

NSRTW_mk4_MQTT

MQTT Protocol

June 2 2023

Bruno Paillard

1 REVISION HISTORY 2

2 INTRODUCTION 3

3 ENDIANNESS 3

4 BASIC TYPES 3

5 CONNECTION 3

6 MESSAGE FORMATS 4

6.1 Topics 4

6.2 Message types 4

6.3 Published messages 5
6.3.1 Vital Signs 5
6.3.2 Recorded Data 6

6.4 Subscribed Messages 8
6.4.1 Settings 8

1 Revision History

1. May 10 2023 Initial version of this document.

2. May 12 2023

Modification
Modified the description of the 4th byte of the *Model/Format* field to indicate the firmware version of the instrument.

3. May 12 2023

Modification
Added remote reset and remote disable commands.

4. May 24 2023

Modification
Introduced second mode of operation with more traditional topics specific to each type of message.
Removed remote reset and remote disable.

1. June 8 2023

Modification
Added note that published messages have their retain flag at 1.

2 Introduction

The *NSRTW_mk4_MQTT* is a new variant of the *NSRT_mk4* series that introduces an MQTT communication protocol. That means that the instrument can communicate with MQTT brokers and report to MQTT-based platforms.

3 Endianness

Unless otherwise noted, the endianness is Little-Endian.

4 Basic Types

The following basic types may be used in this protocol:

Type Name	Description	Endianness
U8	Single byte unsigned	N/A
U16	16-bit word unsigned	Little-Endian
U32	32-bit word unsigned	Little-Endian
U64	64-bit word unsigned	Little-Endian
I8	Single byte signed	N/A
I16	16-bit word signed	Little-Endian
I32	32-bit word signed	Little-Endian
I64	64-bit word signed	Little-Endian
Sgl	32-bit word in IEEE 754 floating point format	Little-Endian
Dbl	64-bit word in IEEE 754 floating point format	Little-Endian
String	Strings are concatenations of 8-bit ASCII characters, terminated by an end-of-text (0x00) byte.	N/A

Table 1

5 Connection

The instrument always connects to the broker with a clean session. It keeps track of what data has already been published, so it will only publish whatever data has not yet been acknowledged by the broker.

The instrument always publishes and subscribes with a QoS of 1. In case the broker has received some of the recorded data, but has failed to acknowledge it, the same data will be sent again (duplicated) at the next connection opportunity, but that should not matter because it will be the same data already published. The data includes a UTC of the first data point in the message. That should serve to localize the data in time.

If the instrument loses power or is manually reset, it will lose track of what it has already published. In that case it will proceed to re-publish all the data in its memory. Again, that should not matter because the data is published together with its corresponding UTCs that will serve to localize it in time.

Since every session is clean, it will re-subscribe to all subscribed messages. That includes the settings. This is done in the first few seconds of the connection. At that time the broker should send the current settings, which must be published with a *retain* flag at 1.

The instrument will only reprogram the settings and restart a recording if the new settings are different than the settings already in effect. If they are identical to the settings in effect no action will be taken.

6 Message formats

6.1 Topics

The *NSRTW_mk4_MQTT* has two MQTT modes of operation:

1. **Forced Topics**: Pub and Sub topics can be defined by the user. In this mode there is only one Pub topic and only one Sub topic for all types of messages. Both the Pub and Sub topic strings can be defined freely and written into Flash by the user using *Instrument Manager*. This is to comply with some MQTT broker platforms that impose the topic strings to the client. In this case, the user must rely on the first 8 bytes of the message content to determine the precise type of message being transmitted/received.

2. **Standard Topics**: Pub and Sub topics are defined by the instrument. In this (more conventional) mode, the Pub and Sub topic strings are defined for each type of message, and indicate explicitly that type. There are as many topic strings as there are types of messages. The topic string for each type of message is indicated in the document below. This mode is selected when the *Forced* Pub and/or Sub topic strings are left empty. In this case the first 8 bytes of the message content still indicate the precise type of the message being transmitted/received, but these 8 bytes can be ignored/zero-padded, since they are redundant with the information indicated by the topic.

Note that the mode of operation can be mixed (*Forced* on the Pub topics and *Standard* on the Sub topics, or vice-versa).

6.2 Message types

In both modes of operation, the beginning of every message is comprised of two U32 numbers that indicate the message type and structure:

- ***Model/Format*:** This field indicates:
    - **Model**: The model of the instrument is indicated in the 3 lower bytes. For the NSRTW_mk4_MQTT instrument, those three bytes are 0x34, 0x53, 0x4E
    - **FW Version**: The firmware version is indicated by the upper byte. The major revision is indicated by the upper 4 bits, and the minor version is indicated by the lower 4 bits. At the time of this writing, the firmware version is 1.2. So the upper byte is 0x12

Implicitly those 4 bytes indicate the nature of the instrument that is connecting to the broker, and how the rest of the message should be interpreted. For the NSRTW_mk4 FW 1.2, the value of that field is equal to 0x1234534E.

