SARA-N3 series

Application development guide

Application note

Abstract

This document provides the technology architecture and AT command examples showing how to use AT commands with u-blox NB-IoT SARA-N3 series modules.

www.u-blox.com

Document information

Title	SARA-N3 series	
Subtitle	Application development guide	
Document type	Application note	
Document number	UBX-19026709	
Revision and date	R03	15-Oct-2021
Disclosure restriction	C1-Public	

This document applies to the following products:

Product name
SARA-N310

Contents

1 Introduction

This document provides guidance when developing applications for NB-IoT. It includes examples of using AT commands to communicate with the u-blox NB-IoT modules. See the SARA-N2 / SARA-N3 series AT commands manual [2] for detailed AT command descriptions.

The following symbols are used to highlight important information within this document:

☞ An index finger points out key information pertaining to module integration and performance.

⚠ A warning symbol indicates actions that could negatively impact or damage the module.

2 NB-IoT technology overview

NB-IoT technology is designed such that it can be used in areas beyond the radio coverage of current cellular standards and in devices that must run from battery power for many years. The devices will generally send small amounts of data infrequently; a typical usage scenario might be 100 to 200 bytes sent twice per day for battery powered devices. For mains powered devices, the limit is not based on battery size, but cost and network bandwidth/resources.

The system operation is analogous to that of SMS, in that it is a datagram-oriented, stored-and-forward system, rather than a GPRS-like IP pipe. This is because NB-IoT devices spend most of their time asleep, making possible the required long battery life. The system implements extended DRX cycles for paging, but as this window will be limited to save battery life, the delivery of downlink messages occurs mainly when the system detects that uplink messages have been received from a device (indicating that it is awake). Here a store-and-forward system, an “IoT Platform”, is useful.

A simplified system is represented in Figure 1.

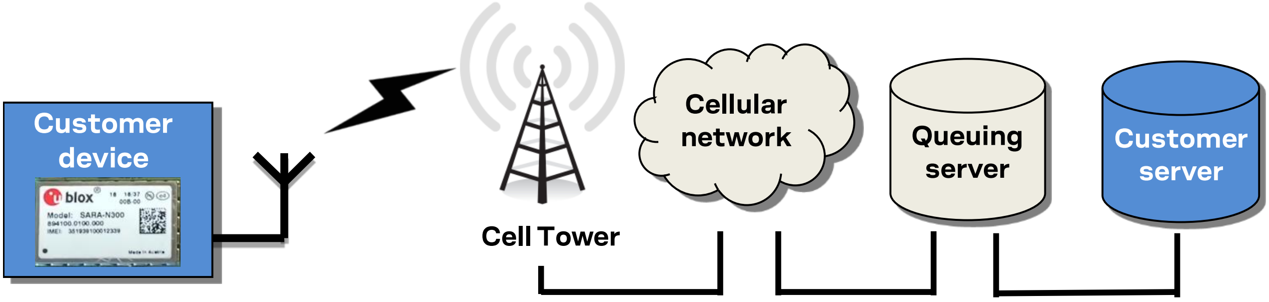


Figure 1: NB-IoT system architecture

At the far left, the customer’s device contains a u-blox NB-IoT module that communicates over the radio network with a cell tower that supports the NB-IoT network. The cellular network links the cell tower with an IoT platform. This IoT platform stores uplink datagrams from the NB-IoT module. The customer server communicates with the IoT platform to retrieve uplink datagrams and to send downlink datagrams to the NB-IoT. The IoT platform holds downlink datagrams until the NB-IoT module is awake to receive them.

The SARA-N3 series modules implement several communication protocols – CoAP, MQTT, MQTT-SN, LwM2M, HTTP, FTP – as well as UDP and TCP sockets.

3 AT command response parser

This section gives some hints about how to develop an AT parser and how to handle the responses to the AT commands and the URCs (unsolicited result code).

In this document the following naming conventions are used:

- DCE (Data Communications Equipment) or MT (Mobile Terminal) is the u-blox NB-IoT module
- DTE (Data Terminal Equipment) or TE (Terminal Equipment) is the terminal that sends the command to the module

When entering AT commands, spaces are ignored. The DCE uses carriage-return line-feed pairs (`\r\n, 0x0D0A`) to end lines on its output. The same termination is required on input to the DCE.

When the DCE has finished processing a command it will output a final result code (either "OK" or "ERROR") indicating that it is ready to accept a new command. The information text responses are issued before the final result code.

3.1 Unsolicited result code

An unsolicited result code (URC) is a string message (provided by the DCE) that is not a response to a previous AT command. It can be output, when enabled, at any time to inform the DTE of a specific event or status change. The implemented URCs are as follows:

`+CIEV: <descr>,<value>`	Mobile termination event reporting
`+CALV: <n>`	Alarm
`+CTZV: <tz>[,time]`	Time zone reporting
`+CREG: <stat>[,<lac>,<ci>[,<AcTStatus>]]`	CS domain network registration
`+CEREG: <stat>[,<tac>,<ci>,<AcT>]`	EPS domain network registration
`+UJAD: <active>`	Jamming detection
`+CSCON: <n>[,<mode>]`	MT connection mode

`+CEDRXP: <AcT_type>[,<Requested_eDRX_cycle>`
`[,<Assigned_eDRX_cycle>`
`[,<Assigned_paging_time_window>]]]`
	eDRX parameters changed

`+CCIOTOPTI: <supported_Network_opt>`	CIoT optimization configuration
`+CRTDCP: <cid>,<cpdata_length>,<cpdata>`	Terminating CP data reporting
`+UUPSDA: <result>[,<ip_addr>]`	Packet switched data action
`+CGEV: <text>[,param1[,param2]…`	GPRS event reporting
`+UULGASP <result>,<bearer>`	Last gasp message
`+NPSMR: <mode>`	Power mode status
`+UUDNSRN: <result_code>[,<resolved_ip_address>]`	Domain name resolving
`+UUDYNDNS: <status>,<code>`	Dynamic DNS
`+UUSOCL: <socket>`	Socket close
`+UUSOCO: <socket>,<erorr>`	Socket connect
`+UUSOST: <socket>,<seq_no>,<UDP_result>`	Socket send to
`+UUSORF: <socket>,<length>`	Socket read from
`+UUSOLI: <socket>,<ip_address>,<port>, …`	Socket listening
`+UUSOAOL, +UUSOAOC`	Socket always on
`+UUFTPCD, +UUFTPCR`	FTP
`+UUHTTPCR: <profile_id>,<command>,<result>`	HTTP
`+UPING: <retry_num>,<remote_address>,<ttl>,<rtt>`	Ping
`+UCOAPCD, +UCOAPCR`	CoAP command result/data
`+UUMQTTC: <op_code>,<param1>[,<param2>…`	MQTT result
`+UUMQTTSNC: <op_code>[,<param1>[,<param2>…`	MQTT-SN result

• `+UULWM2MOPR: <op_result>`	LwM2M operation
• `+UULWM2MDM:<op_code>,<param3>,<param4>…`	LwM2M device manager
• `+UULWM2MIR: <mode>,<ssid>,<res_path>`	LwM2M information report
• `+ULWM2MSTAT: <stat>,<param>`	LwM2M FOTA

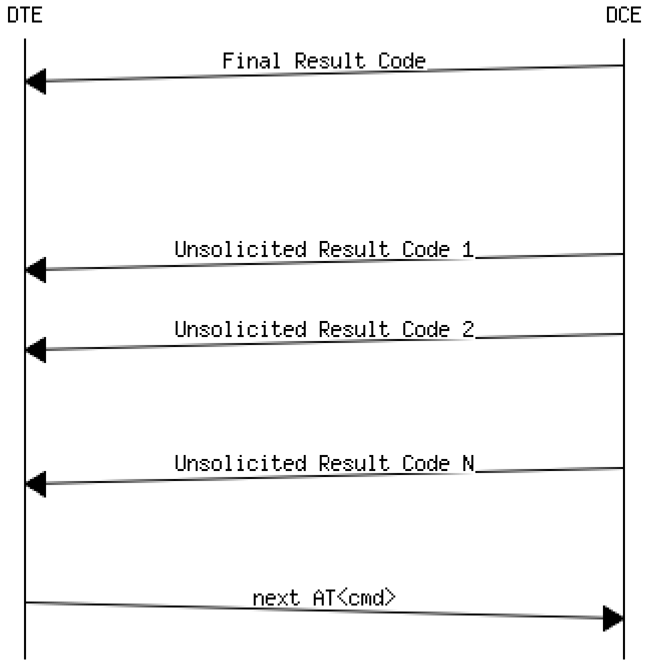

DTE
DCE
Final Result Code
Unsolicited Result Code 1
Unsolicited Result Code 2
Unsolicited Result Code N
next AT<cmd>

Figure 2: DTE-DCE URC flow chart

3.2 Best practices

- The DTE shall flush the AT channel (i.e. check if there is data waiting to be read) before sending a new AT command.
- The DTE shall handle the case of unexpected spaces or line endings.
- The DTE shall handle all the URCs: it can simply ignore them (not suggested) or, better, take a proper action.
- The DTE shall know what answer is expected and shall wait until it is received (i.e. final result code only or information text response with the final result code).
- The final result code marks the end of an AT command and can be OK or an error. When the final result is an error, be sure to handle it before continuing with the next AT command.
- The information text response format is command specific. The DTE will need explicit handling for each one. It is suggested to consult the SARA-N2 / SARA-N3 series AT command manual [2].
- It is suggested to not strictly parse information text responses but rather to check if they contain interesting keywords and/or parameters.
- It is very useful, for debugging an application, to log all the command lines sent to the DCE and received from it.
- Create a state machine for the AT parser (e.g. idle, waiting_response, data_mode).
- The DTE shall wait some time (the recommended value is at least 20 ms) after the reception of an AT command final response or URC before issuing a new AT command to give the module the opportunity to transmit the buffered URCs. Otherwise the collision of the URCs with the subsequent AT command is possible.

4 SARA-N3 series power mode setting

4.1 Power mode states

There are four power mode states for the SARA-N3 series modules.

State	Description	AT Interface	PSRAM	Wake up
PM0	Active mode	On	On	N/A – Always on
PM1	Idle mode	On	On	Wake up via AT interface
PM2	Sleep mode	Off	On	PWR_ON pin
PM3	Deep-sleep mode	Off	Off	PWR_ON pin

The module cannot be directly set into these modes, but it is possible to set how they are used. The `+NVSETPM` AT command is for configuring how the power modes are used.

- In the PM0 state the module is always on and nothing is shut down. The module will need to be in this state for EasyFlash to work.
- In the PM1 state the module is always on, but the AT interface UART is turned off after 6 s. To wake up the AT interface, the host application needs to send an extra character.
- In the PM2 state the module will go into sleep mode, in which the internal memory is still kept and the host application will need to wake the module up using the PWR_ON pin.
- In the PM3 state the module will go into a deep-sleep mode, in which the internal memory is not kept and the host application will need to wake the module up using the PWR_ON pin. When the module is using PM3 state with PSM, the network registration information is kept so that it does not need to re-register.

4.2 Power mode setting +NVSETPM

The `+NVSETPM` AT command controls how the PMx power mode states are entered. Depending on AT idle time, eDRX and PSM timers, the PMx power mode states will be entered. The host application can change the `+NVSETPM` setting at any time.

4.2.1 Disable entering power save modes (+NVSETPM: 0)

To disable the use of the power modes and have a module that is always on set `+NVSETPM: 0`.

☞ Use the `AT+NVSETPM=0` command when upgrading the firmware. The firmware cannot be upgraded if the `+NVSETPM` AT command is not set to 0.

4.2.2 Enter PM1 whenever it can (+NVSETPM: 1)

The default setting is `+NVSETPM: 1` which goes into idle mode (PM1) power mode whenever it can. When the module is in idle mode (PM1) mode it will need to be woken up with an extra character being sent on the AT UART interface.

4.2.3 Deep Sleep after 300 s (+NVSETPM: 2)

To use sleep mode (PM3) for when the module will be sleeping for longer than 300 s, issue the `AT+NVSETPM=2` command. In this case it will use PM1 mode for idling when it is not in PSM or eDRX states, and sleep mode (PM3) for when it is in PSM or eDRX states.

4.2.4 PM2 and PM3 (+NVSETPM: 9)

This will use PM2 mode state when going to sleep between PSM and eDRX cycles less than 300 s. It will use PM3 mode state when sleeping for longer than 300 s.

4.2.5 PM2 and PM1 (+NVSETPM: 10)

This will use PM2 mode state when going to sleep according to PSM and eDRX. It will use PM1 mode state at other times.

4.3 PM2 idle time configuration +NVSETPM2IDLETIME

If the module is configured for the PM2 state, when the module wakes up the PM2 timer will start. If the module has no data to send to the network, then the PM2 idle timer will expire, and it will go back into PM2 power mode state. Even if the host application is still sending AT commands, if there is no attempt to connect to the network, then the PM2 timer will expire.

Use this feature to enable the module to automatically go back to sleep if nothing is happening.

5 Registration on the NB-IoT network

Before the customer's application can send or receive any messages to their cloud server, the module must first register on the NB-IoT network. It is possible to query the registration status using the +CEREG read command, or by configuring the +CEREG URC.

The module will scan the bands that are enabled with the +UBANDSEL AT command.

☞ The +UBANDSEL AT command is different from the SARA-N2 series method of selecting the bands to scan. For more details, see u-blox SARA-N2 / SARA-N3 series AT commands manual [2].

There are a few options for registering the module on a cellular LTE network. The module can automatically perform a cell PLMN search, or the host application can choose the PLMN manually. After registration, the module can automatically perform PDN context activation, depending on the +CIPCA setting.

5.1 Release 13 / Release 14 settings

SARA-N3 series modules are factory programmed with the configuration set to release 13 of the 3GPP NB-IoT specification. To enable the module in release 14 supported features (not all release 14 features are supported – see the u-blox SARA-N3 series data sheet [1]), use the +NVSETRELEASEVERSION AT command.

+NVSETRELEASEVERSION: 0	- release 13 features only
+NVSETRELEASEVERSION: 1	- release 13 features + release 14 supported features

☞ Reset the module with AT+CFUN=16 for the setting to take effect.

5.1.1 Release 14 supported features

Release 14 supported features are provided to the network in the capability Information message:

```
Message: UE Capability Information {
   message c1: ueCapabilityInformation-r13: {
      criticalExtensions ueCapabilityInformation-r13: {
         ue-Capability-r13 {
            accessStratumRelease-r13    rel14
         },
         nonCriticalExtension {
            ue-Capability-ContainerExt-r14 {
               ue-Category-NB-r14 nb2,
               mac-Parameters-r14 {
                  dataInactMon-r14      supported,
                  rai-Support-r14       supported
               },
               phyLayerParameters-v1430 {
                  multiCarrier-NPRACH-r14      supported,
                  twoHARQ-Processes-r14        supported
               }
            }
         }
      }
   }
}
```

5.2 CIoT settings

There are a few optional settings the module can enable/disable that can influence the network connection and the type of services the network provides.

These few settings are controlled via the +CFGCIOT AT command.

5.2.1 User plane (UP) CIoT feature supported and preferred

The `+CFGCIOT` <upciot> parameter allows configuration of S1U data and UP CIoT transports and the preference of use. The factory-programmed configuration is to support them but not to optimize for them. In this case control plane (CP) mode is preferred if the network allows for both.

Some networks may reject the registration of devices that support S1U data/UP CIoT transport. Disable the S1U data/UP CIoT support option on these networks. Unfortunately, there is no 3GPP reject cause (see `+CEREG` URC) for this and the module will simply not register.

5.2.2 Extended protocol configuration options

The ePCO setting, which can be handled by the `+CFGCIOT` <epco> parameter, enables the module to state to the network it supports the extended protocol options. Many cellular networks require ePCO to be enabled before the DNS IP addresses can be provided to the module. On older networks, having ePCO set might not be accepted, therefore the attach process will fail.

5.3 Automatic PLMN registration

SARA-N3 series modules automatically start searching for a NB-IoT network after power-on as `+COPS` <mode> default is 0 (automatic selection mode). After the module has found a cell to connect to, the module will automatically register on to the network.

If the initial PDN context activation is enabled (`+CIPCA: 1,x`), the default bearer PDP context will be assigned by the network.

If the initial PDN context activation is disabled (`+CIPCA: 0,x`), the PDN context will have to be activated by the host application with `+CGACT` AT command.

