

천리안 위성 2A(GK2A)의 데이터 표출

자세한 내용은 이상우의 IDL 블로그를 참고하십시오.

이 글은 이상우의 IDL 블로그(blog.daum.net/swrush)에 있는 천리안위성 2A호(GK2A)의 데이터 표출 [1]과 [2]를 요약한 것입니다. 자세한 내용은 원문을 통해 확인하실 수 있습니다. *

동아시아 영역과 한반도 영역

```
gk2a_ami_le1b_ir112_ea020lc_202002280300.nc
gk2a_ami_le1b_ir112_ko020lc_202002280300.nc
```

위와 같이 파일 이름에서 동아시아 영역(ea)과 한반도 영역(ko)를 구분할 수 있습니다. netCDF 포맷으로 제공됩니다. netCDF 포맷의 데이터를 읽고 표출하는 방법에 대해 자세한 내용은 역시 이상우의 IDL 블로그에서 보실 수 있습니다.**

netCDF로 부터 정보 확보

시작은 무조건 NCDF_LIST 라고 생각하십시오.

```
file = 'gk2a_ami_le1b_ir112_ko020lc_202002280300.nc'
```

```
NCDF_LIST, file, /VARIABLES
```

```
gk2a_ami_le1b_ir112_ko020lc_202002280300.nc
# dimensions: 2
# Variables: 1
# Global attributes: 33
There are no unlimited dimensions.
Variables
  0 image_pixel_values: UINT(900,900)=UINT(dim_x,dim_y)
```

2차원 배열 변수가 하나가 있, 전역 속성이 33개가 있고, 2차원 배열의 크기는 900*900 인 것을 알 수 있습니다. 전역 속성이 중요하겠네요.

```
NCDF_LIST, file, /GATT ; 결과는 오른쪽에
```

netCDF에서 필요 정보 읽기

```
id = NCDF_OPEN(file)
NCDF_VARGET, id, 'image_pixel_values', img
NCDF_ATTGET, id, 'origin_latitude', latc, /GLOBAL
NCDF_ATTGET, id, 'central_meridian', lonc, /GLOBAL
NCDF_ATTGET, id, 'standard_parallel1', latsp1, /GLOBAL
NCDF_ATTGET, id, 'standard_parallel2', latsp2, /GLOBAL
NCDF_ATTGET, id, 'upper_left_easting', x_ul, /GLOBAL
NCDF_ATTGET, id, 'upper_right_easting', x_ur, /GLOBAL
NCDF_ATTGET, id, 'upper_left_northing', y_ul, /GLOBAL
NCDF_ATTGET, id, 'lower_left_northing', y_ll, /GLOBAL
NCDF_ATTGET, id, 'image_width', nx, /GLOBAL
NCDF_ATTGET, id, 'image_height', ny, /GLOBAL
NCDF_ATTGET, id, 'pixel_size', dx, /GLOBAL
NCDF_ATTGET, id, 'pixel_size', dy, /GLOBAL
NCDF_ATTGET, id, 'file_name', fname, /GLOBAL
NCDF_CLOSE, id
```

image_pixel_values는 변수이기 때문에 NCDF_VARGET을 사용하고, 나머지는 속성이기 때문에 NCDF_ATTGET입니다.

