RSI Technical Note 044

MicroRider Proglet for Slocum Glider

Evan Cervelli

Tina Haskins

2019-01-30

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

Contents

1 Introduction

The purpose of this technical note is to provide a guide for setting up the MicroRider Proglet on G1 and G2 Slocum Gliders using TWR Glider Software version 8.0 and higher.

1.1 Overview of the MicroRider Proglet

The MicroRider proglet was created by Teledyne Webb Research (TWR) with input from Rockland Scientific to improve the integration of the MicroRider onto the Slocum Glider platform. The main purpose of the proglet is to allow the glider to capture limited data from the MicroRider channels during the dive cycle. The captured data is stored on the glider science computer. The captured data can then be sent back over Iridium to assess the health of the microstructure probes and the overall performance of the MicroRider. The proglet initiates verbose output from the MicroRider and captures the average (`avg`) and mean absolute deviation (`std`) of each channel during the dive.

An additional and very convenient feature of the proglet is the syncing of the MicroRider clock and the glider's science computer clock each time the MicroRider begins data acquisition. This removes the need for the user to align data in the glider hotel file with the MicroRider data during post processing.

1.2 Requirements for use of MicroRider Proglet

Use of the MicroRider Proglet requires an RS-232 serial connection between the MicroRider and glider science board. Some early MicroRider Glider systems may not have the RS-232 serial connection installed. If you are unsure if the serial connection exists on your system or if you would like it to be added, please contact TWR.

1.3 Important Note Regarding TWR Glider Software Versions

The MicroRider uses the ODAS5IR Instrument Software. The ODAS5IR software is used on many Rockland Scientific products including instruments with weight releases. As a result, 'release events' will occur during each MicroRider profile and are recorded in the `logfile.txt` in the `DATA` directory on the MicroRider. It was discovered that these release events are recognized as errors by the proglet onboard the glider. The MicroRider will be restarted by the proglet whenever an error occurs. The first release event typically occurs after the MicroRider registers a depth greater than 100 dBar. The ODAS5IR software references the MicroRider pressure sensor to determine the conditions for a release event. If you are using TWR Glider Software version 8.0 or 8.1 you must adjust the number of allowable errors in the `uRider.ini` file to prevent unwanted restarts of the MicroRider. Set the number of allowable errors to a high number (5000 is recommended, max number is 31,000). TWR has fixed this bug in TWR Glider Software version 8.2 and

higher. For more information regarding release events please reference the Rockland Scientific ODAS5IR Software User Guide. If you are using a previous version of the glider software or wish not to use the MicroRider Proglet, you can activate the MicroRider with the generic science logger proglet. For further tips on remote troubleshooting using the generic science logger proglet with the MicroRider, please see section 3.

1.4 Flight Controls to Minimize Noise in the Data

Gliders are generally a very quiet platform for turbulence measurements, however specific glider functions will induce noise that will temporarility degrade the quality of microstructure data. A discussion of the sources of noise on a glider including the rudder, pitch battery and buoyancy pump can be found in Technical Note 022: Turbulence Measurements from a Glider. During a typical glider profile, small adjustments to the pitch battery and ballast pump may occur. These small adjustments will render the microstructure data useless for the duration of the adjustment. To avoid compromising this data, it is recommended to set the pitch battery and buoyancy pump to appropriate constant values for the duration of each profile. Test dives are required to determine the appropriate constant values for the pitch battery and buoyancy pump for each individual glider.

2 Setting up the MicroRider Proglet

The MicroRider Glider system is composed of two Persistor computers on the glider and one Persistor computer on the MicroRider. Changes are required to the files on the compact flash (CF) cards associated with all three Persistor computers. Please note that many of the picoDOS commands familiar to glider operators are applicable when communicating with the MicroRider. Please reference the ODAS5IR User Guide for more information about the MicroRider software.