☞ The host application can force a PLMN search by issuing `AT+COPS=0`, but the AT interface may be blocked for many minutes while the module scans for a network. This differs from the SARA-N2 series, which returns the "OK" final result code immediately after the `+COPS` AT command is issued.

☞ Instead of issuing the `AT+COPS=0` to restart the PLMN search, toggle the module radio state by an `AT+CFUN=0` / `AT+CFUN=1` cycle.

5.4 Manual PLMN registration

The host application can search for a specified PLMN, instead of automatically searching by means of the `+COPS` <mode>=1 (manual mode). This could be used to force the module on to a particular cell if the automatic search picks an unwanted cell for the application.

Set the module to minimum functionality (issuing `AT+CFUN=0`) first to disconnect it from any connected network and to stop it automatically from scanning for a cell.

Issuing `AT+CFUN=1;+COPS=1,2,<oper>` AT commands in one concatenated line will start the module scanning only for this specific MNO network, as specified by the<oper> parameter. Using both commands together will make sure the module does not start an automatic selection mode when the module's radio is enabled with `AT+CFUN=1` on its own.

5.5 Deregistration

SARA-N3 series modules do not support the `+CGATT` AT command. Issue the `AT+COPS=2` command to detach from the network.

☞ The `AT+COPS=2` command also deletes the PDP context that was activated. Re-create the PDP context / APN before the application re-registers.

6 Specifying APN (PDN contexts)

The command to specify the access point name (APN) depends on the +CIPA AT command setting. The +CIPCA AT command setting controls how the module activates the packet data network (PDN) context after registration.

If the initial PDN context activation is enabled (+CIPCA: 1,x), use the +CFGDFTPDN AT command to specify an APN.

If the initial PDN context activation is disabled (+CIPCA: 0,x), use the +CGDCONT AT command to specify an APN.

6.1 Network-provided APN with auto PDN context activation

The example below describes how to allow the network to provide the default APN and automatically activate the PDN context after registration. Note that the +CFGDFTPDN AT command is used because the initial PDP context activation is enabled.

Command	Response	Description
AT+CFUN=0	OK	Set the module to minimum functionality
AT+CIPCA=1	OK	Set auto PDN context activation
AT+CFGDFTPDN=1,""	OK	Set a blank APN
AT+CFUN=1	OK	Set the module to full functionality
	+CSCON: 1	RRC connection established
	+CEREG: 1,"0002","1a2d101",9	The module is registered in NB-IoT
AT+CGDCONT?	+CGDCONT: 1,"IP","internet","132.54.23.3" OK	Query PDN context: APN and IP address are provided

6.2 Specifying APN with auto PDN context activation

The example below describes how to specify the APN and automatically activate the PDN context after registration. Note that the +CFGDFTPDN AT command is used because the initial PDP context activation is enabled.

Command	Response	Description
AT+CFUN=0	OK	Set the module to minimum functionality
AT+CIPCA=1	OK	Set auto PDN context activation
AT+CFGDFTPDN=1,"cIoT"	OK	Set a 'cIOT' as the APN to use
AT+CFUN=1	OK	Set the module to full functionality
	+CSCON: 1	RRC connection established
	+CEREG: 1,"0002","1a2d101",9	The module is registered in NB-IoT
AT+CGDCONT?	+CGDCONT: 1,"IP","cIoT","132.54.45.72" OK	Query PDN context: 'cIoT' APN is set and IP address is provided

6.3 Network-provided APN without auto PDN context activation

The example below describes how to allow the network to provide the default APN with the PDN context automatically activated after registration. Note that the +CGDCONT AT command is used because the initial PDP context activation is disabled.

Command	Response	Description
AT+CFUN=0	OK	Set the module to minimum functionality.
AT+CIPCA=0	OK	Disable auto PDN context activation.
AT+CGDCONT=1,""	OK	Set a blank APN.
AT+CFUN=1	OK	Set the module to full functionality.
	+CSCON: 1	RRC connection established.
	+CEREG: 1,"0002","1a2d101",9	The module is registered in NB-IoT.
AT+CGDCONT?	+CGDCONT: 1,"IP","internet","132.54.23.8" OK	Query PDN context: APN and IP address are provided.

6.4 Specifying APN without auto PDN context activation

The example below describes how to allow the network to provide the default APN with the PDN context being automatically activated after registration. Note that the +CGDCONT AT command is used because the initial PDP context activation is disabled .

Command	Response	Description
AT+CFUN=0	OK	Set the module to minimum functionality.
AT+CIPCA=0	OK	Disable auto PDN context activation.
AT+CGDCONT=1,"cIoT"	OK	Set the 'cIoT' APN.
AT+CFUN=1	OK	Set the module to full functionality.
	+CSCON: 1	RRC connection established.
	+CEREG: 1,"0002","1a2d101",9	The module is registered in NB-IoT.
AT+CGDCONT?	+CGDCONT: 1,"IP","cIoT","0.0.0.0" OK	Query PDN context; only APN is provided. No IP address because the context is not activated.
AT+CGACT=1,1	OK	Activate the PDN context 1.
AT+CGDCONT?	+CGDCONT: 1,"IP","cIoT","132.54.45.28" OK	Query PDN context; IP address is now set.

6.5 Example of wrong APN with auto PDN context activation

The example below shows what error occurs when the APN is incorrectly specified and the auto PDN context activation is enabled.

Command	Response	Description
AT+CFUN=0	OK	Set the module to minimum functionality.
AT+CIPCA=1	OK	Set auto PDN context activation.
AT+CFGDFTPDN=1,"zzzz"	OK	Set the wrong APN.
AT+CFUN=1	OK	Set the module to full functionality.
	+CSCON: 1	RRC connection established.
	+CSCON: 0	RRC connection released.
	+CEREG: 3,"0002","1a2d101",9,0,19	Registration denied, cause 19: ESM procedure failure.
AT+CFGDFTPDN=1,"cIoT"	OK	Set the correct APN.

Command	Response	Description
`AT+CGACT=1,1`	`+CME ERROR: Module not registered`	Try to activate the context, but fails as the module is not registered.
`AT+CFUN=0`	`OK`	Set the module to minimum functionality.
`AT+CFUN=1`	`OK`	Set the module to full functionality, and the module now registers.
	`+CSCON: 1`	
	`+CEREG: 1,"0002","1a2d101",9`	

In the example above the application processor has tried to activate the PDN context with the APN "zzzz" but the network denied this. Because the automatic context activation is configured as part of the registration process, the module is not allowed to continue to be registered on the network.

To correct this error, fix the APN using the `+CFGDFTPDN` AT command and issue an `AT+CFUN=0` / `AT+CFUN=1` cycle (since the `AT+COPS=0` command can block the AT interface for minutes).

6.6 Example of wrong APN without auto PDN context activation

The example below shows how the error occurs when the APN is incorrectly specified and the auto PDN context activation is disabled.

Command	Response	Description
`AT+CFUN=0`	`OK`	Set the module to minimum functionality.
`AT+CIPCA=0`	`OK`	Disable auto PDN context activation.
`AT+CGDCONT=1,"zzzz"`	`OK`	Set the wrong APN.
`AT+CFUN=1`	`OK`	Set the module to minimum functionality.
	`+CSCON: 1`	RRC connection established.
	`+CEREG: 1,"0002","1a2d101",9`	The module is registered in NB-IoT.

`AT+CGDCONT?`	`+CGDCONT: 1,"IP","zzzz","0.0.0.0"`
	`OK`
		Query PDN context: APN looks like it is set, but there is no IP address because the context is not activated.

`AT+CGACT=1,1`	`+CME ERROR: Operation not allowed`	Try to activate PDN context 1, but this fails because the APN is invalid.
`AT+CGDCONT=1,"cIoT"`	`OK`	Set correct APN on context 1.
`AT+CGACT=1,1`	`OK`	Activate the PDN context 1.

`AT+CGDCONT?`	`+CGDCONT: 1,"IP","cIoT","132.54.45.28"`
	`OK`
		Query PDN context: APN IP address are set correctly.

In the scenario above the host application can change the APN by means of the `+CGDCONT` AT command and activate the context with the `AT+CGACT=1,1` command. The main difference between this and the scenario shown in section 6.5 is that the module is kept registered on the network and the host application can simply use the `+CGACT` AT command to activate the context.

7 Base station scanning

When the module starts the registration process it must first scan for a base station based on the PLMN of the SIM. This scanning process differs depending on if there is a previous stored PLMN, the SIM is for the home network, or if it is a roaming/international SIM.

7.1 Previous PLMN

If the module has previously camped on a cell, the PLMN will have been stored in the SIM. It will try to use the previous PLMN when the module first turns on. If this fails it will then try the Home PLMN.

7.2 Home PLMN

The module will try to camp on the HPLMN stored in the SIM. In the quick cell search after the module power-on, once the PLMN is found it will start to register on that cell. If this fails it will then try the roaming processes.

7.3 Roaming / International SIMs

The module has an option to disable roaming. The `+CFGCIOT` AT command can be used to disable/enable roaming. The roaming is enabled by default.

When using a Roaming SIM the module always scan for other networks when powered on. The module does not immediately connect to the previous cell.

8 RRC connection & Release assistance

8.1 RRC connection

When the module needs to communicate to the network, it performs a Random Access CHannel (RACH) procedure to attach to the base station. This establishes a Radio Resource Control (RRC) connection to the base station. Once established only the base station can release this connection. The module cannot drop the RRC connection other than turning off the radio using the `AT+CFUN=0` command.

The base station has an "inactivity" timer for each module and if there is no activity the base station will send a RRC release message to the module. The module should respond back to the base station with an acknowledgment. The inactivity timer is nominally 6-10 s.

The `+CSCON` AT command provides the RRC connection status, either by polling or URC. The host application can use this indication to know when the module is connected and disconnected to the base station. Only when the RRC connection is active can the module send and receive data from the network.

8.2 Current drawn in RRC connected mode

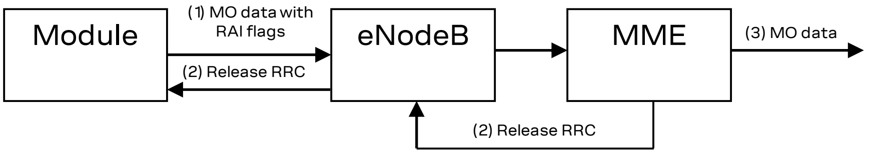
When the module is in RRC connected mode, it will be receiving all the base station signaling. The average power consumed in this mode is about 48 mA. If the RRC connection is left for 10 s of inactivity before the RRC is released, then this will consume about 1 mWh @ 3.6V.

8.3 Release assistance

Some applications may not want to wait for the base station's inactivity timer to expire as this wastes battery power. In 3GPP release 13 and onwards the "Release Assistance" feature allows the module to request for the RRC connection to be dropped as soon as the message has been received by the network. Another option is to drop the RRC connection when the downlink response is returned.

See the `+NSOSTF` AT command to configure the Release Assistance feature.

For 3GPP Release 13, the RAI flag is noticed by the MME on the network and sends a message back to the eNodeB base station to drop the RRC connection. The network must support Release Assistance for this feature.



☞ In 3GPP release 14, RAI is optimized to not require the MME messaging. Enable release-14 for AS-RAI support using the `+NVSETRELEASEVERSION` AT command.

8.4 After RRC is released

After the RRC connection has been released the module then goes into a period where it could be paged by the base station. The timer for this period is called T3324. After T3324 has expired the module goes into Power Save Mode (PSM). In most networks T3324 timer can be set to zero to immediately go into PSM.

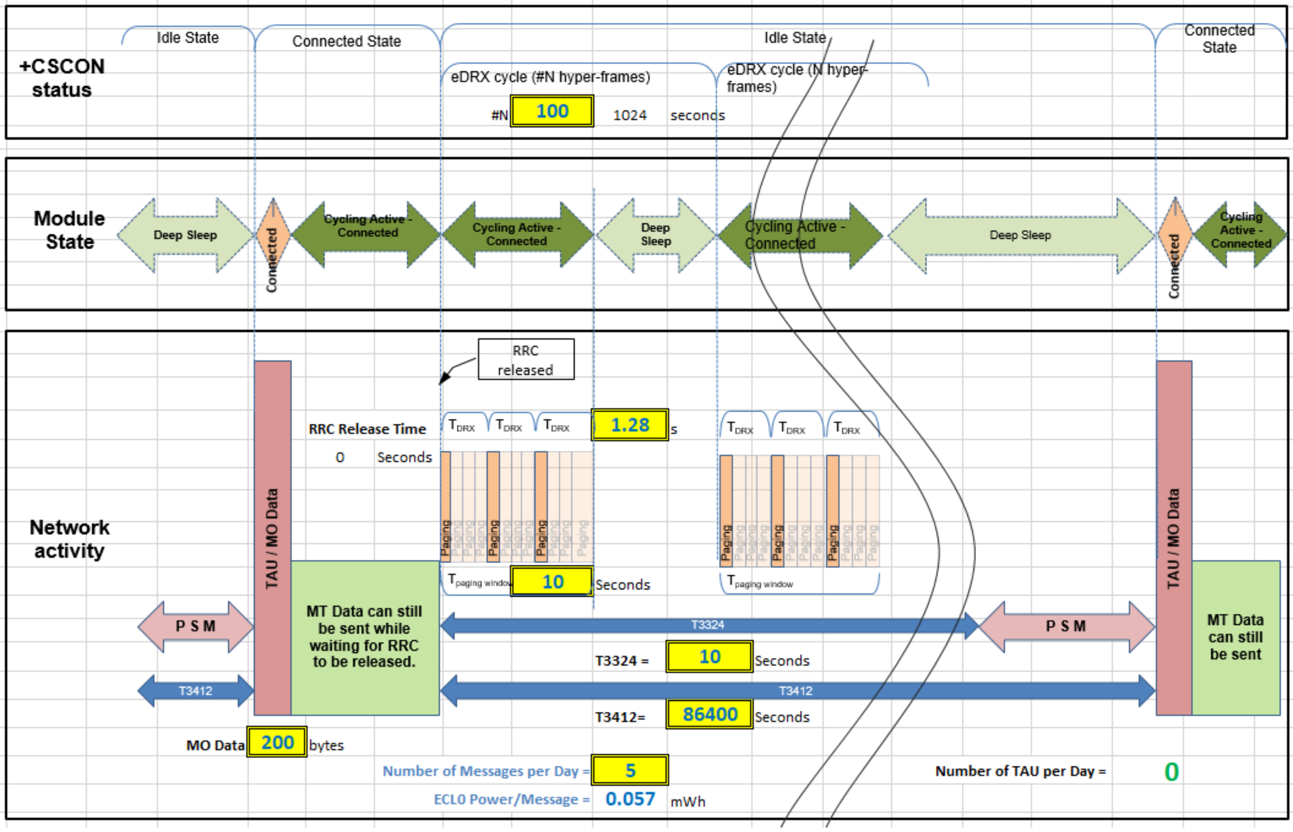
9 Paging, eDRX, PSM and deep-sleep mode

The NB-IoT protocol allows for power save mode (PSM), and the SARA-N3 series modules also support a deep-sleep mode where the module is running at very low current, ~3 µA. The module automatically enters various states depending on the device activity. Here below are listed the common activities and the various states it will be in after registration.

1. The device is in **power save mode (PSM)** and in **deep-sleep mode**; it is already registered and there is no activity.
2. A message is queued or the TAU timer has elapsed.
3. The device re-connects to the network and sends and receives data. This is in Connected State.
4. The RRC connection is released by the network. It is now in Idle State.
5. Within T3324 timer, the paging happens as per Network configuration.
6. Power Saved Mode is entered after T3324 has elapsed.
7. T3412 elapses and Traffic Area Update is triggered, or application sends more data.

☞ T3324 and T3412 timers are assigned by the network. The application can request to the network what timer values it wants, but the network does not need to grant these values.

