Cadilinux:

User's Manual

Scientific Instrumentation Ltd.
Telephone: (306) 244-0881
Facsimile: (306) 665-6263

Cadilinux:
User's Manual

SIL DOCUMENT:

REV.	DATE	PREPARED BY	APPROVAL	DCN
1.0	20 Dec. 2000	John Mac Dougall		
1.1	Mar 2001	Garth Steel		
1.2	20 June 2001	G.S. Broten		
1.3	19 Dec. 2001	G. S. Broten		
1.4	20 March 2002	G. S. Broten		

File : CadiLinuxMan2
Format : MS-WORD 7.0

Table of Contents

1.0 Cadilinux Software

1.1 Overview

The Linux version of Cadi uses two separate programs to capture ionosondes. These two programs are: **cadi** and **cadidisplay.** The cadi program sequences and captures the ionosonde data, while the cadiddisplay program displays a graphical view of an ionogram.

1.2 Operation

The Cadilinux software consists of two separate programs. The cadi program acquires the ionosonde data, while the cadidisplay program displays ionograms. Both programs can be run independently, meaning the cadi program will gather and store ionosonde data without needing the cadidisplay program. Similarly, the cadidisplay program can display ionograms, using saved data, without have the cadi program running. In general, the Cadilinux system has been designed to allow the cadi executable to acquire ionosonde data without the graphical cadidisplay program running. This allows the cadi program to running automatically, via the crontab, once the cadilinux computer is running.

1.3 Cadi Program

/USR/SBIN

The cadi program sequences and gathers ionosonde data. In general, its execution is controlled by the crontab, but it can be run directly if so desired or required. The cadi program requires that the mode definition file, to be used by the scan, be passed as a command line argument.
The cadi program can also be passed optional command line arguments to control its operation. These optional arguments include:

1. -v - version;
2. -h - help;
3. -? - help;
4. -k - Check if code key valid;
5. -c - path to configuration files, default is /etc/cadi;
6. -o - path to output directory, default is /var/cadidata;
7. -d - Debugging outputs;

The software has been tested under RedHat 7.0 and may or may not work on other distributions.

1.3.1 Configuration Files

When cadi executes, instructions are read from various files that tell the program what data to collect and how to collect it. The required configuration files are:

1. /etc/cadi/location.conf
2. /etc/cadi/igram_xx.def

See Section 2 of this document or the cadi man pages for more details with respect to these configurations files.

1.4 Cadidisplay

The cadidisplay program displays graphical images of ionograms. There are options for: viewing a single file, all the files in a directory, or viewing plots as they are acquired by the cadid program. The cadidisplay program supports the following options:

1. -f - View a single file. In this mode you have the ability to select points to be written to a text file. If you left click on the plot the frequency and height will be displayed on the right side of the screen. To save points to a file hit p to begin recording, then click on the points you want. When done hit s and the points will be saved as points.txt to the data directory specified in the .cadidisplayrc file. If you hit p again the current list of points will be cleared. In this mode you can also hit d which creates a jpeg out of the current plot and puts it in the jpeg directory as specified in the .cadidisplayrc file.
2. -d - View all the files in a directory. In this mode it displays all the files in the directory with a small pause between files.
3. -p - View files as they are generated by the data collection software. Give the directory that the collection software uses for new files and it will automatically view files as they are generated. This mode automatically moves the file to the directory listed in the .cadidisplayrc file.
4. -j - Make a jpeg file of the plot and put it in the directory specified in the .cadidisplayrc file. Note this is only valid in polling and directory mode.

You need one, and only one of the modes (f, d, p) and the –j option can be used with the d and p modes.

Example usage: cadidisplay –p /var/cadidata –j

1.4.1 Configuration Files

The cadidisplayrc file defined the path to both the data files directory and to the jpeg files directory. This file must be located under the /etc/cadi directory.

