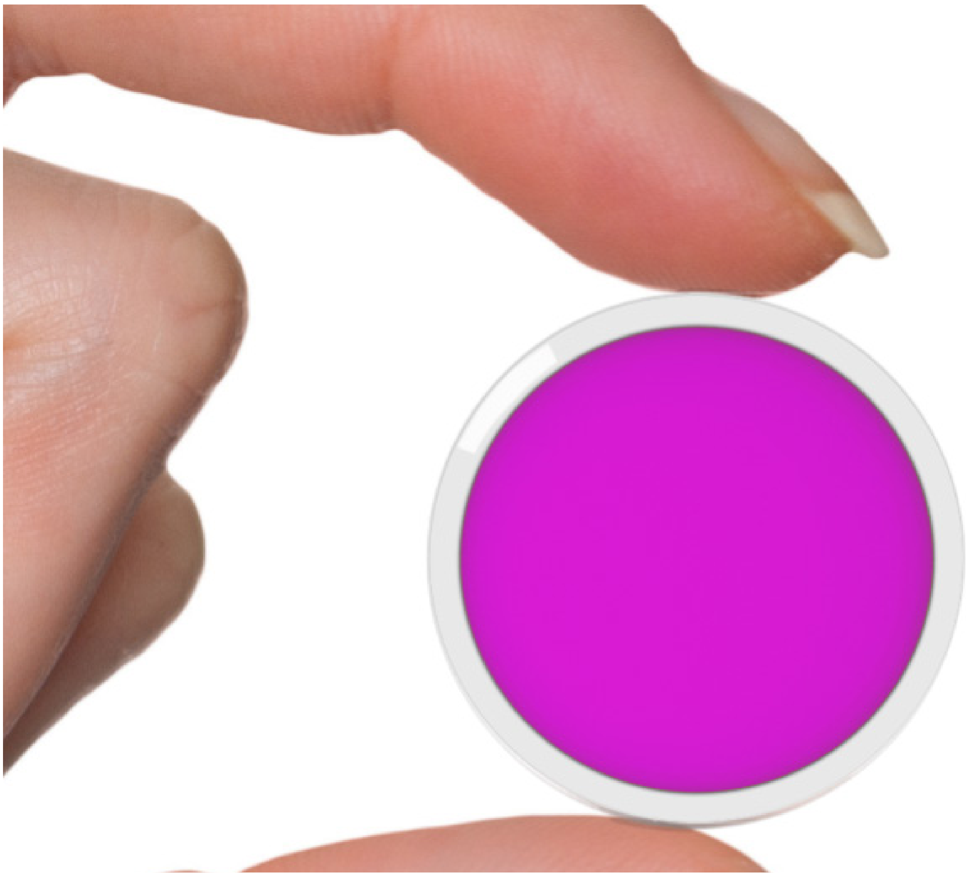
# BLUE MAESTRO BLUETOOTH SENSOR AND LOGGER

# API AND COMMAND BOOKLET



Version 2..5
Apri 2024

BLUE MAESTRO'S STANDARD LICENSE AND DISCLAIMER

No claims, representations or warranties, whether expressed or implied, including but not limited to warranties of merchantability, fitness for a particular purpose, of title, or of noninfringement of third party rights, are made by Blue Maestro as to the safety, reliability, durability or performance of Blue Maestro's products. Blue Maestro is not responsible for any liabilities resulting from negligence, misuse, modification, or alterations to the product by the user. Furthermore, Blue Maestro accepts no liability whatsoever for the safety, reliability, durability or performance of any of its products. IN NO EVENT, REGARDLESS OF CAUSE, SHALL BLUE MAESTRO BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES OF ANY KIND, WHETHER ARISING UNDER BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), STRICT LIABILITY OR OTHERWISE, AND WHETHER BASED ON THIS AGREEMENT OR OTHERWISE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

User assumes responsibility for correct operation of the product and any software associated with it. User assumes responsibility for determining the suitability of the product to the user's needs, for configuring and using the product to meet those needs, and for the proper placement/location of the product in the environment it is being used. User assumes responsibility for verifying and interpreting results obtained from product use.

BLUE MAESTRO RESERVES THE RIGHT TO REVISE ITS STANDARD LICENSE AND DISCLAIMER WITHOUT NOTICE AT ANY TIME.

## Introduction

This document sets out the specifications, API and the commands of the Blue Maestro Temperature & Humidity Sensor and Data Logger (the "Device").

The Device uses the nRF52 Bluetooth Low Energy System on Chip from Nordic Semiconductor and a high quality temperature and humidity sensor from Silicon Laboratories. The Device utilises Bluetooth 4.3.

The Device adopts "ad-hoc" pairing where as much information is passed from Device to receiver (smartphone, iPad, Bluetooth enabled computer or Gateway) without pairing/bonding being required as possible. This is done through encoding information within the advertisement packets of the Device. Pairing is reserved for the passing of commands from receiver to Device or for the extraction of data from the Device where this cannot be achieved in the advertisement packet (such as logged data, for example).

