

Miniature X-ray Solar Spectrometer (MinXSS) CubeSat

Level 1 Science Data Description

Version 1.0

Contacts:

Principal Investigator: tom.woods@lasp.colorado.edu

Data access: james.mason@lasp.colorado.edu

MinXSS data processing code authors:

Thomas N. Woods (tom.woods@lasp.colorado.edu)

James Paul Mason (james.mason@lasp.colorado.edu)

Christopher Moore (christopher.moore-1@colorado.edu)

Amir Caspi (amir@boulder.swri.edu)



Deployment of the MinXSS and CADRE CubeSats from the International Space Station on 2016 May 16. Image credit: Tim Peake NASA/ESA

Table of Contents

INTRODUCTION TO MINXSS AND ITS SCIENCE INSTRUMENTS	3
INTERNAL DATA PROCESSING LEVELS	4
WHAT'S IN LEVEL 1 DATA	5
EXAMPLES (IDL AND PYTHON)	9
LOADING DATA	9
PLOTTING DATA	10
1) <i>A single spectrum</i>	10
2) <i>A time series of a single bin</i>	11

Introduction to MinXSS and its Science Instruments

MinXSS is a 3U CubeSat designed, built, tested, and operated from the University of Colorado at Boulder (CU) and the Laboratory for Atmospheric and Space Physics (LASP). Its size is determined by the standard 3U form factor: 34 cm x 10 cm x 10 cm, though the dimensions change slightly when the two solar arrays deploy. It weighs only 3.5 kg.

