1 PROBA2/LYRA SSW routines

1.1 Environment

The LYRA SSW routines are routines developed in IDL that use some of the main standard SolafSoft functions. They therefore request that both IDL and the SolarSoft are installed on the computer. Information on how to install the SolarSoft can be found on `http://www.lmsal.com/solarsoft/`. If you are installing SolarSoft for the first time, make sure you select the LYRA packages! If you already have IDL and SolarSoft installed, but don't have the LYRA package, or if you want to update it :

- run the solar soft
- `IDL> ssw_upgrade, /lyra, /loud, /spawn`
- add 'lyra' to the SSW_INSTR environment in the SolarSoft configuration file

1.2 Getting LYRA data

The main function to get and read one day of LYRA data is *lyra_get_data*. This function successively :

1. download the corresponding LYRA fits file using the *download_data* routine
2. read it
3. download (if needed) the event files with *lyra_download_events* that is used to clean up the LYRA time series and clean the time-series with *lyra_remove_event*
4. if requested, rebid the LYRA data with *lyra_rebin_series*

The data are then available in a structure, ready to be exploited or plotted. For example, try :

```
IDL> data = lyra_get_data('/home/lyra/', 2011,  8, 1, 'lev3')
IDL> plot, data.julian, data.aluminum, xtickformat = $
'(C(CHI2.2, ":",CMI2.2, ":",  CSI2.2))'
```

Each of the function used in *lyra_get_data* can be also used individually.

1.3 Description of LYRA functions

1.3.1 The *lyra_get_data routine*

```
NAME:
   lyra_get_data
 PURPOSE:
   Download (when needed) and read a daily LYRA fits file
 CALLING SEQUENCE:
   result = lyra_get_data(path_data, year, month, day, level,$
                          /skip_lar, /skip_flare, /skip_occultation,$
                          rebinning_time = rebinning_time)
 INPUTS:
   - path_data: parent directory in which to store the downloaded
     data files
   - year, month, and day (in numeric format) corresponds to date of
     the data to read.
   - level is either "lev1", "lev2", "lev3", "bst", "bca", "cal", or
     "meta" and corresponds to the lyra data level to read
     (lev1 = uncalibrated, lev2 = calibrated, lev3 = calibrated and
     one-minute averaged, bst = other unit than the nominal,
     cal = calibration data, bca= other unit calibration, meta = metadata).
   - skip_lar is an optional keyword. When set, it makes the routine
     to clean up the time series from the perturbations caused by
     spacecraft maneuvers.
   - skip_flare is an optional keyword. When set, it makes the
     routine to clean up the time series from all flares
   - skip_occultation is an optional keyword. When set, it makes the
     routine to clean up the time series from the perturbations
     caused by occultations.
   - rebinning_time is an optional argument in seconds. When a value
     is attributed to it, it asks the routine to rebin the data to
     the specified time sampling. The nominal time sampling in LYRA
     data is 0.05 (except for level 3 data where it is 60).
     If no rebinning is desired, set this parameter to -1
     (default). When not attributed, the nominal sampling is kept.
 OUTPUTS:
   - When level 1, 2, 3, bst, cal or bcal data are requested, a
     structure is returned containing the lyra timeseries, with the 'time'
     (in seconds of the day), the irradiances in the four lyra
     channels 'lyman', 'herzberg', 'aluminum', 'zirconium' (in W m-2),
     the fits_file 'header', and the time in julian date stored
     'julian'.
   - When meta data are requested the file is stored and the fully
     qualified filename is returned
   - When an error occures, -1 is returned.
 EXAMPLE:
   lyra_data = lyra_get_data('home', 2013, 5, 2, 'lev2', $
                             /skip_lar, /skip_flare,$
```

```
                                rebinning_time = 60)
MODIFICATION HISTORY:
  ver 1.0:   First release by M. Dominique (mariedo@oma.be) and
             A. Katsiyannis (katsiyannis@oma.be), 01/11/2014
  ver 1.1:   Creation of the lyra_get_data_error_check function and
             bug fixes. M. Dominique and A. Katsiyannis , 09/07/2015
```

1.3.2 The *lyra_download_data* routine

```
NAME:
  lyra_download_data
PURPOSE:
  Download (when needed) and read data files
CALLING SEQUENCE:
  result = lyra_download_data('address', 'main_dir',
                              new_filename = 'name', /force)
INPUTS
  - address = http or ftp address where the file can be found
  - main_dir = path to the main directory in which to store
    the data locally
  - new_filename = When the file is downloaded, it is given this
    name. Optional parameter
  - the 'force' keyword is used to force re-downloading a file that
    has already been downloaded in the past
OUTPUTS
  - full path of the downloaded file
EXAMPLE:
  path = lyra_download_data('http://proba2.oma.be/lyra/data/eng/2014/' +$
          '04/07/lyra_20140407-000000_lev1_std.fits','.')
MODIFICATION HISTORY:
  ver 1.0:   First release by M. Dominique (mariedo@oma.be) and
             A. Katsiyannis (katsiyannis@oma.be), 01/11/2014
  ver 1.1:   Bug fixes, M. Dominique and A. Katsiyannis, 09/07/2015
```

