SENSO8 Distance Sensor API document

Table of Contents

1. Uplink Data . . . 2
1.1. Data Reporting . . . 2
1.1.1. MQTT Topic Format . . . 2
1.1.2. MQTT Message Payload Format . . . 2
1.2. Device Configuration Reporting . . . 3
1.2.1. MQTT Topic Format . . . 3
1.2.2. MQTT Message Payload Format . . . 4
2. Downlink Command . . . 5
2.1. MQTT Client Setting . . . 6
2.2. Downlink Data Format . . . 6
2.2.1. Distance Threshold Configuration . . . 6
2.2.1.1. MQTT Topic Format . . . 6
2.2.1.2. MQTT Message Payload Format . . . 6
2.2.2. Data Sampling Period Configuration . . . 6
2.2.2.1. MQTT Topic Format . . . 6
2.2.2.2. MQTT Message Payload Format . . . 6
2.2.3. Data Upload Period Configuration . . . 6
2.2.3.1. MQTT Topic Format . . . 6
2.2.3.2. MQTT Message Payload Format . . . 7
2.2.4. SYS Message upload period Configuration . . . 7
2.2.4.1. MQTT Topic Format . . . 7
2.2.4.2. MQTT Message Payload Format . . . 7
2.3. Downlink Data Format using encryption . . . 7
2.3.1. Data Format before Encryption . . . 8
2.3.1.1. Data Payload Format . . . 8
2.3.2. Encrypted Data Format . . . 8
2.3.2.1. MQTT Topic Format . . . 8
2.3.2.2. MQTT Message Payload Format . . . 9
2.3.2.3. Nonce Format . . . 9
2.3.2.4. Token Format . . . 9
2.3.3. Is it safe after encryption? . . . 10
2.3.4. Limitations . . . 10
3. Revision History . . . 10

The API document describes the format of downlink and uplink JSON data for the following models:

- SEN20LD0
- SEN21LD0

Payload templates use unquoted placeholder names to describe field values. Replace each placeholder with a value of the type specified in the field table before sending JSON.

1. Uplink Data

1.1. Data Reporting

1.1.1. MQTT Topic Format

SENSO8/nbiot/data/<IMEI>

1.1.2. MQTT Message Payload Format

```
{
  "ver":protocol_version,
  "msg":msg_seq_no,
  "type":sensor_type,
  "time":epoch_time,
  "sn":sensor_imei,
  "dist":sensor_distance,
  "state":occupancy_state,
  "bat":battery,
  "RAT":radio_type,
  "rssi":radio_rssi,
  "rsrp":radio_rsrp,
  "rsrq":radio_rsrq,
  "ci":radio_ci,
  "tac":radio_tac,
  "low_batt":low_batt,
  "event":event_type,
  "debug":debug_info
}
```

Field	*Description*
protocol_version	Protocol version (int)
msg_seq_no	Message sequence number (uint32)
sensor_type	Sensor type = 13
epoch_time	Unix epoch time in ms
sensor_imei	IMEI of the sensor (string)
sensor_distance	Distance detected by sensor (int, mm)
occupancy_state	Occupancy state 0 – vacant (distance > threshold) 1 – occupied (distance <= threshold)
battery	Battery voltage (mV) (invalid when using AC adaptor)
radio_type	Radio Type, e.g. LTE, 3G, …
radio_rssi	Radio RSSI (int)
radio_rsrp	Radio RSRP (int)
radio_rsrq	Radio RSRQ (int)
radio_ci	Radio cell information (8-digit hex string)
radio_tac	Radio tracking area code (4-digit hex string)
low_batt	0 – battery normal; 1- battery low
event_type	Data event type 0 – data update/trigger 1 – user-initiated (by button press) 2 – heartbeat 4 - user-initiated (by button press) & heartbeat
debug_info	Reserved

