LE910Cx

Software User Guide

1VV0301556 Rev. 14
2023-12-05
Released
Confidential

Technical Documentation

Contents

Contents 2

1 Applicability Table 7

2 Introduction 8

2.1 Scope 8

2.2 Audience 8

2.3 Contact Information, Support 9

2.4 Conventions 9

2.5 Terms and conditions 10

2.6 Disclaimer 10

3 LE910Cx Variants 11

4 High-Level SW Architecture 12

4.1 Architecture Based on Linux 12

4.2 Architecture Based on ThreadX 13

5 Functional Description 14

5.1 General Functionality and Main Features 14

5.2 Application System Overview 14

Memory Configuration 15

LE (Linux) 15

TX (ThreadX) 15

Partition Layout 15

LE (Linux) 16

TX (ThreadX) 18

RAM Memory 20

LE (Linux) 20

TX (ThreadX) 20

Customer Application – Storage and Configuration 20

Power Up Time 21

LE (Linux) 21

TX (ThreadX) 21

Power Up Sequence 21

Location Subsystem 22

Application Development Environment 22

Random Number Generator 22

Wake up Events 23

Wake up Event Examples 25

CFUN=0: Call, SMS, #QSS, +CALA 25

CFUN=4: #QSS, +CALA 28

CFUN=5: Call, SMS, +CALA 30

SPI 34

LE (Linux) 35

TX (ThreadX) 35

GPIO 35

Serial Interfaces 37

LE (Linux) 37

Apps to External MCU 37

Apps to Modem 37

Audio 39
LE (Linux) 39
TX (ThreadX) 40
UART 41
LE (Linux) 41
USB Interface 42
LE (Linux) 42
Changing USB Composition to MBIM only USB Composition 43
TX (ThreadX) 44
HSIC Interface 44
Ethernet Interface 45
SD/MMC Interface 46
RTC 46
Time Services 46
Coordinated Universal Time (UTC) 47
Time Zones 47
Daylight Saving Time (DST) 47
Time Update 47
System Time Base 48
Data Connection 48
QCMAP_CLI 48
WatchDog 48
Hardware Watchdog Characteristics 49
Watchdog Connectivity 50
A7 Apps Processor Watchdogs 50
MPSS Watchdog 51
Power Management 52
Performance Build (Linux) 55
5.3 Basic Operations 57
Command Syntax 57
Command Response Timeout 57
Basic AT Commands 61
AT Error Report Format 61
RAT and Band Selection 61
RAT Selection 61
Band Selection 62
SIM/USIM Management 62
SIM Presence and PIN Request 62
Enter PIN Code 63
Enter PUK Code 64
SIM Status 64
SIM Detection Mode 65
SIM/USIM Access File 67
MSISDN 67
Preferred Operator List 68
Network Checking 70
Network Survey 70
BCCH Survey 71
Network Information 71
Network Status 71

Circuit Service Network Registration Status in UTRAN/E-UTRAN 71
Packet Service Network Registration Status in UTRAN 72
Packet Service Network Registration Status in E-UTRAN 73
Network Operator Identification 74
Signal Strength and Quality 75
Extended Signal Quality 76
Fast Network Status Check 77
3G Network 79
4G Network 80
Voice Call Establishment – Originate 80
Dialing a Phone Number 81
Disconnect a Call 81
Answering an Incoming Call 81
Set Microphone Mute 81
5.4 Advanced Operations 82
Call Management 82
Identifying the Call Type 82
Identify the Caller 83
Calling Line Indication 84
CLIR Service Status 84
Restrict/Allow Caller Line ID Indication 85
Call Barring Control 85
Lock/Unlock the Module 86
Call Barring Service Status 86
Bar/Unbar All Incoming Calls 87
Bar/Unbar Incoming Calls in International Roaming 88
Bar/Unbar All Outgoing Calls 89
Bar/Unbar All Outgoing International Calls 90
Bar/Unbar All Outgoing International Calls Except to Home Country 91
Unbar All Calls 92
DTMF Tones 93
DTMF Transmission on Uplink 93
SMS Management 94
Select SMS Format Type 94
Set Text Mode Parameters 95
Character Sets 97
IRA Character Set 98
UCS2 Character Set 99
Read SMSC Number 99
Set SMSC Number 100
Send an SMS 100
Select/Check SMS Storage Type 103
Store an SMS 103
Send a Stored SMS 105
Delete an SMS 106
Read an SMS 107
SMS Status 107
Cell Broadcast Service 108
GNSS Management 110
Introduction 110

LE910Cx Serial Ports 110
WGS84 111
NMEA 0183 111
GGA – Global Position System Fixed Data 112
GLL - Geographic Position - Latitude/Longitude 114
GSA - GNSS DOP and Active Satellites 115
GSV - GNSS Satellites in View 116
RMC - Recommended Minimum Specific GNSS Data 118
VTG - Course over Ground and Ground Speed 119
GNS - GNSS Fix Data 119
GRS - Range Residuals 120
DTM - Datum Reference Information 122
Checking GNSS Device Functionality 122
Controlling GNSS Receiver 123
Power Control of GNSS Receiver 123
GNSS Reset 123
GNSS Parameters Save 124
Restore GNSS Parameters 124
Read Acquired GNSS Position 125
Blocking access to AT and USB interfaces 127
Settings to ban access to the module from all interfaces 127
Example of AppZone code 127
5.5 Packet Switched Data Operations 129
USB Tethering Connection 129
Dial-Up Networking 129
Standard ECM/RNDIS 129
MBIM/RmNet 130
Socket AT Commands 130
Configuring Embedded TCP/IP Stack 130
Activating PDP Context 131
Online Mode Operation 132
Open Socket Connection 132
Resume Suspended Connection 132
Close the Connection 133
Command Mode Operation 133
Open Socket Connection 133
Send User Data 134
Receive User Data 134
Close the Connection 135
Server Mode(Socket Listen) Operation 135
Create Listen Socket 135
Accept Incoming Connection 135
Command Mode Operation 136
SSL AT Commands 136
Configuration 136
Online Mode Operation 138
Open Secure Socket Connection 138
Exchanging User Data 138
Close the Connection 139
Command Mode Operation 139

Open Secure Socket Connection 139
Send User Data 140
Receive User Data 141
Close the Connection 141
HTTP AT Commands 141
Configuration 141
Query 142
Request 142
Receive 143
FTP AT Commands 143
Configuration 143
Open Connection 144
Online Mode Operation 144
Uploading 144
Appending 145
Downloading 145
Command Mode Operation 146
Uploading 146
Appending 147
Downloading 147
Close the Connection 148
Email AT Commands 148
Configuration 148
Sending an Email 149
IoT Platform AT Commands 149
Connect to IoT Service 150
API and AT Commands 151
Disconnect from IoT Service 152
IoT Platform AT Commands List 152
Data Concurrency 153
LE (Linux) 153
Same PDN 153
Different PDNs 153
TX (ThreadX) 155
Same PDN 155
Different PDNs 155
Maximum Number of PDN Contexts 156

6 Performance Measurements (Linux)....157
6.1 Interrupt Latencies 157
6.2 Memory Bandwidth and Latencies 160

7 Service and Firmware Update 164
7.1 Firmware Update 164
TFI Update 164
XFP Update 165

8 Acronyms and Abbreviations 168

9 Related Documents 170

10 Document History 171

1 Applicability Table

Table 1: Applicability Table

Products
LE910C1-NA
LE910C1-NS
LE910CX-NF
LE910CX-EU
LE910CX-AP
LE910CX-LA
LE910C4-CN
LE910C1-SV
LE910C1-SA
LE910C1-ST
LE910C1-APX
LE910C1-EUX
LE910C1-SAX
LE910C1-SNX
LE910C1-SVX
LE910Cx-WWX

2 Introduction

2.1 Scope

This document introduces the Telit LE910Cx module as well as presents possible and recommended Software solutions useful for the development of a product based on the LE910Cx module. All the features and solutions described in this document apply to all LE910Cx variants, where "LE910Cx" refers to the variants listed in the applicability table.

If a specific feature applies to a specific product, it will be clearly highlighted.

Note: The description text "LE910Cx" refers to all modules listed in the Applicability Table.

All the basic functions of a wireless module will be considered in this document; for each one of them a valid hardware solution will be suggested and usually incorrect solutions and common errors to be avoided will be highlighted. This document cannot embrace every hardware solution or every product that may be designed. Avoiding invalid solutions must be considered mandatory. Where the suggested hardware configurations are not to be considered mandatory, the information provided should be used as a guide and starting point for the proper development of the product with the Telit LE910Cx module.

Note: The integration of the GSM/GPRS/EGPRS/WCDMA/HSPA+/LTE LE910Cx cellular module within the user application must be done according to the design rules described in this manual.

The information presented in this document is believed to be accurate and reliable. However, no responsibility is assumed by Telit Communication S.p.A. for its use, as well as any infringement of patents or other rights of third parties which may arise from its use. No license is granted by implication or otherwise under any patent rights of Telit Communication S.p.A. other than for circuitry embodied in Telit products. This document is subject to change without notice.

2.2 Audience

This document is intended for Telit customers, especially system integrators, about to implement their applications using the Telit LE910Cx module.

2.3 Contact Information, Support

For technical support and general questions, e-mail:

- TS-EMEA@telit.com
- TS-AMERICAS@telit.com
- TS-APAC@telit.com
- TS-SRD@telit.com
- TS-ONEEDGE@telit.com

Alternatively, use: https://www.telit.com/contact-us/

Product information and technical documents are accessible 24/7 on our website: https://www.telit.com

2.4 Conventions

Note: Provide advice and suggestions that may be useful when integrating the module.

Danger: This information MUST be followed, or catastrophic equipment failure or personal injury may occur.

ESD Risk: Notifies the user to take proper grounding precautions before handling the product.

Warning: Alerts the user on important steps about the module integration.

All dates are in ISO 8601 format, that is YYYY-MM-DD.

2.5 Terms and conditions

Refer to https://www.telit.com/hardware-terms-conditions/.

2.6 Disclaimer

3 LE910Cx Variants

The LE910Cx is available in a variety of configurations, however, it is classified into two categories based on the operating system. The first is based on Linux, while the second is based on ThreadX.

The basic operating system for each variant is listed below.

Table 2: Basic OS for Each Variant

Variants	OS
LE910C1-NA	Linux
LE910C1-NS	Linux
LE910Cx-NF	Linux
LE910Cx-EU	Linux
LE910Cx-AP	Linux
LE910Cx-LA	Linux
LE910C4-CN	Linux
LE910C1-SV	Linux
LE910C1-SA	Linux
LE910C1-ST	Linux
LE910C1-APX	ThreadX
LE910C1-EUX	ThreadX
LE910C1-SAX	ThreadX
LE910C1-SNX	ThreadX
LE910C1-SVX	ThreadX
LE910Cx-WWX	ThreadX

Therefore, when referring to the document, please check for differences depending on the OS. Unless otherwise specified, the descriptions apply to both Linux and ThreadX variations.

4 High-Level SW Architecture

4.1 Architecture Based on Linux

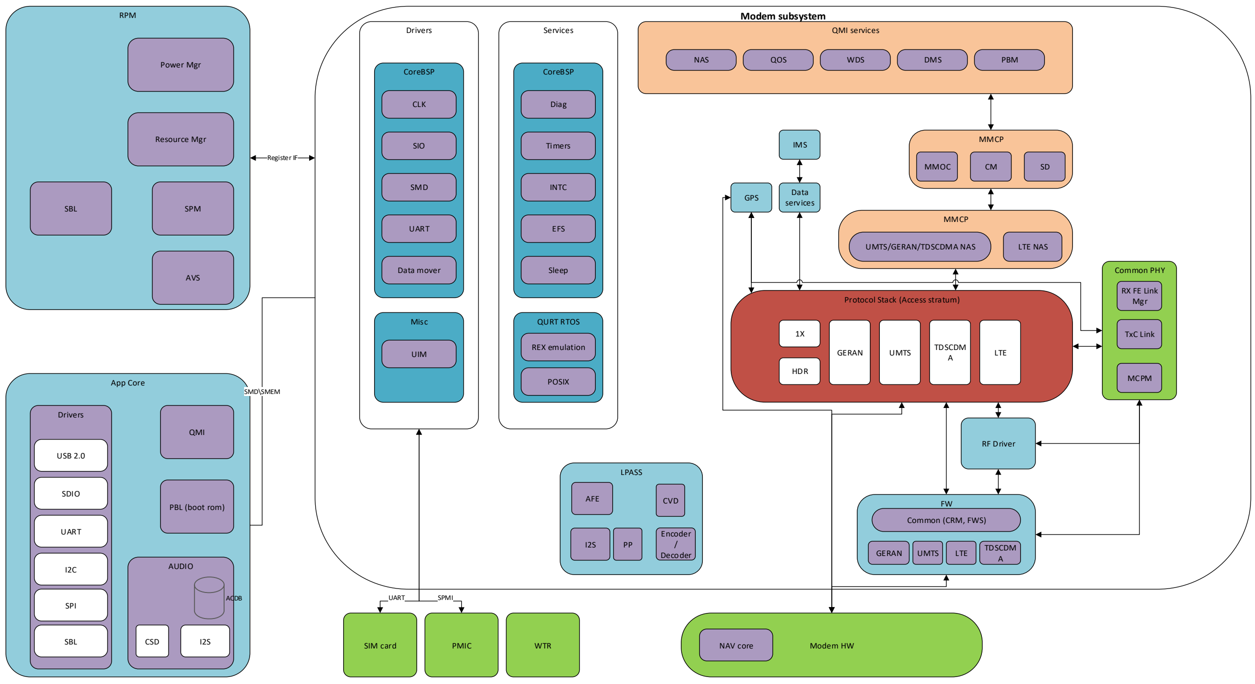


Figure 1: General System Architecture

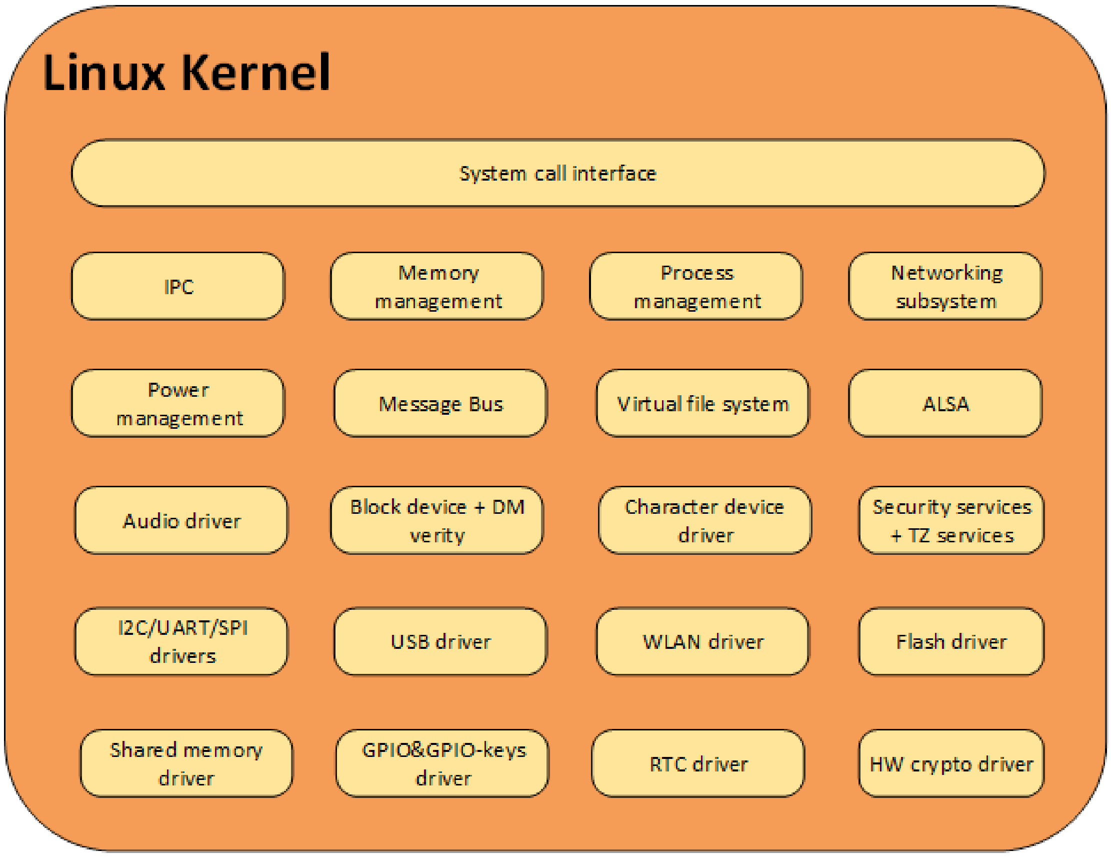


Figure 2: Linux Kernel Components

4.2 Architecture Based on ThreadX

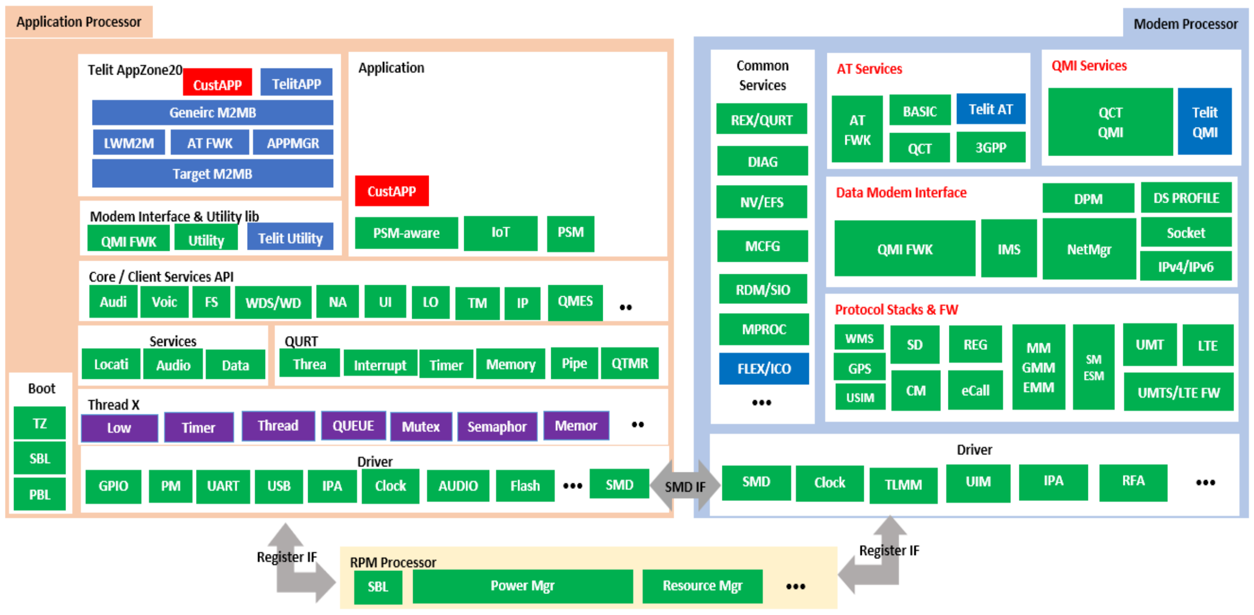


Figure 3: Architecture Based on ThreadX

5 Functional Description

5.1 General Functionality and Main Features

The LE910Cx family of cellular modules features LTE and multi-RAT modem together with an on-chip powerful application processor and a rich set of interfaces.

The main functions and features are listed below:

- Multi RAT cellular modem for voice and data communication
  - LTE FDD/TDD Cat4 (150/50Mbps DL/UL)
  - GSM/GPRS/EDGE
  - WCDMA up to DC HSPA+ Rel. 9
  - Support for SIM profile switching
- Digital audio and analog audio codec
- Application processor to run customer application code
  - 1.3 GHz Cortex-A7 with Linux version 3.18
  - Flash + DDR are large enough to allow for customer's software applications
- High-speed serial interfaces:
  - USB, HSIC
- Tools for firmware update (TFI)
- Stream download protocol (SDL)
- FOTA (Legacy AT FOTA)
- SGMII (optional) for external Ethernet transceiver
- SDIO for (optional) external Wi-Fi transceiver

Note: Small memory of Linux products, ThreadX products don't support Wi-Fi.

Note: ThreadX products do not support HSIC, SGMII, or SDIO.

5.2 Application System Overview

The Application Processor is a 32-bit ARM Cortex-A7 with a clock speed of 1.3 GHz that runs Linux. The application processor comes pre-installed with the following software:

- 32bit Cortex-A7@1.3GHz running the Linux kernel 3.18.
- Telit Unified AT command set, backward compatible with LE920, which is the main control interface to offer features from the variant not enabled to the application, with the following:
  - Hayes standard AT command set
  - ETSI GSM 07.07 specific AT command and GPRS specific commands
  - ETSI GSM 07.05 specific AT commands for SMS (Short Message Service) and CBS (Cell Broadcast Service)

  - Control of pre-integrated Firmware Update Agent (Harman)
  - Antenna diagnostics
- Firmware Over-The-Air (FOTA) update supporting selective update. Backward compatible with LE910.
- Operator specific Device management client, backward compatible with LE910
- SPI device driver for user-space access of the SPI device, including slave to master interrupt, backward compatible with LE910
- GPIO interrupts driver for user-space to listen to interrupts on selected user GPIOs, backward compatible with LE910
- 2G/3G/4G and GNSS jamming detection
- Audio subsystem, backward compatible with LE910
  - PCM digital audio IO
  - Limited support for DTMF detection.
- FTM support, backward compatible with LE910
- Vocoder support and processing
  - GSM vocoders (EFR/HR/FR), all rates
  - AMR-NB, AMR-WB, all rates
  - VoLTE
  - Configurable noise suppressor, echo canceller, and processing chain
- Pre-integrated Wi-Fi driver via SDIO (QualcommA9377)
  - Wi-Fi can be controlled through AT command. For more information refer to AT Commands Reference Guide.

Note: ThreadX products:

- Do not support SDIO.
- Do not support SPI interrupt from the slave.

Memory Configuration

The LE910Cx memory configuration is as follows:

LE (Linux)

- Extended memory: 512Mbyte Flash /256Mbyte DDR
- Regular memory: 256Mbyte Flash /256Mbyte DDR
- Small memory: 256Mbyte Flash /128Mbyte DDR

TX (ThreadX)

128Mbyte Flash /128Mbyte DDR

Partition Layout

The below tables show the flash storage allocation on a partition basis.

Note: Flash allocation size includes the UBI container overhead (usually at 15% of the partition). Partitions marked as Telit: These are specific partitions used by Telit.

LE (Linux)

Table 3: LE Extended Memory

Partition name		Permission	Partition size (MB)
sbl		RO	1.25
mibib		RO	1.25
telit		RO	3.25
efs2		RO	12
tz		RO	1
telit		RO	1
rpm		RO	0.5
telit		RO	0.5
aboot		RO	1
telit		RO	1
boot		RO	11.75
telit		RO	0.625
modem		RO	57.875
telit		RO	0.375
recovery		RO	11.75
fota		RO	0.5
recoveryfs		RO	10.25
telit		RO	0.5
telit		RO	3.5
Customapps (0x77e0000 ~ 0x153e0000)		RW	220
System (0x153e0000 ~ 0x20000000)	rootfs	RO	172.125
	cachefs	RW	
	usrfs	RW	
Summary			512

Note: Available memory: About 180 MB of memory is available for customer applications. (custom apps partition).

Table 4: LE Regular Memory

Partition name		Permission	Partition size (MB)
sbl		RO	1.25
mibib		RO	1.25
telit		RO	3.25
efs2		RO	12
tz		RO	1
rpm		RO	0.5
aboot		RO	1
boot		RO	11.75
telit		RO	0.625
modem		RO	57.875
telit		RO	0.375
recovery		RO	11.75
fota		RO	0.5
recoveryfs		RO	10.25
telit		RO	0.5
System (0x71e0000 ~ 0x10000000)	rootfs	RO	142.125
	cachefs	RW	
	usrfs	RW	
Summary			256

Note: Available memory: About 7 MB of memory is available for customer applications.

Table 5: LE Small Memory

Partition name		Permission	Partition size (MB)
sbl		RO	1.25
mibib		RO	1.25
telit		RO	3.25
efs2		RO	12
tz		RO	1
rpm		RO	0.5
aboot		RO	1
boot		RO	11.75
telit		RO	0.625
modem		RO	47.875
telit		RO	0.375
recovery		RO	11.75
fota		RO	0.5
recoveryfs		RO	20.25
telit		RO	0.5
System (0x71e0000 ~ 0x10000000)	rootfs	RW	142.125
	cachefs	RW	
	usrfs	RW	
Summary			256

TX (ThreadX)

Table 6: TX (ThreadX)

Partition name	Permission	Partition size (MB)
SBL	RO	1.25
MIBIB	RO	1.25
Telit	RO	2.75
telit	RO	5.5
Telit	RO	0.5

Partition name	Permission	Partition size (MB)
EFS2	RO	12
TZ	RO	2.25
telit	RO	2.25
DEVCFG	RO	0.375
APDP	RO	0.375
MSADP	RO	0.5
SEC	RO	0.25
MBA	RO	0.5
ACDB	RO	0.5
RPM	RO	0.375
telit	RO	0.375
QDSP	RO	47.25
APPS	RO	7
telit	RO	7
telit	RO	25.875
Cache_APPS	RO	0.25
Cache_ACDB	RO	0.25
misc	RO	0.25
sec	RO	0.25
Telit	RO	1.25
Telit	RO	2.125
EFS2APPS	RO	5.5
Summary		128

RAM Memory

LE (Linux)

- LE910Cx-NF/EU/AP/LA
- 256Mbyte DDR, of which ~60MB will be available for customers' application usage.
- LE910C1-SV/SA/ST
- 128Mbyte DDR, is not available for customers' application usage due to small memory.
- No Telit AppZone Linux support.

TX (ThreadX)

The details about the available RAM for customer's application can be retrieved from the module through the AT command AT#M2MRAM.

The usage is described in the document 80502ST10950A "LE910Cx AT Commands Reference Guide".

Customer Application – Storage and Configuration

Using the Telit SDK, the customer application will be installed directly into the USRFS (/data is the mountpoint).

The customer application can also be linked to the powerup process on predefined hook points:

- /data/oem_earlystart.sh – This is at order 38 of the rcS (S is for Single user scripts are run first)
- /data/oemstart.sh - This is at order 43 of the rc5 (5 is for multi-user scripts)
- /data/oem_poststart.sh - This is at order 99 of the rc5

Since the root-fs of Telit is RO, no applications can be placed there. The above method allows the application to run at powerup. The customer should choose one of the above installation techniques based on when they want the application to run during powerup (early, normal, and later).

The above scripts should link to the application binary for execution (at least one of them). For installation methods as well as build/installation tools refer to the Telit LE910C1 SDK document.

The Configuration files are stored in three main areas:

1. Telit Linux RO FS: The configuration stored there (mainly in /and so on) cannot be changed.
2. Telit RW USER FS (/data): The configuration stored there can be changed by the customer application. Examples are hosts, iproute, WLAN, and so on. Configuration stored there are persistent, that is written to the flash.

3. Telit RW RAM disk (/var/run): This is an FS mounted on the RAM directly, that is. not persistent.

DNS, mobile AP, and firewall configuration are some examples.

Customer applications are installed in the usrfs storage (/data). This is an RW mountpoint.

The customer application will be automatically linked to the powerup process via predefined scripts in the /data:

- /data/oem_earlystart.sh – This is at order 38 of the rcS (S is for Single user scripts are run first)
- /data/oemstart.sh - This is at order 43 of the rc5 (5 is for multi-user scripts)
- /data/oem_poststart.sh - This is at order 99 of the rc5

The above scripts must then be linked to any customer application that needs to automatically run during the powerup process. Note the order of the scripts, rcS scripts runs first on a Linux machine, rc5 script runs after (43 is first then 99).

Note: Section Customer Application – Storage and Configuration is valid only for LE models.

Power Up Time

LE (Linux)

The following measurements were taken using a special perf build.

Non-secured device:

- Entering kernel: 0.712747
- Entering user space: 0.742889
- Entering customer application: 11.046918
- Modem out of reset: 13.774010

TX (ThreadX)

AT On: about 9 seconds.

Power Up Sequence

The following figure explains the LE910Cx powerup sequence.

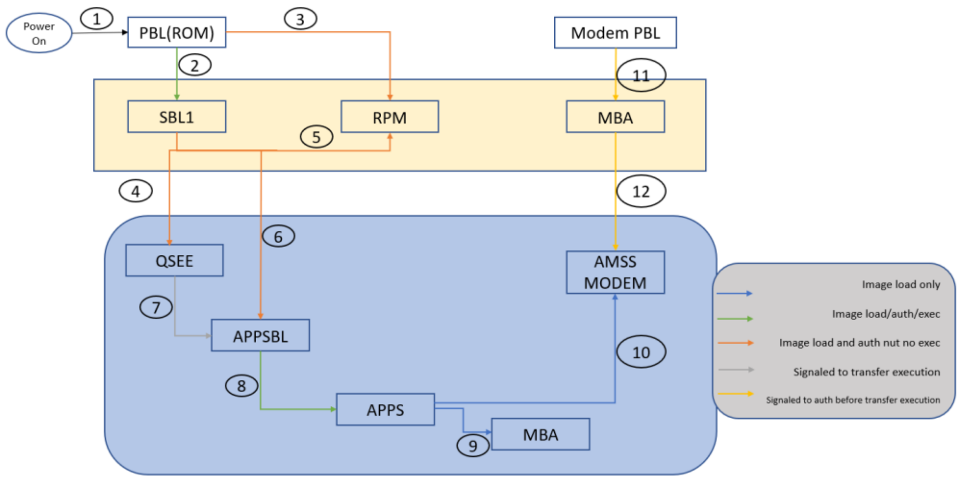


Figure 4: LE910Cx Powerup Sequence

Location Subsystem

The following key features are offered by the Location subsystem:

- Support for GPS, GLONASS, BeiDou/Compass Phase II, Galileo and QZSS
- Supports the following Satellite Based Augmentation Systems (SBAS): WAAS, EGNOS, MSAS (tracked for cross-correlation improvement only)
- Receiver Autonomous Integrity Monitoring (RAIM) & Fault Detection and Exclusion (FDE) support, internal in the receiver.
- Support of assistance data (Ephemerides, location, time…) provided by customer application to ensure faster Time To First Fix (TTFF) through SUPL and LTO injection
- Periodic pulse output for synchronization with the GPS system clock
- NMEA-0183 output on USB

Application Development Environment

Refer to the Telit AppZone Linux documentation in the link, below:

https://s3.amazonaws.com/iot-appzone.telit.com/LE910Cx-TX/25.30.222/index.html

Note: Telit AppZone Linux is not available on LE910C1-SA, LE910C1-ST, LE910C1-SV products, which have low memory,128Mbyte DDR and there is no RAM space for customer application.

