**FOXSI Analysis Guide**

Last update:

2015 Jan 6: Created document LG

Please send questions and bug reports to glesener@ssl.berkeley.edu

1. **Getting FOXSI software**
   1. FOXSI software is written in IDL and requires SSWIDL to be installed.
   2. FOXSI analysis software is kept in a GitHub repository at <https://github.com/foxsi/foxsi-science>. You can download a snapshot of the software as a ZIP file from there. Alternatively, you can sign up for a GitHub account and use **git** to keep your software tree up to date. This second choice takes some studying but is useful if you want to contribute your additions and changes.
   3. Git expects the main directory of the software tree to be foxsi-science. In addition to the directories in the software tree, you need to add three additional directories, called data\_2012, data\_2014, and calibration\_data. These directories are **not** included in the GitHub repository because they contain large data files. Their contents are ignored by GitHub via text in the .gitignore file. You’ll need to populate these directories with data; see the next section.
2. **Getting FOXSI data**
   1. Download the following files from <ftp://apollo.ssl.berkeley.edu/pub/foxsi>. Choose either the data\_2012 or data\_2014 directory (for the first or second flight, respectively). Put these files in a data\_2012 or data\_2014 folder in your foxsi-science directory.
      1. foxsi\_level1\_data.sav
      2. foxsi\_level2\_data.sav
   2. On the same FTP site, download everything in the calibration\_data folder. These contain the necessary components to build the instrument response using whichever pieces (optics, detector, blanketing) are desired. Put these files in a calibration\_data folder in your foxsi-science directory.
3. **Setting up**

By now, you should have a foxsi-science directory with several subdirectories obtained from GitHub, as well as three data subdirectories that you’ve populated yourself.

To start using the FOXSI software, start SSWIDL. Run one of the following setup scripts at the prompt. Run only one script, not both, since the script sets flight-specific parameters like dates, times, and pointing positions.

IDL> @foxsi-setup-script-2012

or

IDL> @foxsi-setup-script-2014

Now you’re ready to go! You can skip the next step (“Processing FOXSI data into higher levels”) and go right to the fun parts, like images, spectra, and lightcurves.

1. **Processing FOXSI data into higher levels**

***For most purposes, you don’t need to do this. Just download the FOXSI Level 1 and Level 2 pre-processed files.***

Should you need to reprocess the FOXSI data for some reason, use the routines in the foxsi-science/proc/ directory. Run these in order, starting with the data file recorded by the WSMR ground station.

Procedure order:

* wsmr\_data\_to\_level0
* foxsi\_level0\_to\_level1
* foxsi\_level1\_to\_level2

A FOXSI data file recorded by the GSE (i.e. not by WSMR) can be processed in the same pipeline by substituting a different routine for the first one. Use the appropriate routine for either FORMATTER (500 frames/sec from formatter in a binary file) or USB (text file containing info from all strips) style data.

* formatter\_data\_to\_level0 OR usb\_data\_to\_level0
* foxsi\_level0\_to\_level1
* foxsi\_level1\_to\_level2

The above choices allow you to process any FOXSI data file into Level 2 data, whether from flight or calibration, no matter whether it was recorded through the formatter interface, USB interface, or ground station. However, note that there is a different set of software that is used for the FOXSI detector calibration, as the flight analysis software is not optimized to work with a large number of events (and was also written much later).

Example of normal processing starting from WSMR Ground Station file:

; Create Level 0 data

filename = 'data\_2012/36.255\_TM1\_Flight\_2012-11-02.log'

data\_lvl0\_D0 = wsmr\_data\_to\_level0( filename, det=0, year=2012 )

data\_lvl0\_D1 = wsmr\_data\_to\_level0( filename, det=1, year=2012 )

data\_lvl0\_D2 = wsmr\_data\_to\_level0( filename, det=2, year=2012 )

data\_lvl0\_D3 = wsmr\_data\_to\_level0( filename, det=3, year=2012 )

data\_lvl0\_D4 = wsmr\_data\_to\_level0( filename, det=4, year=2012 )

data\_lvl0\_D5 = wsmr\_data\_to\_level0( filename, det=5, year=2012 )

data\_lvl0\_D6 = wsmr\_data\_to\_level0( filename, det=6, year=2012 )

save, data\_lvl0\_D0, data\_lvl0\_D1, data\_lvl0\_D2, data\_lvl0\_D3, $

data\_lvl0\_D4, data\_lvl0\_D5, data\_lvl0\_d6, $

file = 'data\_2012/foxsi\_level0\_data.sav'

