Documentation of Logger Tools Software

Olaf Kolle[1]
Max–Planck–Institute
for Biogeochemistry

May 2012

---

[1] olaf.kolle@bgc-jena.mpg.de

Contents

1 Overview

The Logger Tools Software consists of the following programs

- the launch program LOGTOOLS which calls specific programs for conversion of logger data, for editing control files and for remote access of dataloggers as well as some programs which are part of the Campbell Scientific LoggerNet software
- the program CR5000EC for conversion of raw data files from the Campbell datalogger CR5000 when used to collect eddy covariance data
- the program CSLOGGEN for conversion of data files from old and new Campbell dataloggers (Edlog and CRBasic) with the batch file version CSLOGBAT
- the program CSLOGCO2 for conversion of data files from the Campbell dataloggers CR10X and CR23X used in conjunction with the switching unit to measure profiles of CO_2 (LI6251 or GMP343) or CO_2 and H_2O (LI6262)
- the program SoilResp for conversion of data files from old and new Campbell dataloggers (Edlog and CRBasic) used in conjunction with the automated soil respiration system
- the program ConvertDLD to create lists of input location names from Edlog program files (extension dld)
- the program DLLOGGEN for conversion of data files from the DeltaT datalogger DL3000
- the program CSMODEM for remote access of old and new Campbell dataloggers (Edlog and CRBasic) via modem
- the program Edit_Ctl to create or change control files which are used by the data conversion programs

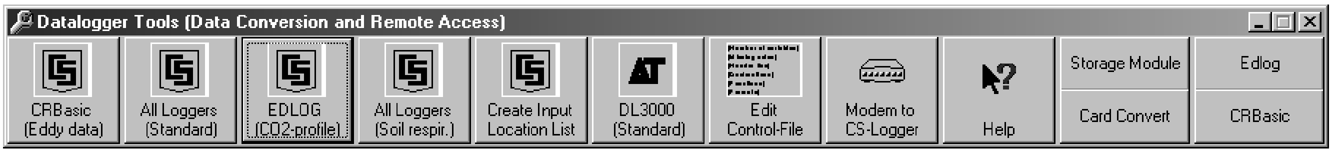


Figure 1: The LOGTOOLS launch bar

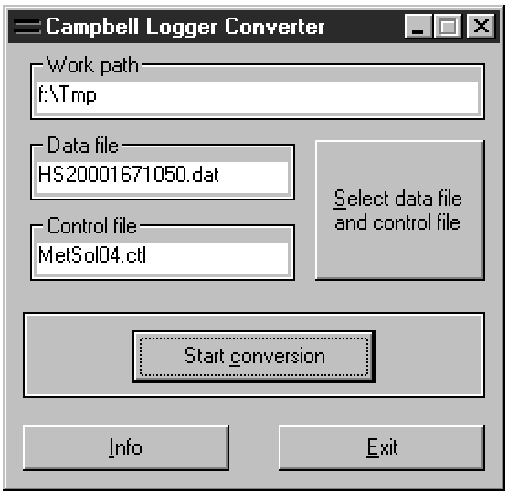
The data conversion programs read the original data file and perform the conversion operations specified in the control files. Normally only those sensor data are converted to real units within the logger program which do not have individual calibration coefficients such as PT100 temperature probes. From sensors with individual calibration coefficients as for example radiation instruments the logger is programmed to store the raw data. The conversion programs are used to convert the raw data into real units, to calculate additional variables and may also be used to check whether data lie in between certain limits.

2 The standard converters CSLOGGEN and DLLOGGEN

The dialog boxes of both programs look the same. For the Campbell dataloggers the data files must be stored as ASCII, comma separated where the PC208W or PC400 or LoggerNet program generates files with the extension dat. For the DL3000 datalogger the AQUIRE program is used to export the data as ASCII files. Normally AQUIRE uses the extension csv but the files should be exported as files with the extension dat. If the files are exported with the extension csv then the conversion program renames that files using the extension dat because the extension csv is used for the output files.

The old Campbell dataloggers **must** be programmed (by Edlog) to output firstly the year then the day of year then the time (hhmm) and then optionally the seconds of each data record. The conversion program works only with one of these two output formats! In the control file it must be indicated if the seconds are missing in the timestamp (see section 2.3.1). The time information for the new Campbell datalogger data (programmed in CRBasic) is generated automatically and cannot be changed. The time information for the DL3000 data is also generated automatically but the timestamp format in the export dialog must be set to dd/mm/yy!

2.1 CSLOGGEN for various Campbell dataloggers



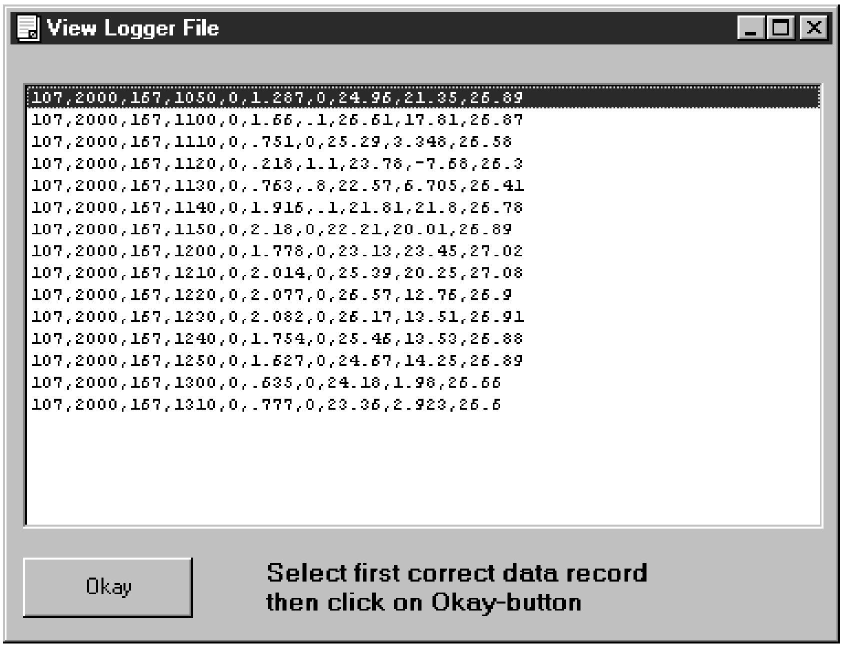
Work path: f:\Tmp

Data file: HS20001671050.dat

Control file: MetSol04.ctl

Figure 2: The CSLOGGEN dialog window

After selecting the data file and the control file a preview window opens (figure 3) where the first few lines of the data file are shown. The first correct data record which is normally the first line must be selected whereupon the window is left using the Okay button. The conversion process is then started using the Start conversion button. The progress of the conversion is shown by a blue bar.

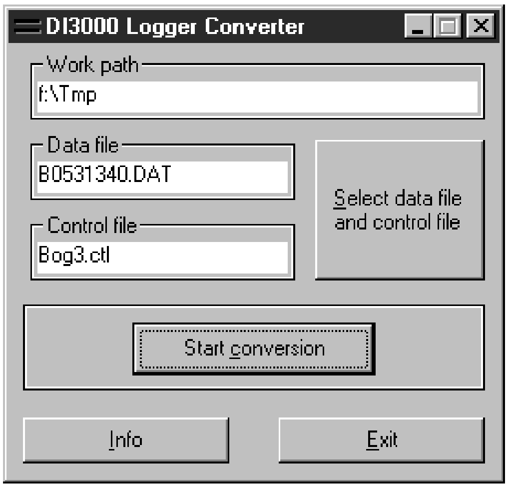


```
107,2000,167,1050,0,1.287,0,24.96,21.35,26.89
107,2000,167,1100,0,1.66,.1,26.61,17.81,26.87
107,2000,167,1110,0,.751,0,25.29,3.348,26.58
107,2000,167,1120,0,.218,1.1,23.78,-7.68,26.3
107,2000,167,1130,0,.763,.8,22.57,6.705,26.41
107,2000,167,1140,0,1.916,.1,21.81,21.8,26.78
107,2000,167,1150,0,2.18,0,22.21,20.01,26.89
107,2000,167,1200,0,1.778,0,23.13,23.45,27.02
107,2000,167,1210,0,2.014,0,25.39,20.25,27.08
107,2000,167,1220,0,2.077,0,26.57,12.76,26.9
107,2000,167,1230,0,2.082,0,26.17,13.51,26.91
107,2000,167,1240,0,1.754,0,25.46,13.53,26.88
107,2000,167,1250,0,1.627,0,24.67,14.25,26.89
107,2000,167,1300,0,.635,0,24.18,1.98,26.66
107,2000,167,1310,0,.777,0,23.36,2.923,26.6
```

Select first correct data record
then click on Okay-button

Figure 3: The CSLOGGEN file preview window

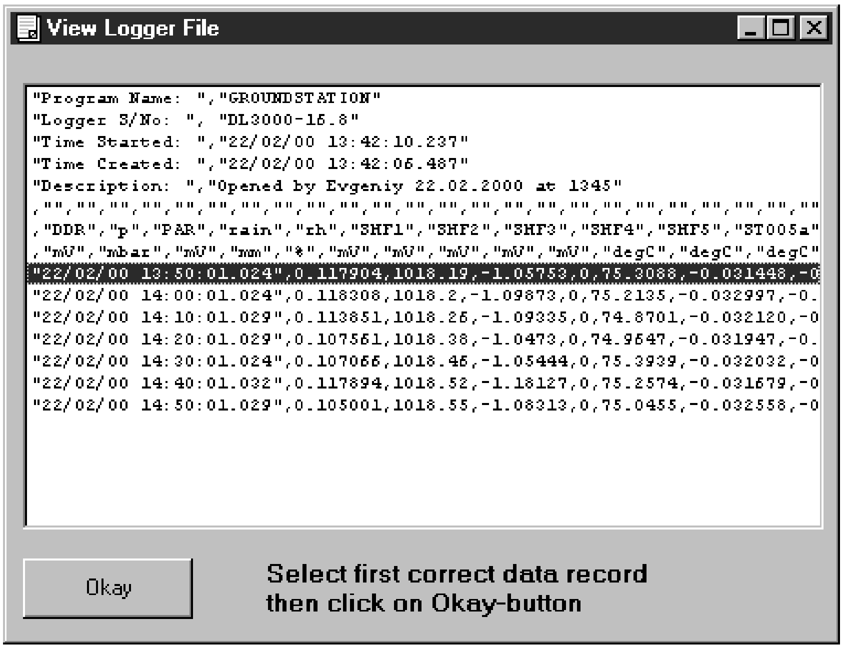
2.2 **DLLOGGEN** for DeltaT DL3000 datalogger



Work path: f:\Tmp
Data file: B0531340.DAT
Control file: Bog3.ctl

Figure 4: The DLLOGGEN dialog window

After selecting the data file and the control file a preview window opens (figure 5) where the first few lines of the data file are shown. The first correct data record which because of the header is normally the ninth line must be selected whereupon the window is left using the Okay button. The conversion process is then started using the Start conversion button. The progress of the conversion is shown by a blue bar.



```
"Program Name: ","GROUNDSTATION"
"Logger S/No: ", "DL3000-16.8"
"Time Started: ","22/02/00 13:42:10.237"
"Time Created: ","22/02/00 13:42:06.487"
"Description: ","Opened by Evgeniy 22.02.2000 at 1345"
,"","","","","","","","","","","","","","","","","","","","","","","",""
,"DDR","p","PAR","rain","rh","SHF1","SHF2","SHF3","SHF4","SHF5","ST005a"
,"mV","mbar","mV","mm","%","mV","mV","mV","mV","mV","degC","degC","degC"
"22/02/00 13:50:01.024",0.117904,1018.19,-1.05753,0,75.3088,-0.031448,-0
"22/02/00 14:00:01.024",0.118308,1018.2,-1.09873,0,75.2135,-0.032997,-0.
"22/02/00 14:10:01.029",0.113851,1018.26,-1.09335,0,74.8701,-0.032120,-0
"22/02/00 14:20:01.029",0.107561,1018.38,-1.0473,0,74.9647,-0.031947,-0.
"22/02/00 14:30:01.024",0.107066,1018.46,-1.05444,0,75.3939,-0.032032,-0
"22/02/00 14:40:01.032",0.117894,1018.52,-1.18127,0,75.2574,-0.031679,-0
"22/02/00 14:50:01.029",0.105001,1018.55,-1.08313,0,75.0455,-0.032558,-0
```

Select first correct data record
then click on Okay-button

Figure 5: The DLLOGGEN file preview window

2.3 The Control File

The most important work is to create the correct control file (extension ctl) for the corresponding datalogger program and instrumentation. The control file is similar to a program where a set of instructions can be used to perform the calculations. The control file consits of several sections where the section headers must be included in the way shown below. Comment lines must start with the character # (hash).

2.3.1 Description of the sections

- [`Number of variables`]
  The number of variables contained in one data record of the logger file is defined here. This number may be smaller than the actual number of data points per record in which case only this smaller number of values is read. Assuming records with 20 data points the number should be 20 or less. If it is less than 20 e.g. 10 then the **first** 10 values are read. It is not possible to force the program to read for example the first 10 values then to skip 5 values and to read the last 5 values. If the data records from old CAMPBELL dataloggers do not contain seconds in their timestamp then this number **must** be negative to indicate this fact.

