Description

This code, provided the shot number and the string corresponding to a given signal, accesses the database, defines the corresponding IDL object if connection has already been established there is no need to create a new object), reads the signal, builds the time vector and returns a 2-column array with timevector as first column and signal as second column. If "verifica"=1 then it firstly checks if the signal requested exists in the database, if it doesn't then the code stops.

The user should modify accordingly the server and port being used in his/her HomeLab. The present code is intended for ISTTOK data only, for which the server is 'baco.cfn.ist.utl.pt' and Port:8888.
Author : Rui Coelho
First Version : 24 April 2006

```
;###################### ROUTINE DEFINITION
####################################

;----------- Establish connection to the server and define corresponding
object -------
FUNCTION CONNECT
oJSDASConnection=OBJ_NEW('IDLjavaObject$ORG_SDAS_CORE_CLIENT_SDASCLIENT',$
'org.sdas.core.client.SDASClient', 'baco.cfn.ist.utl.pt', 8888)
RETURN,oJSDASConnection
END
;-------------------------------------------------------------------------------

;------------ Check if requested signal exists for the selected shot
------------------
FUNCTION CHECKSIG, bridge,sig_label,shotnum
dataFound = bridge -> parameterExists(sig_label,'0x0000', shotnum)
RETURN,datafound    ;boolean (0,1)
END
;-------------------------------------------------------------------------------

;--------------- Get the data given the signal name and shot number
-------------------
FUNCTION GETSDATA, bridge,sig_label,shotnum

dataStruct=bridge->getData(sig_label,'0x0000', shotnum)
dataStruct=dataStruct[0]
y = dataStruct -> getData()

;2. The number of points
npoints = float(n_elements(y))

;3. The start time
tstart= dataStruct -> getTStart()
;4. The end time
tend= dataStruct -> getTEnd()
;4. The time between samples is:
```

```
tbs= (tend->getTimeInMicros() - tstart->getTimeInMicros())/npoints
;5. The events
events = dataStruct -> getEvents()
;6. The event time (I'm assuming the event I want is at the index 0, but
I should check first...)
tevent = events[0] -> getTimeStamp()
;7. The delay of the start time relative to the event time
tini = tstart->getTimeInMicros() - tevent->getTimeInMicros()

;8. Create the time array
x = FLTARR(npoints)
FOR i = 0L, npoints-1 DO x[i] = tini + i*tbs

data=transpose([[x],[y]])   ; put in 2 column matrix: 1st has time, 2nd
has signal
RETURN,data
END
;-------------------------------------------------------------------------------
;######################## END OF ROUTINE DEFINITION
####################################


;####################  MAIN PROGRAM
################################################
;
#
; What code does : connects to SDAS and reads one single signal/shot,
provided    #
;                   by the
user                                                        #
; Usage : * Select what signal from given shot you want to
read                     #
;         * To bypass checking the existence of the given (signal,shot)
pair in   #
;           the server change verify to
0.                                          #
;         * To get different signal/shot just change fields and Ctrl-F5
(compile) #
;           and F5
(Run)                                                          #
;
#
;#############################################################################


signal = 'MAGNETIC_MAGPOL.MAGP4'
shot  = 11672

if n_elements(onSDAS) EQ 0 then onSDAS=CONNECT()
verify = 1
```

```
if verify eq 1 then begin
  signal_exists= CHECKSIG( onSDAS,signal,shot)
  if signal_exists eq 0 then begin
    print,"Signal does not exists for that shot number !"
    stop
  endif $
  else begin
    print,"Signal exists for that shot number !"
  endelse
endif


sinal=GETSDATA(onSDAS,signal,shot)
plot,sinal(0,*),sinal(1,*)

END
```