

    char past[3]; // 基準時刻の取得
    timer=time(NULL);
    timeptr=localtime(&timer);
    strftime(past, strlen(past), "%M", timeptr); // 基準を分とする

    char now[3], datetime[20]; // 現在分と年月日時刻
    while(1){ // 連続動作
        do { // 基準分超過を検出
            sleep(1000);

```

```

        timer=time(NULL);
        timeptr=localtime(&timer);
        strftime(now, sizeof(now), "%M", timeptr);
        printf("%s..", past); // 基準分のプロンプト
    }while(!strcmp(past, now));
timer=time(NULL); // 年月日時刻の取得
timeptr=localtime(&timer);
strftime(datetime, sizeof(datetime), "%Y/%m/%d %H:%M:%S", timeptr);
printf("%s \n", datetime); // 年月日時刻のプロンプト
ト

fputs(datetime, fStore); // 年月日時刻を保存
fputs(" ", fStore);

setSR620="MEAS?0\n"; // 測定データを要求
WriteFile(hPort, setSR620, strlen(setSR620), &dNrSend, NULL);
sleep(1000); // SR620 の反応待ち

ClearCommError(hPort, &dErr, &cStat); // データを受信
lenRcv=cStat.cbInQue;
char rcvData[lenRcv];
ReadFile(hPort, rcvData, lenRcv, &dNrRcv, NULL);
printf(" %s\n", rcvData); // データをプロンプト

fputs(rcvData, fStore); // データを保存
fputs("\n", fStore);

strcpy(past, now); // 連続動作の条件
}

fclose(fStore); // 保存ファイルを閉じる
CloseHandle(hPort); // ポートを閉じる
return 0;
}

```