- ***Type*:** This indicates the type of message that is published.

The value of *Type* completely describes how the rest of the message can be interpreted.

For the *NSRTW_mk4_MQTT*, the *Type* field can have the following values:

Type (Hex)	Function
0x0000000A	Vital Signs
0x0000000B	Levels (Lmax)
0x0000000C	Levels (LEQ)
0x0000000D	Levels (Lmin)
0x0000000E	Levels (Lpeak)
0x0000000F	Settings
0x00000010	Remote Reset
0x00000011	Remote Disable

Table 2

Note: On messages that the instrument publishes, when the mode of operation is *Standard Topics*, those two U32 numbers can be ignored by the recipient since the same information is provided explicitly in the topic.

Note: On messages that the instrument subscribes to, when the mode of operation is *Standard Topics*, those two U32 numbers can be set to any value. The instrument ignores them and relies on the received topic to determine the nature of the message that is being received.

6.3 Published messages

6.3.1 Vital Signs

Topic (*Standard* mode): Inst_Class/*Model*/FW*xx*/*Client_ID*/Vitals

- **Inst_Class**: "NS" for Noise Sentry series
- ***Model***: NSRTW_mk4_MQTT for this instrument.
- ***FWxx***: Two-digit number that indicates the firmware revision (for instance "FW12" indicates firmware version 1.2)
- ***Client_ID***: Client ID string as programmed by the user. This defaults to the serial number of the instrument if the string is empty.

This is a message that is published every time the instrument connects. It contains all the vital signs of the instrument, as well as the instrument UTC and clock error.

This message is published with a retain flag at 1.

Field	Type	Size (bytes)	Value	Function	Scale
Model/Format	U32	4	*NSRTW_mk4->* 0xMm34534E *VSEW_mk4->* 0xMm345356	4 bytes that identify the model and firmware revision. "M" symbolizes the Major rev number. "m" symbolizes the minor rev number. Presently 2 models are recognized: NS4 *NSRTW_mk4* VS4 *VSEW_mk4*	
Type	U32	4	0x0000000A	Indicates a message of type "Vital Signs"	
UTC[1]	U64	8		$UTC_{instrument}$. This represents the instrument's internal clock	seconds
UTC_err	I32	4		This represents the difference between a reference UTC from an SNTP server, and the instrument UTC: $UTC_{SNTP} - UTC_{instrument}$. It is only updated when the SNTP time is accessible. Otherwise, the last updated value is transmitted.	seconds
Batt	Sgl	4		Battery voltage	Volt
Temp	Sgl	4		Temperature	degC
RSSI	Sgl	4		WiFi signal strength	dBm

Table 3

6.3.1.1 Use of the UTC_Err value

The instrument will attempt to connect to an SNTP server and get a precise UTC every time it connects to the internet. That exact UTC is used to reset the internal instrument clock when a recording is started. But the instrument will not set its internal clock while it is recording to avoid any discontinuity. Since a recording can last for several weeks, the drift of its internal clock can grow to large values. Instead, the instrument will transmit the difference between the SNTP's reference UTC and its internal clock's UTC at the time of the connection. This way the graphs can be precisely time-aligned if required.

6.3.2 Recorded Data

Topic (*Standard* mode): Inst_Class/*Model*/*FWxx*/*Client_ID*/*Lxx*

- **Inst_Class**: "NS" for Noise Sentry series
- ***Model***: NSRTW_mk4_MQTT for this instrument.
- ***FWxx***: Two-digit number that indicates the firmware revision (for instance "FW12" indicates firmware version 1.2)

[1] UTCs (Universal Time Codes) are referenced to January 1st 1904. To get a Unix UTC, subtract 2082844800 seconds.

- ***Client_ID***: Client ID string as programmed by the user. This defaults to the serial number of the instrument if the string is empty.
- ***Lxx***: Indicates the type of value transmitted (Lmin, Lmax, LEQ or Lpeak)

These messages are published every time the instrument connects, if some of the values recorded have not yet been sent to the broker. Each message contains successive recorded values of the specified type, as well as the fractional UTC of the first sample in the message and settings of the whole recording. Each message contains N successive recorded values of the specified type (Lmin, Lmax… etc.).

These messages are sent in groups, each group encompassing all the types present in the recording. Several groups can be sent in succession, as required to upload all the data remaining in the instrument's memory that has not been uploaded yet.