- [`Missing value`]
  The value which is used to characterize bad or missing data points is defined here. Usually the value -9999 is used. If there is an instruction in the control file to check certain data lying in between specified limits and if one of those limits is exceeded then the missing value is written instead of the calculated value.

- [`Header line`]
  The header line which is written on top of the output file is defined here. The column header texts must be separated by comma and each included in double quotes. The header line may be wrapped to several lines here. If the actual line is being continued in the next line then the actual line must end with the character & immediately after the last character. Comment lines are not allowed in between continuation lines. Also no comments are allowed at the end of lines in this section.

- [`Declarations`]
  All used variables must be declared here. Variable names may consist of all characters between A and Z or a and z, the numbers 0 to 9 and the character _, they are treated case sensitive, the lenght of the names is in principle not limited. Variable names must not start with a number and spaces are not allowed within variable names. If n is the number of variables i.e. the number of data points per record then the first n variables declared here correspond to those data points which are read from the logger file in subsequent order. Additional variables may be declared and used for further calculations. The declared variables are consequently used

by the program to store the data read from the logger file and they can be accessed by their names in the processing functions described below together with additional declared variables. Comments my be written right of the variable names separated by at least one space, any special leading character is in this case not needed.
For historical reasons the [`Declarations`] section may be omitted. In this case variables are simply accessed by numbers in the [`Functions`] section where the first n variables correspond to the data points of the records of the original data logger file defined as the number of variables in the first section.

- [`Functions`]
  The functions to process the data are put in this section. The functions range from simple mathematic operations to more complex and special procedures including functions for checking data. Most of the functions have the following appearance: $a = fn(b, p_1, p_2, ..., p_n)$, where a is the variable in which the result of the function fn is stored, b is the input variable of the function and p_1 to p_n are parameters (numbers) of the function. An output variable (result of a function) may be the same as an input variable. Some functions need more than one input variable, some functions do not need any parameter and some functions (MEAN, MINI, MAXI) may have a variable number of input variables. The WRITE statement to write the data to the output file is no function (no return value) and its number of input variables is not fixed. Tables 1 to 3 show the complete list of functions and statements. Spaces are not allowed in functions. Long functions or statements may be wrapped to several lines. If the actual line is being continued in the next line then the actual line must end with the character & immediately after the last character. Comment lines are not allowed in between continuation lines. Comments my be written right of the functions separated by at least one space, any special leading character is in this case not needed.
  If the [`Declarations`] section is omitted then the functions appear for example as $1 = fn(2, p_1, p_2, ..., p_n)$ where the numbers 1 and 2 represent the variables no. 1 and no. 2. Only the first n variables corresponding to the data points of the records must be used without discontinuity, additional variables may be represented by any number. Example: in the first section the number 10 was set as number of variables per record. In this case the variables represented by the numbers 1 to 10 are connected to the 10 data points per record. Additional variables may then be represented by any number greater than 10.

- [`Formats`]
  The formats for writing the data to the output file are defined here. This section may be omitted in which case all data are written in the F8.2 format. The format descriptors are adopted from FORTRAN, so there are available the floating point format F, the integer format I and the scientific format with exponent E. In the F format the first number specifies the total number of digits including the decimal point and the second number specifies the number of digits right from the decimal point. F8.2 has then 5 digits left of the decimal point and 2

digits right of it. An extension was made to access formats for date and time, the T format. The T format must look like Td.h where d and h must be numbers between 0 and 3. If d is 0 then no date format is used, if d is 1 then only the day will be returned, if d is 2 then day and month will be returned and if d is 3 then day, month and year will be returned. If h is 0 then no time format is used, if h is 1 then only the hours will be returned, if h is 2 then hours and minutes will be returned and if h is 3 then hours, minutes and seconds will be returned. So T0.3 returns the number in the format hh:mm:ss and T3.3 would return DD.MM.YYYY hh:mm:ss. If this section is used then the formats must be separated by comma. It is possible to put one comma after the other in case the default format F8.2 should be used. A format line like T0.3,,,F8.5 means that the first number will be written in the time format hh:mm:ss, the next two numbers will be written in the default format F8.2 and the last number will be written in the format F8.5. The format line may be wrapped to several lines here. If the actual line is being continued in the next line then the actual line must end with the character & immediately after the last character. Comment lines are not allowed in between continuation lines. Also no comments are allowed at the end of lines in this section. If there are less formats specified than the WRITE statement contains variables then the rest of the variables are written to the output file using the standard F8.2 format.

2.3.2 A Sample Control File

The following example of a control file is used to process the data of a Campbell datalogger data file. The first four lines of this typical data file look as follows:

```
107,2000,167,1050,0,1.287,0,24.96,21.35,26.89
107,2000,167,1100,0,1.66,.1,26.61,17.81,26.87
107,2000,167,1110,0,.751,0,25.29,3.348,26.58
107,2000,167,1120,0,.218,1.1,23.78,-7.68,26.3
```

The first item is the output array ID which is automatically generated by the logger. The 2^{nd} to 5^{th} item are year, day of year, time (hhmm) and seconds which **must** be programmed to be output. The remaining 5 items are the logged data which is diffuse radiation (mV), precipitation (mm), logger temperature (^{o}C), current of solar batteries (A) and voltage of solar batteries (V). All data are in real units except the diffuse radiation. So in the [`Number of variables`] section the entry is 5, the missing value is set to $-9999.$ and the first 5 variables in the [`Declarations`] section are defined in the order they appear in the data records. An additional variable is defined named *power* which is battery current times battery voltage and gives the power consumtion or generation, respectively.

```
# Control file for Hainich solar logger
# From 28.01.2000 afternoon until ...
# Added battry voltage
[Number of variables]
5
[Missing value]
-9999.
[Header line]
"Date Time","ddr (W/m**2)","rain (mm)","Tlog (degC)","IBat (A)","UBat (V)","Power (W)"
[Declarations]
ddr                                     Diffuse downward radiation
rain                                    Precipitation
tlog                                    Logger temperature
ibat                                    Battery current
ubat                                    Battery voltage
power                                   Power consumtion/generation
[Functions]
ddr=MUL(ddr,194.1748)                   Diffuse downward radiation (986678)
ddr=SETLOW(ddr,0.0,0.0)                 Diffuse downward radiation greater zero
ddr=LIMITS(ddr,0.,1500.)                Diffuse downward radiation limits
ddr=SETLOW(ddr,2.0,0.0)                 Diffuse downward radiation greater two
rain=LIMITS(rain,0.,100.)               Rainfall limits
tlog=LIMITS(tlog,-50.,80.)              Logger temperature limits
ibat=LIMITS(ibat,-50.,110.)             Battery current limits
ubat=LIMITS(ubat,0.,50.)                Battery voltage limits
power=MUL(ibat,ubat)                    Battery current times Battery voltage
WRITE(ddr,rain,tlog,ibat,ubat,power)    Write all variables
[Formats]
F8.2,F8.1,F8.2,F8.2,F8.2,I6
```

The first function multiplies *ddr* with the calibration coefficient of 194.1748 and the next function tests *ddr* for the occurance of values less than zero and sets those values to zero. The next function checks whether *ddr* is within the limits of 0 and 1500 and sets it to -9999 if one of the limits is exceeded. The next four functions test the other variables lying in between certain limits. As next the variable *power* is calculated as the product of *ibat* and *ubat*. The last statement writes the data to the output file using the formats specified in the [`Formats`] section. The first four lines of the resulting output file are shown below:

```
"Date Time","ddr (W/m**2)","rain (mm)","Tlog (degC)","IBat (A)","UBat (V)","Power (W)"
15.06.2000 10:50:00,  249.90,    0.0,   24.96,   21.35,   26.89,   574
15.06.2000 11:00:00,  322.33,    0.1,   26.61,   17.81,   26.87,   479
15.06.2000 11:10:00,  145.83,    0.0,   25.29,    3.35,   26.58,    89
15.06.2000 11:20:00,   42.33,    1.1,   23.78,   -7.68,   26.30,  -202
```

Nearly the same result can be achieved using the old formulation of the control file. In this case the [`Declarations`] section is omitted where the variables are accessed by using just numbers. In the example the first five variables which correspond to the five data items in the original logger file are accessed with the numbers 1 to 5. An additional variable is used having the number 10 which corresponds to the variable *power* in the above example. Two important differences are the use of the functions MULVAR and MULVAL instead of just MUL. Using variable numbers instead of variable names makes it necessary to distinguisch between multiplying a variable by a number or by another variable (see tables 1 to 3). Omitting the [`Formats`] section leads to an output of all variables in the standard F8.2 format.

```
# Control file for Hainich solar logger
# From 28.01.2000 afternoon until ...
# Added battry voltage
[Number of variables]
5
[Missing value]
-9999.
[Header line]
"Date Time","ddr (W/m**2)","rain (mm)","Tlog (degC)","IBat (A)","UBat (V)","Power (W)"
[Functions]
1=MULVAL(1,194.1748)               Diffuse downward radiation (986678)
1=SETLOW(1,0.0,0.0)                Diffuse downward radiation greater zero
1=LIMITS(1,0.,1500.)               Diffuse downward radiation limits
1=SETLOW(1,2.0,0.0)                Diffuse downward radiation greater two
2=LIMITS(2,0.,100.)                Rainfall limits
3=LIMITS(3,-50.,80.)               Logger temperature limits
4=LIMITS(4,-50.,110.)              Battery current limits
5=LIMITS(5,0.,50.)                 Battery voltage limits
10=MULVAR(4,5)                     Battery current times Battery voltage
WRITE(1,2,3,4,5,10)                Write all variables
```

The resulting output file is slightly different compared to the above example:

```
"Date Time","ddr (W/m**2)","rain (mm)","Tlog (degC)","IBat (A)","UBat (V)","Power (W)"
15.06.2000 10:50:00,  249.90,    0.00,   24.96,   21.35,   26.89,  574.10
15.06.2000 11:00:00,  322.33,    0.10,   26.61,   17.81,   26.87,  478.55
15.06.2000 11:10:00,  145.83,    0.00,   25.29,    3.35,   26.58,   88.99
15.06.2000 11:20:00,   42.33,    1.10,   23.78,   -7.68,   26.30, -201.98
```

3 Batch converter **CSLOGBAT** for Campbell Dataloggers

This program is equivalent to CSLOGGEN but does not need any manual input to a form. The program uses an ini-file instead which must either reside in the programs directory and be named CSLogBat.ini or which can be defined as command line option and may then have any name. CSLOGBAT was created to be called from other programs, in particular from the Real Time Data Monitor program RTDM by CAMPBELL. From that program CSLOGBAT could be called for instance in the form

```
CSLOGBAT c:\data\metdata.ini
```

The ini-file contains information about the raw source data file as it comes from the datalogger, about the control file as described in the previous sections, about destination files and about the last processed record of the raw data file.

```
[Source File]
D:\cs_log\cr10x\daten\mpi_roof\roof_net.dat
[Control File]
D:\cs_log\cr10x\progs\mpi_roof\mpiroof2.ctl
[Destination File]
E:\data\mpi_roof.dat
E:\data\mpi_roof.csv
T:\wetter\mpi_roof.csv
[Last Timestamp]
121,2004,231,1400,.25,
```

Each time the program is executed, the following steps are performed:

- opening and reading the ini-file
- copying the source file to the first destination file
  e.g. copy `D:\cs_log\cr10x\daten\mpi_roof\roof_net.dat` to `E:\data\mpi_roof.dat`
- open the first destination file (e.g. `E:\data\mpi_roof.dat`) and moving to the line containing the value of last timestamp
- opening the second destination file (e.g. `E:\data\mpi_roof.csv`) in the append mode
- reading and converting the new lines of the first destination file and appending the results to the end of the second destination file
- copying the second destination file to the third destination file
  e.g. copy `E:\data\mpi_roof.csv` to `T:\wetter\mpi_roof.csv`,
  if the line for the third destination file is empty this action is omitted
- updating the ini-file with the new last timestamp

4 Control File Functions

In this section all functions are described which can be used within the control files to process the data.