한반도 영역 파일의 전역 속성

```
Global Attributes
0 _NCProperties:
  version=1|netcdflibversion=4.4.1.1|hdf5libversion=1.8.14
1 projection_type: lambert_conformal_conic
2 standard_parallel1: 30.000000
3 standard_parallel2: 60.000000
4 origin_latitude: 38.000000
5 central_meridian: 126.000000
6 false_easting: 0.0000000
7 false_northing: 0.0000000
8 image_width: 900
9 image_height: 900
10 pixel_size: 2000.0000
11 upper_left_easting: -899000.00
12 upper_left_northing: 899000.00
13 upper_right_easting: 899000.00
14 upper_right_northing: 899000.00
15 lower_left_easting: -899000.00
16 lower_left_northing: -899000.00
17 lower_right_easting: 899000.00
18 lower_right_northing: -899000.00
19 _CoordinateTransformType: 0.0000000
20 _CoordinateAxisTypes: 0.0000000
21 file_name:
  gk2a_ami_le1b_ir112_ko020lc_202002280300.nc
22 origin_source_file:
  gk2a_ami_le1b_ir112_ea020ge_202002280300.nc
23 number_of_columns: 900
24 number_of_lines: 900
25 total_pixel_data_size: 810000
26 channel_center_wavelength: 11.2
27 channel_spatial_resolution: 2.0
28 data_processing_center: NMSC
29 data_processing_mode: operation
30 file_format_version: 1.0.0_20181120
31 instrument_name: AMI
32 satellite_name: GK-2A
```

여기에 나오는 전역속성 숫자를 표출과 관련한 코드에 직접 써 넣어도 되지만, NCDF_ATTGET으로 읽는 코드를 한번 작성해 두면, 저 값들이 바뀌는 것에 문제 없이 대응할 수 있게 됩니다.

Lambert Conformal Conic 투영 표출

다음 페이지에 설명하는 LCC 투영 데이터의 표출에 대해 조금 더 자세히 보시려면, 이상우의 IDL 블로그에서 'GRIB 포맷의 파일을 읽고 데이터를 표출하기'를 검토해 보세요.**

* 천리안위성 2A호(GK2A)의 데이터 표출 [1][2]
** netCDF 포맷의 데이터를 읽고 표출하기 [1][2]
*** GRIB 포맷의 파일을 읽고 데이터를 표출하기

<http://blog.daum.net/swrush/521>
<http://blog.daum.net/swrush/468>
<http://blog.daum.net/swrush/465>

<http://blog.daum.net/swrush/522>
<http://blog.daum.net/swrush/469>

지도 좌표계 세팅 (Map Projection)

```
limit = [29, 114, 47, 138]
win = WINDOW(DIMENSIONS=[600, 600], /NO_TOOLBAR)
m = MAP('Lambert Conformal Conic', LIMIT=limit, $
  STANDARD_PAR1=latsp1, STANDARD_PAR2=latsp2, $
  CENTER_LONGITUDE=lonc, CENTER_LATITUDE=latc, $
  HORIZON_LINestyle=0, HORIZON_THICK=2, $
  Font_Size=11, CLIP=0, MARGIN=0.08, /CURRENT)
m.MapGrid.GRID_LONGITUDE = 5
m.MapGrid.GRID_LATITUDE = 5
m.MapGrid.LINestyle = 2
m.MapGrid.LABEL_POSITION = 0
```

앞서 NCDF_ATTGET으로 읽은 변수들이 모두 사용되고 있습니다. 어떤 상태로 투영되었는지 한번 보시죠.

```
mc = MAPCONTINENTS(/HIRES, COLOR='orange')
```

한반도 영역 자료의 표출

Lambert Conformal Conic 투영 데이터는 아래와 같은 코드를 그대로 사용하면 됩니다. nx, ny, dx, dy, x_ul, y_ll 역시 NCDF_ATTGET으로 가져온 내용이란 점을 확인하세요.