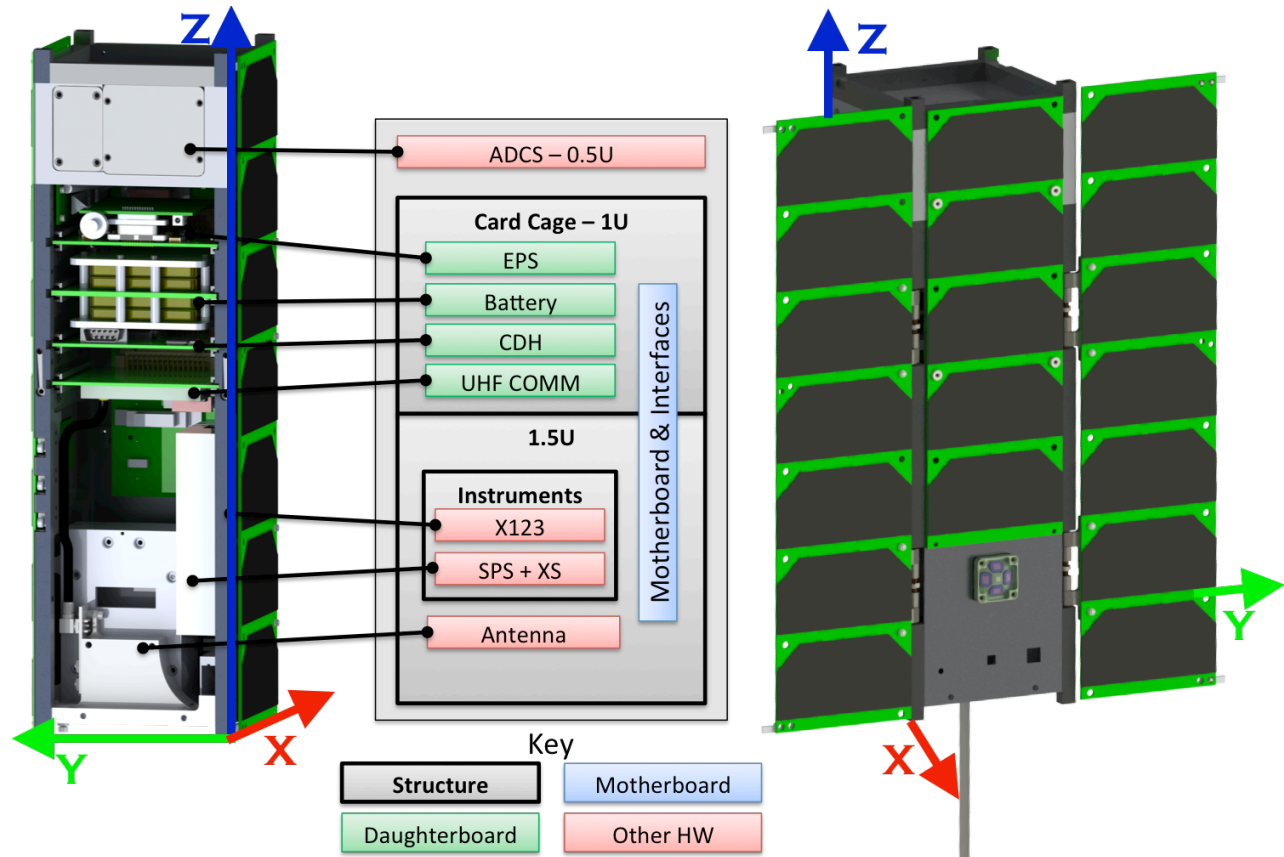


Figure 1: MinXSS mechanical block diagram. Left: stowed configuration. Right: deployed configuration. ADCS = attitude determination and control system; EPS = electrical power system; CDH = command and data handling; UHF COMM = ultra high frequency communications; X123 = primary science instrument; SPS = sun position sensor; XS or XP = X-ray sensor or X-ray photometer.

Two flight MinXSS CubeSats were built. The first one deployed from the International Space Station on 2016 May 16 (captured on camera as shown on title page). Routine science data collection began on 2016 June 8. The expected lifetime from this orbit is anywhere between 4-18 months depending on solar-influenced Earth atmospheric drag. MinXSS-2 will be launched in early 2017 into a higher orbit expected to last about 5 years. For more description of the MinXSS spacecraft, see [Mason, et al. 2016 \(doi: 10.2514/1.A33351\)](#).

The science instruments on MinXSS are the X123, SPS, and XP. The X123 is the primary instrument – a modified commercial-off-the-shelf silicon drift detector soft x-ray spectrometer from Amptek. It measures the solar spectrum from about 0.4 – 30 keV. The spectral resolution varies between 0.15 – ~0.3 keV FWHM across the spectral range 5.9 -> ~ 30 keV. Resolution between 0.4 and 5.9 is still being determined. Photons are counted and put into one of the 1024 energy bins. The SPS provides fine pointing knowledge relative to the sun with about 2 arcsec

precision. The XP measures approximately the same spectral range as the X123 but is spectrally integrated to provide a single number for the band. All of these instruments have been calibrated at the National Institute of Standards and Technology (NIST) Synchrotron Ultraviolet Radiation Facility (SURF). For more details on the instrument and calibration, see [Moore, et al. 2016 \(doi: 10.1117/12.2231945\)](https://doi.org/10.1117/12.2231945).

Internal Data Processing Levels

At this time, we're not releasing every level of data product. By the end of the mission, we'll release Level 0B and up. The raw telemetry won't be released because it contains some telemetry that may be restricted by the International Traffic and in Arms Regulations (ITAR), though we doubt it. Nevertheless, we err on the side of caution and do not release the raw telemetry or the means to decode it. Below is a description of each of the levels so that you can remain informed until we release the lower level products, which will have their own description documents to accompany them.

Raw telemetry: This is the binary we receive over the radio frequency (RF) link from the spacecraft. It contains all of the housekeeping and science data. It can't be interpreted without detailed knowledge of the packet structure.

Level 0B: Here, we interpret the raw binary data and save it as a per-day IDL saveset with several arrays of structures. There is a separate structure for each type of telemetry packet we have defined in MinXSS flight software (housekeeping, log messages, science packets, and four separate attitude determination and control system packets to accommodate the numerous telemetry points). Each individual structure is concatenated into an array corresponding to the index of when that packet was received. This is not necessarily ordered in time since we often request the spacecraft to send us data from days or weeks ago.

