

ReCON Implementation of QARTOD

How the QARTOD tests were programmed to provide quality assurance to the ReCON real-time data collection

Ronald Muzzi, retired NOAA Great Lakes Environmental Research Laboratory

Russ Miller, Cooperative Institute for Great Lakes Research, University of Michigan

Kaitlyn Frank, formerly NOAA Great Lakes Environmental Research Laboratory

Laura Mendoza, formerly FedWriters for NOAA

Steve Ruberg, NOAA Great Lakes Environmental Research Laboratory

March 27, 2026



UNITED STATES
DEPARTMENT OF COMMERCE

Howard W. Lutnick
Secretary

NATIONAL OCEANIC AND
ATMOSPHERIC ADMINISTRATION

Dr. Neil Jacobs
Administrator

TABLE OF CONTENTS

INTRODUCTION	2
1.0 ReCON IMPLEMENTATION OF QARTOD	2
1.1 The QARTOD Flags	2
1.2 The QARTOD Test Types	3
2.0 QARTOD PARAMETERS AND TESTS	4
2.1 Sensor Parameter Files.....	4
2.2 Test Algorithms	7
2.2.1 Data Gap	7
2.2.2 Range Test	7
2.2.3 Climate Test	7
2.2.4 Spike Test	7
2.2.5 Rate of Change Test	8
2.2.6 Flat Line Test	9
2.2.7 Gradient Test	9
2.2.8 Summary Flag	10
2.3 Rationale for Selecting Parameter Values	10
The parameter values used by the above tests are described in this section.....	10
2.3.1 Range Test Values	10
2.3.2 Climatology Test Values	11
2.3.3 Other Test Values	12
2.3.4 Known Failures	12

3.0 SOFTWARE CODING	13
3.1 Rationale for Using the C Compiler	13
3.2 Overall Program Structure	13
3.3 Data Buffer and Calculated Values.....	14
3.4 Parameter File Format.....	16
3.5 Output File	16
4.0 SUMMARY	17
5.0 REFERENCES	17
6.0 SUPPLEMENTARY MATERIALS	18

LIST OF TABLES

Table 1. List of QARTOD flag values and descriptions.	3
Table 2. List of QARTOD test types.	3
Table 3. Description of values in QartodTable.txt file data line.	5
Table 4. Comparison of program execution times.	13

INTRODUCTION

The Realtime Coastal Observation Network (ReCON) collects data throughout the Great Lakes region from various in-situ sensors from the lake bottom to the surface and the immediate atmosphere and disseminates that information in real-time to data centers and stakeholders. The Integrated Ocean Observing System (IOOS), also under the National Oceanic and Atmospheric Administration (NOAA), has implemented a framework of quality control tests that assess the quality of data in real-time by automated processes. These tests are called the Quality Assurance / Quality Control of Real-Time Oceanographic Data or QARTOD. These tests have been used as a starting point for ReCON data quality assessment with some modifications as were deemed appropriate. This document details the ReCON implementation of the algorithms used by each test, the methodology in determining the appropriate parameter values for the tests, and some details of the operation of the software program developed.

1.0 RECON IMPLEMENTATION OF QARTOD

The data from various sensors located on ReCON stations throughout the Great Lakes region are uploaded to a server located in our facilities that automatically processes the data in real-time. The processed data are then distributed to stakeholders via a web interface and also to data centers such as the National Data Buoy Center and the IOOS Great Lakes Observing System regional association Seagull website.

After the data have been collected from the stations, are formatted into a CSV (comma separated value) format, and then evaluated for quality assurance purposes through a compiled C program called QartodCalc.c. This program adds a column to the data with a descriptive heading. This column is a flag value indicating whether or not those data sample passed the quality assurance tests.

1.1 THE QARTOD FLAGS

QARTOD specifies adding flags to each data measurement. The flag is a numeric value that indicates the result of the tests performed on that measurement.

Table 1. List of QARTOD flag values and descriptions.