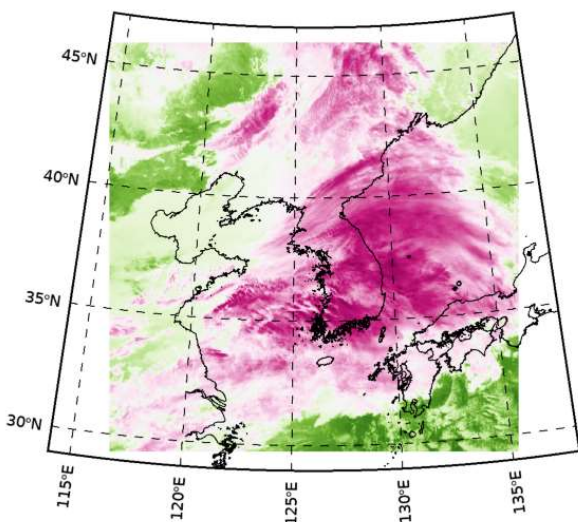
```
xs = FINDGEN(nx)*dx+x_ul
ys = FINDGEN(ny)*dy+y_ll
ctnum = COLORTABLE(67, /REVERSE)
```

```
im = IMAGE(REVERSE(img, 2), xs, ys, RGB_TABLE=ctnum, $
  GRID_UNITS=1, /OVERPLOT)
mc = MAPCONTINENTS(/HIRES, COLOR='black')
tx = TEXT(0.5, 0.94, strjoin(string(fname)), COLOR='blue', $
  ALIGNMENT=0.5, FONT_SIZE=14, /NORMAL)
```

원본 데이터는 IDL 기준으로 상하가 뒤집혀 있습니다. 그래서 REVERSE 함수를 사용하여 Y축 방향을 기준으로 뒤집는 처리를 하였습니다. 배열의 격자가 거리 단위(위경도 단위가 아닌)이기 때문에 GRID_UNITS=1 을 이용합니다.

본문에서는 67번 컬러테이블을 뒤집어서 사용하였는데, 다양한 컬러테이블을 시도해 볼 수 있습니다.

gk2a_ami_le1b_ir112_ko020lc_202002280300.nc



동아시아 영역 자료의 표출

방법이 사실상 같습니다. 데이터의 크기와 전역 속성이 다르다는 점(당연하게도)만 감안하여 처리하면 됩니다.

```
file = 'gk2a_ami_le1b_ir112_ea020lc_202002280300.nc'
```

앞에서와 같이 NCDF_LIST로 데이터와 속성을 확인하고, 같은 방법으로 데이터와 속성을 읽습니다(NCDF_VARGET 과 NCDF_ATTGET)

읽어온 정보를 이용하여 다음과 같이 표출하는 과정은 한반도 영역을 표출할 때와 거의 같습니다.

다만, 영역이 넓어졌으므로, LIMIT의 범위를 크게 잡았고 (동경 75~177도, 북위 10~65도), 화면의 크기도 [800,500]으로 비율을 달리 하였습니다. 관련 코드는 다음과 같습니다. NCDF_VARGET과 NCDF_ATTGET에서 같은 이름의 변수로 값을 읽어오기 때문에, LIMIT와 WINDOW의 DIMENSIONS 외에는 같은 코드를 그대로 쓸 수 있습니다.

```
limit = [10, 75, 65, 177]
win = WINDOW(DIMENSIONS= [800, 500] , /NO_TOOLBAR)
m = MAP('Lambert Conformal Conic', LIMIT=limit, $
  STANDARD_PAR1=latsp1, STANDARD_PAR2=latsp2, $
  CENTER_LONGITUDE=lonc, CENTER_LATITUDE=latc, $
  HORIZON_LINestyle=0, HORIZON_THICK=2, $
  Font_Size=11, CLIP=0, MARGIN=0.08, /CURRENT)
m.MapGrid.GRID_LONGITUDE = 10
m.MapGrid.GRID_LATITUDE = 10
m.MapGrid.LINestyle = 2
m.MapGrid.LABEL_POSITION = 0
```

```
xs = FINDGEN(nx)*dx+x_ul
ys = FINDGEN(ny)*dy+y_ll
ctnum = COLORTABLE(67, /REVERSE)
```

```
im = IMAGE(REVERSE(img, 2), xs, ys, RGB_TABLE=ctnum, $
  GRID_UNITS=1, /OVERPLOT)
mc = MAPCONTINENTS(/HIRES, COLOR='black')
```

```
tx = TEXT(0.5, 0.94, strjoin(string(fname)), COLOR='blue', $
  ALIGNMENT=0.5, FONT_SIZE=14, /NORMAL)
```

gk2a_ami_le1b_ir112_ea020lc_202002280300.nc