Level 0C: This product is the result of time-sorting the Level 0B data. Packets get put into the order they were generated rather than the order they were received on the ground. The final result is another series of per-day IDL savesets.

Level 0D: This is the final stage of low-level data processing. It cleans up the MinXSS telemetry and adds ancillary data in preparation for processing to a usable science product at Level 1. The variety of MinXSS packet types (housekeeping, log, etc.) are not, in general, generated at the same time as the science packet, nor do we necessarily downlink the packets that were generated nearest in time to each other. However, some of that non-science telemetry is needed for the conversion to Level 1 and higher. We interpolate the non-science packet telemetry to the mid-point of the science integration periods, provided that the time step does not exceed a given threshold (presently 5 minutes for ADCS data and 1 minute for all other types). We also generate ancillary data such as the orbital position and the spacecraft's relation to the sun based on our best estimate from an interpolated combination of the publically available two line elements. All of these data are merged into a single structure at each time interval, resulting in a single array of identically-defined structures. Level 0D will be released to the public soon, as well as the IDL code we use to process it to Level 1. That will enable users to e.g., change the threshold on time interpolation of telemetry.

What's in Level 1 Data

The publically available Level 1 MinXSS data focuses on the science product and telemetry related to the science. The raw count spectrum from Level 0D is converted into irradiance units and NIST/SURF calibration applied. The raw 10-second spectra are averaged into 1 minute. The resultant array of structures (variable name = *minxsslevel1*) is stored as an IDL saveset, which can be directly restored in IDL or read into python with the [scipy.io.readsav function](#) (see examples section below). The definition of the structure is provided in the table below.

Note 1: The bad data flag for bytes is 255, i.e., byte(-1). For strings it's empty, i.e., ''. For all other types it's NaN.

Note 2: The primary science instrument is called the X123. The secondary science instruments are the SPS and XP.

Table 1: Level 1 structure definition

Tag Name	Units	Description
flight_model	--	1: MinXSS-1 in ISS orbit from 2016/05/16 - present 2: MinXSS-2 in 500 km, 97° inclination, 13:50 MLTDN sun-sync orbit from 2017/01/?? - ??
irradiance	photons / sec / cm ² / keV (photons per second per square-centimeter per kiloelectronvolt)	Float array (1024 elements). The irradiance value in each of the bins. The bins themselves are defined by the energy variable listed below.
energy	keV (kiloelectronvolt)	Float array (1024 elements). The center of the bins for the spectral measurement in the irradiance variable above. The range is about 0 to 30 keV.
accuracy	Relative *100 = %	Float array (1024 elements). Estimated irradiance accuracy in each of the bins, including pre-flight calibration and measurement precision.
precision	Relative *100 = %	Float array (1024 elements). Measurement precision for each of the bins, defined as 1 / sqrt(counts).
stddev	Relative *100 = %	Float array (1024 elements). Standard deviation of the irradiance in each of the bins combined into the 1-minute average. At most 6 spectra are combined into this average. If only one spectrum is used, then this tag has the same value as the precision tag.
irradiance_low	Same as irradiance	Float array (1024 elements). This version of the irradiance calculation does include solar photoelectron contribution and thus yields

		lower irradiance in the low energy bins below 1 keV.
irradiance_high	Same as irradiance	Float array (1024 elements). This version of the irradiance calculation does not include photoelectron and other contributions. Thus this calculation yields higher irradiance in the low energy bins below 1 keV. We recommend users use the irradiance tag for science analysis and to restrict science analysis to above 0.8 keV. Below 0.8 keV the irradiance_low and irradiance_high results severely diverge.
signal_cps	cps (counts per second)	Float array (1024 elements). The raw counts used in the irradiance conversion.
valid_flag	--	Float array (1024 elements). 0: If irradiance_high and irradiance_low disagree by more than 10%. Data here should be analyzed with extreme caution. 1: If irradiance_high and irradiance_low agree by more than 10%.
integration_time	sec (seconds)	Integration time within the 1-min window. It can range from 10 sec for one X123 spectrum to about 60 sec.
number_spectra	--	The number of spectra used in the 1-minute average. Range is 1-6.
x123_fast_count	DN / sec (Data numbers per second)	X123 total signal from its fast amplifier.
x123_slow_count	DN/sec (Data numbers per second)	X123 total signal from its slow amplifier. The sum of the signal_cps array should equal the X123_slow_count value.
sps_on	--	0: SPS off. 1: SPS on.
sps_sum	fC (femtoCoulombs)	SPS total signal from four quadrants.

