(1) Pre-Processing: (./Src_IDL) 2
(1.a) extract_msu_coef.pro 2
(1.b) extract_amsu_bt.pro 3
(1.c) extract_gpsro_profiles.pro 4
(1.d) merra_hdf2sav.pro 6
(1.e) merra_sav2txt.pro 6
(2) Simulate AMSU/MSU brightness temperature: 7
(2.a) amsu_raob.exe (./Src_fortran) 7
(2.b) amsu_merra.exe (./Src_fortran) 8
(2.c) msu_merra.exe (./Src_fortran) 10
(2.d) bin_daily_raob_amsu.pro (./Src_IDL) 11
(2.e) merra_day_nc.pro (./Src_IDL) 11
(3) Calibration of NOAA MSU using SNO method: (./Src_IDL): 12
(3.a) match_noaa_amsumsu.pro 12
(3.b) sno_step1_preparematchup_msu.pro 13
(3.c) sno_step2_soluteequation_msu.pro 14
(3.d) sno_step3_rebuild_msu.pro 15
(3.e) sno_step3_cnadir_dbin_msu.pro 15
(3.f) sno_step4_intersatbias_msu.pro 16
(3.g) sno_step4_satraobbias_msu.pro 17
(3.h) sno_step5_chrisycorr_msu.pro 18
(3.i) sno_step6_chrisyprod_msu.pro 19
(3.j) sno_step7_new_intersatbias_msu.pro 20
(3.k) sno_step8_select_msu.pro 21
(4) Calibration of NOAA AMSU using coefficients from matched data : (./Src_IDL): 21
(4.a) match_raob_noaa.pro 21
(4.b) offset_slope_raob_noaa_month.pro 23
(4.c) offset_slope_raob_noaa_month_modify.pro ---------- 24
(4.d) bin_noaa_daily.pro 25
(4.e) convert_amsu_byraob.pro 26
(5) Generate product: (./Src_IDL) 27
(5.a) msu_daily_product.pro 27
(5.b) monthly_product.pro 28
(5.c) gen_product.pro 28

```
============================================================
(1)  Pre-Processing: (./Src_IDL)
============================================================

------------------------------------------------------------
 (1.a)    extract_msu_coef.pro
------------------------------------------------------------
```

SYNOPSIS:

For the Specified NOAA polar orbiter 'noaa1', this program loops over the specified range of months and years, and reads the L1B binary files from the specified input directory. (downloaded from NOAA-CLASS TOVS from NOAA MSU) Values of counts and related data are stored for each Julian day in NetCDF files for later use.

USAGE:

For each NOAA orbiter {'noaa15',noaa16','noaa18','noaa19','noaam2'} the user needs to first edit the RUN parameters for the program to set the orbiter name, the range of days to process, and the input/output paths for the data.
The data is then processed by compiling and running the IDL program.

RUN PARAMETERS:

Polar satellite name
Beginning year/month
Ending year/month
Path to Input data
Path to Output data

INPUTS:

AMSU Level 1B files fro each {noaa} orbiter should be stored with the following directory structure and file format:

$InPath/{noaa}/YYYYMM/{noaa}/NSS.AMAX.SS.Dyyddd.Snnnn.Ennnn.CC

OUTPUTS:

Output ASCII files are stored with the following directory structure and file format:

$OutPath/{noaa}/amsu_{noaa}_YYYYMM.nc

ALGORITHM:

---

*Loop over years
*Loop over months
*Loop over days in month
-Construct the set of Input filenames
-Construct the Output filenames
*Loop over the input files for this day

```
  -Open input file
  -Read in header data
  *Loop over records
   -Read in record
   -Skip bad records
   *Loop over scans
    *Loop over channels
     -Calcualte Brightness Temperatures
     *
    -Write values to output file
   *
  *
   -Close input file
 *
*
 -Close output file
*
*
```

(1.b) extract_amsu_bt.pro

SYNOPSIS:

For the Specified NOAA polar orbiter 'noaa1', this program
loops over the specified range of months and years, and reads
the L1B binary files from the specified input directory.
(downloaded from NOAA-CLASS TOVS from from KLM 8.3.1.6.1)
Values of brightness temperature and related data are stored for each
Julian day in ASCII files for later use.

USAGE:

For each NOAA orbiter {'noaa15',noaa16','noaa18','noaa19','noaam2'}
the user needs to first edit the RUN parameters for the program to
set the orbiter name, the range of days to process, and the
input/output paths for the data.
The data is then processed by compiling and running the IDL program.