1.3.3 The *lyra_download_events* routine

```
NAME:
  lyra_download_events
PURPOSE:
  Download a file containing all the events affecting the LYRA data
  (eclipses, maneuvers, specific acquisition campaigns ...) during a
  given time period. This event file can then be used to clean up
  the LYRA time series.
CALLING SEQUENCE:
  result = lyra_download_events(main_path, start_date, end_date)
INPUTS:
  - main_path = path to the main directory in which to store
    the data locally. The file will be named 'yyyymmdd_events.txt',
    where yyyymmdd corresponds to the date provided in the
    'start_date' argument.
  - start_date = beginning date of the time period for which to collect
    the LYRA events. It must be a string argument of 'yyyy-mm-dd'. The
    routine will convert non-existing dates in exsiting ones. For example,
    '2012-02-31' will be interpreted as '2012-03-02' and '2010-13-01' as
    '2011-01-01'. No date before '2010-01-01' will be accepted.
  - end_date = end date (included) of the time period for which to collect
    the LYRA events. It must be a string argument of 'yyyy-mm-dd'
OUTPUTS
  - array containing the full paths of the downloaded files
EXAMPLE:
  result = lyra_download_events('/home', '2012-01-01', '2012-01-05')
MODIFICATION HISTORY:
  ver 1.0:   First release by M. Dominique (mariedo@oma.be) and
             A. Katsiyannis (katsiyannis@oma.be), 01/11/2014
  ver 1.1:   Bug fixes, converted from routine to function
             M. Dominique, A. Katsiyannis, 09/07/2015
```

1.3.4 The *lyra_remove_event* routine

```
NAME:
  lyra_remove_events
PURPOSE:
  Remove from the time series "series_ts" all the samples
  corresponding to times during which an event of type "event_type"
  is recorded in the yyyy-mm-dd_events.txt files. yyyy, mm, dd are
  extracted from "date". If no "event_type" is provided, all the
  events are removed.
CALLING SEQUENCE:
  result = lyra_remove_event(path, date, time_ts, series_ts,$
           event_type = 'LAR')
INPUTS:
  - path = path to the main directory in which to locally store
    the events file. This file will be stored under the name
    'yyyymmdd_events.txt', where yyyymmdd corresponds to the day
    provided in 'date'.
  - date = date for which to clean the data. It must be a string
    argument of 'yyyy-mm-dd'
  - time_ts: the time vector expressed in seconds.
  - series_ts: vector containing the time series to clean up.
  - event_type = string telling the type of event to remove from the
    time series. The accepted values are 'LAR', 'Cover 1 open',
    'Cover 2 open', 'Cover 3 open', 'UV occ.', 'Vis. occ.',
    'SAA', 'A Flare', 'B Flare', 'C Flare', 'M Flare', 'X Flare'.
    The event_type 'all' is also accepted. In this case, all the events
    EXCEPT THE ONES CORRESPONDING TO COVERS are filtered out from
    the timeseries.
    This parameter is optional. If not provided, event_type 'all' is
    assumed.
  - keep_event allows the user to extract the periods corresponding
    to the selected events rather than to reject them from the
    timeseries.
OUTPUTS:
  - A vector corresponding to the cleaned up time series if all goes
    well. If no data survive the removal of events then 0 is
    returned. If an error occured, -1 is returned.
EXAMPLES:
  clean_series = lyra_remove_event('/home', '2012-01-01',$
  lyra.time, lyra.aluminum, event_type = 'UV occ.')
  unit3_data = lyra_remove_event('/home', '2012-01-01', lyra_bakup.time, $
  lyra_backup.aluminum, event_type = 'Cover 3 open', /keep_event)
MODIFICATION HISTORY:
  ver 1.0:   First release by M. Dominique (mariedo@oma.be) and
             A. Katsiyannis (katsiyannis@oma.be), 01/11/2014
  ver 1.1:   New keyword, modification of existing keywords, bug
             fixes,  M. Dominique and A. Katsiyannis, 09/07/2015
```

1.3.5 The *lyra_rebin_series* routine

```
NAME:
  lyra_rebin_series
PURPOSE:
  Provides the input time series rebinned on a lower time scale.
CALLING SEQUENCE:
  result = lyra_rebin_series(time, series, rebinning_time)
INPUTS:
  - time: the time vector expressed in seconds. It can be
    second-of-the-day or seconds ellapsed since any reference time.
    The time vector can be regularly sampled or not.
  - series: table containing the original time series to rebin.
    If several series are provided, they must correspond to  the
    same "time" vector. "series" is then a table with as many lines
    as they are series to rebin.
  - rebinning_time: numerical value corresponding to the desired
    time sampling (in seconds). This parameter should be bigger than
    the nominal time sampling. If this is not the case, the initial
    series will be returned without any rebinning.
    The nominal integration time in LYRA data is 0.05 (except in
    level 3 data where it is 1 min).
OUTPUTS:
  - structure containing the rebinned 'time' and 'series'.
EXAMPLE:
  rebinned_series = lyra_rebin_series(time, [[zirconium_channel],$
                    [aluminum_channel]], 60)
MODIFICATION HISTORY:
  Written by:    M. Dominique and A. Katsiyannis, 01/11/2014
```