sps_x	° (degrees)	Offset from SPS optical center. The sps_x_hk is also provided but it is not as accurate as the sps_x tag.
sps_y	° (degrees)	Offset from SPS optical center. The sps_y_hk is also provided but it is not as accurate as the sps_y tag.
longitude	° (degrees)	Spacecraft location relative to Earth.
latitude		
altitude	km (kilometers)	Spacecraft height above the earth's surface
sun_right_ascension	° (degrees)	Relative angle to sun as determined from ground computation based on TLE.
sun_declination		
earth_sun_distance	AU (astronomical units)	Earth's orbit around the sun is an ellipse so irradiance changes slightly due to the $1/r^2$ dispersion of light.
spacecraft_in_saa	--	0: Spacecraft is not in the South Atlantic Anomaly 1: Spacecraft is in South Atlantic Anomaly (which may cause noise in science measurements)
Time (sub-structure) 1. iso 2. human 3. yyyyymmdd 4. yyyydoy 5. hhmmss 6. sod 7. fod 8. jd 9. TAI 10. spacecraftgpsformat	--	Time in a variety of formats (note: none of these have been corrected for spacecraft clock drift, which is expected to be only a few seconds over the whole mission) 1. International standards organization time format (yyyy-mm-ddThh:mm:ssZ) 2. yyyy-mm-dd hh:mm:ss 3. Year month day 4. Year day of year 5. Hours minutes seconds in day 6. Second of day 7. Fraction of day 8. Julian date 9. International Atomic Time 10. Flight clock in GPS format

Examples (IDL and Python)

Loading data

First, simply download the [mission-length level 1 MinXSS data \(minxss1_l1_mission_length.sav\)](#). This link takes you to the file that is regenerated daily that folds in all new telemetry received. That processing starts at midnight UTC. It takes ~20 minutes to complete processing, so you can be sure that the file is updated by 01:00 UTC every day.

IDLDE

You can simply double click the `minxss1_l1_mission_length.sav` file and it will load the `minxsslevel1` variable. You can alternatively load the data in the interactive console using the IDL console method below.

IDL console

```
IDL> restore, path + path_sep() + 'minxss1_l1_mission_length.sav'
```

where *path* is the path to the file. The built-in `path_sep()` function is used above to return a `/` for Linux/Mac (i.e., Unix) systems and a `\` for Windows systems.

Python (requires scipy)

```
from scipy.io.idl import readsav
data = readsav('path/minxss1_l1_mission_length.sav')
minxsslevel1 = data.minxsslevel1.copy()
```

where *path* is the path to the file. This generates a dictionary (*data*) containing the variables *minxsslevel1* and *minxsslevel1meta*. We copy *minxsslevel1* out for convenience, so that you don't have to call `data.minxsslevel1` thereafter.