Random Number Generator

The LE910Cx RNG is based on FIPS-140-2 PRNG (aka hw_drbg), seeded with QC designed hw entropy unit consisting of the noise source of the ring oscillator (RO).

There are several Linux devices to generate random numbers (under the /dev node):

1. hw_random – This is an HW random number generator, this is the preferred device to get random data from.
2. random – This is, in most cases, an SW random generator.

   The Linux kernel itself (on latest kernel versions >= msm-3.18) adds HW random data to /dev/random if randomness is not sufficient from SW RNG. Random will block if no sufficient randomness is built up.
3. urandom – This device does not have enough randomness and it is not recommended for use unless the quality of the RNG is a concern (this device will probably work faster).

The number of entropies used for the RNG can be checked, and modified, with the following sysfs:

/sys/module/rng_core/parameters/current quality

Max value is 1024.

An example for reading random bytes from the hw_random:

Note: Section Random Number Generator is valid only for LE (Linux) models.

Wake up Events

The Telit Modules provide a function that reduces the power consumption during the period when they are in IDLE state (waiting for a call), allowing for longer activity with a given battery capacity.

The power-saving function can be configured in several modes according to the user's needs.

The Power Saving Modes supported by the Telit Module can be determined via the AT+CFUN=? command's response; for more information, refer to the AT Commands Reference Guide. In power-saving mode, CFUN=5 disables the UART AT interface. The URC is stored in the buffer and is not visible on the UART AT interface. It is flushed to DTE when the modem device exits power saving mode by <DTR=ON>.

AT#PSMRI=<duration time> must be configured as a non-zero value. It enables RI with the specified time if the URC event happened during power-saving mode with CFUN=5.

The power-saving function can be waked up in several events as follows:

Table 7: Wake-up events out of power-saving mode

Wake up Events	CFUN=0	CFUN=4	CFUN=5
	Module enters NON-CYCLIC SLEEP mode	Module performs network deregistration and SIM deactivation. TX and RX are disabled	The power-saving is enabled. DTR is used to exit/enter power saving.
Unsolicited Result Code	The new mode depends on URC	The new mode depends on URC	AT#PSMRI must be configured as a non-zero value to get the URC event via RI. The new mode depends on URC.
Incoming voice call Incoming data call(VoLTE)	Incoming call is managed, and RING is displayed. The module exits the power-saving state and enters CFUN=1 mode.	N/A	Incoming call is managed. RI toggle. DTR is used to exit/ enter a power-saving state in CFUN=5 mode
Incoming SMS AT+CNMI=0,0, ...	Incoming SMS is managed, and URC is not displayed. The module stays in power-saving state in CFUN=0 mode.	N/A	Incoming SMS is managed, and URC is not displayed. DTR is used to exit/enter a power-saving state in CFUN=5 mode.
Incoming SMS AT+CNMI=1,1, ...	Incoming SMS is managed, and URC is displayed. The module exits the power-saving state and enters CFUN=1 mode.	N/A	Incoming SMS is managed. DTR is used to exit/ enter power-saving state in CFUN=5 mode. URC is displayed when the modem device exits the power-saving mode
Incoming GPRS packet	CFUN=1	N/A	CFUN=5

	CFUN=0	CFUN=4	CFUN=5
Wake up Events	Module enters NON-CYCLIC SLEEP mode	Module performs network deregistration and SIM deactivation. TX and RX are disabled	The power-saving is enabled. DTR is used to exit/enter power saving.
RTC alarm	CFUN=1	CFUN=4	CFUN=5
RTS toggling	CFUN=1	N/A	N/A

Note: Since the RTS toggle event periodically checks the pin status in CFUN=0 mode, in the worst case, it may take up to around 1 second for the module to wake up.

LE910C1-EU(4G+2G) does not allow power saving mode because the HSIC configuration of LE910C1-EU(4G+2G) is the master mode. To support power saving mode, HSIC configuration must be disabled by #HSICEN=0 (Manual reboot is required).

For further information, refer to section HSIC Interface.

Wake up Event Examples

CFUN=0: Call, SMS, #QSS, +CALA

Example 1

The wake-up event is an incoming call.

1. Starting control line configuration, UART AT interface is enabled.
   <DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
2. Type in CFUN=0, the module enters NON-CYCLIC SLEEP mode.
   AT+CFUN=0
   OK
3. Below is the new control line configuration. The module is in power saving.
   <DSR=OFF>, RI=OFF, DCD=OFF, <CTS=OFF>, RTS=ON, DTR=ON
   An incoming call arrives.

   RING
4. Below is the new control line configuration. The module is no longer in power saving.
   <DSR=ON>, RI=ON, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
   RING

5. Check the current CFUN.
   AT+CFUN?
   +CFUN: 1 the module is in full functionality mode

   OK

   RING
6. Hang up the call.
   ATH

   OK

   Here is the new control line configuration.
   <DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON

Example 2

The wake-up event is an SMS receiving.

1. Enable URC created by the SMS receiving.
   AT+CNMI=1,1,0,0,0
   OK
   Starting control line configuration, the UART AT interface is enabled.
   <DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
   Type in CFUN=0, the module enters NON-CYCLIC SLEEP mode.
   AT+CFUN=0

   OK
2. Here is the new control line configuration. The module is in power saving.
   <DSR=OFF>, RI=OFF, DCD=OFF, <CTS=OFF>, RTS=ON, DTR=ON
   A SMS is arrived.
   +CMTI: "SM",17
3. Here is the new control line configuration. The module is no longer in power saving.
   <DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
4. Check the current CFUN.
   AT+CFUN?
   +CFUN: 1 the module is in full functionality mode

   OK

Example 3

The wake-up event is the #QSS URC.

1. Starting control line configuration, UART AT interface is enabled.
   <DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
2. Enable Query SIM Status URC.
   AT#QSS=1
   OK
3. Type in CFUN=0, the module enters NON-CYCLIC SLEEP mode.
   AT+CFUN=0
   OK
4. Here is the new control line configuration. The module is in power saving.
   <DSR=OFF>, RI=OFF, DCD=OFF, <CTS=OFF>, RTS=ON, DTR=ON
5. Extract the SIM. After a while, the DTE displays the following URC:
   #QSS:0

   Here is the new control line configuration. The module is no longer in power saving.
   <DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
6. Check the current CFUN mode.
   AT+CFUN?
   +CFUN: 1 the module is in full functionality mode

   OK

Example 4

+CALA URC event forces the module in CFUN=1 mode.

1. Starting control line configuration, UART AT interface is enabled.
   <DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON

2. Set the clock
   AT+CCLK="08/05/16,09:20:30+00"

   OK

3. Set when the alarm wakes up: in two minutes (it is just an example).
   AT+CALA="08/05/16,09:22:30+00",0,2,"ALARM, ALARM, ALARM"
   OK
4. Type in CFUN=0, the module enters NON-CYCLIC SLEEP mode.
   AT+CFUN=0
   OK
   Here is the new control line configuration. The module is in power saving.
   <DSR=OFF>, RI=OFF, DCD=OFF, <CTS=OFF>, RTS=ON, DTR=ON
5. During the ALARM waiting, the module is in power saving and UART AT interface is disabled.
   When the alarm wakes up, the DTE displays the URCs. The module exits power

saving in
CFUN=0 mode and enters CFUN=1 mode.
+CALA: ALARM, ALARM, ALARM

6. Here is the new control line configuration.
<DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
+CALA: ALARM, ALARM, ALARM
Check the alarm mode
AT#WAKE?
#WAKE: 1   the module is in alarm mode

   OK

   +CALA: ALARM, ALARM, ALARM
7. Check the current CFUN.
AT+CFUN?
+CFUN: 1 the module is in full functionality mode
OK
+CALA: ALARM, ALARM, ALARM

   +CALA: ALARM, ALARM, ALARM
8. After 90 sec, the module exits alarm mode.
9. Check the alarm mode.
AT#WAKE?
#WAKE: 0   the module exited alarm mode

   OK

CFUN=4: #QSS, +CALA

Example 1

#QSS URC event leaves the module in CFUN=4 mode.

1. Starting control line configuration, UART AT interface is enabled.
<DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
2. Type in CFUN=4, the module performs network deregistration, and SIM deactivation.
AT+CFUN=4
OK
3. Control line configuration is not changed.
<DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
Enable Query SIM Status URC.
AT#QSS=1
OK

4. Extract the SIM. The URC does not arrive because CFUN=4 mode deactivates the SIM.
   The module stays in CFUN=4 mode.
   AT+CFUN?
   +CFUN: 4
   OK

Example 2

+CALA URC event leaves the module in CFUN=4 mode.

1. Starting control line configuration, UART AT interface is enabled.
   <DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
2. Set the clock
   AT+CCLK="08/05/16,09:20:30+00"
   OK
3. Set when the alarm wakes up: in two minutes (it is just an example).
   AT+CALA="08/05/16,09:22:30+00",0,2,"ALARM, ALARM, ALARM"
   OK
4. Type in CFUN=4, the module performs network deregistration and SIM deactivation.
   AT+CFUN=4
   OK
5. Control line configuration is not changed
   <DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
6. When the alarm wakes up, the DTE displays the URCs.
   +CALA: ALARM, ALARM, ALARM
7. Control line configuration is not changed.
   <DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON

   +CALA: ALARM, ALARM, ALARM

8. Check the alarm mode
   AT#WAKE?
   #WAKE: 1 the module is in alarm mode

   OK

   +CALA: ALARM, ALARM, ALARM
   The module does not change CFUN mode.
   AT+CFUN?
   +CFUN: 4
   OK

+CALA: ALARM, ALARM, ALARM
After 90 sec, the module exits alarm mode.

9. Check the alarm mode.
AT#WAKE?
#WAKE: 0 the module exited alarm mode

OK

The module does not change CFUN mode.
AT+CFUN?
+CFUN: 4
OK

CFUN=5: Call, SMS, +CALA

Example 1

The wake-up event is an incoming call.

1. Starting control line configuration, UART AT interface is enabled.
<DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
2. Force the module in CFUN=5 mode, the power saving is enabled.
AT+CFUN=5
OK
3. Control line configuration does not change, UART AT interface is still enabled.
<DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
4. Force the module in power saving.
DTR OFF
5. The module is in power saving, and the UART AT interface is disabled.
<DSR=OFF>, RI=OFF, DCD=OFF, <CTS=OFF>, RTS=ON, DTR=OFF
An incoming call is arrived
Here is the new control line configuration: RI=ON
<DSR=OFF>, RI=ON, DCD=OFF, <CTS=OFF>, RTS=ON, DTR=OFF
6. Exit power-saving and UART AT interface is enabled
DTR ON
<DSR=ON>, RI=ON, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
The module exits power saving but stays in CFUN=5 mode.

AT+CFUN?
+CFUN:5
OK

RING

7. Hang up the call.
   ATH
   OK
8. Enter power saving.
   DTR OFF

The module enters again the power saving mode, and the UART AT interface is disabled.
<DSR=OFF>, RI=OFF, DCD=OFF, <CTS=OFF>, RTS=ON, DTR=OFF

Example 2

The wake-up event is an SMS receiving.

1. Enable URC created by the SMS receiving.
   AT+CNMI=1,1,0,0,0
   OK
2. Set AT#PSMRI to get the URC event during the power-saving mode
   AT#PSMRI=1000

   OK
3. Starting control line configuration, the UART AT interface is enabled.
   <DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
4. Force the module in CFUN=5 mode, the power saving is enabled.
   AT+CFUN=5
   OK
5. Control line configuration does not change, UART AT interface is still enabled.
   <DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
6. Force the module in power saving.
   DTR OFF

7. The module is in power saving and UART AT interface is disabled.
   <DSR=OFF>, RI=OFF, DCD=OFF, <CTS=OFF>, RTS=ON, DTR=OFF
   An SMS arrives and it is stored in the buffer on UART AT interface and RI is ON for 1 sec
8. <DSR=OFF>, RI=ON, DCD=OFF, <CTS=OFF>, RTS=ON, DTR=OFF
   The module is still in power saving and UART AT interface is disabled.
   <DSR=OFF>, RI=OFF, DCD=OFF, <CTS=OFF>, RTS=ON, DTR=OFF
9. Exit power saving, and AT interface enabled. The buffered URC displayed
   DTR ON
   +CMTI: "SM",17

   <DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON

10. The module exits power saving but stays in CFUN=5 mode.
    AT+CFUN?
    +CFUN:5
    OK
11. Enter power saving.
    DTR OFF
12. The module enters again the power saving mode, and the UART AT interface is disabled.
    <DSR=OFF>, RI=OFF, DCD=OFF, <CTS=OFF>, RTS=ON, DTR=OFF

Example 3

The wake-up event is an SMS receiving

1. Check the number of SMS already arrived.
   AT+CPMS?
   +CPMS: "SM",18,30,"SM",18,30,"SM",18,30 Yes, a new SMS is arrived.
   OK
2. Disable URC created by the SMS receiving.
   AT+CNMI=0,0,0,0,0
   OK
3. Starting control line configuration, the UART AT interface is enabled.
   <DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
   Force the module in CFUN=5 mode, the power saving is enabled.
   AT+CFUN=5
   OK
4. Control line configuration does not change, UART AT interface is still enabled.
   <DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
5. Force the module in power saving.
   DTR OFF
6. The module is in power saving and UART AT interface is disabled.
   <DSR=OFF>, RI=OFF, DCD=OFF, <CTS=OFF>, RTS=ON, DTR=OFF
   An SMS is sent and arrived. The DTE does not display the URC +CMTI.
7. The module is still in power saving and UART AT interface is disabled.
   <DSR=OFF>, RI=OFF, DCD=OFF, <CTS=OFF>, RTS=ON, DTR=OFF
   Exit power-saving and enable again the hardware flow control of the serial line.
   DTR ON
   <DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
8. The module exits power saving but stays in CFUN=5 mode.
   AT+CFUN?

+CFUN:5
OK

9. Check if a new SMS arrives.
AT+CPMS?
+CPMS: "SM",19,30,"SM",19,30,"SM",19,30 Yes, a new SMS is arrived.
OK
10. Enter power saving.
DTR OFF
11. The module enters again the power saving mode, and the UART AT interface is disabled.
<DSR=OFF>, RI=OFF, DCD=OFF, <CTS=OFF>, RTS=ON, DTR=OFF

Example 4

+CALA URC event leaves the module in CFUN=5 mode.

1. Starting control line configuration, AT interface is enabled.
<DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
2. Force the module in CFUN=5 mode, the power saving is enabled.
AT+CFUN=5
OK
3. Control line configuration does not change, UART AT interface is still enabled.
<DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
4. Set the clock
AT+CCLK="08/05/16,09:20:30+00"
OK
5. Set when the alarm wakes up: in two minutes (it is just an example).
AT+CALA="08/05/16,09:21:30+00",0,2,"ALARM, ALARM, ALARM"
OK
6. Set AT#PSMRI to get the URC event during the power-saving mode
AT#PSMRI=1000

OK

7. Force the module in power saving.
DTR OFF
8. During the ALARM waiting, the module is in power saving, and the UART AT interface is disabled.
<DSR=OFF>, RI=OFF, DCD=OFF, <CTS=OFF>, RTS=ON, DTR=OFF
9. When the alarm time is expired, the URC event takes place. It is stored in the buffer on UART AT interface. and RI is ON during 1 sec

<DSR=OFF>, RI=ON, DCD=OFF, <CTS=OFF>, RTS=ON, DTR=OFF
UART AT interface is disabled.

10. Now, enable UART AT interface. The module exits power saving, and the buffered URC is flushed to DTE. but stays in CFUN=5 mode.
DTR ON
<DSR=ON>, RI=OFF, DCD=OFF, <CTS=ON>, RTS=ON, DTR=ON
+CALA: ALARM, ALARM, ALARM
+CALA: ALARM, ALARM, ALARM
11. Check if the module is in alarm mode.
AT#WAKE?
#WAKE: 1
OK
+CALA: ALARM, ALARM, ALARM
12. Check the current CFUN.AT+CFUN?
+CFUN: 5
OK
+CALA: ALARM, ALARM, ALARM
13. After 90 sec the module exits alarm mode.
AT#WAKE?
#WAKE: 0
OK
14. Check the current CFUN.
AT+CFUN?
+CFUN: 5
OK
15. Force the module in power saving.
DTR OFF
16. The module is in power saving in CFUN=5 mode, and the UART AT interface is disabled.
<DSR=OFF>, RI=OFF, DCD=OFF, <CTS=OFF>, RTS=ON, DTR=OFF

SPI

The LE910Cx includes a Serial Peripheral Interface (SPI) bus, which supports:

- Master mode at max 50 MHz
- Programmable data bits (4 to 32 bits)
- Programmable spacing between byte/word transfers (SPI_CS_N WAIT)
- Programmable transfer length (number of bytes transferred within each SS/CS assertion)

LE910Cx has one SPI core using BLSP (board low speed peripheral).

The BLSP includes a UART and QUP (Qualcomm universal peripheral) cores.

This QUP has an SPI/I2C mini cores.

These mini cores implement the following logic:

- A common output FIFO provides system output data to one or more mini cores
- A common input FIFO provides system input data from one or more mini cores

These cores use BAM (Bus access module) to move data to/from the peripheral buffers. Each peripheral is equipped with a pair of BAM pipes.

BAM block periodically reads the input FIFO until the programmed number of words are received.

BAM block periodically populates the output FIFO until the programmed number of words are written to the output FIFO.

Once the transmission is complete, an interrupt is generated.

Note: Since two H/W Pins of SPI are shared with Aux UART, SPI and Aux UART cannot be used simultaneously. To use SPI, use #SPIEN and #SPICFG commands.

ThreadX products do not support #SPICFG.

LE (Linux)

TGPIO 1-10 (customer allocated GPIO) can be configured as an interrupt source of an SPI master device. This allows an SPI slave device to notify the SPI master device of data being transferred.

TX (ThreadX)

SPI functionality is only provided to the customer through the AppZone framework. For more information refer to section Location Subsystem.

GPIO

The LE910Cx has ten GPIOs and eight UART pins that may be configured as inputs and outputs using the Linux device driver and AT command.

These GPIO pins can be used to control external hardware directly, requiring little or no additional hardware.

Table 8: Supported GPIO pins for LE910Cx

Pin Number	Description
1	GPIO1
2	GPIO2
3	GPIO3
4	GPIO4
5	GPIO5
6	GPIO6
7	GPIO7
8	GPIO8
9	GPIO9
10	GPIO10
UART Pins	
20	DCD
21	CTS
22	RI
23	DSR
24	DTR
25	RTS
26	RXD
27	TXD

To use UART pins as GPIO, use the #V24CFG instruction to pre-set them to GPIO. For more information, refer to section 8.1. How to Use the #V24CFG Command.

To use 10 GPIOs, use the #GPIO command or the GPIO interface.

For more information refer to “AT Commands Reference Guide” and “Linux device driver Application Note”)

Warning: During the module power-on procedure, certain GPIOs (GPIO-1, GPIO-5~9) must not be lifted externally (by the carrier board). Pulling these pads up during module startup could result in an undesirable/non-operational boot mode. (For more information, refer to Hardware User Guide).

Note: ThreadX products do not support GPIO device drivers.

Serial Interfaces

LE (Linux)

Apps to External MCU

The LE910Cx includes serial interfaces (cdc-acm and raw data interfaces) on both the UART and USB physical interfaces. These interfaces are accessible from the Linux side and can be used to communicate with an external MCU.

The UART device nodes are:

- /dev/ttyHS0 – This is the high-speed UART (the modem port). The modem uses this port as an AT command port. An application that wants to use this port for communication with an external MCU should first stop the modem service running on top of this port using the following command ("/sbin/ds_uart_script stop") or #PORTCFG should be set to 8.

  Note: When #M2MATP=1, #PORTCFG should be set to 8. For more information, refer to # PORTCFG on the LE910Cx AT Command Reference Guide.

- /dev/ttyHSL0 – This is the debug console port. The modem uses this port as an AT command port. To use this port for communication with an external MCU, set #PORTCFG to 0 or use the following command ("/sbin/ds_uart_script stop") to terminate the modem service that runs on top of this port.

The USB device nodes are:

- /dev/ttyGSX (X is a number) – Depending on the selected USB composition, there might be one or several ttyGS ports. For example, USB composition 1201 includes a single ttyGS0 port (in 1201 this port is meant for NMEA sentences sent by the modem). This is a gadget serial interface and can be used as a standard tty port to communicate with an external MCU if NMEA sentences are not enabled.

ttyGS ports represent:

- NMEA ports - The modem uses these ports to dump NMEA sentences. This port is disabled by default and is enabled when the user issues the following AT commands:
  - AT$GPSP=1
  - AT$GPSNMUN=1,1,1,1,1,1,1

If the NMEA ports are not enabled, these ports can be used by the application to communicate with an external MCU.

Apps to Modem

Modems and apps communicate via shared memory. The devices used for this communication are smd devices and are present in the /dev/node. It is not recommended to work with these ports directly, Telit uses these ports for several use cases and a change might break these functionalities.

Below is the list of smd channels available on the apps:

```
/dev#ls-l/dev/smd*

Crw-rw ---- 1 root   root   248,   11 Jan 1  1970 / dev / smd11

Crw-rw ---- 1 root   root   248,    2 Jan  6  01:58 / dev / smd2

Crw-rw ---- 1 root   root   248,    21 Jan 1 1970 / dev / smd21

Crw-rw ---- 1 root   root   247,    1 Jan   1 1970 / dev / smd22

Crw-rw ---- 1 root   root   248,    36 Jan 1 1970 / dev / smd36

Crw-rw ---- 1 root   root   248,    7 Jan   1 1970 / dev / smd7

Crw-rw ---- 1 root   root   248,    8 Jan   6  01:58 / dev / smd8

Crw-rw ---- 1 root   root   248,    9 Jan   1  1970 / dev / smd9
```

/dev/smd8 is used by the ttyHS0 for AT command over the UART (USIF0 port.)

/dev/smd9 is used by the MCM_ATCOP for AT commands sent via the mcm interfaces.

/dev/smd2 is used by the ttyHSL0 for AT command over the AUX UART (USIF1 port). If #PORTCFG is 8 or 0, User can send an AT command via this smd channel and capture a response as following:

```
/dev # cat /dev/smd2&

/dev # echo-e “at/r/n”>/dev/smd2

/dev # at

OK


/dev # echo-e “at+cpin?\r\n”>/dev/smd2

/dev # at+cpin?

+CPIN: READY

OK
```

To use smd2, run the command "/sbin/ds uart script stop" when #PORTCFG is 14 and #M2MATP is 1.

Telit/QC uses other smd channels internally, and users cannot access them.

Note: If UART (USIF0 port) is assigned as an AT port by #PORTCFG, the /dev/smd8 is used internally. In case of #M2MATP=1, #PORTCFG should be set to 8 to use "/dev/smd8" for customer use.

If AUX UART (USIF1 port) is assigned as an AT port by #PORTCFG, the "/dev/smd2" is used internally. #PORTCFG should be set to 8 or 0 to use "/dev/smd2" for customer use. Or the modem service running on top of this port should be stopped with the following command ("/sbin/ds_uart_script stop") on every boot-up.

For more information about #PORTCFG,.refer to the AT command guide..

Audio

LE (Linux)

Audio subsystem

- Optional embedded analog codec with two microphone inputs
- Optional embedded analog codec with one stereo or two mono outputs
- PCM digital audio IO
  - Clock Type: Master
  - Short frame sync
  - Data format 16 –bit linear
  - <clock>
    - 128 – DVI Clock activated at 128KHz
    - 256 – DVI Clock activated at 256KHz
    - 512 – DVI Clock activated at 512KHz
    - 1024 – DVI Clock activated at 1024KHz
    - 2048 – DVI Clock activated at 2048KHz
    - 4096 – DVI Clock activated at 4096KHz
  - <sample rate>
    - 8KHz
    - 16KHz

The purpose of the Audio user space Application is:

1. Configure and control the audio properties such as Volume, clock, mute device, and so on.
2. Use the UCM file to configure audio paths and audio devices (handset, hands-free)
3. Use AMIX to send commands and data to the ALSA.
4. Manage all voice call events for audio.
5. Manage the digital (external codec) and analog (internal codec) use cases.

The MAXIM9867 and MDM9628 communicate through a dedicated Aux PCM interface.

Audio Application
UCM
ALSA Kernel audio subsystem

ASoC
HW

UCM is the user-space library that provides C API for clients.

ALSA is the kernel audio subsystem that abstracts the PCM and controls.

ASoC is the low-level driver subsystem.

The ACDB file contains audio calibration data and is categorized as follows:

- Handset_cal.acdb – Contains calibration data for handset devices
- Speaker_cal.acdb – Contains calibration data for speakerphone (handheld, handsfree) devices
- Headset_cal.acdb – Contains calibration data for headset devices
- Bluetooth_cal.acdb – Contains BT devices for voice and audio (not supported)
- Global_cal.acdb – Contains voice and audio stream calibration that is independent of device and other global calibration
- Hdmi_cal.acdb – Contains HDMI audio playback device (not supported)
- General_cal.acdb – Contains general calibration data not falling in any of the other categories.

The LE910Cx has several locations to search for ACDB files, which allows customers to override Telit's defaults settings and work with QC tools. These ACDB sets of files are searched in the order below:

1. /data/acdb/acdbdata: This is usually where the QC tool enters calibration data when used.
2. Default code settings: This is Telit's default setting. We will fall back into this if none of the above exists.

This allows the customer to calibrate the audio via the QC tool and apply it. If the calibration data is accepted, the customer puts the calibration files into its rom volume and the device will automatically pick this up in the next power-up.

Command-line examples for using the ALSA soundcard driver:

- Record an active voice call (analog audio):
  - amix 'MultiMedia1 Mixer AUX_PCM_MAX9867_UL_TX' 1
  - amix 'Input Mixer' 1
  - arec -D hw:0,0 -R 8000 -C 1 file.wav
- Record an active voice call (digital audio):
  - amix 'MultiMedia1 Mixer AUX_PCM_UL_TX' 1
  - arec -D hw:0,0 -R 8000 -C 1 file4.wav
- Play an audio file:
  - amix 'AUX_PCM_MAX9867_RX Audio Mixer MultiMedia1' 1
  - aplay filename.wav

TX (ThreadX)

Audio subsystem

- Optional embedded analog codec with two microphone inputs
- Optional embedded analog codec with one stereo or two mono outputs
- PCM digital audio IO
  - Clock Type: Master
  - Short frame sync
  - Data format 16 –bit linear
  - <clock>
    - 2048 –   DVI Clock activated at 2048KHz
    - 4096 –   DVI Clock activated at 4096KHz
  - <samplerate>
    - 8KHz
    - 16KHz
    - Clock 2048KHz supports only Sample Rate 8KHz
    - Clock 4096KHz supports only Sample Rate 16KHz

Note: ThreadX products do not support the following features and functionality.

UCM, ALSA (amix, arec, aplay),

Audio playback, Audio playback during a voice call, Voice recording, In call music delivery, TTY, USB Audio.

UART

The LE910Cx has two UART interfaces, Main and Aux.

The main UART interface is a 4-wires UART with max data rates up to 3.75 Mbps and acts as a DUN channel as default.

Aux UART interface is a 2-wires UART acting as a DUN channel as default.

Following baud rates are supported:

300,600,1200,2400,4800,9600,19200,38400,57600,115200,230400,460800,921600,2000000,2500000,3000000,3500000,3750000

LE (Linux)

If the Customer does not need UART modem ports, then can access the TTY device created under the dev file system. This device can be accessed from user space to read/write data to a peripheral connected to the LE910Cx.

The TTY device can be configured with the STTY utility (included).

To tear down the UART to modem connection, use this command:

/sbin/ds_uart_script stop

After this, the ttyHS0 and ttyHSL0 can be directly accessed (/dev/ttyHS0) from the Linux user space for read/write to an external processor connected on the other side.

Note: : If the customer does not need UART modem ports, #PORTCFG should be set to 8 when #M2MATP is 1.

For more information, LE910Cx AT Command Reference Guide.

USB Interface

The LE910Cx includes a USB2.0 compliant Universal Serial Bus (USB) Transceiver, which operates at USB high-speed (480Mbits/sec). By default, the module is configured as a USB peripheral.

LE (Linux)

The USB port is typically used for:

- Flashing of firmware and module configuration
- Production testing
- Accessing the Application Processor's filesystem (debug bridge)
- AT command access (2 modem ports)
- High-speed WWAN access to an external host
- Diagnostic monitoring and debugging
- NMEA data to an external host CPU

The following standardized device classes can be supported:

- Serial (reduced ACM), CDC-ECM, MBIM, RMNET (Qualcomm proprietary)
- Audio Device class, but limited to voice Rx & Tx and only when in peripheral mode

Note: When the module is attached to USB serial driver on the Linux platform, the DIAG port, MODEM ports, and NMEA port will be created as "/dev/ttyUSBx" not "/dev/ttyACMx". For more information, refer to "Telit_Modules_USB_Drivers_User_Guide".