Typical connection status:

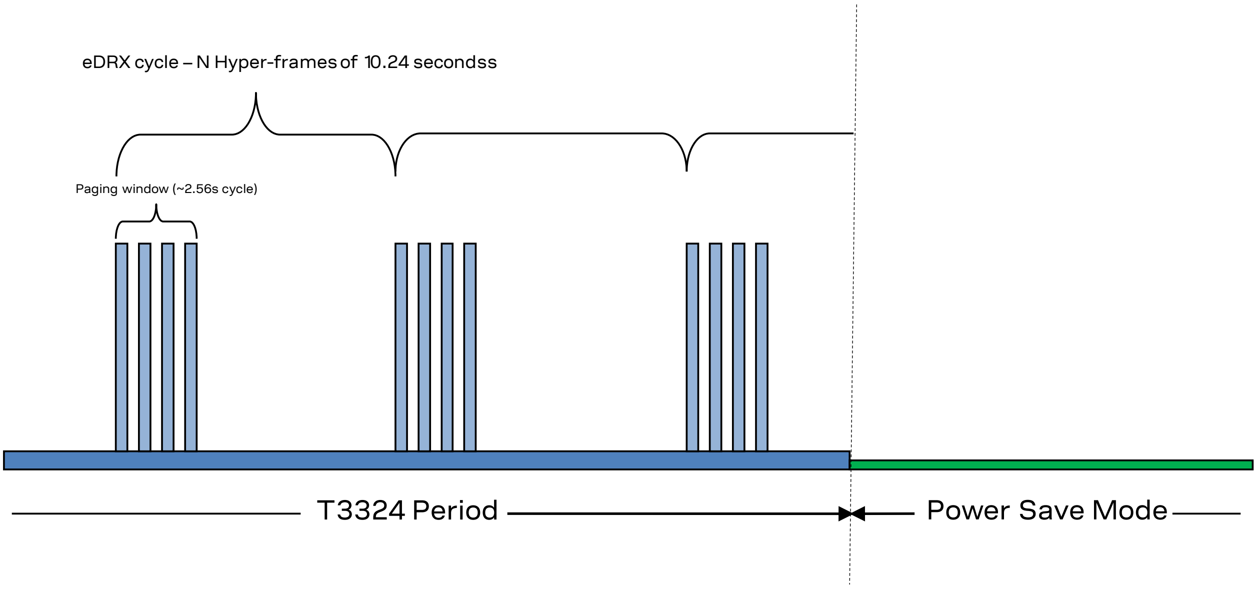


9.1 eDRX

The 3GPP Release 13 introduced a new feature called Extended Discontinuous Reception (eDRX). This allows the paging to only operate for a period and then again sometime later. In between the paging windows the module is in a reduced power mode.

☞ These power mode states can be configured by the +NVSETPM AT command (for more details, see section 5).

eDRX operates while the T3324 timer is running and is specified by several hyper-frames of 10.24 s each.


eDRX cycle – N Hyper-frames of 10.24 secondss
Paging window (~2.56s cycle)
T3324 Period
Power Save Mode

The module can still receive downlink messages when it is in the T3324 period. These messages will be cached by the network knowing the module will awaken for the next paging window after the eDRX cycle has completed.

☞ The application can query the network the eDRX value using the +CEDRXRDP AT command.

9.2 Power Save Mode (PSM)

After the T3324 timer expires the module will enter the Power Save Mode. In this case the module is in deep sleep mode and is only consuming approximately 3 µA.

In the PSM the module is not able to receive any downlink messages. If the customer's cloud application tries to send a downlink message to the module when it is in PSM, the network will most likely ignore it.

Depending on what power mode setting is used with the +NVSETPM AT command, the AT interface may or may not be available without sending one wake-up character or toggling the power online. For further information, see section 5.

10 Monitoring module status

The application can monitor the status of the module's connection, registration and PSM state by polling or configuring URCs. By monitoring the module status, the application can behave more efficiently, depending on the application type. For example, the application may want to know when the module goes into Power Save Mode (PSM).

10.1 Registration status

Register the module to the NB-IoT network before to send or receive any messages. Without being registered the module will not be able to send or receive any messages.

To check the network registration status, issue the `+CEREG` read command. The second parameter of the information text response (`+CEREG: <mode>,<state>`) provides the interested information:

- 0: not registered
- 1: registered, home network
- 2: not registered, but currently in the process to
- 3: registration denied.
- 4: unknown (e.g. out of E-UTRAN coverage)
- 5: registered, roaming network

At power-up, the UE starts searching for HPLMN without checking for any previously attached networks. If the SIM used is an international SIM (roaming SIM) then the registration process can take many minutes. This is different to how SARA-N2 series will search for a network.

The `+CEREG` URC can be enabled to provide the network registration status. Depending on the <mode> parameter it is possible to configure the interested URC parameters (i.e. <mode>=4 or 5 to see the provided network timers). See SARA-N2 / SARA-N3 series AT commands manual [2] for more details.

☞ Properly setting the `+CEREG` AT command (<modem>=3, 4 or 5) it is possible to see the registration EMM cause value. These values are described in the 3GPP TS 24.008 [4]. Typical causes:

#5	IMEI not accepted
#11	PLMN not allowed
#12	Location Area not allowed
#13	Roaming not allowed in this location area
#22	Congestion

10.2 Base station connection (RRC connected state)

When a MO message is sent from the module, the module must first create a RRC connection if there is not already established with the base station. This status can be checked using the `+CSCON` AT command or by enabling the `+CSCON` URC.

To check the signaling connection status issue the `+CSCON` read command. The second parameter of the information text response (`+CSCON: <n>,<state>`) provides the interested information:

- 0: idle mode (no RRC connection)
- 1: connected mode (RRC connection)

To configure a URC for this command, issue the `AT+CSCON=1` command. A URC will be issued at each RRC connection status change.

10.3 Power Save Mode status

The module implements the Power Save Mode (PSM) where it goes in to a “Deep Sleep” mode consuming ~3 µA. It enters the PSM after the T3324 timer expires. It is possible to query the value of the T3324 timer with the +CEREG read command.

Issue the AT+NPSMR=1 command to enable the module’s power mode URC reporting. When the module’s PSM status changes, the +NPSMR URC is issued.

10.4 Connection status matrix

Table 1 provides an overview of the AT command described in the previous sections and their values according with the module status:

+CEREG	+CSCON	+NPSMR	Status
0	0	0	Module is on, but not performing any network activity
2	0	0	Module scanning for base station
2	1	0	Module starting registration process, RRC connected mode
1 or 5	1	0	Module registered, RRC connected mode
1 or 5	0	0	Module registered, RRC released, inside T3324 Period (paging/eDRX)
1 or 5	0	1	Module registered, in Power Save Mode
3	-	-	Registration failed

Table 1: connection status compatibility matrix

10.5 Viewing IP address

Before sending any IP data to the NB-IoT network configure the device with an IP address.

☞ Use the +CGPADDR AT command to query what the IP address of the module has been set to.

The +CGDCONT AT command lists the IP addresses for each defined PDP context.

No IP address is displayed until the module has registered to the network and has been provided with an IP address. The application could simply poll this IP address to see when the entire registration process has finished.

10.6 Viewing DNS addresses

The network must provide the DNS address to the module. There is no AT command to set the DNS addresses. To check if the network has provided DNS addresses issue the +CGCONTRDP read command. This will list the first and second DNS IP addresses.

If the network supports DNS, it will provide the DNS addresses when the module registers on the network. There is no command to request the network to provide DNS addresses, it is done automatically.

The +UDNSRN AT command cannot resolve hostnames or use hostnames in other AT commands if no DNS addresses is set by the network.

10.7 Checking module statistics

When the module is synchronized to the base station and is receiving signaling, then the +NUESTATS AT command can describe the radio, cell, BLER and throughput statistics.

The power, SNR and RSRQ values returned by the +NUESTATS AT command need to be divided by 10 to get the dB value, as for example, 108 => 10.8 dB.

Command	Response	Description

`AT+NUESTATS="RADIO"`	`Signal power: number`
	`Total power: number`
	`TX power: number`
	`TX time: number`
	`RX time: number`
	`Cell ID: number`
	`ECL: number`
	`SNR: last snr value`
	`EARFCN: last earfcn value`
	`PCI: last pci value`
	`RSRQ: last RSRQ value`
	`OK`
		The information text response indicates the module condition and its environment. The values may be useful for monitoring purposes.

- **Signal power** is the power of the wanted part of the receive signal, the NB-IoT part.
- **Total power** is the radio signal strength within the receive bandwidth (both expressed in 10ths of a decibel). From this the signal to noise ratio can be calculated.
- **Transmit power** is the RF power output from the module. It may be a low number if the received signal strength is good (and hence the module assumes that the base station is close by).
- **Tx time** is the duration for which the module's transmitter has been switched on.
- **Rx time** is the duration for which the module's receiver has been monitored for downlink activity (both expressed in milliseconds since the last reboot). Together these can be used to assess the time the module spends in each state and hence estimate the power consumed by the module.
- **Cell ID** is the physical cell ID of the cell that is currently providing service to the module.
- **ECL** is equivalent to "PRACH coverage enhancement level" defined in 3GPP 36.321 [3] sub clause 5.1.

☞ There are other statistics available too, for cell, BLER, and throughput.

- `AT+NUESTATS="CELL"`
- `AT+NUESTATS="BLER"`
- `AT+NUESTATS="THP"`

For more details on the `+NUESTATS` AT command, see SARA-N2 / SARA-N3 series AT commands manual [2].

11 Secure data

11.1 Certificates manager +USECMNG

The +USECMNG AT command is used to manage SSL/TLS certificates and private keys. Particularly, the command enables:

- Import of certificates and private keys
- Listing and information retrieval of imported certificates and private keys
- Removal of certificates and private keys
- MD5 calculation of imported certificate or private key

For more details on this AT command, the number and the format of the certificates, and the private keys accepted, see the SARA-N2 / SARA-N3 series AT commands manual [2].

There are various security configurations the host application can choose, depending on the cloud server requirements. The example below shows how to use the +USECMNG AT command for loading a certification authority (CA) certificate, client certificate and client private key so that they can be referenced by the security profile.

Command	Response	Description
AT+USECMNG=1,0,"ca_cert","ca_cert.crt"	+USECMNG: 1,0,"ca_cert","d10137cee624f cee624418db5eaa" OK	Import the CA certificate from the file "ca_cert.crt" stored on the file system.
AT+USECMNG=1,1,"client_cert","client_cert.crt"	+USECMNG: 1,1,"client_cert","b137ce 137ce5edd6723d8b13" OK	Import the client certificate from the file "client_cert.crt" stored on the file system.
AT+USECMNG=1,2,"client_key","client_key.key"	+USECMNG: 1,2,"client_key","087ab34c9aa03fb ce5edd6723d8b8e05" OK	Import the client private key from the file "client_key.key" stored on the file system.
AT+USECMNG=3	"CA","ca_cert","An MQTT broker","2032/10/18 08:23:32" "CC","client_cert","A client certificate","2032/06/22 12:34:48" "PK","client_key" OK	List all imported certificates or private keys.

11.2 Profile configuration +USECPRF

The `+USECPRF` AT command is used to configure a security profile that is used for an SSL/TLS/DTLS connection.

In particular, the command manages security profiles for the configuration of the following SSL/TLS/DTLS connections properties:

- Certificate validation level
- Minimum SSL/TLS/DTLS version to be used
- Cipher suite to be configured
- Certificate to be used for server and mutual authentication
- Expected server hostname, when using certificate validation level 2 or 3
- Password for the client private key, if it is password protected
- Pre-shared key used for connection
- Server Name Indication (SNI)
- (D)TLS session resumption.

☞ Some application protocols, (e.g. HTTP, MQTT) allow selection of which configured security profiles shall be used.

For more details on this AT command and all the related configurations, see the SARA-N2 / SARA-N3 series AT commands manual [2].

Command	Response	Description
`AT+USECPRF=0`	`OK`	Reset (set to factory-programmed value) all the parameters of security profile #0. ☞ It is recommended to issue the reset as the first command to erase all previously stored values.
`AT+USECPRF=0,0,1`	`OK`	Enable certificate validation without URL integrity check for profile #0. The server certificate will be verified with a specific trusted certificate or with each of the imported trusted root certificates.
`AT+USECPRF=0,2,3`	`OK`	Select legacy cipher suite for profile #0.
`AT+USECPRF=0,3,"ca_cert"`	`OK`	Select trusted root certificate internal name for profile #0.
`AT+USECPRF=0,5,"client_cert"`	`OK`	Select trusted client certificate internal name for profile #0.
`AT+USECPRF=0,6,"client_key"`	`OK`	Select trusted client key internal name for profile #0.

11.3 Complete example

The following example shows how to upload a server certificate to the module, and to set security profile 2 to use this method of authentication. Subsequently, the HTTP profile is configured, to use security profile 2 and execute the HTTP GET request.

Command	Response	Description
Step1: Import a trusted root certificate using the stream of byte like the `+UDWNFILE` AT command		
`AT+USECMNG=0,0,"ThawteCA",1516`	`>`	Start the data transfer using the stream of byte. ☞ Differently from the previous example in Section 11.1, in this case the certificate is transferred as a stream of byte and it is not stored in the SARA-N3 file system.

`-----BEGIN CERTIFICATE-----`
`MIIEIDCCAwigAwIBAgIQNE7VVyDV7exJ9C/OjVaMaA==`
`-----END CERTIFICATE-----`
	`+USECMNG: 1,0,"ThawteCA","8ccadc0b22cef5be72ac411a11a8d812"`
	`OK`
		Input PEM formatted trusted root certificate data bytes. Output MD5hash string of the stored trusted root certificate DER.

Step 2: List all available certificates and private key		

`AT+USECMNG=3`	`"CA", "ThawteCA","thawte Primary Root CA","2036/07/17"`
	`OK`
		List all available certificates and private keys.

Step 3: Set the security profile 2 validation level to a trusted root		
`AT+USECPRF=2,0,1`	`OK`	Security profile 2 has the validation level set to a trusted root.
Step 4: Set the security profile 2 trusted root certificate to the CA certificate imported as "ThawteCA"		
`AT+USECPRF=2,3,"ThawteCA"`	`OK`	Security profile 2 will use the CA certificate imported as "ThawteCA" for server certificate validation.
Step 5: Use the configured USECMNG profile 2 with the UHTTP application		
`AT+UHTTP=0,1,"www.ssl_tls_test_server.com"`	`OK`	Configure the UHTTP server name.
`AT+UHTTP=0,6,1,2`	`OK`	Enable the SSL/TLS for the UHTTP profile #0 and specify the SSL/TLS security profile 2.
`AT+UHTTPC=0,1,"/","https.resp"`	`OK`	Execute the HTTP GET command.
	`+UUHTTPCR: 0,1,1`	HTTP GET URC response.

Due to significant memory fingerprint of an SSL/TLS connection, the number of concurrent SSL/TLS connections is limited. The `+USECMNG` AT command and the underlying SSL/TLS infrastructure allows 4 concurrent SSL/TLS connections (i.e. 4 HTTPS requests or 2 HTTPS and 2 FTPS request).

12 TCP / UDP sockets

SARA-N3 series modules can send raw data through TCP and UDP sockets to an IP address. The data sent over the socket AT commands is not wrapped in any other layer, and the data provided is the data that is sent.

12.1 Creating a TCP / UDP socket

Create a socket to be able to send TCP / UDP data. A socket ID is returned.

Command	Response	Description
UDP socket		

`AT+USOCR=17`	`<socketID>`
	`OK`
		Create a UDP socket with a listening port. Returns the socket ID to be used with other socket commands.

TCP socket		

`AT+USOCR=6`	`<socketID>`
	`OK`
		Create a TCP socket with a listening port. Returns the socket ID to be used with other socket commands.

12.2 Closing a socket

Once a socket is no longer needed, it should be closed.

Command	Response	Description
`AT+USOCL=<socketID>`	`OK`	Specify the socket ID to close.

12.3 Connecting to a remote host

To establish a peer-to-peer connection of the socket to the specified remote host use the `+USOCO` AT command. For a TCP socket this will establish the link over the 3-way handshake. For UDP this command just saves the remote address and port for later use with the `+USOWR` and `+USORD` AT commands.

Command	Response	Description
`AT+USOCO=<socketId>,"<remote_address>",<remote_port>`	`OK`	Connect to the specified IP address and port through the socket noted by the ID.

12.4 Sending data

There are two methods to send data. Either with the `+USOWR` AT command or the `+USOST` AT command. The text response provides the number of bytes successfully sent.

☞ The maximum length of data that can be sent is 512 bytes in HEX mode, 1024 bytes in normal mode and extended binary mode.

12.5 Sending with `+USOWR`

The `+USOWR` AT command can be used with TCP sockets or UDP sockets. This command requires the `+USOCO` AT command to connect to the server IP address and port before it is used.