1.5 Dataconvert Program

The dataconvert program converts cadi data files from DOS to Linux and Linux to DOS. This program is required since the DOS Cadi program file format is not identical to the Linux file format. This program supports the following options:

1. -d, -f Convert a whole directory. If doing a whole directory it will do all files with the extension of mdd(dos) and mdl(linux).
2. -f Convert just a single file.
3. -o, Convert to DOS,
4. -l Convert to Linux. It will automatically use the extensions mdd for DOS and mdl for linux.

2.0 Cadi Configuration Files

The main data collection program is the cadi executable. It uses two files when operating. These files are **location.conf** and the specified mode definition file. The location.conf file is detailed in Section 2.1, while the mode definition files are explained in Section 2.2. All of these files are simple text files and can be edited using VI or any other text editor that the user is familiar with.

2.1 location.conf

The location configuration file defines the basic configuration of the CADI system. This file is a simple text file and may be edited using a text editor such as VI. It is a free format file that is order. space and case insensitive. The parameter name **MUST** appear to the left of the equals sign and the defined value must be to the right of the equals sign. CADI is shipped with a default location.conf that should be modified to suit the user's CADI configuration. An example location.conf file is shown below, with the meaning of each line detailed.

1. station name = sil
    - 3 Characters - Example: SIL
    - The station name can be any three characters that identify the location of the CADI site.
2. number of receivers = 1
    - Integer, Number of receivers in the system, 0 -> 4.
3. DDS type = U
    - Single character: N = SIL DDS, U = SIL DDS with MODES
    - With the U DDS type, several extra options are available. All newer S.I.L. CADI systems support the U DDS type. These extra options are:
        - The maximum height (heightrange) can be either 510 or 1020. (If you select 1020 then the pulserate is changed to 20 pps instead of 40 pps.)
        - The delayed code can be turned ON or OFF
        - The system can select the LISTEN mode, under which there is no transmitter output
        - A 7 bit Barker code or a 13 bit Barker code can be specified. This allows lower minimum height for useful data.
4. Max Ht = 510
    - Integer value
    - The maximum height of the CADI system is either 510 or 1020 km.
5. License key = 3049058854J120399635
    - String: Example - 3049058854J120399635
    - *In order to run cadi you must have a valid key code. To get a valid key you have to email SIL at* s.i.l@sil.sk.ca *and give the unlock code. You can find out your unlock code by trying to run the program without a valid code, or running the program with the –k option.*

2.2 Mode Definition File

The mode definition file defines the operating mode and frequency generation of the CADI system. This file is a free format text file that is space, case and order insensitive (with the exception of the frequency definitions). This file is passed to the cadi executable as a command line argument. The file format consists of a parameter definition that is followed by the value the parameter is assigned. The parameter name **MUST** appear to the left of the equals sign and the defined value must be to the right of the equals sign.

Two frequency generators are available, makefreq.pl and appendfreq.pl. Both take the start, end, number of frequencies, and a choice for linear or logarithmic as options. Makefreq.pl outputs a file with the frequency list, appendfreq.pl takes an extra option of the file to modify, and will append the list to a file, which is useful if you have a header definition file that you just want to add the frequency list to.

The following modes are recognized and are configurable:

1. file extension = mdl
   - 3 characters, Example: mdl
   - This is the file extension which each of the data files written by this mode will use. The file extension is the only way of distinguishing files containing data from a particular mode since the leading characters of the filename are used for date and time. The extension is limited to 3 characters and the mdl extension is commonly used for linux systems.
2. file structure = I
   - I = Individual run in file
   - H = All runs for hour in a single file
3. pulse style = B
   - Single or Barker: S or B
   - This line sets the operating mode for the system. The letter selects the transmitter mode. S sets it in single pulse mode which transmits a single pulse at each frequency in the frequency table. B sets the system to use the Barker mode. In this mode the transmitted signal is Barker code modulated. A new choice (micro code versions 5 +) is C for Complementary. This is an 8 bit code that uses a pair of different codes. When using Complementary you must do averaging of at least 2 pulses. The advantage of Complementary is that there are no coding sidelobes, and the minimum height range can be set as low as 60 km. Note: The current CADI shipped by S.I.L. does not yet support the Complementary code option.
4. pulses average = 1
   - No of samples to average: 1, 2, 4, 8, 16, 32, 64, 128, 256
   - This number represents the number of samples to average before displaying the result, and storing the data. This number can be either 1 or 2n up to a maximum of 256.
5. analysis type = F
   - FFT or None: F or N

