

#	Sample Code	Descript
0	<pre>#!/usr/bin/env python ##### # # This program extract variables information and # image_pixel_values data(Albedo) of # GK2A Visible Ch.4 Level 1B data in netCDF4 format # # Input : GK2A L1B file [sample file : VI004/fd010ge] (netCDF4) # GK2A calibration table(netCDF4) # # Output : sample_output.txt (ASCII) # sample_image_pixel_data.npy (numpy format) # #####</pre>	Program Header
1	<pre>import netCDF4 import numpy</pre>	Library install
2	<pre>input_ncfile_path = 'gk2a_ami_le1b_vi004_fd010ge_201901260310.nc' CT_path='./calibration_table/' output_txt_path='sample_output.txt' output_npy_path='sample_image_pixel_data.npy'</pre>	<p>Input DATA</p> <p>Output data file name (ASCII)</p> <p>Output data file name (npy binary)</p>
3	<pre>ncfile = netCDF4.Dataset(input_ncfile_path,'r',format='netcdf4') ipixel=ncfile.variables['image_pixel_values'] ipixel_process = ipixel[:]</pre>	<p>sample data (*.nc) file leading</p> <p>image_pixel_values leading.</p> <p>Procession load data.</p>
4	<pre>number_of_error_pixels = ipixel.getncattr('number_of_error_pixels') if (number_of_error_pixels > 0): ipixel_process[ipixel_process>49151] = 0</pre>	DQF value error pixel. (0 value filter)
5	<pre>channel=ipixel.getncattr('channel_name') if ((channel == 'VI004') or (channel == 'VI005') or (channel == 'NR016')): mask = 0b0000011111111111 #11bit mask elif ((channel == 'VI006') or (channel == 'NR013') or (channel == 'WV063')): mask = 0b0000111111111111 #12bit mask elif (channel == 'SW038'): mask = 0b0011111111111111 #14bit mask else: mask = 0b0001111111111111 #13bit mask ipixel_process_masked=numpy.bitwise_and(ipixel_process,mask)</pre>	Channel case by processing bit mask and pixel value and bit size.

6	<pre> AL_postfix='_cal_alb.txt' BT_postfix='_cal_bt.txt' if (channel[0:2] == 'VI') or (channel[0:2] == 'NR'): calibration_table=numpy.loadtxt(CT_path+channel+AL_postfix,'float64') else: calibration_table=numpy.loadtxt(CT_path+channel+BT_postfix,'float64') ipixel_process_masked_converted=calibration_table[ipixel_process_masked] </pre>	Albedo/BT Change
7	<pre> output_txt=open(output_txt_path,'wt') var_keys = ncfile.variables.keys() output_txt.write("number of variable keys : %s\n" %len(var_keys)) for i in range(len(var_keys)): output_txt.write("%s\t%s\t%s\t%s\n" %(i,var_keys[i],ncfile.variables[var_keys[i]]. dimensions,ncfile.variables[var_keys[i]].dtype)) output_txt.close() print "variable list wrote to sample_output.txt" </pre>	Output file write.
8	<pre> numpy.save(output_npy_path,ipixel_process_masked_converted,False,False) </pre>	File save. (Approximately 900MB)
9	<pre> ##### #End of Program ##### </pre>	End