Command	Response	Description

`AT+USOWR=<socketId>,<length>,"<data>"`	`<socketId>,<length>`
	`OK`
		Send data to the specified IP address and port through the socket noted by the ID.

12.6 Sending UDP data

It is highly recommended to use the `+USOST` AT command with UDP sockets, without using the `+USOCO` AT command.

Command	Response	Description

`AT+USOST=<socketId>,"<remote_address>",<remote_port>,<length>,"<data>"`	`<socketId>,<length>`
	`OK`
		Send data to the specified IP address and port through the socket noted by the ID.

When the UDP packet has been successfully sent to the cellular network a `+UUSOST` URC will be emitted. This URC can be used to know when the message has been transferred from the module to the cellular network. With SARA-N2 series modules there was no indication when the UDP message was sent up to the network. With SARA-N3 series modules the host application can know when the message has been sent. This URC does NOT mean the UDP message has been received by the server.

12.7 Receiving UDP data

The data reception is performed in two steps. If the module has received data from the network on a socket that is listening, then a URC is given. From this message, the application can read the data on the appropriate socket.

12.7.1 Data arrived indicator

Command	Response	Description
	`+UUSORF:<socketId>,<length>`	This message is provided to tell the application how much data is available to read on the specified socket.

The application should read this message and then read the data from the specified socket.

12.7.2 Reading data from UDP socket

The data reception is performed by means of the `+USORF` AT command, using the information given in the `+UUSORF` URC.

Command	Response	Description

`AT+USORF:<socketId>`	`+USORF: <socket>,<ip_addr>,<port>,<length>,<data>,<remaining>`
	`OK`
		Provides the received data to the application and shows how much data is still left to read out.

☞ SARA-N3 series modules will output all data it has received. There is no <length> parameter with the `+USORF` set command.

☞ If there is no data the response will be: `+USORF: <socketId>,0`.

12.7.3 Reading data from TCP socket

The data reception is performed by means of the `+USORD` AT command, using the information given in the `+UUSORD` URC.

Command	Response	Description

`AT+USORD:<socketId>,<length>`	`+USORD: <socket>,<length>,<data>`
	`OK`
		Provides the received data to the application.

☞ The `+USORF` AT command is not supported for TCP sockets.

12.8 Testing UDP sockets

A simple way to test UDP sockets over the NB-IoT network is to send data to an echo server.

☞ The u-blox echo server is echo.u-blox.com. The IP address is 195.34.89.241.

Here below is an example:

Command	Response	Description

`AT+USOCR=17`	`1`
	`OK`
		Create a socket. The socket ID is 1.
		Ready for next AT command.

`AT+USOST=1,"195.34.89.241",7,5,"hello"`	`1,5`
	`OK`
		Send "hello" to u-blox echo server. Sent 5 bytes on socket ID 1.

	`+UUSORF:1,5`	Received 5 bytes on socket ID 1

`AT+USORF=1`	`1,195.34.89.241,7,5,"hello",0`
	`OK`
		Read all bytes on socket ID 1.
		Received data information provided…"hello".

☞ SARA-N3 series modules do not support reading partial response from the UDP socket. The <length> field is not supported.

☞ SARA-N3 series modules support text and HEX modes for the `+USOWR`, `+USOST` and `+USORF` AT commands.

13 Constrained Application Protocol (CoAP)

The Constrained Application Protocol (CoAP) is a datagram-based client/server application protocol for devices on the constrained network (e.g. low overhead, low-power), designed to easily translate to HTTP for simplified integration with the web. CoAP clients can use the GET, PUT, POST and DELETE methods using requests and responses with a CoAP server.

13.1 CoAP profile

Configure the CoAP client contained within the module via the one of the four profiles. It is possible to store up to four profiles in the module and select between them. Only one profile can be used at a time.

The profiles are configured, stored, and retrieved by setting parameter/value pairs by means of the `+UCOAP` AT command and its <op_code> parameter that can assume these values:

- 0: server IP address (not applicable on SARA-N3 series)
- 1: URI (must include server address/hostname & port)
- 2: PDU option mask configuration (option to be added in PDU header)
- 3: current profile number (0 to 3)
- 4: valid flag
- 5: restore profile from NVM
- 6: store profile to NVM
- 7: read all profiles
- 8: CoAP secure option (SSL encryption)
- 9: Release assistance indication (RAI)

13.1.1 Specifying CoAP URI

The URI of the CoAP server the host application is connecting to is specified by using the `+UCOAP=1, "<URI>"` AT command. For example; `AT+UCOAP=1,"coap://coap.me:5683/test"`

This example is using the "coap://" scheme, hostname "coap.me" (an online CoAP test server), on port "5683". The resource is "test"

☞ If the port for the URI is omitted, the port will default to 5683.

☞ `AT+UCOAP=0,<coap_server>[,<coap_port>]` is not supported by SARA-N3 series.

13.1.2 Specifying CoAP PDU Headers

The internal CoAP client needs to construct the correct PDU header for the CoAP action/request. The host application specifies the PDU headers via the PDU option mask `AT+UCOAP=2,<option>,<state>` command.

On SARA-N3 series the following options can be added to the PDU headers for a CoAP action/request:

`AT+UCOAP=2,0,1`	Include the hostname
`AT+UCOAP=2,1,1`	Include the port
`AT+UCOAP=2,2,1`	Include the path
`AT+UCOAP=2,4,1`	Include the Content Format (not required for Delete action)

13.1.3 Using non-secure CoAP servers (coap://)

CoAP can be used with non-secure CoAP servers. When using non-secure CoAP servers the host application must specify the coap:// scheme.

13.1.4 Using secure CoAP servers (coaps://)

CoAP can be used with secure CoAP servers. When using secure CoAP servers the host application must specify the coaps:// scheme. The <op_code>=8 security profile selection is ignored when using coap:// scheme.

The security "profile" is specified by the `AT+UCOAP=8,<security profile>` command: For example, `AT+UCOAP=8,0` specifies to use Security Profile 0. For more information on security profiles, see section 11.

13.1.5 Enabling and storing the CoAP profile

To enable the profile (to make it "valid") the host application needs to set the valid option. This is done by the `AT+UCOAP=4,1` command. To disable a profile, to make it "not valid" set the option to "0".

To store the profile that has now been configured, set the profile number and then execute the store profile option.

`AT+UCOAP=3,<profile number>`	Set the profile number
`AT+UCOAP=6,<profile number>`	Store to NVM

13.1.6 Loading a stored CoAP Profile

Once the host application has configured the CoAP profile, set the profile number and stored it to NVM it is good practice to "restore" the profile before any CoAP requests are made. This is to make sure the module is using the correct information for the request.

Issue the `AT+UCOAP=5,<profile number>` command to restore the saved profile to the current profile.

☞ Even if a CoAP profile has just been configured, it is best practice to "restore" it.

13.1.7 CoAP Profile example

Below is an example of how to configure CoAP profile 1 for the online test CoAP Server "coap.me"to

Command	Response	Description
`AT+UCOAP=1,"coap://coap.me:5683/test"`	OK	Set CoAP URI
`AT+UCOAP=2,0,1;+UCOAP=2,1,1;+UCOAP=2,2,1;+UCOAP=2,4,1`	OK	Set CoAP option mask (host-port-path-content format)
`AT+UCOAP=3,1`	OK	Set Profile Number to '1'
`AT+UCOAP=4,1`	OK	Set this as a valid profile
`AT+UCOAP=6,1`	OK	Store the CoAP profile number 1 in NVM
`AT+UCOAP=5,1`	OK	Restore the CoAP profile number 1 from NVM

13.2 Sending CoAP action or request

Issue the `+UCOAPC` AT command to trigger a CoAP action/request. The `+UCOAPCR` URC is returned to indicate if the command was successful or not. If data is to be returned from the CoAP server, it is included in the `+UCOAPCD` URC.

```
AT+UCOAPC=<coap_command>[,<payload>,<identifier>]
```

The `<coap_command>` parameter represents the CoAP action or request:

- 1: GET (payload and identifier parameters are not allowed)
- 2: DELETE (payload and identifier parameters are not allowed)
- 3: PUT
- 4: POST

GET, PUT and POST actions accept block operations using the block parameters:

```
AT+UCOAPC=<coap_command>,<payload>,<identifier>,<block_number>,<more_block>
```

To indicate the last block of data, set `<more_block>` to "0".

The `<payload>` parameter included in the PUT or POST actions must be in hexadecimal format.

Supported identifiers are:

- 0: text / plain
- 1: application / link format
- 2: application / xml
- 3: application / octet format
- 4: application / rdf xml https://www.w3.org/TR/rdf-syntax-grammar/
- 5: application / exi https://www.w3.org/TR/exi-primer/
- 6: application / json https://www.w3schools.com/js/js_json_syntax.asp
- 7: application / cbor https://cbor.io/

This example sends a CoAP PUT action to the CoAP server.

Command	Response	Description
`AT+UCOAPC=3,"48656C6C6F20576F726C64",0`	`OK`	Execute the CoAP PUT action with "Hello World".
	`+UCOAPCD:3,"…"`	Result from the CoAP server.

13.3 CoAP errors

If there is an error when executing the action/request command it will fail with the `+UCOAPCR` URC `<coap_result>` parameter set to `0` (fail).

To get the actual error use the `+UCOAPER` AT command. This will return the <error_class> and <error_code> for the error. The `<error_class>` for CoAP is "5". The error codes are listed in the u-blox SARA-N2 / SARA-N3 series AT commands manual [2].

13.4 CoAP connection using PSK to AWS server

The AT commands highlighted are the commands that are important when setting security.

Command	Response	Description
`AT+USECPRF=0,8,"112233aabbcc"`	`OK`	Set pre-shared key on security profile 0 (hex format)
`AT+USECPRF=0,9,"u-blox-test"`	`OK`	Set pre-shared key identity on security profile 0
`AT+UCOAP=1,"coaps://18.219.204.203:5684/hello"`	`OK`	Set CoAPS URI

`AT+UCOAP=2,0,1`
`AT+UCOAP=2,1,1`
`AT+UCOAP=2,2,1`
`AT+UCOAP=2,4,1`
`AT+UCOAP=3,0`
`AT+UCOAP=4,1`
	`OK`
	`OK`
	`OK`
	`OK`
	`OK`
	`OK`
		Configure CoAP option mask
		Set profile number as 0
		Set profile as valid

`AT+UCOAP=8,0`	`OK`	Select USECMNG profile 0
`AT+UCOAP=6,0`	`OK`	Store the CoAP profile number 0 in NVM
`AT+UCOAP=5,0`	`OK`	Restore the CoAP profile 0
`AT+UCOAPC=1`	`OK`	Send GET request

☞ The `+USECPRF` AT command requires Key in HEX format.

☞ The `+USECPRF` AT command requires Key Identity in ASCII format.

14 MQTT

MQTT stands for MQ Telemetry Transport. It is a “publish/subscribe”, simple and lightweight messaging protocol. MQTT is designed for constrained devices and low bandwidths, high latency, or unreliable networks. This makes MQTT protocol well suited Internet of Things devices.

SARA-N3 series modules provide an internal MQTT client base on v3.1 that can connect to unsecure and secure MQTT servers. Just like with the SARA-N3 internal CoAP client, security is selected by the security profile setting using the `+USECMNG` and `+USECPRF` AT commands. For more details on the security profiles, see section 11.

MQTT publishes messages to “topics”. The SARA-N3 internal MQTT client can publish and subscribe to topics that are on the MQTT server. The messages that the module publishes will be received by all the clients who have subscribed to that topic and are online. Only new messages published on a topic will be received by the client from when they subscribe to that topic, or since the last “retained” message.

Just like the CoAP action/requests with SARA-N3’s internal CoAP client, MQTT is much the same. To use MQTT the host application configures a MQTT profile, and then executes a MQTT action.

14.1 MQTT profile

Unlike the CoAP client which can store four profiles in the NVM, the MQTT client only has one current profile. This configured can be stored to the Non-Volatile Memory (NVM) and restored.

14.1.1 Resetting, storing, and restoring profile

The host application can reset the MQTT profile to the factory-programmed settings. Use the `AT+UMQTTNV=0` command to reset the profile.

When finished configuring the MQTT profile, save it to NVM using the `AT+UMQTTNV=2` command.

To restore the MQTT profile from NVM, use the `AT+UMQTTNV=1` command.

14.1.2 Configuring MQTT server connection

The MQTT client ID is automatically set to the IMEI of the module. The host application can change this if required. The maximum number of characters is 23. To change the MQTT client ID issue the `AT+UMQTT=0,"<text>"` command.

To configure what MQTT server to connect to, the host application can specify the hostname or the IP address by using one of these two commands:

```
AT+UMQTT=2,"servername"[,<port>]  or  AT+UMQTT=3,"IP Address"[,<port>]
```

If the server requires a username and password the host application can set them with the `AT+UMQTT=4,<username>,<password>` command. For example:

```
AT+UMQTT=4,"user_one","pa55w0rd"
```

14.1.3 MQTT inactivity timeout

The default value for the inactivity timeout is 0. A timeout of 0 means the MQTT server will not disconnect the SARA-N3’s internal MQTT client if it does not send any messages. If the inactivity timeout value is set, the MQTT server will disconnect the MQTT client if it does not send a message up to it within 1.5x the inactivity timeout value. To set the inactivity timer use the `AT+UMQTT=10,<inactivity timeout>` command. The timeout value can be between 0 and 65535 s.

14.1.4 MQTT security

To use a secure MQTT broker connection using TLS, the host application can configure this by means of the `AT+UMQTT=11,<MQTT_Secure>[,<Security Profile>]` command. Just like the CoAP client, the MQTT client security is configured by specifying the security profile.

☞ The security profile is configured using the `+USECMNG` and `+USECPRF` AT commands. For more information on security profiles, see section 11.

Set the `<MQTT_Secure>` parameter to '1' to enable TLS security and "0" for no TLS security.

14.2 MQTT commands

MQTT commands will return the "OK" final result code if the AT command is accepted and being executed. The end result of the command will be provided via the `+UUMQTTC` URC; `+UUMQTTC: <op_code>,<result>`. The `<op_code>` parameter indicates the command that was executed.

14.2.1 Login and Logout of MQTT server

The MQTT client will need to log into the MQTT server before publishing or subscribing to any topics. To login use the `AT+UMQTTC=1` command. To log out, use the `AT+UMQTTC=0` command.

The host application will need to wait for the `+UUMQTTC` result before knowing if it was successful.

14.2.2 Subscribing to topics

To receive MQTT messages the host application will need to subscribe to a topic. Topic names are hosted on the MQTT broker and need to be known already by the host application. There is no method to retrieve the topic names from the server.

To subscribe to a topic the host application specifies the topic filter. This filter specifies the messages that will be sent to the module. Use the `AT+UMQTTC=4,<QoS>,<topic_filter>` command to subscribe to a topic.

The topic filter can use wildcards; + and #. + wildcard is a single level wildcard, meaning any name at that topic level. For example, "/sensors/temperature/+" will only provide messages at the /sensors/temperature level and none below that.

# wildcard is a multi-level wildcard meaning to match any name at this level and all levels below that. I.e. "/sensors/temperature/#" will match "/sensors/temperature/bedroom/1/ensuite".

When the module has successful registered the subscription with the MQTT server the `+UMQTTC: 4,1` URC will be returned.

14.2.3 Quality of service (<QoS>)

The <QoS> parameter for MQTT commands is to specify how the message will be delivered. Either:

- At the most once: the message might not be delivered at all, and only once at the most. This is the lowest service. This service is no better than the service from TCP protocol.
- At least once: this guarantees the message will be delivered at least once to the receiver. According to MQTT protocol the message may be delivered multiple times also.
- Exactly once: the message will be guaranteed to be received only once. This is the highest level of service in MQTT, and because of the extra overhead consumes the most energy.

14.2.4 Unsubscribing from a topic

To unsubscribe from a topic filter, issue the `AT+UMQTTC=5,<topic_filter>` command.

14.2.5 Sending MQTT messages

SARA-N3 series modules can send MQTT message on a topic using the `AT+UMQTTC=2,<QoS>,<retain>,[<hex_mode>],<topic>,<message>` command.

<retain> parameter tells the MQTT broker how to retain this message. If a message is “retained” then when a new MQTT client subscribes to this topic they will be sent the last retained message. This is useful to allow clients to see the last message of that topic.

☞ The topic name does not contain any wildcards. Wildcards are only used for subscribing

14.3 MQTT errors

If the `+UUMQTTC` URC returns a command failed status, the description of the error can be read out using the `+UMQTTER` AT command.