2.1 MicroRider CF Card file changes

The `autoexec.bat` file must be kept on the MicroRider CF card. The MicroRider Proglet initially looks for the string `Data acquisition will start in`, which will only be generated if data acquisition is started by the `autoexec.bat` file. The verbose output flag `-D` should be added to the odas5ir command line in the `autoexec.bat` file. Note that the proglet will still work if the `-D` flag is not included in the `autoexec.bat` file on the MicroRider, however TWR recommends that it is added. The contents of the `autoexec.bat` file should read:

```
odas5ir -f setup.cfg -l 3000 -D
```

An easy way to change the contents of the `autoexec.bat` file is to use the `capture` command. No other changes are required to the MicroRider CF card. The `c:\` directory of the

MicroRider CF card should read:

```
ODAS5IR.PXE
USBL.PXE
DATA <DIR>
SETUP.CFG
AUTOEXEC.BAT
```

2.2 Flight CF card file changes

The MicroRider proglet will run using the default `sci_on.mi`. However, TWR recommends establishing more control of the MicroRider Proglet by creating a `sample73.ma` file for the MicroRider during a mission. `sci_on.mi` is for testing in gliderDOS and `sample73.ma` is the recommended way to operate the sensor in a mission allowing individual control. The `.ma` file then needs to be included in your mission.

Create a `sample73.ma` in `c:\mafiles`

```
behavior_name=sample
#SAMPLE73.MA (microRider)

<start:b_arg>
   b_arg: sensor_type(enum)             73  # PROFILE     1  _microRider_on
                                            # This is a bit-field, combine:
                                            # 8 on_surface, 4 climbing, 2 hovering, 1 diving
   b_arg: state_to_sample(enum)          7  # 7  diving|hovering|climbing

   b_arg: intersample_time(s)            0  # if < 0 then off, if = 0 then
                                            # as fast as possible, and if
                                            # > 0 then that many seconds
                                            # between measurements

   b_arg: nth_yo_to_sample(nodim)        1  # After the first yo, sample only
                                            # on every nth yo. If argument is
                                            # negative then exclude first yo.

   b_arg: intersample_depth(m)          -1  # supersedes intersample_time
                                            # by dynamically estimating
                                            # and setting intersample_time
                                            # to sample at the specified
                                            # depth interval. If <=0 then
                                            # then sample uses
                                            # intersample_time, if > 0 then
                                            # that many meters between
                                            # measurements

   b_arg: min_depth(m)                  -5  # minimum depth to collect data, default
                                            # is negative to leave on at surface in
                                            # spite of noise in depth reading
   b_arg: max_depth(m)                2000  # maximum depth to collect data

<end:b_arg>
```

Sample73.ma then needs to be called in whatever mission you run c:\missions

```
# SAMPLE : microRider
behavior: sample
b_arg: args_from_file(enum) 73  # c_microRider_on
```

2.3 Science CF card file changes

Use the command consci on your vehicle to transfer to the science computer and navigate to c:\config. Proglets.dat should contain your MicroRider configuration and they should be uncommented. Note: bit and UART numbers will vary depending on your vehicle configuration. The bit and UART numbers may be different for each glider. Check the wiring of the connector from the MicroRider to the science computer board, wiring for each UART number is described in comments found in the proglets.dat file (see below for example). A good way to test that you have the correct UART and power bit is to use the u4stalk command. For more u4stalk command details see section 3.

```
# Rockland Scientific MicroRider MR-1000 (MICRORIDER), baud rate 9600
#
proglet = microRider
uart       = 4          # U4Soem Pins T-2,R-4
bit        = 0          # power control for sensor
start_snsr = c_microRider_on(sec)
#
```

Two files urider.cfg and urider.ini should also be stored in C:\config on the science computer. Note that in urider.ini POWER_WAIT_SECS should be increased from 30 to 60. This change has been set as default in TWR release 8.1 software.

```
# uRider.cfg
# This file is required and must contain a valid 'odas5ir' message with
# the '-D' parameter
# The customer should modify this file with their specific sampling command.
#
# 2017-08-08 mbrissette@teledyne.com Initial

# NOTES:
#  Comments start with '#'
#  This file must contain a single MicroRider 'odas5ir' command.
#  The proglet sends this command to the MicroRider to start sampling.
#  The proglet is responsible for setting the device time, based on the
#     'sync_clock' parameter located in the 'microRider.ini' file.
#  The command must contain a '-D' parameter, "Display Verbose Channel Output"
#     in order to capture and place channel data into glider sensors.

#   usage :
#     odas5ir -F <config file > [-C <all |0..255 >] [-E] [-B] [-R rate ] [-S sample ]
#       -A : use fast serial instrument bus
#       -B : create backup file
#       -C <all |0..255 > : specify channel (or all) to calibrate
#       -D : display verbose channel output
#       -E : enable echo mode [ FACTORY USE ONLY ]
#       -F <config file > : file to use for data acquisition settings
```

```
#       -H : show this help message
#       -L : system clock speed in kHz ( default 16000)
#       -N : skip 10s countdown before starting DAQ
#       -R : specify sample rate ( only in calibrate mode )
#       -S : specify number of samples ( only in calibrate mode )
#       -T : maximum duration of data file ( default : no limit ) [ records ]
#       -V : display battery voltage
#       -Z : maximum size of data file ( default : 2048 MB) [KB]

# Actual command sent to device, must contain '-D'
odas5ir -f setup.cfg -l 3000 -D
```

