OOB Demo Architecture Description

Pinnacle 100 Modem

Application Note *v1.0*

1 Introduction

This application note describes the architecture and functionality of the Pinnacle 100 OOB Demo **[A], [B]**. It is intended to outline how the OOB Demo is structured and how it can be used as the basis for further development or as a template for new applications.

2 Nomenclature

- Source code snippets are displayed using the `Courier New font`.
- References to Zephyr API content in code snippets and main body text are displayed using ***bold, italicized text***.
- References to user code in the main body text are displayed using **bold text**.
- Datatypes used throughout are defined by the Zephyr API.

3 Overview

The Pinnacle 100 **[C]** is Laird Connectivity's first entry into the Cellular/IoT module market. It supports LTE-M **[D]** and NB-IoT **[E]** connectivity via a Sierra HL7800 cellular radio **[F]**. A Nordic nRF52840 SOC **[G]** acts both as the host controller and implements support for BLE v5.2.

4 Assumptions

This document describes the functionality of the v2.1.0 OOB demo release.

The OOB Demo is shipped and compiled for v2.3 of Zephyr **[H]**.

To support initial development efforts, a development kit (DVK) for the Pinnacle 100 is provided by Laird Connectivity **[I], [J], [K]**. It is assumed that the OOB Demo is initially executed in this environment.

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

5 Pinnacle 100 Hardware Architecture

The hardware architecture of the Pinnacle 100 Modem is shown in Figure 1. Its elements are described as in Table 1.

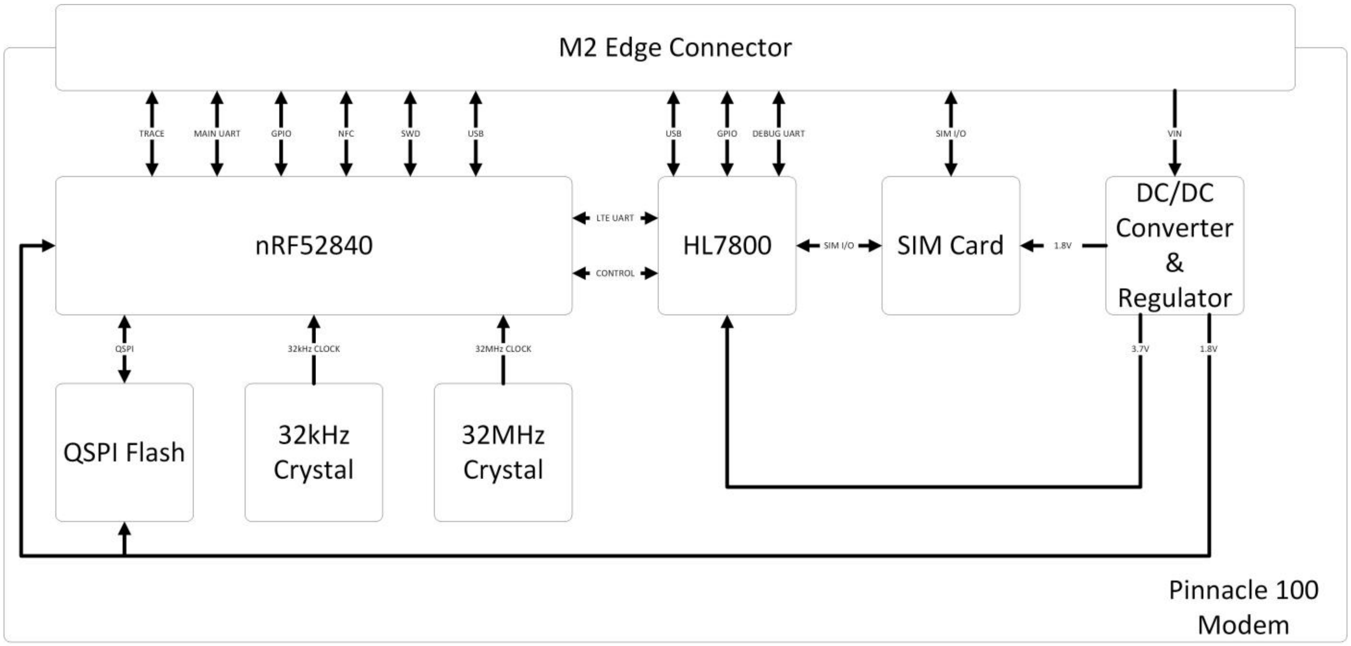

Figure 1: Pinnacle 100 Modem hardware architecture

Table 1: Pinnacle 100 Modem hardware elements and interfaces

Element	Description	Interface	Details
32kHz Crystal	A software selectable 32kHz clock source for the nRF52840.	32 kHz CLOCK	The output from the crystal to the nRF52840.
32Mhz Crystal	A software selectable 32Mhz clock source for the nRF52840.	32 MHz CLOCK	The output from the crystal to the nRF52840.
DC/DC Converter & Regulator	This is the power supply circuit for the module. It accepts an input voltage in the range of 2.2 to 5.5V. A buck/boost DC/DC converter is used to derive the initial 3.7V stage, with a regulator being used to derive the 1.8V supply.	1.8V	This is the 1.8V output from the final regulator stage.
		3.7V	This is the 3.7V output from the boost/buck DC/DC converter stage.
		VIN	This is the input voltage to the boost/buck DC/DC converter stage.
HL7800	This is the Sierra HL7800 modem used to provide LTE connectivity.	Debug UART	Provides access to the debug UART for viewing HL7800 behavior.
		GPIO	Allows access to GPIO exposed by the HL7800.
		USB	Interface to the HL7800 USB controller.
M2 Edge Connector	Provides an M2 standard **[L]** edge connector for physical interfacing with the Pinnacle 100 module.		
nRF52840	This is the host microcontroller and BLE SOC.	Control	These are the control I/O lines for detecting modem events and configuring behavior.
		Main UART	This is the UART allowing user connectivity to the nRF52840.

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

Element	Description	Interface	Details
		GPIO	These are the GPIO lines available for external user connectivity.
		LTE UART	This is the UART used to control the HL7800 modem.
		NFC	Allows connectivity to an external NFC antenna for NFC functionality.
		SWD	These are the SWD lines **[M]** used for programming the nRF52840 in situ, and for connectivity of an appropriate ICD.
		Trace	These are the trace lines **[N]** used to interact with the nRF52840 Trace module.
		USB	Allows connectivity to the nRF52840 USB module.
QSPI Flash	This is a 64Mb QSPI flash memory device **[O]** used for storage of user configuration settings and application update files.	QSPI	This is the QSPI interface used for control of the QSPI Flash part.
SIM Card	This is the SIM Card used to unique identify the modem on the LTE network.	SIM I/O	These are the control lines used to interact with the inserted SIM Card.

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

6 OOB Demo Sensor Support

Data from two sensor types can be collected by the OOB Demo. These are described as follows.

6.1 BL654-BME280

The Pinnacle 100 DVK includes a BL654-BME280 sensor (refer to **Figure 2**). This is based upon the Laird Connectivity BL654 module **[P]**, and incorporates a Bosch BME280 sensor **[Q]**. The BME280 allows measurement of humidity, temperature and pressure. The BL654-BME280 is supplied with a smartBASIC application **[R]**, **[S]** pre-programmed which in addition to the measurement of the above, also calculates dew-point. The smartBASIC application implements the Environmental Sensing Service **[T]** via a GATT Server. This allows the Pinnacle 100 DVK BLE Client to connect to it and read back the measured and calculated data.

Figure 2: BL654-BME280 Sensor

Note: The BLE654-BME280 can be replaced with any sensor implementing the Environmental Sensing Service.

Note: The ESS implementation has not been submitted for PTS compliance **[U]**.