; Create Level 1 data

filename = 'data\_2012/foxsi\_level0\_data.sav'

data\_lvl1\_D0 = foxsi\_level0\_to\_level1( filename, det=0, ground=0 )

data\_lvl1\_D1 = foxsi\_level0\_to\_level1( filename, det=1, ground=0 )

data\_lvl1\_D2 = foxsi\_level0\_to\_level1( filename, det=2, ground=0 )

data\_lvl1\_D3 = foxsi\_level0\_to\_level1( filename, det=3, ground=0 )

data\_lvl1\_D4 = foxsi\_level0\_to\_level1( filename, det=4, ground=0 )

data\_lvl1\_D5 = foxsi\_level0\_to\_level1( filename, det=5, ground=0 )

data\_lvl1\_D6 = foxsi\_level0\_to\_level1( filename, det=6, ground=0 )

save, data\_lvl1\_D0, data\_lvl1\_D1, data\_lvl1\_D2, data\_lvl1\_D3, data\_lvl1\_D4, $

data\_lvl1\_D5, data\_lvl1\_d6, $

file = 'data\_2012/foxsi\_level1\_data.sav'

; Create Level 2 data

; Here, you need to know the specific detectors flown.

file0 = 'data\_2012/foxsi\_level0\_data.sav'

file1 = 'data\_2012/foxsi\_level1\_data.sav'

cal0 = 'calibration\_data/peaks\_det108.sav'

cal1 = 'calibration\_data/peaks\_det109.sav'

cal2 = 'calibration\_data/peaks\_det102.sav'

cal3 = 'calibration\_data/peaks\_det103.sav'

cal4 = 'calibration\_data/peaks\_det104.sav'

cal5 = 'calibration\_data/peaks\_det105.sav'

cal6 = 'calibration\_data/peaks\_det106.sav'

data\_lvl2\_D0 = foxsi\_level1\_to\_level2(file0, file1, det=0, calib=cal0 )

data\_lvl2\_D1 = foxsi\_level1\_to\_level2(file0, file1, det=1, calib=cal1 )

data\_lvl2\_D2 = foxsi\_level1\_to\_level2(file0, file1, det=2, calib=cal2 )

data\_lvl2\_D3 = foxsi\_level1\_to\_level2(file0, file1, det=3, calib=cal3 )

data\_lvl2\_D4 = foxsi\_level1\_to\_level2(file0, file1, det=4, calib=cal4 )

data\_lvl2\_D5 = foxsi\_level1\_to\_level2(file0, file1, det=5, calib=cal5 )

data\_lvl2\_D6 = foxsi\_level1\_to\_level2(file0, file1, det=6, calib=cal6 )

save, data\_lvl2\_D0, data\_lvl2\_D1, data\_lvl2\_D2, data\_lvl2\_D3, $

data\_lvl2\_D4, data\_lvl2\_D5, data\_lvl2\_d6, $

file = 'data\_2012/foxsi\_level2\_data.sav'

1. **Data structures**

Full documentation for the FOXSI data structure is given in a separate document (FOXSI Data Description).

**Caveats for FOXSI-2 data:**

There is an important difference between the FOXSI-1 and FOXSI-2 data structures. For **FOXSI-1**, we kept all triggered events in the higher-level data, even those with the HV off. This was so that preflight data could be used for debugging and troubleshooting as the analysis code was written and the flight data was understood. These data prior to the bias voltage reaching 200V are not useful for analysis, so they are not included in the **FOXSI-2** Level 1 and Level 2 data. As a result, the IDL save files are much smaller. These extra events are still included in the FOXSI-1 higher level data in order that Lindsay’s older codes will still work for that data.