The following is an example of the `uRider.ini` file. Note that the `POWER_WAIT_SECS` attribute should be 60 not 30. For software versions 8.0 and 8.1 you may need to allow for some errors to accommodate the release events which may be generated by the MicroRider. In this case, set `NUM_ERRORS` to a high number such as 5000.

```
# uRider.ini
# This is the Rockland Scientific MicroRider (MICRORIDER) proglet initialization file
#
# This file is used to configure proglet specific parameters,
# configuration parameters take the format:
#              <attribute>: <value>
#
# 2017-08-08 mbrissette@teledyne.com Initial


# The Max Message Wait Seconds attribute determines how long the proglet
# will wait for a Verbose Console Output message. The absence of a message within
# this time is an indication that the MicroRider's internal flash memory is full.
# n - maximum seconds to wait for a Verbose Console Output message
MAX_MESSAGE_WAIT_SECS: 60

# The Maximum Number of messages to store sets the limit for the number of messages
# stored in the 'xxxxxxxx.mrd' files.
# n - store 'n' messages maximum to the 'xxxxxxxx.mrd' file.
#           a value of -1 indicates no maximum
STORE_MAX_MSGS: -1

# ##############################################################################
# The following parameters should only be changed when directed by Teledyne support
# ##############################################################################

# The Diag Level parameter is used to capture diagnostic information, this should only
# be changed at Teledynes request.  When the diag level is > 0, output is generated to
# the science terminal and the science NLG file.
# 0 - DIAG_LEVEL_OFF, normal mode of operation, errors are reported.
# 1 - DIAG_LEVEL_1,   proglet trace info.
# 2 - DIAG_LEVEL_2,   DIAG_LEVEL1 + message tracing.
DIAG_LEVEL: 0

# The Number of Errors attribute determines how many errors will be detected before
# a failing status is returned and a restart is generated.
# n - Number of acceptable errors.
```

```
NUM_ERRORS: 0

# The Power Wait Seconds attribute determines how long the proglet waits
# before communicating with the device after enabling power.
# n - seconds to wait before accessing the device after enabling power.
POWER_WAIT_SECS: 60

# The Command Prompt Wait Seconds attribute determines how long the proglet will
# wait for the command prompt
# n - maximum seconds to wait for the command prompt before generating an error.
CMD_PROMPT_MAX_WAIT_SECS: 5

# The Sync Clock attribute determines if the proglet updates the time on the device
# n - 0 = do not update the device time, 1 = update the device time.
SYNC_CLOCK: 1

# The Stop Sampling Max Wait Seconds attribute determines how long the proglet
# will wait for a response before removing power from the device
# n - maximum seconds to wait for a response prompt before removing power
STOP_MAX_WAIT_SECS: 4
```

If you are having trouble with the MicroRider proglet you may wish to set the `DIAG_LEVEL` to equal 2. This setting will produce messages in the NLG file, when logging is enabled, that can be used for troubleshooting. This setting may be requested by a TWR technician during remote troubleshooting assistance.

2.4 TBD and NBD Setup

All channels are being logged in the EBD file and the MRD file. The variables listed in TBD and NBD are to determine probe health during flight. Probe channels are mapped in the `setup.cfg` on the MicroRider which then has a list of matching channel numbers in the masterdata for TWR release 8.0 and higher. These can differ between MicroRider and how the instrument is setup. Note that EM Current Meter channels are not supported in 8.0 and 8.1 but will be supported in 8.2. The Micro Conductivity probe channels are not supported as they are not recommended for use on a glider.