Flag	Value	Description
Pass	1	Data have passed critical real-time quality control tests and are deemed adequate for use as preliminary data.
Not Evaluated	2	Data have not been QC-tested, or the information on quality are not available.
Suspect or Of High Interest	3	Data are considered to be either suspect or of high interest to data providers and users. They are flagged suspect to draw further attention to them by operators.
Fail	4	Data are considered to have failed one or more critical real-time QC checks. If they are disseminated at all, it should be readily apparent that they are not of acceptable quality.
Missing Data	9	Data are missing; used as a placeholder.

In our implementation, only the Pass (1), Suspect (3), and Fail (4) flags are used. For data that are not evaluated, the QARTOD flag column is not included for those data file. Data that are missing are not included in the data files.

1.2 THE QARTOD TEST TYPES

Instead of implementing a separate series of tests for each sensor type as described by QARTOD, a single standard series of tests has been implemented that can be applied to all sensors.

However, some sensor specific tests are being performed. For example, many of the tests for measuring water currents, such as battery power, sensor tilt, signal-to-noise, correlation magnitude, etc., are already being performed by the water current sensor itself and therefore do not need to be repeated.

Table 2. List of QARTOD test types.

Group 1 <i>Required</i>	Test 1 Test 2 Test 3 Test 4 Test 5	Timing/Gap Test Syntax Test Location Test Gross Range Test Climatology Test
Group 2 <i>Strongly Recommended</i>	Test 6 Test 7 Test 8	Spike Test Rate of Change Test Flat Line Test
Additional <i>Strongly Recommended</i>	Test 20	Gradient Test

The following test types are already implemented in the ReCON data collection infrastructure, so they are not repeated by this QARTOD software program:

The **Timing/Gap Test** determines if the data were received at the expected time to verify if the sensors internal clocks are working as expected, since some of the sensors have their own internal clock to timestamp the data. The ReCON stations use a Network Time Protocol (NTP) to synchronize their clocks. When data are collected, the data timestamp is checked on the station and if they fall outside the expected time then those data are discarded.

The **Syntax Test** determines if the data have the proper structure, such as number of characters in a line, characters in the correct position, numbers are only numeric characters, etc. The ReCON stations perform all these along with additional checks to verify that the syntax and numeric values are correct. It discards the data if it is not.

The **Location Test** determines if the station is within the expected location. This test would not apply to fixed stations, such as lighthouses. It would also not be unique to individual data files, but would apply, for example, to all the data measured on a buoy. For a buoy to drift outside the accepted watch-circle would be a catastrophic problem identified in an automatic warning to the operators. This would then be manually mitigated by the operators instead of through this QARTOD software program.

2.0 QARTOD PARAMETERS AND TESTS

2.1 SENSOR PARAMETER FILES

The parameter values used by the tests are contained in various parameter files. The main file is named **QartodTable.txt**. This file contains the parameter values for each sensor and test in a CSV table format. Comments are indicated by the '#' (pound sign) character at the beginning of the line. Empty lines are ignored and all space and tab characters are removed, so the file can be freely formatted for clarity. Each line in the file includes the parameter values for a specific sensor. The values are separated by commas. The following line is an example and the meaning of each field is explained in table 3:

```
atemp_csi, C, 2m, 1, 1, -50, 55, -40, 45, AirTemp, 1, 3.5, 12,
1, 0.4, 1, 30, 60, 0.005, 0, 0, 0
```

Table 3. Description of values in QartodTable.txt file data line.

Value	Example	Description
Parameter Name	atemp_csi	Standard ReCON parameter name (†)
Units	C	For reference only, not used in the program
Sample Period	2m	Time between data samples, for reference only, not used in the program: s is for seconds m is for minutes h is for hours
Dim	1	Dimension: 1 = single value 2 = column data (e.g., temperature profile at multiple depths) 3 = two column data (e.g., current profile with north/east vector components at multiple depths)
Range Test: Flag	1	1 to apply test, 0 to not apply it
Smin	-50	Sensor minimum valid value
Smax	55	Sensor maximum valid value
Umin	-40	User minimum valid value
Umax	45	User maximum valid value
Climate Test: Flag	AirTemp	Flag value:

		0 = don't apply test AirTemp = Air temperature SrfTemp = Water temperature near surface BtmTemp = Water temperature near bottom ProTemp = Water profile temperature ParBuoy = PAR (in air) SolBuoy = Solar radiation
Spike Test:		
Flag	1	1 to apply test, 0 to not apply it
Tsusp	3.5	Suspect if data change between consecutive values exceeds this amount
Tfail	12	Fail if data change between consecutive values exceeds this amount
Rate of Change Test:		
Flag	1	1 to apply test, 0 to not apply it
Dmin	0.4	Suspect if data change per minute exceeds this amount
Flag Line Test:		
Flag	1	Flag value: 0 = don't apply test 1 = apply test to all values 2 = apply test when value is greater than the threshold value Thr 3 = apply test when value is less than the threshold value Thr
Scnt	30	Suspect if data change is $< Eps$ value for this count of values (*)
Fcnt	60	Fail if data change is $< Eps$ value for this count of values (*)
Eps	0.005	Epsilon value for above
Thr	0	Threshold value for above
Gradient Test:		
Flag	0	1 to apply test, 0 to not apply it
Gdelta	0	Suspect if adjacent values in array exceeds this amount

(*) Note: maximum allowed value for `Scnt` and `Fcnt` is 16384 as specified in the `QartodCalc.c` program. This could be changed if needed by modifying the program.

(†) Note: the ReCON parameter naming convention is an abbreviated measurement type, in this example “airtemp” meaning air temperature, an underscore (`_`) character, and then an abbreviated sensor or manufacturer name.

2.2 TEST ALGORITHMS

In general, the tests apply the flag result only to the data value being assessed. However, the Rate of Change Test and the Flat Line Test may change the flag to previous values when the test does not pass. This behavior differs from the QARTOD specifications that do not allow previous flags to be changed. Our sensor data are collected in approximately two-minute intervals and we have found it requires several measurements to assess the values accurately

2.2.1 Data Gap

A test is applied to each data value to determine if there is a time gap between it and the previous data value. To determine the expected time between data values, the delta time between data values for first 200 data values (or less, if not that many data are available) is put into a histogram. The delta time with the greatest number of occurrences is selected as the typical delta time, `typ_delta_secs`. To allow for some clock jitter in the data, a gap is defined when the time of the previous sample is greater than 1.1 times the typical delta time, which would be 10% more than the expected time.

2.2.2 Range Test

If the data value is less than `Smin` (Sensor) or greater than `Smax`, then the value is flagged as Fail. If the data value is less than `Umin` (User) or greater than `Umax`, then the value is flagged as Suspect.

2.2.3 Climate Test

The climate test to apply is indicated by a flag value, which is a string in the table. The current defined climate tests include: `AirTemp`, `SrfTemp`, `BtmTemp`, `ProTemp`, `ParBuoy`, and `SolBuoy`. Additional tests could be added.

For temperature data, the minimum and maximum values corresponding to the day of the year are read from the table file. If the temperature exceeds these values, then it is flagged as Suspect.

For PAR (Photosynthetically Active Radiation) and solar radiation data, the maximum value corresponding to the day of the year and the Coordinated Universal Time (UTC) hour of the day is read from the table file. If the radiation data value exceeds that value, then it is marked as Suspect.

2.2.4 Spike Test

A spike test requires data from the next measurement for comparison. Those data may not be available in real-time since they occur in the future. In this case, the spike test becomes a step test instead.

This test is not performed for the first value in the data or for data following a data gap as defined above. This test is implemented as follows:

If the data value does not follow a gap in the data, then:

Calculate $\delta = |x_0 - x_{-1}|$

where x_0 is the current data value and x_{-1} is the previous data value, and δ is the change in values.

If $\delta > T_{fail}$ then:

If previous value (x_{-1}) was not a spike or followed a gap, then flag as Fail.

If previous value (x_{-1}) was a spike and the value before (x_{-2}) was not a spike then:

If $|x_0 - x_{-2}| > T_{susp}$ then flag as Fail (the smaller of T_{susp} and T_{fail} is used here). Since the value did not return to within a reasonable range of what was expected, this is a second large unexpected step in the data.