1. Assignment

   $x =$SET(a) means $x = a$, where a is a variable or a number.
   For the old version where variable numbers are placed instead of variable names one of the following functions must be used:
   $x = \text{SETVAL}(n)$ means $x = n$, where n is a number
   $x = \text{SETVAR}(z)$ means $x = z$, where z is a variable

2. Change sign

   $x =$CHS(a) means $x = -a$, where a is a variable or a number.
   For the old version where variable numbers are placed instead of variable names one of the following functions must be used:
   $x = \text{CHSVAL}(n)$ means $x = -n$, where n is a number
   $x = \text{CHSVAR}(z)$ means $x = -z$, where z is a variable

3. Addition

   $x =$ADD(a, b) means $x = a + b$, where a and b are variables or numbers.
   For the old version where variable numbers are placed instead of variable names one of the following functions must be used:
   $x = \text{ADDVAL}(y, n)$ means $x = y + n$, where n is a number
   $x = \text{ADDVAR}(y, z)$ means $x = y + z$, where z is a variable

4. Subtraction

   $x =$SUB(a, b) means $x = a - b$, where a and b are variables or numbers.
   For the old version where variable numbers are placed instead of variable names one of the following functions must be used:
   $x = \text{SUBVAL}(y, n)$ means $x = y - n$, where n is a number
   $x = \text{SUBVAR}(y, z)$ means $x = y - z$, where z is a variable

5. Multiplication

   $x =$MUL(a, b) means $x = a \cdot b$, where a and b are variables or numbers.
   For the old version where variable numbers are placed instead of variable names one of the following functions must be used:
   $x = \text{MULVAL}(y, n)$ means $x = y \cdot n$, where n is a number
   $x = \text{MULVAR}(y, z)$ means $x = y \cdot z$, where z is a variable

6. Division

   $x =$DIV(a, b) means $x = a/b$, where a and b are variables or numbers.
   For the old version where variable numbers are placed instead of variable names one of the following functions must be used:
   $x =$ DIVVAL(y, n) means $x = y/n$, where n is a number
   $x =$ DIVVAR(y, z) means $x = y/z$, where z is a variable

7. Square root

   $x =$SQR(a) means $x = \sqrt{a}$, where a is a variable or a number.
   For the old version where variable numbers are placed instead of variable names the following function must be used:
   $x =$ SQRVAR(y) means $x = \sqrt{y}$, where y is a variable

8. Exponentiation of e

   $x =$EXP(a) means $x = \exp(a)$, where a is a variable or a number.
   For the old version where variable numbers are placed instead of variable names the following function must be used:
   $x =$ EXPVAR(y) means $x = \exp(y)$, where y is a variable

9. Natural logarithm

   $x =$LOG(a) means $x = \ln(a)$, where a is a variable or a number.
   For the old version where variable numbers are placed instead of variable names the following function must be used:
   $x =$ LOGVAR(y) means $x = \ln(y)$, where y is a variable

10. Exponentiation

    $x =$POT(a, b) means $x = a^b$, where a and b are variables or numbers.
    For the old version where variable numbers are placed instead of variable names one of the following functions must be used:
    $x =$ POTVAL(y, n) means $x = y^n$, where n is a number
    $x =$ POTVAR(y, z) means $x = y^z$, where z is a variable

11. Apply linear function

    $x =$LIN(y, a_0, a_1) means $x = a_0 + a_1 \cdot y$,
    where a_0 and a_1 are variables or numbers, for the old version only numbers are allowed.

12. Apply 2^{nd} order function

    $x =$QUAD(y, a_0, a_1, a_2) means $x = a_0 + a_1 \cdot y + a_2 \cdot y^2$,
    where a_0, a_1 and a_2 are variables or numbers, for the old version only numbers are allowed.

13. Apply 3^{rd} order function

    $x =$CUBIC(y, a_0, a_1, a_2, a_3) means $x = a_0 + a_1 \cdot y + a_2 \cdot y^2 + a_3 \cdot y^3$,
    where a_0, a_1, a_2 and a_3 are variables or numbers, for the old version only numbers are allowed.

14. Calculate fraction of day from hours, minutes and seconds

    $x =$HMS(h, m, s) means $x = (h + m/60 + s/3600)/24$,
    where h, m and s (hours, minutes and seconds) are variables or numbers, for the old version only variables are allowed.

15. Bitwise test

    $x =$BIT_TEST(y, b) means $x = 1$ if bit b ist set in y otherwise $x = 0$,
    where b is a variable or a number, for the old version only a number is allowed. Counting of b starts at 1 which is the LSB. If b is negative then all bits are inverted.

16. Replacement of underflows by new value

    $x =$SETLOW(y, l_o, l_n) means IF $(y < l_o)$ THEN $x = l_n$ ELSE $x = y$,
    where l_o and l_n are variables or numbers, for the old version only numbers are allowed. This function may be used to adjust small negative values of short wave radiation during nighttime to zero values.

17. Replacement of overflows by new value

    $x =$SETHIGH(y, l_o, l_n) means IF $(y > l_o)$ THEN $x = l_n$ ELSE $x = y$,
    where l_o and l_n are variables or numbers, for the old version only numbers are allowed. This function may be used to adjust relative humidity values of a little bit more than 100 % to 100 %.

18. Replacement of underflows or overflows by the missing value

    $x =$LIMITS(y, l_l, l_h) means
    IF $(y < l_l)$ OR $(y > l_h)$ THEN $x =$ [*missing value*] ELSE $x = y$,
    where l_l and l_h are variables or numbers, for the old version only numbers are allowed. This function may be used to check values lying in between certain limits. If one of the limits is exceeded the value is set to the missing value defined in the control file.

19. Calculation of mean value

    $x =$MEAN$(y_1, y_2, \cdots, y_n)$ means $x = (y_1 + y_2 + \cdots + y_n)/n$,
    where $y_1, y_2, \cdots, y_n$ are variables or numbers, for the old version only variables are allowed.

20. Calculation of minimum value

    $x =$MINI$(y_1, y_2, \cdots, y_n)$ means $x = \min(y_1, y_2, \cdots, y_n)$,
    where $y_1, y_2, \cdots, y_n$ are variables or numbers, for the old version only variables are allowed.

21. Calculation of maximum value

    $x =$MAXI$(y_1, y_2, \cdots, y_n)$ means $x = \max(y_1, y_2, \cdots, y_n)$,
    where $y_1, y_2, \cdots, y_n$ are variables or numbers, for the old version only variables are allowed.

22. Calculation of total radiation from net radiometer

    $x =$MET_TORAD(y, f_s, f_l, R_s, T_p) where
    y is the output voltage of the net radiometer in mV,
    f_s is the factor for short wave radiation (reciprocal value of sensitivity) in $\mathrm{W\,m^{-2}}$ per mV,
    f_l is the factor for long wave radiation (reciprocal value of sensitivity) in $\mathrm{W\,m^{-2}}$ per mV,
    R_s is the short wave radiation in $\mathrm{W\,m^{-2}}$,
    T_p is the temperature of the net radiometer body in ^{o}C.

    The total radiation in $\mathrm{W\,m^{-2}}$ is calculated according to the following formula:

    $$x = y \cdot f_l - R_s \cdot \left(\frac{f_l}{f_s} - 1\right) + \sigma \cdot (T_p + 273.16)^4$$

    or in case of missing R_s

    $$x = y \cdot \left(\frac{f_l + f_s}{2}\right) + \sigma \cdot (T_p + 273.16)^4$$

    where $\sigma = 5.67051 \cdot 10^8\,\mathrm{W\,m^{-2}\,K^{-4}}$ is the STEPHAN-BOLTZMANN-CONSTANT.

    All parameters may be variables or numbers, for the old version f_s and f_l must be numbers the others must be variables.

23. Calculation of long wave radiation from net radiometer

    $x =$MET_LWRAD(y, f_l, T_p) where
    y is the output voltage of the net radiometer in mV,
    f_l is the factor for long wave radiation (reciprocal value of sensitivity) in $\mathrm{W\,m^{-2}}$ per mV,
    T_p is the temperature of the net radiometer body in ^{o}C.

    The total radiation in $\mathrm{W\,m^{-2}}$ is calculated according to the following formula:

    $$x = y \cdot f_l + \sigma \cdot (T_p + 273.16)^4$$

    where $\sigma = 5.67051 \cdot 10^8\,\mathrm{W\,m^{-2}\,K^{-4}}$ is the STEPHAN-BOLTZMANN-CONSTANT.

    All parameters may be variables or numbers, for the old version f_l must be a number the others must be variables.

24. Calculation of radiation temperature from long wave radiation

    $x =$MET_TRAD(R_l, ε) where
    R_l is the long wave radiation in $\mathrm{W\,m^{-2}}$,
    ε is the long wave emissivity of the surface (between 0 and 1).

    The radiation temperature in ^{o}C is calculated according to the following formula:

$$x = \sqrt[4]{\frac{R_l}{\sigma \cdot \varepsilon}} - 273.16$$
where $\sigma = 5.67051 \cdot 10^8\,\mathrm{W\,m^{-2}\,K^{-4}}$ is the Stephan-Boltzmann-Constant.

Both parameters may be variables or numbers, for the old version ε must be a number R_l must be a variable.

25. Calculation of albedo from short wave downward and upward radiation

    x =MET_ALB($R_s^{\downarrow}, R_s^{\uparrow}$) where
    $R_s^{\downarrow}$ is the short wave downward radiation in $\mathrm{W\,m^{-2}}$,
    $R_s^{\uparrow}$ is the short wave upward radiation in $\mathrm{W\,m^{-2}}$,

    The albedo in % is calculated according to the following formula:
    $$x = 100 \cdot \left(\frac{R_s^{\uparrow}}{R_s^{\downarrow}}\right)$$
    If $R_s^{\downarrow} < 50\,\mathrm{W\,m^{-2}}$ or $R_s^{\uparrow} < 10\,\mathrm{W\,m^{-2}}$ the result is [*missing value*].

    Both parameters may be variables or numbers, for the old version both must be variables.

26. Calculation of albedo from short wave downward and upward radiation with limits

    x =MET_ALBL($R_s^{\downarrow}, R_s^{\uparrow}, R_s^{\downarrow}\_limit, R_s^{\uparrow}\_limit$) where
    $R_s^{\downarrow}$ is the short wave downward radiation in $\mathrm{W\,m^{-2}}$,
    $R_s^{\uparrow}$ is the short wave upward radiation in $\mathrm{W\,m^{-2}}$,
    $R_s^{\downarrow}\_limit$ is the short wave downward radiation limit in $\mathrm{W\,m^{-2}}$,
    $R_s^{\uparrow}\_limit$ is the short wave upward radiation limit in $\mathrm{W\,m^{-2}}$,

    The albedo in % is calculated according to the following formula:
    $$x = 100 \cdot \left(\frac{R_s^{\uparrow}}{R_s^{\downarrow}}\right)$$
    If $R_s^{\downarrow} < R_s^{\downarrow}\_limit$ or $R_s^{\uparrow} < R_s^{\uparrow}\_limit$ the result is [*missing value*].

    All four parameters may be variables or numbers, for the old version all must be variables.

27. Calculation of saturation water vapour pressure

    x =MET_VPMAX(T) where
    T is the air temperature in ^{o}C.

    The saturation water vapour pressure in mbar (hPa) is calculated according to the following formula:
    $$x = 6.1078 \cdot \exp\left(\frac{17.08085 \cdot T}{234.175 + T}\right)$$
    The parameter may be a variable or a number, for the old version it must be a variable.

28. Calculation of actual water vapour pressure

    $x =$MET_VPACT(T, rh) where
    T is the air temperature in ^{o}C,
    rh is the relative humidity in %.

    The actual water vapour pressure in mbar (hPa) is calculated according to the following formulas:
    $$E_s = 6.1078 \cdot \exp\left(\frac{17.08085 \cdot T}{234.175 + T}\right)$$
    $$x = \frac{E_s \cdot rh}{100}$$
    Both parameters may be variables or numbers, for the old version both must be variables.

29. Calculation of water vapour pressure deficit

    $x =$MET_VPDEF(T, rh) where
    T is the air temperature in ^{o}C,
    rh is the relative humidity in %.

    The water vapour pressure deficit in mbar (hPa) is calculated according to the following formulas:
    $$E_s = 6.1078 \cdot \exp\left(\frac{17.08085 \cdot T}{234.175 + T}\right)$$
    $$E = \frac{E_s \cdot rh}{100}$$
    $$x = E_s - E$$

    Both parameters may be variables or numbers, for the old version both must be variables.

30. Calculation of specific humidity

    $x =$MET_SH(T, rh, p) where
    T is the air temperature in ^{o}C,
    rh is the relative humidity in %,
    p is the air pressure in mbar (hPa).

    The specific humidity in $\mathrm{g\,kg^{-1}}$ is calculated according to the following formulas:
    $$E_s = 6.1078 \cdot \exp\left(\frac{17.08085 \cdot T}{234.175 + T}\right)$$
    $$E = \frac{E_s \cdot rh}{100}$$
    $$x = 622 \cdot \frac{E}{p - 0.378 \cdot E}$$
    All parameters may be variables or numbers, for the old version all must be variables.

31. Calculation of potential temperature

x =MET_TPOT(T, p) where
T is the air temperature in ^{o}C,
p is the air pressure in mbar (hPa).

The potential temperature in K is calculated according to the following formula:

$$x = (T + 273.16) \cdot \left(\frac{1000}{p}\right)^{0.286}$$

Both parameters may be variables or numbers, for the old version both must be variables.

32. Calculation of air density

x =MET_RHO(T, rh, p) where
T is the air temperature in ^{o}C,
rh is the relative humidity in %,
p is the air pressure in mbar (hPa).