LE910Cx uses a fixed USB serial number, not the unique USB serial number of each device on normal boot. (Ex 012345789ABCDEF ....)

The following USB compositions are available:

1201 - DIAG + ADB + RMNET + NMEA + MODEM + MODEM + SAP

1203 - RNDIS + DIAG + ADB + NMEA + MODEM + MODEM + SAP

1204 - DIAG + ADB + USB_MBIM + NMEA + MODEM + MODEM + SAP

1205 - USB_MBIM

1206 - DIAG + ADB + ECM + NMEA + MODEM + MODEM + SAP

1250 - RMNET + NMEA + MODEM + MODEM + SAP

1251 - RNDIS + NMEA + MODEM + MODEM + SAP

1252 - USB_MBIM + NMEA + MODEM + MODEM + SAP

1253 - ECM + NMEA + MODEM + MODEM + SAP

1254 - MODEM + MODEM

1255 - NMEA + MODEM + MODEM + SAP

1230 - DIAG + ADB + RMNET + AUDIO + NMEA + MODEM + MODEM + SAP

1231 - RNDIS + DIAG + ADB + AUDIO + NMEA + MODEM + MODEM + SAP

1260 - DIAG + ADB + RMNET + NMEA + MODEM + MODEM + SAP

1261 - DIAG + ADB + RMNET + NMEA + MODEM + MODEM + SAP

1262 - DIAG + ADB + RMNET + MODEM + MODEM + AUX

The USB composition can be changed using the usb_composition utility on the Linux side as well as using the #USBCFG AT command (covered by the LE910Cx AT user guide). The usb_composition utility can be used to select any of the above composition lists.

The 125X USB compositions are specifically made for customers that are willing to allow the LE910Cx device to sleep even if the USB is connected (Enables the USB selective to suspend). The USB selective suspend also depends on the host implementation.

Changing USB Composition to MBIM only USB Composition

Note: Only on Linux operating system, the following commands can be used.

To change USB composition to MBIM only USB composition, perform the following steps:

1. Send the MBIM_OPEN message to the modem with the following command and wait 3 sec.

   echo 01 00 00 00 10 00 00 00 01 00 00 00 00 10 00 00 | xxd -r -p >/dev/cdc-wdm0

2. Send the USB composition with the following command:

   echo 03 00 00 00 32 00 00 00 05 00 00 00 01 00 00 00 00 00 00 00 54 65 6c 69 74 4d 42 49 4d 45 78 74 65 6e 64 53 01 00 00 00 01 00 00 00 02 00 00 00 XX 00 | xxd -r -p >/dev/cdc-wdm0

   The red number should be changed to set USB configuration and the range of the red number is 0x00-0x0c.

TX (ThreadX)

The USB port is typically used for:

1. Flashing of firmware and module configuration
2. Production testing
3. AT command access (2 modem ports)
4. Hig- speed WWAN access to an external host
5. Diagnostic monitoring and debugging

   The following standardized device classes can be supported:

6. Serial (reduced ACM), CDC-ECM, RMNET (Qualcomm proprietary)

Note: When the module is attached to USB serial driver on the Linux platform, the DIAG port and MODEM ports will be created as "/dev/ttyUSBx" not "/dev/ttyACMx".

For more information, refer to "Telit_Modules_USB_Drivers_User_Guide".

The following USB compositions are available:

1031 - DIAG + MODEM + MODEM + RMNET

1033 - DIAG + MODEM + MODEM + ECM

1034 - MODEM + MODEM + RMNET

1035 - MODEM + MODEM + ECM

1036 - MODEM + MODEM

The USB composition can be changed by the #USBCFG AT command (covered by the LE910Cx AT user guide).

HSIC Interface

The LE910Cx includes a 2-wire HSIC (High-Speed Inter-Chip) interface, and the interface supports HSIC master or slave mode.

The HSIC interface of LE910Cx supports the following feature:

1. No hot plug detection
2. No hot removal/attach, interface is always connected
3. No high-speed chirp protocols
4. HSIC master/ slave mode support

The HSIC configuration can be changed using the #HSICEN command and a manual reboot is required.

Example 1

Enable HSIC master mode.

AT#HSICEN=1

OK

Example 2

Enable HSIC slave mode.

AT#HSICEN=2

OK

Example 3

Disable HSIC configuration.

AT#HSICEN=0

OK

To enable power-saving mode when the HSIC configuration of LE910Cx is set to master mode(master and slave connected state) follow the steps as shown below:

- Before setting CFUN to 5, please do the following command on apps
- echo auto > /sys/bus/usb/devices/1-1/power/control

Note: If the HSIC configuration of LE910Cx is in slave mode, the LE910Cx does not allow the power saving mode because the signals between HSIC master and slave resume are not synced.

If the HSIC configuration of LE910Cx is the master mode, the power saving mode can only be entered in the HSIC master and slave connection state.

ThreadX products do not support the HSIC interface.

Ethernet Interface

The LE910Cx has Embedded Ethernet MAC and supports only the SGMII interface.

The embedded Ethernet MAC of LE910Cx supports the following features.

- IEEE 802.3 Ethernet 10/100/1000Mbps, SGMII IF
- SGMII interface can be used using external PHY. (SGMII to external PHY)
  - Giga Ethernet PHY can be used by a transceiver chip (example: Marvell 88EA1512 PHY chip)

Note: ThreadX products do not support the SGMII interface.

SD/MMC Interface

LE910Cx provides an SD port supporting the SD3.0 specification, which can be used to support standard SD/MMC memory cards.

For more information refer to the LE910Cx Linux device driver Application Note.

Note: ThreadX products do not support the SDIO interface.

RTC

The LE910Cx includes an RTC (Real Time Clock), which allows an application to set an alarm to either wake up the module from sleep mode, power up the module or simply interrupt it once the alarm expires. The interface to the RTC, from the SW perspective, takes place via the RTC device exposed in the device file system (standard RTC device interface).

The RTC driver handles the RTC time and alarm requests from the user.

The RTC sysfs interface can be seen at: /sys/class/rtc/rtc0/

Users can set the RTC for wake alarm (using RTC_WKALM_SET ioctl) and receive notification via the RTC device interface (/dev/rtc0). Another way is to set the RTC wake alarm, shutdown the module (shutdown –h now) and the device will automatically power up when the alarm expires.

To allow the user space to know why the device has powered up, the following file has been added:/var/volatile/tmp/rtc_wake_status.

If the file contains 0 as a value, it means a normal user-selected power up. 1 means power-up due to RTC alarm expiration.

Note: ThreadX products do not support the RTC interface.

Time Services

The time daemon manages the system time between the modem and the app processor.

The time daemon service starts during boot and reads the RTC time from /dev/rtc0 and the offset from file /data/time/ats_1 (or /var/tmp/time/ in older versions). This value is then set as the system time.

When the system boots and the time daemon initialize for the first time, the daemon reads the file at /data/time/ats_1 to initialize the offset value. Therefore, as long as the files are present and contain the correct values, the system time is preserved after the system reboot.

Coordinated Universal Time (UTC)

The UTC standard is used to adjust the world clock and time. UTC is based on International Atomic Time (TAI), and time zones around the world are expressed as positive or negative offsets from UTC.

UTC is the time standard used for many Internet and web standards. The Network Time Protocol, designed to synchronize the clocks of computers over the Internet, encodes the times using the UTC system.

UTC is helpful to avoid confusion about time zones and daylight saving time.

Time Zones

The time zones around the world are expressed as positive or negative offsets from UTC.

Daylight Saving Time (DST)

DST is observed in some time zones by advancing the time by some duration in summer and reverting to the original time during winter. DST helps in getting more daylight in the evening and less in the morning.

Time Update

Mobile devices have an internal timer system and the time and time zone can be updated manually. Due to inaccuracies of the device-specific internal timer system, each device deviates with time over a period. There are various network standards to synchronize the device time with the network-provided wall clock time periodically:

Network Identity and Time Zone (NITZ)

NITZ is a mechanism for provisioning the local time, date, and network provider identity information to mobile devices via a wireless network. NITZ is currently an optional part of the official GSM standard (phase 2+release 96).

The NITZ standard allows the network to “transfer its current identity, universal time, DST, and LTZ, but each is optional, and support varies across radio access network vendor and operators.”

With NITZ, the accuracy of the time information is in the order of minutes.

GPS – Clock synchronization

The GPS receiver can also use time information received from GPS time signals. The time accuracy of the GPS network is < 1 ms.

System Time Base

Real-time time base

The Real-time is a wall clock time used by the UI to display the current system time. Generally, this time is updated and synchronized with time information from the network, for example, as the modem uses standard network time protocols, such as NITZ. During synchronization, this time can go forward or backward about the network time.

Monotonic time base

The monotonic time is the time elapsed since the epoch time (01 Jan 1980) or some random start time. Since this time is not used for displaying UI or alarm setup, the start time is not relevant.

The monotonic time always moves forward in time but does not reflect the time on the wall clock in any specific way. In the current implementation, CLOCK_MONOTONIC resembles the number of tick jiffies that starts at 0 when the system boots and monotonically increases from there.

The monotonic time is used by the kernel and the user for all relative time events because this time never goes backward.

Data Connection

QCMAP_CLI

QCMAP_CLI is a QC user space application that allows to activate a PDP context along with network interface setup, firewall configuration, DNS, and so on.

This is a GUI application that resides in the /usr/bin directory.

Note: ThreadX does not support QCMAP_CLI.

WatchDog

A watchdog is a fixed-length counter that enables a system to recover from an unexpected hardware or software catastrophe.

Unless the system periodically resets the watchdog timer, the watchdog timer assumes a catastrophe and tries to handle the situation.

In general, there are two types of watchdog implementations, a hardware watchdog and a software watchdog based on timer interrupt.

Both software and hardware watchdogs are used in the system.

The software watchdog is not implemented in all subsystems; for example, RPM, TrustZone, and APSS do not have software watchdogs, while CNSS and MPSS implement software watchdog.

The hardware watchdog module is a piece of hardware used to ensure that the processor is not stuck or overloaded and consists of a timer that counts down from a predetermined value. If the timer is not reset (also known as "dog_kick" or "petting the dog" or "servicing the dog") by the corresponding CPU core, it eventually counts to 0 and triggers a watchdog timeout.

It is the responsibility of each CPU core to ensure it keeps resetting the counter. If it is unable to do so (if the dedicated task is starved or the CPU core is locked up, etc.), it is assumed that the system has gone into a bad state.

In addition to the reset triggering signal (WatchDog_expired), it is also possible to generate a watchdog interrupt before the watchdog expiration, to allow a processor to attempt the system recovery before resetting it:

1. BARK (FIQ) – Interrupt before watchdog expiration to allow the processor to attempt the system recovery before resetting it
2. BITE (Reset) – When watchdog timeout happens

The watchdog timer continues counting even after BARK occurs.

Hardware Watchdog Characteristics

In the LE910Cx platform, the system has one and only one hardware watchdog counter for every processor or processor cluster located in the corresponding processor subsystem.

1. The only exception is that the app processor has a non-secure and secure watchdog.
2. Each processor or processor cluster is responsible for periodically resetting (kicking/ petting/servicing) its watchdog.
3. The system has one and only one secure watchdog whose primary event (Bite) causes a system reset or cold boot.
4. Except the secure watchdog, the primary event, or bite, shall be sent to the app processor as an interrupt.
5. The secondary event, or bark, must and will be serviced by the corresponding processor.
6. The secure watchdog is always under the control of the Secure Root of Trust (SRoT).
7. The system provides a secure watchdog that enables the pin to be used to connect the watchdog bites to the global reset circuitry for debug purposes.

8. A watchdog enable pin is provided behind a GPIO.
9. The GPIO pin is detected by the fuse detect operation, and the value is stored in a register in the security control block.
10. A watchdog enable fuse is also provided and will be ORed with the GPIO. So, if the watchdog enable fuse is blown, the secure watchdog is forced to be enabled and the GPIO is essentially ignored.
11. The watchdog timer never causes a processor from exiting a sleep or power-collapse state except for the secure watchdog.
12. With the exception of programming the watchdog duration, the watchdog hardware is designed in such a way that the corresponding software should only need to pet its watchdog. All watchdog enables and interruptions due to power collapse, processor sleep, and debug conditions are handled automatically by the hardware.
13. All hardware watchdogs run on the 32 kHz sleep XTAL clock.
14. The size of the watchdog timer counter is 20 bits wide. With the sleep clock running at 32 kHz, a maximum duration of 32 sec is achieved.
15. When a processor is in the Reset state, the associated watchdog is held in the Disabled state. When a processor is hung, the subsequent restart process resets the processor subsystem, not just the CPU; then the watchdog timer including the configuration registers is also reset.
16. Upon cold boot or subsystem restart, the watchdog is held in the Disabled state until the software enables the watchdog timer.

Watchdog Connectivity

All processor subsystems have the watchdog bite sent to the app processor. The barking event is routed as a local interruption.

Note that the non-secure watchdog bark in the app processor is sent to HLOS, and the non-secure bite is sent to TrustZone.

A7 Apps Processor Watchdogs

The A7 app processor subsystem has two watchdogs (secure and non-secure), each handled by a different software block.

Non-secure A7 apps processor watchdog

If the A7 apps processor hardware watchdog goes without being petted for 4 sec, a bark IRQ is asserted. This is known as a nonsecure dog bark.

The kernel handles this as a software error. If the watchdog continues for 5 sec without being petted or if the bark IRQ has not been handled properly, FIQ is reported to TrustZone. This is known as a nonsecure dog bite and it collects register values, downloads the cache contents, and resets the device. The watchdog driver currently schedules itself to pet the watchdog every 3 sec.

For debugging purposes, you can disable the watchdog module by the following method:

- If the watchdog is enabled at bootup (enable=1), the runtime_disable sysfs node at /sys/devices/b017000.qcom,wdt/disable can be used to deactivate the watchdog

Note the dog bark is only received if the processor is not stuck with interrupts enabled and the hardware is still in a functional state, which is not common. If the dog bark cannot be handled and the system gets stuck for any reason, the dog bite occurs and resets the target. The Register information stored by TrustZone is used to debug non-secure apps dog bite issues.

Secure watchdog timer

The secure watchdog timer is managed by SRoT running on the apps processor.

The bark event of this timer is sent to SRoT as an interrupt. If SRoT becomes nonresponsive, the integrity and stability of the entire SoC is in question and the only course of action is a reset to cold boot. Therefore, the bite event of this timer resets the PS_HOLD register, causing the PS_HOLD to transit to a logic-0 state.

This causes the PMIC to enter a chipset reset sequence.

The registers in the secure watchdog timer shall only be accessible by SRoT.

The access control infrastructure must prevent any other EEs than SRoT from accessing the secure watchdog.

The secure watchdog timer will automatically resume from the hardware upon any wakeup event to SRoT when SRoT is in Sleep mode.

The secure watchdog timer must be kept in the Disabled state when the chip exit cold reset until SRoT enables it. Since only the secure watchdog timer can cause the PMIC to reset the MSM, it is also used to respond to exit errors from chip sleep states (XO shutdown and/or VDD minimization) by clearing PS_HOLD to cause PMIC to reset the MSM. The secure watchdog timer will automatically resume on any wakeup event, which directs the chip to come out of a chip sleep state (VDDmin or XO shutdown). Safe watchdog barking is an outage for SRoT only.

MPSS Watchdog

The MPSS watchdog module can generate BARK and BITE events:

1. BARK (FIQ) – Bark time is programmable

2. BITE (Reset) – Watchdog BITE generates an IRQ for the apps processor; MPSS watchdog BITE time is fixed at ~390 ms

An FIQ is generated when BARK time is hit. A BARK event is handled as a software error in the MPSS. Therefore, the MPSS follows generic error handling procedures.

When the MPSS watchdog timer count reaches 0, the hardware watchdog asserts an interrupt to the apps processor (A7 apps processor). An NMI is generated locally. As the name NMI suggests, this interrupt cannot be blocked. The NMI handler on the subsystem then triggers the error handler that saves the context of the system and informs the apps processor of the error.

BITE interrupts on the MPSS indicate unresponsive subsystems due to a software glitch and/or hardware error such as bus lockup, memory corruption, etc. The A7 apps processor generates a kernel panic once it receives a BITE interrupt from MPSS unless the subsystem restart feature is enabled. In addition to hardware watchdog support, the modem implements a software watchdog, which consists of the DOG task, and a protocol imposed on reporting activities in the system. The monitored activities call a function to report to DOG. The activity of the DOG software is one of the most important activities in the system. It defines a software timer, for example, 100 ms, which causes it to run periodically to reset the hardware watchdog timer, that is to stroke the hardware watchdog and prevent hardware watchdog timeout. Each monitored activity has a defined timeout period during which it must report to the DOG activity to indicate proper operation. These activities generally set a software timer to schedule their reporting to the DOG activity. Since activity latency delays make planning and execution of activities uncertain, activities must be reported significantly before their timeout period to reliably avoid a dog timeout.

Periodically, for example, once every 100 milliseconds, the DOG activity checks the signaling status of the monitored activities to detect the signaling of delayed activity (dog timeout).

Power Management

Hardware Power Architecture/Features

- Single XO
  - CXO (19.2 MHz) sources all clocks. CXO is always ON as its buffers are turned off when XO shutdown is exercised. The sleep clock (32 KHz) used by the MDM chipset in XO Shutdown mode is derived from CXO; it is used to clock always-on domains such as the MDM Power Manager (MPM), parts of the modem core, and certain timer circuits.
- Multiple voltage domains – Memory, digital, and APC

- VddMx (memory rail) – VDD_MX represents on-chip memories. This power domain is never switched off but can be minimized to lower power, typically when MDM is in deep sleep with XOs turned off (memory voltage does not scale in Low-Power modes with the logic domains).
- VddCx (digital rail) – VDD_CX represents all digital logic circuits/cores of MDM that must retain a minimum voltage, even when MDM is in deep sleep (with XOs off). RPM core, system fabric, and MDM9x07 digital cores are some of the examples that depend on this power domain.
- Vdd APC (APC rail) – VDD_APC rail powers up the APSS subsystem.

Software Power Features

- XO shutdown
    - During XO shutdown, CXO buffers at PMIC are switched off, leaving the CXO crystal still running.
    - CXO still needs to be generated as it is used to generate the 32 KHz sleep clock in PMIC.
    - No clock is fed to MDM9x07 except the 32 KHz sleep clock, which is required for the always-on parts; for example, the MPM block.
    - To enter XO shutdown, all master processors must be in Power collapse mode, dedicated clocks should be off, and shared clocks must be rated “off” by their clients.
- Vdd minimization
    - Vdd minimization is the deepest Low-power mode that can be achieved in the system by minimizing VddCx and VddMx to their lowest possible voltage.
    - When Vdd minimization is achieved, the chip is not operational, except for detecting wake-up interrupt/timer expiration; however, all hardware (supplied by Vdd Cx/Mx) states are still maintained. These two power domains cannot be power-collapsed, so they are put to a lower voltage that can sustain the contents in memories, and so on.
    - Lowering the voltage saves leakage current, and therefore reduces overall power consumption.
    - To enter into Vdd minimization, all master processors should vote for XO Shutdown as well as retention voltage on Cx/Mx.

APPS Power features

Supported Low-Power modes

- SWFI – SWFI is an ARM-supported instruction. In this mode, the processor clock is stopped, no instructions are executed, and the register states are preserved. Core exits this Low-power mode upon any interruption.
- Standalone power collapse (without RPM notification) – Clock gate and power collapse for the individual core. Processor clock is stopped and power-collapse for individual core without assistance from RPM.
- Power collapse (with RPM notification) – This Low-power mode includes switching off the power of the core in addition to stopping the processor clock.
    - RPM-assisted power-collapse.

    - Subsystems vote for Low-power mode on shared resources such as XO, L2 cache, and Vdd Cx/Mx to the RPM.
    - There is a higher recovery penalty to resume an Active state

Suspend Features for LE(Linux)

Wakeup Sources Replacing Legacy Android Wakelock

- Enables drivers (kernel space/user space) to finish transactions in progress before going into suspension
- To verify active wakeup sources: cat /sys/kernel/debug/wakeup_sources
- Wakeup sources are good for tracking kernel-originated events, but they do not provide any way for user space to indicate that the system should not sleep
- An application can write a name (and an optional timeout) to /sys/power/wake_lock to establish a new, active wakeup source. The wake lock name provided will be written to the wakeup sources. That source will prevent the system from being suspended until either its timeout expires or the same name is written to/sys/power/wake_unlock.
- A driver receiving a wakeup event will mark the associated wakeup source as active, keeping the system running. That source will remain active until user space consumes the event. But before doing so, the user-space application takes a "wake lock" of its own, ensuring it will be able to complete its processing before the system goes back to sleep.

Autosleep

- Automode is enabled or disabled by writing on/off to /sys/power/autosleep (libsuspend)
- try_to_suspend() function is the core of Autosleep
- Continuous looping (without polling) on itself to see if all active wakeup sources have been released while Automode is on

CPUIdle

CPUidle is a kernel PM infrastructure.

CPUs today support multiple idle levels differentiated by varying exit latencies and power consumption during inactivity.

CPUidle enables efficient management of these different idle CPUs.

It separates drivers that can provide support for multiple types of Idle states and policy governors that decide which Idle state to use at runtime.

The CPUidle driver can support multiple idle states based on parameters like variable power consumption, wakeup latency, etc.

The main advantage of this infrastructure is that it allows independent development of drivers and governors and allows for better CPU PM.

CPUidle provides idle state detection capability and can also support entry/exit into CPU Idle states

CPUidle initializes the cpuidle_device structure for each CPU device and registers with cpuidle using cpuidle_register_device

CPUFreq

CPU Frequency (CPUFreq) scaling is the PM method used in running mode.

Sysfs interface – /sys/devices/system/cpu/cpu*/cpufreq/* (Linux)

Switching of CPUFreq is governed by these governors in a static or dynamic manner:

- Performance – CPU runs at static maximum frequency
- Powersave – CPU runs at a static minimum frequency
- User space – Determined by the static user space program
- Ondemand – On-demand dynamic governor sets target frequency based on CPU busy/idle statistics
- Conservative – Conservative dynamic governor sets target frequency, based on CPU busy/idle statistics

User space interface for LE (Linux)

LE910Cx provides the following power management capabilities, accessible to the user space:

1. Wakelocks- Just like the LE910, a wakelock is used to prevent the system from going into sleep and is accessible to a user space application. Wakelock is implemented as a wake-up source.
2. Autosleep- This allows the user to override the above wakelock mechanism.
    - Setting to "off" will force the system not to sleep regardless of the wakelocks.
    - Setting to "mem" will cause the system to use the wake locks to decide whether to sleep or not.

Performance Build (Linux)

The LE910Cx offers a performance-optimized build along with the non-performance-optimized build provided so far. Any tests performed on the A4 regarding performance, especially on the following topics, should be performed on the performance build.

The main differences between these two builds are the powerup time and throughput in various scenarios, including WWAN, WLAN, and Ethernet.

Performance optimized build changes (oppose to the non perf) the following images:

1. Linux LK
2. Linux Kernel
3. Linux root FS

The following configurations are removed when using the performance build:

1. CONFIG_KALLSYMS_ALL - This option ensures that all symbols are loaded into the kernel image (that is, symbols of all sections) at the cost of a larger kernel size
2. CONFIG_PROFILING – This option enables the extended profiling support mechanisms used by profilers
3. CONFIG_NETFILTER_DEBUG – This option adds additional messages useful in debugging the netfilter code
4. CONFIG_DYNAMIC_DEBUG – This option compiles debug level messages into the kernel, which would not otherwise be available at runtime.
5. CONFIG_DEBUG_PAGEALLOC – This option Unmaps pages from the kernel linear mapping after free_pages(). This results in a large slowdown but helps to find certain types of memory corruptions.
6. CONFIG_DEBUG_KMEMLEAK – This option enables the memory leak detector and introduces an overhead to the memory allocation.
7. CONFIG_DEBUG_KMEMLEAK_DEFAULT_OFF
8. CONFIG_DEBUG_STACK_USAGE – This option enables the display of the minimum amount of free stack which each task has ever had available.
9. CONFIG_DEBUG_MEMORY_INIT – This option enables additional sanity checks during memory initialization.
10. CONFIG_DEBUG_SPINLOCK – This option enables catching missing spinlock initialization and certain other kinds of spinlock errors commonly made.
11. CONFIG_DEBUG_MUTEXES – This option allows mutex semantics violations and mutex-related deadlocks (lockups) to be detected and reported automatically.
12. CONFIG_DEBUG_ATOMIC_SLEEP – When this option is enabled, various routines which may sleep will become very noisy if called within atomic sections: when a spinlock is held, within pre-disabled sections, inside an interrupt, and so on.
13. CONFIG_DEBUG_LIST – This option enables extended checks in linked-list walk routines
14. CONFIG_FAULT_INJECTION_DEBUG_FS – This option enables the configuration of fault-injection capabilities via debugs.
15. CONFIG_FAULT_INJECTION_STACKTRACE_FILTER – This option provides a stack trace filter for fault-injection capabilities.
16. CONFIG_BLK_DEV_IO_TRACE – This option enables tracing the block layer actions on a given queue. The trace allows you to view the traffic in progress on a block device queue.

17. CONFIG_DEBUG_USER - When a user program crashes due to an exception, the kernel can print a brief message explaining what the problem was. This is sometimes helpful for debugging but is useless on a production system.

In addition to the above, the perf build also disables the debug console starting from the Linux LK through the Kernel image. Disabling the console is making almost 90% of the differences in bootup time between these two builds. As a result, in a performance build, there would be no messages printed out to the serial console and no login services would be provided using the serial console.

5.3 Basic Operations

Command Syntax

In the next paragraphs the following notations are used:

<cr> represents the Carriage Return Character (13)

<lf> represents the Line Feed Character (10)

<xx> represents a parameter with changing the name is in place of the double x. (< and > characters are only for limiting the parameter and must not be sent to the terminal).

[<xx>] represents an optional parameter whatever the name in place of the xx.

[and] characters are only for limiting the optional parameter and must not be sent to the terminal).

Command Response Timeout

Every command issued to the Telit modules returns a result response if response codes are enabled (default). The time required to process the given command and return the response varies, depending on the type of command. The commands that do not interact with the SIM/UICC or the network, and only involve the internal configuration settings or readings, have an immediate response, depending on the configuration of the SIM/UICC configuration (for example, the number of contacts stored in the address book, number of stored SMS), or on the network with which the command can interact.

The table below lists only the commands whose interaction with the SIM/UICC or the network could lead to long response timings. Unless otherwise specified, timing refers to the set command. For commands related to writing and reading the phonebook and SMS, timing refers to the commands issued after the sorting of the phone book has been completed. For sending and dialing DTMF commands, the timing refers to the module registered on the network ("AT+CREG/+CEREG?" answer is "+CREG/+CEREG: 0,1" or "+CREG/+CEREG: 0,5").

Note: In case no response is received after the timeout time has elapsed, then try repeating the last command and if still no response is received until the timeout time, an Unconditional Shutdown MUST be issued, and the device must be powered ON again.

Table 9: AT Commands with Long Response Timeout

Command	Time-Out (Seconds)
+COPS	180 (test command)
+CLCK	15 (SS operation)
	5 (FDN enabling/disabling)
+CPWD	15 (SS operation)
	5 (PIN modification)
+CLIP	15 (read command)
+CLIR	15 (read command)
+CCFC	15
+CCWA	15
+CHLD	60
+CPIN	30
+CPBS	5 (FDN enabling/disabling)
+CPBR	5 (single reading)
	15 (complete reading of 500 records full phonebook)
+CPBF	10 (string present in a 500 records full phonebook)
	5 (string not present)
+CPBW	5
+CACM	5
+CAMM	5
+CPUC	180
+VTS	20 (transmission of full "1234567890*#ABCD" string with no delay between tones, default duration)
+CSCA	5 (read and set commands)

Command	Time-Out (Seconds)
+CSAS	5
+CRES	5
+CMGS	120 after CTRL-Z; 1 to get '>' prompt
+CMSS	120 after CTRL-Z; 1 to get '>' prompt
+CMGW	5 after CTRL-Z; 1 to get '>' prompt
+CMGD	5 (single SMS cancellation)
	25 (cancellation of 50 SMS)
+CNMA	120 after CTRL-Z; 1 to get '>' prompt
+CMGR	5
+CMGL	100
+CGACT	150
+CGATT	90
D	120 (voice call)
	Timeout set with ATS7 (data call)
A	60 (voice call)
	Timeout set with ATS7 (data call)
H	60
+CHUP	60
+COPN	10
+COPL	180
+WS46	10
+CRSM	180
#MBN	10
#TONE	5 (if no duration specified)
#EMAILD	90
#STSR	30
#GPRS	150
#SKTD	140 (DNS resolution + timeout set with AT#SKTCT)

Command	Time-Out (Seconds)
#QDNS	170
#FTPOPEN	120 (timeout set with AT#FTPTO, in case no response is received from server)
#SGACT	150
#SH	10
#SD	140 (DNS resolution + connection timeout set with AT#SCFG)
#CSURV	180
#CSURVC	180
#CSURVUC	180
#CSURVB	180
#CSURVBC	180
#CSURVP	180
#CSURVPC	180
#CSURVL	180
#CSURVCL	180
#CSURVW	180
#CSURVCW	180
#CSURVG	180
#CSURVCG	180

Basic AT Commands

AT Error Report Format

Disable the Error Report in numerical and verbose format.

AT+CMEE=0

OK

Enable the Error Report in numerical format.

AT+CMEE=1

OK

Enable the Error Report in verbose format.

AT+CMEE=2

RAT and Band Selection

RAT Selection

The following AT command selects the technology: 2G, 3G, 4G, or automatic.

AT+WS46=[<n>]

<n> - integer type, it is the WDS-Side Stack used by the TA.

