Rockland Technical Note 054

— Rockland Data Logger — In Situ Data Processing

June 23, 2026

Rockland Scientific International Inc.
520 Dupplin Rd
Victoria, BC, CANADA, V8Z 1C1
www.rocklandscientific.com

Copyright Notice

info@rocklandscientific.com
tel: +1 250 370 1688

Version History

Date	Description
2021-11-18	Initial version
2026-06-23	Revised to align with latest versions of TNs 50 and 51

Contents

1 Introduction

Ocean microstructure turbulence measurements, especially on autonomous platforms, require careful post-processing, incorporating various ancillary data streams to compute dissipation rates and other parameters of interest. Owing to the high sampling rate and large volume of raw data, near real-time transmission of raw data to shore while an instrument is deployed is usually not feasible, introducing a large delay between data acquisition and their availability for analysis. Without access to near real-time data, users cannot modify missions in response to the observed conditions. Additionally, if a platform is lost before recovery, the locally recorded data is lost as well. Loss of a platform is of particular concern to glider operators, so Rockland Scientific has developed an *In Situ* Data Processing (ISDP) module for Rockland Data Logger (RDL) equipped instruments that can process microstructure data in near real-time, creating processed data products that can be transmitted to shore over satellite channels.

The Rockland Scientific ISDP module can provide users with near real-time processed turbulence data, along with additional information on the health of their instrument and the quality of the acquired data. This document is intended to provide instrument users with

1. A basic overview of the ISDP module data processing flow;
2. Explain the setup, configuration, and pre-deployment checks of the ISDP module;
3. Describe the format of the resulting processed data files.

Beyond the configuration of ISDP on Rockland Instruments described in this document, there could also be platform specific configuration steps (for example, those imposed by the glider manufacturer) which are outside of the scope of this document.

2 Data Processing

The ISDP module is a software package that runs on RDL instruments and processes segments of microstructure data as it is being written to the RDL memory module. The ISDP data acquisition and processing workflow is shown in Figure 1. Following this workflow, ISDP returns spectra computed from the microstructure channels, estimates of the dissipation rate of turbulent kinetic energy (ϵ), and associated quality control parameters. The results are then written to a `.q`-file for telemetry and further analysis and benefit to the user.

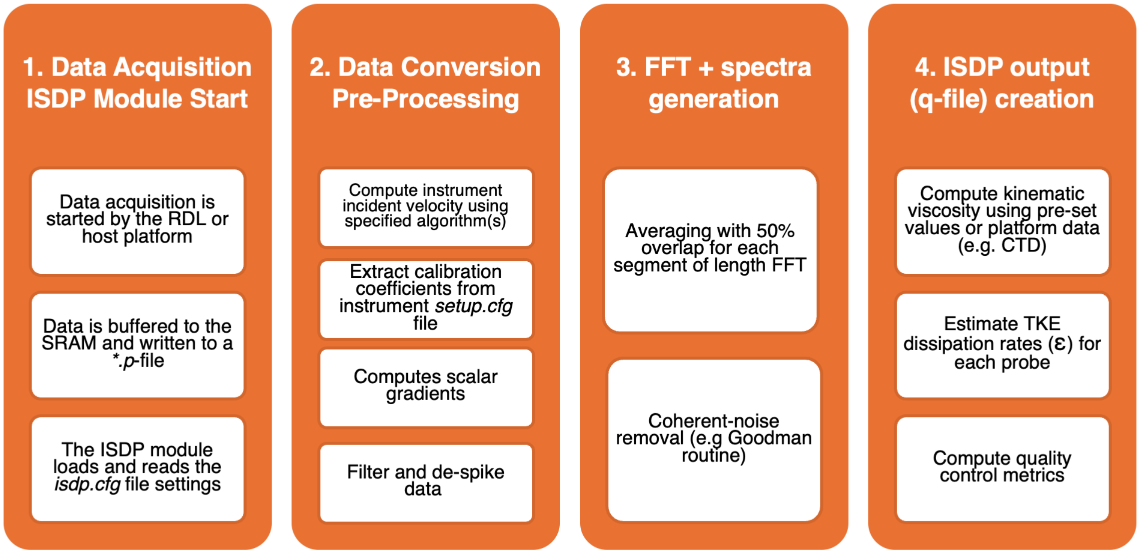

Figure 1: Block diagram illustrating the progression of automated data processing steps performed by the ISDP module.