Otherwise, do not flag value; the previous value was the spike and has already been flagged.

If $\delta > T_{susp}$ then:

Same tests as for $\delta > T_{fail}$, but flag as Suspect.

The parameter T_{susp} and T_{fail} values for this test are dependent upon the time between data samples. They are chosen based upon the data period used for the particular sensor.

2.2.5 Rate of Change Test

This test checks if there is a trend in the data either increasing or decreasing above a threshold. The trend is checked over the current data value and the previous four values. It is executed only if all the previous four values are good (indicated by the Pass flag) and without gaps.

A value called `roc_comp_value` is calculated that represents the rate of change comparison value in data units per minute. It is calculated from the `Dmin` parameter and `typ_delta_secs`, the typical time between data values in seconds, as follows:

$$\text{roc_comp_value} = \text{Dmin} \times \text{typ_delta_secs} / 60 \text{ secs}$$

The absolute value of the difference between the data value and previous data value is calculated for the present data value and the three previous data values as follows, with x_n representing the data value at time n , where $n = 0$ for the current data value, $n = -1$ for the previous data value, etc.:

$$\begin{aligned}\delta_0 &= |x_0 - x_{-1}| \\ \delta_{-1} &= |x_{-1} - x_{-2}| \\ \delta_{-2} &= |x_{-2} - x_{-3}| \\ \delta_{-3} &= |x_{-3} - x_{-4}| \end{aligned}$$

If all the delta values (δ_0 thru δ_{-3}) are greater than the `roc_comp_value`, or if all the delta values (δ_0 thru δ_{-3}) are less than `-roc_comp_value`, then the data are flagged as Suspect. The four data values x_0 through x_{-3} are flagged.

For this test, the parameter value `Dmin` is independent of the time between data samples.

2.2.6 Flat Line Test

This test checks if the data value has not changed by more than the `Eps` value after a count of samples. If the count is equal to or greater than the `Scnt` parameter, then it is flagged Suspect. If the count is equal to or greater than the `Fcnt` parameter, then it is flagged Fail. All data values in the previous `Scnt` or `Fcnt` values are flagged, except for the first value.

If the `Flat Line Test Flag` value is 1, then all data values are tested.

If the `Flat Line Test Flag` value is 2, then the data value is tested only when it is greater than the `Thr` threshold parameter value. An example of testing this way is with turbidity data. There may be long periods of time when the turbidity is near zero, so those data would be valid. But if the turbidity value becomes stuck on a mid-range value, then those data would be suspect.

If the `Flat Line Test Flag` value is 3, then the data value is tested only when it is less than the `Thr` threshold parameter value. An example of testing this way is with dissolved oxygen data. There may be long periods of time when dissolved oxygen is at full saturation. But if the dissolved oxygen value becomes stuck on a mid-range value, then those data would be Suspect.

Since the S_{cnt} and F_{cnt} parameter values are counts, they are dependent upon the time between data values.

2.2.7 Gradient Test

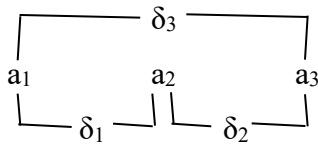
This test is for profile data (e.g., temperature, current) represented in an array format, such as a temperature or current profile. If the difference between the data values in adjoining array elements, e.g., depth bins, exceeds the G_{delta} value, then the data are flagged Suspect.

The following algorithm is used to determine if an array element value is bad. Three adjacent elements from the array are examined at a time, which will be referred to as a_1 , a_2 , and a_3 .