12	GSM Digital Cellular Systems (GERAN only)
22	UTRAN only
25	3GPP Systems (GERAN and UTRAN and E-UTRAN)
28	E-UTRAN only
29	GERAN and UTRAN
30	GERAN and E-UTRAN
31	UTRAN and E-UTRAN
32	TDSCDMA only
33	GERAN and TDSCDMA
34	TDSCDMA and E-UTRAN
35	GERAN and TDSCDMA and E-UTRAN
36	GERAN and TDSCDMA and UTRAN and E-UTRAN
37	GERAN and TDSCDMA and UTRAN
38	TDSCDMA and UTRAN

39 TDSCDMA and UTRAN and E-UTRAN

Note: The <n> parameter is stored in NVM, and the command will take effect on the next power on.

The factory default value depends on each variant.

Band Selection

In manual band selection, the following AT command selects the current band for both technologies GERAN, UTRAN, and EUTRAN:

AT#BND=<GSM band>[,<UMTS band>[,<LTE band>]]

Examples

AT#BND=0,0,2 selected band: GSM(2G) + DCS(2G) + B1 (3G) + B2(4G)

OK

Note: The input range for the supported bands depends on variants.

SIM/USIM Management

SIM Presence and PIN Request

The following AT command checks if the SIM device needs the PIN code:

AT+CPIN?

Examples

Assume that the SIM is inserted into the module and the PIN code is needed.

AT+CPIN?

+CPIN: SIM PIN

OK

Assume that the SIM is not inserted, and the Extended Error result code is not enabled. Check if a PIN code is needed, just to see the response command:

AT+CPIN?

ERROR

Assume that the SIM is not inserted, and the Verbose Extended error result code is enabled. Check if a PIN code is needed, just to see the response command:

AT+CPIN?

+CME ERROR: SIM not inserted

Assume that the SIM is not inserted, and the Numerical Extended error result code is enabled. Check if a PIN code is needed, just to see the response command:

AT+CPIN?

+CME ERROR: 10

Enter PIN Code

Use the following AT command to enter the PIN code:

AT+CPIN=<pin>

Examples

Assume to enter a wrong PIN code, and the Extended Error result is not enabled.

AT+CPIN=1235

ERROR

Now, enter the right PIN code:

AT+CPIN=1234

OK

Enable Verbose Extended error result code:

AT+CMEE=2

OK

Enter a wrong PIN code:

AT+CPIN=1235

+CME ERROR: incorrect password.

Note: After 3 unsuccessful attempts with the PIN, the PIN code is no longer requested, and the SIM is locked. Use SIM PUK to enter a new PIN code and unlock the SIM.

Enter PUK Code

Enter the following AT command if PUK or PUK2 code is required:

AT+CPIN=<pin>[,<newpin>]

Note: After 10 PUK code failed attempts, the SIM Card is locked and no longer available.

SIM Status

Use the following AT command to enable/disable the SIM Status Unsolicited Indication.

AT#QSS=<mode>

Example 1

Enable the unsolicited indication concerning the SIM status change.

```
AT#QSS=1            enable URCs: #QSS:0/1
OK
#QSS: 0             unsolicited indication: SIM is extracted.
#QSS: 1             unsolicited indication: SIM is inserted.
```

Example 2

```
AT#QSS=2            enable URCs: #QSS:0/1/2/3
OK
AT+IPR=19200        select the Main Serial Port speed = DTE speed
OK
AT&W0               store the setting on profile 0
OK
AT&P0               at Power on use profile 0
OK
```

Now, power off the module:

```
#QSS: 0             unsolicited indication: SIM is extracted.
```

Now, power on the module:

```
#QSS: 1             unsolicited indication: SIM is inserted.
```

AT+CPIN?	
+CPIN: SIM PIN	SIM is locked
OK	
AT+CPIN=<PIN>	enter PIN
OK	
#QSS: 2	unsolicited indication: SIM is unlocked.
#QSS: 3	unsolicited indication: SMS and Phonebook are accessible

SIM Detection Mode

Use the following AT command to manage the SIM Detection Mode:

AT#SIMDET=<mode>

Or

Use the following AT command to enable/disable the SIM Status Unsolicited Indication.

AT#SIMPR = <mode>

Example

AT#SIMDET?

#SIMDET: 2,1

OK

2 = automatic SIM detection through SIMIN pin (Factory Setting)

1 = SIM inserted

AT#SIMPR?

#SIMPR: 0, 0, 1

#SIMPR: 0, 1, 0

OK

First Line:

0 = Disable URC

0 = Local UICC

1 = SIM inserted

Second Line:

0 = Disable URC

1 = Remote SIM

0 = Remote SIM not connected (If SIM/UICC Access Profile of BT is supported)

Enable the unsolicited indication concerning the SIM status change.

```
AT#QSS=1
OK
Now, extract the SIM
#QSS: 0               unsolicited indication: SIM is extracted
Now, insert the SIM
#QSS: 1               unsolicited indication: SIM is inserted
AT#SIMDET=0           simulate SIM not inserted, but it is still physically inserted
OK
#QSS: 0               unsolicited indication, but SIM is NOT physically extracted
AT#SIMDET?
#SIMDET: 0,1          0 = simulate the status SIM not inserted, 1 = SIM is
OK                        physically inserted
AT#SIMPR?
#SIMPR: 0, 0, 1       0: Disable URC, 0: Local SIM, 1: SIM inserted
#SIMPR: 0, 1, 0       0: Disable URC, 1: Remote SIM, 0: Remote SIM not inserted
OK
Now, extract/insert the SIM, no unsolicited indication appears on DTE!
Extract the SIM again
AT#SIMDET=1           simulate SIM inserted, but it is still physically extracted
OK
AT#SIMDET?
#SIMDET: 1,0          1 = simulate the status SIM inserted, 0 = SIM is physically
OK                        not inserted
AT#SIMPR?
#SIMPR: 0, 0, 0       0: Disable URC, 0: Local SIM, 0: SIM is physically not inserted
#SIMPR: 0, 1, 0       0: Disable URC, 1: Remote SIM, 1: Remote SIM not inserted
OK
```

Now, insert/extract the SIM, no unsolicited indication appears on DTE!

```
AT#SIMPR=1          Enable URC
OK
```

Extract the SIM and set automatic SIM detection

```
#SIMPR: 0, 0    0: Local SIM, 0: SIM is physically not inserted
#SIMPR: 1, 0        1: Remote SIM, 0: Remote SIM is not connected from SAP
AT#SIMDET=2
OK
AT#SIMDET?
#SIMDET: 2,0        2 = automatic SIM detection through SIMIN pin (Factory Setting),
OK                   0 = SIM not inserted
```

Now, insert/extract the SIM, unsolicited indication appears again on DTE!

```
#SIMPR: 0, 1    0: Local SIM, 0 SIM is physically inserted
#SIMPR: 1, 0    1: Remote SIM, 0: Remote SIM is not connected from SAP
#QSS: 1             unsolicited indication: SIM is logically activated
#QSS: 0             unsolicited indication: SIM is logically deactivated
```

SIM/USIM Access File

AT+CSIM command is used to read/write SIM/USIM files. The format of the AT+CSIM parameters and the sequence of the AT+CSIM commands must be in accordance with the protocol card. The distinction between SIM and USIM <command> format is needed because the AT+CSIM command works directly on the card.

```
AT+CSIM=<length>,<command>
```

To read/write card files, refer to the LE910Cx AT Command Reference Guide.

MSISDN

MSISDN is a number uniquely identifying a subscription to a GSM or UMTS mobile network. MSISDN is defined by the ITU-U Recommendation which defines the numbering plan: a number uniquely identifies a public network termination point and typically consists of three fields, CC (Country Code), NDC (National Destination Code), and SN (Subscriber Number), up to 15 digits in total.

Select the "ON" storage:

```
AT+CPBS="ON"
OK
```

Write a new record on the selected storage:

```
AT+CPBW=1,"+393912457",145,"MyNumber"
OK
```

Read the just entered number:

```
AT+CPBF="MyNumber"
+CPBF:  1," +393912457",145,"MyNumber"
OK
```

Preferred Operator List

Use the following AT command to manage the Preferred Operator List stored on SIM/USIM.

```
AT+CPOL=[<index>][,<format>[,<oper>[,<GSM_AcT>,<GSM_Compact_AcT>,<UTRAN_AcT>
,<EUTRAN_AcT>]]]
```

Examples

Check the supported number of operators in the SIM Preferred Operator List and the format:

```
AT+CPOL=?
+CPOL: (1-16),(0-2)
OK
```

The used SIM supports 16 positions; the supported format (2) is numeric. In addition, format (0) is long format alphanumeric, and (1) is short format alphanumeric.

Reading the entire list:

```
AT+CPOL?
+CPOL: 1,2,"45005",1,1,1,1
+CPOL: 2,2,"45005",0,0,1,1
+CPOL: 3,2,"00102",1,1,1,1
+CPOL: 4,2,"00101",1,1,0,1
+CPOL: 5,2,"00101",1,1,1,1
+CPOL: 6,2,"111222",1,1,1,1
+CPOL: 7,2,"00102",1,1,1,1
+CPOL: 15,2,"45008",1,1,1,1
```

+CPOL: 16,2,"45007",0,0,0,1

OK

The meaning of the string "XXXYY" is:

- XXX = Mobile Country Code

- YY = Mobile Network Code

The last 4 digits are GSM, GSM compact, UTRA, and EUTRAN access technology sequentially.

Delete the first entry using a non-existent <format> value just to see the response when the Extended Error result code is enabled:

AT+CPOL=1,3

+CME ERROR: operation not supported

Now, delete the first entry using the right <format> value:

AT+CPOL=1,2

OK

AT+CPOL?

+CPOL: 2,2,"20810",1,1,0,0

+CPOL: 3,2,"23205",1,0,1,0

...

+CPOL: 15,2,"23802",1,1,0,1

+CPOL: 16,2,"24201",1,0,1,1

OK

The entry on first position is deleted

AT+CPOL=1,2,20801,1,1,1,1 < Write a new entry in the first position

OK

Check if the new entry is written in first position:

AT+CPOL?

+CPOL: 1,2,"20801",1,1,1,1 < The new entry is written on the first position

+CPOL: 2,2,"20810",1,1,0,0

...

+CPOL: 16,2,"24201",1,0,1,1

OK

Network Checking

Network Survey

Use the following AT command to perform a quick survey through channels belonging to the current band.

AT#CSURV[=[<s>,<e>]]

Parameters: <s> - starting channel, <e> - ending channel

Examples

AT#CSURV

Network survey started ...

earfcn: 1350 rxLev: -59 mcc: 450 mnc: 05 cellId: 7323719 tac: 12556 phyCellId: 64 cellStatus: CELL_SUITABLE rsrp: -95 rsrq: -16 bw: 20

earfcn: 2500 rxLev: -66 mcc: 450 mnc: 05 cellId: 448779 tac: 12556 phyCellId: 87 cellStatus: CELL_SUITABLE rsrp: -97 rsrq: -11 bw: 20

earfcn: 100 rxLev: -43 mcc: 450 mnc: 06 cellId: 51999244 tac: 8471 phyCellId: 245 cellStatus: CELL_FORBIDDEN rsrp: -71 rsrq: -11 bw: 10

earfcn: 3743 rxLev: -54 mcc: 450 mnc: 08 cellId: 2486272 tac: 27 phyCellId: 245 cellStatus: CELL_FORBIDDEN rsrp: -85 rsrq: -11 bw: 20

earfcn: 1550 rxLev: -55 mcc: 450 mnc: 08 cellId: 2486275 tac: 27 phyCellId: 245 cellStatus: CELL_FORBIDDEN rsrp: -83 rsrq: -11 bw: 10

earfcn: 1694 rxLev: -59 mcc: 450 mnc: 08 cellId: 2486293 tac: 27 phyCellId: 29 cellStatus: CELL_FORBIDDEN rsrp: -87 rsrq: -11 bw: 10

earfcn: 2600 rxLev: -56 mcc: 450 mnc: 06 cellId: 51999242 tac: 8471 phyCellId: 245 cellStatus: CELL_FORBIDDEN rsrp: -84 rsrq: -11 bw: 10

earfcn: 3895 rxLev: -59 mcc: 450 mnc: 08 cellId: 2486293 tac: 27 phyCellId: 29 cellStatus: CELL_FORBIDDEN rsrp: -87 rsrq: -11 bw: 10

earfcn: 1350 rxLev: -72 phyCellId: 45 cellStatus: CELL_SUITABLE rsrp: -102 rsrq: -20

earfcn: 2500 rxLev: -47 phyCellId: 273 cellStatus: CELL_SUITABLE rsrp: -64 rsrq: -8

uarfcn: 10836 rxLev: -74 mcc: 450 mnc: 08 scr code: 1488 cellId: 14909569 lac: 7170 cellStatus: CELL_FORBIDDEN rscp: -75 ecio: -5.0

uarfcn: 10737 rxLev: -52 mcc: 450 mnc: 05 scr code: 224 cellId: 63808804 lac: 8673 cellStatus: CELL_SUITABLE rscp: -56 ecio: -7.0

Network survey ended

OK

BCCH Survey

Use the following AT command to perform a quick survey of the channels belonging to the current band. The survey stops as soon as <n> BCCH carriers are found.

```
AT#CSURVB=[<n>]
```

Parameters: <n> - number of desired BCCH carriers.

Examples

```
AT#CSURVB=2

Network survey started ...

uarfcn: 10737 rxLev: -90 mcc: 450 mnc: 05 scr code: 224 cellId: 63808804 lac: 8673 cellStatus: CELL_SUITABLE rscp: -91 ecio: -5.0

uarfcn: 10836 rxLev: -98 mcc: 450 mnc: 08 scr code: 1488 cellId: 14909569 lac: 7170 cellStatus: CELL_FORBIDDEN rscp: -101 ecio: -7.0

Network survey ended

OK
```

Note: The #CSURVB command could not be executed on the module that supports only 4G.

Network Information

Network Status

Enter the following AT command to verify if the module is registered on a network.

```
AT+CREG?
```

Circuit Service Network Registration Status in UTRAN/E-UTRAN

Send command AT+CREG?

Wait for a response:

Table 10: +CREG Status

Response	Reason	Action
+CREG: 0,0 or +CREG: 1,0	SIM is not present or damaged or SIM is present, and PIN is required to continue operations	Check the inserted SIM/USIM status (Please refer to the “SIM/USIM management “contents)

Response	Reason	Action
+CREG: 0,1 or +CREG: 1,1	Mobile is registered on its home network.	Proceed ahead. Ready to CS call
+CREG: 0,2 or +CREG: 1,2	Mobile is currently not registered on any network but is looking for a suitable one to register.	Repeat procedure at "Fast Network Status Check" contents to see if it has found a suitable network to register in
+CREG: 0,3 or +CREG: 1,3	Mobile has found some networks, but it is not allowed to register on any of them, no roaming was allowed.	Try in another place or reset, then repeat the procedure at "Fast Network Status Check" contents
+CREG: 0,4 or +CREG: 1,4	Mobile is in an unknown network status	Repeat procedure at "Fast Network Status Check" contents to see if it has found a suitable network to register in
+CREG: 0,5 or +CREG: 1,5	Mobile has found some networks and is currently registered in roaming on one of them	Proceed ahead. Ready to CS call

Packet Service Network Registration Status in UTRAN

Send command AT+CGREG?

Wait for a response:

Table 11: +CGREG Status

Response	Reason	Action
+CGREG: 0,0 or +CGREG: 1,0	SIM not present or damaged or SIM is present and PIN is required to continue operations	Check the inserted SIM/USIM status (Please refer to the "SIM/USIM management "contents)
+CGREG: 0,1 or +CGREG: 1,1	Mobile is registered on its home network.	Proceed ahead. Ready to PS call

Response	Reason	Action
+CGREG: 0,2 or +CGREG: 1,2	Mobile is currently not registered on any network but is looking for a suitable one to register.	Repeat procedure at “Fast Network Status Check” contents to see if it has found a suitable network to register in
+CGREG: 0,3 or +CGREG: 1,3	Mobile has found some networks, but it is not allowed to register on any of them, no roaming was allowed.	Try in another place or reset, then repeat the procedure at “Fast Network Status Check” contents
+CGREG: 0,4 or +CGREG: 1,4	Mobile is in an unknown network status	Repeat procedure at “Fast Network Status Check” contents to see if it has found a suitable network to register in
+CGREG: 0,5 or +CGREG: 1,5	Mobile has found some networks and is currently registered in roaming on one of them	Proceed ahead. Ready to PS call

Packet Service Network Registration Status in E-UTRAN

Send command AT+CEREG?

Wait for a response:

Table 12: +CEREG Status

Response	Reason	Action
+CEREG: 0,0 or +CEREG: 1,0	SIM not present or damaged or SIM is present and PIN is required to continue operations	Check the inserted SIM/USIM status (Please refer to the “SIM/USIM management “contents)
+CEREG: 0,1 or +CEREG: 1,1	Mobile is registered on its home network.	Proceed ahead. Ready to PS call
+CEREG: 0,2 or +CEREG: 1,2	Mobile is currently not registered on any network but is looking for a suitable one to register.	Repeat procedure at “Fast Network Status Check” contents to see if it has found a suitable network to register in

Response	Reason	Action
+CEREG: 0,3 or +CEREG: 1,3	Mobile has found some networks, but it is not allowed to register on any of them, no roaming was allowed.	Try in another place or reset, then repeat the procedure at "Fast Network Status Check" contents
+CEREG: 0,4 or +CEREG: 1,4	Mobile is in an unknown network status	Repeat procedure at "Fast Network Status Check" contents to see if it has found a suitable network to register in
+CEREG: 0,5 or +CEREG: 1,5	Mobile has found some networks and is currently registered in roaming on one of them	Proceed ahead. Ready to PS call

Note: When a response +CREG/+CGREG/+CEREG: x,1 or + CREG/+CGREG/+CEREG: x,5 is received, then the device is ready to place and receive a call or SMS. It is possible to jump directly to call setup procedures or SMS sending procedures.

Network Operator Identification

Use the following AT command to query the module for Network Operators Identifications Once the mobile has registered on some network (or even if it has returned +CREG/+CGREG/+CEREG:x,3), it is possible to query the mobile for network identifications, codes, and names:

- Send command AT+COPS=?
- Wait For a response in the format:

+COPS: [list of supported (<stat>,long alphanumeric <oper>,short alphanumeric <oper>,numeric <oper>,< AcT>)s]

[,,(list of supported <mode>s),(list of supported <format>s)]

where:

<stat> operator availability

0 - unknown

1 - Available

2 - current

3 - Forbidden

<AcT> access technology selected

0 GSM

2 UTRAN

7 E-UTRA UTRAN

Note: Since with this command a network scan is done, this command may require some seconds before the output is given.

For example:

AT Command

```
AT+COPS=?
```

Answer:

```
+COPS: (2,"","SKTelecom","45005",7),(3,"KT","KT","45008",7),(3,"KOR LG Uplus","LG U+","45006",7),,(0-4),(0-2)
OK
```

In this case, the mobile is registered on the network "SKTelecom" which is a network from Korea, code: 450 and Network ID: 05.

The other network is not available for registration:

Note: This command issues a network request and may require quite a long time to respond since the device has to wait for the answer from the network (this can take up to 180 seconds). Do not use this command unless necessary.

Signal Strength and Quality

Assume that the mobile is registered on a Network that can be: GERAN or UTRAN. The following AT command can be useful to know the strength and quality of the received signal indicate the radio link reliability.

```
AT+CSQ
```

Examples

Assume that the antenna is not connected to the Telit Module or Network coverage is not present at all.

```
AT+CSQ
+CSQ: 99,99
OK
```

Now, the antenna is connected to the Telit Module and Network coverage is present. Enter again the previous AT command:

```
AT+CSQ
+CSQ: 17,0
OK
```

17 = <rssi> = Received Signal Strength Indication

0 = <ber> = Bit Error Rate

Now, a wrong parameter is entered just to see the result format when the Verbose Extended Error result is enabled

```
AT+CSQ?
+CME ERROR: operation not supported
```

Extended Signal Quality

Assume that the mobile is registered on a Network that can be: GERAN, UTRAN, and EUTRAN.

The following AT command can be useful to know the strength & quality of the received signal indicate the reliability of the radio link.

```
AT+CESQ
```

Examples

Assume that the antenna is not connected to the Telit Module or Network coverage is not present at all.

```
AT+CESQ
+CESQ: 99,99,255,255,255,255
OK
```

Now, the antenna is connected to the Telit Module and GERAN Network coverage is present.

Enter again the previous AT command:

```
AT+CESQ
+CESQ: 61,5,255,255,255,255
OK
```

61 = <rxlev> = Received Signal Strength Level.

5 = <ber> = Bit error rate (in percent).

Now, the antenna is connected to the Telit Module and UTRAN Network coverage is present.

Enter again the previous AT command:

```
AT+CESQ
+CESQ: 99,99,94,47,255,255
OK
```

94 = <rscp> = Received Signal Code Power.

47 = <ecno> = Ratio of the received energy per PN chip to the total received power spectral density.

Now, the antenna is connected to the Telit Module and EUTRAN Network coverage is present.

Enter again the previous AT command:

```
AT+CESQ
+CESQ: 99,99,255,255,32,95
OK
```

32 = <rsrq> = Reference Signal Received Quality.

95 = <rsrp> = Reference Signal Received Power.

Now, a wrong parameter is entered just to see the result format when Verbose Extended Error result is enabled.

```
AT+CESQ?
+CME ERROR: operation not supported.
```

Fast Network Status Check

Once the module is registered on a network, regardless of the technology (3G or 4G), it is useful to know the strength of the received signal and the network on which the module is registered. This information is collected using the following standard AT commands: +CREG, +COPS, and +CSQ. These commands are not fast in response due to network response time, especially the +COPS command. If the user objective is to keep his application as general as possible, he can use the standard AT command mentioned above.

Telit modules provide proprietary AT commands to gather all the information in a faster and simpler way, they are:

- AT#MONI
- AT#SERVINFO

Use the following AT command to select cells and collect their information:

- UTRAN mode

```
AT#MONI
#MONI: KOR SK Telecom PSC:15 RSCP:-102 LAC:21E1 Id:63809024 EcIo:0.0
UARFCN:10713 PWR:-64dbm DRX:0 SCR:240
OK
```

- E-UTRA`N mode

```
AT#MONI
#MONI: KOR SK Telecom RSRP:-91 RSRQ:-7 TAC:310F Id:135386691 EARFCN:200
PWR:-62dbm DRX:32
OK
```

- UTRAN mode

  Collect only the Serving Cell Network Information:

```
AT#SERVINFO
#SERVINFO: 10713,-64,"KOR SK Telecom","45005",15,21E1,0,3,-0,"I",01,12800
OK
```

- E-UTRAN mode

  Collect only the Serving Cell Network Information:

```
AT#SERVINFO
#SERVINFO: 200,-61,"KOR SK Telecom","45005",811D643,310F,32,3,-89
OK
```

Note: AT#MONI and AT#SERVINFO commands should be used only to collect network name and signal strength information. To check if the module is registered or it is looking for a suitable network to register on, use the +CREG command. If the network signal is too weak and the module loses the registration until a new network is found the two commands report the last measured valid values and not the real ones. The TA (timing advance parameter) is valid only during a call.

Note: Check the network registration with the +CREG command. When the module is registered, query the module for network operator name and signal strength with AT#MONI command.

3G Network

Suppose that the 3G Technology is present in the air. Use the command AT+WS46=22 to force the module into 3G mode.

Examples

Check if the module is using 3G Technology:

```
AT+COPS?

+COPS: 0,0,"KOR SK Telecom",2

OK
```

Yes, it is using 3G Technology.

Select the Serving Cell:

```
AT#MONI=0

OK
```

Collect information:

```
AT#MONI

#MONI: KOR SK Telecom PSC:14 RSCP:-64 LAC:21E1 Id:3CDA520 EcIo:-2.5 UARFCN:10713
PWR:-59 dbm DRX:64 SCR:224

OK
```

Use the following AT command to collect only the Serving Cell Information:

```
AT#SERVINFO

#SERVINFO: 10713,-61,"KOR SK Telecom","45005",14,21E1,64,3,-66,"I",01

OK
```

Use this command to get the current network status.

```
AT#RFSTS

#RFSTS: "450 05",10713,14,-5.0,-68,-63,21E1,01,,64,19,0,1,001,3CDA520,

"450050217220238","KOR SK Telecom",3,1

OK
```

4G Network

Suppose that the 4G Technology is present in the air. Use the command AT+WS46=28 to force the module into 4G mode.

Examples

Check if the module is using 4G Technology:

```
AT+COPS?
```

```
+COPS: 0,0,"KOR SK Telecom",7
```

```
OK
```

Yes, it is using 4G Technology.

Select the Serving Cell:

```
AT#MONI=0
```

```
OK
```

Collect information:

```
AT#MONI
```

```
#MONI: KOR SK Telecom RSRP:-79 RSRQ:-9 TAC:310C Id: 06FC047 EARFCN:1350 PWR:-
53dbm DRX:128
```

```
OK
```

Use the following AT command to collect only the Serving Cell Information:

```
AT#SERVINFO
```

```
#SERVINFO: 1350,-60,"KOR SK Telecom","45005",06FC047,310C,128,3,-94
```

```
OK
```

Use this command to get the current network status.

```
AT#RFSTS
```

```
#RFSTS: "450 05",1350,-94,-59,-
13,310C,255,,128,19,0,06FC047,"450050217220238","KOR SK Telecom"3,3
```

```
OK
```

Voice Call Establishment – Originate

Before setting up the Voice Call, it is assumed that the Telit Module is registered on a network and the signal strength is sufficient to carry on a reliable radio link.

Dialing a Phone Number

Use the following AT command to dial up a phone number.

```
ATD<number>;
```

Examples

Call the national number 040-4X92XYX.

```
ATD0404X92XYX;
OK
```

Now, call the national number 040-4X92XYX in international format +39-040-4X92XYX.

```
ATD+390404X92XYX;
OK
```

Disconnect a Call

Use the following AT command to hang up the current Voice Call:

```
ATH
OK
```

Answering an Incoming Call

When an Incoming Call is recognized, the module sends an Unsolicited Code to DTE. Use the following AT command to answer the call.

```
ATA
OK
```

Set Microphone Mute

The following AT command mutes the microphone of the active path:

```
AT+CMUT=1
OK
```

Check the microphone setting:

```
  AT+CMUT?
  +CMUT: 1
  OK
```

5.4 Advanced Operations

Call Management

The modules provide the SMS Service to store, send, receive, and delete an SMS, which is a short text message up to 160 characters long. Before using the SMS messages, you need to configure the Short Message Service.

Identifying the Call Type

The module can identify the call type before answering. To accomplish this feature, the module provides different ring indications (URC) depending on the call type. It is up to the user to enable the extended format reporting of incoming calls using the following AT command.

```
AT+CRC=[<mode>]

OK
```

Examples

Disable extended format reporting, and then assume that the module receives a call.

```
AT+CRC=?    Check the range value

+CRC: (0,1)

OK

AT+CRC=0    Disable extended format reporting.

OK

AT+CRC?

+CRC: 0

OK
```

The module detects a call. Ring indications are displayed on DTE:

```
RING
RING
```

Now, enable extended format reporting, and then assume the module receives a call.

AT+CRC=1	Enable extended format reporting
OK	
AT+CRC?	Check if extended format reporting is enabled
+CRC: 1	
OK	

The module detects a call. Ring indications in extended format are displayed on DTE:

+CRING: VOICE

+CRING: VOICE

Identify the Caller

The Telit Module can identify the caller's number and provide information about it before the call is answered. The Calling Line Indication is displayed on DTE after each RING or +CRING indication. The following AT command is used to enable/disable the Calling Line Indication.

```
AT+CLIP=[<n>]
OK
```

Examples

Enable extended format reporting and caller number identification, and then assume to receive a call.

Enable extended format reporting.

```
AT+CRC=1
OK
```

Check if extended format reporting is enabled.

```
AT+CRC?
+CRC: 1
OK
```

Check the values range.

```
AT+CLIP?
+CLIP: 0,1
OK
```

Enable caller number identification.

```
AT+CLIP=1

OK

AT+CLIP?

+CLIP: 1,1

OK
```

The module detects a call; ring indications and Calling Line Identification of the calling party are displayed on DTE:

```
+CRING: VOICE

+CLIP: “+390404X92XYX”,145,””,128,””,0

+CRING: VOICE

+CLIP: “+390404X92XYX”,145,””,128,””,0
```

Calling Line Indication

The Telit Module can send the Calling Line Indication (CLI) to the other party through the Network when an outgoing call is established. This indication can be restricted (CLIR) in various ways.

CLIR Service Status

Use the following AT command to query the CLIR Service status.

```
AT+CLIR?
```

Examples

Check the current CLIR settings:

```
AT+CLIR?
+CLIR: 0,4
OK
```

<n> = 0 = CLIR module facility in accordance with CLIR Network Service

<m>= 4 = CLIR temporary mode presentation allowed (it is the facility status on the Network)

The <m> parameter reports the status of the service at the Network level. If the CLIR service is not provided by the Network, this service cannot be used, and changing the first parameter <n> will not change the CLI presentation to the other party behavior of the Network.

Restrict/Allow Caller Line ID Indication

Use the following AT command to enable or disable the presentation of the CLI to the called party.

```
AT+CLIR=[<n>]
OK
```

Examples

Disable the CLI presentation to the other party permanently.

Read the supported values.

```
AT+CLIR=?
+CLIR: (0-2)
OK
```

Read the current Module and Network status.

```
AT+CLIR?
+CLIR: 0,4
OK
```

Set to 1 Module status, CLI not sent.

```
AT+CLIR=1
OK
```

Read the current Module and Network status.

```
AT+CLIR?
+CLIR: 1,4
OK
```

Call Barring Control

The Call Barring Service enables the user to control the calls. The user can block all:

- Outgoing calls
- Outgoing international calls
- Outgoing international calls except those for its Country
- Incoming calls
- Incoming calls while roaming.

The User can activate or cancel Call Barring using the AT commands described below. Moreover, the user needs to enter a special access code (Call Barring Access Code) to modify the Call Barring options. Network Operator provides the Call Barring Code for each subscriber. Hereafter the Call Barring Code is indicated as "Network Password provided by the Network Operator".