There is a maximum of 512 values (1024 bytes) in each message. Messages are sent in succession, as long as there is data to be transmitted in the instrument's memory. Messages for each type of value are interlaced. For instance, the following messages can be sent in succession for a recording containing 700 values of Lmax, LEQ and Lpeak:

[Lmax – LEQ – Lpk] – [Lmax – LEQ – Lpeak]

The messages in the first block of *[Lmax – LEQ – Lpk]* contain 512 values each, and the messages in the second block contain 188 values each.

These messages are published with a retain flag at 1.

Field	Type	Size (Bytes)	Value	Function	Scale
Model/Format	U32	4	*NSRTW_mk4->* 0xMm34534E *VSEW_mk4->* 0xMm345356	4 bytes that identify the model and firmware revision. "M" symbolizes the major rev number. "m" symbolizes the minor rev number. Presently 2 models are recognized: NS4 *NSRTW_mk4* VS4 *VSEW_mk4*	
Type	U32	4	0x0000000B 0x0000000C 0x0000000D 0x0000000E	Indicates a message of type "Lmax" Indicates a message of type "LEQ" Indicates a message of type "Lmin" Indicates a message of type "Lpeak"	
f_UTC	U64	8		This represents the fractional UTC of the first data point in the message	1/8 s (125 ms)

Interval	U16	2		This represents the time between two successive samples in the data stream that follows	1/8 s (125 ms)
Fs	U16	2	32000 or 48000	Sampling Frequency	Hz
Weighting	U16	2	- dBC: 0 - dBA: 1 - dBZ: 2	Frequency weighting curve:	
Tau	Sgl	4		Time constant	seconds
N_Values	U32	4		Number of values in the subsequent array. A value of 0 is valid. That means an empty array (nothing following *N_Values*).	
$Value_1$	I16	2		Level value in 1/10 dB	centi-Bel (1/10 dB)
$Value_2$	I16	2		Level value in 1/10 dB	centi-Bel (1/10 dB)
…	I16	2		Level value in 1/10 dB	centi-Bel (1/10 dB)
$Value_{N_Values}$	I16	2		Level value in 1/10 dB	centi-Bel (1/10 dB)

Table 4

6.4 Subscribed Messages

Since the instrument opens a clean session every time it connects, it will re-subscribe to all subscribed messages whenever it connects.

6.4.1 Settings

Topic (*Standard* mode): Inst_Class/*Model*/*FWxx*/*Client_ID*/Settings

- **Inst_Class**: "NS" for Noise Sentry series
- ***Model***: NSRTW_mk4_MQTT for this instrument.
- ***FWxx***: Two-digit number that indicates the firmware revision for which the message is published (for instance "FW12" indicates version 1.2).
- ***Client_ID***: Client ID string as programmed by the user. This defaults to the serial number of the instrument if the string is empty.

Note: In *Standard Topics* mode the instrument will not accept any *Settings* message that is published with a FW value that is different from its actual firmware revision. For instance, an instrument that has a firmware version of 1.1 will not accept a *Settings* message published with *FW10* FW value. This behavior is slightly different from the behavior in *Forced Topics* mode, where the instrument will accept any *Settings* message published for a firmware revision number *Mm* lower or equal to the instrument's actual firmware revision (see below).

Field	Type	Size (bytes)	Value	Function	Scale
Model/Format	U32	4	*NSRTW_mk4->* 0xMm34534E *VSEW_mk4->* 0xMm345356	4 bytes that identify the model and firmware revision. “M” symbolizes the major rev number. “m” symbolizes the minor rev number. The revision number indicates the minimum revision number that should be accepted by the instrument (i.e. an instrument which firmware revision is lower than this number should refuse the settings because there is a chance that some of the settings or format are unknown to that firmware). Presently 2 models are recognized: NS4 *NSRTW_mk4* VS4 *VSEW_mk4* *In the Standard Topic mode of operation, these 4 bytes are “don’t care”. The information is taken from the Sub topic.*	
Type	U32	4	0x0000000F	Indicates a message of type “Settings”	
Timezone	I32	4		This represents the time zone the instrument is in. For instance, GMT-4 is represented as -14400	Seconds
Manifest	U16	2		This word represents which values should be recorded: Bit 0: Lmax Bit 1: LEQ Bit 2: Lmin Bit 3: Lpk	
Interval	U16	2		This represents the time between two successive samples in the data stream that follows	1/8 s (125 ms)
Fs	U16	2	32000 or 48000	Sampling Frequency	Hz
Weighting	U16	2	- dBC: 0 - dBA: 1 - dBZ: 2	Frequency weighting curve	

Tau	Sgl	4		Time constant. For instance 0.125 for Fast, 1.0 for slow.	Seconds

Table 5