The absolute value of the difference between the values is calculated:

$$\begin{aligned}\delta_1 &= |a_2 - a_1| \\ \delta_2 &= |a_3 - a_2| \\ \delta_3 &= |a_3 - a_1|\end{aligned}$$

This is illustrated in the following chart:



The following tests are applied:

- If $\delta_1 < G_{delta}$ and $\delta_2 \geq G_{delta}$ then a_3 is Suspect
- If $\delta_1 \geq G_{delta}$ and $\delta_2 < G_{delta}$ then a_1 is Suspect
- If $\delta_1 \geq G_{delta}$ and $\delta_2 \geq G_{delta}$ then a_2 is Suspect and also do this test:
 - If $\delta_3 \geq G_{delta}$ then:
 - a_3 is Suspect and
 - If a_1 is first array element then a_1 is Suspect

2.2.8 Summary Flag

The program normally outputs the flags as a single summary flag value in the last column of the CSV data file. This is the highest numerical value associated with the flag value from each test performed on those data value. If any test is flagged as Fail, the summary flag will be Fail. If no tests are flagged as Fail, but any test is flagged as Suspect, the summary flag will be Suspect. If no tests are flagged as Fail or Suspect, the summary flag will be Pass.

2.3 RATIONALE FOR SELECTING PARAMETER VALUES

The parameter values used by the above tests are described in this section.

2.3.1 Range Test Values

The Sensor Minimum and Maximum values (S_{min} , S_{max}) are based on the manufacturer specifications for the sensor. If the sensor uses a data logger and the data logger measures a more limited range, then the data logger range is used. The values in the table are the acceptable values. Values that exceed the table value are flagged as Fail.

The User Minimum and Maximum values (U_{min} , U_{max}) are what would be expected for measurements anywhere in the Great Lakes region. For example, a wave sensor may be rated to measure 20 meter waves, but 9 meter (30 foot) waves is a more reasonable limit for waves in the Great Lakes. Values that exceed the table value are flagged as Suspect.

2.3.2 Climatology Test Values

2.3.2.1 - Air Temperature

To determine a reasonable air temperature range for the Great Lakes region, temperature data from the National Climatic Data Center online database were obtained for four representative cities. These cities were chosen to represent the region where most of our stations are located. The cities of Muskegon, Cleveland, Alpena, and Marquette were selected. The historical record high and low temperatures for each day of the year were used, selecting the highest or lowest value from among the cities. Since new record high or low values can be expected from time to time, the high values were increased by 1 °C and the low values decreased by 1 °C. To account for meteorological variability, we adjusted each day's value by selecting the highest and lowest values that occurred within +/- 7 days from the day of the year. Some manual editing was then applied to smooth out inconsistencies in the data. The values were then placed in a CSV formatted file named **QartodAirTempTable.csv**. This file has three values per line containing the day of the year (1-366), the minimum acceptable temperature, and the maximum acceptable temperature.

2.3.2.2 - Water Temperature

To determine a reasonable water temperature range, the historical ReCON water temperature data were used. For all locations, the water temperature data were divided into three regions: near-surface, middle water column, and near-bottom. The same approach used for Air Temperature was applied using historical data. The files are named **QartodSurfaceTempTable.csv**, **QartodProfileTempTable.csv**, and

QartodBottomTempTable.csv for near-surface, middle water column, and near-bottom water temperatures.

2.3.2.3 - PAR Sensor on Buoy

To determine the reasonable PAR (Photosynthetically Active Radiation) values for a sensor mounted in the air on a buoy, again historical ReCON data were referenced. Minimal values were not evaluated, since dark cloudy conditions can occur at any time. Since radiation varies throughout the day, a maximum value was determined for each hour of the day in UTC and each day of the year. The value was the maximum historical value with 10% added so that the limit would be above the observed values. It should be noted that though radiation increases following a sinusoidal-like curve, the 10% increase was done in the linear domain. To reduce the number of values needed in the table, every 8 days were consolidated into one, taking the maximum value over the 8 days. The lowest value used in the table was set at 10. This value was used instead of 0 to account for some radiation at night during a full moon or for a noisy sensor. The data table was edited manually to smooth out inconsistencies. The values are in units of micromoles per square meter per second ($\mu\text{mole}/\text{m}^2/\text{s}$).

The first value in each row is the day of the year (1-366). The first row and last row represent 7 days, while all the remaining rows represent 8 days each. The first row is for days 1-7, the second for 8-15, the third for 16-23, etc. The formula used to determine which row to use is to take the day of the year and divide it by 8, then discard the remainder, and add 1. This gives the row number starting with 1. After the day of the year value, the data in each row is the maximum value for each hour of the day starting with hour 0 in UTC for a total of 24 values. The hour used is selected by dropping the minutes from the observation time. The file is named **QartodParBuoyTable.csv**.