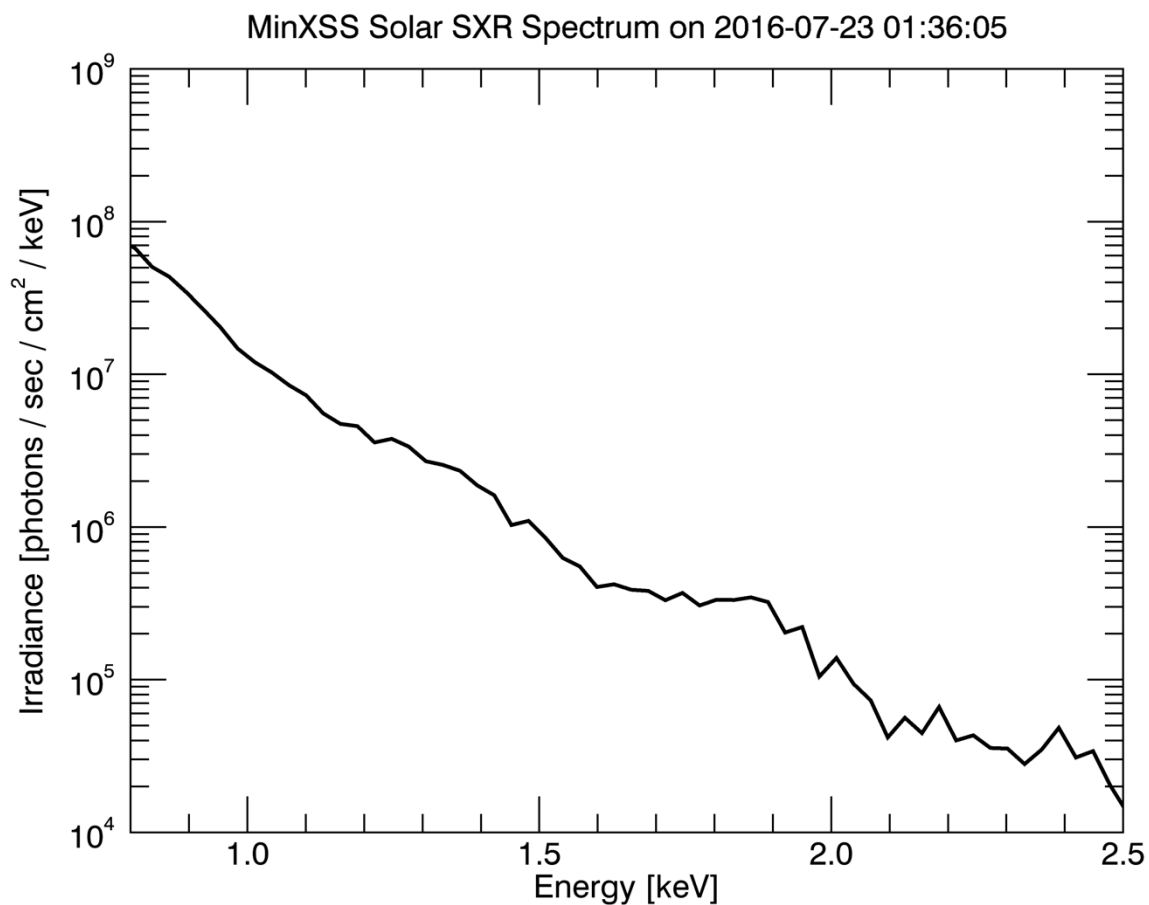
Plotting data

1) A single spectrum

IDL

```
p = plot(minxsslevel1[2060].energy, minxsslevel1[2060].irradiance, $
        TITLE = 'MinXSS Solar SXR Spectrum on ' + $
                minxsslevel1[2060].time.human, $
        XTITLE = 'Energy [keV]', XRANGE = [0.8, 2.5], $
        YTITLE = 'Irradiance [photons / sec / cm2 / keV]', $
        YRANGE = [1e4, 1e9], /YLOG)
```

which should result in this plot:



An IDL procedure that loads the data and creates the above plot can be downloaded [here](#).