1.2. Device Configuration Reporting

1.2.1. MQTT Topic Format

SENSO8/nbiot/sys/<IMEI>

1.2.2. MQTT Message Payload Format

```
{
  "ver":protocol_version,
  "sn":sensor_imei,
  "iccid":sim_iccid,
  "dist_th":[distance_threshold, 0],
  "sampling":[sampling_batt, sampling_dc],
  "upload":[upload_batt, upload_dc],
  "low_batt":low_batt_alert_lvl,
  "batt_pwr_alert":batt_pwr_alert,
  "reg_timeout":registration_timeout,
  "band":[freq_band, freq_band, ...],
  "report":report_interval,
  "local_port":local_tcp_port,
  "lfreq":lfreq,
  "overwrite":overwrite,
  "model":model_no,
  "psm":[psm_enable, active_period, TAU_period],
  "reg_timeout":reg_timeout,
  "en_shutdown":enable_shutdown,
  "secure":secure_mode,
  "firmware":firmware_version,
  "client_prefix":mqtt_client_id_prefix,
  "topic_prefix":mqtt_topic_prefix
}
```

Field	***Description***
protocol_version	Protocol version (int)
sensor_imei	IMEI of the sensor (string)
sim_iccid	ICCID of the SIM card (string)
distance_threshold	Distance threshold for occupancy (mm)
sampling_batt	Data sampling period (sec) when powered by battery
sampling_dc	Data sampling period (sec) when powered by DC adaptor (Reserved)
upload_batt	Data upload period (sec) when powered by battery
upload_dc	Data upload period (sec) when powered by DC adaptor (Reserved)
low_batt_alert_lvl	Battery low alert threshold (mV)
batt_pwr_alert	Reserved, always 1

Field	*Description*
registration_timeout	Mobile network registration timeout (sec)
freq_band	NB-IoT frequency band number (int) “band” can be 0 only for automatic band selection or any combinations of 1, 3, 5, 8 and 20.
report_interval	periodic SYS message upload interval (hour) 0 – No periodic SYS message upload X - periodic upload SYS message every X hours
reg_timeout	Timeout for registering to the NB-IoT network (sec)
local_tcp_port	0 – TCP local port is assigned by NB-IoT module 1 – TCP local port is assigned by MCU (recommended)
psm_enable	0 – PSM disable 1 – PSM enable
active_period	PSM Active Timer (T3324) in seconds
TAU_period	PSM Periodic TAU (T3412) in seconds
enable_shutdown	Reserved
lfreq	Reserved
overwrite	Reserved
model_no	Sensor model number
secure_mode	0 – no encryption in downlink 1 – enable encryption in downlink See Downlink Data Format using encryption
firmware_version	Firmware version (string)
client_prefix	MQTT client ID prefix (string)
topic_prefix	MQTT topic prefix (string)

2. Downlink Command

2.1. MQTT Client Setting

The following setting is required for MQTT client posting downlink data to SENSO8 sensor.

- Publish QoS = 1
- Retain Flag = False

2.2. Downlink Data Format

2.2.1. Distance Threshold Configuration

2.2.1.1. MQTT Topic Format

SENSO8/nbiot/ctrl/<IMEI>

2.2.1.2. MQTT Message Payload Format

```
{
  "sn":sensor_imei,
  "dist_th":[distance_threshold, 0]
}
```

2.2.2. Data Sampling Period Configuration

2.2.2.1. MQTT Topic Format

SENSO8/nbiot/ctrl/<IMEI>

2.2.2.2. MQTT Message Payload Format

```
{
  "sn":sensor_imei,
  "sampling":[sampling_batt, sampling_dc]
}
```

2.2.3. Data Upload Period Configuration

2.2.3.1. MQTT Topic Format

SENSO8/nbiot/ctrl/<IMEI>

2.2.3.2. MQTT Message Payload Format

```
{
  "sn":sensor_imei,
  "upload":[upload_batt, upload_dc]
}
```

2.2.4. SYS Message upload period Configuration

2.2.4.1. MQTT Topic Format

SENSO8/nbiot/ctrl/<IMEI>

2.2.4.2. MQTT Message Payload Format

```
{
  "sn":sensor_imei,
  "report":report_interval
}
```

Field	*Description*	*Type*	*Default*
epoch	Unix time since 1970 Jan 1 00:00:00 UTC	number	-
distance_threshold	Distance threshold for occupancy	number	800
sampling_batt	Data sampling period (sec) when battery power (min: 180, max: 43200)	number	3600
sampling_dc	Data sampling period (sec) when DC adaptor power (min: 180, max: 43200)	number	900
upload_batt	Data upload period (sec) when battery power (min: 180, max: 43200)	number	3600
upload_dc	Data upload period (sec) when DC adaptor power (min: 180, max: 43200)	number	900
report_interval	Periodic SYS message upload interval (hours)	number	24

Note: After issuing a downlink configuration command, the device will publish a Device Configuration Report showing the latest device configuration.