The network handles the Call Barring Service, so the module sends a network request, and it may take several seconds to get a response from the network. Furthermore, all the Call Barring Service AT commands must be used when the module is registered on some network, otherwise, an error code is returned.

Lock/Unlock the Module

Use the following AT command to lock/unlock the Module or a Network facility:

```
AT+CLCK=<fac>,<mode>[,<passwd>[,<class>]]
```

Read the supported facilities:

```
AT+CLCK=?
+CLCK: ("AB","AC","AG","AI","AO","IR","OI","OX","SC","FD","AL","PN","PU","PP","P
C","PF")
OK
```

Call Barring Service Status

Use the following AT command to require the status of the selected network facility.

```
AT+CLCK=<fac>,2
```

Examples

Check the status of the SIM facility:

```
AT+CLCK="SC",2
+CLCK: 1
OK
```

Check the status of a wrong facility just to see the format response. Before doing that verify the Extended Error result code.

```
AT+CMEE?

+CMEE: 2     verbose format

OK

AT+CLCK="S1",2

+CME ERROR: operation not supported
```

Check "IR" network facility status (Bar Incoming Calls status when roaming outside the home country).

```
AT+CLCK=IR,2
+CLCK: 0,1
+CLCK: 0,2
+CLCK: 0,4
OK
```

"IR" network facility is unlocked (0): 1 = voice, 2 = data, 4 = fax.

Check "OI" network facility status (Bar Outgoing (originated) International Calls).

```
AT+CLCK=OI,2
+CLCK: 0,1
+CLCK: 0,2
+CLCK: 0,4
OK
```

"OI" network facility is unlocked (0): 1 = voice, 2 = data, 4 = fax.

Assume that the module is not registered. Try to check the "OI" network facility status just to see the format response when the Extended Error result code is enabled in numeric format.

```
AT+CMEE=1

OK

AT+CLCK=OI,2

+CME ERROR: 100
```

Bar/Unbar All Incoming Calls

Use the following AT command to change the status of the AI network facility (All Incoming Calls):

```
AT+CLCK=AI,<mode>,<passwd>
```

Examples

Lock and unlock the “AI” network facility. Assume that the Network Password provided by Network Operator is 2121.

Check “AI” network facility status:

```
AT+CLCK=AI,2
+CLCK: 0,1
+CLCK: 0,2
+CLCK: 0,4
OK
```

“AI” network facility is unlocked (0): 1 = voice, 2 = data, 4 = fax.

Lock “AI” network facility:

```
AT+CLCK=AI,1,2121
OK
```

Check “AI” facilities status:

```
AT+CLCK=AI,2
+CLCK: 1,8
+CLCK: 1,4
+CLCK: 1,2
OK
```

”AI” network facility is locked (1): 8 = short message service, 4 = fax, 2 = data.

Unlock “AI” facilities:

```
AT+CLCK=AI,0,2121
OK
```

Check “AI” facilities status:

```
AT+CLCK=AI,2
+CLCK: 0,1
+CLCK: 0,2
+CLCK: 0,4
OK
```

“AI” network facility is unlocked (0): 1 = voice, 2 = data, 4 = fax.

Bar/Unbar Incoming Calls in International Roaming

Use the following AT command to change the status of the “IR” network facility (Incoming Calls when Roaming outside the home country).

```
AT+CLCK=IR,<mode>,<passwd>
```

Examples

Lock and unlock the “IR” network facility. Assume that the network password provided by Network Operator is 2121.

Check “IR” network facilities status:

```
AT+CLCK=IR,2
+CLCK: 0,1
+CLCK: 0,2
+CLCK: 0,4
OK
```

“IR” network facility is unlocked (0): 1 = voice, 2 = data, 4 = fax.

Lock “IR” network facility:

```
AT+CLCK=IR,1,2121
OK
```

Check “IR” facilities status:

```
AT+CLCK=IR,2

+CLCK: 1,1

+CLCK: 1,8

+CLCK: 1,4

+CLCK: 1,2

OK
```

“IR” network facility is locked (1): 8 = short message service, 4 = fax, 2 = data.

Unlock “IR” network facility:

```
AT+CLCK=IR,0,2121
OK
```

Read IR facilities status:

```
AT+CLCK=IR,2
+CLCK: 0,1
+CLCK: 0,2
+CLCK: 0,4
OK
```

“IR” network facility is unlocked (0): 1 = voice, 2 = data, 4 = fax.

Bar/Unbar All Outgoing Calls

Use the following AT command to change the status of the “AO” network facility (All Outgoing Calls).

```
AT+CLCK=AO,<mode>,<passwd>
```

Examples

Lock and unlock the “AO” network facility. Assume the network password provided by Network Operator is 2121.

Check “AO” network facility status:

```
AT+CLCK=AO,2
+CLCK: 0,1
+CLCK: 0,2
+CLCK: 0,4
OK
```

“AO” network facility is unlocked (0): 1 = voice, 2 = data, 4 = fax.

Lock “AO” network facility:

```
AT+CLCK=AO,1,2121
OK
```

Check “AO” network facility status:

```
AT+CLCK=AO,2
+CLCK: 1,8
+CLCK: 1,4
+CLCK: 1,2
OK
```

“AO” network facility is locked (1): 8 = short message service, 4 = fax, 2 = data.

Unlock “AO” network facility:

```
AT+CLCK=AO,0,2121
OK
```

Checking “AO” network facility status:

```
AT+CLCK=AO,2
+CLCK: 0,1
+CLCK: 0,2
+CLCK: 0,4
OK
```

“AO” network facility is unlocked (0): 1 = voice, 2 = data, 4 = fax.

Bar/Unbar All Outgoing International Calls

Use the following AT command to change the status of the “OI” network facility (Outgoing International Calls).

```
AT+CLCK=OI,<mode>,<passwd>
```

Examples

Lock and unlock the "OI" network facility. Assume the network password provided by Network Operator is 2121.

Checking "OI" network facility status:

```
AT+CLCK=OI,2
+CLCK: 0,1
+CLCK: 0,2
+CLCK: 0,4
OK
```

"OI" network facility is unlocked (0): 1 = voice, 2 = data, 4 = fax.

Lock "OI" network facility:

```
AT+CLCK=OI,1,2121
OK
```

Check "OI" network facility status:

```
AT+CLCK=OI,2
+CLCK: 1,1
+CLCK: 1,8
+CLCK: 1,4
+CLCK: 1,2
OK
```

"OI" network facility is locked (1): 1 = voice, 8 = short message service, 4 = fax, 2 = data.

Unlock "OI" network facility:

```
AT+CLCK=OI,0,2121
OK
```

Check "OI" network facility status:

```
AT+CLCK=OI,2
+CLCK: 0,1
+CLCK: 0,2
+CLCK: 0,4
OK
```

"OI" network facility is unlocked (0): 1 = voice, 2 = data, 4 = fax.

Bar/Unbar All Outgoing International Calls Except to Home Country

Use the following AT command to change the status of the "OX" network facility (Outgoing International Calls except to Home Country).

```
AT+CLCK=OX,<mode>,<passwd>
```

Examples

Lock and unlock the "OX" network facility. Assume the network password provided by Network Operator is 2121.

Check "OX" network facility status:

```
AT+CLCK=OX,2
+CLCK: 0,1
+CLCK: 0,2
+CLCK: 0,4
OK
```

"OX" network facility is unlocked (0): 1 = voice, 2 = data, 4 = fax.

Lock "OX" network facility. It is a setting not supported by the network:

```
AT+CLCK=OX,1,2121
ERROR
```

Enable extended error result codes in the verbose format:

```
AT+CMEE=2
OK
```

Try again to lock the "OX" network facility:

```
AT+CLCK=OX,1,2121
+CME ERROR: unknown
```

Check "OX" network facility status:

```
AT+CLCK=OX,2
+CLCK: 0,1
+CLCK: 0,2
+CLCK: 0,4
OK
```

"OX" network facility is unlocked (0): 1 = voice, 2 = data, 4 = fax.

Unbar All Calls

Use the following AT command to unlock the "AB" network facility (All Barring services).

```
AT+CLCK=AB,0,<passwd>
```

Examples

Unlock the "AB" network facility. Assume the Network Password provided by Network Operator is 2121.

```
AT+CLCK=AB,0,2121
OK
```

Check "IR" network facility status:

```
AT+CLCK=IR,2
+CLCK: 0,1
+CLCK: 0,2
+CLCK: 0,4
OK
```

"IR" network facility is unlocked (0): 1 = voice, 2 = data, 4 = fax.

Check "OI" network facility status:

```
AT+CLCK=OI,2
+CLCK: 0,1
+CLCK: 0,2
+CLCK: 0,4
OK
```

"OI" network facility is unlocked (0): 1 = voice, 2 = data, 4 = fax.

Check "AI" network facility status:

```
AT+CLCK=AI,2
+CLCK: 0,1
+CLCK: 0,2
+CLCK: 0,4
OK
```

"AI" network facility is unlocked (0): 1 = voice, 2 = data, 4 = fax.

DTMF Tones

DTMF Transmission on Uplink

Using the following AT command, the module sends the suitable command to the network infrastructure to generate on the other audio party the correspondent DTMF tone. The module embeds the DTMF command in a network message and sends it during the voice call.

```
AT+VTS=<dtmf>[,duration]
```

Example

Check the range of supported values:

```
AT+VTS=?
```

```
(0-9,#,*,A-D),(0,10-255)
OK
```

Check the tone duration of the single character:

```
AT+VTD?
1
OK
```

Dialing the number in voice mode:

```
ATD04x419x40y;
OK
```

Send the following sequence of tones:

```
AT+VTS=123456789
OK
```

Hang up the voice call:

```
ATH

OK
```

SMS Management

The modules provide the SMS Service to store, send, receive, and delete an SMS, which is a short text message up to 160 characters long. Before using the SMS messages, you must configure the Short Message Service.

Select SMS Format Type

The Telit Module supports two SMS formats:

1. PDU mode
2. Text mode

The module uses the PDU format to send a message on the air. The PDU mode enables the user to edit the message in PDU format. If the user is familiar with PDU encoding, he can operate with PDU by selecting that mode and using the appropriate commands.

This document uses the Text mode to explain how to operate with SMS. Here is the AT command to select the mode.

```
AT+CMGF=<mode>
```

Examples

Check the supported range of values:

```
AT+CMGF=?

+CMGF: (0,1)
```

OK

Set up Text Mode for the SMS:

AT+CMGF=1

OK

This setting is stored and remains active until the module is turned OFF.

Set Text Mode Parameters

When SMS format is Text mode, the SMS parameters that usually reside on the header of the PDU must be set apart with the +CSMP command.

AT+CSMP=<fo>,<vp>,<pid>,<dcs>

The first field <fo> is an octet in which each bit has the following meaning:

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reply Path Request	User Data Header Indicator	Status Report Request	Validity Period format		Reject Duplicates	Message Type Indicator	

Example 1

Set the SMS parameters as follow:

1. <fo> expressed in binary format, see table below. The binary number expressed in decimal format is 17.

0	0	0	1	0	0	0	1
Reply Path not requested	User Data Header Indicator	Status Report not requested	Validity Period present, relative format		Always 0	SMS-SUBMIT	

2. <vp> validity period (in relative format) = 24 hours is coded into 167 decimal format.
3. <pid> protocol identifier.
4. <dcs> data coding scheme, default value 0.

AT+CSMP= 17,167,0,0

OK

Example 2

Set the SMS parameters as follow:

1. <fo> expressed in binary format, see table below. The binary number expressed in decimal format is 25.

0	0	0	1	1	0	0	1
Reply Path not requested	User Data Header Indicator	Status Report not requested	Validity period field present in absolute format		Always 0	SMS-SUBMIT	

2. <vp> validity period in absolute format represents the expiration date of the message, for example:

   date: 29/06/02; time: 02:20; in the time zone of Italy (+1) is formatted as follows: "29/06/02,02:20:00+1"

3. <pid> protocol identifier.
4. <dcs> data coding scheme:

   - Default Alphabet
   - Class 0 (immediate display SMS)

Data coding scheme is coded in the following binary format: 11110000, corresponding to 240 in decimal format.

AT+CSMP=25,"29/06/02,02:20:00+1",0,240

OK

Note: Use dcs=0 if no particular data coding scheme is needed. Not all dcs combinations described in the 3GPP TS 23.038 are jointly supported by Networks and Telit Modules: some features may be not implemented on Networks or on Telit Modules. This mismatch generates an ERROR result code and uses different dcs.

Character Sets

Use the following AT command to select the character set:

AT+CSCS=<chset>

Here are the supported character sets:

1. "GSM" default alphabet
2. "IRA" – ITU-T.50
3. "8859-1" – ISO 8859 Latin 1
4. "PCCP437" – PC character set Code Page 437.
5. "UCS2" – 16-bit universal multiple-octet coded character set (ISO/IEC10646)

Examples

Check the supported character sets:

```
AT+CSCS=?
+CSCS: ("GSM", "IRA", "8859-1", "PCCP437", "UCS2")
OK
```

Check the current character set:

```
AT+CSCS?
+CSCS: "IRA"
OK
```

Select a non-existent character set, merely to see the response format:

```
AT+CSCS="GSA"
ERROR
```

Enabling the Error report in the verbose format:

```
AT+CMEE=2
OK
```

Select again a non-existent character set:

```
AT+CSCS="GSA"
+CME ERROR: operation not supported
```

IRA Character Set

The IRA character set is used in Text mode. IRA set defines each character as a 7-bit value: from 0x00 to 0x7F. The table below lists all the supported characters and their hexadecimal code.

Table 13: Supported IRA Character Set

		Most Significant Nibble							
		0x	1x	2x	3x	4x	5x	6x	7x
Least Significant Nibble	x0			SP1	0	@	P		p
	x1			!	1	A	Q	a	q
	x2			"	2	B	R	b	r
	x3			#	3	C	S	c	s
	x4			$	4	D	T	d	t
	x5			%	5	E	U	e	u
	x6			&	6	F	V	f	v
	x7			'	7	G	W	g	w
	x8			(	8	H	X	h	x
	x9			)	9	I	Y	i	y
	xA	LF2		*	:	J	Z	j	z
	xB			+	;	K		k	
	xC			,	<	L		l	
	xD	CR3		-	=	M		m	
	xE			.	>	N		n	
	xF			/	?	O	£	o	

1 – SP stands for space character

2 – LF stands for Line Feed character

3 – CR stands for Carriage Return character

The following examples show how to use the IRA table:

1. Get the IRA code of the character '&': the most significant nibble is 2, the least significant nibble is 6, so the IRA code for the '&' character is the hexadecimal value: 0x26.
2. Translate IRA code 0x6B into the corresponding character: the most significant nibble is 6, the least significant nibble is B, the cell at the crossing of column 6 and row B holds the character: "k".

UCS2 Character Set

The UCS2 Character Set is used in Text mode.

1. Phone number 329 05 69 6... converted into "UCS2" format: 3=0033, 2=0032, 9=0039, 0=0030, 5=0035, 6=0036, 9=0039, 6=0036 ...
2. Text HELLO converted into UCS2 format: H=0048, E=0045, L=004C, O=004F

Read SMSC Number

The module sends the SMS to the SMSC where the message is dispatched towards its final destination or is kept until the delivery is possible. To ensure the correct functioning of this service, the number of the SMSC needs to be configured on the module in accordance with the network operator used.

To know the SMSC number stored on the module, use the following AT command:

```
AT+CSCA?
```

Check the stored SMSC number:

```
AT+CSCA?

+CSCA: "+39X20XX58XX0",145

OK
```

SMSC number is compliant with the international numbering scheme.

Set SMSC Number

Use the following AT command to store a new SMSC number. The old number is overwritten.

AT+CSCA=<number>,<type>

Set up the desired SMSC number in international format:

```
AT+CSCA=+39X20XX58XX0,145
OK
```

Enable extended result code in verbose format:

```
AT+CMEE=2
OK
```

Enter the command with no parameters:

```
AT+CSCA=
+CME ERROR: operation not supported
```

Send an SMS

Use the following AT command to send an SMS.

AT+CMGS

Note: To read and set the SMSC number, refer to sections Read SMSC Number and Set SMSC Number.

Example 1

Send an SMS to the module itself and do not store it. Use the UCS2 character set.

Select Text Mode.

```
AT+CMGF=1
OK
```

Select the UCS2 character set.

```
AT+CSCS="UCS2"
OK
```

Set SMS parameters:

AT+CSMP=17,168,0,26

OK

Select how the newly received message event is notified by the DCE to the DTE.

AT+CNMI=1,1,0,0,0

OK

Send the message to the module itself. The UCS2 character set is used:

1. Phone number 329 05 69 628 is converted into "UCS2" format: 3=0033, 2=0032, 9=0039, 0=0030, 5=0035, 6=0036, 9=0039, 6=0036, 2=0032, 8=0038
2. Text CIAO is converted into UCS2 format: C=0043, I=0049, A=0041, O=004F

AT+CMGS=00330032003900300035003600390036003200 38

> 004300490041004F (close the message with Ctrl Z)

+CMGS: 81

OK

The module itself receives the SMS, the following unsolicited indication is shown on DTE:

+CMTI: "SM",3

Note: The SMS was successfully sent to the SMSC, and its network reference number is 81. Do not confuse message reference with the message index position: the first one indicates the network reference for identifying the sent message, the second one – reported by the unsolicited indication – indicates that the module receives the message, and it is stored on the position 3 of the "SM" storage.

Select the "SM" storage as indicated by the unsolicited indication.

AT+CPMS="SM"

+CPMS: 3,50,3,50,3,50

OK

Read the message from the storage position indicated by the unsolicited indication.

AT+CMGR=3

+CMGR: "REC UNREAD","002B0033003900330032003900300035003600390036003200 38",

```
"00570049004E00440020005300490004D","08/05/13,12:22:08+08"
004300490041004F
OK
```

Example 2

Send an SMS to the module itself and do not store it.

Select Text Mode

```
AT+CMGF=1
OK
```

Select how the new received message event is notified by the DCE to the DTE.

```
AT+CNMI=1,1,0,0,0
OK
```

Send the message to the module itself.

```
AT+CMGS="+39329X569YYY"
> SEND THE SMS #1 TO ITSELF (close the message with Ctrl Z)
+CMGS:  76
OK
```

The module itself receives the SMS #1; the following unsolicited indication is shown on DTE:

```
+CMTI: "SM",1
```

The SMS was successfully sent to the SMSC, and its network reference number is 76. Do not confuse message reference with message index position: the first one indicates the network reference to identify the sent message, the second one – reported by the unsolicited indication – indicates that the module has received the message and it is stored on the position 1 of the "SM" storage.

Use the unsolicited indication parameter to read SMS #1 for the first time.

```
AT+CMGR=1
+CMGR: "REC UNREAD","+39329X569YYY","WIND SIM","08/04/18,13:58:04+08"
SEND THE SMS #1 TO THE MODULE ITSELF
OK
```

Select/Check SMS Storage Type

Telit Modules can provide two types of SMS storage, in agreement with the family of belonging:

1. "SM" – SIM Card Memory
2. "ME" – Mobile Equipment Memory
3. "SR" – Status Report Message Memory.

Use the following AT command to select memory storage:

AT+CPMS=<memr>,<memw>,<mems>

The SMS are usually stored (this is true for both the originated and the received SMS) in the SM/ME storage.

The LE910Cx family allows the user to select different storage for the read-delete, write-send, and reception-saving SMS operations.

Examples

```
AT+CPMS=?                                Check the supported SMS storage types
+CPMS: (“ME”,“SM”,“SR”),(“SM”,“ME”),(“SM”,“ME”)
OK
AT+CPMS?                                 Check the current active storage type
+CPMS: “SM”,1,50,”SM”,1,50,”SM”,1,50
OK
AT+CPMS=”ME”                             Select “ME” storage type
+CPMS: 0,50,1,50,1,50
OK
AT+CPMS?                                 Check the current active storage types
+CPMS: “ME”,0,50,”SM”,1,50,”SM”,1,50    Two SMS storage types are active: “ME”
OK                                               and “SM”
```

Store an SMS

Use the following AT command to store an SMS.

AT+CMGW="<da>"

Example

Store an SMS in the "SM" storage, send it to the module itself, and read the message in the receiving storage.

AT+CMGF=1 Select Text Mode

OK

AT+CSMP=17,168,0,240 Assume to send a SMS of Class 0

OK

Select how the new received message event is notified by the DCE to the DTE

AT+CNMI=1,1,0,0,0

OK

Store into "SM" the SMS message to be sent to the module itself.

AT+CMGW="+39329X569YYY"

> SEND THE STORED SMS #1 TO THE MODULE ITSELF (close with Ctrl Z or ESC to abort)

+CMGW: 5

OK

Use index 5 to read SMS #1 from the "SM" storage type.

AT+CMGR=5

+CMGR: "STO SENT","+39329X569YYY","WIND SIM"

SEND THE STORED SMS # 1 TO MODULE ITSELF

OK

Send the stored SMS #1using the storage position 5 returned by the previous command.

AT+CMSS=5

+CMSS: 78

OK

The module itself receives the SMS #1, the following unsolicited indication is shown on DTE:

+CMTI: "SM",6

Check the current SMS storage type.

AT+CPMS?

+CPMS: "SM",6,30,"SM",6,30,"SM",6,30

OK

Use index 6 to read received SMS #1 from "SM" storage memory.

AT+CMGR=6

+CMGR: "REC UNREAD","+39329X569YYY","WIND SIM","08/04/21,09:56:38+08"

SEND THE STORED SMS # 1 TO THE MODULE ITSELF

OK

Use index 6 to read again received SMS #1 from "SM" storage memory.

AT+CMGR=6

+CMGR: "REC READ","+39329X569YYY","WIND SIM","08/04/21,09:56:38+08"

SEND THE STORED SMS # 1 TO THE MODULE ITSELF

OK

Send a Stored SMS

An SMS stored in the "SM" storage type is sent using the following AT command. Its storage location index is needed.

AT+CMSS=<index>

Example

Send the stored SMS to the module itself:

Select Text Mode

AT+CMGF=1

OK

Select “SM” storage to read SMS

AT+CPMS=”SM”

+CPMS: 1,50,1,50,1,50

OK

Read the SMS stored on position 1.

AT+CMGR=1

+CMGR: “STO SENT”,”+39329X569YYY”,”WIND SIM”

SEND THE STORED SMS # 1 TO MODULE ITSELF

OK

Select how the new received message event is indicated by the DCE to the DTE.

```
AT+CNMI=1,1,0,0,0

OK
```

Send the stored SMS # 1 message to the module itself.

```
AT+CMSS=1

+CMSS: 79

OK
```

The module itself receives the SMS #1; the following unsolicited indication is shown on DTE:

```
+CMTI: “SM”,2
```

Delete an SMS

Use the following AT command to delete an SMS stored on the “SM” storage type.

```
AT+CMGD=<index>
```

Example

Deleting an SMS stored in “SM” storage type:

```
AT+CPMS=”SM”                Select memory storage

+CPMS: 13,50,13,50,13,50

OK

AT+CMGD=?            Check the SMS

+CMGD: (1,2,3,4,5,6,7,8,9,10,11,12,13),(0-4)

OK
```

Delete SMS in memory position 1.

```
AT+CMGD=1

OK
```

Check if the SMS is deleted:

```
AT+CMGD=?

+CMGD: (2,3,4,5,6,7,8,9,10,11,12,13),(0-4)

OK
```

Delete all SMS. Disregard the first parameter of the +CMGD.

```
AT+CMGD=1,4

OK

AT+CMGD=?

+CMGD: (),(0-4)

OK
```

Read an SMS

An SMS is read with the following command:

AT+CMGR=<index>

Example

```
AT+CPMS?

+CPMS: "SM",1,50,"SM",1,50,"SM",1,50

OK
```

Read the SMS #1, for the first time, in storage memory "SM", position 1:

```
AT+CMGR=1

+CMGR: "STO SENT","+39329X569YYY","WIND SIM"

SEND THE STORED SMS # 1 TO MODULE ITSELF

OK
```

SMS Status

SMSs can be gathered into five different groups depending on their Status:

1. REC UNREAD: received messages still not read
2. REC READ: received messages already read
3. STO UNSENT: written messages not yet sent
4. STO SENT: written messages already sent
5. ALL: all types of messages

Use the following AT command to query the SMS status:

AT+CMGL=<stat>

Check if Text Mode is active

```
AT+CMGF?
+CMGF: 1                Text Mode is active
OK
```

Check the supported SMS status

```
AT+CMGL=?
+CMGL: (“REC UNREAD”, “REC READ”, “STO UNSENT”, “STO SENT”, “ALL”)
OK
```

Check the available SMS storage type

```
AT+CPMS?
+CPMS: “SM”,6,30, “SM”,6,30, “SM”,6,30
OK
```

List all the SMSs stored on “SM” storage with their Status.

```
AT+CMGL=”ALL”
+CMGL: 1,”REC READ”, •••• SMS body ••••
+CMGL: 2,”REC READ”, •••• SMS body ••••
+CMGL: 3,”REC READ”, •••• SMS body ••••
+CMGL: 4,”STO SENT”, •••• SMS body ••••
+CMGL: 5,”STO SENT”, •••• SMS body ••••
+CMGL: 6,”REC READ”, •••• SMS body ••••
OK
```

List the SMSs stored on “SM” storage with their Status=STO SENT

```
AT+CMGL=”STO SENT”
+CMGL: 4,”STO SENT”, •••• SMS body ••••
+CMGL: 5,”STO SENT”, •••• SMS body ••••
OK
```

Cell Broadcast Service

GSM Standard specifies two different types of SMS:

1. SMS Point to Point (SMS/PP),

2. SMS Cell Broadcast (SMS/CB).

The first type can send a text message up to 160 characters long from one module to another (as stated in the previous paragraphs), the second type allows the Network to send, at the same time, a message to all modules contained in the defined area including one or more radio cells. The availability and the implementation of the Cell Broadcast Service are strictly connected with the Network Operator of the subscriber.

Use the following AT command to enable the Cell Broadcast Service:

```
AT+CSCB=[<mode>[,<mids>[,<dcss>]]]
```

Select Text Mode.

```
AT+CMGF=1
OK
```

Select the District service.

```
AT+CSCB=0,50,0
OK
```

Select how the new received message event is indicated by the DCE to the DTE.

```
AT+CNMI=2,0,2,0,0
OK
```

After a while the "District" broadcast message is displayed on the DTE.

```
+CBM: 24,50,1,1,1
TRIESTE
+CBM: 4120,50,2,1,1
TRIESTE


+CBM: 8216,50,1,1,1
TRIESTE
+CBM: 12312,50,2,1,1
TRIESTE
```

Table 14: Message Identifiers

<mids>	Service name
000	Index
010	Flashes
020	Hospitals
022	Doctors
024	Pharmacy
030	Long Distant Road Reports
032	Local Road Reports
034	Taxis
040	Weather
050	District
052	Network Information
054	Operator Services
056	Directory Inquiries (national)
057	Directory Inquiries (international)
058	Customer Care (national)
059	Customer Care (international)

GNSS Management

Introduction

The LE910Cx module is equipped with IZat™ Gen 8C that can be controlled by the modem using a set of AT commands or dedicated NMEA sentences.

LE910Cx Serial Ports

LE(Linux): Three serial ports are available on the module:

1. MODEM #1 USB SERIAL PORT
2. MODEM #2 USB SERIAL PORT
3. NMEA USB SERIAL PORT

TX (ThreadX): Two serial ports are available on the module:

1. MODEM #1 USB SERIAL PORT
2. MODEM #2 USB SERIAL PORT

WGS84

The GPS receivers perform initial position and velocity calculations using an earth-centered earth-fixed (ECEF) coordinate system. The Results may be converted to an earth model (geoid) defined by the selected datum. For LE910Cx, the default datum is WGS 84 (World Geodetic System 1984) which provides a worldwide common grid system that may be translated into local coordinate systems or map dates. (Local map dates are the best fit to the local shape of the earth and are not valid worldwide)

NMEA 0183

The NMEA 0183 is a specification created by the National Marine Electronics Association (NMEA) that defines the interface between other marine electronic equipment. The standard permits marine electronics to send information to computers and other marine equipment. GPS receiver communication is defined within this specification. The supported version is 4.10.

The provided NMEA sentences are:

GGA	GPS Fix Data. Time, position, and fix type data.
GLL	Geographic Position - Latitude/Longitude
GSA	GPS receiver operating mode, satellites used in the position solution, and DOP values.
GSV	The number of GPS satellites in view satellite ID numbers, elevation, azimuth, and SNR values.
RMC	Time, date, position, course, and speed data.
VTG	Course and speed information relative to the ground
GNS	GNSS fix data.
GRS	Range residuals.
DTM	Datum reference information.

Note: The NMEA (LE (Linux)) or Modem (TX (ThreadX)) USB port provides the following sentences with the $GPSNMUN command: GGA, GLL, GSA, GSV, RMC, VTG.

The NMEA (LE (Linux)) or Modem (TX (ThreadX)) USB port provides the following sentences with the $GPSNMUNEX command: GNS, GRS, DTM.