15 MQTT-SN

MQTT-SN stands for MQTT for Sensor Networks. This protocol is designed for not-always-on LPWA devices with limited payload size. It was designed to work just the same way as MQTT, using the same publish/subscribe model and can be considered a version of MQTT.

MQTT-SN uses the UDP protocol to send and receive the messages.

15.1 Basic setup

MQTT-SN uses a configuration profile to define various client behaviors and server parameters. as with the MQTT profile, there is only one MQTT-SN profile. This can be stored in the NVM with the `+UMQTTSNNV` AT command.

15.1.1 Resetting, storing and restoring profile

The host application can reset the MQTT-SN profile to the factory-programmed settings by means of the `AT+UMQTTSNNV=0` command.

When finished configuring the MQTT-SN profile, it can be stored in the NVM by means of the `AT+UMQTTSNNV=2` command. To set the MQTT-SN profile to the configuration previously stored in the NVM, issue the `AT+UMQTTSNNV=1` command.

Command	Response	Description
AT+UMQTTSNNV=2	OK	Store the current MQTT-SN client profile parameters in the NVM.
AT+UMQTTSNNV=0	OK	Restore MQTT-SN client profile parameters to the factory-programmed setting.
AT+UMQTTSN?	+UMQTTSN: 0,"357862090033897" +UMQTTSN: 1,"",1883 +UMQTTSN: 2,"",1883 +UMQTTSN: 3,0 +UMQTTSN: 4,0 +UMQTTSN: 5,0 +UMQTTSN: 6,"" +UMQTTSN: 7,"" +UMQTTSN: 8,0 +UMQTTSN: 9,0 +UMQTTSN: 10,0 OK	Get the current MQTT-SN client profile configuration.
AT+UMQTTSNNV=1	OK	Set MQTT-SN client profile parameters to values previously stored in the NVM.
AT+UMQTTSN?	+UMQTTSN: 0,"357862090033897" +UMQTTSN: 1,"",1883 +UMQTTSN: 2,"192.168.105.30",1883 +UMQTTSN: 3,0 +UMQTTSN: 4,0 +UMQTTSN: 5,0 +UMQTTSN: 6,"" +UMQTTSN: 7,"" +UMQTTSN: 8,0 +UMQTTSN: 9,0 +UMQTTSN: 10,0 OK	Get the current MQTT-SN client profile configuration.

15.1.2 Default and minimal configuration

The minimum configuration required to start a MQTT-SN session is the unique client id and server name of the MQTT-SN gateway.

Command	Response	Description
AT+CMEE=2	OK	Set verbose error result codes.
AT+UMQTTSN?	+UMQTTSN: 0,"357862090033897" +UMQTTSN: 1,"",1883 +UMQTTSN: 2,"",1883 +UMQTTSN: 3,0 +UMQTTSN: 4,0 +UMQTTSN: 5,0 +UMQTTSN: 6,"" +UMQTTSN: 7,0,"" +UMQTTSN: 8,0 +UMQTTSN: 9,0 +UMQTTSN: 10,0 OK	Read the current profile configuration All the reported values can be modified. For a detailed description, see the SARA-N2 / SARA-N3 series AT commands manual [2]. ☞ The default client id value is the IMEI of the module because it guarantees the uniqueness of the client to the server.
AT+UMQTTSN=2,"192.168.105.30",10000	OK	Set the IP address and port of the remote MQTT-SN gateway. Alternatively, the gateway's server name can be set by means of the AT+UMQTTSN=1,<server_name>[,<server_port>] command.

15.1.3 Last will configuration

The last will parameters configure the message that the MQTT-SN clients connected to the gateway will receive if the SARA-N3 should be disconnected due to an error. Following is an example of setup.

Command	Response	Description
AT+UMQTTSN=4,1	OK	Set the last will QoS level to 1.
AT+UMQTTSN=6,"u-blox/publish"	OK	Set the last will topic.
AT+UMQTTSN=7,"Unrequested disconnect."	OK	Set the last will message.

15.2 Start and end a MQTT-SN session

See the default and minimal configuration described in section 15.1, to configure the MQTT-SN profile before starting a connection.

Command	Response	Description
AT+UMQTTSNC=1	OK +UUMQTTSNC: 1,1	Connect to the gateway and start a MQTT-SN session.
AT+UMQTTSNC=0	OK +UUMQTTSNC: 0,1	Disconnect from the gateway, end of the MQTT-SN session.

15.3 Subscribe to a normal topic and publish a message to the same topic

The following example is a demonstration of the main functionalities that can be performed with the SARA-N3 series AT commands. In this MQTT-SN session the module subscribes to a topic, publishes a message to the topic and receives the published message (since it is subscribed to topic of the published message).

Command	Response	Description

`AT+UMQTTSNC=5,0,0,"room/temperature"`	`OK`
	`+UUMQTTSNC: 5,1,0,1`
		Subscribe to a normal topic (0) with requested QoS level set to 0.
		The gateway granted QoS level is 0 and the topic ID for “room/temperature” is 1.

`AT+UMQTTSNC=4,0,0,,0,"1","20 degrees Celsius"`	`OK`
	`+UUMQTTSNC: 4,1`
		Publish the “20 degrees Celsius” message to the topic ID 1 with requested QoS level and retain value set to 0.

	`+UUMQTTSNC: 9,1,"1",0,0,18`	Notification of the received publish message.

`AT+UMQTTSNC=9`	`+UMQTTSNC: 9,0,"1",0,0,18,"20 degrees Celsius"`
	`OK`
		Request to read the message.

`AT+UMQTTSNC=6,0,"1"`	`OK`
	`+UUMQTTSNC: 6,1`
		Unsubscribe from the previously subscribed topic.

15.4 Register to a topic and publish a message to the same topic

The following example differs from the previous one only for the non-receipt of publish message since the module is not subscribed to the topic.

Command	Response	Description

`AT+UMQTTSNC=2,"kitchen/temperature"`	`OK`
	`+UUMQTTSNC: 2,1,2`
		Register to a normal topic.
		The returned topic ID for “room/temperature” is 2.

`AT+UMQTTSNC=4,1,0,,0,"2","25 degrees Celsius"`	`OK`
	`+UUMQTTSNC: 4,1`
		Publish the “25 degrees Celsius” message to the “kitchen/temperature” topic using the above topic ID.

15.5 Subscribe to a short topic name and publish a message to the same topic

The short topic is composed of only 2 characters.

Command	Response	Description

`AT+UMQTTSNC=5,0,2,"aa"`	`OK`
	`+UUMQTTSNC: 5,1,2,0`
		Subscribe to a short topic (2) with requested QoS level set to 0.
		The gateway granted QoS level is 0

`AT+UMQTTSNC=4,1,0,,2,"aa","test"`	`OK`
	`+UUMQTTSNC: 4,1`
		Publish the “test” message to the “aa” topic with requested QoS level and retain value set to 0.

	`+UUMQTTSNC: 9,1,"aa",1,0,4`	Notification of the received publish message.

`AT+UMQTTSNC=9`	`+UMQTTSNC: 9,0,"aa",1,0,4,"test"`
	`OK`
		Request to read the message

`AT+UMQTTSNC=6,2,"aa"`	`OK`
	`+UUMQTTSNC: 6,1`
		Unsubscribe from the previously subscribed topic.

15.6 Last will

In order to see the last will publish message, two modules shall start a MQTT-SN session with the same gateway. For the first module, before starting a MQTT-SN session, the last will parameter shall be configured, see last will configuration in section 15.1.3. The second module shall subscribe to the last will topic of the first module.

Command	Response	Description
`Module #1`		
`AT+UMQTTSNC=1`	`OK`	Connect to the gateway and start a MQTT-SN session.
	`+UUMQTTSNC: 1,1`	
`Module #2`		
`AT+UMQTTSNC=1`	`OK`	Connect to the same gateway and start a MQTT-SN session.
	`+UUMQTTSNC: 1,1`	
`AT+UMQTTSNC=5,0,0,"u-blox/publish"`	`OK`	Subscribe to the last will topic "u-blox/publish".
	`+UUMQTTSNC: 5,1,0,1`	
`Module #1`		
`AT+COPS=2`	`OK`	Simulate a disconnect
`Module #2`		
	`+UUMQTTSNC: 9,1,"u-blox/publish",0,0,23`	Notification of the received publish message.

`AT+UMQTTSNC=9`	`UMQTTSNC: 9,0,"u-blox/publish",0,0,23,"Unrequested disconnect."`
	`OK`
		Read the received last will publish message.

15.7 Error handling

In case of errors returned by the gateway via the `+UUMQTTSNC: x,0` URC it is possible to investigate the type of error using the `+UMQTTSNER` AT command.

Command	Response	Description
`AT+UMQTTSNC=5,1,0," kitchen/temperature"`	`OK`	Unsuccessful subscribe.
	`+UUMQTTSNC: 5,0`	

`AT+UMQTTSNER`	`+UMQTTSNER: 14,21`
	`OK`
		Error code 21 is “Timeout error” that means the gateway did not replay to the subscribe request.

15.8 Secure MQTT-SN

A secure manager profile must be configured before starting a secure MQTT-SN session (using the DTLS encryption protocol). See the section 11 for further details on this.

In the following example, it is reported how to configure the MQTT-SN profile before starting a secure session with the gateway. Only the secure manager profile and the remote port must be configured, the other MQTT-SN commands will behave as in the case of an unencrypted session.

Command	Response	Description
`AT+UMQTTSN=9,1,2`	`OK`	Enable the secure MQTT-SN option using the USECMNG profile 2.
`AT+UMQTTSN=2,"192.168.105.30",10001`	`OK`	Set the remote MQTT-SN gateway IP address and port.

`AT+UMQTTSNC=1`	`OK`
	`+UMQTTSNC: 1,1`
		Connect to the gateway and start a secure MQTT-SN session.

16 LwM2M

The LwM2M client on the SARA-N310 module is designed as a split client. Half of the client resides in the SARA-N310 module and the other half is required to be provided by the host application.

The LwM2M client will handle the server registration and notification of the objects. The host application must hold on to the LwM2M objects and provide the information reports to the server when the observation requests are made. In other words, the SARA-N310 module does not host the objects internally; this is performed by the host application.

16.1 LwM2M profile

The LwM2M client has a configuration profile that the host application must configure using the `+ULWM2MCC` AT command. This is much the same as the CoAP and MQTT profiles.

16.1.1 Resetting, storing, and restoring a profile

Like the MQTT profile, there is only one LwM2M profile. This profile can be stored in the non-volatile memory using the `AT+ULWM2MCC=8` command. This profile is automatically loaded when the module boots.

16.1.2 Configuring LwM2M server connection

The LwM2M client identity is set to the IMEI of the SARA-N310 by default, but the host application can change this using the `AT+ULWM2MCC=0,<indentity>` command.

To configure if the server is a bootstrap server or a LwM2M server, use the `AT+ULWM2MCC=4,<mode>` command. Set to 0 for LwM2M server, and set to 1 for a bootstrap server.

Set the bootstrap server or LwM2M server address by means of the `AT+ULWM2MCC=2,<hostname>,<remote_port>` command.

16.1.3 Configuring for a secure LwM2M server

The internal LwM2M client can be used with secure LwM2M servers. Unlike CoAP and MQTT only PSK security can be used. Set the PSK ID and PSK KEY using the `AT+ULWM2MCC=5,1,<pskID>,<pskKEY>` command. The ID is a character text, and the KEY is in hexadecimal format.

To disable the use of security use `AT+ULWM2MCC=5,0` command.

☞ Certificate security is not implemented in SARA-N3 series modules.

16.2 LwM2M client

The internal LwM2M client needs to be created and registered to the LwM2M server before the host application can register objects and start to send information reports. When the host application is finished with its LwM2M operation it can deregister the client from the network and then delete the client.

`AT+ULWM2MSC=<code>` is used to create (0), register (1), deregister (2) and delete (3) the client.

The host application can only control the client in this order; Create -> Register -> Deregister -> Delete. That is to say, the client cannot be deleted unless it has been first deregistered. The client cannot be created if there is a client already created.

16.3 LwM2M object management

The internal LwM2M client has a very limited object management responsibility because of the split client design. The host application must manage the objects and resources external to the module.

For the LwM2M server to know about the objects the host application is managing the host application can use the `+ULWM2MOBJ` AT command. In this case the host application only needs to specify the object ID and object instance ID. Multiple objects can be added by separating each object by a forward slash "/".

For example, to add object 3306 instances 1,2,7 and object 3307 instances 1,2:

```
AT+ULWM2MOBJ=1,"3306:1:2:7/3307:1:2"
```

The host application can also delete objects using just the object identity, for example:

```
AT+ULMWM2MOBJ=0,"3307"  Remove all instances of the 3307 object.
```

☞ The object resources are reported to the server with another operation, either through device management read request, or information reporting request from the LwM2M server.

16.4 LwM2M device management

As the internal LwM2M client is a split design, the server's requests to the client are exposed to the host application via URC. The host application must handle the URC and respond with the necessary content back to the module to complete the server request.

The request will specify the response format the content should be formatted in. The host application must reply with the object/resource content formatted in the format that was requested. The possible formats are: Plain Text, Core Link Param, Opaque, TLV, and JSON. It is mandatory for LwM2M servers and clients to support TLV format.

☞ The <content> part of the AT command is encoded in hexadecimal format.

General LwM2M device management sequence is in Figure 3:

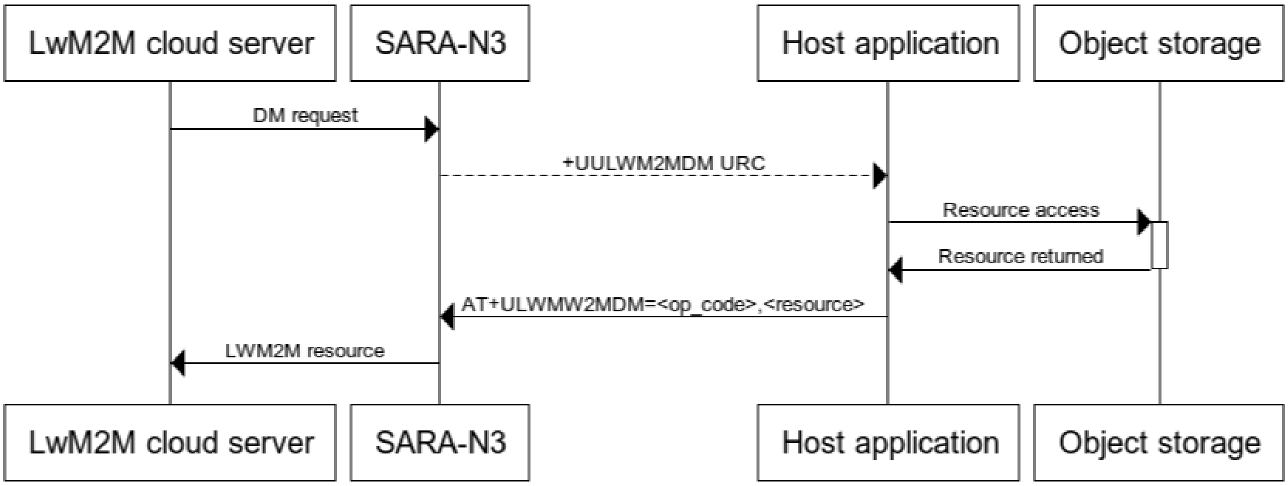


Figure 3: LwM2M device management sequence

16.4.1 Resource paths

When the LwM2M server sends a request to the module, it describes the LwM2M resource via a “resource path”. This resource path is a mapping of the object ID, object instance, resource ID and resource instance. The resource path is a text field in the following allowed formats:

- /{Object ID}/{Object Instance ID}/{Resource ID}
- /{Object ID}/{Object Instance ID}
- /{Object ID}

16.4.2 SSID

The LwM2M server has a “short server ID” which is provided to the module in the LwM2M device management requests. This is to enable the host application to track the SSID of the request – although because there can only be one LwM2M client running internally on the module, this SSID will always be the same for that client.

16.4.3 Read request

The LwM2M server is requesting to read a LwM2M resource in a particular format.

URC: `+UULWM2MDM: 0,<ssid>,<res_path>,<resp_format>`

The host application must handle the `+UULWM2MDM: 0` URC and provide back the content of the resource in the format specified by the `<resp_format>` parameter using the `+ULWM2MDM` AT command.