2.4.1 Important (TBDlist)

The following channels are recommended for TBDlist:

- T1: Average is valuable but pseudo standard deviation is questionable
- T1_dT1: Will tell you if signal is noisy, pseudo standard deviation is valuable but average is not.
- T2: Same as T1
- T2_dT2: Same as T1_dT1

- Incl_Y: Required to distinguish ascends from descends.
- P: Could be useful in determining a leak. If water enters the main pressure tube, one of the first places it will contact is the back of the pressure sensor on downward profiles.
- S1: Pseudo standard deviation is valuable. See notes below.
- S2: Pseudo standard deviation is valuable. See notes below.

Shear probes cannot be evaluated at the surface since they are essentially vibration sensors and the statistics will vary based on surface sea conditions. The record average (`avg`)is of little value because the expectation is zero. Even a completely broken shear probe may have a record average of near zero. Only the mean absolute deviation (`std`) is useful. A broken probe will have a large `std`, more than 1000 counts. You will also get a large `std` from fouling (seaweed, jelly fish, zooplankton) but these will be transient. Persistently high values are a bad sign. The range of values is -32,768 counts to +32,768 counts.

2.4.2 Moderately Important (NBDlist)

The channels in the TBDlist are recommended for the NBDlist in addition to the following two channels:

- Ax: See notes below
- Ay: See notes below

Ax, and Ay channels are the vibration sensors embedded in the front bulkhead of the MicroRider and used in determining if there are excessive vibrations other than turbulence that the MicroRider is experiencing. These channels are very important for post processing and are recommended as an NBD troubleshooting variable.

The following channels are not recommended for the NBDlist

- P_dP
- PV
- EMC_Cur
- Incl_T
- Incl_X
- Special Character

Example TBD:

```
# microrider
# ----
# sci_microrider_last            60   7
# sci_microRider_next            0          # must be present in output msg
# sci_microRider_pressure        0          # must be present in output msg
# sci_microRider_file_number     0          # must be present in output msg
# sci_microRider_c1_avg          60   7     # Ax vibration
# sci_microRider_c1_std          60   7
# sci_microRider_c2_avg          60   7     # Ay vibration
# sci_microRider_c2_std          60   7
sci_microRider_c4_avg            60   7     # T1
sci_microRider_c5_std            60   7     # T1_dT1
sci_microRider_c6_avg            60   7     # T2
sci_microRider_c7_std            60   7     # T2_dT2
sci_microRider_c8_std            60   7     # S1
sci_microRider_c9_std            60   7     # S2
```

Example NBD:

```
# MicroRider output sensors, listed in PRIORITY order
sci_microRider_last
sci_microRider_next
sci_microRider_pressure
sci_microRider_file_number
sci_microRider_c5_avg
sci_microRider_c5_std
sci_microRider_c7_avg
sci_microRider_c7_std
sci_microRider_c8_std
sci_microRider_c9_std
sci_microRider_c11_avg
sci_microRider_c11_std

#sci_microRider_c0_avg            0  # Gnd
#sci_microRider_c0_std            0
#sci_microRider_c1_avg            0  # Ax vibration
#sci_microRider_c1_std            0
#sci_microRider_c2_avg            0  # Ay vibration
#sci_microRider_c2_std            0
#sci_microRider_c3_avg            0
#sci_microRider_c3_std            0
#sci_microRider_c4_avg            0  # T1 w/o pre-emphasis
#sci_microRider_c4_std            0
#sci_microRider_c6_avg            0  # T2 w/o
#sci_microRider_c6_std            0
#sci_microRider_c8_avg            0  # S1
#sci_microRider_c9_avg            0  # S2
#sci_microRider_c10_avg           0  # P w/o
#sci_microRider_c10_std           0
#sci_microRider_c12_avg           0
#sci_microRider_c12_std           0
#sci_microRider_c15_avg           0 added in 8.2
```

```
#sci_microRider_c15_std          0 added in 8.2
#sci_microRider_c32_avg          0  # V_bat voltage
#sci_microRider_c32_std          0
#sci_microRider_c40_avg          0  # Inclinometer Y
#sci_microRider_c40_std          0
#sci_microRider_c41_avg          0  # X
#sci_microRider_c41_std          0
#sci_microRider_c42_avg          0
#sci_microRider_c42_std          0
#sci_microRider_c144_avg         0  added in 8.2
#sci_microRider_c144_std         0  added in 8.2
#sci_microRider_timestamp        0 # secs since 1970
#sci_microRider_file_offset      0 # byte offset of the device data in the *.mrd file
```

3 Remote Troubleshooting Using *Calibrate All*

The *calibrate all* function is a useful system check of the MircoRider. The *calibrate all* function is not a calibration; it is used during the electronics calibration procedure at Rockland Scientific. The command produces statistics for each channel on the MicroRider which can be interpreted to assess the health of the MicroRider. An RS-232 serial connection between the glider and the MicroRider is required to perform *calibrate all* remotely via the glider. Using calibrate all is highly recommended as a systems check for gliders that are not using the MicroRider Proglet. To run the *calibrate all* command, begin by establishing a serial connection between the glider science computer and the MicroRider using the `u4stalk` command.