2.3.2.4 - Solar Radiation

The same procedure that was used for PAR was used to generate the table for Solar Radiation. The values for this parameter are in watts per square meter (W/m^2) and the file is named **QartodSolarBuoyTable.csv**, but otherwise it has the same format as the PAR table.

2.3.3 *Other Test Values*

The threshold values for the Spike Test, Rate of Change Test, Flat Line Test, and Gradient Test were determined by testing historical ReCON data sets. The tests were run and the parameters successively adjusted until only a small number of values were flagged as Suspect or Fail. A “small number” was subjectively defined, as it was expected when processing data for a sensor over a year that there would be between a few to tens of values, but not hundreds of values, flagged bad. If the parameters chosen were too loose, then some invalid values would not be flagged. If the parameters chosen were too tight, then some valid values would be flagged.

Sometimes the data were examined to determine if the flagged values were obviously bad. This method is somewhat subjective and time consuming, but gives confidence that the values are going to perform well.

2.3.4 Known Failures

There are times when a sensor is known to have failed, or known to be questionable because it may have been fouled or is past the calibration date. To help facilitate identifying these situations, a table file called **QartodKnownFailures.txt** is used by the program that will flag sensor values as Suspect or Fail if they are listed in the table.

The table parameters include the station ID, the sensor name, the flag value of 3 (Suspect) or 4 (Fail), the beginning fail time (year, month, day, hour/min), and the ending fail time (month, day, hour/min). A separate line is needed for each year. When the ending time is not known, then the last minute in the year is used, i.e., 12/31 at 23:59. This can be updated after the sensor is repaired.

For example, this line indicates that all data between 04/24/2019 09:00 and 05/22/2019 13:40 for this station and sensor should be flagged as 3 (Suspect):

```
mich-mkg, airtemp_airmar, 3, 2019, 04, 24, 0900, 05, 22, 1340
```

3.0 SOFTWARE CODING

3.1 RATIONALE FOR USING THE C COMPILER

The algorithms were originally coded in the R interpreted language. This turned out to cause an excessive computational load on the server because of the large quantity of data and the number of stations that needed processing. This was the case even when only newly acquired real-time data were processed instead of processing the full data set for the year. The ReCON system currently processes over 300 sensors as often as every 10 minutes, and new sensors and stations are being added from time to time which continues to increase the load. This processing significantly increases the delay between collecting the data and distributing them, which real-time systems seek to minimize.

Therefore, the algorithms were re-coded in the compiled C language. Since the compiled program runs natively on the server platform and doesn't require deciphering the code text each time it is run, it executes more efficiently and in less time. Table 4 gives the execution time of running both programs on the same set of data and demonstrates the time reduction accomplished by using the compiled program.

Table 4. Comparison of program execution times.

Parameter	Year	Station	R time	C time
Air temperature	2018	Toledo Marker #2	20 secs	<1 sec
All station data parameters	2019	Saginaw Bay Buoy	166 secs	3 secs
Barometric Pressure, 2 Hz	2019	Muskegon 45m Buoy	35 mins	21 secs

3.2 OVERALL PROGRAM STRUCTURE

The C source code file is named **QartodCalc.c** and the compiled executable is named **QartodCalc**. The software was programmed in C and does not use objects. Only the standard libraries are used and it is not linked with any special libraries or header files.

The program is executed by the ReCON data processing scripts on a Linux server. The following command line parameters are used:

```
./QartodCalc STATION DATA_TYPE INPUT_FILE OUTPUT_FILE [allflags]
```

```
where STATION      = Station name (eg: "erie-cmt")
      DATA_TYPE   = Data type to process (eg: "airtemp_csi")
      INPUT_FILE   = Input CSV file to process
                    (eg: "~/obs/tmp/erie-cmt/csvdata/airtemp_csi.csv")
      OUTPUT_FILE  = Output CSV file
                    (eg: "/data/obs/now/erie-cmt/csv_format/airtemp_csi.csv")
      allflags     = Option to also output all flags:
                    (known, range, climate, spike, flat line, rate of change)
```

The command line parameters provide the station name, the data type, and the input and output files. All the files are in text format.