16.4.4 Discover request

The LwM2M server is requesting to discover a LwM2M resource in a particular format.

URC: `+UULWM2MDM: 1,<ssid>,<res_path>,<resp_format>`

The host application must handle the `+UULWM2MDM: 1` URC and provide back the content of the resource in the format specified by the `<resp_format>` parameter using the `+ULWM2MDM` AT command.

16.4.5 Write request

The LwM2M server is requesting to write content to a LwM2M resource in a particular format.

URC: `+UULWM2MDM: 2,<ssid>,<res_path>,<resp_format>,<length>,<content>,<write_type>`

The host application must handle the `+UULWM2MDM: 2` URC and read the content in the format as specified by the `<resp_format>` parameter. The content should be written to the resource as specified by the `<res_path>` parameter. The host application must send back the result of this request using the `+ULWM2MDM=2,<resp_code>` AT command.

☞ `<write_type>` is either “0” (Replace) or “1” (Update).

16.4.6 Write attribute

The LwM2M server is requesting to write an attribute to a LwM2M resource.

URC: `+UULWM2MDM: 3,<ssid>,<res_path>,<resp_format>,<length>,<content>`

The host application must handle the `+UULWM2MDM: 3` URC and read the content in the format as specified by the `<resp_format>` parameter. The content is describing the attribute that is attached to the resource object. The attribute describes the trigger for a Notify Information Report to be sent by the host application. The host application must send back the result of this request using the `+ULWM2MDM=2,<resp_code>` AT command.

An example of an attribute would be: “gt=45&st=10” – Greater than 45 and step of 10.

Please see the Open Mobile Alliance (OMA) - Lightweight Machine to Machine technical specification [7], Section 5.1 Attributes for more information about how attributes are defined by LwM2M.

16.4.7 Execute request

The LwM2M server is requesting to execute a LwM2M resource in a particular format.

URC: `+UULWM2MDM: 4,<ssid>,<res_path>,<resp_format>,<length>,<content>`

The host application must handle the `+UULWM2MDM: 4` URC and read the content in the format as specified by the `<resp_format>` parameter. The content should then be "executed" by the host application against the resource as specified by the `<res_path>` parameter. The host application must send back the result of this request using the `+ULWM2MDM=2,<resp_code>` AT command.

16.4.8 Create request

The LwM2M server is requesting to create a LwM2M object instance, with content, in a particular format.

URC: `+UULWM2MDM: 5,<ssid>,<res_path>,<resp_format>,<length>,<content>`

The host application must handle the `+UULWM2MDM: 5` URC. There is no response the host application needs to send back to the LwM2M server.

16.4.9 Delete request

The LwM2M server is requesting to delete a LwM2M object instance.

URC: `+UULWM2MDM: 6,<ssid>,<res_path>`

The host application must handle the `+UULWM2MDM: 6` URC. There is no response the host application needs to send back to the LwM2M server.

16.5 LwM2M information reporting

The SARA-N310 module can update the LwM2M server with new values for the objects/resources the host application is managing. Keep in mind that the SARA-N310 LwM2M is a split client design and the object management must be performed in the host application, so it is up to the host application to perform this information report to the LwM2M server.

The LwM2M server can write an "attribute" to the object/resource item through the `+UULWM2MDM: 3` URC. This attribute instructs the host application's object management when to send an update information report back to the LwM2M server.

The host application will have to manage the attributes on the resources, and when the time comes, the host application will send an information report to the LwM2M server.

The host application must use the `+ULWM2MIR` AT command to send the updated content for the resource that has triggered this update, specified by the attribute set on it.

The LwM2M server will instruct the client what objects it wishes to observe, using the `+UULWM2MIR` URC. If the resource path is for a single resource, the host application must respond immediately.

A typical sequence of events would be:

1. Client registers to LwM2M server
2. Client provides a list of objects it is managing
3. Server writes “attributes” to the object resources on the client, via the host application
4. Server sends observation request to client. Host application notes the object id/instance
5. <application runs and resource values are updated>
   - 5.1. Host application triggers on a resource being updated matching the attribute
   - 5.2. Host application sends Information Report to LwM2M server
6. [sometime later] Server cancels observation on object id/instance

Steps 1 – 5 are shown in Figure 4:

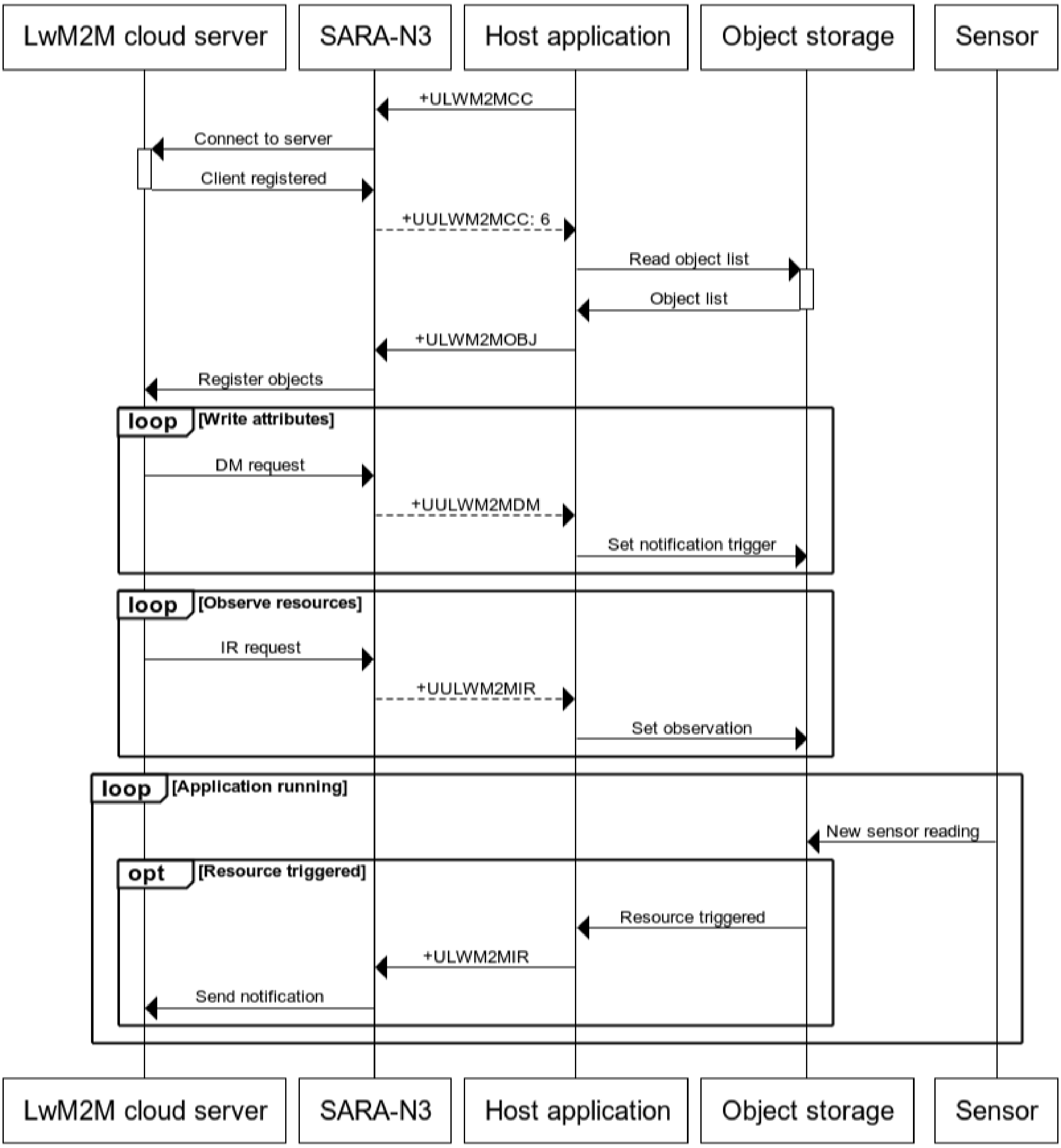


Figure 4: LwM2M information reporting sequence

17 Non-IP messaging

The usage of the Non-IP method during the sending or receiving of messages saves the overhead of needing to send an IP header. For UDP the header is about 48-60 bytes in length, and so an application sending 100 bytes will actually send about 160 bytes. For devices in the extreme coverage class 2, this can be quite costly.

When using Non-IP messaging, the mobile network operator shall provide access to the messages since they are stored by it as there is no destination IP address or port.

17.1 Connection process

Command	Response	Description
`AT+CDGCONT=<cid>,"NON-IP","nonip"`	`OK`	Create a PDP context with type “NON-IP” to connect to the network for Non-IP message.

17.2 Uplink messaging

Command	Response	Description
`AT+CSODCP=<cid>,<length>,"<data>"`	`OK`	Issue the +CSODCP AT command to send an uplink message for Non-IP.
`AT+CSODCP=<cid>,<length>,"<data>",<RAI>`	`OK`	It is possible to release assistance, just like with UDP messaging (where <RAI> is 0, 1 or 2). ☞ <data> is in Hex format.

17.3 Downlink messaging

Command	Response	Description
`AT+CRTDCP=1`	`OK`	Turn on the +CRTDCP URC to receive downlink messages.
	`+CRTDCP:<cid>,<length>,"<data>"`	When the module receives a downlink message, the +CRTDCP URC is issued.

17.4 Non-IP example

Here is a Non-IP messaging example:

Command	Response	Description
`AT+CGDCONT=1,"NON-IP","nonip"`	`OK`	Create a PDP context with type “NON-IP” to connect to the network for Non-IP message.
`AT+CGACT=1,1`	`OK`	Activate NON-IP context.
`AT+CRTDCP=1`	`OK`	Turn on the +CRTDCP URC to receive downlink messages.
`AT+CSODCP=1,3,"AA11BB"`	`OK`	Send NON-IP data of length 3.
	`+CRTDCP: 0,1,"ab"`	The module receives a downlink message.

18 Operation tests

18.1 Band of operation

Use the +UBANDSEL AT command to verify if the module is configured for the correct band of operation. One band, or multiple bands can be specified by means of this AT command. Issue the AT+UBANDSEL=0 command to restore the band list to the factory-programmed configuration. Bands are defined by their frequencies; 1800,850,900,800,700.

The more bands in this list the longer the module will take to perform a full network scan.

To make the band setting effective the module must be deregistered from the network first.

18.2 SIM

The IMSI can be queried by using the +CIMI AT command. Make sure about the correctness of the SIM IMSI. If the error final result code is returned the module cannot read the SIM.

☞ Use the +CPIN AT command, to lock and unlock the SIM card.

18.3 Scanning and registering on the network

After a RRC connection is made to the base station the module will try and register with the network. If the module IMEI or IMSI is not allowed on the network, the module will disconnect from that base station and continue scanning for other base stations. This can be seen if the <mode> parameter of the +CSCON AT command shows the “1” and then “0” response without +CEREG changing to 1 or 5 means that the module was not able to register on that network.

In case the module is registered to the network, the <status> parameter of the +CEREG AT command will be 1 (registered) or 5 (registered & roaming).

After a short period, if no messages are being sent from the module, the +CSCON response will be “0” to show the RRC connection has been released by the eNodeB.

18.4 T3324 and T3412 timers

T3324 and T3412 timers are set by the network after the module registration. These timers control the amount of time the module will be in idle mode before entering Power Save Mode (T3324) and how long it will wait until automatically sending a Tracking Area Update message (TAU) (T3412)

The timers can be read out using the +CEREG AT command if the <mode> parameter is set to 4 or 5.

The timers are encoded by a bit pattern which is described in the 3GPP documentation. An example is available in appendix A.

18.5 Power save mode

After the module has registered to the network and the RRC connection has been released, after T3324 timer has expired the module will enter the PSM. It is possible to query this state by configuring the +NPSMR URC to automatically output this status when it changes.

18.6 Ping

Issue the +UPING AT command to check if the module can send and receive data.

Check to see if the network can communicate to the internet, ping Google's DNS server:

```
AT+UPING="8.8.8.8"
```

To ping OpenDNS DNS server:

```
AT+UPING="208.67.222.222"
```

The URC response to the +UPING AT command will be issued after a few seconds. If the URC is +UUPINGER: 17, this means there is no active PDP context. Use the +CGDCONT and +CGACT AT commands to create and activate the PDP context.

☞ The first ping might fail because it can take a few seconds to connect to the base station. Use the +CSCON URC to show when the module is connected.

18.7 Echo server

For a more advanced check on sending data to an external server, send data to the u-blox echo server at echo.u-blox.com. Check the IP address for echo.u-blox.com before using the IP address reported in the following example.

Command	Response	Description
AT+USOCR=17	0 OK	Create a UDP socket.
AT+USOST=0,"195.34.89.241",7,5,"hello"	0,5 OK	Send data on socket 0.
	+UUSORF: 0,5	Receive data on socket 0.
AT+USORF=0	+USORF: 0,"195.34.89.241",7,5,"hello",0 OK	Request data from socket 0. Echo'd data received

19 Simple AT command example

Below is an example of AT commands and their responses in a scenario starting with the module turning on and registering to a network. It also shows pinging Google DNS and sending data to the u-blox echo server (The "OK" final result codes are removed from list to reduce clutter in the table).

Command	Response	Description
	`u-blox`	Welcome note
`AT+CGMR`		Get revision
`ATI9`		Get detailed revision
`AT+CGSN=1`		Get IMEI
`AT+CSCON=1`	`OK`	Turn on Base Station Connection State URC
`AT+CEREG=1`	`OK`	Turn on network registration state URC
`AT+NPSMR=1`	`OK`	Turn on PSM state URC
`AT+CFUN=1`	`OK`	Turn on module radio/modem
	`+CEREG: 0`	Not registering
`AT+CIMI`	`00101123456789`	Read IMSI from SIM
`AT+COPS=1,2,"00101"`	`OK`	Start manual registration to 00101 PLMN
	`+CEREG: 2`	Starting registration process, scan for Base stations
	`+CSCON: 1`	Found base station, RRC connection created
	`+CEREG: 1`	Registered to network
	`+CSCON: 0`	RRC released, now inside T3324 period
	`+NPSMR: 1`	Entered the Power Save Mode
`AT+UPING="8.8.8.8"`		Ping Google DNS
	`+CSCON: 1`	RRC connection to base station created
	`+UPING: "8.8.8.8",45,1234`	Ping result
	`+CSCON: 0`	RRC released, now inside T3324 period
`AT+USOCR=17`		Create UDP socket
	`0`	Socket "0" created
`AT+USOST=0,"195.34.89.241",7,5,"hello"`		Send 3 bytes to u-blox echo server on socket "0"
	`0,3`	Queued 3 bytes to be sent
	`+UUSORF:0,3`	Received 3 bytes on socket "0"
`AT+USORF=0`		Request 3 bytes from socket "0"
	`+USORF: 0,"195.34.89.241",7,5,"hello",0`	Response
	`+CSCON: 0`	RRC released, now inside T3324 period

20 Power profile

3GPP NB-IoT has been developed for low power applications which provide more than 10 years of operation. It is useful to view the power profile of the module when developing the application to understand the various profiles in each coverage class, and what this means for the battery.

20.1 Release Assistance example

Figure 5 shows a typical power profile of the SARA-N3 series modules sending 200 bytes in coverage class 1, with the Release Assistance feature disabled (`AT+NSOSTF` with 0x000 flag, or `AT+NSOST`)

1. Module wakes up and scans for a base station
2. Starts the RACH process and immediately gets an allocation
3. Sends 200 bytes using UDP message
4. The module waits for the RRC release by the eNodeB after its 20 s of inactivity

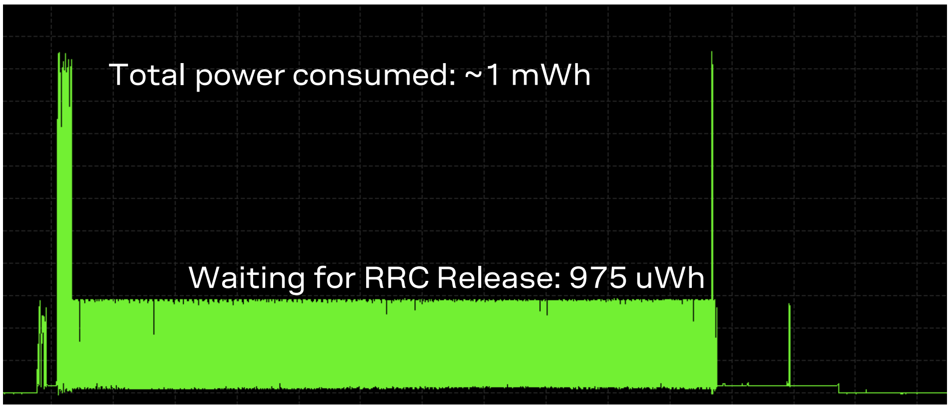


Figure 5: 200 bytes, ECL1, without the Release Assistance feature

Figure 6 shows a typical power profile of the module sending 200 bytes in coverage class 1, with the Release Assistance feature enabled (`+NSOSTF 0x200`).