6.2 BT510

The BT510 (refer to **Figure 3**) is a BLE based sensor intended for IoT applications **[V]**. It supports measurement of temperature, vibration/acceleration and proximity via a magnetic reed switch. Unlike the BL654-BME280, the BT510 can be configured to publicize its measured data via BLE extended advertisements. This allows data to be retrieved in a connectionless manner which requires less overhead than the connection orientated type. The BT510 is intended to be fully user programmable via Zephyr development **[W]**.

Figure 3: BT510 sensor

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

7 OOB Demo Operating Modes

The OOB Demo can be compiled to operate in one of two modes, as follows. In both cases, sensor data is collected via BLE and relayed to a cloud-based service.

7.1 LTE-M connectivity

In this mode **[X]**, as shown in Figure 4, sensor BLE data is gathered and transmitted to an instance of Laird Connectivity's Bluegrass application **[Y]** hosted by AWS. Data from up to one BL654-BME280 sensor and up to 15 BT510 sensors can be transmitted.

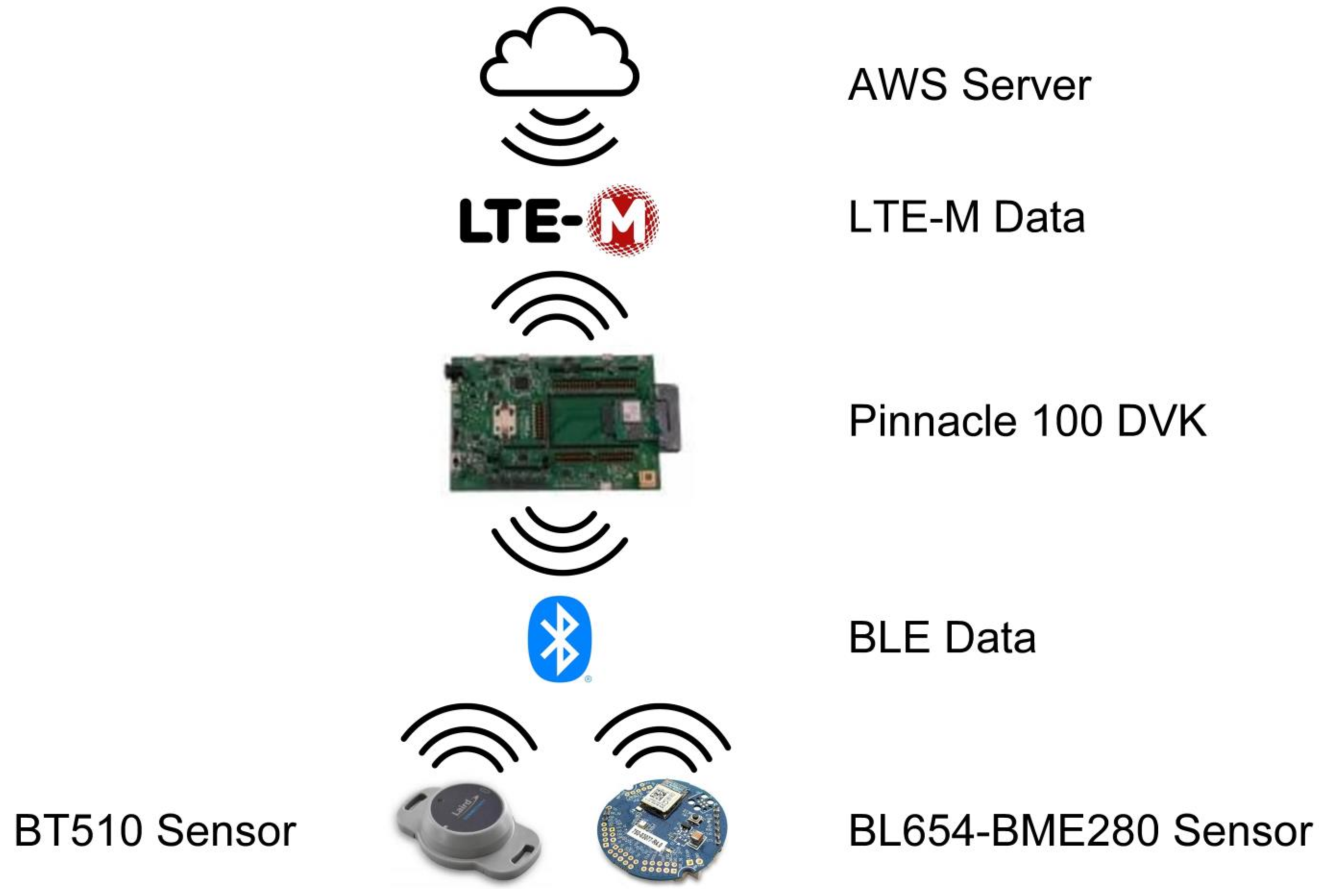

Figure 4: Pinnacle 100 OOB Demo configured for LTE-M operation

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

7.1.1 LTE-M Connectivity Protocol Layers

Figure 5 shows the layering of protocols used for the LTE-M connectivity variant from an OSI reference model perspective **[Z]**. Application level messages are relayed to and from an MQTT broker **[A1]**. Message security is ensured via usage of TLS **[B1]**, with TCP **[C1]** being used to establish a connection with the host system.