Key takeaways include:

- The ISDP module processes data as it is acquired on the instrument and written to the `.p`-file.
- The ISDP module makes certain assumptions that are also outlined in the workflow graphic. Chiefly, the ISDP module does not, by default, have access to all the ancillary data available from the host platform, such as incident velocity U and kinematic viscosity ν (as could be derived from an onboard ADCP or CTD-sensor).
- Data segments are processed one `diss_length` at the time, which leads to divergence from standard `.p`-file processing. This is particularly marked in data filtering and despiking. As such, the accuracy of ϵ can be further improved during post-processing of the full data set. The estimate provided in the `.q`-file are used to inform the user of their instrument health and facilitate operational decision making.
- ISDP output and a `.q`-file can only be generated once sufficient data have been acquired to complete at least one processing interval, as defined by `diss_length`. Profiles shorter

than this interval may not produce a `.q`-file. The value of `diss_length` should therefore be chosen with the expected profile duration, platform speed, and desired vertical resolution in mind. For example, for a glider profiling at 0.12 $\mathrm{m\,s^{-1}}$ over a 60 m dive, a `diss_length` of 30 s would produce approximately 16 ISDP estimates over the dive.

- Field experiments have demonstrated close agreement between the ISDP module outputs and those generated in post-processing using the best practices laid out in Lueck *et al.* (2024).

3 Setup, Configuration, and Pre-Deployment

The following sections are intended to guide the user through the setup, configuration, and pre-deployment checks that will ensure proper ISDP behaviour and while deployed.

3.1 Instrument Requirements

The capability to process acquired data *in situ* has been introduced with the new Rockland Data Logger (RDL) which has sufficient computational power and internal memory to process the microstructure data in real-time. Please note that ISDP is *not* available on Rockland Instruments with CF2 Persistor hardware [1].

The ISDP module is typically factory configured during manufacturing or instrument servicing. Please contact the Sales & Support team if you are interested in using ISDP and have not had ISDP previously configured for your RDL instrument.

For ISDP to work correctly on any RDL instrument, the following four conditions must be met:

1. The ISDP enabled flag on the instrument is set to true. Contact Rockland Support if more information is required, or to request an `.RSU` file to modify this flag.
2. A valid license file is installed correctly on the instrument. Contact Rockland Support to request a license file if you do not yet have one. To install the license file, place it on the RDL memory module. The RDL will automatically detect the license file and copy it to another directory for safety.
3. The `isdp.cfg` configuration file must exist in the RDL memory module.

3.2 Power Consumption

Using ISDP increases the wake time of the data logger's main computer, increasing the power consumption of the instrument. As such, using ISDP increases the instrument power consumption by approximately 100 mW.

---

[1] To inquire about replacing your CF2 hardware with an ISDP-capable solution, please contact the Rockland support team.

3.3 Platform Availability

The ISDP module is available on all RDL equipped Rockland Scientific instruments. If the instrument is integrated onto a platform, the user will also need to verify with the platform manufacturer whether the ISDP capability is supported. Rockland Scientific has partnered with multiple platform manufacturers to integrate ISDP functionality into their platforms. Rockland recommends users that wish to avail themselves of the ISDP capability onboard their chosen platform should contact the manufacturer to get further instructions on the particular integration and workflow for the ISDP data streams. Below is described the typical pathway for ISDP interaction with third-party platforms:

1. The RDL instrument is turned on by the host platform to initiate data acquisition, with the ISDP module pre-configured by the user.
2. `.q`-files are generated by the ISDP module as data is collected.
3. The host platform has the option to request all, or a subset, of the generated `.q`-files from the instrument and download them to the host platform (for example, to transmit them back to shore).
4. The instrument, after transmitting the `.q`-files, resumes normal operation.

3.4 Configuring ISDP

The data processing flow conducted by the ISDP module is controlled by a set of parameters stored in the `isdp.cfg` configuration file. The configuration file is an ASCII-encoded text file with a series of `Key = Value` pairs. Line comments begin with the semicolon (`;`) character. The `isdp.cfg` file must be located on the RDL memory module.

The parameters below follow the same nomenclature as the processing parameters of the Zissou suite, please review the Zissou Premium User Guide for additional information about the meaning and setting of these parameters.

In preparation for a deployment, `isdp.cfg` should be reviewed. The ISDP configuration file contains the following fields:

algorithm ”glide” | ”em” -The method used to compute the speed of flow past the probes. ”em” is only applicable for instruments equipped with an EM current speed sensor. Default = ”glide”

aoa `float` $\in [0, 10]$ - Angle of attack (in degrees) for the platform used to compute speed. *Only* applicable when `algorithm = glide`. Default = 3

band_averaging ”true” | ”false” - Toggle to control whether the spectral data is band averaged into bins. Default = ”true”

diss_length `float` $\in [2 \times \texttt{fft\_length}, 1000]$ - Length of data (in seconds) used to compute spectra and the dissipation rate of turbulent kinetic energy. Default = 8

f_aa `float` - Anti-Aliasing filter cut-off frequency (in Hz). Should not be changed unless recommended by Rockland. Default = 98

fft_length `float` $\in [1/16, 1/2 \times \texttt{diss\_length}]$ - Length of data (in seconds) used to compute the Fast Fourier Transform (FFT). Defualt = 2