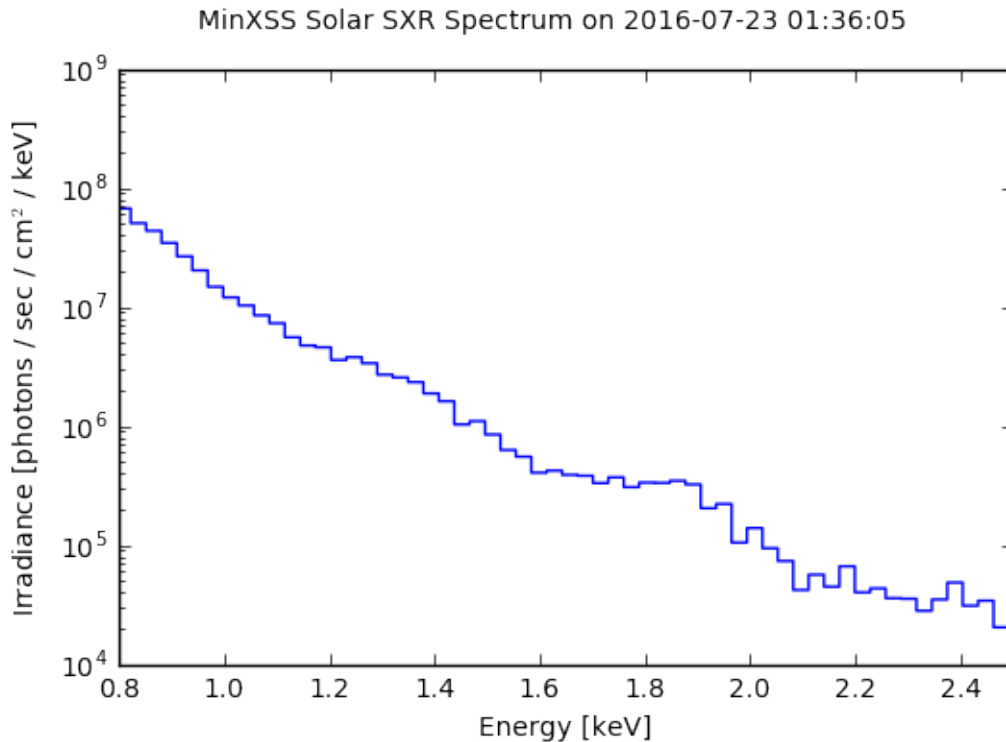
Python

```
%matplotlib inline
import matplotlib.pyplot as plt

plt.plot(minxsslevel1[2060]['energy'], minxsslevel1[2060]['irradiance'], drawstyle = 'steps-mid')

plt.xlim([0.8, 2.5])
plt.xlabel('Energy [keV]')
plt.ylim([1e4, 1e9])
plt.yscale('log')
plt.ylabel('Irradiance [photons / sec / cm$^2$ / keV]')
plt.suptitle('MinXSS Solar SXR Spectrum on ' + minxsslevel1[2060]['time']['human'][0].decode("utf-8"))
plt.show()
```

which should result in this plot:



A python notebook that loads the data and creates the above plot can be downloaded [here](#) and the raw python code can be found [here](#).

2) A time series of a single bin

You would normally want to do this for an emission line. An emission line may cover multiple bins or multiple emission lines may fill an individual bin. For simplicity, we show you the principle here by plotting just a single bin.