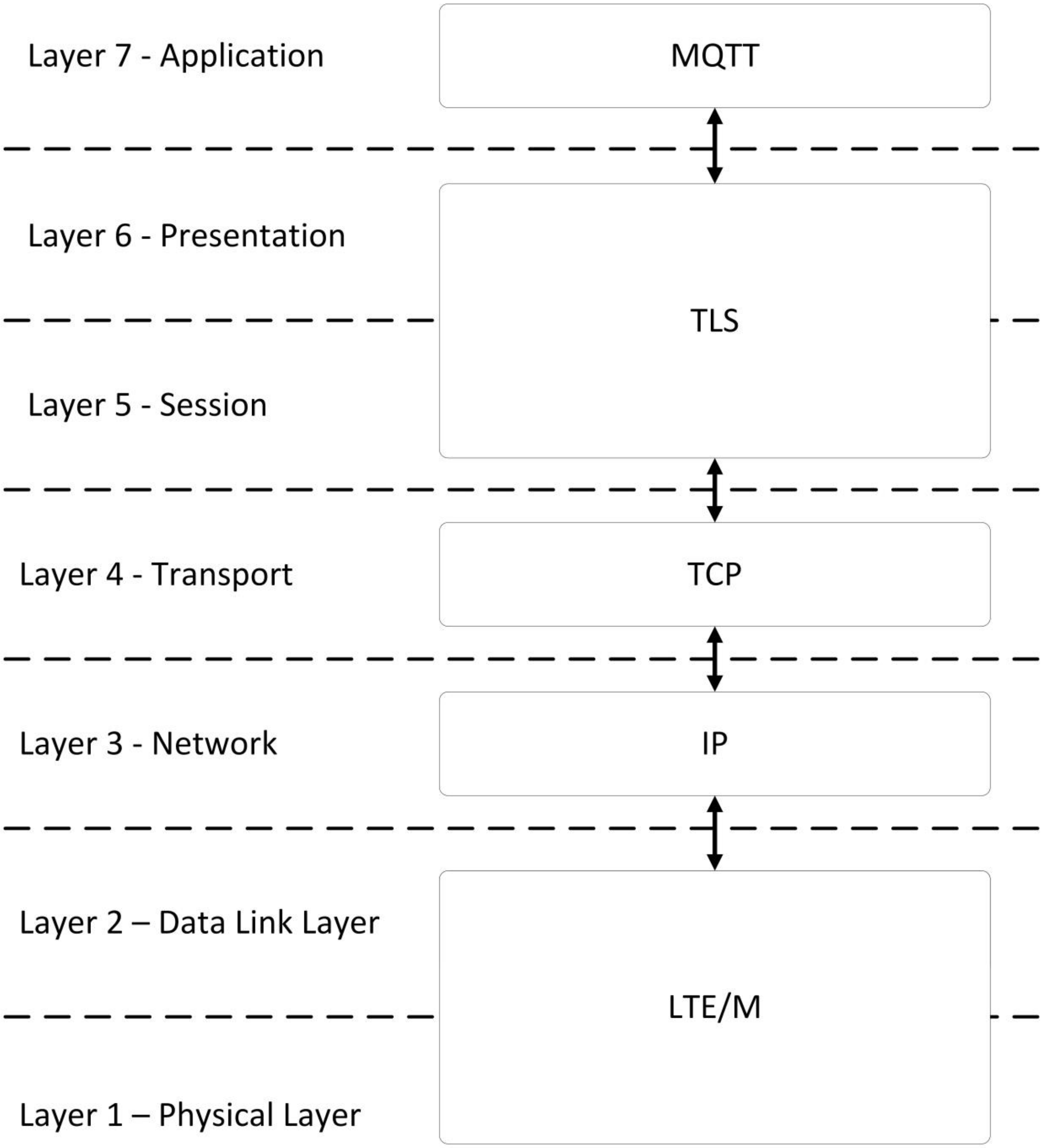

Figure 5: LTE-M connectivity protocol layers

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

7.2 NB-IoT Connectivity

In this configuration **[D1]**, a single BL654-BME280 is connected to via BLE and its data forwarded to a Leshan server instance **[E1]**.

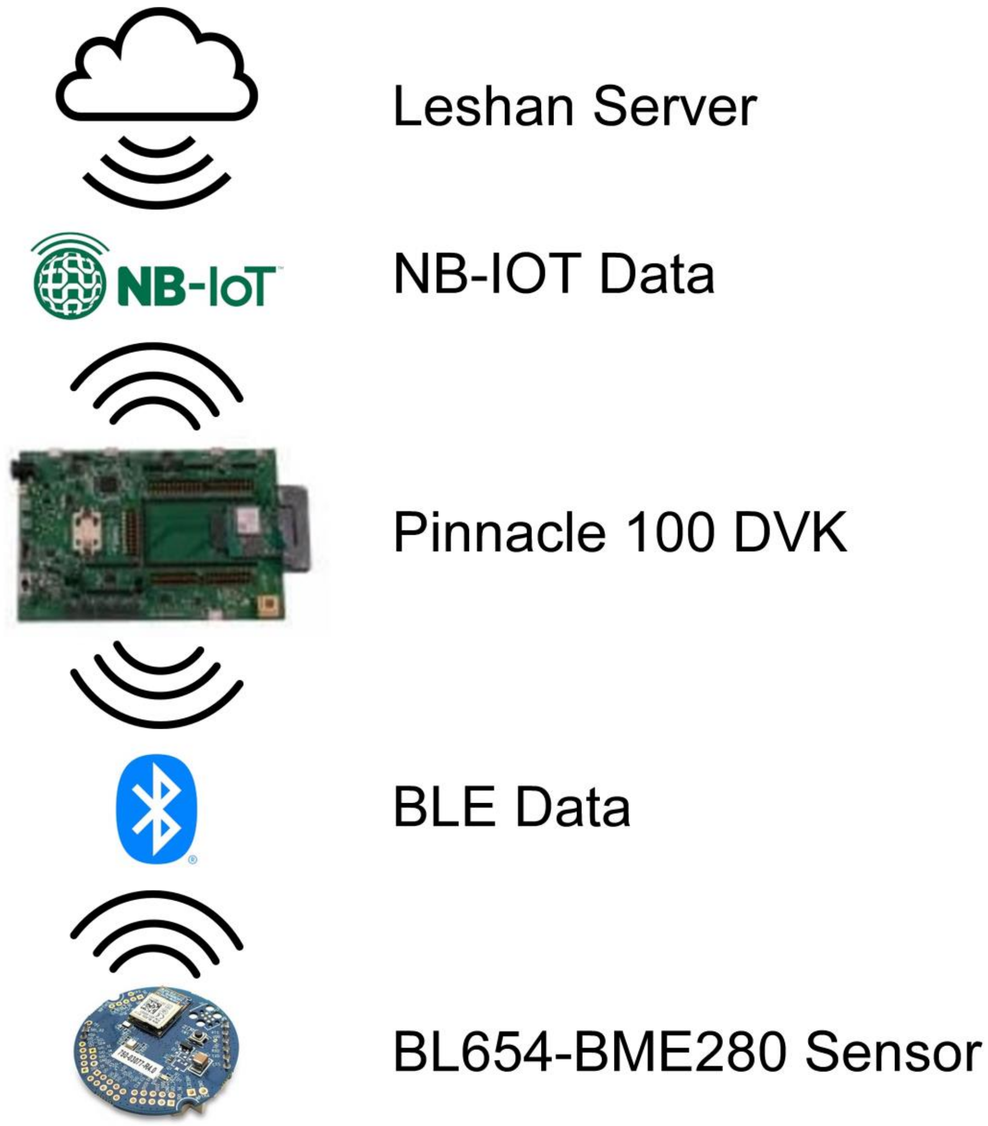

Figure 6: Pinnacle 100 OOB Demo configured for NB-IoT operation

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

7.2.1 NB-IoT Connectivity Protocol Layers

The protocol layers implemented for the NB-IoT connectivity demo are shown in Figure 7. LwM2M [F1] over CoAP [G1] is used at the Application level. To minimize overhead, UDP [H1] is used for the transport layer, with this necessitating the usage of DTLS [I1] for securing the data.