GGA – Global Position System Fixed Data

This sentence provides time, position, and fixes related data for a GPS Receiver. Table 15 contains the values for the following example:

$GPGGA,161229.480,3723.247522, N,12158.341622, W,1,07,1.0,72.1, M,18.0, M, ,*18

Table 15: GGA Data Format

Name	Example	Units	Description
Message ID	$GPGGA		GGA protocol header GP: GPS Talker ID
UTC Time	161229.48		hhmmss.ss
Latitude	3723.247522		ddmm.mmmmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.341622		dddmm.mmmmmm
E/W Indicator	W		E=east or W=west
Position Fix Indicator	1		See Table 16
Satellites Used	07		Range 0 to 12
HDOP	1.0		Horizontal Dilution of Precision
MSL Altitude	72.1	meters	Antenna Altitude above/below mean-sea-level (geoid).
Units	M	meters	Units of antenna altitude
Geoid Separation	18.0	meters	The difference between the WGS-84 earth ellipsoid and the mean-sea-level (geoid), "-" means mean-sea-level below ellipsoid.
Units	M	meters	Units of geoidal separation
Age of Diff. Corr.		second	Null fields when DGPS is not used
Diff. Ref. Station ID			Null fields when DGPS is not used / 0000-1023
Checksum	*18		
<CR> <LF>			End of message termination

Table 16: Position Fix Indicator

Value	Description
0	Fix not available or invalid
1	GPS SPS Mode, fix valid
2	Differential GPS, SPS Mode, fix valid
3	GPS PPS Mode, fix valid
4	Real Time Kinematic
5	Float RTK
6	Estimated (dead reckoning) Mode
7	Manual Input Mode
8	Simulator Mode

GLL - Geographic Position - Latitude/Longitude

This sentence provides the latitude and longitude of vessel position, time of position fix, and status. Table 17 contains the values for the following example:

$GPGLL,3723.247522,N,12158.341622,W,161229.48,A,A*41

Table 17: GLL Data Format

Name	Example	Units	Description
Message ID	$GPGLL		GLL protocol header GP: GPS Talker ID
Latitude	3723.247522		ddmm.mmmmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.341622		dddmm.mmmmmm
E/W Indicator	W		E=east or W=west
UTC Time	161229.48		hhmmss.ss
Status	A		A=data valid or V=data not valid
Mode Indicator	A		See Table 18
Checksum	*41		
<CR> <LF>			End of message termination

Table 18: Mode Indicator

Value	Description
N	Fix not available or invalid
A	GPS SPS Mode, fix valid
D	Differential GPS, SPS Mode, fix valid
P	GPS PPS Mode, fix valid
R	Real Time Kinematic
F	Float RTK
E	Estimated (dead reckoning) Mode
M	Manual Input Mode
S	Simulator Mode

GSA - GNSS DOP and Active Satellites

This sentence reports the GPS receiver's operating mode, satellites used in the navigation solution reported by the GGA sentence and DOP values. Table 19 contains the values for the following example:

$GPGSA, A,3,07,02,26,27,09,04,15,, , , , ,1.8,1.0,1.5,1*33

Table 19: GSA Data Format

Name	Example	Units	Description
Message ID	$GPGSA		GSA protocol header GP: GPS Talker ID GN: GNSS Talker ID BD: Beidou Talker ID GA: GLILEO Talker ID
Mode 1	A		See Table 20
Mode 2	3		See Table 21
Satellite Used1. Satellite used in solution.1	07		Sv on Channel 1 GPS: 1-32 SBAS: 33-64 (offset 87) GLONASS: 65-96. GALILEO:1-36 (offset 300) BEIDOU:1-37 (offset 200)
Satellite Used1	02		Sv on Channel 2

Name	Example	Units	Description
….			
Satellite Used1			
PDOP	1.8		Position Dilution of Precision
HDOP	1.0		Horizontal Dilution of Precision.
VDOP	1.5		Vertical Dilution of Precision.
GNSS System ID	1		1=GPS 2=GLONASS 3=GALILEO 4=BEIDOU
Checksum	*33		
<CR> <LF>			End of message termination

Table 20: Mode 1

Value	Description
M	Manual—forced to operate in 2D or 3D mode
A	2D Automatic—allowed to automatically switch 2D/3D

Table 21: Mode 2

Value	Description
1	Fix not available
2	2D (<4 SVs used)
3	3D (>3 SVs used)

GSV - GNSS Satellites in View

This sentence reports the number of satellites (SV) in view, satellite ID numbers, elevation, Azimuth, and SNR value. There could be four satellites information per transmission so; if the number of satellites in view is greater, separated GSV sentences will be generated. The number of sentences being transmitted and the total to be transmitted are shown in the first 2 fields of the sentence. Table 22 contains the values for the following example:

$GPGSV,2,1,07,07,79,048,42,02,51,062,43,26,36,256,42,27,27,138,42,1*71

$GPGSV,2,2,07,09,23,313,42,04,19,159,41,15,12,041,42,1*41

Table 22: GSV Data Format

Name	Example	Units	Description
Message ID	$GPGSV		GSV protocol header GP: GPS Talker ID GL: GLONASS Talker ID BD: BEIDOU Talker ID GA: GLILEO Talker ID
Number of Messages	2		Range 1 to 3
Message Number1	1		Range 1 to 3
Satellites in View	07		
Satellite ID	07		Channel 1 GPS: 1-32 SBAS: 33-64 (offset 87) GLONASS: 65-96. GALILEO :1-36 (offset 300) BEIDOU :1-37 (offset 200)
Elevation	79	degrees	
Azimuth	048	degrees	
SNR (C/No)	42	dBHz	
....	....	....	
Satellite ID	27		Channel 4
Elevation	27	degrees	Channel 4 (Maximum 90)
Azimuth	138	degrees	Channel 4 (True, Range 0 to 359)
SNR (C/No)	42	dBHz	Range 0 to 99, null when not tracking

Signal ID	1		GPS, SBAS: 1 (L1 C/A);
			GLONASS: 1 (L1 C/A);
			GALILEO: 7(E1B/C);
			BEIDOU: 1(B1I)

Checksum	*71		
<CR> <LF>			End of message termination

RMC - Recommended Minimum Specific GNSS Data

This sentence reports Time, date, position, and course and speed data. Table 23 contains the values for the following example:

$GPRMC,161229.48,A,3723.247533,N,12158.341633,W,0.13,309.62,281118,6.1,W,A,V*1
0

Table 23: RMC Data Format

Name	Example	Units	Description
Message ID	$GPRMC		RMC protocol header GP: GPS Talker ID
UTC Time	161229.48		hhmmss.ss
Status	A		A=data valid or V=data not valid
Latitude	3723.247533		ddmm.mmmmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.341633		dddmm.mmmmmm
E/W Indicator	W		E=east or W=west
Speed Over Ground	0.13	knots	
Course Over Ground	309.62	degrees	True
Date	281118		ddmmyy
Magnetic Variation	6.1	degrees	E=east or W=west
Mag variation direction	W		E/W. E subtracts mag var from true, W adds mag var to true.
Mode Indicator	A		See Table 16
Navigational Status Indicator	V		V (equipment is not providing navigational status indication).
Checksum	*10		
<CR> <LF>			End of message termination

VTG - Course over Ground and Ground Speed

This sentence reports the actual course and speed relative to the ground.

Table 24 contains the values for the following example:

$GPVTG,309.62,T, ,M,0.13,N,0.2,K,A*23

Table 24: VTG Data Format

Name	Example	Units	Description
Message ID	$GPVTG		VTG protocol header GP: GPS Talker ID
Course	309.62		Measured heading
Reference	T		True
Course		degrees	Measured heading
Reference	M		Magnetic
Speed	0.13	knots	Measured horizontal speed
Units	N	Knots	
Speed	0.2	km/hr	Measured horizontal speed
Units	K		Kilometers per hour
Mode Indicator	A		See Table 16
Checksum	*23		
<CR> <LF>			End of message termination

GNS - GNSS Fix Data

This sentence reports the GNSS fix data.

Table 25 contains the values for the following example:

$GNGNS,084509.00,3731.283789,N,12655.755481,E,ANNN,07,1.2,110.7,18.0,,,V*26

Table 25: GNS Data Format

Name	Example	Units	Description
Message ID	$GNGNS		GNS protocol header GN: GNSS Talker ID

Name	Example	Units	Description
UTC Time	084509.00		hhmmss.ss
Latitude	3731.283789		ddmm.mmmmmm
N/S Indicator	N		N=north or S=south
Longitude	12655.755481		dddmm.mmmmmm
E/W Indicator	E		E=east or W=west
Mode Indicator	ANNN	km/hr	Fixed length field; contains four characters, The first symbol relates to GPS The second one – to GLONASS The third one – to GALILEO The fourth one – to BEIDOU See Table 16
Satellites Used	07		Number of satellites in use, (Gps+Glonass+Galileo+Beidou)
HDOP	1.2		Horizontal Dilution of Precision.
MSL Altitude	110.7	meters	Antenna Altitude above/below mean-sea-level (geoid)
Geoid Separation	18.0		The difference between the WGS-84 earth ellipsoid and the mean-sea-level (geoid), "-" means mean-sea-level below ellipsoid.
Age of Diff. Corr.		second	Null fields when DGPS is not used
Diff. Ref. Station ID			Null fields when DGPS is not used / 0000-1023
Navigational Status	V		V (Equipment is not providing navigational status indication)
Checksum	*26		
<CR> <LF>			End of message termination

GRS - Range Residuals

The sentence is used to support the Receiver Autonomous Integrity Monitoring (RAIM).

Table 26 contains the values for the following example:

$GPGRS,085634.00,0,-7.468427,-4.436067,-8.816755,-4.297600,-5.622384,-2.630362,,,,,,,1,1*49

Table 26: GRS Data Format

Name	Example	Units	Description
Message ID	$GPGRS		GRS protocol header GP: GPS Talker ID
UTC Time	085634.00		hhmmss.ss
Mode	0		Mode: 0 = residuals used to calculate the position given in the GGA or GNSmessage 1 = residuals were recomputed after the GGA or GNS message position was computed.
Range residuals	-7.468427	meters	Range residuals, in meters, for satellites used in the navigation solution. Order must match the order of satellite ID numbersthe in GSA messagthe e. When the GRS message is used, the GSA and GSV messages are generally required with this message.
....	....	....	
Range residuals		meters	
GNSS System ID	1		1=GPS 2=GLONASS 3=GALILEO 4=BEIDOU

Signal ID	1		GPS, SBAS: 1 (L1 C/A);
			GLONASS: 1 (L1 C/A);
			GALILEO: 7(E1B/C);
			BEIDOU: 1(B1I)

Checksum	*49		
<CR> <LF>			End of message termination

DTM - Datum Reference Information

The sentence is used to identify the datum used.

Table 27 contains the values for the following example:

$GPDTM,P90,,0000.000025,S,00000.000001,W,0.982,W84*57

Table 27: DTM Data Format

Name	Example	Units	Description
Message ID	$GPDTM		DTM protocol header GP: GPS Talker ID
Local datum code	P90		Local datum code W84 – WGS84 P90 – PZ90
Local datum sub code			Local datum sub code
Latitude offset	0000.000025	minutes	Latitude offset in minutes
Latitude offset mark	S		Latitude offset mark (N: +, S: -)
Longitude offset	00000.000001	minutes	Longitude offset in minutes
Longitude offset mark	W		Longitude offset mark (E: +, W: -)
Altitude offset	0.982		Altitude offset in meters.
Reference Datum Code	W84		Datum (xxx): W84 – WGS84 P90 – PZ90
Checksum	*57		
<CR> <LF>			End of message termination

Checking GNSS Device Functionality

After a proper power on, the device is ready to receive AT commands on the MODEM serial port.

When the $GPSP command is issued, The GNSS receiver also will be powered on and it will start the scan of the available GNSS signals.

LE(Linux): On the NMEA USB Serial port (default 115200 bps, 8, n, 1), there must be the presence of the NMEA sentences when the $GPSNMUN command is issued.

TX (ThreadX): On the Modem USB serial port, there must be the presence of the NMEA sentences when the $GPSNMUN command is issued.

Controlling GNSS Receiver

The LE910Cx module is provided by a set of AT commands that permits to configure and use it through the MODEM serial port.

Power Control of GNSS Receiver

The GNSS receiver is by default switched off at the first power on. It is necessary to switch it on or off is possible to use the AT$GPSP command. The GNSS receiver is usable if the module is switched on (or at least in power saving). This command also switches off the GNSS receiver supply.

Syntax of the command AT$GPSP=<status>

Where:

<status> - 0 GPS controller is powered down(default), 1 GPS controller is powered up? Returns the range of values accepted

AT$GPSP? will return the current status.

Example 1: (to switch on the GNSS)

AT$GPSP=1<CR>

OK

Example 2: (to know the status)

AT$GPSP?<CR>

The answer will be:

$GPSP: 0

OK

GNSS Reset

With the command, AT$GPSR=<reset_type> is possible to reset the GNSS module.

Parameter:

<reset_type>

0 - Factory reset: This option clears all GPS memory including clock drift. It is available in controlled mode only.

1 - Coldstart (No Almanac, No Ephemeris): This option clears all data currently stored in the internal memory of the GPS receiver including position, almanac, ephemeris, and time. However, the stored clock drift is retained. It is available in controlled mode only.

2 - Warmstart (No ephemeris): This option clears all initialization data in the GPS receiver and subsequently reloads the data that is currently displayed in the Receiver Initialization Setup screen. The almanac is retained but the ephemeris are cleared. It is available in controlled mode only.

3 - Hotstart (with stored Almanac and Ephemeris): The GPS receiver restarts by using the values stored in the internal memory of the GPS receiver; validated ephemeris and almanac. It is available in controlled mode only.

Example:

It is available in controlled mode only.

AT$GPSP=1<CR>

OK

Let's suppose to perform a cold start of the GNSS receiver.

AT$GPSR=1<cr>

OK

The Receiver will clear all the parameters in its memory, and it will start a new scanning of the available satellites.

GNSS Parameters Save

This command allows saving the set parameters in the module's memory.

Syntax of the command:

AT$GPSSAV

Restore GNSS Parameters

This command allows restoring the factory default parameters for the GNSS module.

Syntax of the command:

AT$GPSRST

After this command should restart the module to update the modifications.

Note: If the GPS controller is powered up (see $GPSP), the GPS controller is powered down because the GPS parameters should be reset with factory default.

Read Acquired GNSS Position

This command allows reading the acquired position of the GNSS receiver.

Syntax of the command:

AT$GPSACP

The response syntax is:

$GPSACP:<UTC>,<latitude>,<longitude>,<hdop>,<altitude>,<fix>,<cog>,<spkm>,<spkn>,<date>,<nsat_gps>,<nsat_glonass>

The fields contain the following information:

<UTC> - UTC time (hhmmss.sss) referred to GGA sentence

<latitude> - format is ddmm.mmmm N/S (referred to GGA sentence)

where:

dd – degrees - 00..90

mm.mmmm - minutes - 00.0000..59.9999

N/S: North / South

<longitude> - format is dddmm.mmmm E/W (referred to GGA sentence)

where:

ddd - degrees - 000..180

mm.mmmm - minutes - 00.0000..59.9999

E/W: East / West

<hdop> - x.x - Horizontal Diluition of Precision (referred to GGA sentence)

<altitude> - xxxx.x Altitude - mean-sea-level (geoid) in meters (referred to GGA sentence)

<fix> -

0 or 1 -Invalid Fix

2 - 2D fix

3 - 3D fix

<cog> - ddd.mm - Course over Ground (degrees, True) (referred to VTG sentence)

where:

ddd - degrees - 000..360

mm – minutes - 00..59

<spkm> - xxxx.x Speed over ground (Km/hr) (referred to VTG sentence)

<spkn> - xxxx.x- Speed over ground (knots) (referred to VTG sentence)

<date> - ddmmyy Date of Fix (referred to RMC sentence)

where:

dd - day - 01..31

mm – month - 01..12

yy – year - 00..99 - 2000 to 2099

<nsat_gps> - nn - Total number of GPS satellites in use (referred to GGA sentence)

- 00..12

<nsat_glonass> - nn - Total number of GLONASS satellites in use

- 00..12

Example:

```
$GPSACP: 050244.000,3731.3067N,12655.7837E,0.8,53.7,3,132.4,0.0,0.0,160320,07,05
OK
```

Blocking access to AT and USB interfaces

Note: Applicable to ThreadX versions only.

Settings to ban access to the module from all interfaces

Danger: The content of this chapter can cause a complete lock of all USB (ACM – that includes AT interfaces – ECM, Diag, RmNet) and UART interfaces. Please carefully follow the steps below to avoid the module to be unrecoverable.

The sequence of AT commands:

AT#PORTCFG=8

AT#USBCFG=7

completely bans the access to the module from USB and UART.

It is safe to run the above command sequence only from the AppZone application, and only if the same application has a way to change back at least one of the above command settings.

Danger: Never run the sequence AT#PORTCFG=8 and AT#USBCFG=7 if the module has not been previously provided with an active (running automatically from boot) AppZone application able to change back at least one between AT#USBCFG and AT#PORTCFG, otherwise the module will be unrecoverable.

Example of AppZone code

Below is an example of AppZone code locking all USB and UART interfaces

```
M2MB_RESULT_E retVal = M2MB_RESULT_SUCCESS;
M2MB_ATI_HANDLE h;
INT16 instanceID = 2;
struct myStruct myUserdata;
CHAR cmd_str[ATI_CMD_LEN];
SSIZE_T cmd_len;
retVal = m2mb_ati_init(&h, instanceID, myCallback,
(void*)myUserdata);
…
```

```
strncpy(cmd_str, "AT#PORTCFG=8", ATI_CMD_LEN);
cmd_len = strlen(cmd_str);
cmd_str[cmd_len++] = '\r';
retVal = m2mb_ati_send_cmd(h, cmd_str, cmd_len);
if ( retVal == M2MB_RESULT_SUCCESS )
    printf( "m2mb_ati_send_cmd succeeded");
strncpy(cmd_str, "AT#USBCFG=7", ATI_CMD_LEN);
cmd_len = strlen(cmd_str);
cmd_str[cmd_len++] = '\r';
retVal = m2mb_ati_send_cmd(h, cmd_str, cmd_len);
if ( retVal == M2MB_RESULT_SUCCESS )
    printf( "m2mb_ati_send_cmd succeeded");
```

Danger: It is important to add an event in AppZone source code able to set back the module in an #USBCFG different from 7 and/or a #PORTCFG different from 8, otherwise the module will be unrecoverable.

5.5 Packet Switched Data Operations

USB Tethering Connection

Dial-Up Networking

It is a legacy method to access internet service using a public switched telephone network. The DTE uses an attached modem to send and receive internet protocol packets. So, it is limited to supporting high-speed data rates over LTE technology. It is not recommended to use this method for internet access

Standard ECM/RNDIS

ECM stands for Ethernet Control Model and is an Ethernet emulation protocol defined by the USB Implementers Forum. RNDIS (Remote Network Driver Interface Specification) is a Microsoft proprietary protocol used mostly on top of USB. It provides a virtual Ethernet link to most versions of the Windows and Linux operating systems.

Most importantly, the USB is configured to support ECM or RNDIS by issuing AT#USBCFG command.

AT#USBCFG=<composition>

For ECM, <composition> is set 4.

For RNDIS, <composition> is set 1.

After executing the command, DUT reboots automatically and then Host reconfigures the USB composition accordingly.

ECM session can be established by running AT#ECM command.

AT#ECM=<Cid>,<Did>[,<UserId>,[<Pwd>,[<DhcpServerEnable>]]]

On the other hand, RNDIS session can be set up by AT#RNDIS command

AT#RNDIS=<Cid>,<Did>[,<UserId>,[<Pwd>,[<DhcpServerEnable>]]]

For more information on ECM/RNDIS control commands refer to AT command guide.

ECM and RNDIS provide a private IP address to the tethered TE (Host PC) even if the module has a network-assigned IP address and communicates with WWAN N/W using NATing.

Note: ThreadX variants do not support RNDIS.

MBIM/RmNet

MBIM is the communication class subclass specification for the Mobile broadband interface model. It is a protocol by which USB hosts and mobile broadband devices can efficiently exchange control commands and data frames. MBIM extends the Network Control Model (NCM) as a protocol between the host and USB devices, with the difference that devices transfer raw IP packets instead of packets with 802.3 headers. The Mobile Broadband Interface Model (MBIM) class driver is an inbox driver provided by Microsoft; no third-party driver is required.

RmNet is Qualcomm's proprietary mobile broadband network interface, which emulates the network interface for the connected TE and allows the module to behave as a network adapter. RmNet relies on a control interface for any control signal between TE and MS to initiate a data session on demand and send any notifications. The control interface is called QMI (Qualcomm MSM Interface).

USB needs to be configured to support these types of interfaces by issuing the AT#USBCFG command.

AT#USBCFG=<composition>

For RmNet, <composition> is set 0.

For MBIM, <composition> is set 2.

After executing the command, DUT reboots automatically and then Host reconfigures the USB composition accordingly.

There are no AT commands controlling data sessions over these N/W interfaces. Instead, the customer needs to prepare their own connection manager or open-source solution in a Linux environment.

MBIM/RmNet provides the IP address assigned by the network to the tethered TE (Host PC) and the module just plays the role of data modem without NATing

Note: ThreadX variants do not support MBIM.

Socket AT Commands

Configuring Embedded TCP/IP Stack

Use the #SCFG command to configure a socket belonging to the Multi-socket environment, the <connId> parameter identifies the socket. The Multi-socket environment provides N sockets, the N value depends on the module you are using. The configuration is saved in NVM.

The command syntax is:

```
AT#SCFG=<connId>,<cid>,<pktSz>,<maxTo>,<connTo>,<txTo>
```

Example

Check the current PDP contexts configuration.

```
AT+CGDCONT?

+CGDCONT: 1,"IP"," Access_Point_Name ","",0,0

OK
```

Check the current Multi-sockets/PDP contexts configuration.

```
AT#SCFG?

#SCFG: 1,1,300,90,600,50

#SCFG: 2,1,300,90,600,50

#SCFG: 3,1,300,90,600,50

#SCFG: 4,1,300,90,600,50

#SCFG: 5,1,300,90,600,50

#SCFG: 6,1,300,90,600,50

...

OK
```

Check if some PDP context is active. The following response shows that no PDP contexts are active.

```
AT#SGACT?

#SGACT: 1,0

OK
```

Activating PDP Context

The #SGACT command activates/deactivates one of the PDP contexts defined with the +CGDCONT command.

The command syntax is:

```
AT#SGACT=<cid>,<stat>[,<userId>,<pwd>]
```

Example

We want to activate context number one defined with +CGDCONT.

```
AT#SGACT=1,1

#SGACT: “212.195.45.65”
```

OK

Online Mode Operation

Open Socket Connection

The #SD command (Socket Dial) opens the TCP/UDP connection towards the host. If required, a DNS query is performed to resolve the IP address. To open the remote connection, the PDP context to which the <connId> is associated must be active, otherwise, the command returns an ERROR message.

The command syntax is:

AT#SD=<connId>,<txProt>,<rPort>,<IPaddr>[,<closureType>[,<lPort>[,<connMode>[,<txTime>[,<userIpType>]]]]]

Example

Open socket connection with <connId>=1 in ONLINE mode

```
AT#SD=1,0,80,"www.telit.com"

CONNECT
```

If the command is successful, we'll have a CONNECT message and socket number 1 will be connected to the Telit webserver.

From this moment the data incoming in the serial port is sent to the Internet host, while the data received from the host is serialized and flushed to the Terminal Equipment.

The +++ sequence does not close the socket, but only suspends it.

Resume Suspended Connection

Use the #SO command to resume a suspended connection.

The command syntax is:

AT#SO=<connId>

Example

```
AT#SD=1,0,80, "www.telit.com"

CONNECT

…

(+++)

OK

SRING: 1

AT#SO=1
```

CONNECT

(+++)

Close the Connection

Use the #SH command to close a socket connection. The command returns the OK message if the connection is closed.

The command syntax is:

AT#SH=<connId>

Example

Open a socket connection.

AT#SD=1,0,80,"www.telit.com"

CONNECT

...

+++

OK

Type in the #SH command to close the socket connection.

AT#SH = 1

OK

Command Mode Operation

Open Socket Connection

Use #SD command with <connMode>=1 to open a connection in COMMAND mode.

<connMode> is the last parameter in the syntax command.

AT#SD=<connId>,<txProt>,<rPort>,<IPaddr>[,<closureType>[,<lPort>[,<1>[,<txTime>[,<userIpType>]]]]]

Example

Open socket connection <connId>=1 in COMMAND mode

AT#SD=1,0,80,"www.telit.com",0,0,1

OK

Send User Data

Use the #SSEND command to send data on the connection when the module is in COMMAND mode. When the <CR> is entered to close the entering of the command, the ">" prompt appears to indicate that the command is ready to accept the data to be sent.

Enter Ctrl-Z to close the data entering and send the data. Before using the command, the socket must be opened.

The command syntax is:

```
AT#SSEND=<connId>
```

Example

Send the string “hello” on an echo socket with SRING mode set to Data amount.

```
AT#SSEND=1

> hello<CTRL-Z>

OK

SRING: 1,5
```

Use the #SSENDEXT command to include all bytes (0x00 to 0xFF) in the block of data to send. This command allows to include special characters as ESC (0x1B), Ctrl-Z (0x1A), BS (0x08) not accepted by #SSEND. The command syntax is:

```
AT#SSENDEXT=<connId>,<bytestosend>
```

Receive User Data

The module is in COMMAND mode and assumes that it has received an unsolicited SRING indicator notifying that received data are pending in the socket. Use the #SRECV command to get the pending data in the socket buffer. The syntax of the command is:

```
AT#SRECV=<connId>,<maxByte>[,<UDPInfo>]
```

Example

We receive an SRING data amount and then we extract all the five bytes pending with SRECV.

```
SRING: 1,5

AT#SRECV=1,5

#SRECV: 1,5

hello

OK
```

Close the Connection

The #SH (Socket Shutdownl) command is introduced before.

Server Mode(Socket Listen) Operation

Create Listen Socket

Use the #SL (Socket Listening) command to open a socket in listening mode for an incoming TCP connection.

When a remote host tries to connect, the module sends to the DTE the +SRING: <connId> unsolicited indication. The user can accept (#SA) or refuse (#SH) the incoming connection.

The syntax of the command is:

```
AT#SL=<connId>,<listenState>,<listenPort>[,<lingerT>]
```

Example)Open the <connId> = 1 socket in listening mode on <port> = 6543.

```
AT#SL=1,1,6543

OK
```

Accept Incoming Connection

Use the #SA command without the <connMode> parameter to accept the incoming connection, notified by the SRING unsolicited indication, in ONLINE mode.

The command syntax is:

```
AT#SA=<connId>[,<connMode>]
```

Example

Now, if a remote host tries to connect, the module receives an SRING unsolicited indication with the listening <connId>:

```
SRING: 1
```

Accept the incoming connection <connId>=1 in ONLINE mode.

```
AT#SA=1

CONNECT

... exchange data ...
```

The module is in ONLINE mode, the connection is established, and the two hosts can exchange data. With the escape sequence (+++) the connection can be suspended, and the module returns to COMMAND mode.

Command Mode Operation

Use the #SA command with <connMode>=1 to accept the incoming connection in COMMAND mode.

The command syntax is:

```
AT#SA=<connId>[,<connMode>]
```

Example

Now, if a remote host tries to connect, the module receives an SRING unsolicited indication with the listening <connId>:

```
SRING: 1
```

Accept the incoming connection <connId>=1 in ONLINE mode.

```
AT#SA=1,1

OK
```

SSL AT Commands

TLS and its predecessor SSL are cryptographic protocols used over the Internet to provide secure data communication in several applications.

For TLS protocol, see standards:

- RFC 2246 - TLS Protocol Version 1.0
- RFC 4346 - TLS Protocol Version 1.1
- RFC 5246 - TLS Protocol Version 1.2

Configuration

To provide communication security over a channel, enable an SSL socket using the #SSLEN command.

```
AT#SSLEN= <SSId>,<Enable>
```

Ex) Enable SSL functionality,

```
AT#SSLEN=1,1

OK
```

Use the following command to select the protocol.

```
AT#SSLSECCFG2=<SSId>,<version>
[,<unused_A>[,<unused_B>[,<unused_C>[,<unused_D>]]]]
```

Ex) Select TLS1.2 protocol,

```
AT#SSLSECCFG2=1,2
```

OK

The cipher suite is the set of algorithms used to negotiate the security settings for a network connection using the SSL/TLS network protocol.

AT#SSLSECCFG=<SSId>,<CipherSuite>,<auth_mode>[,<cert_format>]

Ex) Set SSL configuration,

AT#SSLSECCFG=1,0,1,1

OK

The following types of security data can be stored in the modules:

- Certificates
- CA Certificates
- Private Key

AT#SSLSECDATA=<SSId>,<Action>,<DataType>[,<Size>]

Ex) Store CA certificate,

AT#SSLSECDATA=1,1,1,<size>

> -----BEGIN CERTIFICATE-----<LF>

[...]

-----END CERTIFICATE-----<LF>

<ctrl>Z

OK

Use the following command to store CA Certificates more than 1 in the modules:

AT#SSLSECDATAEXT=<SSId>,<Action>,<DataType>,<Index>[,<Size>]

It can be saved up to 3 CA Certificates.

Ex) Store CA Certificate to <Index> 1,

AT#SSLSECDATAEXT=1,1,1,1,<size>

> -----BEGIN CERTIFICATE-----<LF>

[...]

-----END CERTIFICATE-----<LF>

<ctrl>Z

OK

Ex) Select CA Certificate to use for H/S

AT#SSLSECDATAEXT=1,3,1,1

OK

Use the following command to configure the SSL socket, before opening it.

AT#SSLCFG=<SSId>,<cid>,<pktSz>,<maxTo>,<defTo>,<txTo>[<sslSRingMode>[<noCarrier Mode>[,<skipHostMismatch >[,<UNUSED_4>]]]]

Ex) Set SSL configuration,

AT#SSLCFG=1,1,300,90,100,50,0,0,1,0

OK

For more information refer to AT Commands Reference Guide.

Online Mode Operation

Open Secure Socket Connection

Use the following command to open an SSL socket.

AT#SSLD=<SSId>,<rPort>,<IPAddress>,<ClosureType>[,<connMode>[,<Timeout>]]

Ex) Open the SSL socket in ONLINE mode,

AT#SSLD=1,443,"123.124.125.126",0,0

CONNECT

... [Bidirectional data exchange] ...

+++   [suspend the connection]

OK

If successful, the CONNECT message is returned, and from this moment all bytes sent to the serial port are forwarded to the remote server.

It is possible to suspend the connection, without closing it, by sending the escape sequence (+++). After that, the module returns the OK response and can parse the AT commands again.