RUN PARAMETERS:

Polar satellite name
Beginning year/month
Ending year/month
Path to Input data
Path to Output data

INPUTS:

AMSU Level 1B files fro each {noaa} orbiter should be stored with
the following directory structure and file format:

$InPath/{noaa}/YYYYMM/{noaa}/NSS.AMAX.SS.Dyyddd.Snnnn.Ennnn.CC

OUTPUTS:

Output ASCII files are stored with the following directory
structure and file format:

$OutPath/{noaa}/YYYY/amsu_{noaa}_YYYY.DDD.dat

ALGORITHM:
---
*Loop over years
*Loop over months
*Loop over days in month
-Construct the set of Input filenames
-Construct the Output filenames
*Loop over the input files for this day
-Open input file
-Read in header data
*Loop over records
-Read in record
-Skip bad records
*Loop over 30 scans
*Loop over channels(1-15)
-Calcualte Brightness Temperatures
*
-Write values to output file
*
*
-Close input file
*
-Close output file
*
*
*

(1.c) extract_gpsro_profiles.pro

SYNOPSIS:

For the specified GPSRO mission, this program loops over
the specified range of Julian days and years, and reads in all
of the WET/ATM profiles for each day.
The daily profiles DRY temperatures from 'atmPrf' files and
profiles of water vapor from 'wetPrf' files are interpolated
to the 100 fixed pressure levels used by the AMSU forward
model. Missing data values are replaced using Standard
atmosphere values.
The resulting interpolated profiles are related values for
each Julian day are stored for later use.

USAGE:

For each GPSRO mission {'champ', 'cosmic', 'grace'},
the user needs to first edit the RUN parameters for the program to
set the mission name, the range of days to process and the
input/output paths for the data.
The data is then processed by compiling and running the IDL program.

RUN PARAMETERS:

Path to standard atmosphere (Ancillary) data
GPSRO mission name
begining year/Julian day
ending year/Julian day
Path to GPSRO input data
Path to output files

INPUTS:

GPSRO level2 profile files should be stored with the following
directory structure and file format:

$InPath/{gps}/level2/atmPrf/YYYY.DDD/atmPrf_MMMM.YYYY.DDD.*_nc
$InPath/{gps}/level2/wetPrf/YYYY.DDD/wetPrf_MMMM.YYYY.DDD.*_nc

** This program requires that matched 'atm' and 'wet' profiles
are used. To achieve this, the set of ATM files are read
from the $InPath directory. Then the 2 occurances of 'atm'
in each full file name are replaced with 'wet'. The resulting
WET profile must exist to be included. Because of this,
the user must be careful not to include 'atm' anywhere in the
$InPath directory name or the program will fail.

OUTPUTS:

Output ASCII files are stored with the following directory
structure and file format:

$OutPath/{gps}/YYYY/Prf_YYYY.DDD.dat'

ALGORITHM:

-Read in Standard Atmosphere data interpolated to
AMSU forward model pressures.
*Loop over the years
*Loop over juilan days
-Construct the set of ATM input files
*Loop over the number of input files
-Construct the corresponding WET input file name
-Read in data (Temperature) from ATM file
-Interpolate profiles to AMSU forward model pressures
-Read in data (Water Vapor) from WET file

-Interpolate profiles to AMSU forward model pressures
*
-Sort profiles by latitude
-Load Standard atmosphere profiles for 5 seasonal latitude bands
-Replace missing profile data using standard atmosphere values
-Open Output file and write profile data for the current day
*
*

(1.d) merra_hdf2sav.pro

SYNOPSIS:

This program loops over the specified range of Julian days
and years, and reads in MERRA HDF file for the day. The
data are shrinked and save to IDL data file for later use.

USAGE:

The RUN parameters specifying the data to process are set directly below this header.
The user needs to edit these values to process the desired values and then execute the
program from within IDL.

INPUTS:

MERRA HDF format data:
$InPath/*hdf

OUTPUTS:

Converted MERRA HDF data to IDL format data IDL format data:
$OutPath/YYYYDDD.sav

Algorithm

Set Julian days that will be dealwithed
Set variables
*Loop over the Julian days to process
Set time variables
Set output file name
Set input file name
Read in data
Select data
Save data
*

(1.e) merra_sav2txt.pro

SYNOPSIS:

This program loops over the specified range of Julian days

and years, and reads in MERRA IDL file for the day. The
data are interpolated and save to ASCII data file for later
use.

USAGE:

The RUN parameters specifying the data to process are set directly below this header.
The user needs to edit these values to process the desired values and then execute the
program from within IDL.