2.3. Downlink Data Format using encryption

To prevent an intruder from changing the sensor configuration, downlink data encryption can be

enabled (`secure_mode` 1).

The encryption algorithm is AES192 with a secret key pre-configured in the device. When data payload encryption is enabled, downlink data must be encrypted with the same secret key stored in the device. Otherwise the device will not be able to decrypt the payload and will ignore the command.

The default secret key is an internal device ID (not IMEI) which is 192 bits long and is unique among all SENSO8 devices. The server should maintain a mapping between the device IMEI and the secret key for this data encryption implementation. Sharing the same secret key among devices is a security risk and is NOT recommended.

2.3.1. Data Format before Encryption

2.3.1.1. Data Payload Format

```
{
  "epoch":epoch,
  key:value
}
```

Field	*Description*	*Type*	*Default*
epoch	Unix time in seconds since 1970 Jan 1 00:00:00 UTC	number	-
key	Downlink command key in JSON string as described in Downlink Data Format	string	-
value	Value for key in JSON string as described in Downlink Data Format	-	-

"epoch" or Unix timestamp is the number of seconds elapsed from 1970 Jan 1 00:00:00 UTC till now. It is a mandatory field when secure mode is enabled. Simply use the current UTC time for the "epoch" value. SENSO8 stores the previous downlink epoch and compares it with the epoch in the current downlink. If the epoch in the current downlink is <= the previous epoch (re-use), SENSO8 will ignore this downlink request.

"epoch" MUST always be the first key in the JSON string.

2.3.2. Encrypted Data Format

2.3.2.1. MQTT Topic Format

SENSO8/nbiot/ctrl/<IMEI>

2.3.2.2. MQTT Message Payload Format

```
{
  "sn":sensor_imei,
  "nonce":nonce,
  "token":token
}
```

2.3.2.3. Nonce Format

AES192 requires an initialization vector (IV) to randomize its generated output so that the encrypted data always looks different even if the encrypted content is the same. For the AES192 algorithm, the IV is a 16-byte stream. The IV should be randomly generated for each encryption. It is sent together with the encrypted content to the peer side. Only the peer having the same AES192 secret key can decrypt the content correctly with the IV.

"nonce" is simply a JSON string which is the **AES192 IV** encoded in base64 format.

Example:

16-byte IV (HEX):

30 31 32 33 34 35 36 37 38 39 41 42 43 44 45 46

Base64 encoded IV:

MDEyMzQ1Njc4OUFCQ0RFRg==

Nonce

"nonce":"MDEyMzQ1Njc4OUFCQ0RFRg=="

2.3.2.4. Token Format

"token" is a JSON string generated from the **AES-192 encrypted payload** encoded in base64 format. The payload is a downlink command in the format described in Data Format before Encryption.

Example:

Encrypted Payload (HEX):

30 31 32 33 34 35 36 37 38 39 41 42 43 44 45 46

Base64 encoded output:

MDEyMzQ1Njc4OUFCQ0RFRg==

token

"token":"MDEyMzQ1Njc4OUFCQ0RFRg=="

2.3.3. Is it safe after encryption?

1. Can an intruder determine the AES192 secret key given an IV, unencrypted input and encrypted output?

   Practically no. Even with a supercomputer, brute force attacks still take years to crack an AES192 key.

2. Is it safe if an intruder re-sends the downlink to the same device?

   Safe. SENSO8 always checks the epoch in the downlink and if it finds that the epoch is re-used or invalid, this downlink will be ignored. The epoch is encrypted within the payload. An intruder is not able to modify the epoch value without knowing the secret key.

3. Is it safe if an intruder replicates the downlink to other devices?

   Conditional. Provided that every SENSO8 device has its own secret key, it is safe, because the decrypted content will not form a JSON if a wrong secret key is used. Other devices using different secret keys cannot decrypt it. However, if all devices share the same secret key, an intruder can replicate the downlink command from one device and control other devices.

2.3.4. Limitations

The maximum string length of the unencrypted MQTT CTRL payload is limited to 200 bytes.

3. Revision History

Revision	*Date*	*Description*
1.0	11th Nov 2022	Initial Release.
1.1	24th May 2024	Added SEN21LD0 to supported models.
2.0	Oct 2026	First release on docs.senso8.com