The air density in $\mathrm{kg\,m^{-3}}$ is calculated according to the following formulas:

$$E_s = 6.1078 \cdot \exp\left(\frac{17.08085 \cdot T}{234.175 + T}\right)$$

$$E = \frac{E_s \cdot rh}{100}$$

$$sh = 622 \cdot \frac{E}{p - 0.378 \cdot E}$$

$$T_v = ((T + 273.16) \cdot (1 + 0.000608 \cdot sh)) - 273.16$$

$$x = \frac{p \cdot 100}{287.05 \cdot (T_v + 273.16)}$$

All parameters may be variables or numbers, for the old version all must be variables.

33. Calculation of dew point temperature

x =MET_DPT(T, rh) where
T is the air temperature in ^{o}C,
rh is the relative humidity in %.

The dew point temperature in ^{o}C is calculated according to the following formulas:

$$E_s = 6.1078 \cdot \exp\left(\frac{17.08085 \cdot T}{234.175 + T}\right)$$

$$E = \frac{E_s \cdot rh}{100}$$

$$x = 234.175 \cdot \frac{\ln(E/6.1078)}{17.08085 - \ln(E/6.1078)}$$

Both parameters may be variables or numbers, for the old version both must be variables.

34. Calculation of water vapour concentration

x =MET_H2OC(T, rh, p) where
T is the air temperature in ^{o}C,

rh is the relative humidity in %,
p is the air pressure in mbar (hPa).

The water vapour concentration in $\mathrm{mmol\,mol^{-1}}$ is calculated according to the following formulas:

$$E_s = 6.1078 \cdot \exp\left(\frac{17.08085 \cdot T}{234.175 + T}\right)$$

$$E = \frac{E_s \cdot rh}{100}$$

$$x = \frac{0.1 \cdot E}{0.001 \cdot p \cdot 100 \cdot 0.001}$$

All parameters may be variables or numbers, for the old version all must be variables.

35. Calculation of relative humidity from dry and wet bulb temperature

$x =$MET_PSY_RH(T, T_f, p, i_{sat}) where
T is the air temperature in °C,
T_f is the wet bulb temperature in °C,
p is the air pressure in mbar (hPa),
i_{sat} is a flag for saturation type, where
  $i_{sat} = 0$ if saturation above water,
  $i_{sat} = 1$ if saturation above water below 0 °C,
  $i_{sat} = 2$ if saturation above ice.

The relative humidity in % is calculated according to the following formulas:

$$E_s = 6.1078 \cdot \exp\left(\frac{17.08085 \cdot T}{234.175 + T}\right)$$

$$E_{sf} = C_1 \cdot \exp\left(\frac{C_2 \cdot T_f}{C_3 + T_f}\right)$$

$$E = E_{sf} - B \cdot p \cdot (T - T_f)$$

$$x = \frac{100 \cdot E}{E_s}$$

The constants C_1, C_2, C_3 and B are different depending on i_{sat}:

If $i_{sat} = 0$ then
  $C_1 = 6.1078$
  $C_2 = 17.08085$
  $C_3 = 234.175$
  $B = 0.00066$

If $i_{sat} = 1$ then
  $C_1 = 6.1078$
  $C_2 = 17.84362$
  $C_3 = 245.425$
  $B = 0.00066$

If $i_{sat} = 2$ then

$$C_1 = 6.10714$$
$$C_2 = 22.44294$$
$$C_3 = 272.44$$
$$B = 0.00058$$

All parameters may be variables or numbers, for the old version i_{sat} must be a number the rest must be variables.

36. Calculation of relative humidity from dew point temperature

    $x =$MET_DPT_RH(T, T_d, i_{sat}) where
    T is the air temperature in ^{o}C,
    T_d is the dew point temperature in ^{o}C,
    i_{sat} is a flag for saturation type as above.

    The relative humidity in % is calculated according to the following formulas:

    $$E_s = 6.1078 \cdot \exp\left(\frac{17.08085 \cdot T}{234.175 + T}\right)$$
    $$E = C_1 \cdot \exp\left(\frac{C_2 \cdot T_d}{C_3 + T_d}\right)$$
    $$x = \frac{100 \cdot E}{E_s}$$

    The constants C_1, C_2, C_3 and B are again depending on i_{sat} as mentioned above.

    All parameters may be variables or numbers, for the old version i_{sat} must be a number the rest must be variables.

37. Calculation of relative humidity from water vapour concentration

    $x =$MET_H2OC_RH$(T, [\mathrm{H_2O}], p)$ where
    T is the air temperature in ^{o}C,
    $[\mathrm{H_2O}]$ is the water vapour concentration in $\mathrm{mmol\,mol^{-1}}$,
    p is the air pressure in mbar (hPa).

    The relative humidity in % is calculated according to the following formulas:

    $$E_s = 6.1078 \cdot \exp\left(\frac{17.08085 \cdot T}{234.175 + T}\right)$$
    $$E = 10 \cdot [\mathrm{H_2O}] \cdot 0.001 \cdot p \cdot 100 \cdot 0.001$$
    $$x = \frac{100 \cdot E}{E_s}$$

    All parameters may be variables or numbers, for the old version all must be variables.

38. Rotation of wind direction

    $x =$MET_WDROT(wd, a) where
    wd is the wind direction in degree,
    a is the rotation angle in degree (positive is clockwise).

The rotated wind direction is calculated according to the following formulas:

$x = wd + a$

IF $x < 0$ THEN $x = x + 360$

IF $x \geq 360$ THEN $x = x - 360$

Both parameters may be variables or numbers, for the old version a must be a number, wd must be a variable.

39. Rotation of u-component of wind vector

    $x =$MET_UROT(u, v, a) where
    u is the u-component of the wind vector,
    v is the v-component of the wind vector,
    a is the rotation angle in degree (positive is clockwise).

    The rotated u-component is calculated according to the following formula:

    $x = u \cdot \cos(a) + v \cdot \sin(a)$

    All three parameters may be variables or numbers, for the old version a must be a number, u and v must be variables.

40. Rotation of v-component of wind vector

    $x =$MET_VROT(u, v, a) where
    u is the u-component of the wind vector,
    v is the v-component of the wind vector,
    a is the rotation angle in degree (positive is clockwise).

    The rotated v-component is calculated according to the following formula:

    $x = -u \cdot \sin(a) + v \cdot \cos(a)$

    All three parameters may be variables or numbers, for the old version a must be a number, u and v must be variables.

41. Calculation of wind velocity from u- and v-component of wind vector

    $x =$MET_UV_WV(u, v) where
    u is the u-component of the wind vector,
    v is the v-component of the wind vector.

    The horizontal wind velocity is calculated according to the following formula:

    $x = \sqrt{u^2 + v^2}$

    Both parameters may be variables or numbers, for the old version u and v must be variables.

42. Calculation of wind direction from u- and v-component of wind vector

$x =$MET_UV_WD(u, v) where
u is the u-component of the wind vector,
v is the v-component of the wind vector.

The horizontal wind velocity is calculated according to the following formulas:

IF $u = 0$ AND $v = 0$ THEN $x = 0$
IF $u = 0$ AND $v < 0$ THEN $x = 360$
IF $u = 0$ AND $v > 0$ THEN $x = 180$
IF $u > 0$ THEN $x = 270 - \arctan\left(\frac{v}{u}\right)$
IF $u < 0$ THEN $x = 90 - \arctan\left(\frac{v}{u}\right)$

Both parameters may be variables or numbers, for the old version u and v must be variables.

43. Calculation of u-component of wind vector from wind velocity and wind direction

    $x =$MET_WVWD_U(wv, wd) where
    wv is the horizontal wind velocity,
    wd is the horizontal wind direction.

    The u-component of the wind vector is calculated according to the following formula:

    $x = -wv \cdot \sin(wd)$

    Both parameters may be variables or numbers, for the old version wv and wd must be variables.

44. Calculation of v-component of wind vector from wind velocity and wind direction

    $x =$MET_WVWD_V(wv, wd) where
    wv is the horizontal wind velocity,
    wd is the horizontal wind direction.

    The v-component of the wind vector is calculated according to the following formula:

    $x = -wv \cdot \cos(wd)$

    Both parameters may be variables or numbers, for the old version wv and wd must be variables.

45. IF-Statements

    $x =$IFEQ(y, a_0, a_1, a_2) means IF $y = a_0$ THEN $x = a_1$ ELSE $x = a_2$
    $x =$IFNE(y, a_0, a_1, a_2) means IF $y \neq a_0$ THEN $x = a_1$ ELSE $x = a_2$
    $x =$IFLE(y, a_0, a_1, a_2) means IF $y \leq a_0$ THEN $x = a_1$ ELSE $x = a_2$
    $x =$IFGE(y, a_0, a_1, a_2) means IF $y \geq a_0$ THEN $x = a_1$ ELSE $x = a_2$
    $x =$IFLT(y, a_0, a_1, a_2) means IF $y < a_0$ THEN $x = a_1$ ELSE $x = a_2$
    $x =$IFGT(y, a_0, a_1, a_2) means IF $y > a_0$ THEN $x = a_1$ ELSE $x = a_2$

    All parameters may be variables or numbers, for the old version y must be a variable and a_0, a_1, a_2 must be numbers.

46. Write variables to a file

    $\mathrm{WRITE}(x_1, x_2, x_3, \cdots, x_n)$

    All variables are written to the file separated by commas.

Tabular representation of all functions

Function	**Old Version**	**Description**
$x = \mathrm{SET}(a)$		Assignment (number or variable to variable)
	$x = \mathrm{SETVAL}(n)$	Assignment (number to variable)
	$x = \mathrm{SETVAR}(z)$	Assignment (variable to variable)
$x = \mathrm{CHS}(a)$		Change sign (of number or of variable)
	$x = \mathrm{CHSVAL}(n)$	Change sign (of number)
	$x = \mathrm{CHSVAR}(z)$	Change sign (of variable)
$x = \mathrm{ADD}(a, b)$		Addition (numbers or variables)
	$x = \mathrm{ADDVAL}(y, n)$	Addition (variable plus number)
	$x = \mathrm{ADDVAR}(y, z)$	Addition (variable plus variable)
$x = \mathrm{SUB}(a, b)$		Subtraction (numbers or variables)
	$x = \mathrm{SUBVAL}(y, n)$	Subtraction (variable minus number)
	$x = \mathrm{SUBVAR}(y, z)$	Subtraction (variable minus variable)
$x = \mathrm{MUL}(a, b)$		Multiplication (numbers or variables)
	$x = \mathrm{MULVAL}(y, n)$	Multiplication (variable times number)
	$x = \mathrm{MULVAR}(y, z)$	Multiplication (variable times variable)
$x = \mathrm{DIV}(a, b)$		Division (numbers or variables)
	$x = \mathrm{DIVVAL}(y, n)$	Division (variable div. by number)
	$x = \mathrm{DIVVAR}(y, z)$	Division (variable div. by variable)
$x = \mathrm{SQR}(a)$		Square root of number or variable
	$x = \mathrm{SQRVAR}(z)$	Square root of variable
$x = \mathrm{EXP}(a)$		Exponentiate e by number or variable
	$x = \mathrm{EXPVAR}(z)$	Exponentiate e by variable
$x = \mathrm{LOG}(a)$		Natural logarithm of number or variable
	$x = \mathrm{LOGVAR}(z)$	Natural logarithm of variable
$x = \mathrm{POT}(a, b)$		Exponentiate numbers or variables
	$x = \mathrm{POTVAL}(y, n)$	Exponentiate variable by number
	$x = \mathrm{POTVAR}(y, z)$	Exponentiate variable by variable
$x = \mathrm{LIN}(y, a_0, a_1)$		Applies linear function
$x = \mathrm{QUAD}(y, a_0, a_1, a_2)$		Applies 2^{nd} degree function
$x = \mathrm{CUBIC}(y, a_0, a_1, a_2, a_3)$		Applies 3^{rd} degree function
$x = \mathrm{HMS}(h, m, s)$		Fraction of day from hours, minutes and seconds
$x = \mathrm{BIT\_TEST}(y, b)$		Bitwise test for bit b in y
$x = \mathrm{SETLOW}(y, l_o, l_n)$		Replacement of underflows by new value
$x = \mathrm{SETHIGH}(y, l_o, l_n)$		Replacement of overflows by new value
$x = \mathrm{LIMITS}(y, l_l, l_h)$		Replacement of under or overflows by missing value
$x = \mathrm{MEAN}(y_1, y_2, \cdots, y_n)$		Mean value
$x = \mathrm{MINI}(y_1, y_2, \cdots, y_n)$		Minimum value
$x = \mathrm{MAXI}(y_1, y_2, \cdots, y_n)$		Maximum value