INPUTS:

MERRA IDL format data:
$InPath/YYYYDDD.sav

OUTPUTS:

Converted MERRA IDL data to ASCII format data IDL format data:
$OutPath/YYYYDDD.txt

Algorithm

Set Julian days that will be dealwithed
Read in some variables example.hdf
Loop over the Julian days to process
Set time variables
Set resolution of Latitude and Lontitude
Set pressure levels
Set input and output file name
*Loop over hours
*Loop over Latitude
*Loop over Longitude
Interpolate data
Outout data
*
*
*

==========
(2) Simulate AMSU/MSU brightness temperature:
==========

(2.a) amsu_raob.exe (./Src_fortran)

SYNOPSIS:

For the radiosonde data, this program loops over the
specified range of Julian days and years, and reads
in all of the radiosonde profiles for each day.
Each profile is passed to an AMSU forward model routine
which calculates the brightness temperatures for 15
AMSU channels. The resulting brightness temperatures for each
day are written to an ASCII file for later use.

USAGE:

The user needs to first edit the RUN parameters for the program contained in the file 'CONFIG.raob_gps' to set the mission name, the range of days to process and the input/output paths for the data.

The program is then compiled to generate an executable:

```
bash> make amsu_raob.exe
bash> vi CONFIG.amsu_raob
bash> ./amsu_raob.exe
```

RUN PARAMETERS:

Begining year/Julian day
Ending year/Julian day
Path to interpolated radiosonde profiles
Path to output brightness temperature files

INPUTS:

Input ASCII files containing the radiosonde profile data should be stored with the following directory structure and file format:

```
$InPath/YYYY/Prf_YYYY.DDD.dat'
```

OUTPUTS:

AMSU brightness temperatures are stored in the files:

```
$OutPath/YYYY/AMSU_raob_atmBt_YYYY.DDD.dat
```

ALGORITHM:

```
*Loop over years
*Loop over Julian days
 -Open input and output files
 *Loop over input profiles
  -Read in radiosonde profiles
  *Loop over 15 AMSU channels
   -Calculate brightness temperatures
   -Write results to output file
  *
 *
 -Close input/output files
*
*
```

(2.b) amsu_merra.exe (./Src_fortran)

SYNOPSIS:

For the MERRA data, this program loops over the specified range of Julian days and years, and reads in all of the MERRA profiles for each day. Each profile is passed to an AMSU forward model routine which calculates the brightness temperatures for 15 AMSU channels. The resulting brightness temperatures for each day are written to an ASCII file for later use.

USAGE:

The user needs to first edit the RUN parameters for the program contained in the file 'CONFIG.amsu_merra' to set the mission name, the range of days to process and the input/output paths for the data.

The program is then compiled to generate an executable:

```
bash> make amsu_merra.exe
bash> vi CONFIG.amsu_merra
bash> ./amsu_merra.exe
```

RUN PARAMETERS:

Beginning year/Julian day
Ending year/Julian day
Path to interpolated MERRA profiles
Path to output brightness temperature files

INPUTS:

Input ASCII files containing the MERRA profile data should be stored with the following directory structure and file format:

```
$InPath/Prf_YYYY.DDD.dat'
```

OUTPUTS:

AMSU brightness temperatures are stored in the files:

```
$OutPath/YYYY/AMSU_merra_atmBt_YYYY.DDD.dat
```

ALGORITHM:
---
*Loop over years
*Loop over Julian days
 -Open input and output files
 *Loop over input profiles
  -Read in GPSRO profiles
  *Loop over 15 AMSU channels
   -Calculate brightness temperatures
   -Write results to output file

```
 *
 *
 -Close input/output files
 *
 *
```

(2.c) msu_merra.exe (./Src_fortran)

SYNOPSIS:

For the MERRA data, this program loops over the specified range of Julian days and years, and reads in all of the MERRA profiles for each day. Each profile is passed to an MSU forward model routine which calculates the brightness temperatures for MSU channels. The resulting brightness temperatures for each day are written to an ASCII file for later use.

USAGE:

The user needs to first edit the RUN parameters for the program contained in the file 'CONFIG.msu_merra' to set the mission name, the range of days to process and the input/output paths for the data.