- This option enables or disables the Fourier transform processing from being performed on the data. The F definition processes the data using a Fast Fourier Transform while the N definition uses no processing.

6. FFT sample size = 8
   - FFT samplesize: 1, 2, 4, 8, 16, 32, 64, 128, 256
   - This sets the number of samples used in the Fourier transform. Must be 2n up to a maximum of 512.
7. Cadi Config = Standard
   - Run Mode: Standard, Listen, Oblique, Obliquelisten
   - The Run Mode option define the how the CADI system runs. The following Run Modes are defined:
     - Standard - Transmitter and Receiver in single system.
     - Listen - Do not transmit, listen only with the receiver.
     - Oblique – The Transmitter is at a remote location from the receiver. This configuration requires a GPS for time (not yet implemented under linux).
     - Obliquelisten – The Receiver is at a remote location. This requires a GPS for time (not implemented under linux).
8. base threshold = 4.0
   - Floating Point value: Default 4.0
   - Base threshold is the minimum power that received echoes must have before they are saved. This would normally be set at about the level of background noise for your site. A value of 4 is fairly typical, and this threshold value is not usually very critical.
9. noise threshold = 1.1
   - Floating_Point: Default 1.3
   - In order to not save random interference (that is above the base threshold in power level) the average power received over all heights is calculated, multiplied by this noise threshold number, and only signals having higher power levels are saved. Whichever threshold is higher (base or this noise calculated one) is used when deciding if received signals will be saved. Note that because of the FFT signal processing that is normally used, this noise threshold can often be set < 1.0 and still not save too much 'noise' because the noise is suppressed by the signal processing. Getting the best setting of this parameter is somewhat a matter of trial-and-error. You can try various settings of these two thresholds using the CADITEST (DOS based application) program.
10. doppler = 1
    - Integer: Default 1
    - For some applications you can reduce the amount of data saved by specifying that several Doppler components must be above threshold
11. min ht = 90
    - Integer: Default 90
    - This value sets the minimum height for data that will be displayed and recorded in the data files.If you set this <90 km, and modes are available, then the length of the Barker sequence is changed from 13 bits to 7 bits.

12. max ht = 510
    - ~~Integer: Default 510~~
    - This value sets the maximum height for data that will be displayed and recorded in the data files. This is normally 510 km. If you set this > 510 but < 1020, and modes are available, then the overall height sampling is changed to 1020 and the pulserate is reduced from 40 to 20 pps.
13. gain control = N
    - N = none,
    - 0..7 = set gain,
    - A = automatic
    - You can set the input attenuator value in 5 dB steps between 0 and 7*5 = 35 dB. If you specify a value (N) it is set to 2*5 = 10 dB. Automatic is not implemented at this time.
14. delayed pulse = no
    - yes or no
    - Simulated a pulse from the E region to verify the receiver operation
15. frequencies = 95
    - 95 - This is the number of frequencies in the table.
16. The following list contains all the individual frequencies. They MUST be defined after the number of frequencies.
    1) 0 = 1.5e+06
    2) 1 = 1.56e+06
    3) 2 = 1.58326e+06
    4) 3 = 1.62661e+06
    5) 4 = 1.67114e+06
    6) 5 = 1.71689e+06
    7) 6 = 1.7639e+06
    8) 7 = 1.81219e+06
    9) 8 = 1.86181e+06
    10) 9 = 1.91278e+06
    11) 10 = 1.96515e+06... would continue to 95 frequencies ...