1. The module wakes up and scans for a base station
2. Starts the RACH process and immediately gets an allocation
3. Sends 200 bytes using UDP message with RAI flag enabled
4. eNodeB releases RRC immediately

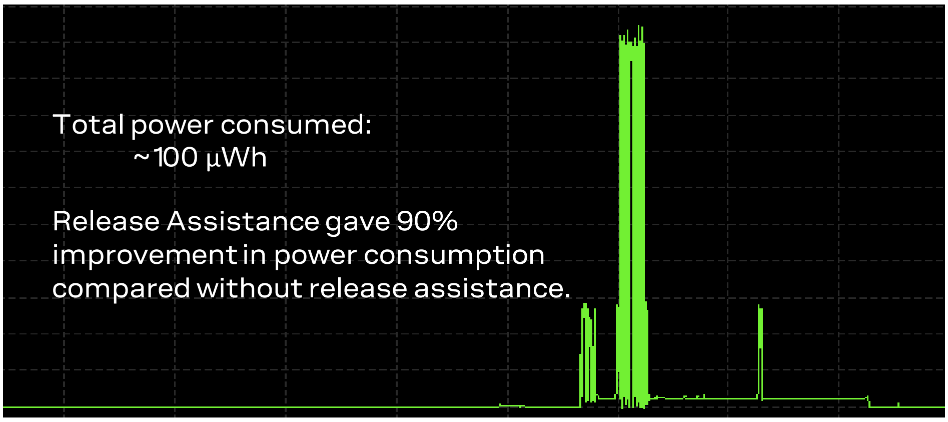


Figure 6: 200 bytes, ECL1, with the Release Assistance feature

20.2 Uplink allocations based on SNR

The UE is entirely under the control of the eNB and the network configuration. The UE must follow the network settings which are broadcast inside the System Information Blocks (SIB) and the allocations for the uplink and downlink data.

The SIB describes the method of attachment (RACH) and what repetitions the UE must use to first transmit to the base station for setting up the Radio Resource Connection (RRC). Once a RRC connection is made, the base station then uses the perceived SNR to configure the uplink allocations the UE will use to transmit the messages.

Because allocations for each uplink/downlink are dynamically set by the base station it is difficult to calculate the power consumption of a single message deployed in the field. Two examples of transmitting 200 bytes in two situations, good and bad SNR are reported in 20.2.1 and in 20.2.2. For both of these examples the RACH process was the same as this is defined in the SIB.

20.2.1 Good SNR, coverage class 2

Figure 7 shows an example profile of sending 200 bytes in coverage class 2 with “good” SNR. The first 6 transmit bursts are the random access procedure, which is specified by the SIB parameters. The next transmission bursts are from the UE sending the 200 byte message, according to the allocations provided by the base station.

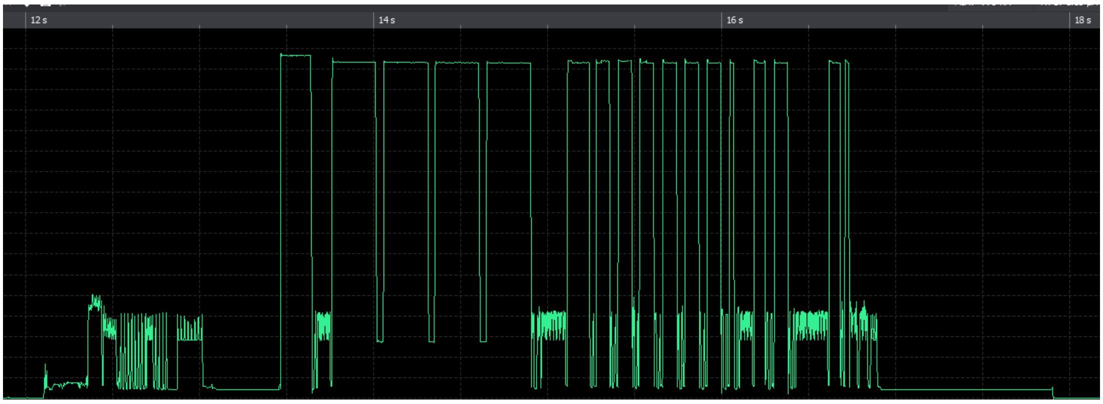


Figure 7: Good SNR, coverage class 2

The debug log for this example showed the Transport Block Size (TBS) allocated 43 bytes for each uplink chunk, with a repetition of just one. This allowed the UE to transmit the 200 byte message in just over 1 s, consuming 200 µWh.

20.2.2 Bad SNR, coverage class 2

Figure 8 shows is an example profile of sending 200 bytes in coverage class 2 with "bad" SNR. The first 6 transmit bursts are the random access procedure (RACH), which is specified by the SIB parameters. The next transmission bursts are from the UE sending the 200-byte message, according to the allocations provided by the base station.

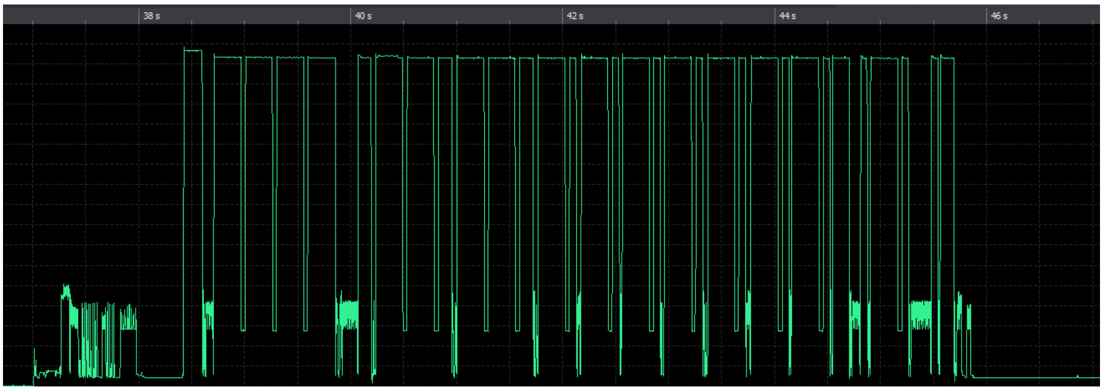


Figure 8: Bad SNR, coverage class 2

The debug log for this example showed the Transport Block Size (TBS) allocated 32 bytes for each uplink chunk, with a repetition of eight or four. This time the UE transmitted the 200-byte message in 5.5 s, consuming 1.07 mWh – five times that as before.

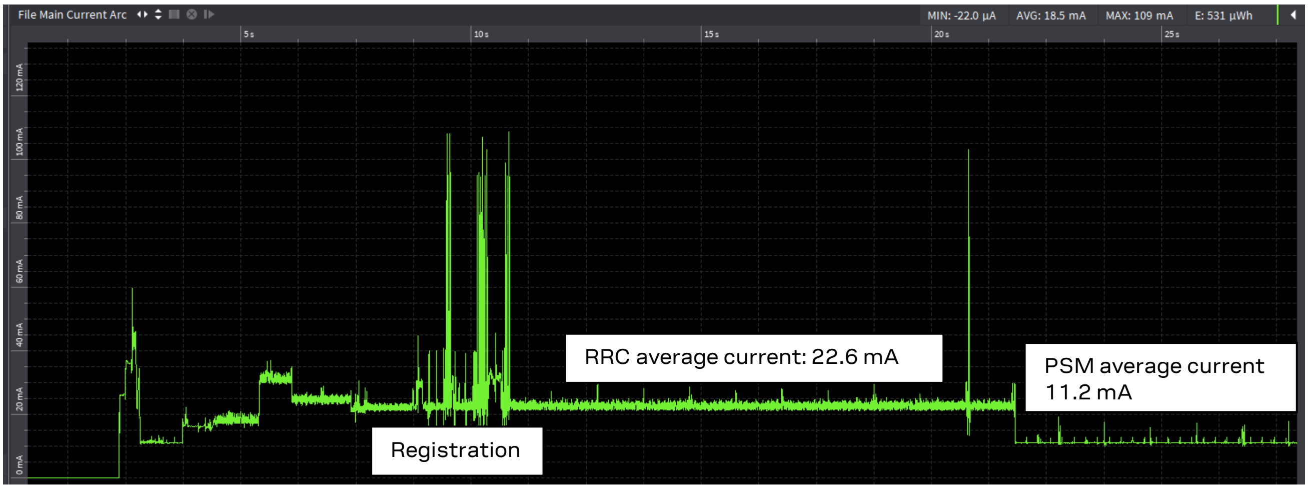
20.3 Power mode profiles

This section will show typical power profiles of the various power modes the SARA-N3 series modules support. All examples are showing the power profile of the module registration on a network in good conditions. PSM is enabled, with T3324 set to 0 s (immediate PSM after RRC).

These power profiles are for this particular module in this test network setup. These graphs are to show examples of how the module behaves and the different power state's current measurements.

20.3.1 Current consumption when +NVSETPM: 0

Figure 9 shows that the module's idle current is high as there are no power saving states it goes into.



RRC average current: 22.6 mA

PSM average current 11.2 mA

Registration

Figure 9: current consumption when +NVSETPM: 0

Even in PSM the module will be drawing 11.2 mA.

20.3.2 Current consumption when +NVSETPM: 1

In Figure 10 there is a slight improvement with the RRC connection average current and some minor drop before the registration. The biggest improvement is in PSM where the module drops into a state only consuming ~940 µA. This is because the UART interface has been turned off. The host application can still send AT commands as the first character of the command will wake the UART up.

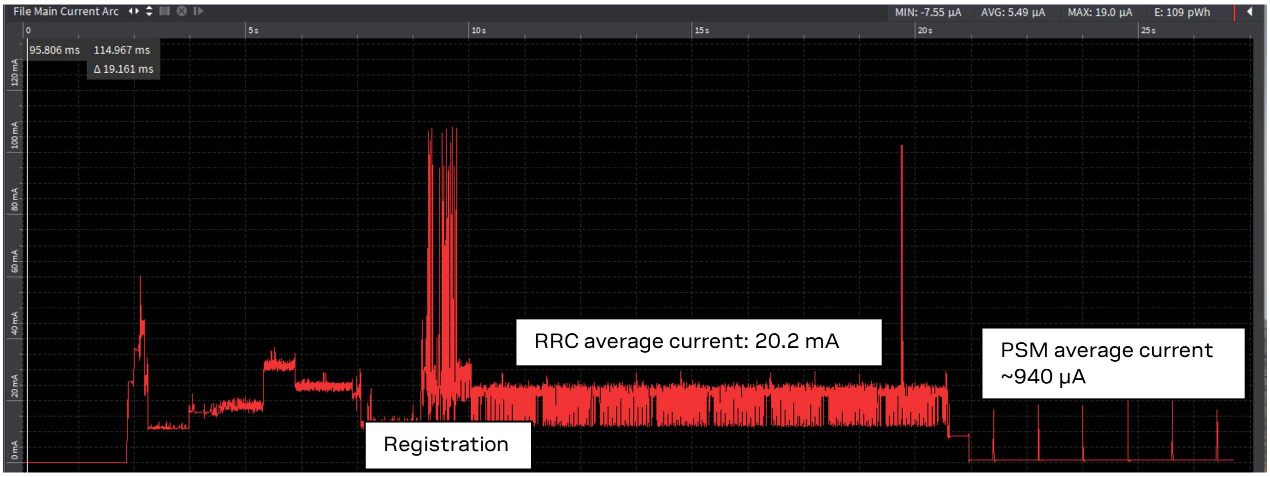


Figure 10: current consumption when +NVSETPM: 1

20.3.3 Current consumption when +NVSETPM: 2

Figure 11 shows PSM dropping to the module's 3 µA current. Also notice another drop of current before the registration has started. Once in PSM the module will need to be woken up by toggling PWR_ON line.

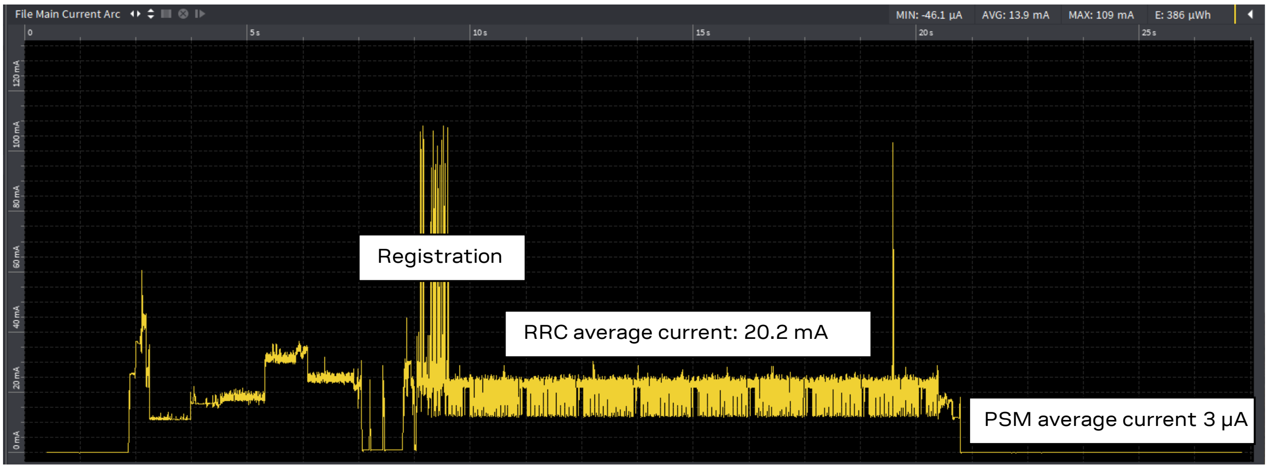
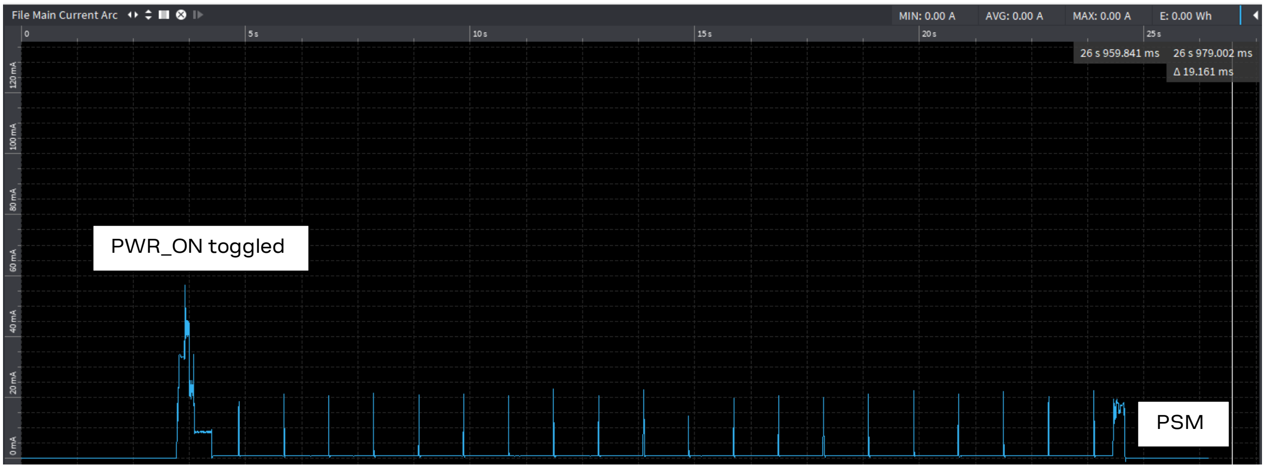


Figure 11: current consumption when +NVSETPM: 2

20.3.4 Current consumption when +NVSETPM: 2, waking up module with PWR_ON in PSM

Once the module is in PSM (`+NVSETPM: 2`) the host application will need to wake the module up by toggling the PWR_ON line. Figure 12 shows the module waking up and staying awake until the PM2 idle time value set by means of the `+NVSETPM2IDLETIME` AT command. The consumption is ~940 µA.