For FOXSI-2, as of this writing (Jan 6 2015), we have not yet received finalized SPARCS pointing positions and times. The times and positions in foxsi-setup-script-2014 are best guesses at present, based on our written records and eyeballed from the data.

There is a 36 second offset in the WSMR recorded data. The source of this offset is unknown. For now, 36 seconds are subtracted in the setup file from the nominal times.

1. **Producing data products**

This is the fun part! Just remember to run @foxsi-setup-script-2014 (or ditto for 2012) before trying any of these.

* 1. **Lightcurves**

The basic routine for generating lightcurves is foxsi\_lc.pro. Pass to this routine a Level 2 data structure and tell it a time interval to integrate over (DT, default 10 seconds). The return variable is a structure containing arrays of times and count rates (in counts per second). The default energy range is [4,15] keV.

Example for a FOXSI-2 lightcurve:

dt = 5. ; time interval over which to integrate

lc0 = foxsi\_lc( data\_lvl2\_d0, year=2014, dt=dt)

lc1 = foxsi\_lc( data\_lvl2\_d1, year=2014, dt=dt)

lc2 = foxsi\_lc( data\_lvl2\_d2, year=2014, dt=dt)

lc3 = foxsi\_lc( data\_lvl2\_d3, year=2014, dt=dt)

lc4 = foxsi\_lc( data\_lvl2\_d4, year=2014, dt=dt)

lc5 = foxsi\_lc( data\_lvl2\_d5, year=2014, dt=dt)

lc6 = foxsi\_lc( data\_lvl2\_d6, year=2014, dt=dt)

lc0.time -= 36 ; this corrects for the 36-sec offset in WSMR data.

lc1.time -= 36

lc2.time -= 36

lc3.time -= 36

lc4.time -= 36

lc5.time -= 36

lc6.time -= 36

loadct,5

hsi\_linecolors

utplot, lc6.time, lc6.persec, /nodata, yr=[0,100], $

charsi=1.2, charth=2, xth=5, yth=5, ytit='Counts s!U-1!N', title='FOXSI 2014'

outplot, lc0.time, lc0.persec, psym=10, col=6, th=4

outplot, lc1.time, lc1.persec, psym=10, col=7, th=4

;outplot, lc2.time, lc2.persec, psym=10, col=8, th=4

outplot, lc3.time, lc3.persec, psym=10, col=9, th=4

outplot, lc4.time, lc4.persec, psym=10, col=10, th=4

outplot, lc5.time, lc5.persec, psym=10, col=12, th=4

outplot, lc6.time, lc6.persec, psym=10, col=2, th=4

al\_legend, ['D0','D1','D3','D4','D5','D6'], /right, /top, box=0, $

textcol=[6,7,9,10,12,2]

* 1. **Spectra**

Coming soon…

* 1. **Images**
     1. **Basic images**

The basic routine for generating FOXSI images is foxsi\_det\_image.pro. This image can then be transferred into a plot\_map with correct pointing and adjusted for detector rotation, etc, using easily referenced parameters from the setup script. The energy range, time range, and n-side threshold can be chosen if desired.

The following example produces a FOXSI-2 image of the first target (after pointing adjustment) from detector 6. Note that this example applies a rough pointing offset that was obtained by comparing detector images with AIA 94A. (This offset should probably be improved.) That offset is stored in the setup script.

; Choose the time range and location.

trange = [t1\_adj2, t1\_end] ; time range

xc = cen1[0] ; coords for Target 1

yc = cen1[1]

; Basic image production

image6 = foxsi\_image\_det( data\_lvl2\_d6, year=2014, trange=trange, $ erange=[4.,15.], thr\_n=4. )

map6 = make\_map( image6, dx=7.78, dy=7.78, xcen=xc, ycen=yc )

; Apply a coarse offset gleaned from comparing images with AIA.

map6 = shift\_map( map6, shift6[0], shift6[1] )

; Rotate the image based on the rotation angle for that specific detector.

map6 = rot\_map( map6, rot6 )

map6.roll\_angle = 0

map6.roll\_center = 0

loadct, 5

plot\_map, map6

plot\_map, map6, /log

* + 1. Deconvolved images

Coming soon…