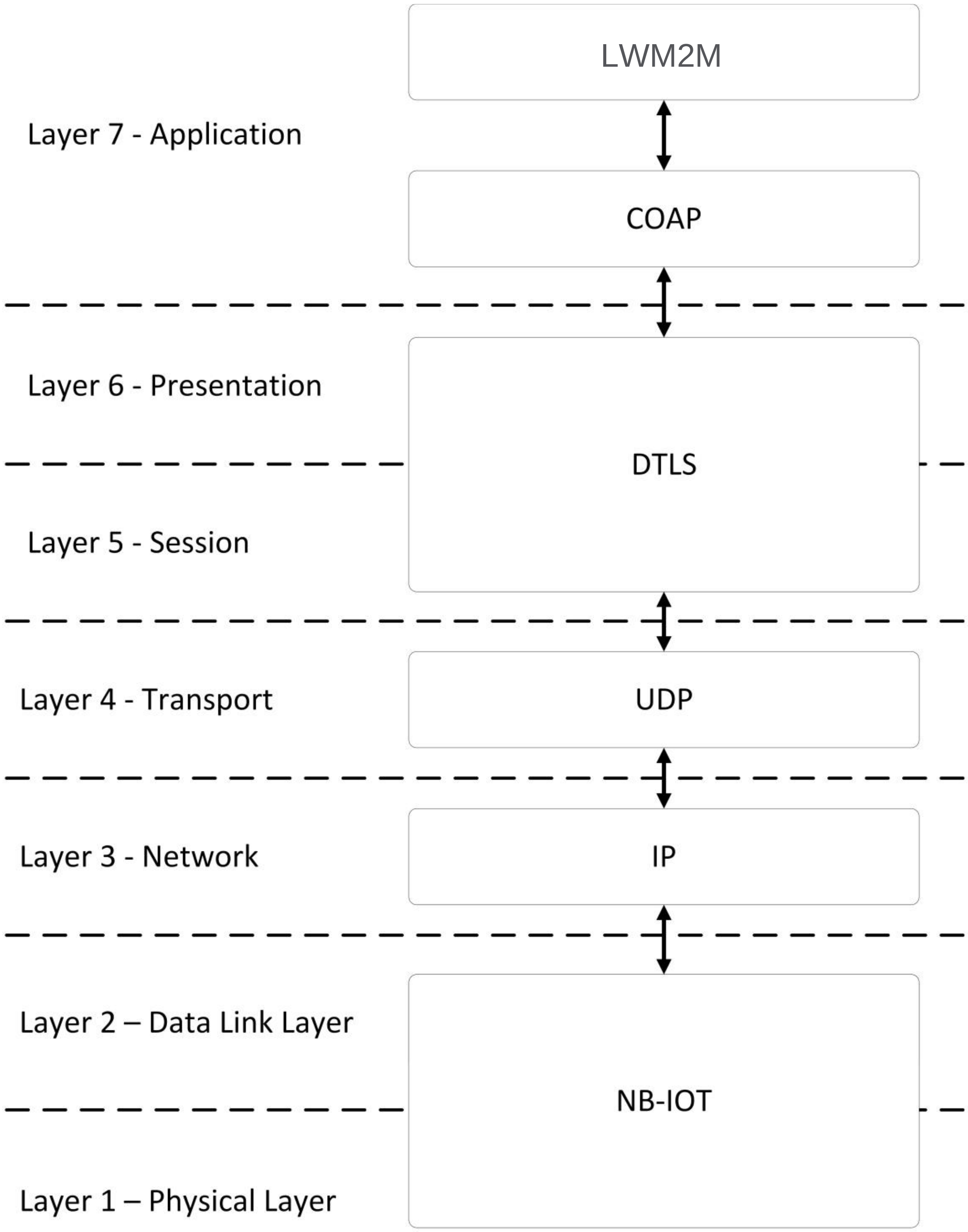

Figure 7: NB-IoT connectivity protocol layers

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

8 OOB-Demo Source Code Packages

The following describes the high-level source code packages used within the OOB Demo. Packages common to both implementations are first described, followed by the packages specific to the different implementations.

8.1 Common Source Code Packages

Error! Reference source not found. shows details of the source code packages used by both OOB Demo implementations.

Table 2: Common source code packages

Package Name	Description	File	Description
Coordinator	Application coordination.	main.c	Implements the application entry point and initial thread. Initialises the application and provides the main application state machine.
BLE Peripheral	BLE Peripheral functionality for application configuration and interrogation.	single_peripheral.c	Low level interface for interacting with underlying BLE hardware.
		dis.c	Extended version of Zephyr's in-built Device Information Service. Exposes the standard Device Information Service characteristics and maps to application values.
		ble_cellular_service.c	Custom service that exposes modem related parameters for read and write access.
		ble_power_service.c	Custom service that exposes the power level of the modem as a pair of 8-bit numbers. Also controls reboot of the module.
BL654 Sensor	BLE Central functionality for interfacing with BL654-BME280 sensors.	bl654_sensor.c	High level interface to BL654-BME280 sensors. Implements the state machine used to read back and format data from connected sensors.
		ble_sensor_service.c	Low level interface for interacting with underlying BLE hardware.
Framework	Utility code and inter-thread messaging service	framework.c	Public interface to the inter-thread messaging service.
		frameworkstubs.c	Base implementations of overrideable inter-thread messaging service functions.
		bufferpool.c	Heap based memory buffer allocation service.
		template.c	Template file used for successive module development.
		bracket.c	Container for JSON messages.
NV	Application non-volatile data management.	nv.c	Non-volatile data management and interface to underlying Zephyr nvs [J1].

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

Package Name	Description	File	Description
LTE	LTE-M & NB-IoT connectivity and communications management.	lte.c	Interface to the underlying Zephyr networking subsystem **[K1]** and HL7800 modem driver.
		hl7800.c	Driver for the HL7800 modem.

8.2 LTE-M Connectivity-Specific Source Code Packages

The source code packages specific to the LTE-M Connectivity implementation are shown in **Error! Reference source not found.**.

Table 3: LTE-M connectivity specific source code packages

Package Name	Description	File	Description
AWS	AWS connectivity and communications management.	aws.c	AWS connectivity and communications management. Connects to the AWS MQTT broker and publicizes events for processing elsewhere.
Bluegrass	Connectivity and communications management with Laird Connectivity's Bluegrass implementation.	bluegrass.c	Public interface to Bluegrass connectivity management.
		sensor_adv_format.c	Describes the content of BT510 sensor advertisements and scan responses.
		sensor_cmd.c	JSON prototypes of the commands used to interact with BT510 sensor instances.
		sensor_gateway_parser.c	Processes JSON messages received from Bluegrass for configuring modem and sensor behavior.
		sensor_log.c	Circular buffer for storing event data from BT510 sensors.
		sensor_table.c	
		sensor_task.c	Implements the thread used to communicate with sensors and the Bluegrass instance.
		shadow_builder.c	Utility module for building AWS shadow data for BT510 sensor instances.
		to_string.c	Utility module for converting numeric data to textual format.

The source code packages and dependencies for the LTE-M Connectivity implementation are shown in Figure 8.

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

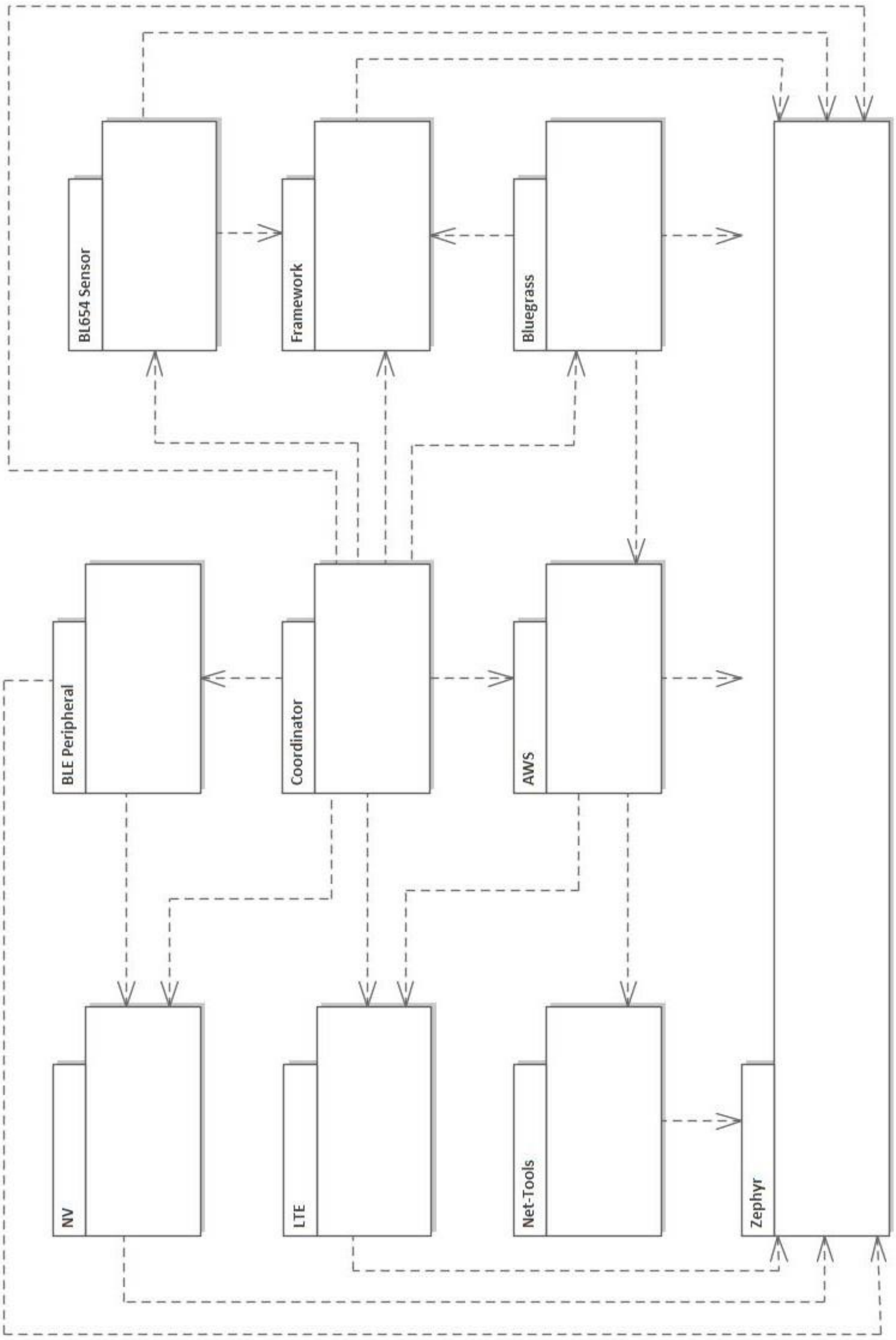

Figure 8: LTE-M implementation packages and dependencies

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

8.3 NB-IoT Connectivity-Specific Source Code Packages

The source code packages specific to the NB-IoT Connectivity implementation are shown in **Error! Reference source not found.**.

