コミッショニングツールワーキンググループ

2018/7/3 12:15-13:00
参加者：高橋，横澤，山本尚弘，佐々井，田越，大原，坂井，
記録：田越
Wiki: http://gwwiki.icrr.u-tokyo.ac.jp/JGWwiki/KAGRA/Subgroups/DAS/WG/CommTools

進捗状況

タスク

- PythonからFrameデータを呼び出す(C)：大原，田越，神田，
  - (田越)
    - KAGALIに新しいFrameファイル読み込み関数作成.
    - PythonからFrameデータ読み込み成功．付録参照.
    - 値を返す関数はXKGLを付ける

- Frameデータ　チャンネル一覧(KAGALI,Python)：FrChannelsを書き換え．譲原
  - (譲原)Frame LibraryにFrFileIgetChannelListという関数があった
  - 今週報告無し

- Daili summary page
  - 神岡からアクセス出来るようにした.
  - 16kHzの24時間データがプロット出来ない．rootの仕様か.
  - daily summary
    - http://www.icrr.u-tokyo.ac.jp/~yuzu/bKAGRA_summary/html/

- 時間指定をしてファイルを開くツール：田越
  - 関数化する．時刻を指定するとframeデータを読みだす関数.
    - lalframe.frreadを使ったもの制作中.

- 長期の時系列(Python)：佐々井

  - 長期とは何か?
    - (横澤)1分ごと，1時間ごと，1日ごとの平均などをみれるとうれしい.
    - 神岡のシステムにはある．主データ装置にもあるとうれしい.
    - (山本)trend/のFrameファイルにそのようなデータが入っている.
    - (山本)1時間の平均，最大，最小などあるとよい.

  - 主データ装置で動くことを確認.

- 時系列のヒストグラム(Python)：佐々井 => 検討中
    - 主データ装置，matplotlibのグラフが開けない．=> ssh -YC付けてOK.

- スペクトログラム(C,...)：佐々井 => 検討中
- detrend(トレンド補正)(C)：大原

    - 今週進展無し
    - 大原：Python, Matlabを調査した
    - これらを参考に早急に作成する
    - HHTのような新しい方法も作れるかも知れない．(研究テーマになる)
    - 大原：pythonと同じものをCで作った． git へはまだ.
    - gslを使っている．piesewise linearのトレンド補正も作っている．平均値を引くというのもある.

- window function(C)：大原

    - 大原：トレンド補正の後にやる.
    - まだ.

- high pass, low pass filter, band pass, band stop filter(C)，filtfilt(C)：坂井，大原

    - 坂井：大原さん，酒井君がつくったものを発展させていく.
    - low pass, high passを追加した.
    - (大原)KAGALIにのせるやり方をこれから教える
    - scipyと比較している.

- 今週以下は進展無し

    - コヒーレンス(２つの信号のクロススペクトラム)(C,Python)：議論無し
    - 1つのCHと多数のCHとのコヒーレンス(C,Python)：議論無し
    - LIGO でやってるiDQのようなこともいずれする必要があるか.
        - iDQ:多変量解析

            - 伊藤：韓国Gがやっているらしい(John Ohと話をする)

GUI ツール

- 今週進展無し
- 田越：PythonによるGUIツールを調べている．GWpyも調べる.
- 山本：三代君がGWpyで遊んでいる．GWpyを使ったツールがあれば，神岡で使える.
- 山本：GWpyは表示が速い．読み出し速度の問題解決に有効そうである.

その他

- 横澤：チャンネル名とその内容の説明のリストを作っている.

    - http://gwwiki.icrr.u-tokyo.ac.jp/JGWwiki/KAGRA/Commissioning/Phase1/Operation/ChannelList?action=AttachFile&do=view&target=bKAGRA_phase1_channel.pdf
    - 新しいサブシステム発足.

- 苔山: Duncan Macleod に話をしてもらう．10月15日の週に来日する.
- 横澤: PEMの予算を考えている． https://gwdoc.icrr.u-tokyo.ac.jp/DocDB/0084/G1808435/001/PEM_plan_180702.pdf

次回

2018年7月9日(月) 13:15 - 13:50 (重力波会議終了後)

付録

田越: swigをつかって，pythonからFrameデータを読み込む

swigとは

- http://www.swig.org
- C/C++コードを，様々な他の言語から呼び出して使うためのツール
- 対応言語： C#, Java, Perl, Python, TCL, R, Ruby, など