Table 1: Mathematical functions to be used in control files

Function	**Description**
x =MET_TORAD(y, f_s, f_l, R_s, T_p)	Total radiation
x =MET_LWRAD(y, f_l, T_p)	Long wave radiation
x =MET_TRAD(R_l, ε)	Radiation temperature
x =MET_ALB$(SWDR, SWUR)$	Albedo
x =MET_ALBL$(SWDR, SWUR, SWDR_l, SWUR_l)$	Albedo with limits
x =MET_VPMAX(T)	Saturation water vapour pressure
x =MET_VPACT(T, rh)	Actual water vapour pressure
x =MET_VPDEF(T, rh)	Water vapour pressure deficit
x =MET_SH(T, rh, p)	Specific humidity
x =MET_TPOT(T, p)	Potential temperature
x =MET_RHO(T, rh, p)	Air density
x =MET_DPT(T, rh)	Dew point temperature
x =MET_H2OC(T, rh, p)	Water vapour concentration
x =MET_PSY_RH(T, T_f, p, i_{sat})	Rel. humidity from dry and wet bulb temperature
x =MET_DPT_RH(T, T_{dew}, p, i_{sat})	Rel. humidity from temperature and dew point
x =MET_H2OC_RH$(T, [\mathrm{H_2O}], p)$	Rel. humidity from temperature and H_2O concentration
x =MET_WDROT(wd, a)	Rotate wind direction
x =MET_UROT(u, v, a)	Rotate u-component of wind vector
x =MET_VROT(u, v, a)	Rotate v-component of wind vector
x =MET_UV_WV(u, v)	Calculate wind velocity from u- and v-component of wind vector
x =MET_UV_WD(u, v)	Calculate wind direction from u- and v-component of wind vector
x =MET_WVWD_U(wv, wd)	Calculate u-component of wind vector from wind velocity and direction
x =MET_WVWD_V(wv, wd)	Calculate v-component of wind vector from wind velocity and direction

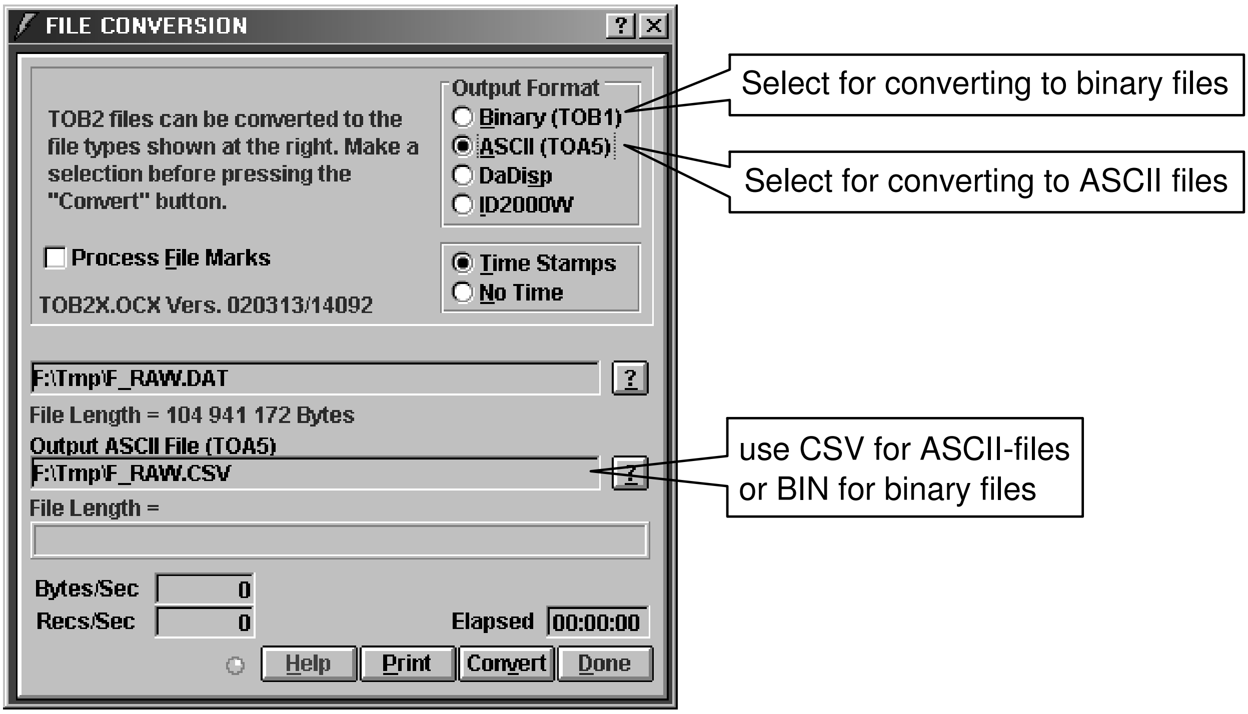
Table 2: Meteorological functions to be used in control files

Function	**Description**
#	Start of comment line
x =IFEQ(y, a_0, a_1, a_2)	IF-statement (equal)
x =IFNE(y, a_0, a_1, a_2)	IF-statement (not equal)
x =IFLE(y, a_0, a_1, a_2)	IF-statement (less equal)
x =IFGE(y, a_0, a_1, a_2)	IF-statement (greater equal)
x =IFLT(y, a_0, a_1, a_2)	IF-statement (less than)
x =IFGT(y, a_0, a_1, a_2)	IF-statement (greater than)
WRITE$(x_1, x_2, x_3, \cdots, x_n)$	Write variables to file

Table 3: Logical and output functions to be used in control files

5 The eddy covariance data converter CR5000EC

If eddy covariance data have been collected by a CR5000 datalogger using the logger program Mob_Eddy.cr5 (by O. Kolle) then the raw data can be converted to binary slt-files to be processed using the Eddysoft software suite (by O. Kolle). Before CR5000EC can be used it is necessary to convert the original raw data file as it comes from the compact flash card using the Campbell software PC9000. The data may be converted to an ASCII file which may become very large or to a binary file which is smaller but not readable.



TOB2 files can be converted to the file types shown at the right. Make a selection before pressing the "Convert" button.

Output Format
○ Binary (TOB1)
● ASCII (TOA5)
○ DaDisp
○ ID2000W

☐ Process File Marks

● Time Stamps
○ No Time

TOB2X.OCX Vers. 020313/14092

F:\Tmp\F_RAW.DAT
File Length = 104 941 172 Bytes
Output ASCII File (TOA5)
F:\Tmp\F_RAW.CSV
File Length =

Bytes/Sec 0
Recs/Sec 0
Elapsed 00:00:00

Select for converting to binary files

Select for converting to ASCII files

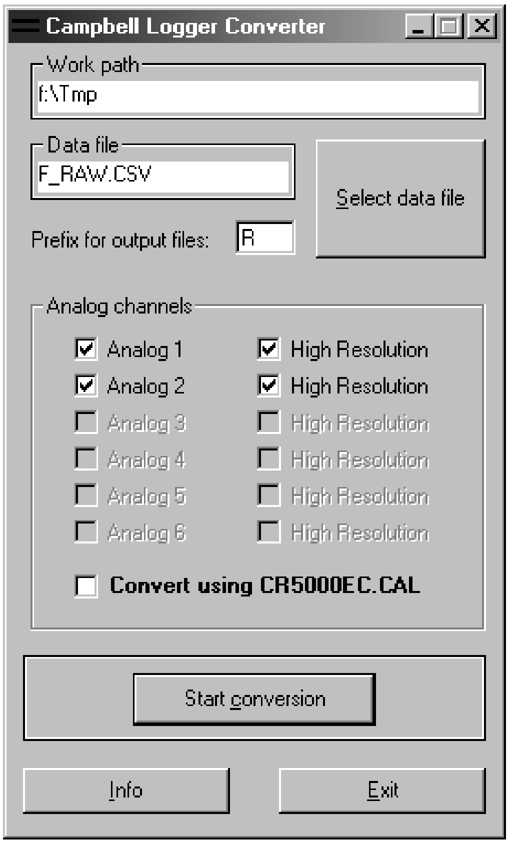
use CSV for ASCII-files
or BIN for binary files

Figure 6: The PC9000 file conversion dialog window with hints

A typical ASCII file looks as follows:

```
"TOA5","MOB_EDDYCPU:MOB_EDDY.CR5","CR5000","1118","1.3.01","CPU:MOB_EDDY.CR5","18630","F_RAW"
"TIMESTAMP","fu(1)","fv(1)","fw(1)","ft(1)","fc(1)","fh(1)"
"TS","m/s","m/s","m/s","degC","mV","mV"
"","Smp","Smp","Smp","Smp","Smp","Smp"
"2003-03-02 11:34:44.8",-2.095,0.218,-0.673,28.33,1434,2234
"2003-03-02 11:34:44.9",-1.961,0.247,-0.594,28.34,1436,2233
"2003-03-02 11:34:45",-1.974,0.148,-0.565,28.33,1435,2238
"2003-03-02 11:34:45.1",-1.895,-0.002,-0.433,28.32,1433,2238
"2003-03-02 11:34:45.2",-1.932,0.027,-0.312,28.32,1432,2237
"2003-03-02 11:34:45.3",-1.813,0.185,-0.163,28.36,1434,2237
"2003-03-02 11:34:45.4",-1.563,0.207,-0.241,28.37,1435,2233
"2003-03-02 11:34:45.5",-1.593,0.066,-0.362,28.32,1434,2234
"2003-03-02 11:34:45.6",-1.555,-0.104,-0.312,28.31,1437,2237
"2003-03-02 11:34:45.7",-1.663,-0.134,-0.283,28.28,1433,2236
```

The file contains from left to right the timestamp then the wind components u, v and w, then the temperature in °C and then the CO_2 and H_2O voltage signals in mV. Both the latter may also be stored in units of a concentration in which case the option to convert the data must be selected in the CR5000EC dialog window.



Work path: f:\Tmp

Data file: F_RAW.CSV

Prefix for output files: R

Analog channels

☑ Analog 1 ☑ High Resolution
☑ Analog 2 ☑ High Resolution
☐ Analog 3 ☐ High Resolution
☐ Analog 4 ☐ High Resolution
☐ Analog 5 ☐ High Resolution
☐ Analog 6 ☐ High Resolution

☐ **Convert using CR5000EC.CAL**

Figure 7: The CR5000EC dialog window

After selecting an ASCII or binary raw data file the correct number of additional analog channels is offered to be selectable. If desired the high resolution option can be switched on individually for each channel. If switched off then the data are stored with a resolution of 1 mv otherwise the resolution is 0.1 mv. If the raw data are not stored in mV they must be converted by CR5000EC into Millivolt. This is forced by switching on the option Convert using CR5000EC.CAL. This file must exist in the programs folder and must look as follows:

```
[analog input channel 1]
0 , 5 , 10 , 25
[analog input channel 2]
0 , 5 , 0 , 1500
[analog input channel 3]
0 , 5 , 0 , 5
[analog input channel 4]
0 , 5 , 0 , 5
[analog input channel 5]
0 , 5 , 0 , 5
[analog input channel 6]
0 , 5 , 0 , 5
```

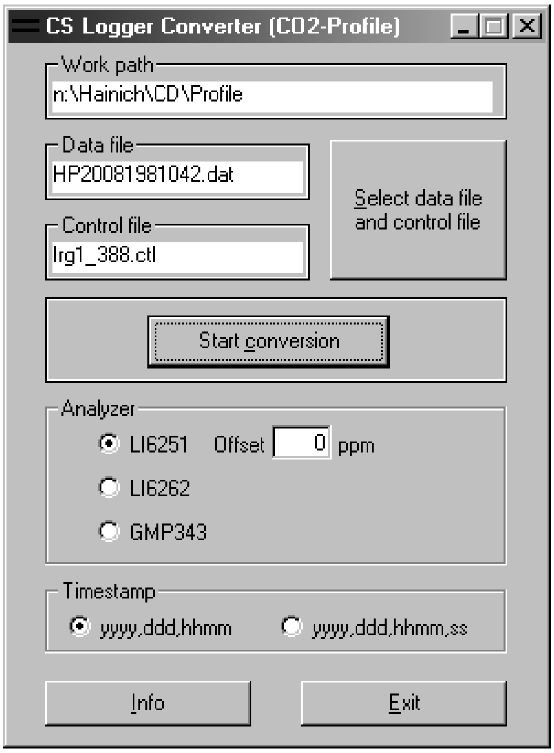
Always all 6 possible channels must be specified in the file, the four numbers stand for

1. minimum voltage signal in V,
2. maximum voltage signal in V,
3. minimum number in units as stored in the raw data file (corresponding to the minimum voltage),
4. maximum number in units as stored in the raw data file (corresponding to the maximum voltage).

Having chosen an appropriate prefix character for the output files the conversion can be launched. The program generates half hour raw data files following the naming convention `Xyyyydddhhmm.slt` where `X` stands for the prefix character, `yyyy` for the year, `ddd` for the day of the year, `hhmm` for hour and minute of the first record in the file. These files are fully compatible with those generated and used in the EDDYSOFT program suite.

6 The CO_2 data converter **CSLOGCO2**

If profiles of CO_2 or of CO_2 and H_2O are measured with an LI6251 or an LI6262 gas analyzer or a GMP343 carbon dioxide probe in conjunction with the gas flow switching unit made by MPI-BGC and the data are logged with a CR10X or CR23X datalogger then the CSLOGCO2 program can be used to convert and check the data.