Table 4: NB-IoT connectivity specific source code packages

Package Name	Description	File	Description
LwM2M	LwM2M interface and collection and formatting of BL654-BME280 sensor data.	lwm2m_client.c	Manages the connection to the Leshan server instance and forwards BL654-BME280 sensor data.

The source code packages and dependencies for the NB-IoT Connectivity implementation are shown in Figure 9.

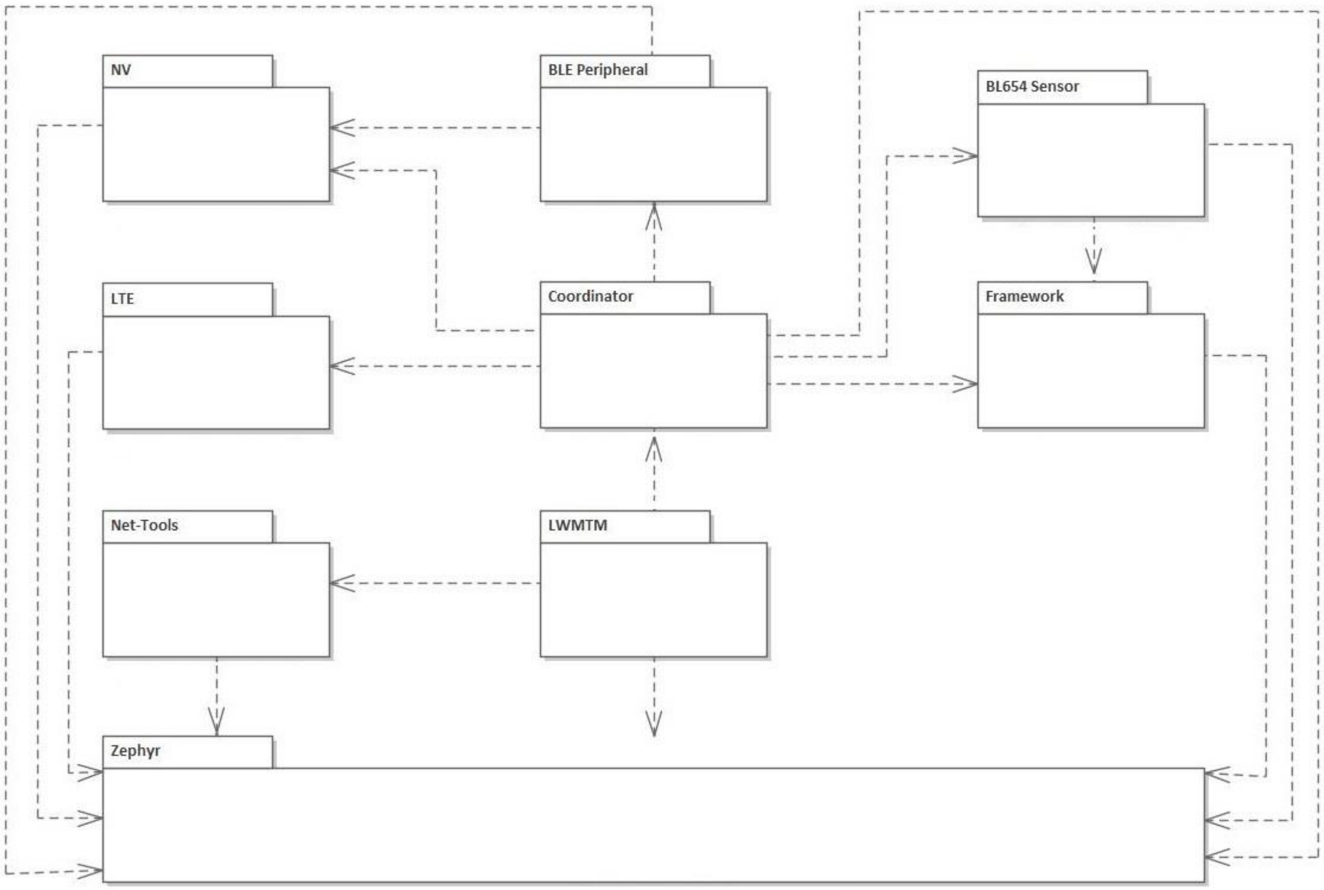

Figure 9: NB-IoT Connectivity demo source code packages and dependencies

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

9 OOB Demo State Machines

The following describe the state machines implemented by the OOB Demo versions.

9.1 LTE-M Connectivity State Machine

The state machine execute by the LTE-M Connectivity implementation is shown in Figure 10.