KAGALI Frameデータ読み込み関数 新しいものを作成

- 複数frameに対応
- ファイル名，チャンネル名(1つ)を与えれば，ファイルにあるデータを最初から最後まで読み込む.
- 保存されているデータ型のまま取り出す.

```
/* KGLframeread.h */

typedef struct _kglframedata
{
  double            srate;
  long              ndata;
  long              ndata_oneframe;
  int               nframe;
  char             Channel[128];
  unsigned short   Datatype;
  double           *dataD;
  float            *dataF;
  short            *dataS;
  int              *dataI;
  long             *dataL;
```

```
  complex float   *dataCF; /* 使わない*/
  complex double  *dataCD; /* 使わない*/
  unsigned short  *dataUS;
  unsigned int    *dataUI;
  unsigned long   *dataUL;
} kglframedata;

void KGLInitframedata
(//begin{proto}
 kglframedata *framedata      /**< [in/out] Frame Data structure*/
);//end{proto}

void KGLDestroyframedata
(//begin{proto}
 kglframedata *framedata      /**< [in/out] Frame Data structure*/
);//end{proto}

void KGLSetupframedata
(//begin{proto}
 char *cFileName,             /**< [in]  The name of Frame file */
 char *cChannel,              /**< [in]  The name of channel */
 kglframedata *framedata      /**< [in/out] Frame Data structure*/
);//end{proto}

kglframedata KGLSetupandReadFrame
(//begin{proto}
 char *cFileName,             /**< [in]  The name of Frame file */
 char *cChannel              /**< [in]  The name of channel */
// kglframedata *framedata      /**< [in/out] Frame Data structure*/
);//end{proto}

void KGLReadFrame
(//begin{proto}
 char *cFileName,              /**< [in]  The name of Frame file */
 char *cChannel,               /**< [in]  The name of channel */
 kglframedata *framedata      /**< [in/out] Frame Data structure*/
);//end{proto}
```

今のところ，Pythonから使うのは `kglframedata KGLSetupandReadFrame()`

swig用設定ファイル

```
/* KGLframeread.i */
%module KGLframeread
%{
/* Put header files here or function declarations like below */
#include "/Users/tagoshi/kagali/include/kagali/KGLStdlib.h"
```

```
#include "/Users/tagoshi/kagali/include/kagali/KGLUtil.h"
#include <FrameL.h>
#include "KGLframeread.h"
%}

%include "carrays.i"
%array_functions(double,doubleArray)
%array_functions(float,floatArray)
%array_functions(short,shortArray)
%array_functions(int,intArray)
%array_functions(long,longArray)
%array_functions(unsigned short,ushortArray)
%array_functions(unsigned int,uintArray)
%array_functions(unsigned long,ulongArray)

typedef struct _kglframedata
{
  double           srate;
  long             ndata;
  long             ndata_oneframe;
  int              nframe;
  char            Channel[128];
  unsigned short  Datatype;
  double          *dataD;
  float           *dataF;
  short           *dataS;
  int             *dataI;
  long            *dataL;
  complex float   *dataCF;
  complex double  *dataCD;
  unsigned short  *dataUS;
  unsigned int    *dataUI;
  unsigned long   *dataUL;
} kglframedata;

enum     {FR_VECT_C,     /* vector of char                          */
          FR_VECT_2S,    /* vector of short                         */
          FR_VECT_8R,    /* vector of double                        */
          FR_VECT_4R,    /* vector of float                         */
          FR_VECT_4S,    /* vector of int                           */
          FR_VECT_8S,    /* vector of long                          */
          FR_VECT_8C,    /* vector of complex float                 */
          FR_VECT_16C,   /* vector of complex double                */
          FR_VECT_STRING,/* vector of string                        */
          FR_VECT_2U,    /* vector of unsigned short                */
          FR_VECT_4U,    /* vector of unsigned int                  */
          FR_VECT_8U,    /* vector of unsigned long                 */
          FR_VECT_1U,    /* vector of unsigned char                 */
```

```
          FR_VECT_8H,    /* halfcomplex vectors (float) (FFTW order) */
          FR_VECT_16H,   /* halfcomplex vectors (double)(FFTW order) */
          FR_VECT_END};  /* vector of unsigned char                  */

extern void KGLInitframedata(kglframedata *framedata);
extern void KGLDestroyframedata(kglframedata *framedata);
extern kglframedata KGLSetupandReadFrame(char *cFileName, char *cChannel);
```