ONLINE mode can be restored at any time by sending the following command.

AT#SSLO=<SSId>

Ex) Restore from Command Mode to Online Mode,

AT#SSLO=1

CONNECT

Exchanging User Data

Refer to the opening of the socket connection in online mode for AT command.

Open the SSL socket and wait for the CONNECT message. After receiving the CONNECT message, you can send/receive data to the module. The Data is encrypted and sent to the server through the secure socket as soon as the packet size has been reached or the <txTo> timeout expires

In ONLINE mode, AT commands cannot be entered on the serial port used.

Close the Connection

The following command closes the SSL socket.

AT#SSLH=<SSId>[,<ClosureType>]

If the secure socket has been opened in ONLINE mode, the user needs to send the escape sequence (+++) before closing it with the #SSLH command, unless the communication is closed remotely, or the idle inactivity timeout expires (NO CARRIER message).

If the secure socket has been opened in COMMAND mode when communication is closed remotely and all data has been retrieved (#SSLRECV), you can also close on the client-side and NO CARRIER message is displayed. At any moment, it is also possible to close the secure socket on the client-side using #SSLH.

Ex) Close SSL connection,

AT#SSLH=1

OK

Command Mode Operation

Open Secure Socket Connection

In COMMAND mode, data can be exchanged through an SSL socket using the #SSLSEND, #SSLSENDEXT and #SSLRECV commands.

Use the following command to open an SSL socket

AT#SSLD=<SSId>,<rPort>,<IPAddress>,<ClosureType>[,<connMode>[,<Timeout>]]

Ex) Open with Command Mode,

AT#SSLD=1,server_port,"server_address",0,1,100

OK

Use the following command to query the status of the SSL socket

AT#SSLS=<SSId>

Ex) Query the status of the Secure Socket Id = 1,

AT#SSLS=1

```
#SSLS: 1,2,<cipher_suite>

OK
```

Use the following command to get information about SSL socket data traffic

```
AT#SSLI[=<SSId>]
```

Ex) Get SSL socket information

```
AT#SSLD=1,server_port,"server_address",0,1

OK

SSLSRING: 1,16384

AT#SSLI=1

#SSLI: 1,0,0,16384,0

OK
```

Send User Data

Use one of the following commands to send data:

```
AT#SSLSEND=<SSId>[,< Timeout >]
```

When the command is closed with a <CR>, the '>' prompt is displayed. Now you can enter the data to be sent. To close the data block, enter <ctrl>Z, then the data are forwarded to the remote server through the secure socket. Response: OK on success, ERROR on failure.

Ex) Send data,

```
AT#SSLSEND=1,100

> (send data)

<ctrl>Z

OK
```

```
AT#SSLSENDEXT=<SSId>,<bytestosend>[,<Timeout>]
```

When the command is closed with <CR>, the '>' prompt is displayed. Now you can enter the data to be sent. When <bytestosend> bytes have been sent, the operation is automatically completed. Response: OK on success, ERROR on failure.

Ex) Send data,

```
AT#SSLSENDEXT=1,100,100

> (send data)

OK
```

Receive User Data

Data can be received in two different ways:

1. Using the #SSLRECV command (the "standard" way),
2. Reading data from the SSLSRING: unsolicited message.

Use the following command to receive data.

AT#SSLRECV=<SSId>,<MaxNumByte>[,<TimeOut>]

On success, the data are displayed in the following format:

#SSLRECV: <numBytesRead>

... received data ....

OK

If the timeout expires, the module displays the following response

#SSLRECV: 0

TIMEOUT

OK

The ERROR message appears on failure.

The SSLSRING: unsolicited message, if enabled, notifies the user of any new incoming data.

Configuring <sslSRingMode>=2 using the #SSLCFG command data is displayed in the URC in this format:

SSLSRING:<SSId>,<dataLen>,<data>

Close the Connection

Refer to closure operation on online mode.

HTTP AT Commands

Configuration

HTTP parameters should be configured using the following AT command before running HTTP operations.

AT#HTTPCFG=<prof_id>,<server_address>,<server_port>,<auth_type>,<username>,<password>,<ssl_enabled>,<timeout>,<cid>,<pkt_size>

Ex) Set HTTP configuration,

AT#HTTPCFG=0,server_address,server_port,1,username,password,0,120,1,1

OK

Query

Using the following command, it is possible to send an HTTP GET, HEAD, or DELETE operation to the HTTP server.

AT#HTTPQRY=<prof_id>,<command>,<resource>,<extra_header_line>

Ex) Query,

```
AT#HTTPQRY=0,0,/MOTDATA_N_LONG.txt
OK
AT#HTTPQRY=0,0,"/client_info.php","user-agent:LE910Cx"
OK
```

If sending is done successfully, the response is OK. Otherwise, an error code is reported.

When the HTTP server answer is received, this product sends the following URC through a serial interface (USB or UART)

#HTTPRING:<prof_id>,<http_status_code>,<content_type>,<data_size>

Request

Using the following command, it is possible to send an HTTP POST or PUT operation to the HTTP server.

AT#HTTPSND=<prof_id>,<command>,<resource>,<data_len>,<post_param>,<extra_header_line>

If sending is done successfully, the response is OK. Otherwise, an error code is reported.

When the HTTP server answer is received, the module sent the URC(refer to #HTTPRING URC)

> **Note:** When using the AT#HTTPSND command, the HTTP header always includes the "Connection: close" line and cannot be removed.

Ex) Send data,

```
AT#HTTPSND=1,0,"/upload/110_01.txt",110
>>> (send data)
OK
```

AT#HTTPSND=1,0,"/cgi-bin/upload.php",343,"3:boundary=----WebKitFormBoundarylejjShOaaqpRD1dO"

>>> (send data)

OK

Receive

When being notified with #HTTPRING URC, it is possible to read data from the HTTP server using the AT#HTTPRCV command.

AT#HTTPRCV=<prof_id>,<maxByte>

Ex) Receive data,

AT#HTTPRCV=0,0

….

OK

FTP AT Commands

Configuration

FTP parameters can be configured using the following AT command. Before running FTP, it is required that customer changes FTP parameters if it is needed.

AT#FTPCFG=<tout>,<IPPignoring>,<FTPSEn>,<FTPext>

AT#FTPTO=[=<tout>]

Ex)

Configure FTP parameters with non-security mode

AT#FTPCFG=500,0,0,1

OK

AT#FTPTO=1000

OK

or

Configure FTP parameters with security mode

AT#FTPCFG=500,0,1,1

OK

AT#FTPTO=1000

OK

Open Connection

To take advantage of FTP capability, it is required that the FTP connection has to be opened to the server first. Afterward, FTP commands could be available on the connected FTP channel.

Ex)

AT#FTPOPEN=<server:port>,<username>,<password>,<mode>

If it is failed to establish FTP connection within Time-out set by AT#FTPCFG, #FTPOPEN command stops and return error message to DTE.

Ex)Open FTP connection in active mode

AT#SGACT=1,1

#SGACT: 174,156,82,131

OK

AT#FTPOPEN="server","username","password",0

OK

Online Mode Operation

Uploading

This example shows how to upload a file to an FTP server when the module is in ONLINE mode. A terminal emulator is connected to the module.

Ex)

AT#SGACT=1,1

#SGACT: 193.199.234.255

OK

Set the FTP time-out.

AT#FTPTO=1000

OK

FTP connection open and in active mode.

AT#FTPOPEN="server","username","password",0

OK

The following #FTPPUT command opens the data connection, and the module enters ONLINE mode. "filename.txt" is the name of the file where the data will be stored on the FTP server. If the file you are sending is a text file, the extension must be .txt.

```
AT#FTPPUT="filename.txt",0

CONNECT


··· type in the data to write in the filename.txt file stored on the FTP server ···


+++

NOCARRIER
```

Close FTP control connection.

```
AT#FTPCLOSE

OK
```

Deactivate the PDP context.

```
AT#SGACT=1,0

OK
```

Appending

If the file must be appended, use the AT#FTPAPP command (with <connMode> = 0) instead of #FTPPUT.

Except for changing from #FTPPUT to #FTPAPP, the procedures are the same as "Uploading"

Downloading

The procedure up to #FTPOPEN is the same as in "Uploading".

And then,

Ex)

Check the working directory.

```
AT#FTPPWD

#FTPPWD: 257 "/"

OK
```

Use the #FTPLIST command to get the list of files on the working directory

Use the #FTPGET command to open e data connection, and download a file from the FTP server.

```
AT#FTPGET="filename.txt"

CONNECT


… the content of the file appears on the terminal emulator …


NO CARRIER
```

Close FTP control connection.

```
AT#FTPCLOSE

OK
```

Command Mode Operation

Uploading

This example shows how to upload data toward an FTP server when the module is in COMMAND mode. A terminal emulator is connected to the module.

The procedure before #FTPOPEN is the same as "Uploading"

Ex)

```
AT#FTPOPEN="server","username","password",1

OK
```

The #FTPPUT command opens the data connection, and the module remains in COMMAND mode. filename.txt is the file name where the data will be stored on the FTP server.

```
AT#FTPPUT="filename.txt",1

OK
```

Enter the #FTPAPPEXT command to upload data. After entering <CR>, the command returns the ">" prompt. Now, enter the data to be sent to the FTP server. As soon as <bytestosend> bytes are written, data are sent to the FTP server, and the #FTPAPPEXT message is returned.

```
AT#FTPAPPEXT=bytestosend

>… type in data…

#FTPAPPEXT: <SentBytes>
```

OK

Use again the #FTPAPPEXT=bytestosend command to send a new data chunk.

To send the last data chunk and close the FTP connection, use the following: AT#FTPAPPEXT=bytestosend,1

Appending

If the file must be appended, use the AT#FTPAPP command (with <connMode> = 1) instead of #FTPPUT.

Except for changing from #FTPPUT to #FTPAPP, the procedure before and after is the same as "Uploading"

Downloading

Let’s assume the FTP connection is open. Use the #FTPGETPKT command to open a data connection and download a file from the FTP server. The data is buffered on the socket; the module remains in COMMAND mode.

Ex)

AT#FTPGETPKT="filename.txt"

OK

The following command reports the number of bytes buffered on the socket.

AT#FTPRECV?

#FTPRECV: 600

OK

Read the first 400 bytes of the available buffered data.

AT#FTPRECV=400

#FTPRECV: 400

Text row number 1 * 1111111111111111111111111 *

...

Text row number 8 * 88888888888888888888

OK

Read 200 bytes – if available – starting from the position + 1 of the last byte read with the previous #FTPRECV command.

AT#FTPRECV =200

#FTPRECV: 200

88888 *

Text row number 9 * 9999999999999999999999999 *

...

Text row number 12 * CCCCCCCCCCCCCCCC

OK

The #FTPGETPKT? read command reports the download state.

AT#FTPGETPKT?

#FTPGETPKT: filename.txt,0,1

OK

The first parameter is the file name, the second indicates text or hexadecimal mode. The third parameter indicates <EOF> (End of File): 0 file transfer is in progress; 1 file transfer is ended.

Close the Connection

If the FTP connection is no longer needed, it can be disconnected using the following command:

AT#FTPCLOSE

OK

Email AT Commands

Configuration

It is required that the customer configures this module with Email parameters before sending email data using this product. Below is the sequence of the AT commands required in the configuration of the e-mail parameters

Configure the SMTP server address used for Email sending

AT#ESMTP=<smtp>

Configure a sender address to be seen as the originator of this email

AT#EADDR=<e-addr>

Configure email user id authenticated by the SMTP server

AT#EUSER=<e-user>

Configure an email password authenticated by the SMTP server

AT#EPASSW=<e-pwd>

If needed, save those parameters into the module permanently

AT#ESAV

Sending an Email

With the assumption of the configured e-mail parameters and PDP cid 1 activated by #SGACT, it is ready to send an email to a specified receiver.

AT#EMAILD=<da>,<subj>

The device responds to the command with the prompt '>' and waits for the message body text. To complete the operation, send Ctrl-Z char (0x1A hex); to exit without writing the message, send the ESC character (0x1B hex)

Ex)

Send the e-mail to the recipient with recipient_address.

AT#EMAILD="recipient_address","email subject"

> Hello<Ctrl-Z>

OK

IoT Platform AT Commands

The LE910Cx module has been updated to include native support for interfacing applications to the M2M Service. These APIs allow you to use a new set of AT commands to easily expand the capabilities of any device built upon a LE910Cx module.

Here is a basic interaction diagram

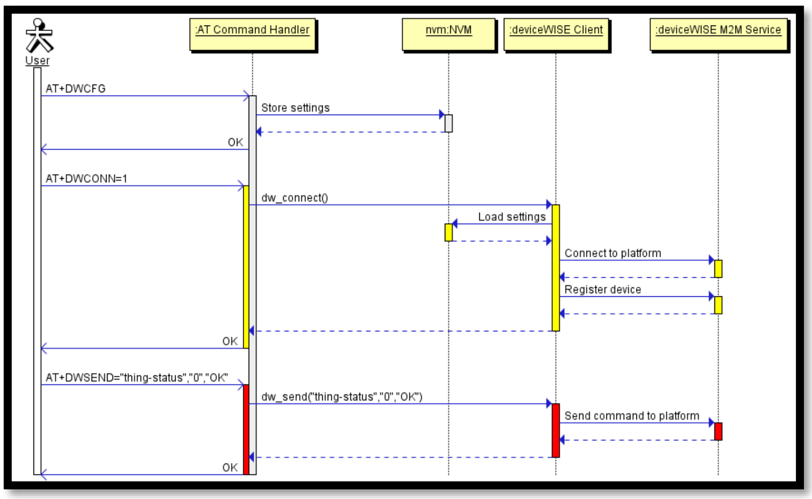
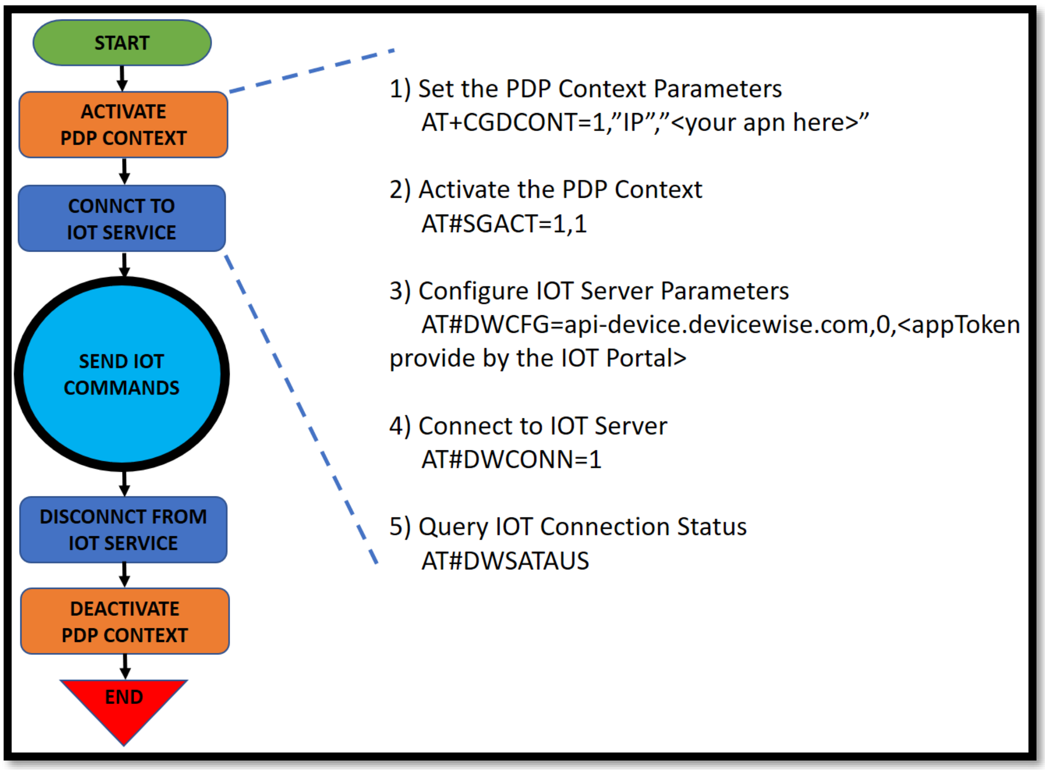


Figure 5: Basic Interaction Diagram of IoT Platform

Connect to IoT Service



1) Set the PDP Context Parameters
   AT+CGDCONT=1,"IP","<your apn here>"

2) Activate the PDP Context
   AT#SGACT=1,1

3) Configure IOT Server Parameters
   AT#DWCFG=api-device.devicewise.com,0,<appToken provide by the IOT Portal>

4) Connect to IOT Server
   AT#DWCONN=1

5) Query IOT Connection Status
   AT#DWSATAUS

Figure 6: Connect to IoT Service

API and AT Commands

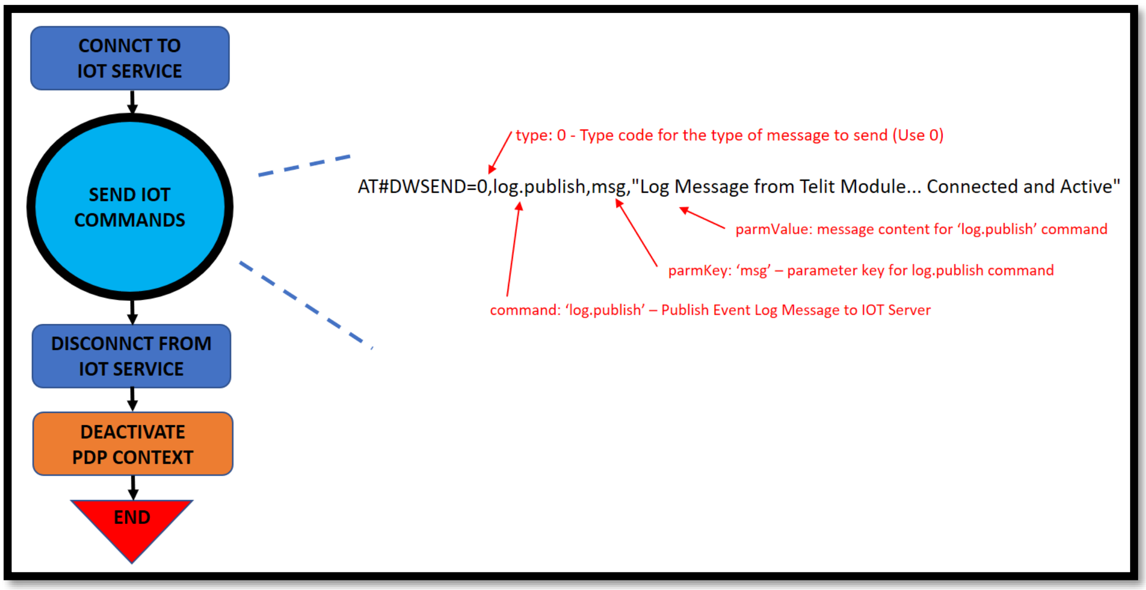
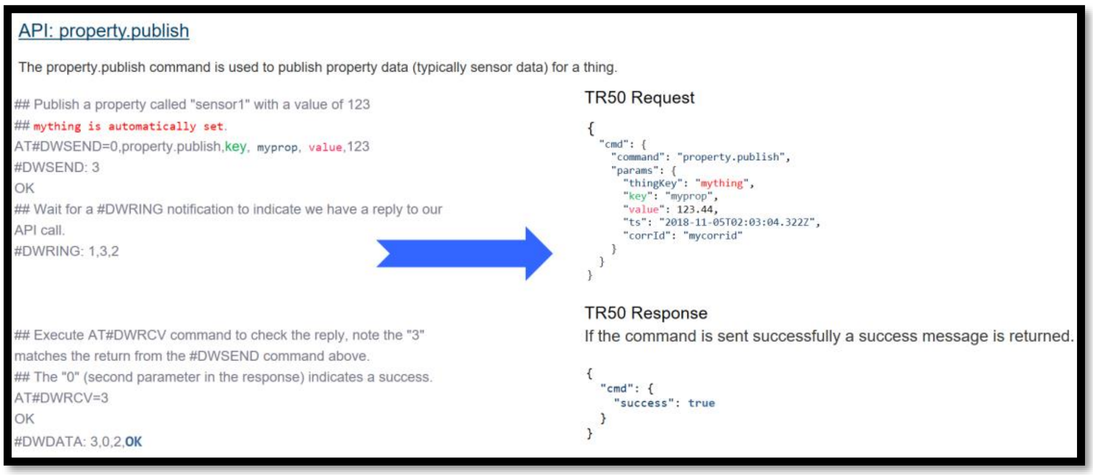

CONNCT TO IOT SERVICE
SEND IOT COMMANDS
DISCONNCT FROM IOT SERVICE
DEACTIVATE PDP CONTEXT
END
type: 0 - Type code for the type of message to send (Use 0)
AT#DWSEND=0,log.publish,msg,"Log Message from Telit Module... Connected and Active"
parmValue: message content for 'log.publish' command
parmKey: 'msg' – parameter key for log.publish command
command: 'log.publish' – Publish Event Log Message to IOT Server

Figure 7: API and AT Commands



API: property.publish

The property.publish command is used to publish property data (typically sensor data) for a thing.

```
## Publish a property called "sensor1" with a value of 123
## mything is automatically set.
AT#DWSEND=0,property.publish,key, myprop, value,123
#DWSEND: 3
OK
## Wait for a #DWRING notification to indicate we have a reply to our
API call.
#DWRING: 1,3,2

## Execute AT#DWRCV command to check the reply, note the "3"
matches the return from the #DWSEND command above.
## The "0" (second parameter in the response) indicates a success.
AT#DWRCV=3
OK
#DWDATA: 3,0,2,OK
```

TR50 Request

```
{
  "cmd": {
    "command": "property.publish",
    "params": {
      "thingKey": "mything",
      "key": "myprop",
      "value": 123.44,
      "ts": "2018-11-05T02:03:04.322Z",
      "corrId": "mycorrid"
    }
  }
}
```

TR50 Response

If the command is sent successfully a success message is returned.

```
{
  "cmd": {
    "success": true
  }
}
```

Figure 8: The property publish command

Disconnect from IoT Service

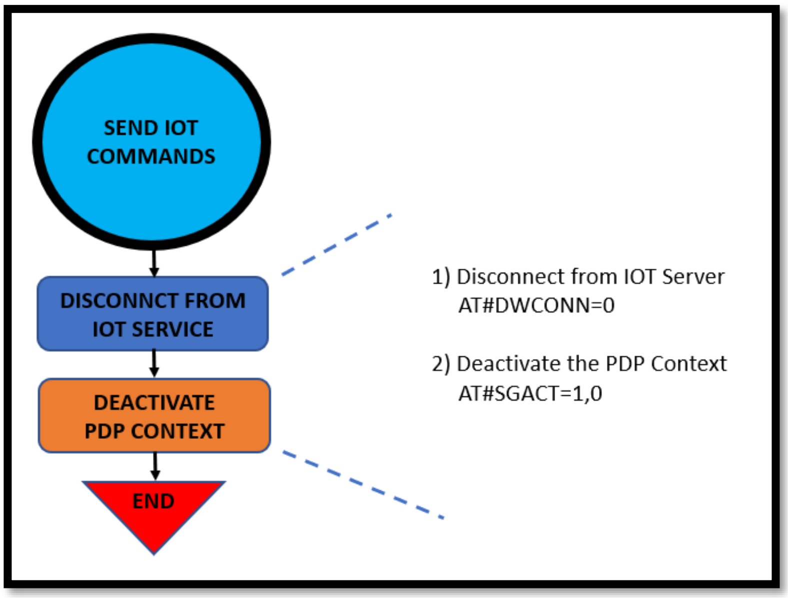

SEND IOT COMMANDS
DISCONNCT FROM IOT SERVICE
DEACTIVATE PDP CONTEXT
END
1) Disconnect from IOT Server AT#DWCONN=0
2) Deactivate the PDP Context AT#SGACT=1,0

Figure 9: Disconnect from IoT Service

IoT Platform AT Commands List

AT#DWCFG – configure deviceWISE parameters

AT#DWCONN – connect to deviceWISE server

AT#DWSTATUS – query connection status

AT#DWSEND – send data to deviceWISE server

AT#DWSENDR – send raw data to deviceWISE server

AT#DWRCV – receive data from deviceWISE server

AT#DWRCVR – receive raw data from deviceWISE server

AT#DWLRCV – List information on messages pending from the deviceWISE server

AT#DWEN - Set command permits to enable/disable up to 8 different deviceWISE features.

Note: For more information refer to the AT Commands Reference Guide.

Data Concurrency

There are 2 kinds of data calls. One is tethered call from the host side and the other is a module embedded socket call. Some concurrencies can be supported and the others not depending on how to share PDN connection. In some cases, they may be supported through different multiple PDNs and not through the same PDN.

The Module has 2 processors. One is a modem processor which is usually responsible for wireless processing and the other is an application processor running based on a Linux environment. The application can run over one of both processors. Nearly all the AT commands required to access to Socket connection are on top of the modem processor which is named Modem application in the table. On the other hand, the AP application is indicated as the application running on the AP processor. The below tables describe how to support data concurrency between Embedded and Tethered call with the Same PDN or Different PDNs.

LE (Linux)

Same PDN

In this case, each data call shares the same PDN

Table 28: LE - Data Concurrency on the same PDN with Multiple Applications

Domains of App		Modem Application	AP Application	USB Tethering		
				RmNet	RNDIS	DUN
USB Tethering	RmNet	Not Support	Not Support	N/A	N/A*	Not Support
	RNDIS	Not Support	Support	N/A*	N/A	Not Support
	DUN	Not Support	Not Support	Not Support	Not Support	N/A
Modem Application		Support	Not Support	Not Support	Not Support	Not Support
AP Application		Not Support	Support	Not Support	Support	Not Support

Different PDNs

In this case, each data call takes up a different PDN without sharing.

Table 29: LE -Data concurrency on the Different PDNs with Multiple Applications

Domains of App		Modem Application	AP Application	USB Tethering		
				RmNet	RNDIS	DUN
USB Tethering	RmNet	Support	Support	Support**	N/A*	Support***
	RNDIS	Support	Support	N/A*	Not Support	Support***
	DUN	Support	Support	Support***	Support***	N/A
Modem Application		Support	Support	Support	Support	Support
AP Application		Support	Support	Support	Support	Support

Note: * RmNet and RNDIS cannot be configured at the same time. Each one is configured by the AT#USBCFG command

** This product supports multiple PDNs over RmNet I/F using multiple logical interfaces named QMAP. The Host side must have a connection manager that supports the QMAP protocol.

*** Two N/W interfaces will be activated, but the Operating System over the Host side should handle the multiple N/W interfaces.

TX (ThreadX)

Same PDN

In this case, each data call shares the same PDN.

Table 30: TX - Data Concurrency on the Same PDN with Multiple Applications

Single PDN shared by both applications						
Domains of App		Modem Application	AP Application	USB Tethering		
				RmNet	RNDIS	DUN
USB Tethering	RmNet	Support	Not Support	N/A	N/A*	Not Support
	RNDIS	N/A*	N/A*	N/A*	N/A*	N/A*
	DUN	Not Support	Not Support	Not Support	N/A*	N/A
Modem Application		Support	Support	Support	N/A*	Not Support
AP Application		Support	Support	Not Support	N/A*	Not Support

Different PDNs

In this case, each data call takes up a different PDN without sharing.

Table 31: TX- Data Concurrency on the Different PDNs with Multiple Applications

Different PDNs for both applications						
Domains of App		Modem Application	AP Application	USB Tethering		
				RmNet	RNDIS	DUN
USB Tethering	RmNet	Support	Support	Support**	N/A*	Support***
	RNDIS	N/A*	N/A*	N/A*	N/A*	N/A*
	DUN	Support	Support	Support***	N/A*	N/A
Modem Application		Support	Support	Support	N/A*	Support
AP Application		Support	Support	Support	N/A*	Support

Note: * RNDIS is not supported

** Same as LE description

*** Same as LE description.

Maximum Number of PDN Contexts

As for the maximum number of PDN contexts that can be activated at the same time, it depends on the maximum number of bearers.

Maximum number of bearers supported = 8 (default + dedicated + emergency)

Maximum number of dedicated bearers supported = 6 (at least on default bearer and reserved one emergency bearer)

That is, the maximum number of PDN contexts activated simultaneously is 7 if no dedicated bearers are activated with one emergency bearer being reserved.

In case of activation of several dedicated bearers, the maximum number of PDN is decreased by as many as there are open dedicated bearers.

6 Performance Measurements (Linux)

6.1 Interrupt Latencies

Interrupt latency is considered in this example using the time from wakeup to the moment the context switch occurred.

The results were obtained on a performance build. The First table shows the results when the CPU governor is set to interactive (default settings) and the second table is while setting the governor to performance.