There has been a great deal of flexibility introduced into the Device to allow settings to be changed. However care must be taken to ensure that settings are changed correctly. For example if certain undefined/out of scope variables are input into settings the Device may not perform as desired and, possibly, lock-up such it may only be able to be recovered by removing and then applying power.

There is a button and LED in the Device. The button is a software button that will cause the Device to consume an ever so small amount of current, even when turned off. To maximise battery life in transportation and storage it is suggest that the battery is removed/not placed and is only inserted when the Device is to be deployed.

## Advertisement Data Structure

The following represents the advertisement packet for the sensor. The first 30 bytes is the standard Advertisement payload and the second 30 bytes is the Scan Response payload. They are represented in bytes with an offset beginning with Blue Maestro's Bluetooth SIG company identifier 0x33 0x01. This is usually located at the 16th byte in the full observed advertisement packet. The data structure is little endian.

Time interval for logging uint16_t

Current Temp Int16_t

Current Dew Point Int16_6

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x33 | 0x01 | 0x17 | 0x64 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |

Blue Maestro's SIG identifier 2 bytes

Version No. uint8_t

Battery lvl in % of voltage uint8_t

Stored log count uint16_t

Current Humidity Int16_t

Mode Int8_t

Breach Count uint8_t

Length of Name

| 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x61 | 0x09 | 0x08 | 0xXX | 0xXX | 0xXX | 0xXX | 0xXX | 0xXX | 0xXX | 0xXX | 0x00 | 0x00 | 0x00 | 0x00 |

Name Identifier 0x09

Name of Device 8 bytes

Reserved 4 bytes

Highest Temperature Recorded int16_t

Lowest Temperature Recorded int16_t

Highest Temperature over last 24 hours - int16_t

Lowest Temperature over last 24 hours int16_t

| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x33 | 0x01 | 0x74 | 0x53 | 0x61 | 0x66 | 0x65 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |

Blue Maestro's SIG identifier 2 bytes

Highest Humidity Recorded int16_t

Lowest Humidity Recorded int16_t

Highest Humidity over last 24 hours - int16_t

Lowest Humidity over last 24 hours - int16_t

Average Humidity over last 24 hours - int16_t

Reference Date uint32_t

| 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x61 | 0x69 | 0x74 | 0x53 | 0x61 | 0x66 | 0x65 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |

Average Temperature over last 24 hours - int16_t

Global Identifier uint8_t

## 1. UART over BLE Communication

The following are a list of the command line syntax and field requirements to allow settings in the Device to be changed.

The application includes one service: the Nordic UART Service. The 128-bit vendor-specific UUID of the Nordic UART Service is 6E400001-B5A3-F393-E0A9-E50E24DCCA9E (16-bit offset: 0x0001).

This service exposes two characteristics: one for receiving and one for transmitting (as seen from the nRF5 application).

TX Characteristic (UUID: 6E400002-B5A3-F393-E0A9-E50E24DCCA9E)
If the peer has enabled notifications for the TX Characteristic, the application can send data to the peer as notifications. The application will transmit all data received over UART as notifications.

RX Characteristic (UUID: 6E400003-B5A3-F393-E0A9-E50E24DCCA9E)
The peer can send data to the device by writing to the RX Characteristic of the service. ATT Write Request or ATT Write Command can be used. The received data is sent on the UART interface.

All commands and responses are ascii characters unless otherwise stated. The parameter fields of commands and responses are indicated by 'X'. It is important that the correct number of fields in the parameters are complied with.

If a command is not recognised, the message "Unknown Command" will be transmitted. If a command is recognised a "Command Recognised" will be transmitted. However even though the command is recognised the parameter may still be incorrect/out of bounds and the Device will not pass any indication.

Where a command is recognised the Device will disconnect and recommence advertising after processing the command. Where a password is entered incorrectly the Device will disconnect and recommence advertising.

## 2. Transmission Strength [ *txpX ]

This sets the transmission strength of the radio from a choice of three settings of -4dBm, 0dBm and +4dBm. This can be useful for extending the battery life of the Device but has a negative impact on range. The default setting is +4dBm.

Parameters:

| | |
|---|---|
| 000 | sets to +4dBm |
| 001 | sets to 0dBm |
| 003 | sets to -4dBm |

## 3. Logging Interval [*lintX]

This sets the logging interval in seconds. The logging interval represents how frequently the sensor values are stored in memory. The default is 3600 seconds which represents one hour. When changed, the current recorded data will be erased and logging will commence afresh. The minimum is 60 seconds and the maximum is 43,200 seconds (or 12 hours).

Parameters: x in seconds (default is 3600 = 1 hourly logging interval)

## 4. Sensor Interval [*sintX]

This sets the sensor interval in seconds. The sensor interval represents how fequently the sensors take readings. The advertisement packets of the device are updated with the latest sensor readings. The defaut is 10 seconds. The minimum is 2 seconds and the maximum is 43,200 seconds (or 12 hours).

Parameters: x in seconds (default is 3600 = 1 hourly logging interval)

## 5. Password Protect [*pwdXXXX]

This locks the Device from receiving commands or having its log accessed. The Device is not password protected on shipping. Entering a password will lock the Device until the password is re-entered whereby the Device will be unlocked until a password is again entered.