- If you are creating a new mode with many frequencies (e.g. an ionogram mode) it would be tedious to enter all the frequencies manually. To simplify the process the web interface has an option for making the frequencies and integrating them into the Mode Definition file automatically. There is also a script called makefreq.pl which will create and append a frequency list to a definition file.

3.0 Other software

3.1 CADITEST.EXE

When setting up a CADI system, the DOS based application, CADITEST.EXE can be used to test the system. This program allows the user to test all the different modes of operation. To use the program power up the fully connected CADI transmitter and then run CADITEST.EXE and use the up and down arrow keys to select an option from the menu. The file LOCATION.INI is called by this program. The used of CADITEST.EXE requires that the Cadi computer be capable of dual booting into both Linux and DOS.

4.0 CADI Data File Format

Format description below is in a quasi C language form. Italicised words are 'keywords' used in C.

Note: in C language a byte is called an unsigned char.

(A) file starts with fileheader which consists of:

```
struct fileheader
      {
       char timestamp[25];
       struct datablock dataBlock;
      };
```

timestamp is a 25 character ascii string consisting of:-

stationname (3 char) + time when file was opened (space + 21 char)

datablock is as follows:-

```
struct datablock
      {
      char filetype;      //  filetypes are: 'I'= individual files, 'H' = one file per hour
      int no of frequ;
      unsigned char doppler series length;
      int min height;
      int max height;
      unsigned char pulses per sec;
      unsigned char noof pulses averaged;
      int basethreshold100;  // basethreshold100 = 100× min power (bits sq.) before save
      int noisethreshold100;  // noisethreshold100 = 100×multiplier for average noisepower
                               (bits sq.) to set threshold based on noise
      unsigned char minimum number of dopplers before saving;
      int seconds between samples;
      char gaincontrol;  //  gaincontrol: 'N'= none, or a number
      char sigprocess; // sigprocess: 'F'=do FFT (then save all components above threshold),
               'T'= timeseq (FFT, saves entire spectrum if any component is above threshold)
               'N'= none (save raw data)
      unsigned char number of receivers;
      unsigned char spares[11];
      };
```

note on data storage in C language: *char*, *unsigned char* are 1 Byte; *int* are 4 Bytes

(B) next the list of all the frequencies used is written to the file as floating point numbers (4 Bytes each)

(C) next comes the data structured as follows:-

start of next_record:
    time_min (Byte)
    time_sec (Byte)

start of next_frequency:
    gainflag (Byte: E0 (hex) to EF: EF = none)
    noiseflag (Byte, F0 (hex) = basethreshold, F1 = noisethreshold)
    averagenoisepower10 (*unsigned int*: 10×averagenoisepower)

start of next height:
    hnum (Byte) virtual height is 3*hnum, for extended height range CADIs: if hnum > 200 then hnum = hnum -200 and tnum = tnum + 128 (use high bit of tnum as a flag)
    tnum (Byte): number of Doppler bins saved
    for each of the tnum Doppler bins write:
        {
    data ((1+2xnumber of receivers) Bytes: Doppler bin # then 2 Bytes real, imaginary components for each receiver)
        }

if: more heights goto start of next height,

if: more frequencies goto start of next frequency, else (end of last frequency) so write

    endfrequflag (Byte = FF (hex))

if: more records in the file then goto start next record
else end of file

5.0 Files and Directories Required by Cadilinux

Cadilinux requires the following files and directories:

1. /var/cadidata
   - This is the default directory for cadi data files. This directory is created by the installation process and should exist.
2. In /etc/cadi:
   - location.conf
   - Mode Definition files (.def)
   - cadidisplayrc Note that this is used only for the cadidisplay program and that the cadi program will run without the display program.
   - Default examples of all of these files are created during the installation process.