At this point the host application can send AT commands. When AT commands are running the UART will wake up and consume ~11 mA (not shown).



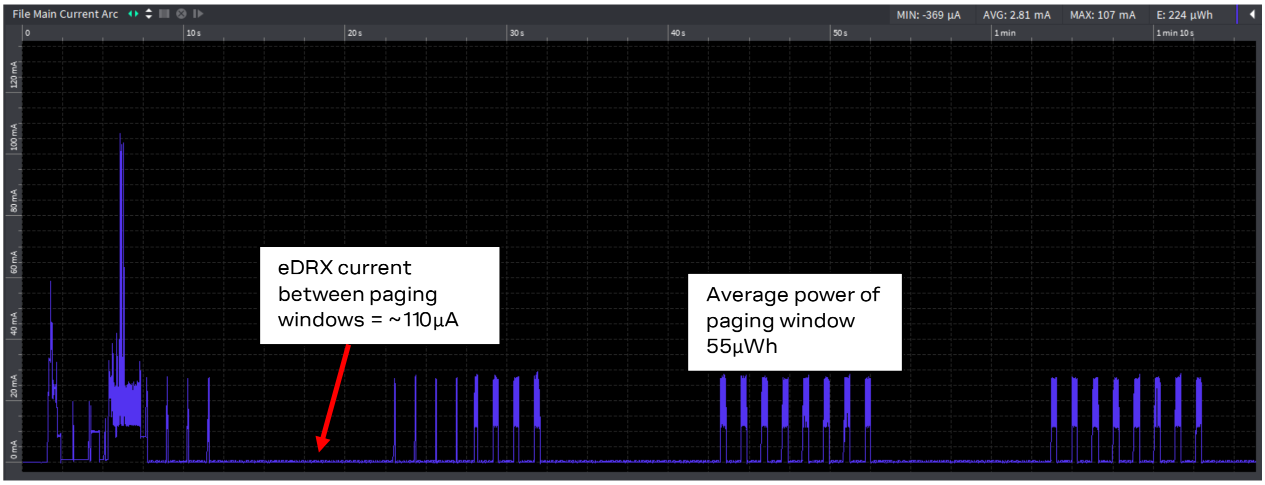
PWR_ON toggled

PSM

Figure 12: current consumption when module is waking up and staying awake until the PM2 idle time

20.3.5 Current consumption when +NVSETPM: 2, PSM off.

In Figure 13 the profile is displaying the eDRX cycles and paging windows when PSM is turned off.



eDRX current between paging windows = ~110µA

Average power of paging window 55µWh

Figure 13: current consumption during eDRX cycles and paging windows when PSM is turned off

21 Design for reduced power consumption

There are multiple factors which will contribute to the overall power consumption of the device. Some of them are listed in the next sections.

21.1 Hardware design

PCB layout, antenna matching, and location will have an effect to the overall interference received by the module.

The antenna matching will influence the current drawn when in Tx mode. Ideally the module should consume 230 mA for +23 dBm. Antenna positioning in the product will also have a role in the overall RF performance.

RF interference to the module will increase the number of repetitions the module is configured to use and therefore increase the Tx and Rx time.

21.2 Release Assistance

The application developers should consider the use of the Release Assistance feature. This drastically reduces the power consumed by the module when sending UDP data.

Applications could consider sending a special “service” type message to the cloud server which does not use the Release Assistance feature, letting the cloud service know that the UE will be still “online” for any downlink message that are required.

21.3 PSM and eDRX

Depending on the network configuration and permissions by the mobile network operator, the PSM and eDRX timers should be configured for the application.

Applications which only send a message once a day and do not expect to have any downlink messages could configure eDRX to be off and enter directly the PSM.

Applications which require an acknowledgment back to the device could consider still using the Release Assistance feature but configure a long T3324 timer and eDRX – as the UE will still be able to be contacted within T3324 period before it goes into PSM.

21.4 Minimizing coverage Level 2

Network operators should provide enough coverage to allow devices to be mostly in coverage class 0 or 1. Depending on the NB-IoT deployment, the network could have large areas, or devices located in deep locations which unfortunately mean they operate in coverage class 2.

Coverage class 2 uses high repetitions for the RACH process and higher coding schemes when transmitting data and therefore fundamentally consumes more power than it would in the other coverage classes.

22 RF receiver and transmitter testing

The +UTEST AT command is used for non-signaling testing. The +UTEST AT command is intended for use in production testing. The manufacturer of the host device integrating the module will implement the test to verify proper assembly of the involved parts in production line. The +UTEST AT command is not expected to be used by the customer of the final product which contains the SARA-N3 series module.

⚠ The usage of the +UTEST AT command shall be restricted to controlled (shielded chamber/box) environments and for test purposes only.

⚠ u-blox assumes no responsibility for the inappropriate use of the +UTEST AT command.

⚠ Improper usage of the +UTEST AT command on a real network could disturb other users and the network itself.

There are two modes of operation for the SARA-N3 series modules; Rx mode for receiver testing and Tx mode for transmitter testing.

To start using the +UTEST AT command, enable the non-signalling (or test) mode, issuing the AT+UTEST=1 command.

☞ The module must be deregistered from the network before enabling the non-signalling mode. Issue the AT+CFUN=0 command to deregister from the network and set the module to minimum functionality, otherwise the "+CME ERROR: Operation not allowed" or "+CME ERROR: 3" (depending on the +CMEE AT command setting) error result code is returned.

22.1 Rx mode

For RX testing, set the frequency and time over which the measurement is to be taken. A continuous waveform (CW) signal needs to be provided to the module for it to measure. Use an offset from center frequency of RF input signal. For example, to test EARFCN=2525 of LTE band 5, which corresponds to channel=881.50 MHz, set RF DL channel = 881.50 + 50 kHz = 881.55 MHz.

The minimum recommended time for the measurement in Rx mode is 10 milliseconds. For any lower value, the reported Rx measurement could be not reliable.

22.2 Tx mode

For Tx testing, set the frequency, power level and time over which the transmitter is enabled for. A power meter can be used to measure the output power level of the module.

22.3 Exiting test mode

Send the AT+UTEST=0 command to turn off the non-signaling mode and to return to normal operation. Powering off or resetting the module will also exit the non-signaling mode.

23 Firmware update via uFOTA

SARA-N310 implements the u-blox uFOTA solution based on LwM2M. It is possible to configure the module's poll timer for when the module checks the uFOTA server for new firmware version.

When the feature is enabled and a new package is available through a uFOTA "campaign", the module will automatically download the uFOTA update and provide URCs about its progress.

☞ The module's firmware is not updated automatically when the download has completed. The host application must start the upgrade process step by sending an AT command.

☞ For more details on the device files and settings after the firmware update, see Appendix B.

23.1 Configuration

There are three steps for configuring uFOTA:

1. Set uFOTA IP address
2. Set the uFOTA poll timer
3. Enable uFOTA

To set the IP address issue the `AT+UFOTACONF=0,<ip_addr>[,<port>]` command. The port value is optional, and the default value is 5683, and 5684 for DTLS secure connection.

To set the poll timer issue the `AT+UFOTACONF=2,<period>` command. By default the period value is set to seven days (604800 s).

To enable uFOTA service issue the `AT+UFOTACONF=1,1` command.

23.2 Monitoring

uFOTA status can be monitored via the uFOTA status `+UFOTASTAT` URC. To enable `+UFOTASTAT` URC issue the `AT+UFOTASTAT=1` command. It is enabled by default.

Use the `+UFOTASTAT` URC to detect problems with the download process and to know when the module is ready to be upgraded.

As is the case with LwM2M clients, the uFOTA client sends registration update messages to the server after regular intervals. The timer value is controlled by `+UFOTACONF` command. It has opcodes for configuration of two timers for uFOTA service, one is the lifetime timer. (registration update is triggered before this expires).

If the uFOTA client is unable to send a registration update message to uFOTA server before the lifetime timer expires due to any reason like connection lost, the connection with uFOTA server will break. New connection will be established again automatically when the connection is live again and the download of the package will start again at the beginning.

☞ If the module loses power (Vcc) or is reset (`+CFUN` or RESET_N pin) while a new firmware package is being downloaded, the next time the module checks the uFOTA server, it will have to start downloading the firmware from the beginning.

☞ If the `+UFOTASTAT` URC does not report a failure, the uFOTA process will continue even through minor drops in cellular connectivity. If the module's cellular connection is disconnected, `+UFOTASTAT` URC will report a failure and the uFOTA process will have to restart from the beginning as the LwM2M registration will need to be sent again when the cellular connection has recovered.

23.3 Starting firmware upgrade process

Once the uFOTA has successfully downloaded the new upgrade package, the FW install can be started. Use the `+UFWINSTALL` AT command to reboot the module and start the firmware upgrade process. If the module is turned off or rebooted before the process has completed, the firmware upgrade process will repeat again until the firmware has been successfully installed.

Once the firmware has been updated a `+UUFWINSTALL` URC will notify the final results of the operation.

☞ Firmware upgrading process can take up to 5 minutes.

23.4 uFOTA campaigns

Please contact u-blox FAE on how to start a uFOTA campaign for upgrading SARA-N310.

Appendix

A T3412 and T3324 timer values

T3412 timer (GPRS timer 3)

The purpose of the GPRS timer 3 information element is to specify GPRS specific timer values. The GPRS timer 3 is a type 4 information element with a 3-octet length. The GPRS timer 3 information elements are coded as described in 3GPP TS 24.008 [4], figure 10.5.147a and table 10.5.163a.

Bits 5 to 1 represent the binary coded timer value. Bits 6 to 8 define the timer value unit for the GPRS timer as follows:

```
BIT   8 7 6
      0 0 0 value is incremented in multiples of 10 minutes
      0 0 1 value is incremented in multiples of 1 hour
      0 1 0 value is incremented in multiples of 10 hours
      0 1 1 value is incremented in multiples of 2 seconds
      1 0 0 value is incremented in multiples of 30 seconds
      1 0 1 value is incremented in multiples of 1 minute
      1 1 0 value is incremented in multiples of 320 hours (Note 1)
      1 1 1 value indicates that the timer is deactivated (Note 2)
```

Example: "**010**00111" = 7 **x10 hours** = 70 hours

☞ NOTE 1: This timer value unit is only applicable to the T3312 extended value IE and the T3412 extended value IE (see 3GPP TS 24.301 [5]). If it is received in an integrity protected message, the value shall be interpreted as multiples of 320 hours. Otherwise the value shall be interpreted as multiples of 1 hour.

☞ NOTE 2: This timer value unit is not applicable to the T3412 extended value IE. If this timer value is received, the T3412 extended value IE shall be considered as not included in the message (see 3GPP TS 24.301 [5]).

T3324 timer (GPRS timer 2)

The purpose of the GPRS timer 2 information element is to specify GPRS specific timer values, e.g. for the timer T3302 or timer T3319. The GPRS timer 2 is a type 4 information element with a 3-octet length. The GPRS timer 2 information elements are coded as shown in figure 10.5.147 and table 10.5.163 in 3GPP TS 24.008 [4].

Bits 5 to 1 represent the binary coded timer value. Bits 6 to 8 define the timer value unit for the GPRS timer as follows:

```
BIT   8 7 6
      0 0 0 value is incremented in multiples of 2 seconds
      0 0 1 value is incremented in multiples of 1 minute
      0 1 0 value is incremented in multiples of deci-hours
      1 1 1 value indicates that the timer is deactivated
```

Example: "**001**00100" = 4 **x1 minute** = 4 minutes

B FW update: device files and settings

Table 2 summarizes each of the update methods and their impact on user files and settings.

Item	FW delta package via FOAT (applied with +UFWUPD)	EasyFlash	FOTA/uFOTA (applied with +UFWINSTALL)
“USER” tagged files stored in user file system	Files are preserved.		
User NVM settings	AT commands setting are preserved.		
User certificate and private keys	User certificate and private keys are preserved.		

Table 2: summary of each firmware update method and impact to device file and settings.

For more details on the `+UFWUPD` and `+UFWINSTALL` AT commands, see the SARA-N2 / SARA-N3 series AT commands manual [2].

C Glossary

Abbreviation	Definition
AMQP	Advanced Message Queue Protocol
APN	Access Point Name
CDP	Connected Device Platform
DC	Data Channel
DCE	Data Communications Equipment
DL	Downlink
DTE	Data Terminal Equipment
EARFCN	Extended Absolute Radio-Frequency Channel Number
GPRS	General Packet Radio Service
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Station Identity
IP	Internet Protocol
MCS	Message Coding Scheme
MNO	Mobile network operator
MO	Mobile Originated
MT	Mobile Terminated
NB-IoT	Narrow Band Internet of Things
PCI	Physical Channel ID
PDN	Packet Data Network
PLMN	Public land mobile network
RF	Radio frequency
Rx	Receiver
SIM	Subscriber Identity Module
SMS	Short Message Service
SNR	Signal to Noise Ratio
TAU	Tracking Area Update
Tx	Transmitter
UE	User Equipment
UL	Uplink
URC	Unsolicited Result Code
UUID	Unique User Identification

Table 3: Explanation of the abbreviations and terms used

Related documentation

[1] u-blox SARA-N3 series data sheet, UBX-15025564

[2] u-blox SARA-N2 / SARA-N3 series AT commands manual, UBX-16014887

[3] 3GPP TS 36.321 Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification

[4] 3GPP TS 24.008 - Annex G (informative): 3GPP specific cause values for mobility management

[5] 3GPP TS 24.301 - Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3

[6] 3GPP TS 36.331 - Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification

[7] Open Mobile Alliance (OMA) - Lightweight Machine to Machine Technical Specification

☞ For regular updates to u-blox documentation and to receive product change notifications, register on our homepage (www.u-blox.com).

Revision history

Revision	Date	Name	Comments
R01	29-Jun-2020	pwar	Initial release
R02	19-Nov-2020	pwar	Added “Secure data”, “MQTT-SN”, “End user test” sections. “FOTA” section renamed to “uFOTA” and improved. Minor improvements and changes to section 5.4.
R03	15-Oct-2021	pwar	Added the +NVSETRELEASEVERSION AT command in section 5. Improved examples for PDN contexts (+CIPCA/+CFGDFTPDN/+CGDCONT). Removed SARA-N300 from the document applicability. Removed Datagram message with Huawei OceanConnect IoT platform.

Contact

For complete contact information, visit us at www.u-blox.com.

u-blox Offices

North, Central and South America

u-blox America, Inc.

Phone: +1 703 483 3180
Email: info_us@u-blox.com

Regional Office West Coast:

Phone: +1 408 573 3640
Email: info_us@u-blox.com

Technical Support:

Phone: +1 703 483 3185
Email: support@u-blox.com

Headquarters
Europe, Middle East, Africa

u-blox AG

Phone: +41 44 722 74 44
Email: info@u-blox.com
Support: support@u-blox.com

Asia, Australia, Pacific

u-blox Singapore Pte. Ltd.

Phone: +65 6734 3811
Email: info_ap@u-blox.com
Support: support_ap@u-blox.com

Regional Office Australia:

Phone: +61 3 9566 7255
Email: info_anz@u-blox.com
Support: support_ap@u-blox.com

Regional Office China (Beijing):

Phone: +86 10 68 133 545
Email: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office China (Chongqing):

Phone: +86 23 6815 1588
Email: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office China (Shanghai):

Phone: +86 21 6090 4832
Email: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office China (Shenzhen):

Phone: +86 755 8627 1083
Email: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office India:

Phone: +91 80 405 092 00
Email: info_in@u-blox.com
Support: support_in@u-blox.com

Regional Office Japan (Osaka):

Phone: +81 6 6941 3660
Email: info_jp@u-blox.com
Support: support_jp@u-blox.com

Regional Office Japan (Tokyo):

Phone: +81 3 5775 3850
Email: info_jp@u-blox.com
Support: support_jp@u-blox.com

Regional Office Korea:

Phone: +82 2 542 0861
Email: info_kr@u-blox.com
Support: support_kr@u-blox.com

Regional Office Taiwan:

Phone: +886 2 2657 1090
Email: info_tw@u-blox.com
Support: support_tw@u-blox.com