fit_order `int` $\in [2, 5]$ - Order of the polynomial used to identify the minimum of the spectra for estimating K_{max}. Default = 3

goodman_spectra ”true” | ”false” - Toggle to activate the goodman frequency-domain coherent noise removal routine. Default = ”true”

hp_cut `float` $\in [0, 4]$ - Cut-off frequency (in Hz) used to high-pass filter the shear and scalar microstructure data, removing low frequency signals from the platform’s motion. Default = 0.125

inertial_sr `float` ≥ 0 - Threshold (in W m^3) above which dissipation estimates are computed using the Inertial Subrange Routine. Default = 1.5×10^{-5}

num_frequency `int` $\in [0, 100]$ - Number of bins to average the frequency and spectra over. The bins are logarithmically spaced. If 0, no spectra are computed.Default = 28

order `int` $\in [1, 4]$ - Fit order used to remove trends from the shear and vibration data prior to computing spectra. A polynomial of this order is fitted and subtracted to eliminate low-frequency variations. Default = 1

overlap `float` $\in [0, 0.75]$ - Overlap fraction between dissipation segments. Default = 0

shear_despiking Three comma separated fields:

> **Threshold ratio** `int` $\in [1, \infty]$ - Threshold for the ratio of the high passed data divided by the low passed data to identify a spike. Default = 8
>
> **Smoothing frequency** `float` $\in [0, 256]$ - Low-pass cut-off frequency (in Hz) applied to the data to identify spikes. Default = 0.25
>
> **Window** `float` $\in [0.01, 0.1]$ - Time window (in seconds) around each detected spike that will be replaced. The algorithm removes approximately the window width before the spike and twice the window width after it, ensuring the entire spike and its transient region are suppressed. Default = 0.04

tau `float` $\in [1, 100]$ - Smoothing time constant (in seconds) applied to the computed speed. It sets the cutoff frequency to $0.68/\tau$ for a first-order low-pass Butterworth filter. *Only* applicable when `algorithm = glide`. Default = 3

3.5 Pre-Deployment Checks

To test and ensure that your ISDP module performs as expected, Rockland recommends ensuring that a `.q`-file is generated for every `.p`-file created during bench tests [2].

---

[2] There is a threshold for q-files to be generated, as such, we recommend acquiring data files of 5 minutes or more.

Rockland strongly recommends testing that `.q`-files are created with every ISDP configuration file (`isdp.cfg`) installed on your instrument.

Rockland provides a software solution for `.q`-file visualization. It can be found on the Rockland Online Portal. The `.q`-file specification is also included in this technical note in Appendix A, allowing those customers who so desire to create their own software to visualize and inspect `.q`-files.

4 ISDP Module Outputs

The following sections describe the outputs of the ISDP module.

4.1 `.q`-files

The `.q`-file is a binary file that stores the processed results produced by the ISDP module. The content of the `.q`-file is meant to provide sufficient information to assess the quality of the data acquired by the instrument, monitor the instrument health, and provide near real-time scientific information.

The contents of the `.q`-file include:

- Sensor data averaged over dissipation length (e.g. pitch, temperature, input voltage, etc.)
- Band-average spectra for microstructure measurements (e.g. shear and temperature spectra)
- Processed outputs such as the dissipation rate of kinetic energy and its associated quality control factors

The `.q`-file output is computed and packaged such that the `.q`-file remains sufficiently small for transmission via satellite whilst still offering valuable information to enable operational decision making. The `.q`-file output size is a function of the number of spectral points (`num_frequency`) and the length of the processed segments (`diss_length`) set in `isdp.cfg`. Assuming 30 spectral points, a dissipation length of 30 s, and an average vertical speed of 0.3 $\mathrm{m\,s^{-1}}$ (providing approximately 10 m of vertical resolution), a 1000 m cast would produce a `.p`-file on the order of 24.5 MB and a `.q`-file on the order of 40 KB, or 0.16% the size of the `.p`-file.

The `.q`-file specification is also included in this technical note in Appendix A.

4.2 `.mri`-files

As a convenience for the platforms which must periodically download the processed data files from Rockland's instruments, the `mergeqfiles` utility is provided to merge multiple `.q`-files into a single file for download and transmission. The merged file has the suffix `.mri`, and is composed of concatenated `.q`-files. See Appendix A for file specification details.

5 Troubleshooting

The ISDP module for new RDL equipped instruments is configured at the factory. For existing RDL equipped instruments, the ISDP module can be configured as part of regular maintenance, instrument servicing, or upgrades. If the ISDP module is properly configured, the user should be able to see and access `.q`-files on the RDL memory module.

If you are experiencing issues with `.q`-file creation, please contact Rockland support for help and further troubleshooting steps.