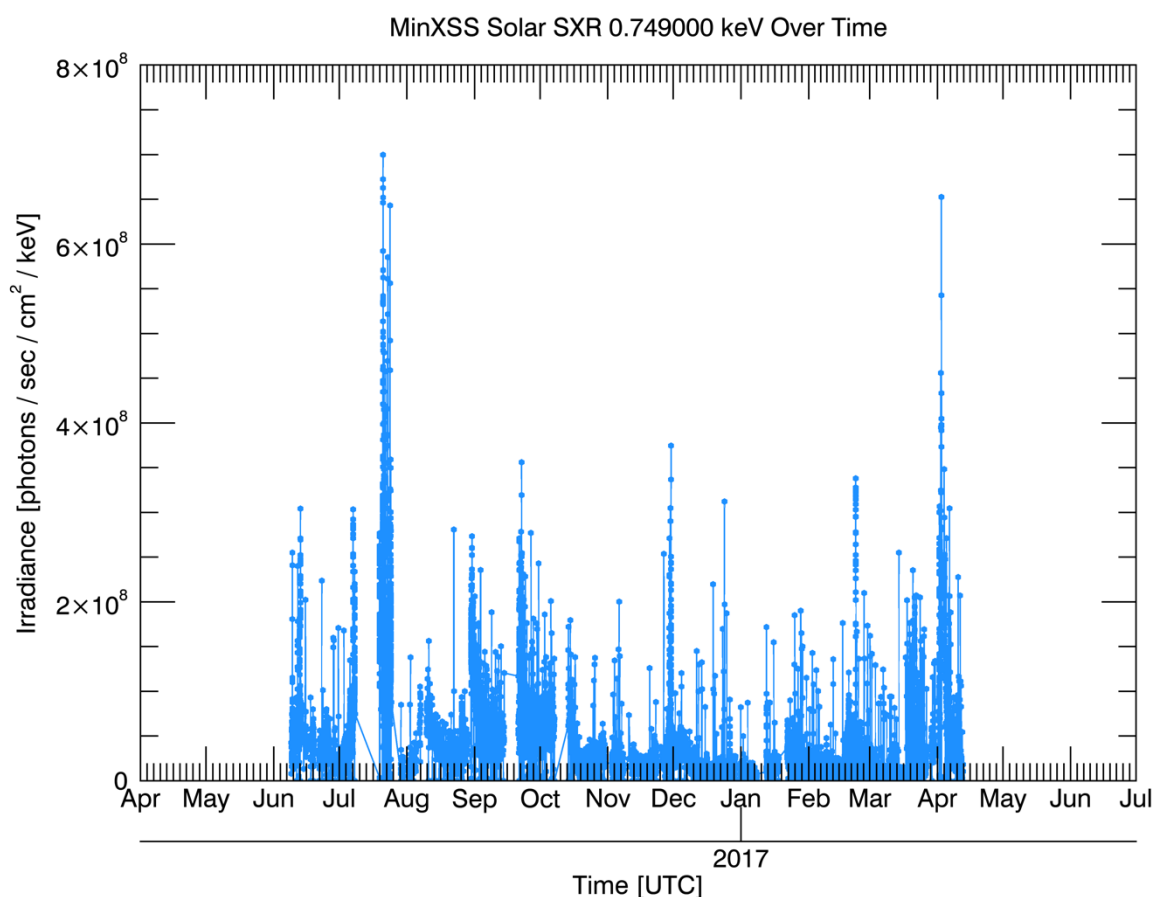
IDL

```

labelDate = label_date(DATE_FORMAT = ['%M', '%Y'])
p1 = plot(minxsslevel1.time_jd, minxsslevel1.irradiance[30], $
          SYMBOL = 'dot', SYM_THICK = 3, COLOR = 'dodger blue', $
          TITLE = 'MinXSS Solar SXR ' + $
                 strtrim(minxsslevel1[0].energy[30], 2) + ' keV Over Time', $
          XTITLE = 'Time [UTC]', $
          XTICKFORMAT = ['LABEL_DATE', 'LABEL_DATE'], $
          XTICKUNITS = ['Month', 'Year'], $
          XTICKINTERVAL = 1, $
          YTITLE = 'Irradiance [photons / sec / cm$^2$ / keV]')

```

which should result in this plot:



An IDL procedure that loads the data and creates the above plot can be downloaded [here](#).

Python

```

%matplotlib inline
from scipy.io.idl import readsav
import matplotlib.pyplot as plt
import numpy as np
import datetime