The program is then compiled to generate an executable:

```
bash> make msu_merra.exe
bash> vi CONFIG.msu_merra
bash> ./msu_merra.exe
```

RUN PARAMETERS:

Begining year/Julian day
Ending year/Julian day
Path to interpolated MERRA profiles
Path to output brightness temperature files

INPUTS:

Input ASCII files containing the MERRA profile data should be stored with the following directory structure and file format:

```
$InPath/Prf_YYYY.DDD.dat' OUTPUTS:
```

AMSU brightness temperatures are stored in the files:

```
$OutPath/MSU_MERRA_BT_YYYY.DDD.dat
```

ALGORITHM:

---

*Loop over years

*Loop over Julian days
-Open input and output files
*Loop over input profiles
-Read in GPSRO profiles
*Loop over MSU channels
-Calculate brightness temperatures
-Write results to output file
*
*
-Close input/output files
*
*

(2.d) bin_daily_raob_amsu.pro (./Src_IDL)

SYNOPSIS:

For the radiosonde simulated AMSU BT data, this program loops over the specified range of Julian days and years, and reads in AMSU BT each day. The BTs are gridded and are written to an NetCDF file for later use.

USAGE:

The RUN parameters specifying the data to process are set directly below this header. The user needs to edit these values to process the desired values and then execute the program from within IDL.

INPUTS:

RAOB simulated NOAA brighness temperature:
$InPath_rebuild/

OUTPUTS:

Gridded RAOB simulated NOAA brighness temperature:
$OutPath_noaa/

ALGORITHM:

---

*Loop over Julian days
- Set time variables
- Outfile
- read in data
- bin data
- write data
*

(2.e) merra_day_nc.pro (./Src_IDL)

SYNOPSIS:

For the radiosonde simulated AMSU BT data, this program
loops over the specified range of Julian days and years, and
reads in AMSU BT each day. The BTs are gridded and are
written to an NetCDF file for later use.

USAGE:

The RUN parameters specifying the data to process are set directly below this header.
The user needs to edit these values to process the desired values and then execute the
program from within IDL.

INPUTS:

MERRA MSU simalated ASCII data:
$InPath/

OUTPUTS:

IDL format data:
$OutPath/YYYYDDD.nc

Algorithm

Loop over the Julian days to process
Set input file name
Set output file name
Read in data
Bin data
Output data
*

==========
(3) Calibration of NOAA MSU using SNO method: (./Src_IDL):
==========

(3.a) match_noaa_amsumsu.pro

SYNOPSIS:

For the MSU or AMSU BT data, this program loops over
the specified range of Julian days and years, and reads in
AMSU or MSU BT, time and locations each day. The BTs
are matched are written to an NetCDF file for later use.

USAGE:

The RUN parameters specifying the data to process are set directly below this header.
The user needs to edit these values to process the desired values and then execute the
program from within IDL.

INPUTS:

NOAA AMSU data should be stored with the following directory structure and file format:

$InPath_noaa/YYYYMM/YYYYDDD_msu_{noaa}_nc

OUTPUTS:

Converted matched NOAA coefficients to IDL format data:
$OutPath/{noaa1}_{noaa2}/match_{noaa1}_{noaa2}_YYYYMM_nc

Algorithm
---

*Loop over the Julian days to process
Out file name and out variables
Set initial values
*Loop over the Julian days to process
Set input file name
Read in data
Select data
Matching two data sets
Output data
*
*

(3.b) sno_step1_preparematchup_msu.pro

SYNOPSIS:

For the MSU matched data, this program loops over the specified range of Julian days and years, and reads in MSU raw count written to an IDL data file for later use.

USAGE:

The RUN parameters specifying the data to process are set directly below this header. The user needs to edit these values to process the desired values and then execute the program from within IDL.

INPUTS:

Monthly matched NOAA MSU data:
$InPath_match/

Monthly NOAA coefficient data:
$InPath_coef/

OUTPUTS:

Converted matched NOAA coefficients to IDL format data:
$OutPath/step1_{noaa1}_{noaa2}.sav

Algorithm

Set Julian days that will be dealwithed
Set satellite names for the match pair
Set output file name
Loop over the Julian days to process
  Set parameters for satellites
  Set input file name
  Read the match data
  Read AMSU coef data
  find the SNO match pairs
  search the corresponding Rc, Rw, Cc, Cw, Ce
  equation of the planck function
  Radiance
*
Set saved data
Save data

(3.c) sno_step2_soluteequation_msu.pro

SYNOPSIS:

For the saved matched MSU raw counts data, this program loops over the names of NOAA missions to calculate the slopes and offsets for all the missions saved to ASCII files for later use.

USAGE:

The RUN parameters specifying the data to process are set directly below this header. The user needs to edit these values to process the desired values and then execute the program from within IDL.

