

# OpenWater Interface Control Document

June 2, 2021

## BLUETOOTH LOW ENERGY INTERFACE

The open water electronics use a “Bluno” microprocessor board for the data collection and communication either with the attached LCD display or via Bluetooth.

The BLE service is part of the bluno firmware and I have not done anything to change it. The service has a UUID of: E6FBF347-C779-AE6F-E2DE-FCE9C0C1D49F and the characteristic **dbf1** is used to communicate with the device. The following values are sent to command the device or request data from the device followed by a carriage return (0x0d). All data is currently returned as ascii with a carriage return and linefeed after the data:

Value Sent	Description	Data Returned
0x00	Collect Data	None
0x01	Last Reading Mean	Mean FNU X.XX
0x02	Last Reading StDev	StDev FNU X.XX
0x03	Number of Averages	XX
0x04	Integration Time	XX.X
0x05	Calibration Slope	XXX.XX
0x06	Calibration Intercept	XXX.XX
0x07	Last Reading Mean Counts	Mean Cnts XXXX
0x08	Last Reading StDev Counts	StDev Cnts X.XX
0x09	All Parameters	See Below
0x10	Read Stored Calibration Date	Unix Epoch Time
0x11	Write Calibration Date	None – see below
0x12	Initiate Calibration	None – use LCD
0x13	Read Calibration Human Readable	
0x14	Read IsMeasuring	0, 1

The all parameters command above returns the following string in comma separated decimal numbers with a carriage Return and Line Feed at the end:

<Last Reading Mean>, <Last Reading StDev>, <Number of Averages>, <Integration Time>, <Calibration Slope>, <Calibration Intercept>, <Mean Counts>, <StDev Counts>, <Calibration Date>

Writing the calibration time and asking for a calibration commands are as follows:

<command byte><date string><0d>

For example, to write a calibration date to the EEPROM you would send this following command for a date of 12345678:

0x11 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x0d

To request a calibration via the BLE interface, you would send the following (assuming the same date):

0x12 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x0d

The calibration is then completed using the LCD display.

NOTE: The code does a rudimentary check on the date string where if the string supplied is earlier than the calibration date stored in the EEPROM, the code will ignore the earlier date and take the EEPROM date and add one second to it and store that new value.

NOTE: If a calibration is initiated using the LCD and front panel buttons, the user is prompted to enter the calibration date and time and the verification described above is performed.

NOTE: The “Write Calibration Date” function (0x11) does no error checking of the date value, so you can reset the date to any value with that function. Use this power wisely.