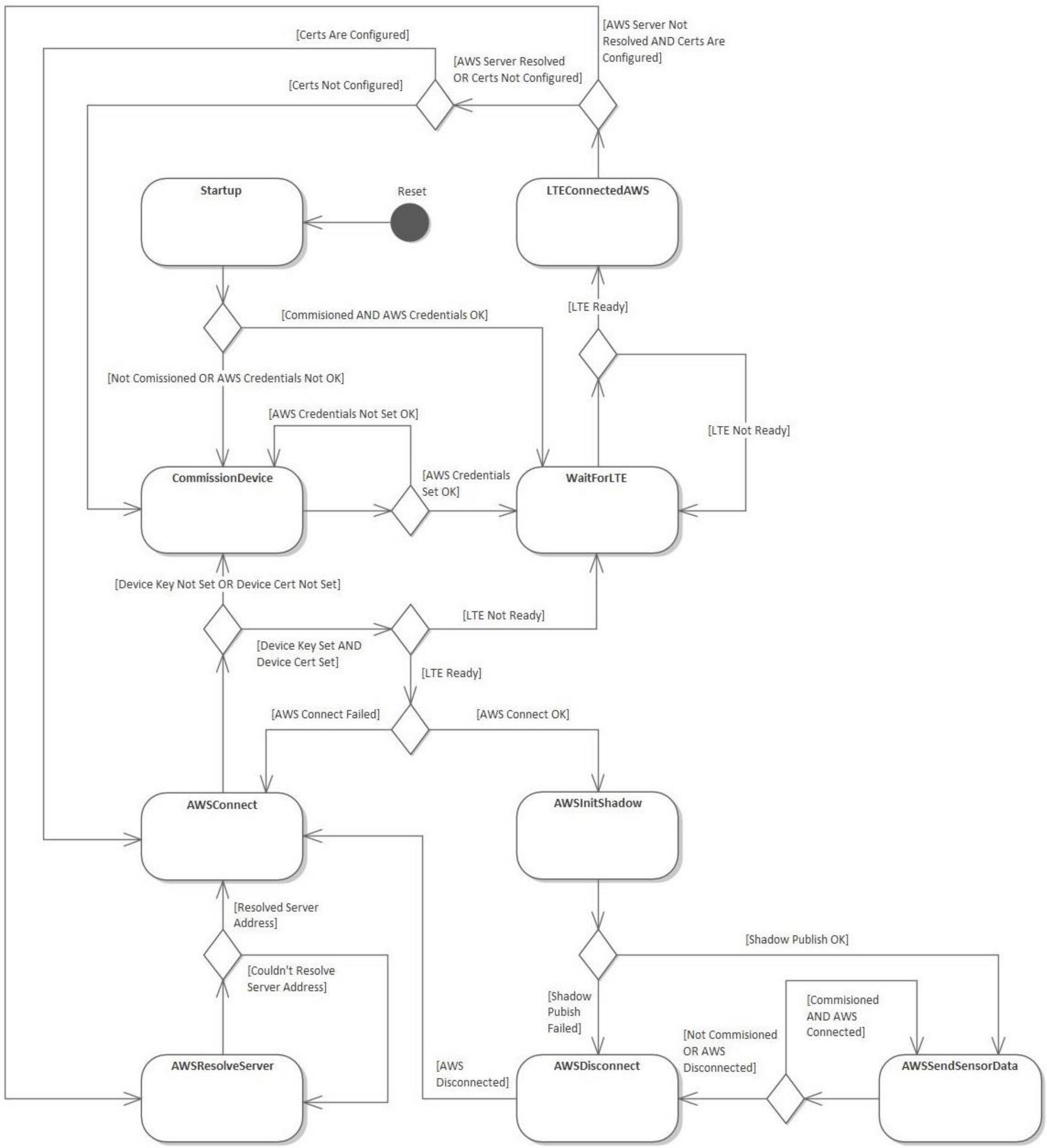

Figure 10: LTE-M Connectivity State Machine

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

Activities performed by each state are described in **Error! Reference source not found.**.

Table 5: LTE-M Connectivity state machine states and transitions

State	Description	Transitions to	Condition
AWSConnect	During this state, connectivity to the AWS server is established. Unsuccessful attempts to connect to the AWS server result in transitioning back to this state to reattempt establishing the connection.	CommisionDevice	If insufficient certificate or key information is available, the CommisionDevice state is transitioned to where the user can enter the data via the mobile app.
		WaitForLTE	If a connection is unavailable to the LTE network, the WaitForLTE state is returned to.
		AWSInitShadow	Once a connection with the AWS server is successfully established, the AWSInitShadow is transitioned to for initializing AWS device shadow data.
AWSDisconnect	This disconnects from the AWS server.	AWSConnect	Transitions to this state upon disconnecting from the AWS server.
AWSInitShadow	The AWSInitShadow state handler attempts to publish sensor shadow data with the AWS server.	AWSDisconnect	If publishing of sensor shadow data fails, the AWSDisconnect state is transitioned to where disconnection from the AWS server is performed.
		AWSSendSensorData	Upon successful publishing of sensor shadow data, a transition is made to the AWSSendSensorData where data read from connected sensors is continuously uploaded.
AWSResolveServer	Upon entry to this state, an attempt is made to perform a DNS lookup of the AWS server IP address. If unsuccessful, a transition is performed back to this state to reattempt the DNS lookup.	AWSConnect	A transition is performed to the AWSConnect state once the IP address of the AWS server has successfully been looked up via the DNS.
AWSSendSensorData	Whilst a connection to the AWS server is available, and sufficient commissioning data is available, this state is maintained continuously to process messages received from the AWS server, and for forwarding sensor data read over BLE to the AWS server.	AWSDisconnect	If insufficient commissioning data, or a connection to the AWS server is unavailable, a transition is made to perform the disconnection from the AWS server.
CommisionDevice	Commisioning data consists of certificate and key information. If unavailable or, validation of them fails, this state is re-entered.	WaitForLTE	Upon successful validation of user certificate and key information, a transition is made to the WaitForLTE state to wait for establishment of a connection to the LTE network.

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

State	Description	Transitions to	Condition
LTEConnectedAWS	The LTEConnectedAWS state is entered once connectivity to the LTE network is established. During this state, connectivity to AWS is established.	AWSConnect	If the IP address of the AWS server has been resolved, and sufficient certificate data is available, this state is transitioned to for establishment of a connection with the server.
		AWSResolveServer	With sufficient certificate data, but the IP address of the AWS server unresolved, this state is transitioned to perform a DNS lookup of the AWS server address.
		CommisionDevice	If certificate data is not available, the CommisionDevice state is reverted to such that the user can add the certificate data via the mobile app.
Startup	This state is entered following initialization of the system.	CommisionDevice	If the application does not have sufficient commissioning and AWS credential data, it enters this state and waits here until the user enters the data via the mobile app.
		WaitForLTE	With sufficient commissioning and AWS credential data, this state is entered and the main thread suspended via the lte_ready_sem semaphore.
WaitForLTE	The WaitForLTE state is entered whilst waiting for a connection to the LTE network to be made.	LTEConnectedAWS	Upon successful connection to the LTE network, the lte_ready_sem semaphore is signaled. This restarts the main thread which then transitions to the LTEConnectedAWS state.

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

9.2 NB-IoT Connectivity Demo State Machine

The state machine of the NB-IoT Connectivity OOB Demo implementation is shown in Figure 11.

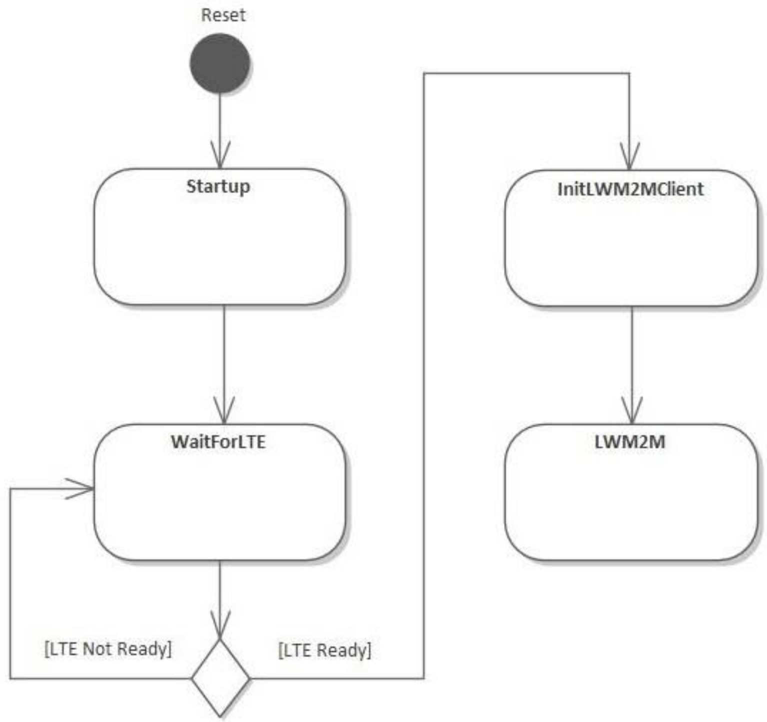

Figure 11: NB-IoT Connectivity state machine

Activities performed by each state are described in **Error! Reference source not found.**.

Table 6: NB-IoT Connectivity state machine states and transitions

State	Description	Transitions to	Condition
InitLWM2MClient	Initializes connectivity with the Leshan server instance and any data structures associated with data for upload.	LWM2M	Upon successful initialization of the LwM2M client, a transition is performed to the LWM2M state to begin uploading BL654-BME280 sensor data.
LWM2M	This state waits for BL654-BME280 messages to be received and forwards them to the Leshan server when received.	N/A	This state is maintained once entered for the lifespan of the application.
Startup	This state is entered following initialization of the system.	WaitForLTE	No additional behaviour is performed in this state for the NB-IoT Connectivity demo, it transitions immediately to the WaitForLTE state.
WaitForLTE	The WaitForLTE state is entered whilst waiting for a connection to the LTE network to be made.	InitLWM2MClient	Upon successful connection to the LTE network, the lte_ready_sem semaphore is signaled. This restarts the main thread which then transitions to the InitLWM2MClient state.