INPUTS:

Matched NOAA coefficient data:
  $InPath/step1_{noaa1}_{noaa2}.sav

OUTPUTS:

Generate NOAA radiance offset and nonlinear coefficients
  $OutPath/*.dat

Algorithm

---

*Loop missions to process
  Get mission names
  Read or generate offset and nonlinear for NOAA

Read in SNO counts
Quality control
Generate the slope and offset
Save data
*

(3.d) sno_step3_rebuild_msu.pro

SYNOPSIS:

This program loops over the specified range of Julian days and years, and reads in MSU raw count and slopes and offsets to rebuild BTs written to an NetCDF data file for later use.

USAGE:

The RUN parameters specifying the data to process are set directly below this header. The user needs to edit these values to process the desired values and then execute the program from within IDL.

INPUTS:

Original NOAA coefficients:
$InPath_rad/

Offset and Nonlinear coefficients:
$InPath_coef/

OUTPUTS:

Rebuilt NOAA MSU Brightness Temperatures
$OutPath_noaa/

Algorithm

Set Julian days that will be deal withed
*Loop over the Julian days to process
Set output file name
Read offset and nonlinear coefficients
Read raw counts of NOAA
Get radiance and counts
Rebuild BT
Output data
*

(3.e) sno_step3_cnadir_dbin_msu.pro

SYNOPSIS:

For specified range of Julian days and years, the MSU rebuilt BTs are read in and calibrated by MERRA simulated BTs. Then the BTs are gridded and written to an NetCDF data file for later use.

USAGE:

The RUN parameters specifying the data to process are set directly below this header. The user needs to edit these values to process the desired values and then execute the program from within IDL.

INPUTS:

Rebuilt NOAA brightness temperature:
$InPath_rebuild/

MERRA simulated MSU brightness temperatures in differents satellite zenith angles
$Inpath_MSU_MERRA/

MERRA simulated AMSU brightness temperatures
$Inpath_AMSU_MERRA/

OUTPUTS:

Gridded calibrated rebuilt MSU brightness temperature
$Outpath/

Algorithm

Set Julian days that will be deal withed
Loop over the Julian days to process
Set Output file
Set input file and read in data
Set hour
Limb and local time correction
Write data to output file
*

(3.f) sno_step4_intersatbias_msu.pro

SYNOPSIS:

For specified range of Julian days, this program read in the
MSU gridded rebuilt BTs and generate the intersatellite
biases written to an IDL data file for later use.

USAGE:

The RUN parameters specifying the data to process are set
directly below this header. The user needs to edit these
values to process the desired values and then execute the
program from within IDL.

INPUTS:

Rebuilt NOAA brightness temperature:
$InPath_rebuild/

OUTPUTS:

Inter satellite bias
$OutPath_noaa/

Algorithm

Set names of missions
set output file name
Set Julian days that will be deal withed
Loop over the Julian days to process
Set input file names
Read in first warm target temperature
Read in brightness temperature
Calculate the bias
Save data
*

(3.g) sno_step4_satraobbias_msu.pro

SYNOPSIS:

For specified range of Julian days, this program read in the
MSU gridded rebuilt BTs, RAOB gridded BTs and generate
the biases between RAOB and NOAA14 written to an IDL
data file for later use.

INPUTS:

Rebuilt NOAA gridded brightness temperature:
$InPath_noaa/

Rebuilt GPS simulated MSU gridded brightness temperature:
$InPath_gps/

OUTPUTS:

Inter satellite biases of NOAA and GPS
$OutPath/

Algorithm

Set Julian days that will be deal withed
Set Bias initial values
Loop over the Julian days to process
Set input file names
Read in first warm target temperature
Read in brightness temperature
Calculate the bias
Save data
*

(3.h) sno_step5_chrisycorr_msu.pro

SYNOPSIS:

For specified names of MSU mission pairs, this program read in the MSU intersatellite biases and bias between NOAA14 and radiosonde and generate the Christy coefficients written to IDL data file and ASCII file for later use.

USAGE:

The RUN parameters specifying the data to process are set directly below this header. The user needs to edit these values to process the desired values and then execute the program from within IDL.

INPUTS:

Intersatellite bias:
$InPath_bias/

Calibration coefficients:
$Inpath_coef/

OUTPUTS:

Christy calibration coefficients:
$OutPath/

Algorithm

Set X,Y between NOAA and RO
Read in bias of GPS and NOAA14
Set brightness temperature and warm target temperature
Set time variables
Set inter satellite bias
Select good data to process
Set seasonal variation
Set X,Y for solve equation
Get X,Y between NOAAs
Read in intersatellite biases
Set time variables
Set brightness temperature and warm target temperature
Set inter satellite bias
Select good data to process
Set X,Y for solve equation
using the coefficients to remove the bias and generate the intersatellite bias

---
(3.i) sno_step6_chrisyprod_msu.pro

SYNOPSIS:

For specified range of Julian days and names of MSU
missions, this program read in the MSU gridded rebuilt BTs
and Christy coefficients to generate the calibrated MSU
BTs for later use.