`KGLframeread.c, KGLframeread.h, KGLframeread.i` の３つのファイルを用意し，コンパイルする.

```
$ swig -python KGLframeread.i  /* KGLframeread_wrap.c作成*/
$ gcc -c `python-config --cflags` $(CFLAGS_KAGALI) KGLframeread.c KGLframeread_wrap.c
$ gcc -bundle `python-config --ldflags` $(LDFLAGS_KAGALI) KGLframeread.o KGLframeread_wrap.o -o _KGLframeread.so
```

```
# Makefile

CC = gcc
PKGCONFIG = pkg-config

LDFLAGS_KAGALI = $(shell $(PKGCONFIG) --libs kagali)
CFLAGS_KAGALI =  $(shell $(PKGCONFIG) --cflags kagali)

CFLAGS_ORI = -O2 -Wall --std=gnu99
CFLAGS_PY = `python-config --cflags`
LDFLAGS  = -bundle -L/opt/local/Library/Frameworks/Python.framework/Versions/3.6/l
ib/python3.6/config-3.6m-darwin -lpython3.6m -ldl -framework CoreFoundation

LDFLAGS := $(LDFLAGS) $(LDFLAGS_KAGALI)
CFLAGS := $(CFLAGS_ORI) $(CFLAGS_KAGALI) $(CFLAGS_PY)

all: _KGLframeread.so TestKGLframeread

_KGLframeread.so: KGLframeread.o KGLframeread_wrap.o
    gcc $(LDFLAGS) KGLframeread.o KGLframeread_wrap.o -o _KGLframeread.so

KGLframeread.o: KGLframeread.c
    gcc -c $(CFLAGS) KGLframeread.c

KGLframeread_wrap.o: KGLframeread_wrap.c
    gcc -c $(CFLAGS) KGLframeread_wrap.c

KGLframeread_wrap.c: KGLframeread.i KGLframeread.c KGLframeread.h
    swig -python KGLframeread.i

TestKGLframeread: TestKGLframeread.c KGLframeread.o
    $(CC) $(CFLAGS_ORI) $(CFLAGS_KAGALI) -o $@ $< KGLframeread.o $(LDFLAGS_KAGALI)


clean:
    rm -fr *_wrap.c *.dSYM *.o *.so KGLframeread.py
```

更にPythonラッパー関数

以上で一応Frameデータは読める．(速度的に改善の余地はある) 便利に読むためにラッパー関数作成 (framereadfn.py)

```
# framereadfn.py

import numpy as np
from KGLframeread import *

def readframe(filename,CH):
  framedata = KGLSetupandReadFrame(filename,CH)
  ndata = framedata.ndata
  srate = framedata.srate
  data = np.empty(ndata)
  type = framedata.Datatype

  if type == FR_VECT_8R:
    ctype = "FR_VECT_8R"
    for i in range(ndata):
      data[i] = doubleArray_getitem(framedata.dataD,i)
  elif type == FR_VECT_4R:
    ctype = "FR_VECT_4R"
    for i in range(ndata):
      data[i] = floatArray_getitem(framedata.dataF,i)
  elif type == FR_VECT_2S:
    ctype = "FR_VECT_2S"
    for i in range(ndata):
      data[i] = shortArray_getitem(framedata.dataS,i)
  elif type == FR_VECT_4S:
    ctype = "FR_VECT_4S"
    for i in range(ndata):
      data[i] = intArray_getitem(framedata.dataI,i)
  elif type == FR_VECT_8S:
    ctype = "FR_VECT_8S"
    for i in range(ndata):
      data[i] = longArray_getitem(framedata.dataL,i)
  elif type == FR_VECT_2U:
    ctype = "FR_VECT_2U"
    for i in range(ndata):
      data[i] = ushortArray_getitem(framedata.dataUS,i)
  elif type == FR_VECT_4U:
    ctype = "FR_VECT_4U"
    for i in range(ndata):
      data[i] = uintArray_getitem(framedata.dataUI,i)
  elif type == FR_VECT_8U:
    ctype = "FR_VECT_8U"
    for i in range(ndata):
      data[i] = ulongArray_getitem(framedata.dataUL,i)
  else:
    print('Datatype %ud is not supported' % type)

  return data, ndata, srate, ctype
```

テスト

```
# testframeread.py

import numpy as np
import matplotlib.pyplot as plt
from framereadfn import *

filename = "H-H1_LOSC_4_V2-1126259446-32.gwf"
#CH = "H1:LOSC-STRAIN"
CH = "H1:LOSC-DQMASK"
#CH="H1:LOSC-INJMASK"

filename = "/data/full/12095/K-K1_C-1209500000-32.gwf"
CH = "K1:CAL-CS_PROC_C00_STRAIN_DQ"
#CH = "K1:GRD-LSC_MICH_STATE_N"

data,ndata,srate,ctype = readframe(filename,CH)
dt=1./srate
timelist=np.arange(ndata)*dt

plt.plot(timelist,data)
plt.title(CH + "( %.0f Hz, %s)" % (srate,ctype))
plt.show()
```

```
$ ls
H-H1_LOSC_4_V2-1126259446-32.gwf*   Makefile
KGLframeread*               Makefile.1
KGLframeread.c              TestKGLframeread*
KGLframeread.h              TestKGLframeread.c
KGLframeread.i              _KGLframeread.so*
KGLframeread.o              __pycache__/
KGLframeread.py             framereadfn.py
KGLframeread_wrap.c         setup.py
KGLframeread_wrap.o         testframeread.py

$ python testframeread.py
```