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

10 HL7800 Modem Driver

The HL7800 modem driver is supplied as part of the OOB Demo. Zephyr provides a generic modem implementation - this is extended to support the HL7800 functionality. The HL7800 acts as the network interface for the application and is interacted with via calls to Zephyr's Network Interface **[L1]**.

This section describes the usage of the HL7800 modem driver.

10.1 Compile Time Configuration

Compile time configuration of the modem is achieved via addition of configuration data to the application prj.conf file. This is located at the same folder level as the CMakelists file used to instruct CMake what files the user application consists of. Modem related settings are applied to the modem during compilation of the user application.

These are summarised in Table 7.

Table 7: HL7800 modem driver compile time configuration

Configuration setting	Purpose	Type	Default
MODEM_HL7800_FW_UPDATE	Enables updates of the modem firmware via XModem from the main user application.	bool	N
MODEM_HL7800_SET_APN_NAME_ON_STARTUP	Automatically sets the modem APN during start-up.	bool	N
MODEM_HL7800_APN_NAME	Defines the APN to be set when the SET_APN_NAME_ON_STARTUP option is enabled.	string	-
MODEM_HL7800_DEFAULT_RAT	Defines the RAT to be used upon start-up. If set to -1, the RAT is not changed. If set to another value, this RAT is always selected at start-up. Note that the RAT can be changed at run-time, although the configuration value defined here will be reverted to following reset.	integer	-1
MODEM_HL7800_CONFIGURE_BANDS	When enabled, this setting allows selection of what LTE bands should be enabled for usage by the modem.	bool	Y
MODEM_HL7800_BAND_1	Enables LTE Band 1 (Transmit 1920–1980MHz, Receive 2110–2170MHz).	bool	Y
MODEM_HL7800_BAND_2	Enables LTE Band 2 (Transmit 1850–1910MHz, Receive 1930–1990MHz).	bool	Y
MODEM_HL7800_BAND_3	Enables LTE Band 3 (Transmit 1710–1785MHz, Receive 1805–1880MHz).	bool	Y
MODEM_HL7800_BAND_4	Enables LTE Band 4 (Transmit 1710–1755MHz, Receive 2110–2155MHz).	bool	Y
MODEM_HL7800_BAND_5	Enables LTE Band 5 (Transmit 824–849MHz, Receive 869–894MHz).	bool	Y
MODEM_HL7800_BAND_8	Enables LTE Band 8 (Transmit 880–915MHz, Receive 925–960MHz).	bool	Y
MODEM_HL7800_BAND_9	Enables LTE Band 9 (Transmit 1749.9–1784.9MHz, Receive 1844.9–1879.9MHz).	bool	N
MODEM_HL7800_BAND_10	Enables LTE Band 10 (Transmit 1710–1770MHz, Receive 2110–2170MHz).	bool	N
MODEM_HL7800_BAND_12	Enables LTE Band 12 (Transmit 699–716MHz, Receive 729–746MHz).	bool	Y
MODEM_HL7800_BAND_13	Enables LTE Band 13 (Transmit 777–787MHz, Receive 746–756MHz).	bool	Y

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

Configuration setting	Purpose	Type	Default
MODEM_HL7800_BAND_14	Enables LTE Band 14 (Transmit 788–798 MHz, Receive 758–768 MHz)	bool	N
MODEM_HL7800_BAND_17	Enables LTE Band 17 (Transmit 704–716 MHz, Receive 734–746 MHz)	bool	N
MODEM_HL7800_BAND_18	Enables LTE Band 18 (Transmit 815–830 MHz, Receive 860–875 MHz)	bool	N
MODEM_HL7800_BAND_19	Enables LTE Band 19 (Transmit 830–845 MHz, Receive 875–890 MHz)	bool	N
MODEM_HL7800_BAND_20	Enables LTE Band 20 (Transmit 832–862 MHz, Receive 791–821 MHz)	bool	Y
MODEM_HL7800_BAND_25	Enables LTE Band 25 (Transmit 1850–1915 MHz, Receive 1930–1995 MHz)	bool	N
MODEM_HL7800_BAND_26	Enables LTE Band 26 (Transmit 814–849MHz, Receive 859–894MHz)	bool	N
MODEM_HL7800_BAND_27	Enables LTE Band 27 (Transmit 807–824 MHz, Receive 852–869 MHz)	bool	N
MODEM_HL7800_BAND_28	Enables LTE Band 28 (Transmit 703–748 MHz, Receive 758–803 MHz)	bool	Y
MODEM_HL7800_BAND_66	Enables LTE Band 66 (Transmit 1710–1780 MHz, Receive 2110–2200 MHz)	bool	N
MODEM_HL7800_LOW_POWER_MODE	Enables modem low power modes	bool	N
MODEM_HL7800_EDRX	Enables eDRX modem low power mode	bool	N
MODEM_HL7800_PSM	Enables PSM modem low power mode	bool	N
MODEM_HL7800_EDRX_VALUE	Sets the desired eDRX value. Refer to **[M1]**, section 10.5.5.32. Also refer to **[N1]** and **[O1]** for further details	4-bit	b’0101’
MODEM_HL7800_PSM_PERIODIC_TAU	Sets the desired PSM TAU length. Refer to **[P1]**, section 4.5.4. Also refer to **[N1]** and **[O1]** for further details	8-bit	b’10000010’
MODEM_HL7800_PSM_ACTIVE_TIME	Sets the desired PSM active time. Refer to **[P1]**, section 4.5.4. Also refer to **[N1]** and **[O1]** for further details	8-bit	b’00001111’
MODEM_HL7800_RX_STACK_SIZE	Defines the size of the stack in bytes used by the receiving thread of the modem driver	int	1024
MODEM_HL7800_RX_WORKQ_STACK_SIZE	Defines the size of the stack in bytes used by the work queue owned by the receiving thread of the modem driver	int	2048
MODEM_HL7800_INIT_PRIORITY	Defines the priority of the main modem thread. Note that this must be set lower than that of the Zephyr networking subsystem such that the modem is initialised first	int	80

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

10.2 Run-Time Configuration

Run-time configuration of the modem driver is possible via the interface published in the hl7800.h header file. The interface functions available are described in Table 8.

Table 8: HL7800 modem driver run-time configuration

Function name	Arguments	Details
mdm_hl7800_power_off	-	Powers the modem off, reducing its power consumption to the minimum.
mdm_hl7800_reset	-	Resets the modem and applies its start-up configuration.
mdm_hl7800_wakeup	[in] **bool wakeup**	If asleep, wakes the modem from sleep mode, if awake, sends the modem to sleep. ***NOTE:*** *This function is for debug purposes only. It should not be used in normal operation by any app.*
mdm_hl7800_send_at_cmd	[in] **const uint8_t *data**	Sends the AT command specified in the data parameter to the modem. ***NOTE:*** *This function is for debug purposes only. It should not be used in normal operation by any app.*
mdm_hl7800_update_apn	[in] **char *access_point_name**	Sets the modem APN via the access_point_name parameter.
mdm_hl7800_update_rat	[in] **enum mdm_hl7800_radio_mode value**	Sets the modem RAT.
mdm_hl7800_valid_rat	[in] **uint8_t value**	Determines if the RAT passed via the value argument is supported by the modem.

10.3 Run-time Parameterisation

The functions described in Table 9 allow the modem parameters to be interrogated during run-time.

Table 9: HL7800 modem driver run-time parameterisation

Function name	Arguments	Details
mdm_hl7800_get_signal_quality	[out] **int *rsrp,** [out] **int *sinr**	Returns the Reference Signals Received Power in the **rsrp** parameter and the Signal to Interference plus Noise Ratio in the **sinr** parameter.
mdm_hl7800_get_iccid	-	Returns the modem SIM card ICCID.
mdm_hl7800_get_sn	-	Returns the modem serial number.
mdm_hl7800_get_imei	-	Returns the modem IMEI.
mdm_hl7800_get_fw_version	-	Returns the modem firmware revision number.