```

```

import matplotlib.dates as mdates

# Extract irradiance for the 30th bin corresponding to energy 0.75 keV
irradiance = minxsslevel1['irradiance']
irradianceBin30 = np.zeros(len(irradiance))
for i in range(len(irradiance)):
    irradianceBin30[i] = irradiance[i][30]

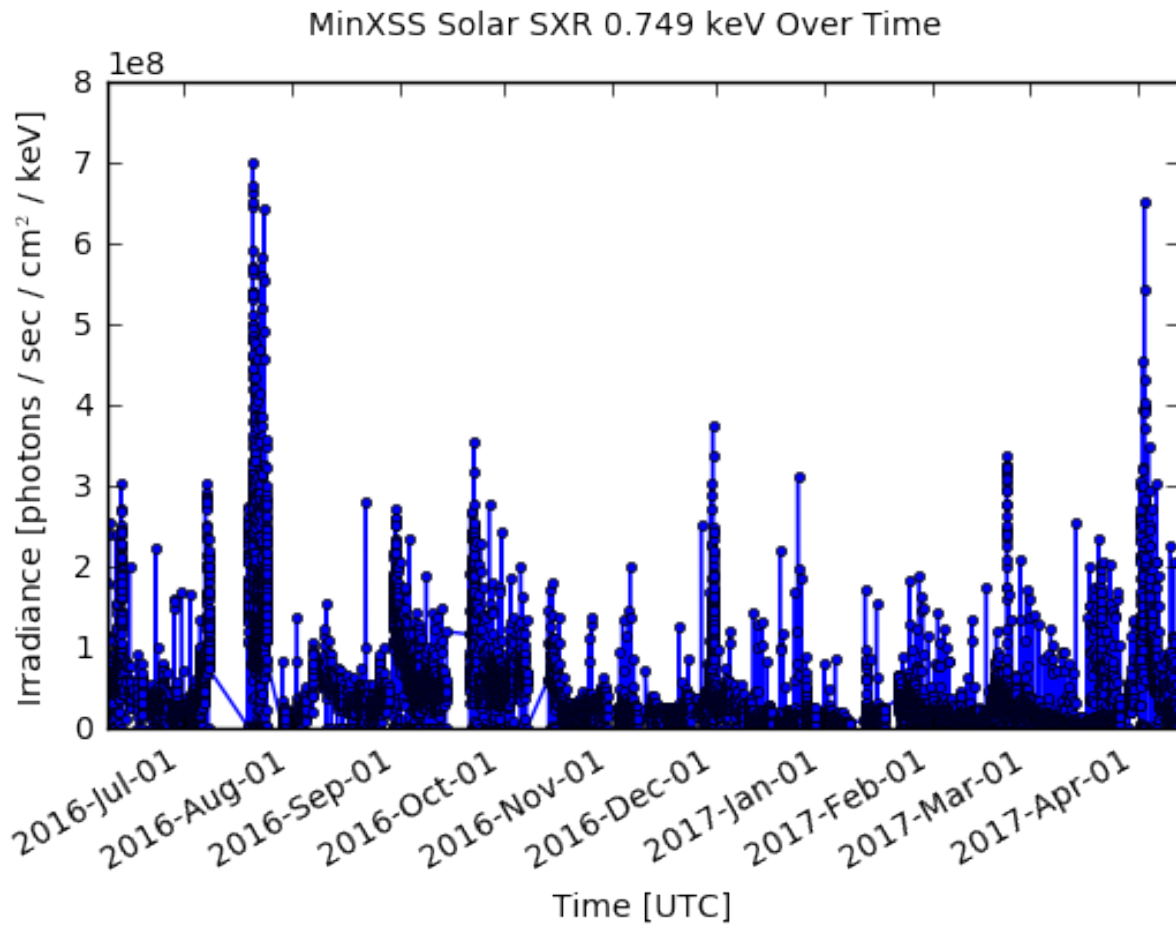
# Extract time and convert it so that matplotlib can understand it
time_human = []
for i in range(len(irradiance)):
    time_human.append(minxsslevel1[i]['time']['human'][0].decode("utf-8"))

datetimes = [datetime.datetime.strptime(t, "%Y-%m-%d %H:%M:%S") for t in ti
me_human]

# Make the plot
fig, ax = plt.subplots(1)
plt.plot(datetimes, irradianceBin30, 'b-o', markersize = 3)
fig.autofmt_xdate()
ax.xaxis.set_major_formatter(mdates.DateFormatter("%Y-%b-%d"))
plt.xlabel('Time [UTC]')
plt.ylabel('Irradiance [photons / sec / cm$^2$ / keV]')
plt.ylim([0, 8e8])
plt.suptitle('MinXSS Solar SXR ' + str(minxsslevel1[0]['energy'][30]) + ' k
eV Over Time')
plt.show()

```

which should result in this plot:



A python notebook that loads the data and creates the above plot can be downloaded [here](#) and the raw python code can be found [here](#).