USAGE:

The RUN parameters specifying the data to process are set
directly below this header. The user needs to edit these
values to process the desired values and then execute the
program from within IDL.

INPUTS:

Daily gridded data:
$InPath_grid/

Monthly NOAA coefficient data:
$InPath_coef/

OUTPUTS:

Converted NOAA AMSU values are written to the
ASCCI file:
$OutPath_noaa/bin_{noaa}_YYYYMM_chan9_converted.dat

Set string of guess number
Set Julian days that will be dealwithed
Set Latitude scopes
generate the correction coefficients
Read in calibration coefficients
Loop over the Julian days to process
    Set the infile
    Set the output file
    Read in data
    Calibrate data
    Write data to output file
*

(3.j) sno_step7_new_intersatbias_msu.pro

SYNOPSIS:

For specified range of Julian days, this program read in the MSU Calibrated rebuilt BTs and generate the intersatellite biases written to an IDL data file for later use.

USAGE:

The RUN parameters specifying the data to process are set directly below this header. The user needs to edit these values to process the desired values and then execute the program from within IDL.

INPUTS:

Gridded Brightness temperatures:
    $InPath/

OUTPUTS:

Inter satellite bias:
$OutPath/

Algorithm

Set string of guess number
Set names of missions
set output file name
Set Julian days that will be deal withed
Loop over the Julian days to process
Set input file names
Read in brightness temperature
Calculate the bias
Save data

(3.k) sno_step8_select_msu.pro

SYNOPSIS:

For specified names of MSU pairs, this program read in the
MSU intersatellite biases to calculate the RMS. The rebuilt
ID of guess which show smallest RMS rebuilt BTs is
written to an IDL data file for later use.

USAGE:

The RUN parameters specifying the data to process are set
directly below this header. The user needs to edit these
values to process the desired values and then execute the
program from within IDL.

INPUTS:

Intersatellite bias data:
$InPath/

OUTPUTS:

Id for best rebuild data:
$OutPath/

Set Julian days that will be deal withed
Set names of satellites
Read in inter satellite bias
Set inter bias
Set root-mean-squre of inter bias
Set mean of root-mean-squre
Save id of best rebuild data

==========
(4) Calibration of NOAA AMSU using coefficients from matched data : (./Src_IDL):
==========

(4.a) match_raob_noaa.pro

SYNOPSIS:

For the given range of Julian days, read in AMSU data
extracted from the specified NOAA dataset and simulated
AMSU brightness temperatures from profiles of the
RAOB.
Find collocated measurements satisfying the matchup
criteria and then save the matched values in an ASCCI
file for later use.

USAGE:

The RUN parameters specifying the data to process are set directly
below this header. The user needs to edit these values to process the
desired values and then execute the program from within IDL.

RUN PARAMETERS:

Name of NOAA polar orbiter to use
Beginning year/Julian day
Ending year/Julian day
Path to RAOB input data
Path to NOAA input data
Path to output file of matched values
Match Criteria: Time
--- Distance
Scan Angle

INPUTS:

NOAA AMSU data should be stored with the following directory
structure and file format:

$InPath_noaa/{noaa}/YYYY/amsu_{noaa}_YYYY.DDD.dat

RAOB simulated AMSU data should be stored in the format:

$InPath_raob/YYYY/AMSU_raob_atmBt_YYYY.DDD.dat

OUTPUTS:

Matched AMSU values are stored in ASCII files with the following
directory structure and file format:

$OutPath/{noaa}_raob/YYYY/match_{noaa}_raob_YYYY.DDD.dat

ALGORITHM:
---
*Loop over the years
*Loop over Julian days
-Open Output file
-Read in RAOB data
-Read in NOAA data
*Loop over RAOB values
-Determine indices of NOAA data that satisfy
the specified match criteria
-Write matched data to output file
*
-Close output file
*
*

(4.b) offset_slope_raob_noaa_month.pro

SYNOPSIS:

For each 1 month or 3 months running in the specified range, read in the matched AMSU temperatures between the specified polar orbiter and for all specified RAOB instrument types. Calculate the linear fit coefficients for each month. Save the resulting monthly values in an ASCII file for later use.

USAGE:

For each NOAA polar orbiter {'noaa15', 'noaa16', 'noaa18', 'noaa19', 'noaam2'} the user needs to edit the RUN parameters to set the orbiter name.
The range of days to process, the input/output paths for the data, and the criteria values for matched coincident values also need to be set. [the criteria enable the use to strengthen the requirements that were used in previous steps]
The data is then processed by compiling and running the IDL program.