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

11 REFERENCES

Ref	Description
[A]	Pinnacle 100 OOB Demo Source Code https://github.com/LairdCP/Pinnacle_100_oob_demo
[B]	Pinnacle 100 OOB Demo Markdown https://github.com/LairdCP/Pinnacle_100_oob_demo/blob/master/README.md#lte-m-and-aws
[C]	Pinnacle 100 Modem Datasheet https://connectivity-staging.s3.us-east-2.amazonaws.com/2020-06/CS-DS-PINNACLE-100%20v1_4_2.pdf
[D]	LTE-M Overview https://www.gsma.com/iot/mobile-iot-technology-lte-m/
[E]	NB-IoT Overview https://www.gsma.com/iot/narrow-band-internet-of-things-nb-iot/
[F]	Sierra Wireless HL7800 Home Page https://www.sierrawireless.com/products-and-solutions/embedded-solutions/products/hl7800/
[G]	Nordic nRF52840 Home Page https://www.nordicsemi.com/Products/Low-power-short-range-wireless/nRF52840
[H]	Zephyr RTOS https://www.zephyrproject.org/
[I]	Pinnacle 100 Development Kit User Manual https://connectivity-staging.s3.us-east-2.amazonaws.com/2020-04/CS-GUIDE-PINNACLE-100-DVK%20HW%20v1_1.pdf
[J]	Pinnacle 100 Development Kit Schematic https://connectivity-staging.s3.us-east-2.amazonaws.com/2020-01/Schematic%20-%20Pinnacle%20100%20DVK.pdf
[K]	Pinnacle 100 Development Kit Zephyr Boards File https://github.com/LairdCP/zephyr_boards/tree/master/pinnacle_100_dvk
[L]	M.2 Interface Specification https://pcisig.com/specifications/pciexpress/M.2_Specification/
[M]	SWD Description https://developer.arm.com/architectures/cpu-architecture/debug-visibility-and-trace/coresight-architecture/serial-wire-debug
[N]	TRACE Description https://developer.arm.com/documentation/ihi0014/q/etmv3-signal-protocol/trace-port-interface
[O]	Macronix MX25R6435F Datasheet https://www.macronix.com/Lists/Datasheet/Attachments/7428/MX25R6435F,%20Wide%20Range,%2064Mb,%20v1.4.pdf
[P]	Laird Connectivity BL654 Module Home Page https://www.lairdconnect.com/wireless-modules/bluetooth-modules/bluetooth-5-modules/bl654-series-bluetooth-module-nfc
[Q]	Bosch BME280 Sensor Home Page https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/

Ref	Description
[R]	Laird Connectivity smartBASIC Home Page https://www.lairdconnect.com/smartbasic-for-ble
[S]	BL654-BME280 smartBASIC Source Code https://github.com/LairdCP/BL654_BME280
[T]	Environmental Sensing Service https://www.bluetooth.com/specifications/assigned-numbers/environmental-sensing-service-characteristics/
[U]	BlueTooth SIG PTS Home Page https://www.bluetooth.com/develop-with-bluetooth/qualification-listing/qualification-test-tools/profile-tuning-suite/
[V]	BT510 IoT Sensor Home Page https://www.lairdconnect.com/iot-devices/iot-sensors/bt510-bluetooth-5-long-range-ip67-multi-sensor
[W]	BT510 Zephyr Boards File https://github.com/LairdCP/zephyr_boards/tree/master/bt510
[X]	Pinnacle 100 OOB Demo LTE-M Markdown https://github.com/LairdCP/Pinnacle_100_oob_demo/blob/master/docs/readme_ltem_aws.md
[Y]	Laird Connectivity Bluegrass https://documentation.lairdconnect.com/Builds/IG60-BL654-BT510-KIT/latest/Content/Topics/5%20-%20Using%20the%20Device/Starter%20Kit%20Quick%20Start%20Demo/Quick%20Start%20Demo.htm
[Z]	Open Systems Interconnection Reference Model https://www.iso.org/ics/35.100/x/
[A1]	MQTT Specification https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.pdf
[B1]	TLS Specification https://tools.ietf.org/html/rfc5246
[C1]	TCP Specification https://tools.ietf.org/html/rfc793
[D1]	Pinnacle 100 OOB Demo NB-IoT Markdown https://github.com/LairdCP/Pinnacle_100_oob_demo/blob/master/docs/readme_nbiot_lwm2m.md
[E1]	Leshan Server Home Page https://www.eclipse.org/leshan/
[F1]	LwM2M Specification http://openmobilealliance.org/RELEASE/LightweightM2M/V1_1-20180612-C/OMA-TS-LightweightM2M_Core-V1_1-20180612-C.pdf
[G1]	CoAP Specification https://tools.ietf.org/html/rfc7252
[H1]	UDP Specification https://tools.ietf.org/html/rfc768
[I1]	DTLS Specification https://tools.ietf.org/html/rfc6347
[J1]	Zephyr Non-Volatile Subsystem Home Page https://docs.zephyrproject.org/1.13.0/subsystems/nvs/nvs.html

Ref	Description
[K1]	Zephyr Networking Subsystem Home Page https://docs.zephyrproject.org/1.13.0/subsystems/networking/networking.html
[L1]	Zephyr Network Interface https://docs.zephyrproject.org/latest/reference/networking/net_if.html
[M1]	3GPP TS 24.008 - Mobile radio interface Layer 3 specification https://www.3gpp.org/ftp/Specs/archive/24_series/24.008/24008-g50.zip
[N1]	HL7800 Low Power Modes Application Note https://source.sierrawireless.com/resources/airprime/application_notes_and_code_samples/airprime_hl7800_low_power_modes_application_note/#sthash.zSgvxmyS.dpbs
[O1]	HL7800 LPWA Power Saving Features https://source.sierrawireless.com/resources/airprime/application_notes_and_code_samples/airprime_hl78xx_lpwa_power_saving_features/#sthash.RwYFuncj.dpbs
[P1]	3GPP TS 23.682 - Architecture enhancements to facilitate communications with packet data networks and applications https://www.3gpp.org/ftp/Specs/archive/23_series/23.682/23682-g70.zip

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

12 Definitions, Abbreviations, and Acronyms

Term	Definition
API	Application Programming Interface
APN	Access Point Name
AWS	Amazon Web Services
BLE	Bluetooth Low Energy
CoAP	Constrained Application Protocol
DNS	Domain Name Server
DTLS	Datagram Transmit Layer Security
DVK	Development Kit
eDRX	Extended Discontinuous Reception
GPIO	General Purpose Input/Output
ICCID	Integrated Circuit Card Identifier
ICD	In-Circuit Debugger
IMEI	International Mobile Equipment Identity
I/O	Input/Output
IoT	Internet of Things
IP	Internet Protocol
JSON	Java Standard Object Notation
LTE-M	Long Term Evolution Machine Type Communication
LwM2M	Lightweight Machine to Machine
MQTT	Message Queuing Telemetry Transport
NB-IoT	Narrow Band Internet of Things
NFC	Near Field Communication
OOB	Out of Box
PSM	Power Saving Mode
PTS	Profile Tuning Suite
QSPI	Quad Serial Peripheral Interface
RAT	Radio Access Technology
SDK	Software Development Kit
SIM	Subscriber Identification Module
SOC	System on Chip
Softdevice	Bluetooth protocol stack supplied in precompiled binary format by Nordic
TAU	Tracking Area Update
TCP	Transmit Control Protocol
TLS	Transport Layer Security
UART	Universal Asynchronous Receiver Transmitter
UDP	Universal Datagram Protocol
USB	Universal Serial Bus
VSP	Virtual Serial Port

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

13 Revision History

Version	Date	Notes	Contributor(s)	Approver
1.0	14 August 2020	Initial Release	Greg Leach	Jonathan Kaye
1.1	17 August 2020	Fixed broken table link; changed LWMTM to LWM2M in image 7		Jonathan Kaye

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610