Table 32: Results for Interactive Governor

Task	Runtime ms	Switches	Average delay ms	Maximum delay ms	Maximum delay at
hwrng:50	0.071 ms	1	avg: 6.756 ms	max: 6.756 ms	max at: 4760.939225 s
kworker/u2:3:22	0.428 ms	2	avg: 0.131 ms	max: 0.159 ms	max at: 4768.610426 s
init:1	0.787 ms	2	avg: 0.121 ms	max: 0.160 ms	max at: 4765.410387 s
kworker/u2:7:903	0.705 ms	3	avg: 0.100 ms	max: 0.140 ms	max at: 4765.470639 s
msm_watchdog:13	0.000 ms	1	avg: 0.088 ms	max: 0.088 ms	max at: 4769.700320 s
qmi_linux_clnt:1168	0.245 ms	1	avg: 0.084 ms	max: 0.084 ms	max at: 4767.834634 s
kworker/0:13:2036	3.350 ms	15	avg: 0.079 ms	max: 0.254 ms	max at: 4769.850374 s
fota_redbend:1808	1.096 ms	4	avg: 0.075 ms	max: 0.111 ms	max at: 4764.300343 s
diagrebootapp:1543	2.303 ms	10	avg: 0.074 ms	max: 0.094 ms	max at: 4768.601101 s
dlt-deamon:1167	3.987 ms	11	avg: 0.073 ms	max: 0.105 ms	max at: 4767.550464 s
rcu_preempt:7	0.689 ms	6	avg: 0.065 ms	max: 0.092 ms	max at: 4760.960225 s

Task	Runtime ms	Switches	Average delay ms	Maximum delay ms	Maximum delay at
ubifs_bgt0_4:362	2.176 ms	9	avg: 0.059 ms	max: 0.111 ms	max at: 4765.460356 s
ksoftirqd/0:3	0.207 ms	3	avg: 0.058 ms	max: 0.080 ms	max at: 4767.550416 s
sleep:2046	8.904 ms	2	avg: 0.049 ms	max: 0.078 ms	max at: 4770.939396 s
cfinteractive:133	0.000 ms	2	avg: 0.029 ms	max: 0.032 ms	max at: 4760.990169 s
perf:2045	1.406 ms	1	avg: 0.029 ms	max: 0.029 ms	max at: 4770.940515 s
QCMAP_Connectio:817	0.799 ms	4	avg: 0.024 ms	max: 0.033 ms	max at: 4767.520488 s
TOTAL	27.152 ms	77			

Table 33: Results for Performance Governor

Task	Runtime ms	Switches	Average delay ms	Maximum delay ms	Maximum delay at
hwrng:50	0.038 ms	1	avg: 0.492 ms	max: 0.492 ms	max at: 4845.200305 s
QCMAP_Connectio:817	0.086 ms	1	avg: 0.170 ms	max: 0.170 ms	max at: 4849.760414 s
rcu_preempt:7	0.393 ms	4	avg: 0.107 ms	max: 0.260 ms	max at: 4845.220644 s
dnsmasq:1518	0.205 ms	1	avg: 0.071 ms	max: 0.071 ms	max at: 4849.760289 s
ksoftirqd/0:3	0.519 ms	4	avg: 0.065 ms	max: 0.111 ms	max at: 4845.200338 s
diagrebootapp:1543	1.479 ms	10	avg: 0.058 ms	max: 0.094 ms	max at: 4845.620221 s
init:1	0.409 ms	2	avg: 0.056 ms	max: 0.062 ms	max at: 4845.490228 s

Task	Runtime ms	Switches	Average delay ms	Maximum delay ms	Maximum delay at
kworker/u2:3:22	0.201 ms	2	avg: 0.053 ms	max: 0.054 ms	max at: 4852.820223 s
fota_redbend:1808	0.648 ms	4	avg: 0.052 ms	max: 0.062 ms	max at: 4845.370228 s
qmi_linux_clnt:1168	0.137 ms	1	avg: 0.050 ms	max: 0.050 ms	max at: 4847.836400 s
dlt-deamon:1167	1.594 ms	10	avg: 0.049 ms	max: 0.049 ms	max at: 4854.549941 s
msm_watchdog:13	0.000 ms	1	avg: 0.044 ms	max: 0.044 ms	max at: 4849.860205 s
kworker/u2:7:903	0.280 ms	3	avg: 0.043 ms	max: 0.045 ms	max at: 4854.760232 s
sleep:2049	8.432 ms	2	avg: 0.037 ms	max: 0.049 ms	max at: 4855.207040 s
kworker/0:13:2036	1.339 ms	13	avg: 0.034 ms	max: 0.050 ms	max at: 4849.943542 s
ubifs_bgt0_4:362	1.314 ms	9	avg: 0.033 ms	max: 0.063 ms	max at: 4849.740229 s
perf:2048	8.859 ms	1	avg: 0.017 ms	max: 0.017 ms	max at: 4855.207575 s
TOTAL	25.933 ms	69			

6.2 Memory Bandwidth and Latencies

The test was carried out using the tiny utility membench which tries to measure the maximum bandwidth of sequential memory accesses and the latency of random memory accesses. Bandwidth is measured by running different assembly codes for the aligned memory blocks and attempting different prefetch strategies.

Bandwidth tests:

Results were obtained on a performance build.

Note: 1MB = 1000000 bytes.

Note: Results for 'copy' tests show how many bytes can be copied per second (adding together read and written bytes would have provided numbers twice as high).

Note: 2-pass copy means we are using a small temporary buffer to first fetch data into it, and only then write it to the destination (source -> L1 cache, L1 cache -> destination).

Table 34: Memory Bandwidth Test Results

Test item	Bandwidth
C copy backwards	278.3 MB/s
C copy backwards (32 byte blocks)	814.5 MB/s
C copy backwards (64 byte blocks)	884.5 MB/s
C copy	847.1 MB/s
C copy prefetched (32 bytes step)	859.8 MB/s
C copy prefetched (64 bytes step)	884.9 MB/s
C 2-pass copy	783.7 MB/s
C 2-pass copy prefetched (32 bytes step)	809.1 MB/s
C 2-pass copy prefetched (64 bytes step)	832.1 MB/s
C fill	2313.4 MB/s
C fill (shuffle within 16 byte blocks)	2313.4 MB/s

Test item	Bandwidth
C fill (shuffle within 32 byte blocks)	532.2 MB/s
C fill (shuffle within 64 byte blocks)	524.2 MB/s
standard memcpy	649.4 MB/s
standard memset	2314.6 MB/s
NEON read	1551.0 MB/s
NEON read prefetched (32 bytes step)	1683.8 MB/s
NEON read prefetched (64 bytes step)	1773.6 MB/s
NEON read 2 data streams	446.1 MB/s
NEON read 2 data streams prefetched (32 bytes step)	840.9 MB/s
NEON read 2 data streams prefetched (64 bytes step)	887.2 MB/s
NEON copy	875.3 MB/s
NEON copy prefetched (32 bytes step)	895.2 MB/s
NEON copy prefetched (64 bytes step)	962.1 MB/s
NEON unrolled copy	871.8 MB/s
NEON unrolled copy prefetched (32 bytes step)	913.2 MB/s
NEON unrolled copy prefetched (64 bytes step)	928.5 MB/s
NEON copy backwards	882.0 MB/s
NEON copy backwards prefetched (32 bytes step)	897.6 MB/s
NEON copy backwards prefetched (64 bytes step)	972.1 MB/s
NEON 2-pass copy	865.1 MB/s
NEON 2-pass copy prefetched (32 bytes step)	909.6 MB/s
NEON 2-pass copy prefetched (64 bytes step)	929.5 MB/s
NEON unrolled 2-pass copy	769.5 MB/s
NEON unrolled 2-pass copy prefetched (32 bytes step)	802.3 MB/s
NEON unrolled 2-pass copy prefetched (64 bytes step)	837.3 MB/s
NEON fill	2313.5 MB/s
NEON fill backwards	2313.5 MB/s
VFP copy	866.4 MB/s
VFP 2-pass copy	788.6 MB/s

Test item	Bandwidth
ARM fill (STRD)	2313.0 MB/s
ARM fill (STM with 8 registers)	2314.2 MB/s
ARM fill (STM with 4 registers)	2314.0 MB/s
ARM copy prefetched (incr pld)	933.2 MB/s
ARM copy prefetched (wrap pld)	907.6 MB/s

Memory Latency Test

The average time is measured for random memory accesses in buffers of different sizes. The larger the buffer, the more significant the relative contributions of TLB, L1/L2 cache misses, and SDRAM accesses. For extremely large buffer sizes we expect to see the page table walk with several requests to SDRAM for almost all memory access (though 64MiB is not nearly large enough to experience this effect to the full).

Note: All numbers represent the extra time, which must be added to the L1 cache latency.

Note: Dual random read means that we are simultaneously performing two independent memory accesses at a time. If the memory subsystem cannot handle multiple outstanding requests, a dual random read has the same timings as two single reads performed one after the other.

Table 35: Memory Latency Test Results

Block Size	Single Random Read / Dual Random Read
1024	0.0 ns / 0.0 ns
2048	0.0 ns / 0.0 ns
4096	0.0 ns / 0.0 ns
8192	0.0 ns / 0.0 ns
16384	0.0 ns / 0.0 ns
32768	0.0 ns / 0.0 ns
65536	4.8 ns / 8.3 ns
131072	7.5 ns / 11.7 ns

262144	9.9 ns / 14.6 ns
524288	76.8 ns / 122.8 ns
1048576	114.8 ns / 163.5 ns
2097152	139.0 ns / 184.0 ns
4194304	151.4 ns / 193.4 ns
8388608	159.2 ns / 200.5 ns
16777216	167.7 ns / 212.5 ns
33554432	180.6 ns / 235.5 ns

7 Service and Firmware Update

7.1 Firmware Update

The Telit Modules firmware is updated through the USB Interface normally used for the AT Commands.

It is recommended to provide a USB interface on the User's Printed Circuit Board (where the Telit Module is soldered) to perform the physical connection between the Telit module and a Windows-based PC. This simple circuitry makes the firmware updating easy when a new firmware version is released.

During the User Application development or evaluation phase of the Telit module, the USB port implemented on the Telit Evaluation Board (Telit EVB) can be used to connect the Telit module to a Windows-based PC on which a dedicated tool for firmware updating is running.

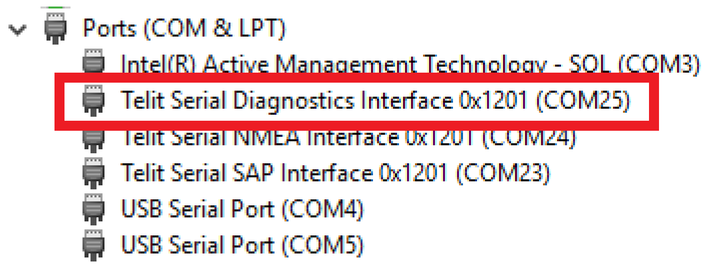
Telit provides the User with two tools to update the module firmware. The following paragraphs describe them.

TFI Update

The firmware update can be performed with a specific software tool supplied by Telit which runs on Windows-based PCs. The program will erase the contents of the flash memory, and then the program will write on the flash memory. The "LE910_xxx_TFI.exe" file includes a binary image. The binary image included in the TFI package will be checked with SHA256 hash prior to the start of the TFI firmware update procedure. The TFI will stop updating if data corruption has been detected.

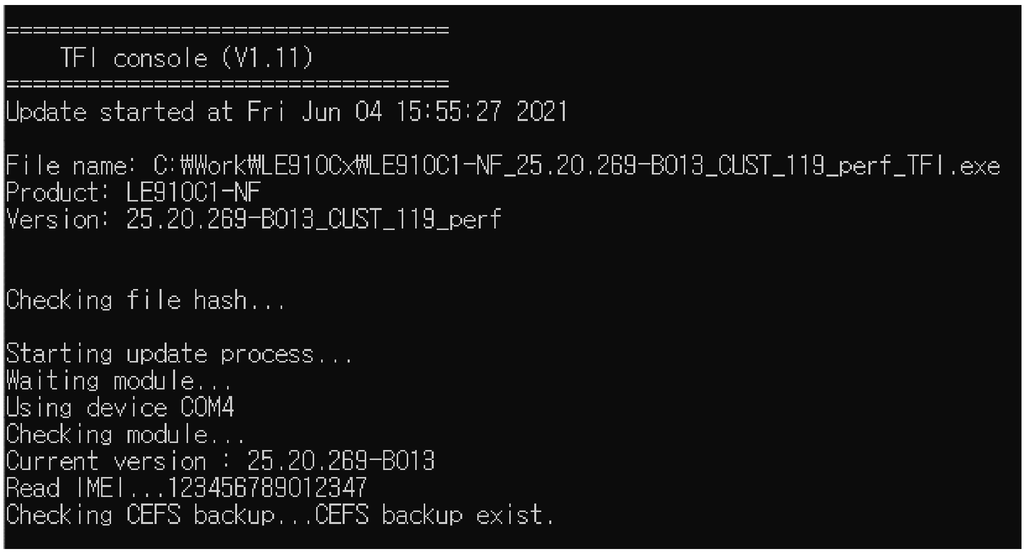
The procedure for updating the TFI is as follows.

1. Check Telit Serial Diagnostics Interface in your Windows device manager, before updating it with TFI.



Ports (COM & LPT)
Intel(R) Active Management Technology - SOL (COM3)
Telit Serial Diagnostics Interface 0x1201 (COM25)
Telit Serial NMEA Interface 0x1201 (COM24)
Telit Serial SAP Interface 0x1201 (COM23)
USB Serial Port (COM4)
USB Serial Port (COM5)

2. Run LE910 xxx TFI.exe. The windows CMD prompt appears.
   The TFI downloader detects Telit Serial Diagnostics and starts downloading.



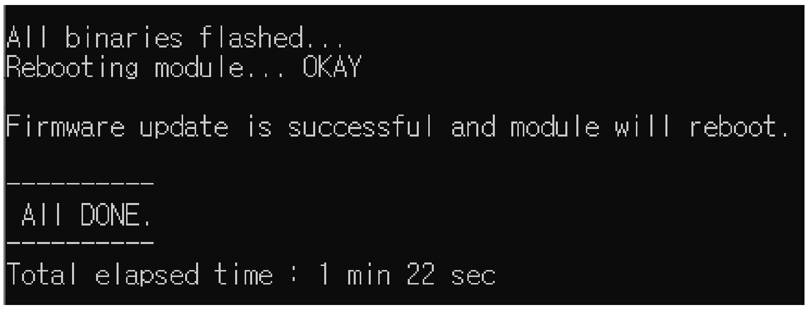
```
===================================
    TFI console (V1.11)
===================================
Update started at Fri Jun 04 15:55:27 2021

File name: C:₩Work₩LE910Cx₩LE910C1-NF_25.20.269-B013_CUST_119_perf_TFI.exe
Product: LE910C1-NF
Version: 25.20.269-B013_CUST_119_perf


Checking file hash...

Starting update process...
Waiting module...
Using device COM4
Checking module...
Current version : 25.20.269-B013
Read IMEI...123456789012347
Checking CEFS backup...CEFS backup exist.
```

3. Wait for the end of programming completed message.



```
All binaries flashed...
Rebooting module... OKAY

Firmware update is successful and module will reboot.

-----------
 All DONE.
-----------
Total elapsed time : 1 min 22 sec
```

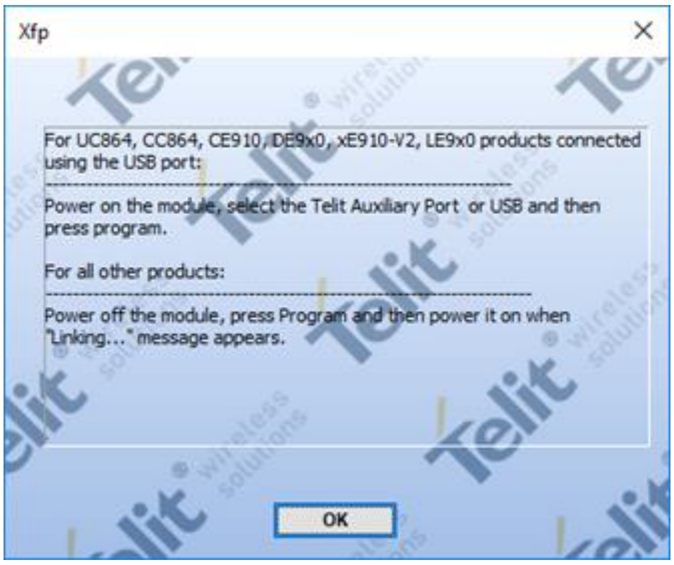
The Telit LE910Cx module is now programmed with the new firmware.

XFP Update

The module firmware update can be performed with the XFP Tool provided by Telit. It runs on Windows-based PCs. It erases the flash memory content, then downloads the new firmware on the flash memory. The XFP firmware image has CRC for each block and this CRC value is transferred to the module with the binary data. The update will be stopped if corrupted data is detected on the module.

The procedure for updating XFP is as follows.

1. Run Xfp.exe. The following window appears.



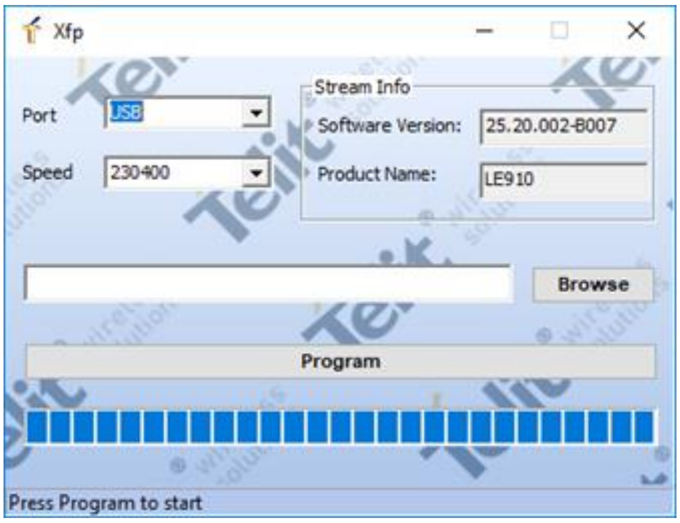
For UC864, CC864, CE910, DE9x0, xE910-V2, LE9x0 products connected using the USB port:

Power on the module, select the Telit Auxiliary Port or USB and then press program.

For all other products:

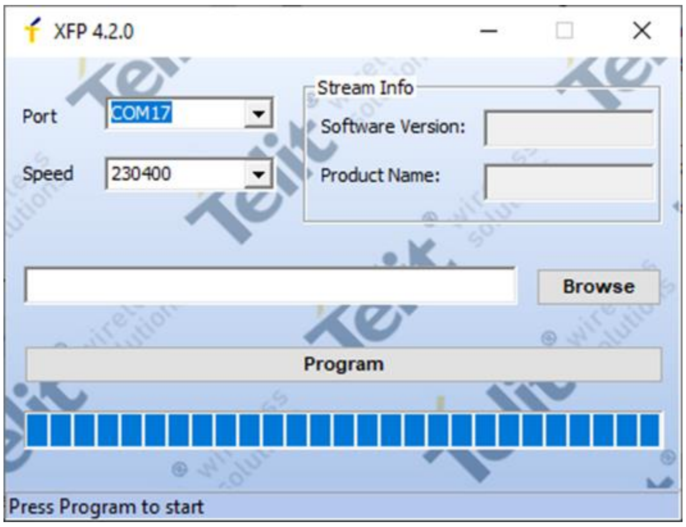
Power off the module, press Program and then power it on when "Linking..." message appears.

2. Click OK to continue.
3. To download over USB, select USB in the port combo box.



Port: USB
Speed: 230400
Stream Info
Software Version: 25.20.002-B007
Product Name: LE910
Press Program to start

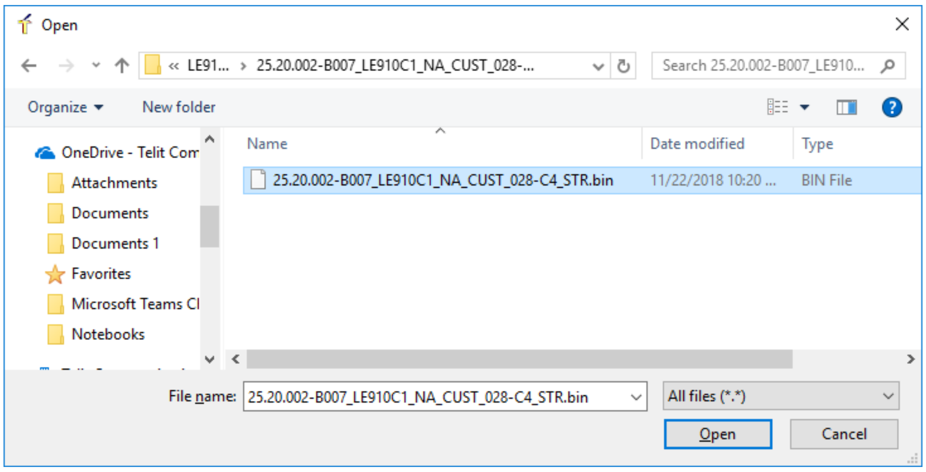
4. Alternatively, to download through UART, select the main UART port in the combo box.



Port: COM17
Speed: 230400
Stream Info
Software Version:
Product Name:
Press Program to start

UART download is supported from version 4.2.0 or higher.

5. Click Browse and select the stream binary that you want to use for the upgrade.

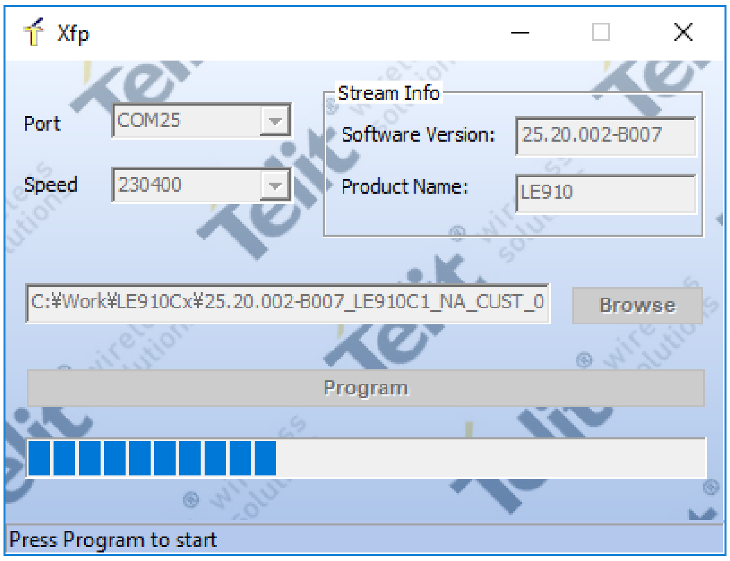


File name: 25.20.002-B007_LE910C1_NA_CUST_028-C4_STR.bin

Name	Date modified	Type
25.20.002-B007_LE910C1_NA_CUST_028-C4_STR.bin	11/22/2018 10:20 ...	BIN File

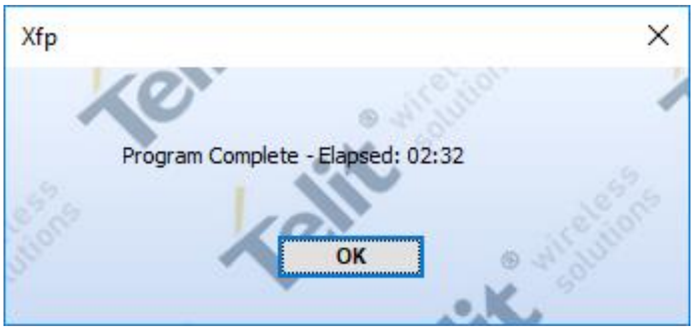
All files (*.*)

6. Click **Program** to start upgrading.



Port: COM25
Speed: 230400
Software Version: 25.20.002-B007
Product Name: LE910
C:¥Work¥LE910Cx¥25.20.002-B007_LE910C1_NA_CUST_0
Press Program to start

Once the module upgrade is successful, the following window appears on the screen.



Program Complete - Elapsed: 02:32

8 Acronyms and Abbreviations

Table 36: Acronyms and Abbreviations

Acronym	Definition
ARFCN	Absolute Radio Frequency Channel Number
AT	Attention command
BA	BCCH Allocation
BCCH	Broadcast Control Channel
CA	Cell Allocation
CBM	Cell Broadcast Message
CBS	Cell Broadcast Service
CCM	Current Call Meter
CLIR	Calling Line Identification Restriction
CTS	Clear To Send
CUG	Closed User Group
DCD	Data Carrier Detect
DCE	Data Communication Equipment
DCS	Digital Cellular System
DGPS	Differential GPS, the use of GPS measurements, which are differentially corrected
DNS	Domain Name System
DSR	Data Set Ready
DTE	Data Terminal Equipment
DTMF	Dual Tone Multi Frequency
DTR	Data Terminal Ready
GGA	GPS Fix data
GLL	Geographic Position – Latitude/Longitude
GLONASS	Global positioning system maintained by the Russian Space Forces
GMT	Greenwich Mean Time
GNSS	Any single or combined satellite navigation system (GPS, GLONASS and combined GPS/GLONASS)
GPRS	Global Packet Radio Service
GPS	Global Positioning System
GSA	GPS DOP and Active satellites
GSM	Global System Mobile
GSV	GPS satellites in view
HDLC	High Level Data Link Control
HDOP	Horizontal Dilution of Precision
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
IRA	International Reference Alphabet
IWF	Interworking Function
ME	Mobile Equipment
MO	Mobile Originated
MT	either Mobile Terminated or Mobile Terminal
NMEA	National Marine Electronics Association
NVM	Non-Volatile Memory
PCS	Personal Communication Service
PDP	Packet Data Protocol
PDU	Packet Data Unit
PIN	Personal Identification Number
PPP	Point to Point Protocol

Acronym	Definition
PUK	Pin Unblocking Code
RLP	Radio Link Protocol
RMC	Recommended minimum Specific data
RTS	Request To Send
SAP	SIM Access Profile
SCA	Service Center Address
SMS	Short Message Service
SMSC	Short Message Service Center
SMTP	Simple Mail Transport Protocol
TA	Terminal Adapter
TCP	Transmission Control Protocol
TE	Terminal Equipment
UDP	User Datagram Protocol
USSD	Unstructured Supplementary Service Data
UTC	Coordinated Universal Time
VDOP	Vertical dilution of precision
VTG	Course over ground and ground speed
WAAS	Wide Area Augmentation System

9 Related Documents

Refer to https://dz.telit.com/ for current documentation and downloads.

Table 37: Related Documents

Ref.	Book Code	Document Title
[1]	80502ST10950A	LE910Cx AT Commands Reference Guide
[2]	1VV0301249	Telit EVB (Evaluation Board) User Guide
[3]	80000NT11246A	LE910Cx Digital Voice Interface Application Note
[4]	80502NT11759A	LE910Cx Audio over USB Application Note
[5]	80000ST10028A	IP Easy User Guide Application Note
[6]	80502NT11769A	LE910Cx Linux device driver Application Note
[7]	1VV0301613	uxfp User Guide
[8]	-	ETSI GSM 03.38, 23.038
[9]	-	ETSI GSM 07.07, 27.07
[10]	-	ETSI GSM 11.11, 51.011, 31.101, 31.102
[11]	-	ETSI GSM 11.14, 51.014
[12]	-	ETSI GSM 27.005
[13]	-	ITU-T Recommendation E.164
[14]	-	ITU-T Recommendation V.24

10 Document History

Table 38: Document History

Revision	Date	Changes
14	2023-12-05	Updated document template with new branding style Added LE910C1-APX and LE910C1-SNX to Applicability Table Modified clauses: 5.2 Application System Overview - RAM Memory - TX (ThreadX) 5.4 Advanced Operations - SMS Management - Select SMS Format Type
13	2022-04-27	Minor language and grammar corrections
12	2022-03-21	Added clause: 4.4.5. Blocking access to AT and USB interfaces
11	2021-08-11	Section 4.2.21.4 NTP deleted
10	2021-06-04	Section 4.2.14.2. TX (ThreadX) updated Section 4.2.12. GPIO updated
9	2021-04-28	Minor changes on the language Minor changes on the layout Legal Notices updated Section 4.2.8. Application Development Environment updated Section 4.2.14.2. TX (ThreadX) updated
8	2021-02-05	Section 4.2.12. GPIO added Section 4.2.18. Ethernet Interface added Section 4.1 General Functionality and Main Features updated Section 4.2. Application system overview updated Section 6.1.1. TFI updated Section 6.1.2. XFP updated Section 4.2.16. USB Interface updated Section 4.2.13.1.2. Apps to Modem updated Section 4.2.13.1.1. Apps to external MCU updated Section 4.2.15.1. LE (Linux) updated Section 4.2.12. GPIO-Keys deleted (moved to "80502NT11769A_LE910Cx_Linux_device_driver_Application_Note")
7	2020-09-25	Section 4.2.10 Wake up Events updated Section 4.2.12. GPIO-Keys updated Section 4.2.15 USB Interface updated

Revision	Date	Changes
6	2020-06-04	Section 2. LE910Cx Variants added Section 3.2. Architecture based on ThreadX added Section 3.2.9 Wake up Events updated Applicability table updated Section 4.1 General Functionality and Main Features updated Section 4.2 Application system overview updated Section 4.2.10 SPI updated Section 4.2.11. GPIO-Keys updated Section 4.2.12 Serial Interfaces updated Section 4.2.14 UART updated Section 4.2.15 USB Interface updated Section 4.2.16 HSIC Interface updated Section 4.2.17. SD/MMC Interface updated Section 4.2.18. RTC updated Section 4.2.22. Power Management updated Section 4.3.2. Command Response Timeout updated Section 4.4.4.4. NMEA 0183 updated Section 4.4.4.2. LE910Cx Serial Ports updated Section 4.4.4.5. Checking GNSS Device Functionality updated Section 4.5.1.2 Standard ECM/RNDIS updated Section 4.5.1.3 MBIM/RmNet updated Section 4.5.8 Data Concurrency updated Section 4.2.10 Wake up Events updated
5	2020-01-31	Section 3.2.17. SD/MMC Interface added Section 3.2.11. GPIO-Keys updated Section 3.3.6. Network Checking updated Section 3.2.9. Wake up Events updated
4	2019-11-21	Section 3.2.9. Wake up Events added Section 3.2.10. GPIO-Keys updated Section 3.3.6. Network Checking updated
3	2019-09-27	Section 3.2.15. HSIC Interface added Section 3.2.14. USB Interface updated Section 3.2.9. SPI updated Section 3.3.2. Command Response Timeout updated
2	2019-06-28	Section 3.3.5.8. Preferred Operator List, added Section 3.3.6. Network Checking added Section 3.2.14. USB Interface, updated Section 3.3.4.1. RAT selection updated Section 3.2.1.1 Added LE910C1-EU 4Gbit memory information updated
1	2019-01-31	Initial version

From Mod.0818 Rev.11

Technical Documentation