RUN PARAMETERS:

Name of NOAA orbiter to use
Beginning year/month
Ending year/month
Path to matched NOAA/RAOB data sets (input)
Path to output file of linear fit coefs
Match Criteria: Time
---------- Distance
Scan Angle
Brightness Temperature

INPUTS:

Matched AMSU values are read from the ASCII files with the following directory structure and file format:

$InPath/{noaa}_raob/YYYY/match_{noaa}_raob_YYYY.DDD.dat

OUTPUTS:

Linear fit coefficients are written to the ASCII file:

$OutPath/timeseries_offset_slope_{noaa}_raob_1monthrunning.dat
$OutPath/timeseries_offset_slope_{noaa}_raob_3monthrunning.dat

ALGORITHM:

-Open Output file
*Loop over the years
*Loop over months
*Loop over days in the month
-Read in collocated RAOB/NOAA data
-Accumulate match data

*
-For channel 7 data
-Get matched data values which satisfy the specified criteria
-Calculate the linear fit coefficients
-Write linear fit coefs to output file
*
*
-Close output file

(4.c) offset_slope_raob_noaa_month_modify.pro

SYNOPSIS:

Because the radionsonde station are mostly located in tropical
and north hemisphere, modify and generate the slope-offset coefs
in two regions, 90N~20S and 20S~90S using the rules below:
In 90N~20S:
always use 1 month slope and offset

In 20S~90S,
if 1 month slope is between 0.95~1.05 then
use 1 month slope and offset
else if 3 month slope is between 0.95~1.05 then
use 3 month slope and offset
else if 3 month slope is lower than 0.95
slope = 0.95-(0.95-slope_3month)/10
offset = mean(bt_raob)-mean(bt_noaa*slope)
else if 3 month slope is larger than 1.05
slope = 1.05+(slope_3month-1.05)/10
offset = mean(bt_raob)-mean(bt_noaa*slope)
endif

USAGE:

The RUN parameters specifying the data to process are set
directly below this header. The user needs to edit these values
to process the desired values and then execute the program
from within IDL.

RUN PARAMETERS:

Set of names for NOAA orbiters to use
Path to file of linear fit coefs (input)
Path to output file of modified linear fit coefs

INPUTS:

Monthly linear fit coefs between NOAA and RAOB data are read from:

$InPath_coef/timeseries_offset_slope_{noaa}_raob_1month_running.dat
$InPath_coef/timeseries_offset_slope_{noaa}_raob_3month_running.dat

OUTPUTS:

Converted NOAA AMSU values are written to the ASCCI file:

$OutPath_coef/timeseries_offset_slope_{noaa}_raob_NH_TROP.dat
$OutPath_coef/timeseries_offset_slope_{noaa}_raob_SH.dat

ALGORITHM:

-Open Output file
-Read in linear fit coefficients for the specified orbiter
*Loop over months
-modify the slope and offset
*
-Write modified linear fit coefs to output file
*
-Close output file

(4.d) bin_noaa_daily.pro

SYNOPSIS:

For the specified polar orbiter and AMSU channel and range
of months, AMSU values are averaged on a 2.5 degree grid.
Monthly gridded values are written to ASCII files for later
use.

USAGE:

For the each NOAA polar orbiter {'noaa15', 'noaa16', 'noaa18',
'noaa19', 'noaam2'}, the user needs to edit the RUN parameters
to set the orbiter name, the AMSU channel to use, the range of
days to process, and the input/output paths for the data.
The data is then processed by compiling and running the IDL program.

RUN PARAMETERS:

Name of NOAA orbiter
AMSU channel number
Beginning year/month
Ending year/month
Path to input NOAA data
Path to output gridded data

INPUTS:

Daily values of AMSU are read from the files:

$InPath/{noaa}/YYYY/amsu_{noaa}_YYYY.DDD.dat

OUTPUTS:

Monthly gridded means are written to the files:
$OutPath/{noaa}/{noaa}_YYYYDDD_nc

ALGORITHM:
---
*Loop over years
*Loop over months
*Loop over days in month
-Read in input data for NOAA orbiter
-Accumulate values
*
-Open output file
-Bin data according to lat/lon grid points
-Calculate average values to each grid point
-Write gridded averages to output file
-Close output file
*
*

(4.e) convert_amsu_byraob.pro

SYNOPSIS:

For the specified AMSU dataset, Read in RAOB correction coefficients from the specified file, read in gridded AMSU data, apply the correction, and write the converted data to the given output file.