Work path: n:\Hainich\CD\Profile
Data file: HP20081981042.dat
Control file: lrg1_388.ctl
Analyzer: ● LI6251 Offset 0 ppm ○ LI6262 ○ GMP343
Timestamp: ● yyyy,ddd,hhmm ○ yyyy,ddd,hhmm,ss

Figure 8: The CSLOGCO2 dialog window

The datalogger file format differs from that described above. The measurements are taken in the following way: the logger controls the switching unit where within e.g. one 10 minute interval the desired number of levels is measured. To measure one level the corresponding inlet line is switched to the inlet of the gas analyzer by a solenoid valve. All other inlet lines are switched to a common outlet. Two membrane pumps perform the airflow where one pump sucks only the air from the level which is actually measured, the other pump sucks air from all other inlets so that all hoses and the valves are permanently flushed. After the logger switched to a new level (inlet) the logging is stopped for 10 seconds to let the system adapt to the air of that level. Then the logging starts with a sampling rate of 1 Hz. Before switching to the next level the mean value of the logged data is written to the loggers memory. Depending how many levels are measured the logging period for one level is a multiple of one minute. If for example the maximum of 9 levels is used then the logging period for the lowest level could be two minutes where for all other levels it has to be one minute.

The switching unit is controlled by the logger using maximum four control ports in binary mode where C1 is coding for 1, C2 is coding for 2, C3 is coding for 4 and C4 is coding for 8. The flow switching unit has a signal output which is proportional to the actually switched level where the output voltage is approximately 160 mV per level. Each switching unit has an individual output signal which is important to know for the postprocessing software.

If the LI6251 is used then the CO_2 signal, the temperature signal and the pressure signal are measured, in case of using the LI6262 the linearized CO_2 and H_2O signals are measured and in case of using a GMP343 probe the linearized CO_2 signal is measured. In all three cases also the level signal of the flow switching unit is registered by the datalogger.

6.1 The Control Files

CSLOGCO2 uses different control files for the three different gas analyzers.

6.1.1 Control File for an LI6251

If an LI6251 is used control files of the following type must be used:

```
# Control file for CO2-profile
# with LI6251
[Number of levels]
5
[Level voltage step (mV)]
159.9
[Time interval for one cycle (min)]
10
[Missing value]
-9999.
[Header line]
"Date Time","CO2, 1m (ppm)","CO2, 2m (ppm)","CO2, 5m (ppm)","CO2, 10m (ppm)","CO2, 20m (ppm)"
[Calibration temperature (degC)]
35.4
[Coefficients for CO2 − signal (a1, a2, a3)]
.1386
1.806E-05
2.858E-09
[Coefficients for pressure − signal (a0, a1)]
58.296
0.01529
```

In the first section the number of levels (inlets) is defined. The entry for the next section is the level output voltage step in millivolt which is around 160 mV but differs slightly from switching unit to switching unit. In the next section the time interval for one full cycle i.e. the period for switching through all levels is defined (this is usually 10 minutes). The following section contains the header line which should be written to the output file. The items in the header line must be included in double quotes and separated by commas. As next the value for missing data or data out of range is defined. The last three sections contain the calibration coefficients of the specific gas analyzer.

A typical logger data file looks as follows:

```
108,2000,159,1112,159.9,2035,2554,2729
113,2000,159,1114,319.5,1996,2524,2731
118,2000,159,1116,479.3,1998,2502,2727
123,2000,159,1118,639.3,1995,2479,2728
```

```
128,2000,159,1120,799,1964,2457,2727
108,2000,159,1122,159.9,2064,2336,2728
113,2000,159,1124,319.5,1992,2305,2728
118,2000,159,1126,479.2,2011,2285,2725
123,2000,159,1128,639.3,2011,2266,2723
128,2000,159,1130,799,1992,2247,2727
108,2000,159,1132,159.8,2094,2149,2726
113,2000,159,1134,319.5,2024,2125,2729
118,2000,159,1136,479,2011,2109,2729
123,2000,159,1138,639.3,2002,2093,2727
128,2000,159,1140,799,1993,2079,2729
```

The first item is the output array ID which is automatically generated by the logger. The 2^{nd} to 4^{th} item are year, day of year and time (hhmm) which **must** be programmed to be output. The remaining 4 items are level voltage, CO_2 signal voltage, temperature signal voltage and pressure signal voltage. Five consecutive lines comply with one complete cycle (5 inlet levels). The conversion program combines those five lines to one output line containing five CO_2 concentration values as shown below:

```
"Date Time","CO2, 1m (ppm)","CO2, 2m (ppm)","CO2, 5m (ppm)","CO2, 10m (ppm)","CO2, 20m (ppm)"
07.06.2000 11:20,  375.01,  365.01,  365.52,  364.44,  356.80
07.06.2000 11:30,  379.42,  361.59,  366.13,  366.04,  360.96
07.06.2000 11:40,  384.52,  366.97,  363.66,  361.46,  359.00
```

6.1.2 Control File for an LI6262

If an LI6262 is used control files of the following type must be used:

```
# Control file for CO2-and H2O-profile
# with LI6262
[Number of levels]
3
[Level voltage step (mV)]
159.5
[Time interval for one cycle (min)]
10
[Missing value]
-9999.
[Header line]
"Date Time","CO2, 1m (ppm)","CO2, 2m (ppm)","CO2, 5m (ppm)","H2O, 1m (ppm)","H2O, 2m (ppm)","H2O, 5m (ppm)"
[Minimum and maximum for CO2 – signal]
300.
600.
[Minimum and maximum for H2O – signal]
0.
30.
```

If an LI6262 is used the first five sections of the control file are the same as for an LI6251. It must be pointed out that in each line of the output file firstly the CO_2 concentration values and then the H_2O concentration values are written. The next section contains the the CO_2 concentration values corresponding to 0 V and 5 V output which are set at the gas analyzer. The last section contains the the H_2O concentration values corresponding to 0 V and 5 V output which are set at the gas analyzer.

A typical logger data file looks as follows:

```
108,2000,142,1003,159.7,1061,840
113,2000,142,1006,319.3,1051,868
118,2000,142,1010,479.3,1048,903
108,2000,142,1013,159.6,1066,835
113,2000,142,1016,319.3,1058,863
118,2000,142,1020,479.3,1054,880
108,2000,142,1023,159.6,1058,887
113,2000,142,1026,319.3,1058,873
118,2000,142,1030,479.3,1056,885
108,2000,142,1033,159.6,1056,895
113,2000,142,1036,319.3,1165,881
118,2000,142,1040,479.3,1053,893
108,2000,142,1043,159.7,1060,880
113,2000,142,1046,319.3,1056,903
118,2000,142,1050,479.3,1056,874
```

The first item is the output array ID which is automatically generated by the logger. The 2^{nd} to 4^{th} item are year, day of year and time (hhmm) which **must** be programmed to be output. The remaining 3 items are level voltage, CO_2 signal voltage and H_2O signal voltage. Three consecutive lines comply with one complete cycle (3 inlet levels). The conversion program combines those three lines to one output line containing three CO_2 concentration values and three H_2O concentration values as shown below:

```
"Date Time","CO2, 1m (ppm)","CO2, 2m (ppm)","CO2, 5m (ppm)","H2O, 1m (ppm)","H2O, 2m (ppm)","H2O, 5m (ppm)"
21.05.2000 10:10,  363.66,  363.06,  362.88,    5.04,    5.21,    5.42
21.05.2000 10:20,  363.96,  363.48,  363.24,    5.01,    5.18,    5.28
21.05.2000 10:30,  363.48,  363.48,  363.36,    5.32,    5.24,    5.31
21.05.2000 10:40,  363.36,  369.90,  363.18,    5.37,    5.29,    5.36
21.05.2000 10:50,  363.60,  363.36,  363.36,    5.28,    5.42,    5.24
```

6.1.3 Control File for a GMP343 probe

If a GMP343 is used control files of the following type must be used:

```
# Control file for CO2-profile
# with GMP343
[Number of levels]
6
[Level voltage step (mV)]
167.0
[Time interval for one cycle (min)]
10
[Missing value]
-9999.
[Header line]
"Date Time","CO2, 1m (ppm)","CO2, 2m (ppm)","CO2, 5m (ppm)","H2O, 1m (ppm)","H2O, 2m (ppm)","H2O, 5m (ppm)"
[Span for CO2 – signal (mV)]
5000.
[Minimum and maximum for CO2 – concentration (ppm)]
0.
1000.
```

If a GMP343 is used the first five sections of the control file are the same as for the other two gas analyzers. The next section contains the maximum output voltage which can be configured in the carbon dioxide probe (the minimum output signal is always zero volt). The last section contains the the CO_2 concentration values corresponding to 0 V and to the maximum output signal defined above which can again be configured in the probe.

A typical logger data file looks as follows:

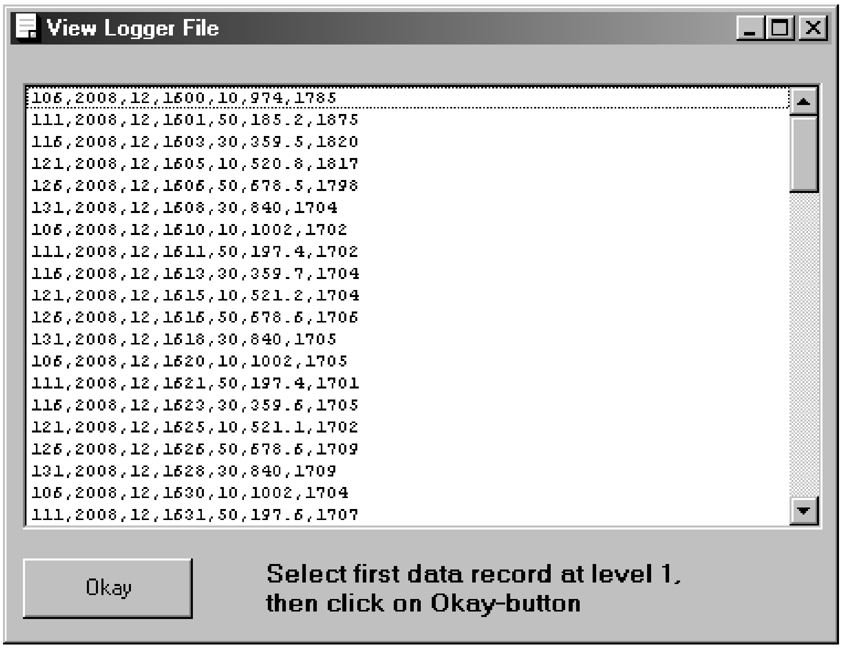
```
111,2007,290,1601,50,196,1934
116,2007,290,1603,30,358.3,1929
121,2007,290,1605,10,518.9,1898
126,2007,290,1606,50,677.1,1900
131,2007,290,1608,30,839,1886
106,2007,290,1610,10,999,1904
111,2007,290,1611,50,196.2,1917
116,2007,290,1613,30,358.5,1912
121,2007,290,1615,10,518.8,1895
126,2007,290,1616,50,677.2,1898
131,2007,290,1618,30,839,1891
106,2007,290,1620,10,999,1907
111,2007,290,1621,50,196.2,1933
116,2007,290,1623,30,358.4,1932
121,2007,290,1625,10,518.9,1897
```

The first item is the output array ID which is automatically generated by the logger. The 2^{nd} to 5^{th} item are year, day of year, time (hhmm) and seconds. If the level-intervals always match full minutes then the output of the seconds may be omitted. The remaining 2 items are level voltage and CO_2 signal voltage. Six consecutive lines comply with one complete cycle (6 inlet levels). The conversion program combines those three lines to one output line containing six CO_2 concentration values values as shown below:

```
"Date Time","CO2, 0.1m (ppm)","CO2, 0.3m (ppm)","CO2, 1.0m (ppm)","CO2, 2.0m (ppm)","CO2, 8.0m (ppm)","CO2, 16.0m (ppm)"
20.09.2007 16:50,  389.00,  392.00,  387.80,  388.40,  388.00,  385.20
20.09.2007 17:00,  391.20,  390.80,  387.40,  387.60,  386.80,  385.80
20.09.2007 17:10,  388.40,  390.00,  388.40,  387.80,  388.60,  386.20
20.09.2007 17:20,  391.00,  391.40,  389.20,  389.40,  388.40,  385.60
20.09.2007 17:30,  391.40,  394.00,  389.60,  388.00,  388.20,  386.40
```

6.1.4 Faulty level voltages in raw data files

If for any reason the level voltage from the valve switching unit is corrupt so that the program would not be able to identify the correct levels then a special option in the control files may be used to solve the problem so that the data files still can be processed. If it can be assumed that the data from the different levels appear in the correct order then it is possible to set the level voltage step in the control file to a value less or equal zero. In this case the program opens a window showing the first 100 lines of data from the raw data file:



```
106,2008,12,1600,10,974,1785
111,2008,12,1601,50,185.2,1875
116,2008,12,1603,30,359.5,1820
121,2008,12,1605,10,520.8,1817
126,2008,12,1606,50,678.5,1798
131,2008,12,1608,30,840,1704
106,2008,12,1610,10,1002,1702
111,2008,12,1611,50,197.4,1702
116,2008,12,1613,30,359.7,1704
121,2008,12,1615,10,521.2,1704
126,2008,12,1616,50,678.6,1706
131,2008,12,1618,30,840,1705
106,2008,12,1620,10,1002,1705
111,2008,12,1621,50,197.4,1701
116,2008,12,1623,30,359.6,1705
121,2008,12,1625,10,521.1,1702
126,2008,12,1626,50,678.6,1709
131,2008,12,1628,30,840,1709
106,2008,12,1630,10,1002,1704
111,2008,12,1631,50,197.6,1707
```