6.0 Operator Instructions

These instructions are for the CADI hardware arrangement that I have at my field sites. If you have a different arrangement you will need to edit this page.

1. Does the CADI appear to be running Ok?
   - Do a listing of the /etc/cadidata directory to see if ionosonde data files are being created.
   - Inspect /var/log/messages for messages from the cadi program
2. Is there info in the /var/log/messages log that indicates a weak receiver? If so the problem is usually in the antenna/preamp/cabling. Check these (see below: inspect antenna and cables for damage). If this does not reveal the problem then see the section "Receiver check and calibration" in this manual.
3. Are all 4 LED indicators on the preamp power supply box lit? (one for each antenna preamp, 4 at my sites) The preamp power supply has constant current output that passes through these LEDs. Therefore if an LED is not lit it tells you that there is no current going out the line that goes to that preamp. Therefore a LED that is not lit tells you that either the cable to the preamp is open circuited, or the preamp itself is open circuited. There should be a spare preamp at your site so you can usually figure out what is wrong by taking it out and plugging it into the cable where there may be a preamp or cable problem. You can plug it in with the preamp power ON since the constant current power supply isn't bothered if you make a temporary short while you are doing the connection.
4. Are all receivers producing approximately equal output? This can be tested by booting into DOS (assuming a dual boot configuration) and running CADITEST. Select "(2) fixed freq, Barker code, I_Q". You will get a screen display of the I (inphase) and Q (quadrature) signals (these are showing you the "real" and "imaginary" components of the received signals). The display is colour coded for the 4 receivers. You need to look closely to see the 4 different coloured traces (see colour "key"). If there is an ionospheric echo it will show up as a small positive or negative "bump" in the 100 - 300 (usually) kilometer height range. It will usually show on both the I and the Q displays. If you see an echo look closely to make sure that it has approximately the same <u>average</u> amplitude from all 4 receivers. If you don't see an echo then you may need to select a different frequency. Try changing the frequency to 3 MHz using option (0) on the CADITEST menu. You can also try running option (11) that displays an ionogram. From this display you can decide what would be a good frequency to look at the echo strengths. Note that sometimes you won't be able to get an echo on any frequency (because the ionospheric electron density is too low, or there is too much absorption). If one, or more, receivers is weak you should first check to see if the LED on the outside preamp is ON. Preamps get damaged by lightning, or by water leaking through the seal. Note also that the LED on the inside preamp power supply box could be ON, and the LED on the outside preamp OFF if the cable has a short. If the cable seems to be Ok and the outside preamp LED is ON, and there is no apparent antenna damage you need to:
   - Try the spare preamp to see if that makes any difference.

- Try the "Receiver check and calibration" procedure given in this manual. Our experience is, however, that the problem is almost always in the antenna/preamp/cabling rather than in the CADI receivers.

7.0 Receiver check and calibration

To check and calibrate the receivers a calibrated signal generator is required. The following steps are followed:

1. Set the generator to a frequency of 4.0 MHz (the default frequency for CADITEST). Leave the generator output OFF or at a very low setting.
2. Connect the generator to whichever receiver you are testing.
3. Leave the transmitter power amplifier turned OFF. Turn on the CADI pc and run CADITEST. Go to "toggle IQ means" and select this <cr>. The only change when you do this is that the line "IQ means:" at the top of the screen changes from OFF to ON. Go to "(1) fixed frequency, single pulse, I_Q" and select this. You will get a graphical display and the outputs from the receivers will show as coloured lines on the I (inphase) and Q (quadrature) plots. At the bottom of the screen the I and Q means for each receiver are shown. These should all be 128. If these are close to 128 (127 or 129) then you do not need to adjust the "zero offset" controls. If you need to adjust the zero offsets see below. You probably want to do the next check before doing the offset adjustments.
4. Now set the generator output to a level of -54dBm (446 uV). On the receiver for which the generator is input you should see large sinewaves on the I and Q displays. The sinewave frequency is proportional to the frequency difference between the 4.0 MHz generator frequency and the CADI 4.0 MHz frequency. You can vary the frequency of the sinewaves on the display by slightly varying the generator frequency. The amplitude of the displayed sinewaves should be full-screen. Estimate the amplitudes of the I and Q waves for this receiver. If they aren't closely the same then you should plan on doing a calibration.
5. Switch the generator input to the other receivers and estimate the amplitudes of the displayed sinewaves. Ideally all receivers should have the same wave amplitudes. Note that some CADI receivers don't have quite enough gain to produce full screen sinewaves with an input of -54 dBm. (Due to gain differences in receiver amplifiers). If this happened then the normal procedure is to set all the receivers to have the same amplitude of displayed sinewave. (But the amplitude should be close to full screen.)
6. Adjustments: If the above tests indicate that you need to adjust either the "zero offsets", or the gains, you will need to be able to access the potentiometers on the receiver boards. This can usually be done by only having one receiver board plugged in at a time. If you are using just 1 receiver, then set the dip switches on the receiver board to have this receiver be "receiver #1". The receiver number (address) is set by the package of 4 dip switches labelled "S1". For receiver #1 these are all OFF. Remember what the previous setting was because you probably want to reset this to its previous receiver number after you finish calibration. You can also opt to have a display of only the 1 receiver output by editing the number of receivers in the LOCATION.INI file. This is not necessary but makes the display simpler. If using only 1 receiver the local oscillator signal from the DDS board will be input to one of the miniature connectors on the receiver board, and the 50 ohm termination must be put on the other miniature connector. The

calibration uses potentiometers R58, R59, R60, and R61 (don't adjust R57 which sets the reference level for the A-to-D conversion):-

- Potentiometer R58 sets the "zero offset" for the "I channel".
- Potentiometer R59 sets the gain for the "I channel"
- Potentiometer R60 sets the "zero offset" for the "Q channel"
- Potentiometer R61 sets the gain for the "Q channel"

8.0 GPS Time Routines

The GPS Time option has not yet been implemented by Scientific Instrumentation Ltd. Please contact s.i.l@sil.sk.ca for more information.

8.1 Oblique Incidence Sounding

To do oblique incidence sounding you need two CADIs with GPS clocks. Actually you don't need two complete CADIs: All you need at one site (the transmitter site) is a DDS board and Power Amplifier, and at the second site (the receiver site) you need a DDS board and receiver boards. The GPS time information is input to the CADI's via a "clock interface" (contact S.I.L. about this).

For oblique sounding you would select the option "Oblique" (or "Obliquelisten") in the "Cadi Config = Oblique" line of the mode definition file.

You need to configure the GPS clocks for optimum timing. See section "GPS time routines" in this manual.

The oblique sounding mode samples are triggered by the 1 sec pulse provided by the GPS clock. A typical 1 sec 'sample' for an ionogram is 8 pulses which are then FFT processed. Such a sample would be acquired on each frequency so an ionogram of 200 frequencies would take 200 seconds. This is obviously quite slow compared to an "untriggered" ionogram, so there is plenty of time for an ionogram plot while waiting for the next frequency to be transmitted.

Note that the first frequency is transmitted 1 second after "t = 0". This 1 second delay gives slow hardware time to respond. To facilitate this delay it was necessary to impose the condition that the repeat interval for an oblique mode must be a multiple of 5 seconds.

CadiLinux Additional Notes

Root password: cadiroot

An extra user was added, just for storing data. Name: cadi Password: cadiuser

Occasionally a run will fail, this seems to be an issue with the motherboard of this particular computer. A workaround is in place. A program is run every minute from the crontab which scans the system log, if a failure has happened it will do the run again. The program is /root/bin/cadicheck. The igram file to re-run is specified as an argument.

The system as shipped has an example of the use of cadicheck and cadirun in the crontab.

/ETC/RC.D
FOR STARTUP