Parameters: 4 characters or numbers.

## 6. Change Name [*namXXXXXXXX]

This changes the Device name from its default. Please note that for some models of smartphone (both iOS and Android) the name change will not necessarily be recognised until Bluetooth is turned off and back on in the smartphone settings.

Parameters: 8 characters

## 7. Advertising Timeout [*atoXXX]

This sets the timeout parameters where advertising will cease after a button push. In all of these cases advertising will commence in “fast mode” for 180 seconds and then in “slow mode” for a further number of seconds as defined in this command. The default is 300 seconds or 5 minutes. It is important to note that without advertising, the Device is not able to be connected to per Bluetooth 4.x protocols.

Parameters: XXX in seconds

## 8. Information [*info]

This gives you a list of the current setting including the devices name , version number etc

Parameters: None

## 9. Telemetrics [*tell]

This gives you a list of the devices telemetrics including sensor frequency, log frequency, number of records, current temperature and the last 24hours metrics and other key data.

Parameters: None

## 10. Units of Measure [*unitsf] [*unitsc]

This will convert temperature measurements into either fahrenheit or celsius.

Parameters: None

## 11. Battery level [*batt]

This gives you the battery level as a percentage of voltage level. As the Device is powered by a CR2032 coin cell, the voltage begins at 3.00 volts (100%). As the coin cell loses its charge, the voltage will drop. It is adviseable to change that the coin cell once the voltage drops to 2.1 volts (70%).

## 12. Alarm one and two [*alrmUYZXXX]

There are two alarms which track how many times a sensor reading goes below or above a set threshold. The occurrence of these alarms form part of the advertisement packet and when scanned by a smartphone or other receiving device can set off an “alarm”. The Device itself does not alarm.

The alarms can be set for temperature, humidity or dew point. Since there are two, you can set them for any combination of these.

Parameters: U is the alarm number, either 1 or 2. So *alrm1t>25 sets alarm 1 for temperature over a threshold of 25 Celsius (if this is the units sets for the Device at the time)

Y is the sensor which the threshold applies, either t for temperature, h for humidity or d for dew point. For example *alrm2h>50 sets alarm 2 for humidity over a threshold of 50% relative humidity.

XXX is the threshold to apply. It is in round numbers, so no decimal points can be specified. For example *alrm2d<10 sets alarm 2 for dew point below 10 Celsius (if this is the units set for the Device at the time).

## 13. Clear alarms [*alrmclr]

This clears the settings for the alarms.

Parameters: None

## 14. Alarm information [*alrmi]

This provides informations on the alarms, such as settings and how many times the sensor has breached any thresholds.

Parameters: None

## 15. Air mode on [*airon]

This sets the air mode on. This causes the Device to log information but not transmit, ideal for transportation by air. Once this command is entered, the Device will continue to transmit for a further 5 minutes before ceasing. Depressing the Device's button will cause the Device to transmit for a further 5 minutes.

Parameters: None

## 16. Air mode off [*airoff]

This sets the air mode off and the Device will continue to continuously transmit as normal.

Parameters: None

## 17. Download log in readable form [*loggert] [*loggerh] [*loggerd]

Either of these commands will download the relevant log for temperature, humidity or dew point for analysis. Note using these commands does not store the data in the smartphone, for this use the specific download button on the smartphone.

## 18. Disable Button [*bd]

This command will prevent the button from turning the device on or off. It will still cause the LED to blink and will cause the device to transmit if it is in airplane mode. This command works on a toggle basis, to enable the button if it has been disabled pass the command [*bd] again.

## 19. Calibrate Temperature [*ctXX]

This command will apply an offset to temperature readings to the value of XX. This needs to be integers (no decimal points) and can be negative. This value needs to conform to the units of measure currently selected for the device.

Parameters: Integer (can be negative)

## 20. Calibrate Humidity [*chXX]

This command will apply an offset to humidity readings to the value of XX. This needs to be integers (no decimal points) and can be negative.

Parameters: Integer (can be negative)

## 21. Reference Date & Time [*dXXXXXXXXXX]

This command sets the reference date and time for timestamping the logs. This date and time should represent when the device was first activated since the device immediately records a log. The reference date and time can be entered at any time and can be changed subsequently.

The format is a 10 digit number representing Year, Month, Day, Hour, Minute. No seconds are referenced.

For example 11:05pm 23 March 2017 would be represented as 1703232305. To set the date and time to this the command would be *d1703232305. This value is part of the advertisement packet from the device.

Parameters: 10 digit number represeting xx[year]xx[month]xx[day]xx[hour]xx[minute].

## 22. Global Identifier [*idX]

This provides a uint8_t number (0-255) that will be advertised in the Devices advertising packets for identification, classification or grouping purposes. This can be useful in scanning for Devices where a filter can be applied based on the global identifier so that only certain devices are picked up in the app.

Parameters: Integer (can be negative)

## 23. Clear Memory [*clr]

This command will erase the stored logs in the device. However it does not affect the logging interval, name, global identifier, reference date or the like.

## 24. Reset to Factory Settings [*rboot]

This command will perform a full reboot of the device and will clear all memory and settings.

END OF DOCUMENT