

CH-CHA Chamau eddy flux raw data issues September/October 2016

2016-10-28 | Werner Eugster

From 2 September 2016 to 27 October 2016 a mismatch between QCL data acquisition (qclread) and Sonic data acquisition program (sonicreadHS) was in place which produced the following issue. The qclread program was sending only five variables to sonicreadHS, but sonicreadHS – which by now was the same version as in Davos – expected eight variables.

All raw data files have a header in which a version number and a variant number of this extension to standard eddy flux measurements is defined. In addition, the number of bytes in a record make it possible to know which data recording format was used. However, there is no bi-directional interaction between sonicreadHS and the programs that should send data to it. Thus if a supplier program (qclread) does not send the data in the format that sonicreadHS expects, then the data are stored in a way that differs from correct conditions.

In this specific case we have two types of data files:

Extension 1 Variant 6 Bytes 22 – this is the old format (sonicreadHS 7.06, qclread 2.00 to 2.02)

Extension 1 Variant 6 Bytes 34 – this is the new format (sonicreadHS 7.07, qclread 2.03)

The issue during the named period at Chamau was that qclread 2.00 was run in combination with sonicreadHS 7.07. There were two options to solve this: (a) change qclread to version 2.03 and record the additional information as in Davos; or (b) go back do sonicreadHS 7.06 to be compatible with the version of qclread 2.00 that was installed on the new Chamau computer. I decided to go for option (b) now. There is also an option (c) to define the expected data format in the source code of sonicreadHS 7.07 (in file extdata.h):

```
#define AERODYNE_INSTRUMENT          QC_TILDAS_ICOS_DAVOS
```

is the current setting, which could be changed to

```
#define AERODYNE_INSTRUMENT          QC_TILDAS_ETHZ_T_P
```

if one wants to combine the newest version of sonicreadHS with an older version of qclread.

In the meantime the files with Extension 1 Variant 6 Bytes 34 need some special treatment to put the cell pressure and temperature variables in the correct column and convert them accordingly. The qclread 2.03 definition for sending the QCL data to sonicreadHS is defined in this way:

```
sprintf(msg.mdata,"QCL CH4 %.1f N2O %.1f H2O %.1f CO2 %.1f TEMP %.1f PRESS %.1f Status %ld Vici %ld", ...
```

In the older qclread 2.00 version running at Chamau it is defined as:

```
sprintf(msg.mdata,"QCL CH4 %.1f N2O %.1f H2O %.1f TEMP %.1f PRESS %.1f", ...
```

Thus, if there is a mismatch between sender and receiver, we get the issue that we observed at Chamau between September and October 2016.

This is a data snippet after we solved the issue on 27 October (only QCL part converted from binary to ASCII):

```
QCL= 22 v 6 000    1971.44    323.441    9849630    296.276    34.991
QCL= 22 v 6 001    1971.44    323.441    9849630    296.276    34.991
```

```

QCL= 22 v 6 000 1972.80 323.310 9852080 296.266 34.984
QCL= 22 v 6 001 1972.80 323.310 9852080 296.266 34.984
QCL= 22 v 6 000 1971.11 323.279 9808710 296.266 34.984
QCL= 22 v 6 001 1971.11 323.279 9808710 296.266 34.984
QCL= 22 v 6 000 1972.01 323.814 9856260 296.275 34.972
QCL= 22 v 6 001 1972.01 323.814 9856260 296.275 34.972
QCL= 22 v 6 000 1972.02 323.769 9824840 296.275 34.972
QCL= 22 v 6 001 1972.02 323.769 9824840 296.275 34.972
QCL= 22 v 6 000 1973.21 323.639 9850610 296.266 34.966
QCL= 22 v 6 000 1970.89 323.998 9819560 296.266 34.966

```

and this is a similar data snipped from before (17 October):

```

QCL= 34 v 6 000 2401.98 330.777 0.098 9641730 0.000 0.000 154154545 926484788
QCL= 34 v 6 001 2401.98 330.777 0.098 9641730 0.000 0.000 154154545 926484788
QCL= 34 v 6 000 2392.78 331.121 0.098 9681950 0.000 0.000 154284083 926484788
QCL= 34 v 6 001 2392.78 331.121 0.098 9681950 0.000 0.000 154284083 926484788
QCL= 34 v 6 000 2401.33 330.580 0.098 9641920 0.000 0.000 154154547 926484788
QCL= 34 v 6 001 2401.33 330.580 0.098 9641920 0.000 0.000 154154547 926484788
QCL= 34 v 6 000 2394.59 330.568 0.098 9693220 0.000 0.000 154483001 926484788
QCL= 34 v 6 001 2394.59 330.568 0.098 9693220 0.000 0.000 154483001 926484788
QCL= 34 v 6 000 2401.03 330.247 0.098 9633820 0.000 0.000 154481206 926484788
QCL= 34 v 6 001 2401.03 330.247 0.098 9633820 0.000 0.000 154481206 926484788
QCL= 34 v 6 000 2398.24 331.089 0.098 9654980 0.000 0.000 154415408 926484788
QCL= 34 v 6 001 2398.24 331.089 0.098 9654980 0.000 0.000 154415408 926484788
QCL= 34 v 6 000 2400.61 330.239 0.098 9632760 0.000 0.000 154417460 926484788
QCL= 34 v 6 001 2400.61 330.239 0.098 9632760 0.000 0.000 154417460 926484788