The input files typically begin with a number of comments that provide station and sensor data. Each data line represents one data sample and includes the date and time, a comma, and then the data value. The time is UTC in the format YYYY/MM/DD HH:MM or YYYY/MM/DD HH:MM:SS with leading zeroes. For example:

```
2021/06/01 08:02
2021/08/04 09:00:10
```

Profile data in arrays contain multiple data values separated by commas representing multiple depths. Current data are represented in an array with two values for each depth: the east and north current vector.

The output file is a duplicate of the input file with the QARTOD summary flag appended to the end of each data line along with a preceding comma to separate it. When the “allflags” option is used, the flag is a string of the flag values in order. This is typically used for testing and debugging.

If the data type parameter does not appear in the QartodTable.txt file, then the data are not evaluated but simply copied to the output file without any modifications or flags added.

3.3 DATA BUFFER AND CALCULATED VALUES

As the data are processed, they are stored in a dynamically allocated data buffer:

```
buffer_base_ptr = ( char * ) malloc( INIT_BUFFER_SIZE );
```

The data are read from the input data file one line at a time. If the data buffer becomes full, the buffer is dynamically increased in size by 50%. Any trailing carriage return and line feed characters are removed and the flag values for each test are appended with the default value of Pass (1 or 111111 if the “allflags” option is used). After a data line is processed and no longer needed by the program, it is written to the output file along with the flag values and then removed from the buffer.

The variable *buffer_keep_lines* is set to the number of lines of data needed in the data buffer. This is used, for example, when the Flat Line Test finds and counts values that fail the test. When the Suspect or Failed count is reached, the Flat Line Test flag is set appropriately for all the relevant data lines. This needs to be done before the data are output. After setting the flags or after reaching a good data value that clears the test, those data are no longer needed by the program and can be output and subsequently removed from the data buffer.

After each line is read, the program does some processing on the data and updates several variables. The names of these variables use the following suffixes to identify where in the data stream they are associated:

- _m4 means value at -4 data lines or 4 data lines earlier
- _m3 means value at -3 data lines or 3 data lines earlier
- _m2 means value at -2 data lines or 2 data lines earlier
- _m1 means value at -1 data line or the previous data line
- _00 means value at the current data line

`_p1` means value at +1 data line or the next data line
`_p2` means value at +2 data lines or 2 data lines later
`_p3` means value at +3 data lines or 3 data lines later

The program reads the data ahead whenever possible. A data valid flag is used to indicate whether or not the data are valid (available) or not, which mostly applies to the `_m4` through `_p3` values. The data value and valid flag variables are named:

<code>value_00</code>	data value
<code>valid_00</code>	data are valid

The following variables are calculated for each data value when it is read in. By doing this, these calculations don't need to be repeated in the data processing:

<code>csec_00</code>	data time stamp in seconds since epoch
<code>data_julian_00</code>	data time day of year (1-366)
<code>data_hour_00</code>	data time hour (0-23)
<code>gap_00</code>	flag indicating a time gap between this data and the previous data
<code>delta_00[]</code>	difference (signed) between this data and the previous data
<code>spike_00[]</code>	spike QARTOD flag for this data
<code>qflag_krsf_00[]</code>	summary of known, range, spike, and flat line tests for this data

Only the `_00` variables are shown above, but there are additional variables for the values from `_m4` through `_p3` when needed. The brackets shown as `[]` identify values that are arrays for input data lines that include multiple columns.

3.4 PARAMETER FILE FORMAT

The following parameter files are read by the program:

```
QartodTable.txt
QartodKnownFails.txt
QartodAirTempTable.csv
QartodSurfaceTempTable.csv
QartodProfileTempTable.csv
QartodBottomTempTable.csv
QartodParBuoyTable.csv
QartodSolarBuoyTable.csv
```

Comments are indicated by the '#' (pound sign) character at the beginning of the line. Empty lines are ignored and all space and tab characters are removed, so the file can be freely formatted for clarity.