A .q-file Specification

The .q-file is a binary file, written by the ISDP module, and used to store the processed data. The general format of the file is shown in Figure 2.

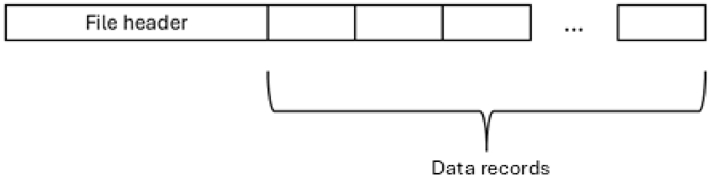

Figure 2: Schematic form of the .q-file

A.1 File Header

The file header includes the meta data required to read the content of the files. The header structure is described in Table 1.

Table 1: Description of the .q-file header fields

Field	Description	Format
File identifier	Integer used to identify .q-file (`0x1729`)	UInt16
File version	Version of the file format (eg. 1.2)	Float32
Date time	Date and time at the start of the file[1]	UInt64
Number of channels (N_c)	Number of channels written to the file	UInt16
Number of spectra (N_s)	Number of spectral types written to the file	UInt16
Number of spectral points (N_f)	Number of spectral points reported[2]	UInt16
Channel ID	List of integers to identify each channel[3]	$N_c \times$UInt16
Spectrum ID	List of integers to identify each spectrum[3]	$N_s \times$UInt16
Frequency bins	Array of centre frequencies for spectral points	$N_f \times$Float16
Configuration record	Record containing the processing parameters	Table 2
Data record size	Size of individual data records	UInt16

1. The date and time in the header is in millisecond since 0000/01/01 00:00
2. The number of spectra points written to the .q-file are typically band averaged. The number of these spectral points written in the file is stored in the file header.

The file header includes a record of the processing parameters that were used in creating the .q-file. The configuration record structure is described in Table 2.

Table 2: Description of the `.q`-file configuration record

Field	**Description**	**Format**
Configuration record identifier	Integer used to identify the configuration record (`0x0827`[1])	UInt16
Configuration record size (s)	Size of the string containing the processing parameters	UInt16
Processing parameters	String of the parameters and their value	$s\times$char

1. There is a known bug in v1.2 `.q`-files where the configuration record is preceded by `0x1657` rather than `0x0827`.

The following is an example of the processing parameter string stored in the configuration record[3]:

```
DictString, Any with 21 entries:
"hp_cut" => 0.4
"ucond_despiking" => [10.0, 0.5, 0.04]
"overlap" => 0.0q
"fit_order" => 3
"band_averaging" => true
"tau" => 3
"num_frequency" => 0
"inertial_subrange" => [1.0e-5]
"order" => 1
"file" => "/home/debian/data/isdp.cfg"
"inertial_sr" => 1.5e-5
"diss_length" => 60.0
"algorithm" => "glide"
"goodman_spectra" => true
"shear_despiking" => [8.0, 0.25, 0.04]
"aoa" => 3
"f_aa" => 98.0
"goodman_length" => 0.0
"Nv" => 2
"instrument" => "sea_explorer"
"fft_length" => 4.0
```

---

[3] The lines are separated with a ”line feed” character (`ASCII 0x0A`).

A.2 Data Records

The remainder of the file is composed of data records. Data records have a short header and contain the output of the processed data for one dissipation length. The data record structure is described in Table 3.

Table 3: Description of the `.q`-file data records

Field	**Description**	**Format**
Data record identifier	Integer used to identify the data record (`0x1657`)	UInt16
Record number	Integer count of the current record	UInt16
Error	Currently unused	Int64
Start time	Start of the record in second since the beginning of the file.	Float16
Stop time	End of the record in second since the beginning of the file.	Float16
Channel data	Value for each of the channels identified in the file header	$N_c \times$Float16
Spectral data	Vector for each of the spectra identified in the file header	$N_s \times N_f \times$Float16

A.3 Reading `.q`-files

The following is a quick guide to aid in parsing the `.q`-files.

1. The objective of parsing a `.q`-file is to extract from it time series for each stored channel and matrices containing the spectra for each dissipation segment.
2. Reading the file header will provide the list of channels and spectra to expect using their respective IDs. **Note:** It is possible to configure ISDP not to return any spectra by setting both the number of spectra and number of spectral points to 0.
3. Although the total number of records is not stored in the file header, it can be extracted by reading the last record of the file and using the record number in the data record header.
4. Finally, when decoding the data records, the values are stored in the same sequence as the channel IDs in the file header, followed by each spectra. Hence all values can be appended to their respective channel vector and spectra matrix.

A.4 `.mri`-files

In addition to `.q`-files, `.mri`-files are another processed data format that are created using the `mergeqfiles` utility. The `.mri`-file are simply made of concatenated `.q`-files.

End of document