```

Thus:

- The CH4 concentration is stored correctly (first data column from qclread)
- The N2O concentration is stored correctly (second data column from qclread)
- The H2O are saved in third position in the raw data file, but in ethconvert the decision was to output the CO2 concentration BEFORE the H2O concentration (although the QCL sends it AFTER H2O, see the source code snippet above). Hence H2O concentrations in the ethconvert output are in the fourth column (not the third one) for 1/6/34 data. But they are correctly saved and correctly converted.
- The third column (CO2) is an arbitrary value (hence it does not vary)
- The fifth column (cell temperature, K) is empty, which is not as it should be
- The sixth column (cell pressure, Torr) is empty, which is not as it should be
- The seventh column (QCL status) contains information, although the Chamau QCL was NOT configured to send any information about its status. So most likely these data have a different meaning.
- The eighth column shows the same symptoms

Thus, some further investigations were required.

This is how qclread 2.00 sends the data to sonicreadHS:

```
QCL CH4 2279460.0 N2O 325972.0 H2O 9300630.0 TEMP 296182.0 PRESS 34247.0
```

sonicreadHS 7.07 reads incoming data using the following patterns depending on the definition of AERODYNE_INSTRUMENT (in the source code):

```

switch(AERODYNE_INSTRUMENT){
  case(QC_TILDAS_ETHZ_T_P) :
    sscanf(qclrecord,"%*3s %*3s %f %*3s %f %*3s %f %*3s %f %*3s %f",
      &ch4,&n2o,&h2o,&qT,&qP);
    break;
  case(QC_TILDAS_ICO5_DAVOS) :
    sscanf(qclrecord,"%*3s %*3s %f %*3s %f %*3s %f %*3s %f %*3s %f %*3s %f %*3s %f %*3s %f %*3s %f %*3s %f %*3s %f %*3s %f",
      &ch4,&n2o,&h2o,&co2,&qT,&qP,&statusW,&vici);
    break;
  default :
    fprintf(stderr,"For -e1.6 you need to hard-code the data format\n");
    exit(ERROR);
}

```

Thus, the string sent by qclread 2.00 is interpreted as this:

```
QCL CH4 2279460.0 N20 325972.0 H20 9300630.0 TEMP 296182.0 PRESS 34247.0
^^^
%*3s
^^^
%*3s
^^^^^^^^
%f for ch4
^^^
%*3s
^^^^^^^^
%f for n2o
^^^
%*3s
^^^^^^^^
%f for h2o
```

Until here everything is correct, but it is unclear what happens with the remaining information since the read pattern expects a 3-character information (for CO2) where we supply a 4-character information (TEMP). For this I need to inspect the raw binary data. I use the line that looks like this in the ethconvert output:

```
QCL= 34 v 6 000 2404.97 330.618 0.098 9642480 0.000 0.000 154677554 926484788
```

which in binary format (hex) is:

```
22600024B26A00050B7A00000062009321F0000000000000000938313237390934
```

Split up into the components:

22	length of record (0x22 = 34 bytes)
6	extension variant = 6
0	status octal 0 = 0 (everything fine)
0024B26A	CH4 2404970 = 2404.970 ppb = 2.404970 ppm correct
00050B7A	N20 330618 = 330.618 ppb correct
00000062	CO2 98 arbitrary value
009321F0	H2O 9642480 ppb = 9642.480 ppm = 9.642480 mmol/mol correct
00000000	here should be cell temperature
00000000	here should be cell pressure
09383132	here should be QCL status
37390934	here should be Vici valve status

Best interpretation is the following: since only the first three columns corresponded to the expected data format, only those were converted. For cell temperature and cell pressure a zero was used since the string did not match expectations, and the two last columns are some other values outside the valid data range. C does not make a range checking, thus if we have a variable with 5 components only and try to read 8 values, the three last are read from the adjacent variables in computer memory (and thus no way to say what it is, but most likely it is NOT temperature and pressure).

OK, so final recommendation is the following:

- *use CH4 as is - it's correct*
- *use N20 as is - it's correct*
- *take H2O from the fourth instead of the third column; it's correct*
- *ignore everything else*

(for a long time we did NOT record cell temperature and pressure, hence this should be no important issue after all).