USAGE:

For each polar orbiter {'noaa15', 'noaa16', 'noaa18', 'noaa19', 'noaam2'}, the user needs to edit the RUN parameters to set orbiter name, the path to the offset_slope calibration coefficients, and the input/output paths for the data.
The data is then processed by compiling and running the IDL program.

RUN PARAMETERS:

Name of NOAA polar orbiter
Path to linear fit coefficient files
Path to input gridded AMSU data
Path to output converted gridded AMSU data

INPUTS:

Monthly linear fit coefs between NOAA and RAOB data are read from:

$InPath_coef/timeseries_offset_slope_{noaa}_raob_NH_TROP.dat
$InPath_coef/timeseries_offset_slope_{noaa}_raob_SH.dat

Gridded NOAA AMSU values are read from:

$InPath_noaa/bin_{noaa}_YYYYDDD_nc

OUTPUTS:

Converted NOAA AMSU values are written to the
ASCCI file:
$OutPath_noaa/bin_{noaa}_YYYYDDD_nc

ALGORITHM:

-Read in linear fit coefficients for specified orbiter
*Loop over the months in the coefficient file
-Read in gridded orbiter data for the month
-Apply linear correction to each gridpoint
-Save converted gridded AMSU data to output file
*

==========
(5) Generate product: (./Src_IDL)
==========

(5.a) msu_daily_product.pro

SYNOPSIS:

For specified range of Julian days, this program read in the
ID of smallest RMS of intersatellite biases to select best
MSU BTs written to an NetCDF data file for later use.

USAGE:

The RUN parameters specifying the data to process are set
directly below this header. The user needs to edit these
values to process the desired values and then execute the
program from within IDL.

INPUTS:

Daily Christy calibrated data:
$InPath_prod/

Selected best guess number:
$InPath_coef/

OUTPUTS:

Best results:
$OutPath/YYYYMM/YYYDDD_{noaa}_nc

Read in best guess number
Set Julian days that will be dealwithed
Loop over the Julian days to process
Set time variables
Set the infile

Set the output file
Copy data file
*

(5.b) monthly_product.pro

SYNOPSIS:

For specified range of Julian days, this program read in the MSU daily calibrated BTs and generate the monthly gridded BTs written to an IDL data file for later use.

USAGE:

The RUN parameters specifying the data to process are set directly below this header. The user needs to edit these values to process the desired values and then execute the program from within IDL.

INPUTS:

Daily NOAA MSU BT data:
$InPath_msu/
Daily NOAA AMSU BT data:
$InPath_amsu/

OUTPUTS:

Converted NOAA AMSU values are written to the ASCCI file:
$OutPath/YYYYMM.sav

Generate the monthly Julian days
Set names of all satellites
Loop months to do the calculation
Set time variables
Set initial values of number and brightness temperatures
Loop noaa satellites
Set path of input data
Loop Julian day
Set input file name
Read in data
Accumulate the BT and number
*
*
Calculate the monthly BT
Open output file
*

(5.c) gen_product.pro

SYNOPSIS:

For specified range of Julian days, this program read in the
MSU monthly gridded calibrated BTs and generate
climatology and anomalies written to NetCDF data file as
Output.

USAGE:

The RUN parameters specifying the data to process are set
directly below this header. The user needs to edit these
values to process the desired values and then execute the
program from within IDL.

INPUTS:

Monthly brightness temperatures are read from file with
the following directory structure and file format:
$InPath/YYYY_MM_nc

OUTPUTS:
Monthly brightness temperatures are written to netCDF files:

$OutPath/RO-CAL-BT-CDR_V01R00_MON_s*.nc

Brightness temperatures climatology are written to netCDF files:

$OutPath/RO-CAL-BT-CDR_V01R00_CLIM_s*.nc

Brightness temperatures anomaly are written to netCDF files:

$OutPath/RO-CAL-BT-CDR_V01R00_ANOM_s*.nc

Algorithm

Definite Longitude and Latitude values
Generate the Julian days for the months to deal with
Initial BT data
Loop over months to process
    Generate the month, and year from Julian day
    Read in BTs in each month
    Copy values for later use
*
Generate BT Climatology and Anomaly
Output Monthly Brightness temperatures
Time variables
File name and dimensions
Variables definition
Global Attributes
Write values to the output file

Output Brightness Temperature Climatology
Time values

File name and dimensions
Variables definition
Global Attributes
Write values to the output file

Output Brightness Temperature Anomaly
Time values
File name and dimensions
Variables definition
Global Attributes
Write values to the output file