3.5 OUTPUT FILE

The output file duplicates the input file while appending the QARTOD flags to the end of the data lines. The output file format is a comma separated file, with a CSV file name. The lines are terminated with the carriage return and line feed character combination which is recognized by the major computer platforms. The format allows the file to be directly read in and recognized by most spreadsheet software programs including online based spreadsheets.

Here's a simplified example of the beginning of a typical output file:

```
"Platform","Muskegon M20 ReCON Buoy"  
"Latitude (deg)",43.1826  
"Longitude (deg)",-86.3600  
"Water Depth (m)",22  
"Sensor Device","Campbell Scientific 109 Temperature Probe"  
  
"Time (UTC)","Air Temperature","Qartod"  
"YYYY/MM/DD HH:MM","C","flag"  
2022/04/12 14:00, 5.828218, 1  
2022/04/12 14:02, 5.798052, 1  
2022/04/12 14:04, 5.788618, 1  
2022/04/12 14:06, 55.789778, 4  
2022/04/12 14:08, 5.808569, 1  
2022/04/12 14:10, 5.829861, 1
```

4.0 SUMMARY

The QartodCalc.c program provides an efficient way of implementing the QARTOD based quality assurance tests on real-time data from ReCON stations. The parameters for the algorithms are contained in separate files allowing for the easy addition of future sensors and easy adjustment of the test parameters to identify suspect or bad data according to the specific conditions in the Great Lakes region. As such, it can be readily adopted to any region or real-time data collection program.

5.0 REFERENCES

RUBERG, S.A., S.B. BRANDT, R.W. MUZZI, N. HAWLEY, T. Bridgeman, G.A. LESHKEVICH, J.C. LANE, and T.C. MILLER. A wireless real-time coastal observation network. EOS Transactions 88(28):285-286 (2007).
<https://www.glerl.noaa.gov/pubs/fulltext/2007/20070022.pdf>

RUBERG, S.A., R.W. MUZZI, S.B. BRANDT, J.C. LANE, T.C. MILLER, J.J. Gray, S.A. CONSTANT, and E.J. Downing. A wireless internet-based observatory: The Real-time Coastal Observation Network (ReCON). Proceedings, Marine Technology Society/IEEE Oceans 2007 Conference, Vancouver, British Columbia, Canada, September 30-October 5, 2007. 6 pp. (2007).
<https://www.glerl.noaa.gov/pubs/fulltext/2007/20070045.pdf>

Manual for Real-Time Quality Control of Wind Data, Version 1.0, October 2014, Integrated Ocean Observing System. <https://ioos.noaa.gov/ioos-in-action/wind-data/>

Manual for Real-Time Quality Control of In-Situ Current Observations, Version 2.0, October 2015, Integrated Ocean Observing System. <https://ioos.noaa.gov/ioos-in-action/currents/>

Climate Data Online Search, National Climatic Data Center.
<https://www.ncdc.noaa.gov/cdo-web/search>

6.0 SUPPLEMENTARY MATERIALS

QartodAirTempTable.csv

File containing minimum and maximum acceptable values for air temperature data.

QartodBottomTempTable.csv

File containing minimum and maximum acceptable values for near-bottom water temperature data.

QartodCalc.c program

The C software program that evaluates the quality of the data and adds flags to identify the quality of each data value.

QartodCalc.R program

The R software program that evaluates the quality of the data and adds flags to identify the quality of each data value.

QartodKnownFails.txt

File containing table of sensors that are known to have failed and the time range of the failure.

QartodParBuoyTable.csv

File containing maximum acceptable values for PAR (Photosynthetically Active Radiation) data.

QartodProfileTempTable.csv

File containing minimum and maximum acceptable values for mid-column water temperature data.

QartodSolarBuoyTable.csv

File containing maximum acceptable values for solar radiation data.

QartodSurfaceTempTable.csv

File containing minimum and maximum acceptable values for near-surface water temperature data.

QartodTable.txt

File containing table of quality parameter values used for each quality test.