3.1 `U4STALK`

From the science computer, use the `u4stalk` command to establish serial communication between the glider and the MicroRider. The syntax for the `u4stalk` command is as follows:

```
u4stalk uart baud_rate escape_character power_bit
```

The UART and power_bit can be found in the `proglets.dat` file. Baud_rate comes from the sensor manufacturer; for the MicroRider it is always 9600. Escape_character is your choice, ctrl+escape_character will end a `u4stalk` session and return you to the glider science computer. An example of the command used for the MicroRider described in section 2.3 is:

```
u4stalk 4 9600 c 0
```

Remember, you can use `u4stalk` to communicate with other science sensors on the glider.

Once you have established serial communication with the MicroRider you will need to stop the 10 second data acquisition countdown by using the command `Q`. You are now communicating directly with the MicroRider Persistor. The *calibrate all* command is as follows:

```
odas5ir -c all
```

The results from *calibrate all* are statistics from one second of data from each MicroRider channel taken sequentially. Units are in raw counts.

3.2 Examples of *Calibrate All* Results

Below are examples of *calibrate all* results from a glider on deck with probes installed, a glider on the surface with probes installed, a glider on the surface after suffering damage to all four microstructure probes. Full discussion of how to interpret the *calibrate all* statistics are not included in this document. Glider MicroRider users are welcome to request further interpretation by Rockland staff by sending results to support@rocklandscientific.com

On Deck Before Deployment: All channels are reporting healthy results given the local conditions. Real probes are installed. Note that the shear (ch 8, ch 9) and temperature probe (ch 4-7) channels have low min and max values.

```
ch:   0  min:     +0  max:     +2  mean:     +0.8  stdev:     0.43
ch:   1  min:  -4529  max:  +4255  mean:    -11.6  stdev:  1824.11
ch:   2  min:  -2862  max:  +2525  mean:     -6.4  stdev:   968.94
ch:   4  min:  +1733  max:  +1816  mean:  +1784.1  stdev:    23.78
ch:   5  min:  +1773  max:  +1984  mean:  +1891.0  stdev:    54.75
ch:   6  min:  +3577  max:  +3624  mean:  +3606.9  stdev:    14.56
ch:   7  min:  +3359  max:  +3789  mean:  +3622.7  stdev:    87.35
ch:   8  min:   -747  max:   +733  mean:     -3.5  stdev:   297.96
ch:   9  min:  -1432  max:  +1436  mean:     -2.7  stdev:   533.39
ch:  10  min:    -59  max:    -56  mean:    -57.1  stdev:     0.40
ch:  11  min:    -59  max:    -53  mean:    -56.8  stdev:     1.27
ch:  12  min:  -1755  max:  -1754  mean:  -1754.7  stdev:     0.44
ch:  32  min: +17199  max: +17546  mean: +17466.8  stdev:    90.65
ch:  40  min: -32768  max: -16385  mean: -27934.8  stdev:  7476.71
ch:  41  min: -16509  max: -16494  mean: -16502.6  stdev:     4.23
ch:  42  min: -31463  max: -31462  mean: -31462.9  stdev:     0.27
ch: 255  min: +32752  max: +32752  mean: +32752.0  stdev:     0.00
```