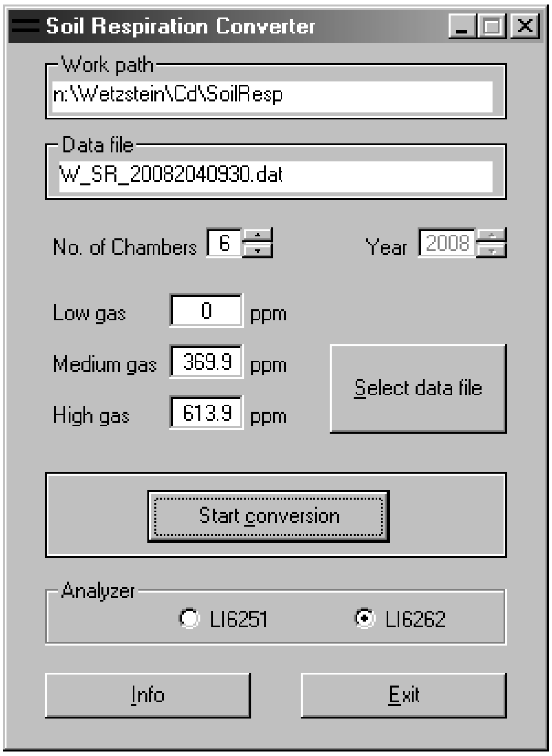
Select first data record at level 1,
then click on Okay-button

Figure 9: The CSLOGCO2 file preview window

Now the line containing data from the first level must be selected. All following lines should be in the correct order. If there would be any disruption in the data file from where on the order of the levels is not any more correct, then the output result will be wrong from that data record on. In this case it would be necessary to split the raw data file into two or more pieces.

7 The soil respiration data converter **SoilResp**

The soil respiration data converter can only be used to convert data from the automated soil respiration system built at the MPI-BGC. This system was running the first few years with a CR10X datalogger which was exchanged against a CR1000 in late 2007. The CR1000 creates two data tables where one contains the CO_2 concentration data from the LI6262 at a rate of 2 seconds. The other table contains chamber temperatures, soil temperatures, soil moisture data, wind data and data from two mass flow controllers as 10-minute averages. This conversion program needs both tables to process the data. The file name of the fast respiration data must contain the string ”_SR_” and the file name of the averaged soil property data must be identical but ”_SR_” must be replaced by ”_SP_”.



Work path: n:\Wetzstein\Cd\SoilResp

Data file: W_SR_20082040930.dat

No. of Chambers: 6

Year: 2008

Low gas: 0 ppm

Medium gas: 369.9 ppm

High gas: 613.9 ppm

Analyzer: ○ LI6251 ● LI6262

Figure 10: The SoilResp dialog window

The number of active soil respiration chambers and the concentration values of the calibration gases must be entered in the dialog window, as well as the type of gas analyser being used.

The program generates three output files named as the input file extended with the strings ”_Resp1”, ”_Resp2” and ”_Resp3”, respectively. The ”_Resp1”-file is only a temporary file which will be deleted after all processing steps have finished.

The ”_Resp2” file will be kept and looks as follws:

```
07.11.2007 11:00, 6 , 51.14694 , 180 , 927.46 , 3.53 , 0.9909030
07.11.2007 12:00, 1 , 54.23859 , 180 , 926.01 , 3.58 , 0.9962588
07.11.2007 12:00, 2 , 19.17682 , 180 , 924.77 , 3.54 , 0.9766823
07.11.2007 12:00, 3 , 8.255951 , 180 , 925.44 , 3.55 , 0.5591485
07.11.2007 12:00, 4 , 36.26361 , 180 , 925.91 , 3.57 , 0.9865864
07.11.2007 12:00, 5 , 30.80316 , 180 , 925.55 , 3.57 , 0.9867820
```

The time stamp is followed by the chamber number, the CO_2 difference in ppm, the time period in seconds during which the CO_2 difference was built up, the pressure in hPa, the temperature in ^{o}C and the regression coefficient for the linear regression of the CO_2 increase.

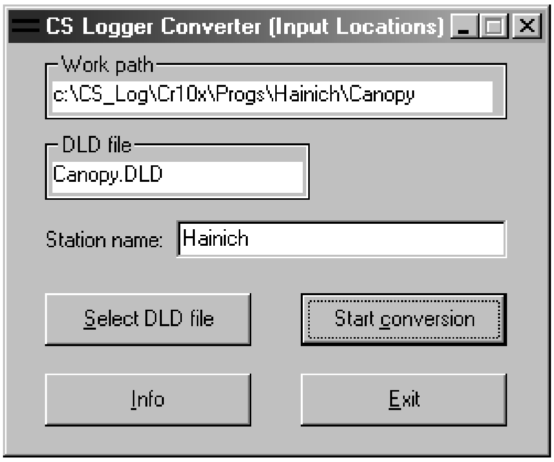
The "_Resp3" file which represents the final results looks as follws:

```
Date Time, Resp1, Regc1, Resp2, Regc2, Resp3, Regc3, Resp4, Regc4, Resp5, Regc5, Resp6, Regc6
07.11.2007 17:00, 1.03, 0.9952, 0.51, 0.9860, 0.08, 0.4803, 0.98, 0.9853, 0.71, 0.9896, 1.22, 0.9876
07.11.2007 18:00, 1.25, 0.9917, 0.59, 0.9792, 0.30, 0.6170, 0.89, 0.9811, 0.78, 0.9920, 1.11, 0.9829
07.11.2007 19:00, 1.22, 0.9975, 0.54, 0.9819, 0.32, 0.4255, 1.06, 0.9834, 0.89, 0.9909, 1.16, 0.9856
07.11.2007 20:00, 1.27, 0.9950, 0.57, 0.9562, 0.27, 0.3812, 1.00, 0.9861, 0.81, 0.9865, 1.57, 0.9966
07.11.2007 21:00, 1.51, 0.9908, 0.68, 0.9965, 0.33, 0.6243, 0.62, 0.9661, 0.78, 0.9872, 1.60, 0.9888
```

Since one cycle for all 6 respiration chambers lasts one hour the final output file contains hourly records of respiration values in $\mathrm{mmol\,m^{-2}\,s^{-1}}$ (columns Respx) together with the regression coefficient for the linear regression of the CO_2 increase (columns Regcx). Values close to 1 for Regcx indicate a good linear increase of CO_2 during the first three minutes after the chamber was closed. Lower values of Regcx indicate that the chamber lid might not close thight onto the chamber frame.

8 Creating input location lists with **ConvertDLD**

This small program allows to create a list of assigned input locations from any DLD-file which represents a logger program for a CR510, CR10X or CR23X logger. Such a list is useful for instance for documentation purposes etc. After selecting a DLD-file and specifying a station name in the dialog window as shown below, the conversion can be started by clicking on the button.



Work path: c:\CS_Log\Cr10x\Progs\Hainich\Canopy

DLD file: Canopy.DLD

Station name: Hainich

Figure 11: The dialog window to create input location lists

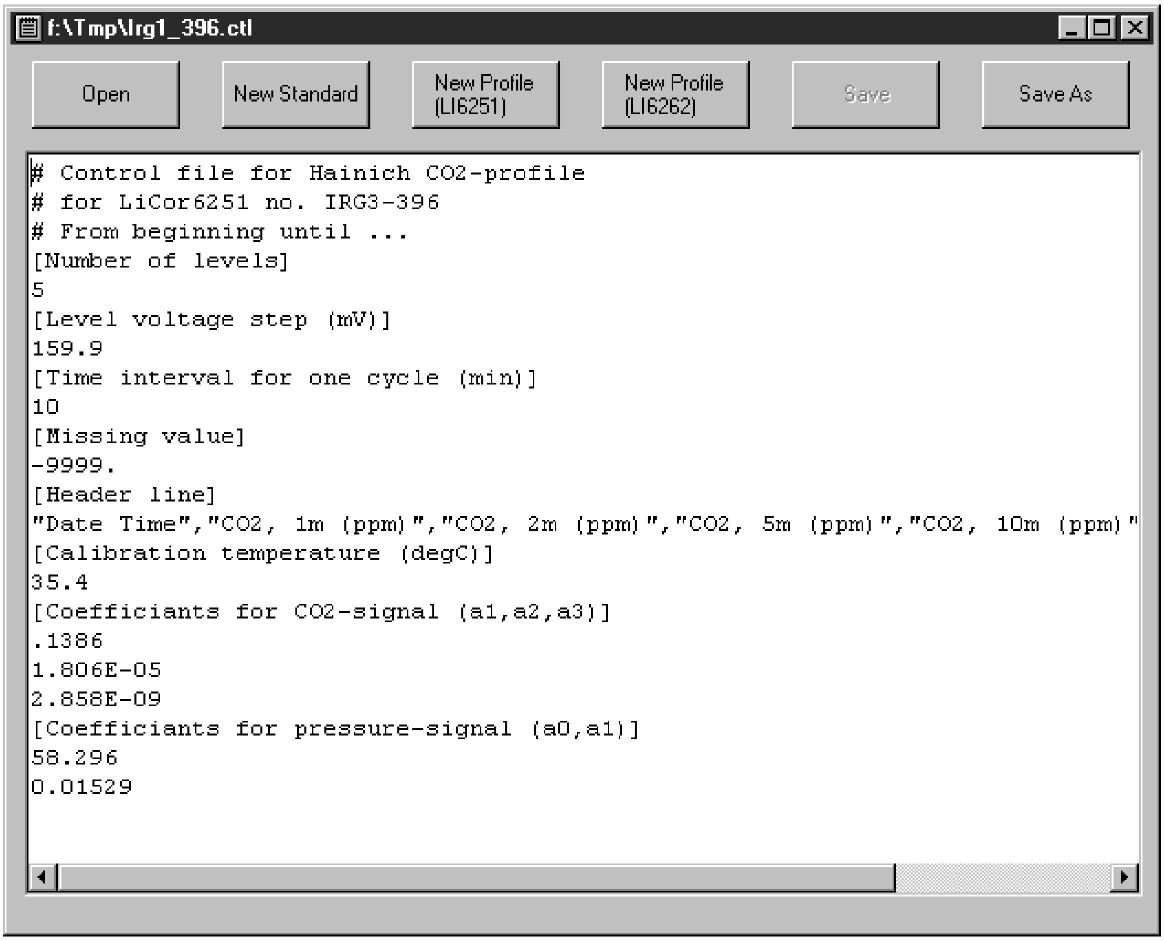
A file like set forth hereunder will then be created within the same folder as the DLD-file with the same name but the extension txt. All information in this file are taken from the DLD-file except the station name which was entered in the form above.

```
Station:    Hainich Canopy
Datalogger: CR10X
Program:    CANOPY.dld

Loc001:  RS_R0
Loc002:  AT1
Loc003:  AT2
Loc004:  AT3
Loc005:  AT4
Loc006:  AT5
Loc007:  STT01
Loc008:  STT02
Loc009:  STT03
Loc010:  STT04
Loc011:  STT05
Loc012:  STT06
Loc013:  STT07
Loc014:  STT08
Loc015:  STT09
Loc016:  STT10
Loc017:  STT11
Loc018:  STT12
Loc019:  swdrc
```

9 Editing Control Files with **Edit_Ctl**

If from the LOGTOOLS program the Edit Control-File button is selected then a simple text editor is started which allows to edit control files. Existing files may be loaded or new control files may be created. If one of the three New.... buttons is selected then the corresponding frame of such a control file containing the section headers is put into the text window. The desired parameters, text lines or numbers can then be filled in.



```
# Control file for Hainich CO2-profile
# for LiCor6251 no. IRG3-396
# From beginning until ...
[Number of levels]
5
[Level voltage step (mV)]
159.9
[Time interval for one cycle (min)]
10
[Missing value]
-9999.
[Header line]
"Date Time","CO2, 1m (ppm)","CO2, 2m (ppm)","CO2, 5m (ppm)","CO2, 10m (ppm)"
[Calibration temperature (degC)]
35.4
[Coefficiants for CO2-signal (a1,a2,a3)]
.1386
1.806E-05
2.858E-09
[Coefficiants for pressure-signal (a0,a1)]
58.296
0.01529
```

Figure 12: The control file edit window

10 Modem Connection to Campbell Dataloggers with CSModem

This program is used to establish a modem connection to a CAMPBELL datalogger which itself is connected to a modem. In the Communication settings and informations frame the correct baud rate and telephone number must be filled in. If it is necessary to dial a zero before the number e.g. from an office then this option must be checked.
In the same directory as the program file there must reside the file Csmodem.ini which contains in the first line the number of stations which properties are defined in a separate file. The second line contains either 0 or 1 where 0 means not to dial a zero before the telephone number and 1 means to dial the zero before the telephone number. The third line contains the number of the communications port to be used to establish the connection. Below is an example for this ini-file:

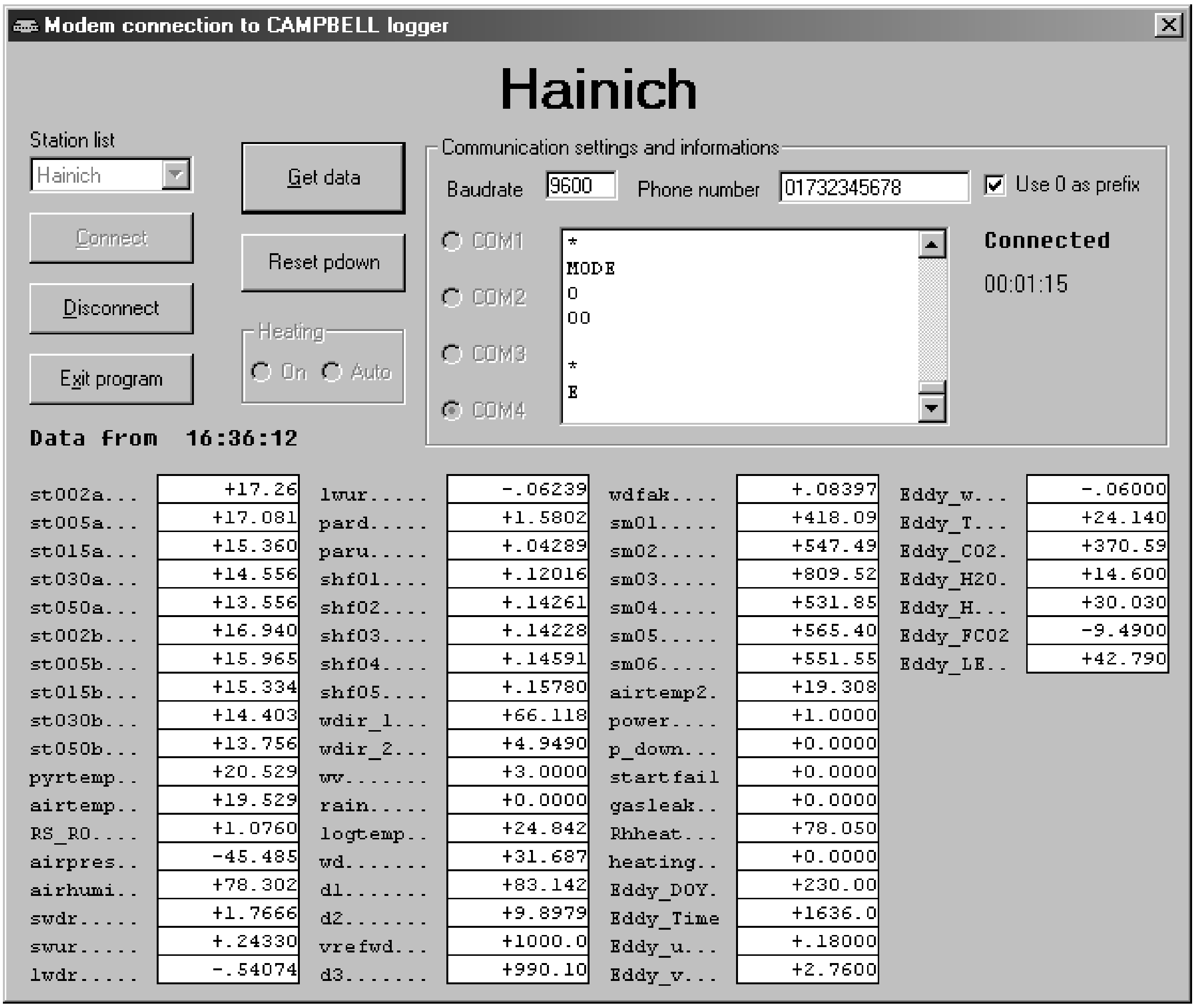
```
7
1
4
```

The station properties file (stations.cfg) must as well reside in the same directory as the program file. For each stations it contains three sections which are

1. the name of the station or location and a number coding for the type of datalogger (1 stands for a CR10X or CR510, 2 stands for CR23X),
2. the telephone number,
3. the number of logger input locations which should be displayed and a number coding for the instrument heating status detection (0 stands for no heating status detection, 1 stands to perform heating status detection. In the latter case the program automatically checks for the status of flag 5 of the logger as soon as the connection is established: if flag 5 is set then the instrument heating is switched on manually, if flag 5 is not set then the heating is in the automatic mode controlled by values of environmental variables as defined in the logger program). If a CR10X or a CR510 is used at a particular station then the appropriate number of additional lines must follow containing the individual names of the input locations because only the CR23X transmits these names.

Following is an example station properties files, where the last 17 lines define the names of the variables within the input locations. For a CR23X these lines would be omitted:

```
[station name]
Biodiversitt, 2
[phone number]
01731234567
[input locations]
45, 0
[station name]
Hainich, 2
[phone number]
01732345678
[input locations]
61, 0
[station name]
Gebesee, 2
[phone number]
01733456789
[input locations]
67, 0
[station name]
Wetzstein, 2
[phone number]
01734567890
[input locations]
66, 1
[station name]
Leinefelde, 2
[phone number]
01731456789
[input locations]
60, 0
[station name]
Mehrstedt 1, 2
[phone number]
01732567891
[input locations]
62, 0
[station name]
Mehrstedt 2, 1
[phone number]
01733214567
[input locations]
17, 0
airtemp
airhumi
surtemp
logtemp
power
p_down
Eddy_DOY
Eddy_Time
Eddy_u
Eddy_v
Eddy_w
Eddy_T
Eddy_CO2
Eddy_H2O
Eddy_H
Eddy_FCO2
Eddy_LE
```



Hainich

Station list: Hainich

Baudrate: 9600

Phone number: 01732345678

☑ Use 0 as prefix

● COM4

```
*
MODE
0
00

*
E
```

Connected

00:01:15

Data from 16:36:12

st002a...	+17.26	lwur.....	-.06239	wdfak....	+.08397	Eddy_w...	-.06000
st005a...	+17.081	pard.....	+1.5802	sm01.....	+418.09	Eddy_T...	+24.140
st015a...	+15.360	paru.....	+.04289	sm02.....	+547.49	Eddy_CO2.	+370.59
st030a...	+14.556	shf01....	+.12016	sm03.....	+809.52	Eddy_H2O.	+14.600
st050a...	+13.556	shf02....	+.14261	sm04.....	+531.85	Eddy_H...	+30.030
st002b...	+16.940	shf03....	+.14228	sm05.....	+565.40	Eddy_FCO2	-9.4900
st005b...	+15.965	shf04....	+.14591	sm06.....	+551.55	Eddy_LE..	+42.790
st015b...	+15.334	shf05....	+.15780	airtemp2.	+19.308		
st030b...	+14.403	wdir_1...	+66.118	power....	+1.0000		
st050b...	+13.756	wdir_2...	+4.9490	p_down...	+0.0000		
pyrtemp..	+20.529	wv.......	+3.0000	startfail	+0.0000		
airtemp..	+19.529	rain.....	+0.0000	gasleak..	+0.0000		
RS_RO....	+1.0760	logtemp..	+24.842	Rhheat...	+78.050		
airpres..	-45.485	wd.......	+31.687	heating..	+0.0000		
airhumi..	+78.302	d1.......	+83.142	Eddy_DOY.	+230.00		
swdr.....	+1.7666	d2.......	+9.8979	Eddy_Time	+1636.0		
swur.....	+.24330	vrefwd...	+1000.0	Eddy_u...	+.18000		
lwdr.....	-.54074	d3.......	+990.10	Eddy_v...	+2.7600		

Figure 13: The modem connection window

The settings for different stations are loaded by selecting the station from the Station list. Pressing the Connect button initiates dialing and connecting to the remote modem. After the connection is established the Get data button can be used to receive the time and the actual values of the defined number of input locations from the datalogger. The power down counter p_down can be reset to zero by pressing the Reset pdown button (this forces flag 8 to be set in which case the logger resets the p_down variable. If the heating status detection has been performed directly after successfully connecting to the logger, then the actual instrument heating operating mode is shown and the Heating can be switched on manually or can be set to the automatic mode where the actual values of environmental variables as programmed in the logger are controlling the instrument heating.

11 Variables their Symbols and Units

In the following table all possible variables of datalogger files are listed together with their units and symbols.

Symbol	**Unit**	**Variable**
p	hPa	air pressure
T	^{o}C	air temperature
$T1, T2$	^{o}C	air temperatures of Bowen-Ratio-System
$T2$	^{o}C	air temperature at 2 m height
T_xxm	^{o}C	air temperature at xx m height
dt	K	air temperature difference of Bowen-Ratio-System
$Tpot$	K	potential temperature
$Tdew$	^{o}C	dew point temperature
rh	%	relative humidity
$rh1, rh2$	%	relative humidities of Bowen-Ratio-System
$VPmax$	mbar	saturation water vapor pressure
$VPact$	mbar	actual water vapor pressure
$VPdef$	mbar	water vapor pressure deficit
sh	$\mathrm{g\,kg^{-1}}$	specific humidity
$sh1, sh2$	$\mathrm{g\,kg^{-1}}$	specific humidities of Bowen-Ratio-System
dsh	$\mathrm{g\,kg^{-1}}$	specific humidity difference of Bowen-Ratio-System
H2OC	$\mathrm{mmol\,mol^{-1}}$	water vapor concentration
rho	$\mathrm{g\,m^{-3}}$	air density
wv	$\mathrm{m\,s^{-1}}$	wind velocity from cup anemometer
wv_sonic	$\mathrm{m\,s^{-1}}$	wind velocity from 2D sonic anemometer
wd	deg	wind direction from wind vane
wd_sonic	deg	wind direction from 2D sonic anemometer
$rain$	mm	precipitation
$rain, tot$	mm	total precipitation without interception losses
$rain, for$	mm	precipitation at forest floor
$Tpyr$	^{o}C	temperature of net radiometer
TDR	$\mathrm{W\,m^{-2}}$	total downward radiation
TUR	$\mathrm{W\,m^{-2}}$	total upward radiation
SWDR	$\mathrm{W\,m^{-2}}$	short wave downward radiation
SWUR	$\mathrm{W\,m^{-2}}$	short wave upward radiation
SWDR	$\mathrm{W\,m^{-2}}$	short wave downward radiation
SWDRC	$\mathrm{W\,m^{-2}}$	short wave downward radiation below canopy
SWUR_G	$\mathrm{W\,m^{-2}}$	short wave upward radiation below canopy

Albedo	%	albedo ($SWUR/SWDR$)
LWDR	W m^{-2}	long wave downward radiation
LWUR	W m^{-2}	long wave upward radiation
$Trad$	^{o}C	radiation temperature from *LWUR*
$T0$	^{o}C	surface temperature from radiation thermometer
DDR	W m^{-2}	diffuse (short wave) radiation
SDUR	s	sunshine duration
SDIR	W m^{-2}	direct sun radiation
PAR	μmol m^{-2} s^{-1}	photosynthetically active radiation
PARD	μmol m^{-2} s^{-1}	downward photosynthetically active radiation
PARU	μmol m^{-2} s^{-1}	upward photosynthetically active radiation
PAR_G	μmol m^{-2} s^{-1}	photosynthetically active radiation below canopy
Rn	W m^{-2}	net radiation
SHFn	W m^{-2}	soil heat flux from sensor n
SHFM	W m^{-2}	mean soil heat flux of all sensors
SMnn	Vol. %	soil moisture from sensor nn
SMxxx	Vol. %	soil moisture at depth xxx (xxx in cm)
SMxxxa	Vol. %	soil moisture from profile a at depth xxx (xxx in cm)
SMxxxb	Vol. %	soil moisture from profile b at depth xxx (xxx in cm)
SMxxxc	Vol. %	soil moisture from profile c at depth xxx (xxx in cm)
STxxx	^{o}C	soil temperature at depth xxx (xxx in cm)
STxxxa	^{o}C	soil temperature from profile a at depth xxx (xxx in cm)
STxxxb	^{o}C	soil temperature from profile b at depth xxx (xxx in cm)
STxxxM	^{o}C	mean soil temperature at depth xxx (xxx in cm)
STTnn	^{o}C	stem temperature from sensor nn
WTn	^{o}C	water temperature of sensor n
Tlog	^{o}C	internal logger temperature
Vref1	mV	voltage of current source #1 for temperature sensors
Vref2	mV	voltage of current source #2 for temperature sensors
Heating	%	percentage of time with instrument heating switched on
Cal	samp	number of samples with Bowen-Ratio-System in calibration position
IBat	A	battery current (+ = charge, − = decharge)
UBat	V	battery voltage
CO2, z	ppm	CO_2-concentration at height z
H2O, z	mmol mol^{-1}	H_2O-concentration at height z