On the surface a few days after deployment: Shear channel min and max are higher due to local surface condition. Remember that shear probes are vibration sensors and can reach very high raw counts in rough surface conditions. It is not possible to fully assess health of shear probe channels when the glider is on the surface.

```
ch:   0  min:     +0  max:     +2  mean:     +0.7  stdev:     0.45
ch:   1  min:  -7477  max:  +7291  mean:   -219.0  stdev:  1187.45
ch:   2  min:  -1808  max:  +1749  mean:    -92.4  stdev:   475.84
ch:   4  min:   +637  max:   +652  mean:   +644.0  stdev:     4.71
ch:   5  min:   +506  max:   +761  mean:   +662.6  stdev:    42.63
ch:   6  min:  +2450  max:  +2463  mean:  +2456.1  stdev:     3.89
ch:   7  min:  +2363  max:  +2678  mean:  +2482.9  stdev:    37.79
ch:   8  min:  -4957  max:  +6899  mean:    -14.6  stdev:  1642.71
ch:   9  min:  -3573  max:  +6394  mean:    -56.8  stdev:  1259.54
ch:   10  min:    -56  max:    -53  mean:    -54.8  stdev:     0.73
ch:  11  min:    -99  max:    -55  mean:    -70.6  stdev:     9.15
ch:  12  min:  -1871  max:  -1868  mean:  -1869.6  stdev:     0.49
ch:  32  min: +17020  max: +17257  mean: +17157.5  stdev:    86.48
ch:  40  min: -32574  max: -32502  mean: -32545.1  stdev:    26.16
ch:  41  min: -18828  max: -18557  mean: -18667.7  stdev:    82.57
ch:  42  min: -31455  max: -31455  mean: -31455.0  stdev:     0.0
ch: 255  min: +32752  max: +32752  mean: +32752.0  stdev:     0.00
```

On the surface after damage to all four microstructure probes: Note the high min and max values in the shear and temperature probe channels. In some cases these values may slowly increase over a number of days indicating minor damage that allows water to slowly ingress into the tip of a microstructure probe.

```
ch:   0  min:     +0  max:     +1  mean:     +0.8  stdev:     0.42
ch:   1  min:   -466  max:   +336  mean:    -42.4  stdev:   146.83
ch:   2  min:   -274  max:   +391  mean:    +35.6  stdev:   104.24
ch:   4  min:   +602  max:  +8239  mean:  +5077.3  stdev:  2126.21
ch:   5  min: -32212  max: +32767  mean:   +813.8  stdev: 24494.84
ch:   6  min: +18392  max: +22723  mean: +20113.3  stdev:  1325.76
ch:   7  min: -32128  max: +32767  mean: +11801.5  stdev: 24875.95
ch:   8  min: -15654  max: +32767  mean:    -95.5  stdev:  6832.56
ch:   9  min: -14848  max: +32767  mean:    -35.6  stdev:  7129.56
ch:  10  min:    -49  max:    -47  mean:    -47.8  stdev:     0.43
ch:  11  min:   -106  max:    -32  mean:    -65.9  stdev:    20.52
ch:  12  min:  -1969  max:  -1738  mean:  -1803.4  stdev:    25.03
ch:  32  min: +16828  max: +17321  mean: +17118.8  stdev:   182.82
ch:  40  min: -31858  max: -31625  mean: -31746.0  stdev:    75.62
ch:  41  min: -31242  max: -30702  mean: -31108.4  stdev:   139.15
ch:  42  min: -31458  max: -31458  mean: -31458.0  stdev:     0.00
ch: 255  min: +32752  max: +32752  mean: +32752.0  stdev:     0.00
```

4 Channel Reference

Channels that may appear on a MicroRider-1000. For a list of channels specific to your instrument check the instrument `setup.cfg` file. Software releases 8.0 and 8.1 do not support the two EM Current Meter channels however these will be supported in 8.2. Micro Conductivity channels are not supported as they are not recommended for use on gliders. More information available in the ODAS5IR User Guide.

Ch#	Name	Signal
0	gnd	pseudo-ground
1	Ax	piezo-accelerometer 1
2	Ay	piezo-accelerometer 2
4	T1	Thermistor 1
5	T1_dT1	Thermistor 1 with pre-emphasis
6	T2	Thermistor 2
7	T2_dT2	Thermistor 2 with pre-emphasis
8	sh1	Shear Probe 1
9	sh2	Shear Probe 2
10	P	Pressure
11	P_dP	Pressure with pre-emphasis
12	PV	Voltage on pressure transducer
15	EMC_Cur	EMC drive current reference signal for EM noise removal
32	V_bat	Power Supply Voltage divided by 10
40	Incl_Y	y-inclinometer (pitch)
41	Incl_X	x-inclinometer (roll)
42	Incl_T	Inclinometer temperature
64	C	Micro Conductivity, not recommended for glider
65	C_dC	Micro Conductivity with pre-emphasis, not recommended for glider
144	U_EM	JAC Electromagnetic Velocity Digital output
255	Sp_char	Special Character, 7FF0H

-End of document-